



NeuronDepot: keeping your colleagues in sync by combining modern cloud storage services, the local file system, and simple web applications

Philipp L. Rautenberg^{1,2*}, Ajayrama Kumaraswamy¹, Alvaro Tejero-Cantero³, Christoph Doblender⁴, Mohammad R. Norouzi⁴, Kazuki Kai⁵, Hans-Arno Jacobsen⁴, Hiroyuki Ai⁵, Thomas Wachtler¹ and Hidetoshi Ikeno⁶

¹ Department of Biology II, G-Node, Ludwig-Maximilians-Universität München, Planegg-Martinsried, Germany

² Department for Innovations, Max Planck Digital Library, München, Germany

³ MRC ANU, Department of Pharmacology, University of Oxford, Oxford, UK

⁴ Department of Informatics, Technische Universität München, München, Germany

⁵ Department of Earth System Science, Fukuoka University, Fukuoka, Japan

⁶ School of Human Science and Environment, University of Hyogo, Hyogo, Japan

Edited by:

Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:

Daniel Gardner, Weill Cornell Medical College, USA

Kyle H. Ambert, Intel, USA

Gregory A. Book, Hartford Hospital, USA

*Correspondence:

Philipp L. Rautenberg, Department for Innovations, Max Planck Digital Library, Amalienstraße 33, 80799 München, Germany
e-mail: rautenberg@mpdl.mpg.de

Neuroscience today deals with a “data deluge” derived from the availability of high-throughput sensors of brain structure and brain activity, and increased computational resources for detailed simulations with complex output. We report here (1) a novel approach to data sharing between collaborating scientists that brings together file system tools and cloud technologies, (2) a service implementing this approach, called NeuronDepot, and (3) an example application of the service to a complex use case in the neurosciences. The main drivers for our approach are to facilitate collaborations with a transparent, automated data flow that shields scientists from having to learn new tools or data structuring paradigms. Using NeuronDepot is simple: one-time data assignment from the originator and cloud based syncing—thus making experimental and modeling data available across the collaboration with minimum overhead. Since data sharing is cloud based, our approach opens up the possibility of using new software developments and hardware scalability which are associated with elastic cloud computing. We provide an implementation that relies on existing synchronization services and is usable from all devices via a reactive web interface. We are motivating our solution by solving the practical problems of the GinJang project, a collaboration of three universities across eight time zones with a complex workflow encompassing data from electrophysiological recordings, imaging, morphological reconstructions, and simulations.

Keywords: morphology, electrophysiology, imaging, data management, neuroinformatics, cloud services, research data management

1. INTRODUCTION

Science today deals with a “data deluge” caused by the widespread use of high-throughput sensors in experiments, and the ever more complex simulations afforded by increased computational power (Moore, 1965). Both measured and simulated data need to be stored in raw form, preprocessed, contextualized with meta-data, organized to facilitate queries, and then analyzed to produce scientific statements. Ideally, peer-reviewed data should also be available for replication and re-analysis to test new hypotheses as knowledge progresses.

In addition, the need for multi-university collaboration is particularly acute in neuroscience being a multilevel discipline. It tackles questions spanning disparate levels of organization such as genes, neurons, circuits, and behavior with a variety of methods including sequencing, electrophysiology, and computer simulations (Shepherd et al., 1998). Projects with such multi-university collaborations benefit from well organized coordination of the participating specialists (Cummings and Kiesler, 2007).

One challenging aspect of project workflows might concern immediate sharing of highly structured and voluminous data across labs. Tasks of such a project workflow can interdepend: a further step of the local work depends on another operation that is remotely carried out. In this case, scientific workflows allow to optimize and then more efficiently execute scientific processes (Ludäscher et al., 2009). For example, analysis results can motivate further collection of experimental data, whereupon it is clearly of advantage that they are made available once they are produced.

Proposals to alleviate the data management overhead frequently might require scientists to change and diminish their local processing workflow in order to be able to offer distributed access for collaborators to participate in the project. We propose here a novel approach which integrates seamlessly the widespread filesystem-based acquisition, analysis, and publication workflows by leveraging proven cloud synchronization technology. Our implementation of this approach, a service called

NeuronDepot, enables researchers to continue interacting with the scientific project data through the filesystem and at the same time opens up the data for further processes in cloud-based web applications. In this way NeuronDepot exploits the existing substantial investment in development, acquisition, and training in local applications with their mature and rich interfaces and local access to data. We illustrate the approach with a deployment of NeuronDepot tailored to the specific needs of the GinJang project (<http://projects.g-node.org/ginjang/>), a complex use case that combines data from electrophysiological recordings, imaging, morphological reconstructions, and simulations.

Several initiatives have established databases to make neuromorphological or neurophysiological research data publicly available. NeuroMorpho (<http://neuromorpho.org/>) is a curated inventory of digitally reconstructed neurons. The goal of the project is to provide dense coverage of available reconstruction data for the neuroscience community (Ascoli et al., 2007). The neurodatabase.org project (<http://neurodatabase.org>) and the Collaborative Research in Computational Neuroscience (CRCNS) site (<http://crcns.org>) host electrophysiological data that have been specifically selected by contributing labs for the purpose of making the data available to the public. Typically, data in these databases are from studies that have been published and are provided for use in further investigations after they have served their primary purpose. Only a few projects have been designed to support data sharing in collaborative research.

The CARMEN portal (<https://portal.carmen.org.uk/>) allows neuroscientists to share data and programs from neurophysiological experiments. Data analysis functions are provided as services that can be applied to the data stored in the system (Austin et al., 2011). Data, metadata, and analysis workflows are accessible via a web interface.

The German Neuroinformatics Node (G-Node) provides a platform for management and sharing of neurophysiological data (<http://www.g-node.org/data>). Users can upload, organize, and annotate data, and make them accessible to other users or the public. Data annotation follows a flexible schema (Grewe et al., 2011) so that any metadata necessary can be entered. An API provides fine-grained data access through common languages like Python or Matlab, enabling data management and collaborative data sharing directly from the scientists' local data workflow environments (Sobolev et al., 2014a,b).

Recently, the International Neuroinformatics Coordinating Facility (INCF) established the INCF Dataspace (<http://incf.org/dataspace>), a cloud-based file system to share all kinds of neuroscience data.

One of the first databases integrating results from various fields like morphology, physiology, and immunohistochemistry is the Bombyx Neuron Database for assembling and sharing experimental and analytical data (Kazawa et al., 2008). Its integrative approach inspired also the development of NeuronDepot.

2. SCOPE OF THE NeuronDepot APPROACH

In contrast to some of the infrastructure solutions presented above, NeuronDepot does not focus on a particular field or type of data but leaves the specifics of each data type to the well-established working environments of the participating members.

NeuronDepot supports the scientist by providing a service that integrates data flows with the corresponding management and data analysis.

Beyond facilitating collaboration, the development of a database to properly store and backup all the data of the project makes it accessible to further projects. Putting data into structured databases facilitates its reuse and enables replication and verification of analyses.

2.1. THE GinJang PROJECT AND ITS WORKFLOW

NeuronDepot was developed around the German–Japanese collaboration GinJang (<http://projects.g-node.org/ginjang/>). This project provides a perfect opportunity for use-case-driven development and field-testing of the NeuronDepot infrastructure because (1) it involves three universities with several labs across multiple time zones, (2) it deals with different types of data from neuroanatomy and electrophysiology, and (3) it requires quick synchronization and reliable transfer of large quantities of raw data with complex associated metadata, including both recorded data and simulation results.

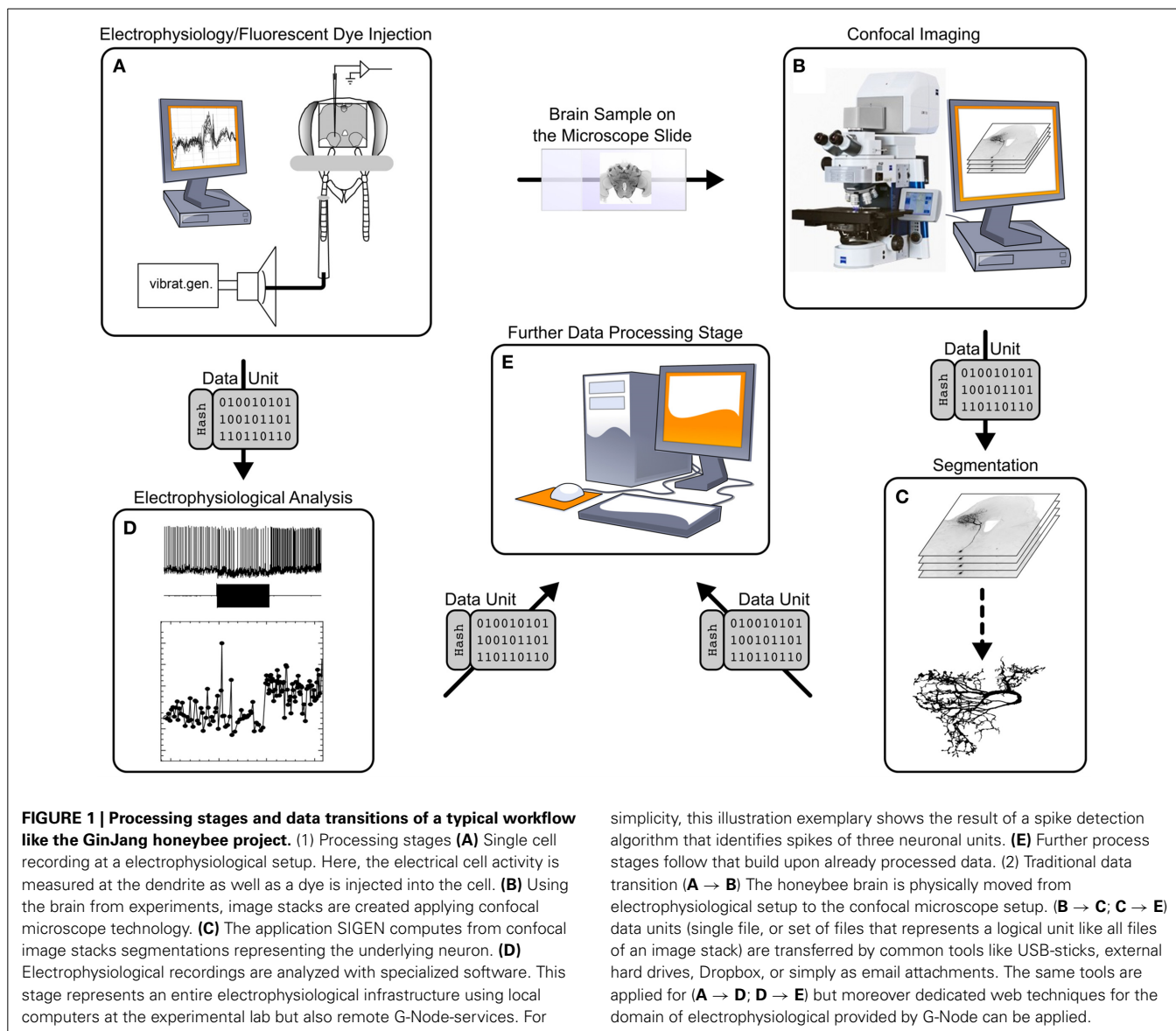
The GinJang project studies the processing of auditory signals in the honeybee. Honeybees communicate the direction and distance to food sources with hive-mates by waggle dance (Frisch, 1967). The hive-mates detect and process airborne vibration caused by the bee's wingbeat during the waggle dance, which consists of vibration pulses with a highly specific temporal pattern. Several critical interneurons for processing the airborne vibration have been identified (Ai et al., 2007, 2009; Ai, 2010; Ai and Itoh, 2012; Ai and Hagio, 2013). However, the neural processing of these vibration signals has rarely been studied: types and roles of neurons involved, their circuitry, and their development are largely unknown.

Members of the GinJang project also developed a program (SIGEN, see Minemoto et al., 2009) that is used to automatically extract and segment the morphology of interneurons that are involved in vibration processing. Goal of the GinJang project is to clarify the morphological characteristics of the vibration-processing neurons and their morphological development according to age and experience of the bees.

The workflow of the GinJang project is illustrated in **Figures 1, 2**. The experimental setup is at Fukuoka University where electrophysiological measurements (**Figure 1A**), electrophysiological analyses (**Figure 1D**), and imaging (**Figure 1B**) are performed. The image stacks are used at the University of Hyogo for neuronal segmentation (**Figure 1C**). The resulting 3D neuronal segmentations are then normalized by registering them to the Honeybee standard brain (HSB; <http://www.neurobiologie.fu-berlin.de/BeeBrain/Project.html>), which is done at Fukuoka University. Finally, morphological analyses, simulations, and further analyses are done at Ludwig-Maximilians-Universität München (LMU) (**Figure 1E**).

2.1.1. Data acquisition

The vibration-sensitive neurons in the honeybee auditory system are electrophysiologically and anatomically characterized at Fukuoka University. Using sharp electrodes, voltage traces are recorded from interneurons in response to several sensory input



protocols (Ai et al., 2009; Ai and Itoh, 2012; Kai et al., 2013). Then the neuron is filled with a dye and imaged at a different setup using confocal microscopy to generate anatomical image stacks (Ai, 2010; Ai and Hagio, 2013). Thus, every experiment generates three kinds of data:

- Electrophysiological data (e.g., voltage and current traces).
- Microscopy image stacks.
- Honeybee metadata (e.g., age or colony) and neuron metadata (e.g., phenotype).

2.1.2. Segmentation

Image stacks are transferred to the University of Hyogo, Japan. Here, using automated image analysis software SIGEN (Yamasaki et al., 2006; Minemoto et al., 2009) the 3D structure of the neuron is extracted and stored using the SWC file

format (<http://www.neuronland.org/NLMorphologyConverter/MorphologyFormats/SWC/Spec.html>). At this stage two kinds of data are generated:

- Segmented neuron (e.g., SWC file).
- Parameters used for segmentation (which constitute segmentation metadata).

2.1.3. Registration

The morphological segmentations of the neurons are transferred back to Fukuoka University for registration into the Honeybee Standard Brain using various transformations. We use the honeybee standard brain to analyze the spatial relationships among morphologically and physiologically characterized vibration-sensitive neurons. The neuronal profiles of stained interneurons, obtained from different preparations, are segmented as explained

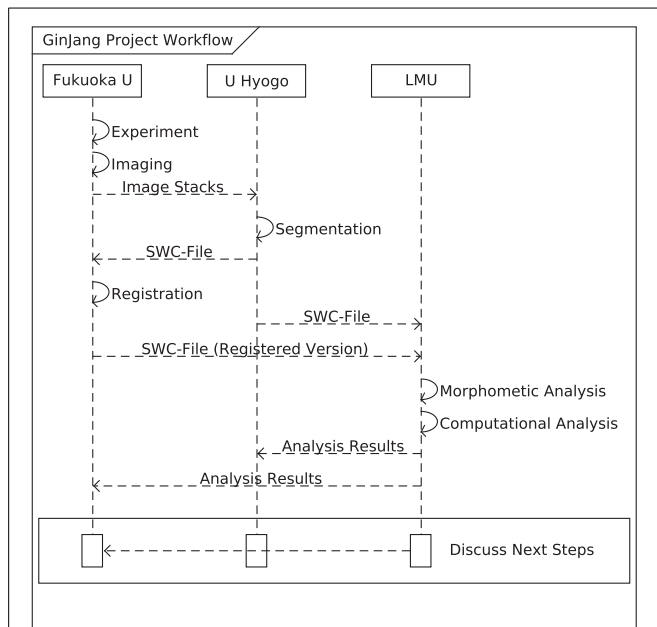


FIGURE 2 | Sequence Diagram of the GinJang workflow (morphological scope). The GinJang workflow starts at the Fukuoka University with two processing stages (indicated by solid arrows): the experimental data collection and the imaging processing stage. Anatomical image stacks are transferred (dashed arrow) to University of Hyogo where they are segmented. Segmented neurons are transferred to the LMU and also back to Fukuoka University where they are registered to the honeybee standard brain. Unregistered segmentation and registered segmentation are used for simulations and analysis at the LMU. Using analysis results, scientists in Fukuoka can tweak existing experiments or design new ones.

in the previous section. Subsequently, the neuropilar outlines are traced semi-automatically with Amira 4.1 (Evers et al., 2005) and ITK-SNAP (<http://www.itksnap.org/>). These neuropilar label fields are used to register the segmented neuron of each preparation into the honeybee standard brain following the method described by Brandt et al. (2005). Data generated at this stage are:

- Registered neuron morphology.
- Parameters used for registration.

2.1.4. Analysis

These segmentations (both registered and unregistered) are transferred to the LMU, Germany, where 3D segmentations are used for morphometric analysis and simulation studies. Multiple kinds of data are generated at this stage:

- Model files for simulations.
- Simulation metadata, e.g., parameters of simulation, location of stimulation (input) and measurements (output).
- Simulation results: visualizations and summary data.
- Morphometric analysis metadata, e.g., subregion of analysis, metrics used.
- Results of morphometric analysis: volume, surface area, number of branch points.

2.1.5. Traditional data transfer methods

The workflow of the GinJang project requires multiple data transfers between diverse processing stages. These transfers were previously done via e-mail, usb-sticks, external hard drives, ftp-servers, or cloud storage services (like Dropbox, <http://www.dropbox.com/>). NeuronDepot replaces these traditional data transfer methods.

3. REQUIREMENTS ANALYSIS

We asked the members of the GinJang Project to specify the features that they expect to have in NeuronDepot. Based on those we came up with the following set of requirements:

1. Data Management

- Replacement of “manual” data transitions that are using memory devices like e-mail, USB-sticks, external hard drives, FTP-servers, or cloud storage services.
- Ease of metadata assignment for various kinds of data like image stacks, voltage trace, and neuronal reconstructions.
- Interrelate various kinds of data.
- Visibility of the current state of the project through a web browser.
- Automatic update and synchronization across project workstations.

2. Integration

- Maintenance of the well-established work environments of the participating scientists.
- Minimization of the integration effort.

3. Data Security

- Reliability of upload and download of large data.
- Access control.

4. Automated Backup

5. Additional Requirements

- Easy adaptation to new data-specific requirements that emerge during the project.
- Support for automated data processing like metadata extraction, analysis, and simulation.
- Quick overview of contents and metadata.
- Flexible search of data.

4. CONCEPT

4.1. NeuronDepot AS A SERVICE

NeuronDepot is designed as a service. As opposed to a product, functionalities of a service are set up to meet a specific set of requirements at a point in time (Truex et al., 1999; Bennett et al., 2000; Bullinger et al., 2003). Therefore, the specific form of NeuronDepot changes as the project progresses and its requirements continually evolve. Moreover, NeuronDepot brings together other already existing service-modules, which are reassembled and configured to meet the current requirements. While building up NeuronDepot from its sub-services, we make sure that NeuronDepot stays functional even when its sub-services develop with time. Also, these developments can be

utilized in evolving NeuronDepot. By offering new functionality in a way that is compatible with existing services, tools, training, and working environments, the costs of data sharing in a collaboration are brought down to a minimum while the accessibility of research assets is future-proofed.

4.2. CORE IDEA

When handling a large amount of data, it is common for scientists to arrange the corresponding files in a directory tree. By doing this, they often encode metadata in the name of directories, for example, the date of recordings or experimental parameters. NeuronDepot also uses this well established principle. The difference is that NeuronDepot automatizes this. It employs a set of rules to automatically create such a directory structure and arrange the data. It uses the associated metadata (available in the database) for naming the directories. The rules for forming this directory structure can be changed. Thus, the same data can be organized in different structures as required by the scientist.

4.3. DEFINITIONS

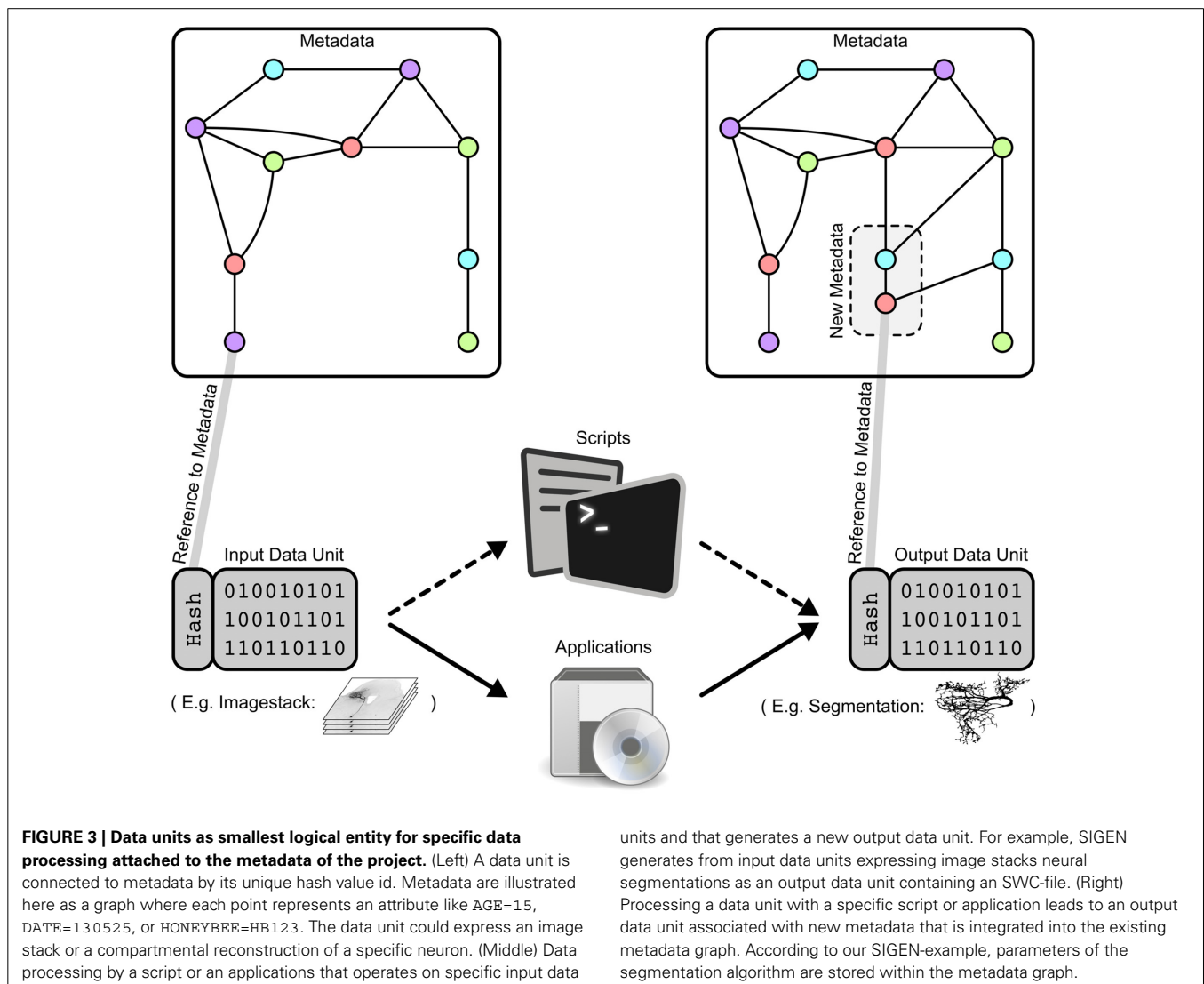
In this section we define terminologies which are used in explaining NeuronDepot.

4.3.1. Data unit

A data unit (Figure 3, bottom left and bottom right) is a logical grouping of one file (trivial case) or multiple files which are generated by a single process. Examples of individual data units in the GinJang context include: an image stack consisting of several image files, the morphology of a neuron represented in a single SWC-file, or several plots and tables resulting from simulations of a neuron's electrophysiological characteristics.

4.3.2. Context path and context trees

Any data unit can be uniquely identified by a subset of the metadata attributes associated with it (Figure 3). We define the context path of a data unit as an ordered list of the specific attributes that uniquely identify it (Figure 4). This context path can be used to construct a path in the file system where the order of metadata



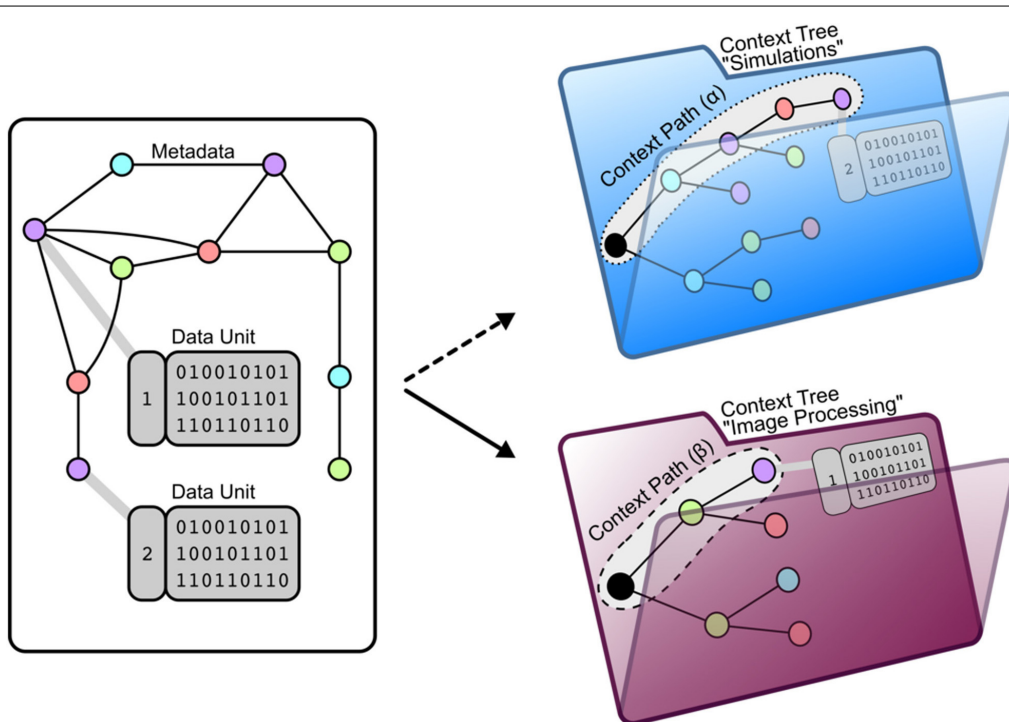


FIGURE 4 | Mapping a data unit to context trees using attached metadata. (Left) Each data unit is connected to metadata. Using this metadata allows the organization of data units by two aspects: The entire data of a project can be (1) sub-divided into divers subsets of data where (2) data units are arranged within a tree structure where the nodes represent metadata and the leaves represent the data unit. As a meaningful specification of such a arrangement depends on the context of data usage, we call this arrangement *context tree*. (Right) One context tree of NeuronDepot arranges project data for the morphometric processing stage. The file format SWC serves as a filter

argument as just SWC-files are needed for simulation. Metadata attributes LABOR_STATE, REGION, HONEYBEE_ID, and SIGEN_PARAMETERS serve for grouping. *Example of context path (α) pointing to a data unit containing a segmentation: /forager/left_DL/HB130427/D20V05C01S01/morphology.swc* Another context tree of NeuronDepot arranges project data for our imaging processing stage. Here, the project data are reduced to image stacks. Metadata attributes HONEYBEE_ID and REGION serve for grouping. *Example of context path (β) pointing to a data unit containing an image stack: /HB130427/left-DL/*.tiff*

attributes corresponds to the hierarchical levels in the file system.

Example: If a member of the project wants to analyze one particular segmentation of neuron NRN-1 of honeybee HB123, the following two paths leading to the corresponding data unit would represent these attributes:

- (1) HB123/NRN-1/segmentation/
- (2) segmentation/HB123/NRN-1/

The desired order of the attributes depends on how the data units are to be queried for specific analyses: the path order is projected into a hierarchy and therefore defines different grouping levels specific analyses.

4.3.3. Projection

A projection is the representation of a context tree within the file system. It is comparable to materialized views of relational database management systems.

4.4. DESIGN CONSIDERATIONS

The architecture of NeuronDepot follows these principles.

4.4.1. Incorporation of existing open source components

The open source ecosystem holds multiple solutions solving very specific tasks. Some examples are SQLAlchemy¹ (for controlling the persistence of objects by mapping them to database structures), numpy² (solving highly optimized numerical tasks), or matplotlib³ (illustrate data by drawing graphs and figures). Moreover, the community of neuroinformatics has added several domain-specific tools for simulations, analysis, and processing of data from the field. We have incorporated some of these solution while developing NeuronDepot (see section 6). NeuronDepot is also structured so that other such solutions can be integrated to it.

4.4.2. Utilization of established cloud services

Cloud services have rapidly emerged as a widely accepted paradigm built around core concepts such as on-demand computing resources, elastic scaling, elimination of up-front investment, reduction of operational expenses, and establishing a

¹<http://www.sqlalchemy.org/>

²<http://www.numpy.org/>

³<http://matplotlib.org/>

pay-per-use business model for information technology and computing services. The use of cloud services helps to reduce development time and effort.

5. HOW NeuronDepot WORKS

5.1. DATA ARRANGEMENT: FLAT ON THE SERVER AND HIERARCHICAL ON USER WORK STATIONS

NeuronDepot applies systematically the principle of using folder names and file names as carrier for metadata describing the data contained in the filesystem. For flexibility, the collection of data units is stored in a central server in a flat structure where each data unit has a unique identifier, and the metadata are kept separate and referenced to those identifiers. When users define a subset of data they are interested in, along with a hierarchical arrangement that suits their needs, NeuronDepot creates a user- and task-specific context tree as a hierarchy of symbolic links with the data units at the leaves. By exposing these hierarchies to a synchronization daemon, the projection is made available to every workstation that subscribes to it.

5.2. DATA ASSIGNMENT

NeuronDepot also leverages advances in synchronization technology for the data upload process: the user simply places new data units in a designated floating folder (comparable to the Camera Upload folder of Dropbox, see section 5.3). This folder is synchronized to the server. Then, the data units appear as available for metadata assignment via a graphical user interface of NeuronDepot. Once metadata assignment is complete, data units can be projected, as described above, to hierarchies that are adapted to the local users' workflows. Data units are now also accessible to cloud analytic services that directly query the metadata database without demanding a specific projection, as these clients are not constrained by the hierarchical data model of filesystems. NeuronDepot thus maintains consistency all the way from the scientists' local copy of acquired data to the cloud-based analysis platforms.

5.3. CLOUD-BASED DATA FLOW

NeuronDepot's mechanism for data transmission is based on synchronization by GWDG Cloud Share (Figure 5). This cloud storage service is used to keep all local computers that are involved in the project updated by the server and, therefore, updated among each other. This core update process is based on synchronization on the file system level. In order to integrate data units into workflows, the system provides two types of base folders: floating folders (Figure 5-1) and context tree folders (Figure 5-3). Floating folders are provided with read/write permissions for project members. Data within floating folders are not assigned to the metadata structure and, therefore, are in a floating state. Floating folders are part of the data-assignment process (Figure 5-2). The second type of folder is the context tree folder with read-only permissions for project members that synchronize projected context trees to the local work environment.

The underlying data transfer workflow replaces traditional transfer methods as described above and consists of three steps: (1) new data units are stored within the floating folder and synchronized to the server. (2) Synchronized data units within

the floating folder are assigned to the existing project data via a web application. As NeuronDepot's web GUI uses responsive web design it provides optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices from mobile phones to desktop computer monitors (Marcotte, 2010). The system ensures that all data are correctly related to each other and that all data stay consistent. Project members can plug scripts into this assignment process to automate and facilitate data processing. Moreover, the system provides diverse reports to brief the scientists about the current state, or about recent changes. (3) NeuronDepot distributes data units back to project members. According to the underlying context tree, NeuronDepot synchronizes projected context tree folders by cloud storage services to the workstations of the scientists.

6. SYSTEM ARCHITECTURE

6.1. GRAPHICAL USER INTERFACE

The architecture underlying NeuronDepot consists of individual layers and components (Figure 6). Users can access NeuronDepot via a web application or through cloud storage synchronization clients. NeuronDepot distinguishes two kinds of users: registered project members which can manage the entire project data and administrators with global permissions including user management. NeuronDepot uses OpenIDs (<http://openid.net/>) for authentication.

The graphical user interface (GUI) consists of two parts: a web application (Figure 6A, left) and local applications (Figure 6, right). The web application provides forms for assigning data, entering metadata, annotating data with metadata, and deleting data units and metadata. We used the micro web framework Python-Flask (<http://flask.pocoo.org/>) for rapid development.

Upload and download processes are handled by GWDG Cloud Share (<https://powerfolder.gwdg.de/>) incorporated by the virtual filesystem projection layer (see below). Thus, scientists can use established tools like Windows Explorer, Mac Finder, Linux Nautilus, or other file managers to copy files for upload in dedicated folders (Figure 6A, right).

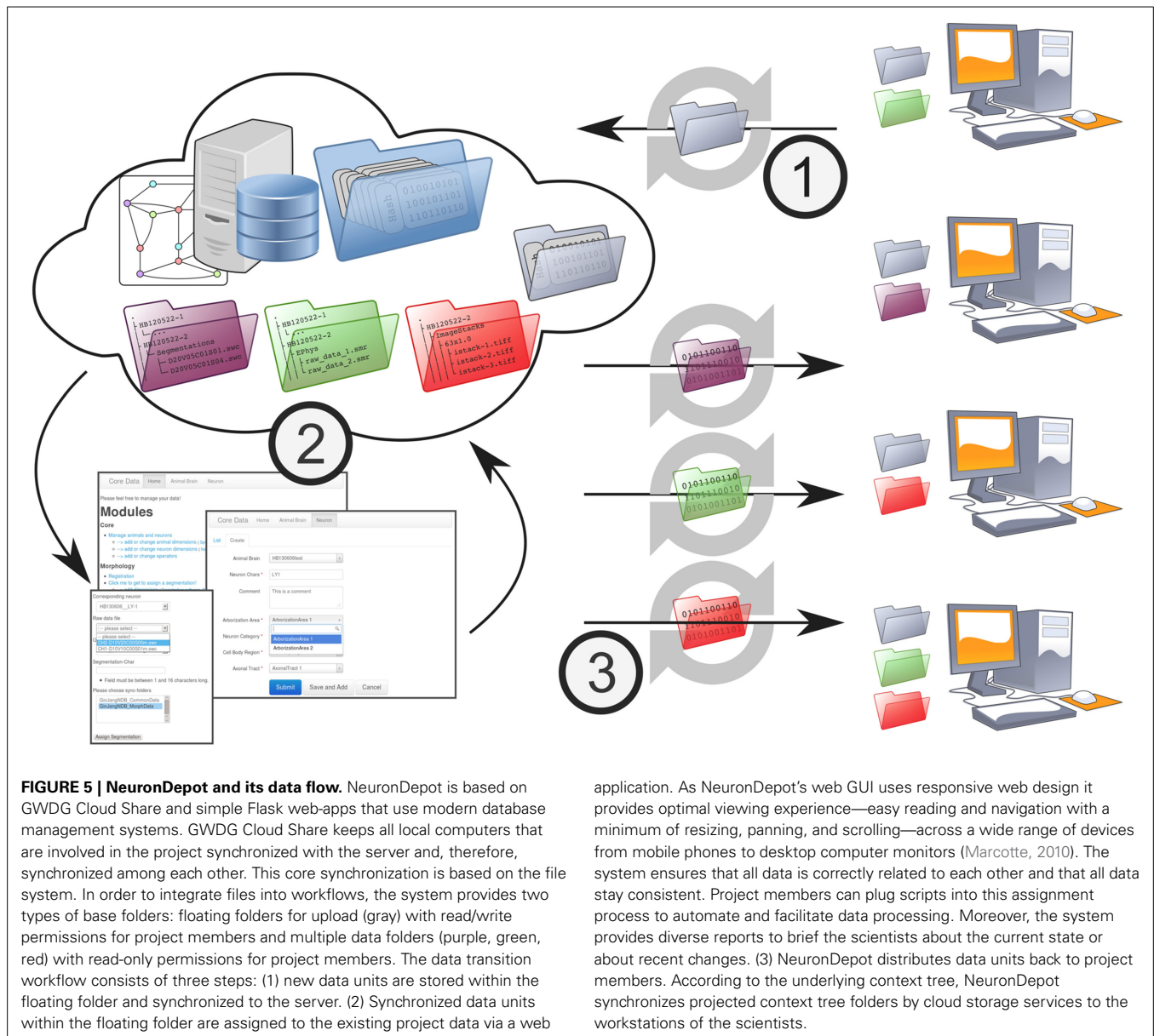
6.2. BUSINESS LOGIC AND VIRTUAL FILESYSTEM PROJECTION

The business logic (Figure 6B, left) encodes the NeuronDepot logic rules that determine how data can be created, read, updated, and deleted. We therefore propose a Virtual Filesystem Projection layer (Figure 6B, right) which can map data items to cloud storage synchronization clients based on project-specific metadata and provides a consistent view of the file system structure for computational workflows.

A workflow is composed out of multiple tasks. Typically tasks extract metadata, index data items, manipulate images or calculate statistics. Workflows are created in Python with the help of libraries like Snakemake (Köster and Rahmann, 2012).

Workflows are triggered explicitly by user interactions over the web frontend or implicitly by the Virtual Filesystem Projection layer when new files are added. The execution state of the workflow is displayed in the web application.

The filesystem projection layer projects the data items based on metadata to directories and files. The hierarchy of directory tree



is generated by rules using project metadata and the data items which are controlled by the persistence layer.

6.3. PERSISTENCE LAYER

The persistence layer (Figure 6C) consists of the following two components:

Project Metadata Within a project database additional metadata are stored. This can be metadata which were extracted by a computational workflow or manually-entered data. For mapping Python objects to database objects we use SQLAlchemy (<http://www.sqlalchemy.org/>) storing metadata in an PostgreSQL (<http://www.postgresql.org/>).

Storage Backend The responsibility of the storage backend is to consistently store data items and provide abstractions for the file system projection layer. NeuronDepot uses Camlistore (<https://camlistore.org/>), which stores files like a

traditional filesystem. Moreover, it's specialized in storing higher-level objects.

7. GinJang USING NeuronDepot

In the context of the GinJang project, NeuronDepot manages image stacks and morphological reconstructions of neurons (see section 2.1). The database contains all the image stacks and neuronal reconstructions currently being analyzed as part of the project. It also contains the associated metadata (see section 2.1). The web application presents all the data annotated with metadata in easily readable tables so that the scientists can keep track of it. Such a central presentation of all the data and metadata of the project is useful during the web-based discussions of collaborators in tracking the progress of the project.

By using NeuronDepot, the process of sharing data between the collaborators has been made simple. The data are uploaded at

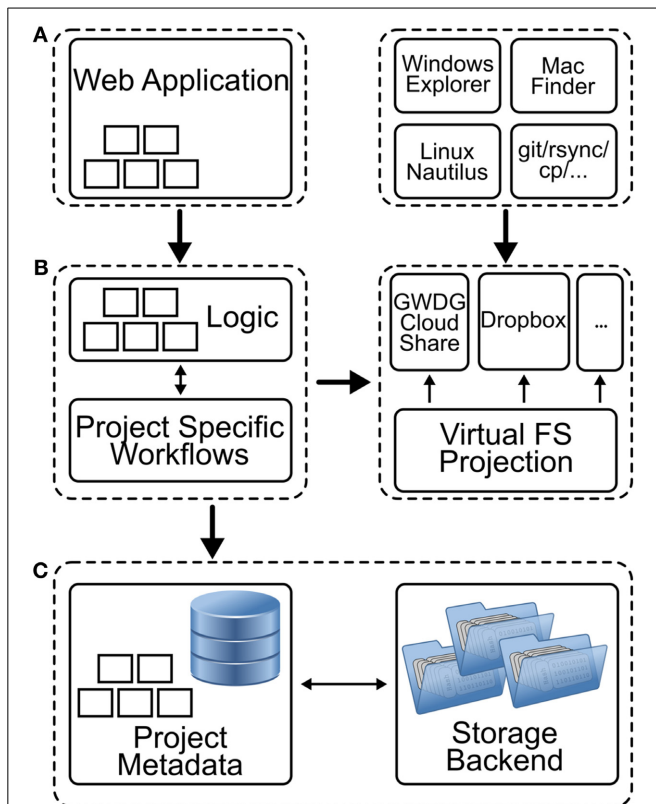


FIGURE 6 | Schematic representation of the different components of NeuronDepot. (A) The graphical user interface consists of two parts: a web application and local applications. The web application (*left*) provides forms for assigning data, enter metadata, and annotating data with metadata. The upload and download processes are handled by GWGD Cloud Share incorporated by the filesystem projection layer (see below). Here, scientists can use established tools like Windows Explorer, Mac Finder, Linux Nautilus, or other file managers to copy files for upload in dedicated folders (*right*) which are connected to the cloud services. **(B)** The business logic (*left*) encodes the NeuronDepot logic rules that determine how data can be created, read, updated, and deleted. Moreover, when new data items are added, deleted, or modified, project-specific workflows can be triggered for each processing stage (*illustrated by five small rectangles*). A Virtual Filesystem Projection layer (*right*) maps data items and directories to GWGD Cloud Share synchronization clients based on project-specific metadata and provides a consistent view of the file system structure. **(C)** Within a project database additional metadata are stored. This can be metadata which was extracted by a computational workflow or manually entered data. The responsibility of the storage backend is to consistently store data items and provide abstractions for the file system projection layer.

the source of generation once and is automatically made available to the workstations where it is analyzed. Any further changes to this data, for example, if an improved neuronal reconstruction is generated, is automatically made available to the collaborator who is analyzing reconstructions. Thus, data sharing is achieved with minimum manual intervention.

The data assigned to NeuronDepot are analyzed by two collaborators (at University of Hyogo, Japan, and at LMU, Germany), each requiring them in a different hierarchical structure for their analyses. NeuronDepot automatically provides the data in the

structure the collaborators specify and thus alleviates the need for manual organization.

8. DISCUSSION

8.1. ADAPTABILITY

The system architecture of NeuronDepot can be conceptually divided into two parts: the core engine, which is not specific to any processing stage, and plain and focused modules, which are project-specific. In the GinJang project, segmentation is a processing stage that is implemented as a dedicated plain module storing, analyzing, and reporting segmentation data. Moreover, such a module provides all the required features for the data and metadata produced by this processing stage like connecting it to other existing data in NeuronDepot, handling upload of this data and specifying the information necessary while presenting it to the user.

NeuronDepot can be adapted to other projects by incorporating project-specific plain modules upon its core engine. These plain modules correspond to the different processing stages of a project, while the core engine remains the same.

8.2. DISTINGUISHING FEATURES OF NeuronDepot

NeuronDepot provides data via file system. This opens up a plethora of tools that are available at the local work bench like (1) desktop search using diverse indexing methods (spotlight, locate, Copernic, Google Desktop), (2) file system explorers (for searching and sorting), (3) Backup, (4) Version-Control, (5) Unix-world applications like `grep`, `find`, and `tree` (since “everything is a file”), (6) transmission protocols like `ftp`, `ssh`, and `http`, and (7) file synchronization services.

An important feature of NeuronDepot is the isolation of the upload process from the GUI. In the conventional upload process the user indicates the file to be uploaded and waits until the upload process is finished. This way of uploading can be very inconvenient when uploading large files (several hundreds of MBs). This problem is further compounded when the network connection is not stable. Our approach solves this problem by isolating the upload process from the data assignment process. The upload process of NeuronDepot consists of two steps. Data is copied into the GWGD Cloud Share and then assigned from there to the database using the GUI. This upload procedure facilitates assisted assignment of data since the data are available beforehand. Certain analysis scripts can be started on the data in the virtual file system and its results can be later used during the assignment of the data via the web-GUI.

At the end user, a subset of the data in the database is presented in a tree structure. Such a representation of a desired subset of the data in a hierarchical structure provides a partitioning/grouping of the data which becomes very handy if the user intends to perform analysis or comparison on a specific subset of the data.

In NeuronDepot, a specific subset of data is encapsulated into an entity via the concept of context trees. Such an encapsulation facilitates management operations in which treatment of the subset of the data as an entity is essential such as referencing, tagging, and sharing. This is very much like a book encapsulating a set of concepts/facts and making them a single entity.

8.3. COMPARISON WITH OTHER SYSTEMS

Other file-based solutions for collaborative data sharing provide access through a web browser like the web platforms of CARMEN (<https://portal.carmen.org.uk/>) and G-Node (<http://www.g-node.org/data>). There, manual download is required to access new data when a dataset has been updated, whereas in NeuronDepot the new data are automatically provided locally.

CARMEN enables access to analysis services on its platform (Austin et al., 2011). G-Node provides access to data in a common representation through an API (Sobolev et al., 2014a) with client tools for integration with the scientist's analysis scripts (Sobolev et al., 2014b). NeuronDepot complements these approaches by presenting the data in the usual file system way. This is particularly useful for collaborations between specific labs where all partners know how to access the data.

Unlike with other existing solutions, using NeuronDepot does not require learning a new GUI or any other infrastructure specific usage features since NeuronDepot provides the data as directory trees to the user. Having this feature, project members could keep their established working environments. In other words: NeuronDepot adapts for existing workflows whereas other systems require the scientist to adapt its workflow to the new system.

8.4. FURTHER DIRECTIONS, LIMITATIONS, AND OPEN QUESTIONS

A package/extension for an existing web-framework like Flask or Django can be developed by reorganizing the system components of NeuronDepot. Several existing solutions of the Open Source Ecosystem were used in the development of NeuronDepot and this is a way of contributing back to it. Moreover, it serves as a good building block for the development of new data software.

As explained in section 4.1, NeuronDepot is a service which develops as the associated project progresses. In the context of the GinJang project, extensions to NeuronDepot are being developed which automate morphological analysis and simulations using the neuronal reconstructions.

At the moment, the context trees used to provide the user with data are hard-coded. The user has to communicate with the developers to have different data structures provided. This process can be slow and can prove to be a hindrance to the scientist's work. A service can be incorporated which enables the user to specify what data structure is needed. This would reduce the user's dependence on the developers and also allow the user to quickly adjust the data that are required from NeuronDepot.

9. CONCLUSION

With this software architecture, we contribute an approach to scientific data workflow and specifically a tool to the neuroscientific infrastructure. NeuronDepot's principal merit is that it integrates smoothly with established tools and resolves the transition from local to cloud-based processing. In doing so, it enables researchers to leverage the advantages of cloud services while not requiring them to relinquish control of their data or analysis.

ACKNOWLEDGMENTS

We thank Andrey Sobolev, Adrian Stoewer, and Christian Kellner for discussions and contributions to a previous version.

Supported by the Japan Science and Technology Agency (JST) and the Federal Ministry of Education and Research (BMBF, grants 01GQ1116 and 01GQ1302) and a Hyogo Overseas Research Network (HORN) Scholarship.

REFERENCES

- Ai, H. (2010). Vibration-processing interneurons in the honeybee brain. *Front. Syst. Neurosci.* 3:19. doi: 10.3389/neuro.06.019.2009
- Ai, H., and Hagio, H. (2013). Morphological analysis of the primary center receiving spatial information transferred by the waggle dance of honeybees. *J. Comp. Neurol.* 521, 2570–2584. doi: 10.1002/cne.23299
- Ai, H., and Itoh, T. (2012). "The auditory system of the honeybee," in *Honeybee Neurobiology and Behaviors, 2nd Edn.*, eds D. Eisenhardt, C. G. Galizia, and M. Giurfa (Berlin, Heidelberg: Springer Verlag), 269–284.
- Ai, H., Nishino, H., and Itoh, T. (2007). Topographic organization of sensory afferents of Johnston's organ in the honeybee brain. *J. Comp. Neurol.* 502, 1030–1046. doi: 10.1002/cne.21341
- Ai, H., Rybak, J., Menzel, R., and Itoh, T. (2009). Response characteristics of vibration-sensitive interneurons related to Johnston's organ in the honeybee, *Apis mellifera*. *J. Comp. Neurol.* 515, 145–160. doi: 10.1002/cne.22042
- Ascoli, G. A., Donohue, D. E., and Halavi, M. (2007). NeuroMorpho.Org: a central resource for neuronal morphologies. *J. Neurosci.* 27, 9247–9251. doi: 10.1523/JNEUROSCI.2055-07.2007
- Austin, J., Jackson, T., Fletcher, M., Jessop, M., Liang, B., Weeks, M., et al. (2011). CARMEN: code analysis, repository and modeling for e-neuroscience. *Proc. Comput. Sci.* 4, 768–777. doi: 10.1016/j.procs.2011.04.081
- Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., and Munro, M. (2000). "Service-based software: the future for flexible software," in *Seventh Asia-Pacific Software Engineering Conference (APSEC 2000)*, (Singapore: IEEE Computer Society Press), 214–221. doi: 10.1109/APSEC.2000.896702
- Brandt, R., Rohlfing, T., Rybak, J., Kroczyk, S., Maye, A., Westerhoff, M., et al. (2005). Three-dimensional average-shape atlas of the honeybee brain and its applications. *J. Comp. Neurol.* 492, 1–19. doi: 10.1002/cne.20644
- Bullinger, H.-J., Fähnrich, K.-P., and Meiren, T. (2003). Service engineering-methodical development of new service products. *Int. J. Prod. Econ.* 85, 275–287. doi: 10.1016/S0925-5273(03)00116-6
- Cummings, J. N., and Kiesler, S. (2007). Coordination costs and project outcomes in multi-university collaborations. *Res. Pol.* 36, 1620–1634. doi: 10.1016/j.respol.2007.09.001
- Evers, J., Schmitt, S., Sibila, M., and Duch, C. (2005). Progress in functional neuroanatomy: precise automatic geometric reconstruction of neuronal morphology from confocal image stacks. *J. Neurophysiol.* 93, 2331. doi: 10.1152/jn.00761.2004
- Frisch, K. (1967). "The tail-wagging dance as a means of communication when food sources are distant," in *The Dance Language and Orientation of Bees*, (Cambridge, MA: Belknap Press of Harvard University Press), 57–235.
- Grewe, J., Wachtler, T., and Benda, J. (2011). A bottom-up approach to data annotation in neurophysiology. *Front. Neuroinform.* 5:16. doi: 10.3389/fninf.2011.00016
- Kai, K., Ikeno, H., Haupt, S. S., Rautenberg, P. L., Wachtler, T., and Ai, H. (2013). "Response properties of auditory interneurons in the honeybee brain," in *Bernstein Conference 2013*, Sept. 24–27 (Tuebingen, Germany).
- Kazawa, T., Ikeno, H., and Kanzaki, R. (2008). Development and application of a neuroinformatics environment for neuroscience and neuroethology. *Neural Netw.* 21, 1047–1055. doi: 10.1016/j.neunet.2008.05.005
- Köster, J., and Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics* 28, 2520–2522. doi: 10.1093/bioinformatics/bts480
- Ludäscher, B., Altintas, I., Bowers, S., Cummings, J., Critchlow, T., Deelman, E., et al. (2009). "Scientific process automation and workflow management," in *Scientific Data Management*, eds A. Shoshani, and D. Rotem (London: Chapman & Hall).
- Marcotte, E. (2010). Responsive web design. *A List Apart*, 306.
- Minemoto, T., Saitoh, A., Ikeno, H., Isokawa, T., Kamiura, N., Matsui, N., et al. (2009). "SIGEN: system for reconstructing three-dimensional structure of

- insect neurons," in *Proceedings of the Asia Simulation Conference, JSST2009, CDROM* (Shiga), 1–6.
- Moore, G. E. (1965). *Cramming More Components Onto Integrated Circuits*. New York, NY: McGraw-Hill.
- Shepherd, G. M., Mirsky, J. S., Healy, M. D., Singer, M. S., Skoufos, E., Hines, M. S., et al. (1998). The human brain project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data. *Trends Neurosci.* 21, 460–468. doi: 10.1016/S0166-2236(98)01300-9
- Sobolev, A., Stoewer, A., Leonhardt, A. P., Rautenberg, P. L., Kellner, C. J., Garbers, C., et al. (2014a). Integrated platform and API for electrophysiological data. *Front. Neuroinform.* 8:32. doi: 10.3389/fninf.2014.00032
- Sobolev, A., Stoewer, A., Pereira, M., Kellner, C. J., Garbers, C., Rautenberg, P. L., et al. (2014b). Data management routines for reproducible research using the G-Node Python Client library. *Front. Neuroinform.* 8:15. doi: 10.3389/fninf.2014.00015
- Truex, D. P., Baskerville, R., and Klein, H. (1999). Growing systems in emergent organizations. *Commun. ACM* 42, 117–123. doi: 10.1145/310930.310984
- Yamasaki, T., Isokawa, T., Matsui, N., Ikeno, H., and Kanzaki, R. (2006). Reconstruction and simulation for three-dimensional morphological structure of insect neurons. *Neurocomputing* 69, 1043–1047. doi: 10.1016/j.neucom.2005.12.042
- Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Received: 31 December 2013; accepted: 30 April 2014; published online: 12 June 2014.*
Citation: Rautenberg PL, Kumaraswamy A, Tejero-Cantero A, Doblender C, Norouzian MR, Kai K, Jacobsen H-A, Ai H, Wachtler T and Ikeno H (2014) NeuronDepot: keeping your colleagues in sync by combining modern cloud storage services, the local file system, and simple web applications. Front. Neuroinform. 8:55. doi: 10.3389/fninf.2014.00055
- This article was submitted to the journal Frontiers in Neuroinformatics. Copyright © 2014 Rautenberg, Kumaraswamy, Tejero-Cantero, Doblender, Norouzian, Kai, Jacobsen, Ai, Wachtler and Ikeno. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.*