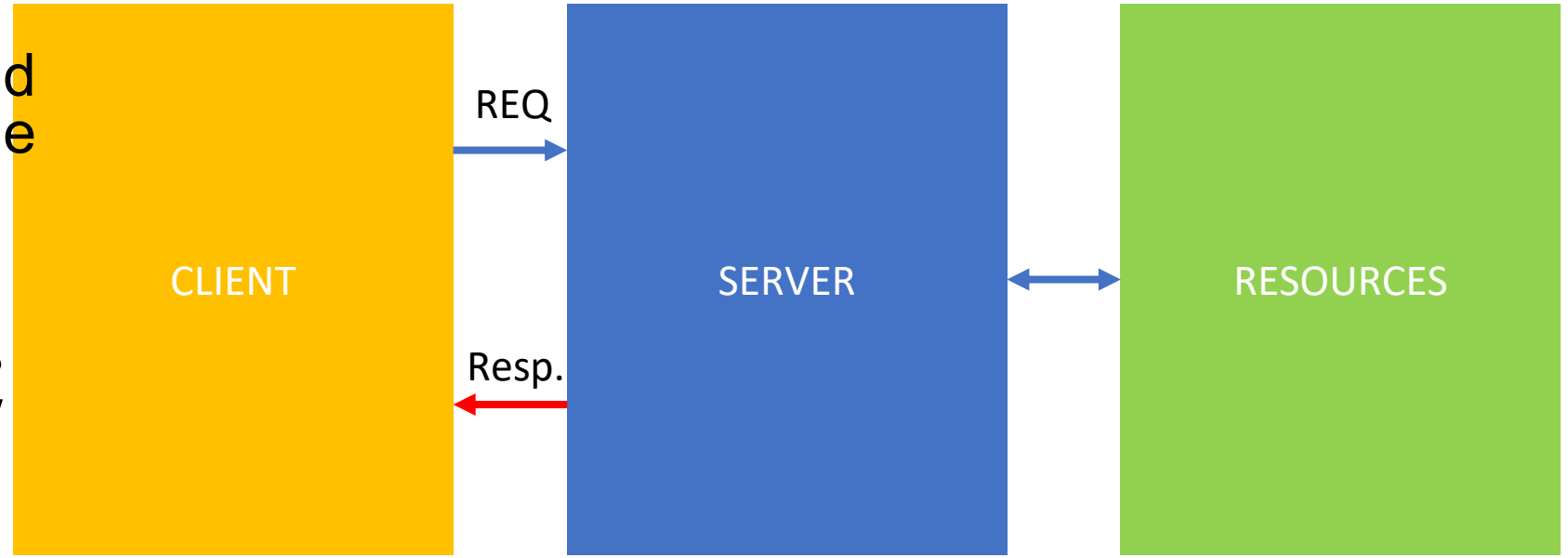
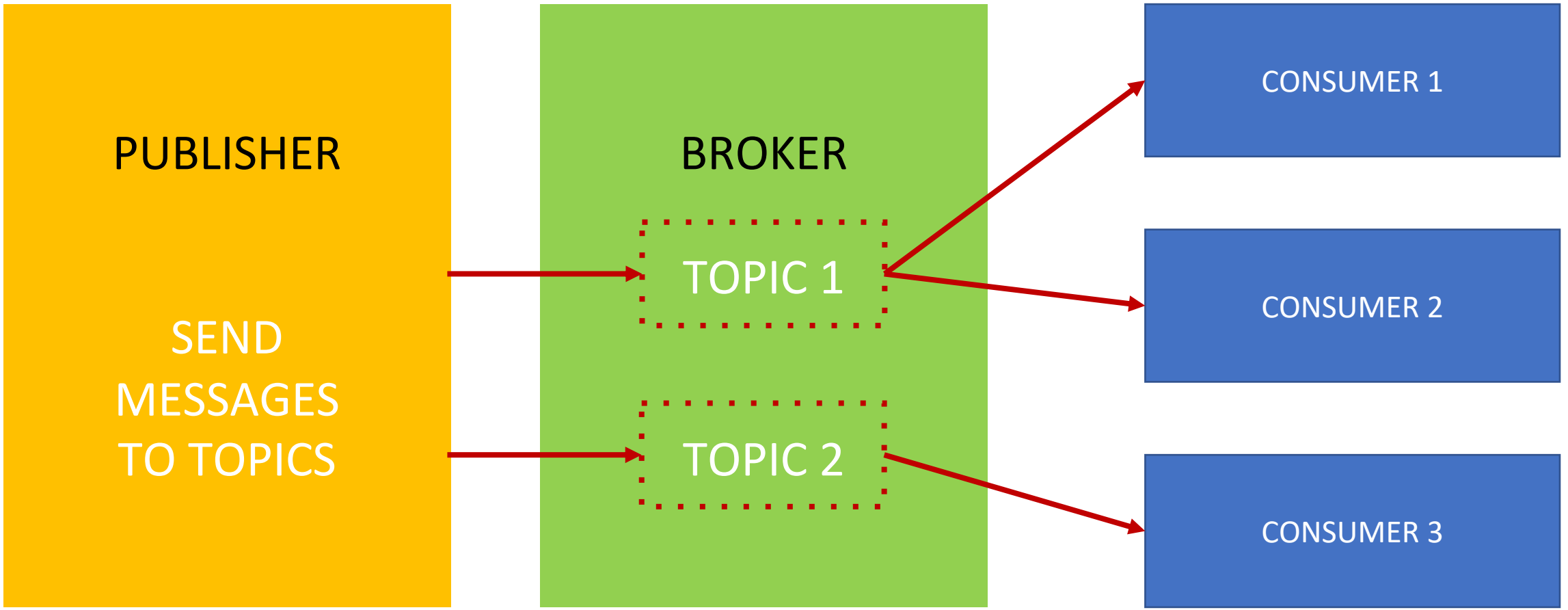
Communication Models of IoT

Client-Server Model

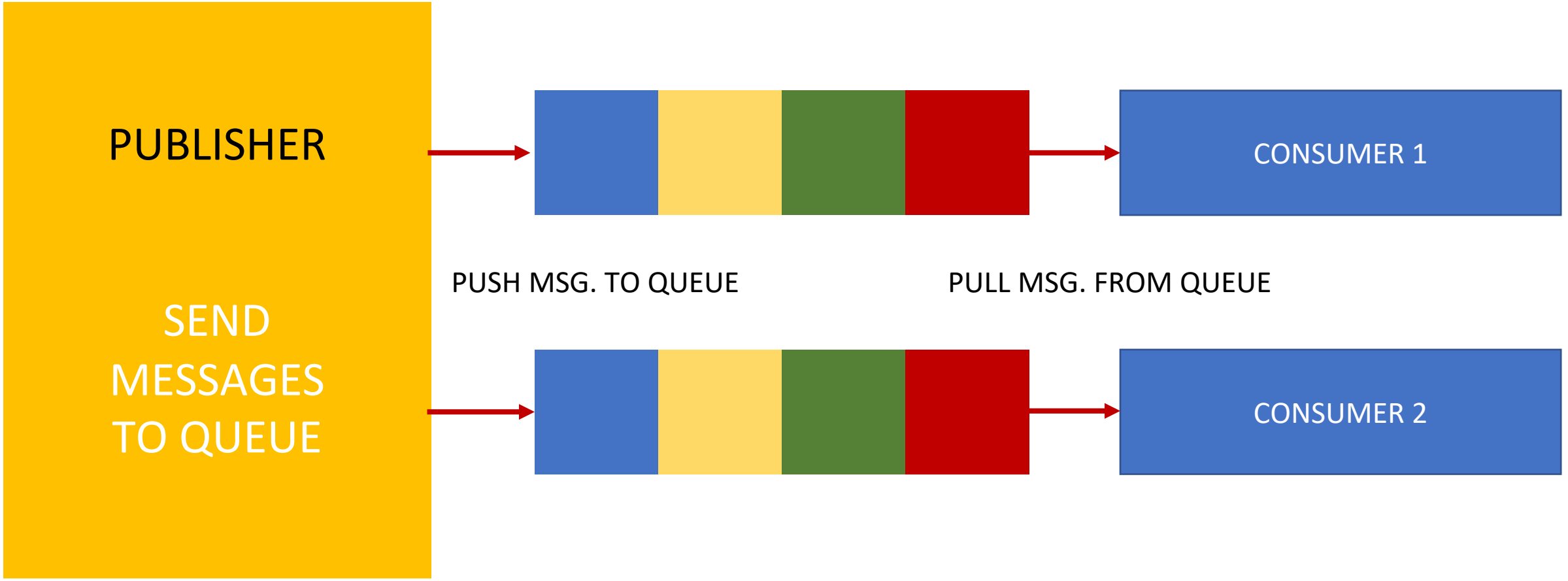
- Client-server is a communication model in which the client sends requests to the server and the server responds to the requests.
- When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client.



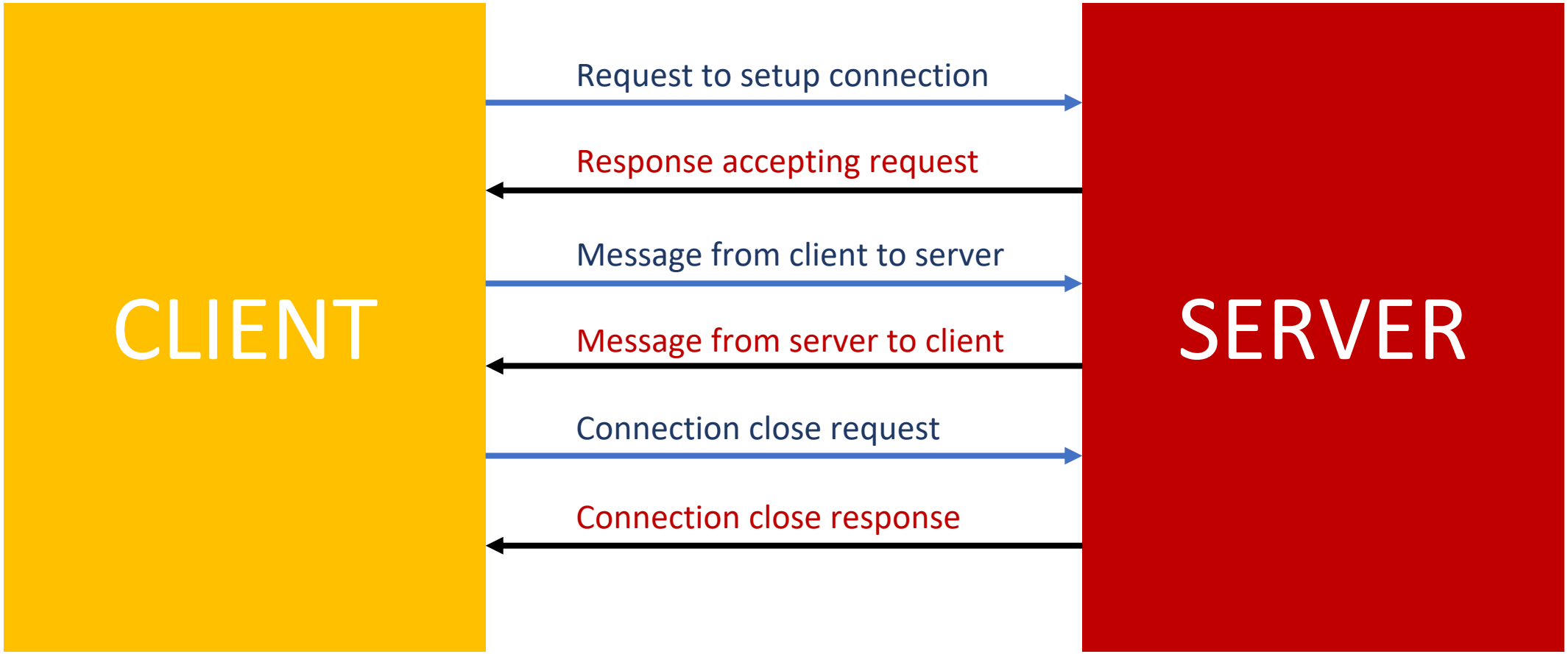
Publish-Subscribe Model



Push-Pull Model



Exclusive Pair Model



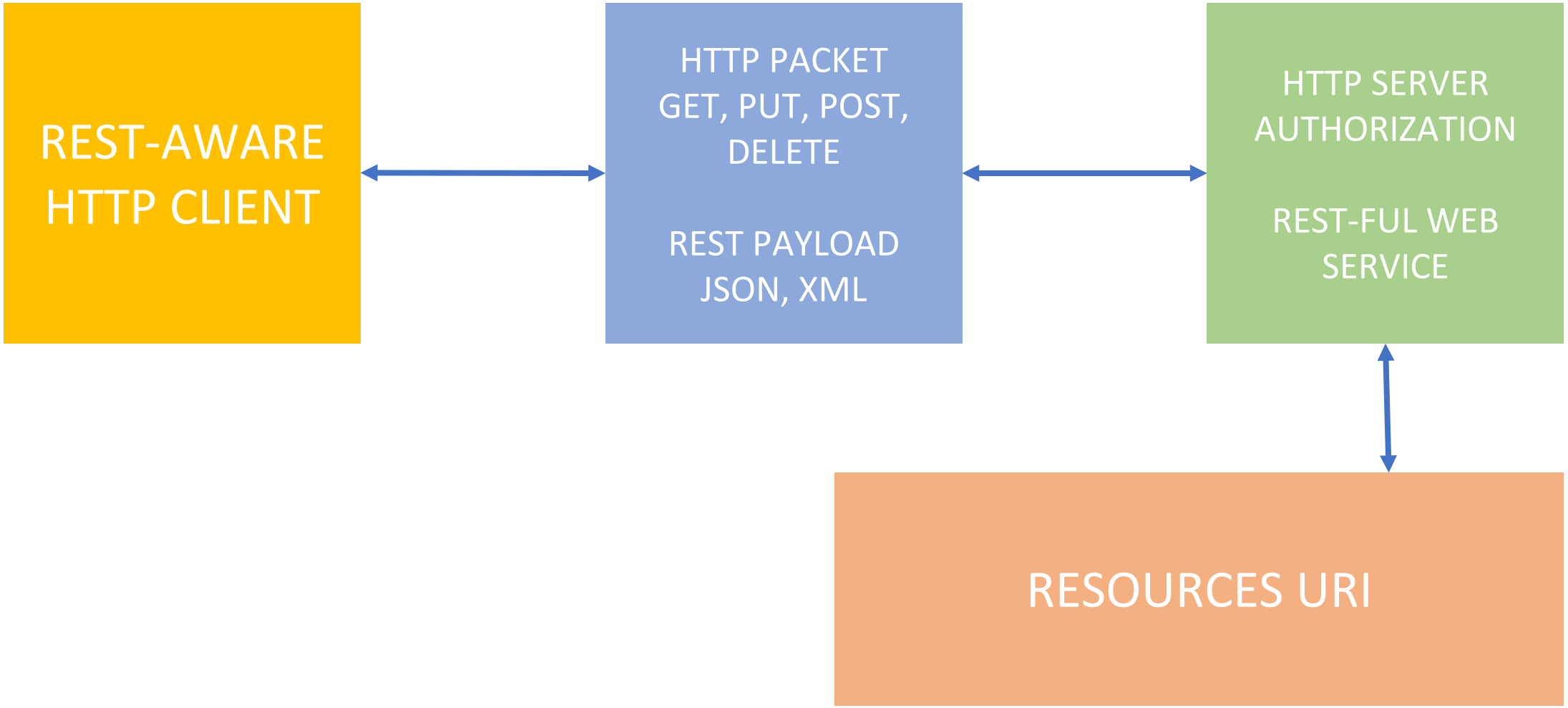
IoT Communication API

REST Communication APIs

- REpresentational State Transfer
- REST API is a way for two computer systems to communicate over HTTP in a similar way to web browsers and servers
- 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 - requesting client need not know whether it’s communicating with the actual server, a proxy, or any other intermediary
- Uniform interface
- Code on demand

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

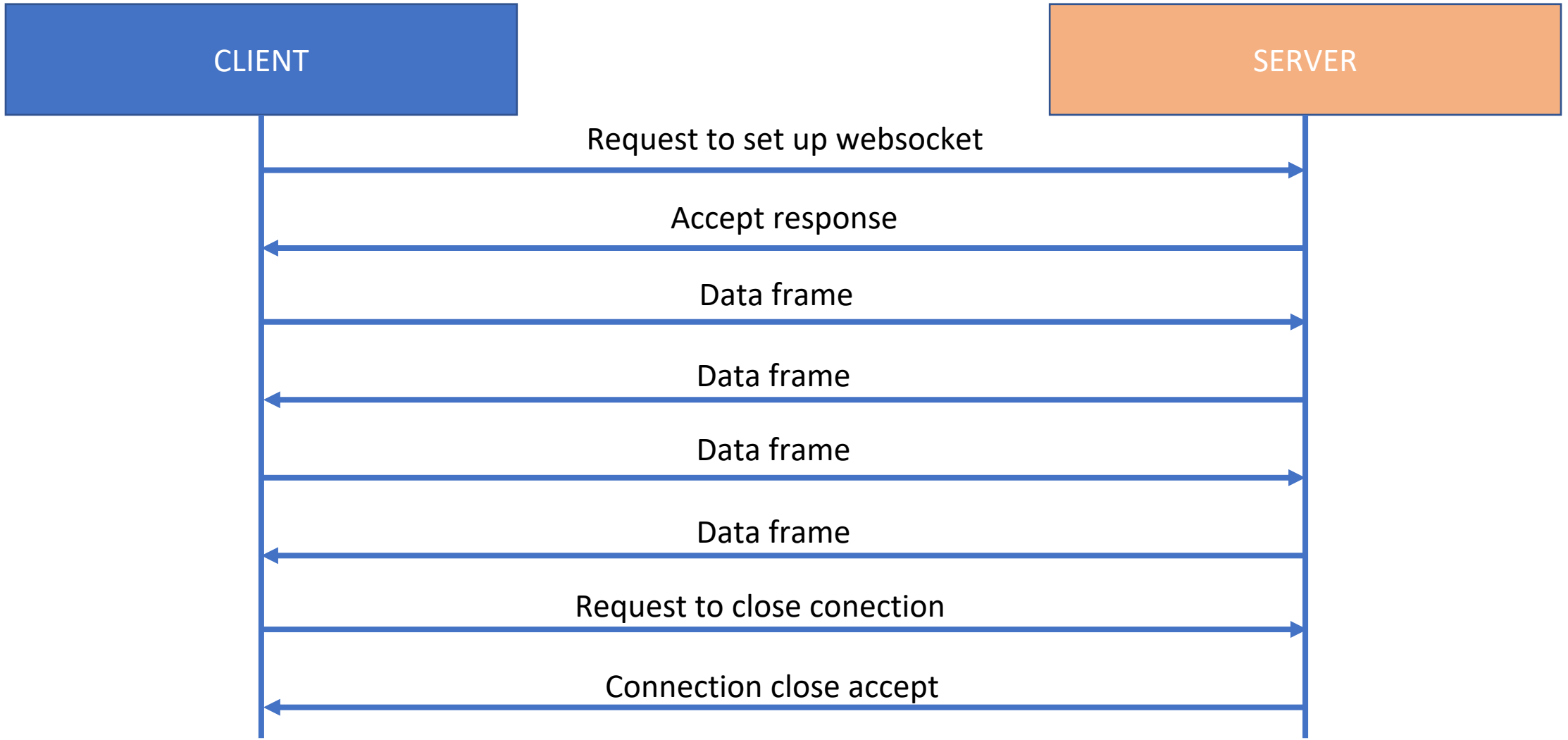
Examples

- 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

RESTful Web Service Reply

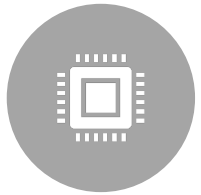
- **Response** payload can be whatever is practical: data, HTML, an image, an audio file, etc.
  - Typically JSON-encoded, but XML, CSV, simple strings, or any other format can be used
- An appropriate HTTP status code should also be set in the response header
  - 200 OK is most often used for successful requests
  - 201 Created may also be returned when a record is created
  - Errors should return an appropriate code (400 Bad Request, 404 Not Found, 401 Unauthorized)

WebSocket Based Communication



IoT Levels & Deployment Templates

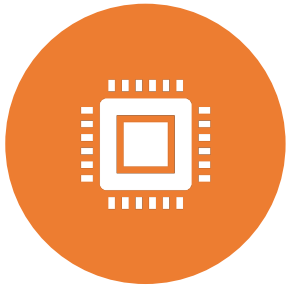
Device: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.



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.

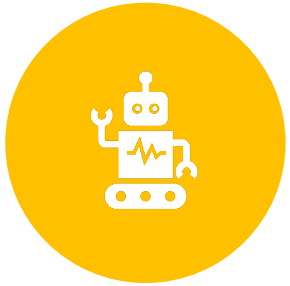
Controller Service:Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

IoT Levels & Deployment Templates

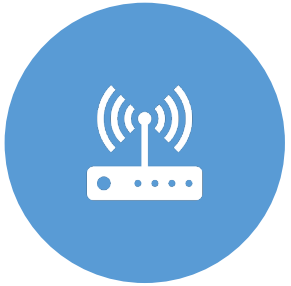


Database: Database can be either local or in the cloud and stores the data generated by the IoT device.

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).



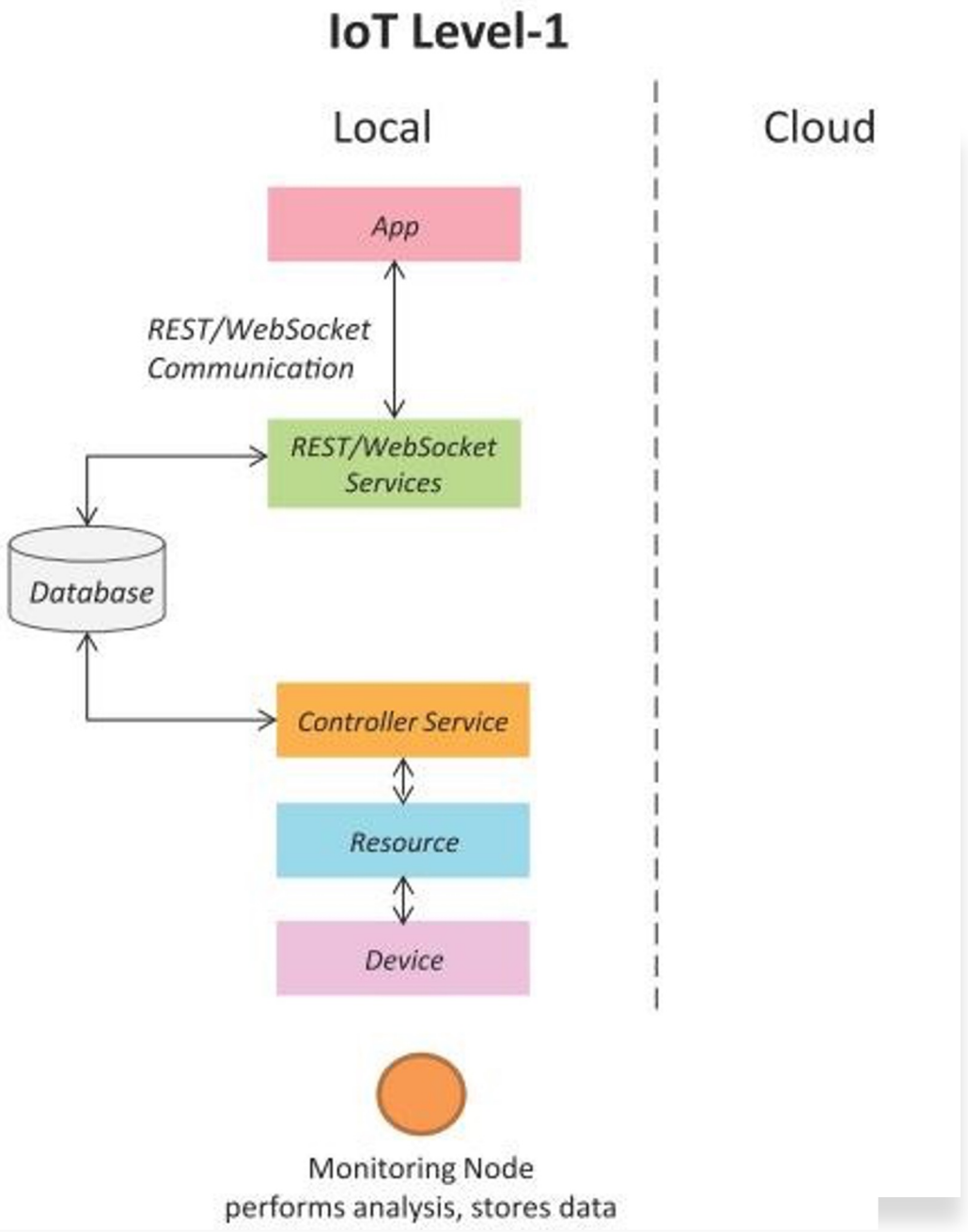
Analysis Component: The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.



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.

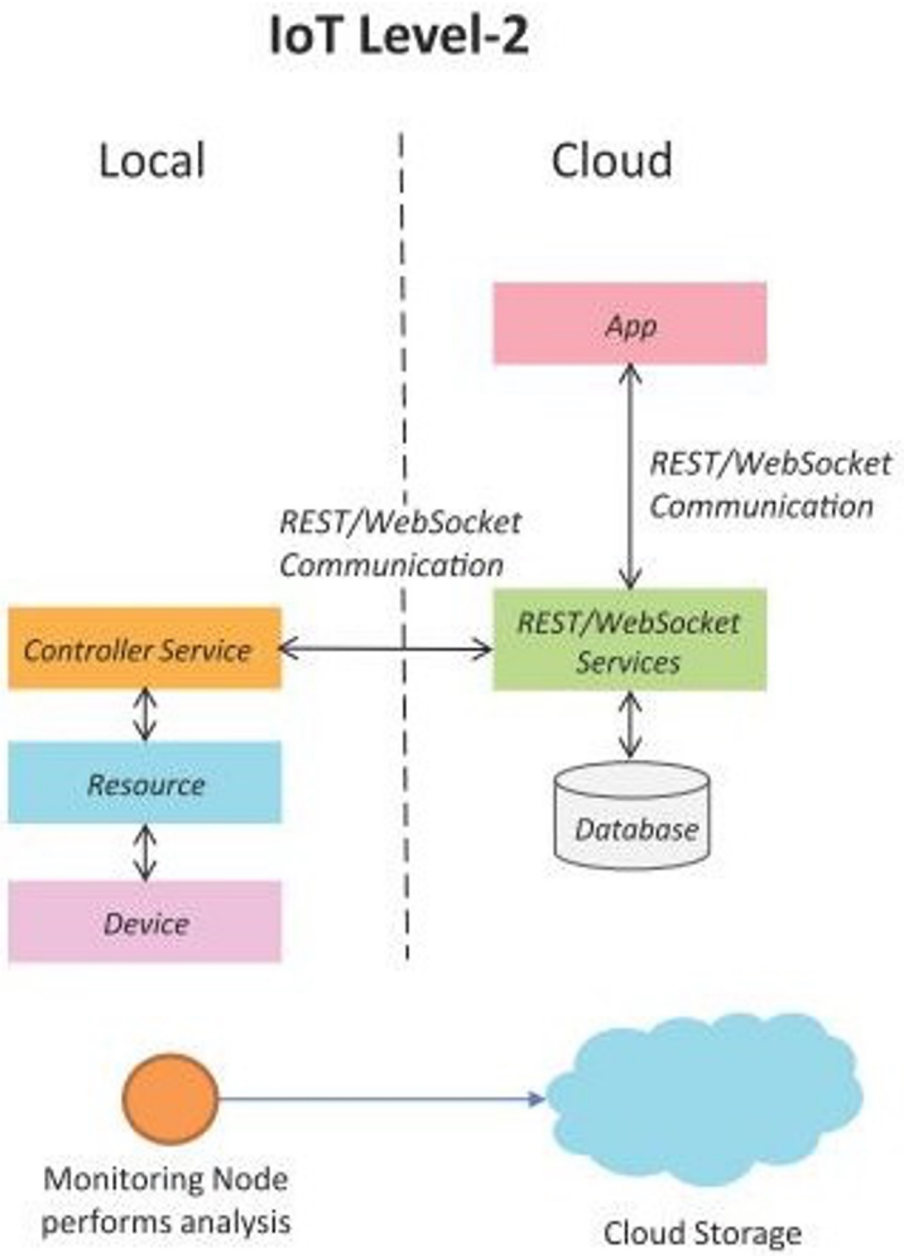
IoT Level-1

- A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application
- Level-1 IoT systems are suitable for modeling low- cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.



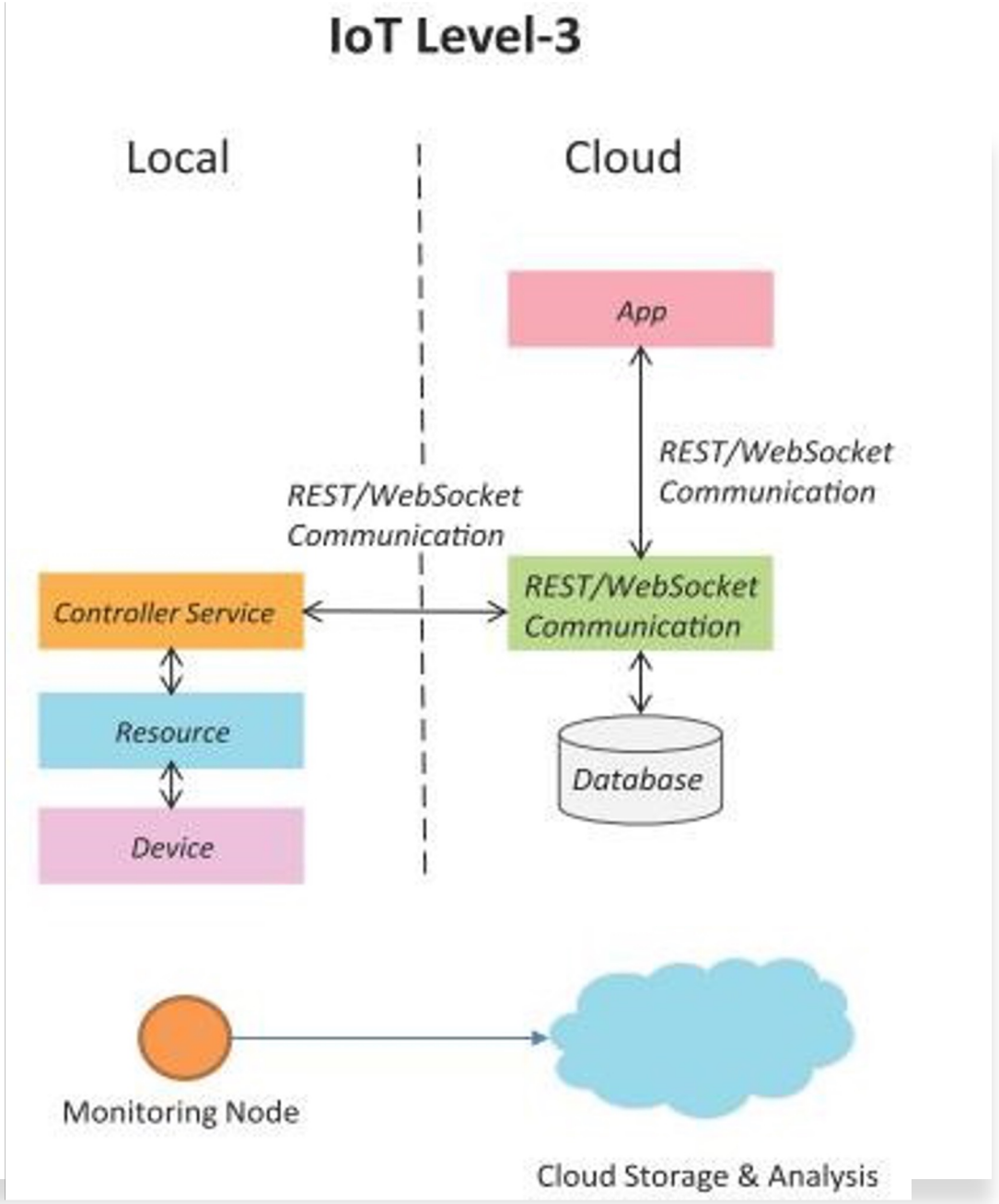
IoT Level-2

- A level-2 IoT system has a single node that performs sensing and/or actuation and local analysis.
- Data is stored in the cloud and application is usually cloud- based.
- Level-2 IoT systems are suitable for solutions where the data involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself.



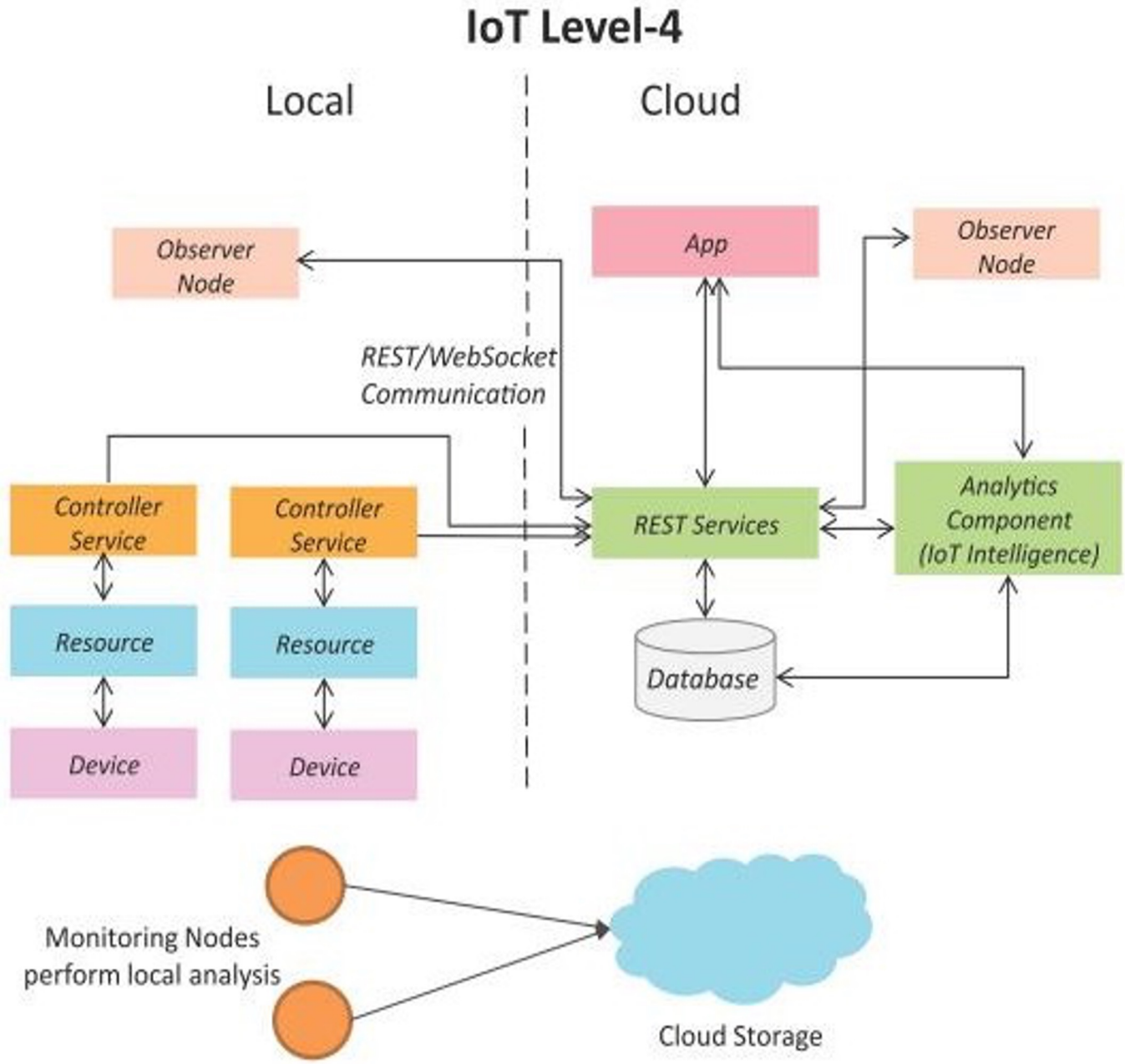
IoT Level-3

- A level-3 IoT system has a single node. Data is stored and analyzed in the cloud and application is cloud- based.
- Level-3 IoT systems are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.



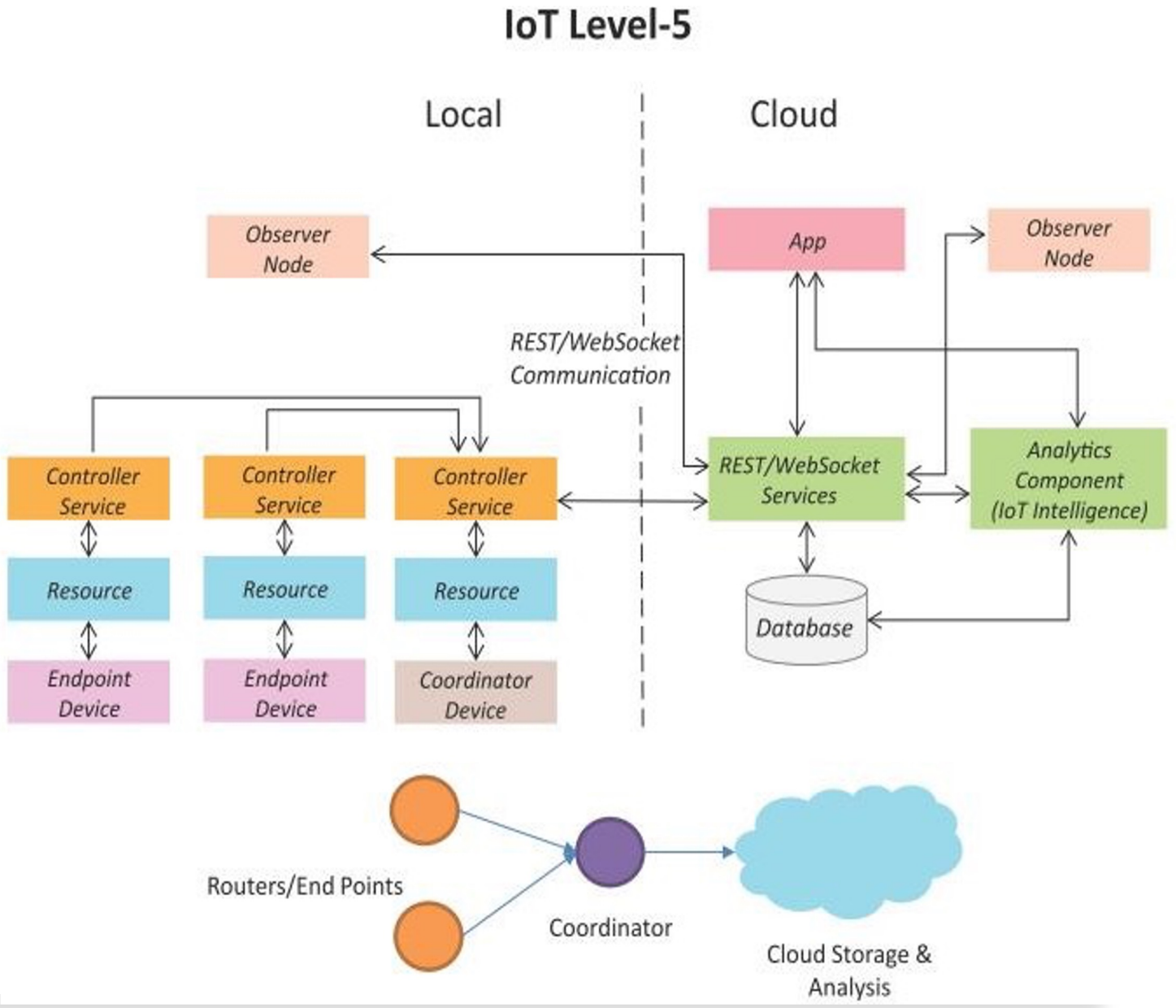
IoT Level-4

- A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based.
- Level-4 contains local and cloud- based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.
- Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.



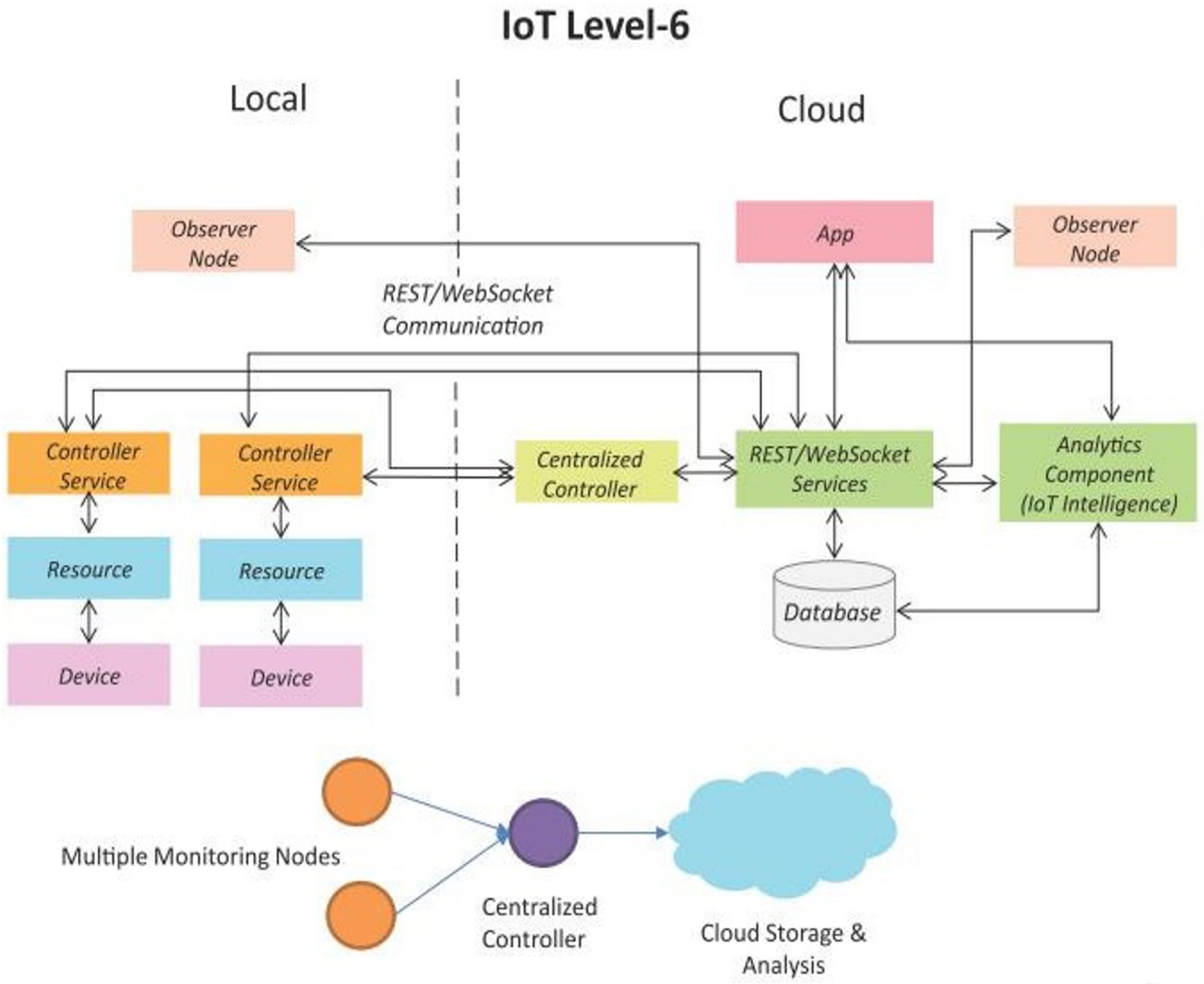
IoT Level-5

- A level-5 IoT system has multiple end nodes and one coordinator node.
- The end nodes that perform sensing and/or actuation.
- Coordinator node collects data from the end nodes and sends to the cloud.
- Data is stored and analyzed in the cloud and application is cloud-based.
- Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.



IoT Level-6

- A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.
- Data is stored in the cloud and application is cloud-based.
- The analytics component analyzes the data and stores the results in the cloud database.
- The results are visualized with the cloud-based application.
- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.



These slides contain materials from

- Internet of Things - A Hands-On Approach

  Book website: http://www.internet-of-things-book.com

  Bahga & Madisetti, © 2015