

OTT STREAMING

FUNDAMENTALS

COGICOM



Clement Duval

ABOUT YOUR INSTRUCTOR

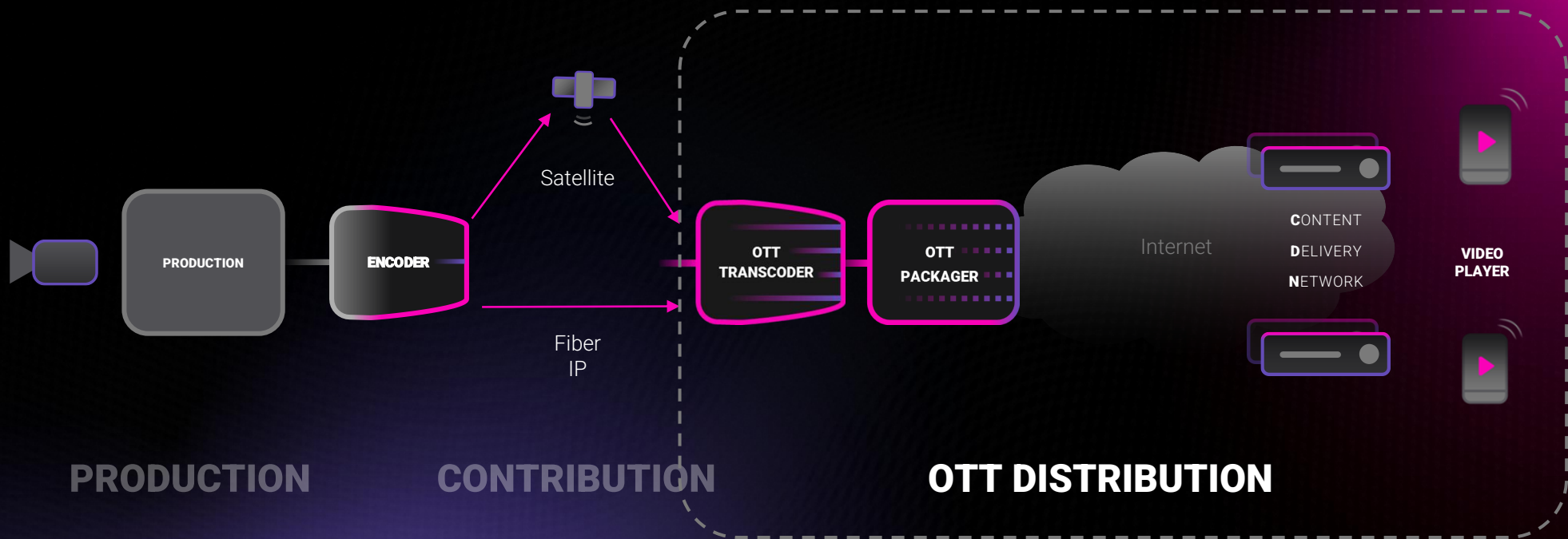


- ▶ *Founder at Video Stream Stack*
- ▶ Any questions? Write me at clement@videostreamstack.com
- ▶ Let's connect on [LinkedIn](#)

COGICOM

ABOUT YOUR INSTRUCTOR

OVER-THE-TOP ORIGINATION



OVER-THE-TOP ORIGINATION



OTT DISTRIBUTION

UNDERSTANDING OVER-THE-TOP

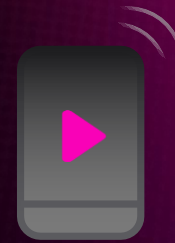
IPTV vs OTT

COGICOM

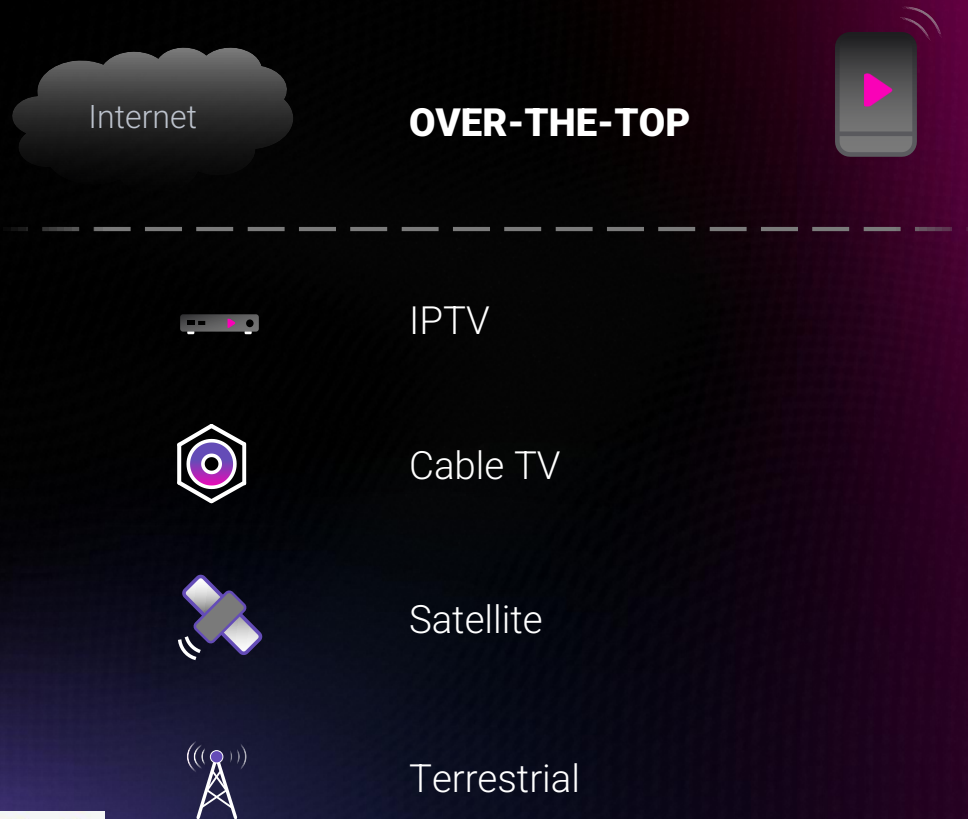
OVER-THE-TOP



OVER-THE-TOP



OVER-THE-TOP



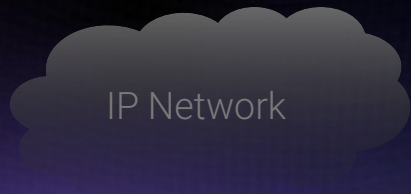
OTT STREAMING MODEL



OTT



Versus



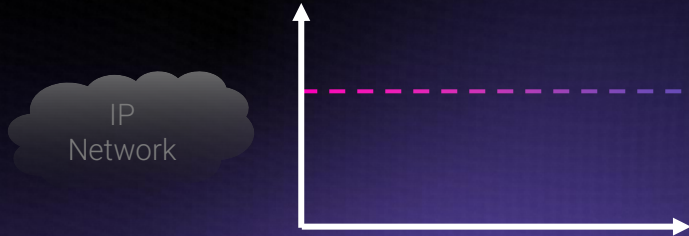
IPTV



IPTV vs OTT: THE TRANSMISSION NETWORKS

IPTV

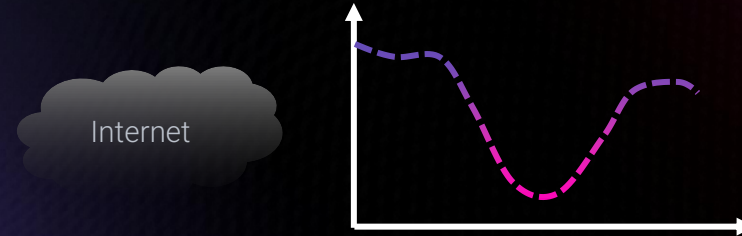
- ▶ Privately **managed network**
- ▶ Bandwidth is constant & guaranteed



COGICOM

OTT

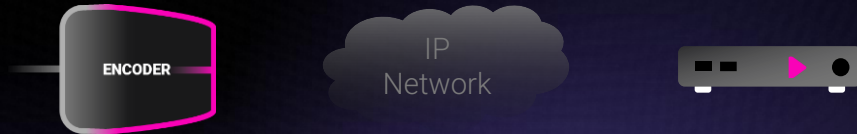
- ▶ **Unmanaged** public network
- ▶ Bandwidth fluctuates/not guaranteed



IPTV vs OTT: THE TRANSMISSION

IPTV

▶ One-way **UDP** transmission



OTT

▶ Two-way **TCP** transmission



IPTV vs OTT: INTEROPERABILITY

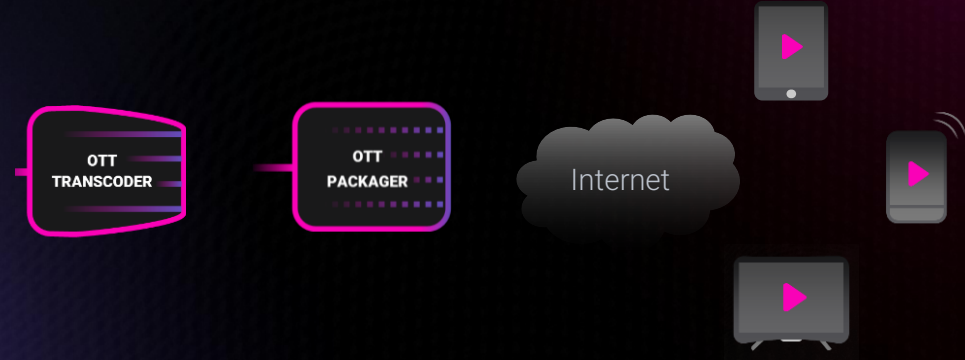
IPTV

- ▶ Known hardware capabilities
- ▶ Easy interoperability



OTT

- ▶ Any hardware device
- ▶ Complex interoperability



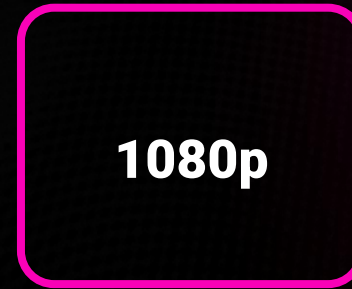
IPTV vs OTT: DISPLAY MODE

IPTV



- ▶ Interlaced (historical reasons)
- ▶ To remain compatible with old CRT devices
- ▶ De-interlacer needed to display on modern devices

OTT



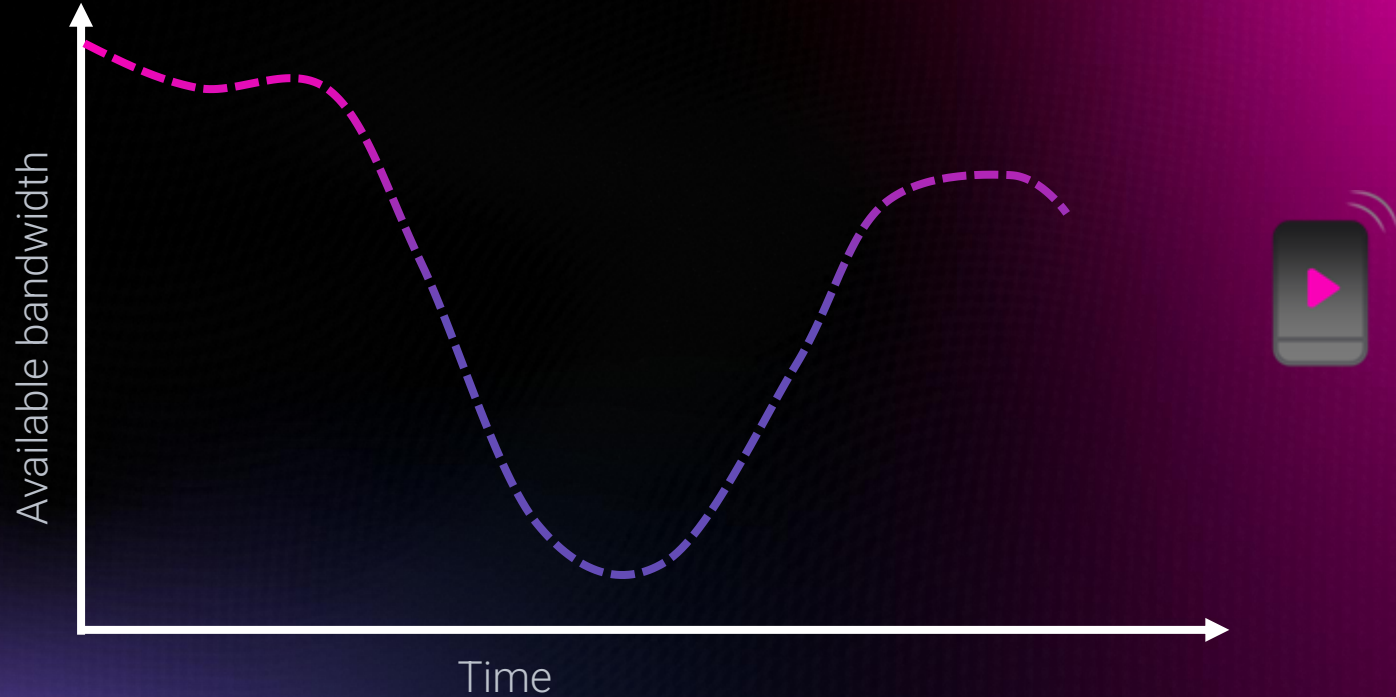
- ▶ Progressive display
- ▶ Suited for modern OTT displays

CHALLENGES OF INTERNET STREAMING

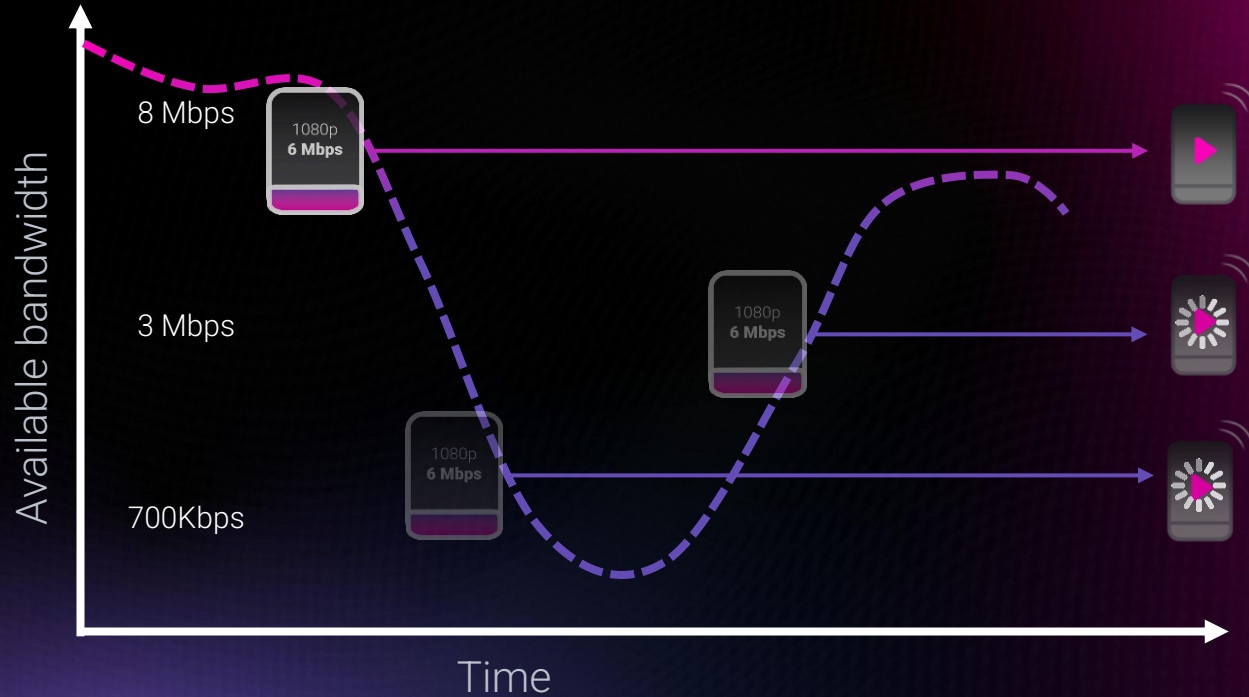
PLAYER DRIVEN ADAPTIVE STREAMING

COGICOM

CHALLENGES OF STREAMING OVER THE PUBLIC INTERNET



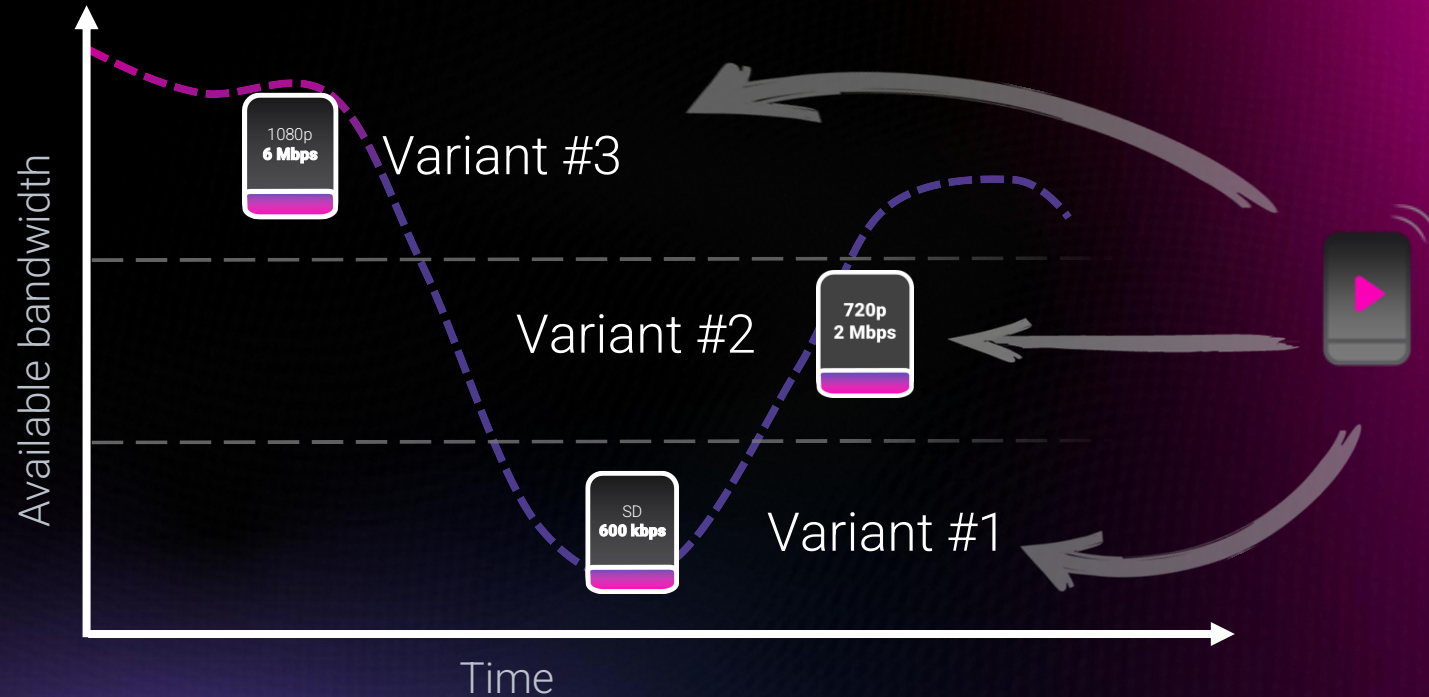
LET'S STREAM OVER OTT USING UDP



CREATING RENDITIONS OF THE SAME CONTENT



ADAPTIVE STREAMING OVERCOMES BANDWIDTH CHANGES



OTT PLAYER CHOOSES VARIANT BASED ON CPU



THE OTT PLAYER HAS AN ACTIVE ROLE
IT CHOOSES THE APPROPRIATE VARIANT ACCORDING TO THE
AVAILABLE BANDWIDTH
& ITS CAPABILITIES



Quiz 1

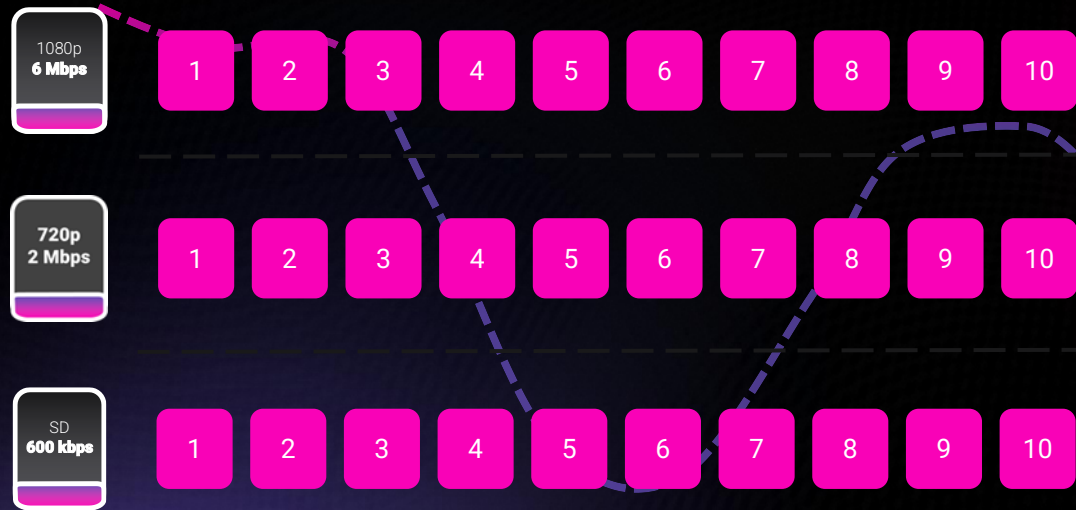
COGICOM

OTT & SEGMENT STREAMING

& HTTP TRANSFERS

COGICOM

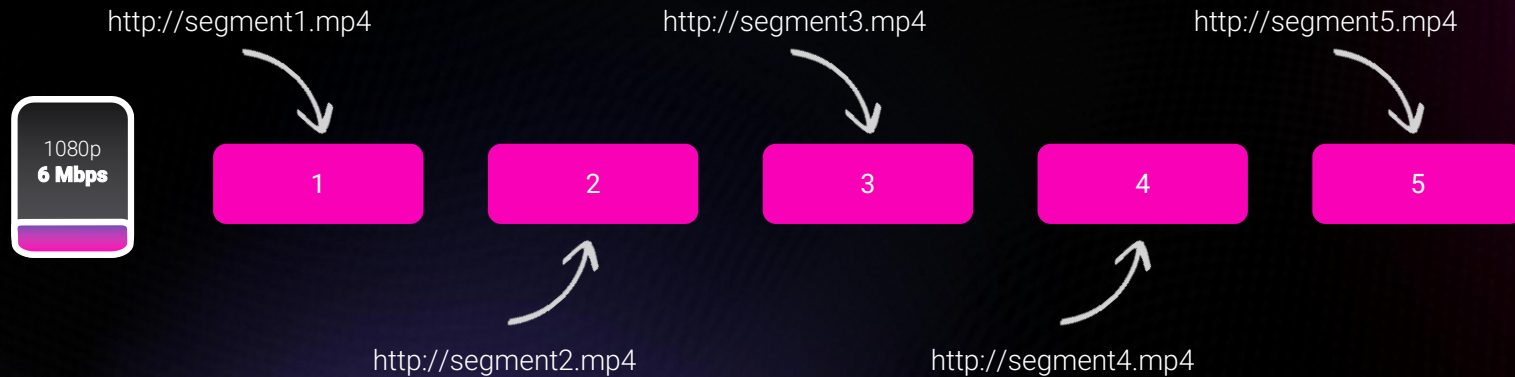
CHOPPING THE STREAM INTO INDEPENDENT SEGMENTS



TO ENSURE SEAMLESS SWITCHING BETWEEN VARIANTS
EACH SEGMENT SHOULD BE
DECODABLE INDEPENDENTLY



CHOPPING THE STREAM INTO INDEPENDENT SEGMENTS



Each segment is downloaded as a file via HTTP

**OTT STREAM ARE NO LONGER CONSIDERED
AS A CONTINUOUS FLOW OF DATA,
BUT RATHER AS A SERIES OF FILE TRANSFERS
DOWNLOADED OVER HTTP**



STREAMING USING HTTP: WHY HTTP?

- ▶ HTTP/S is a well-known protocol, default in firewalls
- ▶ Runs over TCP, reliable transmission which retransmits lost packets
- ▶ Streaming is a series of file transfers
- ▶ Each file is downloaded using an **HTTP address** (URL)

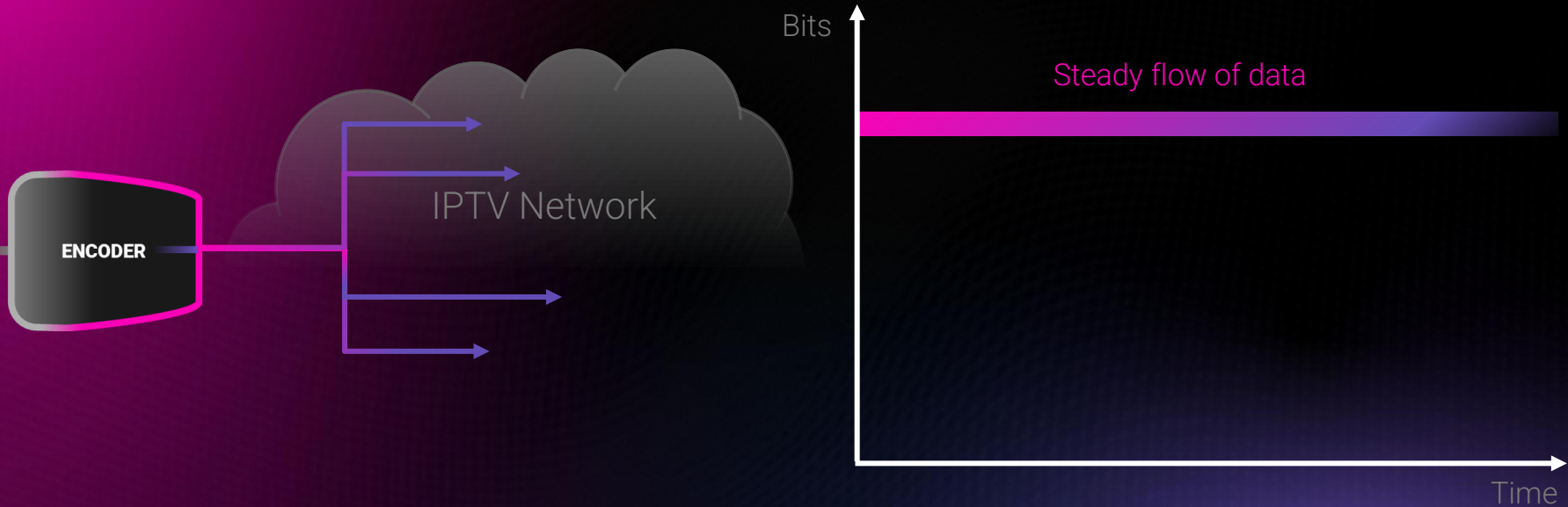
`http://origin_server/segment1.mp4`

`http://origin_server/segment2.mp4`

`http://origin_server/segment3.mp4`

TRADITIONAL vs HTTP ADAPTIVE STREAMING

UDP STREAMING



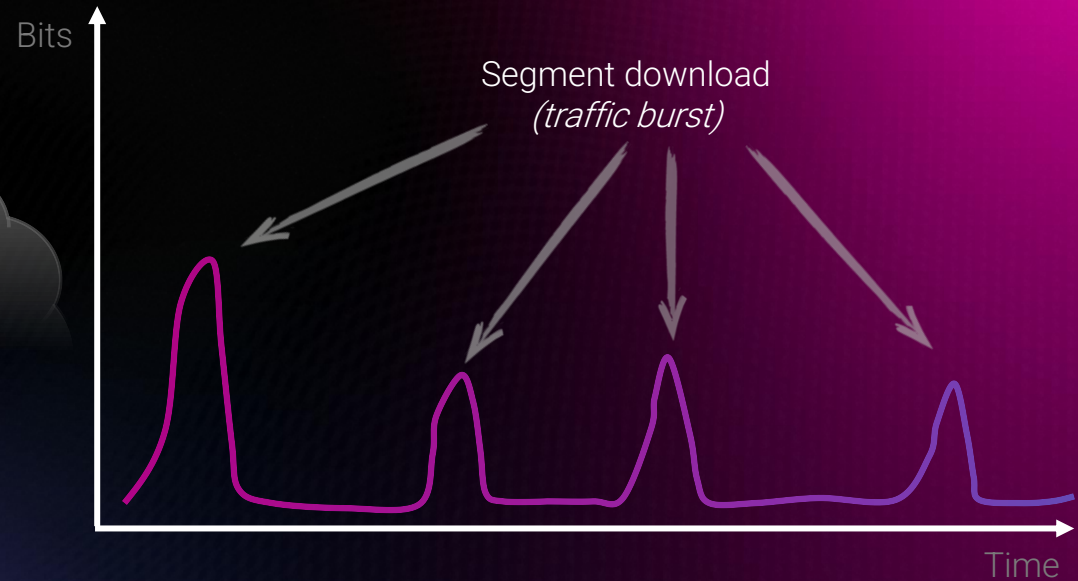
COGICOM

HTTP ADAPTIVE STREAMING

TRADITIONAL vs HTTP ADAPTIVE STREAMING



HTTP/TCP STREAMING

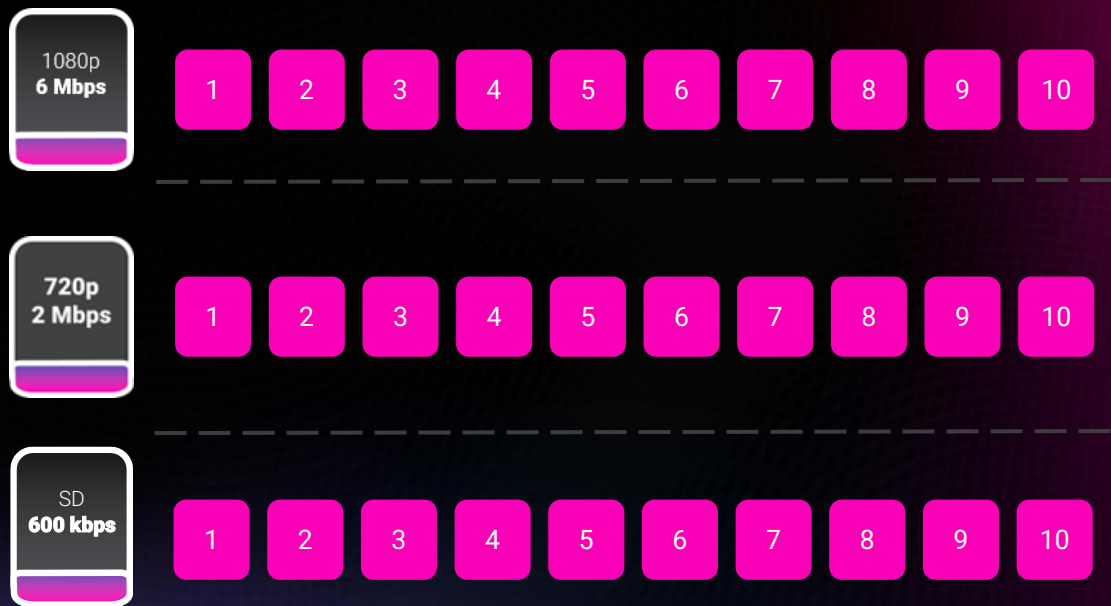


THE OTT MANIFEST

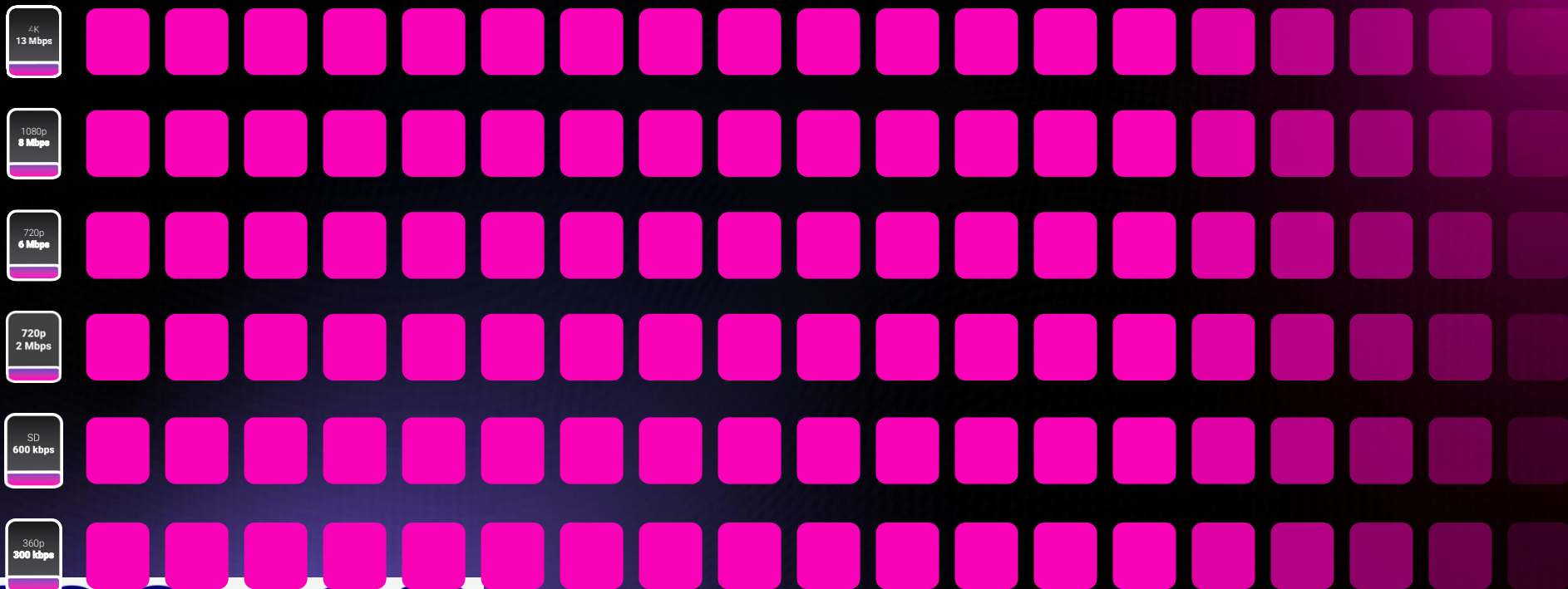
& THE PACKAGED OTT STREAM

COGICOM

THE NEED FOR AN INDEX FILE



THE NEED FOR A MANIFEST FILE



THE MANIFEST PROVIDES URL FOR MEDIA SEGMENTS

Variant_1: http://segment_1_1 http://segment_1_2 http://segment_1_3

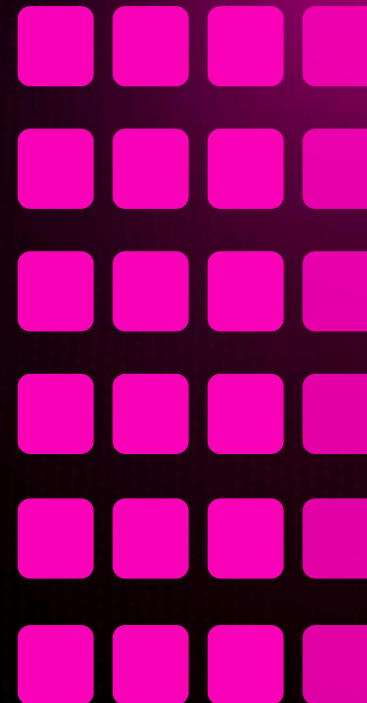
Variant_2: http://segment_2_1 http://segment_2_2 http://segment_2_3

Variant_3: http://segment_3_1 http://segment_3_2 http://segment_3_3

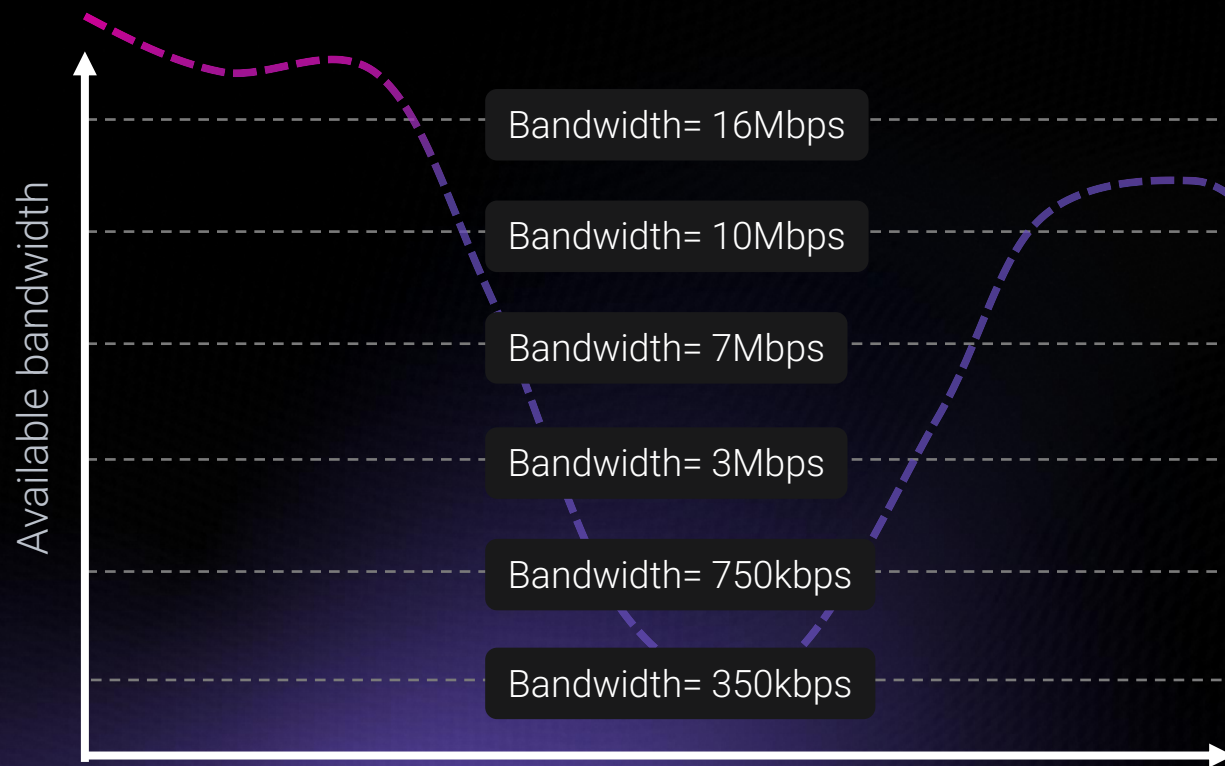
Variant_4: http://segment_4_1 http://segment_4_2 http://segment_4_3

Variant_5: http://segment_5_1 http://segment_5_2 http://segment_5_3

Variant_6: http://segment_6_1 http://segment_6_2 http://segment_6_3



THE MANIFEST PROVIDES URL FOR MEDIA SEGMENTS



Variant_1: http://segment_1_1

Variant_2: http://segment_2_1

Variant_3: http://segment_3_1

Variant_4: http://segment_4_1

Variant_5: http://segment_5_1

Variant_6: http://segment_6_1

THE OTT MANIFEST FILE PROVIDES PLAYERS WITH:

- **THE BANDWIDTH VALUES ASSOCIATED TO EACH VARIANT**
- **THE DOWNLOAD URL FOR MEDIA SEGMENTS FOR EACH VARIANT**



THE OTT PACKAGED STREAM

Variant_1: http://segment_1_1 http://segment_1_2 http://segment_1_3

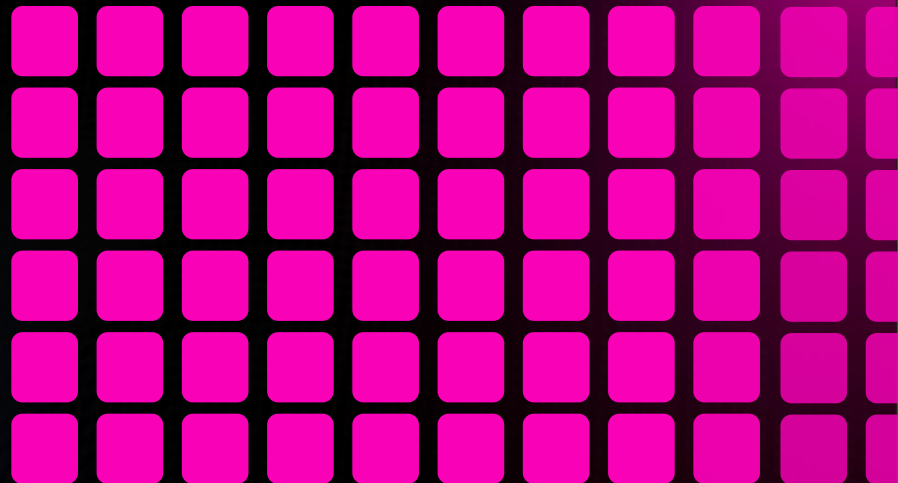
Variant_2: http://segment_2_1 http://segment_2_2 http://segment_2_3

Variant_3: http://segment_3_1 http://segment_3_2 http://segment_3_3

Variant_4: http://segment_4_1 http://segment_4_2 http://segment_4_3

Variant_5: http://segment_5_1 http://segment_5_2 http://segment_5_3

Variant_6: http://segment_6_1 http://segment_6_2 http://segment_6_3



Quiz 2

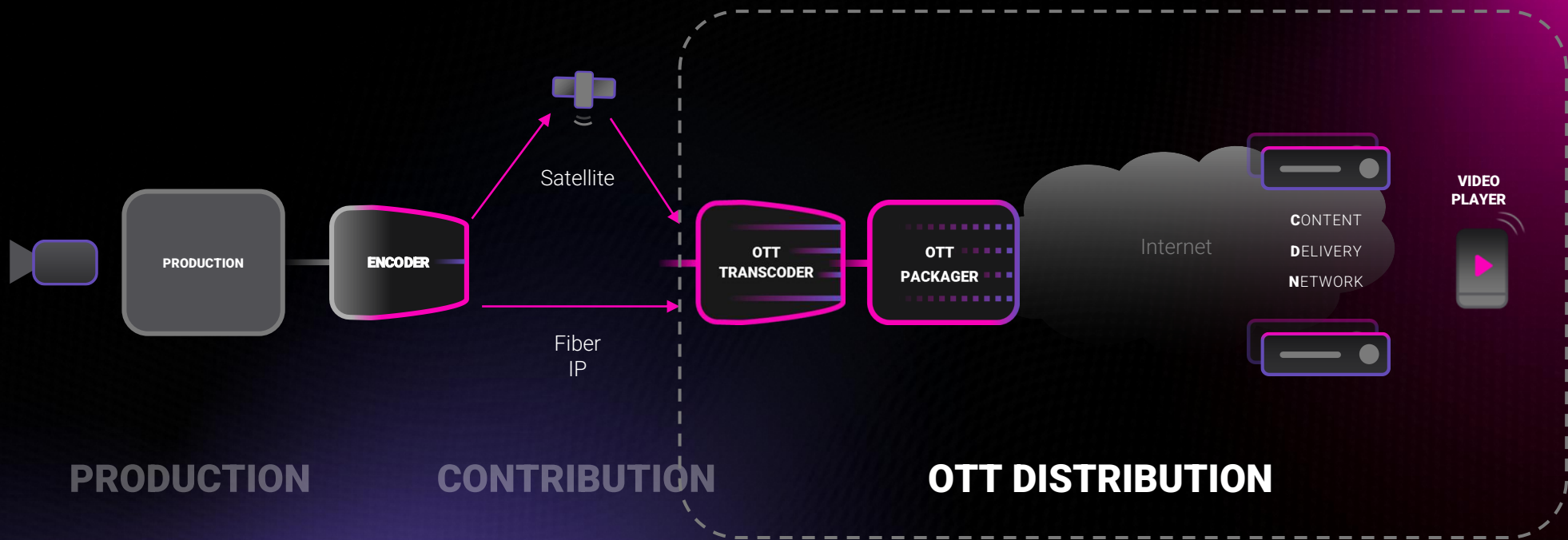
COGICOM

OTT DELIVERY CHAIN

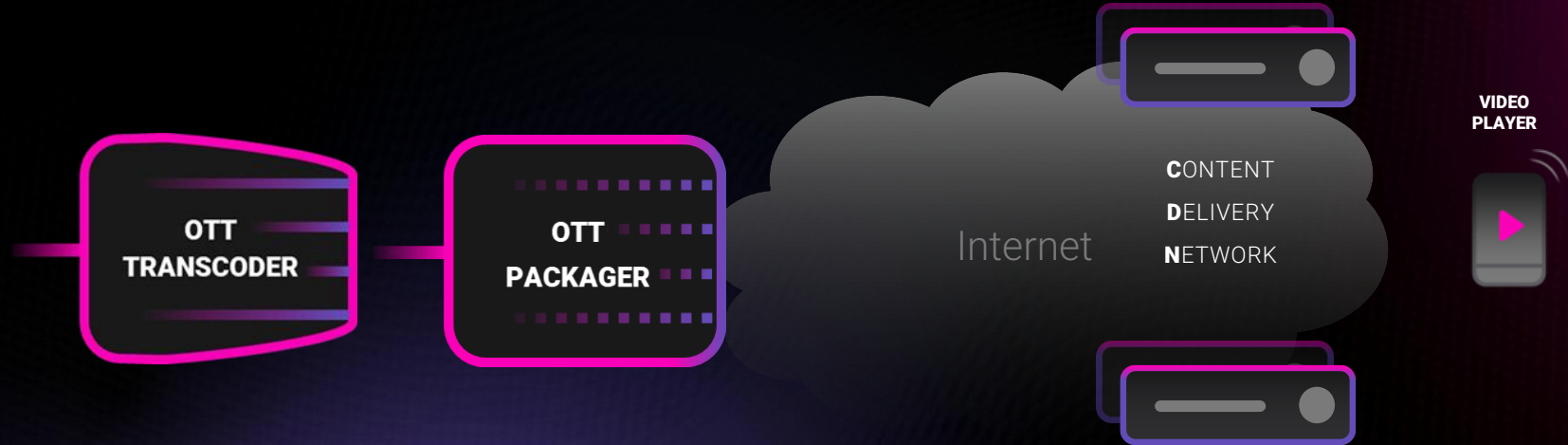
COMPONENT & ROLES

COGICOM

OVER-THE-TOP ORIGATION

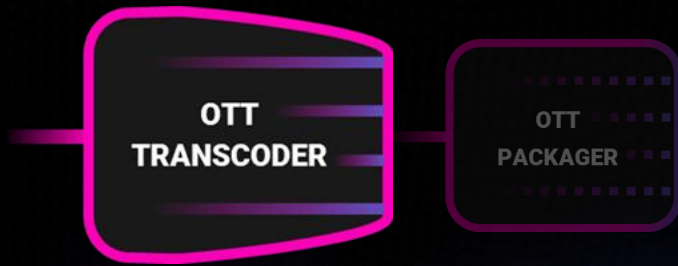


TRANSCODER DEFINES VARIANTS & VIDEO QUALITY

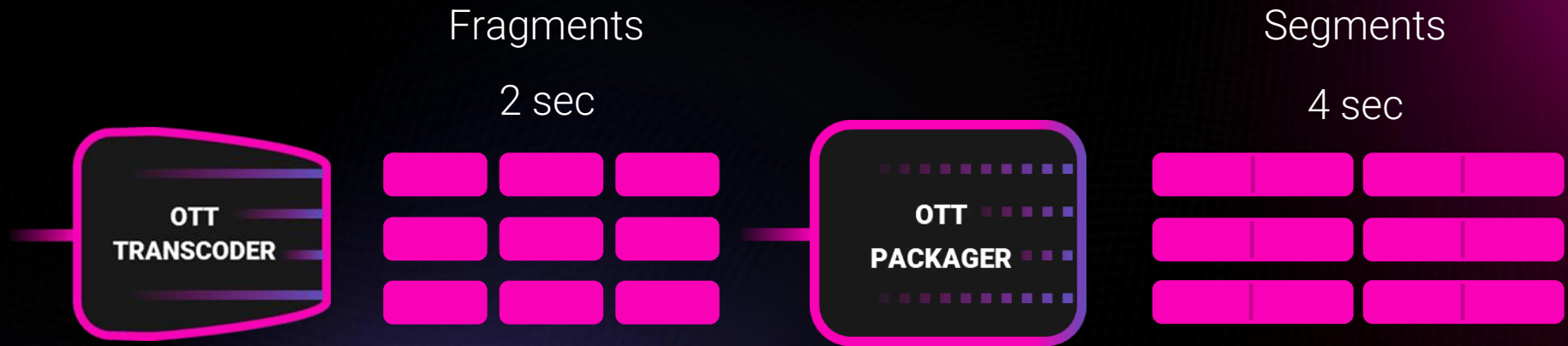


TRANSCODER DEFINES VARIANT & VIDEO QUALITY

- ▶ Variants, Bitrates & Video quality
- ▶ Audio & subtitles



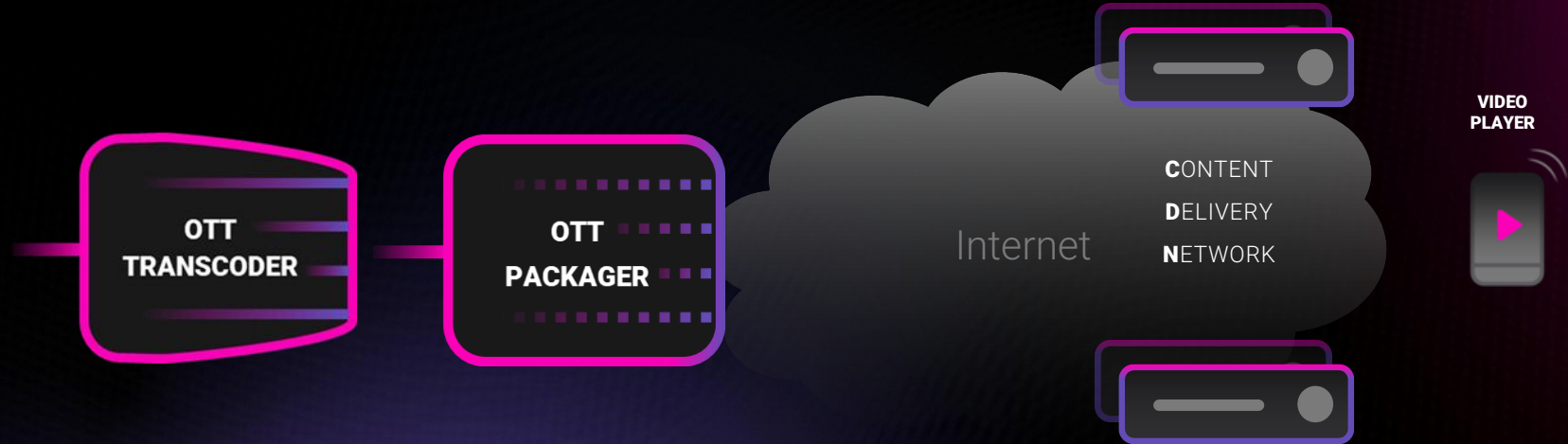
- ▶ Create **OTT fragments**
- ▶ Send fragments to the packager via: *(either)*
 - ▶ UDP over MPEG2TS *(packager on premise)*
 - ▶ CMAF Ingest *(packager in the cloud)*



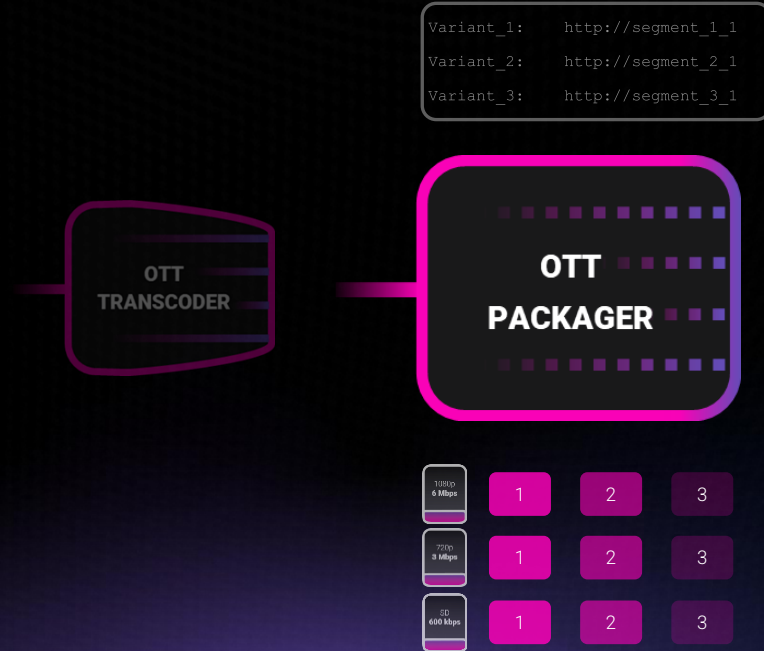




THE OTT PACKAGER/ORIGIN SERVER

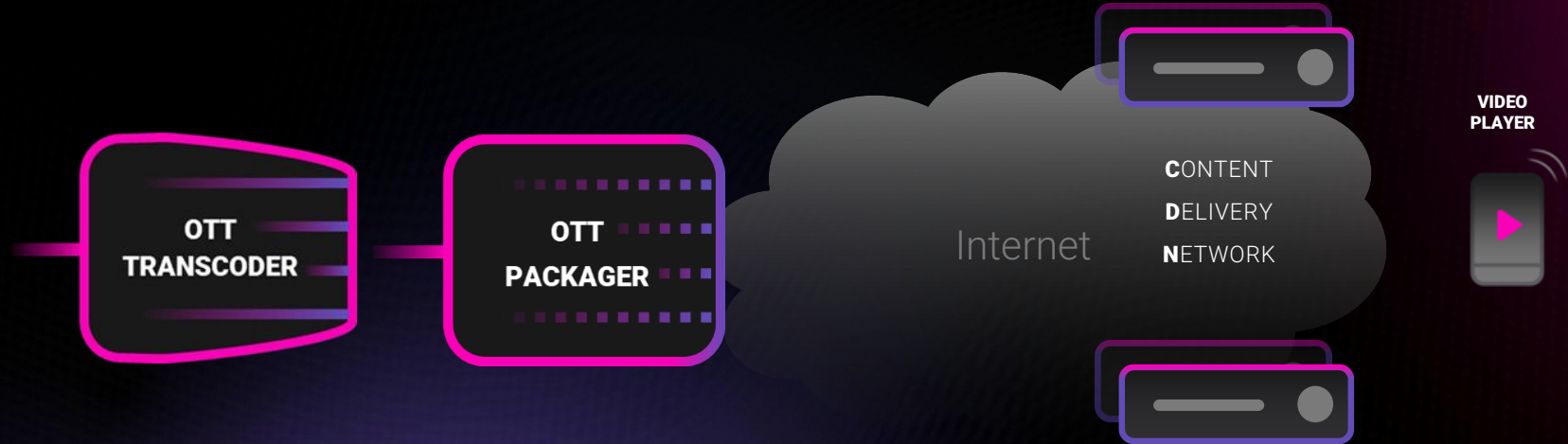


PACKAGER CREATES SEGMENTS & MANIFESTS

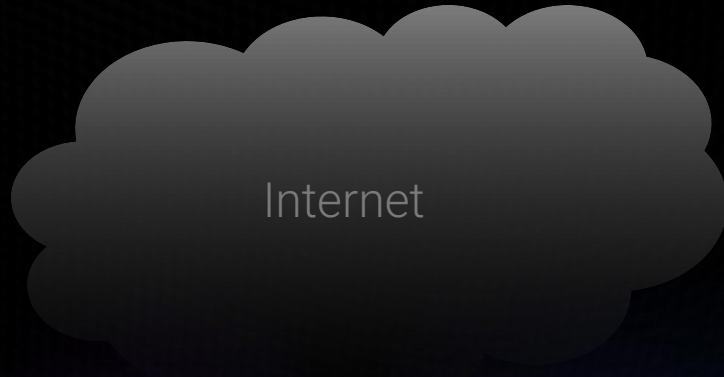


- ▶ Also called the **origin server**
- ▶ Creates manifests files (*HLS or DASH*)
- ▶ Creates OTT segments *from fragments*
- ▶ Segments / manifests are then pulled by players through the CDN

THE INTERNET

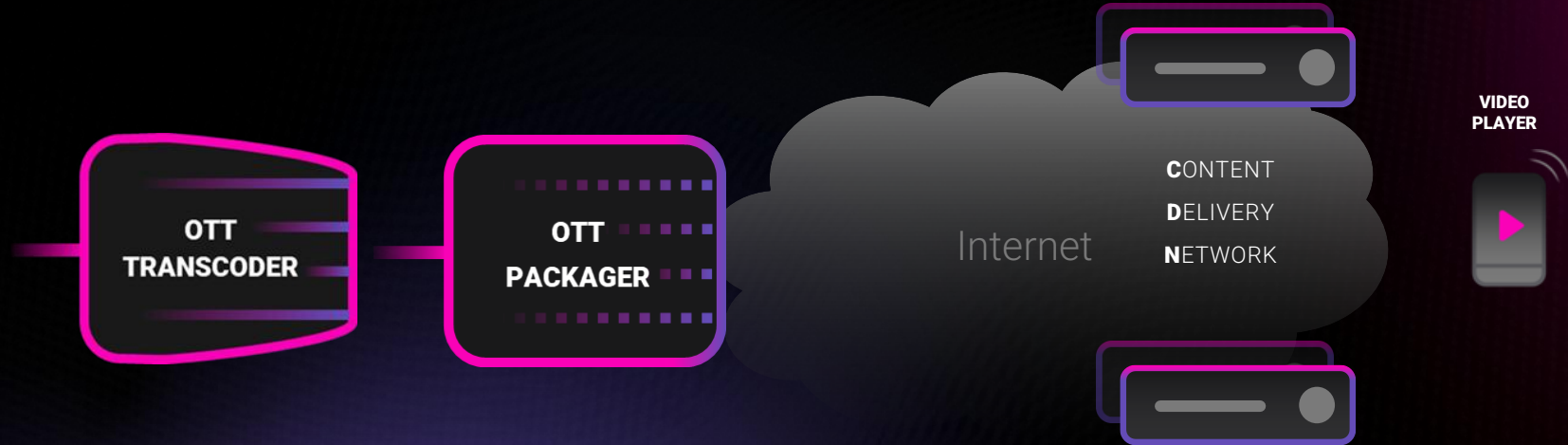


THE OPEN INTERNET: AN UNRELIABLE NETWORK

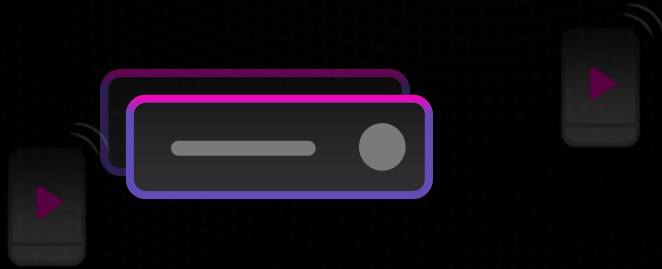


- ▶ The Inter-networks (network of networks)
- ▶ Unreliable network
- ▶ No Quality of Service applied to traffic
 - ▶ Bandwidth / delay varies
- ▶ Need for reliable delivery

THE CONTENT DELIVERY NETWORK

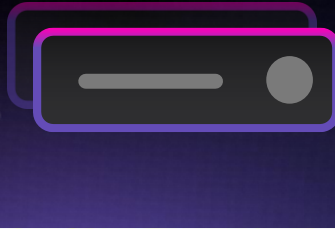


THE CONTENT DELIVERY NETWORK



CONTENT
DELIVERY
NETWORK

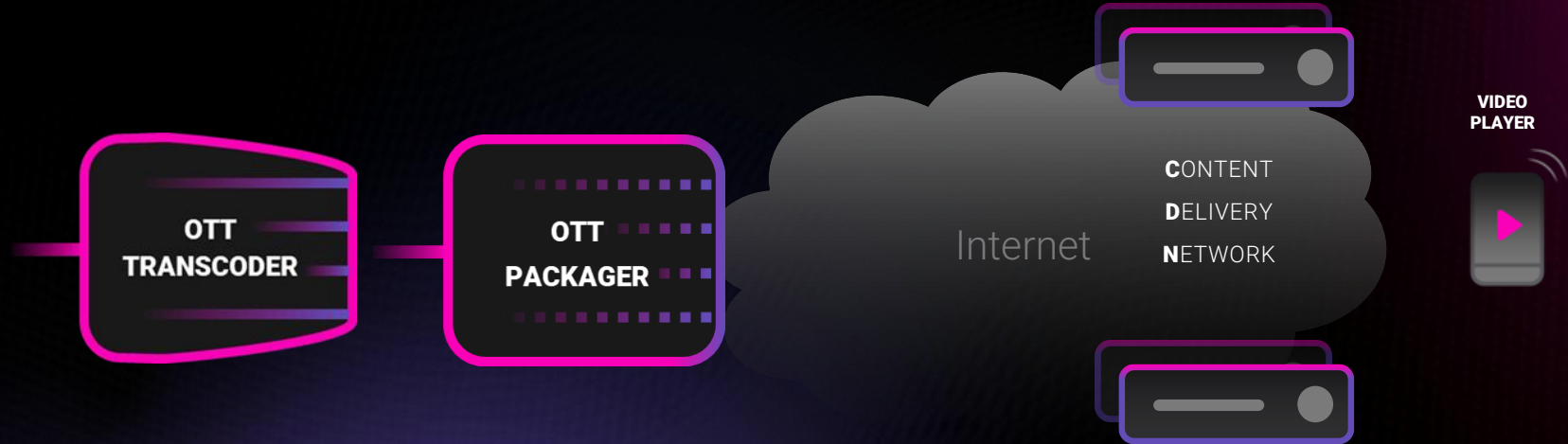
- ▶ Set of **caching servers** located in different geographical zones
- ▶ Also called Point of Presence (PoP)
- ▶ Close to viewers to optimize delivery
- ▶ The entire traffic go through them



COGICOM

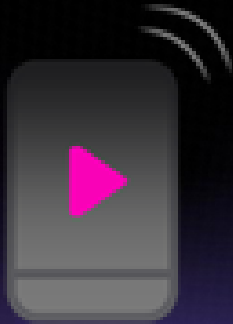
THE OTT DELIVERY CHAIN

THE OTT PLAYER



OTT PLAYER: THE MOST COMPLEX COMPONENT

VIDEO PLAYER



- ▶ Device orchestrating & initiating all messages
- ▶ Decides which variant to use & when to download segments
- ▶ The **most complex component** of the entire workflow



COGICOM

THE OTT DELIVERY CHAIN

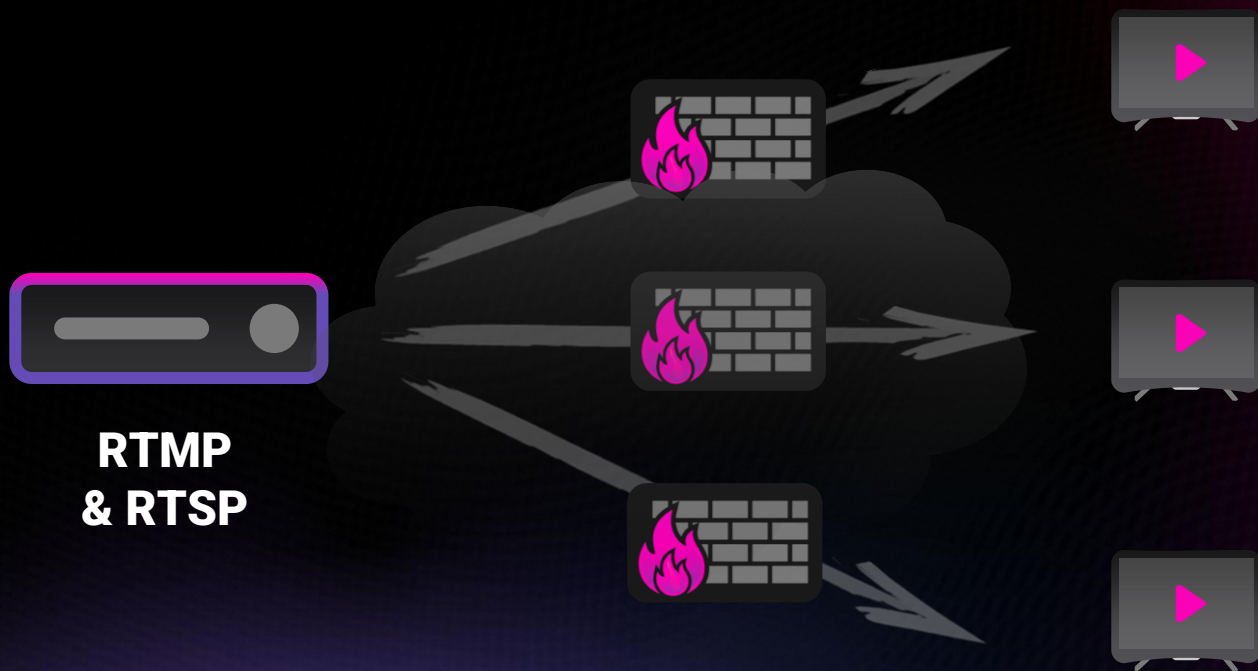
BIRTH OF THE ADAPTIVE STREAMING

COGICOM

Real Time Messaging Protocol & Real Time Streaming Protocol



THE LIMITATIONS OF RTMP & RTSP

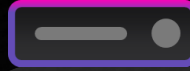


PUSH

- ▶ Stateful
- ▶ Centralized
- ▶ UDP based

ADVENT OF CDN TO OPTIMIZE WEB FILE CACHING

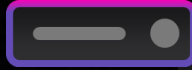
**Origin
server**



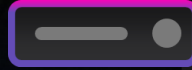
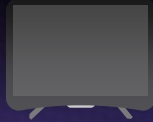
Index.html
image.jpg

Internet

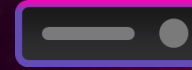
**Edge Cache
servers**



Index.html
image.jpg



Index.html
image.jpg



Index.html
image.jpg



ADVANTAGES OF CDN FOR THE INTERNET VIDEO DELIVERY

- ▶ Treating video like web files & using CDN infrastructure would allow to :
 - ▶ Scale globally
 - ▶ Reduce Costs
 - ▶ Reach viewers globally

Why Adaptive Streaming was not use at scale before the Iphone 3 ?

Before the iPhone 3G and 3GS era (2008–2009), the tech landscape simply didn't have the infrastructure, the hardware, or—honestly—the urgent need to deploy adaptive streaming at a massive scale.

While the concept of adaptive streaming existed in labs and small proprietary rollouts (like Move Networks or early Microsoft Silverlight), it didn't become the global standard until mobile video exploded.

Several major roadblocks kept adaptive streaming from scaling before that point:

1. The Dominance of Flash and "Progressive Download"

In the mid-2000s, desktop computers ruled the internet, and Adobe Flash Player was king. Websites like YouTube delivered video using progressive download over standard HTTP.

How it worked: The browser downloaded one single video file from start to finish. If your internet was fast, it buffered ahead. If it was slow, you paused the video and waited for the gray bar to fill up.

Why it stopped scaling: On a wired desktop computer or a stable home Wi-Fi network, progressive download worked "good enough." Nobody felt a pressing need to completely overhaul the internet's video infrastructure until millions of people started trying to watch video while moving around on highly unstable cellular networks.

2. The Legacy Streaming Protocols Were a Networking Nightmare

Before HTTP-based adaptive streaming (like HLS), true streaming relied on older protocols like RTSP (Real-Time Streaming Protocol) and RTMP (Real-Time Messaging Protocol). These had two massive flaws:

Firewalls hated them: They required opening specific, non-standard networking ports. Corporate firewalls and cellular provider networks frequently blocked these ports, meaning video streams would just randomly fail for users

They didn't scale cheaply: They required specialized, expensive streaming servers that had to maintain a constant, active connection with every single viewer. Standard HTTP web servers (which host normal websites and HLS chunks) are incredibly cheap, highly efficient, and easily cached by Content

Delivery Networks (CDNs).

3. Mobile Hardware and Networks Were Not Ready

Adaptive streaming requires a lot of heavy lifting behind the scenes.



DOMINANT OTT PROTOCOLS

HLS & MPEG DASH

COGICOM

TWO MAJOR OTT PROTOCOLS



HTTP Live Streaming



 **COGICOM**



MPEG DASH



DOMINANT OTT PROTOCOLS

TWO MAJOR OTT PROTOCOLS



HTTP Live Streaming



MPEG DASH

Differ in the
format of the manifest

The way variants & segments are presented to players



DOMINANT OTT PROTOCOLS

TWO MAJOR OTT PROTOCOLS



HTTP Live Streaming



MPEG DASH

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-STREAM-INF:BANDWIDTH=300000
variant_720p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=600000
variant_1080p.m3u8
```

```
#EXTINF:4,
segment1.ts
#EXTINF:4,
segment2.ts
```

```
#EXTINF:4,
segment1.ts
#EXTINF:4,
segment2.ts
```

```
<MPD
  <Period id="1" ...>
    <AdaptationSet >
      <SegmentTemplate >
        <SegmentTimeline>
          <S t="0" d="120" r="7" />
          <S d="40" />
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="720p" bandwidth=" 300000 "/>
      <Representation id="720p" bandwidth=" 600000 "/>
```



DOMINANT OTT PROTOCOLS

TWO MAJOR OTT PROTOCOLS

Vendor



Name

HTTP
Live Streaming

MPEG DASH

*Media
Container*

MPEG2-TS

ISO BMFF
Profile: DASH

Player



DOMINANT OTT PROTOCOLS

Quiz 3

COGICOM