An EISCAT 3D frontend using DIRAC engine

EISCAT 3D CC & DIRAC Technical Discussion

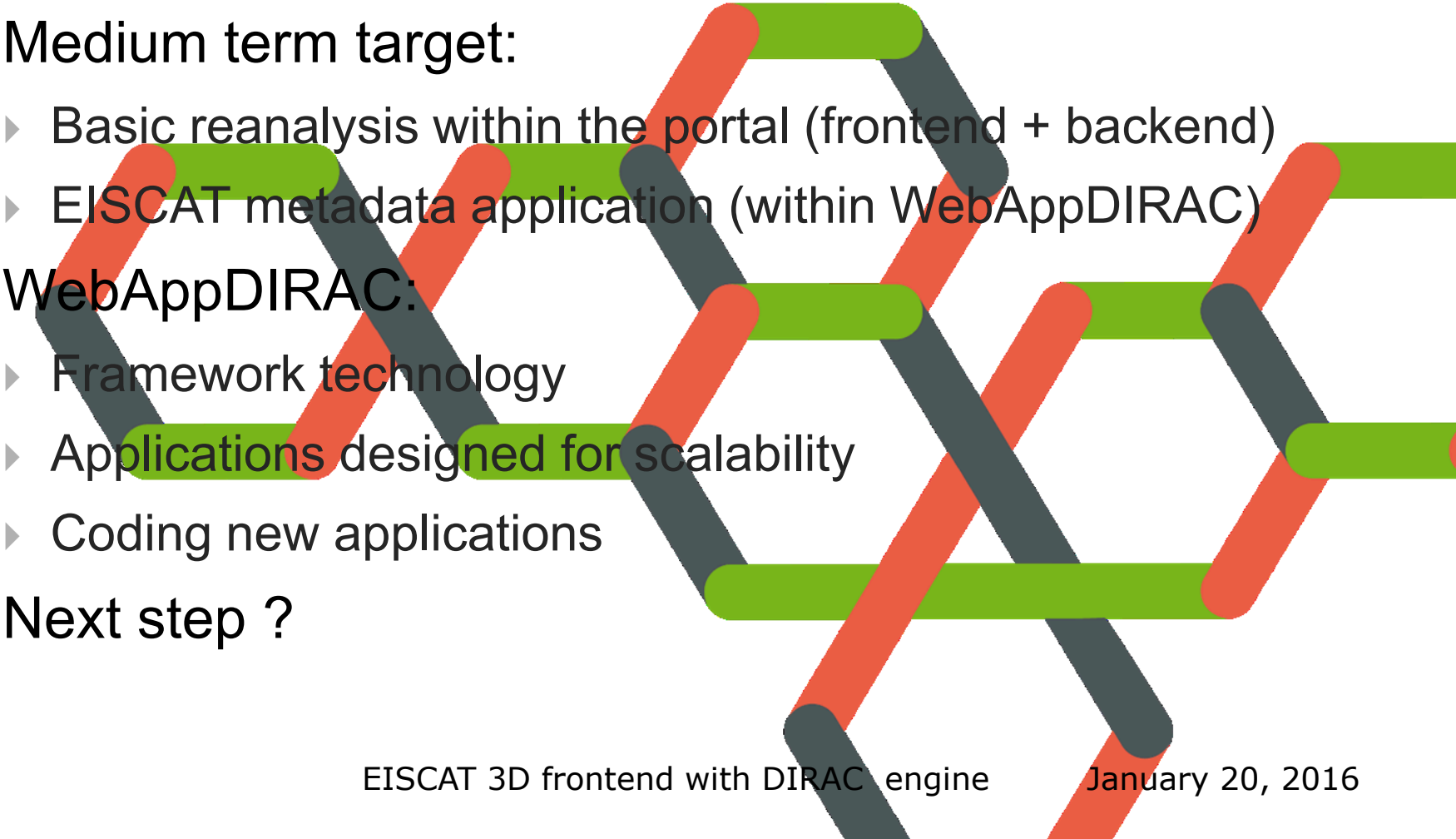
January 20, 2016

Víctor Méndez

University Autònoma of Barcelona (UAB)

Agenda

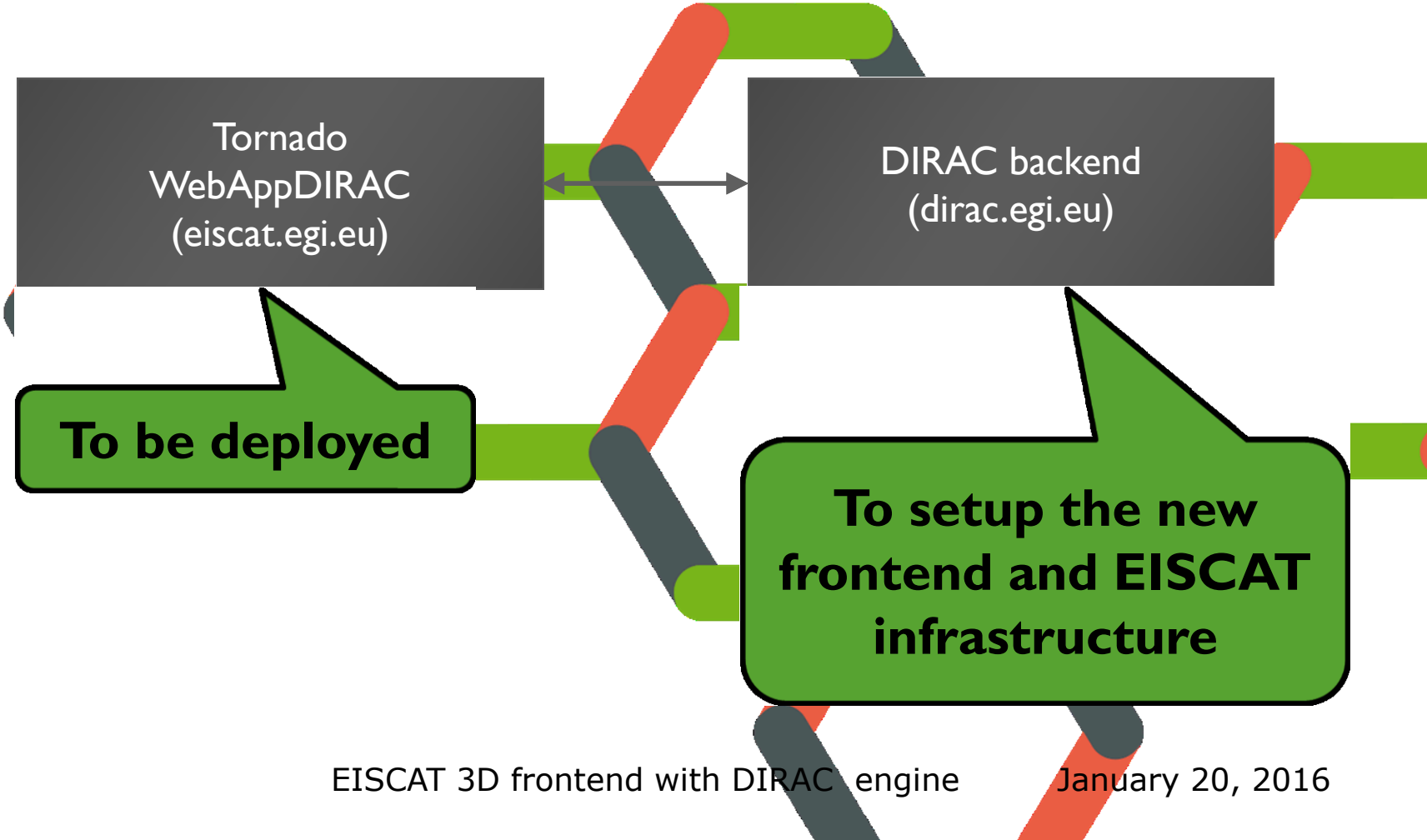
- Short term target: Deploy a prototype portal as a production system
- Medium term target:
  - Basic reanalysis within the portal (frontend + backend)
  - EISCAT metadata application (within WebAppDIRAC)
- WebAppDIRAC:
  - Framework technology
  - Applications designed for scalability
  - Coding new applications
- Next step ?



Short term target: Deploy the portal as a production system

- DIRAC engine has a robust technology to provide a prototype of an EISCAT frontend:
  - DIRAC as a service is a robust backend enabling distributed computing in a transparent and interoperable manner to the end user
  - dirac.egi.eu service is providing user and technical support within EGI Engage project
  - WebAppDIRAC is a Web framework integrated in DIRAC engine with good enough set of functionalities to deploy a prototype of EISCAT 3D frontend,
  - WebAppDIRAC can be used to develop a production EISCAT 3D frontend
    - Re-using much of the code
    - Taking advantage of robust and tested code

Short term target: Deploy the portal as a production system



Short term target: Deploy the portal as a production system

- WebAppDIRAC provides a set of applications which can be deployed

Which of them would you like to deploy in eiscat.egi.eu prototype ?

Short term target: Deploy the portal as a production system

- ✔ Job monitor
- ✔ Pilot monitor
- ✔ Accounting
- ✔ Configuration manager
- ✔ File Catalog
- ✔ System administration
- ✔ Proxy Upload
- ✔ Job Launchpad
- ✔ Registry/Proxy Manager
- Resource Manager
- ✔ Transformation Manager
- ✔ Virtual Machine Monitor
- ✔ System Monitor
- ✔ Request Monitor
- ✚ LHCb specific ones
- ✚ CTA specific ones
- ✚ Belle II specific ones
- ✚ others

Short term target: Deploy the portal as a production system

- What it is needed to deploy eiscat.egi.eu prototype using WebAppDIRAC technology ?
  - A host in production basis to deploy EISCAT frontend
  - Allocate effort to deploy EISCAT frontend prototype
- dirac.egi.eu backend is supported by the current assigned effort to the UAB
  - eiscat.egi.eu portal integration, setup and test
  - Setup and operations in the top of EISCAT infrastructure

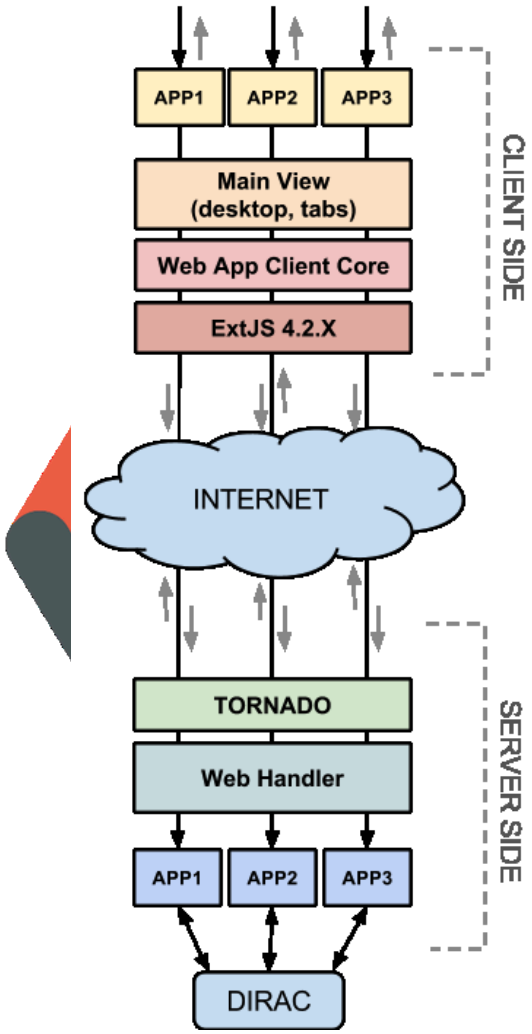
Medium term target: Basic reanalysis within the portal

- We can do basic reanalysis campaigns using the prototype portal ….
  - Even when this first portal will not be completely friendly, just a *prototype*
  - This would be a valuable experience to EISCAT experts in order to get some know-how in DIRAC capabilities and distributed computing productions (daily operations)
  - This would be a valuable experience to DIRAC experts in order to dig in EISCAT specifications details to support future production system

Medium term target: EISCAT metadata application

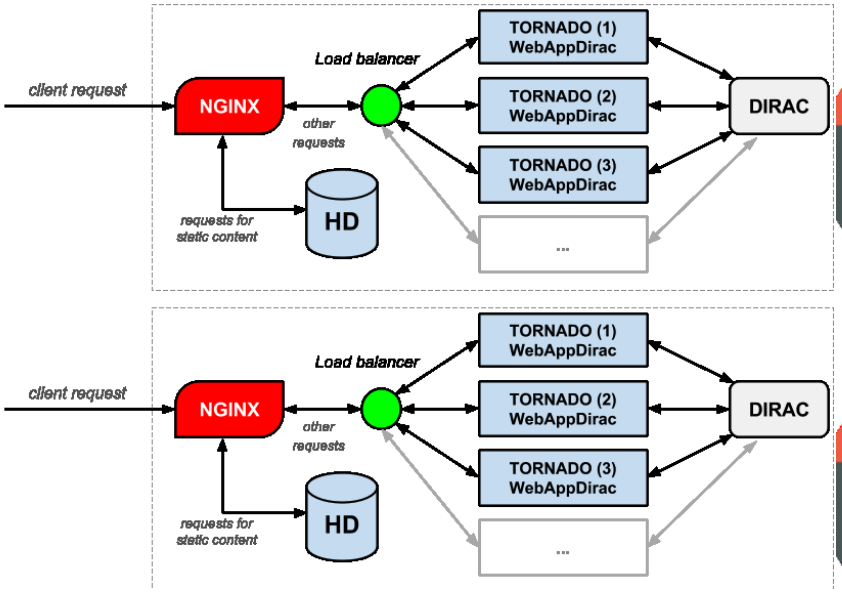
- A data as metada application for EISCAT to be designed with WebAppDIRAC framework
  - Using existing WebAppDIRAC technology ensuring:
    - Scalability
    - Tested functionality integration in DIRAC framework
    - A friendly desktop like web environment
      - Including user profile management (share desktop setup, load, save…)
    - A tested widget toolbox
  - Re-using existing FileCatalog application

WebAppDIRAC: framework technology



- Client side (browser):
  - Javascript
    - Heavy use of ExtJS 4.2.X
  - HTML
  - CSS
  - Google Charts
- Server Side (web server)
  - Python
    - Tornado
    - DIRAC client tools

WebAppDIRAC: applications designed for scalability



- Tornado
  - well tested framework for web servers (**scale UP**)
  - Feedback:
    - asynch. Server methods
  - Multi-process (**scale UP**)
- NGINX
  - Serves JavaScript (caching)
  - Redirects to Tornado
    - Load balancing for the same installation (**scale UP**)
    - Can connect to different “tornados” for different URLs (**scale DOWN**)
- State less architecture:
  - DNS load balancing (**scale UP**)

WebAppDIRAC: Coding new Applications

- Developer must be familiar with underlying technologies (python, javascript, ExtJS)
- Developer must have working experience with concurrent programming (AJAX, web sockets)
- Server side is “similar” to other DIRAC services, but it has “synchronous” and “asynchronous” methods. No direct access to DBs, must use service interface.
- Client side is well encapsulated, well defined interface to the framework:
  - Load (to instantiate and “load” a given state)
  - Save (returning the current state, to be saved)
  - Framework makes info available via a Global variable
- Each application can define its own internal layout
- https://github.com/DIRACGrid/WebAppDIRAC/wiki
- http://pos.sissa.it/archive/conferences/210/042/ISGC2014_042.pdf

Next step ?

- Happy to support you
- Happy to foster eiscat.egi.eu with DIRAC engine
- Roadmap should be re-designed with DIRAC engine
- Efforts should be re-allocated