

**Learn to  
Design Cloud  
Architecture**





# Let's build a Video Streaming solution

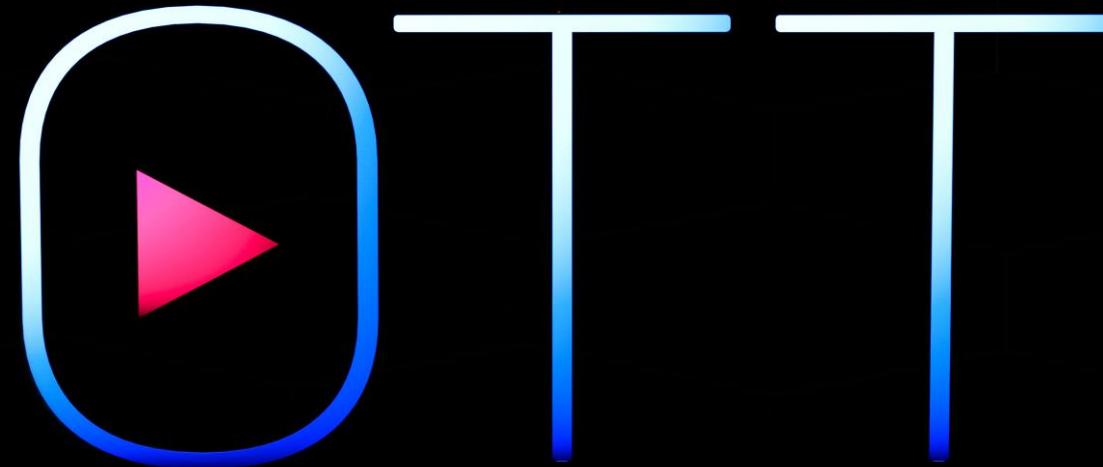
# What is Video Streaming?

- Streaming is the continuous transmission of audio or video files from a server to a client allowing users to watch videos without needing to download the entire file first.
- Streaming can be used for live-videos (events) or pre-recorded videos.



# Over the top (OTT) Platform

- A service that delivers video, audio, or other media content directly to users via the internet, bypassing traditional cable, satellite, or broadcast television platforms.



- Key Features of OTT Platforms:
  - On-Demand
  - Subscription or Ad-Based
  - Multi-Device Access
  - Original Content

# Some popular OTT platforms

Entertainment

**NETFLIX**

amazon  
**prime video**



Live TV/News

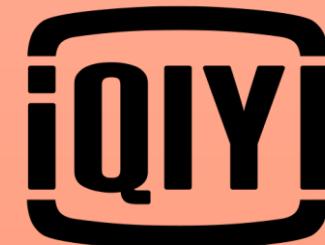
 **YouTubeTV**

**sling**

**BBC**  
**iPLAYER**

Regional/Local

**hotstar** 

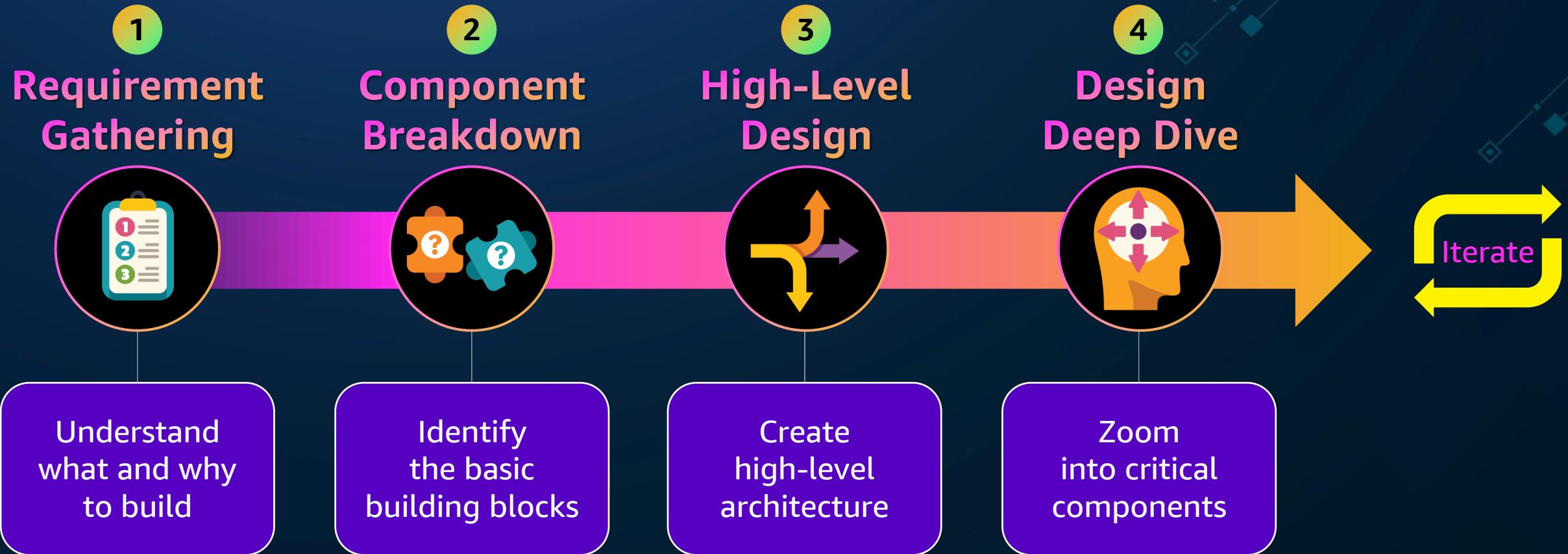


# Netflix

- **Founded:** 1997
- **Original Model:** DVD-by-mail rental service in the U.S.
- **Streaming Launch:** 2007
- **Total Subscribers:** ~270 million paid subscribers globally
- **Top features:**
  - Cross-Device Support
  - Adaptive Streaming (ABR)
  - Multiple Profiles
  - Offline Downloads
  - Localized Content & Subtitles

NETFLIX

# Designing a System – A simple framework



# Requirement Gathering



# OTT PLATFORM

Who's using our platform ?



Content Viewer



Content Creator

# Functional Requirements

- Video Streaming & Playback
- Search & Discovery
- Recommendation Engine
- User and Subscription Management
- Geo Fencing and Regional Content



The goal is to serve the best quality video with the minimum possible bandwidth.

# Non-Functional Requirement

## High Availability

Users expect 24/7 access across regions with minimal downtime.

## No Buffering

Smooth playback and responsiveness are critical to user experience.

## Reliability

Content should never be corrupted or lost; uptime and video integrity matter.

## Security

Protects user data, content, and enforces subscription and geo restrictions.

## Scalability

Must support millions of concurrent users and growing content libraries.

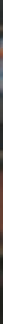
# Out of scope

- Multiple Profiles
- Payments
- Offline Download
- Parental Control
- Viewing History
- Digital Rights Management
- Content Delivery Network
- Analytics
- Recommendation Engine
- Etc..



# NEW SERIES

# STRANGER STREAMS





## Episode 1

### Welcome to Streamverse

Overview of the streaming ecosystem



## Episode 2

### From Director to Disk

Content upload & ingestion pipeline



## Episode 3

### Where Videos Sleep

Video storage service design



## Episode 4

### Making Videos Watchable

Video processing, encoding, chunking



## Episode 5

### The Stream Engine

Video streaming service & ABR logic



## Episode 6

### The CDN Highway

Content Delivery Networks (CDNs)



## Episode 7

### Frontends on Fire

Frontend interface architecture



## Episode 8

### The Silent Helpers

Operational backbone of the system



## Episode 9

### Techflix behind Netflix

Technology beyond the Play button



## Episode 10

### Cloud Behind the Curtain

The AWS backbone of Netflix



## Episode 11

### Build Your Own Mini Netflix

Create your streaming platform



## Episode 12

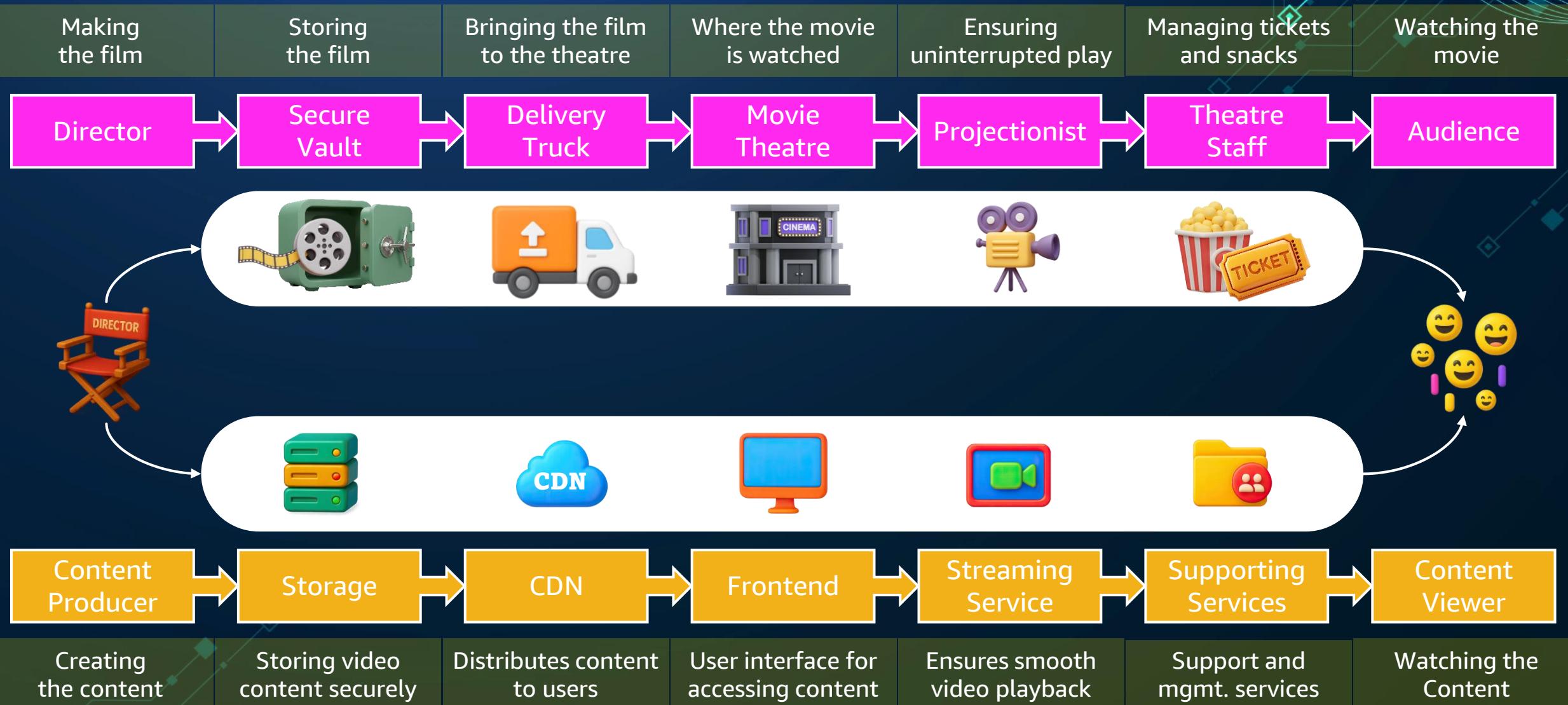
### Streaming Must Go On

Building for scale and resilience

A white and black humanoid robot is seated in a row of red theater seats. It is holding a dark bucket filled with popcorn in its right hand. The robot has a sleek, metallic design with glowing red lights on its joints and chest. The background shows rows of red theater seats stretching into the distance.

# Episode 1: Welcome to Streamverse

# Traditional Approach



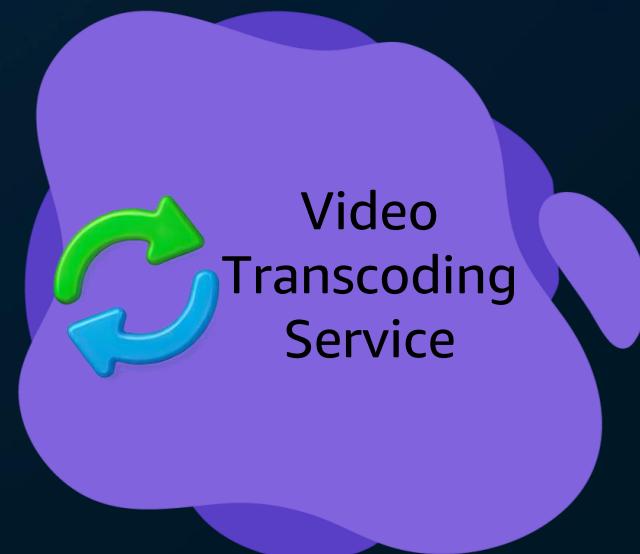
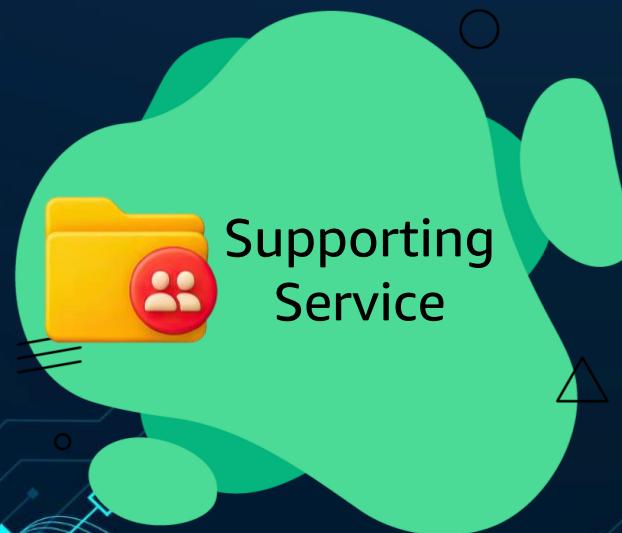
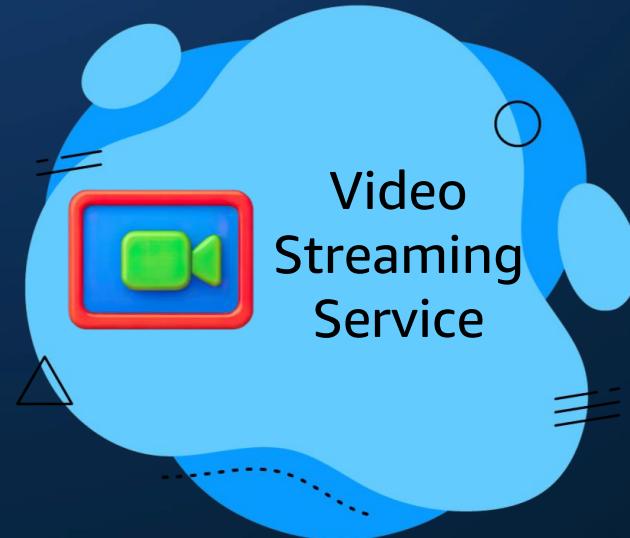
# Streaming Approach



# Component Breakdown



# Core components

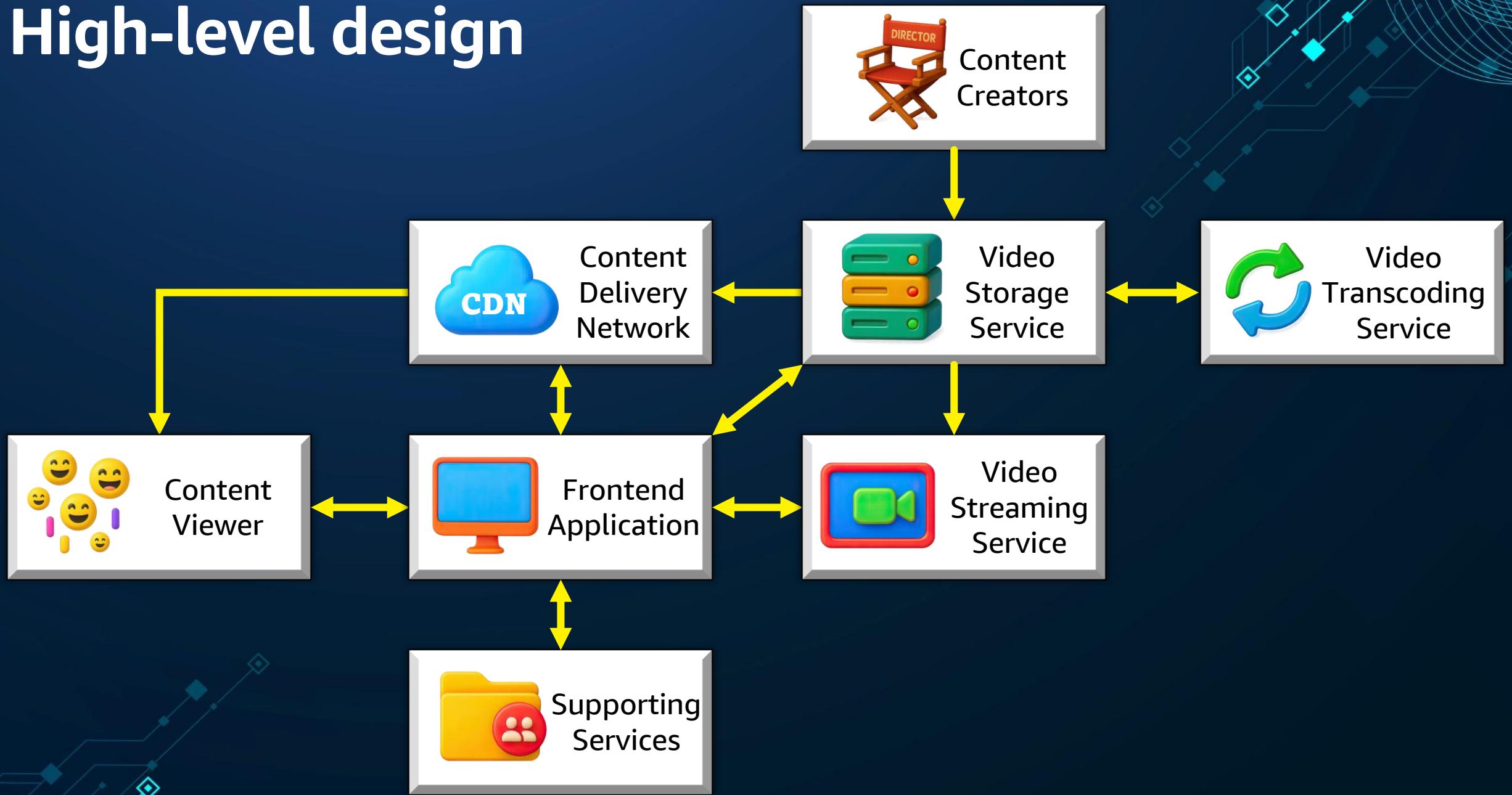




**High-Level  
Design**



# High-level design





# Design Deep Dive



# Episode 2: From Director to Disk



# Content Creators



Production Houses

Individuals

2 Hours raw 8K footage ~ 4 TB

# Mezzanine Floor



# Mezzanine Floor

- 
- The Mezzanine format serves as a "middle ground" format (hence the name, like a mezzanine floor in a building) that:
    - Maintains high quality (but with reasonable compression)
    - Is easier to work with than raw footage
    - Can be used to create multiple lower-quality versions

# Mezzanine File Formats



Content  
Creators

Film Shoot – 100 TB of raw footage



Converted into mezzanine – 20 TB of ProRes format



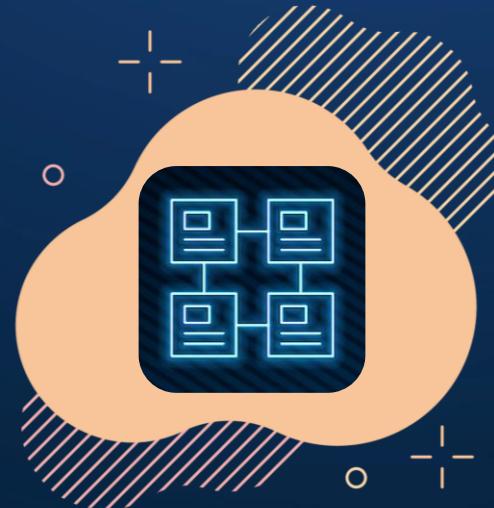
Exported for streaming – 2 GB H.264 format

## Common Mezzanine formats:

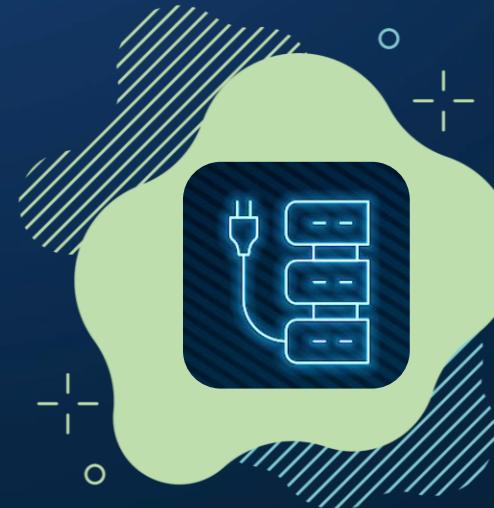
- ProRes 422
- DNxHD/DNxHR
- XDCAM HD
- J2K

# Uploading Videos

Support  
for  
large files



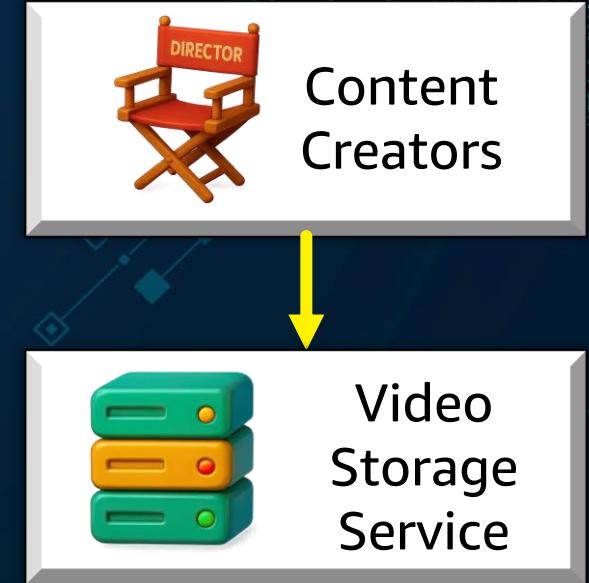
Network  
failure  
tolerance



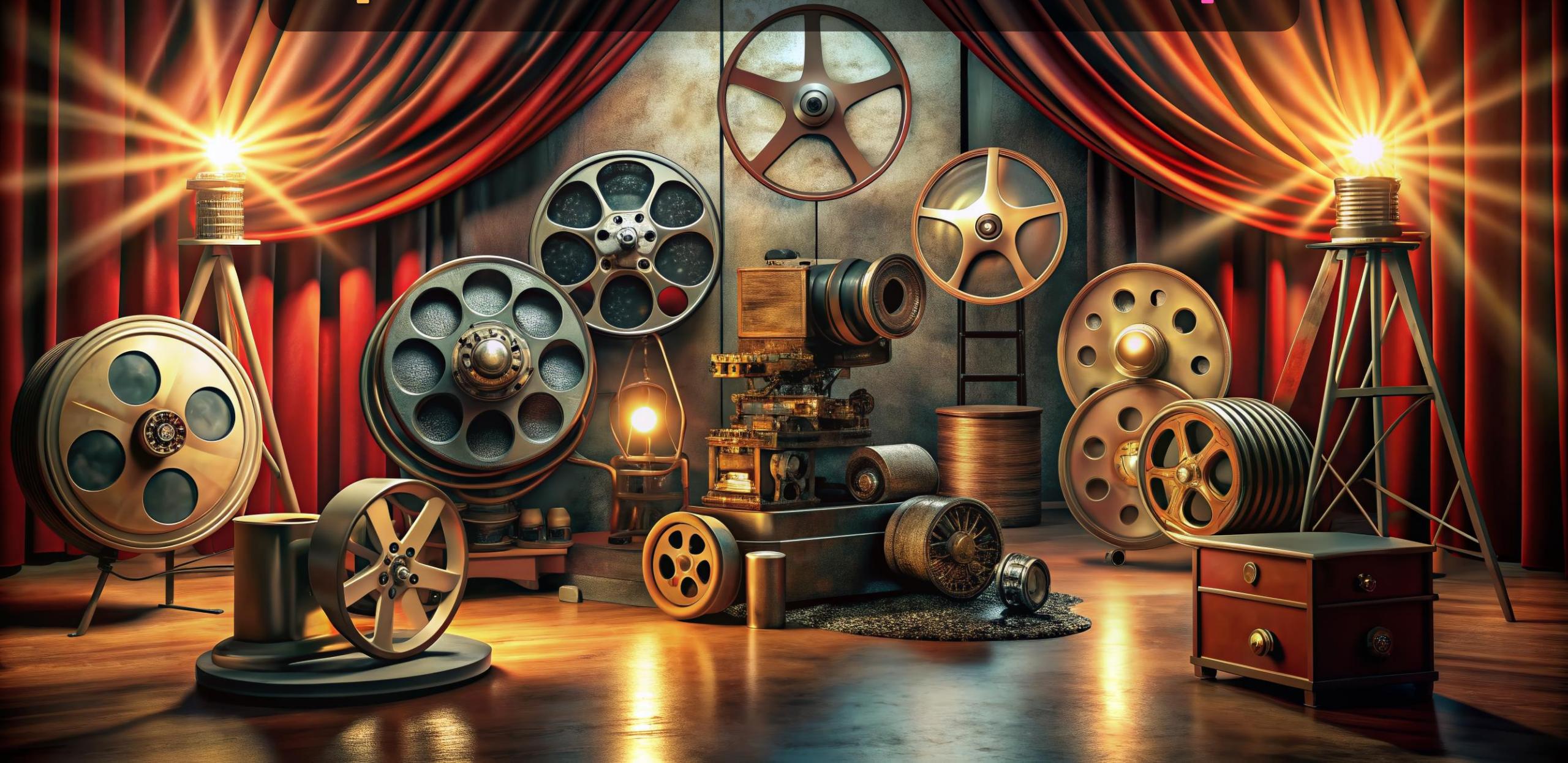
Physical  
transport of  
the data



Leveraging  
edge  
network



# Episode 3: Where Videos Sleep



# Video Storage Service

- Persistently storing all video assets, including:



Original  
Uploads

Transcoded  
Videos

Video  
Metadata

Archived  
Content

# Storing Video Metadata

- A database-backed service that stores all the descriptive, relational, and operational data about each video

Layer	Metadata Type	Example
Content-Level	Title, synopsis, genre, language, age rating	"Inception", Sci-Fi, PG-13
Video File Info	Duration, resolution, codec, bitrate, file size	1080p, 2.1GB, H.264
Storage Mapping	Where chunks are stored (Storage paths, CDN URLs)	s3://video-bucket/1234/720p/segment3.ts
Localization	Subtitles, audio tracks, translations	Hindi subs, Japanese audio
Ownership/Rights	Creator ID, license type, region restrictions	Creator #332, Geo-restricted to US
Processing Status	Ingested, Transcoded, CDN synced, Errors	Transcoding: Success
Analytics Metadata	View counts, watch-through %, popularity rank	Views: 500K, Retention: 67%

# Episode 4: Making Videos Watchable



# Streaming for Every Screen and Speed



Devices



Format



Dimension



Bandwidth

Smartphones	Video: H.264/AVC	240p (426×240)	< 1 Mbps – Low bitrate
Smart TVs	Video: H.265/HEVC	480p (854×480)	1–3 Mbps - Standard
Streaming Devices	Video: AV1	720p (1280×720)	3–5 Mbps – HD
Game Consoles	Audio: AAC	1080p (1920×1080)	5–8 Mbps – Full HD
Laptop/Desktop	Audio: Dolby Atmos	2160p (3840×2160)	15+ Mbps – 4K UHD

# Transcoding Videos



- Transform uploaded videos into multiple optimized formats



## Decode the Source Video

Uploaded video is decoded from its original format for further processing



## Generate Multiple Variants

Re-encoded into different bitrates, resolutions, and formats



## Segment Each Variant

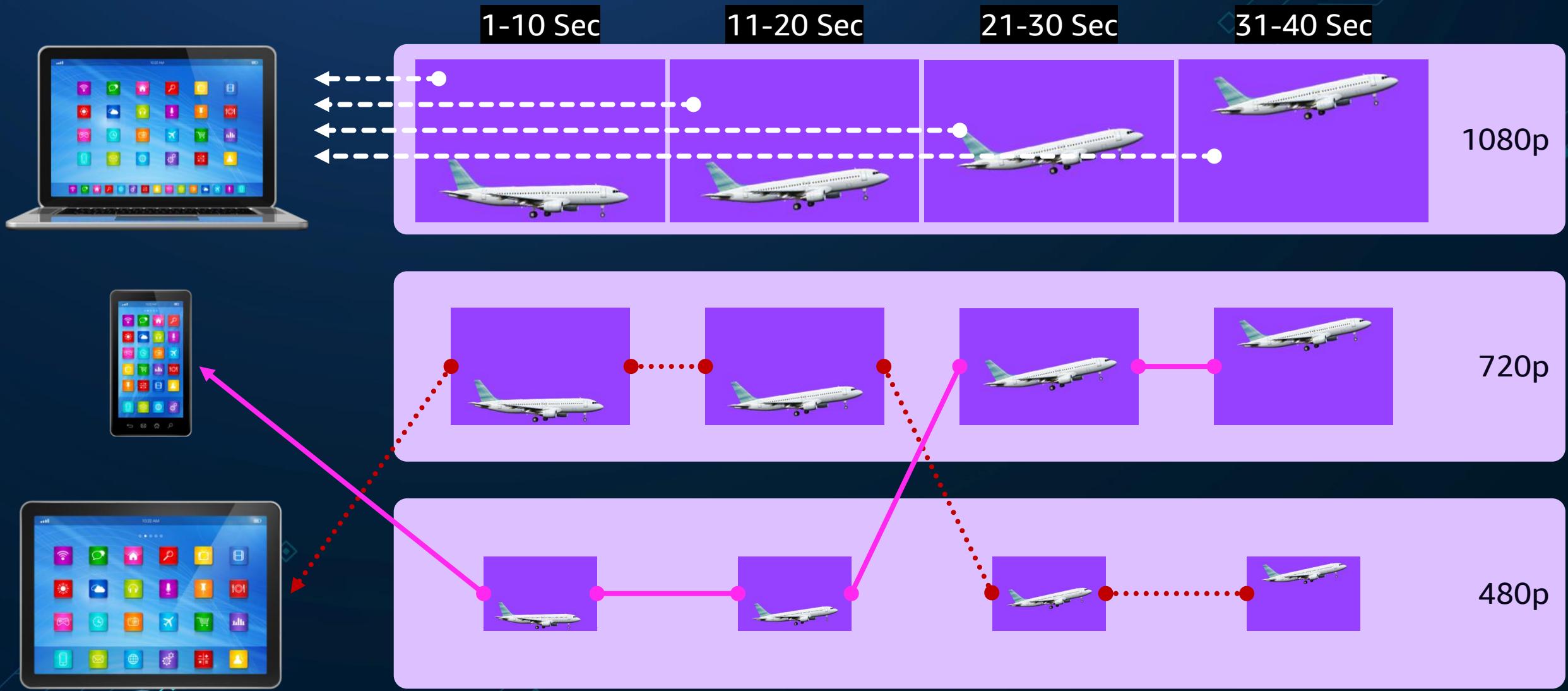
Each variant is broken into small chunks (usually 2–6 seconds)



## Create Manifests for Playback

Manifest lists video variants and segments for adaptive playback

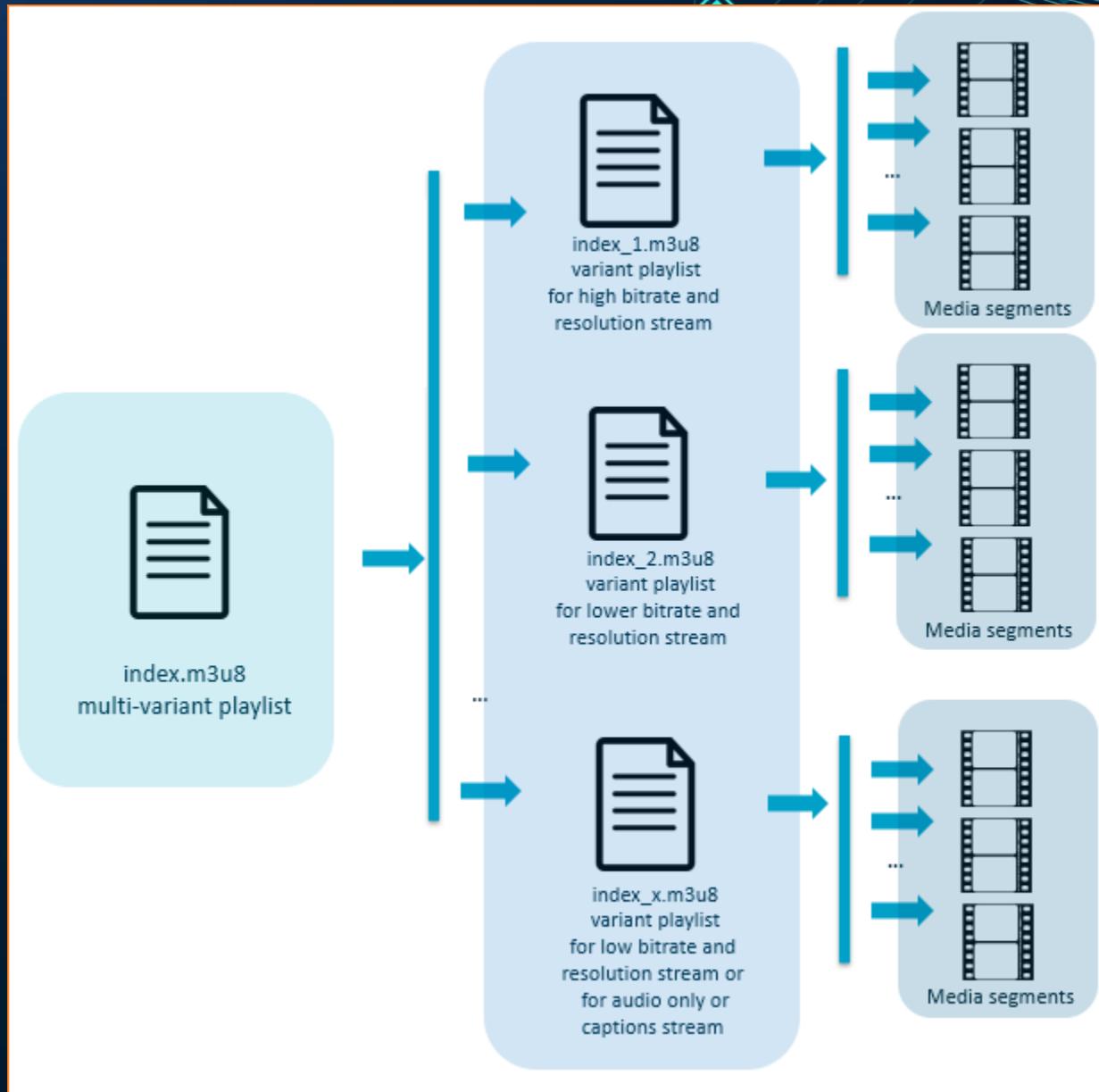
# Adaptive Bit Rate Streaming (ABR)



# Sample Manifest File

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF: BANDWIDTH=500000, RESOLUTION=640x360, CODECS="avc1.42e00a,mp4a.40.2"
http://example.com/low/index.m3u8
#EXT-X-STREAM-INF: BANDWIDTH=1500000, RESOLUTION=854x480, CODECS="avc1.42e00a,mp4a.40.2"
http://example.com/mid/index.m3u8
#EXT-X-STREAM-INF: BANDWIDTH=2500000, RESOLUTION=1280x720, CODECS="avc1.42e00a,mp4a.40.2"
http://example.com/high/index.m3u8
```

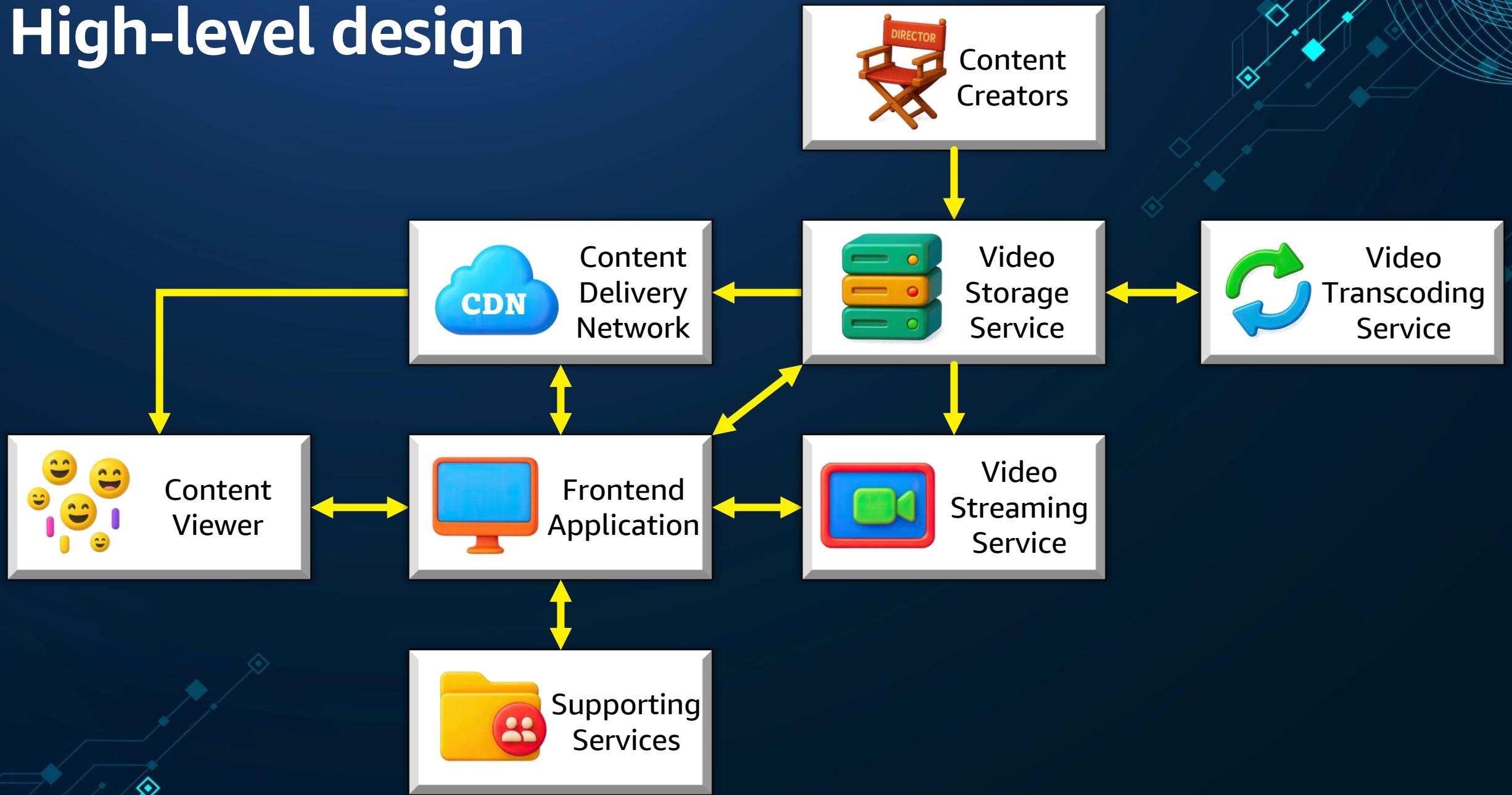
```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:21
#EXTINF:10.00000,
test_1_00025.ts
#EXTINF:10.00000,
test_1_00026.ts
...
```



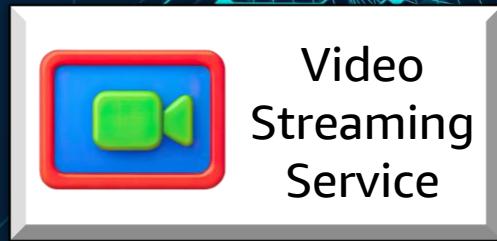
# Episode 5: The Stream Engine



# High-level design



# Video Streaming Service



- **Core Responsibilities:**

- Accepts playback requests from client apps (TV, mobile, web).
- Fetches or locates manifest files generated earlier by the Video Transcoding Service.
- Applies DRM/license integration, token signing, and geo/access restrictions.
- Returns secure playback URLs to the client.
- Optionally logs playback analytics, errors, and session starts.



# Sub-components

## Playback Request Handler

Entry point for any "Play" action from the client  
(TV, web, mobile).

## Access Validator

Validates entitlement of the user  
(e.g. Regional, Parental Control)

## Manifest Service

Locates the correct manifest file  
(with Ads, Watermarked, Language)

## CDN Token Manager

Issues signed URLs pointing to  
segments stored in a CDN

## Device Compatibility Service

Ensures that playback settings are  
compatible with the client device

## Playback Rules Engine

Applies business rules on top of playback  
(e.g. Max streams per account)

# Episode 6: The CDN Highway



# Delivering a new iPhone from Cupertino



# Using Global Supply Chain



Traditional supply chains move physical goods in the same way CDNs move digital content.



### Traditional Supply Chain



### Content Delivery Network

#### Product

Physical goods (e.g., shoes, electronics)

Digital content (e.g., images, videos)

#### Transportation

Shipping trucks, planes, etc.

Internet protocols and networks

#### Source

Central factory or manufacturer

Origin server (where content resides)

#### Regional Distribution

Warehouses or regional distributors

Regional cache servers (edge locations)

#### Local Delivery

Dealers or retail shops

Local cache (close to end-user devices)

#### Delivery Time

Depends on distance and logistics

Optimized for low latency and high speed

#### Cost Efficiency

Higher shipping/storage costs

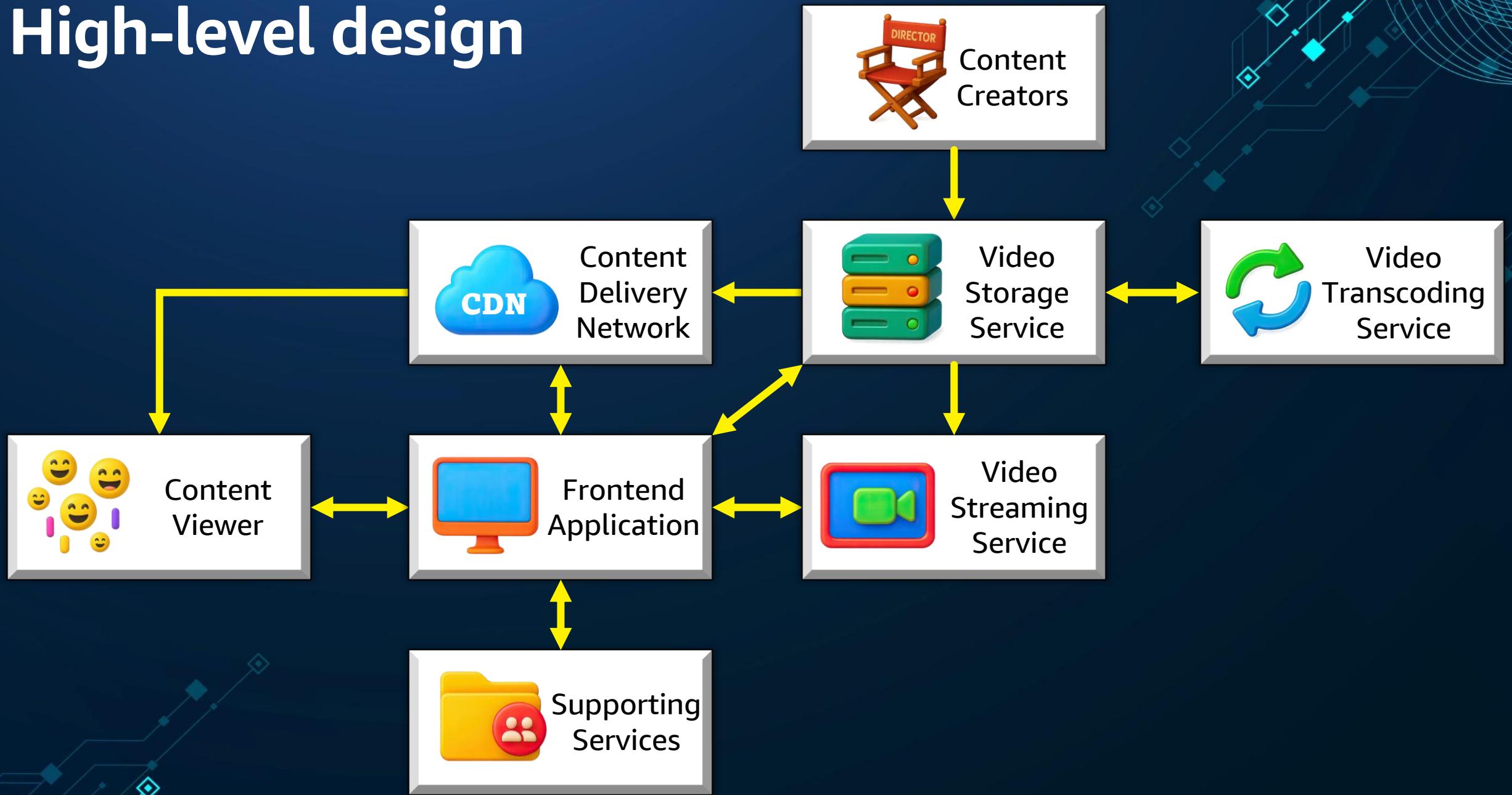
Reduced bandwidth and server load

#### Scalability

Limited by physical infrastructure

Highly scalable with global coverage

# High-level design



# Popular CDNs

fastly

Akamai

CLOUDFLARE®

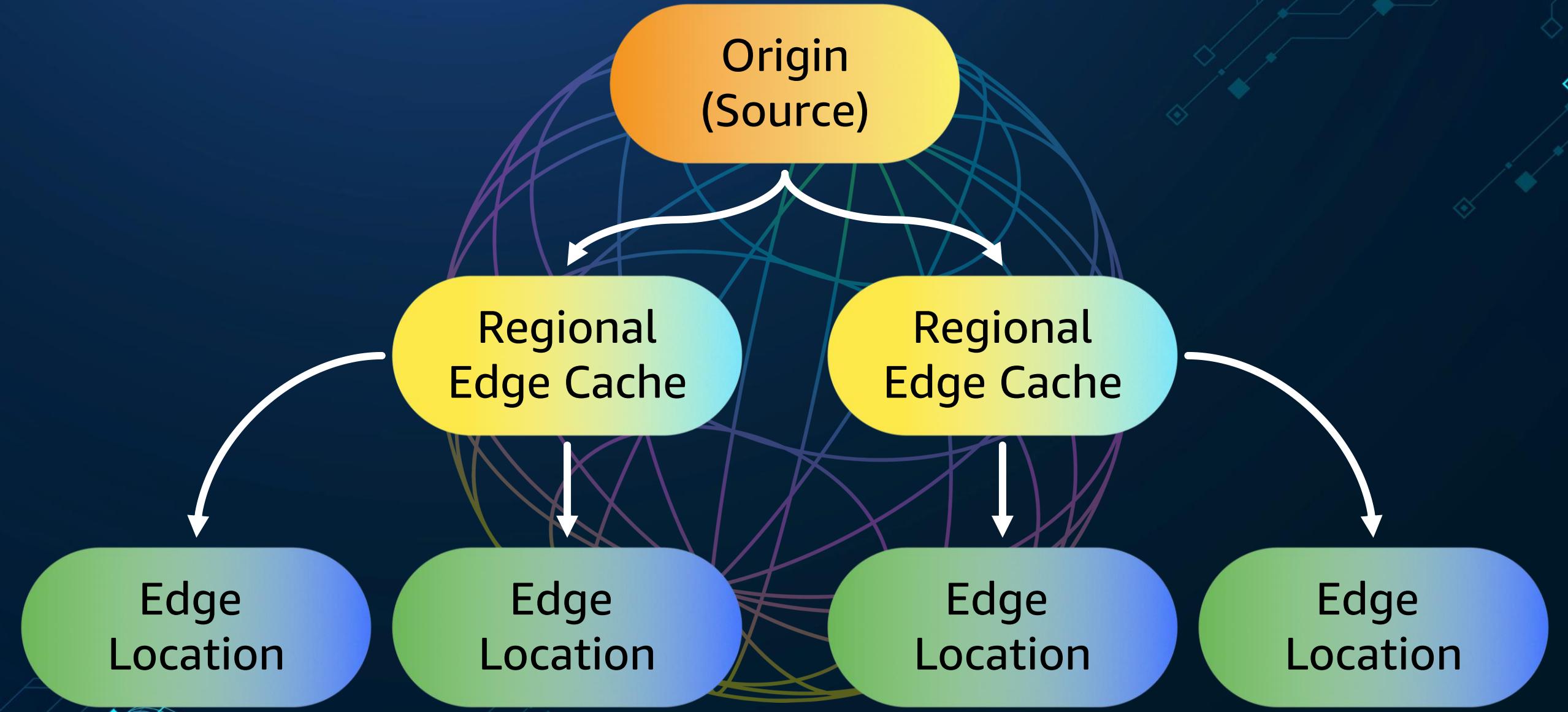


Amazon  
CloudFront

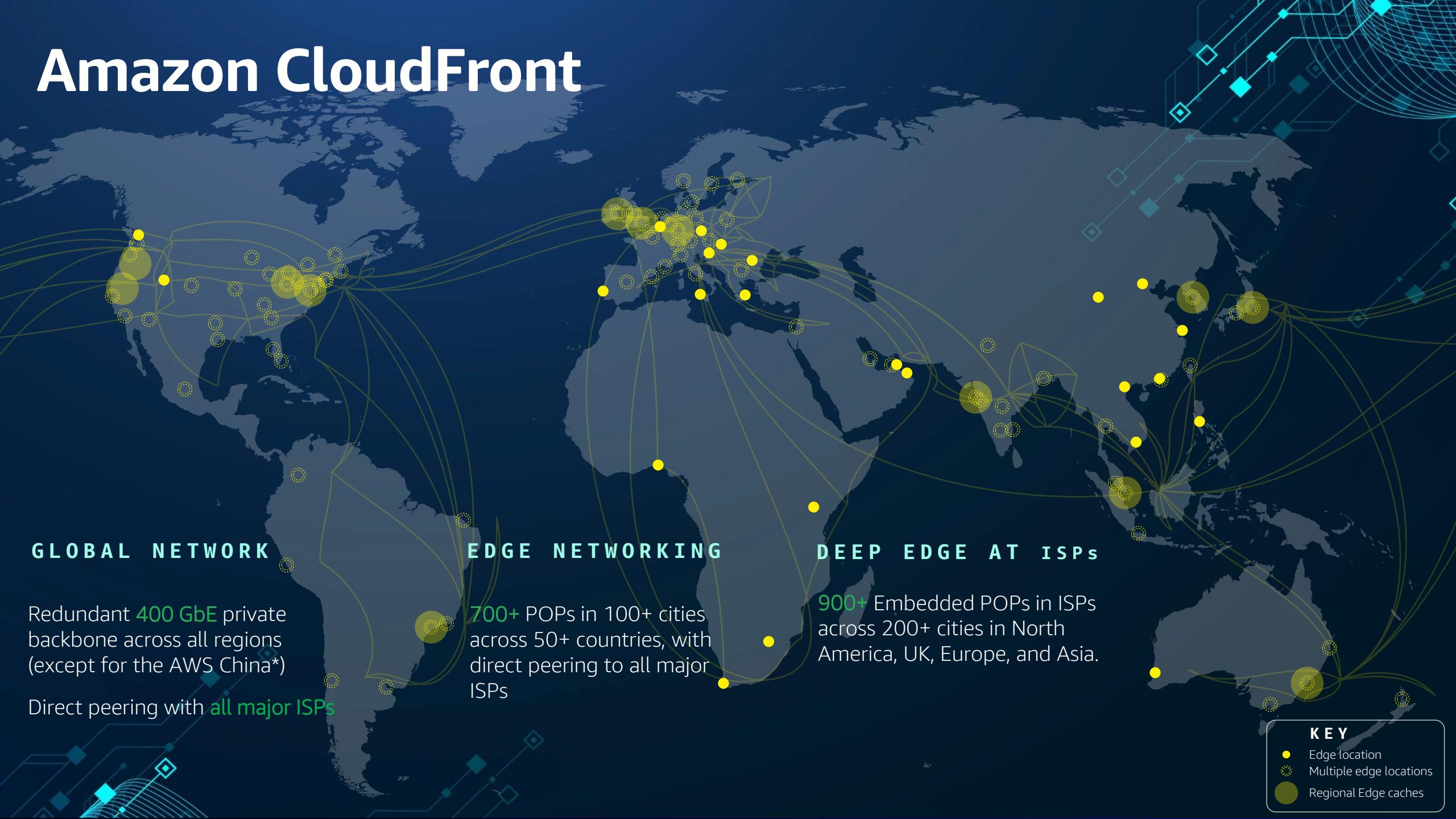




# Amazon CloudFront



# Amazon CloudFront



# Netflix Open Connect



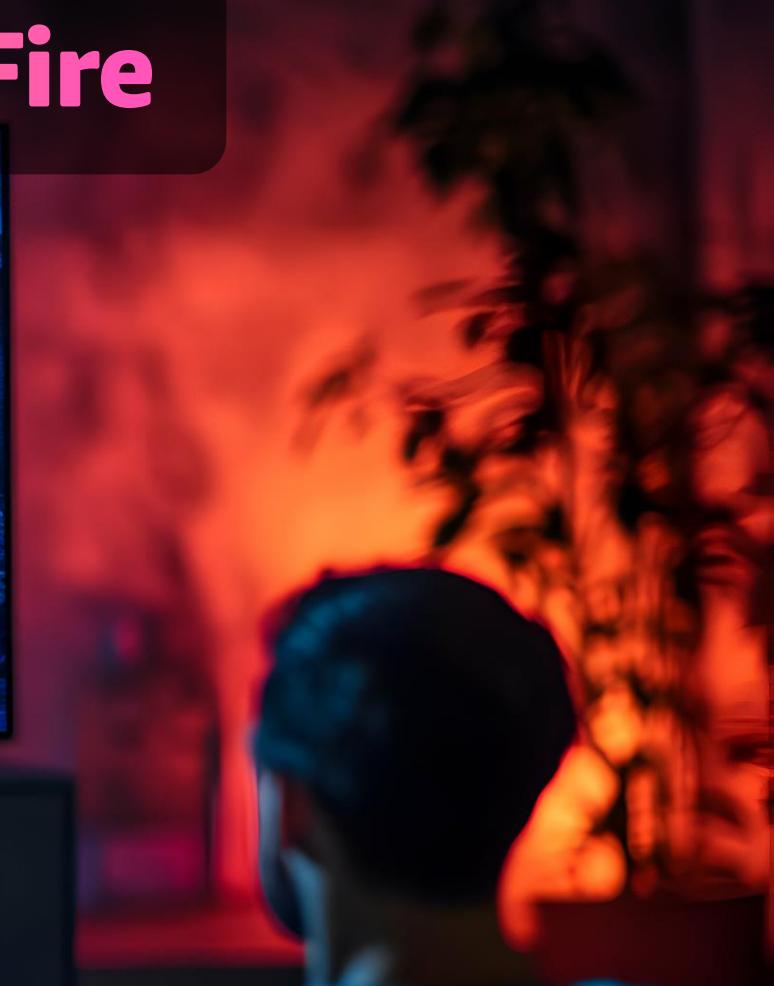
**Open Connect**



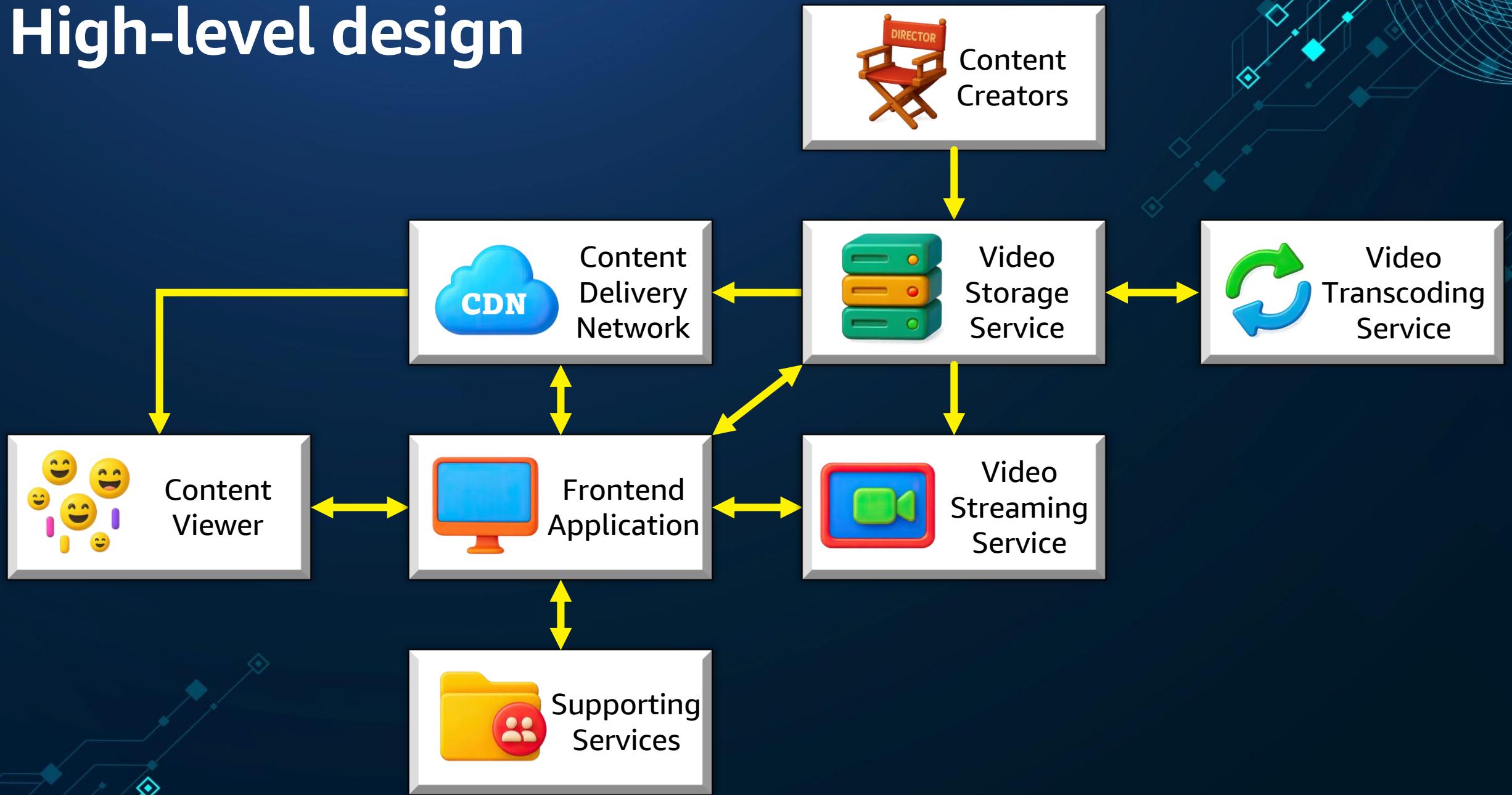
- Custom-Built CDN
- Open Connect Appliances (OCAs)
- Optimized for video-specific workloads



# Episode 7: Frontend on Fire



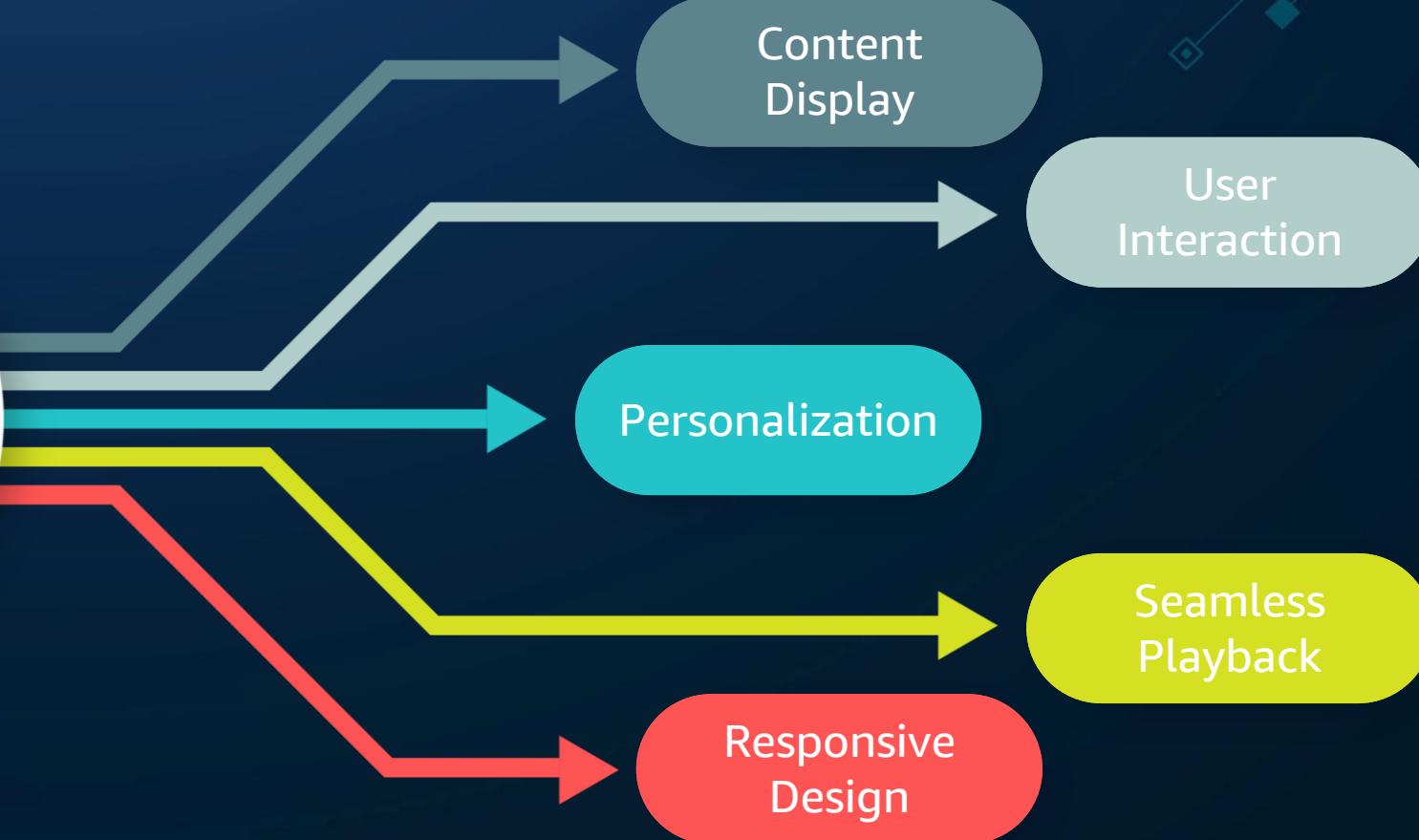
# High-level design



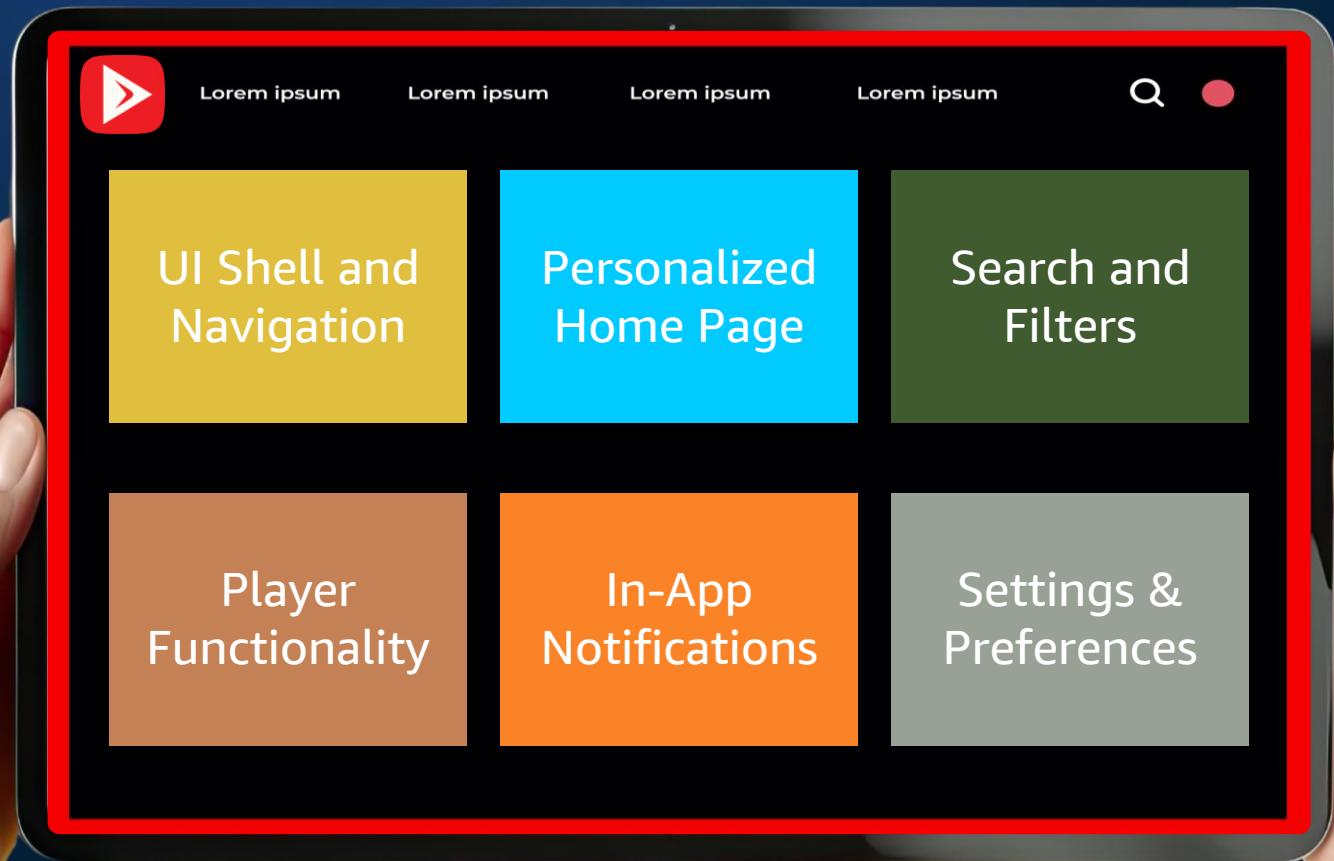
# Frontend Application



User-facing interface of the streaming platform.



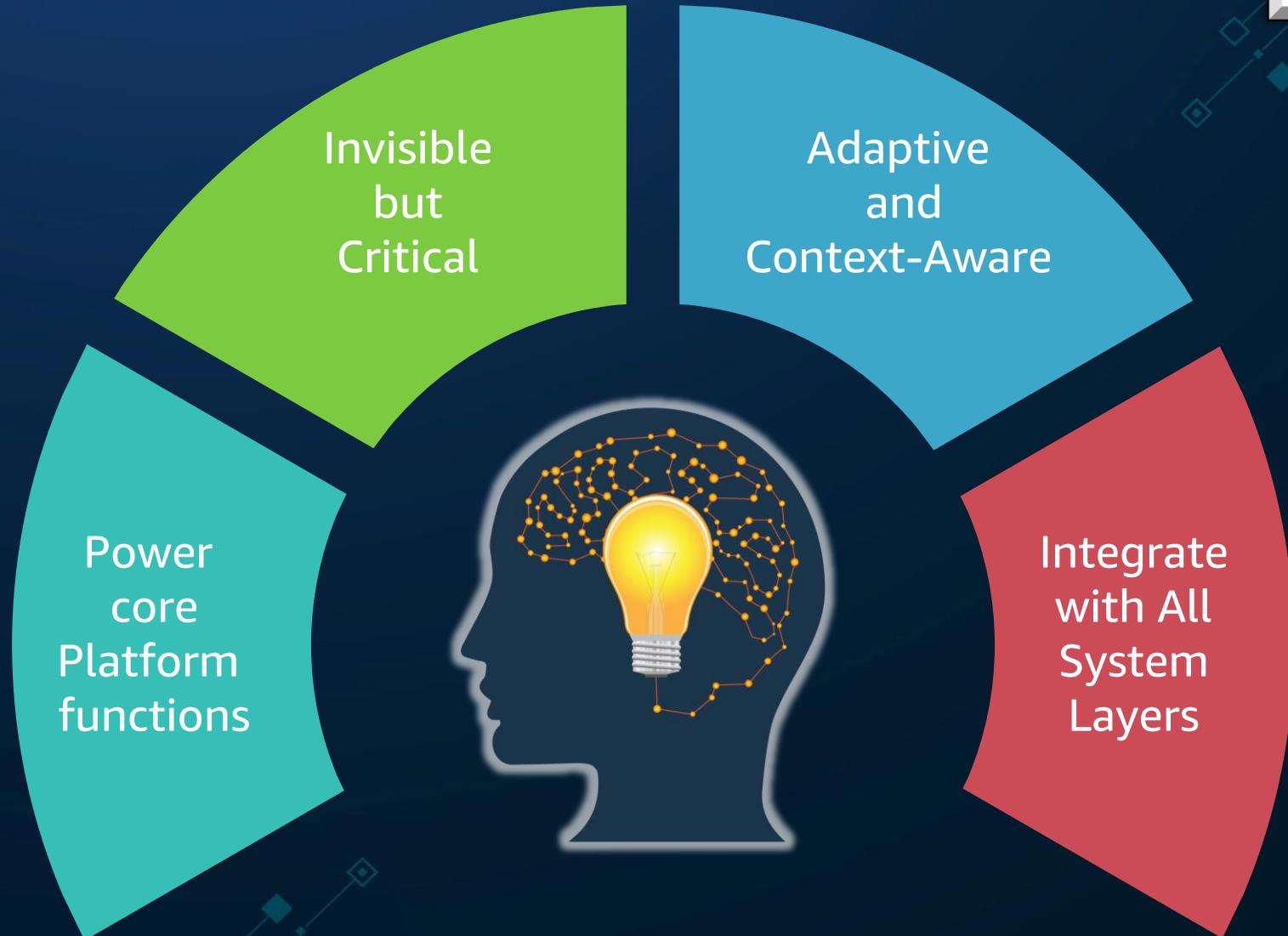
# Sub Components



# Episode 8: The Silent Helpers



# Supporting Services



# Sub Components



## Meet the Matchmaker

Recommendation Engine

A/B Testing Service

Watch History & Preference Engine

Search Index Service

Auto-complete & Suggestions

Genre/Category Tagging Service

## The Content Compass

## Subscriptions & Stores

Payment Gateway Integration

Plan & Subscription Management

Trial & Promotion Management

Authentication & Profiles Service

Parental Controls & Restrictions

Preferences & Language Settings

## Profiles & People

## Guardians of the Stream

DRM License Server

Token Authentication Service

Device Whitelisting

Playback Telemetry

Performance Metrics Collection

Content Performance Dashboard

## The Analytics Observatory

## Global Reach, Local Flavor

Geo restrictions

Multi-language support

Regional Ads

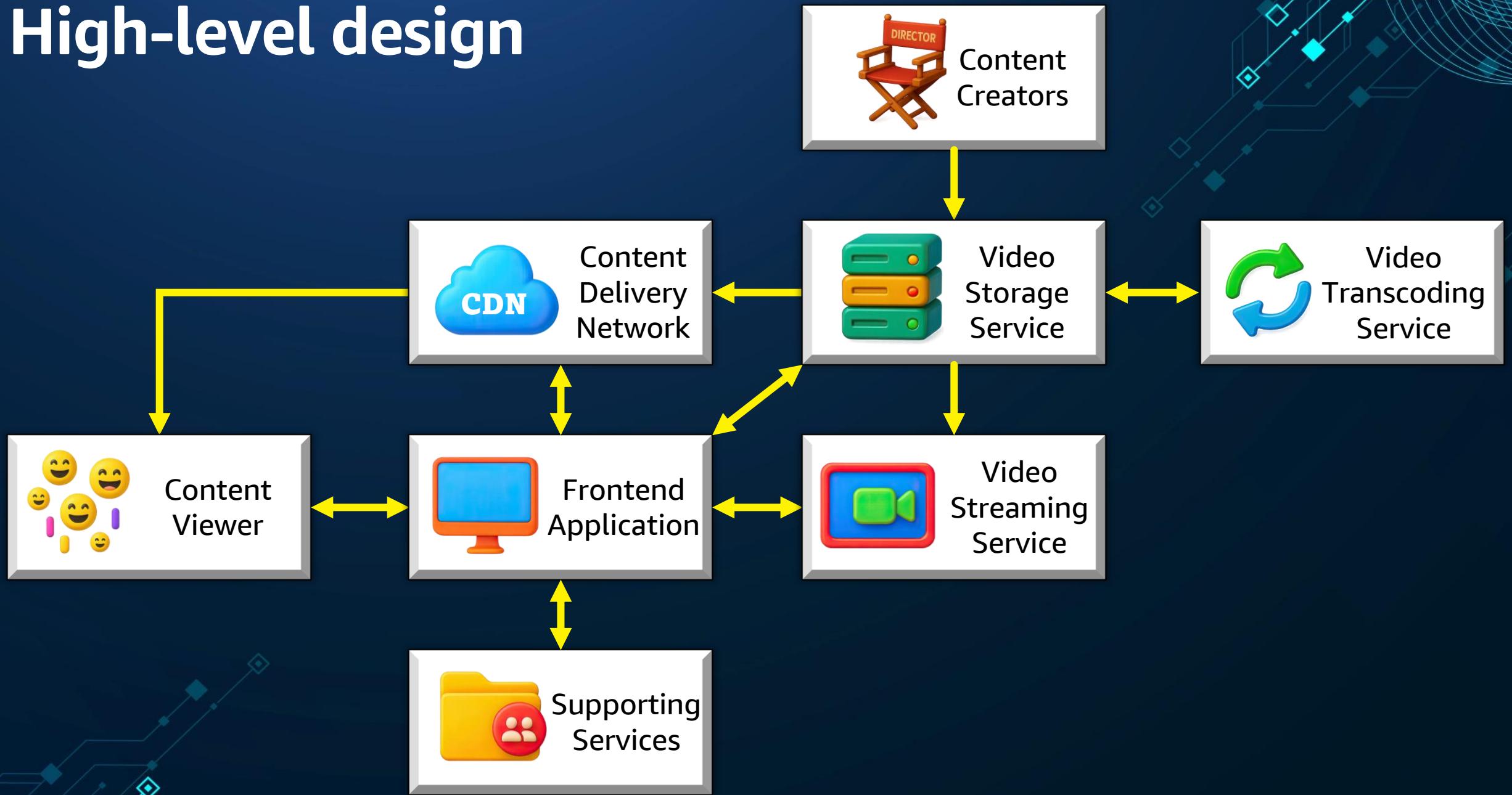
Customer Support & Feedback

Ticketing System Integration

Chatbot Integration

## How Can I Help You?

# High-level design



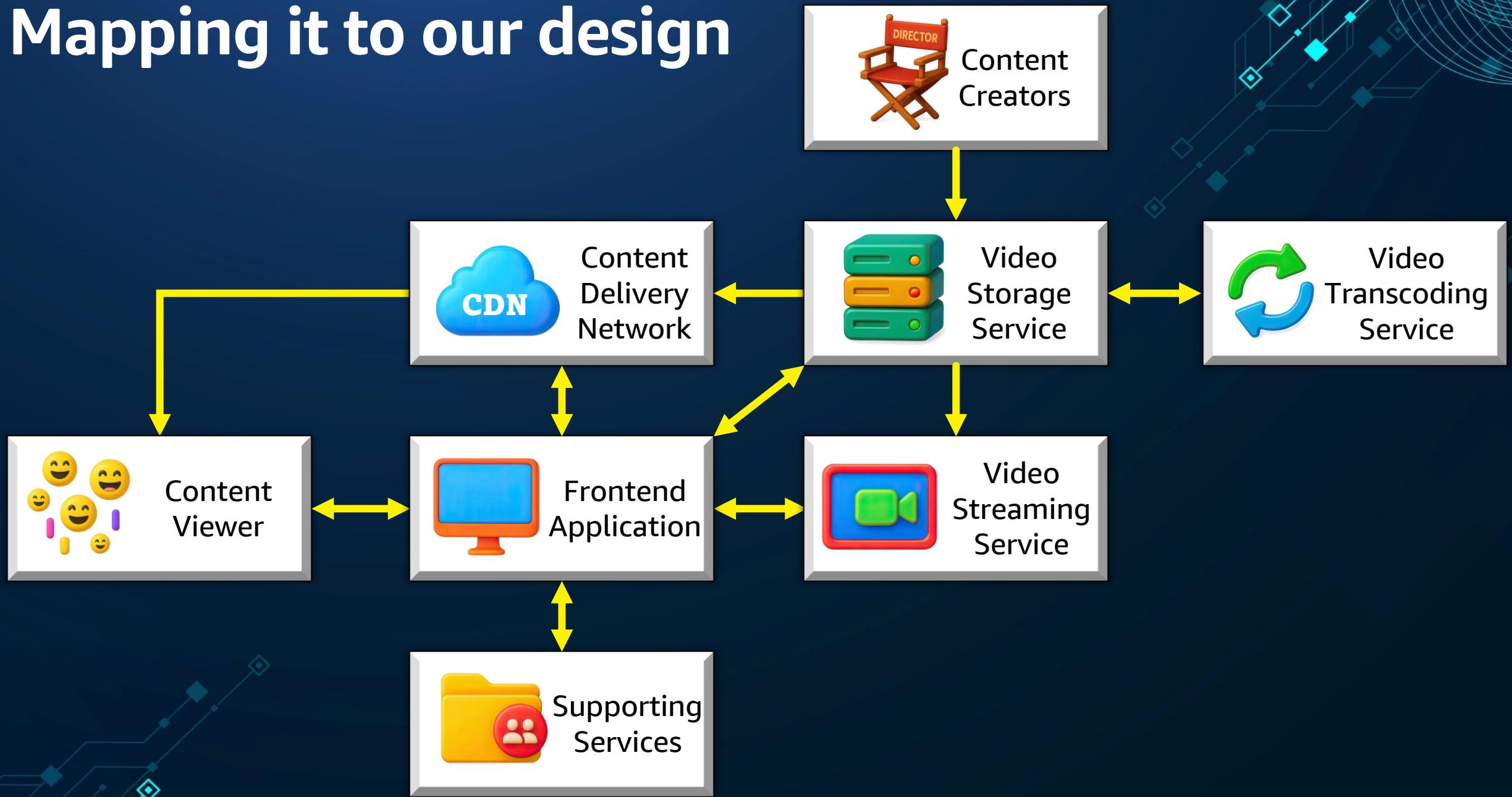
# Episode 9: Techflix behind Netflix



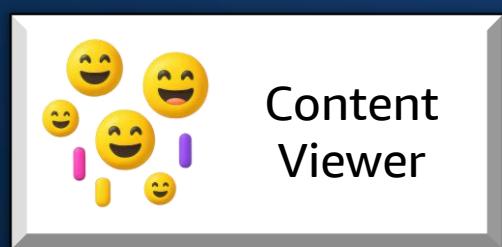
# Netflix's Key Architectural Principles



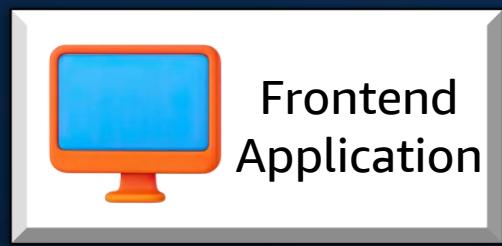
# Mapping it to our design



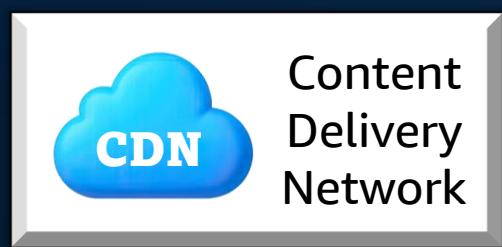
# Mapping it to our design



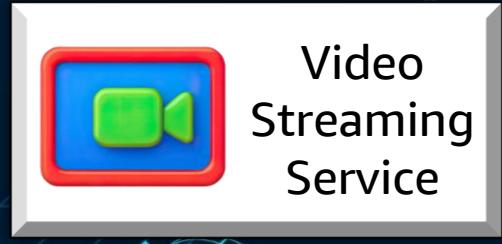
- Adaptive Streaming Algorithms (BOLA, DASH)
- MSL (Message Security Layer) (Encryption)



- React.js (Web)
- React Native/ Kotlin / Swift (Mobile)
- GraphQL/Falcor (APIs)



- Netflix Open Connect (Proprietary CDN)
- AWS CloudFront (Supplemental)



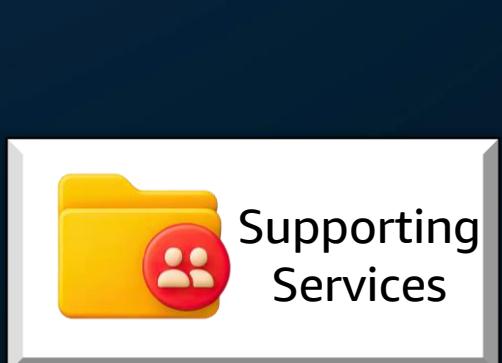
- Zuul (API Gateway)
- EVCache (Distributed caching)
- Titus (Container orchestration)



- Amazon S3 (Primary storage for source videos)
- Cassandra, DynamoDB, CockroachDB (Metadata)



- AWS Elemental MediaConvert
- C++ (native libraries) – Encoding



- Microservices/Runtime
  - ▶ Archaius, Karyon, Governator, Fenzo
- Container & Orchestration
  - ▶ Titus, Apache Mesos
- DevOps & CI/CD
  - ▶ Asgard, Spinnaker, Jenkins, Docker, Gradle
- [Any many more...]

# Netflix Technology Stack

Category	Technologies/Services Used
Mobile App	React Native (partial), Swift (iOS), Kotlin (Android), Proprietary adaptive streaming
Front End	React.js, Node.js, GraphQL, Falcor (Netflix's data-fetching library)
Video Storage	Amazon S3 (source files), Open Connect Appliances (OCA) for edge caching
CDN	Netflix Open Connect (primary), AWS CloudFront (supplementary)
Recommendation Engine	Apache Spark, TensorFlow, PyTorch, Scikit-learn, A/B Testing framework
Video Transcoding	AWS EC2, FFmpeg, Conductor (orchestration), Dynamic Optimizer (per-title encoding)
CI/CD	Spinnaker (CD), Jenkins (legacy), Kayenta (canary analysis), Atlas (monitoring)
Data Warehousing	Snowflake, Amazon Redshift, Apache Iceberg, Presto/Trino
Observability & Monitoring	Atlas (metrics), Vector (logs), Edgar (failure analysis)
Event Streaming	Apache Kafka, Keystone (Netflix's stream processing pipeline)
Container Orchestration	Titus (Netflix's container management on AWS)
Security	Lemur (certificate management), MSL (encryption for streaming)
Content Delivery Optimization	Dynamic Adaptive Streaming over HTTP (DASH), Netflix's BOLA algorithm
Workflow Orchestration	Conductor (Netflix's microservices orchestration tool)
Cloud Infrastructure	AWS (EC2, S3, DynamoDB, RDS, Lambda), Netflix's own CDN (Open Connect)

# Reference



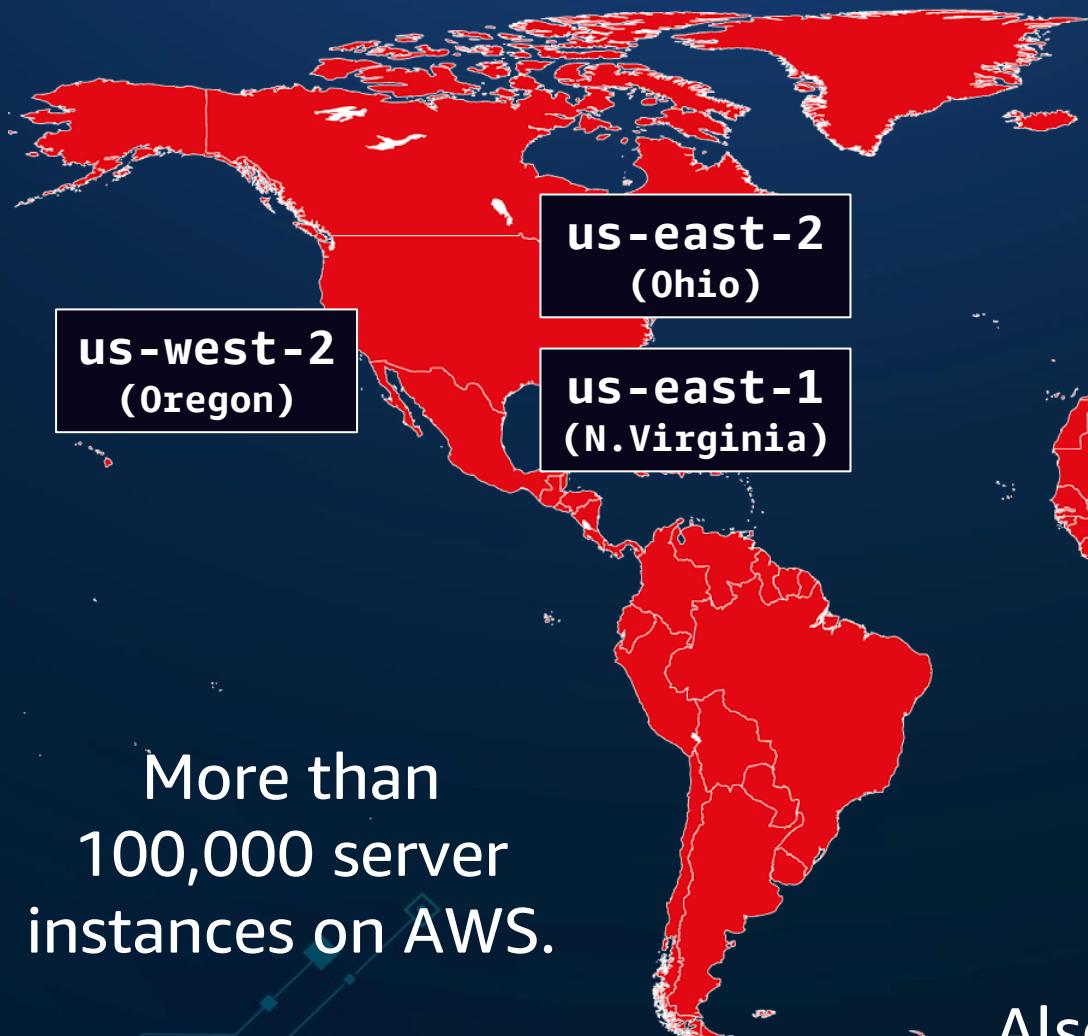
THE NETFLIX  
TECH BLOG



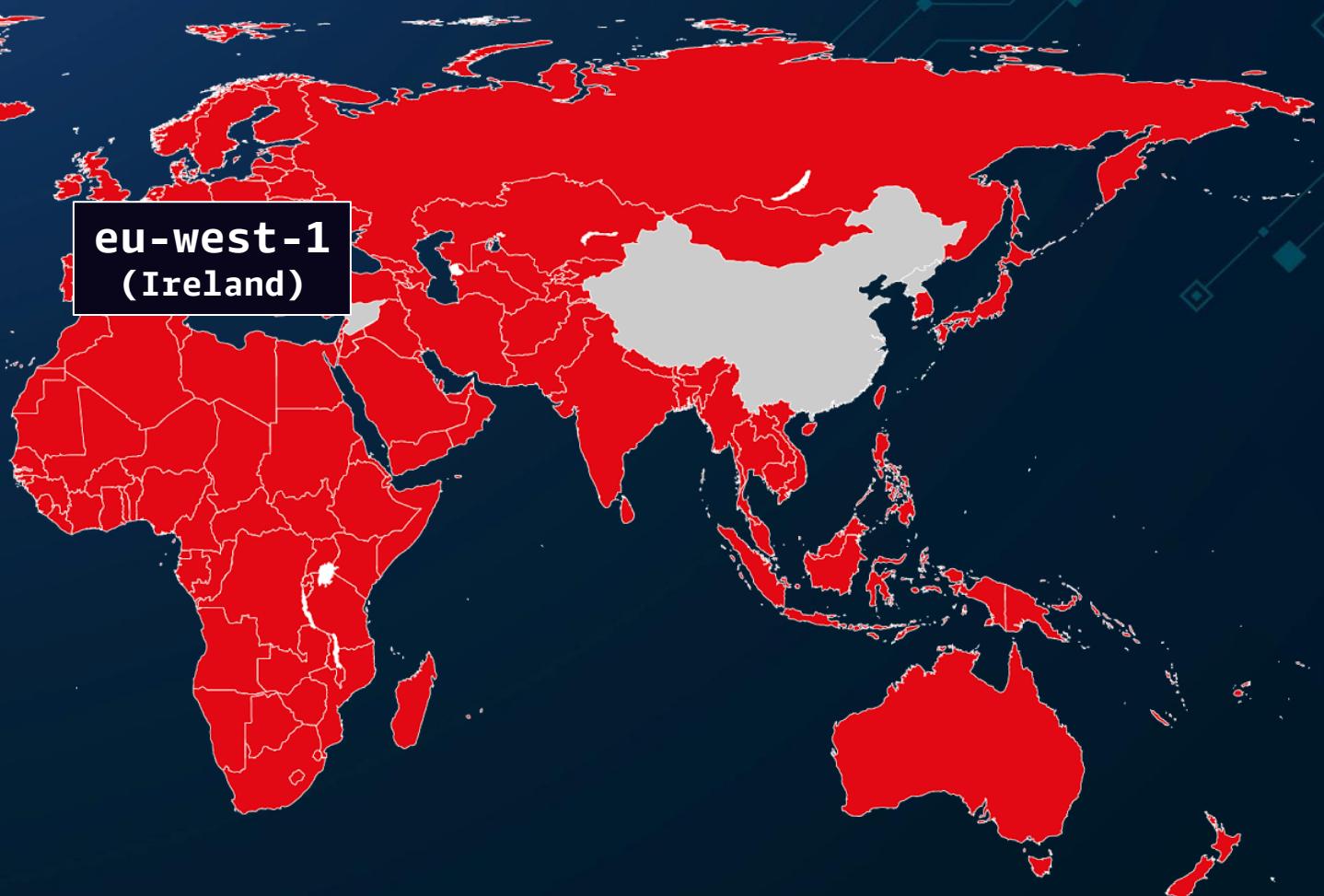
# Episode 10: The Cloud Behind the Curtain



# Netflix on AWS

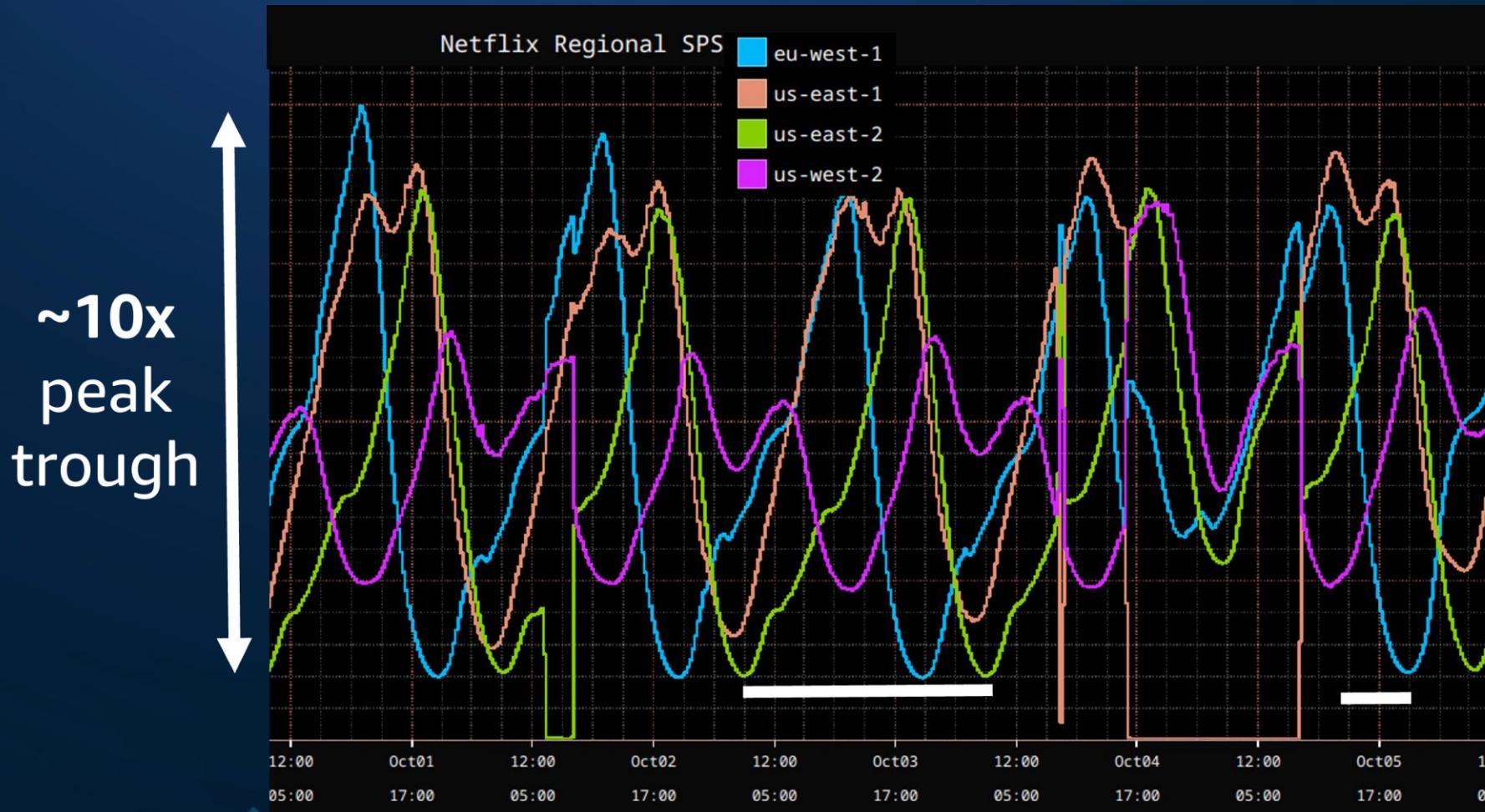


More than  
100,000 server  
instances on AWS.



Also uses AWS Local Zones for deploying visual effects studio (virtual workstation) closer to artists.

# Gradual traffic increases are the norm

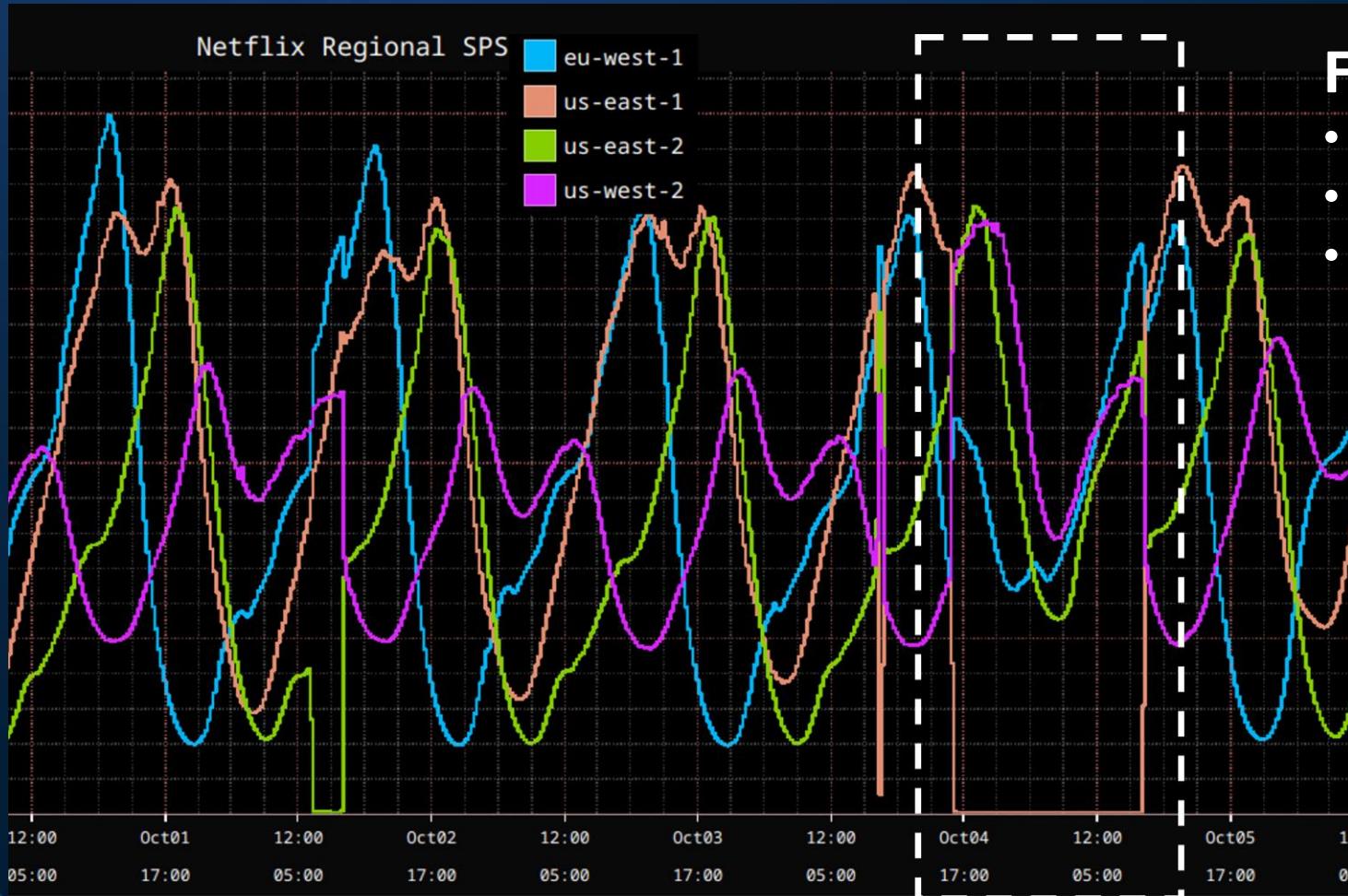


24-hour  
periodicity

Traffic  
phase shifts

Starts per  
second (SPS)

# Load spikes are common

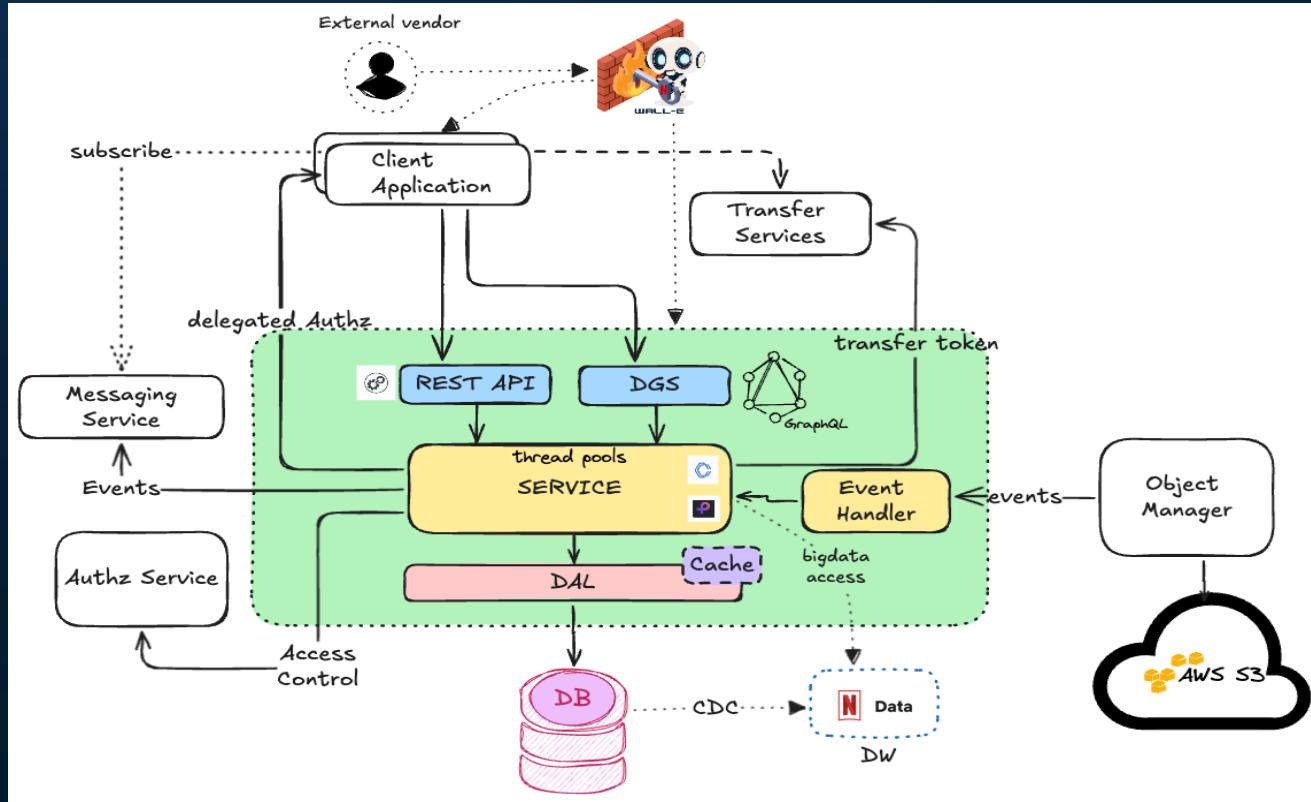


## Failovers due to:

- Regular practice
- Bad software deployment
- Regional impairment

# Content Drive (CDrive)

- Content Drive (or CDrive) is a cloud storage solution that provides file/folder interfaces for storing, managing, and accessing the directory structure of Netflix's media assets in a scalable and secure way.



# Hybrid multi-tier smart storage

Hosted on Amazon's multi-tier storage infrastructure



Netflix's in-house smart storage algorithm

## Smart categorization

Identify which data and when it belongs to which storage tier

## Custom rules

Have more control over setting custom rules to allocate storage based on user needs

## Easy Retrievals

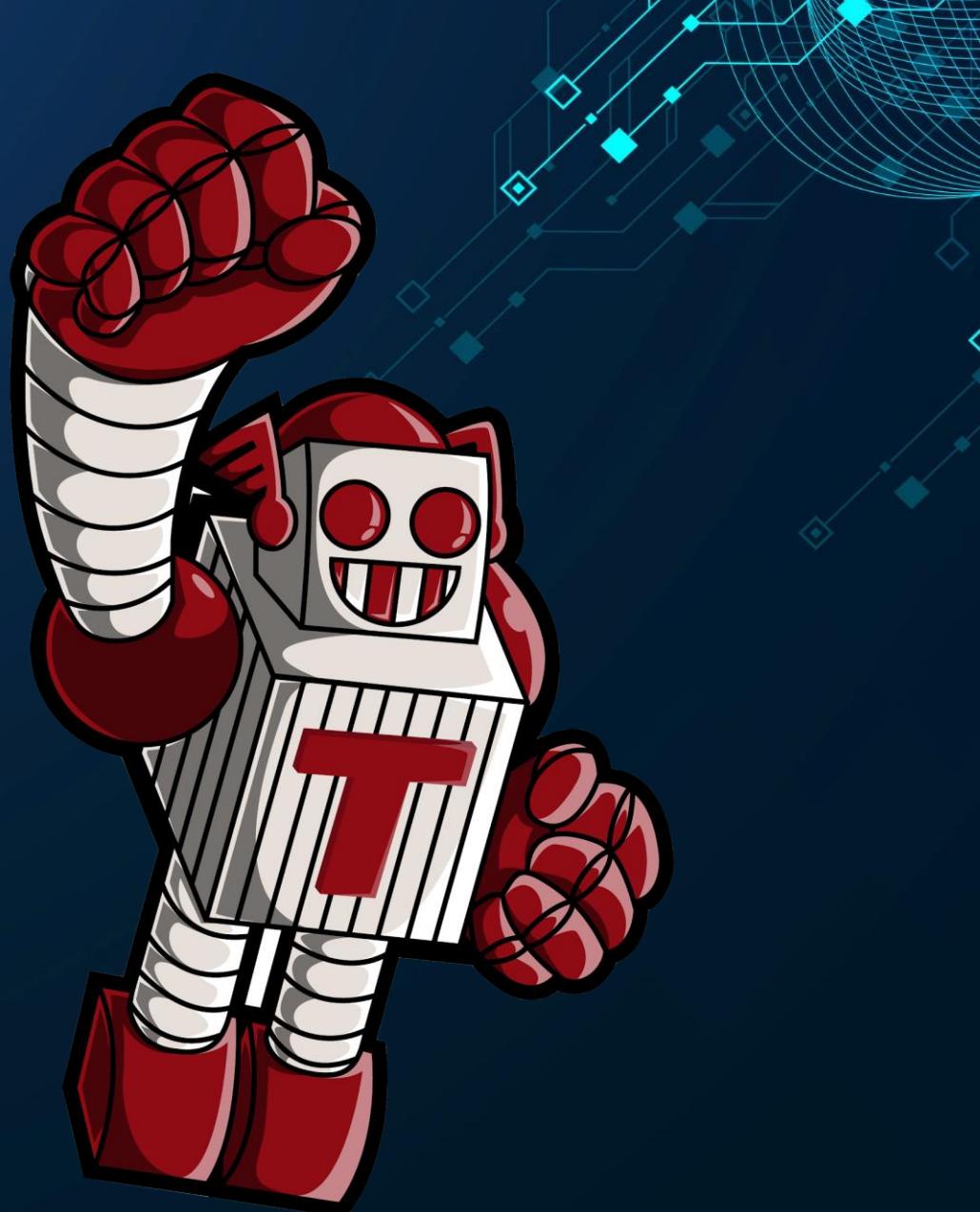
Due to custom data storage the data retrieval becomes fast and easy – many times data could be intelligently restored before being requested

## Reusable framework

Portable and reusable framework applicable to unmanaged and edge storage

# Titus

- Netflix's multi-tenant container orchestration system with tight AWS integration
- Titus uses Kubernetes APIs.



# IPMan

- Netflix's container IP address assignment service.



# AWS Services

(Partial List)

Compute & Serverless	
Amazon EC2	AWS Lambda
AWS Auto Scaling	Amazon EKS

Storage & Data Management	
Amazon S3	Amazon RDS
Amazon DynamoDB	Amazon Cassandra

Analytics & Streaming Data	
Amazon Kinesis	Amazon EMR
Amazon Redshift	AWS Glue

Networking & Content Delivery	
Amazon CloudFront	Amazon Route 53
Amazon VPC	AWS Global Accelerator

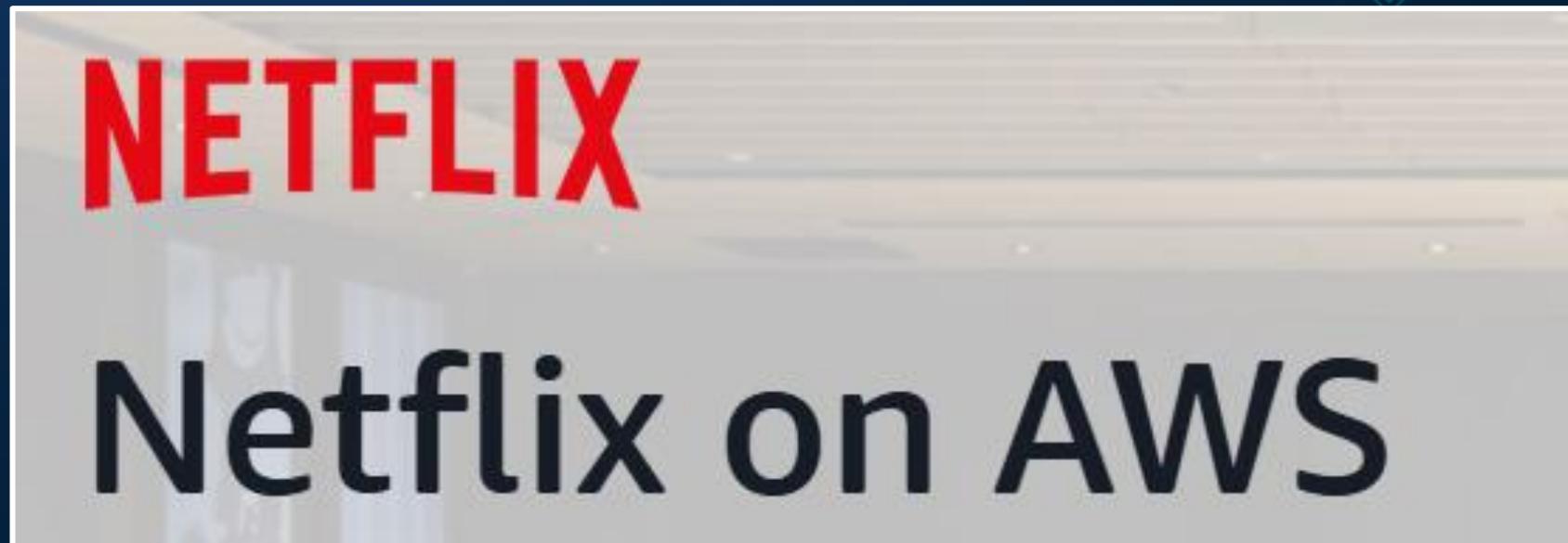
Messaging & Queuing	
Amazon SQS	Amazon SES

Security and Identity	
AWS IAM	AWS KMS

Media Processing	
AWS Elemental MediaConvert.	

Machine Learning & Personalisation	
AWS SageMaker	

# Additional Reading

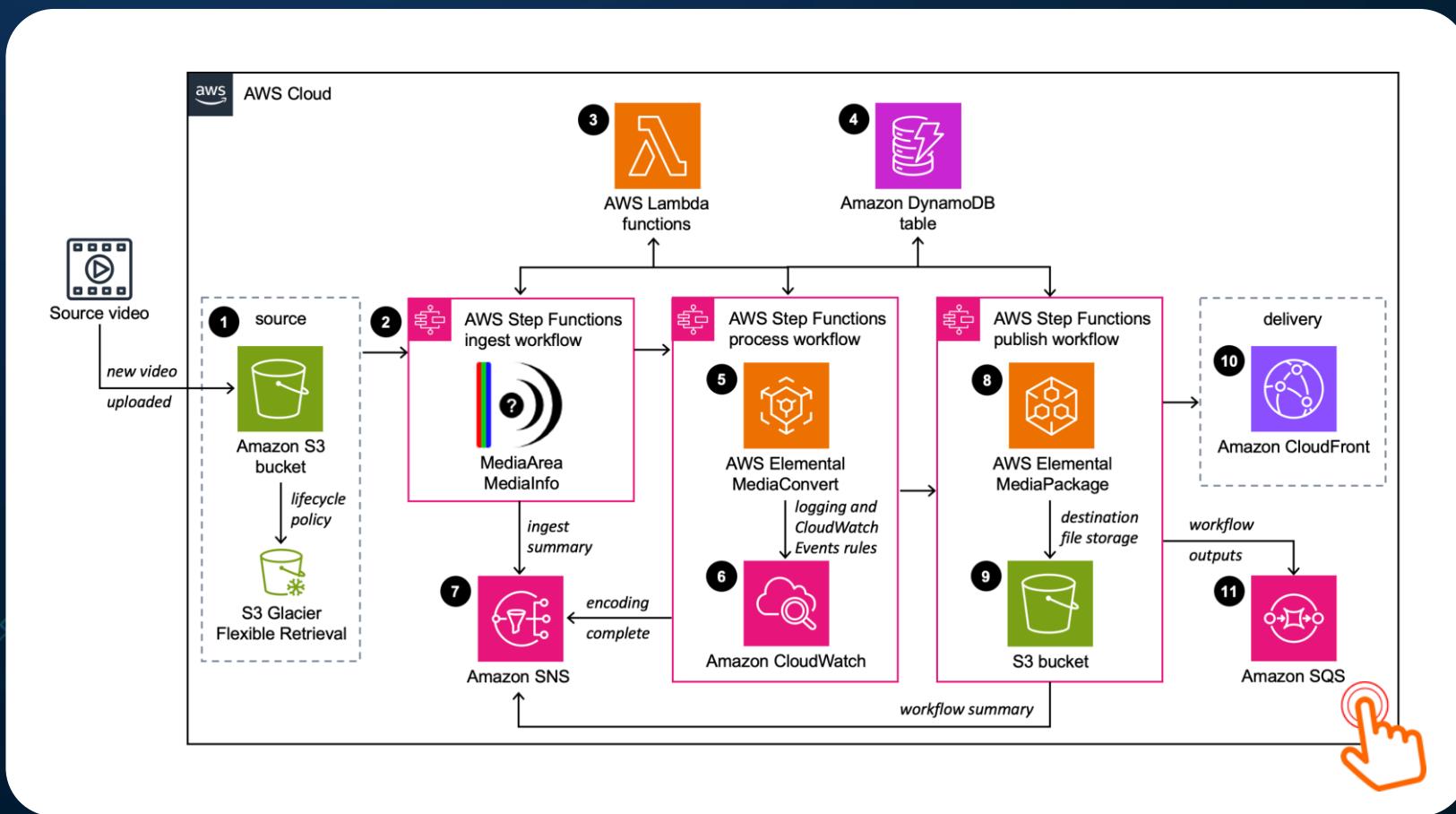


# Episode 11: Build Your Own Mini Netflix



# Reference - Video on Demand on AWS

- An example architecture to build a global consumer video workflow on AWS.



# References

- Blog - [Back to basics: HTTP video streaming](#)
- Blog - [Live video streaming using Amazon S3](#)
- Workshop - [File Streaming with AWS Media Services](#)
- AWS Solutions Library - [Live Streaming on AWS](#)

# Episode 12: Streaming Must Go On



# Addressing Non-Functional Requirements.

## High Availability

Users expect 24/7 access across regions with minimal downtime.

## No Buffering

Smooth playback and responsiveness are critical to user experience.

## Reliability

Content should never be corrupted or lost; uptime and video integrity matter.

## Security

Protects user data, content, and enforces subscription and geo restrictions.

## Scalability

Must support millions of concurrent users and growing content libraries.

# High Availability (HA)

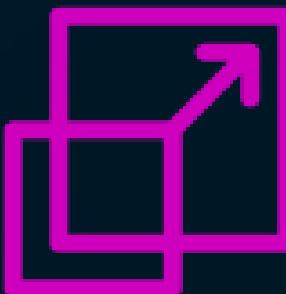
- **Goal:** Ensure minimal downtime with 24/7 access globally.
- **Solutions:**
  - Multi-AZ and Multi-Region Deployments:
    - Use Elastic Load Balancer (ELB) across multiple Availability Zones (AZs).
    - Replicate services like EC2, ECS, Lambda, and RDS across Regions.
    - Use Route 53 for DNS failover.
  - Amazon S3 for video storage offers 11 9's durability and is regionally resilient.
  - Amazon S3 cross region replication.
  - RDS Multi-AZ or Aurora Global Databases for relational DB HA.
  - Auto Scaling Groups for EC2-based components.

# Low Latency & No Buffering

- **Goal:** Deliver smooth, responsive playback worldwide.
- **Solutions:**
  - Amazon CloudFront (CDN) for globally distributed content caching.
    - Integrate with S3 or MediaPackage as origin.
  - AWS Media Services:
    - MediaConvert for on-demand transcoding.
    - MediaLive + MediaPackage for live streaming with adaptive bitrate support.
  - WebSockets / AppSync for real-time interactions (comments, reactions) services.
  - Edge locations reduce first-byte latency.

# Scalability

- **Goal:** Handle millions of users and content growth.
- **Solutions:**
  - S3 scales automatically for video storage.
  - ECS/EKS/Lambda for microservices—scale based on demand.
  - DynamoDB or Aurora Serverless for highly scalable databases.
  - Kinesis for real-time analytics and user interaction ingestion.
  - Use Amazon Auto Scaling and Application Load Balancer (ALB) for frontend/backend services.



# Reliability

- **Goal:** Avoid video corruption, data loss, and ensure uptime.
- **Solutions:**
  - S3 versioning + Cross-Region Replication (CRR) for backup and failover.
  - Glacier for long-term archive storage.
  - Use CloudWatch + CloudTrail + X-Ray for end-to-end monitoring and tracing.
  - Step Functions and SQS/SNS for reliable service orchestration and messaging.
  - Implement circuit breakers, retries, and fallbacks for microservices.



# Security & Authorization

- **Goal:** Protect content, ensure access control, enforce geo/subscription rules.
- **Solutions:**
  - IAM for secure access control between services.
  - Cognito for user authentication and authorization.
  - Signed URLs & Signed Cookies in CloudFront for secure content access.
  - WAF + Shield to prevent DDoS and application attacks.
  - KMS for encryption of data at rest and in transit.
  - Geo-restriction policies in CloudFront to enforce regional licensing.





*To be continued...*

# Expand your design

- Multiple Profiles
- Payments
- Offline Download
- Parental Control
- Viewing History
- Digital Rights Management
- Content Delivery Network
- Analytics
- Recommendation Engine
- Etc..

