



# cloud.iO Technical Presentation

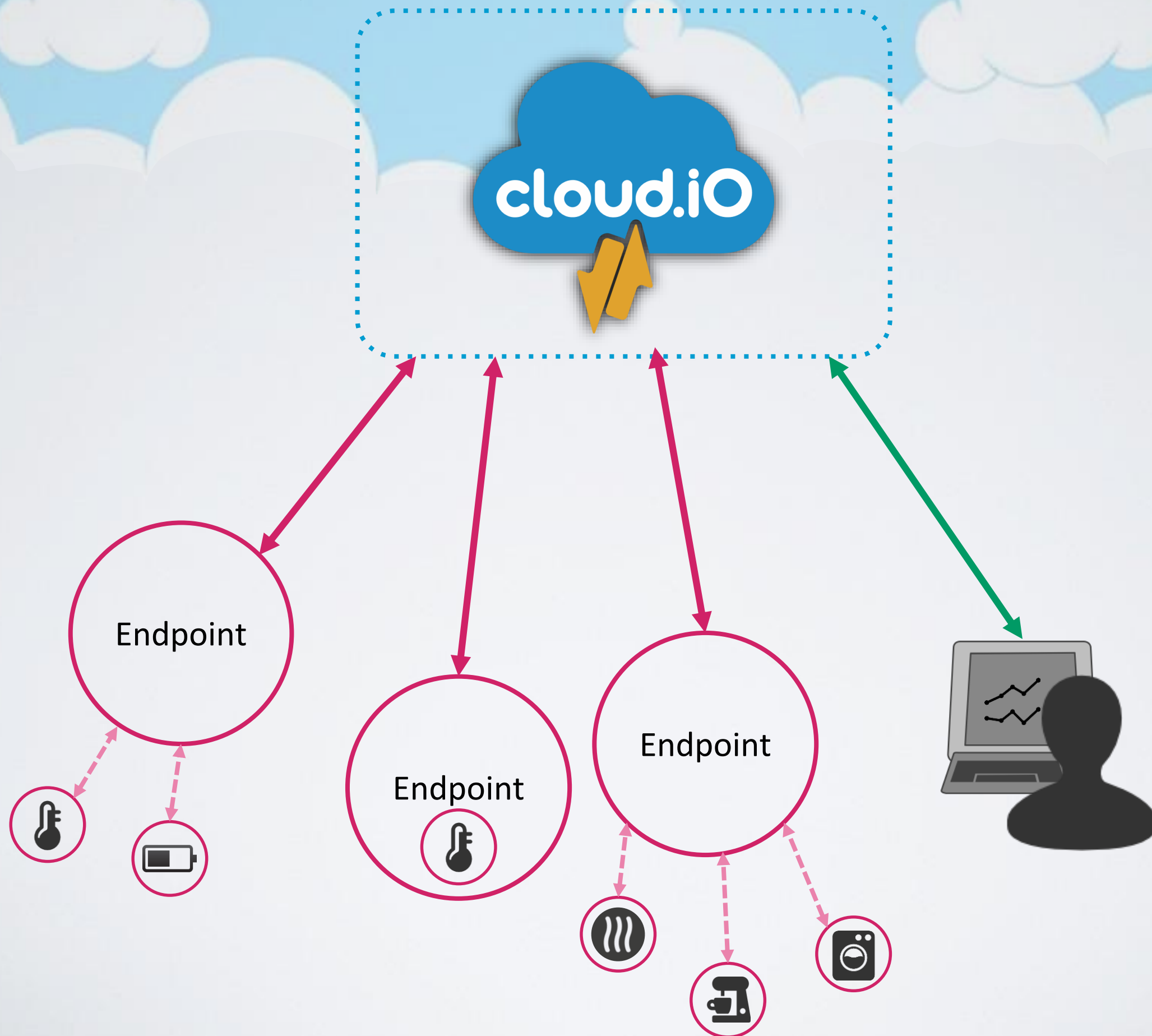
Lucas Bonvin





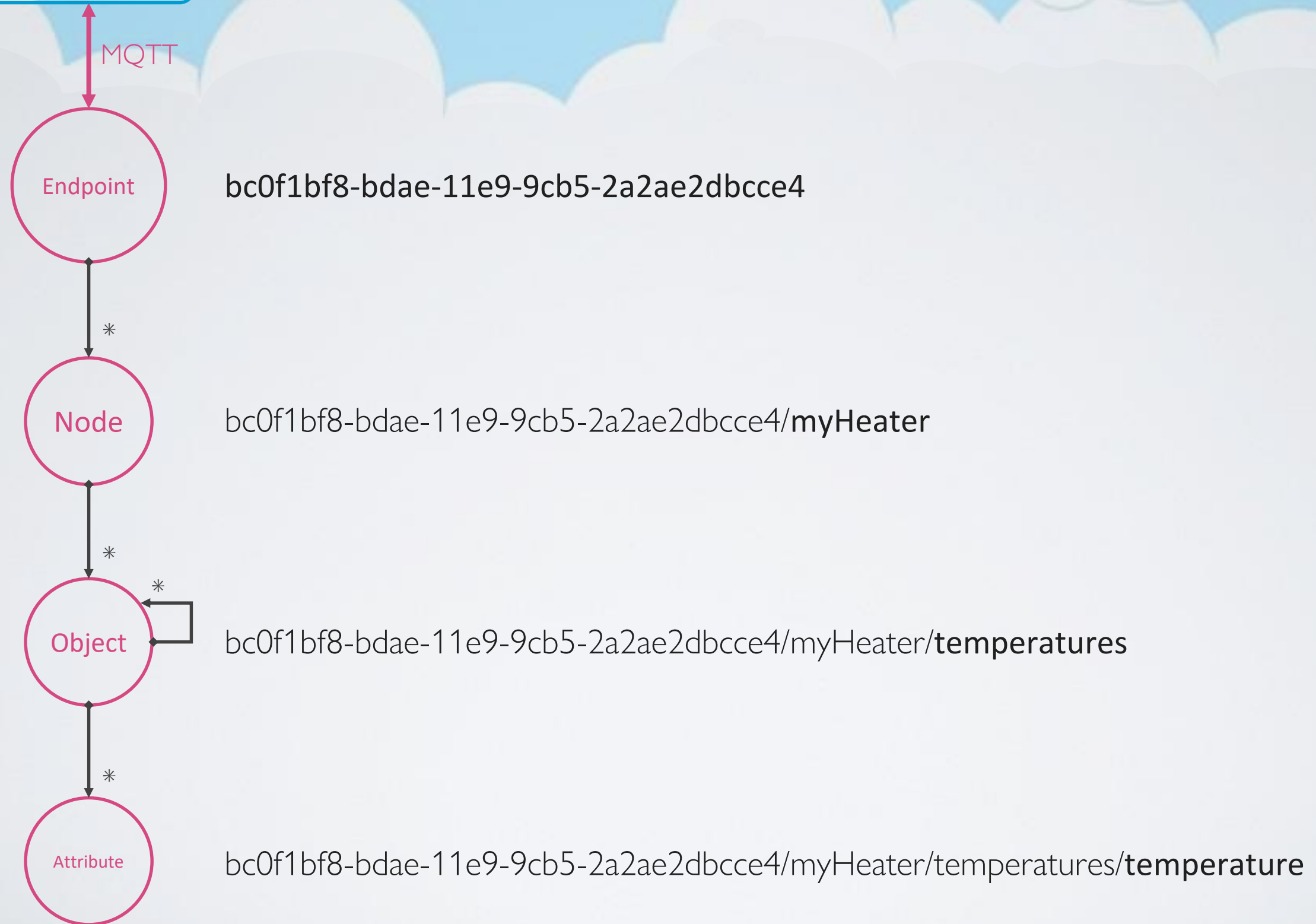
Quick Reminder of **cloud.iO**

# cloud.iO environment





# Endpoint structure



# Topic / json Object model



EndpointUUID: bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4

```
{
  "endpointUuid": "bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4",
  "friendlyName": "test",
  "logLevel": "ERROR",
  "online": false
  "blocked": false,
  "endpoint": {
    "nodes": {
      "myHeater": {
        "implements": [],
        "objects": {
          "temperatures": {
            "attributes": {
              "setPointTemperature": {
                "constraint": "SetPoint",
                "timestamp": -1.0,
                "type": "Number",
                "value": 21.0
              },
            },
            "temperature": {
              "constraint": "Measure",
              "timestamp": 1575547591858.0,
              "type": "Number",
              "value": 25.000000762939507
            }
          },
        },
        "conforms": null,
        "objects": {}
      }
    }
  }
}
```

Topic:

@update/ bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4  
/myHeater/temperatures/temperature

```
{
  „type“: „Number“,
  „constraint“: „Measure“,
  „timestamp“: 1575547591859.0,
  „value“: 25
}
```



# cloud.iO Features & Example



# RESTful API



## UserManagement



POST	/api/v1/createUser	
POST	/api/v1/getUser	
DELETE	/api/v1/deleteUser	
POST	/api/v1/modifyUserPassword	
POST	/api/v1/addUserAuthority	
DELETE	/api/v1/removeUserAuthority	
POST	/api/v1/getUserList	

## UserAccessControl



POST	/api/v1/getUserAccessRight	
POST	/api/v1/addUserAccessRight	
POST	/api/v1/modifyUserAccessRight	
DELETE	/api/v1/removeUserAccessRight	
POST	/api/v1/giveUserAccessRight	

- Http tools

- User authentication
- Support Https

- Tools for User and Administrator

- User & User Group Management
- Endpoint Management
- Certificate
- History
- Logging
- Remote Jobs Execution

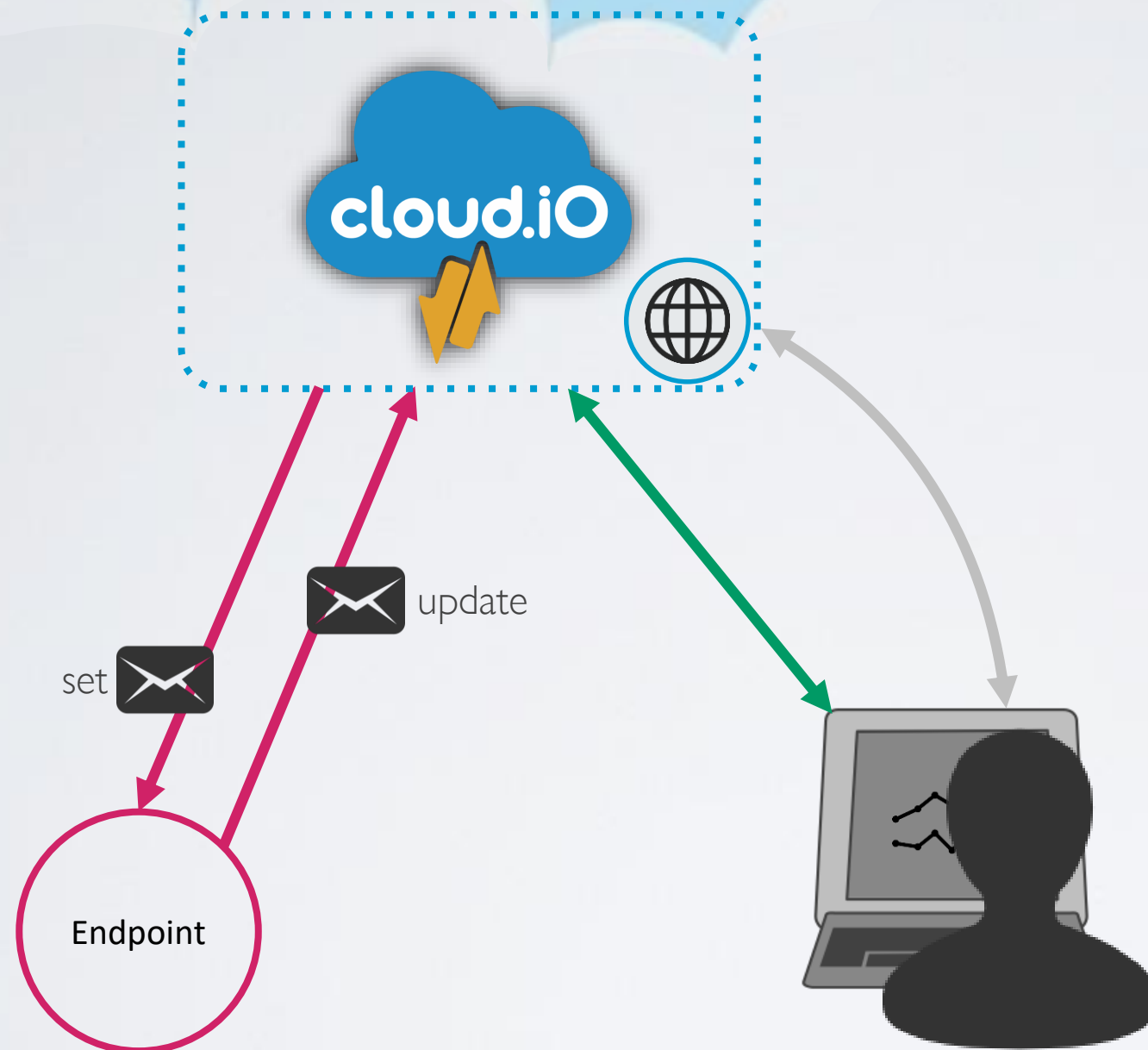


Available online

...



# Device Monitoring



- **Access Control**

- Control who can access cloud.iO resources (From Endpoint to Attribute)

- **Endpoint Library**

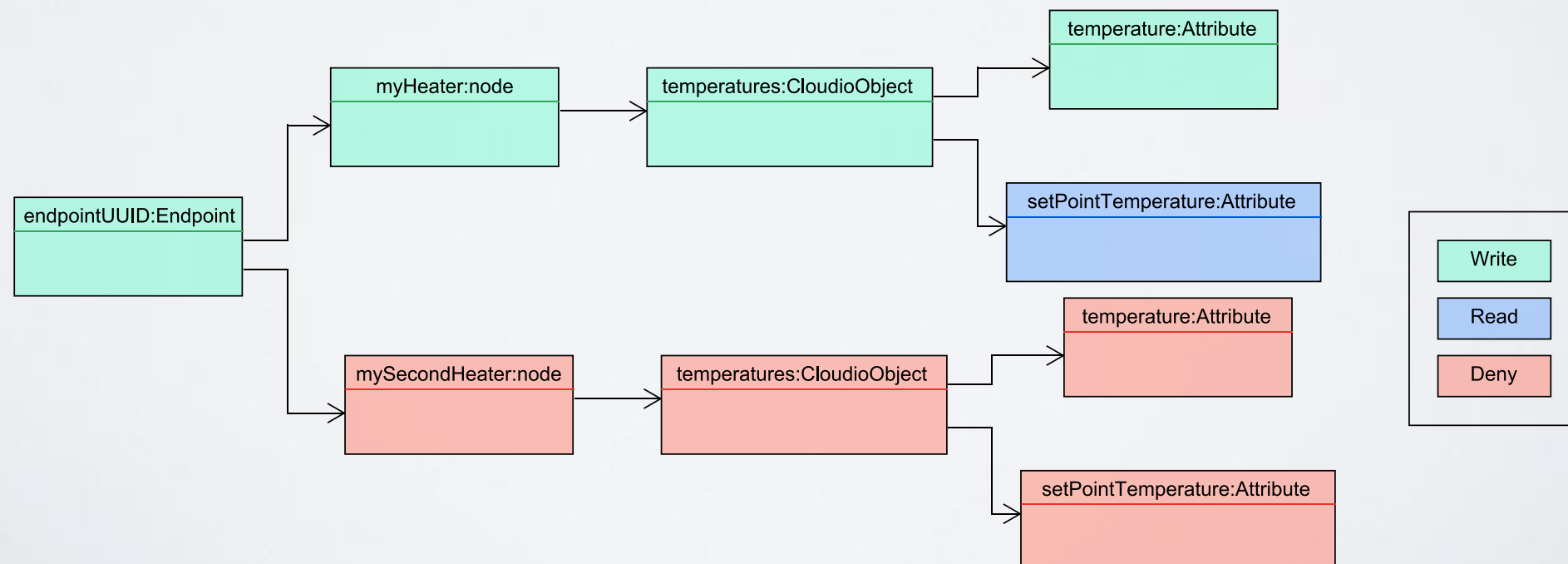
- Send @update, receive @set
- Concept of Transaction for low bandwidth Endpoint

# Device Monitoring - Access control (1)

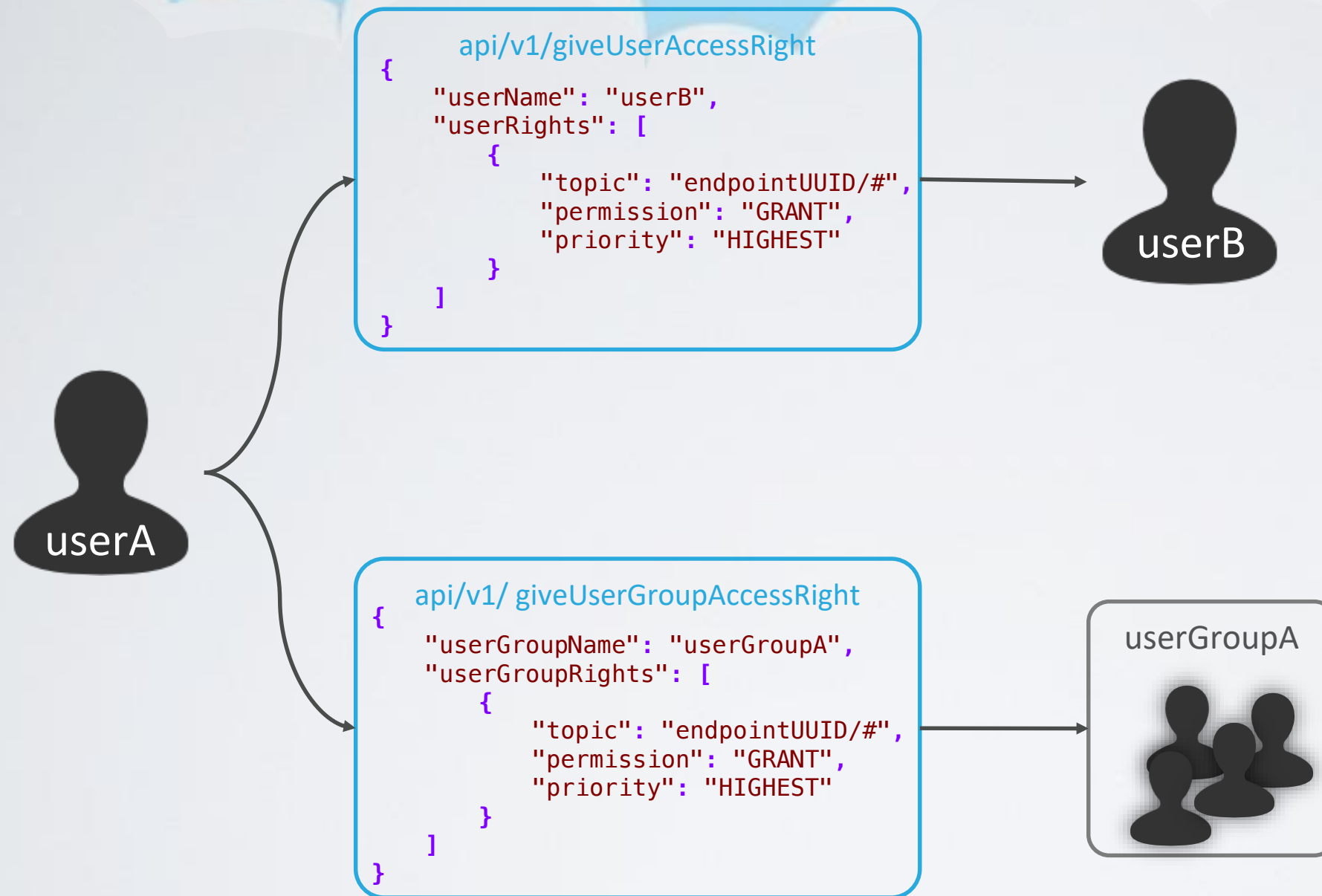


Access right topic based for a User

```
{
  "permissions": {
    "endpointUUID/#": {
      "permission": "WRITE",
      "priority": "LOW"
    },
    "endpointUUID/mySecondHeater/#": {
      "permission": "DENY",
      "priority": "HIGH"
    },
    "endpointUUID/myHeater/temperatures/setPointTemperature": {
      "permission": "READ",
      "priority": "HIGHEST"
    }
  }
}
```



# Device Monitoring - Access control (2)



## • Access control

- User own created Endpoint
- User and User Group
- User can give access to its Endpoints to another User
- User can give access to its Endpoints to a UserGroup

# Device Monitoring – Endpoint Library(1)



## Java Example of simple Heater System

```
public class Application {
    CloudioEndpoint myEndpoint;
    DemoHeater demoHeater;

    static Logger logger = LogManager.getLogger(Application.class);

    public static void main(String[] args) {

        try {
            myEndpoint = new CloudioEndpoint("EndpointUUID");

            myEndpoint.addNode("myHeater", DemoHeater.class);

            demoHeater = myEndpoint.getNode("myHeater");

            demoHeater.temperatures.setPointTemperature.addListener(
                new CloudioAttributeListener() {
                    @Override public void attributeHasChanged
                        ↳ (CloudioAttribute attribute) {
                        logger.trace("New Temperature set");
                        setSetPointTemperature((double)attribute.getValue());
                    }
                });

            while(true){
                demoHeater.temperatures.temperature.setValue
                    ↳ (retrieveAmbientTemperature());
                Thread.sleep(1000);
            }
        } catch (Exception e) {
            logger.error("Error In Application");
        }
    }
}
```

## • Endpoint Library

- Listener for @set messages
- @update messages send with setValue



Todo link github

# Device Monitoring – Endpoint Library(2)



## Java Example of Transaction

```
public class Application {  
  
    static CloudioEndpoint myEndpoint;  
  
    public static void main(String[] args) {  
  
        myEndpoint = new CloudioEndpoint("endpointUUID");  
        //demoNode contains demoObject with demoStatus  
        //and demoMeasure  
        myEndpoint.addNode("demoNode", DemoNode.class);  
        DemoNode demoNode = myEndpoint.getNode("demoNode");  
  
        myEndpoint.beginTransaction();  
  
        //this setValue is forgotten by the transaction  
        //because of the rollback  
        demoNode.demoObject.demoStatus.setValue(20);  
        myEndpoint.rollbackTransaction();  
  
        demoNode.demoObject.demoStatus.setValue(20.0);  
  
        demoNode.demoObject.demoMeasure.setValue(20.0);  
        demoNode.demoObject.demoMeasure.setValue(21.0);  
        //only the second demoMeasure.setValue will  
        //be in the transaction  
  
        //demoStatus = 20.0 demMeasure = 21.0  
        myEndpoint.commitTransaction();  
  
    }  
}
```

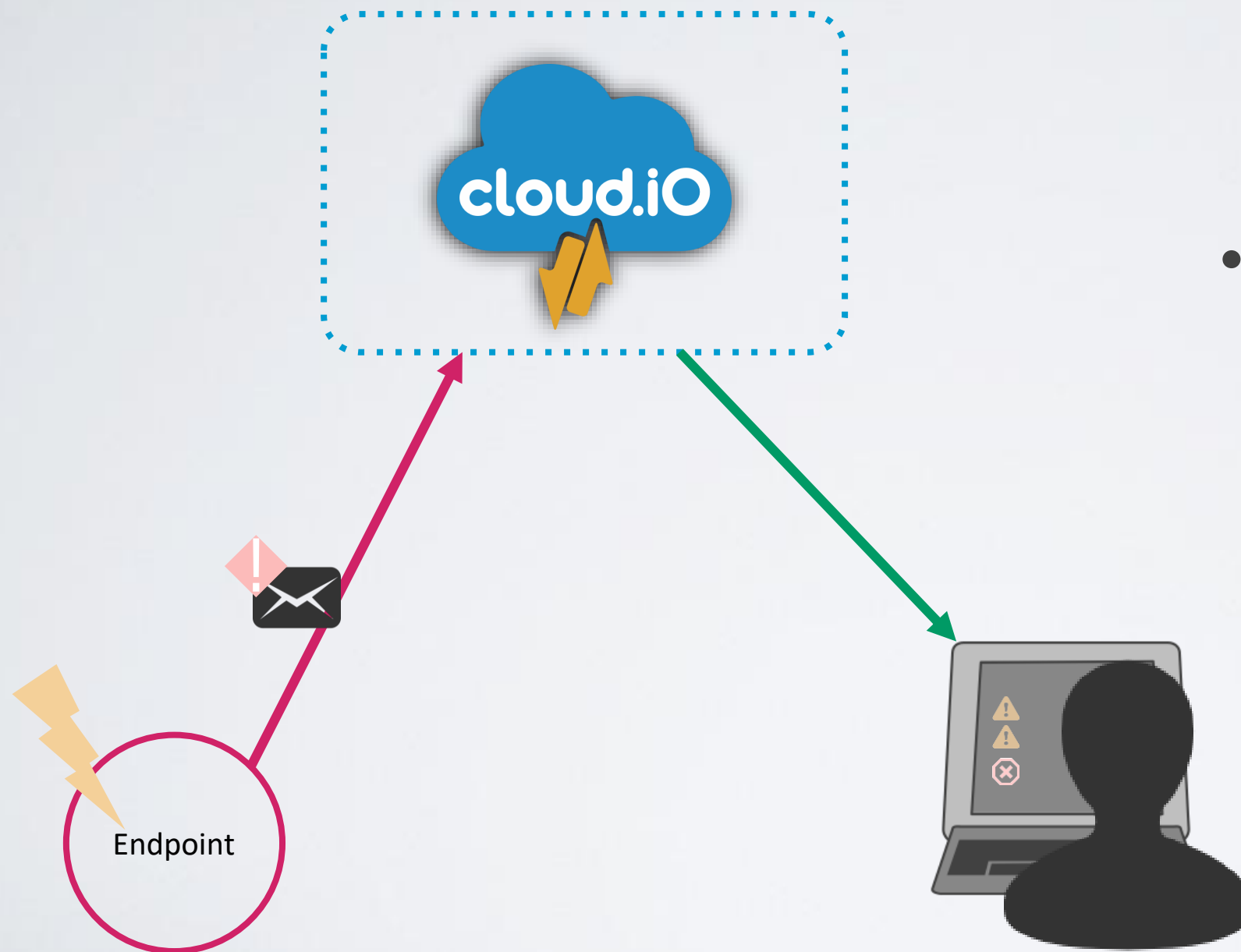
## • Transaction

- Start with beginTransaction()
- Send all updates in Transaction with commitTransaction()
- Reset Transaction with rollbackTransaction()



Todo link github

# Logging



- **Logging**

- Saved as time-series
- Possibility to filter by log level
- Retrieve through RESTful API

- **Different log level**

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- ALL



# Logging



## Java Example of Log with log4j

```
import ...
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class Application {

    static CloudioEndpoint myEndpoint;
    static Logger logger =
        LogManager.getLogger(Application.class);
    static Logger rootLogger =
        LogManager.getRootLogger();

    public static void main(String[] args) {

        // create endpoint structure
        myEndpoint = new CloudioEndpoint("endpointUUID");
        ...

        //all the following logs will be pushed
        //to the cloud according to Endpoint log level
        logger.trace("This is a trace from Application logger");
        rootLogger.trace("This is a trace from Root logger");

        logger.error("Error log");
        logger.debug("Debug log");
    }
}
```



- Logging with Endpoint library
  - Use log4j library
  - Possibility to add log Appender
  - Logs send to depending Endpoint log level

# Remote jobs Execution



Execute script.sh

script.sh

```
#!/bin/sh  
echo Hello  
world !
```

Endpoint



- Remote Jobs Execution

- Execute Shell script remotely
- Execution launched with MQTT or RESTful API
- Possibility to retrieve output of script



# Remote jobs Execution



- Remote Jobs Execution

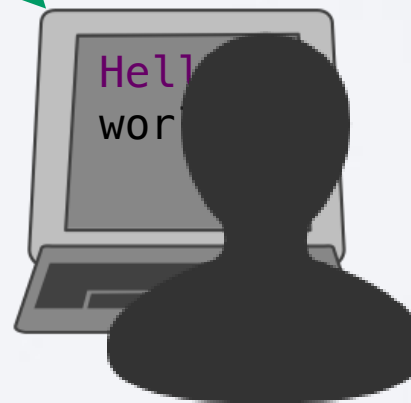
- Execute Shell script remotely
- Execution launched with MQTT or RESTful API
- Possibility to retrieve output of script

```
script.sh  
  
#!/bin/sh  
echo Hello  
world !
```

Endpoint



Output: Hello world!



# Remote jobs Execution



## Execution of Remote Jobs: Shell script

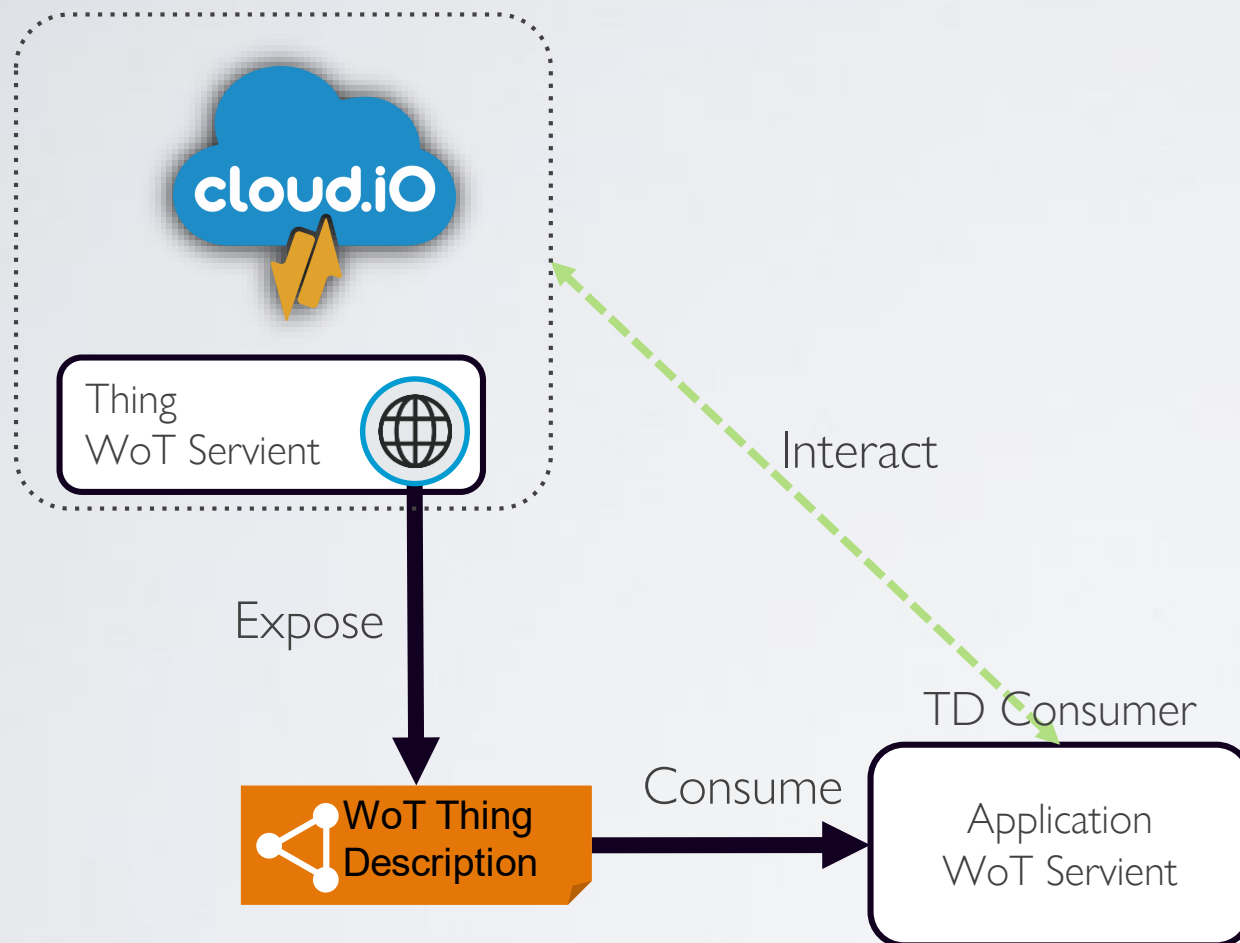
```
{  
  "endpointUuid": "bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4",  
  "jobURI": "file://script.sh",  
  "sendOutput": true,  
  "correlationID": "1a45a6ca",  
  "data": "",  
  "timeout": 5000,  
}
```

## Execution of Remote Jobs: Update script command

```
{ "endpointUuid": "bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4",  
  "jobURI": "cmd://updateJobs",  
  "getOutput" : true,  
  "correlationID" : "fa8sf64j",  
  "timeout": 5000,  
  "data": "https://github.com/lucblender/  
    ↳ downloadableFileTest/raw/master/ZipFile.zip"  
}
```

- Pre-implemented command
  - Listing all available commands/scripts
  - Update Scripts from an URL

# W3C compatibility



- **World Wide Web Consortium (W3C)**
  - Standardization organism
  - Behind HTML, RDF, CSS, PNG, SVG....
  - Standardization of the IoT
- **Web Of Thing (WoT)**
  - Candidate Recommendation
  - Enhance IoT Interoperability
  - Define a data model for the *Things* :TD
  - TD contain a self-describing API

# W3C compatibility (1)



## Thing Description of Heater System without Interaction Affordance

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:bc0f1bf8-bdae-11e9-9cb5-2a2ae2dbcce4:myHeater",
  "title": "myHeater",
  "securityDefinitions": {
    "https_sc": {
      "scheme": "basic",
      "in": "query"
    },
    "mqttsc_sc": {
      "scheme": "cert"
    }
  },
  "security": ["https_sc",
    "mqttsc_sc"],
  ...
}
```

## • Thing Description

- Normalized device twin
- Json-LD
- Self describing API
- Interaction Affordance (properties, actions, events)

# W3C compatibility (2)



## Property affordance of Heater System setpoint

```
{
  ...
  "properties": {
    ...
    "endpointUUID.myHeater.temperatures.setPointTemperature": {
      "type": "object",
      "properties": {
        "constraint": {
          "type": "string",
          "enum": ["SetPoint"]
        },
        "type": {
          "type": "string",
          "enum": ["Integer"]
        },
        "timestamp": {
          "type": "number"
        },
        "value": {
          "type": "number"
        }
      },
      "required": ["constraint", "type", "timestamp", "value"],
      "forms": [{
        "href": "https://cloud.i0:8081/api/v1/getAttribute/endpointUUID.myHeater.temperatures.setPointTemperature",
        "op": "readproperty",
        "contentType": "application/json"
      }],
      ...
    }
  },
  ...
}
```

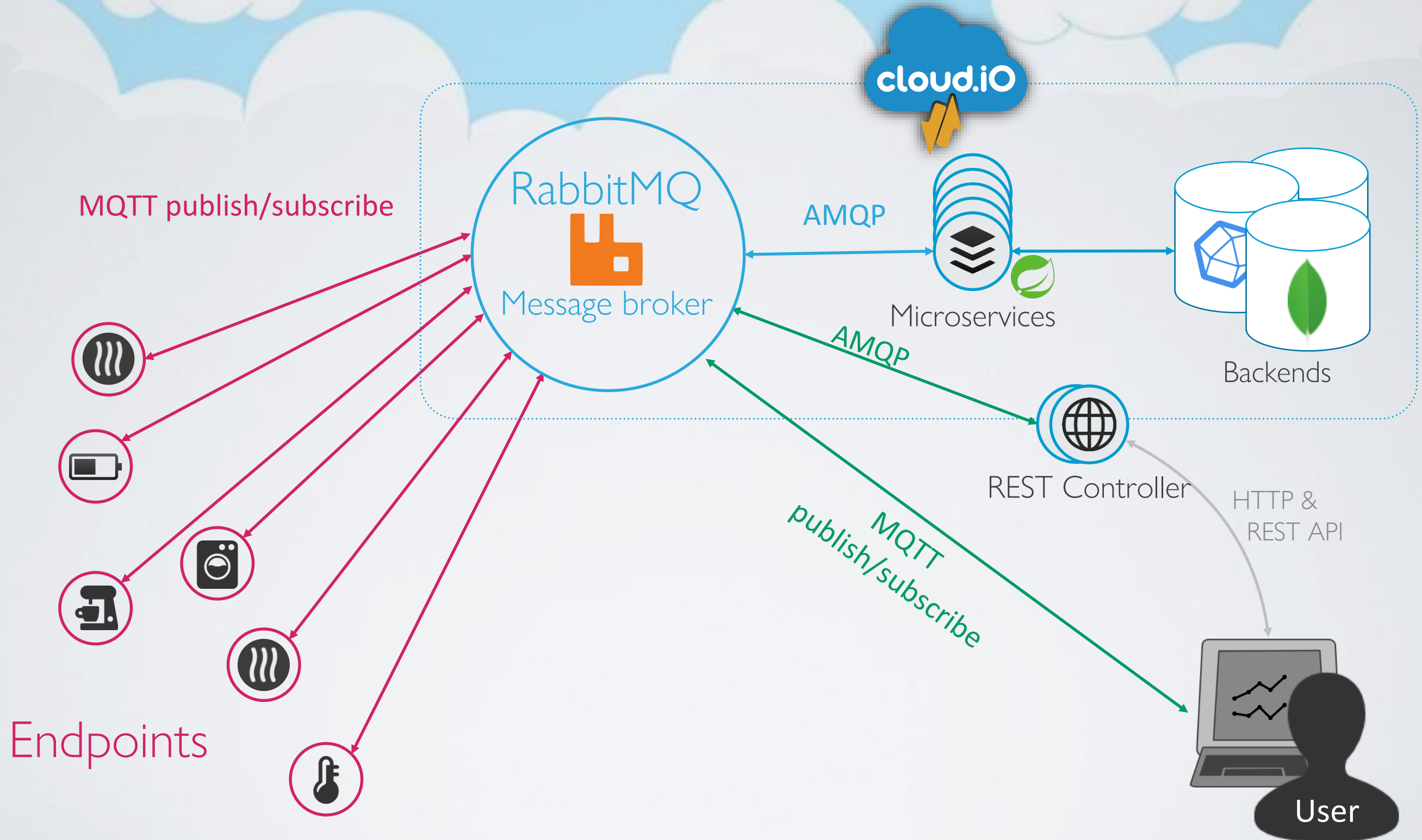
## • Thing Description

- Normalized device twin
- Json-LD
- Self describing API
- Interaction Affordance (properties, actions, events)



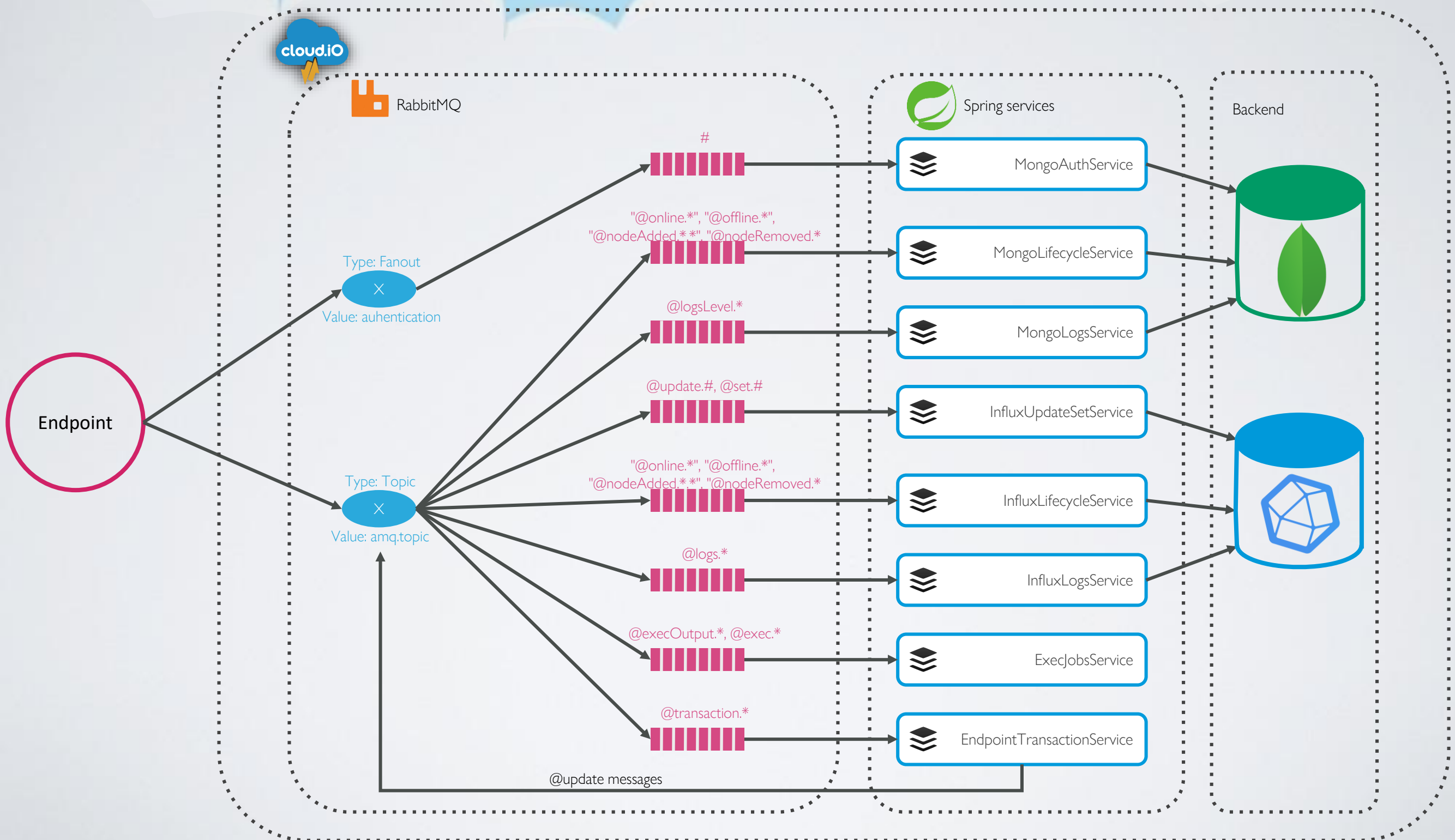
# cloud.iO Architecture

# Overview



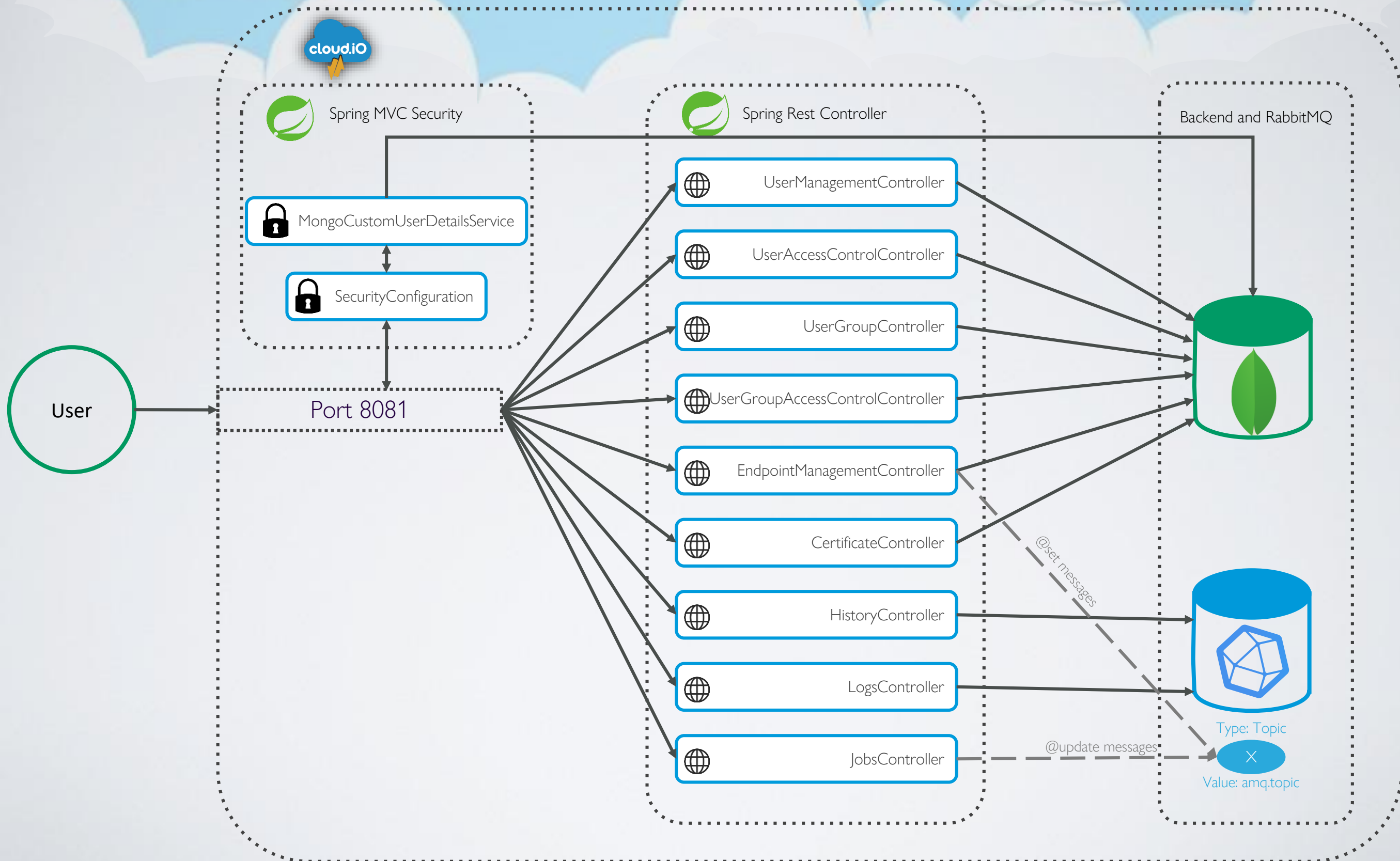


# Core messaging linked architecture





# Core Rest linked Architecture



# Backends



Authentication /  
Permission

Endpoint Repository

History  
Repository

Logs  
Repository

MongoDB  
Backend

InfluxDB  
Backend

# Open source



<http://claudio.hevs.ch>

<https://github.com/claudio-project>



claudio-rabbitmq-docker



claudio-services



claudio-documents



claudio-documentation



claudio-endpoint-java



claudio-endpoint-python



claudio-grafana-plugin



claudio-project.github.io

Questions?