Computing Resource Information Catalog

A unified Information System for WLCG and beyond

Alexey Anisenkov (BINP)

on behalf of CRIC team

WLCG
Worldwide LHC Computing Grid

CERN

CHEP 2018, Bulgaria, 10 July 2018

Distributed Computing Environment (WLCG)

- LHC Experiments rely on huge **heterogeneous** distributed environment
  - variety of Computing Resources involved (GRID, Clouds, HPC)
  - variety of Infrastructures and middleware providers

- Each community uses and describes Resources in its own way
- Computing Models are similar but still have different implementation

WLCG Information landscape

Components of the WLCG Information System

GOCDB

OIM

Open LDAP BDII

REBUS

HTCondor

Other sources

WLCG information consumers

Rucio

PanDA

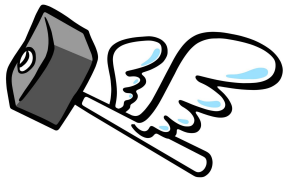
Phedex

GlideinWMS

DIRAC

sam

HammerCloud



ATLAS dashboard

Monitoring tools

And many others…

Current limitations of the WLCG Information System

- Multiple sources of information. Data is sometime contradictory or incomplete. Debugging is complicated.
  No central place where data can be validated
- Integration of new type of resources is not straightforward
- Complex objects like storage services with variety of access protocols and storage shares is not properly described

No high-level information middleware which completely covers Experiments use-cases and describes resource as VOs need

Currently every experiment has to solve all those problems on its own

Solution: a unified Information system

CRIC
Computing Resource Information Catalog

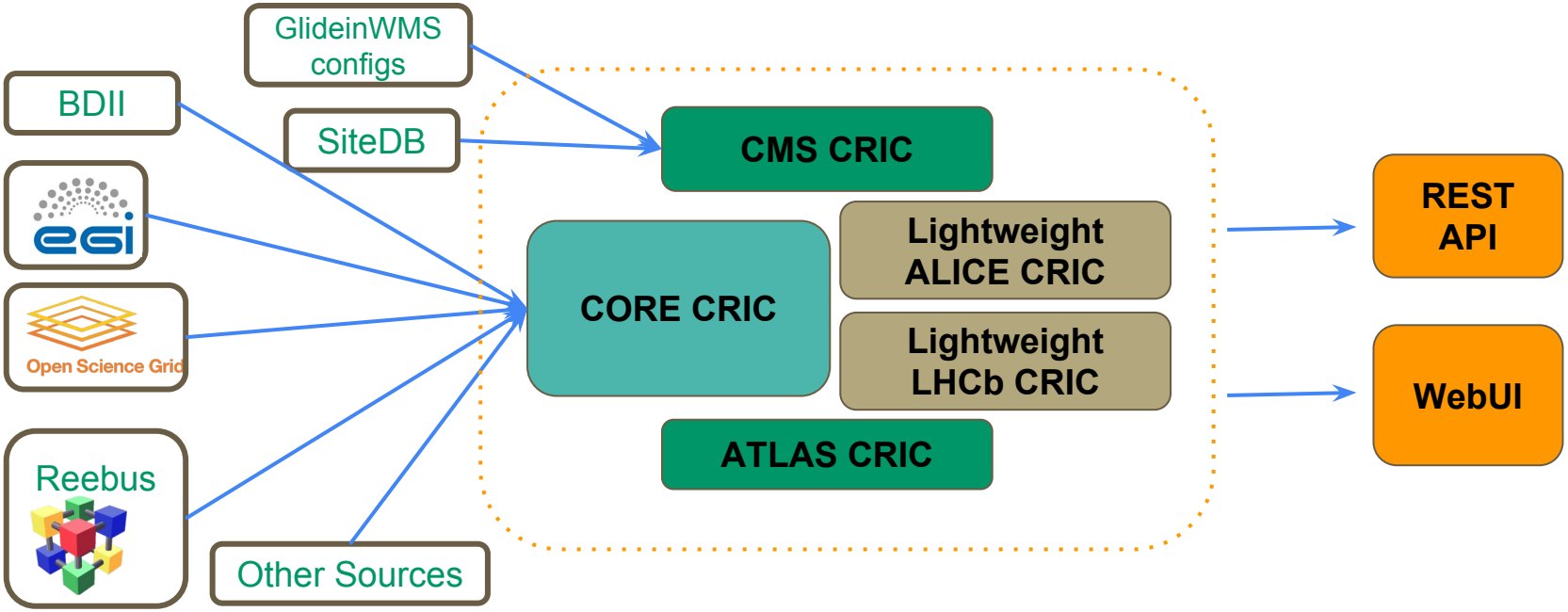
High level information system aiming to describe the topology of the WLCG infrastructure (resources provided by the WLCG sites) and experiment-specific configuration required to exploit this infrastructure according to the experiments Computing models.

CRIC is a framework providing a centralized (and flexible) way to describe which resources LHC experiments are using and also how they use them:

- Clear distinction between resources ***provided by*** (Sites) and resources ***used by*** (Experiments)
- **Experiment independent, but still experiment-oriented**
- Plugin based approach allows customization to address various experiment requirements and implementation of the dedicated experiment instances
- Shared building blocks to optimize development process and to ensure common look and feel. *Think about it in terms of lego bricks*
- Flexibility to address technology evolution and changes in the experiment computing models and applications. *Lego bricks again!*

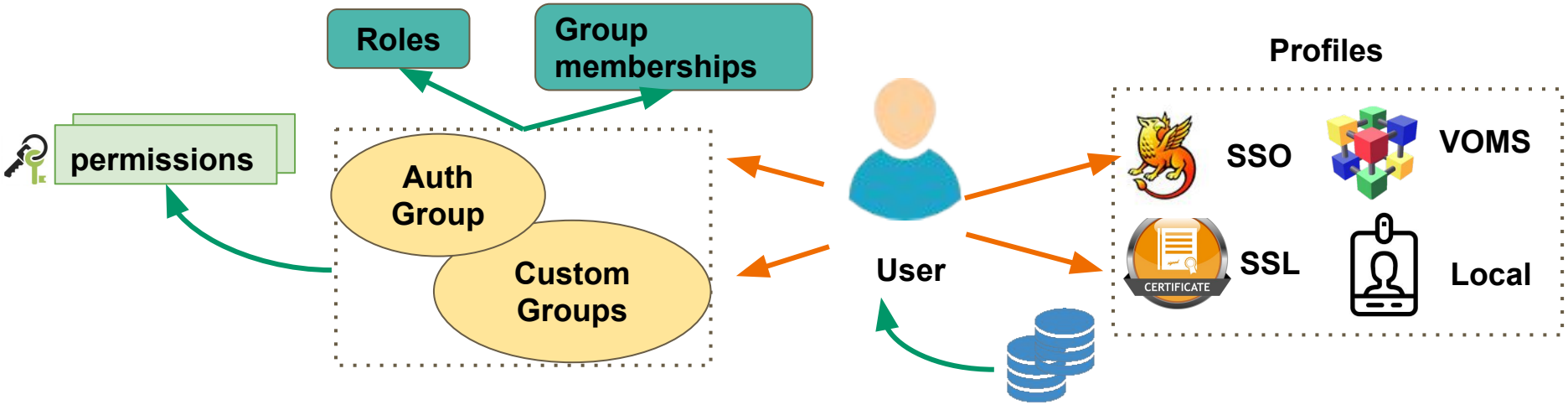
CRIC Architecture: plugin based

- Modular architecture is based on the Django framework, implementing different apps for CORE (provided by) and experiment (used by) parts
- Data are exposed via REST API which is configurable by filters and different presets (views).
- Bootstrap, jQuery, Web services and many other modern tools and technologies are used



Authorization and Authentication (A&A)

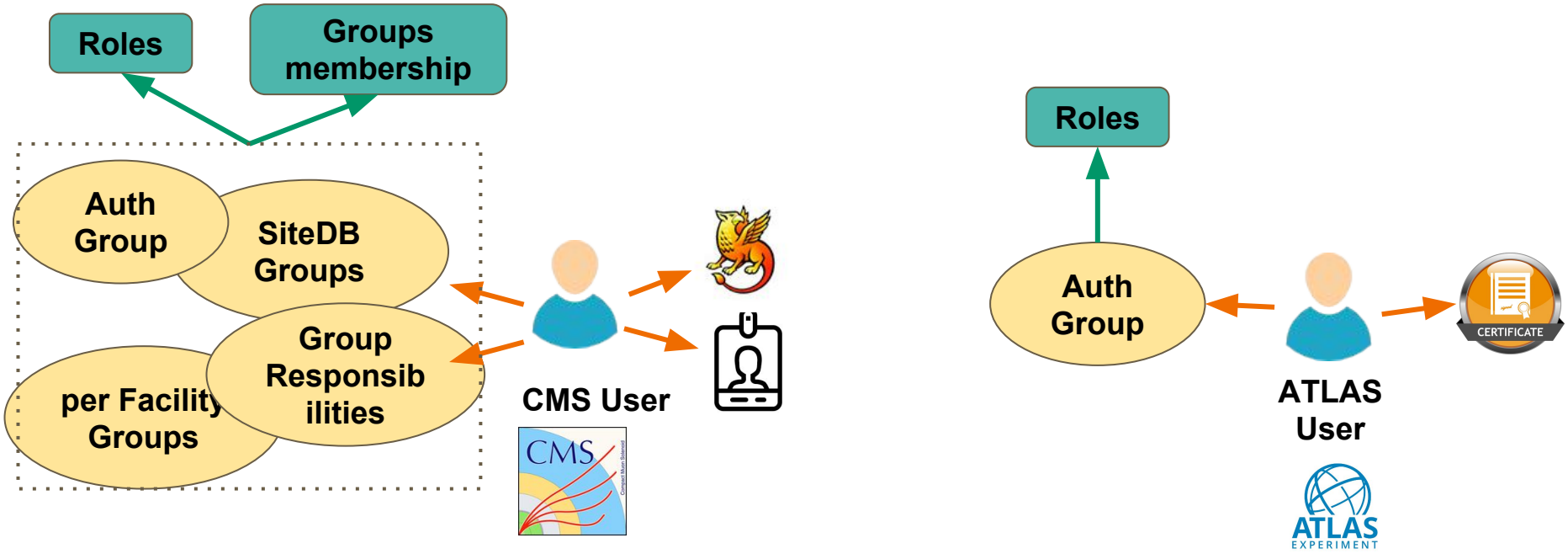
- CRIC supports enhanced Access controls and user Group management
- Several Authentication methods are enabled
- Flexible utilisation of Permissions, Roles and Groups at various levels
- Fine grain A&A on the level of a single CRIC object
- Ability to bootstrap User info from whatever external source (CERN user DB, Experiment DBs, config files, e-groups, etc)



Each Experiment configures own Data access policies!

Example of A&A use-cases for different experiments

- CMS is planning to use CRIC not only to define access rights for CRIC objects, but also to define/expose user privileges for other **CMS applications** (CRAB, Phedex, etc…). Relies on CERN SSO and local authentication.
- ATLAS uses a simpler 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.

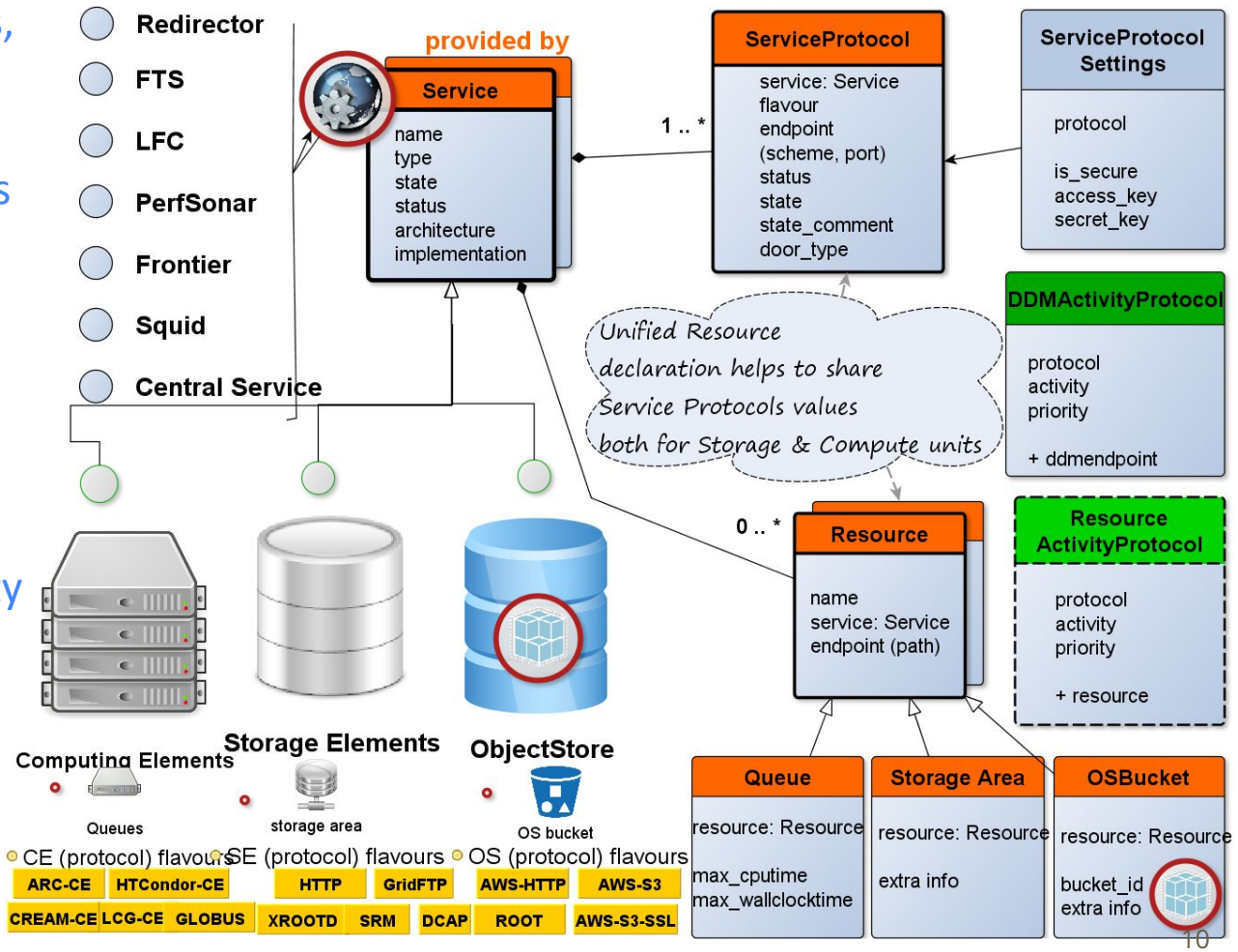
Logging functionality - track the changes

- CRIC provides an advanced logging functionality to monitor, administer and troubleshoot the system
- Logging is performed at the object level (a given object or any other object related to it)
- Full list of changes (including old values) is provided through build-in table view.
- You can check **who**, **when** and **how** interacted with an object.

Common concepts. Storage example

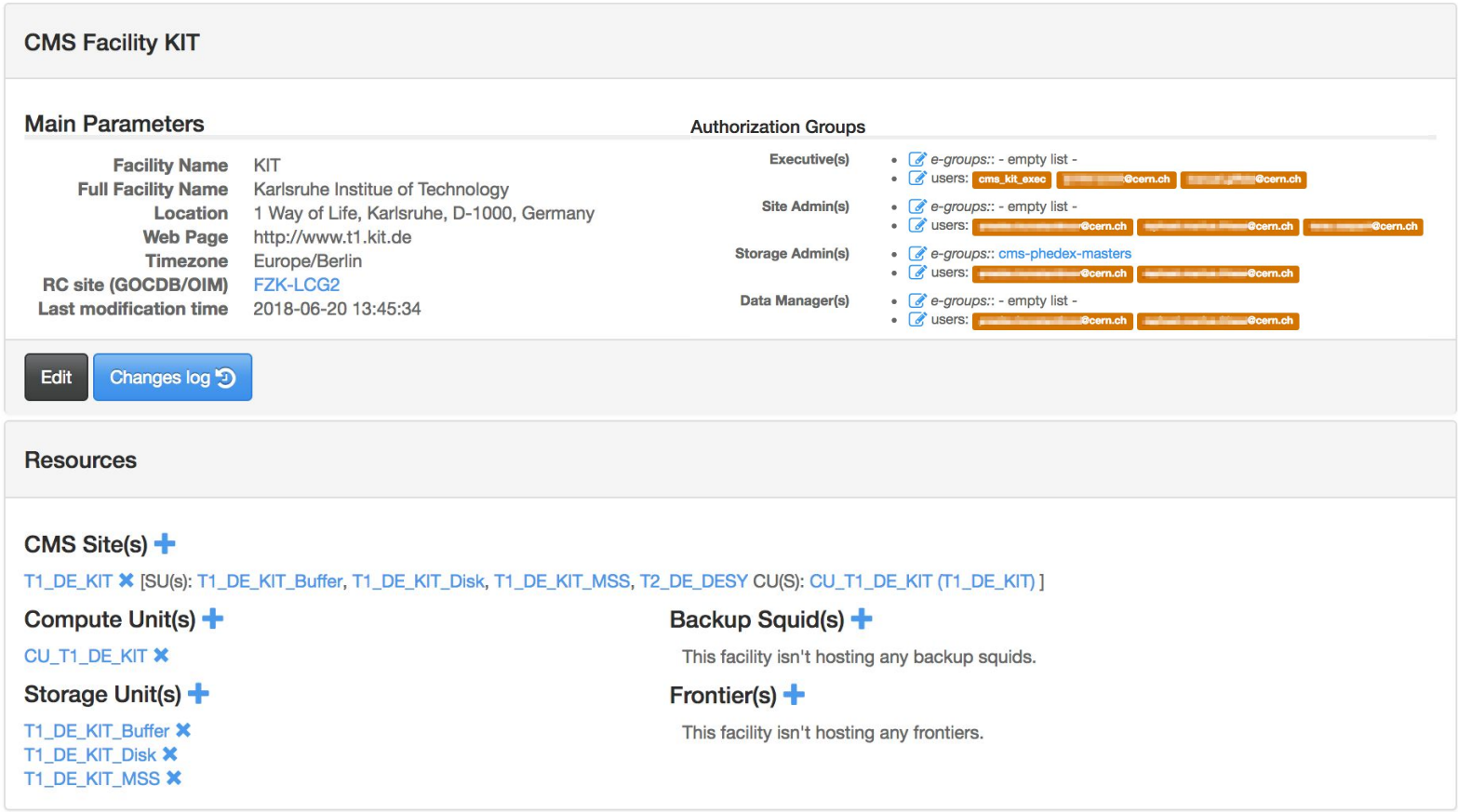
CRIC introduces unified description of Service Resources in particular to target and resolve the complexity of SE definition

- Link together all protocols, activities, closeness metrics, space tokens, other experiment specifics belong to same Storage into unified Resource
- Multiple protocols concept
- Connect associated CE to default SE for given activity
- Integration of new SE technologies (e.g. ObjectStores) within the experiments



Experiment specific concepts. CMS Facility example

- Facility is a **CMS specific** concept
- An aggregation of set of sites and services in a CMS administrative domain

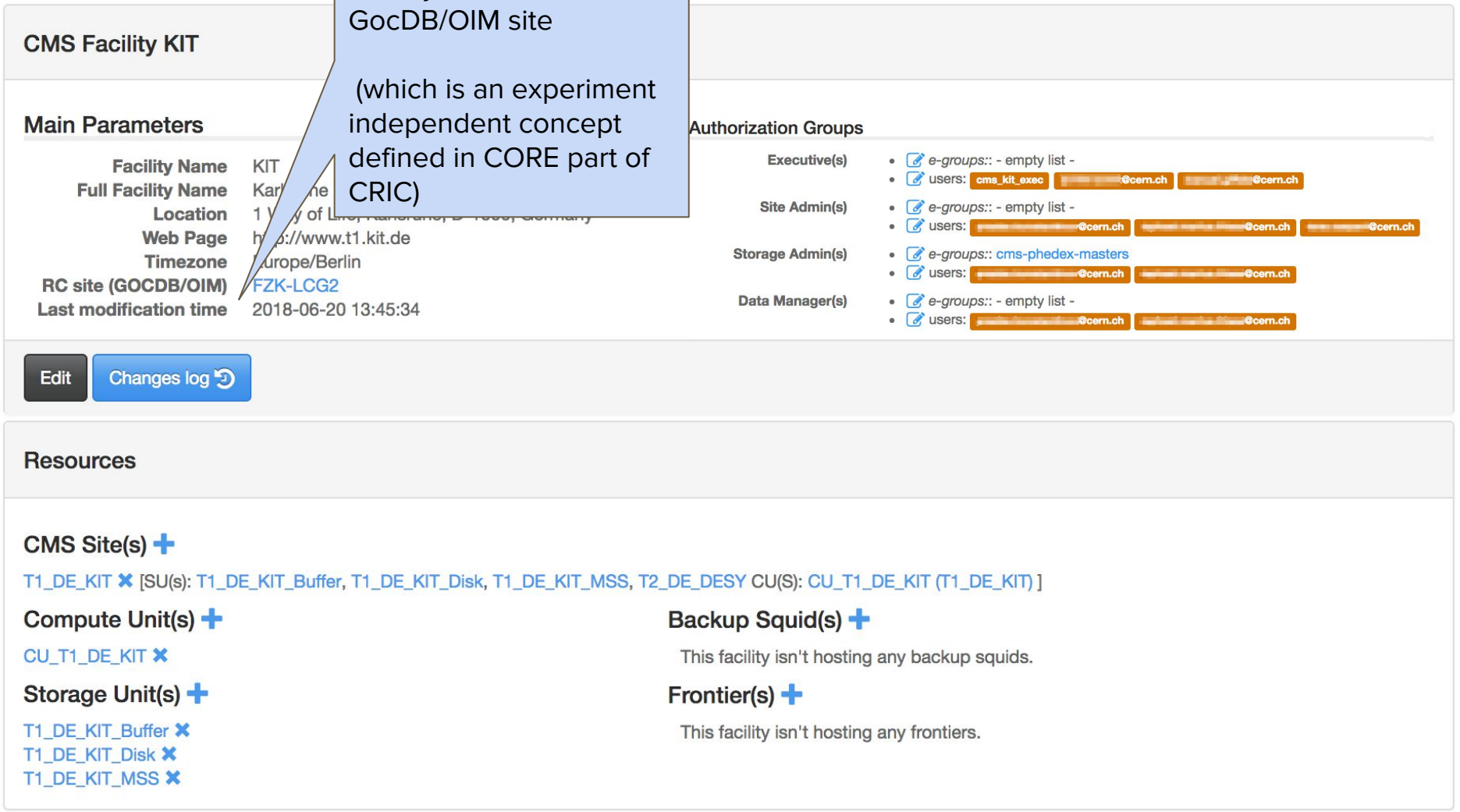


Experiment … epts. CMS Facility example

General details and links to other objects.

Facility is linked to a GocDB/OIM site

(which is an experiment independent concept defined in CORE part of CRIC)



Experiment [illegible] epts. CMS Facility example

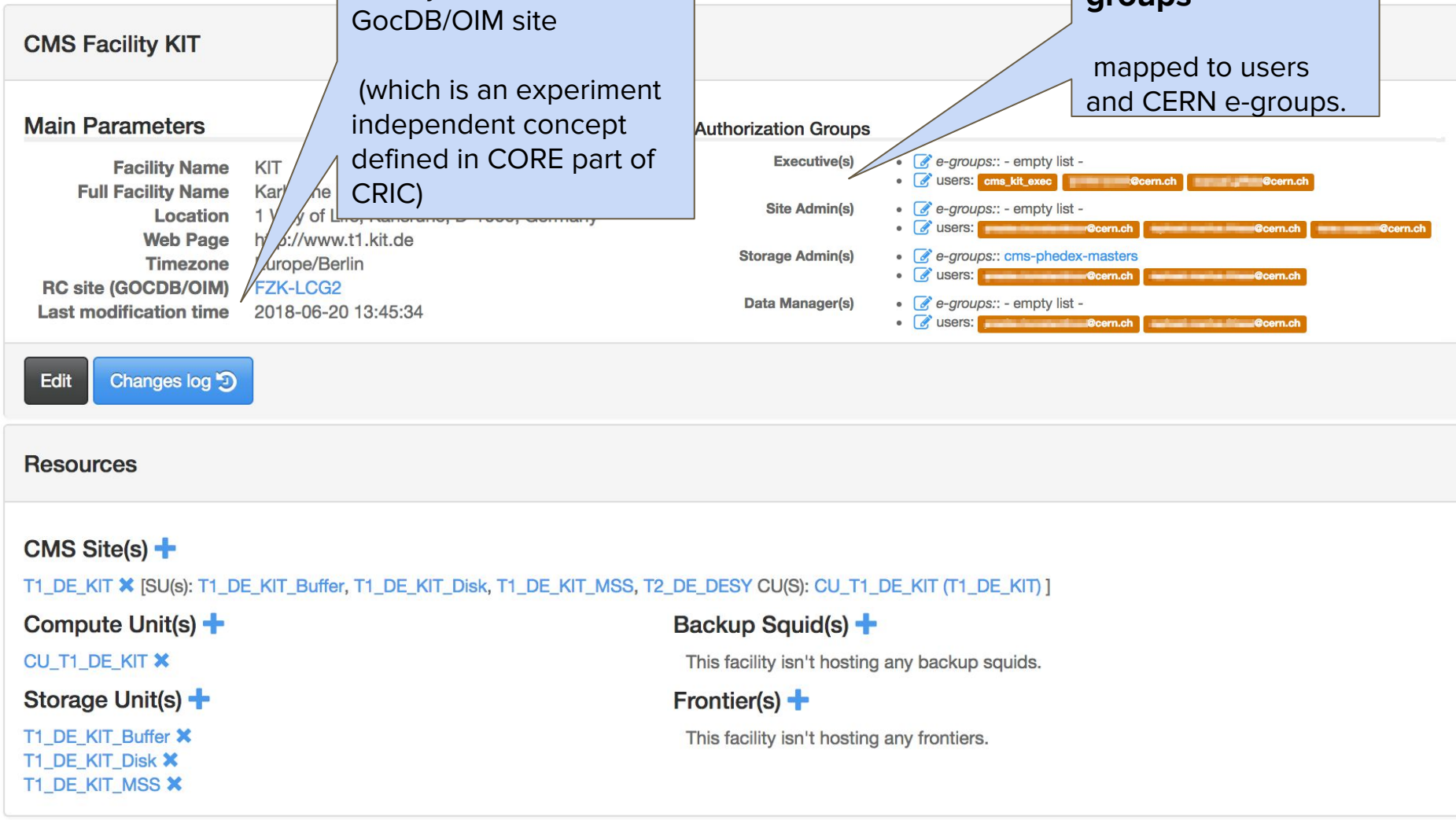
General details and links to other objects.

Facility is linked to a GocDB/OIM site

(which is an experiment independent concept defined in CORE part of CRIC)

Per-Facility authorization groups

mapped to users and CERN e-groups.



Experiment ... epts. CMS Facility example

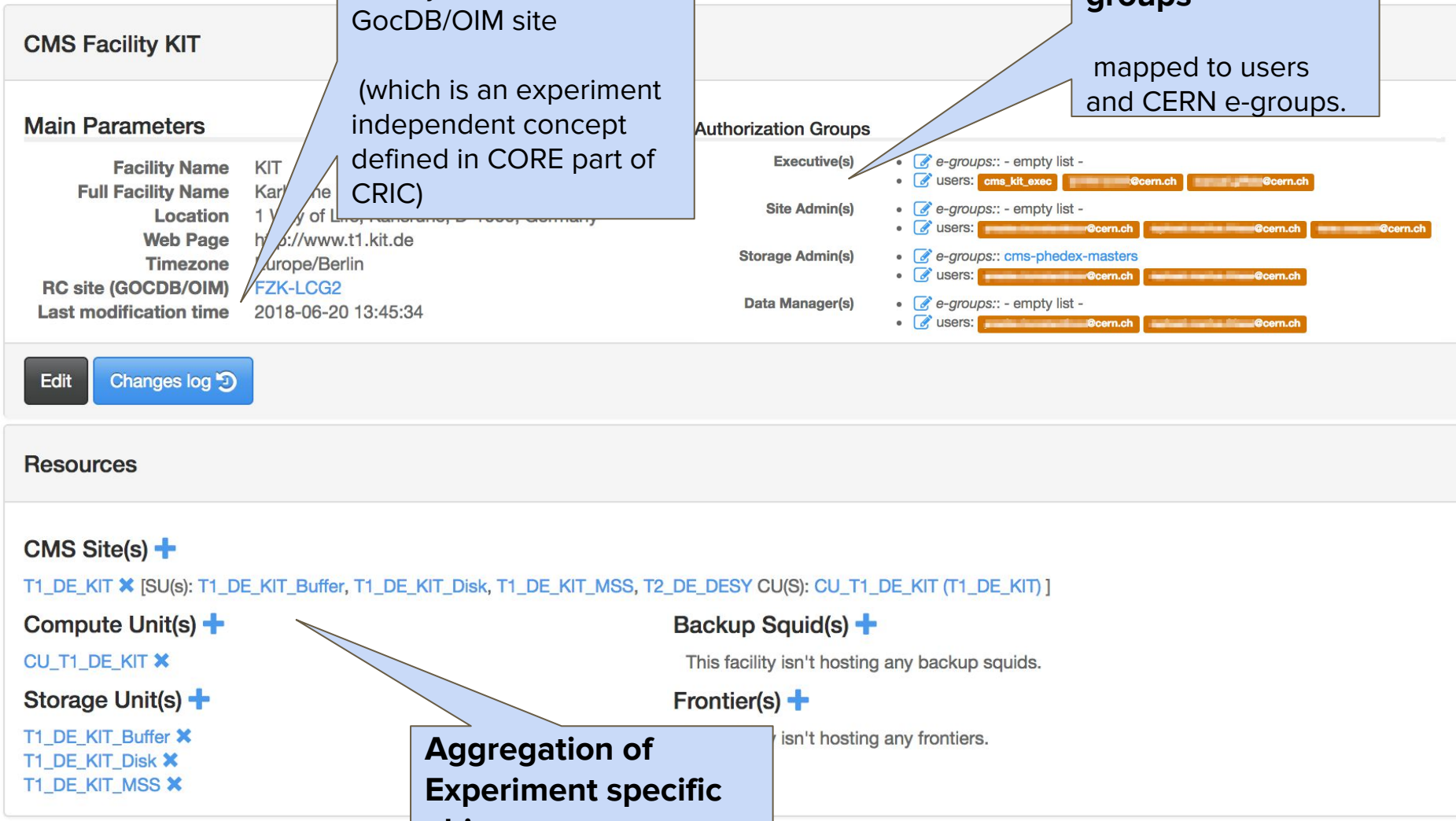
General details and links to other objects.

Facility is linked to a GocDB/OIM site

(which is an experiment independent concept defined in CORE part of CRIC)

Per-Facility authorization groups

mapped to users and CERN e-groups.



Aggregation of Experiment specific objects

like Sites, Compute and Storage units ...

Experiment [illegible] epts. CMS Facility example

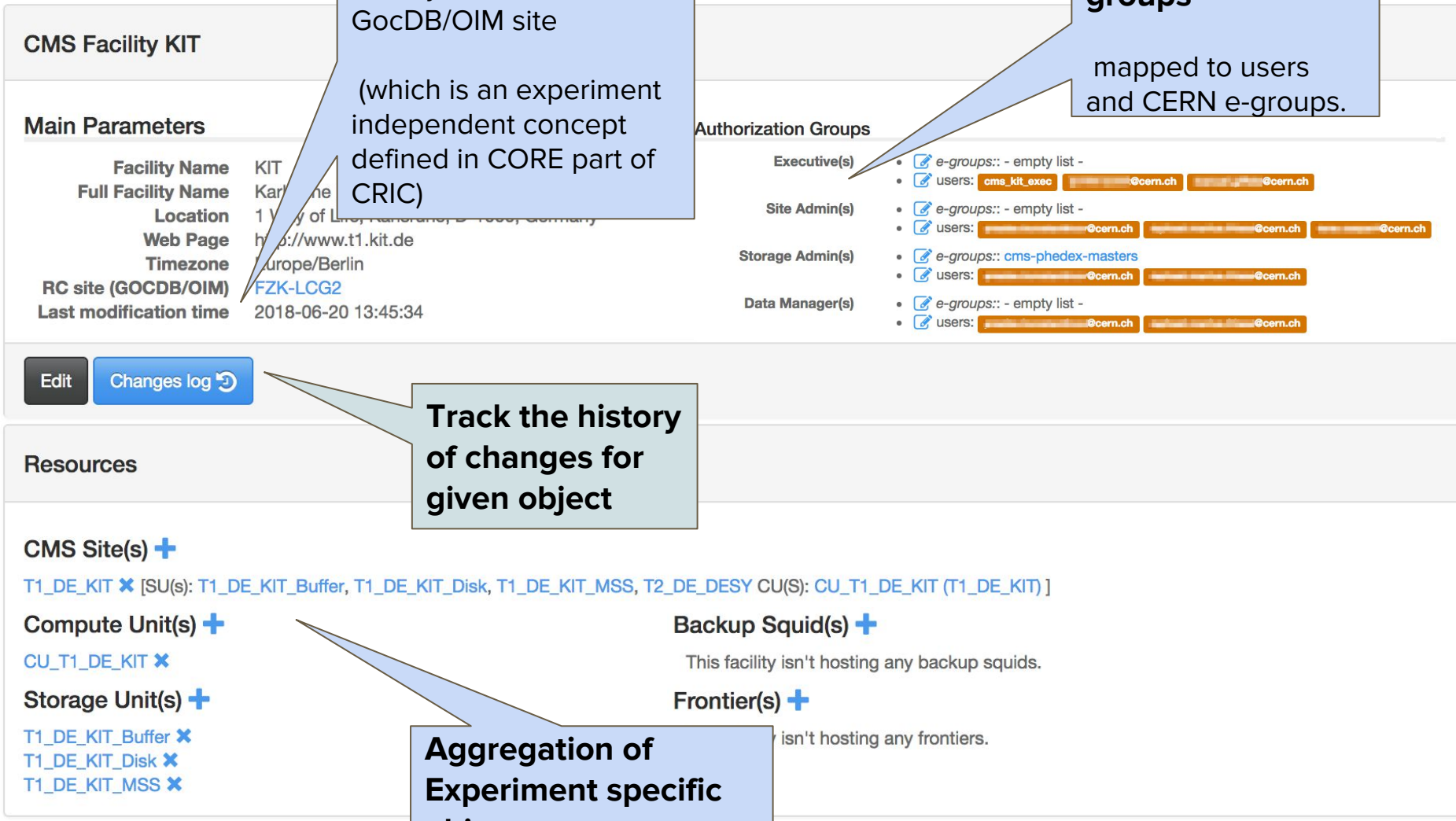
General details and links to other objects.

Facility is linked to a GocDB/OIM site

(which is an experiment independent concept defined in CORE part of CRIC)

Per-Facility authorization groups

mapped to users and CERN e-groups.



Track the history of changes for given object

Aggregation of Experiment specific objects
like Sites, Compute and Storage units ...

Experiment-specific concepts. Computing resource example

- **Experiments use different systems to submit jobs in the grid**
- **CMS** is submitting Pilots through GlideinWMS. Currently GlideinWMS configuration is described in the XML files hosted by Github. This configuration has been imported into CRIC.
- **ATLAS** is submitting Pilots through AutoPilotFactories (APF), ARC ControlTower (aCT), and now Harvester.
  - These frameworks require config files which describe the Computing Elements (and batch system underlying)
  - These configs are auto-generated from AGIS, exploiting the resource description in "core", and in the future will come from **CRIC**

Status and plans

- **CMS CRIC** instance is ready for validation by the CMS community
- Ongoing implementation of CRIC instance for WLCG central operations. It represents CORE part with sites and services used by all 4 LHC experiments

Next steps:

- Porting **REBUS** functionality into CRIC
- Extending **CMS CRIC** functionality following CMS feedback
- Migrating **AGIS** to **ATLAS CRIC**

Conclusions

- All **LHC experiments** are sharing common computing infrastructure. CRIC offers a common framework describing this infrastructure, but also an advanced functionality 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.

- First CRIC version mainly focused on the **CMS-required** functionality is ready for validation. We need you to start using **CMS CRIC** and provide us with use cases you think need improvement and error-proofing!

- Check CRIC at http://cms-cric-qa-01.cern.ch