Integrating Data management into Compute Workflows

Arthur Newton, Christine Staiger

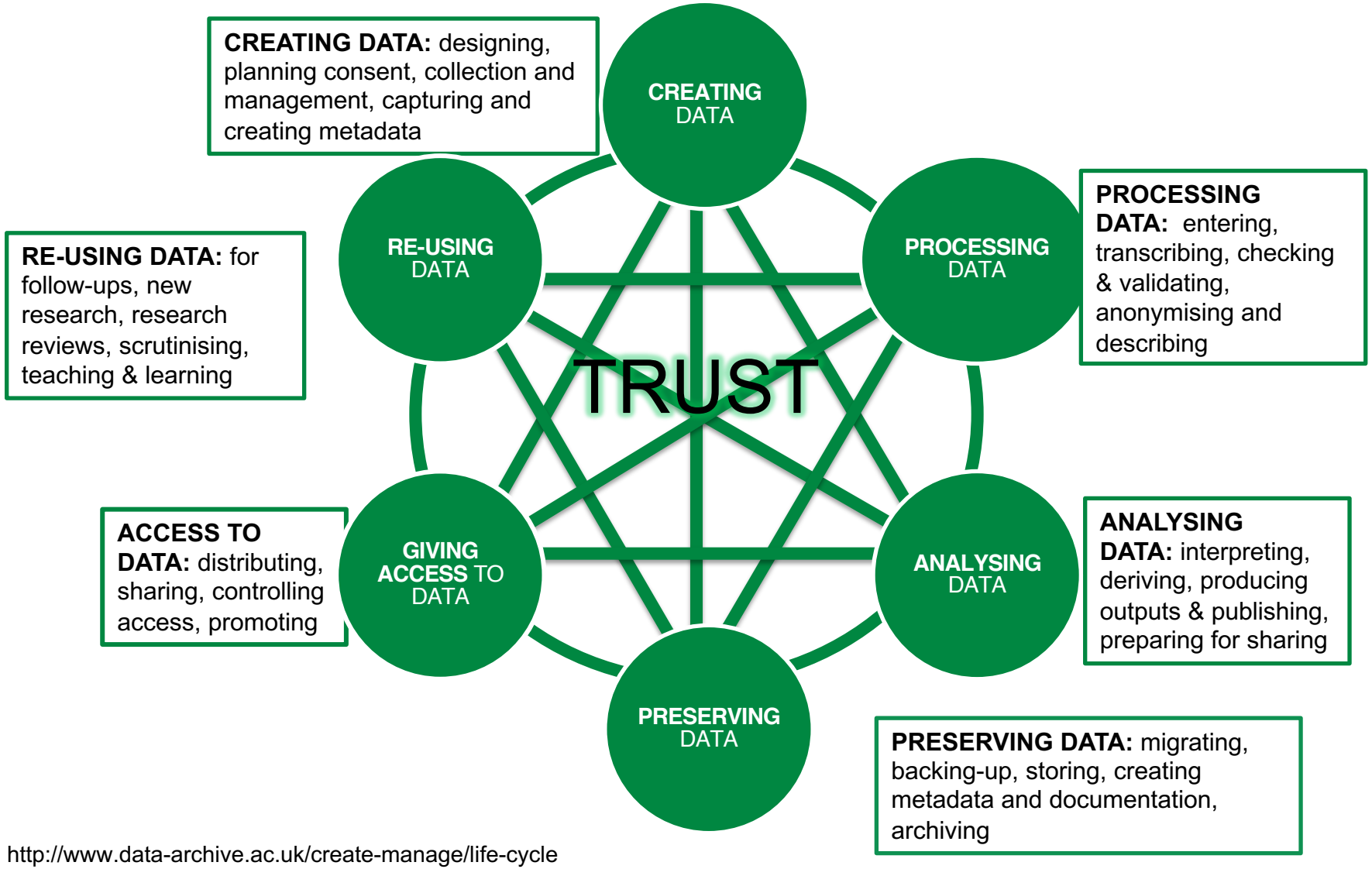
PRACE

SURFsara, Amsterdam, 6th Sept. 2018

Agenda

9:30 – 9:45	Welcome and Introduction
9:45 – 10:30	Data Management, FAIR and iRODS
10:30 – 11:15	**Hands-on**: Data handling with the python API
11:15 – 11:30	**Coffee break**
11:30 – 12:30	**Hands-on**: Data handling with the python API
12:30 – 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**

The data Life cycle



CREATING DATA: designing, planning consent, collection and management, capturing and creating metadata

PROCESSING DATA: entering, transcribing, checking & validating, anonymising and describing

ANALYSING DATA: interpreting, deriving, producing outputs & publishing, preparing for sharing

PRESERVING DATA: migrating, backing-up, storing, creating metadata and documentation, archiving

ACCESS TO DATA: distributing, sharing, controlling access, promoting

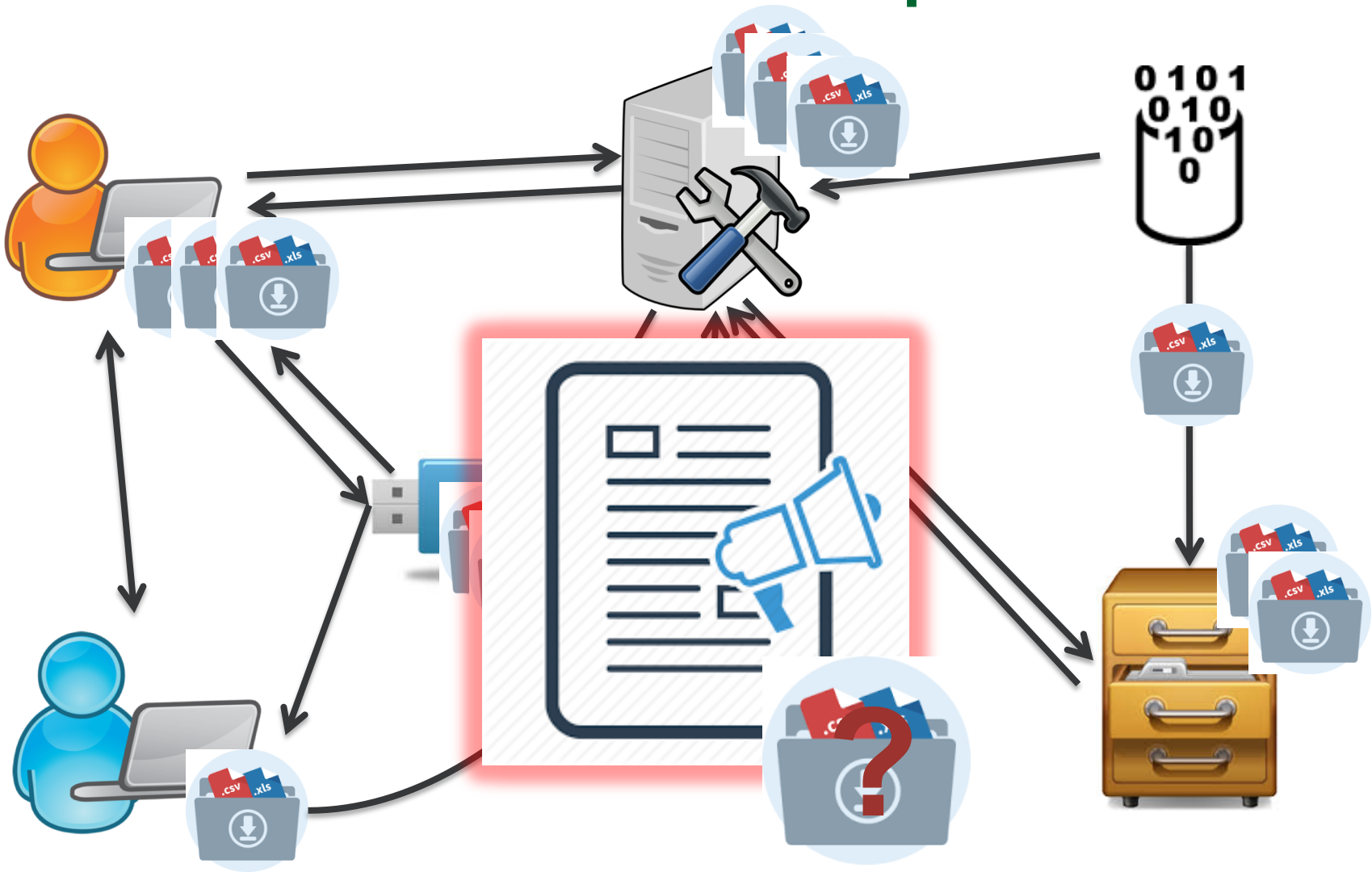
RE-USING DATA: for follow-ups, new research, research reviews, scrutinising, teaching & learning

http://www.data-archive.ac.uk/create-manage/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.
- http://www.datafairport.org/

Data – where is the problem?



The researchers' 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

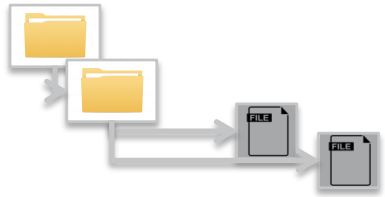
- …

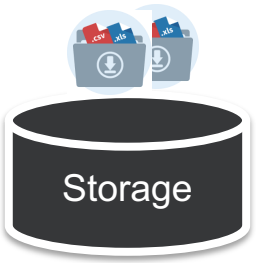
Storage – The users' challenge

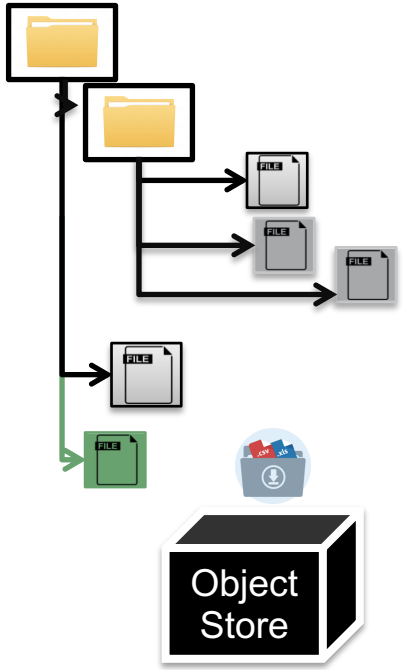
Offer:
Different storage media
Different protocols to steer data
Different clients and user interfaces
Different services

Needs:
Proper book-keeping
Knowledge about storage infrastructure

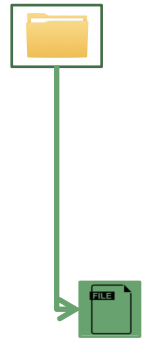
Local storage system



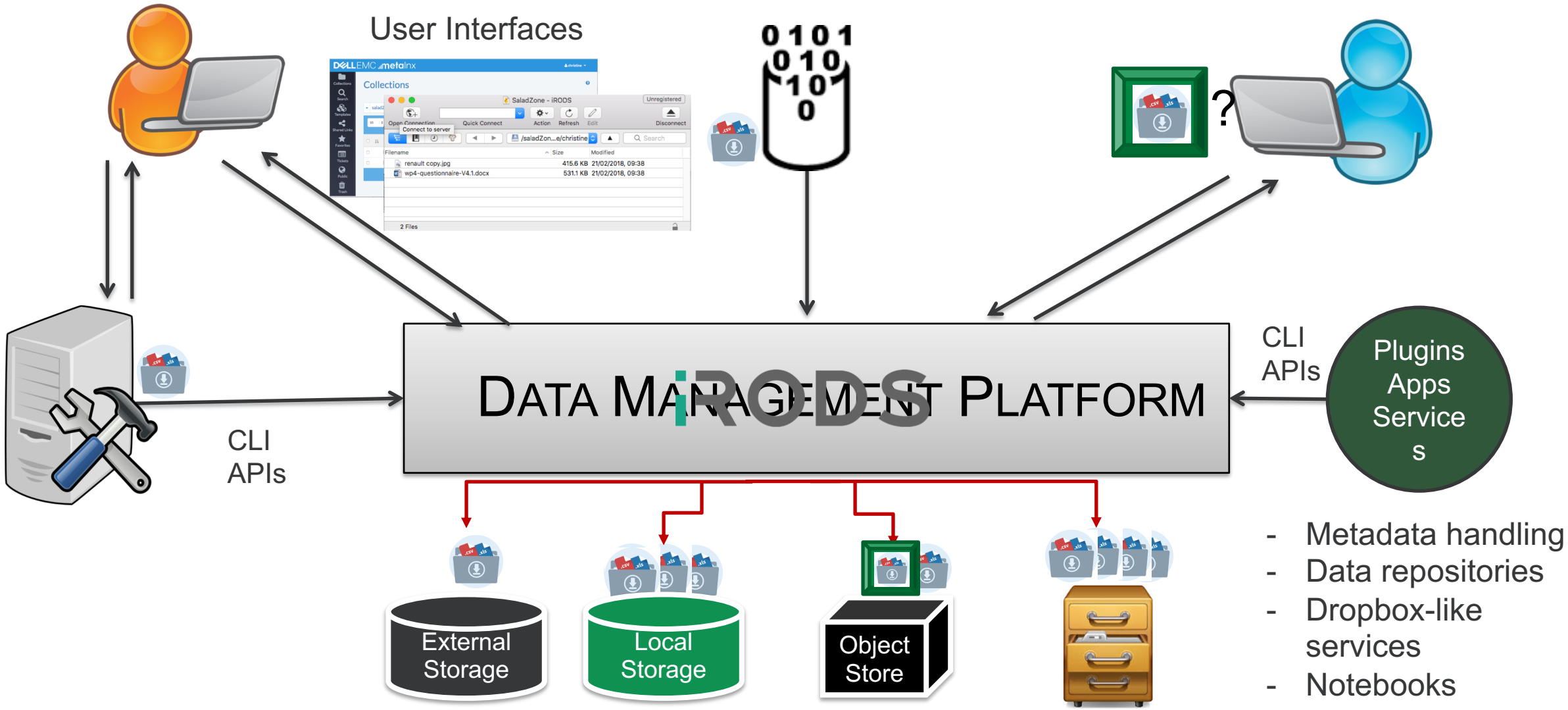




SURF data archive



A solution – Data Management Platform



- Metadata handling
- Data repositories
- Dropbox-like services
- Notebooks

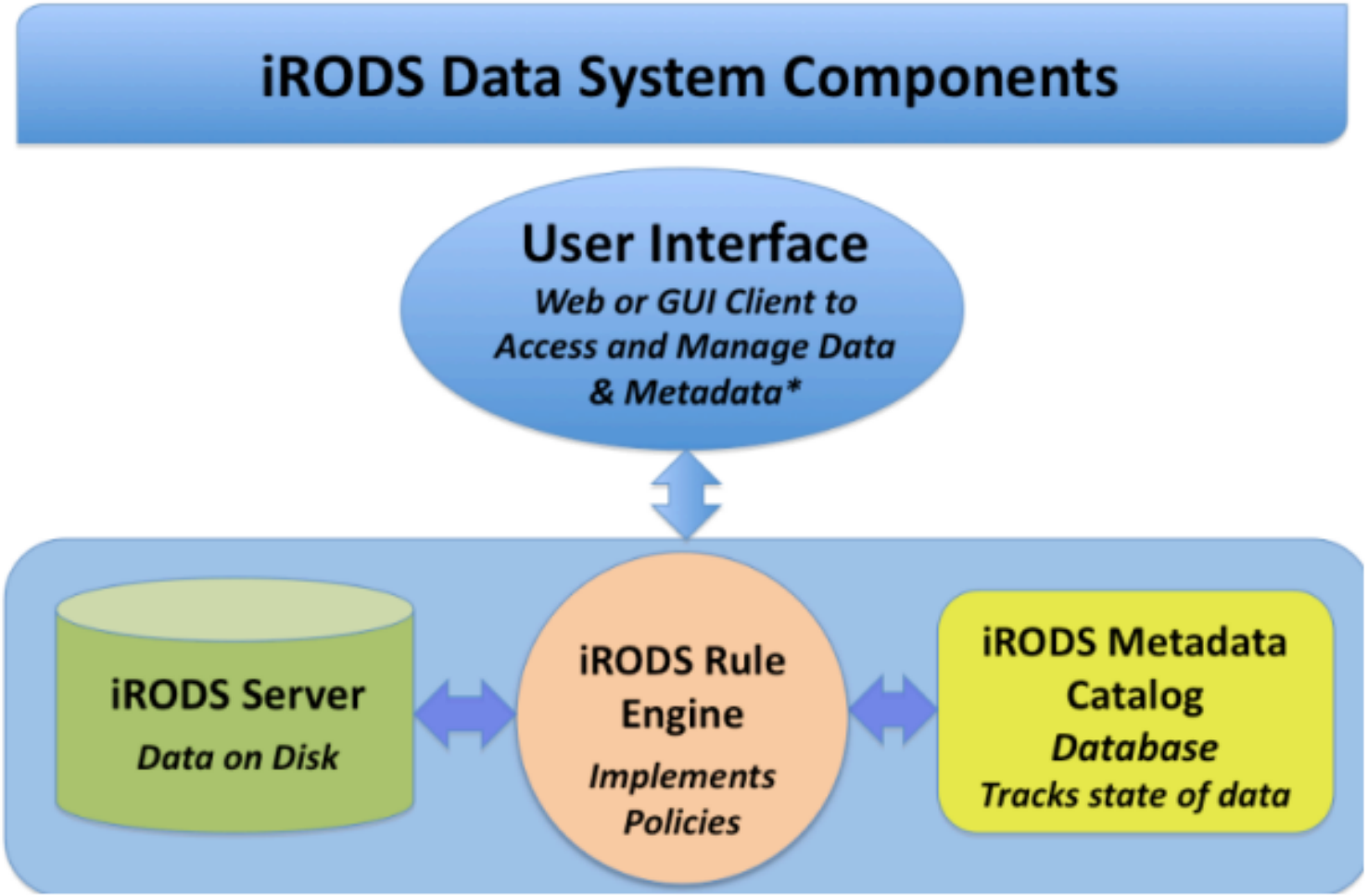
Data management platform

- **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

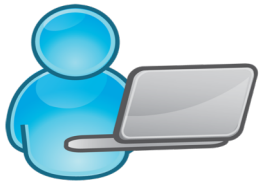
- **Findable:** within the data management platform, external findability by e.g. persistent identifiers ✅
- **Accessible:** coupling with federated identities, accession control lists, well-defined transfer protocols ✅
- **Interoperable:** strongly dependent on data formats and employed metadata standards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

iRODS

The iRODS system



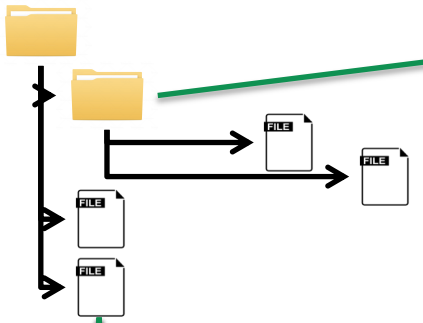
The iCAT Metadata Catalogue



User information:
Name
Groups
Access rights
Roles

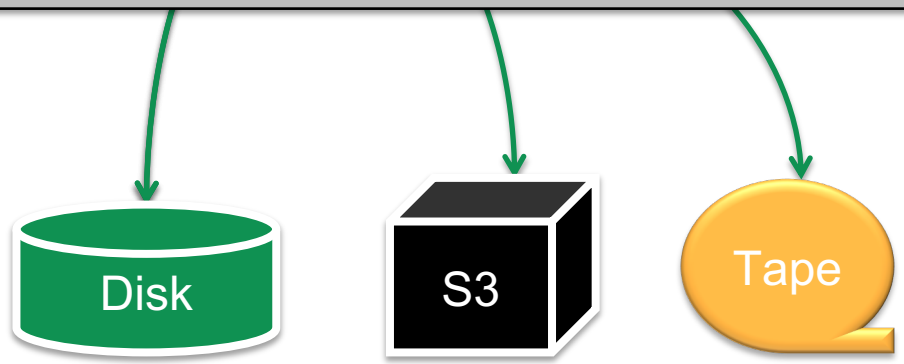
Abstraction layer:
Mapping from logical to physical namespace

Storage layer:
Different storage media
Different protocols to steer data

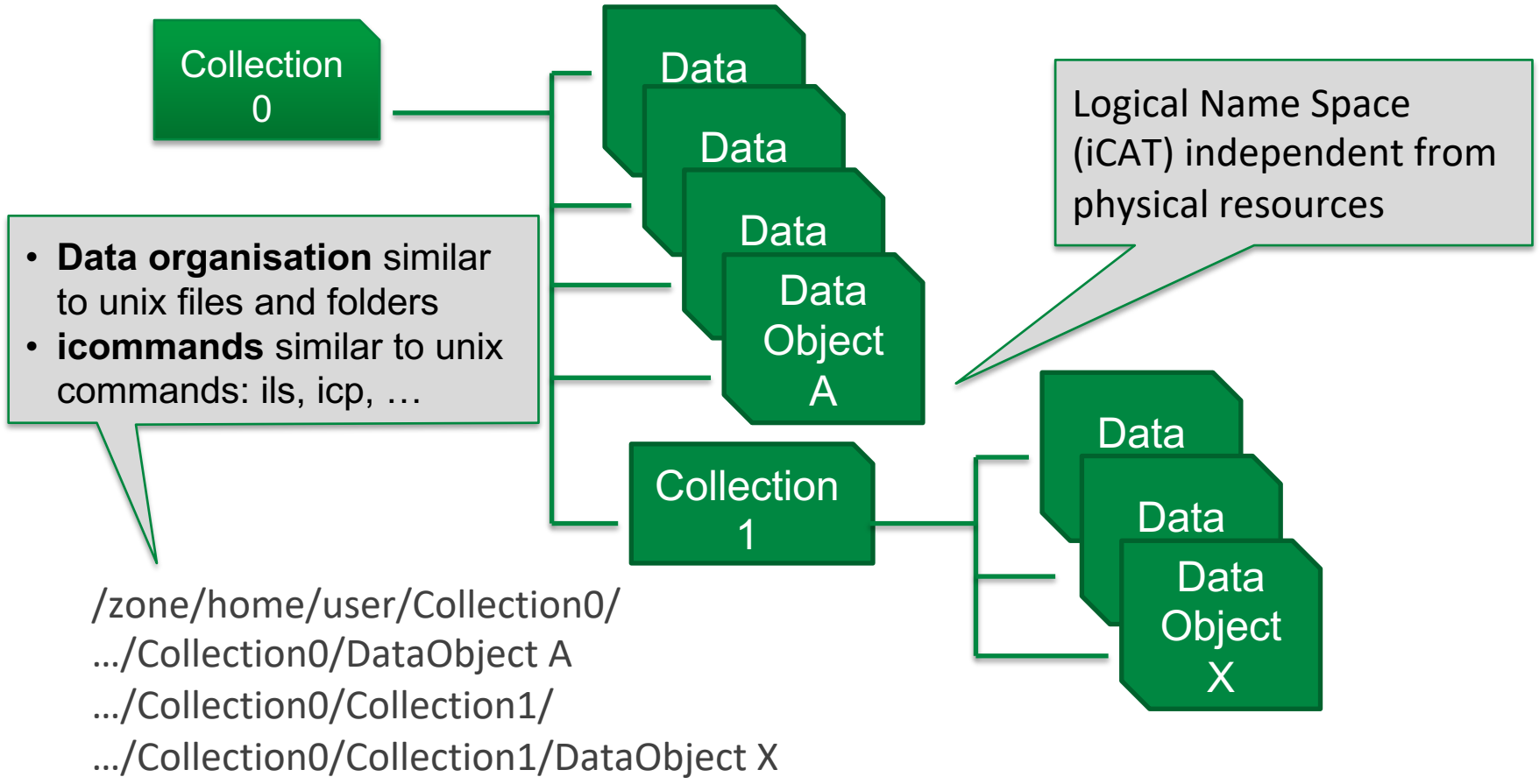


+ Extra information:
attribute: distance
value: 12
units: miles
---
attribute: author
value: Alice
units:

iCAT (iRODS metadata catalogue):
/irodsZone/home/user/Collection0/testfile.txt →
/irodsVault/home/rods/testfile.txt

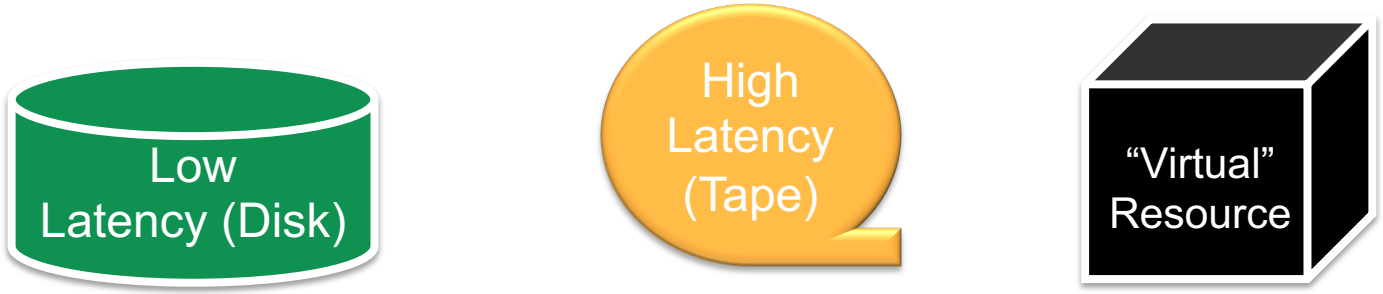


The users view:
iRODS Data Collections



In the Background: iRODS resources

- (Storage) Resource is a Software or Hardware system that stores data
- 3 Resource classes:



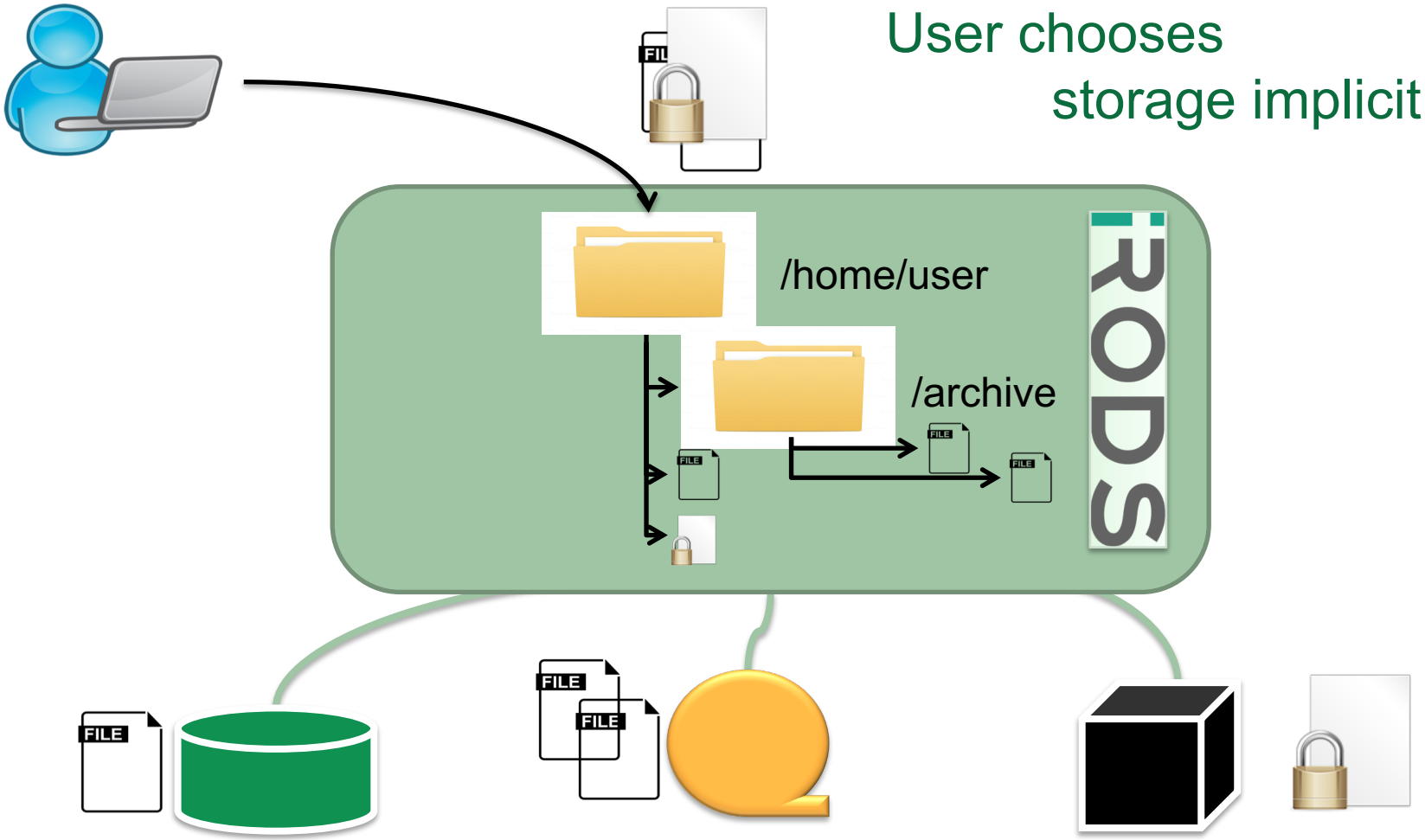
- Storage Resource: unix file system, s3, structured file type, univMSS, opendap, tds (THREDDS)

What does iRODS provide?

- **Storage virtualization** of different disk and tape storage systems
- A **logical namespace** across storage locations
- A **rule engine** to automate data management according to defined policies
- **Federations** between iRODS instances

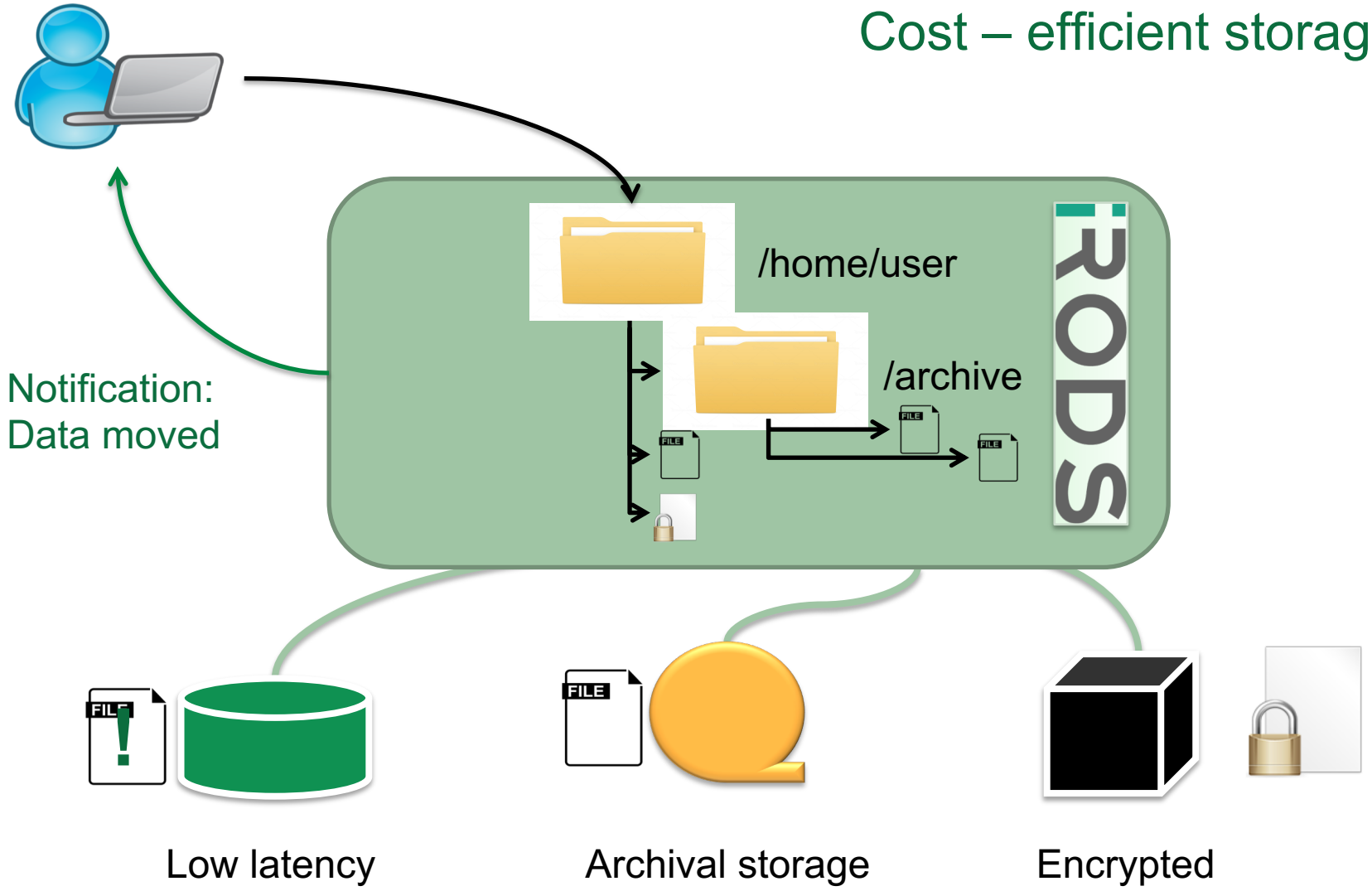
User policies

User chooses
storage implicitly

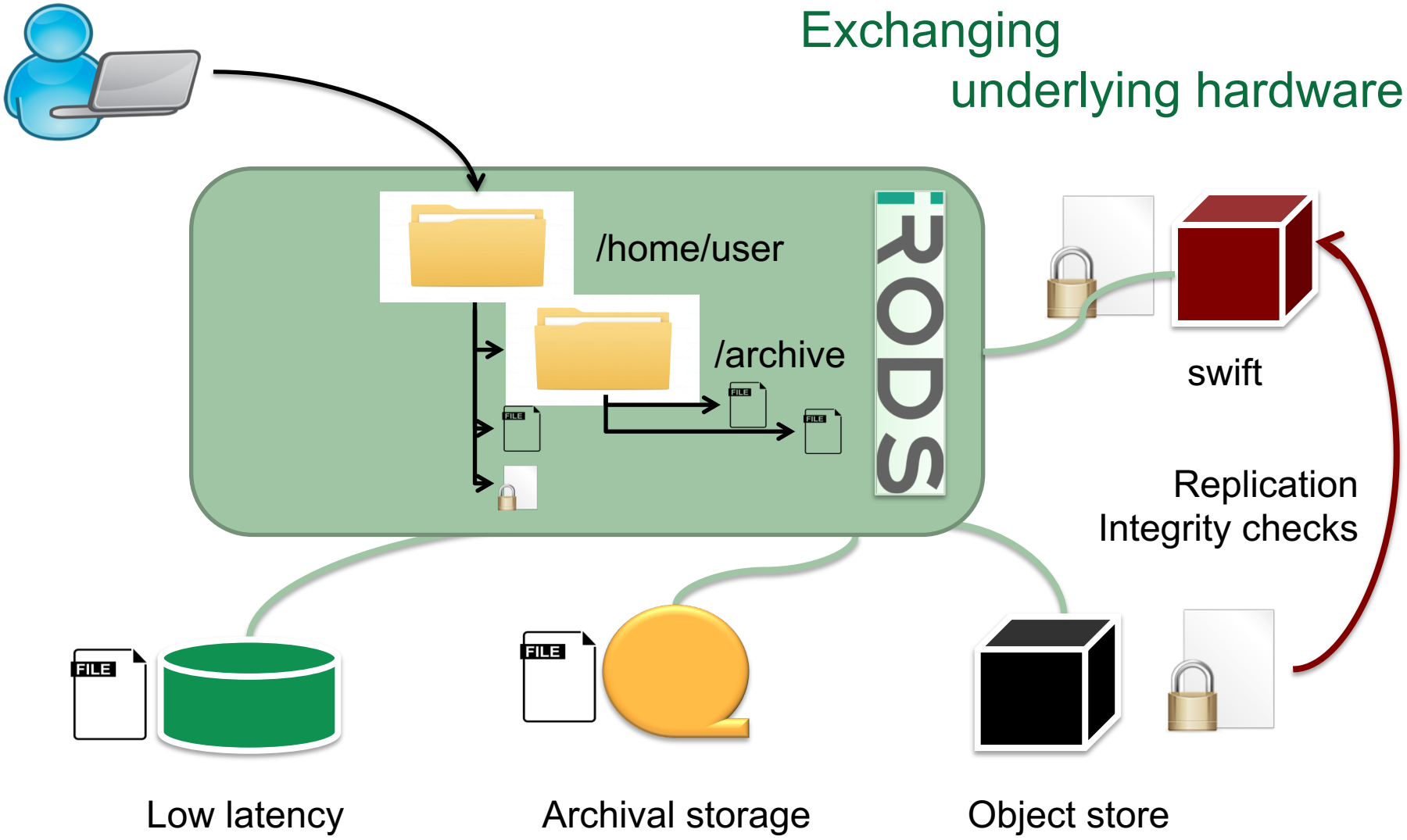


System policies

Cost – efficient storage



Maintenance policies



iRODS is …

- Not meant for **ad hoc data management**
- **Overkill** when you decided for **one and only one storage system**
- **Not a storage monitoring system**
- **Not an accounting system**
- **Not a syncing client or publishing platform**

Use iRODS when …

- Managing data across **different administratory domains**
- Combining **several storage systems** that you want your users to **access and steer in a uniform way**

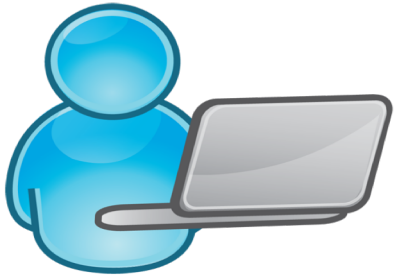
  → spare users to employ tons of different protocols
- Scale out easily → simply **plug in new storage** system
- **Automatise** data management: execute data workflows regularly or upon action frequently

Training Setup

Training setup

Lisa login node

login



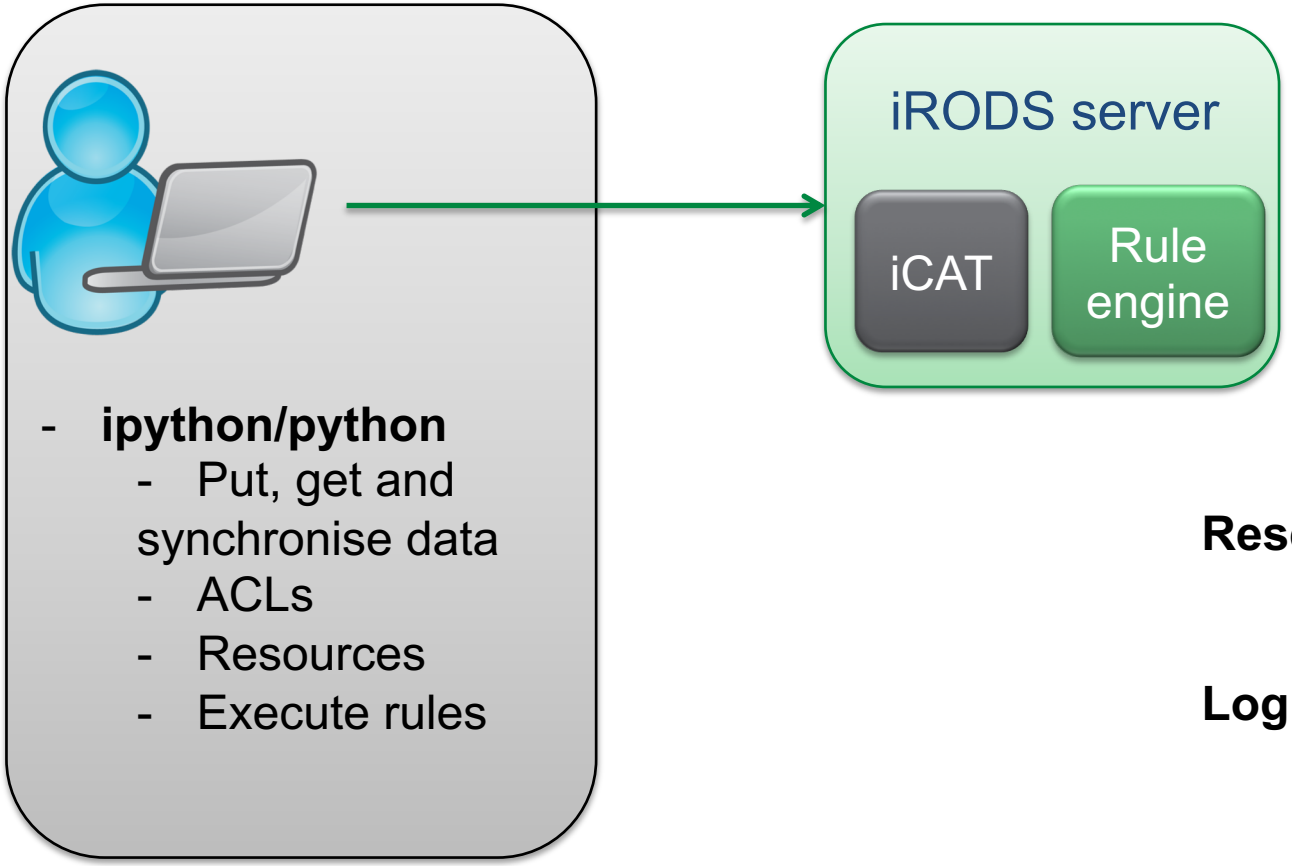
Python API

1. Connect to iRODS
2. Up and down load data
3. Annotate data
4. Search for data in iRODS

Training setup

lisa.surfsara.nl

sara-alice.grid.surfsara.nl



- **ipython/python**
  - Put, get and synchronise data
  - ACLs
  - Resources
  - Execute rules

Reset password:

https://portal.surfsara.nl/password

Login User Interface

ssh sdemo@lisa.surfsara.nl

Password: …

Thank you! Questions?

Slides based on PRACE iRODS training:

Zheng Meyer-Zhao, Arthur Newton, Christine Staiger
Data Management with iRODS, Sept 2017
https://events.prace-ri.eu/event/638/

arthur.newton(at)surfsara.nl
christine.staiger(at)surfsara.nl

SURF SARA