The Neurodata Without Borders Ecosystem for FAIR Neurophysiology Data

Driving Collaboration in Neuroscience

Oliver Rübel

Computational Biosciences Group
Scientific Data Division
Lawrence Berkeley National Laboratory

BERKELEY LAB
Bringing Science Solutions to the World

Contents

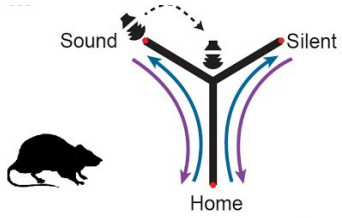
- Motivation
- What is NWB?
- NWB Data Analysis, Visualization, and Management Tools
- Core NWB Data Standard Ecosystem
- The Energy Barrier in Data Standardization

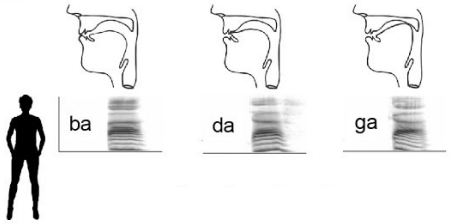
Motivation:

- Data are expensive to collect
  - Neuroscience datasets are often rich and can be used to answer many different questions
  - Reusing data saves money and time and reduces animal use
- Sharing neurophysiology data within a lab and with collaborators is tedious
  - Everyone has a different way of storing their data
  - Metadata required for data interpretation is often missing or stored ad-hoc
- Scientific results and analyses are hard to reproduce and compare
  - Processing, analyzing, and visualizing neurophysiology data requires converting data from one esoteric format to another
  - Metadata is inconsistent and often missing

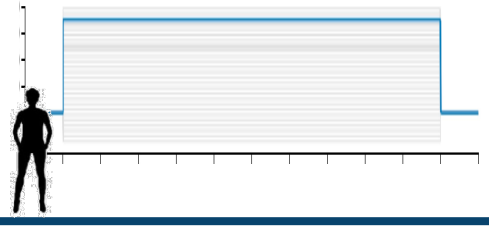
Challenge: Diversity of experiments and rate of innovation

Species & Task

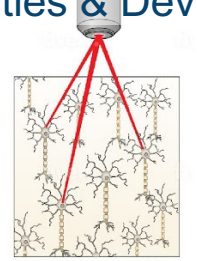


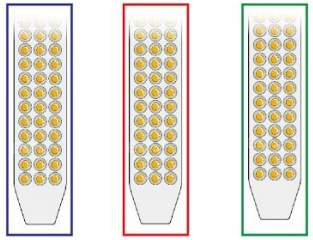


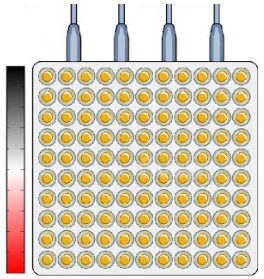

```
run Simulation
     (species == mouse,
      area == hippocampus,
      #neurons == N)
        while 1, t++
              do Vm(:,t) =
                    NEURON(N, Vm(t-1))
```

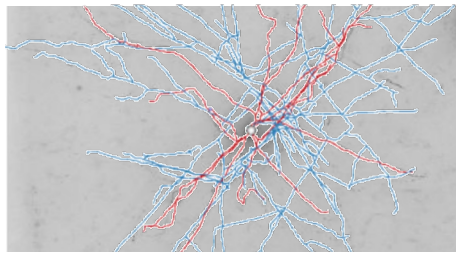


Modalities & Devices

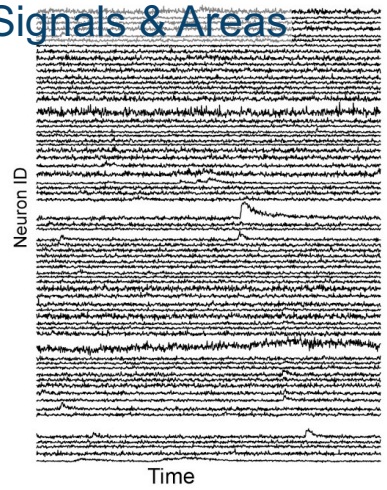


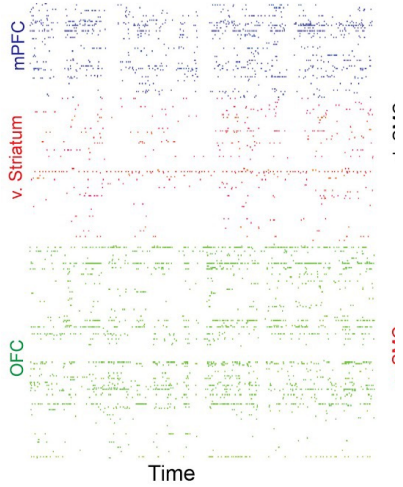


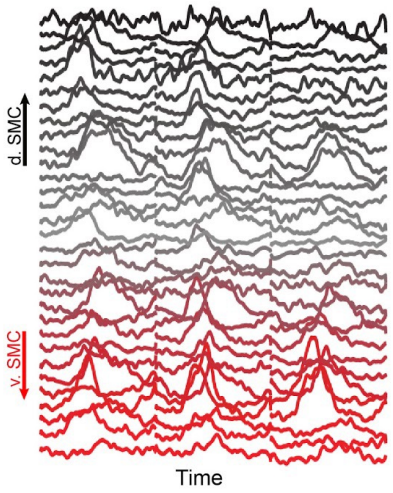


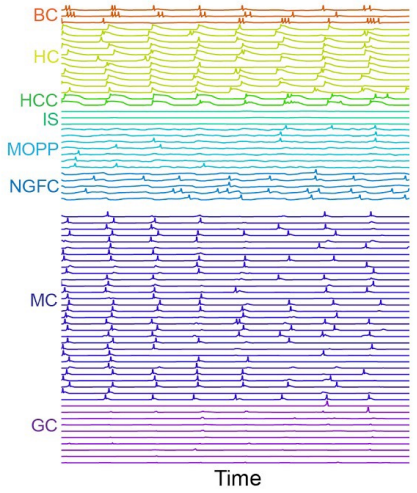


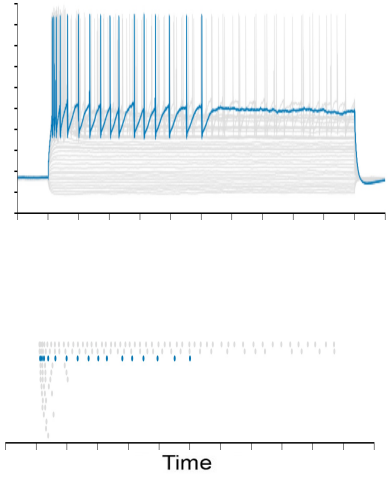
Signals & Areas









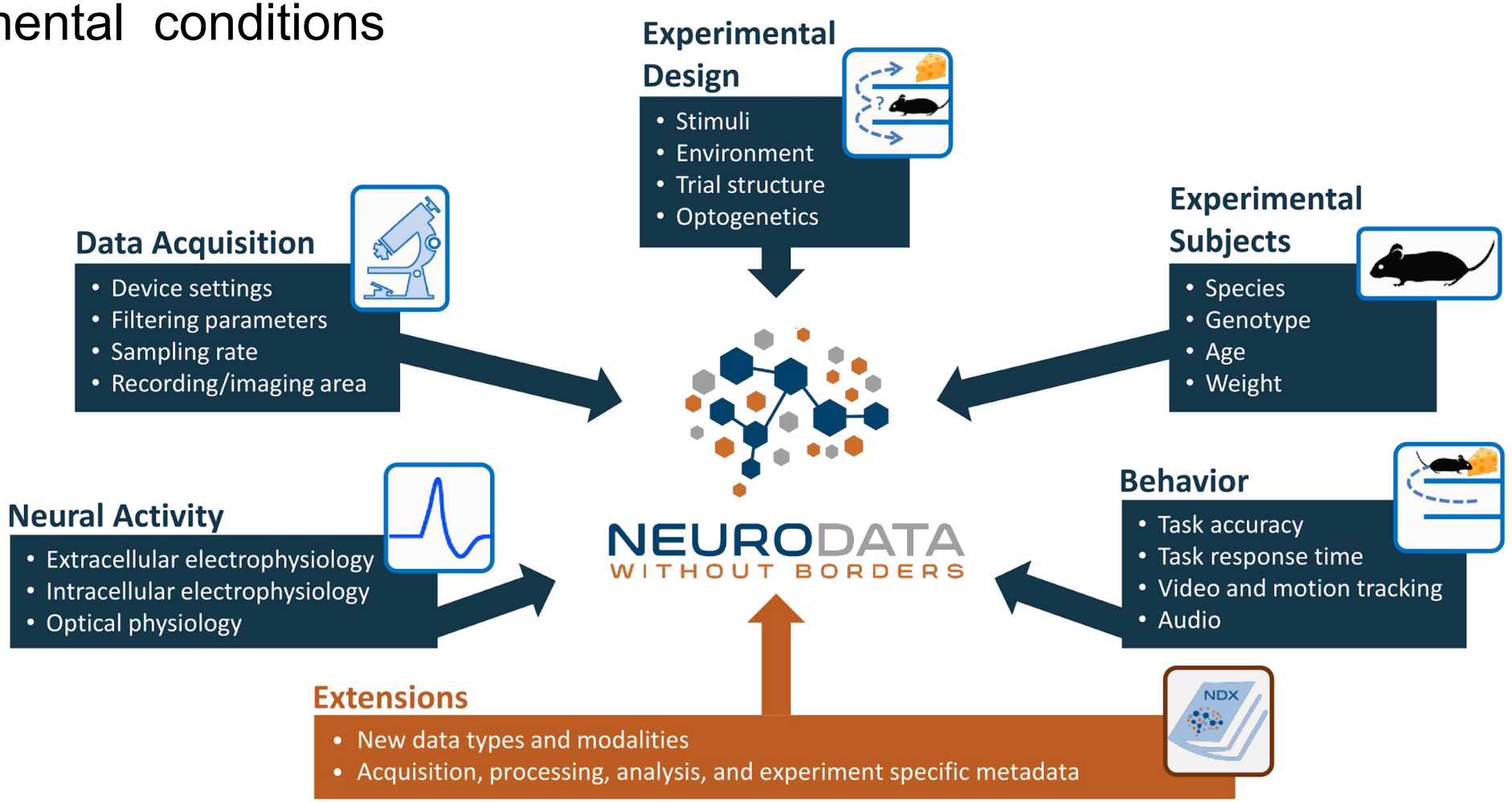


What is NWB?

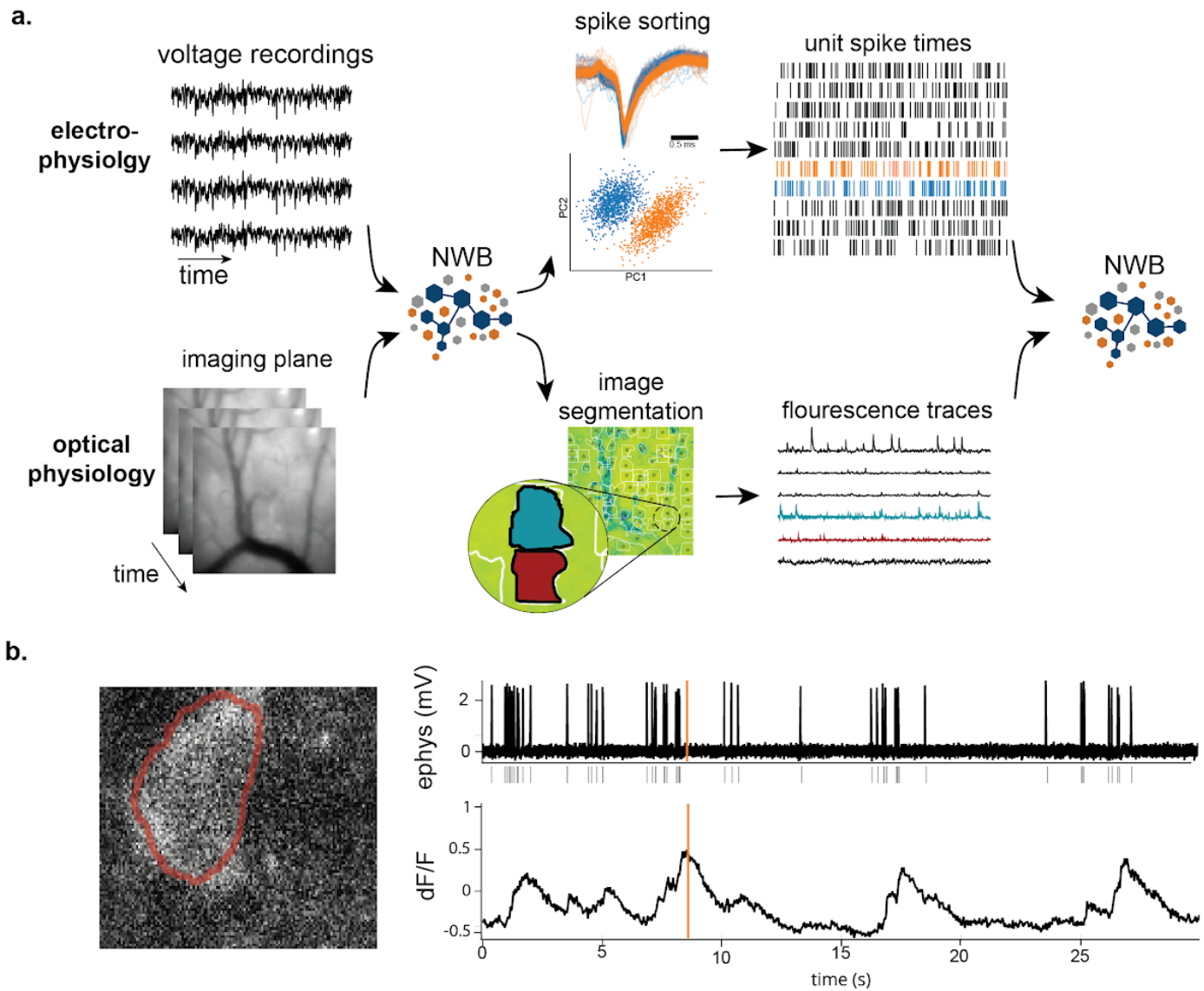
An Ecosystem for Neuroscience Data Standardization

A unified data standard for neurophysiology

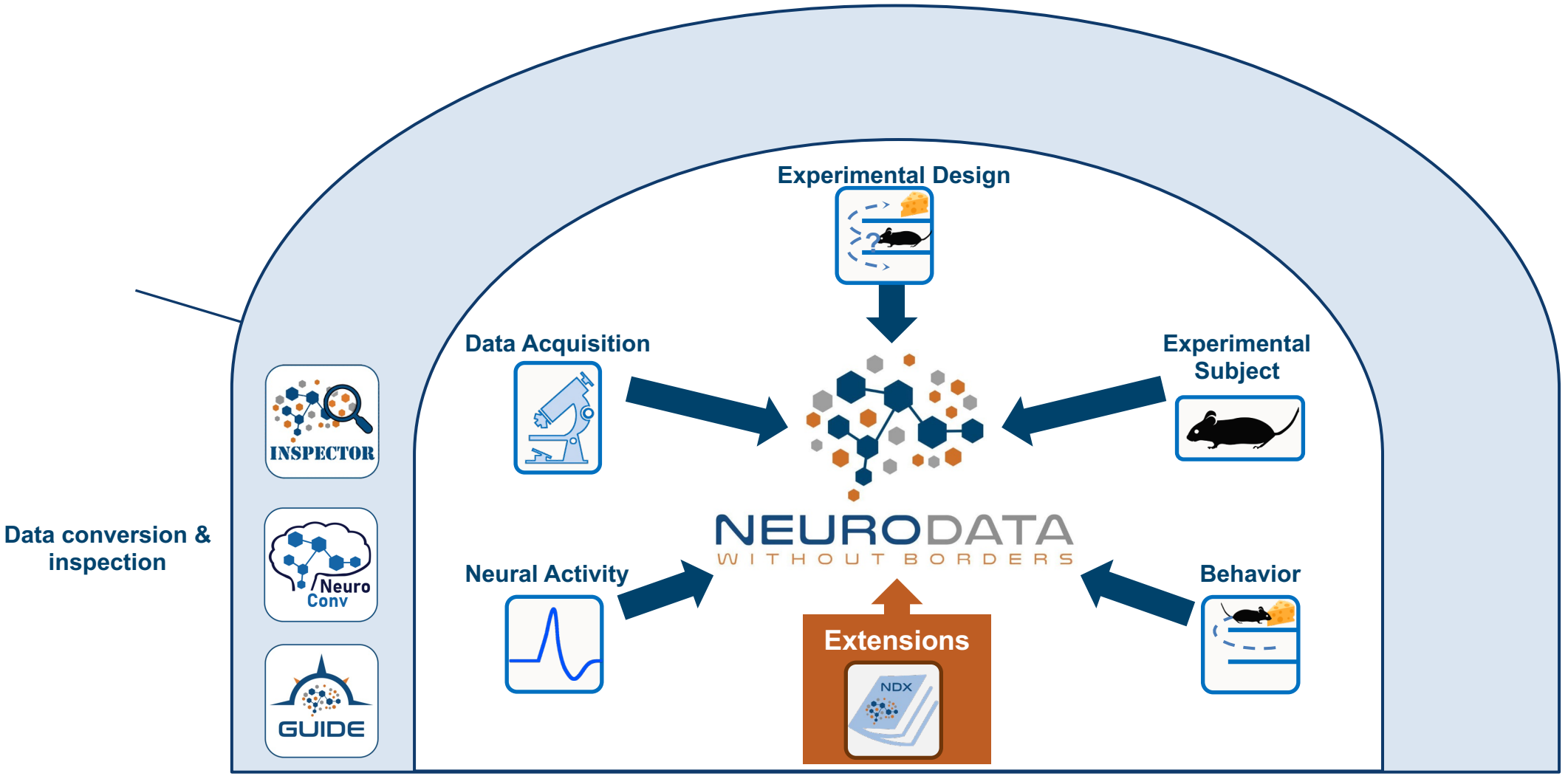
The NWB data standard defines a unified data format for neurophysiology data, focused on the dynamics of groups of neurons measured under a large range of experimental conditions



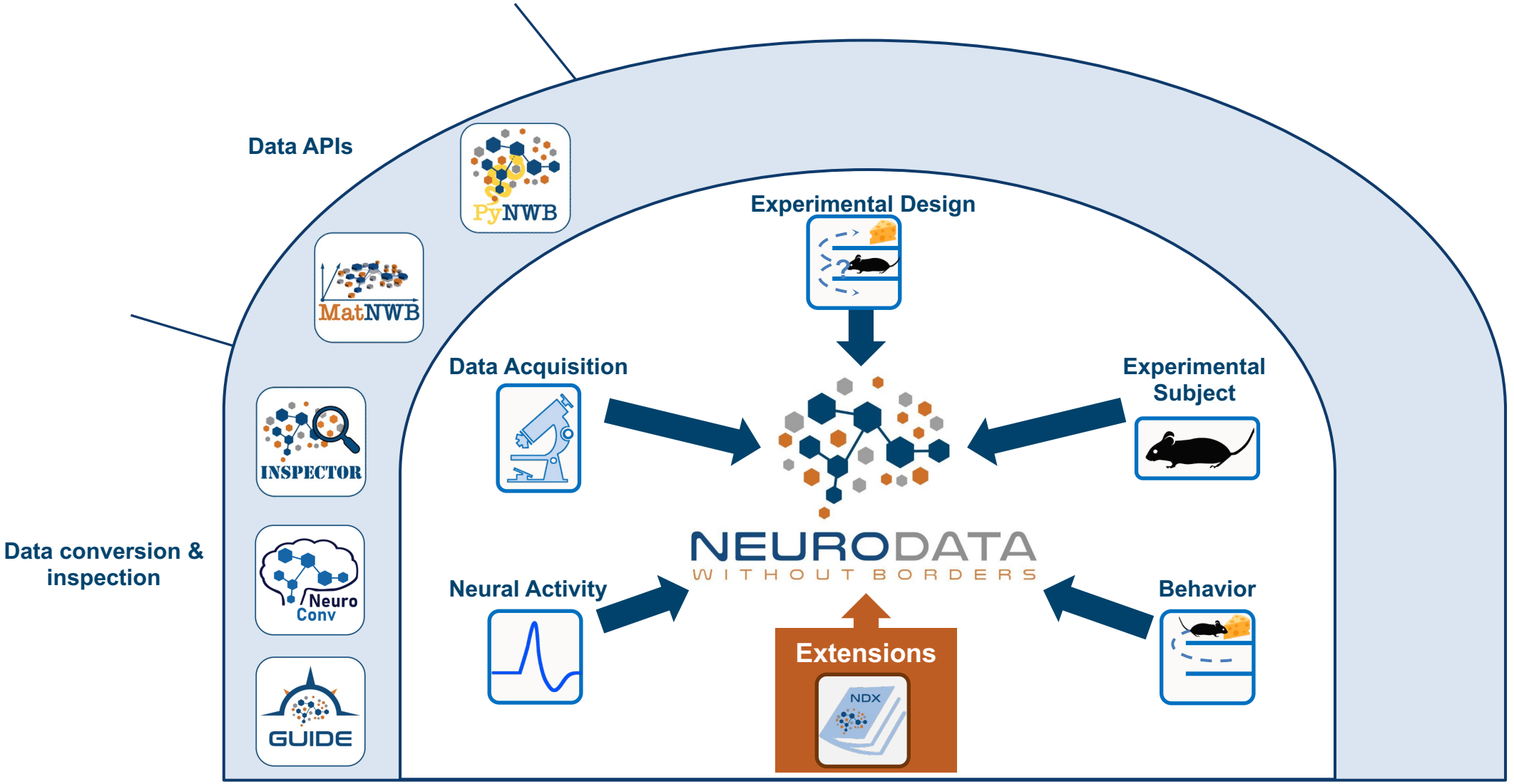
NWB enables unified storage of multimodal raw and processed data



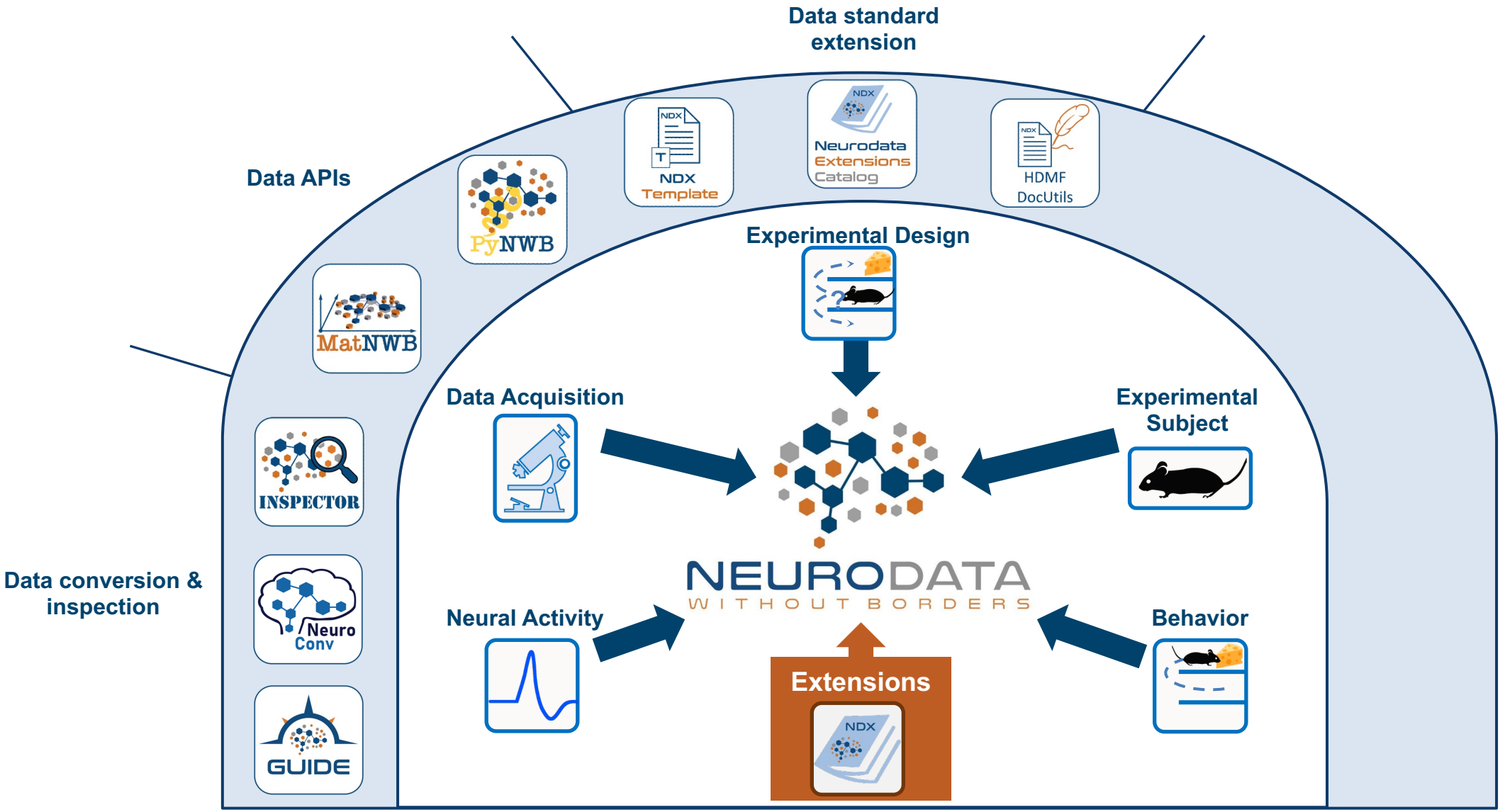
An ecosystem for neuroscience data standardization



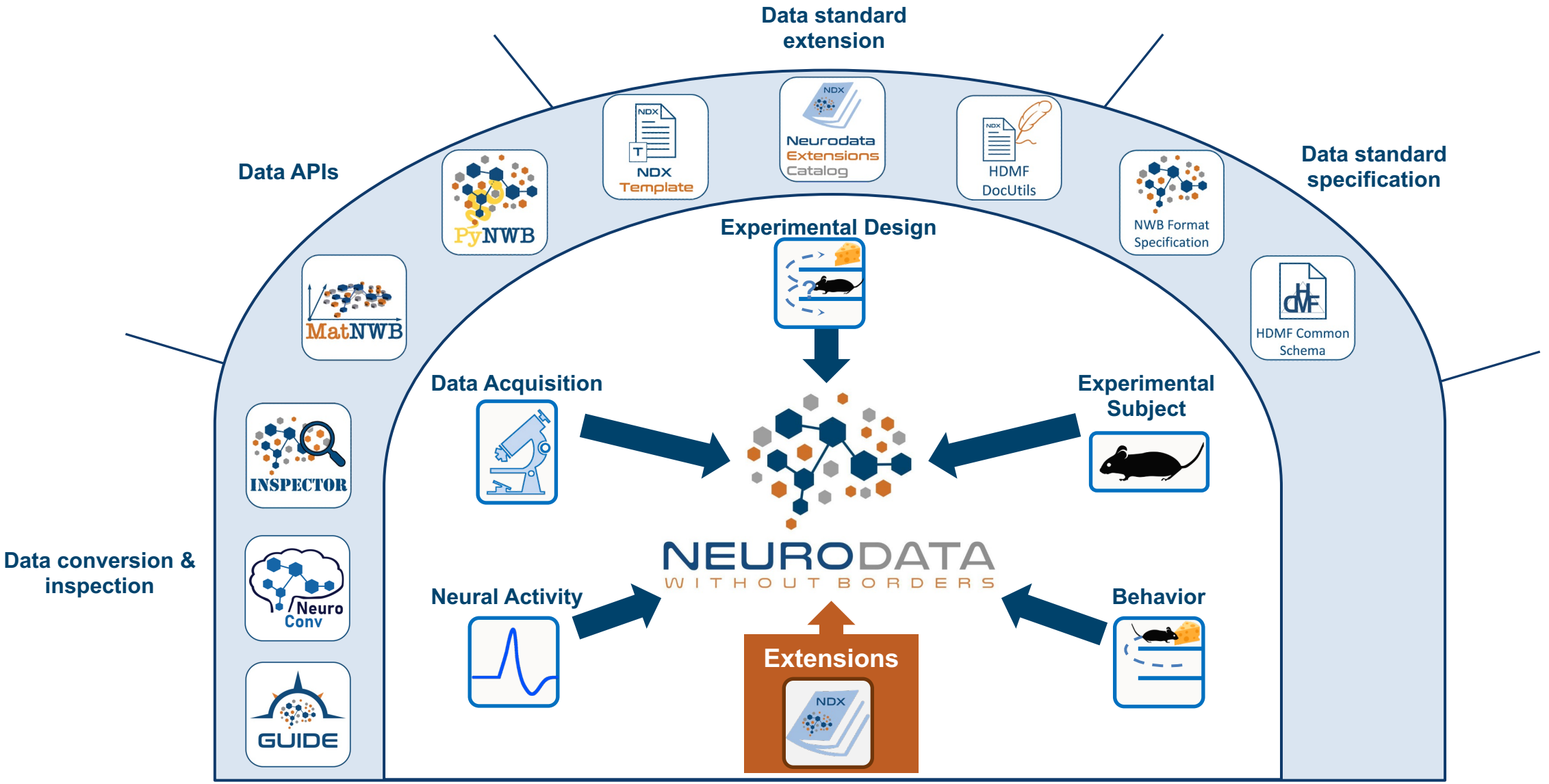
An ecosystem for neuroscience data standardization



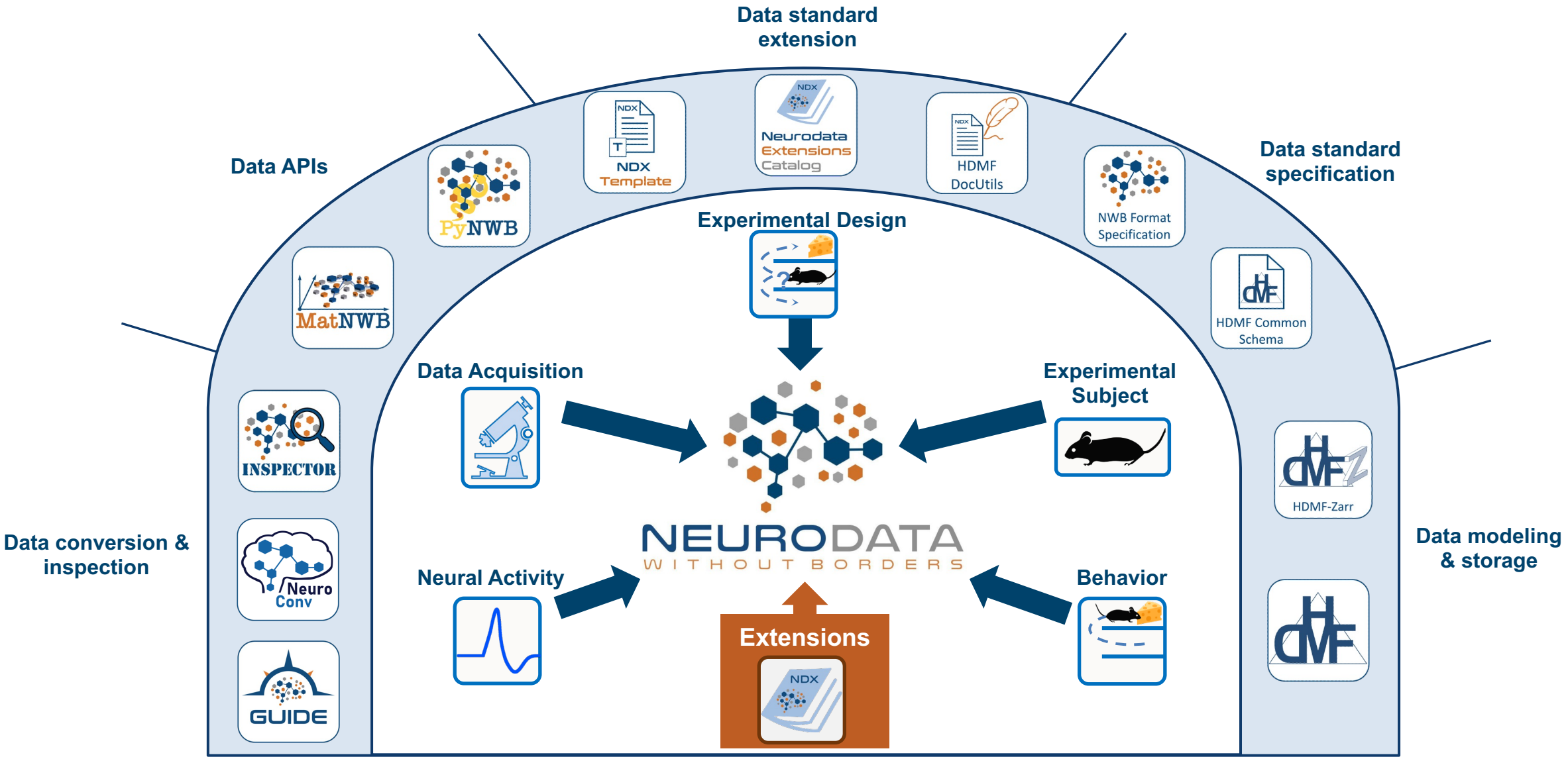
An ecosystem for neuroscience data standardization



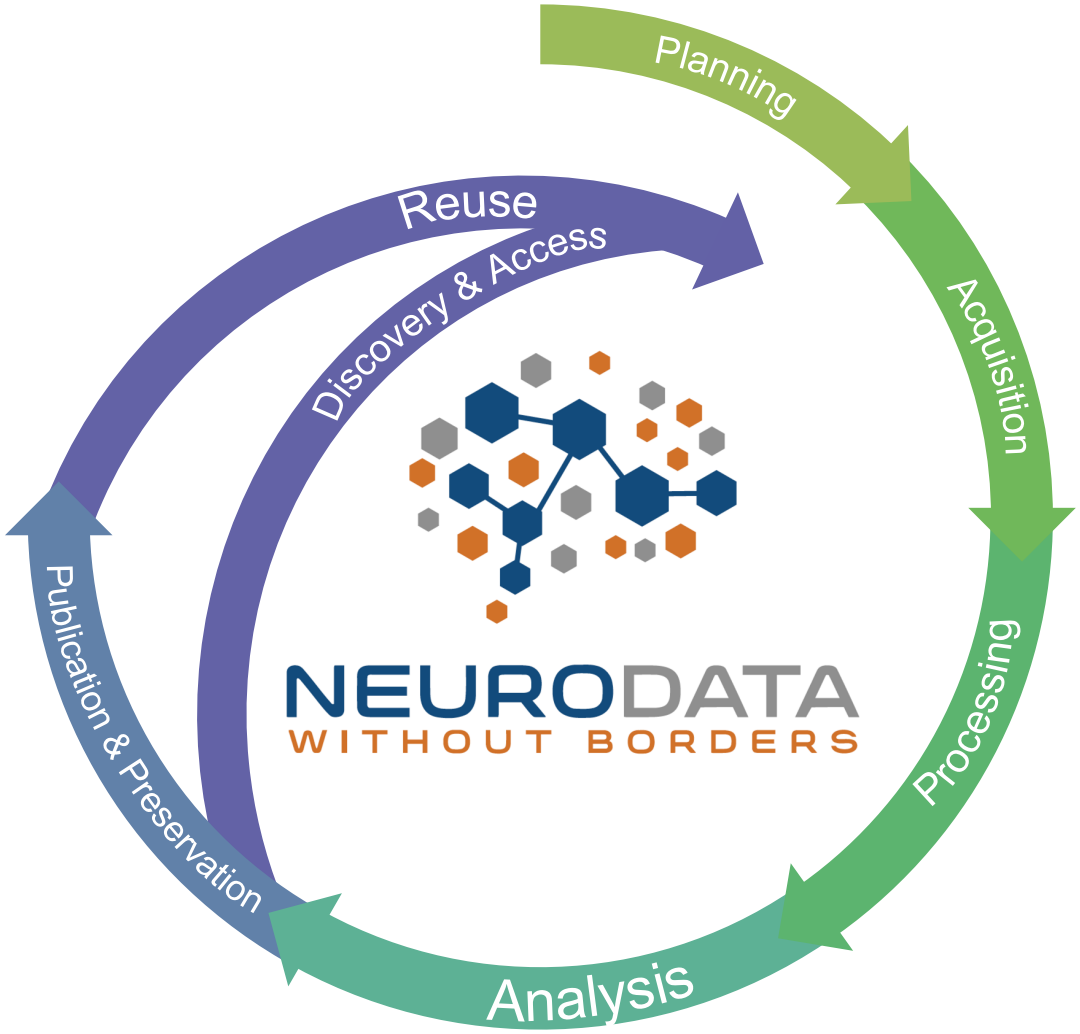
An ecosystem for neuroscience data standardization



An ecosystem for neuroscience data standardization



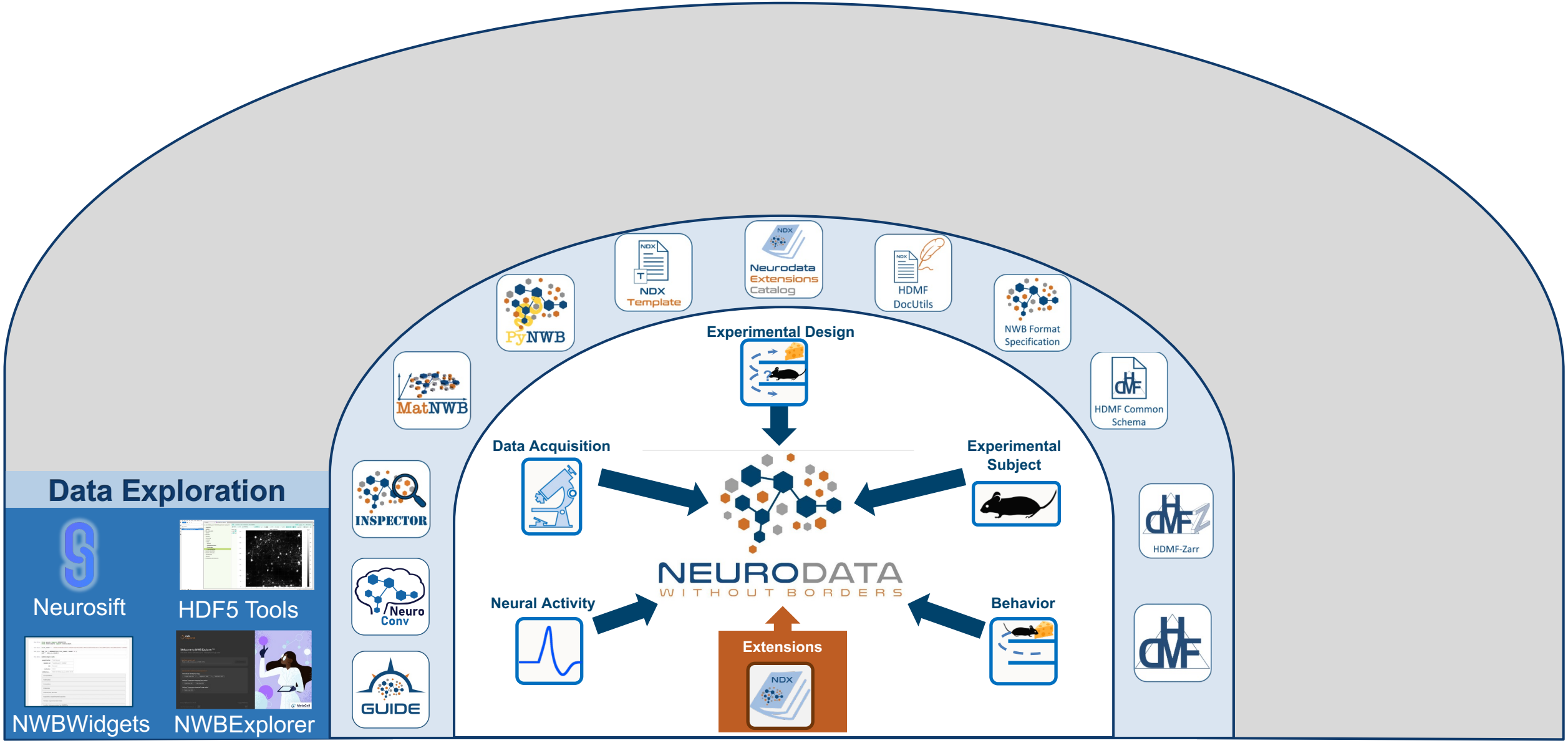
NWB technologies at the heart of the neurodata lifecycle



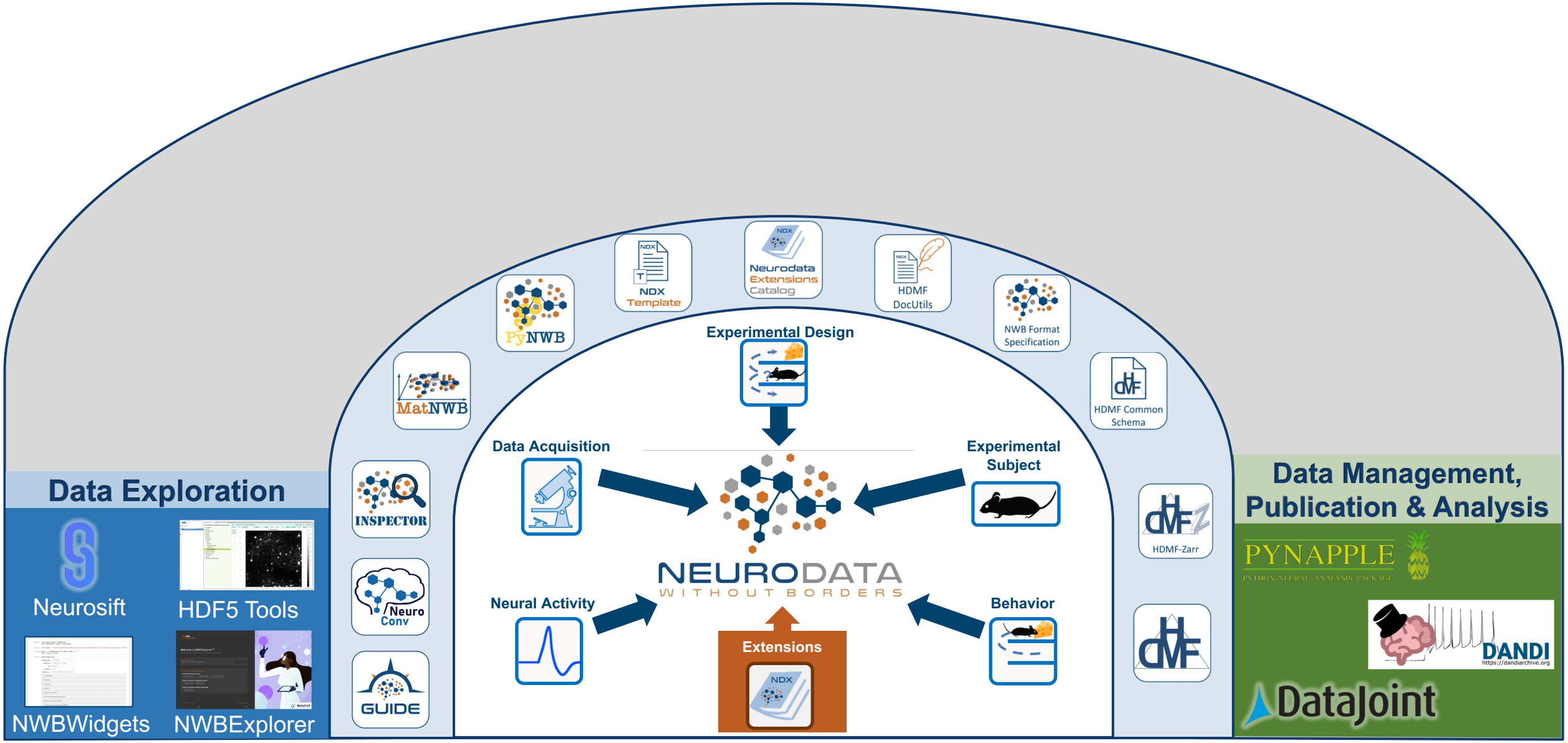
- Data standards are a critical conduit that facilitate:
  - Flow of data throughout the data lifecycle
  - Integration of data and software across phases of the data lifecycle
    - NWB needs to support the needs of and integrate with technologies across the data lifecycle:
      - Work with (not compete with) existing and emerging data technologies
- NWB is a data standard for (not a standard of) neurophysiology experiments

NWB Data Analysis, Visualization, and Management Tools

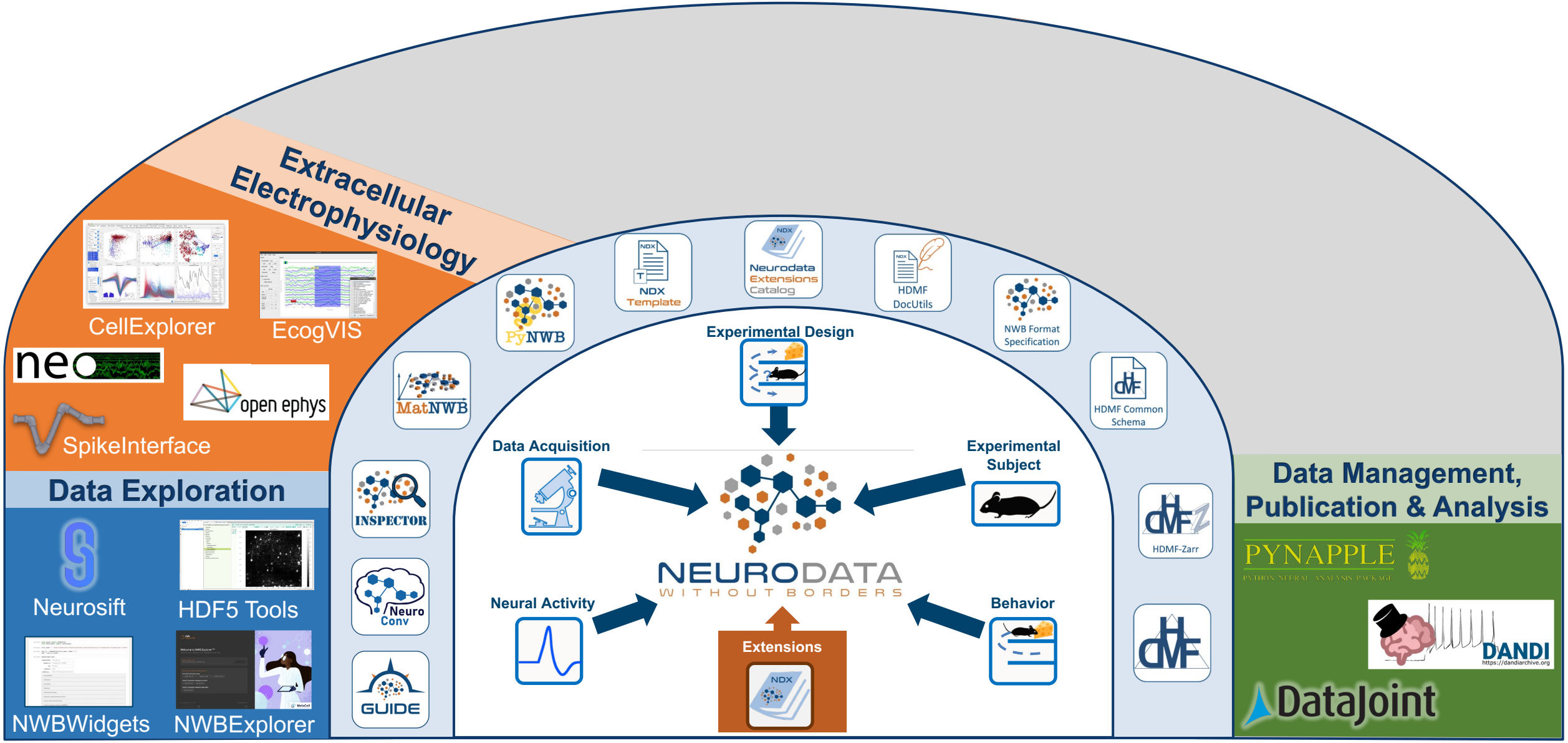
A unified data standard and software ecosystem for neurophysiology



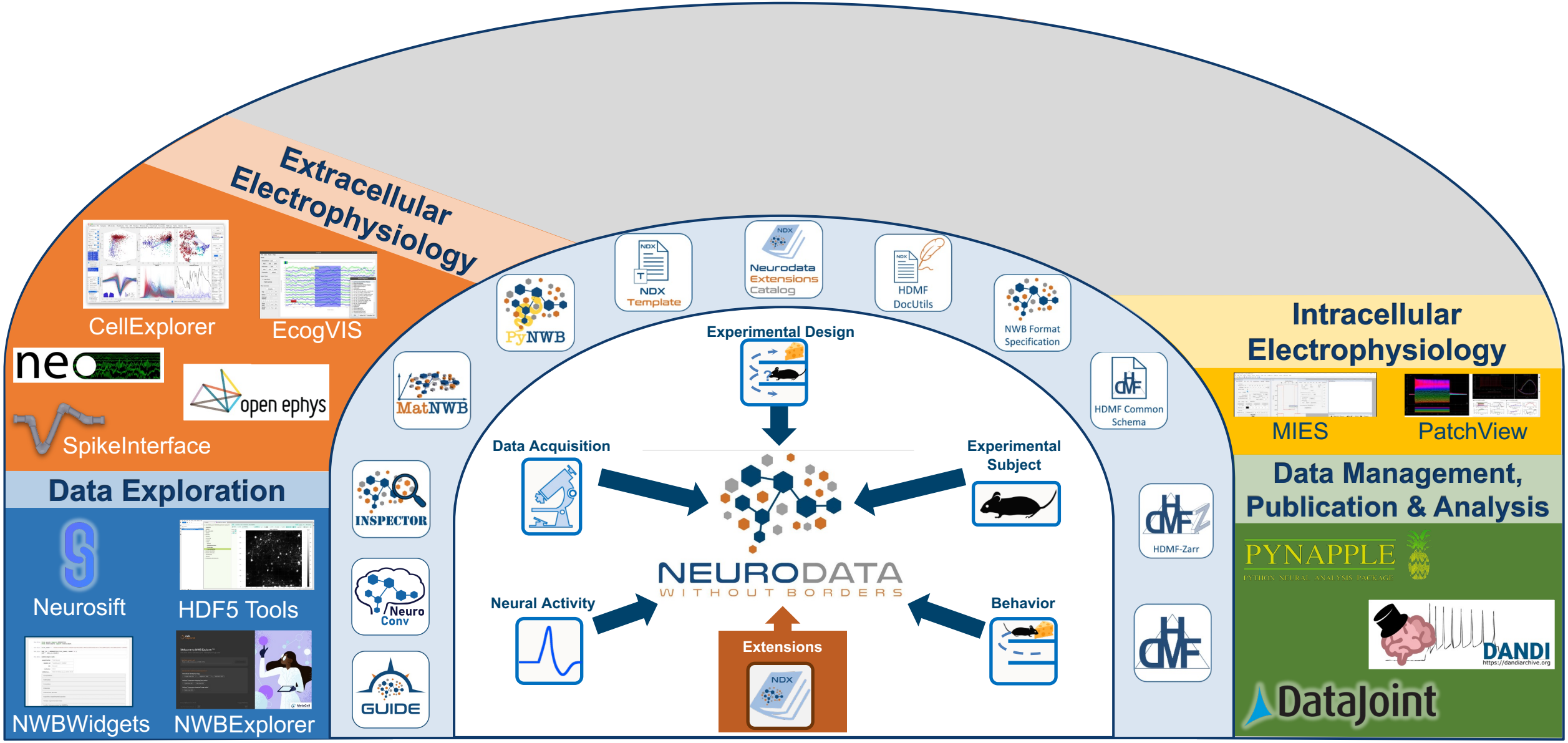
A unified data standard and software ecosystem for neurophysiology



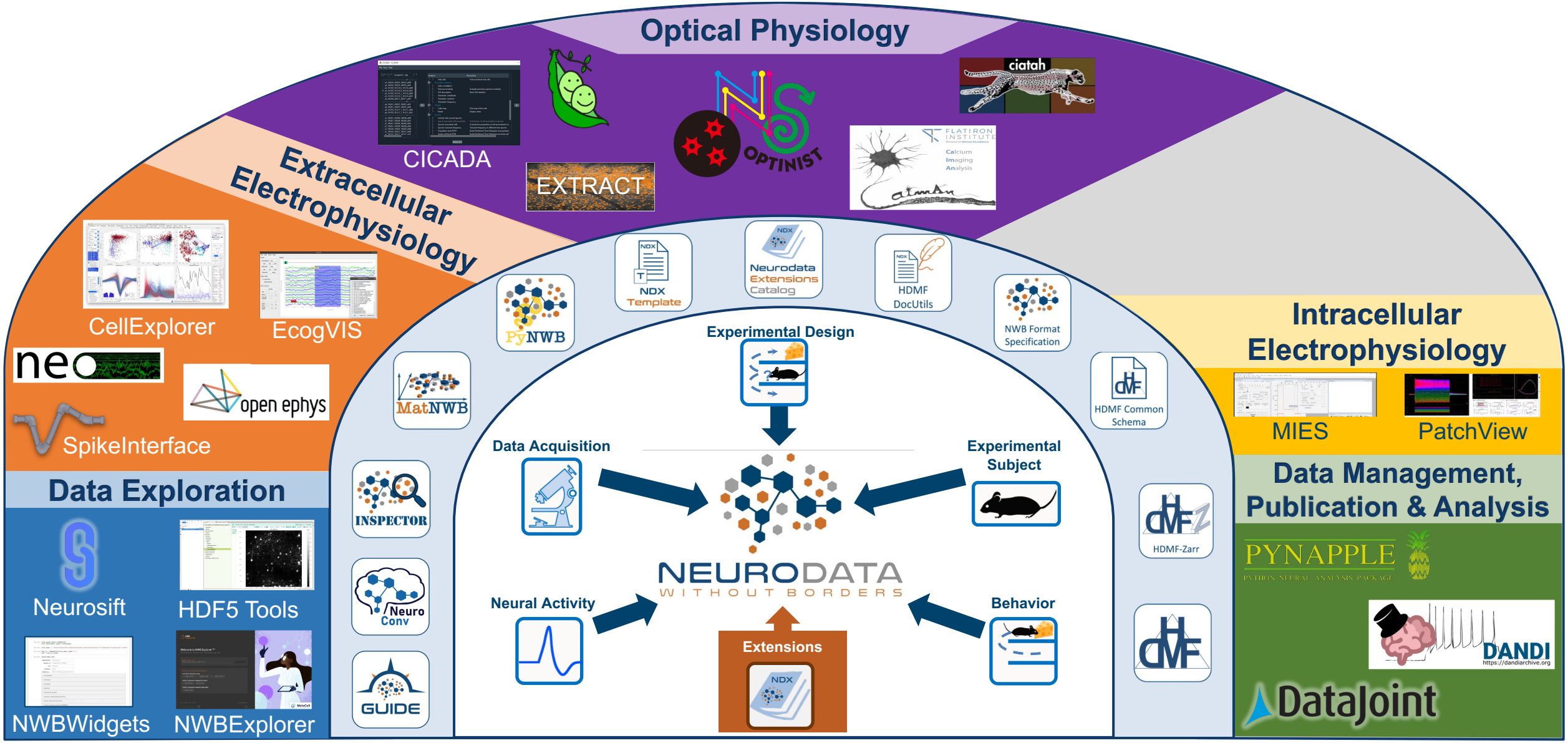
A unified data standard and software ecosystem for neurophysiology



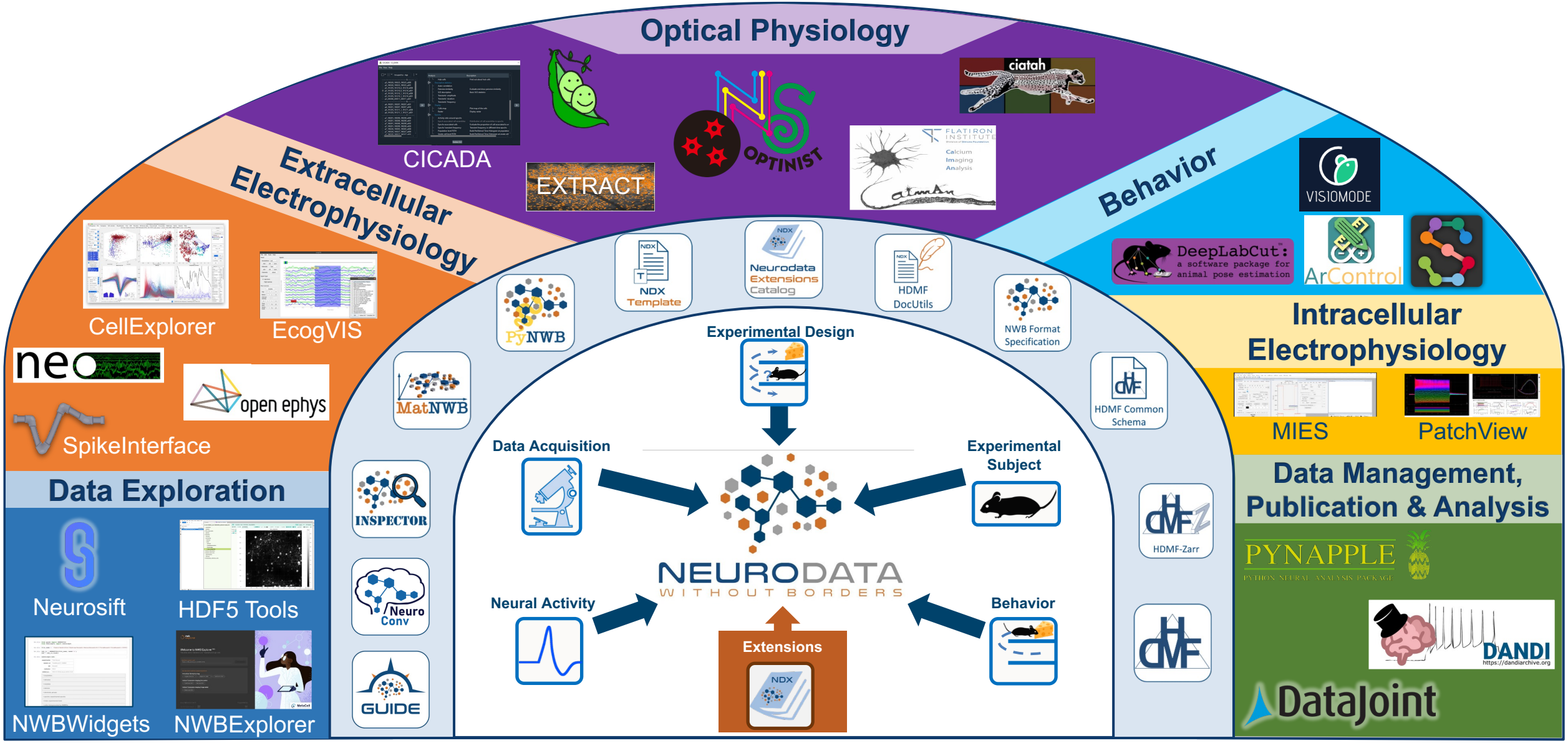
A unified data standard and software ecosystem for neurophysiology



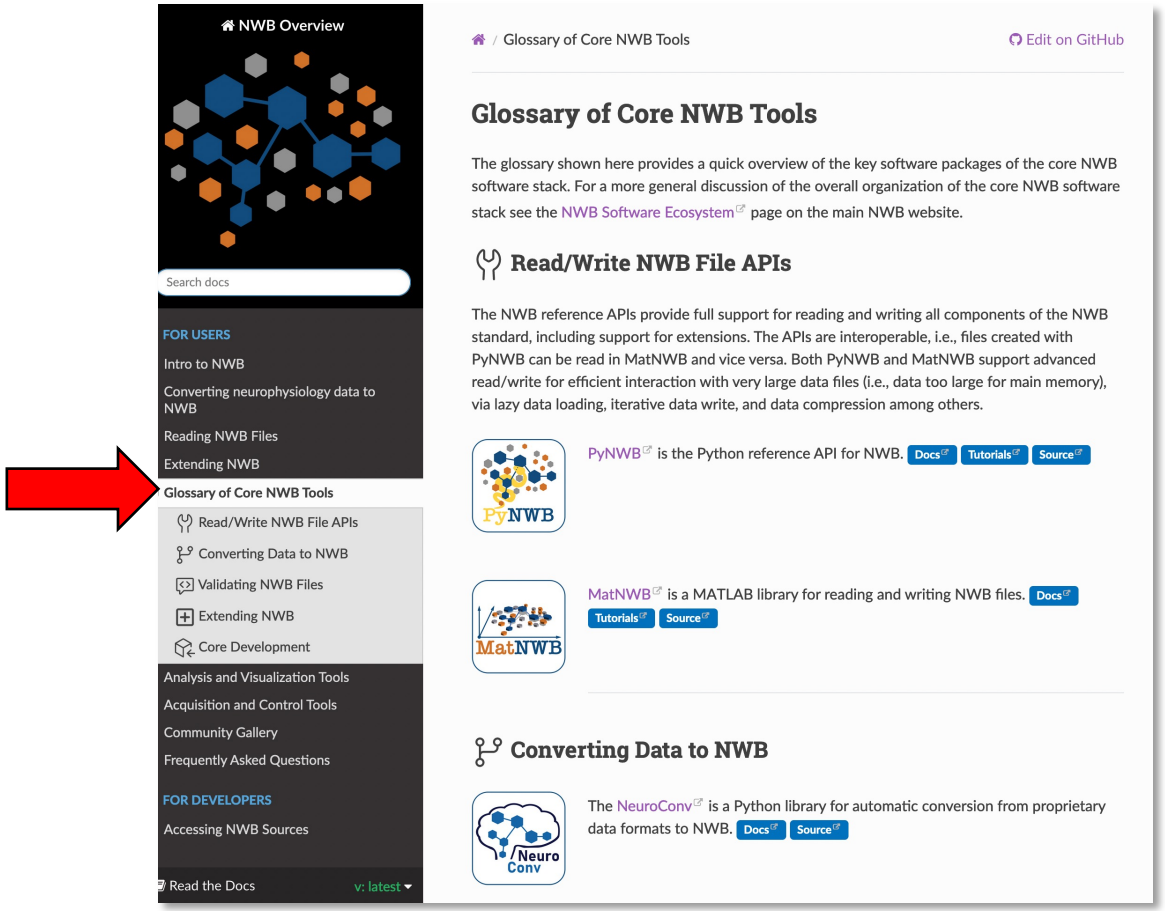
A unified data standard and software ecosystem for neurophysiology

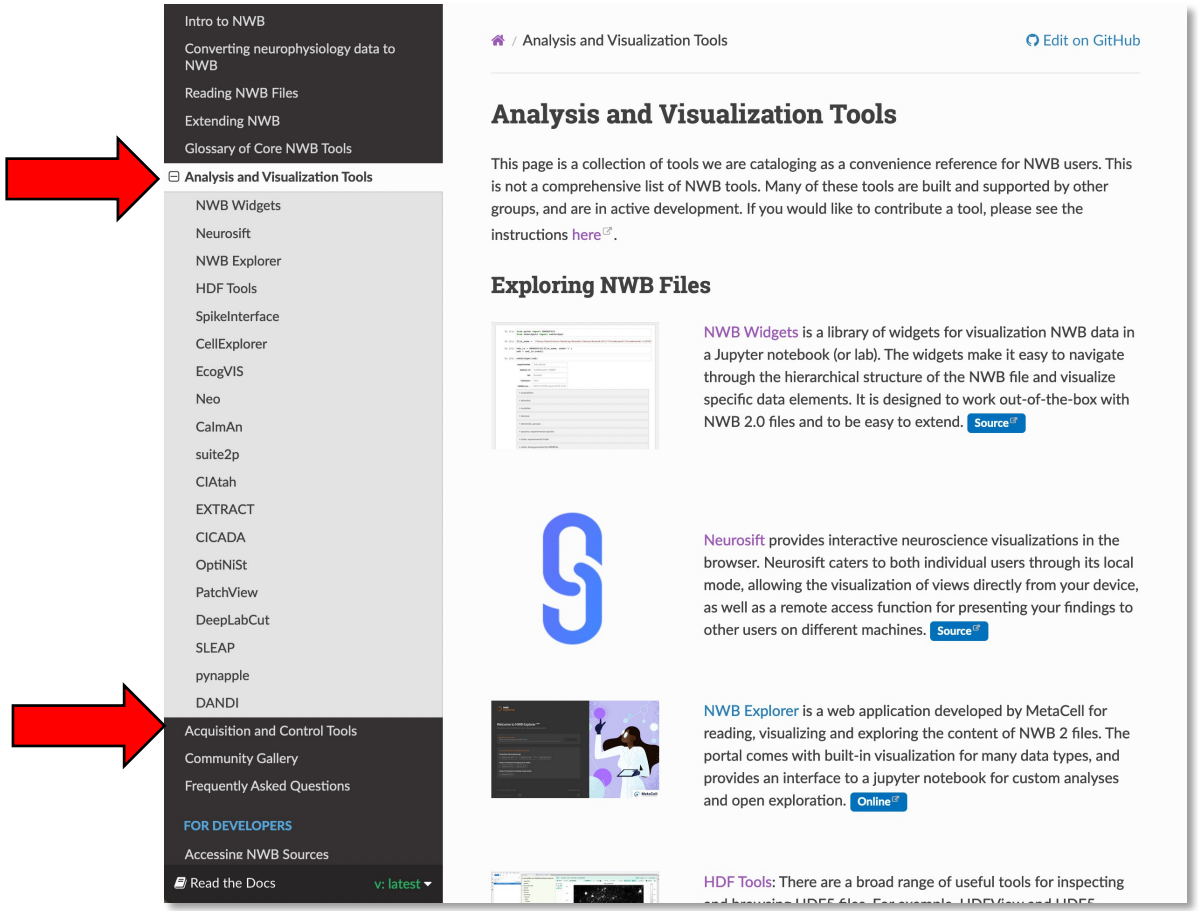


A unified data standard and software ecosystem for neurophysiology

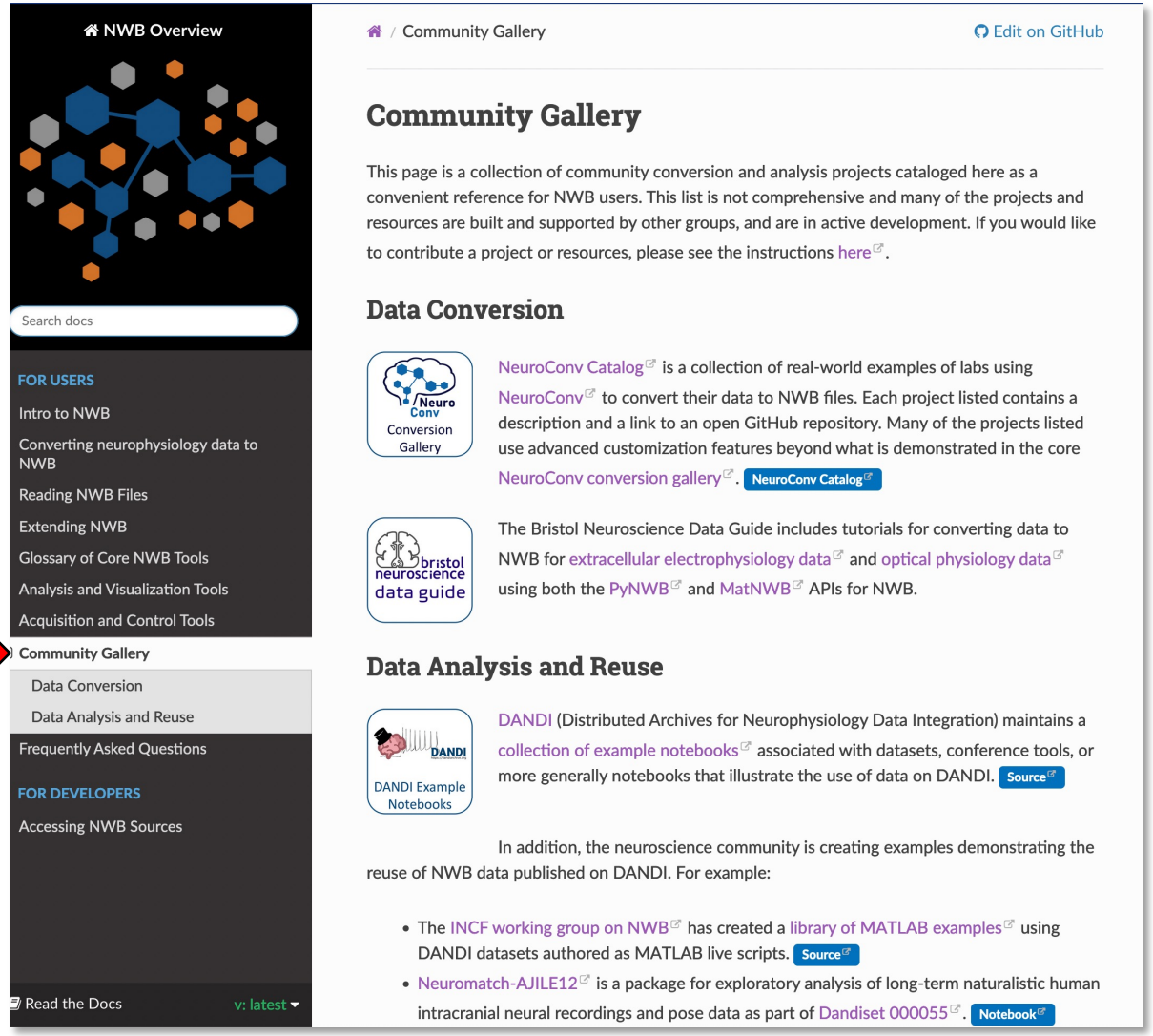


Tools Overview: https://nwb-overview.readthedocs.io

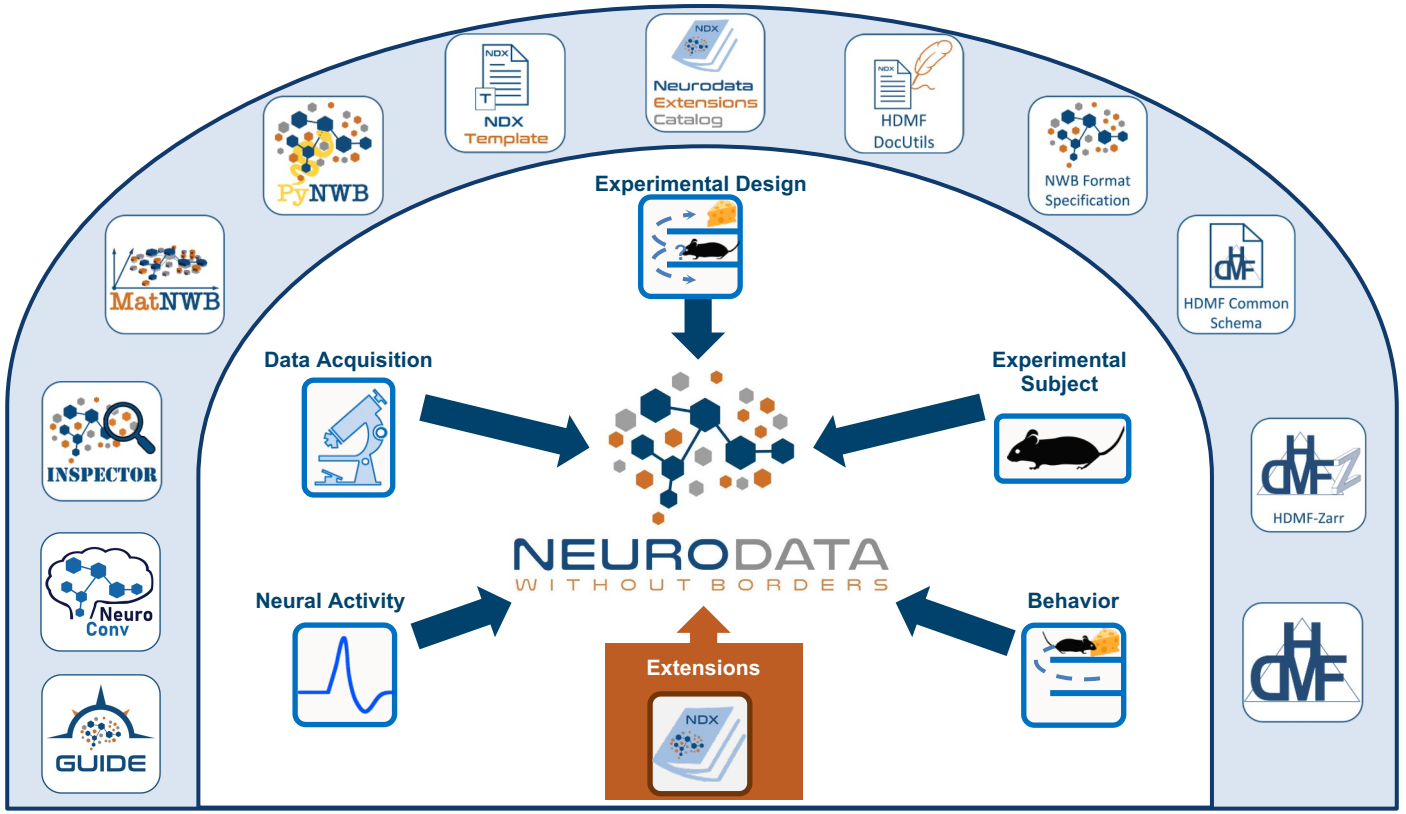




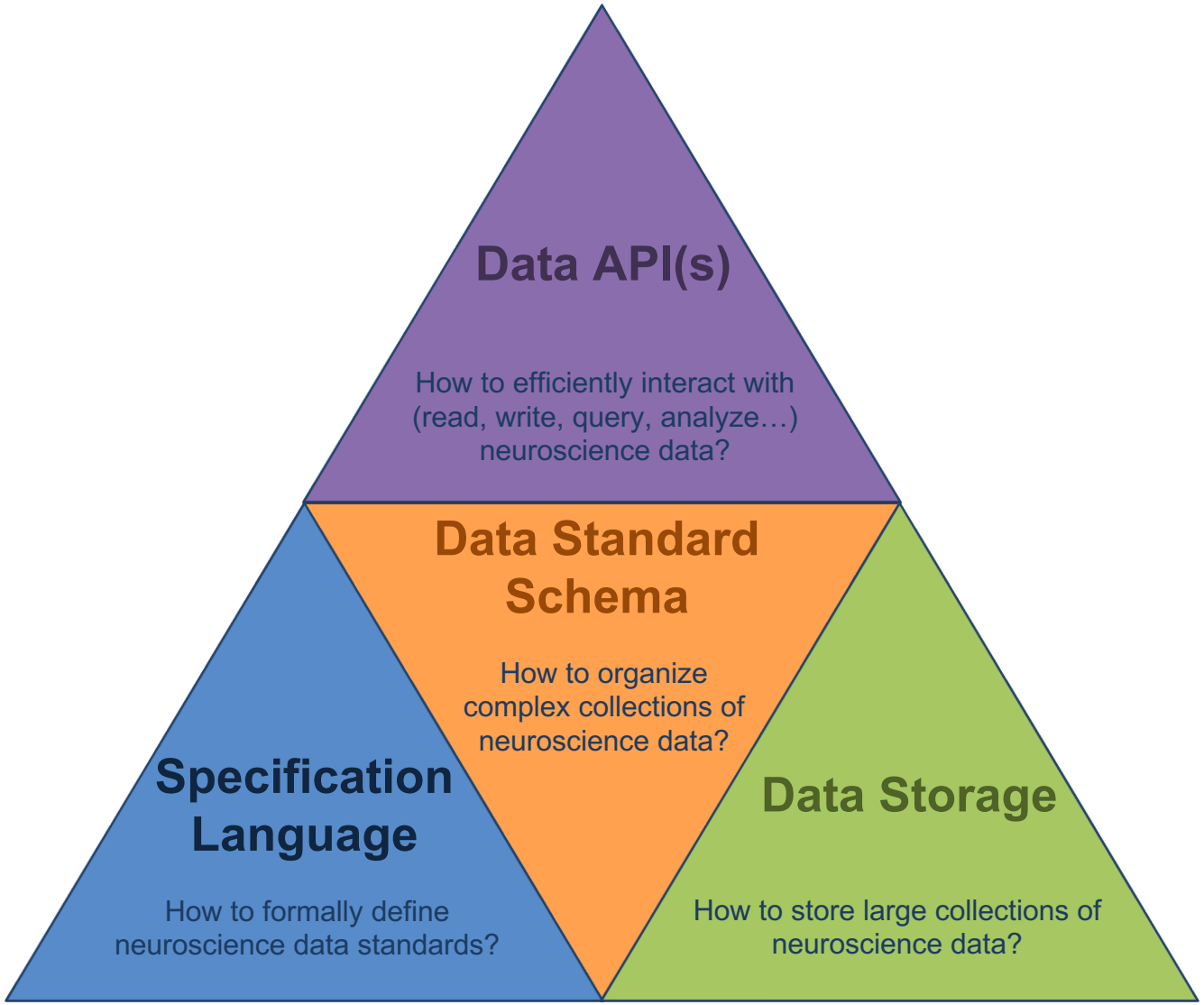
Community Gallery: https://nwb-overview.readthedocs.io



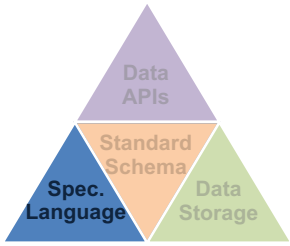
Core NWB Data Standard Ecosystem



Main components of the NWB ecosystem



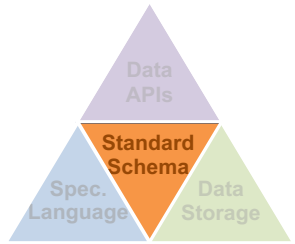
Specification Language



- Defines data objects (Group, Dataset, Link, Attribute) and their features (name, data type, dimensions, description, etc.) used to model data
- Defines common data elements as **neurodata_types** (object-oriented)
  - Supports inheritance and composition of types
- Extensions to NWB are defined using same language and formal rule

See: https://schema-language.readthedocs.io

NWB Standard Schema



- Uses the NWB Specification Language to define a large collection of reusable **neurodata_types** for storing a large set of neurophysiology data types and to define their organization as part of the NWBFile hierarchy

NEURODATA WITHOUT BORDERS

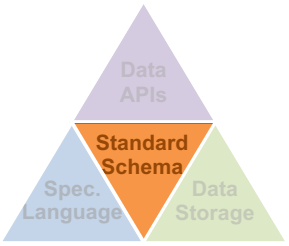
L general
  L subject
    `neurodata_type: Subject`
    L species : Mouse
  L experimenter :
    - Oliver Rübel
    - Ryan Ly
  L extracellular_ephys
    L electrodes :

x	y	z	imp	location	…
0.1	0.5	0.1	..	SSp-II1	…
1.1	0.5	2.1	..	SSp-II1	…
0.5	1.0	1.1	..	SSp-II2/3	…

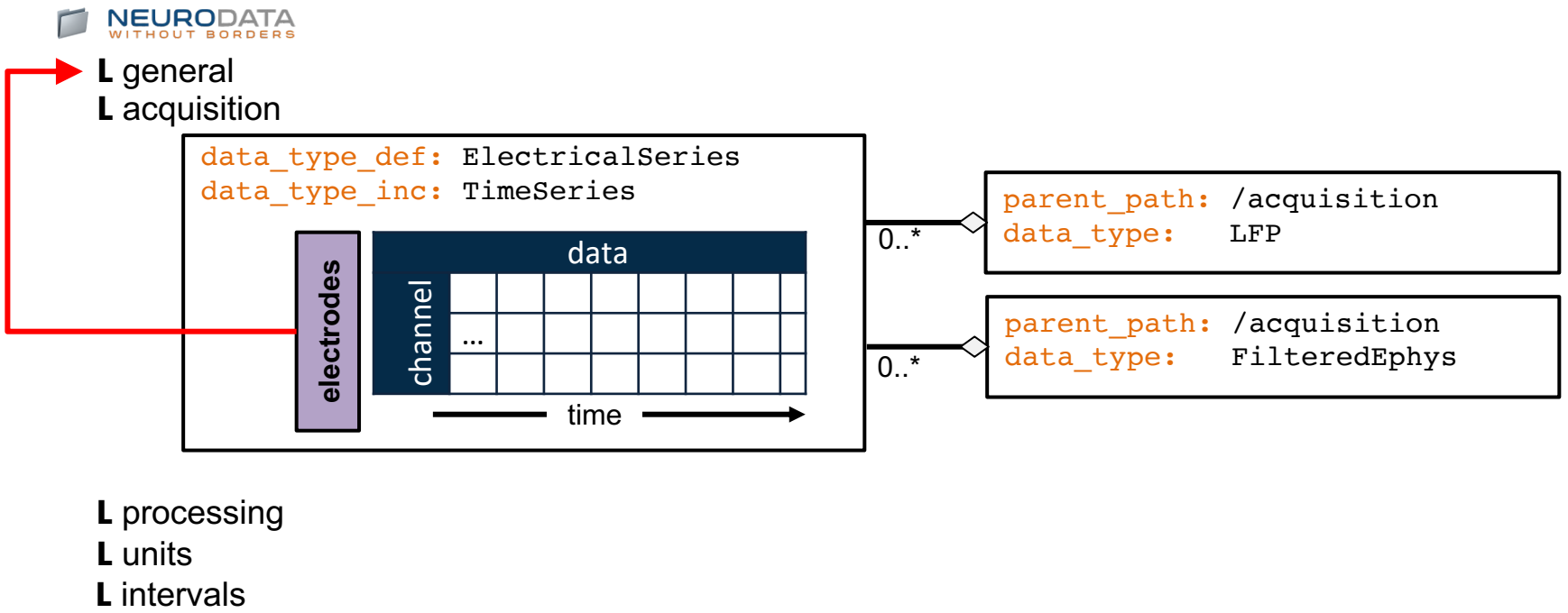
L acquisition
L processing
L units
L intervals
…

See: https://nwb-schema.readthedocs.io

NWB Standard Schema

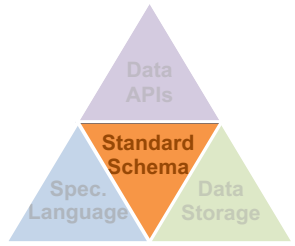


- Uses the NWB Specification Language to define a large collection of reusable **neurodata_types** for storing a large set of neurophysiology data types and to define their organization as part of the NWBFile hierarchy

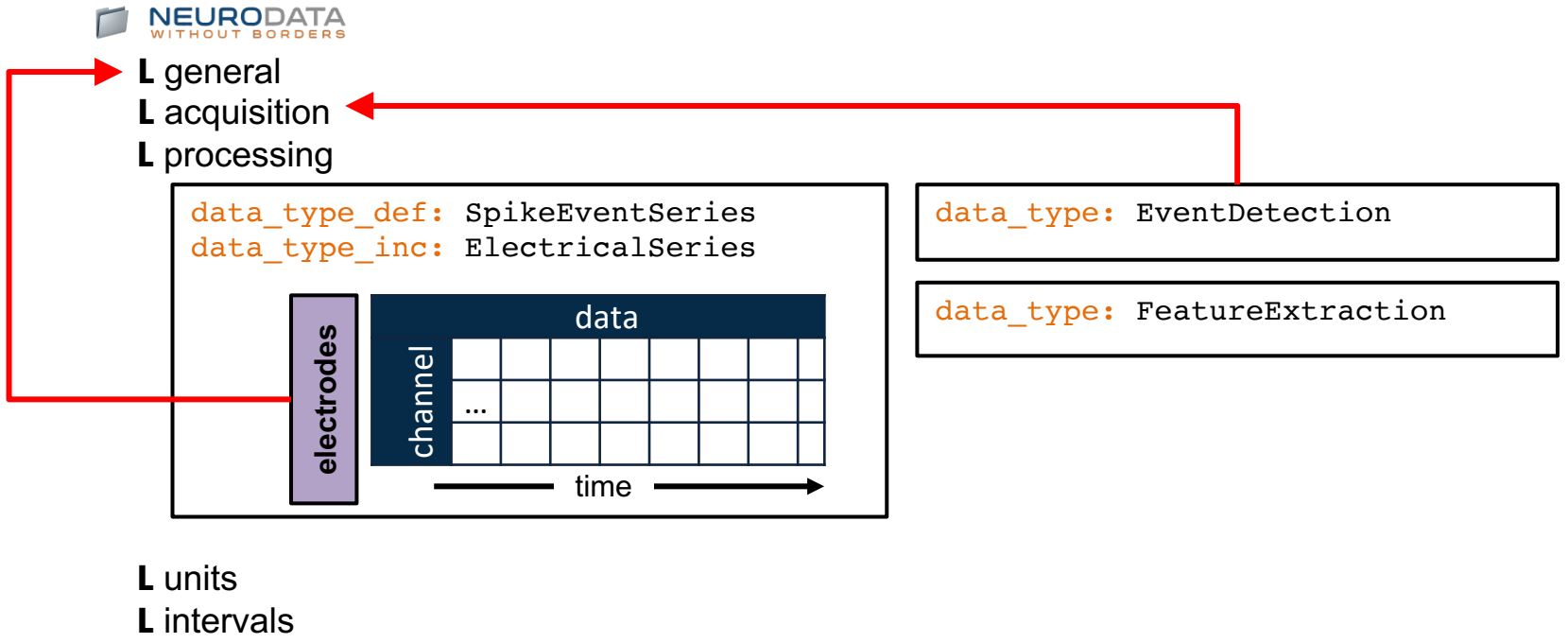


See: https://nwb-schema.readthedocs.io

NWB Standard Schema

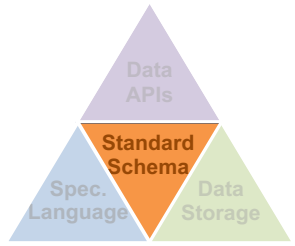


- Uses the NWB Specification Language to define a large collection of reusable **neurodata_types** for storing a large set of neurophysiology data types and to define their organization as part of the NWBFile hierarchy



See: https://nwb-schema.readthedocs.io

NWB Standard Schema



- Uses the NWB Specification Language to define a large collection of reusable **neurodata_types** for storing a large set of neurophysiology data types and to define their organization as part of the NWBFile hierarchy

NEURODATA WITHOUT BORDERS

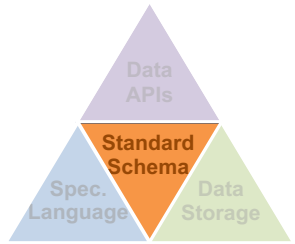
L general
L acquisition
L processing
L units

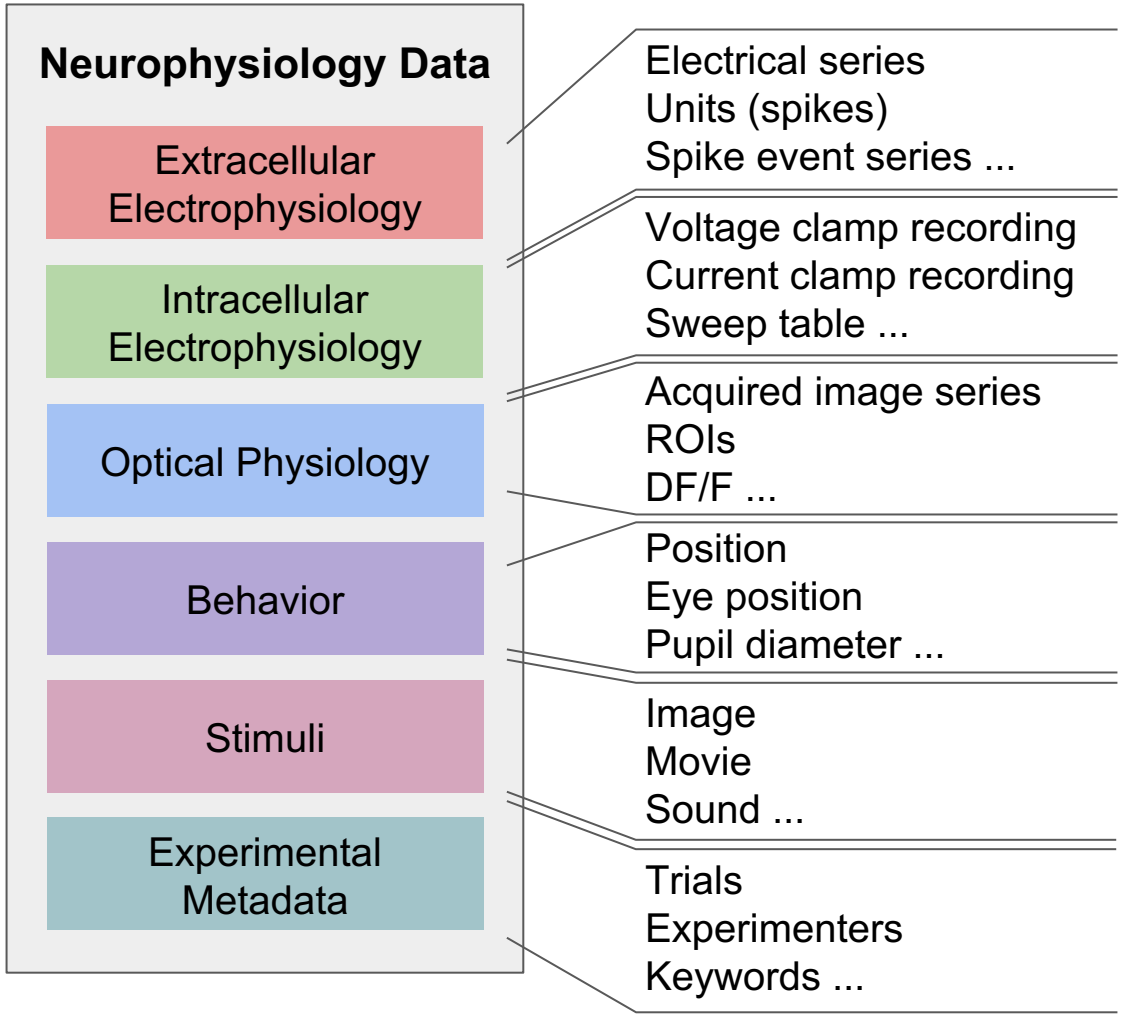
waveform_mean	waveform_sd	electrodes	spike_times	obs_intrevals	…
[0.1, 0.3, …]	[0.04, 0.06, …]	[0, 1, 2]	[0.03, 0.14]	[[0.03, 0.05], [1.01, 1.06]]	…
[1,2, 1.3,	[0.06, 0.08, …]	[3, 5, 6, 8]	[0.6, 0.71, 0.82]	[[0.07, 0.09], [1.52, 1.57], [2.63, 2.68]]	…
…	…	…	…	…	…

L intervals

See: https://nwb-schema.readthedocs.io

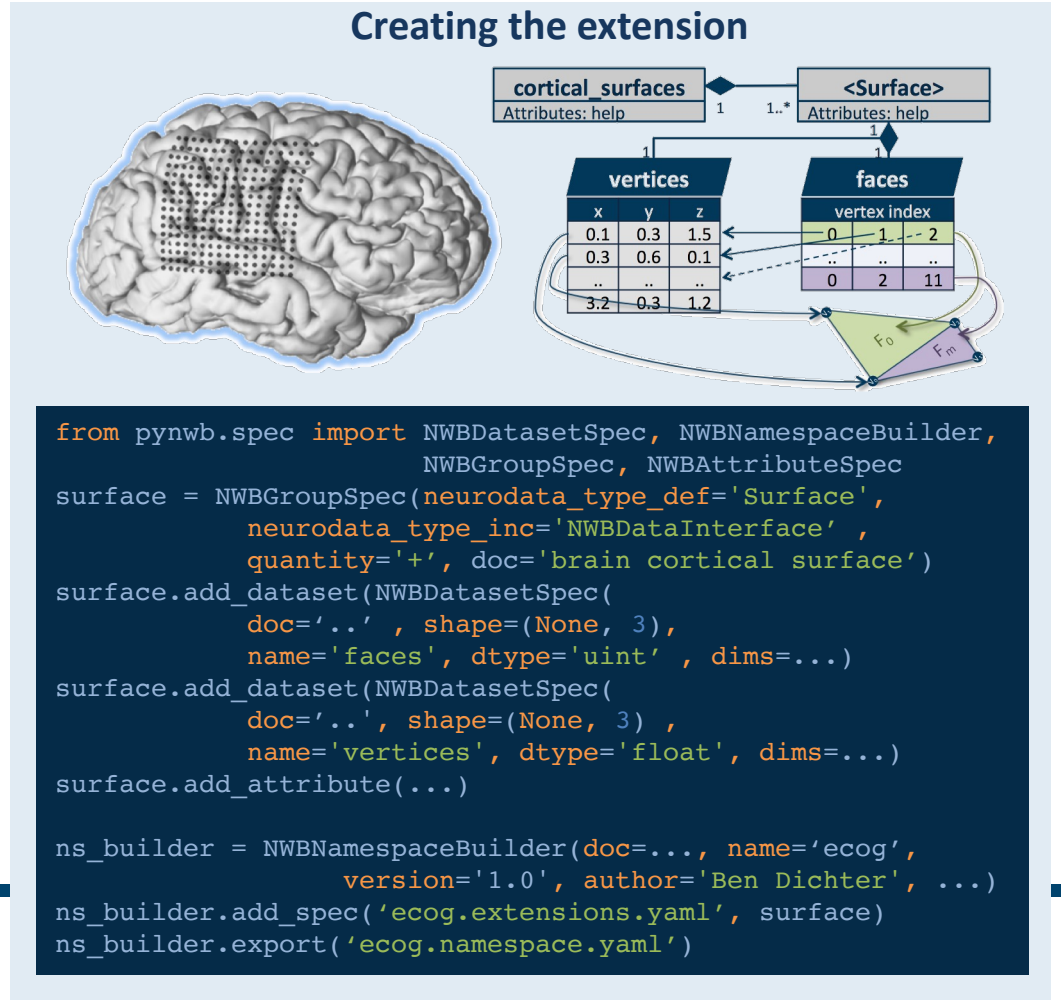
NWB provides structured **neurodata_types** for most of neurophysiology data

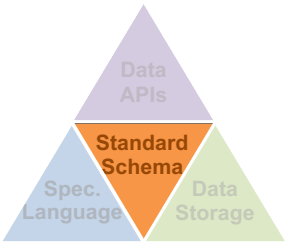




NWB Extensions (NDX)

- Using the NWB Specification Language, NWB supports the formal extension of the data standard to define new data types and metadata

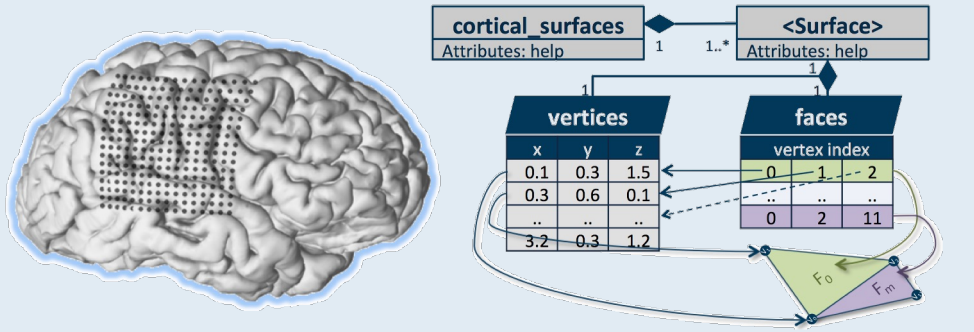




NWB Extensions (NDX)

- Using the NWB Specification Language, NWB supports the formal extension of the data standard to define new data types and metadata

Creating the extension



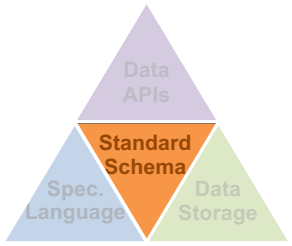

```
from pynwb.spec import NWBDatasetSpec, NWBNamespaceBuilder,
                       NWBGroupSpec, NWBAttributeSpec
surface = NWBGroupSpec(neurodata_type_def='Surface',
            neurodata_type_inc='NWBDataInterface' ,
            quantity='+', doc='brain cortical surface')
surface.add_dataset(NWBDatasetSpec(
            doc='..' , shape=(None, 3),
            name='faces', dtype='uint' , dims=...)
surface.add_dataset(NWBDatasetSpec(
            doc='..', shape=(None, 3) ,
            name='vertices', dtype='float', dims=...)
surface.add_attribute(...)

ns_builder = NWBNamespaceBuilder(doc=..., name='ecog',
                  version='1.0', author='Ben Dichter', ...)
ns_builder.add_spec('ecog.extensions.yaml', surface)
ns_builder.export('ecog.namespace.yaml')
```

Using the extension

PyNWB: write

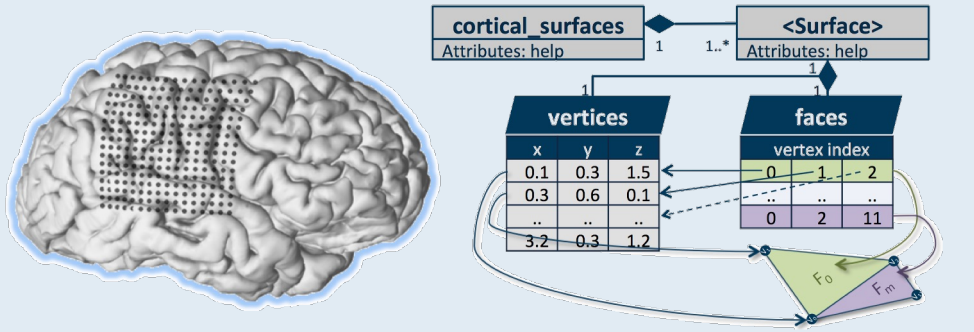
```
from pynwb import load_namespaces, get_class,
                  NWBHDF5IO, NWBFile ...
nwbfile = NWBFile(...)
load_namespaces('ecog.namespace.yaml')
Surface = get_class('Surface'," 'ecog')
surf = Surface(faces=... , vertices=...,
               name='Surface_1'...)
nwbfile.add_acquisition(surf)
with NWBHDF5IO('surface_example.nwb', 'w') as io:
      io.write(nwbfile)
```



NWB Extensions (NDX)

- Using the NWB Specification Language, NWB supports the formal extension of the data standard to define new data types and metadata

Creating the extension




```
from pynwb.spec import NWBDatasetSpec, NWBNamespaceBuilder,
                       NWBGroupSpec, NWBAttributeSpec
surface = NWBGroupSpec(neurodata_type_def='Surface',
            neurodata_type_inc='NWBDataInterface' ,
            quantity='+', doc='brain cortical surface')
surface.add_dataset(NWBDatasetSpec(
            doc='..' , shape=(None, 3),
            name='faces', dtype='uint' , dims=...)
surface.add_dataset(NWBDatasetSpec(
            doc='..', shape=(None, 3) ,
            name='vertices', dtype='float', dims=...)
surface.add_attribute(...)

ns_builder = NWBNamespaceBuilder(doc=..., name='ecog',
                  version='1.0', author='Ben Dichter', ...)
ns_builder.add_spec('ecog.extensions.yaml', surface)
ns_builder.export('ecog.namespace.yaml')
```

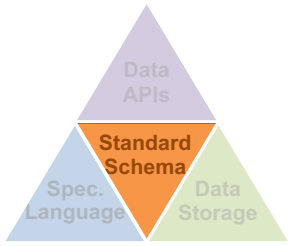
Using the extension

PyNWB: write

```
from pynwb import load_namespaces, get_class,
                    NWBHDF5IO, NWBFile ...
nwbfile = NWBFile(...)
load_namespaces('ecog.namespace.yaml')
Surface = get_class('Surface'," 'ecog')
surf = Surface(faces=... , vertices=...,
                name='Surface_1'...)
nwbfile.add_acquisition(surf)
with NWBHDF5IO('surface_example.nwb', 'w') as io:
      io.write(nwbfile)
```

PyNWB: read

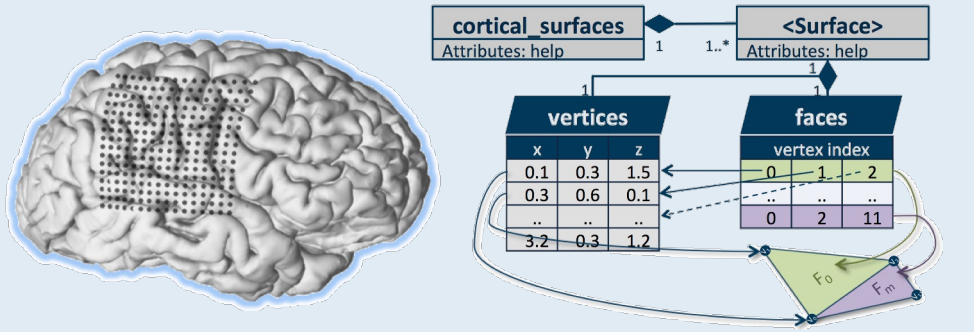
```
load_namespaces('ecog.namespace.yaml' )
io = NWBHDF5IO('surface_example.nwb', 'r')
nwbfile = io.read()
nwbfile.get_acquisition('Surface 1').vertices
```



NWB Extensions (NDX)

- Using the NWB Specification Language, NWB supports the formal extension of the data standard to define new data types and metadata

Creating the extension




```
from pynwb.spec import NWBDatasetSpec, NWBNamespaceBuilder,
                       NWBGroupSpec, NWBAttributeSpec
surface = NWBGroupSpec(neurodata_type_def='Surface',
            neurodata_type_inc='NWBDataInterface' ,
            quantity='+', doc='brain cortical surface')
surface.add_dataset(NWBDatasetSpec(
            doc='..' , shape=(None, 3),
            name='faces', dtype='uint' , dims=...)
surface.add_dataset(NWBDatasetSpec(
            doc='..', shape=(None, 3) ,
            name='vertices', dtype='float', dims=...)
surface.add_attribute(...)

ns_builder = NWBNamespaceBuilder(doc=..., name='ecog',
                  version='1.0', author='Ben Dichter', ...)
ns_builder.add_spec('ecog.extensions.yaml', surface)
ns_builder.export('ecog.namespace.yaml')
```

Using the extension

PyNWB: write

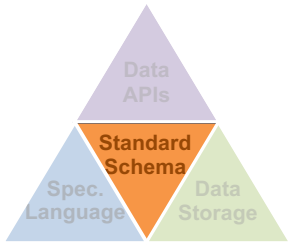
```
from pynwb import load_namespaces, get_class,
                    NWBHDF5IO, NWBFile ...
nwbfile = NWBFile(...)
load_namespaces('ecog.namespace.yaml')
Surface = get_class('Surface'," 'ecog')
surf = Surface(faces=... , vertices=...,
                name='Surface_1'...)
nwbfile.add_acquisition(surf)
with NWBHDF5IO('surface_example.nwb', 'w') as io:
      io.write(nwbfile)
```

PyNWB: read

```
load_namespaces('ecog.namespace.yaml' )
io = NWBHDF5IO('surface_example.nwb', 'r')
nwbfile = io.read()
nwbfile.get_acquisition('Surface 1').vertices
```

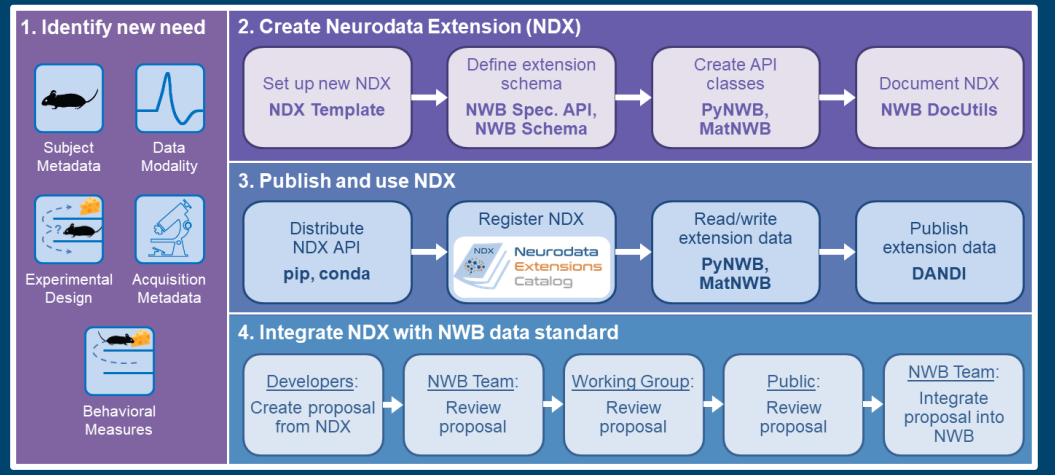
MatNWB: write

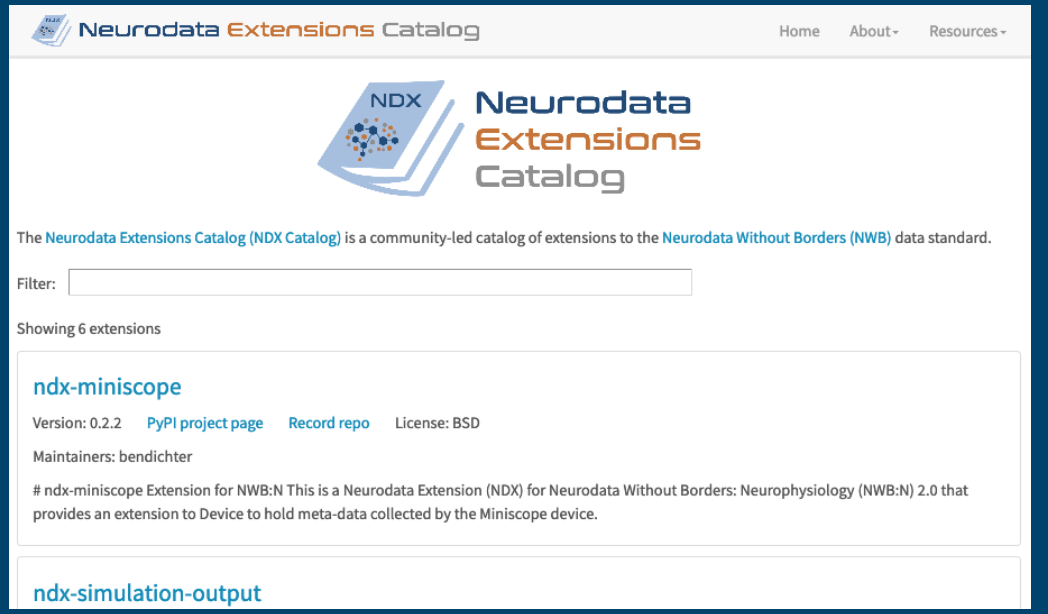
```
generateExtension('ecog.namespace.yaml');
file = nwbfile(...)
surf = types.ecog.Surface(...
                    'faces', faces, 'vertices',
vertices);
file.acquisition.set('Surfaces_1', surf);
nwbExport(file, 'ecephys_tutorial.nwb')
```



Neurodata Extension Catalog

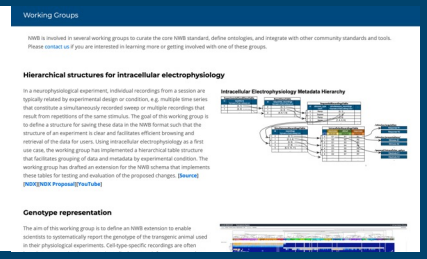
- Provide easy-to-use tools and processes for creating, using, sharing, evaluation, reviewing, and extensions to the NWB data standard:
  - **Extension API:** Using the NWB Specification Language and API, NWB supports the formal extension of the data standard to define new data types and metadata
  - **NDX Catalog:** Online catalog for sharing and reviewing extensions:
  - **ndx_template:** Cookiecutter-based template for creating new extension code repositories, schema, documentation, and interfaces
  - **staged_extensions:** Submit extensions to the NDX catalog
  - **hdfm_docutils:** Tools to auto-generate documentation from extension schema
  - **Guidelines and rules:** for versioning, sharing, proposing, and reviewing extension
- Using the concept of Neurodata Extensions (NDX) enables:
  - users to adopt NWB and share their data even if they need to store additional metadata or new data types
  - Curation of the NWB data standard via community proposals
  - Complete testing and evaluation of proposals prior to integration with the NWB core standard and software API



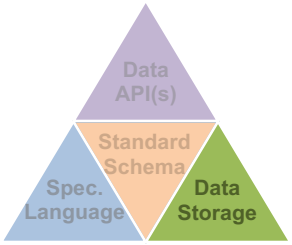


For a list of current NWB working groups visit:

www.nwb.org/working-groups



NWB Data Storage

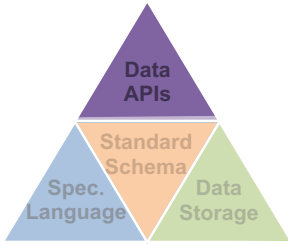


The data storage maps NWB Specification Language primitives (Groups, Datasets, Attributes etc.) to storage

- NWB uses HDF5 as its main file storage backend:
  - Supports large-scale storage of complex data collections in a single file
  - Optimized for performance (parallel I/O, advanced I/O filters etc.)
  - Supported across platforms and languages (Matlab, Python, C/C++, Fortran, R…)
  - Targets long-term support
  - Supports advanced I/O features (e.g., lazy load, chunking, compression, links and references etc.)

- Working with the community to explore and support integration with other data management and storage technologies:
  - HDMF-Zarr
  - DataJoint

Data APIs



PyNWB (Python) and MatNWB (Matlab) reference APIs:

- Efficiently read, write, query, and analyze NWB files
- APIs are interoperable
- APIs support advanced I/O features, e.g., lazy data load, chunking, compression, iterative data write, streaming etc.

GUIDE

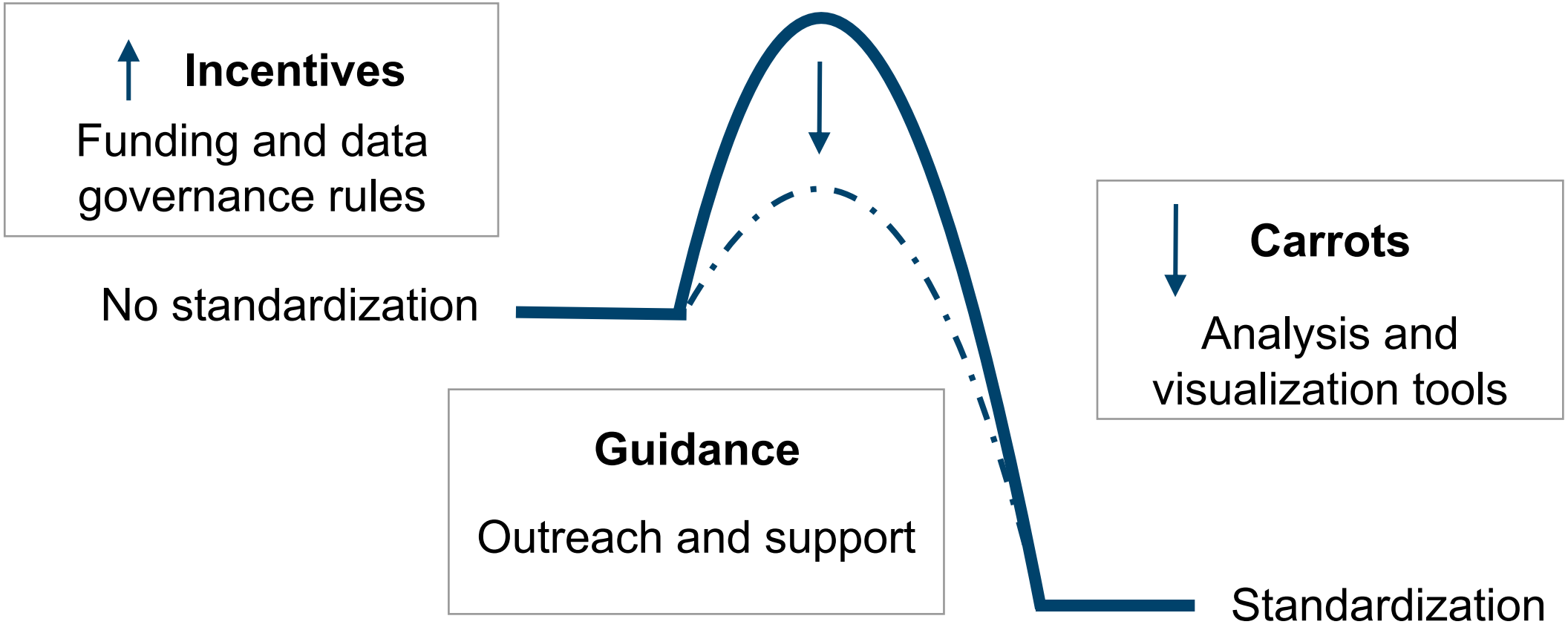
Neuro Conv

Convert data to NWB with support for many common acquisition and processing data formats

INSPECTOR

Validate and inspect NWB files

Overcoming the energy barrier in data standardization to advance FAIR neurophysiology data through dissemination of NWB



Enable neuroscience researchers to effectively utilize the NWB data standard and software

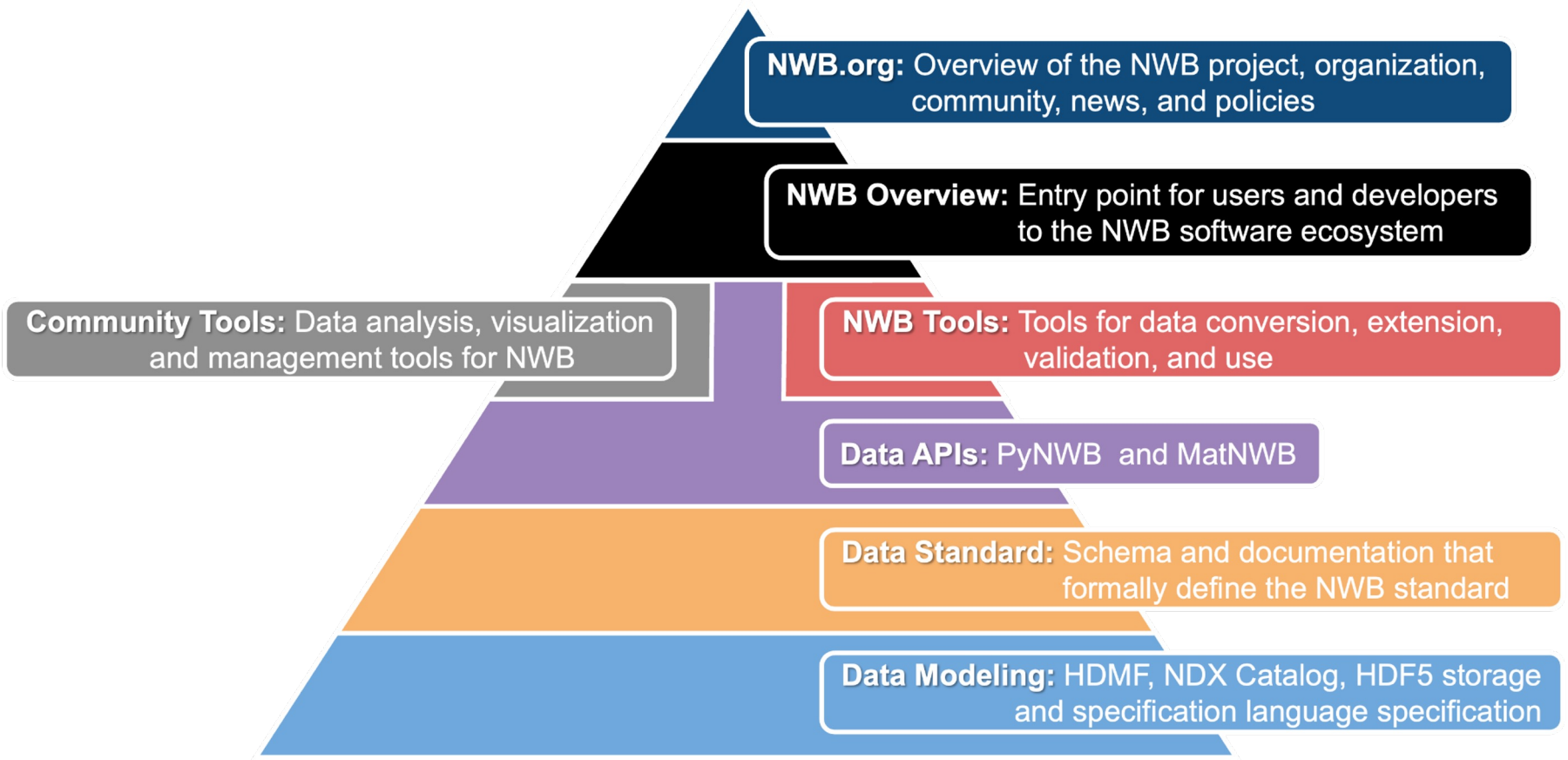
Goal: Lower the barriers of adopting NWB and promote broad dissemination of NWB by focusing on the needs of neuroscience researchers and labs by:

- Online and in-person training

  - **July 24 – 26:** NWB User Days @ HHMI Janelia (Ashburn, VA, USA)
  - **July 27 – 29:** NWB Developer Days @ HHMI Janelia (Ashburn, VA, USA)
  - **Sept. 5 – 8:** NeuroDataReHack @ IRBO World Conference (Granada, Spain)

- Consulting and support for labs to integrate with NWB
- Create community-driven processes for integration of new neuroscience technologies with NWB
  - Founded NWB Technical Advisory Board (TAB)
  - Growing collection of community extensions, e.g., ndx-pose, ndx-beadl, ndx-events, ndx-geneotype, etc.

Navigating the online documentation



How to get help?

Slack, Mailing List, and Twitter

NWB Help Desk GitHub

Issues and Pull Requests GitHub

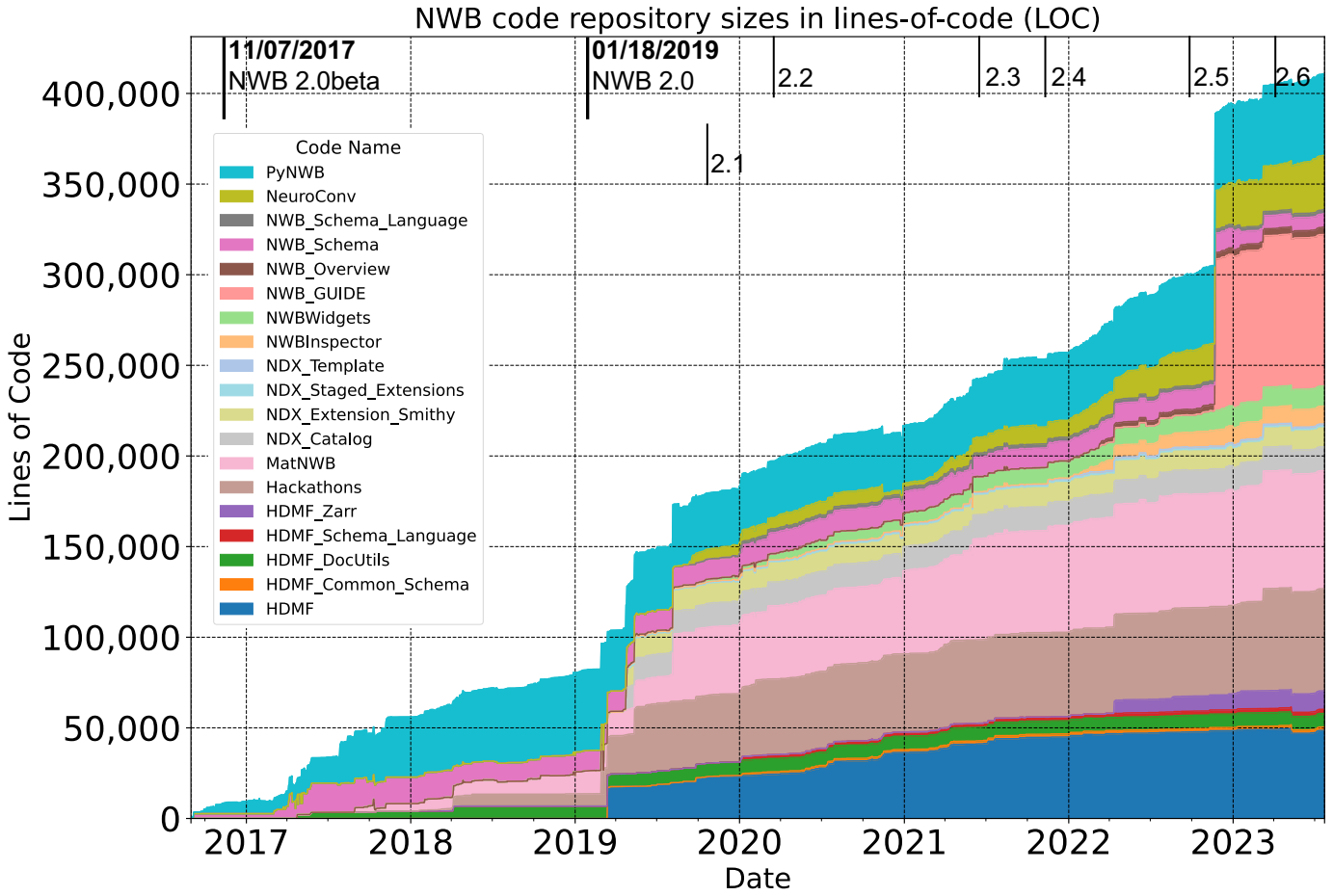
Enable integration of NWB with neurophysiology data analysis and management tools and technologies

Goal: Promote and support integration of NWB with neuroscience tools creating a vibrant software ecosystem around NWB and creating incentive for researchers to adopt NWB through:

- reductions in cost, time, and effort;
- improved quality, reliability, and accuracy of results
- access to new scientific capabilities

Approach:

- Adapt NWB software to support and integrate with community tools
- Consulting and support for developers
- Dedicated standard and software projects
  - BEADL (behavioral task description language)
  - AqNWB (C++ API for data acquisition)
  - NWB GUIDE user interface for data conversion



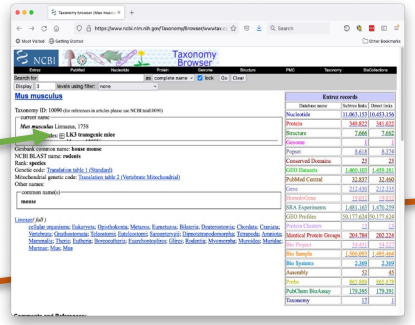
NWB + External Resources

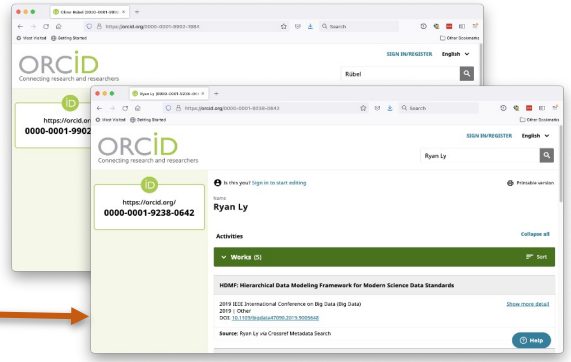
NEURODATA WITHOUT BORDERS

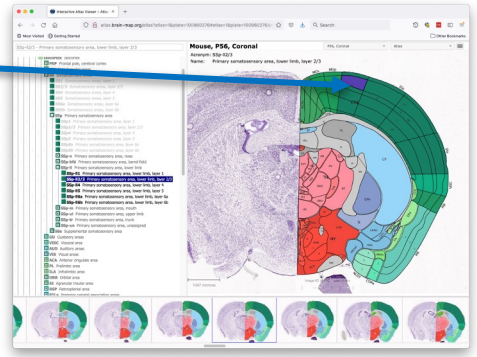
- general
  - subject
    - `neurodata_type: Subject`
    - species : Mouse
  - experimenter :
    - Oliver Rübel
    - Ryan Ly
  - extracellular_ephys
    - electrodes :

x	y	z	imp	location	…
..	..	..	..	SSp-II1	…
..	..	..	..	SSp-II1	…
..	..	..	..	SSp-II2/3	…

- acquisition
  - neuropixel_recordings
    - `neurodata_type: ElectricalSeries`
- ...







objects_idx	object_id	relative_path	field	keys_idx	key	entities_idx	entity_id	entity_uri
0	ea7c0fd3-48ab-4a88-9188-05f7789f7661	Subject/species		0	Mus musculus	0	NCBI:txid10090	https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/...
0	ea7c0fd3-48ab-4a88-9188-05f7789f7661	Subject/species		1	Homo sapiens	1	NCBI:txid9606	https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/...
1	b8c014a8-cc5e-400f-b337-d5d6dfa3746a	root/general/experimenter		2	Ruebel, Oliver	2	0000-0001-5725-6910	https://orcid.org/0000-0001-5725-6910
,..	…	…						

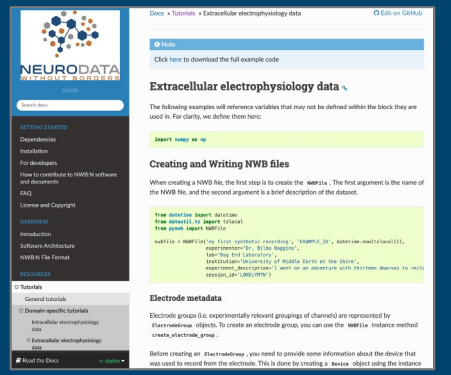
objects **keys** **entities**

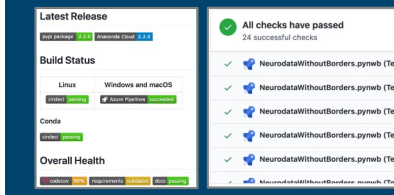
Develop and maintain an accessible and sustainable open source software ecosystem for NWB

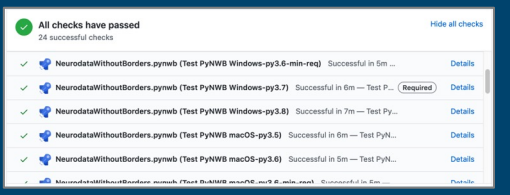
Goal: Ensure readiness of NWB for production use, enhance adoption, and enable the community to contribute to NWB

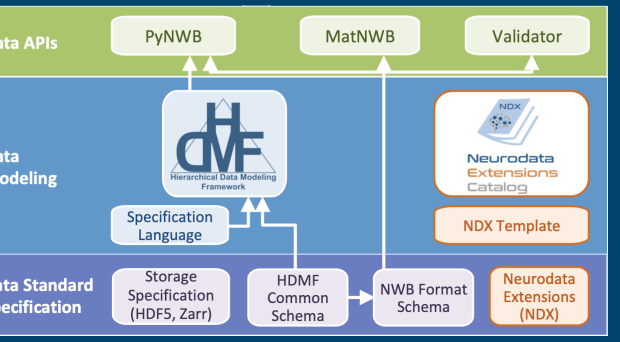
Software Strategy:

- **Accessibility:** Open source, easy-to-install (pip, conda etc.), detailed documentation, INCF endorsed, R&D100 award
- **Reliability:** Continuous integration, testing, pull requests, and code reviews
- **Stability:** Software management, insulate components, rules and guidelines, regular releases
- **Support:** nwb.org, GitHub, Slack (600+ members), mailing list (>1000 members), Twitter, YouTube, INCF Training Space, ReadTheDocs, newsletter etc.
- **Functionality:** Validator, extensions, support advanced data types, advanced I/O, integration with data libraries (Pandas, numpy etc.)



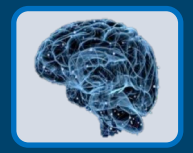


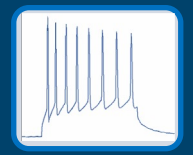


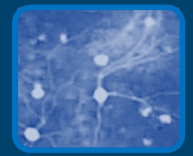


Multidisciplinary team science at work

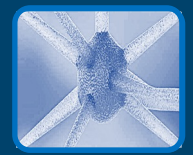
Applications


Extracellular electrophysiology


Intracellular electrophysiology


Optical physiology

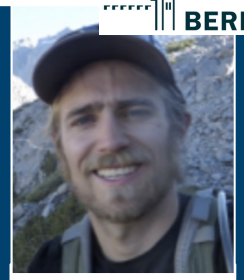

Behavior


Simulations

Technology Teams

BERKELEY LAB

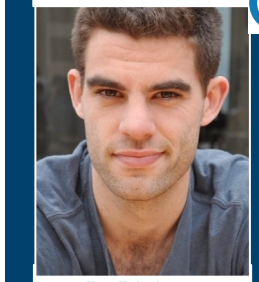

O. Rübel

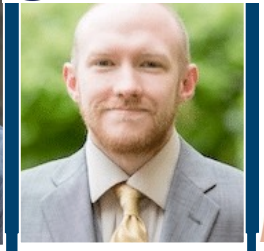

A. J. Tritt


R. Ly


M. Avaylon

CATALYST NEURO


B. Dichter


C. Baker


A. Weigl

mbf BIOSCIENCE


L. Niu

ALLEN INSTITUTE for BRAIN SCIENCE


S. De Vries


A. Buccino


L. Ng

DARTMOUTH


Y.O. Halchenko

NWB Executive Board


K. Bouchard (LBNL)


B.W. Brunton (UW)


E. Buffalo (UW)


A. Churchland (UCLA)

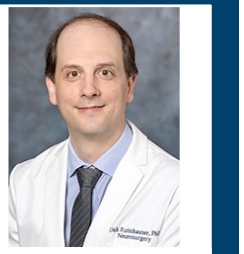

L. M. Frank (UCSF)


S. Ghosh (MIT)


A. Kepecs (WUSTL)


M. Murthy (Princeton)


H.D. Mansvelder (VU Amsteram)


U. Rutishauser (Cedars Sinai)

Alumni: L. Ng, C. Koch, F. Sommer, K. Svoboda, M. Meister, K. Amunds

Sponsors

NIH The BRAIN Initiative®

THE KAVLI FOUNDATION

SIMONS FOUNDATION

Industry Engagement

- CatalystNeuro
- DataJoint
- Vidrio
- MathWorks
- Kitware

Broader User and Developer Community

Apologies if your name/team is missing!
This slide only shows a very rough cut of some of the teams and people that work on developing NWB. Pictures and names of many, many important members of the NWB community are missing!

NEURODATA
WITHOUT BORDERS

Visit us at NWB.org and nwb-overview.readthedocs.io

2019 R&D 100 WINNER

incf ENDORSED

BERKELEY LAB
Bringing Science Solutions to the World

Legal

BERKELEY LAB
Bringing Science Solutions to the World

- **Disclaimer:** This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
-