NWB-N: Overview

Release v1.0.0

Aug 15, 2017

Table of Contents

CHAPTER 1

Neurodata Without Borders: Neurophysiolgy (NWB-N)

Mission

Neurodata Without Borders: Neurophysiology (NWB-N) is a project to develop a unified data format for cellular-based neurophysiology data, focused on the dynamics of groups of neurons measured under a large range of experimental conditions. Participating labs provided use cases and critical feedback to the effort. The design goals for the NWB format included:

- **Compatibility**
  - Cross-platform
  - Support for tool makers
- **Usability**
  - Quickly develop a basic understanding of an experiment and its data
  - Review an experiment's details without programming knowledge
- **Flexibility**
  - Accommodate an experiment's raw and processed data
  - Encapsulate all of an experiment's data, or link to external data source when necessary
- **Extensibility**
  - Accommodate future experimental paradigms without sacrificing backwards compatibility.
  - Support custom extensions when the standard is lacking
- **Longevity**
  - Data published in the format should be accessible for decades

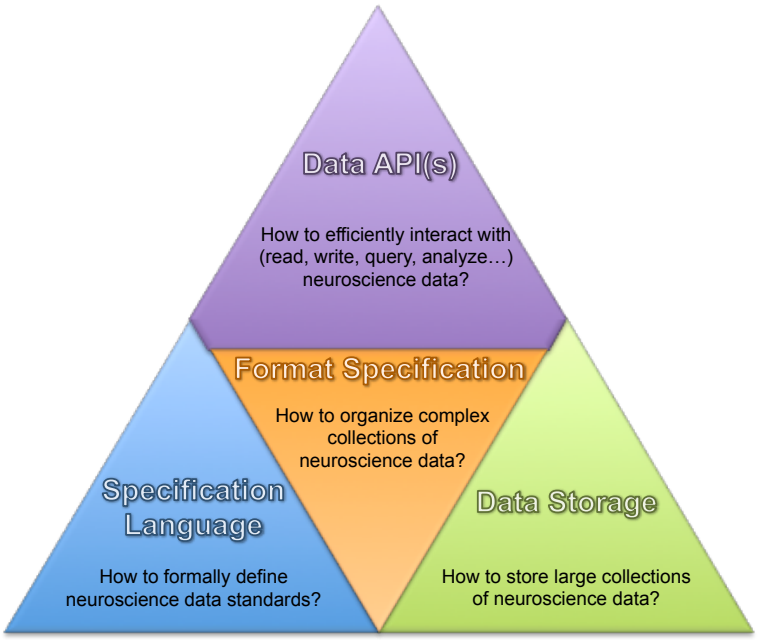

Fig. 1.1: Main components of NWB-N

Project Components

Fig. 1.1 provides a high-level overview of the main components of NWB-N. The following subsection provide a high-level overview of the problem, approach, description, and function of these components. Further details about the specific components are provided in the corresponding documentations.

Specification Language

Problem: How to formally define neuroscience data standards?

Approach: In order to support the formal and verifiable specification of neurodata file formats, NWB-N defines and uses the NWB specification language.

Description: The specification language is defined in YAML (or JSON). The specification language defines formal structures for describing the organization of complex data using basic concepts, e.g., Groups, Datasets, Attributes, and Links. Data publishers can use the specification language to extend the format in order to store types of data not managed by the base format.

Function: The primary function of the specification language is to enable the formal specification of data organizations.

Documentation: http://schema-language.readthedocs.io

Format Specification:

Problem: How to organize complex collections of neuroscience data?

Approach: Organize data hierarchically using easy-to-use primitives, e.g., Groups (similar to Folders), Datasets (n-D Arrays), Attributes (Metadata objects on Groups and Datasets), and Links (links to Groups and Datasets).

Description: The NWB format standard is governed by a formal format specification, the NWB-N schema that is formally specified using the NWB specification language. A new schema file will be published for each revision of the NWB format standard. Developers can use the schema to validate NWB files or create advanced APIs for NWB data.

Function: The primary function of the format specification is to formally specify the NWB format describing the organization of neuroscience data. The format specification provides a verifiable, computer and human readable document that governs the NWB format. The format specification is, hence, central to support development of API's and codes compliant with the NWB format and extension of the NWB format.

Documentation: http://nwb-schema.readthedocs.io

Sources: https://github.com/NeurodataWithoutBorders/nwb-schema

Data Storage

Problem: How to store large collections of neuroscience data?

Approach: NWB-N format currently uses the Hierarchical Data Format (HDF5) as primary storage mechanism.

Description: HDF5 was selected for the NWB format because it met several of the project's requirements. First, it is a mature data format standard with libraries available in multiple programming languages. Second, the format's hierarchical structure allows data to be grouped into logical self-documenting sections. Its structure is analogous to a file system in which its "groups" and "datasets" correspond to directories and files. Groups and datasets can have attributes that provide additional details, such as authorities' identifiers. Third, its linking feature enables data stored in one location to be transparently accessed from multiple locations in the hierarchy. The linked data can be external to the file. Fourth, HDF5 is widley supported across programming languages (e.g., C, C++, Python, MATLAB, R among others) and tools, such as, HDFView, a free, cross-platform application, can be used to open a file and browse data. Finally, ensuring the ongoing accessibility of HDF-stored data is the mission of The HDF Group, the nonprofit that is the steward of the technology.

Function: A primary function of the data storage is to map NWB-N primitives (Groups, Datasets, Attributes, Links etc.) to storage. In the case of HDF5 this is currently a 1-to-1 mapping as the NWB primitives match HDF5 primitives.

Documentation: http://nwb-storage.readthedocs.io

Data API(s)

Problem: How to efficiently interact with neuroscience data?

Approach: The PyNWB API provides users easy-to-use representations of NWB-N types for programmatic use and enables the mapping of these representations to/from data storage based using the NWB-N format specification.

Description: PyNWB provides critical functionality neede to read, write, use, and analyse data stored in NWB-N. PyNWB provides users an easy-to-use interface and abstractions for integrating NWB types with their codes while insulating them from implementation details with respect to specification language, format, and storage.

Function: The role of data API(s) is to facilitate efficient interaction with neuroscience data stored in the NWB-N data format (e.g,. for reading, writing, querying, and analyzing neuroscience data). A main funcion of an API is provide users a stable and usable interface for programmatic use and development of new applications. As such, a central function of the API is also to insulate developers and users from implementation details regarding the specifciation language, format specification, and data storage.

Documentation: http://pynwb.readthedocs.io

Sources: https://github.com/NeurodataWithoutBorders/pynwb

CHAPTER 2

Software Architecture

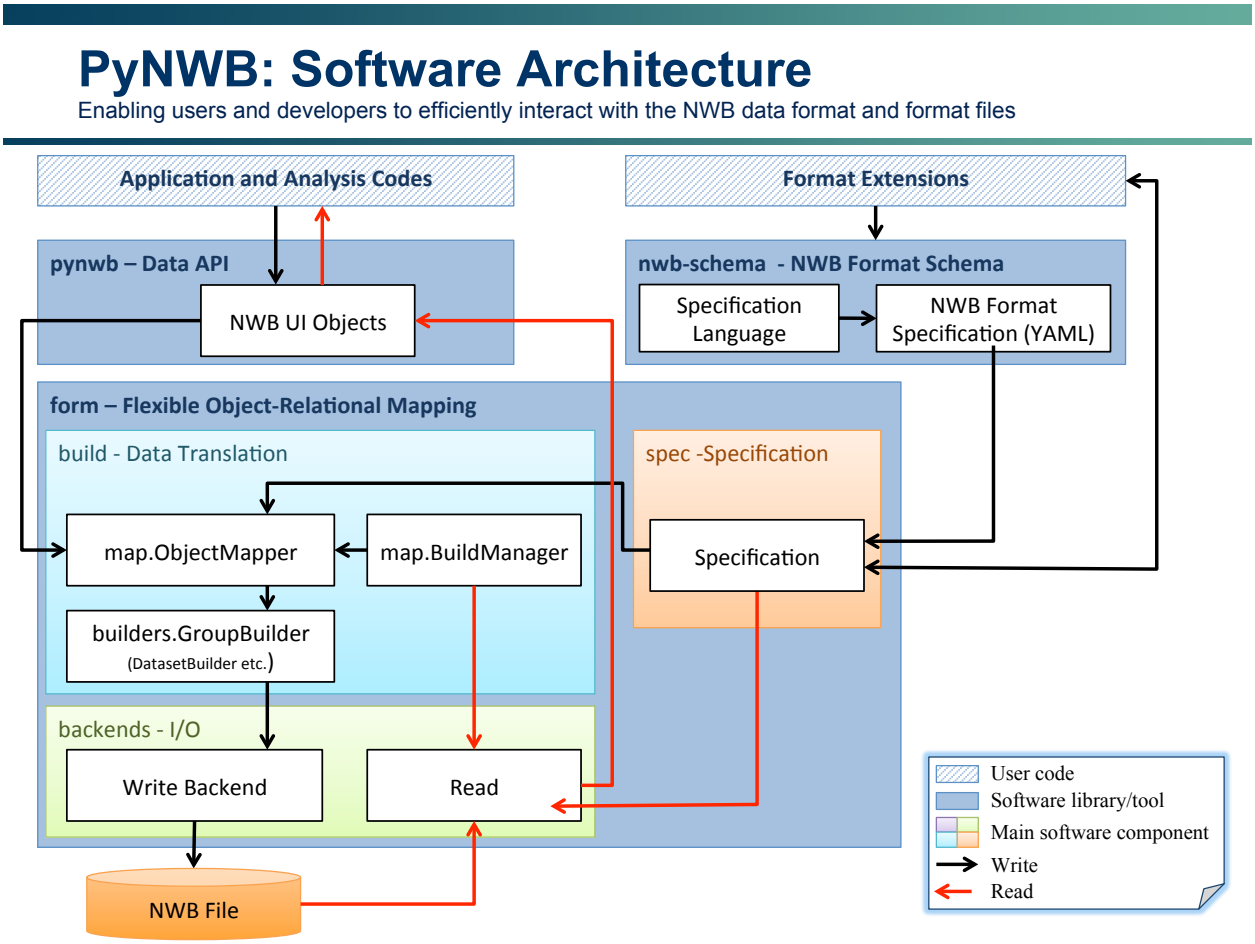

Fig. 2.1: Overview of the high-level software architecture and read/write process.

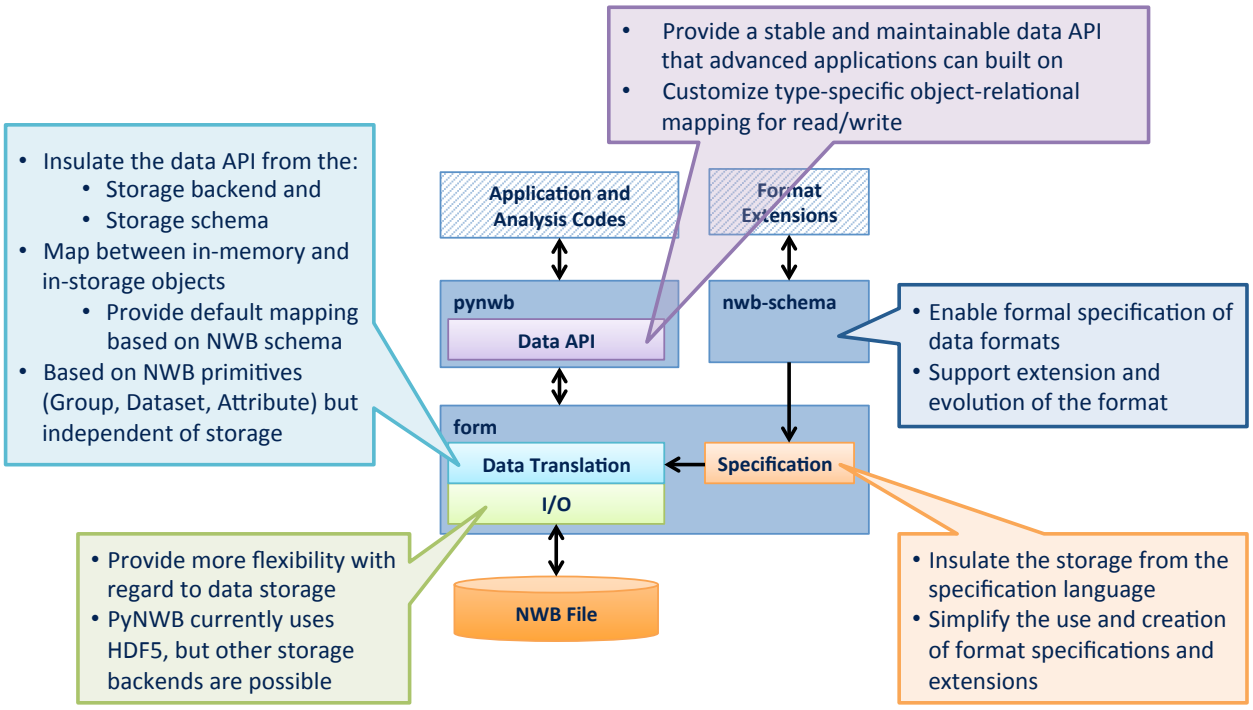

Fig. 2.2: Motivation for and function of the main high-level components of the software architecture.

CHAPTER 3

General

Getting Help

Detailed documentation of the various aspects of the NWB-N project are available here:

- **General Overview** : http://nwb-overview.readthedocs.io
- **Specification Language:** http://schema-language.readthedocs.io
- **Format Specification:** http://nwb-schema.readthedocs.io
- **Data Storage:** http://nwb-storage.readthedocs.io
- **PyNWB (APIs):** http://pynwb.readthedocs.io

The documents are also available in PDF, Epub, and zipped HTML form for print and offline browsing:

- **PDF:**
  - **General Overview** : http://readthedocs.org/projects/nwb-overview/downloads/pdf/latest/
  - **Specification Language:** http://readthedocs.org/projects/schema-language/downloads/pdf/latest/
  - **Format Specification:** http://readthedocs.org/projects/nwb-schema/downloads/pdf/latest/
  - **Data Storage:** http://readthedocs.org/projects/nwb-storage/downloads/pdf/latest/
  - **PyNWB (APIs):** http://readthedocs.org/projects/pynwb/downloads/pdf/latest/
- **Epub:**
  - **General Overview** : http://readthedocs.org/projects/nwb-overview/downloads/epub/latest/
  - **Specification Language:** http://readthedocs.org/projects/schema-language/downloads/epub/latest/
  - **Format Specification:** http://readthedocs.org/projects/nwb-schema/downloads/epub/latest/
  - **Data Storage:** http://readthedocs.org/projects/nwb-storage/downloads/epub/latest/
  - **PyNWB (APIs):** http://readthedocs.org/projects/pynwb/downloads/epub/latest/
- **HTML Zip:**

  - **General Overview** : http://readthedocs.org/projects/nwb-overview/downloads/htmlzip/latest/
  - **Specification Language:** http://readthedocs.org/projects/schema-language/downloads/htmlzip/latest/
  - **Format Specification:** http://readthedocs.org/projects/nwb-schema/downloads/htmlzip/latest/
  - **Data Storage:** http://readthedocs.org/projects/nwb-storage/downloads/htmlzip/latest/
  - **PyNWB (APIs):** http://readthedocs.org/projects/pynwb/downloads/htmlzip/latest/

Sources

The sources for the API, format specification, and all documents are available here:

- **PyNWB (APIs) https://bitbucket.org/lblneuro/pynwb . The PyNWB repository includes among others:**
  - `src/pynwb` : Sources of the PyNWB data API
  - `src/form` : Sources of the Flexible Object-Relational Mapping API
  - `docs/` : Sphinx documentation of the APIs
  - `tests/` : Unit test suite for the APIs
- **NWB Schema https://bitbucket.org/lblneuro/nwb-schema . The nwb-schema repositry includes among others:**
  - `core` : YAML specification of the NWB core format
  - `docs/general` : Sphinx sources for the general overview documentation
  - `docs/language` : Sphinx sources for the specification language documention
  - `docs/format` : Sphinx sources for the format specification documentation
  - `docs/storage` : Sphinx sources for the data storage documention
  - `docs/utils` : Python utilities used for generation of the format documentation from the YAML specification. This includes convenient helper functions for rendering specification hierarchies and for generating RST docs.
  - The original spec and sources for converting the original spec to the new format are included as part of `original` and `bin`

Reporting Issues

Attention: TODO: Add documentation on how to report issues for the various aspects of the project

Contributing to NWB

Attention: TODO: Add documentation on how to contribute to different parts of NWB

CHAPTER 4

Credits

Contents

Authors

NWB-N: Version 1.1.x and later

Version 1.1.x of the NWB-N format, Version 1.2.x and later of the specification language, and Version 1.x and later of the NWB storage documents have been created by Oliver Ruebel and Andrew Tritt et al. in collaboration with original members of the NWB pilot project at the University of California in Berkeley and San Francisco and at the Allen Institute for Brain Science.

NWB-N: Version 1.0.x and earlier

Version 1.0.5g (and earlier) of the NWB file format and Version 1.1c (and earlier) of the specification language were created by Jeff Teeters et al. as part of the first NWB pilot project. The documents governing the specification language and format have been adopted from the final versions released by the original NWB pilot project.

Acknowledgments

2016 - 2017

Note: #TODO Add Acknowledgement for LBL Kavli project and acknowledge other projects as well

2015 - 2016

The Neurodata Without Borders: Neurophysiology Initiative is funded by GE, the Allen Institute for Brain Science, the Howard Hughes Medical Institute (HHMI), The Kavli Foundation and the International Neuroinformatics Coordinating Facility. Our founding scientific partners are the Allen Institute, the Svoboda Lab at the Janelia Research Campus of HHMI, the Meister Lab at the California Institute of Technology, the Buzsaki Lab at New York University School of Medicine, and the University of California, Berkeley. Ovation.io is our founding development partner. Ken Harris at University College London provided invaluable input and advice.

Copyright

Licence

3. Neither the name of the University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search