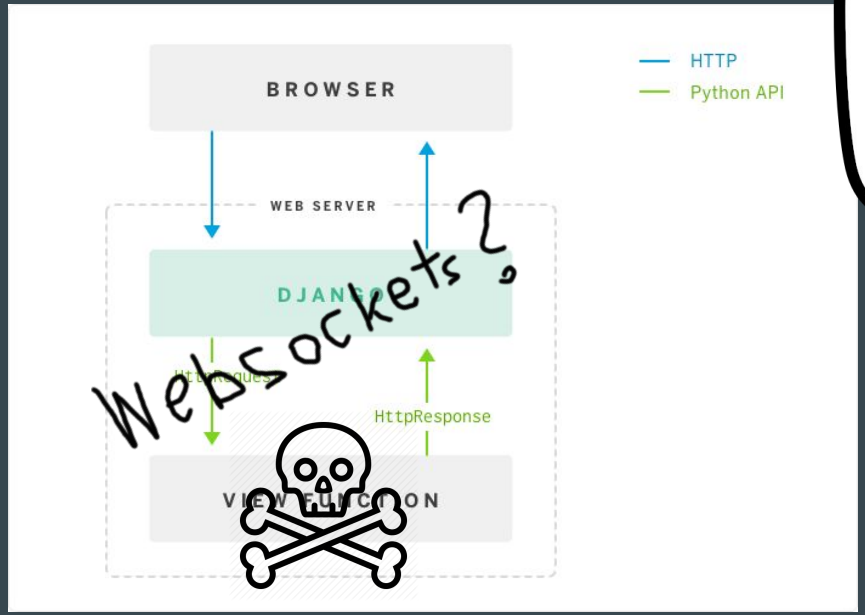
Asynchronous Django with Django Channels

...

Django Vienna 11/2019

I have heard about this async stuff but what do I actually really need it for?!?



Let's add some real-time notifications!

https://blog.heroku.com/in_deep_with_django_channels_the_future_of_real_time_apps_in_django

What is this good for?

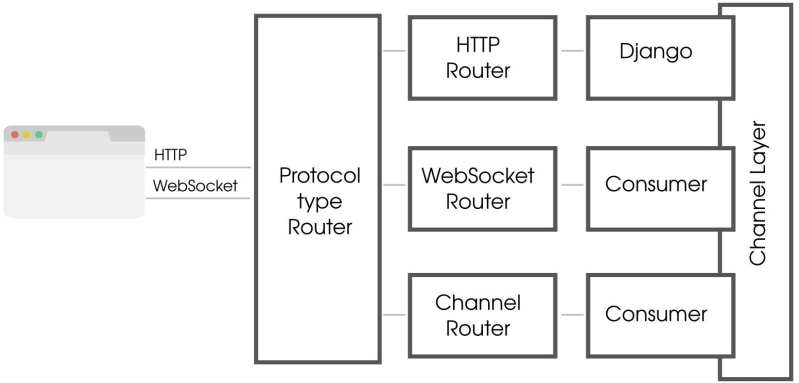
- Use asynchronous code with Django: WebSockets, HTTP long-polling, MQTT (IoT)...
- ... while preserving Django's synchronous and easy-to-use nature
- ... and keep Django's auth and session system, ORM and much more.

Evolution

- Channels 1 - Python 2
- **Channels 2 - Python 3.5 and up, uses `asyncio`**
- “Offical” django.contrib project
  https://github.com/django/channels/

Building Blocks

- (Protocol) Router - our “urls.py” for different protocols
- Consumers - our “views” for different protocols
- Channel Layer - we just use Redis for this!
- ASGI



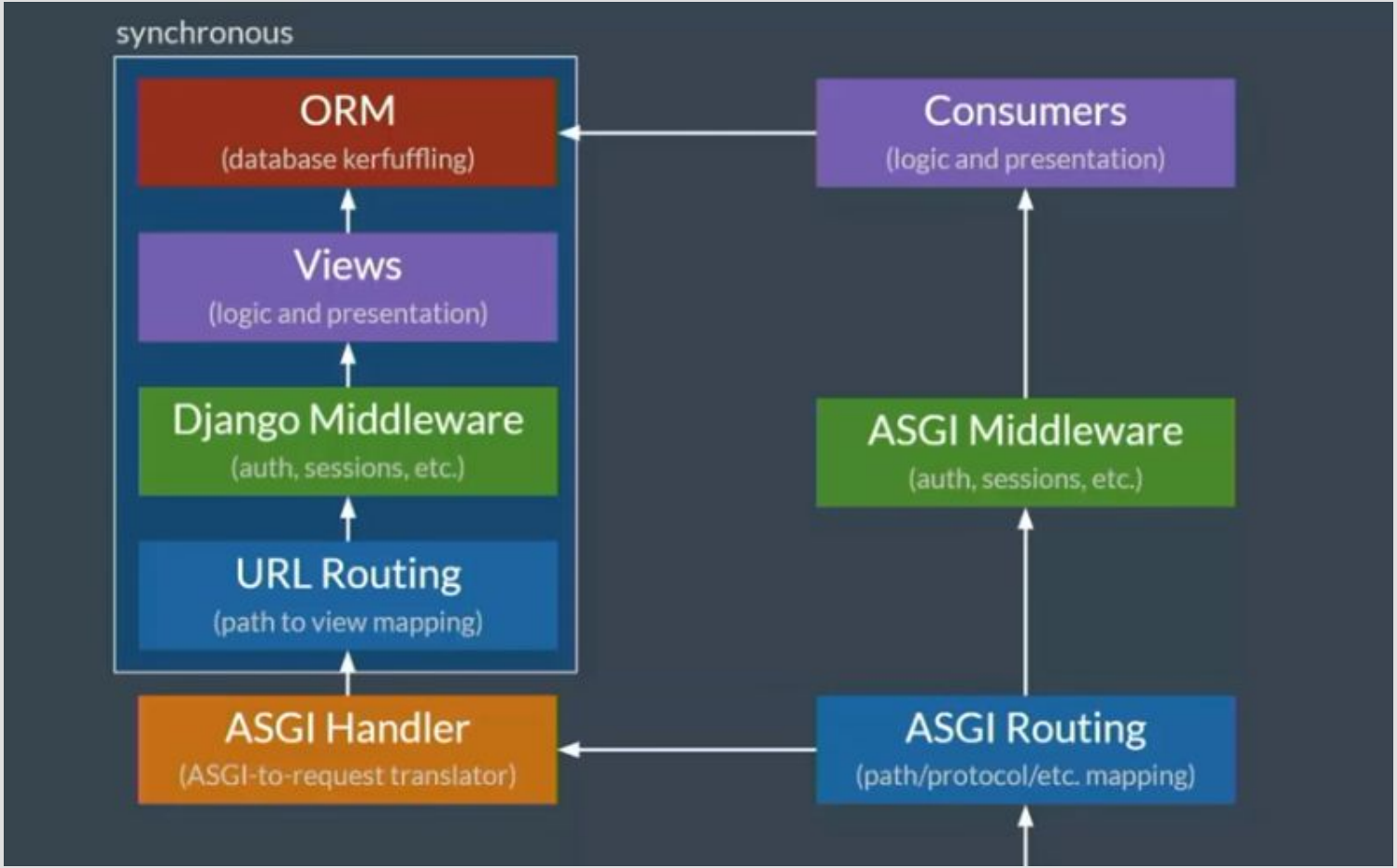


Image by Andrew Goodwin

Routers

```
application = ProtocolTypeRouter({

    # WebSocket chat handler
    "websocket": AuthMiddlewareStack(
        URLRouter([
            url(r"^chat/admin/$", AdminChatConsumer),
            url(r"^chat/$", PublicChatConsumer),
        ])
    ),

    # Using the third-party project frequensgi, which provides an
APRS protocol
    "aprs": APRSNewsConsumer,

})
```

Consumers

- Instantiated for each connection
- Receive a `scope` object
- Methods for sending and receiving data, “messages”

```
from channels.consumer import AsyncConsumer

class EchoConsumer(AsyncConsumer):

    async def websocket_connect(self, event):
        await self.send({
            "type": "websocket.accept",
        })

    async def websocket_receive(self, event):
        await self.send({
            "type": "websocket.send",
            "text": event["text"],
        })
```

Worth Knowing

- Be careful when calling sync-code from async methods: eg. use `channels.db.database_sync_to_async()` when using the ORM
- You can also run workers/background tasks, but guarantees are weak ("at-most-once delivery")
- With Django 3 (RC this week) Django itself is starting to become Async, but WebSocket support will stay in django-channels. https://github.com/django/deps/blob/master/accepted/0009-async.rst

Thank You!

•••

https://github.com/bvallant/django-channels-demo/

https://channels.readthedocs.io/

Thank You!

Bernhard Vallant
bernhard@pagestrip.com