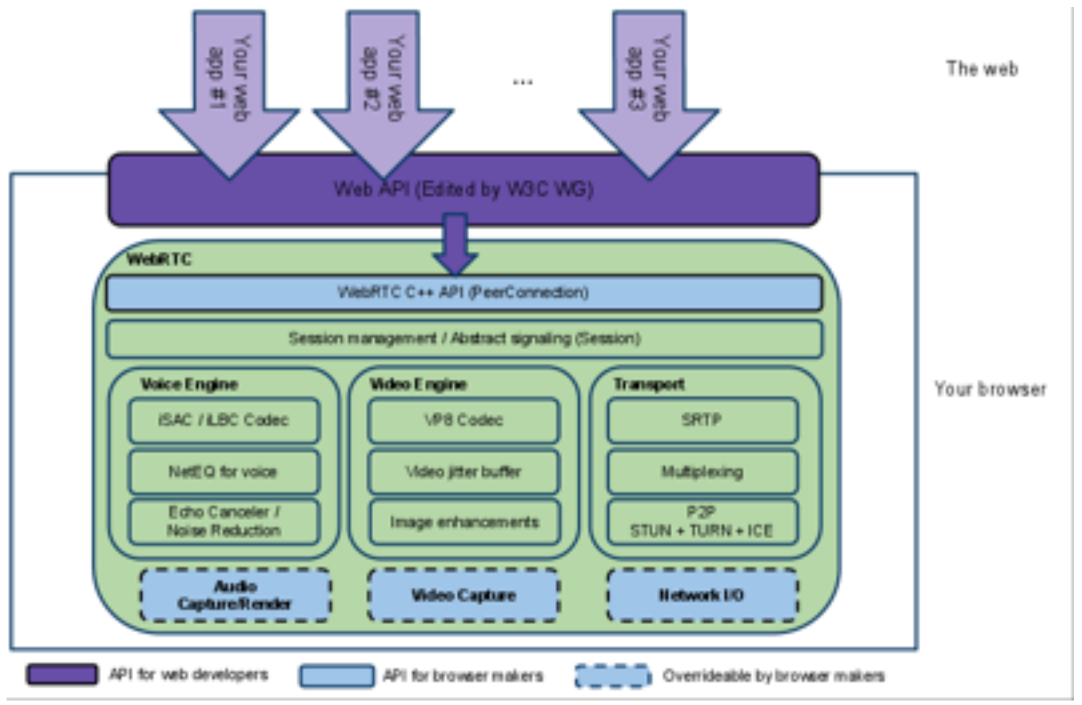
WebRTC and Media Delivery

“WebRTC (Web Real-Time Communication) is an API definition drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling, video chat, and P2P file sharing without the need of either internal or external plugins”

http://en.wikipedia.org/wiki/WebRTC

WebRTC



FOSS WebRTC

Core project is open source:
http://www.webrtc.org/ under the BSD license

Full Codec Stack:
Google spent $192M in 2010 to acquire two companies, and open sourced its codecs:

- Global IP - iLBC and iSAC
- On2 Technologies - VP8

OpenH264 by Cisco

WebRTC

Data Channels API

New protocol stack in the browser

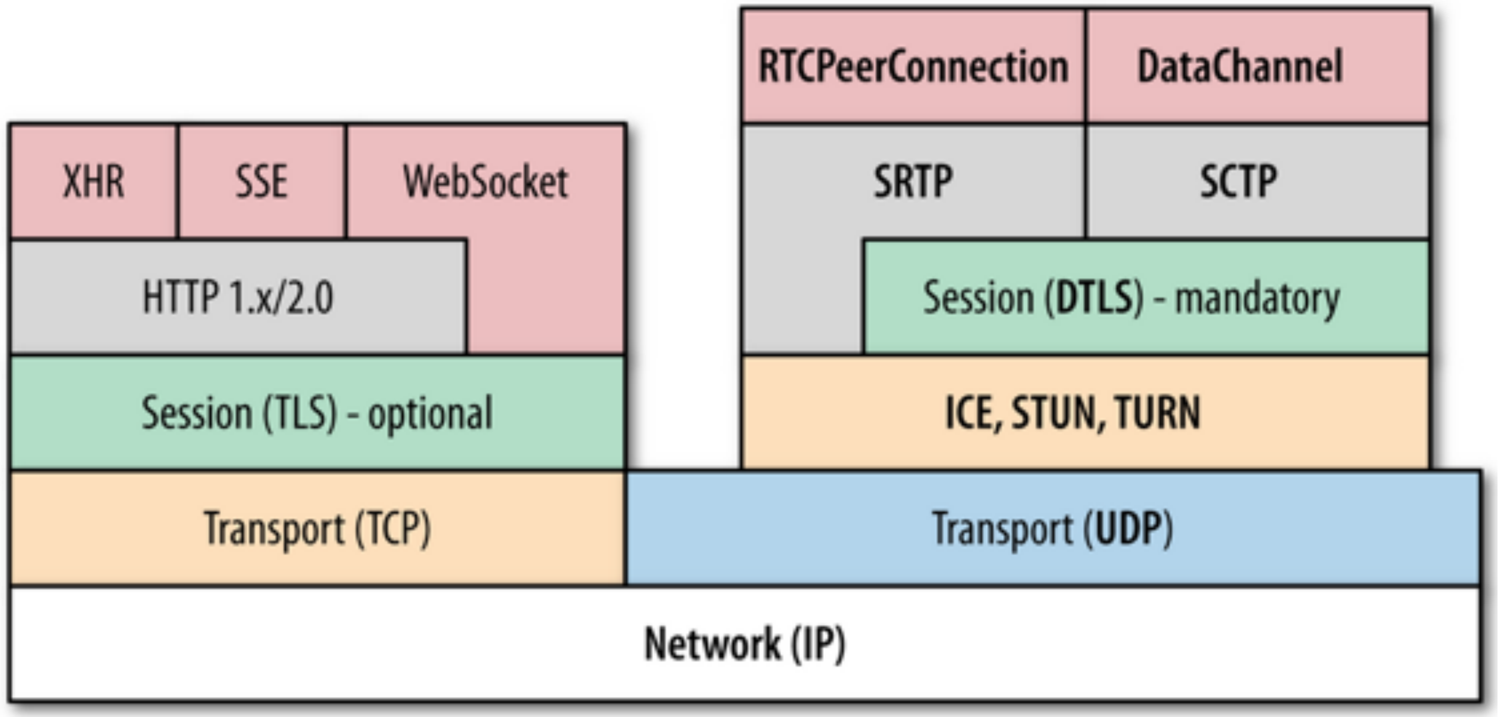


Figure 18-3. WebRTC protocol stack

PEER5

Why it is huge?

Decentralized

The web meant to work in a distributed manner

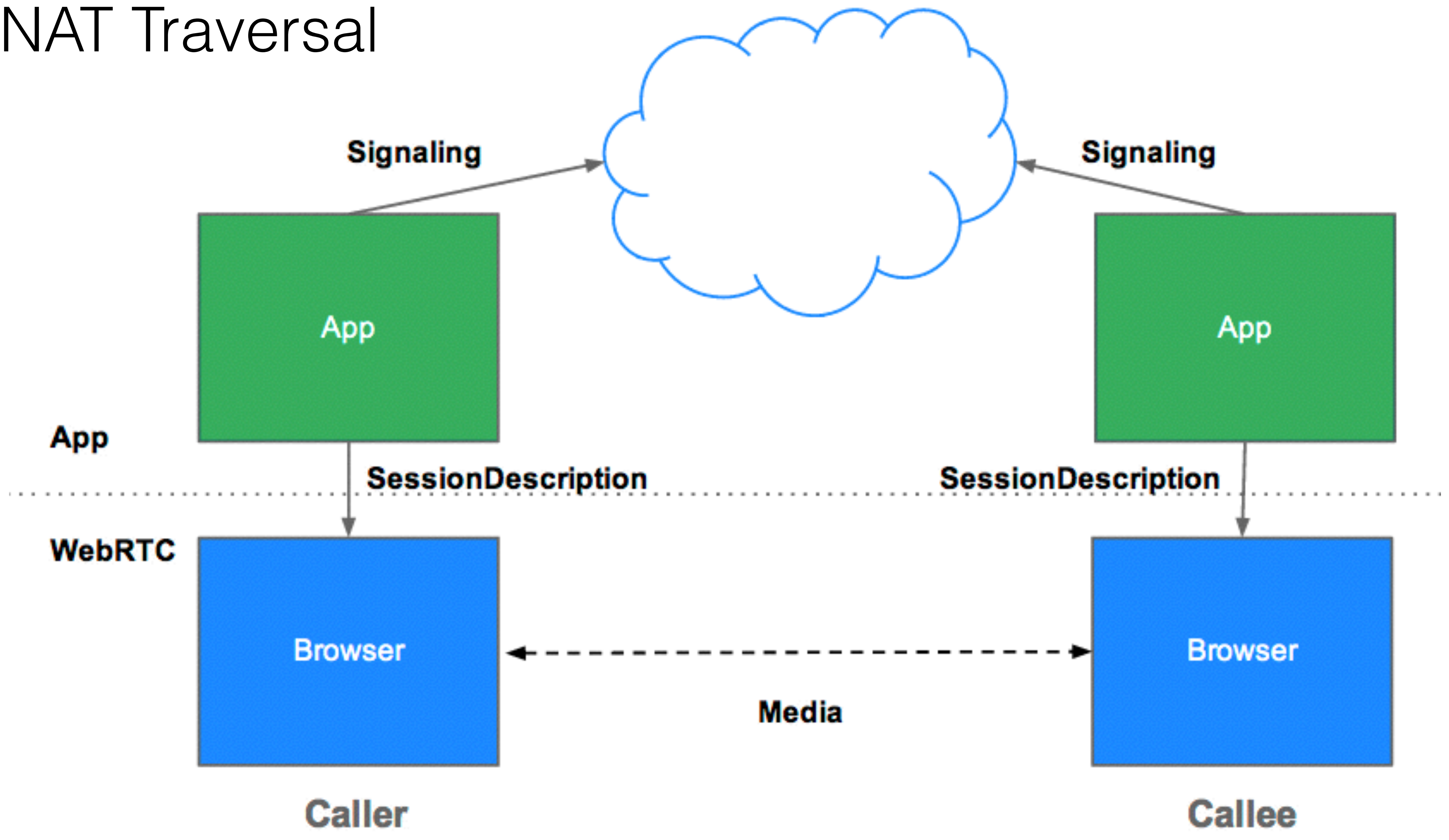
Centralized

In reality we are quite centralized due to the fact:
web browsers can only communicate with servers
(HTTP, WebSockets)

WebRTC will (hopefully) make the web fully distributed

How does it work

NAT Traversal



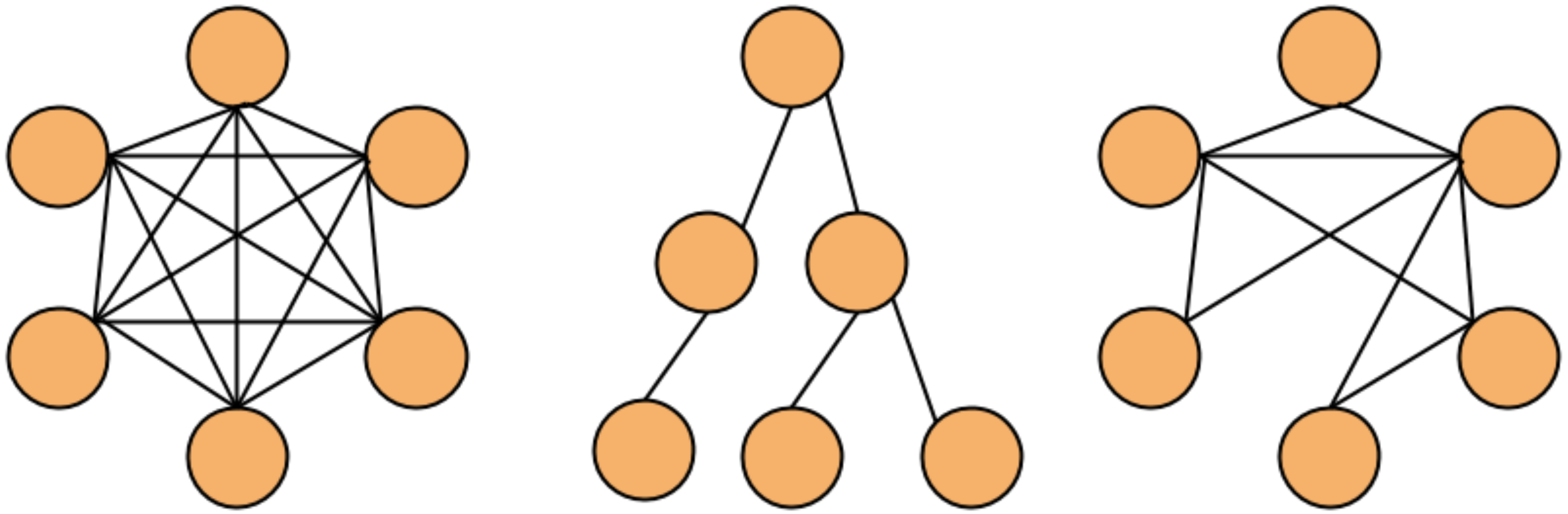
What can we do with it?

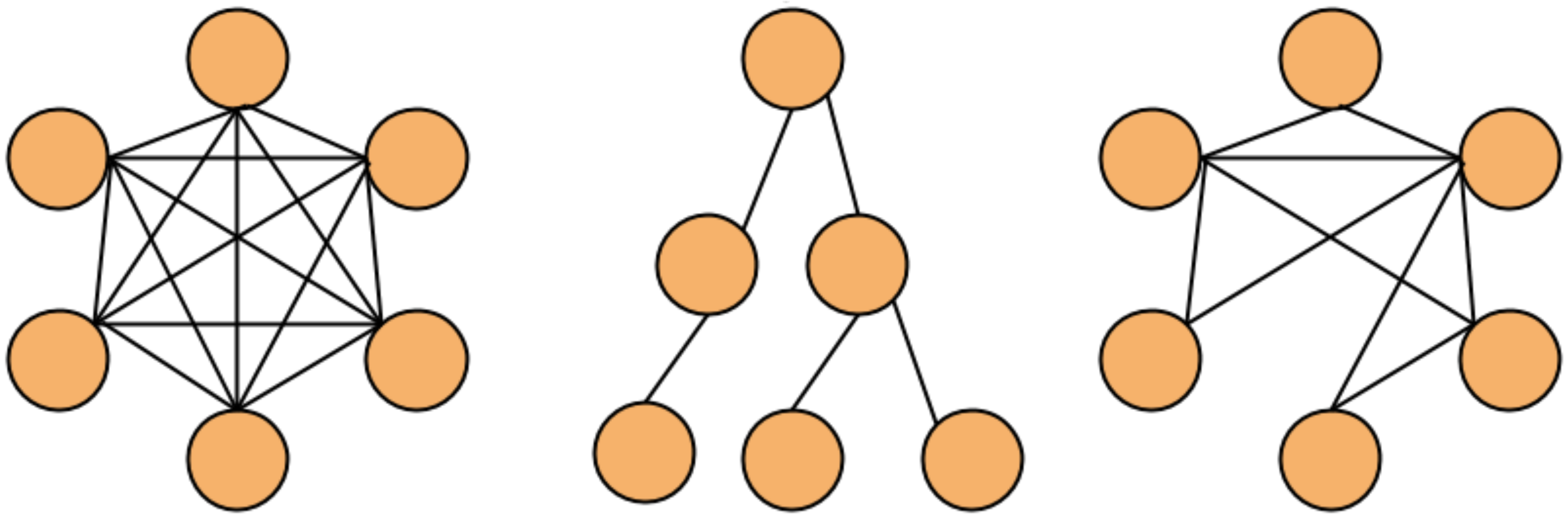
PEER5

Send files, send metadata, chat (1-to-1)

Mesh networks

Good for rich content, media





Mesh use case

Games (cubeslam, mozilla games)
Video (tokbox, room.co, bem.tv)
Audio and radio
Filesharing (sharefest.me, webtorrent)

Peer assisted delivery

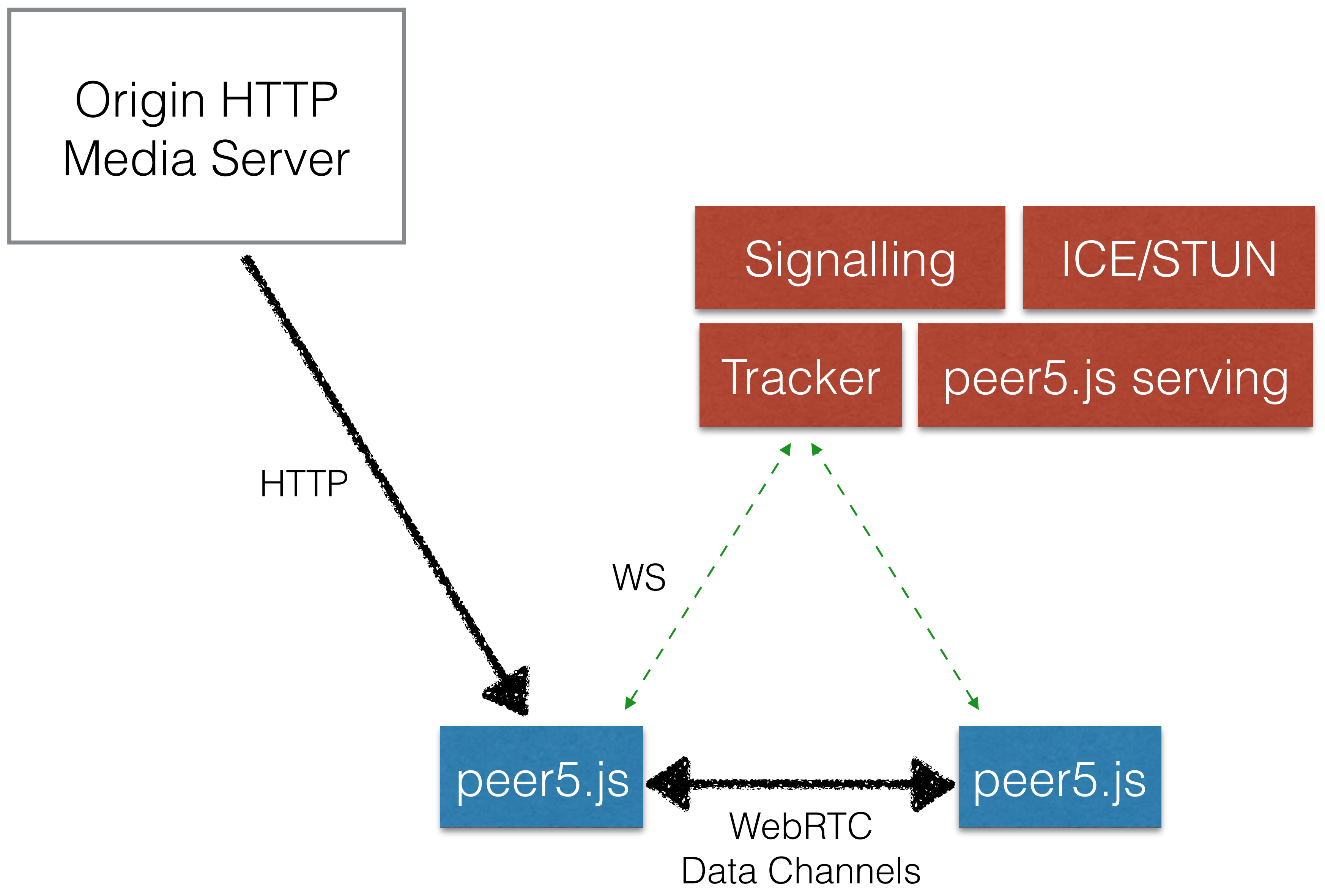
Servers are still helpful and the two technologies complete each other very well:

- Infinitely scalable
- Resilient
- Faster, lower latency
- Cost effective

Building a WebRTC CDN

How do you build a highly dynamic network and keep it simple?
- Needs to be easy to use, SaaS
- Agnostic to existing webservers and CDNs
- Work in many different use cases
- Secured
- Fast
- Scalable
- **FOSS helps a lot, saves a lot of work**

High-Level Architecture

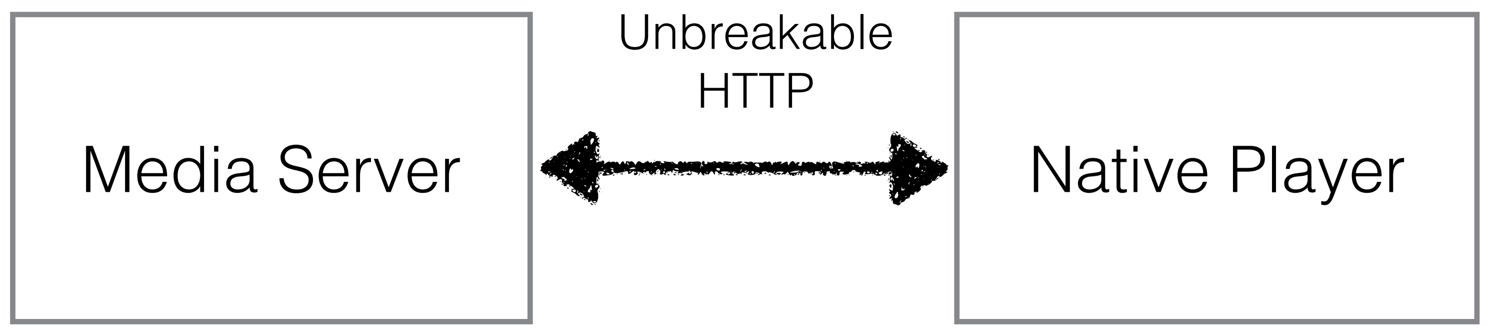


node.js https://github.com/websockets/ws

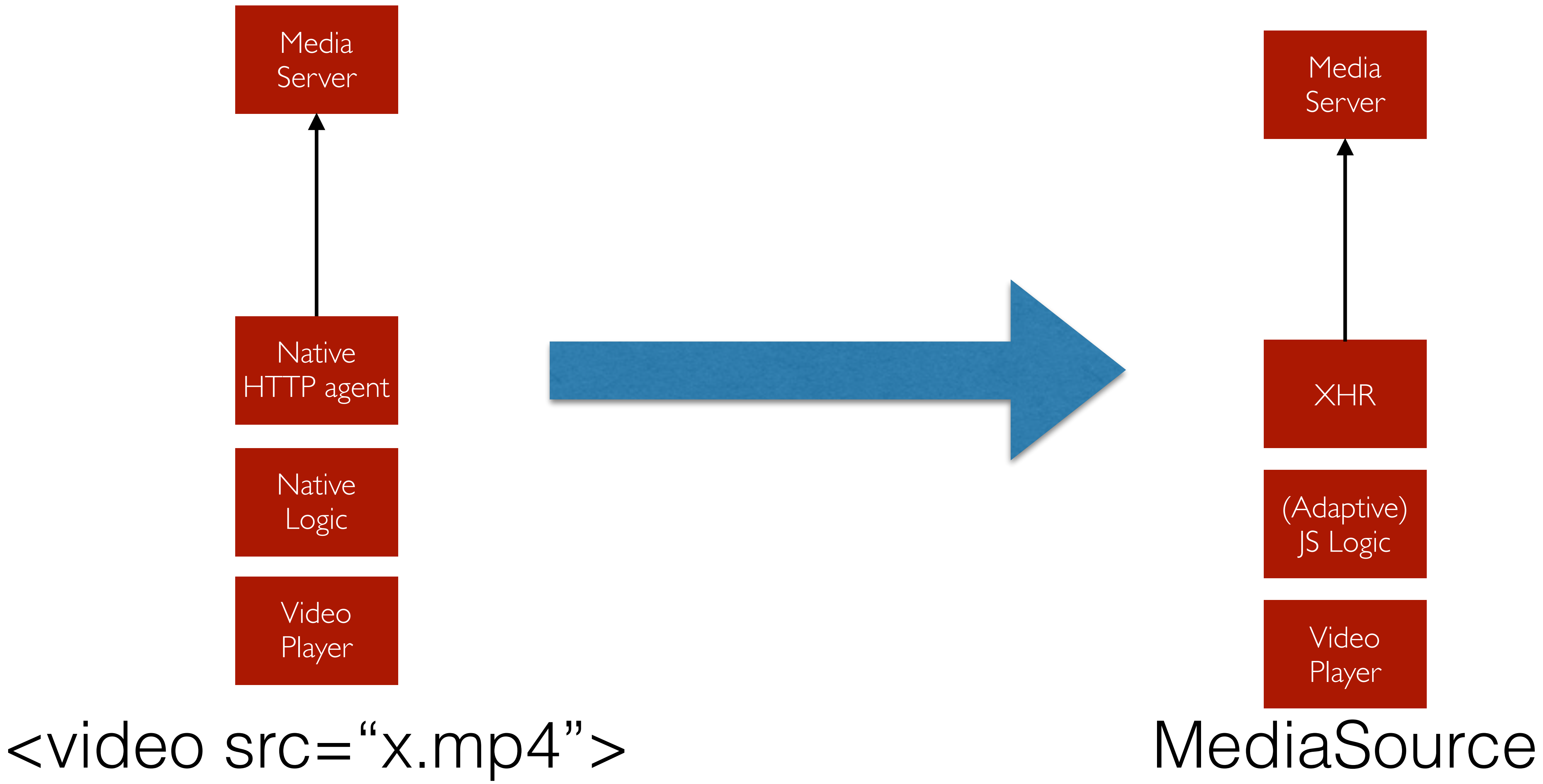
DEMO

The (our) problem with video players today

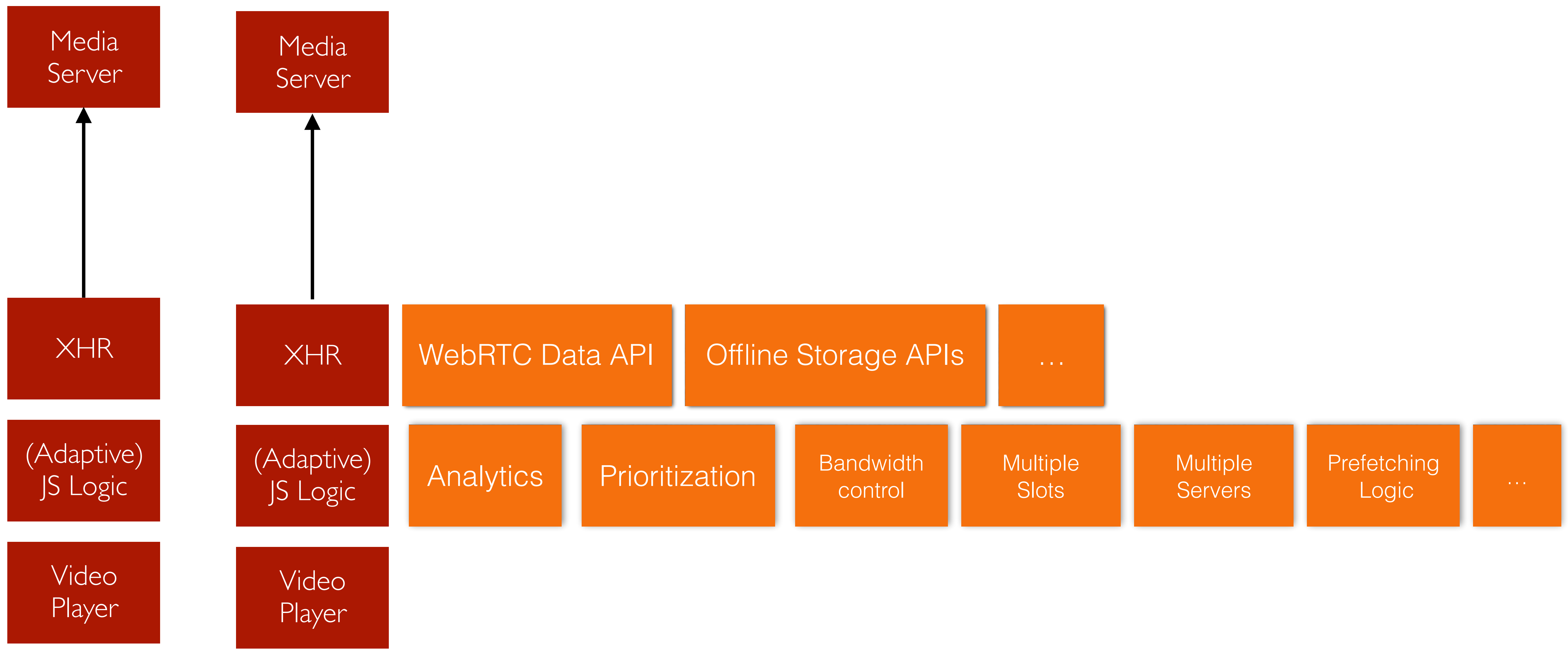
like <video src=x.mp4>



JS-based delivery



js-based delivery



PEER5

JS-based delivery

A more distributed approach that adds power to the client
The client can cache content smarter, measure and decide on better what to fetch, when and from where.

Enablers:
MediaSource Extensions API - power to the people!
DASH:dash.js, mp4box.js,
Flash and HLS: flashls, clappr.io, video.js (JW, Kaltura soon)
MP4->HLS: https://github.com/kaltura/nginx-vod-module
HTML5 and HLS: MSE-HLS

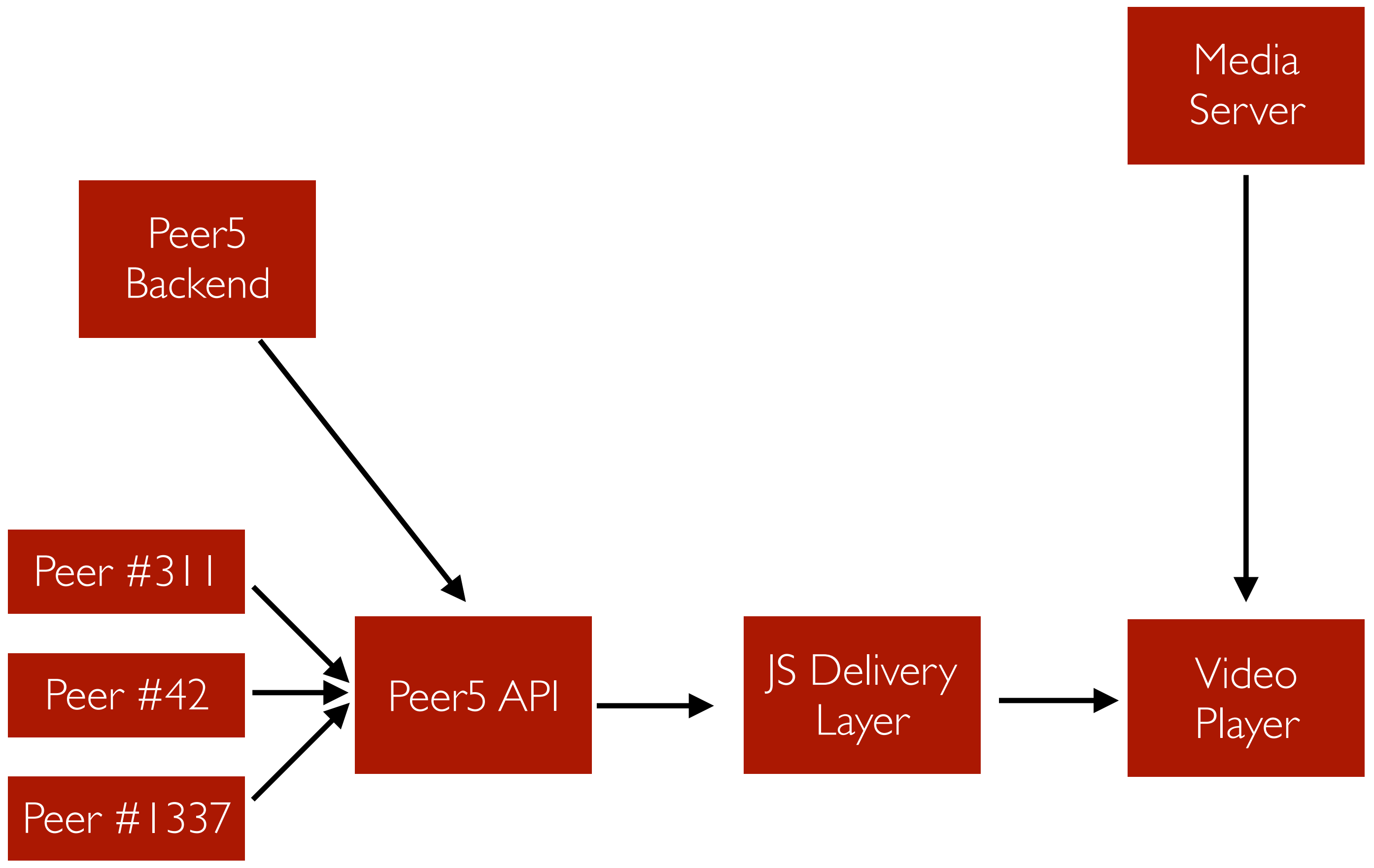
The Peer5 API

XMLHTTPRequest Compliant

```
var request = new peer5.Request();
request.open(“GET",url);
request.onload = function(e){
 …
}
```

https://github.com/Peer5/P2PXHR

Summary


Media Server
Peer5 Backend
Peer #311
Peer #42
Peer #1337
Peer5 API
JS Delivery Layer
Video Player

Thanks

http://peer5.com