oneM2M-Advance

Giacomo Tanganelli
PostDoc @ University of Pisa
g.tanganelli@iet.unipi.it

Allow CoAP on MN

- By default CoAP is not allowed on MN
- To enable it edit:

  $HOME/git/org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/om2m.product

- Add CoAP plugin

```
<plugins>
…
    <plugin id="org.eclipse.jetty.util"/>
    <plugin id="org.eclipse.om2m.binding.coap"/>
    <plugin id="org.eclipse.om2m.binding.http"/>
…
</plugins>
```

Allow CoAP on MN (2)

- Edit Configurations

```
<configurations>
…
<plugin id="org.eclipse.equinox.http.jetty" autoStart="true"
startLevel="1" />
<plugin id="org.eclipse.om2m.binding.coap" autoStart="true”
startLevel="2" />
<plugin id="org.eclipse.om2m.binding.http" autoStart="true"
startLevel="2" />
…
</configurations>
```

- Note that default port for MN is 5684

Allow CoAP on MN (3)

- Edit the code:
  - org.eclipse.om2m.binding.coap.CoAPServer
  - Change line 145:
    - requestPrimitive.setTo(request.getUri());
  - To:
    - requestPrimitive.setTo(targetId);

Clean and rebuild

- Open a shell and issue:

$ cd $HOME/git/org.eclipse.om2m

$ mvn clean install

- Reconfigure oM2M
  - IN-CSE

$HOME/git/org.eclipse.om2m/org.eclipse.om2m.site.in-cse/target/products/in-cse/linux/gtk/x86

  - MN-CSE

$HOME/git/org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/target/products/in-cse/linux/gtk/x86

Configure oM2M

- In the IN-CSE folder edit the file:

  configuration/config.ini

  - Edit:
    - org.eclipse.om2m.dbReset=**true**
    - org.eclipse.om2m.cseBaseId=**yourname-in-cse**
    - org.eclipse.om2m.cseBaseName=**yourname-in-name**

- In the MN-CSE folder edit the file:

  configuration/config.ini

  - Edit:
    - org.eclipse.om2m.dbReset=**true**
    - org.eclipse.om2m.cseBaseId=**yourname-mn-cse**
    - org.eclipse.om2m.cseBaseName=**yourname-mn-name**
    - org.eclipse.om2m.remoteCseId=**yourname-in-cse**
    - org.eclipse.om2m.remoteCseName=**yourname-in-name**

Discovery of resources

- IN connection: All resources
  - GET to 127.0.0.1:5683/~/yourname-in-cse/?fu=1
- IN connection: AE resources
  - GET to 127.0.0.1:5683/~/yourname-in-cse/?fu=1&rty=2
- IN connection: Container resources
  - GET to 127.0.0.1:5683/~/yourname-in-cse/?fu=1&rty=3
- IN connection: Container resources on MN
  - GET to 127.0.0.1:5683/~/yourname-mn-cse/?fu=1&rty=3

Exercise 1

- Modify the Exercise of last lesson to register the AE (and its container) on the MN
  - MN CoAP port 5684
  - Last lesson Exercise

    https://github.com/IoTLabUnipi/IoT-Lab/tree/master/2016/03-oneM2M/adn
- Write another application that retrieve Container registered in the MN by connecting to the IN.
  - IN CoAP port 5683

Subscriptions

- Subscriptions are used by Applications to receive asynchronous updates on the status of resources
- AEs can create subscriptions for “any” kind of event. We will focus on contentInstance subscriptions to retrieve updated readings.
- Notification messages are always in XML
- Two type of subscriptions for two type of AEs:
  - Server capable
  - Server incapable

Subscriptions – Server capable

- In order to receive notifications a CoAP server is needed
- The server must expose a resource which allows POST requests
- The server must respond with 2.01 CREATED

- Choose a port that is not already used on your machine (Ex. 5685)

Create a subscription

- Create a subscribtion for Container resources in the MN by interacting with the IN:
  - POST to 127.0.0.1:5683/~/yourname-mn-cse/yourname-mn-name/TempApp/DATA
  - Payload in json:

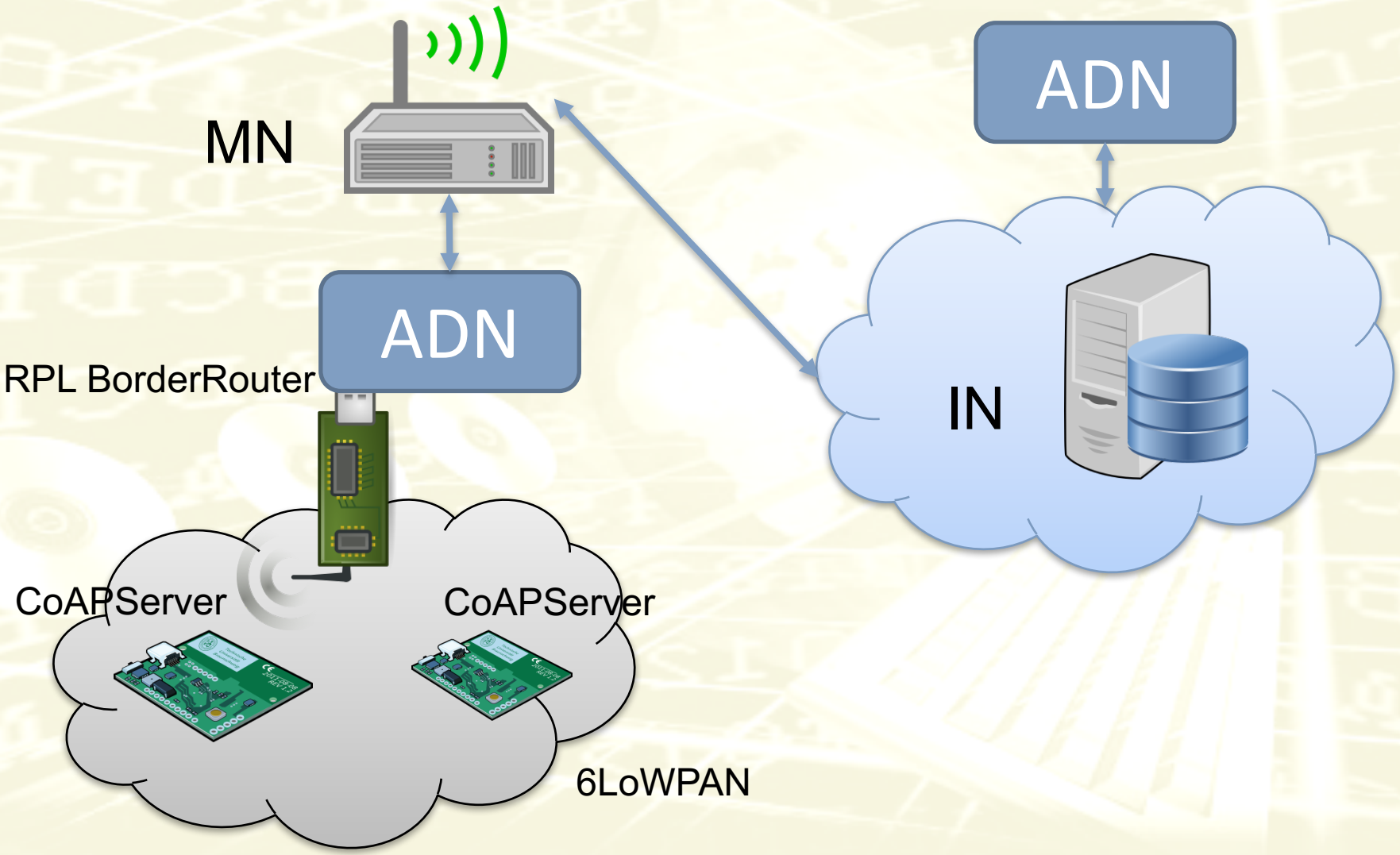
```
{
    “m2m:sub”:{
        “rn”: “Monitor”,
        “nu”: “coap://127.0.0.1:5685/yourResourceName”,
        “nct”: 2
    }
}
```

rn = ResourceName
nu = NotificationURI
nct = NotificationContentType

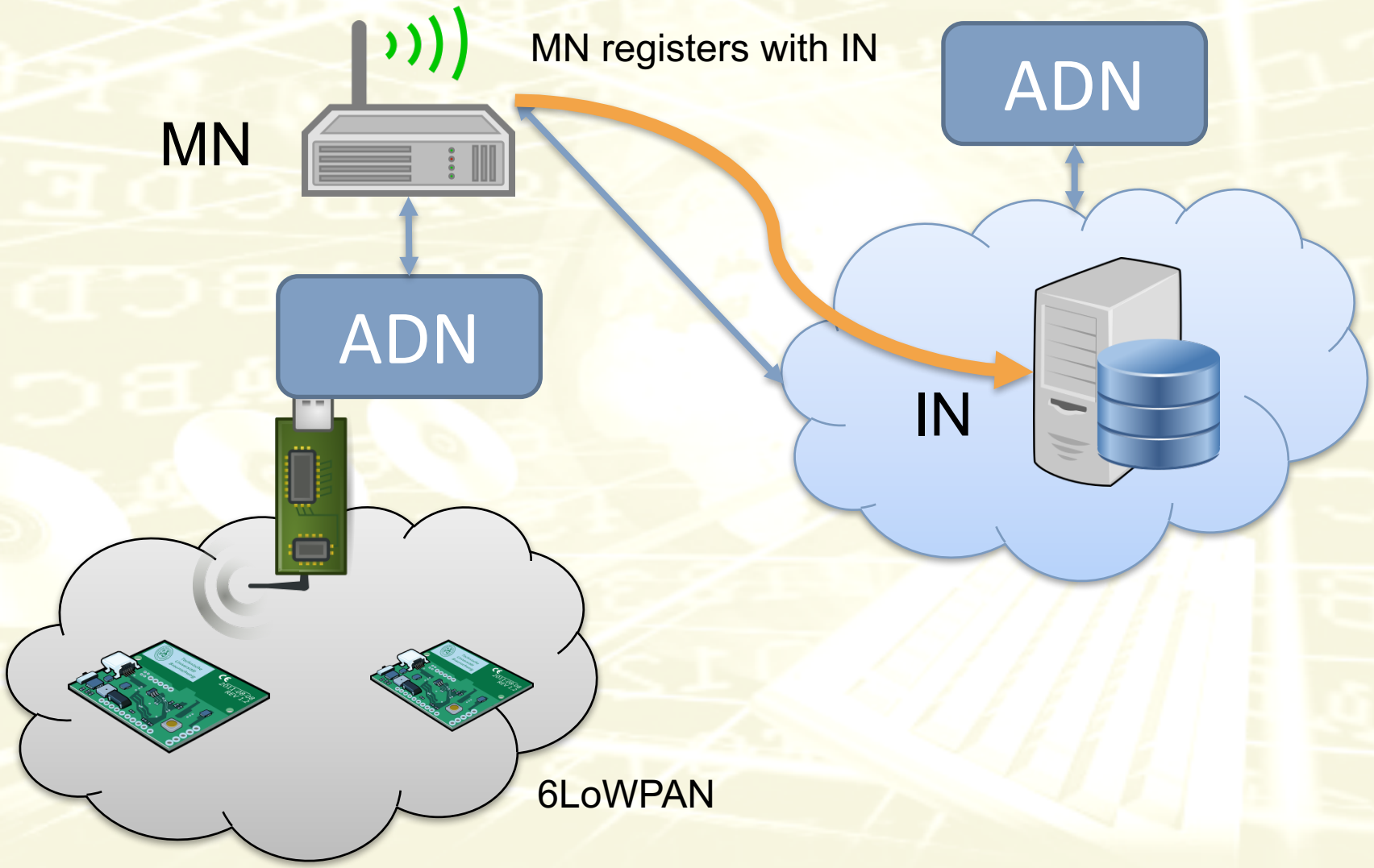
Exercise 2

- Edit the Exercise of last lesson to periodically publish new contentInstances.
- Edit the previous Exercise to create an application that:
  1. Issue a Discovery request to the IN for Container Resources hosted by the MN
  2. Start a CoAP server, in a separate Thread, on port 5685 with a resource that accept POST and replies with 2.01 CREATED
  3. Create a subscription on the URI retrieved at point 1.

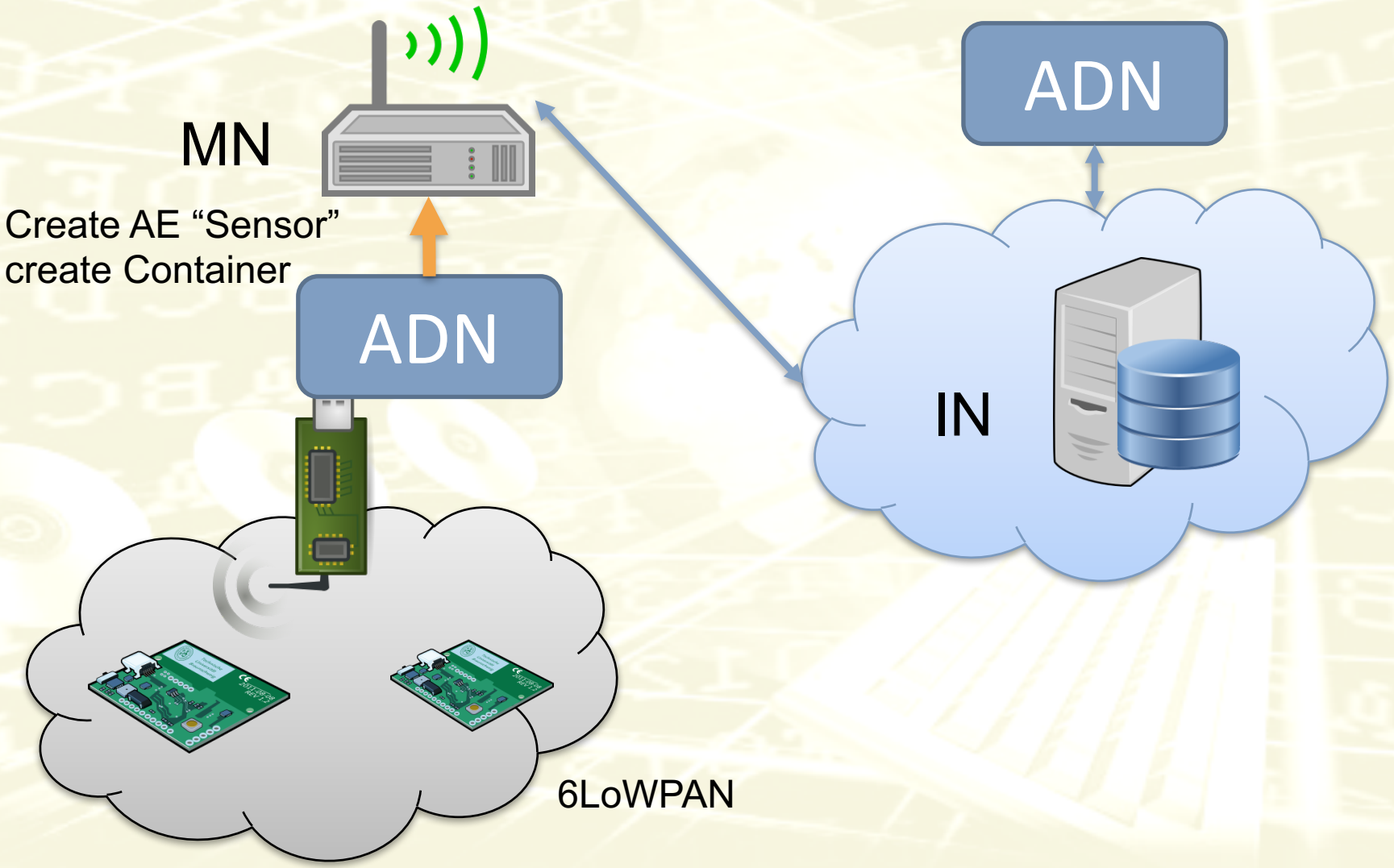
All together



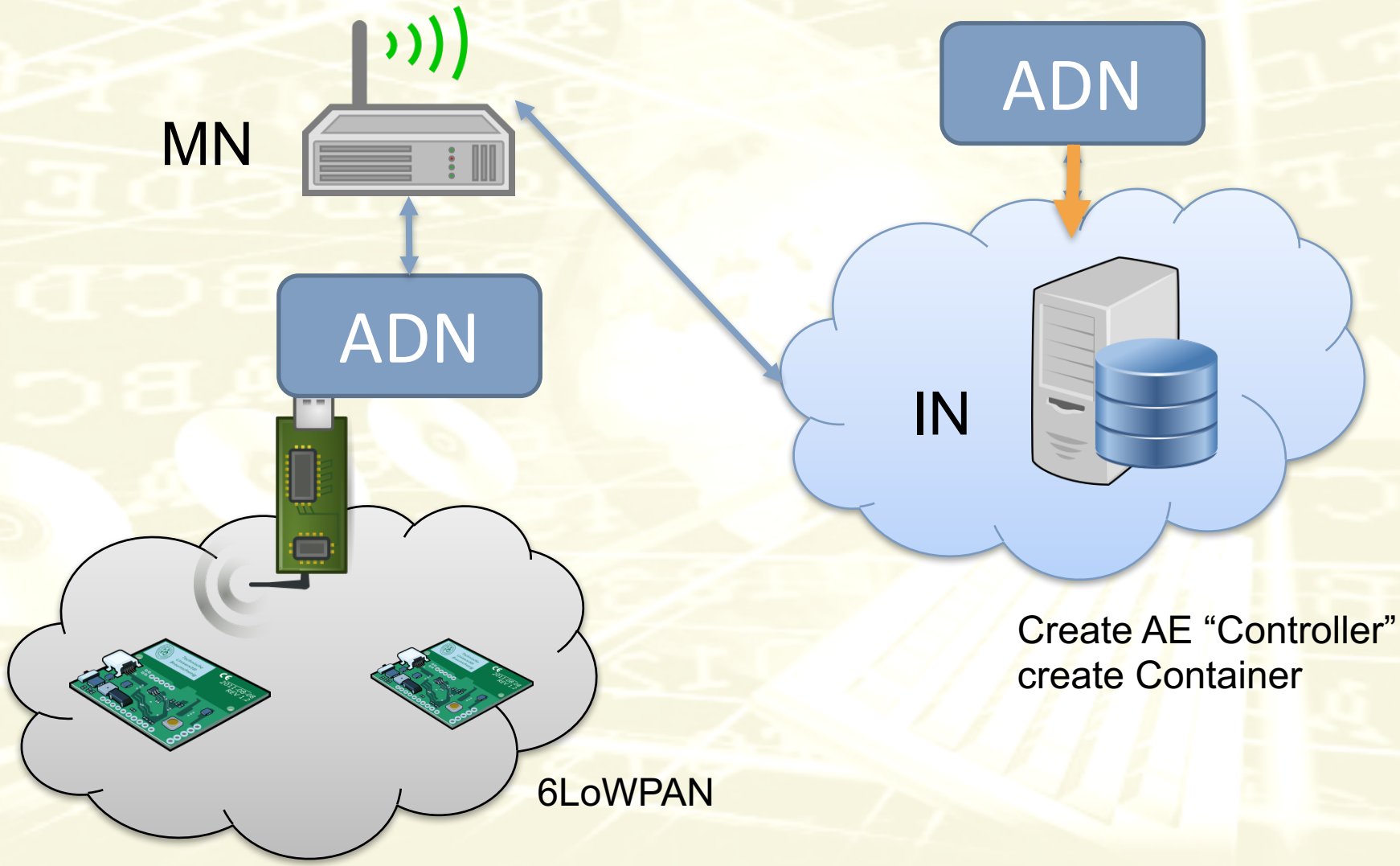
All together



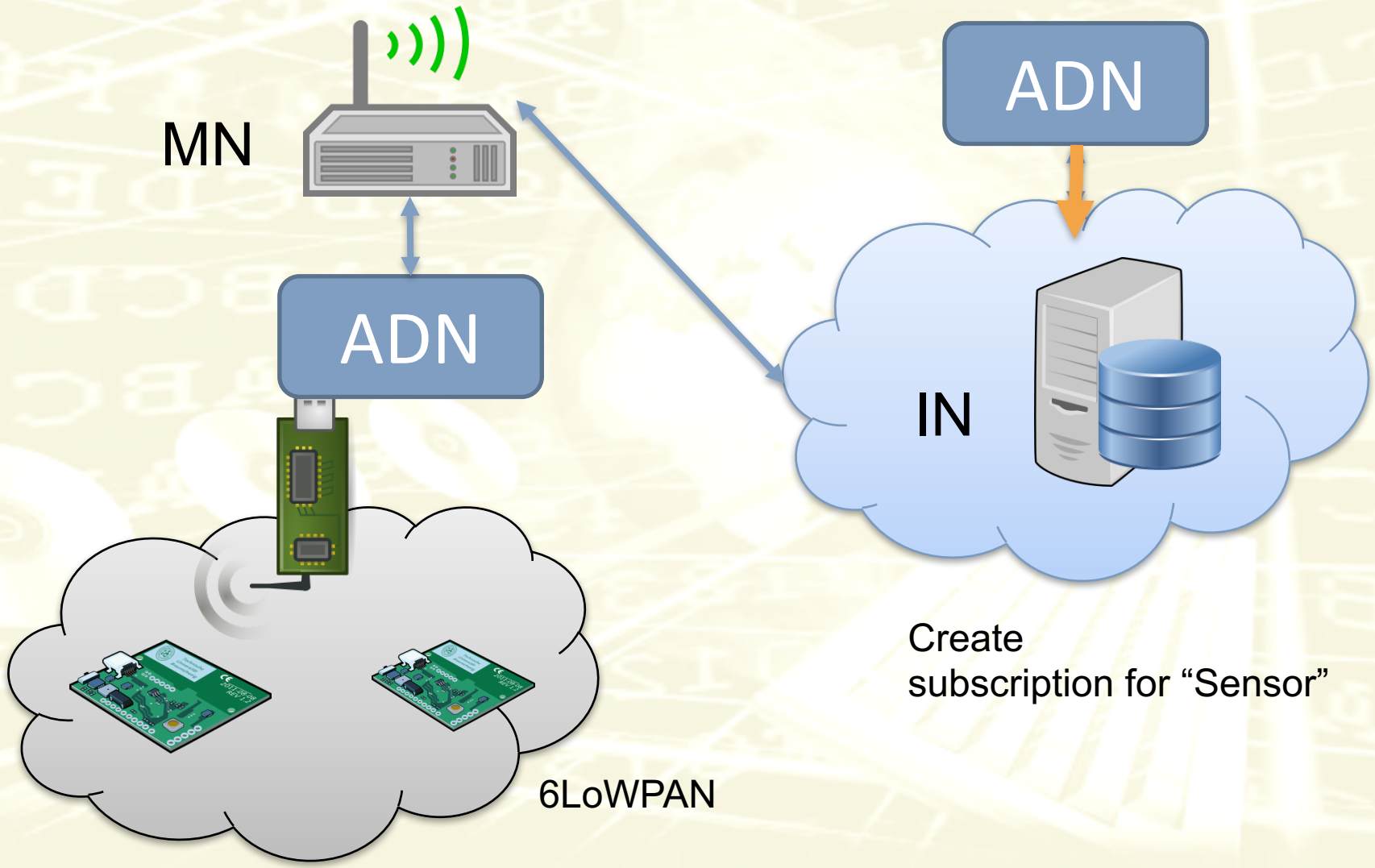
All together



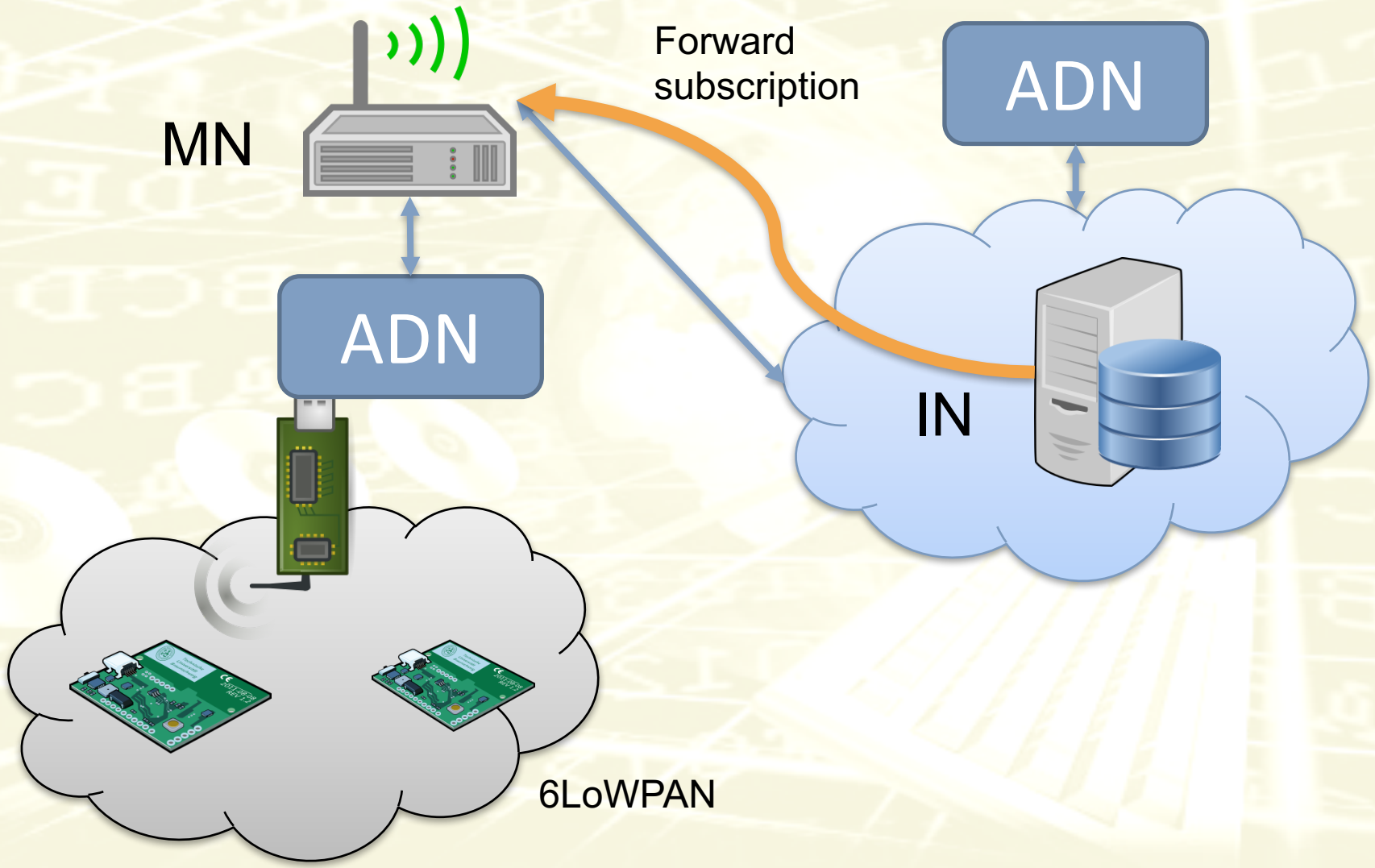
All together



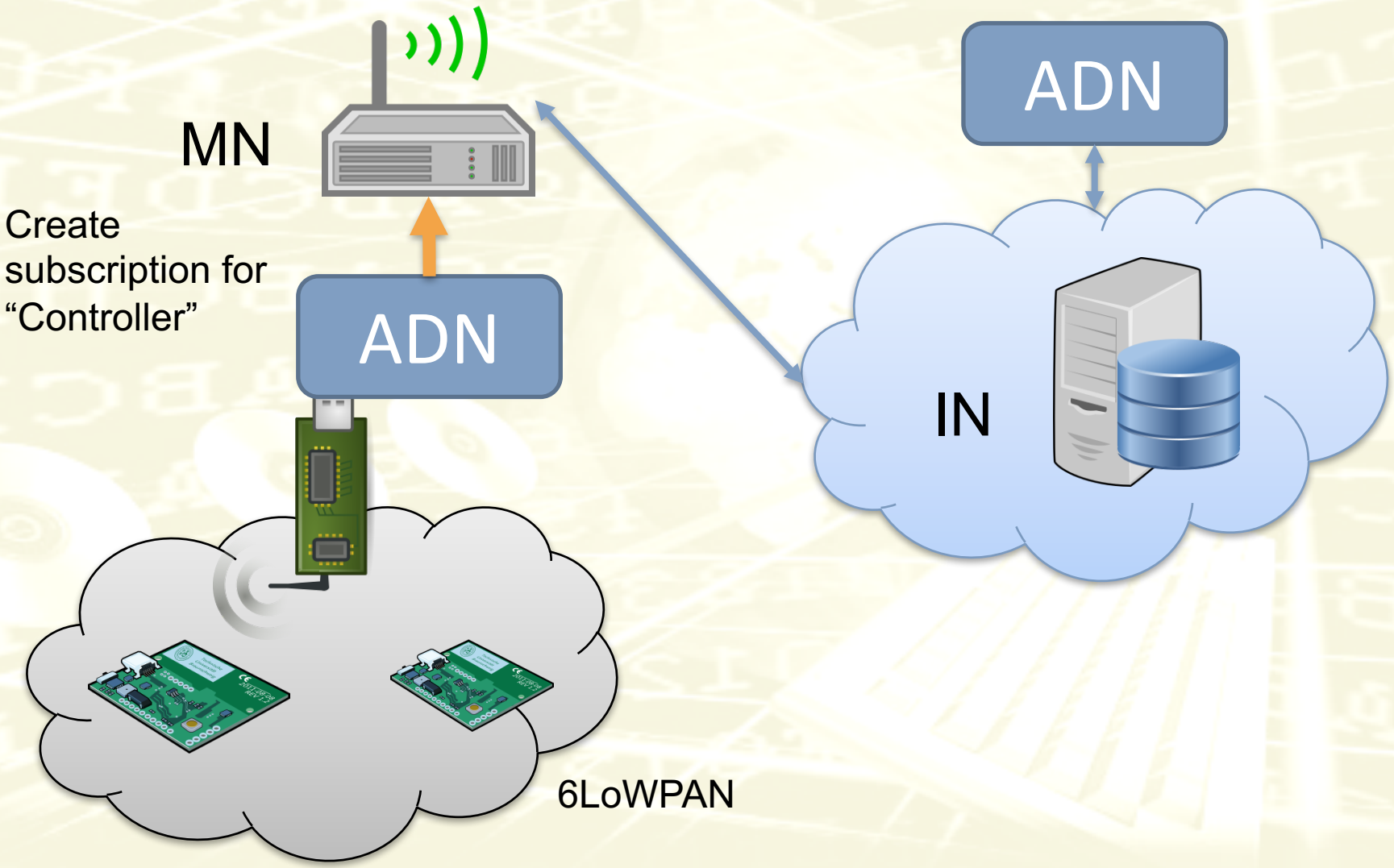
All together



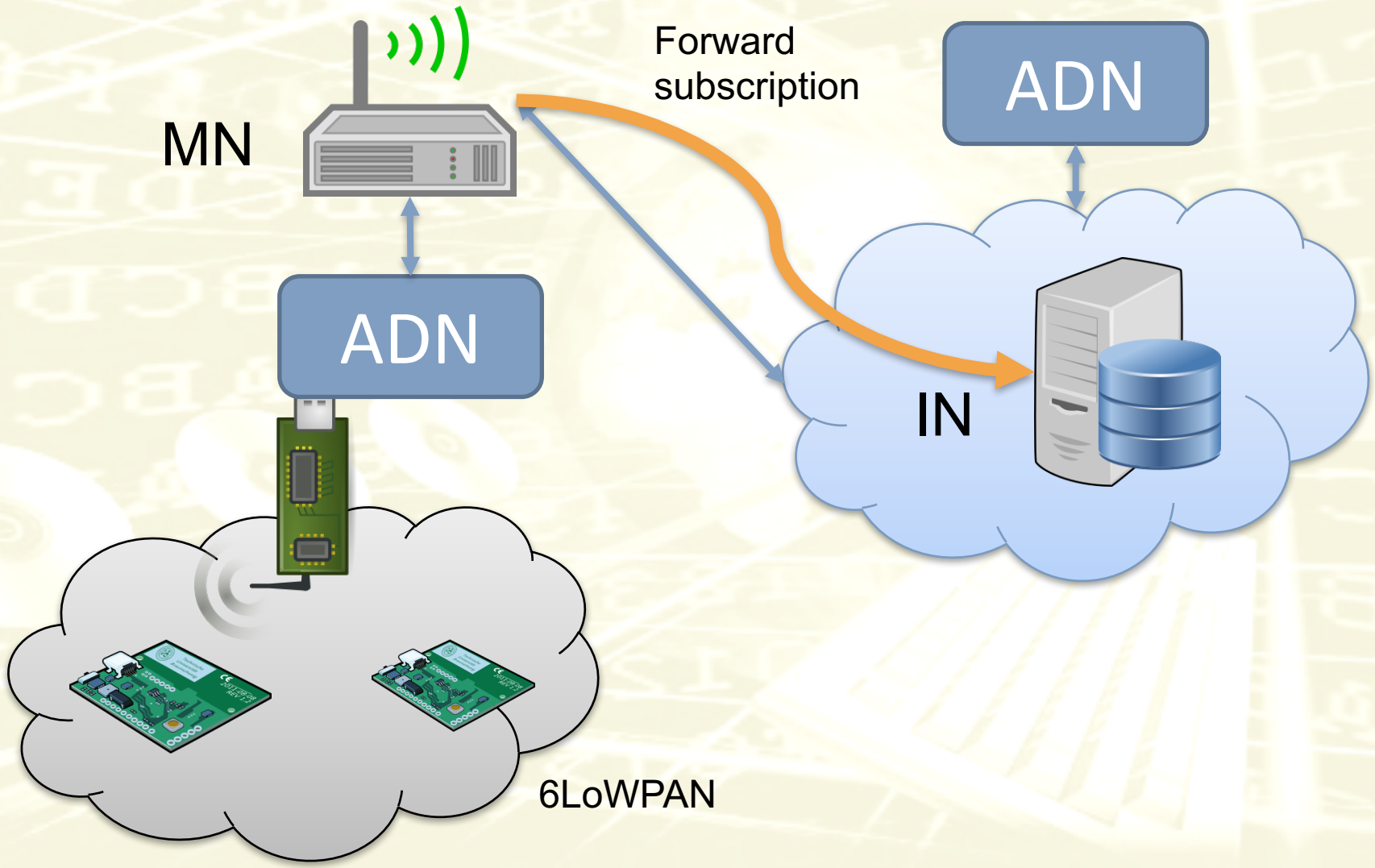
All together



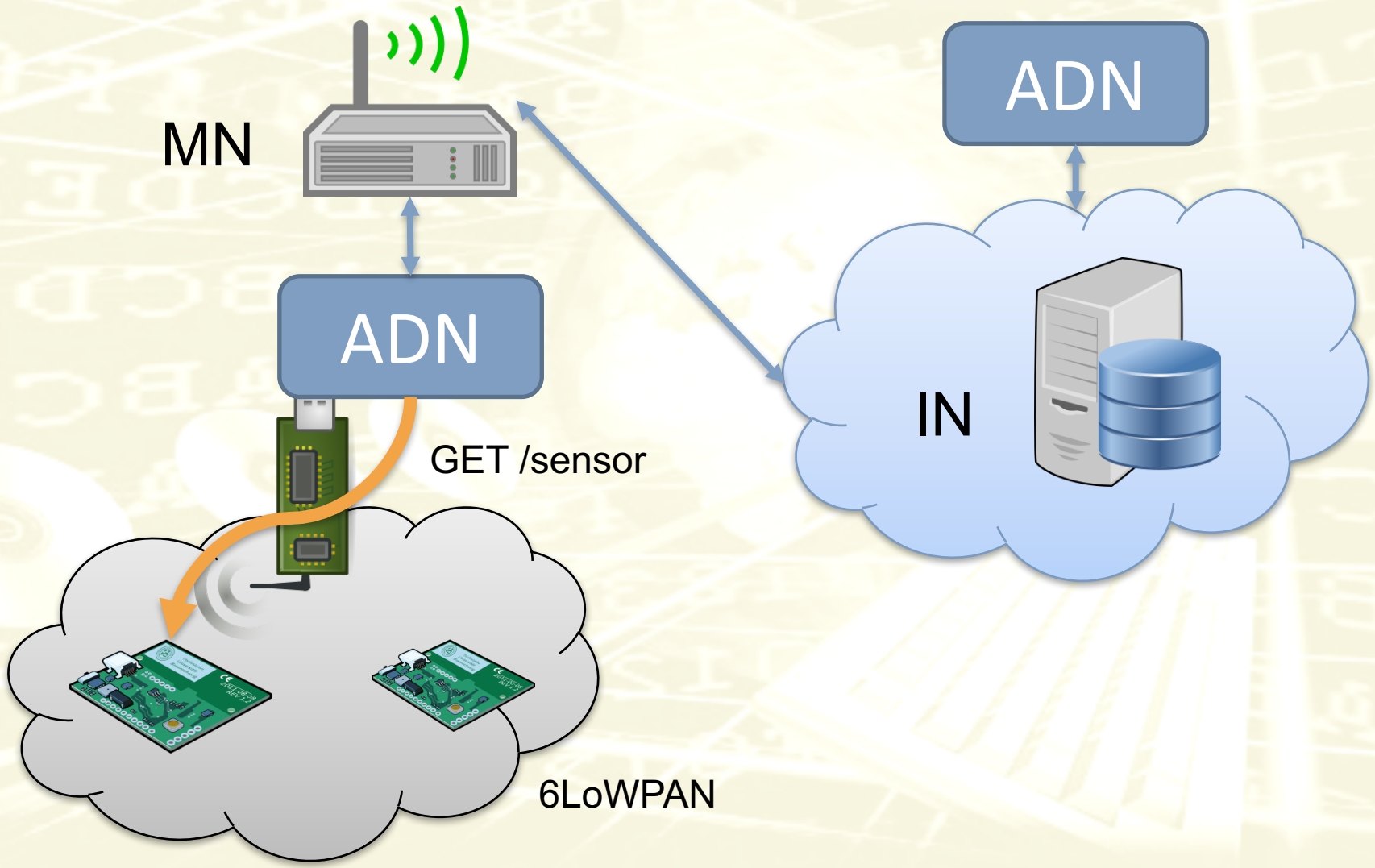
All together



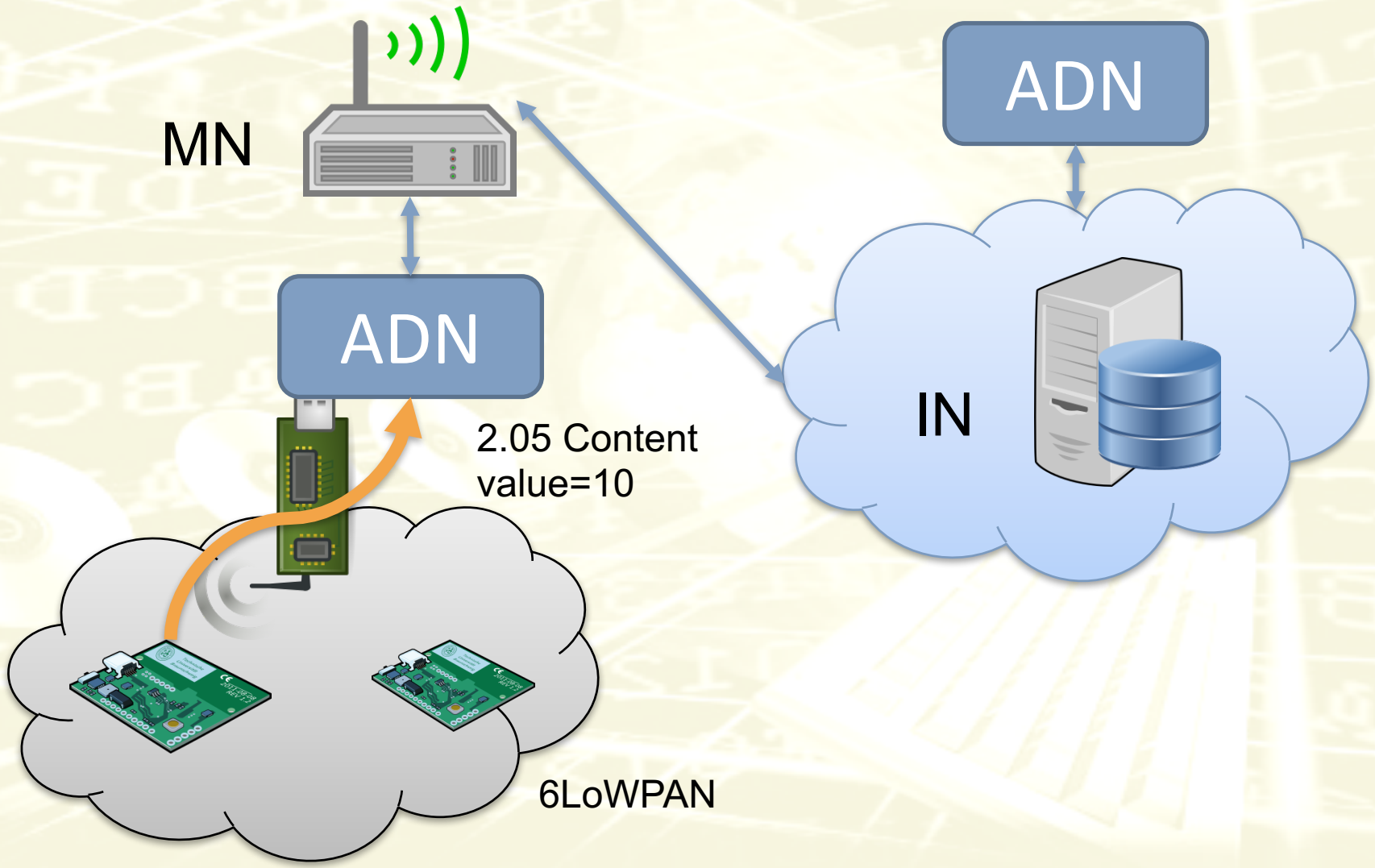
All together



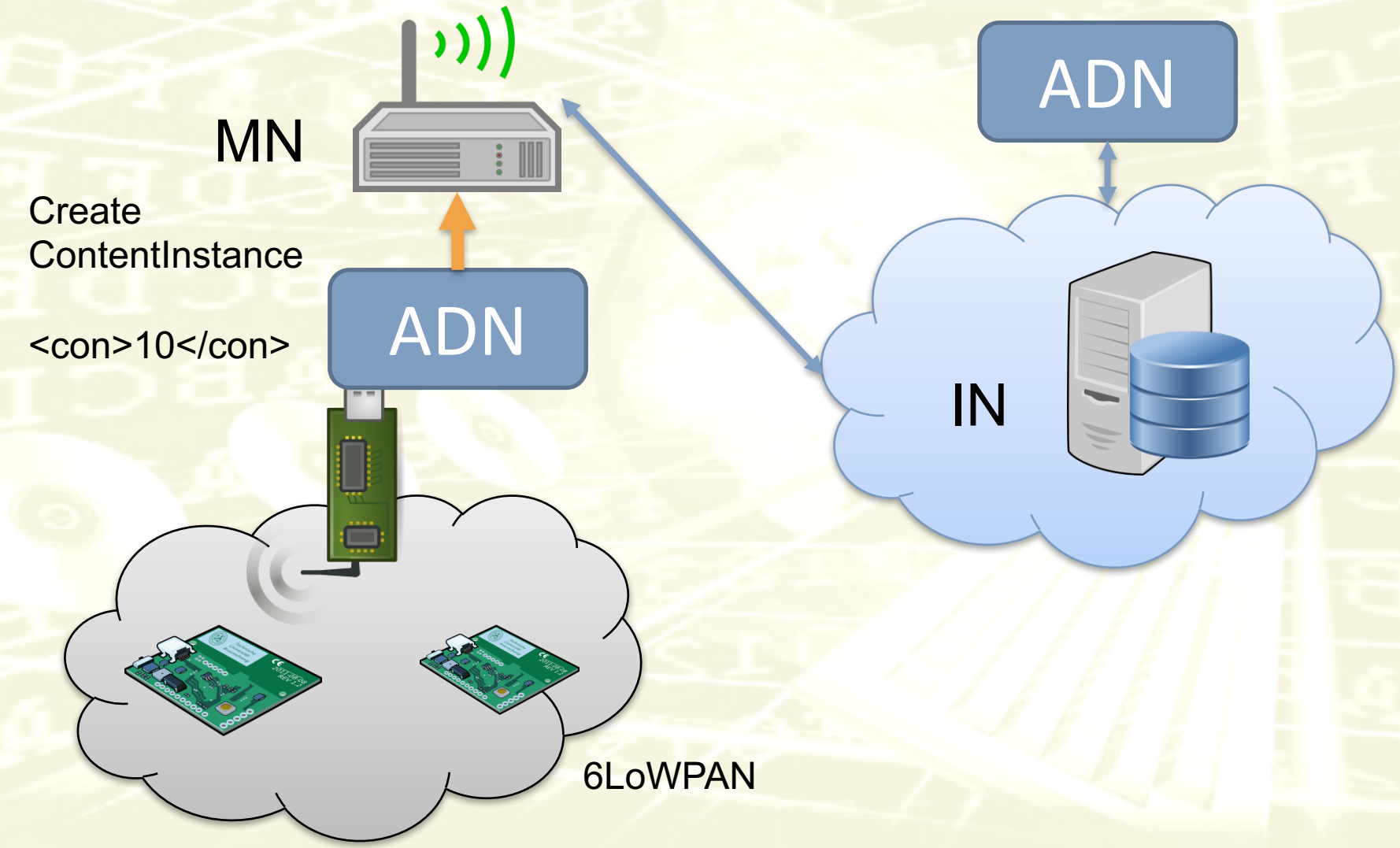
All together



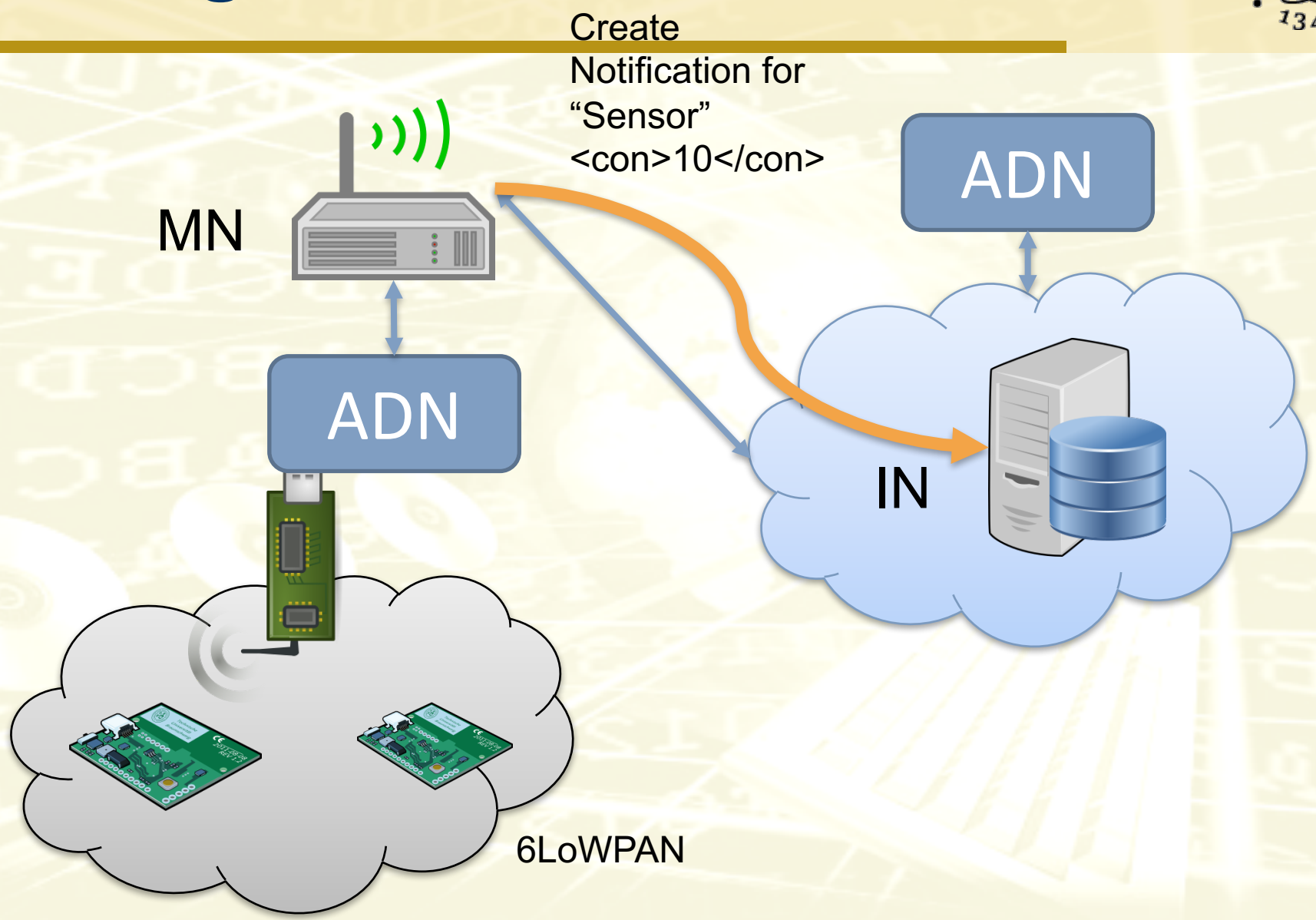
All together



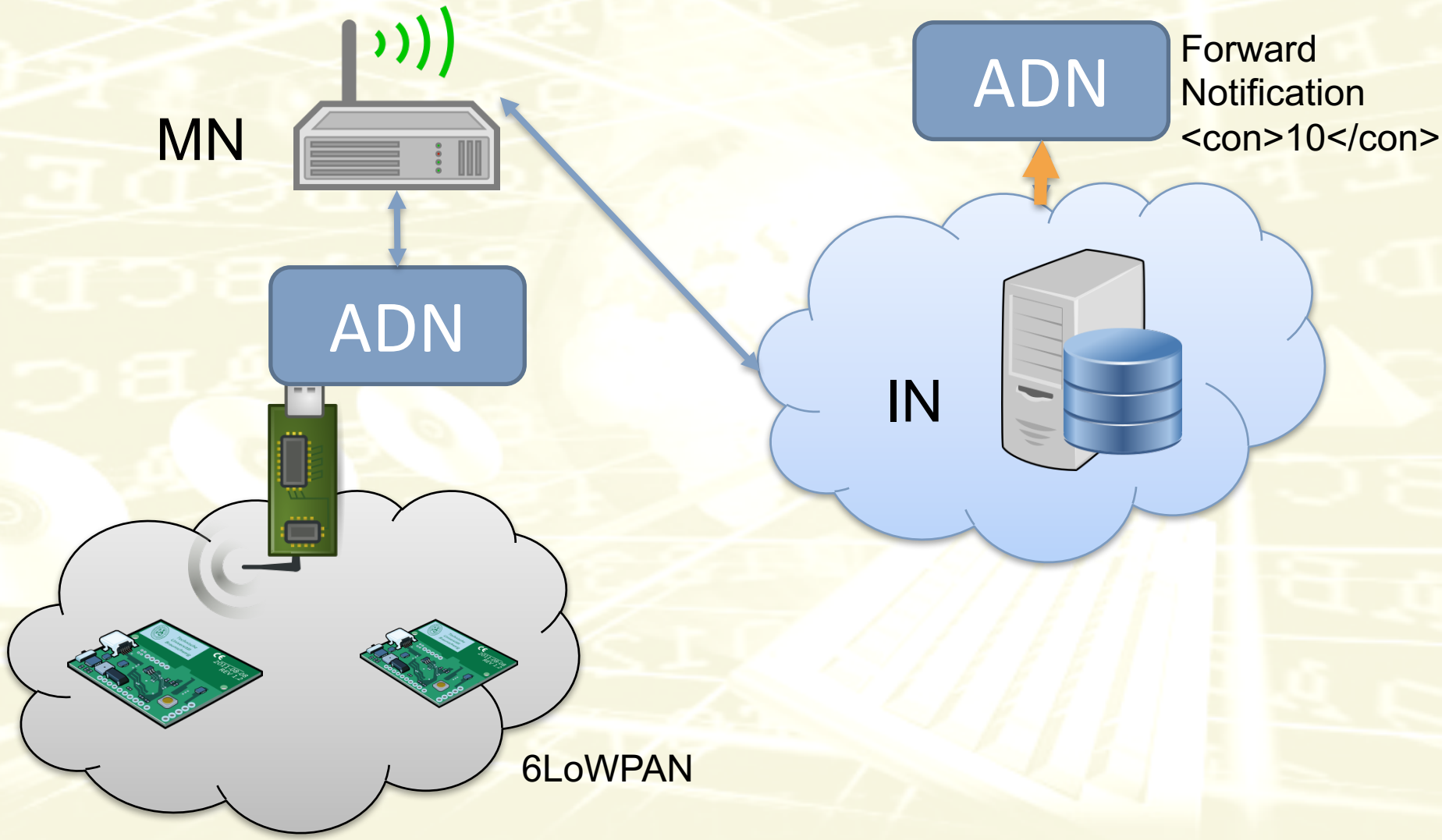
All together



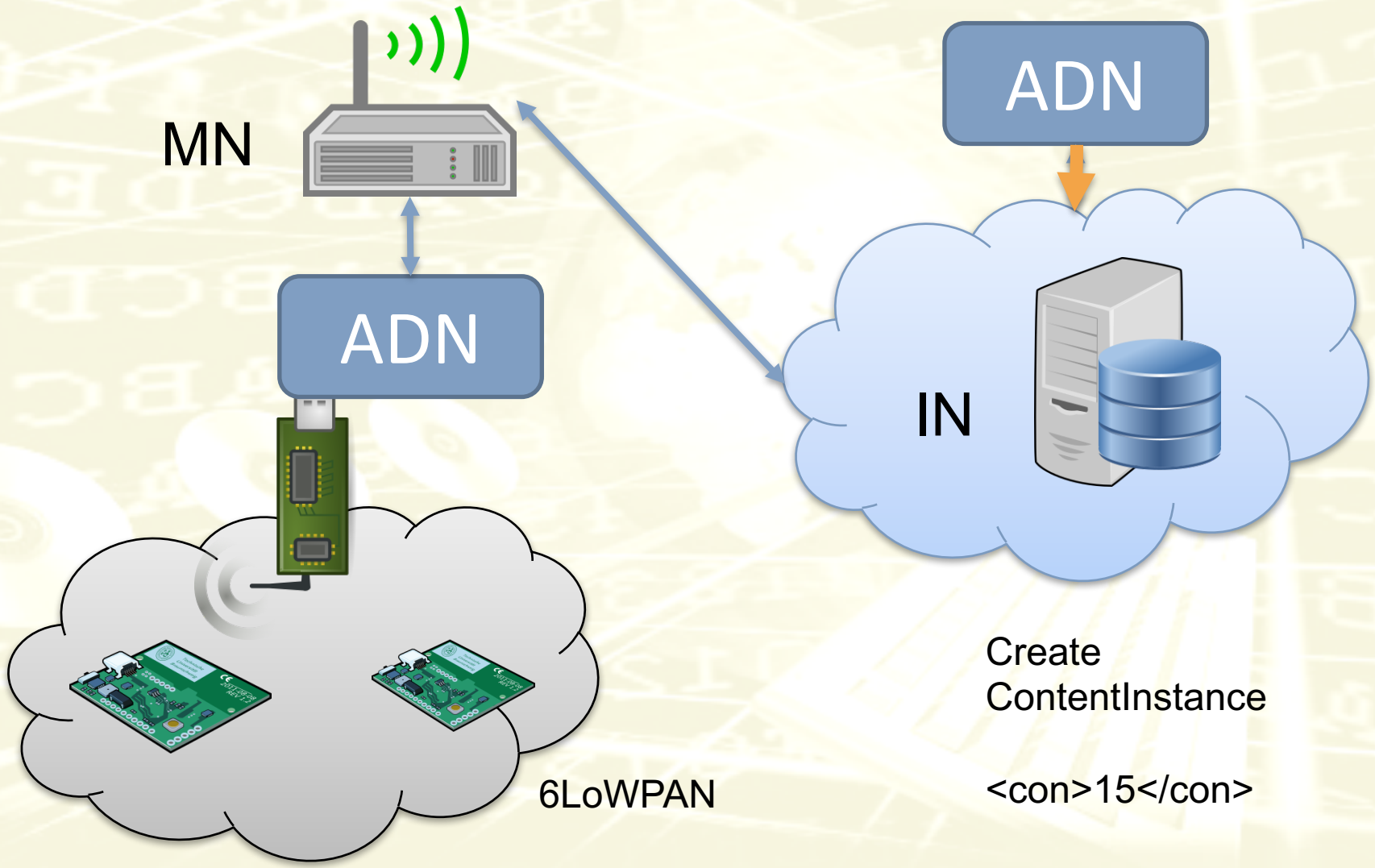
All together



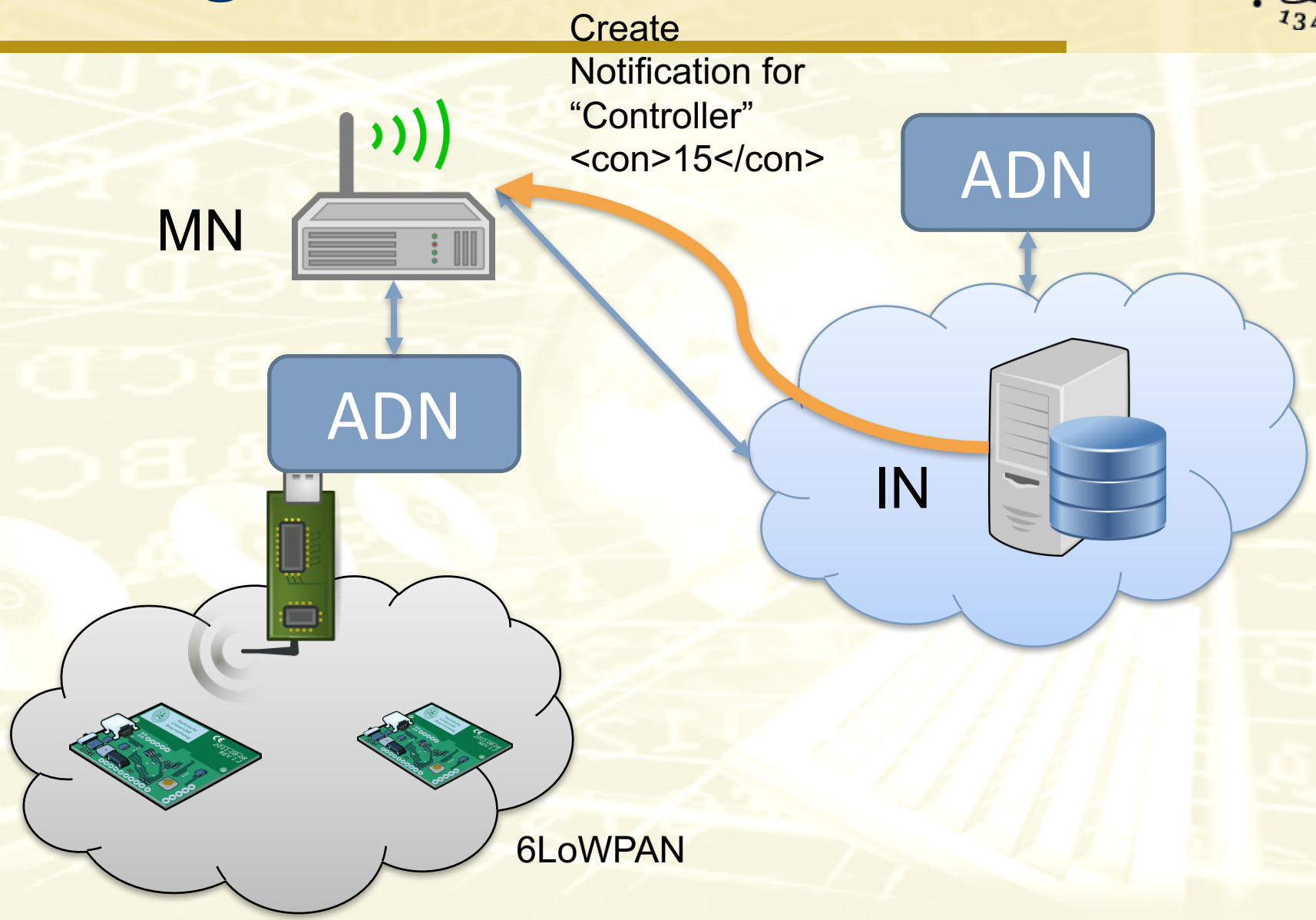
All together



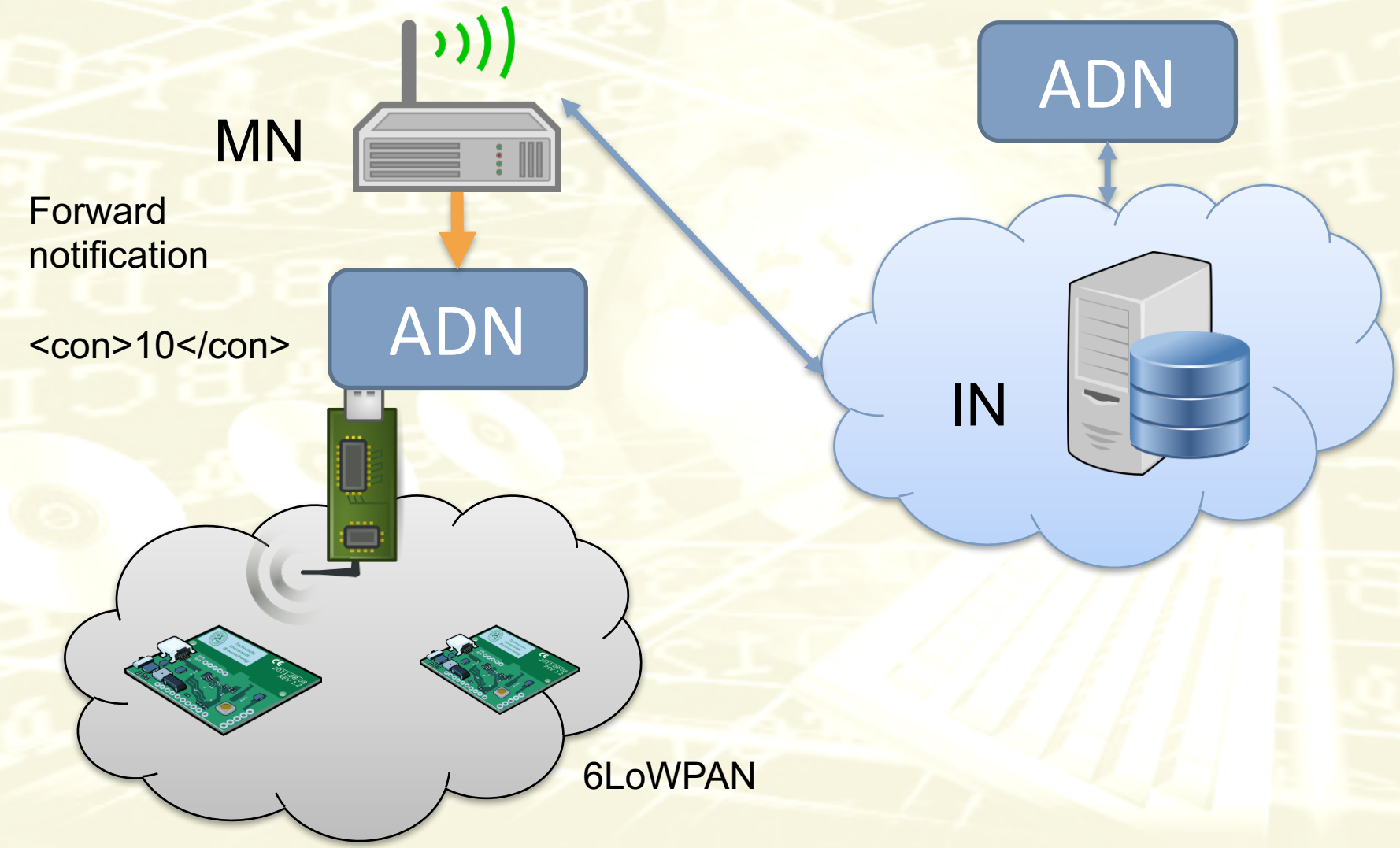
All together



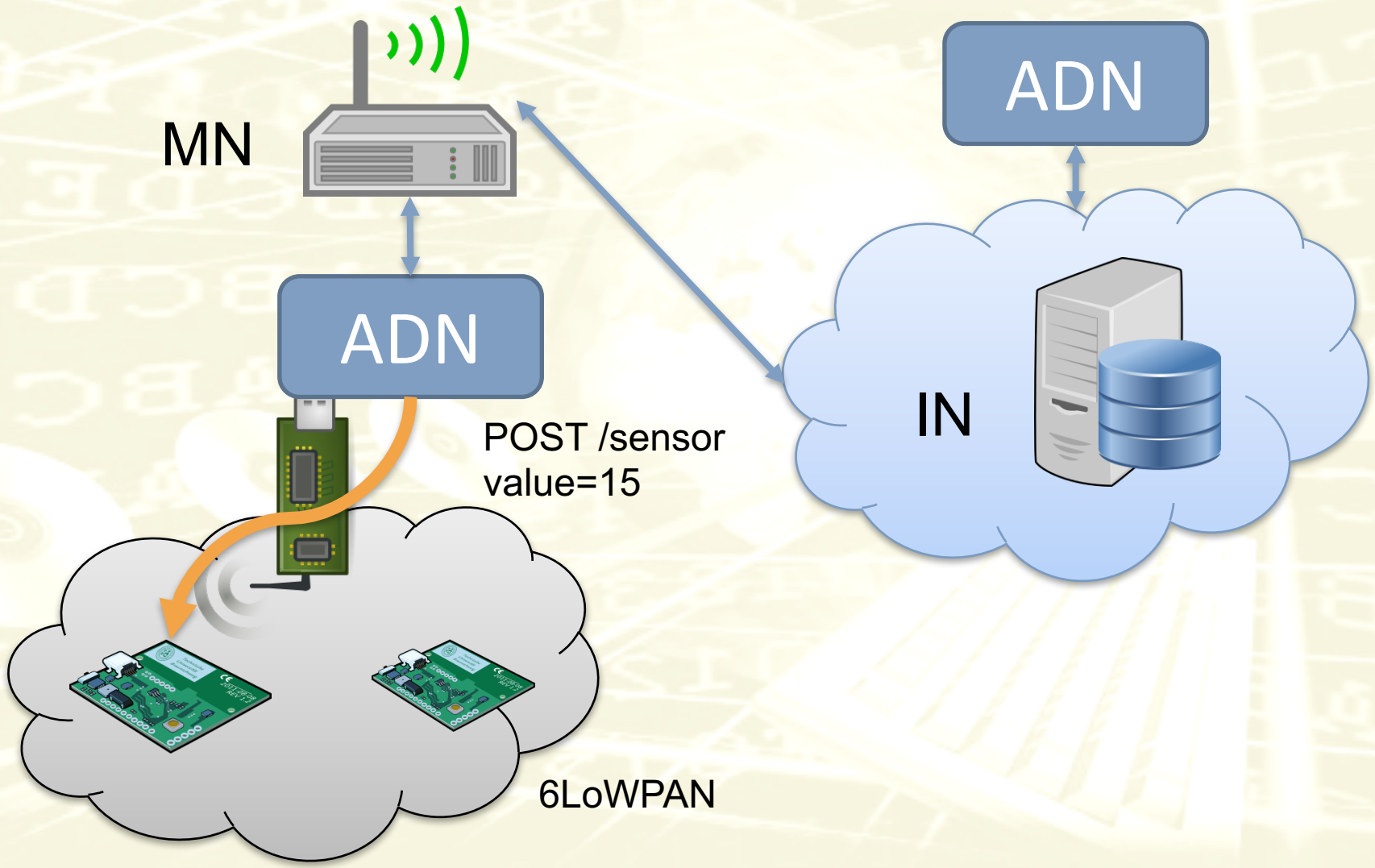
All together



All together



All together



Exercise 3

- Write a CoAP Server on Contiki with an internal integer value that can be retrieved with a GET and set with a POST.
- Deploy everything in Cooja. Remember the border router.

Exercise 3 (2)

- Write an ADN that:
  - Creates AE on the MN (Sensor)
  - Create a Container on the MN
  - Read data from the CoAP Server every minutes and publish the values as contentInstance.
- Write an AND that:
  - Connects to the IN to subscribe for Container resources on the MN

Exercise 4

- Modify the IN AND to:
  - Creates AE on the IN “Controller”
  - Create a Container on the IN
  - Create contentInstances on the IN each one with the value set equal to the value obtained by notifications incremented by 1.
- Modify the MN AND to:
  - Subscribe for Container resources hosted by the IN
  - Updates values on sensors through POST requests where the value is equal to the value of received notifications