MQTT-TLS Profile of ACE

draft-sengul-ace-mqtt-tls-profile-00

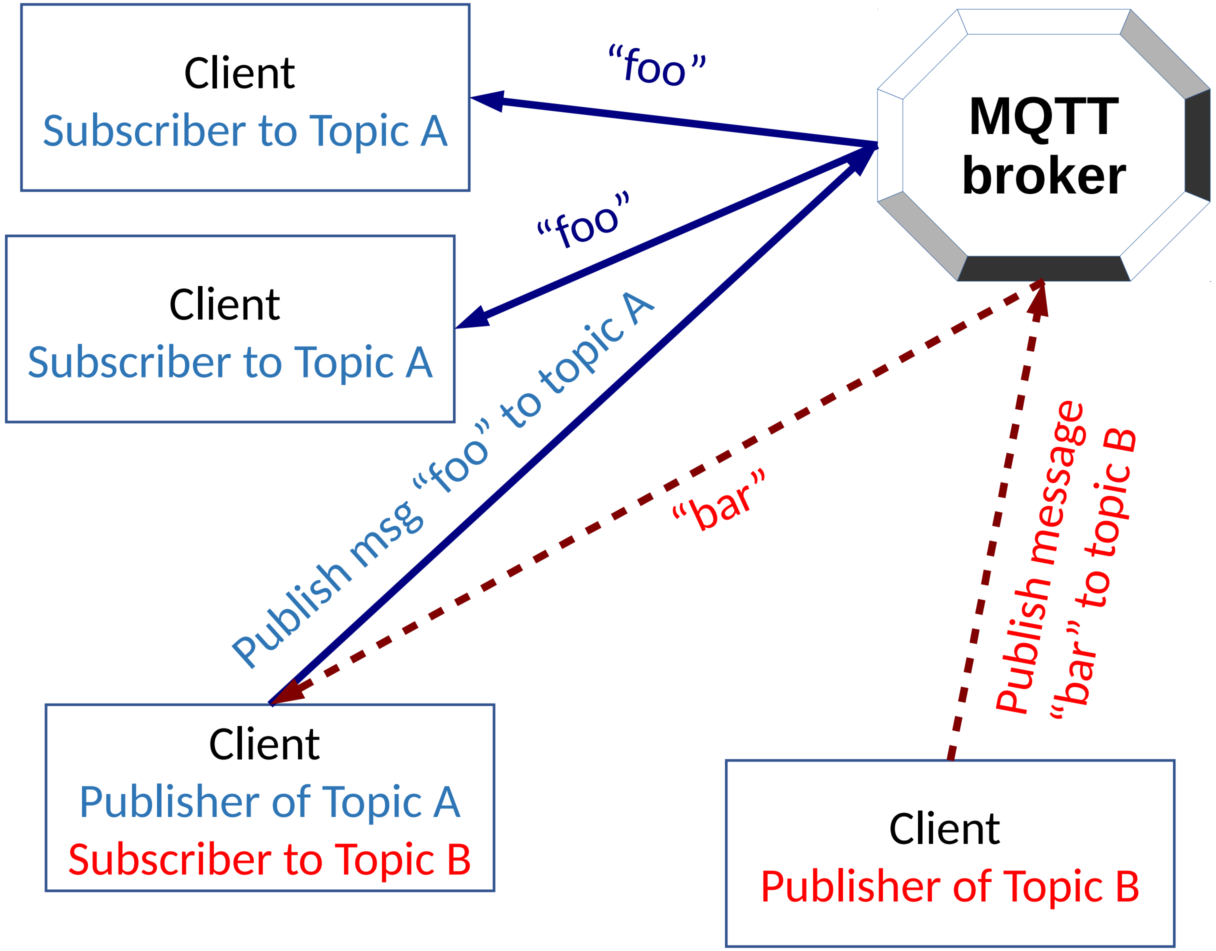
Cigdem Sengul and **Anthony Kirby**

(Cigdem.Sengul@nominet.uk & Anthony.Kirby@nominet.uk)

IETF 99 ACE WG meeting

17th July 2017

Background: MQTT



MQTT in a nutshell:

- runs over TCP & supports TLS
- “pub/sub” messaging
- subject-based filtering by *topic*, e.g. “IETF/99/ACE/MQTT”
- clients first *CONNECT* to the broker
- clients can *SUBSCRIBE* to *topics*
- clients can *PUBLISH*, and messages are forwarded to subscribers

ACE options for MQTT

MQTT Security:

- TLS
- username + password, in *CONNECT* message

Goal of this draft:

- How do we support tokens, ACE style, in MQTT?

We need a profile!

ACE options for MQTT

MQTT Security:

- TLS
- username + password, in *CONNECT* message

Goal of this draft:

- How do we support tokens, ACE style, in MQTT?

We need a profile!

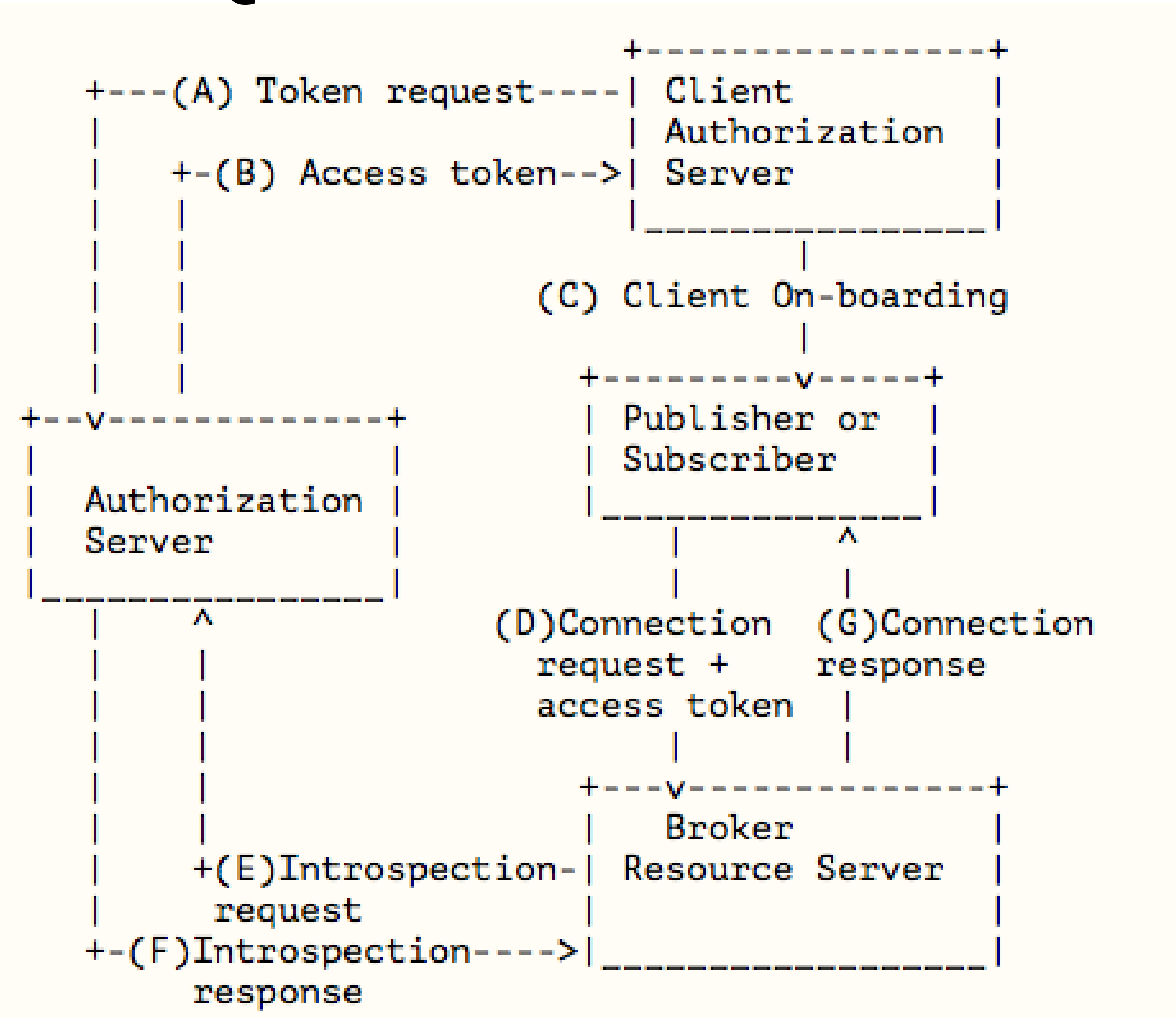
- Options for MQTT-ACE profile(s)
  1. **MQTT over TLS**
  2. MQTT with Application Layer Security + ACE
  3. MQTT-SN over UDP with DTLS?

- This draft is an ACE profile for MQTT over TLS (follows MQTT v3.1.1 – OASIS standard)
- Draft content:
  - Authorizing connection establishment
  - Authorizing *PUBLISH* messages
  - Authorizing *SUBSCRIBE* messages

Profile summary

Profile identifier	mqtt_tls
Communication protocol	MQTT
Security protocol	TLS
AS discovery	Not supported
Client & RS mutual authentication	Client authenticates RS using certificate in TLS handshake RS authenticates client using token + MAC in MQTT *CONNECT* message
PoP protocols	Symmetric/asymmetric
Token transport	MQTT *CONNECT* message (alternatives in Appendix)
Token introspection	/introspect (HTTPS)
Token request	/token (HTTPS)
/authz-info	May be supported (See Appendix)

MQTT-ACE actors



```
                            +----------------+
   +---(A) Token request----| Client         |
   |                        | Authorization  |
   |   +-(B) Access token-->| Server         |
   |   |                    |________________|
   |   |                            |
   |   |                (C) Client On-boarding
   |   |                            |
   |   |                  +---------v-----+
+--v-------------+        | Publisher or  |
|                |        | Subscriber    |
|  Authorization |        |_______________|
|  Server        |            |       ^
|________________|            |       |
   |    ^             (D)Connection  (G)Connection
   |    |               request +    response
   |    |               access token  |
   |    |                     |       |
   |    |                 +---v--------------+
   |    |                 |   Broker         |
   |    +(E)Introspection-| Resource Server  |
   |     request          |                  |
   +-(F)Introspection---->|__________________|
        response
```

- Resource Server ≡ MQTT broker
- Protected Resources ≡ MQTT Topic
- Publisher & subscribers are treated similarly (real clients do both)

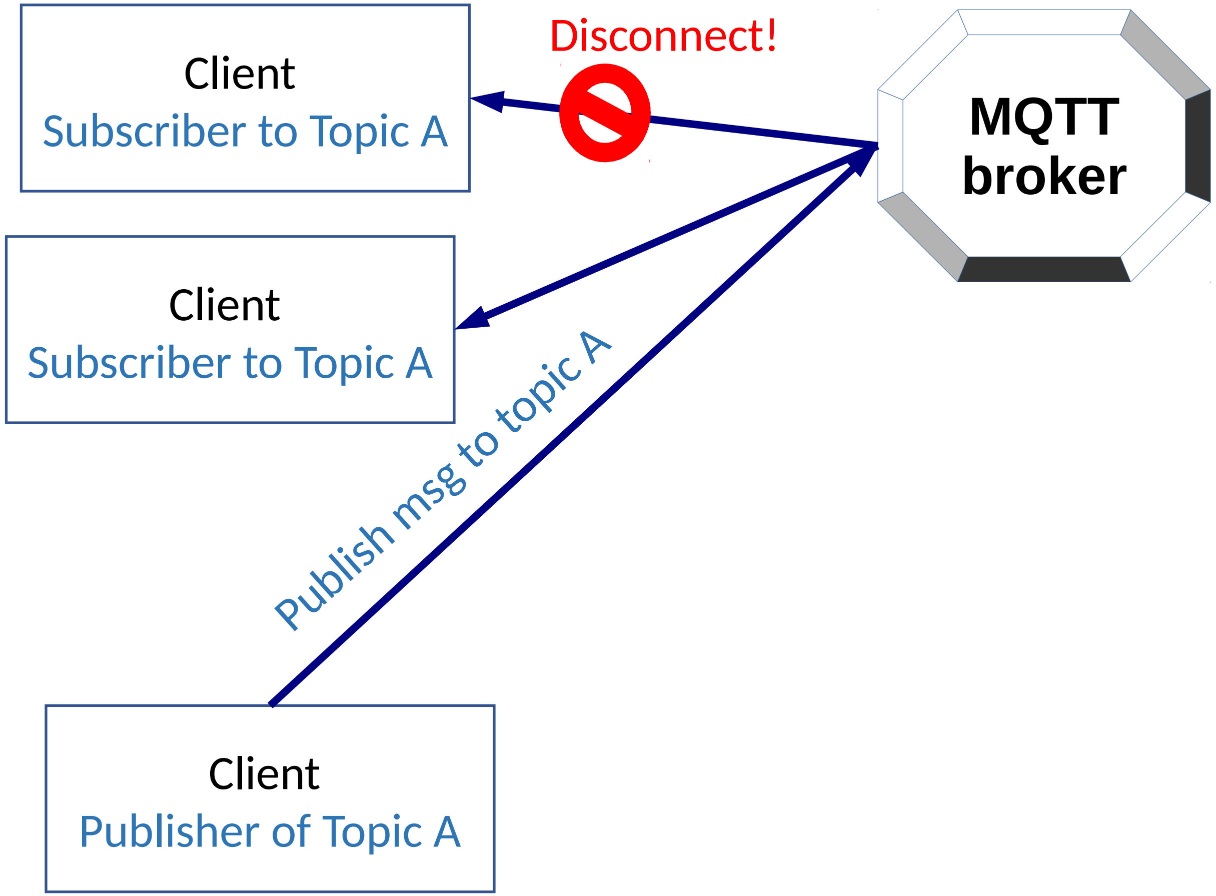
Connection establishment

- Create TCP + TLS connection
- Send MQTT *CONNECT* message:
  - username = “ace”
  - password = <JSON string containing PoP token + MAC>
- Broker responds with *CONNACK* + status
- Broker caches token for duration of session

Subscribing to protected topics

- A subscriber may *SUBSCRIBE* to multiple topics
- Permission to *SUBSCRIBE* to each topic is contained in client’s Token (transported during *CONNECT*)
- For each topic requested:
  - The broker checks the token, returns a *SUBACK* Code (just success or failure)
- Topic filters may include wildcards
  - In this case, the broker need to check that the scopes in the token cover all possible topics under the wildcard

Publishing to protected topics



Broker algorithm:

```
check(publisher token)
if valid:
    for each subscriber:
        check_expiry_and_scope
                  (subscriber token)
        if valid:
            forward message to client
        else:
            disconnect(subscriber)
else:
    disconnect(publisher)
```

Next Steps:

- Publish our prototype (based on plugin for mosquitto)
- Revise this draft:
  - Integrate feedback (thank you!)
  - Simplify with basic functionality for MQTT 3.1.1
  - Support richer functionality in MQTT 5.0 (Auth Method, Auth Data etc)
- Get more feedback

Thank you!

Questions?