SAM Database and relation with GridView

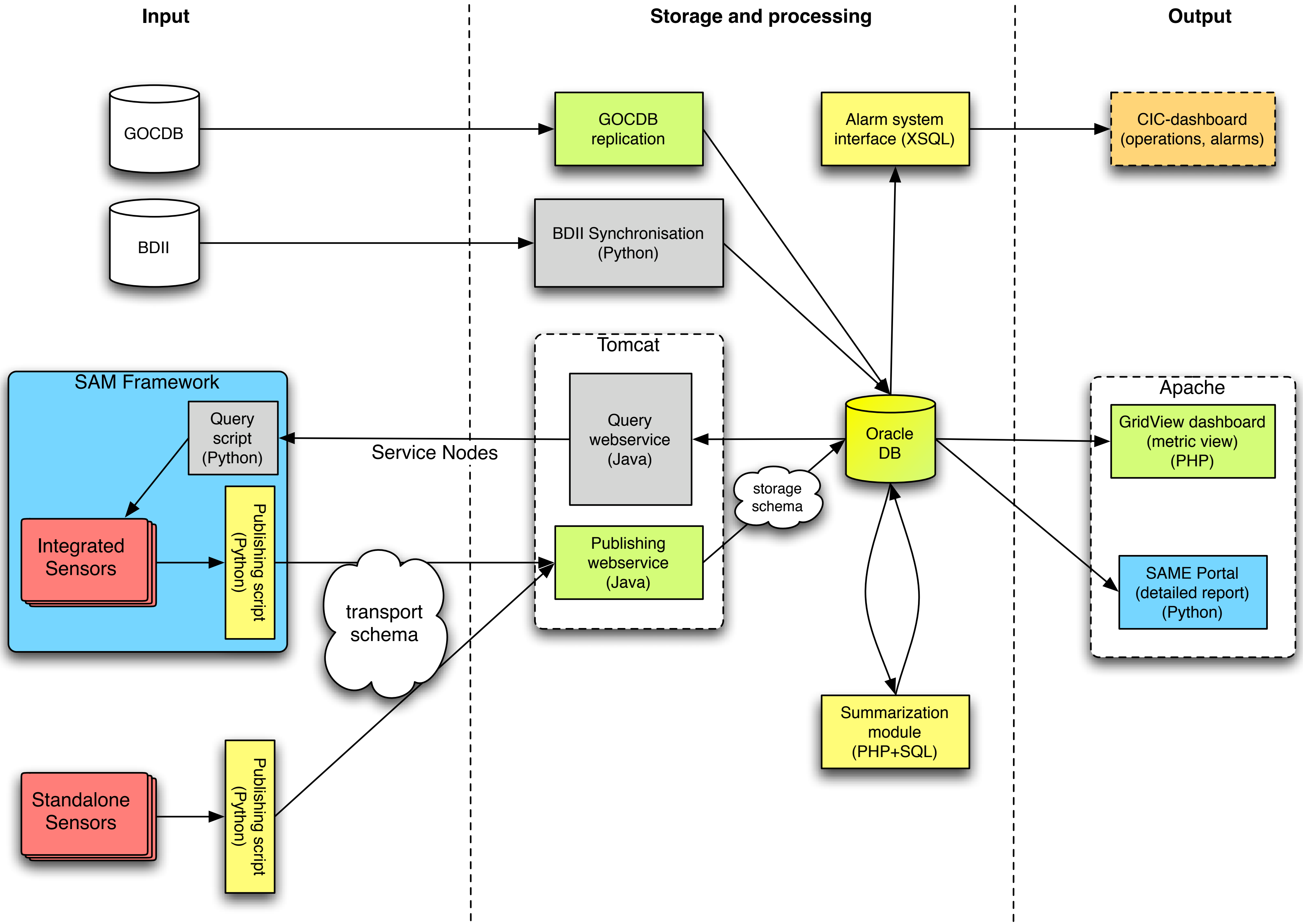
Piotr Nyczyk

SAM Review

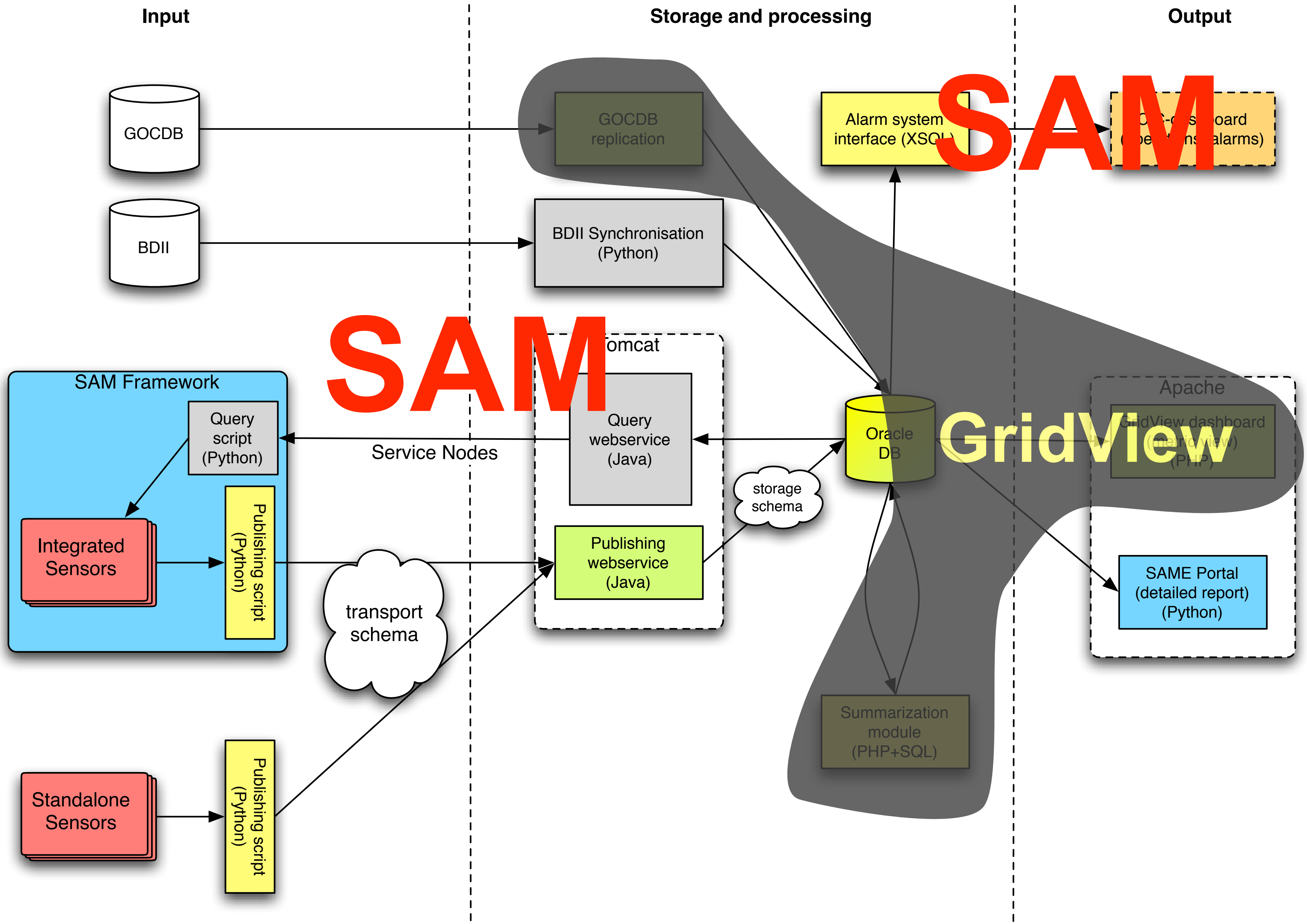
CERN, 2007

Architecture

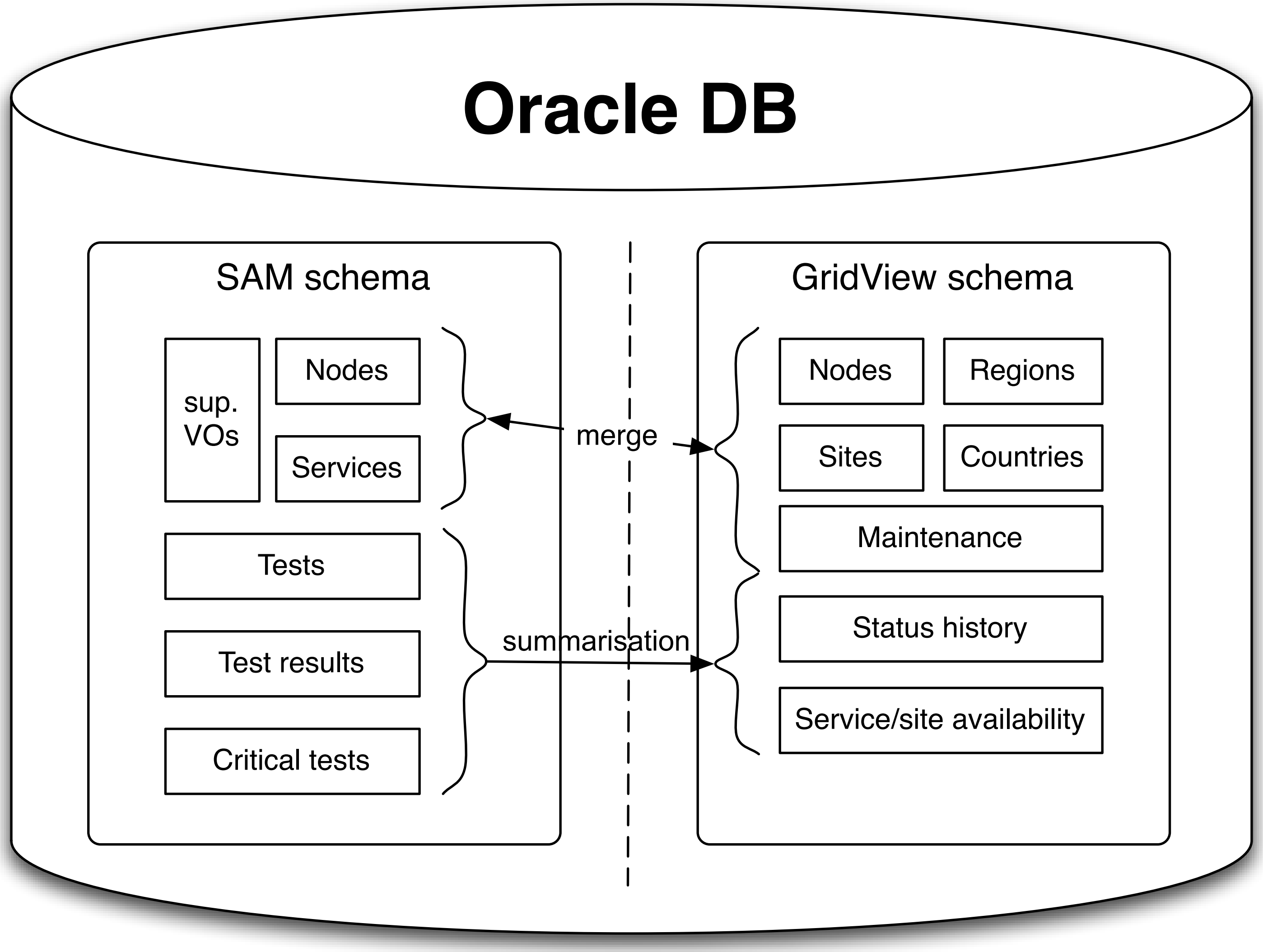
Enabling Grids for E-sciencE


Input
Storage and processing
Output
GOCDB
BDII
GOCDB replication
BDII Synchronisation (Python)
Alarm system interface (XSQL)
CIC-dashboard (operations, alarms)
Tomcat
Query webservice (Java)
Publishing webservice (Java)
storage schema
Oracle DB
Apache
GridView dashboard (metric view) (PHP)
SAME Portal (detailed report) (Python)
SAM Framework
Query script (Python)
Integrated Sensors
Publishing script (Python)
Service Nodes
transport schema
Summarization module (PHP+SQL)
Standalone Sensors
Publishing script (Python)

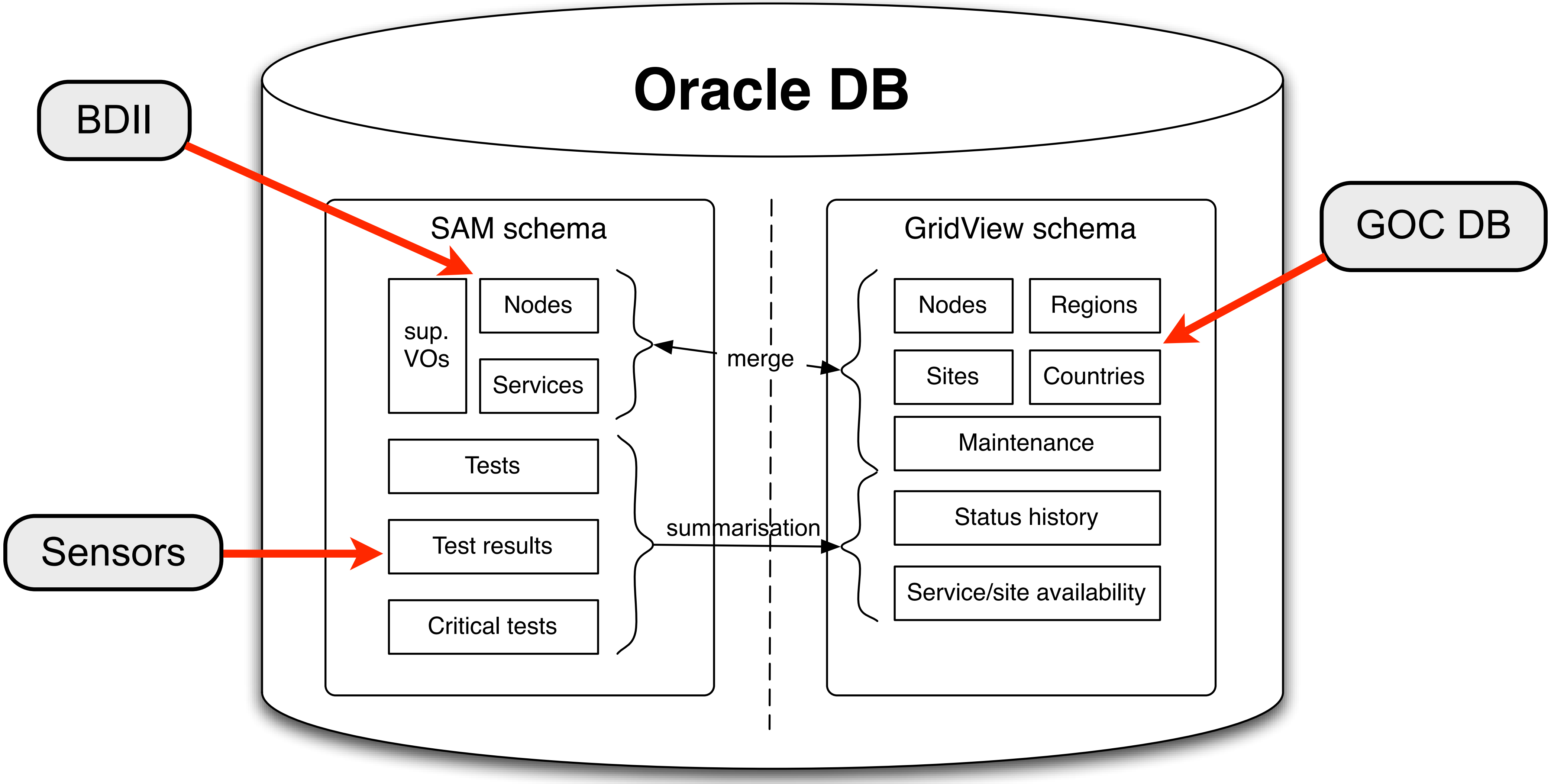
Architecture


Input
Storage and processing
Output
GOCDB
BDII
GOCDB replication
BDII Synchronisation (Python)
Alarm system interface (XSQL)
CIC-dashboard (operations alarms)
SAM
SAM
GridView
Tomcat
Query webservice (Java)
Publishing webservice (Java)
storage schema
Oracle DB
Apache
GridView dashboard (metric view) (PHP)
SAME Portal (detailed report) (Python)
SAM Framework
Query script (Python)
Integrated Sensors
Publishing script (Python)
Service Nodes
transport schema
Summarization module (PHP+SQL)
Standalone Sensors
Publishing script (Python)

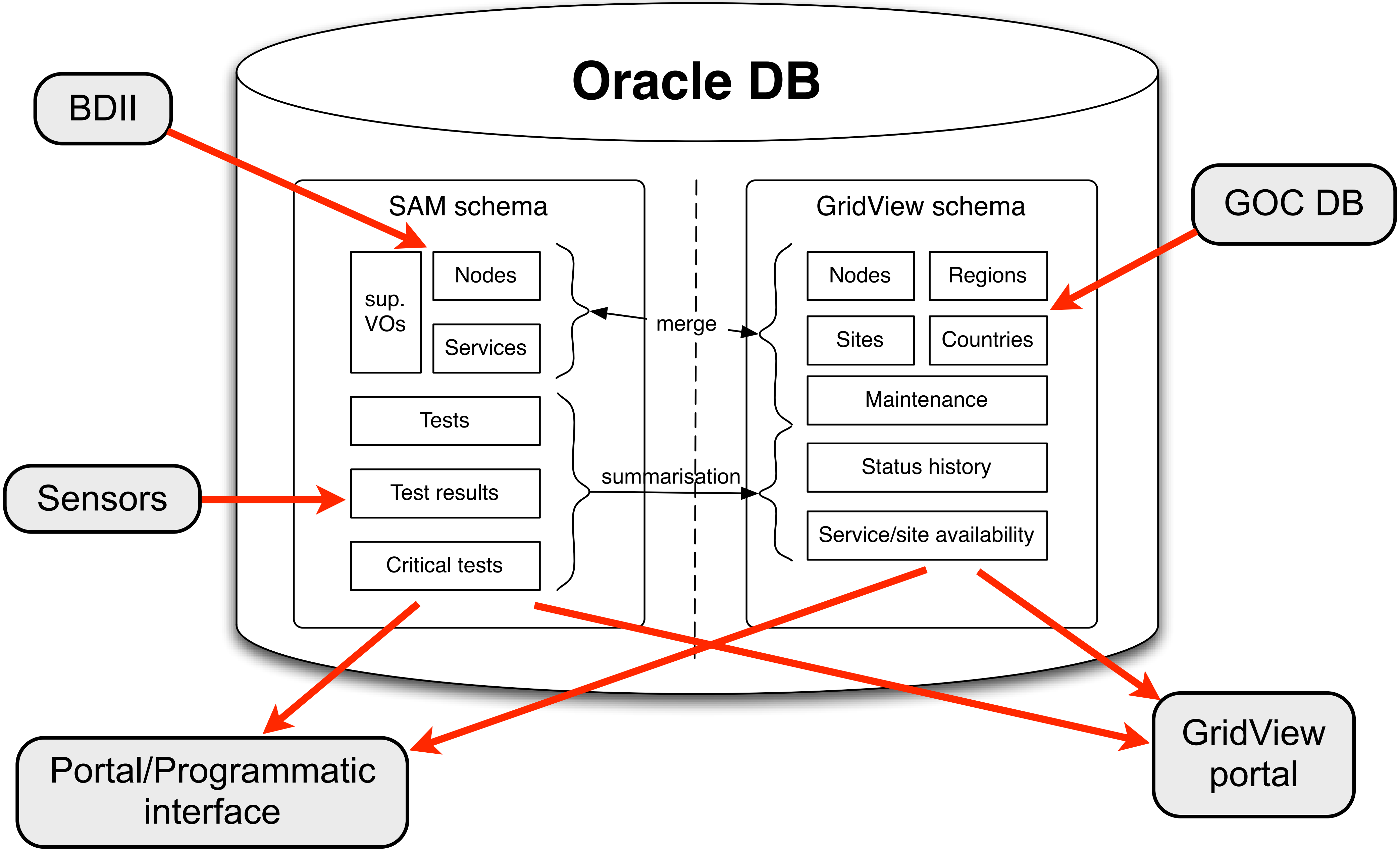
Database layout



Database layout



Database layout



DB Business logic (SAM schema)

- Processing and merging of GOC DB data
- Service discovery (BDII2Oracle)
- Alarm triggering and masking
- On-line status calculation (views)
- Availability metrics calculation

DB Business logic (SAM schema)

- Processing and merging of GOC DB data
- Service discovery (BDII2Oracle)
- Alarm triggering and masking
- On-line status calculation (views)
- ~~Availability metrics calculation~~

Moved to GridView

GOCDB data processing

- PL/SQL code running inside Oracle (Oracle job, 1/h)
- Analyses GOC DB data structures (as replicated by GridView)
- Builds new normalised representation:
  - Nodes, ServiceInstance, ServiceVO, etc.
- Uses a bit heuristic approach
  - trying to map GOCDB nodetype to Service
  - ambiguity in Node definition (multiple entries)
- Administrative topology (Sites, Countries, Regions) taken as provided (views)

Service discovery

- BDII2Oracle - external program run once/hour
- Discovers Sites, Nodes, ServiceInstances, supported VOs
- Additional services read from a static file updated through HTTP (HEP VOs requirement)
- Updates the DB merging new information with reprocessed GOCDB content (set union)
- In case of conflicts GOCDB taken with higher priority
- Ageing of nodes and service instances based on “last seen” timestamp (also from GOCDB)

Service discovery (cont.)

- When service becomes monitored?
  - site appears in GOCDB **or** in BDII (or in both)
  - node appears in GOCDB **or** in BDII (or in both)
  - monitoring flag for the site in GOCDB is ON **or** site not registered in GOCDB at all
  - monitoring flag for the node in GOCDB is ON **or** node not registered in GOCDB at all
  - the same logic for scheduled downtime
- How to disable monitoring of a service?
  - both site and node have to be registered in GOCDB
  - monitoring flag switched off either on the site or on the node level
  - the same logic for scheduled downtime (But! scheduled downtime doesn’t trigger monitoring off)

Alarm triggering

- Procedure to trigger an alarm:
  - – The test result is ERROR or CRIT,
  - – The node belongs to a certified site,
  - – VO is 'OPS‘,
  - – The test is critical for OPS VO,
  - – No alarm already for that test, vo and node,
  - – The node is not in maintenance.

Alarms Info

- Data stored of each alarm:
  - alarmid
  - vo
  - test
  - node
  - test exec time
  - alarm status (new, assigned, masked, off)
  - update time
  - ticket id (GGUS)

Alarms Masking

- Automatic Alarms Masking:
  - – Simple rule based correlation engine
  - – If there is one or more alarms with status='new' for this VO, node and test => new alarm triggered as masked.
  - – Rules defining test relationships among alarms:
    - http://lcg-sam.cern.ch:8080/alarms/mask_alarm.xsql

Prioritisation of alarms

- Depending on the Service:
  - – 40.000 points: VOBOX, BDII, VOMS, LFC, WMS, RB.
  - – 30.000 points: SRM, MyProxy, FTS.
  - – 20.000 points: RGMA, sBDII.
  - – 10.000 points: gCE, CE, SE.
- Depending on n° of alarms getting masked:
  - – 1.000 points per alarm masked by the new alarm,
  - – But up to a maximum of 9.000 points.

Prioritisation of alarms (cont.)

- Depending on the test status:
  - – 100 points if ‘INFO’
  - – 200 points if ‘NOTE’
  - – 300 points if ‘WARN’
  - – 400 points if ‘ERROR’
  - – 500 points if ‘CRIT’
- Depending on n° of CPUs in the site:
  - – Value taken from the 'CE-totalcpu' test divided by 100.
  - – This gives a [0-50] number.

Open issues

- Irresolvable conflicts in GOCDB (2.0)
  - migrate to GOCDB3 - short term (1 month)
  - redesign SAM/GV data schema - long term (6 months)
- BDII2Oracle needs refactoring (modules) and improvements in data processing:
  - basic modularisation (Input/Output) - short term (3 months)
  - new processing model (synchronised with redesign of data schema) - long term (>6 months)
- Differentiation of test criticality level needed
  - Alarm level - for example WARN for host certificate
  - Unavailability level
  - timeline: ~3 months