DPM collaboration

Fabrizio Furano - CERN IT-SDC

France Grilles Nov 2013

About this talk

- We introduce the DPM software lifecycle
  - Dev, build, release, test
- We show where activities by external partners can plug in the lifecycle
- We relate this to the DPM collaboration
- We propose a few items that would be interesting for the DPM community

DPM

- ~45PB, ~200 sites, ~10 with more than 1PB
- Evolution: modern codebase, new DMLITE sw stack, important plugins have been rewritten with performance in mind
- Apache or gridFTP or Xrootd or <whatever> can load the DMLITE stack, giving standardized and very flexible features
- DMLITE can talk to the historical daemons (dpm_) or go straight to the DB
- DPM keeps historical daemons for compatibility with the legacy dpm_ and dpns_ clients, also used by the SRM.
  - The historical dpns (and hence LFC) has been enhanced to make it perform better (up to 4X-5X)

- A fully plugin-based architecture calls for plugins
- The decision has been to emphasize the modularity, by releasing plugins individually packaged
- Very big leap with the evolution of the builds
- Full EPEL release happened in the first week of September
  - 47 SEs already upgraded after a few weeks!

Architectures

- We have 32 components
- We have 2 archs (i386 and x86_64)
- We have 2 platforms (el5, el6)
  - Minus el5-i386, which we don't support
- Hence, ±100 RPMs, as some components produce more than one

- This process started ±2 years ago, when we incrementally started using 'mock' to build
- The build management overhead of such a mass can be overkill for us humans, a detail out of place can trigger the need for a lot of manpower just to fix little configurations
- We prefer to concentrate on the quality than on the overhead, hence… we want VERY reliable builds, babysitting is out of question

Our continuous builds now

- Minimalistic approach
  - The devs develop
  - The build system builds (what the devs do)
  - The nightly tests … deploy and test (what's built)
  - The testbeds self update and run (what's blessed)
- We remove all the config bits that are redundant for this purpose
- Master rule: Never have to touch the build system after changes in the code.
  - The build system just … builds!

Repositories we provide

- Master rule: everything has always to compile and install correctly
- The repositories are OPEN, you can install them

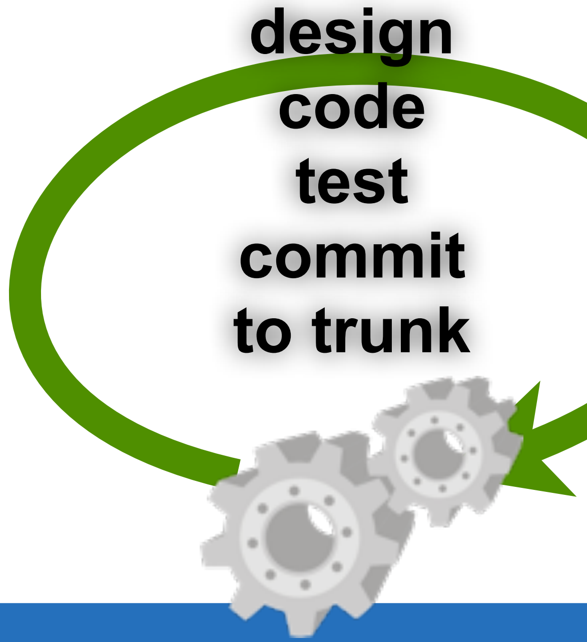
- Trunk-level: where we work and add features. Things may be slightly incomplete here.
- Release-candidate: the set of the component versions that have been blessed as "well working" by the developers
- When we are happy, we proceed with the EPEL workflow

Continuous *Trunk* repo

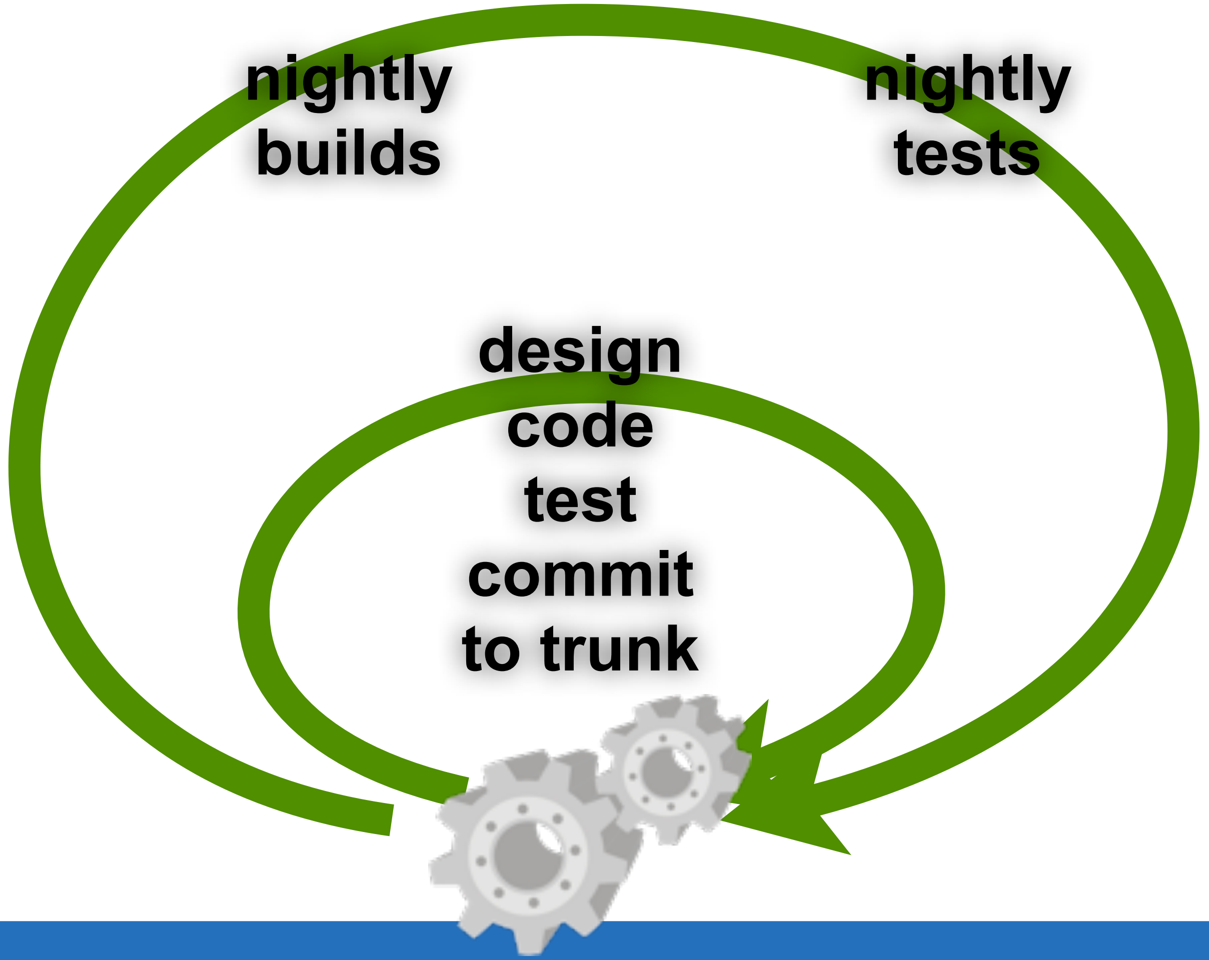
- We build in a continuous way the trunk of all the components using epel-testing as external package provider
- Populate 3 YUM repos (2 archs, 2 platforms minus el5_i386) with the rpm and srpm artifacts
- Make these repos available for
  - The developer, with a low latency (minutes)
  - The nightly deployment and functional tests
    - *yum* like a normal repository would
- Our trunk testbed selfupdates every day

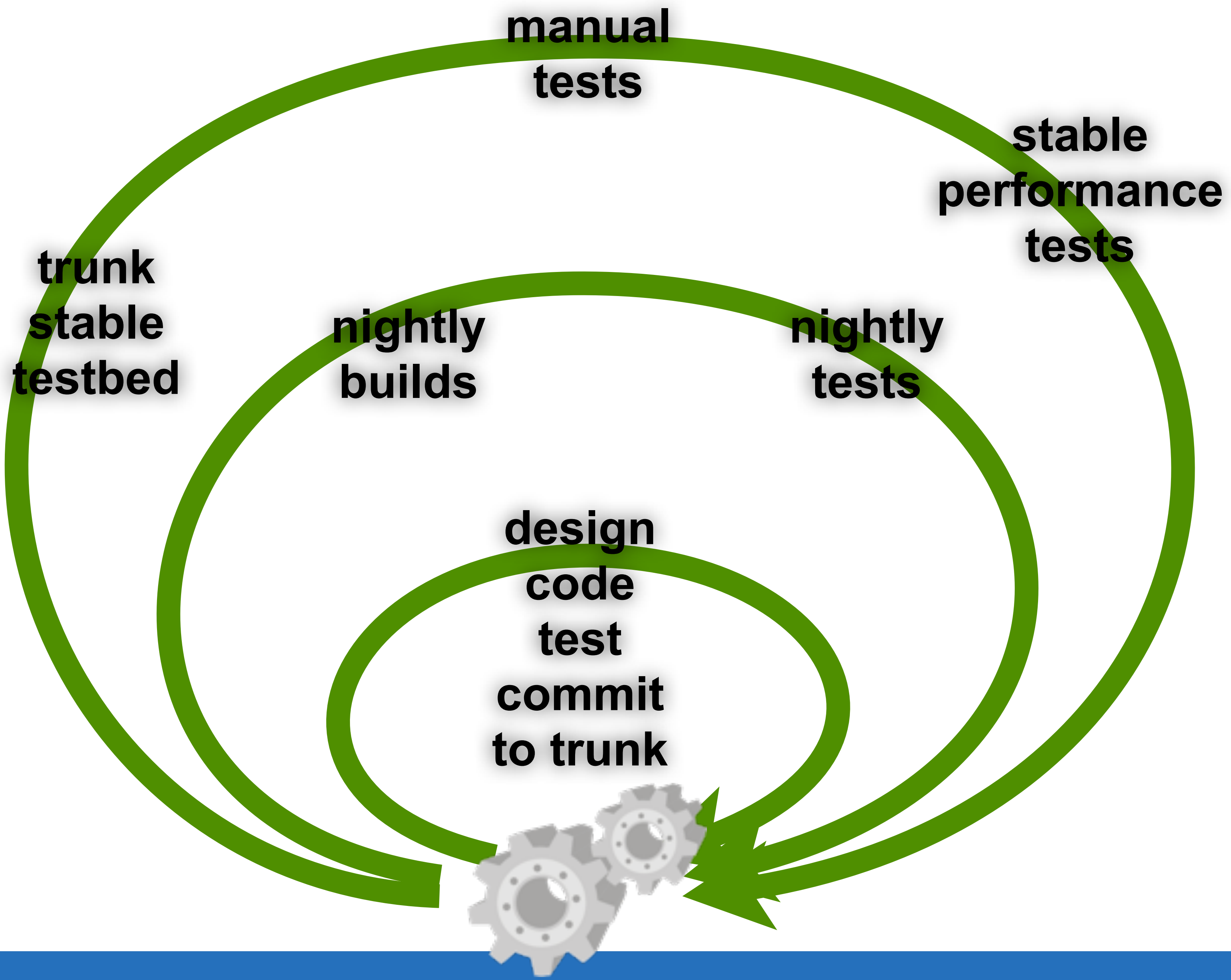
Continuous *Candidate Release* repo

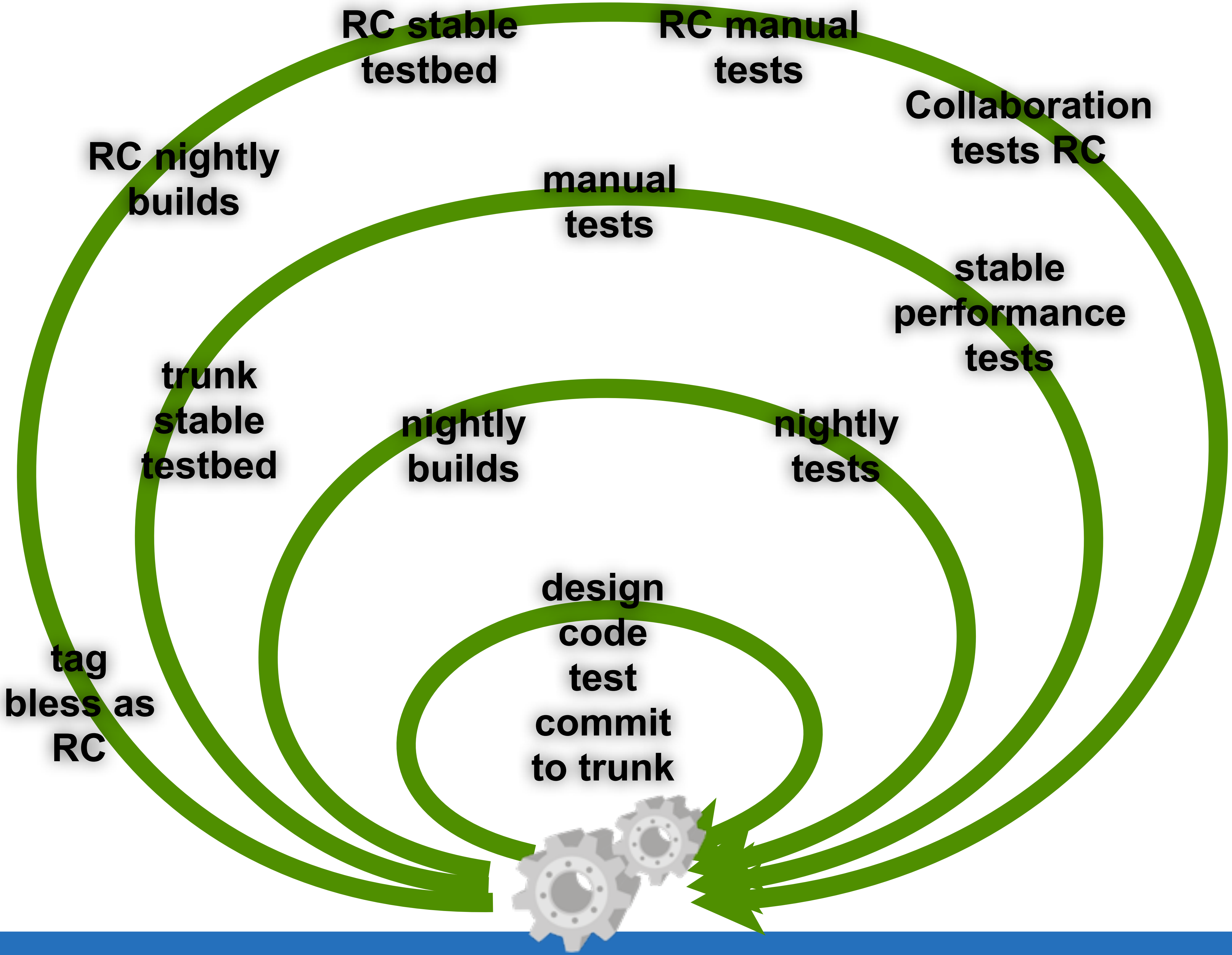
- Minimalistic approach
- The set of packages of the component versions that have been blessed by the devs
  - The responsible dev feels comfortable with users using his component
  - The responsible dev “marks” a tag as RC using an “svn:externals” link
    - A directory with a known name (candidate-release :-)) points to the blessed tag, like a symlink
  - The build system ... builds whatever is named candidate-release every time it changes
    - No parameters, hence no tweaks are needed, can only work.
- Our RC testbed selfupdates every day



design
code
test
commit
to trunk


nightly
builds
nightly
tests
design
code
test
commit
to trunk


manual
tests
stable
performance
tests
trunk
stable
testbed
nightly
builds
nightly
tests
design
code
test
commit
to trunk



design
code
test
commit
to trunk

nightly
builds

nightly
tests

trunk
stable
testbed

manual
tests

stable
performance
tests

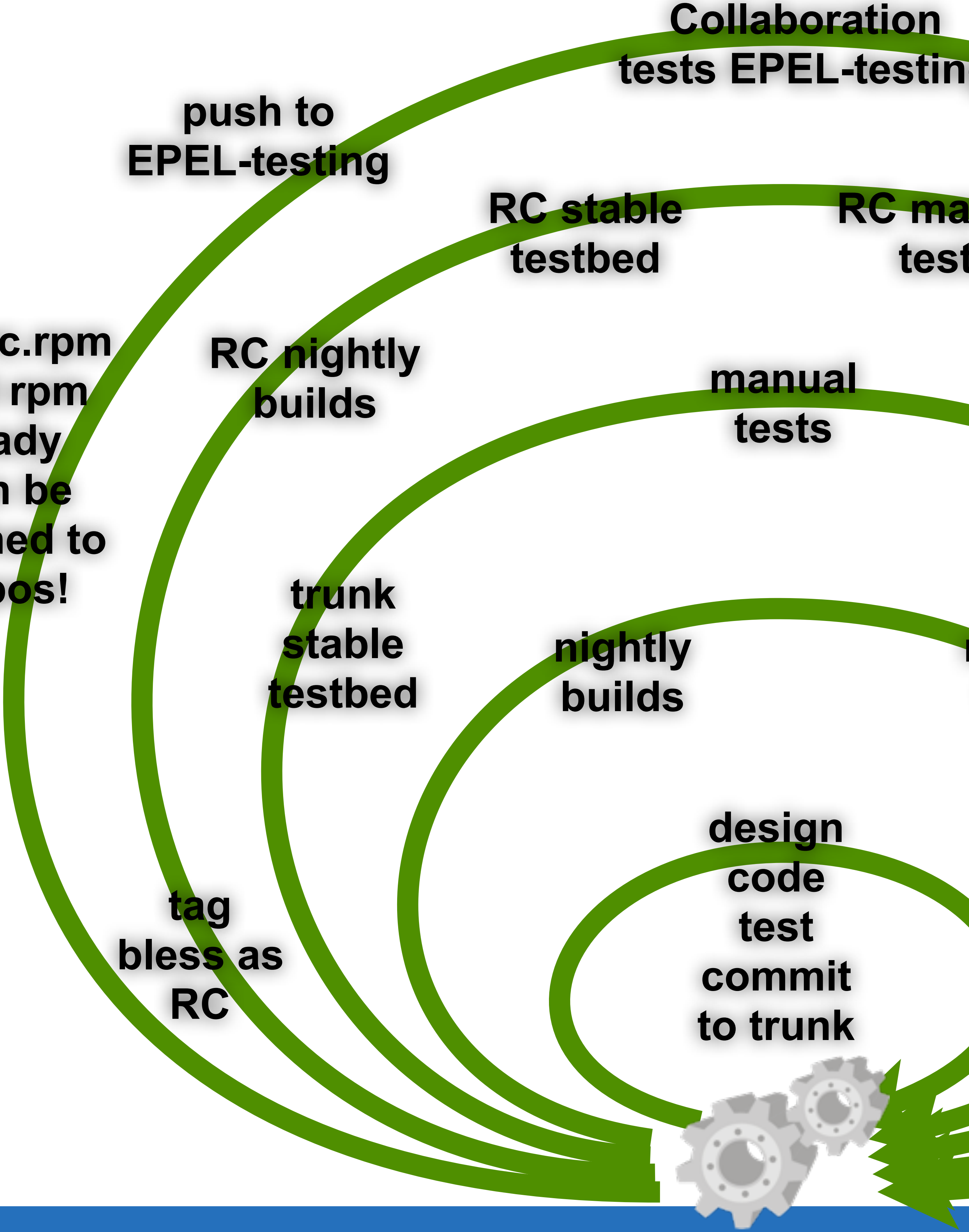
RC nightly
builds

RC stable
testbed

RC manual
tests

Collaboration
tests RC

tag
bless as
RC



design
code
test
commit
to trunk

nightly
builds

nightly
tests

trunk
stable
testbed

manual
tests

stable
performance
tests

tag
bless as
RC

RC nightly
builds

RC stable
testbed

RC manual
tests

Collaboration
tests RC

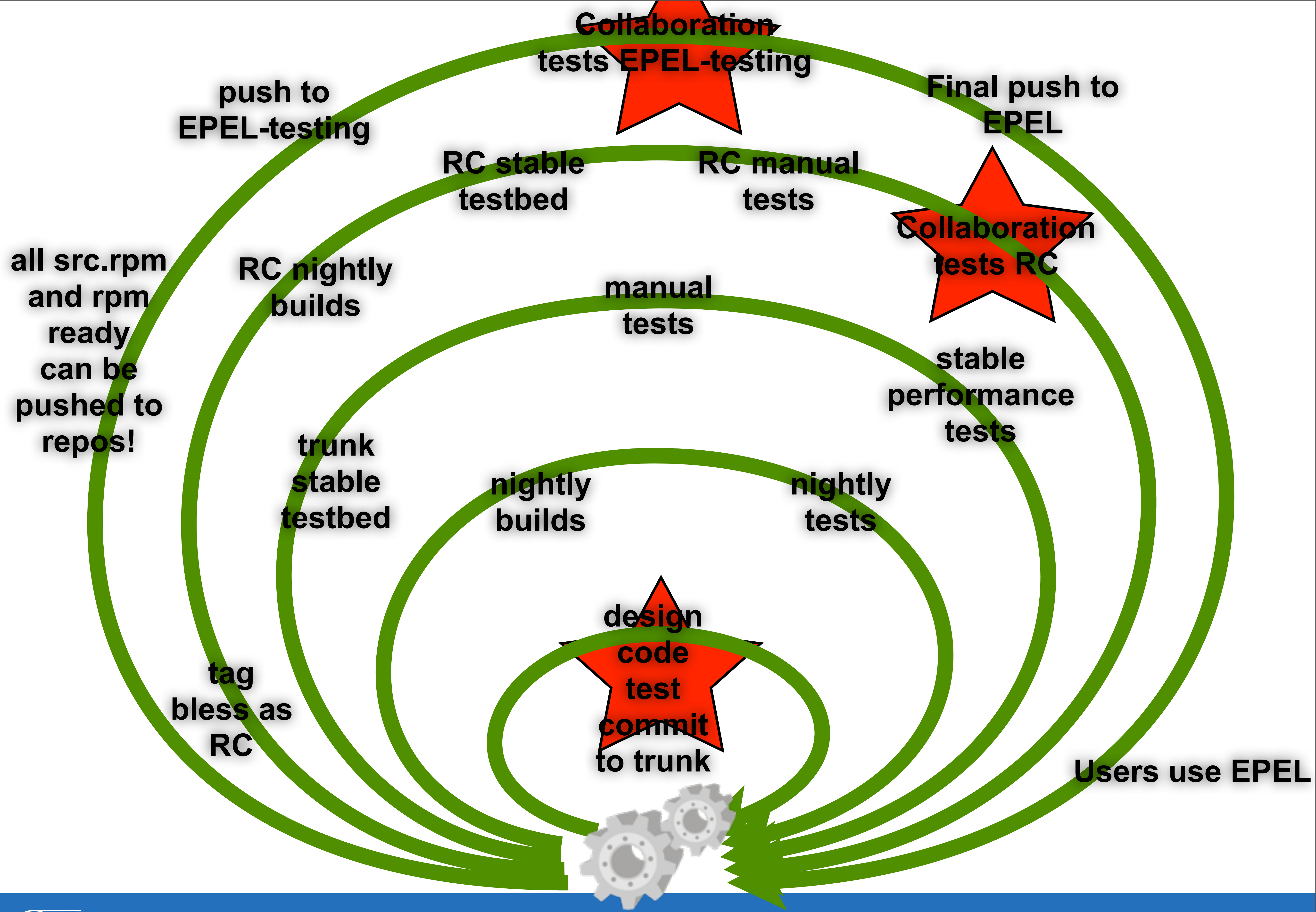
all src.rpm
and rpm
ready
can be
pushed to
repos!

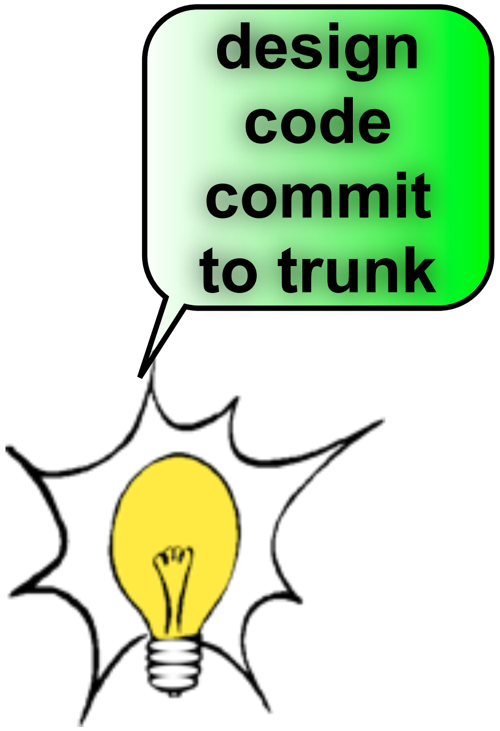
push to
EPEL-testing

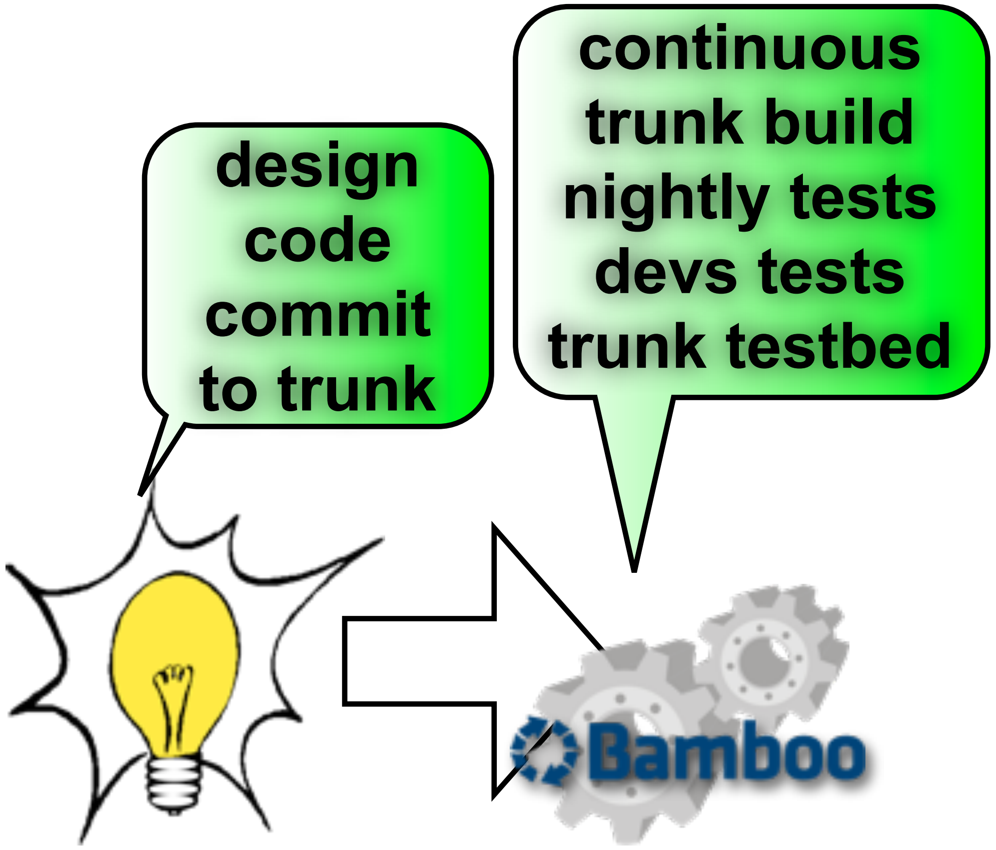
Collaboration
tests EPEL-testing

Final push to
EPEL

Users use EPEL


Collaboration
tests EPEL-testing
push to
EPEL-testing
Final push to
EPEL
RC stable
testbed
RC manual
tests
Collaboration
tests RC
all src.rpm
and rpm
ready
can be
pushed to
repos!
RC nightly
builds
manual
tests
stable
performance
tests
trunk
stable
testbed
nightly
builds
nightly
tests
design
code
test
commit
to trunk
tag
bless as
RC
Users use EPEL

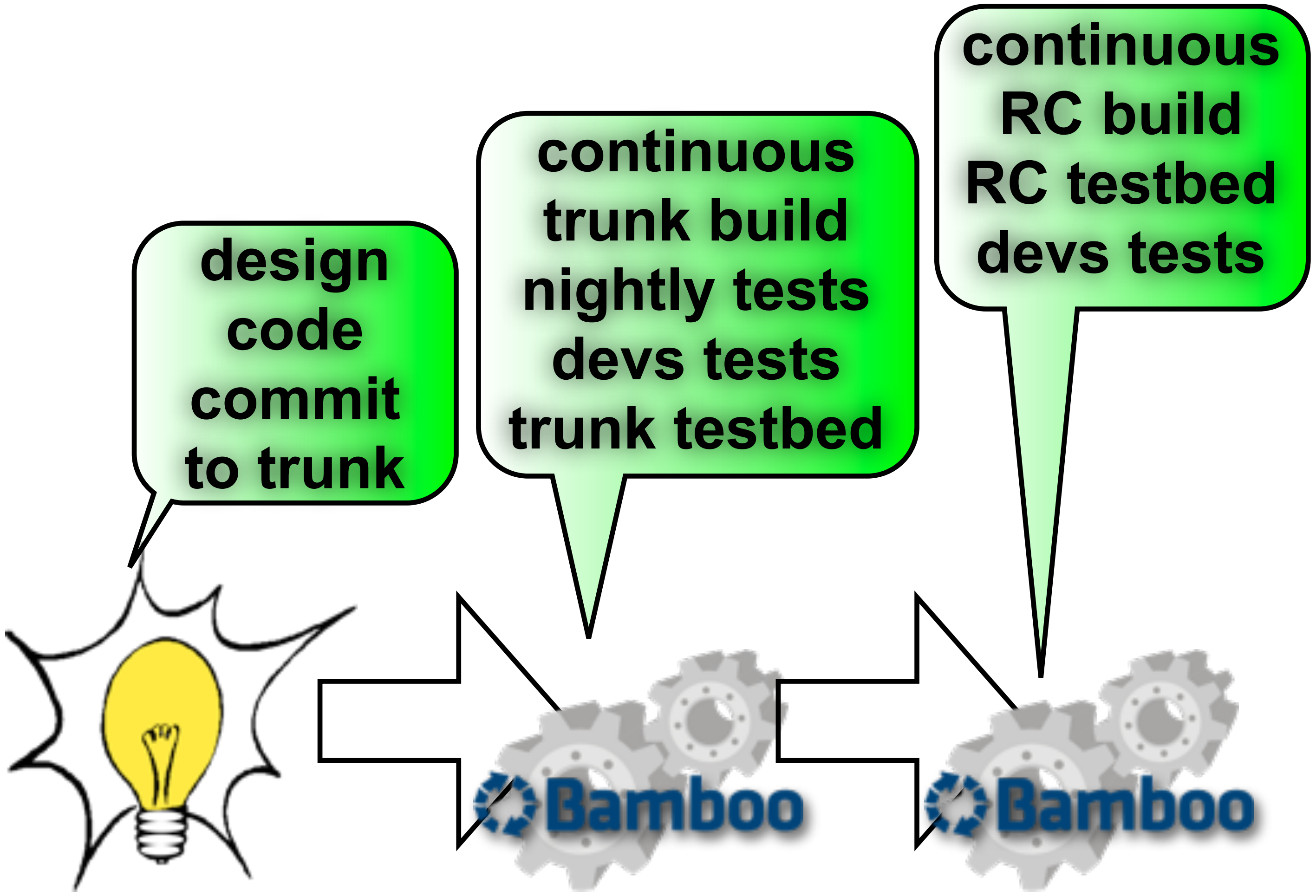

design
code
commit
to trunk


design code commit to trunk

continuous trunk build nightly tests devs tests trunk testbed

trunk
YUM repo

Trunk
stable testbed



design
code
commit
to trunk

continuous
trunk build
nightly tests
devs tests
trunk testbed

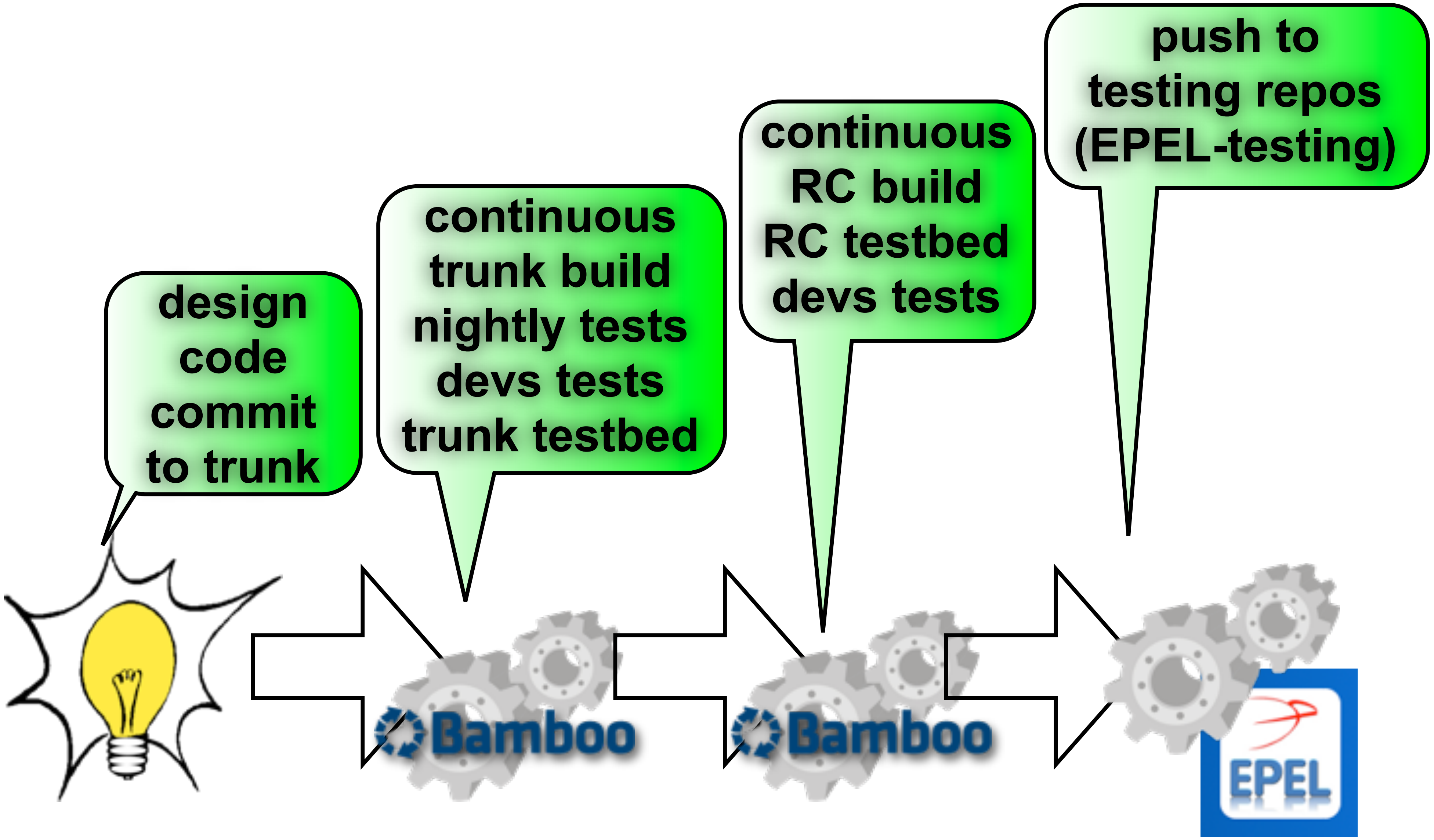
continuous
RC build
RC testbed
devs tests

trunk
YUM repo

RC
YUM repo

Trunk
stable testbed

RC
stable testbed
(open)



design
code
commit
to trunk

continuous
trunk build
nightly tests
devs tests
trunk testbed

continuous
RC build
RC testbed
devs tests

push to
testing repos
(EPEL-testing)

trunk
YUM repo

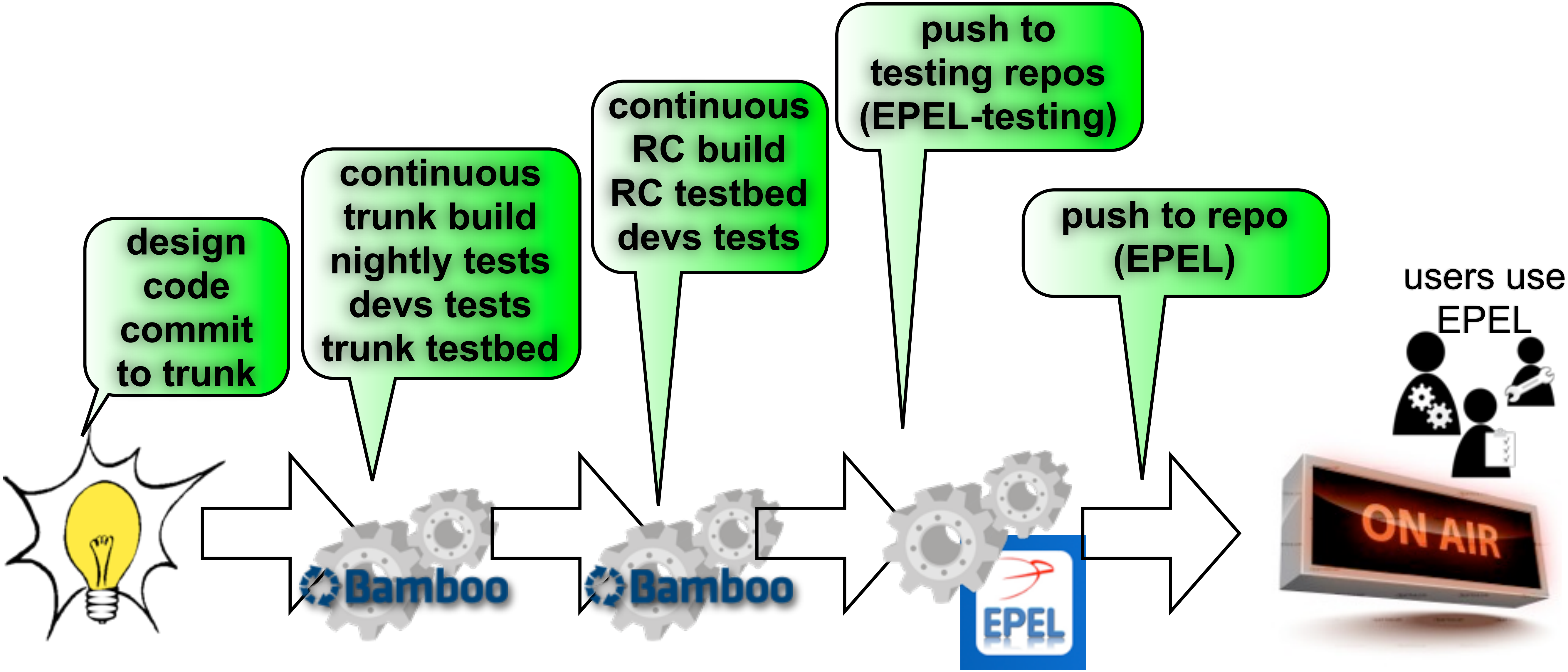
RC
YUM repo

EPEL-test
YUM repo

Trunk
stable testbed

RC
stable testbed
(open)

EPEL-test
stable testbeds
(collaboration)



design
code
commit
to trunk

continuous
trunk build
nightly tests
devs tests
trunk testbed

continuous
RC build
RC testbed
devs tests

push to
testing repos
(EPEL-testing)

push to repo
(EPEL)

users use
EPEL

trunk
YUM repo

RC
YUM repo

EPEL-test
YUM repo

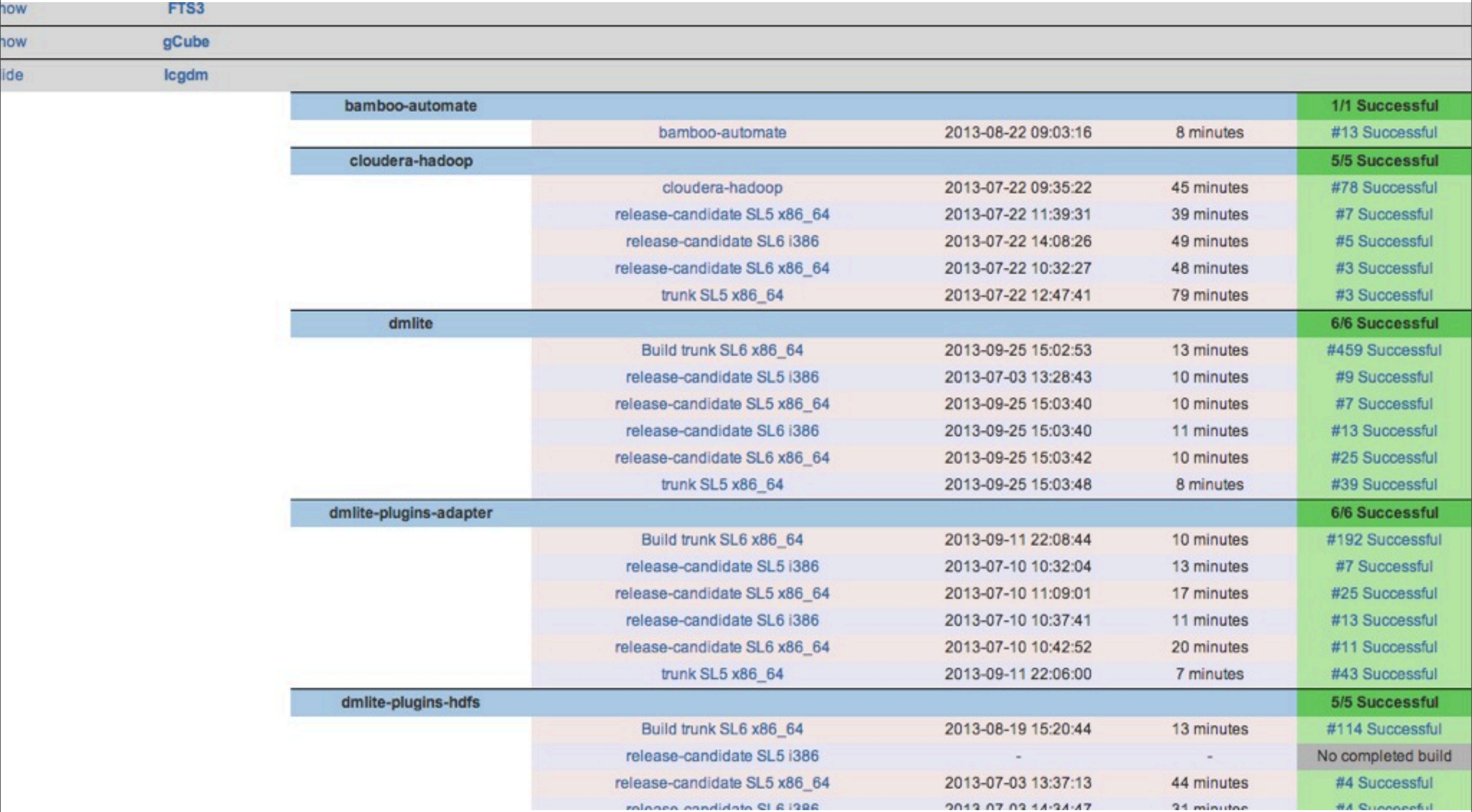
EPEL
YUM repo

Trunk
stable testbed

RC
stable testbed
(open)

EPEL-test
stable testbeds
(collaboration)

The new visualization



how	FTS3			
how	gCube			
ide	lcgdm			
	bamboo-automate			1/1 Successful
	bamboo-automate	2013-08-22 09:03:16	8 minutes	#13 Successful
	cloudera-hadoop			5/5 Successful
	cloudera-hadoop	2013-07-22 09:35:22	45 minutes	#78 Successful
	release-candidate SL5 x86_64	2013-07-22 11:39:31	39 minutes	#7 Successful
	release-candidate SL6 i386	2013-07-22 14:08:26	49 minutes	#5 Successful
	release-candidate SL6 x86_64	2013-07-22 10:32:27	48 minutes	#3 Successful
	trunk SL5 x86_64	2013-07-22 12:47:41	79 minutes	#3 Successful
	dmlite			6/6 Successful
	Build trunk SL6 x86_64	2013-09-25 15:02:53	13 minutes	#459 Successful
	release-candidate SL5 i386	2013-07-03 13:28:43	10 minutes	#9 Successful
	release-candidate SL5 x86_64	2013-09-25 15:03:40	10 minutes	#7 Successful
	release-candidate SL6 i386	2013-09-25 15:03:40	11 minutes	#13 Successful
	release-candidate SL6 x86_64	2013-09-25 15:03:42	10 minutes	#25 Successful
	trunk SL5 x86_64	2013-09-25 15:03:48	8 minutes	#39 Successful
	dmlite-plugins-adapter			6/6 Successful
	Build trunk SL6 x86_64	2013-09-11 22:08:44	10 minutes	#192 Successful
	release-candidate SL5 i386	2013-07-10 10:32:04	13 minutes	#7 Successful
	release-candidate SL5 x86_64	2013-07-10 11:09:01	17 minutes	#25 Successful
	release-candidate SL6 i386	2013-07-10 10:37:41	11 minutes	#13 Successful
	release-candidate SL6 x86_64	2013-07-10 10:42:52	20 minutes	#11 Successful
	trunk SL5 x86_64	2013-09-11 22:06:00	7 minutes	#43 Successful
	dmlite-plugins-hdfs			5/5 Successful
	Build trunk SL6 x86_64	2013-08-19 15:20:44	13 minutes	#114 Successful
	release-candidate SL5 i386	-	-	No completed build
	release-candidate SL5 x86_64	2013-07-03 13:37:13	44 minutes	#4 Successful
	release-candidate SL6 i386	2013-07-03 14:34:47	31 minutes	#4 Successful

1.8.7 is the EPEL DPM

- Big improvements to the puppet configuration
- DB Connection pooling in the DPNS/LFC daemon
  - Improves performance 3X-5X and reduces famous hiccups
- The EGI/OGF Storage Accounting Record producer (StAR)
- Lots of bug fixes
- Brand new Xrootd interface using native DMLite (good feedback on the prev one, from ALICE and ATLAS FAX)
- Brand new GridFTP2 with redirections is coming
- Dynamic HTTP Federations (implemented as a dmlite plugin called "UGR", only in RC by now)

YAIM deprecated

- We will maintain YAIM until August 2014
- Its successor is a Puppet-based setup, totally mainstream tools
- We encourage the members of the collaboration to learn the new setup technology and give feedback
  - Now it's the right moment

The HTTP Ecosystem

- We refer as "HTTP Ecosystem" to a sort of critical mass of services that opens possibilities that are flexible, powerful and "industry standard"

- DPM and the other SEs provide now many features through HTTP/WebDAV
  - They are in production in many sites, ATLAS uses them
- LFC has an HTTP/WebDAV native interface
- The Dynamic Federations can federate this kind of things, including opportunistic storage and cloud storage

- This has to do with deploying testbeds, as they can be federated easily

EMI->DPM Collaboration

- The DPM Collaboration has been formed to support the ongoing development and maintenance of DPM.
- Started 2nd May 2013.
- Collaboration Meeting on 23rd April, 2013 with representatives from :
  - CERN
  - Czech Rep.
  - France
  - Italy
  - Japan
  - Taiwan
  - UK

What's in the collaboration (From April)

- Approval of a Collaboration Agreement
- Pledges of effort (over 4FTE)
- In return for influence on direction
- Approval of distribution of responsibility
- Development, testing, support inc all derived and related services
  - LFC and Dynamic Federations
  - HTTP/DAV, NFS, xrootd, gridftp frontends
  - S3, HDFS, ClusterFS backends
  - Memcached for the nameserver
  - Some new developments (eg cluster FS support)
- Best effort support will be provided to non-member communities

Partner responsibilities (From April)

- To give a flavour of tasks (incomplete list)
  - ASGC: NFSv4.1, performance testing
  - Czech: ClusterFS support, GSSAPI/X509
  - France: User support, testing, admin interface
  - Italy: Testing, federation
  - Japan: Testing
  - UK: Admin tools, support, performance evaluations

How can one contribute to the DPM collab

- Contribution: Run a Testbed
  - Install one with your preferred features
  - Make sure that you USE it. Can be Hammercloud tests, whatever
  - Raise your hand if anything breaks after an update of any component
  - From RC: May break sometimes, I think it happened once, lasted a few days
  - From EPEL-test: should never break! Important to doublecheck!
  - Think about a regional federation built on top of these (e.g. Italy)

- Contribution: develop new things
  - features: things that the system can do
  - components: new parts of the system
  - interfaces: ways to interact with the system

France in the DPM collaboration

- BDII reports 13PB handled by DPM in France
- User support, testing, admin interface
  - We had contacts on HTTP/WebDAV interfaces. Ours is minimalistic, plenty of features can be added using Web technologies
  - We had contacts about log handling/browsing/filtering/mining.
    - Can tools like Splunk be useful in our case?
    - Will the benefits pay for the effort?
  - Primary importance: EPEL-test testbeds
    - Primary importance. Alert us on time whether the features that are important for you stop working.
    - Make them available for Hammercloud testing
    - Make them available for HTTP federations or setup a National one
- Who comes at the DPM workshop in December?