

**INDIGO - DataCloud**

With kind contributions by  
Shaun DEWITT, EUDAT

# 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



# Content

---

- Introducing INDIGO-DataCloud.
- What is the issue with QoS in Storage ?
- Which part are we trying to solve ?
- What is our approach ?



INDIGO - DataCloud

# INDIGO DataCloud Cheat Sheet

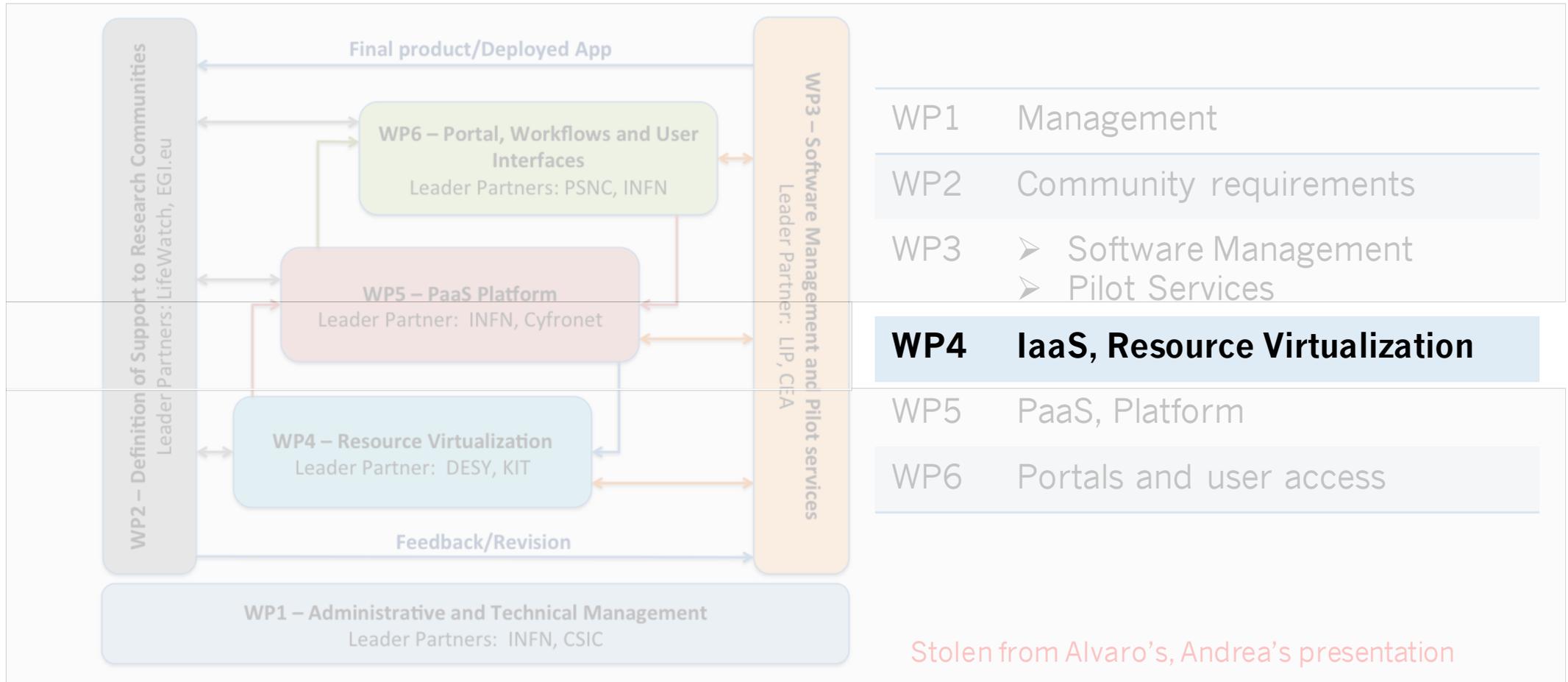
---

## ■ **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





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

# 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.

# Introducing part of the issue

---

Storage provisioning for large public infrastructures is facing two contradicting problems

- The complexity of storage and storage management
- The large variety of sciences and their diverging expectations on storage



INDIGO - DataCloud

# Infrastructure Problem

---

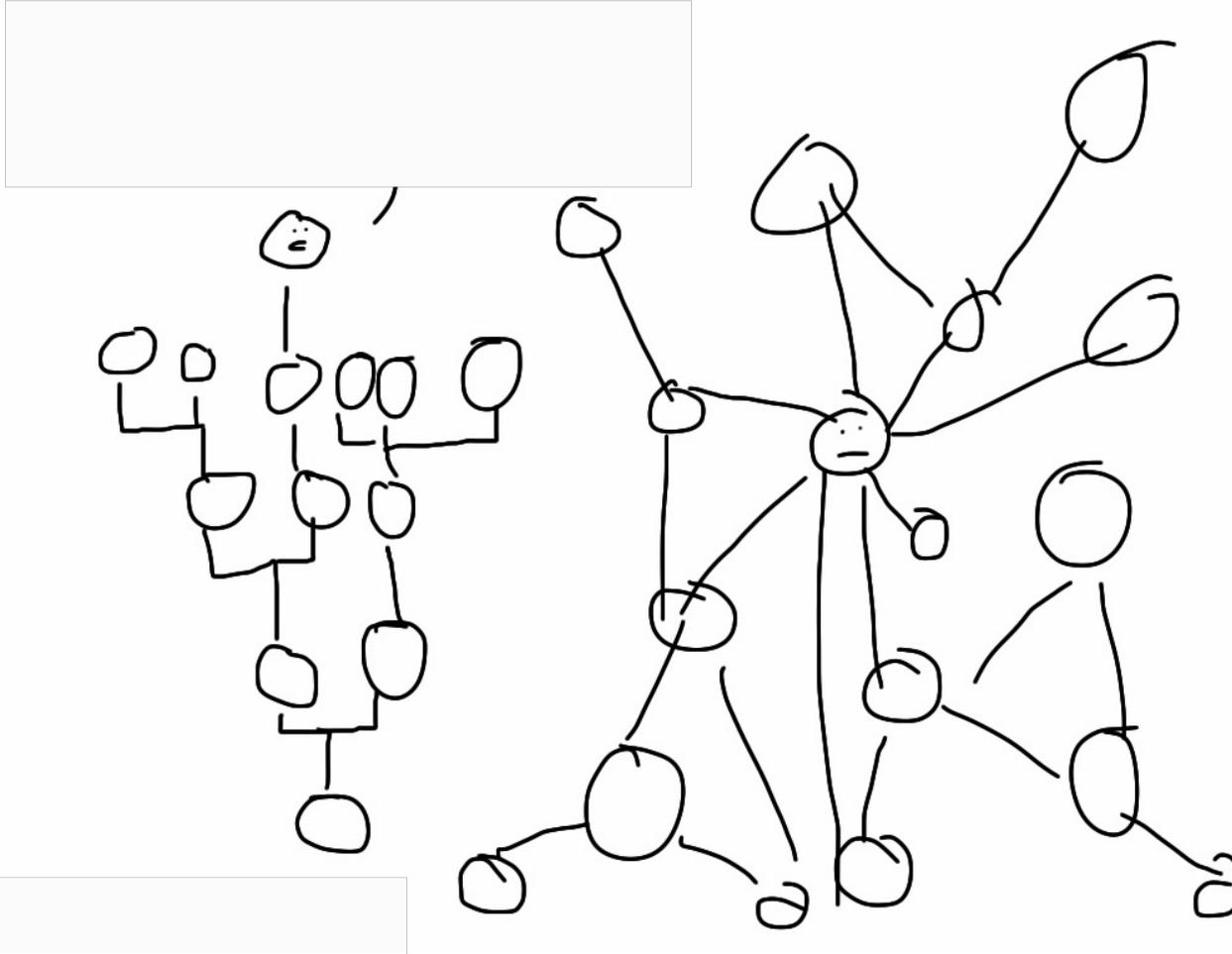
- **Infrastructures**

- Are growing in
  - size of storage and
  - number of supported sciences and communities and
  - Number of direct customers accessing storage
- They all have different ideas on how to use storage.
- Serving them in the old fashion doesn't scale any more
- So you need an API's or portals to let them select what they need

- **Infrastructures are used by platforms,** which

- tend to federated resources from different locations and storage providers.
- So storage needs to be brokered and procured automatically (or programatically)

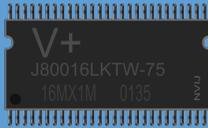
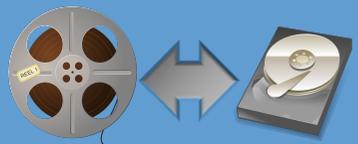
# Examples for Storage Complexity





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.

Starting with:

- a) What are storage properties.
- b) What are storage property values.



INDIGO - DataCloud

# Storage quality properties and values

---



Property



Property Value

- **Access Latency**
  - How long does it take from the request for a byte to receiving that byte.
- **Retention Policy**
  - What is the probability of data loss.
- **Access Mechanisms**
  - http, GridFTP, NFS, ....
- **Security**
  - encrypted during the transfer, on disk, end – to – end.
- **Authentication**
  - SAML, Open ID Connect, Password, X509



INDIGO - DataCloud

## How many 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

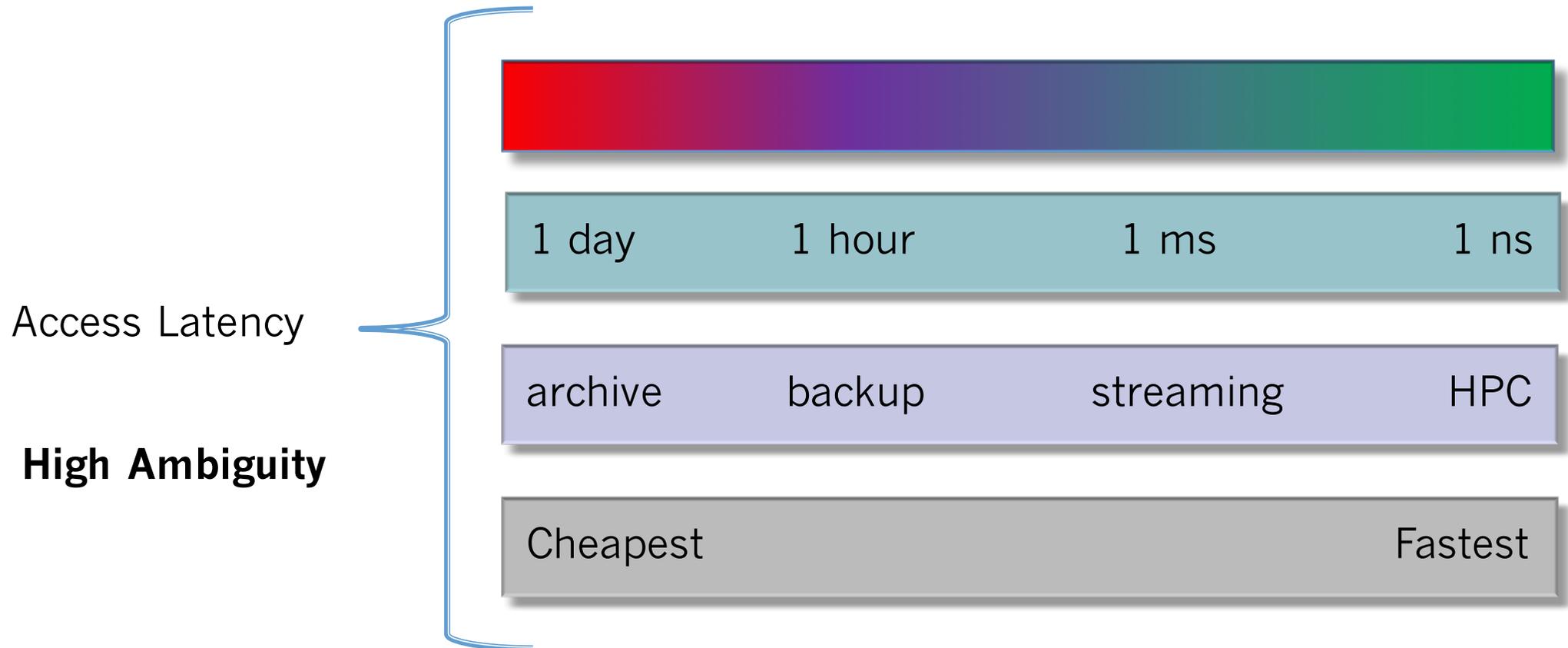
# Even more complexity

---

- QoS Property “Value Ambiguity”
- Property dependencies
- Property Quantization
- Non standard property zoo of existing system

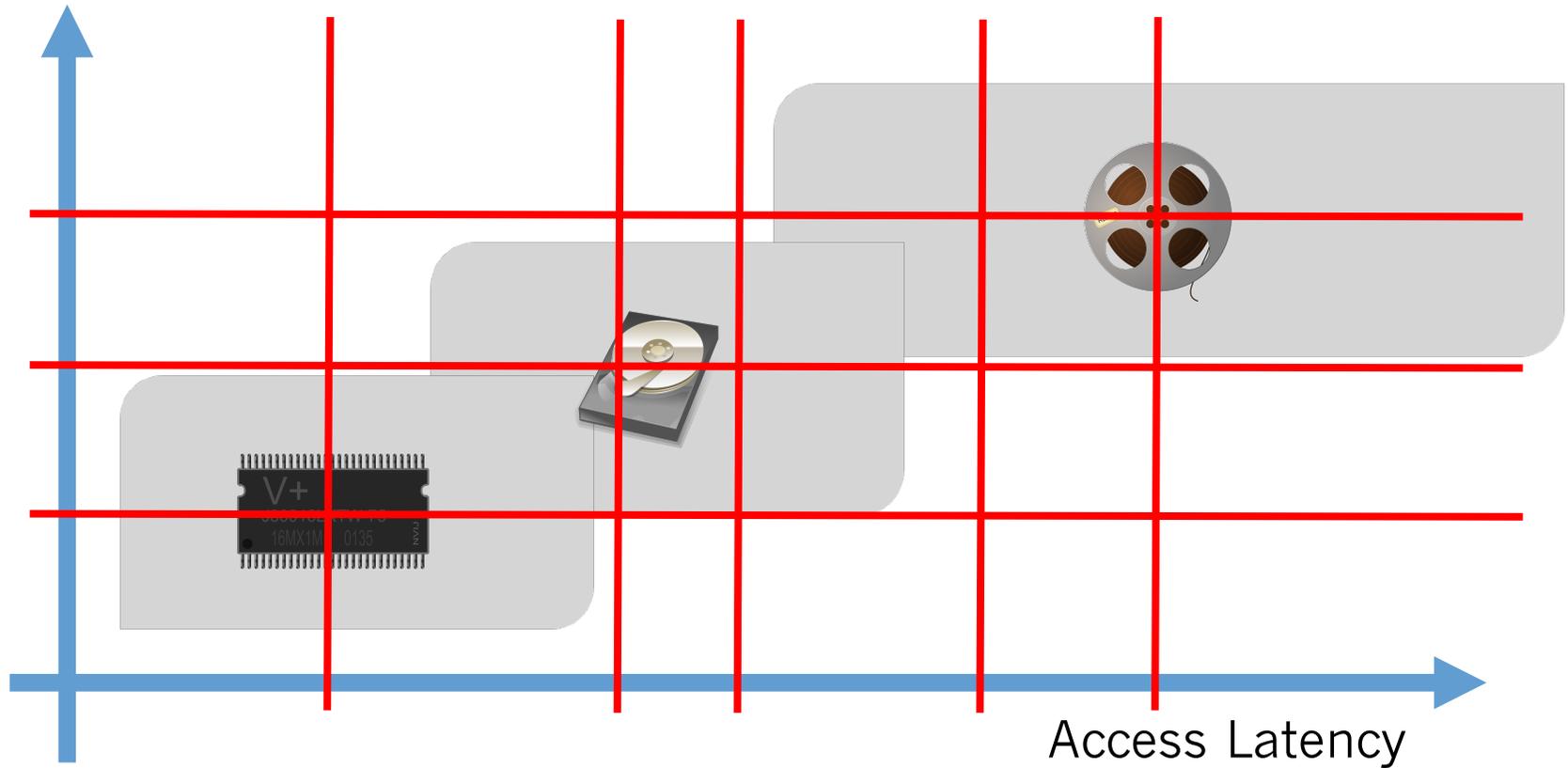
# QoS Property Value Ambiguity

---

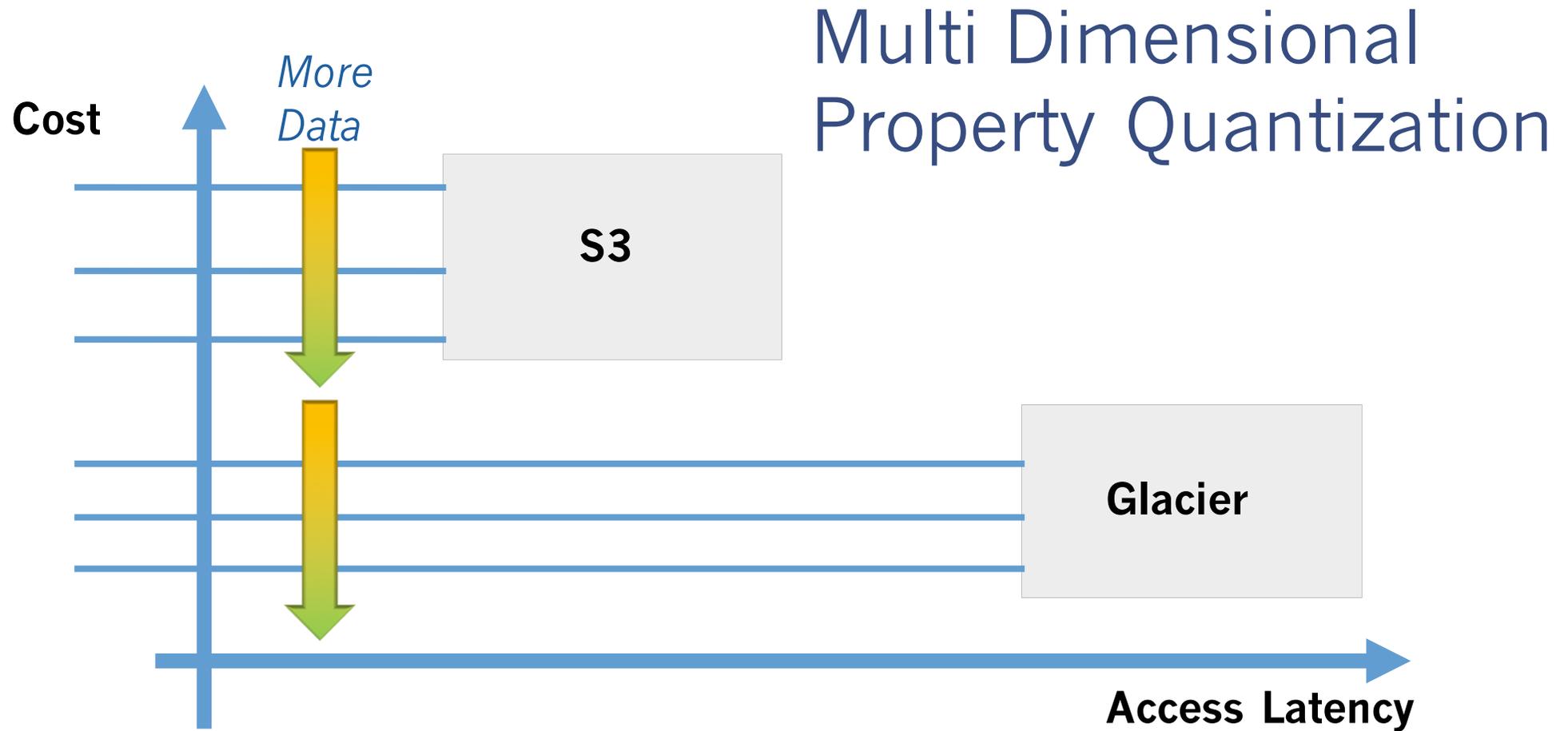


# Property dependencies

Durability



# 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		0.999999999
Media	Disk / SSD	*****	Tape
Replicas	1 Disk		2 Tape
Price	10 E/m/GB		20 E/m/GB

Avoiding ambiguities

!!! For EUDAT, those “Classes” are close to their “Services”

## How to get ...

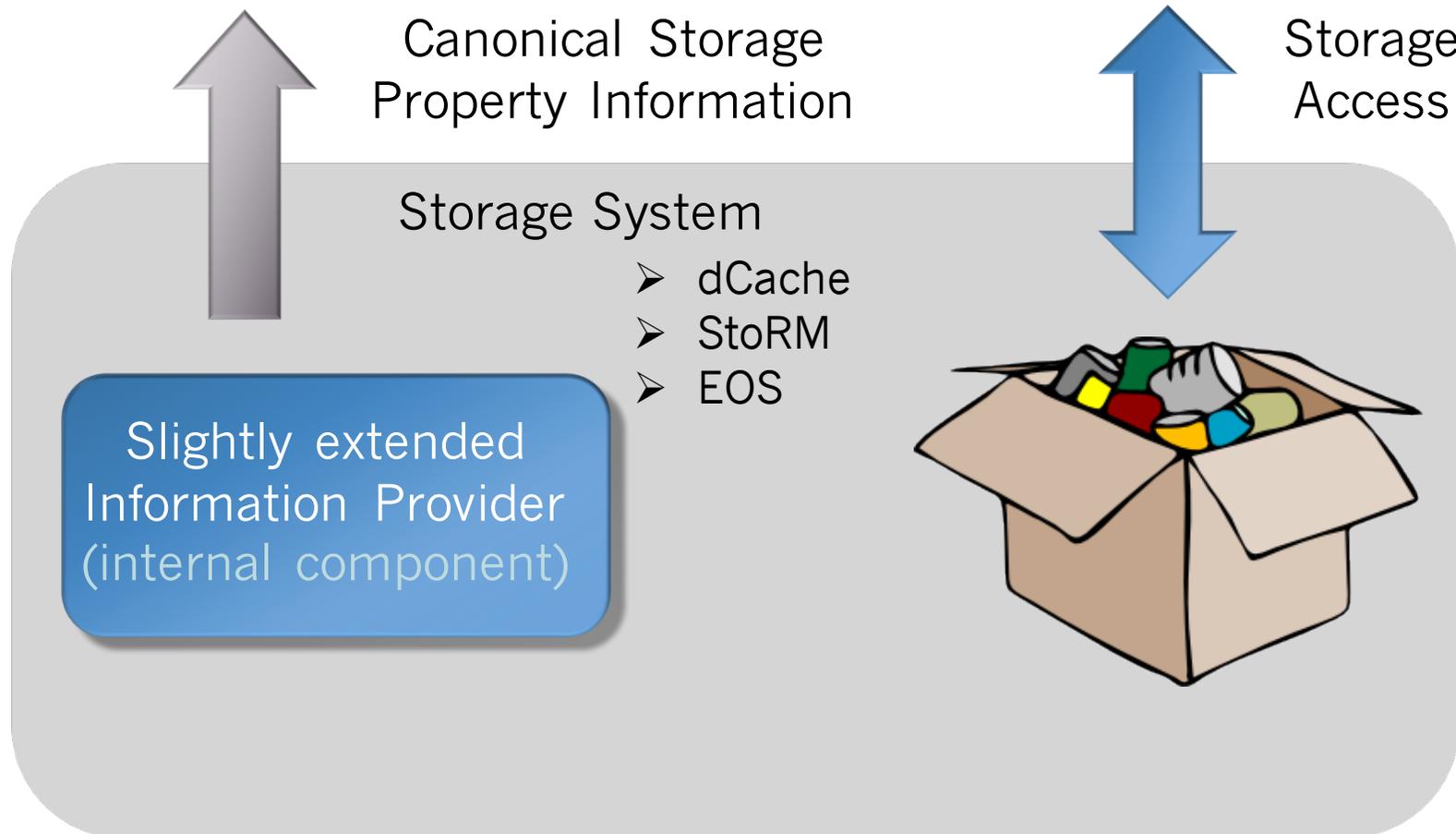
---

So after having defined  
**Canonical Storage Properties**  
and their values .....

How to get them

out of existing storage systems ?

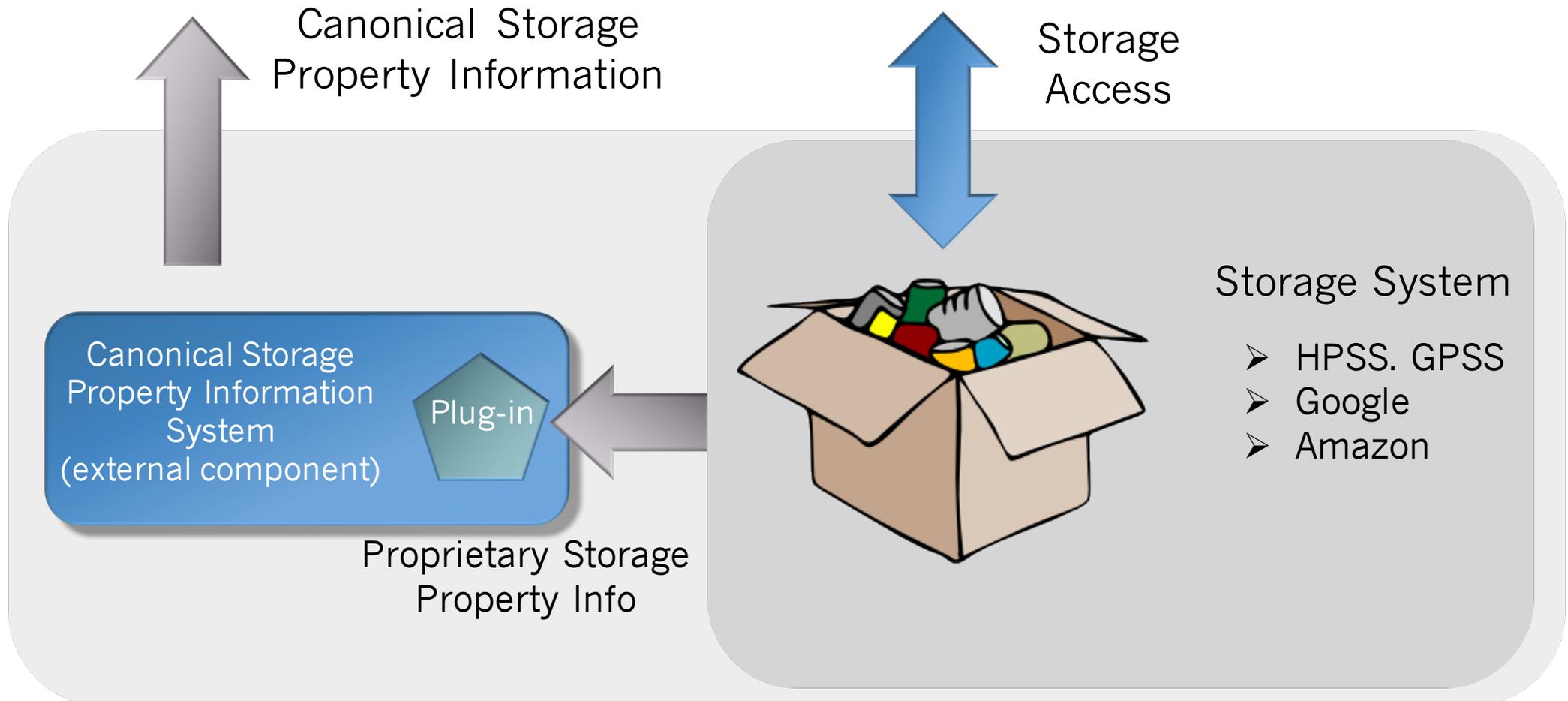
# Canonical Storage Properties





INDIGO - DataCloud

# Canonical Storage Properties





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 enduser.

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



INDIGO - DataCloud

## QoS views

---

### Examples on how a user would describe his/her needs

- Low latency & Lowest price
- Highest possible throughput & Short term
- Scratch & Very cheap
- Long Term Storage & Price not important



INDIGO - DataCloud

# That's what customers would expect

Your Magic Storage Wand



Euros/Month

1,05

▼ Basic

How much storage do you need ?



Quality



Scratch



Pretty Good



Rock Solid

Access



WebDAV



GridFTP



NFS 4.1 / pNFS

▶ Advanced

▶ Expert ( Extra Costs may apply 😊 )



INDIGO - DataCloud

# That's what customers would expect

Your  
Magic  
Storage  
Wand



Euros/Month

1,05

▶ Basic

▶ Advanced

▼ Expert ( Extra Costs may apply 😊 )

Media  Disk  Tape  SSD  Tape Remote

Access Latency  Nano Seconds

Retention  Absolute

Access  http  WebDAV  GridFTP  NFS 4.1 / pNFS

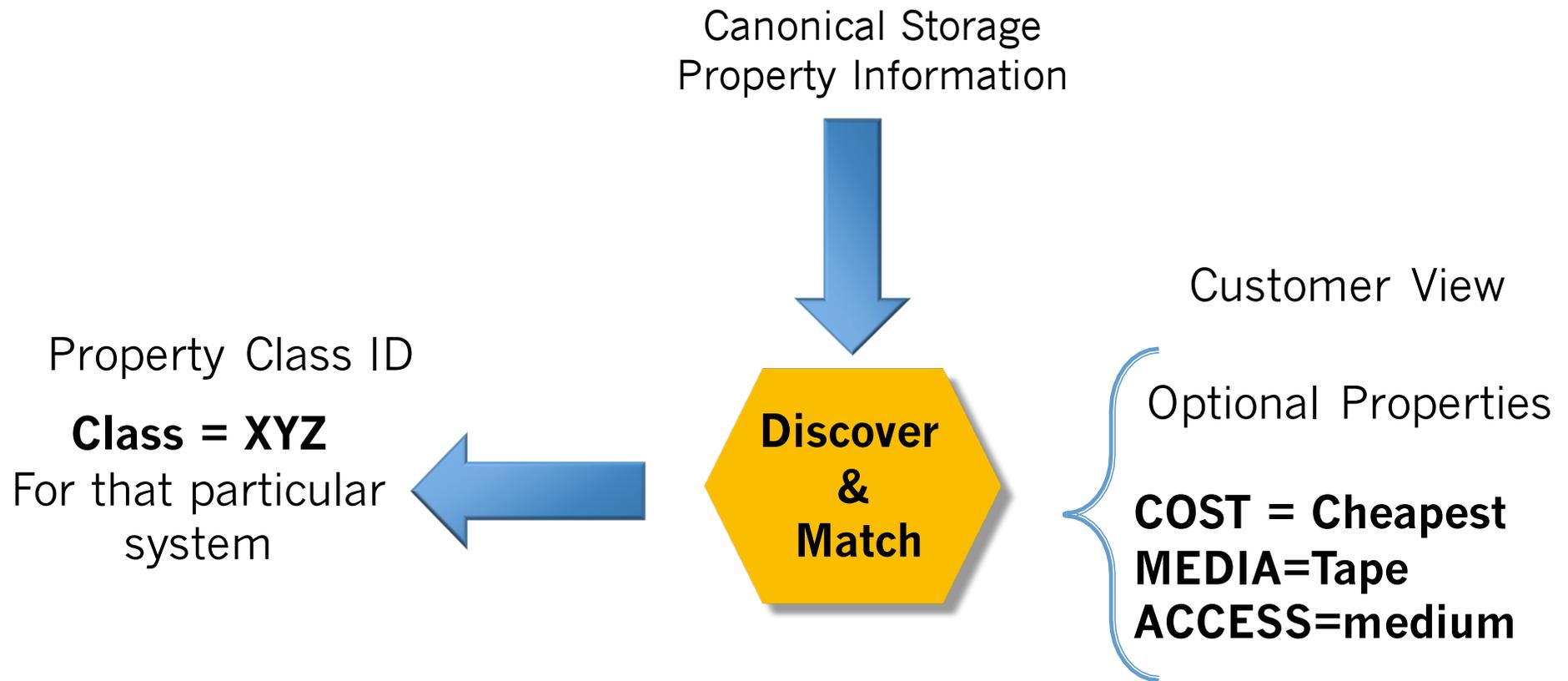
Security  X501  SAML  Open ID Connect  Password

Extre  Attach OID's  Support Macaroons

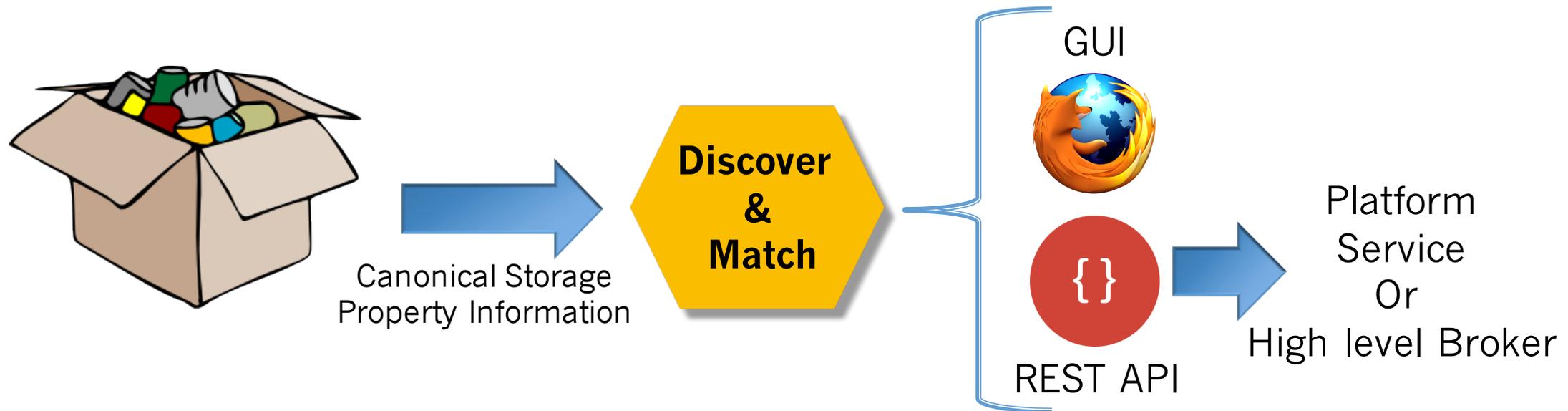
Therefore: Introducing a new service

# Discover and Match

---



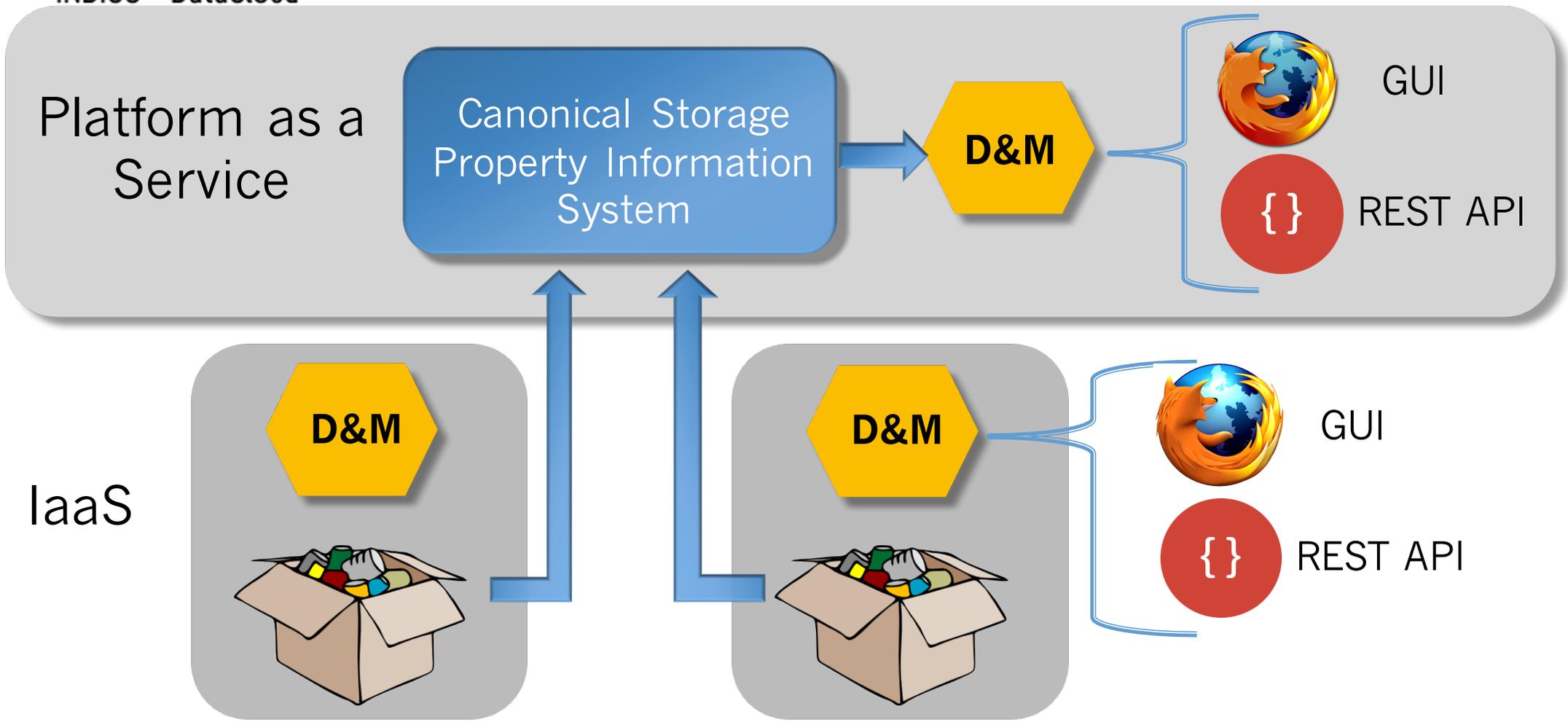
# Translation and discovery





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



INDIGO - DataCloud

## To summarize the procedure

---

- Storage Systems provide a **set of 'classes' describing standardized storage properties with standardized values.**
- Neither the name of the classes nor the combination of properties are standardized, they depend on the storage system.
  - Like S3 and Glacier are the names of the class
- **Matchmaking software** tries to match the various classes to the non standard and site specific requirements of the communities or individuals and returns the **closest match** to the customer.
- For further requests, the customer will use the 'class name' in the request. That could be a **directory, a space token or a container.**



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.



## Current status

---

- **Creating a RDA working** group (Paris and Tokyo)
  - Name : Quality of Service and Data Life Cycle Definitions WG
  - Currently agreeing on a Charter.
  - 10 Committed members (sites and communities, Flexier ...)
- **Contributing to the SNIA CDMI reference implementation**, as this is our planned transport or QoS steering.
- Defined version 1 of RESTFUL API
- Defining a **CDMI extention** to describe the storage properties and values.
- **Implementations** are ongoing for dCache, StoRM and the GPFS and TSM pluggins.

[Paul.millar@desy.de](mailto:Paul.millar@desy.de)

**INTERESTED ?**



INDIGO - DataCloud

# Summary

---

- INDIGO provides funding to standardize QoS and possibly Data Life Cycle of systems
- Scientific communities and EUDAT are showing 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 implementations are in progress (dCache, StoRM, HPSS, ...)
- **Contribution or ideas from your side are more than welcome.**

# Further reading

---

First Proposal for restful representation of our ideas.

