Storage QoS, Caching, Protocol Hiding

Paul Millar (on behalf of WP2.2)

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Overview

- Introduction:
  - QoS
  - Caching
  - “Protocol hiding”
- Possible topics for discussion

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Storage QoS

Everything you always wanted
to know about storage but
were afraid to ask

Funded by the European Union's
Horizon 2020 - Grant N° 824064

How expensive is recreating lost data?

- Data from telescopes/detectors: **very expensive**.
  - Researcher would need to reapply for observation/beam time.
  - For telescope observations, data might be irreplacable.
  - Overall, considerable expense and loss of research time.
- Output from simple file processing: **almost no cost**.
  - No need to book time at the facility.
  - Recovery would be very quick, no significant loss of research time.
- Data-loss can **very different costs**, depending on the data.
- Making storage robust against data-loss is **expensive**.

  Why do it for data where durability doesn't matter so much?

Funded by the European Union's Horizon 2020 - Grant N° 824064

How “fast” does data access need to be?

- Data being processed by HPC/GPGPU farm: **very fast**.
  - HPC / GPGPU resources are expensive – limited resources.
  - Want to maximise use – Do not want HPC/GPU waiting for data.
- Long-term data storage: **not fast at all**.
  - Data where all expected processing has taken place.
  - Data stored to fulfil FAIR requirements.
  - Can afford for it to take time to read data.
- Data has a range of **different performance requirements**.
- Making storage high performance is **expensive**.

  Why store data on high-performance capacity when the data doesn’t need it?

Funded by the European Union’s Horizon 2020 - Grant N° 824064

How to build different storage options?

- "Disk" → "Disk" + TAPE
- "Disk" → Different options from …

  Media layer: PMR / NVMe-SSD / SMR / [HAMR, MAMR, SSD-low-endurance, …] / TAPE

  Media compositing layer: JBOD/RAID-0, RAID-5, RAID-6, RAID-Z, RAID-Z1 …

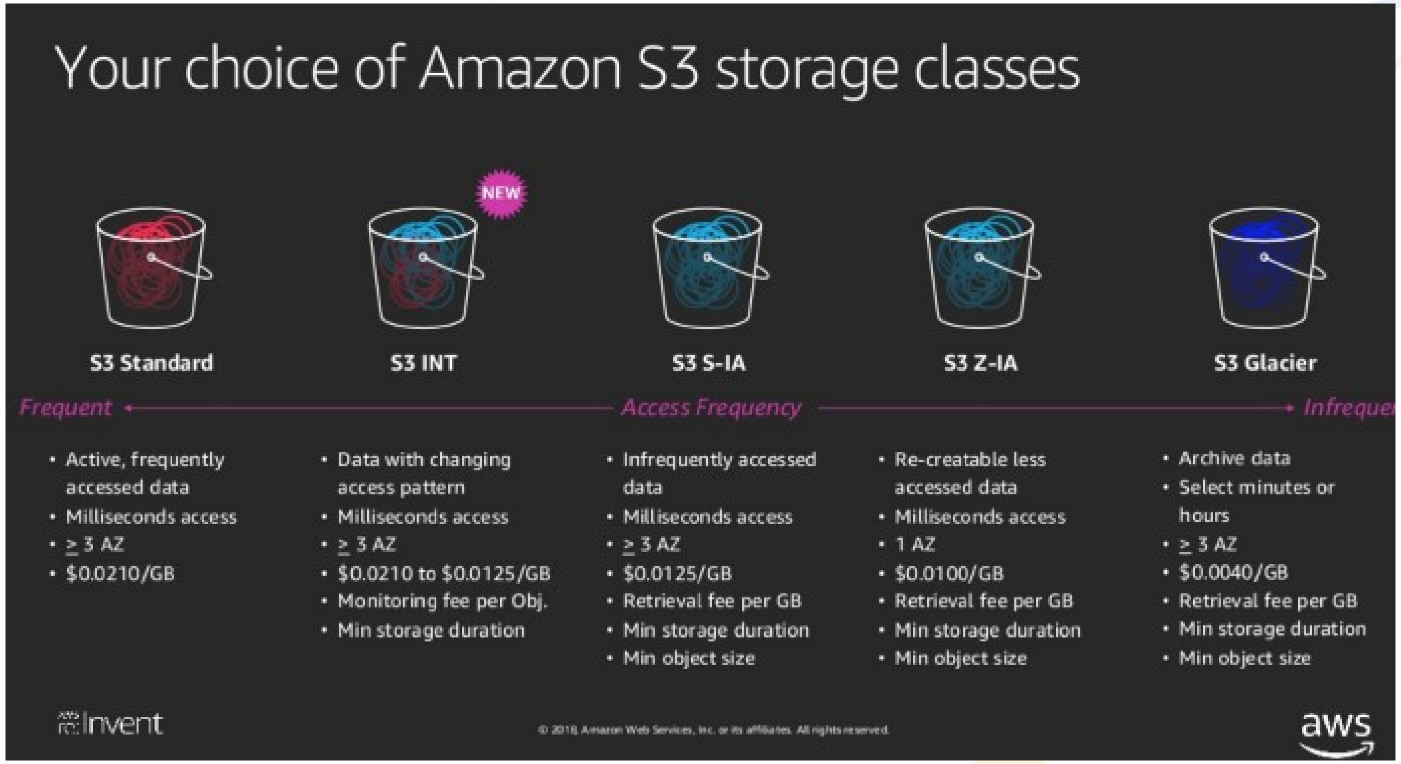
  Intra-cluster compositing layer: Single-copy, Replication (n), Erasure code (n,m)

  Inter-storage compositing layer: How many replicas? … which which characteristics?

- As a researcher you **don't want to see** this complexity!
- Abstraction is a common solution: **hide the details**
  - Instead, you say what kind of storage you want in general terms, and the system uses the most reasonable available option.
- This abstration is called **(storage) QoS**.

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Storage QoS is what industry is doing

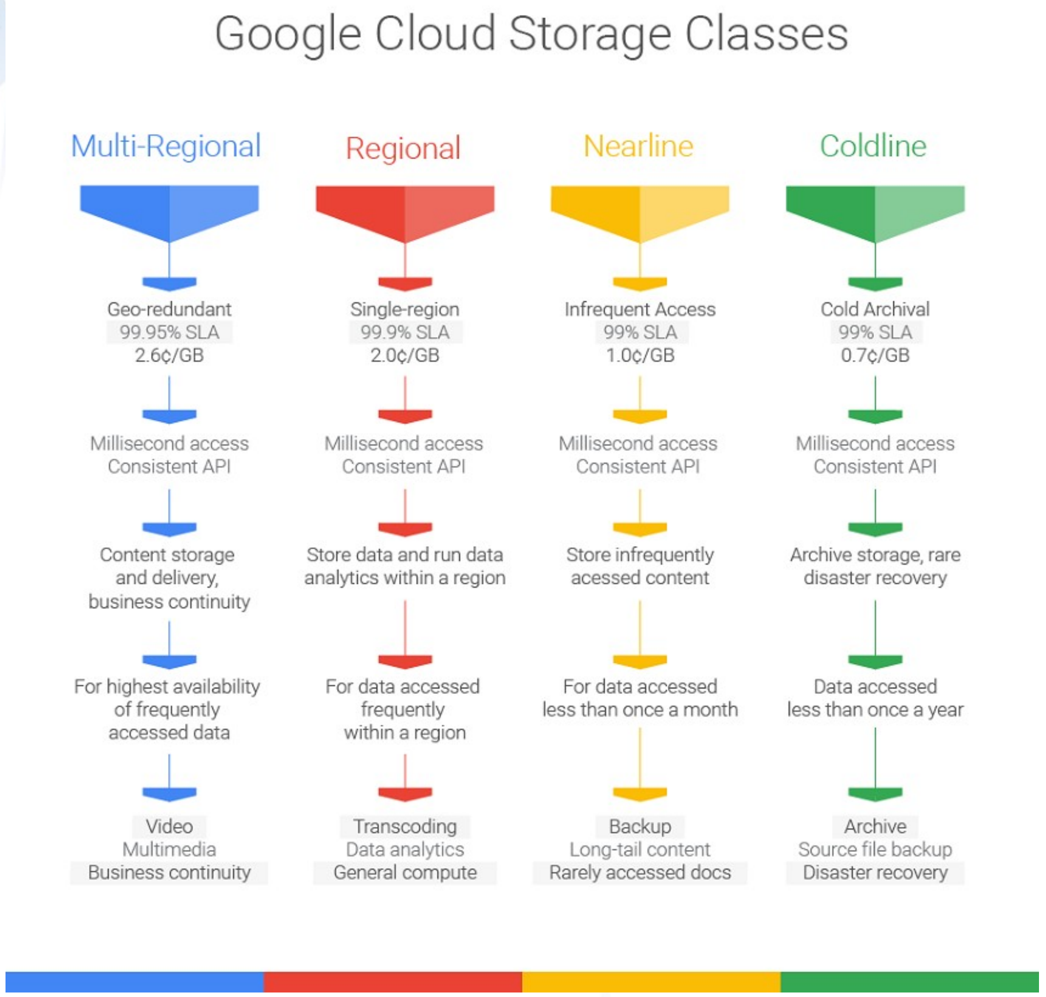


Your choice of Amazon S3 storage classes

Frequent ← *Access Frequency* → *Infreque*

S3 Standard	S3 INT (NEW)	S3 S-IA	S3 Z-IA	S3 Glacier
• Active, frequently accessed data • Milliseconds access • ≥ 3 AZ • $0.0210/GB	• Data with changing access pattern • Milliseconds access • ≥ 3 AZ • $0.0210 to $0.0125/GB • Monitoring fee per Obj. • Min storage duration	• Infrequently accessed data • Milliseconds access • ≥ 3 AZ • $0.0125/GB • Retrieval fee per GB • Min storage duration • Min object size	• Re-creatable less accessed data • Milliseconds access • 1 AZ • $0.0100/GB • Retrieval fee per GB • Min storage duration • Min object size	• Archive data • Select minutes or hours • ≥ 3 AZ • $0.0040/GB • Retrieval fee per GB • Min storage duration • Min object size

re:Invent © 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. aws



Google Cloud Storage Classes

Multi-Regional	Regional	Nearline	Coldline
Geo-redundant 99.95% SLA 2.6¢/GB	Single-region 99.9% SLA 2.0¢/GB	Infrequent Access 99% SLA 1.0¢/GB	Cold Archival 99% SLA 0.7¢/GB
Millisecond access Consistent API	Millisecond access Consistent API	Millisecond access Consistent API	Millisecond access Consistent API
Content storage and delivery, business continuity	Store data and run data analytics within a region	Store infrequently acessed content	Archive storage, rare disaster recovery
For highest availability of frequently accessed data	For data accessed frequently within a region	For data accessed less than once a month	Data accessed less than once a year
Video Multimedia Business continuity	Transcoding Data analytics General compute	Backup Long-tail content Rarely accessed docs	Archive Source file backup Disaster recovery

Funded by the European Union's Horizon 2020 - Grant N° 824064

Why use QoS?

- **Save money** – limited budget.
  - At some point “just buy more hardware” doesn’t work.
  - Get more done with the available budget.
- Provide tailored storage behaviour.
  - **Faster storage**, when processing data.
  - **More durable storage**, where storing important data.
- Support possible **future storage technology**.
  - You don’t know what changes tomorrow will bring. How will you handle changes needed when new storage types become available?

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

How to make use of QoS

- Data is **not homogenous**.
  - Some data has low(er)-value; for example, some data may be recreated if lost.
  - Some data is not used in time-critical analysis (e.g., log files).
- Data use **changes over time**.
  - Data fresh from camera/detector is of interest; month-old or year-old data is less interesting.
  - Only use expensive storage for embargoed data; public data is stored on cheaper (and slower) storage.
  - Pre-emptively move data to hot storage for thematic analysis runs.
- Different users/work-flows have **different needs**.
  - Identify “power users” (or work-flows) and store their data on more capable hardware.

… potentially more options! …

Funded by the European Union’s Horizon 2020 - Grant N° 824064

QoS in ESCAPE

- Building up **knowledge** within scientific communities.
  - Storage QoS is a new concept; new ideas are hard. It requires new ways of thinking about storage.
  - In ESCAPE, we are working on an iterative approach to building up case studies on how scientific communties can benefit from QoS.
- Working with **technology providers**.
  - Require some new concepts at the data management layer.
  - Storage technologies can be improved to make QoS transitions “faster” or “better”.
  - In ESCAPE, we are working closely with the Rucio development team to deploy and test new features, and with the storage providers to allow testing of these concepts.
- Proving **this really works**, in practice.
  - Use the ESCAPE testbed to gain (as close to) real-world experience as possible

Funded by the European Union’s Horizon 2020 - Grant N° 824064

General approach in Task 2.2

- **Capture** each ESFRI communities QoS requires.
  - Initial interview, to go through the document template, describing what are the different elements and what they mean.
  - The ESFRI representatives discuss the document with other scientists in their community.
    - This might trigger further, ad-hoc meetings to discuss details.
  - Once document is complete, it is presented at the T2.2 regular meetings.
- **Build** a coherent document that takes input from these documents
  - More than just chapters for each community.
  - An initial version, that will contain some open questions
- **Update** the document, based on further discussion.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Current status

- Documentation:
  - **Received documents** from ATLAS, CTA, SKA, FAIR (CBM and PANDA)
  - Document from LOFAR is **in preparation**.
  - This document is (necessarily) an **initial version** we anticipate an updated version, based on feedback from ESFRI communities
- Preparing for DAC21:
  - Interviewing ESFRI to build plans for what we are going to demonstrate.
  - Discover the QoS needed for these demos.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Closing remarks on QoS

- This probably all sounds quite complicated, but it isn’t really.
  - The job is to **hide complexity**, to make managing storage managable.
- QoS is **baked-in** to the DataLake concept.
  - You don’t need to add anything to start using it.
- QoS is an **optional feature**.
  - You don’t need to use QoS if you want to use the Data Lake concept.

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Caching

Didn’t I leave my data
around here, somewhere?

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

What is a cache?

- **Temporary** storage of data
  - Data may be evicted at any time
- Reading data from a cache is **faster** than without a cache
- Cache contents is **unknown,** usually, except…
  - Rucio has mechanisms to “know” what is in a cache: the cache could keep Rucio up-to-date,
  - Control over which jobs use a cache → some knowledge of cache’s contents.
  - Possible to “warm” (or pre-populate) a cache, if input dataset is known.

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Why is caching important

- **Single entry point** when reading data.
  - Some facilities (typically HPC) have strict rules on network connectivity
  - Having an on-site cache provides an easy way to support data access from the data lake.
- Increase **entropy** of the network
  - If the network is a limiting factor (bandwidth, latency) then you want to get the “best” from it → maximise the information.
  - Try not asking for the same data multiple times.
- Reducing (or hiding) **latency**
  - Many applications are bad at managing their IO requests.
  - To storage, looks like: **OPEN**, **READ**, *SLEEP*, **READ**, *SLEEP*, …
  - A cache can continue downloading data while the application is processing the previous read.

Funded by the European Union’s Horizon 2020 - Grant N° 824064

Caching in ESCAPE

- A Caching “task force” established by CERN (and Riccardo, in particular), to kick-start cache deployments (LAPP, IN2P3, etc).
- So far, work has mostly been done by INFN, GSI and CERN.
- Deployments have mostly focused on xcache.
- Interest from CMS, FAIR, SKA (and likely others, too).
- More work is anticipated in developing a better understanding of the technologies and use-cases (FAIR and SKA)

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Protocol Hiding

As if by magic, the data appears!

Funded by the European Union’s Horizon 2020 - Grant N° 824064

What is protocol hiding?

- These are technologies that make it **easier** to use the data lake.
- The goal is to **reduce barriers** for users.
  - This is part of scaling-down: to support smaller development teams and (ultimately) end scientists in getting their job done.
- Updating scientific domain software to interact with Rucio and storage systems **requires effort**: can we reduce that effort?

  Direct storage interaction is already abstracted through the gfal2 library.
- Perhaps the principal example here is JupyterHub integration with Rucio (SWAN) from CERN, with RUG/FAIR working on something similar.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Possible discussion points

Blah blah blah...

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Possible discussion points: QoS

- Triggering QoS transitions when someone requests to run an analysis
  - Wait for QoS transition to complete?
  - Who is allowed to do this?
  - Do we understand for which jobs this makes sense to use faster (== more expensive) QoS?
- Writing data into the datalake
  - With which QoS should results be written?
  - Support for intermediate results?

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Possible discussion points: caching

- Cache lifetime:
  - Which model to support on-demand caching vs permanent caches?
  - Cache deployments when jobs are deployed on public/hybrid clouds.
- What size cache do we need?
  - What is the working set the cache should support?
- Do we need to limit whether jobs use a cache?
  - Can we avoid that a job scans through too many files, destroying the cache’s content?

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Possible discussion points: protocol hiding

- Providing a POSIX access?
  - A fuse-mount access that allows access to data; e.g., xrootdfs
- Updating “large” applications to support DataLake
  - JupyterHub is the obvious example
  - Which software to target?
  - How is this software deployed and configured at the target sites?
- Providing a friendlier environment

  What if a scientist isn’t using JupyterHub in the analysis; is there something to help them use the DataLake?

Funded by the European Union’s Horizon 2020 - Grant N° 824064

Final remarks

- That was a *lot* of infomation
- The take-home points are:
  - We have considerable flexibility in the Data Lake.
  - We want the Data Lake to be useful for analysis and data preparation need.
  - We need to develop some joint WP2-WP5 use-cases, so we can demonstrate this really works

Funded by the European Union's Horizon 2020 - Grant N° 824064

Thanks for listening!

Funded by the European Union's
Horizon 2020 - Grant N° 824064

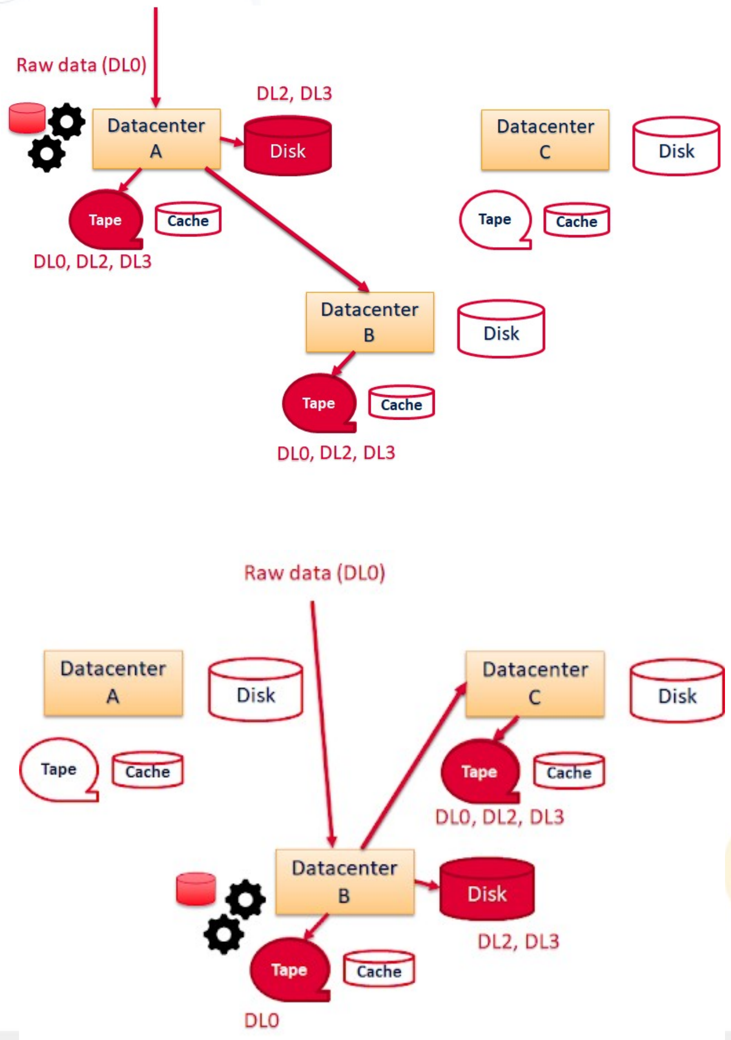
Backup slides

Funded by the European Union's
Horizon 2020 - Grant N° 824064

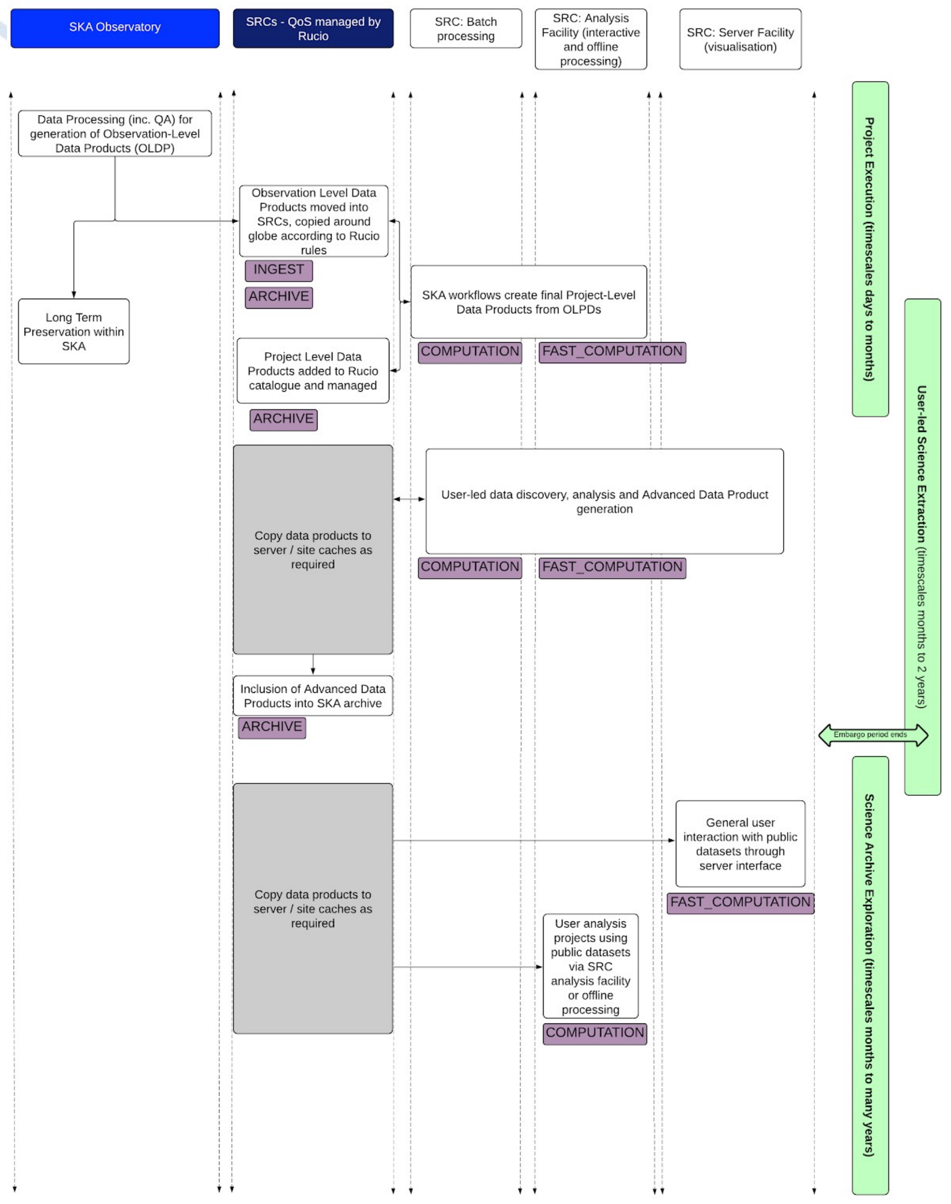
Document format

- The ESFRI-specific QoS document is split into four sections.
- The **QoS policies section** is a table, containing information about how the experiment sees QoS.
  - Name, Where it is used, Important characteristics, Example media.
- The **data life-cycle/work-flows section** is more free-form.
  - Describes the best guess at how data will be handled.
- The **interactions with Rucio** section describes how the experiment framework should achieve the desired QoS
  - List of operations to satisfy life-cycles
- The **use-cases section** provide terse description of user interactions.
  - Bring previous three items together.

Funded by the European Union's
Horizon 2020 - Grant N° 824064



CTA workflow (cred. Nadine Neyroud)



SKA Observatory
SRCs - QoS managed by Rucio
SRC: Batch processing
SRC: Analysis Facility (interactive and offline processing)
SRC: Server Facility (visualisation)
Data Processing (inc. QA) for generation of Observation-Level Data Products (OLDP)
Observation Level Data Products moved into SRCs, copied around globe according to Rucio rules
INGEST
ARCHIVE
Long Term Preservation within SKA
SKA workflows create final Project-Level Data Products from OLPDs
COMPUTATION
FAST_COMPUTATION
Project Level Data Products added to Rucio catalogue and managed
ARCHIVE
Copy data products to server / site caches as required
User-led data discovery, analysis and Advanced Data Product generation
COMPUTATION
FAST_COMPUTATION
Inclusion of Advanced Data Products into SKA archive
ARCHIVE
Copy data products to server / site caches as required
General user interaction with public datasets through server interface
FAST_COMPUTATION
User analysis projects using public datasets via SRC analysis facility or offline processing
COMPUTATION
Project Execution (timescales days to months)
User-led Science Extraction (timescales months to 2 years)
Embargo period ends
Science Archive Exploration (timescales months to many years)

SKA data lifecycle (cred. Rohini Joshi)

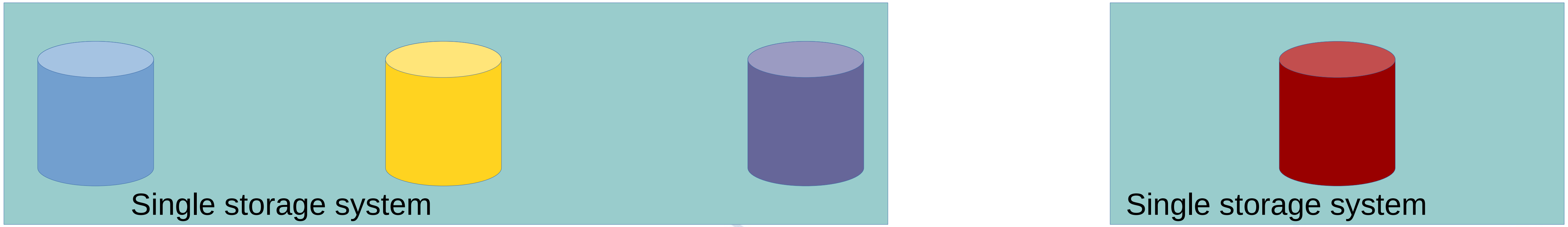
Funded by the European Union's Horizon 2020 - Grant N° 824064

Why are you using storage?

- To store my data (… well … yes, obviously)
- Some specific examples:
  - "Scratch" – to store analysis output, which can be recreated if needed
  - "Reliable" – to store important data where performance isn't an overriding goal
  - "High Performance" – for analysis in HPC /GPGPU cluster
  - "Archive" – long-term storage for unknown/possible future analysis
  - "Public" – non embargoed data that external users can access
  - "Ingest" – storage that is tuned to for high performance write operations
  - … etc …
- A single storage that supports all use-cases will be expensive.
- Storage QoS (supporting specialised storage) allows you to cut costs.

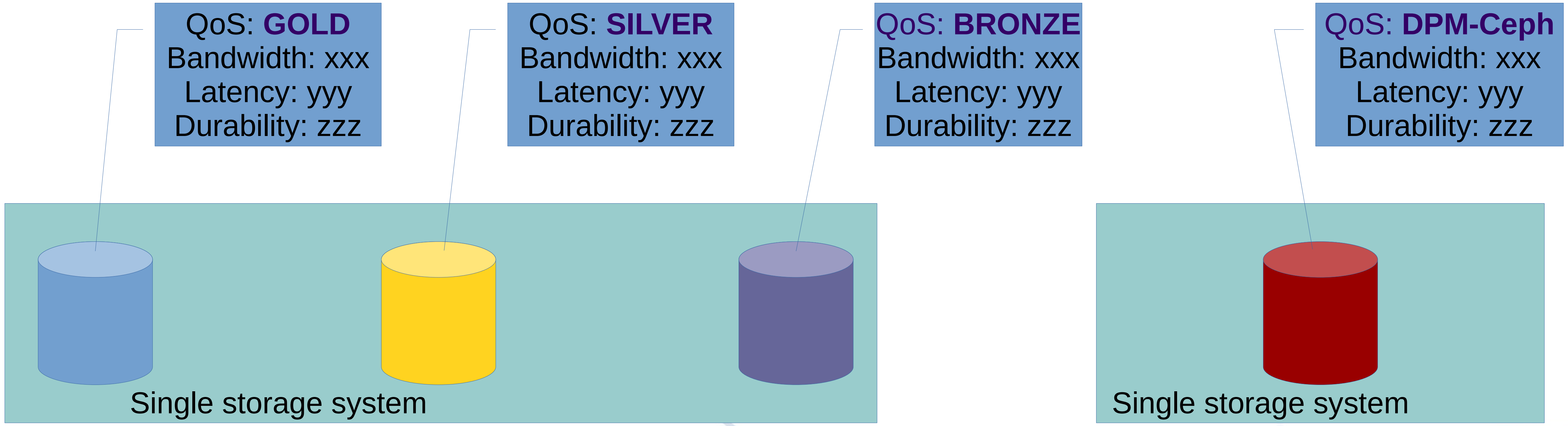
Funded by the European Union's Horizon 2020 - Grant N° 824064

Concepts: Storage QoS classes



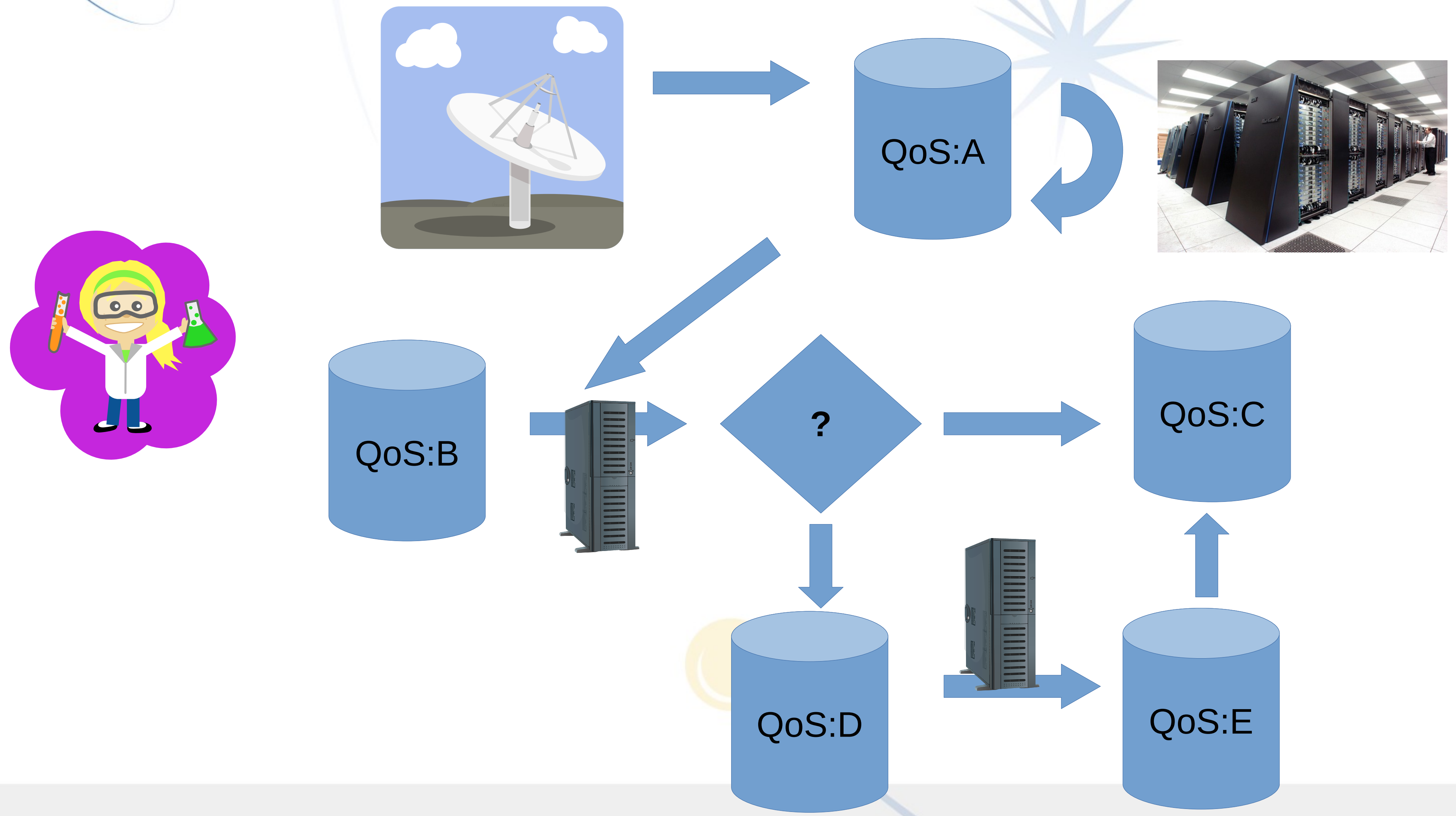
Funded by the European Union's
Horizon 2020 - Grant N° 824064

Concepts: Storage QoS classes



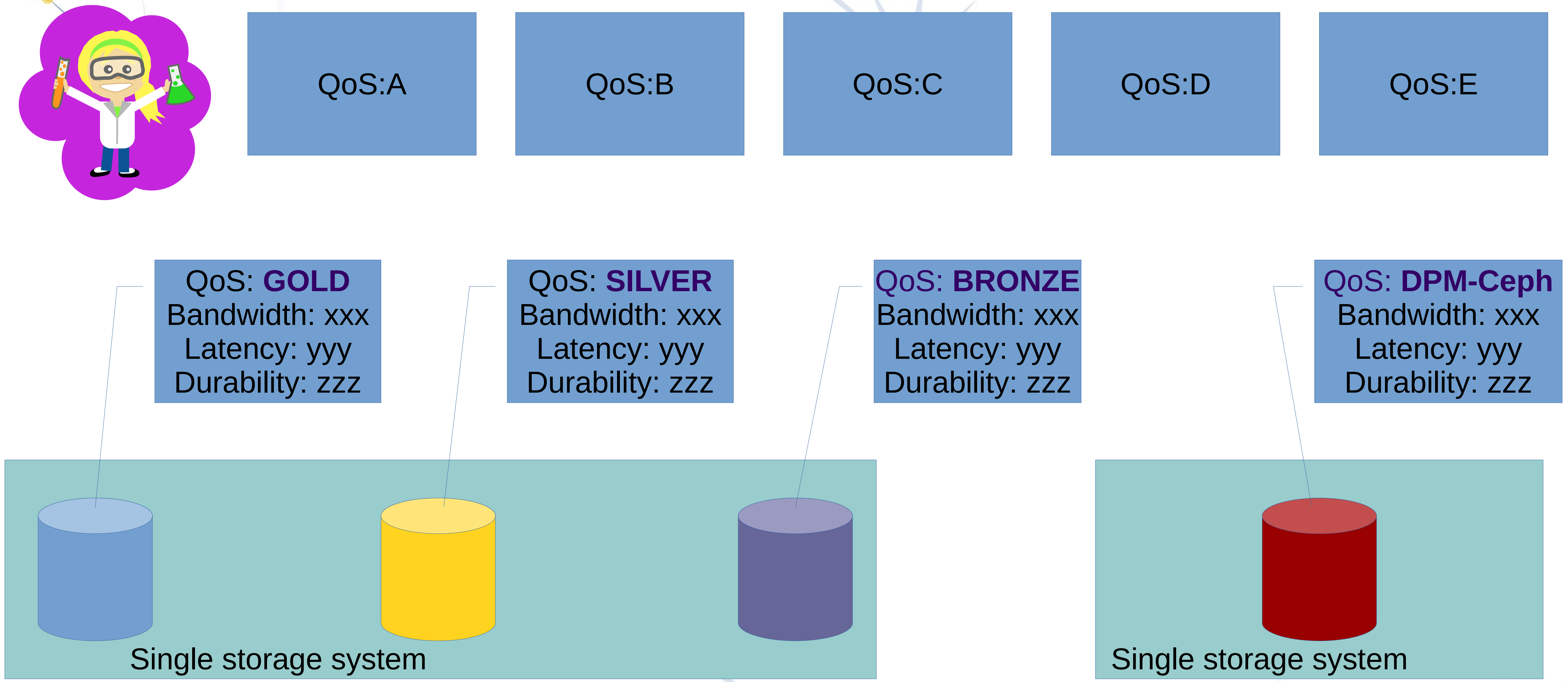
Funded by the European Union's
Horizon 2020 - Grant N° 824064

Concepts: work-flow / data lifecycle



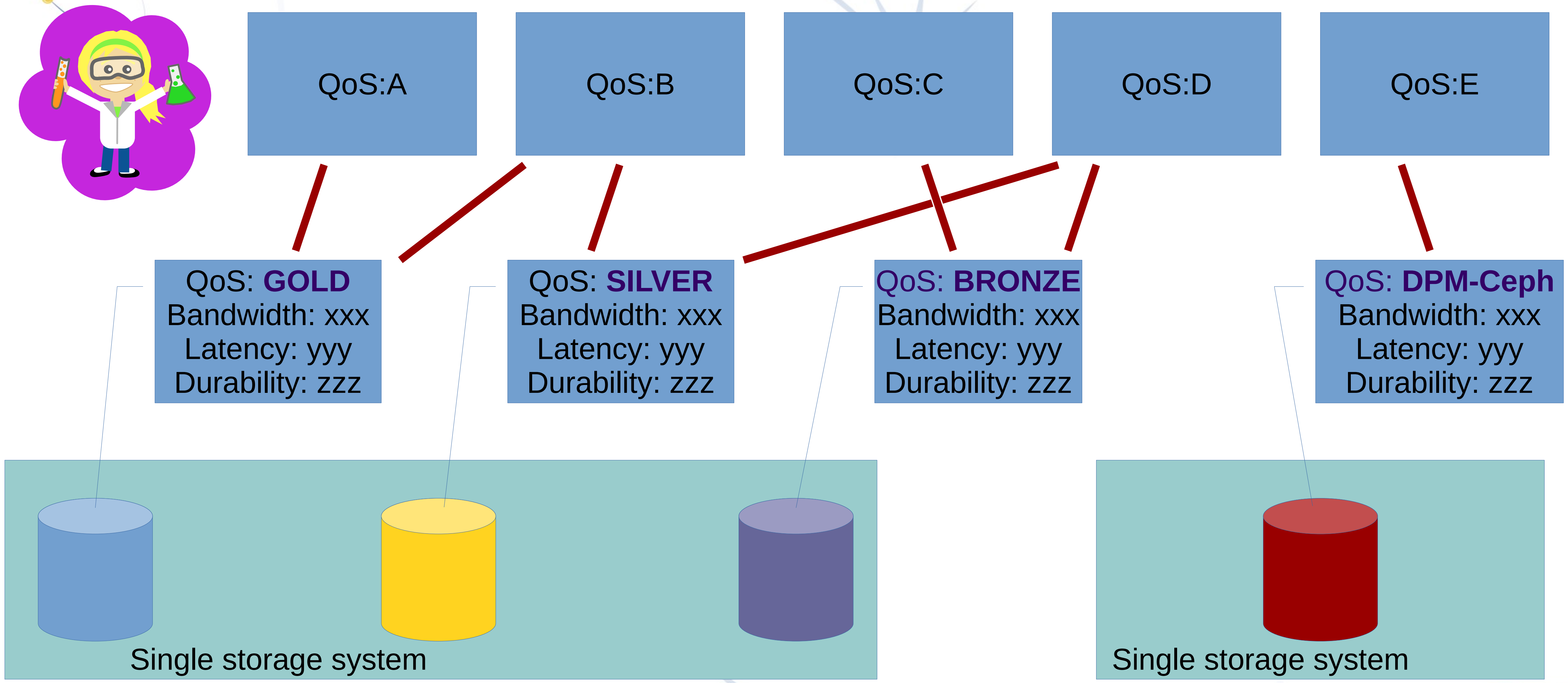
Funded by the European Union's
Horizon 2020 - Grant N° 824064

Concepts: VO-QoS-Policies



Funded by the European Union's Horizon 2020 - Grant N° 824064

Concepts: VO-QoS-Policies



Funded by the European Union's
Horizon 2020 - Grant N° 824064

Storage endpoint deployment

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Testbed: the storage QoS classes

- Currently, we have **storage endpoints** from (in no particular order):
  - CERN, SurfSARA, CNAF, Napoli, Roma, IN2P3, PIC, LAPP, GSI and DESY.
- There is a **wiki page** that documented what storage technology is being used.
- These are grouped together to build four **storage QoS Classes**:
  - SAFE (e.g. tape),
  - FAST (e.g., SSD),
  - CHEAP-ANALYSIS (e.g., EC or RAID-6),
  - OPPORTUNISTIC (e.g. JBOD).
- Four is a **balance** between
  - too few → difficult to make specific requirements; each class is too general.
  - too many → difficult to work with; each class is not general enough.

Funded by the European Union's
Horizon 2020 - Grant N° 824064

The “Aleem” QoS demo

- Answering the **question**: what can we do with what we have now?
- How it **works**:
  - In Rucio, label RSE endpoint, using the attribute **`QoS=<storage-class>`**
  - When creating rules, use this attribute to describe desired storage QoS
    - E.g., **`(QoS==JBOD || QoS== RAID) && Site=IN2P3`**
  - Rucio takes care to copy data to specific locations.
  - QoS transitions involve adding or removing rules.
- **Positive**: we can do this right now
- **Negative**: we lose the VO-QoS policy level abstraction
  - This makes the system fragile; changes become difficult.
- Demoed with LOFAR data, ESFRI representatives are replicating this.

Funded by the European Union’s
Horizon 2020 - Grant N° 824064