CERN IT Department

Building packages at CERN

FROM SOURCES TO RPMS

Presented by Thomas Oulevey / @thomasnomas

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Agenda

CERN IT Department

- Introduction,
- Koji / mock / mash,
- Repositories management,
- Users' feedback,
- Limitations,
- Next challenges,
- Questions.

Introduction

Started in 2012.

2 main use cases:

- Build Scientific Linux updates

  RHEL 5 & 6 updates : from src.rpm

  Cern specific package from SVN

- Provide agile infrastructure team flexibility

  From source : src.rpm, SVN, GIT

  Available but not supported distribution e.g : fedora

  Additional repositories (swrep replacement)

Scientific Linux Cern

CERN IT Department

Problems:

- Old script based system,
- rpmbuild in use, no mock,
- a lot of manual steps for qa.

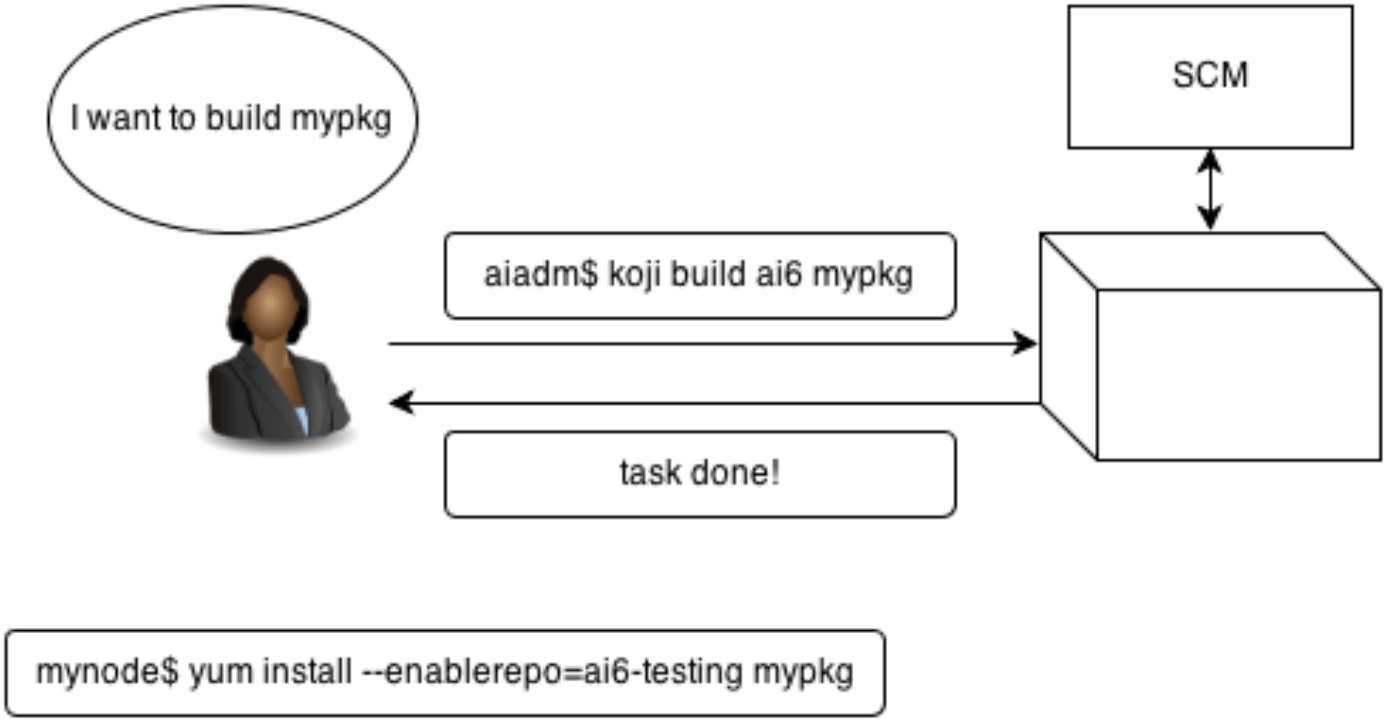
Integrate Koji gradually,

- Build updates,
- Build kernel modules (e.g : openafs),
- Build more addons (e.g : mrg),
- Better QA

Agile infrastructure

CERN IT Department

- How to provide package to puppet machine,
- Allow IT project teams to experiment with tool we don't provide in official repositories,
- Repository management.



The solution

CERN IT Department

Mock: http://fedoraproject.org/wiki/Projects/Mock

Mock creates chroots and builds packages in them. Its only task is to reliably populate a chroot and attempt to build a package in that chroot.

Koji: https://fedorahosted.org/koji/

Koji's goal is to provide a flexible, secure, and reproducible way to build software.

Mash:

mash is a tool that queries a koji buildsystem for the latest RPMs for any particular tag, and creates repositories of those RPMs

Glossary

tag: slc6-testing, slc5_devtoolset_el5_4-testing

target: a tag associated to a buildroot

package: An RPM name e.g: gcc

build: A build in Koji e.g: gcc-4.4-1.el6

to tag : associate a package to a specific tag and therefore ultimately to a repository.

The solution : gui

CERN IT Department

Wed, 30 Oct 2013 16:38:38 CET | login

CERN buildsystem

Packages SEARCH

Summary | Packages | Builds | Tasks | Tags | Build Targets | Users | Hosts | Reports | Search

Builds

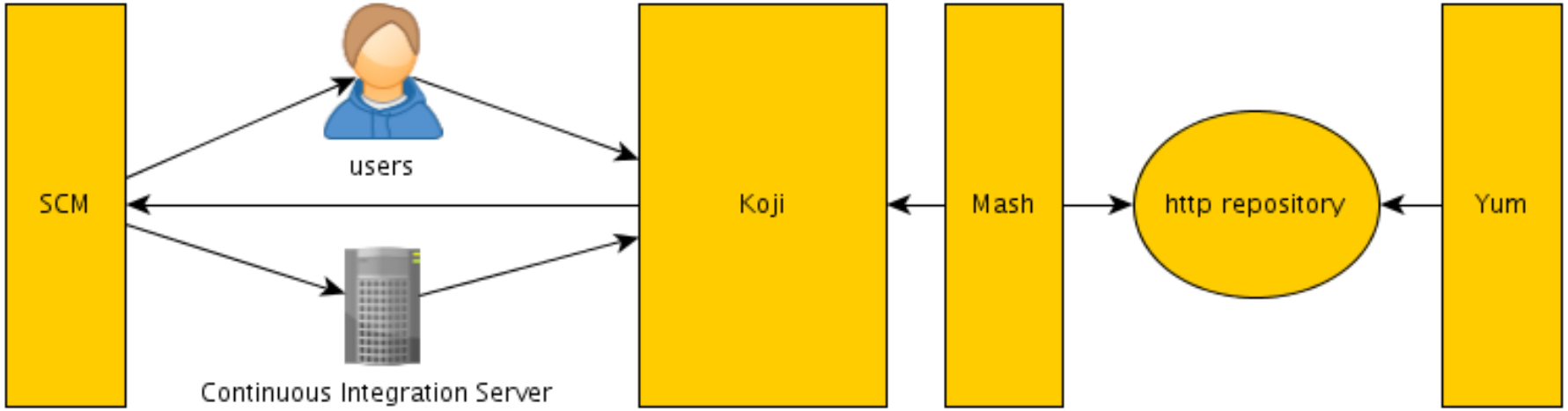
State: all Built by: everyone

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | all

Builds 1 through 50 of 8010 >>> Page: 1

ID	NVR	Built by	Finished	State
8824	flexnet_agent-4.1.3-1.el6	agonzale	2013-10-30 14:33:37	
8823	CERN-CC-LSF-glibc2.3-7.0.6-35.ai5	uschwick	2013-10-30 10:07:40	✓
8822	CERN-CC-LSF-glibc2.3-7.0.6-35.ai6	uschwick	2013-10-30 10:08:42	✓
8821	xulrunner-17.0.10-1.el6_4	jpolok	2013-10-30 10:06:19	✓
8820	spice-server-0.12.0-12.el6_4.5	jpolok	2013-10-30 09:26:22	✓
8819	postgresql-8.4.18-1.el6_4	jpolok	2013-10-30 09:36:36	✓
8818	lvm2-2.02.98-9.el6_4.2	jpolok	2013-10-30 09:28:51	✓
8817	firefox-17.0.10-1.el6_4	jpolok	2013-10-30 10:26:40	✓
8816	xulrunner-17.0.10-1.el5_10	jpolok	2013-10-30 10:07:37	✓
8815	qspice-0.3.0-56.el5_10.1	jpolok	2013-10-30 09:28:30	✓

The big picture



The solution : cli

```
# koji  add-pkg <tag> <NAME>
# koji  add-pkg <tag>-testing <NAME>
# koji  add-pkg <tag>-stable <NAME>
# koji build <tag> <NAME-RELEASE-VERSION>.src.rpm

# koji tag-pkg <tag>-stable
```

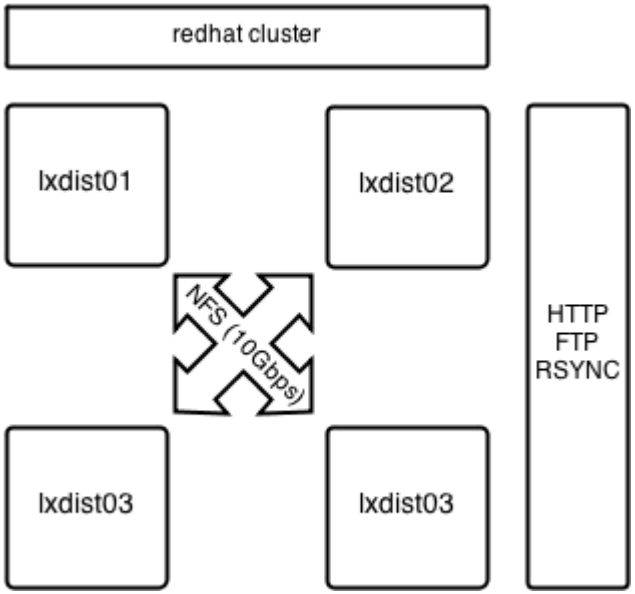
In numbers

Number of users: **98** (agile, security, castor, wlcg)

Packages built: **8010** (75% linux support, 20% AI)

Number of tags : **40**

Build hosts: 4 x Xeon L5520 with 24GB of Memory



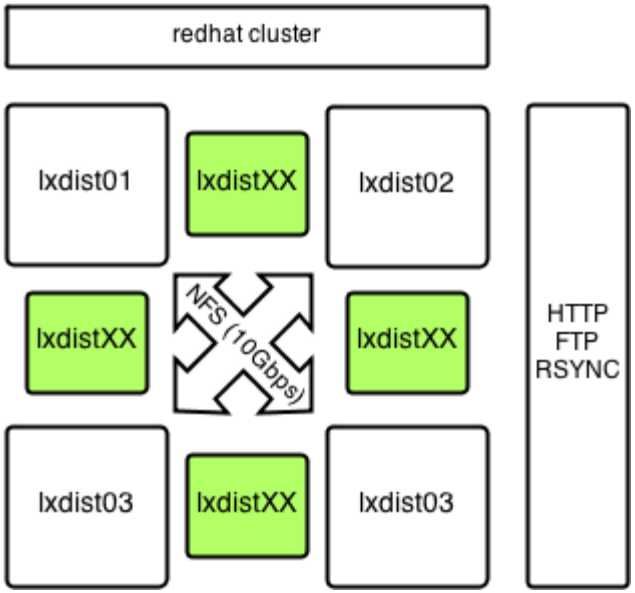
In numbers

Number of users: **98** (agile, security, castor, wlcg)

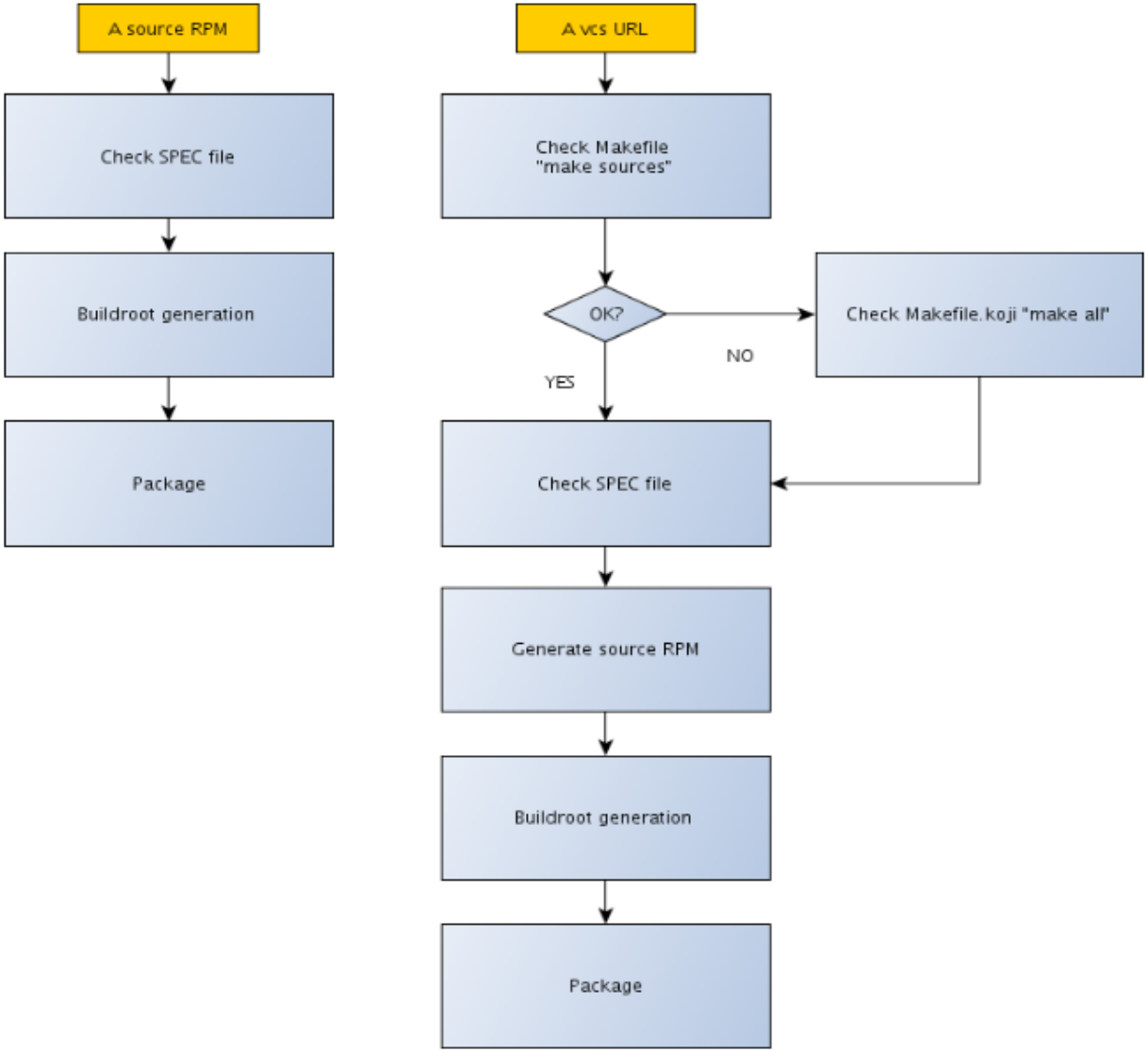
Packages built: **8010** (75% linux support, 20% AI)

Number of tags : **40**

Build hosts: 4 x Xeon L5520 with 24GB of Memory



RPM creation workflow



Koji workflow

1. Build RPM with Koji
2. 5 minutes later it appears in -qa repository
3. 4 hours later it appears in -testing repository
4. User can tag it -stable repository
5. 4 hours later it appears in -stable

One default workflow, so automatisation is easy.

Repositories

http://linuxsoft.cern.ch/internal

mash run e.g: mash slc6-testing.

Distribution through our servers.

Snapshots:

hardlinks

never delete packages

puppet module to manage the freeze (ai121yumrepo wrapper for yumrepo)

Snapshots

CERN IT Department

http://linuxsoft.cern.ch/internal/yumsnapshot/

- 20131010/
- 20131011/
- 20131012/
- 20131013/
- 20131014/
- 20131015/
- 20131016/
  - ai/
  - cern/
  - emi/
  - epel/
  - rhel/

Users feedback

- Very good feedback but...
- User frustrated. Our packaging in general need some fixing. (afs file dependencies, etc...)

- Differences between Distribution. (version of tar between 5 and 6, %if %endif)

- When SCM workflow in place developers are happy.

```
koji build ai6 git+https://git.cern.ch/myproject?#<tag>
```

Limitations

- Single namespace
  ….but no collision on package name

- Policy
  a file and service need a reload on updates.

- Kernel modules
  can’t recompile on a specific kernel
  rpmfusion solution
- No fine repository access control (cern or public).

The challenges

- Support, support, support.
  - nobody read the doc at first
  - .spec file creation, NVR questions
  - “It works on my machine...”
- Existing RPMs with no sources.
  - case by case
  - supported in koji but no advertise to users.
- Signing
  - sigul => Daemon
  - koji-plugin-sign.py => Python script.

Nice side effects

- Meet packagers.
- Better packaging guideline. RPM quality.
- Traceability.
- Repositories centralized within our infrastructure.
- More addons recompiled.

(Software Collections:
devtoolset 2.0
scl 1.0
community interest? (who says llvm))

Future

- Better policy integration with Active Directory, and enable SSO.

- Support SLC7
- Look into repository management tools:

  pulp : http://www.pulpproject.org/

  copr : https://fedorahosted.org/copr/

- Puppetize: module coded, cleanup needed.
- Keep an eye on OBS: http://openbuildservice.org/

QUESTIONS ?

HTTPS://LINUX.CERN.CH

Thank you !