QoS and DLC in IaaS
INDIGO-DataCloud

Presenter : Patrick Fuhrmann
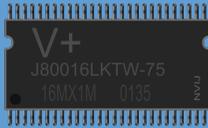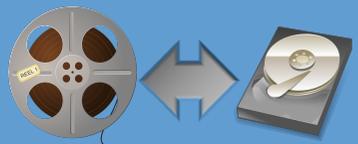
Contributions by:

Giacinto Donvito, INFN
Marcus Hardt, KIT
Paul Millar, DESY
Alvaro Garcia, CSIC
Zdenek Sustr, CESNET

And many more

INDIGO - DataCloud

Quality of Service based on media

Media Quality					
Access Latency	HIGH	MEDIUM	LOW	MEDIUM	MEDIUM
Durability	OK	MEDIUM	Not so clear	Quite OK	OK
Datarate	OK	OK	MEDIUM	OK	OK
Cost	Very low	Reasonable	Very high	MEDIUM	MEDIUM

Not quite as easy as that

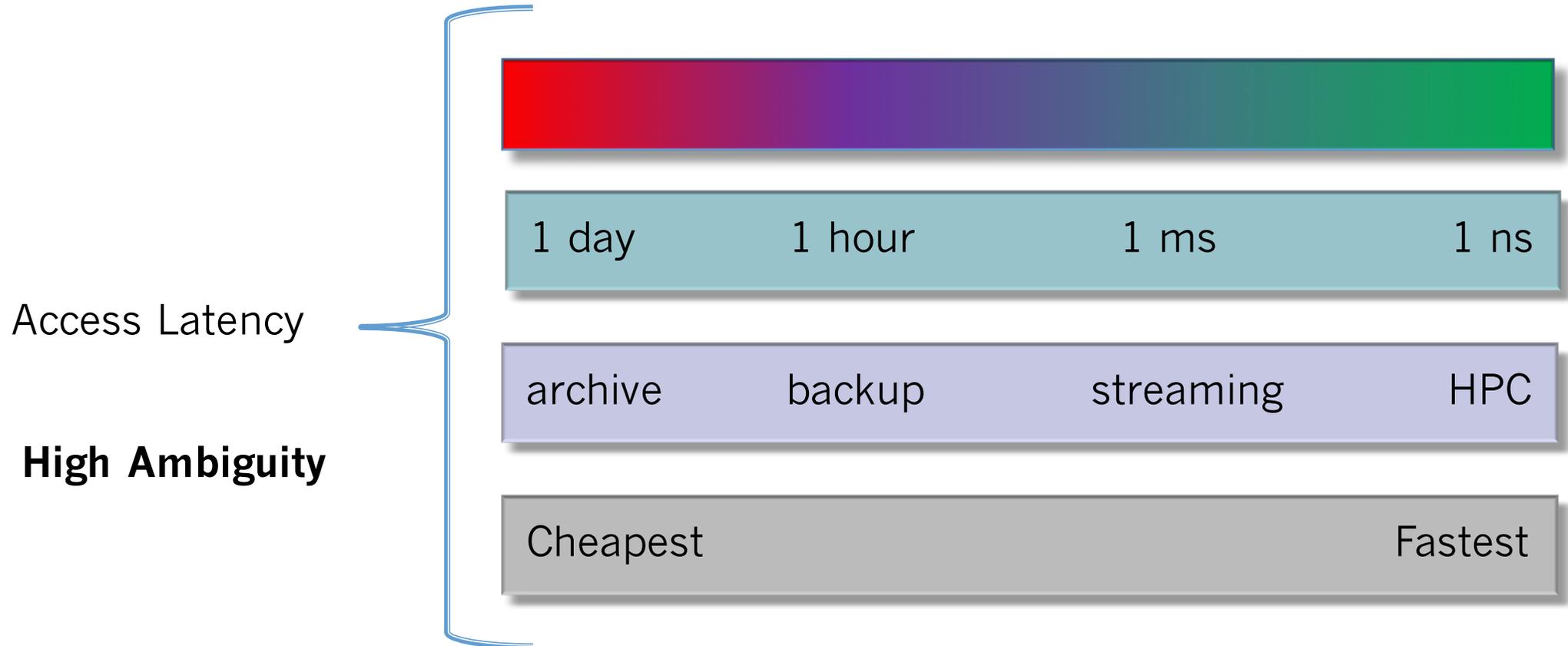It looks simple, but there
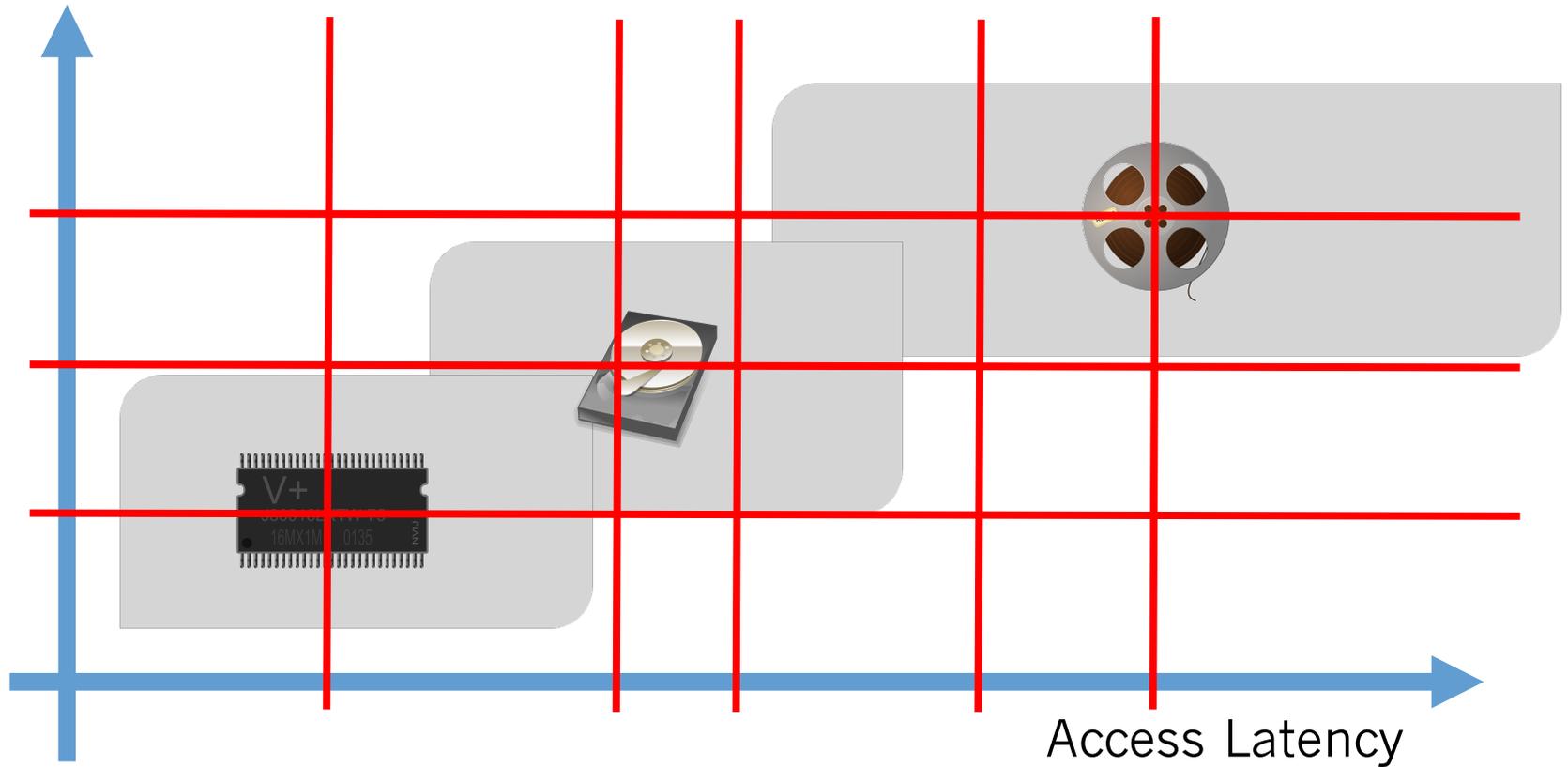are issues.

The QoS **properties**

➢ Is there a sufficiently complete set of properties ?

➢ In WCLG we only had two properties :

  ➢ Access Latency

  ➢ Retention policy

➢ That was already too much for most people ☺

➢ Talking to Reagan Moore (IRODS) at the Paris RDA meeting:

  ➢ He is suggesting about 200 properties

  ➢ That might be a bit over the top for a start

QoS Property **Value Ambiguity**

Access Latency

High Ambiguity

1 day	1 hour	1 ms	1 ns
archive	backup	streaming	HPC
Cheapest			Fastest

Property **dependencies**

Durability

Access Latency

Property **Quantization**

Multi Dimensional Property Quantization
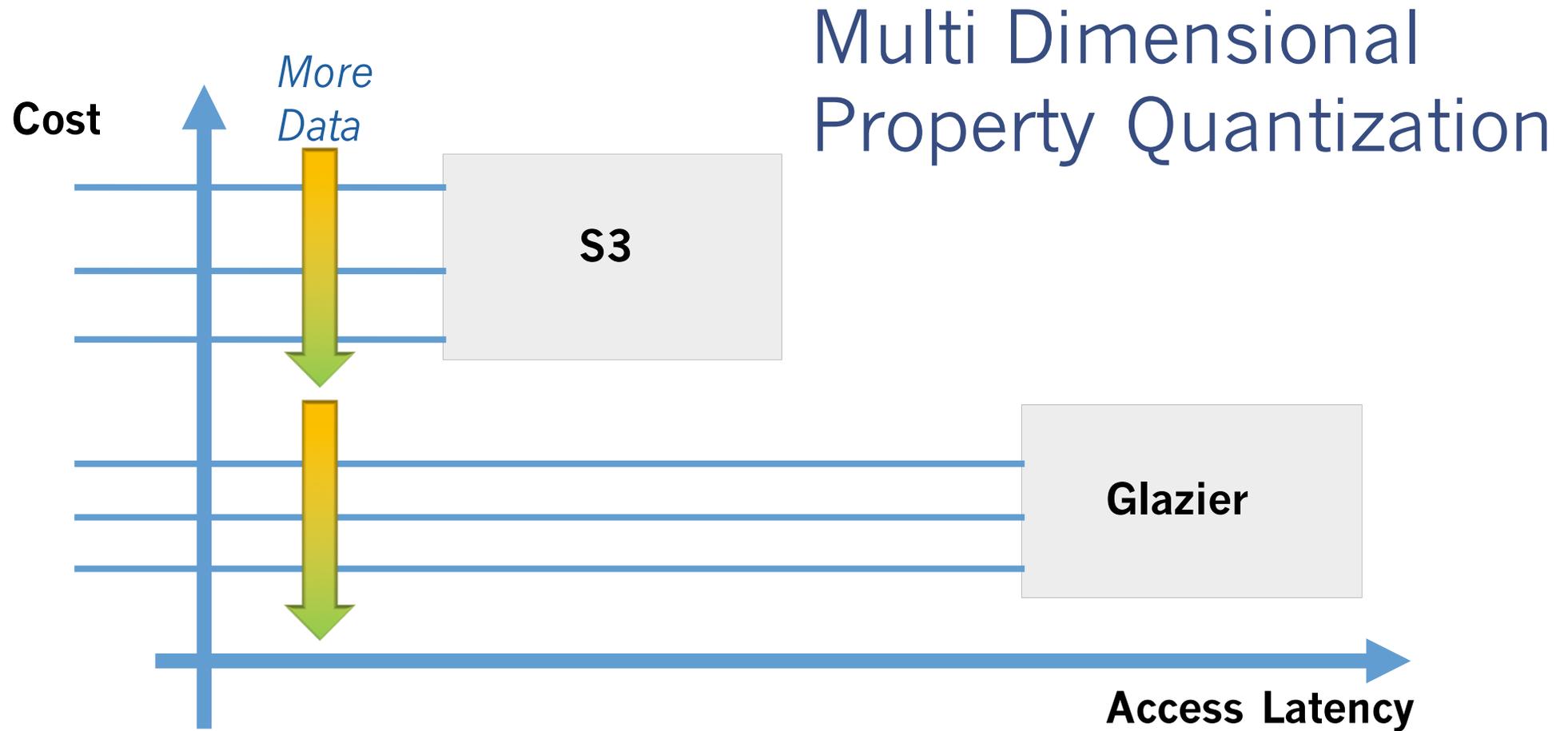
Properties zoo of existing systems

Amazon	S3		Glacier
Google	Standard	Durable Reduces Availability	Nearline
HPSS/GPSS	Corresponds	to the HPSS Classes	(customizable)
dCache	Resilient	disk+tape	TAPE

Time to tidy up !

Starting with the unambiguous technical view, seen by the storage system.

Canonical Properties

What are canonical properties ?

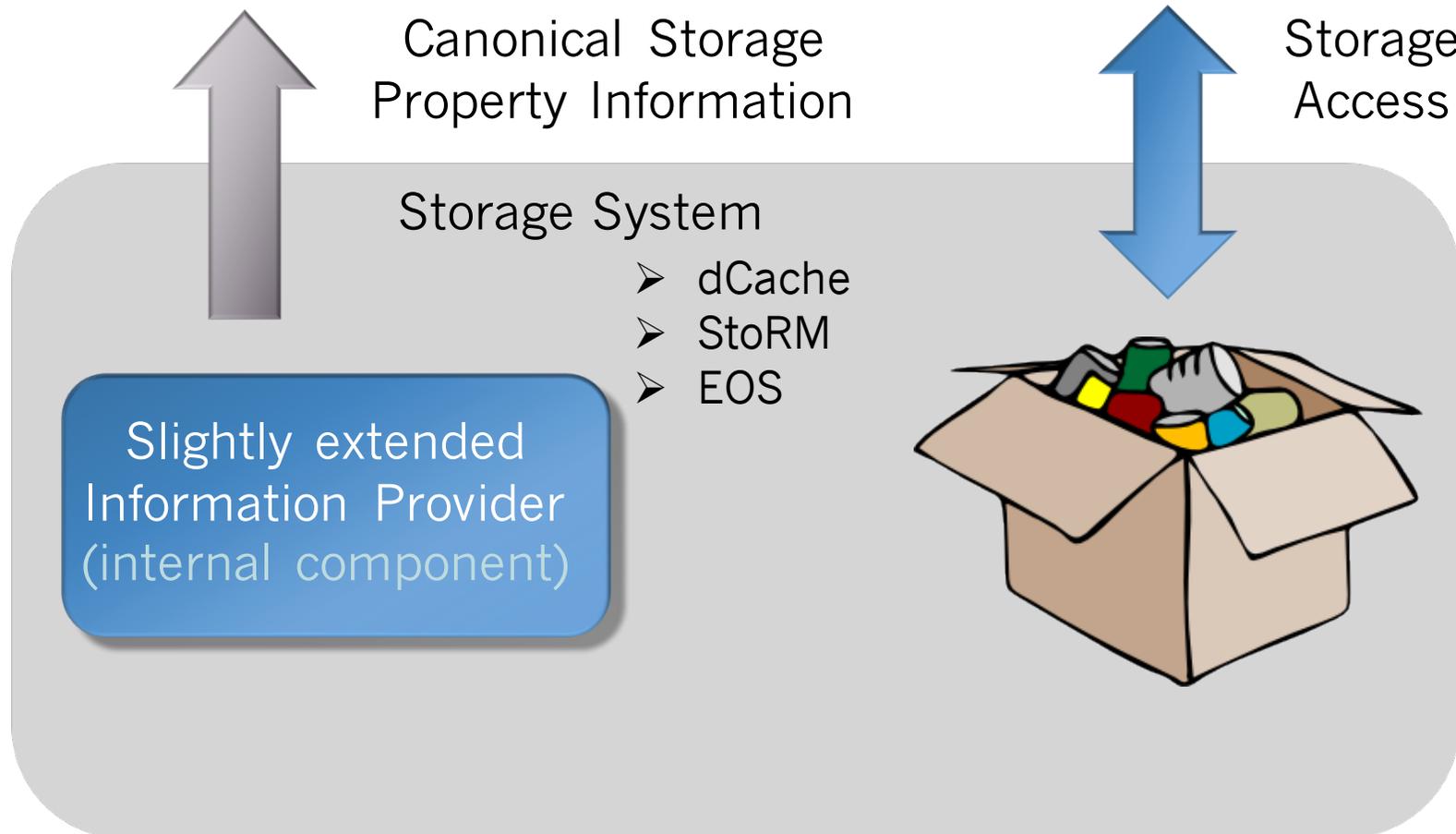|	Class A	Class B	Class C
Access Latency	< 1 ms		< 10 min
Durability	> 0.9999	******	0.99999999
Media	Disk / SSD		Tape
Replicas	1 Disk		2 Tape

Avoiding ambiguities

How to get ...

How to get those properties out of
exiting storage systems ?

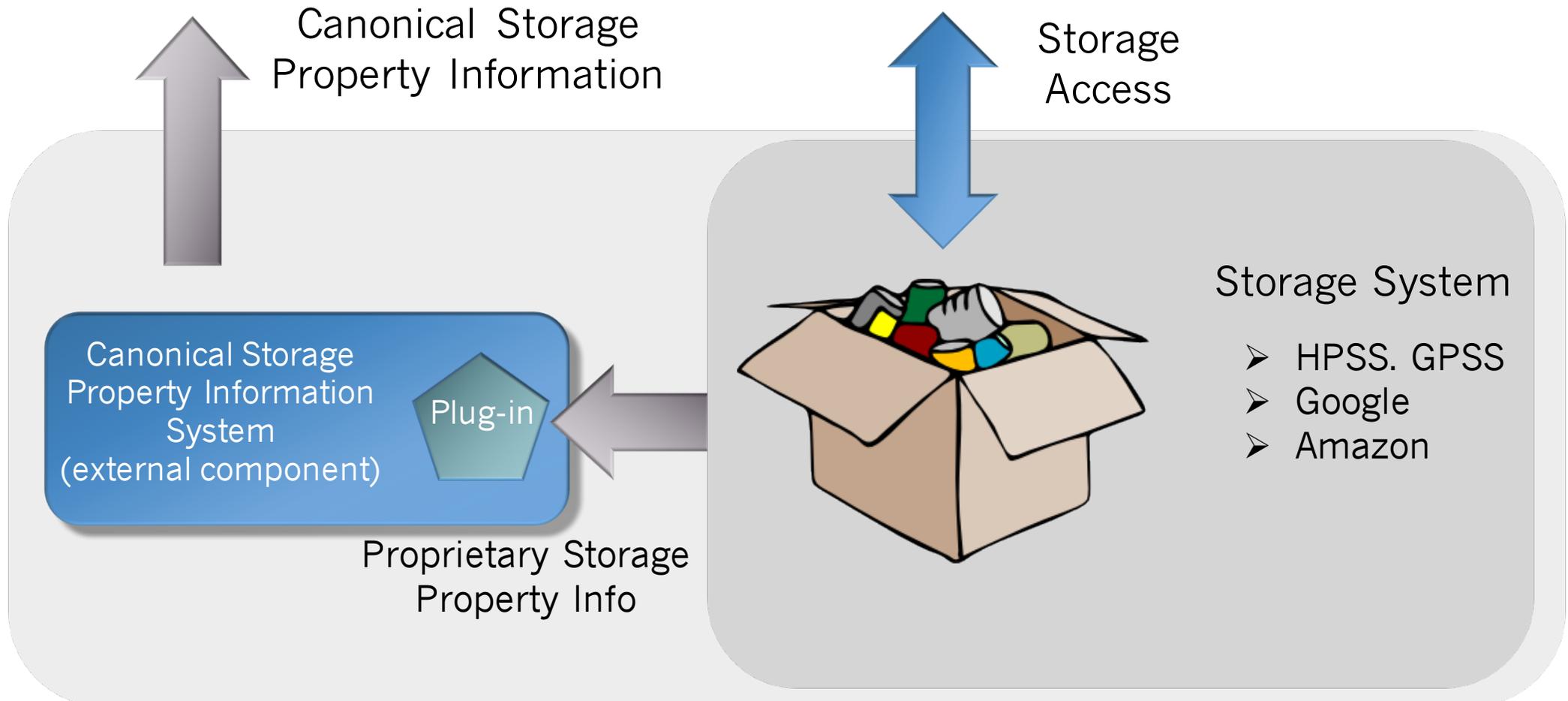# Canonical Storage Properties

Canonical Storage
Property Information

Storage
Access

Storage System

- ➢ dCache
- ➢ StoRM
- ➢ EOS

**Slightly extended
Information Provider
(internal component)**

Canonical Storage Properties

Canonical Storage
Property Information

Storage
Access

Canonical Storage
Property Information
System
(external component)

Plug-in

Storage System

- HPSS. GPSS
- Google
- Amazon

Proprietary Storage
Property Info

The **canonical view** only helps to describe the system on the **technical level.**

It's not very helpful for the storage user.

We need to introduce more convenient **QoS views**.

QoS views

Ambiguous, non canonical, dependent, combined properties.

Examples :

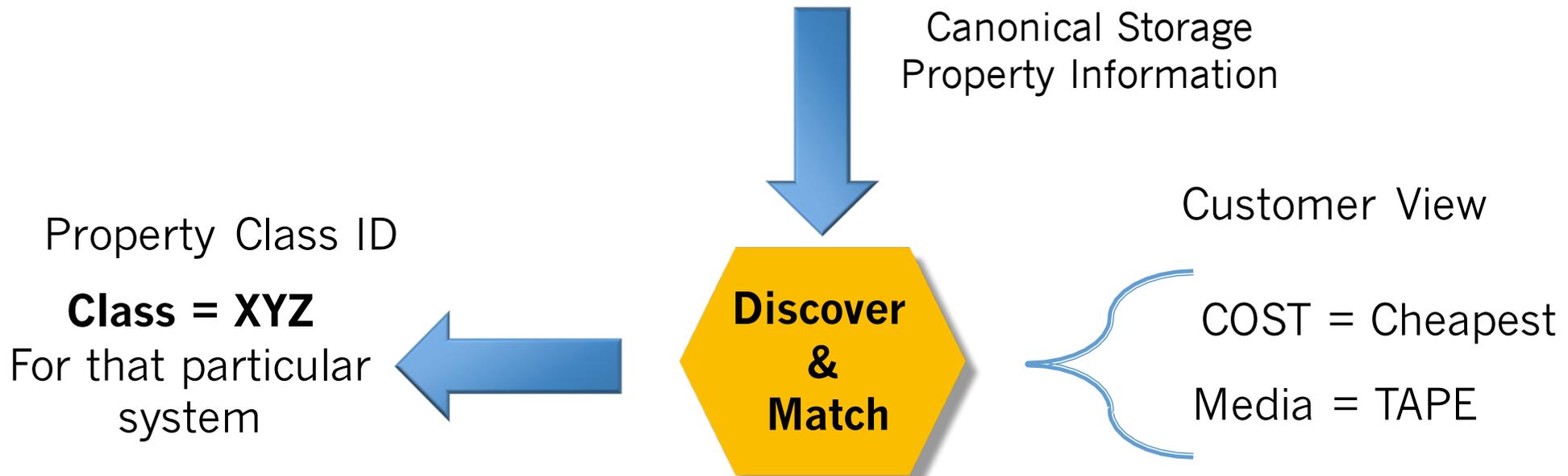**Low latency & lowest price**

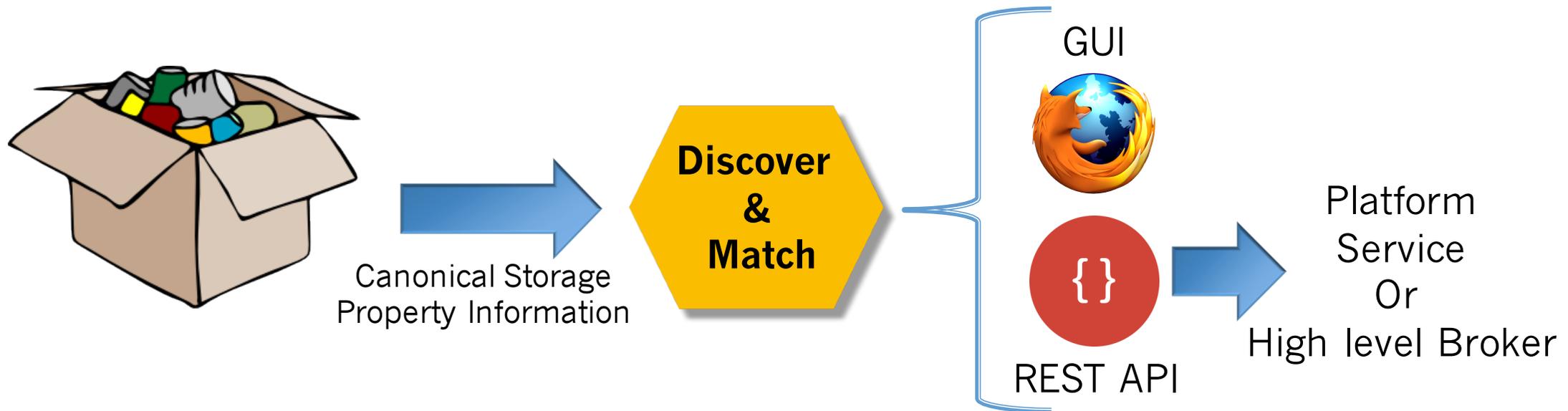**High throughput & super durable**

Large volume & cheap & archive

Discover and Match

Therefore: Introducing a new service

Canonical Storage
Property Information

Customer View

Property Class ID

Class = XYZ
For that particular
system

**Discover
&
Match**

COST = Cheapest

Media = TAPE

Translation and discovery

Canonical Storage
Property Information

Discover & Match

GUI

REST API

Platform
Service
Or
High level Broker

Canonical property federation

INDIGO - DataCloud

Platform as a Service

Canonical Storage Property Information System

D&M

GUI

{ } REST API

IaaS

D&M

D&M

GUI

{ } REST API
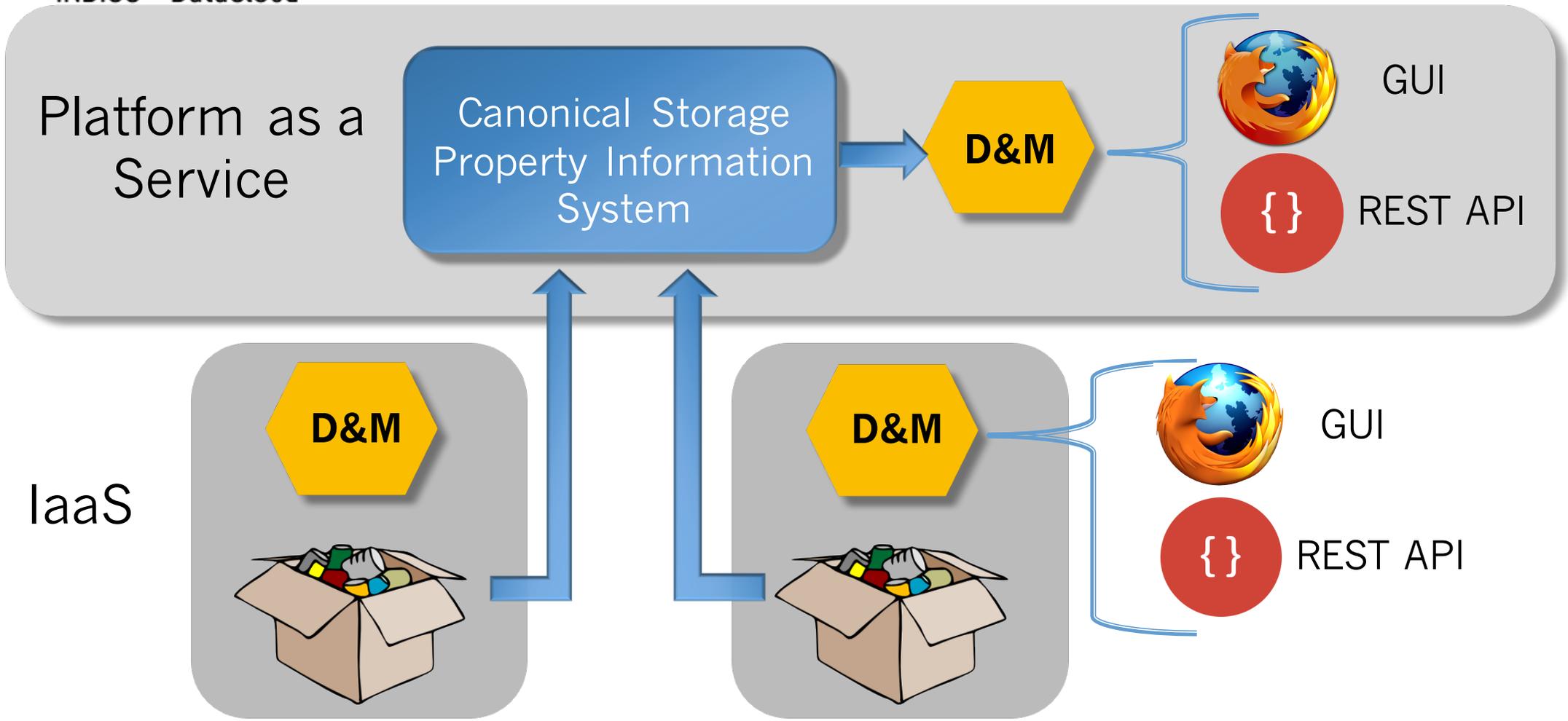
Federated Systems

- The federated system provides additional QoS properties.

  - Number of copies, not in the same location

  - Minimum geographic distance for disaster cases. (fire, earthquakes)

  - Legal implications : Privacy laws

- Federated system might need more higher level services attached:

  - FTS or Globus Online to create replicas

  - DynaFed to federate distributed resources.

More problems to solve

- How does the client provide the storage class to the storage system ?
  - Bucket
  - Directory
  - Additional argument in WebDAV, FTP etc
- The system only provides the class, it doesn't 'promise' the space.
  - Do we need a space reservation protocol ?
  - Similar to hotels.com. Check hotel pictures first, reservation only after payment.
  - Is reservation required in systems with unlimited space (Clouds) ?
- Do we allow to change the storage class, assuming the system will do the necessary data movements ?
  - This is of course just a storage system property.
    - Amazon and Goolge don't
    - dCache and HPSS do.

Next step : Data Life Cycle

- Data Life Cycle is just the time dependent change of
  - Storage Quality of Service
  - Ownership and Access Control (PI Owned, no access, Site Owned, Public access)
  - Payment model : Pay as you go ; Pay in advance for rest of lifetime.
  - Maybe other things

6 m 1 years 10 years
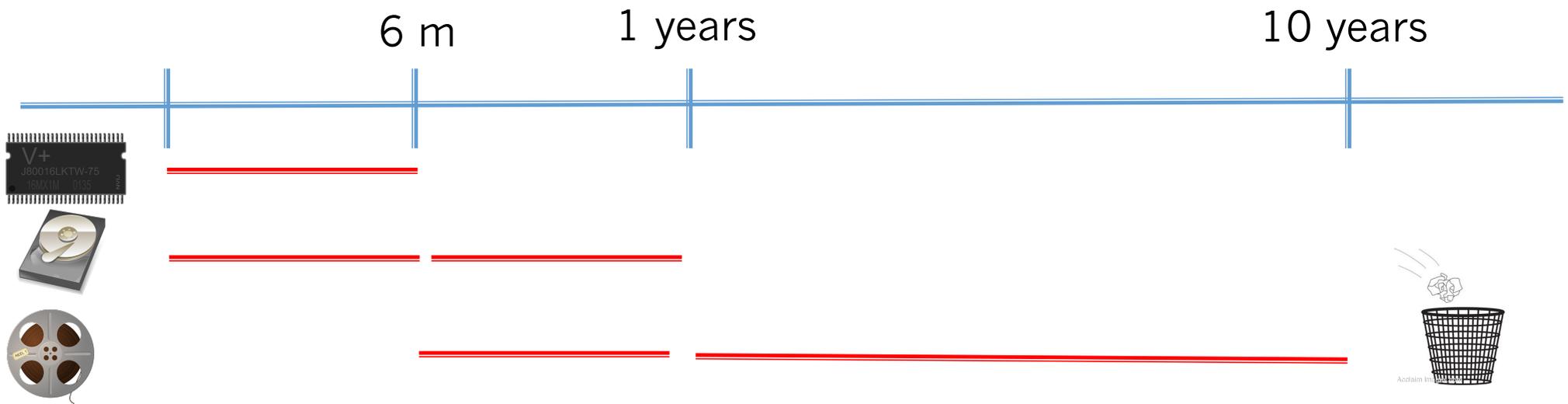
Current status (definitions)

- Introduced at the research data alliance (RDA) in Paris

- Lots of interested communities and sites.

- Creating of interest group in progress.

  - Name still in heavy discussion ☺

- 10 Committed members

- Will be followed up on in Tokyo end of Feb 2016

Current status (technically)

- Canonical Information providers are being build
  - dCache (internal)
  - Common external system for
    - GPFS/HPSS
    - CEPH
    - StoRM/GEMSS
    - Cloud (Amazon and Google)
- Information Provider Protocol in discussion (candidate : CDMI)

Summary

- INDIGO provides funding to standardize QoS and possibly Data Life Cycle of systems
- Scientific communities are showing great interest in those activities.
- Common definition of QoS is essential for Platform as a Service for storage.
- RDA 'Interest Group' being built to get in touch with more communities.
- Prototype systems, including
  - IBM : GPSS, HPSS
  - Grid storage systems: dCache, StoRM, …
  - Public Clouds: Amazon, Google
- Prototypes will be provided within the next 12 months.
- Contribution of ideas from your side is more than welcome.
- Contact : Dr. Paul Millar