Data Replication

Automated move and copy of data

Data Staging

Moving large amounts of data around, and moving it close to compute resources

EUDAT 2nd Conference
Rome, Oct. 28th-30th 2013

Giovanni Morelli
g.morelli@cineca.it
Claudio Cacciari
c.cacciari@cineca.it
Giuseppe Fiameni
g.fiameni@cineca.it
Johannes Reetz
johannes.reetz@rzg.mpg.de

SEVENTH FRAMEWORK PROGRAMME

EUDAT

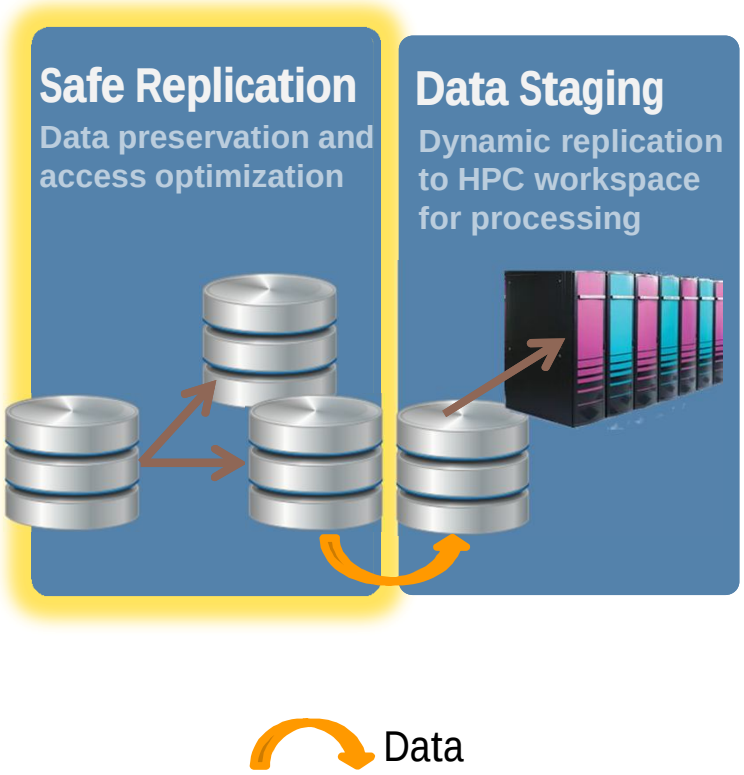
Safe Replication Outline

- Principles
- Data movement:
  - Which kind of service?
  - Which kind of users?
- Flexibility
- Different transfer strategies
- Policies
- Performances
- Different federation strategies
- PID and registered data

Principles – where we want to be

1. Data deposited will be preserved in perpetuity
2. Data are best curated in their own communities
3. Access to data in the Collaborative Data Infrastructure (CDI) is free at the point of use
4. The Collaborative Data Infrastructure will not assert ownership of any data it holds

Data movement: staging or replication?



Replication

Safe Replication to enable communities easily create replicas of their scientific datasets in multiple data centres for improving data curation and accessibility

- *data bit-stream preservation*
- *more optimal data curation*
- *better accessibility of data*
- *identification of data through Persistent Identifiers (PIDs)*

Persistent Identifiers (PID)

- EUDAT relies on the **EPIC service** to associate persistent identifier to digital objects (http://www.pidconsortium.eu).
- EPIC is an identifier system using the **Handle infrastructure.**

- Its focus is the registration of data in an early state of the scientific process, where lots of data is generated and has to become referable to collaborate with other scientific groups or communities, but it is still unclear, which small part of the data should be available for a long time period.

http://www.ands.org.au/services/pid-policy.html

Wath users want

replicate my collection X to three data centres and store the collection safely for 10 years

Are you talking about clouds?

I already do it every day in my cloud space !

We are talking about a …

robust

safe

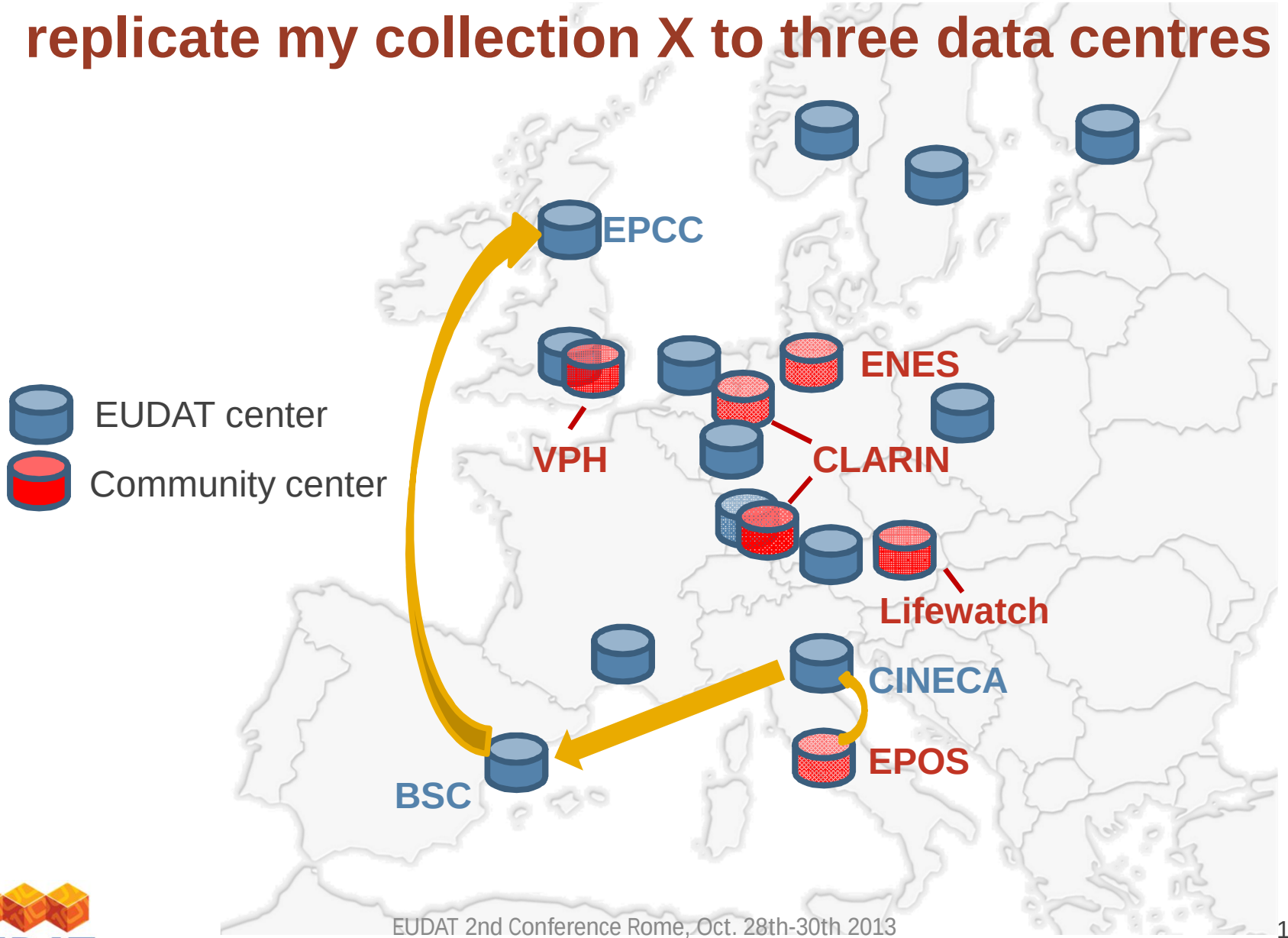
highly available

Replication Service

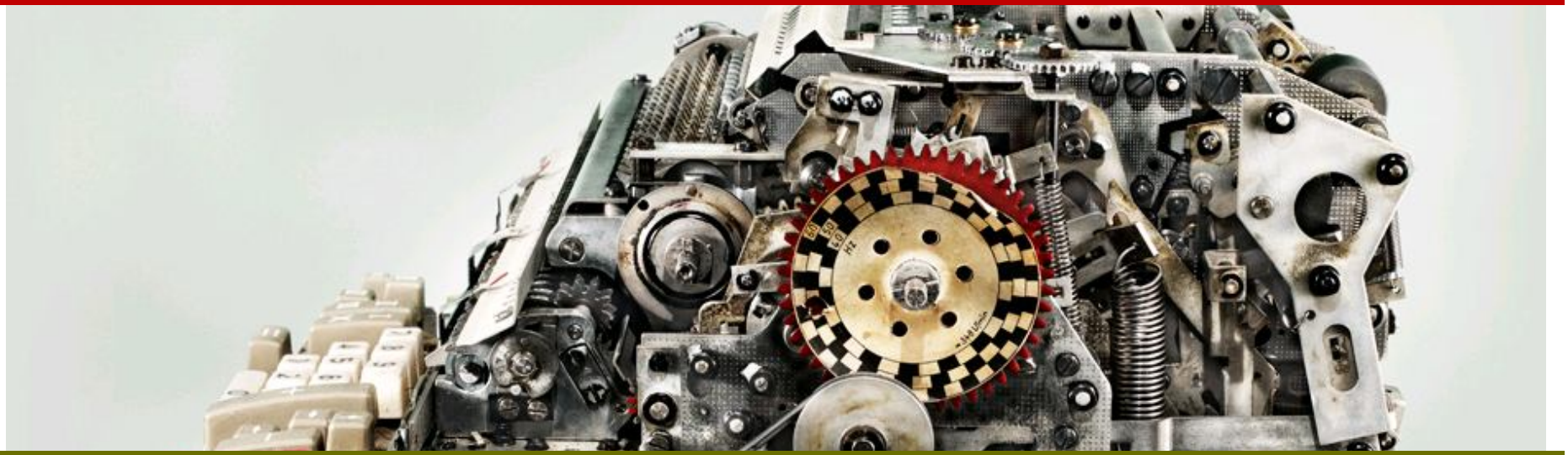
… which is not a *personal* cloud space


What about trust?
Can you find where your data are physically stored on the cloud?
Or who can access them?
No, because clouds are opaque
While a Collaborative Data Infrastructure is transparent

replicate my collection X to three data centres

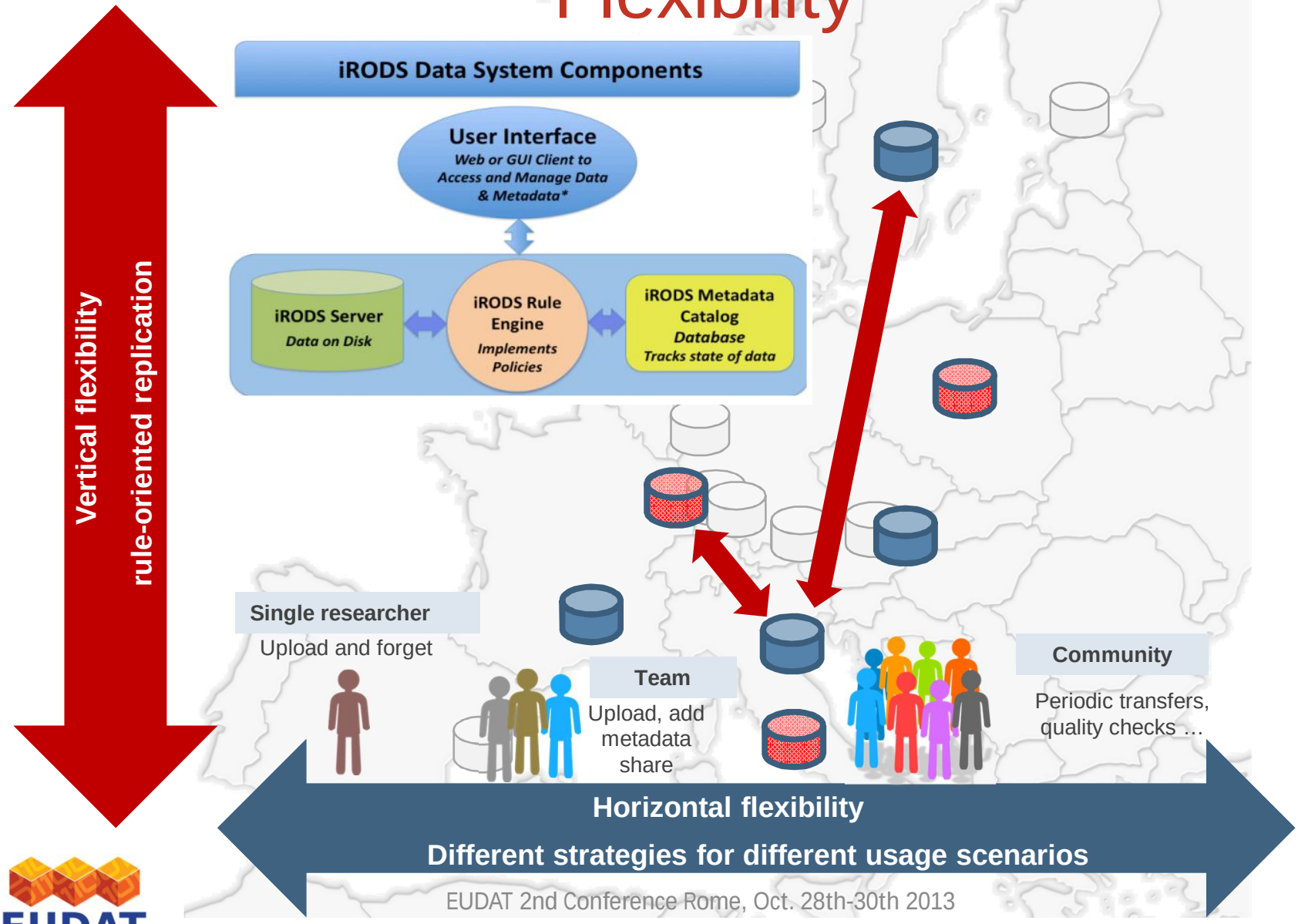


Then is it a complex mechanism suited only for expert data managers?



No.
It is a flexible approach able to encompass quite different usage scenarios.

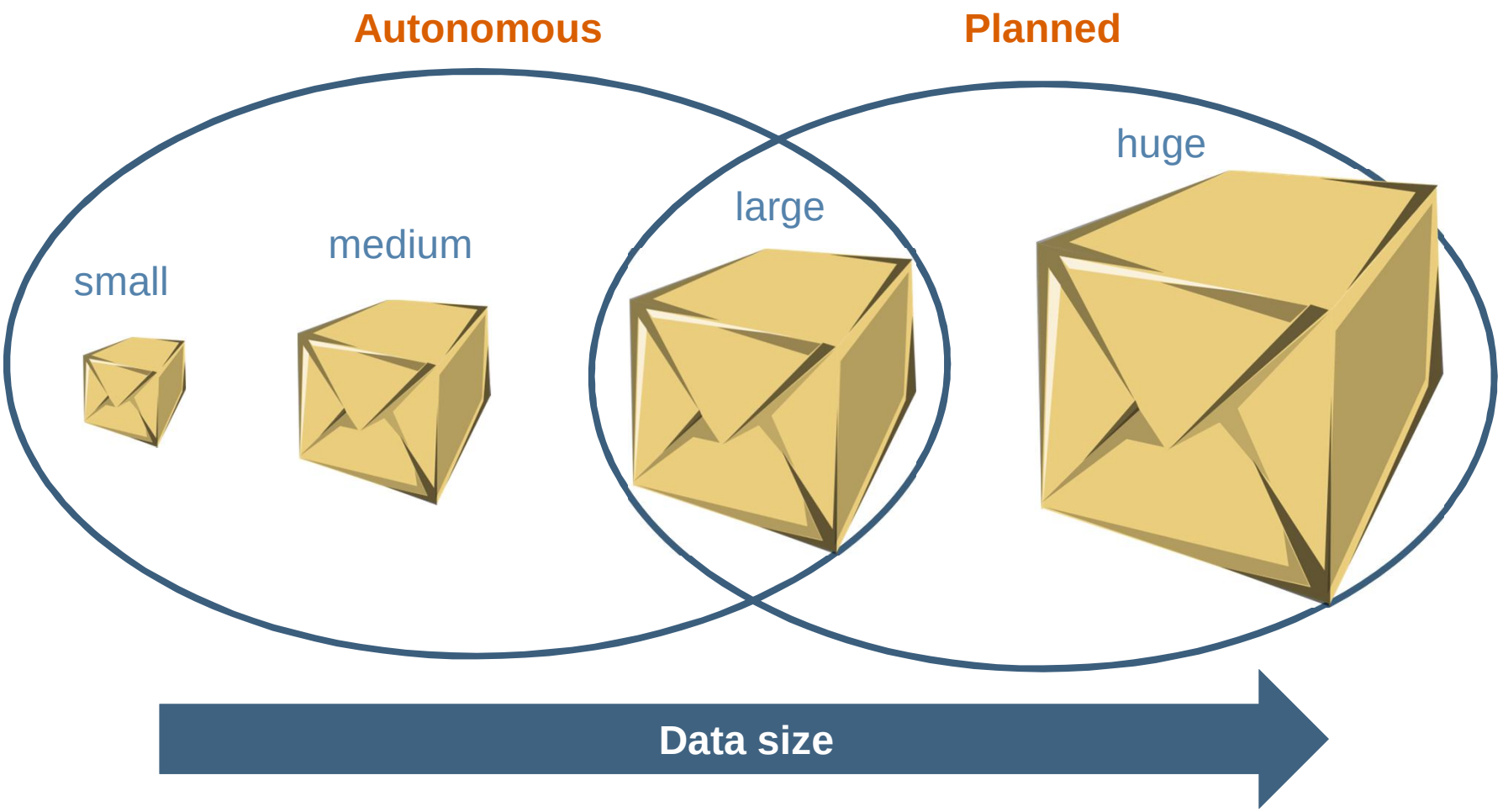
Flexibility



iRODS

- iRODS is a data management system, which integrates a **rule engine**
- **The rules can be triggered automatically** based on specific events (for example a data set is moved to a particular directory)
- **Invoked from remote** via iRODS command line client or integrated into applications based on iRODS java (Jargon) and python (PyRods) API

Transfer approaches



Coming back …

replicate my collection X to three data centres and store the collection safely for 10 years

Apparently a simple statement

But you need to plan it, then you need …

Policies !

EUDAT consortium is working on

Policies for the Collaborative Data Infrastructure management

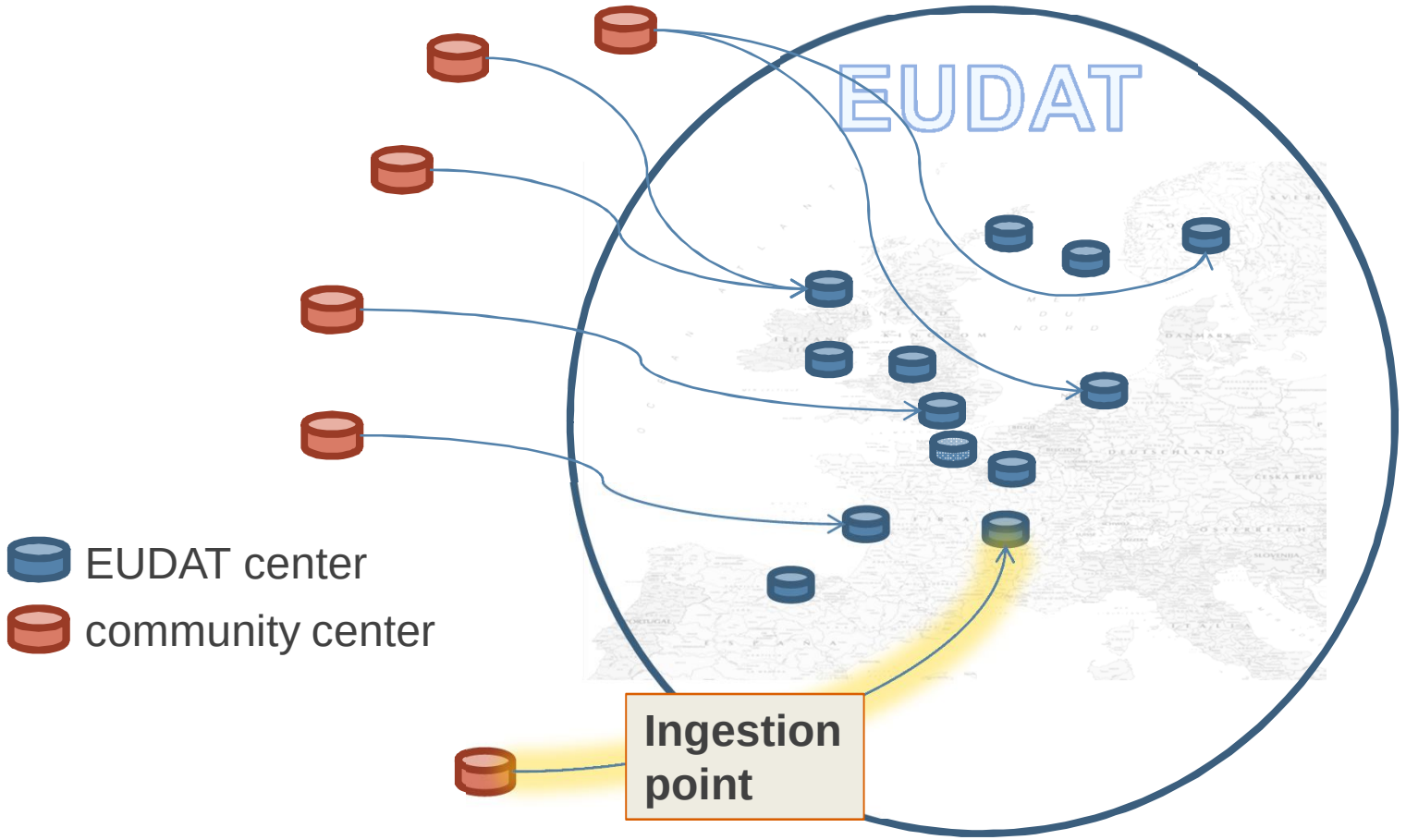
We can call them “internal policies”

How to manage software updates?

What about security bugs?

Which systems are monitored?

Each community has one or more doors to connect to the infrastructure



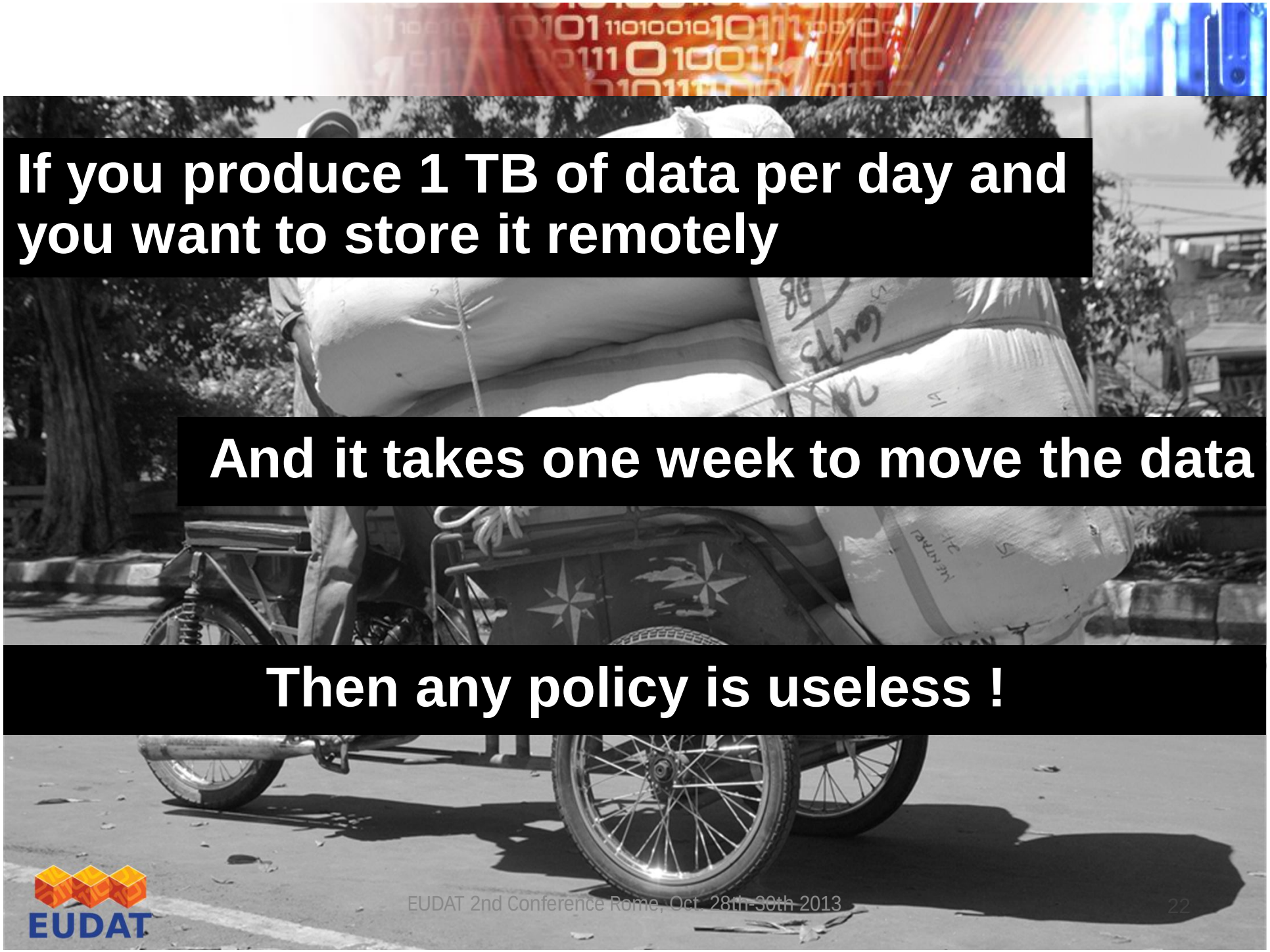
replicate my collection X to three data centres and store the collection safely for 10 years …

Updating the sub-collection X_1 weekly

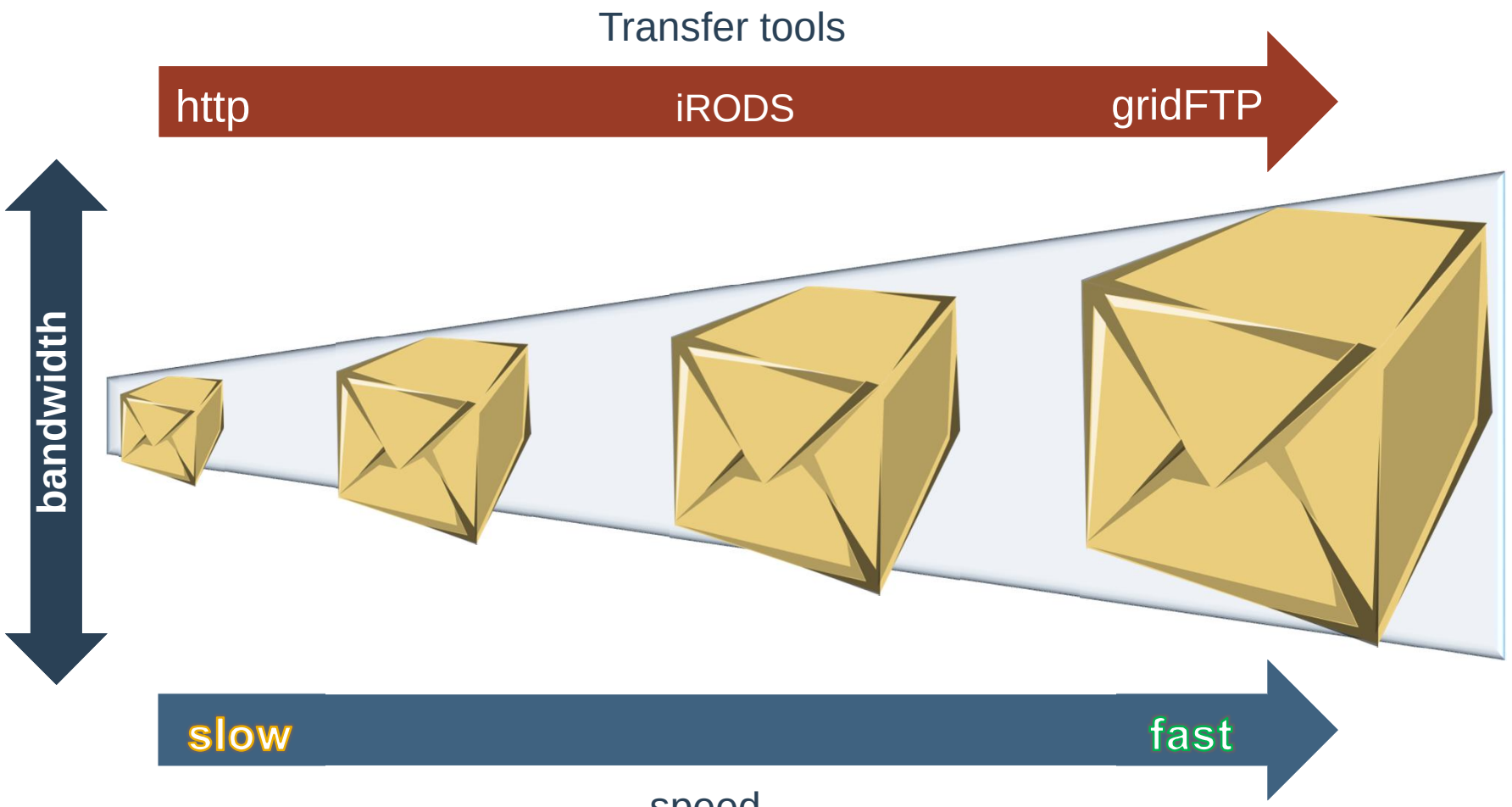
And the sub-collection X_2 hourly

And keeping on-line the data uploaded during the last six months


So far so good, we have our infrastructure, our policies
What else is missing?
Performances?

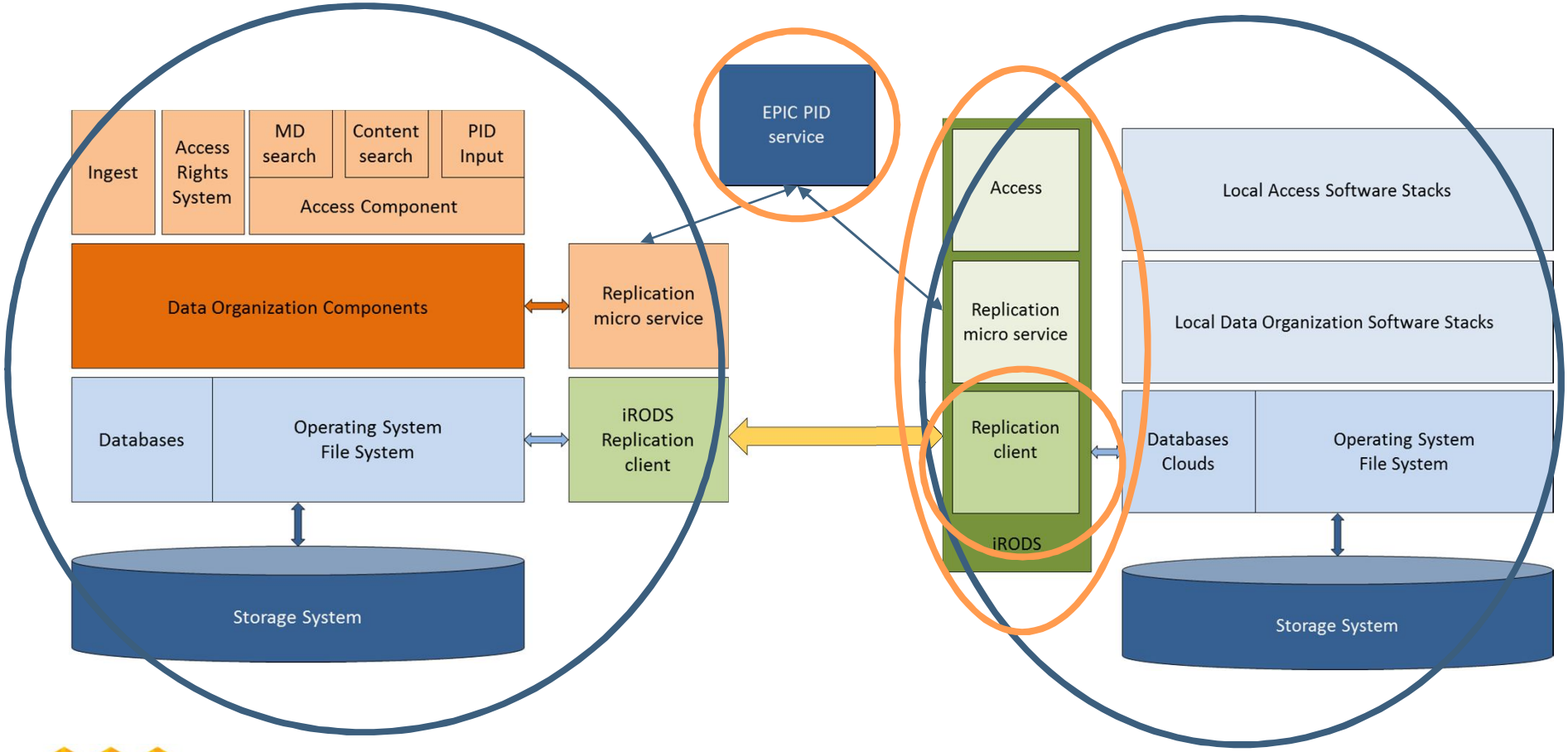

If you produce 1 TB of data per day and you want to store it remotely
And it takes one week to move the data
Then any policy is useless !

High Performance Transfers



To join or not to join

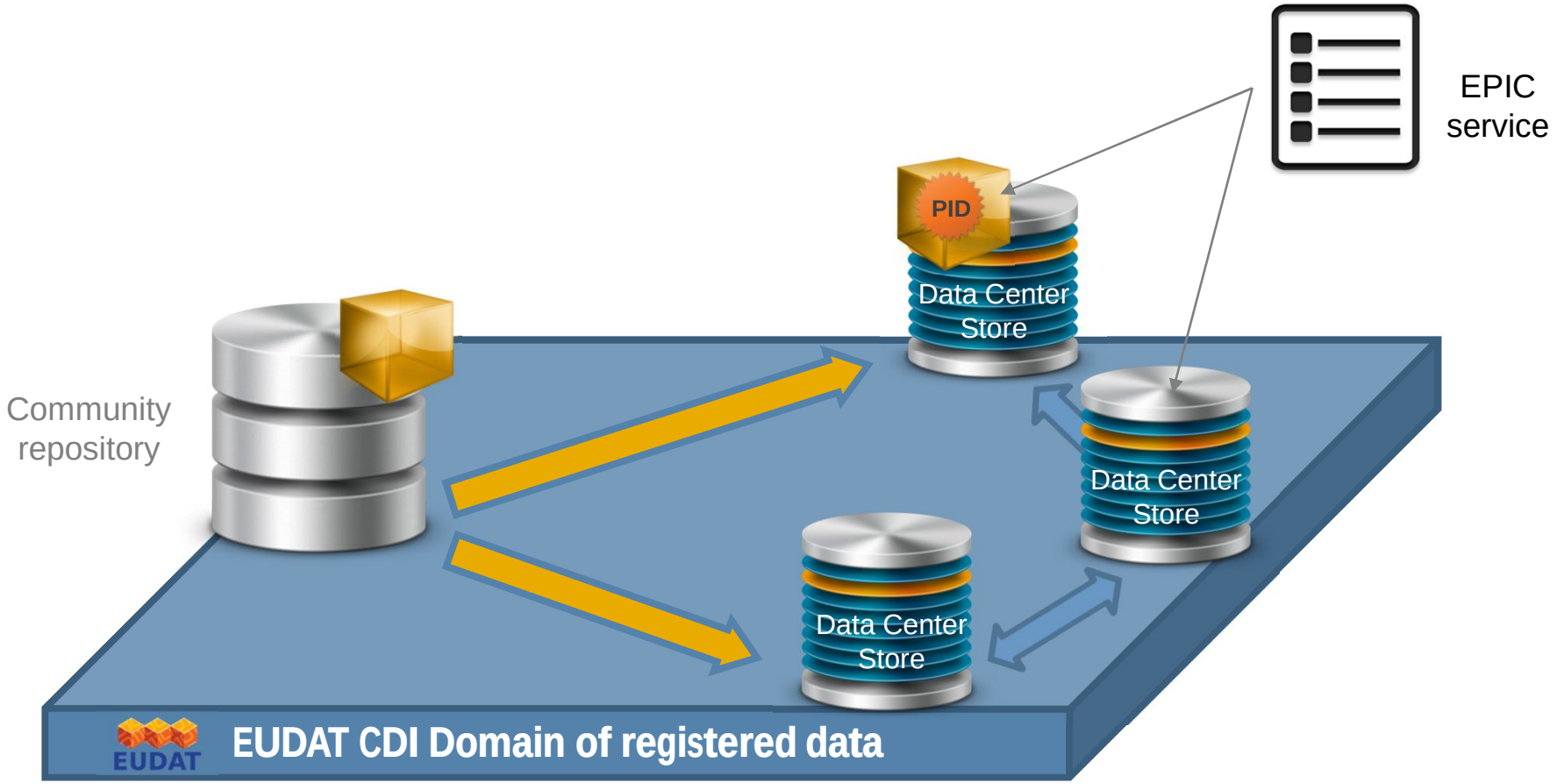
Join: automated data movement server to server



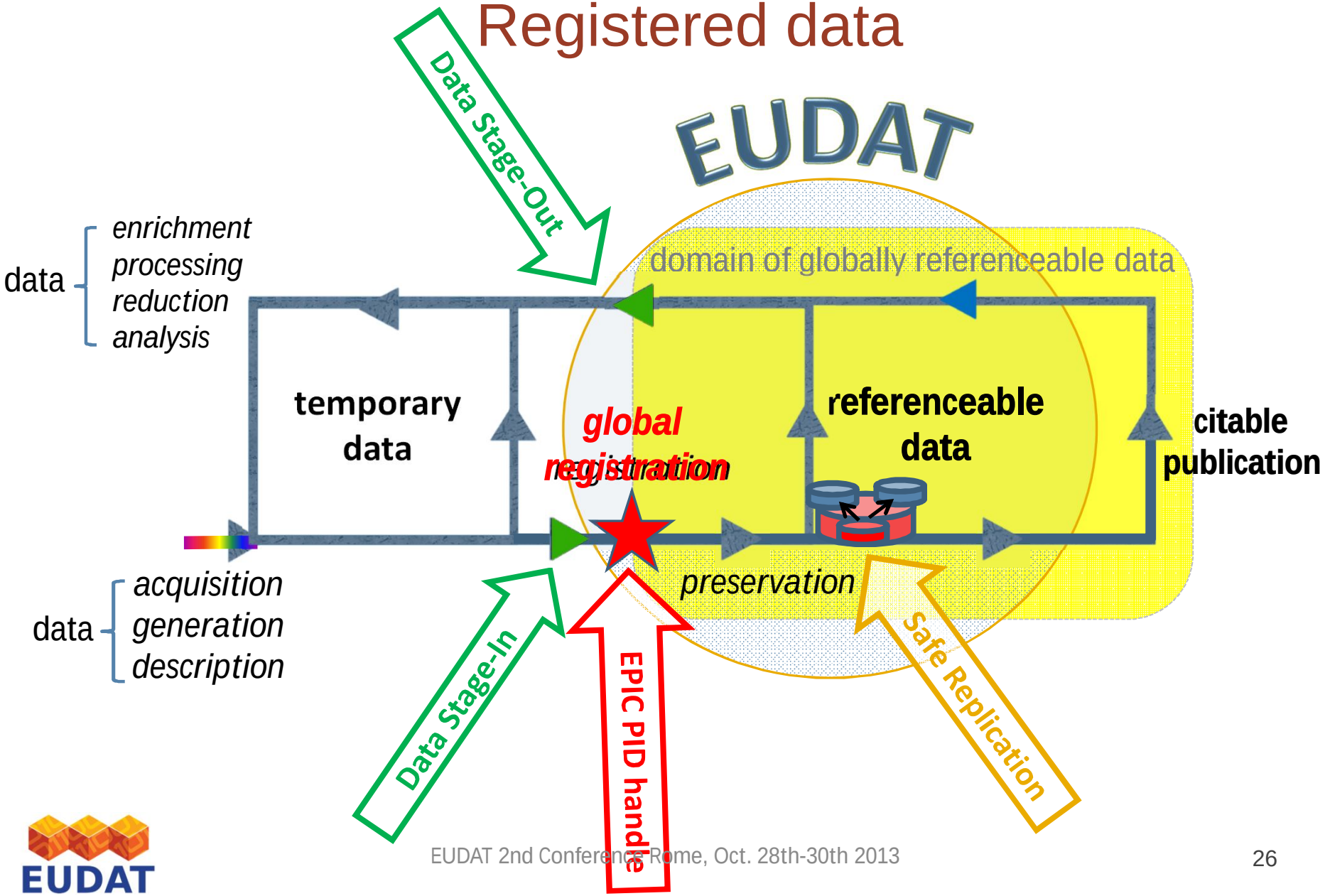
Data center

Community repository

Safe replication explained



Registered data



Finally, all together

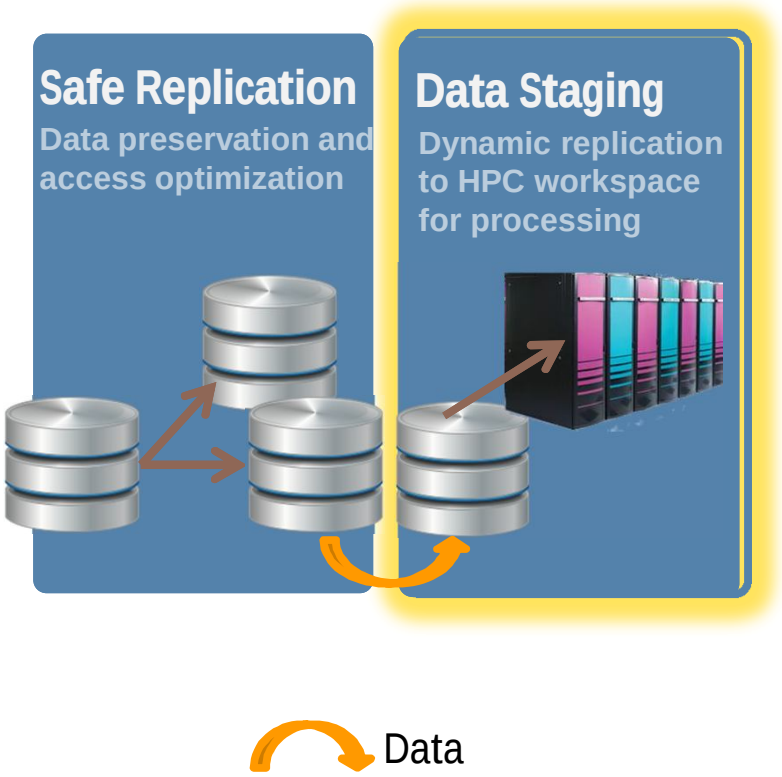
The set up of the automated data transfer between EPOS community and EUDAT touched all the aspects we mentioned so far:

EPOS ***joined*** the EUDAT CDI
We defined a ***specific policy*** with them
We ***tuned the transfer tools*** to achieve the best performance, but the HP tool (GridFTP) was useless since the bottleneck was the bandwidth
So we chose a ***more flexible*** tool like iRODS irsync protocol
In fact in order to achieve a hourly synchronization we exploited checksum sync and file age limit options.

Data Staging Outline

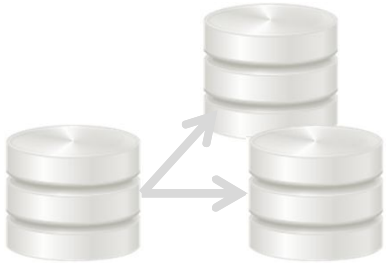
- Definition
- Moving data around
- Size/Performances
- Transfer options
- StageIN/StageOut
- Data Staging Script
- All services together

Data movement: staging or replication?



Staging

- **Safe Replication** to enable communities easily create replicas of their scientific datasets in multiple data centres for improving data curation and accessibility



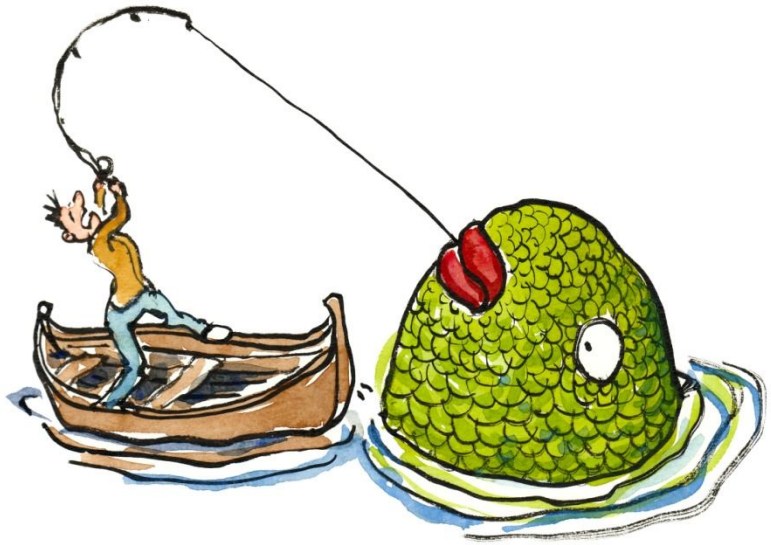
- **Data Staging** to facilitate communities to stage stored data onto external computational facilities, such as HPC resources

Moving large amounts of data around

Data sets can be large both in terms of

Numbers of objects

Single object size



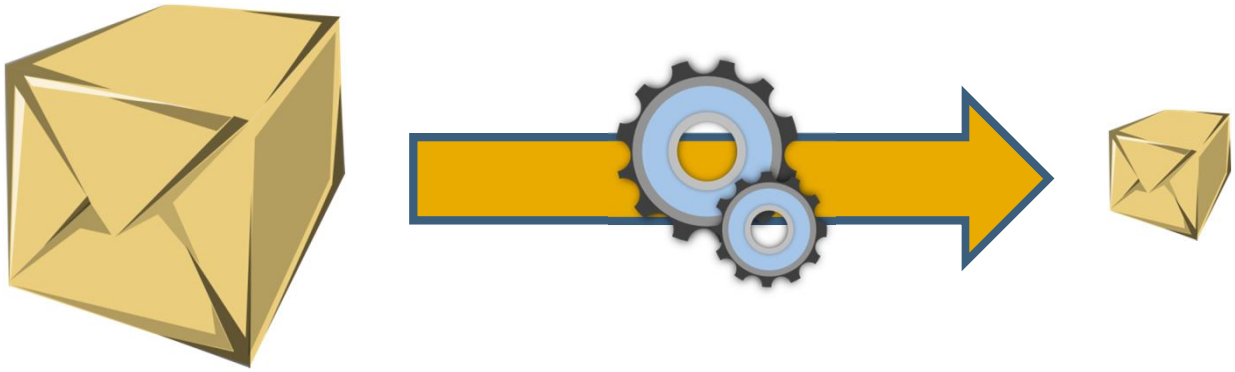
All data are gray in the data staging

The data types are irrelevant

Except when they can affect the size or the number of the files

When I should care about data types?

When you can compress them



Data transfer efficiency is crucial in data staging

Data Throughput – Transfer Times

Bandwidth Requirements to move Y Bytes of data in Time X

Bits per Second Requirements

	1H	8H	24H	7Days	30Days
10PB	25,020.0 Gbps	3,127.5 Gbps	1,042.5 Gbps	148.9 Gbps	34.7 Gbps
1PB	2,502.0 Gbps	312.7 Gbps	104.2 Gbps	14.9 Gbps	3.5 Gbps
100TB	244.3 Gbps	30.5 Gbps	10.2 Gbps	1.5 Gbps	339.4 Mbps
10TB	24.4 Gbps	3.1 Gbps	1.0 Gbps	145.4 Mbps	33.9 Mbps
1TB	2.4 Gbps	305.4 Mbps	101.8 Mbps	14.5 Mbps	3.4 Mbps
100GB	238.6 Mbps	29.8 Mbps	9.9 Mbps	1.4 Mbps	331.4 Kbps
10GB	23.9 Mbps	3.0 Mbps	994.2 Kbps	142.0 Kbps	33.1 Kbps
1GB	2.4 Mbps	298.3 Kbps	99.4 Kbps	14.2 Kbps	3.3 Kbps
100MB	233.0 Kbps	29.1 Kbps	9.7 Kbps	1.4 Kbps	0.3 Kbps

This table available at **http://fasterdata.es.net**

Data staging is Just in Time

You do not plan it

You can pre-stage… sometimes

There are techniques to "improve the efficiency"

Improve the efficiency

TCP tuning: refers to the proper configuration of buffers that correspond to TCP windowing

Pipelining (of commands): speeds up lots of tiny files by stuffing multiple commands into each login session back-to-back without waiting for the first command's response

Improve the efficiency…

Parallelizing: on wide-area links, using multiple TCP streams in parallel (even between the same source and destination) can improve aggregate bandwidth over using a single TCP stream

…improve the efficiency

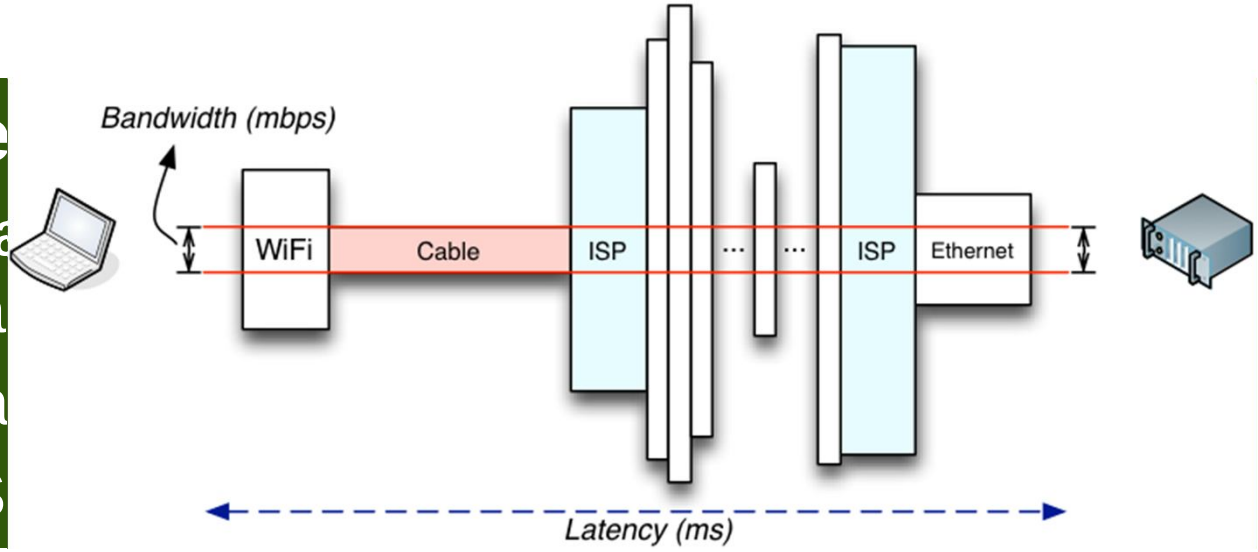
Striping: data may be striped or interleaved across multiple servers

Parallelism and TCP tuning are the keys

- It is much easier to achieve a given performance level with four parallel connections than with one
- A good TCP tuning can improve drastically performances

Latency inte

- Wide area da
  than LAN tra
- Many tools a
- Examples: S



High Performance Browser Networking by Ilya Grigorik

But efficiency is not all

Easiness of use

High reliability

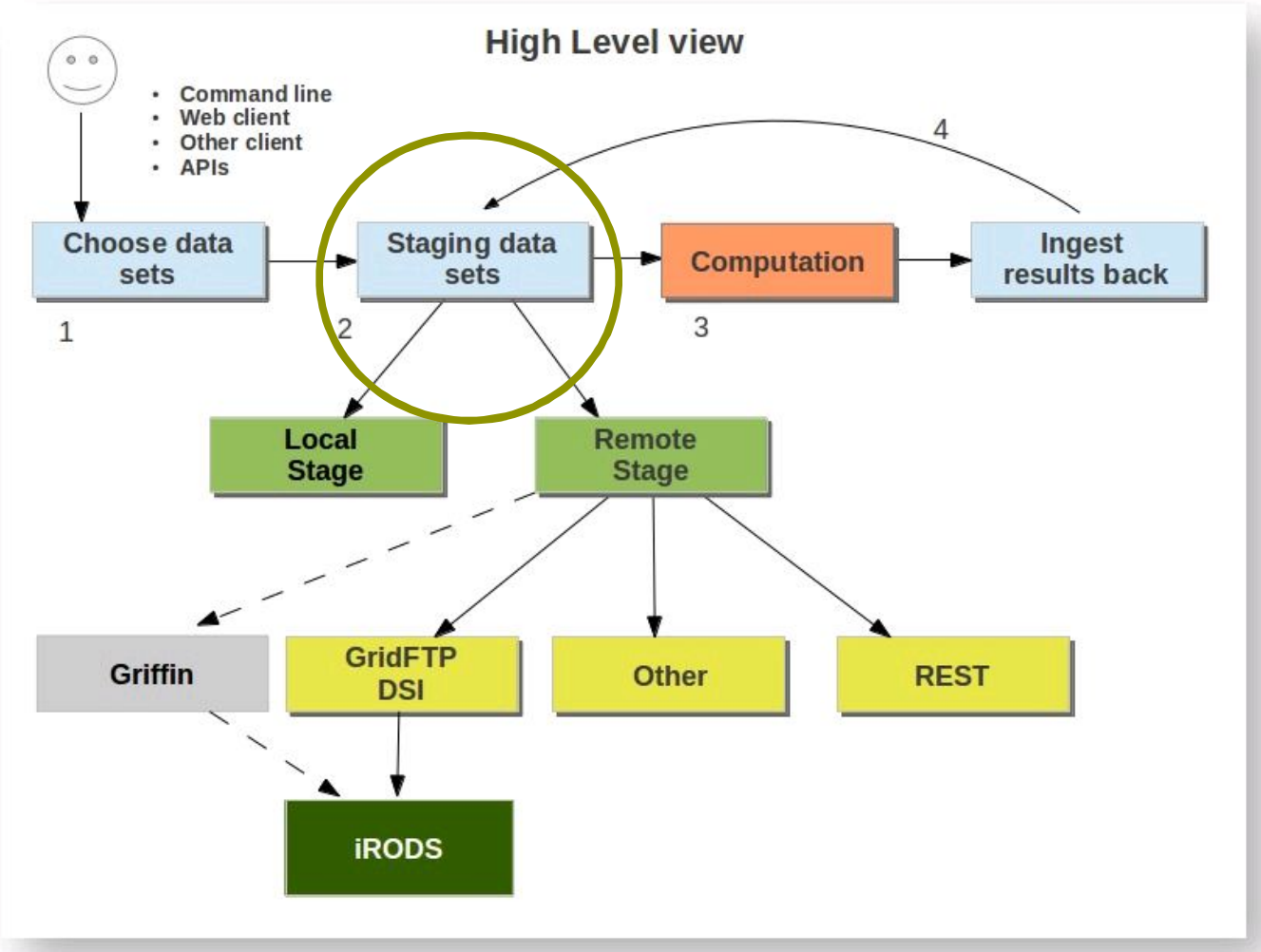
Third-party transfer

Possibility to control/limit the transfer throughput to avoid engulfing the network

Which priority?

Different scenarios, different needs

Move the data



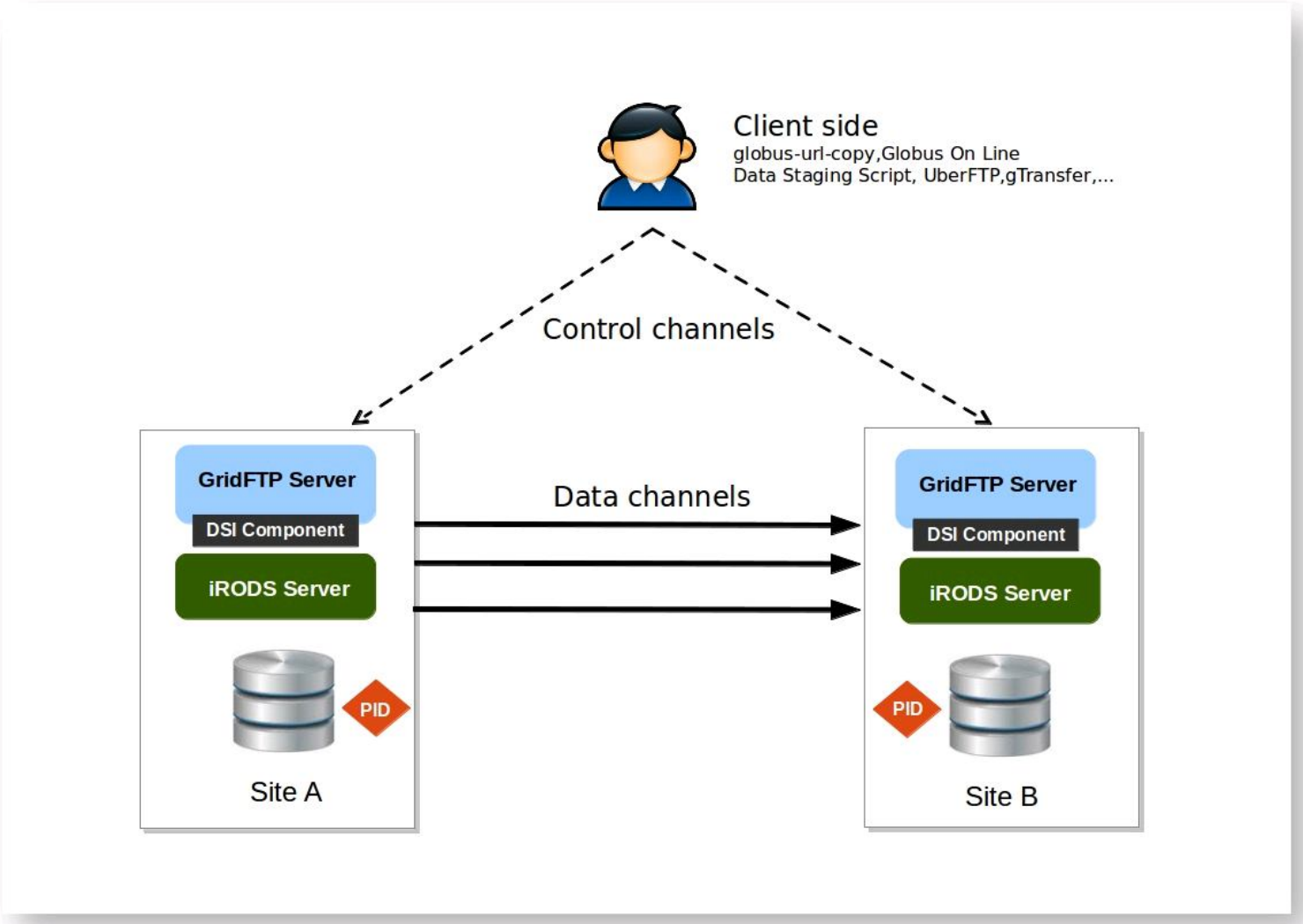
How it works

- **Server side**
  - the data staging functionality is realized by extending the **iRODS system with a GridFTP interface (Data Storage Interface - DSI)** so to permit the transfer of data through a reliable, high-performance protocol.

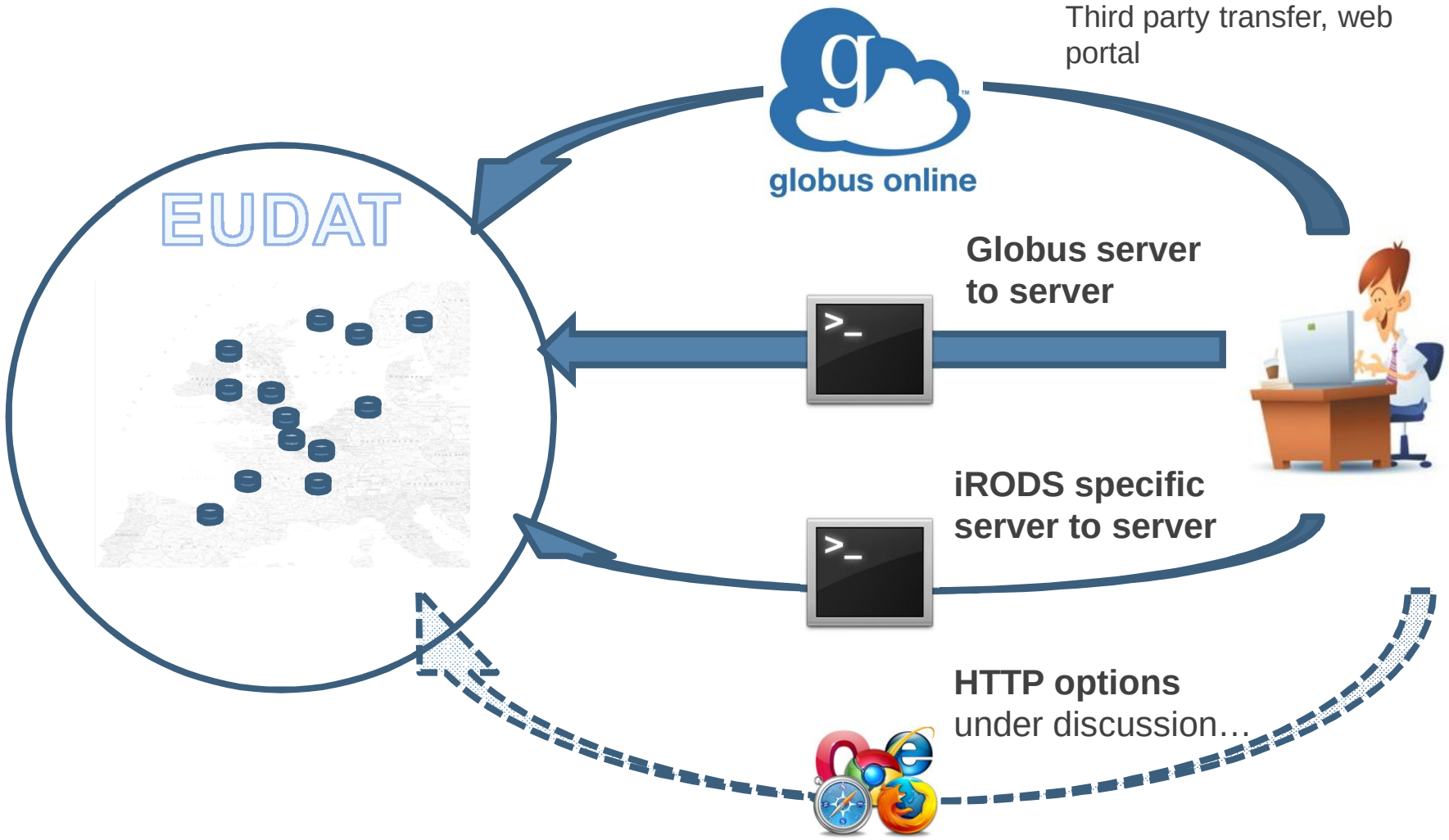
- **Client side**
  - any existing client, supporting the GridFTP protocol can be employed – globus-url-copy, Globus On Line, UberFTP, gTransfer, etc.

- Users need a personal certificate (X.509) to fully exploit the service

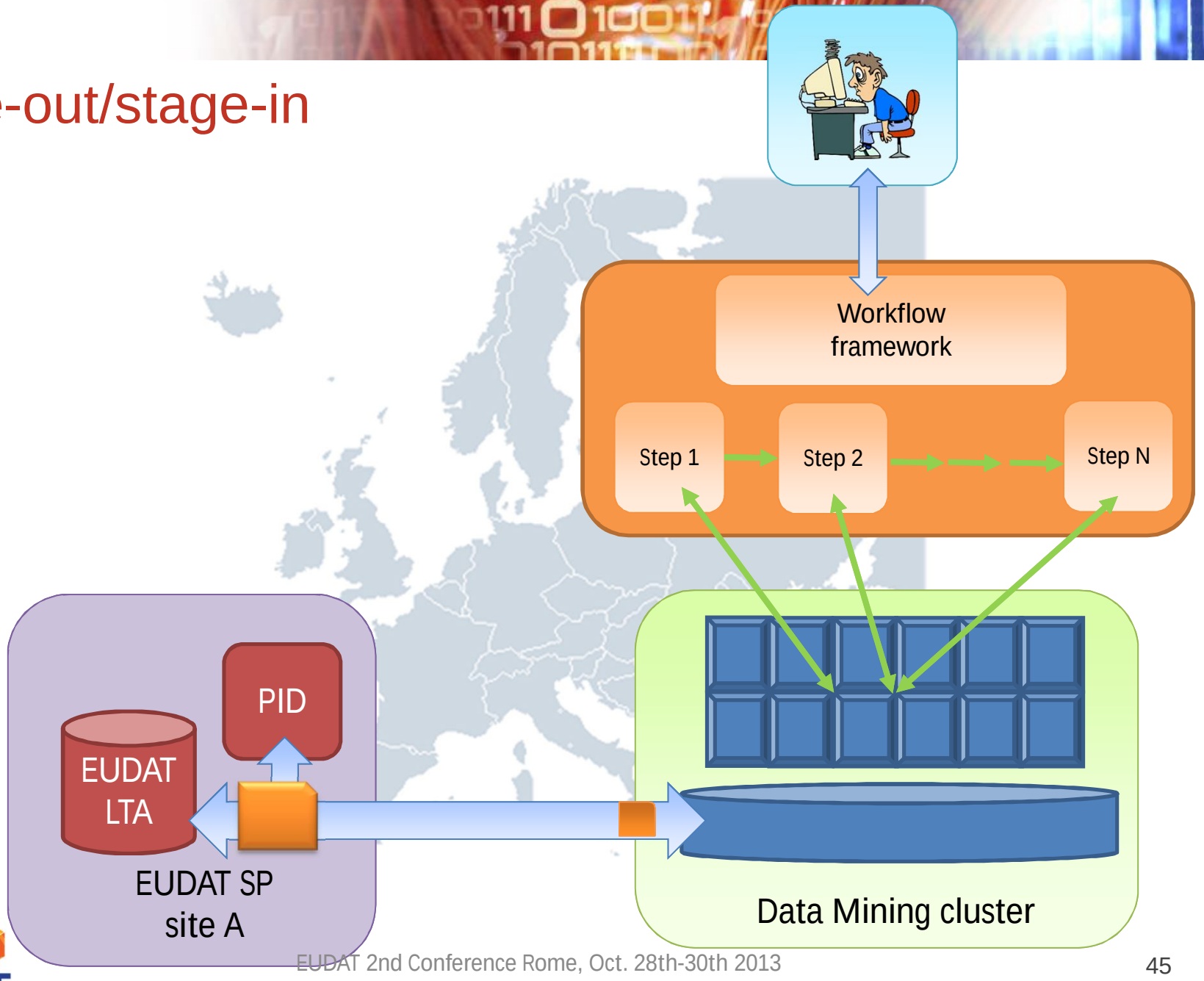
GridFTP: third Party Transfer



Eudat Transfer options



Stage-out/stage-in



Data Staging Script

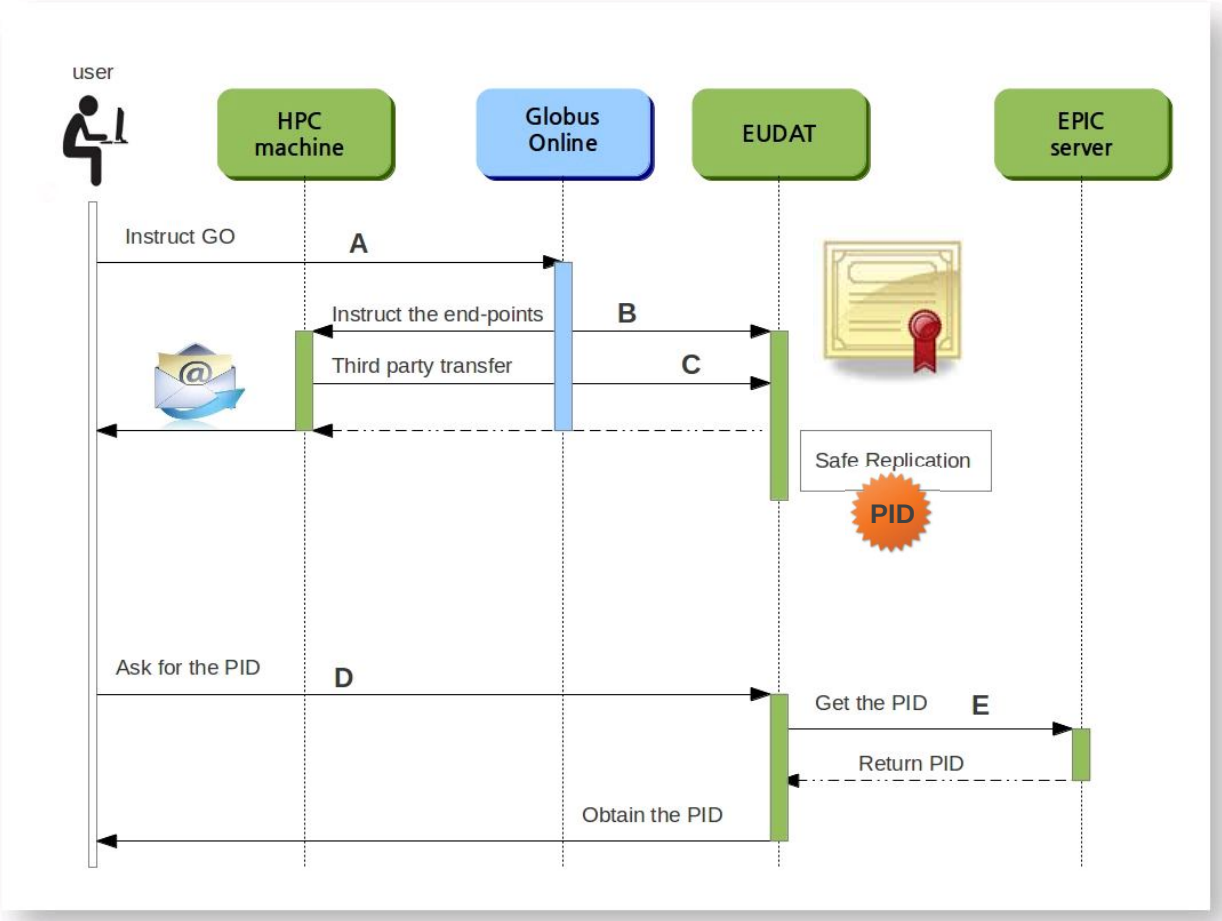
- A simple python modular staging script to help communities integrate the data staging service within their exiting solutions
- Based on Globus Online API and iRODS rule mechanism for data selection

```
./datastager.py <what to do> -u <username> --ss  <HPCmachine> --sd <UnixPath> -p <filename>
                  --ds <EUDAT-NODE> --dd <iRODS-Path>
```

```
./datastager.py in taskid -u <USER> --ss <SRC_SITE> --sd <SRC_DIR> -p <PATH> \\
                  --ds <DST_SITE> --dd <DST_DIR>
```

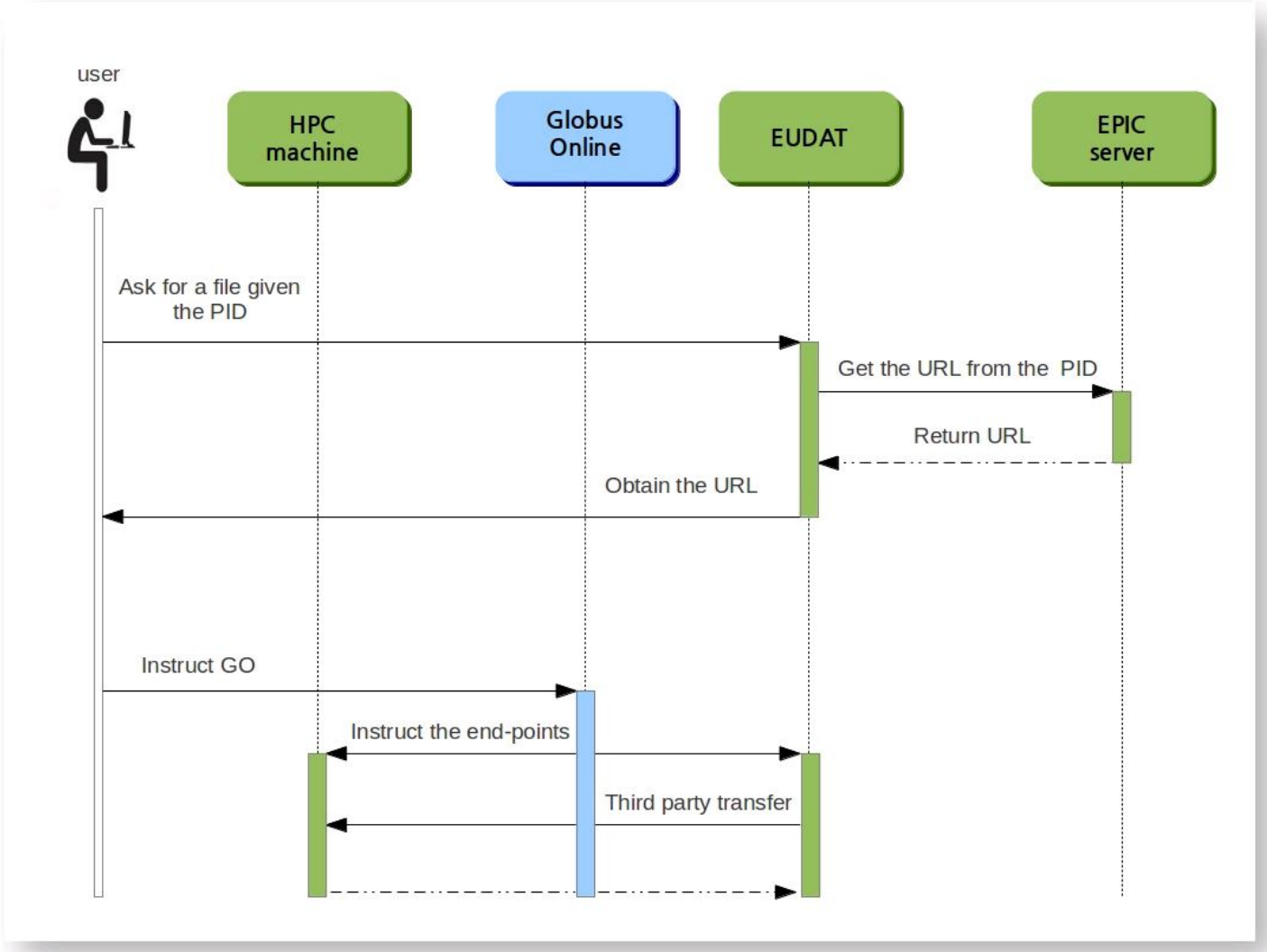
```
./datastager.py out pid --pid <PID> -u <USER> --ds <DST_SITE> --dd <DST_DIR>
```

Data Staging Script: Stage-in

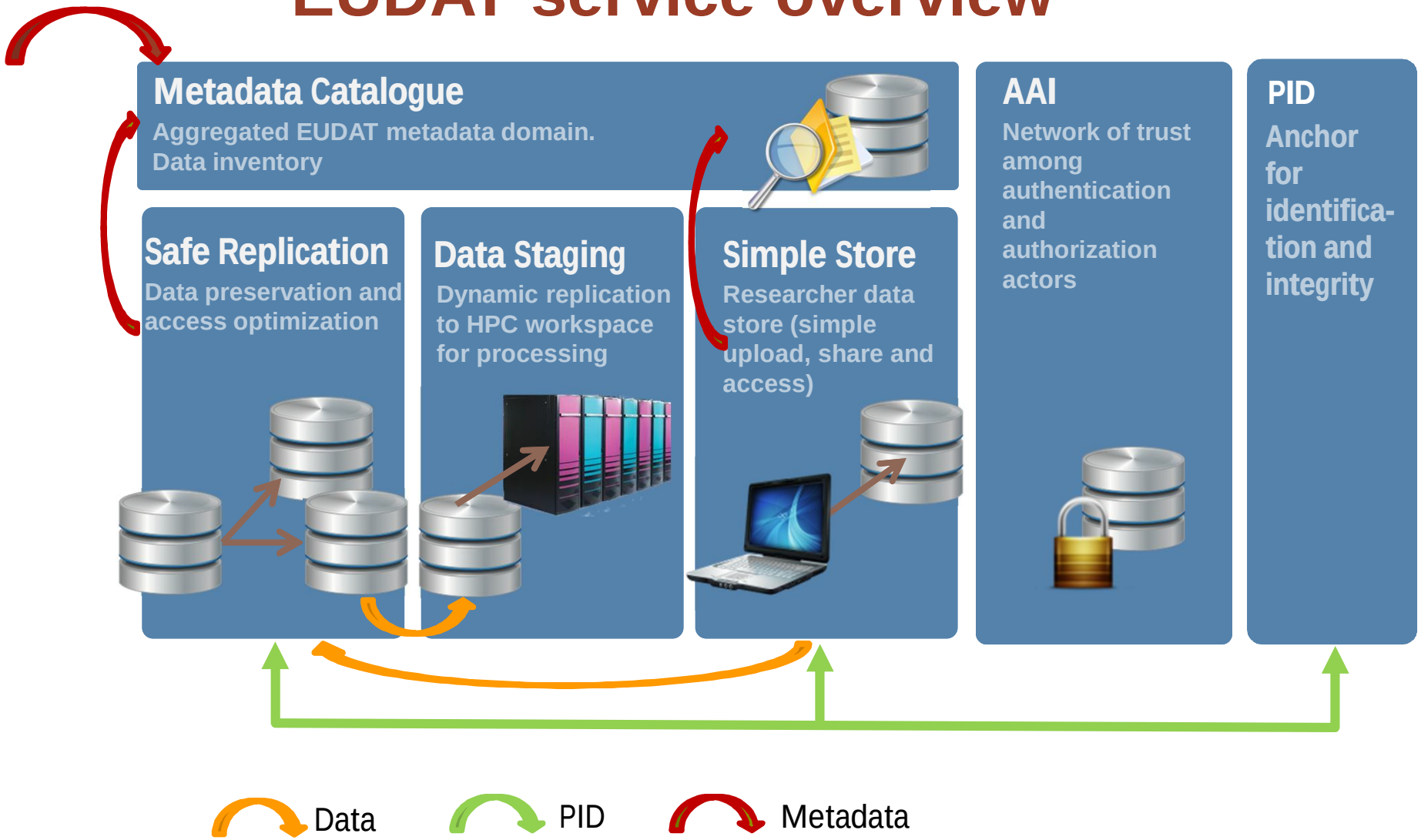


1. DSS contact GO via API (**A**)
2. GO contacts endpoints and activate them (**B**)
3. DSS gives to GO the list of files
4. A third party transfer is executed between endpoints (**C**)
5. User recive email when transfer is finished
6. User can retrive PID for furher processing (**D,E**)

Data Staging Script: Stage-out



EUDAT service overview



Conclusions

- The way to move data has to be enough **flexible** to accomodate different transfer protocols, different access mechanisms.
- Flexibility means also that the transfer tools can be used as they are, with default parameters, for average performances, but also **fine tuned** by experts **for faster transfers**.
- No solution fits all, so different services are provided

Thank you!

http://eudat.eu/safe-replication | eudat-safereplication@postit.csc.fi

http://eudat.eu/datastaging | eudat-datastaging@postit.csc.fi