# Actors and Functions for MOQ

*Spencer Dawkins*

This presentation describes discussions within the Intermediaries design team, launched after the January 2023 MOQ interim meeting. That's not to say we all agreed on **everything**, of course …

The design team members were Rajeev RK, Spencer Dawkins, Suhas Nandakumar, Varun Singh, Vincent Stone (Shihang), and Will Law.

## Background for this session

- "Like Ted mentioned, we've all got our own preconceived notions of how media is *supposed* to work over the last 30+ years. We need to step back and really explore the different roles, why they exist, and what data they need to function. We'll be able to justify any messages and their contents once we can map out how data needs to flow." - *Luke Curley, 2 February 2023*

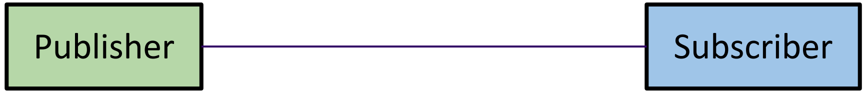
- Because we work on similar but not identical media use cases, we use ideas that have similar but not identical meanings across use cases.
- The "Intermediaries" design team has been working on figuring out what a group of six people can agree on.

## MoQ Roles - something an entity is TRUSTED with

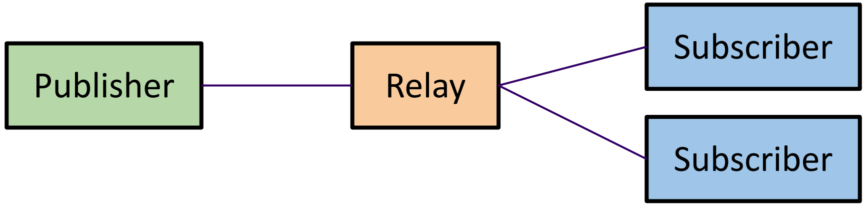
- Publishers
  - Provide Publish Requests
  - Provide Catalog Messages
  - Provide Object Messages
- Subscribers
  - Provide Subscribe Requests
  - Consume Catalog Message
  - Consume Object messages
- Media Source
  - Generates Media Content, encodes, encrypts
- Media Sink
  - Decrypts, decodes, and consumes Media Content

# (Minimal) Distribution Topologies

*Minimal Topology*

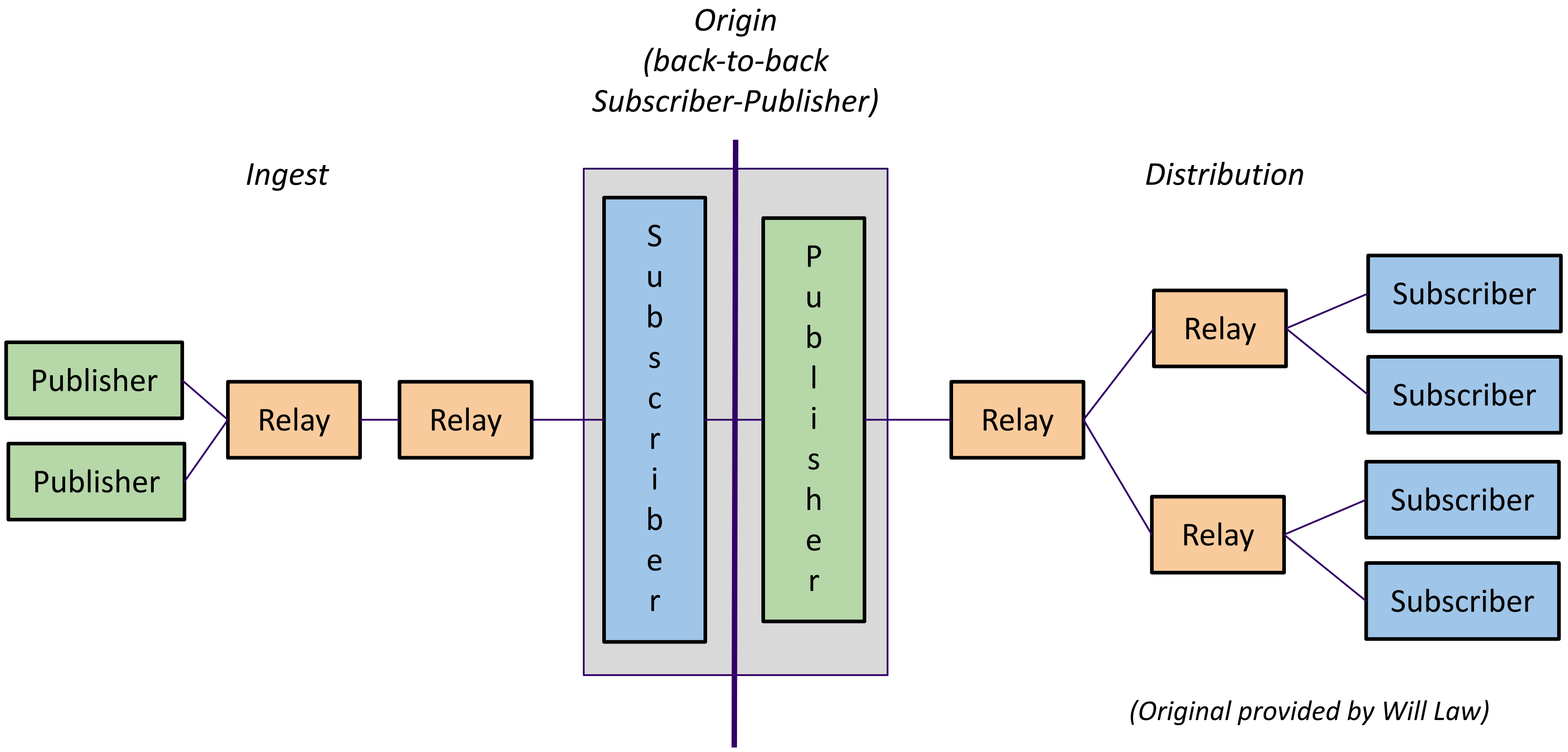


*Introducing a Relay for fan-out*



*(Original provided by Will Law)*

# Ingest and Distribution Topologies



*(Original provided by Will Law)*

# MOQ Roles and Entities for various use cases

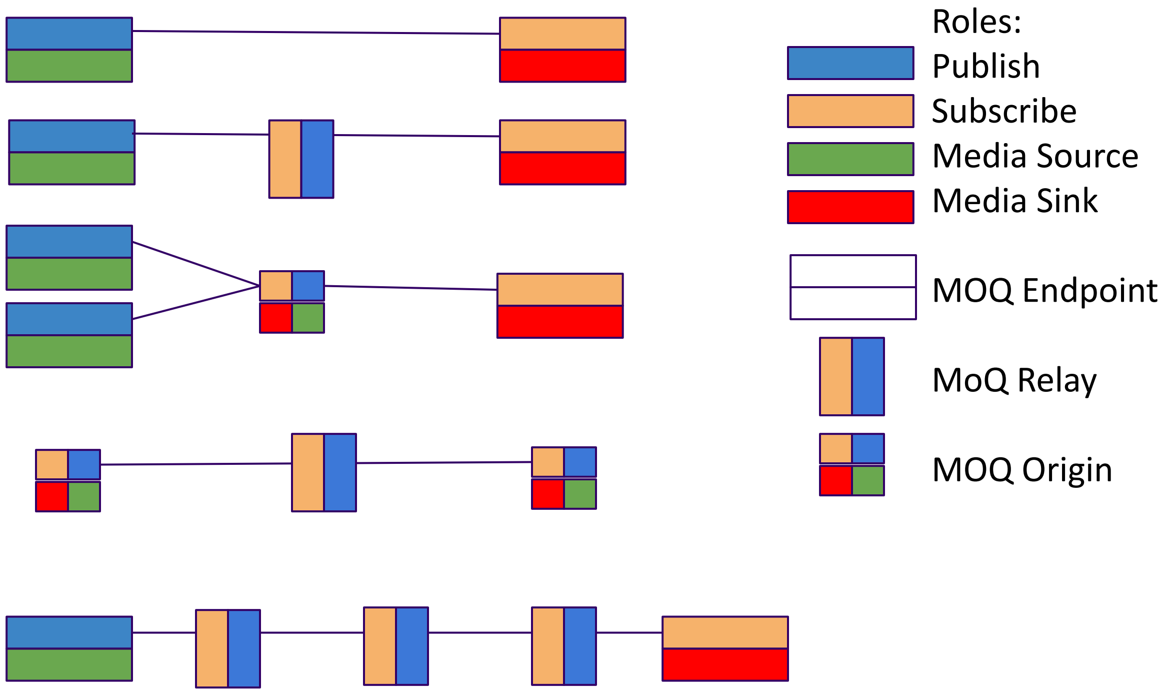
OBS publishing media directly to a client

OBS publishing media via relay to a client

Two endpoints publishing media to a service that mixes them together

Two endpoints having a web conference through a relay

Streaming the sports game through a CDN to a client



## MoQ Endpoints

- Roles
  - Media source
  - Media sink
  - Publisher
  - Subscriber
- "What am I trusted with?"
  - Keys to encrypt/decrypt the raw content, if provided
  - To modify and/or originate object headers.
  - To modify and/or originate object payloads
  - Parsing and manipulating catalog messages
  - Parsing and manipulating object messages
  - Originating subscribe messages
  - Originating publish messages

# Examples of MoQ Endpoints

- OBS publishing a stream
- Web conference client
- Middle box transmuxer
- Middle box transcoder
- Middle box compositor

## MoQ Relays

- Roles
  - Publisher
  - Subscriber
- What am I trusted with ?
  - To modify object header
  - To forward subscriptions to other entities
  - To forward the publishes to matched subscribers
  - To store the published content
  - To validate authorization credentials for publish/subscribe requests

## MoQ Relays

- What am I **NOT** trusted with?
    - Parsing catalog messages
    - Parsing object message payload
    - Originating subscribe messages
    - Originating publish messages

## What MoQ Relays Do

- ***"Anything that <u>does not require</u> access to unencrypted media payloads"***

- Replication: Subscribe to one broadcast and replicate the same broadcast
- Fanout: replicate one broadcast to multiple subscribers

- Hop-by-hop reliability: up to some level of "reliability" ("configurable latency")
- Caching functionality: for reliability, for rewind/replay …

- Rate adaptation: if this can be performed using media metadata