



INDIGO - DataCloud

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

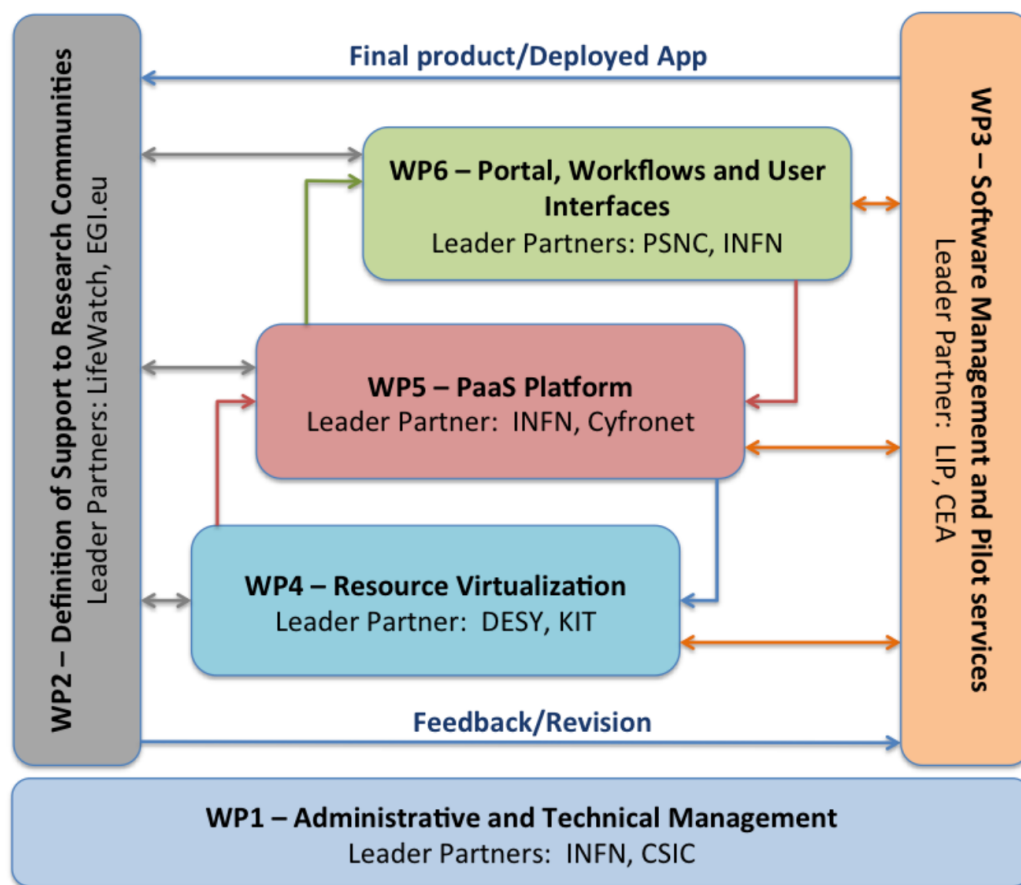
INDIGO DataCloud Cheat Sheet

■ H2020 Project

- Approved Jan 2015
- Started April 2015 – Ends Sep 2017= 30 months
- 26 European Partners
- 11 European Countries
- > 11 Million Euros
- Objective : Develop an Open Source platform for computing and data, deployable on public and private cloud infrastructures.
- Requirements and use-cases collected from 11 INIDIGO communities.
- For further details : <http://indigo-datacloud.eu>



INDIGO DataCloud WP structure



| | |
|------------|---|
| WP1 | Management |
| WP2 | Community requirements |
| WP3 | ➤ Software Management ➤ Pilot Services |
| WP4 | IaaS, Resource Virtualization |
| WP5 | PaaS, Platform |
| WP6 | Portals and user access |

Stolen from Alvaro's, Andrea's presentation



INDIGO - DataCloud

WP4 in detail

- Virtualized Computing Resources
 - Full Container support for Cloud Management Infrastructures and Batch
 - Container support for special hardware (Infiniband, GP-GPU's)
 - Spot Instances
 - Fair Share Scheduling
- Virtualized Storage Resources
 - QoS and Data Life Cycle for storage (storage management)
 - Access to data by meta data instead of name space
 - Dual access to data (Object Store versus POSIX file name space)
 - Identity Harmonization for storage
- Virtualized Network Resources
 - Orchestrating local and federated network resources
 - “Software Defined Network” evaluation
 - Services and Appliances for for virtual networks



INDIGO - DataCloud

Why QoS and DLC

- EU requires to provide a “Data Management Plan” from all data intensive EU projects.
- Problem :
 - No common way to describe QoS or Data Life Cycle
 - No common way to negotiate QoS with storage endpoints (except for SRM systems 😊)
- Common definitions for QoS would be very convenient in general but inevitable for PaaS layers, as the negotiation resp. brokering is done by engines. (Similar to hotel or flight finders)

Description of Work for WP4

1. Define a common vocabulary for QoS storage properties and their values based on use cases from scientific communities :
 - Involve standardization bodies, e.g. RDA, OGF
2. Define a semantics to negotiate QoS with endpoints
3. Find a real network protocol (prototype or demonstrator) and implement the defined QoS semantics for different systems.



INDIGO - DataCloud



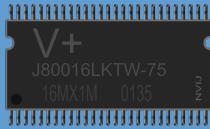

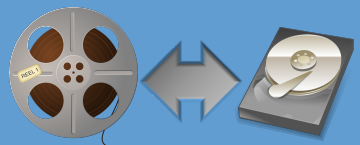
Independent developments

- Despite the fact that QoS definitions are required for INDIGO DataCloud
- Considerations in WLCG to provide a platform layer, partially replacing common parts in experiment frameworks.
 - This layer needs to query storage endpoints for their properties.
 - Needs to enforce requested QoS values by finding the right endpoint and possibly replicate data between different endpoints.
- Yemi's presentation from Monday on SLAC storage services and pay as you go, requires a portfolio of storage classes with different QoS properties (and a pricetag)



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.

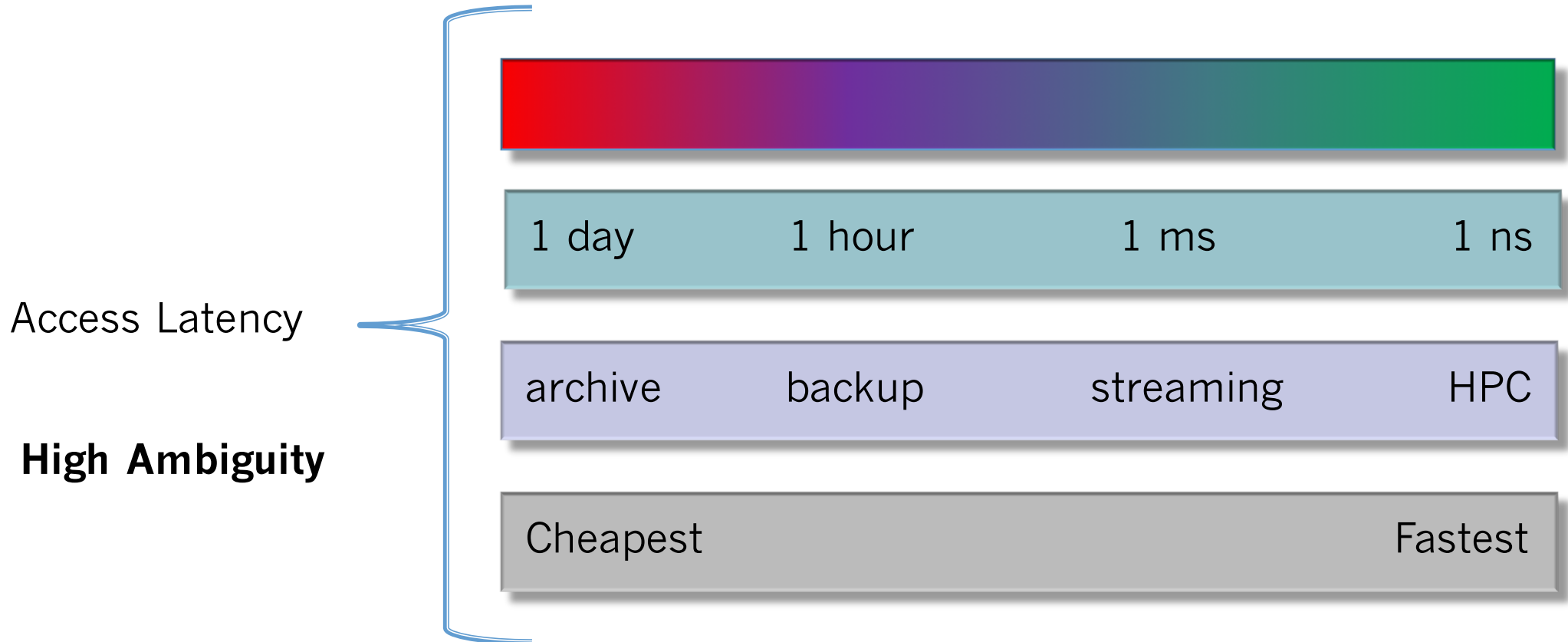


INDIGO - DataCloud

The QoS **properties**

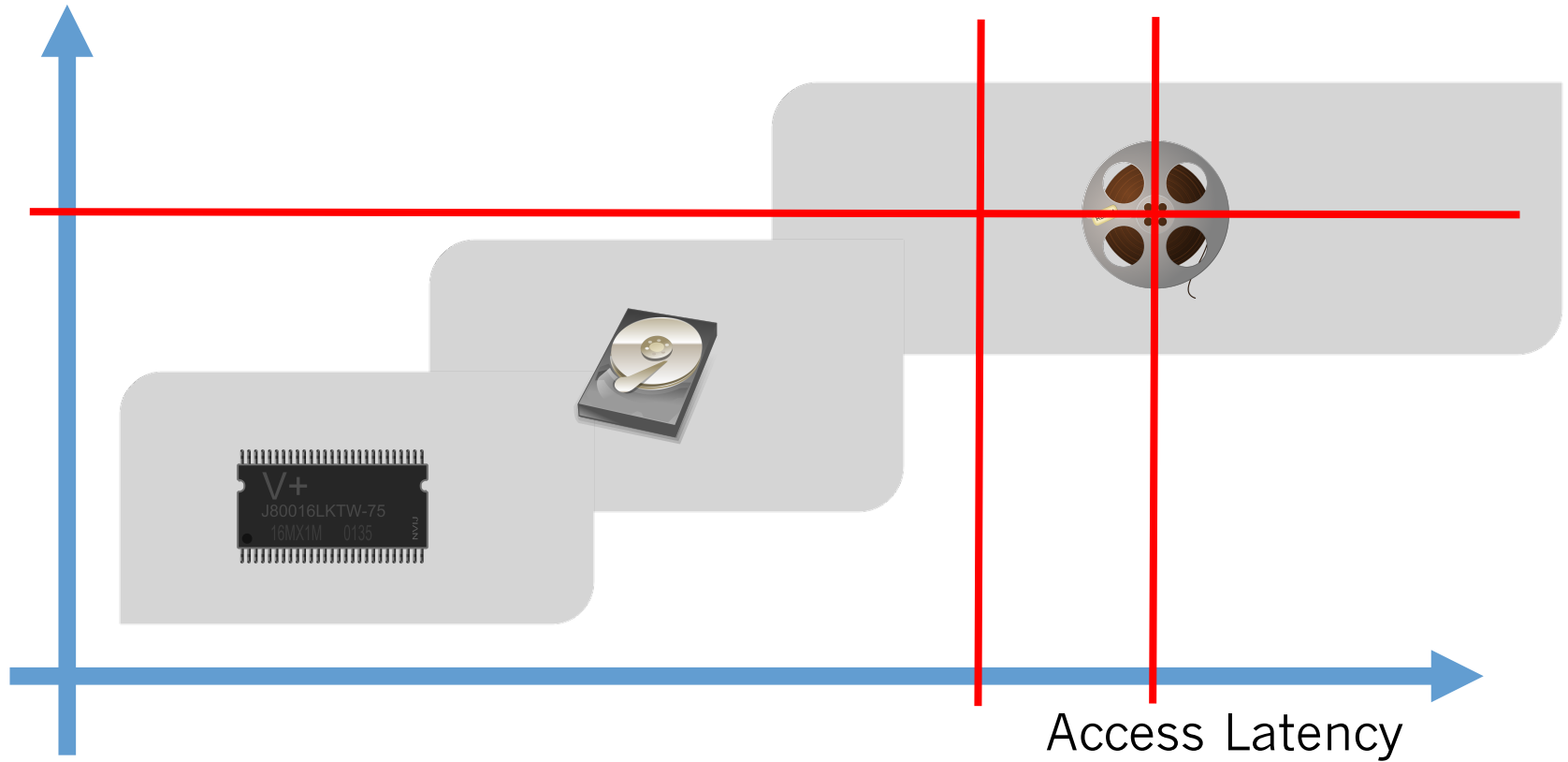
- Is there a sufficiently complete set of properties ?
- In WCLG we only had two properties :
 - Access Latency
 - Retention policy
- That was already too much for most people 😊
- Talking to Reagan Moore (IRODS) at the Paris RDA meeting:
 - He is suggesting about 200 properties
 - That might be a bit over the top for a start

QoS Property Value Ambiguity



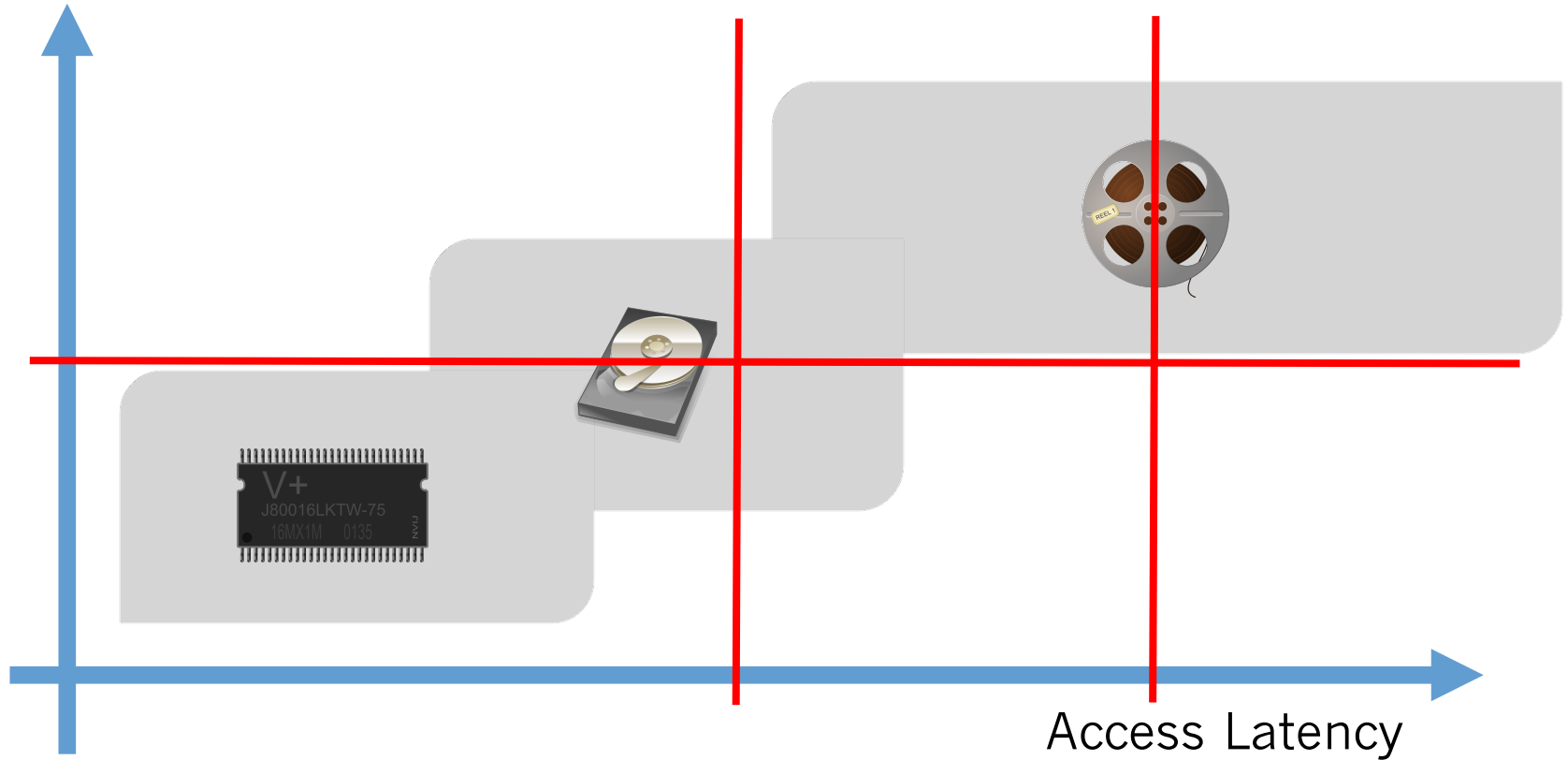
Property dependencies

Durability



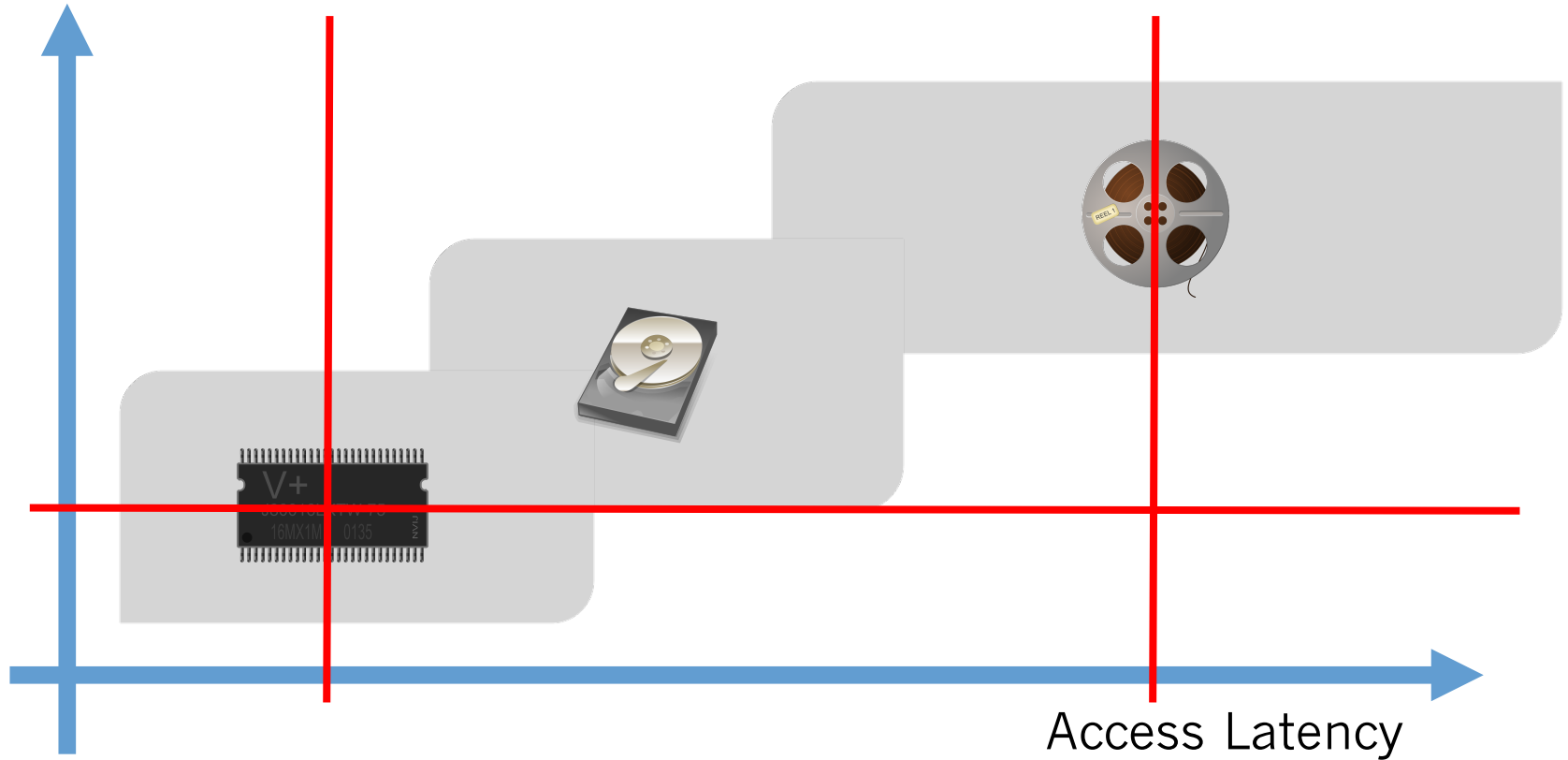
Property dependencies

Durability



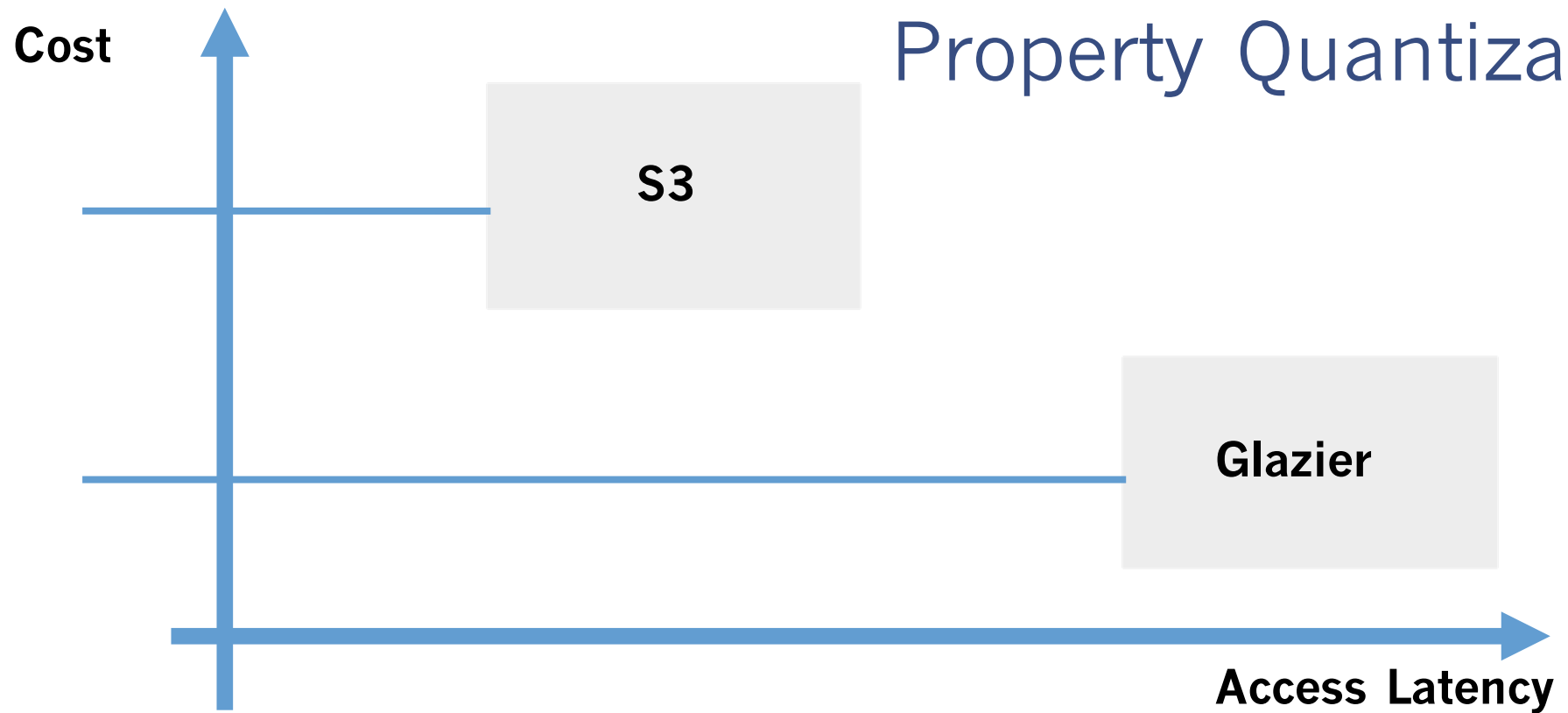
Property dependencies

Durability

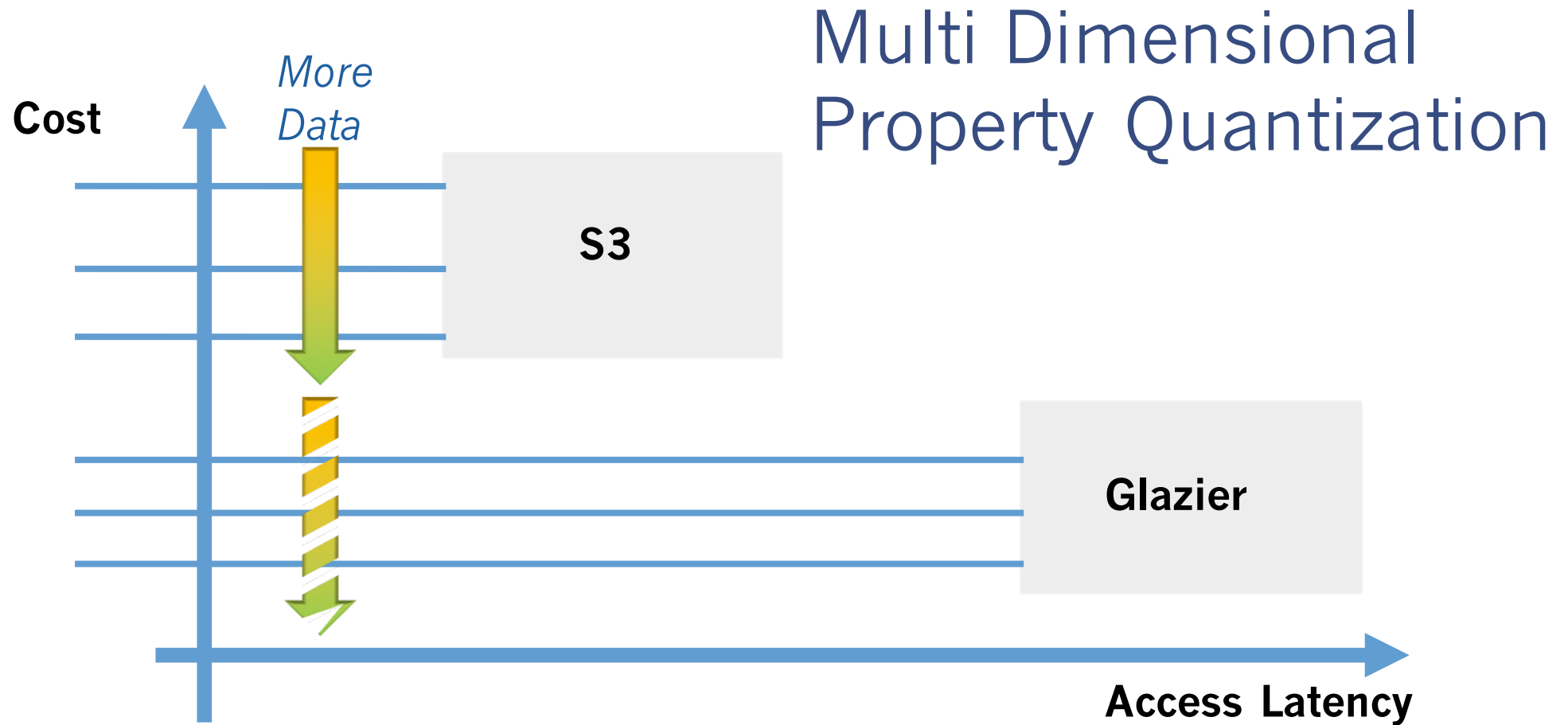


Property Quantization

Multi Dimensional Property Quantization



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 | $< 1 \text{ ms}$ | | $< 10 \text{ min}$ |
| Durability | > 0.9999 | | 0.999999999 |
| Media | Disk / SSD | ***** | Tape |
| Replicas | 1 Disk | | 2 Tape |

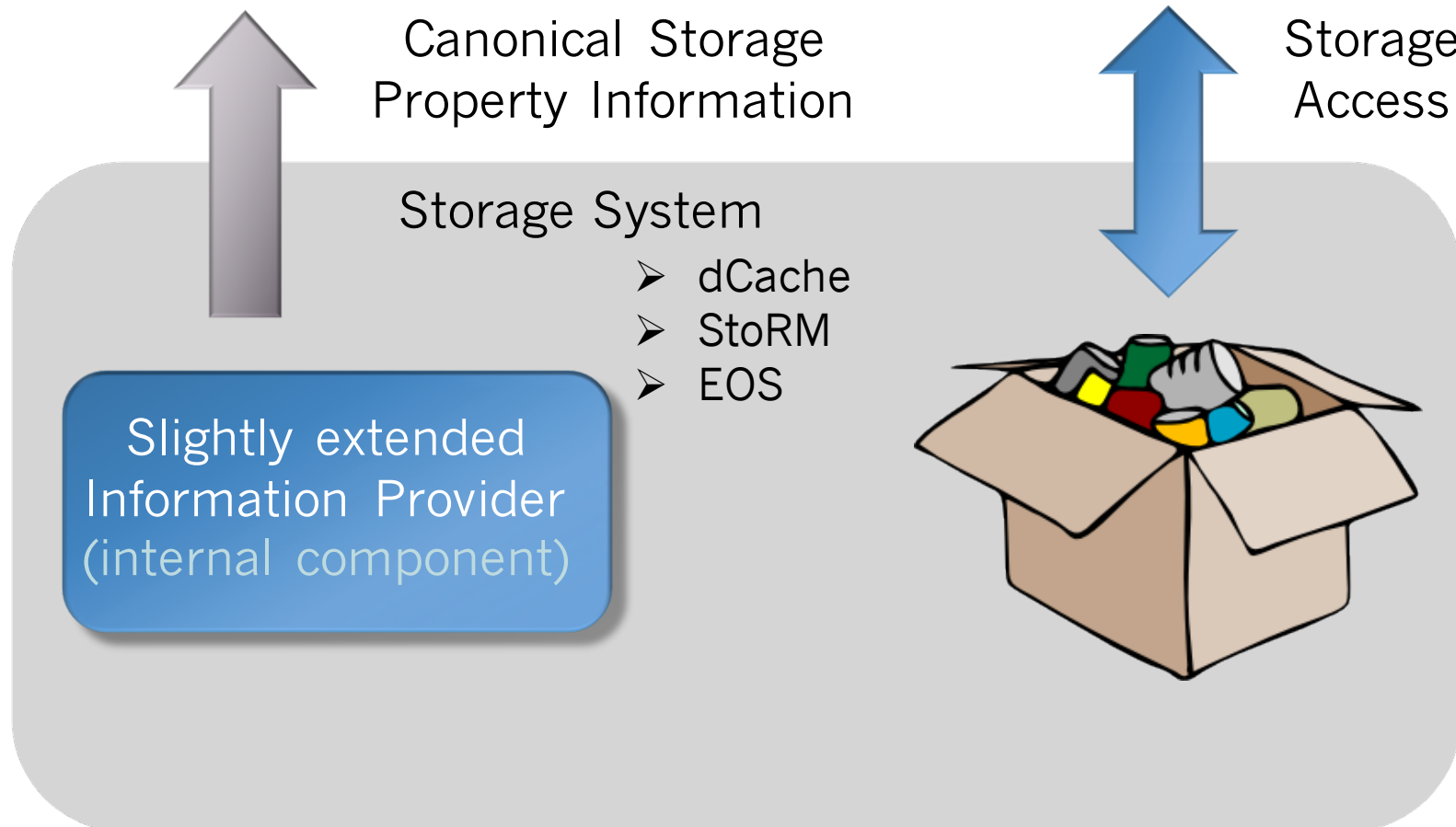
Avoiding ambiguities



How to get ...

How to get those properties out of exiting storage systems ?

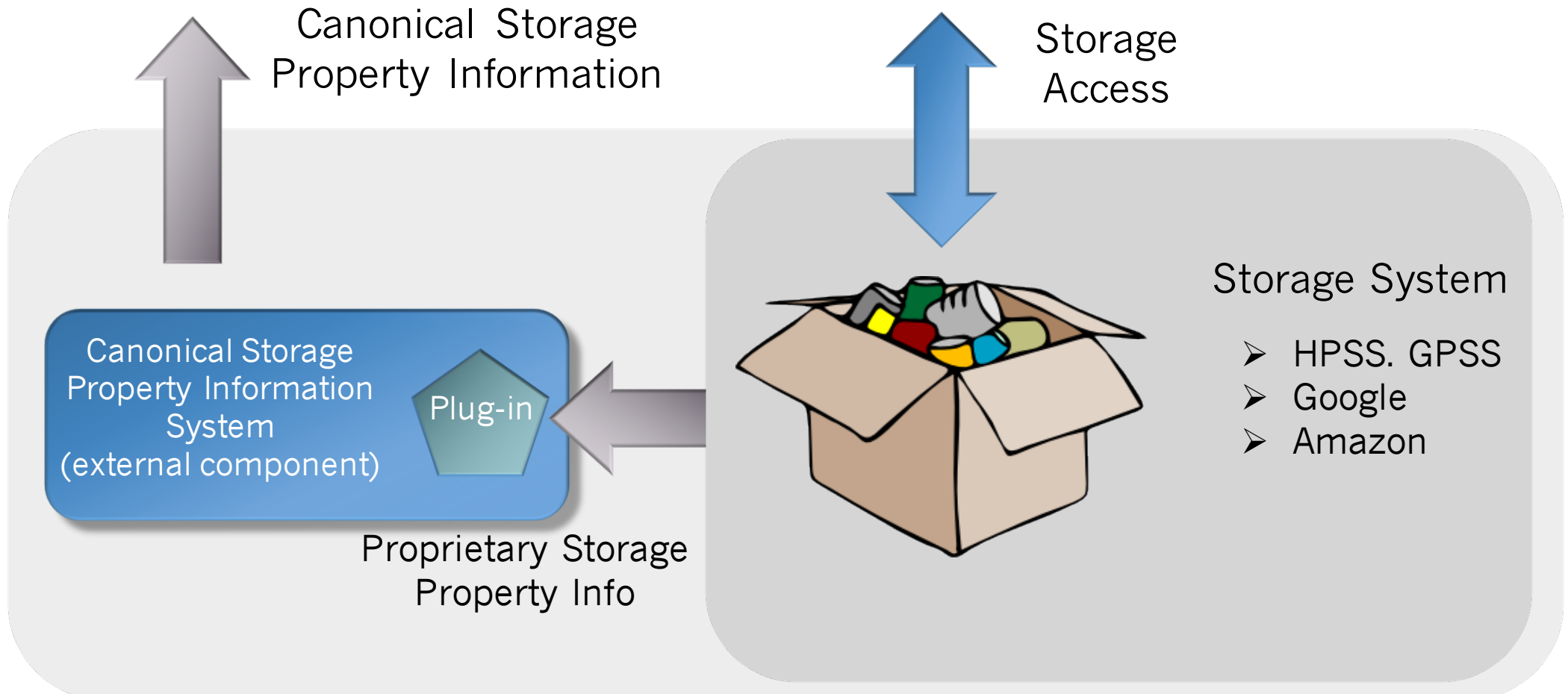
Canonical Storage Properties





INDIGO - DataCloud

Canonical Storage Properties



What are Canonical Properties ?

| | Class A | Class B | Class C |
|----------------|------------|---------|-------------|
| Access Latency | < 1 ms | | < 10 min |
| Durability | > 0.9999 | | 0.999999999 |
| Media | Disk / SSD | ***** | Tape |
| Replicas | 1 Disk | | 2 Tape |



INDIGO - DataCloud

Customer View

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**.



INDIGO - DataCloud

QoS views

Ambiguous, non canonical, dependent,
combined properties.

Examples :

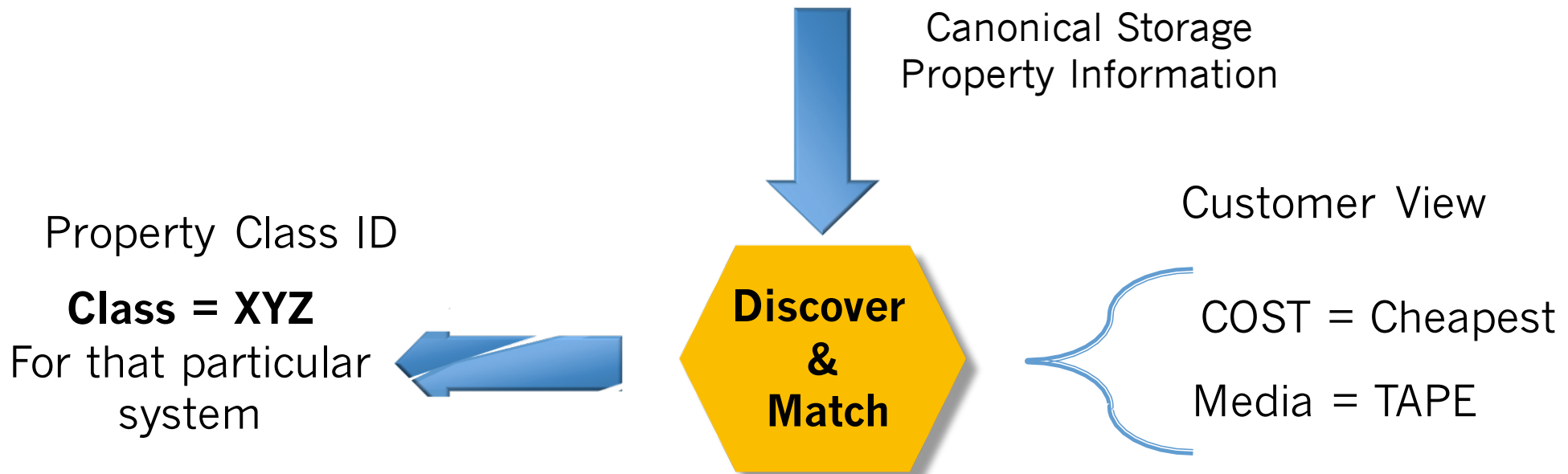
Low latency & lowest price

High throughput & super durable

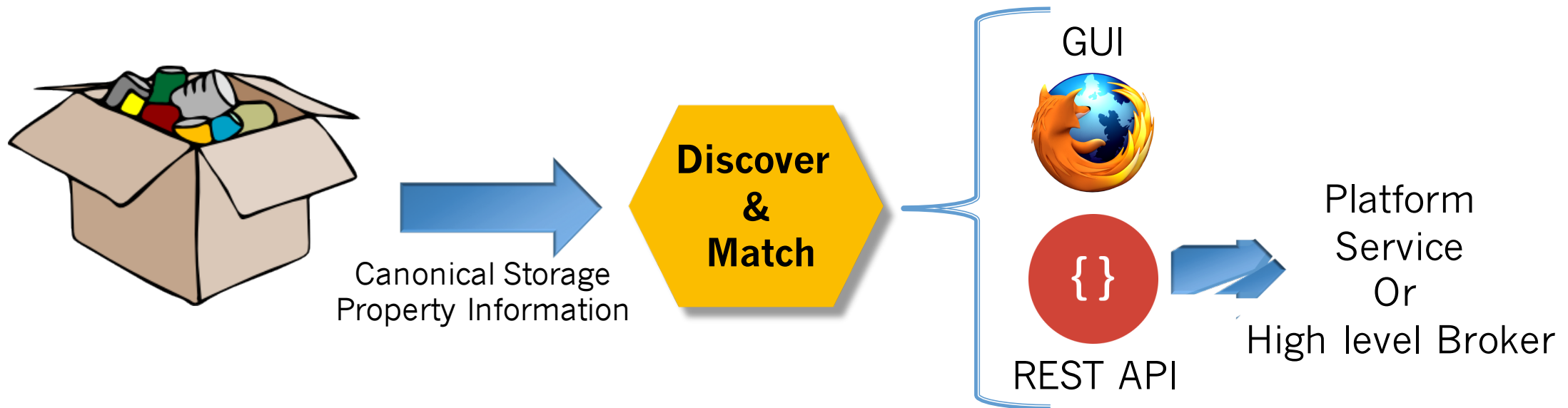
Large volume & cheap & archive

Discover and Match

Therefore: Introducing a new service

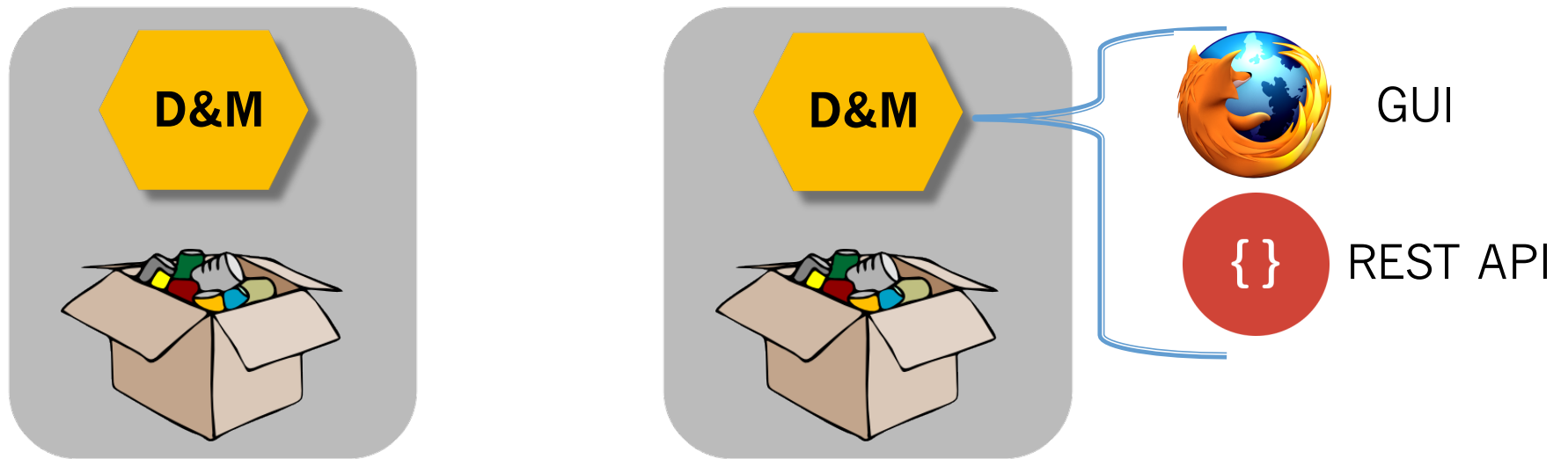


Translation and discovery



Canonical property federation

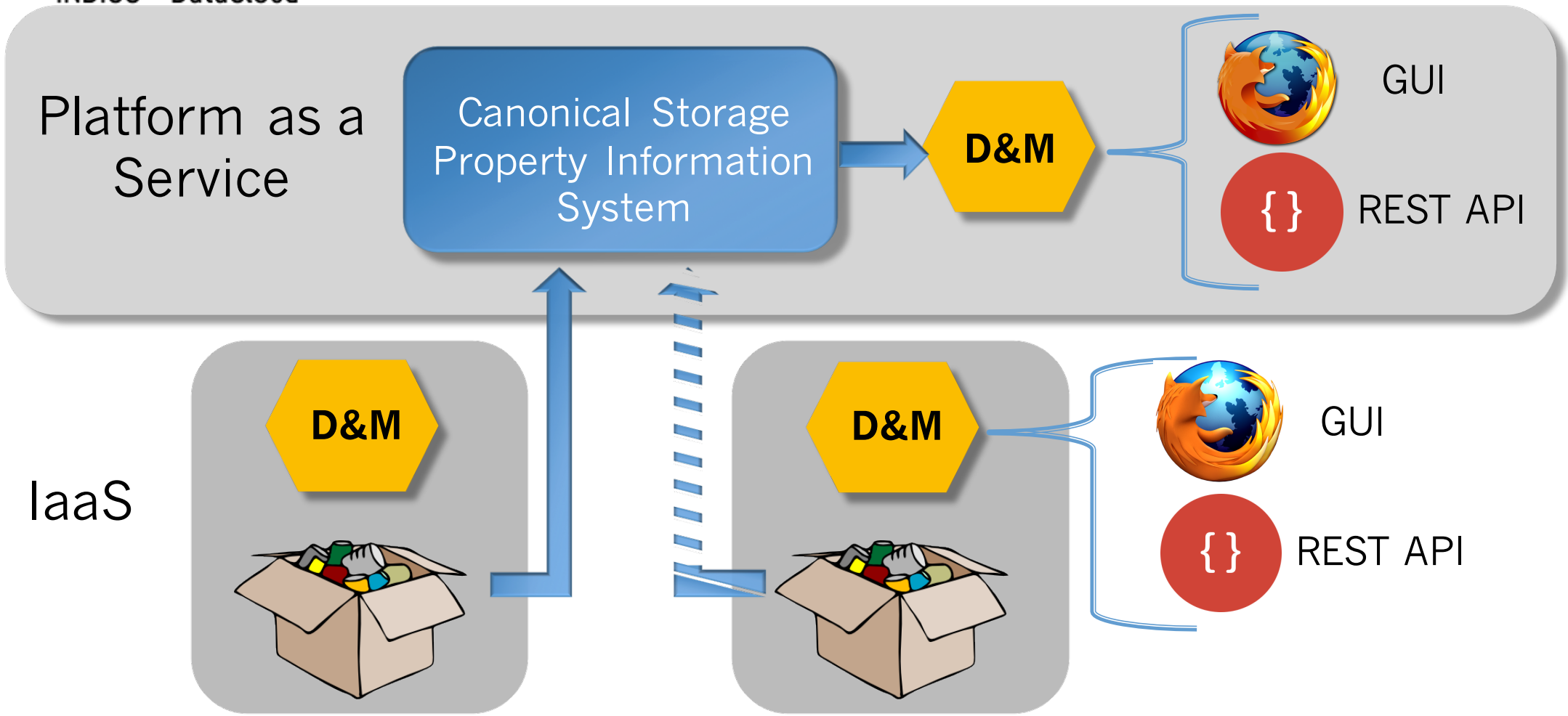
IaaS





INDIGO - DataCloud

Canonical property federation



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.



INDIGO - DataCloud

More problems to solve

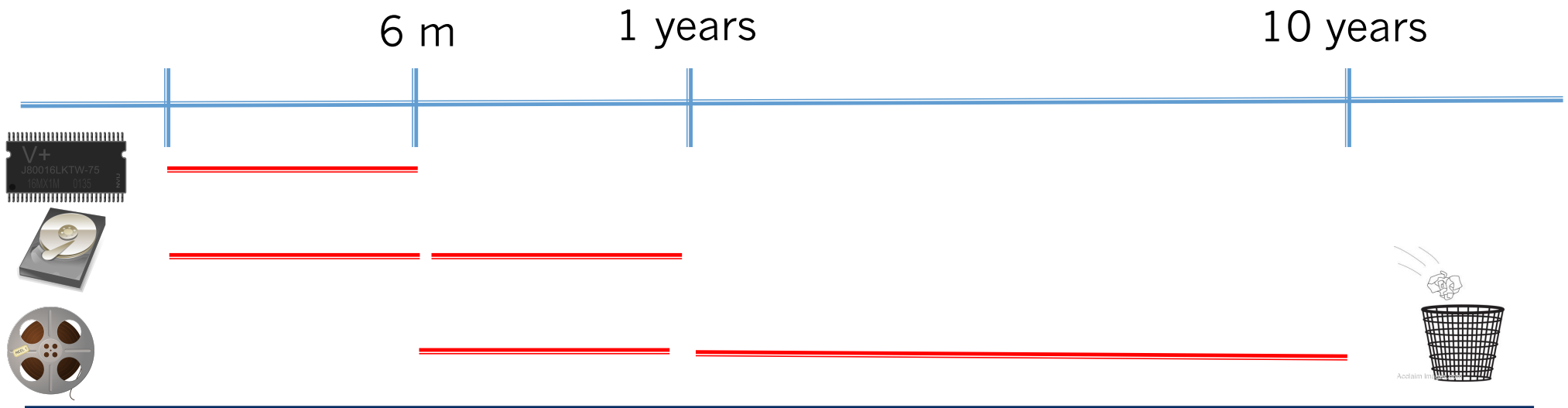
- How does the client provide the storage class to the storage system ?
 - Bucket
 - Directory
 - Additional argument in WebDAV, FTP etc
- The system only provides the class, it doesn't 'promise' the space.
 - Do we need a space reservation protocol ?
 - Similar to hotels.com. Check hotel pictures first, 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 Google don't
 - dCache and HPSS do.



INDIGO - DataCloud

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





INDIGO - DataCloud

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



INDIGO - DataCloud

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)



INDIGO - DataCloud

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**