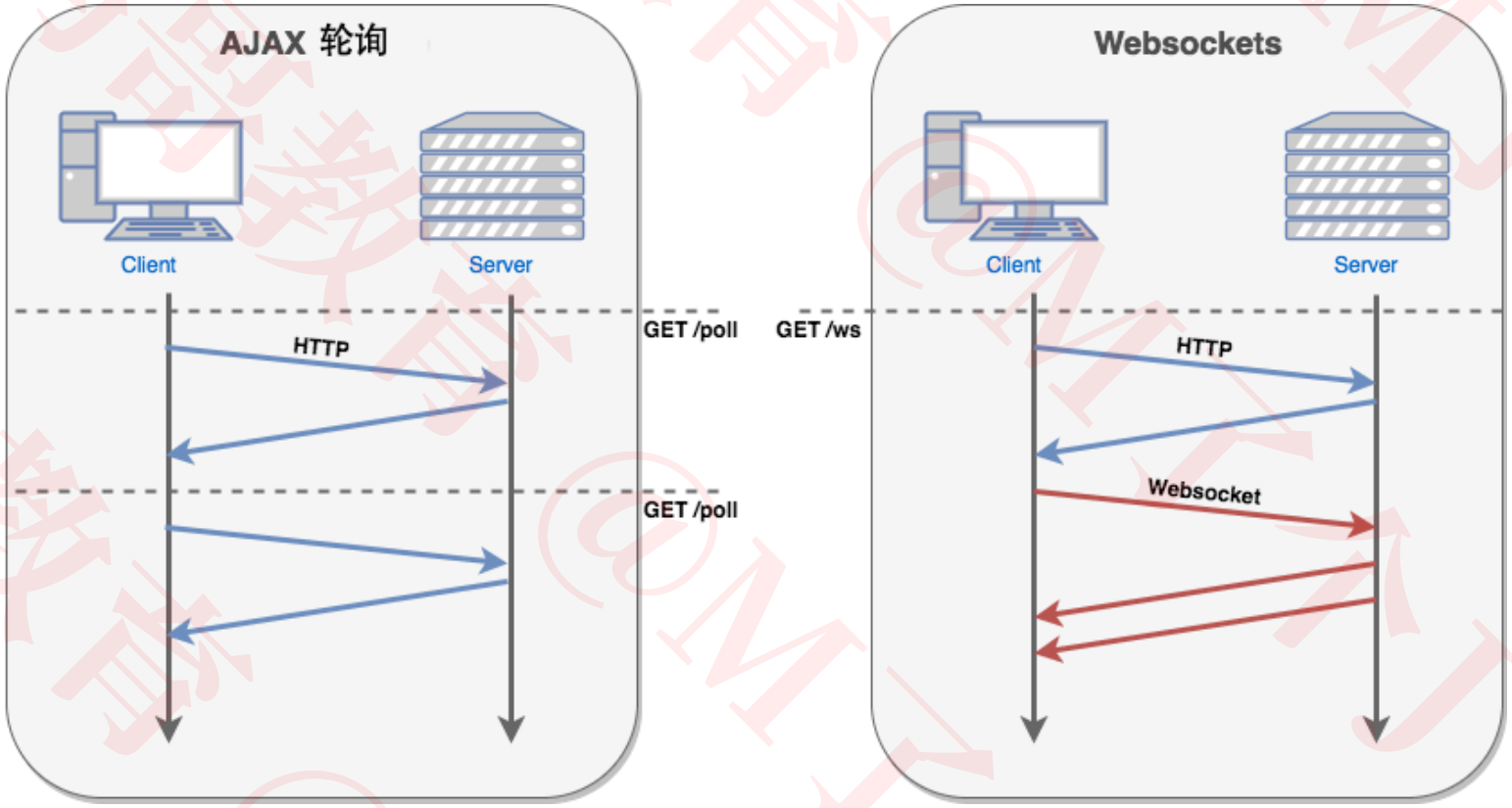
HTTP vs WebSocket

- HTTP请求的特点：通信只能由客户端发起。所以，早期很多网站为了实现推送技术，所用的技术都是轮询
  - 轮询是指由浏览器每隔一段时间（如每秒）向服务器发出HTTP请求，然后服务器返回最新的数据给客户端
  - 为了能更好的节省服务器资源和带宽，并且能够更实时地进行通讯，HTML5规范中出现了WebSocket协议



WebSocket

- WebSocket，是基于TCP的支持**全双工**通信的应用层协议
  - 在2011年由IETF标准化为RFC 6455，后由RFC 7936补充规范
  - 客户端、服务器，任何一方都可以主动发消息给对方

- WebSocket的应用场景很多
  - 社交订阅、股票基金报价、体育实况更新、多媒体聊天、多玩家游戏等

HTTP vs WebSocket

- WebSocket和HTTP属于平级关系，都是应用层的协议
  - 其实TCP本身就是支持全双工通信的（客户端、服务器均可主动发消息给对方）
  - 只是HTTP的“请求-应答模式”限制了TCP的能力

- WebSocket使用80（ws://）、443（wss://）端口，可以绕过大多数防火墙的限制
  - ws://example.com/wsapi
  - wss://secure.example.com/wsapi

- 与HTTP不同的是，WebSocket需要先建立连接
  - 这就使得WebSocket成为一种有状态的协议，之后通信时可以省略部分状态信息
  - 而HTTP请求可能需要在每个请求都额外携带状态信息（如身份认证等）

WebSocket – 建立连接

- WebSocket需要借助HTTP协议来建立连接（也叫作握手，Handshake）
  - 由客户端（浏览器）主动发出握手请求

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

- Connection必须设置Upgrade
  - 表示客户端希望连接升级
- Upgrade必须设置websocket
  - 表示希望升级到WebSocket协议
- Sec-WebSocket-Version
  - 表示支持的Websocket版本
  - RFC 6455要求使用的版本是13

WebSocket – 建立连接

- Sec-WebSocket-Key是客户端生成的随机字符串，比如例子中的dGhlIHNhbXBsZSBub25jZQ==

- 服务器接收到客户端的Sec-WebSocket-Key后，会进行以下操作

① Sec-WebSocket-Key加上一个固定的GUID值（258EAFA5-E914-47DA-95CA-C5AB0DC85B11）

- dGhlIHNhbXBsZSBub25jZQ==258EAFA5-E914-47DA-95CA-C5AB0DC85B11

② 将①的结果进行SHA-1摘要计算

- b37a4f2cc0624f1690f64606cf385945b2bec4ea

③ 将②的结果进行Hex To Base64编码

- s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

④ 将③的结果做为Sec-WebSocket-Accept响应头的值，返回给客户端

- 如此操作，可以尽量避免普通HTTP请求被误认为WebSocket协议

WebSocket – 使用

- WebSocket体验和演示
  - https://www.websocket.org/echo.html

- W3C标准化了一套WebSocket API，可以直接使用JS调用

```
let ws = new WebSocket('wss://example.com')
```