# MOQ Usages

ing audio/video applications over MOQT

**Cullen/Suhas/Mo**

V1

# Agenda for this session

This draft and presentation is only about followings architectural questions

1) Ways that audio/video/media maps to MOQT Objects
    - ( eg. frame per object, CMAF Segment or Chunk  per object, slice per object, … )
2) Which QUIC streams is used to send a given MOQT Object
    - ( eg. Stream per object, stream per group,  stream per priority, stream per track … )
3) Mapping of Simulcast / SVC video to MOQT Model
    - ( eg. One track all layers, Track per layer, … )

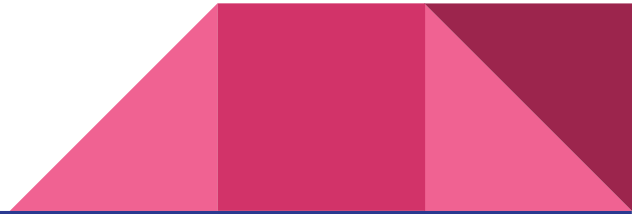
Have tried to document option that people have requested

Document some of the pro/cons

In 10 minutes do not expect us to resolve any of these, just look at framework for discussion going forward.

# Metrics to consider in looking at solutions

- How this interacts with Priority schemes
- Latency on lossy / constrained networks
- Bandwidth Overhead / Efficiency
- Complexity
- Future Flexibility



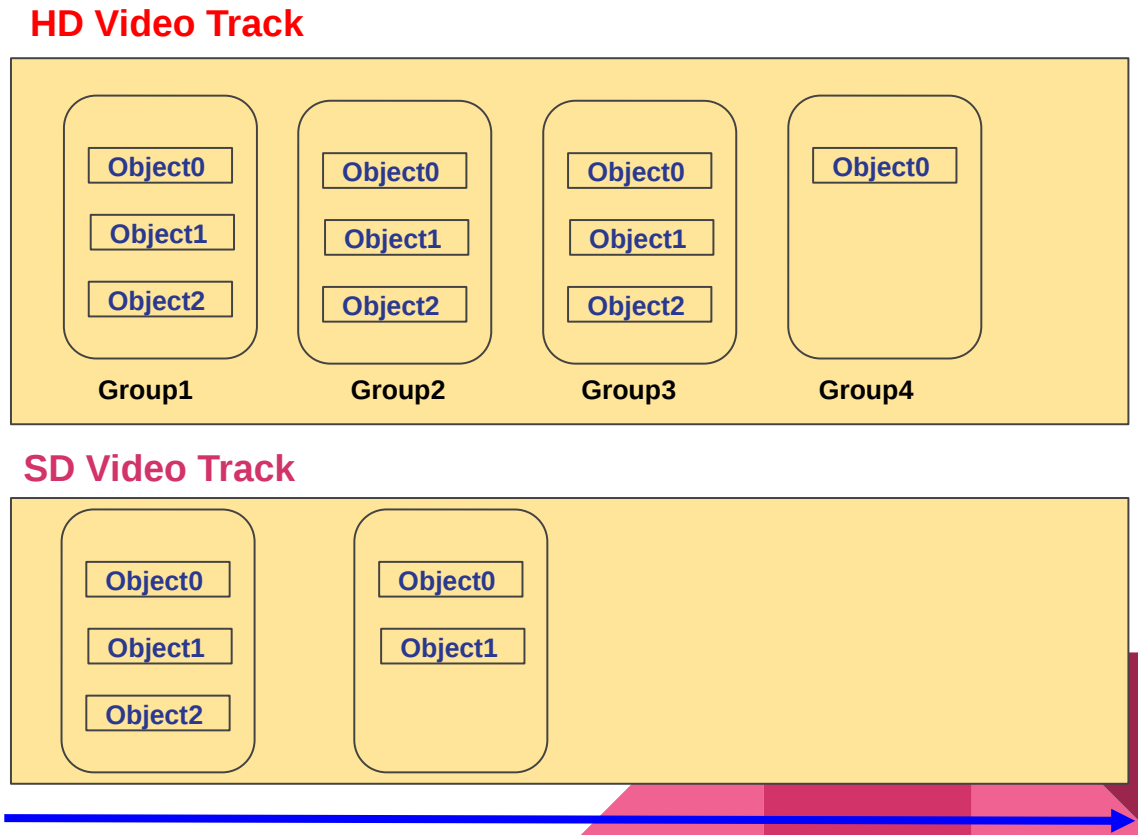
# Reminder about what MOQT defines today

MOQT identifies Tracks, Group and Objects as opaque entities

Applications define

- How application data us mapped to tracks/groups/objects

Also need to define how that is mapped to QUIC



# MOQT Object Model Mapping

Map **“Application Data Units ” → MOQT Objects/Groups/Tracks**

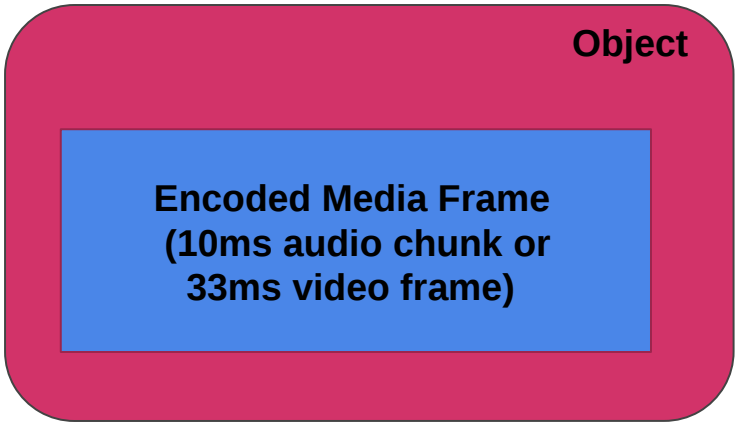
Application defined and controlled.

Agnostic to Relays.

Application Data Units can be :-

- Encoded Frame,
- Encoded Slice,
- CMAF Segment, CMAF Chunk,
- Something application specific that we haven’t seen so far

# MOQT Model - One Encoded Frame Per Object



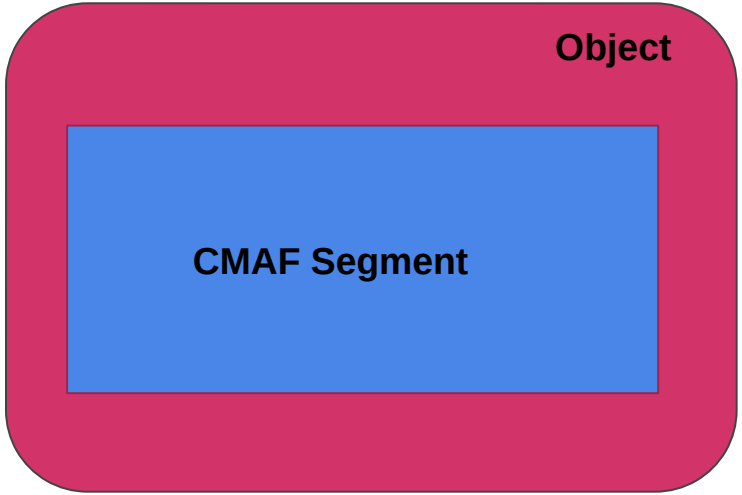

**Pros**

- Publish/Subscribe happens at Frame Boundaries
- Appropriate for Low Latency use-cases
- Allows application concealment at frame boundaries

**Cons**

- Consumer need to order out of order objects

# MOQT Model - One CMAF Segment per Object



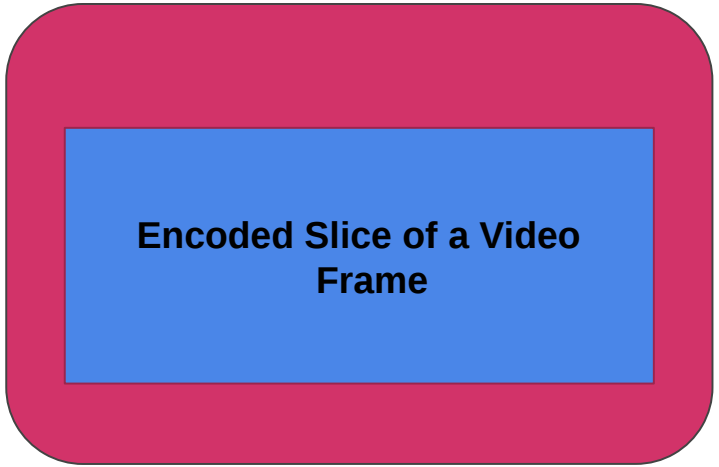

Pros

- Media delivered in decode order

Cons

- Object duration maps GOP duration
- Prone to HOL Blocking for object consumption

# MOQT Model - One Encode Slice per Object



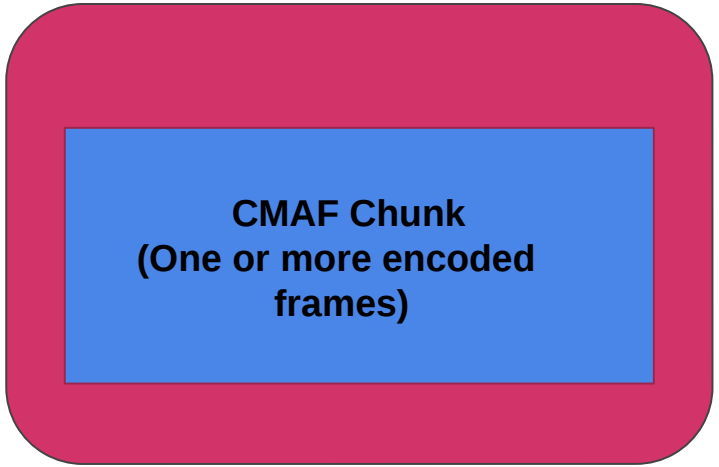

**Pros**

- Publish/Subscribe happens at Slice boundaries
- Appropriate for Low Latency use-cases

**Cons**

- More complex
- Out out of order handling needs to both frame and slice aware, at consumer

# MOQT Model - One CMAF Chunk per Object




**Pros**

- Publish/Subscribe happens at CMAF Chunk boundaries
  - Typically one or more Encoded Frames
- Appropriate for Low Latency use-cases

**Cons**

- Consumer need to order out of order objects

# MOQT QUIC Mapping

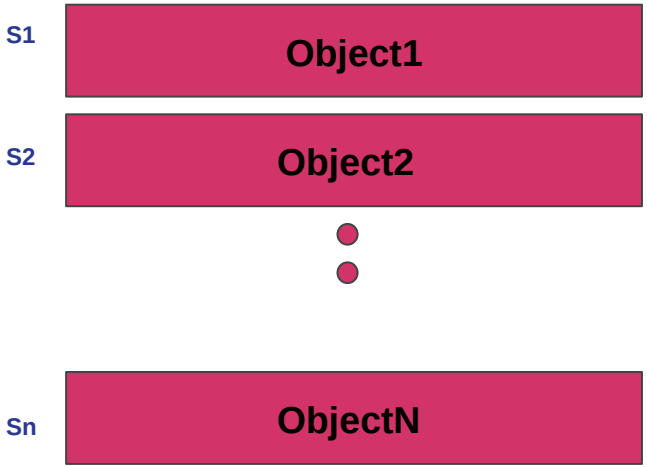
Map **MOQT Objects/Groups/Tracks → QUIC Streams**

Modifiable by Relays

Mapping options :-

- One QUIC Stream per Object
- One QUIC Stream per Group
- One QUIC Stream per Track
- One QUIC Stream per Priority
- One QUIC Stream per Multiple Tracks

# Transport Mapping - One QUIC Stream per Object



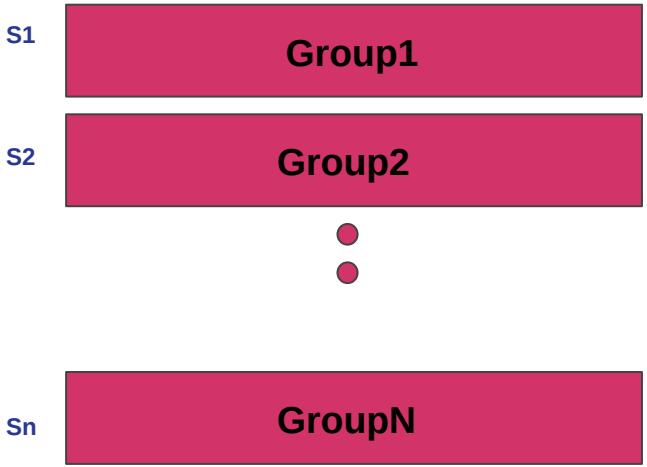

## Pros

- Less prone to HOL Blocking when Object is a frame/slice/chunk
- GOP  Object implies decoder order
- Relays cache at Frame/Chunk boundaries

## Cons

- GOP Object is more prone to HOL Blocking
- Lot of short lived streams when Objects are frames/slices/chunks
- Consumers need to handle out of order objects
- Relays cache at the GOP boudaries

# Transport Mapping - One QUIC Stream per Group



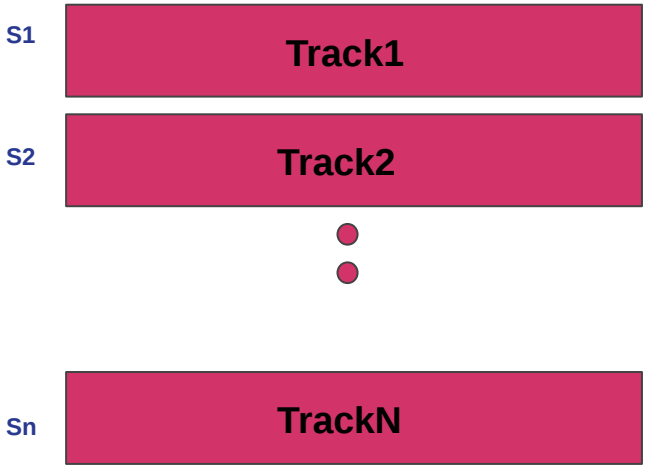

**Pros**

- Stream lifetime matches GOP Duration
- Less number of streams overall
- Group implies natural decoding order.

**Cons**

- More prone to HOL Blocking, esp when objects(s) duration is longer.
- May need complex & custom API when object is an entire GOP.

# Transport Mapping - One QUIC Stream per Track



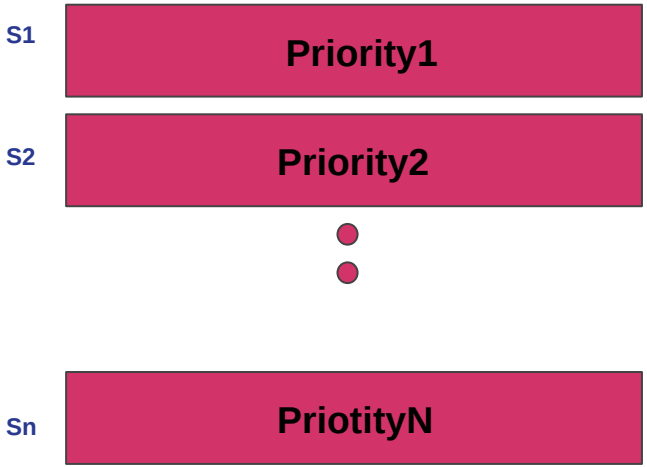

**Pros**

- Stream lifetime matches Track Duration → Lowest number of streams overall
- Least complex mapping, may serve best over well engineered links (say Datacenters)

**Cons**

- More prone to HOL Blocking under lossy networks
- May need complex byte stream API when object is an entire GOP

# Transport Mapping - One QUIC Stream per Priority




**Pros**

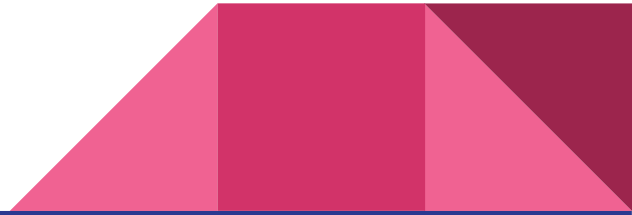
- Beneficial for low latency use-cases
- Number of streams map to priority levels set by the application

**Cons**

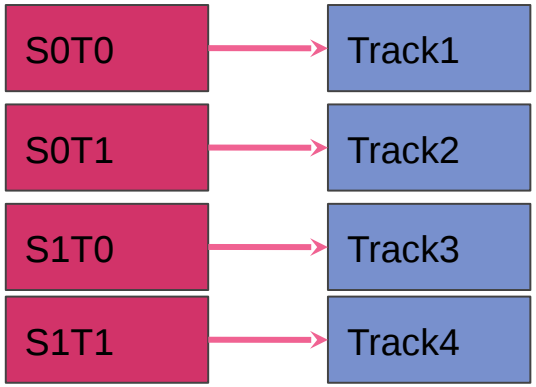
- Prioritization is not trivial, needs more study and experimentation

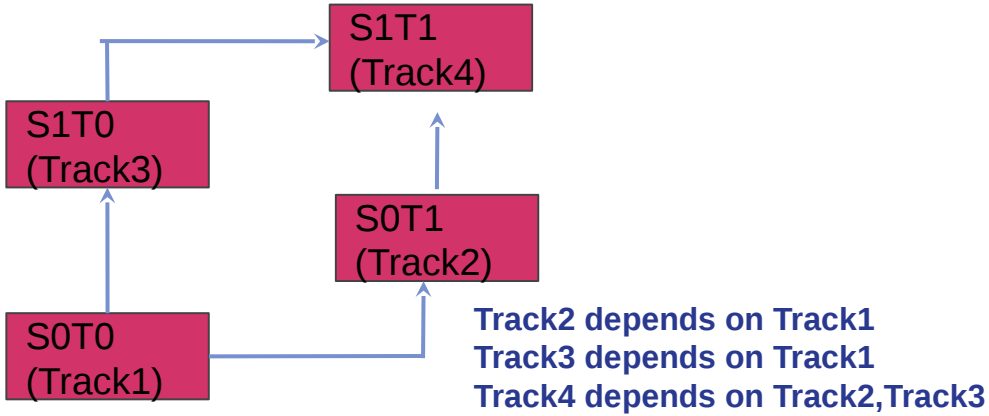
# SVC - All layers in one track

- Catalog indicates track as SVC source
- Dependency information for all the layers are encoded in the bitstream and/or container.
- Simple to support in MOQT object model
- However, lacks selective subscriptions of layers by the subscribers



# SVC - One Layer Per Track




**Track2 depends on Track1**
**Track3 depends on Track1**
**Track4 depends on Track2,Track3**

- Catalog indicates tracks, layers and dependencies
- Consumers use objects across related tracks
- Allows selective subscriptions across layers
- Allows congested links to drop layers

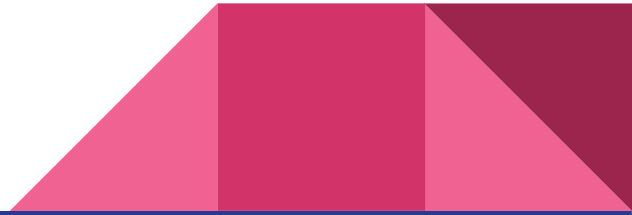
# Proposal

Next Steps

- Don’t decide now …
- Collect more implementation data of how this works
- Document pro/cons and results

Get this conversation to be data driven.

Use this information 3 to 6 months down the road to decide which technique or techniques need to be part of MoQ Transport.



# Backup Slides