

# NETFLIX

## FE System Design

### Things TO COVER In Netflix System Design

Requirements

Scoping

Tech choices

Component Arch.

DATA API & protocols  
Implementation

### ① Requirements

Functional

Module level thinking

- Supply

- Video upload
- Analytics
- Meta Info
- Tagging

- Demand

- multi-user management
- Pricing & subscription

- Account management.

- Help & support

- Auth module

- Movie / Series catalog

- Detail Page

- Watch List

- Reviews

Feature Level Thinking

- Homepage

- language change

- sign/signup

- config UI

- Catalog

- search

- card review

- banner review

- multi user manager

- switching bw users

- parental control.

- video player

- speed

- quality

- language

- subtitle

- thumbnails over view

- review like/dislike comments.

## Non-functional

- Mobile / Desktop

- Scrolling

- Responsive

- Device / Location

- Asset Optimization

(Video, Image, CSS, JS)

- Resource Hints

- Open Graph Tags

- Deep linking

- Performance

- CSR / SSR

- Auth

- Security

- HTTP2 / HTTP3

- Caching

Offline

PLA

- A/B Testing

- Versioning

→ Internationalization & Localization

→ Testing

## SCOPING

- Lang. change
- Sign in / Signup
- config UI
- switching b/w users
- speed
- quality
- language
- subtitle
- thumbnail view

## Scanning

- Asset optimization
- HTTP 2 / HTTP3
- Internationalization
- Localization

## Tech Stack

### Library / Framework

MicroFrontend

Mono Repo

Design System

Build & tools

Dependencies

reactJS

TypeScript

RxJS

Restify

apollo / GraphQL

Lattice - module - federation

(Leona)

Hawkins

webpack

(IMX) React

Image blobs / sprite / SVG

Video tag / Media sources

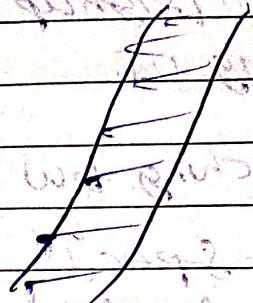
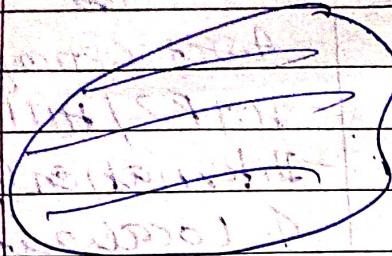
HTTP 1.1 / 2 / 3

Scanning

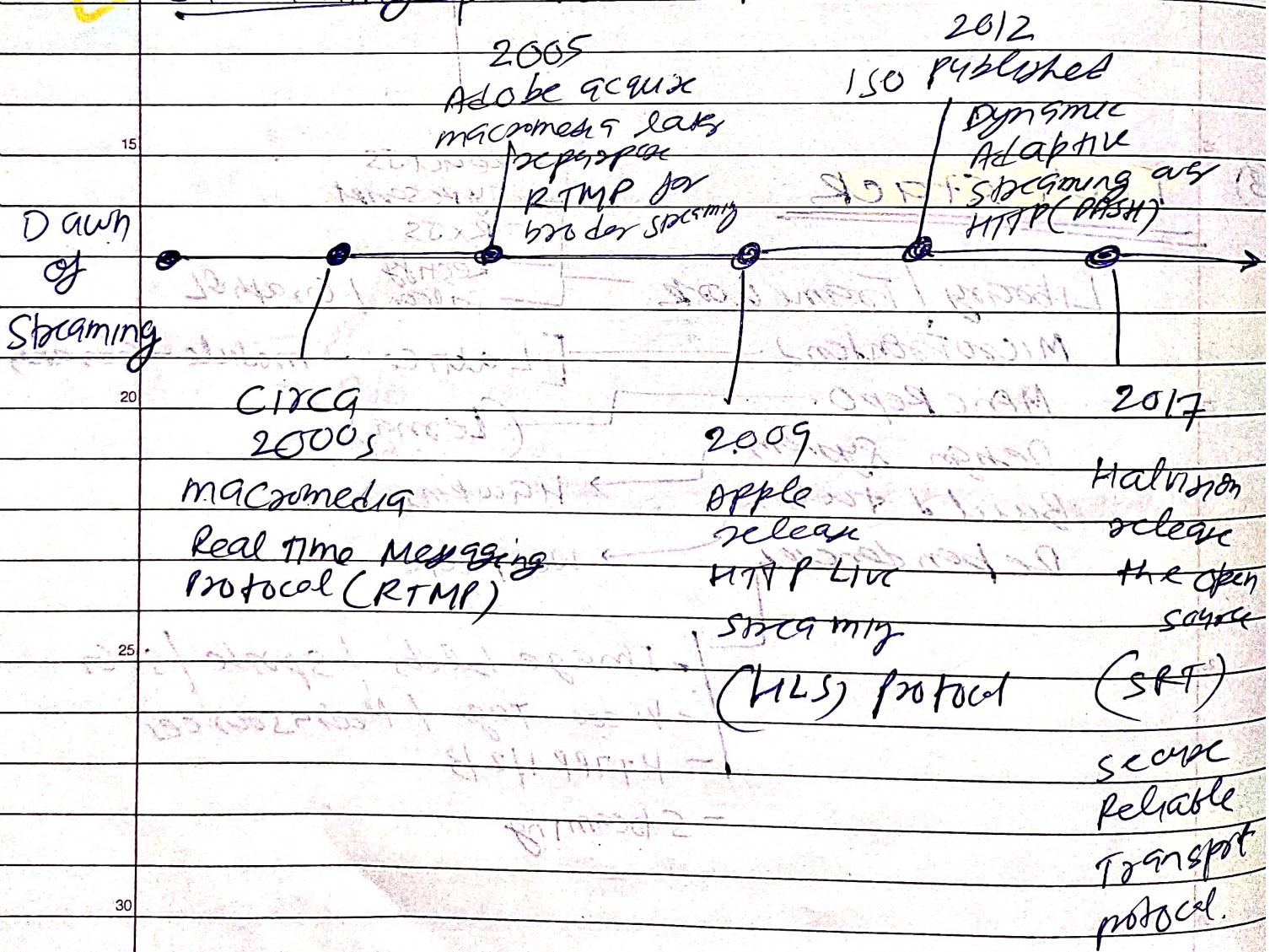
Eg

## ✓ Downloading Vs Streaming

A lake & a stream

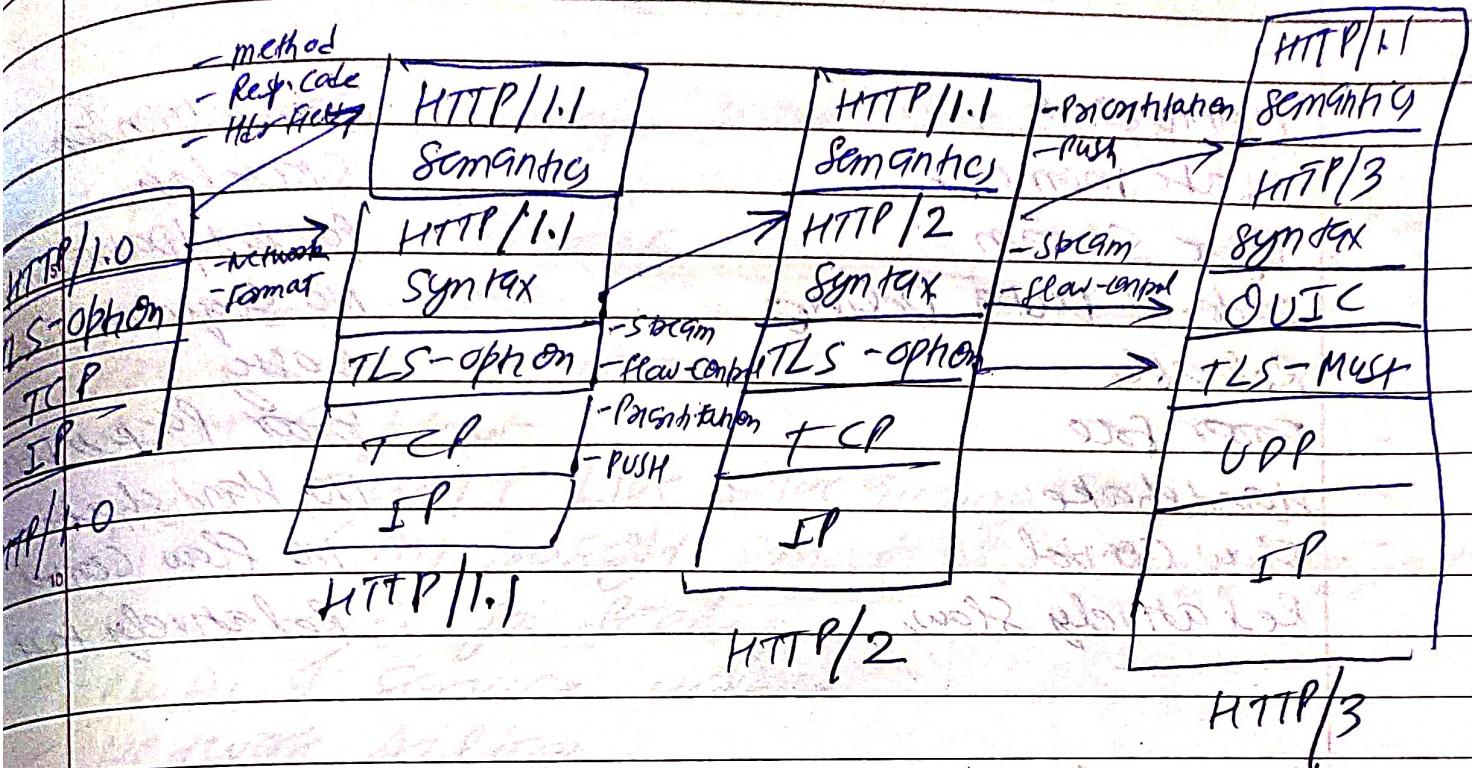


## ✓ Streaming protocol timeline

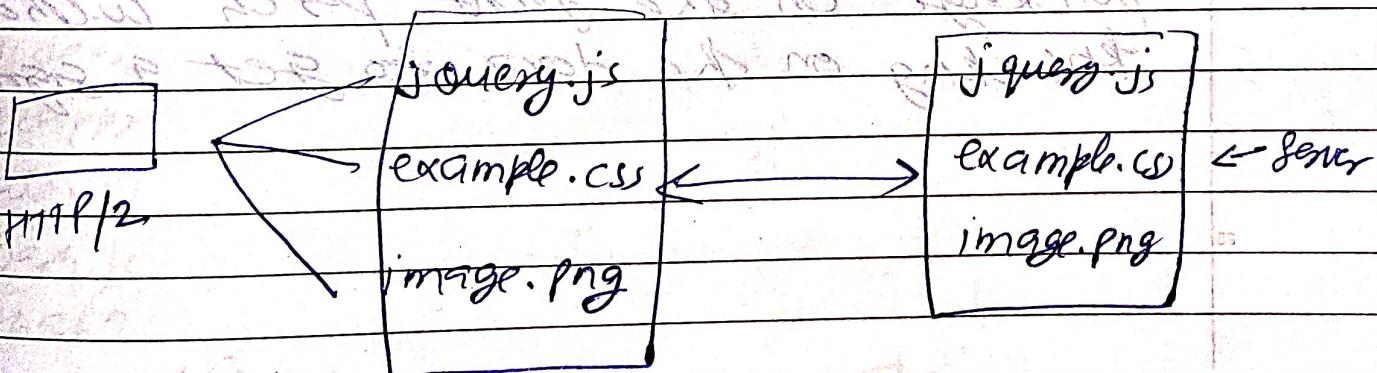
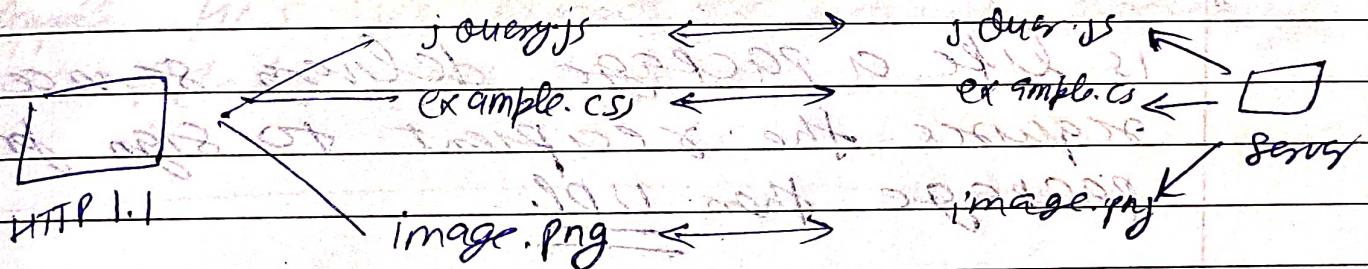


# HTTP Protocol Comparison

Camlin Page  
Date / /



HTTP connection



TCP

VS

UDP

Connected

State memory

Byte Stream

ordered data delivery

Reliable

Error free

Handshake

Flow Control

Relatively slow

connectionless

stateless

Packet/Datagram

No Sequence, Guar.

Lossy

Error Packet Disc.

No Handshake

No flow control

Relatively fast

if TCP

is like a package delivery service that requires the recipient to sign for the package than UDP.

is like a delivery service that leaves package on the front porch without knocking on the door to get a signature.

## Challenges with player

### (Video Tag + MediaSource)

- automatically jumps gap in content.
- support for strong content offline, including protected content.
- push activation
- Pre-Load feature
- cast support.

10 HLS (HTTP Live Streaming)

11 M3U8 (Microsoft Smooth Streaming) support.

- thumbnails for seeking
- PWAs & service workers
- network filters
- Retros

15 Apple HLS vs Microsoft Smooth Streaming  
vs Adobe HDS support

2000 of 069

10 minutes | 120s

MOZ -

## Player

videos.js

clappr

HLS.js

Dash.js

Shaka

medias Element

Fluid Player

DPlayer

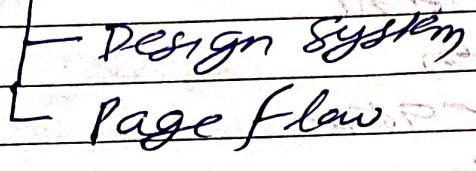
Plyr

open PlayerJS

(4)

## Component Design

- Skeleton Visualization
- component Hierarchy



- Service

- Routing

- Data sharing

—x— 50:53 / 1:03:17

(6) Upgrade

(5)

## Data API & Implementation

### Protocols

- Rest / GraphQL
- JSON

20

### Implementation Details

- Pagination / Infinite scrolls
- Debouncing / Throttling
- Video Streaming
- Configurable UI
- Previews
- SSR
- Image optimization

30

## Data Modeling

- URL
- Method
- Request (query, body)
- Response (Data, Error)
- Status Code

## Component API

- State / Props
- Event handling
- Customization (Theming)
- Reusability

## API's

- getGenres()

- getVideos()

- getVideoDetails()

- UserPersonalization()

uiLocale

appVersion

uiVersion

userAgent

bandwidth

3