EUROPEAN MIDDLEWARE INITIATIVE

EMI Data, dCache.org and starndards

Patrick Fuhrmann (DESY)

EMI Data Area lead

The EMI – data team / credits

- Alejandro Alvarez
- Alex Sim
- Claudio Cacciari
- Christian Loeschen
- Dirk Duellmann
- Elisabetta Ronchieri
- Fabrizio Furano
- Giuseppe Fiameni
- Giacinto Donvito
- Giuseppe Lo Presti
- Jon Kerr Nilsen
- Jan Schaefer
- Jean-Philippe Baud
- Michele Carpene
- Michele Dibenedetto
- Michail Salichos
- Mischa Salle
- Oscar Koeroo
- Oliver Keeble
- Paul Millar
- Ralph Mueller-Pfefferkorn
- Ricardo Rocha
- Riccardo Zappi
- Tigran Mkrtchyan
- Zsolt Molnar
- Zsombor Nagy

Our wiki : https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T3Data

Outline

The European Middleware Initiative within the FP7 Framework

- EMI in the European FP7 context.
- What is EMI doing ?
- Why are we doing this ?
- ***EMI Data*** in the EMI context.
- When are we doing what ?
- What is ***EMI Data*** doing in particular ?

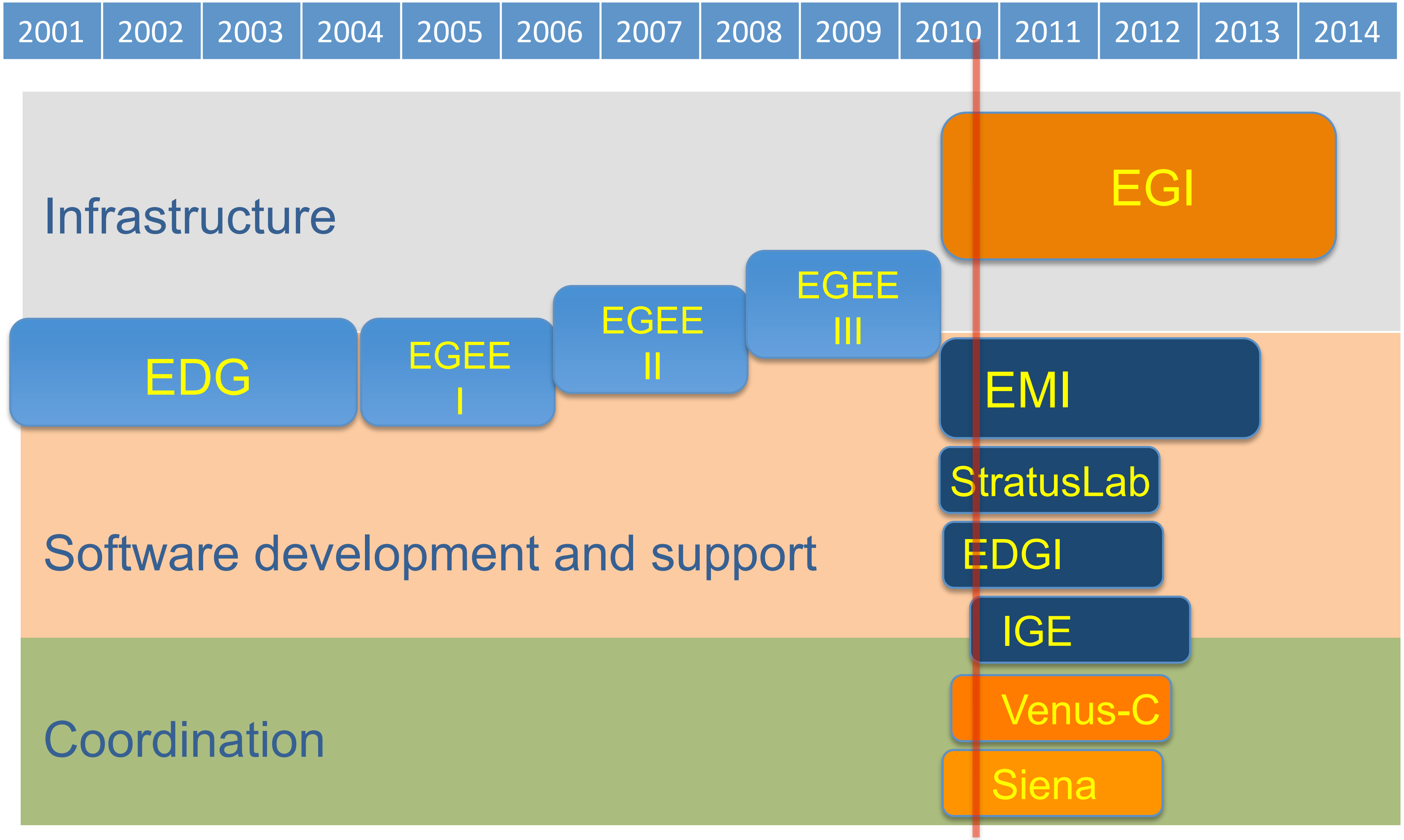
dCache.org and EMI

- dCache in a nutshell
- dCache in use.

Standardization

- SRM, spec plus security protocol
- WebDav
- NFS 4.1

The last Decade in Europe (HTC)



Project details

StratusLab

Statuslab.eu

StratusLab is developing and deploying cloud technologies with the aim of simplifying and optimizing the use and operation of distributed computing infrastructures such as the European Grid Infrastructure (EGI).
The StratusLab Toolkit will integrate cloud and virtualization technologies and services within grid sites and enrich existing computing infrastructures with “Infrastructure as a Service” (IaaS) provisioning paradigms.

VENUS-C

Venus-c.eu

VENUS-C is focused on a reliable, industry-quality, sustainable platform: letting scientists be scientists and supporting small & medium enterprises.

SIENA

sienainitiative.eu

SIENA will support Europe’s Distributed Computing Infrastructure (DCI) initiatives and the European Commission in working towards the delivery of a future e-Infrastructures roadmap that will be aligned with the needs of European and national initiatives.

EDGI

Edgi-project.eu

Desktop Grids : EDGI will develop DG-Cloud bridge middleware with the goal to get instantly available additional resources for DG systems if the application has some QoS requirements that could not be satisfied by the available resources of the DG system.

IGE

Edgi-project.eu

IGE wants to knit a tight European network between the European Globus developers and users, thus ensuring a fast response time to European user requests and the provision of up-to-date information to the European developers of the European user requirements.

The European Grid Infrastructure

egi
European Grid Infrastructure
Towards a sustainable grid infrastructure

EGI.eu coordinates the European Grid Infrastructure with National Grid Initiatives, European International Research Organizations and other parties, to provide a generic e-infrastructure for all European researchers.

The European Middleware Initiative

According to our Project Director, Alberto Di Meglio :

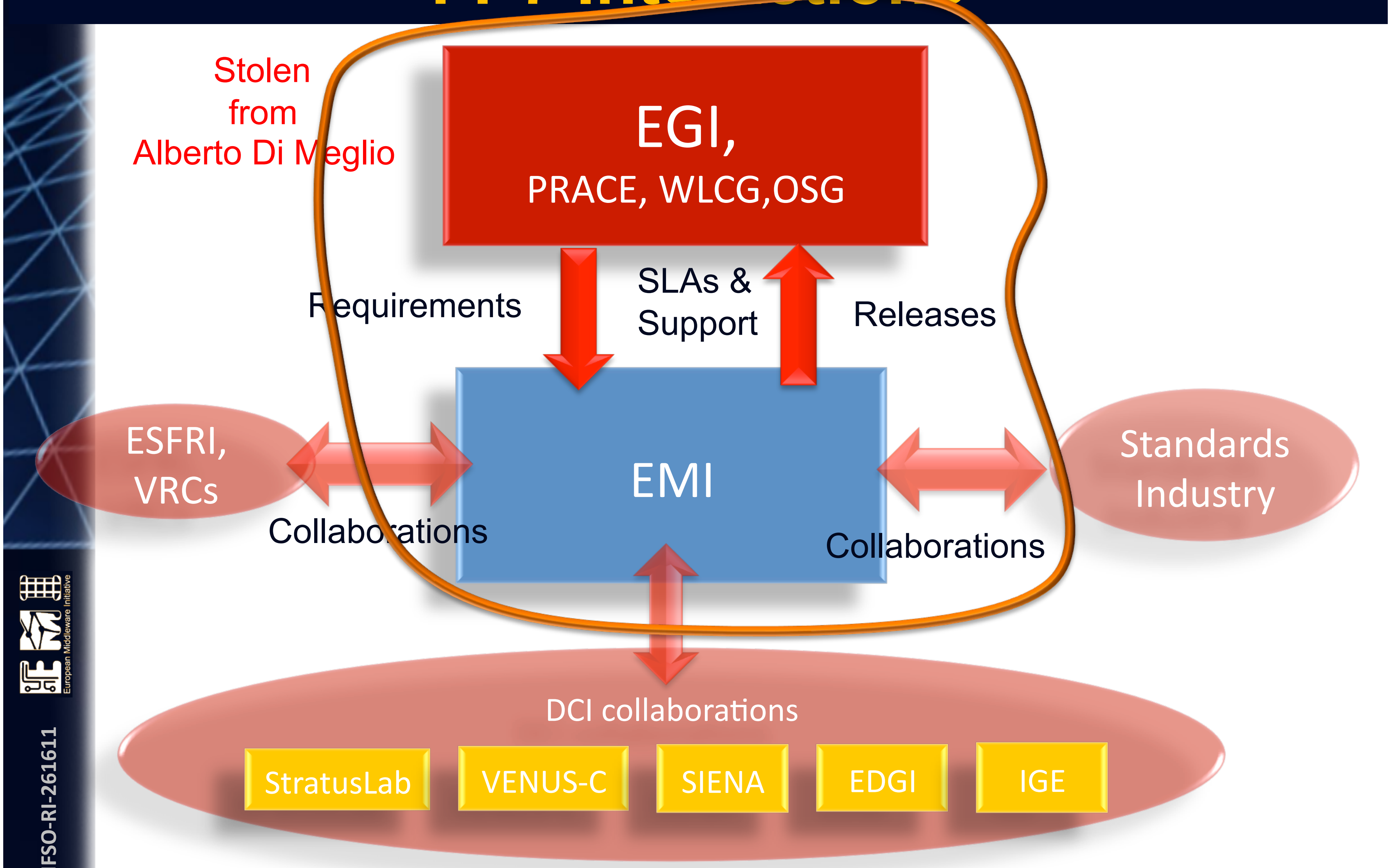
The European Middleware Initiative (EMI) project represents a close collaboration of the major European middleware providers - ARC, gLite, UNICORE and dCache - to establish a sustainable model to support, harmonise and evolve distributed computing middleware for deployment in EGI, PRACE and other distributed e-Infrastructures (DCI's)

Interactions

How this all works together

FP7 Interactions

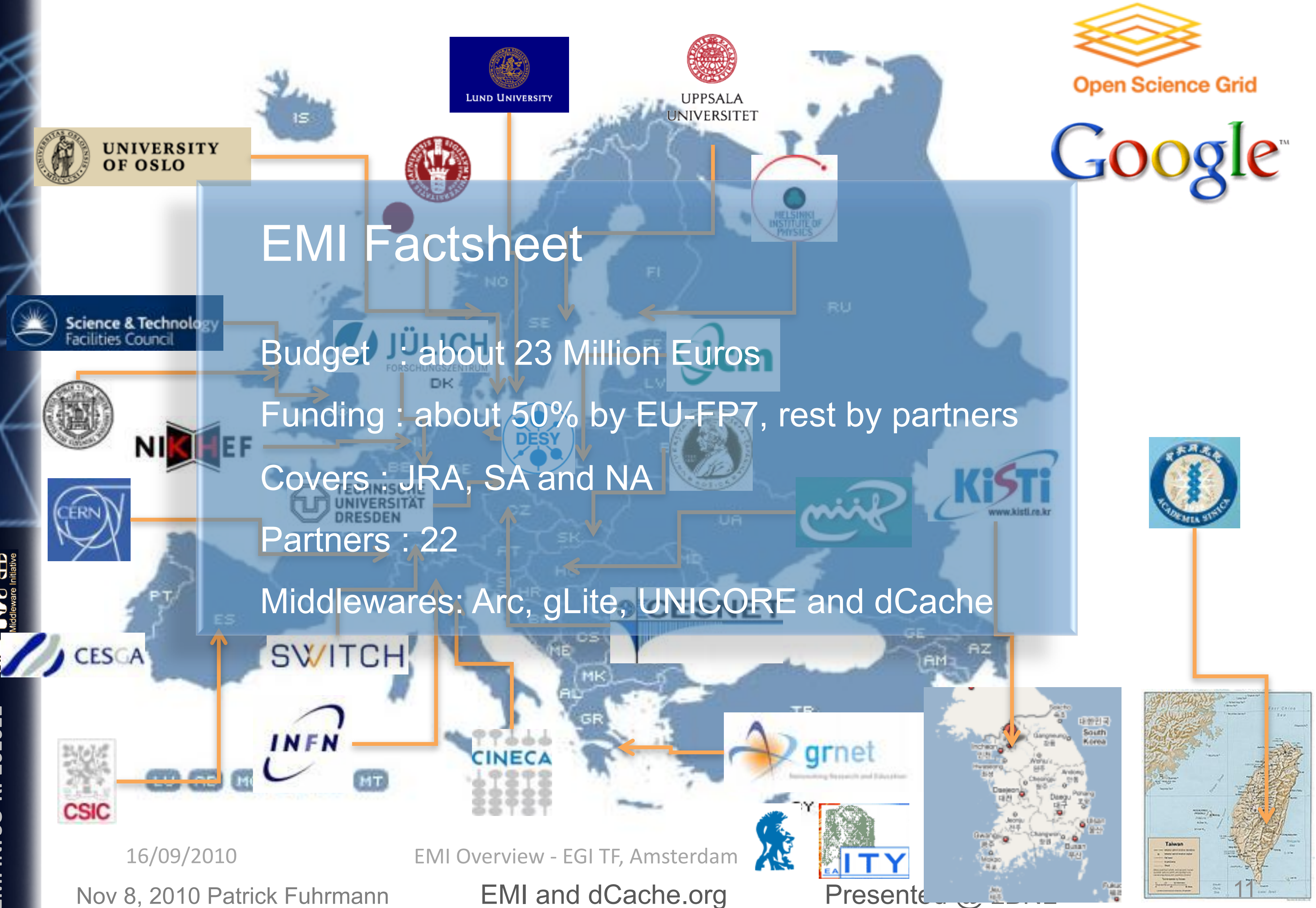
Stolen
from
Alberto Di Meglio



EMI

Now about EMI

EMI Factsheet



EMI Factsheet

Budget : about 23 Million Euros

Funding : about 50% by EU-FP7, rest by partners

Covers : JRA, SA and NA

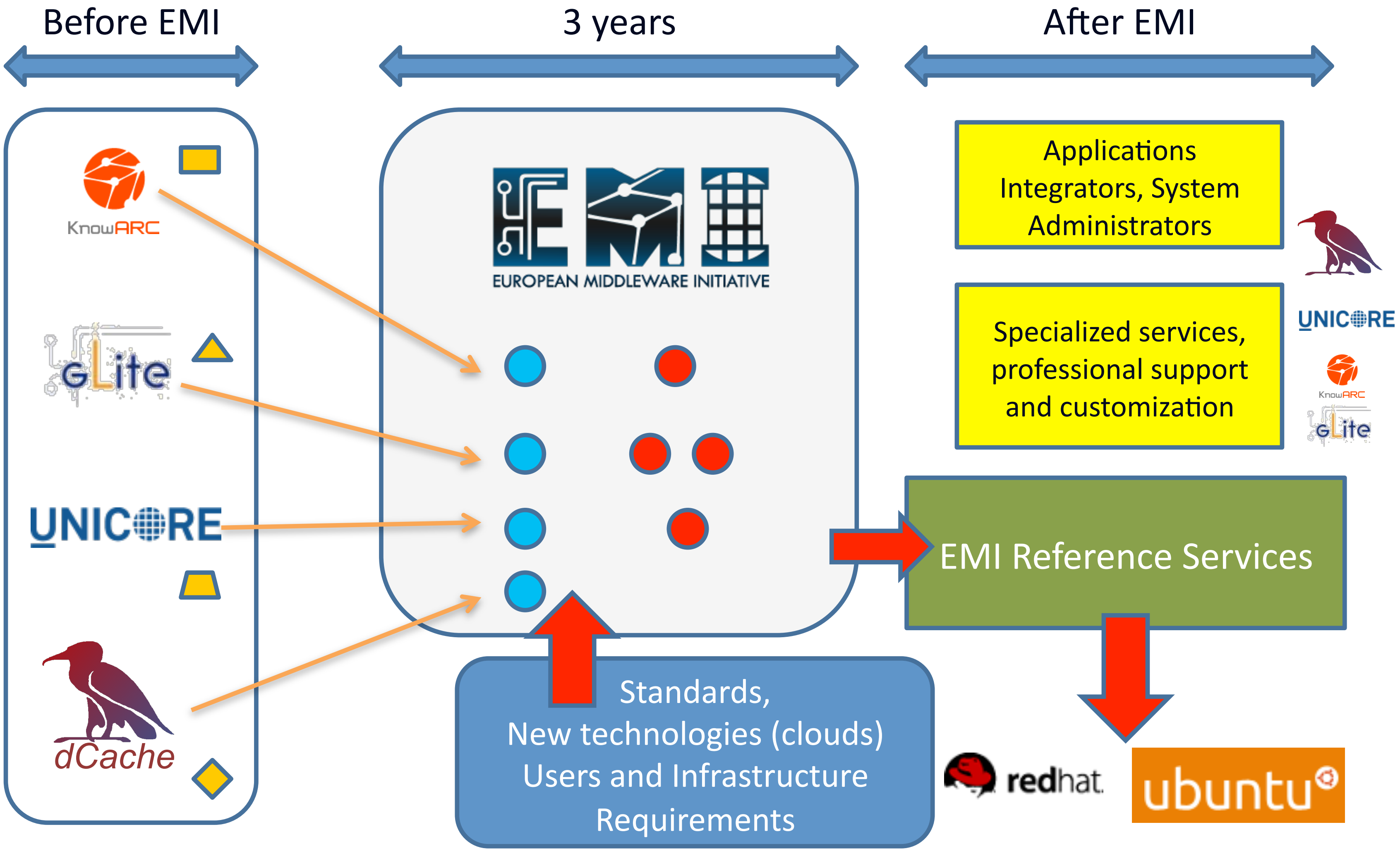
Partners : 22

Middlewares: Arc, gLite, UNICORE and dCache

What is EMI doing

Stolen from Alberto Di Meglio

EMI Middleware Evolution


Before EMI
3 years
After EMI
KnowARC
gLite
UNICORE
dCache
EUROPEAN MIDDLEWARE INITIATIVE
Applications Integrators, System Administrators
Specialized services, professional support and customization
EMI Reference Services
Standards, New technologies (clouds) Users and Infrastructure Requirements
redhat
ubuntu

Why again ?

Why are WE doing this ?

Because with EMI we got the money and the organizational infrastructure to achieve goals, which we were planning to do anyway but didn't find time nor money yet, e.g. :

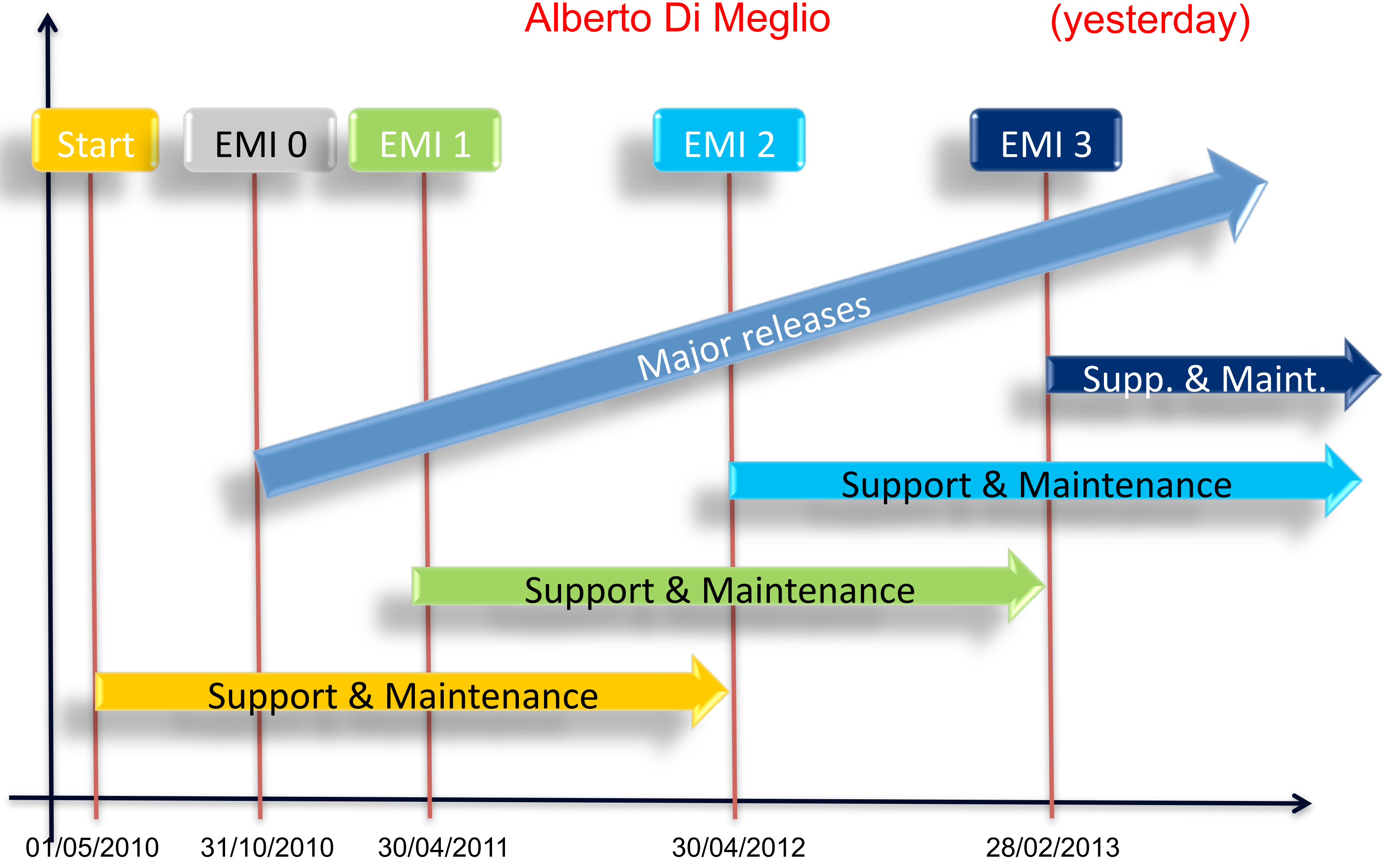
- ➢ Moving towards standards
  - ✓ https / webDav
  - ✓ NFS 4.1
  - ✓ SRM
- ➢Fixing flaws
  - ✓ Catalogue synchronization
- ➢Improving usability
  - ✓ Storage Accounting
  - ✓ Monitoring Interface
  - ✓ Individual efforts of product teams of components

When will it happen ?

Release Plan

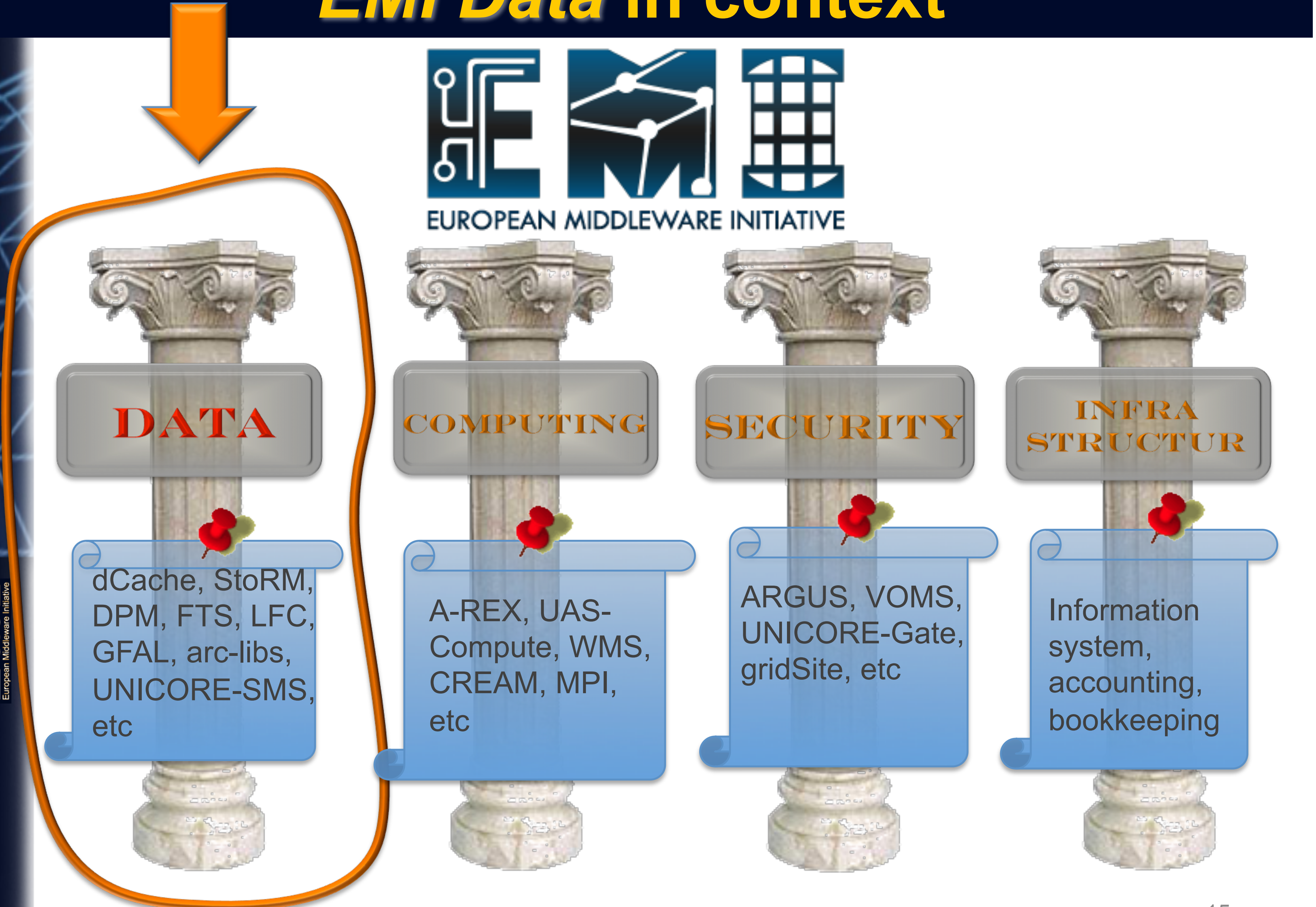
Stolen
from
Alberto Di Meglio

See Alberto Aimar's
presentation for details
(yesterday)



EMI Data in context

EUROPEAN MIDDLEWARE INITIATIVE



DATA

dCache, StoRM, DPM, FTS, LFC, GFAL, arc-libs, UNICORE-SMS, etc

COMPUTING

A-REX, UAS-Compute, WMS, CREAM, MPI, etc

SECURITY

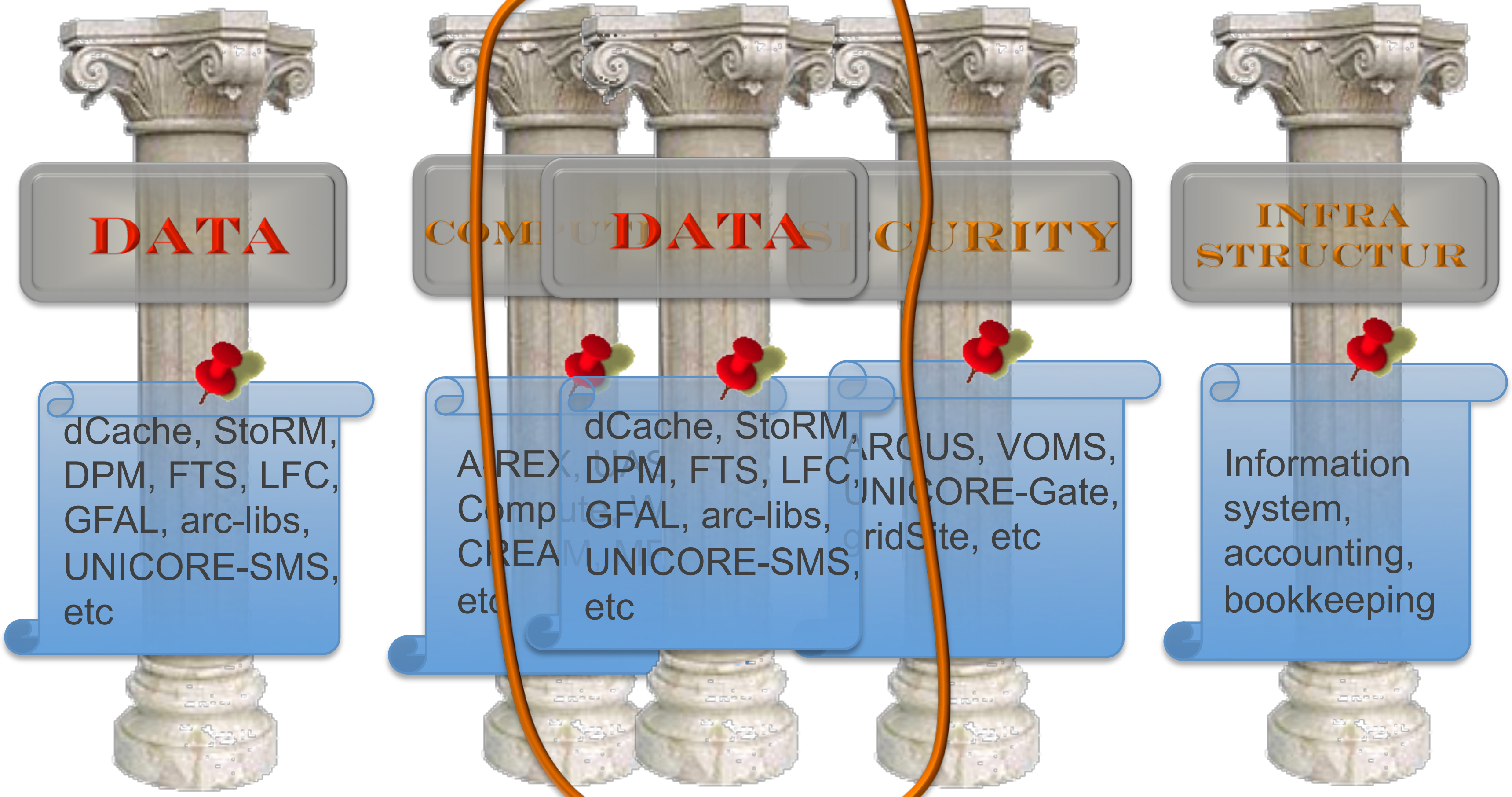
ARGUS, VOMS, UNICORE-Gate, gridSite, etc

INFRA STRUCTUR

Information system, accounting, bookkeeping

EMI Data in context

EUROPEAN MIDDLEWARE INITIATIVE



DATA

dCache, StoRM, DPM, FTS, LFC, GFAL, arc-libs, UNICORE-SMS, etc

COM

A-REX

Comp

CREA

etc

DATA

dCache, StoRM, DPM, FTS, LFC, GFAL, arc-libs, UNICORE-SMS, etc

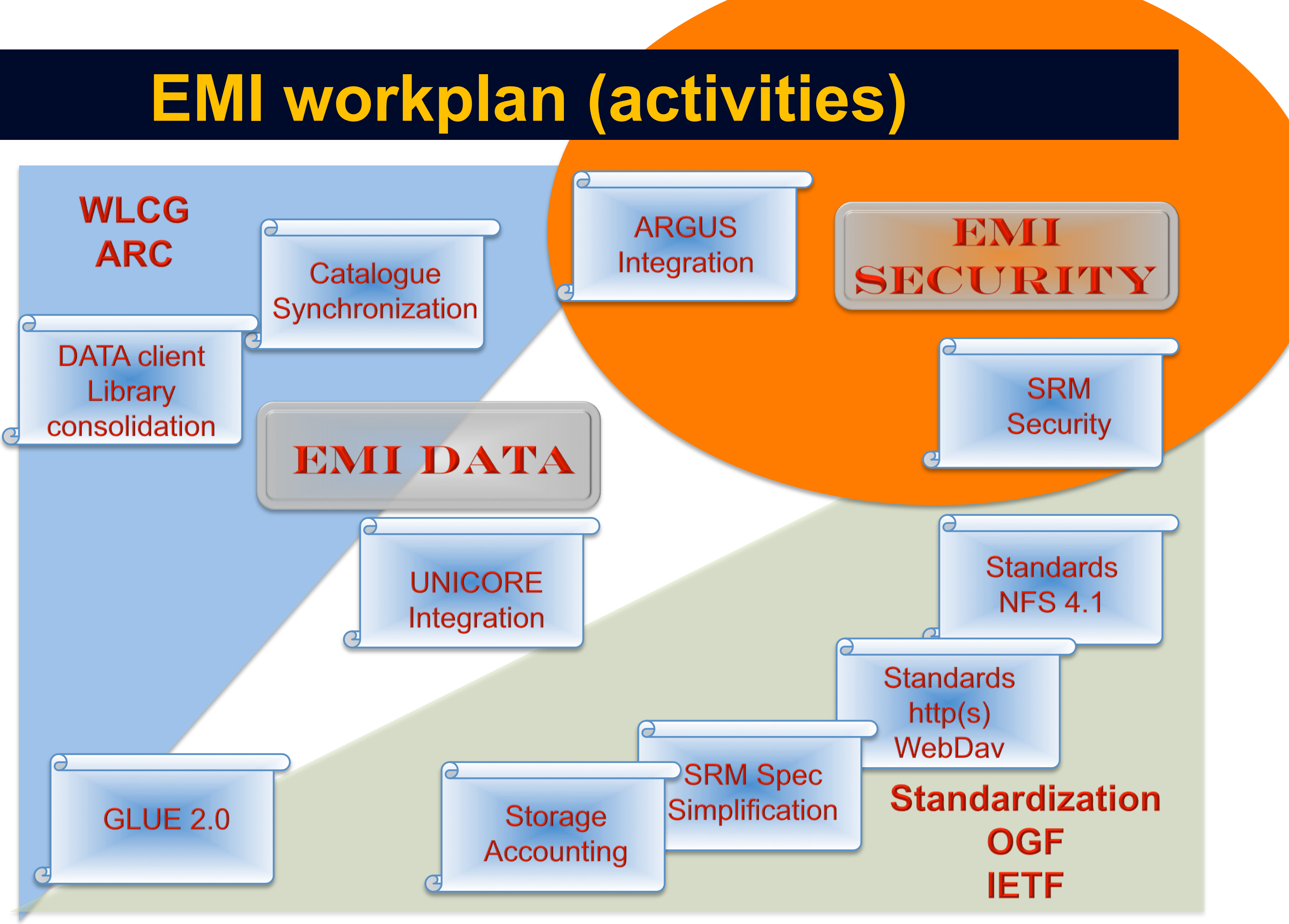
CURITY

ARGUS, VOMS, UNICORE-Gate, gridSite, etc

INFRA STRUCTUR

Information system, accounting, bookkeeping

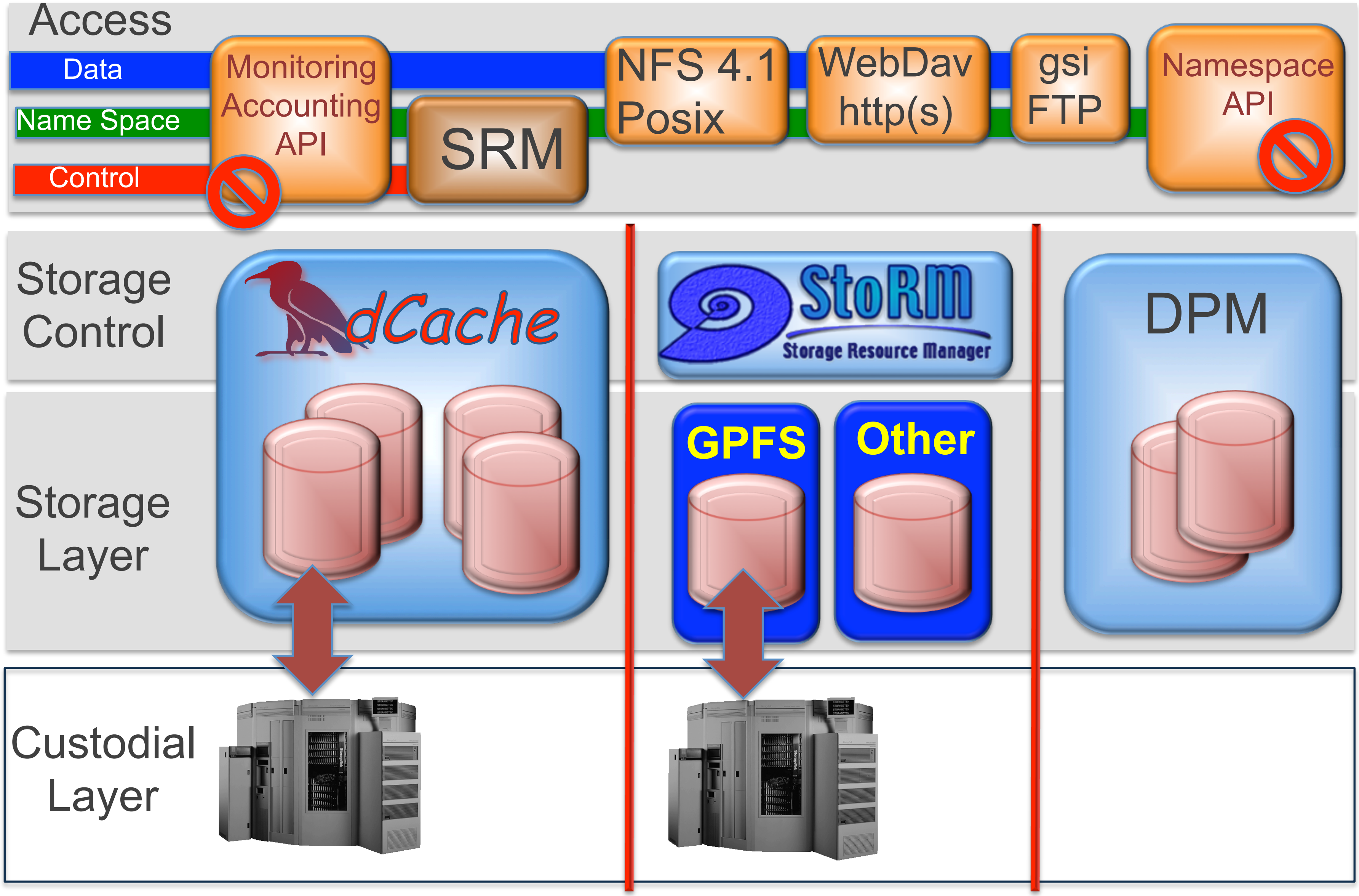
EMI workplan (activities)



Standardization

Standardization efforts within EMI

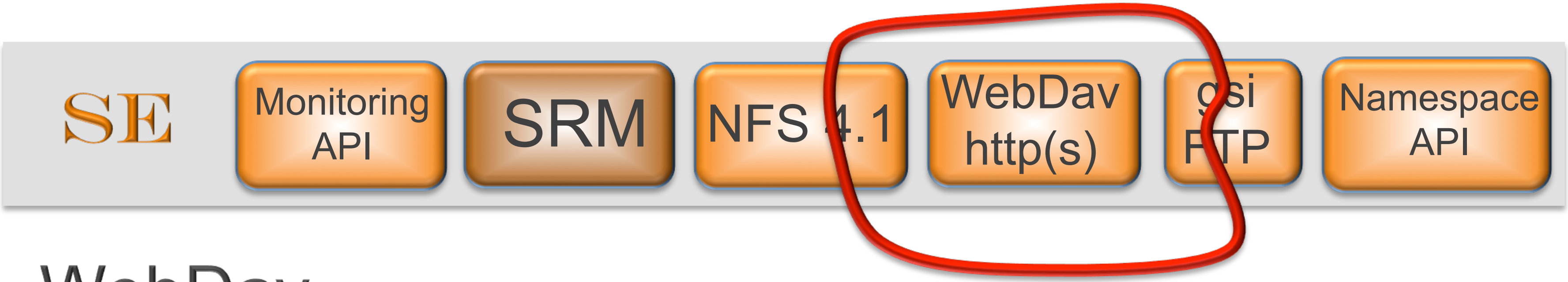
The EMI Storage Elements



Standardization

Standardization : the easy bit

Standardization : WebDav



SE

Monitoring API

SRM

NFS 4.1

WebDav http(s)

gsi FTP

Namespace API

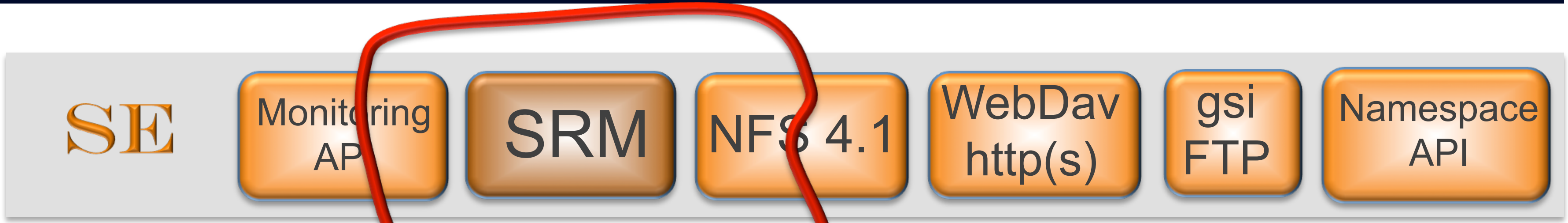
WebDav

- Very useful for new (non-LHC) communities.
- Already available in dCache.
- Will be added to StoRM and DPM after EMI-1.
- Allows “File system like” access with
  - Mac OS
  - Linux
  - Windows

Standardization

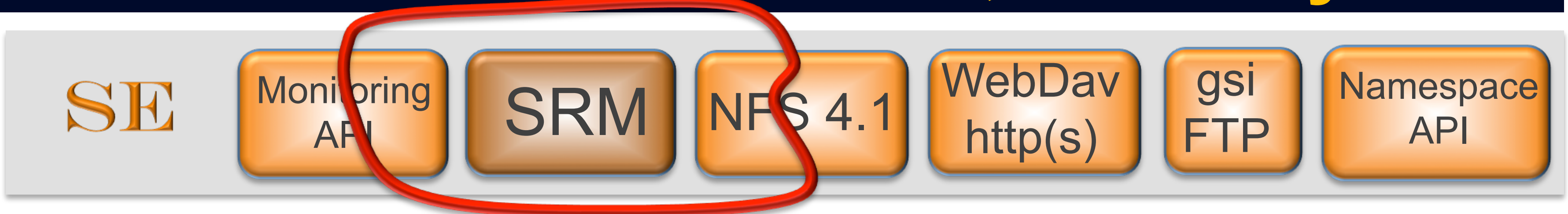
Standardization : fixing the missing bits

Standardization : SRM, specification



- SRM is a remote *storage management* protocol.
- The SRM does :
  - Transfer protocol negotiation
  - Name space operations
  - Space management
  - Storage Management : access latency, retention policy (tape, disk,…)
  - Allows bulk operations.
- Specification not easy to understand by customers.
- Spec might need a cleanup based on our experience.
- Better documentation from user perspective.
- The SRM is an extremely useful and btw the only tool to remotely manage data in a standardized way across SE's.

Standardization : SRM, security



SE

Monitoring API

SRM

NFS 4.1

WebDav http(s)

gsi FTP

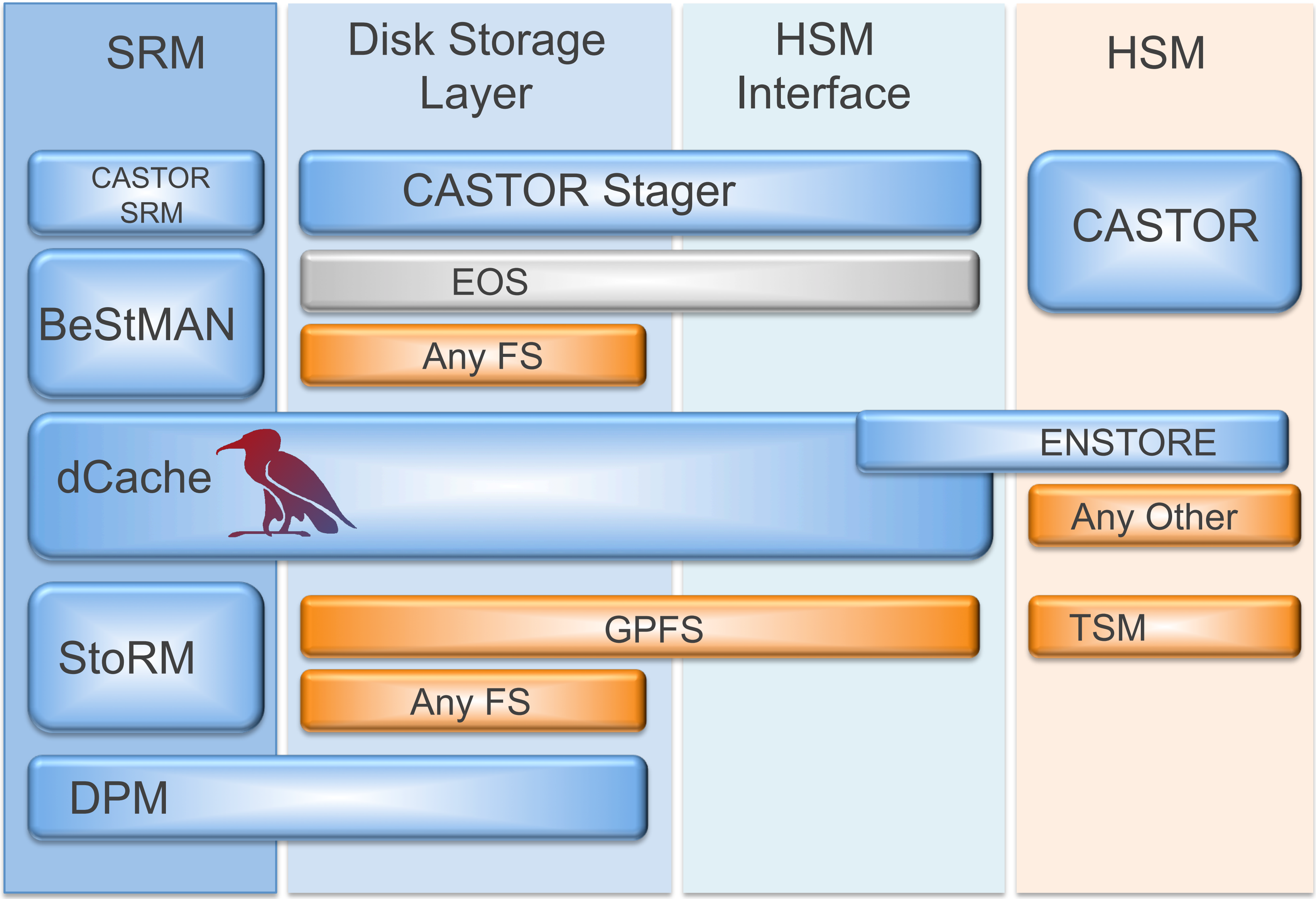
Namespace API

- Right now : GLOBUS : library and protocol (non standard)
- Goal : replacing GSI by SSL/TLS-X509
- Step I :
  - No delegation (srmcp)
  - GLOBUS library in SSL compatibility mode.
  - Prove of concept done : dCache SRM server and client.
- Step II
  - No delegation.
  - Server and client can use standard java/openssl libraries.
- Step III
  - Agreement on delegation service : done GDS
  - Agreements in progress ☺
    - Who tells to create delegated proxy : client or server
    - How does the server tell the client w/o changing the WSDL
    - Where do we store the delegation ID (w/o WSDL change)
    - How close should the delegation service be to the SRM service

Wider agreement necessary

However, things are slightly more complicated because …

The big picture



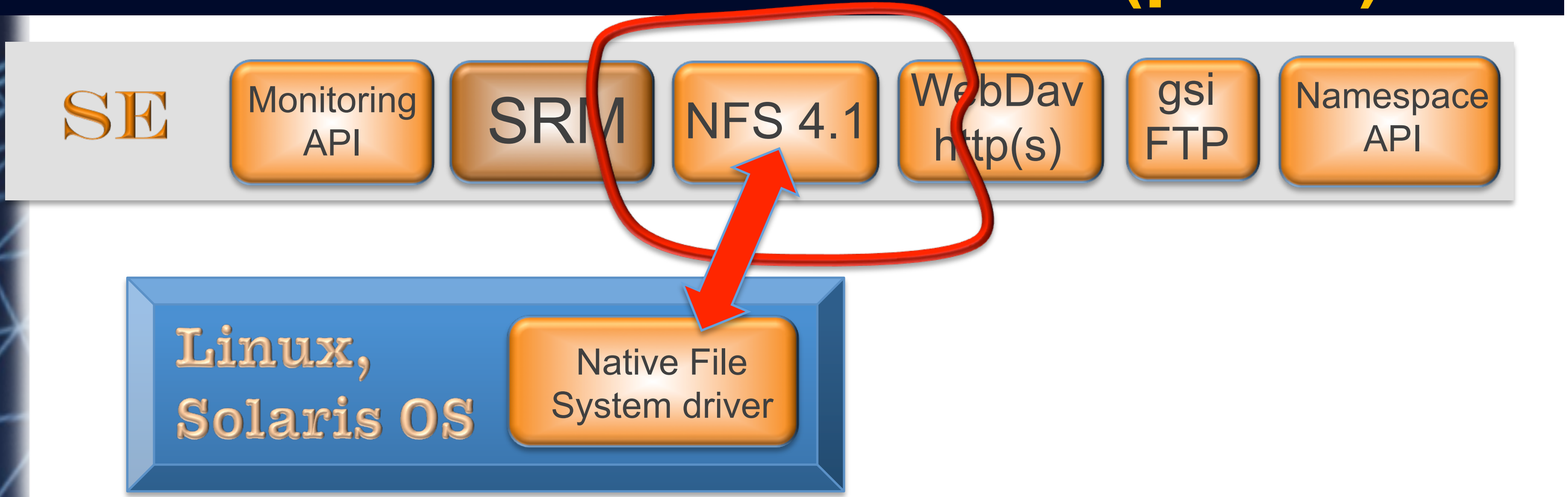
Wider agreement

All agreements, concerning the SRM security and the SRM specification cleanup, have to be coordinated with Alex (BeStMAN) and people from CASTOR.

Standardization

Standardization : the tough part

Standardization : NFS 4.1 (pNFS)



- NFS 4.1(pNFS) : industry standard (defined by IETF)
- Genuine POSIX access through mounted file system.
- pNFS supports highly distributed data sources.
- Clients provided and maintained by OS.
- Will be used by industry heavyweights : IBM, EMC, Panasas…
- Production dCache 1.9.10

The NFS 4.1 Initiative

Funding

dCache.org

DESY

HELMHOLTZ
ASSOCIATION

European Middleware Initiative

LCG

Small Distortion

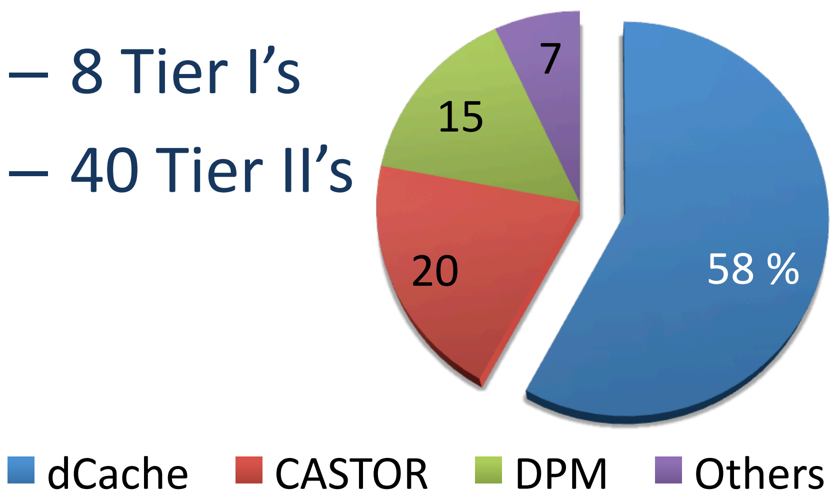
In order to understand why dCache is so keen on NFS 4.1 we need to understand a bit more about dCache.

(Shameless product placement ☺)

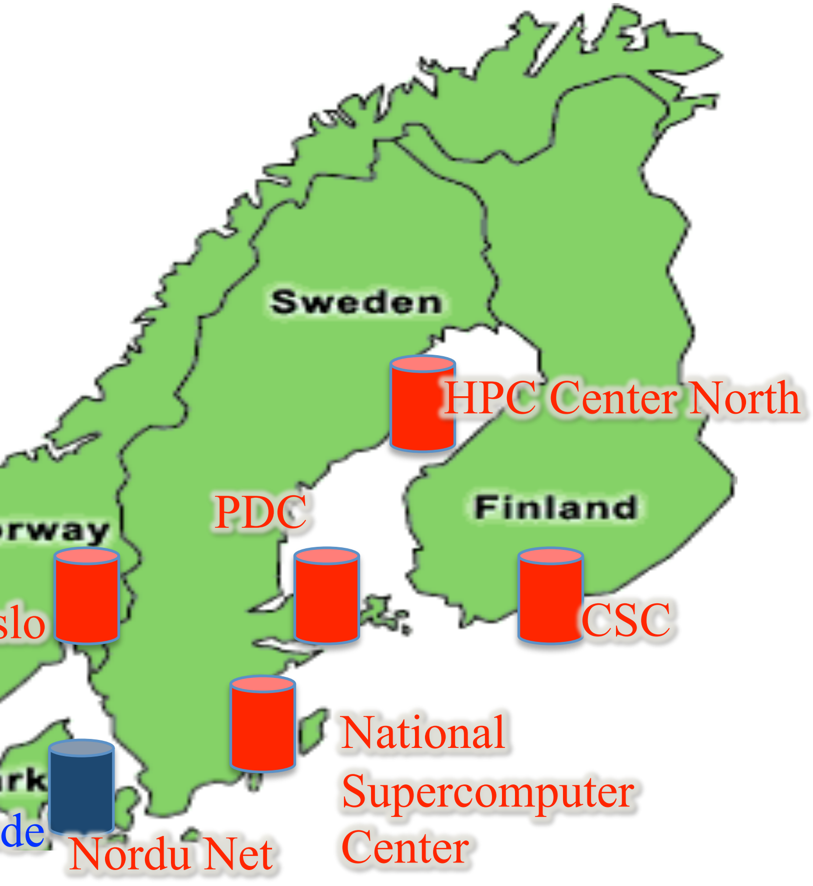
dCache in use

- WLCG
  - 8 Tier I's
  - 40 Tier II's

Percentage Data Stored



One dCache in the Nordics



- Academic storage network in Sweden
- LOFAR, European radio antenna
- Lot's of groups at DESY and FERMI

And what does this mean

Native file system extremely useful for WLCG analysis

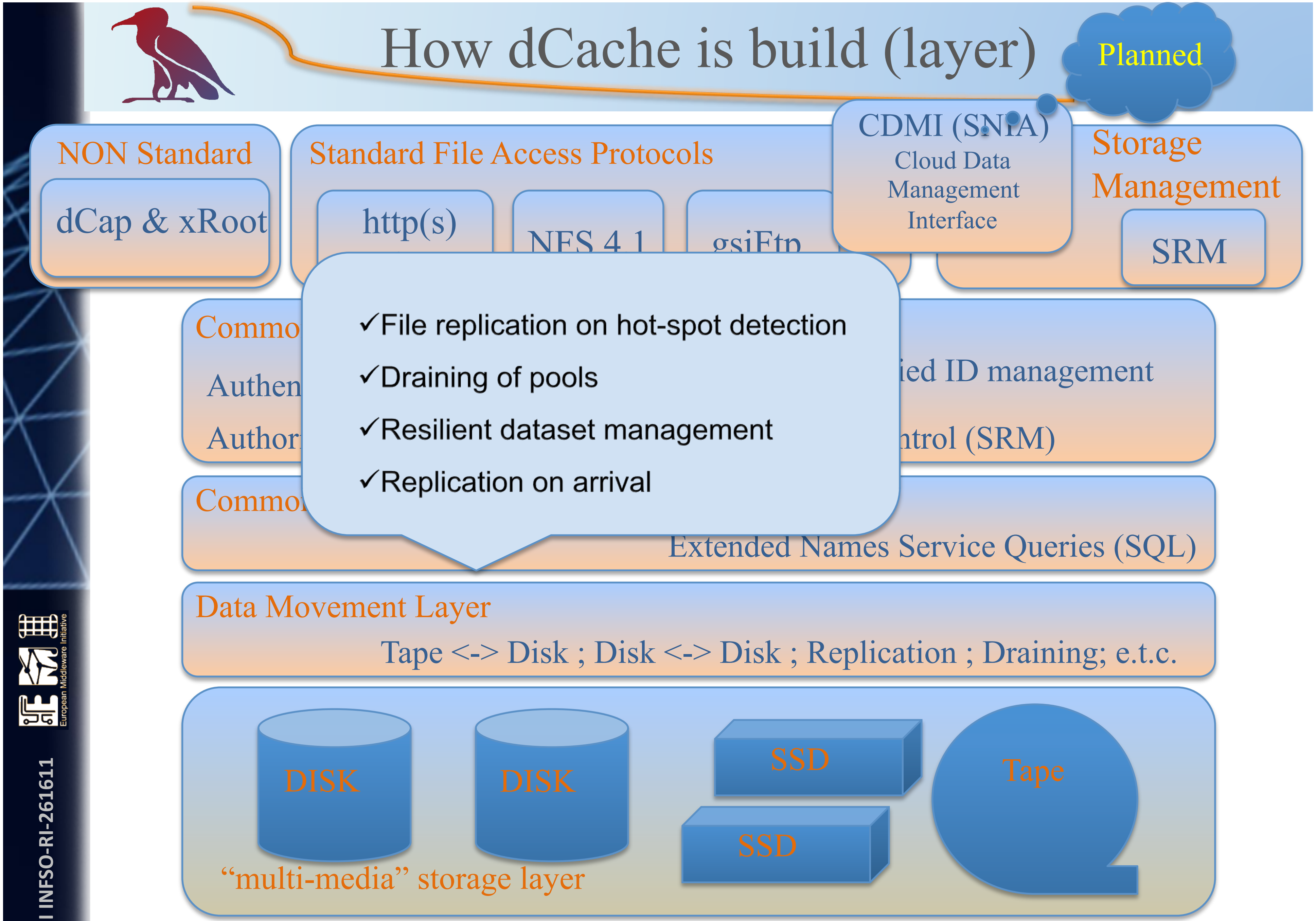
dCache supports a lot of communities for which direct file system access is essential.

Open '/foo/filename' is the only way they know to open a file.

Two slides on how dCache works

(more product placement)

How dCache is build (layer)



Planned

NON Standard

dCap & xRoot

Standard File Access Protocols

http(s)

NFS 4.1

gsiFtp

CDMI (SNIA)

Cloud Data Management Interface

Storage Management

SRM

Commo

Authen

Author

ied ID management

ntrol (SRM)

- ✓File replication on hot-spot detection
- ✓Draining of pools
- ✓Resilient dataset management
- ✓Replication on arrival

Commo

Extended Names Service Queries (SQL)

Data Movement Layer

Tape <-> Disk ; Disk <-> Disk ; Replication ; Draining; e.t.c.

DISK

DISK

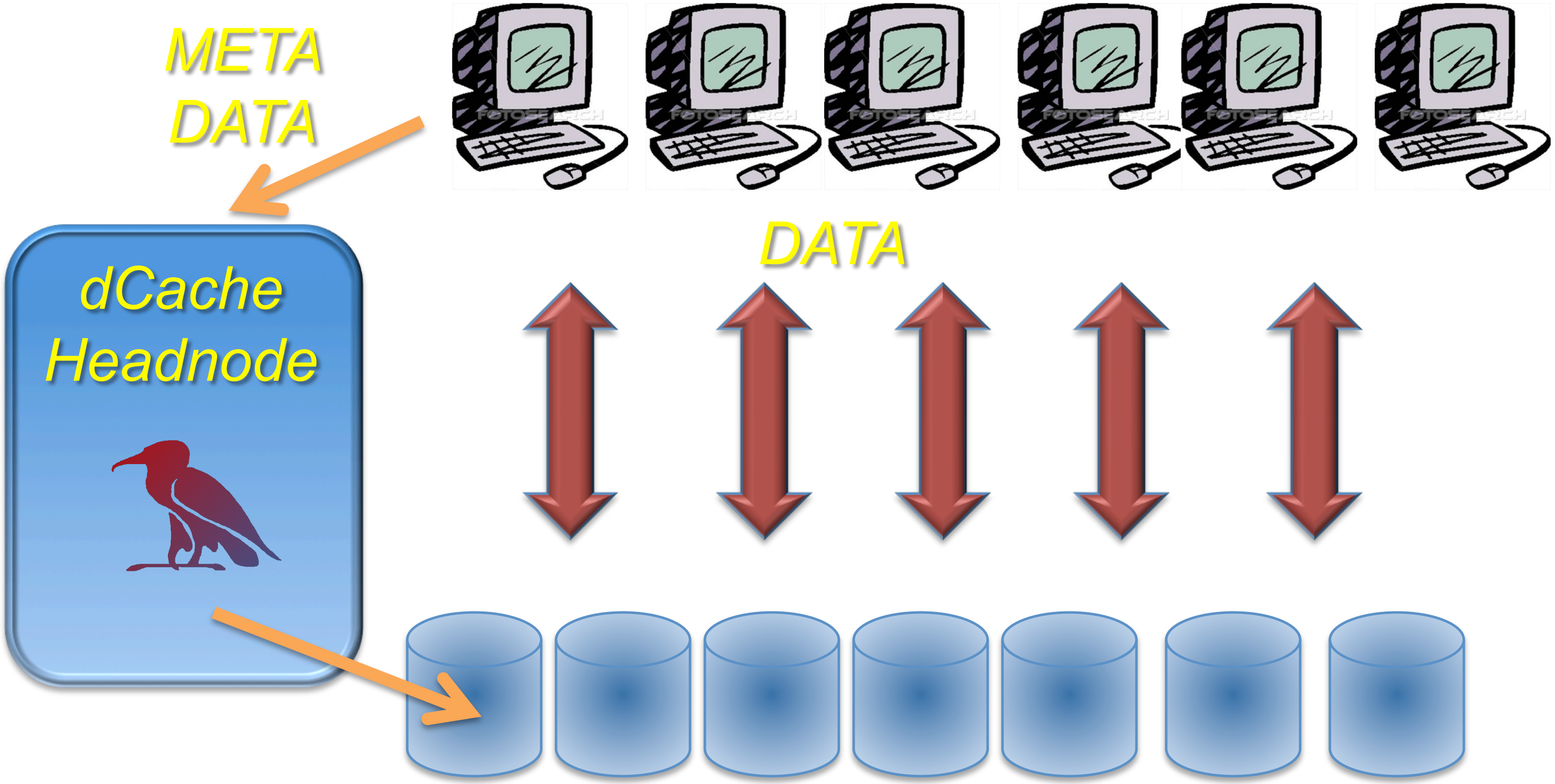
SSD

SSD

Tape

“multi-media” storage layer

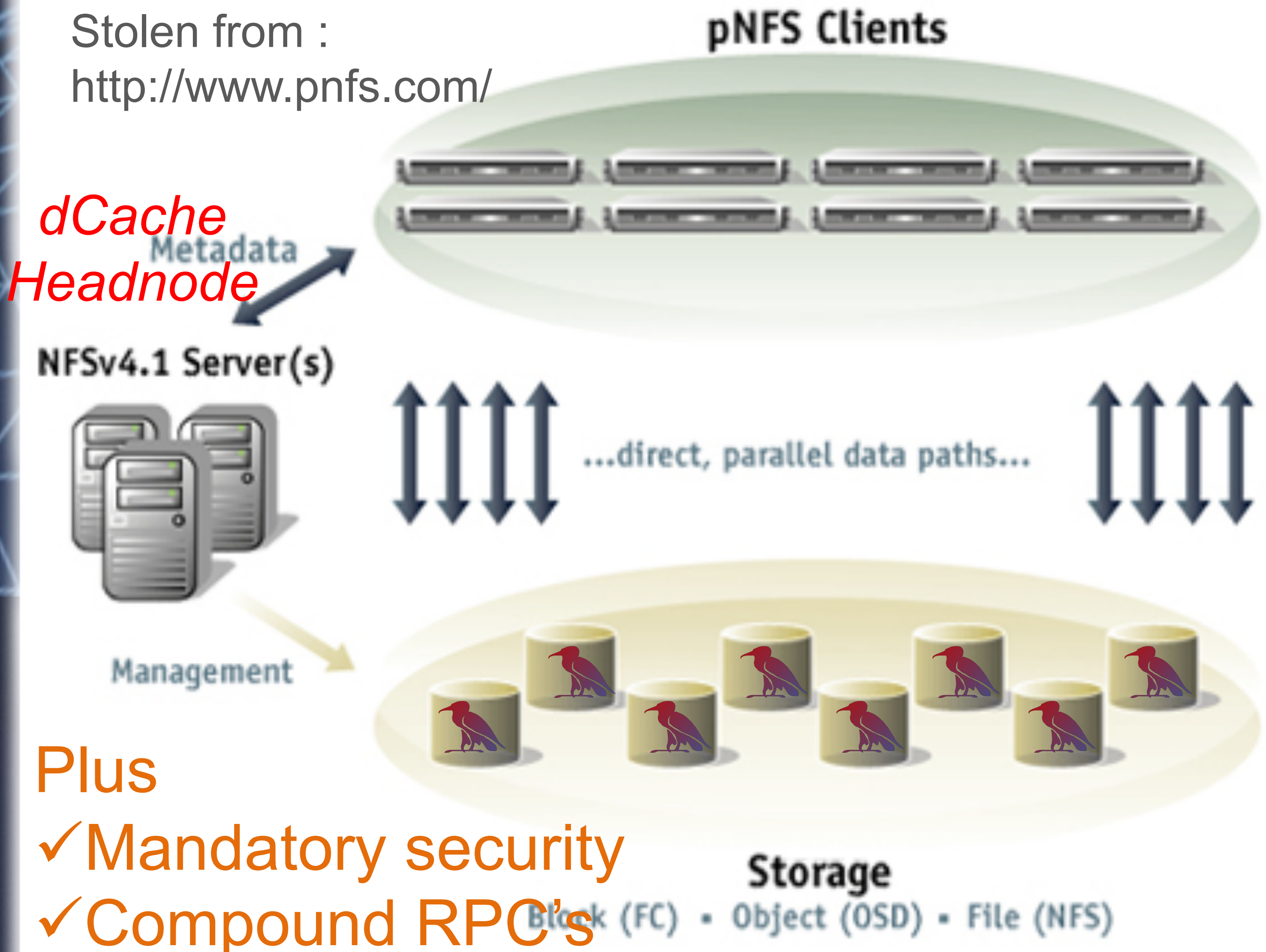
How dCache is build (data flow)



How is this related to NFS 4.1 ?

How does NFS 4.1 (pNFS) work ?

Stolen from :
http://www.pnfs.com/


pNFS Clients
Metadata
NFSv4.1 Server(s)
...direct, parallel data paths...
Management
Storage
Block (FC) • Object (OSD) • File (NFS)

dCache Headnode

Plus
✓Mandatory security
✓Compound RPC's

So NFS 4.1 (pNFS) fits perfectly into the dCache design.
It will benefit from all dCache features, like ACL's and automated file location management and it takes full advantage of the highly distributed way dCache works.

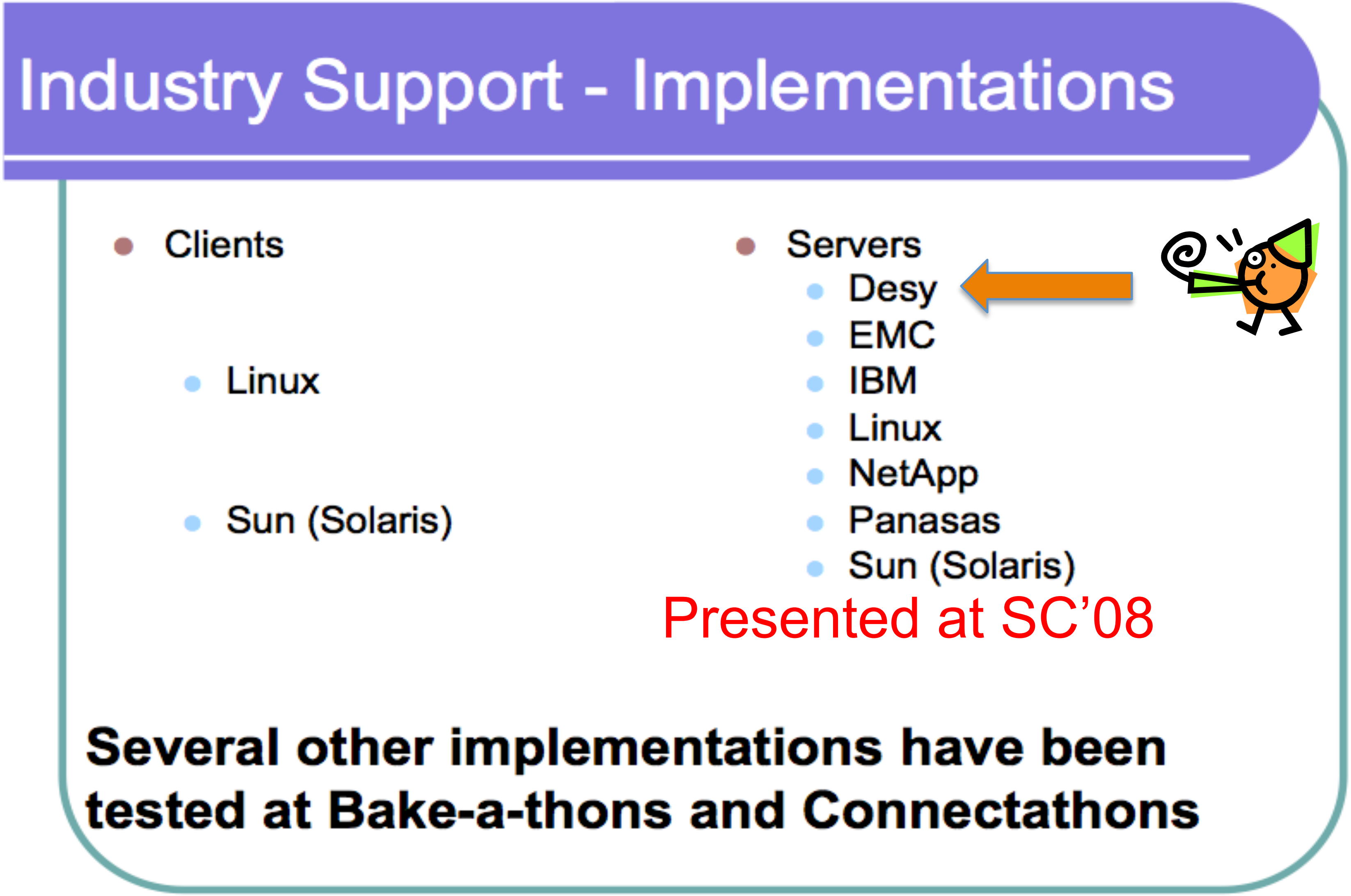
So what's the NFS 4.1 initiative ?

What is the NFS 4.1 initiative ?

- Industry initiative between all the major storage and OS vendors.
- Coordinated by CITI at the University of Michigan
- It is an WLCG demonstrator.
- Funded effort within the European Middleware Initiative
- Major effort in dCache
  - For non LCG communities
  - Hopefully for HEP as well

Who is behind NFS 4.1 (pNFS) ?

Stolen from : http://www.pnfs.com/



Industry Support - Implementations

- Clients
  - Linux
  - Sun (Solaris)
- Servers
  - Desy
  - EMC
  - IBM
  - Linux
  - NetApp
  - Panasas
  - Sun (Solaris)

Presented at SC’08

Several other implementations have been tested at Bake-a-thons and Connectathons

Why is industry interested ?

Stolen from : http://www.pnfs.com/

Benefits of Parallel I/O

- Delivers Very High Application Performance
- Allows for Massive Scalability without diminished performance

Benefits of NFS (or most any standard)

- Ensures Interoperability among vendor solutions
- Allows Choice of best-of-breed products
- Eliminates Risks of deploying proprietary technology

Why is HEP interested ?

- Don’t have to care about client software anymore.
- No specific ROOT drivers (dCap,rfio,xroot). Just ‘open /foo/blah’
- Less software components to maintain.
- Can be used by unmodified applications (e.g. Mathematica®)
- regular mount-point as any other FS e.g. /afs, /pnfs.
- File/Block caching algorithms provided by professional computer scientists within the OS kernel.

More more arguments see :

“11 reasons you should care” by Gerd Behrmann

At dCache.org/manuals

Who is supporting/funding it in HEP

European Middleware Initiative

Within the European Middleware Initiative, DPM, dCache and very likely StoRM will provide an NFS 4.1 (pNFS) interface.

Imposed by the EC : EMI will only fund standards.

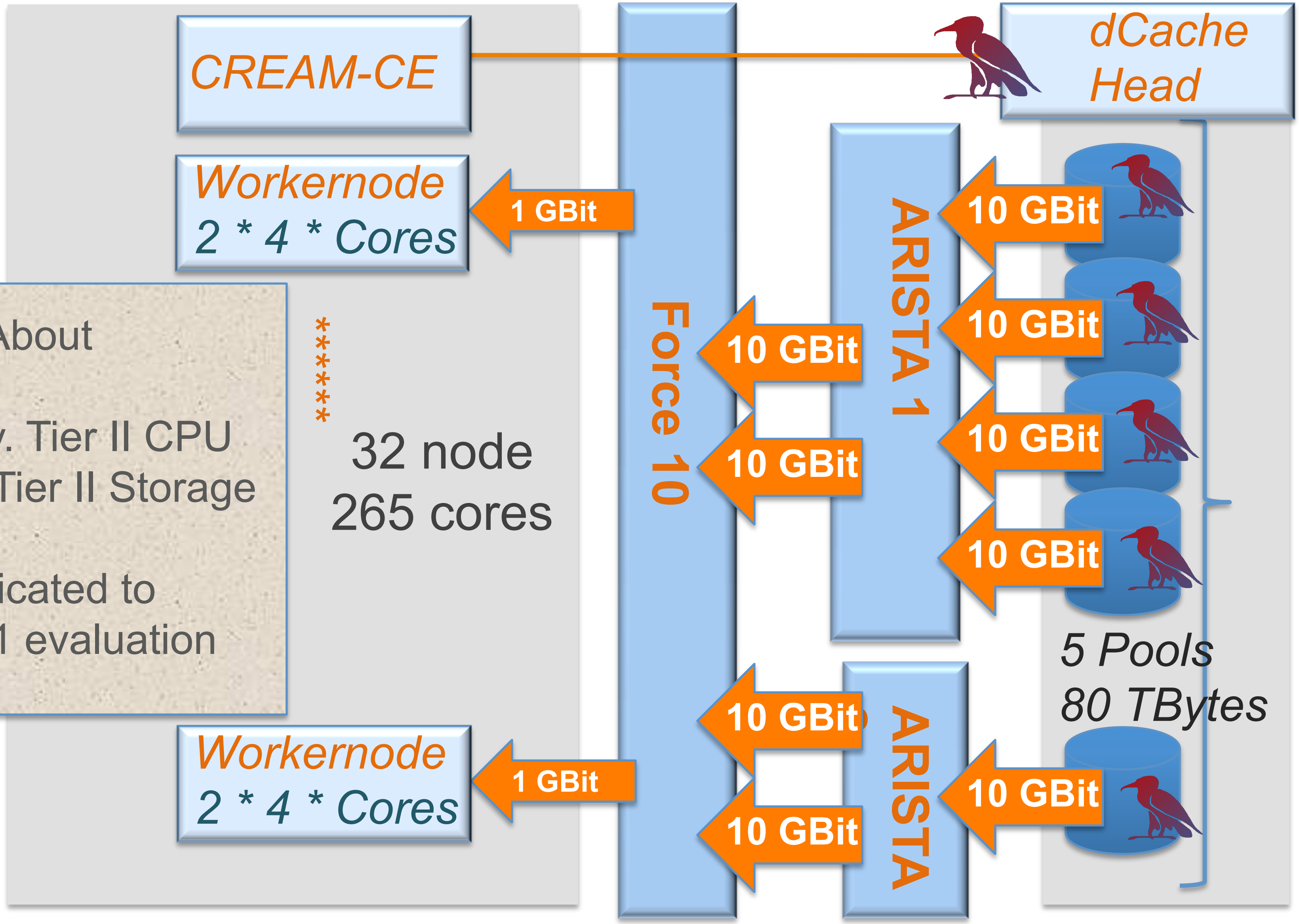
dCache production ready : 1.9.10
DPM : pNFS being finished later.

NFS 4.1 (pNFS) evaluation in dCache

dCache NFS 4.1 evaluation done by :

Yves Kemp
Tigran Mkrtchyan
Dmitri Ozerov

Our NFS 4.1 (pNFS) small Tier II ?



CREAM-CE

Workernode
2 * 4 * Cores

1 GBit

Force 10

10 GBit

10 GBit

ARISTA 1

10 GBit

10 GBit

10 GBit

10 GBit

dCache
Head

******

32 node
265 cores

5 Pools
80 TBytes

Workernode
2 * 4 * Cores

1 GBit

10 GBit

10 GBit

ARISTA

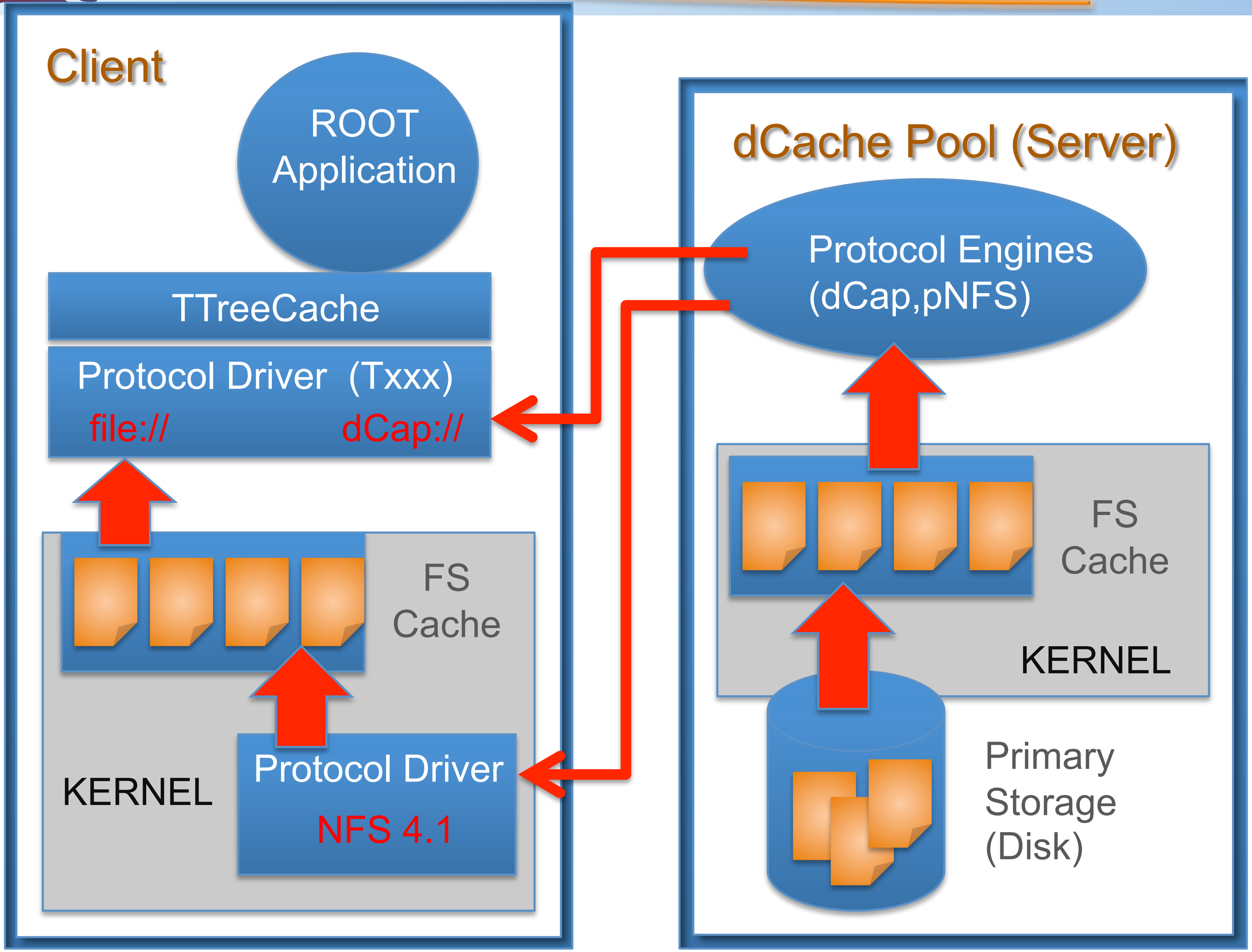
10 GBit

About

50 % av. Tier II CPU
20 % av. Tier II Storage

Dedicated to
NFS 4.1 evaluation

NFS 4.1 / dCap evaluation logic



Class of test

- Stability evaluation
- Simple I/O testing
- ROOT tests
- ATLAS HammerCloud

All tests done with :

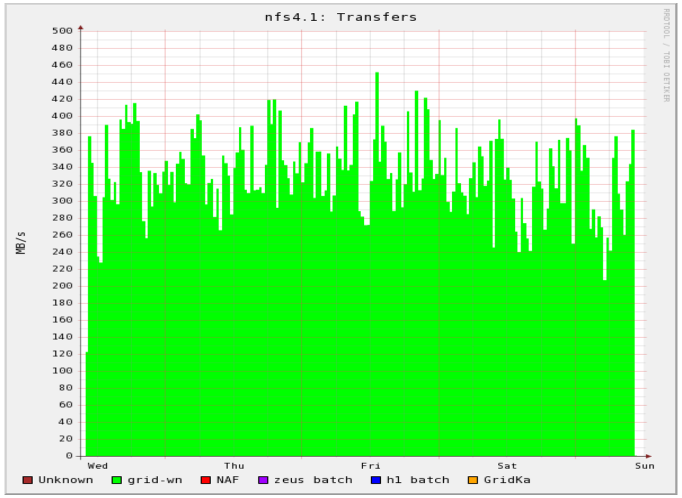
dCache 1.9.10

SL 5.3 2.6.36-rc3.pnfs

Stability

Stability

- CFEL Production Transfers from SLAC to DESY
  - 13 TBytes over 10 days
  - 100 GBytes average file size
  - No crash, no unexpected behaviour
- Un-taring Linux Kernel into NFS 4.1
  - No crash
- High-latency test
  - Recursive 'ls –l' over 60.000 files via DSL from home.
  - Finished w/o problem.
- 4 days at 330 MB/sec sustained Hammercloud. (stopped after 4 days)
- 128 Processes writing into the same file
  - Client nodes get stuck
  - Server was still ok



Simple I/O

Simple I/O Setup

Either

dccp <filename> /dev/null’

Or

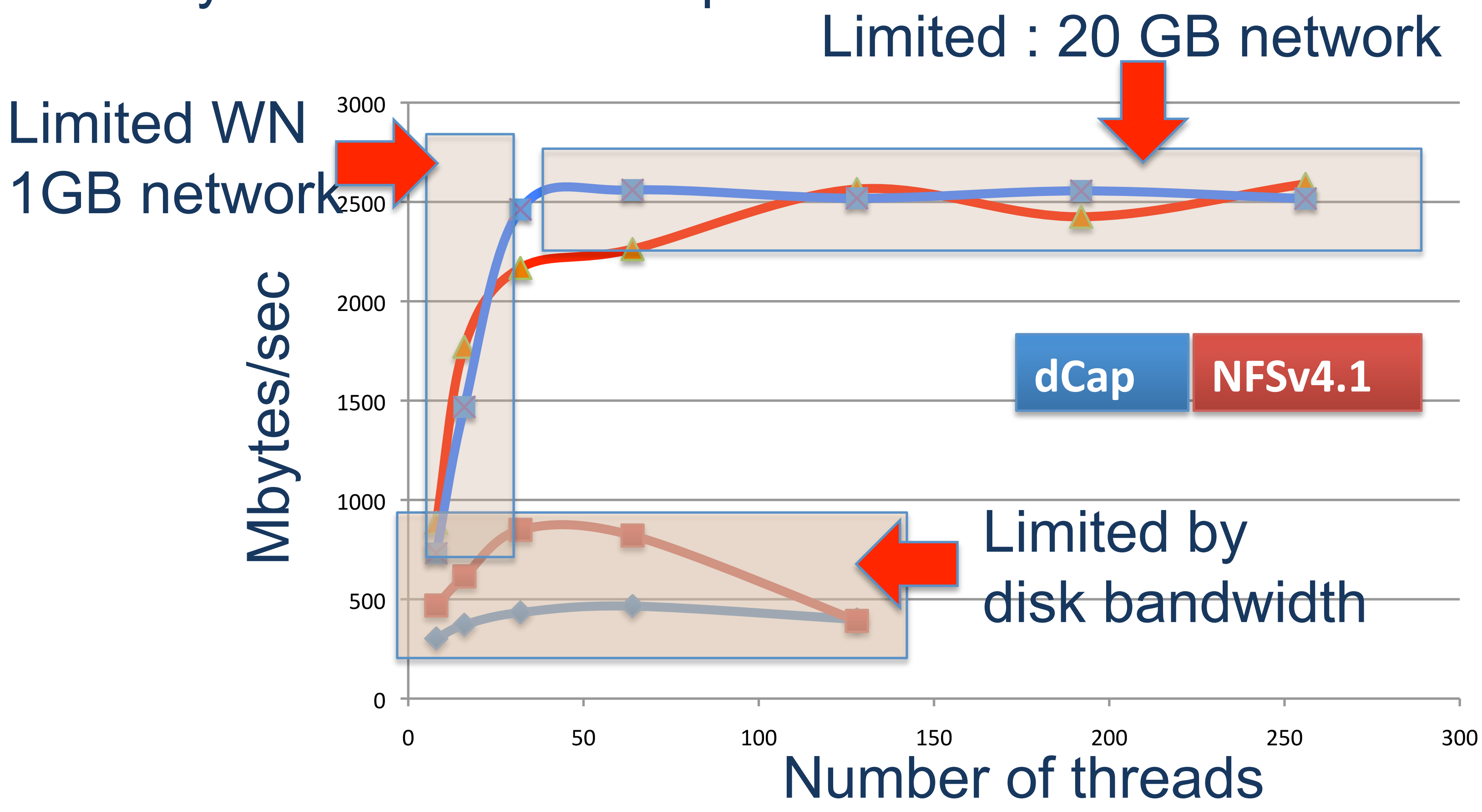
cat <filename> /dev/null

Only interested in protocol performance.
Preventing any client side caching effect.

✓Reading each file only once.
✓Reading files sequentially only.

Limits

Removing server disk congestion effect by keeping all data in file system cache of the pool.



Total throughput doesn't depend on the protocol.

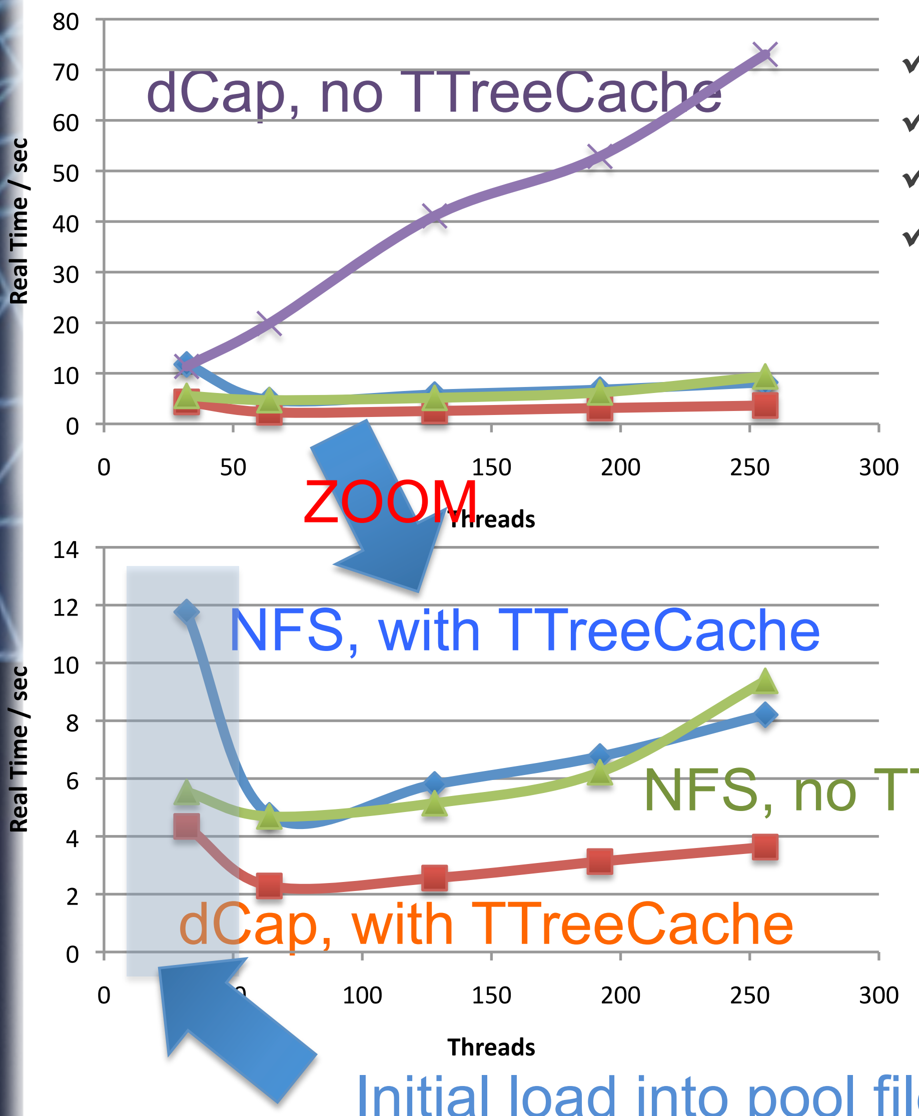
ROOT

ROOT Setup

- New ROOT version 5.27.06, compiled with dCap support
- Files provided by René Brun: atlasFlushed.root (re-organized files with optimized buffers) and AOD.067184.big.pool_4.root (some other original file) (optimized: 1GByte, original 1.3 GByte)
- Test script provided by René: simple script reading events: taodr.C
- Different test runs:
  - Reading via NFS or dCap
  - Reading with 60MByte TreeCache, or with 0Byte TreeCache
  - Reading all branches or only 2 branches
  - 32, 64, 128, 192 or 256 jobs running in parallel

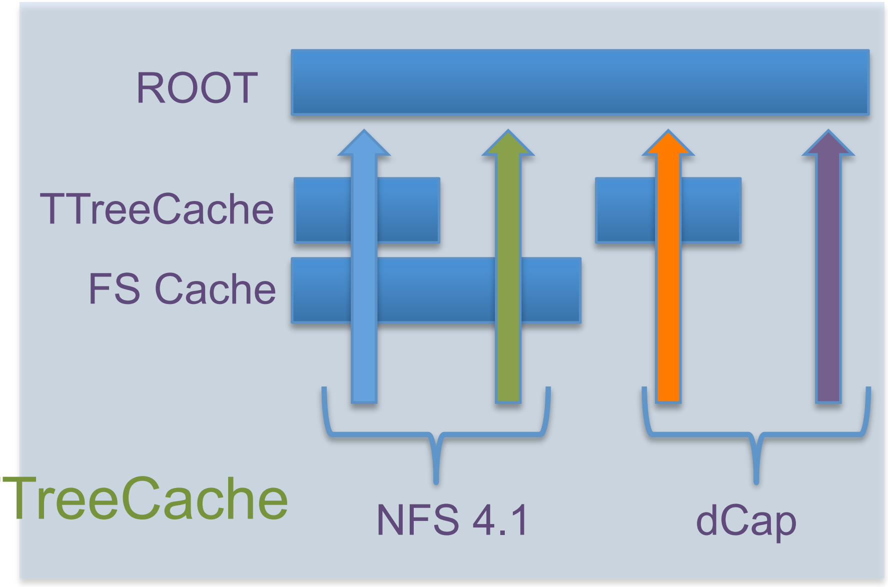
- Last minute-result! Have not spoken with ROOT people!

ROOT : Non optimized files, 2 trees only



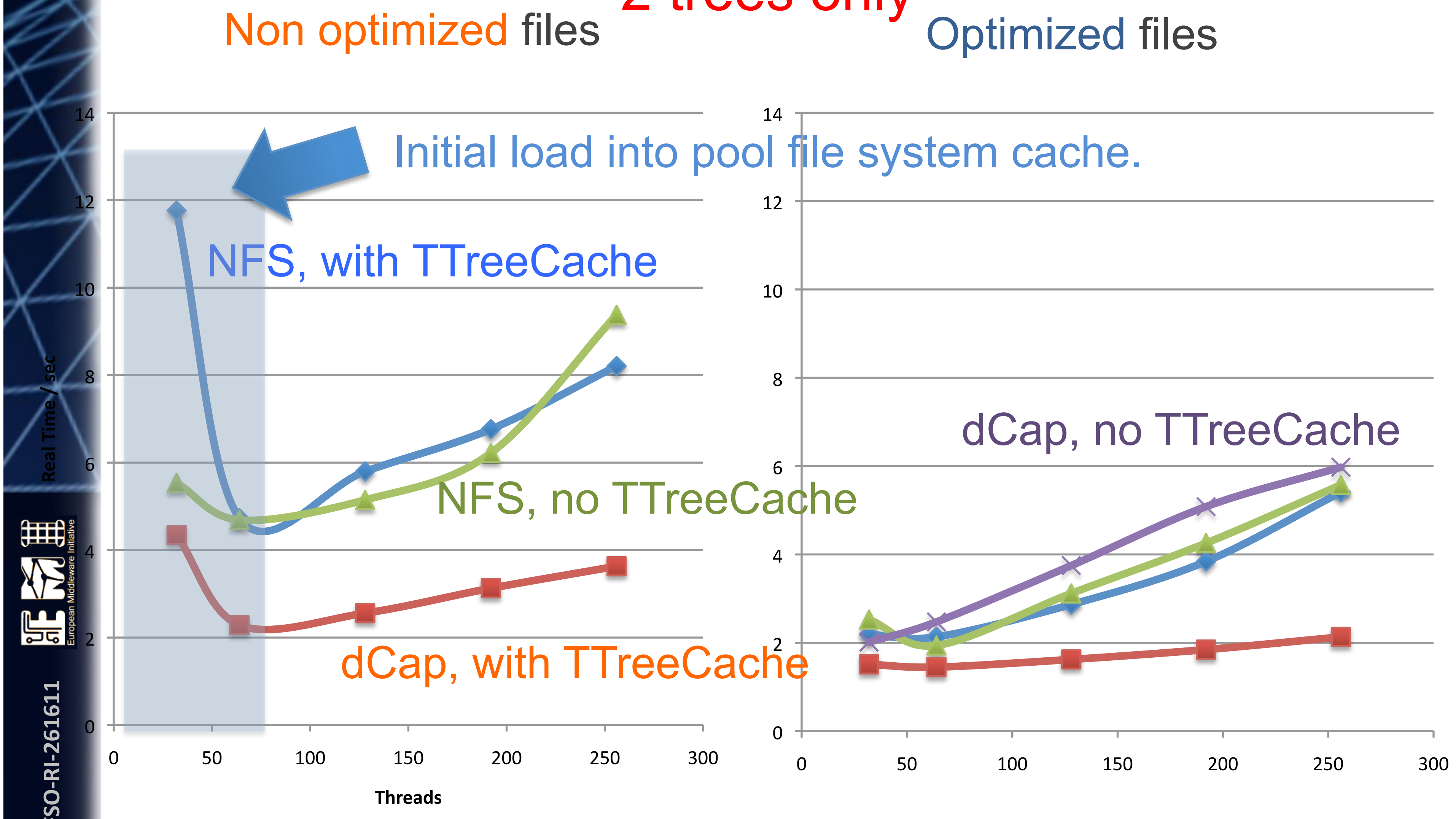
Initial load into pool file system cache.

- ✓Non optimized files
- ✓Reading only 2 trees.
- ✓TTreeCache does vector read with dCap.
- ✓VR = fadvise disabled in ROOT for NFS.



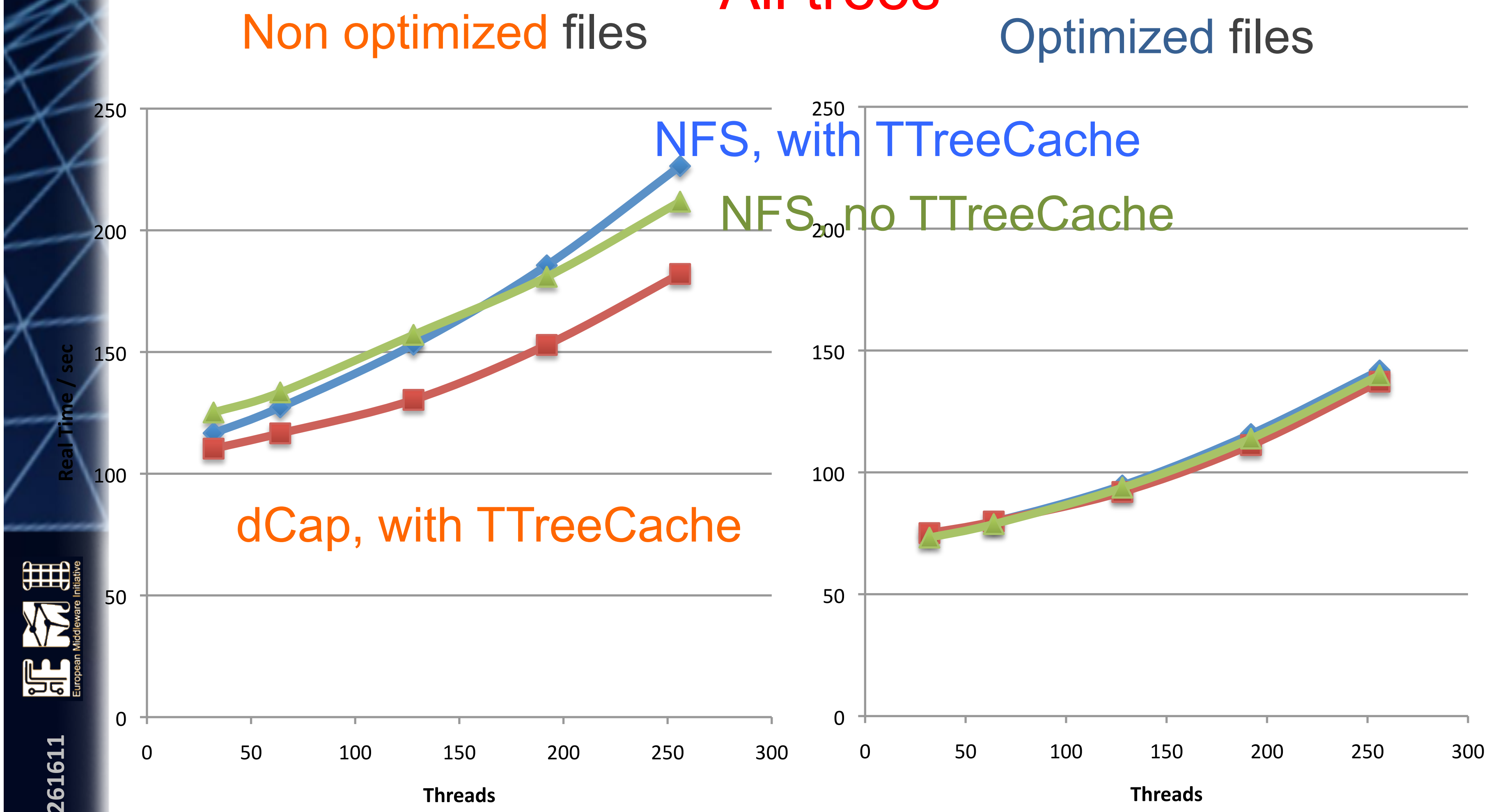
ROOT : optimized versus non optimized files

2 trees only



ROOT : optimized versus non optimized files

All trees



Two important concepts dominate analysis performance :

Client side caching

Vector Read

On client side caching

The above evaluation doesn't at all use client side caching

But

- From evaluation (last hepix) we know that caching is 50 % of the game.
- This can be achieved by
  - TTreeCache for ROOT application
  - dCap ++ (see Patrick's talk at Lisboa Hepix) any application using dCap.
  - Or client file system cache for NFS 4.1 (pNFS)
- For ROOT application, the TTreeCache has a slight advantage, as it knows the structure of the ROOT files and can act accordingly

The vector read magic

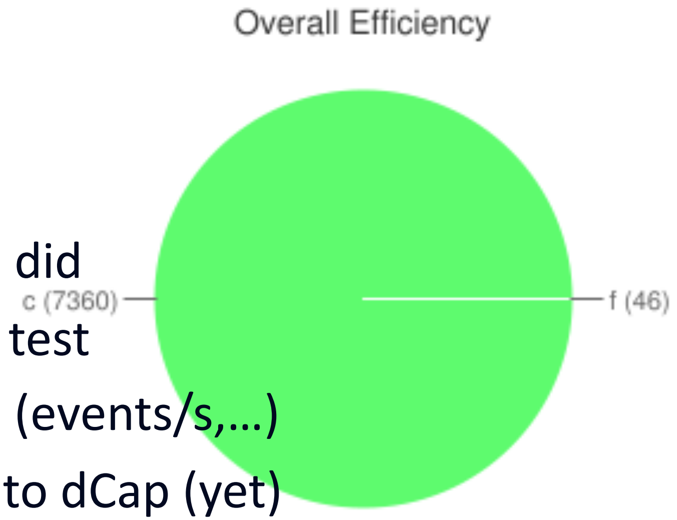
The above evaluation demonstrates the advantages of Vector-Read by ROOT.

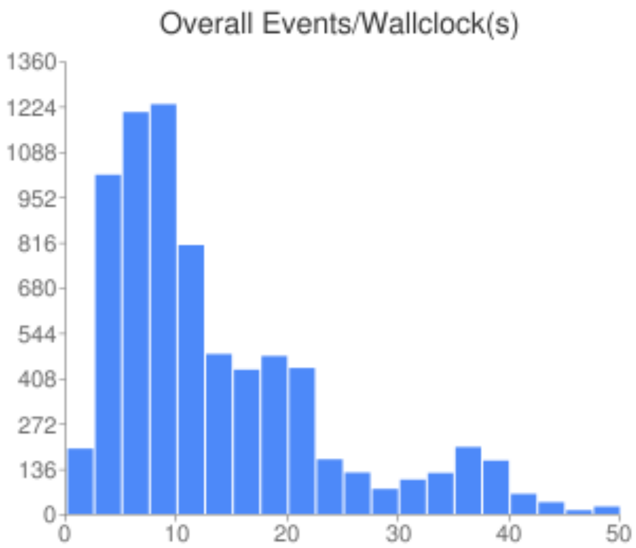
- Vector read can only be used through proprietary protocols (dCap,..)
- The file system semantics doesn't allow direct vector read. (bad)
- However, the is the famous 'fadvise' file system call :
  - Advised the file system (kernel) to prefetch certain portions of a file, if CPU time allows.
  - If those portions are read later, they are already available in the FS cache.
- Has been added to the 'file://' driver of ROOT and, according to Fons, improved access with 'file://' by up to 20%.
- Has been removed from the code again because it spoiled the TTreeCache I/O statistics. (very bad).

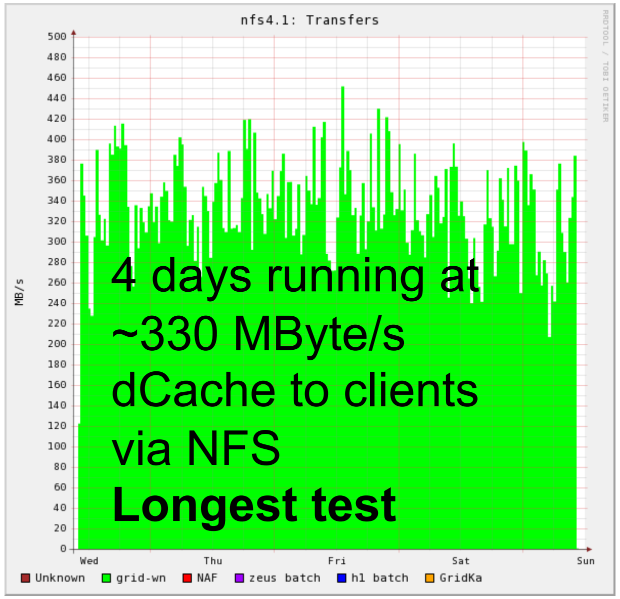
Hammer Cloud

ATLAS Hammer Cloud tests

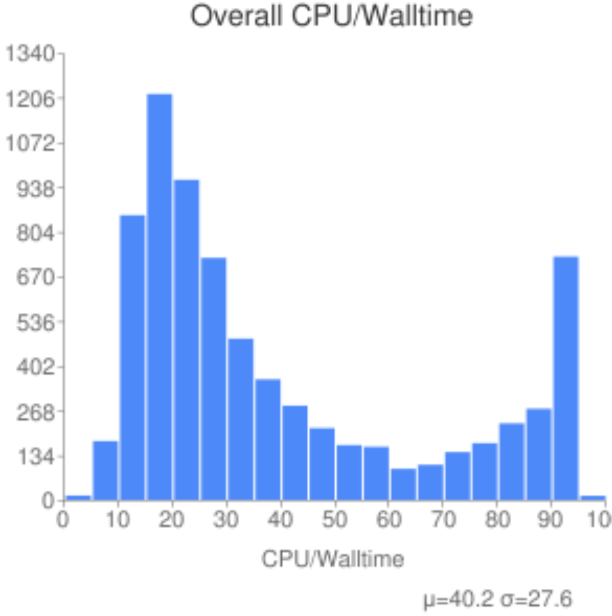
- 8248 jobs in total
- Cancelled after 4 days
- Longest single test we did
  - No trouble during test
- Reasonable outcomes (events/s,…)
- No comparison made to dCap (yet)







4 days running at ~330 MByte/s dCache to clients via NFS
Longest test



Client (kernel) availability

Kernel availability

- Kernel used for evaluation : 2.6.36_rc3
- NFS 4.1 (pNFS) kernels expected in SL6.(>2)
- 2.6.36 back-port to SL5 available from DESY
  - Plus ‘mount tools’ RPM.
  - Kernel will very likely not cover all hardware setups.
- With a Joined Effort (e.g. CERN, FNAL, DESY), we would be able to provide an SL5 with NFS 4.1 (pNFS) kernel within months. (If we really want)

Kernel availability

```
commit a4dd8dce14014665862ce7911b38cb2c69e366dd
Merge: b18cae4 411b5e0
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Tue Oct 26 09:52:09 2010 -0700

  Merge branch 'nfs-for-2.6.37' of
git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git

  * 'nfs-for-2.6.37' of git://git.linux-nfs.org/projects/trondmy/nfs-2.6:
    net/sunrpc: Use static const char arrays
    nfs4: fix channel attribute sanity-checks
    NFSv4.1: Use more sensible names for 'initialize_mountpoint'
    NFSv4.1: pnfs: filelayout: add driver's LAYOUTGET and
GETDEVICEINFO infrastructure
    NFSv4.1: pnfs: add LAYOUTGET and GETDEVICEINFO infrastructure
    NFS: client needs to maintain list of inodes with active layouts
    NFS: create and destroy inode's layout cache
    NFSv4.1: pnfs: filelayout: introduce minimal file layout driver
    NFSv4.1: pnfs: full mount/umount infrastructure
    NFS: set layout driver
    NFS: ask for layouttypes during v4 fsinfo call
    NFS: change stateid to be a union
    NFSv4.1: pnfsd, pnfs: protocol level pnfs constants
    SUNRPC: define xdr_decode_opaque_fixed
    NFSD: remove duplicate NFS4_STATEID_SIZE
```

First part of pNFS now in 2.6.37

Next Steps

- For more details check CHEP'10 presentation by Yves and Dmitri.
- More investigation with various different ROOT setups.
- Working with the CMS official test-case.
- Investigating X509 Certificate/Proxy security.
- Wide area transfer evaluation. (DPM, dCache, DESY, CERN)
- Setting up a regular NFS 4.1 (pNFS) system e.g. : NetApp and Pillar.
- Evaluation by the HEPIX working group.
- Trying to find groups as guinea-pigs for NFS4.1 production.

NFS 4.1 Conclusion

- Stability is much better than expected : Production ready.
- Kernel situation : short term solution for SL5 would be available, if we want.
- pNFS is partially already in 2.6.37
- Performance already comparable with existing solutions.
- Nevertheless : more evaluation on ROOT framework interaction needed. (vector read, fadvise)
- Efforts will continue within the EMI/dCache.org framework.
- You want to volunteer ?
  - Get dCache 1.9.10 from dCache.org
  - Get nfs enabled kernel : http://www.dcache.org/chimera/x86_64/

Conclusions

- *EMI Data* is a good opportunity to get our storage management middleware into a maintainable shape.
- It provides the money and the infrastructure.
- Standardization is the way to get broader acceptance by other communities.
- Everybody can join or may provide suggestions through WLCG or EGI.eu.

EUROPEAN MIDDLEWARE INITIATIVE

Further reading

https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T3Data

EMI is partially funded by the European Commission under Grant Agreement INFSO-RI-261611