

# Central IT-CMDB

VIA Project

---

Petar Tonkovic

petar.tonkovic@cern.ch

**Supervisor:**

David Martin Clavo

david.martin.clavo@cern.ch

IT-CF-SM

June 28, 2018

1. Motivation
2. Terminology and concepts
3. Implementation so far
4. Future work
5. Demo

# Motivation

---

# Problems we are facing

- IT provides a **variety** of technical services
- Many users (systems and people) use **multiple** services daily
- These systems are often accessible by different interfaces and are often **not** connected in a convenient manner
- Users have to **manually** find workarounds or swap between systems

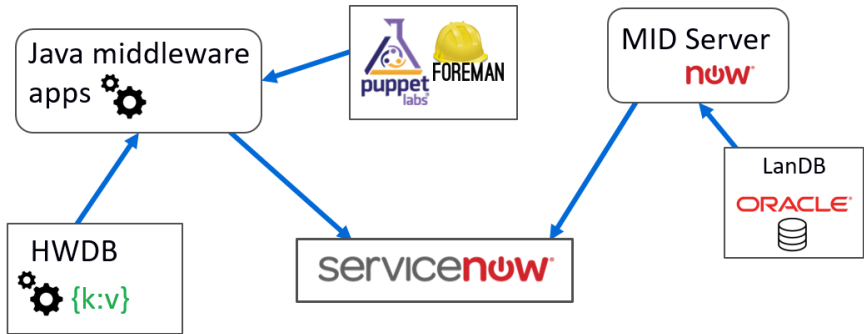
**Solution:** Centralized data access point

# Conveniences of centralized data

- Easier to **analyze data**
- Easier to **extract knowledge** from a centralized data source
- Easier to **keep track of relations** between two systems/services
- Possibility to **add new layers** on top of the existing data

## The ServiceNow CMDB in 2016

(before the HWDB migration to INFOR)



## Procurement problems

- People at FPP use **Infor** heavily, but they also often need to see information from **other** systems as well (example: host group)
- It would be useful to have all the information in one view

## Service manager problems

- They also need information from Infor (example: warranties)
- Usually they do not have **access** to this information and need to ask FPP for it (time consuming on both sides)

# Terminology and concepts

---



# Configuration Management

- **Configuration Management (ITILv2)**: The process of planning for, identifying, controlling and verifying the Configurations Items (CIs) within a service, recording their status and, in **support** of **Change Management**, assessing the potential IT **impact** of changing those items.
- **Configuration Management (ITILv3)**: The Process responsible for **maintaining information** about Configuration Items required to deliver an **IT Service**, including their **relationships**. This information is managed throughout the Lifecycle of the CI.

# Configuration Management Database

- **Configuration Management Database (ITILv2):** A database (CMDB) that **contains details** about the **attributes** and **history** of each Configuration Item (CI) and details of the **important relationships** between CIs. The CMDB is effectively a **data map** of the **physical reality** of the IT Infrastructure.
- **Configuration Management Database (CMDB) (ITILv3):** A database used to **store Configuration Records** throughout their lifecycle. The Configuration Management System maintains one or more CMDBs, and each CMDB stores **attributes** of CIs, and **relationships** with other CIs.

# Configuration Item vs Asset

The term **Configuration Item** is often confused with the term **Asset**. While these two concepts have some overlapping, their core motivation is different:

- **Asset management** - monitoring and maintaining **things of value** to an entity or group
- **Configuration management** - monitoring and keeping track of items that are **configured to perform some role** in a service, as well as the relationships between these items

# Benefits of having a CMDB

- Gaining the **benefits of data centralization**
- Ability to model relationships between CIs on a **database level** (adding attributes to relationships and modeling dependencies)
  - **Predict** how changing one CI can **impact** other CIs
  - Ability to **visualize dependencies** between CIs as **dependency graphs**
  - **Investigate** causes of **failure** of CIs by analyzing dependencies
- Possibility to implement backend of future systems directly into the CMDB to prevent unnecessary data decentralization

**Tradeoff:** Data redundancy

Implementation so far

---

After considering many different possibilities, the following technologies were chosen for the realization of the project:

- **PostgreSQL** - backend of the IT-CMDB
- **OpenShift** - platform used for hosting the IT-CMDB application and orchestrating various data source synchronizations
- **Python Flask** - framework used for implementing the web interface of the IT-CMDB application together with **SQLAlchemy**

# Reasons for choosing technologies

## PostgreSQL

- **Free** and **open-source** software
- Available in DBOD, good **documentation** and **support**
- **Fully** ACID compliant and **largely** SQL compliant
- Used by other large projects in IT and healthy **community**

## OpenShift

- Good for implementing **modular** software and highly scalable
- Good to follow **technological trends**

## Python Flask

- **Simple** and **lightweight**, which means it is highly **customizable**

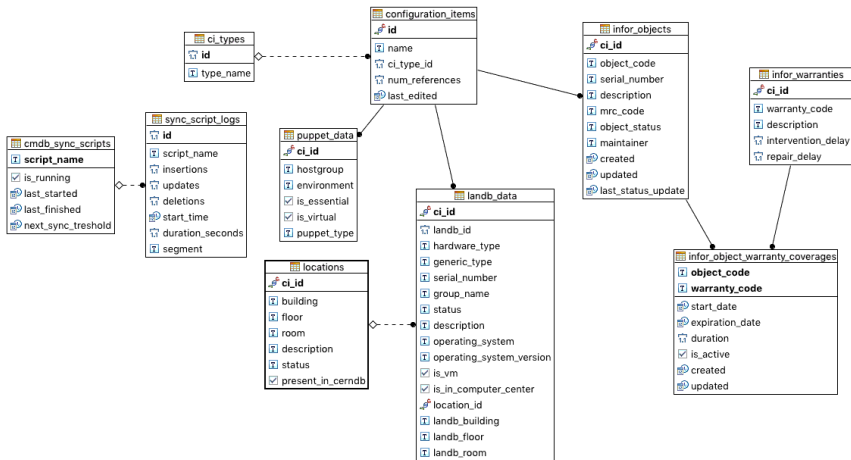
# Data sources currently implemented

Currently the system is synchronizing hourly with 4 different data sources:

- **PuppetDB** - some facts about servers ( $\approx$  41k records)
- **LanDB** - information about network devices ( $\approx$  332k records)
- **InforDB** - information about network equipment and computer center CIs ( $\approx$  107k records)
- **Locations Database** - CERN buildings and rooms ( $\approx$  21.5k records)

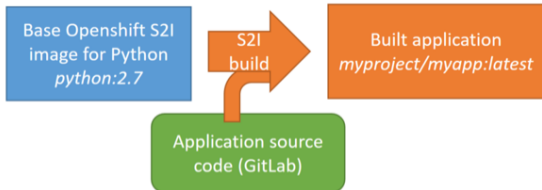


# Current state of the Entity Relation (ER) Diagram



# OpenShift application configuration

## Default build process

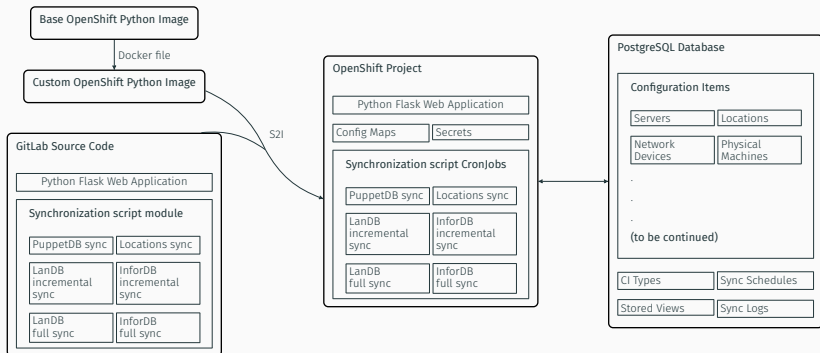


Note: "S2I" stands for Source-to-Image

## 2-step build process allowing RPM installation



# Complete Architecture Diagram



## Future work

---

- Move what has been done to production with CF services as the first clients of the IT-CMDB
- Search for other potentially interested clients and expand the IT-CMDB according to their needs
- Continue working on the web interface and add in new functionalities
- Possibly implement a REST API

Demo

---

Thank you for your time!  
Questions?

## Lessons learned

- It is good to define a **custom** Python image in **OpenShift** with a **Docker File** to make sure dependencies are installed
- **SQLAlchemy drivers** for different databases behave **differently** (for example passing parameters to raw queries)
- **OpenShift monitoring** does not necessarily detect **memory spikes** that can cause pod failures
- Sometimes is better to use **raw SQL** for big queries instead of an Object Relational Mapper (ORM) due to **memory limitations**



## Welcome to the IT-CMDB

Search for configuration item

**Enter some property value according to which the CMDB will be queried**

## Welcome to the IT-CMDB

Search for configuration item

Enter some property value according to which the CMDB will be queried

Result

Show  entries

Search:

CI Name	CI Type	Last Edited (in CMDB)	
gregorysnow	mobile_device	2018-04-11 12:33:36.574014+02:00	<a href="#">Details</a>
gsnowdell	physical machine	2018-04-11 12:33:36.574014+02:00	<a href="#">Details</a>
gsnowdell6000	physical machine	2018-04-11 12:33:36.574014+02:00	<a href="#">Details</a>
gsnowmacbookpro	physical machine	2018-04-13 09:55:52.590501+02:00	<a href="#">Details</a>
gsnowtempmacbook	physical machine	2018-04-13 09:55:52.590501+02:00	<a href="#">Details</a>
ipad2cmssnow	physical machine	2018-04-13 09:55:52.590501+02:00	<a href="#">Details</a>
itcs-asosnow01	physical machine	2018-04-13 09:55:52.590501+02:00	<a href="#">Details</a>
itmon-snowconsumer-6f9e8a217c	virtual machine	2018-04-06 15:55:41.206926+02:00	<a href="#">Details</a>
itmon-snowconsumer-985aeff14d	virtual machine	2018-04-06 15:55:41.206926+02:00	<a href="#">Details</a>
itmon-snowconsumer-a3665ccb70	virtual machine	2018-04-06 15:55:41.206926+02:00	<a href="#">Details</a>

Showing 1 to 10 of 48 entries

Previous **1** 2 3 4 5 Next

## CI details for ithdp1301

### Basic CI Info

**CI name:** ithdp1301

**CI type:** physical machine

**Last edited:** 2018-05-07 14:38:24.509854+02:00

### Additional info

**Hardware type according to LanDB:** hns2600kp

**Generic type according to LanDB:** computer

**LanDB status:** active

**Operating system:** linux

**Operating system version:** slc6

**Location:** 513/R-050

**Hostgroup:** hadoop/hdpqa/hamenode

**Environment:** hdp\_prod

**PuppetDB essential flag:** False

# Backup slides (5/8)

## Hardware info

**Serial number:** DL6833098-4M0717IN096-2

**Infor object code:** HCCSSYS011-00003558

**MRC code:** CDC01

**Infor status:** I

**Maintainer:** ITPROCOS

## Warranty info

**Warranty code:** DL6833098

**Warranty description:** standard on-site warranty (parts exchange only)

**Start date:** 2017-10-02 00:00:00+02:00

**Expiration date:** 2020-06-30 00:00:00+02:00

**Duration:** 1002

**Active flag:** True

**Warranty code:** 4

**Warranty code:** 12

**Warranty code:** DL6833098-EXTENDED

**Warranty description:** standard on-site warranty (parts exchange only)

**Start date:** 2020-07-01 00:00:00+02:00

**Expiration date:** 2022-07-01 00:00:00+02:00

**Duration:** 730

**Active flag:** False

**Warranty code:** 4

**Warranty code:** 12

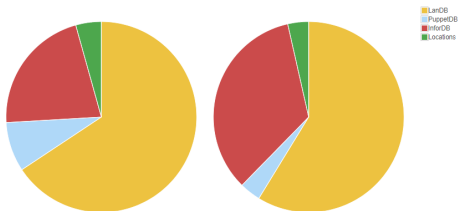
# Backup slides (6/8)

## Dashboard

### Last synchronization times

👤 LanDB (full sync)	2018-06-11 00:49:58
👤 LanDB (incremental sync)	2018-06-11 19:16:19
👤 PuppetDB	2018-06-27 15:03:26
👤 InforDB	2018-06-27 15:36:09
👤 Locations	2018-06-27 15:47:45

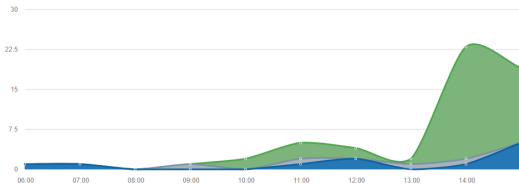
### General Information



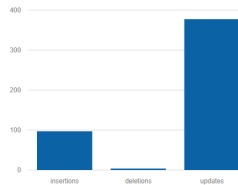
Number of records per source

Average synchronization time [s]

### Number of records affected in recent synchronizations (Puppet)

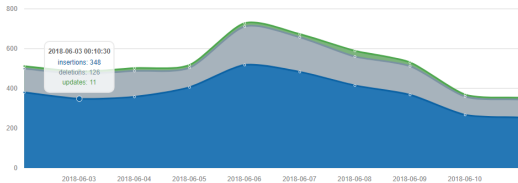


### Average number of affected records (Puppet)

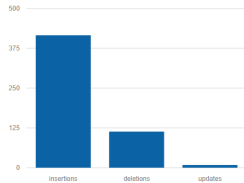


# Backup slides (7/8)

Number of records affected in recent synchronizations (LanDB)



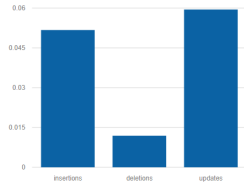
Average number of affected records (LanDB)



Number of records affected in recent synchronizations (Locations)



Average number of affected records (Locations)



# Backup slides (8/8)

name	description	object_code	serial_number	warranty	warranty_start	warranty_end	is_vm	is_in_computer_center	is_essential
ihdp1301	FORMAT DB2 server system unit	HCCSSYS011-00003558	DL6833098-4M0717IN096-2	standard on-site warranty (parts exchange only)	2017-10-02 00:00:00	2020-06-30 00:00:00	false	true	false

