

Pulse

pulse.mozilla.org

Initial Reasons

- Familiar with a simpler WebHook system attached to the build system @ Apple
- Sign up, give URL, POSTed when builds start, finish, etc
- Feel blind during release development
- I'm lazy and want to write tools to automate my job

Vision

- All interesting Bugzilla events
- All interesting HG events (pushes, tree closures, etc)
- All interesting build events (build started, build finished, etc)
- All system status event
- All testing-related events (passes, failures)

Beginnings

- Started to write a simple WebHook system on top of whatever existing data I could find (build json, hg web, etc)
- Complexity grew quickly
- Why am I reinventing the wheel?

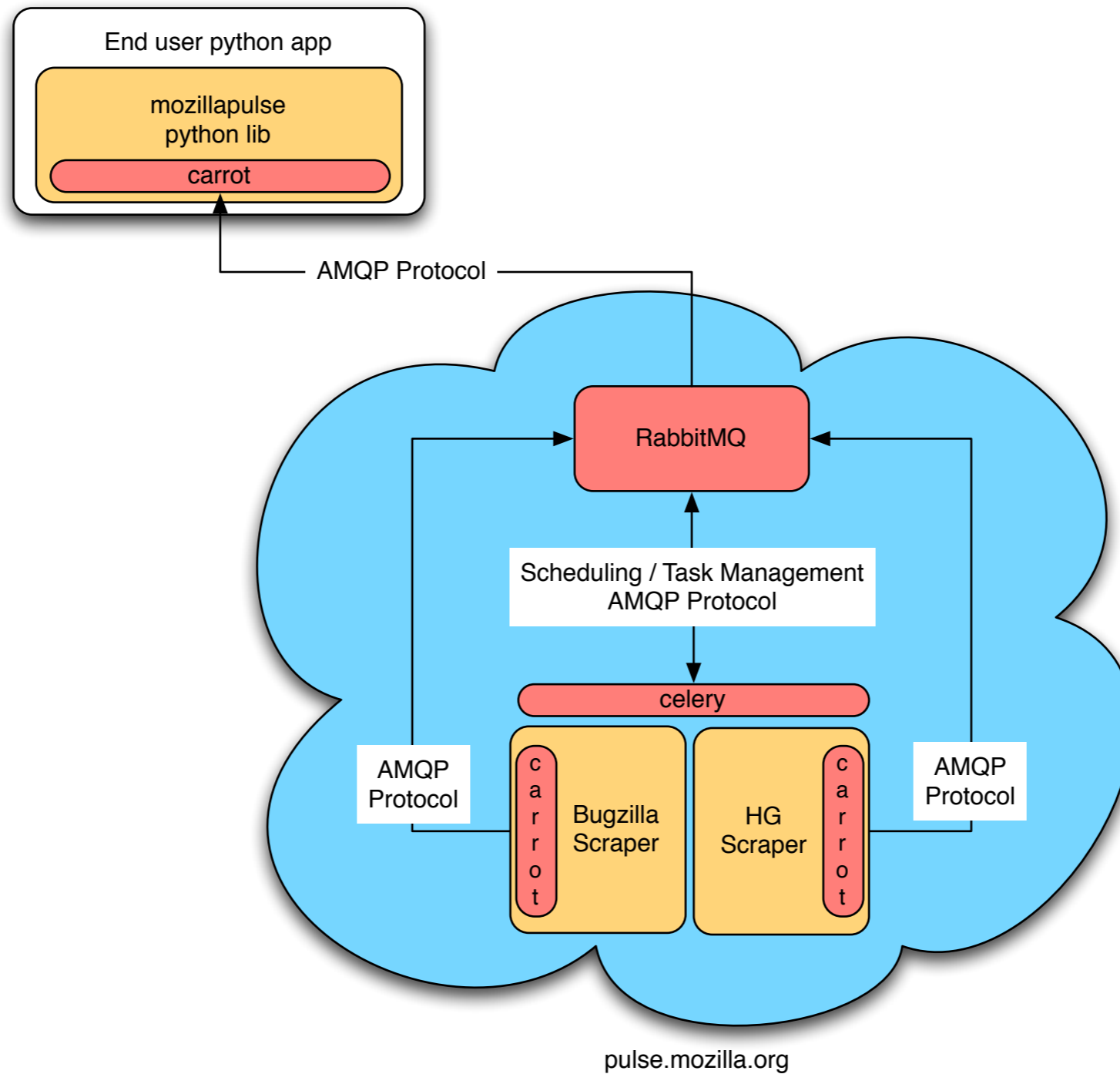
Message Broker Concepts

- Exchanges
- Queues
- Bindings
- Routing keys
- Certain guarantees
- No historical storage...not a database

Summit

- Set up RabbitMQ on pulse.mozilla.org
- Created a simple python lib so users didn't need to know about AMQP details
- Created scrapers that fed information into the broker
 - Bugzilla API
 - HG web / RSS

Summit Architecture



What?

- **RabbitMQ** - Open source message broker written in Erlang
- **AMQP** - Open source protocol for clients to talk to a broker.
Currently only really supported by RabbitMQ and Redhat MRG, though others have pledged to support it
- **STOMP** - Open source protocol for clients to talk to a broker.
Supported by all brokers, but doesn't have many features
- **Carrot** - Open source python lib with backends that support STOMP, AMQP, etc

Excitement!

- Tons of interest at the summit
- Catlee instrumented buildbot and published into pulse
- TBPL wanted to switch right away
- Kept coming up as a solution when new tools were discussed

Issues

- RabbitMQ would periodically hang
 - Needed a reset, deleted messages
- Permissions?
- AMQP versioning woes
- Scraper lag and consistency
- No (easy) support for website consumers

Fixing Scrapers

- They were a stop gap so that existing systems wouldn't need to change for a prototype project
- Known inherent problems with approach
- Why waste time working on them?

Bugzilla Extension

- Bugzilla has an extension system, but there are bugs
- Perl AMQP libs are fairly unsupported
- Will it bring down b.m.o?

What I did

- Patched Bugzilla to make the extension work (bug 587793 and bug 590334)
- Contacted AMQP perl lib maintainers
- Created bugzilla-amqp project on github
- Rolled out on landfill.bugzilla.org
- Nokia and the RabbitMQ guys want to use the extension

HG Hook

- Created a hg hook
- Tested it locally
- Fairly straightforward, though there is more to think about
 - How much processing up front?
 - Shadow repos?

Is RabbitMQ Right?

- Slow consumer issues
- Stats / monitoring issues
- No built-in HA
- Erlang experience is hard to come by
- Management issues
- Permission issues

Broker Requirements

- Good user management / permissions
- Built in clustering and HA
- Interoperable with different languages natively or via STOMP
- Sane method for dealing with slow consumers
- Useful plugins available

Ditching RabbitMQ

- AMQP vs STOMP
- Other broker overview
 - ActiveMQ
 - HornetQ
 - Qpid / Redhat's MRG

ActiveMQ

- Don't want use
- Seems to fit requirements best
- Doesn't support AMQP
- Java w/ verbose XML config files (yuck)

AMQP to STOMP

- Had to change the Bugzilla extension and HG hook to support STOMP as well
- Had to patch carrot's stomp backend
- Broker-specific extensions and behavior
 - Switched publishers and consumers to STOMP with RabbitMQ, same setup didn't work for ActiveMQ

Hard problems

- Message security (Bugzilla mainly)
- Message verbosity
- Upstream processing vs downstream
- Error detection and recovery (is it even needed)

See also

- Docs on pulse.mozilla.org
- Mozillapulse python lib
http://hg.mozilla.org/users/clegnitto_mozilla.com/mozillapulse
- Bugzilla extension
<http://github.com/LegNeato/bugzilla-push>
- HG Hook
http://hg.mozilla.org/users/clegnitto_mozilla.com/hg-broker/