Eclipse Long Term Support
EAC Session – Jan. 12, 2012

Agenda

- Common Build Infrastructure, Eclipse Platform, and Juno
- Why are we doing LTS?
- LTS elsewhere in open source
- LTS Duration and Cadence
- LTS Forge and Maintenance Committers
- LTS Readiness definition
- LTS Business & Governance model

Common Build Infrastructure

CBI Is:

- An initiative combining technologies and practices for building Eclipse Software
- An iteration of the CBI if you consider PDE, Athena, etc.
- Vendor Neutral C.I. Facility Operated by the Eclipse Foundation for Eclipse projects

CBI Initiative Goals:

- Make it really easy to build (& thus easier to contribute)
- Get all Eclipse Software build on Foundation hardware (& locally for developers)
- Enable the **LTS program**

Eclipse Platform build: Today

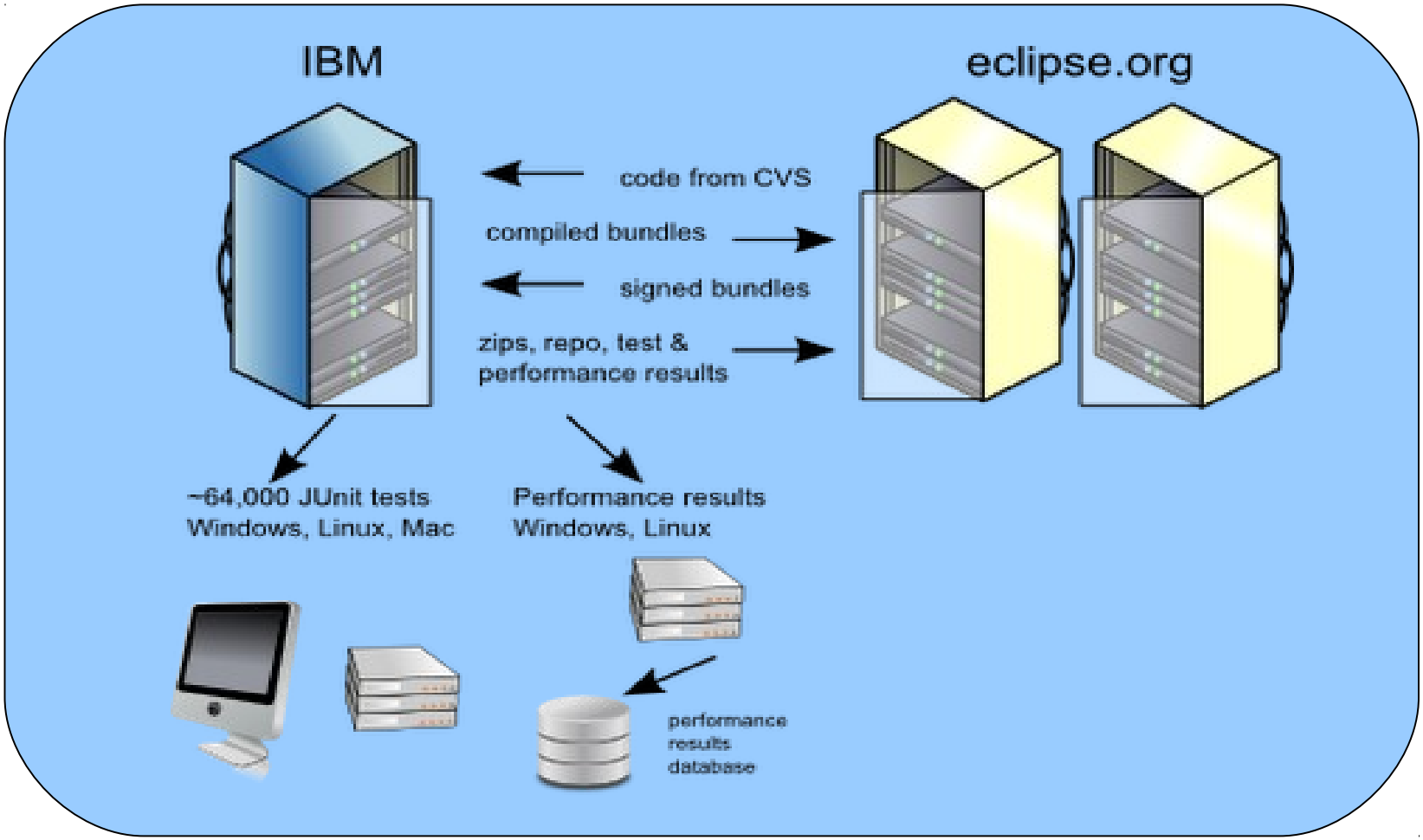


Image Credit: Kim Moir

Eclipse Platform build: Future on CBI

Prototype ready NOW

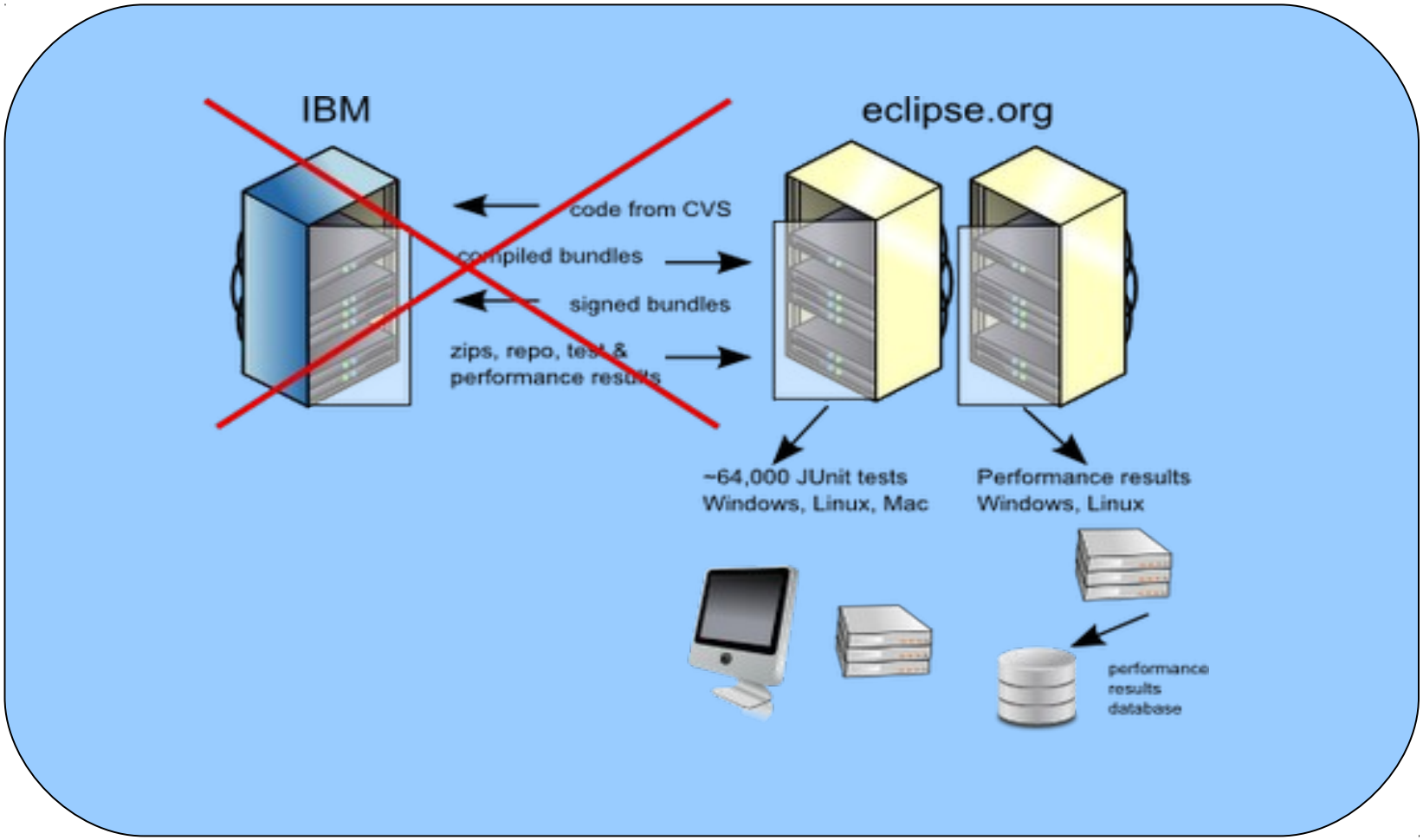
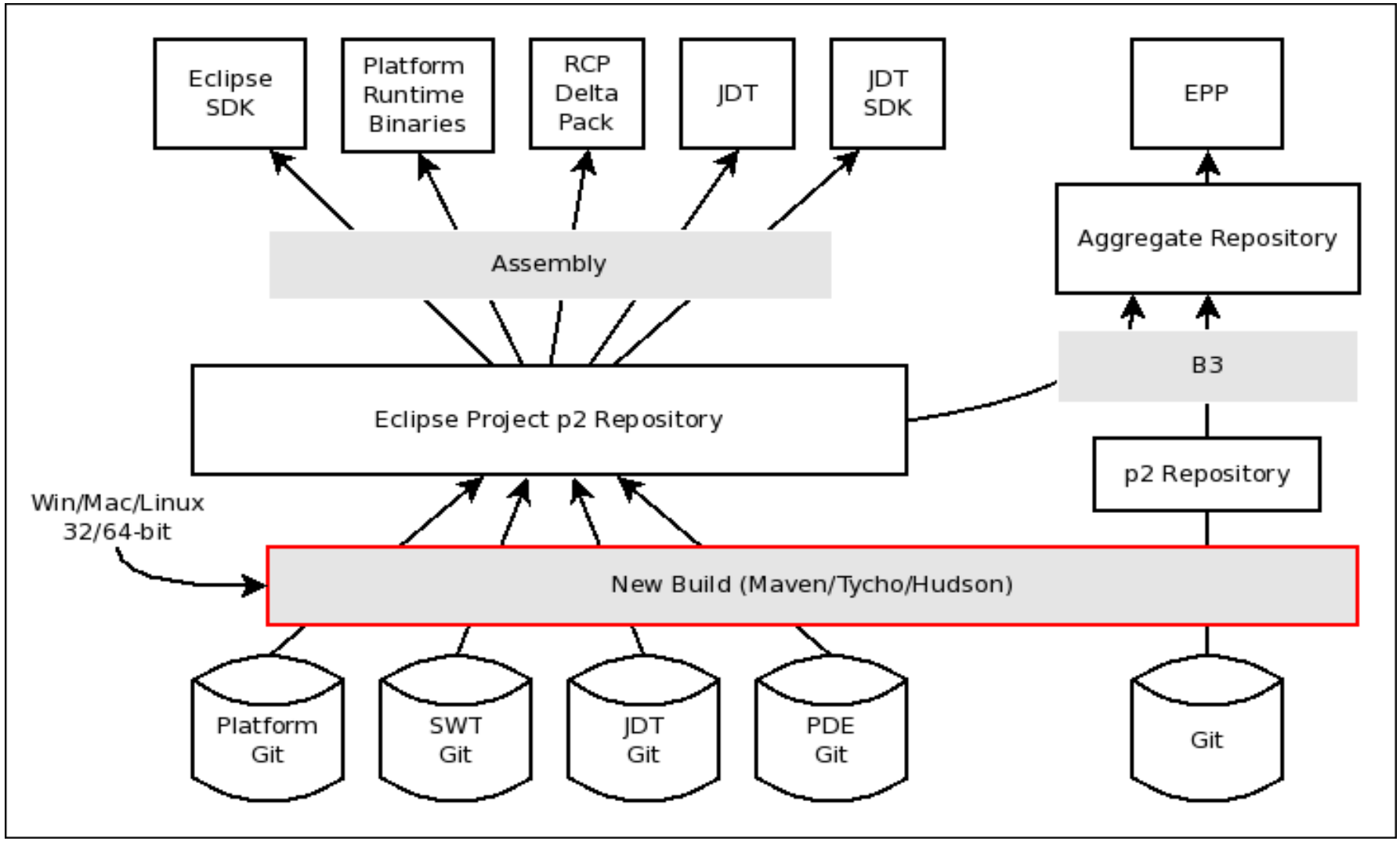


Image Credit: Kim Moir

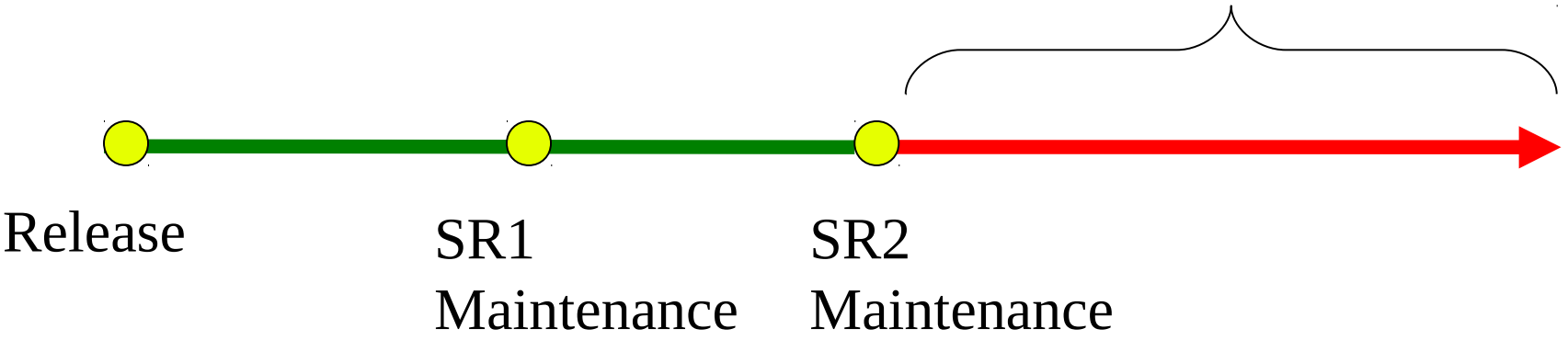
CBI Builds and Juno



The need for Long Term Support (LTS)

1) Support and maintenance offerings are needed that cover a variety of Eclipse projects and a number of releases.

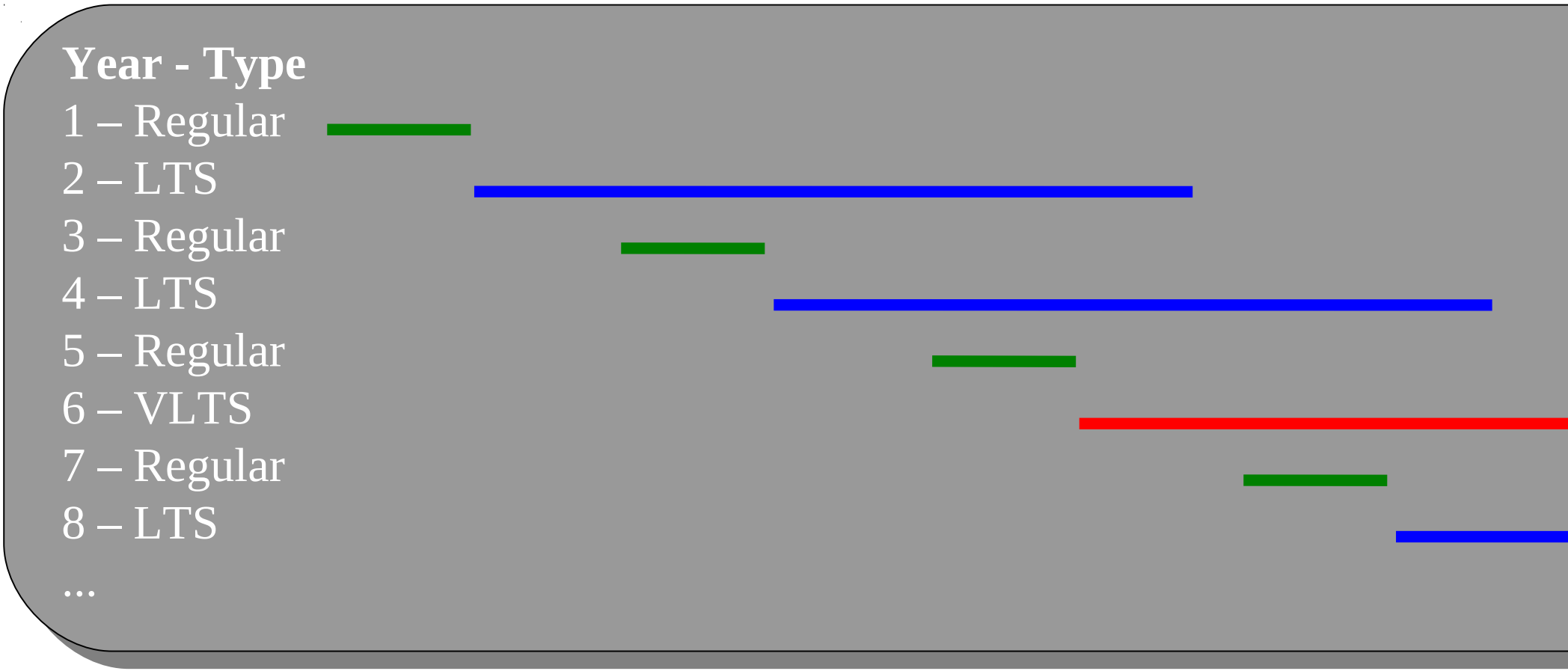
2) Product and service life-cycles are often much longer than Eclipse community based maintenance as it stands today. What do you do after SR2?



How Long?

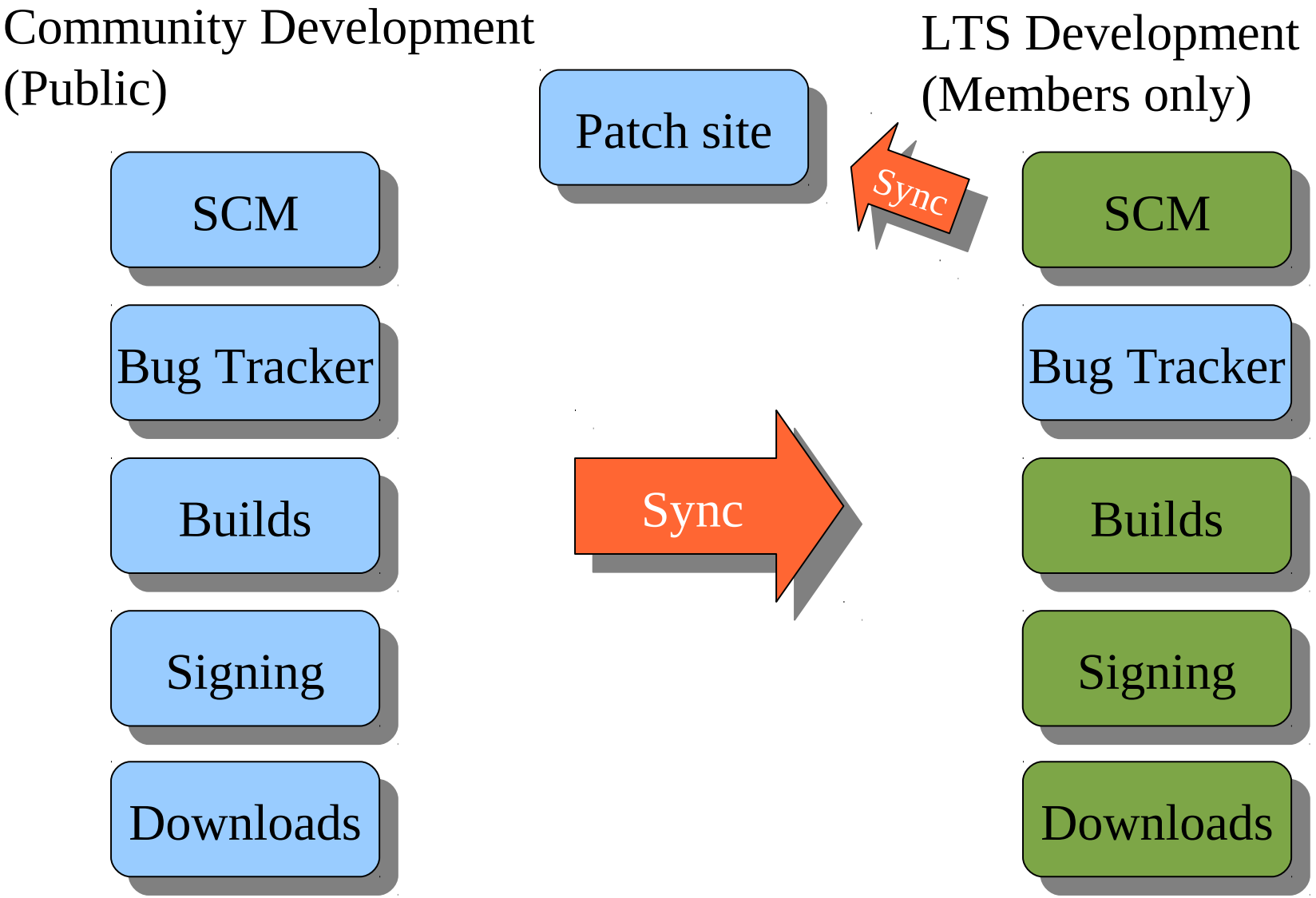
Organization	LTS Duration
Red Hat Enterprise Linux	4-10 Years
Ubuntu Linux	Desktop: 3 years Server: 5 years
MySQL	5 years to 9 years
Ingres	5 to 15 years
Asterisk	4 to 5 years
Mozilla Firefox	54 weeks

LTS Cadence



Juno will be the first LTS and VLTS release to prime the system.
Retroactive 3.6 & 3.7 LTS releases may be offered in the future.

Infrastructure



Maintenance committers

Role	Community Forge	LTS Forge
Committer	Access like normal.	Also get access.
Maintenance Committer	Need to earn commit	LTS Member organizations can appoint maintenance committers.

Are you LTS Ready?

To be considered LTS-Ready, a project must:

- Store code in a distributed code repository supported at Eclipse (e.g. Git)
- Offer a build that is: 1) documented 2) version controlled 3) automated 4) deterministic given the same code & third party libraries 5) can refer to compilers & tools from a non-standard location 6) capable of building without Internet 7) only pulls dependencies from Foundation controlled sources such as Orbit, Maven.eclipse.org, downloads.eclipse.org, etc
- Adheres to Eclipse IP policies
- Manage bugs in Bugzilla and have meta data capable of routing bugs to each LTS release

LTS Member privileges (draft)

Privilege	Steering Committee	Participating
Member of the Steering Committee	X	-
Member of the Change control board	X	-
Host custom build on LTS infrastructure	X	X
Host custom branch on LTS infrastructure	X	X
Access the LTS build and signing infrastructure	X	X
Access binary releases	X	X

Q&A

Thank you!

For more information: http://wiki.eclipse.org/LTS

Back up materials

Git has Taken the Lead

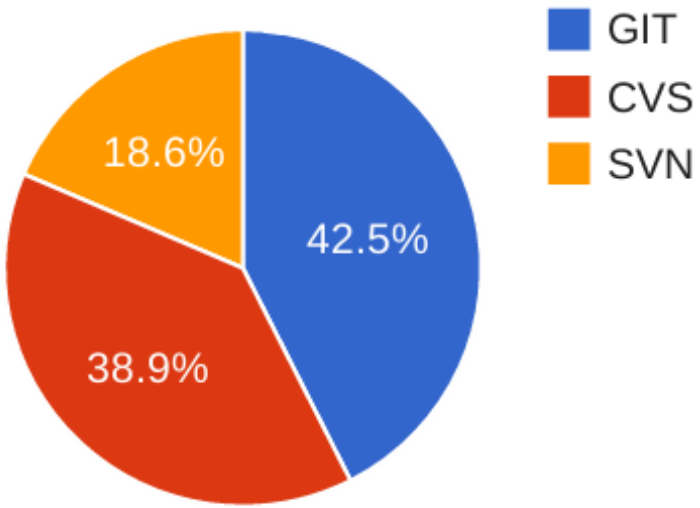
94 Projects currently have Git repositories

On December 21/2012 CVS becomes read-only; SVN to be deprecated soon

Need help?

Post a bug against

Community/Git



http://wiki.eclipse.org/Git/Migrating_to_Git