



# CERN DB On Demand

A DBaaS platform

ignacio.coterillo@cern.ch

7 Nov 2017

# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

### Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

### Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer

# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

### Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

### Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer

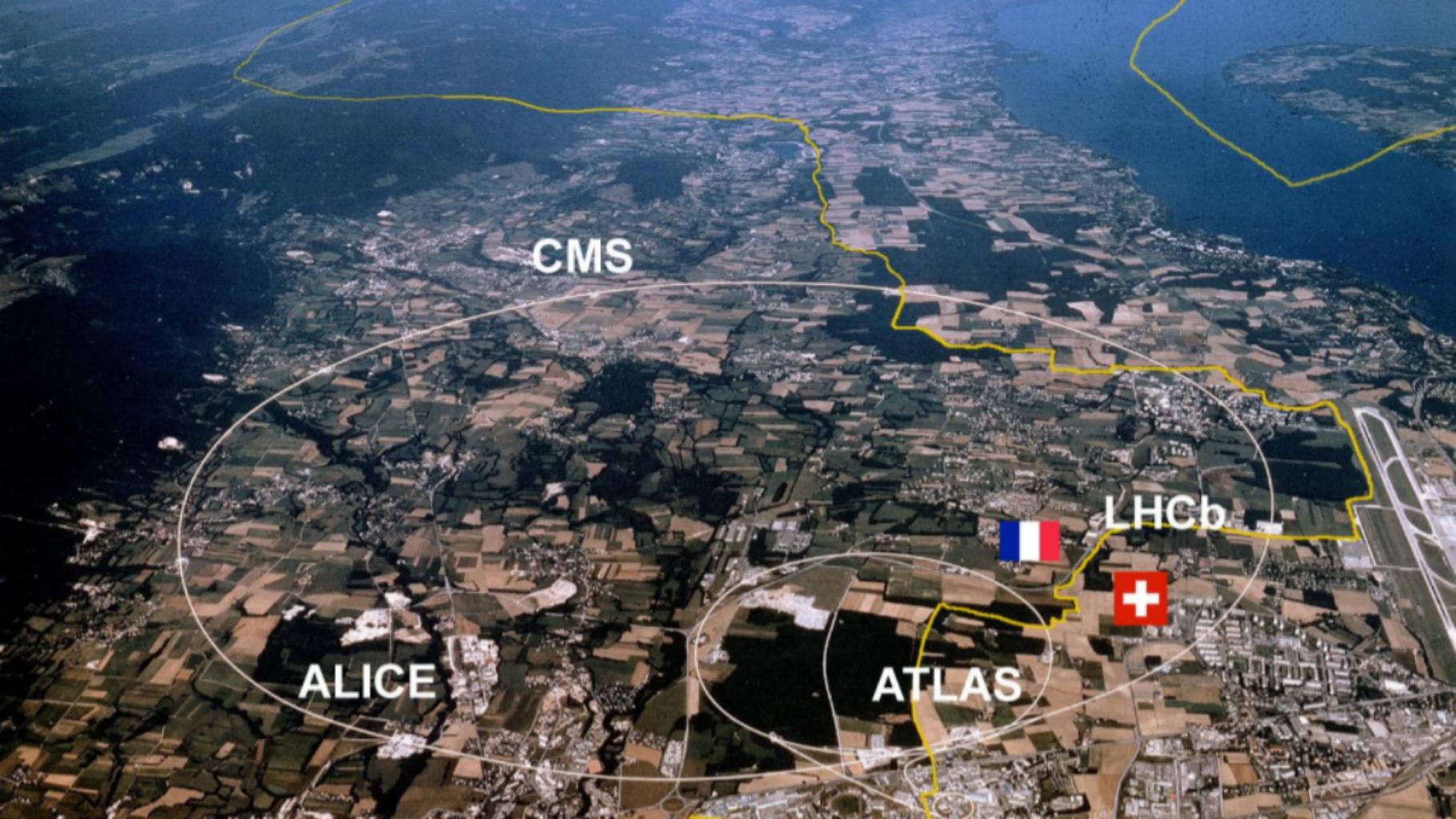


On the French-Swiss border  
22 member states  
Collaborators from all around the world  
What do we do?





# Large Hadron Collider (LHC)



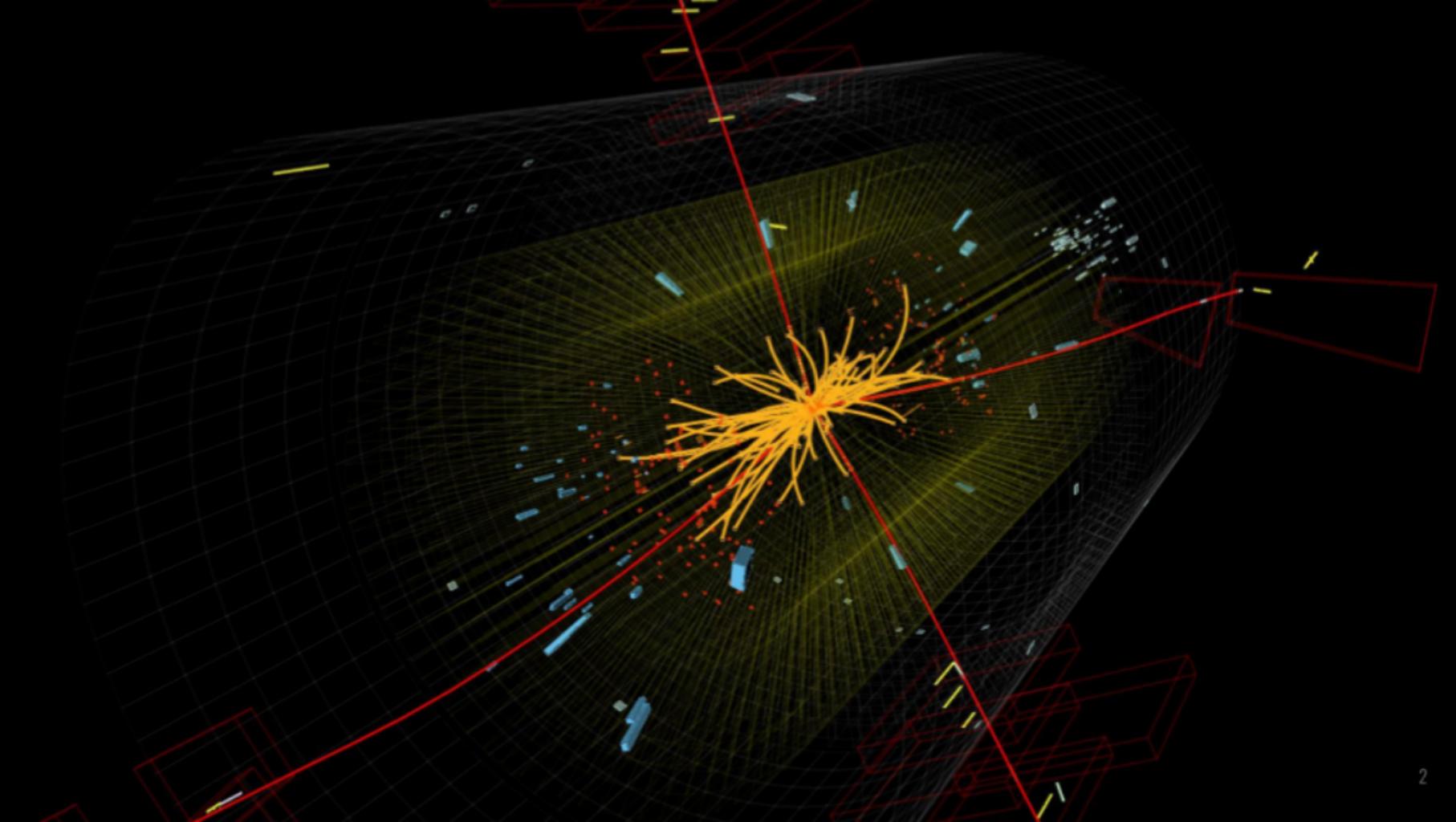
CMS

ALICE

ATLAS

LHCb









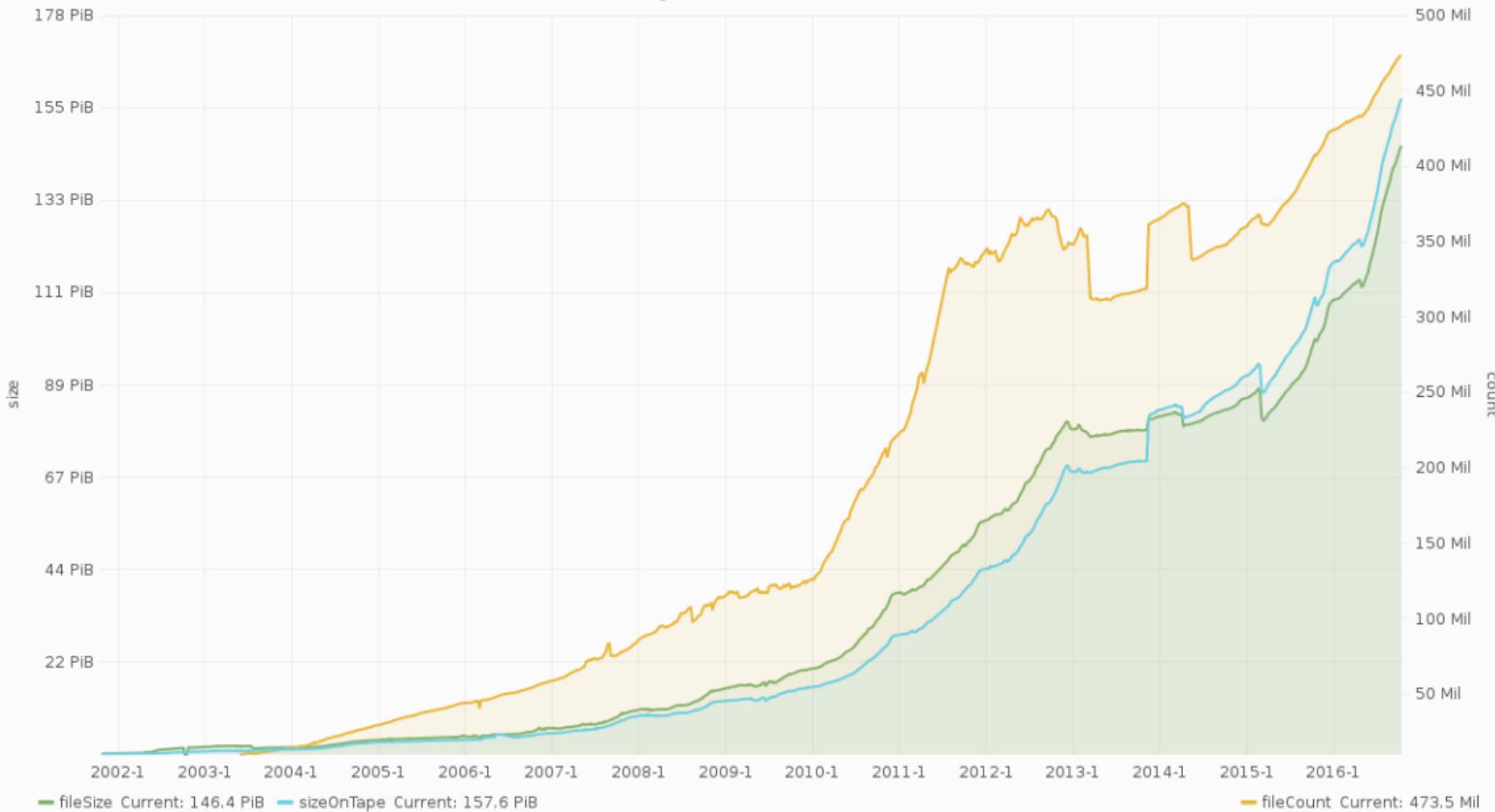
**STORAGETEK**

06  
1+

06  
10

10

## Physics Data in CASTOR



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

### Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

### Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# Databases Group

- ▶ Part of the IT Department
- ▶ Around ~ 30 people
- ▶ Manages database services for the whole CERN community
  - ▶ Experiments (CMS, Atlas, LHCb, Alice, ...)
  - ▶ Accelerator databases
  - ▶ Access control, Human Resources, Purchases, etc
  - ▶ IT tools (Puppet, OpenStack, ...)
- ▶ Manage also application servers



## Oracle at CERN: Key facts

- ▶ ~ 70 databases (most of them RAC) (plus ~ 30 replicas)
- ▶ Running versions 11g (11.2.x) and 12c (12.1.x)
- ▶ Over 700 TBi of data files for *production* databases



## Key facts

- ▶ 9 Oracle special cases (being decommissioned)
- ▶ 71 InfluxDB (time Series)
- ▶ 123 PostgreSQL
- ▶ 386 MySQL instances



# Other "databases"

## Hadoop

- ▶ 4 clusters (3 prod and 1 QA)

## ElasticSearch

- ▶ ~ 25 Clusters<sup>1</sup>

## MS SQL Server

- ▶ ~ 10s of databases for MS Applications<sup>2</sup>

---

<sup>1</sup>Managed by the IT Monitoring team

<sup>2</sup>Managed by the application owners



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

### Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# Service Timeline

- ▶ **2011 Q?** IT-DB hosts MySQL databases for the Drupal service
- ▶ **2011 Q4:** Started operations as a service intended for managing single instance MySQL databases
- ▶ **2012 Q2:** Added **Oracle** instances
- ▶ **2013 Q2:** Added **HA Cluster** instances
- ▶ **2013 Q4:** Added **PostgreSQL** instances
- ▶ ... Infrastructure migration (puppetization)
- ▶ **2015:** Introduced **replication** support, New platform architecture
- ▶ **2016:** New architecture, SSL, InfluxDB



# Overview

## Key facts

### User is DBA

Database owners have *almost full* DBA privileges<sup>3</sup>

### Multi DBMS support

MySQL, PostgreSQL, Oracle<sup>4</sup>, InfluxDB

### Common Backup and Recovery operations

- ▶ B&R based on storage level snapshots.
- ▶ Point in Time Recovery depending on each DB system capabilities

---

<sup>3</sup>Not allowed to modify host system critical parameters

<sup>4</sup>Phasing out



## Resource allocation

- ▶ High consolidation. Several databases hosted on the same servers/VM's.
- ▶ Each application/service gets its own separated database process/cgroup
- ▶ Each database uses two NFS volumes (data, binary logs)
- ▶ Standard initial configuration.



# Overview

## Functionality

- ▶ Start/Stop
- ▶ File management
  - ▶ Configuration files
  - ▶ Log files
- ▶ Backup: manual and automatic backups based on storage snapshots)
- ▶ Recovery: To an specific snapshot and PITR
- ▶ Upgrade: Enabled only when a certain version is made available
- ▶ Database and host monitoring



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

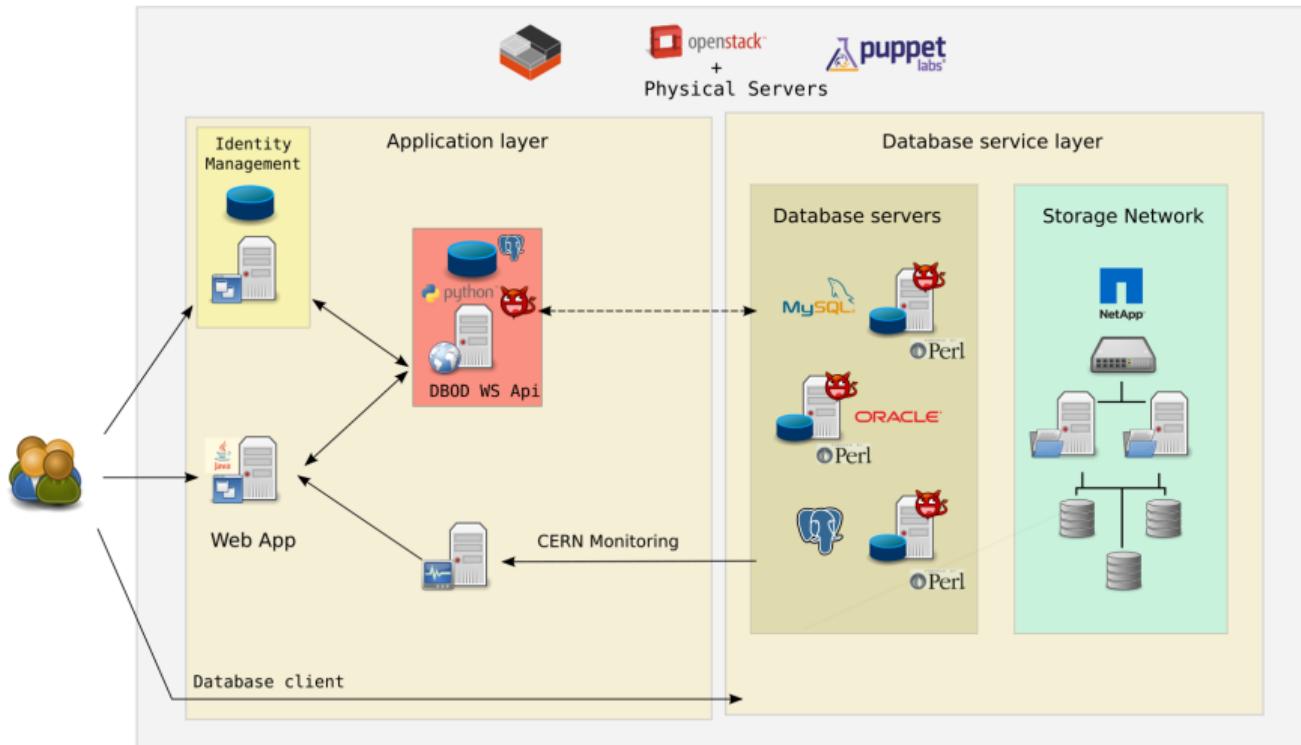
URIs

API responses

Data Abstraction Layer



# Simplified architecture



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# Web Interface

The screenshot shows a web-based management interface for database instances. The top navigation bar includes tabs for 'DB On Demand' and 'ng2-admin/calendar'. The main header displays the CERN logo and the text 'Accelerating science'. The user is signed in as 'jocorder (CERN)'.

**INSTANCES OVERVIEW**

Search...  
DB Name Username Host Category DB Type Version State

■	DB Name	Username	Host	Category	DB Type	Version	State
■	node02	user05	host04	DEV	ZOOKEEPER	3.4.9	RUNNING
■	node01	user05	host04	DEV	ZOOKEEPER	3.4.9	RUNNING
■	dbod06	user04	host01	TEST	MYSQL	5.6.17	RUNNING
■	dbod05	user04	host01	TEST	MYSQL	5.6.17	RUNNING
■	dbod03	user02	host01	TEST	MYSQL	5.5	RUNNING
■	dbod01	user01	host01	TEST	MYSQL	5.6.17	RUNNING
■	dbod04	user03	host01	PROD	PG	9.4.5	RUNNING
■	dbod02	user01	host03	PROD	PG	9.4.4	RUNNING

**JOB OVERVIEW**

Search...  
State Command DB Name DB Type DB Category Creation Completion

State	Command	DB Name	DB Type	DB Category	Creation	Completion



7 Nov 2017

26

# Web Interface

The screenshot shows a web-based management interface for a database system. At the top, there are several tabs: 'DB On Demand', 'DB On Demand', 'ng2-admin/calendar.ser...', and 'ng2-admin/calendar.com...'. The main window title is 'jose-desktop.cern.ch:3000/#/pages/home'.

The header includes the CERN logo and the tagline 'Accelerating science'. It also shows the user is signed in as 'jocorder (CERN)' with options to 'Sign out' and 'Directory'.

The left sidebar has three items: 'Home' (selected), 'Help', and 'External Links'.

The central content area displays 'INFORMATION FOR INSTANCE DB0D01' with the following details:

State:	RUNNING	DB Type:	MYSQL	Username:	user01
Project:	API	Category:	TEST	Email:	alice@cern.ch
e-Group:	testgroupA	Version:	5.6.17	Full name:	Alice Lastname
Creation Date:	2017-09-08	Host:	host01	Org. Unit:	ITC-DBC-EEC
Expiry Date:		Port:	port		
Description:	Test instance 1				

Below this, there is a navigation bar with tabs: 'Jobs' (selected), 'Backup', 'Recovery', 'File Editor', and 'Metadata Editor'. A search bar labeled 'Search...' is present.

The 'Jobs' section contains a table of completed tasks:

State	Command	Creation	Completion
FINISHED_FAIL	CLEANUP	2017-08-01	2017-08-01
PENDING	BACKUP	2017-08-02	2017-08-02
FINISHED_OK	CLEANUP	2017-08-03	2017-08-03
FINISHED_OK	BACKUP	2017-08-04	2017-08-04
FINISHED_OK	CLEANUP	2017-08-05	2017-08-05



7 Nov 2017

27

# Web Interface

The screenshot shows a web-based management interface for a database system. On the left, a sidebar menu includes Home, Help, and External Links. The main content area displays a configuration card for a database instance named 'testgroupA'. The card includes fields for e-Group (testgroupA), Version (5.6.17), Full name (Alice Lastname), Creation Date (2017-09-08), Host (host01), Org. Unit (ITC-DBC-EEC), Expiry Date, Port (port), and Description (Test instance 1). Below this is a navigation bar with tabs: Jobs, Backup, Recovery (selected), File Editor, and Metadata Editor. The central part of the page is a calendar for September 2017. Specific dates are highlighted in green, indicating scheduled snapshots. For example, on Wednesday, September 6, there are two scheduled snapshots: '10:30a Snapshot #1' and '6:30p Snapshot #2'. On Friday, September 1, there are three scheduled snapshots: '9a Snapshot #1', '12p Snapshot #2', and '5p Snapshot #3'. Other dates with scheduled snapshots include September 3, 4, 5, 7, 13, 14, and 19.



7 Nov 2017

28

# Web Interface

The screenshot shows a web-based management interface for a database system. The top navigation bar includes tabs for "DB On Demand", "ng2-admin/calendar.sen", "ng2-admin/calendar.com", and "File Editor". The user is signed in as "jocorder (CERN)". The main menu on the left has options for "Home", "Help", and "External Links". The central content area features a "File Editor" tab selected, displaying a configuration file named "my.cnf". The file contains the following MySQL configuration parameters:

```
[mysqld]
max_user_connections = 300
max_heap_table_size = 32M
server_id = 1
general-log-file = /DRIVE/DATABASE/mysql/mysql.log
max_connections = 1000
performance_schema
innodb_flush_method = O_DIRECT
innodb_read_io_threads = 4
innodb_flush_log_at_trx_commit = 1
log_slave_updates
binlog_format = MIXED
port = 5500
socket = /var/lib/mysql/mysql.sock.5500
tmp_table_size = 32M
innodb_io_capacity = 200
sync_binlog = 1
query_cache_size = 128M
expire_logs_days = 32
innodb_write_io_threads = 4
slow_query_log = 1
thread_cache_size = 50
innodb_open_files = 500
table_definition_cache = 1000
table_open_cache = 1000
key_buffer_size = 256M
innodb_file_per_table
log_output = FILE
datadir = /DRIVE/DATABASE/mysql
```

A "Submit" button is located in the top right corner of the editor area.



7 Nov 2017

29

# Web Interface

The screenshot shows a web-based management interface for a database system. The top navigation bar includes tabs for "DB On Demand", "DB On Demand", "ng2-admin/calendarsen", "ng2-admin/calendar.com", and a local file. The URL in the address bar is "jose-desktop.cern.ch:3000/#/pages/home". The header also displays the CERN logo and a sign-in message: "Signed in as: jocorder (CERN) Sign out Directory".

The main content area has a left sidebar with "Home" selected, "Help", and "External Links". Below the sidebar, there are tabs for "Jobs", "Backup", "Recovery", "File Editor", and "Metadata Editor", with "Metadata Editor" currently active.

In the "Metadata Editor" section, there is a form with fields for "Expiry Date", "Port", and "port". The "Description" field contains "Test instance 1".

The bottom half of the screen displays a "Tree" view of the database instance's properties. The tree structure is as follows:

- object {24}
  - status : ACTIVE
  - creation\_date : 2017-09-08
  - cluster\_id : null
  - port : port
  - egroup : testgroupA
  - basedir : /usr/local/mysql/mysql-5.6.17
  - datadir : /ORA/dbs03/D80001/mysql
  - id : 1
  - state : RUNNING
  - version : 5.6.17
  - logdir : /ORA/dbs02/DB0D01/mysql
  - type : MYSQL
  - bindir : /usr/local/mysql/mysql-5.6.17/bin
  - username : user01
  - description : Test instance 1
  - db\_name : dbod01



7 Nov 2017

30

# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

## Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

## API

Elements

URIs

API responses

Data Abstraction Layer



# What is apiato?

## apiato

A REST API sitting in the middle of the DB On Demand and Nile platforms<sup>5</sup> working as both a single entry point for service data, and as a gateway to support services such as Rundeck, FIM, *OpenStack*, or the *Storage API*)

## HTTP spoken

External APIs integration and the system database all accessed through HTTP(s)

---

<sup>5</sup>A similar platform offering Streaming solutions on demand (Apache Kafka)



# API server

## Facts

- ▶ Based on Tornado<sup>6</sup> a Python web framework and asynchronous networking library
- ▶ Hosted on GitHub<sup>7</sup>. Open sourced under the **GPLv3** License
- ▶ Interaction with external API's, authentication layer, ...
- ▶ 92% of code covered by tests
- ▶ Available for installation via **pip (Python Package Index)**

---

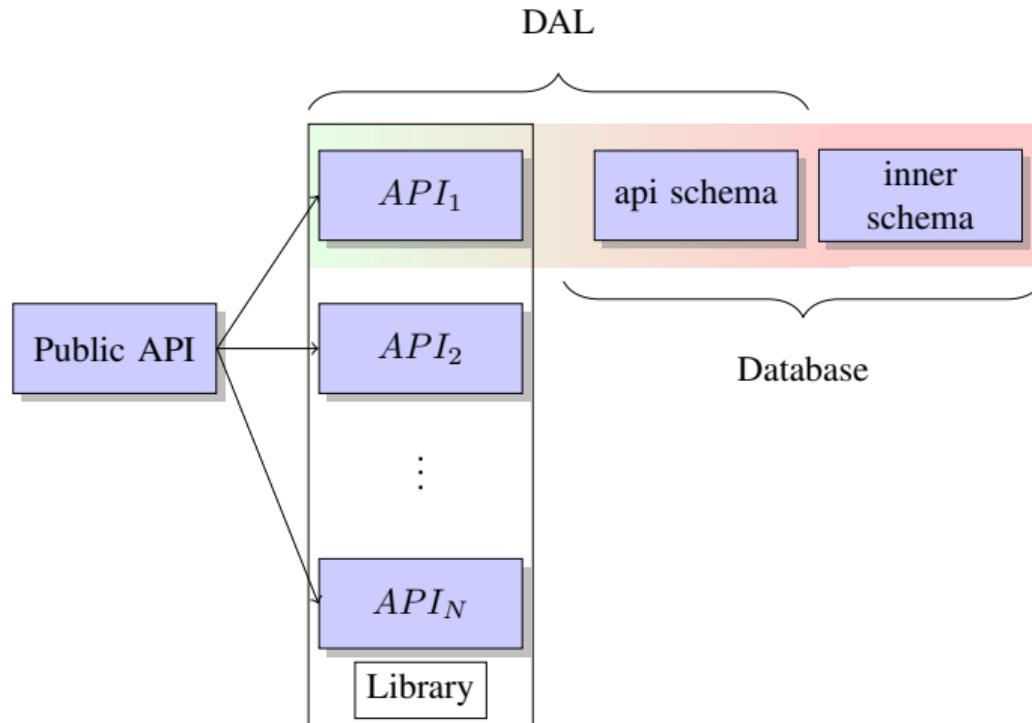
<sup>6</sup><https://tornadoweb.org>

<sup>7</sup><https://github.com/cerndb/dbod-api>



# What is apiato?

## Elements



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# Rundeck, an Automation/Orchestration tool

The screenshot shows the Rundeck web interface. At the top is a dark navigation bar with the Rundeck logo, user name 'dbod-admin', and links for 'Jobs', 'Nodes', 'Commands', 'Activity', and 'Project'. Below the bar is a search header with 'Browse' dropdown, a search input field containing 'Enter a node filter, or .\* for all nodes', and a 'Search' button. To the right of the search is a help icon and a 'FILTERS' dropdown set to 'All Nodes 624'. On the left, under 'TAGS', is a list of categories with counts: DEV 2, InfluxDB 65, MYSQL 380, ORA 1, ORACLE 9, PG 111, PROD 390, REE 9, and TEST 165.

**TAGS**

- [DEV](#) 2
- [InfluxDB](#) 65
- [MYSQL](#) 380
- [ORA](#) 1
- [ORACLE](#) 9
- [PG](#) 111
- [PROD](#) 390
- [REE](#) 9
- [TEST](#) 165

**FILTERS**

All Nodes 624 ▾



RUNDECK dbod-admin Jobs Nodes Commands Activity Project ▾

Jobs (25) Filter ▶ Expand All Collapse All

▼ Instance specific

puppetdb-bloat-monitor ▾ Monitors puppetdb bloat in 52m on 9f



## create-instance Action

Prepare and Run...

Definition

**type** InfluxDB

Run Job Now ▶

Run Job Later ⏲

Follow execution

**entity**



Instance name as requested in FIM.

No prefixes



**host**

Hostname (without .cern.ch)



**owner**

Owner's Username

**size** 10

Initial data volume size in GB.

**class** PROD



**version**



## Apiato

- ▶ Resource file generation
- ▶ Gateway to Rundeck API for Job execution and scheduling

## e.g: Rundeck resources.xml

```
<project>
<node name="ac3a_db" description="" hostname="dbod-ac3a-db" username="dbod" type="PROD"
subcategory="PG" port="6605" tags="PG,PROD"/>
<node name="acc_test" description="" hostname="dbod-acc-test.cern.ch" username="dbod"
type="TEST" subcategory="MYSQL" port="5506" tags="MYSQL,TEST"/>
```



# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# Puppet Module

## Custom fact

```
Facter.add(:dbod_instances) do
setcode do
begin
# API url and base endpoint to host metadata
base_url = 'https://dbod-api.cern.ch:5443/api/v1/host'
uri = URI([base_url,
Socket.gethostname.split('.')[0],
'metadata', ].join('/'))
conn = Net::HTTP.new(uri.host, uri.port)
conn.use_ssl = true
conn.ca_file = '/etc/ssl/certs/CERN-bundle.pem'
conn.read_timeout = 5
conn.open_timeout = 5
conn.verify_mode = OpenSSL::SSL::VERIFY_PEER
response = conn.start { |http|
request = Net::HTTP::Get.new(uri.path)
http.request request
}
case response
when Net::HTTPNotFound
# No instances defined for host
{}
else
format( (JSON.parse response.body)['response'] )
end
rescue SocketError, Timeout::Error, Errno::ECONNREFUSED
# Return cached result
if File.exist? '/etc/dbod/cache/entities.json'
format(
JSON.parse File.read('/etc/dbod/cache/entities.json')
)
end
end
end
end
end
end
```



# Puppet Module

## In hostgroup manifest

```
# Operate on dbod_instances fact
if (is_hash($::dbod_instances)) {
$defaults = {
require => [
Class['::dbod::certificates'],
Class['::dbod::users'],
]
}
create_resources(dbod::instance, $::dbod_instances, $defaults)
}
```



# Monitoring

## Metrics

- ▶ Database metrics are collected using **Telegraf** and sent to an **InfluxDB**<sup>8</sup> instance
- ▶ Visualization dashboards are implemented with **Grafana**<sup>9</sup>, integrating database, system and storage system metrics

## Server Logs and Slow Query logs

- ▶ Both server logs and slow query logs are sent to an **ElasticSearch**<sup>10</sup> Cluster
- ▶ Visualization is being integrated in the platform Web Interface

---

<sup>8</sup><https://www.influxdata.com/>

<sup>9</sup><https://grafana.com/>

<sup>10</sup><https://www.elastic.co/>





[Dashboard](#)[Timelion](#)[Indices](#)[Own Home](#)[Management](#)

? \_source

### Available Fields



#### Popular

- ⌚ @timestamp
- ⌚ httpd\_timestamp
- ⌚ msg\_timestamp
- ⌚ tags

- ⌚ @version
- ⌚ \_id
- ⌚ \_index
- # \_score
- ⌚ \_type
- ⌚ beat.hostname
- ⌚ beat.name
- ? beat.version
- ⌚ host
- ⌚ httpd\_ip
- ⌚ httpd\_req\_method
- ⌚ influx\_endpoint
- ⌚ influx\_msg
- ⌚ influx\_process
- ⌚ influx\_response
- ⌚ influx\_user
- ⌚ input\_type
- ⌚ log\_source
- ⌚ logstash\_source
- ⌚ message
- # offset

Time ▾

\_source

▶ September 12th 2017, 15:08:06.000	<code>offset: 37,768,755 logstash_source: dbod-logstash-2.cern.ch input_type: log source: /ORA/dbs03/HLTMON/influxd.log message: [I] 2017-09-12T13:08:06Z SELECT uptime / 3600 FROM plancton_hltdev.autogen.container WHERE started = 'True' AND time &gt; now() - 12h AND uptime &gt; 120 service=query type: inflog tags: beats_input_codec_plain_applied, matched @timestamp: September 12th 2017, 15:08:06.000 log_source: DBOD DB LOG MESSAGE influx_process: I @version: 1</code>
▶ September 12th 2017, 15:08:06.000	<code>website: http://alihtcloud2.cern.ch:3000/dashboard/db/plancton-monitor?from=now-12h&amp;to=now status_code: 200 offset: 37,769,284 logstash_source: dbod-logstash-2.cern.ch input_type: log httpd_timestamp: 12/Sep/2017:15:08:06 source: /ORA/dbs03/HLTMON/influxd.log message: [httpd] 128.141.224.178,137.138.116.75 - plancton_ro [12/Sep/2017:15:08:06 +0200] "GET /query?db=plancton hltdev&amp;epoch=ms&amp;q=SELECT+%22uptime%22FROM%22container%22+WHE</code>
▶ September 12th 2017, 15:08:06.000	<code>offset: 37,769,646 logstash_source: dbod-logstash-2.cern.ch input_type: log source: /ORA/dbs03/HLTMON/influxd.log message: [I] 2017-09-12T13:08:06Z SELECT mean(uptime) FROM plancton_hltdev.autogen.container WHERE started = 'True' AND time &gt; now() - 12h AND uptime &gt; 120 GROUP BY time(30s) fill(previous) service=query type: inflog tags: beats_input_codec_plain_applied, matched @timestamp: September 12th 2017, 15:08:06.000 log_source: DBOD DB LOG MESSAGE</code>
▶ September 12th 2017, 15:08:06.000	<code>offset: 37,769,812 logstash_source: dbod-logstash-2.cern.ch input_type: log source: /ORA/dbs03/HLTMON/influxd.log message: [I] 2017-09-12T13:08:06Z SELECT sum(last) FROM plancton_hltdev.autogen.sum_containers WHERE time &gt; 1505178306s AND time &lt; 1505221506s GROUP BY time(3m) service=query type: inflog tags: beats_input_codec_plain_applied, matched @timestamp: September 12th 2017, 15:08:06.000 log_source: DBOD DB LOG MESSAGE influx_process: I</code>
▶ September 12th 2017, 15:08:06.000	<code>offset: 37,769,968 logstash_source: dbod-logstash-2.cern.ch input_type: log source: /ORA/</code>

# Table of Contents

## Introduction

CERN

Databases at CERN

## DB On Demand

## Platform elements

Architecture

Web Interface

Apiato

Rundeck

Puppet

Monitoring

The dbod-core framework

Open Source

## Questions

## Extra

Oracle Databases

API

Elements

URIs

API responses

Data Abstraction Layer



# The dbod-core framework

## What is it?

The actual programs executed (typically) on the database hosts performing top level functionality tasks (backups, recoveries, upgrades, etc.)

## Why Perl

- ▶ Re-using pre-existing code
- ▶ NetApp SDK Perl distribution support superior to alternatives<sup>11</sup>

---

<sup>11</sup>At the time. Soon it will no longer be required



<https://github.com/cerndb>

This organization Search Pull requests Issues Gist ToDo

CERN Database Group Geneva, Switzerland

Repositories People 19 Teams 1 Settings

Filters Find a repository... New repository

**dbod-api** Python ★ 2 ⚡ 1  
DB On Demand API Updated a day ago

**dbod-core** Perl ★ 2 ⚡ 0  
DB On Demand managing infrastructure core library

People 19 >

CERN

7 Nov 2017

48

# Questions?



7 Nov 2017

49

<https://home.cern>

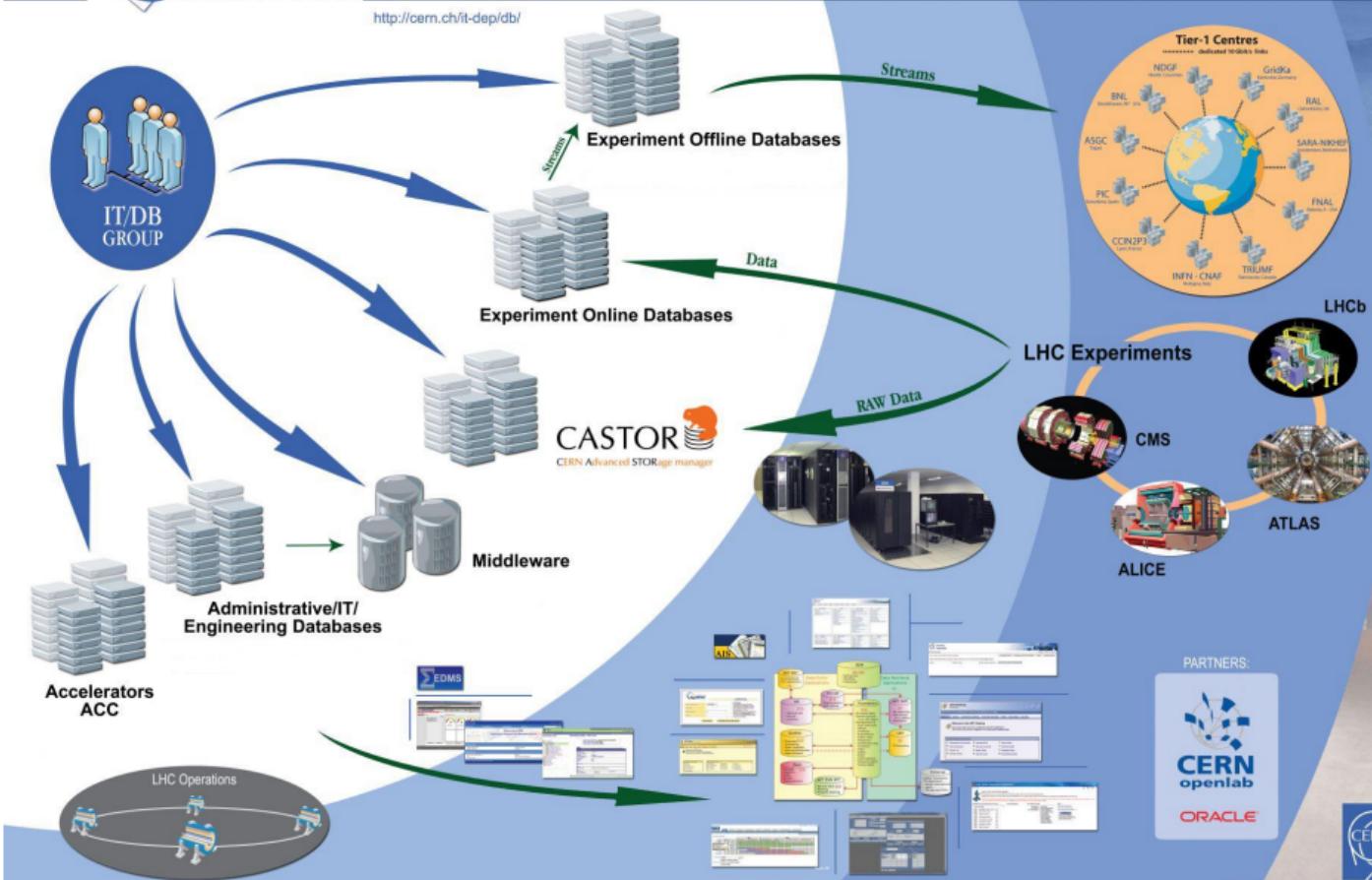
<https://visit.cern>





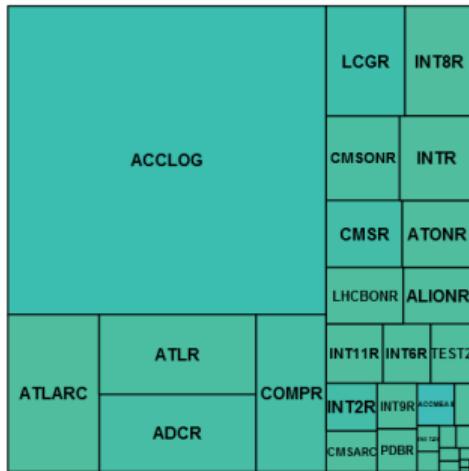
**[www.cern.ch](http://www.cern.ch)**

<http://cern.ch/it-dep/db/>



CERN databases October 2012

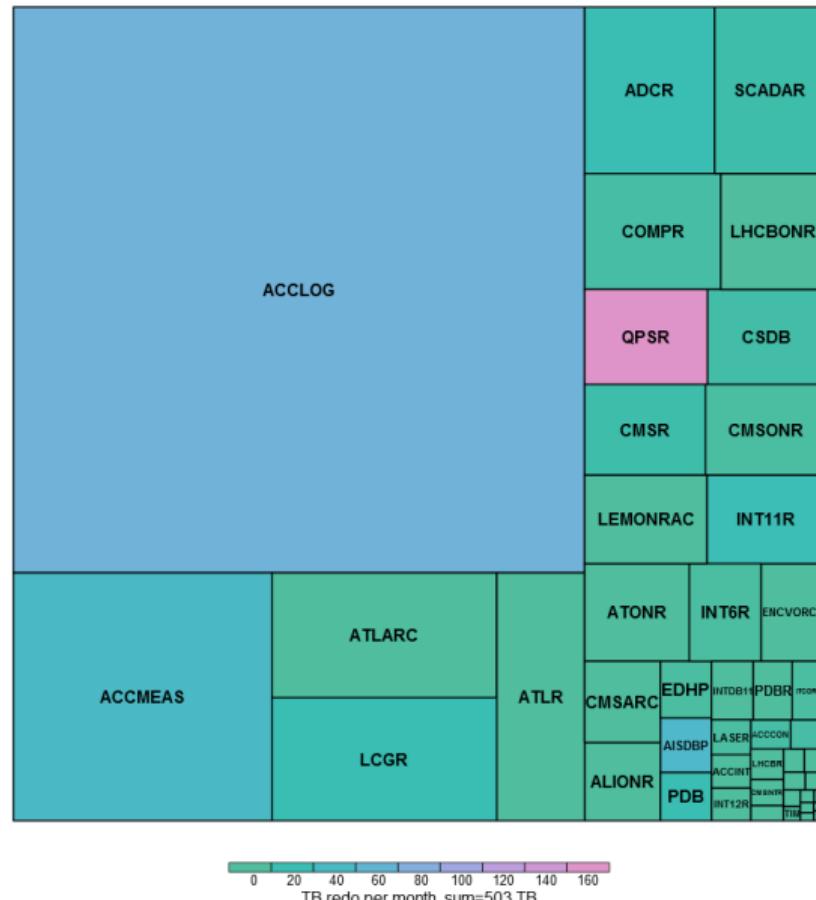
area is size (total=299TB)



TB redo per month, sum=120 TB

CERN databases May 2016

area is size (total=865TB), color is redo activity



TB redo per month, sum=503 TB

# What is apiato?

## Basic Concepts

### Instance

An individual element, generally linked to a process which run on a certain computing resource (host, vm, container), can be a member of a cluster, and can own resources.

### Cluster

A set of instances related in a certain way

### Resource

A *computing* element used by one or more instances (Storage volumes, *containers*, *virtual machines*, ...)



# API server

## Sample URIs

- ▶ `server:port/api/v1/instance/:name`
- ▶ `server:port/api/v1/instance/:name/metadata`
- ▶ `server:port/api/v1/host/:name`
- ▶ `server:port/api/v1/host/:name/metadata`
- ▶ `server:port/api/v1/cluster/:name`
- ▶ `server:port/api/v1/cluster/:name/metadata`
- ▶ `server:port/api/v1/rundeck/resources.xml`
- ▶ `server:port/api/v1/rundeck/job/:jobid/:instance`



# API server

## Nile cluster response

```
{  
    "response": [  
        {  
            "username": "certsoc",  
            "master_name": "zookeeper-sec-test",  
            "name": "kafka-sec-test",  
            "master_hosts": [  
                "nile-zk-sec-test-02.cern.ch",  
                "nile-zk-sec-test-01.cern.ch",  
                "nile-zk-sec-test-03.cern.ch"  
            ],  
            "version": "0.10.2.0",  
            "class": "DEV",  
            "project": "SECURITY TEAM Pilot",  
            "instances": [  
                {  
                    "username": "certsoc",  
                    "cluster_id": 5,  
                    "version": "kafka-0.10.1.1-scala-2.11",  
                    "class": "DEV",  
                    "port": null,  
                    "db_name": "kafka-sec-test-node-02",  
                    "hosts": [  
                        "nile-kafka-sec-test-02.cern.ch"  
                    ]  
                }  
            ]  
        },  
        {  
            "mounting_path": "/nbapps/sas/dev_kafka007"  
        }  
    ],  
    "attributes": {  
        "broker_id": "2"  
    },  
    "type": "KAFKA",  
    "id": 19  
},  
[],  
[],  
[],  
[],  
["lb_alias": null],  
["hosts": [  
    "nile-kafka-sec-test-02.cern.ch",  
    "nile-kafka-sec-test-04.cern.ch",  
    "nile-kafka-sec-test-03.cern.ch",  
    "nile-kafka-sec-test-01.cern.ch",  
    "nile-kafka-sec-test-05.cern.ch"  
]],  
["e_group": "it-db-nile-developers"],  
["attributes": {  
    "super.users": "User:kafka;User:aromerom;User:certsoc",  
    "log.retention.hours": "120",  
    "replication_factor": "3",  
    "transactional_log": "true"  
}],  
[],  
[],  
[],  
[]]
```



# API server

## DBOD instance metadata response (Oracle instance)

```
{  
  "active": true,  
  "attributes": {  
    "home": "/ORA/dbs01/oracle/product/rdbms1",  
    "listener_log": "/ORA/dbs01/oracle/product/rdbms1/network/admin",  
    "listner_config": "/ORA/dbs01/oracle/product/rdbms1/network/admin",  
    "listner_trace": "/ORA/dbs01/oracle/product/rdbms1/network/trace",  
    "log": "/ORA/dbs00/CSTEST1/bdump",  
    "notifications": "true",  
    "orainst": "/ORA/dbs01/oracle/product/rdbms1/oraInst.loc",  
    "port": "10121",  
    "rman_command": "/ORA/dbs01/syscontrol/projects/rman/cmd/rman_templates",  
    "tnsnames": "/ORA/dbs01/oracle/product/rdbms1/network/admin",  
    "tsm_server": "TSM615_ORA",  
    "version": "oracle-rdbms-1-11.2.0.4.0.cern5-1.x86_64"  
  },  
  "basedir": "/ORA/dbs01/oracle/product/rdbms",  
  "bindir": "/ORA/dbs01/oracle/product/rdbms",  
  "class": "TEST",  
  "datadir": "/ORA/dbs03/CSTEST1",  
  "db_name": "cctest1",  
  "db_type": "ORACLE",  
  "hosts": [  
    "itrac5175"  
  ],  
  "id": 160  
}
```



# Data abstraction layer

## External (*api*) database schema

- ▶ One<sup>12</sup> view per external API endpoint
- ▶ One update procedure per view

```
postgres@dbod-dbod01:dbod> \d
+-----+-----+-----+-----+
| Schema | Name      | Type   | Owner  |
+-----+-----+-----+-----+
| api    | attribute  | view   | dbod   |
| api    | fim_data   | view   | dbod   |
| api    | functional_aliases | view   | dbod   |
| api    | host       | view   | dbod   |
| api    | host_aliases | view   | dbod   |
| api    | instance   | view   | dbod   |
| api    | metadata   | view   | dbod   |
| api    | rundeck_instances | view   | dbod   |
| api    | volume     | view   | dbod   |
+-----+-----+-----+-----+
```

---

<sup>12</sup>At this moment



# Data abstraction layer

## postgrest

### What

- ▶ Standalone web server which turns a PostgreSQL database directly into a RESTful API<sup>13</sup>
- ▶ Single binary
- ▶ Running locally on the same node as the API servers

### Simple use

```
postgrest postgres://dbod@dbod-dbod01.cern.ch:6601/dbod -a dbod -s api
```

---

<sup>13</sup><https://postgrest.com/en/v0.4/intro.html>



# Data abstraction layer

curl localhost:3000 | jq

```
[  
  {  
    "schema": "api",  
    "name": "attribute",  
    "insertable": true  
  },  
  {  
    "schema": "api",  
    "name": "fim_data",  
    "insertable": true  
  },  
  {  
    "schema": "api",  
    "name": "functional_aliases",  
    "insertable": true  
  },  
  {  
    "schema": "api",  
    "name": "host",  
    "insertable": true  
  },  
  {  
    "schema": "api",  
    "name": "host_aliases",  
    "insertable": false  
  },  
  {  
    "schema": "api",  
    "name": "instance",  
    "insertable": true  
  },  
  {  
    "schema": "api",  
    "name": "metadata",  
    "insertable": false  
  },  
  {  
    "schema": "api",  
    "name": "rundeck_instances",  
    "insertable": false  
  },  
  {  
    "schema": "api",  
    "name": "volume",  
    "insertable": true  
  }]  
]
```

