Continuous Integration

Using Open Build Service (OBS)

Adrian Schroeter
Project Manager
adrian@suse.com

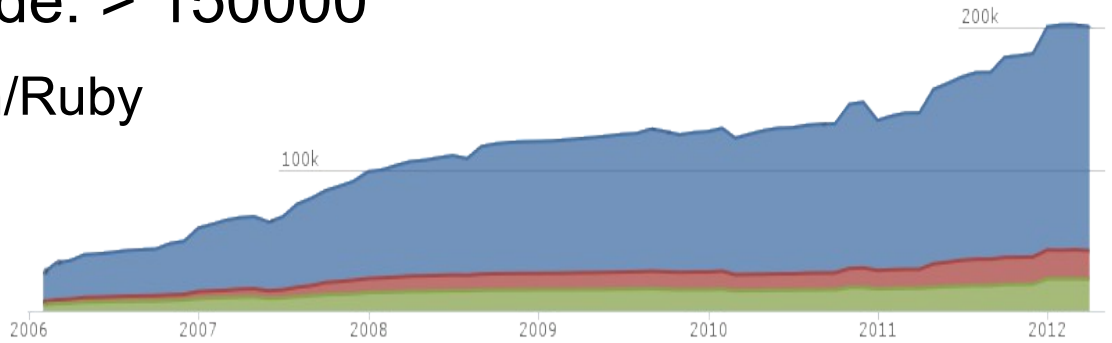
What is Open Build Service (OBS)?

OBS History

- Created in 2005 as a rewrite of SUSE's internal autobuild system
  - Goals: transparency, flexibility, openness
  - First presented at FOSDEM 2006
- 2010: OBS-1.5 openSUSE distribution got build by OBS
- 2011: OBS-2.1 openSUSE distribution submissions
- Current Release: OBS-2.4
  - Support for openSUSE maintenance updates

Development

- Licensed under GPLv2
  - https://github.com/openSUSE/open-build-service/
- Lines of Code: > 150000
  - Perl/Python/Ruby
  -

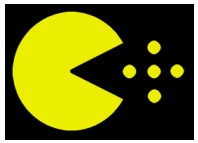


- Mostly maintained by SUSE, but many contributions from community members & other companies

Users

- Distribution development, Maintenance Updates

- Open Source Communities

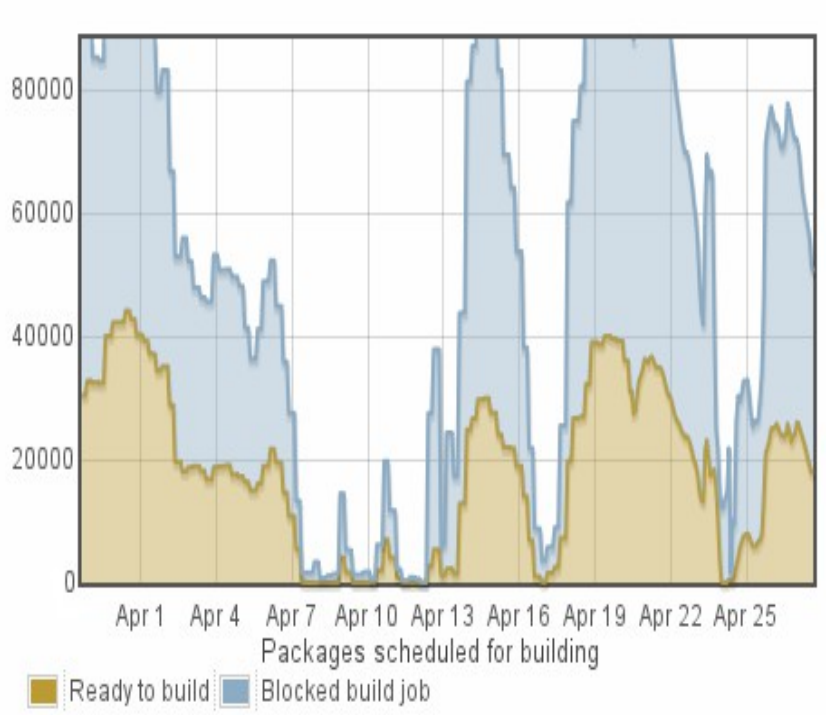


- Add-Ons: Driver Developer and ISVs

- Researchers/Universities
- Administration Teams

Numbers (from build.opensuse.org)

- Confirmed Users: >37000
- Package builds per day: > 75000
  - Build farm: 70 hosts, 440 workers
- Storage:
  - Sources: 7 TBytes
  - Binaries: 19 TBytes



Support

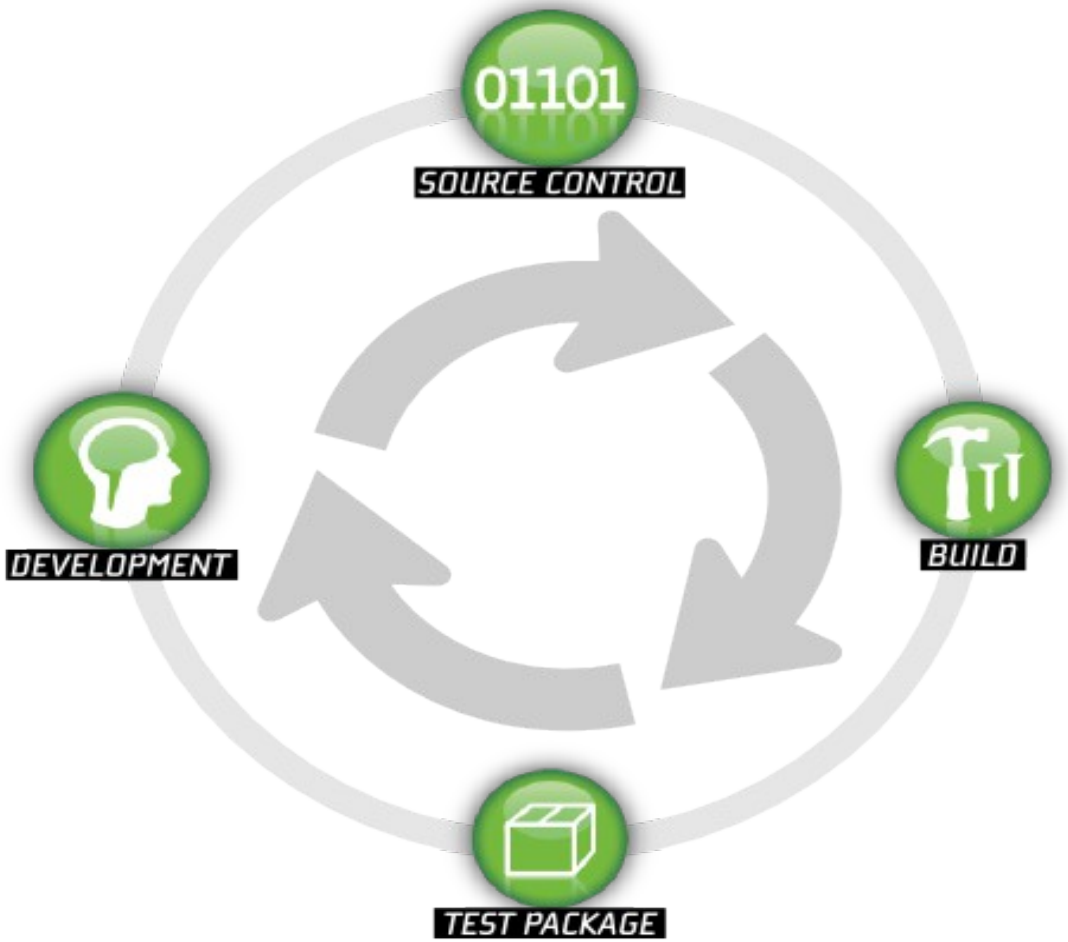
- Community
  - opensuse-buildservice@opensuse.org
  - Irc: #opensuse-buildservice on freenode
- Professional
  - http://www.open-build-service.org/contact/
  - B1 Systems (L3 backing by SUSE)

Features

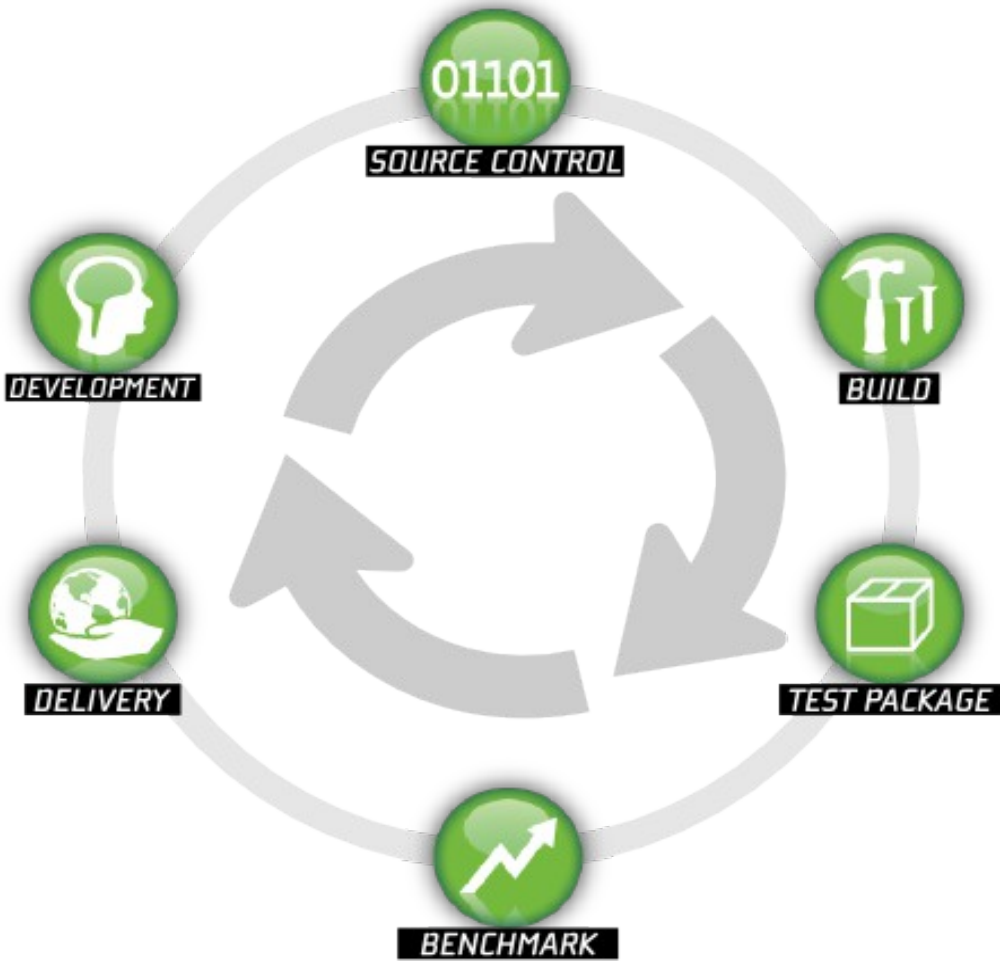
- Multiple distributions, multiple architectures
  - i586, x86_64, ppc64, armv*, aarch64, ….
  - rpm, deb, archlinux, image creation
- Sand-boxed builds (kvm/xen) on a build farm
- Easy branching with automatic merges
- Continuous Integration
  - Automatic rebuilds on changes (both source and build packages), automatic ordering of builds
  - **Consistent, reproducible builds**

Continuous Integration

What Is CI?



What Is CI?


01101
SOURCE CONTROL
BUILD
TEST PACKAGE
BENCHMARK
DELIVERY
DEVELOPMENT

What Is CI Good For?

- Errors are catched as early as possible
- Constant Quality Assurance
- Production deployment can be integrated

- Saves time and money in bug hunting
- Restructuring code can be done with less risks

What Is Needed For CI?

- All components accessible
- Non-interactive build possible
- Non-interactive tests possible
- Difference between test setup and production must be minimal

Different CI Systems (examples)

- Jenkins
  - Highly flexible via scripts
  - Focus on Java
  - Needs a worker for each component and base OS
  - Server needs to be setup
- Travis-CI.org
  - Can be integrated nicely with github.com
  - Needs a worker for each component. Only one single base OS.
  - No in-house option, but freely available for open source.

What Can OBS Offer?

- Final product, including appliance media can be build and tested
- Another base OS configuration is easy and fast to setup
- Reproducible builds including the base OS
- Public and private instances available
- Scalability

How To Do It With OBS

Source Update Variant #1

- Write own script and submit it frequently

```
# bash my_source_update_script.sh
# osc addremove
# osc ci -m “update sources”
```

Source Update Variant #2

- Define tar ball updates via source service

One time

- Write spec file

osc add git://....

Update it via

osc service remoterun <PROJECT> <PACKAGE>

or simple HTTP api call

Source Update Variant #3

- Create a source service like in #2
- Create an authorization token

  # osc token –create

=> Update sources via api call using the token

=> Or use github.com OBS service hook

Test Cases

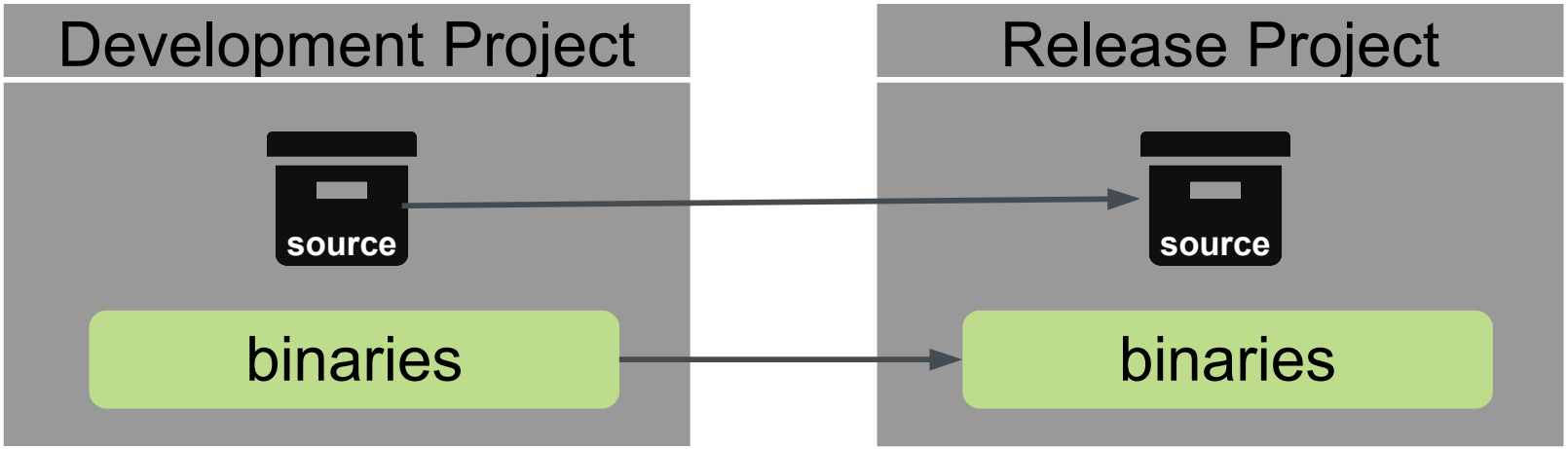
- Package specific test cases in %check
- Additional packages can be used to test multiple packages.
- Appliance build
  - Product version
  - Test version running tests on boot-up

Benchmarking

- Ensure to build on a reliable host via build constraints
- Export benchmark results as extra files
- Use build times as benchmarks

How To Deliver

- Successfully tested builds can be released to another repository:
  - Define releasetarget in repository
  - # osc release <PROJECT>



Future

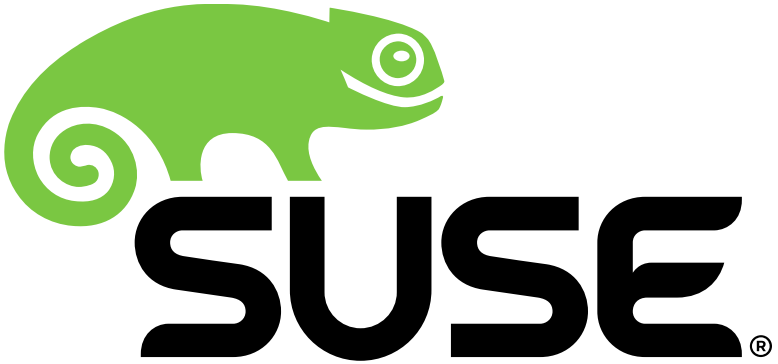
SCM Integration

- OBS will be able to host git trees
- Tar balls will be created during build time only
  - Reducing storage and bandwidth needs
- Plug-able commit notifications at least for github.com

QA Support

- OBS handle build and QA results independent
- QA tests for appliances will be possible
- Auto-apply of test cases affected by component
- Auto-branch and test runs for github.com pull requests

- Multi-instance network setup test cases



Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org