

# Updates on The SIMPLE Framework

Mayank Sharma (CERN)

Maarten Litmaath (CERN)

Eraldo Silva Junior (CBPF)

Renato Santana (CBPF)

# SIMPLE Grid Project: Recap



Solution for **I**nstallation **M**anagement and **P**rovisioning of **L**ightweight **E**lements.

SIMPLE is a **private PaaS** for quickly setting up **WLCG services** at your site, on demand. It can also setup **Hadoop/Spark/Jupyter** clusters and much more...

Out of the box, SIMPLE can create a **production ready site** that runs, for instance, a **HTCondorCE**, **HTCondor Batch**, **HTCondor workers** (and/or **HTCondor Submitters** if local users need to be supported).

Site admins:

- Prepare hosts (install puppet and simple\_grid\_puppet\_module)
- Supply a **single YAML based configuration** file on a **single node** (called Config Master or CM)
- Execute the framework!

SIMPLE appropriately deploys **pre-configured WLCG services** in containers, that have been **tested** to work well together.



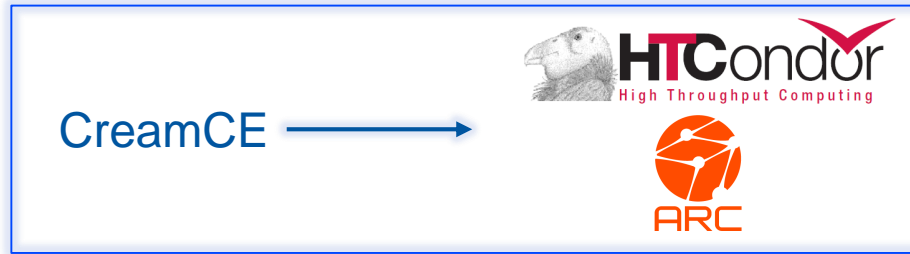
# Why SIMPLE?

- Based on **classic grid site model** used/preferred at most sites.
  - **CEs, Batch, WNs, ...**
  - Support **non-LHC VOs** with ease.
- Combines benefits of popular tools like **Puppet** with **Docker** and offers more...
  - Helps **avoid common pitfalls**.
  - Expects **basic sys-admin** know-how
  - **Significant reduction** in amount of **config-info**.
  - **Validation of configuration**
  - **Validation of infrastructure**
- Easy to update or re-instantiate services
  - **Rollback functionality**
- **Support** from the SIMPLE team.



# Use case

- A first natural use case for the SIMPLE framework is migration from CREAM-CE.



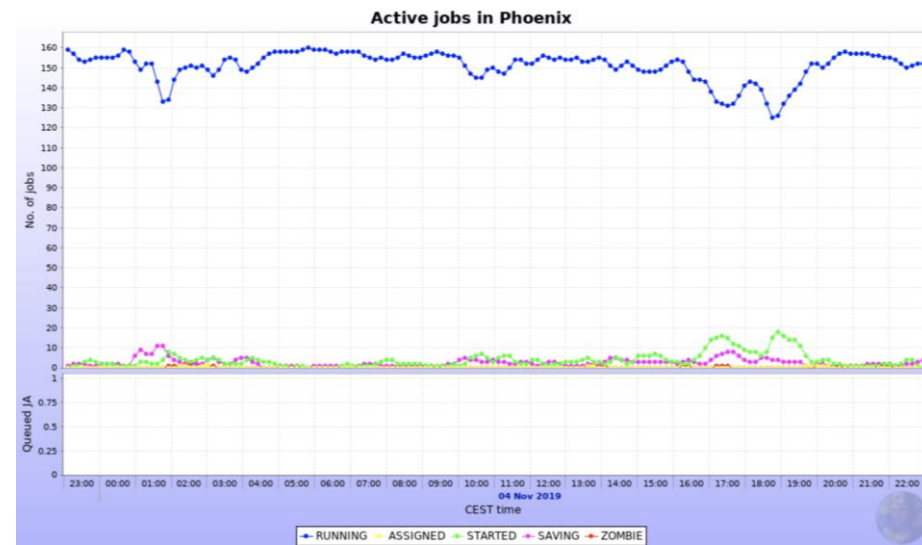
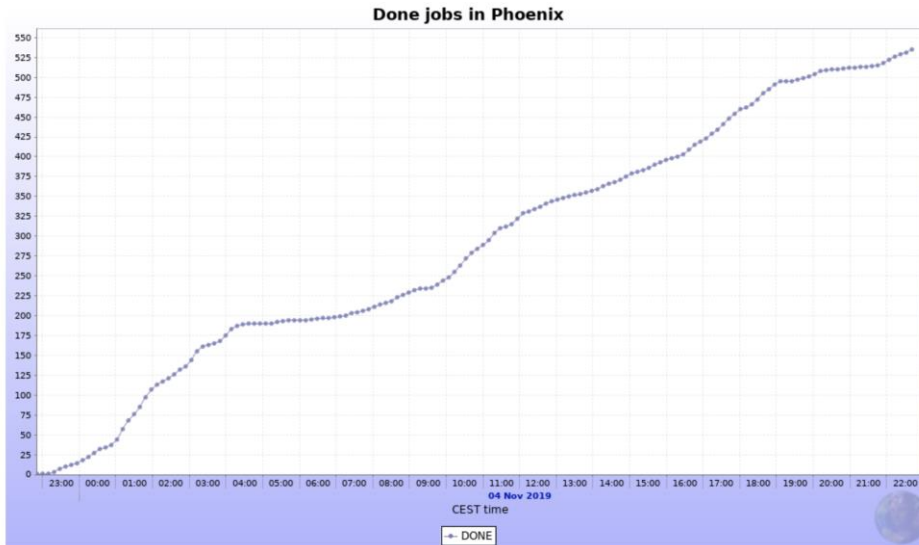
- Simplify **switching to HtCondorCE/HTCondor batch** powered site.

# SIMPLE Framework: Deployments

Centro Brasileiro de Pesquisas Físicas (CBPF, Tier-2 in Brazil)

HTCondorCE, HTCondor Batch, HTCondor workers.

\*Test site running real production jobs, 160 cores

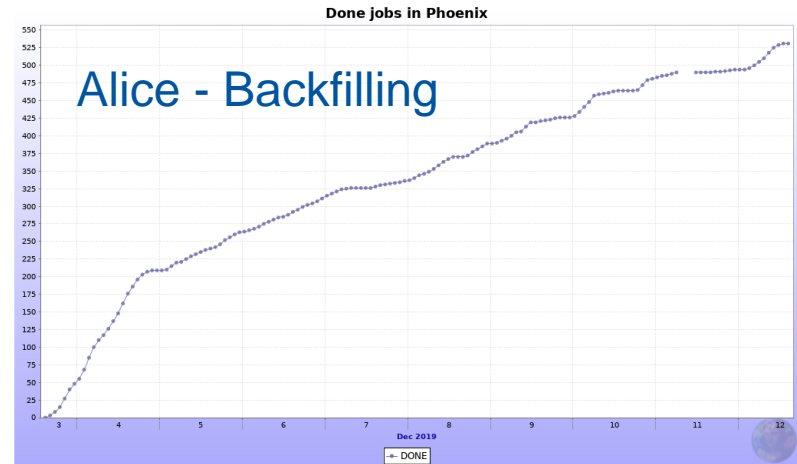
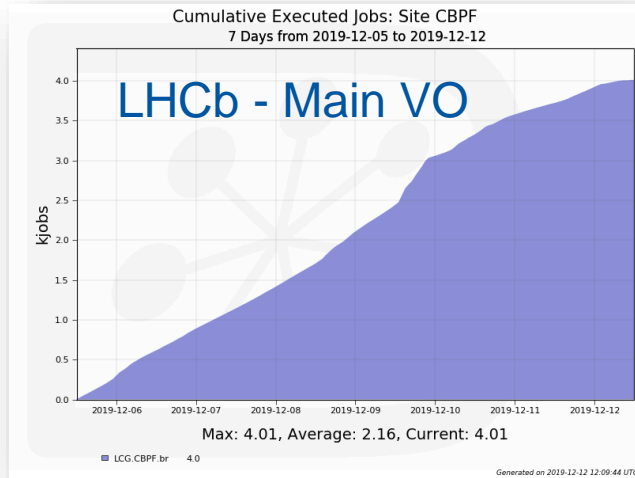


# SIMPLE Framework: Deployments

Centro Brasileiro de Pesquisas Físicas (CBPF, Tier-2 in Brazil)

HTCondorCE, HTCondor Batch, HTCondor workers.

Main Site running real production jobs, 1040 cores



```
$ condor_ce_q
```

```
Total for query: 2726 jobs; 1706 completed, 0 removed, 237 idle, 781 running, 2 held, 0 suspended  
Total for all users: 2726 jobs; 1706 completed, 0 removed, 237 idle, 781 running, 2 held, 0 suspended
```

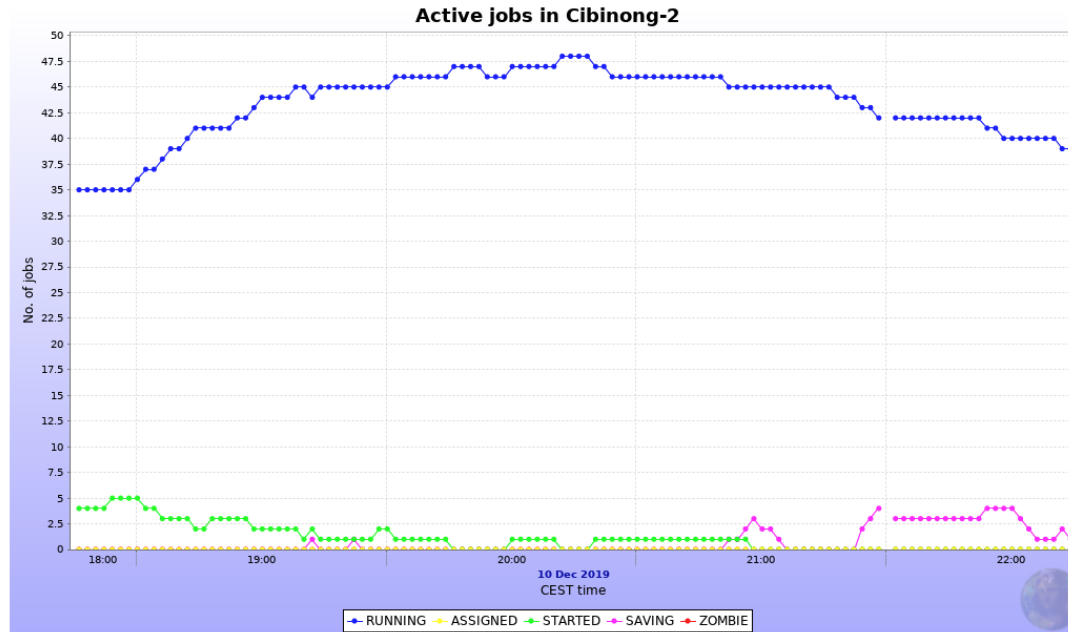


# SIMPLE Framework: Deployments

Indonesian Institute of Sciences (Cibinong, Tier-3 in Indonesia)

HTCondorCE, HTCondor Batch, HTCondor workers.

Test site running real production jobs, 56 cores



# SIMPLE Framework: Deployments

CERN

Dynamic Apache Spark Cluster for Economic Analysis

\* Mini cluster that runs Apache Spark, Hadoop, Yarn, HDFS, Jupyter Notebook frontend.

The screenshot shows the Hadoop cluster management web interface. The browser address bar indicates the URL is `http://hebesonacho-spark-hadoop-master-0.cern.ch`. The page title is "Nodes of the cluster". The Hadoop logo is visible in the top left. The main content area is titled "Nodes of the cluster" and contains several summary tables and a detailed node list.

**Cluster Metrics**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
6	0	1	5	3	4.50 GB	15 GB	0 B	3	40	0

**Cluster Nodes Metrics**

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
5	0	0	0	0	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:256, vCores:1>	<memory:3072, vCores:4>	0

**Node List Table**

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack		RUNNING	spark-hadoop-add-1-worker.cern.ch:34042	spark-hadoop-add-1-worker.cern.ch:8042	Fri Apr 19 13:25:31 +0200 2019		1	1.50 GB	1.50 GB	1	7	2.8.5
/default-rack		RUNNING	spark-hadoop-submit.cern.ch:45121	spark-hadoop-submit.cern.ch:8042	Fri Apr 19 13:25:31 +0200 2019		0	0 B	3 GB	0	8	2.8.5
/default-rack		RUNNING	spark-hadoop-add-3-worker.cern.ch:35466	spark-hadoop-add-3-worker.cern.ch:8042	Fri Apr 19 13:25:31 +0200 2019		0	0 B	3 GB	0	8	2.8.5
/default-rack		RUNNING	spark-hadoop-worker-0.cern.ch:44695	spark-hadoop-worker-0.cern.ch:8042	Fri Apr 19 13:25:31 +0200 2019		0	0 B	3 GB	0	8	2.8.5
/default-rack		RUNNING	spark-hadoop-add-2-worker.cern.ch:40365	spark-hadoop-add-2-worker.cern.ch:8042	Fri Apr 19 13:25:31 +0200 2019		2	3 GB	0 B	2	6	2.8.5

Showing 1 to 5 of 5 entries



# SIMPLE Framework: Example



Data Center

## Config Master(CM)

Centrally manage installation and configuration of grid services on the LC nodes.

simple-condor-cm



188.184.91.176

## Lightweight Component(LC)

The nodes on which grid services are deployed by the framework.

simple-condor-ce



188.185.112.251

simple-condor-batch



188.185.84.16

simple-condor-node0



188.185.78.135

simple-condor-node1



188.185.64.158

simple-condor-node2



188.185.68.214

# Simple Framework: Example



## CM node: Install puppet

```
total
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puppet5-release
Importing GPG key 0xF8D349F:
Userid : "Puppet, Inc. Release Key (Puppet, Inc. Release Key) <release@puppet.com>"
Fingerprint: 6159 9c10 e59e 6e16 9f32 7f43 8280 ef80 349f
Package : puppet5-release-5.0.0-4.el7.noarch (installed)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-puppet5-release
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : libssl-1.1.28-5.el7.x86_64
Installing : python-lxml-3.2.1-4.el7.x86_64
Installing : python-javapackages-3.4.1-11.el7.noarch
Installing : javapackages-tools-3.4.1-11.el7.noarch
Installing : tzdata-java-2019c-1.el7.noarch
```

## CM node: Install puppet module

```
[mayanksharma:~]$ ssh root@simple-condor-cm
Enter passphrase for key '/Users/mayanksharma/.ssh/id_rsa':
Last login: Thu Oct 24 10:58:14 2019 from mayank-macbook.dyndns.cern.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file
[root@simple-condor-cm ~]# puppet module install many-simple_grid
2019-10-24 11:06:00.304214 WARN puppetlabs.facter - locale environment variable: Preparing to install into /etc/puppetlabs/code/environments/production
Notice: Downloading from https://forgeapi.puppet.com ...
```

## LC nodes: Configure puppet and SIMPLE

```
[mayanksharma:~]$ ssh root@simple-condor-ce
Enter passphrase for key '/Users/mayanksharma/.ssh/id_rsa':
Last login: Fri Oct 25 18:14:54 2019 from mayank-macbook.dyndns.cern.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[root@simple-condor-ce ~]# rpm -ivh https://yum.puppetlabs.com/puppet5/puppet5-release-el-7.noarch.rpm
Retrieving https://yum.puppetlabs.com/puppet5/puppet5-release-el-7.noarch.rpm
Preparing... [#####] 100%
package puppet5-release-5.0.0-4.el7.noarch is already installed
[root@simple-condor-ce ~]# yum -y --disablerepo=cern,e17,epel install puppet-agent
Failed to set locale, defaulting to C
Loaded plugins: changelogs, fastestmirror, kernel-module, ovl, protectbase, tsflags, versionlock
Loading mirror speeds from cached hostfile
0 packages excluded due to repository protections
Package puppet-agent-5.5.17-1.el7.x86_64 already installed and latest version
Nothing to do
[root@simple-condor-ce ~]# puppet --version
2019-10-25 18:15:57.266585 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C LC_ALL=C
5.5.17
[root@simple-condor-ce ~]# puppet module install many-simple_grid
2019-10-25 18:16:09.838015 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C LC_ALL=C
```

## CM node: Sign puppet certificates

```
[mayanksharma:~]$ ssh root@simple-condor-cm
Enter passphrase for key '/Users/mayanksharma/.ssh/id_rsa':
Last login: Fri Oct 25 18:21:12 2019 from mayank-macbook.dyndns.cern.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[root@simple-condor-cm ~]# puppet cert show --all
2019-10-25 18:37:19.450803 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C
Warning: puppet cert is deprecated and will be removed in a future release.
(Location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/application.rb:378:in 'run')
Error: level method show is apply
[root@simple-condor-cm ~]# puppet cert list --all
2019-10-25 18:37:26.344173 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C
Warning: puppet cert is deprecated and will be removed in a future release.
(Location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/application.rb:378:in 'run')
simple-condor-batch.cern.ch [SHA256] 52:CD1F:19:30:53:52:10:89:38:CE:20:06:20:00:38:04:CE:4F:40:0E:47:96:8B:38:EE
*simple-condor-execute.cern.ch [SHA256] 89:AE:8F:35:FC:8B:12:AB:A7:54:E9:15:F7:37:74:62:7E:3F:04:C3:48:66:E8:DC:65:00:59
*simple-condor-submit.cern.ch [SHA256] 9C:04:97:05:20:65:33:8B:0A:47:04:1F:F8:E8:6F:28:80:15:8E:23:64:C8:64:D3:FA:8B:8F
*simple-condor-ce.cern.ch [SHA256] 35:13:08:D6:3F:DE:04:78:8E:28:33:08:BE:25:E4:83:BC:39:BA:85:94:28:C0:09:79:80:44
*simple-condor-cm.cern.ch [SHA256] 82:41:A4:24:93:83:84:82:29:48:02:76:78:3A:CD:64:7E:54:A6:13:33:0F:87:82:72:54:81
*simple-condor-ce.cern.ch [SHA256] 82:41:A4:24:93:83:84:82:29:48:02:76:78:3A:CD:64:7E:54:A6:13:33:0F:87:82:72:54:81
[root@simple-condor-cm ~]# puppet c
```



## CM node: Configure puppet and SIMPLE

```
[mayanksharma:~]$ ssh root@simple-condor-cm
Enter passphrase for key '/Users/mayanksharma/.ssh/id_rsa':
Last login: Fri Oct 25 16:06:49 2019 from mayank-macbook.dyndns.cern.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[root@simple-condor-cm ~]# puppet apply -e 'class { 'simple_grid': install::config_master::simple_installer; }'
2019-10-25 18:02:21.478139 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C LC_ALL=C
Warning: /etc/puppetlabs/puppet/hiera.yaml: Use of 'hiera.yaml' version 3 is deprecated. It should be converted to version 5
Warning: This method is deprecated, please use the stdlib validate_legacy function, with Pattern[]: There is further documentation for validate_legacy function in the README. at ["etc/puppetlabs/code/environments/production/modules/python/main/manifests/init.pp", 182]:
(Location: /etc/puppetlabs/code/environments/production/modules/stdlib/lib/puppet/functions/deprecation.rb:28:in 'deprecation')
Notice: Compiled catalog for simple-condor-cm.cern.ch in environment production in 0.78 seconds
Notice: Creating simple config directory
Notice: /Stage[main]/Simple_grid::Install::Config_master::Simple_installer/Notify[Creating simple config directory]/message: de
fined 'message' as 'Creating simple config directory'
Notice: Setting node type via file /etc/simple_grid/node_type
```



\* Click on the images to see the terminal captures

# SIMPLE Framework: Example



## Example 1

### Current Example - Site Level Configuration File

## Example 2

### Cibinong Test cluster - Site Level Configuration File

# SIMPLE Framework: Example



- Execute the framework

```
[root@simple-cm ~]# puppet agent -t
```

```
[root@simple-condor-cm simple_grid]# puppet agent -t
Info: Using configured environment 'simple'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Loading facts
Info: Caching catalog for simple-condor-cm.cern.ch
Info: Applying configuration version '1572702641'
Notice: /Stage[main]/Simple_grid::Nodes::Config_master::Init/Simple_grid::Components::Execution_stage_manager::Set_stage[
loy]/File[Updating Stage to pre_deploy]/content:
--- /etc/simple_grid/.stage      2019-11-02 14:48:28.481046657 +0100
+++ /tmp/puppet-file20191102-16462-fau8mc      2019-11-02 14:50:41.847197652 +0100
@@ -1 +1 @@
-config
\ No newline at end of file
+pre_deploy
\ No newline at end of file
```

\* [Click on the image to see the terminal capture](#)

# SIMPLE Framework: Example



- The HTCondor pool is ready!

## HTCondorCE

```
-- Schedd: simple-condor-ce.cern.ch : <10.0.0.10:8767> @ 11/02/19 15:39:45
OWNER BATCH_NAME      SUBMITTED  DONE  RUN  IDLE  HOLD  TOTAL JOB_IDS
-----
Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
Total for all users: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended

sh-4.2# condor_ce_q

-- Schedd: simple-condor-ce.cern.ch : <10.0.0.10:8767> @ 11/02/19 15:39:57
OWNER BATCH_NAME      SUBMITTED  DONE  RUN  IDLE  TOTAL JOB_IDS
-----
simple ID: 7           11/2  15:39  _     _     1     1 7.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
```

## HTCondor Submit Node

```
universe           = grid
executable         = sleep.sh
log                = sleep.log
output             = outfile.txt
error              = errors.txt
should_transfer_files = Yes
when_to_transfer_output = ON_EXIT
use_x509userproxy = true
+WantJobRouter    = true
+TransferOutput   = ""
grid_resource      = condor simple-condor-ce.cern.ch simple-condor-ce.cern.ch:9619
queue
[condor_user@simple-lc02 sleep_job]$ condor_submit sleep_simple_condor_ce.sub
```

\* Click on the images to see the terminal captures

# SIMPLE Framework: Example



- Summing up:
  - Install puppet and **simple grid puppet module** on all nodes.
  - Write a **site-level-config-file.yaml**.
  - Execute the **SIMPLE framework**.
- Getting Started Guide

# Community Driven!

- Open Source community!
- Looking for:
  - **Site admins** who wish to try out and/or beta test creating HTCondorCE/HTCondor Batch sites with the SIMPLE framework.
  - **ARC/Slurm experts** to help support these grid services through SIMPLE.

# What's Ongoing?

- Support for hosts with **multiple network interfaces**.
- **HTCondor Submit nodes** for direct submission to the batch system.
- New CE container with **Monitoring (grafana), APEL and fair share** functionality.
- SIMPLE **Command Line Interface** for managing the entire installation process.
- Online site level configuration file compiler for **assisting admins in writing their configuration**.





# What's next?

- Upcoming Component repositories:
  - **Squid**
  - **ARC** and **SLURM**
  - ...
- **RedHat Rundeck** web interface for using the framework (real-time deployment monitoring, get email notifications)
- Support for **Kubernetes** in addition to Docker-Swarm
- Support for **Ansible** in addition to Puppet.
- Request support for grid services/ features/ bug report: [GitHub Project](#)



# Communication channels

**Website:** <https://simple-framework.github.io>

**Slack Channel:** [simple-framework.slack.com](https://simple-framework.slack.com)

**Mailing List:** [Google Groups](#), [E-Groups](#)

**GitHub Org:** [WLCG-Lightweight-Sites](#)

**Technical Roadmap (WLCG):** [CERN TWiki](#)



# Backup Slides



# SIMPLE Grid Project



Solution for **I**nstallation **M**anagement and **P**rovisioning of **L**ightweight **E**lements.

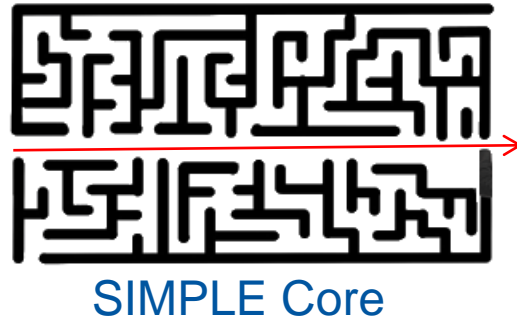
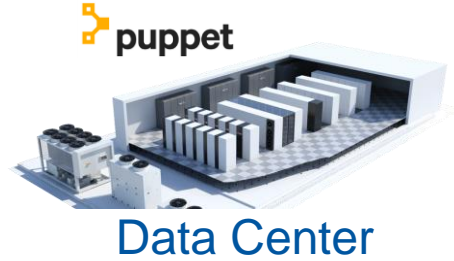
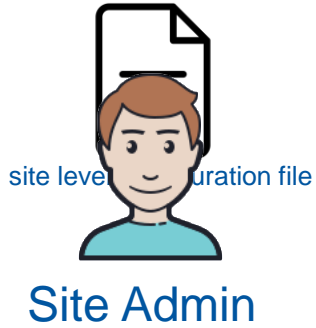
A **private PaaS** that automates configuration and deployment of **WLCG services**, popular software frameworks like Hadoop, Spark etc.. on demand.

Setup and run services with **minimal oversight** and **operational effort**.

Under the hood, we leverage popular configuration management tools like **Puppet /Ansible** and container orchestrators such as **Docker Swarm/Kubernetes**.

**Full autonomy to site admin** to configure grid services through various framework hooks and easy access to containers running grid services.

# SIMPLE Workflow



SIMPLE Component repositories



Grid Service Experts

# SIMPLE Framework: Example

Config Master(CM)

Lightweight Component(LC)

Install puppetserver, puppet

Install puppet

simple-condor-cm



simple-condor-ce

simple-condor-batch

simple-condor-node0

simple-condor-node1

simple-condor-node2



Then, install `simple_grid_puppet_module` on all nodes. For instance,

```
[root@simple-condor-cm ~]# puppet module install maany-simple_grid
```

Then, initialize the nodes using the puppet module. For instance,

```
[root@simple-condor-cm ~]# puppet apply -e "class {'simple_grid::install::config_master::simple_installer':}"
```

```
[root@simple-condor-node0 ~]# puppet apply -e "class {'simple_grid::install::lightweight_component::simple_installer': puppet_master => 'simple-condor-cm.cern.ch'}"
```

Click to view  
Terminal  
captures

# SIMPLE Framework: Example

- Write a **site-level-configuration.yaml** File:

declare variables

```
4   ### Variable declaration:
5   global_variables:
6     - &lightweight_component01_ip_address 188.185.112.251
7     - &lightweight_component01_fqdn simple-condor-ce.cern.ch
8     - &lightweight_component02_ip_address 188.185.84.16
9     - &lightweight_component02_fqdn simple-condor-batch.cern.ch
10    - &lightweight_component03_ip_address 188.185.78.135
11    - &lightweight_component03_fqdn simple-condor-node0.cern.ch
12    - &lightweight_component04_ip_address 188.185.64.158
13    - &lightweight_component04_fqdn simple-condor-node1.cern.ch
14    - &lightweight_component05_ip_address 188.185.68.214
15    - &lightweight_component05_fqdn simple-condor-node2.cern.ch
```

# SIMPLE Framework: Example

Details about your site's infrastructure

```

51  site_infrastructure:
52  | - fqdn: *lightweight_component01_fqdn
53  |   ip_address: *lightweight_component01_ip_address
54  | - fqdn: *lightweight_component02_fqdn
55  |   ip_address: *lightweight_component02_ip_address
56  | - fqdn: *lightweight_component03_fqdn
57  |   ip_address: *lightweight_component03_ip_address
58  | - fqdn: *lightweight_component04_fqdn
59  |   ip_address: *lightweight_component04_ip_address
60  | - fqdn: *lightweight_component05_fqdn
61  |   ip_address: *lightweight_component05_ip_address

```

} Use variables

```

154 supported_virtual_organizations:
155 | - *default_yo_alice

```

} Pick from several default variables



# SIMPLE Framework: Example

Details about your site's infrastructure

```

51  site_infrastructure:
52  - fqdn: *lightweight_component01_fqdn
53    ip_address: *lightweight_component01_ip_address
54  - fqdn: *lightweight_component02_fqdn
55    ip_address: *lightweight_component02_ip_address
56  - fqdn: *lightweight_component03_fqdn
57    ip_address: *lightweight_component03_ip_address
58  - fqdn: *lightweight_component04_fqdn
59    ip_address: *lightweight_component04_ip_address
60  - fqdn: *lightweight_component05_fqdn
61    ip_address: *lightweight_component05_ip_address
    
```

} Use variables

```

152 supported_virtual_organizations:
153 - *default_vo_alice
154 - *default_vo_dteam
155 - *default_vo_ops
    
```

} Pick from several default variables

# SIMPLE Framework: Example

Describe the component repositories should be deployed at the site

```

67  lightweight_components:
68  - type: compute_element
69    name: HTCondor-CE
70    repository_url: "https://github.com/WLCG-Lightweight-Sites/simple_htcondor_ce"
71    repository_revision: "master"
72    execution_id: 0
73    lifecycle_hooks:
74      pre_config: []
75      pre_init: []
76      post_init: []
77    deploy:
78      - node: *lightweight_component01_fqdn
79        container_count: 1
80    config:
81      uid_domain: cern.ch
82      condor_host_execution_id: 1
83    preferred_tech_stack:
84      level_2_configuration: sh
85    supplemental_config:
86      condor-ce:
87        {condor_knob} : {value}

```

} SIMPLE's repository for HTCondorCE

} Specify which node in the data center should run HTCondorCE

} Specify configuration parameters requested by HTCondorCE repository developers

} Additional Condor configuration knobs for your setup

# Grid Service Experts

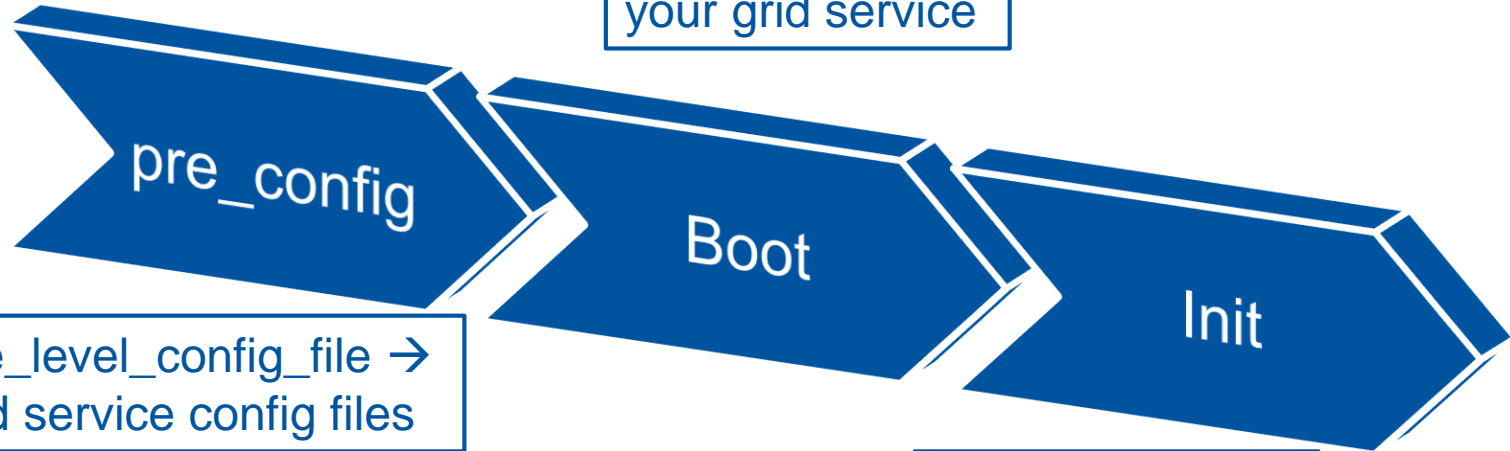


- Easily **add support for grid services** by creating component repository
- Add **code + Dockerfile** to repository lifecycle events that instruct the core framework on how to deploy your grid service containers.
- Get in touch with us to learn more.

# Component Repository



Dockerfile for  
your grid service



site\_level\_config\_file →  
grid service config files

Entrypoint for  
your grid service  
container

# Diversity in WLCG

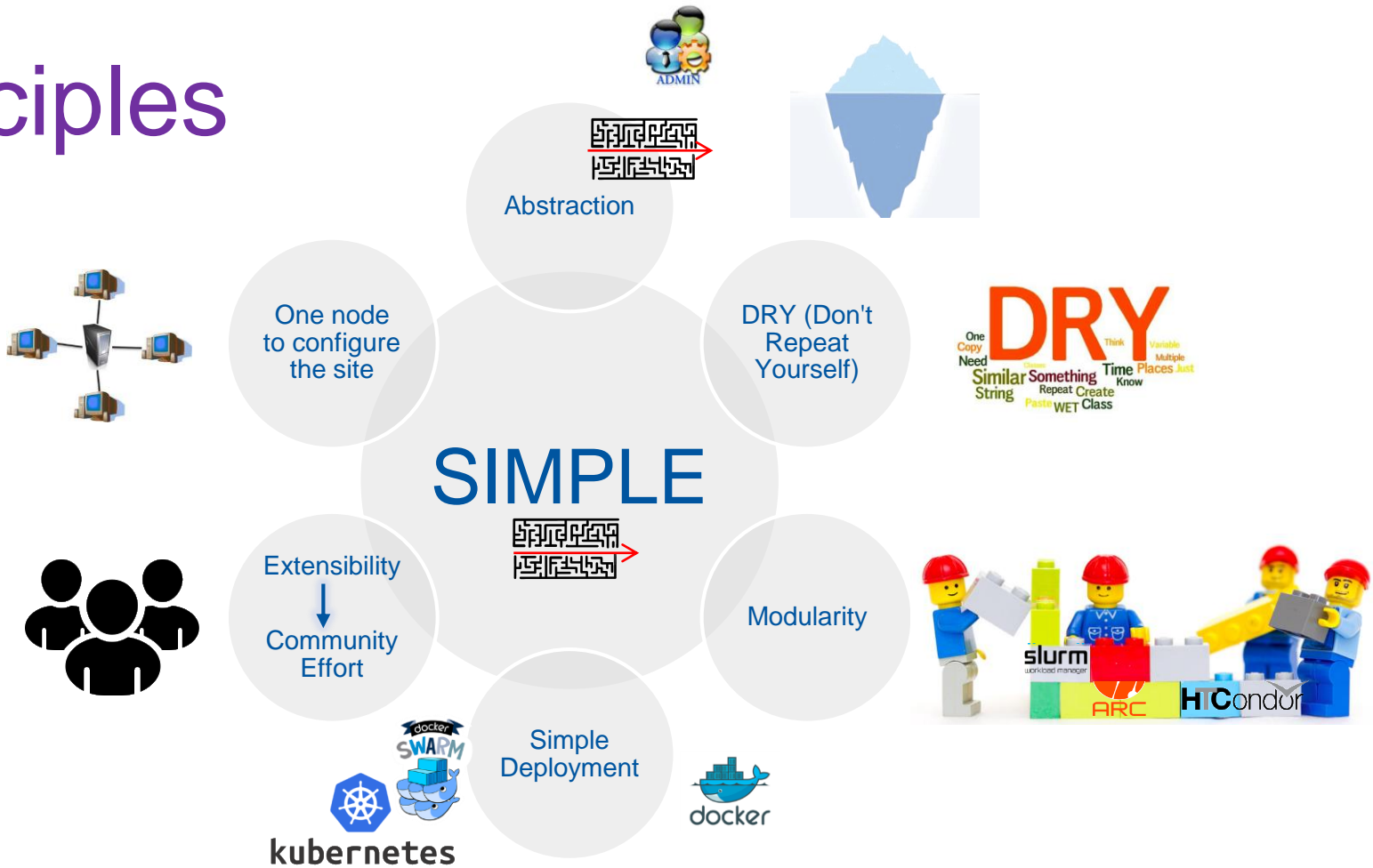
Types of **CE/Batch/WN/Middleware** packages



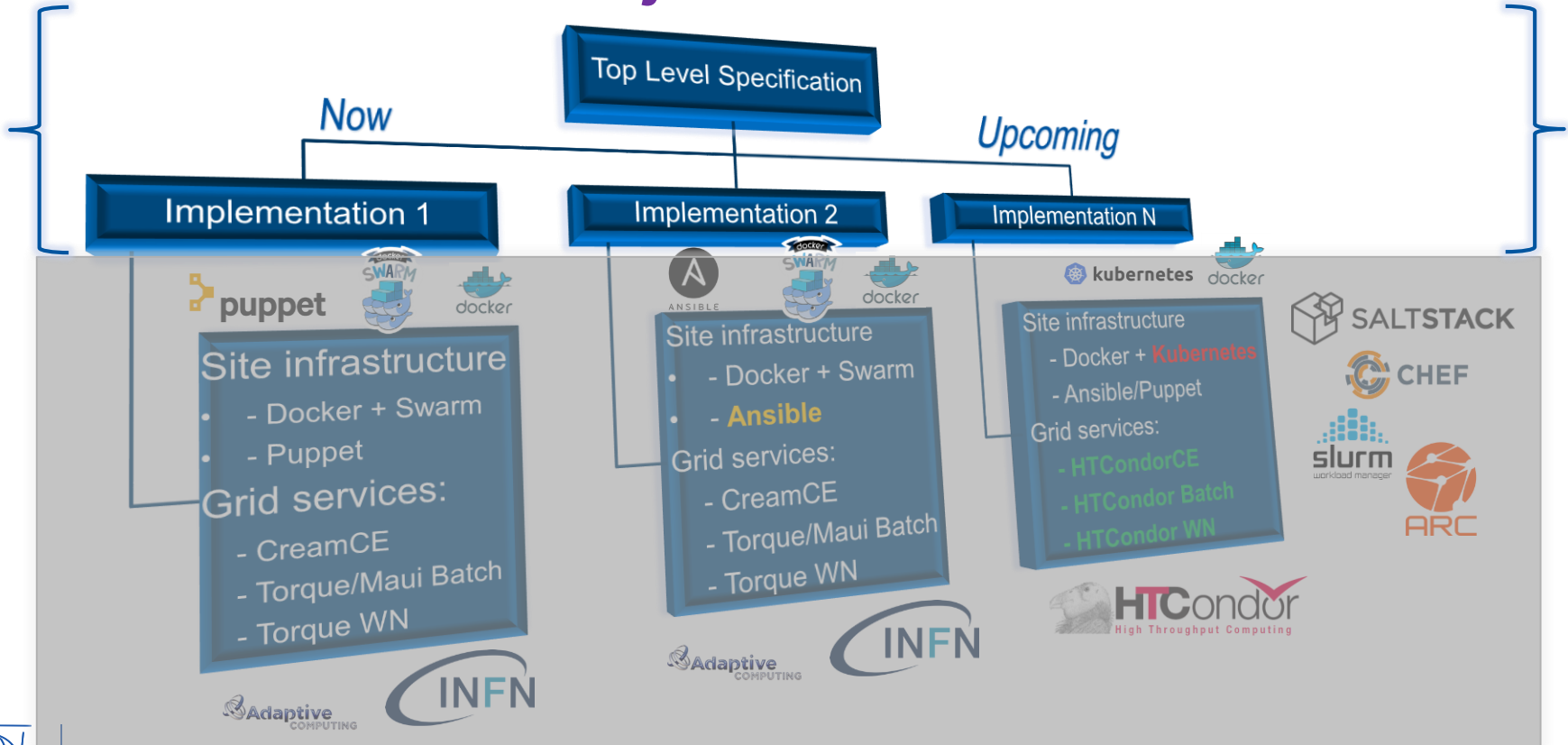
**Technologies preferred** by site admins for managing their infrastructure



# Principles



# SIMPLE – Project Structure



# Site Level Configuration File



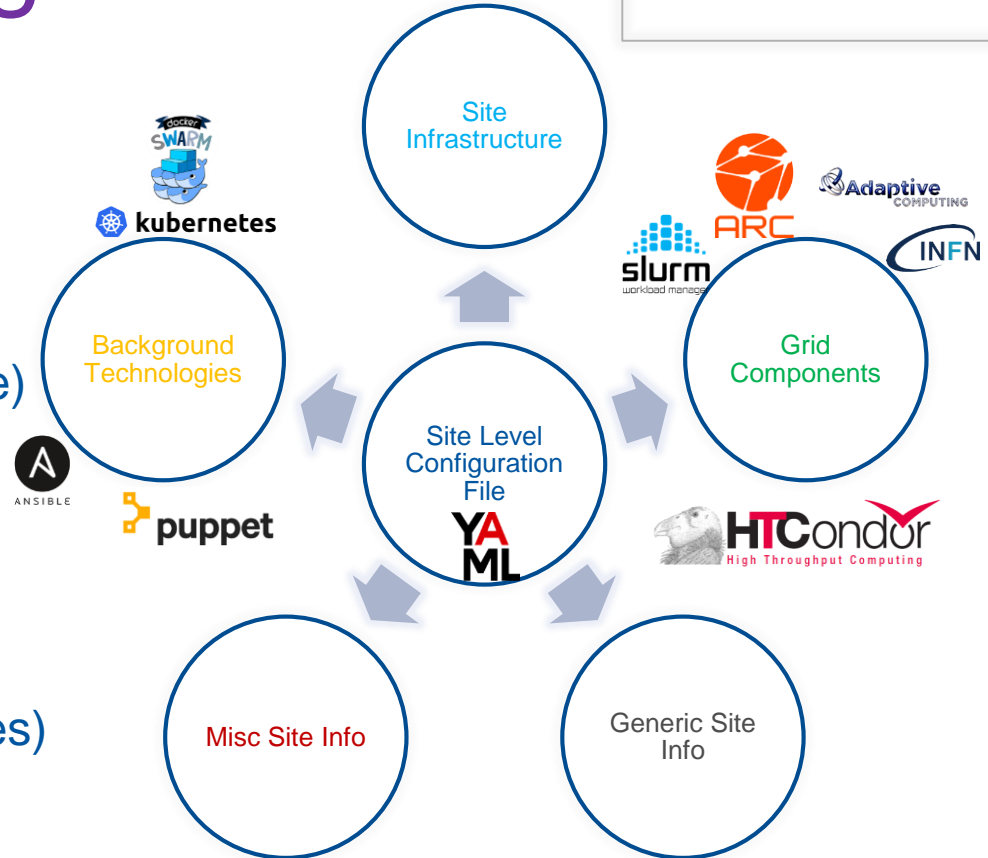
A single **YAML** file to describe:  
**Site-Infrastructure** (Hostnames, IP addresses, OS/Kernel, Disk/Memory)

**Grid Components** (What grid components to install and configure)

**Generic Site Info** (Users, Groups, Supported VOs)

**Misc. Site Info** (Security emails, location etc.)

**Background Technologies**  
(Puppet/Ansible, Docker/Kubernetes)





# Site Level Configuration File



- **Minimize configuration** requirements via
  - **Variables**
  - **Sensible default values** for site-level configurations
  - **Ability to override values**
  - **support additional parameters** not defined in the system
  - Tested: **O(100) lines of YAML code** to set up the site
  - Split configuration into **multiple logically related YAML files** that can be shared

# Component Repositories



- Publicly hosted repositories on GitHub that provide
  - **Dockerized** CE/WN/Batch/Squid etc.
  - **Meta information** for configuration of images using different configuration management tools
- 1 repository for every component (for instance, CreamCE, CondorCE, Torque, Slurm reside in separate repositories)
- Examples: [HTCondorCE](#), [HTCondorBatch](#), [CreamCE](#), [TorqueWN](#)

# Configuration Validation

- Configuration validation engine to ensure information supplied in site configuration file:
  - **meets the configuration requirements** of desired site component
  - **is realizable on the available infrastructure** using available background technologies
- <http://cern.ch/go/CvS8>
- Possibility to inject custom validation rules

# Central Configuration Manager



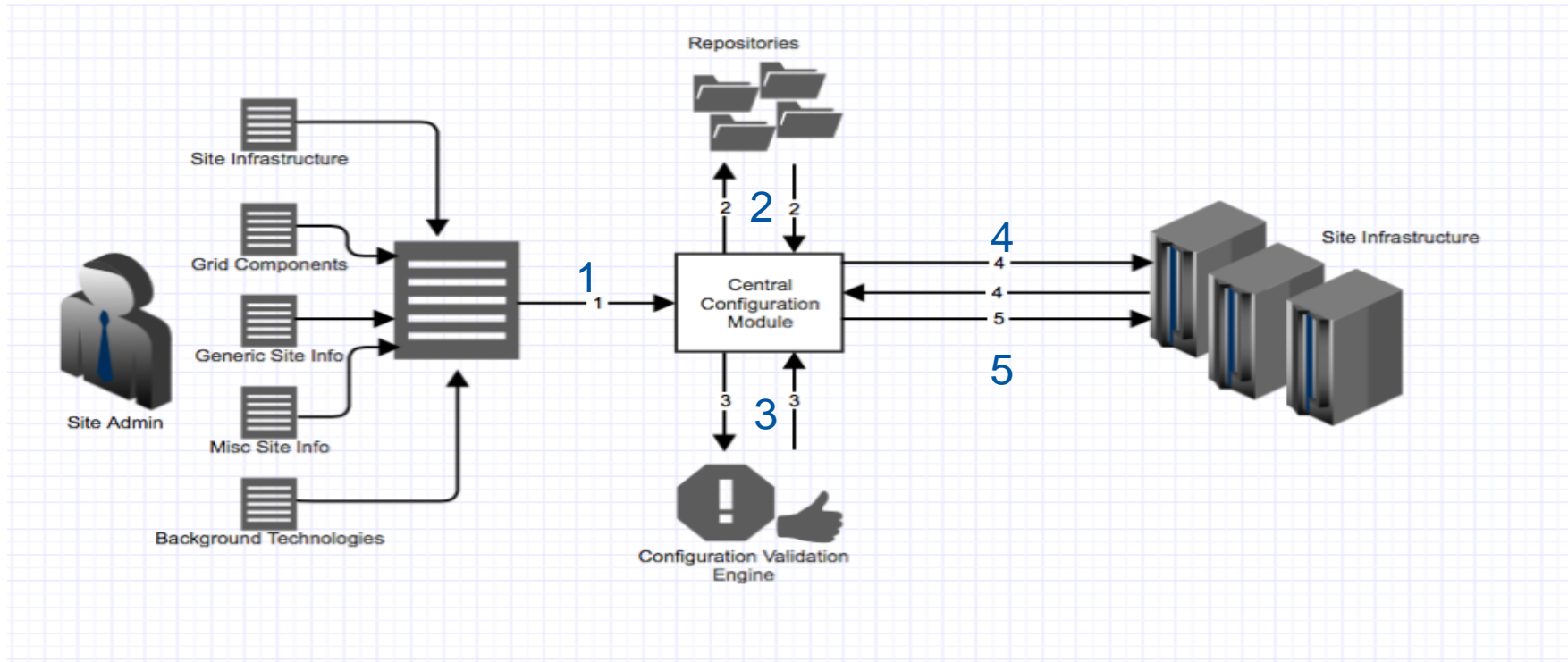
- The **main module** for centrally configuring everything at the site
- **Uses Validation Engine** to check site-configuration file
- Checks **status of available Site Infrastructure** that needs to be orchestrated
- Installs and **configures Grid components** from the repositories

# Central Configuration Manager

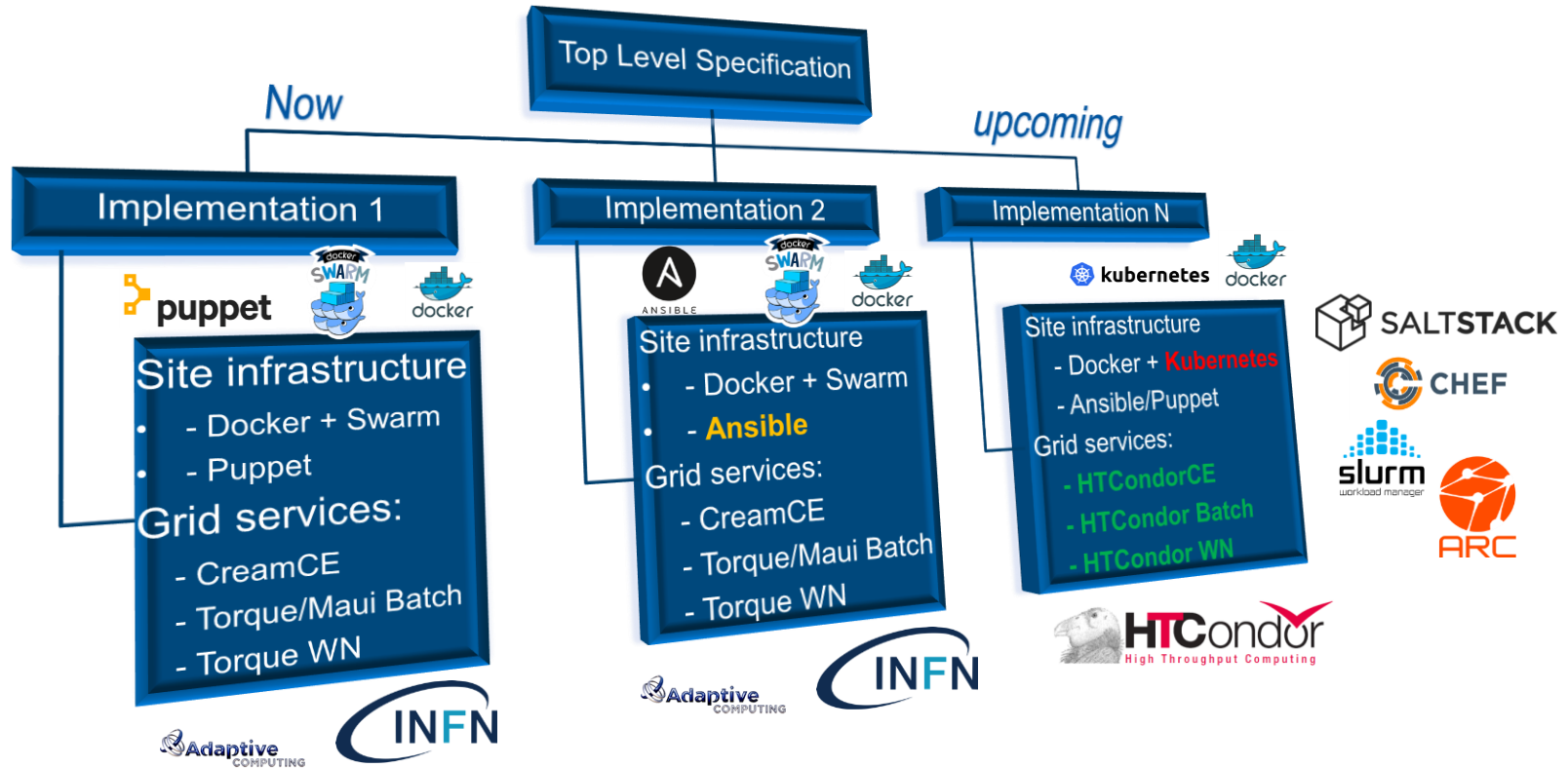


- Implements a **Networking strategy** (overlay/dedicated)
- Ensures availability of **CVMFS** to the containers
- Runs **tests** to check for success or failure of site configuration

# Specification: Putting it Together



# Flashback – Project Structure



# Implementations

- **Site Level Configuration File YAML Compiler**

- Python command line utility

- **Configuration Validation Engine**

- Python command line utility

Google Summer of Code  
2019 Project



- **Repositories for Grid Components**

- Cream Compute Element + Torque Batch System

- Torque Worker Node

- ...   

- **Central Configuration Management System**

- Puppet

- Ansible

- ...   SALTSTACK  CHEF

Google Summer of Code  
2018 Project





# Conclusions

- Set up a **grid site with O(100) lines of YAML**
- **Modular and easy to extend to support other grid services**
- **Community Driven: Open source and open discussion channels. Join Now!!**