

CGI:1993 - WSGI:2003 - ASGI:2016 草案进行时

ASGI



# Asynchronous Server Gateway Interface

## 异步服务网关接口

BACK TO 1989....





**Tim Berners-Lee**





**http://www**







声音 | 1 小时前

# 公投退欧几个小时后，英国公民却在 Google “欧盟是什么？”

17  
SHARES

[卡梅隆](#)
[欧盟](#)
[英国退欧](#)
[英国首相](#)
[谷歌趋势](#)

4 2



在英国公投退欧后，似乎应该是大部分英国民众狂欢的场景。但在事件发生后 Google Trends 上呈现了一种更有趣的搜索趋势——比如这个问题“欧盟是什么？”

在北京时间 6 月 24 日下午公投结果显示，在 3300 多万有效投票的英国民众之中有 51.9% 的英国民众投票支持退欧，剩下 48.1% 的投票人士支持留在欧盟。

英国这是要退出欧盟了，就连英国首相卡梅隆也在新闻发布会上[宣布将辞去首相职务](#)。

Washingtonpost 今日发表文章报道称，在英国公投最终结果公布后的几小时内，

王飞

高级编辑

421

文章总数

最近：公投退欧几个小时后，英国公民却在 Google “欧盟是什么？”

## 相关文章



Android 在欧洲到了，最危险的时刻

2015/4/28 12:56



为了欧洲的网络和谐，这四家美国企业准备一同行动

6/01 10:15



苹果：欧洲用户可 14 天无条件退货

2014/12/30 02:13



为保护隐私，欧盟宣布禁用微软最新版 Outlook


2015/2/11 06:53



一出生就带悲剧色彩的 Windows RT，将有怎样的结局 | 爱范早读

2015/8/10 00:09





➡ ~ telnet www.ifanr.com 80

Trying 54.223.237.139...

Connected to ifanrcom-729880750.cn-north-1.elb.amazonaws.com.cn.

Escape character is '^['.

GET /675724 http/1.1

Host: www.ifanr.com

**HTTP Request**

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Date: Sat, 25 Jun 2016 08:11:35 GMT
ETag: "576e39a4-d57e"
Server: nginx/1.11.1
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Join-Us: tifan at ifanr.com
X-UA-Compatible: IE=Edge,chrome=1
X-XSS-Protection: 1; mode=block
transfer-encoding: chunked
Connection: keep-alive
```

## Response Headers

```
<!DOCTYPE html>
<html lang="en-US">
<head>
  <meta charset="UTF-8">
  <meta property="og:site_name" content="爱范儿" />
```

## Response Body

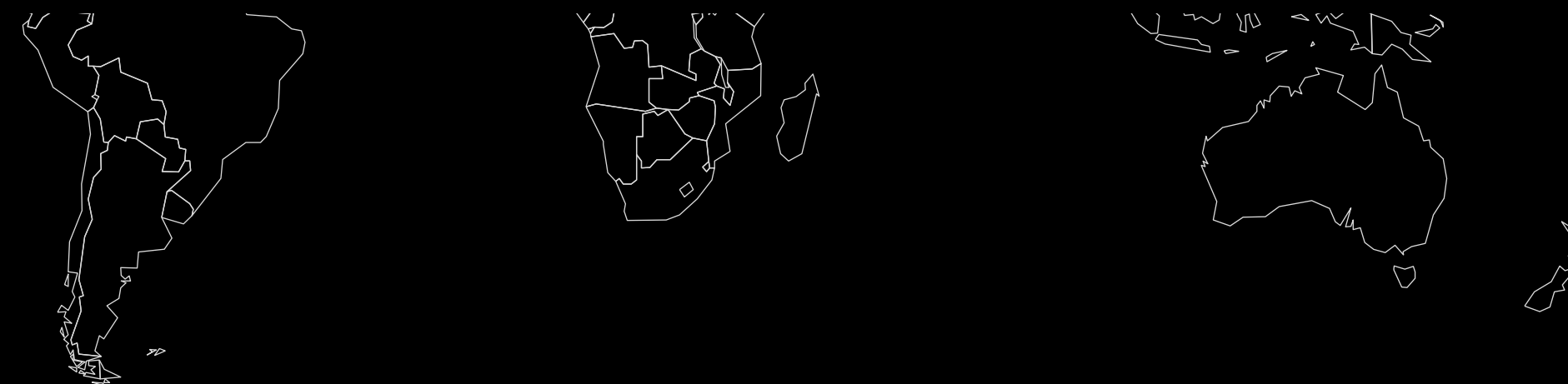








"The response from the server was always an HTML page"





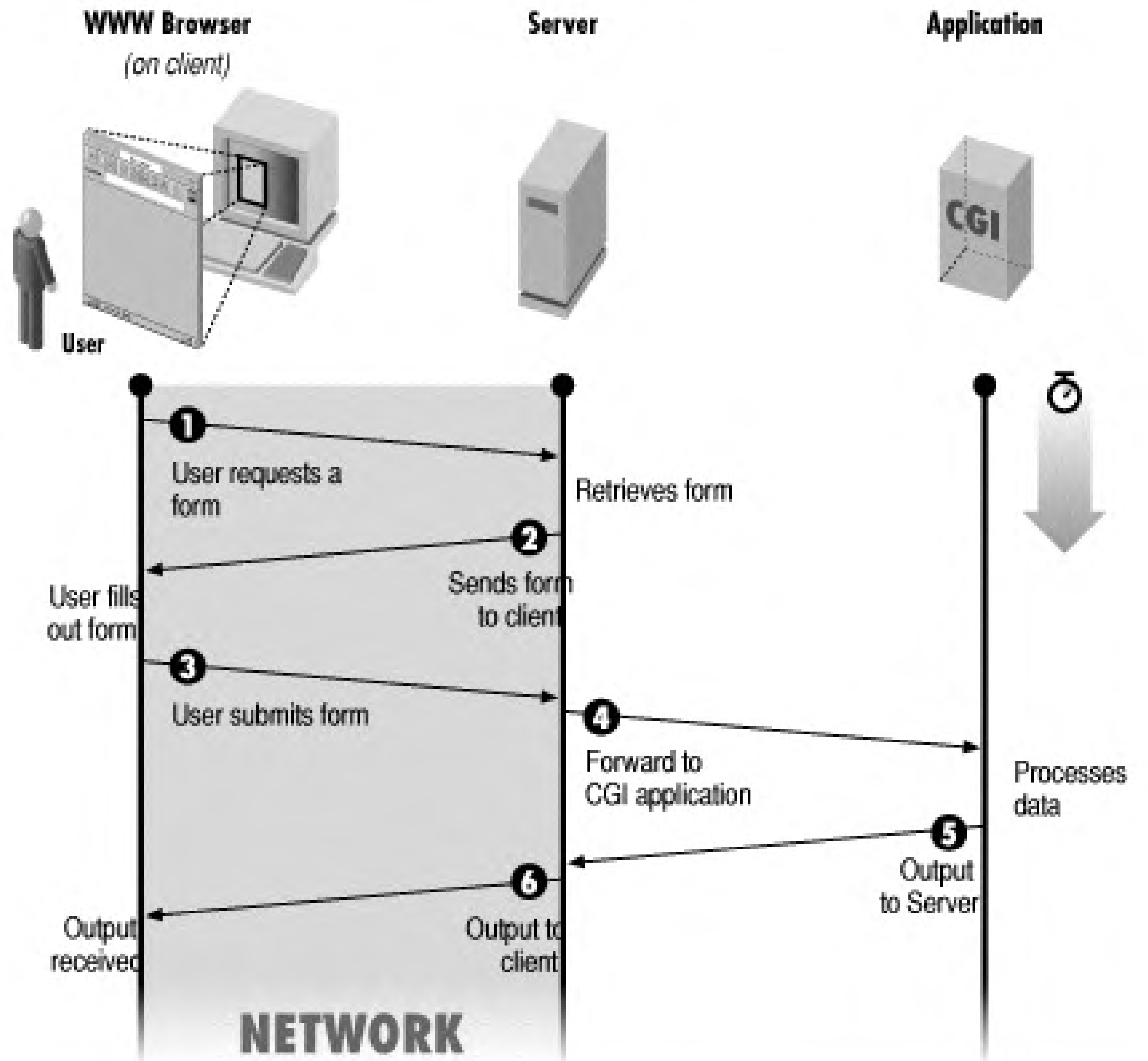
A white outline of a world map is centered in the background, showing the continents of North America, South America, Europe, Africa, Asia, and Australia.

AND THEN 1993  
COMMON GATEWAY INTERFACE  
CGI



Static HTML Documents

Executable Commands





```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <form action="add.py" method="POST">
5       Enter two numbers to add:<br />
6       First Number: <input type="text" name="num1" /><br />
7       Second Number: <input type="text" name="num2" /><br />
8       <input type="submit" value="Add" />
9     </form>
10  </body>
11 </html>
```

**add.html**



```
1 #!/usr/bin/python
2
3 import cgi
4 import cgitb
5 cgitb.enable()
6
7 input_data=cgi.FieldStorage()
8
9 print 'Content-Type:text/html' #HTML is following
10 print                          #Leave a blank line
11 print '<h1>Addition Results</h1>'
12 try:
13     num1=int(input_data["num1"].value)
14     num2=int(input_data["num2"].value)
15 except:
16     print '<p>Sorry, we cannot turn your inputs into numbers (integers).</p>'
17     return 1
18 sum=num1+num2
19 print '<p>{0} + {1} = {2}</p>'.format(num1,num2,sum) '
```

**add.py**

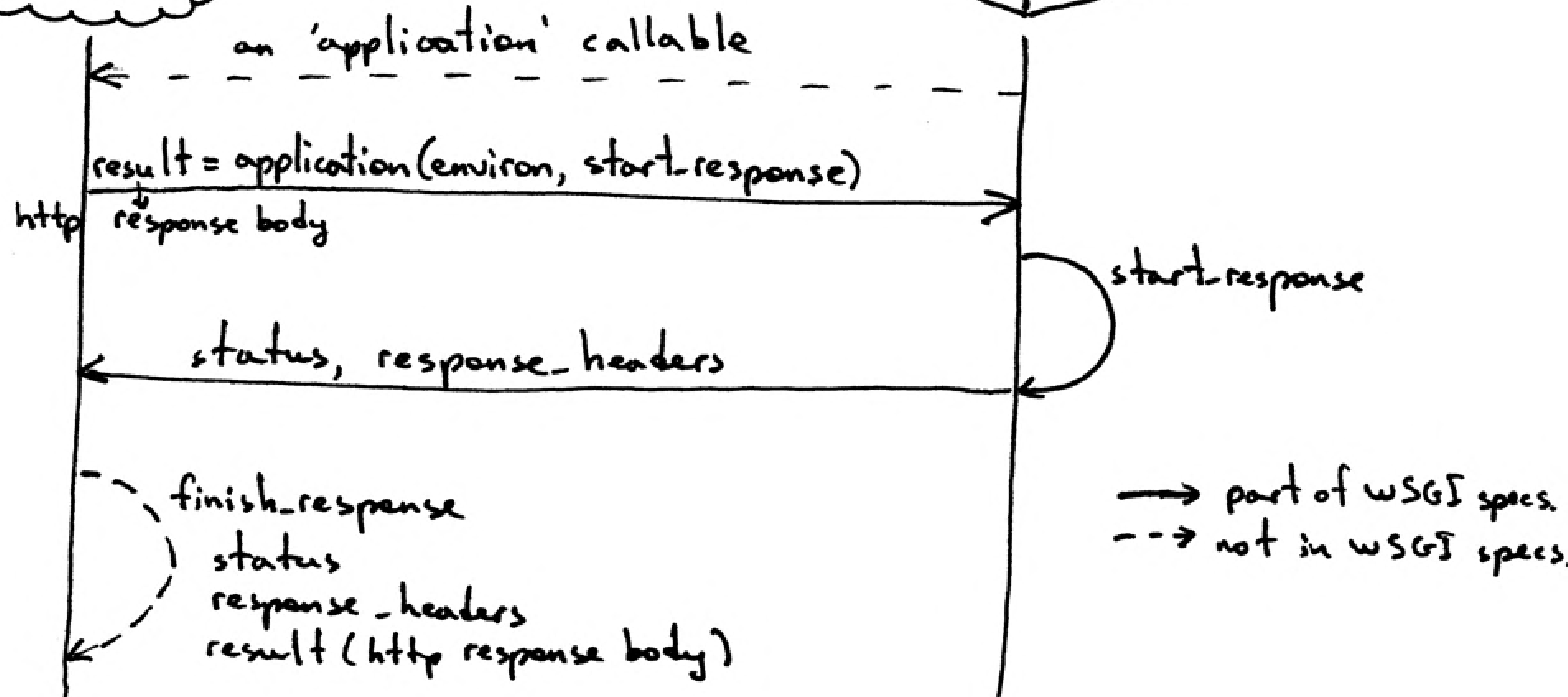
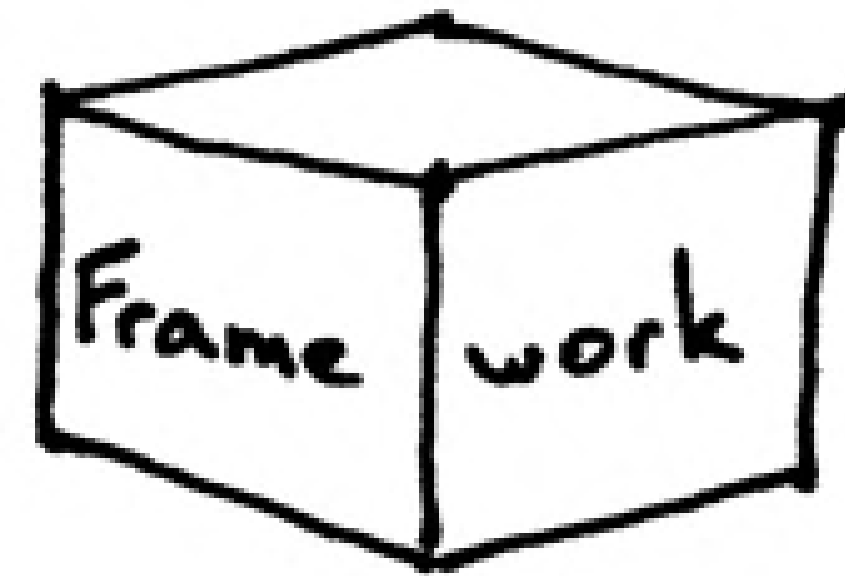
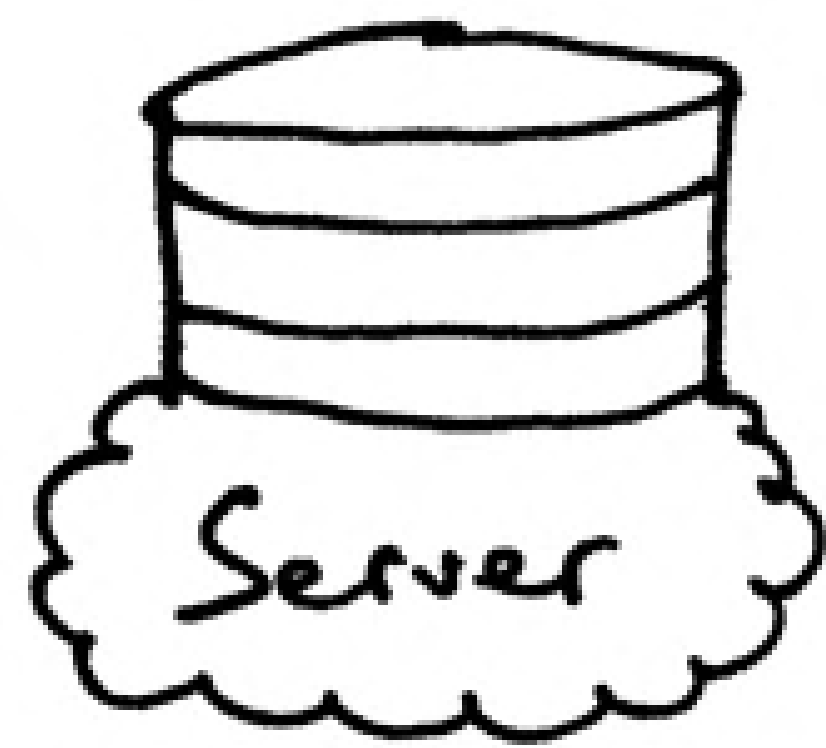


AND THEN 2003

WEB SERVER GATEWAY INTERFACE

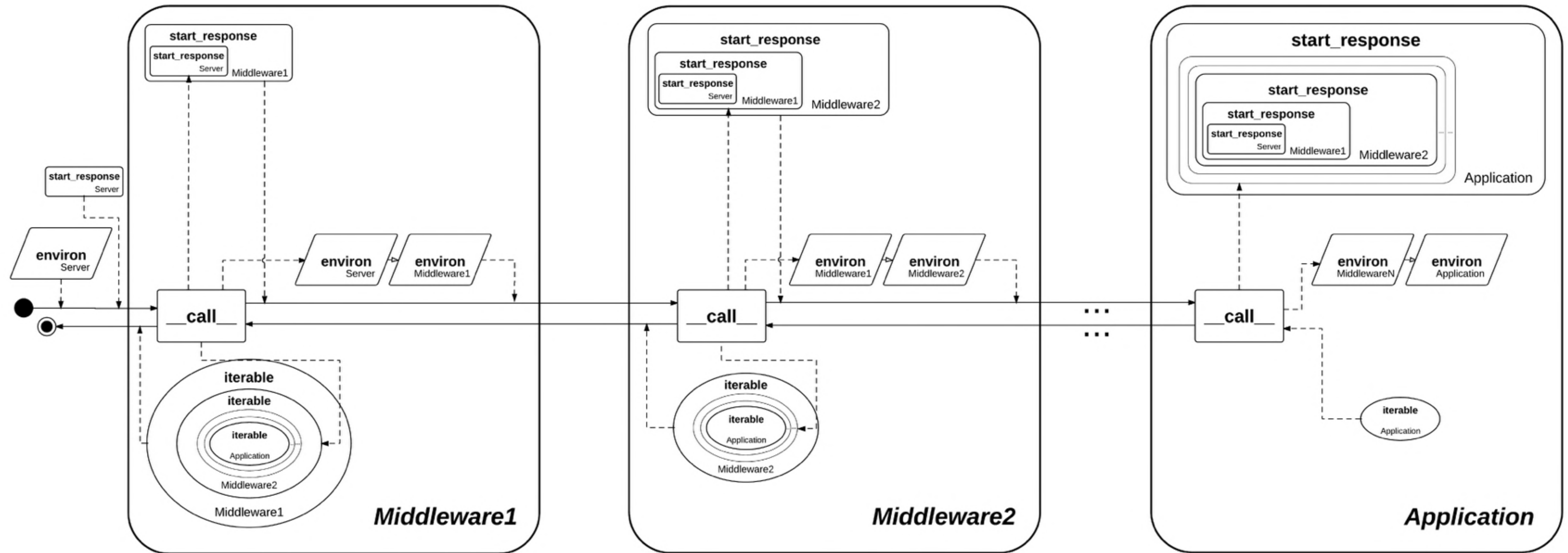
WSGI

# WSGI Interface





# WSGI flow (PEP 3333)



**WSGI Server -> (WSGI Middleware)\* -> WSGI Application**



WSGI MiddleWare

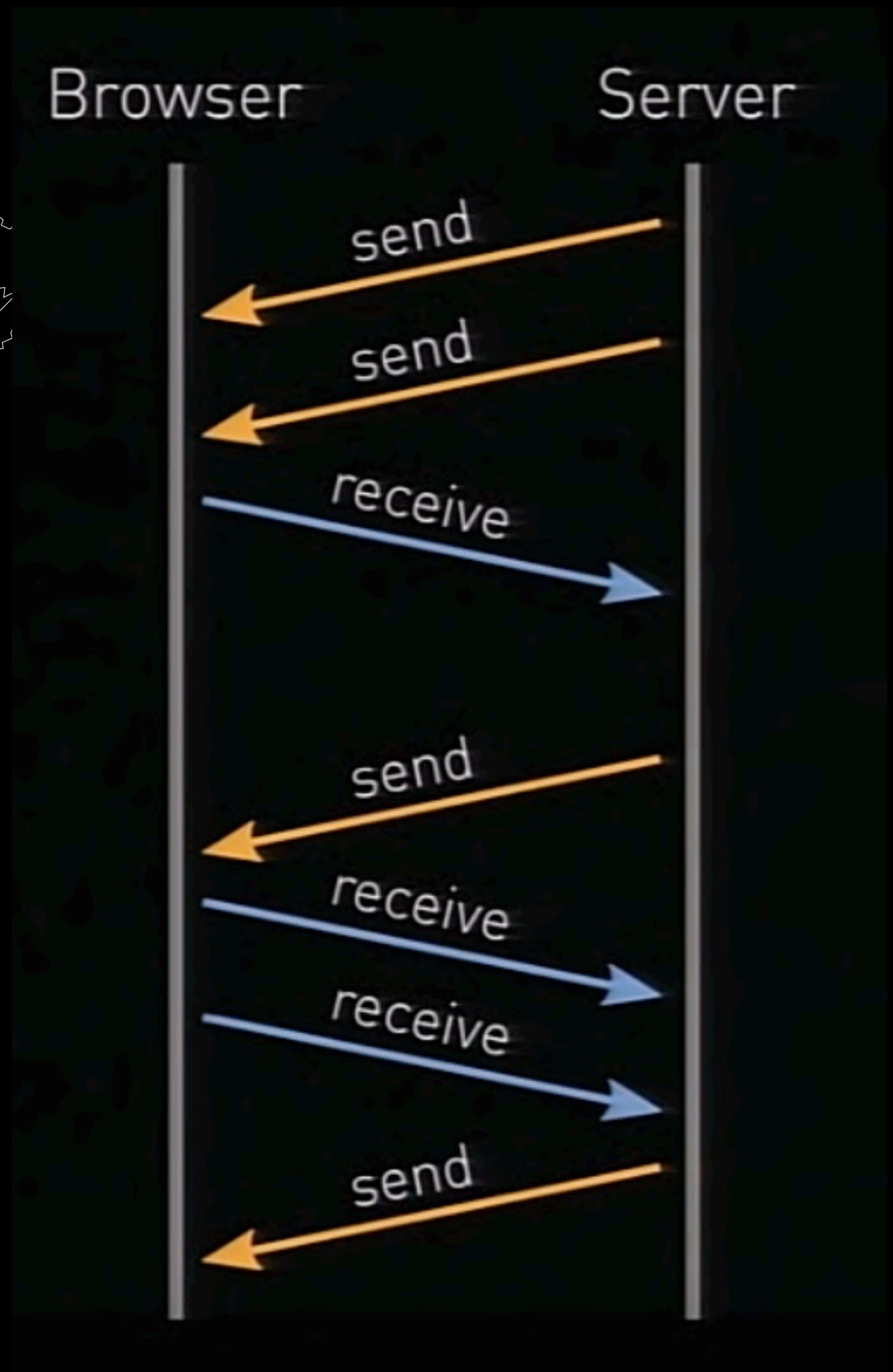




“有了 WSGI，写 web 框架就像喝水一样简单。”

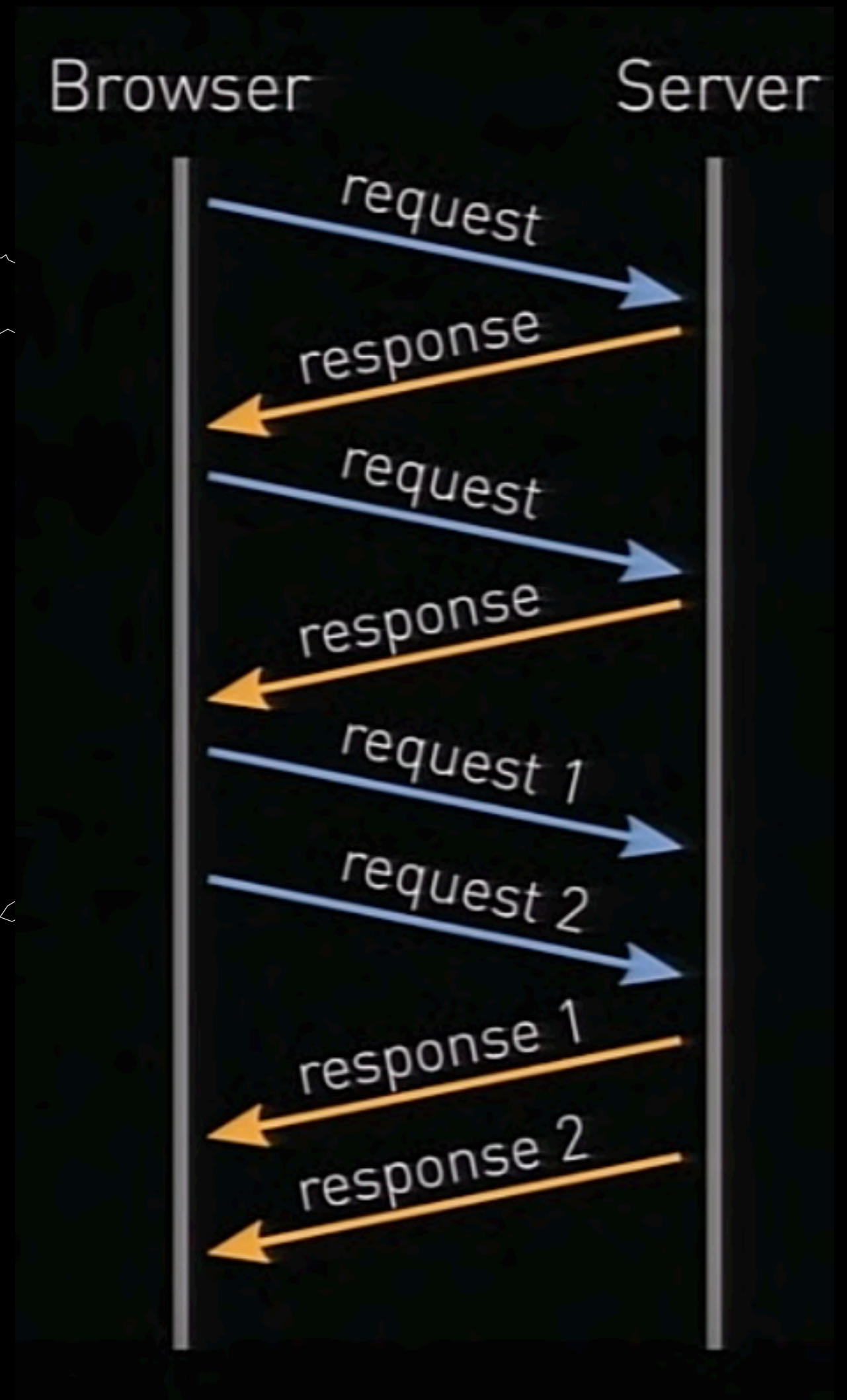
—WEB 开发者

# AND THEN 2011 WEBSOCKET (RFC 6455)





# AND THEN 2012 HTTP2 (RFC 7540)



A white outline of a world map is centered in the background of the slide.

# AND THEN 2016

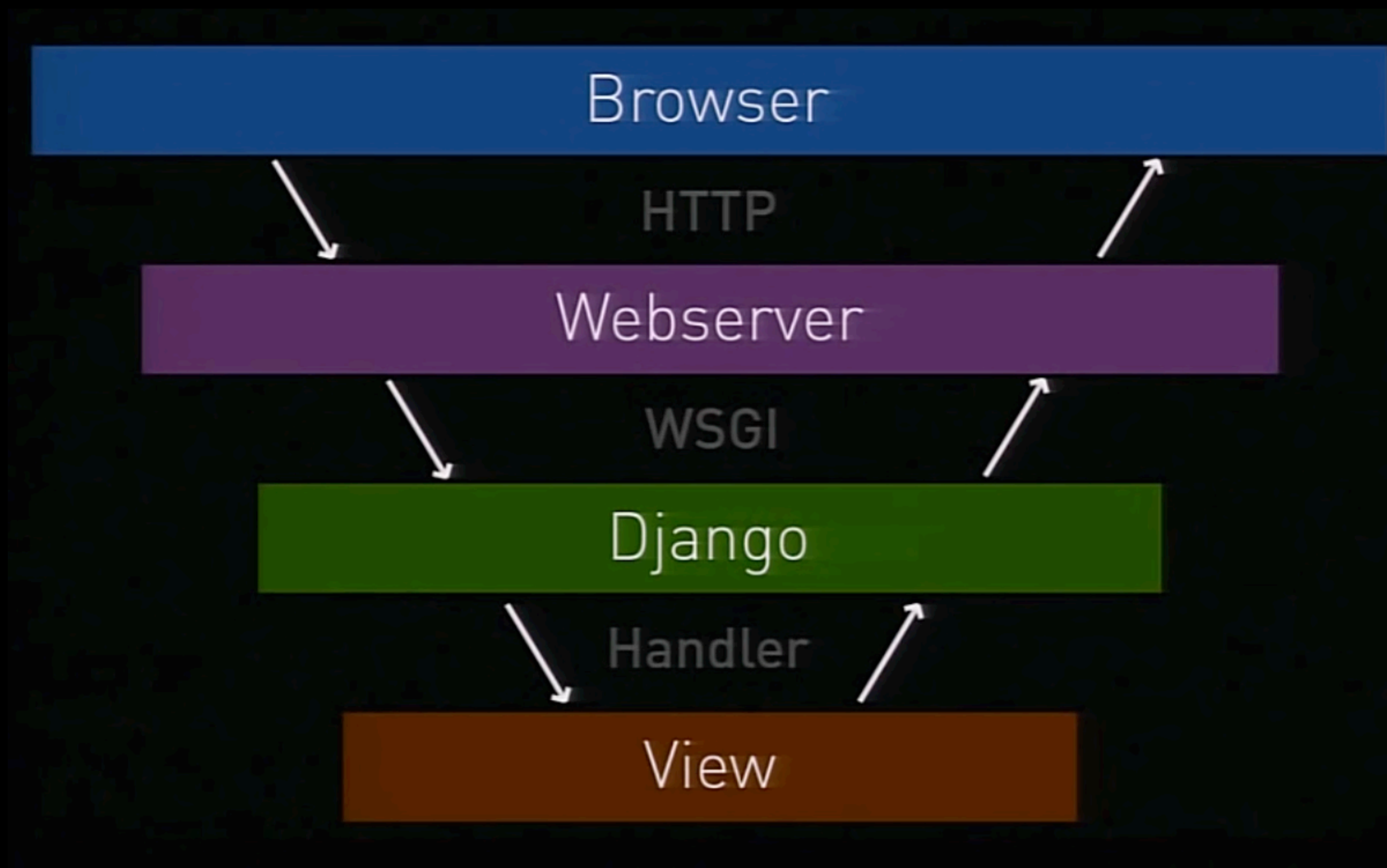
## ASYNCHRONOUS SERVER GATEWAY INTERFACE

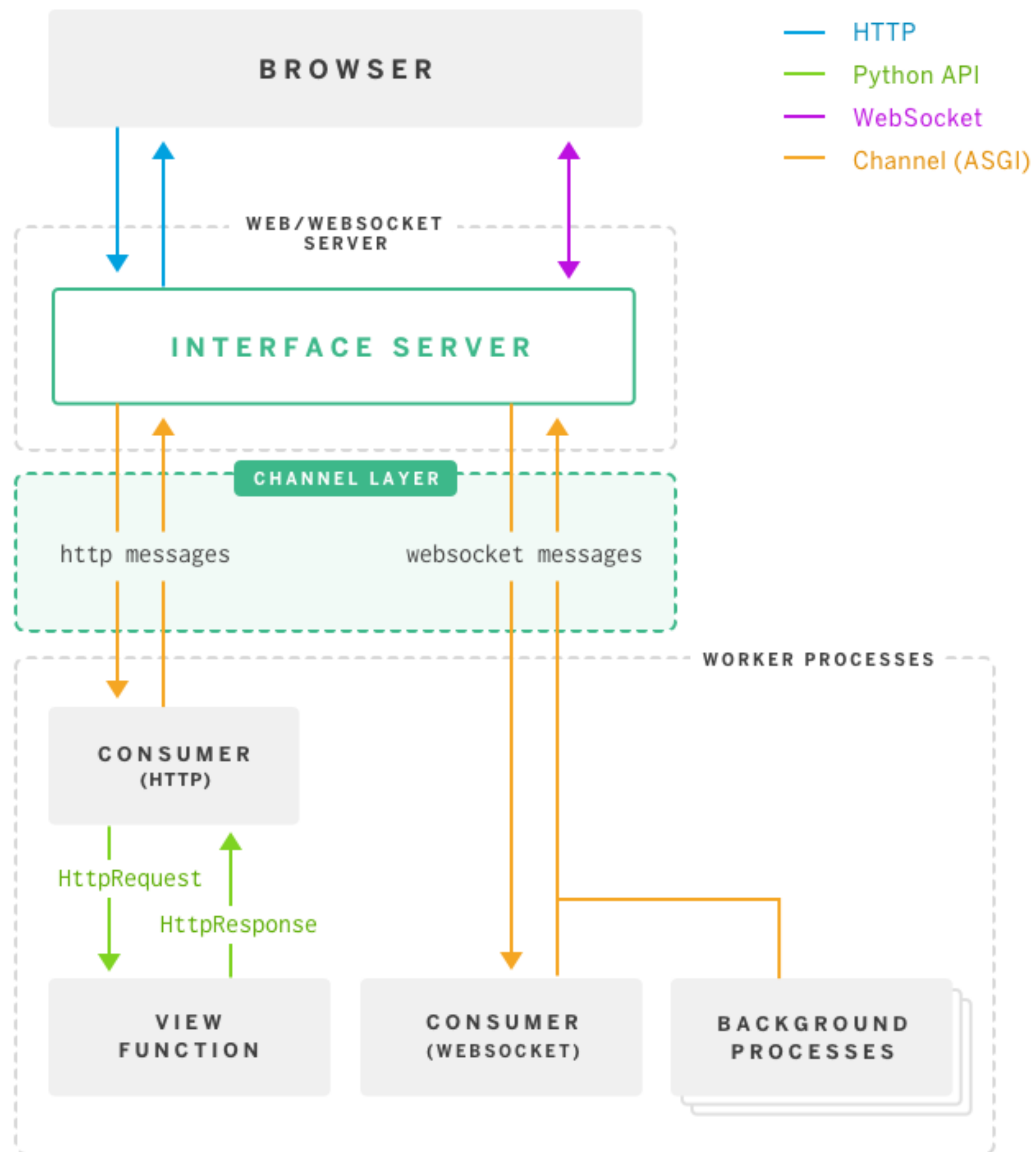
### ASGI



A white line-art map of the world is centered in the background of the slide.

**protocol server -> channel layer -> application**





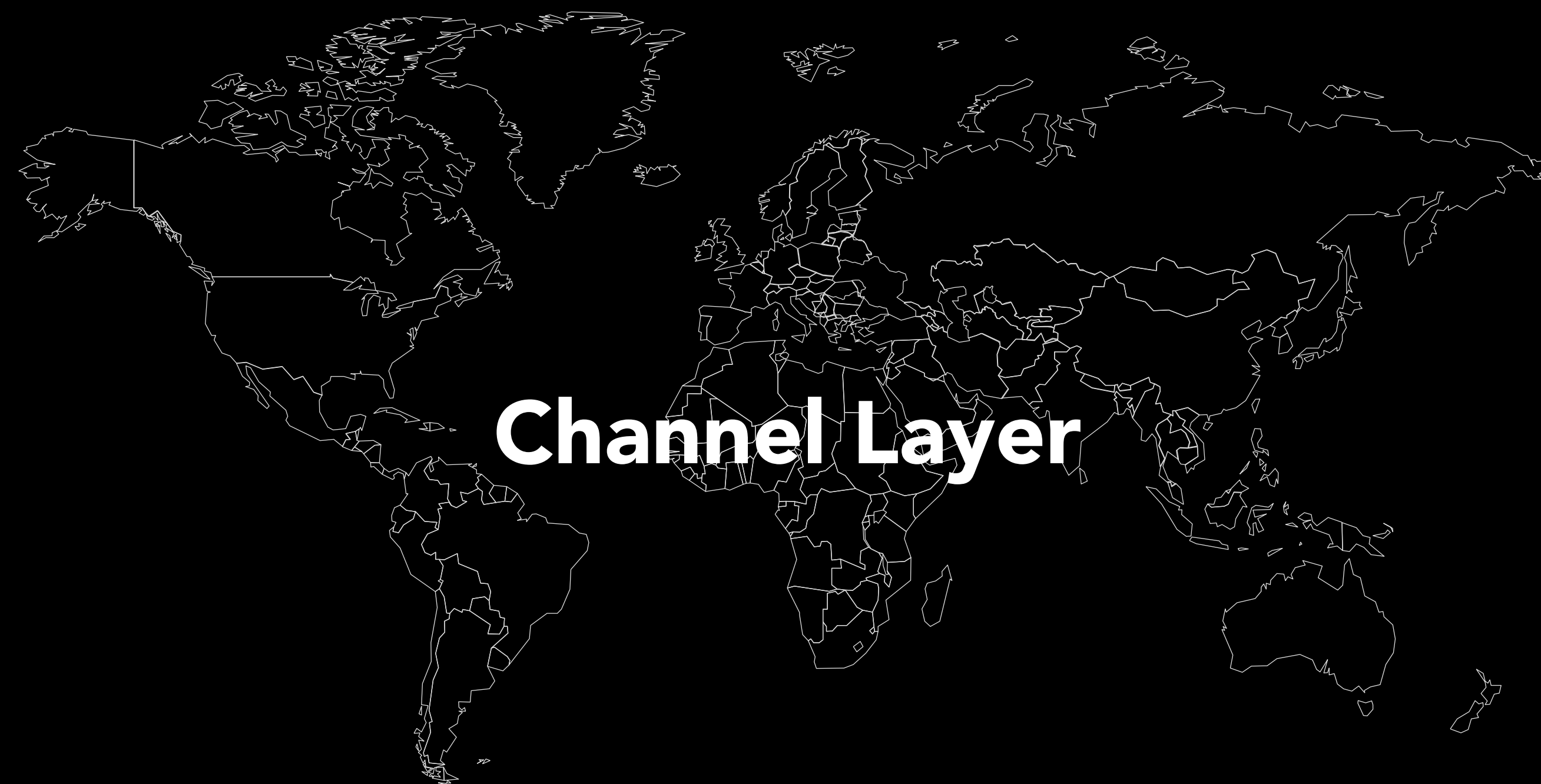
# Channel Layer





Channel





**Channel Layer**



```
41 ①↑ def send(self, channel, message):
42      # Make sure the message is a dict at least (no deep inspection)
43      assert isinstance(message, dict), "Message is not a dict"
44      # Channel name should be text
45      assert self.valid_channel_name(channel), "Channel name not valid"
46      # Add it to a deque for the appropriate channel name
47      with self.thread_lock:
48          queue = self._channels.setdefault(channel, deque())
49          if len(queue) >= self.capacity:
50              | raise self.ChannelFull(channel)
51          queue.append((
52              | time.time() + self.expiry,
53              | message,
54              ))
```



```
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 --
def receive_many(self, channels, block=False):
    # Check channel names
    assert all(self.valid_channel_name(channel) for channel in channels), "C
    # Shuffle channel names to ensure approximate global ordering
    channels = list(channels)
    random.shuffle(channels)
    # Expire old messages
    self._clean_expired()
    # Go through channels and see if a message is available:
    with self.thread_lock:
        for channel in channels:
            if self._channels.get(channel, None):
                _, message = self._channels[channel].popleft()
                return channel, message
    # No message available
    return None, None
```



```
73 74 75 76 77 78 79 80 81 82
def new_channel(self, pattern):
    assert isinstance(pattern, six.text_type)
    assert pattern.endswith("!=") or pattern.endswith("?"), "New channel
    # Keep making channel names till one isn't present.
    while True:
        random_string = "".join(random.choice(string.ascii_letters) for
        new_name = pattern + random_string
        # Basic check for existence
        if new_name not in self._channels:
            return new_name
```



# Handling Protocols





Extension

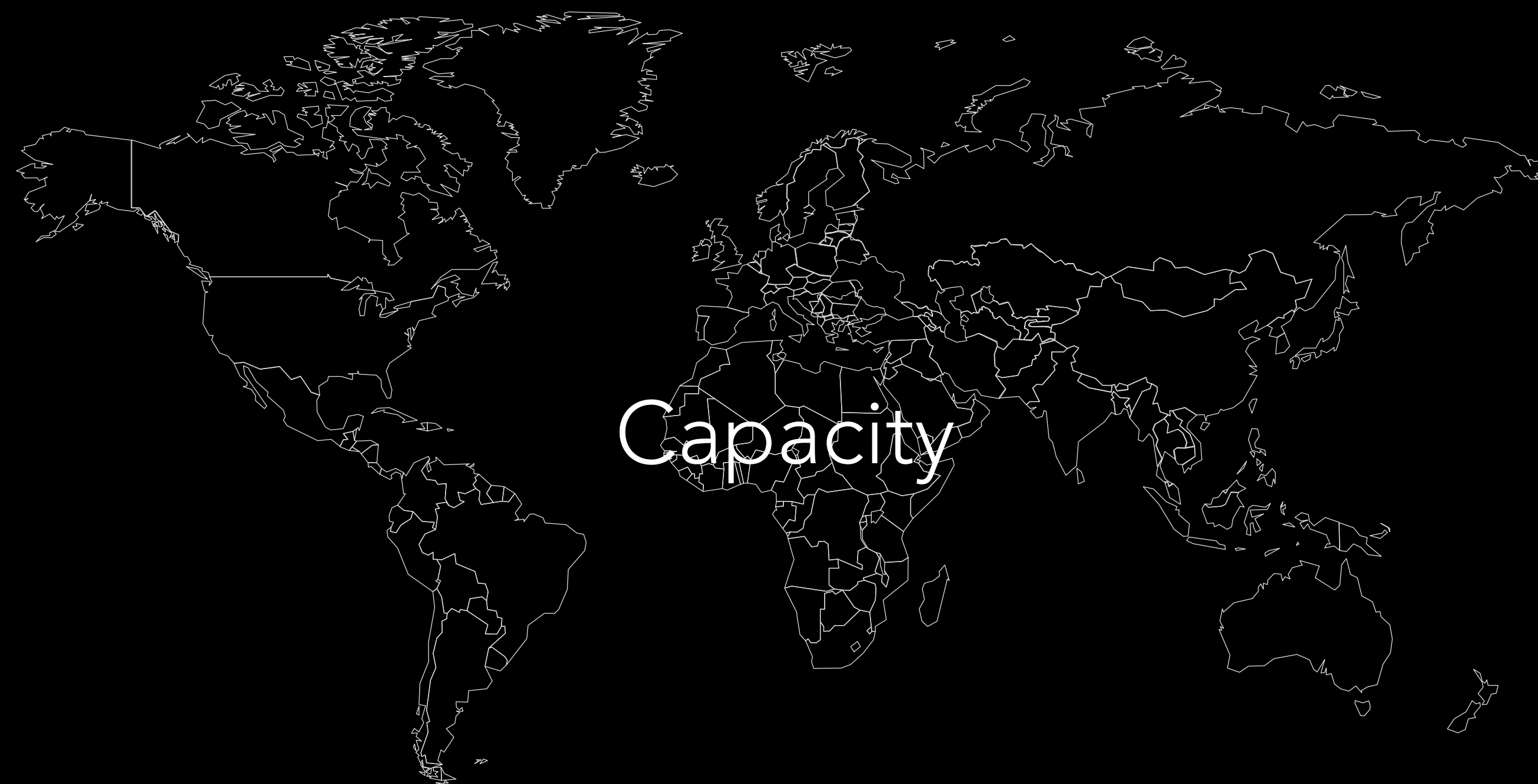


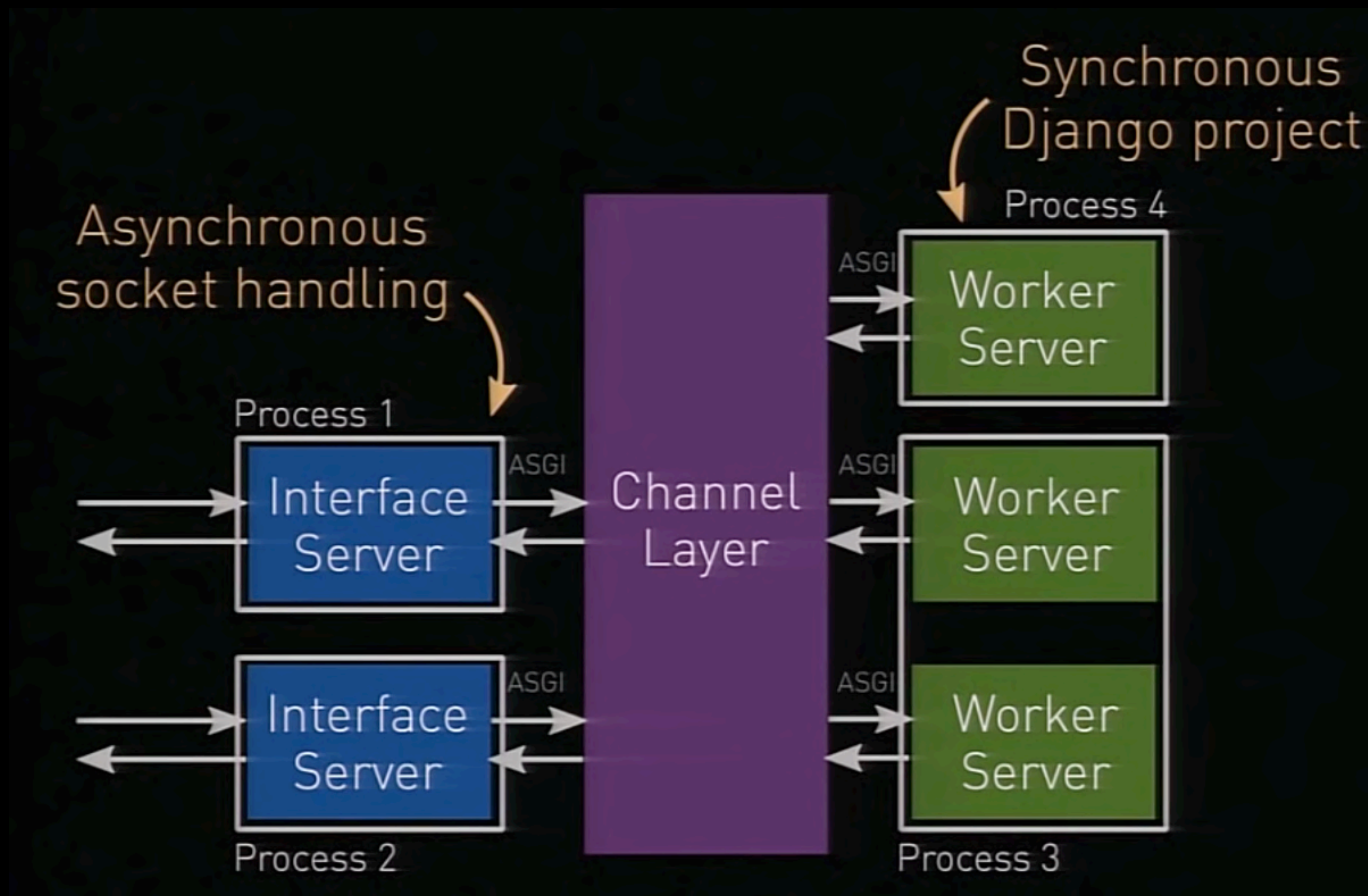


# Shared Nothing Architecture

## 无共享架构









Interface  
Server

Daphne  
HTTP + WS

WSGI Adapter  
HTTP

Channel  
Layer

asgi\_redis  
Cross-network, shards

asgi\_ipc  
Single-machine

Worker  
Server

Django  
Consumer system

WSGI Adapter  
Most WSGI apps



# 能用来做些什么？（AKA 有哪些坑）

- 能显著提高开发效率（基础代码的迁移成本降低）
- 需要更强力的 interface server（Daphne 暂不牢靠）
- 需要特殊的 Scaling 技巧
- 需要和 WSGI 和谐共处
- .....



**Django Channels**



Documentary / Specification





Ernest 何世友