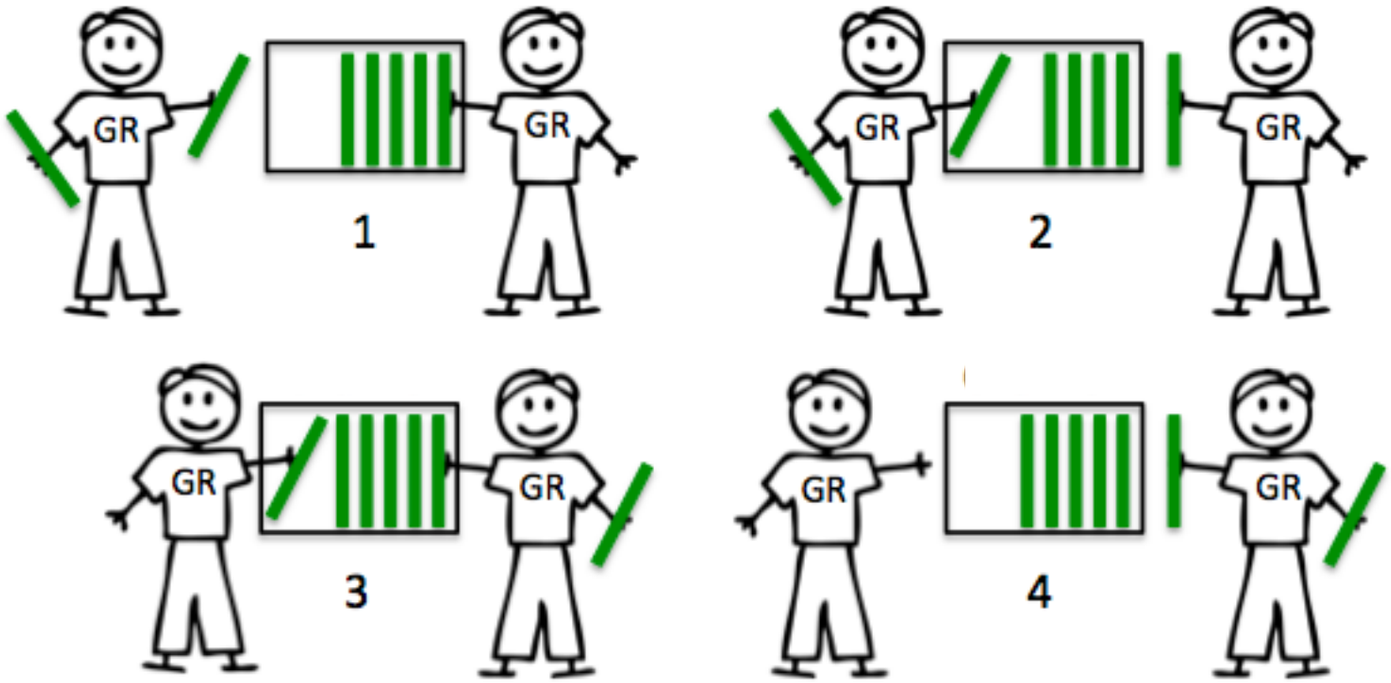
Django Channels

Greg Schafer



Outline

- Overview
- Context: Django history
- Context: WebSockets history/ecosystem
- Goals, Architecture, Considerations
- Demo + Code

Overview

- Architecture change to Django
- Motivation = solution for websockets
- Andrew Godwin
- Separate library

History of Django

Version	Date	Notes
0.90[32]	16 Nov 2005	
0.91[33]	11 Jan 2006	"new-admin"
0.95[34]	29 Jul 2006	"magic removal"
0.96[35]	23 Mar 2007	"newforms", testing tools
1.0[36]	3 Sep 2008	API stability, decoupled admin, unicode
1.1[37]	29 Jul 2009	Aggregates, transaction based tests
1.2[38]	17 May 2010	Multiple db connections, CSRF, model validation
1.3[39]	23 Mar 2011	Class based views, staticfiles
1.4[40]	23 Mar 2012	Timezones, in browser testing, app templates. [41]
1.5[42]	26 Feb 2013	Python 3 Support, configurable user model
1.6[43]	6 Nov 2013	Dedicated to Malcolm Tredinnick, db transaction management, connection pooling.
1.7[44]	2 Sep 2014	Migrations, application loading and configuration.
1.8[45]	1 Apr 2015	Native support for multiple template engines. **Long-term support release, supported until at least April 2018**
1.9[46]	1 Dec 2015	Automatic password validation. New styling for admin interface.

Django Release Schedule

Release Series	Release Date	End of mainstream support[1]	End of extended support[2]
1.9	December 2015	August 2016	April 2017
1.10	August 2016	April 2017	December 2017
1.11 LTS	April 2017	December 2017	Until at least April 2020
2.0	December 2017	August 2018	April 2019

- Sidenote: Django 2.0 drops Python 2 support

Channels History

- Proposed June 2015 (after v1.8)
- Targeted for August 2016 (v1.10)
- MOSS grant ($150k) in Dec 2015
- Concerns (complexity, latency, flexibility)
- Withdrawn May 2016
    - Now targeting 1.11 or 2.0

WebSockets History

- Started in 2008, standardized in 2011
- Full-duplex TCP over port 80
- Preceded by Comet (usu. long-polling)
- Low overhead
- Upgrade header

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

Server response:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

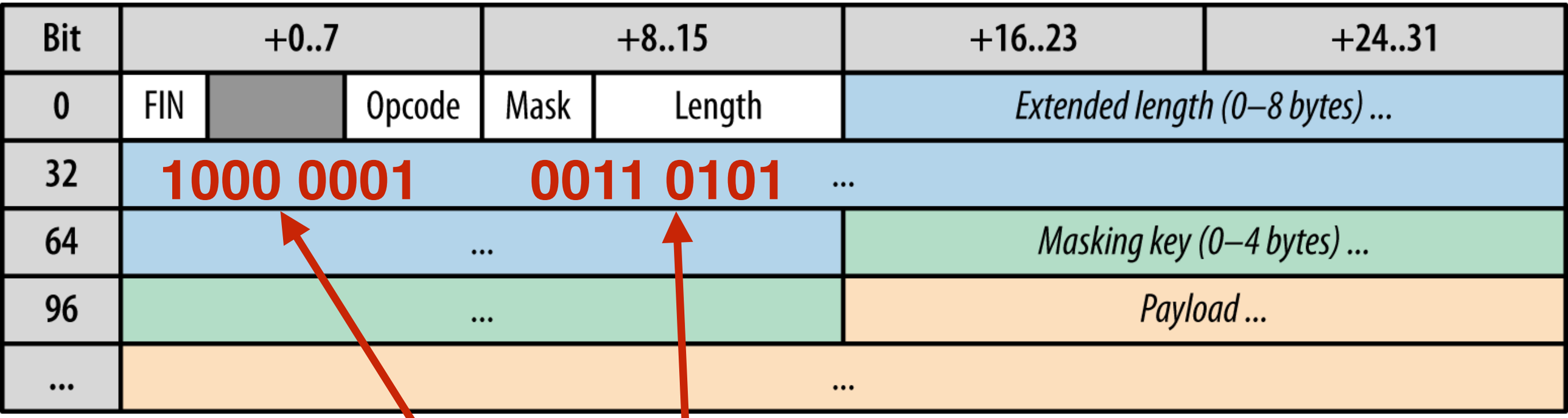
Frames

Bit	+0..7			+8..15		+16..23	+24..31
0	FIN		Opcode	Mask	Length	*Extended length (0–8 bytes) ...*	
32	...						
64	...					*Masking key (0–4 bytes) ...*	
96	...					*Payload ...*	
...	...						

```
▶ Frame 5: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 6, Src: ::1 (::1), Dst: ::1 (::1)
▶ Transmission Control Protocol, Src Port: 8000 (8000), Dst Port: 59923 (59923), Seq: 1, Ack: 35, Len: 55
▼ Data (55 bytes)
    Data: 81357b2274797065223a202263686174222c202275736572...
    [Length: 55]
```

```
0000  1e 00 00 00 60 05 70 71  00 57 06 40 00 00 00 00   ....`.pq .W.@....
0010  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 01 1f 40 ea 13   ........ .....@..
0030  d4 21 16 3b 91 58 42 11  80 18 31 ae 00 5f 00 00   .!.;.XB. ..1.._..
0040  01 01 08 0a 37 0f aa 32  37 0f aa 20 81 35 7b 22   ....7..2 7.. .5{"
0050  74 79 70 65 22 3a 20 22  63 68 61 74 22 2c 20 22   type": " chat", "
0060  75 73 65 72 6e 61 6d 65  22 3a 20 22 62 6f 62 22   username ": "bob"
0070  2c 20 22 6d 65 73 73 61  67 65 22 3a 20 22 31 32   , "messa ge": "12
0080  33 22 7d                                          3"}
```

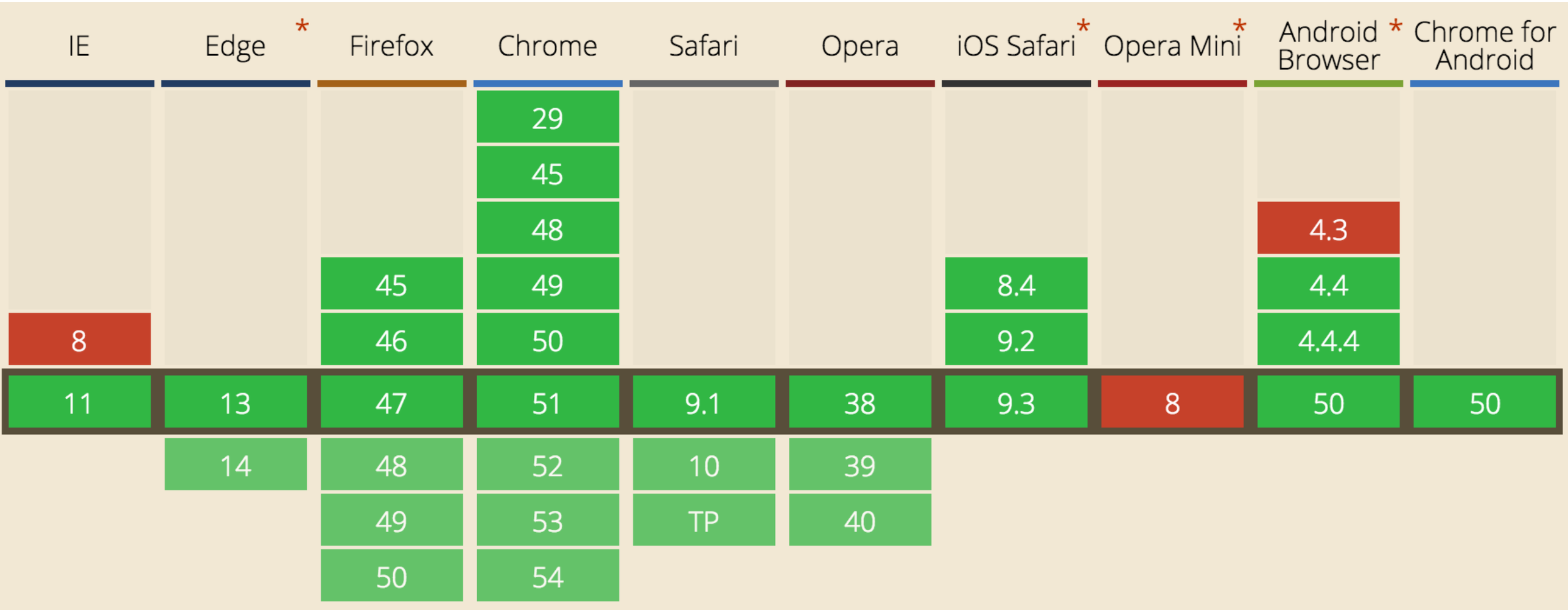
Frames




```
▶ Frame 5: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 6, Src: ::1 (::1), Dst: ::1 (::1)
▶ Transmission Control Protocol, Src Port: 8000 (8000), Dst Port: 59923 (59923), Seq: 1, Ack: 35, Len: 55
▼ Data (55 bytes)
    Data: 81357b227479706522 3a20226368617422 2c20227573657...
    [Length: 55]
```

```
0000  1e 00 00 00 60 05 70 71  00 57 06 40 00 00 00 00   ....`.pq .W.@....
0010  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 01 1f 40 ea 13   ........ .....@..
0030  d4 21 16 3b 91 58 42 11  80 18 31 ae 00 5f 00 00   .!.;.XB. ..1.._..
0040  01 01 08 0a 37 0f aa 32  37 0f aa 20 81 35 7b 22   ....7..2 7.. .5{"
0050  74 79 70 65 22 3a 20 22  63 68 61 74 22 2c 20 22   type": " chat", "
0060  75 73 65 72 6e 61 6d 65  22 3a 20 22 62 6f 62 22   username ": "bob"
0070  2c 20 22 6d 65 73 73 61  67 65 22 3a 20 22 31 32   , "messa ge": "12
0080  33 22 7d                                          3"}
```

WebSockets Support



WebSockets Ecosystem

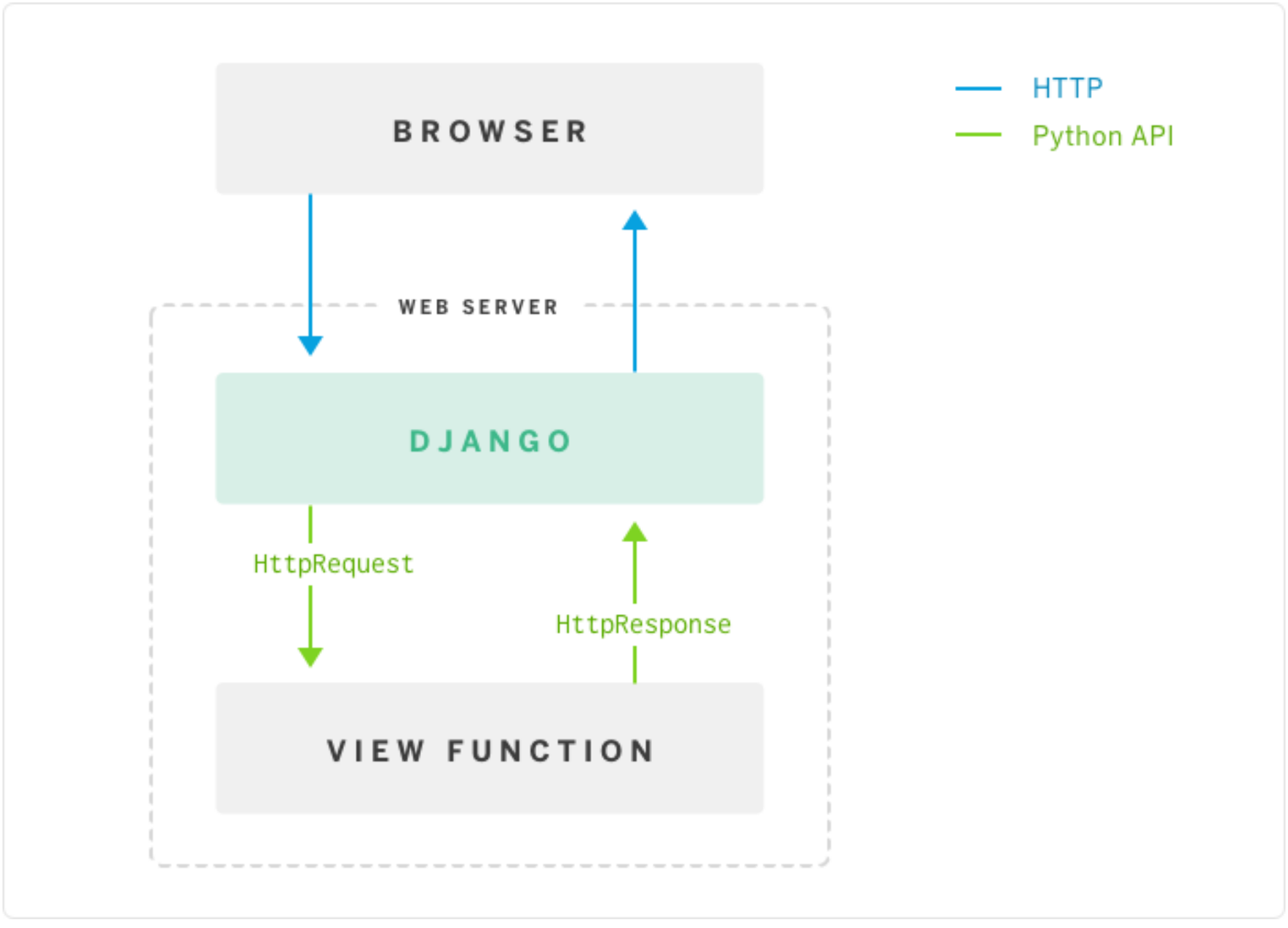
- Rails
  - Rails 5 (June 22) introduced ActionCable
    - Faye, em-websocket, many gems
- Node (lots)
  - Faye, socket.io, Primus (abstraction layer)
- Python
  - Autobahn, crossbar.io

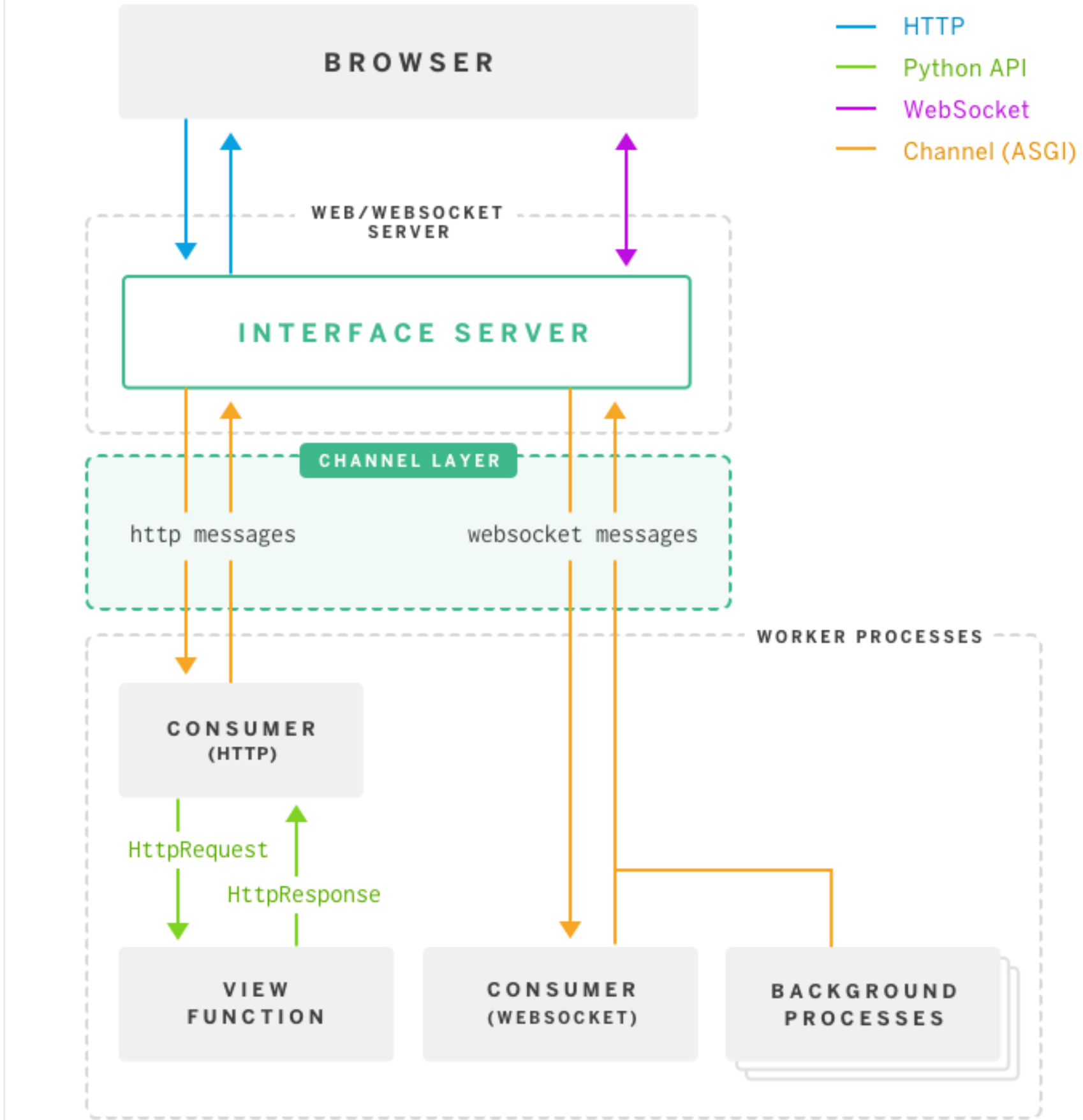
Goals of Channels

- Backwards compatible
- Simple (workers run synchronous code)

- Low latency + high throughput
  - Tradeoff = sacrifice guaranteed delivery
- Network transparent

Architecture Change




BROWSER
HTTP
Python API
WebSocket
Channel (ASGI)
WEB/WEBSOCKET SERVER
INTERFACE SERVER
CHANNEL LAYER
http messages
websocket messages
WORKER PROCESSES
CONSUMER (HTTP)
HttpRequest
HttpResponse
VIEW FUNCTION
CONSUMER (WEBSOCKET)
BACKGROUND PROCESSES

Spec

- Channel layer interface (asgi_redis)
  - `send(channel, message)`
  - `receive_many(channels, block=False)`
- http://channels.readthedocs.io/en/latest/asgi.html#specification-details

Groups

- Channel layer interface (asgi_redis)
  - `group_add(group, channel)`
  - `group_discard(group, channel)`
  - `send_group(group, message)`

Considerations

- Channel = ordered, first-in first-out queue with message expiry and **at-most-once delivery** to only one listener at a time

- Celery replacement? No

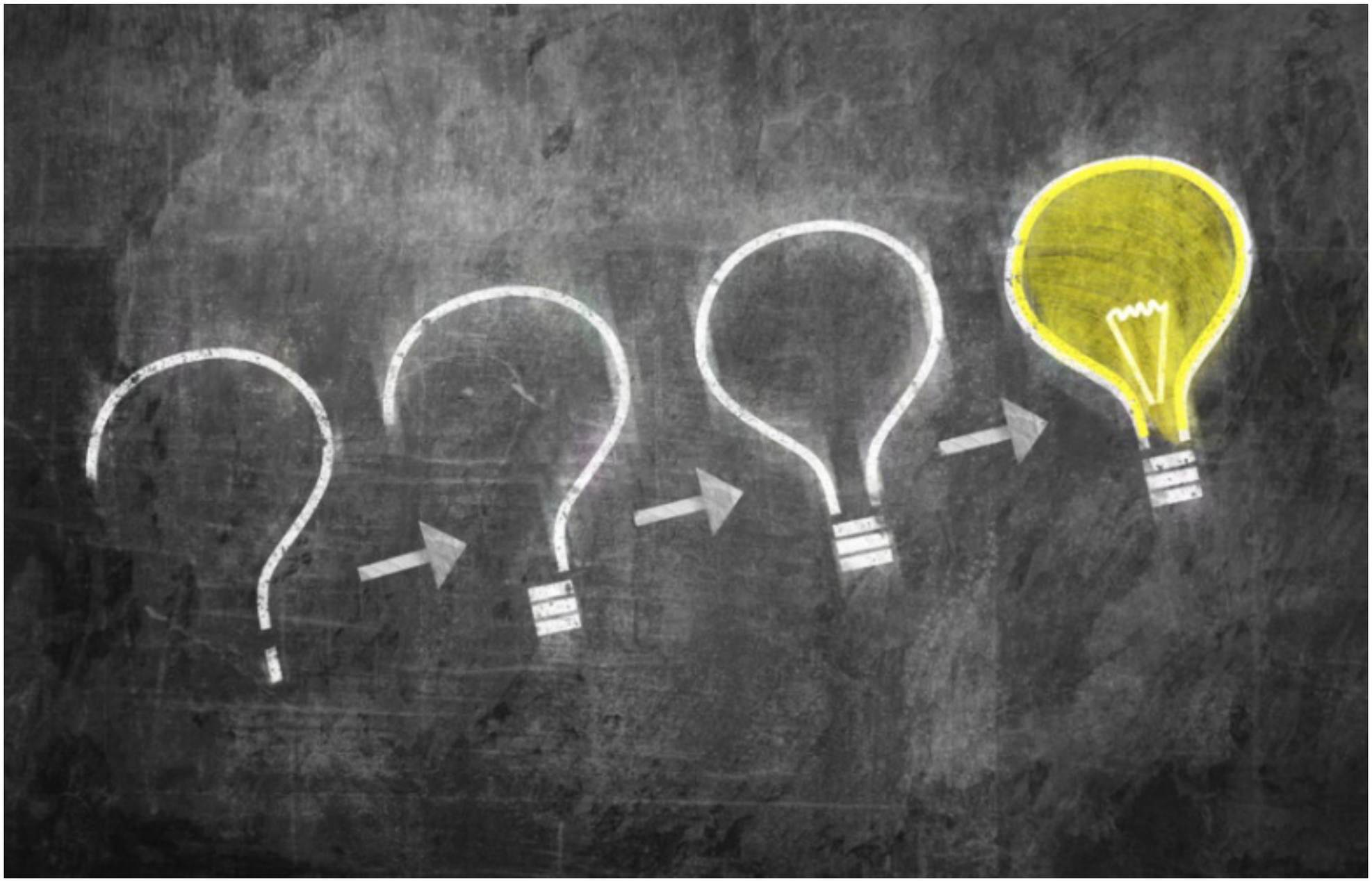
Demo: Chat

- http://10.1.10.242:8000/
- Kill worker, inspect redis
  - Message timeout, no delivery
- Next steps
  - Models: http://channels.readthedocs.io/en/latest/getting-started.html#models

Demo: Drawing

- Generalized handling
- Throughput

Questions?



https://github.com/grschafer/pythonweb-meetup-channels