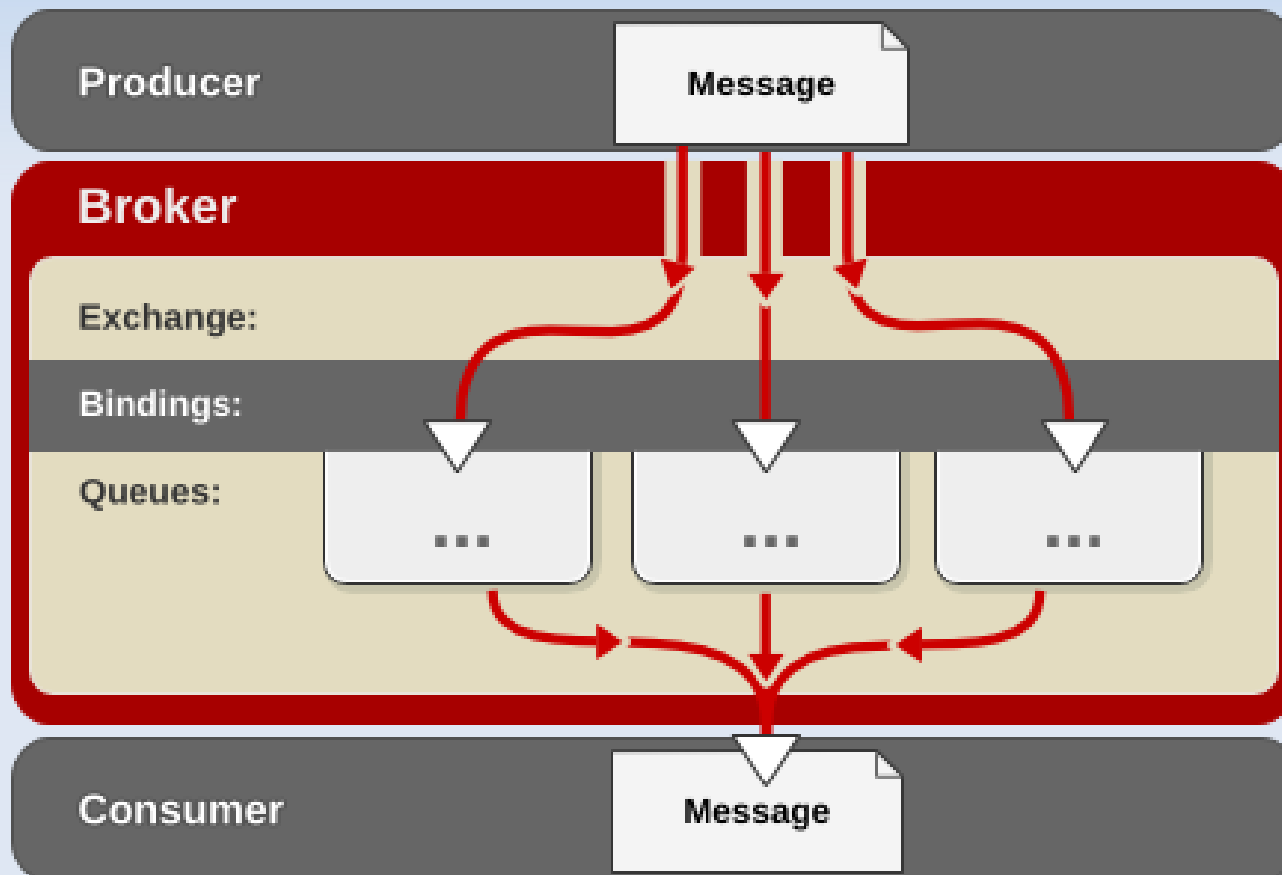


Message Queues in Python

John Mitchell johnlmitchell@gmail.com

Producer Consumer



About Me

- Using Python since 1995
- C/C++, Java, PHP, Perl, Tcl, CLIPS, Scheme, bash, awk
- servers, graphics, pipelines, Web
- <http://www.linkedin.com/in/johnlmitchell>

Why MQ

- loosely-coupled
- scalable
- multiple news publishers, single receiver
- decentralized service
- durable single app
- actor model

AMQP: Advanced Message Queuing Protocol

- Interoperability
 - wire level, not just API (vs: JMS)
- financial industry
 - mostly asynchronous
 - *very* fast, low latency
- RabbitMQ
 - < 1ms transient mode, 10k msgs/sec
 - persistent mode, 3-5k msgs/sec stored to disk

Comparison: gmail

- Email address ~ named exchange
- Gmail filters ~ exchange has multiple bindings
- Gmail label ~ queue
- No list of addresses/exchange names

Exchange ~ gmail address

The screenshot shows the Gmail interface for the account **johnmitchell@gmail.com**. A search for **label:e-meetup** has been performed, displaying 71 results. The interface includes a left sidebar with navigation options like **Compose Mail**, **Inbox (12)**, **Sent Mail**, **Drafts**, and **Spam (1)**. A list of labels is visible, including **e-meetup**, **event**, **health**, **j-Digisynd**, **J-Piggles (1)**, **j-Salesforce**, **job (9)**, and **sale (2)**. The main content area shows a list of search results with columns for checkboxes, stars, sender names, labels, subject lines, and times. The label **e-meetup** is highlighted in red in the first row. Action buttons at the top of the results list include **Remove label "e-meetup"**, **Report spam**, **Delete**, **Mark as read**, **Move to**, and **Labels**. A **Refresh** button is also present. The page indicates **1 - 50 of 71** results.

Select: All, None, Read, Unread, Starred, Unstarred					
<input type="checkbox"/>	<input type="checkbox"/>	Max	e-meetup	Inbox	New Meetup: Fo 11:0
<input type="checkbox"/>	<input type="checkbox"/>	New Meetup	e-meetup	Inbox	Announcing ""Ac 9:4
<input type="checkbox"/>	<input type="checkbox"/>	Max	e-meetup	Inbox	[AdventuresInSo N
<input type="checkbox"/>	<input type="checkbox"/>	David	e-meetup	Inbox	[SocialFoodieCo N
<input type="checkbox"/>	<input type="checkbox"/>	Meetup Reminder	e-meetup	Inbox	Reminder: "Get N
<input type="checkbox"/>	<input type="checkbox"/>	Max	e-meetup	Inbox	[AdventuresInSo N
<input type="checkbox"/>	<input type="checkbox"/>	Meetup Reminder	e-meetup	Inbox	Reminder: "Sant N
<input type="checkbox"/>	<input type="checkbox"/>	Meetup Reminder	e-meetup	Inbox	Reminder: "Palo N
<input type="checkbox"/>	<input type="checkbox"/>	Robert. me (2)	e-meetup	Inbox	Meetup details c N

Binding ~ gmail filter

The screenshot displays the Gmail 'Filters and Blocked Addresses' interface. On the left sidebar, there are labels for 'health', 'j-Digisynd', 'J-Piggles (1)', 'j-Salesforce', 'Job (9)', and 'sale (2)', along with a '7 more' dropdown. Below these are sections for 'Contacts' and 'Tasks'. A contact card for 'John Mitchell' is visible, with a search bar for adding or inviting contacts. A Google Calendar snippet shows events for 'Tomorrow (Fri, Nov 20)', 'Sat, Nov 21' (Tiffaney party?), and 'Sun, Nov 22'.

The main area shows a list of filters:

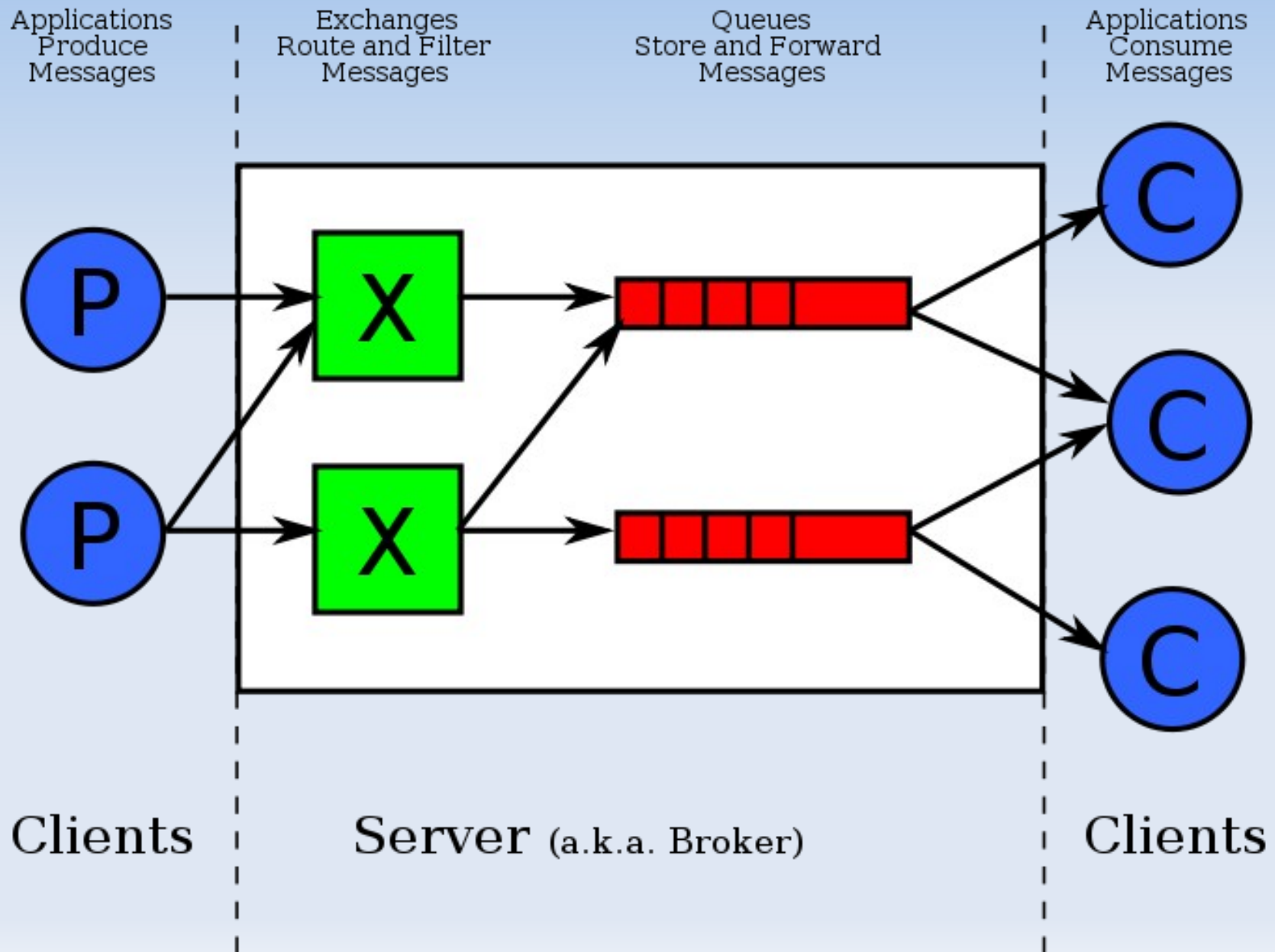
- Do this: Apply label "event"
- Matches: **from: (@jobserve.com OR @ammassociates.com OR social-piggles-request@ OR @startuply.com OR usc.edu OR @aerotek.com OR alerts@yahoo-inc.com OR linkedin.com OR dice.com OR fox.com OR careerbuilder.com OR brightfuse.com OR careerbuilder.com OR @indeed.com OR @monster.com OR kelly@lavirtual.net OR moyirv@yahoo.com)**
Do this: Apply label "job", Never send it to Spam
- Matches: to: (@meetup.com)**
Do this: Apply label "e-meetup"
- Matches: **from: (@ingboo.com)**
Do this: Apply label "job"
- Matches: **from: (@e.tripadvisor.com)**
Do this: Apply label "sale"
- Matches: **from: (noreply@sologig.com)**
Do this: Apply label "job"
- Matches: **from: (store-news@amazon.com)**
Do this: Apply label "sale"

A red rectangular box highlights the filter rule for '@meetup.com'.

Terminology

- broker=server
- connection: socket with authentication
- channel: virtual, per-thread
- **exchange**: router recv msgs. Named, typed
 - passive, durable, auto-delete
- **queue**: holds msgs. Named, props, no types
- message: unnamed. Header, opaque body

Model

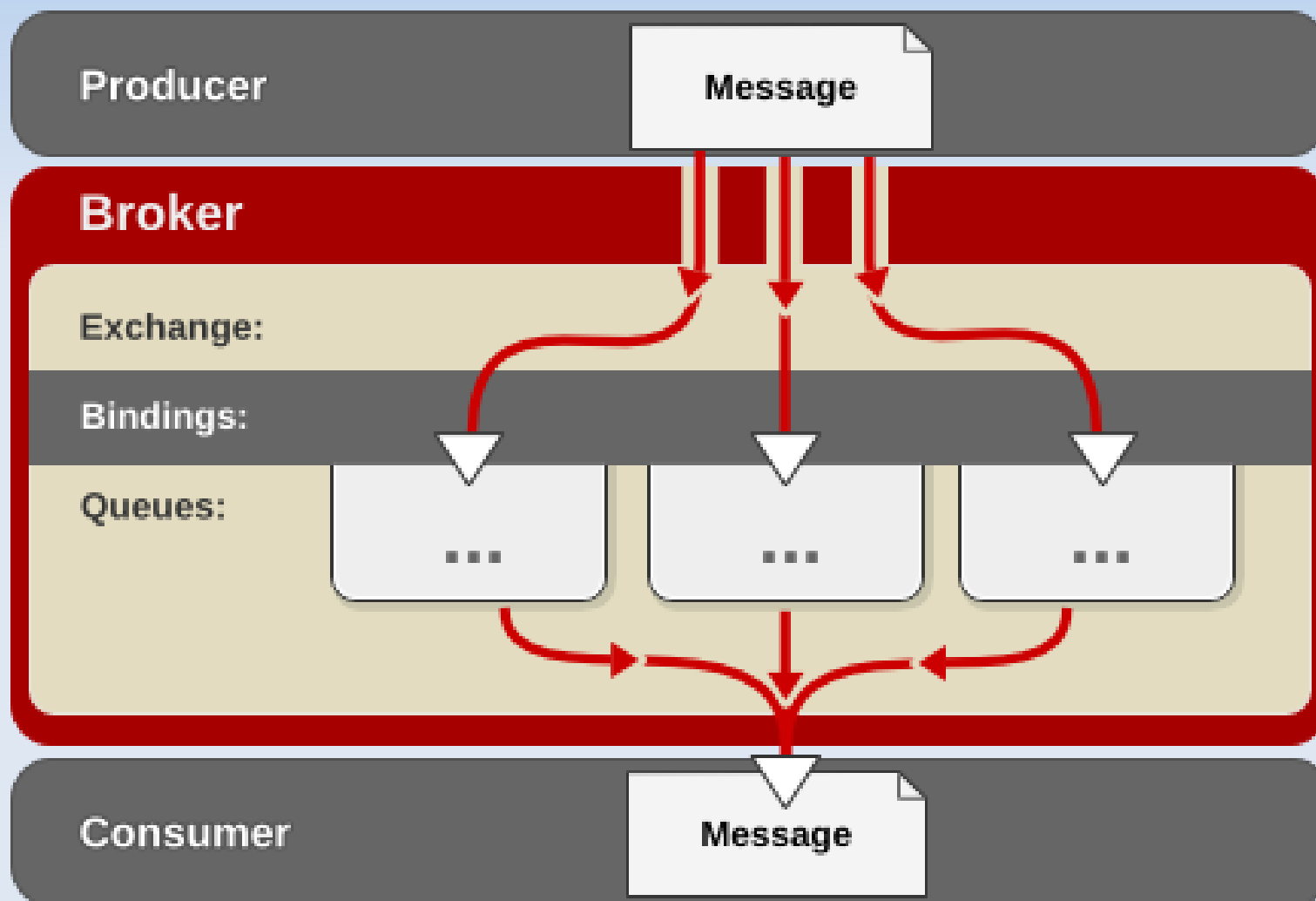


Routing

- binding: exchange-queue mapping
- Fanout
 - message → exchange → queue(s)
- Direct
 - key, message → exchange → queue(s)
- Topic
 - key.subkey, message → exchange → queue(s)

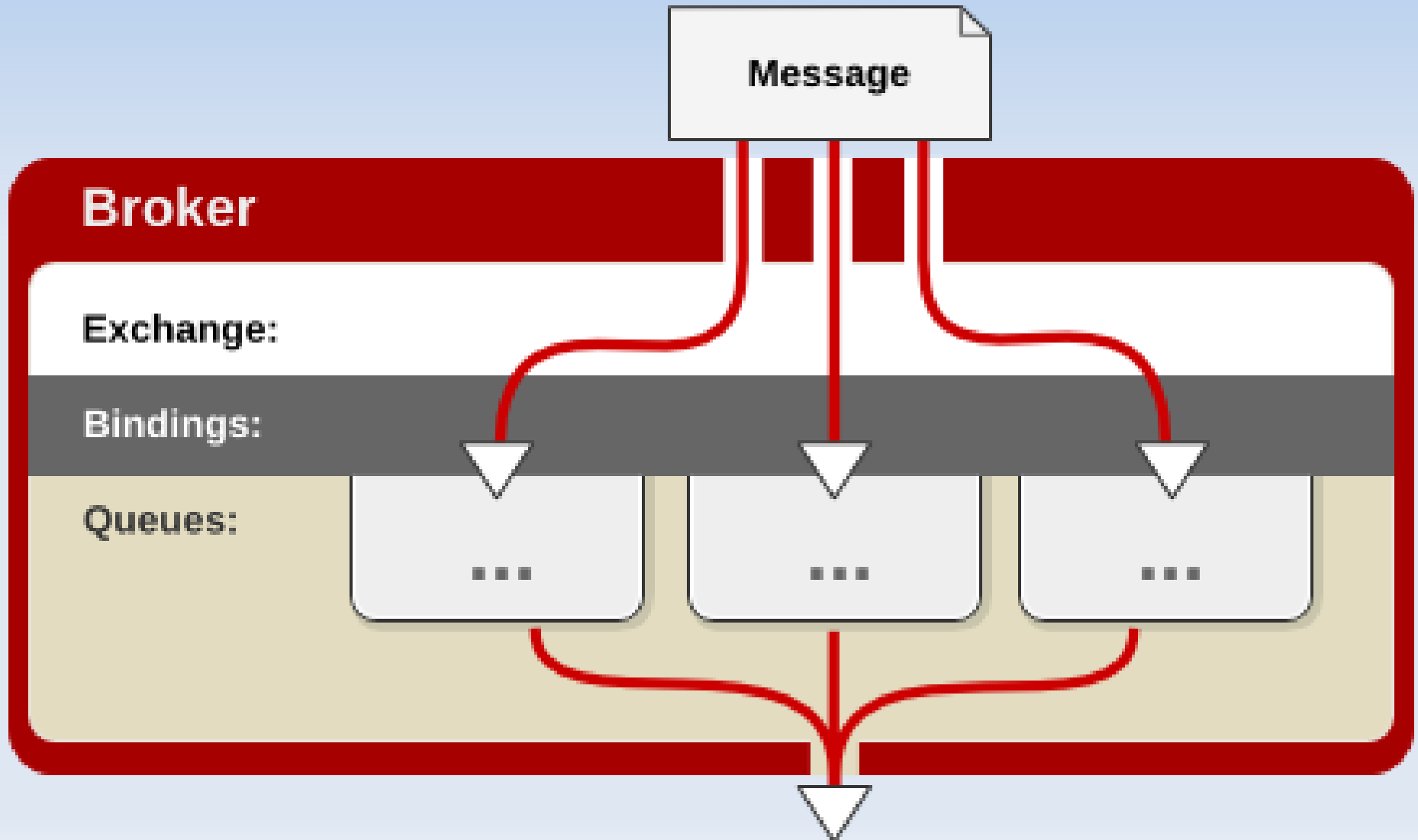
Producer-Consumer

Producer Consumer



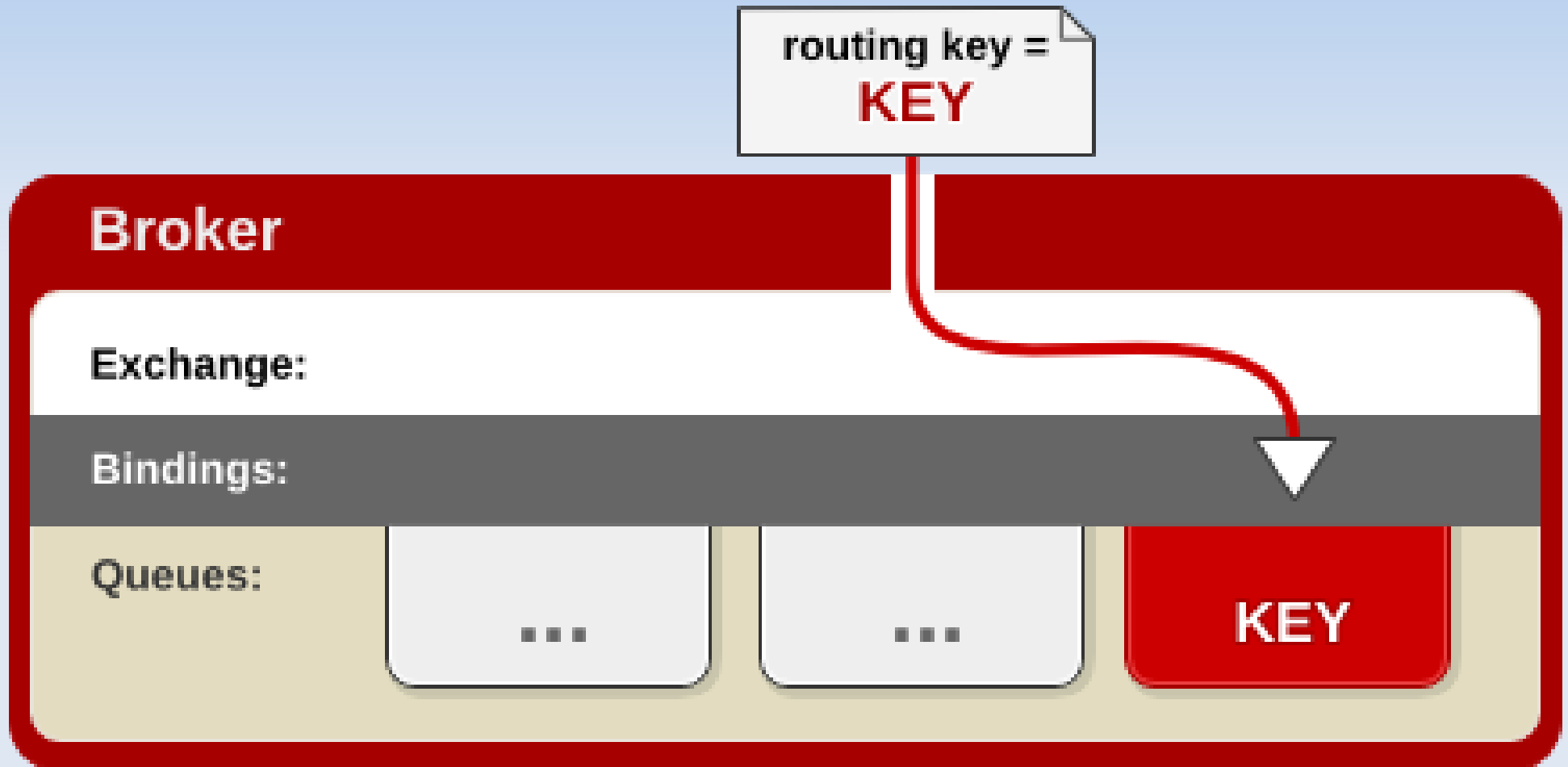
Fanout

Fanout Exchange



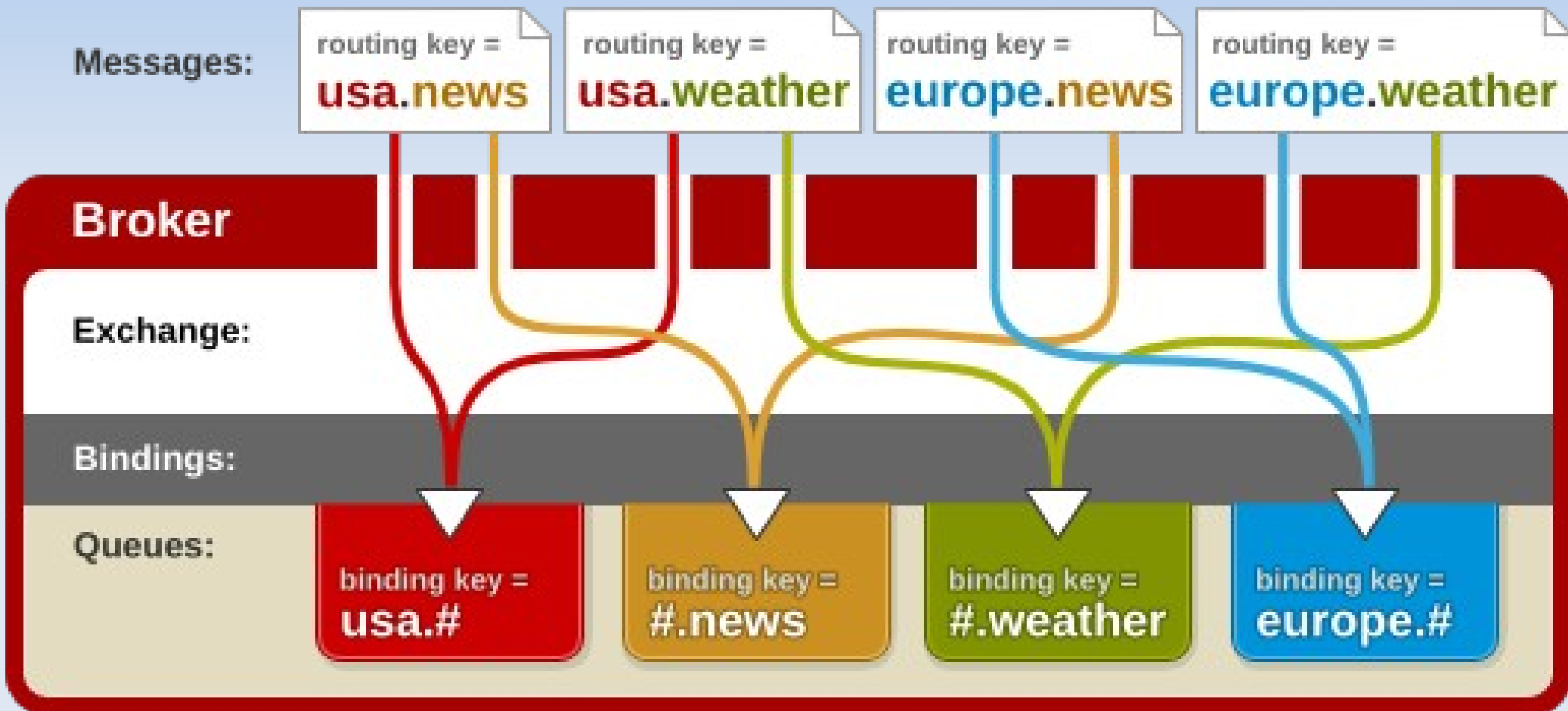
Direct

Direct Exchange



Topic

Topic Exchange



Broadcast send

```
from amqplib import client_0_8 as amqp
conn = amqp.Connection()
chan = conn.channel()
msg = amqp.Message( ' '.join(sys.argv[1:]) )

try:
    chan.basic_publish(msg, exchange="amq.fanout")
    chan.close()
except amqp.AMQPChannelException as exc:
    print sys.argv[0], exc[1]
    sys.exit(1)
finally:
    conn.close()
```

Broadcast receive

```
conn = amqp.Connection()
chan = conn.channel()

chan.exchange_declare('amq.fanout', 'fanout')
chan.queue_declare(exclusive=True)
chan.queue_bind(queue='', exchange='amq.fanout')

chan.basic_consume(callback=recv)
try:
    while True:
        chan.wait()
except KeyboardInterrupt:
    pass
chan.close() ; conn.close()
```

Direct send

```
conn = amqp.Connection()
chan = conn.channel()

args = sys.argv[1:]
queuename = args.pop(0)
msg = amqp.Message( ' '.join(args) )
chan.basic_publish(msg, exchange="amq.direct",
    routing_key=queuename)

chan.close()
conn.close()
```

Test: Simple

- `$./qlisten.py &`
- `[1] 23030`
- `$ Thu Nov 19 11:45:43 2009 23030 listening`
-
- `$./qecho.py `fortune -s``
- `Thu Nov 19 11:46:13 2009 23030 C: {} God is dead and I don't feel all too well either..... -- Ralph Moonen`

Test: No server

- `$ kill 23030`
-
- `$./qecho.py `fortune -s``
- `./qecho.py NOT_FOUND - no exchange 'myfanout' in vhost
'/'`
-

Test: Two servers

- `$./qlisten.py &`
- `[1] 23452`
- `$./qlisten.py &`
- `[2] 23456`
- `$./qecho.py `fortune -s``
- `Thu Nov 19 11:50:22 2009 23452 C: {} "I keep seeing spots in front of my eyes." "Did you ever see a doctor?" "No, just spots."`
- `Thu Nov 19 11:50:22 2009 23456 C: {} "I keep seeing spots in front of my eyes." "Did you ever see a doctor?" "No, just spots."`
-

AMQP/1.0 changes

- service locations in DNS
- no exchanges, now "queues + links"
- addresses now first class entry

Further Reading

- Rabbits and Warrens
- `help()` is really good!
- RedHat

AMQP & other messaging

- Implementations of AMQP
 - RabbitMQ (in Erlang!)
 - Apache Qpid
- Others:
 - RestMS: RESTful, based on HTTP
 - XMPP: chat-like, now extended
 - - Google Wave federation

Thanks!

No designers were injured in the making of this presentation

Credits

- Pretty routing diagrams
 - RedHat
 - Wikipedia "AMQP"



Comparison: D-Bus

- Desktop bus for GNOME et al
- program registers a service
- routes hardware abstraction layer (HAL) messages

- "disable screen saver when I watch a movie"
- "switch chat status from panel applet"

Publish/subscribe

- send to *classes* of receivers, not one
 - receive filtered subset of msgs
-
- + loosely-coupled
 - + scalable
 - silent crashes
 - confusing routes
 - network storms

Message Queuing

- message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security.
- Messages, not connections.

Further Reading

- Rabbits and Warrens
- `help()` is really good!
- RedHat

Other Standards

- RestMS: RESTful, based on HTTP
- XMPP: chat-like, now extended
 - Google Wave federation

Comparison: Email

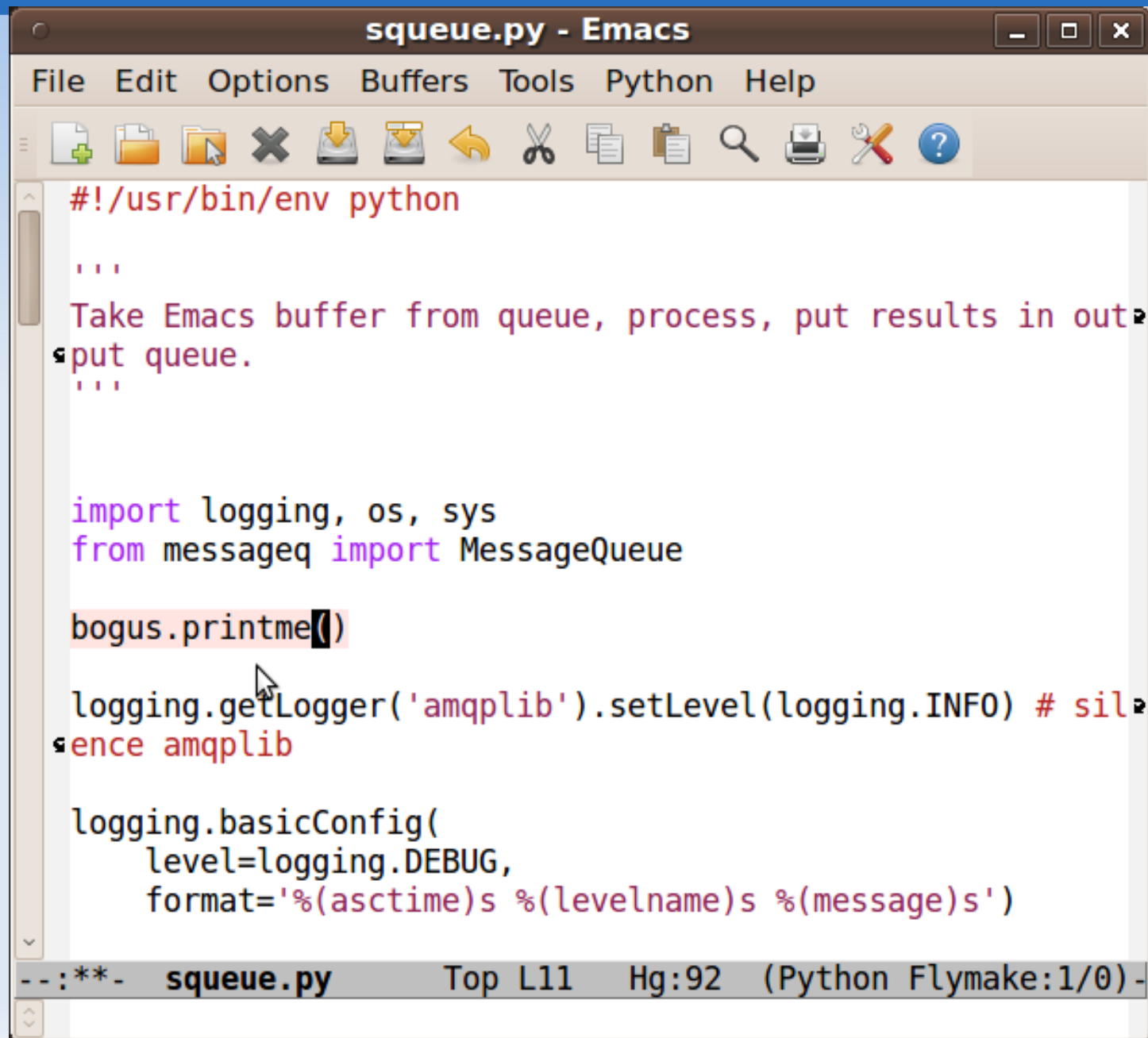
- message: header, body
- header: From, To, Subject
- other fields like "x-face"
- "Reply-To:"

Further Reading

- Rabbits and Warrens
- `help()` is really good!
- RedHat

Flymake

Flymake detail



```
#!/usr/bin/env python

'''
Take Emacs buffer from queue, process, put results in out
put queue.
'''

import logging, os, sys
from messageq import MessageQueue

bogus.printme()

logging.getLogger('amqplib').setLevel(logging.INFO) # sil
ence amqplib

logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s %(levelname)s %(message)s')
```

--:**- queue.py Top L11 Hg:92 (Python Flymake:1/0)-

AMQP & others

- Implementations

```
#!/usr/bin/env python
...
import logging, os, sys
from messageq import MessageQueue

bogus.printme()
logging.getLogger('amqplib').setLevel(logging.INFO) # sil
ence amqplib

logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s %(levelname)s %(message)s')
```

RabbitMQ (in Erlang!)

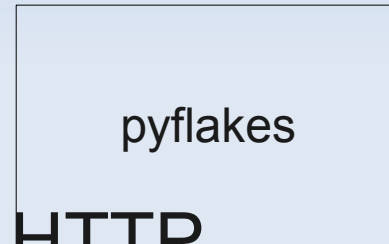
Apache Qpid

- Other Standards

RestMS: RESTful, based on HTTP

XMPP: chat-like, now extended

- Google Wave federation



sumodev

Implementations of AMQP

- RabbitMQ (in Erlang!)

```
squeue.py - Emacs
File Edit Options Buffers Tools Python Help
#!/usr/bin/env python
...
Take Emacs buffer from queue, process, put results in out
put queue.
...

import logging, os, sys
from messageq import MessageQueue

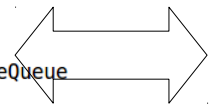
bogus.printme()

logging.getLogger('amqplib').setLevel(logging.INFO) # sil
ence amqplib

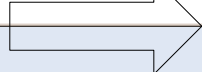
logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s %(levelname)s %(message)s')

--:**- squeue.py Top L11 Hg:92 (Python Flymake:1/0)-
```

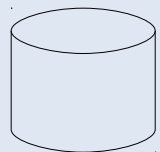
Apache Qpid



sumodev



source	pylint
source	nosetests



Implementations of AMQP

- RabbitMQ (in Erlang!)
- Apache Qpid