Data Management and Workload Management in CMS

DESY Computing Seminar
11.06.2012

Manuel Giffels
CERN-Fellow PH-CMG-CO

Outline

- Introduction
  - The CMS Experiment
  - The Computing Model
- First data processing at TIER-0
- Handling meta-data
- How to transfer data
- Tools for distributed processing at TIER 1/2
- User analysis in CMS

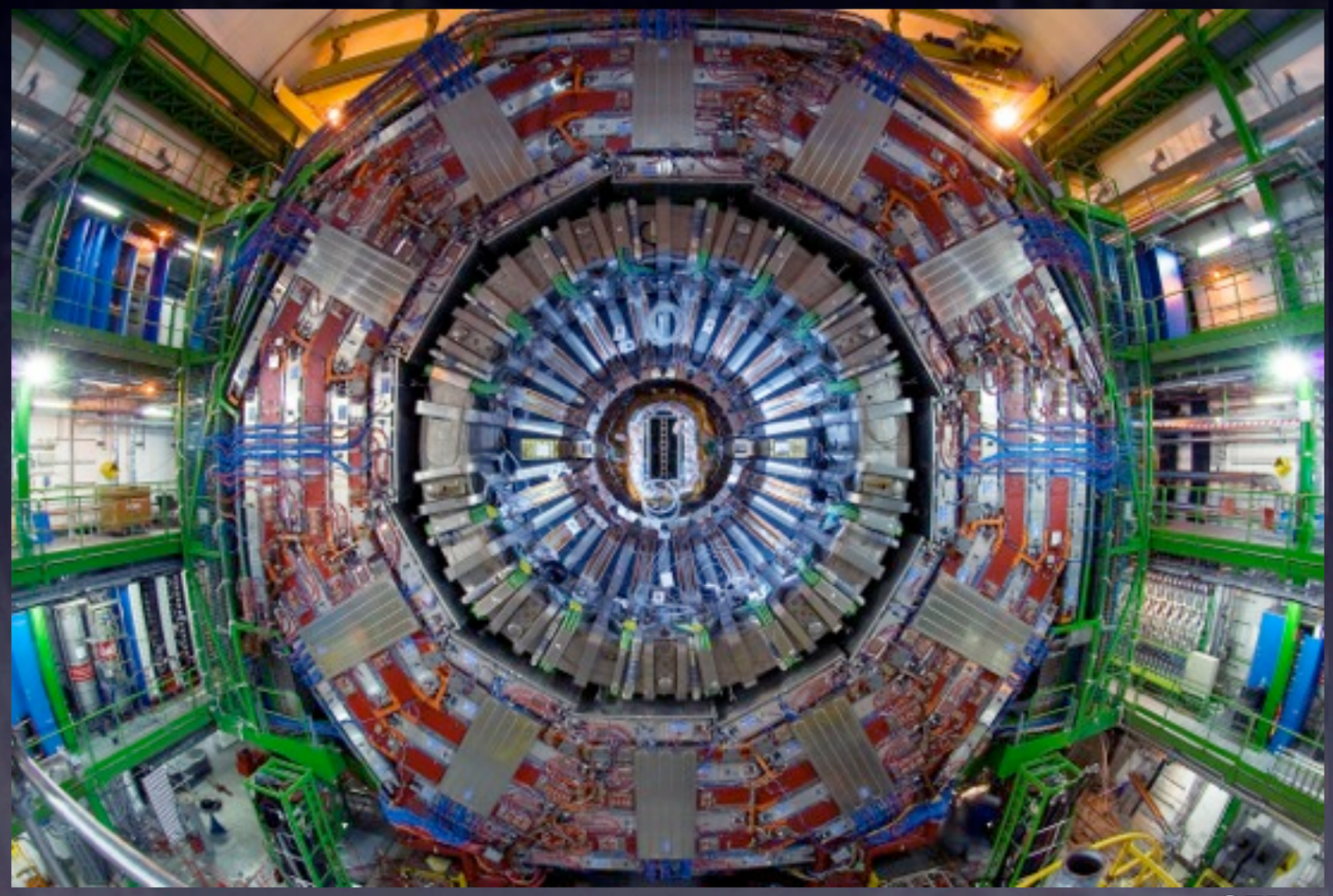
The CMS Experiment

The **C**ompact **M**uon **S**olenoid is a general purpose detector at the Large Hadron Collider at CERN (LHC Point 5)

Trigger	350+600 Hz
RAW Size	0.5 MB
RECO Size	1 MB
RECO Time	~10 s

<u>Huge Data Source:</u>

- Read out tens of TB/s
- After triggers still ~ 0.5 GB/s

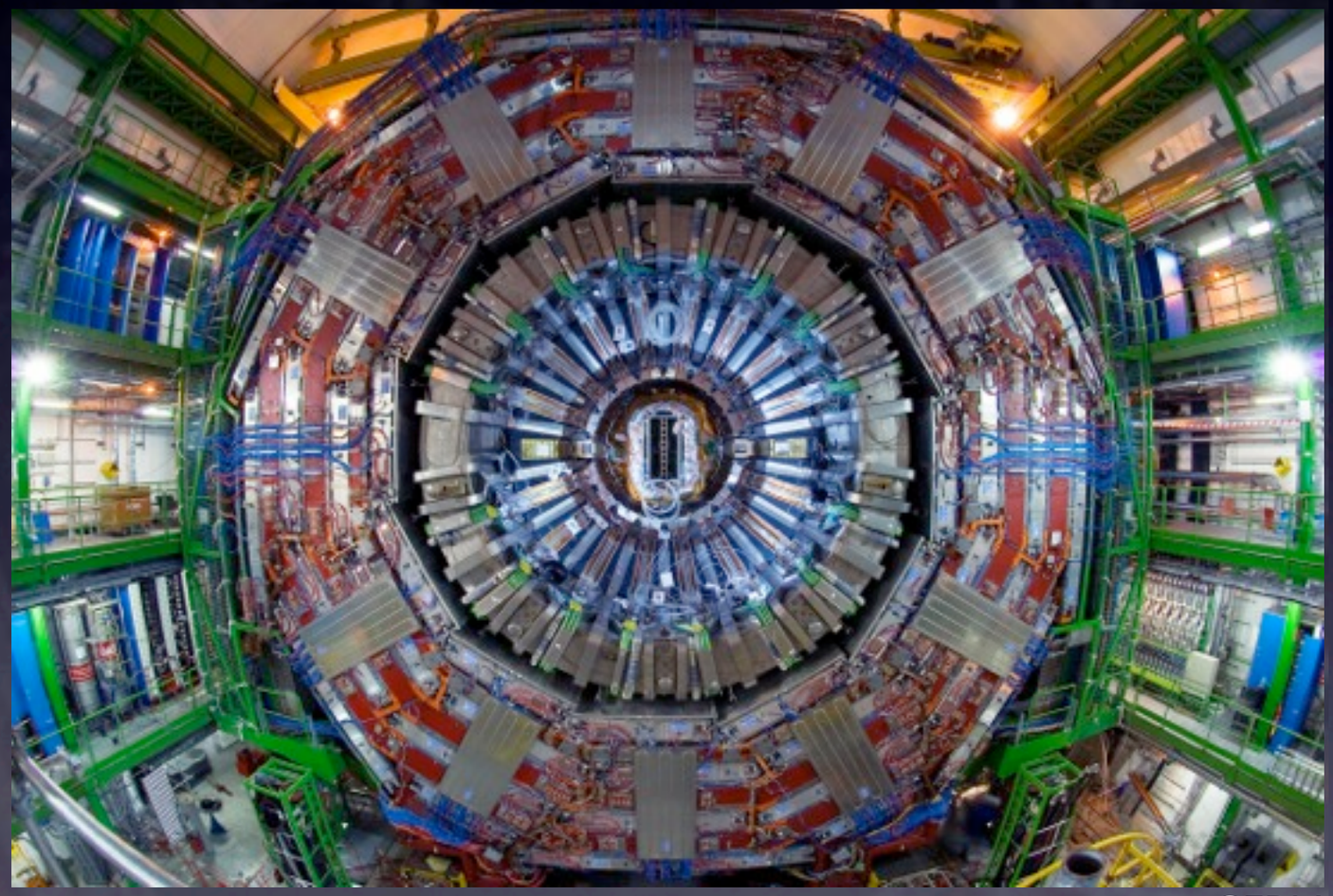


CERN

The CMS Experiment

The **C**ompact **M**uon **S**olenoid is a general purpose detector at the Large Hadron Collider at CERN (LHC Point 5)

Trigger	350+600 Hz
RAW Size	0.5 MB
RECO Size	1 MB
RECO Time	~10 s



CERN

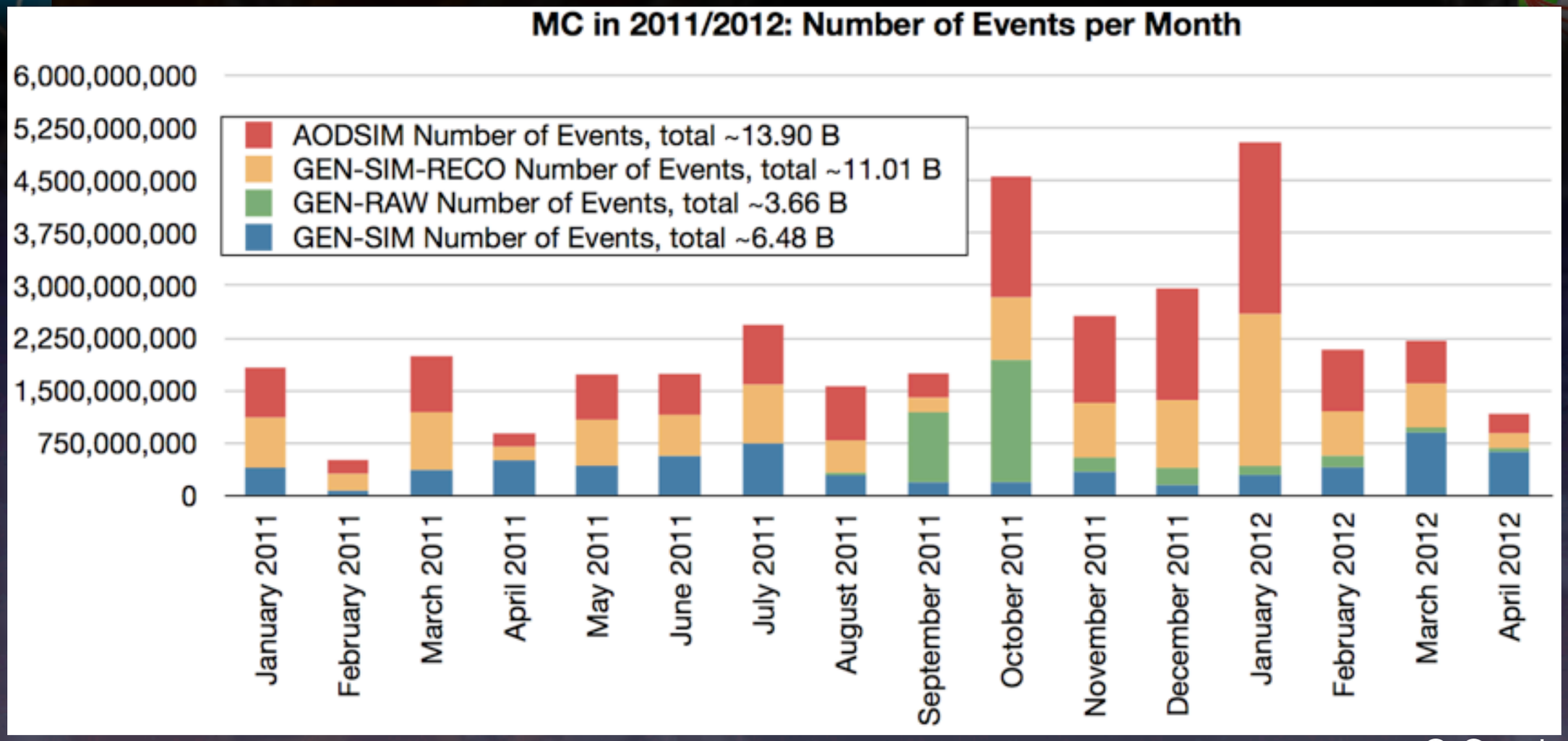
Huge Data Source:

- Read out tens of TB/s
- After triggers still ~ 0.5 GB/s

▸A lot of data to cope with

▸A lot of CPUs and storage needed

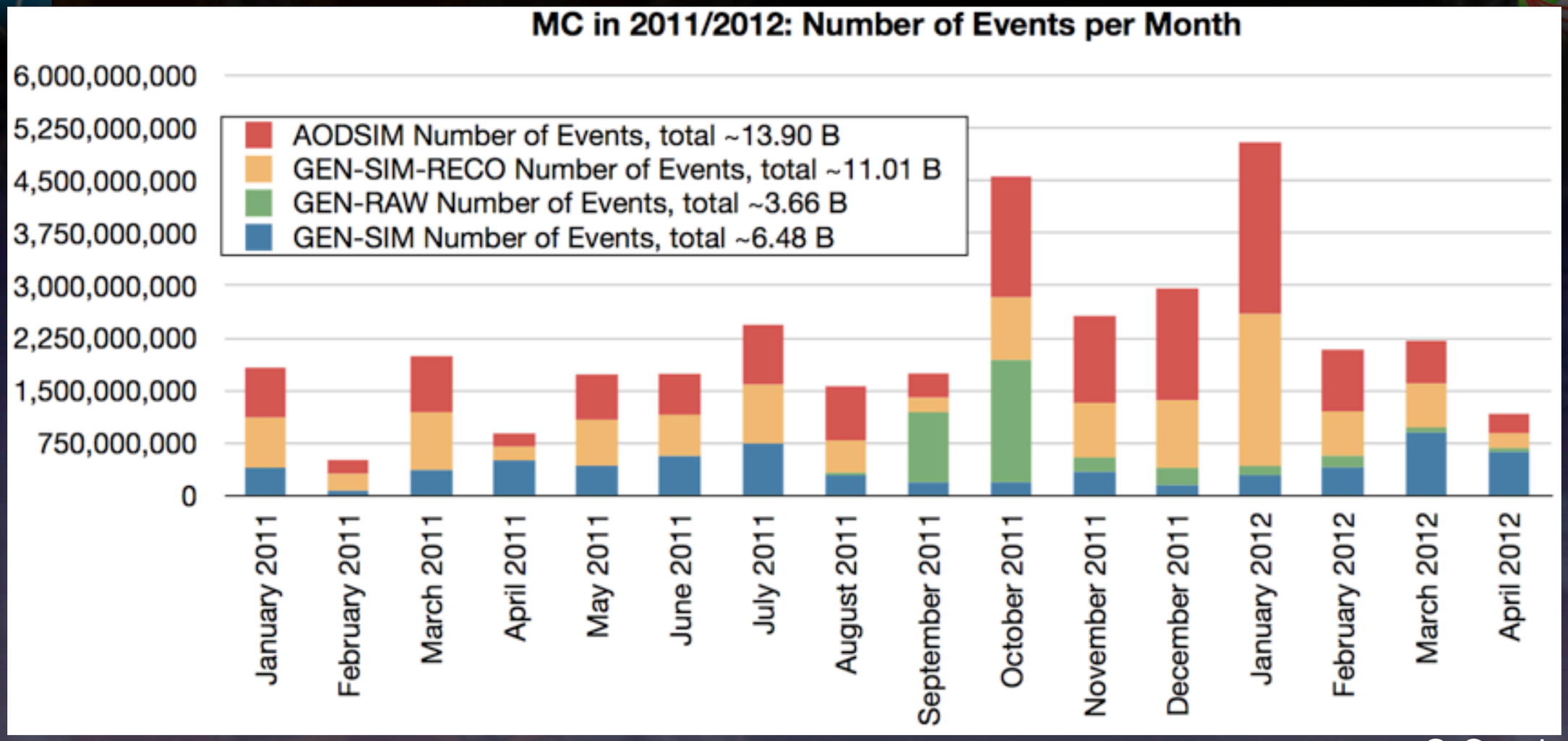
Monte Carlo



O. Gutsche

- Simulated 4.25 billion events in 2011, 2.22 billion in 2012
- Reconstructed 2.5 billion twice with different pile-up scenarios
- plus re-reconstruction due to changed software version

Monte Carlo

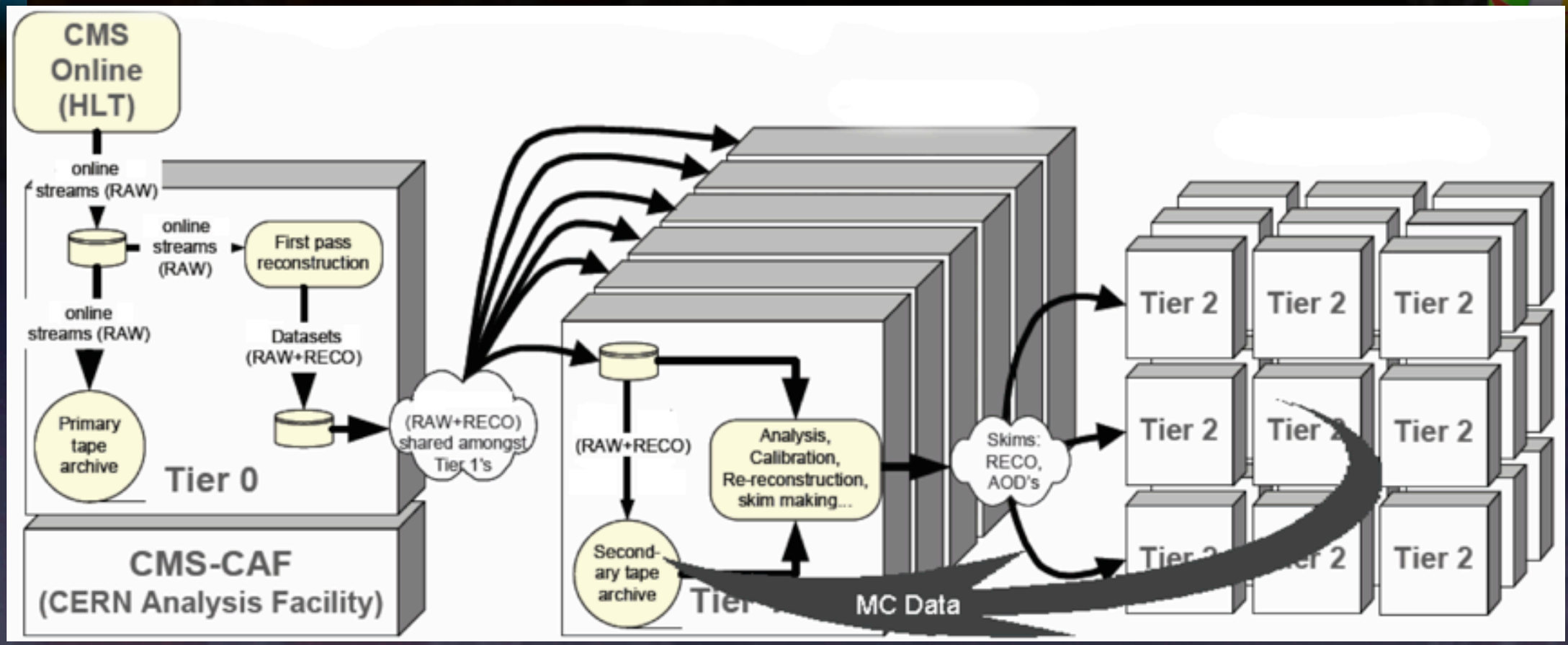


O. Gutsche

- Simulated 4.25 billion events in 2011, 2.22 billion in 2012
- Reconstructed 2.5 billion twice with different pile-up scenarios
- plus re-reconstruction due to changed software version

▸More data to cope with ▸More CPUs and storage needed

CMS Computing Model



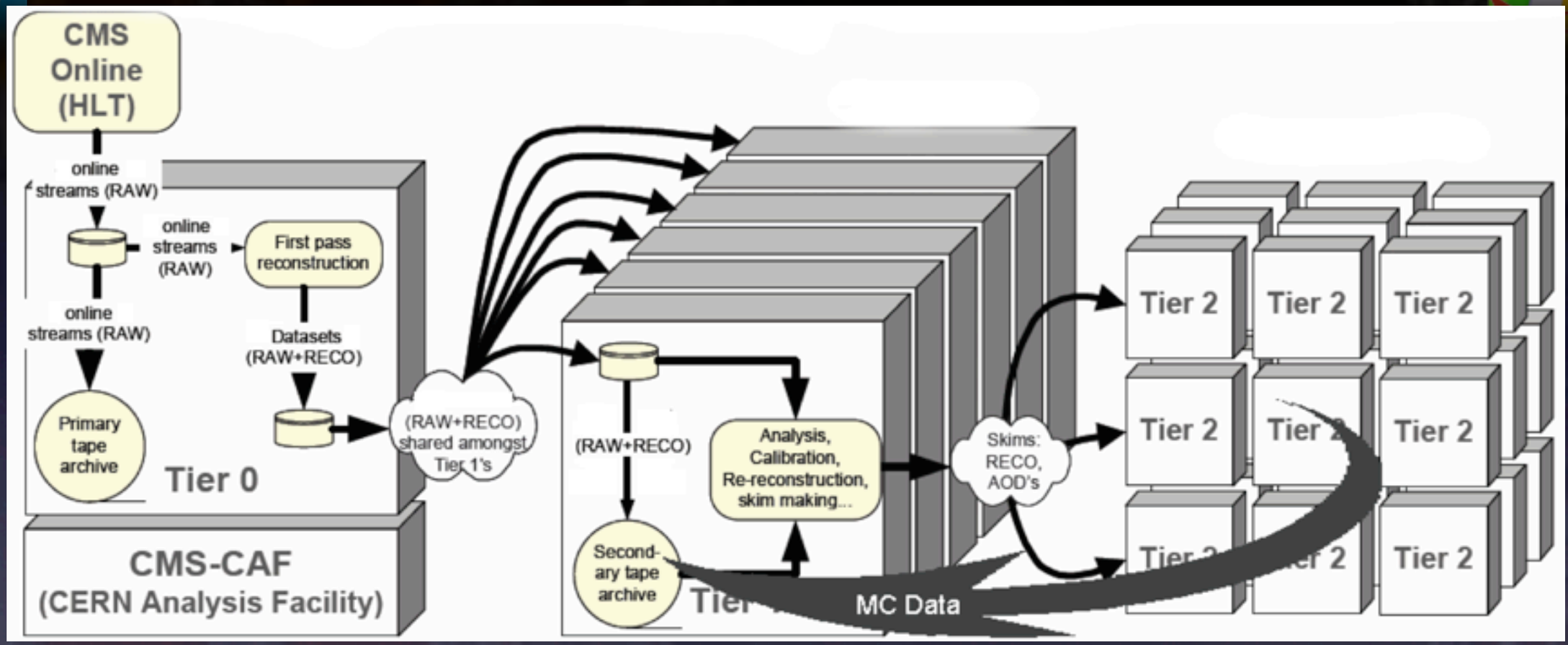
Computing Model TDR 2005 w/o recent evolution

- Using the **W**orld **L**HC **C**omputing **G**rid =>Distributed computing
- 4-Layers (TIERs) each serving resources for different tasks
- Most of the resources world-wide distributed

	CERN	T1s	T2s
kHEP SPEC06	106	131	312
Disk PB	4.5	16.1	20.4
Tape PB	21.6	44.1	-

Available Resources 2011, K. Bloom

CMS Computing Model



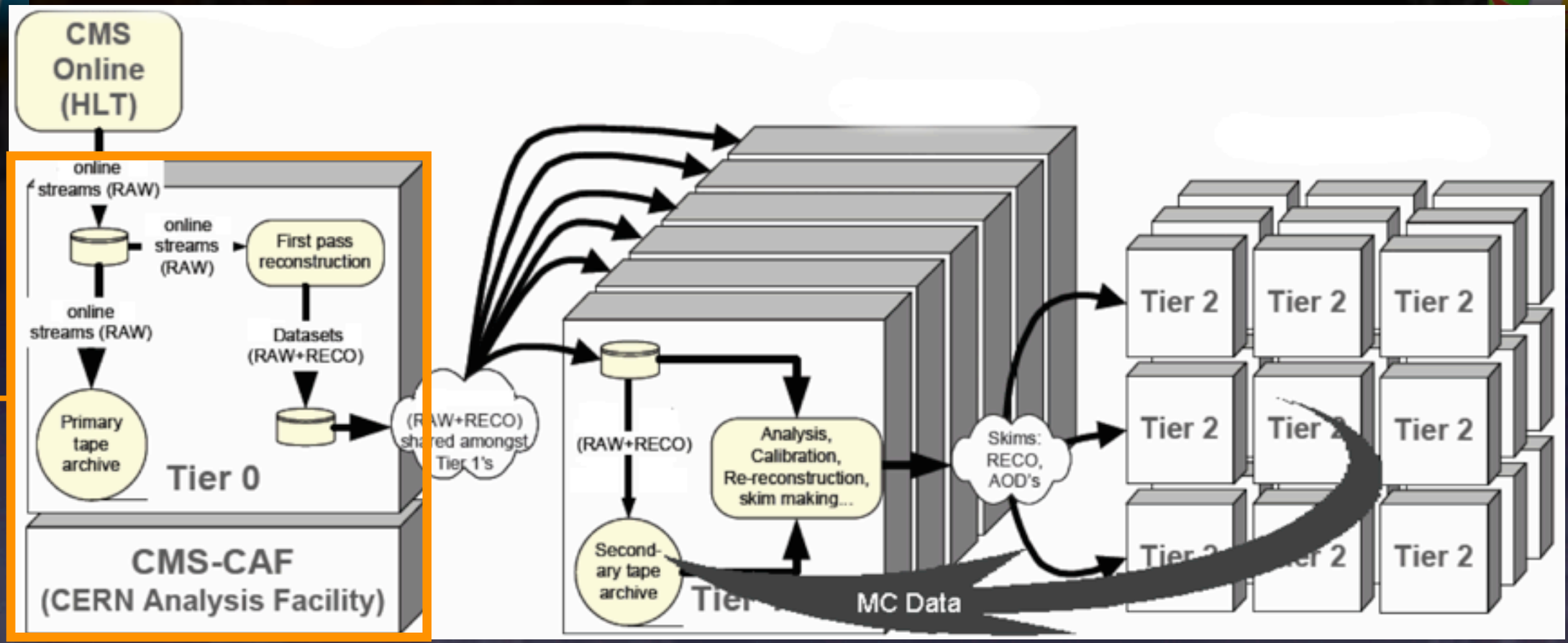
Computing Model TDR 2005 w/o recent evolution

- Using the **W**orld **L**HC **C**omputing **G**rid =>Distributed computing
- 4-Layers (TIERs) each serving resources for different tasks
- Most of t

	CERN	T1s	T2s
kHEP SPEC06	106	131	312
Disk PB	4.5	16.1	20.4
		44.1	-

Available Resources 2011, K. Bloom

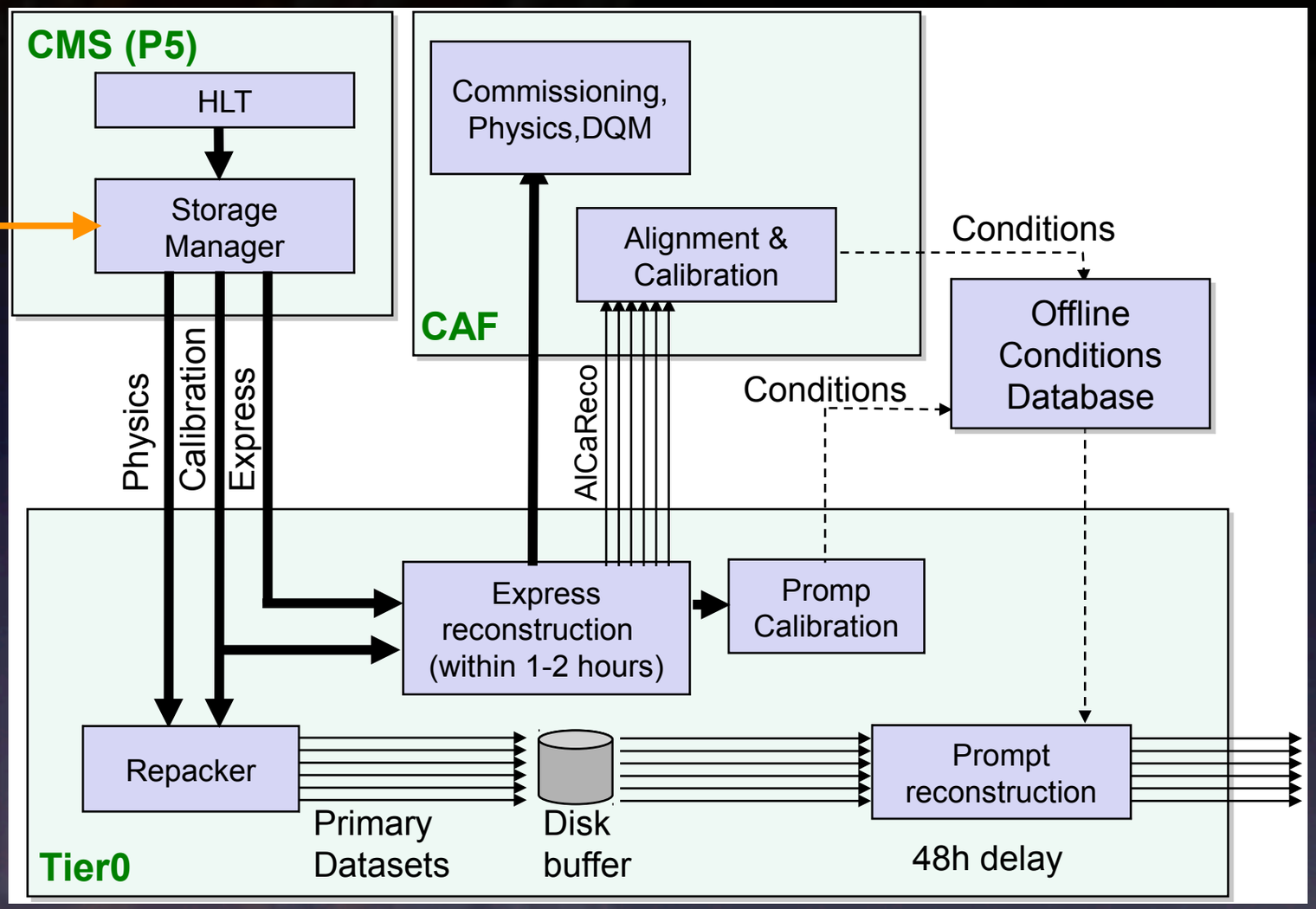
▸Software for distributed processing necessary



Computing Model TDR 2005 w/o recent evolution

TIER 0 Processing

TIER 0 - Processing

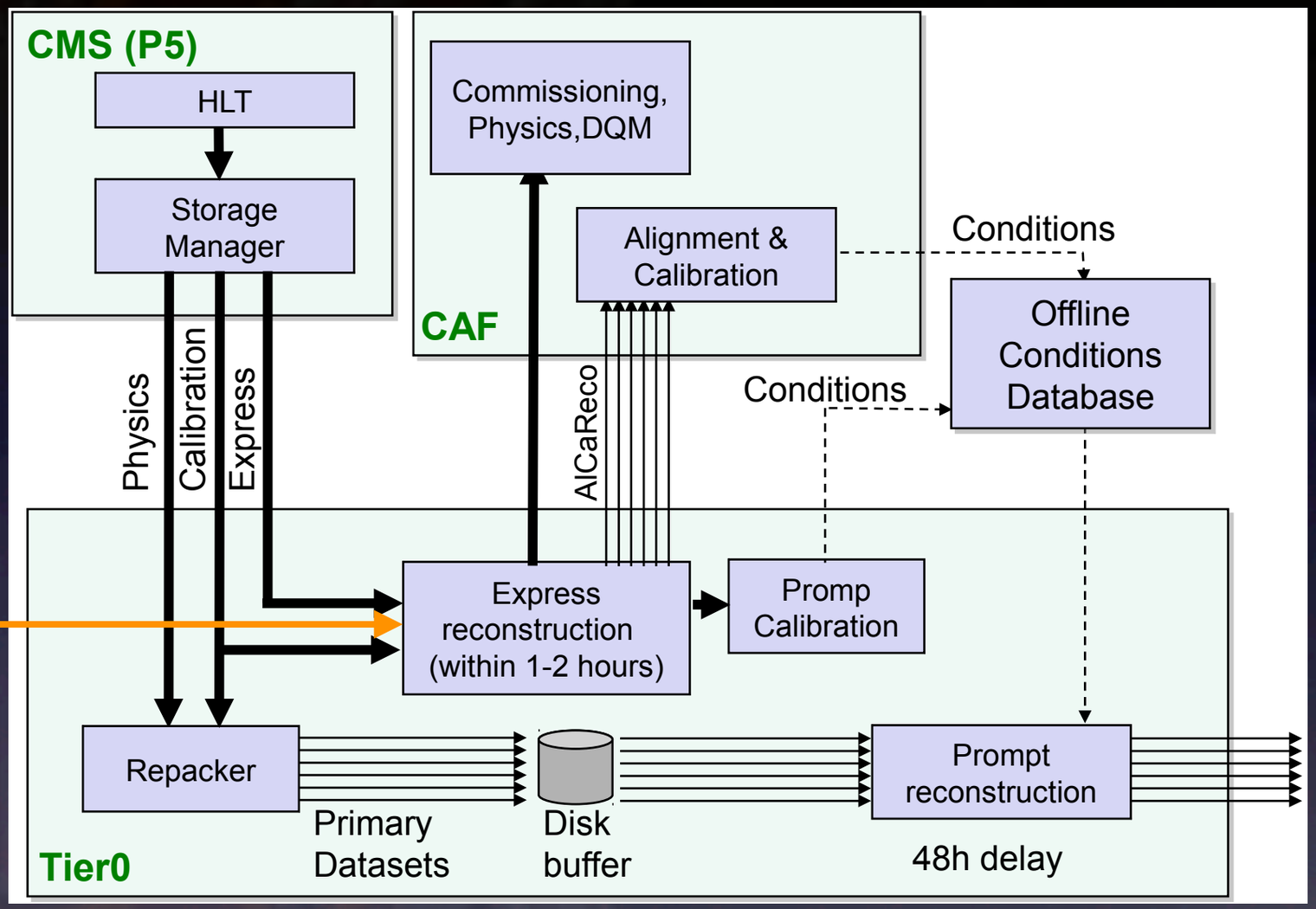


D. Hufnagel

Storage Manager:

- Writes data to a disk buffer at P5 in three online streams for p-p-collisions
  - Physics stream, rate of 350 Hz / 600 Hz (Prompt Reconstruction / data parking)
  - Express stream for fast monitoring and Prompt Calibration (~30Hz)
  - Calibration/monitoring streams
- Multiple instances causes splitting of lumi section across different streamer files

TIER 0 - Processing

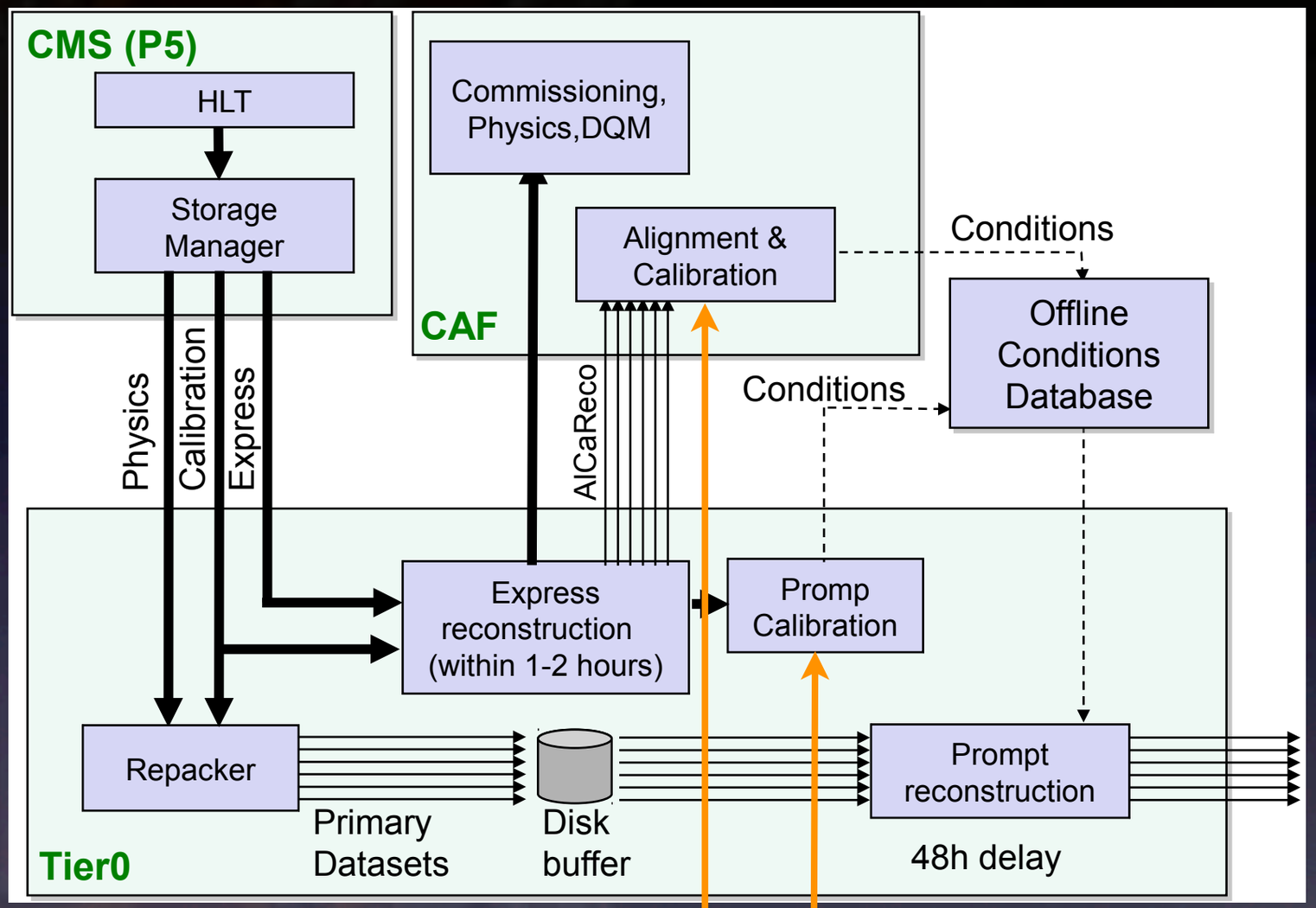


D. Hufnagel

Express Reconstruction:

- Full reconstruction of a subset of data within 1 hour
- Input for fast monitoring/analysis and feedback
- Also Prompt Calibration runs on output of Express Reconstruction

TIER 0 - Processing

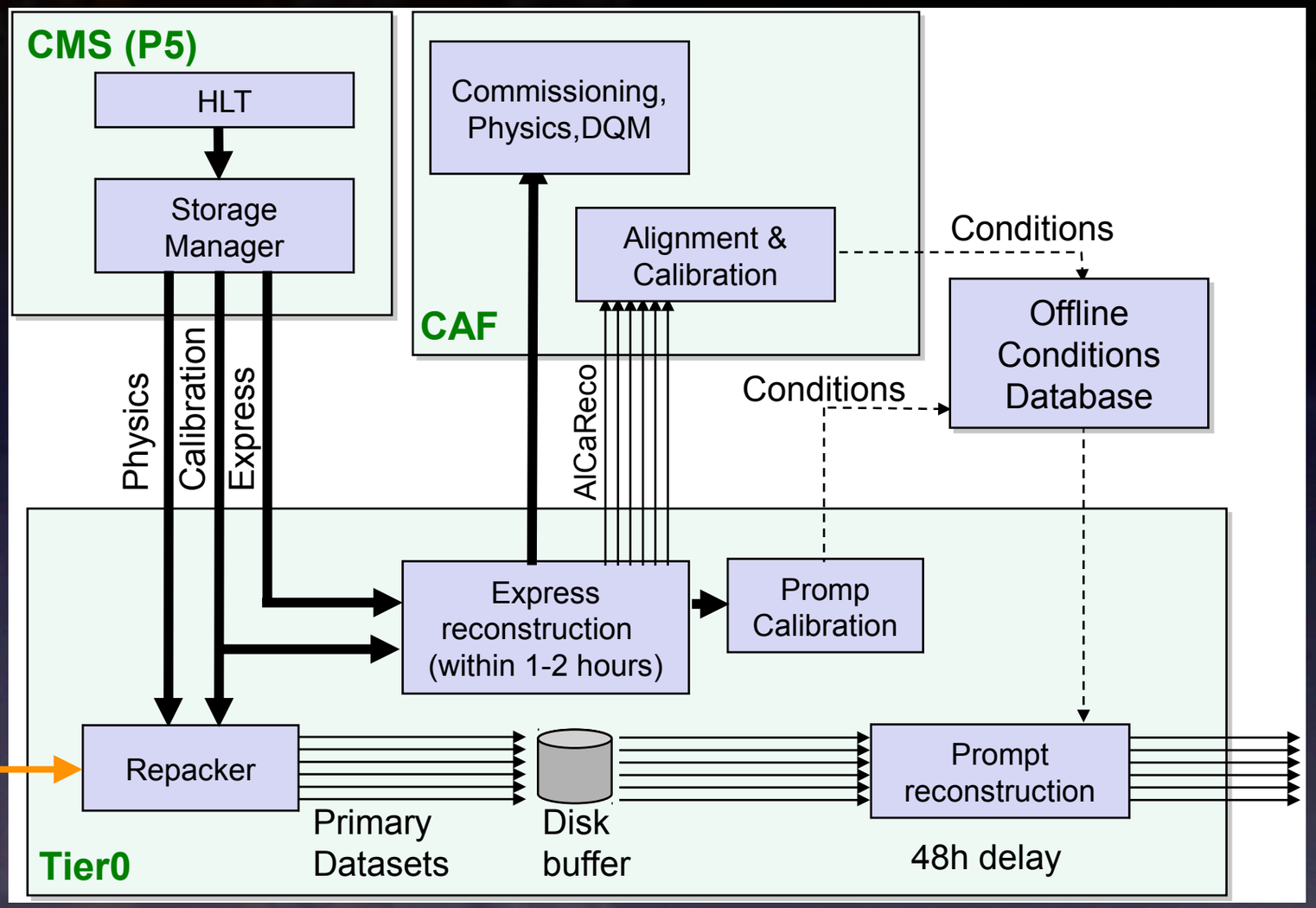


D. Hufnagel

Calibration:

- CAF: Alignment and Calibration run manually or semi-automated
- T0: Workflows are fully automated

TIER 0 - Processing

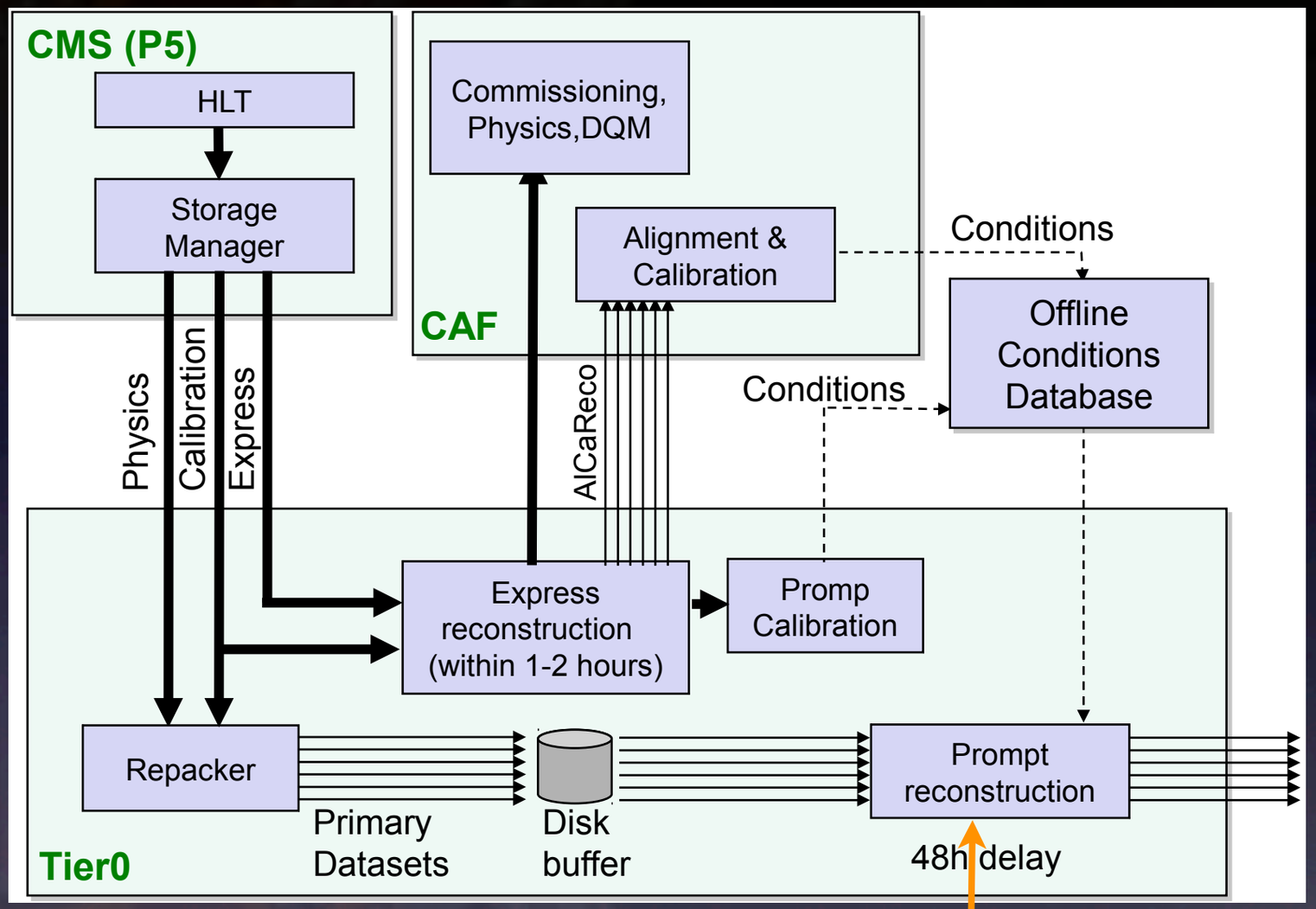


D. Hufnagel

Repacker:

- Conversion of streamer files to backward compatible RAW files for custodial storage
- Re-arrange lumi sections to avoid spreading over multiple files
- Split streams into different primary datasets according to trigger classification

TIER 0 - Processing

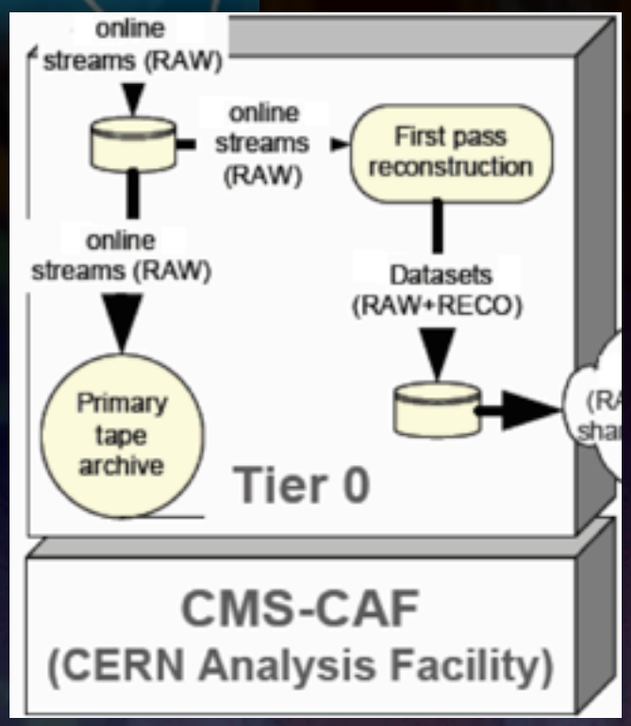


D. Hufnagel

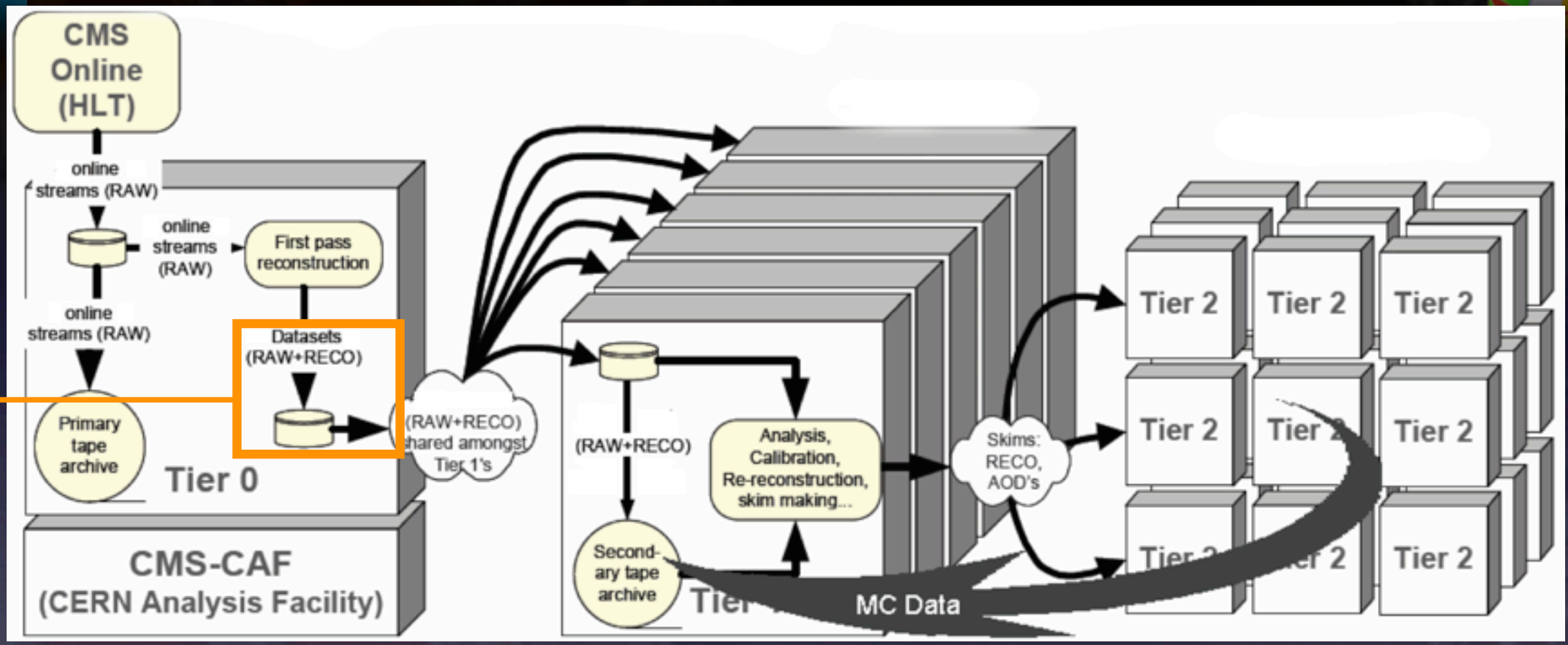
<u>Prompt Reconstruction:</u>

- Data is reconstructed delayed by 48 hours latency (configurable) to wait for updated conditions from Prompt Calibration (Finished within 24 hours after start)

TIER 0 - Processing



- Based on the ProdAgent architecture
  - ProdAgent is a workload management system originally designed for simulation tasks
  - Not providing all features needed for TIER 0 processing (dependent/nested workflows, automation, reliable job tracking)
  - ProdAgent job execution layer used to run jobs
  - Additional components developed to create TIER 0 specific types of jobs
  - Custom file and job bookkeeping developed (T0AST)
  - These experiences also contributed to the development of a new WM-System
- TIER 0 Activity State Tracker (T0AST)
  - Bookkeeping of the current system state, all files, all jobs and their associated metadata
  - State consistency guaranteed by database transactions
    - Action of the component and state change done in one transaction
- TIER 0@WMAgent currently in development
  - Feature complete concerning data processing
  - DQM harvesting and PromptCalibration are missing



Computing Model TDR 2005 w/o recent evolution

Meta Data?

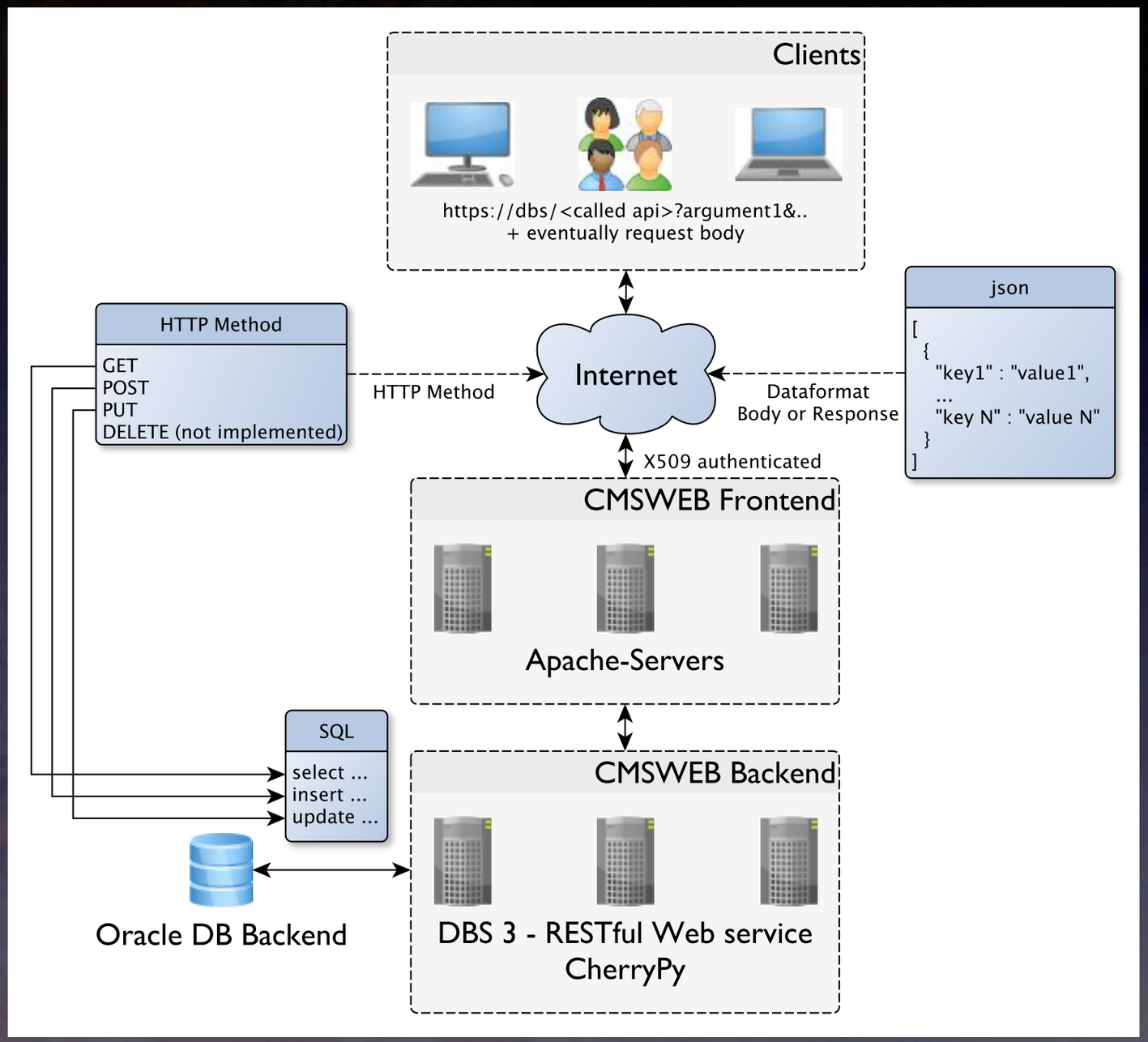
DBS

- Data Bookkeeping Service provides an event data catalog for CMS
- Contains information used for tracking datasets, their data-processing history, associations between runs, files and datasets
- On a large scale of about 10^5 datasets and more than 10^7 files
- All the data-processing relies on the information in DBS
- DBS 3 is currently in development driven by
  - lessons learned from its predecessor DBS 2
    (Lightweight and well-defined API, better scalability, optimized DB interactions)
  - revision of the data and workload management (DMWM) software
- Main foci of the development were
  - Adaptation of the database schema to better match the evolving CMS data-processing model
  - Achievement of a better scalability
  - Better integration into the DMWM software (DBS becomes a data-service, no UI anymore)

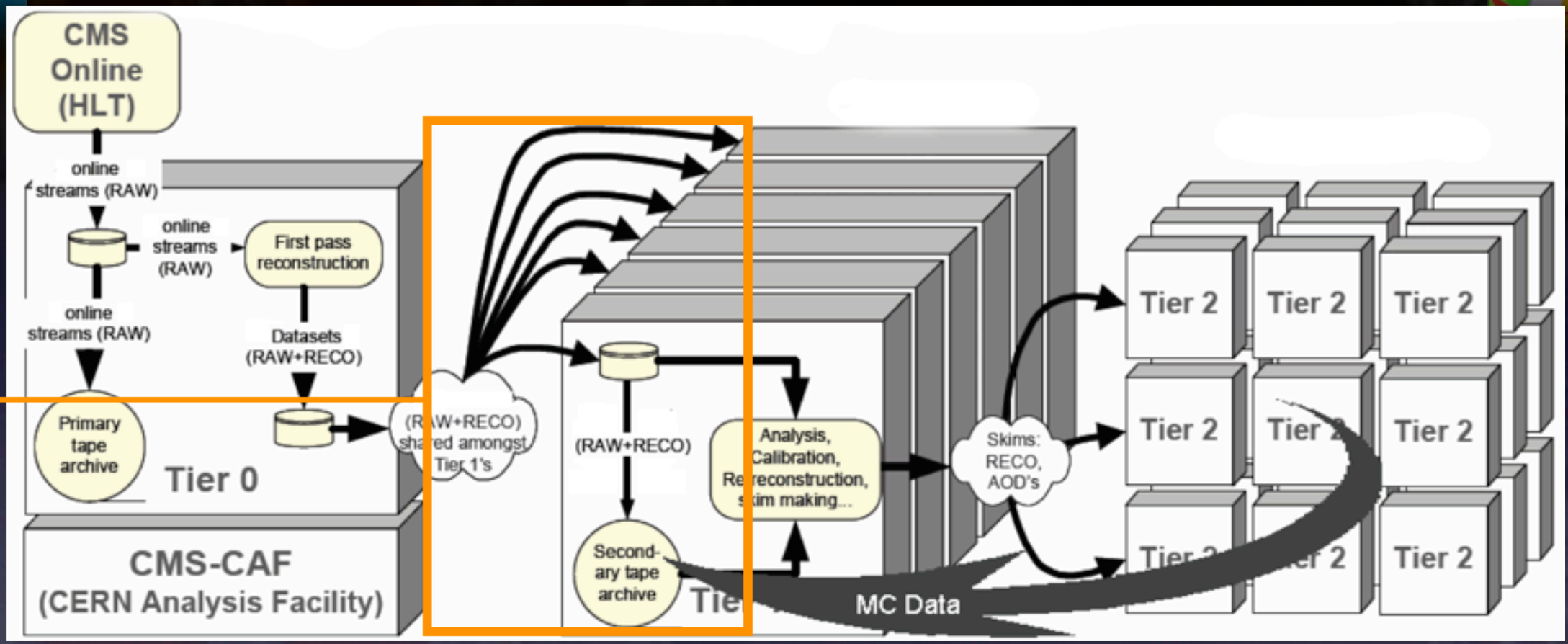
DBS

- DBS 3 was re-designed and re-implemented in Python using CherryPy
- Improved scalability is achieved by its RESTful (Representational State Transfer) design
  - using lightweight APIs (Amdahl's law scaling limits)
  - a stateless client-server communication
- Deletion of data inside the catalog is not provided to ensure perpetual traceability
- Java Script Object Notation (JSON) data-format is used for interchanging information with its clients
- Oracle DB is utilized as persistent storage (improved schema)
- All these tools are commonly used in CMS Computing, therefore DBS 3 is well integrated in the new DMWM architecture and one can profit from synergistic effects.

DBS



DBS 3 is already deployed and currently in testing phase



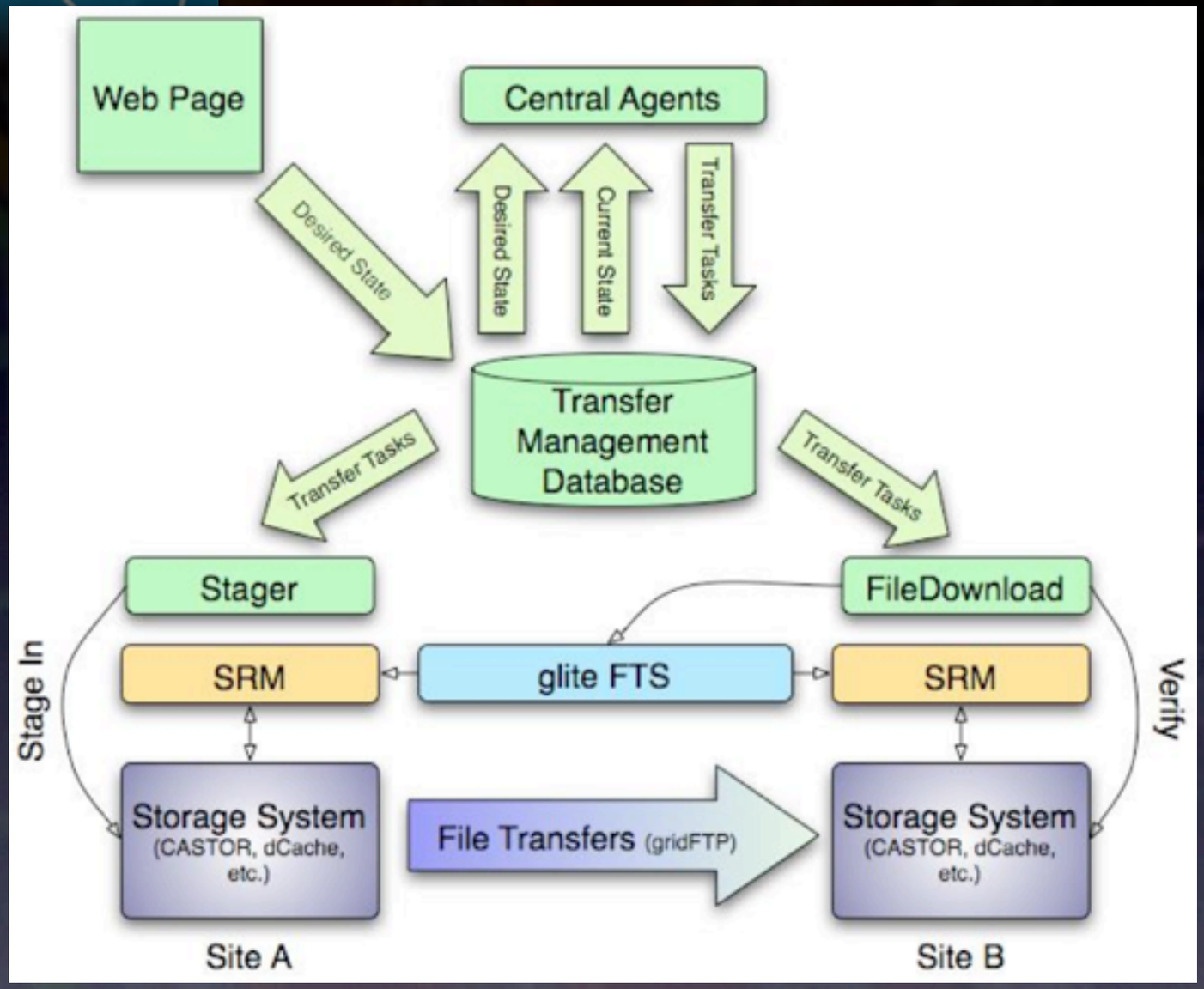
Computing Model TDR 2005 w/o recent evolution

Data Transfers?

PhEDEx

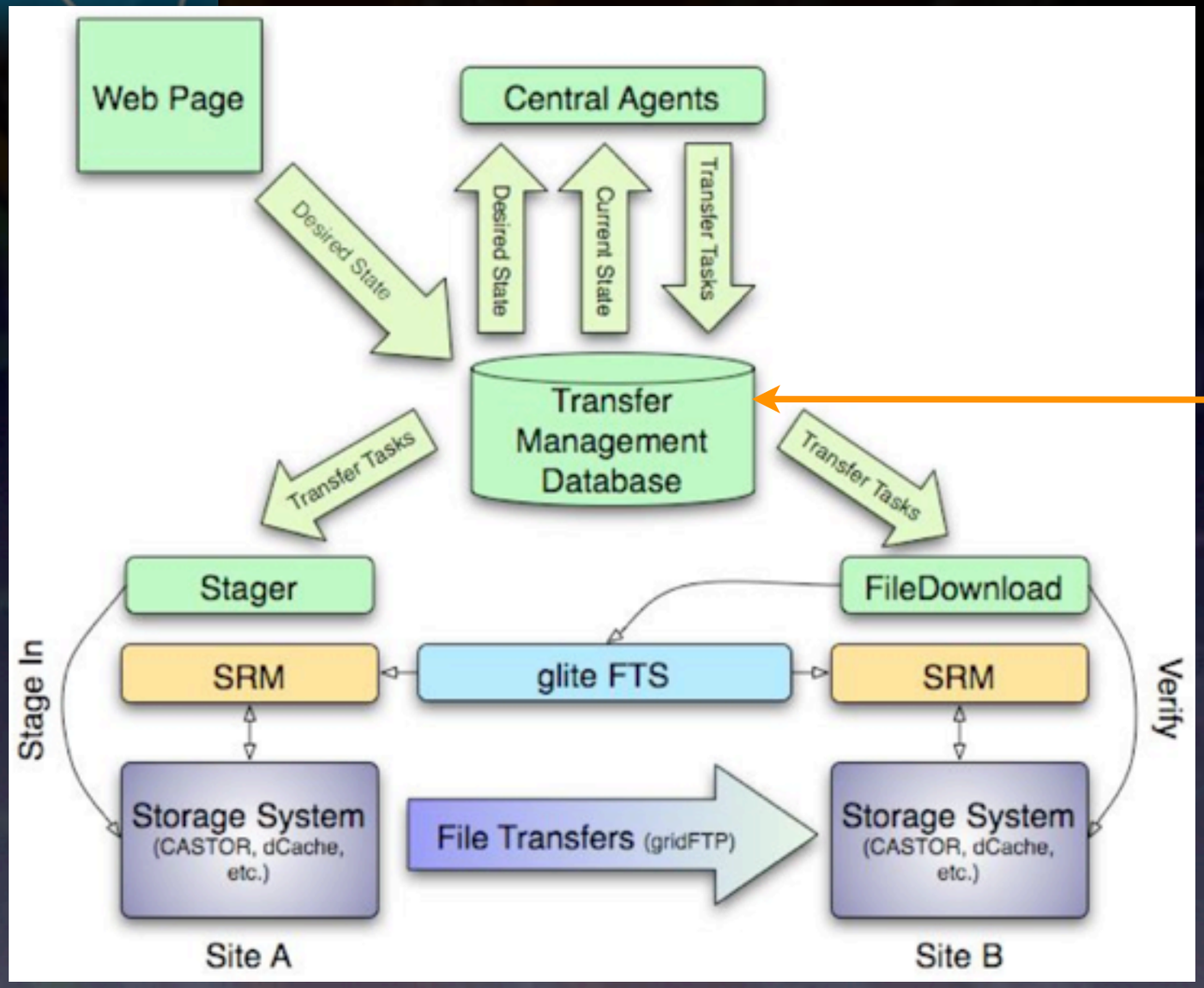
- The Physics Experiment Data Export (PhEDEx) taking care of large scale data transfers across the Grid
- 30 PB (including replicas) data registered in PhEDEx
- About 150 TB transfered each day
- Daily in use since 2004 and extremely reliable and scales well beyond the specifications requested by CMS
- Over the years PhEDEx has evolved from a ToolKit for transfers to a collection of open frameworks
  - PhEDEx data service: web-service to access information (JSON, XML or Perl format) and to submit/update tasks (inject data, subscription, request approval, ...)
  - Next-gen web-site: interacts directly with data service, information displayed using JavaScript YUI)
  - Agent-lite framework: writing lightweight agents even for non-PhEDEx tasks/outside CMS
  - Namespace framework: Interface between high level and low level activities (like checking existence of files on SE vs. optimized access to SE in PhEDEx block consistency checks)
  - LifeCycle Agent: Driving a system through a sequence of events (processing of payloads). Very useful for debugging, release validation and scalability-, stress- and integration-testing
    - Also used/planned to be used for testing DBS 3 and CRAB 3

PhEDEx



Nicolo Magini

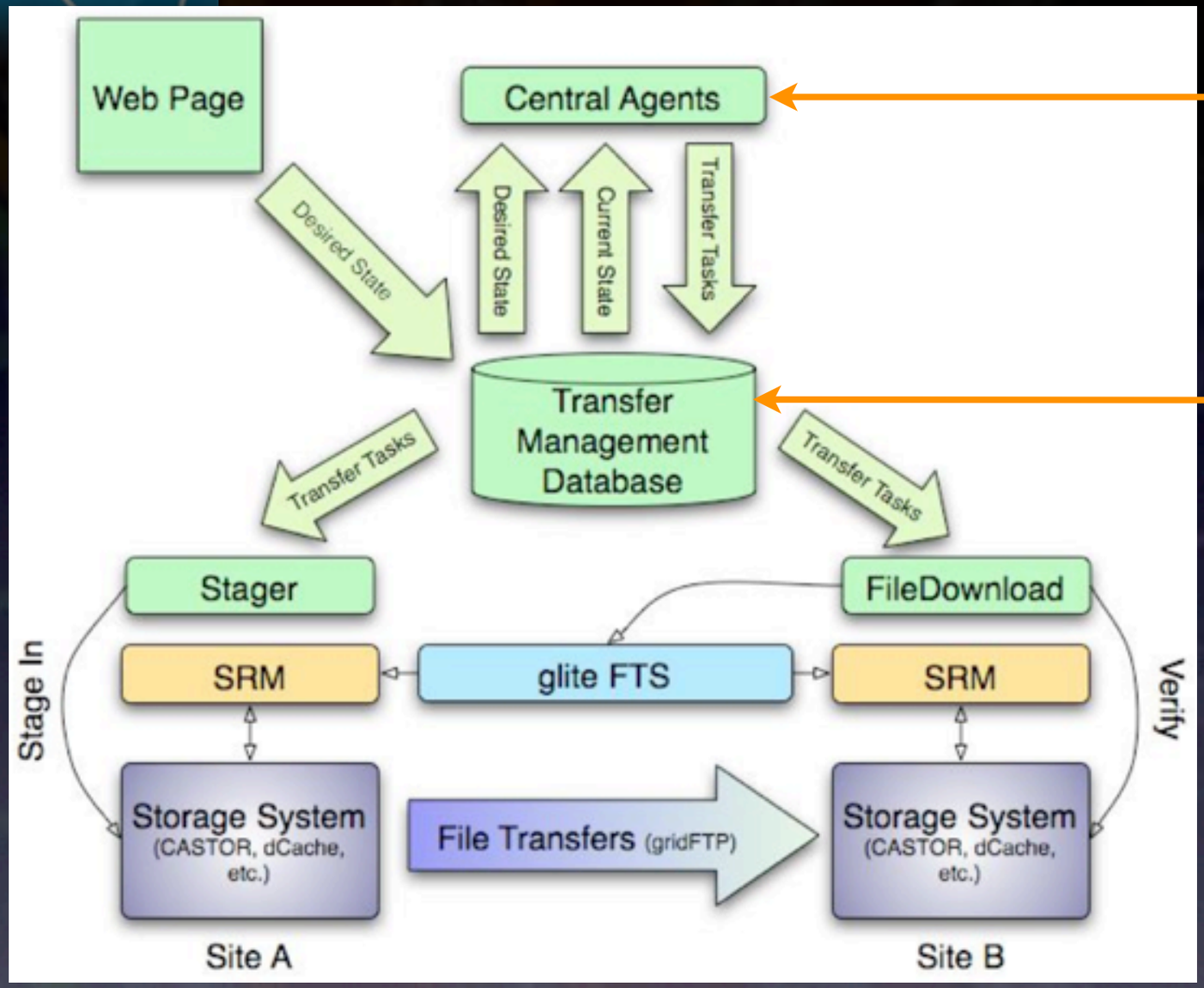
PhEDEx



Nicolo Magini

Oracle Database as „blackboard“ for the system state (TMDB)

PhEDEx

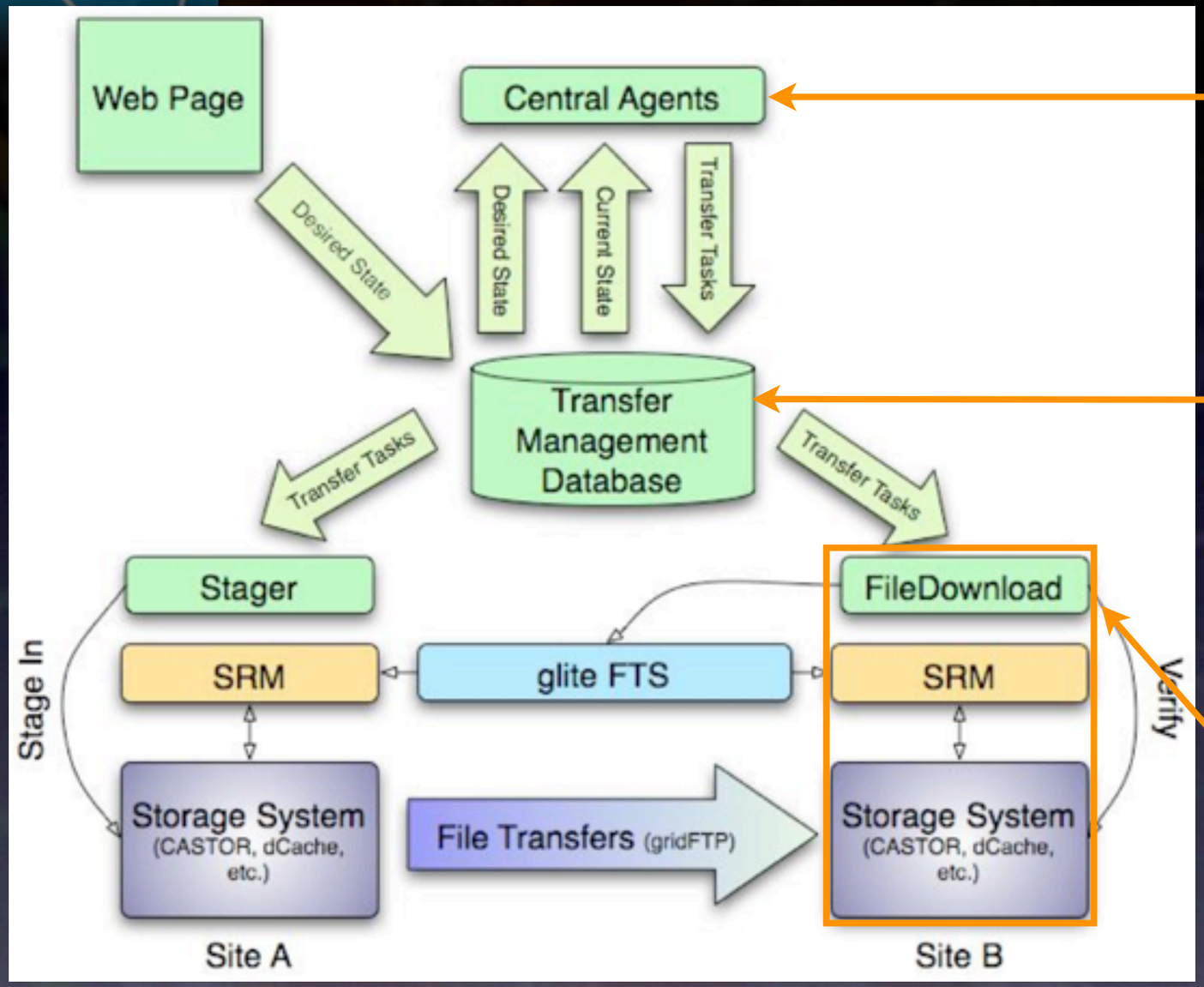


Nicolo Magini

Central Agents running at CERN responsible for data routing and transfer task creation

Oracle Database as „blackboard“ for the system state (TMDB)

PhEDEx



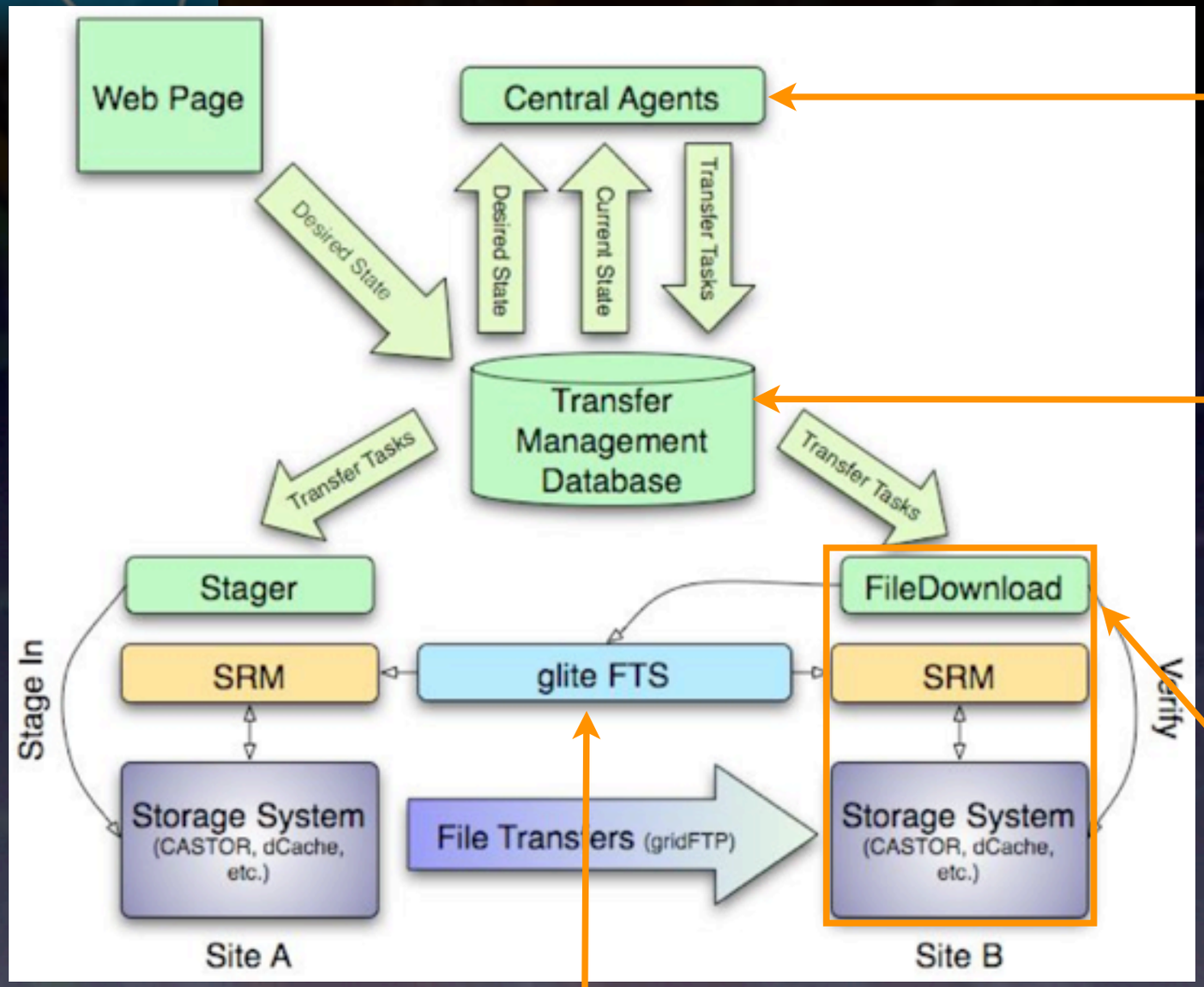
Nicolo Magini

Central Agents running at CERN responsible for data routing and transfer task creation

Oracle Database as „blackboard“ for the system state (TMDB)

Agents connect TMDB directly to get current and desired state and perform action if needed

PhEDEx



Nicolo Magini

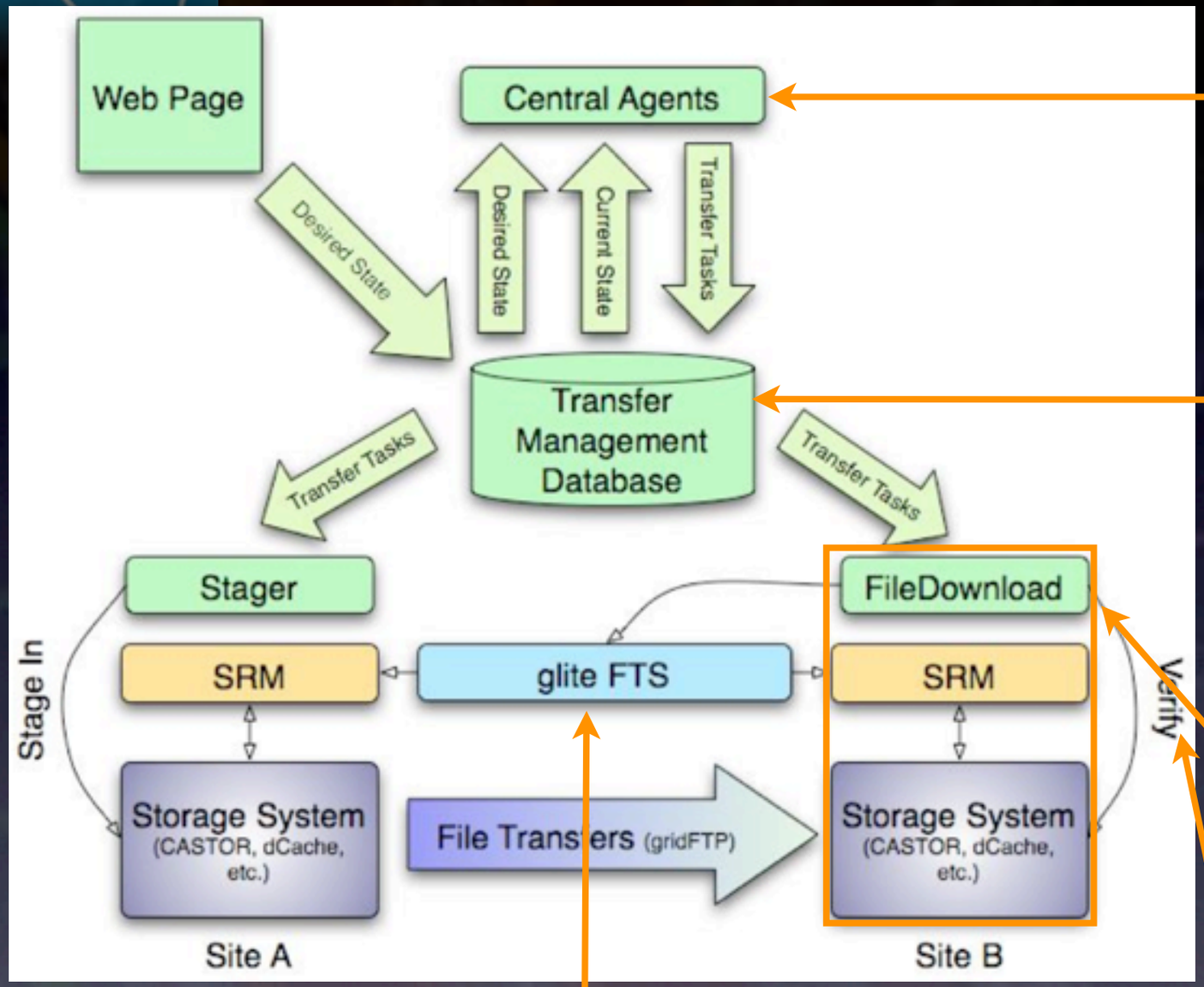
Central Agents running at CERN responsible for data routing and transfer task creation

Oracle Database as „blackboard“ for the system state (TMDB)

Agents connect TMDB directly to get current and desired state and perform action if needed

Utilize File Transfer Service (FTS) provided by the glite middleware

PhEDEx



Central Agents running at CERN responsible for data routing and transfer task creation

Oracle Database as „blackboard“ for the system state (TMDB)

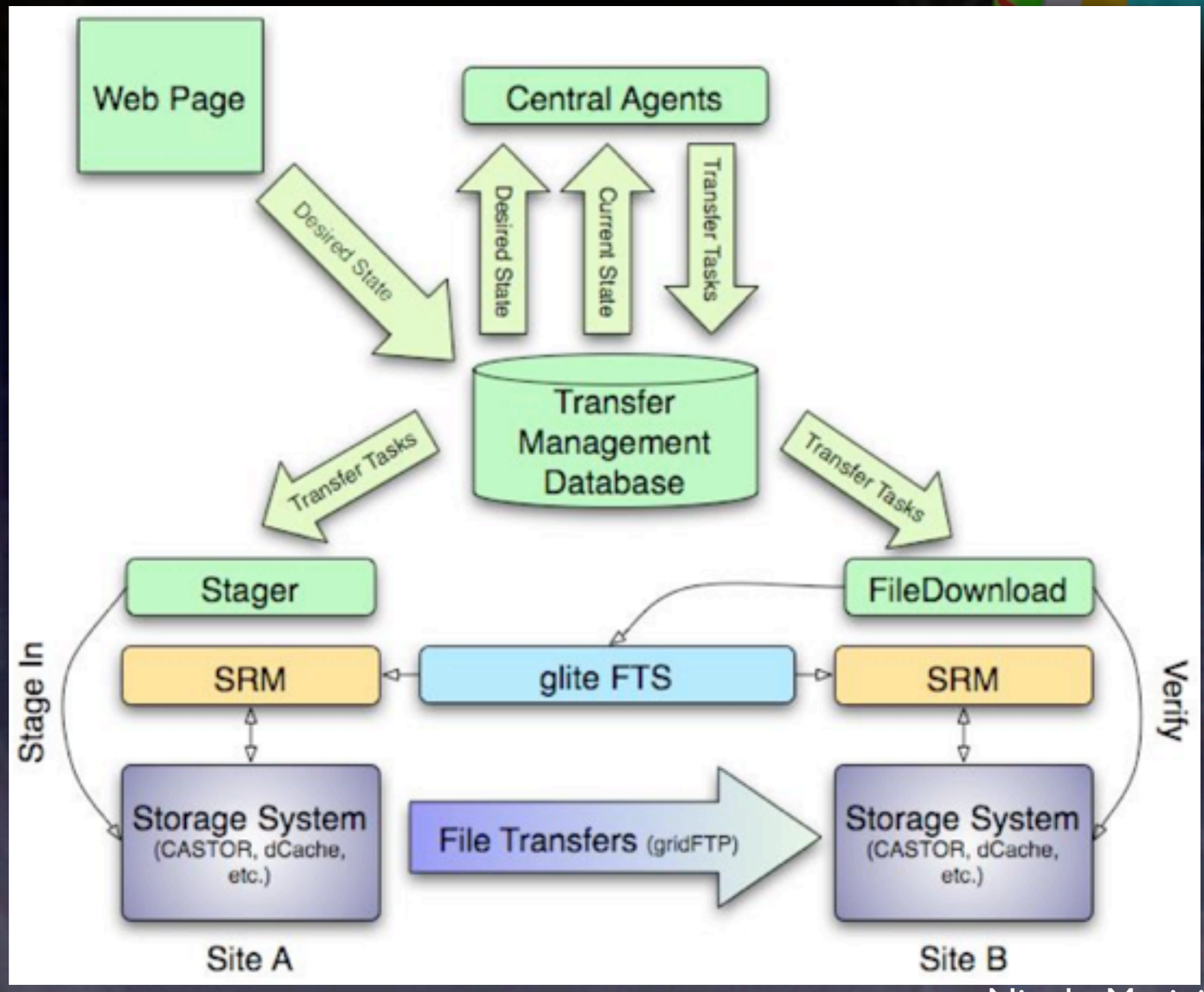
Agents connect TMDB directly to get current and desired state and perform action if needed

Nicolo Magini

Utilize File Transfer Service (FTS) provided by the glite middleware

Ensuring reliable transfers by verifying each file after transfer using site specific scripts

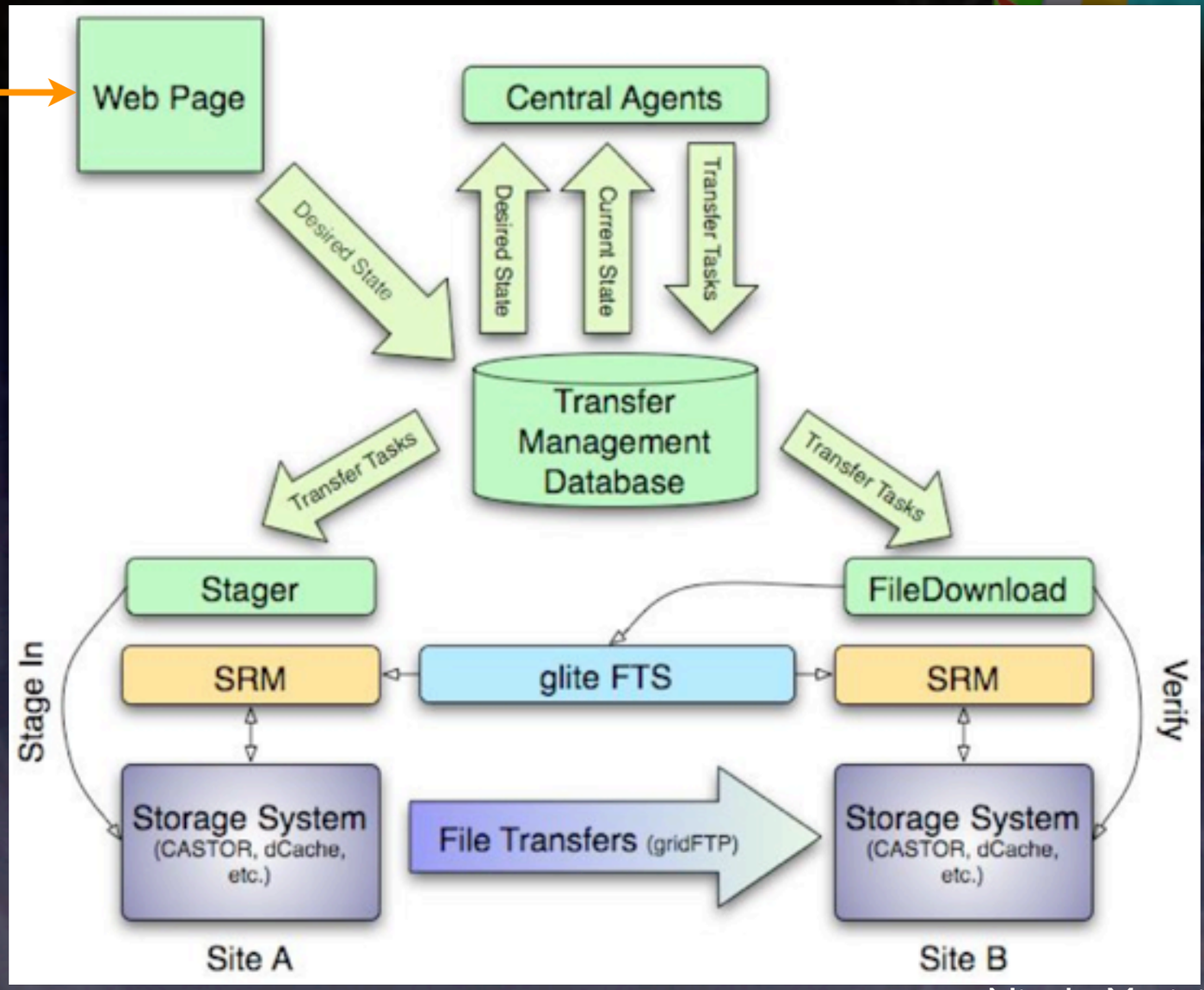
PhEDEx



Nicolo Magini

PhEDEx

User interacts with PhEDEx using web-site

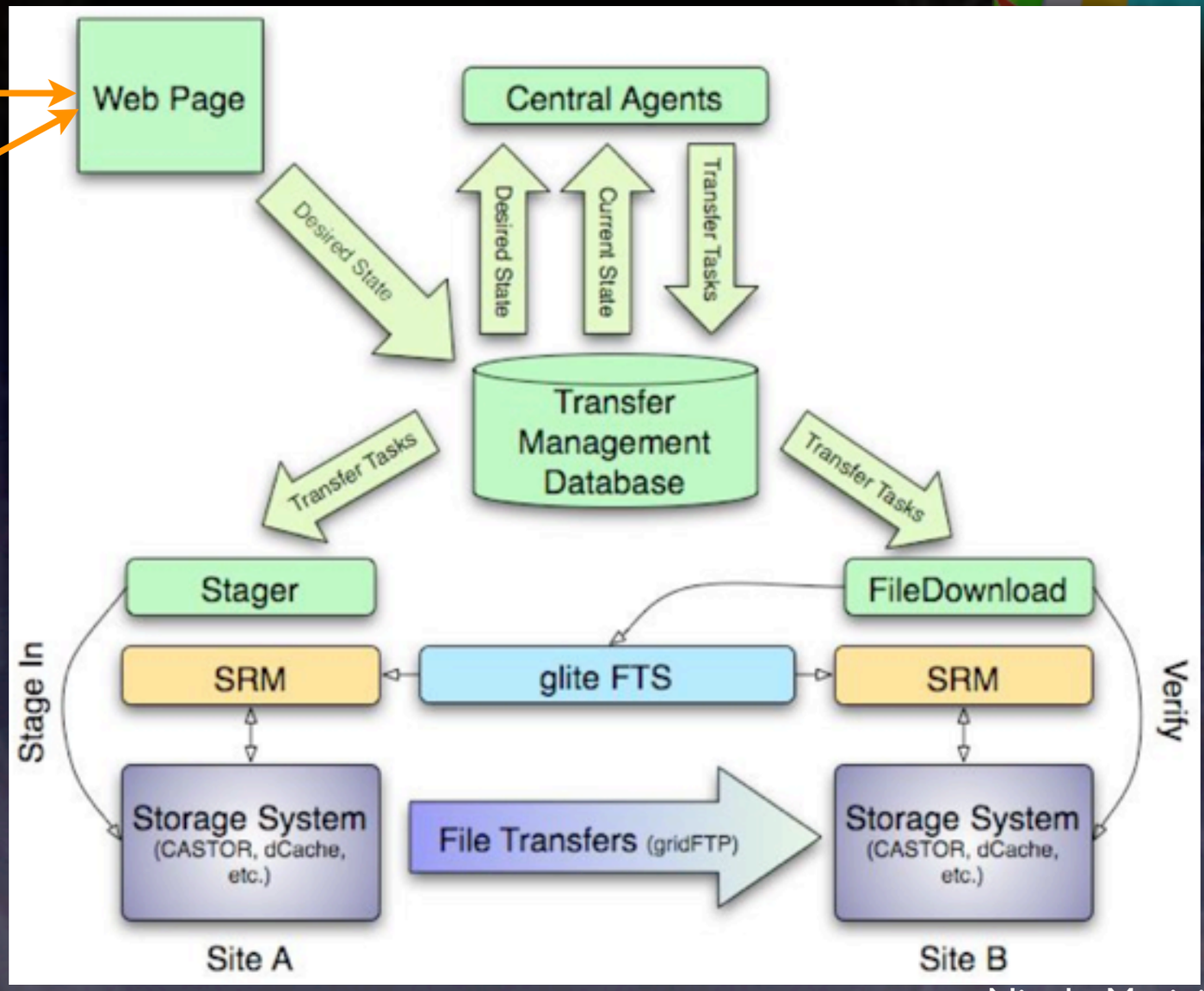


Nicolo Magini

PhEDEx

User interacts with PhEDEx using web-site

Admin(s) notified via email



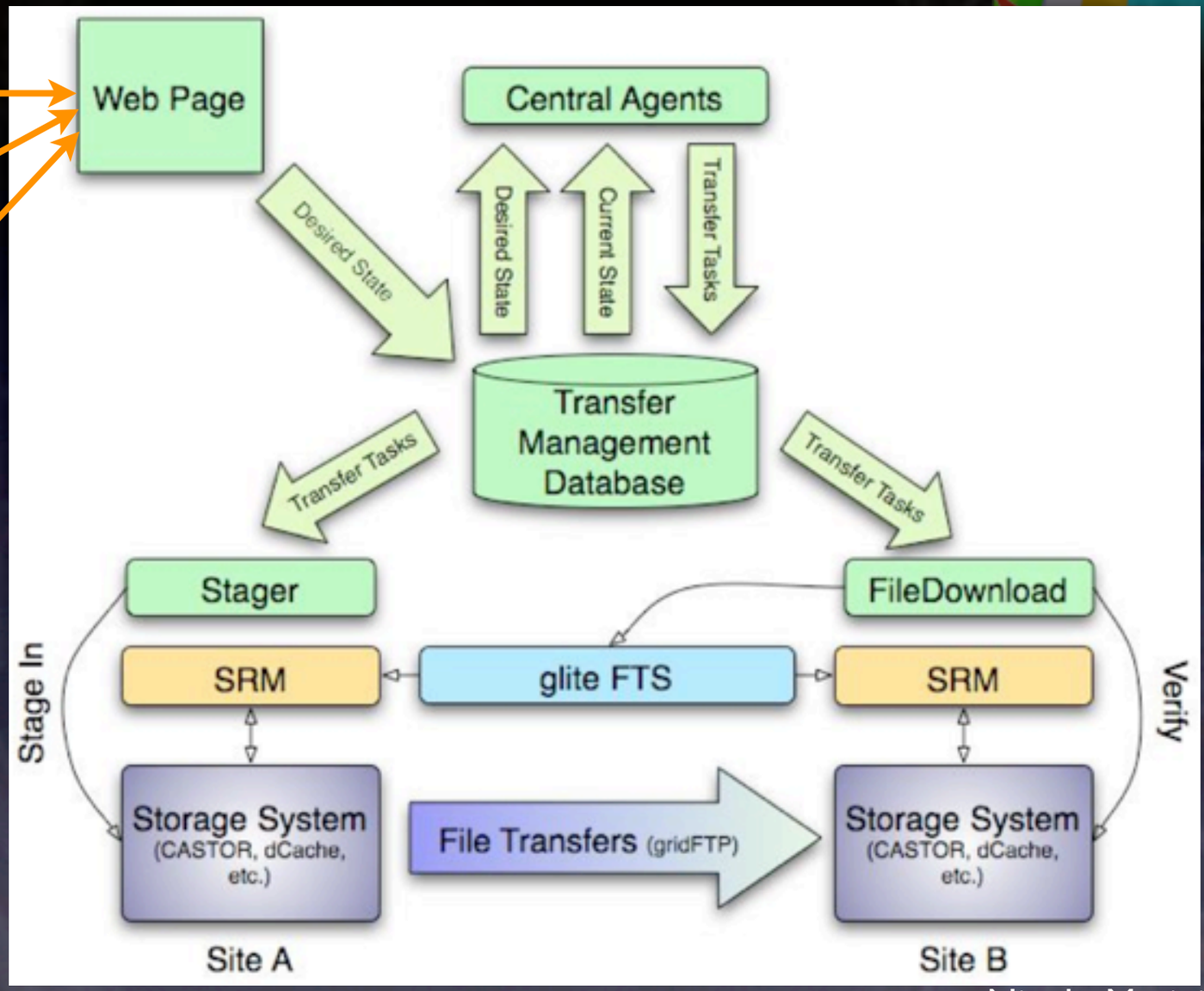
Nicolo Magini

PhEDEx

User interacts with PhEDEx using web-site

Admin(s) notified via email

Approval necessary



Nicolo Magini

PhEDEx

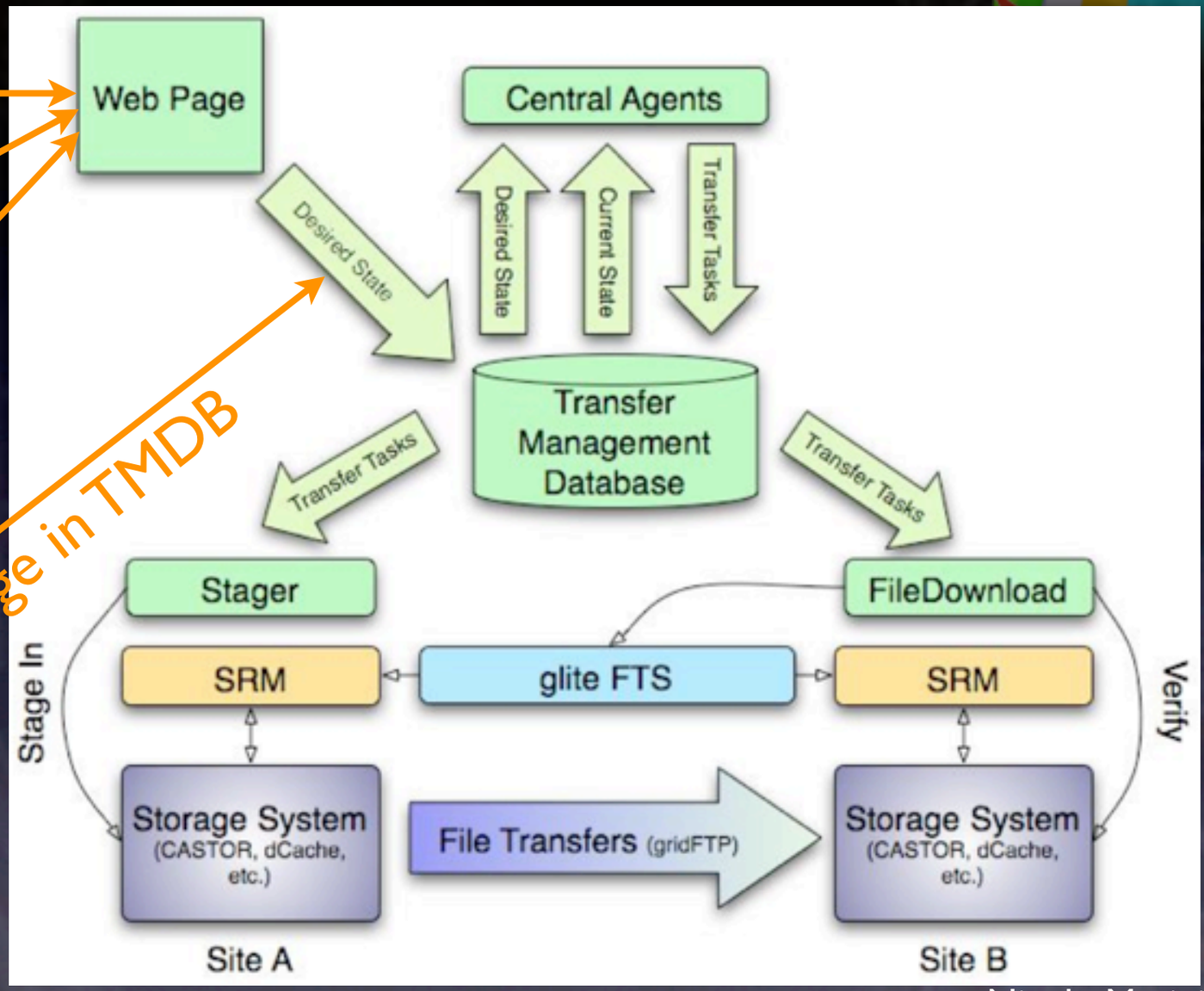
User interacts with PhEDEx using web-site

Admin(s) notified via email

Approval necessary

Transfer request becomes subscription

State change in TMDB



Nicolo Magini

PhEDEx

User interacts with PhEDEx using web-site

Admin(s) notified via email

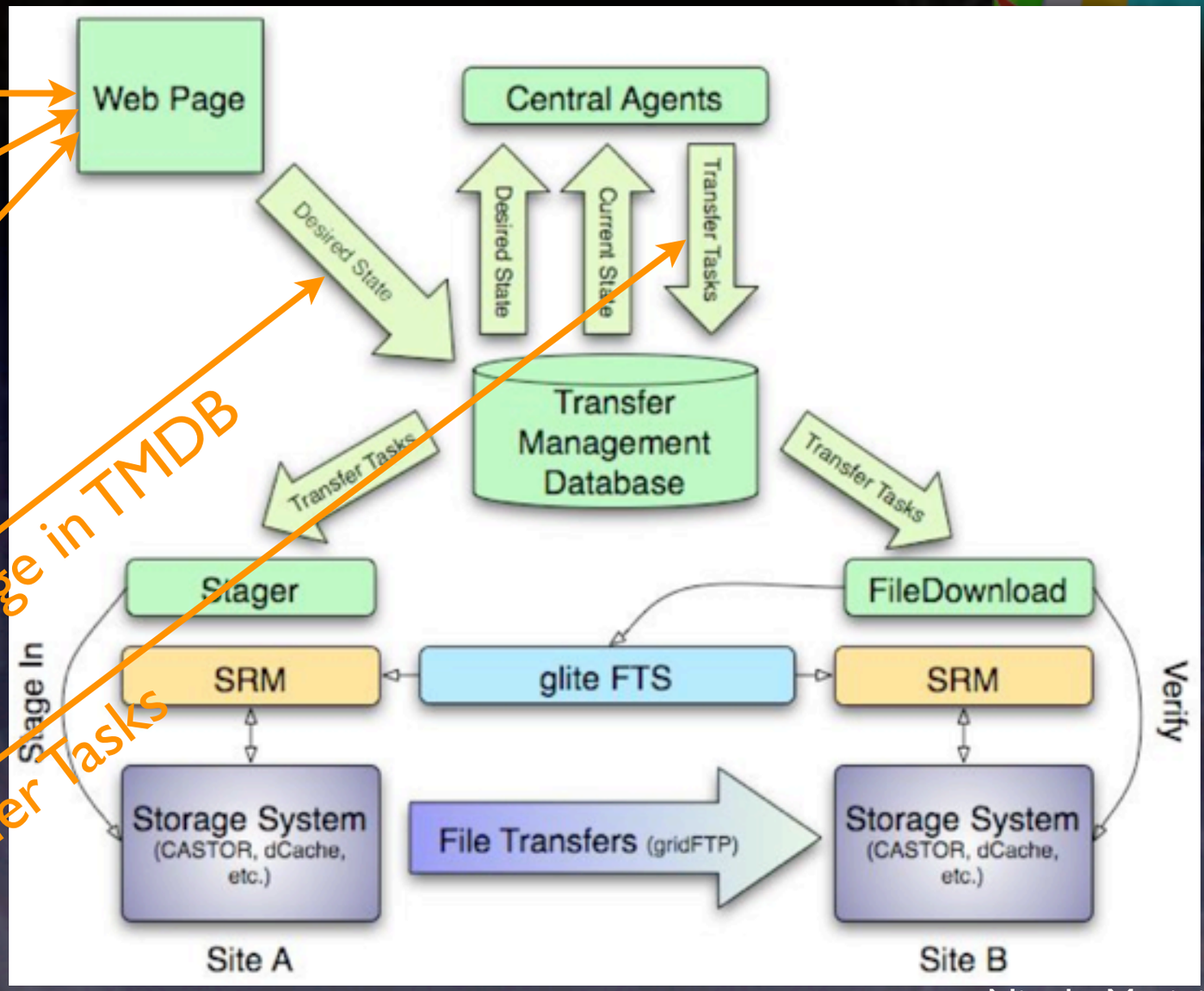
Approval necessary

Transfer request becomes subscription

CAgents recognizes desired/current states

State change in TMDB

Create Transfer Tasks



Nicolo Magini

PhEDEx

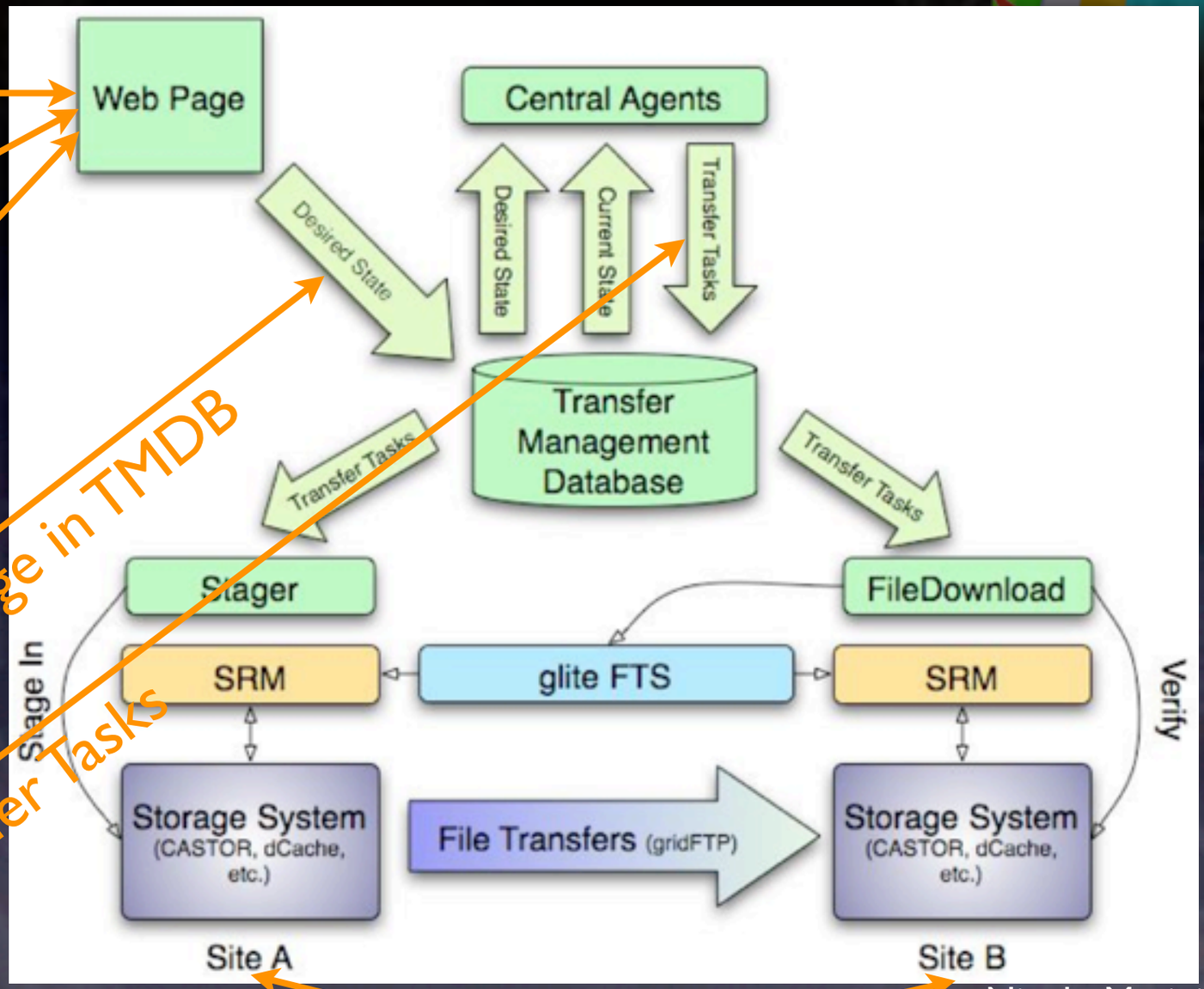
User interacts with PhEDEx using web-site

Admin(s) notified via email

Approval necessary

Transfer request becomes subscription

CAgents recognizes desired/current states



Nicolo Magini

Download agents at sites receive tasks and initiate transfer

PhEDE

Transfer Monitoring

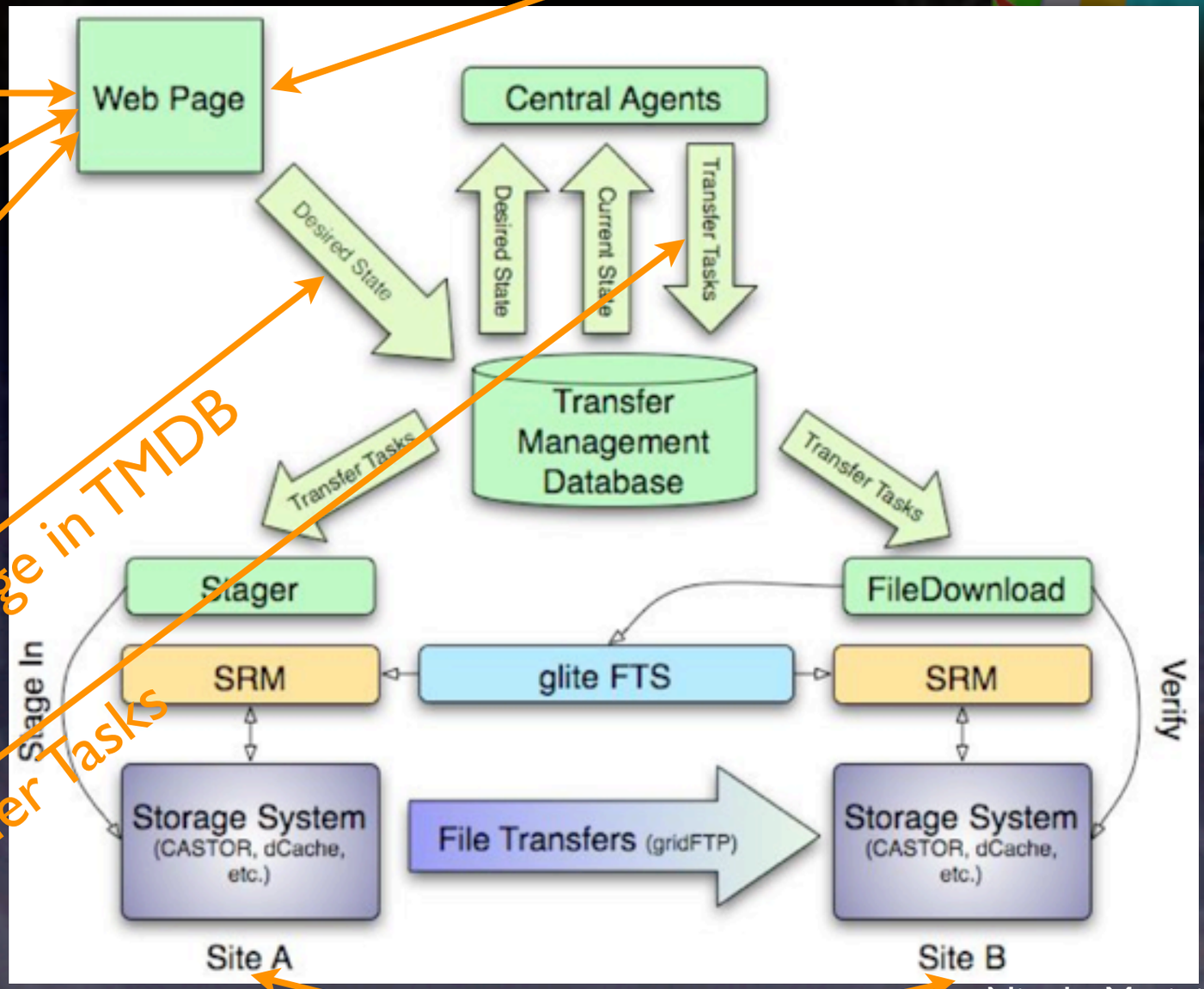
User interacts with PhEDEx using web-site

Admin(s) notified via email

Approval necessary

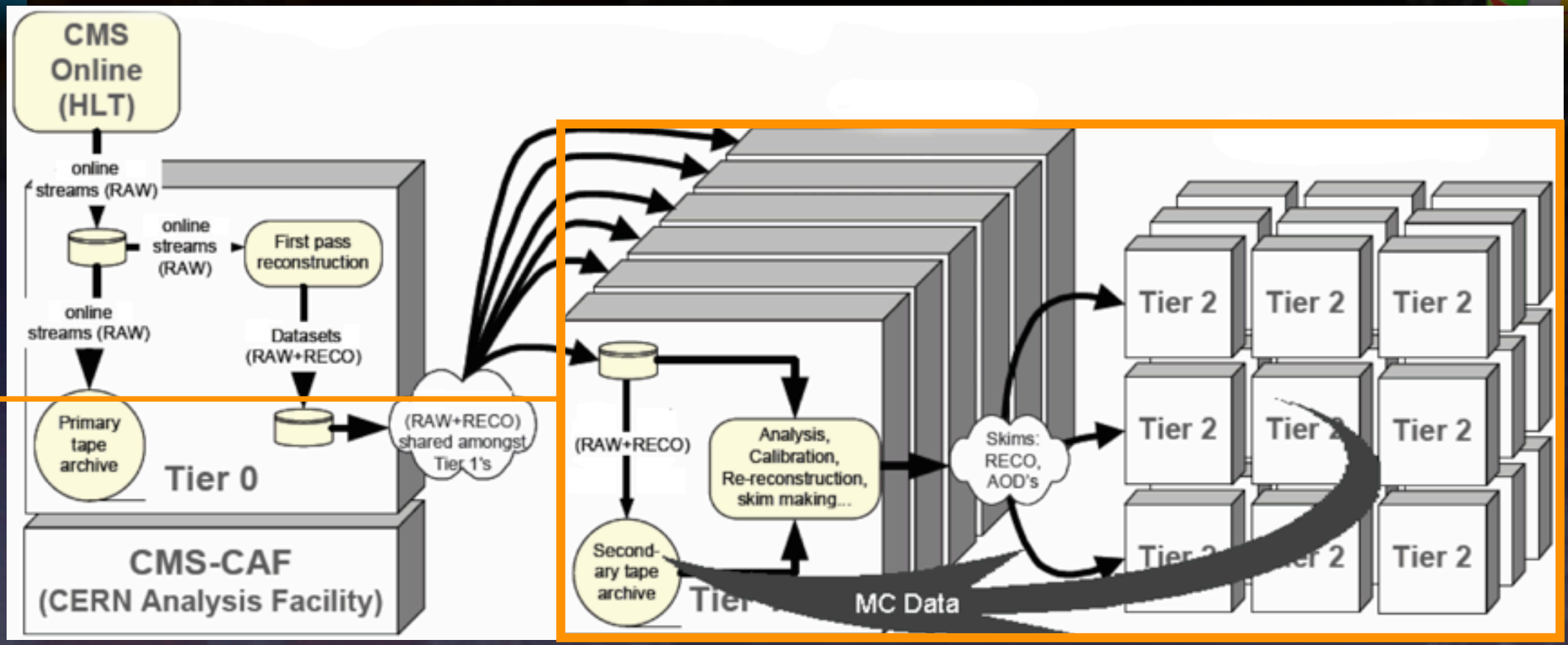
Transfer request becomes subscription

CAgents recognizes desired/current states



Nicolo Magini

Download agents at sites receive tasks and initiate transfer



Computing Model TDR 2005 w/o recent evolution

Distributed Processing

WMAgent

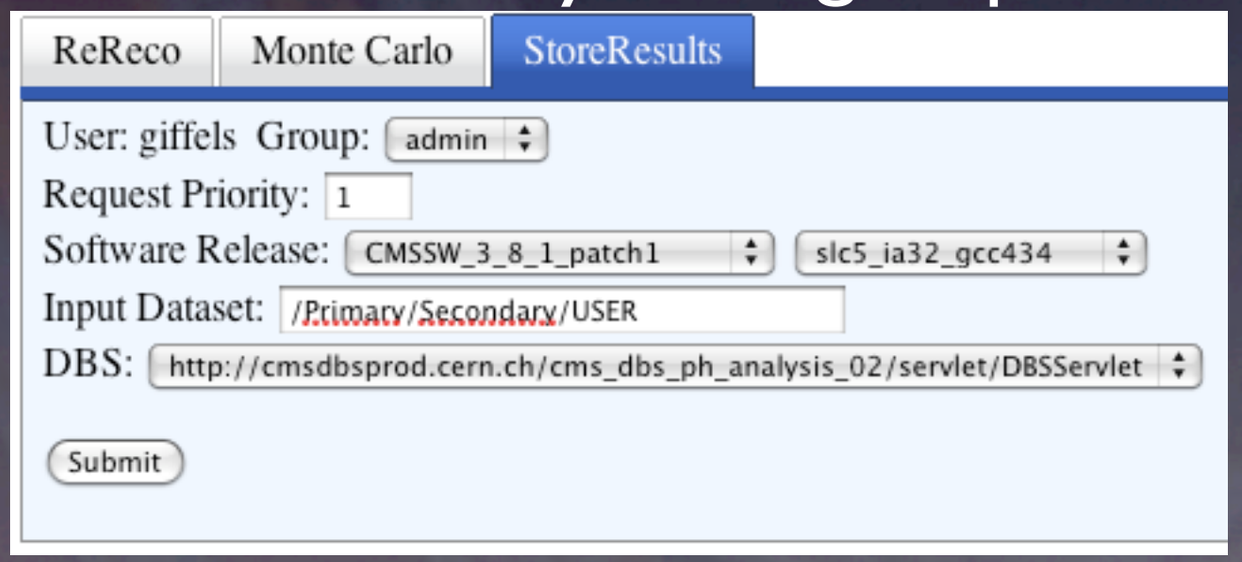
Evolution from ProdAgent to WMAgent:

- ProdAgent mainly designed for simulating data and was adapted for data processing
  - Shortcomings led to a small but significant failure rate (Does not matter for MC!)
- Many instances responsible for single activities w/o links in between
- Required manual feeding of work per instance by an operator
- Only some code is shared between TIER 0, ProdAgent and the analysis system (duplicated implementation of functionality)
- Experiences led to the decision to design a new system

- New system should act as common layer for all WM workflows
- Central system managing all distributed computing workflows
- Still separated agents submitting and managing jobs (increased scalability and reliability)

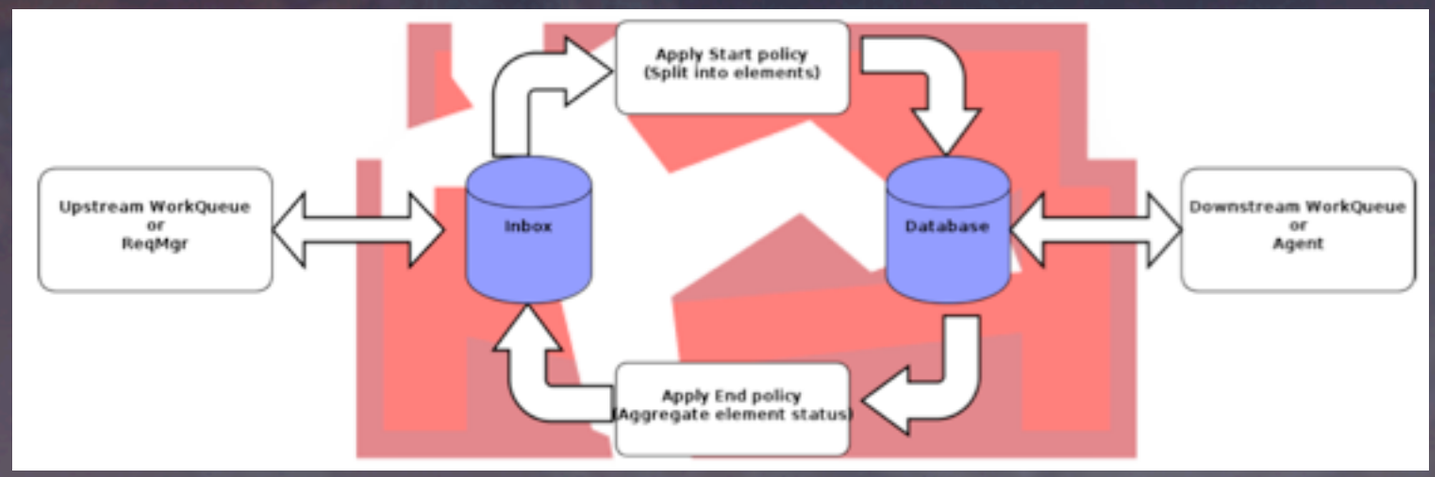
Request Manager

- ReqMgr was introduced as central entry point for all request creations and monitoring
  - increased traceability and provence
- Creates a workflow distributed to WMAgent
- RESTful web-service
- CouchDB (NoSQL) to store workflows and CMSSW configs
- SQL DB to additional parameters to validate input (users, groups, software versions, etc.)
  - Errors can be detected already during request submission



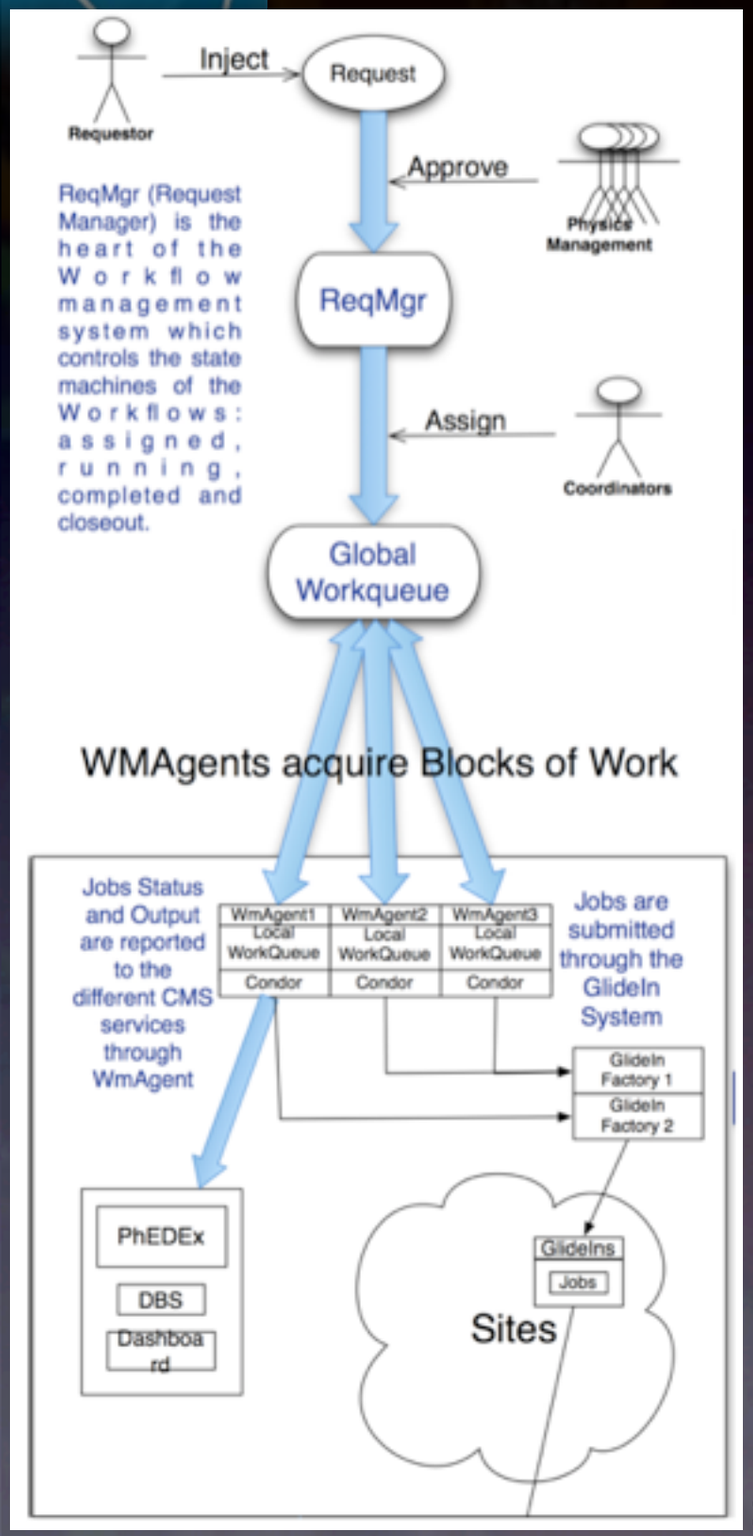
WorkQueue

- Introduced a central task queue (Global WorkQueue) to distribute work to all agents utilizing CouchDB
- Improving reliability and introduces automation
- Takes requests from ReqMgr and considering priorities
- WorkQueue splits the request into chunks of work (Blocks of files)
- Feeds work to the local WorkQueue of the best suited WMAgent
- Local WorkQueue introduced to reduce latency, for example if global is not available
- ReqMgr, global WorkQueue and local WorkQueue kept in sync using bi-directional replication provided by CouchDB



S. Wakefield

WMAgent

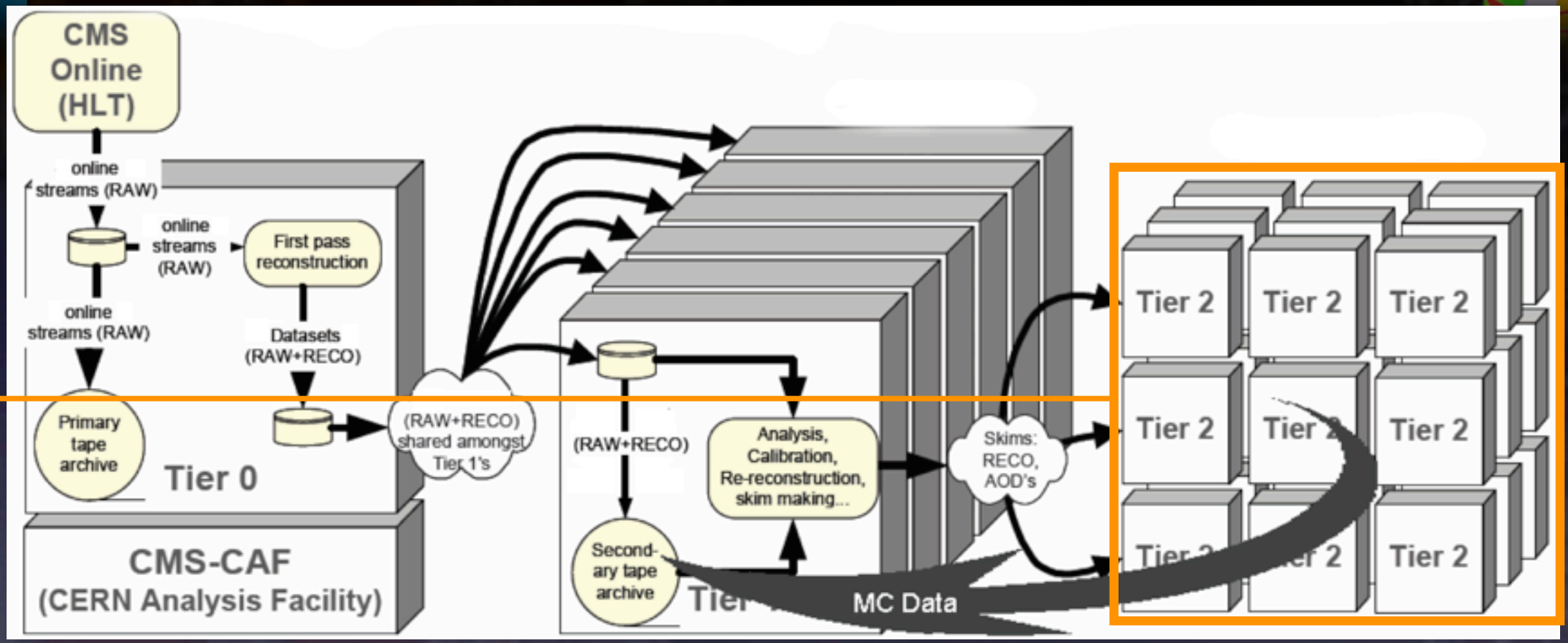


Improvements in a nutshell:

	ProdAgent	WMAgent
Setup	Autonomous Instances	One integrated system
Submission	gLite+glidein	glidein allows better error handling (gLite and local batch submission is supported)
Design	Message based	State Machine
Error Handling	Resubmit	Intelligent resubmission depending on error
Operator	Running work	Monitoring work

WMAgent already used in production!

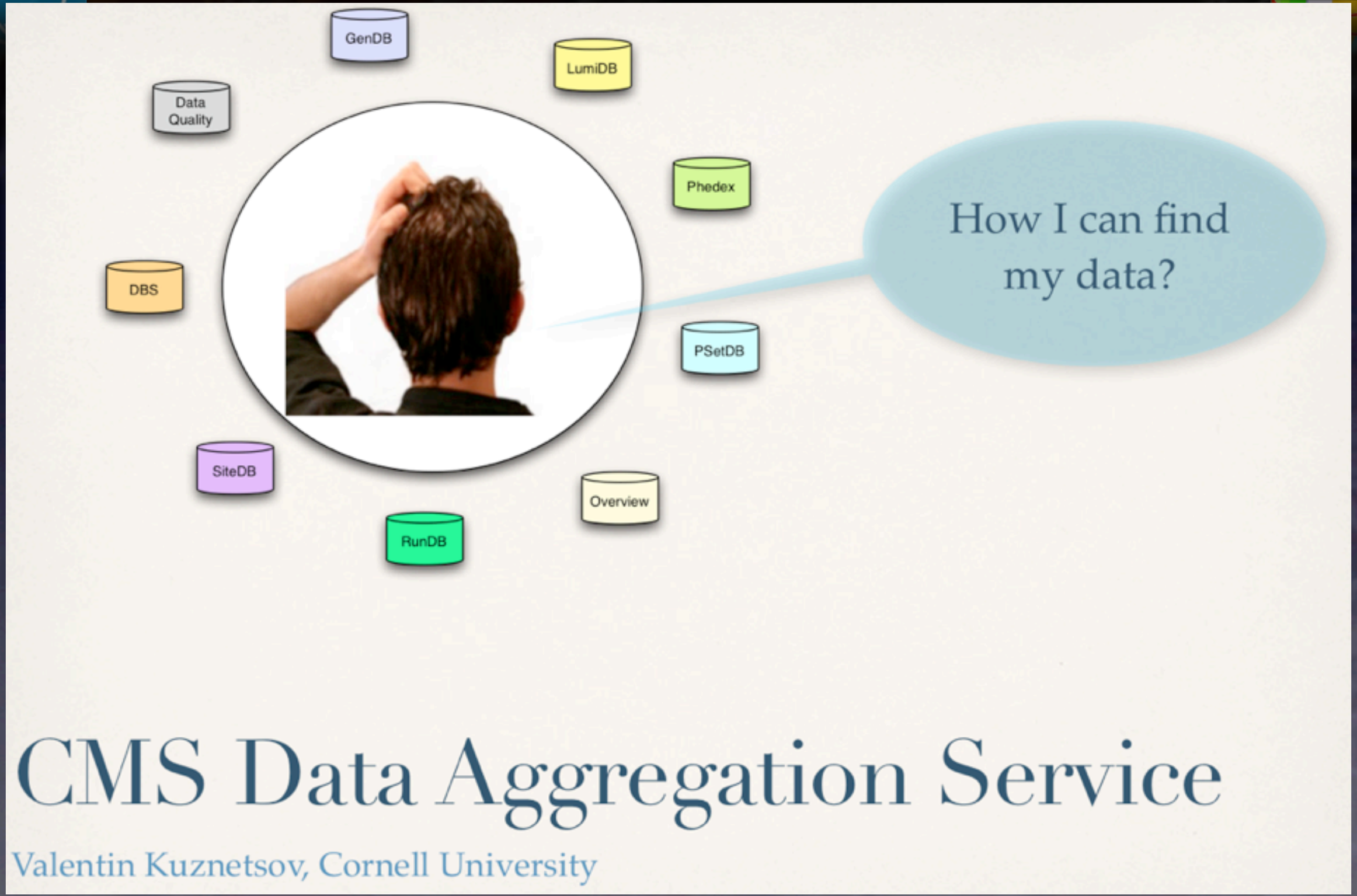
S. Wakefield



Computing Model TDR 2005 w/o recent evolution

User Analysis?

DAS

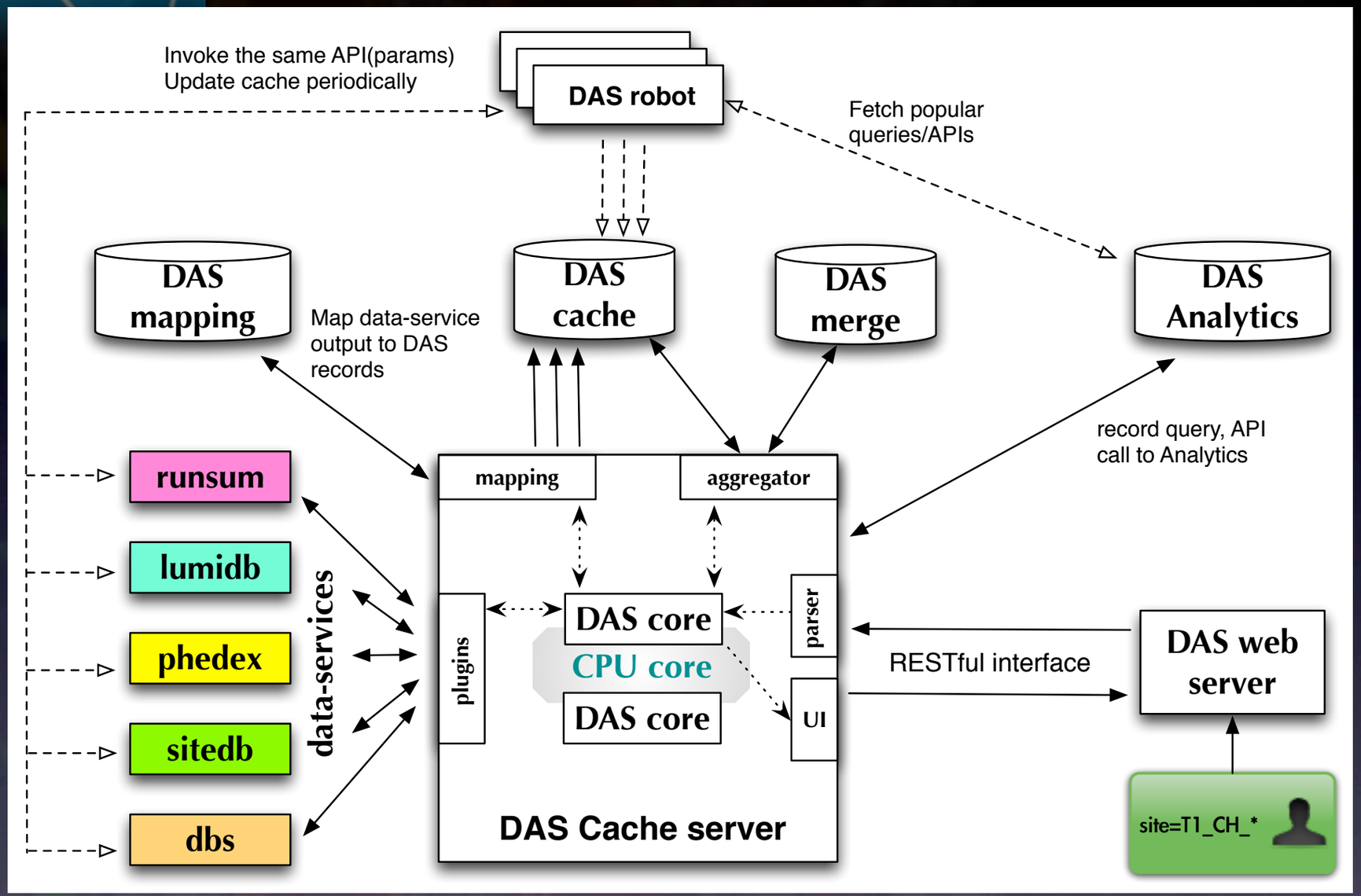

GenDB
LumiDB
Data Quality
Phedex
DBS
How I can find my data?
PSetDB
SiteDB
Overview
RunDB
CMS Data Aggregation Service
Valentin Kuznetsov, Cornell University

DAS

- Meta-data (~1TB per year) is distributed across different data services using a variety of different technologies (formats, DBs, ...)
- Usually it is necessary to combine informations stored in different locations (Which one to query?)
  - For example to find location of files belonging to a dataset
  - need to query DBS and PhEDEX
- DAS provides a single point of access for the users
- DAS knows which services to query and merges the results convient for users
- DAS provides a caching layer to reduce load on data services
- User queries are analysed by DAS to spot most popular queries
  - Pre-fetched and update those queries in cache

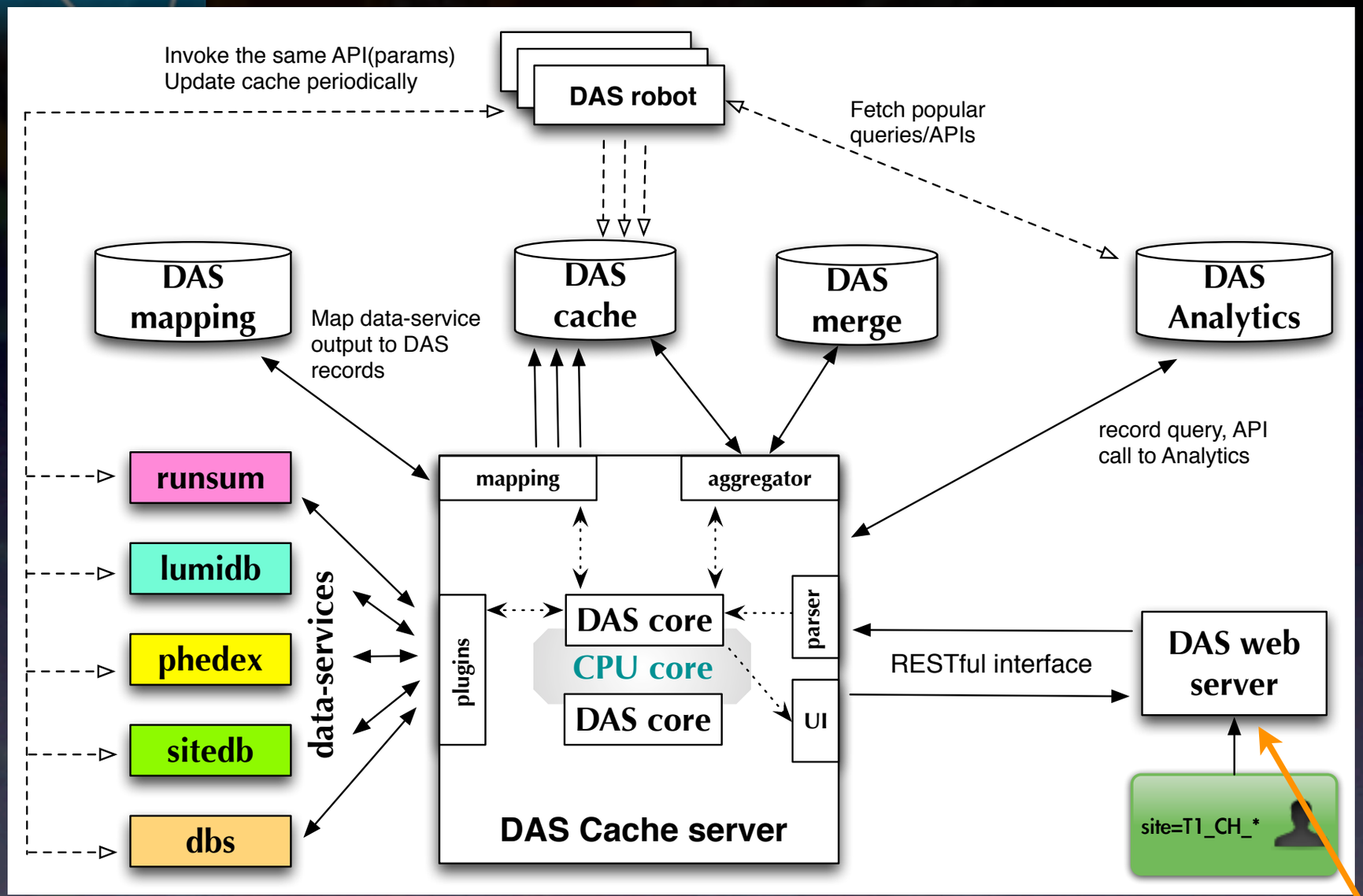
DAS

V. Kuznetsov



DAS

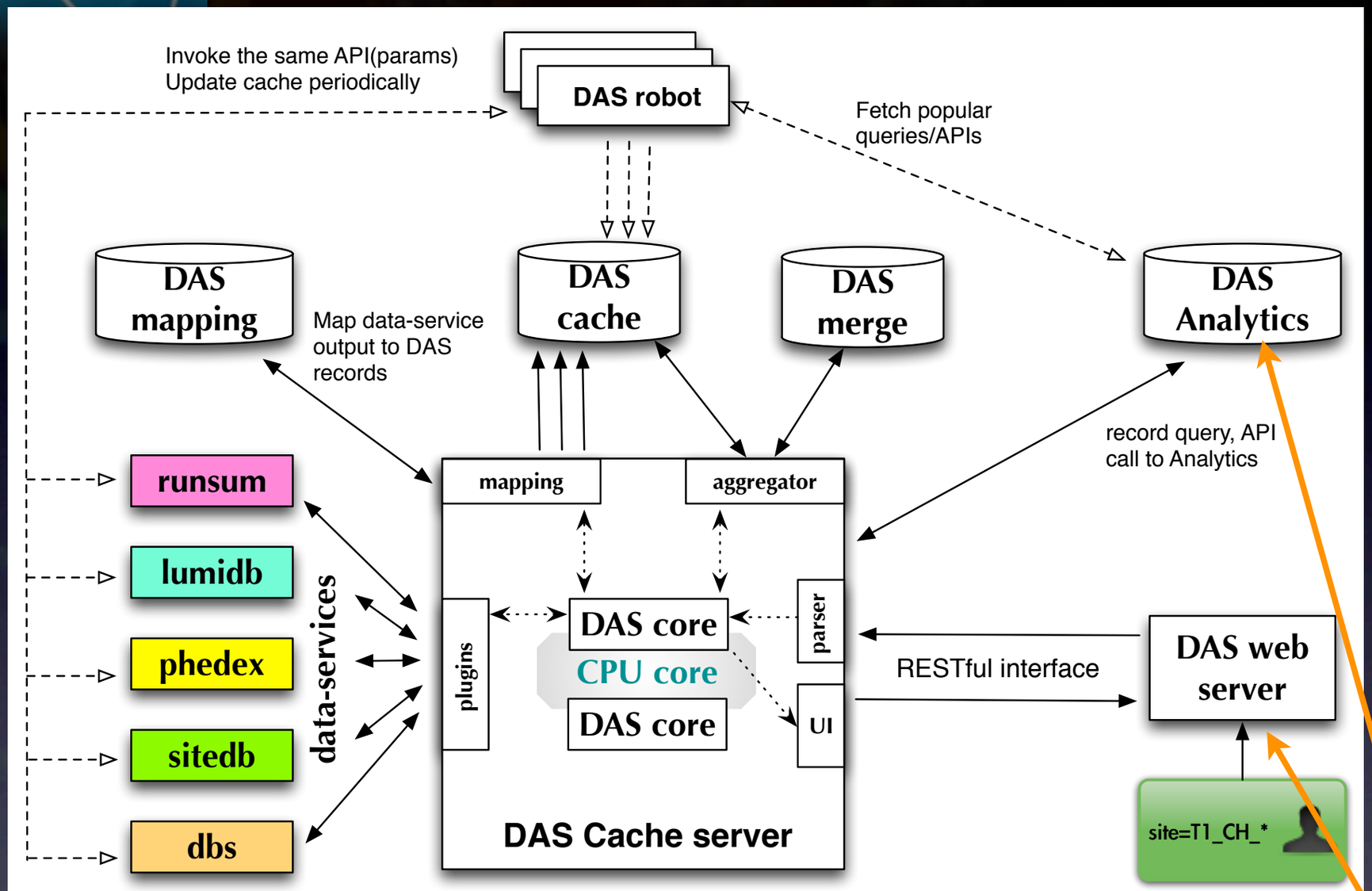
V. Kuznetsov



Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov

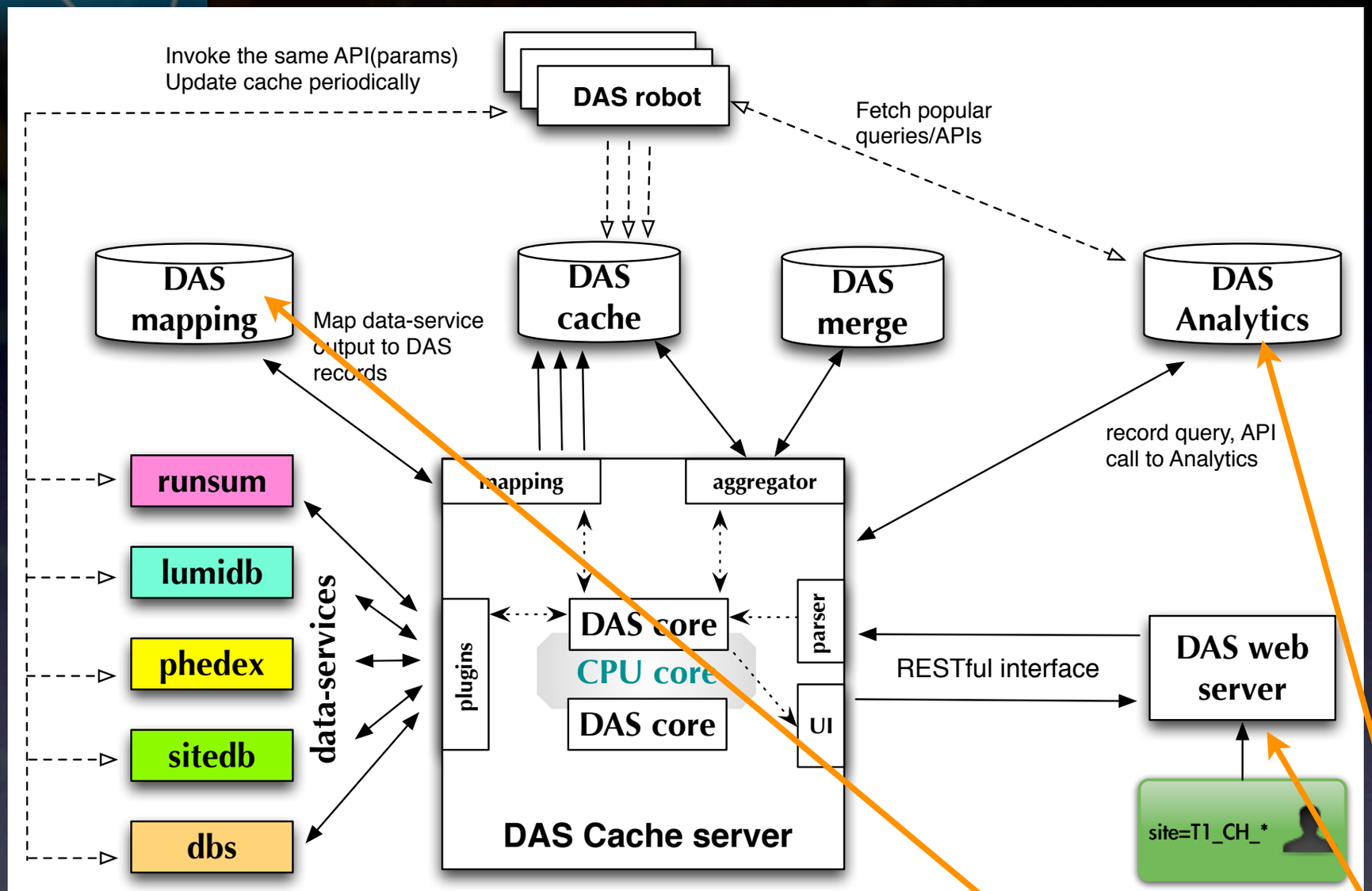


Collect user requests for system analysis and to spot popular queries pre-fetched to cache

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov



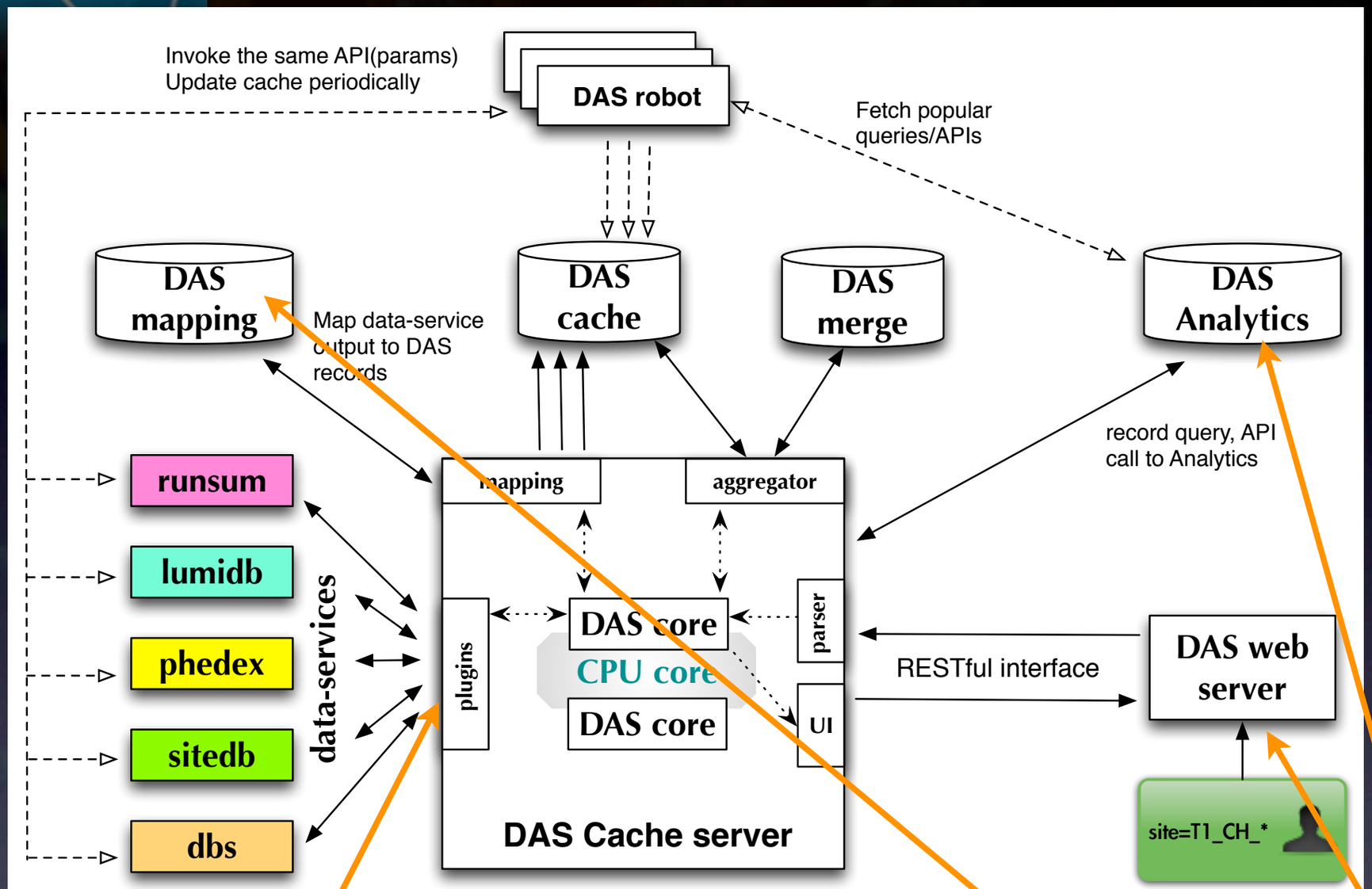
Collect user requests for system analysis and to spot popular queries pre-fetched to cache

How DAS interacts with services and interpret their results.

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov



Collect user requests for system analysis and to spot popular queries pre-fetched to cache

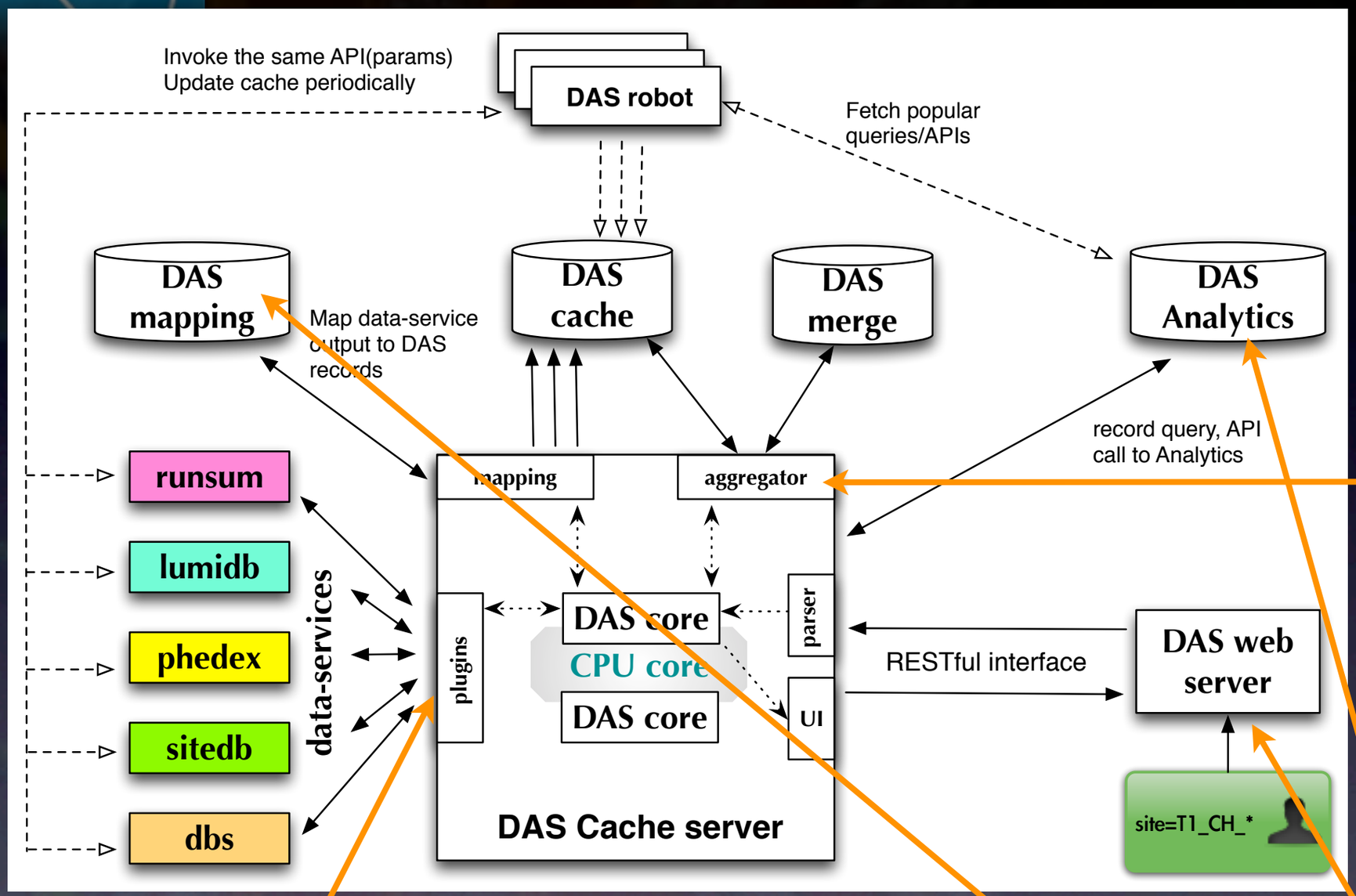
Querying data-services using their provided APIs

How DAS interacts with services and interpret their results.

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov



Aggregate meta-data from different provider like PhEDEx, DBS

Collect user requests for system analysis and to spot popular queries pre-fetched to cache

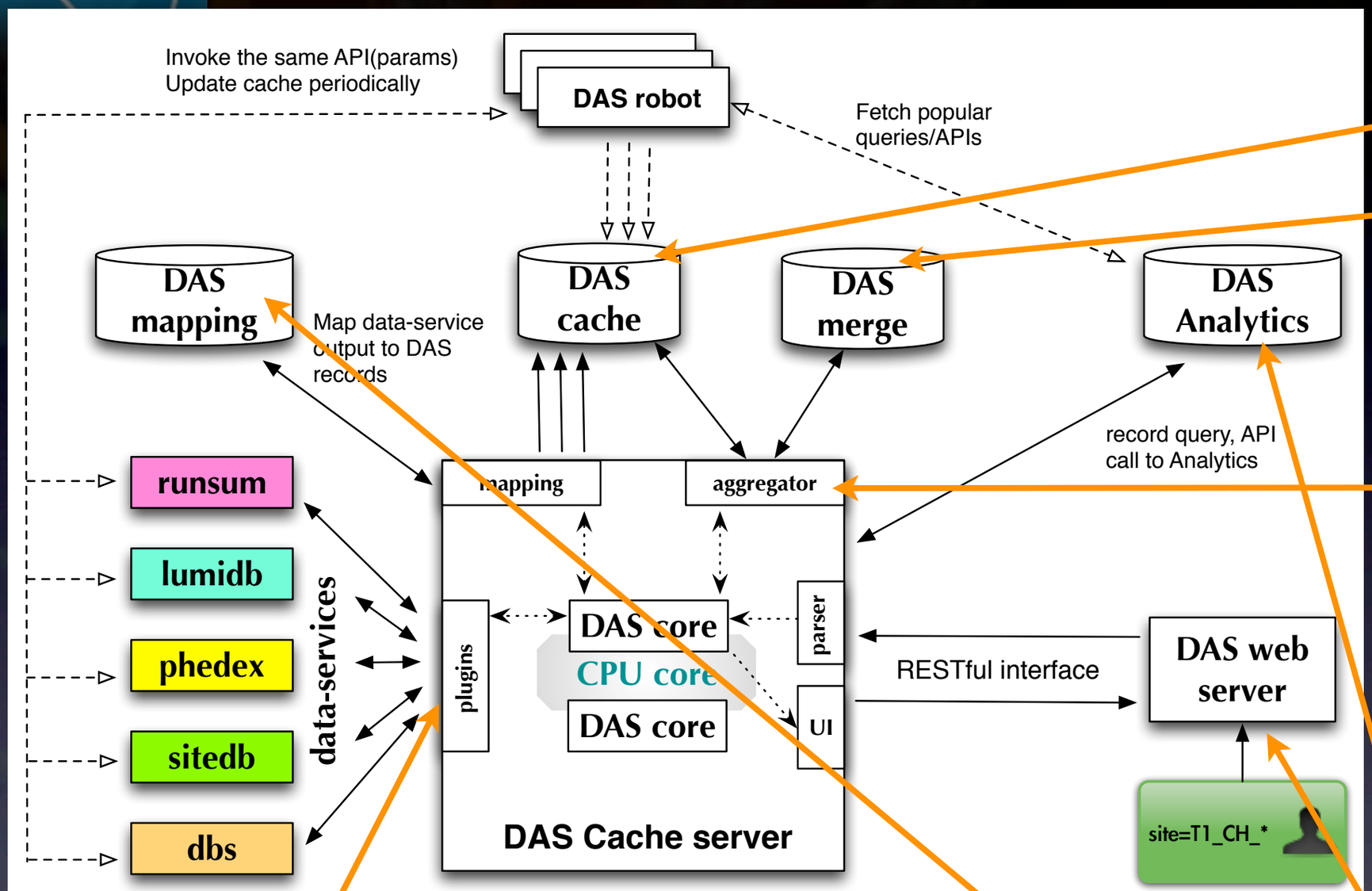
Querying data-services using their provided APIs

How DAS interacts with services and interpret their results.

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov



Cache raw data from data services as well as aggregated records

Aggregate meta-data from different provider like PhEDEx, DBS

Collect user requests for system analysis and to spot popular queries pre-fetched to cache

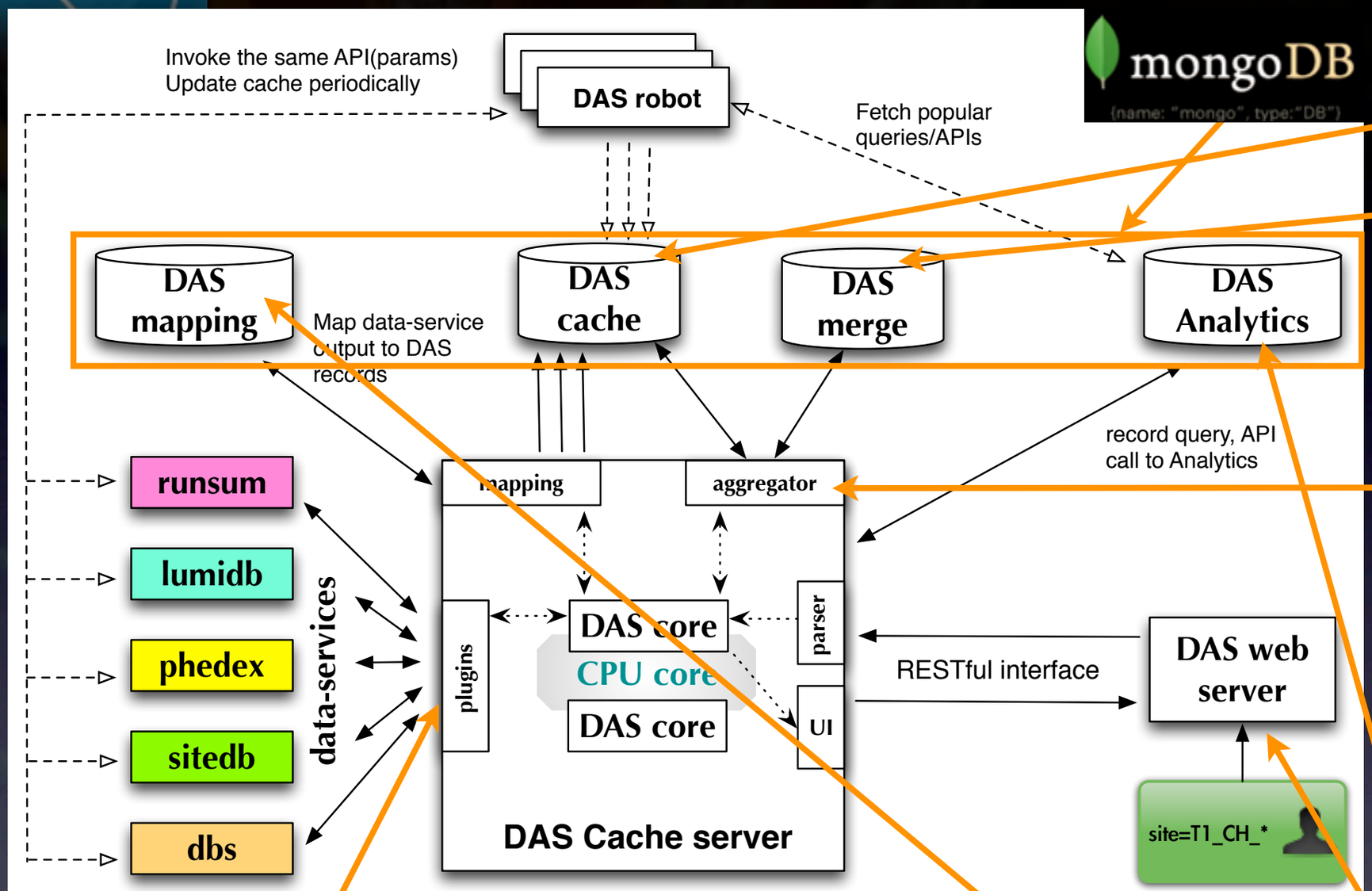
Querying data-services using their provided APIs

How DAS interacts with services and interpret their results.

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS

V. Kuznetsov



Cache raw data from data services as well as aggregated records

Aggregate meta-data from different provider like PhEDEx, DBS

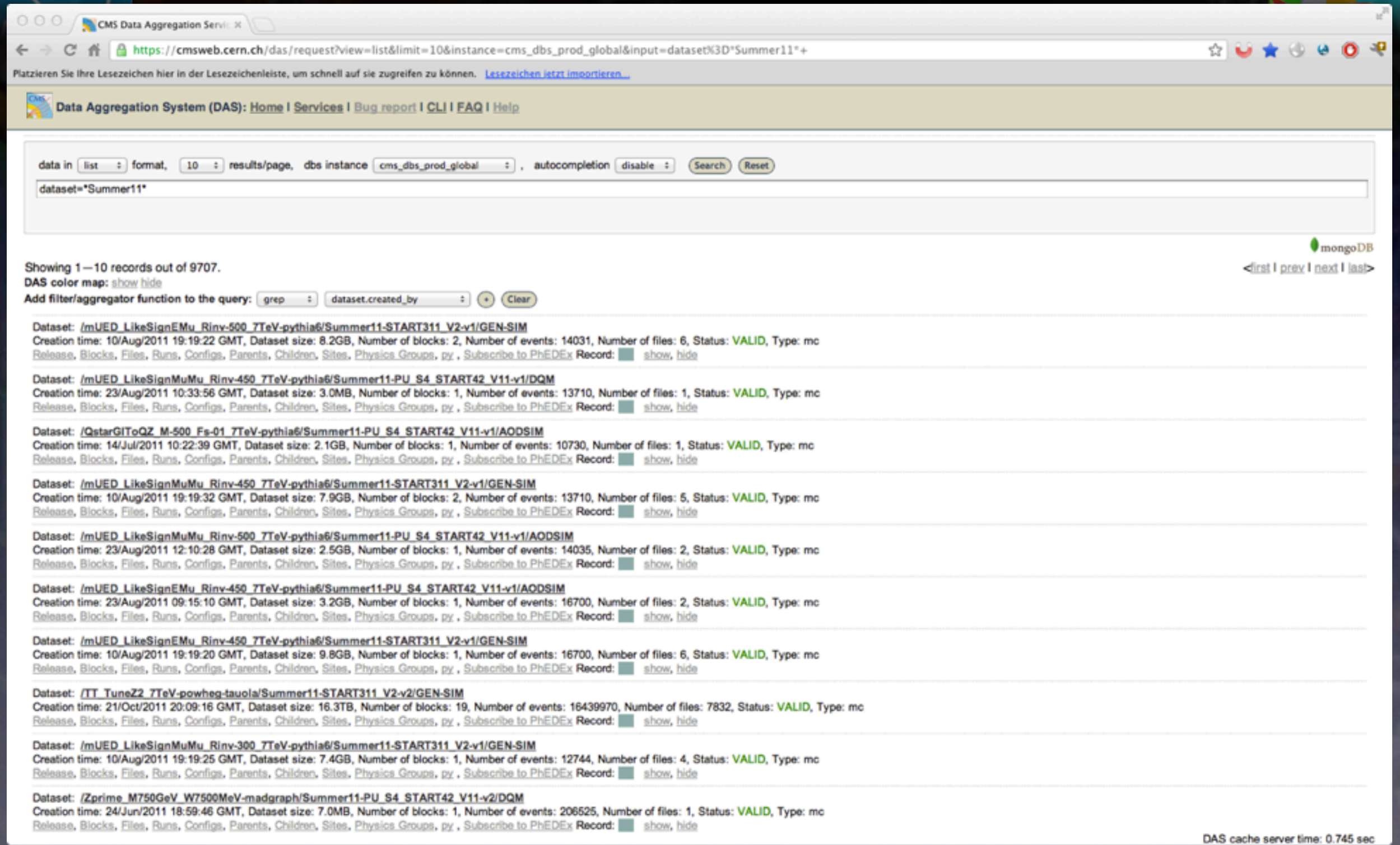
Collect user requests for system analysis and to spot popular queries pre-fetched to cache

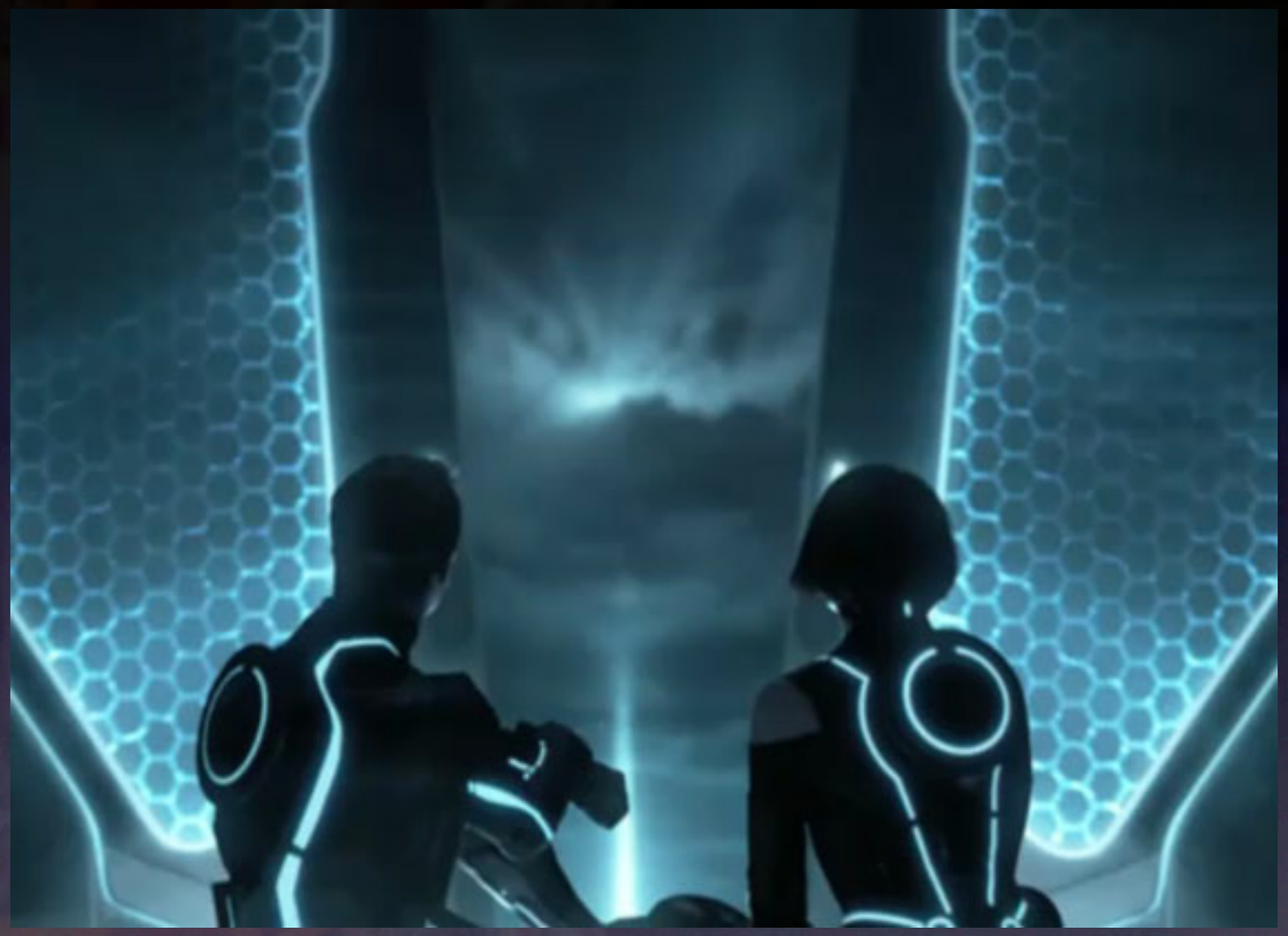
Querying data-services using their provided APIs

How DAS interacts with services and interpret their results.

Free text-based queries using web-site or CLI
like dataset=*Summer11*

DAS


CMS Data Aggregation Service
https://cmsweb.cern.ch/das/request?view=list&limit=10&instance=cms_dbs_prod_global&input=dataset%3D*Summer11*+
Platzieren Sie Ihre Lesezeichen hier in der Lesezeichenleiste, um schnell auf sie zugreifen zu können. Lesezeichen jetzt importieren...
Data Aggregation System (DAS): Home | Services | Bug report | CLI | FAQ | Help
data in list format, 10 results/page, dbs instance cms_dbs_prod_global , autocompletion disable Search Reset
dataset=*Summer11*
mongoDB
Showing 1—10 records out of 9707.
<first | prev | next | last>
DAS color map: show hide
Add filter/aggregator function to the query: grep dataset.created_by + Clear
Dataset: /mUED_LikeSignEMu_Rinv-500_7TeV-pythia6/Summer11-START311_V2-v1/GEN-SIM
Creation time: 10/Aug/2011 19:19:22 GMT, Dataset size: 8.2GB, Number of blocks: 2, Number of events: 14031, Number of files: 6, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignMuMu_Rinv-450_7TeV-pythia6/Summer11-PU_S4_START42_V11-v1/DQM
Creation time: 23/Aug/2011 10:33:56 GMT, Dataset size: 3.0MB, Number of blocks: 1, Number of events: 13710, Number of files: 1, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /QstarGIToQZ_M-500_Fs-01_7TeV-pythia6/Summer11-PU_S4_START42_V11-v1/AODSIM
Creation time: 14/Jul/2011 10:22:39 GMT, Dataset size: 2.1GB, Number of blocks: 1, Number of events: 10730, Number of files: 1, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignMuMu_Rinv-450_7TeV-pythia6/Summer11-START311_V2-v1/GEN-SIM
Creation time: 10/Aug/2011 19:19:32 GMT, Dataset size: 7.9GB, Number of blocks: 2, Number of events: 13710, Number of files: 5, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignMuMu_Rinv-500_7TeV-pythia6/Summer11-PU_S4_START42_V11-v1/AODSIM
Creation time: 23/Aug/2011 12:10:28 GMT, Dataset size: 2.5GB, Number of blocks: 1, Number of events: 14035, Number of files: 2, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignEMu_Rinv-450_7TeV-pythia6/Summer11-PU_S4_START42_V11-v1/AODSIM
Creation time: 23/Aug/2011 09:15:10 GMT, Dataset size: 3.2GB, Number of blocks: 1, Number of events: 16700, Number of files: 2, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignEMu_Rinv-450_7TeV-pythia6/Summer11-START311_V2-v1/GEN-SIM
Creation time: 10/Aug/2011 19:19:20 GMT, Dataset size: 9.8GB, Number of blocks: 1, Number of events: 16700, Number of files: 6, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /TT_TuneZ2_7TeV-powheg-tauola/Summer11-START311_V2-v2/GEN-SIM
Creation time: 21/Oct/2011 20:09:16 GMT, Dataset size: 16.3TB, Number of blocks: 19, Number of events: 16439970, Number of files: 7832, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /mUED_LikeSignMuMu_Rinv-300_7TeV-pythia6/Summer11-START311_V2-v1/GEN-SIM
Creation time: 10/Aug/2011 19:19:25 GMT, Dataset size: 7.4GB, Number of blocks: 1, Number of events: 12744, Number of files: 4, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
Dataset: /Zprime_M750GeV_W7500MeV-madgraph/Summer11-PU_S4_START42_V11-v2/DQM
Creation time: 24/Jun/2011 18:59:46 GMT, Dataset size: 7.0MB, Number of blocks: 1, Number of events: 206525, Number of files: 1, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx Record: show, hide
DAS cache server time: 0.745 sec

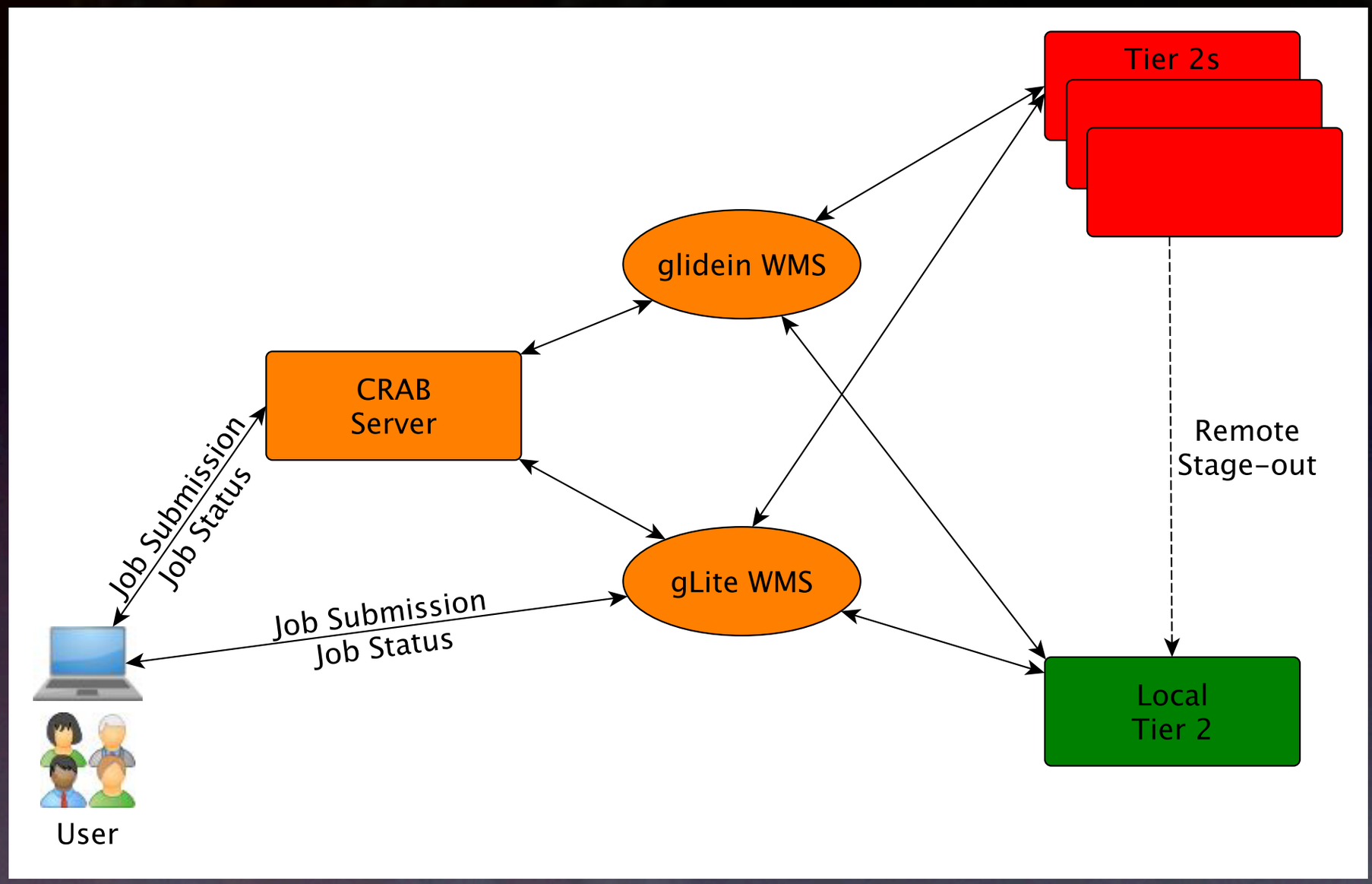


Found my data - What next?
The Grid - How to use it?

CRAB

- The CMS Remote Analysis Builder CRAB creates analysis jobs and submits them to the Grid
- CRAB hides most of the complexity of the Grid from the end-user
  - Limited knowledge of the underlaying technologies are required
- CRAB provides support for local batch systems, glite and glidein WMS for Grid submission
- Grid jobs are send to the data and user output is staged-out remotely to the users home TIER 2
- CRAB deals with about 200k jobs/day with a success rate of about 80%
- About 60% of the unsuccessful jobs are failed due to stage-out problems
- CRAB 3 based on WMAgent is currently in development

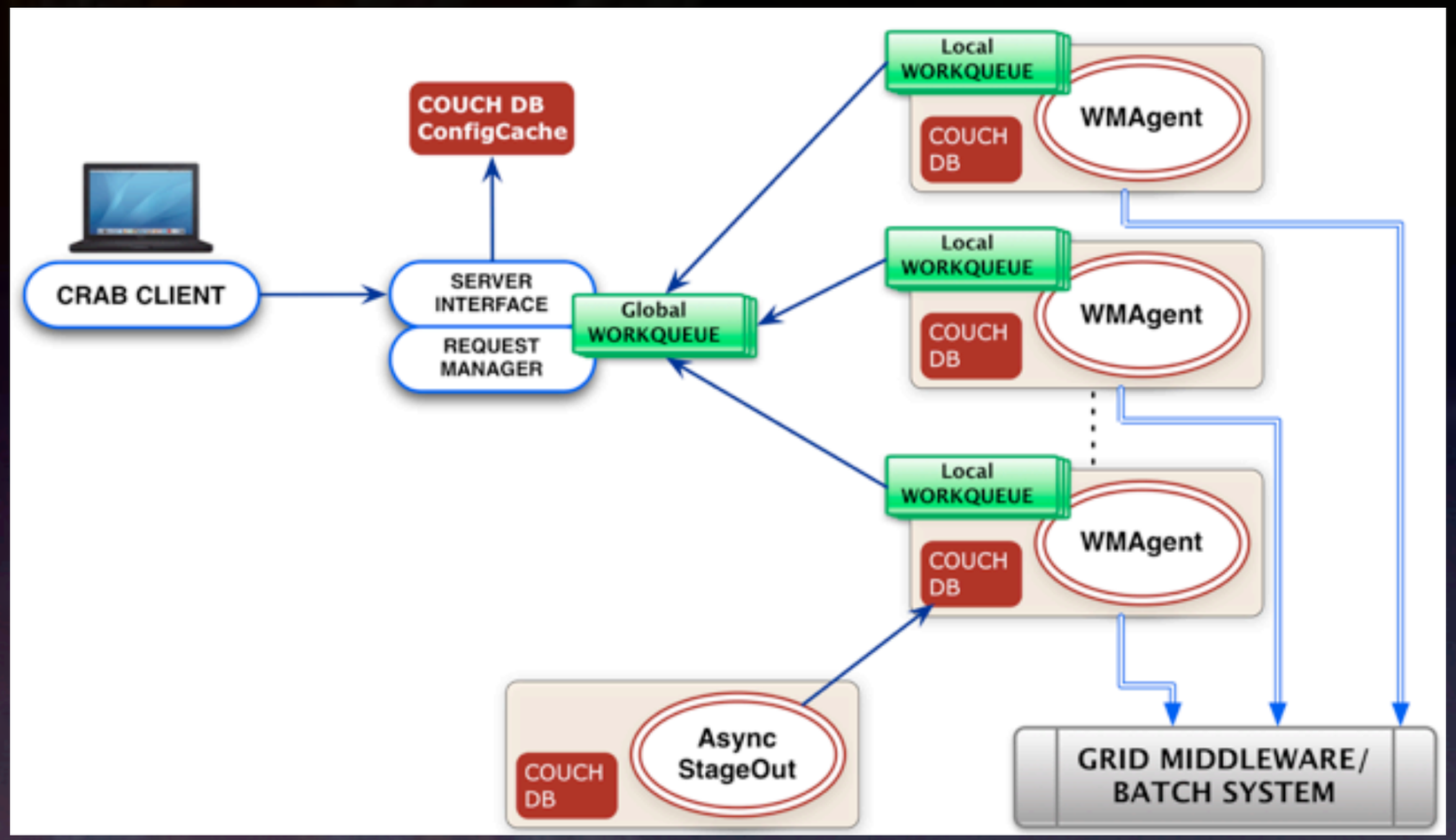
CRAB 2



Despite of the success of CRAB 2 evolution is needed:

- Client is still „thick“ - Move more logic and operation to the server part
- Synchronous stage-out is inefficient and wasting resources
- Missing features like support for growing datasets, improved job splitting (lumi section vs. events), support for multiple output files (including registration to DBS)

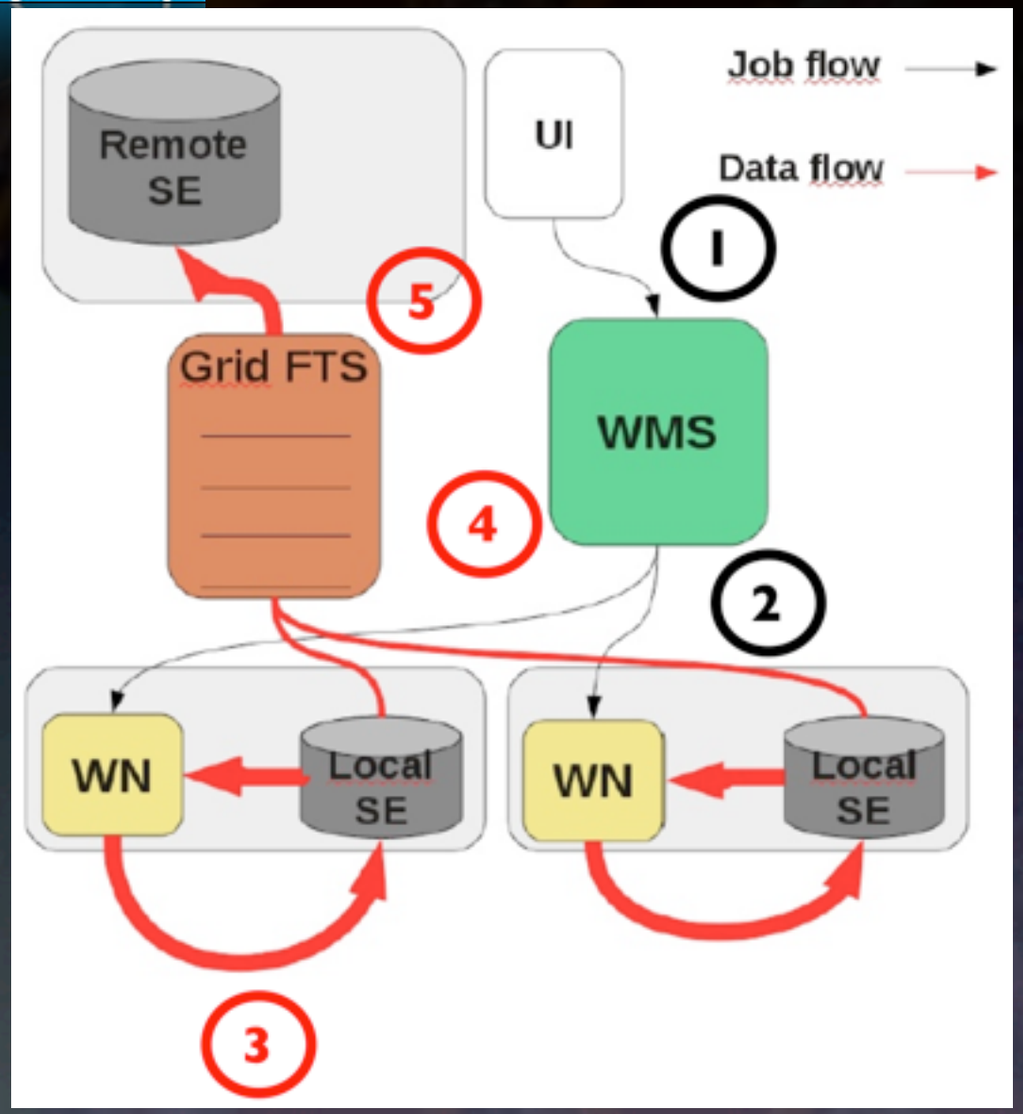
CRAB 3



M. Cinquilli

- Lightweight stateless client talking to a RESTful web-service
- Utilize WMAgent framework for overlapping parts
- No direct submission supported anymore
- Asynchronous stage-out improved situation a lot

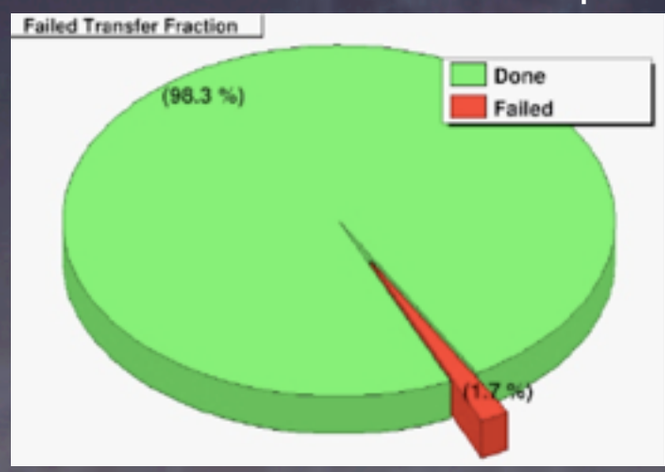
CRAB 3



M. Cinquilli

1. User submits jobs
2. Based on data location, jobs are scheduled on matched TIER 2s
3. Output is staged-out on the TIER 2 site, where job is running
4. Request file transfer using Grid FTS to user home TIER 2
5. Request is tracked and resubmitted if required

M. Cinquilli



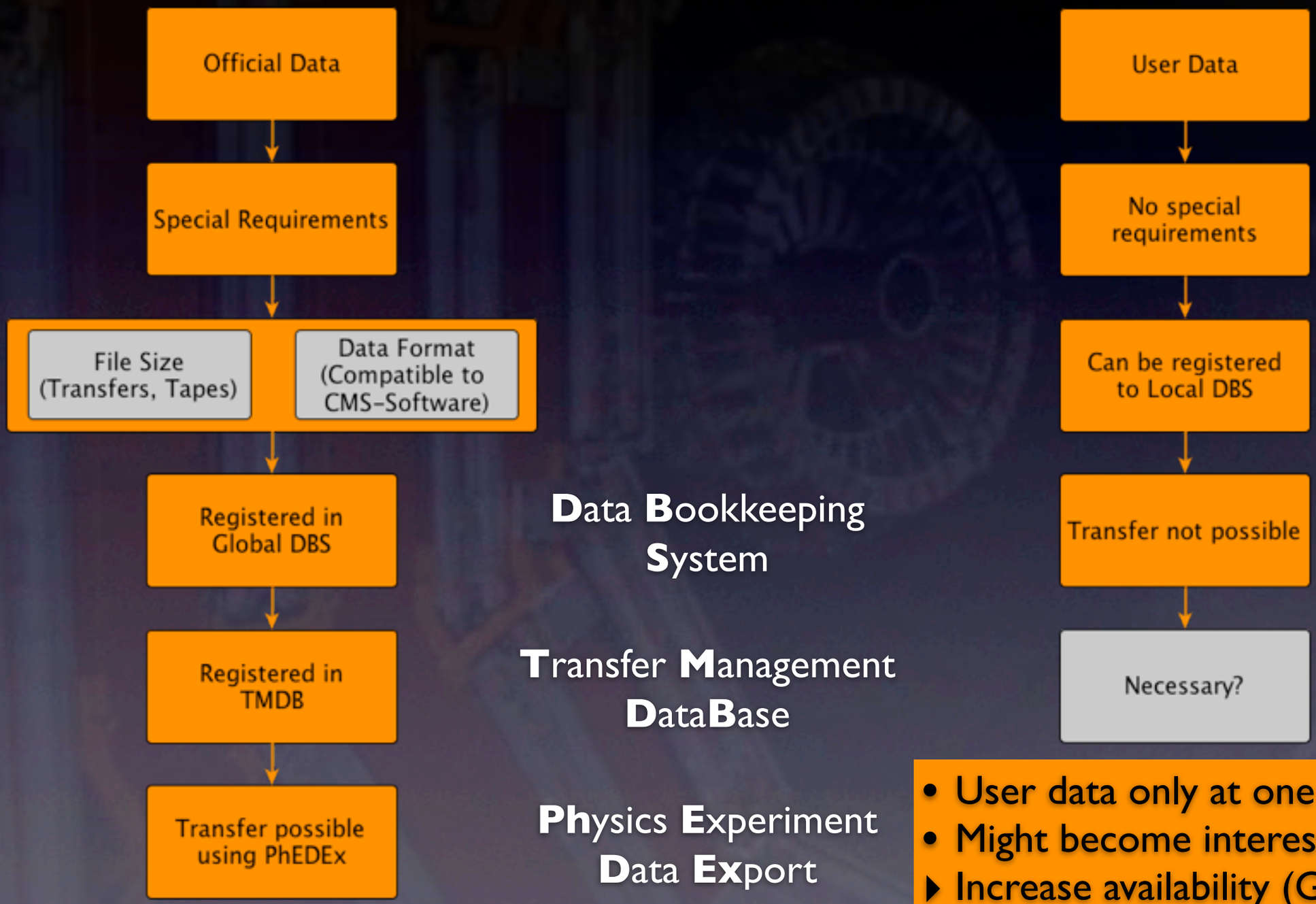
Advantages:

- Using already commissioned PhEDEx FTS channels
- Transfer takes places well-arranged (No distributed denial of service by WNs)

Want to share data within my physics group?

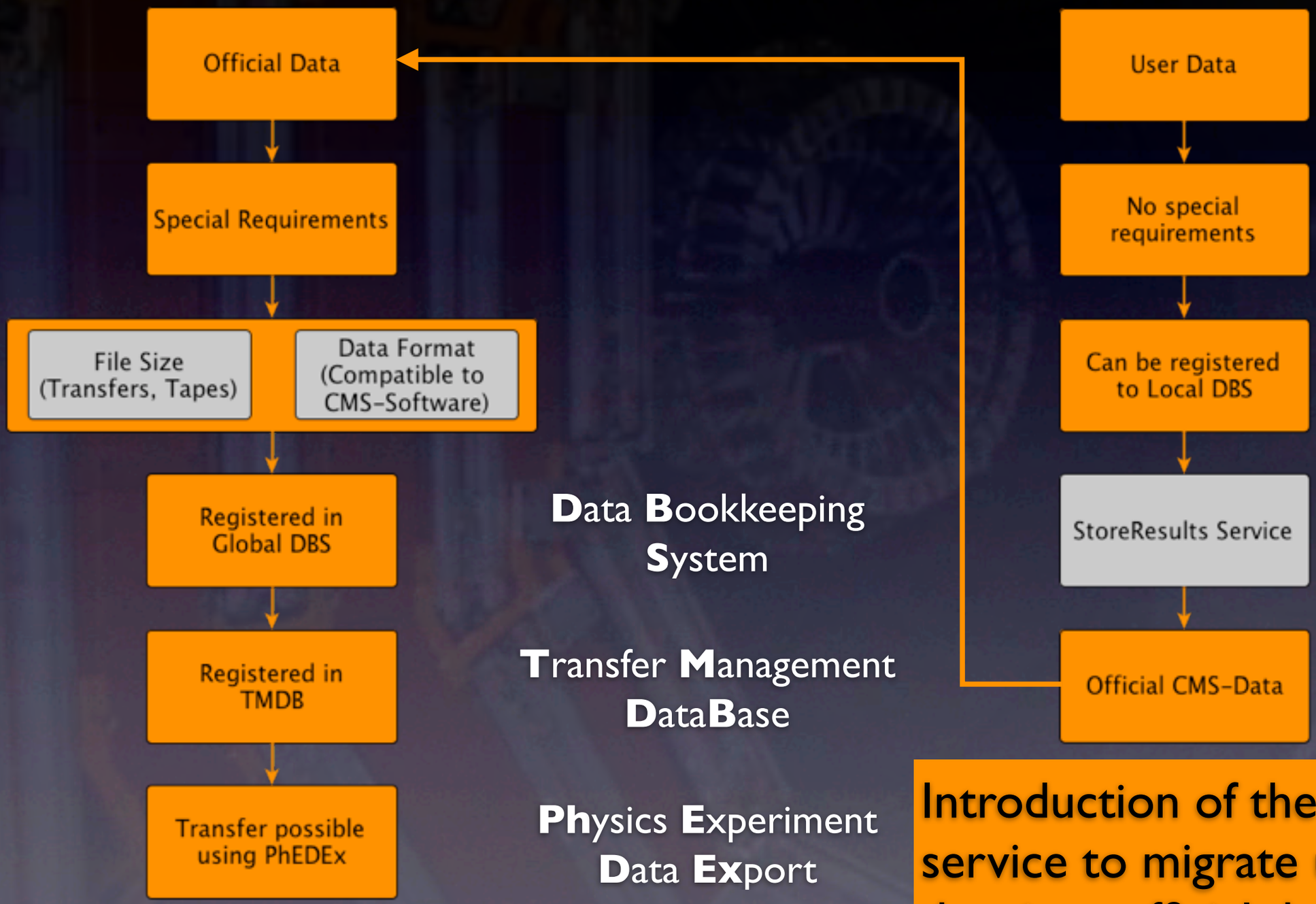
StoreResults

Problem: Transfer of user-created data is missing in computing model



- User data only at one T2 available
- Might become interesting for a group
  - Increase availability (Group T2s)
  - Custodial storage at T1

StoreResults



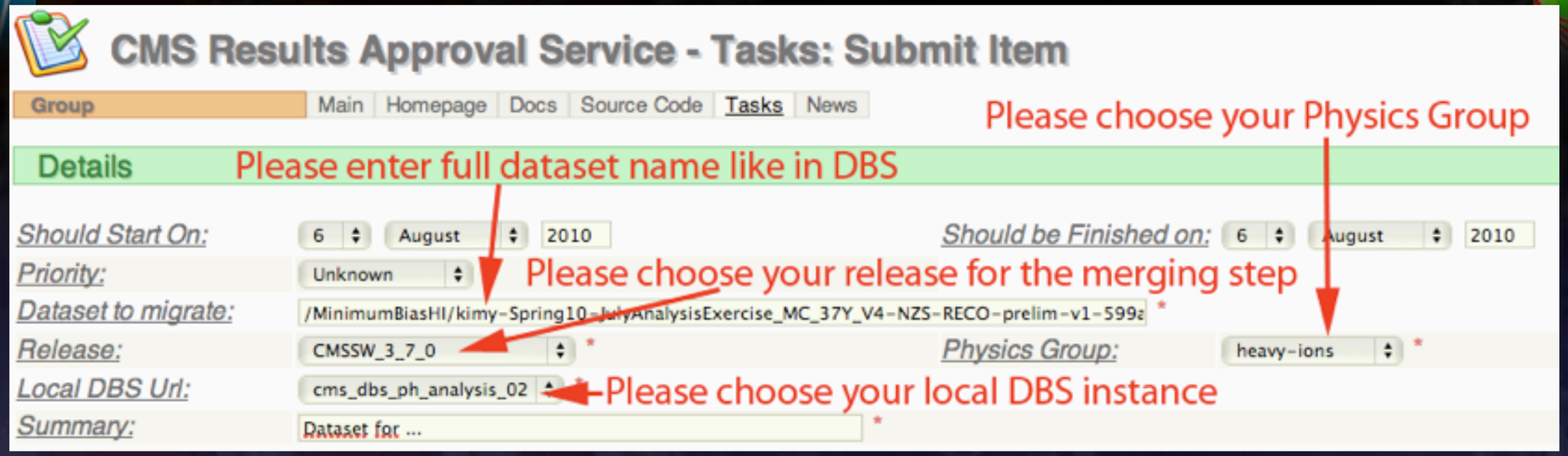
Data **B**ookkeeping **S**ystem

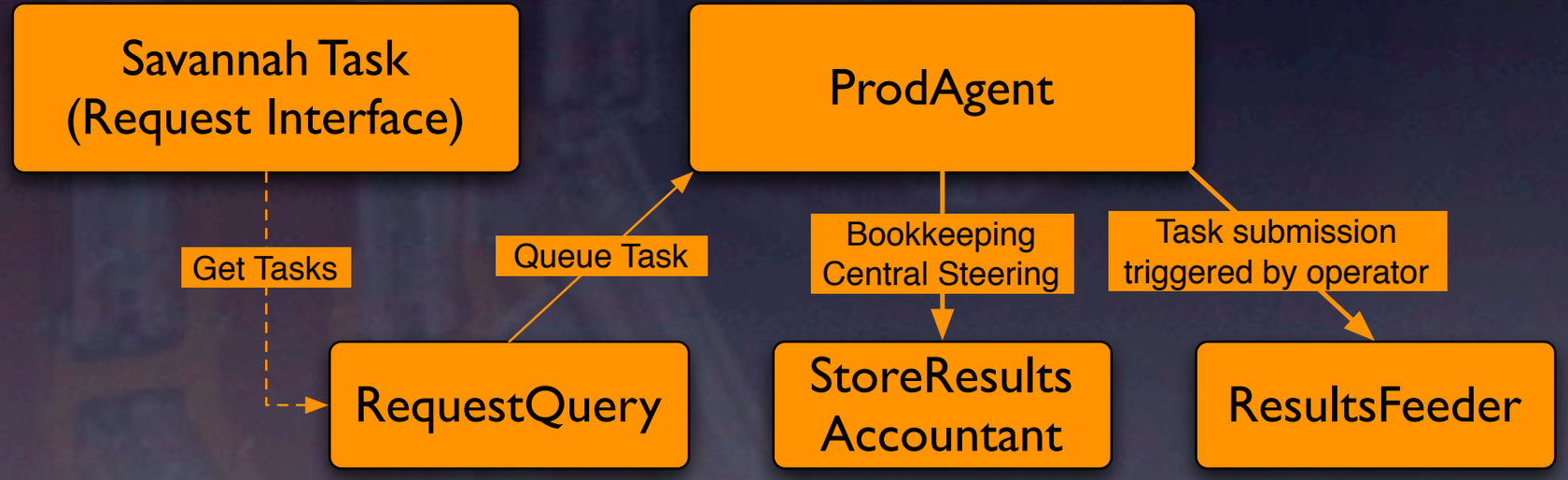
Transfer **M**anagement **DataB**ase

Physics **E**xperiment **D**ata **Ex**port

Introduction of the StoreResults service to migrate user-created data into official data

StoreResults





Ad-hoc based system around a Savannah request tracker for approvals and the CMS ProdAgent production framework for distributed processing

StoreResults

Procedure on user side

- Private dataset produced by a user
- Interesting for the whole group
- Request user space to group space migration
- Approval by group convener
- StoreResults service does the migration
- Cleanup user space

StoreResults Procedure

- Submit merge jobs
- Stage out to group space at the local site
- Ownership is now CMS-Production (Official Dataset)
- Dataset is injected in Global DBS
- Dataset is injected in PhEDEx
- Transfer dataset on request by the group

What else?

Not covered in this talk:

- HTTP Group (CMSWEB)
- SiteDB

Not part of DMWM group, but very interesting topics

- Data Popularity
- Dynamic Dataset Placement

Summary

- CMS has developed a couple of tools to master the challenge of distributed computing
- Necessary because provided middleware functionality was not sufficient
- The provided tools evolved very much influenced by the experience made during/before the first years of data taking
- Tools can profit from synergistic effects by using common base in DMWM
- CMS has developed a very efficient system, data is quickly processed and distributed over the world
- Nevertheless, further improvements still needed with respect to the growing needs in the future (LHC Upgrades)