CernVM – a virtual software appliance for LHC applications

2nd ASPERA Workshop

30-31 May 2011, Barcelona, Spain

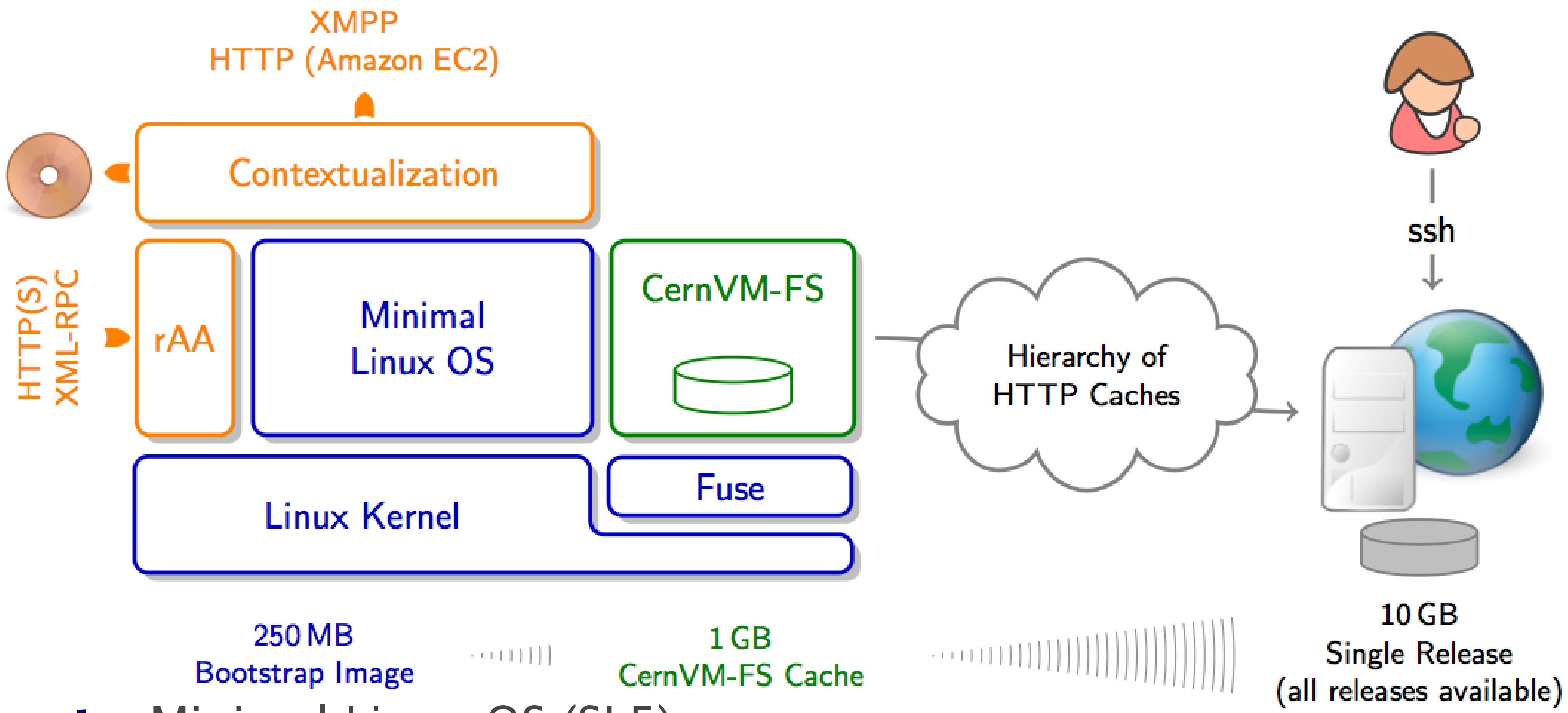
P. Mato /CERN

CernVM R&D Project

- Aims to provide a complete, portable and easy to configure user environment for developing and running LHC data analysis locally and on the Grid independent of physical software and hardware platform (Linux, Windows, MacOS)
  - Code check-out, edition, compilation, local small test, debugging,…
  - Grid submission, data access…
  - Event displays, interactive data analysis,
  - Suspend, resume…
- Decouple application lifecycle from evolution of system infrastructure
- Reduce effort to install, maintain and keep up to date the experiment software

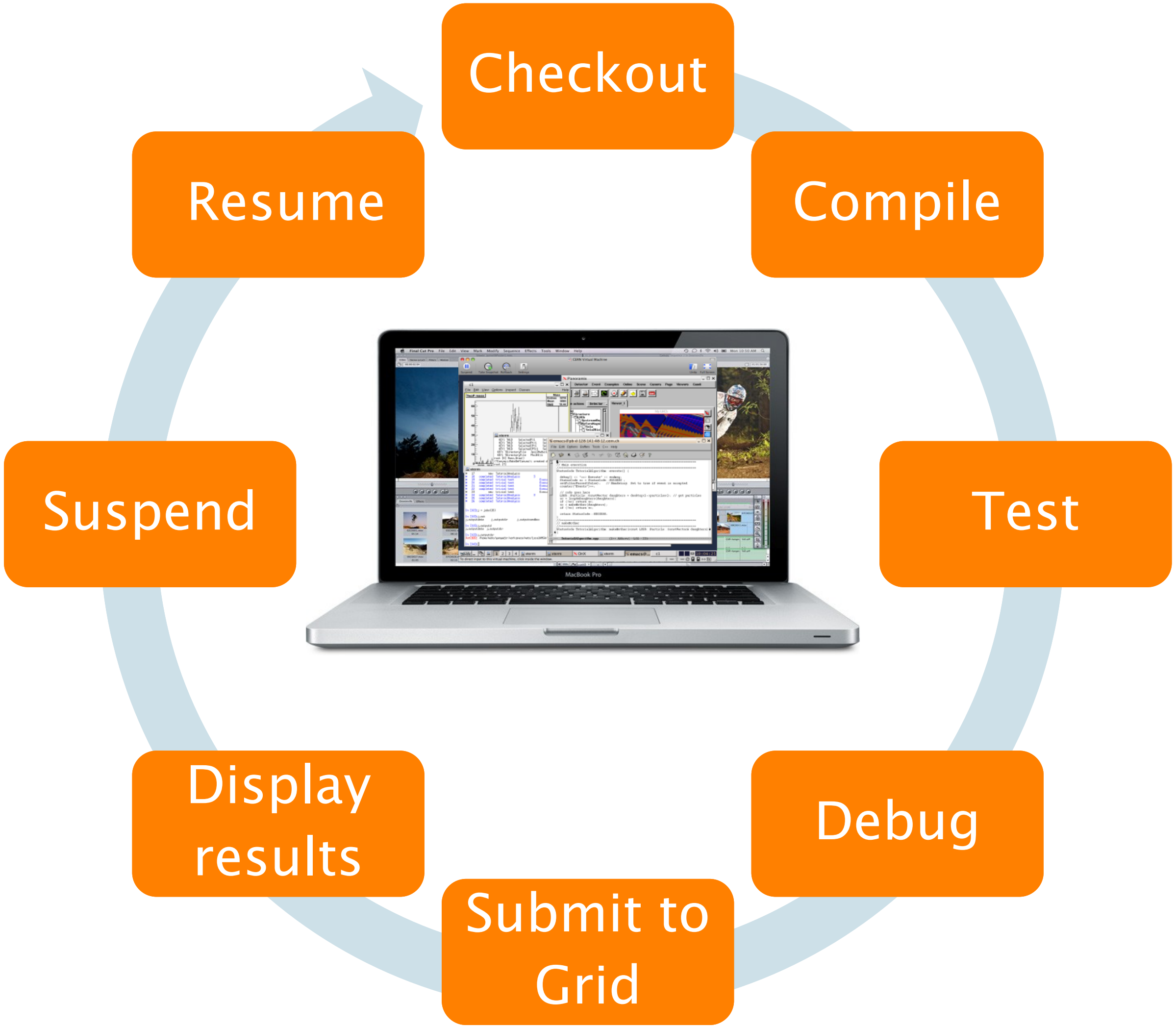
http://cernvm.cern.ch

CernVM Elements



1. Minimal Linux OS (SL5)
2. CernVM-FS - HTTP network file system optimized for jus in time delivery of experiment software
3. Flexible configuration and contextualization mechanism based on public Cloud API

Initial Scope



Checkout

Compile

Test

Debug

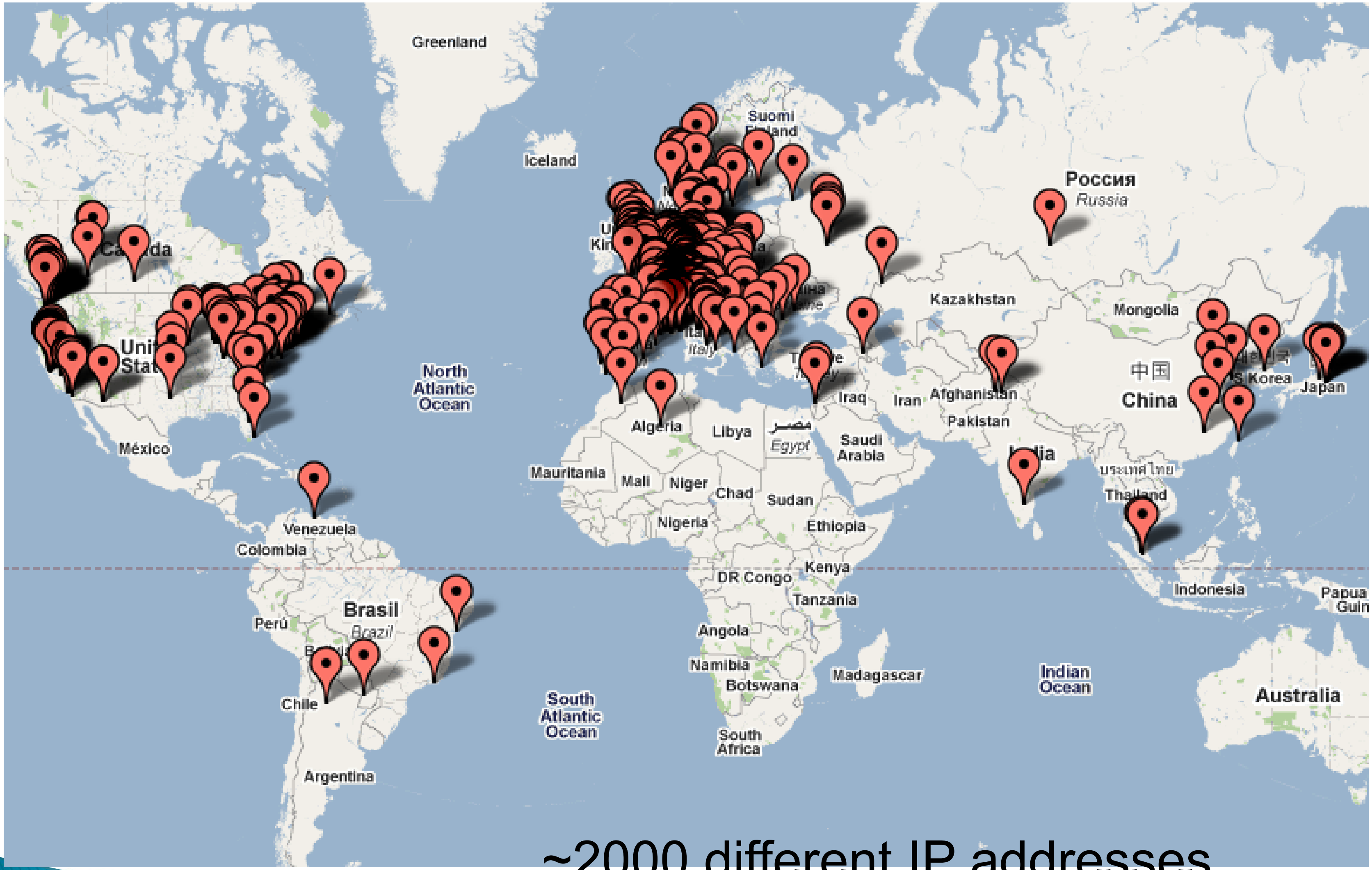
Submit to Grid

Display results

Suspend

Resume

Where are our end-users?



~2000 different IP addresses

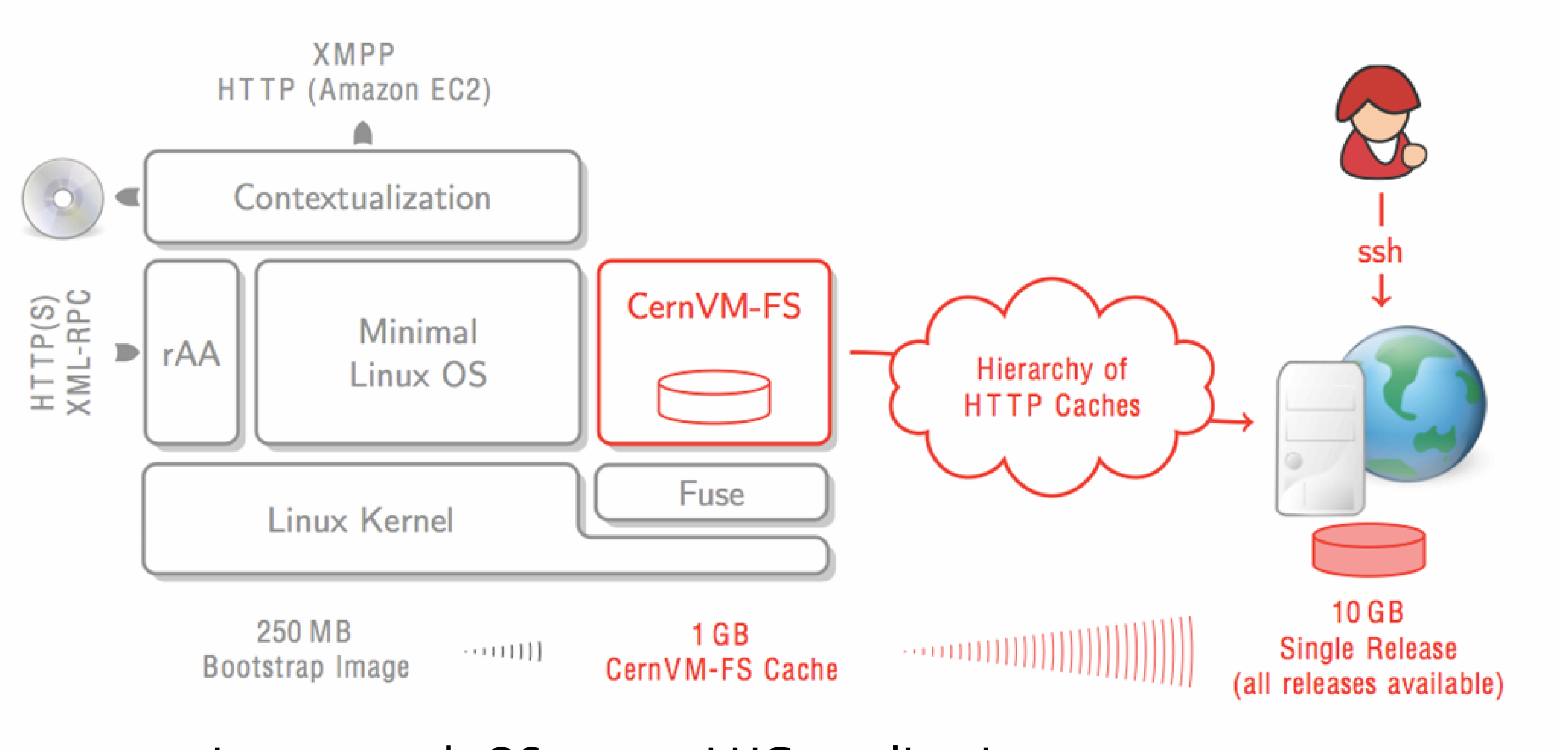
Is CernVM suitable for deployment on Grid and Cloud infrastructure?

What are the benefits of going CernVM way comparing to more traditional[1)] approach to batch node virtualization?

1) Traditional approach:

- Take “standard” batch node [2GB] and add experiment software [10GB] and generate VM image. Have experiment and security team certify the image, deploy it to all sites and worker nodes. Repeat this procedure 1-2 times per week and per experiment.

Part #1: Minimal OS image

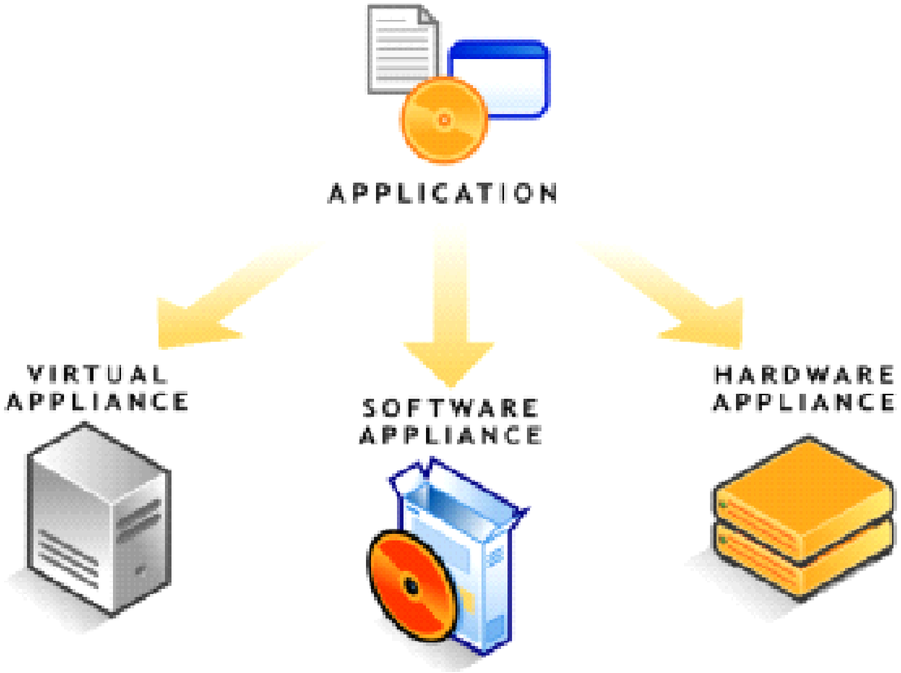


- Just enough OS to run LHC applications
- Built using commercial tool (rBuilder by rPath)
    - Top-down approach – starting from application and automatically discovering dependencies
- Small images (250MB), easy to move around

Appliance Builder

CernVM
SoFTware Appliance

Starting from experiment software…



…ending with a custom Linux specialised for a given task

rBuilder™

- Installable CD/DVD
- Stub Image
- Raw Filesystem Image
- Netboot Image
- Compressed Tar File
- Demo CD/DVD (Live CD/DVD)
- Raw Hard Disk Image
- Vmware ® Virtual Appliance
- Vmware ® ESX Server Virtual Appliance
- Microsoft ® VHD Virtual Apliance
- Xen Enterprise Virtual Appliance
- Virtual Iron Virtual Appliance
- Parallels Virtual Appliance
- Amazon Machine Image
- Update CD/DVD
- Appliance Installable ISO

Conary Package Manager

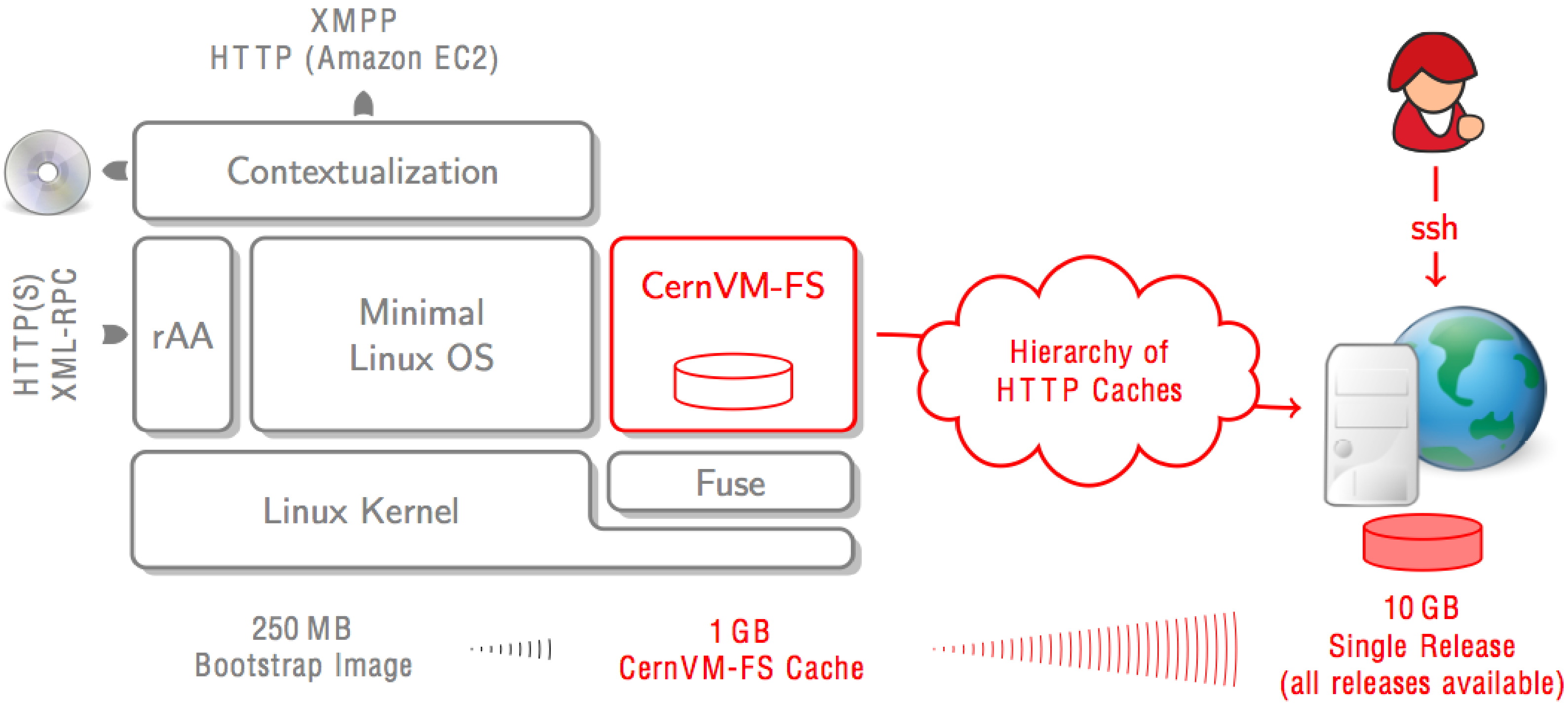
1. Find what you need

2. Build your recipe

3. Cook it!

Every build and every file installed on the system
is automatically versioned and accounted for in a database

Part #2: CernVM-FS

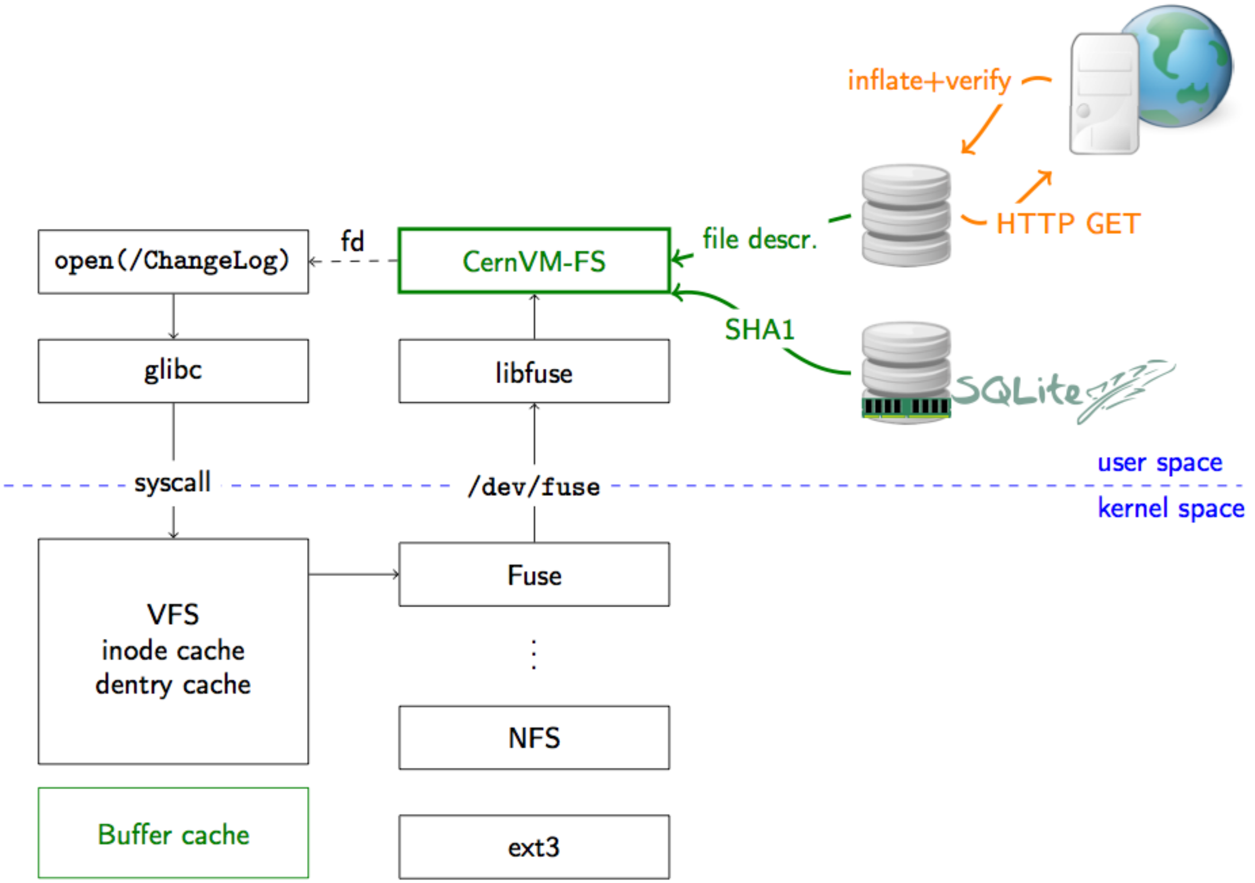


- Experiment software is changing frequently and we want to avoid need to frequently update, certify and redistribute VM images with every release
- Only a small fraction of software release is really used
- Demonstrated scalability and reliability
- Now being deployed on across all Grid sites as the channel for software distributions

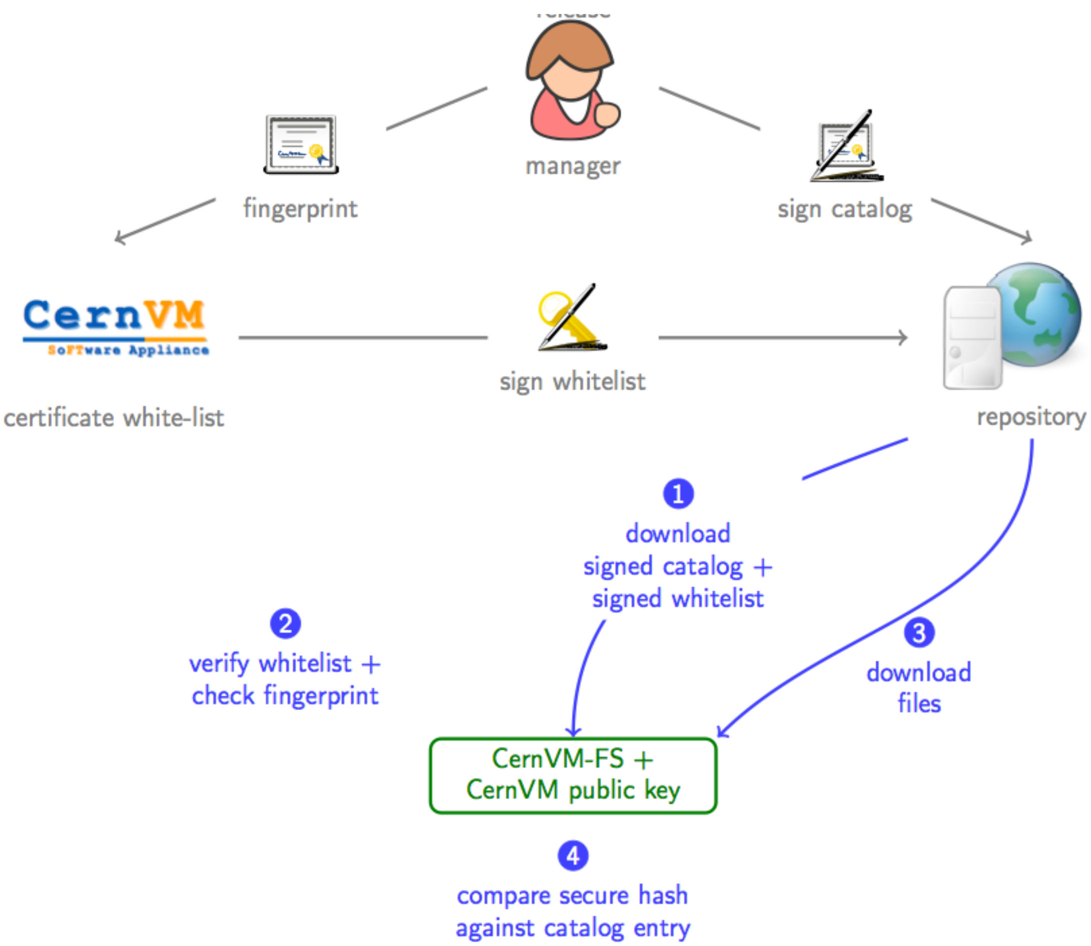
Application Software Delivery

- CernVM comes with the read-only file system (CernVM-FS) optimized for software distribution
  - Very little fraction of the experiment software is actually used (~10%)
  - Very aggressive local caching, web proxy cache (squids)
  - Transparent file compression
  - Integrity checks using checksums, signed file catalog
  - Operational in off-line mode
- No need to install any experiment software
  - 'Virtually' all versions of all applications are already installed
  - The user just needs to start using it to trigger the download

Fuse Module


inflate+verify
HTTP GET
file descr.
SHA1
SQLite
open(/ChangeLog)
fd
CernVM-FS
glibc
libfuse
syscall
/dev/fuse
user space
kernel space
VFS
inode cache
dentry cache
Fuse
⋮
NFS
Buffer cache
ext3

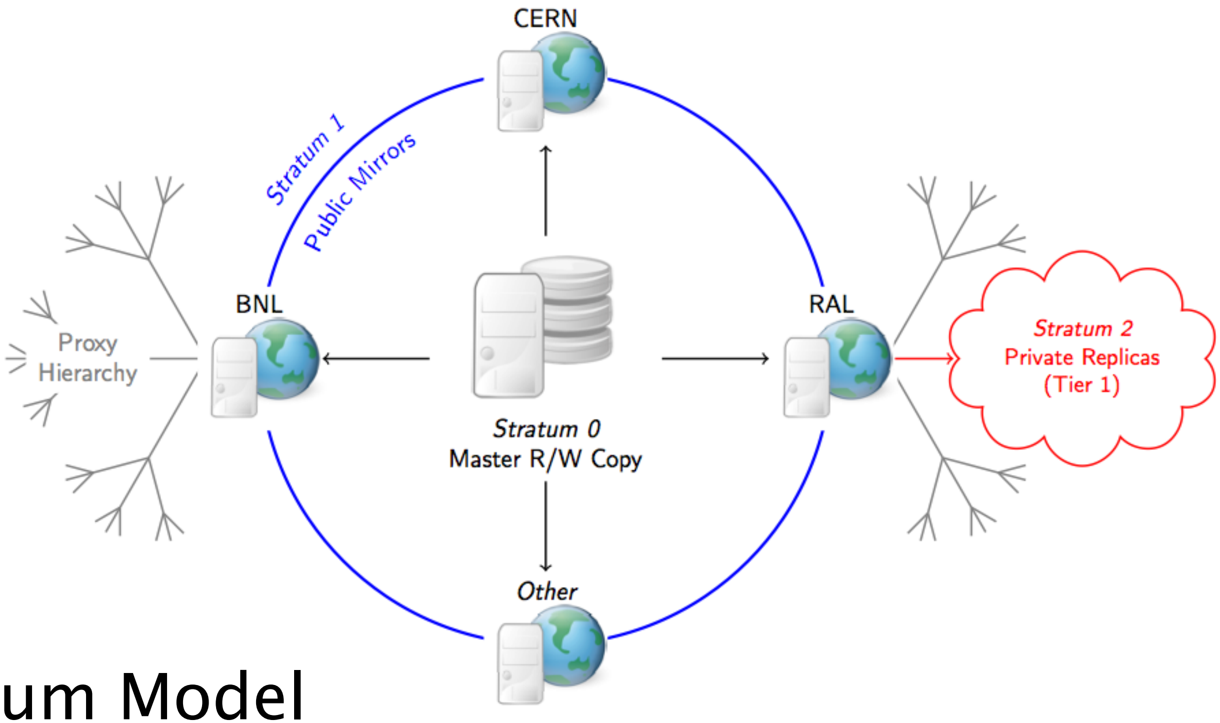
Integrity and Authenticity



release
manager
fingerprint
sign catalog
CernVM Software Appliance
certificate white-list
sign whitelist
repository
1 download signed catalog + signed whitelist
2 verify whitelist + check fingerprint
3 download files
CernVM-FS + CernVM public key
4 compare secure hash against catalog entry

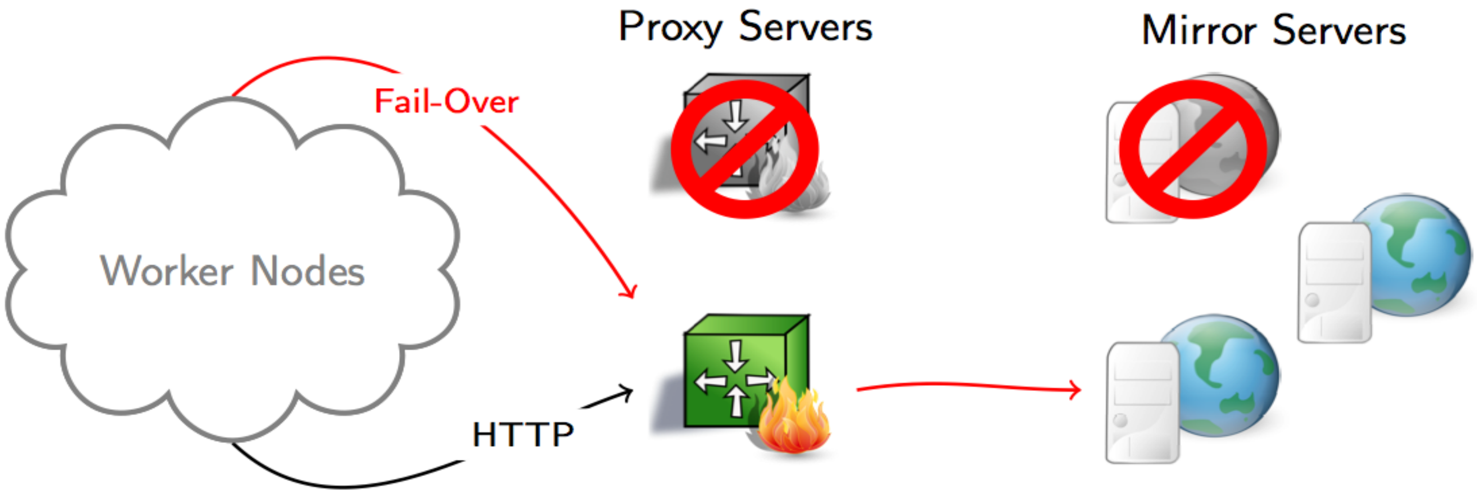
- Principle: Digitally signed repositories with certificate white-list

Content Distribution



- Stratum Model
  + Fast and Scalable
  + No single point of failure
  – Complex hierarchy

Client-Side Fail-Over



- Proxies
  - SL5 Squid, load-balancing + fail-over
    e. g. CVMFS_HTTP_PROXY="A|B|C"
- Mirrors
  - Fail-over mirrors at CERN, RAL, BNL
    For roaming users automatic ordering based on RTT

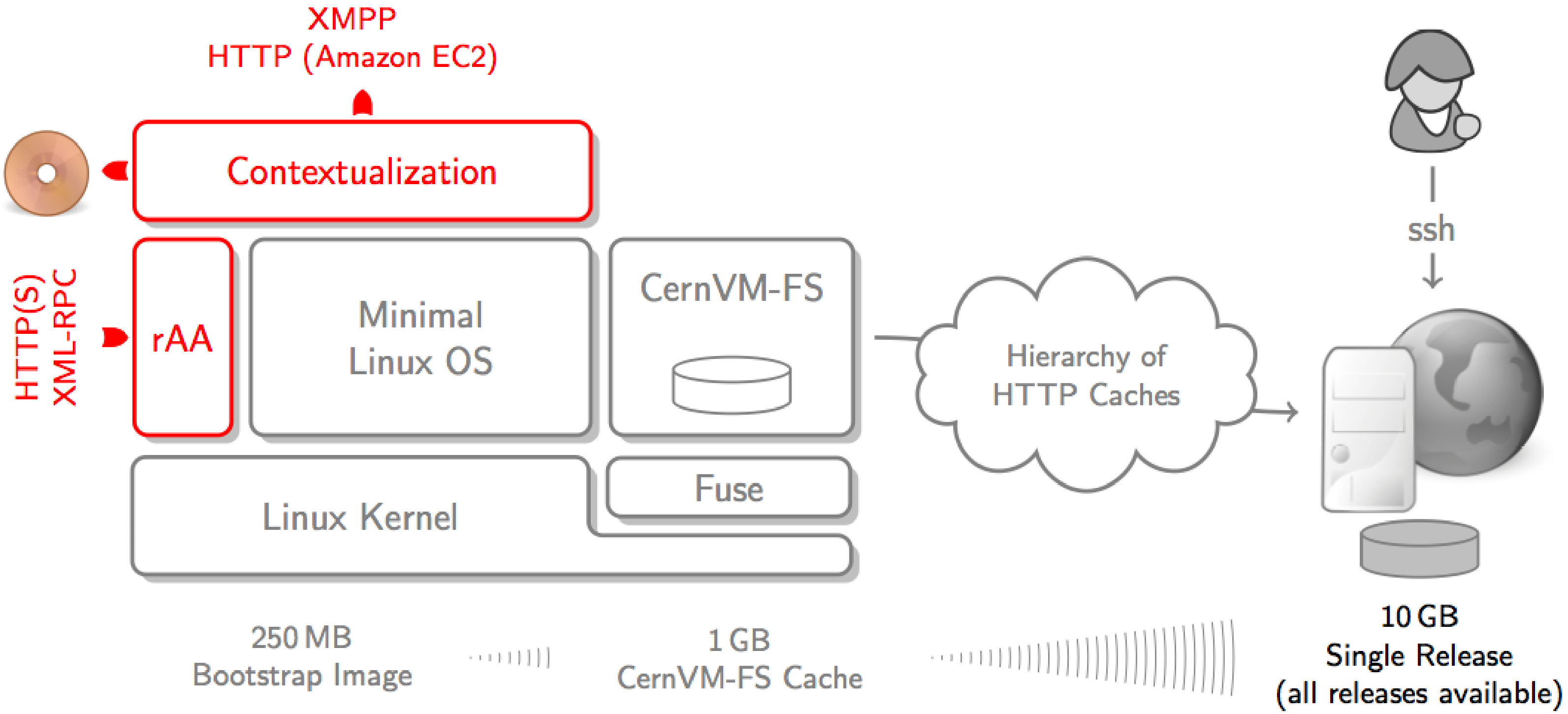
CernVM-FS within WLCG

- CernVM-FS is set to replace current software installation methods
- Current method
  - Run <VO>sgm job via grid
  - Write files within job to some shared storage
  - Validate software
  - Publish tag in BDII
  - Process has to be repeated (an debugged) at every site
- Being deployed at Tier-1s, Tier-2s and Tier-3s
  - ATLAS and LHCb running productions with it

Advantages for WLCG Grid

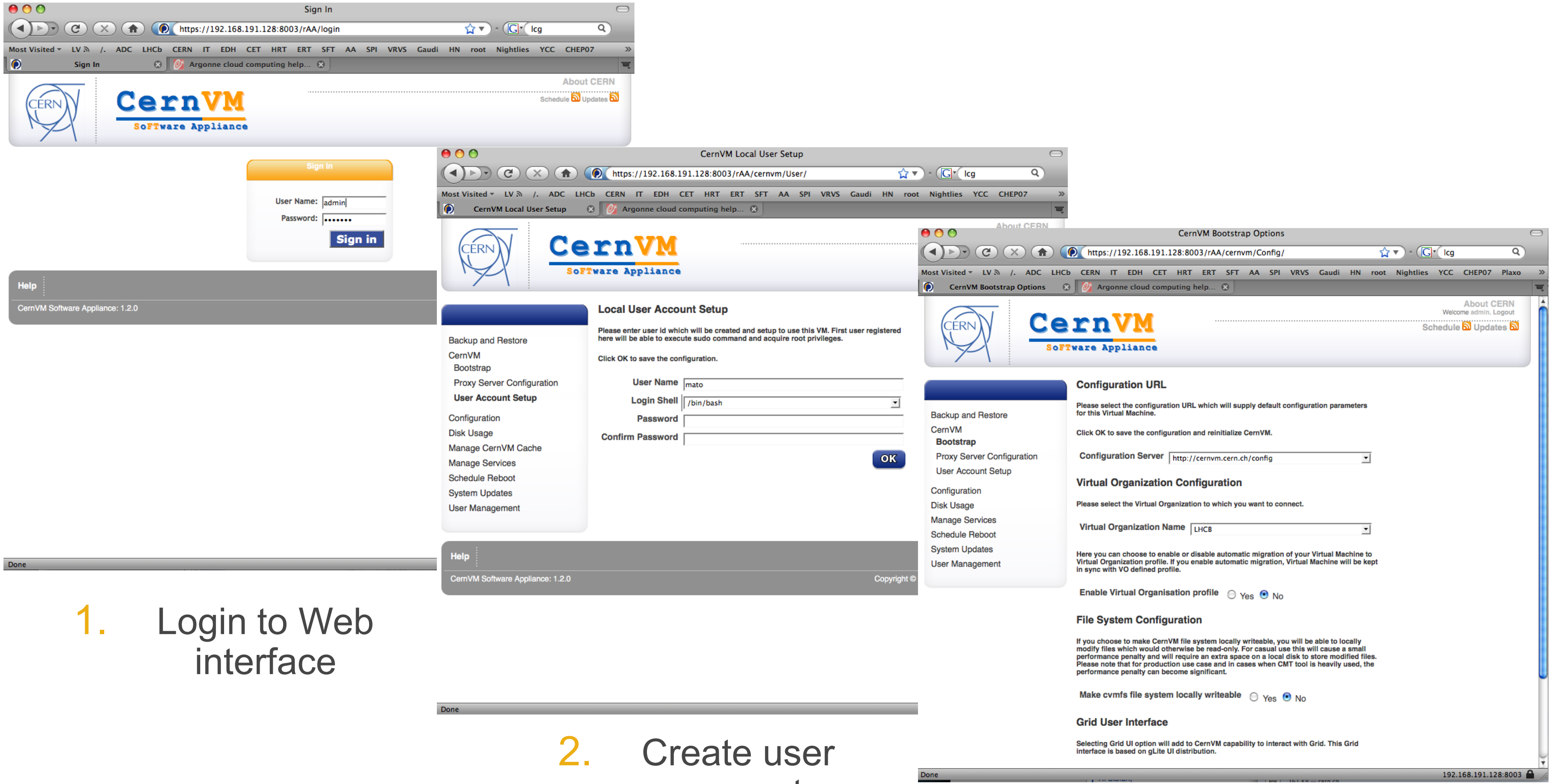
- Install once (on stratum zero)
  - Files appear everywhere across WLCG
- This can be many days faster
- Hopefully less variation across sites
  - Common path /cvmfs/…
  - Very few variables: Cache size, Squid QOS
  - Same install bugs every where – fix once
- Some sites are struggling to provide scalable NFS/AFS
  - One shared area read by every batch worker
  - It scales better than NFS/AFS

Part #3: Contextualization



- There are several ways to contextualize CernVM
  - ✓ Web UI (for individual user)
  - ✓ CernVM Contextualization Agent
  - ✓ Hepix CDROM method
  - ☞ Amazon EC2 API user_data method

As easy as 1,2,3



Sign In
https://192.168.191.128:8003/rAA/login
Sign In
User Name: admin
Password: •••••••
Sign in
Help
CernVM Software Appliance: 1.2.0
Done

CernVM Local User Setup

https://192.168.191.128:8003/rAA/cernvm/User/

Backup and Restore
CernVM
Bootstrap
Proxy Server Configuration
User Account Setup
Configuration
Disk Usage
Manage CernVM Cache
Manage Services
Schedule Reboot
System Updates
User Management

Local User Account Setup

Please enter user id which will be created and setup to use this VM. First user registered here will be able to execute sudo command and acquire root privileges.

Click OK to save the configuration.

User Name: mato
Login Shell: /bin/bash
Password:
Confirm Password:

OK

CernVM Software Appliance: 1.2.0 Copyright ©

CernVM Bootstrap Options

https://192.168.191.128:8003/rAA/cernvm/Config/

Welcome admin. Logout

Configuration URL

Please select the configuration URL which will supply default configuration parameters for this Virtual Machine.

Click OK to save the configuration and reinitialize CernVM.

Configuration Server: http://cernvm.cern.ch/config

Virtual Organization Configuration

Please select the Virtual Organization to which you want to connect.

Virtual Organization Name: LHCB

Here you can choose to enable or disable automatic migration of your Virtual Machine to Virtual Organization profile. If you enable automatic migration, Virtual Machine will be kept in sync with VO defined profile.

Enable Virtual Organisation profile ○ Yes ● No

File System Configuration

If you choose to make CernVM file system locally writeable, you will be able to locally modify files which would otherwise be read-only. For casual use this will cause a small performance penalty and will require an extra space on a local disk to store modified files. Please note that for production use case and in cases when CMT tool is heavily used, the performance penalty can become significant.

Make cvmfs file system locally writeable ○ Yes ● No

Grid User Interface

Selecting Grid UI option will add to CernVM capability to interact with Grid. This Grid interface is based on gLite UI distribution.

Done 192.168.191.128:8003

1. Login to Web interface
2. Create user account
3. Select experiment, appliance flavor and preferences

EC2 Contextualization

- Basic principles:
  - Owner of CernVM instance can contextualize and configure it to run arbitrary service as unprivileged user
  - Site can use HEPIX method to inject monitoring and accounting hooks without functionally modifying the image
- The contextualization is based on rPath amiconfig package extended with CernVM plugin
  - This tool will execute at boot time (before network services are available), parse user data and look for python style configuration blocks.
  - If match is found the corresponding plugin will process the options and execute configuration steps if needed.
- For more info on CernVM contextualization using EC2 API, see:https://cernvm.cern.ch/project/trac/cernvm/wiki/EC2Contextualization

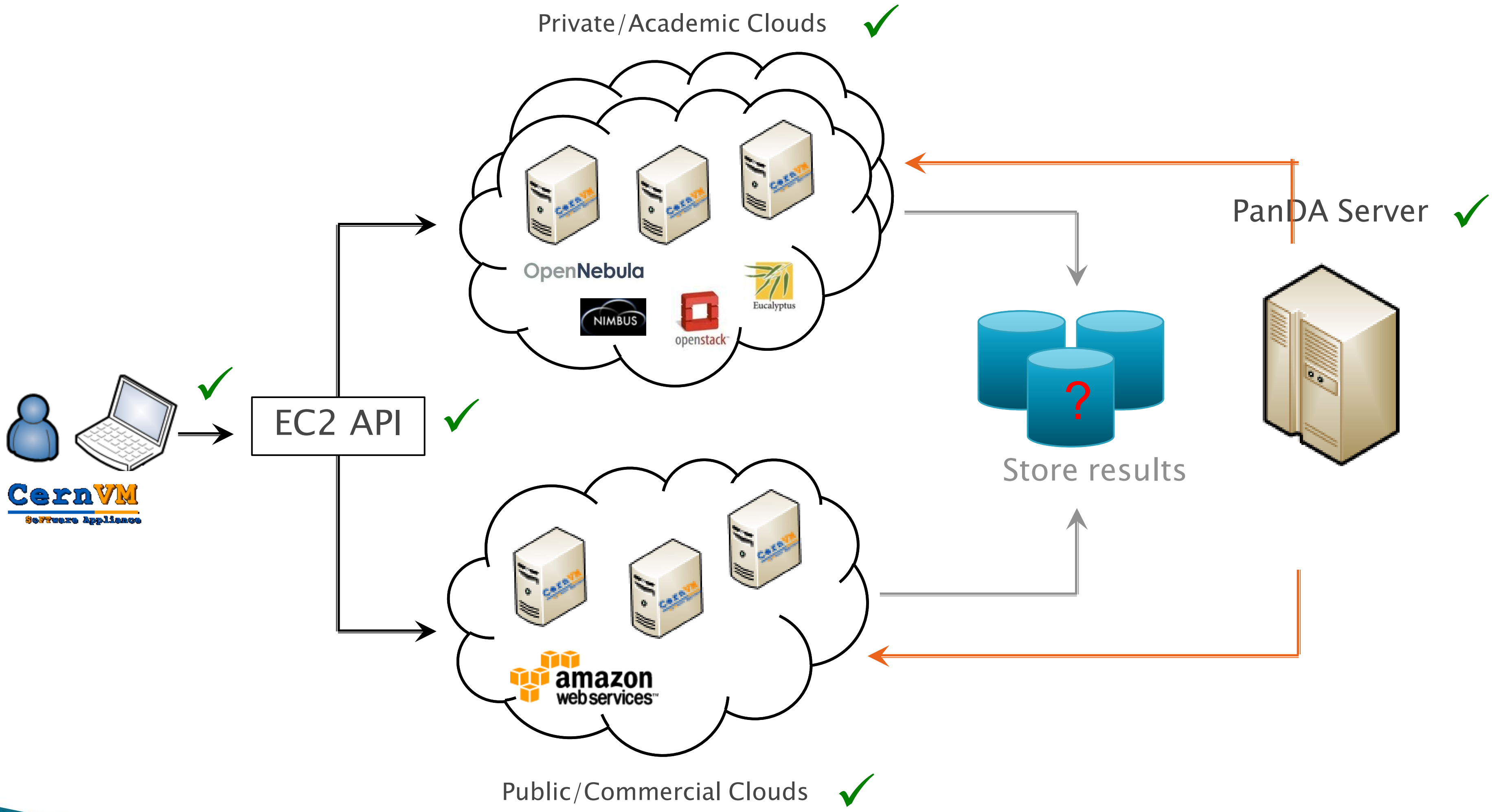
user_data example

```
[cernvm]
# list of ',' separated organizations/experiments (lowercase)
organisations = cms
# list of ',' separated repositories (lowercase)
repositories = cms,grid
# list of ',' separated user accounts to create
<user:group:[password]>
users = cms:cms:
# CVMFS HTTP proxy
proxy = http://<host>:<port>;DIRECT
# install extra conary group
group_profile = group-cms
# script to be executed as given user: <user>:/path/to/script.sh
contextualization_command = cms:/path/to/script.sh
# list of ',' seperated services to start
services = <list>
# extra environment variables to define
environment = CMS_SITECONFIG=EC2,CMS_ROOT=/opt/cms
```

Now we have CernVM OS, FS, Contextualization, Cloud API, …

What’s next?

Run PanDA (ATLAS) in CernVM on the Clouds



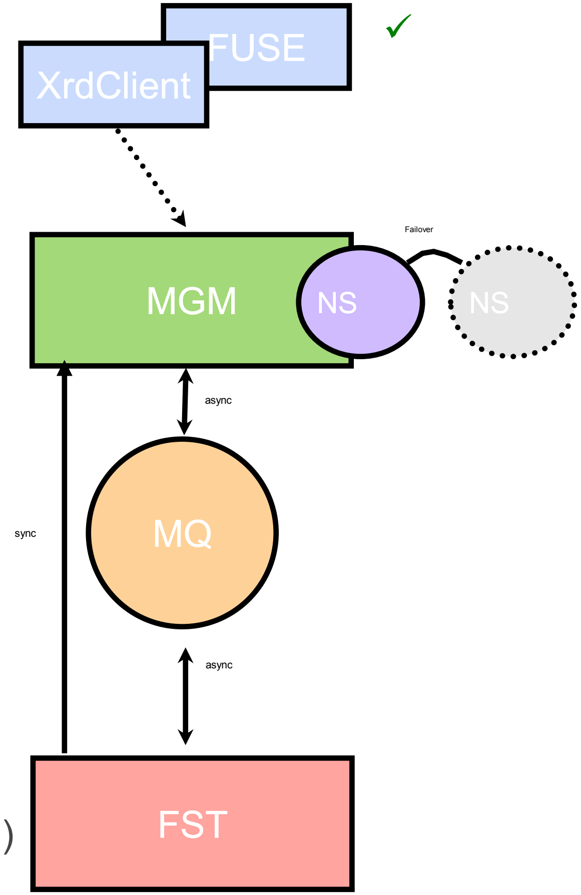
EOS: Scalable Service Architecture

Clients
XROOT client & FUSE
KRB5 + X509 authenticated

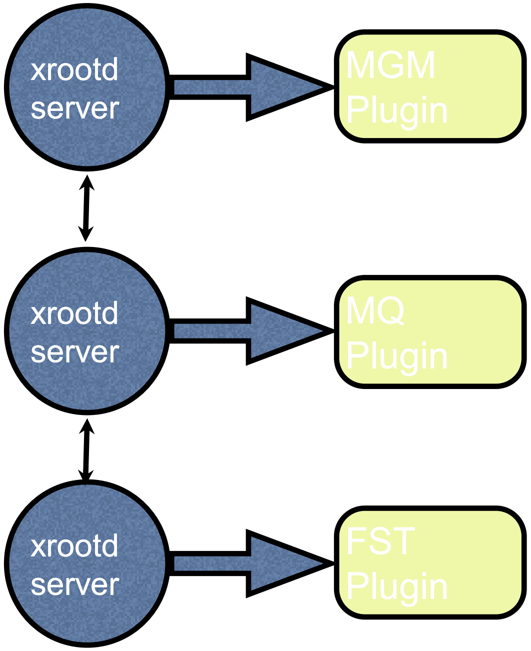
Management Server
Pluggable Namespace, Quota, ACLs,HA
Strong Authentication
Capability Engine
File Placement
File Location

Message Queue
Service State Messages
File Transaction Reports

File Storage
File & File Meta Data Store
Capability Authorization
File & Block Disk Error Detection (Scrubbing)
Layout Plugins [currently Raid-1(n)]



Implemented as plugins in xrootd



Andreas Peters IT/DM

```
[root@vmbsq090700 ~]# df /eos
Filesystem           1K-blocks      Used Available Use% Mounted on
eos                  3413376370848  23790724 3413352580124   1%
/eos
```

Using cernvm-tools

Listing available images:

```
[pbuncic@localhost ~]$ cvm --region CERN -H ls -i
AMI          LOCATION                           STATE     VISIBILITY ARCH TYPE
ami-00000008 hepix_sl55_x86_64_kvm              available Public     i386 machine
ami-00000002 cernvm232_head_slc5_x86_64_kvm     available Public     i386 machine
ami-00000004 cernvm231_slc5_x86_64_kvm          available Public     i386 machine
ami-00000003 cernvm232_batch_slc5_x86_64_kvm    available Public     i386 machine
ami-00000010 lxdev_slc6_quattor_slc6_x86_64_kvm available Public     i386 machine
```

Support for multiple regions:

```
[pbuncic@localhost ~]$ cvm --region EC2 -H ls -i ami-5c3ec235
AMI          LOCATION
STATE     VISIBILITY ARCH   TYPE
ami-5c3ec235 download.cernvm.cern.ch.s3.amazonaws.com/cernvm-2.3.1-
x86_64_1899.img.manifest.xml available Public     x86_64 machine
```

Initializing Credentials

Proxy certificate

```
[pbuncic@localhost ~]$ lcg grid-proxy-init
Your identity: /DC=ch/DC=cern/OU=computers/CN=pilot/copilot.cern.ch
Creating proxy ................................................ Done
Your proxy is valid until: Thu May 19 06:43:50 2011

[pbuncic@localhost ~]$ lcg grid-proxy-info
subject  : /DC=ch/DC=cern/OU=computers/CN=pilot/copilot.cern.ch/CN=1766683191
issuer   : /DC=ch/DC=cern/OU=computers/CN=pilot/copilot.cern.ch
identity : /DC=ch/DC=cern/OU=computers/CN=pilot/copilot.cern.ch
type     : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 512 bits
path     : /tmp/x509up_u500
timeleft : 11:59:56
```

- This proxy certificate is time limited and authorized only to
  - Request jobs from dedicated PanDA queue
  - Write (but not delete) files in a given EOS directory

LXCloud

Starting contextualized CernVM images on lxCloud:

```
[pbuncic@localhost ~]$ cvm --region CERN run ami-00000003 --proxy --template panda-wn:1
r-47a5402e predrag  default i-195 ami-00000003 128.142.192.62 128.142.192.62 pending
default 0 m1.small 1970-01-01T01:00:00+01:00 default

[pbuncic@localhost ~]$ cvm --region CERN -H ls
ID    RID     OWNER   GROUP   DNS            STATE   KEY     TYPE
i-195 default predrag default 128.142.192.62 running default m1.small

[pbuncic@localhost ~]$ cvm --region CERN -H ls
ID    RID     OWNER   GROUP   DNS            STATE   KEY     TYPE
i-195 default predrag default 128.142.192.62 running default m1.small
i-196 default predrag default 128.142.192.63 running default m1.small
i-197 default predrag default 128.142.192.64 running default m1.small
i-198 default predrag default 128.142.192.65 running default m1.small
i-199 default predrag default 128.142.192.66 pending default m1.small
i-200 default predrag default 128.142.192.67 pending default m1.small
i-201 default predrag default 128.142.192.52 pending default m1.small
i-202 default predrag default 128.142.192.53 pending default m1.small
i-203 default predrag default 128.142.192.54 pending default m1.small
i-204 default predrag default 128.142.192.55 pending default m1.small
i-205 default predrag default 128.142.192.56 pending default m1.small
i-206 default predrag default 128.142.192.57 pending default m1.small
```

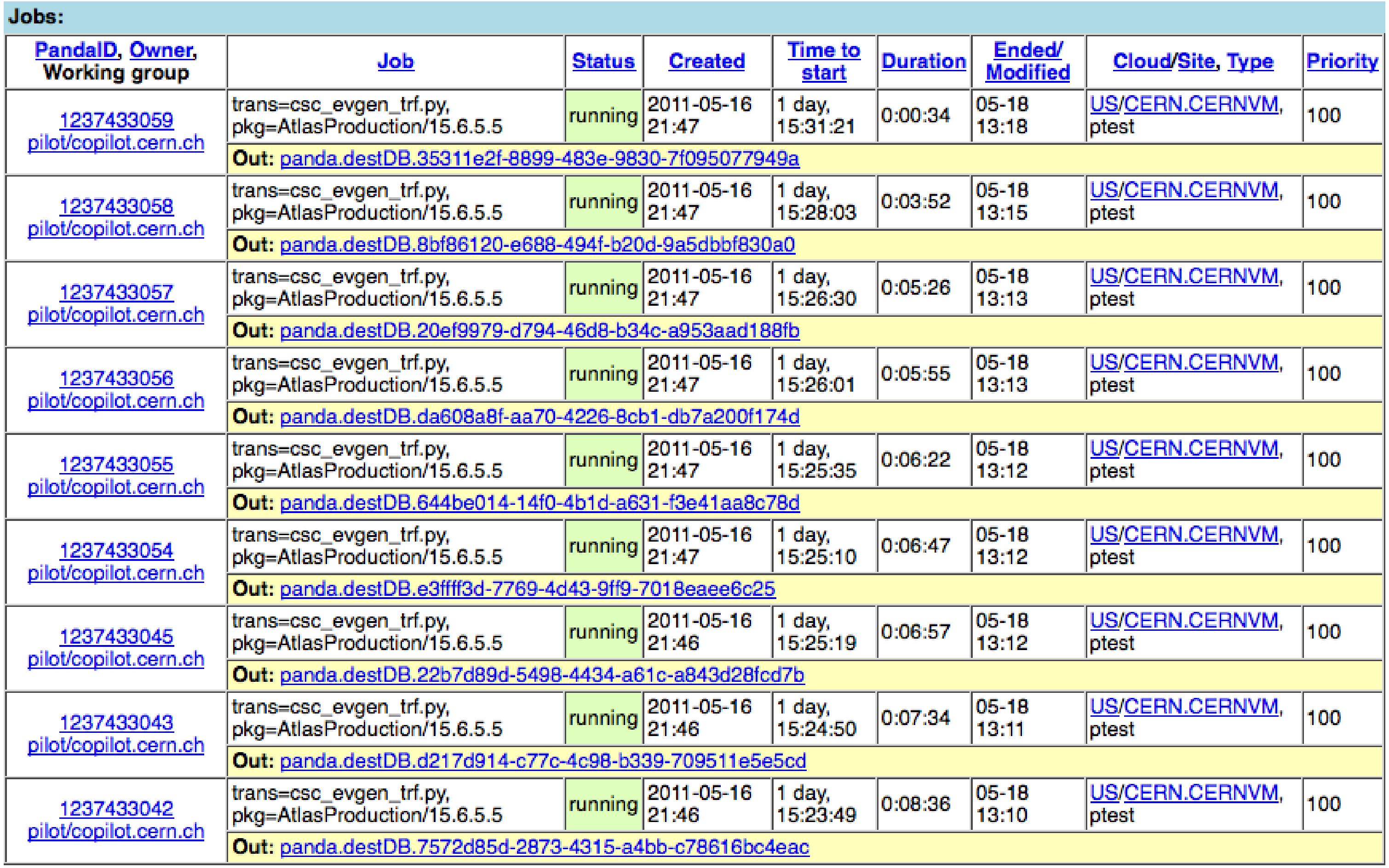
Amazon EC2

Starting more contextualized CernVM images on EC2:

```
[pbuncic@localhost ~]$ cvm run ami-5c3ec235 -g default -t m1.large --kernel aki-9800e5f1 -
-key ami --proxy --template panda-wn:10
r-ad962dc1 392941794136  default i-f3b04a9d ami-5c3ec235   pending ami 0 m1.large
r-ad962dc1 392941794136  default i-f1b04a9f ami-5c3ec235   pending ami 1 m1.large
r-ad962dc1 392941794136  default i-cfb04aa1 ami-5c3ec235   pending ami 2 m1.large
r-ad962dc1 392941794136  default i-cdb04aa3 ami-5c3ec235   pending ami 3 m1.large
....

[pbuncic@localhost ~]$ cvm --region EC2 -H ls
ID         RID        OWNER        GROUP   DNS                STATE   KEY TYPE
i-f3b04a9d r-ad962dc1 392941794136 default ec2-50-16-144-41   running ami m1.large
i-f1b04a9f r-ad962dc1 392941794136 default ec2-75-101-214-247 running ami m1.large
i-cfb04aa1 r-ad962dc1 392941794136 default ec2-184-72-183-26  running ami m1.large
i-cdb04aa3 r-ad962dc1 392941794136 default ec2-184-73-56-72   running ami m1.large
i-cbb04aa5 r-ad962dc1 392941794136 default ec2-50-16-32-51    running ami m1.large
i-c9b04aa7 r-ad962dc1 392941794136 default ec2-75-101-184-46  running ami m1.large
i-c7b04aa9 r-ad962dc1 392941794136 default ec2-50-19-38-225   running ami m1.large
i-c5b04aab r-ad962dc1 392941794136 default ec2-50-16-105-241  running ami m1.large
i-c3b04aad r-ad962dc1 392941794136 default ec2-174-129-86-61  running ami m1.large
i-c1b04aaf r-ad962dc1 392941794136 default ec2-50-19-9-47     running ami m1.large
```

PanDA Monitor



Jobs:

PandaID, Owner, Working group	Job	Status	Created	Time to start	Duration	Ended/ Modified	Cloud/Site, Type	Priority
1237433059 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:31:21	0:00:34	05-18 13:18	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.35311e2f-8899-483e-9830-7f095077949a							
1237433058 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:28:03	0:03:52	05-18 13:15	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.8bf86120-e688-494f-b20d-9a5dbbf830a0							
1237433057 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:26:30	0:05:26	05-18 13:13	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.20ef9979-d794-46d8-b34c-a953aad188fb							
1237433056 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:26:01	0:05:55	05-18 13:13	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.da608a8f-aa70-4226-8cb1-db7a200f174d							
1237433055 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:25:35	0:06:22	05-18 13:12	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.644be014-14f0-4b1d-a631-f3e41aa8c78d							
1237433054 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:47	1 day, 15:25:10	0:06:47	05-18 13:12	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.e3ffff3d-7769-4d43-9ff9-7018eaee6c25							
1237433045 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:46	1 day, 15:25:19	0:06:57	05-18 13:12	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.22b7d89d-5498-4434-a61c-a843d28fcd7b							
1237433043 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:46	1 day, 15:24:50	0:07:34	05-18 13:11	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.d217d914-c77c-4c98-b339-709511e5e5cd							
1237433042 pilot/copilot.cern.ch	trans=csc_evgen_trf.py, pkg=AtlasProduction/15.6.5.5	running	2011-05-16 21:46	1 day, 15:23:49	0:08:36	05-18 13:10	US/CERN.CERNVM, ptest	100
	Out: panda.destDB.7572d85d-2873-4315-a4bb-c78616bc4eac							

Summary

- Described the three elements that constitutes the CernVM image
  - Minimal Linux OS (SL5)
  - CernVM-FS - HTTP network file system optimized for jus in time delivery of experiment software
    → Can also be used independently of CernVM
  - Flexible configuration and contextualization mechanism based on public Cloud API
- User environment petty well understood, evolving towards a job hosting environment (grid, cloud, volunteering computing)
- Testing CernVM on Amazon-EC2 and LXCloud
  - A way to simplify software deployment and jump on the Cloud-wagon