Base Scenarios + moq

(Interactive Media and More)

Suhas/Christian/Cullen

What’s been happening out there ?

Thread on identifiers for Tracks :
https://mailarchive.ietf.org/arch/msg/moq/u_BgKcaQ49wzI0frIC6rlUj61yg/

Thread on Identifiers in general:
https://mailarchive.ietf.org/arch/msg/moq/UqG0nPGOB3lZVzBaKox2MK9TeZY/

Thread on object priorities vs delivery order and data model:
https://mailarchive.ietf.org/arch/msg/moq/u_BgKcaQ49wzI0frIC6rlUj61yg/

End to End encryption and object model thread:
https://mailarchive.ietf.org/arch/msg/moq/FzSZN7Ji7BdXTKj7pXsDzM2U538/

Webtranport and Native QUIC Thread:
https://mailarchive.ietf.org/arch/msg/moq/2GWHpohGHAQ9caUdVWz8yJ-IvyI/

Track Bundles related threads:
https://mailarchive.ietf.org/arch/msg/moq/KaFWG1aUF8rLVSAnW_lPLg8g100/

Receiver Driven focus (linked to bundles) :
https://mailarchive.ietf.org/arch/browse/moq/?index=u_BgKcaQ49wzI0frIC6rlUj61yg

Issues/PR
Track Bundles and Track uniqueness Related:
https://github.com/kixelated/warp-draft/issues/103
https://github.com/kixelated/warp-draft/issues/101

Authz Subscribe/Publish
https://github.com/kixelated/warp-draft/pull/96
https://github.com/kixelated/warp-draft/pull/97

Updated Data Model and Relay Support
https://github.com/kixelated/warp-draft/pull/95
https://github.com/kixelated/warp-draft/pull/69
https://github.com/kixelated/warp-draft/pull/67

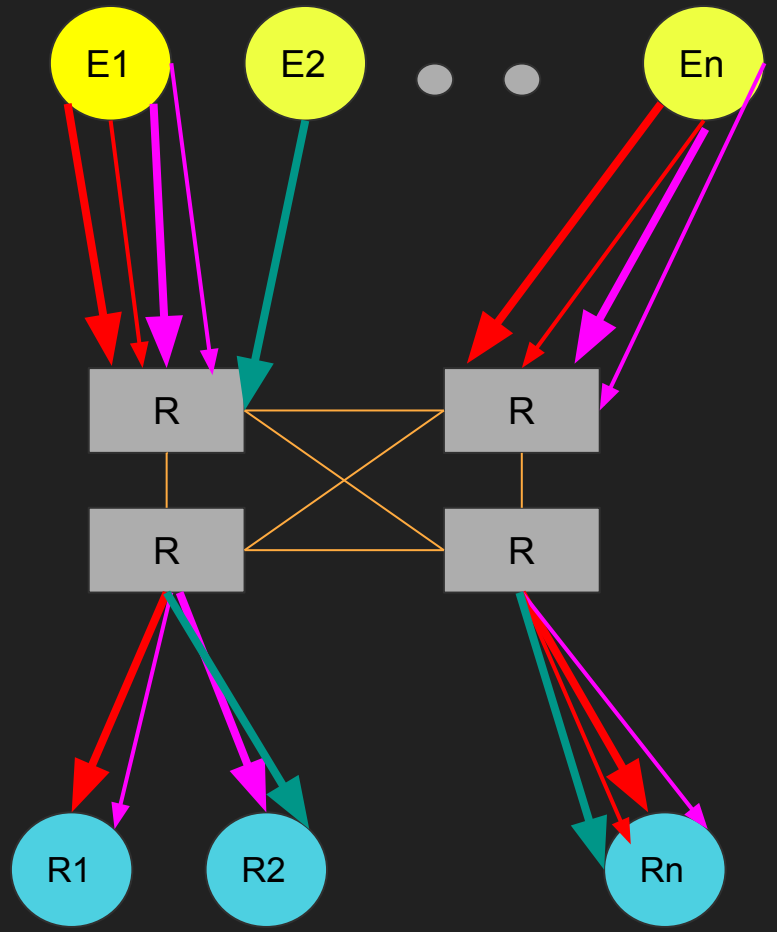
Priority and congestion response at Relays
https://github.com/kixelated/warp-draft/pull/93

Scenario(1)

Multiple Emitters - Multiple qualities

Relay Network - High Speed Fanouts to large number of receivers across differing network conditions

Multiple Receivers - Various Capacities



Scenario(1) - Requirements

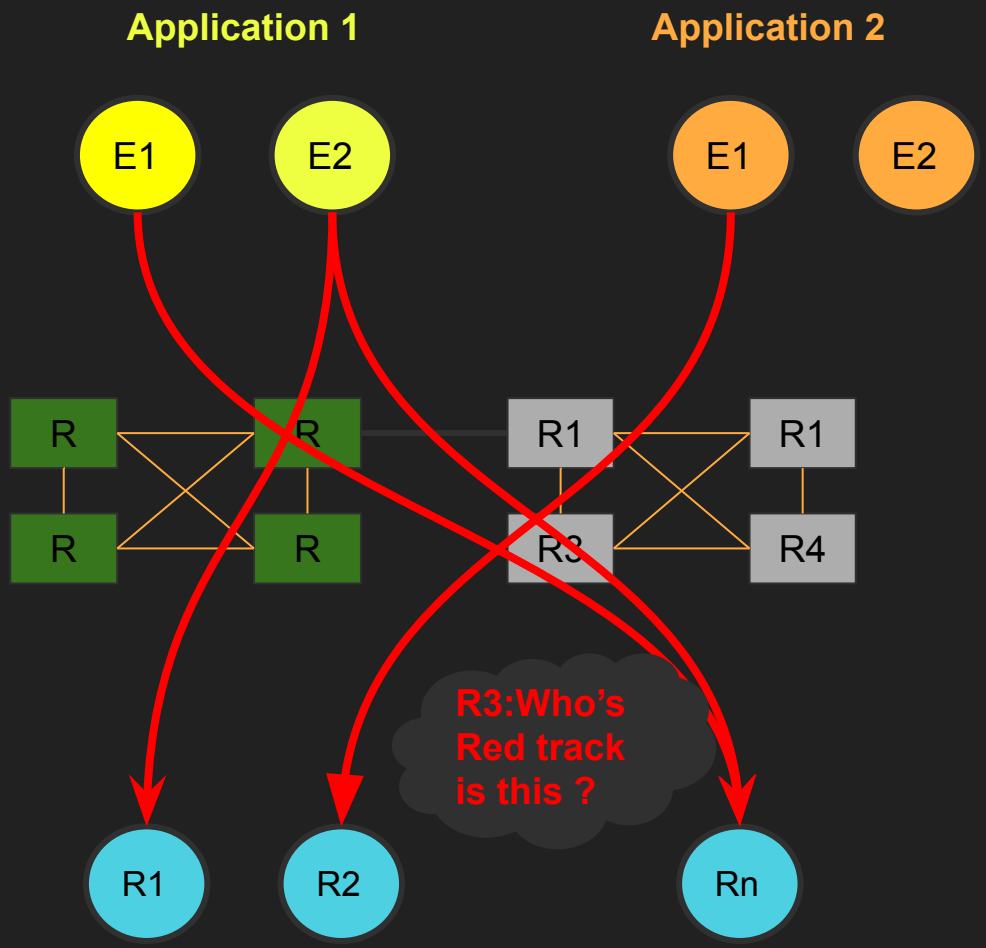
- Unified Ingest and Distribution System
  - Track Identifiers are visible through the system - **Consensus is forming**
- Multiple Emitters, Multiple Media Types, Multiple Qualities
  - Tracks can be **dynamically** added and removed.
  - Emitters join and leave.
- Media Switch
  - Receives track bundles from multiple emitters.
  - Split bundles into individual tracks
  - May compose a new catalog - retain tracks, add new tracks (Active Speaker…)
  - Forward Preference Order needs to be coordinated uniformly across all the tracks
- Relay Mesh
  - Need flexibility Merge and Split Bundles across one or more QUIC connections
  - Need flexibility to pick and choose quality or respond to quality queries for tracks
  - Forward Preference Order needs to be coordinated uniformly across all the tracks
- Multiple Receivers across different capacities and network conditions
  - Receivers conditions changes dynamically - quick switch will be needed
  - Catalog received can represent original emitter tracks or virtual tracks

Scenario(2)

Emitter emit multiple tracks with identifiers

Relays publish and subscribe on track identifiers

Receiver subscribe to tracks identifiers

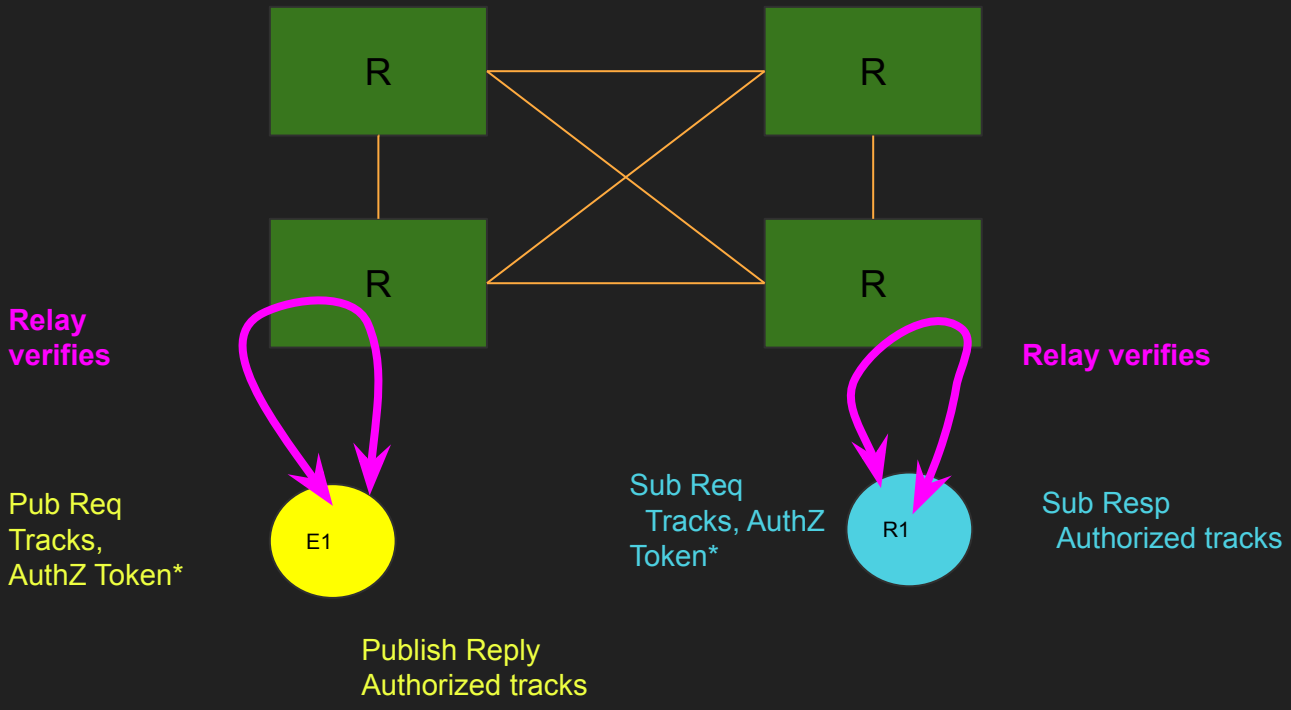


Scenario(2) - Requirements

- MoQ must support media flows **within and across** distribution networks
  - Unambiguous track identifiers
  - Uniform protocol behavior across hops

- Track must be **independently published to or subscribed from**
  - Should work over Native QUIC Session, without WebTransport Session ID
  - Object header need to carry shorter versions of Track to keep the sizes small

Scenario(3)


R
R
R
R
Relay verifies
Relay verifies
Pub Req Tracks, AuthZ Token*
E1
Sub Req Tracks, AuthZ Token*
R1
Sub Resp Authorized tracks
Publish Reply Authorized tracks

Scenario(3) Authorization Requirements

- **Emitters and Receivers need permissions to publish/receive on set of tracks**
  - Application provide a token to the emitters/receivers that can be verified by the applications and is tied to a given track namespace.
- **Relays are the policy enforcement points to participate in publishing or subscribing to tracks**
  - Catalogs are opaque to Relays
  - Relays need a way to know the authorized tracks (in publish/subscribe)

Interactive Media, Layered Encoding

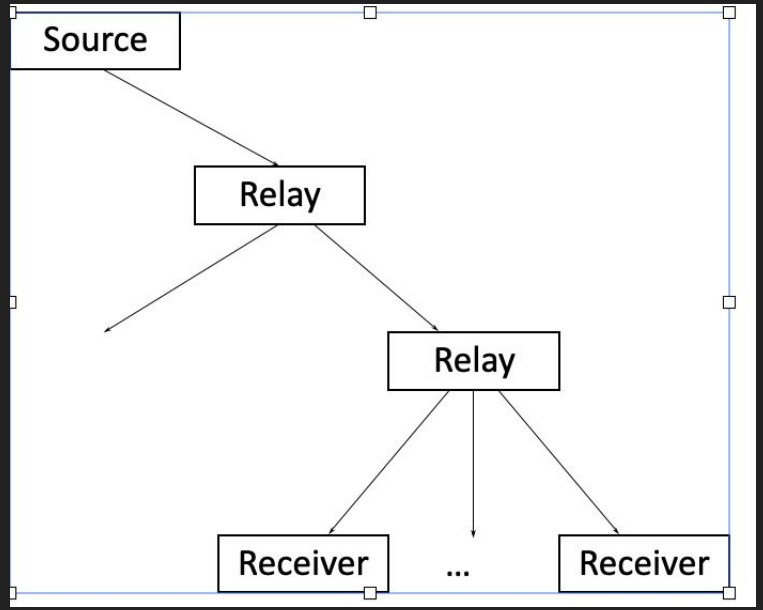
Large groups, large fan out

- Different conditions at each receiver

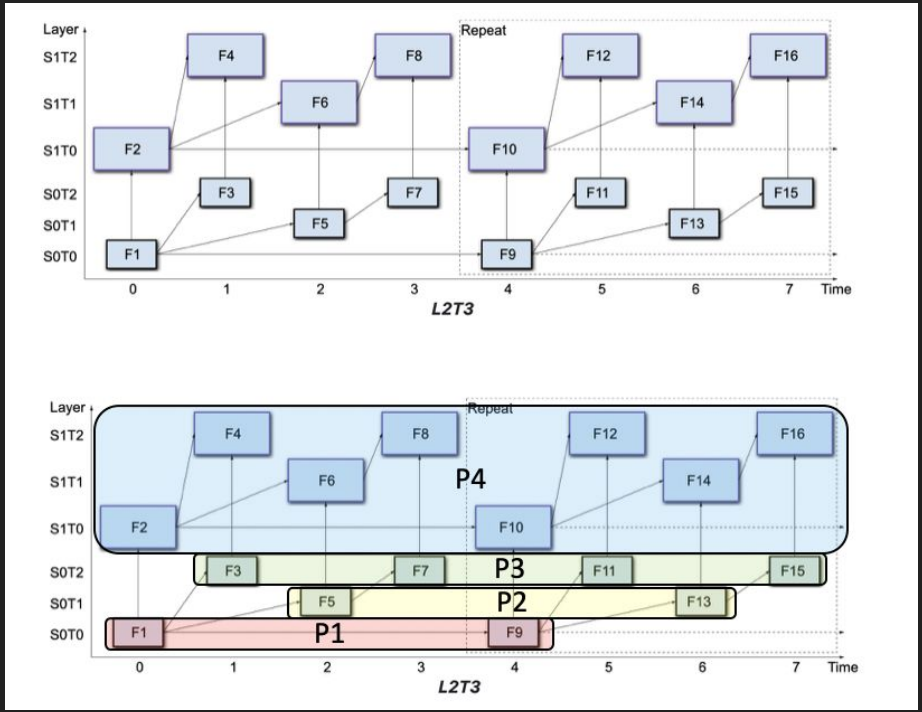
Two solutions

- Adapt at client: unsubscribe track A, subscribe to track A’ with different bitrate
- Adapt at relay: use layered encoding, peel layers when client is slow

MoQ should (also) support “layered” (simulcast/svc)



Layered Encoding → Object Number and Priority



- We need two variables per object:
  - Object number within a group (e.g., F1, F14, etc)
  - Priority level
- Priority requires a choice about what is most important
  - e.g. definition or frame rate
- Few “drop priority” levels required
  - Compatible with RFC 9114
- See scenario & transport drafts for details

← Ex: Frame rate first grouping

Object Number + Priority → Scheduling

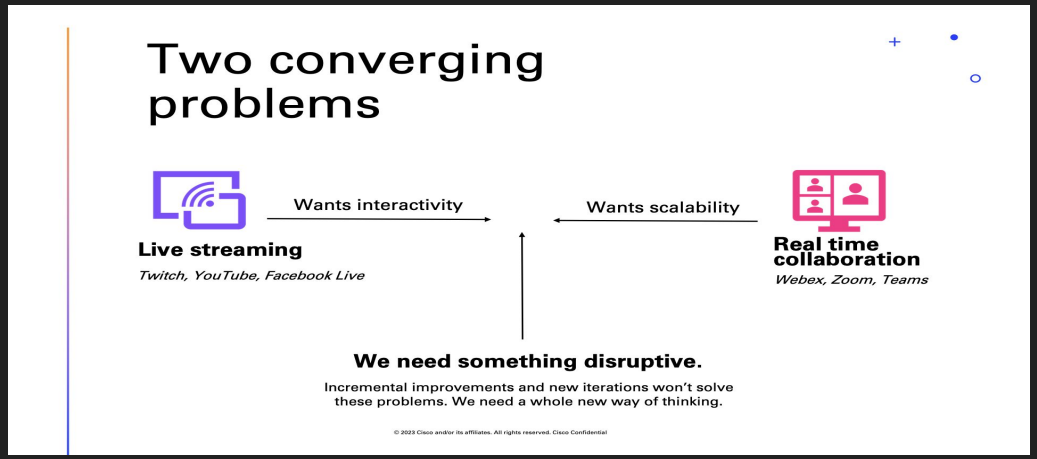
- In same group, **SEND** A before B if:
  - priority (A) < priority (B)
  - Or
    - priority (A) == priority (B)
    - And
    - Object number(A) < Object number (B)
- After group boundary
  - Consider dropping priorities of previous group streams, or resetting streams.

- Two plausible mappings
  - One stream per group and priority level
  - One stream per object.
    - Requires FIFO within one priority level
- If multiple tracks and coordinated priorities:
  - One stream per group + priority: use round robin between priority level
  - One stream per object: require FIFO within priority level, thus coordinated scheduling.

Objects in MoQ

- Objects include **timed media &** can be one of these
  - Media Frame (output of encoder)
  - Media Slice (part of a media frame, slice-based encoding)
  - Tile in Volumetric Media
  - Group of pictures
  - Deadline Aware Block
  - Game State & more
- Objects are **uniquely named** (scoped to a moqsession, track and group)
- Relays **cache individual objects** for serving subscriber queries (get game state object1234)
- Objects are **pipelined** through the relays
- Some applications will E2E encrypt the object with **AEAD ciphers**
- Object queries **MAY support byte-ranges*** (as additional parameter byte-range=20-200)

MoQTransport (moqt)
Unified Media Delivery Protocol



Highlights - 1

- MoQ delivers Tracks
  - Transform of uncompressed media, specific encoding and encryption process
  - Namespace & authorization scope for media data
  - Owned by a single MoQ Entity under a single provider domain
  - Has single encoding and encryption configuration
  - TrackID = Owning Provider Ref + Track Name (application specific)
- Publish Requests happen on tracks
- Subscribe happens on tracks (addition to group and object references)
- Does not need Relays to understand bundle.

Highlights -2

- Control Streams/Messages
  - Sets up necessary authorization for group of tracks.
  - Manages life cycle for tracks (one or more data streams).
- Data Streams (Unidirectional QUIC Streams or QUIC Datagrams) for carrying media
- Catalog is a special track at the Provider session
  - Catalog TrackID :=<provider-domain>/<moq-session-id>/catalog

Objects Priorities

- Publishers mark objects with sequence numbers within groups and priority values.
- Standard QUIC features with API aligned with HTTP priorities
- **“Droppable”** Flag - Should relays drop or queue under congestion
- **“Priority”** value - Relative priority of object vs other objects in the same track or across tracks in a connection
- Drop Priority through QUIC Stack
  - One object per stream → Stream Priority == Object Priority
  - 1+ objects per stream → 1 Stream per group and priority level (== Stream Priority)

Objects Priorities (2)

Drop Priority through active scheduling (via application enforcement)

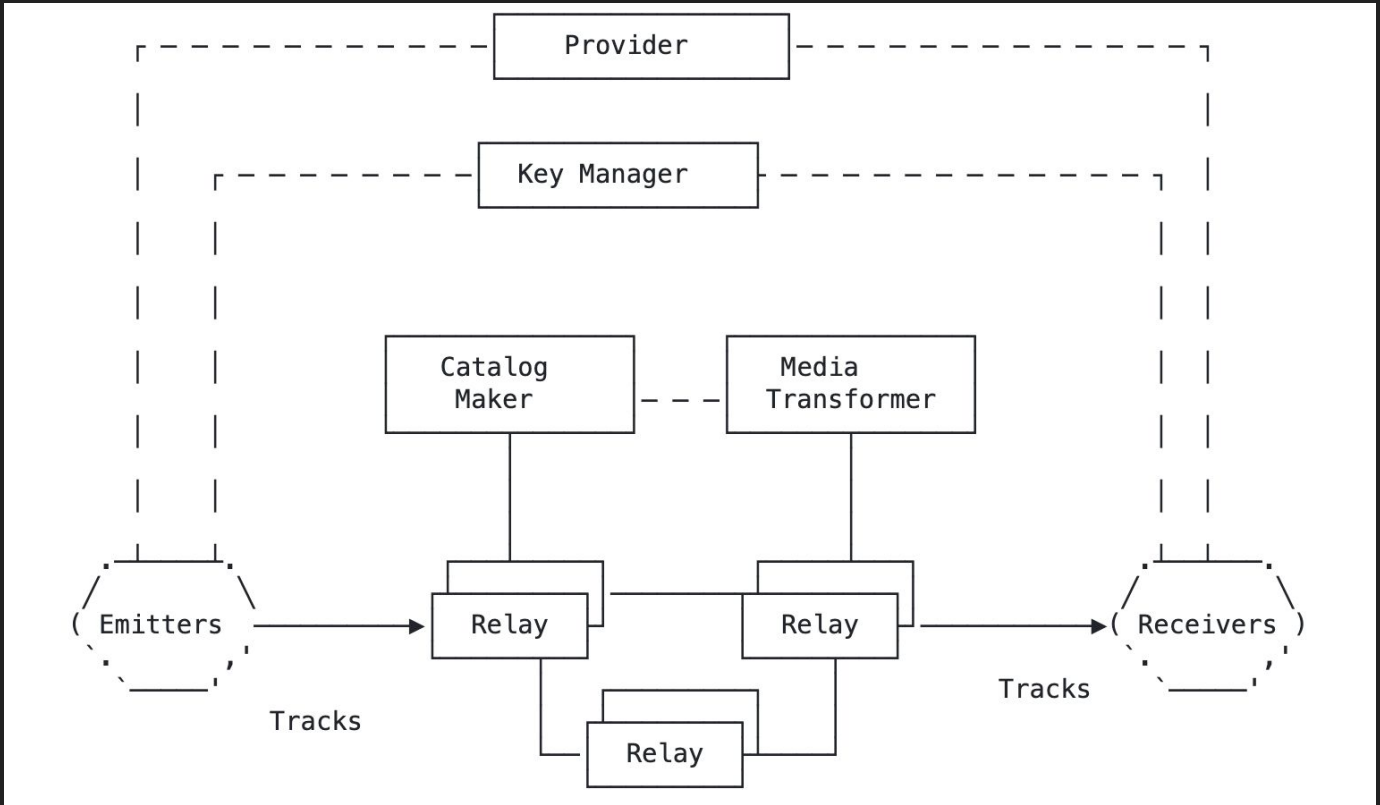
Under Congestion,

- Drop delay/drop higher priority level (until the end of the group)
- Continue to the next higher priority level, until congestion eases
- Carry the drop level through the track

Summary - Asks for warp-05

1. Tracks as independent components
    1.1 Revisit track naming considerations
    1.2 Control Streams/Messages mapped to Subscribe/Publish of tracks
    1.3 Authorization is scoped to tracks (may be derived via other means if enabled by the transport usage, say, WT)
    1.4 Applications can choose to group(bundle) them, if needed.
2. Flexibility to support bundle and non-bundling as prioritization constructs
3. Base transport to not define how a forwarding preference is set, but explain how it can be used
4. Allow Objects be independently named, cached & queried.
5. Support for WT and Native QUIC usage
6. Support for
    6.1 Forwarding Preference needs to work uniformly across all objects inside a connection.
    6.2 Certain encodings of “Forwarding Preference” can include info on track groups.

Backup


Provider
Key Manager
Catalog Maker
Media Transformer
Emitters
Relay
Relay
Receivers
Tracks
Relay
Tracks