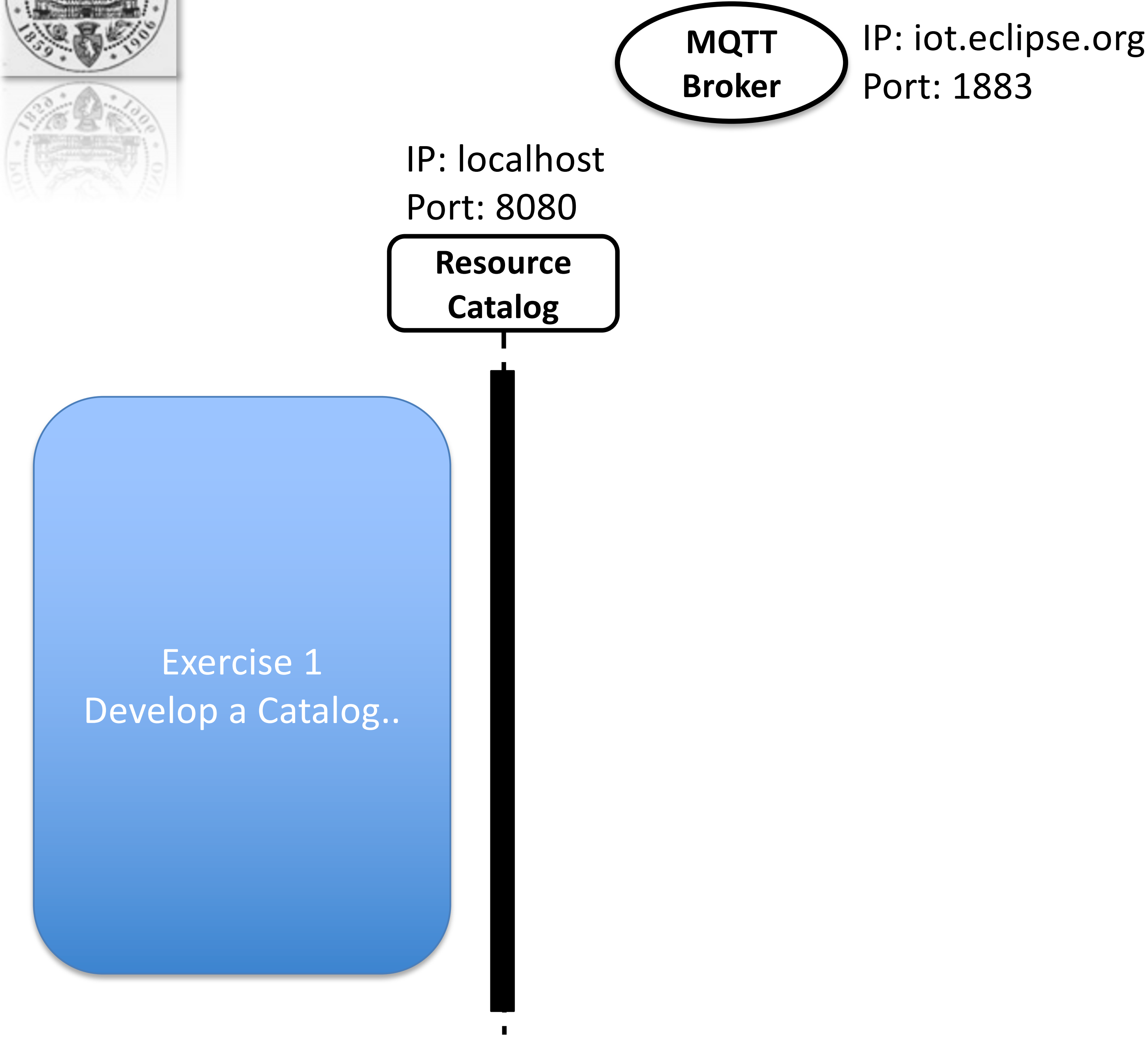
Tecnologie per IoT

Edoardo Patti

Luca Barbierato

Laboratorio SW2

PLATFORMS FOR INTERNET OF THINGS APPLICATIONS



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost
Port: 8080

Resource Catalog

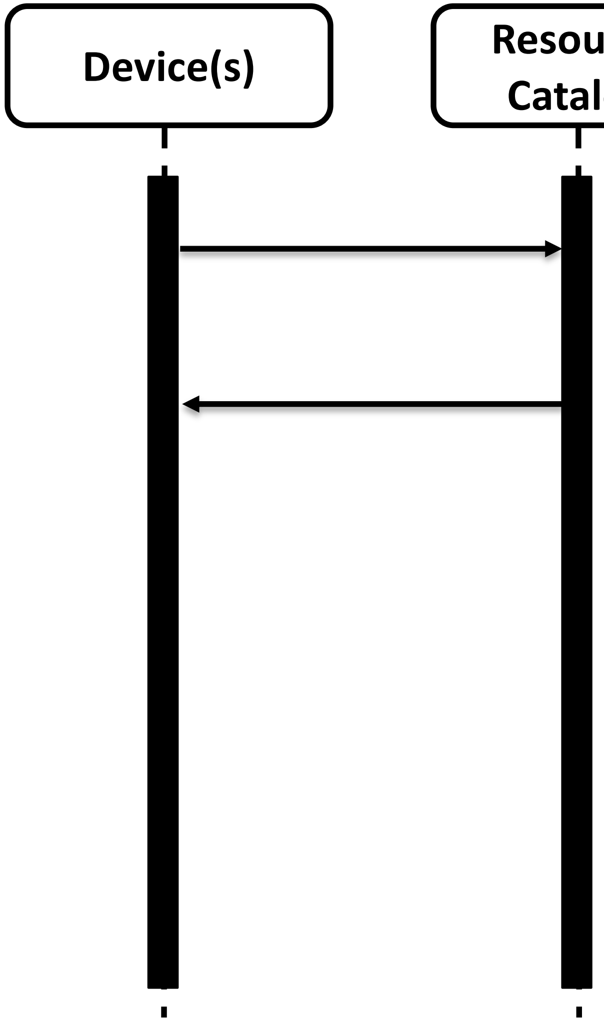
Exercise 1
Develop a Catalog..

MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost

IP: localhost
Port: 8080



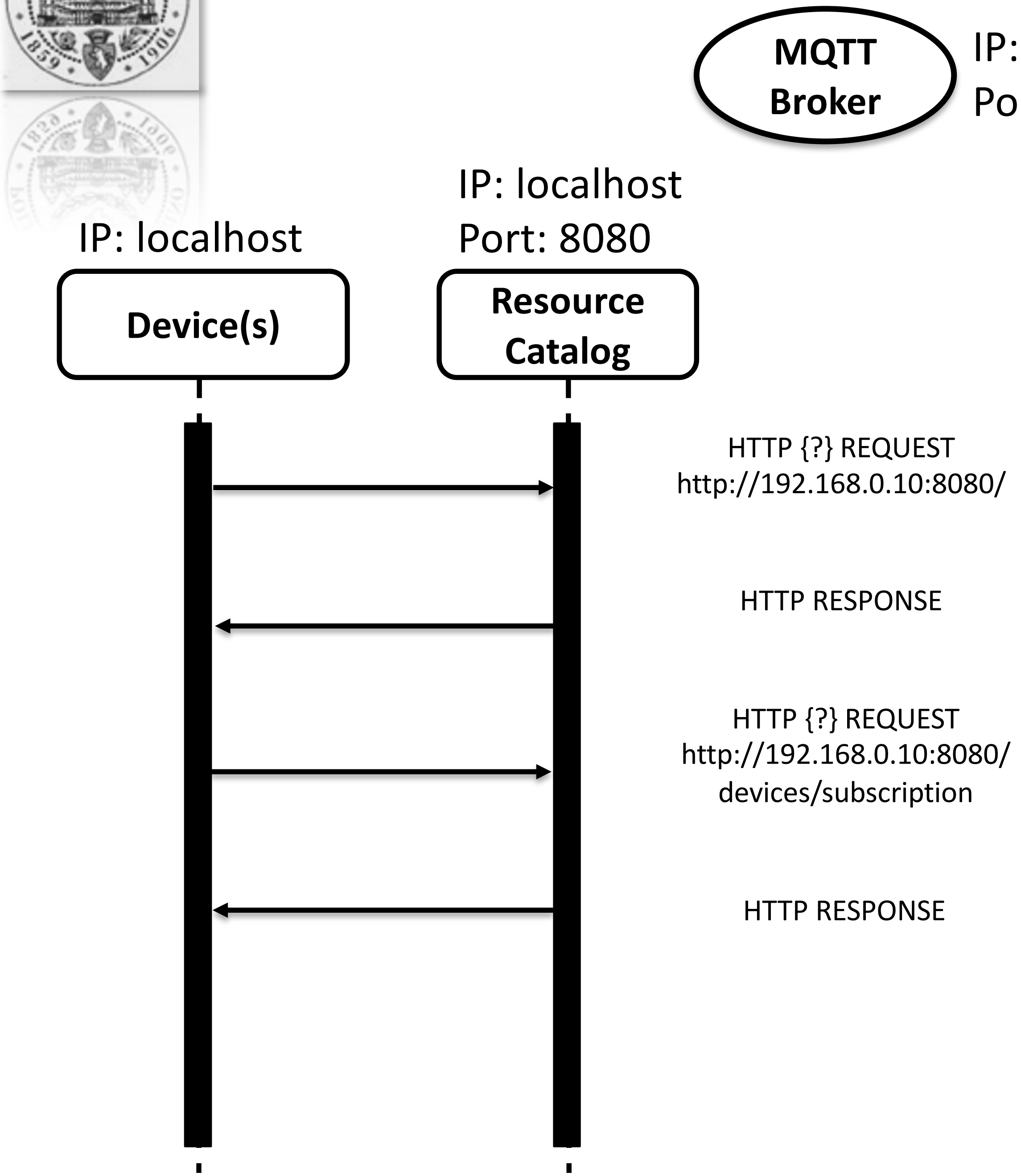
HTTP {GET?POST?PUT?} REQUEST
http://192.168.0.10:8080/

HTTP RESPONSE

```
{
        “subcriptions”:{
                “REST”:{
                        “device”: “http://192.168.0.10:8080/devices/subscription”,
                        “service”: “http://192.168.0.10:8080/services/subscription”,
                        “user”: http://192.168.0.10:8080/users/subscription
                }
                “MQTT”: {
                        “device”:{
                                “hostname”: “iot.eclipse.org”,
                                “port”:  “1883”
                                “topic”: “tiot/{group}/catalog/devices/subscription”
                        }
                }
        }
}
```

Device(s) will:

- Retrieve information regarding subscription containing:
    - REST APIs for subscribing a device, service, or a user
    - MQTT info for subscribing a device (ip, port, topic)



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost

Device(s)

IP: localhost
Port: 8080

Resource Catalog

HTTP {?} REQUEST
http://192.168.0.10:8080/

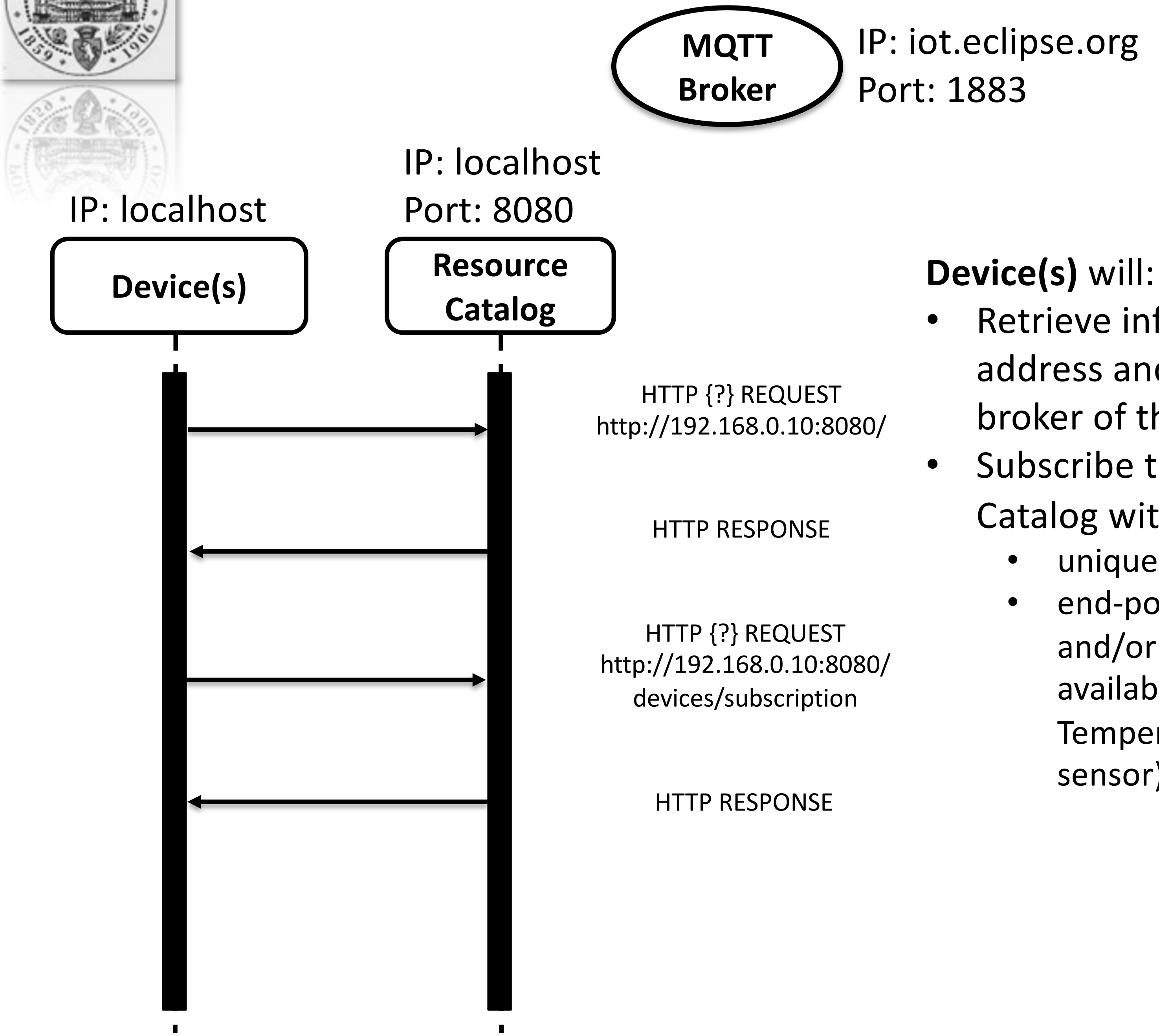
HTTP RESPONSE

HTTP {?} REQUEST
http://192.168.0.10:8080/devices/subscription

HTTP RESPONSE

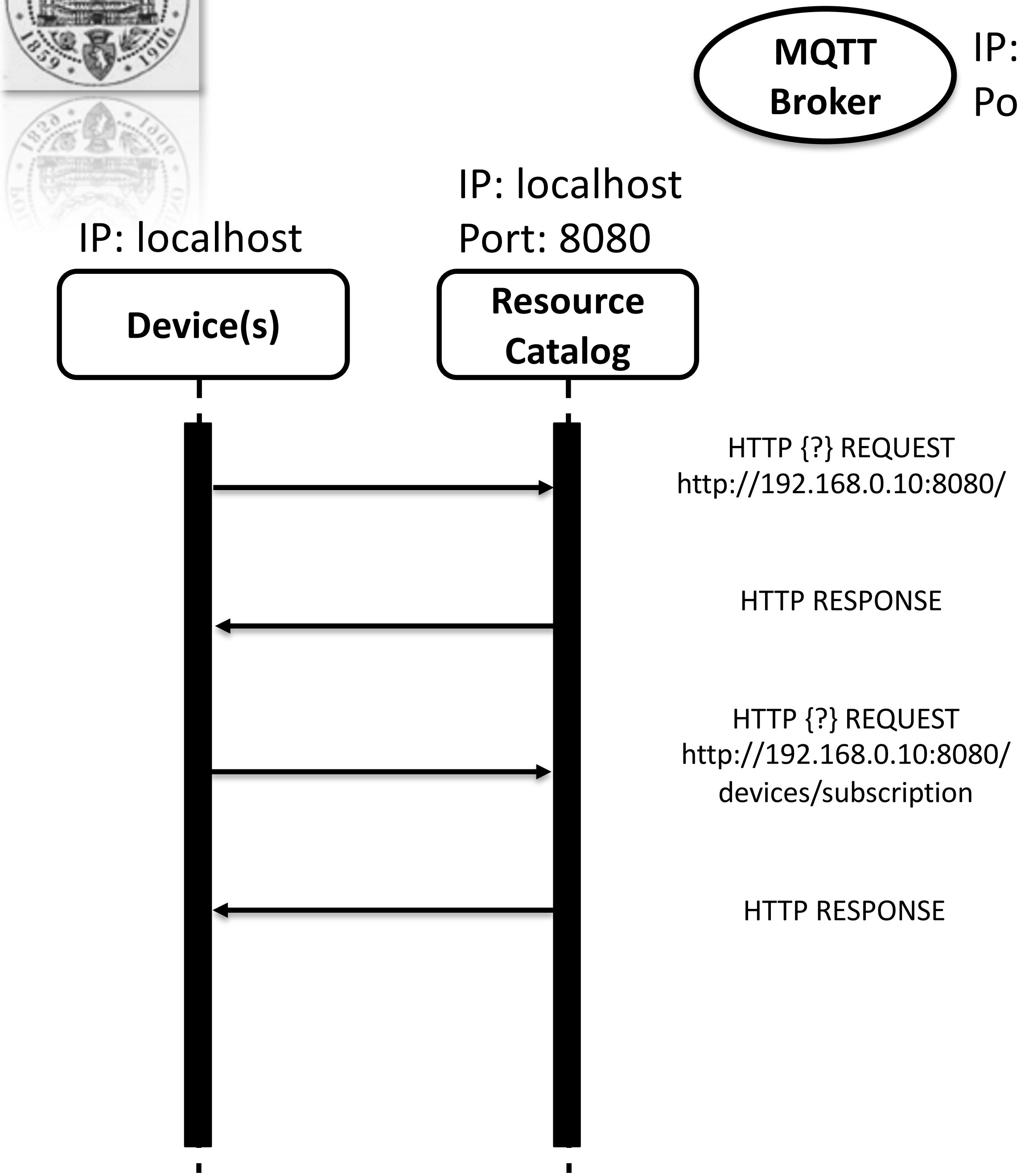
Device(s) will:

- Retrieve information regarding the IP address and port and/or MQTT broker of the available service
- Subscribe the first time to the Service Catalog with the followings info:
  - unique deviceID (**uuid library**)



Device(s) will:

- Retrieve information regarding the IP address and port and/or MQTT broker of the available service
- Subscribe the first time to the Service Catalog with the followings info:
  - unique deviceID (**uuid library**)
  - end-points (i.e. Rest Web Services and/or MQTT topics)
    available resources (e.g. Temperature, Humidity and Motion sensor)



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost

Device(s)

IP: localhost
Port: 8080

Resource Catalog

HTTP {?} REQUEST
http://192.168.0.10:8080/

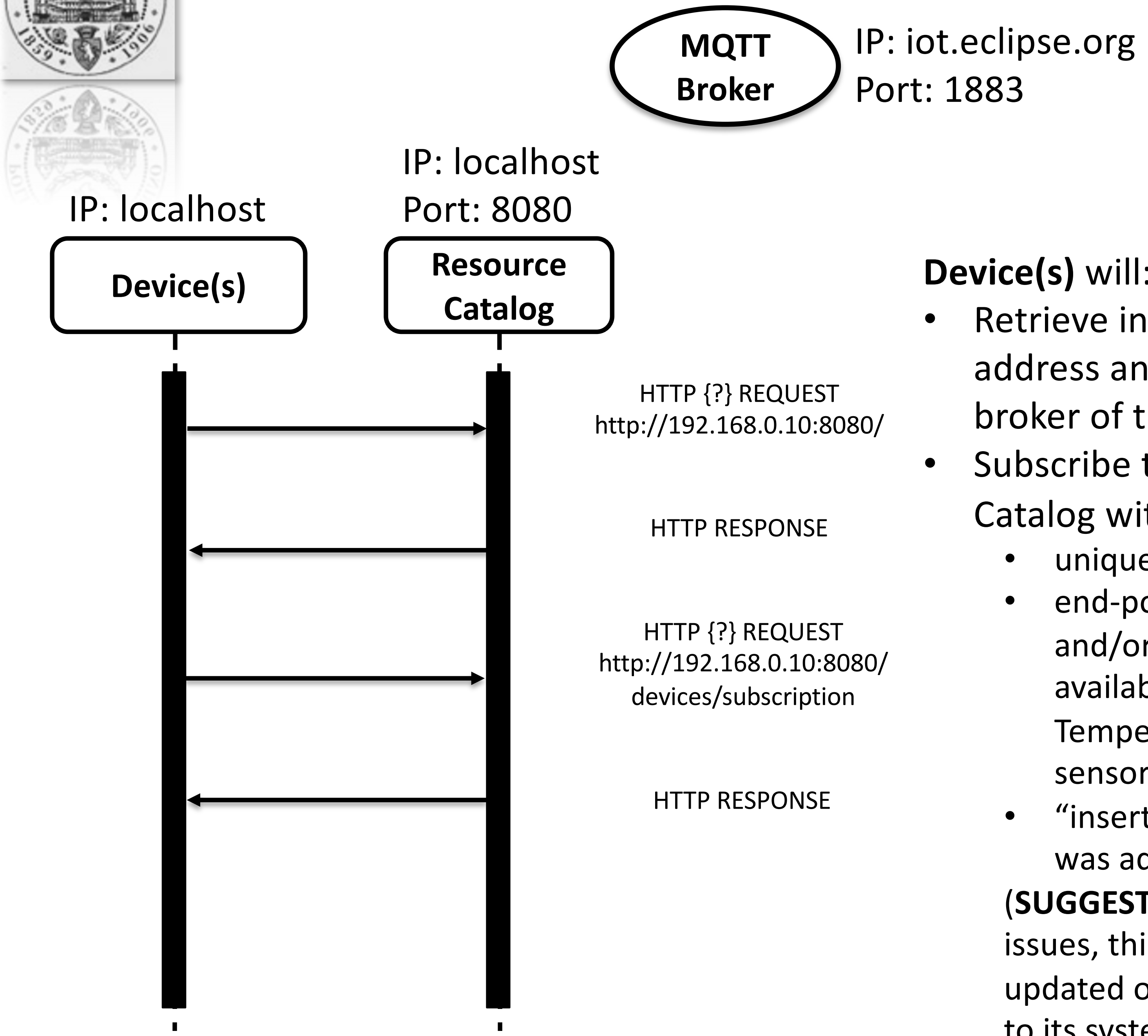
HTTP RESPONSE

HTTP {?} REQUEST
http://192.168.0.10:8080/
devices/subscription

HTTP RESPONSE

Device(s) will:

- Retrieve information regarding the IP address and port and/or MQTT broker of the available service
- Subscribe the first time to the Service Catalog with the followings info:
  - unique deviceID (**uuid library**)
  - end-points (i.e. Rest Web Services and/or MQTT topics)
    available resources (e.g. Temperature, Humidity and Motion sensor)
  - “insert-timestamp” when this device was added



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost

Device(s)

IP: localhost
Port: 8080

Resource Catalog

HTTP {?} REQUEST
http://192.168.0.10:8080/

HTTP RESPONSE

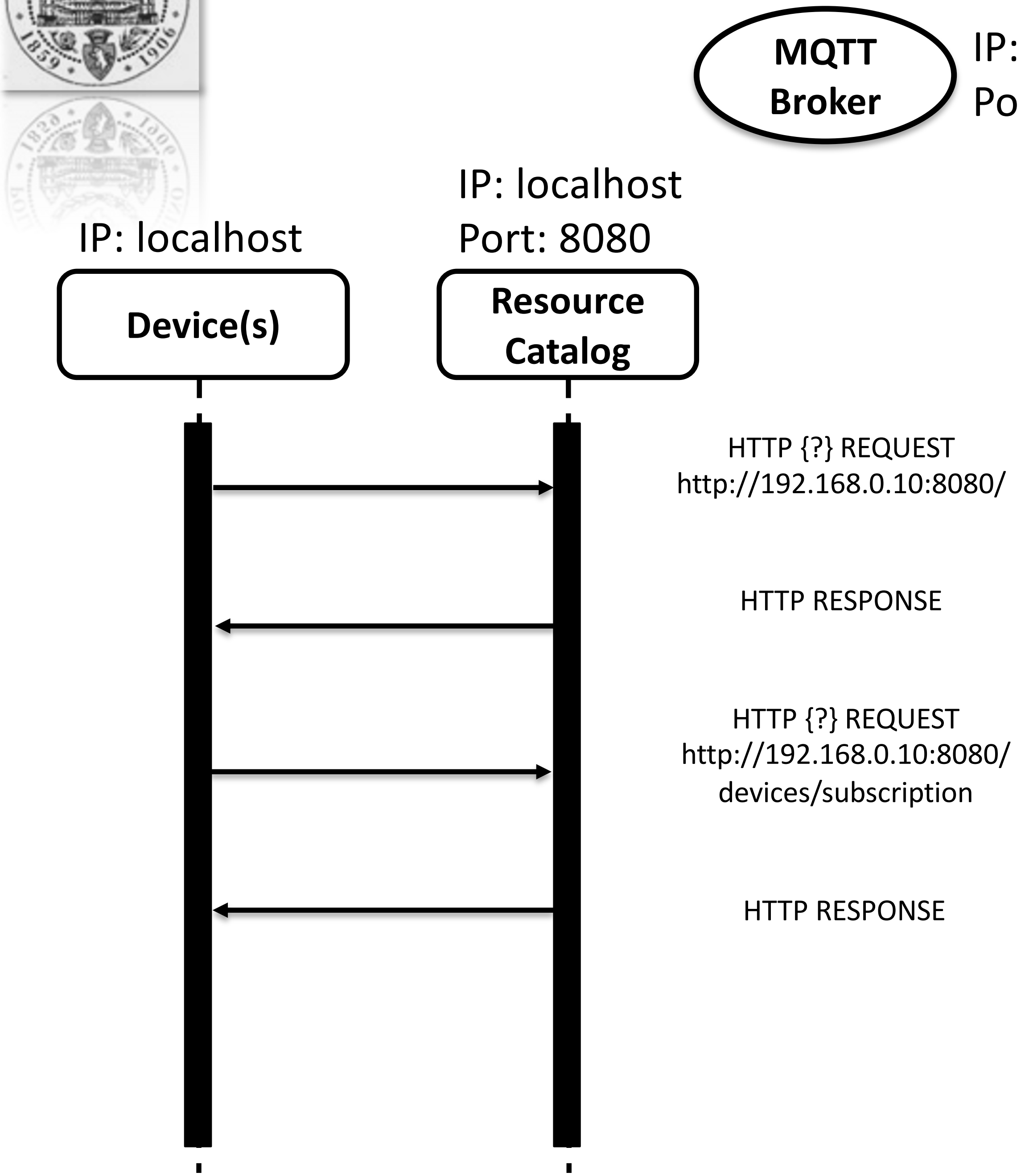
HTTP {?} REQUEST
http://192.168.0.10:8080/
devices/subscription

HTTP RESPONSE

Device(s) will:

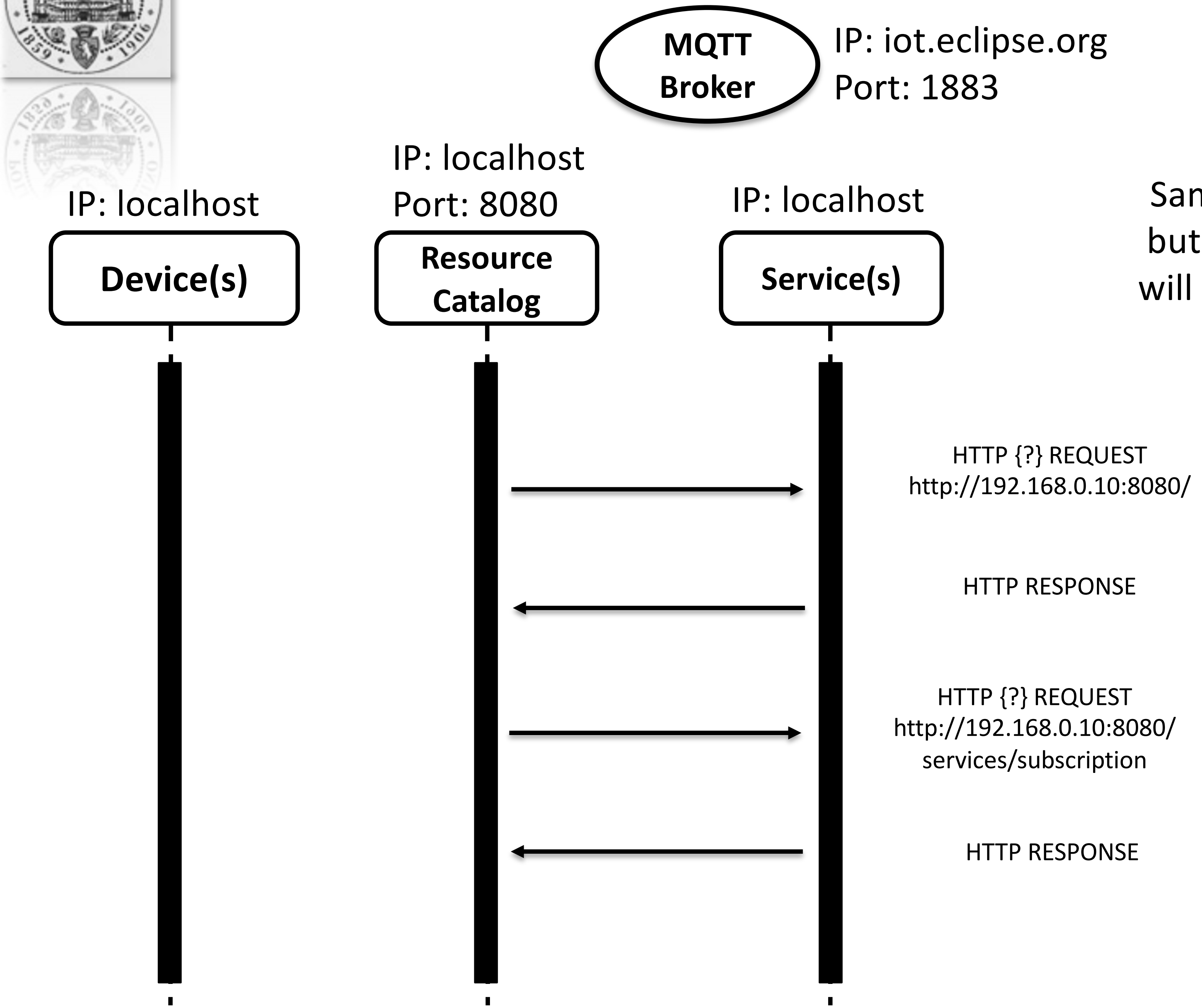
- Retrieve information regarding the IP address and port and/or MQTT broker of the available service
- Subscribe the first time to the Service Catalog with the followings info:
  - unique deviceID (**uuid library**)
  - end-points (i.e. Rest Web Services and/or MQTT topics)
    available resources (e.g. Temperature, Humidity and Motion sensor)
  - “insert-timestamp” when this device was added

  (**SUGGESTION:** to avoid synchronization issues, this attribute is managed and updated only by the Catalog according to its system clock)



Device(s) will:

- Retrieve information regarding the IP address and port and/or MQTT broker of the available service
- Subscribe the first time to the Service Catalog
- Renew the subscription each 1 minute, after 2 minutes without receiving info the Catalog will eliminate the device, or will change a status flag connected inside the resource



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost

Device(s)

IP: localhost
Port: 8080

Resource Catalog

IP: localhost

Service(s)

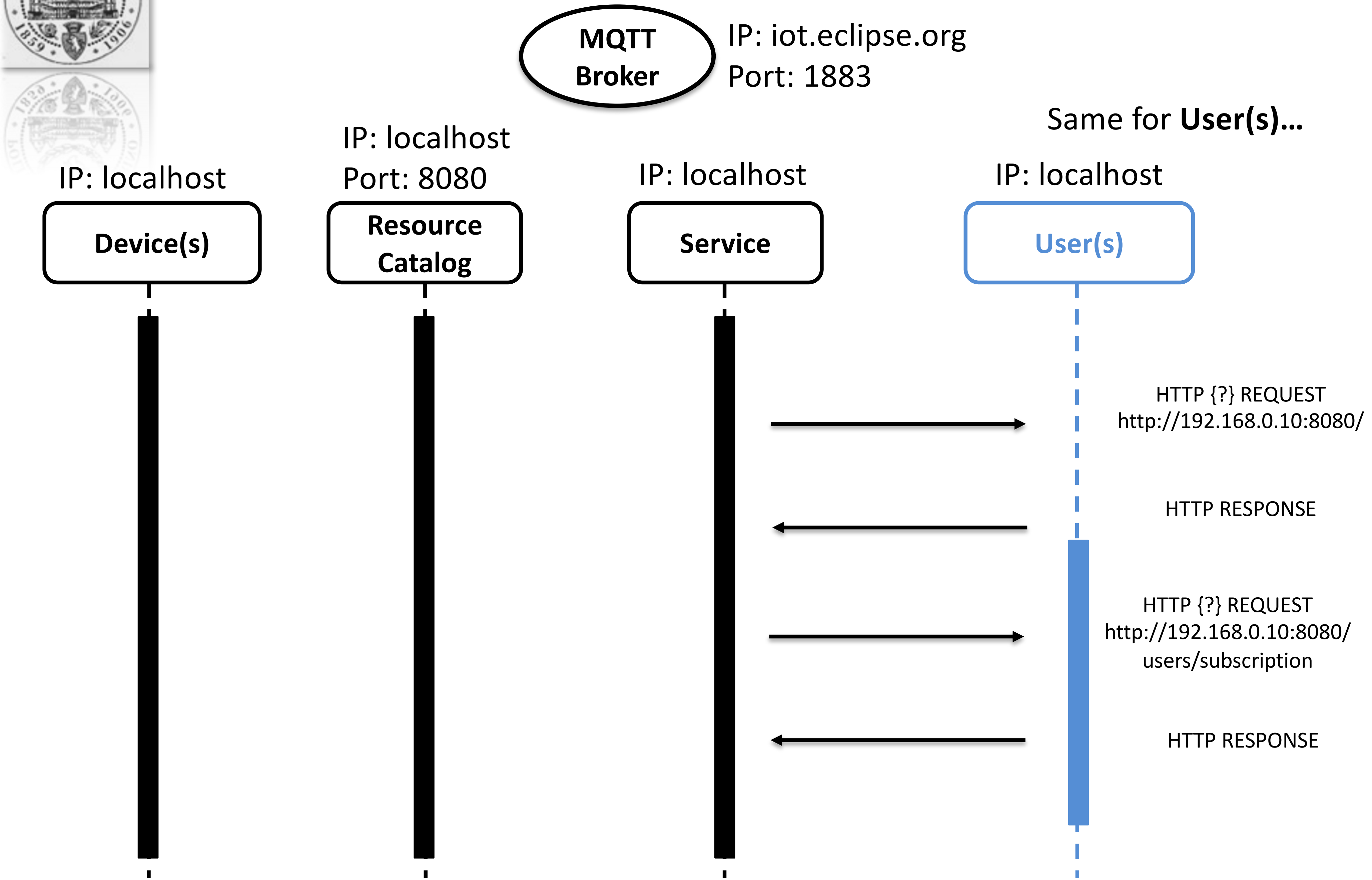
HTTP {?} REQUEST
http://192.168.0.10:8080/

HTTP RESPONSE

HTTP {?} REQUEST
http://192.168.0.10:8080/
services/subscription

HTTP RESPONSE

Same for **Service(s)**,
but the subscription
will contain different
parameters…



MQTT Broker

IP: iot.eclipse.org
Port: 1883

Same for **User(s)…**

IP: localhost

Device(s)

IP: localhost
Port: 8080

Resource Catalog

IP: localhost

Service

IP: localhost

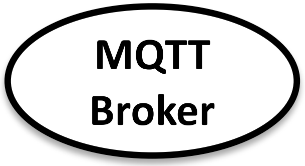
User(s)

HTTP {?} REQUEST
http://192.168.0.10:8080/

HTTP RESPONSE

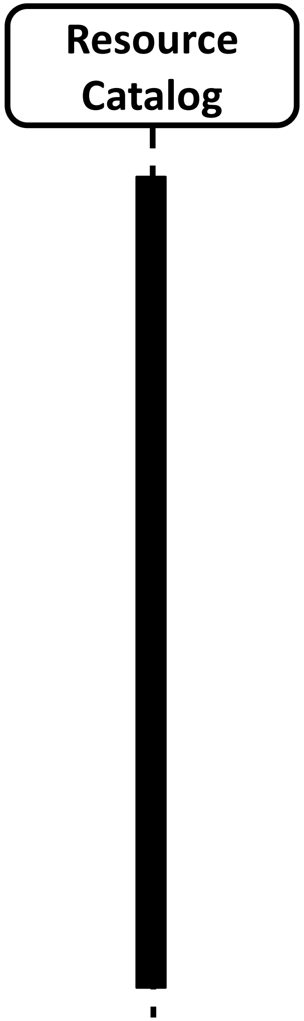
HTTP {?} REQUEST
http://192.168.0.10:8080/
users/subscription

HTTP RESPONSE



IP: iot.eclipse.org
Port: 1883

IP: localhost
Port: 8080



The service catalog will store in JSON all the information of subscriptions for devices, services, and users

And will offers APIs tree to retrieve specific information, such as:

- **Retrieve devices list**

HTTP {?} REQUEST http://192.168.0.10:8080/devices/

- **Retrieve a device with unique ID**

HTTP {?} REQUEST http://192.168.0.10:8080/devices/{uid}

- **Retrieve services list**

HTTP {?} REQUEST http://192.168.0.10:8080/services/

- **Retrieve a service with unique ID**

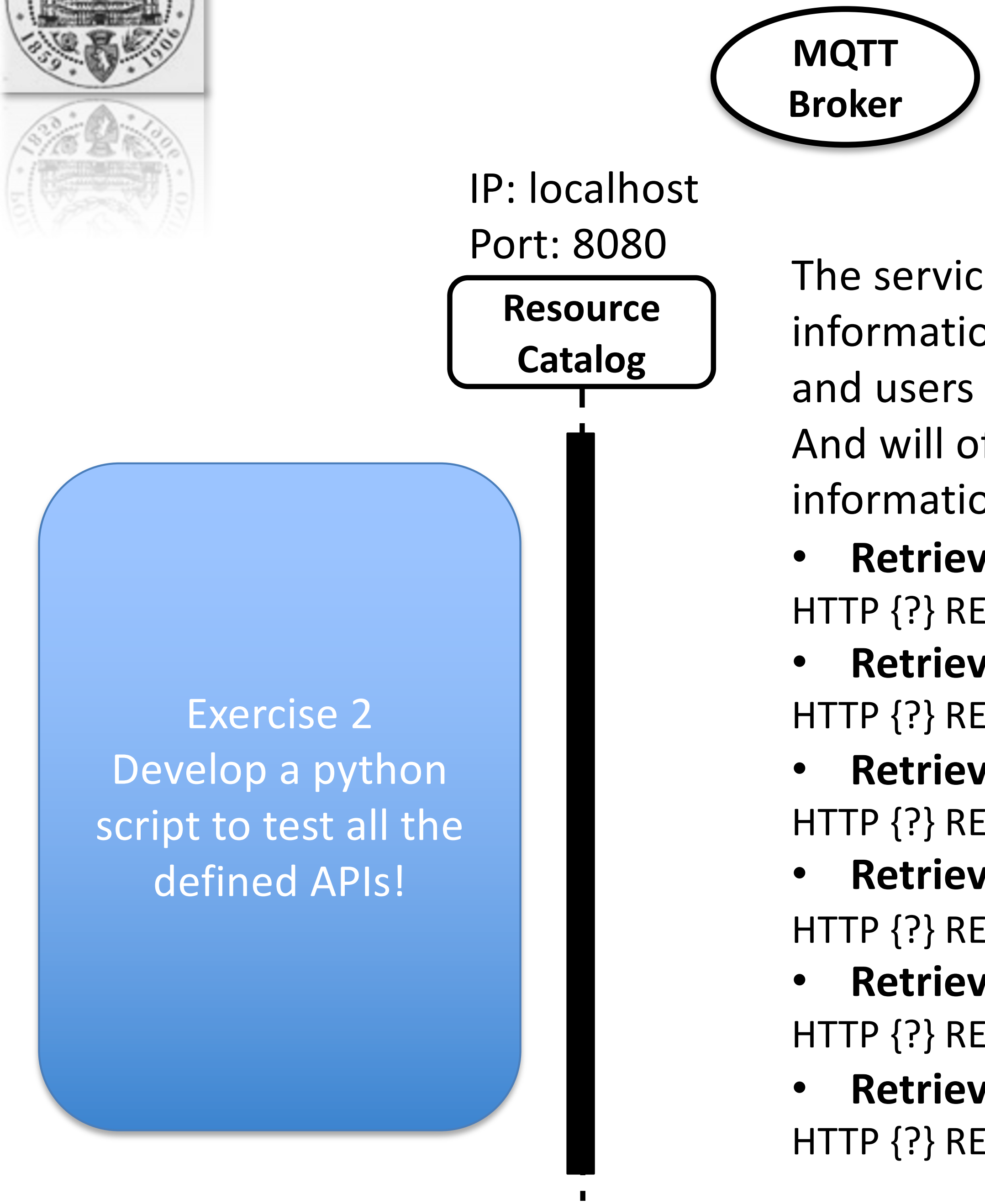
HTTP {?} REQUEST http://192.168.0.10:8080/services/{uid}

- **Retrieve users list**

HTTP {?} REQUEST http://192.168.0.10:8080/users/

- **Retrieve a user with unique ID**

HTTP {?} REQUEST http://192.168.0.10:8080/users/{uid}



MQTT Broker

IP: iot.eclipse.org
Port: 1883

IP: localhost
Port: 8080

Resource Catalog

Exercise 2
Develop a python script to test all the defined APIs!

The service catalog will store in JSON all the information of subscriptions for devices, services, and users

And will offers APIs tree to retrieve specific information, such as:

- **Retrieve devices list**

HTTP {?} REQUEST http://192.168.0.10:8080/devices/

- **Retrieve a device with unique ID**

HTTP {?} REQUEST http://192.168.0.10:8080/devices/{uid}

- **Retrieve services list**

HTTP {?} REQUEST http://192.168.0.10:8080/services/

- **Retrieve a service with unique ID**

HTTP {?} REQUEST http://192.168.0.10:8080/services/{uid}

- **Retrieve users list**

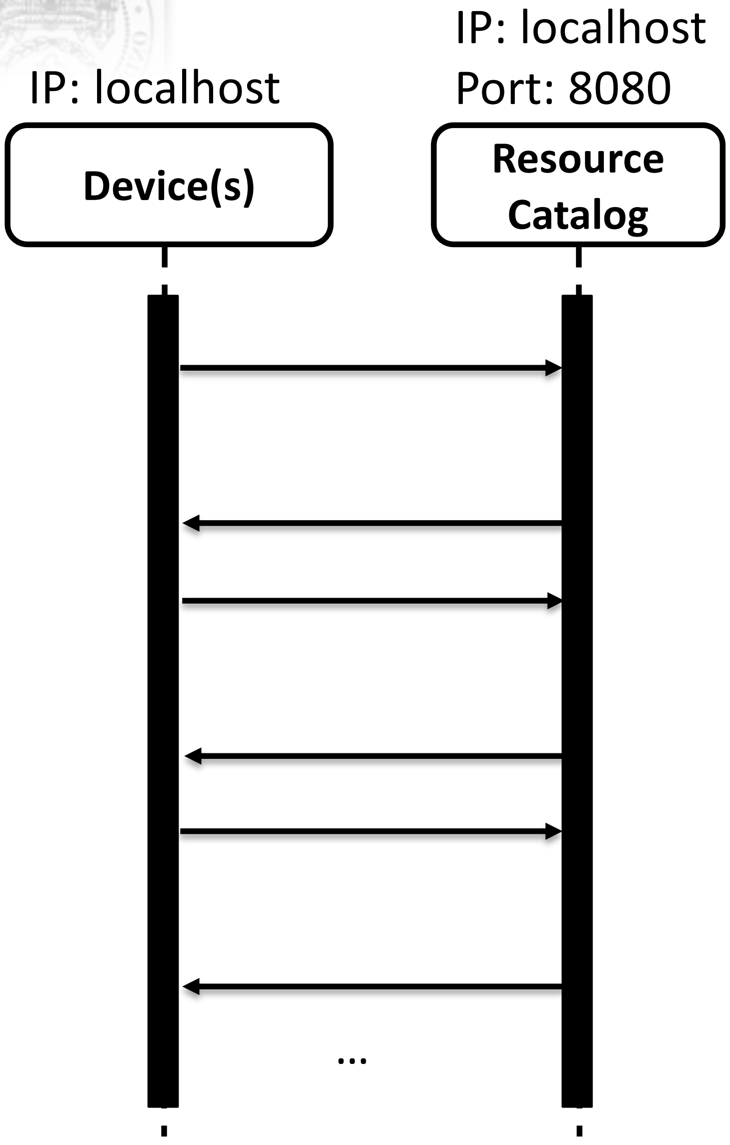
HTTP {?} REQUEST http://192.168.0.10:8080/users/

- **Retrieve a user with unique ID**

HTTP {?} REQUEST http://192.168.0.10:8080/users/{uid}

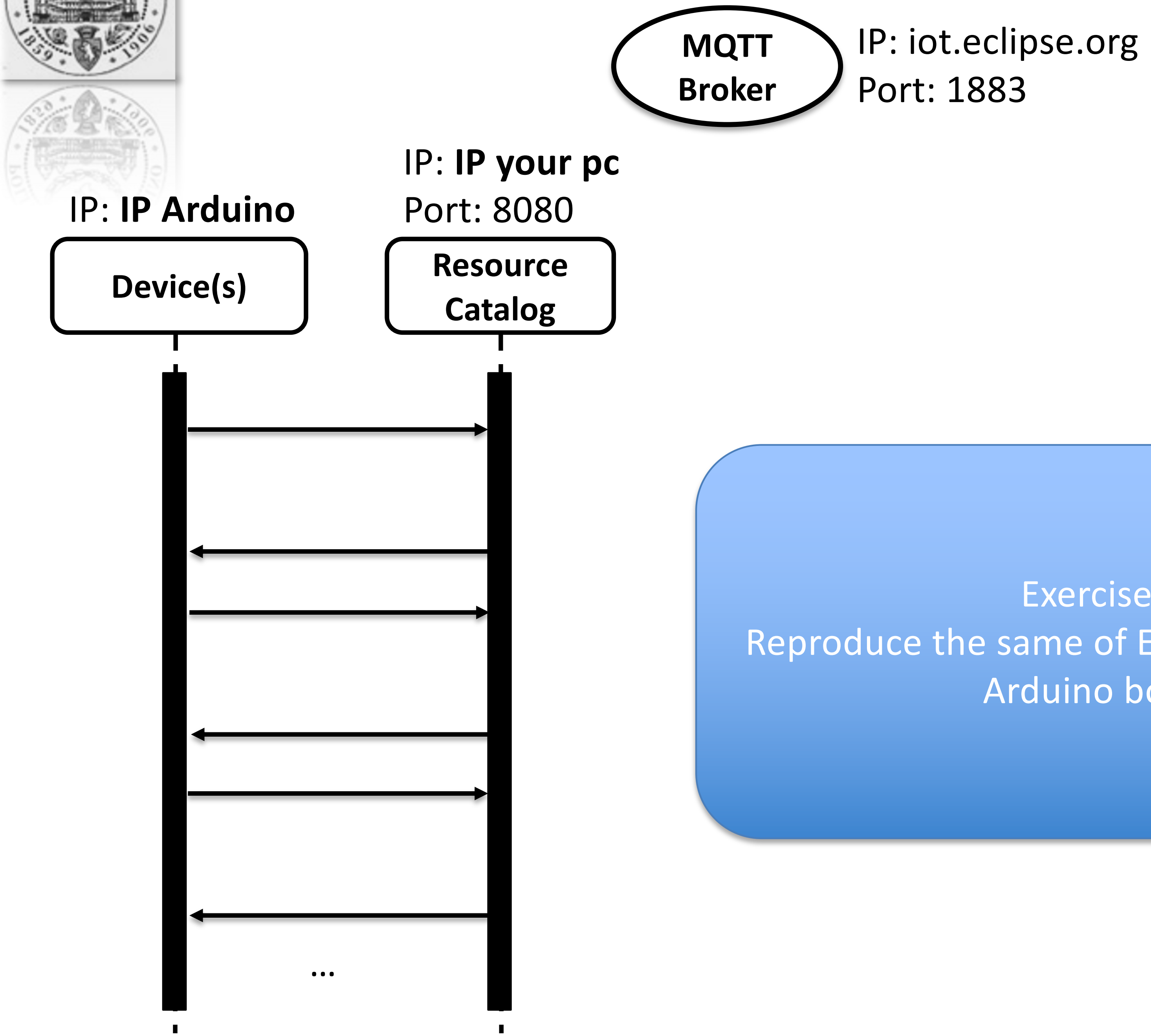
MQTT Broker

IP: iot.eclipse.org
Port: 1883

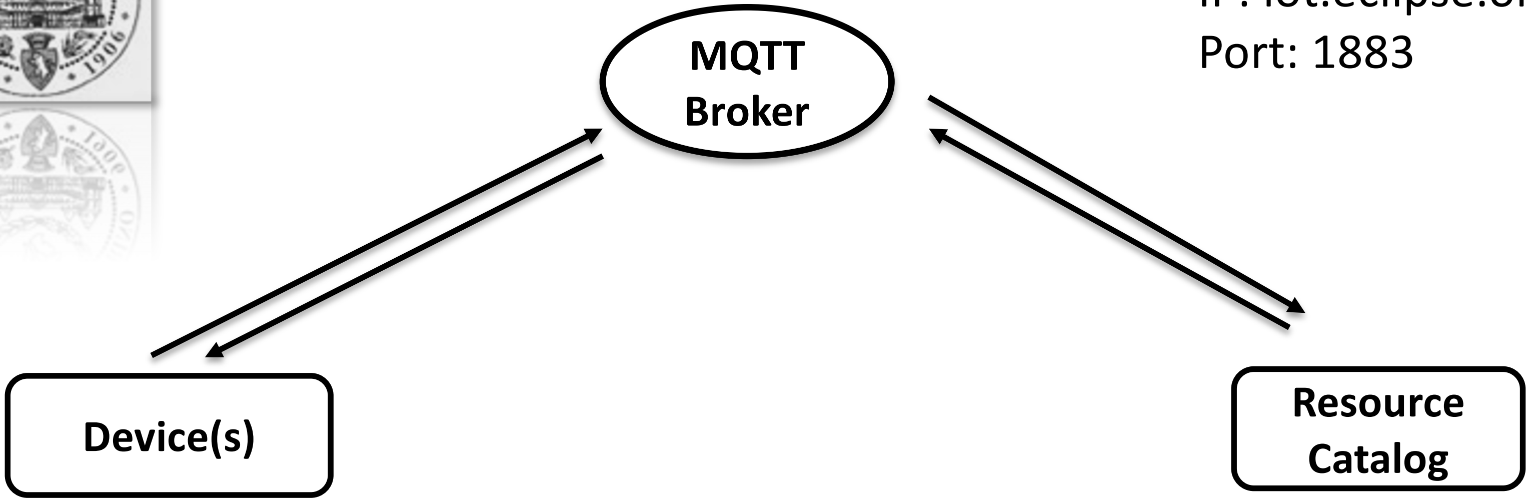


Exercise 3
Develop a python script to test an IoT devices that will:

i) Retrieve catalog information
ii) Subscribe to catalog
iii) Re-subscribe each 1 minute



Exercise 4
Reproduce the same of Exercise 3 on top of Arduino board



IP: iot.eclipse.org
Port: 1883

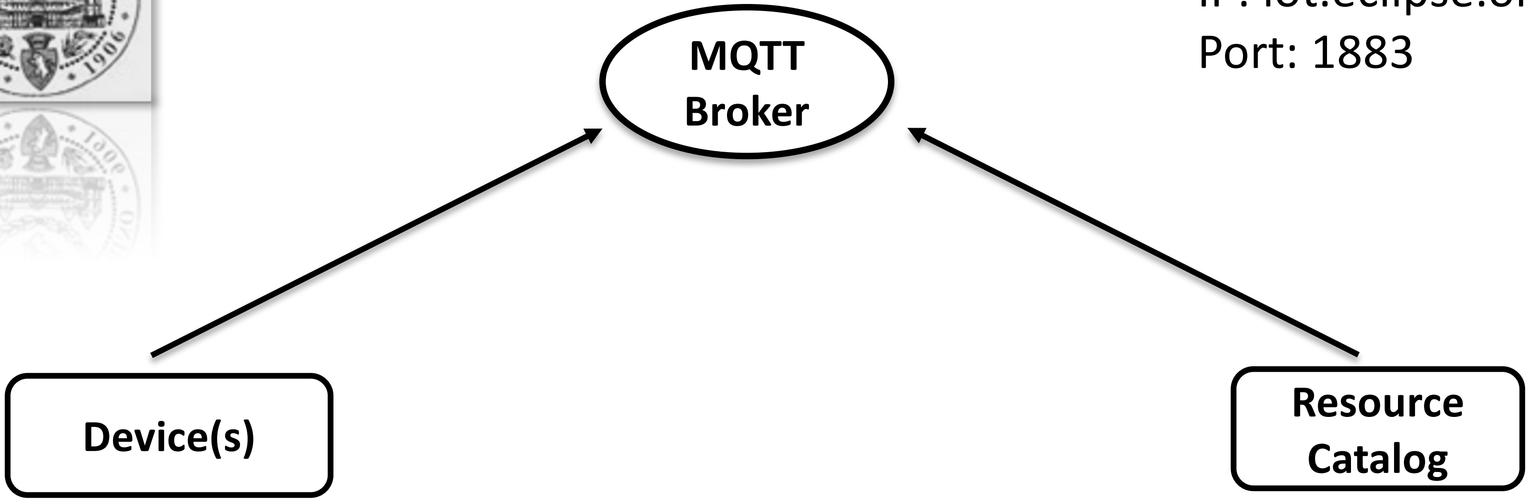
Exercise 5

The service catalog must manage also devices with MQTT protocols, reproduce the same functionalities by employing a specific topic tree to receive:

e.g. /tiot/{group_number}/catalog/devices/subscription

And to send response:

e.g. /tiot/{group_number}/catalog/devices/subscription/{uid}



IP: iot.eclipse.org
Port: 1883

Exercise 5

The service catalog must manage also devices with MQTT protocols, reproduce the same functionalities by employing a specific topic tree to receive:

e.g. /tiot/{group_number}/catalog/devices/subscription

And to send response:

e.g. /tiot/{group_number}/catalog/devices/subscription/{uid}

Exercise 6

Implement Exercise 2 to test this configuration that exploits MQTT