Media Streaming on AWS

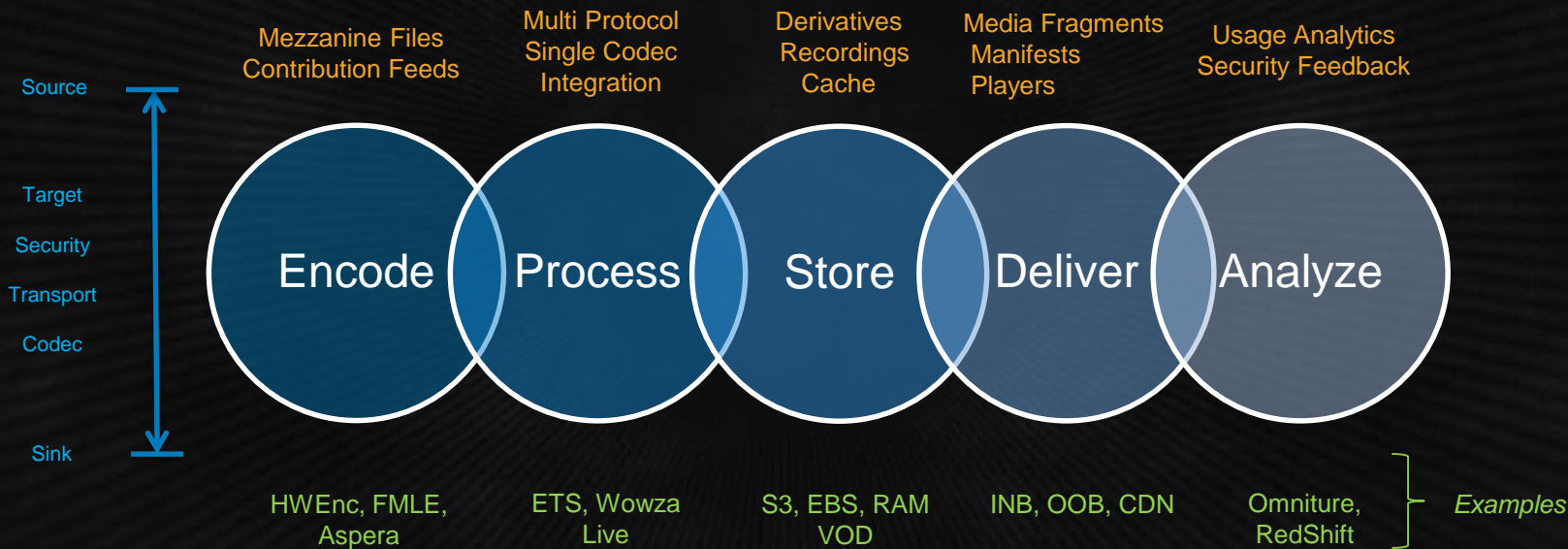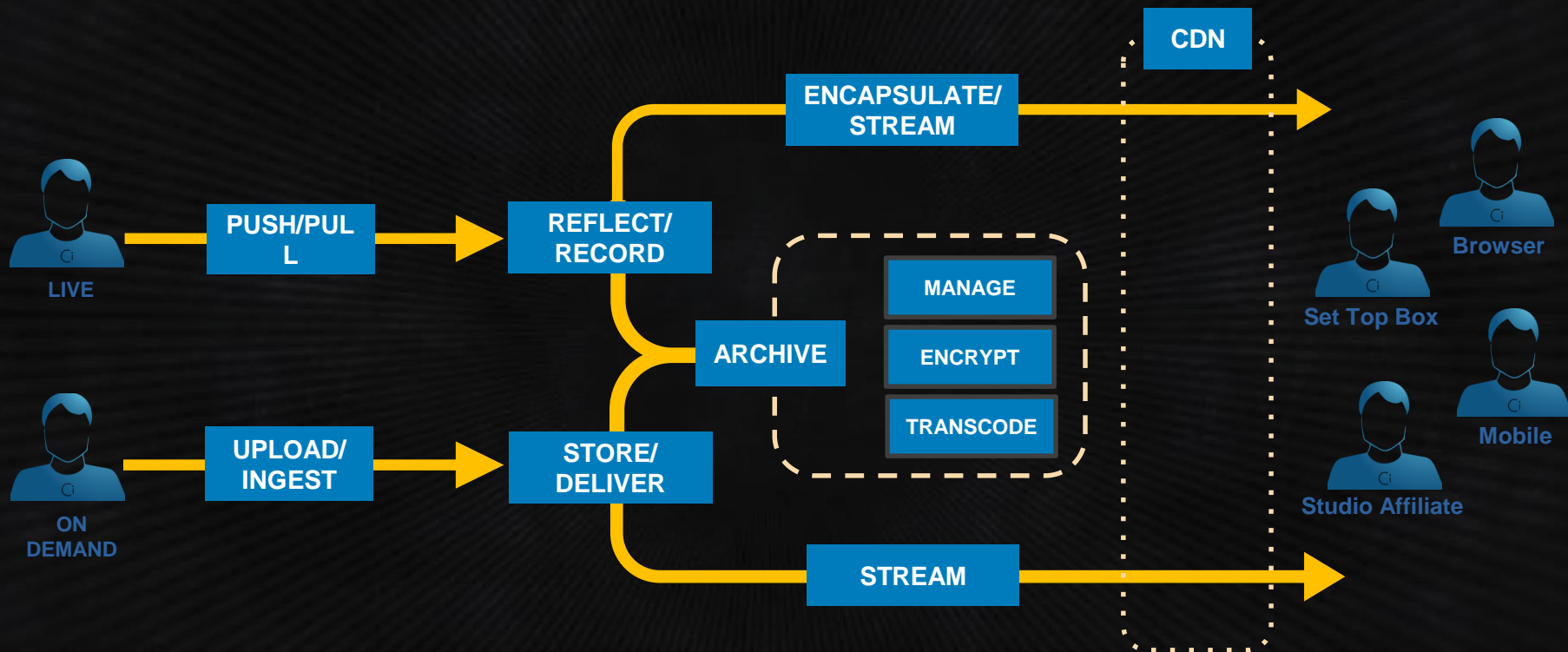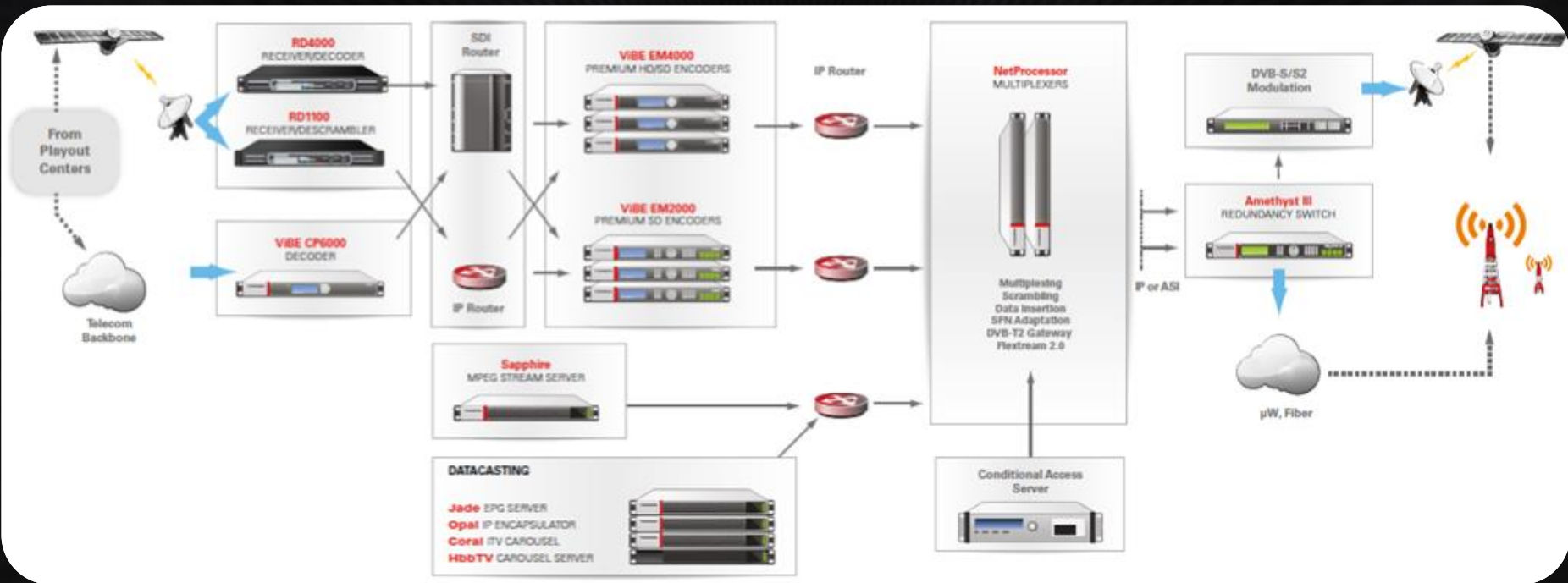## April 2015

Overview

High Level Concepts

Concepts

End-to-end multi-protocol, multi-codec delivery of live and/or on-demand media, targeting commercial and related OTT B2C multi-platform delivery models.

Workflow

Architecture

Architecture
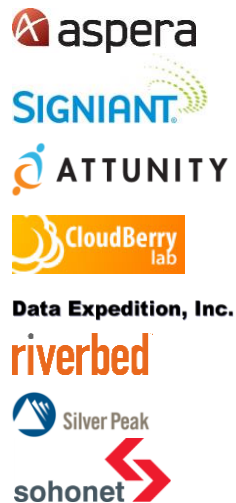

Streaming Origin / Edge Nodes

Chunks + Manifests

App Nodes

Keys

Live & On-Demand

Ingest

Recording / Archival

CloudFront Delivery

Chunks + Manifests

Mez + Derivatives

Encryption Key(s)

Edge Location Streaming / Download
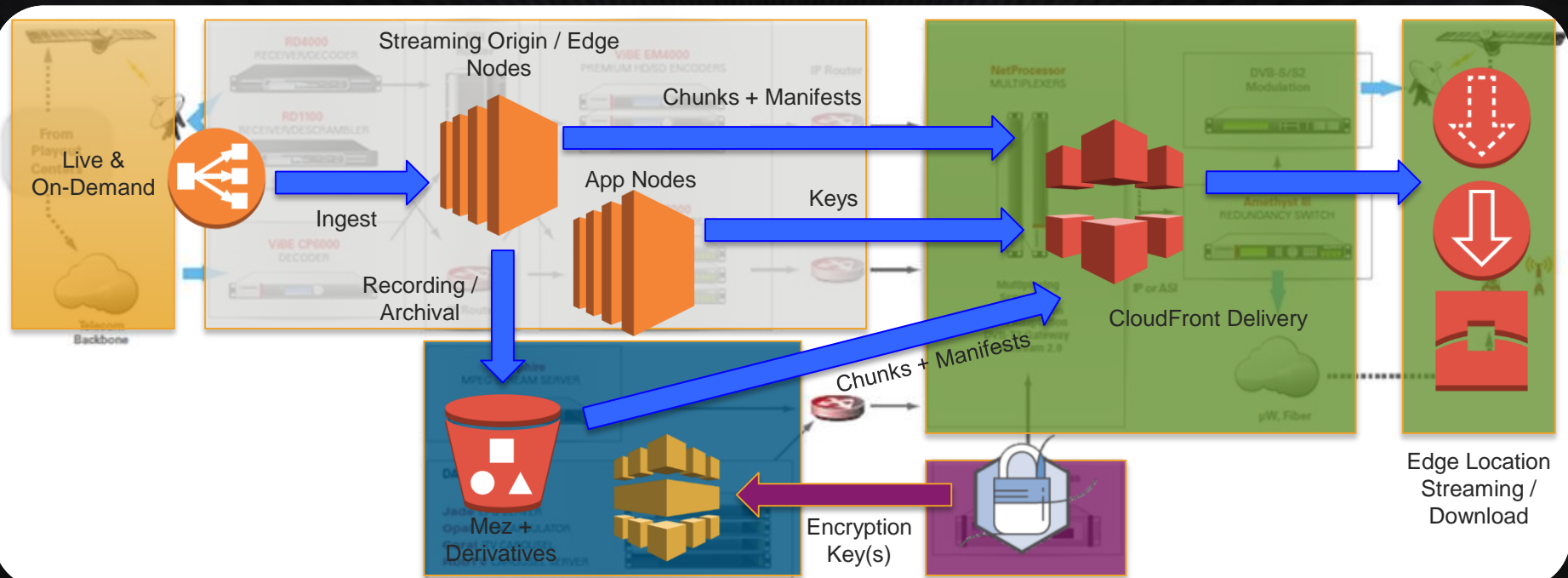
Partners

Solutions

- Ingest - Aspera (MP), Signiant (MP), TsunamiUDP (OSS)
- Streaming - Wowza (MP), Adobe AMS (MP), NGINX+ (MP), IIS (DIY EC2), USP (MP)
- Toolkits - FFMPEG/BC (OSS), GPAC/MP4Box (OSS)
- DRM - BuyDRM (MP), EzDRM (MP), CA/DRM (DIY EC2)
- Encoding - Haivision (MP), Harmonic (MP), Encoding.com (PaaS), Zencoder (MP), Elastic Transcoder ETS (Managed), Telestream (SaaS), Elemental Cloud (SaaS)
- Playback - JWPlayer (OSS), BitMovin (OSS), Other OSS (OSMF/OVP/VideoJS/etc.)
- Analysis - Manzanita (DIY EC2), Harris (DIY EC2), Sencore (DIY EC2)
- Reflectors - evoStream (MP), crtmpd (OSS), Haivison (MP)
- Security – AlertLogic (MP), OSSEC/Tripwire (OSS), TrendMicro (MP), AlienVault (MP)
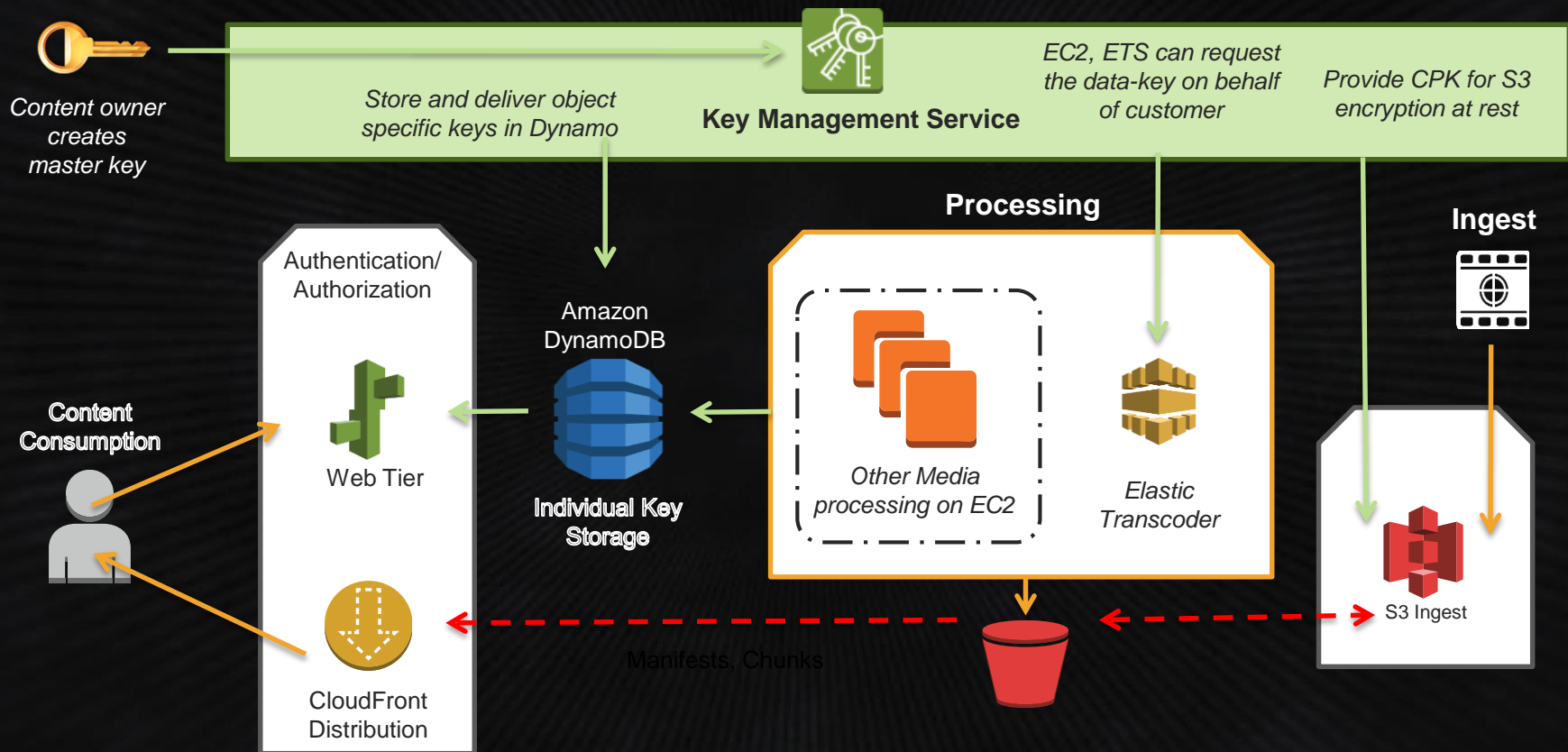
Media Streaming

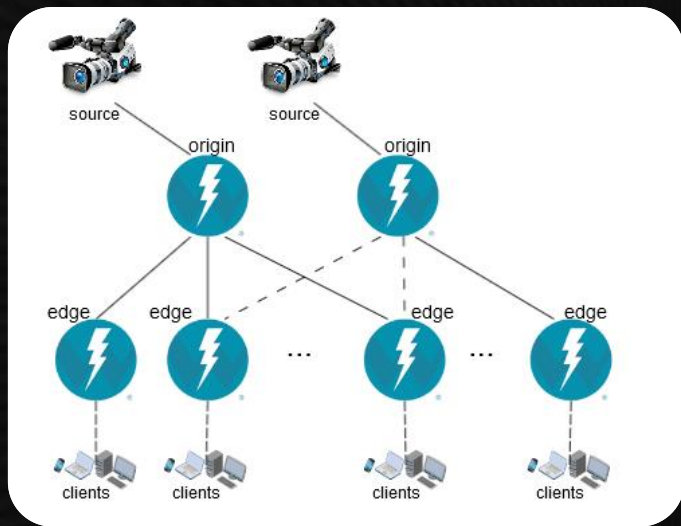Processing, Encryption, Streaming & Delivery

Processing

- **Pre-process**, (encrypt) and chunk with S3, ETS, Lambda, KMS
- Deliver **non-cacheable** data through native web tier + ELB or CloudFront
- Key fulfillment **OOB** with store on S3, DynamoDB, RDS
- **Cache** offload for Key, Manifest, Session, Chunks via ElastiCache (Redis or Memcached)
- Scale delivery of adaptive streaming and security through **CloudFront**
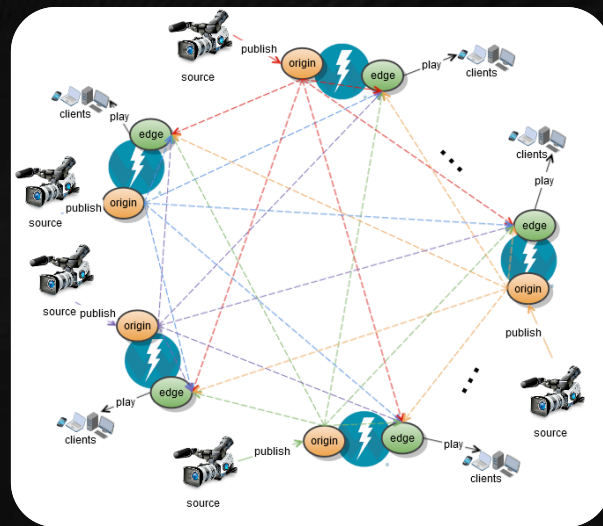- Advanced DDoS protection at the **CF** and **ELB** layers

Encryption

Content owner creates master key

Store and deliver object specific keys in Dynamo

Key Management Service

EC2, ETS can request the data-key on behalf of customer

Provide CPK for S3 encryption at rest

Processing

Ingest

Authentication/ Authorization

Amazon DynamoDB

Content Consumption

Web Tier

Individual Key Storage

Other Media processing on EC2

Elastic Transcoder

S3 Ingest

CloudFront Distribution

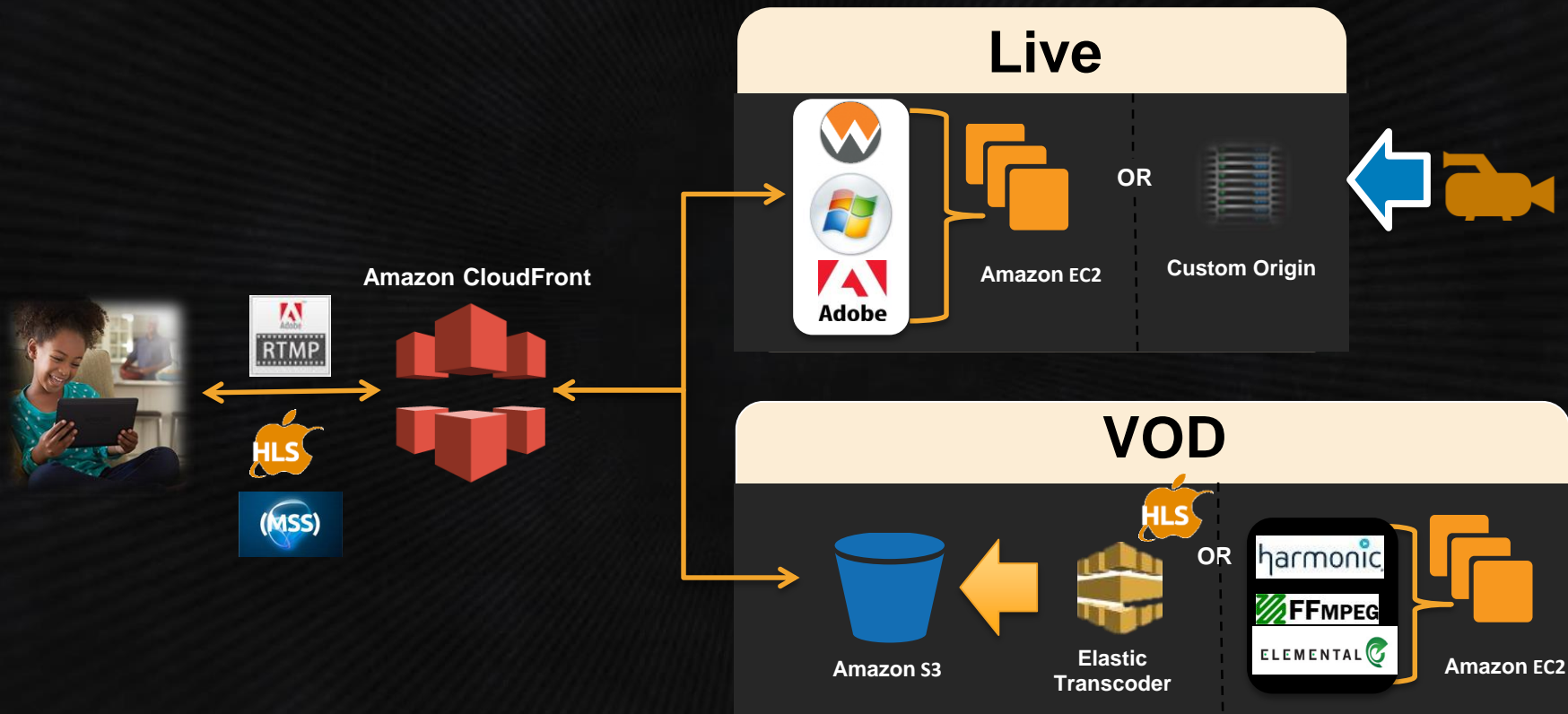Manifests, Chunks

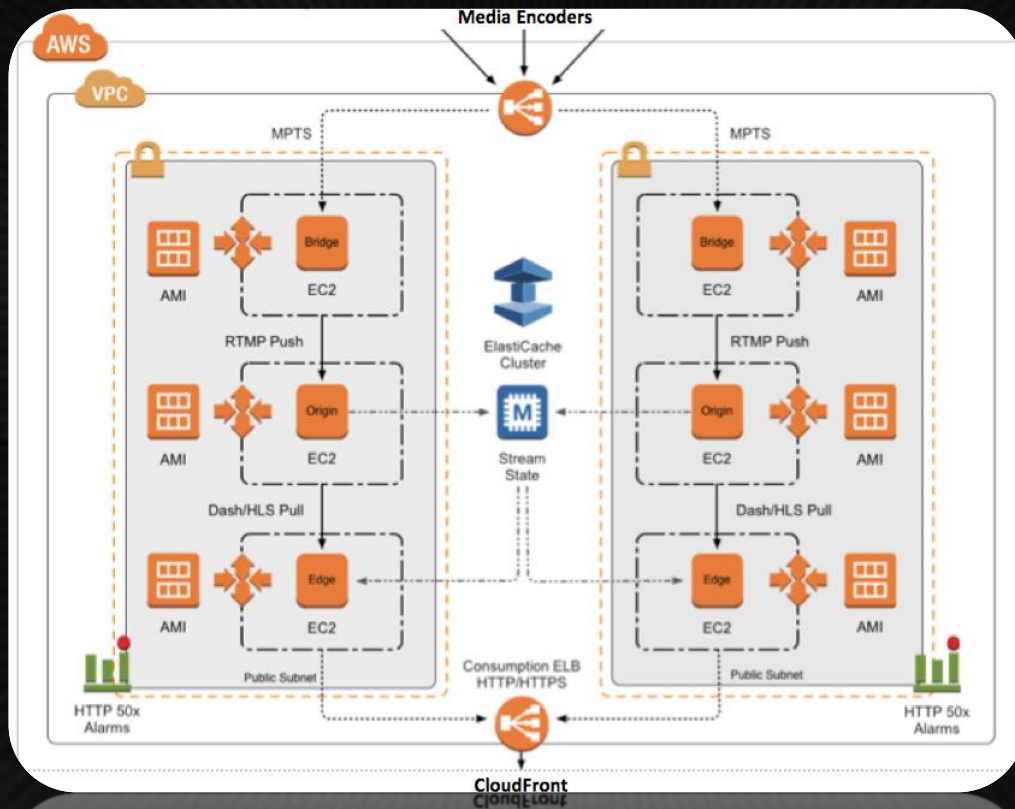# Patterns

Tree

Mesh

Streaming

Delivery

- Optimizing for **Live** Delivery, specific to HTTP, not simple
- Set **TTL** headers at **Origin**
- Set **frame-rate + key-frame** interval appropriately - 30fps, 4s/KF
- **Align** KFI desired to chunk size – 10s-12s chunk ~ 5 but 2,4,5,6s KFIs ok

- Preferred chunk size important – device decoders
- Tune chunk count per manifest request – cache/memory impact!
- Low TTL on Manifests, ~ chunk duration * 3 TTL on chunks
- Deliver **scoped** set level manifests

- Looping video – TTL too high! Pulsing video – starved encoder!

End to End

Thank You !