



Pilots (3.0)

Wojciech Krzemień (NCBJ)

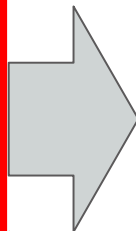
Andrew McNab

Federico Stagni

The 8th DIRAC Users Workshop
23.05 2018, Lyon

- Intro
- Pilots model
- Pilots in action
- Summary & Outlook

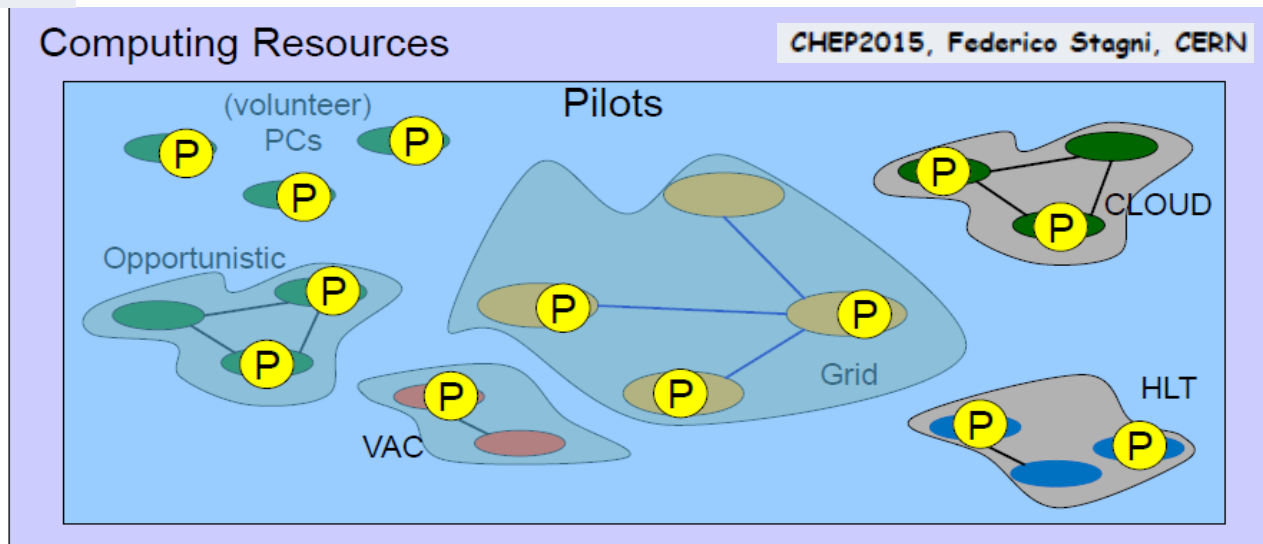
**It seems like the grid is not
anymore
“The Grid”**



Heterogeneity is the norm

(WLCG, CREAM, ARC, HTCondor, HLT, HPC, Opportunistic,
Volunteer, Virtual Machines, Clouds, VAC, BOINC, containers,
Docker...)

DIRAC pilots



It seems like the grid is not anymore
“The Grid”

Heterogeneity is the norm

(WLCG, CREAM, ARC, HTCondor, HLT, HPC, Opportunistic, Volunteer, Virtual Machines, Clouds, VAC, BOINC, containers, Docker...)

Pilot model provides the layer to hide diversity of computing resources

What are pilots really?

- Simple, small, standalone Python scripts
- Common for all DIRAC communities
- Pilot makes possible to run jobs on a worker node (e.g CREAM, VM, etc.)

What are pilots really?

- Simple, small, standalone Python scripts
- Common for all DIRAC communities
- Pilot makes possible to run jobs on a worker node (e.g CREAM, VM, etc.)

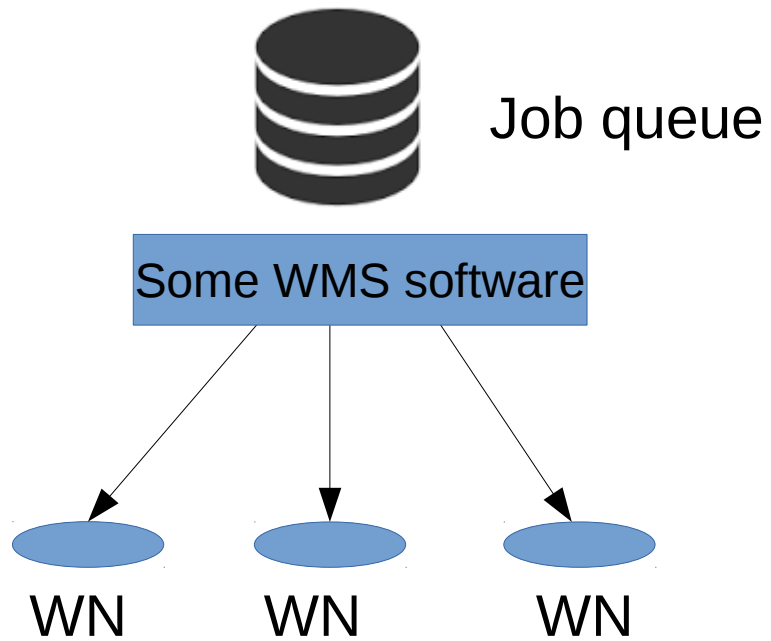


- 1) Start
- 2) Install and configure DIRAC environment
- 3) Launch a process for matching jobs (**JobAgent**)
- 4) Provide monitoring

Minimum requirements:

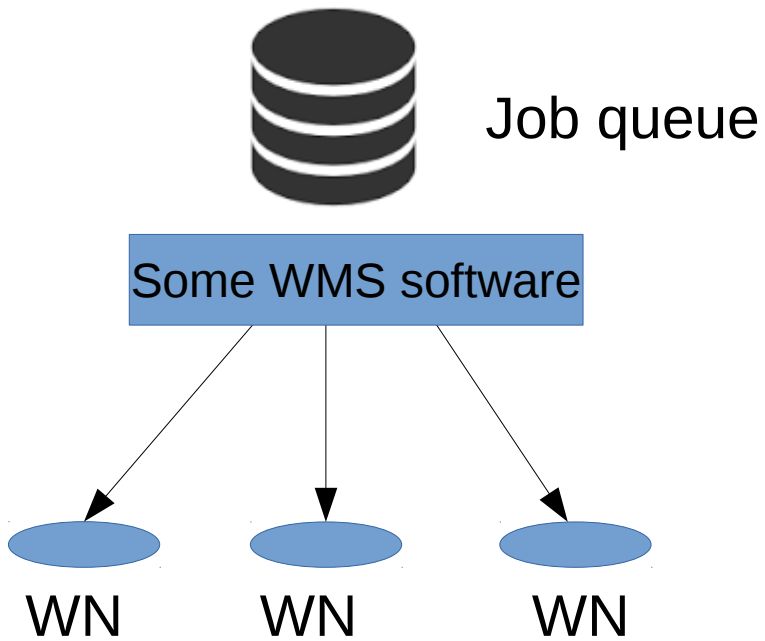
- Python 2.6+ installed on a Worker Node
- DIRAC can be installed on this Worker Node

Grid 'push' model

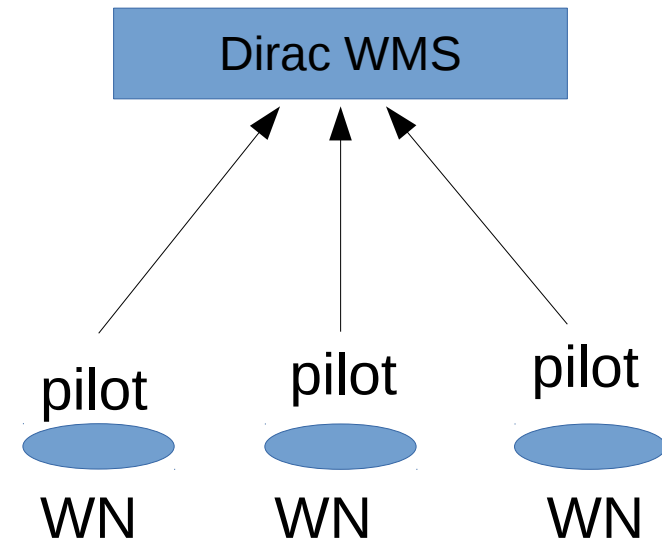


Pilots model

Grid 'push' model



Pilots model



Pilots evolution



Agents → Pilots

~2006



Pilots 2.0

2014

(DIRAC v6r12)



Pilots 3.0

2018

(DIRAC v6r20)

optional

Extensibility

Universality



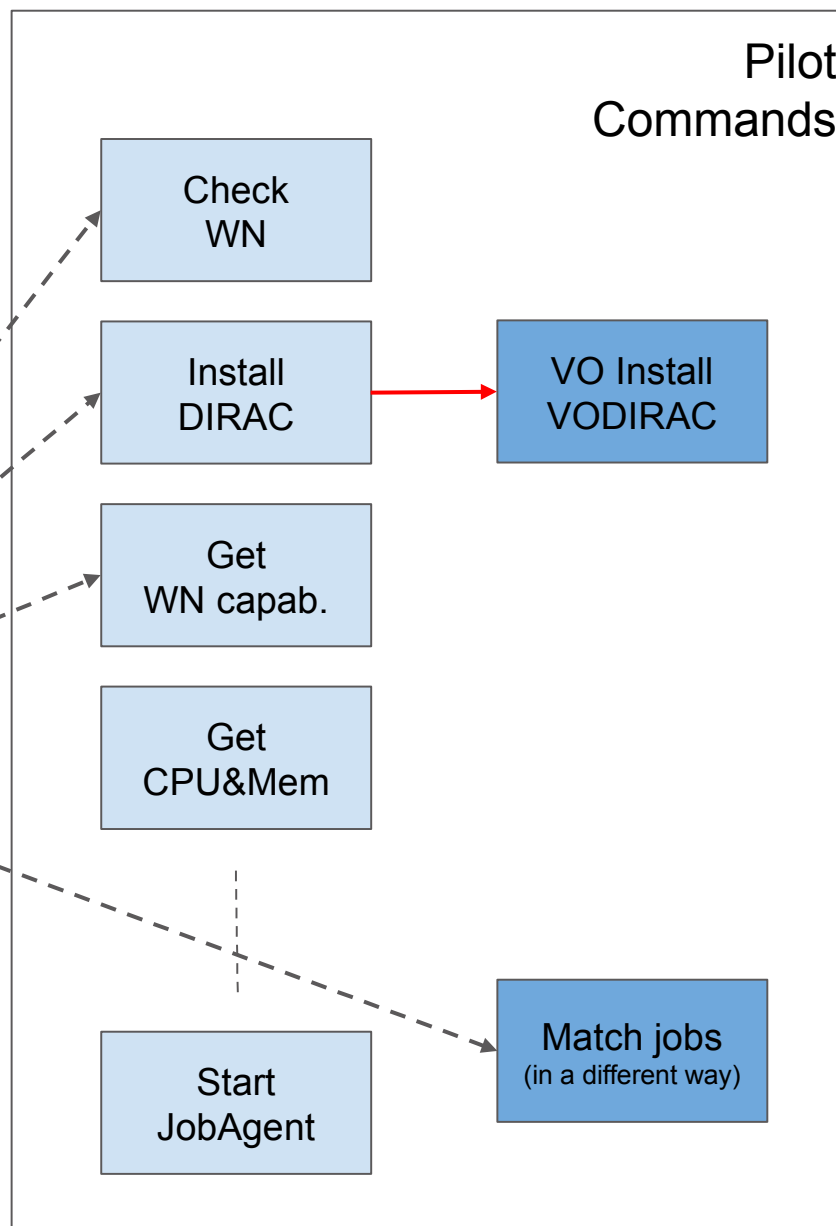
Pilots command

A toolbox of pilot capabilities (“pilot commands”) is available for the pilot.

Pilots can be configured to run a set of them



Any configuration possible, any order possible



Extensions possible via VO-specific commands

- 1) Start it
- 2) Configure it
- 3) Get the jobs
- 4) Monitor it

Start it

Vac - the first Vacuum system

Infrastructure-as-a-Client (IaaS)



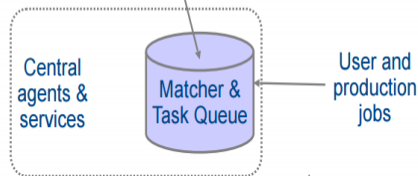
Since we have the pilot framework, we could do something really simple

Strip the system right down and have each physical host at the site create the VMs itself.

Instead of being created by the experiments, the virtual machines appear spontaneously "out of the vacuum" at sites.

Requests for real jobs

Use same VMs as with IaaS clouds



Just start it here

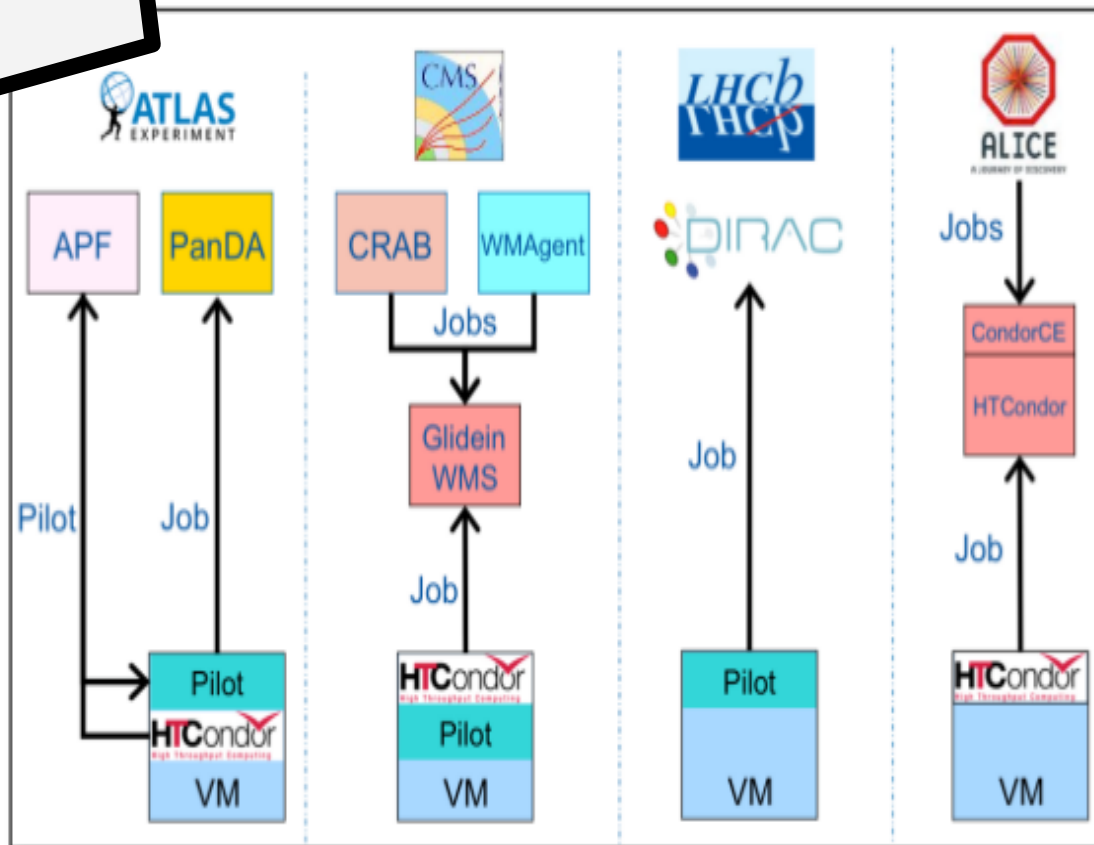


Send it there

- Can be sent as 'pilot job' to any Grid CE type
- Can be run as part of the contextualization of a Virtual Machine

WLCG WS
2016

2015 approaches



- Simple and effective architecture with the resources like clouds and VACs
- Use pilots 2.0 as a part of the contextualization
- No need to use additional batch script

Configure it

- Pilot minimal role is to:
 - Setup the environment
 - Install DIRAC
- Self-discovering WN capabilities
 - Including CPU power
 - Using DB12
 - And #processors
 - MJF supported
 - And memory
- Commands list configurable:

With REST interface on top of DIRAC Configuration System

 - By type of “Grid”
 - i.e. some commands may be needed only for volunteer computing resources
 - By setup (e.g. production, test, ...)

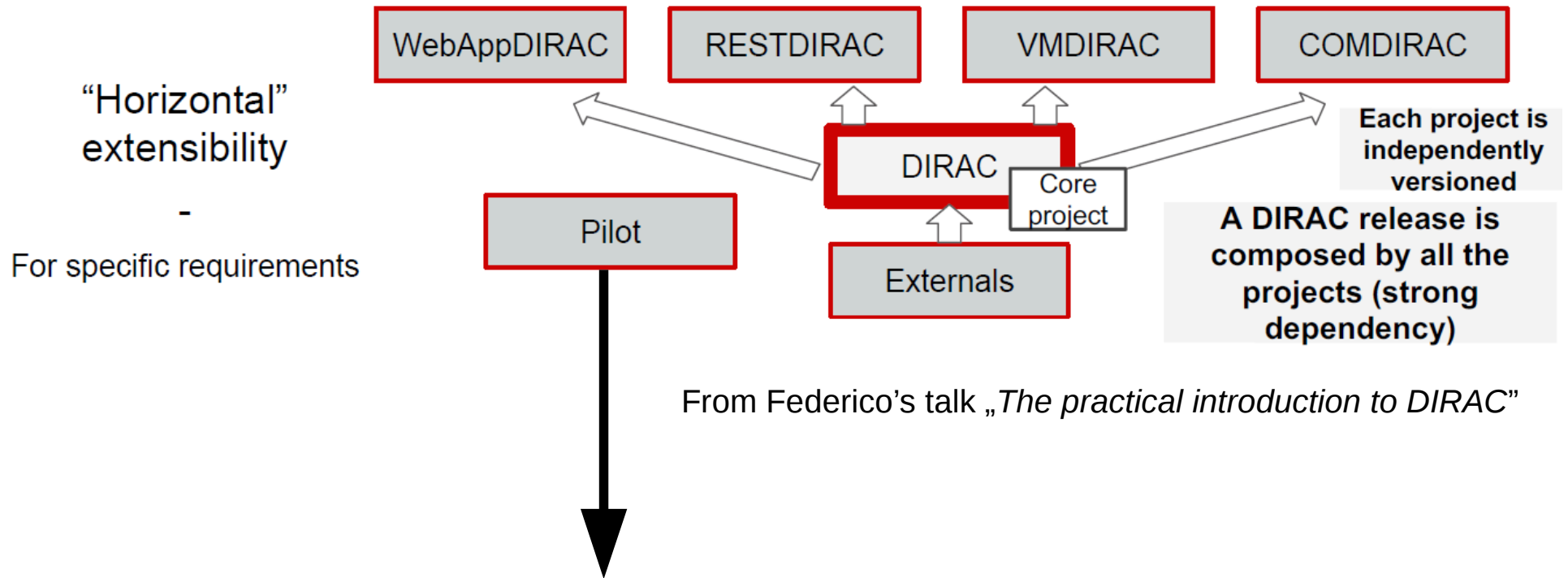
Get the jobs

- A DIRAC client is installed
- By default a “JobAgent” is used to match the capabilities of the WN with the requirements of the waiting jobs.
- Support for MultiProcessor jobs:
 - Pilots advertise multi-processor payload slots
 - Multi-processor payloads matched
 - No mix/backfilling allowed yet



Pilots 3.0

Code organization



<https://github.com/DIRACGrid/Pilot>

Pilot repository is independent of any other DIRAC repositories.

(since pilots must be able to start in the DIRAC-free environment)

Bootstrap issue

In some environments pilot wrapper starts in the „vacuum” (e.g. VM)

Must find information to start:

- Where to find pilot code
- Info about Queue, sites, CE

Idea: General solution that would work „everywhere” with minimal pilot wrapper

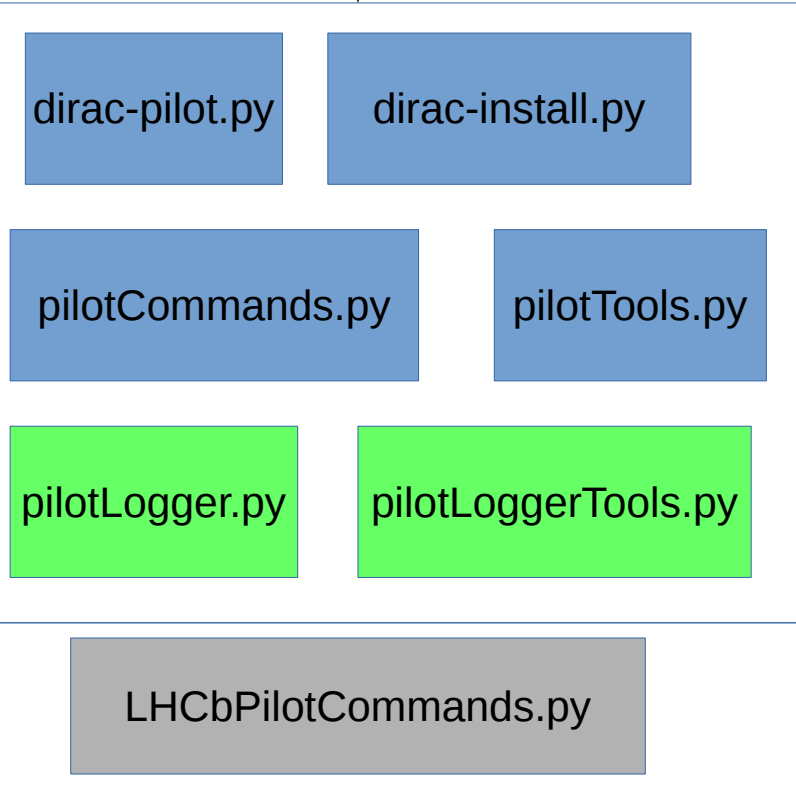
```
#!/bin/bash
#-----
#
# pilot_wrapper.sh
#
#-----
#
# Characteristics:
# * VO/communities agnostic
# * IMMUTABLE!
#
# Args:
# $1 : URL from where to get the pilot files
# $2 : CE name
# $3 : queue name
#
#-----
if [ $1 ]
then
  if [[ $1 == 'http'* ]]
  then
    wget --no-directories --recursive --no-parent --execute robots=off --reject 'index.html*'
$1
  elif [[ $1 == 'file'* ]]
  then
    es=''
    cp "${1/file:\//}$es"/*.py .
    cp "${1/file:\//}$es"/*.json .
  fi
else
  echo "ERROR: no URL supplied"
  exit 1
fi
# Now run the pilot script
# X509_USER_PROXY=/scratch/plt/etc/grid-security/hostkey.pem \
python dirac-pilot.py \
--debug \
--Name $2 \
--Queue $3
```

Pilot code

Pilot wrapper

Pilot code

Pilot wrapper

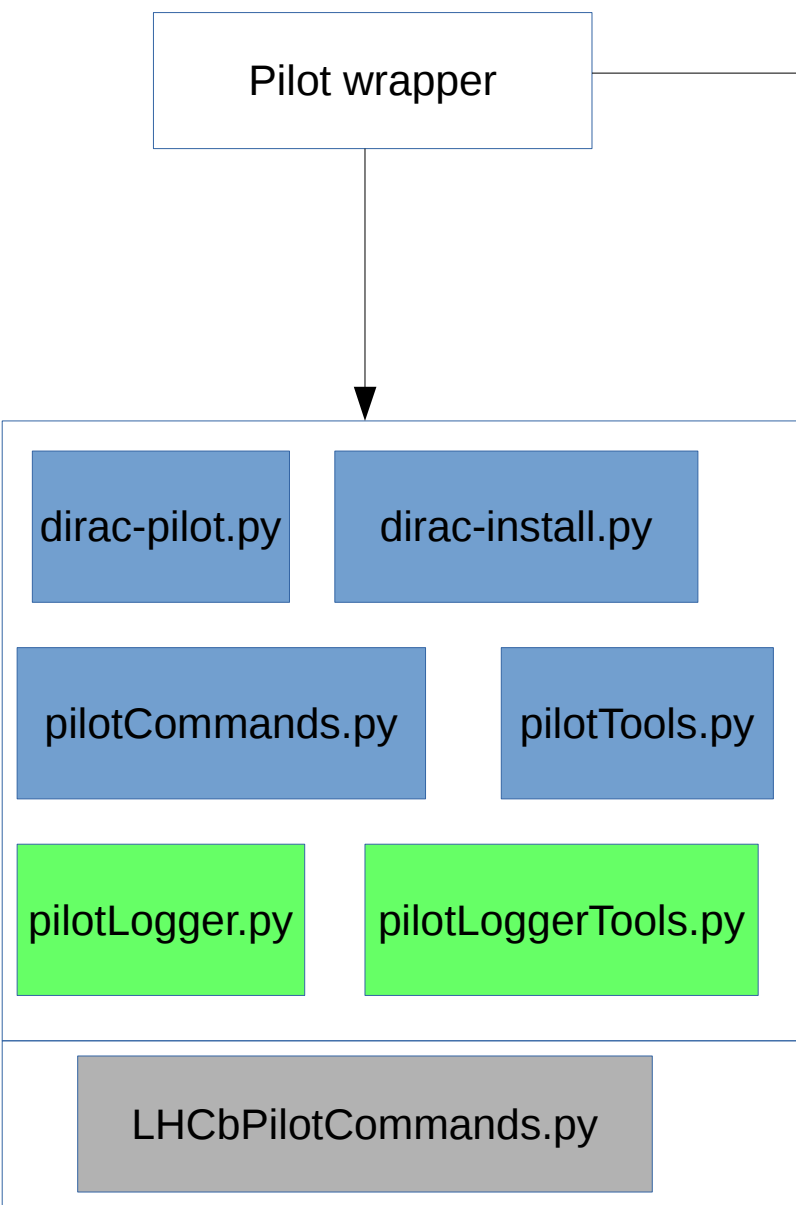


Universal code

Monitoring

Extensions

Pilot code



pilot.json
json file

```
{
  "Setups": {
    "LHCb-Production": {
      "CheckVersion": "True",
      "Project": "LHCb",
      "Commands": {
        "Vac": [
          "NagiosProbes",
          "CheckWorkerNode",
          "LHCbInstallDIRAC",
          "LHCbConfigureBasics",
          "LHCbConfigureSite",
          "LHCbConfigureArchitecture",
          "LHCbConfigureCPURequirements",
          "MultiLaunchAgent"
        ],
        "Vcycle": [
          "NagiosProbes",
          "CheckWorkerNode",
          "LHCbInstallDIRAC",
          "LHCbConfigureBasics",
          "LHCbConfigureSite",
          "LHCbConfigureArchitecture",
          "LHCbConfigureCPURequirements",
          "MultiLaunchAgent"
        ],
        ...
      }
    },
    "CREAM": [
      "CheckWorkerNode",
      "LHCbInstallDIRAC",
      "LHCbCleanPilotEnv",
      "LHCbConfigureBasics",
      "CheckCECapabilities",
      "CheckWNCapabilities",
      "LHCbConfigureSite",
      "LHCbConfigureArchitecture",
      "LHCbConfigureCPURequirements",
      "LaunchAgent"
    ]
  },
  ...
}
```

Universal code

Monitoring

Extensions

<https://github.com/DIRACGrid/Pilot/tree/master/Pilot>

pilot.json file

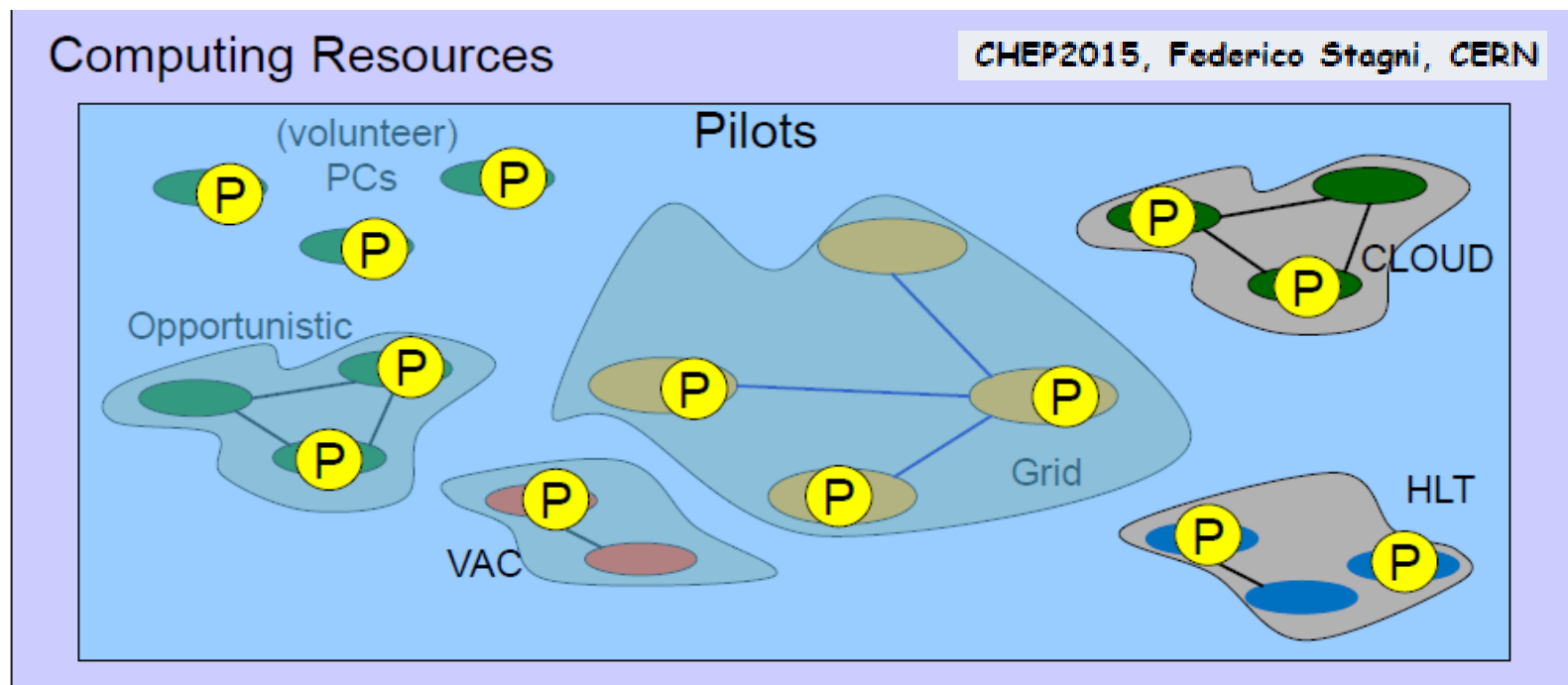
- Generated based on Configuration System settings
- Pilot wrapper expects to find it in the given location

To synchronize pilot.json ↔ CS

- **UpdatePilotCStoJSONFile** option (introduced in v6r20)
in the Configuration/Server to keep pilot.json in sync
- If **UpdatePilotCStoJSONFile** set to True, then also **pilotFileServer**
must be set

<http://dirac.readthedocs.io/en/integration/AdministratorGuide/Configuration/ConfReference/Systems/Configuration/Services/Server/index.html#configurationserver>

Monitor it



- We want info **before** DIRAC is installed
- Some logs are available - „WLCG” (from CREAM and ARCs CEs only),
- No automatized, general (and scalable) log system exists

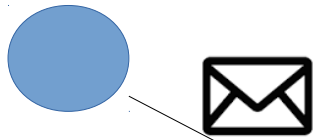
"I've booted up" ...
"I found the DIRAC pilot ok" ...
"I'm about to shutdown"...

"I installed DIRAC via SetupProject/dirac-install" ...
"This machine has power of 11 HS06"....
"This machine is SLC6/CC7" ...
"I matched a job" or
"I failed to match a job"... and so on.

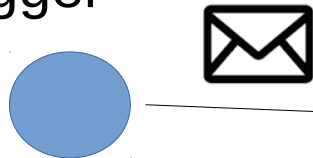
Pilot Logger & MQ

Producers

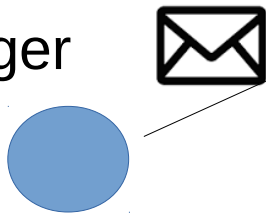
Pilot Logger



Pilot Logger



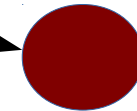
Pilot Logger



Message Queue



Consumer1

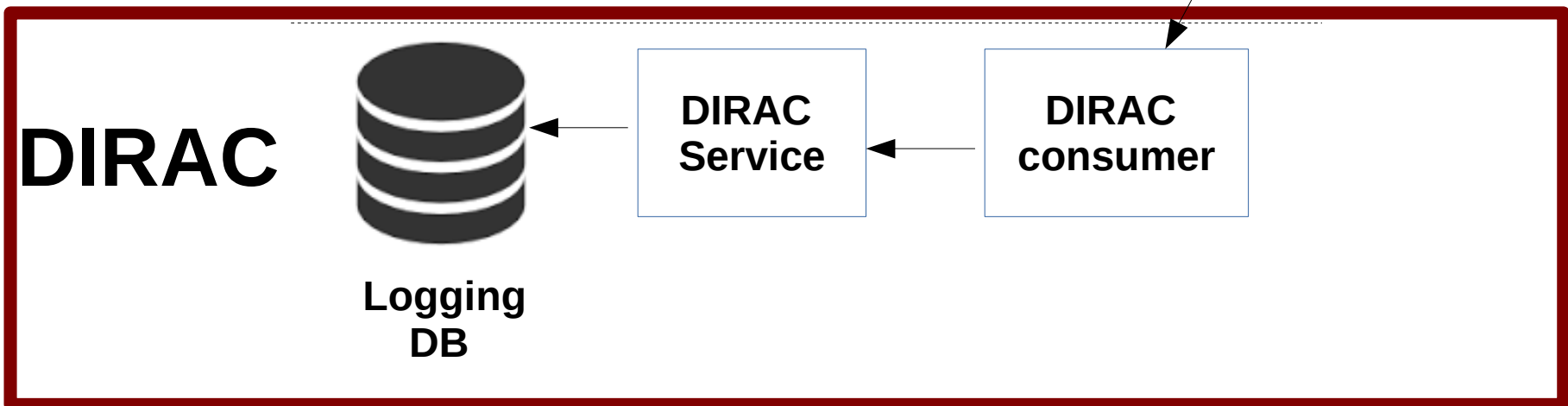
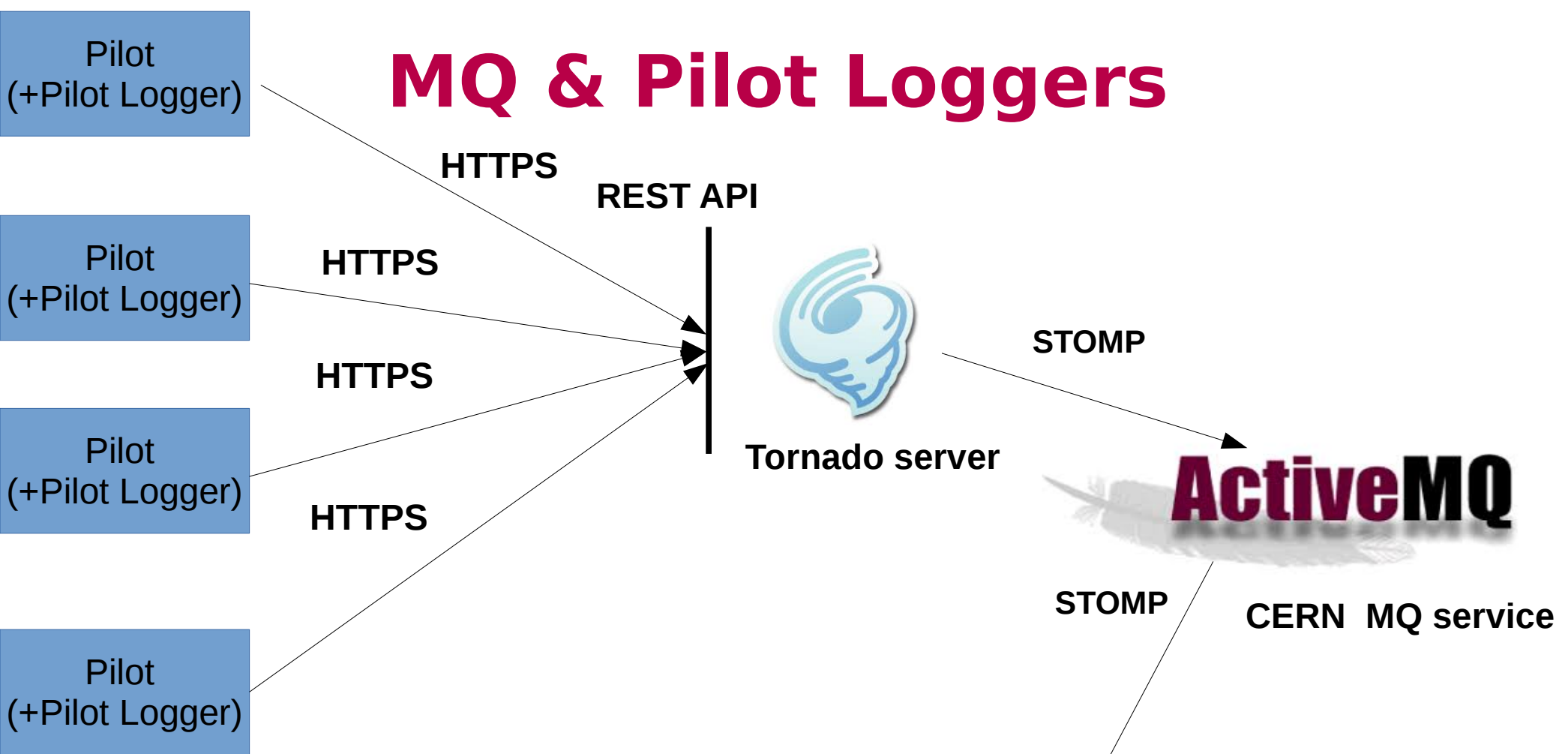


Consumer2

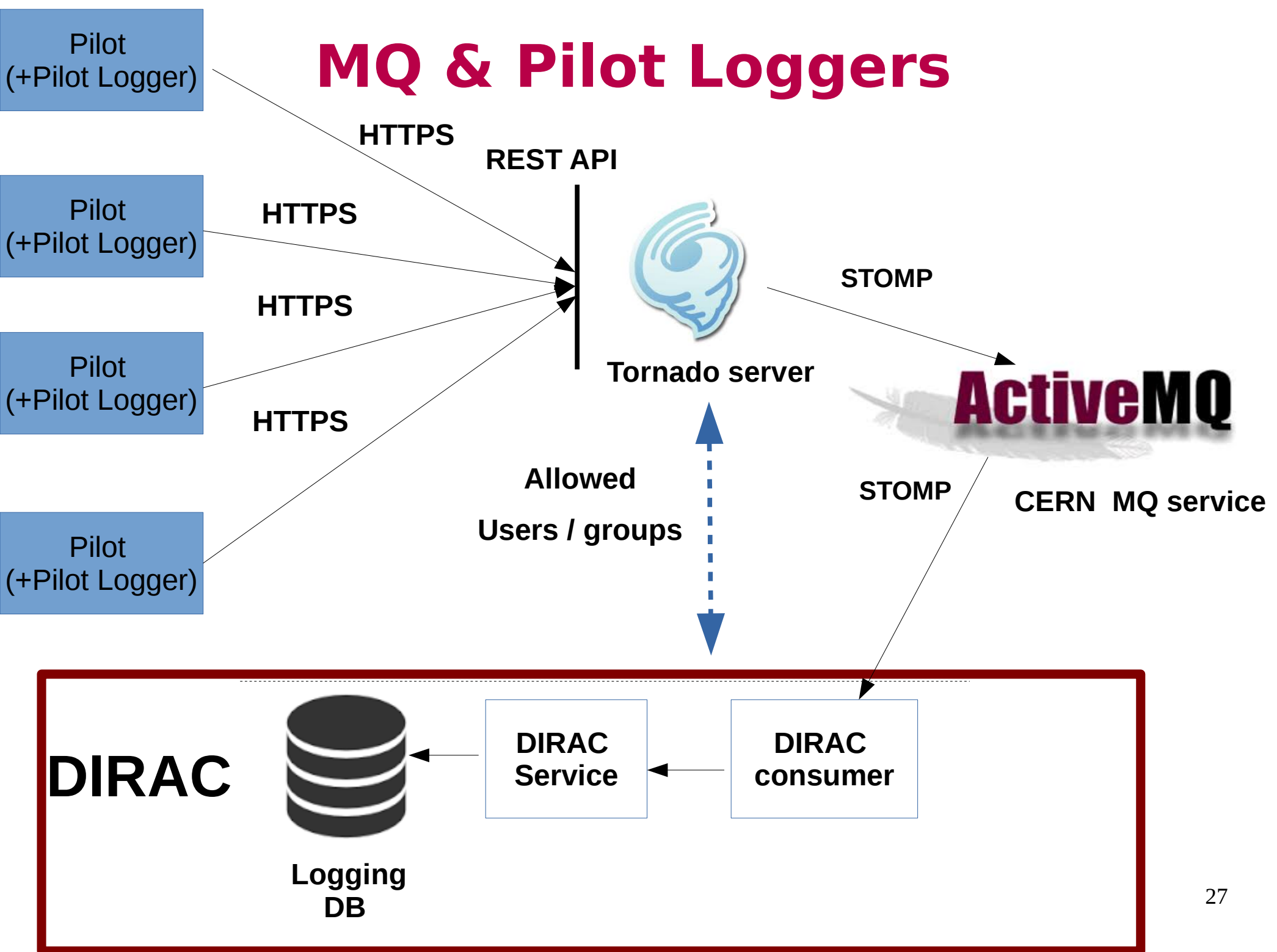
Idea: send logs to some dedicated MQ server

- Pilot Logger transparently added to Pilot repository
- Can be activated using a dedicated pilotCommand option

MQ & Pilot Loggers



MQ & Pilot Loggers



Starting Pilot 3.0

- Pilots 3.0 available from DIRAC v6r20
- Be default SiteDirector agents sends pilots 2.0
- To enable pilots 3.0:
 - Set Pilot3 flag

- For details see the documentation:

<http://dirac.readthedocs.io/en/integration/AdministratorGuide/Configuration/ConfReference/Systems/WorkloadManagement/Agents/SiteDirector/index.html#conf-sitedirector>

Summary

- Pilot model has been proven as a approach that:
 - provides an uniform interface that hides underlying variety of resources
 - is much more flexible than traditional ‘push’ model
 - is highly configurable and extensible (commands)
- Used by all DIRAC communities in every DIRAC installation
- Next generation of Pilot 3.0 available as an option
- Universal monitoring with Pilot Loggers approach is being developed

Thank you

To extend pilots

RFC 18 - Pilots 2.0: generic, configurable pilots:

<https://github.com/DIRACGrid/DIRAC/wiki/Pilots-2.0:-generic,-configurable-pilots>

Dirac Docs:

<http://dirac.readthedocs.io/en/integration/AdministratorGuide/Systems/WorkloadManagement/Pilots/index.html>