

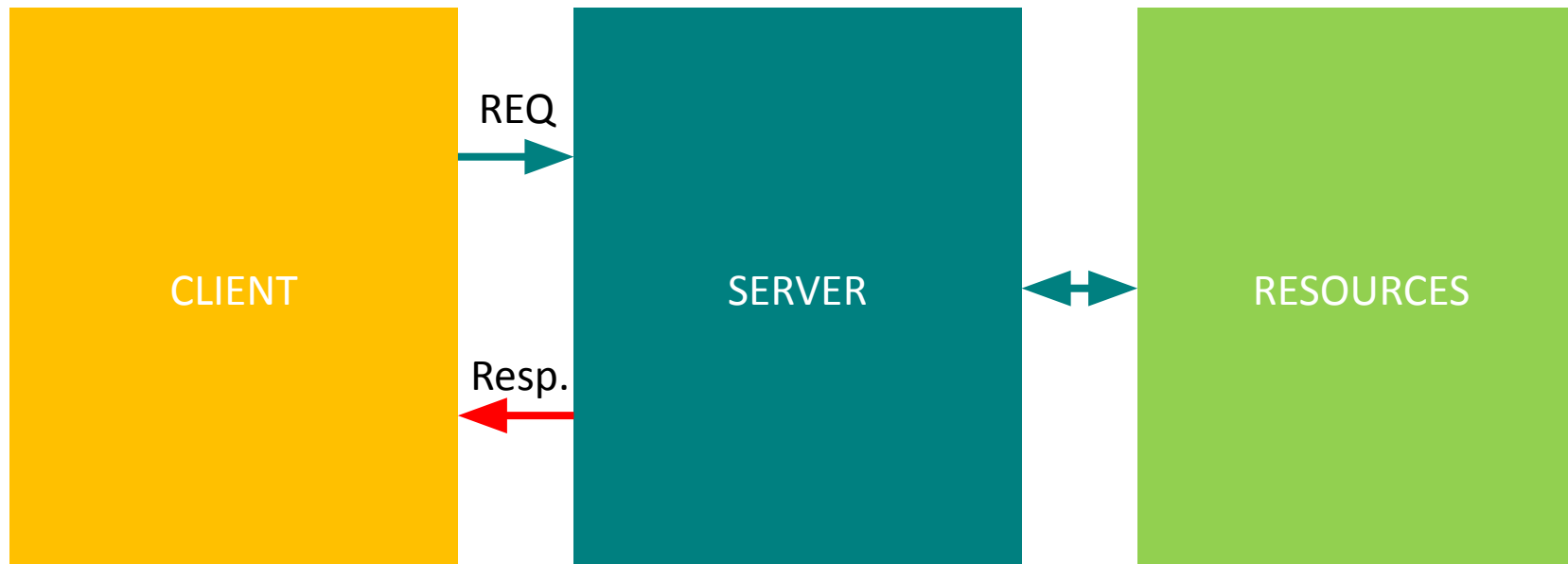
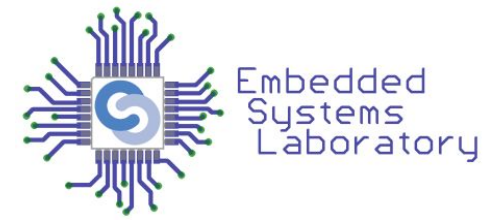
---

# Internet of Things

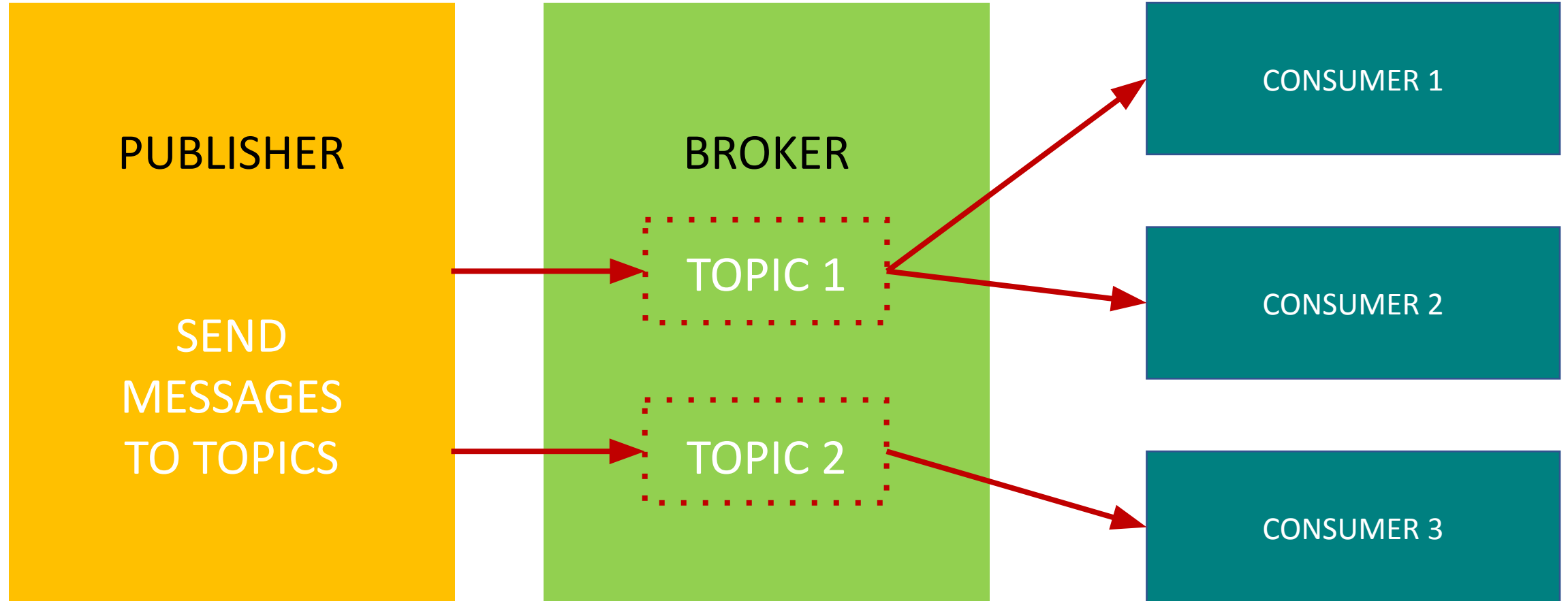
## Lecture 5 - Communication Models

---

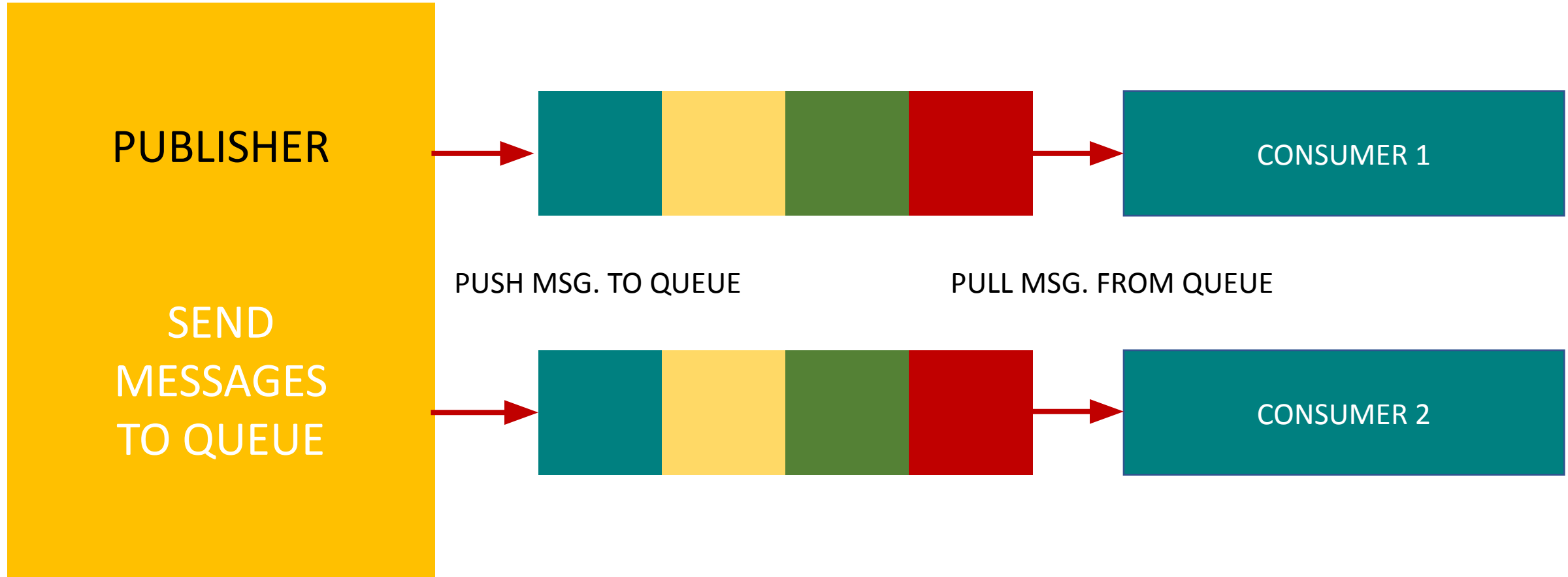
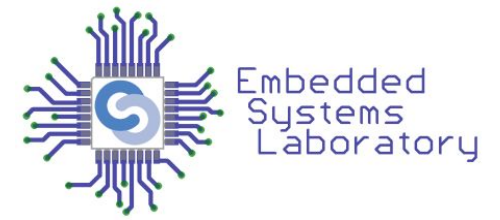
# Client-Server Model



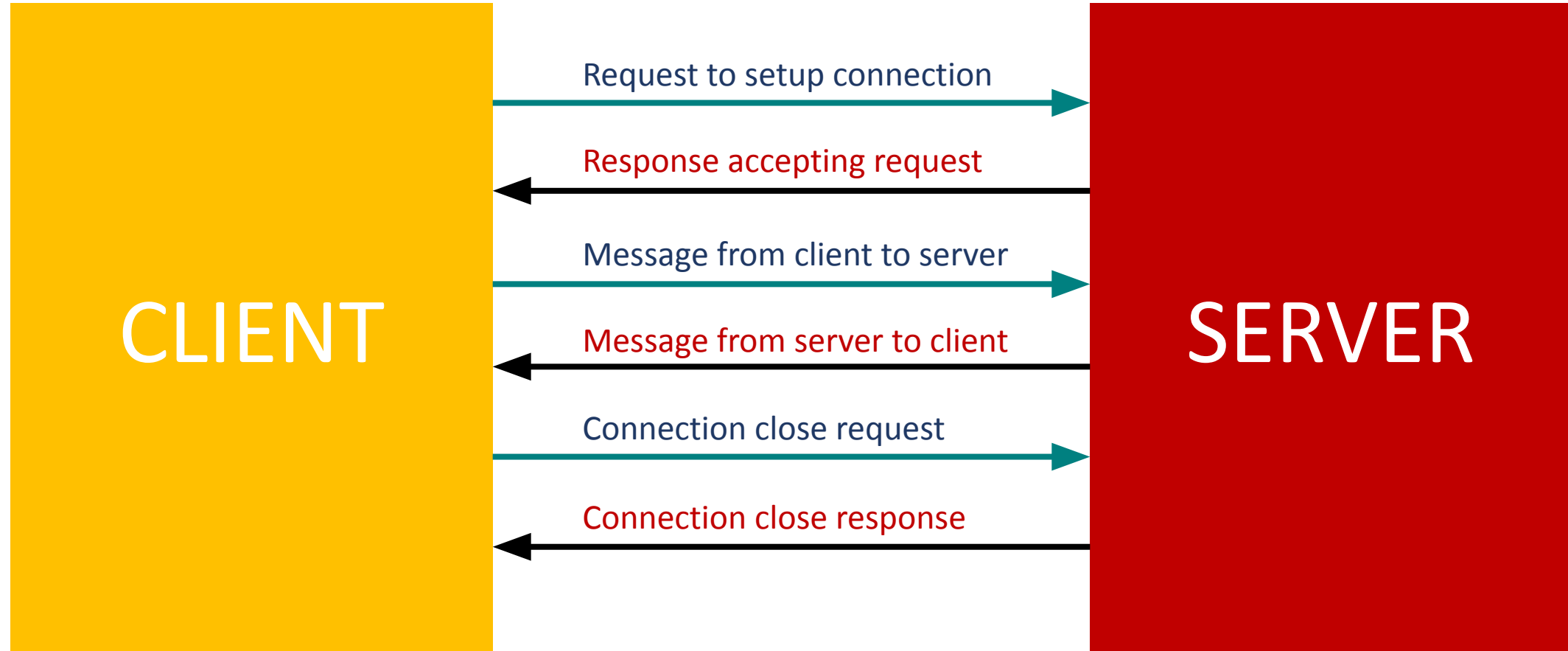
# Publish-Subscribe Model



# Push-Pull Model

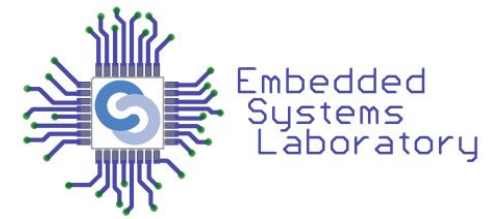


# Exclusive Pair Model



# REST Communication APIs

---

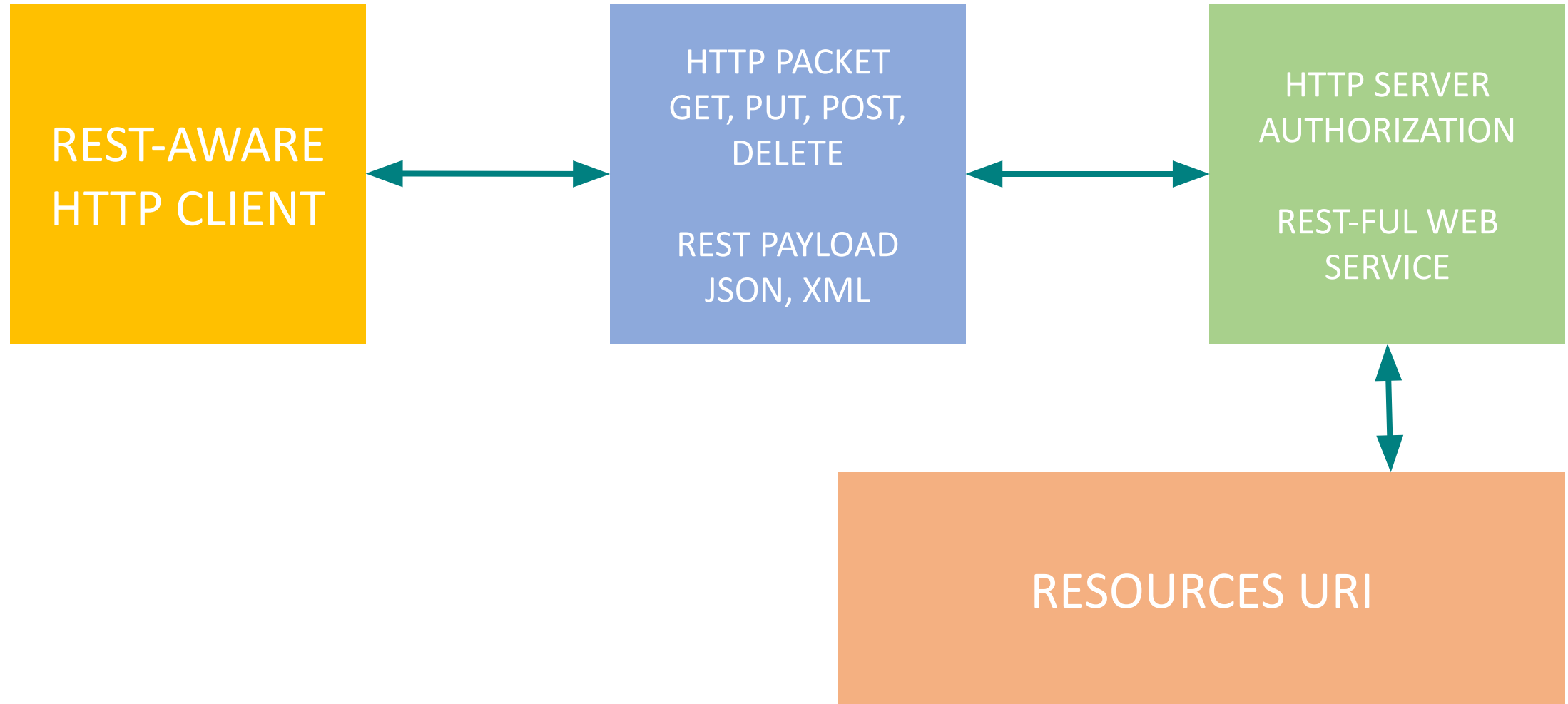
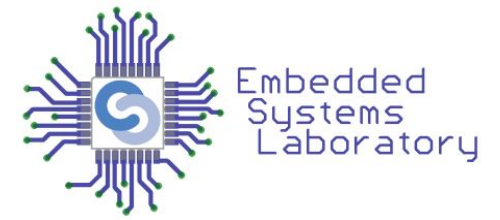


- REpresentational State Transfer
- Client Server considerations
  - Client does not care about how data is stored at the server
  - Server does not care about the user interface at the client
- Stateless
  - The client request should contain all the information necessary to respond to a request
- Cache-able
- Layered
  - The client does not need to know if it's communicating with the actual server, a proxy, or any other intermediary

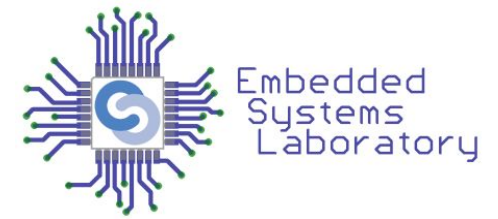
Source:

<https://www.sitepoint.com/developers-rest-api/>

# REST-Based APIs



# RESTful Web Service Request



1. An Endpoint URL `https://mydomain/user/123?format=json`

2. The HTTP method

HTTP method	CRUD	Action
GET	read	returns requested data
POST	create	creates a new record
PUT or PATCH	update	updates an existing record
DELETE	delete	deletes an existing record

3. HTTP headers

4. Body Data

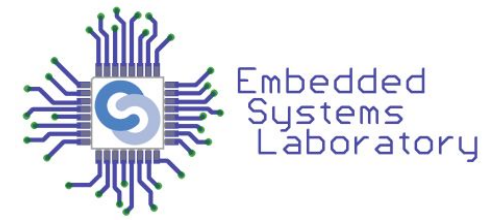
Source:

<https://www.sitepoint.com/developers-rest-api/>



# Examples

---



- a GET request to `/user/` returns a list of registered users on a system
- a POST request to `/user/123` creates a user with the ID 123 using the body data
- a PUT request to `/user/123` updates user 123 with the body data
- a GET request to `/user/123` returns the details of user 123
- a DELETE request to `/user/123` deletes user 123

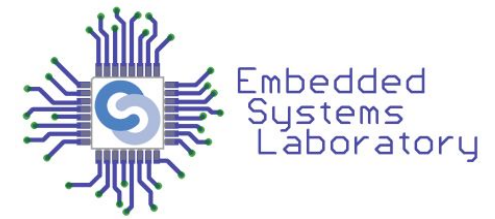
---

Source:

<https://www.sitepoint.com/developers-rest-api/>

# RESTful Web Service Reply

---

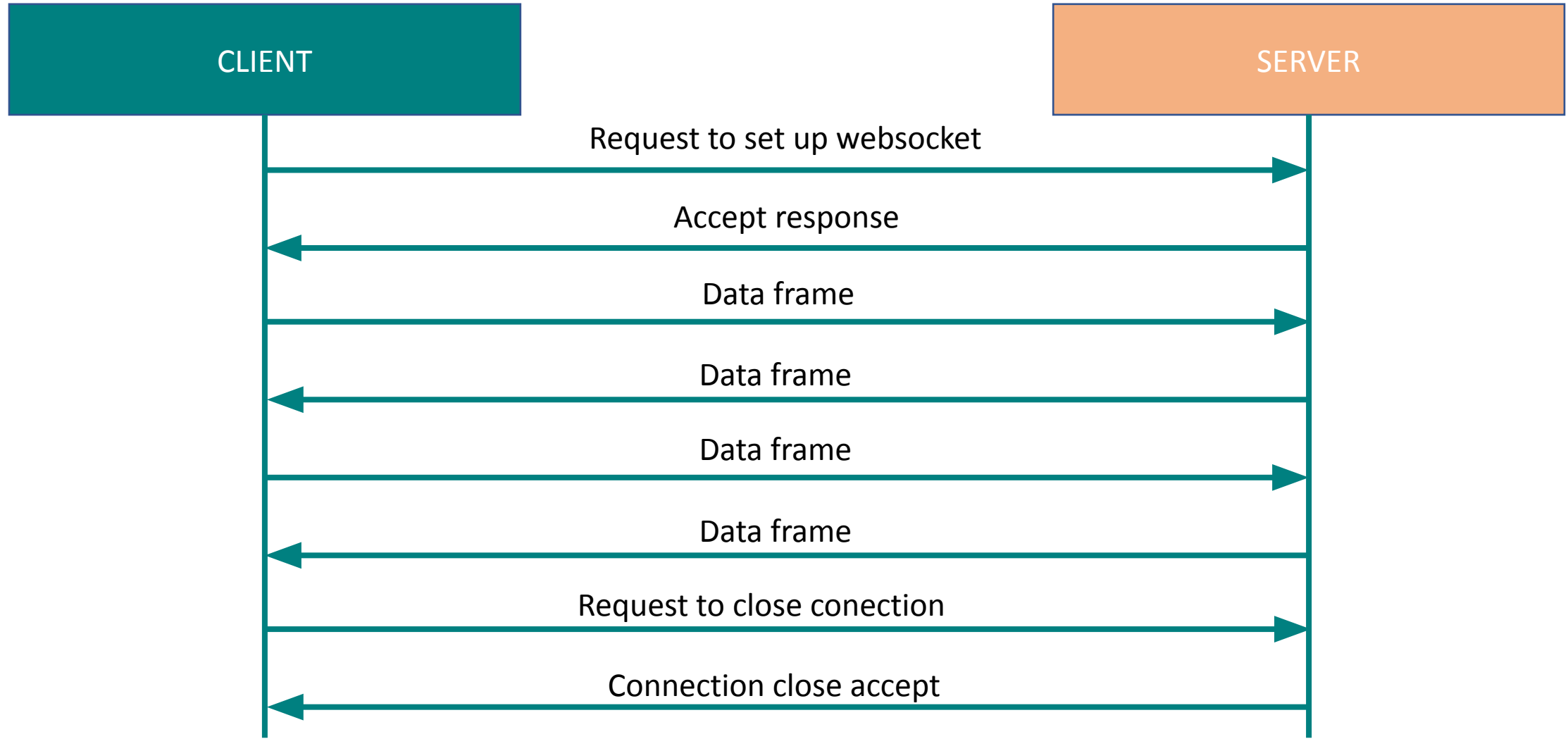
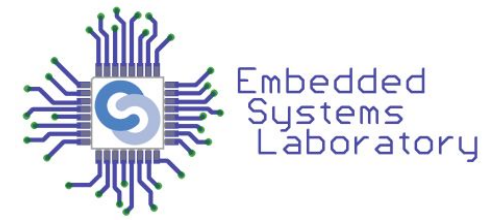


- Response payload:
  - data, HTML, image, audio file, etc.
  - JSON, XML, CVS, simple strings, etc.
- HTTP status code set in the header
  - 200 OK
  - 201 Created
  - Errors: 400 Bad Request, 404 Not Found, 401 Unauthorized

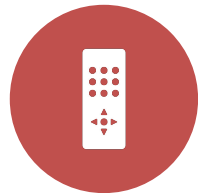
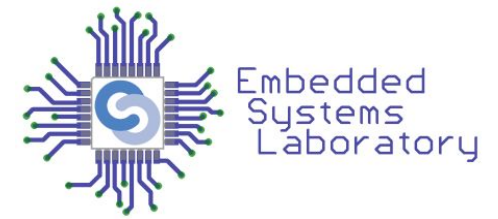
Source:

<https://www.sitepoint.com/developers-rest-api/>

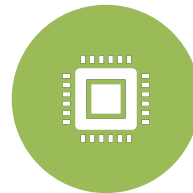
# WebSocket Based Communication



# IoT Levels & Deployment Templates



**Device:** An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.

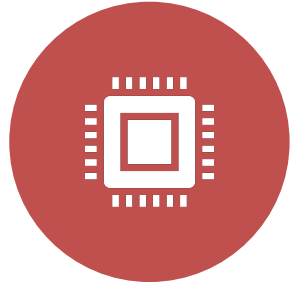
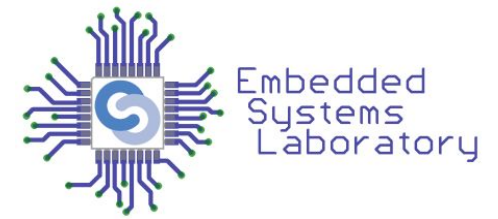


**Resource:** Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.



**Controller Service:** Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

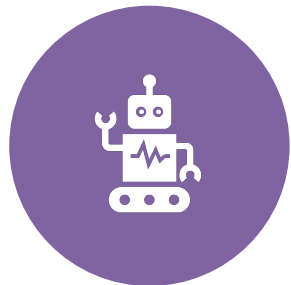
# IoT Levels & Deployment Templates



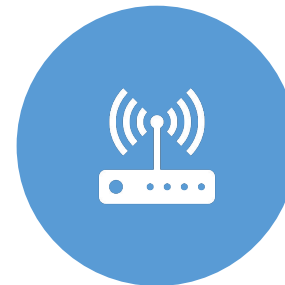
**Database:** Database can be either local or in the cloud and stores the data generated by the IoT device.



**Web Service:** Web services serve as a link between the IoT device, application, database and analysis components. Web service can be either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service).



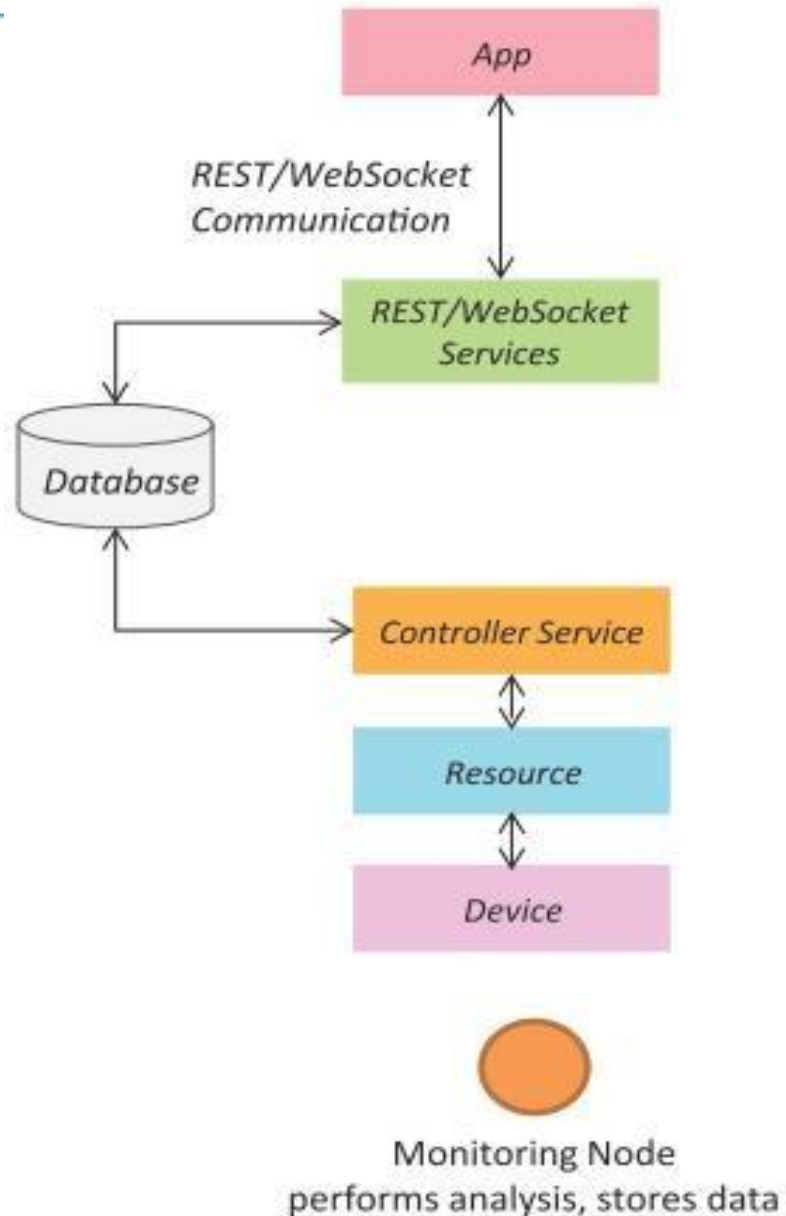
**Analysis Component:** The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.



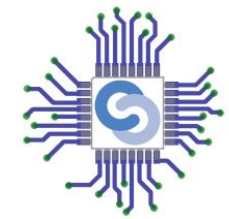
**Application:** IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view the processed data.

# IoT Level-1

Local



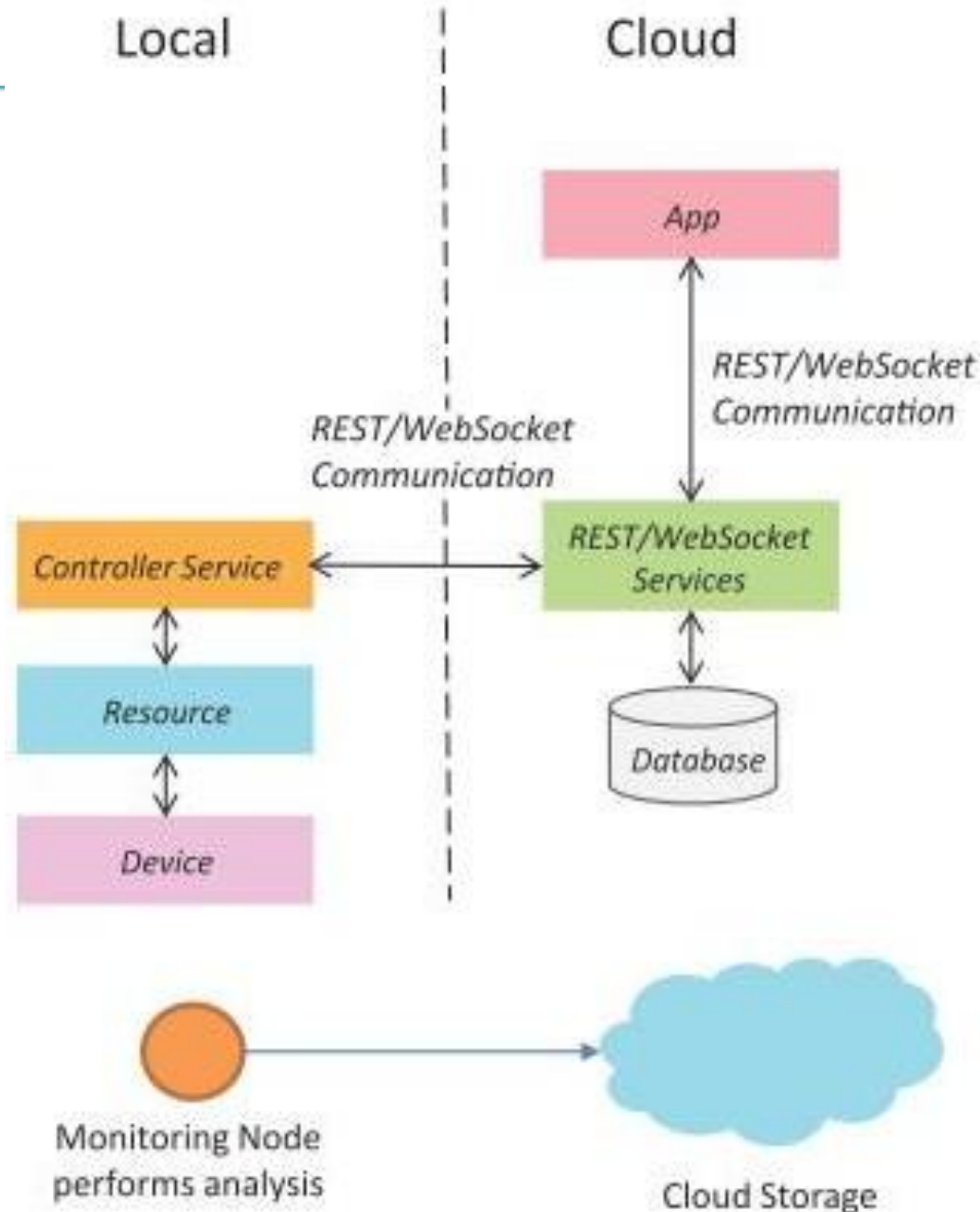
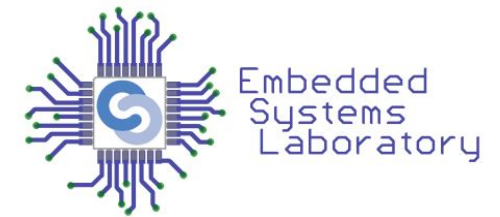
Cloud



Embedded  
Systems  
Laboratory

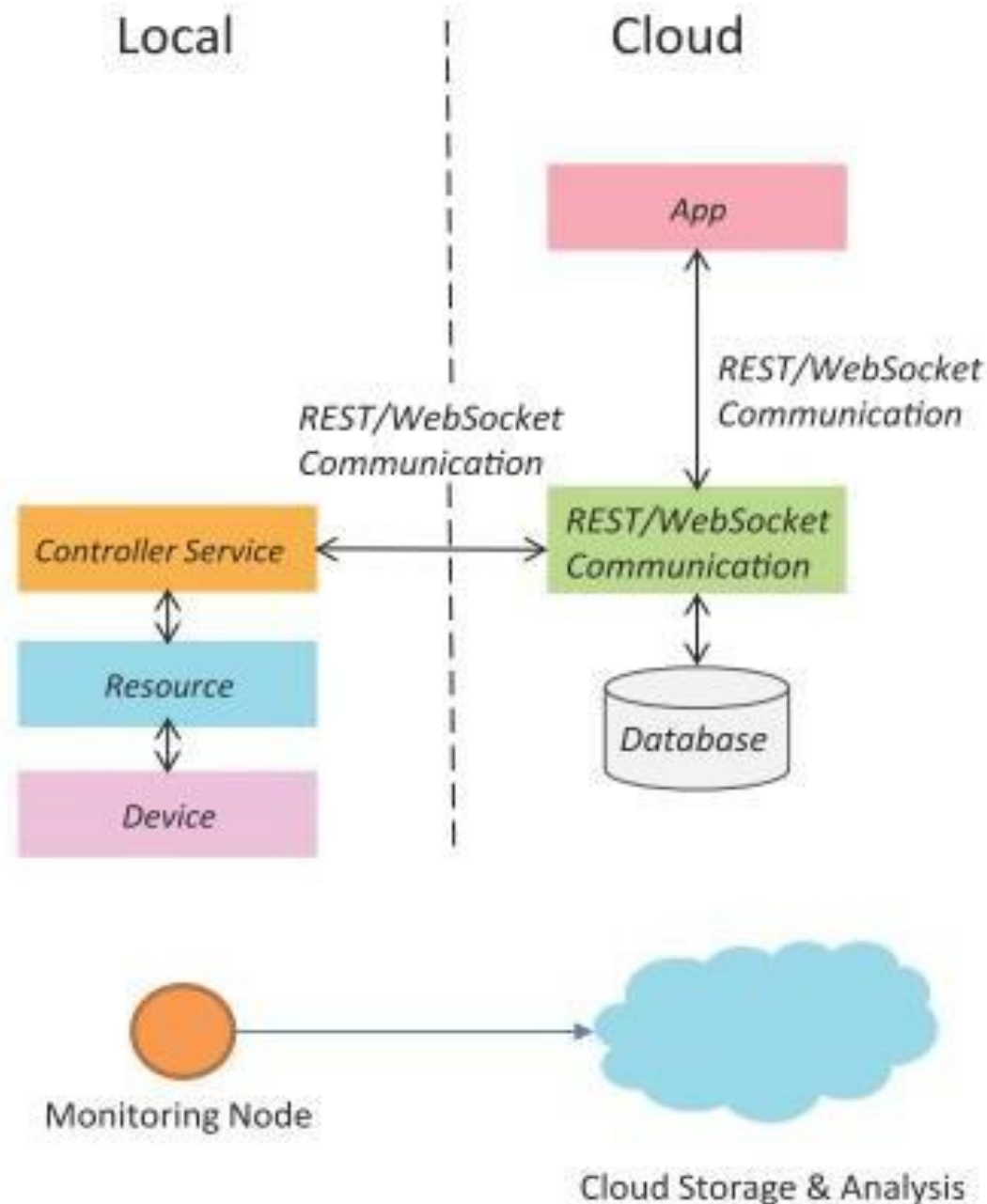
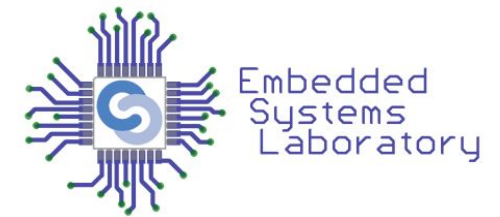
Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

# IoT Level-2



Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

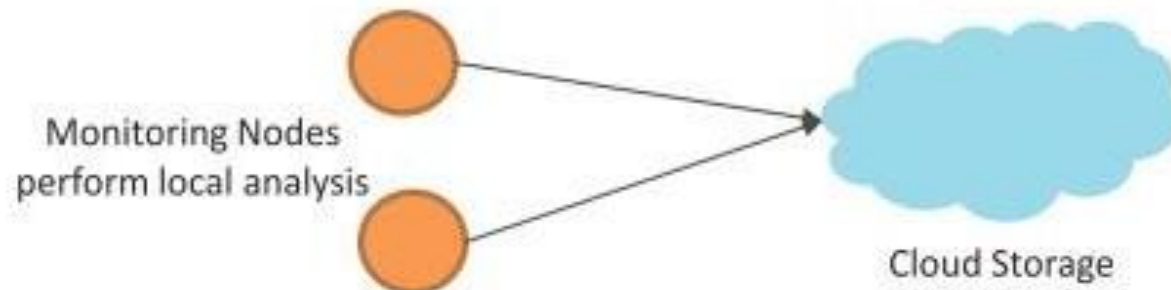
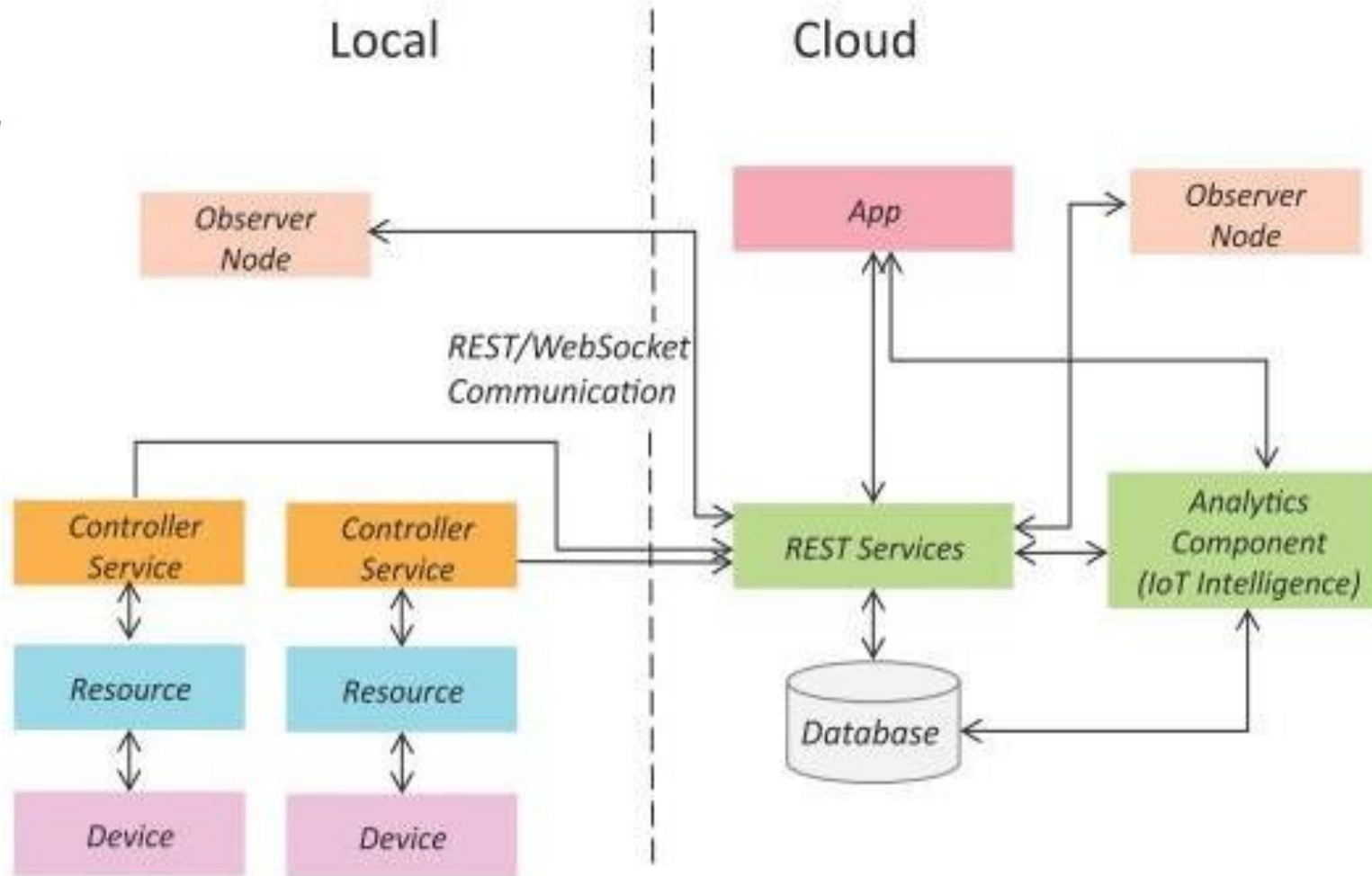
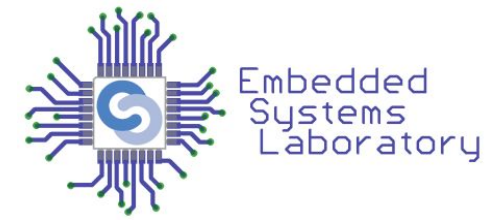
# IoT Level-3



Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

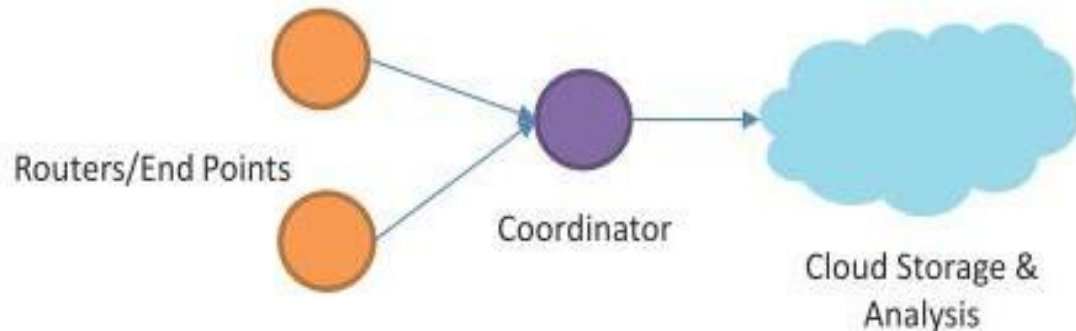
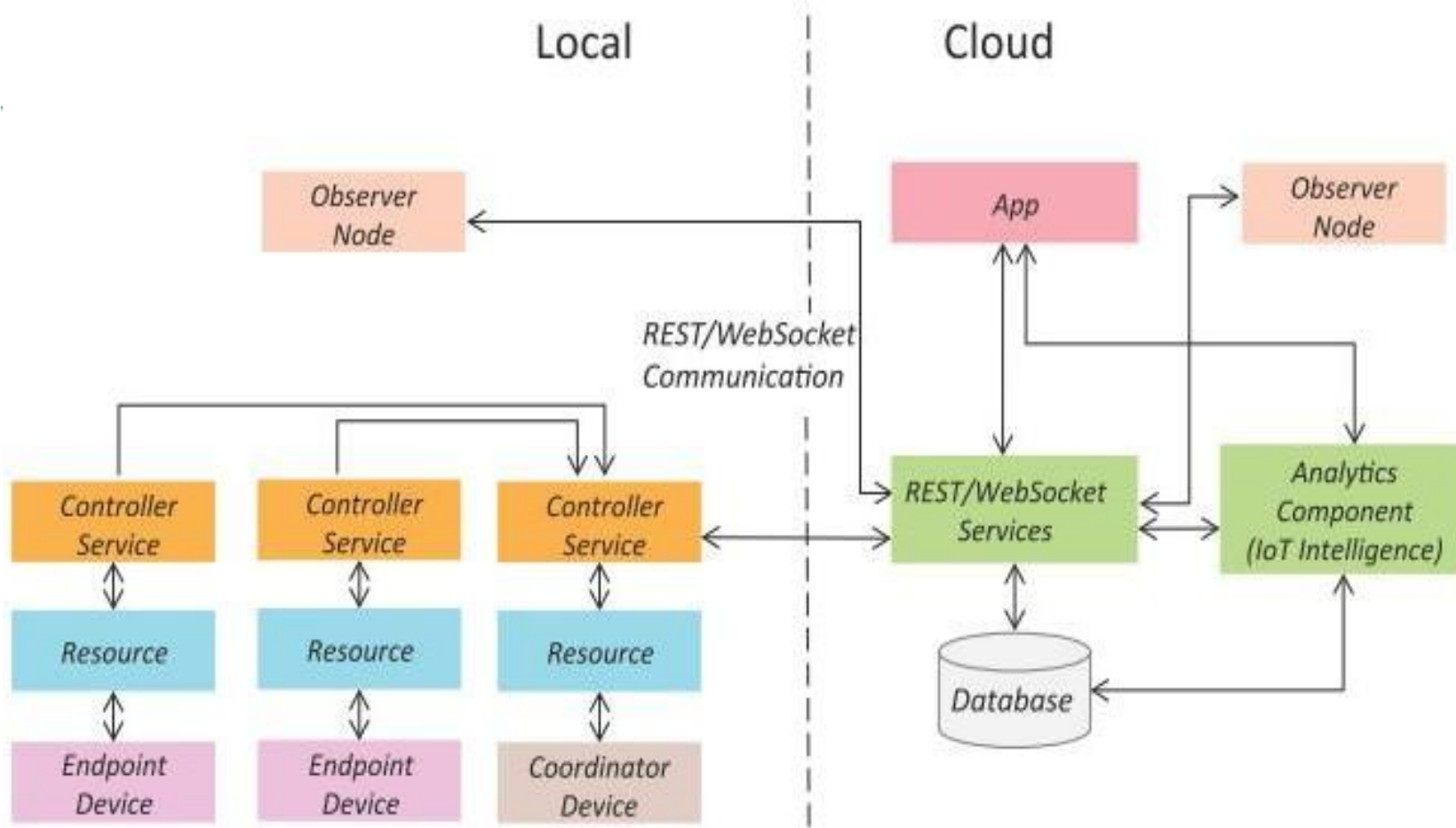
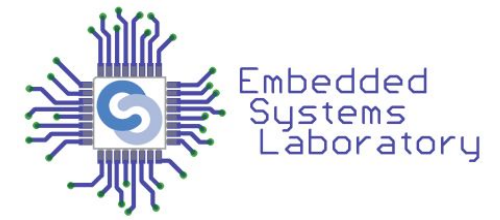


# IoT Level-4



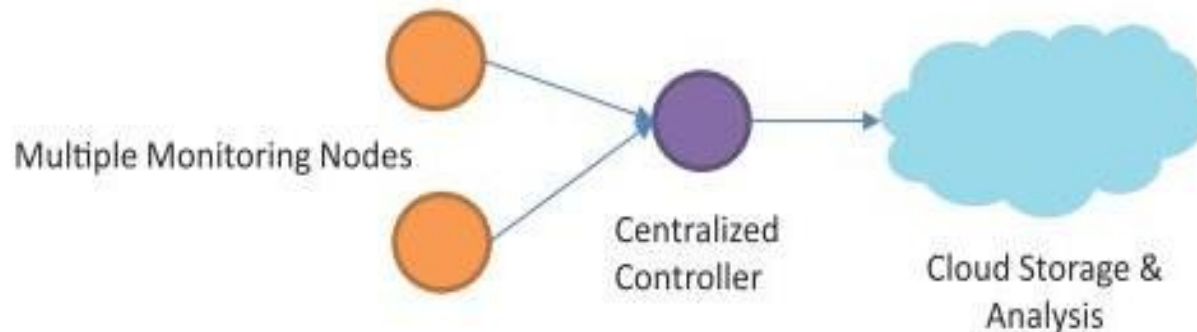
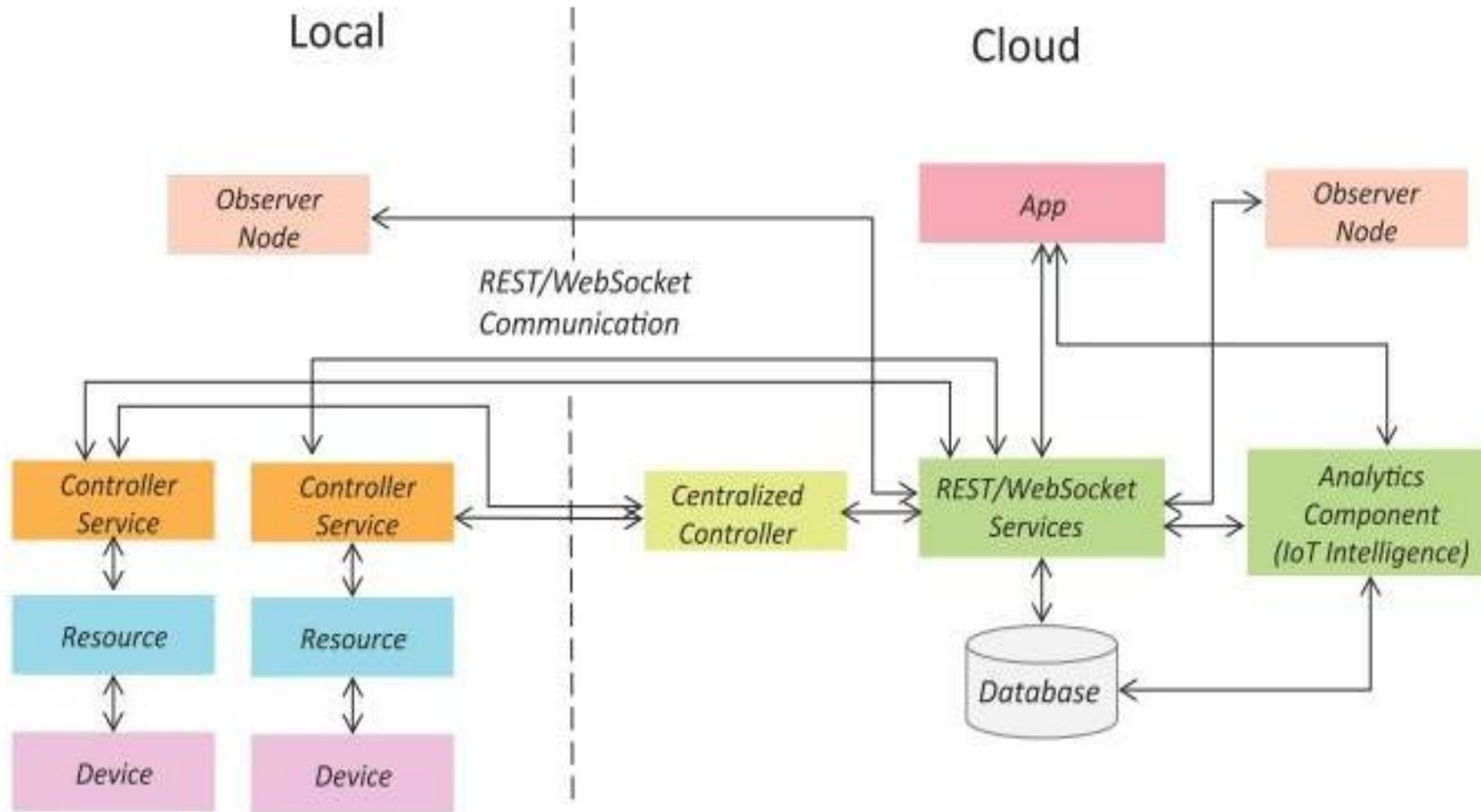
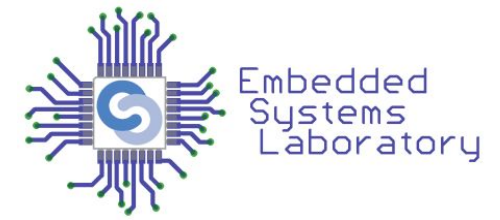
Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

# IoT Level-5



Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

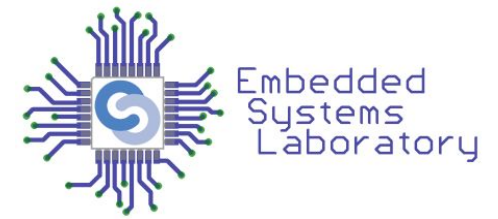
# IoT Level-6



Source: Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.

# Bibliography

---



- Arshdeep Bahga and Vijay Madisetti. 2014. Internet of Things: A Hands-On Approach.
  - Book website: <http://www.hands-on-books-series.com/iot.html>
- <https://www.sitepoint.com/developers-rest-api/>