RPM Building with MOCK, KOJI and MASH May 2009

White Area Lecture, IT-GD, CERN
Steve Traylen
steve.traylen@cern.ch

Overview

- RPM Basics
  - SOURCE, SPEC, SRPM and RPM
  - Dependencies.
- MOCK
  - RPM Building in a CHROOT.
- KOJI
  - Multi-User RPM Build Daemon.
- MASH
  - YUM Repository Creation
- Use and Progress for SA1 OAT Repositories

RPM Introduction

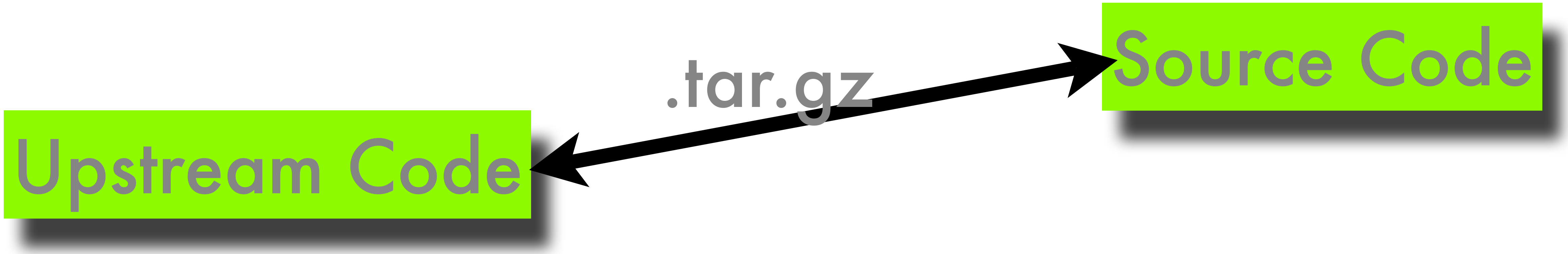
- RPM = Redhat Package Manager
  - Now RPM Package Manager
- Currently two RPM versions.
  - Redhat/Fedora = http://www.rpm.org/
  - RPMv5 = http://rpm5.org/
- The closest there is to a standard.
  - LinuxStandardBase dictates a subset of RPM.

RPM Build Process

RPM Build Process

Upstream Code

RPM Build Process

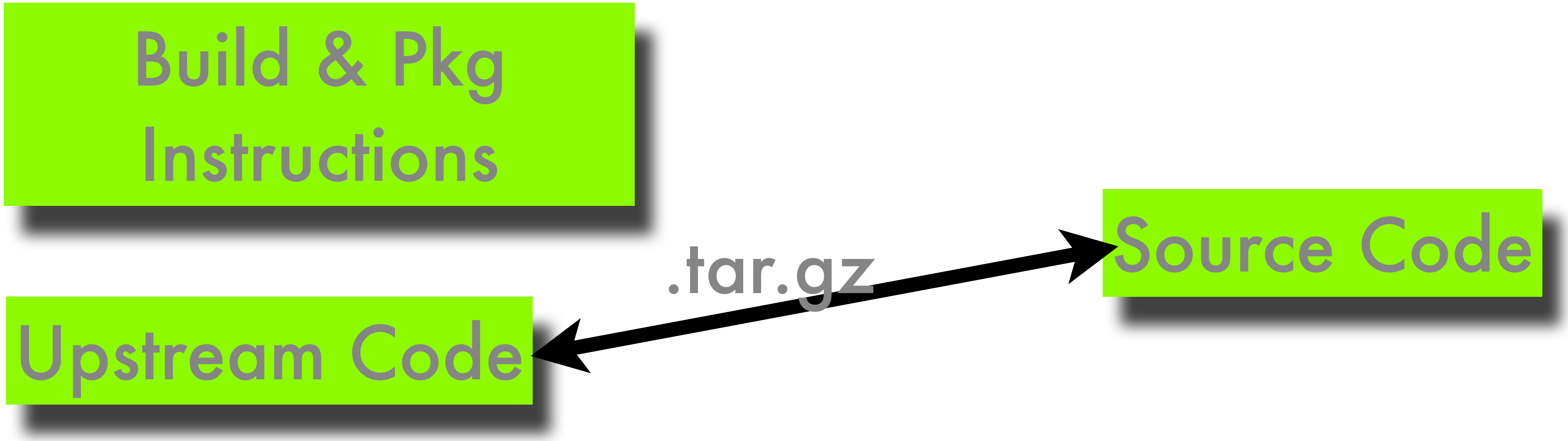


Upstream Code

.tar.gz

Source Code

RPM Build Process



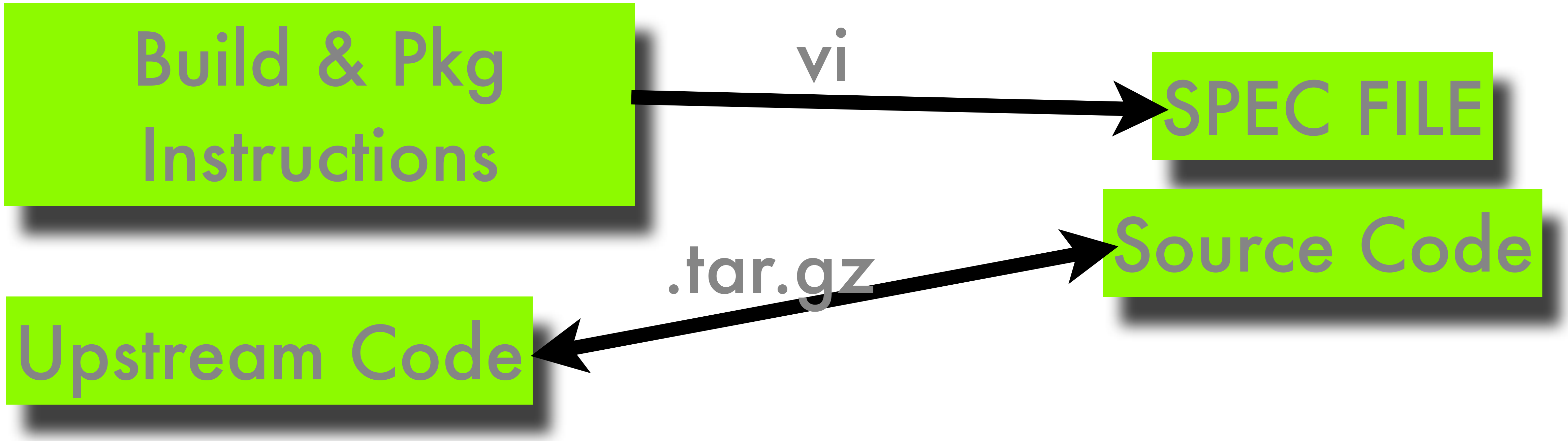
Build & Pkg Instructions

Upstream Code

.tar.gz

Source Code

RPM Build Process



Build & Pkg Instructions

vi

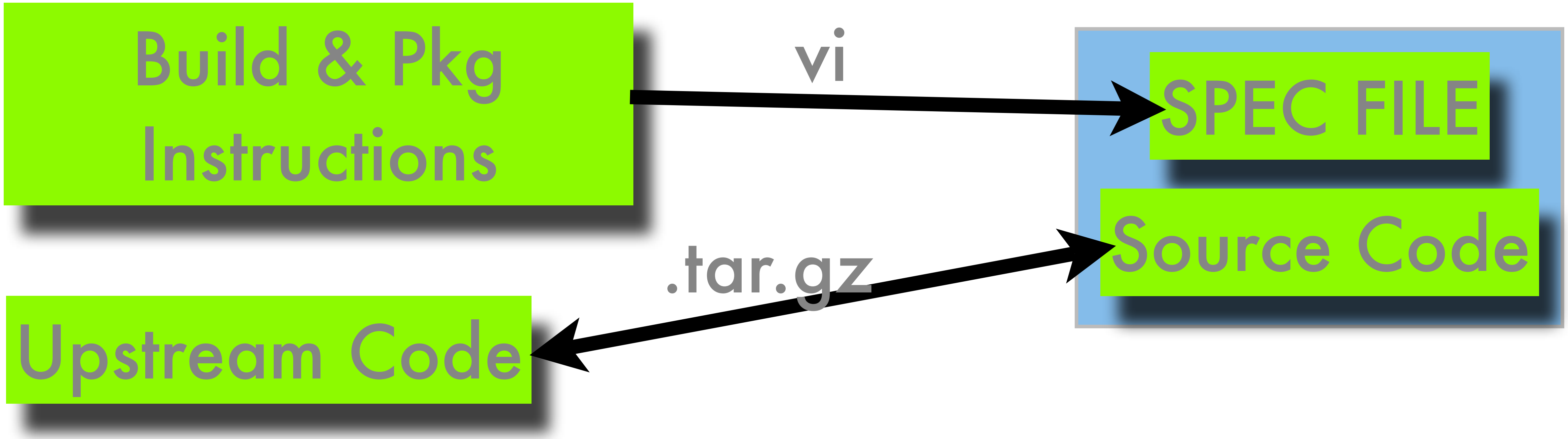
SPEC FILE

Source Code

.tar.gz

Upstream Code

RPM Build Process


Build & Pkg Instructions

vi

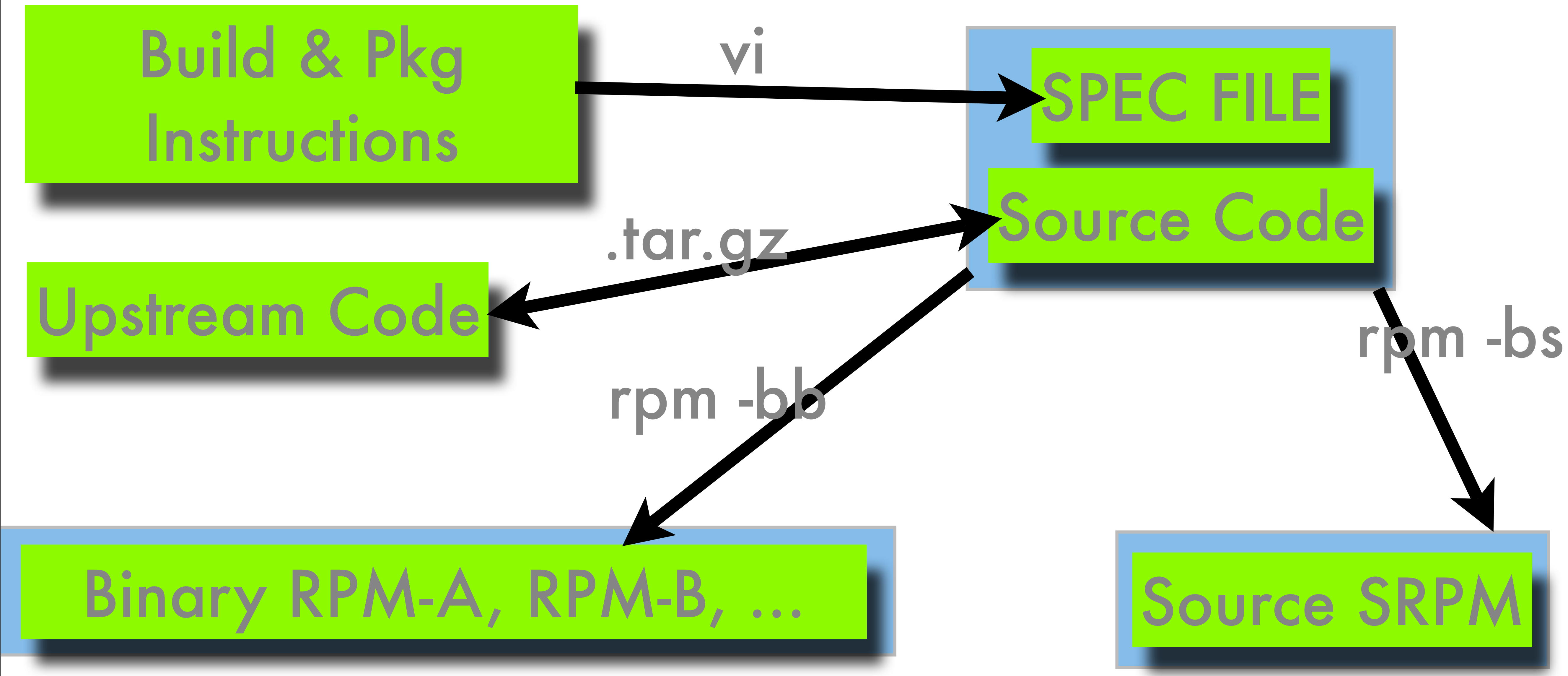
SPEC FILE

Source Code

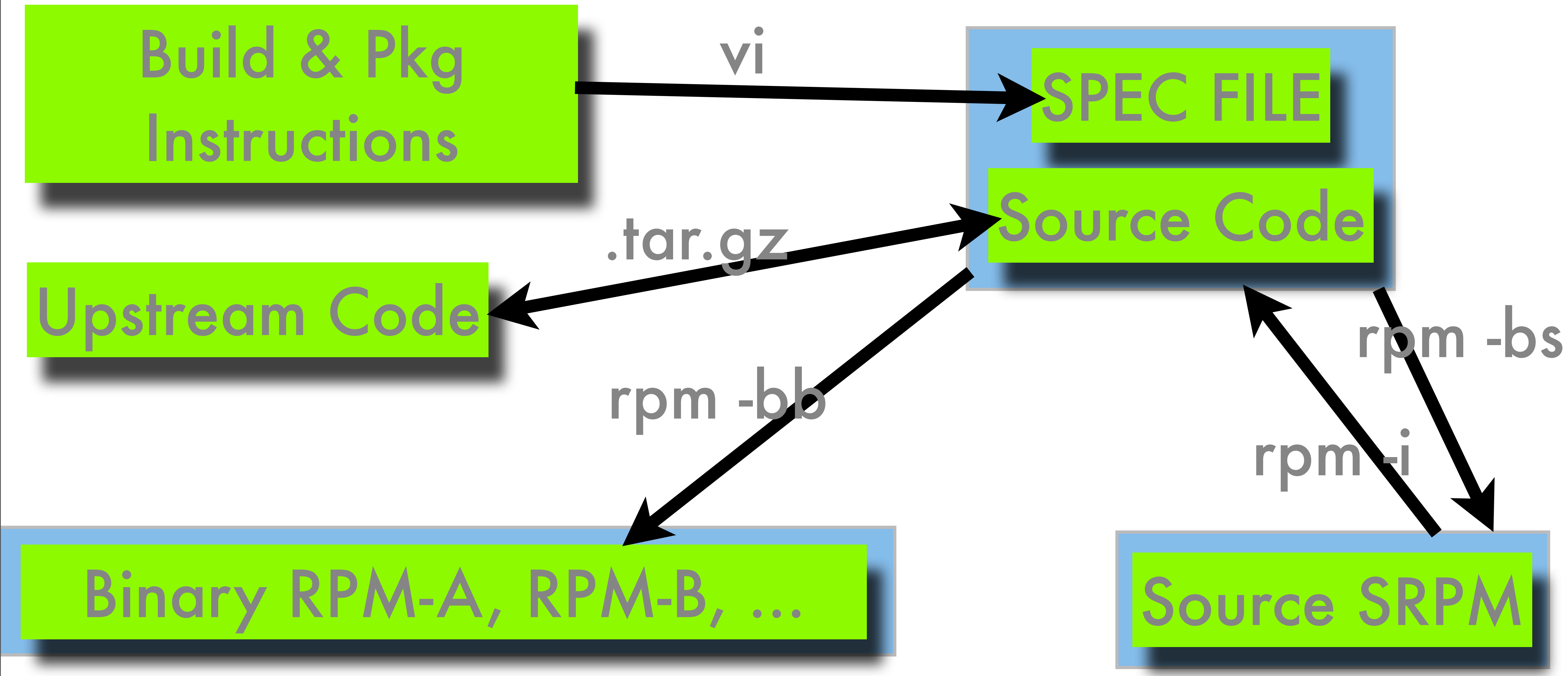
.tar.gz

Upstream Code

RPM Build Process



RPM Build Process



Build & Pkg Instructions

vi

SPEC FILE

Source Code

.tar.gz

Upstream Code

rpm -bb

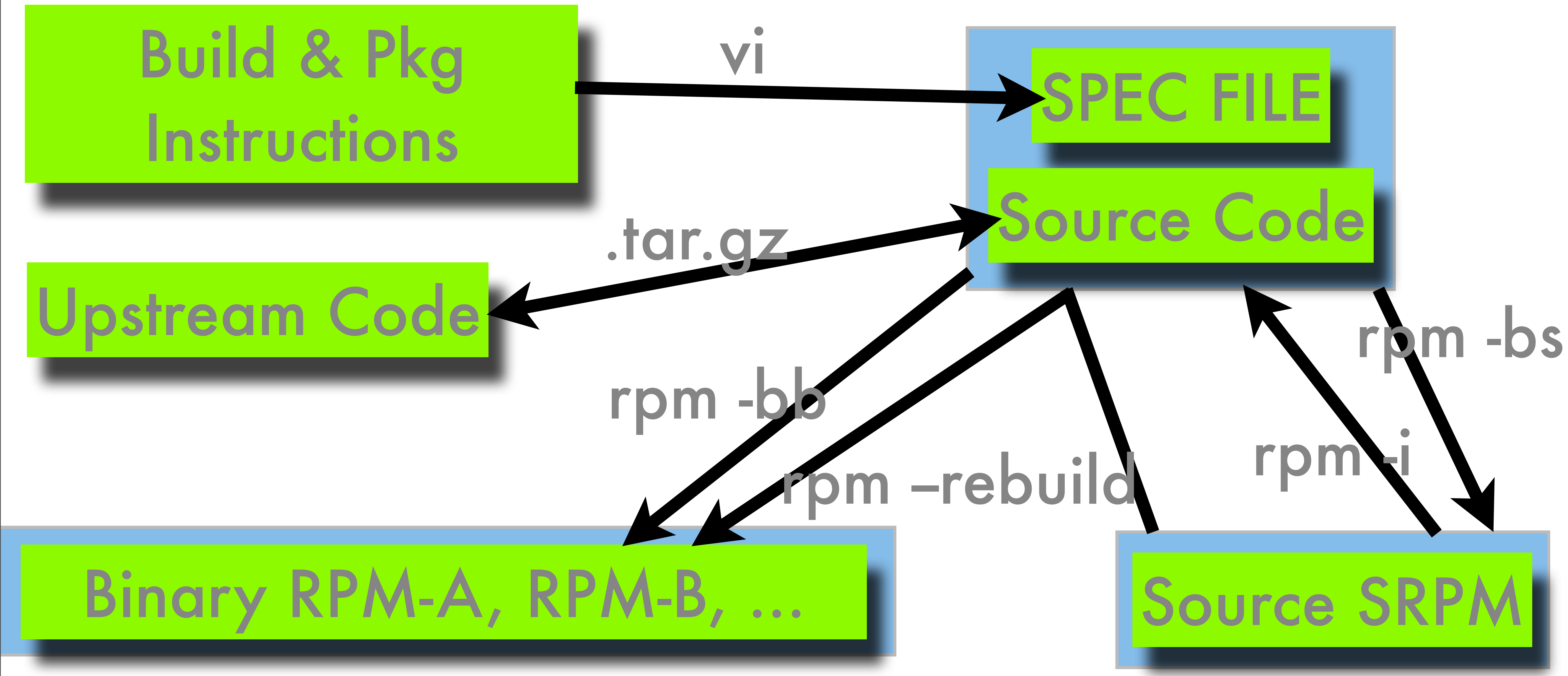
rpm -bs

rpm -i

Binary RPM-A, RPM-B, ...

Source SRPM

RPM Build Process



Build & Pkg Instructions

vi

SPEC FILE

Source Code

.tar.gz

Upstream Code

rpm -bb

rpm –rebuild

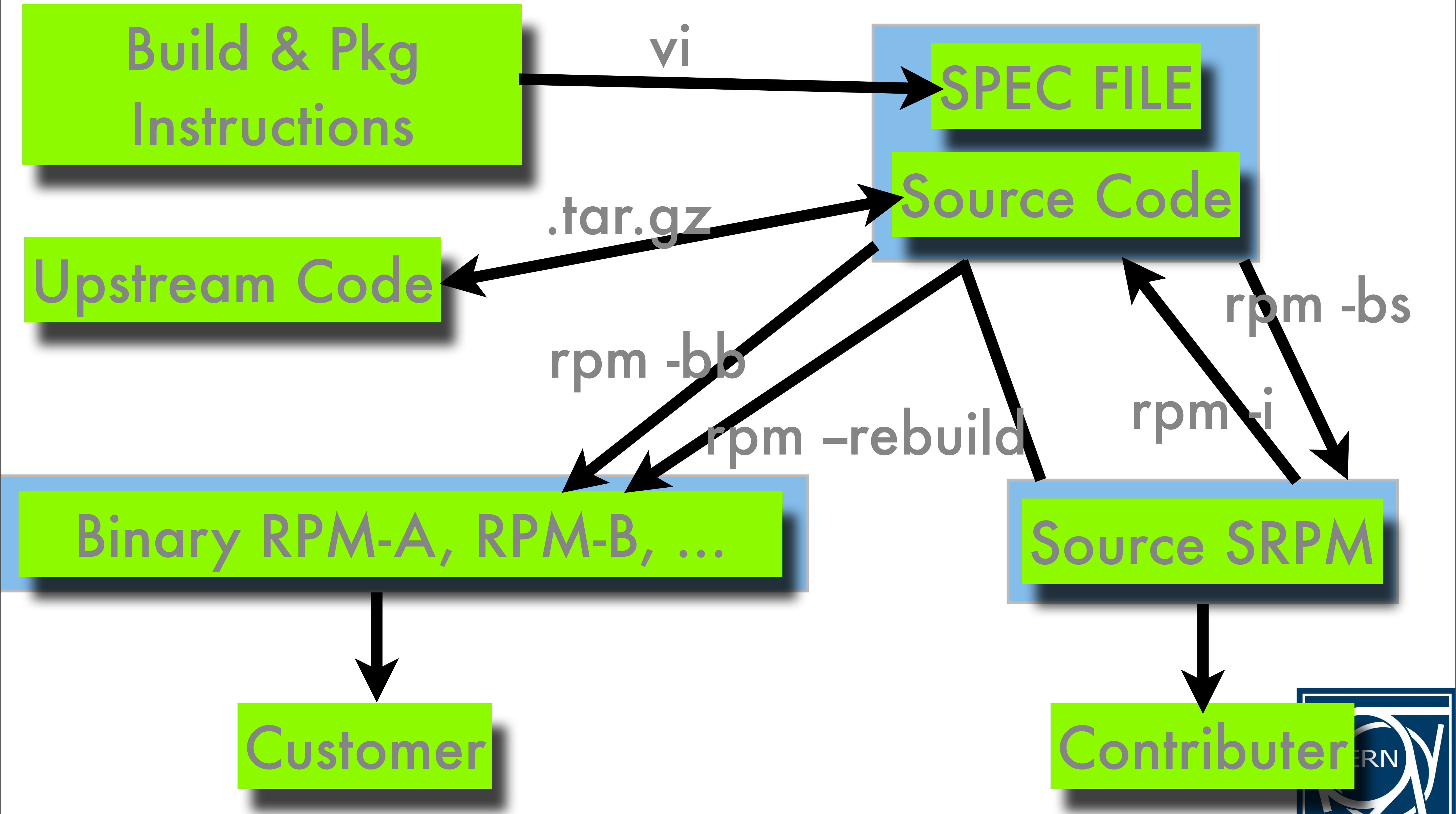
rpm -i

rpm -bs

Binary RPM-A, RPM-B, ...

Source SRPM

RPM Build Process



RPM Dependencies

- Binary RPMs list what they provide, a union of
  1. Their name e.g *openssh-client = 1.2*
  2. Their contents e.g *libkrb5.so.3*.
     - Dynamically discovered at RPM creation time.
  3. Anything specified by hand in RPM spec file.
     - e.g Sendmail and Exim provide “smtpdaemon”
- Binary RPMS require other Binary RPMs
  - Runtime requirements - Requires (or PreRequires)
  - Specified by hand or dynamically.
- Source RPMS also require Binary RPMs
  - Build time requirements - by hand always.

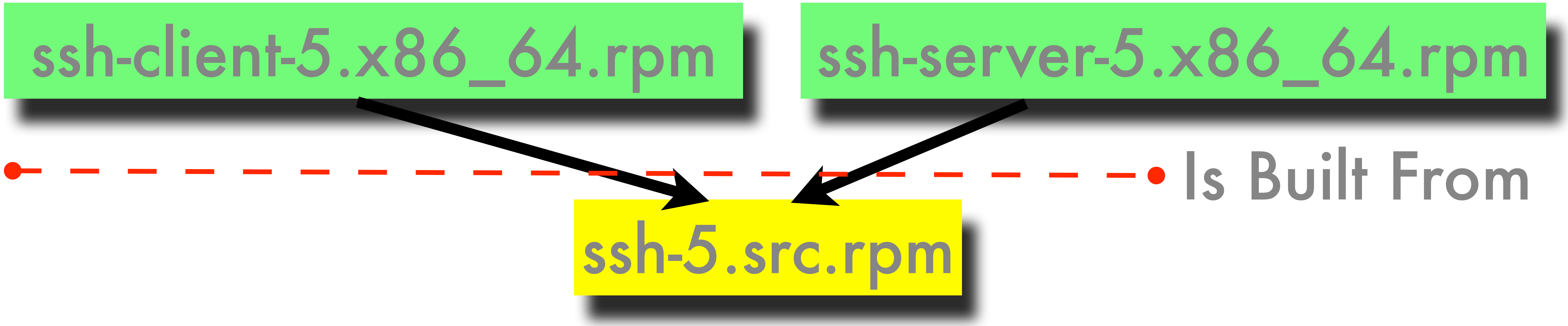
RPM Dependency Example

RPM Dependency Example

ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

RPM Dependency Example



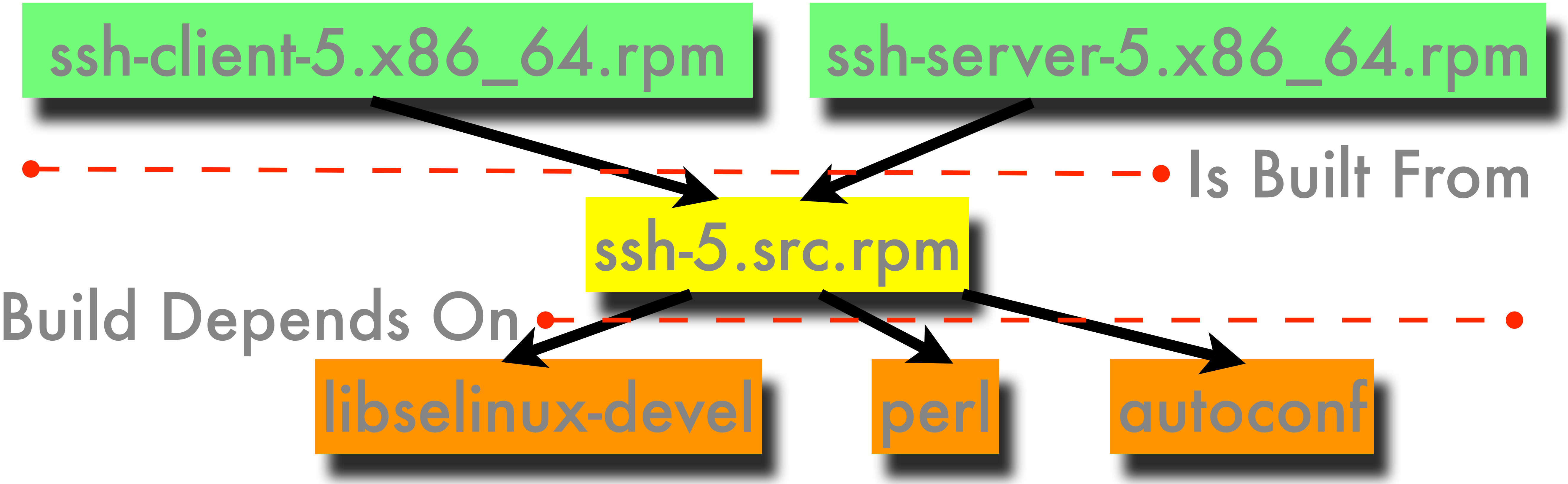
ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

Is Built From

ssh-5.src.rpm

RPM Dependency Example



ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

Is Built From

ssh-5.src.rpm

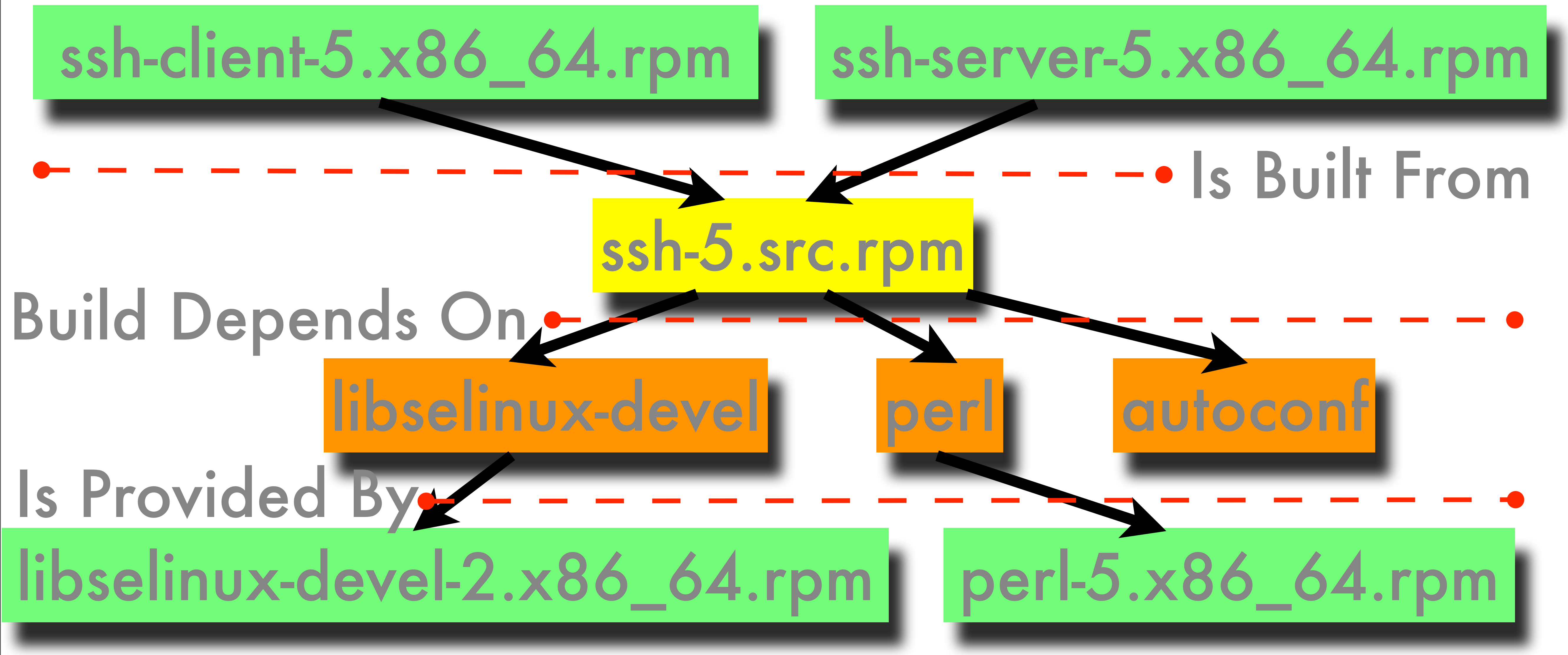
Build Depends On

libselinux-devel

perl

autoconf

RPM Dependency Example



ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

Is Built From

ssh-5.src.rpm

Build Depends On

libselinux-devel

perl

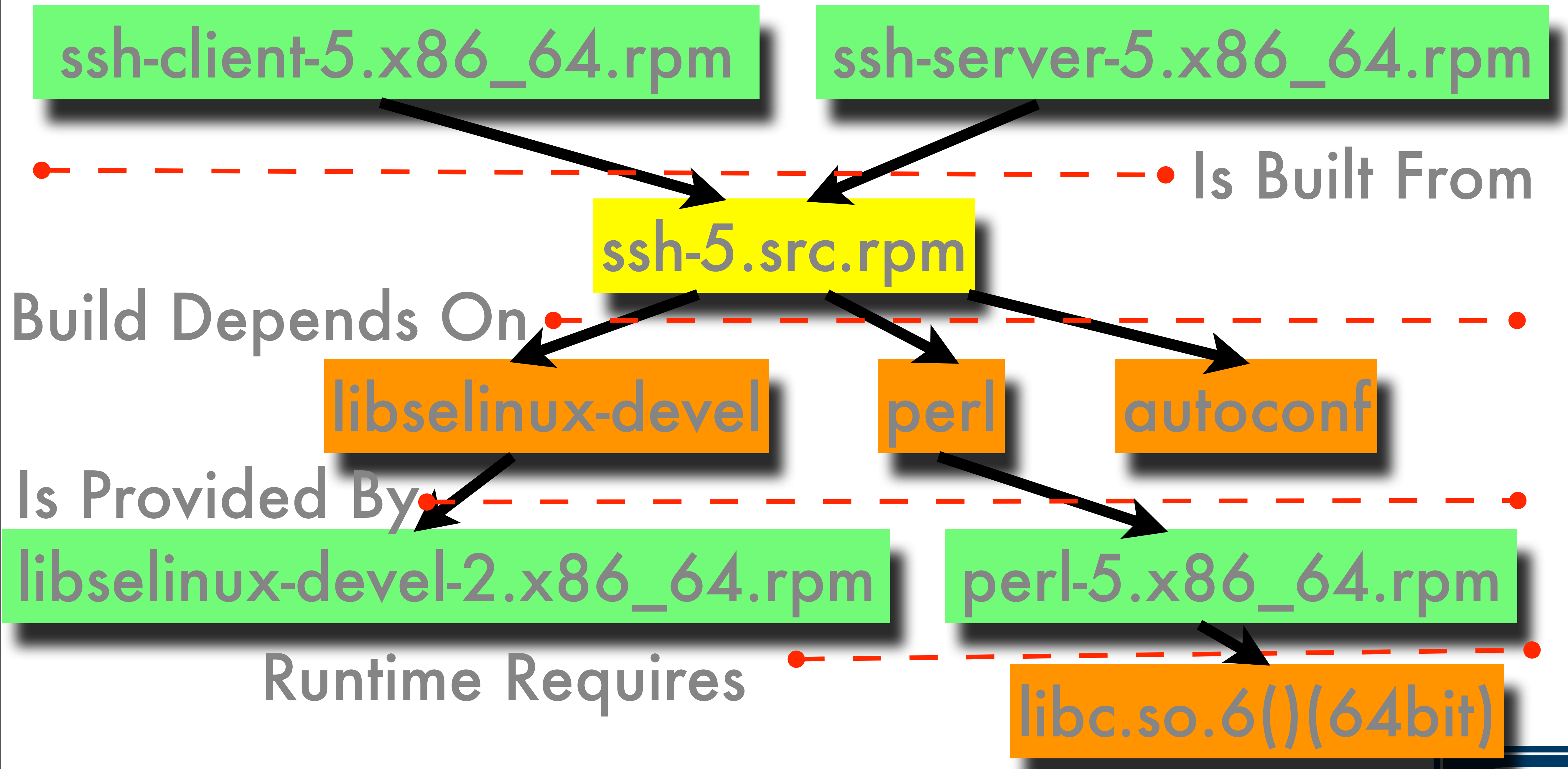
autoconf

Is Provided By

libselinux-devel-2.x86_64.rpm

perl-5.x86_64.rpm

RPM Dependency Example



ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

Is Built From

ssh-5.src.rpm

Build Depends On

libselinux-devel

perl

autoconf

Is Provided By

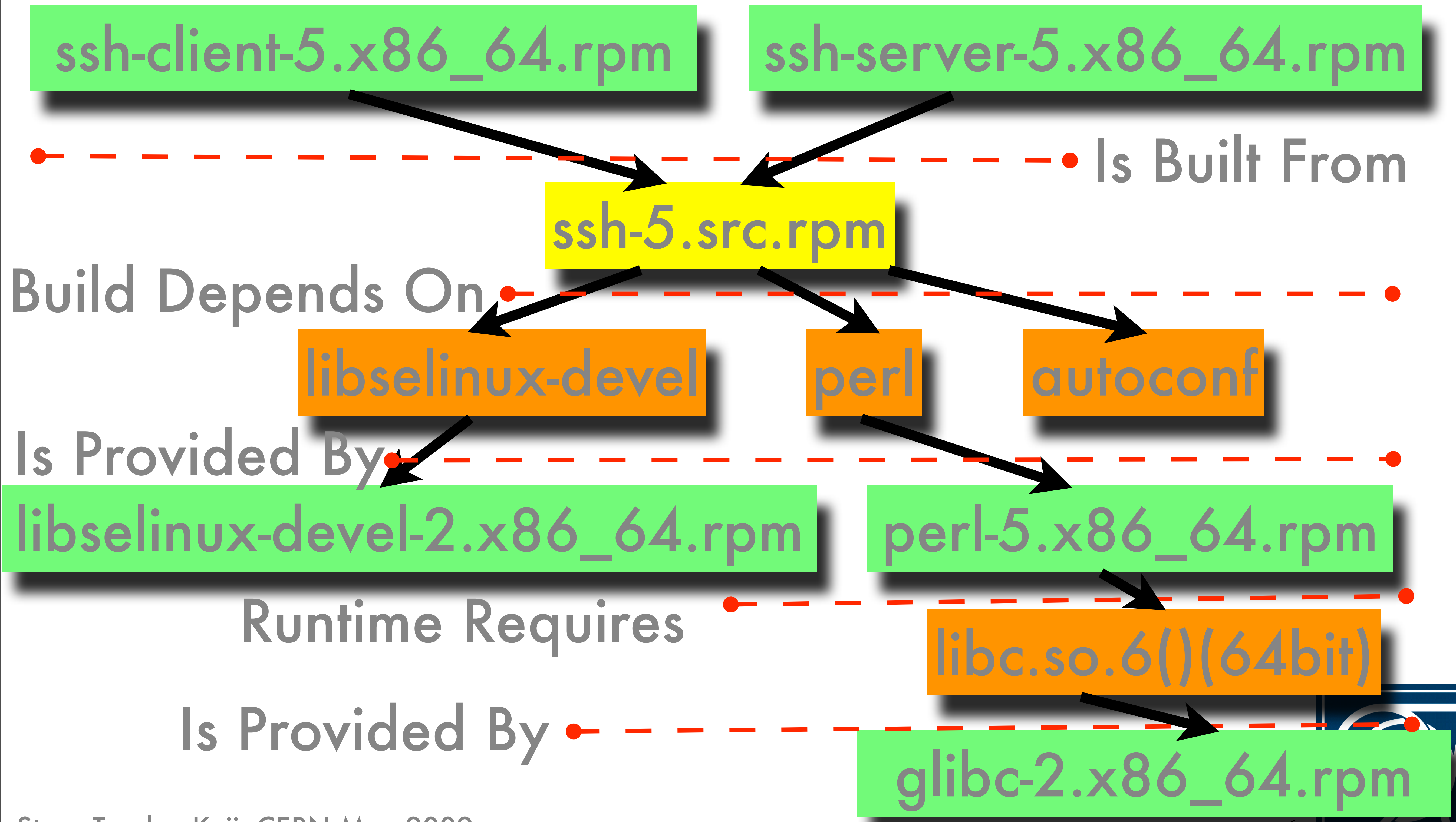
libselinux-devel-2.x86_64.rpm

perl-5.x86_64.rpm

Runtime Requires

libc.so.6()(64bit)

RPM Dependency Example



ssh-client-5.x86_64.rpm

ssh-server-5.x86_64.rpm

Is Built From

ssh-5.src.rpm

Build Depends On

libselinux-devel

perl

autoconf

Is Provided By

libselinux-devel-2.x86_64.rpm

perl-5.x86_64.rpm

Runtime Requires

libc.so.6()(64bit)

Is Provided By

glibc-2.x86_64.rpm

From SRC RPM to RPM

- Starting Point - A SRC RPM.
  - Extract BuildRequirments from SPEC File.
  - Set up a machine or build environment.
  - Ensure all build dependencies are installed.
  - Ensure all build dependencies’ dependencies are installed.
  - Build RPM. Keep the result.
- End Point - A Binary RPM.
- Considerations.
  - Ensure process is prescribed and repeatable
- Solution - MOCK.

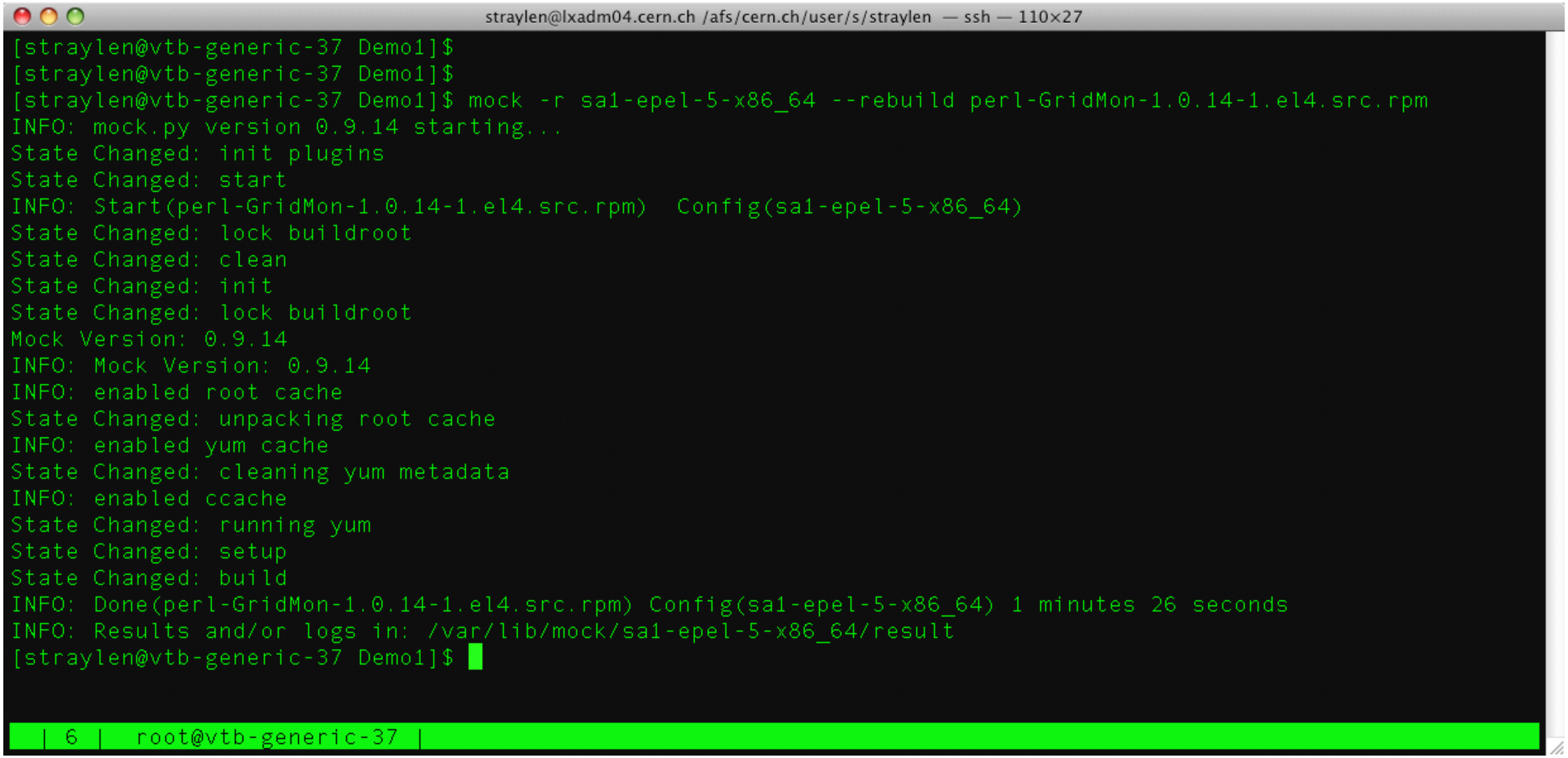
MOCK

- CHROOT creation and command execution.
  - Designed for building RPMs but anything.
  - Each chroot defined by:
    - YUM configuration - a list of repo's.
    - YUM command e.g install meta-package.
    - Lots of caching options.
- Example Use -
  - mock -r f10-s390 --init
  - mock -r f10-s390 --shell <cmd>
  - mock -r f10-s390 --rebuild openssh-2.src.rpm
    - This example produces openssh-* binary RPMS

MOCK Details as steve

- steve must be a member mock group. mock setguid.
- $ mock -r centos5-i386 --rebuild openssh1.src.rpm
  - Create a skeleton - /v/l/mock/centos5-i386
  - Populate it , e.g /etc/ , /dev/null/, /dev/urandom.
  - Install a meta package and dependencies.
    - yum --installroot /v/l/mock/centos5-i386 install buildsys
  - Examine openssh1.src.rpm - install dependencies.
    - yum --installroot /v/l/mock/centos5 install pam-devel
  - Create an account “steve” in the chroot.
    - Allows package to be built as steve.
  - Copy srpm into chroot.
  - Build in chroot rpmbuild --rebuild openssh1.src.rpm
- Copy binary RPMS out of chroot. - Job done.

DEMO 1 - MOCK


straylen@lxadm04.cern.ch /afs/cern.ch/user/s/straylen — ssh — 110×27

```
[straylen@vtb-generic-37 Demo1]$
[straylen@vtb-generic-37 Demo1]$
[straylen@vtb-generic-37 Demo1]$ mock -r sa1-epel-5-x86_64 --rebuild perl-GridMon-1.0.14-1.el4.src.rpm
INFO: mock.py version 0.9.14 starting...
State Changed: init plugins
State Changed: start
INFO: Start(perl-GridMon-1.0.14-1.el4.src.rpm)  Config(sa1-epel-5-x86_64)
State Changed: lock buildroot
State Changed: clean
State Changed: init
State Changed: lock buildroot
Mock Version: 0.9.14
INFO: Mock Version: 0.9.14
INFO: enabled root cache
State Changed: unpacking root cache
INFO: enabled yum cache
State Changed: cleaning yum metadata
INFO: enabled ccache
State Changed: running yum
State Changed: setup
State Changed: build
INFO: Done(perl-GridMon-1.0.14-1.el4.src.rpm) Config(sa1-epel-5-x86_64) 1 minutes 26 seconds
INFO: Results and/or logs in: /var/lib/mock/sa1-epel-5-x86_64/result
[straylen@vtb-generic-37 Demo1]$

  | 6 |  root@vtb-generic-37 |
```

- mock build of perl-GridMon
  - sa1-epel5 = Centos5 + DAG Repositories.

MOCK Caching.

- Lots of speed up tricks - all configurable.
  - Normal yum caching.
  - The base system is stored in a tar.gz first time.
    - A yum update is executed every time.
    - A necessary update causes .tar.gz to be replaced.
  - ccache used. i.e cache compilations
  - tmpfs used. The build machine is created in RAM.
    - It's a chroot, e.g no kernel needed. 100 MB or so.

Method	Time
rpm --rebuild ssh-5.src.rpm	1 min 4 seconds.
mock --rebuild ssh-5.src.rpm (no cache)	3 min 27 seconds.
mock --rebuild ssh-5.src.rpm (with cache)	1 min 46 seconds.

Mock Summary

- RPMS from SRPMs with a reproducible method.
  - Fantastic does this and not a lot else really.
- SA1 OAT packages
  - RPMS built in this way till recently.
  - One 64 bit SL5 vnode machine. (i.e 1/8 of machine)
  - Simple 10 line shell script to loop over build targets.
  - SL4/5 32bit/64bit RPMS built and uploaded to repo.
- Comments.
  - Its easy to have collisions if on the same box.
    - It's a management problem to keep track of it all.
    - Mock is a one person solution. --- Enter Koji.

KOJI.

- KOJI is the RPM build system for Fedora.
  - It replaced plague around F7/8.
- Its is built on top of MOCK.
- KOJI is also the build system for EPEL
  - Extra Packages for Enterprise Linux
  - Fast replacing dag, jpackage, ... it's closer to RedHat basically.
- Two models supported - can coexist.
  - KOJI owns everything e.g Fedora.
  - KOJI owns packages on top of a repo. e.g EPEL.
    - Only supported from March this year.

Fedora’s KOJI Statistics

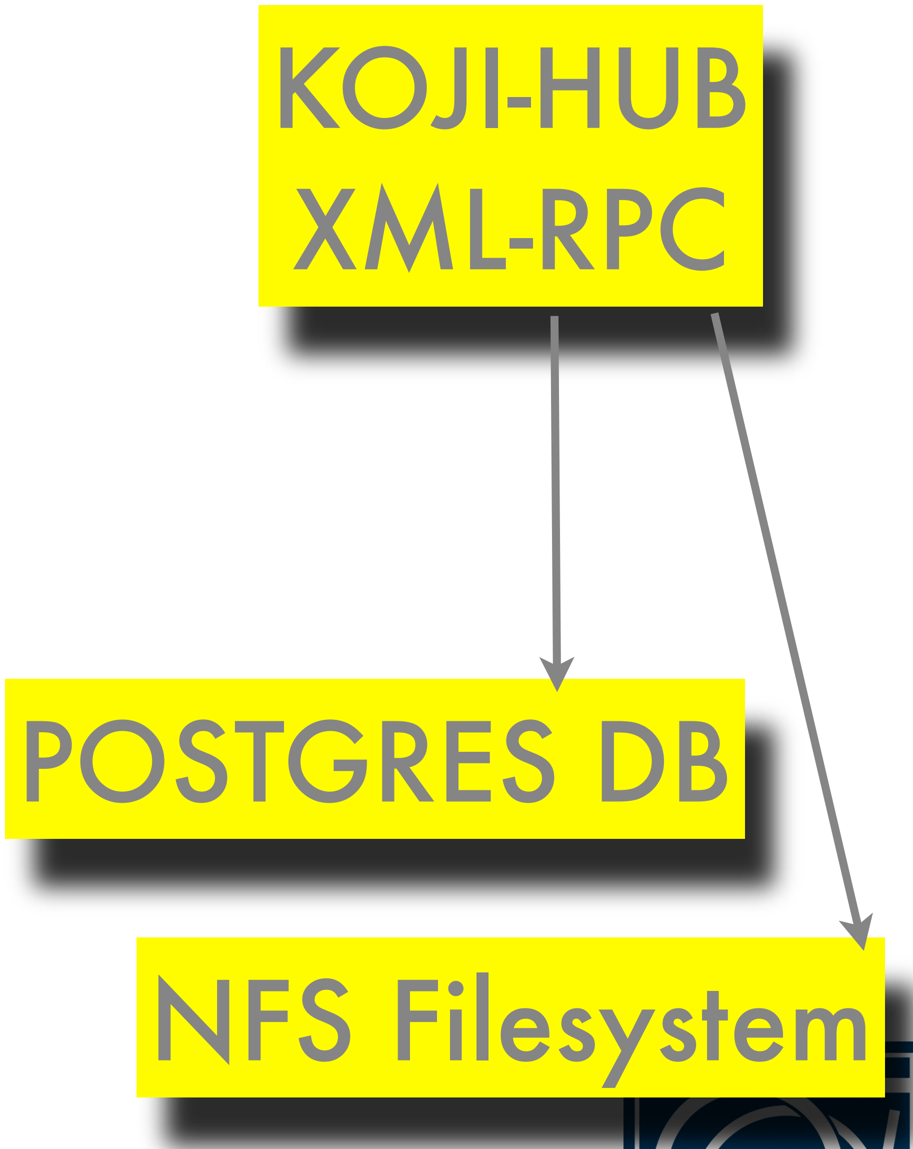
Metric	Value
Number of Users	994
Number of Build Hosts	20 ish
Architectures	i386, x86_64, ppc, ppc64 and s390
SRPM Names “N”*	8300, e.g openssh
Unique Builds. “NVR"*	101563, e.g openssh-3.2.fc9
Build Success/Failed/Cancelled*	73927/6525/271
Targets	F9,10,11,12 and EL4, 5.

*Statistics Start April 2007

KOJI Architecture.

KOJI Architecture.

Service

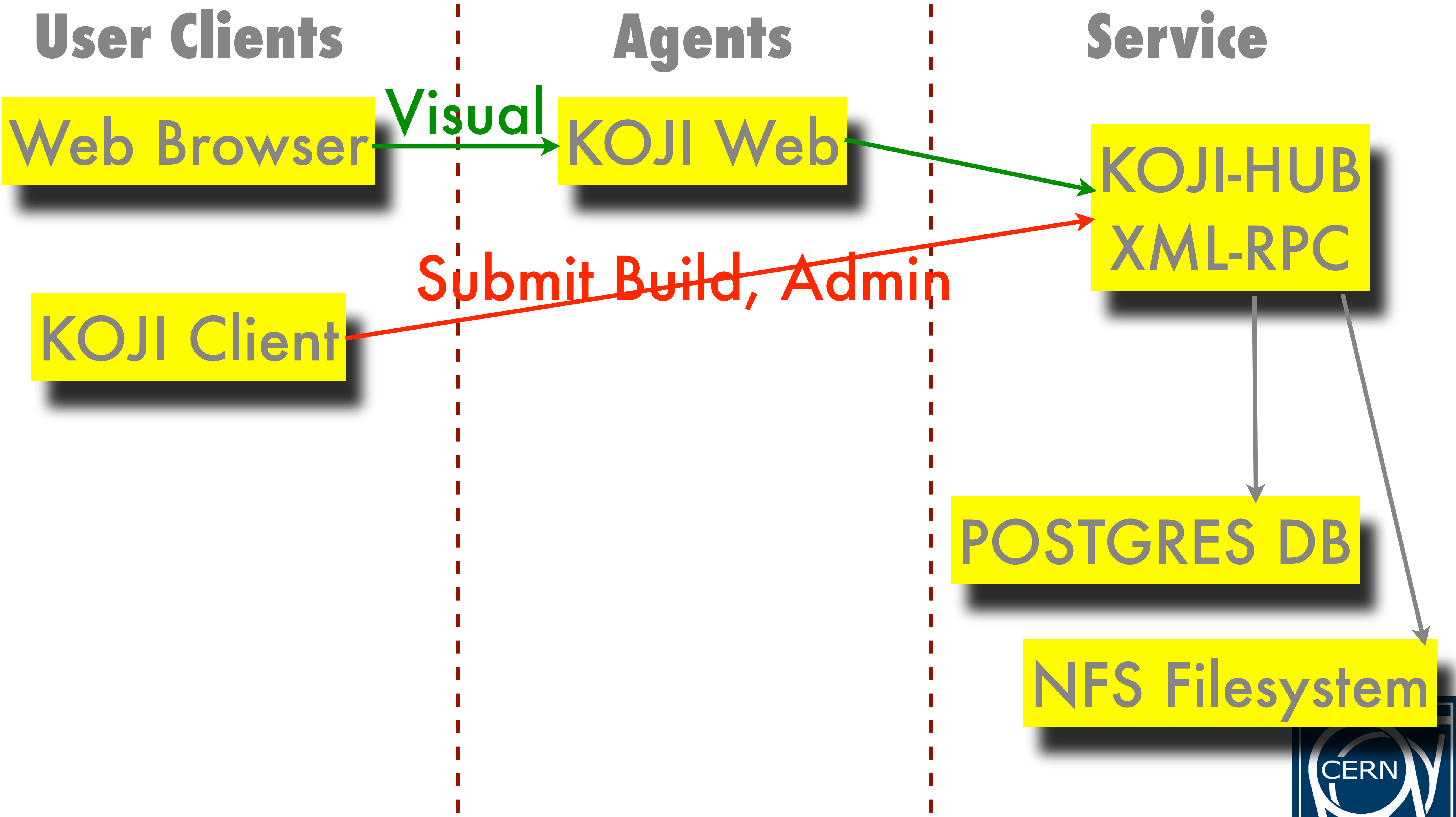


KOJI-HUB XML-RPC

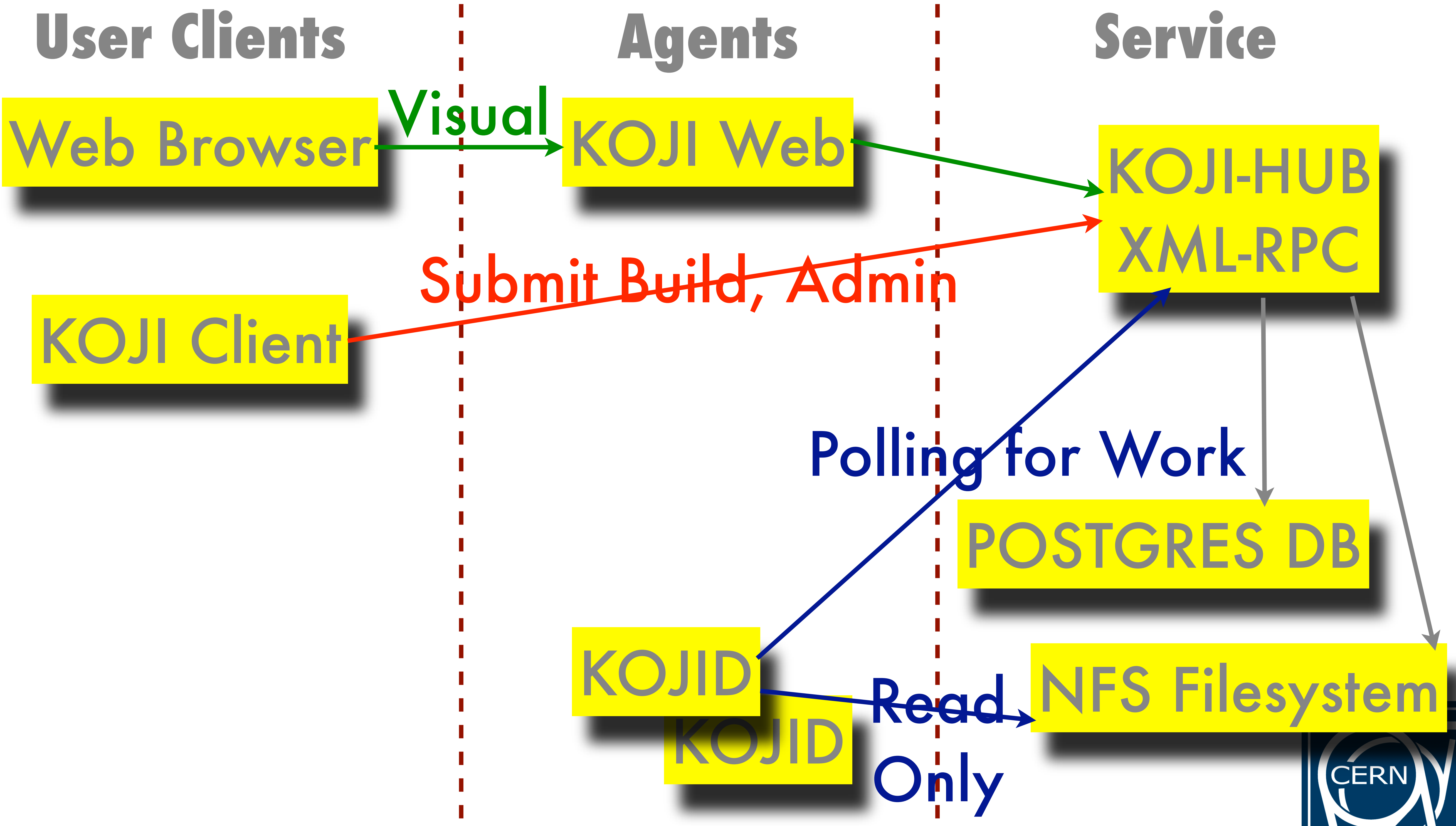
POSTGRES DB

NFS Filesystem

KOJI Architecture.



KOJI Architecture.



User Clients

Agents

Service

Web Browser

Visual

KOJI Web

KOJI-HUB
XML-RPC

Submit Build, Admin

KOJI Client

Polling for Work

POSTGRES DB

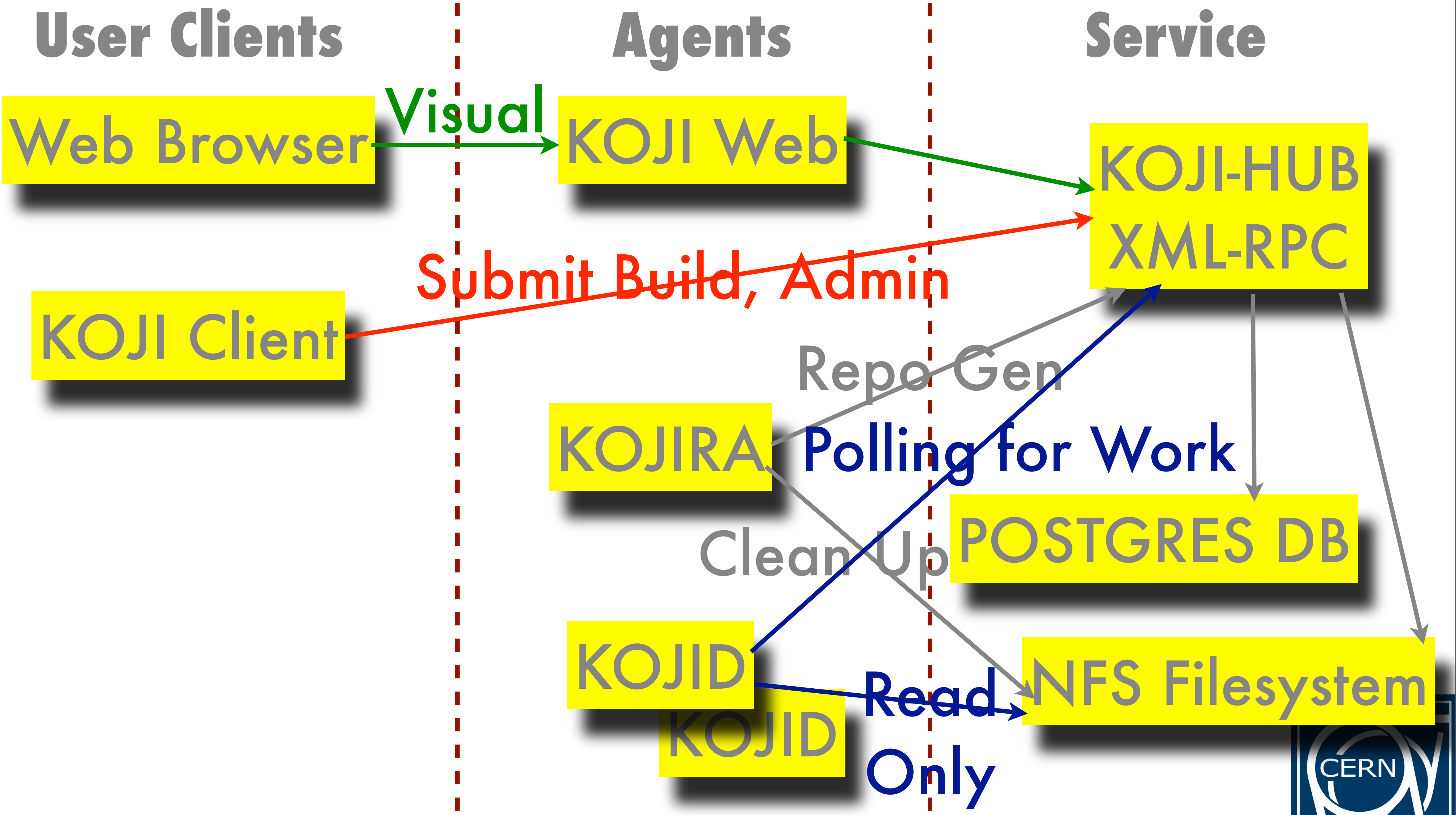
KOJID

KOJID

Read Only

NFS Filesystem

KOJI Architecture.



KOJI Components

- KOJI-HUB - PYTHON xml-rpc
  - Interacts with database , the brain.
- KOJI Client - very simple client.
  - Adding users, submitting builds.
- KOJI Web - a client to koji-hub.
  - Visualization, Some admin, Some build resubmission.
- KOJID - Ask koji-hub for work.
  - e.g Building an RPM or rebuilding a yum repo
- KOJIRA - Clean up tasks.
  - Destroy redundant buildroots and repositories.
  - Ask koji-hub to rebuild new repositories as needed.

KOJI Tags

- A koji tag contains a list of packages.
  - Typically dist-centos4 or dist-centos4-test.
- Each tag may contain:
  - A list of SRPM names owned by a user. eg:
    - openssh is owned by *straylen*
    - gzip is owned by *mrfoo*
  - A list of builds - “NVR” SRPM and built RPM
    - ssh-5.src and ssh-5.i386, ssh-askpass-5.i386
  - A list of external YUM Repositories.
    - e.g the official centos4 and centos4-updates
- koji (kojira) maintains 1 yum repository / tag.
  - Uses mergerepo to merge YUM repos.

KOJI Build Targets

- Builds are submitted to a build target.
  - A build target specifies buildroot and result tags.

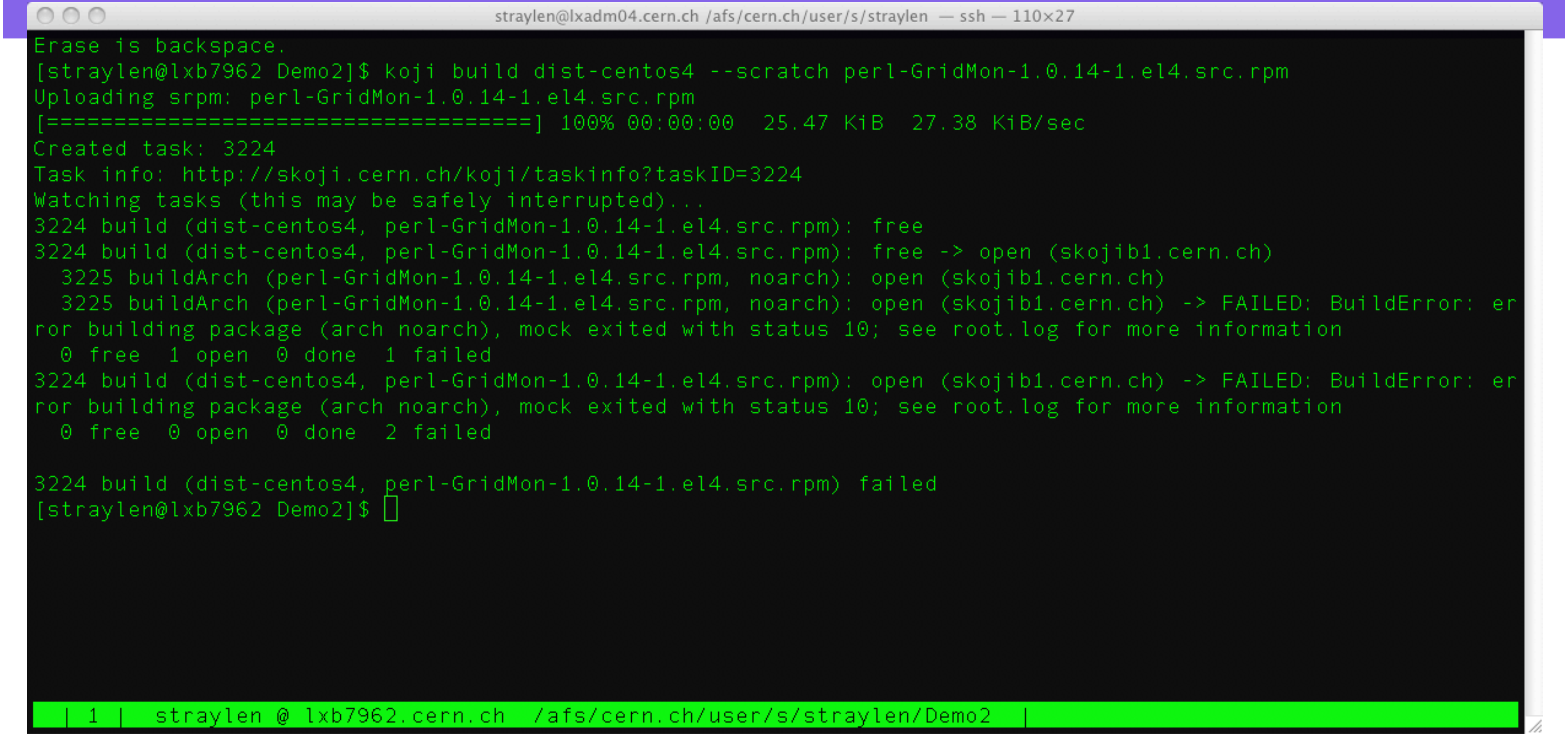
Target	BuildRoot	Destination
dist-centos5	dist-centos5-build	dist-centos5
dist-centos4	dist-centos4-build	dist-centos4-test

- A buildroot is a tag with a list of packages
  - It forms the base system for compiles.
- Successful builds are are added to dest' tag.
  - Results are available for subsequent builds
- Build results can be moved between tags.
  - e.g -test to -release.

Building from SRPM

- KOJI can build from a SRPM using koji client.
  - koji build [--scratch] dist-centos4 ssh-5.src.rpm
- KOJI (or rpm rather) sets the %dist tag
- Architectures are specified per tag.
- KOJI will try and build all archs - all must work.
  - ssh-clients-5.el4.i386 and ssh-clients-5.el4.x86_64
- KOJI will tag results as part of dist-centos4
    - Assuming not a “--scratch” build
  - ssh-clients-5.el4.i386, ssh-clients-5.el4.x86_64 ssh-5.el4.src.rpm
- Once a build is tagged it cannot be rebuilt (non-scratch) ever again.
  - Unique key = N-V-R.src.rpm + successful build.
- All subsequent builds can use ssh-clients as a BuildReq

DEMO2 - KOJI - SRPM



straylen@lxadm04.cern.ch /afs/cern.ch/user/s/straylen — ssh — 110×27

```
Erase is backspace.
[straylen@lxb7962 Demo2]$ koji build dist-centos4 --scratch perl-GridMon-1.0.14-1.el4.src.rpm
Uploading srpm: perl-GridMon-1.0.14-1.el4.src.rpm
[====================================] 100% 00:00:00  25.47 KiB  27.38 KiB/sec
Created task: 3224
Task info: http://skoji.cern.ch/koji/taskinfo?taskID=3224
Watching tasks (this may be safely interrupted)...
3224 build (dist-centos4, perl-GridMon-1.0.14-1.el4.src.rpm): free
3224 build (dist-centos4, perl-GridMon-1.0.14-1.el4.src.rpm): free -> open (skojib1.cern.ch)
  3225 buildArch (perl-GridMon-1.0.14-1.el4.src.rpm, noarch): open (skojib1.cern.ch)
  3225 buildArch (perl-GridMon-1.0.14-1.el4.src.rpm, noarch): open (skojib1.cern.ch) -> FAILED: BuildError: er
ror building package (arch noarch), mock exited with status 10; see root.log for more information
  0 free  1 open  0 done  1 failed
3224 build (dist-centos4, perl-GridMon-1.0.14-1.el4.src.rpm): open (skojib1.cern.ch) -> FAILED: BuildError: er
ror building package (arch noarch), mock exited with status 10; see root.log for more information
  0 free  0 open  0 done  2 failed

3224 build (dist-centos4, perl-GridMon-1.0.14-1.el4.src.rpm) failed
[straylen@lxb7962 Demo2]$ 
```

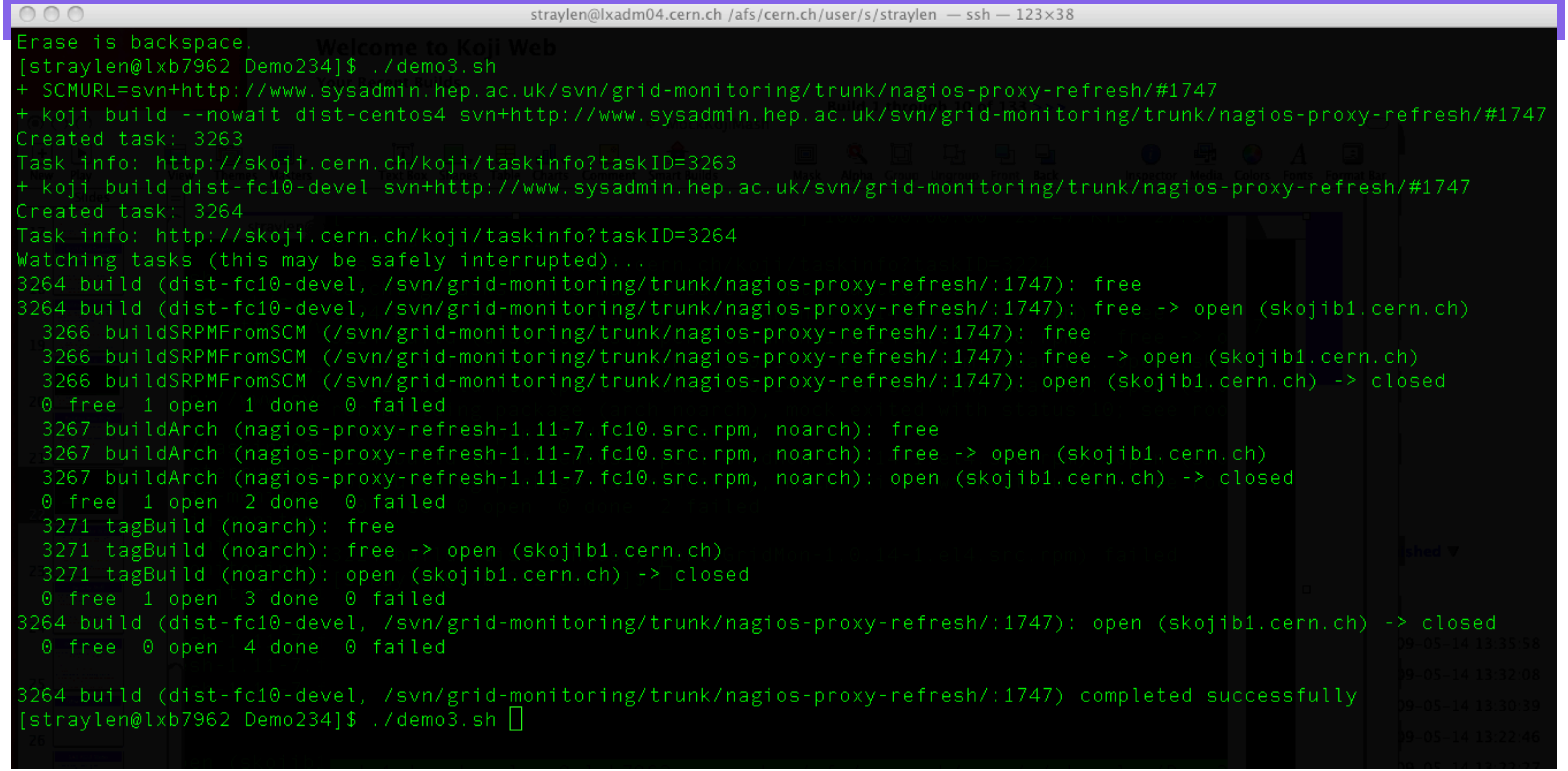
| 1 | straylen @ lxb7962.cern.ch /afs/cern.ch/user/s/straylen/Demo2 |

- A task id is returned.
    - Real time logs on the web or CLI.
    - Results and logs available via web, CLI or koji API

Building from SCM

- KOJI client accepts an SCM URL
  - cvs://host.example.org/cvsroot?software/ran#cvstag
  - svn+http://host.example.org/svn/soft#1234
  - git://git.example.org/26f47643
- koji build [--scratch] dist-centos4 <SCMURL>
  - SCM contents is checked out.
  - SPEC file located via a glob.
    - SPEC=*.spec
  - make sources
    - Result should be that SRC files present at '.'
  - rpmbuild -bs $SPEC
    - SRPM created.
  - SRPM build to RPMS continue as before.

DEMO 3 - KOJI SCM


straylen@lxadm04.cern.ch /afs/cern.ch/user/s/straylen — ssh — 123×38

```
Erase is backspace.
[straylen@lxb7962 Demo234]$ ./demo3.sh
+ SCMURL=svn+http://www.sysadmin.hep.ac.uk/svn/grid-monitoring/trunk/nagios-proxy-refresh/#1747
+ koji build --nowait dist-centos4 svn+http://www.sysadmin.hep.ac.uk/svn/grid-monitoring/trunk/nagios-proxy-refresh/#1747
Created task: 3263
Task info: http://skoji.cern.ch/koji/taskinfo?taskID=3263
+ koji build dist-fc10-devel svn+http://www.sysadmin.hep.ac.uk/svn/grid-monitoring/trunk/nagios-proxy-refresh/#1747
Created task: 3264
Task info: http://skoji.cern.ch/koji/taskinfo?taskID=3264
Watching tasks (this may be safely interrupted)...
3264 build (dist-fc10-devel, /svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): free
3264 build (dist-fc10-devel, /svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): free -> open (skojib1.cern.ch)
  3266 buildSRPMFromSCM (/svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): free
  3266 buildSRPMFromSCM (/svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): free -> open (skojib1.cern.ch)
  3266 buildSRPMFromSCM (/svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): open (skojib1.cern.ch) -> closed
  0 free  1 open  1 done  0 failed
  3267 buildArch (nagios-proxy-refresh-1.11-7.fc10.src.rpm, noarch): free
  3267 buildArch (nagios-proxy-refresh-1.11-7.fc10.src.rpm, noarch): free -> open (skojib1.cern.ch)
  3267 buildArch (nagios-proxy-refresh-1.11-7.fc10.src.rpm, noarch): open (skojib1.cern.ch) -> closed
  0 free  1 open  2 done  0 failed
  3271 tagBuild (noarch): free
  3271 tagBuild (noarch): free -> open (skojib1.cern.ch)
  3271 tagBuild (noarch): open (skojib1.cern.ch) -> closed
  0 free  1 open  3 done  0 failed
3264 build (dist-fc10-devel, /svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747): open (skojib1.cern.ch) -> closed
  0 free  0 open  4 done  0 failed

3264 build (dist-fc10-devel, /svn/grid-monitoring/trunk/nagios-proxy-refresh/:1747) completed successfully
[straylen@lxb7962 Demo234]$ ./demo3.sh
```

- 2 Build Jobs to dist-centos4 and dist-fc10-devel
  - 1st with “--nowait”. i.e async submission.

Strategy for SCM

- “RPM=*.spec ; make sources ; rpm -bs $RPM”
  - this must work but it does not matter how.
- EGEE SA1 Strategy.
  - SVN contains actual code.
  - make sources is really a “make dist”.
    - i.e it creates .tar.gz, and in some cases a .spec also.
- Fedora Strategy.
  - SVN contains .spec file only almost.
  - “make sources” downloads .tar.gz release files.
    - Calls curl to local .tar.gz repository
- Fedora method has advantages.
  - RPM release decoupled from software release.
    - Makes the branch point for .debs obvious.
  - To be considered for SA1 - in reality can live with both.

Security Stuff

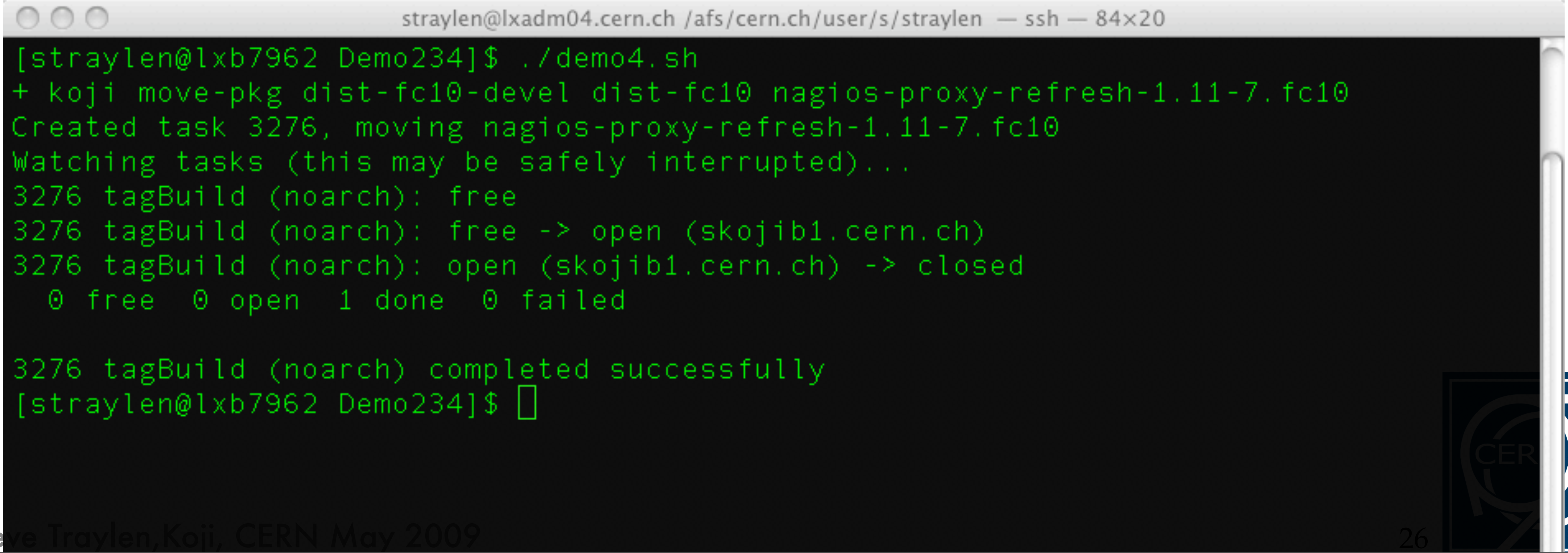
- All interfaces and clients support
  - kerberos, passwd or X509
- My testing all X509 based - CERN CA certs.
  - Koji mixes A & A. i.e it assumes single CA.
    - Usernames are taken from “CN=straylen”
    - emails are sent to straylen@cern.ch
    - Requires unencrypted keys on client in reality!!!
      - Solution proxies, gnome-keyring, seahorse, ..
- Users can be given permissions. e.g
  - admin , build, ...
  - policies can be set for given tasks..
    - e.g - build_from_srpm, has_perm build :: allow
    - policies not well documented, the rest is.

MASH - Koji 2 YUM

- MASH - simple koji client - (koji has an API)
  - Input a KOJI tag outputs to YUM Repository
    - i386/*.rpm
    - x86_64/*.rpm
    - SRPM/*.src.rpm
  - A repoview can be generated.
    - i.e pretty HTML and RSS.
- YUM Repository can be managed with KOJI.
  - An ssh BUILD "NVR" may be tagged on dist-centos4-testing.
  - $ koji tag-pkg dist-centos4-release ssh-5
    - Build now tagged on -release so appears there to.

DEMO 4 - MASH

- $ mash --output /pub/sa1 dist-fc10-devel
- $ mash --output /pub/sa1 dist-fc10-testing
  - Builds can be moved between tags.
  - e.g A devel , testing and release corresponding to devel , testing and release YUM repositories.


straylen@lxadm04.cern.ch /afs/cern.ch/user/s/straylen — ssh — 84×20

```
[straylen@lxb7962 Demo234]$ ./demo4.sh
+ koji move-pkg dist-fc10-devel dist-fc10 nagios-proxy-refresh-1.11-7.fc10
Created task 3276, moving nagios-proxy-refresh-1.11-7.fc10
Watching tasks (this may be safely interrupted)...
3276 tagBuild (noarch): free
3276 tagBuild (noarch): free -> open (skojib1.cern.ch)
3276 tagBuild (noarch): open (skojib1.cern.ch) -> closed
  0 free  0 open  1 done  0 failed

3276 tagBuild (noarch) completed successfully
[straylen@lxb7962 Demo234]$ 
```

SA1 KOJI Status

- Take a look: http://skoji.cern.ch/koji
- Early March - Mock used for all new packages.
  - About 40 package names. - James and I had MOCK collisions.
  - About 250 SL4 and 60 SL5 packages.
- Early April - Koji Used for all packages.
  - Contains 60 packages currently
    - SA1 packages.
    - Redhat MRT Packages (e.g condor, qpid,..)
    - gstat packages.
- Next Steps.
  - Write a web page how to use it. - Reference central docs.
  - Improve upload to Manchester repos, i.e MASH.
  - Worry about the poor quality service -
    - Classical “under desk server” I’m ashamed.
    - Does it matter? Less than you might think.

SA1 KOJI - Stats

Metric	Value
Number of Users	8 (3 active)
Number of Build Hosts	1 for everything
Architectures	i386, x86_64
SRPM Names “N”*	60, e.g gocdb-downtime
Unique Builds. “NVR"*	150, e.g gocdb-downtime-0.1.el4
Build Success/Failed/Cancelled*	135/10/4
Targets	EL4, 5 and FC10.

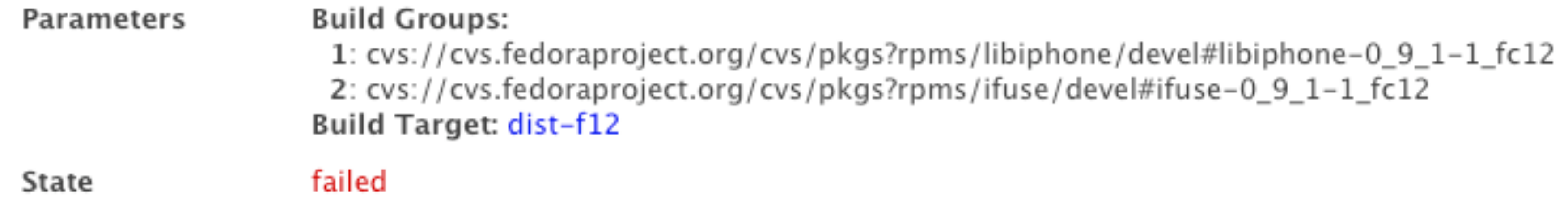
- *Logs from 27th March 2009
- MOCK only built packages, about 60
  - These existing pkgs will be imported to koji.

The Obvious Questions

- Bulk submission.
  - A *wait-repo* job - sleeps on X being in the repository.
  - A *chain-build* - allows multiple SCMURLs to be submitted, *wait-repo* jobs in between.
- Moving from EL-N to EL-N+1
  - Tags can be created as copies or can inherit details of other tags.
- Notifications
  - Successful builds go to submitter and pkg owner.
  - You can subscribe to events as you want.

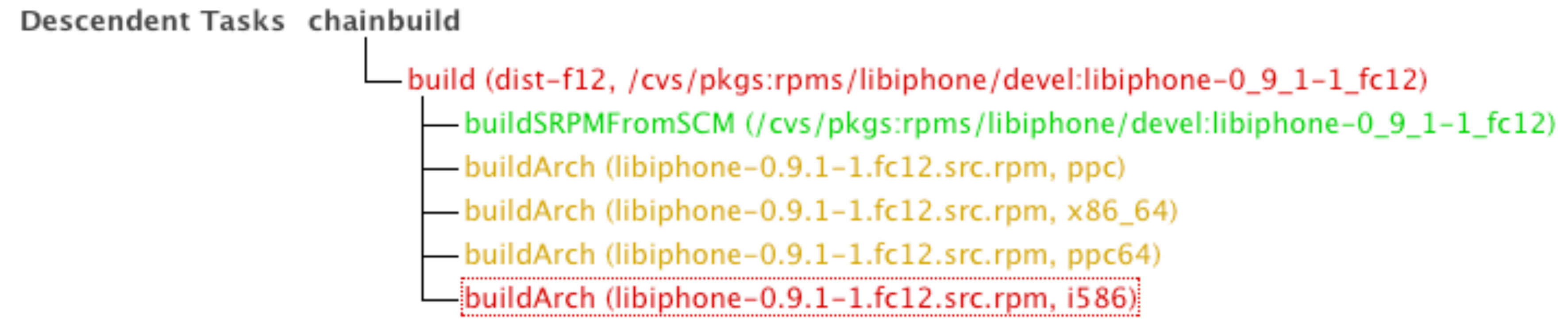
Chain Build Examples

libiphone and ifuse



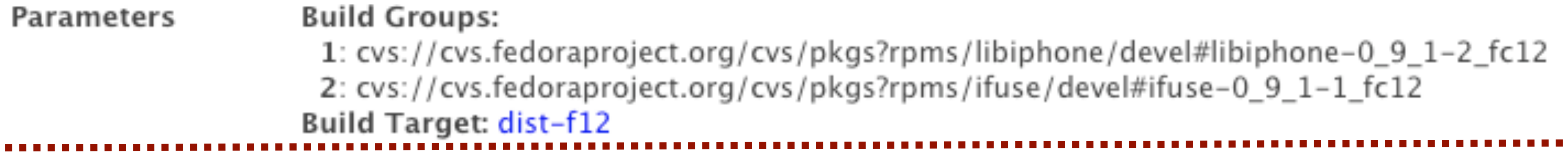
```
Parameters          Build Groups:
                      1: cvs://cvs.fedoraproject.org/cvs/pkgs?rpms/libiphone/devel#libiphone-0_9_1-1_fc12
                      2: cvs://cvs.fedoraproject.org/cvs/pkgs?rpms/ifuse/devel#ifuse-0_9_1-1_fc12
                    Build Target: dist-f12

State               failed
```

libiphone i586 fails
Other archs cancelled
ifuse not started.



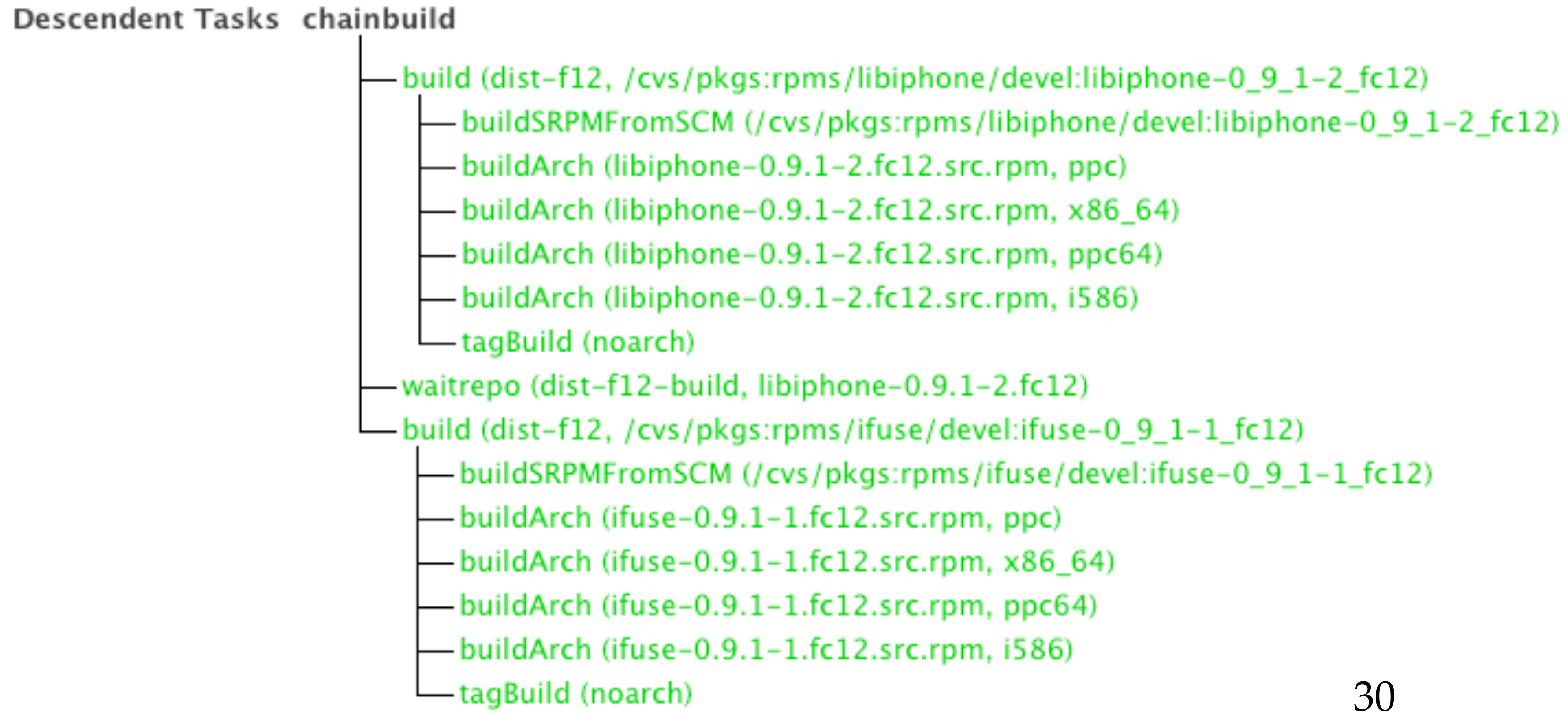
```
Descendent Tasks  chainbuild
                    └build (dist-f12, /cvs/pkgs:rpms/libiphone/devel:libiphone-0_9_1-1_fc12)
                       ├buildSRPMFromSCM (/cvs/pkgs:rpms/libiphone/devel:libiphone-0_9_1-1_fc12)
                       ├buildArch (libiphone-0.9.1-1.fc12.src.rpm, ppc)
                       ├buildArch (libiphone-0.9.1-1.fc12.src.rpm, x86_64)
                       ├buildArch (libiphone-0.9.1-1.fc12.src.rpm, ppc64)
                       └buildArch (libiphone-0.9.1-1.fc12.src.rpm, i586)
```

swig now added as buildreq



```
Parameters          Build Groups:
                      1: cvs://cvs.fedoraproject.org/cvs/pkgs?rpms/libiphone/devel#libiphone-0_9_1-2_fc12
                      2: cvs://cvs.fedoraproject.org/cvs/pkgs?rpms/ifuse/devel#ifuse-0_9_1-1_fc12
                    Build Target: dist-f12
```

- libiphone built
  - all archs built, tagged
  - Added to repo
- wait for addition
  - ifuse built.
  - tagged
- Success
- Non-Scratch only!



```
Descendent Tasks  chainbuild
                    ├build (dist-f12, /cvs/pkgs:rpms/libiphone/devel:libiphone-0_9_1-2_fc12)
                    │  ├buildSRPMFromSCM (/cvs/pkgs:rpms/libiphone/devel:libiphone-0_9_1-2_fc12)
                    │  ├buildArch (libiphone-0.9.1-2.fc12.src.rpm, ppc)
                    │  ├buildArch (libiphone-0.9.1-2.fc12.src.rpm, x86_64)
                    │  ├buildArch (libiphone-0.9.1-2.fc12.src.rpm, ppc64)
                    │  ├buildArch (libiphone-0.9.1-2.fc12.src.rpm, i586)
                    │  └tagBuild (noarch)
                    ├waitrepo (dist-f12-build, libiphone-0.9.1-2.fc12)
                    └build (dist-f12, /cvs/pkgs:rpms/ifuse/devel:ifuse-0_9_1-1_fc12)
                       ├buildSRPMFromSCM (/cvs/pkgs:rpms/ifuse/devel:ifuse-0_9_1-1_fc12)
                       ├buildArch (ifuse-0.9.1-1.fc12.src.rpm, ppc)
                       ├buildArch (ifuse-0.9.1-1.fc12.src.rpm, x86_64)
                       ├buildArch (ifuse-0.9.1-1.fc12.src.rpm, ppc64)
                       ├buildArch (ifuse-0.9.1-1.fc12.src.rpm, i586)
                       └tagBuild (noarch)
```

More Questions

- What about debian? Not mock's or koji's job.
  - Ask someone who knows about it, buildd?
    - Good argument for splitting source from SPEC.
- What about .so.N updates? koji does nothing.
  - Identify dependents via repoquery.
  - Bump RPM release and build.
    - If you miss one it will be obvious on v.first install or first subsequent build that requires something.
- Reproduce build machine for a developer.
  - $ koji mock-config dist-centos4
  - Provides mock config for mock
    - Developer can create a chroot with a shell in it.

Conclusions

- mock is one step above rpmbuild.
  - Creating lots of RPMS from SRPMS - a no brainer.
- koji a few steps above mock.
  - Extremely good at integrating software when
    - Lots of submitters/architectures or platforms.
    - Living in a world of RPMs,
      - Does not interfere with other items though.
  - Rubbish at everything else, e.g continuous, ..
- Support and thanks.
  - Mike Bonnet, Bill Nottingham & Seth Vidal.
  - Mail and IRC response great - an hour or two.
  - Same day bug fixes and an invitation to LinuxTag

References

- http://skoji.cern.ch/koji/
- http://koji.fedoraproject.org/koji
- http://fedoraproject.org/wiki/Koji
- http://fedoraproject.org/wiki/Koji/ServerHowTo
- http://fedoraproject.org/wiki/Projects/Mock
- http://people.redhat.com/notting/mash/
  - (all software standard in FC10 at least)