OCF Jakarta+ – MQTT proxy and Cloud mapping – Core Technology WG CR 3477

Legal Disclaimer

****** new section <number>: MQTT proxy ******

7.6.3.9.1.1 OCF MQTT Proxy

An MQTT proxy is an OCF Device that acts as a proxy between MQTT and CoAP transports. An MQTT proxy enables the ability for OCF clients in the MQTT domain to talk to OCF Devices in the CoAP domain.

An MQTT proxy contains the following functionality:

– An OCF Client, talking to OCF Servers on the local network
– An MQTT Client that creates OCF Servers in the MQTT domain, e.g. it represents the proxied devices in MQTT domain.

This is depicted in figure [A0].

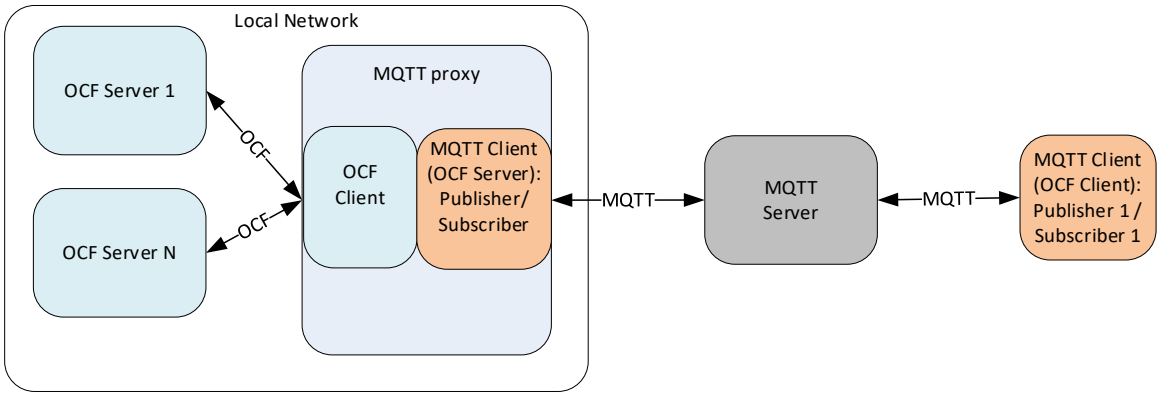


Figure 1 [A0] OCF CoAP domain and OCF MQTT domain interconnected by Proxy

The OCF Servers to be proxied will be provisioned by an OCF (mediator) Client on the MQTT Proxy

7.6.3.10 Resources for MQTT proxy

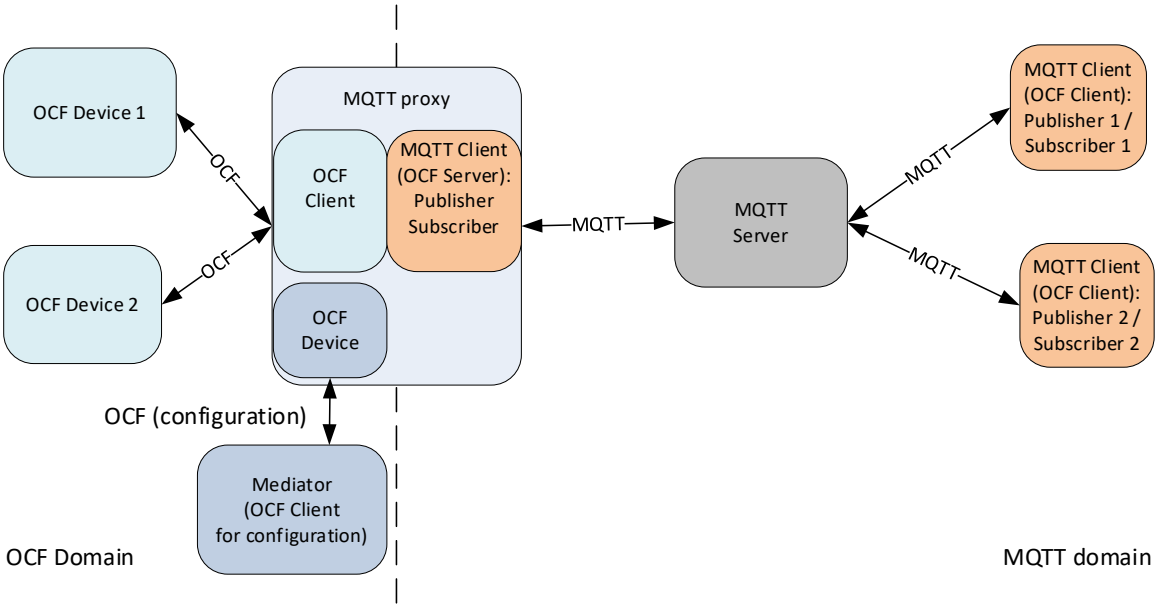


Figure 2 [A] OCF extended with OCF Clients in the MQTT domain

To represent the OCF CoAP Servers on the MQTT network, the following Resources are exposed by the OCF Device “oic.d.mqttproxy” on the CoAP side of the MQTT proxy:

– "oic.r.mqtt.conf" Resource Type

– "oic.r.d2dserverlist" Resource Type

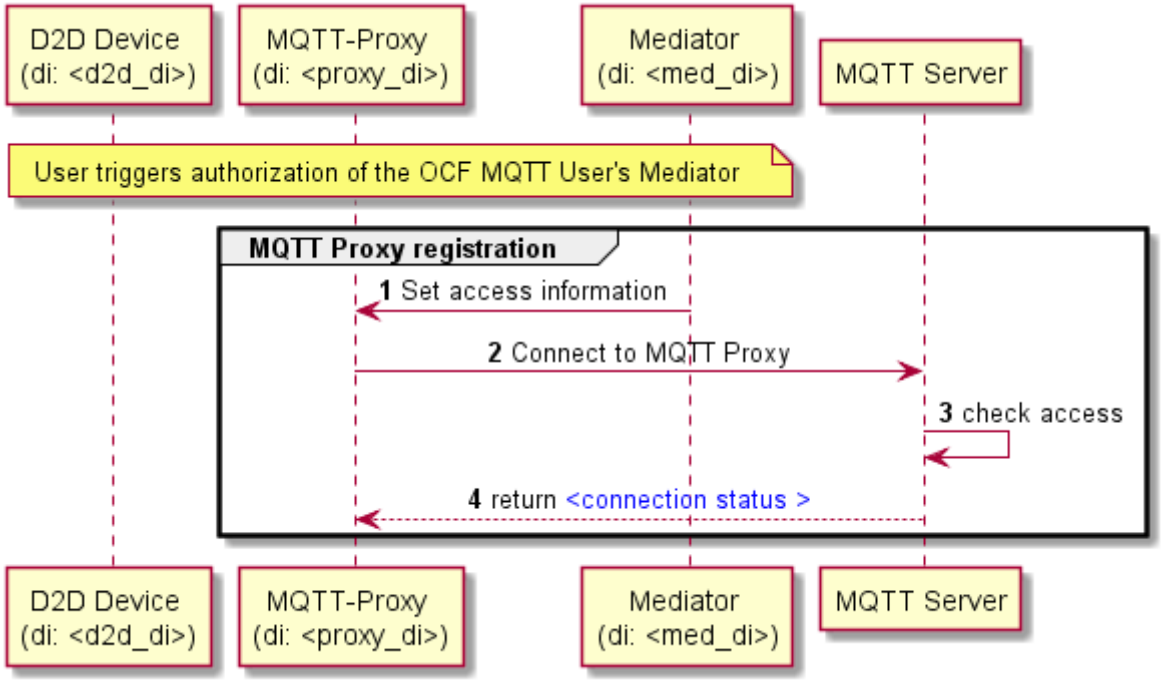
The "oic.r.mqtt.conf" Resource Type is used to configure the MQTT client to connect to an MQTT server. The Resource Type is available on the CoAP side so that proximal interaction is possible to configure the MQTT proxy to connect to an MQTT server. The information supplied to the MQTT client also includes security information. The connection to the MQTT server should be secured by TLS. The information to be supplied to the Mediator to configure the "oic.r.mqtt.conf" Resource is provided out of band; this information determines how the MQTT proxy is being used in a larger setup. The "oic.r.d2dserverlist" Resource Type is used to list OCF Devices that will be proxied from the CoAP domain to the MQTT domain. This list is maintained from the CoAP OCF domain.

7.6.3.10.1 Connecting to a MQTT Server

The information of the MQTT client to connect to the MQTT server may be conveyed by an OCF Resource. This means that the MQTT proxy may be headless and may be configured with a Mediator (OCF client). The configuration information consists of data to contact the MQTT server and also of data to secure the connection.

```
@startuml

' participant "AMS \nService" as ams
participant "D2D Device \n(di: <d2d_di>)" as d2d
participant "MQTT-Proxy \n(di: <proxy_di>)" as cp
participant "Mediator \n(di: <med_di>)" as med
' participant "OBT" as obt
participant "MQTT Server" as cloud

' global configuration
autonumber

skinparam {
  ' DefaultFontSize 50
```

```
  SequenceMessageAlignment center
  ' SequenceTitleFontSize 40
}


note over d2d, med
  User triggers authorization of the OCF MQTT User's Mediator
end note


group MQTT Proxy registration
  med -> cp: Set access information

  cp -> cloud: Connect to MQTT Proxy
  cloud -> cloud: check access
  cp <-- cloud: return <color blue><connection status ></color>

end group

'group Resource publication
'  cp -> cloud: Publish D2DServerList Resource
'end group

@enduml
```



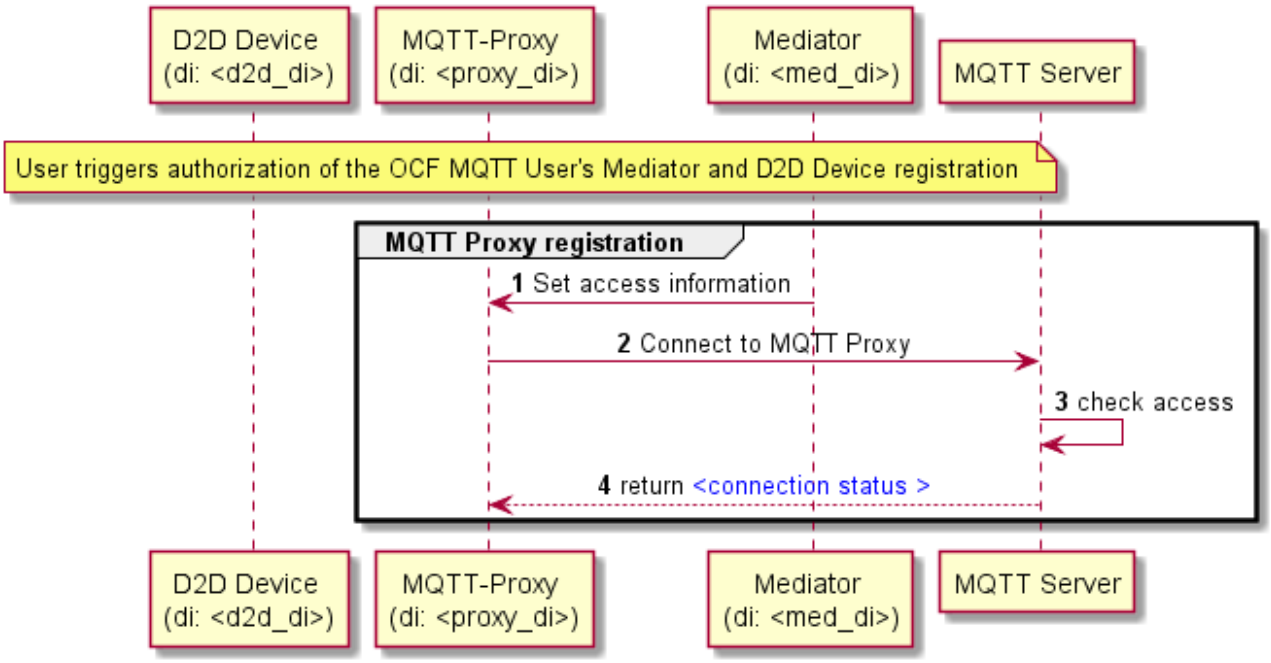


Figure 3 - Registration of the MQTT proxy (as MQTT client) with an MQTT Server

Table 1 – Properties of "oic.r.mqtt.conf" Resource

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory to implement	Description
Server address	"server"	string	N/A	N/A	RW	Yes	The connection information of the MQTT server may be an ip addres or a URI e.g: "192.168.178.89" "test.mosquitto.org"
connection	"port"	integer	N/A	N/A	RW	Yes	The port number of the MQTT server e.g. 1883 for unsecured ports or 8883 for secured ports
Keep alive interval	"kai"	integer	N/A	seconds	RW	Yes	The keep alive interval for the MQTT client sServer connection.
User identifier	"uid"	string	N/A		RW	Yes	User ID, if supported by the MQTT server
password	"pwd"	string	N/A		RW	Yes	Password or token belonging to the user ID.
certificate authority file of the MQTT server	"cacert"	string	As byte array		R	Yes	The credential, if supported by the MQTT server
Client certificate to authenticate	"clcert"	string	As byte array		R	Yes	The credential, if supported by the MQTT server

the connection							
log	“log”	string	NA		R	Yes	Logging of the connection status
MQTT connection reason codes	“crcode”	integer	NA		R	Yes	See MQTT table 3-1 Note that before connecting, the value should be initialized on -1 indicating, “not yet connected”

All Properties listed in table XXX are required to be implemented, e.g., listed in the IDD as optional. Not all Properties have to be on the wire though, the Property usage depends on the used MQTT server.

7.6.3.10.2 Proxying an OCF Device

The OCF Devices to be proxied are listed in the d2dserver list Resource. Which Vertical Resources are proxied per OCF Device is implementation dependent.

```
@startuml


participant "Mediator \n(di: <med_di>)" as med
participant "D2D Device \n(di: <d2d_di>)" as d2d
participant "MQTT Proxy \n(di: <proxy_di>)" as cp
' participant "OBT" as obt
participant "MQTT server" as cloud
' participant "Authorization Server" as auth
' participant "OCF Cloud Client" as cc

' global configuration
autonumber

skinparam {
  ' DefaultFontSize 50
  SequenceMessageAlignment center
  ' SequenceTitleFontSize 40
}


'
' Onboard new D2D Device
'
d2d <- med: Mediator provisions access control entries \nto the D2D Device and peer
Devices


'
' Register D2D Device with MQTT server
'
group D2D Device registration (OCF Domain)

cp <- med: UPDATE /d2dserverlistURI&di=<d2d_di>
cp -> cp: Adds received "di" to "oic.r.d2dserverlist:dis"
' cp --> med: UPDATE Response

d2d <- cp: RETRIEVE /oic/res
```

```
d2d --> cp: RETRIEVE Response

end group

'
' == Publish Device ==
'
Group Device Publication (MQTT domain)
  cp -> cloud: Publish Resources as MQTT topics
end group

@enduml
```

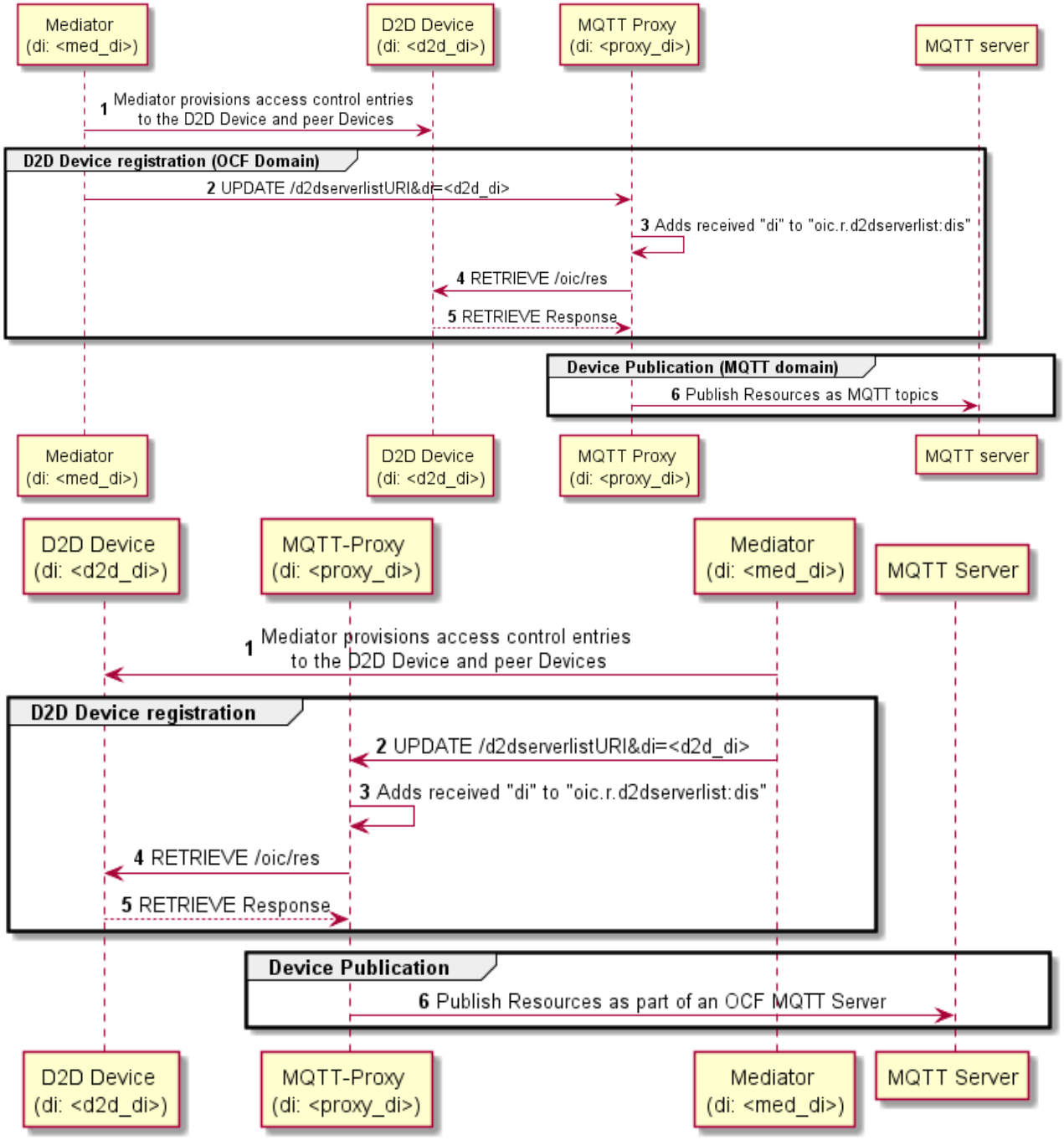


Figure 4 – Device publication to an MQTT server

7.6.3.10.3 Security considerations

The OCF Client in the MQTT proxy needs to be granted access to the D2D Device. Even if a privileged OCF Client adds a D2D Device to the "oic.r.d2dserverlist" Resource, the access may still be denied by the D2D Device. This is because the OCF Client in the MQTT proxy needs to be given access to any of the D2D Devices in its list and having the correct access levels set up for the OCF Client in the MQTT proxy.

The connection from a MQTT client to the MQTT server needs to be secure, e.g., using a TLS connection. However, MQTT specifies multiple mechanisms to create a secure connection from an MQTT client to an MQTT server. The used MQTT server should only connect to other MQTT clients via a secure connection.

See also Security considerations [MQTT security considerations].