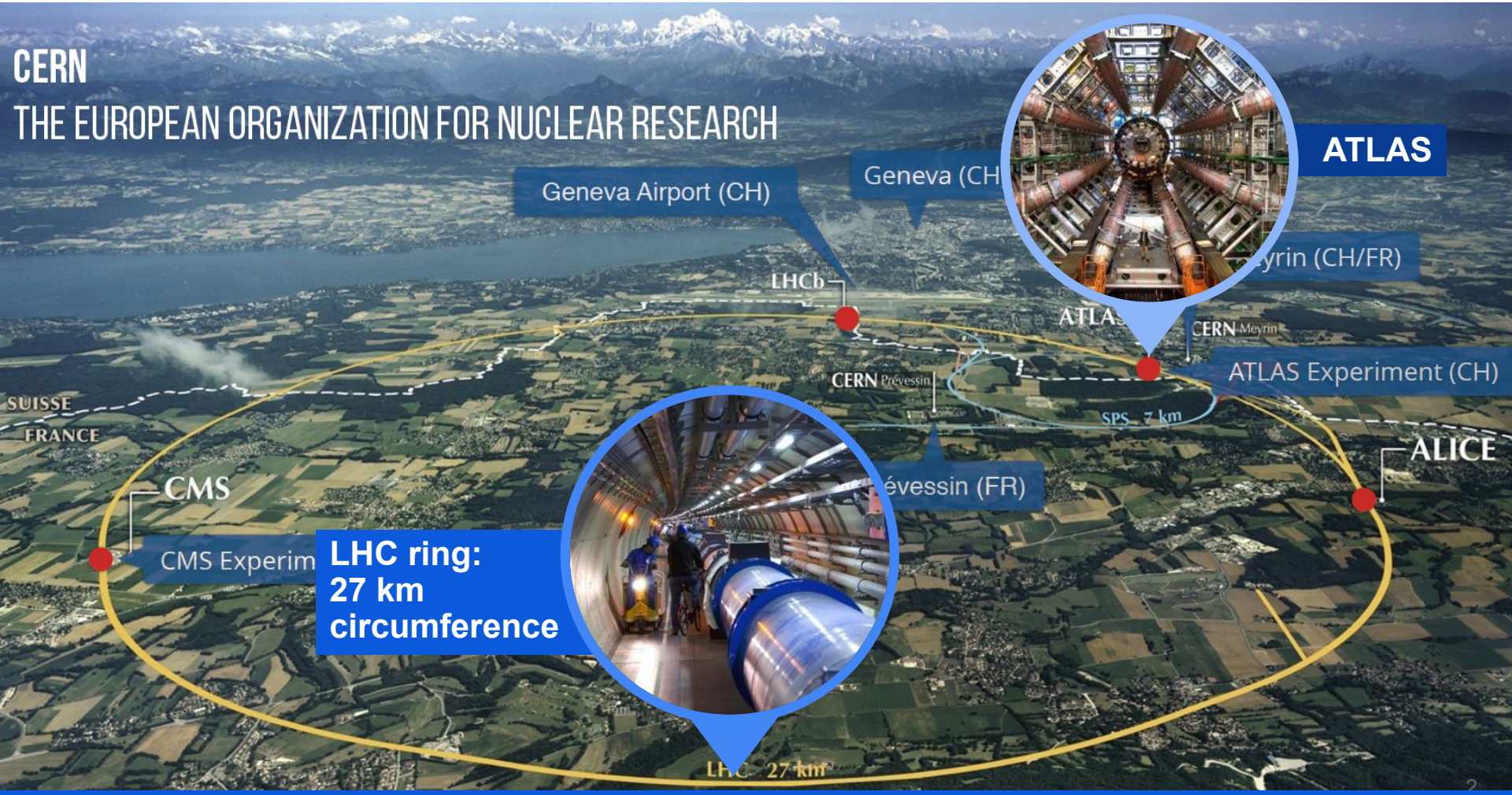
CRIC:

A unified Information framework for LHC distributed computing and beyond

Alexey Anisenkov (BINP)

SPD S&C Weekly, 5 Dec 2023

The Large Hadron Collider (LHC) at CERN



Exploration of new physics and energy frontier in pp and pb-pb collisions

LHC Challenges

- 100+ PB/year capacity production
  - Current total:
    - All LHC: ~ 1.5 EB
    - ATLAS: 0.5 EB (raw+sim+derived+replicas)
- Data analysis requires at least ~500k cores (typical PC processor cores)
- Scientists in tens of countries worldwide
- CERN can provide only up to 20-30% of the storage and CPU

Requires powerful large-scale computing & storage system;
distributed-grid concept

Computing Evolution in a nutshell

Serial Computing: single CPU/memory space

Parallel Computing

Parallel/Concurrent use of multiple nodes or CPUs working together on a *common* task (shared memory)

- MCORE computer
- "Distributed Computer" (distributed memory multiprocessor)

Distributed Computing

A model in which components of a software system are shared among multiple computers to improve efficiency and overall performance
("distribute" memory)
(communication via messages)

- Client/Server model
- p2p model, cloud model

Grid Computing

Highly distributed ("globally distributed") architecture of large numbers of computers/centers connected to solve a complex problem.

Consider redundancy, and robust failure recovery as built-in concept.

HPC (Supercomputer)

Inter-connected massively parallel high performance computing facility required a large number of processors, shared or distributed memory, and multiple disks. Housed in specialized data centers

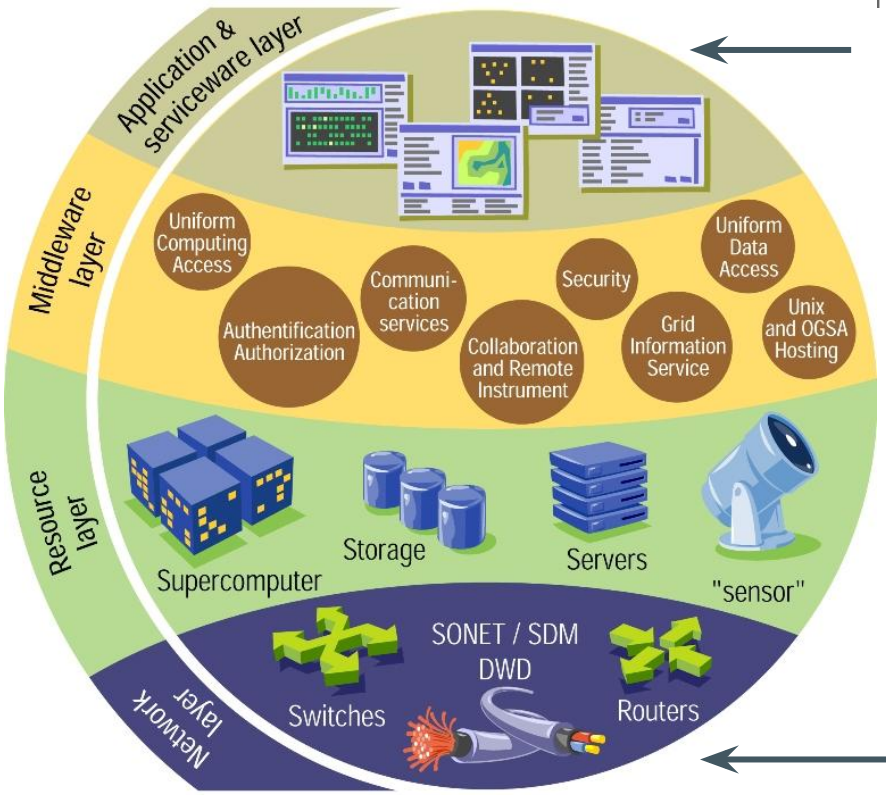
Cluster

A group of loosely/tightly coupled computers that work together closely (connected via LAN) at single location. Centrally managed and usually homogeneous

Cloud Computing

High level of virtualization, on-demand computing, software/hardware/platform..--as-a-service..

Traditional Global Grid Infrastructure layers



High-level VO-oriented middleware services & applications (e.g. for ATLAS: PanDA, Rucio, AGIS/CRIC, MONIT..)

The middleware exposes heterogeneous resources to VOs in a uniform interface through the Grid:

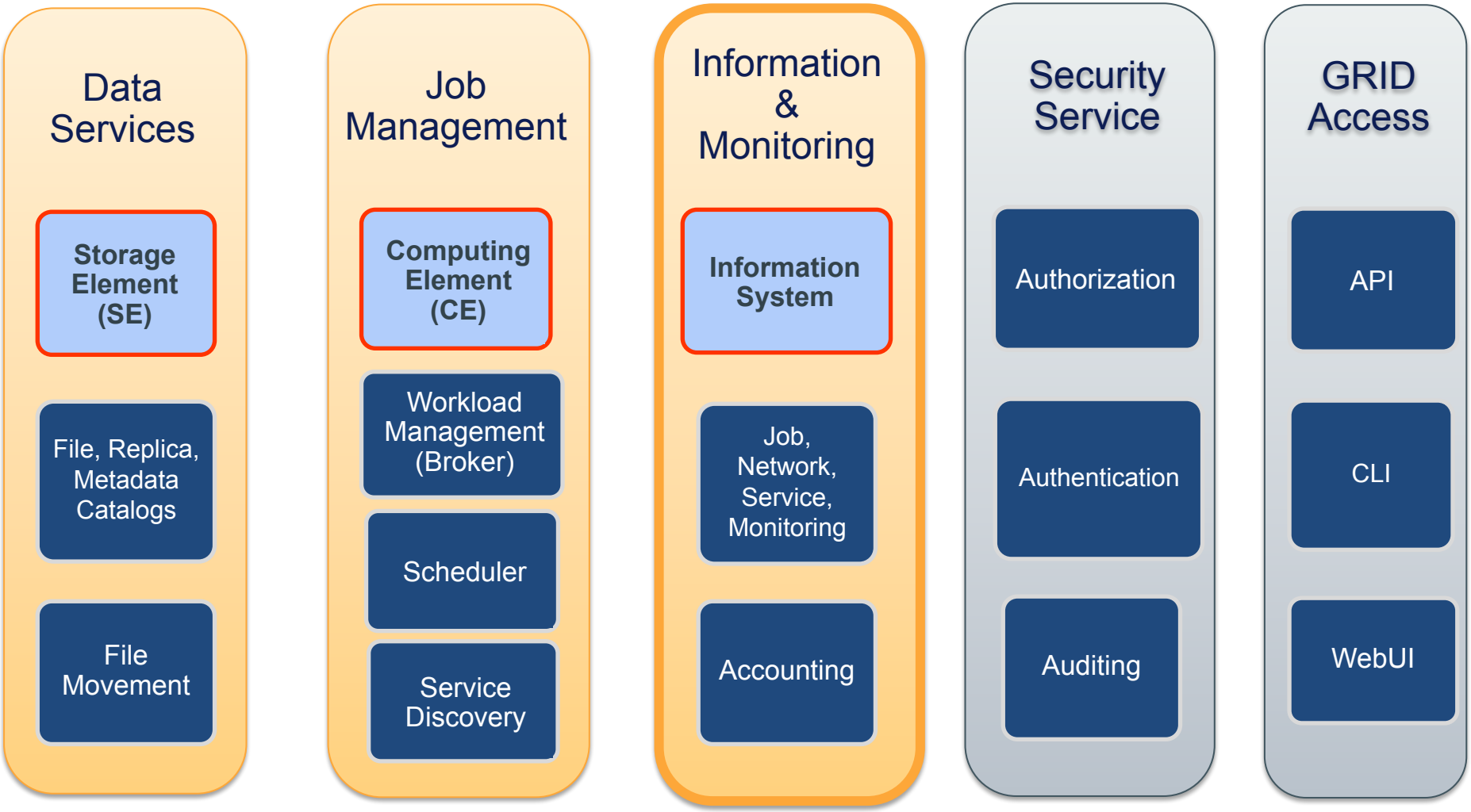
- Computing Elements give access to CPUs
- Storage Elements give access to data
- Information systems describe the resources
- Authentication & Authorization

Dedicated LHC optical Private Network

LHCOPN

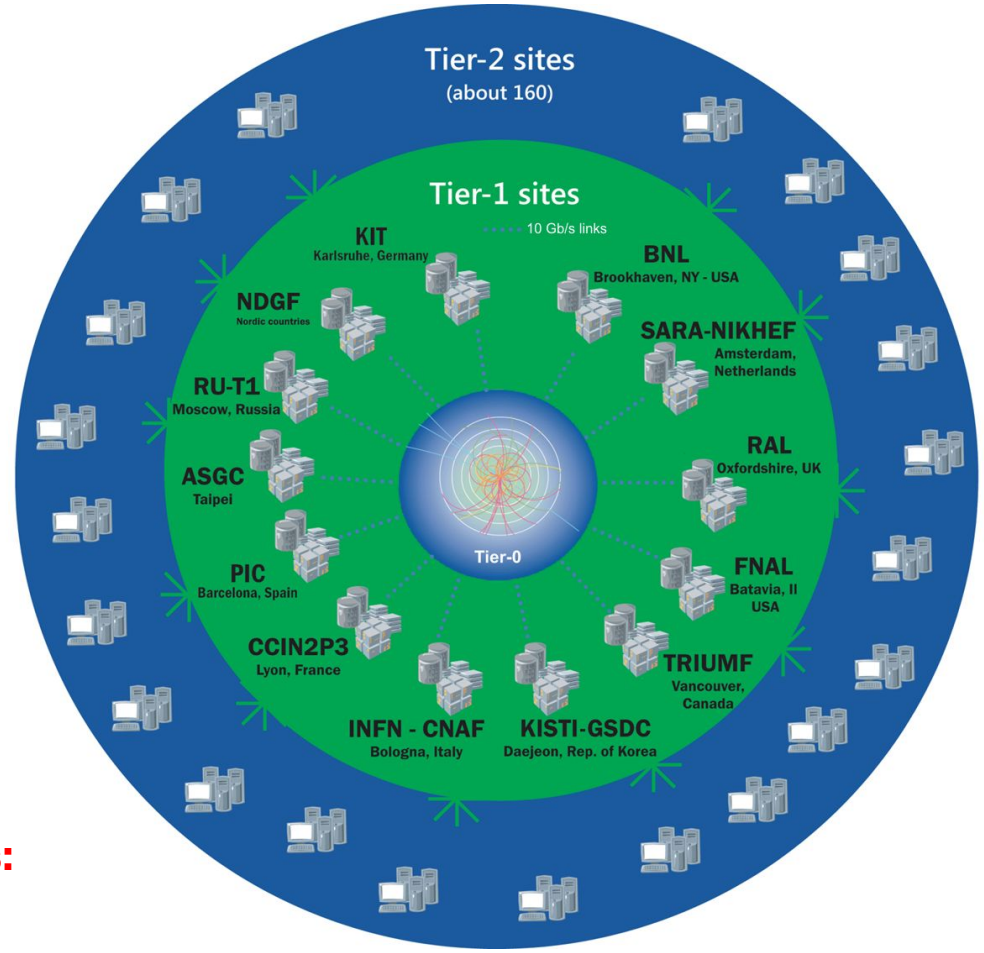
Middleware makes Illusion that distributed infrastructure is a single resource.

Basic Components of Grid Middleware



WLCG: Computing Model (continuously evolving)

- **Tier-0 (CERN):**
  data recording and archival, prompt reconstruction, calibration and and distribution
- **Tier-1s:**
  permanent storage, second tape copy of data, re-processing, memory & CPU intensive tasks, analysis
- **Tier-2s + Tier-3s:**
  Simulation, end-user analysis



nearly 170 sites, 42 countries

> 2 million jobs/day
10-100 Gb links (2016)

Pledges resources (2016):

350k cores (3.8M HEPSpec06)

700 PB of storage (310PB disk + 390PB Tape)

Integrates computer centres worldwide that provide computing and storage resource into a single infrastructure accessible by all LHC physicists

* Initial MONARC architecture (1999) - Models of Networked Analysis at Regional Centers for LHC Experiments

Distributed Computing Environment (Resources)

~~LHC Experiments~~ (any modern HEP experiments) rely on **heterogeneous** distributed computing

- variety of computing resources involved

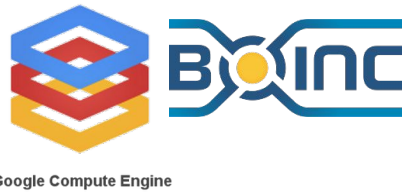
WLCG Pledged resources

Rented, on demand

Opportunistic

Opportunistic backfilling

- variety of infrastructures and middleware providers



others ..

Distributed Computing Environment (Experiments)

- Each **Community** uses and describes **Resources** in its own way

...

  - Computing Models are similar but still have different implementation
  - Various high level VO-specific frameworks & middleware services (e.g. for Data and Workflow management)
  - Cross experiments applications (monitoring, accounting, testing frameworks, resource usage descriptors, etc)

- Apart from **resources** description, high level VO-oriented middleware services and applications also **require** the diversity of common configurations to be centrally stored and shared

Experiments SW tools,applications, services: a big world

Some examples from ATLAS

ADC services require the diversity of common configurations as well.

ATLAS Data Management System (Rucio)

ATLAS Workload Management System

ATLAS Monitoring tools

DDM Accounting

WLCG Squid monitoring

Pilots, AutoPilot Factories

BigPanda Monitors

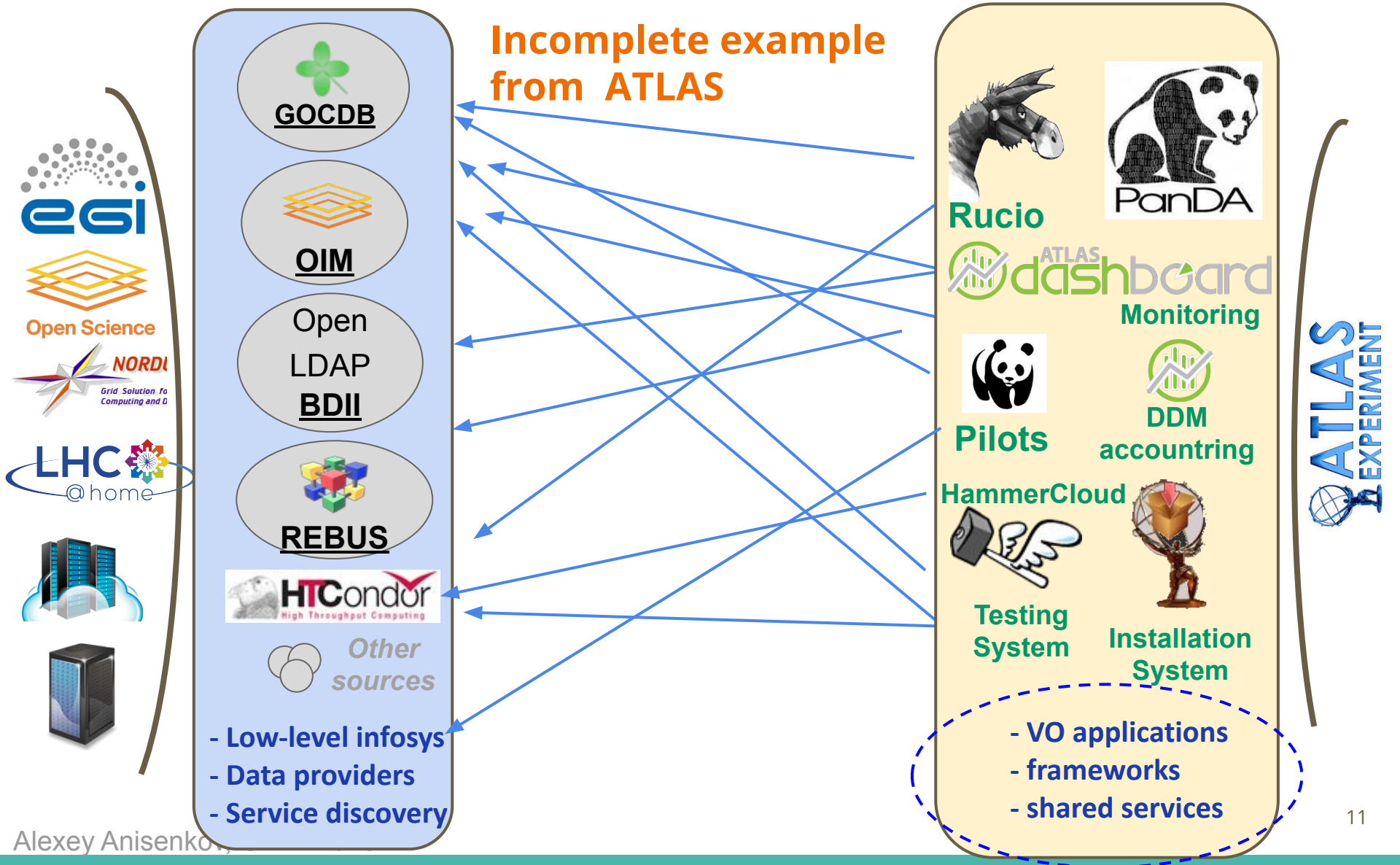
Software Installation systems

Testing systems (HammerCloud)

Resource Configurations & VO applications

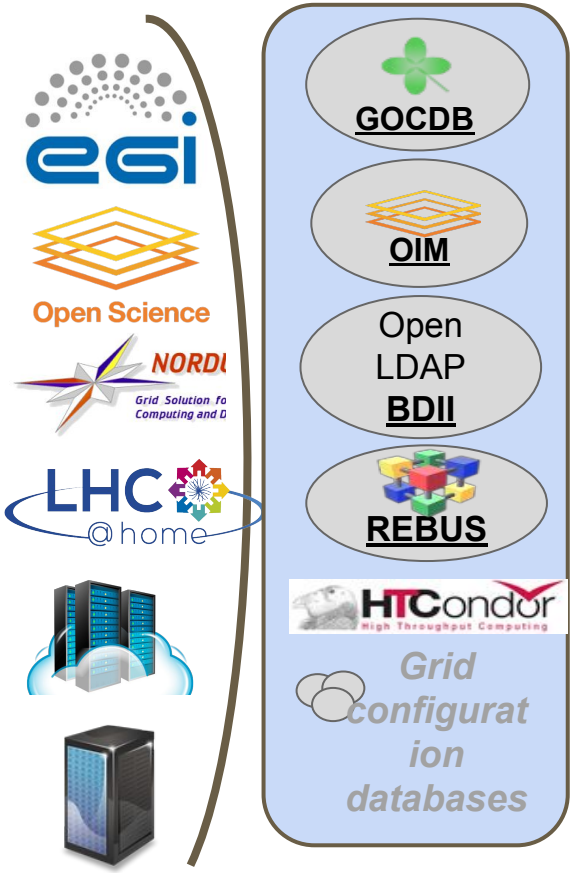
Resources description

Experiment apps (Services)



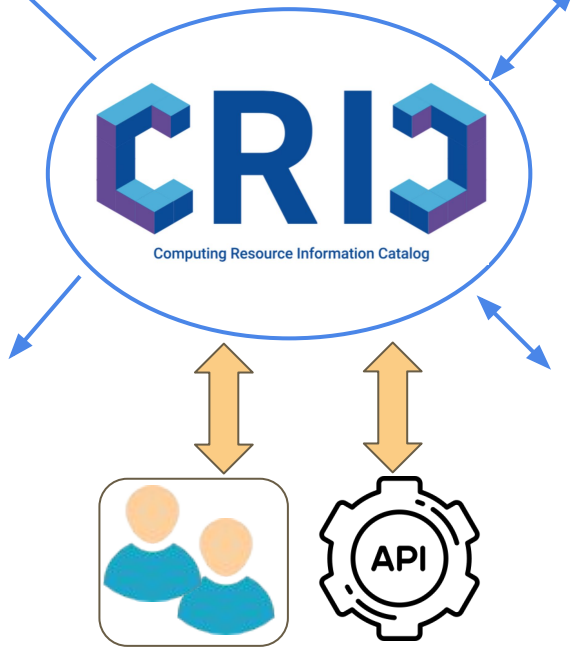
CRIC: a unified topology system for a large scale, heterogeneous and dynamic computing infrastructure

HW/SW Resources Configuration layer



- Low-level configuration systems
- Data providers
- Service discovery components

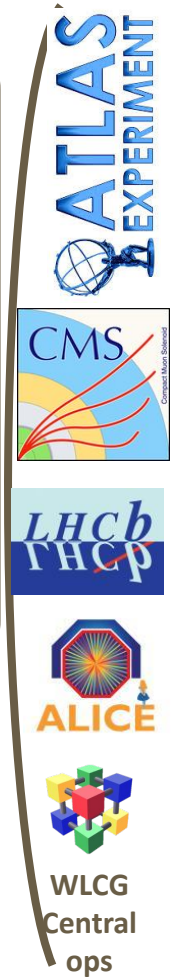
High-level Information middleware level



- Central WebUI portal
- REST API services for data export and modification
- Automatic data collectors
- Data validation

Experiment Applications Frameworks, services layer

- Data Management System
- Workload management systems
- Monitoring tools
- SW installation services
- Testing frameworks



Alexey Anisenkov

CRIC mission: link Resources & VOs together

- ➢ Consolidate topology information of a large scale distributed computing environment
- ➢ Facilitate distributed computing operations for (LHC)** Experiments

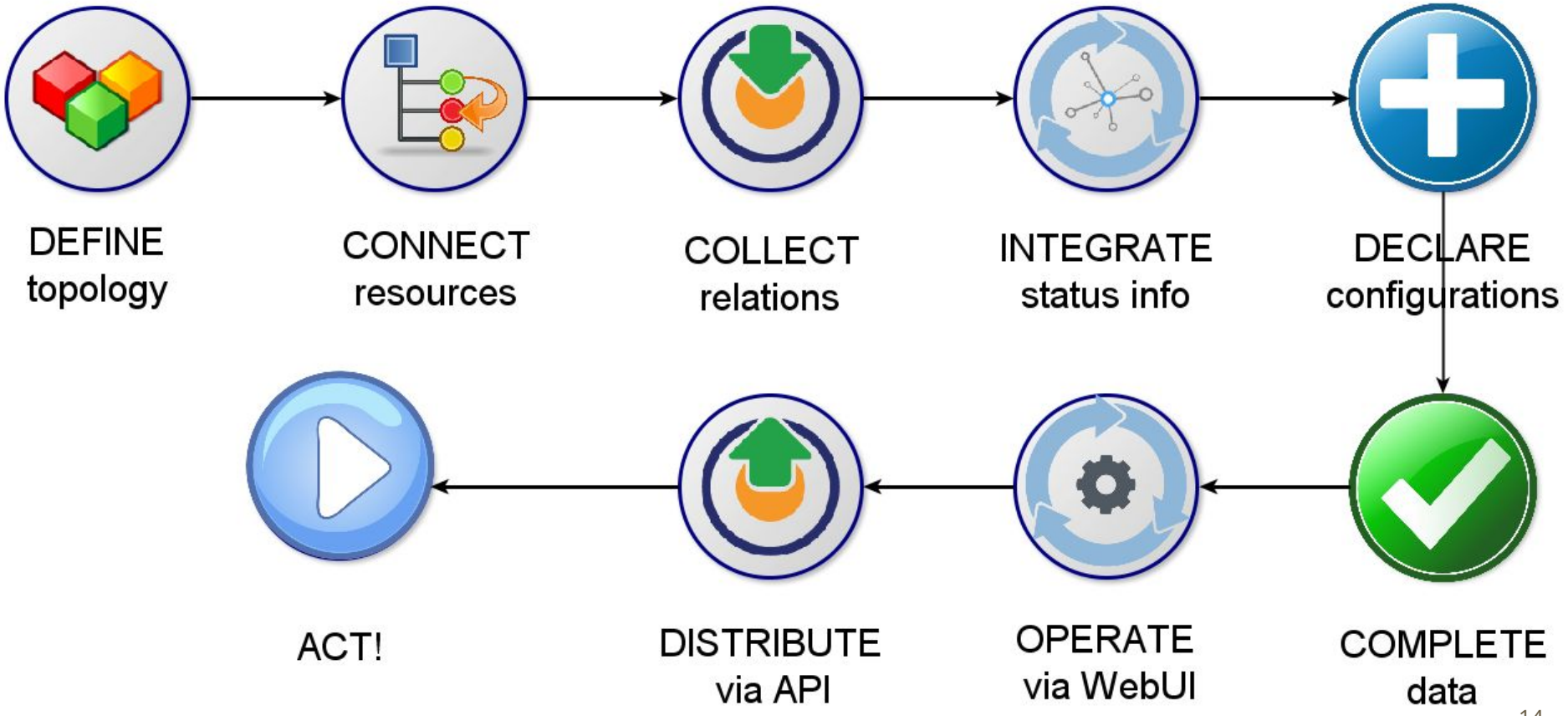
Key functional capabilities of the CRIC information concept:

- Built-in information Model(s) for Resource descriptions
- Clear distinction between (physical) resources ***provided by*** (Sites) and how they are ***used by*** Experiment(s)
- Built-in aggregation and validation of data collected from various low-level information providers (sources)
- Ability to extend and complement Information Model(s) with Experiment(s) specific data structures. Flexibility to address technology evolution and changes in the VO Computing models and applications
- Experiment-oriented but still Experiment-independent information framework; Plugin based approach allows experiments to address own reqs

** Initially AGIS (CRIC) has been developed for the ATLAS experiment and then evolved to whole LHC computing environment. For today thanks to **Plugin based approach** CRIC can be successfully applied **beyond WLCG** for generic computing environment as unified information system to address custom VO requirements.

CRIC capabilities

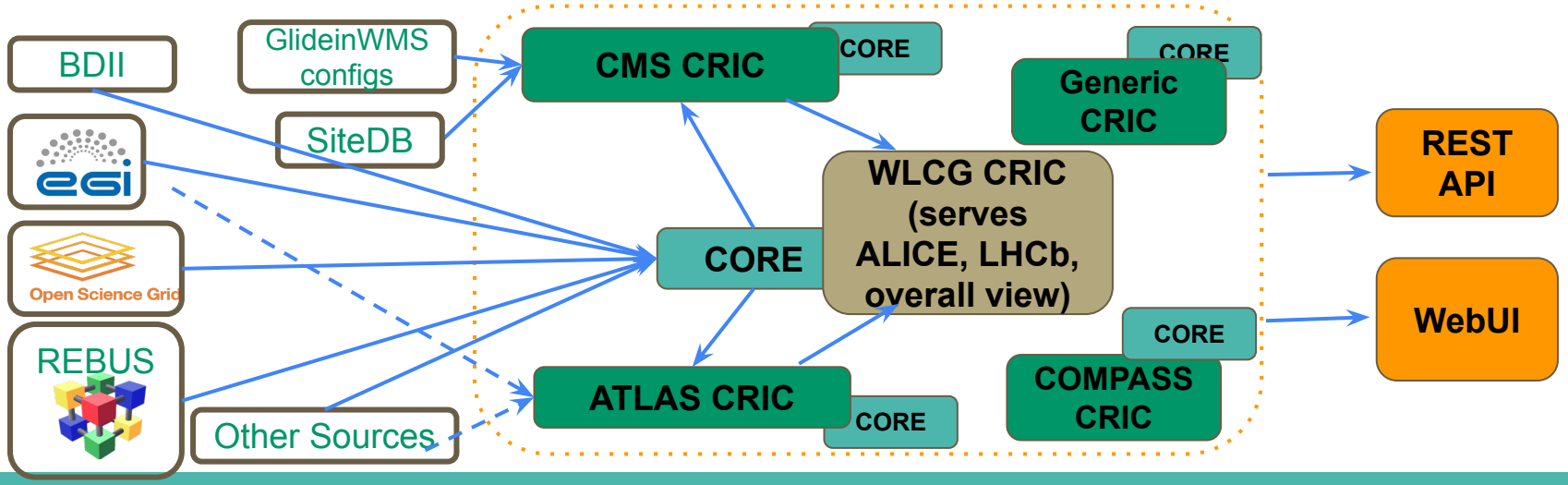
CRIC is the middleware initially designed to describe the topology of the Computing models, providing unified description of resources and services used by Experiment applications



CRIC Architecture: examples of shared features out of the box

- Plugin based: VO can configure default behaviour
- Base implementation for the Resource/Topology description
- REST API data export (filters, presets, various output formats)
- Shared engine/widgets for WebUI (downtime calendars, table view, tree view, inline editors, etc..)
- Enhanced Authorization (CERN SSO, SSL, paswd based;local accounts)
- Enhanced Authentication (instance specific permissions, groups, roles, etc, map permissions to e-groups, fetch info from ext sources)
- Detailed History of Changes

VOs can optionally use shared WLCG CRIC instance as the source or fetch data directly from low-level info providers



AGIS vs CRIC: some history

AGIS was born in 2009 in ATLAS as the ATLAS Grid Information System:

- A collaborative project involving several institutes (BINP, JINR, BNL, Mephi, CERN IT)
- Several people involved in the course of the years
- More than 2 years to go from the design phase into production phase
- In full production as one of the ATLAS critical framework since LHC Run-1 (~2011)
- mainly ATLAS oriented information system

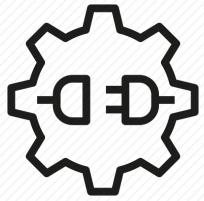
Successful experience of AGIS within ATLAS triggered WLCG management to consider AGIS as a base platform for WLCG Information system.

CRIC is the evolution of AGIS framework beyond ATLAS (**2016: CRIC era, WLCG applications and beyond**):

- Next-generation system, feractored and unified engine for WLCG applications, VO-agnostic implementation
- focused to fit the needs of major experiments at LHC and beyond

Few Implementation details: Web2.0 based

Bootstrap



Shibboleth

- Apache/WSGI + Python + Django framework as server backend
- Independent database backends (Oracle, MySQL, etc)
- Web Services technologies (REST API, WebUI, widgets)
- Bootstrap framework as HTML/CSS/JS client frontend (responsive, interactive, mobile-friendly)
- Client AJAX, JQuery plugins, own widgets (datatables, treeview, calendar view, inline editors ..)
- Plugin based approach (shareable applications in “core” re-used by many components)

CRIC features as the infosys middleware for VOs

- Helps to easily integrate new Computing technologies which have not yet appeared in WLCG as the services or can not be part of WLCG in general, ATLAS examples:
  - newer type of SE based on ObjectStore technology
  - Federated Access to storage (FAX redirectors, direct access to remote files from Worker Nodes)
  - Description of opportunistic/volunteer resources

- Helps to minimize side effects for end-user applications of various internal migrations/changes/tests/evolution of Distributed Computing components/infrastructure:
  - Consolidation of protocols description that should be applied only for few sites, unification of resources, migration to HTCondor
  - Keeps data export in several format for backward compatibility

- Masks incompatible updates in external data providers, implement missing functionality/overwrite/fulfill data:
  - e.g. fix wrongly published number of cores, core-power
  - remove direct dependency to ext sources (obsolete data providers)

CRIC family (2019)

in production		development	integration	future
CMS CRIC core-0.3.7 cms-0.2.7	**ATLAS CRIC** core-0.3.7 atlas-0.3.1	**lightweight CRIC** ongoing	**COMPASS CRIC** core-0.2.3 compass-rc	**NICA CRIC** .
WLCG CRIC core-0.3.7 wlcg-0.2.1	**DUNE CRIC** ..	**Datalake (LSST)**	**DOMA CRIC** core-0.3.7 doma-rc	.

➢ **CRIC** offers a common framework describing generic distributed computing infrastructure with also an advanced functionality enabled to define all necessary Experiment specific configurations and settings.

Thank you for your attention!

Backup slides

- Check **CRIC**:
  - http://atlas-cric.cern.ch (ATLAS-CRIC)
  - http://cms-cric.cern.ch (CMS-CRIC)
  - http://cms-cric-docs.web.cern.ch (CMS-CRIC documentation)
  - http://wlcg-cric.cern.ch (WLCG-CRIC)
  - http://escape-cric.cern.ch (ESCAPE-CRIC)
  - http://dune-cric.cern.ch (DUNE-CRIC)
  - http://datalake-cric.cern.ch (DATALAKE-CRIC)
  - http://compass-cric.cern.ch (COMAPASS testbed)

Lightweight CRIC plugin

Universal topology description of generic distributed infrastructure

- Enables all CRIC features but with simplified Computing Model description
- Basic models for **Compute** and **Storage** Resources (StorageUnit+StorageResource, ComputeUnit+ComputeResource)
- Completely CERN-independent
- Standalone distribution (via images), not coupled to CERN Openstack deployment infrastructure
- Suitable for small VOs or Experiments beyond LHC

Requested by Experiments at JINR (COMPASS, NICA and beyond)
as the Information component
for the Unified Resource Management System

DOMA CRIC

Dedicated CRIC instance for Rucio TPC tests and DOMA related activities

- Provides **Storage description** and related data structures for Experiment agnostic DOMA Rucio instance
- In close cooperation with Rucio experts to polish RSE related models and CRIC interfaces in order to provide appropriate API export for Rucio clients (probes)
- Rucio team has tested RSE configuration coming from DOMA CRIC with Rucio ESCAPE instance. All works well. Look forward for the next integration steps.

- Once CRIC and Rucio integration will be completely tested and evaluated, developed CRIC models and interfaces within DOMA CRIC will be shared with other plugins (ATLAS CRIC, CMS CRIC)

http://escape-cric.cern.ch

WLCG CRIC

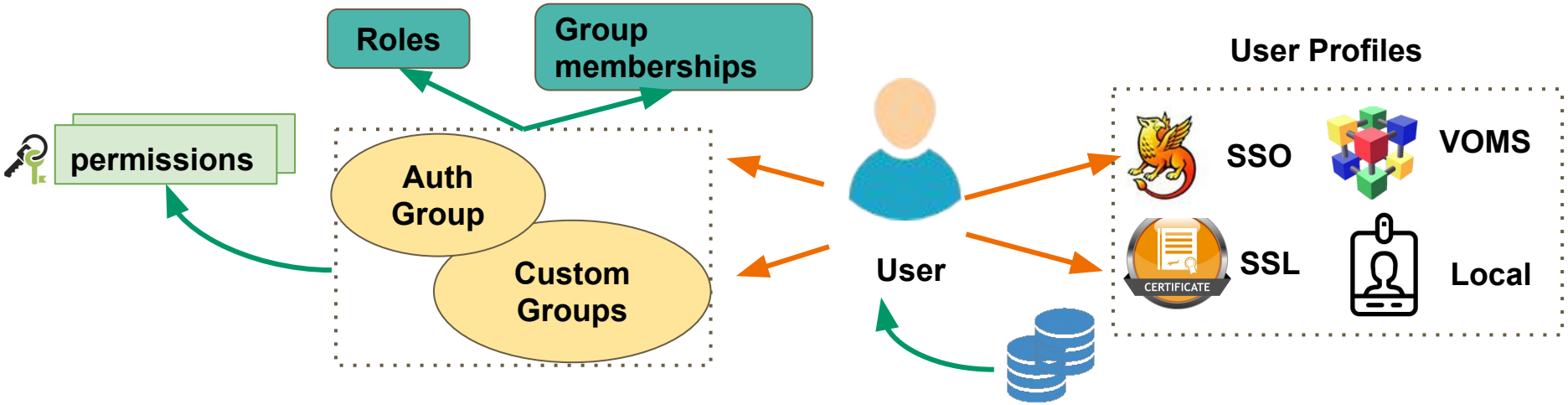
Dedicated CRIC instance for central WLCG operations

- Single entry point for complete WLCG topology description and service configurations for the all 4 LHC experiments
- Main info provider for cross-experiment tools:
  WLCG Accounting, Monitoring, Service Availability, Test submission systems,..
- Federation Pledges management and topology export (**REBUS** replacement)
- VOFeed XML generation (ALICE, LHCb)
- Management of VO Pledge Requirements
- Tracking of various Task Forces and Migration activities
- WLCG Accounting data validation (storage space and CPU capacity from WSSA)
- WLCG Accounting Reporting

http://wlcg-cric.cern.ch

Examples: Authorization and Authentication (A&A)

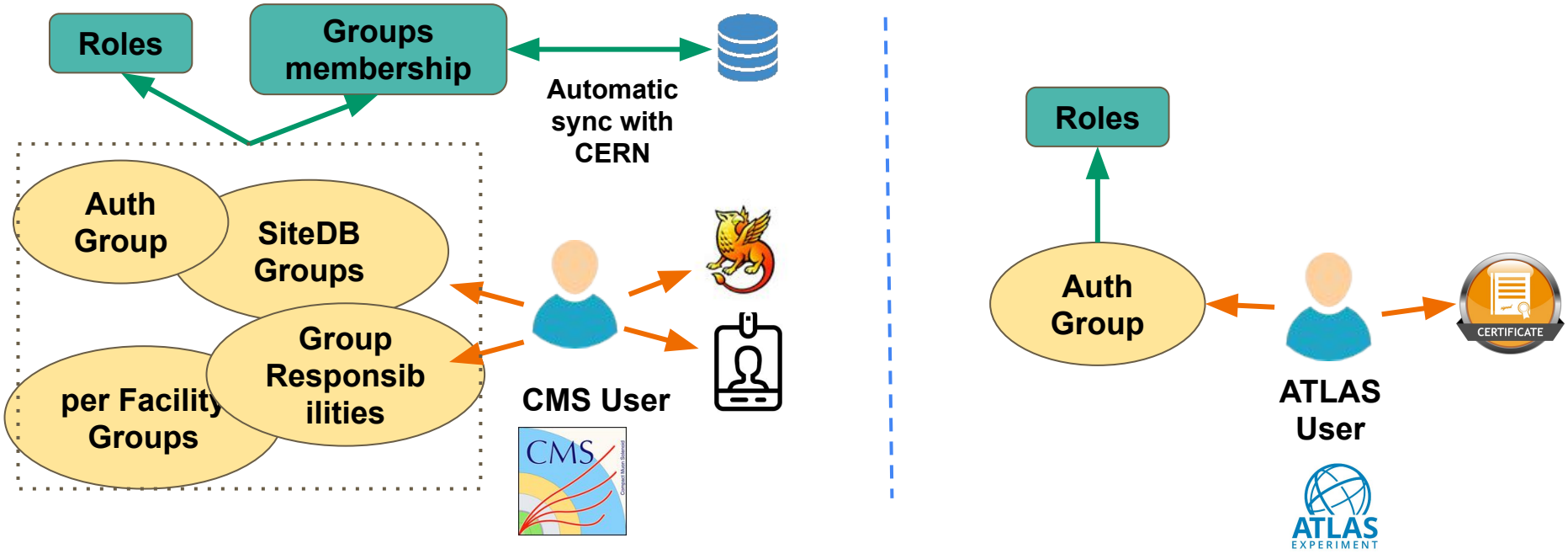
- CRIC supports enhanced **Access controls** and user Group management
- Several Authentication methods enabled (SSO, SSL, VOMS, local)
- Flexible utilisation of Permissions, Roles and Groups at various levels
- Fine grain A&A on the level of object (class, instance, global permissions)
- Ability to bootstrap User info/DB from whatever external source (CERN DB, Experiment DBs, config files, e-groups, VOMS roles, etc)



Each Experiment could configure own Data access policies!

Example of A&A use-cases for different VOs

- **CMS** considers CRIC not only to define access rights within the system, but also to control user privileges for **CMS applications** (CRAB, WMAgent, Phedex, etc…). Relies on CERN SSO and local authentication.
- **ATLAS** uses a simpler Auth concept based on user's DNs coming from VOMS



Experiment decides what elements should be used out of the CRIC box to implement own policies and follow own workflow.

Ongoing core developments

Moving to use-case oriented approach of updating data

- Classical approach assumes to update some set of parameters for specific model of Information schema (="configure these variables for these resources")
- In reality, typical use-case oriented (workflow) update
  - involves modification of **some** attributes of **several** affected models depending on user input
  - requires extra validation and conditional modification
  - user usually does not know which parameter is affected (for example in ATLAS, PandaQueue model has ~ 100 parameters)
- **CRIC will provide a wizard-like workflow forms to process specific use-case for data modification**
- Example: I want to enable remote-io mode for jobs:
  - at which site?
    - for which type of jobs? (ANALY, PROD)
      - for which input storage?
        - for which type of access ? (LAN/WAN), ..

WLCG Conclusion (2019)

- All LHC experiments share common Computing infrastructure.
- **CRIC** offers a common framework describing this infrastructure with also an advanced functionality enabled to describe all necessary Experiment-specific configuration.

  The way the system is designed each experiment can independently describe it's world and still coexist with the others under the same roof.

- **WLCG CRIC** instance represents Computing topology description for sites and services used by all 4 LHC VOs
- **WLCG CRIC** will be used as central entry point for WLCG operations and administration, as well as the main info provider for the cross-experiments WLCG tools.

- Check **CRIC**:
  - http://cms-cric.cern.ch (CMS-CRIC)
  - http://cms-cric-docs.web.cern.ch (CMS-CRIC documentation)
  - http://wlcg-cric.cern.ch (WLCG-CRIC)
  - http://atlas-cric.cern.ch (ATLAS-CRIC)
  - http://escape-cric.cern.ch (ESCAPE-CRIC)
  - http://dune-cric.cern.ch (DUNE-CRIC)
  - http://datalake-cric.cern.ch (DATALAKE-CRIC)