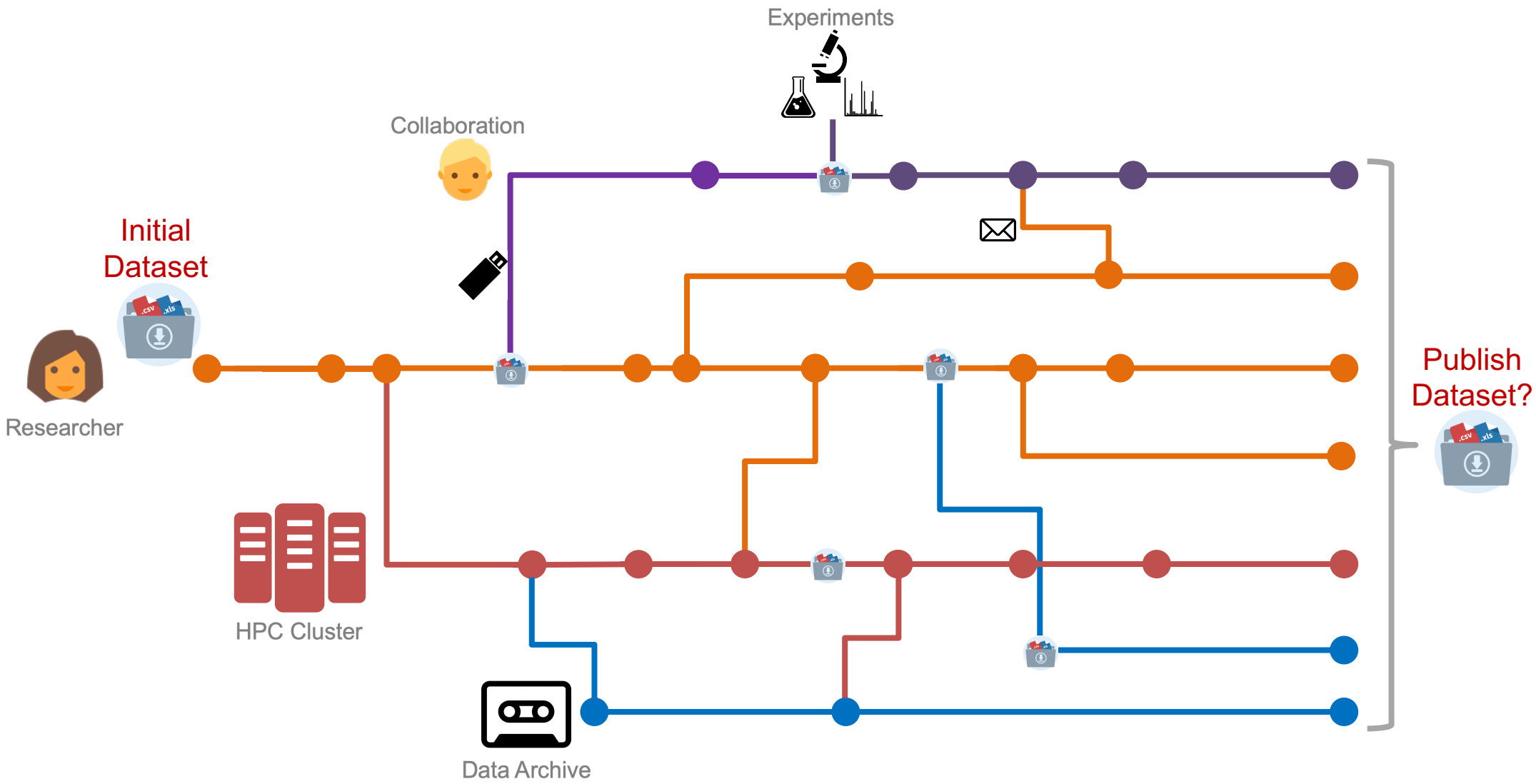
Integrating Data Management into compute workflows with iRODS

Agenda

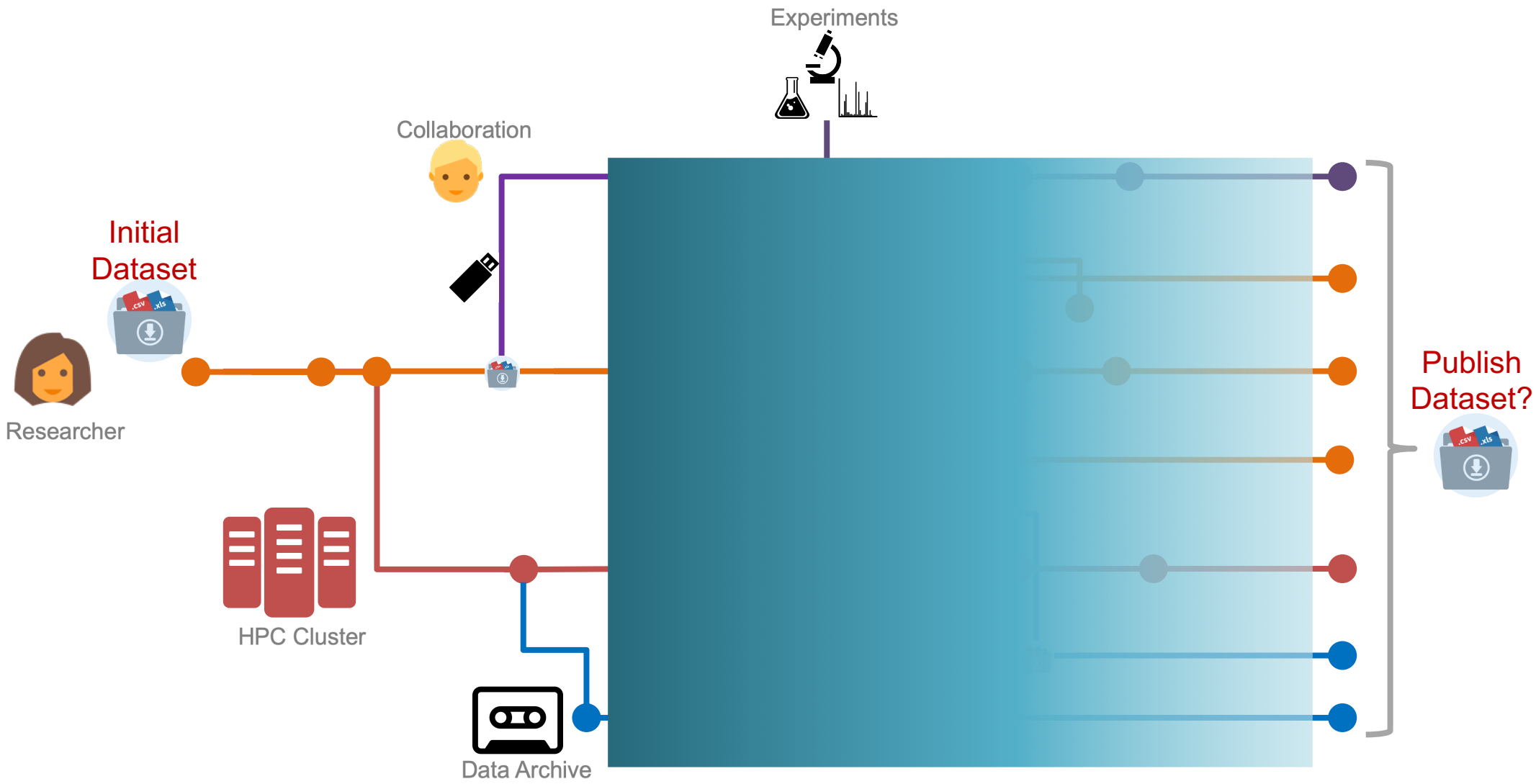
9:00 – 9:30	Welcome and Introduction
9:30 – 10:00	Data Management, FAIR and iRODS
10:00 – 11:00	**Hands-on**: Data handling with the python API
11:00 – 11:15	**Coffee break**
11:15 – 12:00	**Hands-on**: Data handling with the python API continued
12:00 – 13:30	**Lunch**
13:30 – 14:00	The HPC system, compute workflows and iRODS
14:00 – 15:00	**Hands-on**: Two compute workflows
15:00 – 15:15	**Coffee break**
15:15 – 16:15	**Hands-on**: Two compute workflows
16:15 – 16:30	**Wrap up and evaluation**

Data, what’s the problem?

Data, what's the problem?



Provenance typically completely lost



Data Life Cycle

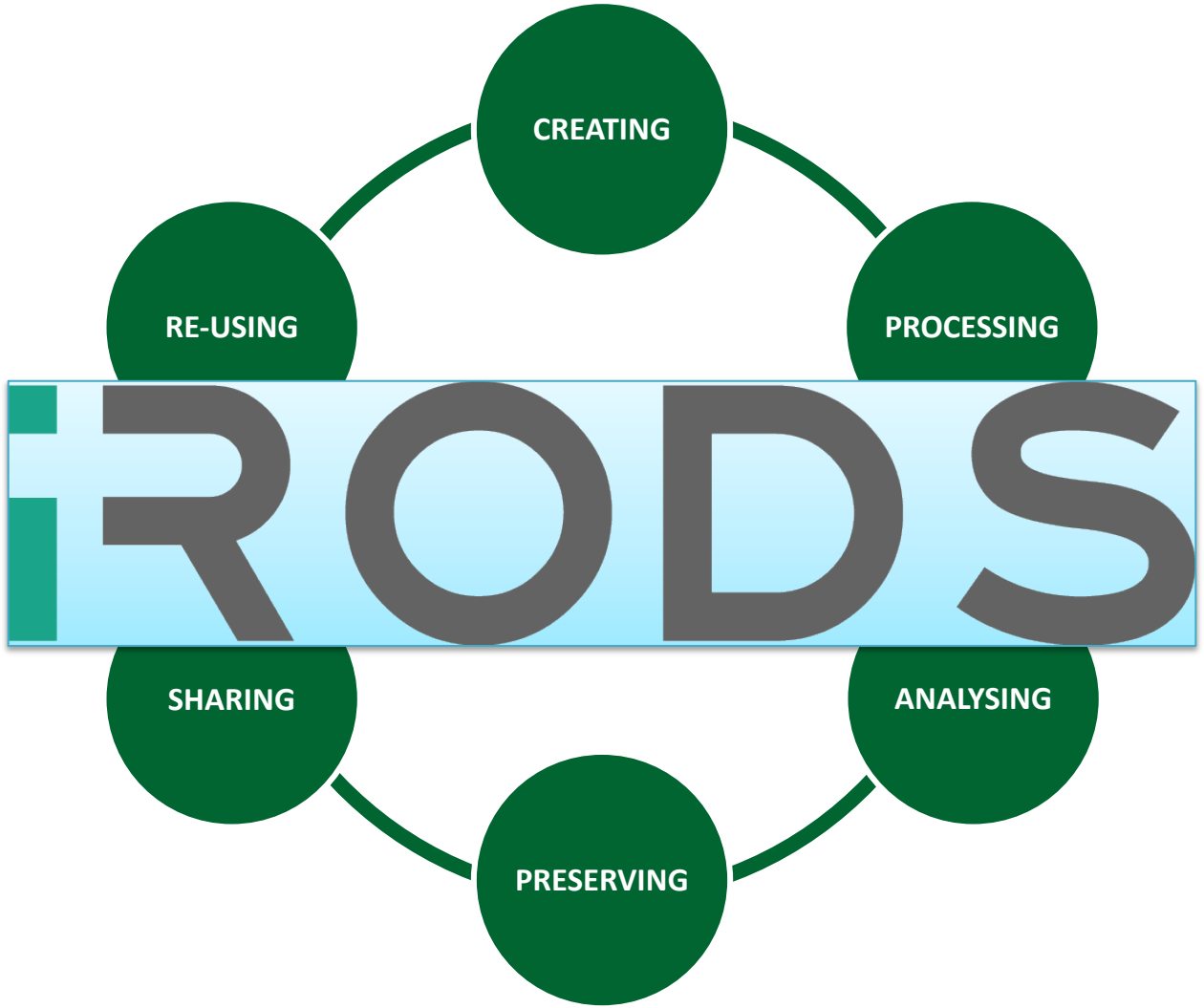
The FAIR principles

- **Findable** – Easy to find by both humans and computer systems → Metadata
- **Accessible** – Stored for long term, accessed and/or downloaded with well-defined license and access
- **Interoperable** – Ready to be combined with other datasets by humans as well as computer systems;
- **Reusable** – Ready to be used for future research and to be processed further using computational methods.
- https://www.nature.com/articles/sdata201618

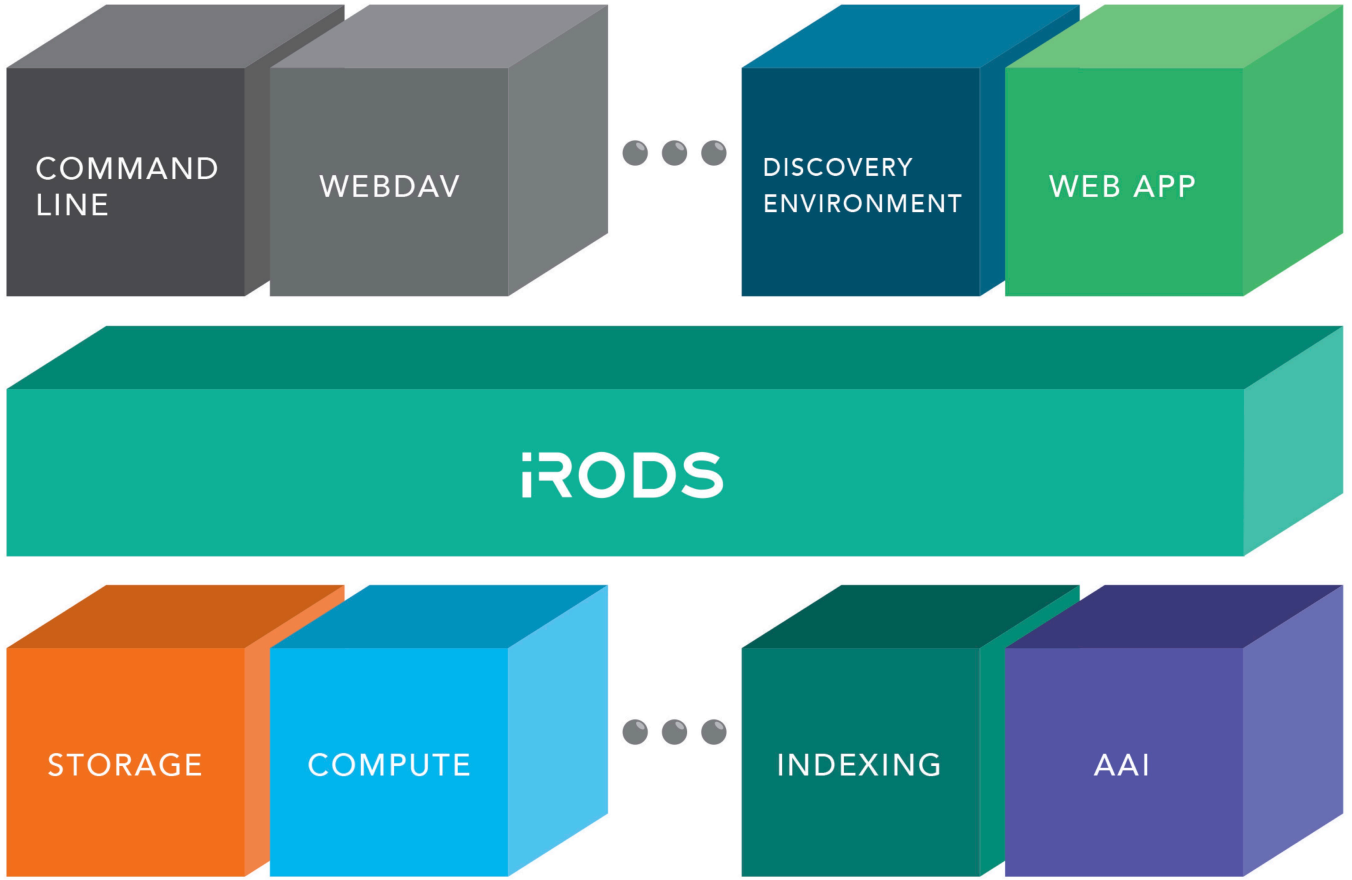
Researcher data needs

- **Store** data during research
- **Share** data during and after research
- **Archive** data
- **Synchronise** data across different locations, client – server, server-server synchronisation
- **Link** publication to processed and raw data
- **Publish** data
- **Find** data and **make data findable** by others
- Data **transfers**
- Data **provenance**: what happened with the data
- …

Data Life Cycle



Where is iRODS in a data infrastructure?



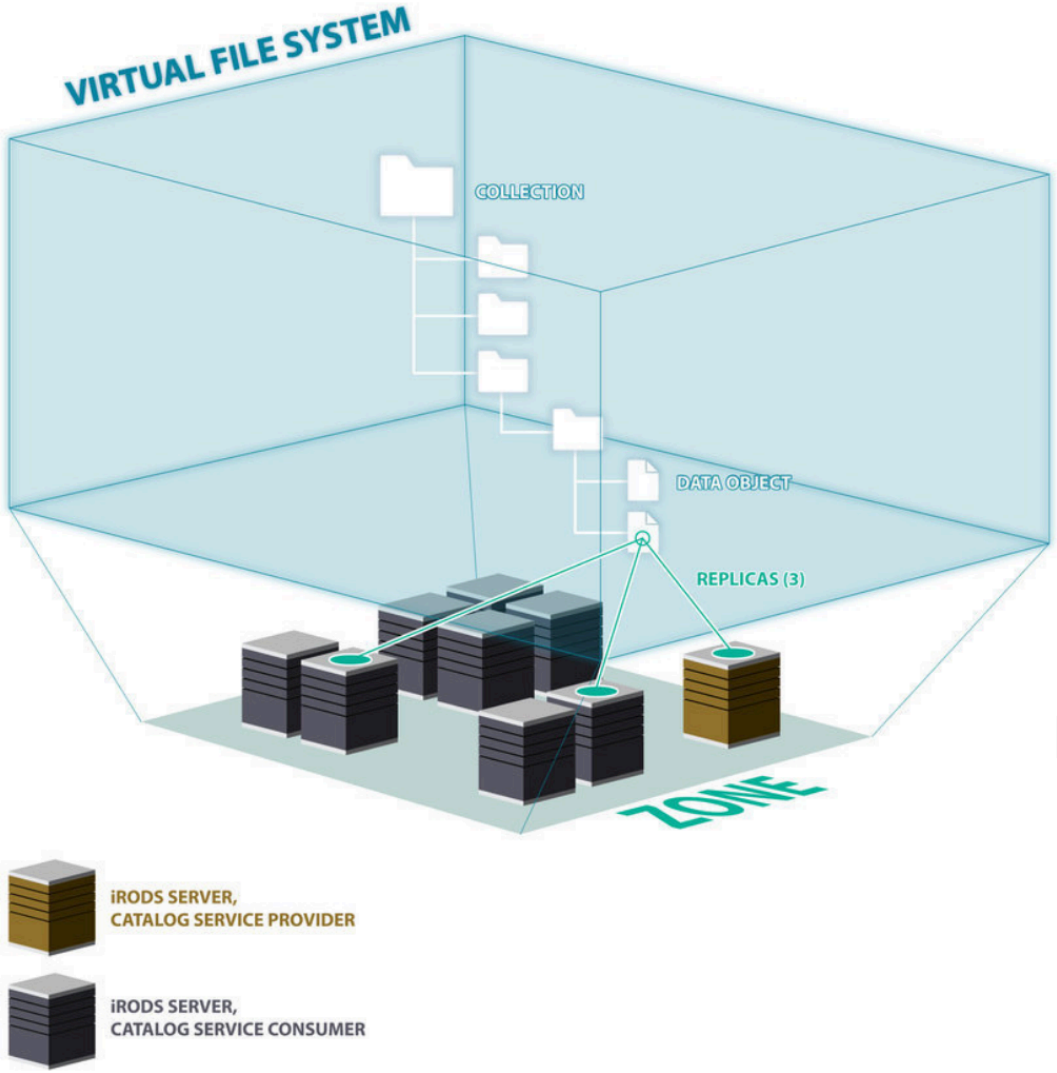
iRODS Clients

iRODS provides a layer of abstraction which integrates with your pre-existing infrastructure.

This flexibility allows your infrastructure to continue to change over time.

Existing Infrastructure

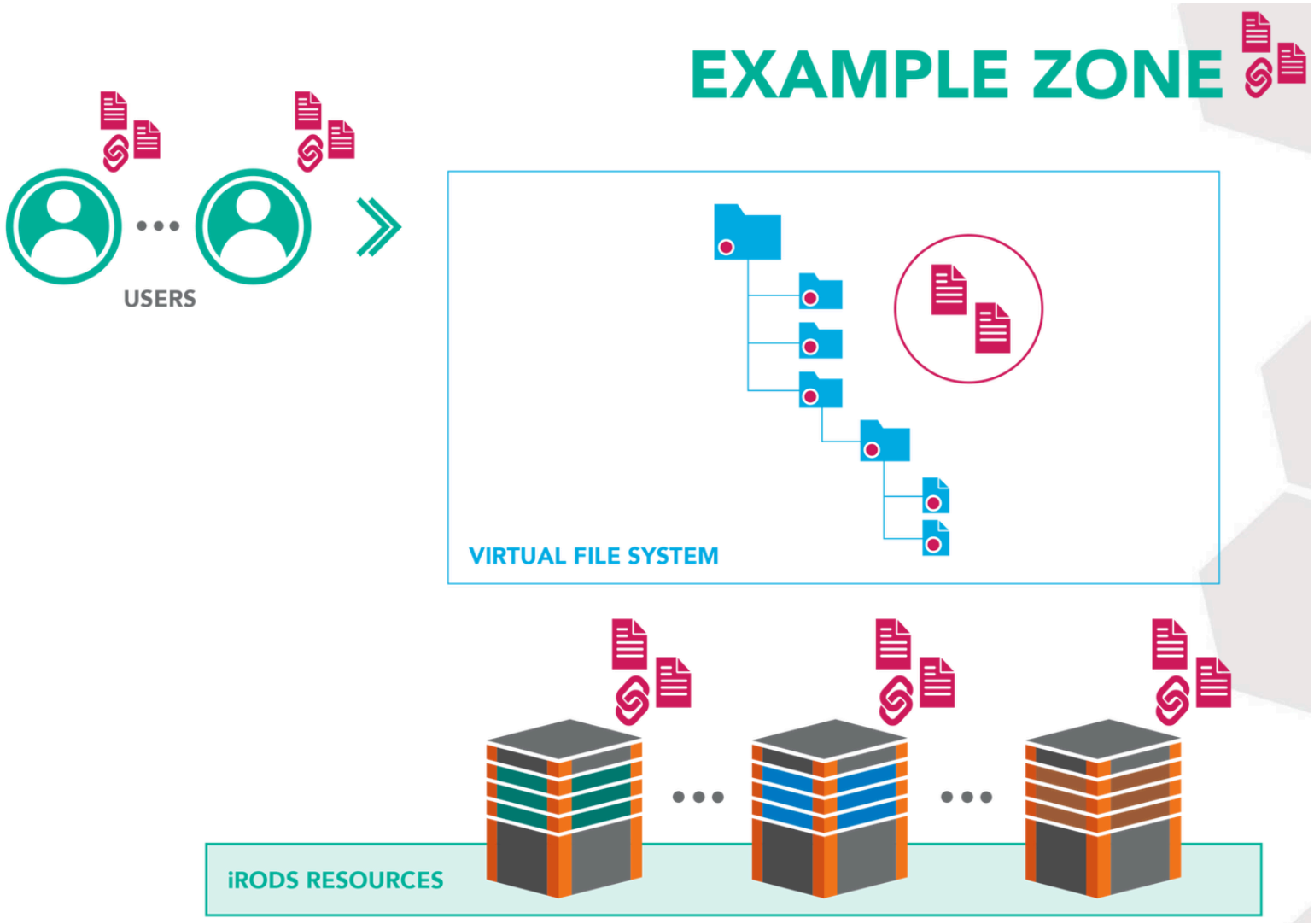
iRODS core feature: data virtualization



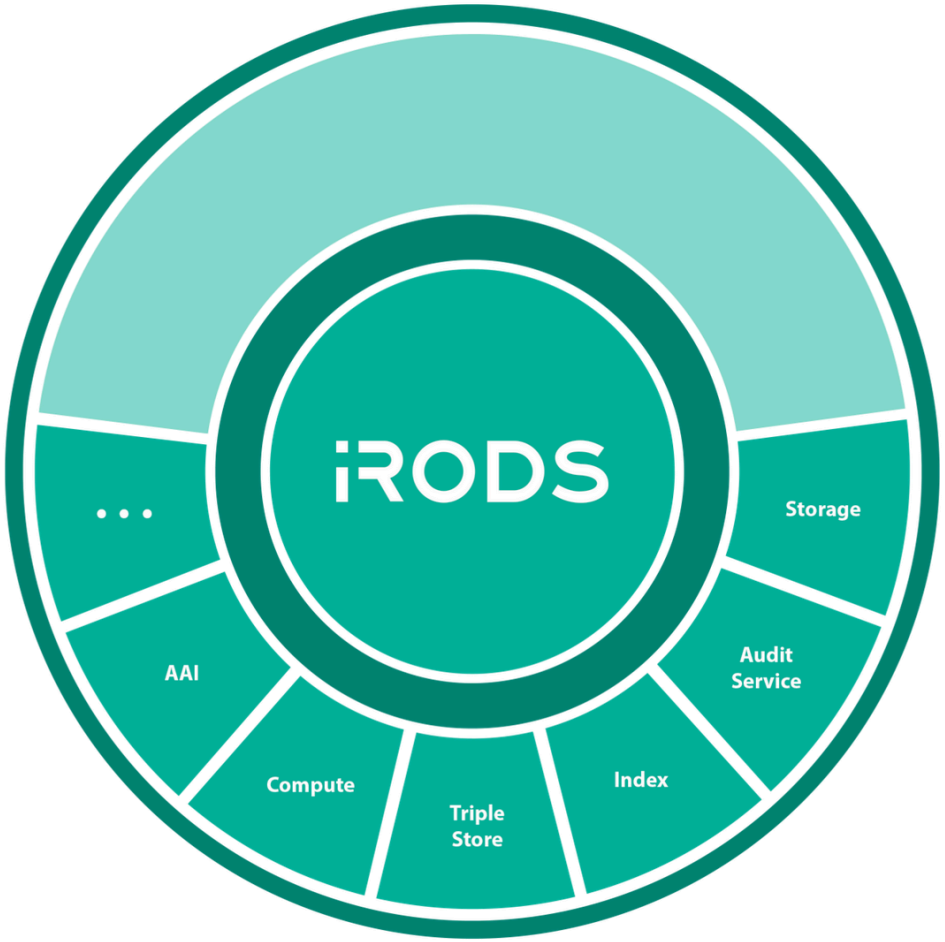
Logical Path

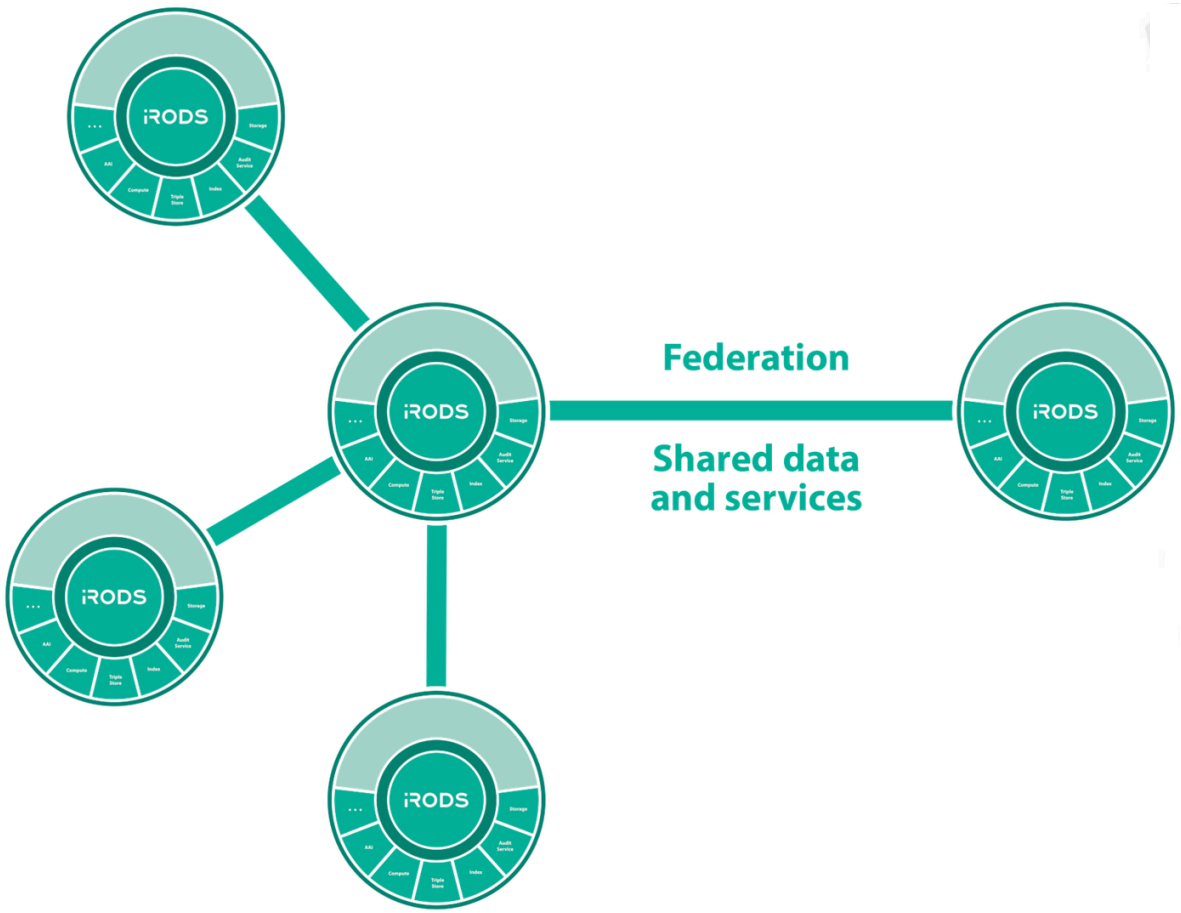
Physical Path(s)

iRODS core feature: metadata everywhere



iRODS core feature: federation





Data management platform based on iRODS

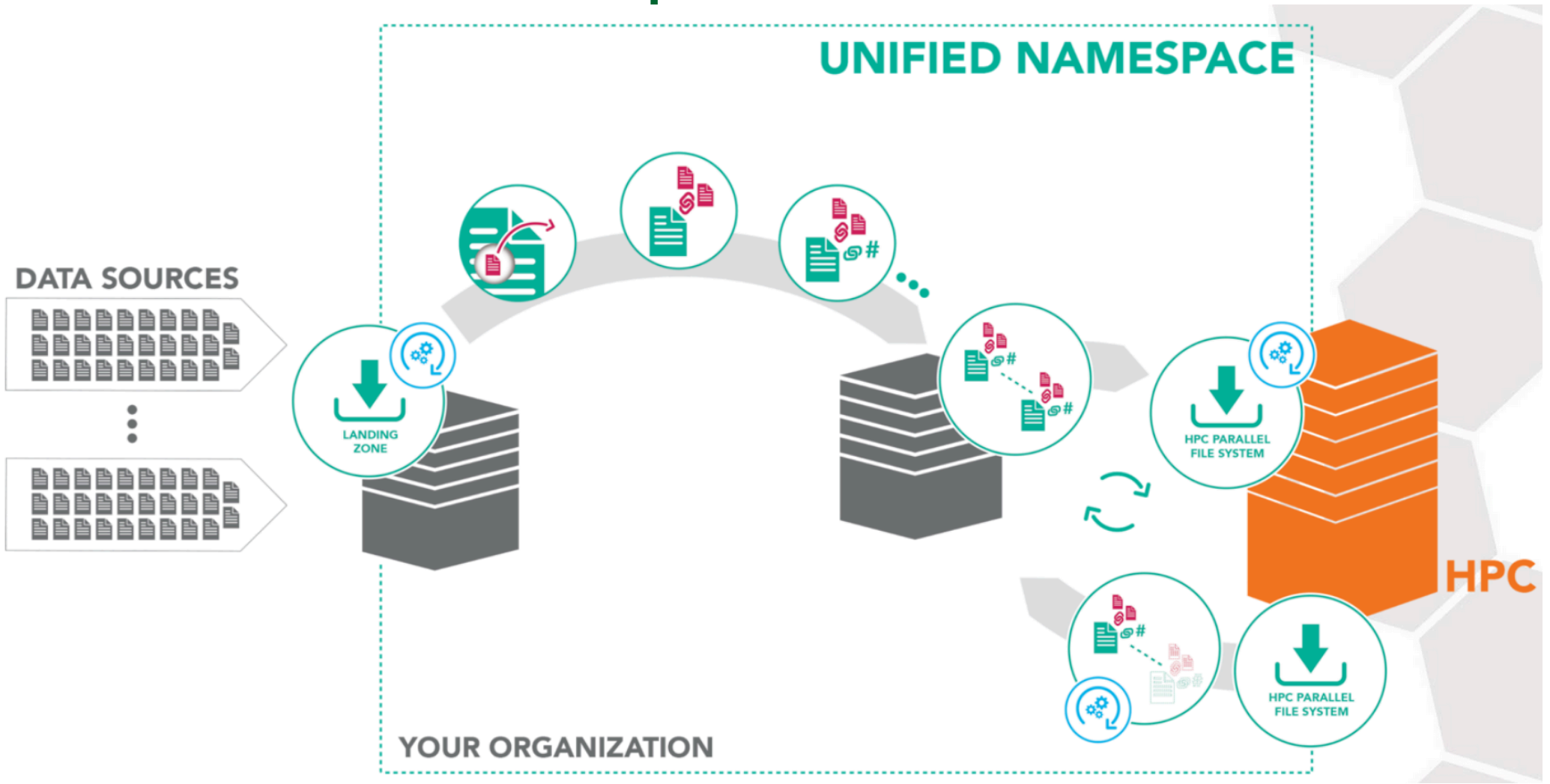
- **One entrance point** for the user to many storage services
- **Good interface to compute services and other data applications**
- **Data policies:** configure the behavior of data throughout the data life cycle
- **Data sharing** within research groups and with external collaborators

- **Findable:** within the data management platform, external findability by e.g. persistent identifiers ✅
- **Accessible:** coupling with federated identities, accession control lists, well-defined transfer protocols ✅
- **Interoperable:** strongly dependent on data formats and employed metadata standards ❌
- **Reusable: Depending on implemented data policies;** within the limits of metadata annotation and standards for data formats ☑️

Why iRODS for HPC?

- High performant data transfer protocol accessible via *e.g.* icommands, iRODS out of the box CLI tools
- Programmatic access to data through APIs : C++, Python, Java, Golang, R, REST
- Open source

iRODS computation workflows



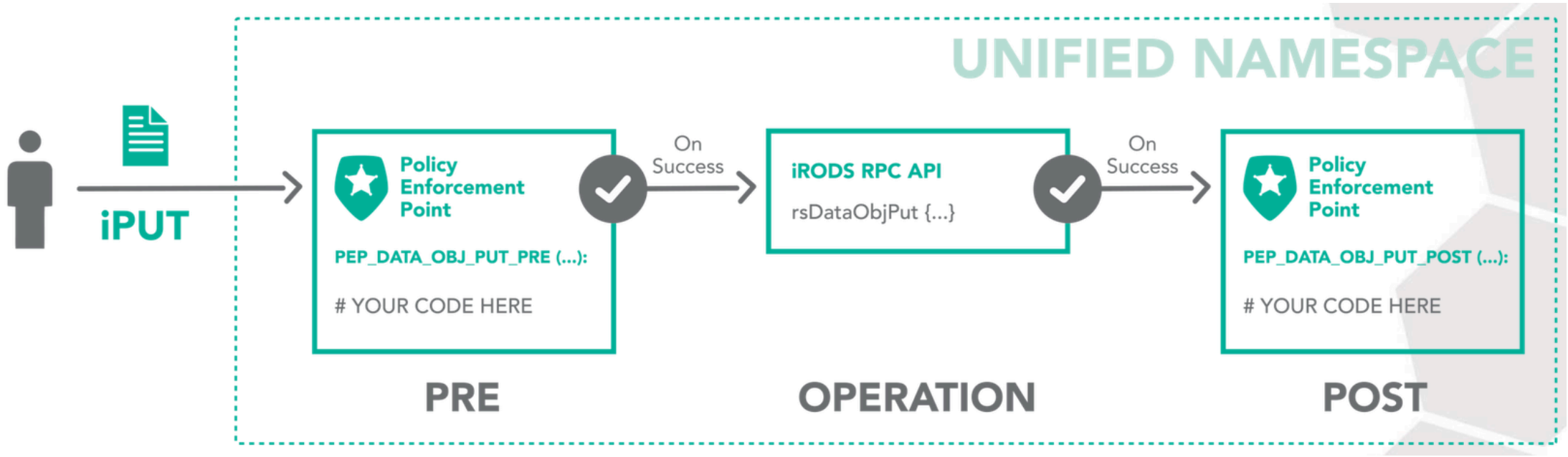
Hands on for today

- Learn how to programmatically access data stored in iRODS via the iRODS python API
- Learn about iRODS concepts: data objects, collections, metadata handling and querying
- Learn how to find data based on metadata (not on some knowledge of hand made directory trees) and use it in an HPC system

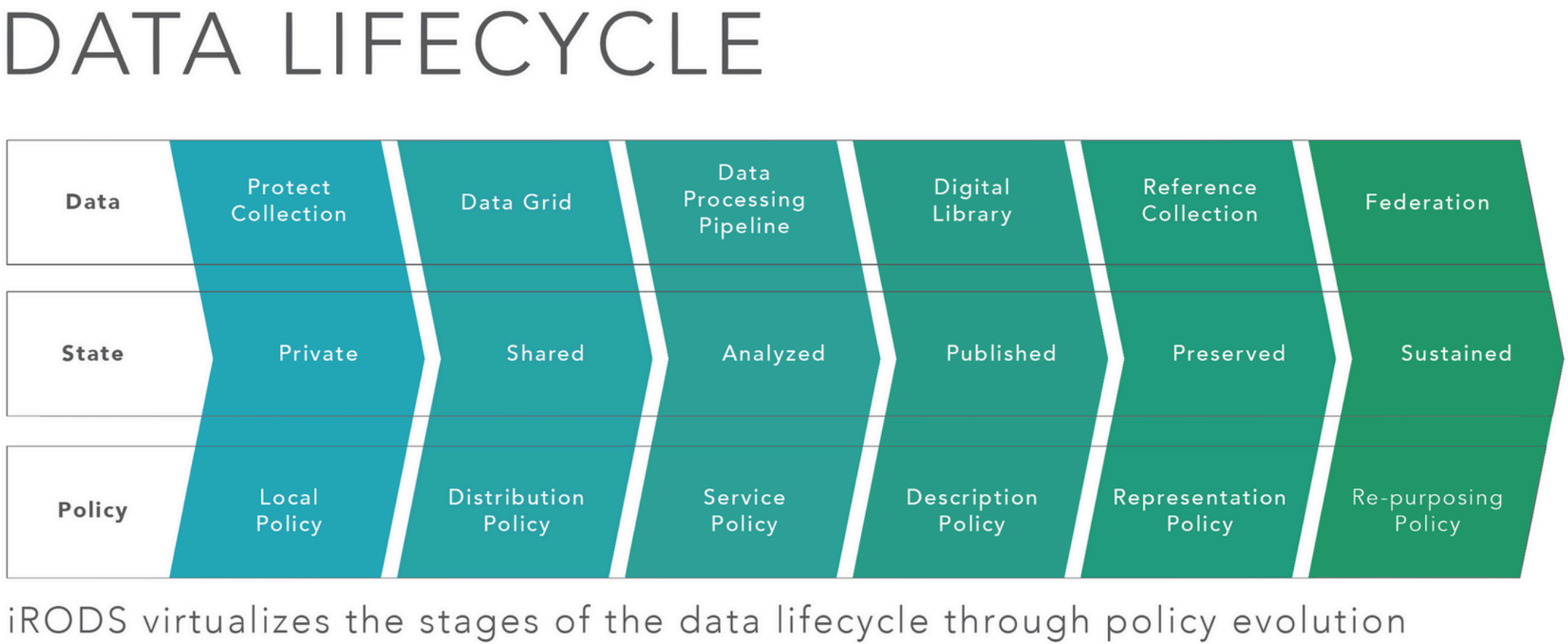
Some questions for the audience

- Who got a sdemo (or your own) account for Lisa?
- Who changed their password and was able to login to Lisa via a ssh client?
- Who knows the basics of Python?
- Who knows what iPython is?
- Who knows iRODS already?
- Who is doing data intensive compute jobs already on Lisa?
- Who is ready for the course!?

iRODS core feature: dynamic policy enforcement



iRODS can support the entire data life cycle



iRODS virtualizes the stages of the data lifecycle through policy evolution