CERN Summer Student Program

Site Mapping in DIRAC

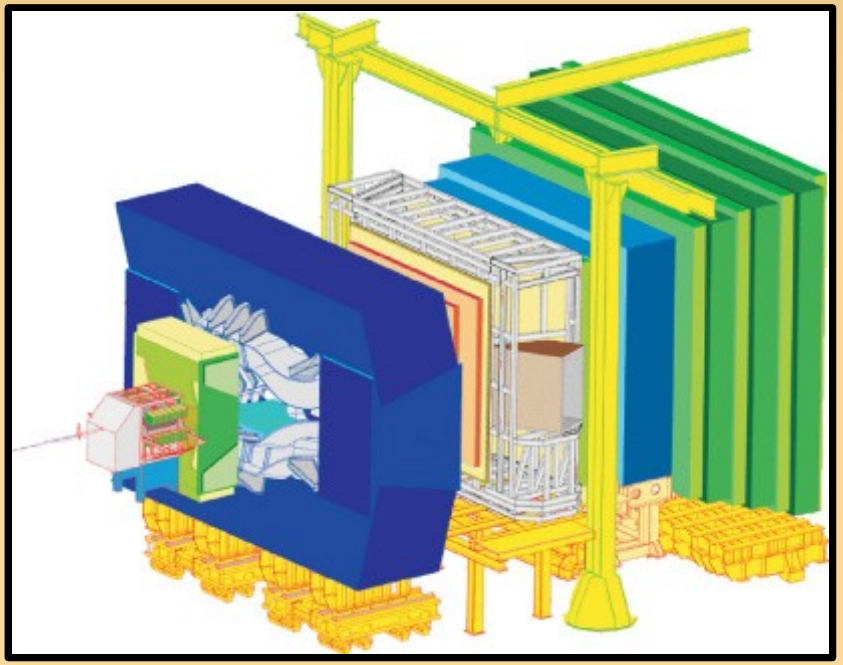
Adam J. Sypniewski, Alma College
Dr. Stuart Paterson

LHCb Experiment

- LHCb is a forward, single-arm spectrometer
- It will investigate CP violation and rare decays in B-mesons

Computing Requirements

- Monte Carlo simulation data
- Collision data
- 80 MB/s detector output
- 1 Petabyte/year total



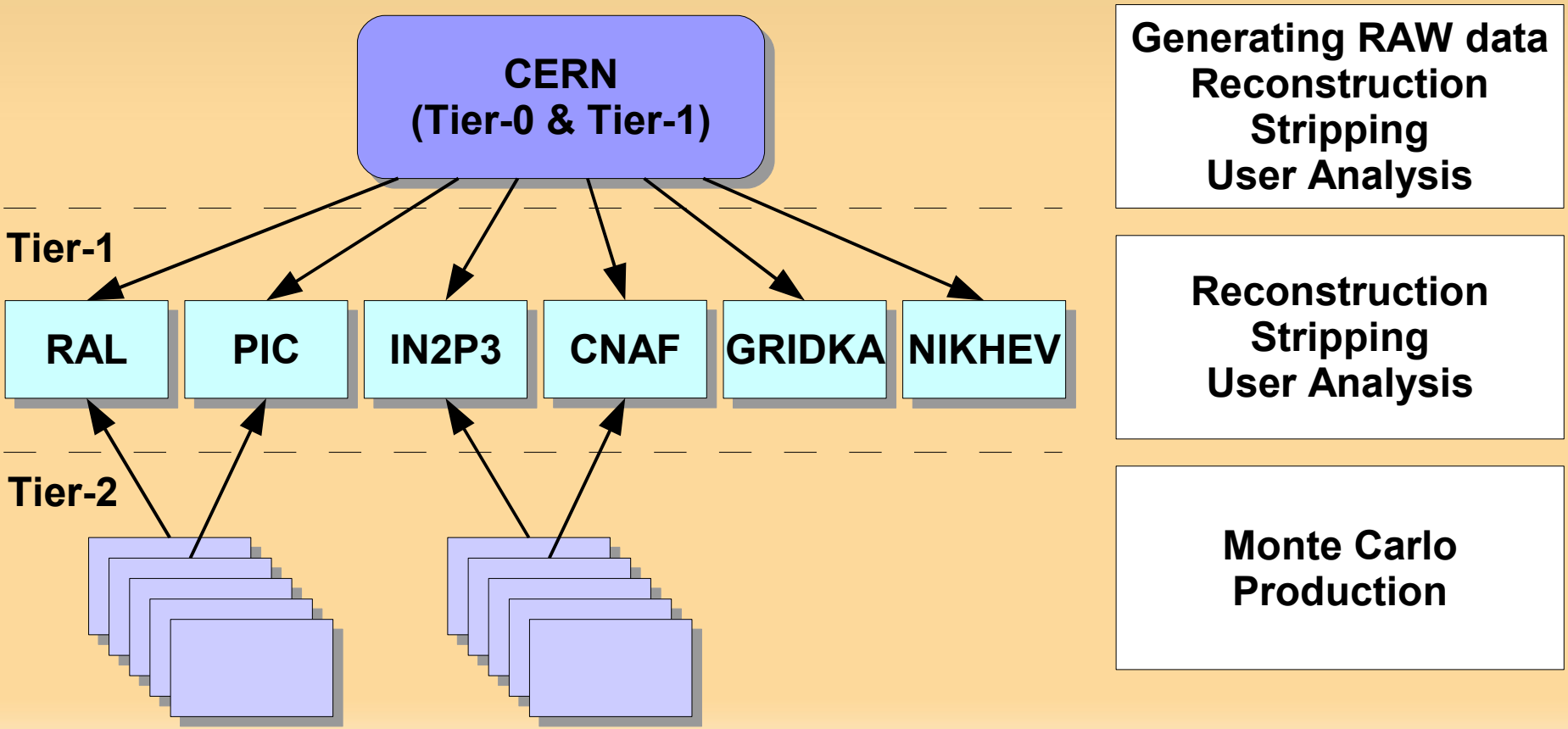
Intensive computing resources required.

Far more than any single computer system can handle.

LHCb Computing Model

Decentralized resources – computers on the Grid share resources (e.g., processing power, data storage)

- Scientists can run jobs without thought of where they will actually be executed.
- Job status and output can be monitored.
- Grid paradigm is the necessary solution to handle volume of LHCb data.



DIRAC Overview

DIRAC (**D**istributed **I**nfrastructure with **R**emote **A**gent **C**ontrol) is the Grid solution for LHCb.

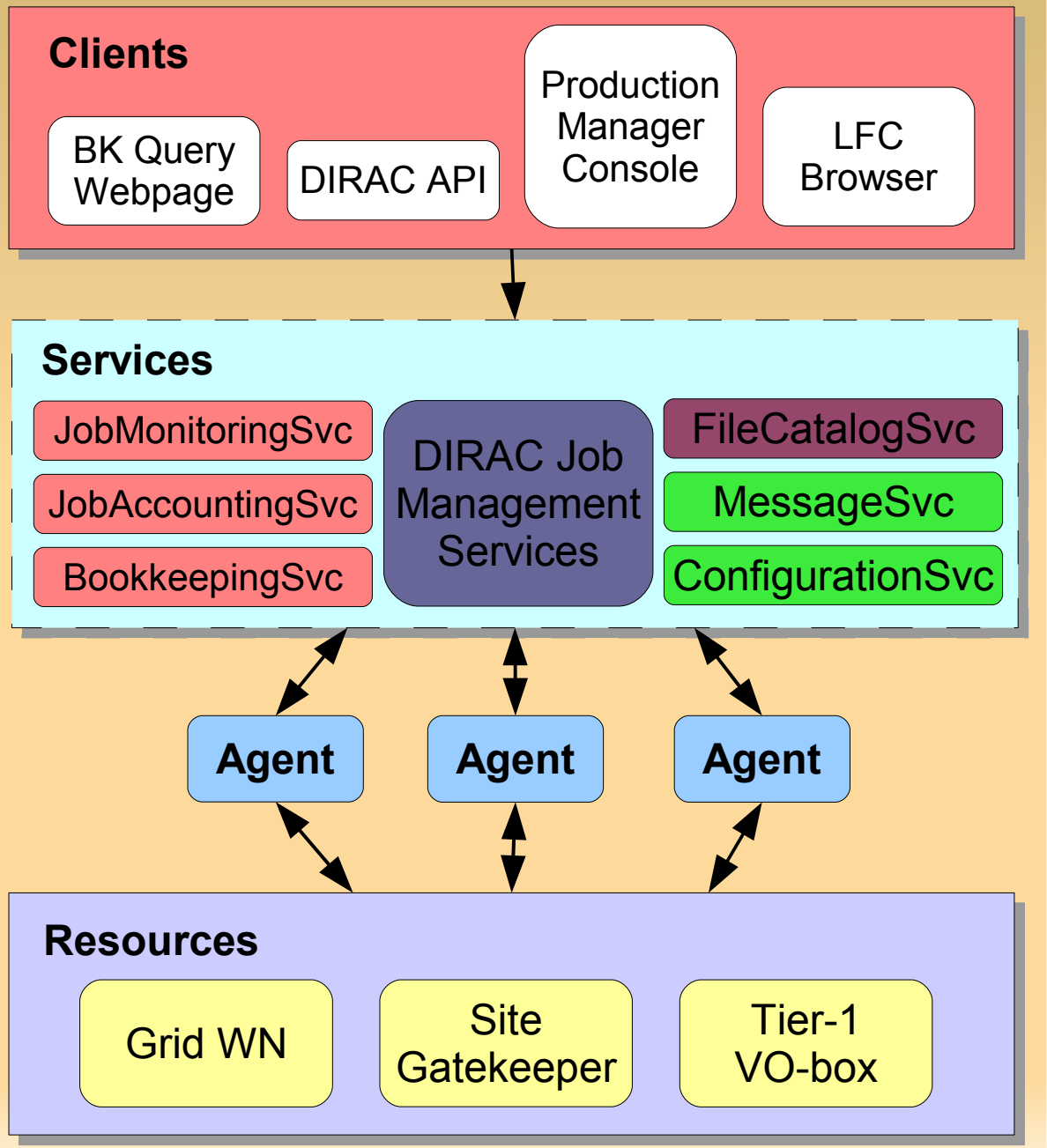
DIRAC aggregates different types of computing resources:

- Individual computers
- Computing sites
- Grids

DIRAC uses a combination of:

- Clients
- Services
- Agents
- Resources

DIRAC is robust, modular, and portable.



DIRAC Pilot Agents

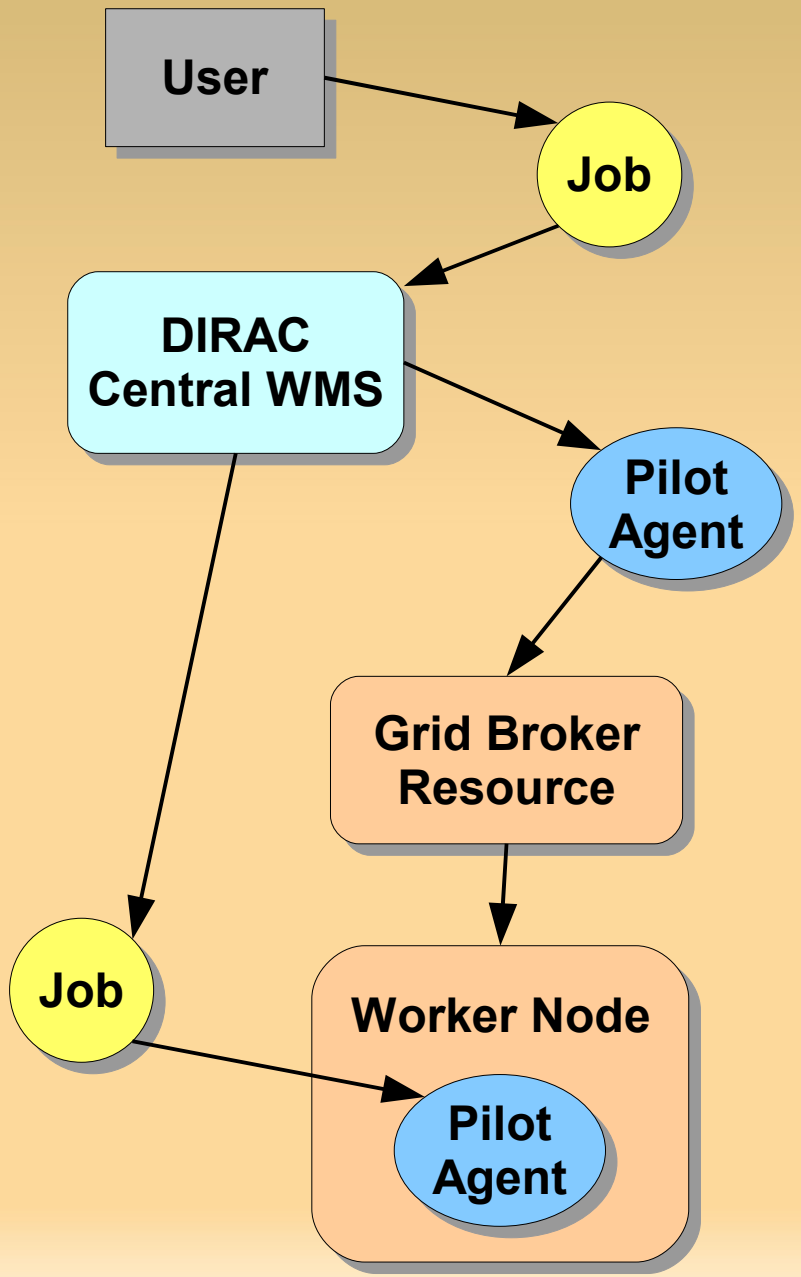
DIRAC uses **pilot agents** to run jobs

- Maximize LCG efficiency
- Minimize load

DIRAC WMS chooses a suitable site (e.g., access to files on LFC)

Pilot Agents

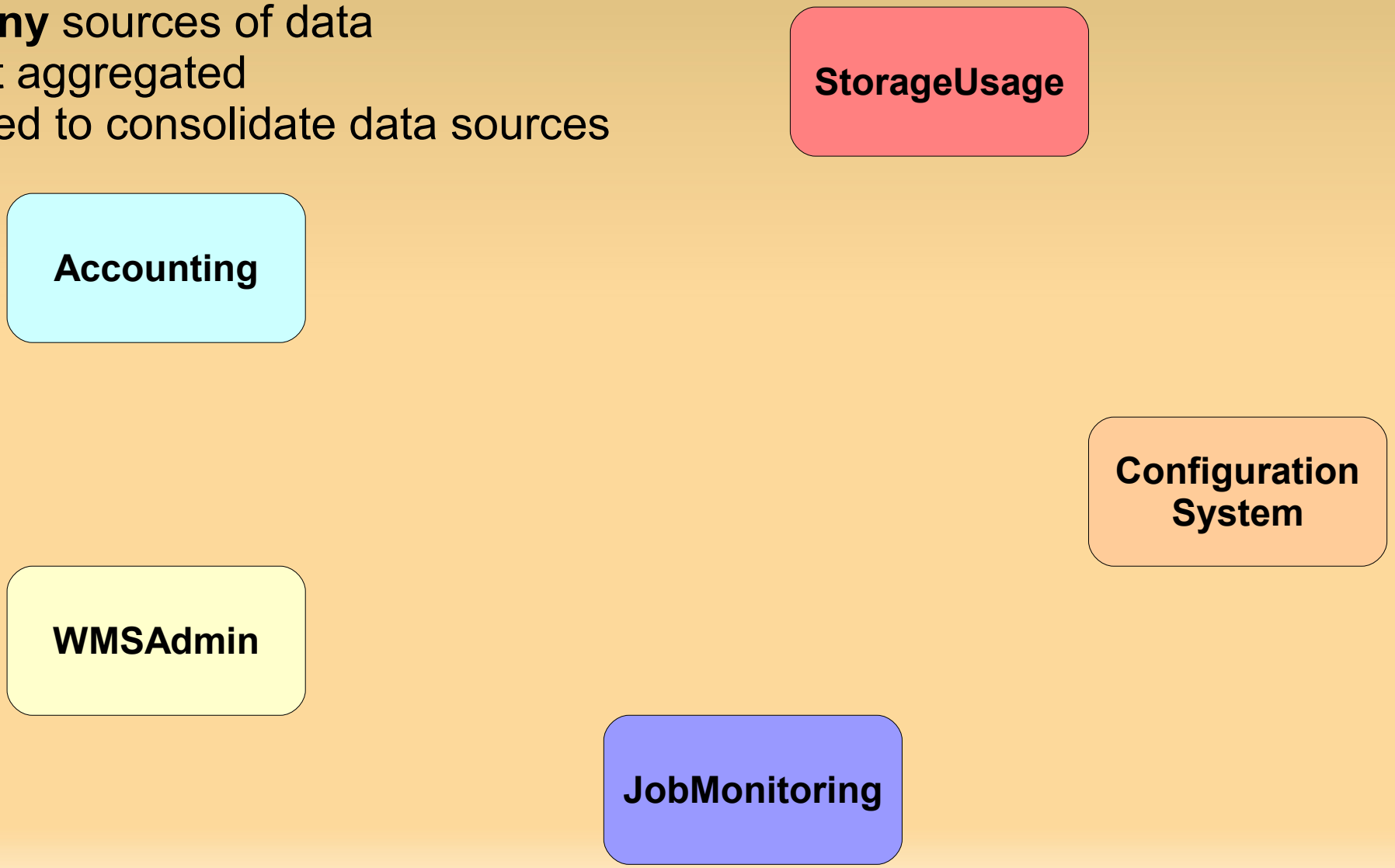
- Install & configure DIRAC on the WN
- Can reuse WN for additional jobs



Motivation

DIRAC is complex

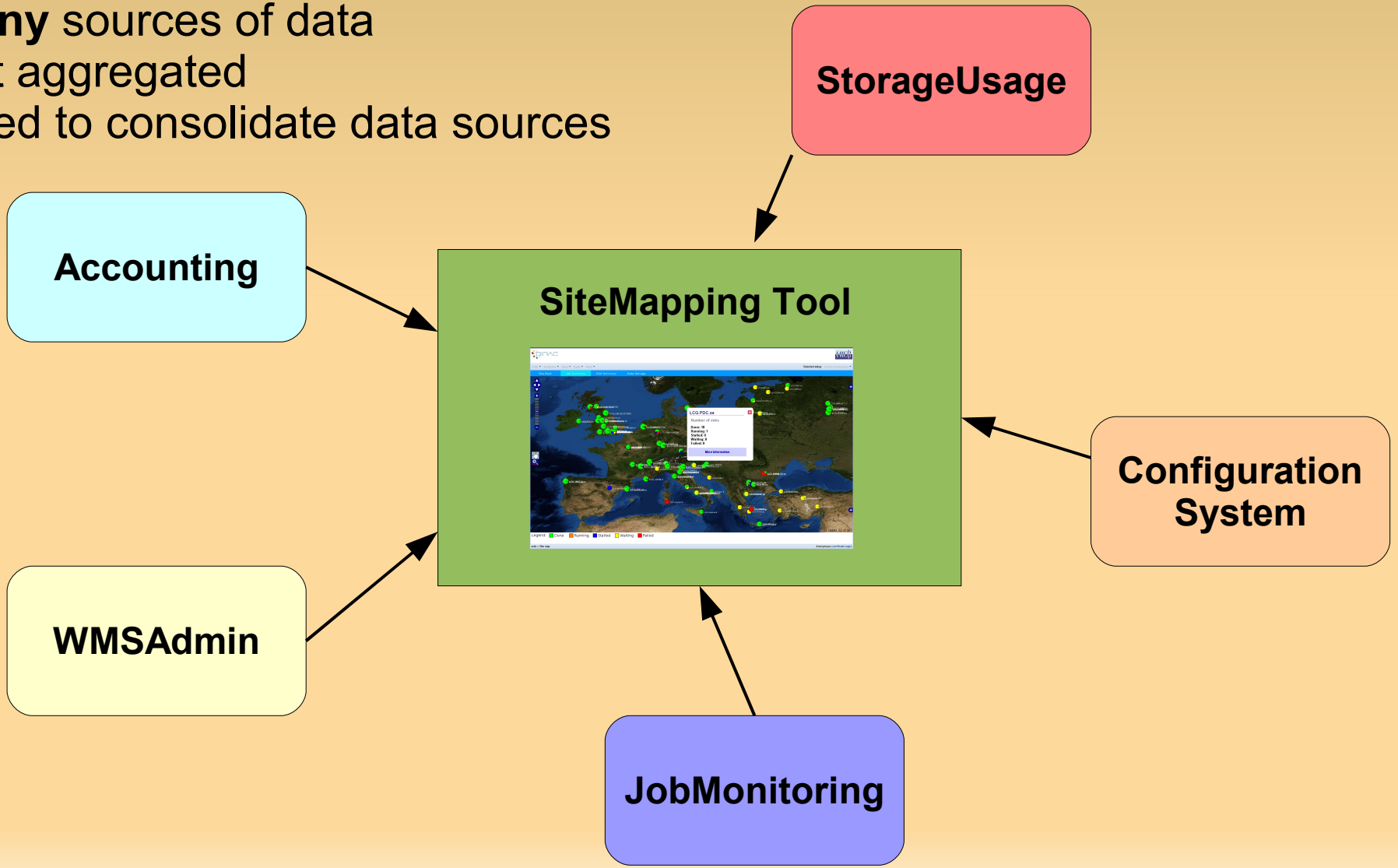
- **Many** sources of data
- Not aggregated
- Need to consolidate data sources



Motivation

DIRAC is complex

- **Many** sources of data
- Not aggregated
- Need to consolidate data sources



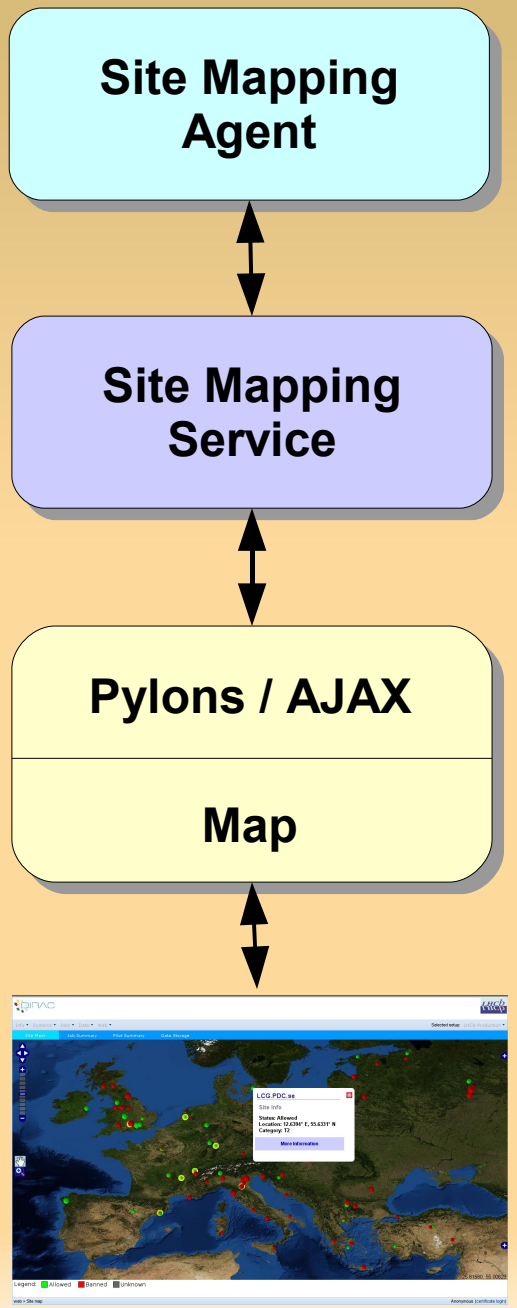
Site Mapping

Intuitive site mapping tool

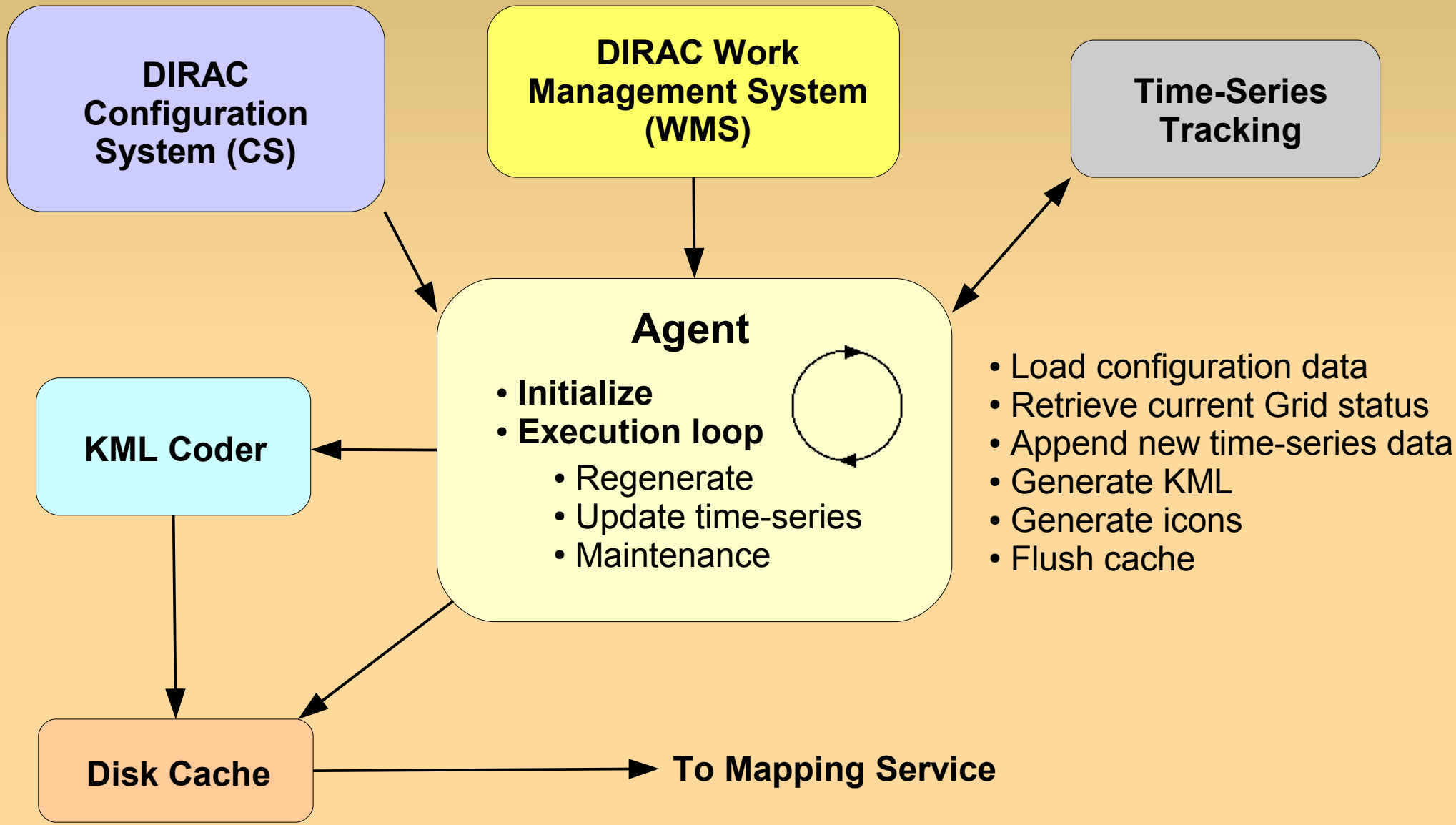
- Map-based system
- Geographical, status-based icons
- Interactive tools for administration
- Up-to-date information
- Small data sizes

Three part system

- Agent – data gathering
- Service – file transfers
- Front-end – interactive display



The Mapping Agent



The Mapping Service

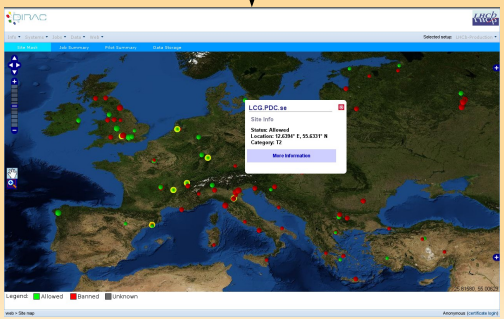
From Mapping Agent → **Disk Cache**

DIRAC Configuration System (CS) → **Service**

Service

- **Initialize**
- **Service loop**
  - Wait for request
  - Transfer binary data

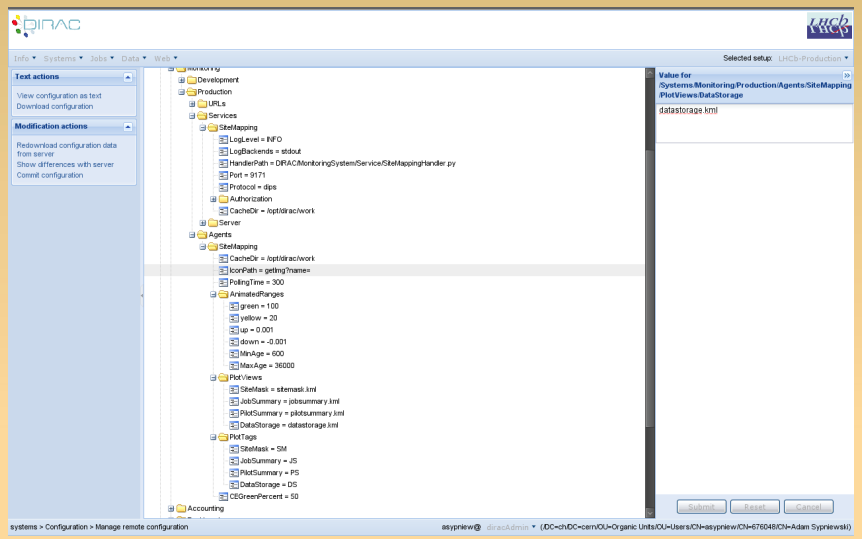
- Load configuration data
- Receive file request
- Find KML / image file
- Transfer data

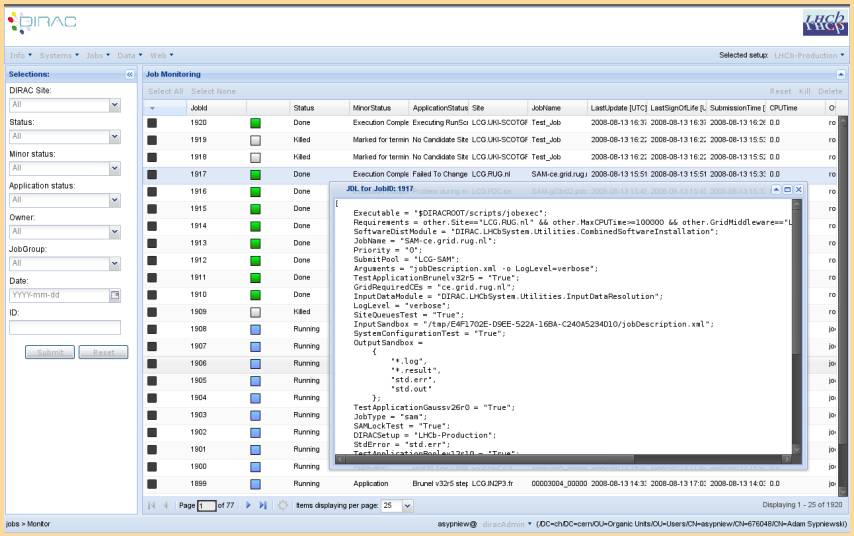


Inside the Map

The Actual Map

- OpenLayers framework
- Implemented in JavaScript
- Easy to use
- Integrates with existing framework



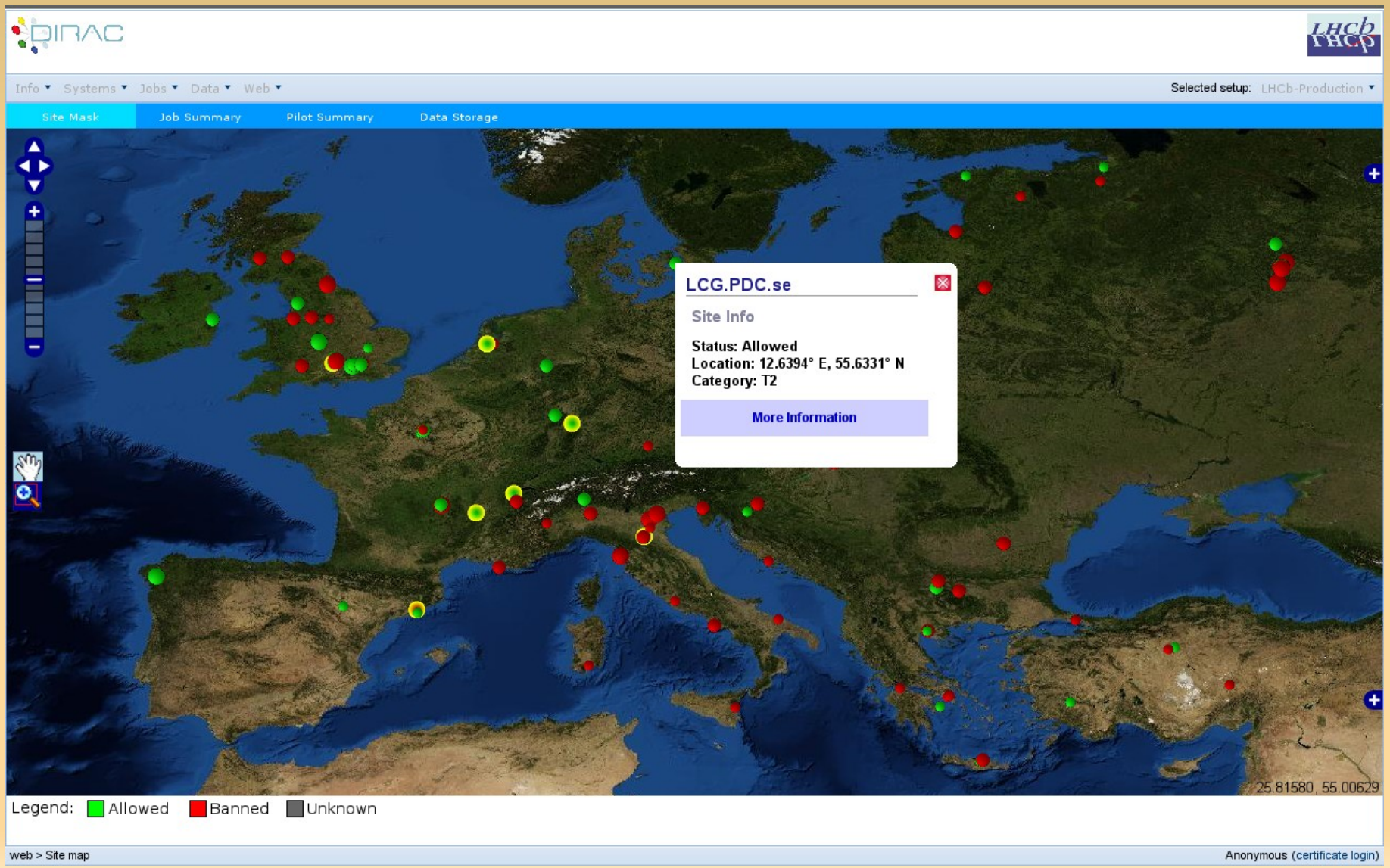


Pylons & AJAX

- Interface between map and service to retrieve KML & image data

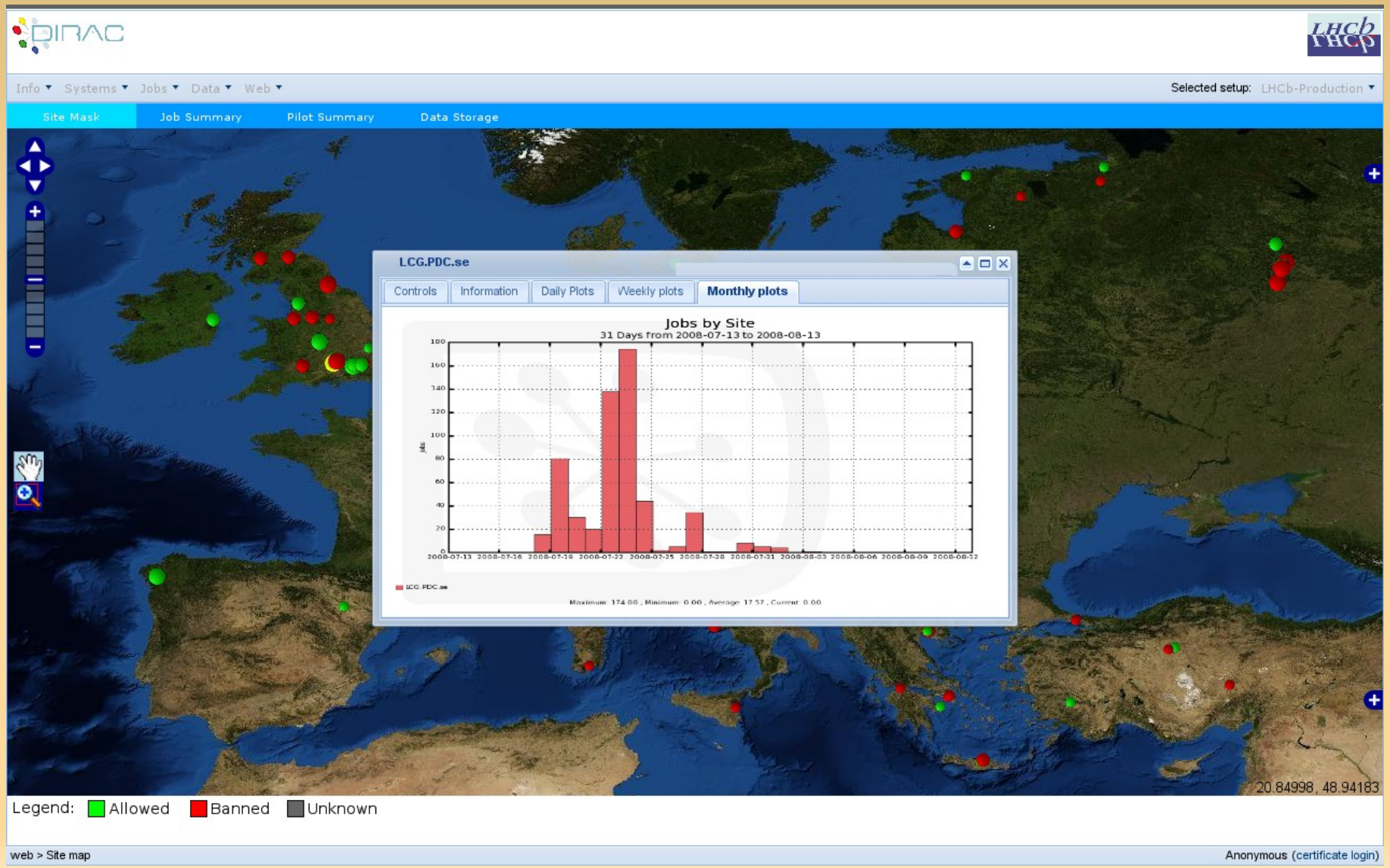
Prototype

View: Site Mask



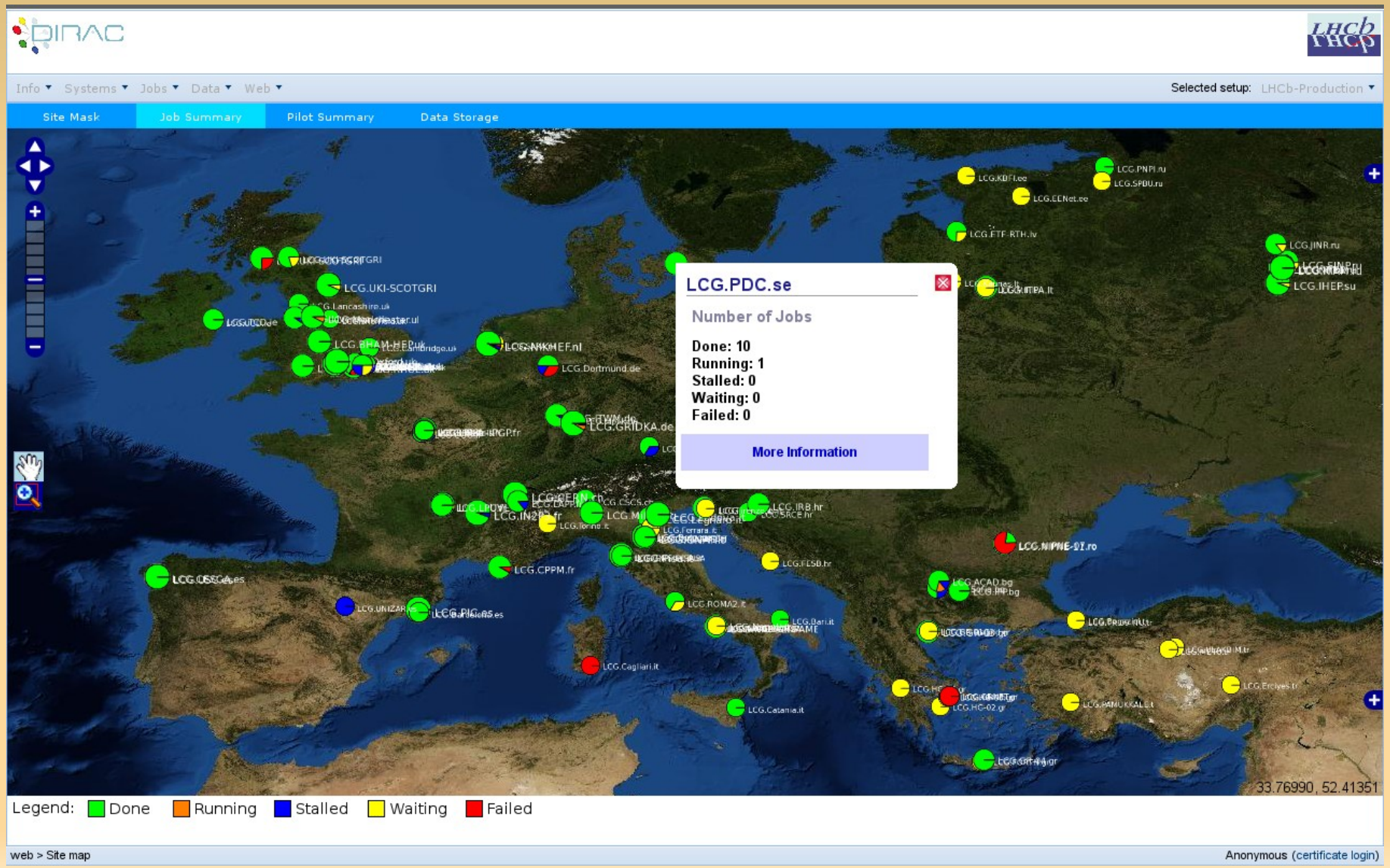
Prototype

View: Site Mask



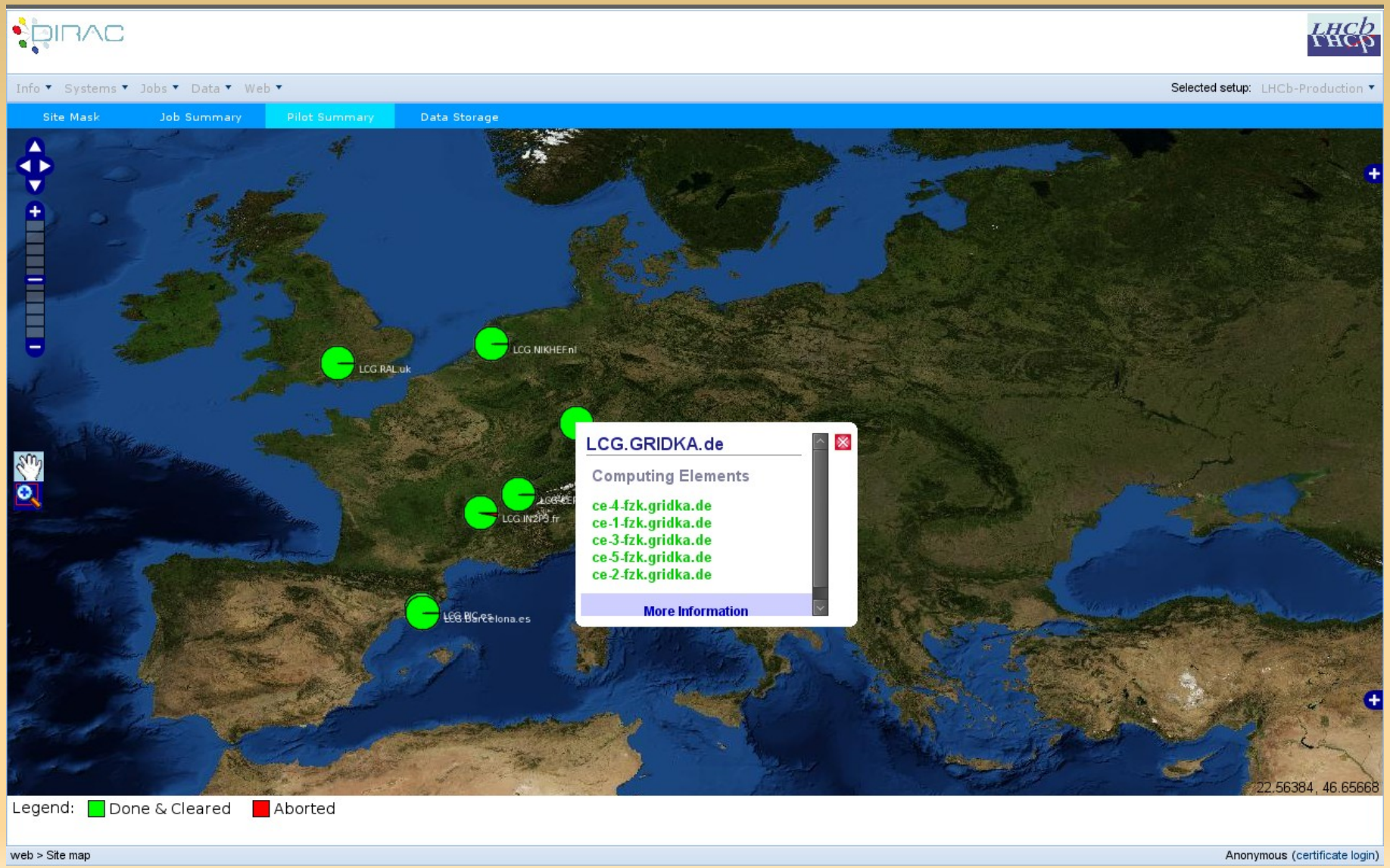
Prototype

View: Job Summary



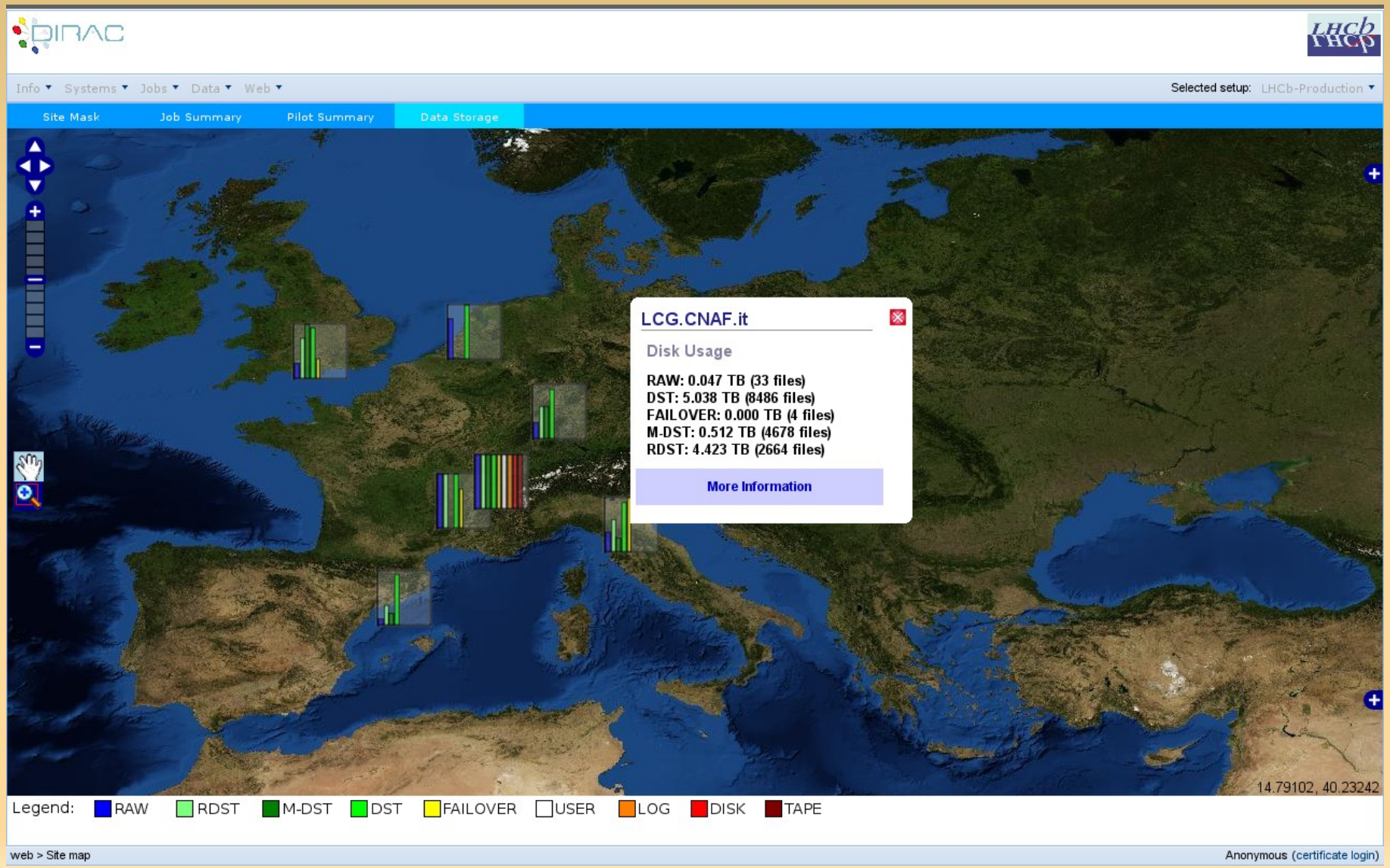
Prototype

View: Pilot Summary



Prototype

View: Data Storage



In Conclusion

Comments

- Learned about Grid computer
- Understand the intricacies of DIRAC
- Had fun with lots of languages
- Created a useful tool for shifters and site administrators

Special Thanks To:

- Stuart Paterson, Andrew Smith
- Adrian Casajus, Matvey Sapunov, Andrei Tsaregorodtsev
- Homer Neal, Jean Krisch, Myron Campbell, Jeremy Herr
- Ingrid Schmid, Virginie Blondeau, Catherine Nederman

Thanks for listening!