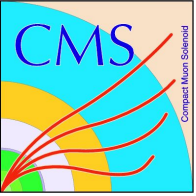# The CMS Global pool, from a pilot-based to an heterogeneous mix of resources

September, 2018

Diego Dávila
on behalf of the Submission Infrastructure team of the CMS experiment.
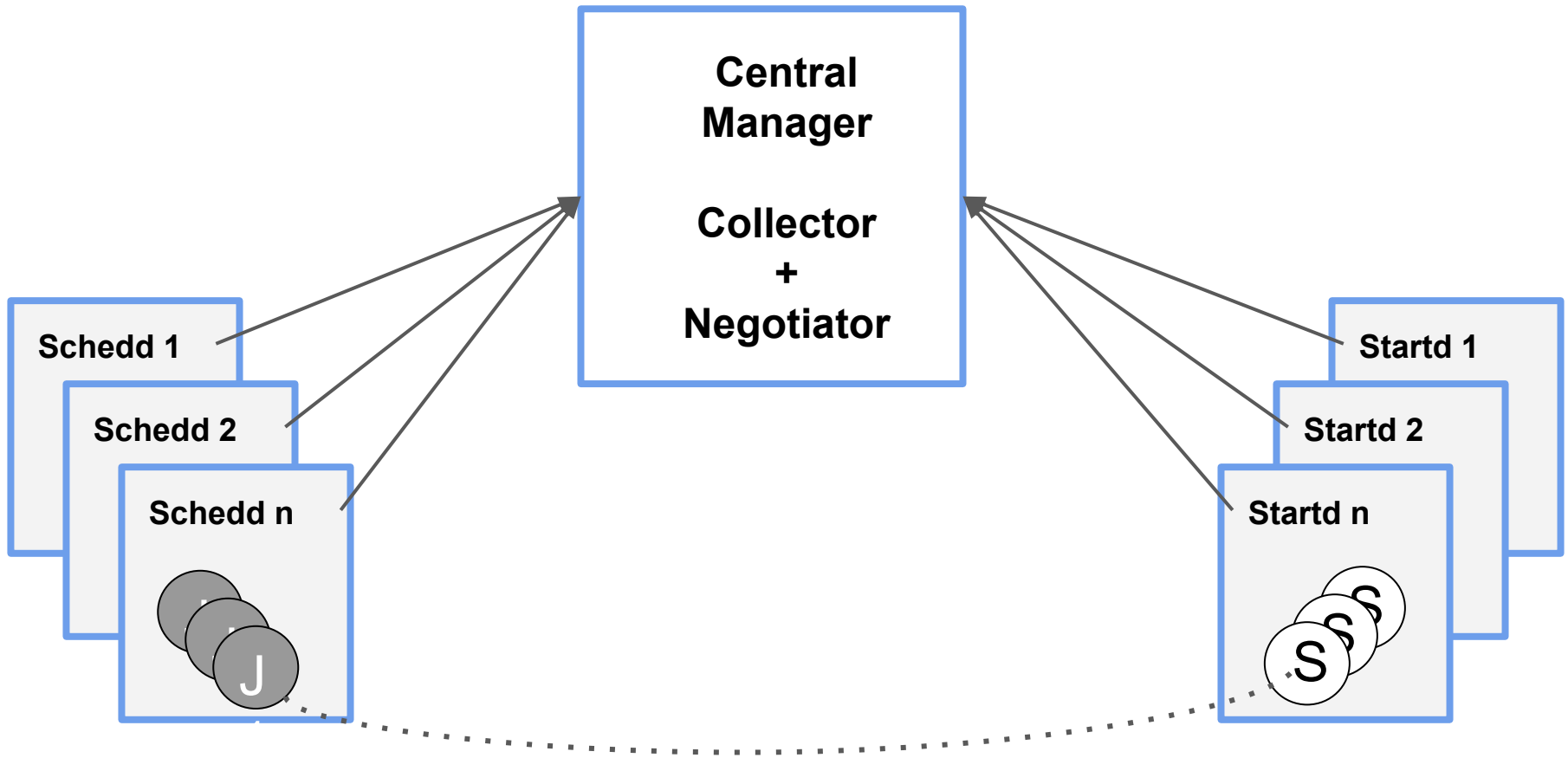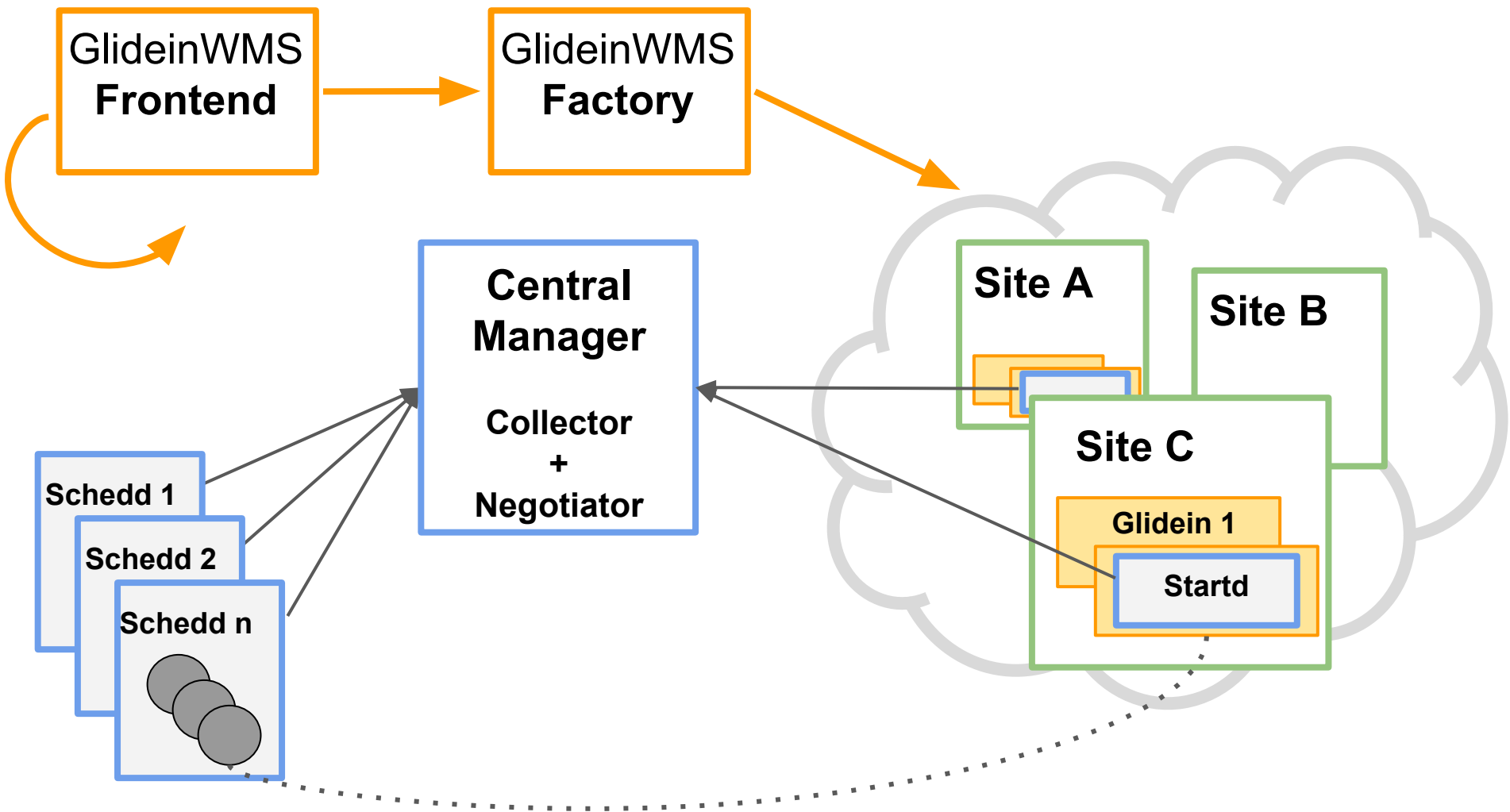
Outline

1. The pilot model of the Global pool
2. Federated pools
3. DODAS: Site launched startds
4. CMS@HOME: Volunteer computing
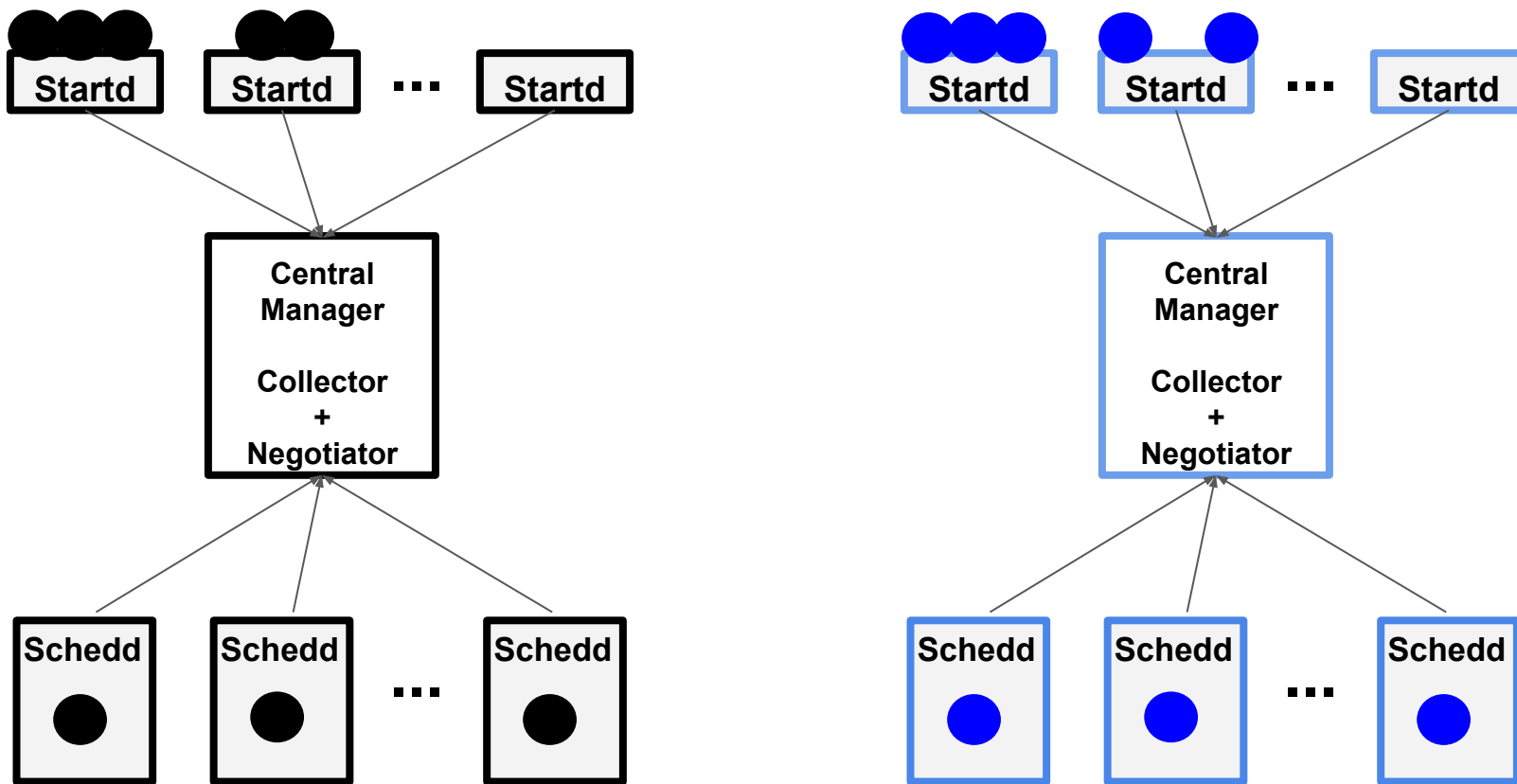
The pilot model of the Global pool
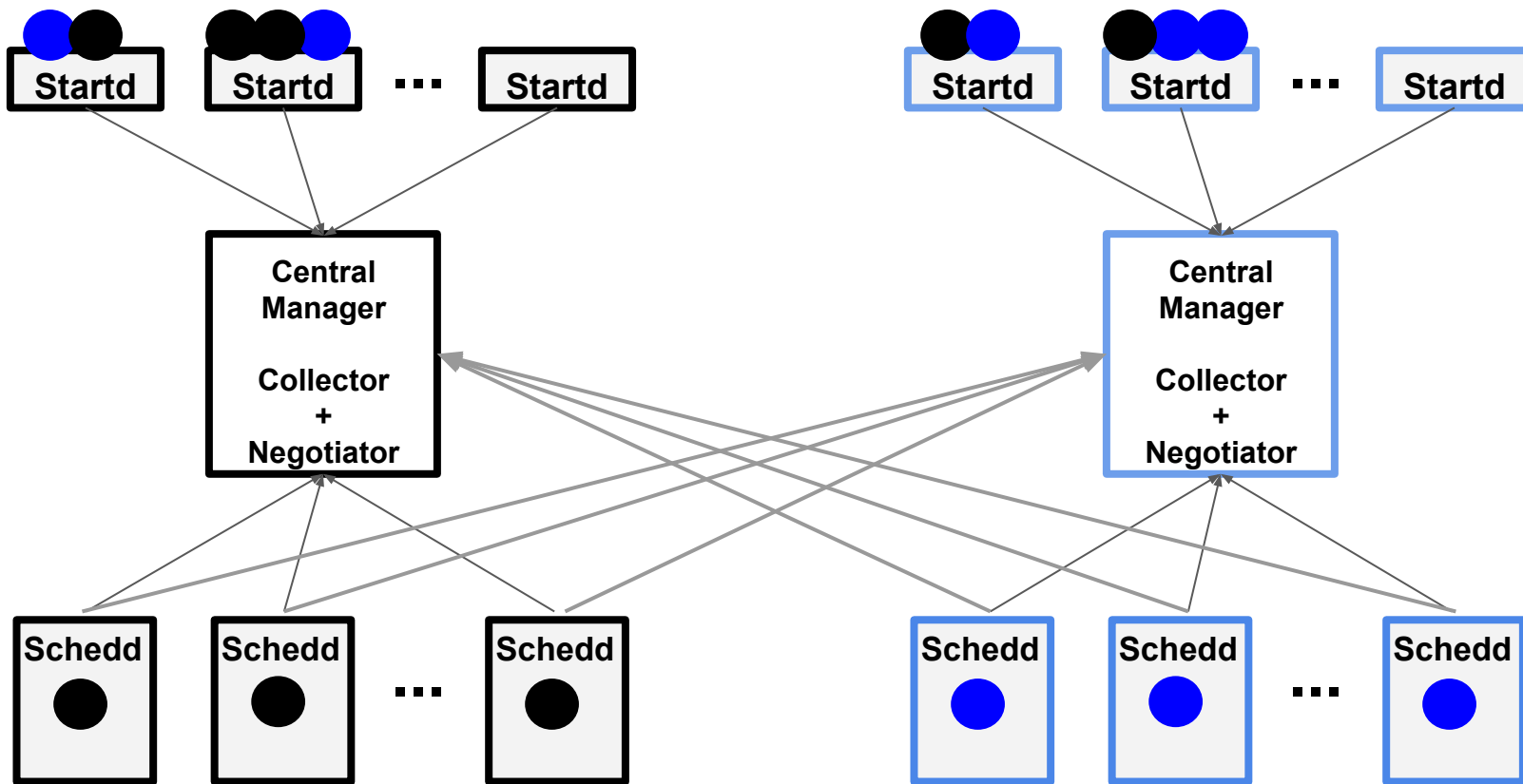
The pilot model of the Global pool

Federated pools

Instead of adding startds to our pool (pilot model)
we flock our scheds to other pools that already have startds.

Federated pools

Instead of adding startds to our pool (pilot model)
we flock our scheds to other pools that already have startds.
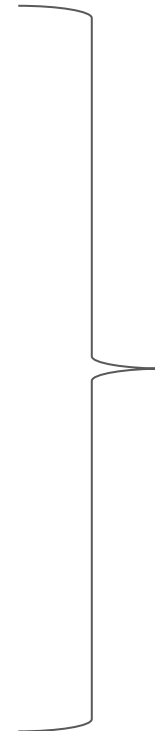
Federated pools

CERN pool
(8 schedds) →

| T0 Negotiator |

50K cores

Global pool
(44 schedds)

| T1 Negotiator |

| US T2 Negotiator |

| Non-US T2 Negotiator |
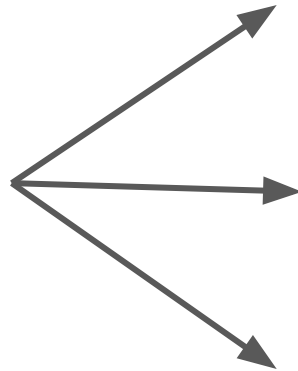
250K cores

Federated pools

CERN pool
(8 schedds)

Global pool
(44 schedds)

T0 Negotiator

T1 Negotiator
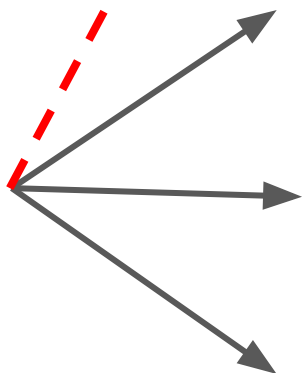
US T2 Negotiator

Non-US T2 Negotiator

50K cores
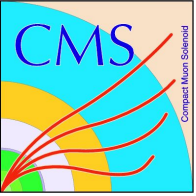
200K cores

Federated pools

1. The more the negotiators, the longer the time schedds spent negotiating
   a. Slow schedds
      i. Priority inversion

2. The more the schedds to serve, the longer the negotiation time
   a. Negotiation time increases
      i. Matching to recently disconnected resources
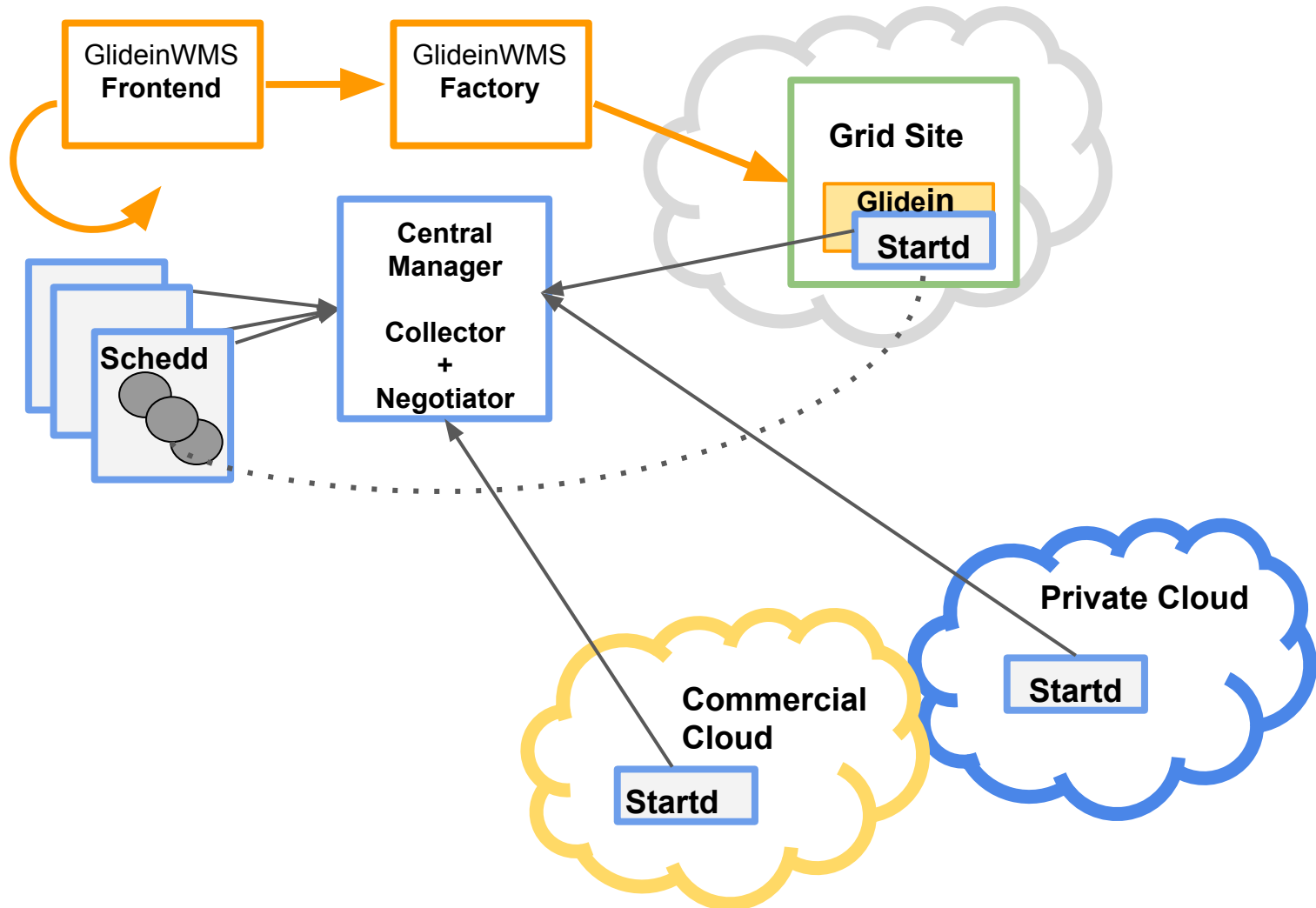
DODAS: Site launched startds

What it is?

Dynamic **O**n **D**emand **A**nalysis **S**ervice

A way to provisioning resources in the cloud either by launching a standalone HTCondor pool or <u>HTCondor startds to join an existing pool (CMS case)</u> on any cloud provider with almost zero requirements [1]

Provisioning and Usage is tightly coupled

DODAS: Site launched startds

DODAS: the concern

In the pilot model, the startd advertise a custom attribute showing the site to which it belongs and use it to match the users jobs.

Since we don't have control over the configuration of DODAS' startds, how could we trust they are coming from the site they advertise?

User's job

| RequestCpus=8 |
| RequestMemory=16 |
| |
| **+DESIRED_Sites="** |
| **T2_CH_CERN"** |

matchmaking

Site: T2_CH_CERN

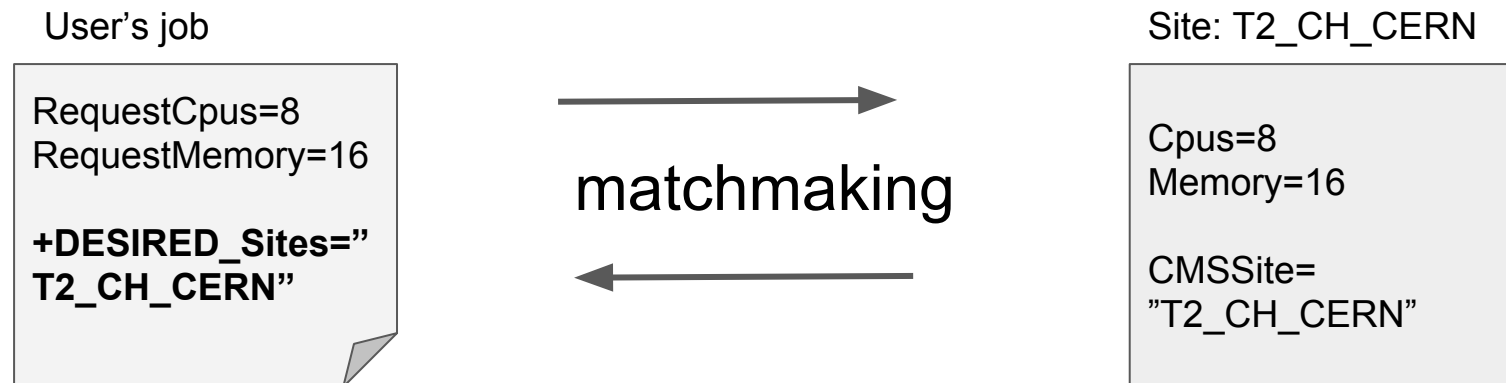| Cpus=8 |
| Memory=16 |
| |
| CMSSite= |
| "T2_CH_CERN" |

DODAS: the concern

In the pilot model, the startd advertise a custom attribute showing the site to which it belongs and use it to match the users jobs.
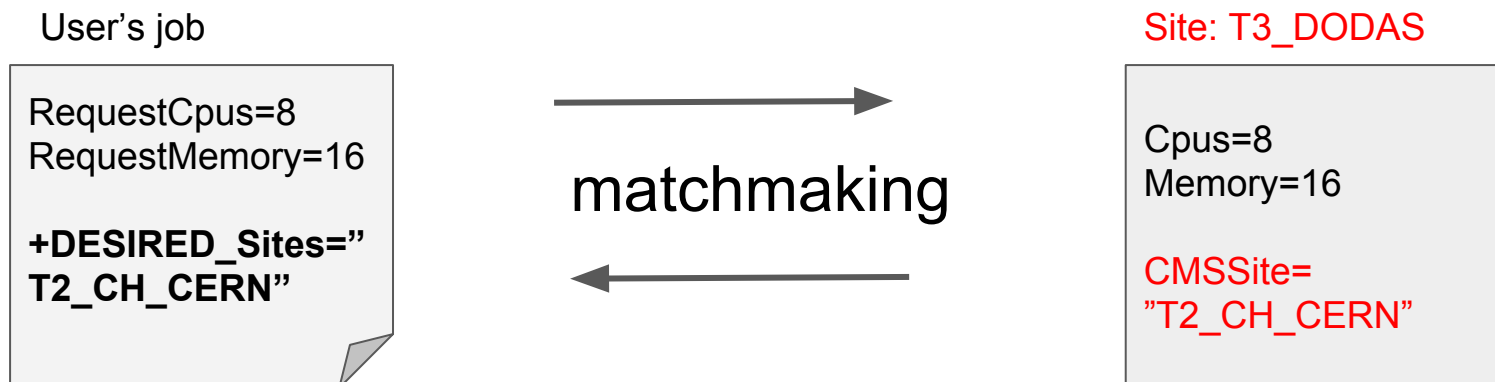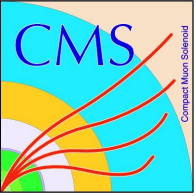
Since <u>we don't have control over the configuration of DODAS' startds</u>, how could we trust they are coming from the site they advertise?

User's job

Site: T3_DODAS

```
RequestCpus=8
RequestMemory=16

+DESIRED_Sites="
T2_CH_CERN"
```

matchmaking

```
Cpus=8
Memory=16

CMSSite=
"T2_CH_CERN"
```

DODAS: the need

1. We need 2 layers in the authentication mechanism:
   a. Similarly to the pilot model, the DODAS startd needs to present a valid certificate that must be whitelisted in the Central Manager.
   b. <u>The certificate and the custom attribute CMSSite are a valid combination</u>, e.g.:
      i. certificate 'x' can be used with CMSSite "A" or "B"
      ii. certificate 'y' can be used with CMSSite "C"
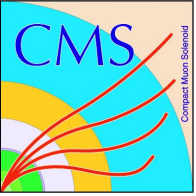
   How we do that in condor?

DODAS: the solution

"There is a knob for that"

DODAS: the solution

… well there are actually 3 knobs:

1. CERTIFICATE_MAPFILE

2. CLASSAD_USER_MAPFILE

3. COLLECTOR_REQUIREMENTS

DODAS: the solution

CERTIFICATE_MAPFILE

GSI   "The DN x "   dodas_x
GSI   "The DN y "   dodas_y
…

GSI   "The DN z "   dodas_z
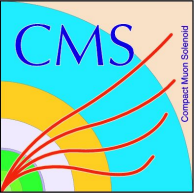
dodas_mapfile.map

*   dodas_x   "A, B"
*   dodas_y   "C"

…

* dodas_z "D,F,G"
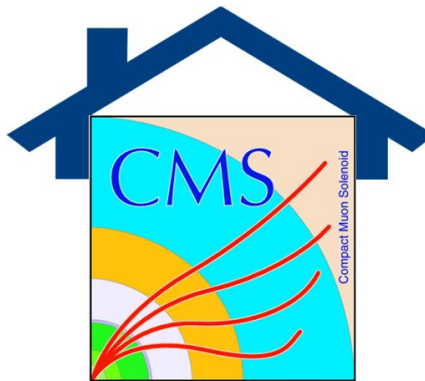
CLASSAD_USER_MAPFILE_**dodas**= dodas_mapfile.map

COLLECTOR_REQUIREMENTS:

CMSSite =?= userMap("dodas", AuthenticatedIdentity)

Volunteer computing

Volunteer computing

CMS@HOME

- The idea is to use volunteer machines to run low priority production workflows
- Tasks run in a VM, and obtain jobs from HTCondor
- Technically it's very similar to DODAS: an independent startd that joins the pool

Volunteer computing

CMS@HOME

- The idea is to use volunteer machines to run low priority production workflows
- Tasks run in a VM, and obtain jobs from HTCondor
- Technically it's very similar to DODAS: an independent startd that joins the pool
- Good news: anyone can provide resources!

Volunteer computing

CMS@HOME

- The idea is to use volunteer machines to run low priority production workflows
- Tasks run in a VM, and obtain jobs from HTCondor
- Technically it's very similar to DODAS: an independent startd that joins the pool
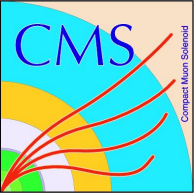- Good news: anyone can provide resources!
- Bad news: **anyone** can provide resources

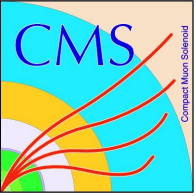*"volunteers are volunteers ... no commitment, no trust "*

Volunteer computing

CMS@HOME

- The project is still on testing phase. It will be soon joined to our infrastructure.
- We are in the process of evaluating:
  - whether we should add this independent startds or as a federated pool.
  - Security constraints
- Hopefully, we will find a condor trick to adapt this project to our needs

Conclusions

- With the HL-LHC on the horizon, we need to be able to make use of all available compute power.

- There are many resources out there that don't fit with our pilot model and it is our job to be able to integrate them.

- Federated pools seems to be a good alternative to join different compute sources and at the same time grow horizontally but there are few scalability issues that need to be addressed first.

- Condor's flexibility has been key when it comes to merge heterogeneous resources under a single pool

Questions?

References

1. Daniele Spiga, *DODAS: How to effectively exploit heterogeneous clouds for scientific computations*, International Symposium on Grids and Clouds 2018,http://indico4.twgrid.org/indico/event/4/session/19/contribution/29/material/slides/0.pdf

2. *LHC@HOME*, http://lhcathome.web.cern.ch
3. Federica Fanzago, Ivan D Reid, *CMS@HOME a short update,* https://indico.cern.ch/event/711343/contributions/2956096/attachments/1628717/2594993/CMSHome_2018_April_9th_.pdf