

# Koschei

## Continuous integration in Koji

Author:

*Mikolaj Izdebski* [mizdebsk@redhat.com](mailto:mizdebsk@redhat.com)

Date: *11th July 2014*

### Abstract

Koschei is a service for scratch-rebuilding RPM packages in Fedora Koji instance when their build-dependencies change or after some time elapse.

This presentation is about the problem Koschei is trying to solve, design decisions, system structure, current status, plans for the nearest future and further evolution possibilities.

Section 1

# The problem



## Where is the problem?

- Buildability as a measure of software quality
  - tests ran during build
- Constantly growing number of packages
  - software collections
- People are unaware of FTBFS
  - bugs are not seen until mass rebuild
  - or worse, until there is critical bug to fix

## Time elapse

Time elapse increases cost of fixing bugs

- People forget what they were working on
- More bugs appear
  - Harder to discover where the real problem is
  - Fixing means working in recursive, parallel mode
    - to fix A you need to fix B first
    - Koji repo regeneration
    - ARM builders

Section 2

## **The solution**

## What can be done

### Continuous integration

- continuous monitoring of package buildability
- helping maintainers to reason on FTBFS

## How?

- Rebuild all packages from time to time
  - weekly?
  - too long delay
- Rebuild important packages more often
  - nightly?
  - only a few packages can be rebuilt
- Rebuild all rev deps after each update
  - way too much resources needed
- Middle ground solution?

## Where?

- Options considered
  - maintainers' machines
  - Fedora Koji
  - Copr
  - cloud
- The choice – Fedora Koji
  - existing, stable platform
  - spare resources
  - maintained by Fedora infrastructure
  - no networking problems
  - canonical build environment



## Koschei

A tool for continuously scratch-rebuilding packages using Fedora build infrastructure – Koji

## Etymology

**KO**ji **C**ontinuous **I**ntegration

Where did the name came from

```
$ grep -xi ko*c*i /usr/share/dict/words  
Koschei
```



Section 3  
**Design**

## The concept

- A set of packages
- Reporting buildability
- Resource monitoring
- Rebuild prioritizing

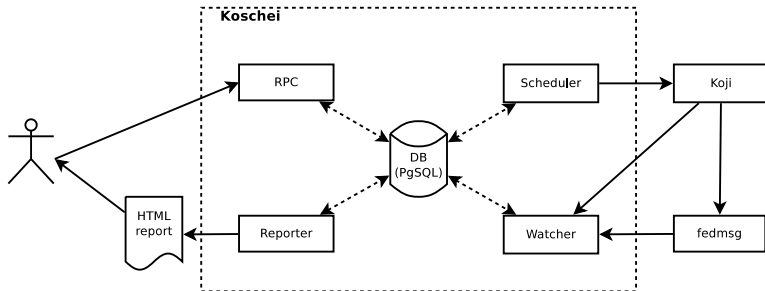
## Priority

- Time since last rebuild
- Dependency changes
  - consider distances
- Previous state
  - prioritize failures
- Importance
  - aka static priority
- Manual trigger
  - aka dynamic priority
- Plugins

# Database

- Packages
  - name
  - priorities
- Builds
  - status
  - Koji task ID
  - time stamps
  - logs
- Repositories
  - dependencies
- Package groups

# Architecture overview



# Watcher

- Await Koji build state changes
  - fedmsg
  - periodic polling as fallback
- Await new Koji repos
  - not builds, not tags
  - fedmsg
  - no polling
- Analyze dependency changes
  - hawkey
  - download SRPM headers
- Update priorities
  - increase priority on dependency change
  - reset priority on build success



# Scheduler

- Schedule builds for execution
  - priority scheduling
- Conditions
  - package is not disabled
  - build dependencies are resolvable
  - priority is high enough
  - Koji load is low enough

## Submitter

- Request scratch builds on Koji
  - from existing SRPM
  - very low priority
  - needs Koji certificate

## Reporter

- Generate HTML reports
- Per group, not per maintainer
- Failures separately
- Dependency problems
- Detailed package history

# RPC

- Add and disable packages
- Adjust package importance
- Force build



Section 4  
**Implementation**

# Implementation

- Python
- PostgreSQL
- SQLAlchemy, Alembic
- Modularity
- systemd

## Current state

- code at Github
- packaged as RPM
  - not yet in Fedora
- running at Openstack

## Creating SRPM metadata

```
$ curl http://koji.fp.o/.../eclipse-4.4.0-5.fc21.src.rpm \  
| tee package.src.rpm \  
| rpm -qp /dev/stdin >/dev/null  
curl: (23) Failed writing body (2332 != 4096)
```

```
$ ls -go  
total 20  
-rw-rw-r--. 1 20480 Jul 11 09:10 package.src.rpm
```

```
$ createrepo .  
Spawning worker 0 with 1 pkgs  
Workers Finished  
Saving Primary metadata  
Saving file lists metadata  
Saving other metadata  
Generating sqlite DBs  
Sqlite DBs complete
```





Section 5  
**Future**

# TODO

- Move to Fedora
  - within of scope of Env and Stacks WG
  - already announced
  - cloud machine
  - Koji certificate
  - extra Koji hardware?
  - storage?
- Improve reporting
  - feedback and new ideas needed!
- Generate SRPM metadata
  - compose is too late

# Links

- Code repository
  - <https://github.com/msimacek/koschei>

The background is a dark blue, textured surface. It features several wavy, light-colored lines that sweep across the frame. In the lower right quadrant, there is a cluster of small, light-colored birds in flight, scattered across the blue background.

The end.

Thanks for listening.