
CERN

Configuration automation for CERN's new Wi-Fi infrastructure

Quentin Barrand
CERN IT / Communication Services

Agenda

- Configuring Aruba controllers and APs
- Why and how to automate ?
- Interacting with the devices
- Questions & feedback

Our PROD cluster

Uplink
(Brocade MLXe routers)

Mobility Masters
(master / standby)

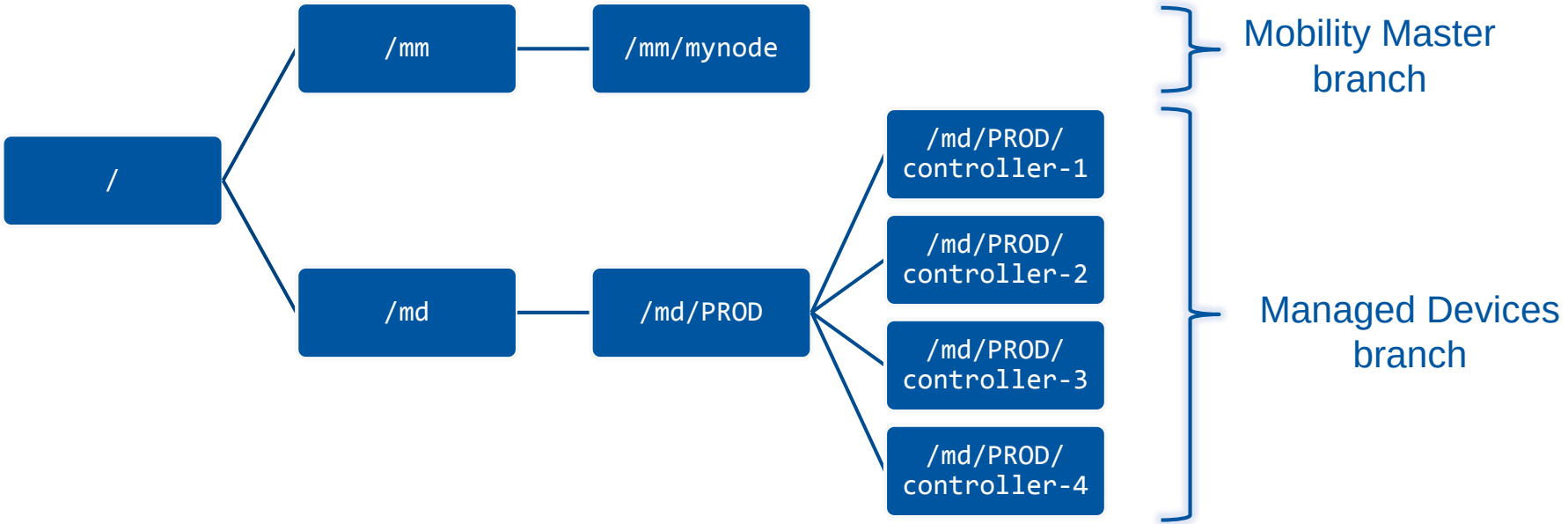
7240 controllers
(all active)

Access Points
(9 models)

How are APs managed ?

- Any AP belongs to an AP Group
- The group's settings are applied to all its APs
  - Transmitted power
  - Allowed channels & channel width
  - Broadcasted SSIDs
  - …
- But !
  - RF settings *can* also be defined per-AP
  - External antenna gains *must* be defined per-AP

Configuration hierarchy



How to deploy an AP ?

- **Simple !**
  - Include the cluster's IP address as **DHCP option 43**
  - Add an entry in the Mobility Master **specifying the AP Group**
  - Optionally, define AP-specific settings (radio parameters, antenna gains…)

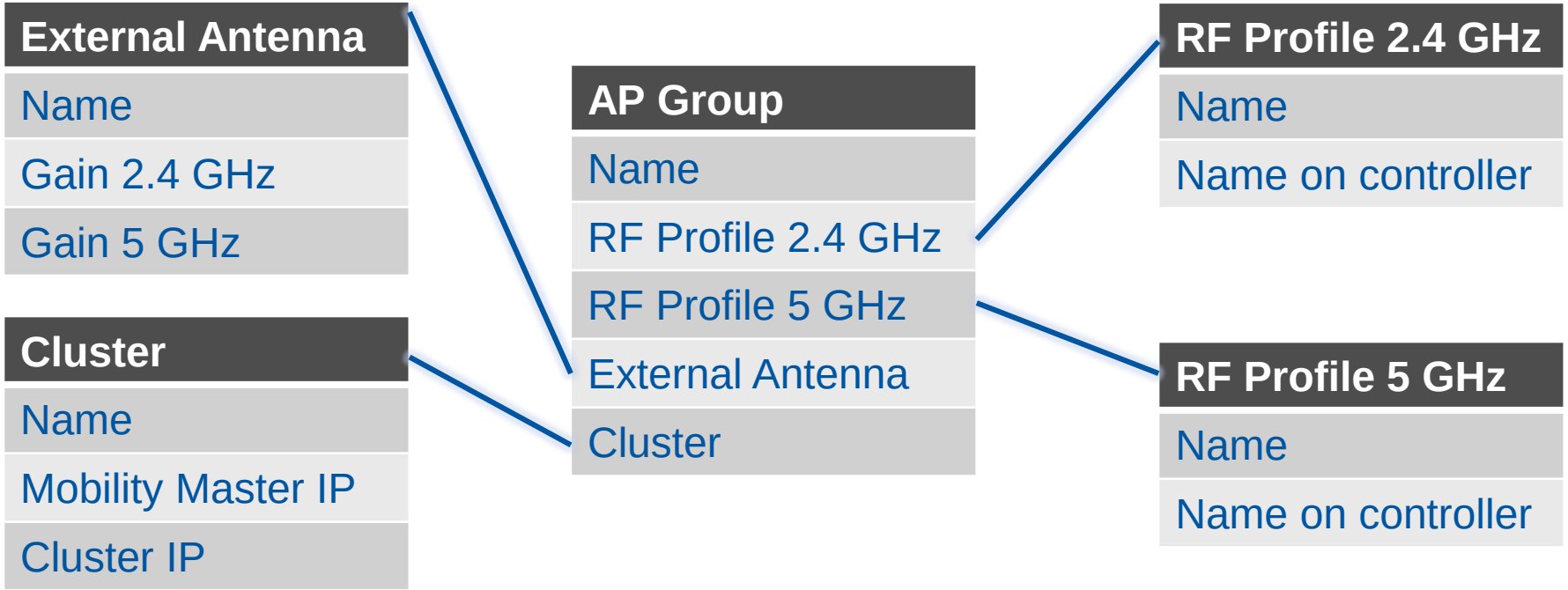
- **Except…**
  - We're deploying at least 4000 AP
  - We have 3 clusters (lab, pilot, production)
  - How do we make sure that the same rules will be enforced in the future ?

CERN Network's Management System

- One central place of truth: **LANDB**
  - Oracle database with a web front-end
  - Stores the information of all the devices connected to the network

- One main management application: **CFMGR**
  - Uses LANDB data to build the network topology
  - Configures equipments (switches, routers...) and services (DNS, DHCP…) mainly via CLI and SNMP
  - ~200k lines of highly-sophisticated Perl code

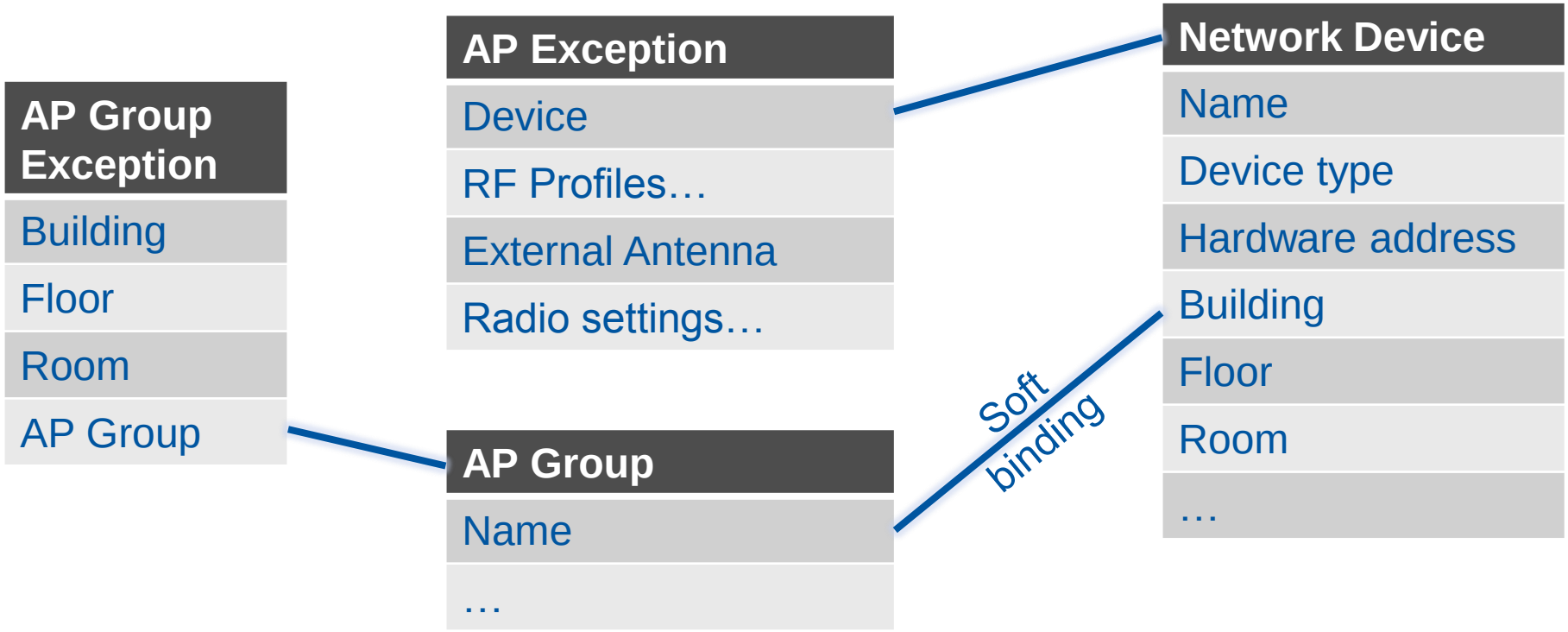
Adding data to LANDB



Maintain exceptions only

- Infer the AP group from the building number
  - AP in building 33 → `0033_default`
  - Implemented in LANDB directly (Oracle view)
- A table to define exceptions
  - `600/*-*` → `0031_default`
  - `31/S-*` → `TEST`
  - `40/1-A032` → `0040_Small-Office`
- Another one to define exceptions at the AP level
  - RF profiles
  - Very specific radio settings
  - External antenna

Exception tables



How to deploy an AP ? (continued)

- Integrate Aruba into CERN's NMS
  - Only use the device's CLI for troubleshooting and engineering
  - Script all repetitive tasks

- Just declare the AP as a network equipment in LANDB
  - Name
  - Manufacturer & model
  - Wired MAC address
  - Location (building and office)
  - IP network to which it is connected

- Optionally, add an AP exception for the AP
- Synchronize the data regularly

NETMGR

- **Python 3** application
  - Very large ecosystem (`requests`, `SQLAlchemy`…)
  - Big community at CERN
  - Easier to maintain (vs. Perl)
- ~3.5k SLOC (excluding tests)
- Features
  - Compares and pushes LANDB changes to the Mobility Master
  - Also includes a few cross-cluster `show` commands

Interacting with the Mobility Master

- ItProvides a HTTP REST API !
  - JSON output ☺
  - No need to use expect-like patterns anymore ☺

- Better than plain text, but…
  - The API is not versioned ☹
  - The HTTP return code is always 200 OK ☹
  - The error messages are either missing or useless ☹
  - There is no consistency in the parameter names and returned payloads
  - A lot of interesting endpoints are private / undocumented

CLI vs JSON

```
(R513-C-CARMM-21) #show ap database

AP Database
-----------
Name                Group          AP Type  IP Address      ...
----                -----          -------  ----------      ...
U28-S-AP1-BARI1-13  0028_default   325      172.28.18.136   ...
U28-S-AP1-BARI1-14  0028_default   325      172.28.18.149   ...
U28-S-AP1-BARI1-15  0028_default   325      172.28.18.137   ...
U28-S-AP1-BARI1-19  0028_default   325      172.28.18.138   ...
U28-S-AP1-BARI1-21  0028_default   325      172.28.18.139   ...
U28-S-AP1-BARI1-24  0028_default   325      172.28.18.140   ...
U28-S-AP1-BARI1-27  0028_default   325      172.28.18.141   ...
U28-S-AP1-BARI1-29  0028_default   325      172.28.18.142   ...
U28-S-AP1-BARI1-33  0028_default   325      172.28.18.143   ...
U28-S-AP1-BARI1-36  0028_default   325      172.28.18.144   ...
U28-S-AP1-BARI1-39  0028_default   325      172.28.18.145   ...
U28-S-AP1-BARI1-4   0028_default   325      172.28.18.133   ...
U28-S-AP1-BARI1-40  0028_default   325      172.28.18.146   ...
U28-S-AP1-BARI1-44  0028_default   325      172.28.18.147   ...
```

```
$> curl -X GET \
/v1/configuration/showcommand?json=1&command=show+ap
+database HTTP/1.1
```


```
{
  "AP Database": [
    {
      "AP Type": "325",
      "Flags": "2",
      "Group": "0028_default",
      "IP Address": "172.28.18.136",
      "Name": "U28-S-AP1-BARI1-13",
      "Standby IP": "188.184.0.152",
      "Status": "Up 5d:10h:12m:38s",
      "Switch IP": "188.184.0.151"
    },
    ...
  ]
}
```

Features

Every 5 minutes	Fetch and store the list of wireless associations (security)
Every 15 minutes	Synchronize AP entries and radio settings
Every 30 minutes	Synchonize antenna gains (requires reading the AP's boot parameters)

```
(netmgr) cfmgr/qubarran[1002] netmgr show wifi ap --loc 31/r -u
11 devices match your query.
AP name             Model  AP Group      Location  Status              Flags  Zone      2GHz (M/Chan/EIRP/Cli)  ...
------------------  -----  ------------  --------  ------------------  -----  --------  ----------------------  ...
U31-S-AP1-BARI1-12  325    0031_default  31/R-030  Up 5d:21h:3m:1s     2      IT-PILOT  APHT / 1 / 7 / 0        ...
U31-S-AP1-BARI1-14  325    0031_default  31/R-023  Up 5d:21h:31m:32s   2      IT-PILOT  APHT / 6 / 8 / 0        ...
U31-S-AP1-BARI1-15  325    0031_default  31/R-016  Up 5d:21h:3m:2s     2      IT-PILOT  APHT / 1 / 8 / 1        ...
U31-S-AP1-BARI1-17  325    0031_default  31/R-208  Up 5d:21h:23m:23s   2      IT-PILOT  APHT / 6 / 8 / 0        ...
U31-S-AP1-BARI1-19  325    0031_default  31/R-006  Up 5d:21h:40m:26s   2      IT-PILOT  APHT / 11 / 7 / 0       ...
...
```

Wrap-up

- Deploying thousands of devices definitely requires automation
  - Brand new infrastructure but similar workflow and tools as before
  - Deployment and maintenance teams use NETMGR instead of going to the controllers
- More vendors are implementing APIs
  - Removes a lot of boilerplate code
  - Unfortunately not all are ready for prime time…
- **Python** is a great environment for network-related projects
  - Some components of NETMGR have already been reused

Thank you !
ありがとう

Questions, remarks & thoughts


CERN