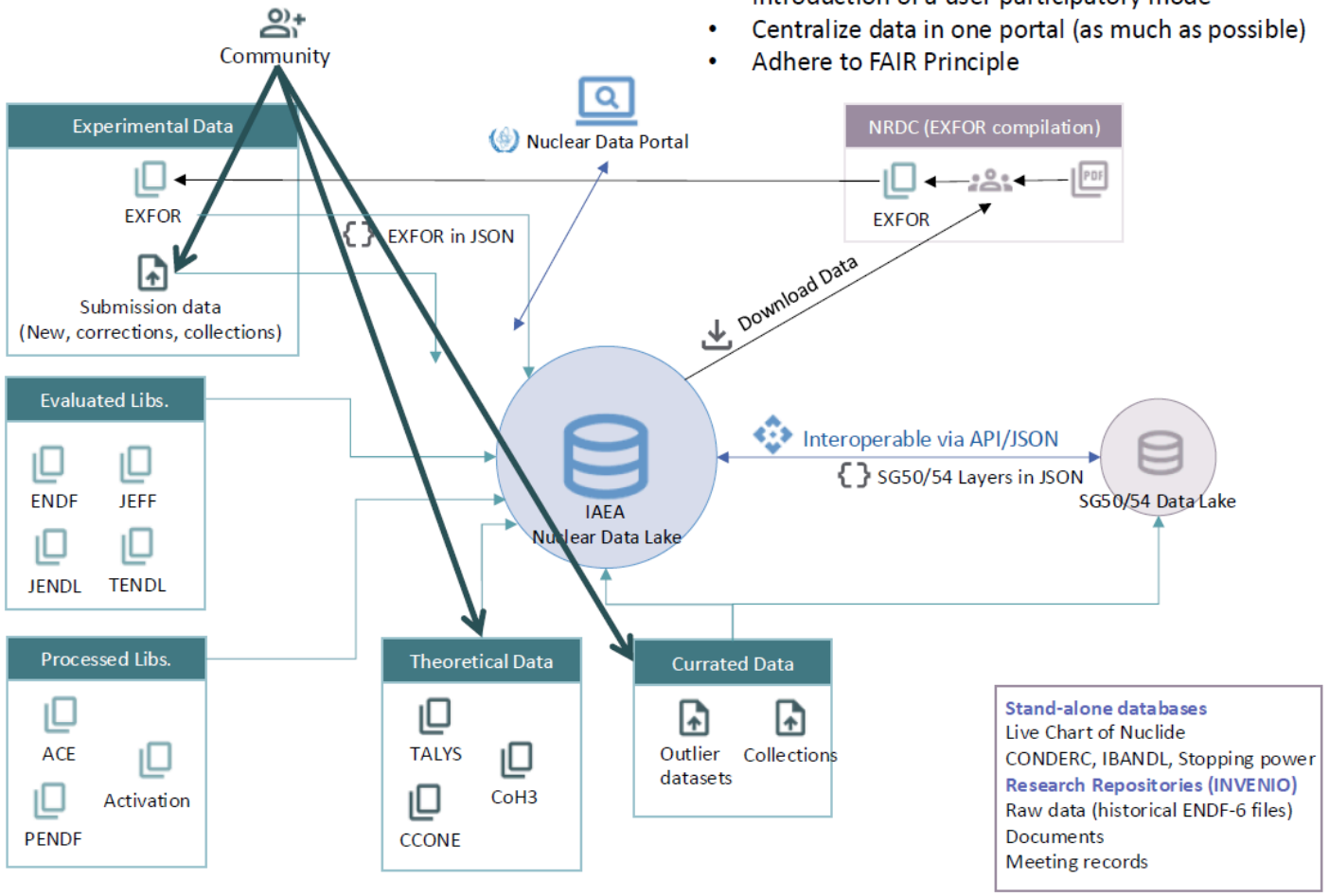
NEA
NUCLEAR ENERGY AGENCY

Ideas and concept for nuclear data management

A brainstorming session on Data Portal building blocks

A small reminder

- Introduction of a user participatory mode
- Centralize data in one portal (as much as possible)
- Adhere to FAIR Principle



User needs from NEA experience

- Graphical (Web)interface for exploration
- API access for fast access to underlying data (growing importance)

Community Contribution and Interaction

- Technical Skills & Motivation vary greatly within and between communities

Data Portal Requirements

- Rely on open tools and standards
- Shared maintenance
- Decentralized system architecture
- Data and code change tracking including contributor identity
- Data processing provenance tracking
- Simple licensing framework
- API & no-code data access
- Restricted, validated data structure

(Exchange) Format requirements

- requires validation
- needs to capture hierarchical **and** tabular data

(Exchange) Format options

***Formats*:**	**Pro**	**Con**
json/yaml	readability	no table support
xml	some table support	verbosity
hdf5	versatile	requires framework
internet object	best of json + xml	relatively unknown
database	dedicated & versatile	requires system DB knowledge for maintenance & contribution

Neuroscience inspired: mixed formats

- Open-Ephys Binary format
  - binary array data for continuous data streams
  - numpy for non continuous data streams
  - json for metadata

Design focus

- designed for robust acquisition of continuous large data streams

Neuroscience inspired: hdf5

NWB:N: Neurophysiology Data Standard

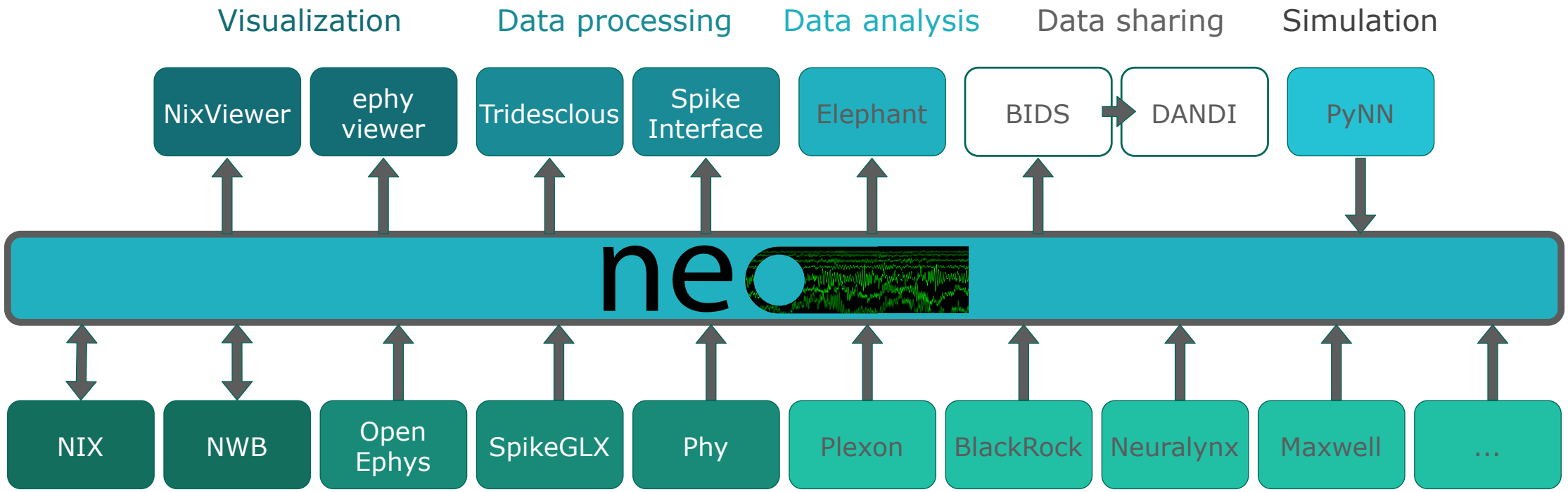
- for archival of recording data
- multimodal
- restricted modality descriptions

Design focus

- customly extensible (with some experience)
- archival, not working data format
- restrictive and compact (autocompression)

NEO

The universal data interface for microelectrode electrophysiology data (Python)



Infrastructures

Requirements

- decentralized framework
- version control capabilities
- shared maintenance and governance

Git (-hub/-lab)

- state of the art for code / text file versioning
- integrates with multiple project managements platforms
- integrates with with diverse project automation features (CD/CI)
- not suited for large data files

GitLFS 

- main stream solution for combining git + large files
- has drawbacks on usability / backend choices

Git-annex

- advanced large file storage
- not anymore supported by Gitlab

GIN

- alternative platforms integrating git + git-annex
- currently not maintained
- can be self-hosted

Datalad

- DataLad is a free and open source distributed data management
- based on git-annex
- includes provenance tracking features

Provenance tracking

- essential feature if data processing is included in Data Lake
- tracks environment and dependencies used to generate file
- integrated feature in Datalad

Open Brainstorming

- Further key requirements?
- Additional tools to consider?
- Any more ideas on interfaces?