INDIGO - DataCloud

RIA-653549

CMS Experience with Indigo DataCloud

WLCG Grid Deployment Board meeting

Andrea Ceccanti; Davide Salomoni; Sonia Taneja; Stefano dal Pra: INFN-CNAF
Marica Antonacci; Giacinto Donvito: INFN-BA
Tommaso Boccali: INFN-PI
Daniele Spiga: INFN-PG

INDIGO - DataCloud is co-founded by the Horizon 2020Framework Programme

Outline

- Brief introduction to INDIGO-DataCloud project

- CMS implementation using INDIGO solutions
  - Motivations and objectives
  - Architectural description
  - Status and Next steps

The Project: INDIGO-DataCloud

INDIGO-DataCloud

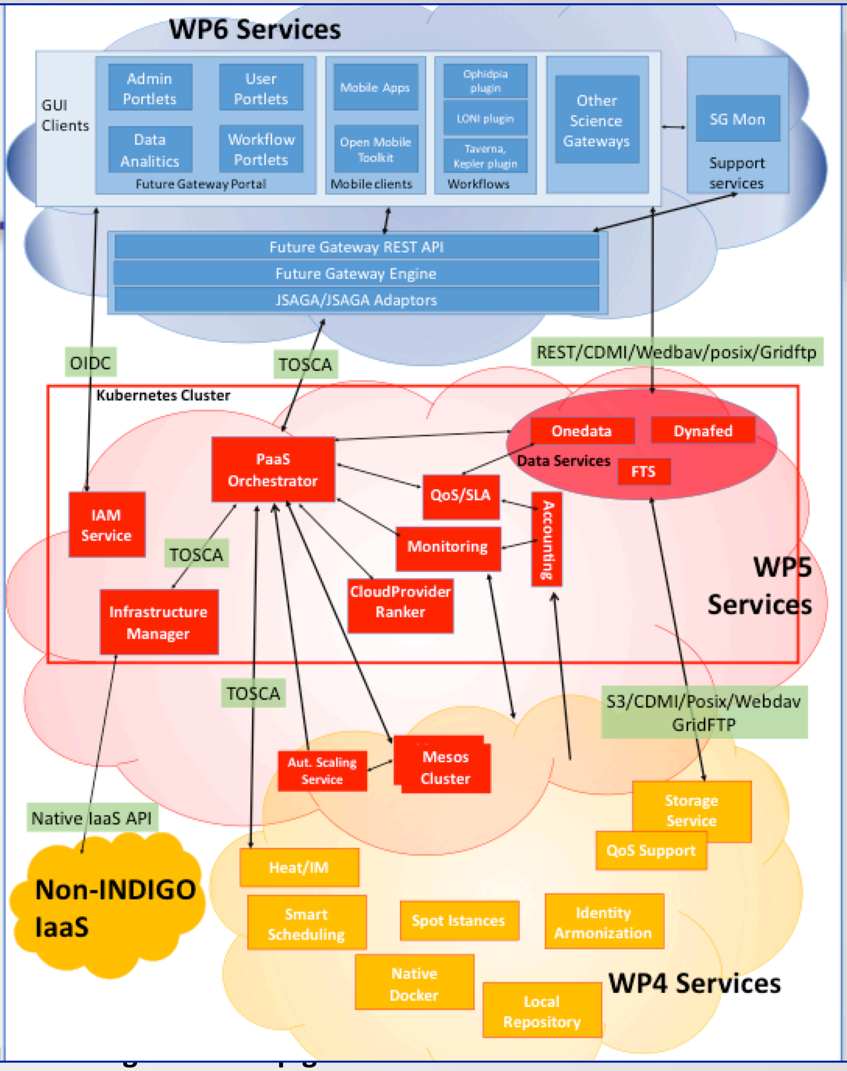
- **An H2020 project** approved in January 2015 in the EINFRA-1-2014 call
  - 11.1M€, 30 months (**from April 2015 to September 2017**)
- **Who**: **26 European partners** in 11 European countries
  - Coordination by the Italian National Institute for Nuclear Physics (INFN)
  - Including developers of distributed software, industrial partners, research institutes, universities, e-infrastructures
- **What**: **develop an open source Cloud platform** for computing and data ("DataCloud") tailored to science.
- **For**: **multi-disciplinary scientific communities**
  - E.g. structural biology, earth science, physics, bioinformatics, cultural heritage, astrophysics, life science, climatology
- **Where**: deployable on **hybrid (public or private) Cloud infrastructures**
  - INDIGO = **IN**tegrating **D**istributed data **I**nfrastructures for **G**lobal Expl**O**itation
- **Why**: answer to the technological **needs of scientists** seeking to easily exploit distributed Cloud/Grid compute and data resources.

INDIGO“simplified” architecture

INDIGO - DataCloud

The long road to the release, from the architecture…

This is the INDIGO-DataCloud General Architecture*



INDIGO - DataCloud

*: see details in http://arxiv.org/abs/1603.09536 or in https://www.indigo-datacloud.eu/documents-deliverables

Status of the INDIGO Project

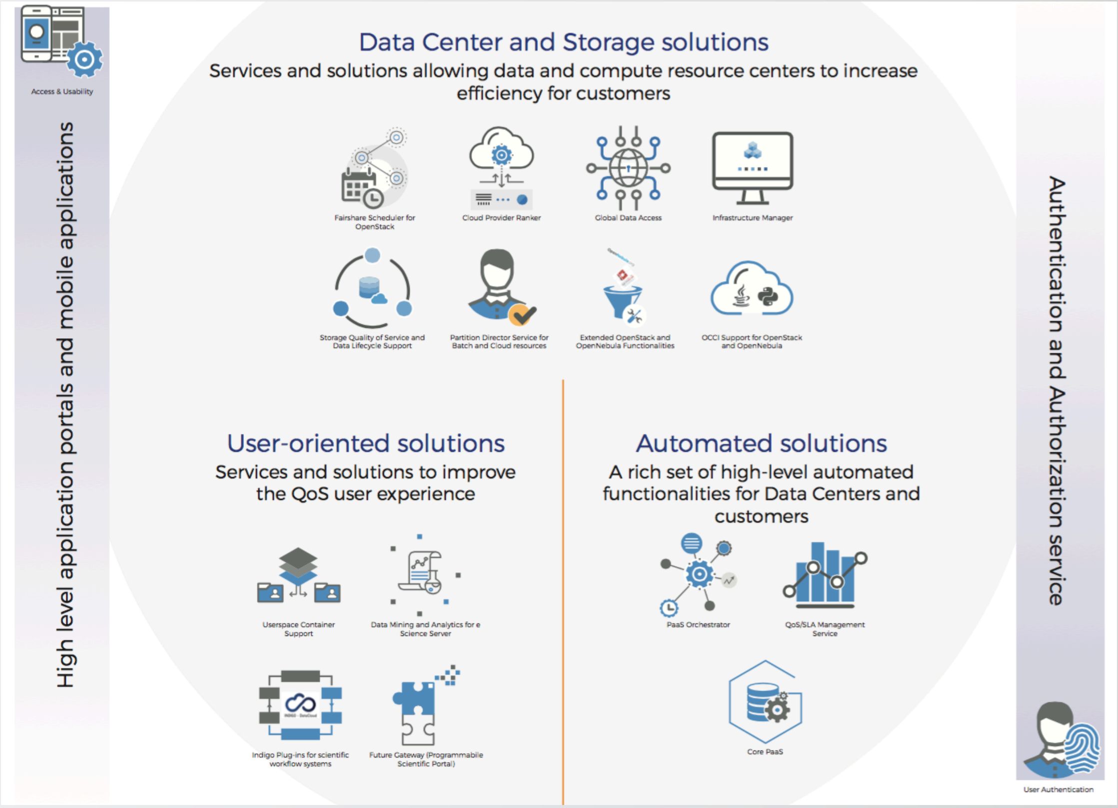
The Objective

- **To develop a simple and automated solution to create, manage and access a HTCondor cluster on cloud computing resources enabling LHC (CMS) data analysis workflow.**

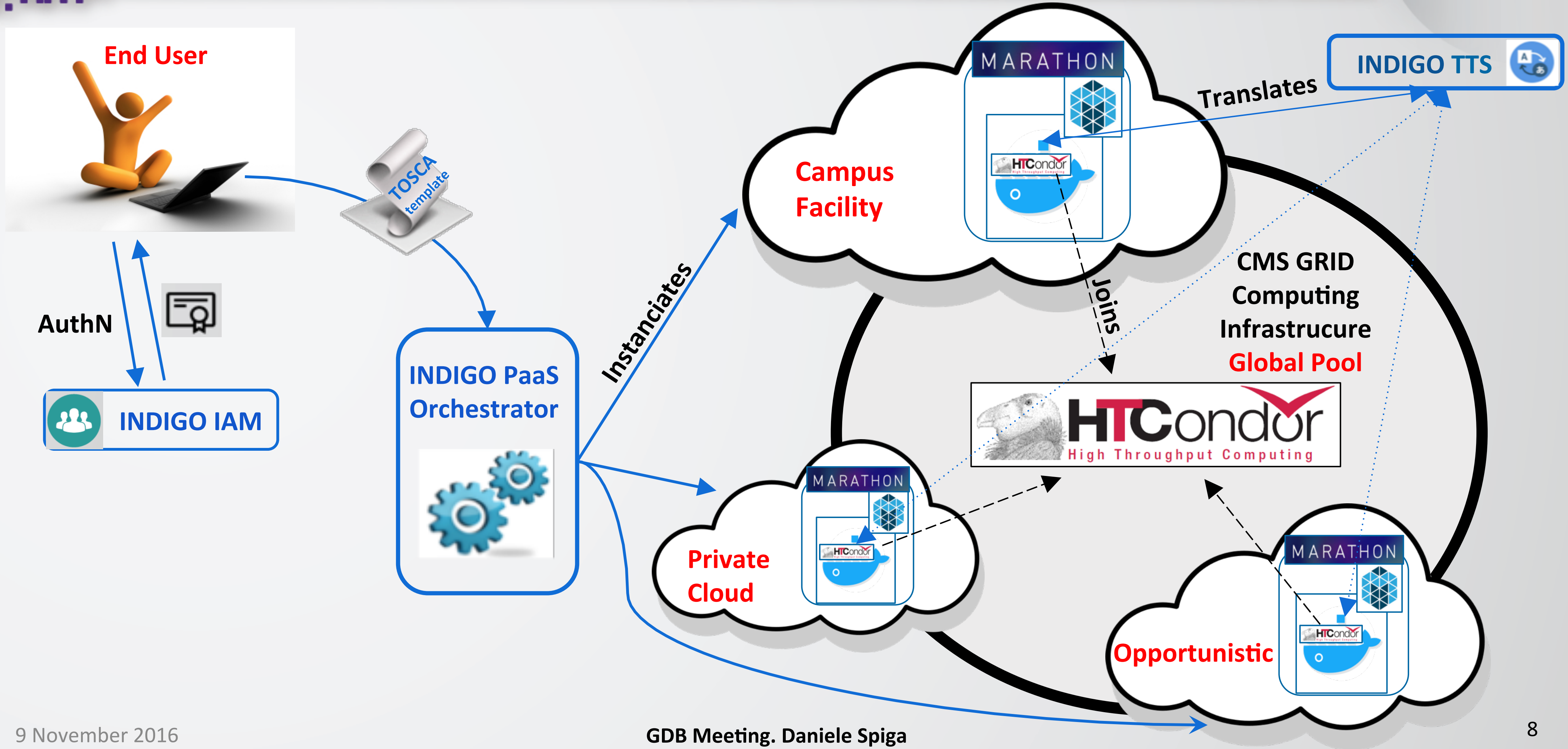
- **Advantages of a seamless integration of On-Demand opportunistic LHC/CMS computing centers:**
  - **Sites management:**
    - A simple solution for elastic computing site extensions on “opportunistic” resources
    - A easy procedure to dynamically instantiate a spot ‘ Data Analysis Facility’
  - **Users experience:**
    - Generation of a ephemeral WLCG-Tier 3 as a Service and share resources with collaborators, using standard CMS Tools (such as CRAB).
  - **Experiment-Collaboration resources:**
    - A comprehensive approach to opportunistic computing. Orchestrating multiple campus centers to gather all free CPU cycles.

INDIGO Services and solutions adopted

- **Data Center Solutions**
  - **Mesos, Marathon, CLUES**
- **Data/Storage Solution:**
  - **Dynafed, FTS , Onedata**
- **Automated Solutions**
  - **TOSCA templates, Orchestrator**
- **Common Solution**
  - **Identity Access Management (IAM), Token Translation Service (TTS)**



Solution Developed



Four Pillars

Cluster Management:

- **Mesos clusters** as a solution in order to execute docker for all the services required by a regular CMS site (Worker Node, HTCondor Schedd and squids).
- **Marathon** guarantees us the dynamic scaling up and down of resources, a key point.

AuthN/Z & Credential Management:

- **INDIGO Identity Access Management (IAM)** service is responsible for AuthN/Z to the cluster generation.
- **Token Translation Service (TTS)** enables the conversion of IAM tokens in to a X.509 certificates
  - NOTE: This allow Mesos slaves (running HTCondor_startd daemon) to join CMS central queue (HTCondor_schedd) as a regular Grid WN

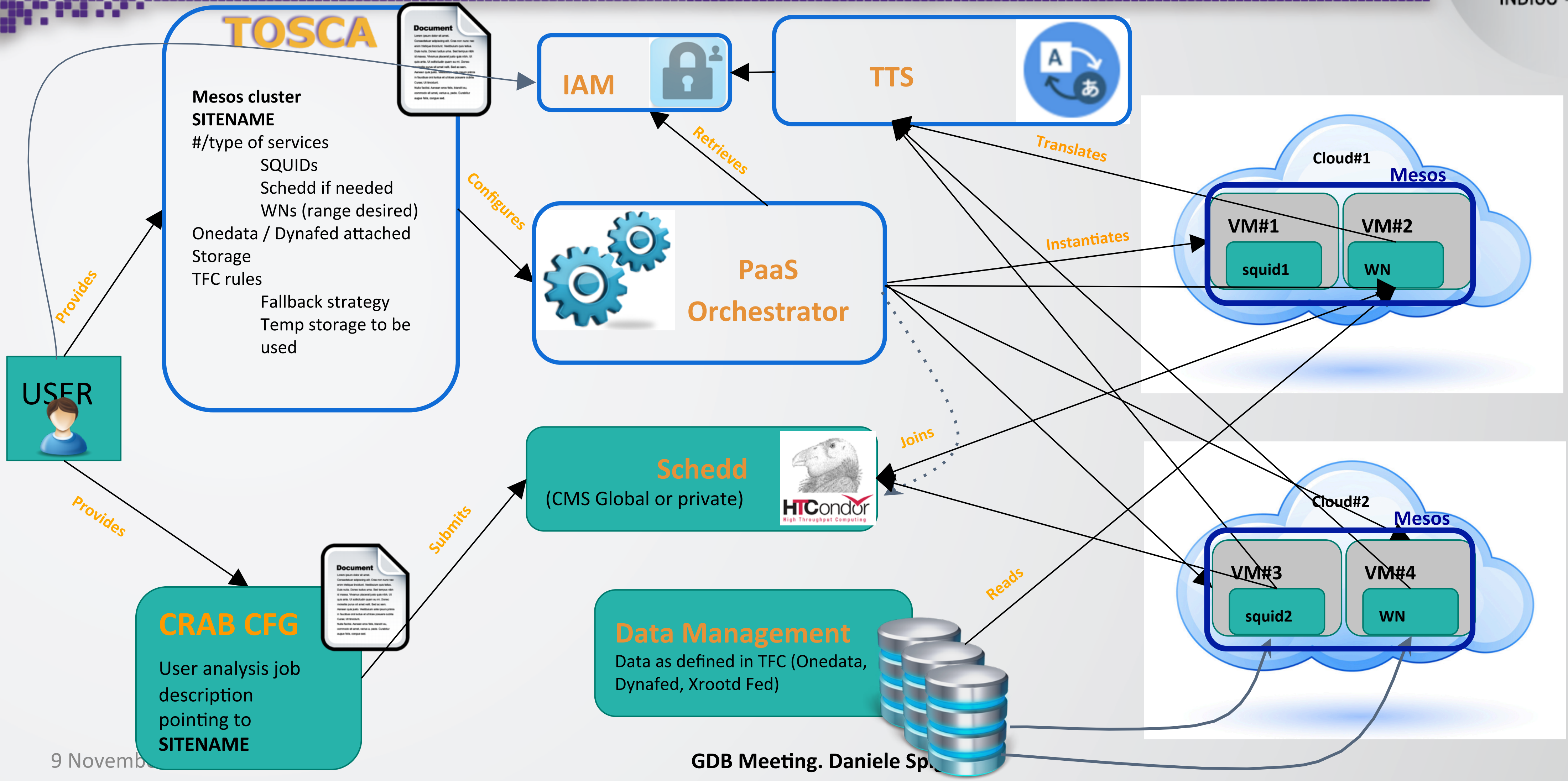
Data Management:

- **Dynafed & FTS** is the approach currently followed by the project. We will investigate **Oneclient** (from **Onedata**) as a tool allowing to mount remote Posix file-system.

Automation:

- **TOSCA templates,** meant to be managed by INDIGO PaaS **Orchestrator**, allow the automation of the overall setup.
  - The aim is to produce a single YAML file describing the setup of all required services and deps.
- **CLUES** is able to scale the Mesos cluster as needed by the load of the users jobs.

A more detailed schema



Status: The prototype… It is working

INDIGO - DataCloud

```
working_dir — bash — 139×29
get_token.sh            mesos_cluster_cms.yaml
spiga-usb0:working_dir ds$ sh get_token.sh
-ne Password:

export ORCHENT_TOKEN="eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI5RTQ5OUUwRS1DRTI1LTQ4NUItOTM1My1EMTk3MjZGRDI4N0EiLCJpc3MiOiJodHRwcz
pcL1wvaWFtLXRlc3QuaW5kaWdvLWRhdGFjbG91ZC5ldVwvIiwiZXhwIjoxNDc4MTQwMjMyLCJpYXQiOjE0NzgxMjU4MzIsImp0aSI6Ijg1YzIzNWQ5LWY2OTItNDg1ZC1iN2NhLTk4M
2NjOWI5OWUzMyJ9.Y4TgzJfDHI3FbLL_qKpzKZdxMjtTq0aESqKm0EBC1FLOlPMqIW9cuJljgZ3dpvXQ-yRkT7riJZkV2kasu7d3_p4DIaT6TMFNRAI78G-irDrueg8UyHg_vU4Wmcd
te-hbpAVRQJtKUIxHSSIPhjJx1H04Ov6oe1L8zakc14aXgd4"
spiga-usb0:working_dir ds$ export ORCHENT_TOKEN="eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI5RTQ5OUUwRS1DRTI1LTQ4NUItOTM1My1EMTk3MjZ
GRDI4N0EiLCJpc3MiOiJodHRwczpcL1wvaWFtLXRlc3QuaW5kaWdvLWRhdGFjbG91ZC5ldVwvIiwiZXhwIjoxNDc4MTQwMjMyLCJpYXQiOjE0NzgxMjU4MzIsImp0aSI6Ijg1YzIzNW
Q5LWY2OTItNDg1ZC1iN2NhLTk4M2NjOWI5OWUzMyJ9.Y4TgzJfDHI3FbLL_qKpzKZdxMjtTq0aESqKm0EBC1FLOlPMqIW9cuJljgZ3dpvXQ-yRkT7riJZkV2kasu7d3_p4DIaT6TMFN
RAI78G-irDrueg8UyHg_vU4Wmcdte-hbpAVRQJtKUIxHSSIPhjJx1H04Ov6oe1L8zakc14aXgd4"
spiga-usb0:working_dir ds$ echo $ORCHENT_URL
http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator
spiga-usb0:working_dir ds$ orchent depcreate mesos_cluster_cms.yaml '{}'
Deployment [76529d66-a813-4331-9521-31fb46a4d0bf]:
  status: CREATE_IN_PROGRESS
  creation time: 2016-11-02T22:31+0000
  update time:
  callback:
  status reason:
  outputs:
  {}
  links:
    self [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a813-43
    resources [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a8
    template [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a81

spiga-usb0:working_dir ds$
```

Deploying a CMS Mesos cluster through INDIGO PaaS Orchestrator

```
working_dir — bash — 139×29
spiga-usb0:working_dir ds$ orchent depshow 76529d66-a813-4331-9521-31fb46a4d0bf
Deployment [76529d66-a813-4331-9521-31fb46a4d0bf]:
  status: CREATE_COMPLETE
  creation time: 2016-11-02T22:31+0000
  update time: 2016-11-02T23:01+0000
  callback:
  status reason:
  outputs:
  {
      "mesos_lb_ips": [
          "90.147.170.45"
      ],
      "mesos_master_ips": [
          "90.147.170.56"
      ]
  }
  links:
    self [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a813-4331-9521-31fb46a4d0bf]
    resources [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a813-4331-9521-31fb46a4d0bf/resources]
    template [http://orchestrator01-indigo.cloud.ba.infn.it:8080/orchestrator/deployments/76529d66-a813-4331-9521-31fb46a4d0bf/template]

spiga-usb0:working_dir ds$
```

- The Mesos Cluster generation has been fully automated
- TOSCA (+Ansible) INDIGO template for a Mesos cluster:
  - Squid proxy (docker), CVMFS setup, WN (docker), proxy manager service (docker)

… A real CMS Analysis Workflow

INDIGO - DataCloud

ds — spiga@lxplus037:~ — ssh — 124×26

```
drwxr-xr-x. 8 spiga zh 2048 Nov  2 08:33 crab_projects_demo
drwxr-xr-x. 3 spiga zh 2048 Nov  2 16:58 _my_utils
-rw-r--r--. 1 spiga zh  538 Nov  2 16:57 pset_my_analysis.py
-rw-r--r--. 1 spiga zh  853 Oct 21 09:03 pset_my_analysis.pyc
bash-4.1$ vim pset_my_analysis.py
bash-4.1$ vim crabConfig.py
bash-4.1$ crab submit
Enter GRID pass phrase for this identity:
Contacting voms2.cern.ch:15002 [/DC=ch/DC=cern/OU=computers/CN=voms2.cern.ch] "cms"...
Remote VOMS server contacted succesfully.


Created proxy in /tmp/x509up_u16858.

Your proxy is valid until Thu Nov 03 23:39:20 CET 2016
Will use CRAB configuration file crabConfig.py
Importing CMSSW configuration pset_my_analysis.py
Finished importing CMSSW configuration pset_my_analysis.py
Sending the request to the server
Success: Your task has been delivered to the CRAB3 server.
Task name: 161102_223933:spiga_crab_demo_wf_1
Please use 'crab status' to check how the submission process proceeds.
Log file is /afs/cern.ch/work/s/spiga/CRAB3-tutorial/CMSSW_7_3_5_patch2/src/INDIGO/crab_projects_demo/crab_demo_wf_1/crab.lo
g
bash-4.1$ crab status
```

A Regular CRAB Job Submission

condor_q

ds — spiga@lxplus037:~ — ssh — 126×26

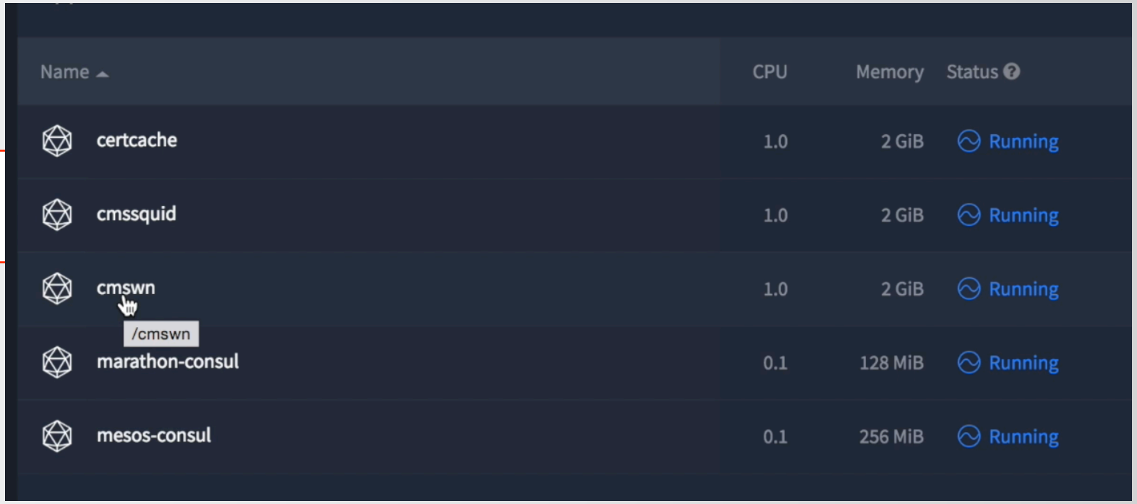
```
: 1019090   2/16 22:29    2554      _      _  10000 0.0
: 1039082   3/12 19:28    1981      _      _   2000 0.0
: 1041206   3/15 04:49    1627      _      _   2000 0.0
: 1470235  10/5  11:08    9512      _      _  10000 0.0
: 1480403  10/7  14:26    9797      _      _  10000 0.0
: 1490603  10/9  20:22    9792      _      _  10000 0.0
: 1500971  10/12 14:36    9771      _      _  10000 0.0
: 1511195  10/14 19:08       _      _      _     18 0.0
: 1511530  10/14 22:41    9856      _      _  10000 0.0
: 1520990  10/16 19:24    9826      _      _  10000 0.0
: 1531787  10/19 01:51    9854      _      _  10000 0.0
: 1534017  10/19 14:10       _      _      _     18 0.0
: 1538314  10/20 17:07       _      _      _     18 0.0
: 1540615  10/21 09:03       _      _      _     18 0.0
: 1541346  10/21 13:39       _      _      _     18 0.0
cms005 DAG: 1542290  10/21 18:27    9845      _      _  10000 0.0
crab3  DAG: 1551305  10/23 20:49       _      _      _     18 0.0
cms005 DAG: 1552817  10/24 02:51    9722      _      _  10000 0.0
cms005 DAG: 1563451  10/29 00:35    9801      1      5  10000 1563457.0 ... 1567056.0
cms005 DAG: 1573626  10/30 23:16    9844      4      5  10000 1573629.0 ... 1585895.0
cms005 DAG: 1584249  11/1  22:21    3834    639   1001  10000 1584252.0 ... 1590137.0
crab3  DAG: 1586618  11/2  08:34       _      1     17     18 1586620.0 ... 1586637.0
crab3  DAG: 1590109  11/2  23:40       _      _     18     18 1590115.0 ... 1590132.0

1719 jobs; 0 completed, 0 removed, 1054 idle, 656 running, 9 held, 0 suspended
bash-4.1$
```

Behind the scenes

- Worker nodes (docker) are running

```
root@mesos-s2:/home/ubuntu# docker ps | grep cmswn
76e8f6335228        spiga/cmswndemo                     "/bin/sh -c /root/lau"   42 minutes ago
d54ad839a8f3        spiga/cmswndemo                     "/bin/sh -c /root/lau"   49 minutes ago
root@mesos-s2:/home/ubuntu#
```

Marathon Apps Monitoring



- And executing CMS Analysis Payload

```
root@mesos-s2:/home/ubuntu# docker exec 76e8f6335228 ps auxf
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      4808  0.0  0.0  13364   996 ?        Rs   20:05   0:00 ps auxf
root         1  0.0  0.0  11356  1372 ?        Ss   19:34   0:00 /bin/bash /root/launchAndrew_spiga.sh
root       395  0.0  0.0  44072  1380 ?        S    19:35   0:00 su - glidein_pilot -c /home/glidein_pilot/runWithAndrew.sh
502        396  0.0  0.0 106112  1400 ?        Ss   19:35   0:00  \_ /bin/bash /home/glidein_pilot/runWithAndrew.sh
502        415  0.0  0.0 106508  1876 ?        S    19:35   0:00      \_ /bin/bash ./glidein_startup.sh -v std -name v3_2_11_2 -entry T3_IT_Opportunistic
fwz.cfg -descriptentry description.g6j8cz.cfg -dir . -param_GLIDEIN_Client frontend_service-v3_2_7.main -slotslayout partitionable -clientweb http://lcggw
uk:8319/vofrontend/stage/frontend_frontend_service-v3_2_7/group_main -clientsigngroup 988b368247cf8e1fd20c8bc9f8e5ffe47e1d4fea -clientdescriptgroup descri
OR_OS default -param_GLIDEIN_Collector lcggwms02.dot,gridpp.dot,rl.dot,ac.dot,uk.colon,9619.minus,9623
502       3927  0.0  0.0   9384  1476 ?        S    19:35   0:00          \_ /bin/bash /home/glidein_pilot/glide_Jwh0o2/main/condor_startup.sh glidein_con
502       4509  0.0  0.1  95948  9608 ?        S    19:35   0:00              \_ /home/glidein_pilot/glide_Jwh0o2/main/condor/sbin/condor_master -f -pidfi
502       4511  0.0  0.0  20328  2124 ?        S    19:35   0:00                  \_ condor_procd -A /home/glidein_pilot/glide_Jwh0o2/log/procd_address -L
502       4512  0.0  0.1  96628 10372 ?        S    19:35   0:00                  \_ condor_startd -f
502       4524  0.0  0.1  96612  9292 ?        S    19:50   0:00                      \_ condor_starter -f lcggwms02.gridpp.rl.ac.uk
502       4528  0.0  0.0   9252  1396 ?        S    19:50   0:00                          \_ /bin/bash /home/glidein_pilot/glide_Jwh0o2/execute/dir_4524/c
alse --firstEvent=None --firstLumi=None --lastEvent=None --firstRun=None --seeding=AutomaticSeeding --scriptExe=None --eventsPerLumi=None --scriptArgs=[]
502       4561  0.0  0.0   9256  1428 ?        S    19:50   0:00                              \_ sh ./CMSRunAnalysis.sh -a sandbox.tar.gz --sourceURL=http
nt=None --firstRun=None --seeding=AutomaticSeeding --scriptExe=None --eventsPerLumi=None --scriptArgs=[] -o {} --oneEventMode=0
502       4587  0.0  0.1  51636 13324 ?        S    19:50   0:00                                  \_ python CMSRunAnalysis.py -r /home/glidein_pilot/glide
putFiles=False --firstEvent=None --firstLumi=None --lastEvent=None --firstRun=None --seeding=AutomaticSeeding --scriptExe=None --eventsPerLumi=None --scri
502       4730  0.0  0.0   9532  1592 ?        S    19:50   0:00                                      \_ /bin/bash /home/glidein_pilot/glide_Jwh0o2/execut
502       4769  7.9  4.9 721236 401252 ?       Sl   19:50   1:09                                          \_ cmsRun -j FrameworkJobReport.xml PSet.py
root@mesos-s2:/home/ubuntu#
```

About AuthN/Z

- The Mesos cluster generation requires a valid IAM access token (Openid Connect)
  - All the rest is handled transparently to the user thanks to:
    - Delegation of credential among services
    - Credential Renewal
      - (through *token exchange* feature)
    - Token Translation Service
      - Creates credential for services not integrating OIDC

- We generate our cluster passing IAM token as incoming credential, and our WN (condor startd) authN with schedD using the following proxy:

```
root@mesos-s2:/var/log# docker exec 76e8f6335228 su - glidein_pilot -c " grid-proxy-info -file /home/glidein_pilot/gwms_proxy"
subject  : /C=EU/O=INDIGO/OU=TTS/CN=9E499E0E-CE25-485B-9353-D19726FD287A@indigo-iam-test/CN=889915561
issuer   : /C=EU/O=INDIGO/OU=TTS/CN=9E499E0E-CE25-485B-9353-D19726FD287A@indigo-iam-test
identity : /C=EU/O=INDIGO/OU=TTS/CN=9E499E0E-CE25-485B-9353-D19726FD287A@indigo-iam-test
type     : RFC 3820 compliant impersonation proxy
strength : 1024 bits
path     : /home/glidein_pilot/gwms_proxy
timeleft : 91:14:15  (3.8 days)
root@mesos-s2:/var/log#
```

Few words on Data Ingestion

- The current implementation of the use case relies on the storage solution already adopted by CMS Computing model:
  - Job input data is accessed through **xrootd data federation**
  - Job produced output go through “gfal-cp”

- **As next step:**
  - The plan for the second phase of the use case development foresees the usage of INDIGO data management solutions, **Onedata, Dynafed and FTS**

Next Steps

- ❑ Exploiting INDIGO Data Management: Onedata
- ❑ Multi IaaS Interoperability
- ❑ Testing on commercial provider: Microsoft Azure
- ❑ Extending the cluster service with a local schedD
- ❑ Evaluate HTCondor flocking mechanism
- ❑ Dynamic auto scaling of cluster : CLUES

Conclusion

- ✓ We developed a prototype for generating a "Tier* ephemeral site as a service"
- ✓ We enabled the complete automation of the cluster generation
- ✓ We integrated the INDIGO credential harmonization
- ✓ We successfully execute CMS analysis jobs
- ➢ By simplifying and automating the process of creating, managing and accessing a pool of computing resources the project aims to impact on:
  - ✓ Sites, Users and Experiment collaboration

Although the presented prototype is focused on CMS workflows, the adopted approach can be easily generalized e.g. to whoever can integrate HTCondor

Thank you

https://www.indigo-datacloud.eu

Better Software for Better Science.