



# CAT-4-MOQT

## Authentication and Access Control for MOQT

Will Law, Akamai

<https://datatracker.ietf.org/doc/draft-law-moq-cat4moqt/>

IETF #123, Madrid, July 21-25 2025

## Merged PR#9: First round of edits to fill out the definition of the claim

The "moqt" claim limits the MOQT actions for which the token can provide Access.

It is an array of **action limits**. Each limit is an array with three elements:

- an integer or array of integers that identifies the action(s).
- a match object for the namespace,
- a match object for the track name.

```
moqt-label = TBD_MOQT
moqt-value = [ + moqt-limit ]
moqt-limit = [ moqt-actions,
moqt-ns-match, moqt-track-match ]
moqt-actions = moqt-action | [ +
moqt-action ]
moqt-action = int
moqt-ns-match = match
moqt-track-match = match
```

```
Prebuilt match types in CAT [ exact-match = 0
prefix-match = 1
suffix-match = 2
contains-match = 3
```

# Merged PR: #15 - Updating action table and examples

This aligns cat-4-moqt with the latest updates to MOQT in terms of which messages carry AUTHORIZATION tokens.

Action	Key	Reference
CLIENT_SETUP	0	[MoQTransport] Section 8.3
SERVER_SETUP	1	[MoQTransport] Section 8.3
ANNOUNCE	2	[MoQTransport] Section 8.23
SUBSCRIBE_NAMESPACE	3	[MoQTransport] Section 8.28
SUBSCRIBE	4	[MoQTransport] Section 8.7
SUBSCRIBE_UPDATE	5	[MoQTransport] Section 8.10
PUBLISH	6	[MoQTransport] Section 8.13
FETCH	7	[MoQTransport] Section 8.16
TRACK_STATUS	8	[MoQTransport] Section 8.20

Table 1

Any action can be independently protected by any combination of these claims

## Action types

These are the MOQT actions we can protect:

0 - CLIENT\_SETUP

1 - SERVER\_SETUP

2 - ANNOUNCE

3 - SUBSCRIBE\_NAMESPACE

4 - SUBSCRIBE

5 - SUBSCRIBE\_UPDATE

6 - PUBLISH

7 - FETCH

8 - TRACK\_STATUS

- Expiration time, notBefore
- IP: address, CIDR, range, method, ALPN
- AS number
- Geo location
- URL (scheme, host, port, path, query, parent,filename,stem,extension)
- Renewable
- Issuer
- Audience
- Token ID
- Replay
- DPOP
- Probability of Rejection
- Version
- TLS public key
- Issued At

# Example #1: Allow with a prefix match "example.com/bob"

```
{  
  /moqt/ TBD_MOQT: [[  
    [ /ANNOUNCE/ 2,  
      /SUBSCRIBE_NAMESPACE/ 3,  
      /PUBLISH/ 6,  
      /FETCH/ 7 ],  
    { /exact/ 0: "example.com"},  
    { /prefix/ 1: "bob"}  
  ]]  
}
```

## Example #2: evaluating multiple actions in the same token.

Here the client can publish to an track beginning with "bob" but can only publish logs exactly to "logs/12345/bob", both before the same expiry date.

```
{  
  
  /moqt/ TBD_MOQT: [  
    [ /PUBLISH/ 6, { /exact/ 0: "example.com"}, { /prefix/ 1: "bob"} ],  
    [ /PUBLISH/ 6, { /exact/ 0: "example.com"}, { /exact/ 0: "logs/12345/bob"} ]  
  ],  
  
  /exp/ 4: 1750000000  
  
}
```

Evaluating "example.com/bob/123" would succeed on test 1 and test 2 would never be evaluated.

Evaluating "example.com/logs/12345/bob" would fail on test 1 but then succeed on test 2.

Evaluating "example.com" would fail on test 1 and on test 2.

# Composite claims

<https://datatracker.ietf.org/doc/draft-lemmons-composite-claims/>

If there are other claims that depend on which MOQT limit applies, a logical claim is required. Here the client can publish to an track name beginning with "bob" but can only publish logs exactly to "logs/12345/bob", both with different expiry dates.

{

```
/or/ TBD_OR: [
```

```
{
```

```
  /moqt/ TBD_MOQT: [[/PUBLISH/ 6, { /exact/ 0: "example.com"}, { /prefix/ 1: "bob"}]],
```

```
  /exp/ 4: 1750005555
```

```
},
```

```
{
```

```
  /moqt/ TBD_MOQT: [[/PUBLISH/ 6, { /exact/ 0: "example.com"}, { /exact/ 0: "logs/12345/bob"}]],
```

```
  /exp/ 4: 1750009000
```

```
}
```

## Issue #7: When are claims evaluated in long duration sessions?

CAT claims are written assuming a request, which occurs at a single point in time. However MOQT has actions which can persist over the life of the session, which may last a long time.

PR#9 introduced the new **moqt-reval claim**

```
$$Claims-Set-Claims // = (moqt-reval-label =>
moqt-reval-value)
moqt-reval-label = TBD_MOQT_REVAL
moqt-reval-value = number
```

- Indicates the interval in seconds with which the token must be revalidated for ongoing streams.
- If the token is no longer acceptable, the actions authorized by it **MUST NOT** be permitted to continue.
- The default value is 0, which means the token **MUST NOT** be revalidated.

# Open Issue #16: What type of matches do we really need?

The current claim definition defines

```
bin-match = {  
  ? exact-match ^ => bstr,  
  ? prefix-match ^ => bstr,  
  ? suffix-match ^ => bstr,  
  ? contains-match ^ => bstr,  
}
```

Question 1: Do we need all of these match types, or would EXACT and PREFIX be sufficient?

Question 2: Are prefix matches forced to be complete tuple field boundaries? For example, assuming tuple boundaries are represented by a "/", is `cust` a prefix match for `customer/application/instance`

# Open Issue #18: Definition of catif for MoQ actions

Today's catif sections in CAT was written assuming an HTTP response

*The type of this claim is a map; the keys of the map are either claim keys or arrays of claim keys and the values are arrays; the arrays have up to three elements: **the status code to return**, a map with headers and values (the header map), and a TextString that identifies a key to sign the resultant URI with (the key id)*

A moqt-reval-label can typically be used as the key, but the response would ideally be a **MoQ message**.

For example, if an ongoing Subscription is stopped, the Publisher should send a SUBSCRIBE\_ERROR with code 0x1, while an ongoing fetch would require a FETCH\_ERROR.

We need to define the appropriate MOQT message to be sent when the moqt revalidation fails for each type of MOQT Action.

# Call for action

We **request adoption** by the moq workgroup.

While we realize that moq workgroup should not be the home for anything-over-moqt, a common and generically applicable authentication scheme seems like a core requirement for moqt interop.

Thank you for your time.