D3.6

Scientific Use Case Requirements Documentation

Work package:	**WP3 Technical specification and coordination**	
Author(s):	Wouter Klijn, Dirk Pleiter	JUELICH
Editor	Anne Carstensen	JUELICH
Reviewer #1	Carlo Cavazzoni	CINECA
Reviewer #2	Boris Orth	JUELICH
Dissemination Level	Public	
Nature	Report	

Date	Author	Comments	Version	Status
28.06.2018	Wouter Klijn	Initial structure and info	V0.1	Draft
29.06.2018	Wouter Klijn	Add current known use cases	V0.2	Draft
30.06.2018	Wouter Klijn	All use cases added, analysis	V0.3	Draft
30.06.2018	Dirk Pleiter	Editorial changes	V0.4	Draft
01.07.2018	Boris Orth	Internal review	V0.5	Draft
02.07.2018	Wouter Klijn	Merged review by Carlo Cavazzoni	V0.6	Draft
03.07.2018	Wouter Klijn	Insert delayed contributions Merge additional review Boris Orth Remove Automatic date fields	V0.7	Draft
04.07.2018	Anne Carstensen	Editorial changes	V0.8	Draft
06.07.2018	Anne Carstensen	Editorial updates	V1.2	Final

Changes for Resubmission				
20.08.2018	Anne Carstensen	Editorial updates	V1.3	Draft
01.09.2018	Wouter Klijn	Review of use case documents regarding missing information	V1.4	Draft
28.09.2018	Wouter Klijn/Anne Carstensen	Update of use case documents	V1.5	Draft
01.10.2018	Dirk Pleiter/Wouter Klijn	Update of section 1 and 2	V1.6	Draft
05.10.2018	Anne Carstensen	Merge of all updates and editorial changes	V1.7	Draft
08.10.2018	Wouter Klijn	Internal review	V1.8	Draft
11.10.2018	Carlo Cavazzoni	Internal review	V1.8.1	Draft
11.10.2018	Boris Orth	Internal review	V1.8.2	Draft
12.10.2018	Anne Carstensen	Merge of internal review feedback	V1.9	Draft
14.10.2018	Dirk Pleiter	Editorial updates	V2.0	Final
27.11.2018	Anne Carstensen, Dirk Pleiter	Integration of updates based on reviewer feedback	V3.0	Draft
30.11.2018	Dirk Pleiter	Editorial updates	V3.1	Final

The ICEI project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement No 800858.

Executive Summary

This report documents information on science and use cases of the HBP that are expected to exploit the infrastructure services developed and deployed within SGA ICEI. The work was performed by Task 3.1, which should continuously analyse, in close collaboration with HBP scientists, the HBP's scientific use cases, select those that will need the ICEI infrastructure, and analyse the requirements. In parts based on the information collected for this deliverable, the architectural specification of the ICEI infrastructure as outlined in the original proposal has been updated and refined in deliverable D3.1 "Common Technical Specifications".

Contents

Acronyms

AAI	Authentication and authorization infrastructure
ACD	Active Data Repositories
ACL	Access Control list
API	Application Programming Interface
ARD	Archival Data Repositories
BSC	Barcelona Supercomputing Center
CapEx	Capital Expenditure
CDP	Co-design Project
CEA	Commissariat à l'énergie atomique et aux énergies alternatives
CINECA	Consorzio Interuniversitario
CLI	Command line interface
CSCS	Centro Svizzero di Calcolo Scientifico
DL	Data Location Service
DM	Data Mover Service
DT	Data Transfer Service
FPA	Framework Partnership Agreement
FURMS	Fenix User and Resource Management Services
GoP	Group of Procurers
GUI	Graphical User Interface
HBP	Human Brain Project
HPAC	High Performance Analytics and Computing
HPC	High Performance Computing
HPDA	High Performance Data Analytics
HPST	High-Performance Storage Tier
IaaS	Infrastructure as a service

IAC	Interactive Computing Services
ICCP	Interactive Computing Cloud Platform
ICEI	Interactive Computing E-Infrastructure for the Human Brain Project
ICN	Interactive Computing Node
IdP	Identity provider
IPR	Intellectual property rights
JP	Joint Platform
JSC	Jülich Supercomputing Centre
LCST	Large-Capacity Storage Tier
MS	Monitoring Services
NDA	Non-Disclosure Agreement
NETE	External Interconnect
NETI	Internal Interconnect
NMC	Neuromorphic Computing
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OIDC	OpenID Connect
OpEx	Operational Expenditure
PaaS	Platform as a service
PCP	Pre-Commercial Procurement
PI	Principal Investigator
PID	Persistent Identifier
PIE	Public Information Event
PRACE	Partnership for Advanced Computing in Europe
Q&A	Questions and Answers
QoS	Quality of Service
R&D	Research & Development
R&I	Research & Innovation
RBAC	Role-Based Access Control
RFI	Request For Information
SCC	Scalable Computing Services

SGA	Specific Grant Agreement
SIB	Science Infrastructure Board
SLA	Service Level Agreement
SP	Subproject
TCO	Total Cost of Ownership
TGCC	Très Grand Centre de Calcul
UI	User Interface
US	User Support Services
VM	Virtual Machine Services

1. Introduction

This deliverable contains information on science and use cases that are driving the ICEI requirements. We use the term “science case” when a specific scientific challenge is the basis for describing IT infrastructure needs. The term “use case” is used for more generic cases for describing needs through a planned scenario for using the IT infrastructure.

To identify the needs of the HBP users we interacted with each of the science or use case owners on the basis of a template (Appendix C). The feedback received per case is of different level of quality. This probably reflects the situation that a number of science and use cases are still in an emerging state. Feedback has been documented as received and reflects the thinking of the science and use case owners, who did not necessarily have a chance to obtain a sufficiently deep understanding of ICEI architecture requirements.

The science and use cases provide a varying challenge for the possible ICEI infrastructure. We performed a classification in terms of their relevance for driving the infrastructure requirements:

- "High": The use case is considered challenging and special care is needed to meet the requirements.
- "Medium": Care is needed to ensure that the use case can be realized within the ICEI infrastructure, but use case is not driving any specific requirements.
- "Low": No specific analysis of the ICEI infrastructure is needed for the given use case.

For details on how the needs identified for each of the science and use cases are translated into requirements we refer to the section “Co-Design Process” in deliverable D3.1 (“Common Technical Specifications”).

The methodology used for collecting the science and use cases and their needs is described in section 2.1. An estimation of the coverage of the HBP needs is given in section 2.2. A table of the actual use cases with links to the detailed forms as filled out by the domain scientists and the classification described above can be found in Table 2. Limitations and future work are briefly discussed in section 2.4. The use cases themselves can be found in Appendix D.

2. Science and use cases and the collection of their needs

2.1 Collection method

For the collection of the information about the science and use cases, as well as the collection of information on their needs, a template was created on the basis of the use case specification used in HBP SGA1. The template was developed in collaboration with technical and scientific experts of the SimLab Neuroscience at JUELICH-JSC and the HBP. The final version was reviewed by the ICEI Technical Board. The template with accompanying explanations can be found in Appendix C 7.1.

The templates were not sent empty to potential science owners: Available information from known sources as detailed in section 2.2 was included in the document. Additionally, a first diagram, with a break-down of the use case in smaller components, potentially mapping on ICEI resources, was added. Where possible, the characterizations of the nodes in the diagram were pre-filled. The pre-filled template was sent to the PI as identified for a use case, including a cover letter and a detailed explanation of the requested effort (see Appendix C 7.2).

The specific task for the science owner was to validate the scientific part and, where possible, to complete the template with technical resource information. To expedite the answering of this technical part, we explicitly asked to be brought into contact with computer experts connected to the project. Where possible, SimLab Neuroscience experts already familiar with use cases were enlisted, specifically to help in the coordination.

The filled templates received were checked for completeness. Where necessary and possible, clarifications were requested, such as: estimate of the maximum resource usage, number of expected concurrent users of systems, and an absolute number of experiments to be performed over the course of the HBP.

Closer to the deadline, all incomplete use case owners were contacted again, and a reduced set of information was requested. An example of this reduced set can be found in Appendix C 7.3.

2.2 Coverage

The ICEI use case collection was not performed in a vacuum. Members of the HBP have spent considerable time on the collection of science and use cases, especially in the preparation of SGA2. To avoid repeated efforts by HBP scientists, a literature study was performed of the most relevant documents and presentations:

- HBP SGA2 Grant Agreement

- HBP SGA1 Review presentations

Additionally, slides of the Co-Design Workshop on Interactive Supercomputing (9 February 2018, ETH Zurich) and the weekly HBP-JP coordination meetings were taken into account. The ICEI use case activity was further advertised during a SIB meeting (7-9 May, Alpbach, Austria) and during JP Coordination meetings. Core members of the HBP have provided direct input, including: Prof. Amunts, Prof. Schürmann and Prof. Lippert. During the two use case collection rounds, upwards of 230 e-mail interactions occurred (overview in Appendix B). Additionally, numerous telephone and skype conversations took place.

The combination of these efforts leads us to the conclusion that the majority of relevant data and compute resource requirements should be covered by the collection of use cases as presented in this document. Table 1 contains a count of the use cases collected for ICEI from the different sources. There are 54 cases that are potentially relevant. 49 of these are partially or completely covered in the current ICEI use case and requirements collection, corresponding to a 91% coverage.

Table 1: The relevant and ICEI covered use cases. Tabulated per source.

Type	#Relevant science/use cases	#Subject of an ICEI science/use case document	Percentage covered
SGA2 SP Use cases	27	24	89%
SGA2 CDP Use cases	5	4	80%
HBP JP Use cases	13	12	92%
ICEI Co-Design Workshop	9	9	100%
Total	**54**	**49**	**91%**

2.3 Clustering of use cases

An overview of the science and use cases, their owners, references to the Appendix D with the filled templates and the use case classification in terms of relevance as defined in section 1 is provided in Table 2.

Based on the provided documentation a manual clustering of the use cases was performed and the following distinguishing features were marked:

- **Simulation**: Use of simulation at any point in the processing.
- **Multi-scale coupled simulation**: Use of multiple simulators at different scales exchanging data at runtime.
- **Co-deployment of apps**: Processing pipelines needing multiple applications running concurrently and exchanging data at runtime.

- **Streaming visualization**: In-situ / in-transit visualization of applications running on HPC resources.[1]
- **Machine learning**: Machine learning somewhere in the processing pipeline.
- **In-the-loop machine learning**: Machine learning on data from an online data source.
- **Big data processing**: Big data collection, pre-processing, curation, processing and storing.
- **Big data visualization**: Visualization of big data sets.

As illustrated in Table 3, two major clusters appear based on these features. The first cluster (cluster #1 in light blue) constitutes the simulation use cases, typically with online visualization of the applications running on HPC (use cases #1, 3, 5, 6, 7, 10, 11, 16). Two sub-clusters can be seen within that cluster: Multi-scale coupled simulation (use cases #1 and 16) and In-the-loop machine learning (use cases #1 and 3). The second cluster (cluster #2 in light green) involves big data use cases, collection, curation, processing and storing of big data sets. This cluster has one sub-cluster: Big data with machine learning (use cases #8 and 9). Use case #9 is an outlier being the only use case of the second cluster that needs simulation.

Two use cases fall outside of the clustering based on the current feature set, namely use cases #12 and 13. Use case #12 is a simulation project not needing online visualization. The output is stored raw, with the size of this data making it big data. Use case #13 has some features of (big) data visualization, but the actual data processed is small in comparison with the other use cases.

2.4 Limitations and future work

Although the current use case collection effort has been advertised on multiple occasions, it is possible that informative cases have been missed. During the HBP Summit 2018[2], an update was given to the larger HBP community, and as detailed in section 3, we expect this report to remain a living document.

Most notably, no science and use cases have been collected for SP8, which underwent a significant reorganisation. The new concept for a Medical Informatics Platform (MIP) within the HBP foresees local infrastructure at various hospitals ("MIP local"), which will be augmented by a federation layer that facilitates queries to enable the retrieval of information from all local infrastructure components. This local infrastructure will not be part of the ICEI infrastructure to ensure a clear separation from infrastructure components that are suitable for holding personal, i.e. highly sensitive data. ICEI infrastructure services may possibly be used for operating the federation layer. The requirements are expected to be simple and aligned with requirements from other platform services. A more detailed analysis of SP8 science and use cases has therefore been postponed.

[1] The terminology "in-situ" versus "in-transit" seems to have first been introduced in [3]. In-situ refers to cases where the primary compute resources are used for visualisation, while in-transit processing refers to offloading computations to a set of secondary resources, which requires data to be transferred over the network.

[2] HBP Summit 2018, 15-18 October 2018, Maastricht, NL

Regarding the collected use cases, a number of use case owners did not supply sufficient information to generate complete template documents. The maturity of projects might have prevented the answering of the more detailed technical questions. The following use cases specifically were not fully transparent, hindering the analysis:

- Ilastik as a service on the HBP Collaboratory (#13 in Appendix D, section 13)[3]
- Towards a novel decoder of brain cytoarchitecture using large scale simulations (#9 in Appendix D, section 17)
- Neurorobotics Platform, large-scale brain simulations (#11 in Appendix D, section 20)

The use case collection effort as reported here will flow into the larger HBP use case efforts of SP7 and the HBP-JP. A similar template as used in the current effort will be also be used for these other efforts. Possible adaptations to capture software and platform needs will be addressed at a later stage.

[3] The developers of Ilastik are, however, starting to use ICEI resources already available at CSCS. Their request for resources is rather moderate compared to the overall currently available ICEI resources for HBP.

Table 2: Title, Principal Investigator (PI), reference to the available use case information in Appendix D and classification in terms of relevance as defined in section 1.

#	Working Title	PI	Reference	Relevance
1	Data-driven cellular models of brain regions, Olfactory Bulb	Migliore	Appendix D: 9	High
3	Learning-to-learn (LTL) in a complex spiking network on HPC and Neuromorphic hardware interacting with NRP	Maass, Meier	Appendix D: 22	Medium
5	Large scale simulations of models: Cerebellum	D' Angelo	Appendix D: 11	High
6	Large scale simulations of models: Hippocampus	Migliore	Appendix D: 10	High
7	Elephant big data processing	Grün, Denker	Appendix D: 12	High
8	Mouse Brain Atlas	Pavone	Appendix D: 21	High
9	Towards a novel decoder of brain cytoarchitecture using large scale simulations	Poupon, Axer	Appendix D: 17	High
10	Multi-scale co-simulation: Connecting Arbor/Neuron, NEST and TVB to simulate the brain	Morrison, Destexhe, Diesmann, Jirsa	Appendix D: 23	High
11	Neurorobotics platform, large-scale brain simulations	von Arnim, Cruz	Appendix D: 20	High
12	Blue Brain Project Microcolumn	Schürmann	Appendix D: 18	Medium
13	Ilastik as a service on the HBP Collaboratory	Kreshuk	Appendix D: 13	High
14	Online visualization of multi-resolution reference atlases	Amunts	Appendix D: 14	High
15	Data management and big data analytics for high throughput microscopy	Dickscheid	Appendix D: 19	High
16	Multi-area macaque NEST simulation with life visualization and interaction	v. Albada, Diesmann	Appendix D: 16	Medium
17	Data management and big data analytics for large cohort neuroimaging	Caspers, Eickhoff	Appendix D: 15	High

Table 3: Use case features marked for each use case. Colouring is used to distinguish the major clusters, light blue for cluster #1 and light green for cluster #2. Additionally, the number of use cases asking for a specific feature is totalled in the last row.

	Feature							
#	**Simulation**	**Multi-scale coupled simulation**	**Co-deployment of apps**	**Streaming visualization**	**Machine learning**	**In-the-loop machine learning**	**Big data processing**	**Big data visualization**
1	1	1	1	1	1	1		
3	1		1		1	1		
5	1		1					
6	1		1	1				
7	1		1	1				
10	1	1	1	1				
11	1		1	1				
16	1		1	1				
12	1							1
13				1				1
8					1		1	1
9	1				1		1	1
14							1	1
15					1		1	1
17							1	1
Total	**10**	**2**	**8**	**7**	**5**	**2**	**5**	**7**

3. Concluding remarks

The use cases as collected are based on information provided by the respective science owners. The requirements collected are informative and not prescriptive. Technical limitations or other constraints might prevent the fulfilment of a need by the ICEI infrastructure.

The amount of information available for the different use cases does not reflect the maturity of the science case. The collection and management of use cases will be an ongoing effort over the duration of ICEI. This report will thus stay a living document.

4. References

[1] "ANNEX 1 (Part A) SGA-RIA NUMBER — 800858 — ICEI," 2018.

[2] "HBP SGA2 Grant Agreement 785907 Annex 1 – Description of the Action (Part B)", 2018.

[3] Janine C. Bennett et al., "Combining In-situ and In-transit Processing to Enable Extreme-Scale Scientific Analysis," SC'12, 2012 (doi: 10.1109/SC.2012.31).

5. Appendix A: Summary of changes

Since version 1.2 of this deliverable, which had been the basis for the special review meeting on July 23, 2018, the following main changes have been performed:

- Use cases have been classified in terms of their relevance for driving the requirements of the ICEI infrastructure (see Table 2).
- A clustering of the use cases has been performed (see section 2.3).
- A paragraph has been added in section 2.4, explaining that for some use cases the collected information was not sufficient to generate complete template documents.
- A table detailing the interactions with the HBP science community has been added (see Appendix B).
- The following science and use cases have been dropped:
    - Science case #2 ("Enabling data management and analysis for the Human Brain Atlas"): This use case had already earlier been split in use case #14, #15 and #17.
    - Use case #4 ("Connecting the HBP Collaboratory with HPC resources"): This use case was not found to be useful for driving requirements, as it turned out to be too generic. The key need for connecting ICEI infrastructure services and the Collaboratory is contained in other use cases.

In the following the (major) changes in this version of the document are listed for each use case.

- Use case #1
    - Added missing captions
    - Additional details on expected bottlenecks added:
        - Data transport: 1,3,5: Absolute numbers added
        - Data ingest: Live Visualization: Number of expected users
        - Data ingest: Spinnaker: Co-location of neuromorphic hardware is not possible
        - Data Repository: iStore: Information on current performance
        - Data Repository: oStore: Data production per year
        - Processing station: Model creation: scaling of current implementation
    - Estimation of infrastructure services requirements

- Use case #3
    - Added missing captions
    - Additional details on expected bottlenecks added:

        - Data transport: 4, 6: Basic information on data products
        - Data repository: Long term storage: Estimation of storage needs
        - Processing station: L2L: Basic information on software needs
        - Processing station: SIM: NEST: Expected compute requirements added
        - Processing station: sensor, actor, environment: NRP. Added link to RNP ICEI use case
    - Estimation of infrastructure services requirements

- Use case #4
    - This use case ("Connecting the HBP Collaboratory with HPC resources ") was found not to be useful for driving requirements. The need to connect ICEI infrastructure services and the Collaboratory is contained, with a scientific motivation, in other use cases.

- Use case #5
    - Additional details on expected bottlenecks added:
        - Data object: Circuit building: Absolute numbers
        - Data object: Simulation and analysis: Absolute numbers
        - Data transport: simulation object – HPC centre to Knowledge graph transport: No current estimates are possible
        - Data ingest: Name: Basic information on input models
        - Processing station: HPC and NRP: Details in the base information.

- Use case #6
    - Only editorial changes, use case was described at the correct level of detail

- Use case #7
    - Only editorial changes, use case was described at the correct level of detail

- Use case #8
    - Added need for deep learning methods
    - Updated diagram now with high degree of detail
    - Additional details on expected bottle-necks added:
        - Data object: 1, Raw data
        - Data object: 2: Raw stitched images
        - Data object: 3: Compressed and downsampled images
        - Data object: 4: Acquisition metadata

        - Processing node: 1: Image stitching
        - Processing node: 2: Image compression and downsampling
        - Processing node: 3: Neuronal soma detection
        - Processing node: 4: Neuronal segmentation

- Use case #9
    - Write-up of Node characterization
    - Write-up of platform needs

- Use case #10
    - Clarifications in use case description
    - Node characterization
        - Raw requirements
        - Characterization tables are now included and filled where possible
    - Estimation of infrastructure services requirements

- Use case #11
    - Use case description added
    - Estimation of infrastructure services requirements

- Use case #12
    - All information in this template is new:
        - Use case description
        - Diagram figures
        - Node characterization
        - Infrastructure requirements

- Use case #13
    - Some project specific information added to the explanation chapter
    - Diagrams added
    - Estimation of infrastructure services requirements
    - Use case scenario added as reference information

- Use case #14
    - Redundant information shared with use case #15 and #17 removed
    - Estimation of infrastructure services requirements

- Use case #15

    - Redundant information shared with use case #14 and #17 removed
    - Diagram captions
    - Estimation of infrastructure services requirements

- Use case #16
    - Additional use case description
    - Updated figure captions
    - Additional details on expected bottlenecks added:
        - Data transport: 1: Base information
        - Data ingest: Istore: Base information
        - Data repository: oStore: Base information
        - Processing station: NEST: Highly detailed breakdown of computational needs
        - Processing station: Elephant: Detailed listing of expected processing
    - Estimation of infrastructure services requirements

- Use case #17
    - Redundant information shared with use case #14 and #15 removed

6. Appendix B: Number of e-mail contacts during use case collection

Table 4: Number of e-mail contacts for each use case. The first round ended in July and the second round ended beginning of October.

#	Working Title	E-mail contacts 1st round	E-mail contacts 2nd round
1	Data-driven cellular models of brain regions, Olfactory Bulb	6	7
3	Learning-to-learn (LTL) in a complex spiking network on HPC and Neuromorphic hardware interacting with NRP	6	10
5	Large scale simulations of models: Cerebellum	8	9
6	Large scale simulations of models: Hippocampus	8	4
7	Elephant big data processing	14	4
8	Mouse Brain Atlas	3	6
9	Towards a novel decoder of brain cytoarchitecture using large scale simulations	8	6
10	Multi-scale co-simulation: Connecting Arbor/Neuron, NEST and TVB to simulate the brain	9	5
11	Neurorobotics platform, large-scale brain simulations	15	8
12	Blue Brain Project Microcolumn	6	8
13	Ilastik as a service on the HBP Collaboratory	7	8
14	Online visualization of multi-resolution reference atlases	4	7
15	Data management and big data analytics for high throughput microscopy	6	10
16	Multi-area macaque NEST simulation with life visualization and interaction	6	9
17	Data management and big data analytics for large cohort neuroimaging	9	9
Total		**127**	**111**

7. Appendix C: Use case template

7.1 Template: Use Case Description and Specification

The following section contains the unedited use case template as sent to domain scientists and associated computer experts. The references in the document have not been updated after inclusion in the current report.

Title: Version 2.0.1

Use Case Description and Specification

<Date> Author Names,
Partners
Institutions
Principal
Investigators

Date	Version / Change

7.1.1 Introduction

This use case description and specification document provides a tool for developers and scientists to collaboratively transform a free form description of a science use case into technical specifications. Specifications that guide the implementation of hardware and software fulfilling the science use case. This document should help a project in a number of ways: its structured methodology will help to find the essential parts, and it will assist in separation of the ***must*** haves and ***nice*** to haves [1]. The specifications should result in a standalone document that can be given to new partners of the project as introduction into the science and technical details of the project. On a more abstract level this document could be seen as a contract formalizing the expectations of both, the engineer and the scientist.
An important guideline when creating a use case analysis document is the separation of user requirements and technical details. A user is ultimately only interested in the functionality of a software / hardware product and not in the underlying technical details of the implementation. Separating these concerns is a non-trivial matter: This document will therefore typically be written in an iterative manner, with the document bouncing from scientist to developer getting more detailed on each iteration. It will also be living document: details of the project can and will change over time; Components might be hard to implement and trade-offs might be made depending on availability of

manpower. The amount of work needed for this document might appear large, however it is work that, for a typical software/science project, should be performed anyways.
The different elements/chapters in the template should be kept in order and contain the content described. This will allow comparison of use cases and allow identification of shared / overlapping functionality. This document and the accompanying PowerPoint introduce a set of visual components that can be used to describe the use cases and systems (Section 1.2). The symbols should cover the majority of systems encountered, but if the need arises, new elements can be introduced. Do keep in mind that this will complicate comparison of the diagrams created. The main goal for collecting the information is to foster the reuse of efforts and components. Although the introductory chapters can be removed, it will limit the use as an introduction for new project partners.
In the next sections the goal of the individual parts of the template will be introduced. The first section (1.1) details the use case description, it should provide the scientific reasoning behind the case. Section 1.2 explains the set of visual components that can be used to create the model diagrams. In section 1.3 we provide the typical data point that can be used to characterize the different components in more technical detail. In section 1.4 we explain list of potential infrastructure requirements specific questions. High-level needs and services that can be cross-checked with the node characterizations.
Section 2 is the actual template, it contains just the titles and list of infrastructure questions. Other components can be copied from the introduction chapter 1. If you add multiple diagrams/systems it is best to copy the template multiple times, or, use different documents. This will improve coherence in the descriptions.

7.1.1.1 Use Case Description

The workflow description is a high-level description of the workflow of the use case. It is typically written by the scientist and provides the reasons why to build or use a software or hardware system. Topics that might be encountered in this section are: How new (or better, bigger, faster) science is possible with this software. Problems and challenges encountered in current software.
Typically, the workflow is broken down in steps with partial goals for each step. It is advisable to keep implementation and technical details out of this section. Implementation details are not part of the description: An example of such an ***implementation detail*** would be: "The software must be fast, to allow fast turnover of experiments. ***We have to use GPUs***". A complete separation of concerns is hard to arrive at. It is one of the more complicated exercises in system design. Having a starting point is more important that being completely correct. This is one of examples where the dialog with technical experts will help to arrive at a correct description.

An example of a science (and not technology) centric description:
"As a researcher I want to be able to perform a large scale computational experiment. This experiment cannot be performed on my local cluster due the size of parameter space I want to explore. The analysis of the results will need to be performed in my local

institute due to A and B. The access of the results should be structured based on X and Y."

Two widely different technical solutions would support this case:

1. Analysis of results on a virtual machine with data staying in a central location. Results selectable via a database, accessed via a web interface.
2. Transport of results to the local cluster with processing on the local machines with the data stored in clearly labelled directories.

Which of these solutions is implemented can now be made on available resources, software limitations, etc.

7.1.2 Annotated Use Case Diagrams

An annotated use case diagram is a relatively freeform graphical depiction of the textual description as detailed in section 1.1. We would suggest to use the diagram components as shown in Figure 1. As this will allow easy comparison between different use case descriptions. The flowcharts in this document follow the practices as described in [2], [3].

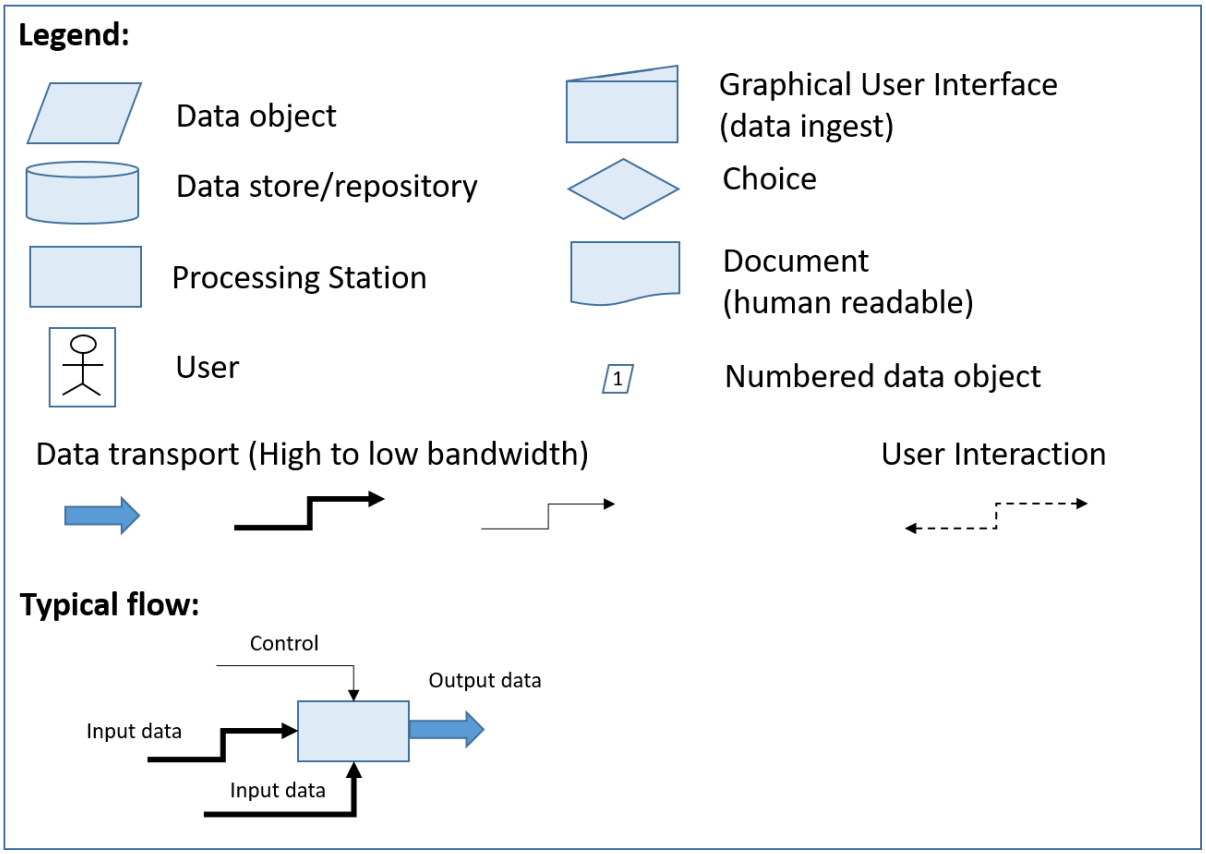


Figure 1: Overview of suggested symbols for a use case diagram. The symbols are based on [2], [3]. The symbol for GUI is a combination of processing station and data object. A suggested typical data and information flow is shown. Additionally, a simple bandwidth range is depicted. An editable version of the diagram below (a PowerPoint presentation) will accompany the current document.

To prevent cluttering of complicated workflow we suggest the following:

- Make use of specialized symbols to allow for a visual distinguishing of salient features (GUI would be an example).
- Use only a small pictogram for data objects annotated with a number.
- Use the suggested locations for the connectors: Control at the top; Inputs from the left or bottom; Outputs leave on the right side.

To reiterate: these are suggestions, the diagrams are in principle freeform and not all symbols might be used in your specific use case.

7.1.3 Node Characterization

In this section a characterization of each component is depicted in the annotated use case diagram. This is done in a table format with typical information points listed. The entries are typically split in different sets: The ***base*** information set without which an informed discussion might be complicated; The description is typically at a user / functional level. Secondly, **technical specifications** of the requirements. The use case is not yet solved thus this information will by necessity be added incrementally and optionally by a domain specialist. The third information set is regarding ***current solutions*** that one is aware of.
Not all information might be available. Fill in what is known at this stage. Having a start point for a dialog is more important that having perfect information, especially in the beginning stages.

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: needed per node, in total
- IO: bandwidth, latency, always on/dedicated
- CPU: large size jobs / farming
- Specialized hardware: (GPU, KNL, FPGAs)
- Storage: size, access rate
- Specialized software: VM/containers
- Specialized features: in-situ visualization

Architecture Requirements:

- Minimal compute performance (excluding acceleration)
- Minimal volatile memory footprint of 192 GByte
- MPI point-to-point bandwidth of 10 GByte/s or higher
- MPI latency of 2 micro-seconds or less
- Access to active data repositories with a bandwidth of up to 8 GByte/s per node
- GPU requirements per node (minimum)
- GPU configuration (minimum HBM)

7.1.3.1 Data objects

Data object: ***Number in diagram***, name

Base information	General description of what data is stored • Formats • Metadata • Database requirements
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. • Short-term (Campaign): Data used throughout the execution of the scientific workflow. • Permanent (Forever): Data outliving the machine used to generate it. Additional information
Current solution	Name
	URL to additional information
	Limitations

7.1.3.2 Data transport

Data transport: Name	
Base information	General description of what data is transported
	Data access patterns (request rate, transfer sizes)
Technical specifications	Maximum required bandwidth
	Average required bandwidth
	Interface requirements for attached entities
	Additional information
Current solution	Name
	URL to additional information
	Limitation

7.1.3.3 Data ingest / GUI

Data ingest: Name	
Base information	Description of input data source
	Description of data introduction (upload? scanner characteristics? simulation characteristics?)
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports
	Additional information
Current solution	Name
	URL to additional information
	Limitation

7.1.3.4 Data repository

Data repository: Name	
Base information	Classification of the data objects (see below)
	Access control requirements

	Access requirements
	Data availability requirements
Technical specifications	Maximum and average capacity requirements
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time.
	In terms of size & file number
	Additional information
Current solution	Name
	URL to additional information
	Limitation

7.1.3.5 Processing stations

Processing station: Name	
Base information	General description of data processing
	Typical processing steps
	Number of processing steps
Technical specifications	Data processing hardware architecture requirements
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses
	Ratio of data processing rate versus data consumption and production rate
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern
	Additional information
Current solution	Name
	URL to additional information
	Limitation

7.1.4 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	**Questions to address**

Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? • What software stacks need to be available? • Is it possible to define memory capacity requirements?
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services?
Virtual Machine Services	• Which parts of the workflow require such services?
Active Data Repositories	• Which parts of the workflow require such services?
Archival Data Repositories	• Which parts of the workflow require such services?
Data Mover Services	• Which parts of the workflow require such services?
Data Transfer Services	• Which parts of the workflow require such services? • Between which ICEI sites is data planned to be transferred? • How much data is expected to be transferred per time unit? • How are transfer patterns expected to change over time?
Data Location Service	• Which parts of the workflow require such services?
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site?
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services?

Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples:
	  ○ Special accounts for running services
	  ○ Needs for fine-granular control of access to data

User Support Services	• Are the specific foreseeable needs for user support services?

7.1.5 Use Case references

7.1.6 References

[1] *MoSCoW Analysis (6.1.5.2)*. International Institute of Business Analysis, 2009.
[2] "Flowchart Symbols Meaning | Standard Flowchart symbol images and usage." [Online]. Available: https://creately.com/diagram-type/objects/flowcharts. [Accessed: 17-Aug-2017].
[3] "Flowchart Symbols and Notation | Lucidchart." [Online]. Available: https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained. [Accessed: 17-Aug-2017].
[4] "UseCaseDescription_and_Specification_v1.".

7.2 Example use case cover letter

Dear <Name>,

Summary: To assure that the ICEI systems meets the requirements for <use case name> I kindly ask you to validate the write up I compiled from multiple sources in the HBP (pages 7-10 in <Prefilled document name>). The final deadline for this is <date>, earlier return will allow fine-tuning of the requirements document.
To coordinate merging I suggest to send me two documents and let me do that.

The technical details on pages 11-22 are best validated and completed by the person with the most detailed knowledge of the HPC in your project. For this use case I do not know who this is. It would be great if you could forward this e-mail and include me in cc. I would suggest to have this section returned <date>.

If you have any questions do not hesitate to ask! I am available per e-mail, phone or skype.

Details:
Within HBP workpackage 7.1 I am tasked with use case specification and requirements documentation. Although this will take the majority of the first year of SGA2, outside priorities (ICEI procurement) necessitate that a number of high priority cases need to be completed at an earlier time.
We are aiming for the end of June for completion of a first iteration.
Your <project > use case is one of these high-priority cases that will allow us to find upper-bounds for HPC and communication requirements.

I send you the HBP/ICEI use case description and specification document for your use cases. I pre-filled a first science write-down and technical details as I understand it from the available documentation:
-SGA2 GA use case
-Information supplied during the L2L workshop
-SGA1 review slides

I also broke down the system into a simplified design pattern. This diagram should allow us to find system bottle necks. The document contains a detailed explanation how it can be used (page 1-7) and what is the role of the different sections. I included a PowerPoint with the diagrams if you want to adapt these to your needs.

The Word document has track changes enabled, this will allow streamlining a possible merge process later on.

Again, if you have any questions do not hesitate to ask! I am available per e-mail, phone or skype.

7.3 Example request for minimal requirements information

Dear <Name>,

To assure that your requirements can be taken into account during the ICEI procurement it is of great importance to participate in the minimal analysis as detailed below. With the final deadline nearing the detailed requirements analysis requested previously is out of scope.

The final deadline for input to be taken in account is the 29th of June.

Regarding the Collab to HPC connection we need at a minimum the technical details of the HPC/cloud compute resources, the long term storage needs and the bandwidth requirements. If you are not in the position to answer these questions, would it be possible to put Wouter Klijn (in cc) in contact with a compute expert in your team?

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: needed per node, in total
- IO: bandwidth, latency, always on/dedicated
- CPU: large size jobs / farming
- Specialized hardware: (GPU, KNL, FPGAs)
- Storage: size, access rate
- Specialized software: VM/containers
- Specialized features: in-situ visualization

Architecture Requirements:

- Minimal compute performance (excluding acceleration)
- Minimal volatile memory footprint of 192 GByte
- MPI point-to-point bandwidth of 10 GByte/s or higher
- MPI latency of 2 micro-seconds or less
- Access to active data repositories with a bandwidth of up to 8 GByte/s per node
- GPU requirements per node (minimum)

8. Appendix D: Use cases

The following sections contain the use cases as collected from the domain and computer experts.

9. Data-driven cellular models of brain regions, Olfactory Bulb (#1)

Data-driven cellular models of brain regions, Olfactory Bulb

Use Case Description and Specification

21-06-2018 Michele Migliore, Alexander Peyser, Wouter Klijn

Partners:	Michele Migliore Alexander Peyser
Institutions	Institute of Biophysics, National Research Council, Palermo, Italy Jülich Supercomputing Centre
Principal Investigators	Michele Migliore

Date	Version / Change
09-06-2018	(*Wouter Klijn*) Initial scientific and technical write-up
11-06-2018	(Wouter Klijn) Updated pre-processing diagram based on validation by MM
21-06-2018	(Wouter Klijn) Merge in changes by AP
28-08-2018	(Anne Carstensen) Editorial changes
01-09-2018	(Wouter Klijn) Fix problems with captions and missing chapter titles; Add questions to get at the next iteration of technical information
04-09-2018	(Michelle Migliore) Additional details expected bottle necks
18-09-2018	(Wouter Klijn) Merge in clarifications send by e-mail

9.1 Introduction

This use case is interesting because it combines aspect of two HBP Meta use cases: Multi-scale co-simulation and machine learning in an interactive loop, the spinnaker Neuromorphic hardware in this. After as short scientific introduction based on the Co-design workshop presentation from M. Migliore the workflow will be split into two diagram sections allowing the precise determination of the specific resource requirements.

9.1.1 Use Case Description

The scientific aim of this use case is the development of a brain prosthesis, using a morphologically and physiologically realistic computational model of a brain region

involved with sensorial inputs, in order to activate the cortical neurons of a live mammal bypassing the real system. The model, implemented in its natural 3D layout and directly driven by experimental data, will be interfaced with a living animal in an almost natural setting, to guide behavioural experiments.

9.1.2 Software

Starting from a full scale morphologically detailed model of the olfactory bulb, subsequent model reductions will be performed until finally a reduced model is created, running on neuromorphic hardware capable of real-time streaming communication to the implant in the behaving animal. Large scale morphologically detailed simulations will be performed in Arbor. Spiking network simulations will be performed in the NEST, and spinnaker is the target neuromorphic hardware. The HPC systems are in the current design not interacting with the live animal. Validation of NMC model necessitates on-line co-simulation with the HPC systems.

9.1.3 Estimations regarding needed compute resources

The current morphologically detailed simulations are performed in the Neuron simulator. Results are from the JUQUEEN supercomputer for a model at 1/20 of the real system.

Table 2 | Model parameters and execution times for a typical simulation.

	Seg (min-max)	States (min-max) (v, channels, and syn. gates)	Syn (min-max)
MC ($n = 635$)	380,748 (189–1433)	5,259,735 (2536–20,028)	707,216 (308–2799)
GC ($n = 69013$)	4,344,724 (33–257)	26,892,317 (261–869)	707,216 (1–62)
Total	4,725,472	32,152,052	

	Computation time	Comm. time (spike exchange)	Comm. time (multisplit)	Total run time (2048 procs)
Average (sec)	27,149.35	68.53	555.94	32,552.86
Max (sec)	27,756.25	813.44	1453.96	

Figure 2: Raw table with expected processing times.

Typical 40 sec of sim. on 2048 processors, fully integrated NEURON+python implementation, $750 \cdot 10^6$ spikes: 9 hours, 10 GByte output, 99% eff. 635 mitral cells 100K granule cells $7 \cdot 10^5$ synapses (1/20 of the real system area 32,000,000 nonlinear ODEs)

9.1.4 Model generation

Although not detailed in the current version of this document. A detailed processing chain is available for model generation, see Figure 3.

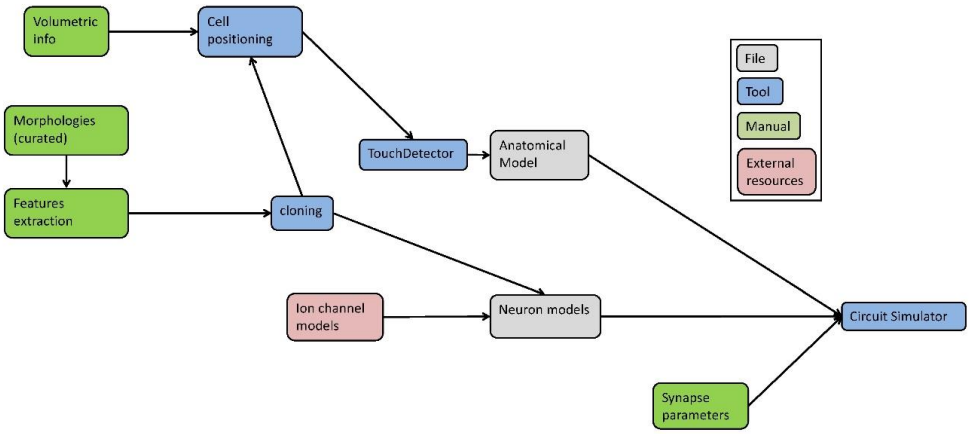


Figure 3: Detailed flowchart for generation of neuron models to be simulated in Arbor/neuron.

9.2 Diagrams

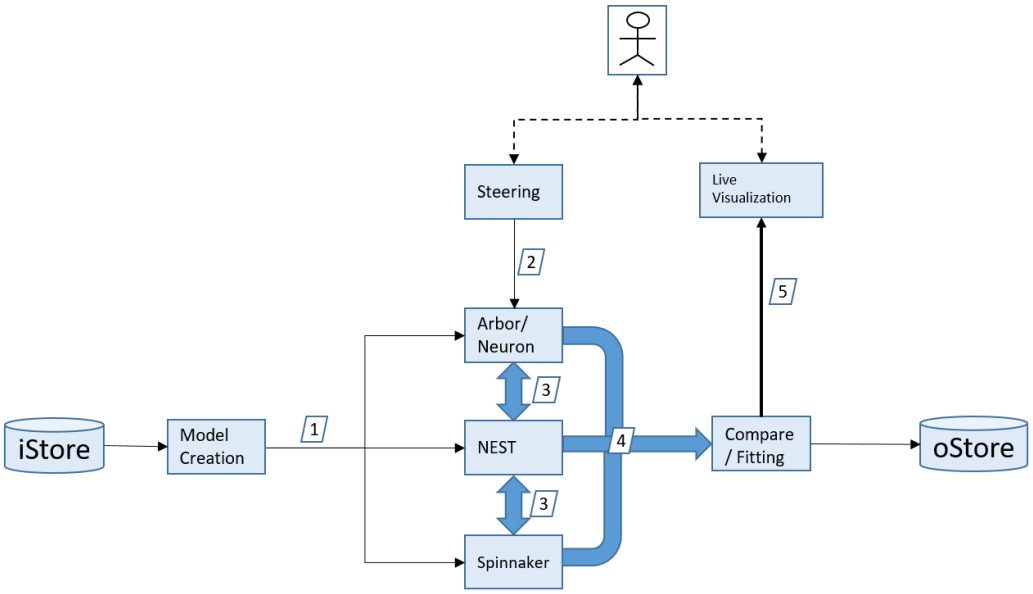


Figure 4: Multi-scale co-simulation diagram of the olfaction use case. Two simulators types, NEST and Arbor/Neuron should be able to run parallel with spinnaker hardware. Data produced in should be streamed to online analytics tools to be visualized life. This to allow the user to interact with the systems. The Model creation is further detailed in Figure 3. The major communication channels between the scales/systems are 3-7 with mostly neuro-physical data ranging from spikes to LFPs. 8 is the in-situ visualization stream this could be a screen cast or structure data to visualize on a local system. 2 Is the steering information for the simulators and 1 the model parameters.

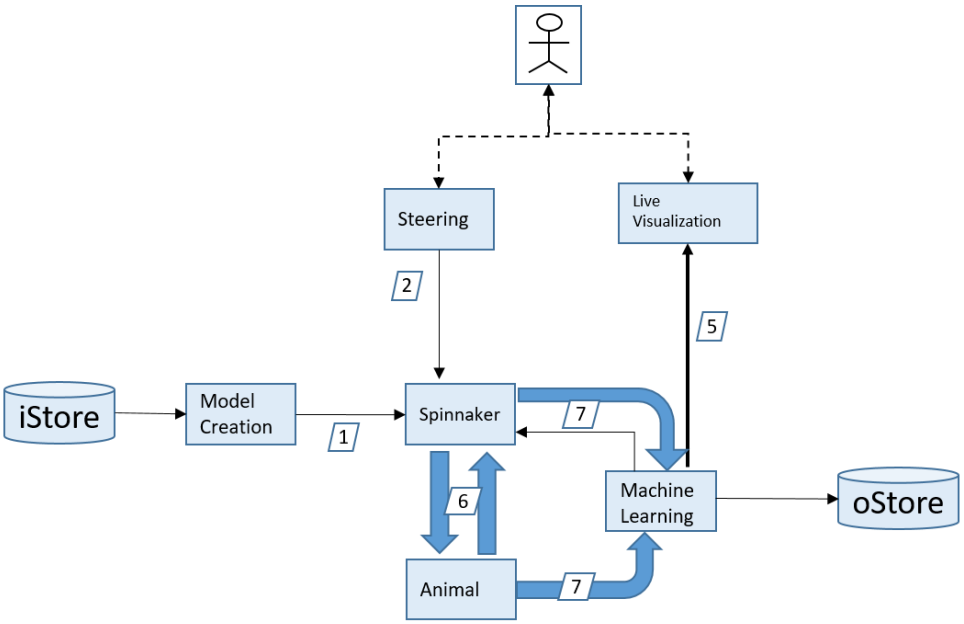


Figure 5: Machine learning in an interactive loop: Models retrieved from the store are loaded on the Spinnaker NMH system that is interacting via wireless to the brain sensor and current injector. The dynamics of the Spinnaker simulation should map on the measured dynamics in the animal. This matching will be performed with machine learning. The complexity of this system necessitates life steering and visualization. The Model creation is further detailed in Figure 3. Of particular interest is the two way connection 6. This should have an extreme low latency since it is a live animal loop.

9.3 Node Characterization

9.3.1 Data objects

Data object: ***1***, Network Models	
Base information	General description of what data is stored: • Formats: NMODL, NESTML, Python scripts, PyNN • Metadata: Version number • Database requirements: None specific
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. • Short-term: cached for use by multiple simulators/tools The database might be the brain atlas
Current solution	Name: Local file directory
	URL to additional information: NA
	Limitations: Ad hoc, un-optimized and not linked to RBA/NIP infrastructure

Data object: ***2***, Commands from front to backend	
Base information	General description of what data is stored: • Formats: XML / JSON • Metadata: None • Database requirements: None
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded.
Current solution	Name: None, only partially implemented in NEST with 'nett'
	URL to additional information: https://doi.org/10.3389/fninf.2018.00032
	Limitations: Not implemented

Data object: ***3***, Between simulator data: Spikes	
Base information	General description of what data is stored: • Formats: GID (id, Time), MUSIC • Metadata: None • Database requirements: None
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded.
Current solution	Name: Music
	URL to additional information: https://github.com/INCF/MUSIC
	Limitations: Currently takes ownership of the MPI world and application execution. MUSIC2 should resolve this issue

Data object: ***4***, Spikes / simulation data to analysis	
Base information	General description of what data is stored: • Formats: GID (id, time), MUSIC

	• Metadata: None • Database requirements: None
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded.
Current solution	Name: MUSIC
	URL to additional information: https://github.com/INCF/MUSIC
	Limitations: See data object 3

Data object: ***5***, Analysis result visualization	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded.
Current solution	Name: Not implemented yet. See RTNeuron inputs for examples.
	URL to additional information: NA
	Limitations: NA

Data object: ***6***, Life animal / Spinnaker data transport	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: Not defined yet
	URL to additional information: NA
	Limitations: NA

Data object: ***7***, Life animal / Spinnaker to analytics / machine learning platform	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

9.3.2 Data transport

Data transport: ***1,*** Network models	
Base information	General description of what data is transported: Model information needed to instantiate models in the simulators. Data is stored in the local HPC centre.
	Data access patterns (request rate, transfer sizes): If raw network input: NEST / Spinnaker: N_neurons * N_synapses_per_neuron Arbor: N_neurons * (N_synapses + N_Morphology + N_processes) If generative: Trivial SpiNNaker will require a generative model for realistic loading times
Technical specifications	Maximum required bandwidth:100 GBit/s The loading of the network is part of the HPC runtime and should be minimum. Model generation is negligible (22 Sec.) compared to simulation time with current HPC solutions (Infiniband).
	Average required bandwidth: Load time network / total simulation duration < 10%
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: Panoply of scripts, databases, Blueron infrastructure and others
	URL to additional information: NA
	Limitation: Current solutions are not limiting

Data transport: ***2***, Commands from front to backend	
Base information	General description of what data is transported: Xml or JSON command
	Data access patterns (request rate, transfer sizes): Thousands of commands per second of minimal size eg: 1000 * 1 k
Technical specifications	Maximum required bandwidth: (in Mb/Sec)
	Average required bandwidth: 100K/Sec
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NETT / ZeroMQ
	URL to additional information: NA
	Limitation: Not meant how big data transport

Data transport: ***3***, Between simulator data: Spikes	
Base information	General description of what data is transported: Simulation data, typically spikes. The full spike output of the network should be transported.
	Data access patterns (request rate, transfer sizes):

	Rate is 1000 Hz per simulated second? Size is depending on the size of the network and the amount of neurons sending between simulators $750 \cdot 10^{6}$ Spikes Spikes for the 1/20 size network. Runtime 9 hours. The network communication even at full scale is << below the standard HPC bandwidth.
Technical specifications	Maximum required bandwidth: This needs to be calculated based on neurons and might be a bottle neck .
	Average required bandwidth: <<100 GBit/s range (depends on number of neurons)
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: Music / MPI
	URL to additional information: NA
	Limitation: NA

Data transport: ***4***, Spikes / simulation data to analysis	
Base information	General description of what data is transported: Simulation data, typically spike or membrane voltages Subset of the total population of neurons
	Data access patterns (request rate, transfer sizes): Rate is 1000 Hz per simulated second Size is depending on the size of the network and the amount of neurons sending between simulators (20 Hz * N_neurons)* 16 Byte/Sec
Technical specifications	Maximum required bandwidth: <<100 GBit/s
	Average required bandwidth: <<100 GBit/s
	Interface requirements for attached entities: NA
	Additional information: Number of neurons simulated on SpiNNaker will be the limiting factor
Current solution	Name: MUSIC / NESTIO / output to disk
	URL to additional information: NA
	Limitation: NA

Data transport: ***5***, Analysis result visualization	
Base information	General description of what data is transported: A) Screen cast of visualization application B) Structured data to be visualized on a front end

	Data access patterns (request rate, transfer sizes): A) and B) Continuous streams A) NA B) NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: ***6***, Life animal / Spinnaker data transport	
Base information	General description of what data is transported: Electrode voltage values: samples/s * number of electrodes * 64 B
	Data access patterns (request rate, transfer sizes): Continuous
Technical specifications	Maximum required bandwidth: @ 20Hz, approx.. 10 MB/s
	Average required bandwidth: Same – continuous
	Interface requirements for attached entities: electrodes to local A/D converter
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: ***7***, Life animal / Spinnaker to analytics / machine learning platform	
Base information	General description of what data is transported: To the life animal: 30 spike channel (or analog signal?) From: 1024 channel 1000 Hz analog signal
	Data access patterns (request rate, transfer sizes): Rates are slow and they are specialty connections, see 6 as well.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: Delay should be extreme low (ms range?)
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

9.3.3 Data ingest / GUI

Data ingest: Live Visualization	
Base information	Description of input data source / drain: The output size of the models is in the direction of millions of neurons. Not all can be visualized or stored. Transient data should be displayed in-situ.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): Either structured data to be visualized on the frontend –or- Screen casting of HPC generated imagery
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Not more than 5 concurrent users under most conditions; Maybe be up to 30-50 in case of dissemination activities such as School/workshop/lecture Format: xml / json messages Loads: The generation of the images when on HPC resources can be large Bandwidth: 4 Mbit/s per user Latency: Below 30 ms/s two way 30 ms/s allows for 40 ms processing of commands for usability of GUI
	Additional information: GPU's on the HPC visualization cluster
Current solution	Name: HBP in-situ pipeline
	URL to additional information: Aachen uni in combination with HEP
	Limitation: Only proof of concept

Data ingest: Steering	
Base information	Description of input data source / drain: The parameter space prevent grid based parameter space exploration. There will be the need for interactive changing of parameters in the models and experiment
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Spinnaker The requirements of this node are not applicable to ICEI: The 2 ms/s latency between neuromorphic hardware and the life animal places this node in the diagram outside of ICEI. This should be addressed with colocation of the hardware or even a purpose build experimental system.	
Base information	Description of input data source / drain: From a systems design standpoint, it is best to see the NMH as a data source or drain, mostly outside of direct control. When in co-simulation with NEST or Arbor/Neuron it should have a high bandwidth connection. When in live animal interaction the bandwidth requirements are lower.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): Input should be spike and model.
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Format: PyNN and spikes Loads: NA Bandwidth: 10 Gb/s or 1 Mb/s Latency: Below 2 ms/s for life animal loop When in co-sim: it should be low the 1 ms divided by the xrealtime we are with the simulation.
	Additional information: This specific setup is not feasible within ICEI. Neuromorphic hardware collocated with HPC resources is not budgeted for.
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

9.3.4 Data repository

Data repository: iStore	
Base information	Classification of the data objects (see below): Model based information or raw point to point connections Local buffering of models and atlases: - Neuroscience models - The HBP rodent brain atlas? NIP? Seattle Allen Institute data? Both statistics but detailed individualized network graphs are to be expected.
	Access control requirements: NA
	Access requirements
	Data availability requirements

Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: Bandwidth should be normal HPC bandwidth because the loading time will be taken form the runtime on the HPC resources. Current solution has a (22 Sec.) build time. The data source should be able to produce the model at least in that time.
Current solution	Name: NA
	URL to additional information: NA
	Limitation: Current solution is not limiting science production

Data repository: OStore	
Base information	Classification of the data objects (see below): Generated brain models Statistical aggregate activity Detail full snapshot information for selected neurons Additional Meta information: stored in a separate query-able database Currently estimated at 1-20 GB/Simulations Less than 1000 simulations per year Older simulations will become obsolete. Reprocessing is rarely expected. 1/2 times per year Data can be stored on low availability medium (tape)
	Access control requirements
	Access requirements
	Data availability requirements Diverse formats Loads should be low enough that simulation is not slowed down Latency can be low
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

9.3.5 Processing stations

Processing station: Arbor/Neuron	
Base information	General description of data processing: Full scale simulation of the olfactory bulb
	Typical processing steps: Load of network model Co-Simulate model with NEST
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern After initial load minimal Data production access pattern: Continues spike output for selected neurons
	Additional information: Arbor: supports GPU, KNL, ARM and multi-core/vector Neurons: support GPU and multi-core Estimates: 1/20 systems size 40 second simulated time: 2048 processors NEURON+python 750*10e6 spikes 9 hours wallclock 10 GB output
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: NEST	
Base information	General description of data processing: Full scale simulation of the olfactory bulb
	Typical processing steps: Load of network model Co-Simulate model with Arbor/Neuron

	Number of processing steps: 1
Technical specifications	Data processing hardware architecture requirements: Multi-core (NO GPU or KNL)
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: After initial load minimal Data production access pattern: Continues spike output for selected neurons Processing requirements compared to Arbor/neuron simulation minimal: 10 nodes should be fine (240 Cores) We need estimates for memory requirements, but again lower then Arbor/Neuron
	Additional information: NA
Current solution	Name: NA
	URL to additional information : NA
	Limitation: NA

Processing station: Model creation	
Base information	General description of data processing: Combination of manual and automatic pre-processing steps that generate an instantiated network models with individualized neurons from e.g. HBP Atlas resources Detailed diagram can be found in Figure 3 The creation is non interactive. With current resources takes this step 30 seconds. Model generation is expected to scale linearly with the size of the network. Currently is performed on the master proc. (Neuron limitation?)
	Typical processing steps: Manual feature, and parameter selection Specialized tools for generating neuron morphologies, placement in 3d space and connection generation
	Number of processing steps: 10
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA

	Ratio of data processing rate versus data consumption and production rate: Per simulation
	Variability, availability, bandwidth and latency: "blue steps are carried out at setup and do not use/require unusual resources."
	Additional information: Could be a target for containerization.
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

9.4 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? 1hr • What software stacks need to be available? Visualization software • Is it possible to define memory capacity requirements? No
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? No
Virtual Machine Services	• Which parts of the workflow require such services? No
Active Data Repositories	• Which parts of the workflow require such services? Simulation output, visualization and offline analyses
Archival Data Repositories	• Which parts of the workflow require such services? Offile analyses

Data Mover Services	• Which parts of the workflow require such services? Maybe
Data Transfer Services	• Which parts of the workflow require such services? NA • Between which ICEI sites is data planned to be transferred? JSC, Cineca and CSCS • How much data is expected to be transferred per time unit? Order of GBytes • How are transfer patterns expected to change over time? Probably increase by 10x
Data Location Service	• Which parts of the workflow require such services? None
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site? No
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? Visualization should be available from the web
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples: ○ Special accounts for running services ○ Needs for fine-granular control of access to data No
User Support Services	• Are the specific foreseeable needs for user support services? Minimum (e.g. Installation of visualization software, or python libraries)

9.5 Use Case references

The information in this use case is collected from a wide range of different information sources. An import information source has been the diverse co-design workshop opportunities in the last years. Access to a repository with resources can is available from the SimLab / Wouter Klijn.

Co- design workshop 9-Feb-2018 presentation: “Data-driven cellular models of brain regions: the Hippocampus and the Olfactory Bulb use cases” Michele Migliore

Migliore M, et al., Proc Natl Acad Sci U S A. 2015 Jul 7;112(27):8499-504
Migliore M, et al., Front Comput Neurosci. 2014 Apr 29;8:50

10. Large scale simulations of models: Hippocampus (#6)

Large scale simulations of models: Hippocampus

Use Case Description and Specification

26-06-2018 Michele Migliore, Wouter Klijn

Partners	jean-denis.courcol@epfl.ch
Institutions	
Principal Investigators	Michele Migliore

Date	Version / Change
15-06-2018	(Wouter Klijn) Collection of initial information
26-06-2018	(Wouter Klijn) Merge in updates and clarification e-mail
28-08-2018	(Anne Carstensen) Editorial changes
10-09-2018	(Anne Carstensen) Added new Figure 7 provided by Michele Migliore

10.1 Use Case Description

10.1.1 SGA2-SP6-UC003 - Community user can do in silico experimentation with HBP brain region models through the Collaboratory

In silico experimentation with HBP brain/brain region models is a core target of HBP. It allows linking results from experimental neuroscience with model predictions for discovery and validation. The “scaffold” models that will be made available are derived from those that are constructed and validated in SP6 and CDP2 to investigate microcircuit dynamics and plasticity across scales. They are based on a close bidirectional interaction with anatomical and physiological data produced in SP1. The models focus on the cerebellum, the hippocampus, and the basal ganglia.

This Use Case describes the execution of an in silico experiment of a biophysically detailed model and the execution of a pre-defined analysis by a community user against models released to the community. It uses data and provides feedback from/to SP2, SP3 and SP4. This Use Case will be applicable to the priority brain region models developed by SP6, or community-contributed models. Users can now devise in silico experiments that they could not do before in the absence of the required storage and compute resources for downloading and executing potentially large models. At the same time, the resulting artefacts remain within the HBP platform ecosystem and become easily available for reuse in other contexts (analysing, visualisation, sharing with the community etc.). The work builds on tasks from the RUP and from SGA1 and SGA2.

10.1.2 CDP2 KRc2.3 Hippocampus – Demonstrating multi-scale plasticity

This Key Result will integrate several tasks and components from different SPs, with the main aim to reach the main goals of HBP, and in particular FO4 (Build multi-scale scaffold theory and models for the brain) and FO3 (Simulate the brain). The focus here will be on models of synaptic plasticity of hippocampal synapses, and how they can be integrated into cellular level microcircuit models using data-driven subcellular pathways and/or rule-based effective implementation. The effect at the microcircuit level will be investigated in terms of network self-organisation during synaptic inputs activated under different conditions of timing and spatial activation. The emphasis will be on the mechanisms underlying associative memory processes and spatial navigation, integrated into a user-friendly user interface allowing an easy community engagement to the Brain Simulation Platform and its functionalities.

10.1.3 Hippocampus data from ICEI co-design workshop

Why an interesting simulation target:

- A few millions neurons
- Strongly involved in higher brain functions (learning, memory, spatial navigation)
- Implicated in Alzheimer's disease, temporal lobe epilepsy, cognitive aging, post-traumatic stress disorder, transient global amnesia, schizophrenia, depressive and anxiety disorders.

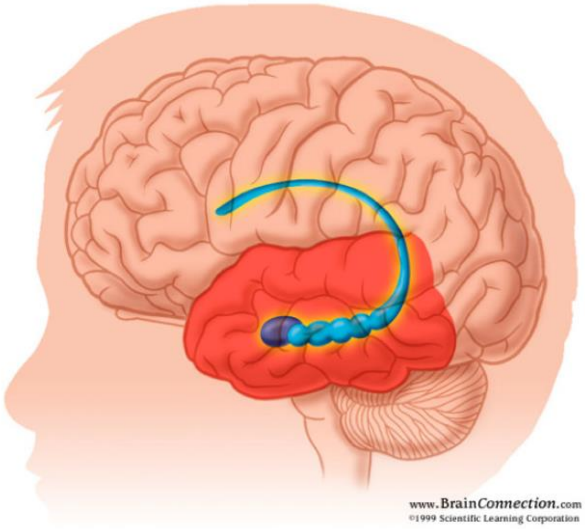


Figure 6: Location of the hippocampus in the brain.

Raw Numbers:

- 450000 neurons, ~$1 \cdot 10^8$ memb seg, 20 ODE/seg
- $2 \cdot 10^9$ ODEs + synapses
- 1 second of sim time: 5 hr on BG/Q using 32000 procs
- ~2 TByte of input, up to ~3 TByte of output

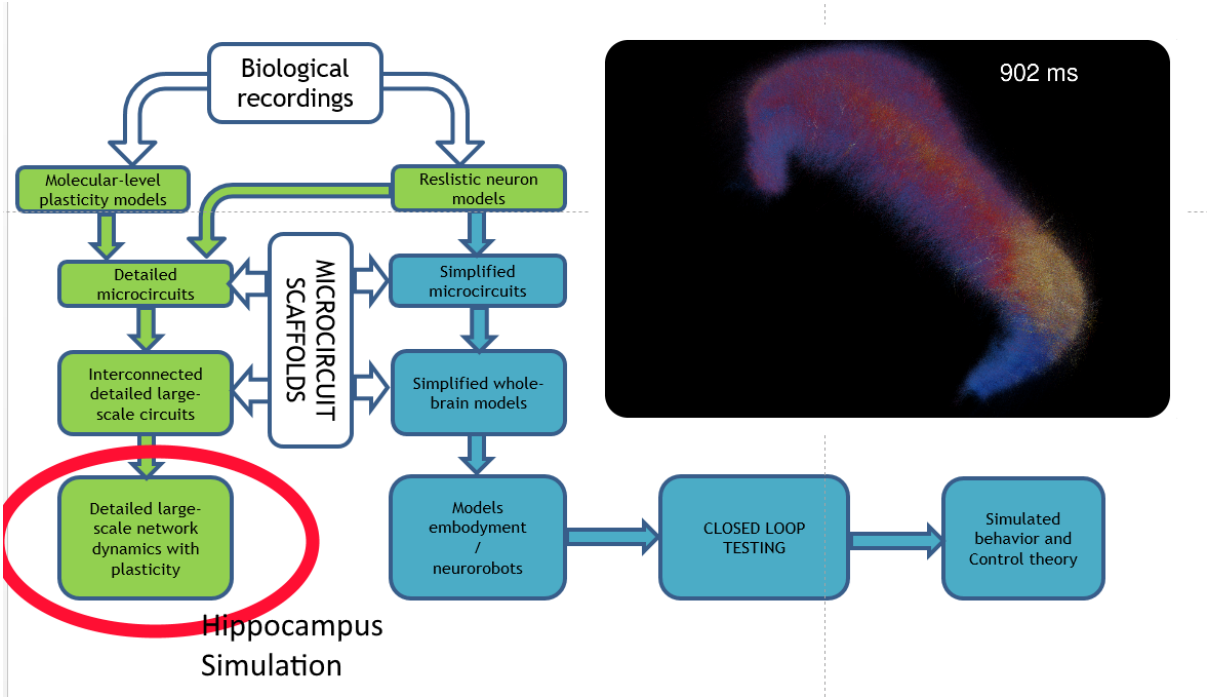


Figure 7: Image adapted from CDP2 review slides.

10.1.4 Basic Workflow

1. Peter selects the Small Circuit or Brain Area Circuit in silico Experiment function of the Brain Simulation Platform.
2. Peter selects a detailed circuit model from NIP.
3. Peter has a new HPC project on a HPC centre. The project is empty.
4. Peter selects target regions he wants to stimulate.
5. For each selected target region, he defines the stimulus he wants to apply.
6. Peter defines the particular parameters of each stimulus (e.g. start, duration).
7. Peter selects what he wants to record from the circuit (e.g. soma voltage of a particular subset of neurons).
8. Peter defines global parameters for the simulation (e.g. time steps).
9. Peter defines additional parameters related to requesting compute resources for the simulation (HPC centre and system, HPC project, number of nodes, memory, ...).
10. Peter defines the analysis he wants to perform from a predefined set and configures this analysis.
11. Peter defines additional parameters related to the allocation of the compute resources for the analysis (HPC centre system, HPC project, number of nodes, memory, ...).
12. The simulation and the analysis are executed on the different compute resources defined by and accessible to the user. The circuit is available on this compute centre at this stage.
13. Peter investigates the simulation result and the circuit interactively within a jupyter notebook.

14. Peter wants to visualize the simulation on the visualization web service (currently only possible using VMs at ETHZ/CSCS).
15. Peter decides to register and store his simulation in the Knowledge graph.
16. The HPC project allocation ends and HPC storage get erased.

10.2 Diagrams

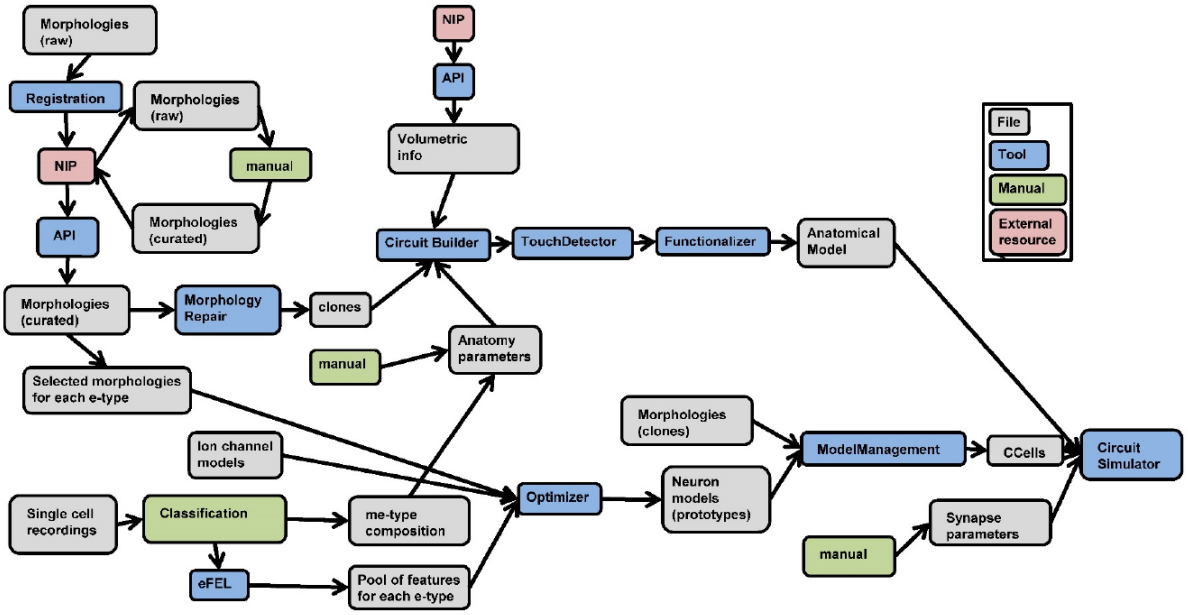


Figure 8: Pre-processing and model generation pipeline as described in Hippocampus model generation (ICEI co-design workshop).

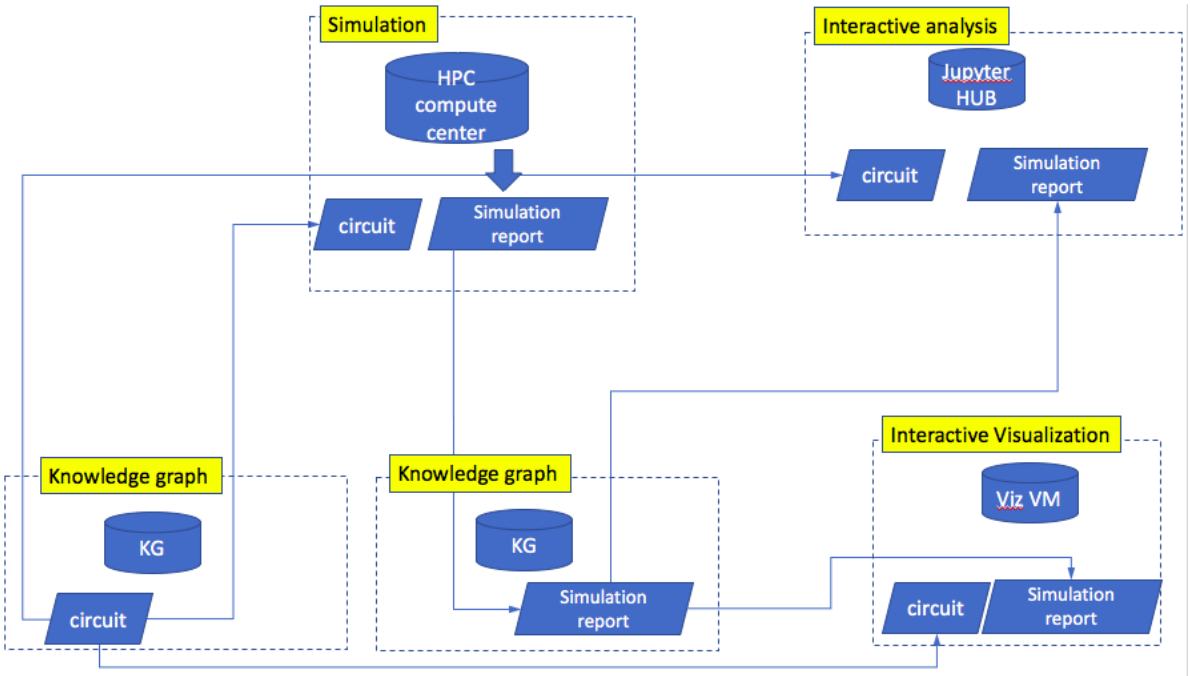


Figure 9: Major components in "SIM started from Brain Simulation Platform". As based on the 'basic workflow' in SGA2-SP6-UC003.

10.3 Node Characterization

The output of the experiments is following the current use case description:
Ingestion into long term storage without anyone looking at the data.
This use case will include in-situ visualization at a later stage.

10.3.1 Data objects

Data object: Circuit	
Base information	General description of what data is stored: A model of a brain region neuron network • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): up to 2 TB for use/reference/analysis/access; Additional information: 1 Circuit is ~1000 s of files with a total size of >200 GByte. This is increasing as we are building bigger brain region. We usually have several releases (~4) of the circuit per year plus a ~20 iteration (temporary version) of the circuit per year
Current solution	Name: Circuit are stored in CSCS storage. Either in private container or public container. Link to KG is in progress.
	URL to additional information: NA
	Limitations: No proper solution to register and store circuit in KG Circuit are **manually** duplicated to HPC compute centre. (error prone/no automation) Circuit are not managed in jupyter notebook. Current solution is limited (no ACL, hardcoded slow) No proper solution for copy into CSCS visualization VM

Data object: Simulation report	
Base information	General description of what data is stored: A report on the spiking activity and on various variable for each of the compartment of the simulated neurons for each time step of the simulation • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): Yes, depending on visualization/analysis • Short-term (Campaign): up to 5 TB/job • Permanent (Forever): up to 5 TB/job, older files may be deleted if they become obsolete for the community

	Additional information: The mean size of a simulation is several GByte, however, this can increase depending of the experiment performed.
Current solution	Name: Output not accessible from jupyter notebook (analysis KO) No proper solution for copy into CSCS visualization VM No proper solution for KG registration and storage
	URL to additional information: NA
	Limitations: NA

10.3.2 Data transport

Data transport: circuit object - Knowledge graph to HPC centre transport	
Base information	General description of what data is transported: Move the circuit object from the knowledge graph storage to the HPC centre storage.
	Data access patterns (request rate, transfer sizes): 1 time per circuit usually
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities
	Additional information NA
Current solution	Name: This does not exist. It is done through manual scp from a copy located at BBP.
	URL to additional information: NA
	Limitation: NA

Data transport: circuit object – Knowledge graph to Jupyter notebook transport	
Base information	General description of what data is transported: Make the circuit available from the jupyter notebook kernel
	Data access patterns (request rate, transfer sizes): This is happening after each simulation ~ 100+/year
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

Data transport: circuit object – Knowledge graph to CSCS visualization VM transport	
Base information	General description of what data is transported: Make the circuit available on the CSCS VM

	Data access patterns (request rate, transfer sizes): 1 time per circuit if the circuit can be stored in the VM for a long time.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – HPC centre to Knowledge graph transport	
Base information	General description of what data is transported: Copy the simulation output to the knowledge graph storage.
	Data access patterns (request rate, transfer sizes): several GB. 100+ per year (for GB ones). Some will be TB.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – Knowledge graph to CSCS visualization VM transport	
Base information	General description of what data is transported: Copy the simulation object from the knowledge graph storage to the CSCS VM performing interactive visualization.
	Data access patterns (request rate, transfer sizes): 100+ year.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: N
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – Knowledge graph to jupyter notebook transport	
Base information	General description of what data is transported: Move the simulation object from the knowledge graph to the jupyter hub kernel storage
	Data access patterns (request rate, transfer sizes): 100+ / year

Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

10.3.3 Data ingest/ GUI

NA at this stage

10.3.4 Data repository

Data repository: Knowledge graph data storage (a.k.a POLLUX storage)	
Base information	Classification of the data objects (see below): Object storage maintained by SP5
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: HPC compute centre storage	
Base information	Classification of the data objects (see below): Storage allocation provided for a HPC project.
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current	Name: NA

solution	URL to additional information: NA
	Limitation: NA

10.3.5 Processing stations

Overall resources requirements for the entire workflow:
Core/hours per year: +60 M
TB/Year output data: approximately 200 TB/year

Processing station: Parameter collection / control script	
Base information	General description of data processing: single cell optimization and circuit building
	Typical processing steps: single cell optimization
	Number of processing steps: hundreds
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.)entire software stack deployed by SP6
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Pre-processing (detailed in Figure 8)	
Base information	General description of data processing: Most time consuming pre-processing is currently done elsewhere.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

10.4 Use Case references

"Data-driven cellular models of brain regions: the Hippocampus and the Olfactory Bulb use cases" ICEI Co design workshop Migliore

"CDP2 Mouse-Based Cellular Cortical and Sub-Cortical Microcircuit Models", Egidio D'Angelo/Michele Migliore

HBP SGA2 GA

10.5 Discussion

1. Regarding the "Data object, Simulation report"
a. You state 5 TByte/job. How many jobs do you expect to run per year?
100-200

"Data transport: circuit object – Knowledge graph to Jupyter notebook transport"
Gives the number 100+/year. Could I put in an upper bound of 200 jobs per year?
Yes.

b. How many individual files does a report have (or is it a single large HDF5 file?)
A single large file and a few smaller files.

2. Core/hours per year +60 M
a. Is this for the full size of the hippocampus at high detail?
Yes
In an email received on 3.10.2018 it has been clarified that the resource estimate referred to Piz Daint.

b. Do you need GPUs or other specialized hardware for these simulations?
Not for the simulation. Visualization can use GPSs

c. How many cores do you need at the same time? 1000?
The full system requires at least 15000 cores.
In an email received on 3.10.2018 it has been clarified that this refers to Piz Daint multi-cores, i.e. nodes with 36 cores and 128 GiByte of memory.

d. How much memory do you need to perform to full scale simulation. And is this total memory or per node?
So far on JUQUEEN and JURECA we have used the entire memory available in each node.

e. Are these large scale simulation running in VMs or are they normal
HPC jobs?
Normal HPC jobs

11. Large scale simulations of models: Cerebellum (#5)

Large scale simulations of models: Cerebellum

Use Case Description and Specification

26-06-2018 Egidio D'Angelo, Wouter Klijn

Partners	claudia.casellato@unipv.it elisa.marenzi@unipv.it simona.tritto@unipv.it
Institutions	
Principal Investigators	Egidio D'Angelo (egidiougo.dangelo@unipv.it)

Date	Version / Change
15-06-2018	(Wouter Klijn) Collection of initial information
26-06-2018	(Egidio D'Angelo) Validation
28-08-2018	(Anne Carstensen) Editorial changes
07-09-2018	(Wouter Klijn) Recreate template and add questions to get at the next iteration of technical information
01-10-2018	(Wouter Klijn) Integrate answers from Claudia Casellato, clean up of document

11.1 Use Case Description

11.1.1 SGA2-SP6-UC003 - Community user can do in silico experimentation with HBP brain region models through the Collaboratory

In silico experimentation with HBP brain/brain region models is a core target of HBP. It allows linking results from experimental neuroscience with model predictions for discovery and validation. The "scaffold" models that will be made available are derived from those that are constructed and validated in SP6 and CDP2 to investigate microcircuit dynamics and plasticity across scales. They are based on a close bidirectional interaction with anatomical and physiological data produced in SP1. The models focus on the cerebellum, the hippocampus, and the basal ganglia.

This Use Case describes the execution of an in silico experiment of a biophysically detailed model and the execution of a pre-defined analysis by a community user against models released to the community. It uses data and provides feedback from/to SP2, SP3 and SP4. This Use Case will be applicable to the priority brain region models developed by SP6, or community-contributed models. Users can now devise in silico experiments that they could not do before in the absence of the required storage and compute resources for downloading and executing potentially large models. At the same time, the resulting artefacts remain within the HBP platform ecosystem and become easily

available for reuse in other contexts (analysing, visualisation, sharing with the community etc.). The work builds on tasks from the RUP and from SGA1 and SGA2.

11.1.2 CDP2 KRc2.2 Cerebellum – Demonstrating sensorimotor loop using cerebellar model in a neurorobotics setting with learning

This SGA2 Key Result aims to refine and apply molecular/cellular level models of cerebellum to (1) simulate dynamic control of plasticity in trial-and-error learning, (2) integrate the cerebellum with extra-cerebellar circuits for large-scale network simulations, (3) simplify such models and integrate them into whole-brain robotic simulators, and (4) extend cerebellar modelling through the Collaboratory. (5) All this activity will be coordinated with the development and refinement of neuroinformatics tools. The cerebellum models will also be exploited for simulations of pathological alterations of plasticity and circuit dynamics in SP8. Therefore, there will be a multiple fallout at the level of brain modelling, theoretical understanding of brain function and disease and infrastructure implementation.

11.1.3 CDP2 KRc2.3 Hippocampus – Demonstrating multi-scale plasticity

This Key Result will integrate several tasks and components from different SPs, with the main aim to reach the main goals of HBP, and in particular FO4 (Build multi-scale scaffold theory and models for the brain) and FO3 (Simulate the brain). The focus here will be on models of synaptic plasticity of hippocampal synapses, and how they can be integrated into cellular level microcircuit models using data-driven subcellular pathways and/or rule-based effective implementation. The effect at the microcircuit level will be investigated in terms of network self-organisation during synaptic inputs activated under different conditions of timing and spatial activation. The emphasis will be on the mechanisms underlying associative memory processes and spatial navigation, integrated into a user-friendly user interface allowing an easy community engagement to the Brain Simulation Platform and its functionalities.

11.1.4 Hippocampus data from ICEI co-design workshop

Why an interesting simulation target:

- A few millions neurons
- Strongly involved in higher brain functions (learning, memory, spatial navigation)
- Implicated in Alzheimer's disease, temporal lobe epilepsy, cognitive aging, post-traumatic stress disorder, transient global amnesia, schizophrenia, depressive and anxiety disorders.

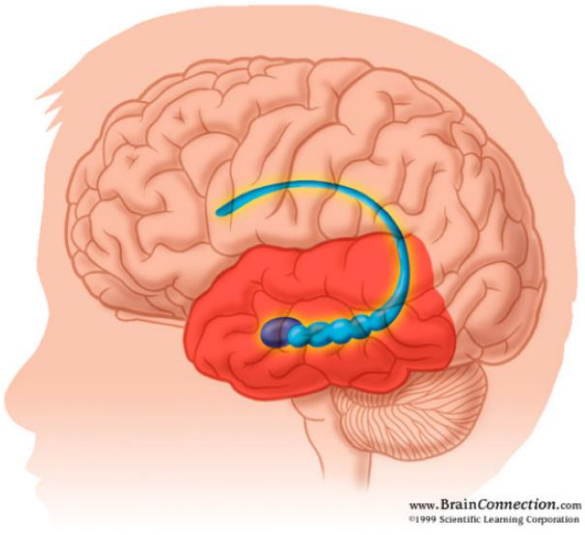


Figure 10: Location of the hippocampus in the brain.

Raw Numbers:

- 700000 neurons, ~350·106 memb seg, 20 ODE/seg
- 7·109 ODEs + synapses
- 1 second of sim time: 5 hr on BG/Q using 32000 procs
- ~8 TByte of input, up to ~3 TByte of output

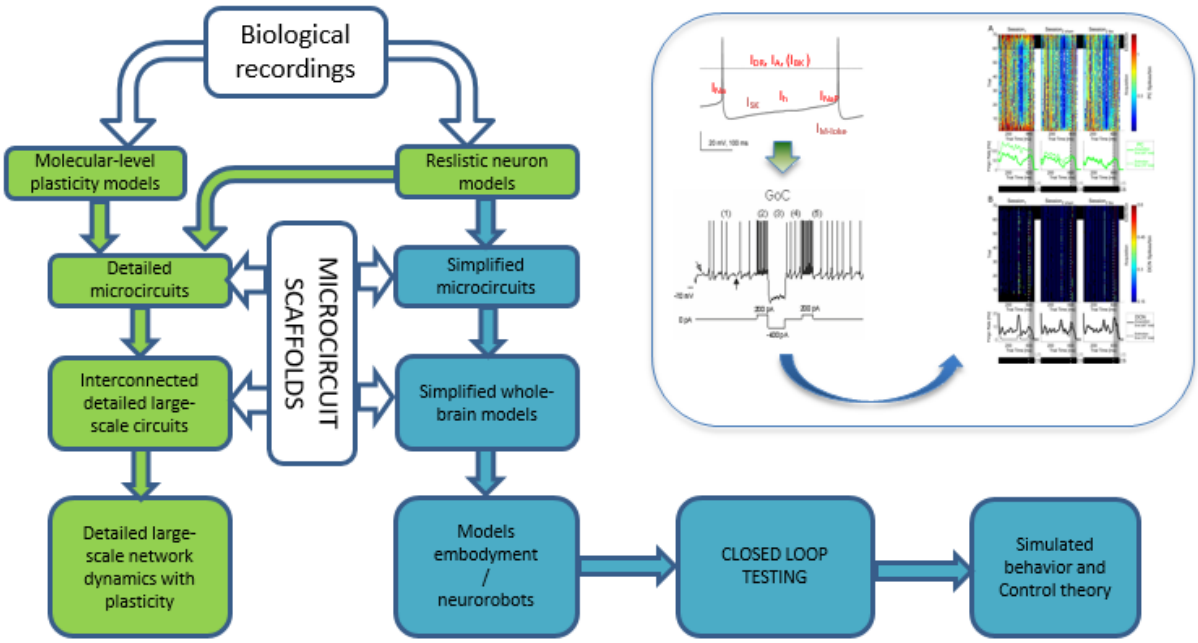


Figure 11: Image taken from CDP2 review slides.

11.1.5 Basic Workflow

1. Peter selects the Small Circuit or Brain Area Circuit in silico Experiment function of the Brain Simulation Platform.
2. Peter selects a detailed circuit model from NIP (pyNEURON).
3. Peter selects target regions he wants to stimulate.
4. For each selected target region, he defines the stimulus he wants to apply.
5. Peter defines the particular parameters of each stimulus (e.g. start, duration).
6. Peter selects what he wants to record from the circuit (e.g. soma voltage of a particular subset of neurons).
7. Peter defines global parameters for the simulation (e.g. time steps).
8. Peter defines additional parameters related to the allocation of the compute resources for the simulation (HPC centre and system, HPC project, number of nodes, memory, ...).

9. Peter defines the analysis he wants to perform from a predefined set and configures this analysis.
10. Peter defines additional parameters related to requesting compute resources for the analysis (HPC centre system, HPC project, number of nodes, memory, …).
11. The simulation and the analysis are executed on the different compute resources defined by and accessible to the user.
12. Peter investigates the simulation result and the circuit interactively within a jupyter notebook through the Collaboratory (exploiting also HBP visualization tools).

Remark (Wouter Klijn): This use case shares functionality with use case #6 “Large scale simulation of models hippocampus”. The workflow in #6 contains additional steps, due to the clarification added in a later stage.

11.2 Diagrams

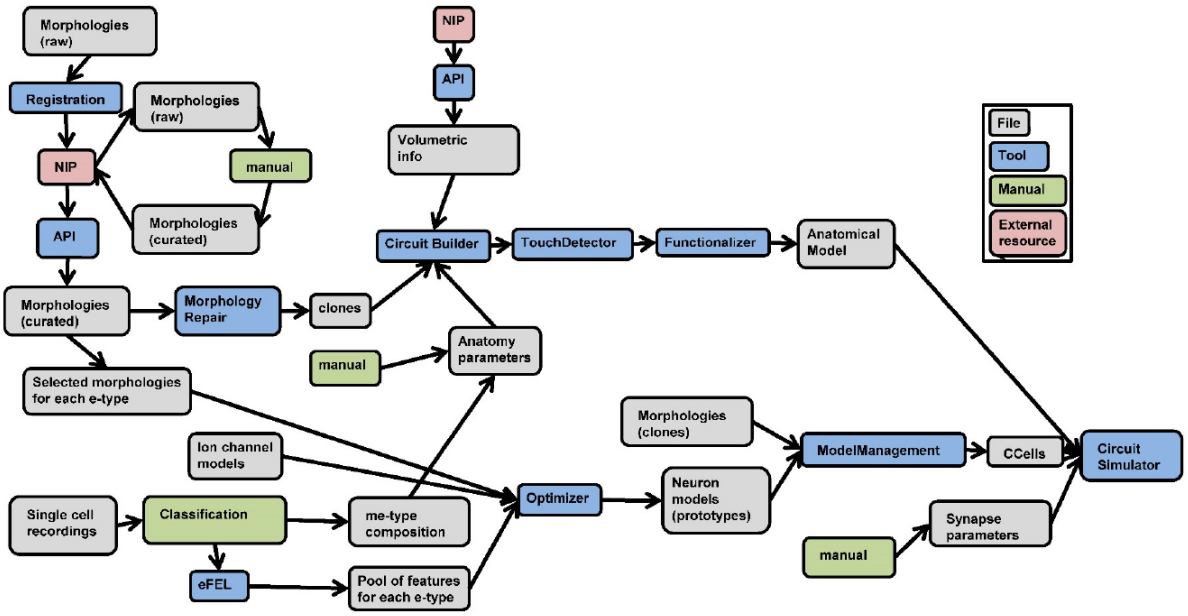


Figure 12: Pre-processing and model generation pipeline as described in Hippocampus model generation (ICEI co-design workshop).

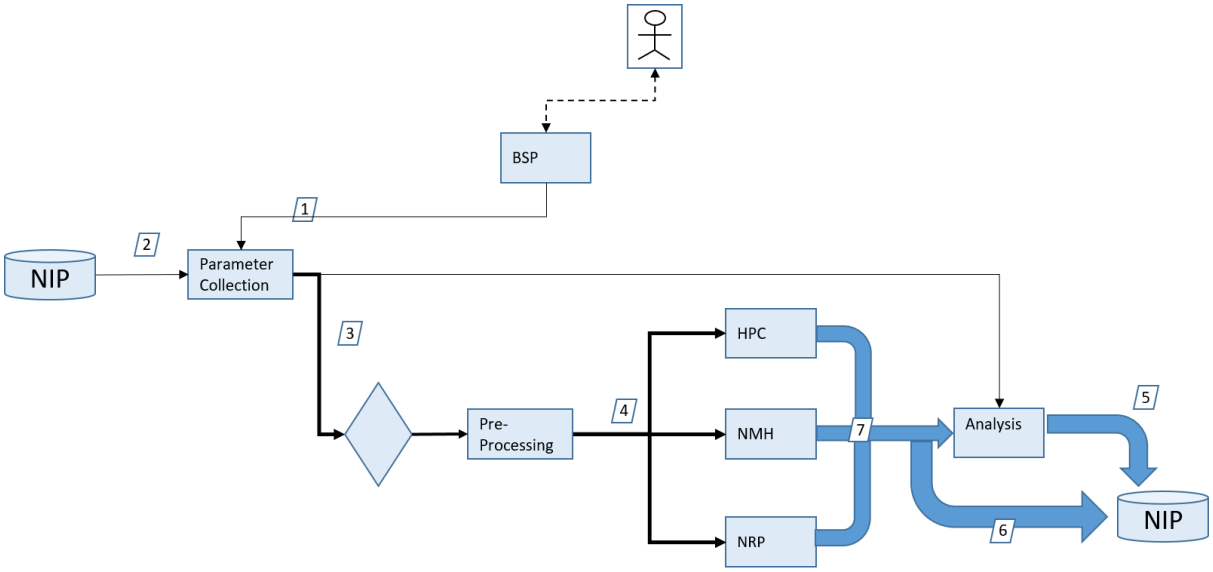


Figure 13: Major components in "SIM started from Brain Simulation Platform". As based on the 'basic workflow' in SGA2-SP6-UC003.

11.3 Node Characterization

Remark (Wouter Klijn): The output of the experiments is send to long term storage without anyone looking at the data. Is this correct?
This use case appears ideal for in-situ visualization.

11.3.1 Data objects

Data object: Circuit building	
Base information	General description of what data is stored: A model of a brain region neuron network
Technical specifications	• Permanent (Forever): up to 2TB for use/ reference/ analysis/ access Additional information: 1 Circuit of ~98000 neurons is built [python] using a number of hdf5 files (depending on nodes) of about 120 Mb. Each detailed neuron type (7 types) is plugged in by using 15 files of about 150 kb in total. We usually have several releases of the circuit per year.
Current solution	Name: Circuit information is generated and stored in HBP Local Collaboratory storage. Detailed single neuron models are stored in KG/NIP (after optimization)
	URL to additional information: NA
	Limitations: NA

Data object: Simulation and analysis	
Base	General description of what data is stored:

information	A report on the spiking activity and/or on various variable for each of the compartment of the simulated neurons for each timestep of the simulation
Technical specifications	• Transient (Temporary): Input files (placement and connectome files) can be discarded at the end of the simulation • Permanent (Forever): up to 5TB/job for use/analysis, the mean size of a simulation is several GByte however, this can increase depending on the experiment (stimulus patterns, plasticity embedded, number of repetitions of tasks…) performed • ~3-4 jobs are expected at the same time. The mean size of a simulation is ~300GB, but new single cell models will be added, thus potentially increasing the overall size
Current solution	Name: Functional simplified circuit simulations (pyNEST) are run in HBP Collab and in HPC (e.g CINECA), Output files are spiking activity of each neuron (.gdf files). Single cell simulations are run in HBP Collab (NEURON as a Service) and in HPC (e.g CINECA), Output files are voltage and currents of each compartment. Validation is missing, Analysis and visualization are managed in HBP Local Collaboratory by interactive Jupiter notebooks
	URL to additional information: NA
	Limitations: NA

11.3.2 Data transport

Data transport: circuit object – Knowledge graph to Jupyter notebook transport	
Base information	General description of what data is transported: Circuit architecture is driven by Jupyter notebook (general placement and connectome). Single neuron models are plugged into from KG/NIP.
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: No current solution
	URL to additional information: NA
	Limitation: NA

Data transport: circuit object - Jupyter notebook to HPC centre transport	
Base information	General description of what data is transported: Circuit building placing and connecting the detailed single neuron models (smaller circuits can be built directly in Jupyter notebook)
	Data access patterns (request rate, transfer sizes):

	1 time per circuit usually
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: This does not exist. It is done manually from HBP Collaboratory to HPC centres with simplified neurons.
	URL to additional information: NA
	Limitation: NA

Data transport: circuit object – Knowledge graph/Jupyter to CSCS visualization VM transport	
Base information	General description of what data is transported: Make the circuit available on the CSCS VM
	Data access patterns (request rate, transfer sizes): 1 time per circuit if the circuit can be stored in the VM for a long time
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: no current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – HPC centre to Knowledge graph transport	
Base information	General description of what data is transported: Copy the simulation output to the knowledge graph storage
	Data access patterns (request rate, transfer sizes): several GB, 100+ per year (for GB ones), some will be TB An upper limit per year is currently not available.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: no current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – Knowledge graph to CSCS visualization VM transport	
Base information	General description of what data is transported: Copy the simulation object from the knowledge graph storage to the CSCS VM performing interactive visualization (network

	dynamics)
	Data access patterns (request rate, transfer sizes): 100+ year
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: no current solution
	URL to additional information: NA
	Limitation: NA

Data transport: simulation object – Knowledge graph to jupyter notebook transport	
Base information	General description of what data is transported: Move the simulation object from the knowledge graph to the jupyter hub kernel storage (for offline customized analyses)
	Data access patterns (request rate, transfer sizes): 100+ / year
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: no current solution
	URL to additional information: NA
	Limitation: NA

11.3.3 Data ingest/ GUI

In Jupyter notebook, GUI for reconstruction of user-defined circuit volume, stimulus setting, metrics calculation (analysis), visualization features and related parameters.

Data ingest: Name	
Base information	Description of input data source: detailed neuron single cell models (python and .mod files), python files with all parameters related to placement and connectome constructions
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): dimension of simulation volume, possible cellular types, types of connections and synaptic parameters, simulation input, duration and output (e.g. spike times, membrane voltage etc.)
Technical specifications	Characteristics of data: formats py, .mod and .hdf5; loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current	Name: NA

solution	URL to additional information: NA
	Limitation: NA

11.3.4 Data repository

NIP as source for single neuron models and NIP as repository for built circuits and simulation outputs. maintained by SP5? Storage allocation provided for a HPC projects/resources.

Data repository: Name	
Base information	Classification of the data objects (see below): In the current state of the research no information is available regarding the data repository needs.
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

11.3.5 Processing stations

Processing station: Parameter collection / control script	
Base information	General description of data processing: single cell optimization and circuit building
	Typical processing steps: single cell optimization
	Number of processing steps: hundreds
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.): entire software stack deployed by SP6
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: HPC	
Base information	General description of data processing: Simulations of cerebellar circuit with different stimulus patterns. With circuit characterized by physiological or "altered" structural and functional features
	Typical processing steps: neuron-specific dynamics, synaptic evolution, and signal transmission
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: It depends on circuit scale (scalability). CURRENTLY, a simplified circuit (pyNEST) with 98000 neurons and 4.5 millions of connections, simulated for 1 sec, on 95 nodes (36 CPUs/node) on BDW architecture
	Required software stacks (libraries, software frameworks etc.): pyNEURON, pyNEST
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: By exploiting MPI run, and compacting the output files (.gdf with spike times of each neuron) among cores (# threads), a pyNEST network is reconstructed and simulated for 500 ms in less than 1 minute with 3420 cores, in about 6 minutes with 720 cores. The reconstruction is done by splitting a priori the connection matrices for each core (following the assignment principles intrinsic in NEST), in order to provide each core only with the information useful to itself. When into the scaffold network, detailed multi-compartment neurons will be plug.in (thus running in pyNEURON), the runtimes/resources will change, also depending on how many compartments, how many state variables will be recorded, which plasticity model will be put on each connection, and so on
	URL to additional information: NA
	Limitation: NA

Processing station: NRP	
Base information	General description of data processing: Cerebellar circuit connected to input/sensors and output/actuators

	in a closed-loop task (Neuro Robotic Platform). A simplified version of the network (pyNEST) will be used to be integrated into the NRP, by exploiting translation into pyNN. Specific cerebellar tasks are defined (e.g. Pavlovian paradigms or upperlimb motion under force perturbations) into a simulated plant. The sensory information, from simulated sensors of the plant, (e.g conditioned and unconditioned stimuli, desired/planned and actual limb positions…) is fed as input to Mossy Fibers and Inferior Olive. The cerebellar output from Deep Cerebellar Nuclei is sent to simulated actuators of the plant. Transfer functions spike/analog signals are designed and implemented.
	Typical processing steps: Ad-hoc encoding/decoding of circuit signals, task-dependent movement generation
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.): pyNN (.nest and .neuron)
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

11.4 Use Case references

"Data-driven cellular models of brain regions: the Hippocampus and the Olfactory Bulb use cases" ICEI Co design workshop Migliore

"CDP2 Mouse-Based Cellular Cortical and Sub-Cortical Microcircuit Models" , Egidio D'Angelo/Michele Migliore

HBP SGA2 GA

12. Elephant big data processing (#7)

Elephant: interactive supercomputing for the analysis of neuronal activity

Use Case Description and Specification

29-06-2018 Michael Denker, Wouter Klijn, Kim Sontheimer

Partners	m.denker@fz-juelich.de k.sontheimer@fz-juelich.de
Institutions	
Principal Investigators	Sonja Grün

Date	Version / Change
20-06-2018	(Wouter Klijn) Initial template fill
25-06-2018	(Denker) Additional information
29-06-2018	(Kim Sontheimer) Estimation Elephant computate density.
29-06-2018	(Wouter Klijn) Add discussion from e-mail
28-08-2018	(Anne Carstensen) Editorial changes
04-09-2018	(Michael Denker) Review and update
24-09-2018	(Anne Carstensen) Integration of review comments and updates

12.1 Use Case Description

12.1.1 HBP SGA2 GA

Elephant

The Electrophysiology Analysis Toolkit (Elephant) is a toolbox for the analysis of electrophysiological data, i.e., activity data recorded either in experiments or neural network simulations. Elephant provides fundamental methods that are in use by the community to analyse both spike time data as well as time-series data of neuronal population signals, such as local field potentials (LFPs). Besides methods to characterise the dynamics of single neurons or population signal recordings, its focus is on methods that analyse the ensemble activity in massively parallel data, as well as methods that bridge scales of observation (e.g., spike-LFP relationships). The library follows several design principles. All analysis functions are based on the Neo data object model. This common data representation allows methods to be easily applied to neuronal data coming from different sources, including experimental file formats or neuronal network simulations. Furthermore, the library follows a modular design; such that complex analysis methods can be built from simpler analysis steps where appropriate. This approach guarantees results of complementary methods can be meaningfully related to one another. In order to follow a principle of co-design, methods are typically provided

by experts in utilizing a particular analysis function, or by authors of the original method. The library is structured by the types of analysis methods it provides. A full documentation is provided with the methods. Elephant is a toolbox for the analysis of electrophysiological data based on the Neo framework.

Elephant Visualisation
The component provides visualisations of electrophysiological data and of analysis results of such data obtained by means of the Elephant library. The component will deliver Python-based methods to visualize (i) source data represented in the Neo data model that allow to quickly view datasets given in that representation, and (ii) provide at least one standard visual representation for each analysis method contained in Elephant. The latter may be visualisations that are common practice in the field, or visualisations that mimic influential papers that have developed and/or applied the method.

Other Use cases
Elephant is named specifically in a large number of workflows for other SP use cases and tasks, such as SP3 (performing analysis on spatially distributed activity dynamics), SP4 (comparison of experimental activity data with simulation), SP6 (creation of a validation framework for neural network simulations based).

12.1.2 Science Case 1: Interactive analysis and control of running simulations

In this scenario, a scientist would like to access and analyse the results of a simulation on-line (i.e., while the simulation is running), and exert interactive control over the simulation. A common scenario, where this use case holds, is when scientists start a network simulation where it is unclear whether the parameter regime selected by the scientist is in the correct range for the simulation to display the desired dynamics. This scenario may hold in a situation where the network model is too complex to render itself to analytic treatment, or in plastic networks where the evolution of synaptic weights may evolve into pathologic network states. In such situations, scientists would like to avoid spending precious compute time on the full simulation of such a network, but detect the undesired state early on. Since check-pointing the simulation in order to continue later is often not feasible due to storage constraints, an on-line view of the dynamics is the desired option.

As a counter-measure to observed problems, the scientist would then like to exert control on the running simulation. In the most extreme case, this would be an abort of the simulation in order to start a new run. However, in some situations (e.g., when a certain basic network has already been learned by the simulation), the scientists would also like to play with a certain parameter of the simulation in real time in order to observe whether this parameter changes alleviates the observed problem in the output dynamics.

In a variant of this protocol, simulation results are additionally fed into an environmental feedback generator (e.g., a robotics simulation) that alters simulation parameters in response to the environmental change.

What is not possible so far:
• Early online feedback on the quality/validation result of a running simulation
• Interactive exploration of how certain simulation parameters influence the simulation
• Monitor simulations with time-varying connectivity (learning) and evaluate their suitability for further analysis
• Select interesting neurons to record from based on a preliminary analysis of the network dynamics

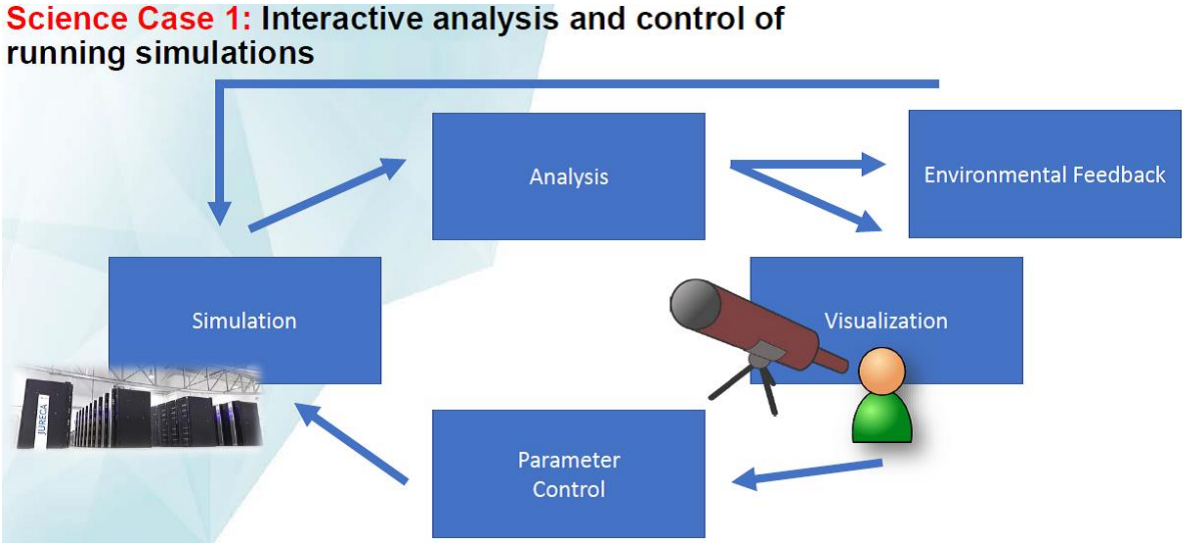


Figure 14: Interactive analysis and control of running simulations.

12.1.3 Science Case 2: Interactive, on-line explorative analysis backed by computational power of HPC

The analysis of experimental activity data, such as massively parallel spike train recordings from a behavioural experiment, often requires – at least in the first stage of analysis – an interactive and explorative approach to data analysis. The justification for this lies in the high degree of variability in the observed neuronal dynamics: Different neurons can vary drastically in their response properties, different artefacts may be hidden in the data which cause spurious results. Additionally, for many types of analysis many analysis parameters must be explored, and different types of analysis must be contrasted against each other.

In such a scenario, scientists currently often prefer to work with a mixture of analysis scripts and interactive shells on their personal computer. In this scenario, the data are moved to this computer, and computation is carried out locally. However, this approach puts limitations in terms of the size of tractable datasets, and the computational load of performed analyses. In addition, the exploration of analysis parameters in such a manner is cumbersome.

This science case aims at an HPC-enabled version of this scenario to lift these limitations. To this end, a possible scenario is one in which the user works on the personal computer through a control software, that handles (i) data management in the background by accessing a central data source and making available relevant data, and (ii) execution of the requested analysis on HPC resources based on on-line manipulation of analysis parameters through the control software on the user side, and (iii) performing a real-time visualization of the analysis result that are transferred back to the user.

A concrete realization of such a scenario is the investigation of the temporal evolution of the graph of spike correlations. Such graphs are generated by a variety of methods, and most of these feature distinct parameters that affect the graph structure. For example, for simple correlation matrices one may set a threshold on the correlation coefficient above which two nodes, respective neurons, are considered correlated and are linked by an edge. However, such methods are computationally involved. Therefore, an interactive exploration of the effect of parameters (e.g. bin size) on graph structure is currently not possible.

• Large data size prohibits transfer of data to individual workstations of users
• Users must be able to navigate and rearrange complex datasets

Requirements:
• HPC storage solutions provide services to visualize and analyse data on the server side, and on demand extract partial data for transmission to the client side
• Common data and metadata representations as interface

• Analysis is carried out in an explorative, interactive manner using remote compute resources
•
Requirements:
• On-demand execution of parallel analysis on server side, visualization of results, and transfer of end results to user side
• Interactive control of analysis parameters

12.2 Diagrams

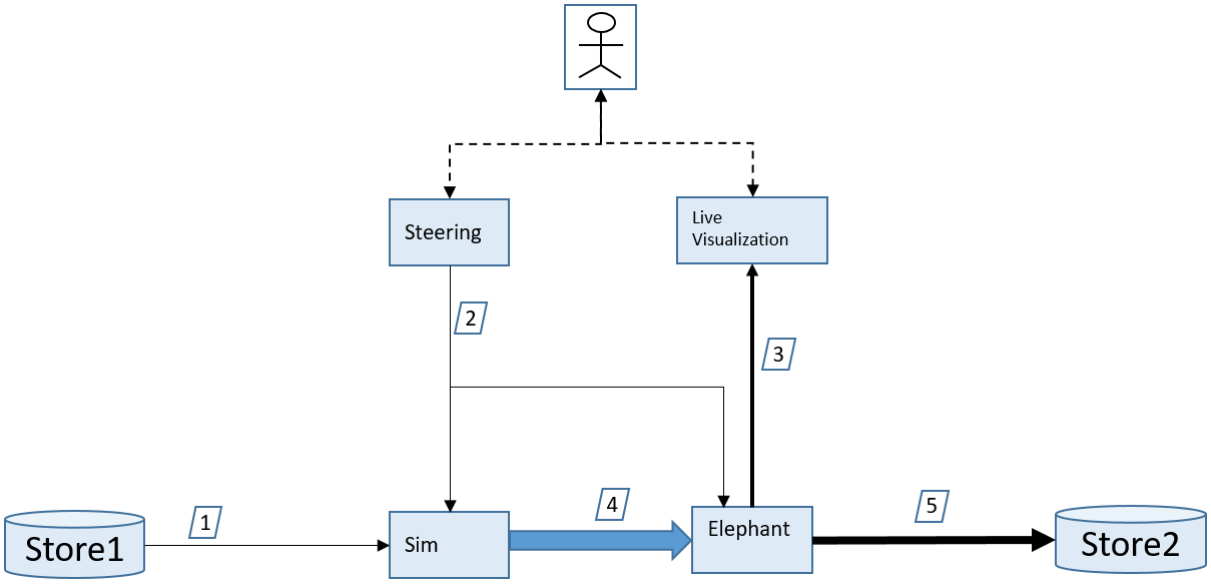


Figure 15: Schematic representation of the data flow for science case 1. Simulation data, fed by seed data from store 1, creates a continuous output stream of spike data that is analysed in an on-line fashion by the Elephant tool. Analysis results are visualized and displayed real-time to the user. In response to simulation output, the user interactively controls simulation and analysis parameters. On demand, simulation outcomes and/or analysis results are stored for archival.

12.3 Node Characterization

12.3.1 Data objects

Data object: ***1***, Spike and time series data from store to simulation (sim) (or directly to Elephant)	
Base information	General description of what data is stored: Spike time stamps of N neurons. The number of spikes per neuron per unit time is variable, therefore, the data object takes the form of a list of arrays, where each list entry corresponds to the data ("spike train") of one neuron. Each spike train has optional annotations in the form of key-value pairs comparable to a Python dict structure, which indicate metadata associated with each spike train. Time series data is sampled at regular intervals and may thus be represented as an NxM matrix, where M is the number of time series recorded from simulation, and N is the number of time stamps per transmitted time interval. As with spike trains, this signal may be annotated by key-value pair metadata. As data may contain multiple sets of time series sampled at different time intervals, multiple of such arrays may be existing. On the Elephant application side, the data objects are represented as SpikeTrain and AnalogSignal objects in the Neo Python library.

	On the data store side, Neo is the method to read data from disk, which may reside any of the file formats supported by Neo, including formats to store simulated data.
Technical specifications	• Permanent (Forever): The data is permanent in the store. It is used as external input to simulations in science case 1, or in science case 2 where recorded experimental data is analyzed.
Current solution	Name: Neo library
	URL to additional information: https://github.com/NeuralEnsemble/python-neo Nix library (file format) http://www.g-node.org/
	Limitations: NA

Data object: ***4***, Online spike and time series data from sim to Elephant	
Base information	General description of what data is stored: This data includes spike time stamps of N selected neurons in the simulation, as well as continuously sampled time series data levied in the simulation. The data is either a continuous data stream to Elephant, or a cumulative data bundle recorded over regular intervals. The number of spikes per neuron per unit time is variable, therefore, this data object takes the form of a list of arrays, where each list entry corresponds to the data (“spike train”) of one neuron. Each spike train has optional annotations in the form of key-value pairs comparable to a Python dict structure, which indicate metadata associated with each spike train. Time series data is sampled at regular intervals and may thus be represented as an NxM matrix, where M is the number of time series recorded from simulation, and N is the number of time stamps per transmitted time interval. As with spike trains, this signal may be annotated by key-value pair metadata. On the Elephant application side, the data objects are represented as SpikeTrain and AnalogSignal objects in the Neo Python library, represtively.
Technical specifications	• Transient (Temporary): The data is transient.
Current solution	Name: Neo library
	URL to additional information: https://github.com/NeuralEnsemble/python-neo
	Limitations: NA

Data object: ***2***, Steering commands for sim, Elephant, and LiveViz	
Base information	General description of what data is stored: These are commands sent between applications in order to evoke changes to the way the simulation, analysis, or visualization is carried out. In particular, these steering commands must: (i) inform the other components on what data they provide and which data they accept, (ii) inform other components about the adjustable parameters that expose to the framework, and (iii) allow to define actions to be taken in the components. Applications need a front-end to acquire, send and process these steering commands.
Technical specifications	• Transient (Temporary): The data is transient.
Current solution	Name: Nett
	URL to additional information: NA
	Limitations: NA

Data object: ***3***, Live elephant visualization	
Base information	General description of what data is stored: This data consists of generic data objects in the Python language, including in particular numpy arrays and dictionaries. In some instances these outputs may also have the form of spike train and time series as in data object 4. The output depends on the analysis function that has been applied. Also this information is transferred to the visualization component. The live visualization component will then launch a fitting visualization based on the analysis type. Currently no framework for Elephant analysis result data types exists, however, there are considerations to do this as an extension of the Neo library. The data will be generated by Elephant in fixed time intervals, and data is sent as a packet for each time interval.
Technical specifications	• Transient (Temporary) : The data is transient.
Current solution	Name: NeuralCorrelationAnalyzer (prototype visualization component developed by RWTH VR group under B. Weyers and D. Zielasko; includes definition of data model for correlation graphs)
	URL to additional information: NA
	Limitations: NA

Data object: ***5***, Output elephant to long term storage	
Base information	This data consists of generic data objects in the Python library, including in particular numpy arrays, dictionaries. In some instances these outputs may also have the form of spike train and time series as in data object 1. The output depends on the analysis function

	that has been applied. Also this information is transferred to the visualization component. The live visualization component will then launch a fitting visualization based on the analysis type. Currently no framework for Elephant analysis result data types exists, however, there are considerations to do this as an extension of the Neo library. Data are saved as numpy pickle, hdf5, or in the NIX file format. Do you need a Meta data server storage also? Michael: This is not clear to me at this stage. In terms of finding the analysis results later on, and in particular in terms saving provenance of the interactive work, most likely something like this must exist. However, I would put it at not too high priority at first – my feeling is that such a server would run parallel to the work described in this use case. A use case that would make use of such a metadata server could be one where the steering component can tell the Store 1 to select a specific input data set based on the a metadata query on such a server.
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it.
Current solution	Name: Neo
	URL to additional information: https://github.com/NeuralEnsemble/python-neo Nix library (file format) http://www.g-node.org/
	Limitations: NA

12.3.2 Data transport

Data transport: Long term storage to sim/elephant	
Base information	General description of what data is transported: Spike train data and time series data (see above)
	Data access patterns (request rate, transfer sizes): Data transfer in the range of Gigabytes Requested at the start of the scenario, infrequent thereafter
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: Steering commands from front end

Base information	General description of what data is transported: Control commands for communication between components
	Data access patterns (request rate, transfer sizes): Small data size Infrequent communication
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: Online visualization stream	
Base information	General description of what data is transported: Visualization data to display on the user side. This may consist of either individual pre-rendered frames of graphics directives.
	Data access patterns (request rate, transfer sizes): Continuous stream. Expected size on the order of typical movie streaming formats.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: Online spike train and time series data	
Base information	General description of what data is transported: Continuous stream of spike times and sampled time series data, or buffered transfer of the data in fixed time windows
	Data access patterns (request rate, transfer sizes): Assuming recordings from 100 electrodes, sampled at 1kHz and spiking at 10 Hz, data is on the order of 100*1000*2=200.000 bytes per second for time series data 100*10*2=2000 bytes per second for spike data Numbers are expected to grow. Data stream is continuous.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA

	Limitation: NA

Data transport: Transport of analysis results to long term storage	
Base information	General description of what data is transported: Data are generic numeric data and dictionaries in Python (see data object 5 above)
	Data access patterns (request rate, transfer sizes): Size of data is typically comparable to that of input data (1). In some cases, e.g., for Monte-Carlo type analysis where surrogates of the original data are created, temporary data (short-term) may be saved on the order of 1000 times the original data. Data are saved either at continuous intervals, or at the end of the simulation/analysis scenario.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

12.3.3 Data ingest / GUI

Data ingest: Store1	
Base information	Description of input data source: Stored spike trains from previous experiments. Data are either uploaded, or saved from a previous simulation.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Steering application	
Base information	Description of input data source: Data events are generated by a steering application, either scripted or in interactive mode.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Live visualization	
Base information	Description of input data source: Data are continuously generated by a visualization app.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

12.3.4 Data repository

Data repository: Store 1	
Base information	Classification of the data objects (see below): Activity data stored in file formats supported by the Neo library.
	Access control requirements: Possibility to limit access control is mandatory.
	Access requirements: NA
	Data availability requirements: Permanent storage.
Technical specifications	Maximum and average capacity requirements: Requirement: typically around 20GB per experiment to analyze, number of files <10. Typical number of experiments to consider in a study: approx. 100.
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: Number of experiments expected per year: approx.. 200.
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: Store 2	
Base information	Classification of the data objects (see below): Activity data and results data in file formats supported by the Neo library, and general purpose formats such as hdf5.
	Access control requirements:

	Possibility to limit access control is mandatory.
	Access requirements: NA
	Data availability requirements: Permanent storage.
Technical specifications	Maximum and average capacity requirements: Expected size per analysis project 200GB in results data, up to 1TB.
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: Number of files on the order of hundreds, each on the order of GBs.
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

12.3.5 Processing stations

Processing station: Sim	
Base information	General description of data processing: NA
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Elephant	
Base information	General description of data processing: Analysis and post-processing of neuronal data
	Typical processing steps: Filtering; calculating population rates; calculating spike train statistics; calculating cross-correlation coefficients
	Number of processing steps: Few per analysis instance (<10).

Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.): (?) • MPI • High bandwidth for on line analysis • Low Latency for live visualization
	Required software stacks (libraries, software frameworks etc.): General: • Python >= 2.7 • numpy >= 1.8.2 • scipy >= 0.14.0 • quantities >= 0.10.1 • neo >= 0.5.0 Specific: • scikit-learn >= 0.15.1 • pandas >= 0.14.1 Dependencies resolved with installation via pip.
	Ratio of data processing rate versus data consumption and production rate: Single core: Typical input size of 20kB, Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with ~4GFlops, between 1 second and 1 minute runtime. (4*10^9 Flop/s * 60 s) / 20.000 byte = 12 Mflop/byte
	Variability, availability, bandwidth and latency: Possibly random access to experiment data from user: high variability. Sequential access to each analysis instance. High availability and bandwidth, low latency (small input sizes: RAM and Cache) Variable output size. Random access. (Output size?)
	Additional information: Not accounting for number of memory accesses per input byte. Ratio of data processing rate to memory access rate per input byte ~1:1.
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

12.4 Use Case references

"The role of interactive supercomputing for the analysis of neuronal activity" Denker, 9-2-2018 ICEI co-design workshop.

12.5 Discussions

Alper: We used 1 second of artificial data which we generated via an homogeneous poison process in our analysis. Such a dataset containing 100 spiketrains (~15 spikes per spiketrain, since simulated with 15Hz) is around 40 Kilobytes. For this kind of dataset our analysis (tested with MPI) would take from a few minutes to several minutes but under 100 minutes dependent on the type of analysis or algorithm. We also have a chart for SPADE https://www.frontiersin.org/articles/10.3389/fncom.2017.00041/full (see figure2).
Our hardware environment was our blaustein cluster with nodes consisting of 2 × Intel Xeon E5-2680v3 processors with 2.5 GHz processing speed (48 cores but with Hyperthreading). I am not sure if one can come from these numbers to byte per operation.
Michael Denker: 1. The transport of data to the simulator or Elephant

a. Where is this data stored? Local computer or in some long term storage?
I think this could be both, but most likely, the data is stored either on a long term storage visible by the HPC infrastructure, or on a web-based data base (e.g., a git annex server, or some other http server where one could download the data from).

b. Could you confirm the size of the data: 200 * 20GB = 4 TeraByte/year
I think the order is about right for the science case covering experimental data. For our new experiment we had now about 2.5TB coming in over 6 months, however, experiments will not go on continuously. So in other words, if a monkey is implanted, maybe we get a bit over 4TB per year, at other times less. In general, I would assume that for the current experiment, there will be about 3-4 monkeys in total over the next 5 years or so, each totalling on the order of 4-6TB each. Of course, we will work on this data for a long time to come.
So yes, I think the figure is about right for the interactive data analysis scenario.

I think for the network simulation scenario, it should be rather lower figures, but this I cannot judge this at all.

2. Elephant to long term storage:
a. A confirmation of the size of the data: 200 * 1TB = 200 TeraByte/year
This is really difficult to say because it depends a lot on the research project and how many scientists will use the system. However, 200TB per year sounds way too much. I think I misunderstood some calculation in the background.
Let's put it this way: The data I have for one past research project totals at about 1TB, however, this is already with a clean-up of everything unnecessary. So, I think, one research project that uses the interactive facilities would probably be in the range of about 1-3 TB during operation. Then, it's mainly a question of how many projects are done by different scientists per year, so let's assume 2-3 in the beginning. That would more be on the scale of about 10TB per year.

13. Ilastik as a service on the HBP Collaboratory (#13)

Ilastik as a service on the HBP Collaboratory

Use Case Description and Specification

29-06-2018 Wouter Klijn

Partners	anna.kreshuk@iwr.uni-heidelberg.de jeffrey.muller@epfl.ch
Institutions	
Principal Investigators	

Date	Version / Change
29-06-2018	Add basic contact details and partial information from e-mail
06-09-2018	(Wouter Klijn) Recreate template from deliverable
25-09-2018	(Anne Kreshuk) Insert information
02-10-2018	(Wouter Klijn) Merge documents, add disclaimer (section 2)

13.1 Introduction

This use case description and specification document provides a tool for developers and scientists to collaboratively transform a free form description of a science use case into technical specifications. Specifications that guide the implementation of hardware and software fulfilling the science use case. This document should help a project in a number of ways: its structured methodology will help to find the essential parts, and it will assist in separation of the ***must*** haves and ***nice*** to haves [1]. The specifications should result in a standalone document that can be given to new partners of the project as introduction into the science and technical details of the project. On a more abstract level this document could be seen as a contract formalizing the expectations of both, the engineer and the scientist.

An important guideline when creating a use case analysis document is the separation of user requirements and technical details. A user is ultimately only interested in the functionality of a software / hardware product and not in the underlying technical details of the implementation. Separating these concerns is a non-trivial matter: This document will therefore typically be written in an iterative manner, with the document bouncing from scientist to developer getting more detailed on each iteration. It will also be living document: details of the project can and will change over time; Components might be hard to implement and trade-offs might be made depending on availability of manpower. The amount of work needed for this document might appear large, however it is work that, for a typical software/science project, should be performed anyways.

The different elements/chapters in the template should be kept in order and contain the content described. This will allow comparison of use cases and allow identification of shared / overlapping functionality. This document and the accompanying PowerPoint introduce a set of visual components that can be used to describe the use cases and systems (Section 1.2). The symbols should cover the majority of systems encountered, but if the need arises, new elements can be introduced. Do keep in mind that this will complicate comparison of the diagrams created. The main goal for collecting the information is to foster the reuse of efforts and components. Although the introductory chapters can be removed, it will limit the use as an introduction for new project partners.
In the next sections the goal of the individual parts of the template will be introduced. The first section (1.1) details the use case description, it should provide the scientific reasoning behind the case. Section 1.2 explains the set of visual components that can be used to create the model diagrams. In section 1.3 we provide the typical data point that can be used to characterize the different components in more technical detail. In section 1.4 we explain list of potential infrastructure requirements specific questions. High-level needs and services that can be cross-checked with the node characterizations.
Section 2 is the actual template, it contains just the titles and list of infrastructure questions. Other components can be copied from the introduction chapter 1. If you add multiple diagrams/systems it is best to copy the template multiple times, or, use different documents. This will improve coherence in the descriptions.

13.1.1 Use Case Description

The workflow description is a high-level description of the workflow of the use case. It is typically written by the scientist and provides the reasons why to build or use a software or hardware system. Topics that might be encountered in this section are: How new (or better, bigger, faster) science is possible with this software. Problems and challenges encountered in current software.
Typically, the workflow is broken down in steps with partial goals for each step. It is advisable to keep implementation and technical details out of this section. Implementation details are not part of the description: An example of such an ***implementation detail*** would be: "The software must be fast, to allow fast turnover of experiments. ***We have to use GPUs***". A complete separation of concerns is hard to arrive at. It is one of the more complicated exercises in system design. Having a starting point is more important that being completely correct. This is one of examples where the dialog with technical experts will help to arrive at a correct description.

An example of a science (and not technology) centric description:
"As a researcher I want to be able to perform a large scale computational experiment. This experiment cannot be performed on my local cluster due the size of parameter space I want to explore and data being stored at the CSCS storage. Some of the analysis of the results will need to be performed in my local institute as the computational resources will be sufficient for post processing. Besides, the important results should be stored at the central storage, with intention of submitting them for curation and sharing

with others. The metadata for the results in the form of KnowledgeGraph links should be preserved.

Two widely different technical solutions would support this case (We need both depending on the user workflow):

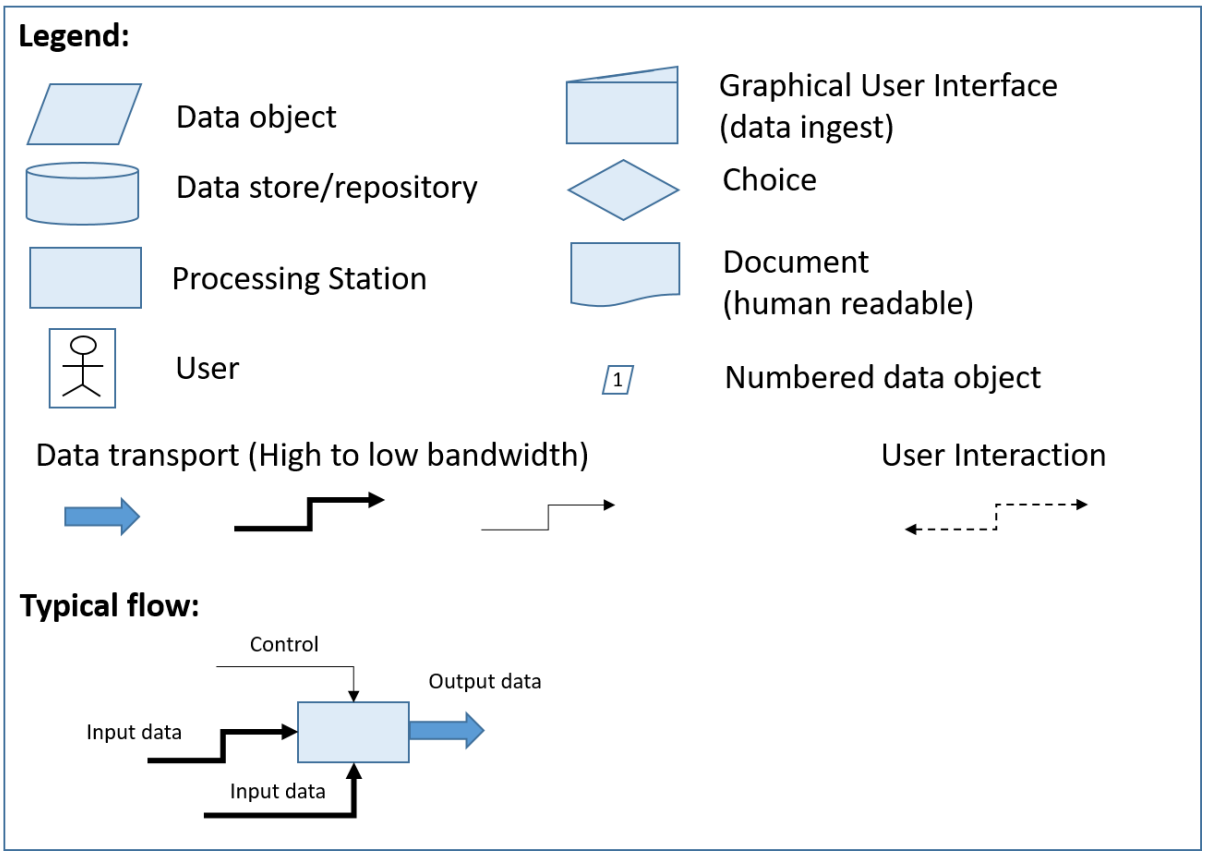
1. Analysis of results on a virtual machine with data staying in a central location. Results selectable via a database, accessed via a web interface.
2. Transport of results to the local cluster with processing on the local machines with the data stored in clearly labelled directories.

Which of these solutions is implemented can now be made on available resources, software limitations, etc.

13.1.2 Annotated Use Case Diagrams

An annotated use case diagram is a relatively freeform graphical depiction of the textual description as detailed in section 1.1. We would suggest to use the diagram components as shown in Figure 1. As this will allow easy comparison between different use case descriptions. The flowcharts in this document follow the practices as described in [2], [3].

Figure 1: Overview of suggested symbols for a use case diagram. The symbols are based on [2], [3]. The symbol for GUI is a combination of processing station and data object. A suggested typical data and information flow is shown. Additionally, a simple bandwidth range is depicted. An editable version of the diagram below (a PowerPoint presentation) will accompany the current document.



To prevent cluttering of complicated workflow we suggest the following:

- Make use of specialized symbols to allow for a visual distinguishing of salient features (GUI would be an example).
- Use only a small pictogram for data objects annotated with a number.
- Use the suggested locations for the connectors: Control at the top; Inputs from the left or bottom; Outputs leave on the right side.

To reiterate: these are suggestions, the diagrams are in principle freeform and not all symbols might be used in your specific use case.

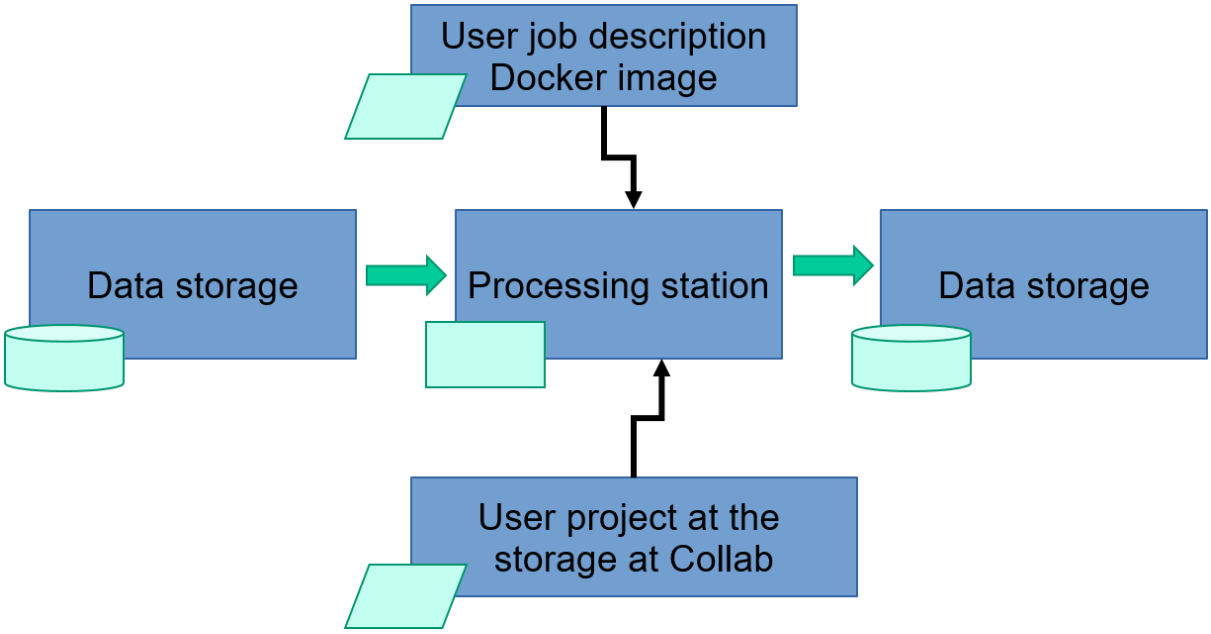


Figure 16

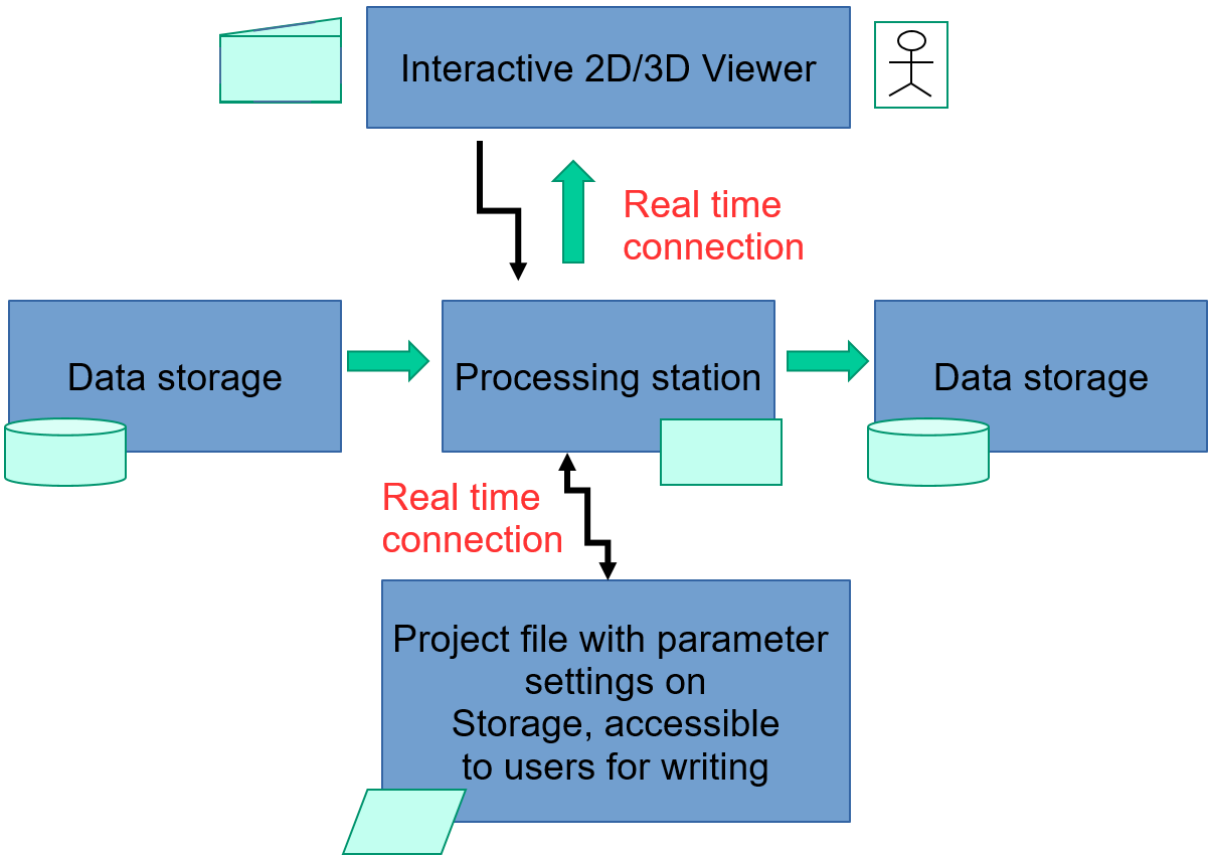


Figure 17

13.1.3 Node Characterization

In this section a characterization of each component is depicted in the annotated use case diagram. This is done in a table format with typical information points listed. The entries are typically split in different sets: The ***base*** information set without which an informed discussion might be complicated; The description is typically at a user / functional level. Secondly, **technical specifications** of the requirements. The use case is not yet solved thus this information will by necessity be added incrementally and optionally by a domain specialist. The third information set is regarding ***current solutions*** that one is aware of.
Not all information might be available. Fill in what is known at this stage. Having a start point for a dialog is more important that having perfect information, especially in the beginning stages.

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: needed per node, in total at least 8GB per core, assuming 1 job per core
- IO: bandwidth, latency, always on/dedicated for interactive connection (2nd diagram), very low latency from user to processing station (VM). High bandwidth to and from data storage.
- CPU: large size jobs / farming scales linearly with the input data size

- Specialized hardware: (GPU, KNL, FPGAs) At least a few GPU node access preferable, with standard NVIDIA GPUs configured with CUDA for deep learning
- Storage: size, access rate Multiple datasets, from 10s of MBs to TBs, depending on the curation process
- Specialized software: VM/containers VMs for interactive use case (2nd diagram), Docker containers via Shifter (1st diagram)
- Specialized features: in-situ visualization

Architecture Requirements:

- Minimal compute performance (excluding acceleration)
- Minimal volatile memory footprint of 192 GByte
- MPI point-to-point bandwidth of 10 GByte/s or higher
- MPI latency of 2 micro-seconds or less
- Access to active data repositories with a bandwidth of up to 8 GByte/s per node
- GPU requirements per node (minimum)
- GPU configuration (minimum HBM)

13.1.3.1 Data objects

Data object: ***Number in diagram***, name	
Base information	General description of what data is stored • Formats precomputed image tiles or blocks, such as from DVID image service, allowing for partial access to large imaging datasets • Metadata as in the KnowledgeGraph • Database requirements
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. • Short-term (Campaign): Data used throughout the execution of the scientific workflow. • Permanent (Forever): Data outliving the machine used to generate it. Additional information
Current solution	Name
	URL to additional information
	Limitations

13.1.3.2 Data transport

Data transport: Name	
Base information	General description of what data is transported
	Data access patterns (request rate, transfer sizes)
Technical specifications	Maximum required bandwidth
	Average required bandwidth
	Interface requirements for attached entities
	Additional information

Current solution	Name
	URL to additional information
	Limitation

13.1.3.3 Data ingest / GUI

Data ingest: Name	
Base information	Description of input data source
	Description of data introduction (upload? scanner characteristics? simulation characteristics?)
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports
	Additional information
Current solution	Name
	URL to additional information
	Limitation

13.1.3.4 Data repository

Data repository: Name	
Base information	Classification of the data objects (see below)
	Access control requirements
	Access requirements
	Data availability requirements
Technical specifications	Maximum and average capacity requirements
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time.
	In terms of size & file number
	Additional information
Current solution	Name
	URL to additional information
	Limitation

13.1.3.5 Processing stations

Processing station: Name	
Base information	General description of data processing See use case descriptions
	Typical processing steps execute workflow
	Number of processing steps 1 step, potentially with follow-up Jupyter jobs acting on the produced data
Technical specifications	Data processing hardware architecture requirements
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies Shifter • Need for licenses
	Ratio of data processing rate versus data consumption and

	production rate CPU-bound
	Variability, availability, bandwidth and latency: Data consumption access pattern Data consumption access pattern for the interactive use case: random block access into large imaging datasets. For the batch processing: see above Data production access pattern
	Additional information
Current solution	Name
	URL to additional information
	Limitation

13.1.4 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? • A few hours at most • What software stacks need to be available? • VM with our Docker image • Is it possible to define memory capacity requirements? • At least 32GB, for 3D data preferably more
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services?
Virtual Machine Services	• Which parts of the workflow require such services? • See above
Active Data Repositories	• Which parts of the workflow require such services? • In blocked (tiled, DVID) format, all parts of the workflow
Archival Data Repositories	• Which parts of the workflow require such services?
Data Mover	• Which parts of the workflow require such services?

Services	
Data Transfer Services	• Which parts of the workflow require such services? • Between which ICEI sites is data planned to be transferred? • How much data is expected to be transferred per time unit? • How are transfer patterns expected to change over time?
Data Location Service	• Which parts of the workflow require such services?
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site?
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services?
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples: ○ Special accounts for running services ○ Needs for fine-granular control of access to data ○ We need to start jobs from the Collab, preferably without a special account beyond the HBP one
User Support Services	• Are the specific foreseeable needs for user support services? • Yes, and also developer support

13.1.5 Use Case references

Detailed use case description attached.
Remark (Wouter Klijn): Included here as text

Copied from collab 2018-09-25

JP-UC09 Ilastik-as-a-Service workflow in CSCS
For now this should be considered a placeholder for SP5-VA-UC03a and SP5-VA-UC03b found in SGA1-D5.8.1

Use case owner: Anna Kreshuk
Users:
• Ada: is a neuroscientist, not necessarily part of HBP, but with an HBP account and an interest in processing images with ilastik

Preconditions:
1. Data has been released in a process ending in SP5-UC04
2. Data is available in a format suitable for the NeHuBa, the 3D image data viewer based on Neuroglancer. Such formats include precomputed chunks, dvid

Success Scenario:

1. Ada browses the collaboratory to discover interesting datasets
2. She finds a dataset she wants to work with and opens a Nehuba viewer to look at the data.

---- Branching Point;
---- Branch A:

1. She decides that the data she is looking at is sufficiently similar to data she has already trained an ilastik pixel classification project and she wants to see how the prediction/classification performs on this dataset.
2. She presses a button in the Nehuba interface which uploads her ilastik project file to an ilastik server docker instance dedicated to her and this job, running on a virtual machine provided through openstack.
3. Once the upload is complete, the predictions will be computed on the server and displayed in Nehuba
4. Adas ilastik project might involve computations on GPUs. The virtual machines have to be prepared to make available GPU resources.
5. Ada navigates through the dataset and views predictions made by ilastik on the fly in different regions of the dataset. She decides that the trained ilastik project is good enough for use with this dataset. She triggers the batch prediction of the whole dataset.

---- End of Branch A
---- Branch B:

1. Ada wants to train a new ilastik project in NeHuBa. She presses a button in the UI of the viewer that makes available an ilastik server docker instance dedicated to her and this job, running on a virtual machine provided through openstack.
2. Ada can train an ilastik classifier by supplying sparse annotations with drawing tools provided by NeHuBa.
3. She can trigger live updates that retrain the classifier based on the provided annotations and display the predictions computed on the remote ilastik server for the current field of view in the NeHuBa viewer.
4. Ada navigates through the data to see how the classifier performs in different regions. She supplied more annotations in order to better train the classifier where the predictions are not accurate enough.

---- End of Branch B
---- Branches merge:

1. The batch prediction is queued to be processed on a supercomputer. The predictions are generated on the supercomputer.
2. The generated predictions (N_voxels_image * N_prediction_channels * 32bit) are transferred to the image storage and is made available through the knowledge graph. Ada can adjust access rights/visibility of the data. Furthermore, the produced pixel probability images are stored in a way suitable displaying them in the NeHuBa Viewer (e.g. precomputed chunks, dvid).

Target Scenario Execution time:
Current Scenario Execution time:

13.2 Case specific use case information

Disclaimer (Wouter Klijn): This use case was received as three separate document. The template document with sparse information inserted in the explanation section and examples. These answers are marked light grey in the text. Additionally, two diagrams were send. These are inserted as figures 2 & 3. Finally, a use case description that has been added in section 1.5.

13.2.1 Discussion

Anna Kreshuk:
In terms of hardware we don't need anything unusual, but we need to run VMs, preferably with GPU access. The sizes will vary depending on the user job. We are planning to make a more detailed analysis of the existing science use cases in early July.

For the very big jobs (>1TB), never more than 1 concurrently and in total, I would expect, less than 10. This can change if more huge datasets get into the system, but for now that's what I would plan for. I don't think more than 10 users will use the system simultaneously, but the computation runs on multiple cores. Right now the regular datasets are in the <10GB range, but this will likely change as more data enters the system, especially light-sheet data from the Pavone lab.

Concerning the GPUs, they are needed for deep learning models. We could potentially scrap by CPU-only at inference time, but we'll have to train smaller networks and the results will likely get worse. However, I don't think we need many of those. Also, more groups in the HBP are exploring these models now, so I think they'll be needed not just by us.

Dirk Pleiter:
I also noticed the request for processing a very large amounts of data. We are considering of adding high-performance SDDs that could provide both, significant memory footprint (likely accessed only through IO interfaces) as well sufficient bandwidth. However, the extra costs per node are significant. Therefore, a good justification is needed.

The same concerns the integration of GPS. Server-class GPUs can easily double the costs per server. As the budget is limited this means less servers. This may be the right thing to do, but only if you can use the GPUs efficiently. Adobe your services will use GPUs, if would be important to understand, how many nodes should be suitable for this kind of services.

14. Online visualization of multi-resolution reference atlases (#14)

Online visualization of multi-resolution reference atlases

Use Case Description and Specification

29-06-2018 Pavel Chervakov, Wouter Klijn, Timo Dickscheid

Partners	Pavel Chervakov
Institutions	INM1 FZJ
Principal Investigators	Timo Dickscheid

Date	Version / Change
13-06-2018	(Wouter Klijn) Initial scientific write down and technical workflow breakdown
14-06-2018	(Wouter Klijn) Merge in SGA1 template information
21-06-2018	(Wouter Klijn) insert TD information on split use case
25-06-2018	(Pavel Chervakov) Added additional reference information
29-06-2018	(Wouter Klijn) Add discussion information
07-09-2018	(Klijn and Carstensen) Recreate template plus editorial changes
14-09-2018	(Timo Dickscheid) Remove redundant parts from other INM1 use cases

14.1 Use Case Description

14.1.1 SGA2-SP2-UC003 - Interactive access to multilevel human atlas data through the HBP atlas

SP2 and SP3 are aggregating a unique portfolio of high-quality human brain templates, maps and multilevel data. To generate an impact for the wider community, these data have to be provided in a simple, user-friendly fashion to users of the web interfaces and APIs of the Neuroinformatics Platform. At the macroscale, we will provide neuroimaging researchers with the data and functionality to interactively overlay HBP's whole-brain maps and parcellations with their own data in the template space of their choice, and subsequently download the transformed data for their own purposes. Here, we address the fact that today, the existence of several incompatible reference spaces used to aggregate human neuroimaging data is an impediment for meta-analysis. By providing a set of spatial transformations between the main reference spaces in the Neuroinformatics Platform, we provide a simple solution to the outside community. As of today, no other repository provides a comparable set of cross-aligned labelled human brains, and the possibility to adapt them to most of the standard spaces.

At the mesoscale, we will target modellers and theoreticians, and provide them interactive access to realistic numbers of cell counts for different brain areas. SP2 will provide realistic experimental 3D measures of neuronal cell numbers and densities that go significantly beyond the tables provided by von Economo. Such quantitative numbers are still missing today, in spite of being a critical requirement for setting up simulations of cortical network models. We will co-design the functionality to comfortably retrieve such data from the HBP atlas by visual exploration, or by API access in Python.

Key problem here: We have very large, high-resolution 2D and 3D images that cannot be downloaded en bloc by the user for visualization. For visualization, such data needs to be streamed dynamically as multi-resolution tiles via http, so that only the data that the user actually looks at is transmitted: Either large parts are downloaded at much lower resolution, or small parts at the full resolution. This process is realized by streaming through https, as known from Google maps for the 2D case.

- Typical formats of such images: stacks of bigtiff images, hdf5, sometimes stacks of png images
- The Big Brain is a prominent but still moderate example: 1TByte of volumetric image data at 20 micron isotropic resolution. For analysis, it is stored as stacks of 2D png images along the 3 orthogonal planes. Future datasets will be even larger: INM-1 develops in the next years a 1 micron brain which will be 1-2 PByte of image data (details: M. Huysegoms, INM-1). More datasets in the order of TBytes will come from Polarized Light Imaging (M. Axer, INM-1) and 2 Photon Imaging (F. Pavone, LENS).
- Streaming of the data for online visualization can be achieved in two ways:
    1. Convert the dataset explicitly into a large number of multi-resolution chunks that can be individually addressed by a URL scheme, depending on position and scale. This is the current strategy for smaller reference datasets, leading to two different file formats (and consequently storage systems) for visualization and analysis of the same dataset.
    2. Implementing a backend “image service”, which has very efficient access to the data in its original format, extracts tiles on request at the required scale and position, and streams them via http. This is envisioned in SGA2/3 for as during interactive segmentation).
- In upcoming years, SP5 will most probably rely on both: 1.) Makes sense for reference atlas data that is accessed by many users but is updated very rarely, while 2.) is needed for images that are accessed by a smaller group of users, but change frequently if not in realtime. This is the case for preview visualization during data ingest, processing and analysis (see, or even interactive image segmentation workflows (as established in WP5.6; ilastik software). **This is also required to continuously monitor the data acquired and processed from high-throughput microscopy at INM1, as detailed in use case 15 “Data management and big data analytics for high throughput microscopy”.**
- An efficient image service will require to efficient random access to subparts of the files (as in hdf5 or bigtiff). We expect that this requires a different file system

than the SWIFT currently provided by CSCS for SP5 (since I understand SWIFT allows only to retrieve complete files, not parts of files). Computational requirements still need to be analysed, most knowledge here maybe have Jeff Muller (EPFL) or Pavel Chervakov (INM-1).

Remarks:
Conventional storage systems are usually designed for high data throughput, meaning that the user can efficiently stream large parts of continuously stored data at the expense of random access speed. On the other hand, backend "image service" by its very nature requires high number of IOPS with low latency. Since it's almost impossible to predict which area of high-resolution dataset will be requested next, usual pre-fetching and caching techniques on the application level are not expected to be helpful.
A pilot prototype implementation of such a service is being developed at INM-1, where a chunked multi-resolution 3d brain volume is assembled on demand from original 2d scans of brain slices in tiled bigtiff format, applying provided necessary transformations to individual slices as well as partial downscaling on-the-fly. This work is at the very early stage, so there are no metrics and benchmarks available yet, but it is already evident that data access is a bottleneck for this implementation unless data is served from local SSD, which inherently provides faster random access speeds. Therefore, fast random access to the underlying storage system is of paramount importance.

Expected needs:

- Multiple VMs with high bandwidth, continuous random access to large images stored on the federated storage (thousands of image files in the up to TB range each, or volumetric hdf5 in the range of many TBs)
- VMs will run web services used by possibly hundreds of concurrent users, so the webserver will run many parallel threads.
- Memory and CPU requirements for the VMs have not been evaluated, maybe Jeff Muller (EPFL) has a good guess. GPUs most probably not required.
- Storage needs to be constantly accessible at good bandwidth. In the scope of 5 years, up to 10 Petabytes might be needed
- These are no HBP batch jobs. It is a web service with considerable large random data access demands.

Infrastructure need:
This Use Case requires the availability of the core functionality of the SP5 atlas. Most importantly, it requires a working installation of the metadata database (knowledge graph) holding information about the required human datasets, as well as web-accessible storage that is linked to the Neuroinformatics Platform. To implement the front end functionality, we build on the interactive atlas viewer developed in WP5.4, which will have to provide a plugin mechanism to implement the graphical user interactions. The Use Case also requires hardware to run the back end services for applying pre-computed spatial transformations.

One important question here is where we store the microscopic scans described in use case 15 after the processing steps documented there. Neuroscientists at

INM-1, and in the future also outside FZJ, will frequently request online visualization for arbitrary (e.g. not known in advance) selections of these images. The remote visualization should be able to access them in reasonable time. Assuming however that the data cannot all be kept on "hot" storage systems, fast staging of images that had been moved to long term archival storage would be needed.

14.2 Diagrams

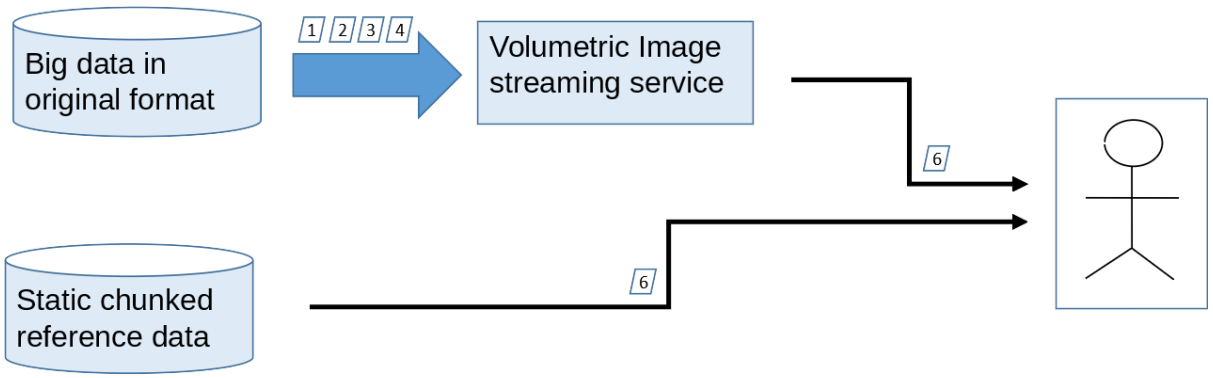


Figure 18: Data-flow of multilevel human atlas data visualization.

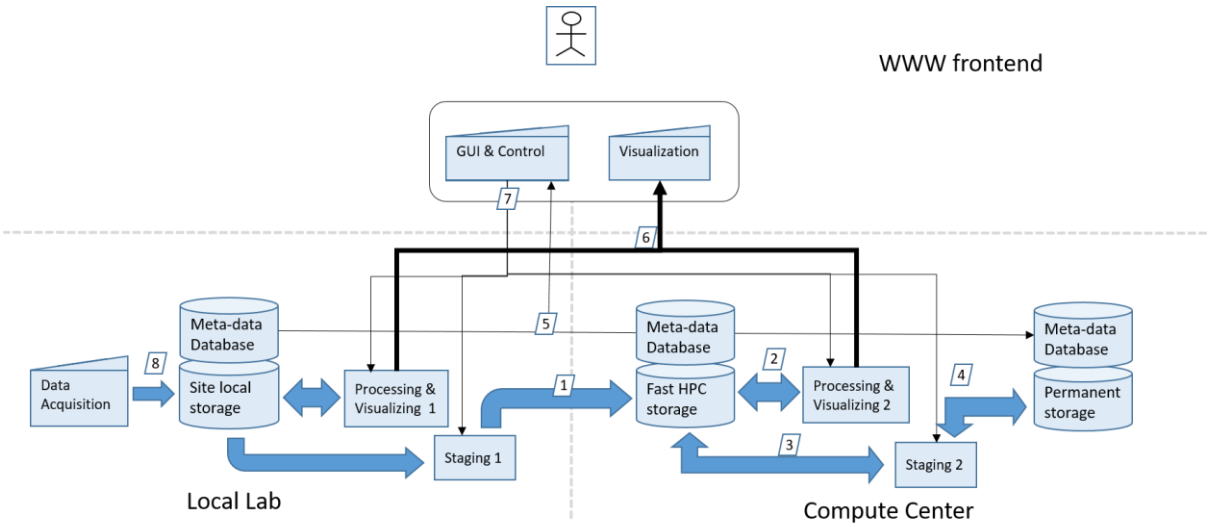


Figure 19: This diagram is by necessity an abstraction from your science use case.

14.3 Node Characterization

14.3.1 Data objects

Data object: ***5*:** Science data product: Meta data	
Base information	General description of what data is stored: • Formats: NA

	• Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***6*****:** Remote visualization	
Base information	General description of what data is stored: Multi-resolution chunked representation of volumetric big-data to be visualized in web-based interactive viewer. • Formats: viewer-specific data format, currently "precomputed" format of neuroglancer is used • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: NA
	URL to additional information: Description of data format: https://github.com/google/neuroglancer/tree/master/src/neuroglancer/datasource/precomputed
	Limitations: NA

14.3.2 Data transport

Data transport: **6,** In situ 3D visualization data transport	
Base information	General description of what data is transported: The data being worked on and visualized in these workflow is too big to store at a local pc. Visualization will have to be done in-situ. "An extensible interactive web-based 3D atlas viewer with interactive brain region selection." (SGA2-SP2-UC003) The visualization should accessible from external location and should be integrated into the collab This is partly in-situ so a low latency is needed A http service reads image tiles from a subset of 1 micron tissue scans and segmentations stored on the fast storage. Read access depends on user requests. The serves is only provide within INM1, we expect 10-15 users per day accessing some sections.

	Data access patterns (request rate, transfer sizes): This would be mostly during the day (Europa) with some world wide access. Sizes are directly related to the duration of the session * streaming rate. The maximum number of users needs to be determined
Technical specifications	Maximum required bandwidth: N * 8 MB/s when streaming a screen Lower when structure data is send
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **7,** Control signals for in-situ visualization and processing control	
Base information	General description of what data is transported: Message based control data for performing the remote actions on the HPC This is partly in-site so a low latency is needed
	Data access patterns (request rate, transfer sizes): This would be mostly during the day (Europa) with some world wide access. Sizes are minimal
Technical specifications	Maximum required bandwidth: N * 8 MB/s when streaming a screen Lower when structure data is send
	Average required bandwidth: low
	Interface requirements for attached entities: Message based: current design of the connecting systems is unknown
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

14.4 Discussion

Pavel Chervakov: scanners scan brain slices and upload them to JSC. These are very large 2d images.

There is a machine at INM-1 with NFS mount to JSC which is “listening to request for visualization…” etc. and people here are using this service.

To view those huge images and work with them. In 2d, plain picture, one slice at a time. Images are already in JSC storage.

This 2d service reads one image file per user at a time.

What I was writing about is a 3d service. Which assembles a 3d volume from those 2d images in realtime: 3d service reads many small 2d tiles from many 2d files at once to produce one 3d chunk and that's why we are talking about fast random access to the storage there.

Therefore, there are currently two image services we are talking about:

2d service. Deployed and in operation at INM-1. Has internal INM-1 users (I don't have a number).
3d service. Under development. Working prototype. Not in production. No users yet, only used by developers. Potentially will be used for everything. Is going to be used in HBP.

From e-mail Timo 14-9-2018:

You should be aware that this addresses ad-hoc remote visualization over the web for large data in the context of brain atlases. A special case of this is the „http visualization", which is mentioned but not further documented in use case 15.

15. Data management and big data analytics for large cohort neuroimaging (#17)

Data management and big data analytics for large cohort neuroimaging

Use Case Description and Specification

29-06-2018 Timo Dickscheid, Wouter Klijn, Felix Hoffstaedter

Partners	Felix Hoffstaedter and Jan Schreiber
Institutions	INM1/INM7 FZJ
Principal Investigators	Timo Dickscheid

Date	Version / Change
13-06-2018	(Wouter Klijn) Initial scientific write down and technical workflow breakdown
14-06-2018	(Wouter Klijn) Merge in SGA1 template information
21-06-2018	(Wouter Klijn) insert TD information on split use case
29-06-2018	(Felix Hoffstaedter)
07-09-2018	(Wouter Klijn / Anne Carstensen) Recreate template and editorial changes
14-09-2018	(Timo Dickscheid) Remove redundant parts from other INM1 use cases
19-09-2018	(Wouter Klijn) Review adaptations Timo, additional questions.

15.1 Use Case Description

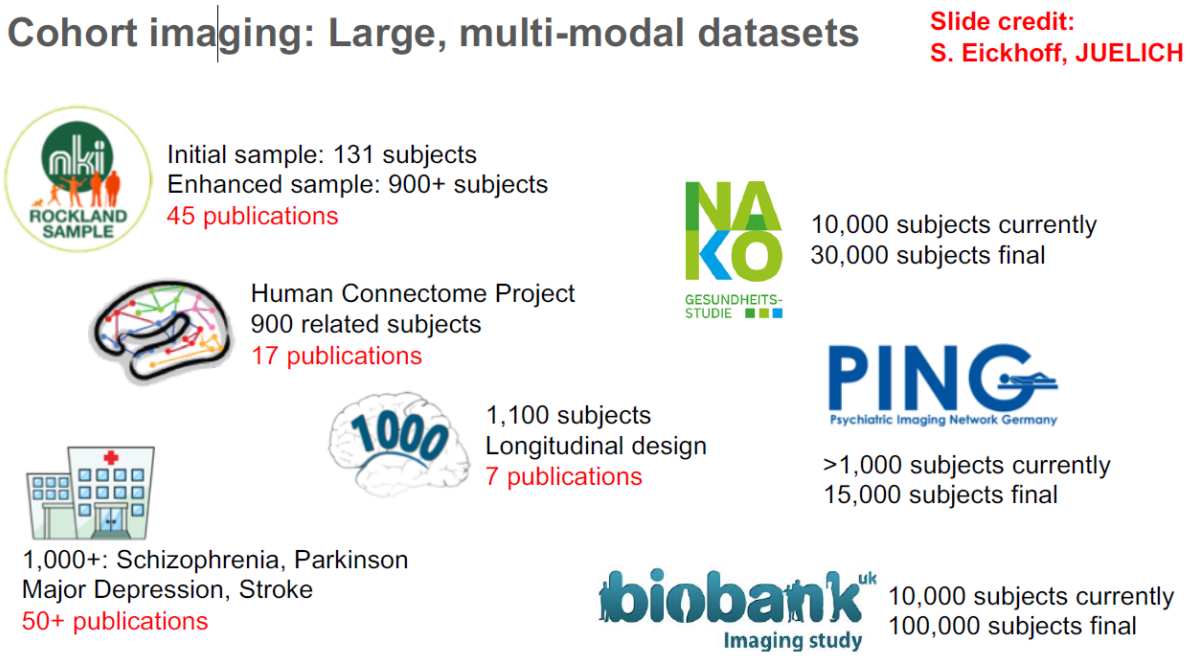


Figure 20: Overview of the expected data sets to be included in the Brain Atlas.

Neuroimaging analytics: important aspects

- Large amounts of data to be processed
  (HCP uncompressed data for 1200 subjects ~100TB)
- Data not coming in continuously - coupled to releases of repositories
- Huge amounts of single files (eg. NIfTI) cause problems with inode quotas
- Inherently parallel preprocessing pipelines (applied to each image)
- Frequent sampling of different subsets of derived data for analysis
  Planned to become an NIP service
- Software and workflows well established and standardized, but image formats not designed for HPC

Structural & functional MRI Preprocessing of raw imaging data including external datasets

Currently 11.333 datasets 101.970 files / 81,6 GB
Pre-processed for analysis 1,73 million files / 32,9 TB
Data subsets will be highly variable and mostly non-identical data query at least once a day
Machine & deep learning approaches need all available data

Module	1000BRAINS 1300	NKI Rockland 800	HCP 1200	ADNI 1000	Patients 1100	# data sets per node	total # data sets
Preprocessing							**41,900**
Brain Structure: freesurfer	1300	800	1200	1000	1000	24	5,300
Brain Structure: FSL	2600	1600	2400	2000	2000	48	10,600
Functional Connectivity: FSL	2600	4800	13200	1000	1000	48	22,600
Structural Connectivity: Preprocessing	1300	800		1000	300	15	3,400
Group Analysis							**48,500**
Randomise: RS & VBM	5000	5000	5000	2500	5000	24	22,500
Randomise: Network-based	20000	2000	2000	1000	1000	24	26,000
Modelling							**22,600**
Local Model: Tensor	2600	800	2400	1000	300	48	7,100
Local Model: CSD	1100	800	1200	1000	300	14	4,400
Local Model: BedpostX (GPU)	1100	800	1200	1000	300	1	4,400
Local Model: NODDI	1100		1200			1	2,300
Fiber Tractography	1100	800	1200	1000	300	1	4,400
						Data sets:	**113,000**

Figure 21: Structural & functional MRI raw imaging data (INM-1 & INM-7, Jülich).

15.2 Diagrams

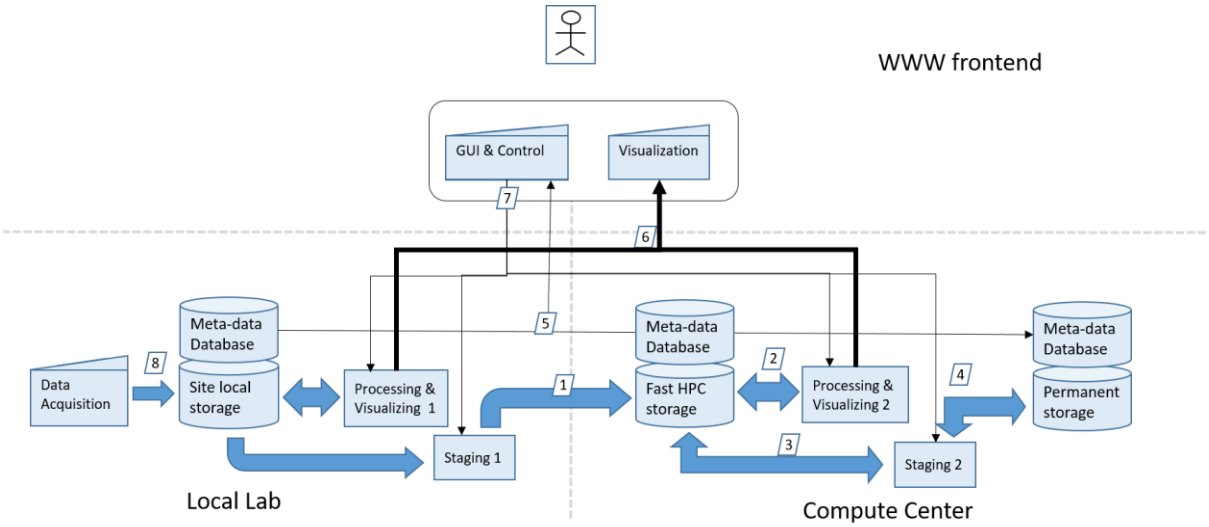


Figure 22: High-level abstract overview of local lab to compute centre processing and data pipeline.

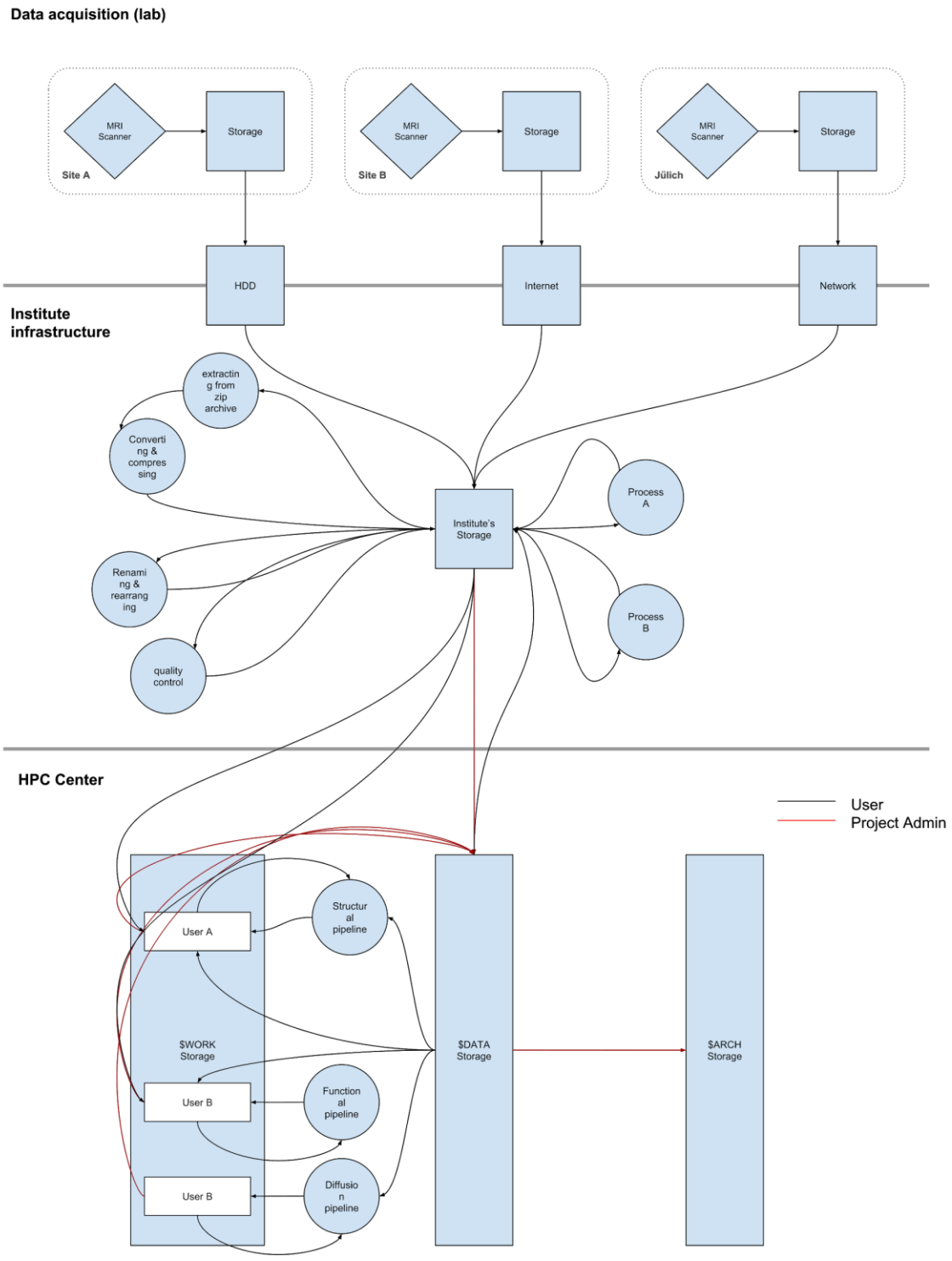


Figure 23: Current macroscale data handling workflow (INM-1 & INM-7, Jülich).

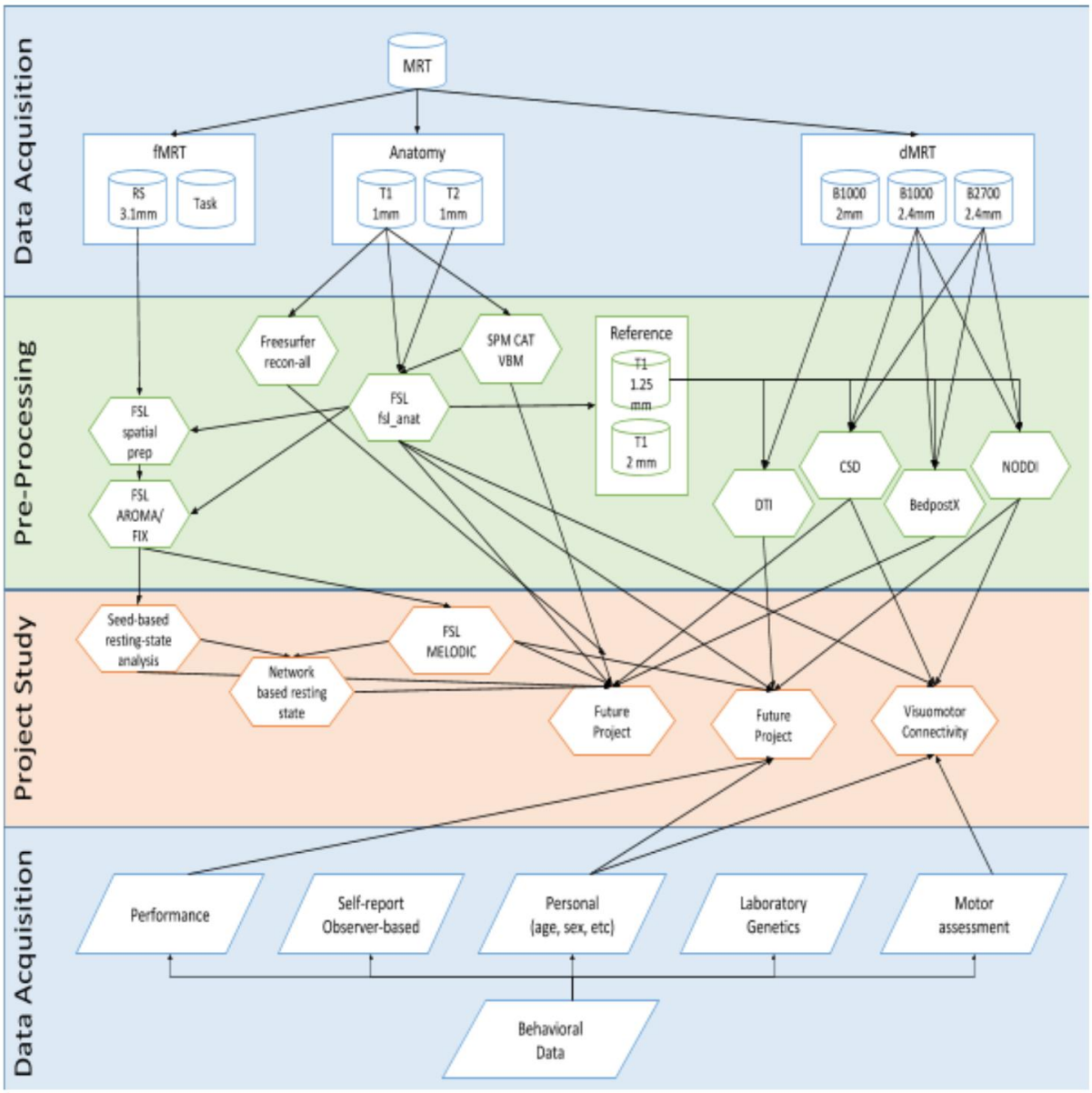


Figure 24: Macroscale data processing scheme (INM-1 & INM-7, Jülich).

Module	# data sets per node	total # data sets	Disk Space [GB]	Number of files per subject	Sum Number of files
Preprocessing		**327.000**	**245.696,39**		**48.371.200**
Brain Structure: freesurfer 5.3.0	24	800	252,34	479	383.200
Brain Structure: freesurfer 6.0.0	24	30.800	9.023,44	306	9.424.800
Brain Structure: ANTs	48	99.600	24.316,41	4	398.400
Brain Structure: LGI	24	49.800	14.589,84	4	199.200
Brain Structure: FSL	48	47.600	18.454,30	200	9.520.000
Brain Structure: CAT12	24	35.600	17.382,81	51	1.815.600
Functional Connectivity: FSL	48	47.700	40.759,28	400	19.080.000
Structural Connectivity: Preprocessing	15	15.100	120.917,97	500	7.550.000
Group Analysis			**240,00**		
Brain Morphometry			50,00		
Network-based analysis			75,00		
Structural Connectivity			100,00		
Spectral Reordering			5,00		
Matrix Factorization			10,00		
Neuro Imaging Inspired Simulation			**0,00**		
Brain Folding			0,00		
Modelling		**53.800**	**214.245,36**		**1.698.050**
Local Model: Tensor	48	13.500	1.318,36	15	202.500
Local Model: CSD	14	13.050	2.548,83	10	130.500
Local Model: BedpostX (GPU)	1	13.050	22.939,45	50	652.500
Local Model: NODDI (MATLAB)	1	0	0,00	0	0
Local Model: NODDI (MDT)	1	10.750	451,42	65	698.750
Fiber Tractography	1	3.450	186.987,30	4	13.800
Total		**380.800**	**460.181,75**		**50.069.250**

Figure 25: Resource usage as expected for the different cohorts studies.

15.3 Node Characterization

15.3.1 Data objects

[to be added]

15.3.2 Data repository

F. Hoffstaedter (Juelich) has additional information on data management.

Data repository: Fast HPC storage and data-base	
Base information	Classification of the data objects (see below): NA
	Access control requirements:

	Embargo of data possible
	Access requirements: Access data for remote visualization
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: Todays $DATA Fast random access: 10 of TBs High bandwidth sequential: 10 – 100 of TB 1GB per node Huge amounts of single files (eg. NIfTI) cause problems with inode quotas For Big Brain 3; one optical section (the centre one) should stay on fast storage for frequent visual inspection. We expect ~50% additional data per section (downscaled, feature attributes, etc.). 1µm z-scans: 1TB/day * 8 scanners / 30 * 1.5 = 0.4TB daily data growth that stays on disk permanently Cell segmentation, if applied in streaming mode to all incoming data, is expected to produce another 50% on top ~ = 0.15TB per day We expect to process 5-10 ROIs per year. Per ROI, we need ~ 5TB fast storage (campaign, for the time of the project, e.g. several months)
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: Permanent/Long term storage and meta-data base	
Base information	Classification of the data objects (see below): hdf5 compressed files, ~80%; will archive packages from multiple files in the order 1-2 TB
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories

	where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

15.3.3 Processing stations

The nature of the use case results in a large amount of processing steps. They cannot be detailed individually. The best way forward is a general description of the work to be done and one or two detailed breakdown of processing station, once with high resource requirements. Which there are is up to the use case HPC expert / scientist.
This section is highly task specific and should be filled in by the domain expert.

Processing station: Staging	
Base information	General description of data processing: Transfer to the central HPC sites needs to be controlled with a specific software.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: Grid FTP?
	URL to additional information: NA
	Limitation: NA

15.4 Discussion

Felix Hoffstaedter: The problem with the exact description of the files and their size is that there are too many cases to be put in these tables. Jan and me did a rough estimation of the files and their size for the last compute time proposal which I put in the document now. To illustrate the problem, have a look at the following table in relation to the processing work-flow. Each package needs mostly (not exclusively) Niftis

and writes Niftis and other files. In my view, it makes no sense to fill in the tables for our work flow. The relevant numbers are Numbers of files and Diskspace needed for the full processing of all data sets. Of note, these are only numbers for the data we have available at the moment and more will be released soon.

16. Multi-area macaque NEST simulation with life visualization and interaction (#16)

Multi-area macaque NEST simulation with live visualization and interaction

Use Case Description and Specification

21-06-2018 Wouter Klijn, Markus Diesmann, Sacha van Albada

Partners	Markus Diesmann
Institutions	INM6 FZJ
Principal Investigators	Sacha van Albada

Date	Version / Change
21-06-2018	(Wouter Klijn) Initial write down
29-06-2018	(Wouter Klijn) Add summary information from Sacha
20-08-2018	(Anne Carstensen) Editorial changes
01-09-2018	(Wouter Klijn) review and questions for specific information added
11-09-2018	(Sacha van Albada) Review and update on questions for specific information
19-09-2018	(Wouter Klijn) Review of new information and further questions for specific information added
20-09-2018	(Sacha van Albada) Update on further questions for specific information
25-09-2018	(Anne Carstensen) Integration of review comments and updates

16.1 Use Case Description

KR4.7 Release of multi-layer point-neuron network model of all vision-related areas of macaque cortex, improved using new connectivity and activity data. Account at cellular resolution for properties essential for cortical function, focusing on excitability and feedforward-feedback interactions.

Construct multi-layered multi-area models of the cortex relating the local microscopic connectivity to the macroscopic connectivity of the brain. On the local level, this leads to models with a higher degree of self-consistency than previously possible, because the origins of synapses from remote sources are included, and the lower parts of the power spectrum of neuronal activity missing in purely local models can be investigated. On the global level, the bottom-up and top-down flow of activity between cortical areas in these hierarchical models are investigated. The models include millions of spiking neurons and tens of billions of synapses. Anatomical data from various sources are combined

with statistical data prediction strategies to define the population sizes and the population-specific connection probabilities. New data are integrated into the model as they become available. Both the statistics of the spiking activity (firing rates, synchrony, regularity) and the inter-area functional connectivity are compared with experimental data (spike-sorted extracellular recordings, fMRI). Discrepancies between the simulated and experimental data are used to improve the model, in part using mean-field reductions of the spiking models. The dependence of the network activity on parameters including the external drive and the synaptic strengths is investigated.

16.1.1 ICEI Co-design workshop cases:

1. Online monitoring: pathologic network states are often detectable in first few seconds
2. Interactive experimentation: re-using the network structure for multiple experiments
3. Interactive data selection: activity data cannot be recorded for all elements indefinitely

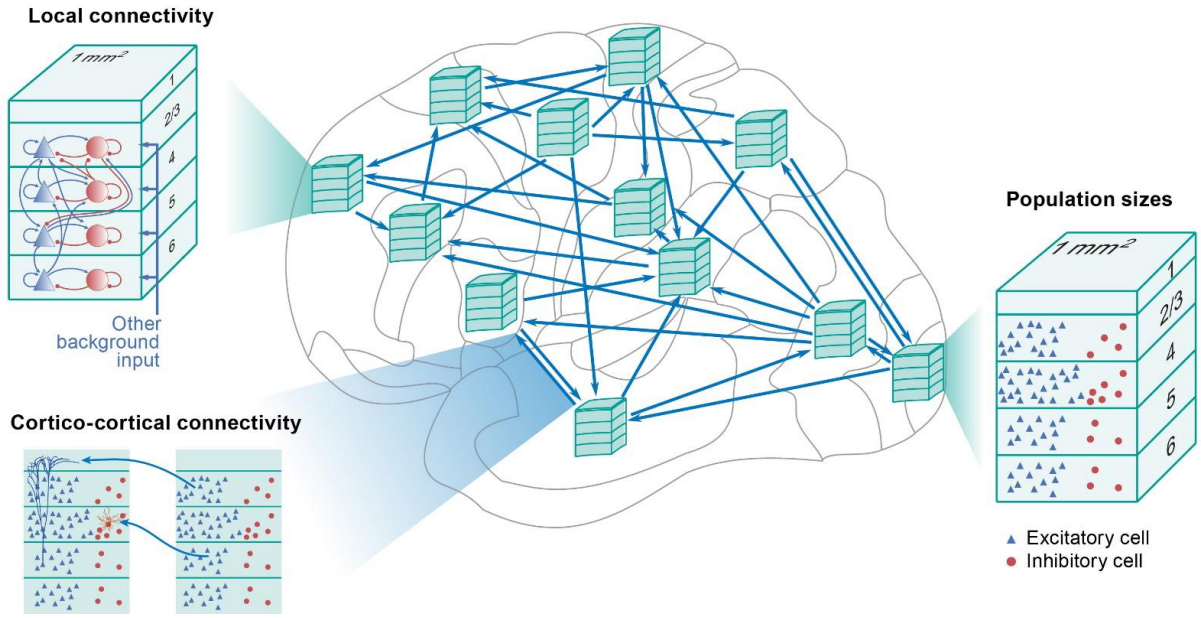
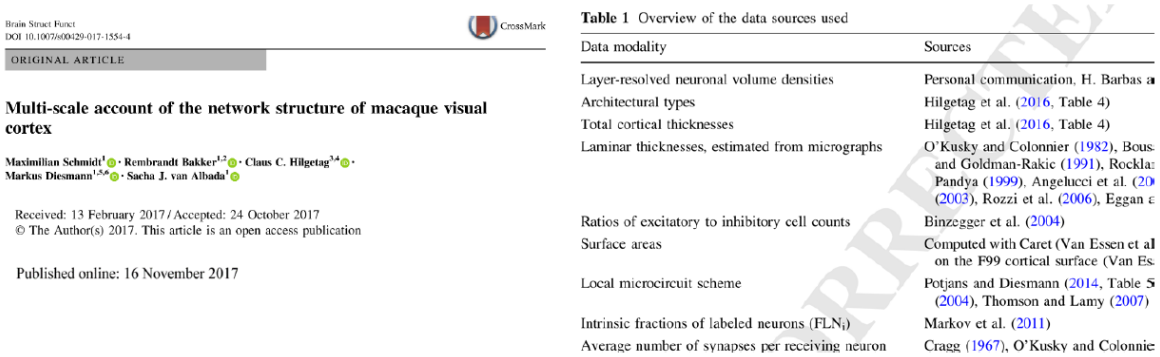


Figure 26: Overview of the multi area model of macaque visual cortex.



Brain Struct Funct
DOI 10.1007/s00429-017-1554-4

CrossMark

ORIGINAL ARTICLE

Multi-scale account of the network structure of macaque visual cortex

Maximilian Schmidt[1] · Rembrandt Bakker[1,2] · Claus C. Hilgetag[3,4] · Markus Diesmann[1,5,6] · Sacha J. van Albada[1]

Received: 13 February 2017 / Accepted: 24 October 2017
© The Author(s) 2017. This article is an open access publication

Published online: 16 November 2017

Table 1 Overview of the data sources used

Data modality	Sources
Layer-resolved neuronal volume densities	Personal communication, H. Barbas a
Architectural types	Hilgetag et al. (2016, Table 4)
Total cortical thicknesses	Hilgetag et al. (2016, Table 4)
Laminar thicknesses, estimated from micrographs	O'Kusky and Colonnier (1982), Bous and Goldman-Rakic (1991), Rockla Pandya (1999), Angelucci et al. (20 (2003), Rozzi et al. (2006), Eggan a
Ratios of excitatory to inhibitory cell counts	Binzegger et al. (2004)
Surface areas	Computed with Caret (Van Essen et al on the F99 cortical surface (Van Es
Local microcircuit scheme	Potjans and Diesmann (2014, Table S (2004), Thomson and Lamy (2007)
Intrinsic fractions of labeled neurons (FLN_i)	Markov et al. (2011)
Average number of synapses per receiving neuron	Cragg (1967), O'Kusky and Colonnie

Figure 27: Data sources entering into the macaque mulit-area model.

16.2 Diagrams

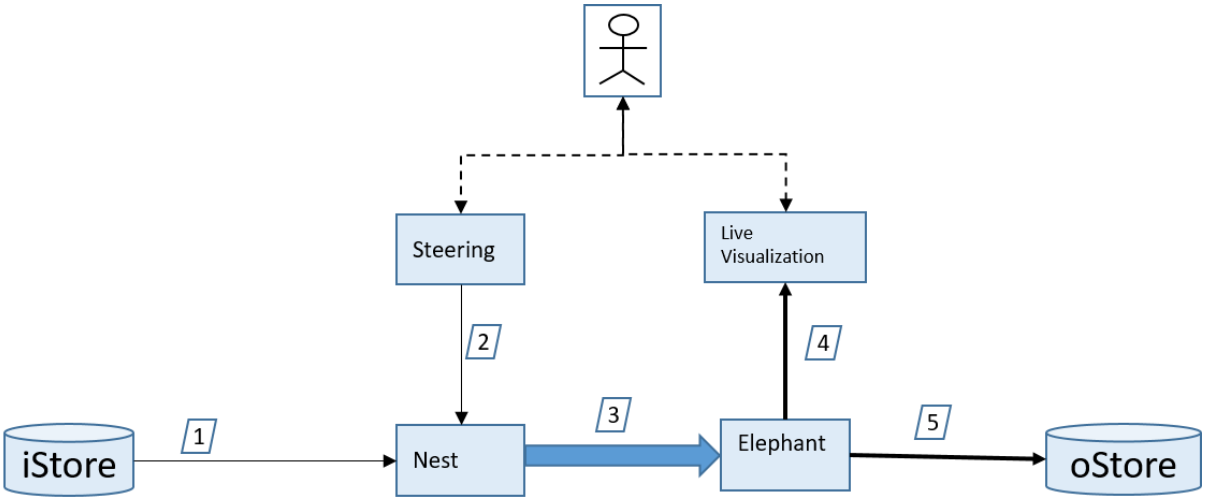


Figure 28: A simplified system breakdown for the macaque cortical multi-area modelling project.

16.3 Node Characterization

16.3.1 Data objects

Data object: ***1***, Neuronal models	
Base information	General description of what data is stored • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***2***, Steering messages	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***3***, Simulation output to online analysis	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***4***, in situ visualization	
Base information	General description of what data is stored: • Formats • Metadata • Database requirements
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***5***, Analysis and simulation output to long term storage	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

16.3.2 Data transport

Data transport: 1, Model info from storage to node	
Base	General description of what data is transported:

information	We don't use data from the NIP. Also in the coming five years, I don't foresee that we will use a large amount (or perhaps even any) data from the NIP.
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 2, GUI messages.	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 3, Big data transport between simulation and analysis	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport:4, Online visualization data stream	
Base information	General description of what data is transported: Can probably be copied from external solution
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 5, Experiment results for long term storage	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

16.3.3 Data ingest / GUI

Data ingest: Istore	
Base information	Description of input data source: CoCoMac, axonal tracing data from the lab of Henry Kennedy, neuron densities and laminar thicknesses from the lab of Helen Barbas (in future potentially from Neuroinformatics Platform), further parameters extracted from the literature. These data are small.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Steering	
Base information	Description of input data source: GUI sending control commands for the underlying systems. (I suspect this is an external solution!)
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current solution	Name: HBP in situ pipeline (Aachen) might be an acceptable solution.
	URL to additional information: NA
	Limitation: NA

16.3.4 Data repository

Data repository: oStore	
Base information	Classification of the data objects (see below): Up to 1 TB of raw output data per simulation.
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

16.3.5 Processing stations

Processing station: NEST	
Base information	General description of data processing: 2 GB-2 TB memory per simulation. About 100-200 experiments per year. About 1,000,000 node*h each year. Two new projects: 1) A study of visuomotor interactions in macaque, which will entail increasing the number of areas, resulting in an approximately 1.2x increase in number of neurons; 2) a study of V1, V2, and V4 including spatial convergences and divergence resulting in a 3x increase in number of neurons, and roughly 10x in number of synapses.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA

	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Elephant	
Base information	General description of data processing: What analysis do you expect to do? Calculate firing rates of each area as a function of time and population-specific average firing rates. Calculate a measure of irregularity (e.g. LV) and synchrony (e.g. average pairwise correlations) for each population. Calculate firing rate distributions and population spectra for selected areas. Calculate area-level functional connectivity (zero-lag correlations) between synaptic inputs. Calculate area-level correlation functions between smoothed PSTHs. Calculate LFP spectra from hybridLFPy predictions. Compute measures of directed interactions such as Granger causality, on the area and population levels.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

16.4 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	Questions to address
Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? • What software stacks need to be available? • Is it possible to define memory capacity requirements? It is not strictly necessary, but it could be nice to have interactive visualization of the simulation output using for instance VisNEST, along with the possibility of changing parameters such as the strength of the external drive or synaptic weights. Such an interactive session could take from a few minutes to a few hours. The data transferred could be on the order of 5 MB/s.
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? none
Virtual Machine Services	• Which parts of the workflow require such services? none
Active Data Repositories	• Which parts of the workflow require such services? none
Archival Data Repositories	• Which parts of the workflow require such services? Simulation data output storage (mostly pre-analysis) – so far stored on our local cluster but the data to be stored are likely to increase in future.
Data Mover Services	• Which parts of the workflow require such services? none
Data Transfer Services	• Which parts of the workflow require such services? none • Between which ICEI sites is data planned to be transferred? • How much data is expected to be transferred per time unit? NA • How are transfer patterns expected to change over time? NA
Data Location Service	• Which parts of the workflow require such services? none
Internal interconnect	• Are there known minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single

	site? no
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? no
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples: ○ Special accounts for running services ○ Needs for fine-granular control of access to data no
User Support Services	• Are the specific foreseeable needs for user support services? no

16.5 Use Case references

SCIENCE CASES FOR INTERACTIVE SUPERCOMPUTING SIMULATION IN COMPUTATIONAL NEUROSCIENCE 09.02.2018 Fenix/ICEI Co-Design Workshop/ ETH Zürich
MARKUS DIESMANN

HBP SGA2 Grant agreement

17. Towards a novel decoder of brain cytoarchitecture using large scale simulations (#9)

Towards a novel decoder of brain cytoarchitecture using large scale simulations

(of realistic white matter tissue samples, of the water diffusion process within tissues, and of the diffusion MRI signature)

Use Case Description and Specification

27-Jun-18 Cyril Poupon, Kévin Ginsburger

Partners	Neurospin / TGCC / INM1
Institutions	CEA DRF JOLIOT / CEA DAM / Forschungszentrum Jülich (FZJ)
Principal Investigators	C. Poupon / J.-C. Lafoucrière & C. Menaché / M. Axer

Date	Version / Change
2018, June 24th	Initial version
06-09-2018	(Wouter Klijn) Recreate template from deliverable
02-10-2018	(Cyril Poupon / Thomas Leibovici) Update of use case information
04-10-2018	(Anne Carstensen) Integration of updates

17.1 Use Case Description

Mapping and understanding the cytoarchitectonics of the human brain is a challenge that started back at the beginning of the 20th century with famous neuroanatomists who segregated the cortex of a post-mortem human brain sample into a few dozens of areas from the observation of the laminar structure of the cortical ribbon and of its cellular organization using optical microscopy. The most famous atlas was developed in 1905 by Korbinian Brodmann and remains today widely used by neuroscientists even if it suffers from several biases: first, because it was developed from a single sample, it cannot capture the inter-subject variability of the cytoarchitecture maps; second, boundaries of the areas have been drawn from visual observations, and may not reflect the real boundaries of functional areas. The community is fighting to go beyond Brodmann areas and during the last decade, several teams attempted new strategies to map the brain cytoarchitectonics. On the one hand, the Institute of Neuroscience for Medicine (INM1, Juelich Forschungszentrum, headed by Prof. K. Amunts) has launched a decade ago a large project aiming at developing a new approach to establish a novel cytoarchitectural atlas, called the Big Brain, based on the mapping of the receptor neurotransmitters. This project has become a core development in the Human Brain Project and will provide a unique mapping of functional areas of a few post-mortem

samples. On the other hand, several teams have tried to establish maps from the acquisition of large cohorts of in vivo human healthy volunteers. For instance, the team of Mangin et al has investigated the structural connectivity of the cortex inferred from diffusion MRI as a potential information to further segregate Brodmann areas and propose a new parcellation of the cortical surface (Lefranc et al 2017). More recently, the team of Glasser & al published a novel atlas including more than two hundreds of cortical areas established from the individual relaxometric and functional MRI scans acquired on the Human Connectome Project cohort (Glasser et al 2017).

The next challenge is now to develop methods to segregate the human brain cytoarchitecture in vivo and at the individual scale. The success of such a challenge relies on the capability of modern neuroimaging methods to probe the variations of the cellular organization of brain tissues in vivo. Quantitative and diffusion MRI are known to be sensitive to the myelo- and cyto-architecture of tissues and might be good candidates to perform virtual biopsy in vivo. Quantitative MRI has been successfully used during the last decade to map the myelin water fraction using T1-weighted and T2 weighted MRI scans. Similarly, diffusion MRI has proven its potential to probe not only the structural connectivity of the human brain through the observation of the anisotropy of the random displacement of water molecules in brain tissues, but also some quantitative microstructural features characterizing their cellular organization such as the axon density or the axon diameter. Unfortunately, the cellular organization of brain tissues (gray and white matter) can be extremely complex, and today, few is known about the diffusion MRI signature of the plethora of possible cellular environments met in the brain. Diffusion MRI scans require the tuning of several sequence parameters that obviously impact the nature of the diffusion contrasts obtained at the end and few is known about the parsimony of the resulting parameter space with respect to this contrast. Investigating this question is essential to establish the reduced set of parameters to be used in vivo to preserve a reasonable scan time and still be able to collect enough diffusion MRI data to segregate the cytoarchitectural areas both in gray and white matter. Obviously, one cannot achieve an exhaustive scanning of the sequence parameter space in vivo.

This Use Case project aims at replacing in vivo diffusion MRI scans by in silico diffusion MRI scans enabling to reach a much higher level of completeness of the parameter space sampling. To do so, the Use Case will first focus on white matter (WM) cytoarchitecture being simpler than gray matter (from a cytoarchitectural point of view) and will require to:

Task #1 - create an exhaustive bunch of in silico realistic white matter virtual tissue samples by numerical simulations of cellular membrane geometries,

Task #2 - simulate the diffusion process of water molecules in every realistic in silico WM tissue sample using a Monte-Carlo approach,

Task #3 - simulate the diffusion MRI signature of every WM tissue sample for an exhaustive set of diffusion MRI sequence parameters achievable on actual preclinical and clinical MRI systems,

Task #4 - learn a deep neural network to build a decoder/regressor of the WM microstructure

Task #5 - use the decoder to establish an atlas of the WM microstructure

Dedicated software tools have already been developed by the teams of Cyril Poupon and Markus Axer in the frame of HBP SGA1 for the 3 first tasks (see Figure 29), and simulations have been started on the JURECA HPC facility at JUELICH. However, there is a clear need to extend simulations to the TGCC facility in order to run the plethora of needed simulations and obtain results within the frame of SGA2 for white matter tissues.

If successful, the extension to gray matter (cortex and deep nuclei) will be straightforward but will need even more computational resources, due to the higher level of complexity of the cellular environment in gray matter.

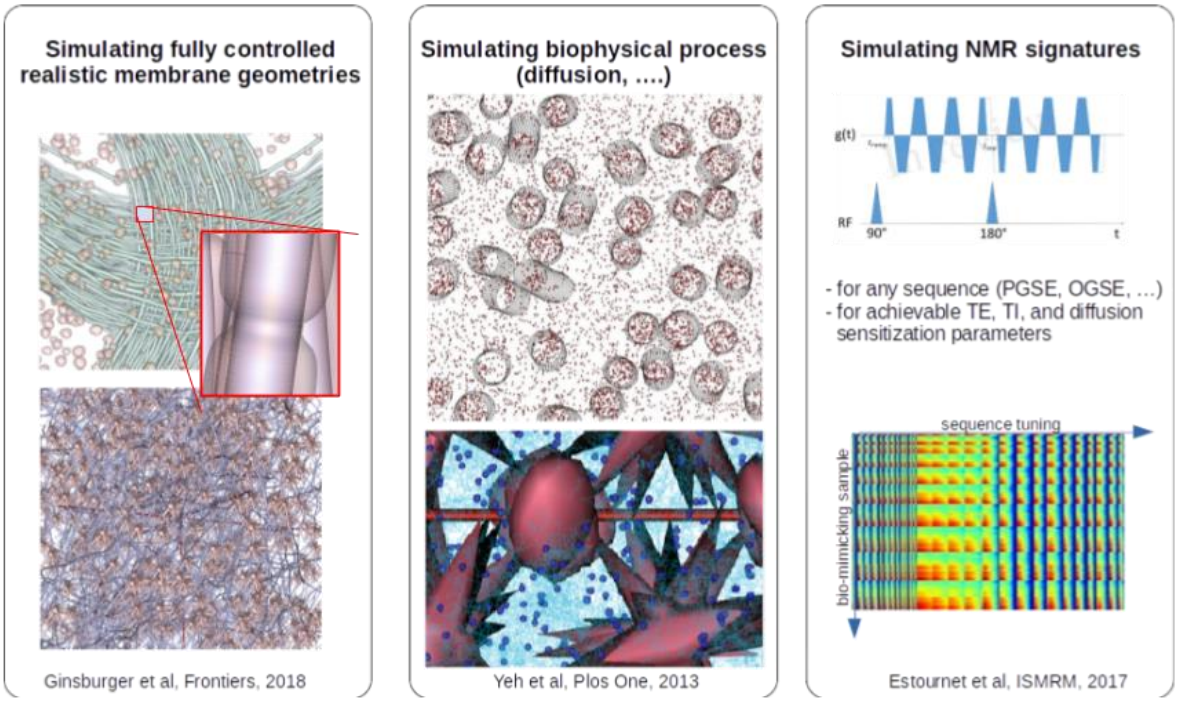


Figure 29: Example of virtual tissues (left), Monte-Carlo simulation of the diffusion process of water in virtual tissues (middle), and simulation of the diffusion MRI signal for various pulse sequences and tunings (right) obtained with the in-house developed software.

17.1.1 Annotated Use Case Diagrams

The Diagram 1 provides a flowchart of the task #1 consisting of simulating a dictionary of virtual white matter tissues. Each virtual tissue is designed from a set of geometrical parameters including:

- the number of white matter fibre populations (from 1 to 3)
- the properties of each fibre population including its volume fraction, the main direction of the population, the dispersion and tortuosity of its fibres, the statistics of the axon diameter, the statistics of the g-ratio characterizing the myelin sheath, myelin g-ratio, the statistics of the Ranvier nodes, the permeability of axons
- the properties of the glial cell population including their mean diameter, the statistics of the number of branches per cell, and the statistics of the diameter of these branches.

A graphical user interface will be developed to facilitate the prescription of tissue parameters, and a 3D viewer will be developed to visualize 3D renderings of virtual tissues.
A satisfactory sampling of the parameter space indicates that around 10^4 tissue samples have to be simulated for the class of tissue including a single fibre population, 10^7 tissue samples for the class including two fibre populations, and 10^{10} tissue samples for the class of three fibre populations.

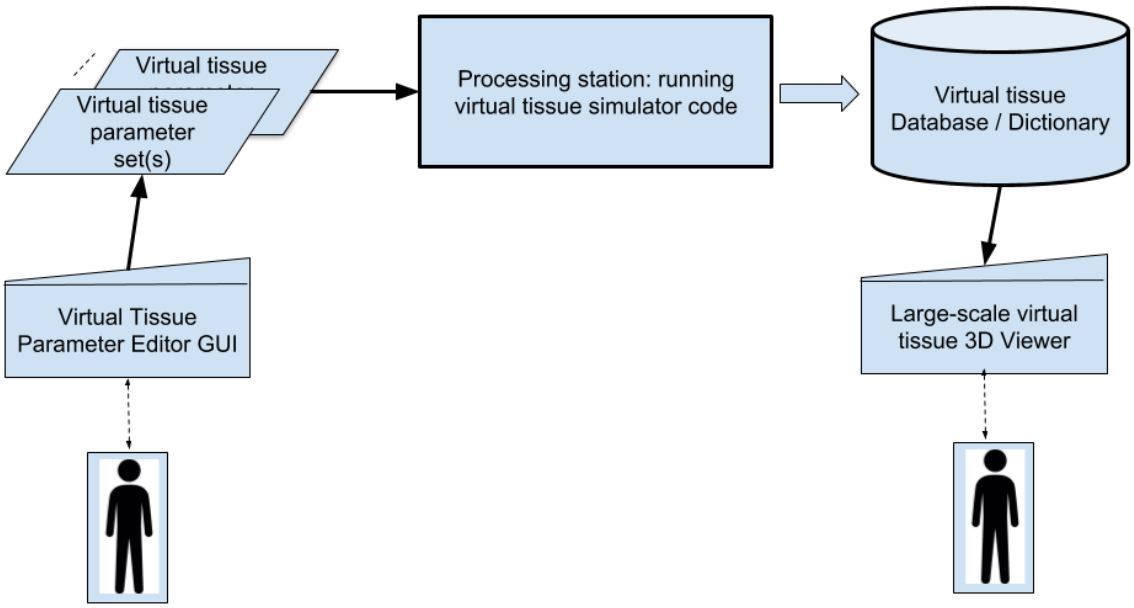


Figure 30: Flowchart of the construction of a large dictionary of virtual brain white matter tissues.

Task #1 storage capacity requirements:

Task #1 consists in generating realistic geometries of white matter: each generated voxel of size 100x100x100 µm will contain from 500 to 10 000 axons (depending on the mean diameter of axons and on the packing density).
Each axon is represented as a set of spheres, which is the basic unit of all our algorithms.
We estimate an upper bound of 500MB for each geometry to store the position (3 float values) and the radius of all spheres within a voxel.

→upper bound of 500 MB to store one generated geometry (with a mean of 200 MB)

Task #1 computing capacity requirements:

Our geometry generation algorithm can be decomposed in 2 steps:
- creation of overlapping axons in the voxel according to the required biophysical parameters (diameter, packing density, angular dispersion…)
- solving the overlapping between axons using the decomposition of axons into spheres and applying repulsion forces between overlapping spheres
We estimate an upper bound of 30 minutes to generate a given geometry.
However, this upper bound has been obtained on a Tesla K40 GPU and corresponds to the worst possible case: very small mean axons diameter (0.1µm) and high packing density (0.8) for which the number of spheres is maximal.
In most cases (diameters > 0.5µm and volume fraction inferior to 0.7), the geometry generation will take less than a minute.

→upper bound of 30 minutes to generate a geometry (with a mean of 5 minutes)

The Diagram 2 provides the flowchart of task #2 required to establish a huge database of Monte-Carlo simulations of the diffusion process of water molecules within each white matter virtual tissue belonging to the Virtual Tissue dictionary. Biophysical parameters characterizing the diffusion process in brain tissues have to be fed into the Monte-Carlo simulator as well as the individual virtual tissue sample. Trajectories followed by random walkers are then stored for each tissue sample. The number of random walkers has to be tuned with respect to the complexity of the geometry of cell membranes populating every virtual tissue, typically on the order of 10^5 particles. Temporal constraints are imposed by the specifications of the simulated diffusion MRI sequence (echo time and temporal resolution of gradient waveforms). To cover a large scope of NMR experiments, we propose to simulate a total duration of 300 milliseconds with a temporal resolution of 10 microseconds.

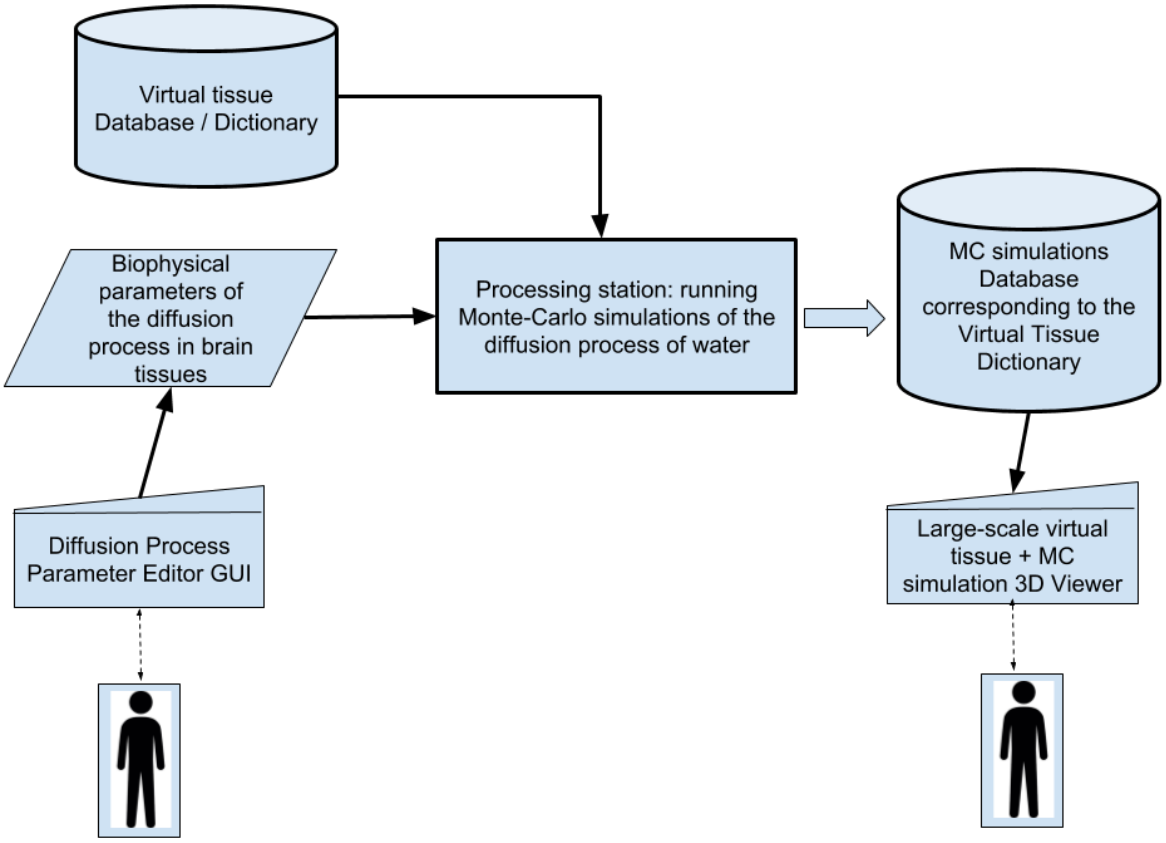


Figure 31: Large-scale Monte-Carlo simulations of the Brownian motion of water molecule corresponding to the Virtual Tissue dictionary established in Figure 30.

Task #2 storage capacity requirements:

Task #2 consists in performing a Monte-Carlo simulation of the Brownian motion of 2.10^5 random walkers during 300 ms with a time step of 10 µs for each geometry generated in Task #1.
The trajectories of all the random walkers have to be stored, leading to a size of 97 GB per simulation.

→97 GB to store the trajectories of random walkers for one geometry

Task #2 computing capacity requirements:

We estimated a runtime of 1h30 on a Tesla K40 GPU to perform the simulation for one voxel. However, the CUDA code has not yet been fully optimized for an optimal usage of GPU capabilities.

→1h30 to perform 1 simulation on Tesla K40 without CUDA optimization

The Diagram 3 provides the flowchart of task #3 required to establish the huge dictionary of (virtual tissues / diffusion MRI signature) required to learn the decoder mentioned in Task #4. The diffusion MRI signature will consist of a few thousands of simulated NMR contrasts corresponding to Pulsed Gradient Spin Echo (PGSE) and Cosine Trapezoidal Oscillating Gradient Spin Echo (CT-OGSE) sequences achievable on an actual clinical 3T MRI system. Each of these sequence offers the possibility to tune parameters impacting the diffusion sensitization such as the diffusion gradient magnitude, the diffusion direction, the diffusion pulse width and separation for the PGSE sequence or the diffusion pulse frequency and number of lobes for the CT-OGSE sequence. The simulated signal will strongly depend on these parameters, and we have chosen to sample the parameter space densely (100 times more than in real acquisitions) in order to better capture the parsimony of the diffusion MRI signal in this space (known as the q-space).

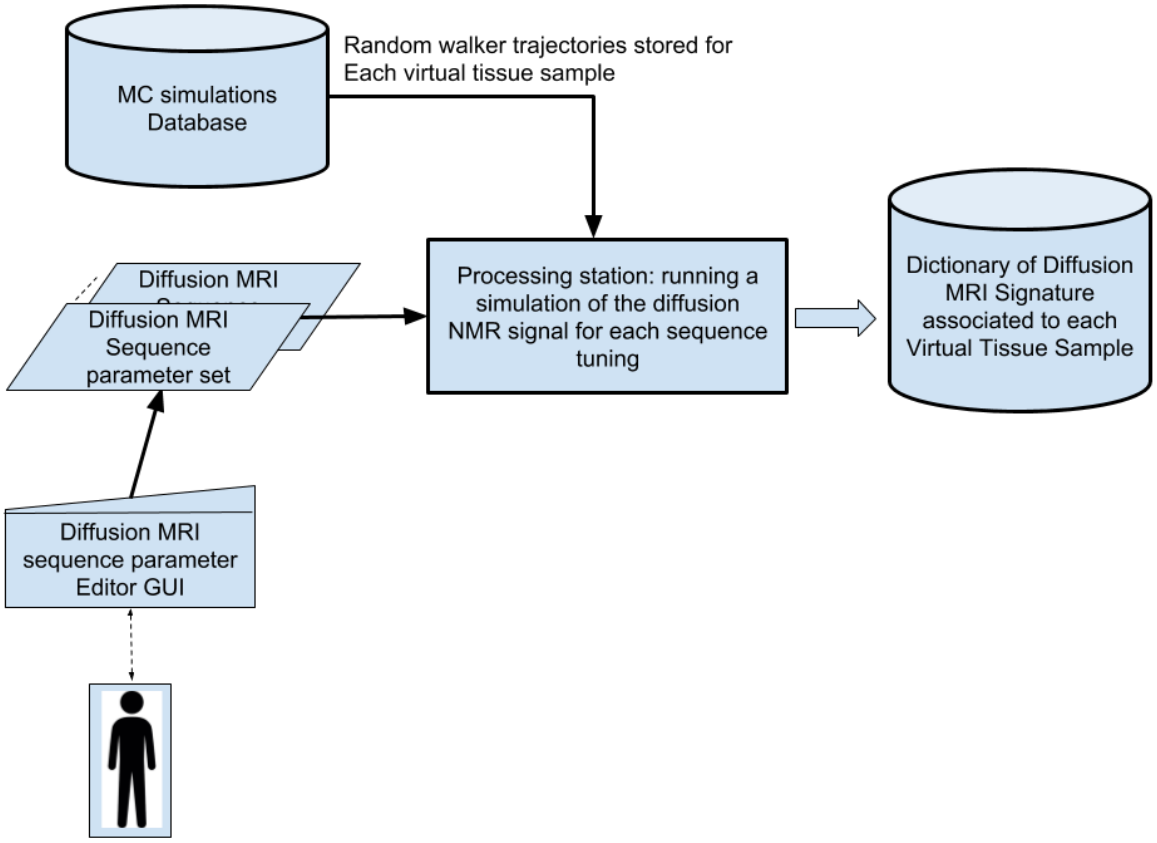


Figure 32: Large-scale simulations of the diffusion-weighted MRI signal over a large set of sequence tunings from the Monte-Carlo simulations obtained in Diagram 2 in order to establish a dictionary of (virtual tissues / diffusion MRI signatures).

Task #3 storage capacity requirements:

Task #3 consists in synthesizing the NMR signatures of each generated geometry by applying diffusion sequences with varying parameters on the previously generated random walkers trajectories (see Task #2).

We estimated that 3000 NMR signatures have to be computed for each geometry to fully explore the parameter space. Each geometry has a size of 10KB, thus leading to a total of 30MB for each generated geometry.

→30 MB to store all NMR signatures for one geometry

Task #3 computing capacity requirements:

The computation of NMR signatures can be easily parallelized on GPU, leading to an estimated runtime of ~1.2s per signature, and thus a total runtime of 1h for each geometry.

→1h to generate all NMR signatures for one geometry

The diagram 4 depicts how the former dictionary of virtual white matter tissue samples / diffusion MRI signatures enables to train a machine learning tool, like a deep neural network, in order to create a decoding tool able to recognize / extrapolate the set of quantitative features characterizing the cytoarchitecture at each voxel of the brain, from a real and individual set of diffusion MRI scans, corresponding to various sequences and sequence settings. The input database used to train the DNN is composed of around ~10^{10} entries resulting from the previous large-scale simulations.

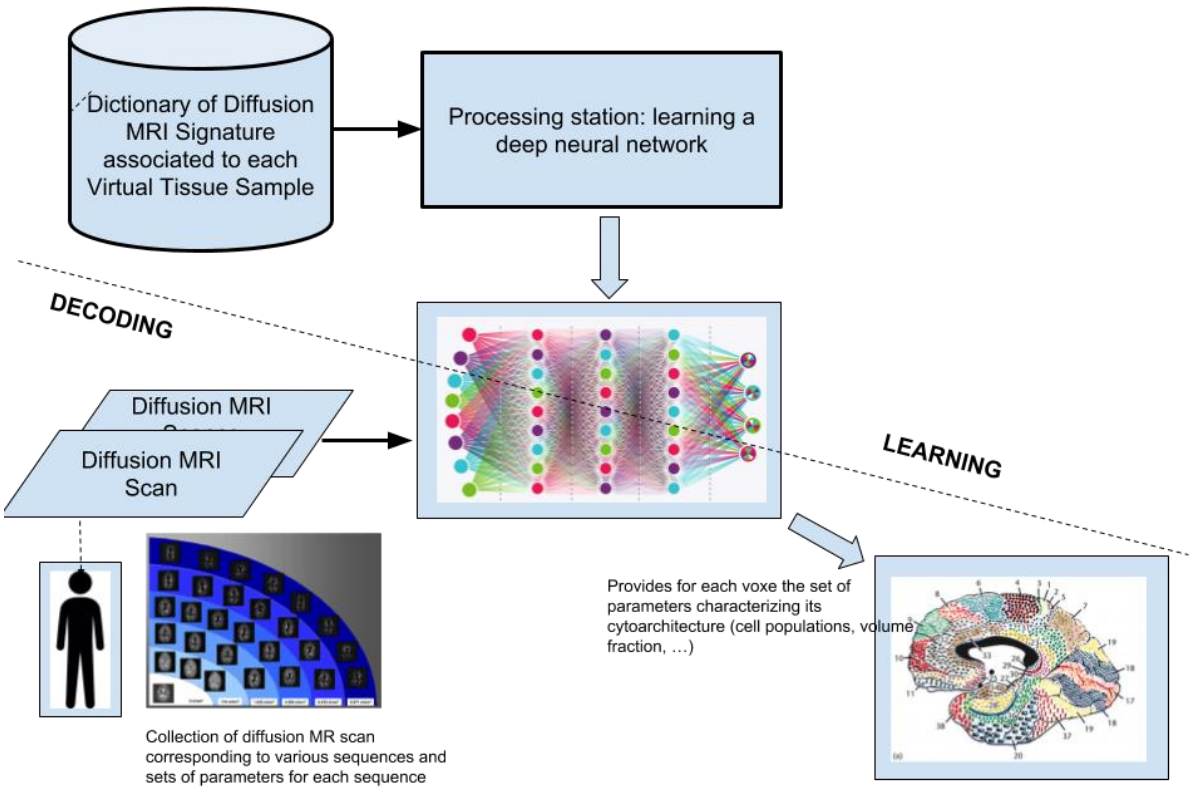


Figure 33: Use of the simulated "virtual sample/diffusion MRI signature" dictionary to training a deep neural network and use of the trained network to decode the brain cytoarchitecture of individuals in vivo.

Task #4 storage capacity requirements:

→no additional storage capacity

Task #4 computing capacity requirements:

The last step of this project is to train a neural network with all the generated NMR signatures. For each signature, we know the parameters of the employed diffusion sequence and the biophysical parameters of the generated geometry.
The aim of the training is that, when a real NMR signature is fed to the network with known diffusion sequence parameters, the network will be able to estimate the underlying biophysical parameters.

→need for a "Deep-Learning oriented" GPU to train a neural network with all previously simulated data (NMR signatures)

We now sum up the required storage and computing capacities needed for the whole project (Tasks #1, #2 and #3).

Project storage capacity requirements:

From what was estimated before in the different tasks, approximately 100 GB are needed for each generated geometry to store all the information (geometry+trajectories+NMR signatures).
However, if the files are compressed in .zip format, the total size can be reduced by a factor 1.75.
As was emphasized in the introduction, we estimated that 10^4 geometries have to be simulated to cover all possible geometries with 1 fibre population inside the voxel, 10^7 geometries have to be simulated to cover all possible geometries with 2 fibre populations inside the voxel, and 10^{10} geometries have to be simulated to cover all possible geometries with 3 fibre populations inside the voxel.

We will thus give the estimated storage for 1, 2 and 3 fibre populations.

- 1 fibre population: 10^4 geometries →1200 TB
- 2 fibre populations: 10^7 geometries →1.2e6 TB (pessimistic upper bound)
- 3 fibre populations: 10^{10} geometries →1.2e9 TB (pessimistic upper bound)

The announced storage capacity for 2 fibre populations (and a fortiori, for 3 fibre populations), is very high. However, starting the project with 1 fibre population will enable us to better understand the parsimony of the parameter space: we will probably observe very small differences between NMR signatures from 2 distinct generated geometries with close biophysical parameters, which will not be distinguishable in the presence of noise in the NMR signature.

These observations will enable us to reduce the number of generated geometries for 2 and 3 fibre populations.

Also, our geometry generation algorithm might not be able to construct all the geometries that were counted here. In particular, geometries with high packing densities, high angular dispersion and 2 or 3 fibre populations might not be physically achievable, thus reducing again the final number of simulated geometries.

Finally, in the case where the storage size would still remain too big, it is possible to keep only the NMR signatures (only 30MB per geometry) and not store all the trajectories, though this would be a major concession to the project since we would not be able to create any other NMR signature without performing again the Monte-Carlo simulation. This could be a problem if new diffusion sequences are proposed in the future and we want to compute their associated NMR signatures.

Project computing capacity requirements:

From what was estimated before in the different tasks, approximately 3 hours are needed for each generated geometry to perform all computations on a Tesla K40 GPU (geometry generation+Monte-Carlo simulation+NMR signature synthesis).
However, we divide this time by 2 since more recent GPU will be used.

We give the estimated GPU computing hours for 1, 2 and 3 fibre populations.

- 1 fibre population: 10^4 geometries $\rightarrow 1.5 \times 10^4$ hours
- 2 fibre populations: 10^7 geometries $\rightarrow 1.5 \times 10^7$ hours (pessimistic upper bound)
- 3 fibre populations: 10^{10} geometries $\rightarrow 1.5 \times 10^{10}$ hours (pessimistic upper bound)

10 modern GPUs will be sufficient to perform the 1 fibre population simulations in 2 months.
However, if we aim at performing the 2 fibre population simulations within 1 year, we would need one thousand GPUs, which seems unrealistic.

We already explained in the previous section that there will probably not be as much geometries to simulate as estimated here due to the parsimony of the parameter space which has to be determined with the 1 fibre population study and geometrically unachievable configurations for 2 and 3 fibre populations.

We want to add here that we expect to reduce the total runtime for each generated geometry to 2 hours (instead of 3) on a Tesla K40 after a thorough optimization of the CUDA code, especially on the Monte-Carlo simulation part. We might also have underestimated the performance gain from the Tesla K40 GPU to a modern GPU (we applied a x2 speed-up factor).

If we account for 2 hours of total runtime per geometry on a Tesla K40 and a x4 speed-up factor, we obtain a runtime of 30 minutes per geometry.
We could then perform the simulations for 2 fibre populations in ~1 year with 500 GPUs.
It is also possible to extend the simulation time (up to 2 years), thus reducing the number of needed GPUs accordingly.

17.2 Node Characterization

- Compute nodes with GPU at least 256 GB of memory (512 GB would be better).
- Throughput to storage: at least 1.5GB/s (100GB in 1 min)

17.3 Platform needs

3 levels of requirements:

- Stage 1: single fiber population: $1.5x10^4$ hours of computation on GPU, 1200 TB of storage
- Stage 2: two fiber populations: $1.5x10^7$ hours of computation on GPU, 1.2e6 TB of storage.
- Stage 3: three fiber populations: $1.5x10^{10}$ hours of computation on GPU, 1.2e9 TB of storage.

18. Blue Brain Project Microcolumn (#12)

Blue Brain Project Microcolumn

Use Case Description and Specification

<Date> Author Names,

Partners	Muller, Eilif Benjamin <eilif.mueller@epfl.ch>
	Courcol, Jean-Denis <jean-denis.courcol@epfl.ch>
Institutions	EPFL
Principal Investigators	Schürmann, Felix <felix.schuermann@epfl.ch>
	Markram, Henry

Date	Version / Change
20-06-2018	(Wouter Klijn)Initial version
26-06-2018	(Wouter Klijn) Copy in proposed use cases and email addresses of experts
20-08-2018	(Anne Carstensen) Editorial updates
01-09-2018	(Wouter Klijn) Create word template document, mark high priority information.
02-10-2018	(Wouter Klijn) Move technical detail to numbered tables. Create overview diagram with numbers. Add disclaimer

18.1 Use-case description and specification template

Disclaimer (Wouter Klijn). This version of the use case document is a restructuring of the information as received. Time constraints prevented review of this update template document by the domain expert. Both the quality of the information provided and priority of this case are high thus we include it in this state, with changes clearly marked.

Figure 38 has been added new. Newly added text are marked light grey including questions for clarification. Additionally, the numbered and named tables are based on the interpretation by WK.

18.2 Use Case Description

User: Sam – a scientist who wants to run a microcircuit simulation

Preconditions:

- Microcircuits are registered and referenced in the HBP Knowledge Graph.
- Microcircuits data are stored in the HBP Knowledge graph Object Storage (currently CSCS Object storage)
- The user has an HBP account and an account in the HPC Centre where he wants to run a simulation

Success scenario:

1- Sam selects a microcircuit from the ones referenced in the HBP Knowledge graph from a web GUI.
2- Sam selects a HPC Centre where he wants to run his simulation from a web GUI
3- Sam selects a configure the simulation he wants to run and the parameter for the HPC job he wants to run (number of nodes for instance) from a web GUI
4- Sam launches the simulation from a web GUI, waits for confirmation that the job has been queued and logout
5- Later, Sam checks the job status from a web GUI
6- Sam sees that the simulation finished.
7- Sam configures and launches pre-canned analysis jobs, Sam may select a different compute centre
8- Sam waits for completion of the analysis job and he can check the status in a web GUI
9- Sam visualizes the analysis results in a web GUI
10- Sam wants to interactively analysis the simulation results in the HBP collaboratory in a jupyter notebook
11- Sam wants to visualize interactively the simulation. The compute resource for the visualization may or may not be in the same location than the compute resource.
12- Sam registers his simulation in the HBP Knowledge graph
13- Sam simulation reports are stored in the HBP Object storage

Sam does not need his HPC allocation anymore

18.3 Annotated Use Case Diagrams

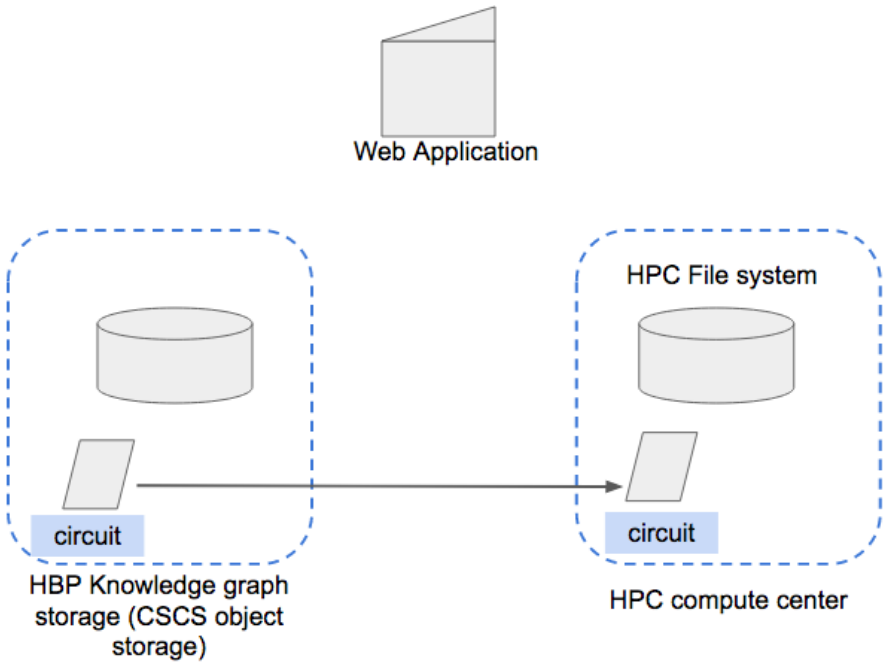


Figure 34: The circuit data have to be somehow made accessible to the HPC compute centre.

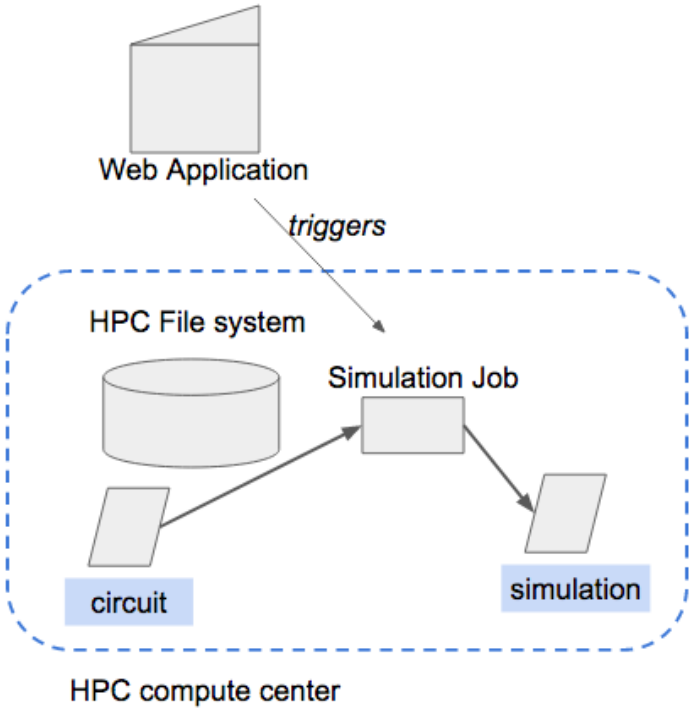


Figure 35: The simulation generates output file (simulation reports) on the HPC compute centre.

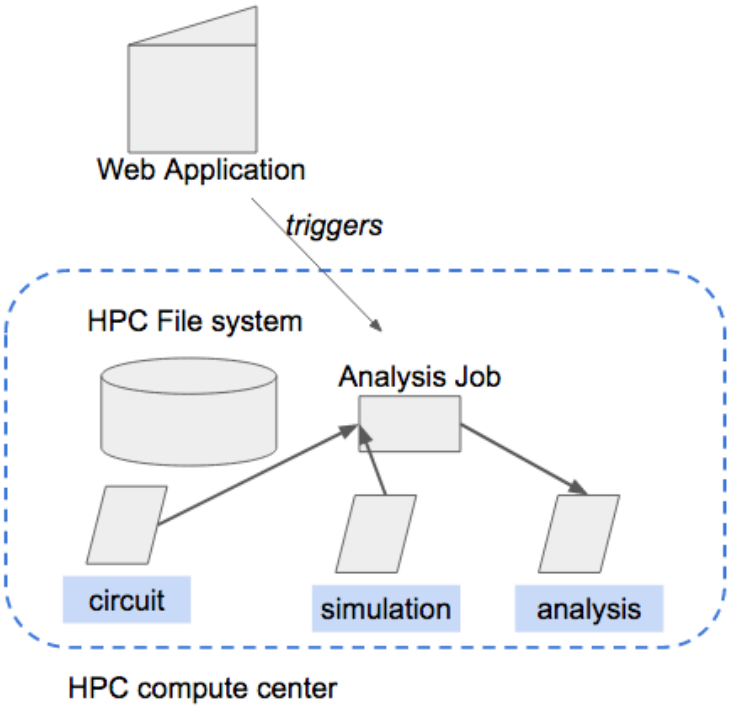


Figure 36:The analysis job generates analysis data.

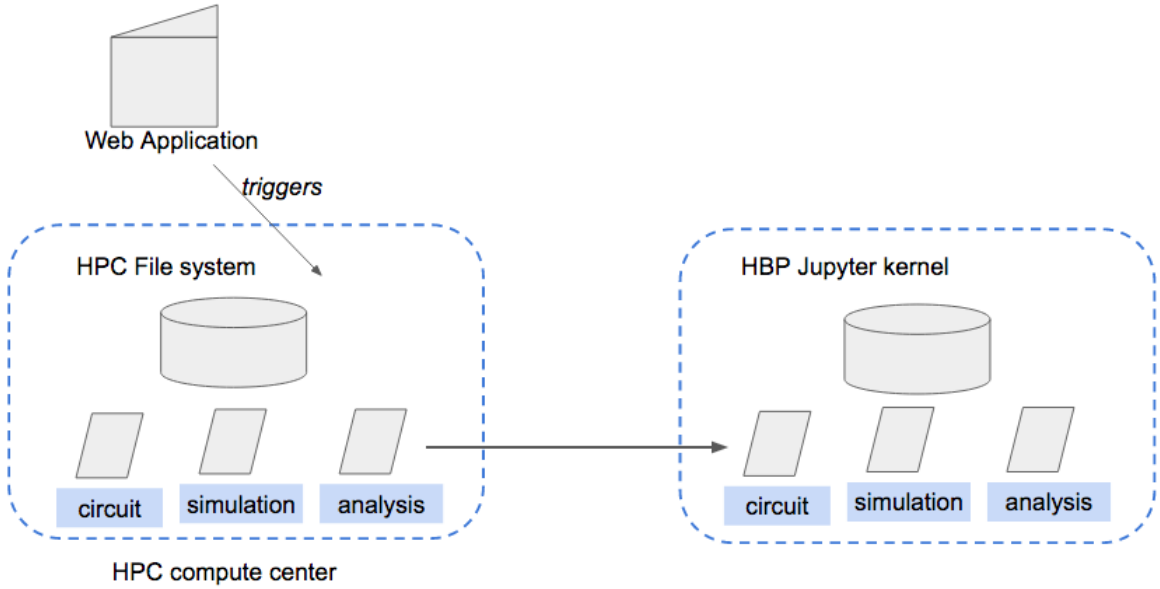


Figure 37: Sam performs interactive analysis in Jupyter notebook Somehow, the data are made accessible from the jupyter kernels.

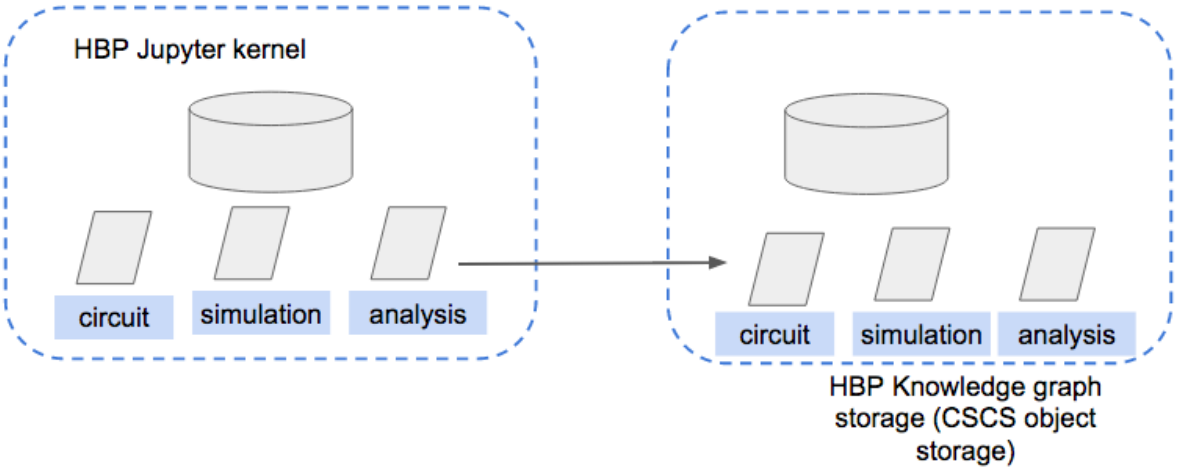


Figure 38: Simulation and analysis are stored in the HBP Knowledge graph object storage.

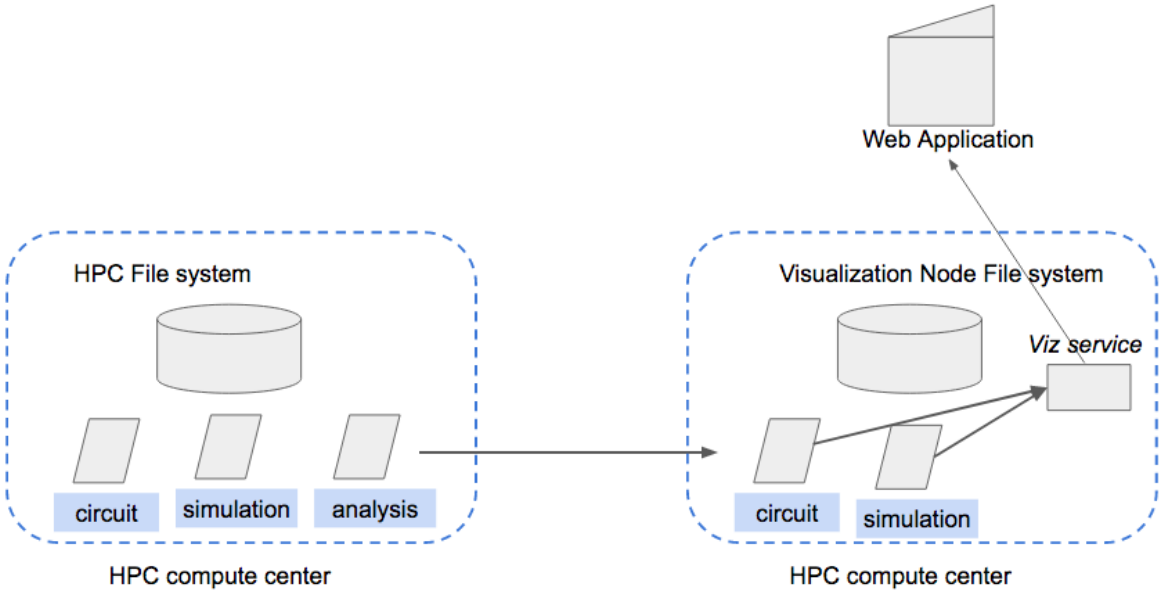


Figure 39: Sam visualizes the simulation.

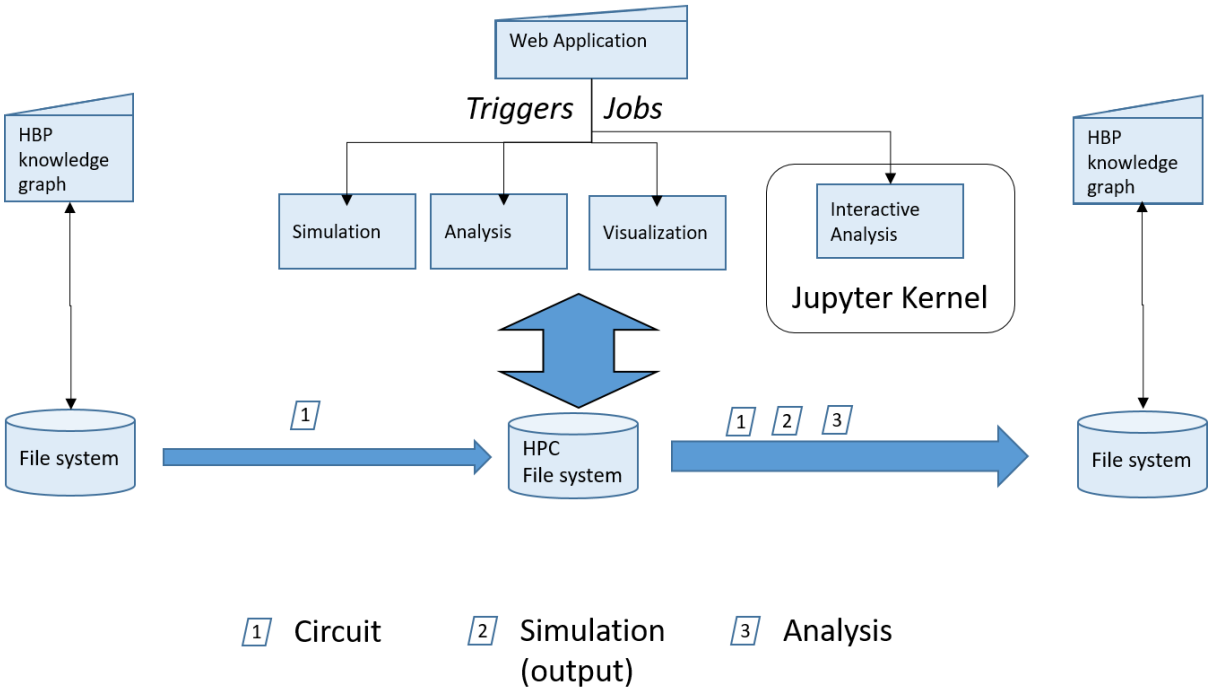


Figure 40: High-level overview combining all previous diagram into a single picture.

18.4 Node Characterization

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: maximum available on the node; the simulation is memory bound on the use of cores/node
- CPU: large size jobs
- Specialized hardware: KNL (KNL architecture will not be built anymore by Intel. Should this be GPU?)
- Storage: 5TB/job

Architecture Requirements:

- Minimal compute performance (excluding acceleration)
- GPU requirements per node (minimum): N/A
- GPU configuration (minimum HBM): N/A

The default entries in this list have been deleted.

18.4.1 Data objects

Data object: ***1***, Circuit	
Base information	General description of what data is stored • Morphology topology description in text files or in hdf5 formats

	• Electrical behavior of the neurons as code (hoc files) • Channel models as code (mod files) • Synapse models as code (mod files) • Cell properties (position, orientation, metadata…) hdf5 file • Synapses location and properties, hdf5 files.
Technical specifications	• Short-term and/or Permanent. Additional information *Access control requirement* The circuit when located in the CSCS Object storage is subject to the knowledge graph ACLs and the CSCS Object storage ACLs The circuit becomes accessible to all the user of the HPC project storage it is copied into. The circuit data are visible only by the user when located in its jupyter kernel *Access requirements* The circuit data needs only to be read-only *Data Size:* The total number of circuit increase slowly in time (~ 4 per year) For a circuit, we have ~1000s of files with a total size of >200GByte Most of the files are electrical and morphology model (ascii files). Most of the size is consumed by synapses files (~ 6 of them) of ~20 GByte. These are HDF5 files.
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***2***, simulation data	
Base information	General description of what data is stored The data represents: • spike time reports. A 2 column text file with time and gid of the neuron spiking • variable reports. A binary file that contains the value of Neuron simulator variables for different locations at different time steps.
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information *Access control requirement* The simulation data are under the HPC project ACL when located in

<table>
<tr><td></td><td>the HPC compute centre.

The simulation data ACL are defined by the knowledge graph and its storage when copied in the HPC Storage (they can be different from the HPC Project one)

The simulation data are accessible only by the user when visible in its jupyter kernel

Access requirements

The simulation needs only to be read-only

Maximum and average capacity requirements

The total number of simulations will increase in time (~100 per year)

Data size:

We have 10s of files with a total size of ~10 GByte. The biggest file is the simulation report.

Size in KB for a set of simulation reports (there is one report per simulation)

<table>
<tr><th></th><th>size</th></tr>
<tr><td>count</td><td>3.463000e+03</td></tr>
<tr><td>mean</td><td>7.531331e+06</td></tr>
<tr><td>std</td><td>1.119266e+07</td></tr>
<tr><td>min</td><td>1.996000e+03</td></tr>
<tr><td>25%</td><td>2.451604e+06</td></tr>
<tr><td>50%</td><td>3.676056e+06</td></tr>
<tr><td>75%</td><td>4.060008e+06</td></tr>
<tr><td>max</td><td>1.714423e+08</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Current solution</td><td>Name: NA</td></tr>
<tr><td>URL to additional information: NA</td></tr>
<tr><td>Limitations: NA</td></tr>
</table>

<table>
<tr><td colspan="2">Data object: 3, analysis data</td></tr>
<tr><td>Base information</td><td>General description of what data is stored
Output of analysis both python based and from interactive sessions.

This output is task and experiment specific and typically hard</td></tr>
</table>

	written by a scientist. It is not possible to provide abstractions of the formats of meta data. Typically analysis results in orders of magnitude of data size reduction compared to raw simulations output. Sizes << Data object: 2, simulation data
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

18.4.2 Data transport

The data transport should be triggered by a HBP authenticated REST API.

Data transport: HBP knowledge graph to HPC file system	
Base information	General description of what data is transported: Circuit data as stored in the platform (1) Q: Would local buffering of this data be acceptable? High bandwidth connections are typically only available at local site.
	Data access patterns (request rate, transfer sizes) NA
Technical specifications	Maximum required bandwidth: NA Q: Is the current solution limiting in the science production?
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: HPC site data transport to compute node	
Base information	General description of what data is transported: Circuit files (1) simulation output (2) and analysis results (3).
	Data access patterns (request rate, transfer sizes) NA
Technical specifications	Maximum required bandwidth: NA Q: Is the current solution limiting in the science production? Is the NEURON simulation bottle neck network communication or compute?
	Average required bandwidth: NA

	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: Simulation and analysis results to Long term storage in the Knowledge graph	
Base information	General description of what data is transported: Circuit files (1) simulation output (2) and analysis results (3).
	Data access patterns (request rate, transfer sizes) NA
Technical specifications	Maximum required bandwidth: NA Q: Does this need to happen at maximum speed, or would a batched mode with runs the transfer at night be allowed. In other words would it be ok if the data is only available a couple of days later?
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

18.4.3 Data ingest / GUI

Data ingest: NA	
Base information	Description of input data source: The Data GUI are web application provided by the Brain Simulation Platform. They rely entirely on HBP authenticated REST API to run any of the needed service. For instance, we use UNICORE REST API to launch a job on a support computer, or checking its status, we use the HBP Knowledge graph REST API to query metadata about the circuit and simulation data.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?) NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

18.4.4 Data repository

Data repository: The HBP Knowledge graph and the HBP Knowledge graph storage	
Base information	Classification of the data objects (see below) NA The HBP Knowledge graph is a graph database managing metadata for artifacts consumed and produced by the HBP. The HBP Knowledge graph object storage is an Object Storage located at CSCS that stores the artefact consumed and produced by the HBP
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: The computer centre file system	
Base information	Classification of the data objects (see below) NA The compute centre file system is the file system allocated for the user as they get a HPC project.
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: The jupyter notebook kernels	
Base information	Classification of the data objects (see below) NA The jupyter hub is a system deployed by the HBP for the end user

	to perform interactive analysis. The kernel are currently deployed at CSCS.
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

18.4.5 Processing stations

Processing station: The HPC compute centre	
Base information	General description of data processing: The HPC compute centre is the location where the simulation job and the analysis job will be executed. Core/hours per year: +60M TB/Year output data: approximately 200TB/year Q: Do you have any special requirements for the hardware? GPU, large memory, etc. Remark: We are buying for years ahead. So looking at things in the future is informative for us.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA Q: Neuron or CoreNeuron installed?
	Ratio of data processing rate versus data consumption and production rate NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: A simulation job of a microcircuit requires	
Base information	General description of data processing: The "node" where the visualization will be executed. The service will act as a server that streams the visualization to the external world Q: Will all jobs be running on a single node?
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

18.4.6 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? NA • What is the expected typical duration of interactive sessions? NA

	• What software stacks need to be available? NA • Is it possible to define memory capacity requirements? NA Yes for the visualization.
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? NA Running a simulation or an analysis job.
Virtual Machine Services	• Which parts of the workflow require such services? NA No – outside of the jupyter kernels that are instantiated at CSCS.
Active Data Repositories	• Which parts of the workflow require such services? Unknown
Archival Data Repositories	• Which parts of the workflow require such services? No
Data Mover Services	• Which parts of the workflow require such services? most parts of the workflow require these services: • Moving data from HBP Object Storage to HPC compute centre • Moving data between compute centre for analysis • Moving data between HPC compute centre and the jupyter kernels • Moving data between HPC compute centre and the HBP knowledge graph object storage
Data Transfer Services	• Which parts of the workflow require such services? NA • Between which ICEI sites is data planned to be transferred? NA • How much data is expected to be transferred per time unit? NA • How are transfer patterns expected to change over time? NA
Data Location Service	• Which parts of the workflow require such services? The transfer service may need that requirement to avoid copying data already present on the target
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site?

	The transfer of the circuit and/or the simulation data should be done within one hour.
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? Yes for the visualization service
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? All the services that triggers data movement or a job execution should be executed through a HBP authenticated REST service
User Support Services	• Are the specific foreseeable needs for user support services? No

19. Data management and big data analytics for high throughput microscopy (#15)

Data management and big data analytics for high throughput microscopy

Use Case Description and Specification

22-06-2018 Wouter Klijn, Timo Dickscheid

Partners	Marcel Huysegoms, Christian Schiffer
Institutions	INM1 FZJ
Principal Investigators	Timo Dickscheid

Date	Version / Change
13-06-2018	(Wouter Klijn) Initial scientific write down and technical workflow breakdown
14-06-2018	(Wouter Klijn) Merge in SGA1 template information
21-06-2018	(Wouter Klijn) insert TD information on split use case
22-06-2018	(Wouter Klijn) Initial Telecon with explanation of the effort
07-09-2018	(Klijn & Carstensen) recreate template and editorial changes
14-09-2018	(Timo Dickscheid) Remove redundant parts from other INM1 use cases
19-09-2018	(Wouter Klijn) Editorial changes and resolution of remarks Timo.
25-09-2018	(Schiffer, Haas & Huysegoms) Fill in infrastructure requirements.
25-09-2018	(Wouter Klijn) Editorial changes

19.1 Use Case Template

UC-Atlas-002: Data management and big data analytics for high throughput microscopy
Expert contacts: Marcel Huysegoms (INM-1) for data sizes formats, Christian Schiffer (INM-1) for hardware requirements in Deep Learning with Convnets

These are your Figures 2 and 5.
Key problem here: Continuous data ingest in the order of multiple Terabytes per day from a lab with fast network connection to the HPC centre. Quality control and pre-processing needs to be in sync with the acquisition, ideally realized by stream processing instead the classical batch processing.

- Quality control at the full resolution are realized as HPC jobs, and triggered after successful initial checks at reduced resolution in the lab and subsequent transfer of each image to the HPC storage systems. Quality control should be scheduled

in a timely manner after the data has been transmitted to the HPC filesystems. Can such jobs be scheduled with priority?
- Expected data ingest in Jülich 2-3 TByte per day
- Video compression of such data is a research topic. It may reduce the footprint to below 50%, but implications of compression for random access for analysis and remote visualization (see UC-Atlas-001) are yet unclear. Giacomo Mazzamuto at LENS did some initial tests, in Jülich Marcel Huysegoms and Pavel Chervakov will investigate in SGA2.
- Data after quality control is further processed by MPI jobs: image segmentation, image registration. This includes basic image processing (image filtering operations parallelized by domain decomposition, Deep Learning with Convnets). At least 50% of the processing jobs require GPUs with large working memory.
- Data processing generates derived data, multiplying roughly by a factor 3.
- We are currently investigating whether the derived data is stored explicitly as images, or highly compressed as metadata (contours of image segments, registration parameters) and then transformed to images on demand (applying deformation to images or generating pixel masks for segments when needed). This would mean that the data is not multiplied as mentioned above, but visualization and analytics of the derived data require more computational resources – still unclear.

19.2 Use Case Description

19.2.1 SGA2-SP7-UC002 - Enabling data management and analysis for the Human Brain Atlas

The HBP Human Brain Atlas is a multi-scale, multi-model atlas with highly diverse qualitative and quantitative datasets that need to be spatially and semantically registered. These datasets have different file formats, come from different partners and need different kinds of user interface functionality. The data is collected within various HBP Subprojects, first of all in SP2. To make this data discoverable and accessible, the Neuroinformatics Platform (NIP) supports users in curating and sharing the data with other researchers in the HBP. It builds upon the infrastructure and services provided by Fenix and the HPAC Platform. In particular, the NIP requires “central” HBP data repositories (which may be federated as long as this is transparent to the user), access control and long-term data storage. The different types of user interfaces need to be supported and provenance tracking should be enabled. An efficient, sophisticated data management is of particular importance since some of the datasets are quite large. To give an example, a single complete human brain at 1-micron resolution requires, depending on the method, 2-6 Petabytes of storage space just for the original data.
The HBP Rodent Brain Atlases have very similar requirements with respect to the infrastructure, data management and tools. This use case focuses on the Human Brain Atlas, but synergies will be used to enable also the data management and analysis for the Rodent Brain Atlases.

This use case is an important contribution to the creation of the HBP Joint Platform since it describes key interfaces between SP5 and SP7 that are required to achieve the SGA2 High-Level Objectives HO1 "Establish, operate and disseminate the HBP Joint Platform (HBP-JP), based on the existing individual Platforms" and HO2 "Establish the gathering, organization and dissemination of neuroscience and medical data as the core of the HBP Joint Platform [...]", as well as HO5 "Develop neuromorphic computing, high-performance computing and neurorobotics into a pioneering approach [...] and enable extreme-scale computing for neuroscience simulation and data science applications" and HO6 "Establish [...] open data and open science as guiding principles of HBP research, which beneficially affects society as a whole". It is needed to enable the achievement of the SP5 FPA Operational Objectives OO5.2 "Identify, curate and integrate multilevel human data from the neuroscience community, as well as SP2 and SP3" and the SGA2 Objective SO5.1 "Ensure that HBP data and models are discoverable and accessible, via meta data enrichment and Fenix provided storage".

Infrastructure need:
The HPAC Platform will be migrated during SGA2 towards a unified platform running on top of the Fenix infrastructure services (SO7.2), that will be implemented in the ICEI project. Therefore, the infrastructure-level requirements of this Use Case will be targeted in ICEI, where this is described in the Use Case "Enrichment of the human brain atlas with qualitative and quantitative datasets".
ICEI will in particular take care of providing:
• Virtual Machine (VM) services for hosting the Collaboratory and SP5 Knowledge Graph.
• VM services, interactive computing services and scalable computing services for data processing and analysis.
• Archival data repositories for long-term archiving (for the duration of the ICEI project).
• Active data repositories for data processing and analysis.
• Enabling services with database back end, that will be integrated with the federated Fenix
AAI, e.g. providing VM services with pre-installed DBMS, and interfaces to connect the components running in the VMs with the Fenix and HBP AAI.
• Support and guidelines to integrate community services with the federated Fenix AAI.
Details about these capabilities can be found in the ICEI Grant Agreement. For more information on ICEI, see Appendix 3.

19.2.2 Data acquisition and analysis in the context of human brain atlasing

Important aspects Microscopic image analysis:
• Scalable long-term storage capacity (growth in the range PBs/year)
• Efficient workflows to package & compress data to/from long term storage
    • Efficient compression of microscopic images (cf. Giacomo Mazzamuto) – video compression?
• Access data with image services for remote visualization
• Fast random access to large image datasets during HPC jobs for e.g.
Deep learning (10s of TBs)
• High-bandwidth sequential access to large image datasets for image registration

• 10s to 100s of TByte with a bandwidth of about 1 GByte/s/node
• Efficient 3D spatial range queries for microstructure attributes
• Ad-hoc generation of database instances for compute jobs?

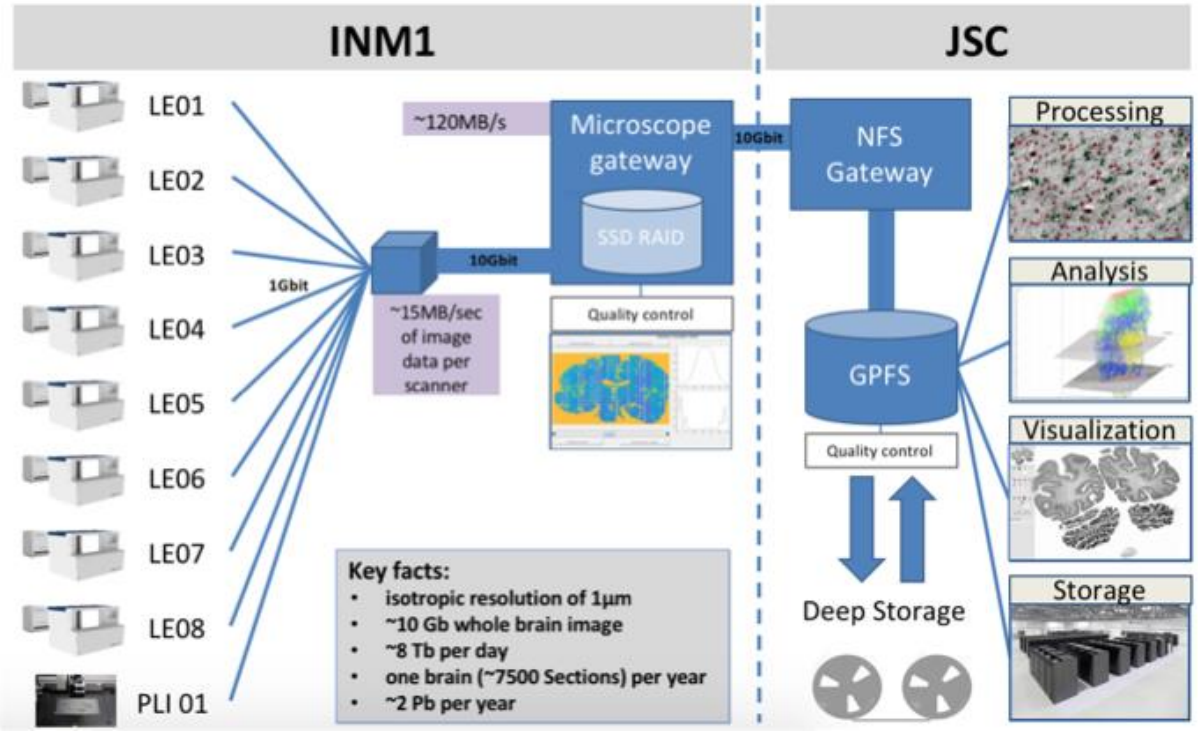


Figure 41: High level data acquisition workflow (K. Amunts, Jülich).

Current solution(s)

19.2.3 all data is on $DATA, because the sections are currently on $DATA

- cell segmentations are saved dense, planned: sparse (openvdb) to save memory

currently all scans stay on $DATA, nothing is moved to $ARCH

- currently local validation
- planned: easy remote visualization using microdraw / neuroglancer to validate results

currently sbatch job scripts are used

- single scripts are started manually by the user
- planned: snakemake workflow
- planned: unicore workflow

Goal(s) Timo

- non experts should be able to start the workflow
- start the workflow without own JURECA account
- dependency handling for different workflow steps
- efficient resource usage

19.3 Diagrams

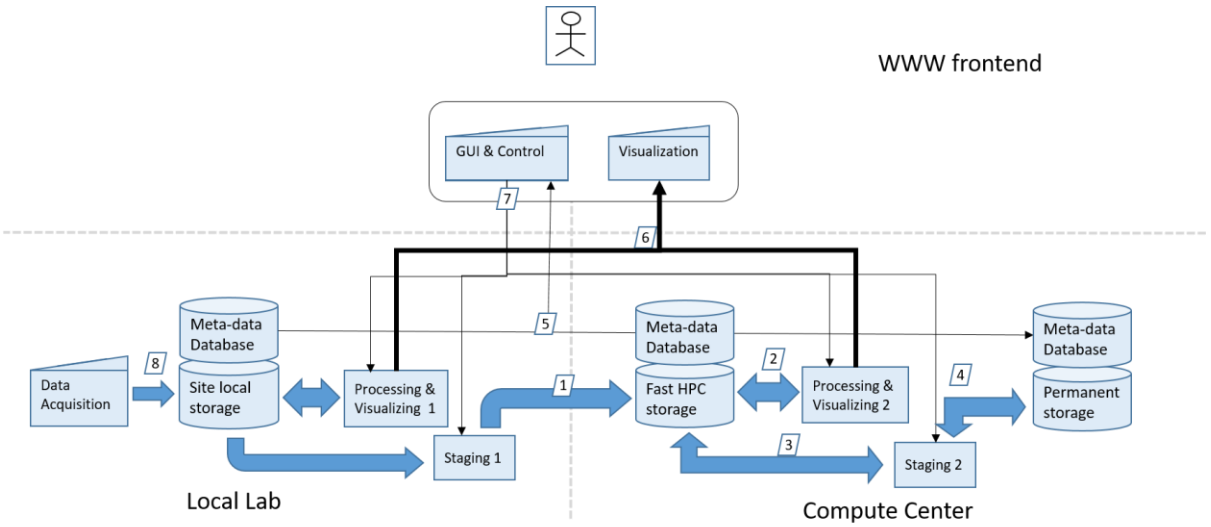


Figure 42: Highlevel abstraction of the INM1 processing pipelines.

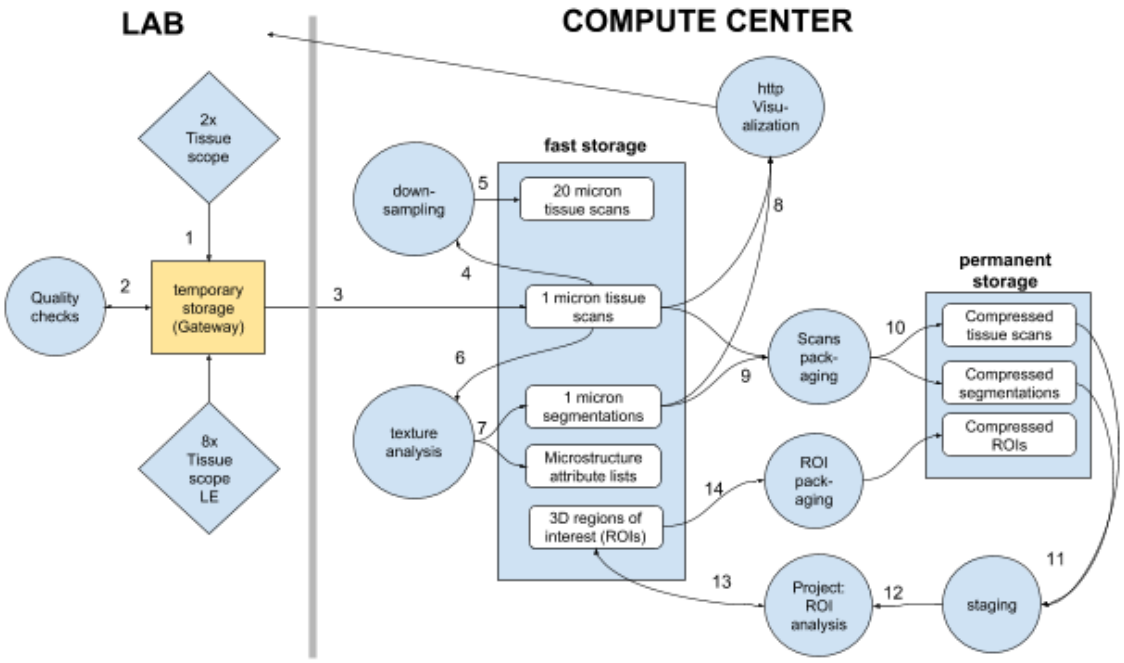


Figure 43: SGA1 highly detailed processing diagram. "http visualization" is a minor issue for a few users. It is addressed for many users, in particular to the open web, in use case 14. The techniques described there can be applied here, so we can ignore the requirements for the visualization here more or less.

19.4 Node Characterization

This use case description covers multiple workflow. The different nodes in the diagram can occur multiple times.

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: needed per node, in total
- IO: Bandwidth, latency, always on/dedicated
- CPU: large size jobs / farming
- Specialized hardware: (GPU, KNL, FPGAs)
- Storage: size, access rate
- Specialized software: VM/containers
- Specialized features: in-situ visualization

19.4.1 Data objects

Data object: ***1*, 2, 3, 4:** Science data products: 1 micron scans	
Base information	General description of what data is stored: Raw or curated big-data to be transported from the local lab to the HPC centres. • Formats: BigTiff • Metadata: NA • Database requirements: NA BigTiff Pyramid images of ~12GB per file. Tissuescope LE produces 30 images per tissue section, Tissuescope only 1.
Technical specifications	• classification: campaign (becomes permanent in compressed form, see below)
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***1*, 2, 3, 4:** Science data products: 1 micron segmentations	
Base information	General description of what data is stored: Raw or curated big-data to be transported from the local lab to the HPC centres. • Formats: NA • Metadata: NA • Database requirements: NA Segmentation result after processing the 1 micron scans. Labelled arrays, stored as sparse arrays. Format under investigation - sparse matrix hdf5, or openvdb. Cassandra has been suggested by BSC. Currently we generate 1 segmentation image per tissue section for cell segmentations.
Technical specifications	• classification: campaign (becomes permanent in compressed form, see below)
Current solution	Name: NA
	URL to additional information: NA

	Limitations: NA

Data object: ***1*, 2, 3, 4:** Science data products: 20 micron scans	
Base information	General description of what data is stored: Raw or curated big-data to be transported from the local lab to the HPC centres. • Formats: TIFF • Metadata: NA • Database requirements: NA Downscaled scans (1µm -> 20µm), ~30MB per file (8 bit), 1 file per original scan, saved as TIFF.
Technical specifications	• classification: permanent
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***1*, 2, 3, 4:** Science data products: Compressed segmentations	
Base information	General description of what data is stored: Raw or curated big-data to be transported from the local lab to the HPC centres. • Formats: NA • Metadata: NA • Database requirements: NA When writing to permanent storage, we convert each 1 micron segmentation result to compressed format, t.b.d. We envisage compressed, sparse 8bit hdf5, or openvdb.
Technical specifications	• classification: permanent
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***1*, 2, 3, 4:** Science data products: 3D regions of interest (ROIs)	
Base information	General description of what data is stored: Raw or curated big-data to be transported from the local lab to the HPC centres. • Formats: NA • Metadata: NA • Database requirements: NA Stacks of hundreds of cropped regions of interest from the 1µm section. These are computed in consecutive projects, not

	synchronized with data acquisition. We envisage to analyze 5-10 ROIs per year.
Technical specifications	• classification: permanent **600GB** per small ROI
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: **2**, Science data product: Cell/ Microstructure attribute list	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA HDF5 files containing attributes of microstructural features (e.g. cells, blood vessels) ~1MB per file, 1 file per 1 micron scan Additional information: Icei co-design workshop presentation, slide 8-9 Coordinates with attributes • Small brain area (few mm3): hundreds thousands of neurons • complete human brain: almost 90 billion neurons
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it.
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: **7**, GUI and control signals	
Base information	General description of what data is stored: • Formats: control messages (JSON / XML) • Metadata: None • Database requirements: None
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

19.4.2 Data transport

Data transport: **8,** Lab Local data transport	
Base	General description of what data is transported:

information	Lab internal transport of raw images from scanner to local storage and to staging application Copy scans to data mover • Data is stored on local disk of microscope PC in bigtiff format • After scanning each tissue section, a network replication of the bigtiff file is initiated (envisaged: use CERN's fdt for this) • Bandwidth:15MB/sec per Tissuescope LE scanner (normal Tissuescopes require less), approx. 150MB/sec in total for all devices Bigtiff file sizes approx. 25GB/each From each scanned section, a thumbnail image is read by a quality checking (QC) batch job. If QC is passed, replication to GPFS is initiated. Negligible bandwidth.
	Data access patterns (request rate, transfer sizes): Single pass
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: Figure 43: Shows the details regarding this data transport
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **1,** Data transport from lab to HPC centres On-site high throughput microscopy setup (K. Amunts)	
Base information	General description of what data is transported: Data object 1: After successful QC, file transfer of each individual bigtiff to fast disk storage at JSC is initiated (currently $DATA on GPFS at /data/inm1). Required bandwidth approx. 150MB/sec continuously. 10GBit line currently being established. Transport through nfs mount of /data/inm1 to the data mover gateway at INM1.
	Data access patterns (request rate, transfer sizes) Single pass 5-10 TB/day ~2PB per year (one brain) Moved to HPC centre (Juelich) GPFS using an nfs mount pointing to /data/inm1/...
Technical specifications	Maximum required bandwidth: 10Gbit/s
	Average required bandwidth: 150 MByte/s

	Interface requirements for attached entities: NA
	Additional information: Figure 43: Shows the details regarding this data transport
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **2,** Local HPC to Compute transport General introduction	
Base information	General description of what data is transported: Data object 1 This is transport from local fast storage to the compute location. Typically expected to be Infini band or equally fast Parallel read of 1 micron scans for downscaling. For the 3d scans, we only downscale 1/30 of all data (the centre section within each 30-stack). Required bandwidth and processing speed therefore faster than This needs to happen faster than image acquisition, so we need to downscale at Texture analysis. For segmenting cells and classifying vessel-like structures, all data is processed in overlapping chunks in parallel (MPI). A http service reads image tiles from a subset of 1 micron tissue scans and segmentations stored on the fast storage. Read access depends on user requests. The serves is only provide within INM1, we expect 10-15 users per day accessing some sections.
	Data access patterns (request rate, transfer sizes): This is the most variable type of access. Strongly depending on the task at hand
Technical specifications	Maximum required bandwidth: 100 GB/s
	Average required bandwidth: NA
	Interface requirements for attached entities: HDF5 Normal file access
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **2,** Local HPC to Compute transport Cell Segmentation	
Base information	General description of what data is transported: Sections of data: 80k * 100k * 30 pixel 250GB / Section 25 sections per day

	Every section appears to be parallel Output: Cell Attribute list
	Data access patterns (request rate, transfer sizes): 8 TB per day
Technical specifications	Maximum required bandwidth: 100 GB/s
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **2,** Local HPC to Compute transport 3D ROI reconstruction	
Base information	General description of what data is transported: Data object 1 This is transport from local fast storage to the compute location. Typically expected to be Infini band or equally fast N consecutive sections have a ROI cropped / extracted. This is then used for 3d reconstruction This in combination with and Attribute list: Data object: 2, Science data product: Cell attribute list Data size: N * 10k * 10K * 30 Output data is not specified: This should result in an additional data object
	Data access patterns (request rate, transfer sizes): Rate is not clear. Important is the cropping: Not the whole stored section is needed but a subset.
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: Slide 10/11 Icei co-design workshop presentation
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **3, 4** : Transport of working data to and from long term storage	
Base	General description of what data is transported:

information	Data object 1, highly depending on specific task Data to be stored for long term on slow (tape) storage. Meta data should be stored outside to allow search ability. From a systems design perspective it is maybe best to only stage to HPC centres and not to the user pcs immediately
	Data access patterns (request rate, transfer sizes): PBs/year Request rate is totally unknown
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: Scalable long-term storage capacity (growth in the range PBs/year) Efficient workflows to package & compress data to/from long term storage • Efficient compression of microscopic images (cf. Giacomo Mazzamuto) – video compression?
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **5,** Meta data transport	
Base information	General description of what data is transported: In this workflow a large amount of processing steps is done. Each will produce meta-data. This needs to be synchronized between all the different storage locations.
	Data access patterns (request rate, transfer sizes): High rate / low size: Need for publicly accessible data-vase (nips?) A separate copy of this data should be send to long term storage
Technical specifications	Maximum required bandwidth: Minimum
	Average required bandwidth: Minimum
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NIP / Collab
	URL to additional information: NA
	Limitation: NA

19.4.3 Data ingest / GUI

Data ingest: Data Acquisition: INM Scanners	
Base	Description of input data source:

information	8 X scanners located in INM 1 PLI scanner. Each producing 15/MB/sec of image data High throughput light microscopy scanner for 3D scans. 8 devices in Each device produces approx. 1 TB/day. Devices run 7 days a week. From 2018, we expect to produce 200-250 TB/month of scans with these devices (Tissuescope + Tissuescope LE). Each producing 15/MB/sec of image data
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Data object 1. Other information unknown
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Data Acquisition: External data sources	
Base information	Description of input data source: A diverse set outside data sources downloaded and uploaded and to become part of the Brain atlas
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): Characteristic: Upload
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Unknown
	Additional information: S. Eickhoff, (s.eickhoff@fz-juelich.de)
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data ingest: Gui and control	
Base information	Description of input data source: These are use interfaces thus producing control signals Potential source of meta- information This is probably the NIP / one of the HBP user-platforms
	Description of data introduction (upload? scanner characteristics? simulation characteristics?):

	Characteristic: User input
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Unknown Formats: task specific Loads: low Bandwidth: low Latency: Should be low Transports: msg based (ZeroMQ)
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

19.4.4 Data repository

Data repository: Site local storage and meta-data base	
Base information	Classification of the data objects (see below): "Data mover" server, hosted at INM1. Collects data from all microscopes to temporary disk storage Planned to cache incoming data for ~8 hours
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

19.4.5 Processing stations

The nature of the use case results in a large amount of processing steps. They cannot be detailed individually. The best way forward is a general description of the work to be done and one or two detailed breakdown of processing station, once with high resource requirements. Which there are is up to the use case HPC expert / scientist.
This section is highly task specific and should be filled in by the domain expert.

Processing station: Processing & Visualizing 1: Quality checks

Base information	General description of data processing: Site local preprocessing and visualization of data We run quality checks on thumbnails of the data (lower resolution previews) to avoid transferring bad quality scans to the compute centre. If the check fails, the file is not transferred, and an operator is notified to manually transfer or repeat the scan. Although not part of the ICEI HPC mission the constraints of these steps partly overlap with the Processing & Visualizing 1. The raw data visualizer is an important component that should be detailed.
	Typical processing steps: NA
	Number of processing steps: Numerous
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Staging	
Base information	General description of data processing: Transfer to the central HPC sites needs to be controlled with a specific software.
	Typical processing steps: Bundling of meta and data Setting up remote connection Validating integrity of received data
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA

	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: Grid FTP?
	URL to additional information: NA
	Limitation: NA

Processing station: Processing & Visualizing 2: Cell segmentation / Texture analysis	
Base information	General description of data processing: MPI-parallelized analysis jobs for feature extraction (cells, vessel-like structures, etc.). Will read all incoming 1 micron scans immediately after acquisition, and write segmentation images as well as feature attribute lists. Jobs should be scheduled upon successful transfer of 1 micron scans (1 job per scan).
	Typical processing steps: NA
	Number of processing steps: Numerous
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Processing & Visualizing 2: Downscaling	
Base information	General description of data processing: MPI-parallelized jobs for downscaling the incoming data to 20 micron at high quality. Jobs should be scheduled upon successful transfer of 1 micron scans (1 job per scan).
	Typical processing steps: NA
	Number of processing steps:

	Numerous
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Processing & Visualizing 2: 3D ROI reconstruction	
Base information	General description of data processing: NA
	Typical processing steps: NA
	Number of processing steps: Numerous
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Staging 2: Scans packaging	
Base information	General description of data processing: HPC to long-term data transfer "Efficient workflows to package & compress data to/from long term storage • Efficient compression of microscopic images (cf. Giacomo Mazzamuto) – video compression?" Timo Dickscheid 1µm scans are converted into a compressed data format, packaged into larger chunks of 1-2 TB, and moved to permanent storage. For

	visualization and ad-hoc analysis however, approx. 5% of the data should be kept on the fast storage permanently (~every 30th section). See “compressed tissue sections” below
	Typical processing steps: Bundling of meta and data
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: Grid FTP?
	URL to additional information: NA
	Limitation: NA

Processing station: Staging 2: ROI Staging	
Base information	General description of data processing: long-term data store to HPC We will continuously process 3D regions of interest (ROIs) from the data. For each such ROI project, we need to extract image crops from hundreds of sections and stage them on fast storage for the processing workflows. Raw data sizes of ROIs will vary significantly, in the range of a few percent of the whole brain size.
	Typical processing steps: Bundling of meta and data
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: Grid FTP?
	URL to additional information: NA
	Limitation: NA

Processing station: Processing & Visualizing 2: ROI analysis	
Base information	General description of data processing: Each staged ROI is analyzed: a 3D cell density estimate is computed and classified. The result are 2-3 additional representations of the ROI at a slightly lower resolution. Typical sizes t.b.d. Uses a pipeline composed of multiple MPI-parallelized jobs that is currently being developed.
	Typical processing steps: several batch jobs (registration, segmentation)
	Number of processing steps: Numerous
Technical specifications	Data processing hardware architecture requirements:
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Staging 2: ROI Packaging	
Base information	General description of data processing: After processing and publication of the ROI, the project is packaged to permanent storage. Package format and typical sizes (as a function of the region size in% of a typical brain) t.b.d.
	Typical processing steps: Bundling of meta and data
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: Grid FTP?
	URL to additional information: NA

	Limitation: NA

19.4.6 Infrastructure requirements

This section of the template will map from the infrastructure to the use case. Per envisioned infrastructure service we ask specific questions how this service might be used for your use case. There will be overlap with information provided through annotated use case model diagrams. This duplication is **intended** it will allow consistency checks. This avoids the need of fixing the mapping between the model and specific infrastructure services at a later stage.

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? ○ No interactive computing required • What software stacks need to be available? NA • Is it possible to define memory capacity requirements? NA
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? ○ The data acquisition process runs at a constant rate, so processing and storage requirements are constant over time, so no scalable computing services are needed.
Virtual Machine Services	• Which parts of the workflow require such services? ○ None
Active Data Repositories	• Which parts of the workflow require such services? ○ Quality checks, downsampling, texture analysis and ROI analysis are performed on the original 1 micron tissue scans located on fast storage.
Archival Data Repositories	• Which parts of the workflow require such services? ○ Scan packaging and ROI packaging save compressed tissue scans, compressed segmentations and compressed ROIs to the permanent storage system for long term storage.
Data Mover Services	• Which parts of the workflow require such services? ○ Scanned 1 micron tissue sections have to be moved from the lab to the HPC storage (2-3 TB / day). After performing quality checks and downsampling, the original scans and compressed results of the texture

	analyses and ROI analyses are moved to the permanent storage.
Data Transfer Services	• Which parts of the workflow require such services? ○ None • Between which ICEI sites is data planned to be transferred? NA • How much data is expected to be transferred per time unit? NA • How are transfer patterns expected to change over time? NA
Data Location Service	• Which parts of the workflow require such services? ○ None
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site? ○ New brain sections are scanned at a rate of 2-3 TB per day, which have to be transferred from the lab to the JSC file systems.
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? ○ None

Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples:
	  ○ Special accounts for running services
	Higher priority job scheduling for quality checks to check quality shortly after data is transferred to HPC filesystems
	  ○ Needs for fine-granular control of access to data
	NA

User Support Services	• Are the specific foreseeable needs for user support services? ○ No

20. Neurorobotics Platform, large-scale brain simulations (#11)

Neurorobotics Platform, large-scale brain simulations

Use Case Description and Specification

22-06-2018 Felipe Cruz,

Partners Institutions	FORTISS, ETHZ/CSCS
Principal Investigators	Axel Von Arnim, Felipe Cruz

Date	Version / Change
13-06-2018	(Wouter Klijn)Template instantiation
22-06-2018	(Felipe Cruz)Initial version with information
30-06-2018	(Wouter Klijn) add png version of diagram
20-08-2018	(Anne Carstensen) Editorial changes
02-09-2018	(Wouter Klijn) Requests for specific information added
07-09-2018	(Felipe Cruz and Colin McMurtrie) Additional information added

20.1 Use Case Description

The Neurorobotics Platform (NRP) is a tool for studying models for brain, body, and environment in closed perception-action loops through interactive in-silico experiments. It effectively allows scientists to virtualize brain and robotics research.

In the NRP, web-enabled and interactive in-silico experiments connect a brain simulation and an environment simulation in advanced closed-loop experiments. Full models of robot and environment are part of an interactive computer simulation, where the simulated sensors of the robot relay environment information to a simulated nervous system that models a biological brain at different levels of detail, which in turn controls the robot. The user of the NRP can then control the simulation via a web-browser interactively and in real-time.

In terms of the platform utilization, the platform service expects that once it enters production that the number of users can fluctuate between 5 to 50 concurrent users. Where each simulation would be distinctive depending on the application and user, as such, the NRP project expects that the complexity of the brain models and robots used in simulations to be wide-ranging, starting from small (brain models with thousands of neurons and simple robots) to large scales (hundreds of thousands of neurons with complex robotic environments). From a system requirement perspective, the complexity of the models translates to computational and resource requirements that scale quadratically with the number of neurons.

Estimating the total mix of simulations (small-, medium-, or large-scale) is difficult, however, the expected average number non-trivial large simulations per week is estimated to be close to 14, for a total of 700 non-trivial simulations per year.

20.2 Diagrams

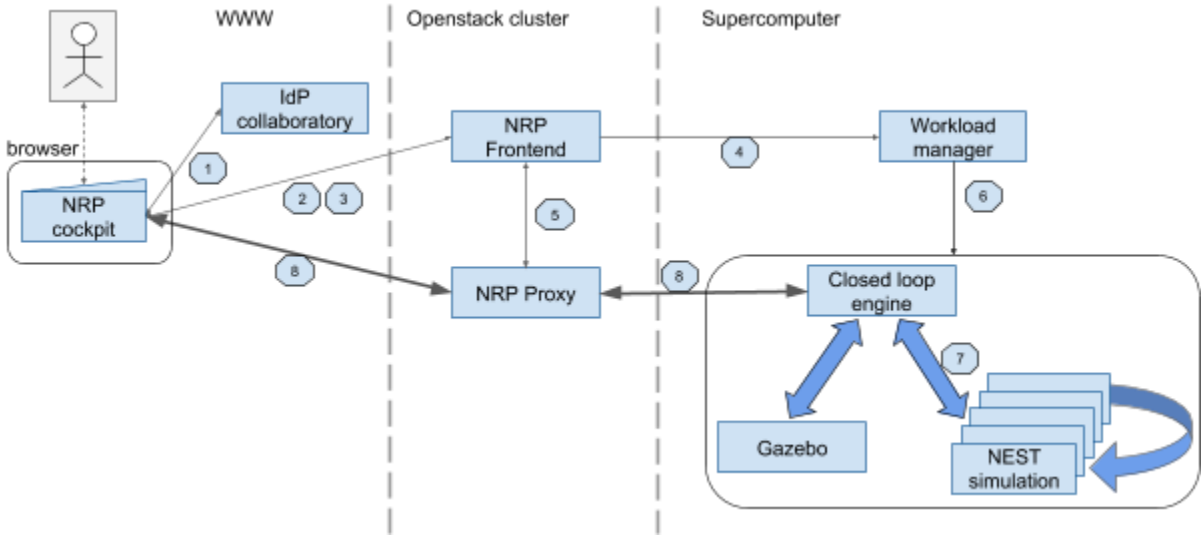


Figure 44

NRP enables the following workflow for a user through a web browser:

1. The users logins into the federated IdP.
2. Robot, environment, and brain models are selected for the experiment by the researcher using the Neurorobotics Cockpit (Frontend).
3. Experiment setup parameters are selected for simulation execution through the Frontend.
4. Frontend submits the experiment for execution on HPC infrastructure.
5. Frontend configures and redirects the user to a proxy server that will become a gateway for the user to interact with the experiment being executed on the HPC infrastructure.
6. Core NRP components including brain and robot model simulations are started on HPC infrastructure (compute nodes).
7. Brain (NEST) and robot (Gazebo) simulations synchronized by the Closed Loop Engine component of NRP, where output of one simulation is used as input of the other and vice versa.
8. Researcher monitors the experiment in real-time, being able to control all states and parameters of the ongoing experiment.

20.3 Node Characterization

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- Minimal compute performance (excluding acceleration)
- Minimal volatile memory footprint of 192 GiByte
- MPI point-to-point bandwidth of 10 GByte/s or higher
- MPI latency of 2 micro-seconds or less
- Access to active data repositories with a bandwidth of up to 8 GByte/s per node
- GPU requirements per node (minimum)
- GPU configuration (minimum HBM)
- Specialized software: Virtual Machines for hosting web-services and Docker-compatible container runtime for running containerized software.
- Specialized features: AAI integration

These minimum requirements seem acceptable. The highest demands will likely come from the NEST simulations (which is an MPI-enabled application take scales on large systems and its requirements have been documented elsewhere). More details can be found in the table below.

Of high interest:

Connections 4 and 8. None of the other workflows have HPC resources communicating directly with an open stack cluster. What are the requirements of this connection? Are there existing solutions? Is it possible that communication is done via File system (and that a data store should be places there?

- NRP itself takes care of the communication between components running on the OpenStack and Piz Daint, however, this requires the creation of an ssh tunnel between the compute node on Piz Daint running the Closed Loop Engine component and the NRP proxy to enable the link between the OpenStack environment and the supercomputer.
- With respect to your second question, communication cannot be done via File system as Step 8 requires bi-directional real-time communication between the user browser and the Closed Loop Engine, this connection is enabled by the "NRP proxy" component.

Step 8 what are the real-time requirements? How big is this data flow?

- NRP by definition is real-time, it provides interactive control, visualization, and data feedback from the simulation.
- In terms of data flow: (I) At a minimum, parametric information of the simulation is passed to the user browser: parameter information of the robot and environment so that it can be rendered by the browser; (II) Normally, the user can stream data from the simulation (NEST and robot sensor), thus, the amount of data will depend on the complexity of the simulation (robot sensor data, environment, and size of brain model).

20.4 Infrastructure requirements

Infrastructure service	Questions to address

Interactive Computing Services	• Which parts of the workflow require such services?
	  ○ The interactive component is a central part of the workflow, enabling scientists to monitor and interact with the high-performance simulation in real-time.
	• What is the expected typical duration of interactive sessions?
	  ○ Duration varies depending on the usage, the interactive component can be used during multiple stages of the scientific workflow, e.g.: preparation of the simulation where close observation is required; monitoring of the production simulation through the real-time output; interactive teaching during workshops.
	• What software stacks need to be available?
	  ○ NRP is a fully containerized platform so other than access to the docker containers it needs a container runtime that support Docker. Currently, all components used by NRP has been containerized, this includes Gazebo and NEST components.
	• Is it possible to define memory capacity requirements?
	  ○ For the VM-hosted services (NRP frontend and NRP proxy) the requirements are 4 virtual CPUs and 16GB RAM
	  ○ The Closed Loop Engine has minimal requirements (1 core, 1GB RAM) as its job is mainly of coordination between the more resource intensive tasks of NEST and Gazebo.
	  ○ For the NEST simulations the requirements will be those coming from NEST (Captured as part of the NEST use case)
	  ○ Gazebo has a requirement for at least one GPU per node/VM with a minimum of 6GB HBP memory

(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? ○ All of it, the efficient use of the available computational resources means that resources must be allocated depending on the number of active users and the type and size of experiments they want to run.
Virtual Machine	• Which parts of the workflow require such services? ○ All workflows use Virtual Machines. Virtual Machines

Services	are an integral component of the production service of the Neurorobotics platform as they: (1) allow for the efficient deployment of the web-services that work as frontend of the platform; and (2) provide the fundamental link between the web and the High-Performance Infrastructure.
Active Data Repositories	• Which parts of the workflow require such services? ○ Reading of input data by NRP at a location where it can be readily accessible by the simulation engines: Gazebo and NEST. While Gazebo store information of robot models and environments that can reach up to a few Gigabytes, it is NEST that has considerable resource requirements. Consider the following: ■ Simplified mouse model (7000 neurons) 32MB per user per simulation ■ Short-range mouse model 100-300 GB per user per simulation ■ Full-scale models can be in the TB range per user per simulation ○ Outputs from NRP: This point is harder to quantify because the ability to save NRP experiment output has not yet been implemented. However, some rough estimates have been made: ■ Recording a small-scale brain simulation experiment: 7000 neurons with 50 spikes per second (8 bytes per spike) for 24 hours written to disk => 7000*50*8*3600*24 = 241 GB per experiment per user (maximum for experiments of this type since not all neurons spike at 50Hz all the time) ■ Recording a large-scale brain simulation experiment 700000 neurons (under the same conditions as above) = 100*241 = 24 TB per experiment per user (maximum for experiments of this type since not all neurons spike at 50Hz all the time)
Archival Data Repositories	• Which parts of the workflow require such services? ○ harder to quantify because the ability to save NRP experiment output has not yet been implemented. However, considering the estimates of the output of a single simulation we can estimate requirements in the order of hundreds of Petabytes.

Data Mover Services	• Which parts of the workflow require such services? ○ Initial setup of data required by NRP and the access to the simulation results.
Data Transfer Services	• Which parts of the workflow require such services? • Between which ICEI sites is data planned to be transferred? • How much data is expected to be transferred per time unit? • How are transfer patterns expected to change over time? ○ It is currently not clear what data, if any, needs to be shared between sites. The current workflow description is site-local. However, one can imagine that some researchers will want to move at least some subset of data between sites.
Data Location Service	• Which parts of the workflow require such services? ○ It is not clear what the requirements on data location services will be. However, one can imagine that users will want to know where their input and output data sets are. They will also likely put some subset of data into the NIP (and therefore KnowledgeGraph).
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single site? ○ One of the core components of NRP is NEST, a simulator for spiking neural network models, which greatly benefits of the high-performance interconnects of supercomputers.
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? ○ NRP frontend portal needs to be reachable from the internet, while bandwidth consideration might be needed to support large number of users.
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? ○ Users of the platform will login using the Identity Provider of the collaboratory, the federated OIDC infrastructure will then allow the users to run using the appropriate site-specific accounts.
User Support Services	• Are the specific foreseeable needs for user support services? ○ As the usage of the NRP grows it is likely that new users will need help and this will likely exercise the User Support Services.

21. Mouse Brain Atlas (#8)

Mouse Brain Atlas

Use Case Description and Specification

02-07-2018 Wouter Klijn,

Partners	Carlo Cavazzoni <c.cavazzoni@cineca.it> Giuseppe Fiameni <g.fiameni@cineca.it> Debora Testi <d.testi@cineca.it> Roberto Mucci <r.mucci@cineca.it >
Institutions	European Laboratory for Non-linear Spectroscopy
Principal Investigators	Ludovico Silvestri

Date	Version / Change
18-06-2018	(Wouter Klijn) Initial seed scientific write-up
28-06-2018	(Ludovico Silvestri) First draft
02-07-2018	(Roberto Mucci) Added infrastructure requirements
20-08-2018	(Anne Carstensen) Editorial changes
02-09-2018	(Wouter Klijn) Insert template information, questions for specific information added
22-09-2018	(Ludovico Silvestri) Review and update on questions for specific information
24-09-2018	(Anne Carstensen) Integration of review comments and updates

21.1 SGA2-SP1-UC07: A multilevel atlas of the whole mouse brain

SP1, with the coordinating support of CDP1, is generating a unique ensemble of structural and functional whole-brain data that are laying the basis for quantitative and realistic modelling of the mouse brain at organ-wide level. The aim of this Use Case is to bring together all the relevant experimental data in a unique framework, allowing efficient analysis of teravoxel-sized datasets, and guaranteeing long-term archiving of information. More fundamentally, we aim at interfacing data generation in SP1 with data sharing services provided by SP5, allowing effective use of experimental data generated in the project.
Whole-brain data generated during SGA2 include whole-brain distribution of different neuronal types at single-cell resolution, whole-brain activation maps in different behaviours at single-cell resolution, whole-brain segmentation of cells. A unified and curated dataset with these properties is not yet available, and it will provide a unique resource both as a reference tool and a ground truth for simulation developers.

Infrastructure need:
We need to define and implement a standardized pipeline to move raw data from the generation site to an active repository, perform analysis with deep learning methods to extract relevant information, integrate this information with spatial reference (atlas), expose the refined data and metadata to the KnowledgeGraph, and finally providing long-term storage and curation of raw and analysed data. Two main deep learning strategies will be pursued. First, we will exploit semantic deconvolution [Frasconi et al., Bioinformatics 2014], a 3D CNN that transforms the original image into an 'ideal' one where cell bodies are clearly visible (and with homogeneous contrast) while other fluorescent structures are dimmed. On this ideal image, a simple clustering algorithm (mean shift) can reliably localize the centre of cell bodies. The second deep learning approach we will use is based on direct segmentation using a 2D CNN followed by a contour finding algorithm [Mazzamuto et al., LNCS 2018]. In this way it is possible to obtain not only the centre of fluorescent neurons, but also their shape and volume, allowing further classification of different cell types.

The whole pipeline must be capable of handling dozens of TBytes of raw data, and should also include visualization tools tailored for the different stages of data processing.

General specs of this pipeline include:

- Data size: about 8 TByte per sample, about 30-40 samples forecasted in SGA2. 3-400 TByte in total. In the PByte range for SGA3
- Efficient data transfer and handling
- Efficient data compression strategy allowing fast random access
- Visualization services
- Stitching tools
- Deep learning for feature extraction (cell localization, cell segmentation)
- Image registration to reference atlas
- Data and metadata ingestion in HBP Knowledge Graph

21.2 Diagrams

No case specific diagram has been produced. "Generalized data acquisition, validation, processing, and storage" might be a good starting point (can be found on page 4 of the PowerPoint).

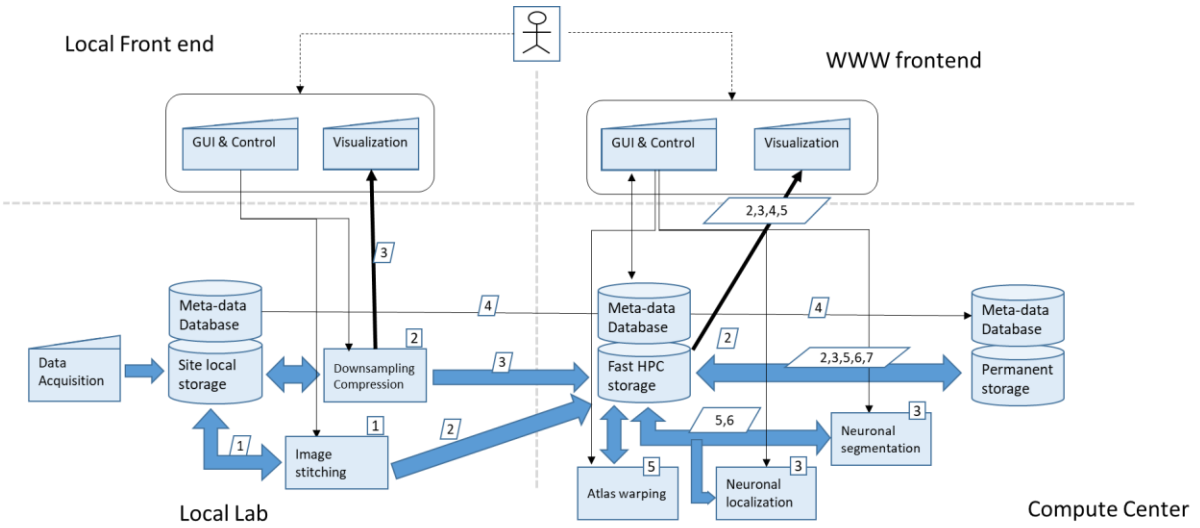


Figure 45: Data processing pipeline for the use case. Still under development. It would be nice to be able to interact with the HPC environment not only with the usual batch job submission, but also more interactively via www. Ideally, the www front end should allow to 1) visualize data, 2) download data, 3) launch analysis job, 4) transfer data between the lab and the compute centre, 5) transfer data between fast storage and long-term storage.

21.3 Node Characterization

Information as received outside of the template:
Q. How much data will be produced until the end of the project that needs to be stored in archival data repositories?

A: The yearly need of storage would be around 1PB (8TB per mouse brain raw data, about 100 mice acquired per year in the future, derived images (segmented, stitched etc) should also be long term preserved.

Q: What are the expectations of the researchers wrt to compute resources? Are there special needs, e.g. use of accelerators, nodes with large memory capacity?

A: Researchers expect to have fast I/O, GPUs, large amount of RAM (>128GB) together with the possibility to interactively access the data/tools.

Q: What kind of services are planned to be deployed in the context of this use case?

A:
ZetaStitcher (https://github.com/lens-biophotonics/ZetaStitcher)
Brain cell finder (http://bcfind.dinfo.unifi.it/)
ANTs (https://stnava.github.io/ANTs/)
NiftyReg (http://cmictig.cs.ucl.ac.uk/wiki/index.php/NiftyReg)
Aliquis (http://www.bioretics.com/aliquis)

21.3.1 Data objects

Data object: ***1***: Raw data	
Base information	General description of what data is stored: Raw big-data generated at the local lab, to be transported to the HPC centres. • Formats: .dcimg or .tiff (one file per image stack) • Size (typ): 25 GB per image stack, 160 stacks (4 TB) per dataset (half mouse brain), 8 TB per mouse brain • Resolution: 0.65x0.65x2 um Light-sheet microscope • Metadata: NA • Database requirements: allow transparent propagation of the metadata to KnowledgeGraph
Technical specifications	• Long-term (10 years): Raw data should be preserved as long as possible, as they guarantee the full reproducibility of the whole analytics pipeline Additional information: NA
Current solution	Name: GridFTP [to move data to HPC centre] iRODS [to move data to HPC centre] SWIFT API [to move data to HPC centre, not tested yet]
	URL to additional information:
	Limitations: We were not able to use SWIFT yet. We anyway are very comfortable with iRODS. We should decide a strategy as soon as possible. Further, we still do not have a real standard for metadata (we've used Google sheets until now...)

Data object: **2:** Raw stitched images	
Base information	General description of what data is stored: As 1, but after performing image stitching. Original files are integrated with a stitching file containing stitching metadata. • Formats: original files as in 1. Stitching file: .yml • Metadata: NA • Database requirements: NA
Technical specifications	• Long-term (10 years): Raw data should be preserved as long as possible, as they guarantee the full reproducibility of the whole analytics pipeline Additional information: NA
Current solution	Name: ZetaStitcher [to transform raw data into curated one] Virtual Fused Volume (from ZetaStitcher): a Python API to access the volume as a unique array but keeping the data in place

	URL to additional information: https://github.com/lens-biophotonics/ZetaStitcher
	Limitations: further optimization of I/O for high-speed random access is needed.

Data object: **3:** Compressed and down-sampled images	
Base information	General description of what data is stored: Compressed (lossy) full-resolution, and uncompressed downscaled version of the dataset. • Formats: compressed: .zip (1 for each image stack, composed of one .jp2 for each slice). Down-sampled: 3D uncompressed tiff (1 for each image stack). Stitching file: .yml • Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: JPEG2000 lossy for compression.
	URL to additional information: NA
	Limitations: Compression strategy is still suboptimal. (see comments below)

Data object: ***4*:** Acquisition metadata	
Base information	General description of what data is stored: Metadata summarizing all relevant aspects of the experiment • Formats: .xml or yml or similar, allowing queries. Ideally, it should be compatible with NIP • Metadata: type of stained cells, preparation protocol, resolution, excitation wavelength, fluorophore, animal ID, task/component ID • Database requirements: database over the metadata fields
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: none
	URL to additional information: NA
	Limitations: NA

Data object: ***5,6*:** Segmented images	
Base information	General description of what data is stored: Vectorial representation of the pixelated big-data images. Could

	include the position of cell bodies (point cloud representation), or also shape, volume and other features of single cells. • Formats: .pcd (point cloud), .xml (vectorial representation with cell shapes). • Metadata: NA • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: BCFind for point cloud extraction, Aliquis for segmentation
	URL to additional information: ,
	Limitations: Software testing on HPC still to be done

Data object: ***7*** Imaged warped to atlas	
Base information	General description of what data is stored: Original curated images, and segmented data, spatially warped to reference atlas, in a way that spatial queries can be done using standard coordinates. • Formats: Container with lossy compressed files (.jp2000 or .mp4) for curated data, .pcd (point cloud), .xml (vectorial representation with cell shapes). • Metadata: Pixelated images are warped such that there is a voxel-by-voxel correspondence to reference image. In vectorial representation, the coordinates are expressed in a standard reference system • Database requirements: NA
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: ANTs, NiftyReg
	URL to additional information: ,
	Limitations: Software testing on HPC still to be done, final pipeline not yet implemented

21.3.2 Processing stations

Processing station: ***1*****:** Image stitching	
Base information	General description of the processing: Raw data, consisting of many partially overlapping image stacks, are stitched with a 2-step strategy: 1) mutual alignment between adjacent stacks is computed via cross-correlation;

	2) a global optimum position is found.
Technical specifications	The stitching software should minimize the I/O operations needed (which are usually the bottleneck of the process). We propose a sampling strategy where cross-correlations between adjacent stacks is computed only at given position along the stack depth.
Current solution	Name: ZetaStitcher
	URL to additional information: https://github.com/lens-biophotonics/ZetaStitcher
	Limitations: Some optimization of the code is still needed. Successfully running (quite fast) on lab workstations.

Processing station: ***2*****:** Image compression and downsampling	
Base information	General description of the processing: Raw data (image stacks) are processed to allow easier visualization and sharing. 1) Compression. Images are kept at the same resolution but heavily (>100X) compressed with lossy algorithms. Useful for sharing data at full resolution. 2) Downsampling. Images are downsampled by averaging groups of voxels (16x16x5) and collapsing each groups in a single voxel. Useful for data visualization.
Technical specifications	Compression and downsampling should make optimal use of I/O (e.g. by loading the data in suitable blocks). Ideally, one would need a fast compression method, working in 16 bit video stream. Indeed, video compression is well-suited to image stacks, where subsequent slices are highly correlated (as subsequent frames in a movie).
Current solution	Name: jpeg2000 compressor, embedded in custom software
	URL to additional information: https://jpeg.org/jpeg2000/
	Limitations: Available lossy video compressors works only with 8bit grayscale or RGB data. FFV1 can compress 16-bit grayscale video, but only lossless. To the best of our knowledge the best (and most supported) algorithm for 16-bit lossy compression is JPEG2000, which however works only on 2D images.

Processing station: ***3*****:** Neuronal soma detection	
Base information	General description of the processing: The method described by Frasconi et al. (Bioinformatics 2014) is used to localize the position of the soma of labeled cells. This algorithm is based on two steps: 1) Semantic deconvolution. A convolutional 3D deep neural network transform the original image into a 'cleaned' version, where soma appear (ideally) as homogeneously bright spheres, and all other structures (dendrites, axons) are (ideally) removed.

	2) Mean shift clustering applied on the deconvolved image to retrieve the centre of the bright spheroids.
Technical specifications	Step (1) needs training of the deep network on a ground truth set of about 1 GB raw data. Both training and prediction runs on GPUs. High-end GPU boards are needed (we fully use both a Pascal P100 and a GeForce GTX 1080). For prediction, we need nodes with one or two high-end GPUs (the higher the clock the better), and fast CPUs. Memory should be at least 128 GB per node.
Current solution	Name: Brain Cell Finder (bcfind)
	URL to additional information: http://bcfind.dinfo.unifi.it/ this is an older version of the software. The newer one will be released soon (hopefully before the end of the year).
	Limitations: prediction phase of step (1) and step (2) are inherently parallel, and can be easily distributed on multiple GPUs and cores, respectively. Training in step (1) can be made parallel, but we have never tested it. Also, the software has never been tested on HPC. Finally, the generalization properties of the approach (CNN trained in one sample and applied in others) need to be validated.

Processing station: ***4*****:** Neuronal segmentation	
Base information	General description of the processing: The method described by Mazzamuto et al. (LNCS 2018) is used to segment the soma of labeled cells. This algorithm is based on two steps: 1) Computation of a probabilistic heatmap using a 2D convolutional neural network. The heatmap represents the probability of each pixel of being part of a neuronal object. 2) Contour finding algorithm (marching squares) on the heatmap.
Technical specifications	Step (1) needs training of the deep network on a ground truth set of few hundreds GB of raw data, with data augmentation. Both training and prediction runs on GPUs. High-end GPU boards are needed (we fully use both a Pascal P100 and a GeForce GTX 1080). For prediction, we need nodes with one or two high-end GPUs (the higher the clock the better), and fast CPUs. Memory should be at least 128 GB per node.
Current solution	Name: Aliquis
	URL to additional information: http://www.bioretics.com/aliquis
	Limitations: Aliquis has been tested in HPC environment, but not our pipeline. As before, the inter-subject generalization properties of the method must be properly assessed.

Processing station: ***5*: Image registration to atlas**	
Base information	General description of the processing: Downsampled data are used for this step, and the transformations found are virtually extended to the full-resolution data (without regenerating the full-res volume). First, we register the two halves of the mouse brain by means of an affine transformation. The parameters are optimized by maximizing normalized cross correlation. Second, the whole brain is registered to the Allen mouse brain template. Here, first an affine transformation and then a non-linear one is computed by maximizing cross-correlation and mutual information.
Technical specifications	Nonlinear methods usually require at least a memory 5-6 times larger than the dataset. Thus, we estimate about 128 GB per node. As for computing power, we have never tried parallel implementations of the algorithms. However, in the literature GPUs and multi-core architectures are used as well. We would like to use nodes equipped with high-end GPUs, and with at least 16 cores.
Current solution	Name: Advanced Normalization Tools (ANTs), NiftyReg
	URL to additional information: http://cmictig.cs.ucl.ac.uk/wiki/index.php/NiftyReg https://stnava.github.io/ANTs/
	Limitations: The optimization of affine transform is usually quite fast on a high-end workstation. On the other hand, non-linear one is quite slow (almost one day per dataset). We are aware of HPC testing of these tools, but we have never tried them.

21.4 Infrastructure requirements

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? Visualization • What is the expected typical duration of interactive sessions? 2 hours • What software stacks need to be available? Paraview • Is it possible to define memory capacity requirements? Order of magnitude: tens of GBs
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? Stitching, deep learning and spatial registration to Atlas

Virtual Machine Services	• Which parts of the workflow require such services? NA
Active Data Repositories	• Which parts of the workflow require such services? Visualization, stitching and deep learning
Archival Data Repositories	• Which parts of the workflow require such services? Raw and curated datasets
Data Mover Services	• Which parts of the workflow require such services? Visualization, stitching and deep learning
Data Transfer Services	• Which parts of the workflow require such services? Moving data from lab to HPC sites • Between which ICEI sites is data planned to be transferred? On demand, in case a user from a different ICEI site need to access the datasets • How much data is expected to be transferred per time unit? NA • How are transfer patterns expected to change over time? NA
Data Location Service	• Which parts of the workflow require such services? Curated datasets
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single side? NA
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? NA
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples: ○ Special accounts for running services ○ Needs for fine-granular control of access to data NA
User Support Services	• Are the specific foreseeable needs for user support services? NA

21.5 Use Case references

HBP CDP2 SGA1 review slides (Silvestri)

HBP SGA2 GA

22. Learning-to-learn (LTL) in a complex spiking network on HPC and Neuromorphic hardware interacting with NRP (#3)

Learning-to-learn (LTL) in a complex spiking network on HPC and Neuromorphic hardware interacting with NRP

Use Case Description and Specification

26-06-2018 Sandra Diaz,

Partners	Sandra Diaz
Institutions	Simlab
Principal	Prof. W. Maass
Investigators	Prof. K. Meier

Date	Version / Change
14-06-2018	(Wouter Klijn) Template Initialization
26-06-2018	(anonymous) Fill in of known information
20-08-2018	(Anne Carstensen) Editorial changes
02-09-2018	(Wouter Klijn) Add high prio questions
21-09-2018	(Sandra Diaz) Review and update on questions for specific information
25-09-2018	(Anne Carstensen) Integration of review comments and updates

22.1 Use Case Description

This use case covers all three SP9 SGA2 use cases.

22.1.1 SGA2-SP9-UC001 - Open loop run of a complex spiking network with input data, output data and network reconfiguration by learning

This is the prototype Use Case for a feed forward data analysis without a time-critical closed perception-action loop, like in SGA2-SP9-UC002. It corresponds to the basic way of using classical artificial neural networks (ANN), but transfers the concept to the HBP brain-inspired systems SpiNNaker and BrainScaleS, which are spiking networks. Use Cases of this type are directed towards an understanding of learning in spiking networks to exploit the potential benefits of this technology over traditional deep networks. Such benefits are energy efficiency, resilience and the ability for real-time or accelerated learning.

For this Use Case, a user has developed a spiking neural network architecture, which may also apply supervised, unsupervised or reinforcement learning mechanisms. The network receives spike-coded input data, either from recorded biological sensors, or

from generic databases with abstract data. Abstract data can originate from various sources in science, business or government. On a traditional computer, network simulation, and in particular learning, takes a long time. This may be prohibitive if the research includes extensive parameter scans for optimization purposes. The user can take advantage of the real-time capability of SpiNNaker or the acceleration factor of BrainScaleS to reduce run time, so that the effects of learning mechanisms and parameter variations can be studied. Since a user expects to submit many jobs with different parameters, the job submission process should be scripted. Output data consist of spike recordings, selected membrane traces and the learned network configurations for further interpretation.

22.1.2 SGA2-SP9-UC002 - Closed loop run of a complex spiking network with input data, output data and network reconfiguration by learning

This is a Use Case which exploits learning in timing critical closed perception-action loops. The workflow is fundamentally different from the one in SGA2-SP9-UC001, which has no such loop. Input and output happen via a simulated environment. It makes use of the specific hardware infrastructure of the neuromorphic computing platform, which allows for timing critical real-time or accelerated emulation of a network and the corresponding timing constrained simulation of an environment, sensors and actuators on a conventional computer are linked via a very low latency (µs) network.
A user plans to implement and evaluate large-scale networks of neural sensorimotor cortical areas in a closed-loop configuration, linking sensory processing to a behavioural output (active perception) and a reward. The goal is to study how the functional and encoding roles of diverse neuronal populations across areas vary in time and how they are connected to the intra- and inter-cortical dynamics. This time-varying encoding across cortical areas should be considered as the key underlying mechanism for both stimulus-encoding and perceptual behaviour, which have not been studied before.
On a traditional computer, network simulation, in particular learning, takes a long time. This may be prohibitive if the research includes extensive parameter scans for optimization purposes. The user can take advantage of the real-time capability of SpiNNaker or the acceleration factor of BrainScaleS to reduce run time, so that the effects of learning mechanisms and parameter variations can be studied. Since a user expects to submit many jobs with different parameters, the job submission process should be scripted. Output data consist of spike recordings, selected membrane traces, the learned network configurations and the changes to the simulated environment during the closed-loop operation.

22.1.3 SGA2-SP9-UC003 - Learning-to-learn (LTL) in a complex spiking network with input data, output data and network reconfiguration by learning

This is an ambitious Use Case with a strong research component and the potential for a very high impact in basic research and applications of biologically-inspired machine learning. It goes one step beyond SGA2-SP9-UC002, in the sense that it runs through a

second loop for the optimization of the neuromorphic system parameters to achieve optimal learning capabilities.
Traditional learning approaches start from a predetermined network architecture and adjust the synaptic connection strengths by established learning algorithms, mostly based on a gradient-descent method. Neural networks in biological brains are the result of a very long evolutionary process, which provided them with the ability to learn. This ability is based on a multitude of parameters, including the network architecture and size, the parameters of neurons and synapses, and, of course, the synaptic connection strength. The result of evolution is a high degree of variability in those parameters which cannot be tuned by traditional learning approaches.
The learning-to-learn approach follows a double-loop strategy. An inner loop made of a neuromorphic circuit and a simulated environment, observed by sensors and modified by actuators running in a timing constraint fashion like SP9 Use Case 2. An outer loop runs an optimization algorithm to tune the network parameters to achieve optimal learning in the inner loop. The outer loop has no latency constraints, but requires fast execution of the optimization algorithm. The outer loop is ideally suited to run on an external computer, like the facilities provided by SP7.
On a traditional computer, inner loop simulation, and in particular the learning process, takes a long time. This may be prohibitive if the outer loop includes extensive parameter scans for optimization purposes. The user can take advantage of the real-time capability of SpiNNaker or the acceleration factor of BrainScaleS to reduce run time, so that the effects of learning mechanisms and parameter variations can be studied. Since a user expects to submit many jobs with different parameters, the job submission process should be scripted. Output data consist of spike recordings, selected membrane traces and the learned network configurations.

Technical details

Preconditions

1) The spiking neural network model and the experiment descriptions are written using the PyNN language and are in separate Python files in a Git repository or an HBP storage repository.
2) The learning rules are either part of the PyNN network description (e.g. STP or STDP configurations), or coded on an external control computer.
3) The data files to be processed by the network experiment have to be part of the repository. The data files need to be spike-coded in order to be used as inputs to the spiking network.
4) Registration and time allocation at the SP9 neuromorphic computing machines.
5) Participation at an initial training event offered by SP9 is recommended.

HBP infrastructure required for this Use Case

Infrastructure identical to SGA2-SP9-UC002: Availability of the SpiNNaker and/or BrainScaleS neuromorphic computing machines (generation 1 full-size or generation 2 prototypes) and their support software.

The Use Case also uses data storage and analysis facilities from SP7. A local compute cluster with a small physical distance and a network latency of the order of µs is required. Clusters with these requirements are available at the HBP neuromorphic machines.

In addition, a compute cluster for the optimization of metaparameters of the neuromorphic network is required. The latency of this machine is not critical and can be in the ms range. HBP HPC machine at distances >100km from the actual neuromorphic experiment can be considered. This would be one of the first applications for interactive supercomputing.

Workflows
Identical to SGA2-SP9-UC002: Potential users are computational neuroscientists, theoretical neuroscientists, data scientists or computer scientists.

1) Write a PyNN script for the network model.
2) Recode input data to a spike format using a coding scheme (e.g. rate coding).
3) If required, code the learning algorithm to run on the local computer attached to the neuromorphic machine.
4) Produce a job request script including the name of the system (SpiNNaker or BrainScaleS), a Collab ID, the URL of the Git repository, the path to the main script within the repository, and the list of arguments (parameter file name, etc.) required by the script.
5) After submitting the job request, the script receives a URL that returns a document indicating the job status.
6) The script polls the job status URL repeatedly until the job is complete, at which point the job status document contains the URLs of the output data files and the log file.
7) The script downloads the output data files and saves them to the local disk.
8) Save the network parameters as a result of the learning process.

Outputs delivered by the experiment
1) Simulation spike recordings.
2) Membrane traces for selected neurons.
3) Learned network configuration files for further analysis.

In addition, software for the learning optimization (like genetic algorithms) is required.
Outputs delivered by the experiment 003:
- Simulation spike recordings.
- Membrane traces for selected neurons.
- Learned network configuration files for further analysis.
- Neuromorphic network metaparameters optimized for learning.
- Readout monitoring the action performed on the simulated environment.

Outputs delivered by the experiment 002:
In addition, the closed loop software comprising sensor, actor and environment have to be set-up and run on the compute cluster available at the neuromorphic partner site.

Additional data exchange to a supercomputing site could be beneficial for processes evolving over longer time-scales.

- Simulation spike recordings.
- Membrane traces for selected neurons.
- Learned network configuration files for further analysis.
- Readout, monitoring the action performed on the simulated environment.

Resources required:

- Generation 1 SpiNNaker and/or BrainScaleS machine.
- Possibly generation 2 SpiNNaker and/or BrainScaleS machine/chip
- Possibly small scale SpiNNaker or Spikey systems
- Remotely accessible Neuromorphic Job Execution Service with access to GitHub to download external simulation files on behalf of users.
- Storage for spike reports (Output 2.2.1) -> 20 GB/simulation * 3000 simulations per study.

22.2 Diagrams

The SGA2 use cases for SP9 are all embedded in the third use case. Only a single diagram is need to capture all moving parts. Additional diagrams can be added for additional clarity or to explained detailed interactions. NMH to NRP interactions would be a potential diagram. Also a NMH running in England with outer loop optimization in an HPC centre.

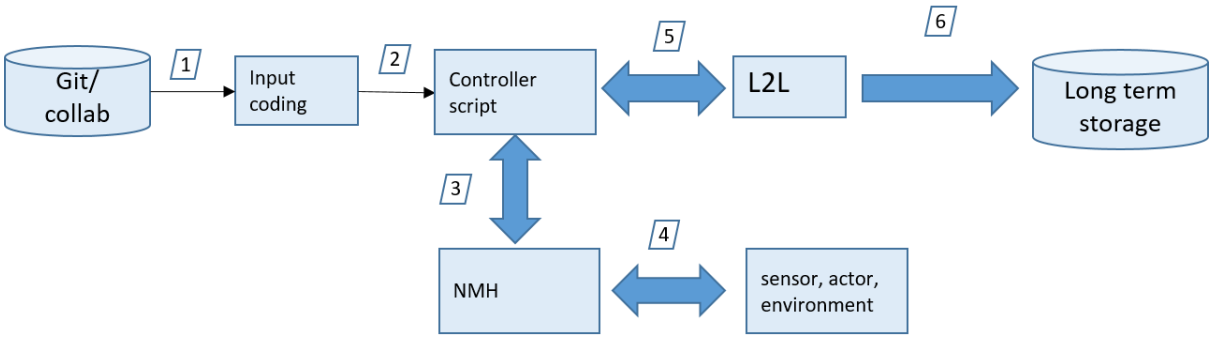


Figure 46

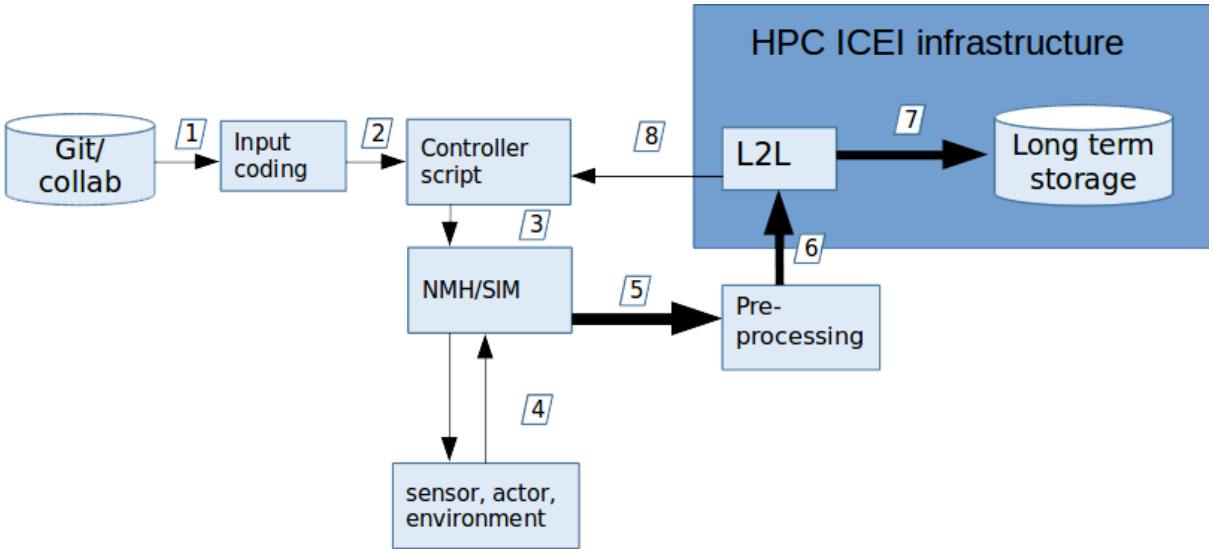


Figure 47: Diagram of the L2L use case. Information about the configuration and deployment of the experiments stored in a git or collab repository is used as input to start a controller script. This script interacts with simulations running on CPUs or neuromorphic hardware which send and receive information from the virtual/real environment. Results from the simulations running in the NMH/SIM are pre-processed and then sent to the outer loop L2L algorithm running on HPC. This algorithm evaluates the fitness of the simulations and produces new configurations to be run in a next iteration of fitting. The output of the simulations is sent to the long term storage, where it can be later retrieved/analysed/post-processed.

22.3 Node Characterization

22.3.1 Data objects

Data object: ***1***, PyNN network description	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***1***, Network input	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA Permanent (Forever): NA Additional information: NA

Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: **2**, Spike coded network input	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: **3**, Job scripts and experiment data	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA Includes data object 1 and 2
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***4***, Input/output between NMH and NRP	
Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***5***, System output to L2L system

Base information	General description of what data is stored: • Formats: NA • Metadata: NA • Database requirements: NA I suspect spikes and fitness?
Technical specifications	• Transient (Temporary): NA • Short-term (Campaign): NA • Permanent (Forever): NA Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: **6**, Data products for long term storage	
Base information	General description of what data is stored: Simulation spike recordings. Membrane traces for selected neurons. Learned network configuration files for further analysis. Neuromorphic network metaparameters optimized for learning. Readout monitoring the action performed on the simulated environment.
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

22.3.2 Data transport

Data transport: 1, Experimental inputs	
Base information	General description of what data is transported: Experimental parameters and network input
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 2, Spike encoded network input	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes): NA

Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 3, Job scripts and experimental parameters	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 4: NMH to NRP	
Base information	General description of what data is transported: Pre-processed data output from simulations on the NMH. Q: Is this an online process or batched? Online
	Data access patterns (request rate, transfer sizes): NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: Q: Is there an existing example of this connection? Or is this new functionality? There might be already a non-optimal solution for this.
	URL to additional information: NA
	Limitation: NA

Data transport: 5: Controller to L2l (JUPEX?)	
Base information	General description of what data is transported: NA
	Data access patterns (request rate, transfer sizes) : NA
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: 6: L2l to long term storage	
Base information	General description of what data is transported: Output data from best individuals per generation Checkpointing data to resume simulations: Q: Could you clarify: check pointed runs in Long term storage? Are there procedures in place to assure the backward compatibility of the checkpoints? By storing check points in the long term storage one could use these as new start points for divergent simulations/analysis. No procedures in place to assure backward compatibility. At this point there is only basic checkpointing.
	Data access patterns (request rate, transfer sizes) : NA
Technical specifications	Maximum required bandwidth: between 10 and 100MB/s
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

22.3.3 Data ingest / GUI

Data ingest: Git / Collab Q: Is this online? Do the hpc machines need web access to the outside world? Q: local mirror an acceptable solution?	
Base information	Description of input data source: Local mirror is an acceptable solution.
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

22.3.4 Data repository

Data repository: Long term storage	
Base information	Classification of the data objects (see below): NA
	Access control requirements: NA
	Access requirements: NA
	Data availability requirements: NA
Technical	Maximum and average capacity requirements: In the current

specifications	compute time proposal we have requested 10TiB of permanent storage, 5TB of temporary storage for simulations involving long training sessions with tensorflow and also simulations with neuromorphic hardware in the loop.
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: each training takes approximately 10k runs with 100 individuals, each producing 1-10 files per run, depending on the application. It is expected that several of these runs will be executed in parallel. Not all data will be preserved permanently but some intermediate storage is required for post processing. Gzipping is a solution. Q: The number of files is big and can be problematic in HPC filesystems. Would gzipping per generation be a solution?
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

22.3.5 Processing stations

Processing station: Input coding	
Base information	General description of data processing: NA
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Controller script (Jupex)	
Base information	General description of data processing: NA
	Typical processing steps: NA

	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: L2L	
Base information	General description of data processing: NA
	Typical processing steps: Simulation, assessment of fitness and generation of new individuals per generation
	Number of processing steps: 10k x 100 per training run. Q How many runs are expected? 20-50
Technical specifications	Data processing hardware architecture requirements: Q; NA or no special requirements? No special requirements. This should be able to run on standard CPUs.
	Required software stacks (libraries, software frameworks etc.) Python JUBE Unicore • Version requirements and dependencies: NA • Need for licenses: No
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: NMH : BrainScaleS Special purpose hardware outside of the ICEI procurement process.	
Base information	General description of data processing: NA
	Typical processing steps: • Receive data from compute node/site (e.g. input data)

	• Experiment setup (configuring hardware, preparing input data) • Experiment runtime (provide input data, receive output data) • Experiment teardown (read-out remaining recorded data) • Transmit data to compute node/site (for analysis)
	Number of processing steps: Depends on the experiment. Between 1 and a very large number (e.g. the inner-loop could be implemented locally).
Technical specifications	Data processing hardware architecture requirements: The BrainScaleS software interfaces accept input data in proprietary data formats and generate proprietary output data formats. All data formats are accessible using open-source libraries on the group's github repository mirror site: https://github.com/electronicvisions/
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses The NMH software stack depends on: • System C Compiler to build gcc >=7.2 • System Python installation to execute spack (https://github.com/spack/spack/) • Spack packages: autoconf, automake, bazel, gccxml, gsl, intel-tbb, libelf, liblockfile, npm, pkg-config, py-cartopy, py-lxml, py-mock, texinfo, xerces-c, binutils+gold+plugins, vim, emacs ~X, tmux, ncdu, units, ranger, py-ranger, mosh, mercurial, git, git-review, py-git-review, cmake, doxygen, doxygen+graphviz, bear, rtags, cppcheck +htmlreport, ffmpeg, gdb, llvm, genpybind, node-js, openssh, emscripten, boost@1.66.0+graph+icu+mpi+python+numpy, yaml-cpp+shared, tensorflow, log4cxx, googletest, googletest +gmock, gflags, cereal, py-pybind11@2.2.0:, py-bokeh, py-pygtk, gtkplus, cairo+X, py-pyside, py-slurm-pipeline, nest@2.2.2+python, py-brian, py-brian2, py-elephant, py-pynn@0.7.5, python, py-cython, py-pip, py-pylint, py-ipython, py-virtualenv, py-matplotlib~tk+qt+ipython, py-numpy, py-pandas@0.19.0:, py-pytables@3.3.0:, py-scipy, py-scikit-image, py-seaborn, py-sympy, py-statsmodels, py-lmfit, py-symfit, py-sqlalchemy, py-pyyaml, py-autopep8, py-flake8, py-jedi, py-sphinx, py-doxypy, py-nose, py-junit-xml, py-xmlrunner, py-pytest, py-pytest-xdist, py-line-profiler, py-attrs, py-setuptools, py-tabulate, py-html, py-html5lib, py-pillow (cf. our meta spack package https://github.com/electronicvisions/spack/blob/20180129/var/spack/repos/builtin/packages/visionary-defaults/package.py)

	Ratio of data processing rate versus data consumption and production rate: The BrainScaleS-1 system consists of 20 Wafer each accepting up to 48GBit/s and providing up to 48GBit/s. The data processing rate depends on exact type of experiment analysis code.
	Variability, availability, bandwidth and latency: Data consumption access pattern: Sequential/burst transfer; <40GbE Data production access pattern: Sequential/burst transfer; <40GbE Data processing access pattern: Random access Experiment latency: the experiment execution rate depends on the type of reconfiguration and the duration of an experiment; synapse weight updates require O(10us), analog neuron parameter requires O(1s).
	Additional information: NA
Current solution	Name: Site-local compute cluster and data storage system
	URL to additional information: NA
	Limitation: Data storage capacity, throughput as well as compute power.

Processing station: NMH : SpiNNaker	
Base information	General description of data processing: A PyNN script is received and converted into a SpiNNaker network, which is then run on the SpiNNaker machine, with results being extracted post-execution.
	Typical processing steps: • RemoteSpiNNaker server receives PyNN job • RemoteSpiNNaker server starts virtual machine (VM) to run job • VM starts executing RemoteSpiNNaker client • RemoteSpiNNaker client reads PyNN script from server • RemoteSpiNNaker client starts sPyNNaker • sPyNNaker maps PyNN network description to available SpiNNaker hardware • sPyNNaker loads converted PyNN network on the SpiNNaker hardware • sPyNNaker waits for job to complete on SpiNNaker hardware (sleep) • sPyNNaker reads requested results from the network • RemoteSpiNNaker client pushes results to RemoteSpiNNaker server • Job results are read from RemoteSpiNNaker server data storage
	Number of processing steps: 11
Technical	Data processing hardware architecture requirements:

specifications	RemoteSpiNNaker server currently runs on a VM with 2 Intel Xeon CPUs, 2GB RAM, 100GB system disk and 2TB results storage disk. Each VM is currently set up with an Intel Xeon CPU, 32GB RAM, 100GB disk
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses All software is provided open source, and no licenses are required. RemoteSpiNNaker server requires Oracle Java >= 1.8 with Maven >= 3.5 installed. Other libraries are automatically provided via Maven. The VM system runs XenServer 7. RemoteSpiNNaker client requires Oracle Java >= 1.8 installed on a VM template, along with scripts to start and execute the client, which should be loaded on to the VM and executed when it starts. sPyNNaker requires CPython 2.7 and pip >= 9. Other libraries are automatically provided via pip.
	Ratio of data processing rate versus data consumption and production rate: The data consumption rate depends on the size of the PyNN network to simulate. Generally, a PyNN description is quite short in length, so will take very little time to transfer to the machine. The amount of data processing depends on the number of neurons and synapses described by the network, as well as the run duration specified. Each SpiNNaker board can produce 100Mbit/s of data, though only 40Mbit/s is achievable with the current software. There are 600 boards available; the network size determines how many of these boards are used, and thus the data production rate achievable in total.
	Variability, availability, bandwidth and latency: Data consumption access pattern Whole PyNN script is read before any data processing takes place. Spike input data can be read during script processing. Script execution can be split across multiple cycles; spike input data will be read at the start of each cycle, and the consumption will be determined by the number of spikes in that cycle. Data production access pattern Data consisting of anything up to the total SDRAM of the machine can be produced in each cycle of execution. Data is transferred in bulk at the end of the execution cycle.
	Additional information: NA
Current solution	Name: RemoteSpiNNaker / sPyNNaker on local VM server
	URL to additional information: https://spinnakermanchester.github.io
	Limitation: Data storage and mapping and data generation compute required

Processing station: SIM : NEST Q: How big are the networks? Q: Single node jobs or multi node? Q: Estimation of core hours consumed per year.	
Base information	General description of data processing: For the simulations of biological relevance, networks of about 3 Million neurons would be acceptable. About 2M core hours in a system like Jureca. The models could be also ported and simplified to be run in Tensorflow, where the number of neurons would be reduced to 10,000 and we would expect to use about 1M core hours in a system like JUWELS. They require multi-node jobs.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: Given that also several of the current L2L applications require GPUs, it would be desirable to have access to nodes with fast GPUs which also have fast access to local memory (2-4Tb SSD) for storing training datasets.
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: sensor, actor, environment: NRP The NRP is subject of a separate ICEI use-case. The processing details can be found there: “Neurorobotics Platform, large-scale brain simulations (#11)”	
Base information	General description of data processing: NA
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies: NA • Need for licenses: NA

	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern: NA Data production access pattern: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

22.3.6 Components running on or communicating with ICEI Infrastructure. Reference to the second diagram.

Data transport: **6,** Pre-processing unit (local to NMH facilities) to HPC centre	
Base information	General description of what data is transported: Spiking data produced by each instance in the parameter space to be explored. This data has been pre-processed in the local cluster.
	Data access patterns (request rate, transfer sizes): ~ 1GByte of data per instance. We can have hundreds of parallel instances and up to tenths of thousands of runs.
Technical specifications	Maximum required bandwidth: 8GByte/s
	Average required bandwidth: 1GByte/s
	Interface requirements for attached entities: NA
	Additional information: This is the most complicated part of the data flow, as it involves moving data from the local NMH facilities to the HPC facilities. A data transport service solution should manage this link. The information flow can have delays and is not critical to the performance of each instance, however the more delays added the less time performant the whole parameter exploration will be. This is the main bottleneck for enabling a LTL loop with the outer loop being calculated in an HPC centre and the inner loop in the NMH facilities.
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **6**, L2l to long term storage	
Base information	General description of what data is transported: Spiking data, fitness calculation results.
	Data access patterns (request rate, transfer sizes):

	~ 1 GByte of data per instance We can have hundreds of parallel instances and up to tenths of thousands of runs.
Technical specifications	Maximum required bandwidth: 1 GByte
	Average required bandwidth: 1 GByte
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: **6**, L2l to NMH controller	
Base information	General description of what data is transported: New parameters for the next n individual parallel run
	Data access patterns (request rate, transfer sizes): <250Mb
Technical specifications	Maximum required bandwidth: 1GByte
	Average required bandwidth: 1GByte
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name
	URL to additional information: NA
	Limitation: NA

Processing station: L2L	
Base information	General description of data processing: L2L will take care of gathering the spiking or output data from individual simulations, calculate the fitness and then calculate a new set of parameters to be explored by individual instances running on the NMH. The fitness calculation is expected to require per instance to be analysed: 1GByte RAM, ~ 0.5 core/hours (6 cores for 5 minutes running on a CPU like the compute nodes in JURECA). In order to process 100 individuals in parallel, one would require (using as base again JURECA nodes) 50 core hours per iteration and minimum 25 nodes. In order to process a whole training of 10,000 iterations, the total core hours required would be 500,000. The assessment of the fitness and generation of new parameters is relatively less computationally expensive.
	Typical processing steps:

	1.Fitness calculation 2.New parameter set calculation
	Number of processing steps: 2
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.): • Python scripts • L2L + JUBE (Jupex) framework
	Ratio of data processing rate versus data consumption and production rate: Between 6-1 and 15-1
	Variability, availability, bandwidth and latency: Data consumption access pattern sequential: NA Data production access pattern sequential: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

22.4 Infrastructure requirements

Infrastructure service	**Questions to address**

Interactive Computing Services	• Which parts of the workflow require such services?
	  ○ Neuromorphic hardware can be used as an accelerator for spiking neural networks. Exploiting the accelerated emulation of such networks, the BrainScaleS platform benefits from “joint scheduling” of experiment preparation, execution and analysis steps. Non-interactive/batch scheduling introduces additional delays between each step which reduces the overall speed-advantage of accelerated neuromorphic hardware.
	• What is the expected typical duration of interactive sessions?
	  ○ O(4h)
	• What software stacks need to be available?
	  ○ As we want to stream data between both compute sites, all software defined in table “Processing station: NMH” is required.
	• Is it possible to define memory capacity requirements?
	  ○ Ignoring possible experiment-specific additional requirements, the software stack requires at least 16GiB per node (64GiB would be better).

(Elastic) Scalable Computing Services	• Which parts of the workflow require such services?
	  ○ The input data generation and analysis steps.
	  ○ Elastic scaling to an arbitrary amount of wafers is currently not supported for the Heidelberg system. The currently installed hardware platform consists of 20 wafer modules.

Virtual Machine Services	• Which parts of the workflow require such services?
	  ○ Software deployment via singularity would be beneficial, but a flat spack-based software installation is possible.
	  ○ The NMH compute site uses singularity containers for software deployment.

Active Data Repositories	• Which parts of the workflow require such services? ○ None
Archival Data Repositories	• Which parts of the workflow require such services? ○ Analysis and (depending on experiment requirements) generated input data
Data Mover Services	• Which parts of the workflow require such services? ○ None

Data Transfer Services	• Which parts of the workflow require such services?
	  ○ The BrainScaleS platform consumes and generates large data volumes during experiment phasis. The input data has to be transferred to the BrainScaleS system before the runtime phase, the output has to be transferred to a compute site after the runtime phase.
	• Between which ICEI sites is data planned to be transferred?
	  ○ cf. above; HTTP transfers are always an option, but optimized inter-site transfers (orchestrated via UNICORE) are beneficial.
	• How much data is expected to be transferred per time unit?
	  ○ Short O(1s) bursts of experiment data can reach 40GbE wire-speed; the repetition rate depends on the experiment-defined reconfiguration speed; typically in the second to minute range. Higher repetition rates are possible, but imply a lower data rate.
	• How are transfer patterns expected to change over time?
	  ○ We expect experiments to increase in size and duration.

Data Location Service	• Which parts of the workflow require such services? ○ None?
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single side? ○ No, reduced bandwidth does not invalidate experiments but reduces overall job/experiment throughput. However, 10GbE are typically required for reasonable performance.
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? ○ Same as internally.

Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples:
	  ○ Special accounts for running services
	  ○ Needs for fine-granular control of access to data
	  ○ The neuromorphic platform uses the HBP account/authentication system.
	  ○ Export control rules apply to the BrainScaleS platform.

User Support Services	• Are the specific foreseeable needs for user support services? ○ HLST?

22.5 Use Case references

HBP SGA2 grant agreement

23. Multi-scale co-simulation: Connecting Arbor, NEST and TVB to simulate the brain (#10)

Multi-scale co-simulation: Connecting Arbor, NEST and TVB to simulate the brain

Use Case Description and Specification

03-07-2017 Wouter Klijn, Jennifer Sarah Goldman

Partners	Jennifer Sarah Goldman Wouter Klijn Alex Peyser Sandra Diaz
Institutions	
Principal Investigators	Morrison, Destexhe, Diesmann, Jirsa

Date	Version / Change
29-06-2018	(Wouter Klijn) Initial write-up
02-07-2018	(Jennifer Goldman) rewrite
03-07-2018	(Wouter Klijn) Process review comments
25-09-2018	(Wouter Klijn) Next iteration, completion of the template. New diagram.
27-09-2018	(Jennifer Goldman & Wouter Klijn) Clarifications and grammatical corrections
04-10-2018	(Viktor Jirsa) Correction on upper limit TVB size
05-10-2018	(Wouter Klijn) Translate new upper limit into resource requirements

23.1 Use Case Description

What are useful abstractions to understand the brain? What level of detail is needed to simulate cognition? This use case provides a testbed for answering these questions by integrating HBP supported and developed simulators in a single neuroscientific system: Morphologically detailed neurons simulated in Arbor or Neuron, spiking neurons networks in NEST, and large scale whole brain models in TVB. Each platform has produced insight at specific spatial scales, but their incorporation has been limited due to different resource requirements of each simulator.

The aim is to incorporate existing infrastructure to assess the contribution of neural mechanisms across scales to the generation of neural signals and ultimately cognition. This aim will be pursued with a multi-scale model: Populations of morphologically detailed neurons are simulated, embedded in a brain area of spiking neurons,

communicating with connected brain regions through TVB, and producing brain imaging signals commonly used in clinic and research (iEEG, EEG, MEG, fMRI), see 1. In this way, neural signals from spikes to local field potentials and macroscopic brain signals can be examined across scales and empirical modalities.

Two sets of models can be distinguished when focusing at resource requirements, differing in the manner the morphologically detailed and spiking neuron models are related to each other. The TVB model details are typically not changed between these two model sets. Although TVB modelling is essential, from a resource usage perspective it requires trivial amounts compared to NEST and Arbor/Neuron.
In the first set of models, the areas simulated by spiking and morphologically detailed neurons are spatially discrete. The different simulators are modelling spatially separate parts of the brain. The result of this is that the majority of the spikes generated in each simulator will only need to be transported within the same simulator. Only the subset of neurons connected between the scales will have spikes transported to the different simulator. This relatively small interface allows the three simulators to be located on differently optimized physical systems (in the same MPI network). E.g. a large memory system with high bandwidth interconnect for NEST and a many-core booster system for Arbor/Neuron.

The second set of models, spatially mixed, models would see morphologically detailed neurons embedded into the spiking network directly. The neurons at the different scales are potentially fully connected. This results in a high bandwidth demand between the two simulators. This type of system can run ideally on fat GPU nodes. Nodes with both a high performance multi-core CPU, large memory and a many-core accelerator. Co-location of these two simulators on one node will make optimal use of all resources. For both model sets, the resource requirements for TVB are modest.

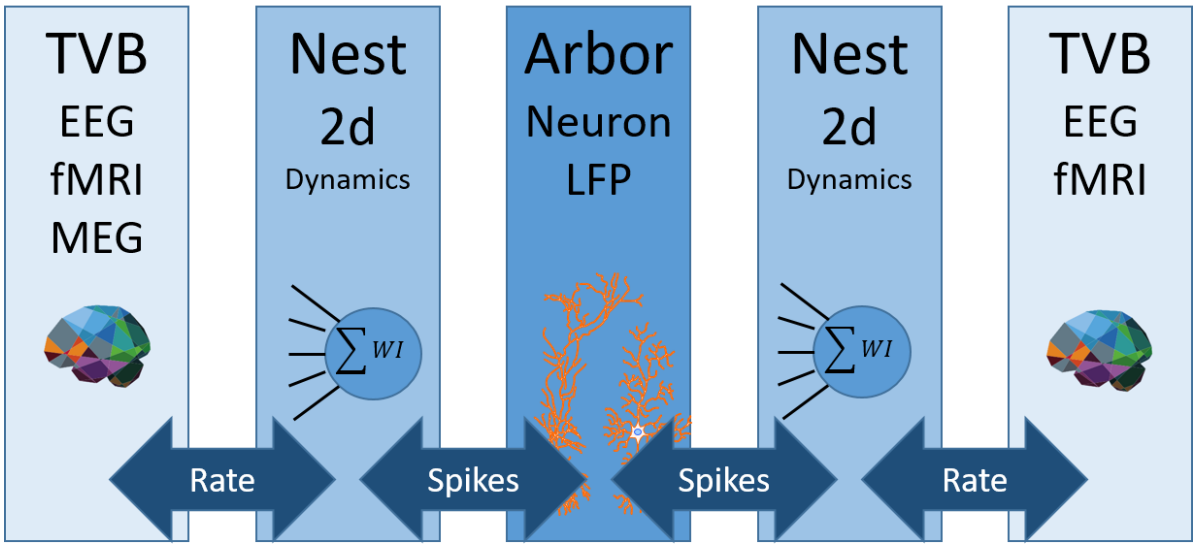


Figure 48: Spatially discrete multi-scale co-simulation models. The different simulators are modelling separate brain locations. The simulators now only have to communicate at the boundaries.

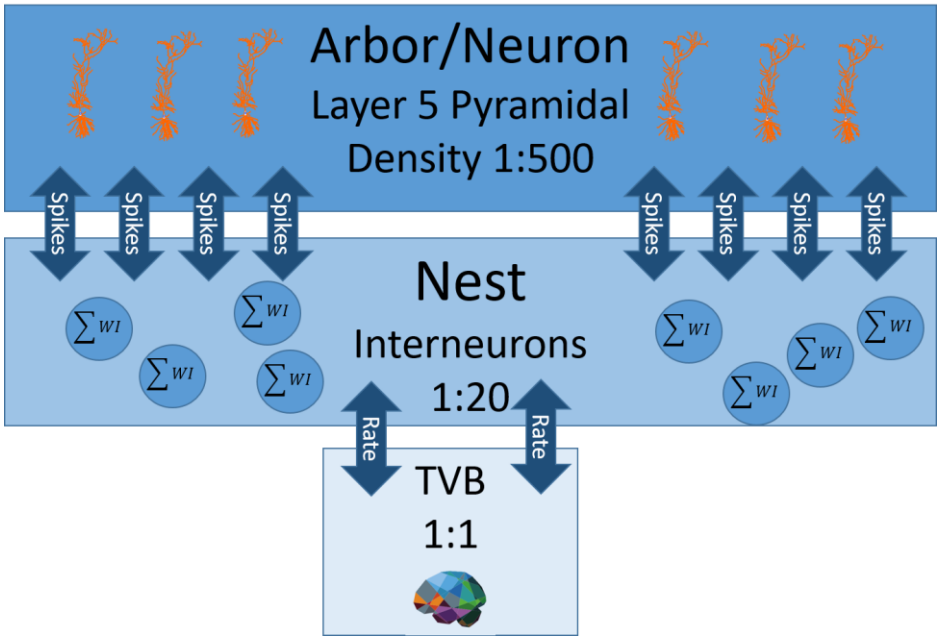


Figure 49: Fully overlapping or even mixed multi-scale co-simulation models would see potentially all neurons communicating, as if the neurons are located in a single mixed scale model.

23.2 Diagrams

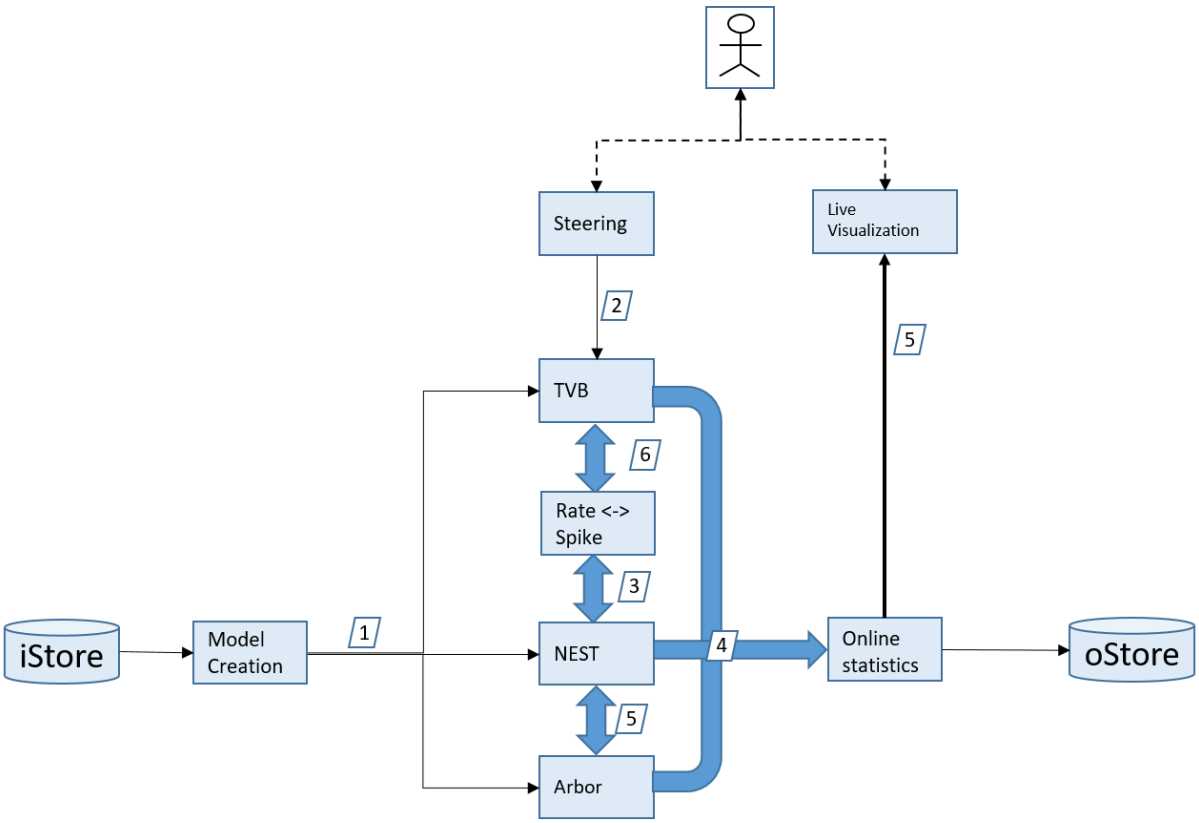


Figure 50: High level overview of a TVB / NEST / Arbor co-simulation run.

23.3 Node Characterization

In this section a characterization of each component is depicted in the annotated use case diagram. This is done in a table format with typical information points listed. The entries are typically split in different sets: The ***base*** information set without which an informed discussion might be complicated; The description is typically at a user / functional level. Secondly, **technical specifications** of the requirements. The use case is not yet solved thus this information will by necessity be added incrementally and optionally by a domain specialist. The third information set is regarding ***current solutions*** that one is aware of.
Not all information might be available. Fill in what is known at this stage. Having a start point for a dialog is more important that having perfect information, especially in the beginning stages.

For ICEI the following set of requirements are important. Any information that might inform this is appreciated:

- RAM: needed per node, in total
- IO: bandwidth, latency, always on/dedicated
- CPU: large size jobs / farming
- Specialized hardware: (GPU, KNL, FPGAs)
- Storage: size, access rate
- Specialized software: VM/containers
- Specialized features: in-situ visualization

Architecture Requirements:

- Minimal compute performance (excluding acceleration)
- Minimal volatile memory footprint of 192 GByte
- MPI point-to-point bandwidth of 10 GByte/s or higher
- MPI latency of 2 micro-seconds or less
- Access to active data repositories with a bandwidth of up to 8 GByte/s per node
- GPU requirements per node (minimum)
- GPU configuration (minimum HBM)

23.3.1 Data objects

Data object: ***1***, Model to simulator	
Base information	General description of what data is stored: Currently the model generation is performed as part of the simulation step. In the future this is expected to be performed in a separate application. Direct memory transfer would be the optimal choice for this communication.
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: Application specific
	URL to additional information: NA

	Limitations: NA

Data object: ***2***, Control signals from GUI front end to applications	
Base information	General description of what data is stored: ZeroMQ messages. Partly shared between application partly application specific.
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: ZeroMQ, Nett
	URL to additional information: https://hbp-hpc-platform.fz-juelich.de/?hbp_software=multi-view-framework
	Limitations: Not suitable for big data communication

Data object: ***3***, Nest to spike/rate transformer	
Base information	General description of what data is stored: Spike trains Pairs of spike time and gid
Technical specifications	• Short-term (Campaign): Data used throughout the execution of the scientific workflow. • Permanent (Forever): Data outliving the machine used to generate it. Additional information: https://gitlab.version.fz-juelich.de/nest_mpi_stream
Current solution	Name: Proof of concept unnamed
	URL to additional information: NA
	Limitations: One off non general proof of concept

Data object: ***4***, Simulator outputs to statistics engine	
Base information	General description of what data is stored: One directional potentially big data stream
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: No current solution exists
	URL to additional information: NA
	Limitations: NA

Data object: ***5***, Nest /Arbor	
Base information	General description of what data is stored: Pairs of GID and times Synchronisation signals

Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: This should be a low delay connections. Depending on the implementation this could be: Direct memory, when simulators are collocated on the node Or MPI, when in different nodes. A new transport stack is currently in development: https://gitlab.version.fz-juelich.de/nest_mpi_stream
Current solution	Name: MUSIC
	URL to additional information: https://github.com/INCF/MUSIC
	Limitations: Music takes control of the MPI word which complicates integration.

Data object: ***6***, spike/rate transformer to TVB	
Base information	General description of what data is stored: Voltage traces
Technical specifications	• Permanent (Forever): Data outliving the machine used to generate it. Additional information: Low bandwidth, max 400 * 10000Hz per simulated second voltage trace. Should all be stored, but is in the context of HPC trivial
Current solution	Name: NA
	URL to additional information: NA
	Limitations: NA

Data object: ***7***, statistics to online visualization	
Base information	General description of what data is stored: HBP in-situ pipeline data stream Data to be visualized live
Technical specifications	• Transient (Temporary): Data discarded on simulation completion or when later processing steps are concluded. Additional information: NA
Current solution	Name: HBP in-situ pipeline
	URL to additional information: https://hbp-hpc-platform.fz-juelich.de/?hbp_software=multi-view-framework
	Limitations: Connections are currently all implemented with zeroMQ messages. This might not be fast enough Alternate solutions would be screen casting of application running on visualization cluster.

23.3.2 Data transport

Data transport: ***1***, Model to simulator	
Base information	General description of what data is transported: Neuronal and circuit models
	Data access patterns (request rate, transfer sizes): Currently in memory per simulator
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: ***2***, Steering commands from GUI to simulators	
Base information	General description of what data is transported: ZeroMQ commands
	Data access patterns (request rate, transfer sizes): 200 / sec (in case of continues/ analogue controls) Start of simulation might see peaks of 1000s of messages Size is typically in the 100s of Bytes
Technical specifications	Maximum required bandwidth: Trivial bandwidth
	Average required bandwidth: Trivial bandwidth
	Interface requirements for attached entities: Tunnelling from the outside. Low latency is needed for usability and prevention of oversteering
	Additional information: NA
Current solution	Name: HBP in-situ steering
	URL to additional information https://hbp-hpc-platform.fz-juelich.de/?hbp_software=multi-view-framework
	Limitation: Not all simulators are connected

Data transport: ***3***, Nest to spike train to rate translation	
Base information	General description of what data is transported: Spikes
	Data access patterns (request rate, transfer sizes): Continues during the simulation, might be big data
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: MPI
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data transport: ***4***, Simulator to Online statistics	
Base information	General description of what data is transported: Spikes and other neuron measures
	Data access patterns (request rate, transfer sizes): Continuous stream of data
Technical specifications	Maximum required bandwidth: Up to MPI max bandwidth
	Average required bandwidth: Half of the MPI bandwidth
	Interface requirements for attached entities: MPI
	Additional information: We are currently building a MPI / Conduit alternative
Current solution	Name: MUSIC
	URL to additional information: https://github.com/INCF/MUSIC
	Limitation: Untested for 80% of the connections. Only meant for simulator connections

Data transport: ***5***, NEST / Arbor interconnect	
Base information	General description of what data is transported: Spikes
	Data access patterns (request rate, transfer sizes): High continues rate. Possible all to all
Technical specifications	Maximum required bandwidth: Maximum MPI bandwidth (NEST is bandwidth limited at extreme scales)
	Average required bandwidth: NA
	Interface requirements for attached entities: MPI
	Additional information: MPI transport stack in development https://gitlab.version.fz-juelich.de/nest_mpi_stream Purpose build for this communication
Current solution	Name: POC
	URL to additional information: NA
	Limitation: NA

Data transport: ***6***, TVB to translator	
Base information	General description of what data is transported Voltages / rate traces
	Data access patterns (request rate, transfer sizes): Continuous but max 420000 * 10000hz per simulated second This should be well below 1GBit/Sec

Technical specifications	Maximum required bandwidth: Trivial
	Average required bandwidth: Trivial
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: Exists as POC
	URL to additional information: NA
	Limitation: It is a POC

Data transport: ***7***, HBP in-situ visualisation pipelines	
Base information	General description of what data is transported: Raw or post processed simulation or analysis results
	Data access patterns (request rate, transfer sizes): Continuous output
Technical specifications	Maximum required bandwidth: 5Mbyte / sec when screen casting per user
	Average required bandwidth: We expect structured data transport thus 1 Mbyte / Sec per user
	Interface requirements for attached entities: NA
	Additional information: NA
Current solution	Name: HBP in situ pipeline
	URL to additional information: https://hbp-hpc-platform.fz-juelich.de/?hbp_software=multi-view-framework
	Limitation: No limitations: platform is written with exactly this use case in mind.

Data transport: ***8***, Transport to longterm storage	
Base information	General description of what data is transported: Simulation output, analytic results, instantiated models
	Data access patterns (request rate, transfer sizes): Once at the end of the processing chain
Technical specifications	Maximum required bandwidth: NA
	Average required bandwidth: NA
	Interface requirements for attached entities: NA
	Additional information: This can be a staged process, no bandwidth requirements
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

23.3.3 Data ingest / GUI

Data ingest: Steering	
Base	Description of input data source

information	User control actions. Potentially both manual or scripted
	Description of data introduction (upload? scanner characteristics? simulation characteristics?) NA
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports Format: ZeroMQ msg Loads: Very low Bandwidth: Very low Latencies: Below 20ms (allows for reactive user interface.) Transport: ZeroMQ
	Additional information
Current solution	Name: HBP in- situ pipeline
	URL to additional information: https://hbp-hpc-platform.fz-juelich.de/?hbp_software=multi-view-framework
	Limitation: This use case is a major driver for features

Data ingest: In situ-visualization	
Base information	Description of input data source: User control actions
	Description of data introduction (upload? scanner characteristics? simulation characteristics?): User control actions
Technical specifications	Characteristics of data: formats, loads, bandwidths, latencies, transports: Format: ZeroMQ msg Loads: Very low Bandwidth: Very low Latencies: Below 20ms (allows for reactive user interface.) Transport: ZeroMQ Demands of visualization itself (such as visualizing 1,000 morphologically detailed neurons) can be handled with desktop systems with a moderate amount of memory. However, the post- processing leading to such visualization is highly dependent on the details of the post-processing, but is usually flexible to trade- offs between different levels of the memory hierarchy.
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

23.3.4 Data repository

Data repository: iStore	
Base information	Classification of the data objects (see below): Network and neuron models
	Access control requirements: Public, probably internet accessible (modelDB)
	Access requirements: Public, probably internet accessible (modelDB)
	Data availability requirements: NA
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. Should grow slowly over time.
	In terms of size & file number: NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Data repository: oStore	
Base information	Classification of the data objects (see below): Simulation output, analytic results, instantiated models
	Access control requirements: Embargo for first year, after that fully public
	Access requirements: Internet / public (provenance tracking for publications)
	Data availability requirements: Long term storage: low availability. Tape would suffice
Technical specifications	Maximum and average capacity requirements: NA
	In case of repository for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time. NA
	In terms of size & file number: • up to 5TB/job, older files may be deleted if they become obsolete for the community The mean size of a simulation is several GByte, however, this can increase depending of the experiment performed. Single gzip file is acceptable
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA

	Limitation: NA

23.3.5 Processing stations

Processing station: Model Generation	
Base information	General description of data processing: Instantiation of a network from a statistical or structural model
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: TVB	
Base information	General description of data processing: Neural mass model based whole brain simulation TVB resolution can vary from 100 to 20000, where in the latter the regions are decomposed into numbers of connected vertex points. For 20000 vertices, the spatial resolution is about 3mm, which is approximately the maximum. Beyond that, the mean field approximations will start breaking down.
	Typical processing steps: Read 'teacher data' Generate network from description Simulate for set time
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: None Extremely low resource requirements compared to the other components in the network
	Required software stacks (libraries, software frameworks etc.)

	• Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: https://www.thevirtualbrain.org/tvb/zwei
	Limitation: NA

Processing station: Rate <-> Spike translation	
Base information	General description of data processing: TVB generates a rate based signal. Although this could be injected directly into NEST or Arbor it is theorized that this uncorrelated spike activity might introduce unwanted effects. A new design includes a pre-processing step in NEST. Worst case we might be looking at 1500 Neurons times 20000 brain areas to 'translate'. This would entail in 30E6 neurons to simulate. 20000completely disjoint simulations, at a minimum not fully connected. 15 normal Jureca nodes would be able to run this .2 second simulation time at 9 minutes clocktime, from comparable networks. The other way around would entail time / neuron averaging to produce a rate code. This is a trivial processing step.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: High data consumption and low generation -or- Low data consumption high data generation

	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: NEST	
Base information	General description of data processing: The main NEST simulation. Simulation of complete brain areas in spiking neuron resolution. Possible interesting research questions include 2d dynamics and large scale network dynamics. Sizes of the network might go up top 100E6 neurons or more. With realistic synaptic connectivity this would entail large multi node simulations. Upwards of 1000s of nodes. The bottleneck of computational demands for this class of use cases is the network simulation, using tools like NEST, Neuron, Arbor or TVB. To estimate the scale of the problem: the "record" NEST simulation on the K supercomputer in 2013 used approximately 1.1 PByte of memory while taking up most of a 10 PFlop/s supercomputer. To move from 10^9 neurons to 10^10 neurons using the current class of software architectures would take on the order of 10 PByte and 100 PFlop/s.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Arbor	
Base information	General description of data processing: For morphologically detailed simulations, initial estimates for "Arbor" indicate that peak performance for a GPU- based architecture occurs with 10k cells/GPU. Since 10,000 P100 GPUs is at the order of *50 PFlop/s*, to sustain such activity we would need 10,000 sockets * 10,000 cells/socket * (1~1000) kBytes/cells = *0.1~100 * 10^12 Bytes*, for 100 * 10^6 cells, which is on the order of the size of the human hippocampus. Other simulators may require more memory.
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

Processing station: Online statistics	
Base information	General description of data processing: ELEPHANT See relevant ICEI Use case #7
	Typical processing steps: NA
	Number of processing steps: NA
Technical specifications	Data processing hardware architecture requirements: NA
	Required software stacks (libraries, software frameworks etc.) • Version requirements and dependencies • Need for licenses NA
	Ratio of data processing rate versus data consumption and production rate: NA
	Variability, availability, bandwidth and latency: Data consumption access pattern

	Data production access pattern NA
	Additional information: NA
Current solution	Name: NA
	URL to additional information: NA
	Limitation: NA

23.4 Infrastructure requirements

Infrastructure service	**Questions to address**
Interactive Computing Services	• Which parts of the workflow require such services? • What is the expected typical duration of interactive sessions? • What software stacks need to be available? • Is it possible to define memory capacity requirements? None
(Elastic) Scalable Computing Services	• Which parts of the workflow require such services? The Elephant analysis and the visualization front end
Virtual Machine Services	• Which parts of the workflow require such services? NA
Active Data Repositories	• Which parts of the workflow require such services? Elephant Analysis might use fast active data storage
Archival Data Repositories	• Which parts of the workflow require such services? Output of the processing pipeline should be stored for archival
Data Mover Services	• Which parts of the workflow require such services? Moving the simulation results and data to the archive
Data Transfer Services	• Which parts of the workflow require such services? • Between which ICEI sites is data planned to be transferred? • How much data is expected to be transferred per time unit? • How are transfer patterns expected to change over time? None
Data Location Service	• Which parts of the workflow require such services? None
Internal interconnect	• Are there know minimal performance requirements to data transfer between e.g. ICEI infrastructure services at a single

	site? • The multi scale simulation would match with proposed modular supercomputing in Juelich. The online nature of the computations require that the different modular have a high speed interconnect
External interconnect	• Are there particular requirements with respect to network accessibility of platform or user services? None
Authentication / Authorization Services	• Are there specific requirements related to authentication and authorization? Examples: ○ Special accounts for running services ○ Needs for fine-granular control of access to data None
User Support Services	• Are the specific foreseeable needs for user support services? The pipeline is extremely complex. Advanced monitoring services are needed.

23.5 Discussion

23.5.1 Relevant write down High Level Support Team

A common workflow in neuronal network simulations involves the generation of connectivity, simulation of the resulting network and post-processing (analysis) of the results. This workflow may run on a user's computer, on a local cluster or use an HPC system, depending on the resource demands for each component of the workflow.
These resource demands are largely determined by the size, detail and complexity of the network in question. The simulation itself may be implemented using NEST, Neuron, Arbor, TVB, ug4 or other tools, or may be composed of even more complex pipelines involving several simulators and components. Each component may additionally include in situ visualization or interactive steering. Post-processing often involves comparison with real-world results such as Allen Institute electrophysiological data, live macaque experiments or SP1 brain atlas connectivity maps, leading to refinement of models and experiments in "productive" or "integrative loops".
Such workflows require multiple entry points: Collaboratory interfaces, complex specialized GUIs, and traditional command line interfaces, according to the complexity of the pipeline and expertise of the user. The traditional interface to complex workflows is a command line interface to a batch processing system such as SLURM using ad-hoc coupling between components; this support task will enable the use of common, reusable APIs and user-friendly interfaces in an HPC context.