
CERN

WLCG Lightweight Sites

Mayank Sharma (IT-DI-LCG)

WLCG Sites

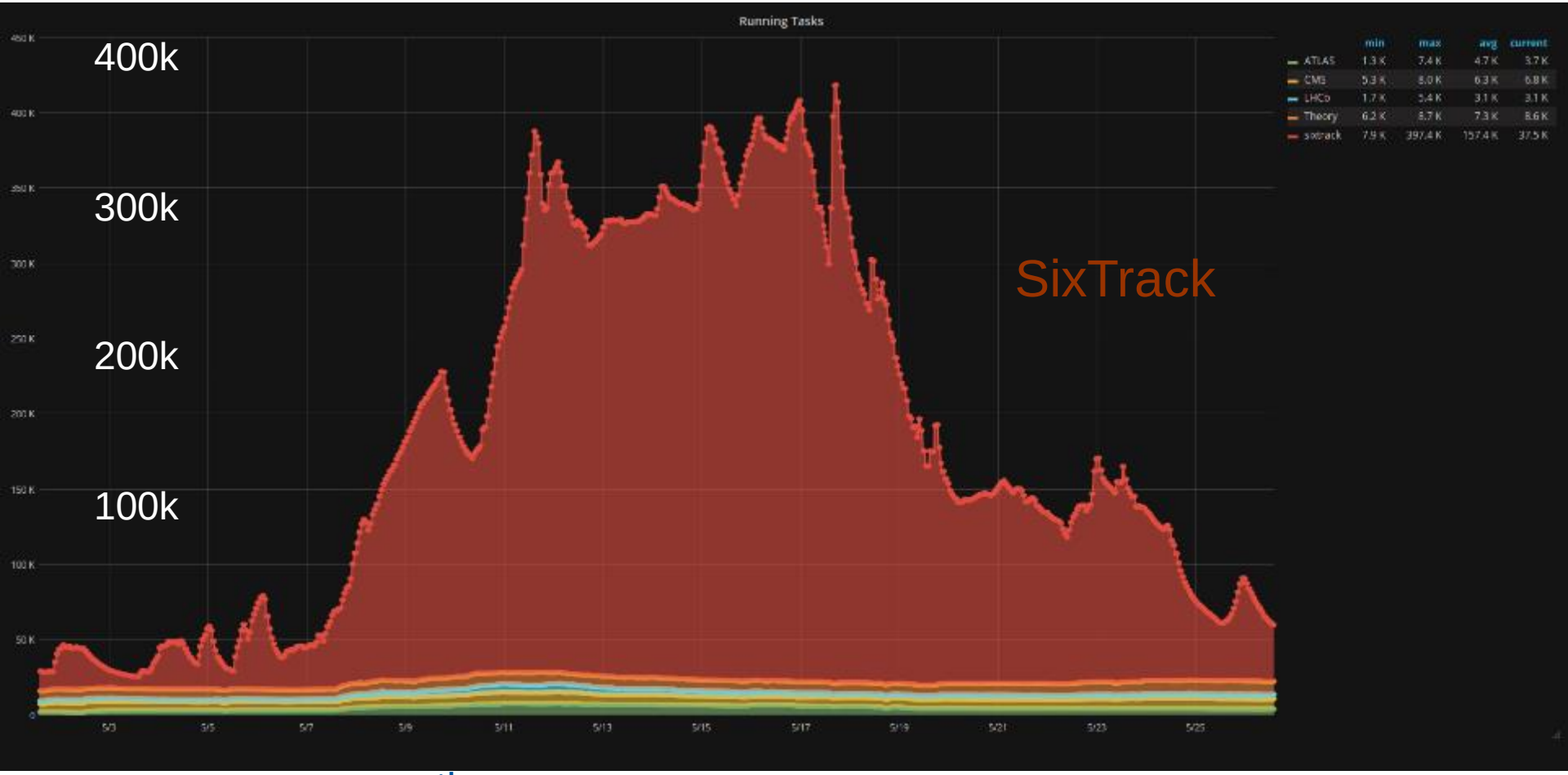
- Grid is a diverse environment (Various flavors of CE/Batch/WN/ +various preferred tools by admins for configuration/maintenance)

YAIM

WLCG Sites

- Site Admin : require significant insight into middleware and services for installing/configuring/maintaining grid services and infrastructure
- Easier for bigger and experienced sites (T1 and many T2). Not very intuitive for smaller/ newer sites.

Potential of smaller sites



8th BOINC Pentathlon 2017

Survey results!

- Link : http://cern.ch/go/rhV9
- 51 Sites respond to the questionnaire that shows potential benefits of **shared repositories**
- Strong support for **Docker, Puppet, OpenStack images**

WLCG Lightweight Sites

- We would like to have sites that can run with **minimal oversight and operational efforts** from people at the site.
- They run almost ”by themselves”.
- Provide resources with **preferred technology** with less effort (configuration management, maintenance etc.)
- Keep things basically the **same for us**, but easier for admins

Lightweight Site Principles

1. **Abstraction**: to abstract the nuances of several popular CE/WN/Batch technologies as much as possible from site-admin's view.
2. **Modular Design**: Allow admins to use existing and popular tools for setting up their sites.
3. **Simple Deployment**: Packaged into containers/ VM's for easy distribution and deployment

Lightweight Sites Principles

4. **Centrally Configurable**: Instead of individually configuring components on nodes, configure everything at site level rather
5. **Extendable**: A community driven effort to develop implementations for various CE/Batch in parallel

Lightweight Site Specification

- **Describe the components of lightweight sites**.
  - Main function of the component
  - Configuration parameters
  - Communication protocols to interact with each other
  - Repository structure for modular implementations
  - Deployment/Release processes
  - Maintenance guidelines

Lightweight Sites Components

1. Site Level Configuration File
2. Repositories for containers of different CE/Batch/WN
3. Configuration Validation System
4. Central Configuration Manager
5. Networking strategy
6. File System (CVMFS)/ Caches

Site Level Configuration File

- A site-level YAML file to describe:
  1. **Site Infrastructure:**
     1. Hostnames, IP Addresses, OS/Kernel, SSH access, Disk/ Memory/ CPU/ Network information
  2. **Grid Components:**
     1. Site Components: CE/Batch/WN/Middleware etc.
     2. What to use(Arc, Condor, Slurm) and what versions
     3. Node on which they should be configured.
     4. Component specific configuration(fetched from component repositories)

Site Level Configuration File

3. **Generic Site Info:** Users, Groups, VO's, Host Certificates
4. **Misc Site Info:** security emails, support emails etc.
5. **Background Technologies:** preferred tools for container orchestration(Kubernetes, Docker Swarm)/ configuration management(Puppet, Ansible) to be used for configuring the site.

Site Level Configuration File

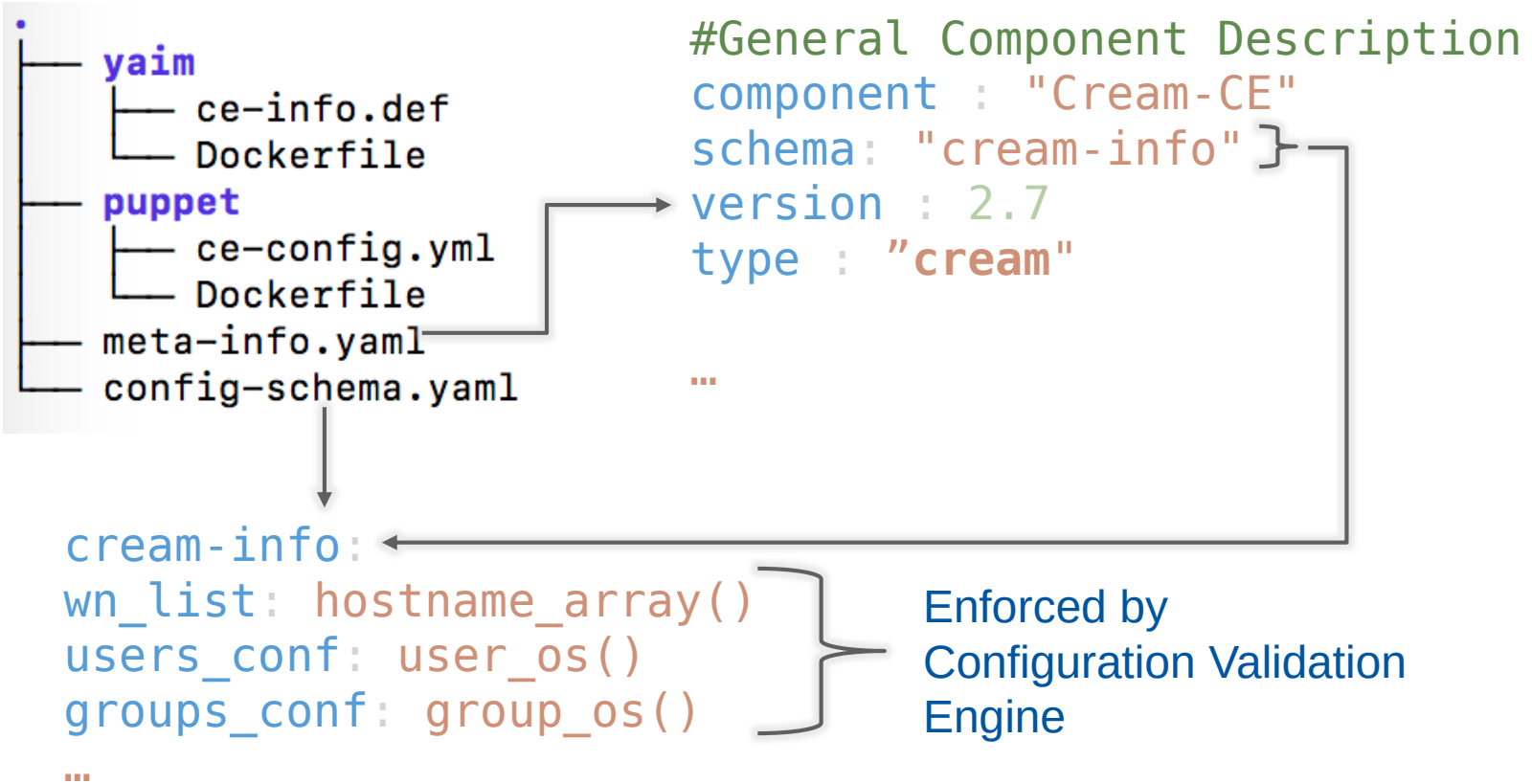
```
component-info:
ce:
repo: “https://github.com/...”
type: {cream, arc, condor, other}
hostnames: {‘ce-01.domain’, ‘ce-02.domain’}

wn:
repo: “https://github.com/...”
type: {pbs, condor, slurm}
hostnames: {‘wn-01.domain’, ‘wn-02.domain’}

batch:
repo: “https://github.com/...”
type: {pbs, condor}
…
```

Component Repositories

- Publicly hosted repositories on GitHub that provide
  - code for the images of 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)

Component Repositories

- Repository Structure



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

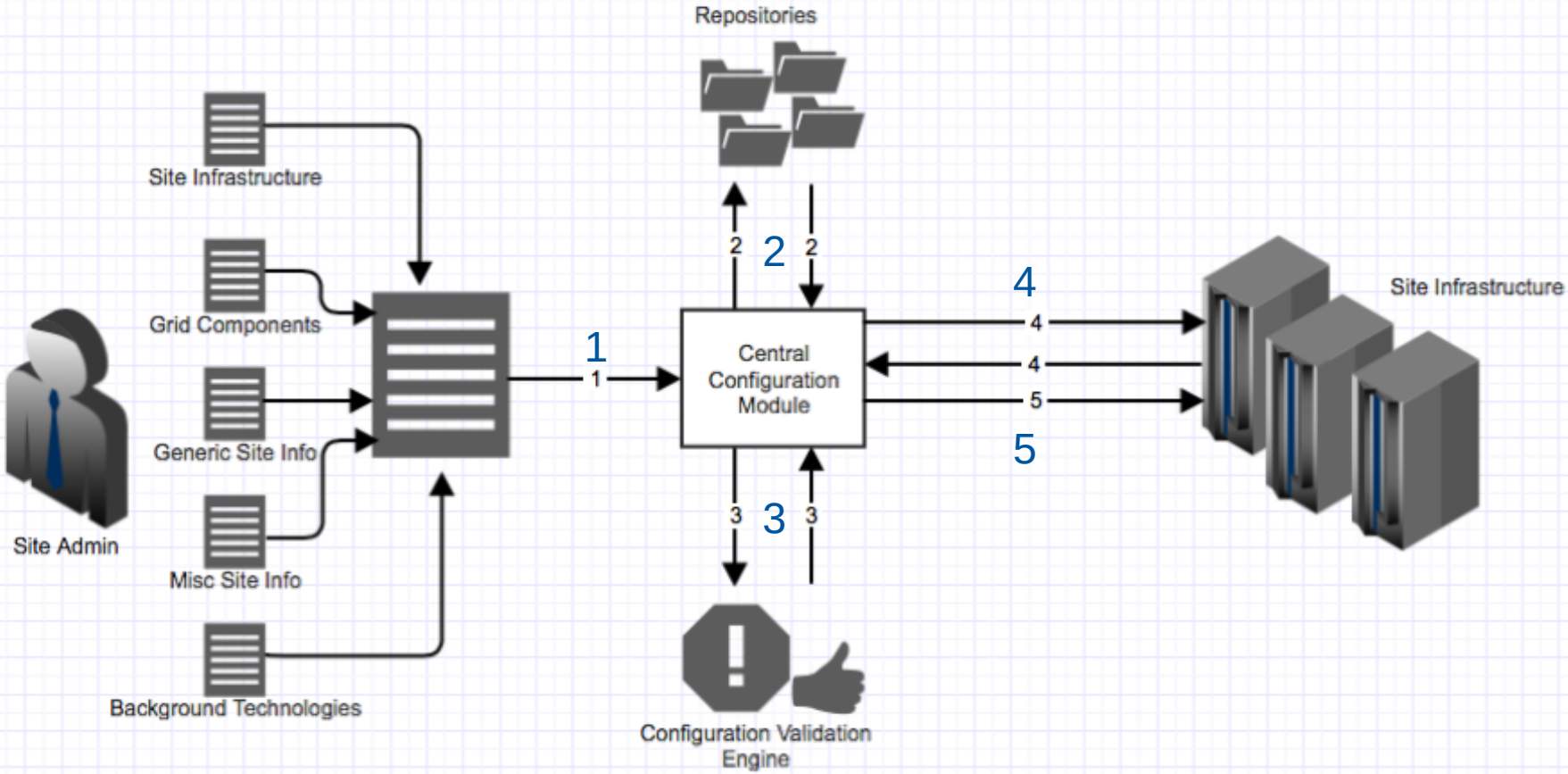
Central Configuration Manager

- The **main module** for centrally configuring everything
- **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**

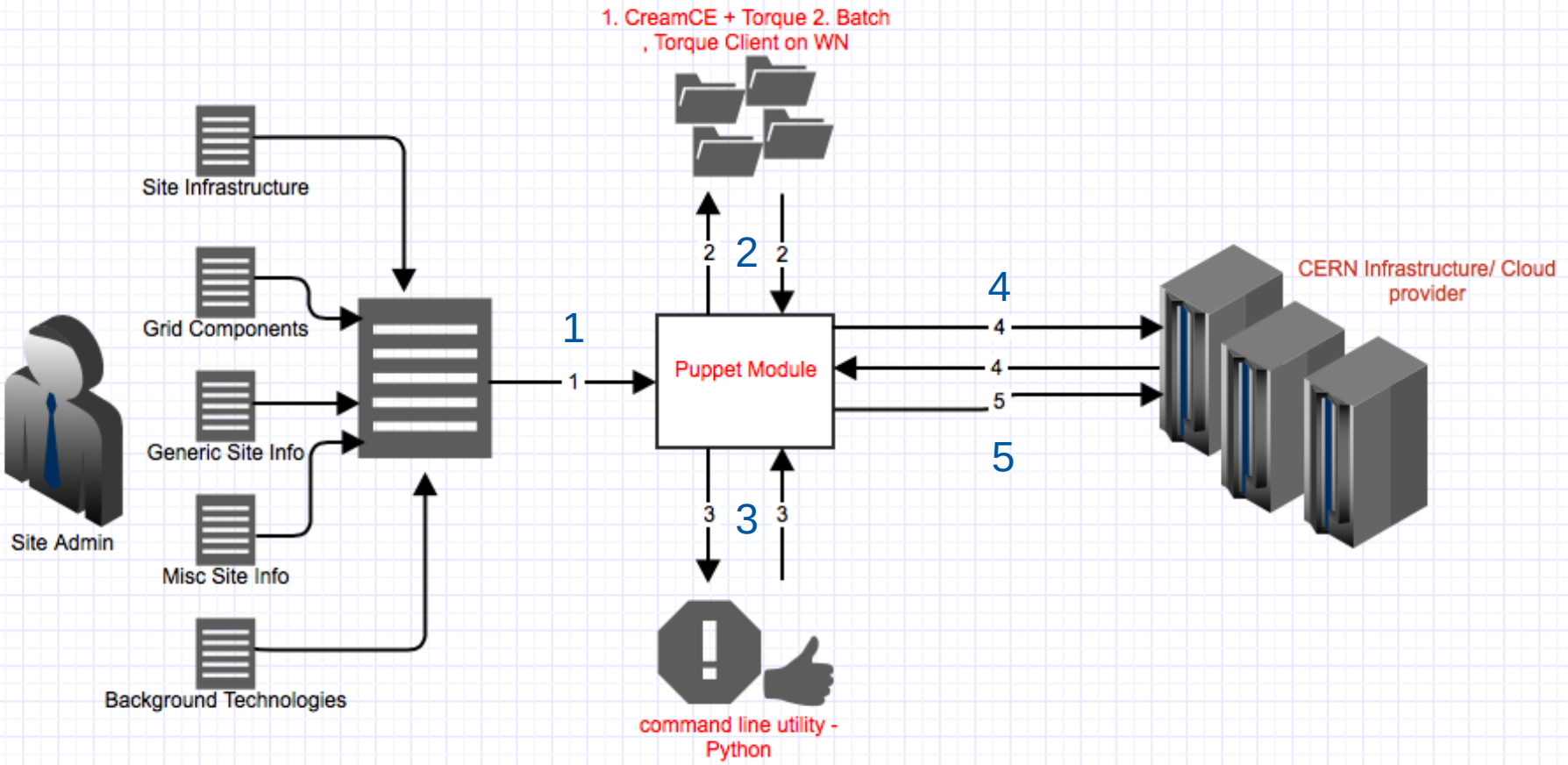
Central Configuration Manager

- Implements a **Networking strategy** (overlay/ dedicated)
- Ensures availability of the **File System (CVMFS)** and **Caches** to the containers
- Runs Tests like submitting jobs to check for success or failure of site configuration

Specification: Put it together



1st Implementation



Implementation Status

- CE: Cream
- Batch: Torque
- WN: Torque client
- Background Technologies:
  - Docker (containers)
  - Docker Swarm(container orchestration)
  - Puppet (configuration management)
- Infrastructure: CERN OpenStack/ Public cloud infrastructure providers (not yet final)

Implementation Status

- Central Configuration Manager: Puppet
- Configuration Validation Engine: Python command line utility
- Overall Status:
  - Complete:
    - Containers for CreamCE, TorqueWN (test job. ✅)
    - YAIM based configuration of containers
  - En-route (within 2 weeks):
    - Public repositories
    - Puppet module for central config management
    - More documentation

2nd Implementation: GSoC 2018

- 2 Google Summer of Code 2018 Projects
- Background Technologies:
  - Docker (containers)
  - Kubernetes (container orchestration)
  - Ansible (configuration management)
- Timeline (May 2018 – September 2018)

Supporting new components

- Modular design can support ARC, SLURM, Condor etc.
- New repository for the components
  - Dockerfile: instructions for setting up OS , relevant packages, middleware.
  - Entrypoint/ init script: used by container to configure itself on startup based on information available through the central configuration module.

Community

- Technical Discussion List (E-Groups):
  - Name: WLCG-Lightweight-Sites-Dev
  - Link: http://cern.ch/go/l9wZ

- Google Group (Open Source Community)
  - Name: WLCG Lightweight Sites
  - Link: http://cern.ch/go/Hz7S

Other ‘Lightweight’ ideas

- Not classic grid sites.
- Regional HTCondor Pools.
- Small sites boot up containers that connect to the regional pool for workloads
- 1 or 2 proof of concepts exist
- On our roadmap after release of version 1.0.0

Questions