

AMQP and Beyond

Messaging by Extending RabbitMQ

Tony Garnock-Jones <tonyg@rabbitmq.com>

Why messaging?

Database *is to* Filesystem

as

Messaging *is to* Network

Messaging abstracts away from the details of
your network topology

Why messaging?



SQL

Database *is to* Filesystem

as

Messaging *is to* Network

Messaging abstracts away from the details of
your network topology

Why messaging?

SQL

Database *is to* Filesystem

AMQP

as

Messaging *is to* Network

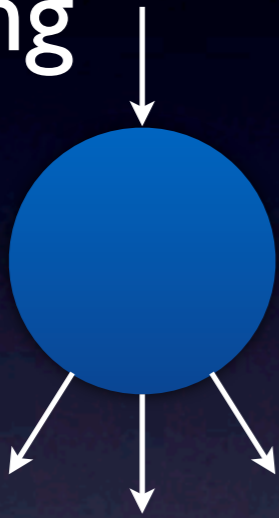
Messaging abstracts away from the details of
your network topology

Why messaging?

- Scaling, load-balancing
- Delayed jobs, task queues
- Multicast, broadcast
- Trusted store-and-forward
- Management, monitoring
- ★ Decoupling of components

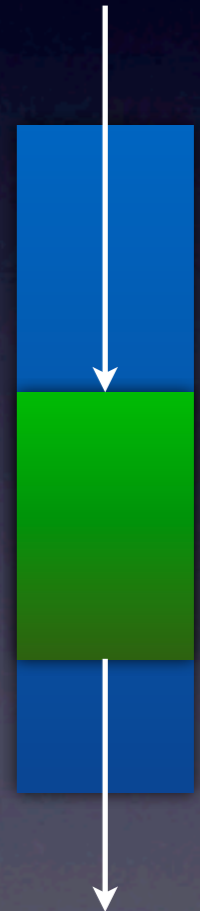
What is messaging?

Naming

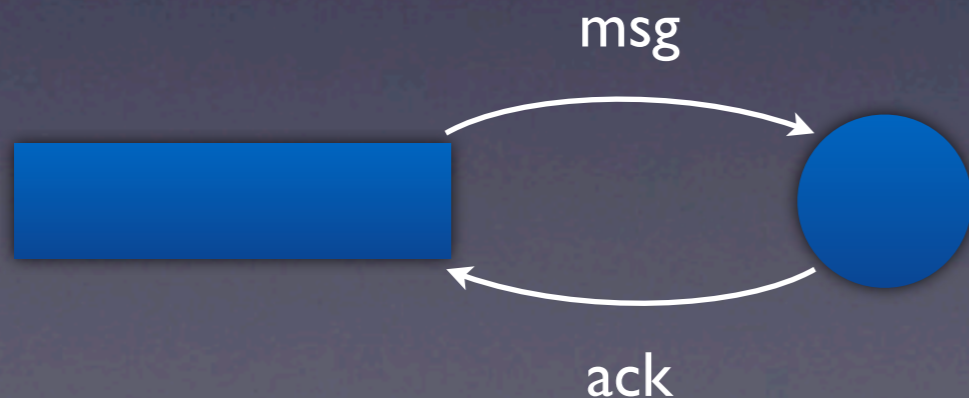


Relaying and filtering

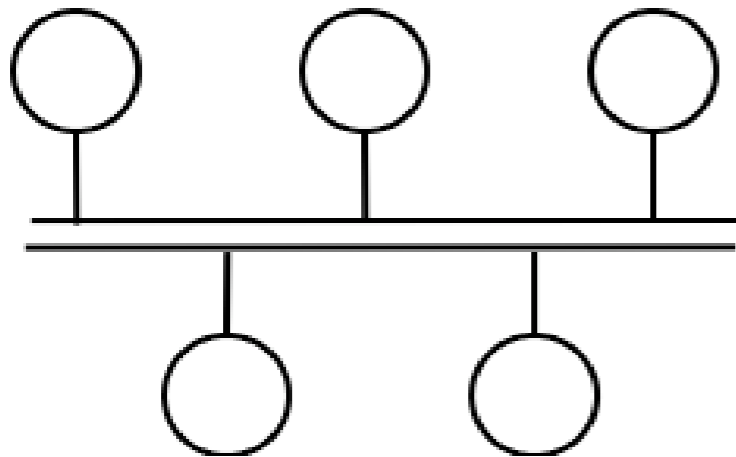
Buffering and queueing



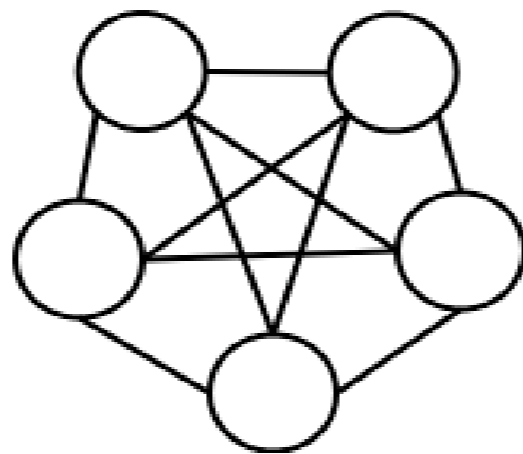
Transfer of responsibility



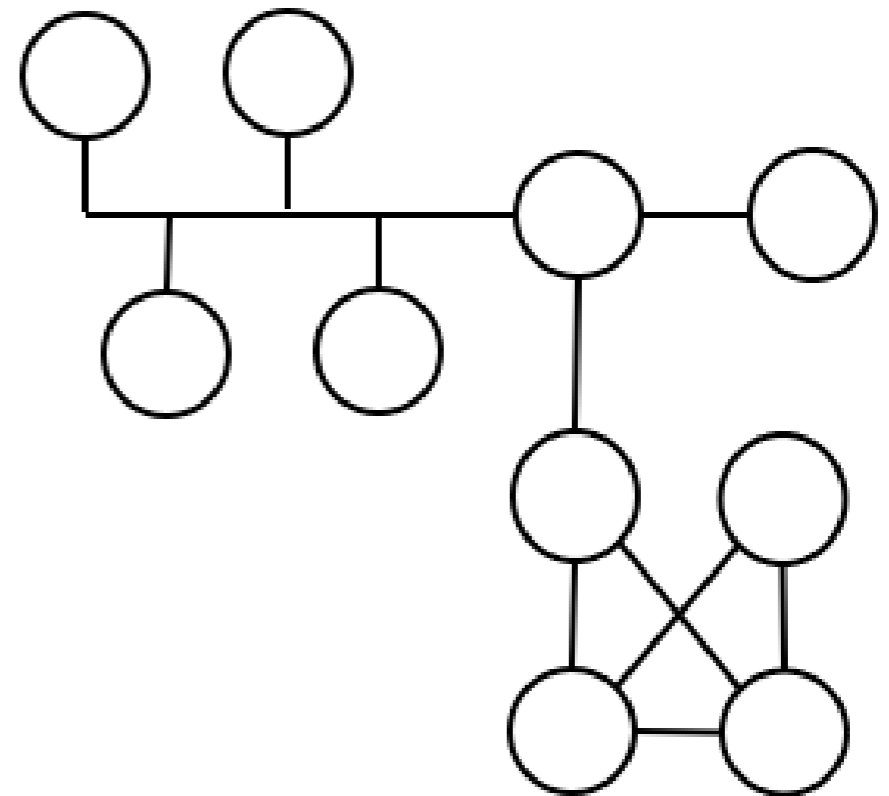
Where is messaging?



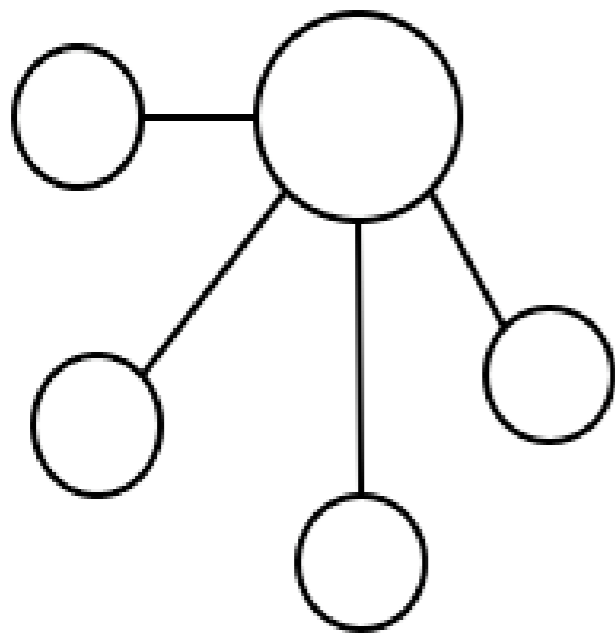
Enterprise Service Bus



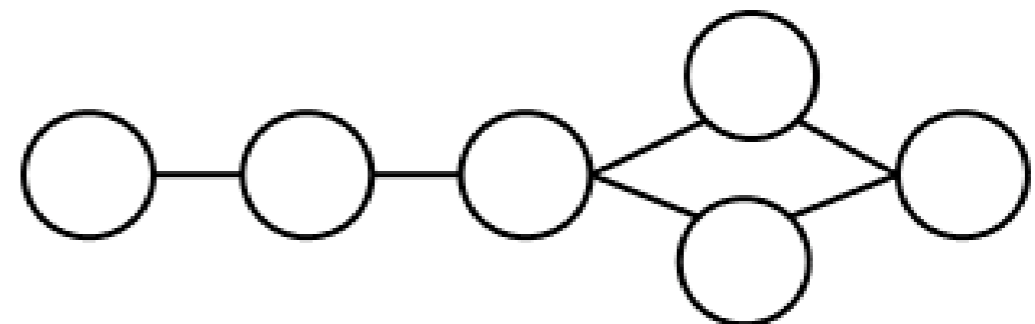
Peer Network



Enterprise Service Network



Client/Server and Hub n' Spoke



Pipeline

Why AMQP?

The Options

- JMS?
 - API-only: lots of baggage if you're dealing with multiple brokers; lock-in
 - Fine for Java, but ...
- Tibco RV? \$\$\$
- IBM MQ Series? \$\$\$

The Options

- XMPP?
 - transport, not messaging (modulo xep60)
- HTTP?
 - transport, not messaging
- SMTP + mailman?
 - slow, heavy; fiddly to set up
- Redis, kestrel, starling, ...
 - vertical solutions

The Options

- Queue up work in your DB?
 - Not designed for messaging!
 - Scaling up can be awkward
- Use the filesystem?

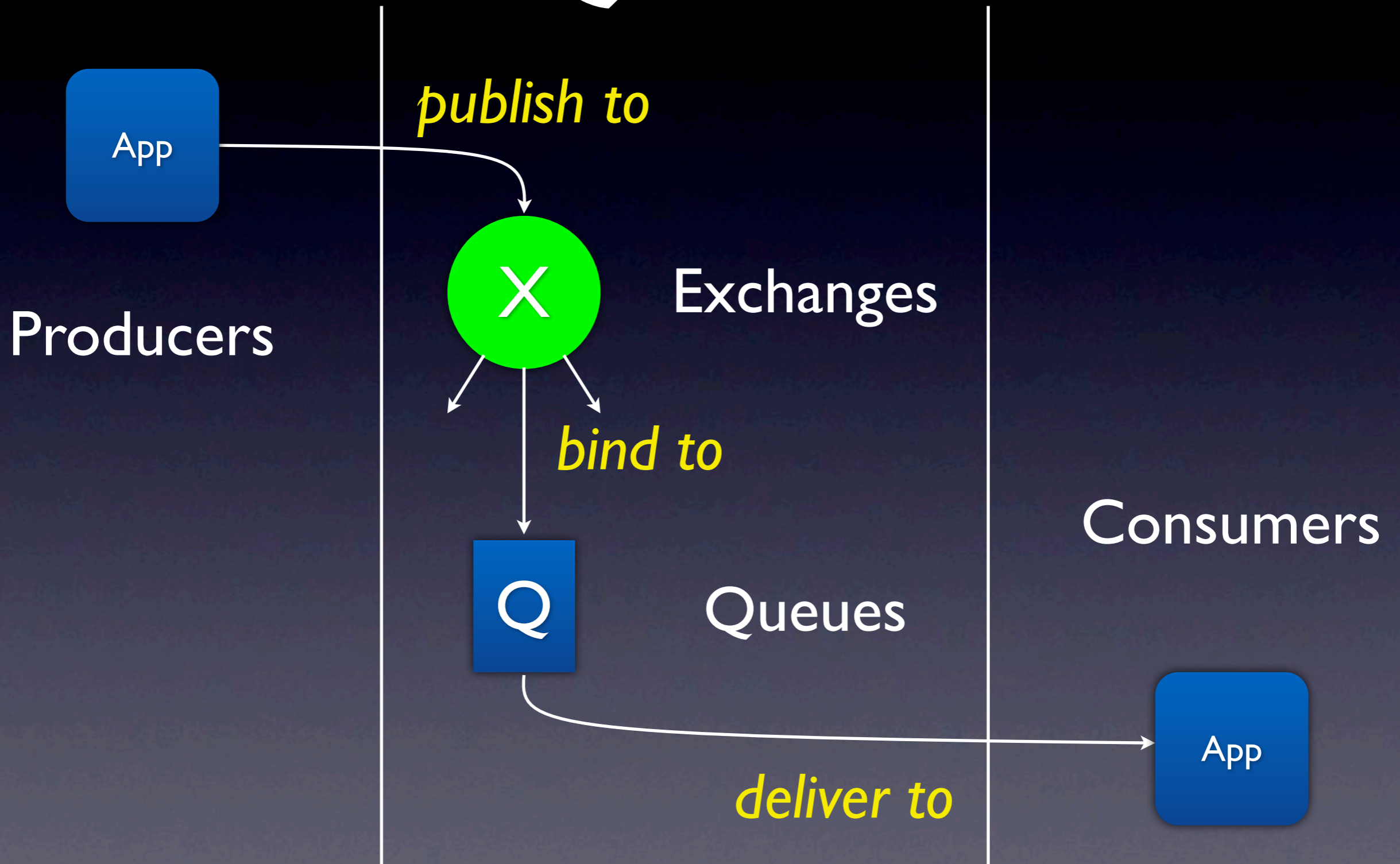
AMQP

- General *model* for messaging
- Simple, efficient *transport*
- Wire-level protocol helps avoid vendor lock-in
- API-neutral, multi-language
- Configuration & management within the protocol

AMQP Specification

- Version 0-8 released June 2006
- ★ Version 0-9-1 released Nov 2008
("Interop" release of spec)
- Version 1.0 being developed currently

AMQP Model (0-9-1)



RabbitMQ

Goal: Do the right thing, out-of-the-box

“RabbitMQ is a pleasure to use and it just works.
Everyday, every time, every message” - Michael
Arnoldus, project lead, algo trading firm

RabbitMQ

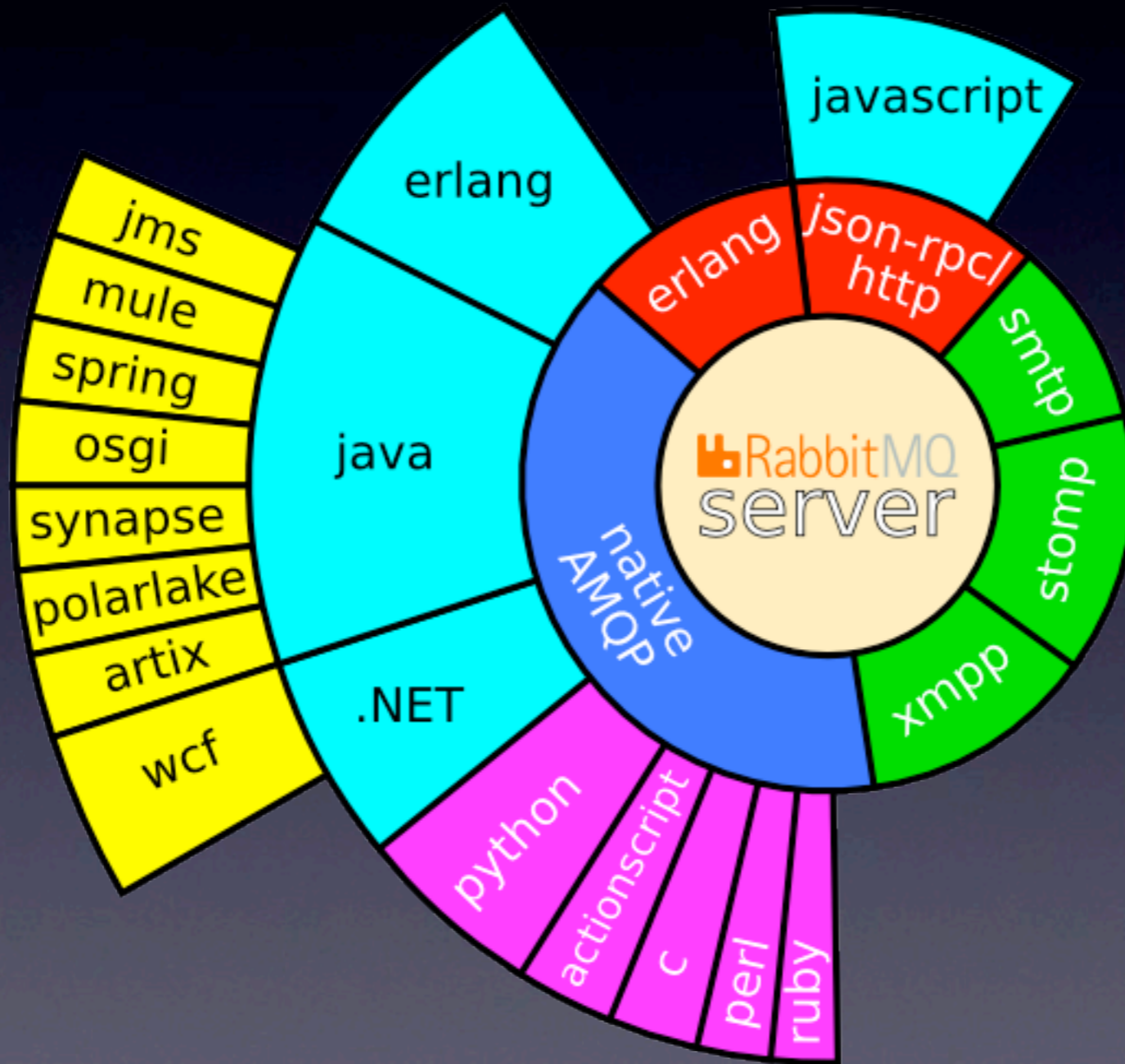
Goal: Do the right thing, out-of-the-box

“In my experience, you can have a clustered
rabbitmq setup running at home in under 20
minutes. It's all in the admin guide.”
Steve Jenson, Scala hacker

RabbitMQ

- Erlang/OTP AMQP broker
- Ships with Ubuntu
- Clients for C, Java, C#, Python, Ruby, PHP, ...
- Supports (almost) all the protocol
- Persistent, Transactional, Clusterable, ...
- Connectors for lots of other networks

RabbitMQ Universe



Beyond core AMQP

RabbitMQ Plugins

- Erlang *applications* bundled into a jar-like archive (*.ez)
- Booted as part of RabbitMQ startup
- “Boot steps” for integrating with Rabbit’s own startup sequence
- Exchange type registry (to appear in v1.8)
- Can do anything at all!

RabbitMQ Plugins

rabbitmq-status	Broker health check web page
rabbitmq-jsonrpc-channel	AMQP over JSONRPC, for browsers
rabbitmq-jsonrpc	JSONRPC server instance for Rabbit
rfc4627_jsonrpc	JSON and JSONRPC library
rabbitmq-mochiweb	Mochiweb instance for Rabbit
mochiweb	Mochiweb library plugin
rabbit_stomp	STOMP message transport
rabbithub	PubSubHubBub implementation
rabbitmq-bql	SQL-like Broker Query Language
amqp_client	Embedded AMQP client
rabbitmq-shovel	Broker-to-broker relay plugin
rabbitmq-toke	Tokyo Cabinet support for new persister

AMQP model limitations

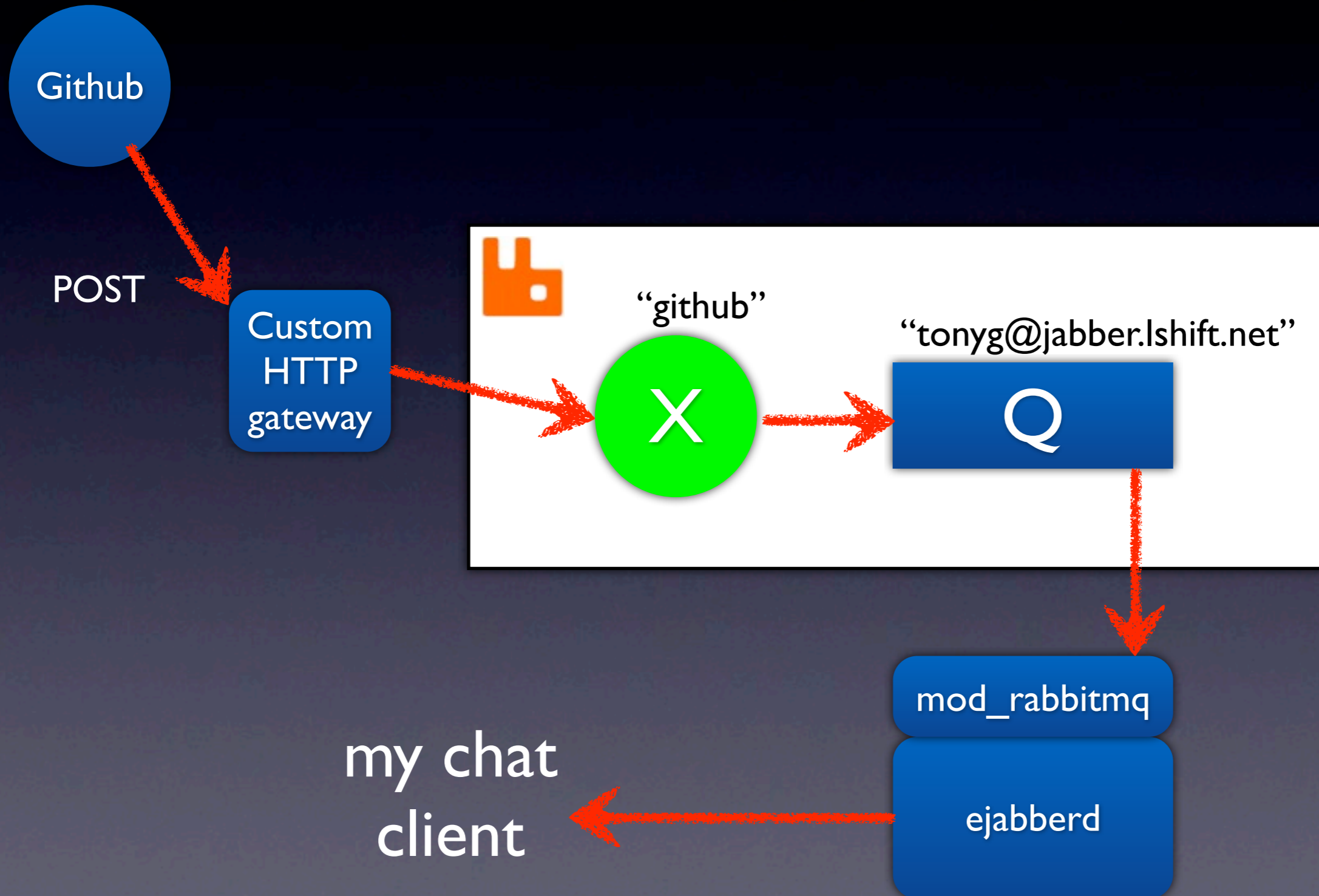
- No message *transformation*
- No custom message *routing*; you get exactly one of:
 - Fanout,
 - Direct,
 - Topic, or
 - Headers
- No *stateful multicast*

Example 1.

Github Commit Hooks

- URL-encoded JSON data blob
- Delivered by HTTP POST
- I want them in my XMPP client
- (Let's ignore Github's existing XMPP gateway!)

Github Commit Hooks



Github Community Books



Github



POST

Custom
HTTP
gateway

github@dev.rabbitmq.com/#

From: tonyg@jabber.lshift.net

```
r88%2fsrc%2fmochiweb_multipart.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_request.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_response.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_skel.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_socket_server.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_sup.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2fmochiweb_util.erl%22%2c%22deps%2fmochiweb-
r88%2fsrc%2floader.erl%22%2c%22deps%2fmochiweb-
r88%2fsupport%2finclude.mk%22%2c%22scripts%2frabbit.in%22%2c%22src%2
fMakefile%22%2c%22src%2frabbit.hrl%22%2c%22src%2frabbit_framing.hrl%22%2c
%22src%2frabbit.app%22%2c%22src%2frabbit_app.erl%22%2c%22src%2fra
bbithub_deps.erl%22%2c%22start-
dev.sh%22%2c%22start.sh%22%2c%22support%2finclude.mk%22%25d%2c%22auth
or%22%3a%7b%22email%22%3a%22mikeb%40lshift.net%22%2c%22name%22%3a%
22Michael%20Bridgen%22%7d%2c%22timestamp%22%3a%222010-03-
19T05%3a18%3a10-
07%3a00%22%2c%22id%22%3a%22bce55acf3b6d67d4cad6a731308aba4a4e777
4cc%22%2c%22modified%22%3a%5b%22.gitignore%22%2c%22Makefile%22%2c%2
2priv%2fwww%2fstatic%2fapplication.xsl.xml%22%2c%22src%2frabbit.erl%22
%2c%22src%2frabbit_auth.erl%22%2c%22src%2frabbit_consumer.erl%22%2
2c%22src%2frabbit_pseudo_queue.erl%22%2c%22src%2frabbit_subscripti
on.erl%22%2c%22src%2frabbit_sup.erl%22%2c%22src%2frabbit_web.erl%2
2%5d%7d%2c%7b%22added%22%3a%5b%22test%2fsend.sh%22%2c%22test%2fsub
.sh%22%5d%2c%22message%22%3a%22Update%20test%20scripts%20to%20be%2
0more%20generic%2c%20and%20to%20use%20plugin-
ified%20rabbit%20host%20and%20paths%22%2c%22url%22%3a%22http%3a%2
f%2fgithub.com%2ftonyg%2frabbit%2fcommit%2f97467e5a68e8adbb5bd958
ae368e5292010a28b5%22%2c%22removed%22%3a%5b%22test%2fsend_x_foo.sh
%22%2c%22test%2fsub_x_foo.sh%22%5d%2c%22author%22%3a%7b%22email%22
%3a%22mikeb%40lshift.net%22%2c%22name%22%3a%22Michael%20Bridgen%22%
7d%2c%22timestamp%22%3a%222010-03-19T05%3a21%3a21-
07%3a00%22%2c%22id%22%3a%2297467e5a68e8adbb5bd958ae368e5292010a
28b5%22%2c%22modified%22%3a%5b%22%5d%7d%5d%2c%22before%22%3a%22dc0
56d55112c436104b5d9a0b85d784f2110bc%22%2c%22after%22%3a%2297467
e5a68e8adbb5bd958ae368e5292010a28b5%22%7d
```

lshift.net"

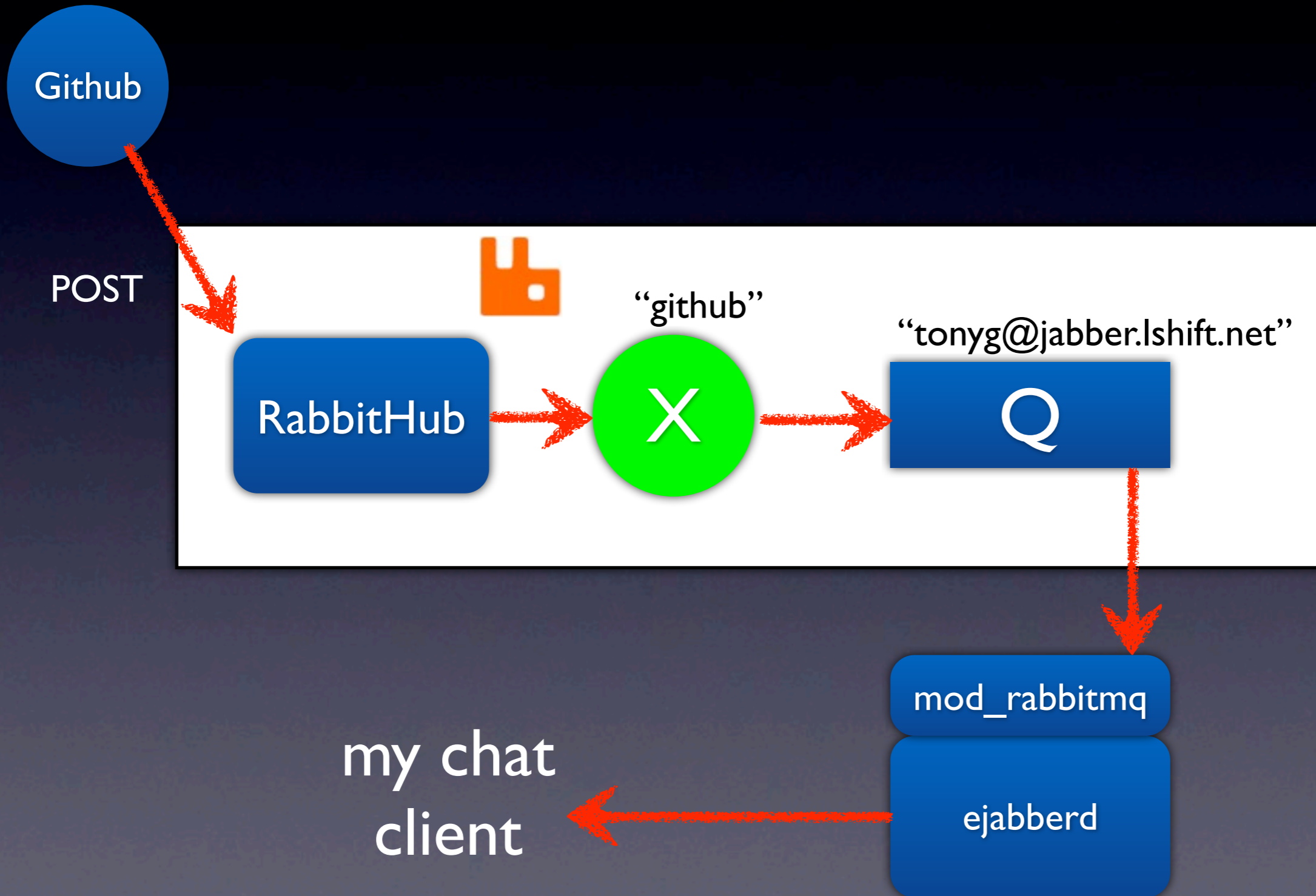
m

Donna github@dev.rabbitmq.com/#

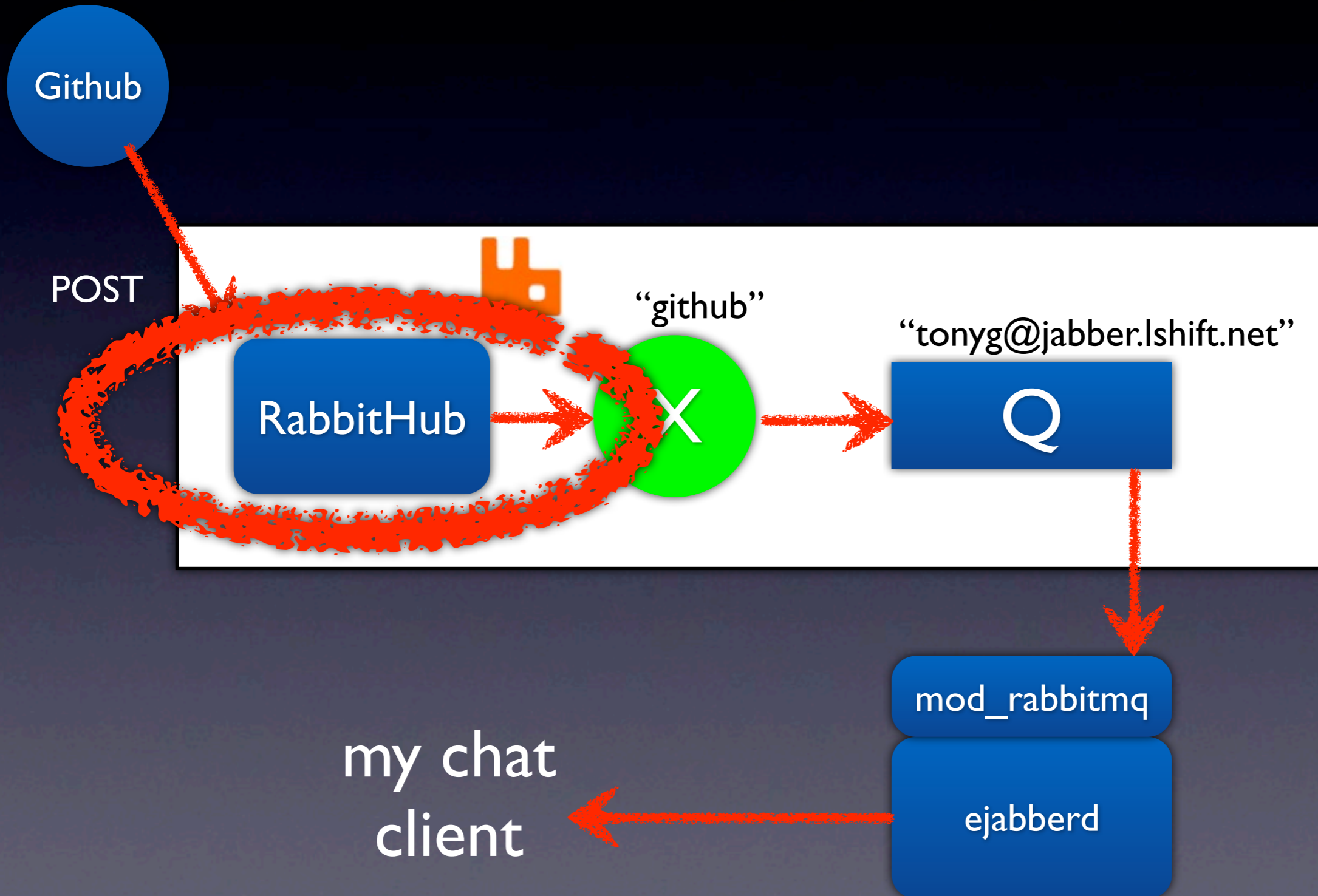
Problems

- Need to respond to HTTP POST
- Want to transform “payload=%7b%22repository%22%3a%7b%22watchers%22...” to something nicer before sending it out over XMPP

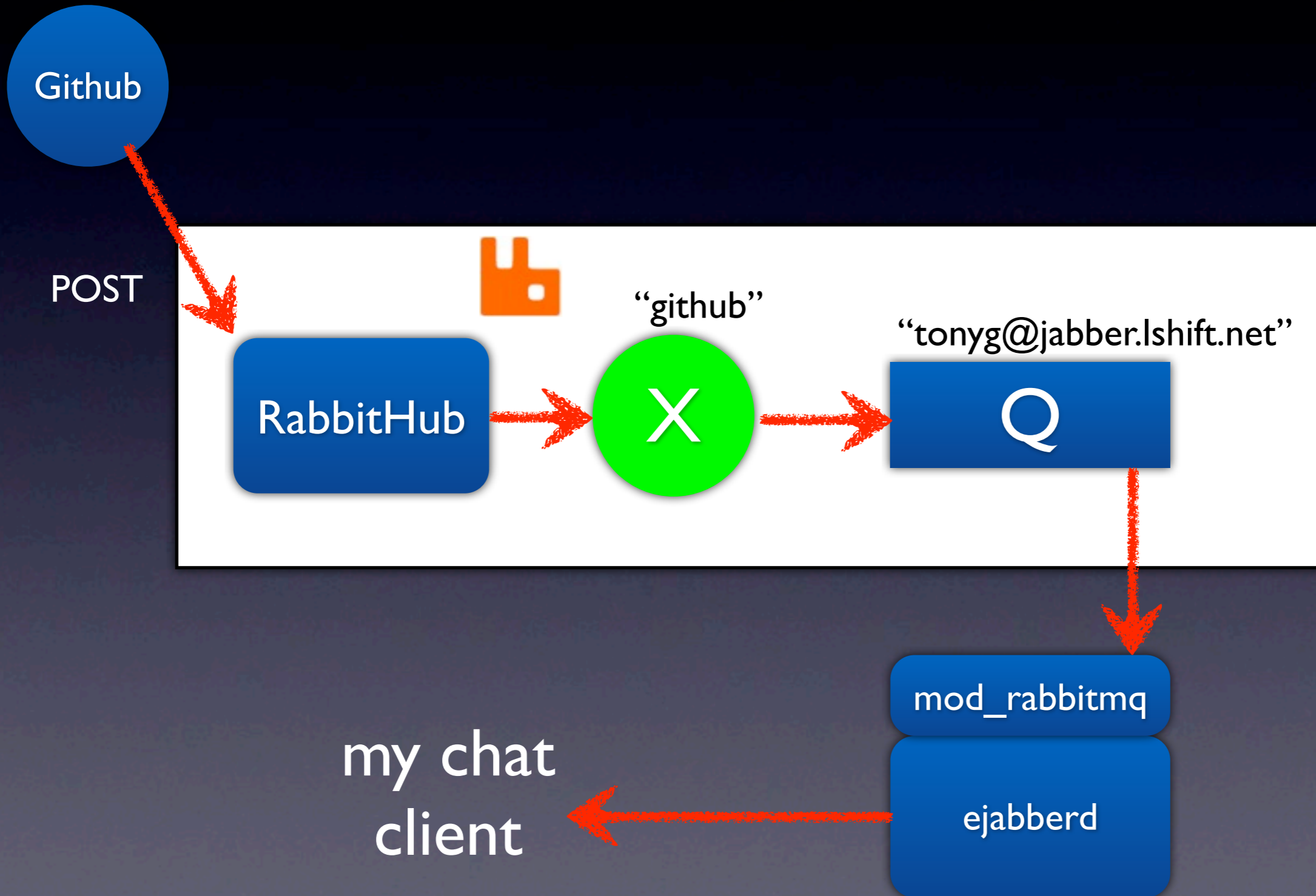
Github Commit Hooks



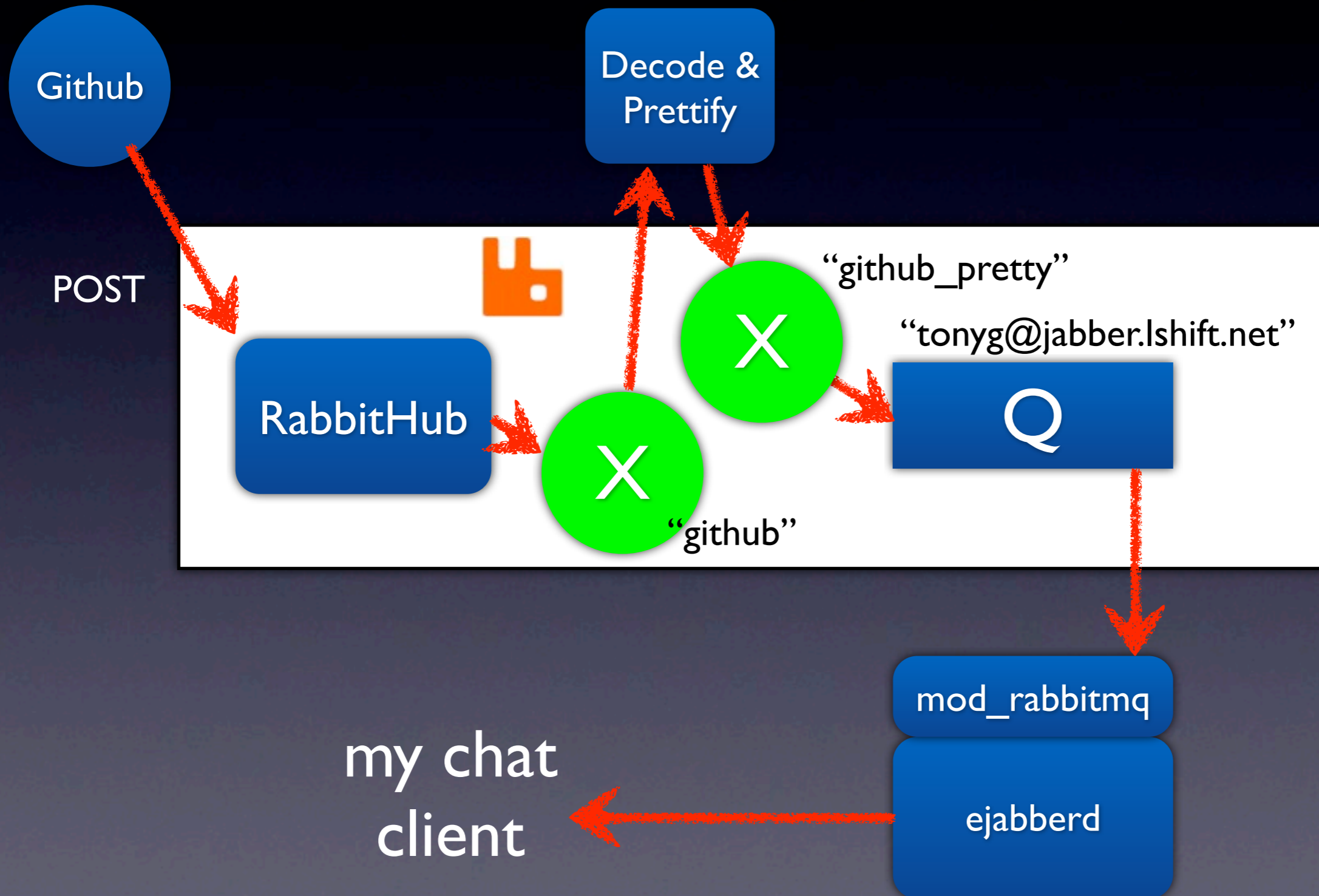
Github Commit Hooks



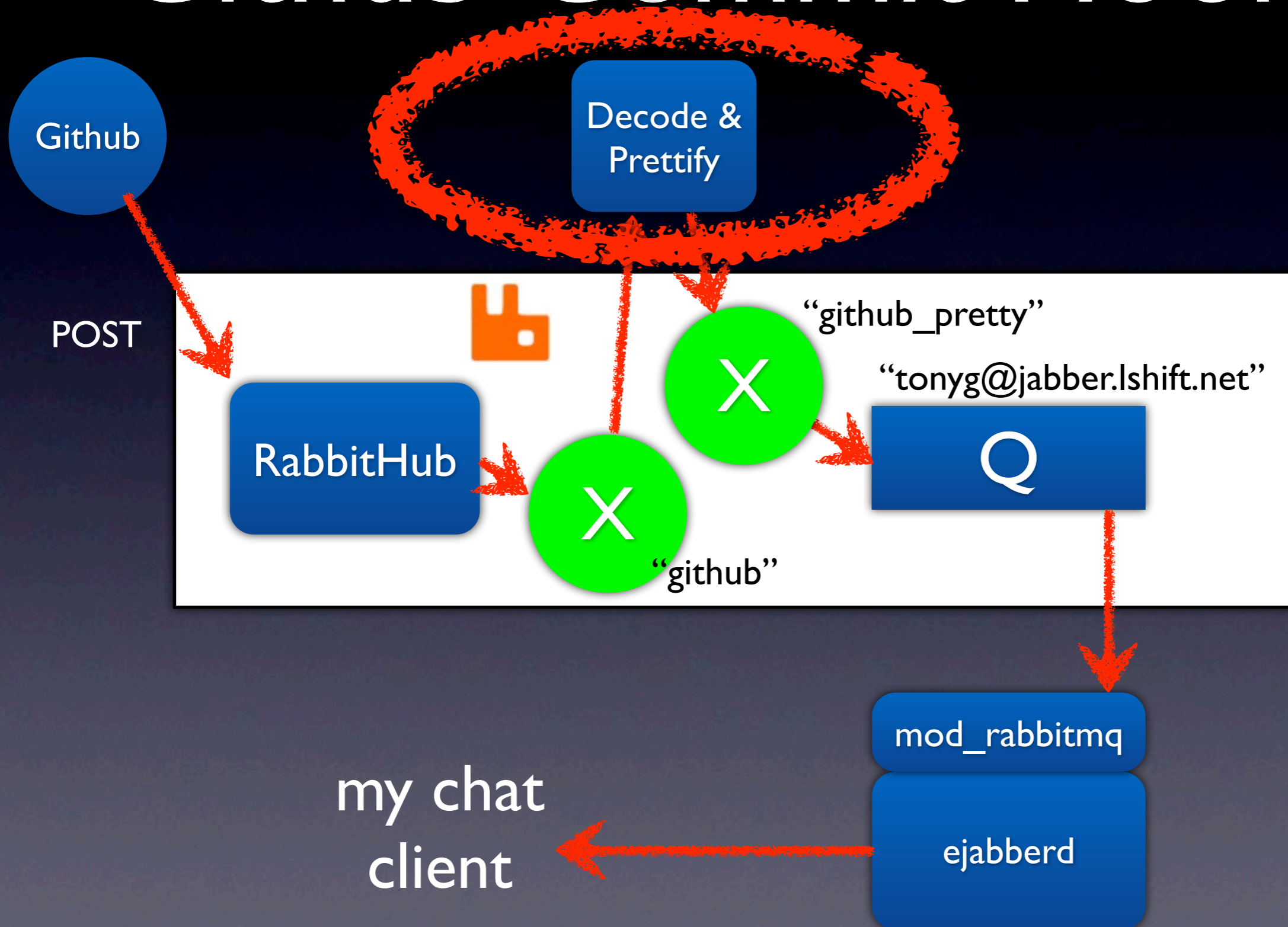
Github Commit Hooks



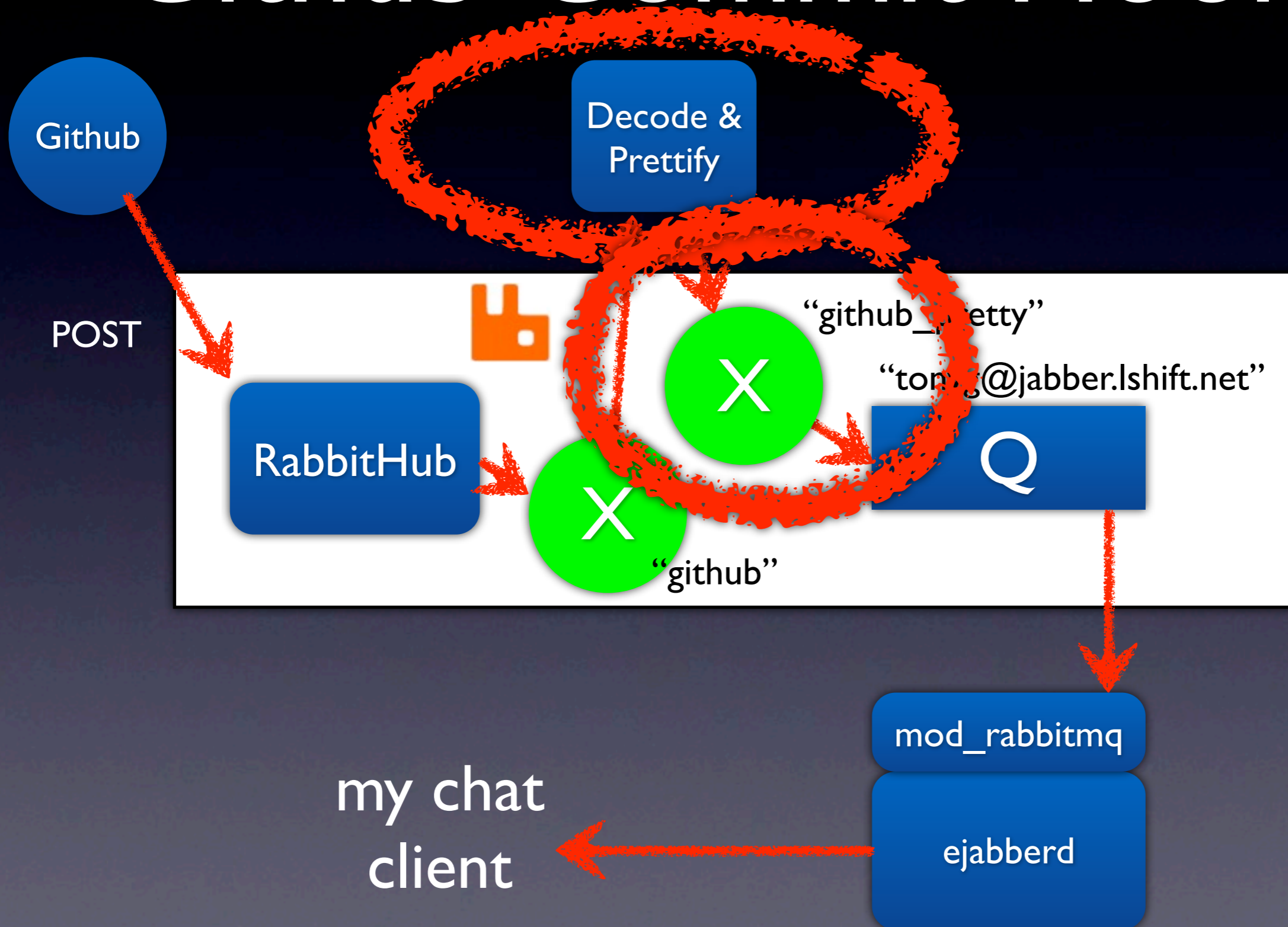
Github Commit Hooks



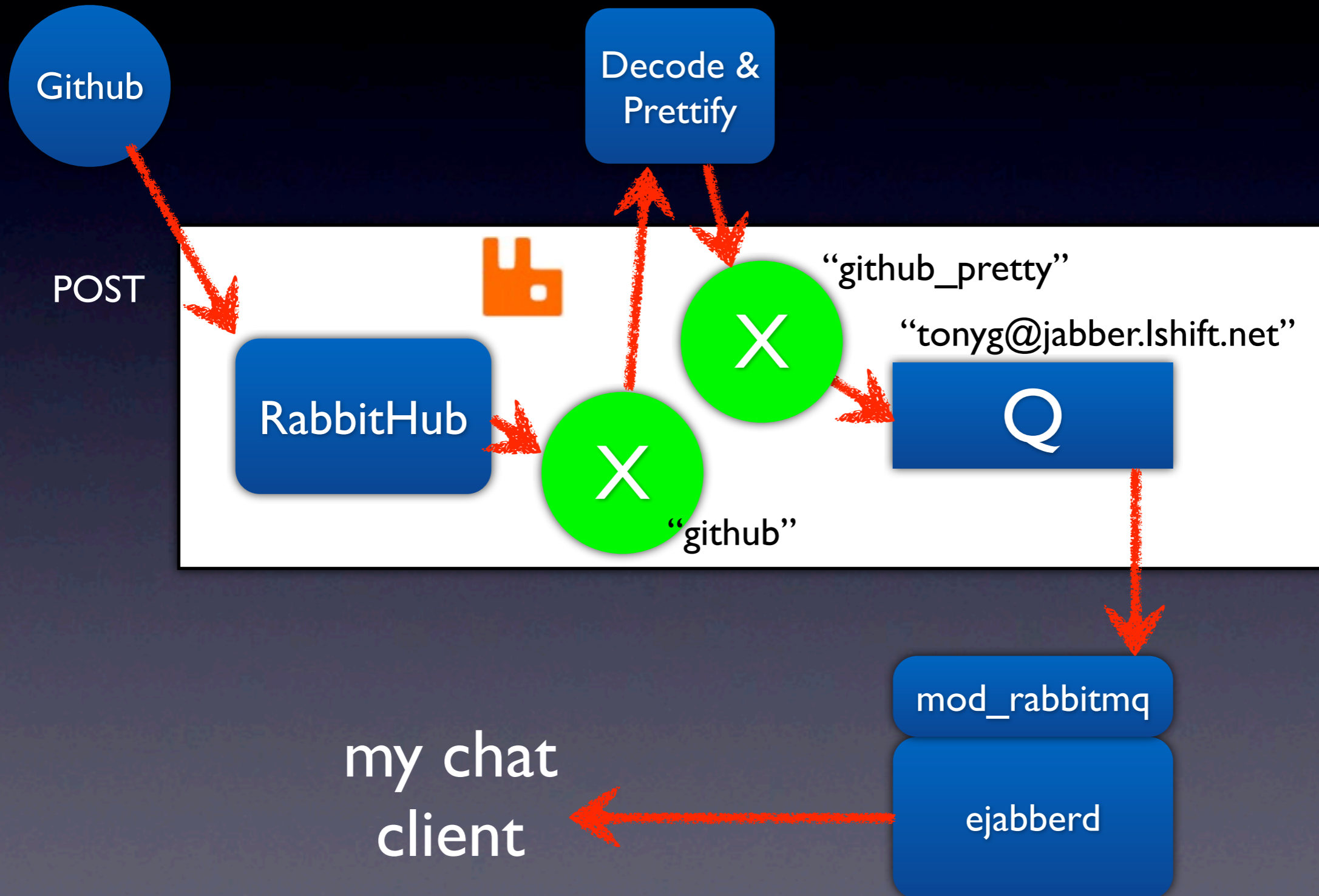
Github Commit Hooks



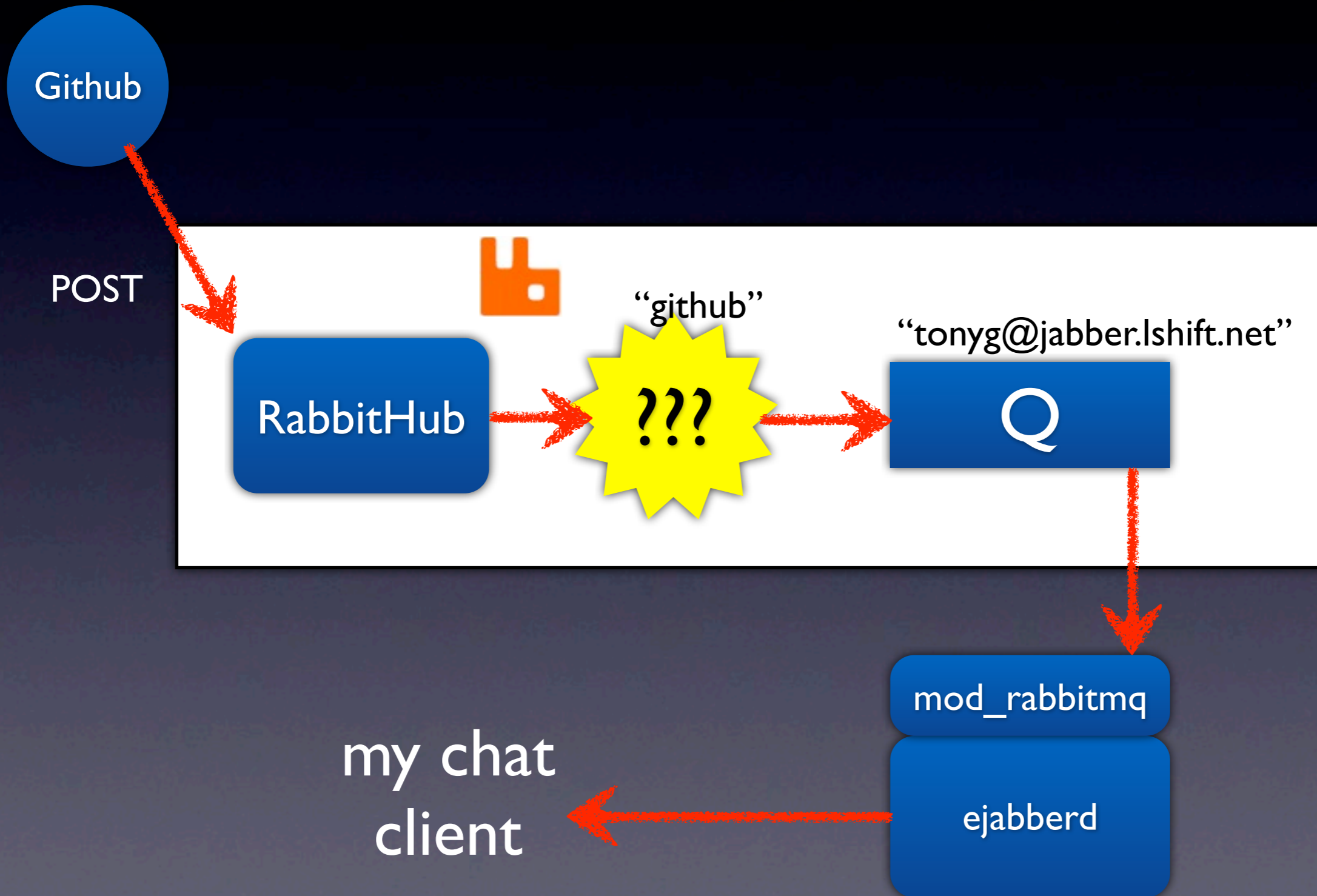
Github Commit Hooks



Github Commit Hooks



Github Commit Hooks



'x-script' Exchange Type

- github.com/tonyg/script-exchange
- Embeds Spidermonkey and Python
- Upload filter & transformation *scripts* to the server (like stored procedures!)
- It's a prototype/demo, for now

'x-script' Exchange Type

```
ch.exchange_declare(exchange='github',
                    type='x-script', arguments={
    "type": "text/javascript",
    "definition": r"""
        return function (msg) {
            if (msg.body.substring(0, 8) == "payload=") {
                // It's a github hook! Unescape and
                // pretty-print.
                var b = msg.body.substring(8);
                b = JSON.parse(unescape(b));
                msg.body = JSON.stringify(b, null, 2);
            }
            msg.fanout();
        }
    """ })
```

'x-script' Exchange Type

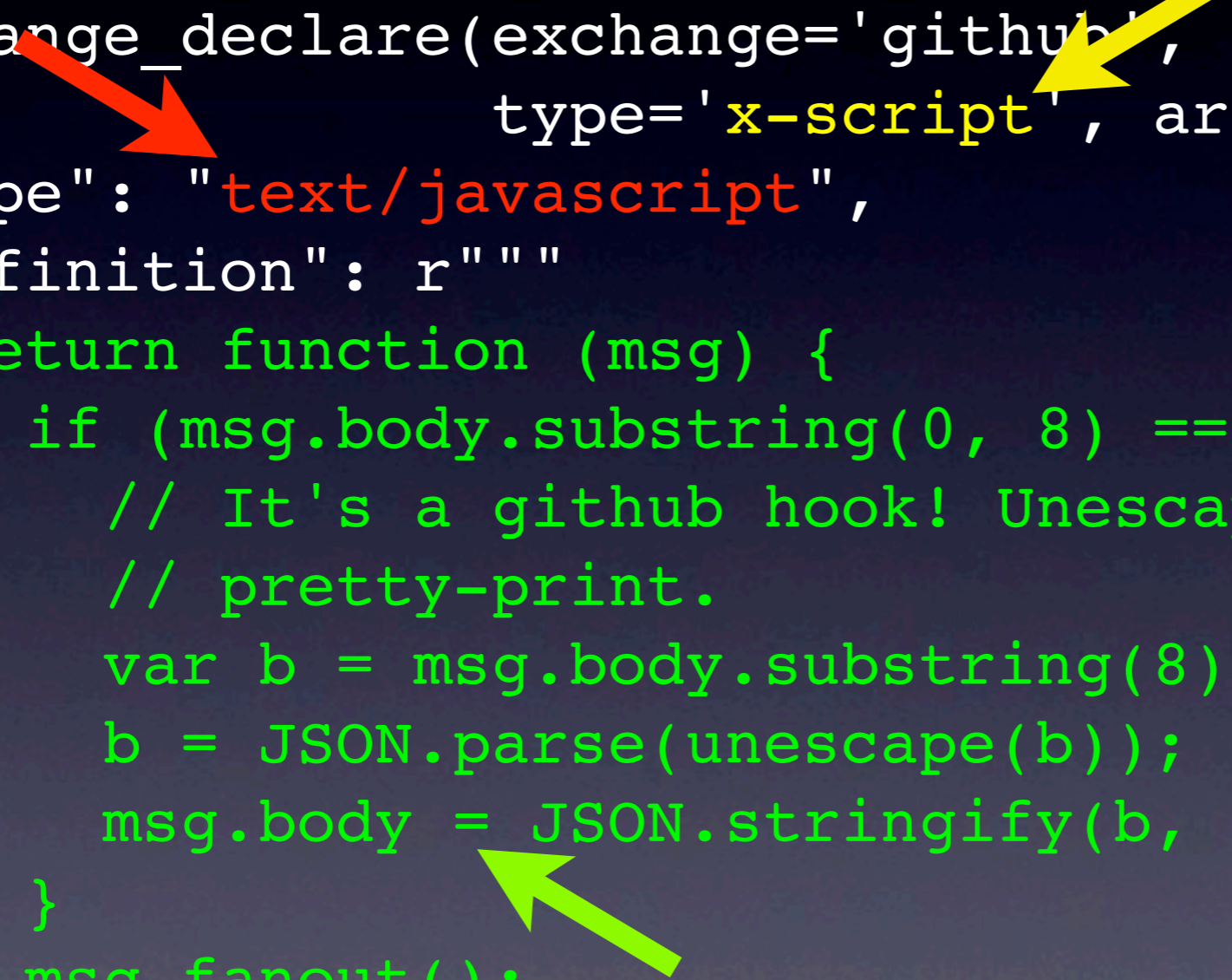
```
ch.exchange_declare(exchange='github',  
                    type='x-script', arguments={  
    "type": "text/javascript",  
    "definition": r"""  
        return function (msg) {  
            if (msg.body.substring(0, 8) == "payload=") {  
                // It's a github hook! Unescape and  
                // pretty-print.  
                var b = msg.body.substring(8);  
                b = JSON.parse(unescape(b));  
                msg.body = JSON.stringify(b, null, 2);  
            }  
            msg.fanout();  
        }  
    """ } )
```

'x-script' Exchange Type

```
ch.exchange_declare(exchange='github',  
                    type='x-script', arguments={  
  "type": "text/javascript",  
  "definition": r"""  
    return function (msg) {  
      if (msg.body.substring(0, 8) == "payload=") {  
        // It's a github hook! Unescape and  
        // pretty-print.  
        var b = msg.body.substring(8);  
        b = JSON.parse(unescape(b));  
        msg.body = JSON.stringify(b, null, 2);  
      }  
      msg.fanout();  
    }  
  """})
```

'x-script' Exchange Type

```
ch.exchange_declare(exchange='github',  
                    type='x-script', arguments={  
  "type": "text/javascript",  
  "definition": r"""  
    return function (msg) {  
      if (msg.body.substring(0, 8) == "payload=") {  
        // It's a github hook! Unescape and  
        // pretty-print.  
        var b = msg.body.substring(8);  
        b = JSON.parse(unescape(b));  
        msg.body = JSON.stringify(b, null, 2);  
      }  
      msg.fanout();  
    }  
  """})
```



'x-script' Exchange Type

```
ch.exchange_declare(exchange='github',  
                    type='x-script', arguments={  
    "type": "text/javascript",  
    "definition": r"""  
        return function (msg) {  
            if (msg.body.substring(0, 8) == "payload=") {  
                // It's a github hook! Unescape and  
                // pretty-print.  
                var b = msg.body.substring(8);  
                b = JSON.parse(unescape(b));  
                msg.body = JSON.stringify(b, null, 2);  
            }  
            msg.fanout();  
        }  
    """)})
```

Transforms this...

```
payload=%7b%22repository%22%3a%7b%22watchers%22%3a1%2c%22description%22%3a%22RabbitMQ%20%5c%22Script%20Exchange%5c%22%20plugin%22%2c%22open_issues%22%3a0%2c%22fork%22%3afalse%2c%22forks%22%3a0%2c%22url%22%3a%22http%3a%2f%2fgithub.com%2ftonyg%2fscript-exchange%22%2c%22private%22%3afalse%2c%22homepage%22%3a%22%22%2c%22owner%22%3a%7b%22email%22%3a%22tonygarnockjones%40gmail.com%22%2c%22name%22%3a%22tonyg%22%7d%2c%22name%22%3a%22script-exchange%22%7d%2c%22before%22%3a%220ed4a60180857f1b00afcc93fff6b42a7d27b39e%22%2c%22ref%22%3a%22refs%2fheads%2fmaster%22%2c%22commits%22%3a%5b%7b%22message%22%3a%22Make%20JS%20scripts%20interpreted%20as%20function%20bodies%20with%20their%20own%20scope.%22%2c%22removed%22%3a%5b%5d%2c%22url%22%3a%22http%3a%2f%2fgithub.com%2ftonyg%2fscript-exchange%2fcommit%2f095931d3f3c01284423999349a55164ab6a964d3%22%2c%22modified%22%3a%5b%22examples%2fexample_js.py%22%2c%22priv%2fjs_exchange_boot.js%22%5d%2c%22author%22%3a%7b%22email%22%3a%22tonygarnockjones%40gmail.com%22%2c%22name%22%3a%22Tony%20Garnock-Jones%22%7d%2c%22timestamp%22%3a%222010-03-07T01%3a45%3a48-08%3a00%22%2c%22added%22%3a%5b%5d%2c%22id%22%3a%22095931d3f3c01284423999349a55164ab6a964d3%22%7d%2c%7b%22message%22%3a%22Documentation.%22%2c%22removed%22%3a%5b%5d%2c%22url%22%3a%22http%3a%2f%2fgithub.com%2ftonyg%2fscript-exchange%2fcommit%2f7606a22d66169044c1352d956c63b362e783aca5%22%2c%22modified%22%3a%5b%22README.md%22%5d%2c%22author%22%3a%7b%22email%22%3a%22tonygarnockjones%40gmail.com%22%2c%22name%22%3a%22Tony%20Garnock-Jones%22%7d%2c%22timestamp%22%3a%222010-03-07T01%3a45%3a56-08%3a00%22%2c%22added%22%3a%5b%5d%2c%22id%22%3a%227606a22d66169044c1352d956c63b362e783aca5%22%7d%5d%2c%22after%22%3a%227606a22d66169044c1352d956c63b362e783aca5%22%7d
```

...to this

```
{
  "repository":{
    "watchers":1,
    "description":"RabbitMQ \"Script Exchange\" plugin",
    "open_issues":0,
    "fork":false,
    "forks":0,
    "url":"http://github.com/tonyg/script-exchange",
    "private":false,
    "homepage":"","
    "owner":{
      "email":"tonygarnockjones@gmail.com",
      "name":"tonyg"
    },
    "name":"script-exchange"
  },
  "before":"0ed4a60180857f1b00afcc93fff6b42a7d27b39e", ...
}
```

What can it do?

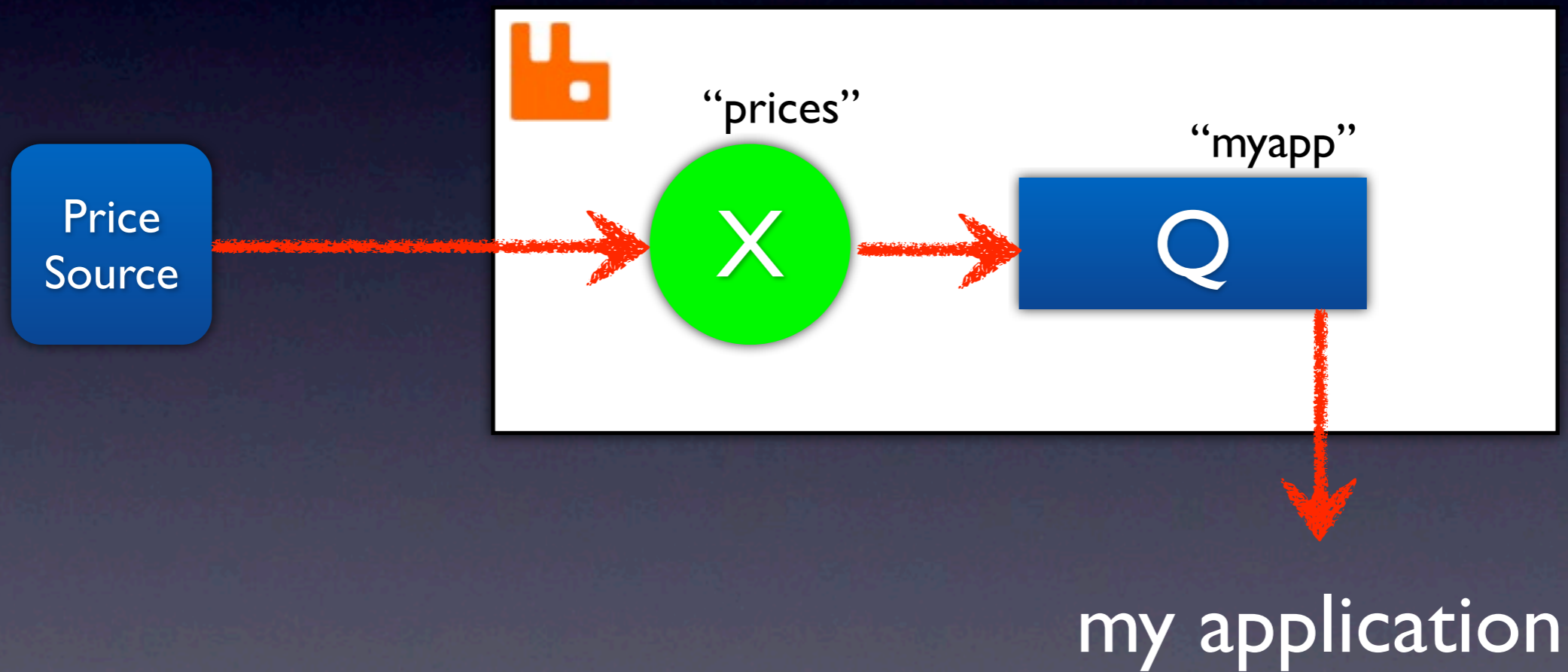
- Rewrite routing key, properties, headers and body as desired
- Any or none of fanout-, direct-, and topic-style routing for each message
- No custom binding filter languages yet
- Security implications! (Salmon signatures)
- Be warned: it's just a prototype for now

Example 2.

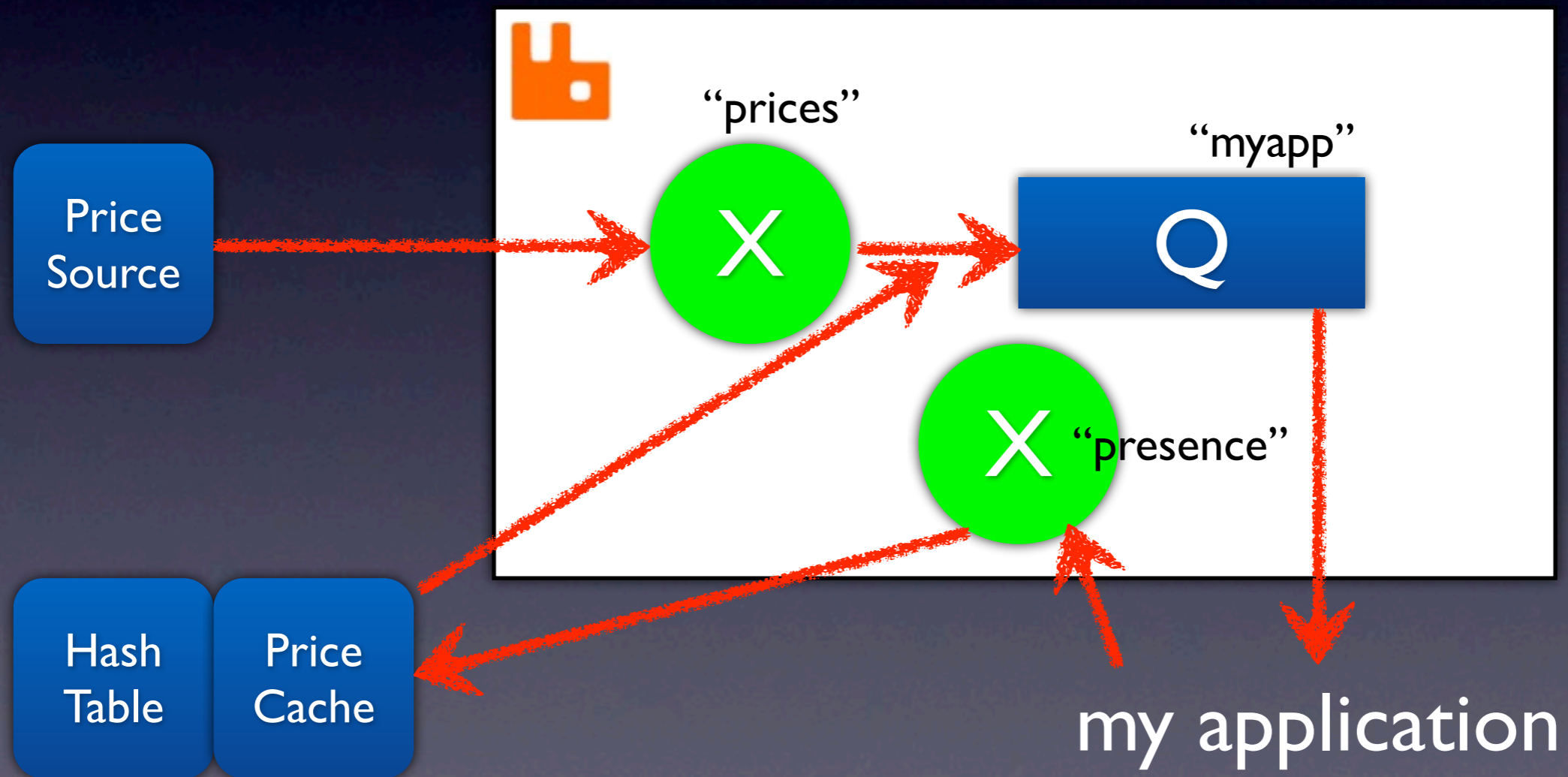
Stock Ticker

- Classic “Last Value Cache” pubsub
- New prices are arriving all the time
- When I connect, I want the *newest* price so far, without delay
- When the price updates, I want to hear it

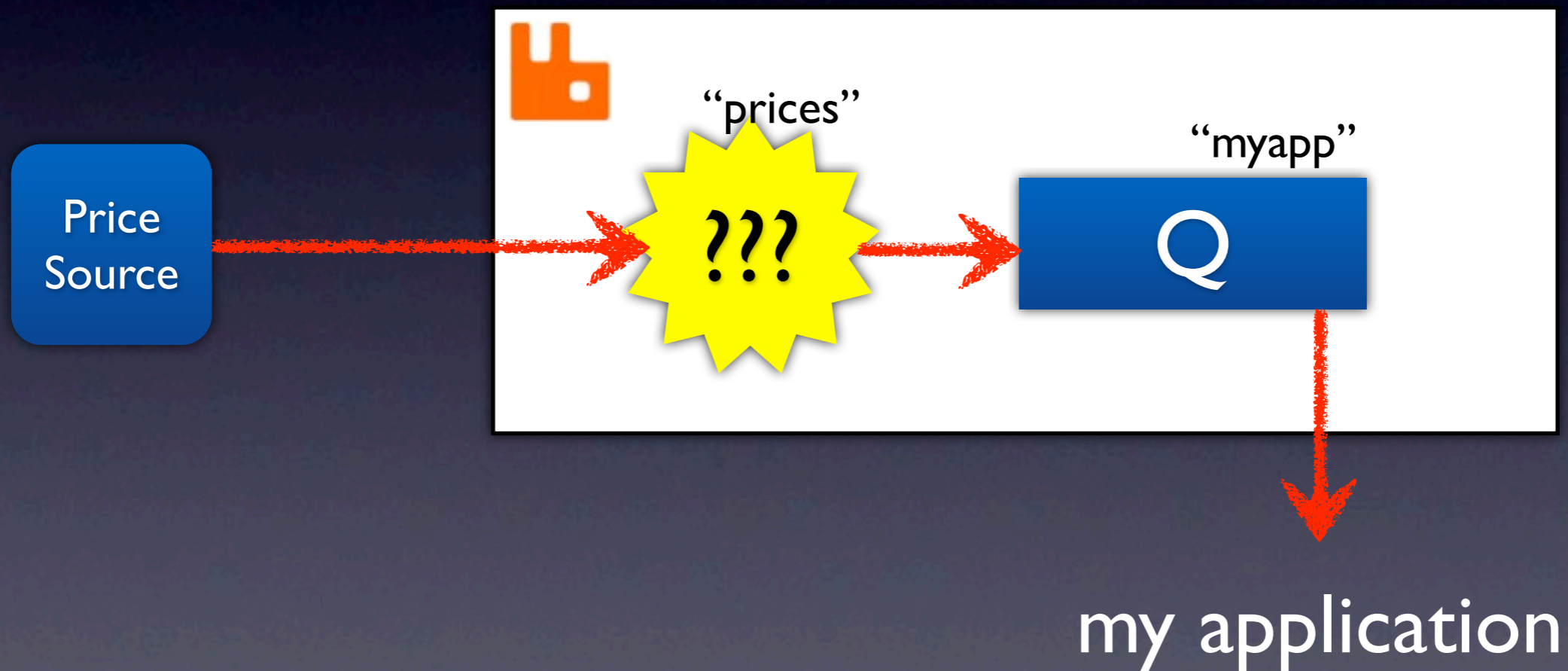
Stock Ticker



Stock Ticker



Stock Ticker



'x-lvc' Exchange Type

- github.com/squaremo/rabbitmq-lvc-plugin
- A Last Value Cache exchange type plugin
- Declared just like built-in exchange types, using `Exchange.Declare`
- Uses `mnesia` to hold the cache
- It's a prototype/demo, for now

'x-lvc' Exchange Type

```
import amqpplib.client_0_8 as amqp
ch = amqp.Connection().channel()
ch.exchange_declare("lvc", type="x-lvc")
ch.basic_publish(amqp.Message("value"),
                exchange="lvc",
                routing_key="rabbit")
...
ch.queue_declare("q")
ch.queue_bind("q", "lvc", "rabbit")
print ch.basic_get("q").body
```

Writing Plugins and Exchange Types

It's easy

- Write an Erlang OTP application, including its .app file
- Package it up as a .ez file
- Consider using `rabbitmq-public-umbrella`

.ez files

```
WALK XTERM
Mar 24 22:09:35 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$ unzip -v rabbit_lvc_plugin.ez
Archive:  rabbit_lvc_plugin.ez
 Length  Method      Size  Ratio   Date    Time    CRC-32   Name
-----  -
    0     Stored      0     0%    01-22-10 04:04  00000000  rabbit_lvc_plugin/
    0     Stored      0     0%    01-22-10 04:04  00000000  rabbit_lvc_plugin/ebin/
 4596   Defl:N      3878  16%    01-22-10 04:04  b6c5a234  rabbit_lvc_plugin/ebin/rabbit_exchange_type_lvc.beam
   271   Defl:N      161   41%    01-22-10 04:04  38d95fc2  rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.app
 1872   Defl:N      1424  24%    01-22-10 04:04  ce42bfcc  rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.beam
-----  -
 6739                5463  19%                5 files

Mar 24 22:09:49 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$ █
```

.ez files

```
WALK XTERM
Mar 24 22:09:35 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$ unzip -v rabbit_lvc_plugin.ez
Archive:  rabbit_lvc_plugin.ez
Length  Method      Size  Ratio   Date    Time    CRC-32   Name
-----  -
  0     Stored        0     0%    01-22-10 04:04  00000000  rabbit_lvc_plugin/
  0     Stored        0     0%    01-22-10 04:04  00000000  rabbit_lvc_plugin/ebin/
 4596   Defl:N       3878  16%    01-22-10 04:04  b6c5a234  rabbit_lvc_plugin/ebin/rabbit_exchange_type_lvc.beam
  271   Defl:N        161  41%    01-22-10 04:04  38d95fc2  rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.app
 1872   Defl:N       1424  24%    01-22-10 04:04  ce42bfcc  rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.beam
-----
 6739                5463  19%
                                     5 files


Mar 24 22:09:49 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$
```



.ez files

```
WALK XTERM
Mar 24 22:09:35 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$ unzip -v rabbit_lvc_plugin.ez
Archive:  rabbit_lvc_plugin.ez
Length  Method  Size  Ratio  Date   Time   CRC-32  Name
-----  -
 0     Stored    0     0%    01-22-10 04:04 00000000 rabbit_lvc_plugin/
 0     Stored    0     0%    01-22-10 04:04 00000000 rabbit_lvc_plugin/ebin/
4596   Defl:N   3878  16%    01-22-10 04:04 b6c5a234 rabbit_lvc_plugin/ebin/rabbit_exchange_type_lvc.beam
 271   Defl:N   161   41%    01-22-10 04:04 38d95fc2 rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.app
1872   Defl:N   1424  24%    01-22-10 04:04 ce42bfcc rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.beam
-----
6739           5463  19%
5 files

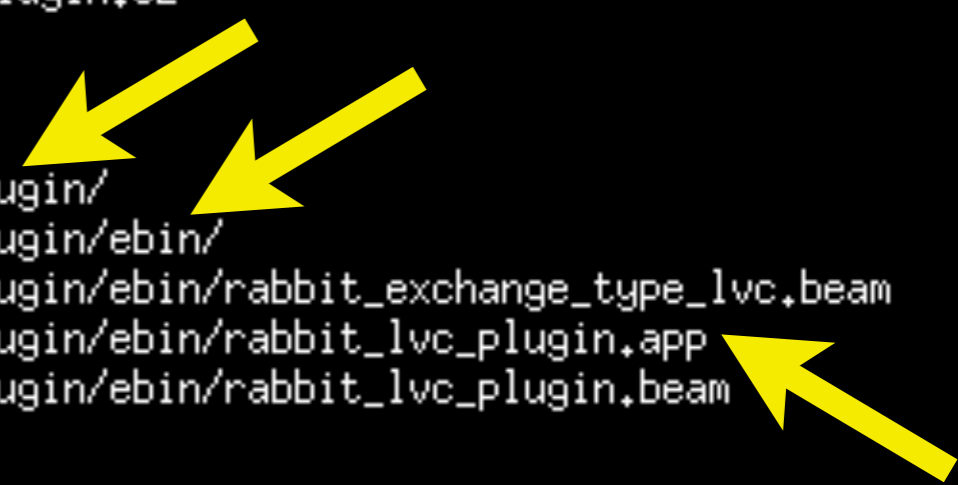
Mar 24 22:09:49 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$
```



.ez files

```
WALK XTERM
Mar 24 22:09:35 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$ unzip -v rabbit_lvc_plugin.ez
Archive:  rabbit_lvc_plugin.ez
Length  Method      Size  Ratio   Date    Time    CRC-32  Name
-----  -
 0      Stored      0     0%    01-22-10 04:04 00000000 rabbit_lvc_plugin/
 0      Stored      0     0%    01-22-10 04:04 00000000 rabbit_lvc_plugin/ebin/
4596    Defl:N      3878  16%    01-22-10 04:04 b6c5a234 rabbit_lvc_plugin/ebin/rabbit_exchange_type_lvc.beam
 271    Defl:N      161   41%    01-22-10 04:04 38d95fc2 rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.app
1872    Defl:N      1424  24%    01-22-10 04:04 ce42bfcc rabbit_lvc_plugin/ebin/rabbit_lvc_plugin.beam
-----
6739    5463  19%
5 files

Mar 24 22:09:49 tonyg@walk
~/dev/rabbitmq-umbrella/rabbitmq-lvc-plugin/dist$
```



The .app file

```
{application, rabbit_lvc_plugin,  
  [{description, "RabbitMQ LVC exchange"},  
   {vsn, "0.01"},  
   {modules, [  
     rabbit_lvc_plugin,  
     rabbit_exchange_type_lvc  
   ]},  
   {registered, []},  
   {env, []},  
   {applications,  
     [kernel, stdlib, rabbit, mnesia]}}].
```

Writing Exchange Types

- Write an ordinary plugin
- Add a boot step that registers the new type
- Implement the `rabbit_exchange_type` behaviour

Boot Steps

- For starting up *with* RabbitMQ, not *after*
- Uses Erlang's *module attributes*
- Uses Erlang's built-in support for reasoning about graphs
- Topological sort of dependencies gives order-of-operations

Boot Steps

```
-rabbit_boot_step({my_boot_step_name,  
  [{description, "do something cool"},  
   {mfa, {mymodule, myfunc1, [arg0, arg1]}}},  
  {mfa, {mymodule, myfunc2, [arg]}}},  
  {enables, some_feature},  
  {enables, another_step},  
  {requires, something_to_be_ready_first}}}).
```

Boot Steps

```
%% Use topological sort to find a consistent ordering (if
%% there is one, otherwise fail).
```

```
SortedStepsRev =
  [begin
    {StepName, Step} = digraph:vertex(G, StepName),
    Step
  end || StepName <- digraph_utils:topsort(G)],
```

```
SortedSteps = lists:reverse(SortedStepsRev).
```

Registering 'x-lvc'

```
-rabbit_boot_step({?MODULE,  
  [{description, "last-value cache exchange type"},  
   {mfa, {rabbit_lvc_plugin, setup_schema, []}},  
   {mfa, {rabbit_exchange_type_registry, register,  
         [<<"x-lvc">>, rabbit_exchange_type_lvc]}}},  
  {requires, rabbit_exchange_type_registry},  
  {enables, exchange_recovery}]})
```

AMQP 8-0

Copyright (C) 2007-2010 LShift Ltd., Cohesive Financial Technologies LLC., and Rabbit Technologies Ltd.

Licensed under the MPL. See <http://www.rabbitmq.com/>

```
node           : rabbit@walk
app descriptor: /Users/tonyg/dev/rabbitmq-umbrella/rabbitmq-server/scripts/../ebin/rabbit.app
home dir       : /Users/tonyg
cookie hash    : mwtySYvzRGJyIxBy7NFuLA==
log            : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit.log
sasl log       : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit-sasl.log
database dir   : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbitmq-rabbit-mnesia
```

```
starting worker pool           ...done
starting database              ...done
-- external infrastructure ready
starting exchange type registry ...done
starting last-value cache exchange type ...done
starting exchange type topic   ...done
starting exchange type headers ...done
starting exchange type fanout  ...done
starting exchange type direct  ...done
starting internal event notification system ...done
starting logging server        ...done
-- kernel ready
starting alarm handler         ...done
starting queue supervisor      ...done
starting node monitor          ...done
starting cluster router        ...done
-- core initialized
starting empty DB check        ...done
starting codec correctness check ...done
starting script_manager_sup     ...done
starting script_exchange       ...done
starting exchange recovery     ...done
starting queue recovery        ...done
starting persister             ...done
starting guid generator         ...done
-- message delivery logic ready
starting error log relay       ...done
starting networking            ...done
starting RabbitHub             ...done
-- network listeners available

broker running
```


AMQP 8-0

Copyright (C) 2007-2010 LShift Ltd., Cohesive Financial Technologies LLC., and Rabbit Technologies Ltd.

Licensed under the MPL. See <http://www.rabbitmq.com/>

```
node           : rabbit@walk
app descriptor: /Users/tonyg/dev/rabbitmq-umbrella/rabbitmq-server/scripts/../ebin/rabbit.app
home dir       : /Users/tonyg
cookie hash    : mwtySYvzRGJyIxBy7NFuLA==
log            : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit.log
sasl log       : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit-sasl.log
database dir   : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbitmq-rabbit-mnesia
```

```
starting worker pool           ...done
starting database
-- external infrastructure ready
starting exchange type registry
starting last-value cache exchange type
starting exchange type topic   ...done
starting exchange type headers ...done
starting exchange type fanout  ...done
starting exchange type direct  ...done
starting internal event notification system ...done
starting logging server        ...done
-- kernel ready
starting alarm handler         ...done
starting queue supervisor      ...done
starting node monitor          ...done
starting cluster router        ...done
-- core initialized
starting empty DB check        ...done
starting codec correctness check ...done
starting script_manager_sup     ...done
starting script_exchange       ...done
starting exchange recovery     ...done
starting queue recovery        ...done
starting persister             ...done
starting guid generator         ...done
-- message delivery logic ready
starting error log relay        ...done
starting networking            ...done
starting RabbitHub             ...done
-- network listeners available

broker running
```

starting last-value cache exchange type

AMQP 8-0

Copyright (C) 2007-2010 LShift Ltd., Cohesive Financial Technologies LLC., and Rabbit Technologies Ltd.

Licensed under the MPL. See <http://www.rabbitmq.com/>

```
node           : rabbit@walk
app descriptor: /Users/tonyg/dev/rabbitmq-umbrella/rabbitmq-server/scripts/../ebin/rabbit.app
home dir       : /Users/tonyg
cookie hash    : mwtySYvzRGJyIxBy7NFuLA==
log            : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit.log
sasl log       : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit-sasl.log
               : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbitmq-rabbit-mnesia
```

starting exchange type registry

```
...done
starting database
-- external infrastructure ready
starting exchange type registry
starting last-value cache exchange type
starting exchange type topic           ...done
starting exchange type headers         ...done
starting exchange type fanout          ...done
starting exchange type direct          ...done
starting internal event notification system ...done
starting logging server                 ...done
-- kernel ready
starting alarm handler                  ...done
starting queue supervisor               ...done
starting node monitor                   ...done
starting cluster router                 ...done
-- core initialized
starting empty DB check                 ...done
starting codec correctness check        ...done
starting script_manager_sup             ...done
starting script_exchange                ...done
starting exchange recovery              ...done
starting queue recovery                 ...done
starting persister                      ...done
starting guid generator                 ...done
-- message delivery logic ready
starting error log relay                ...done
starting networking                     ...done
starting RabbitHub                      ...done
-- network listeners available

broker running
```

starting last-value cache exchange type

AMQP 8-0

Copyright (C) 2007-2010 LShift Ltd., Cohesive Financial Technologies LLC., and Rabbit Technologies Ltd.

Licensed under the MPL. See <http://www.rabbitmq.com/>

```
node           : rabbit@walk
app descriptor: /Users/tonyg/dev/rabbitmq-umbrella/rabbitmq-server/scripts/../ebin/rabbit.app
home dir       : /Users/tonyg
cookie hash    : mwtySYvzRGJyIxBy7NFuLA==
log            : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit.log
sasl log       : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbit-sasl.log
               : /var/folders/YT/YTZtM5Y9GOG6zMVaktTkLE+++TI/-Tmp-//rabbitmq-rabbit-mnesia
```

starting exchange type registry

```
...done
starting database
-- external infrastructure ready
starting exchange type registry
starting last-value cache exchange type
starting exchange type topic           ...done
starting exchange type headers         ...done
starting exchange type fanout          ...done
starting exchange type direct          ...done
starting internal event notification system ...done
starting logging server                 ...done
-- kernel ready
starting alarm handler                   ...done
starting queue supervisor                ...done
starting node monitor                    ...done
starting cluster router                  ...done
-- core initial
starting empty                            ...done
starting codec                            ...done
starting script_management_sup           ...done
starting script_exchange                 ...done
starting exchange recovery               ...done
starting queue recovery                  ...done
starting persister                       ...done
starting guid generator                  ...done
-- message delivery logic ready
starting error log relay                  ...done
starting networking                       ...done
starting RabbitHub                       ...done
-- network listeners available

broker running
```

starting last-value cache exchange type

starting exchange recovery

rabbit_exchange_type

`description/0 :: () -> [{atom(), any()}]`

`publish/2 :: (exchange(), delivery()) -> {routing_result(), [pid()]}`

`validate/1 :: (exchange()) -> 'ok'`

`create/1 :: (exchange()) -> 'ok'`

`recover/2 :: (exchange(), list(binding())) -> 'ok'`

`delete/2 :: (exchange(), list(binding())) -> 'ok'`

`add_binding/2 :: (exchange(), binding()) -> 'ok'`

`remove_bindings/2 :: (exchange(), list(binding())) -> 'ok'`

'x-lvc' Implementation

- `publish` updates a row in mnesia before routing like a direct exchange
- `add_binding` queries mnesia and sends on the cached value to the newly-bound queue
- `delete` removes rows from mnesia

(almost done)

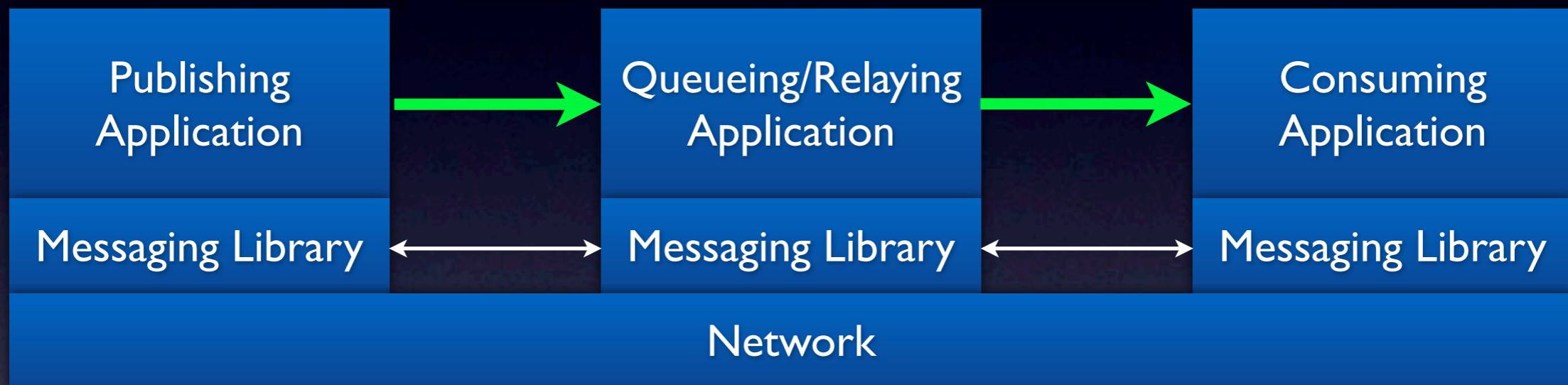
Traditional View



Traditional View



Revised View



Revised View



Your Plugin Here?

- ...for new transports (e.g. RabbitHub)
- ...for new exchange types (lvc, script)
- Plugins are straightforward to write (esp. with umbrella)
- We're looking forward to seeing what you all come up with!

www.rabbitmq.com/how
github.com/tonyg/rabbithub
github.com/tonyg/script-exchange
github.com/squaremo/rabbitmq-lvc-plugin

Thanks!