



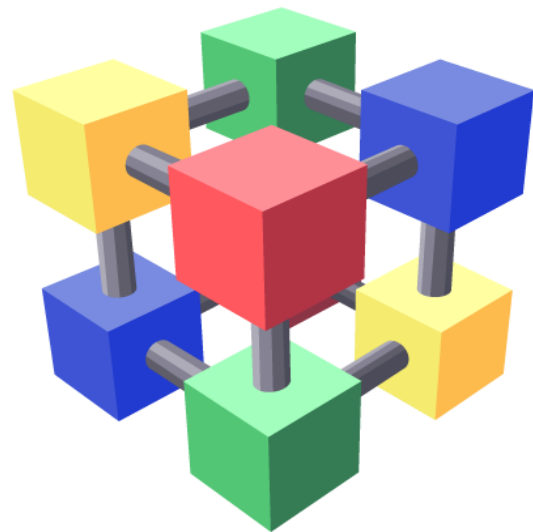
API for Computing Resource Information Catalog (CRIC)

Ravin Kohli

Supervisors

Julia Andreeva,
Alessandro Di Girolamo

WLCG



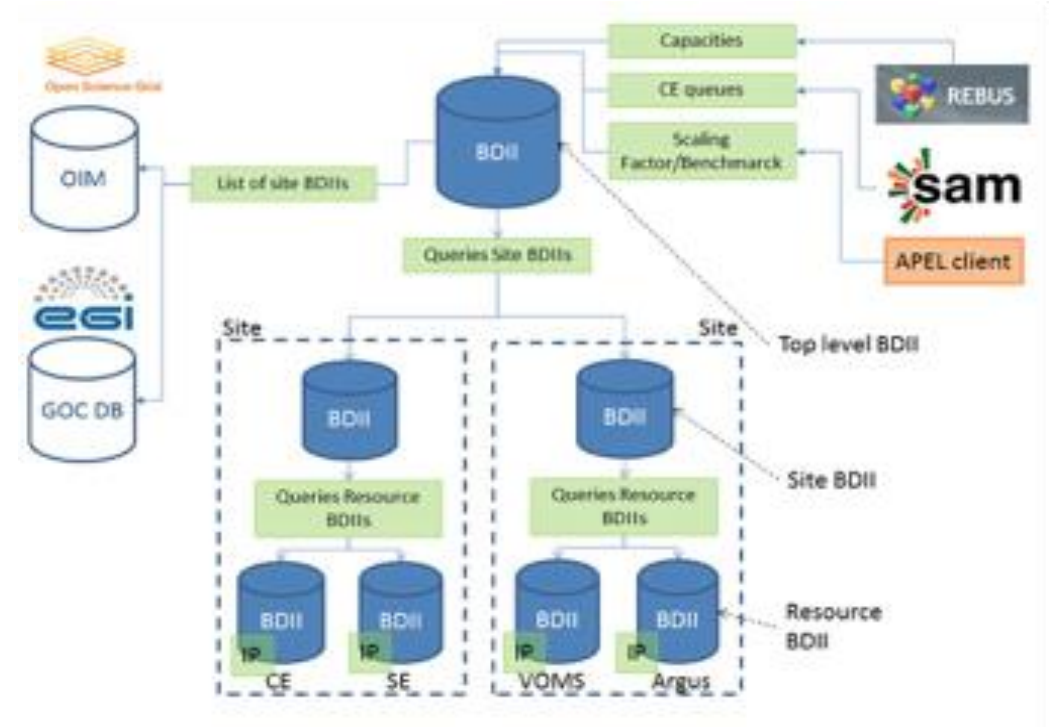
WLCG

Worldwide LHC Computing Grid

The Current System

WLCG Information System

1. Provide detailed information about grid sites and services
1. Fundamental building block is Berkeley Database Information Index (BDII) which can be visualised as a LDAP database



The Problem

Weaknesses of the Current WLCG Information System

1. Lack of reliability of provided data
1. Incomplete description of all available resources (opportunistic, HPC, commercial clouds)
1. No single entry point for WLCG topology and configuration

What Is CRIC?

Computing Resource Information Catalog

1. CRIC is an information catalog for the WLCG
1. The main requirements for CRIC are simplicity, robustness, modifiability and agility
1. Implemented in Python using Django framework

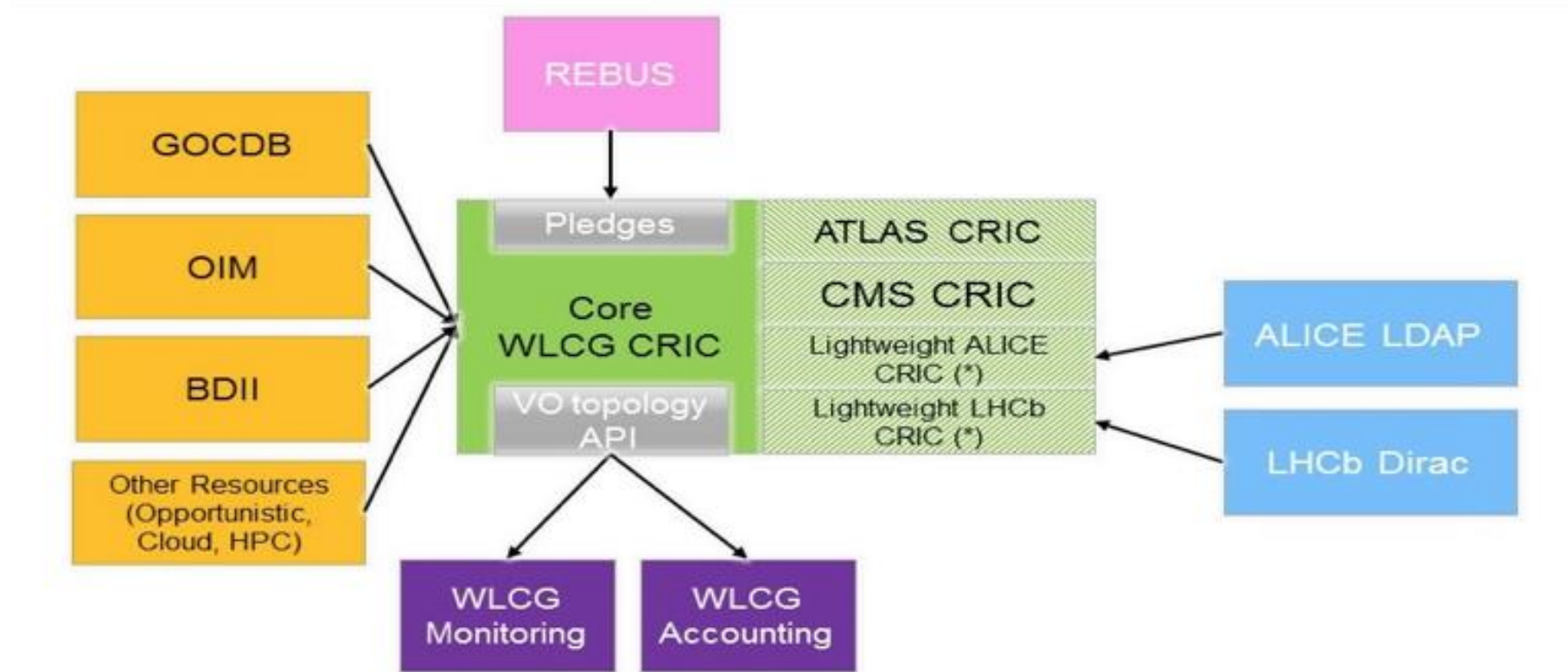


Motivation

Why do we need CRIC?

1. Combines WLCG service topology information with experiment specific configuration
1. Allows to integrate all kind of resources, not just classical GRID resources
1. Provides logging functionality (when/how/by whom data has been modified)

CRIC Architecture



(*) Maintained by WLCG to store very simple experiment topology information (i.e. experiment names)

My Work

RESTful API in Django for CRIC information export

1. Need: RESTful APIs for export

1. Aim: To provide relevant and concise information for potential clients. (monitoring, accounting, testing, workload management systems and data management systems)

Results

Federation API

```
{
  "AT-HEPHY-VIENNA-UIBK": {
    "accounting_name": "AT-HEPHY-VIENNA-UIBK",
    "country": "Austria",
    "id": 2,
    "name": "AT-HEPHY-VIENNA-UIBK",
    "pledges": {
      "2017": {
        "atlas": {
          "CPU": 1857,
          "Disk": 120
        },
        "cms": {
          "CPU": 10000,
          "Disk": 500
        }
      }
    },
    "sites": [
      "HEPHY-UIBK",
      "Hephy-Vienna"
    ],
    "tier_level": 2,
    "vo": [
      "atlas",
      "cms"
    ]
  },
  "-----"
}
```

Results (Now Prettier!!)

Federation API

```
{
  "AT-HEPHY-VIENNA-UIBK": {
    "accounting_name": "AT-HEPHY-VIENNA-UIBK",
    "country": "Austria",
    "id": 2,
    "name": "AT-HEPHY-VIENNA-UIBK",
    "pledges": {
      "2017": {
        "atlas": {
          "CPU": 1857,
          "Disk": 120
        },
        "cms": {
          "CPU": 10000,
          "Disk": 500
        }
      }
    },
    "sites": [
      "HEPHY-UIBK",
      "Hephy-Vienna"
    ],
    "tier_level": 2,
    "vo": [
      "atlas",
      "cms"
    ]
  },
  ...
}
```

Federation API

```
▼ AT-HEPHY-VIENNA-UIBK {8}
  accounting_name : AT-HEPHY-VIENNA-UIBK
  country : Austria
  id : 2
  name : AT-HEPHY-VIENNA-UIBK
  ▼ pledges {1}
    ▼ 2017 {2}
      ▼ atlas {2}
        CPU : 1857
        Disk : 120
      ▼ cms {2}
        CPU : 10000
        Disk : 500
  ▼ sites [2]
    0 : HEPHY-UIBK
    1 : Hephy-Vienna
    tier_level : 2
  ▼ vo [2]
    0 : atlas
    1 : cms
```

What I learnt?

1. Django REST framework Vs Using native Django code

1. Mapping different model relationships using filters

1. Information system of the WLCG infrastructure and various operational aspects of the distributed GRID infrastructure



THANKS

kohlravin7@gmail.com

Ravin Kohli
B. Tech
Software Engineering
Delhi Technological University