

Maintaining reproducible data workflows while keeping data in sync, backed up, and easily accessible from within and outside the lab is a key challenge in research. To minimize time and effort scientists have to spend on these tasks, we provide a suite of tools designed for comprehensive and versioned management of scientific data including convenient storage of data, analysis and metadata annotation for easy reproducibility, data sharing and re-usability.



Collect and manage all information about your experiment

Main features

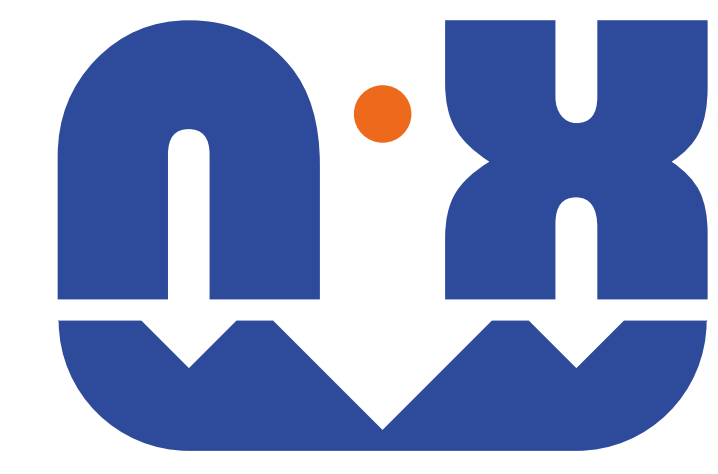
- Open metadata format [1]
- Flexible hierarchical key-value storage
- Save to common structured formats: XML, JSON, YAML
- Terminology repository [2] for reusable definitions
- Template system for reusable metadata structures

Graphical user interfaces

- odml-ui [3]: browsing and template editing
- odmltables [4]: filtering and tabular export
- Available for macOS and Linux

Export to RDF retaining your own terms

- Query metadata using semantic web technologies
- Search cross document via SPARQL queries



Manage data and metadata in one open, versatile format

Main features

- Open data format
- Store data, analysis results, and metadata conveniently in the same file
- Descriptive associations between data, analysis results, and metadata



- Free open source libraries for multiple programming languages: [5] C++, Python, Matlab, Java
- NIX IO for Neo [6]
- Interoperability with Neo compatible tools, e.g., the Elephant toolkit [7]

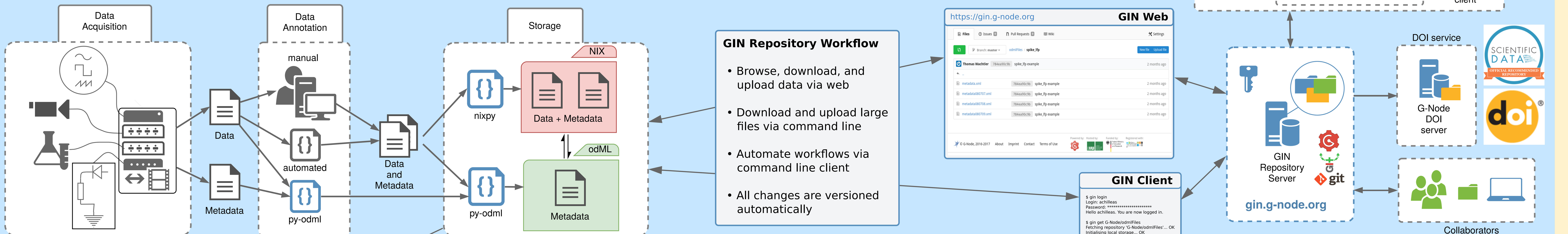


Store data securely; publish and collaborate with ease

Main features

- Access data from any location
- Free storage for scientific data [9]
- 'Built-in' versioning (built on git [10])

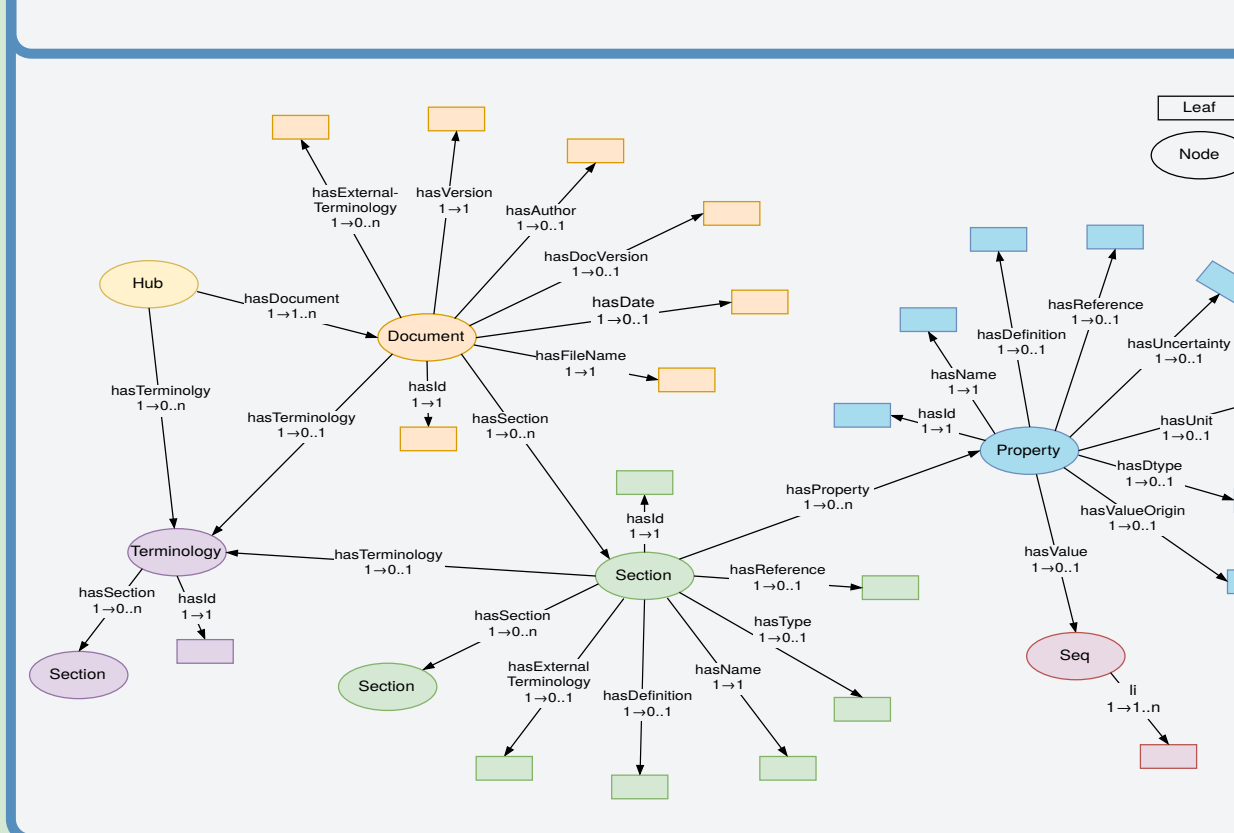
- Platform independent
- Secure access
- Public and private repositories
- Citable data by DOIs



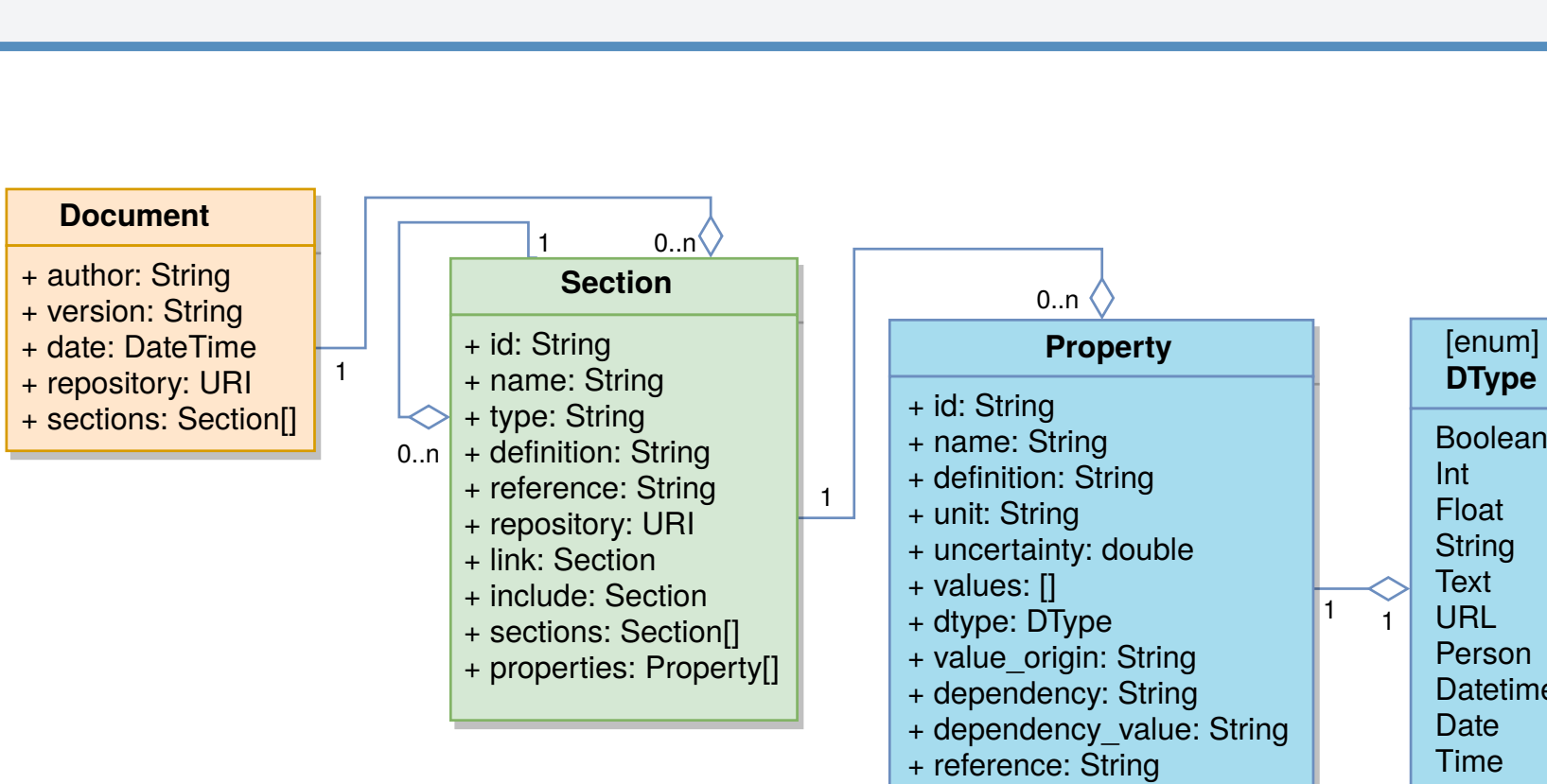
meta.g-node.org

- access diverse metadata datasets
- all datasets are publicly available
- searchable by SPARQL via API and web

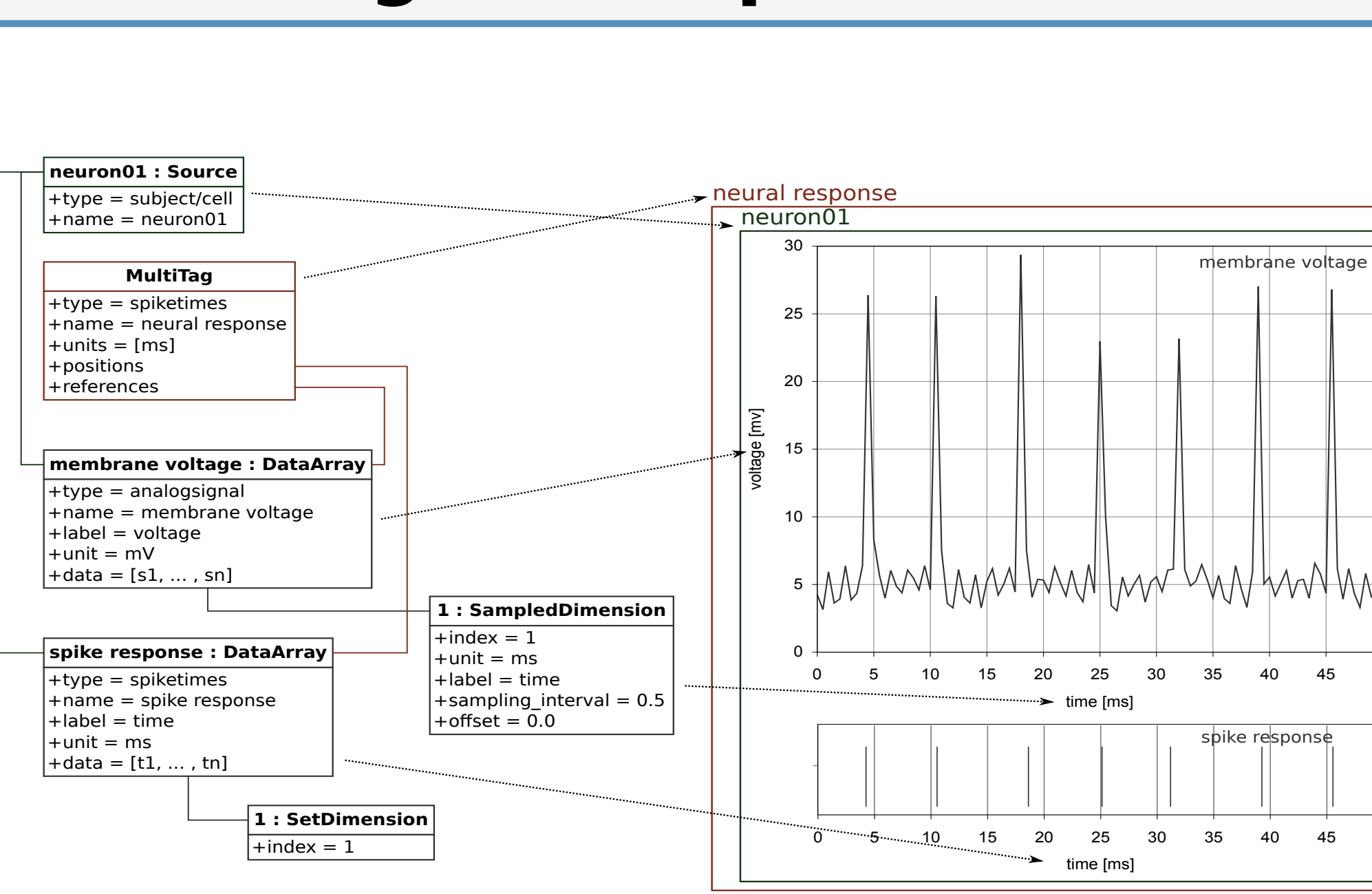
odML RDF schema



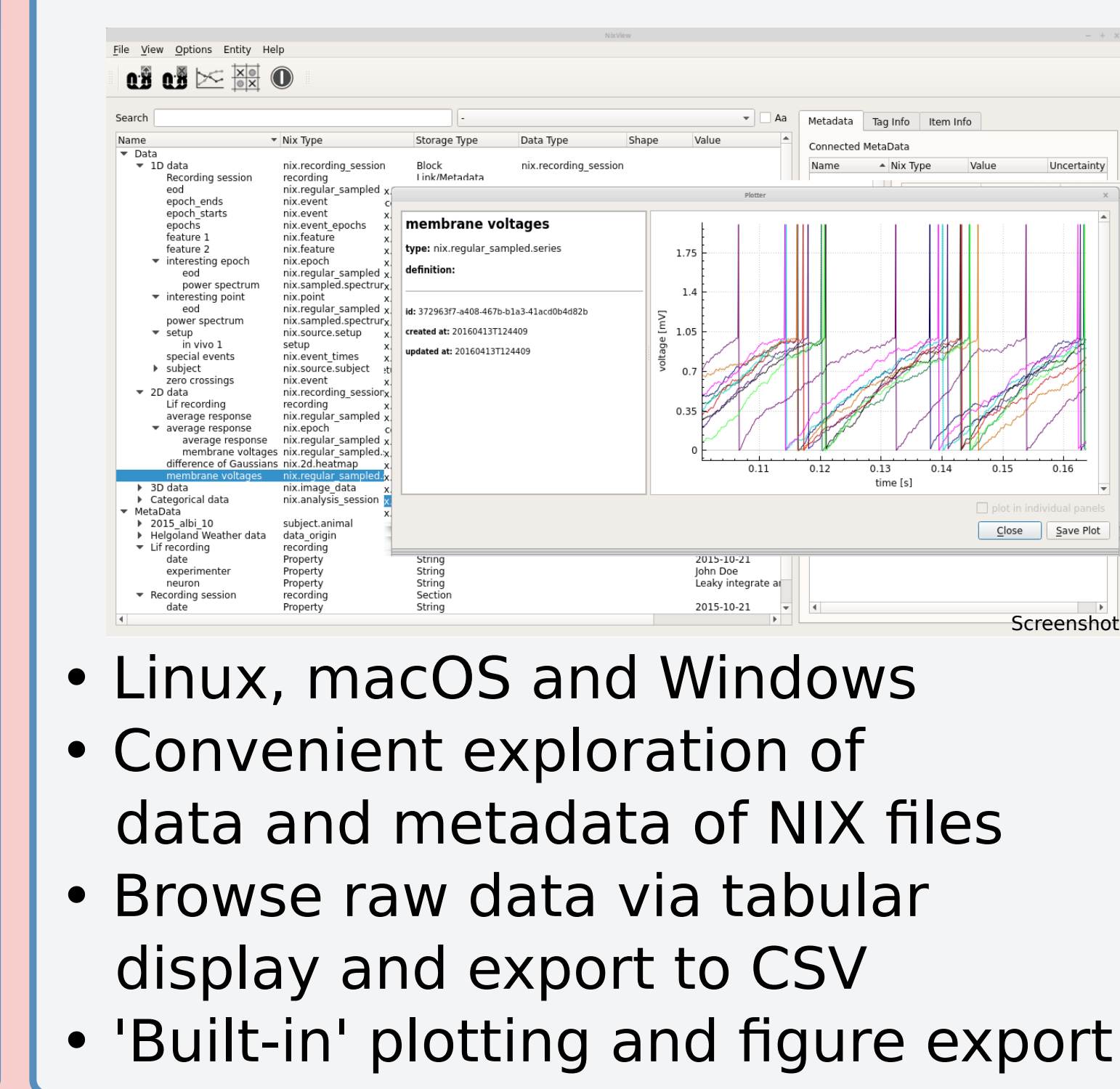
The odML Metadata format



NIX usage example



NixView GUI [8]



GIN Client

```
$ gln login
Login: achilles
Password: *****
Hello achilles. You are now logged in.

$ gln get G-Node/odmlFiles
Fetching repository 'G-Node/odmlFiles'... OK
Initialising local storage... OK

$ gln ls
Synced:
  metadata.xml
  metadata080707.xml
  $
```

Upcoming features

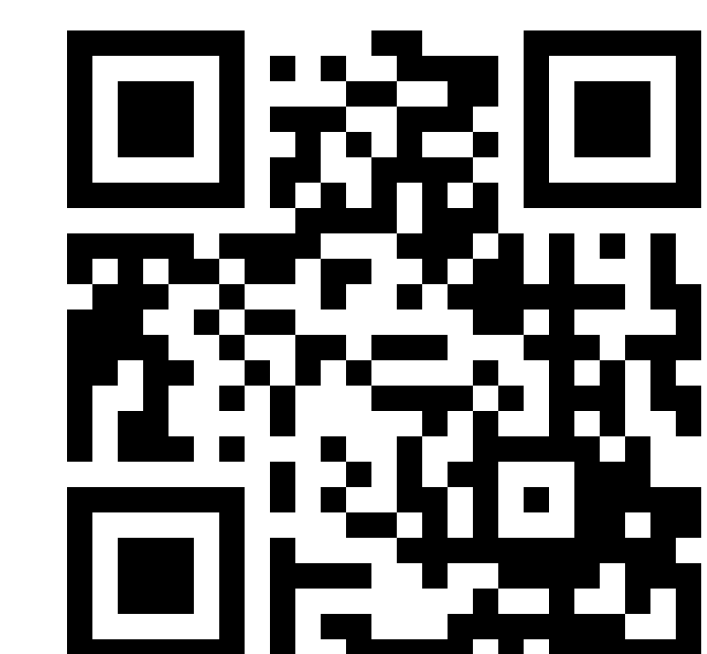
- Format validation
- BIDS, odML, NIX, custom formats
- CI for scientific data: automated tests for scripts and data integrity
- automated export of odML to RDF

Collaboration

- User management
- User Access Levels
- On and offline collaboration

Resources

- [1] Grewe et al (2011), doi:10.3389/fninf.2011.00016
- [2] <http://www.g-node.org/projects/odml/terminologies>
- [3] <https://github.com/G-Node/odml-ui>
- [4] <https://github.com/INM-6/python-odmltables>
- [5] <http://g-node.github.io/nix>
- [6] <http://neuralensemble.org/neo>
- [7] <http://neuralensemble.org/elephant>
- [8] <http://bendalab.github.io/NixView>
- [9] <https://gin.g-node.org>
- [10] <https://git-scm.com>



Contact: dev@g-node.org

Supported by BMBF grants
01GQ1302, 01GQ1509

