

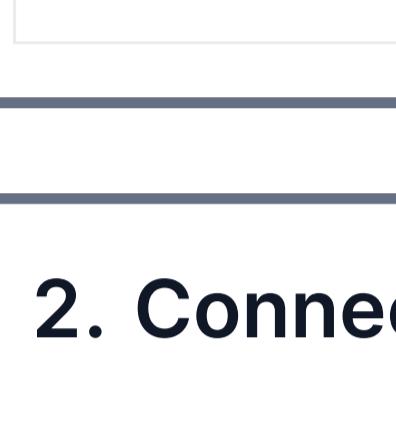


Design AWS Diagram

Empowering Design, Empowering You: Enhance your AWS journey with an all-in-one diagram solution.

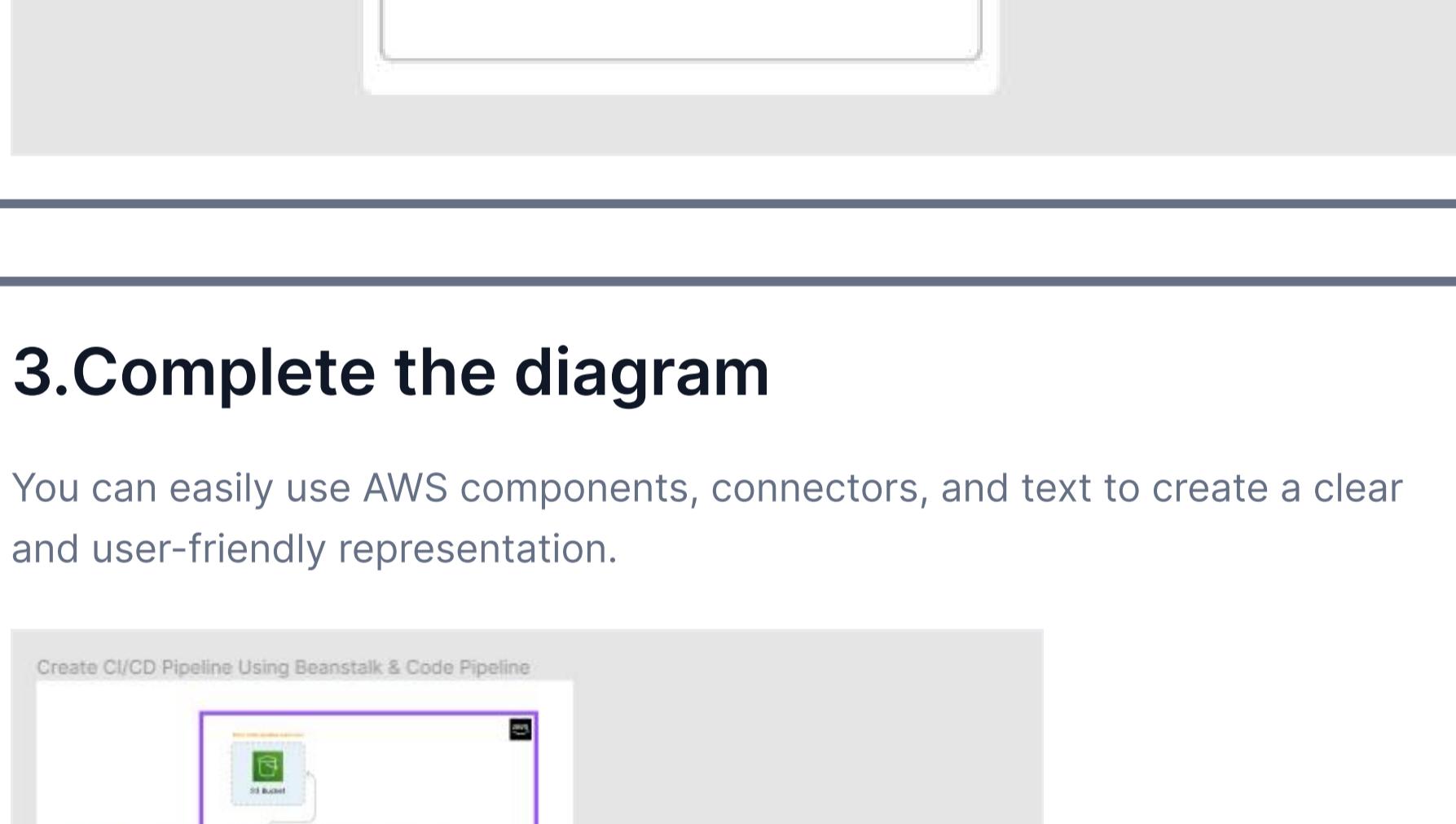
1. Assets panel / Aws-icon & components page

Open the asset panel, search for AWS components by name, and drag them to the playground. You can also copy components from the aws-icon & components page using Ctrl+C. This way you can efficiently work with the AWS components in your projects and experiments.



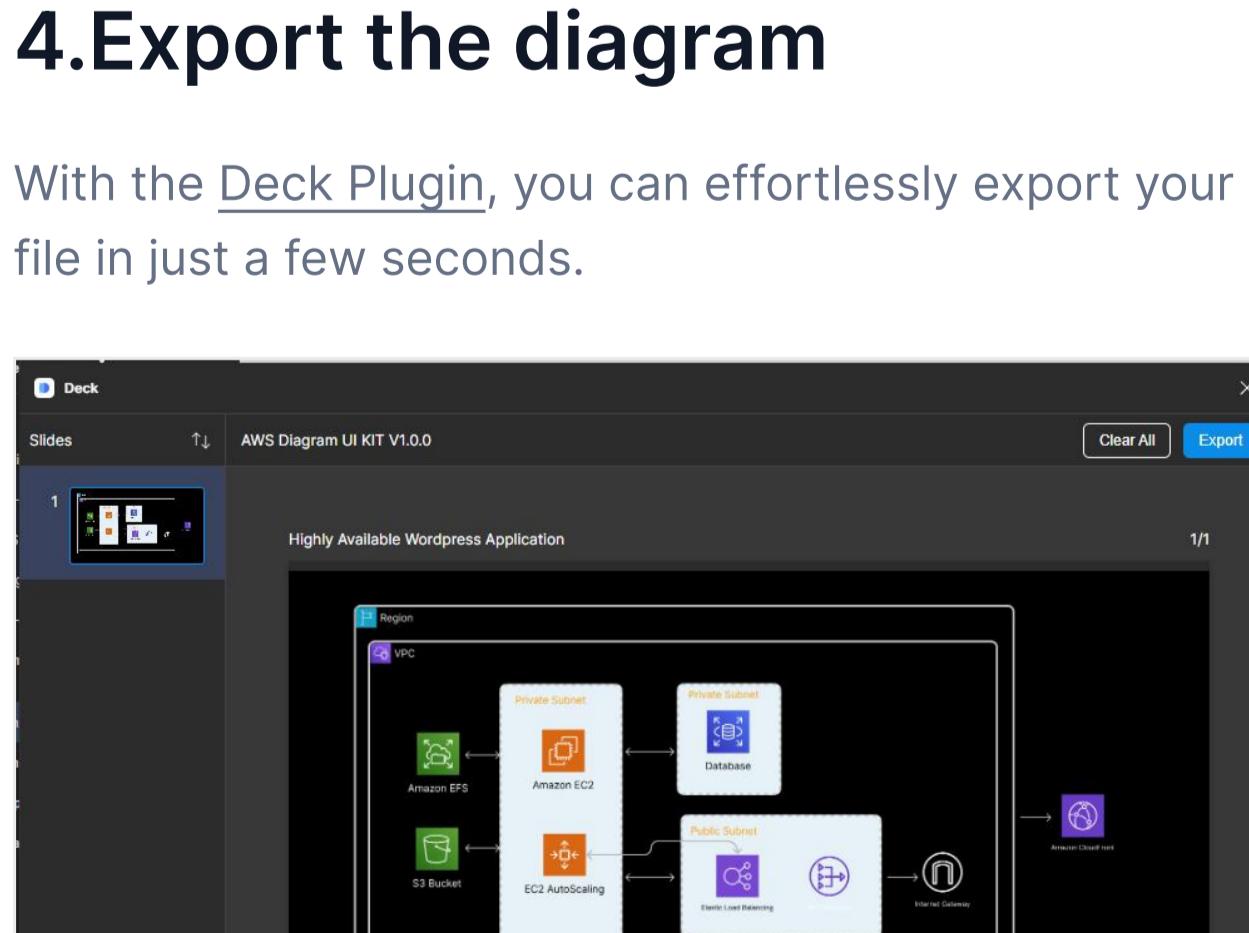
2. Connector line and box

Utilize the connector line and box from the AWS icon & components to establish connections between different AWS components. Also you can use [simpleflow plugin](#) for automatic connection.



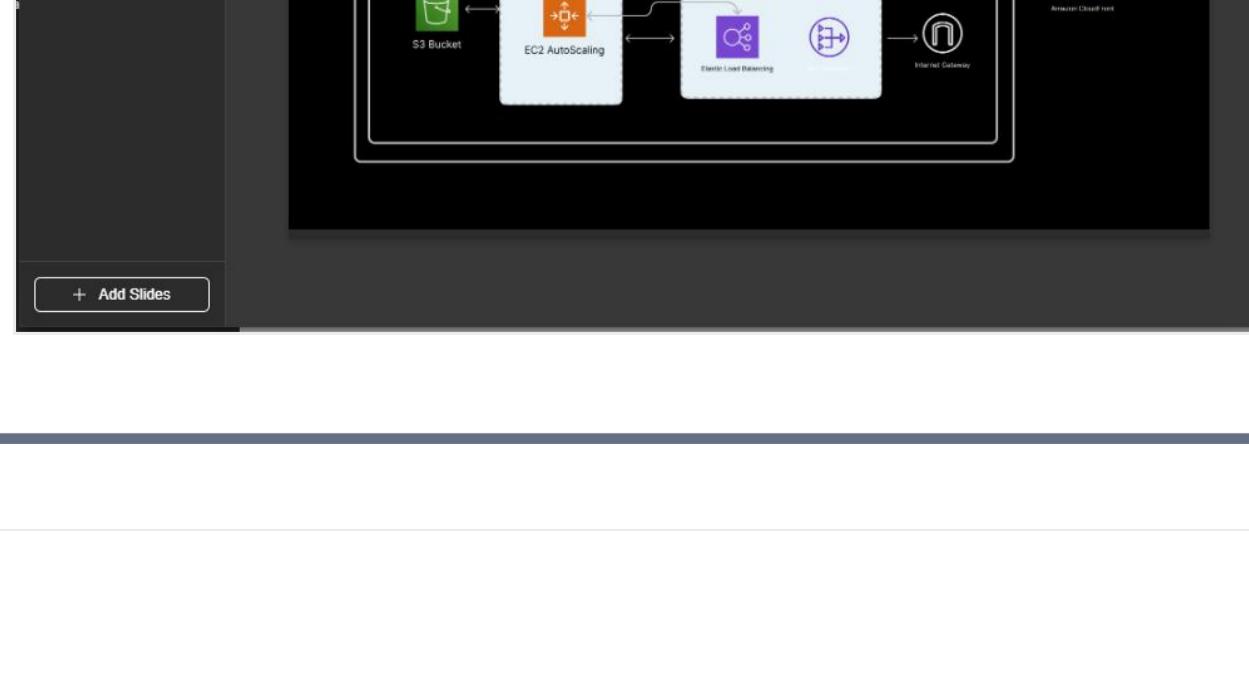
3. Complete the diagram

You can easily use AWS components, connectors, and text to create a clear and user-friendly representation.



4. Export the diagram

With the [Deck Plugin](#), you can effortlessly export your design to a PowerPoint file in just a few seconds.

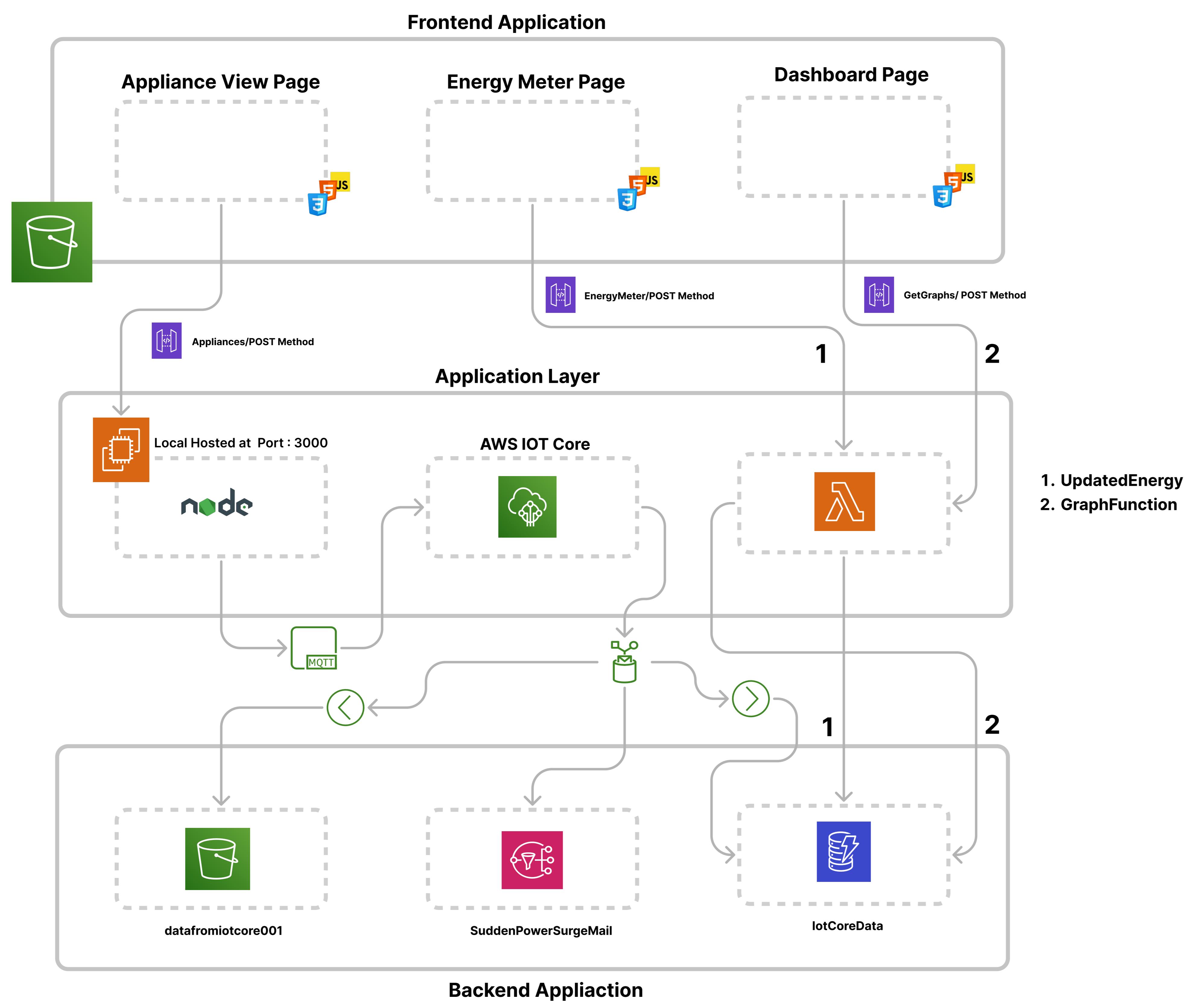




Route 53

→

Deploy With Autoscale



emPower - Home | Non Intelligent En | Smart Energy Met | AWS Diagram UI | AWS Diagram UI | Appliance Control

127.0.0.1:5500/Appliance.html

Appliance Control Panel

Add Appliance Schedule Appliance

Light Bulb

Appliance Name: Living Room Light

Switch On
Switch Off



Fan

Appliance Name: Bedroom Fan

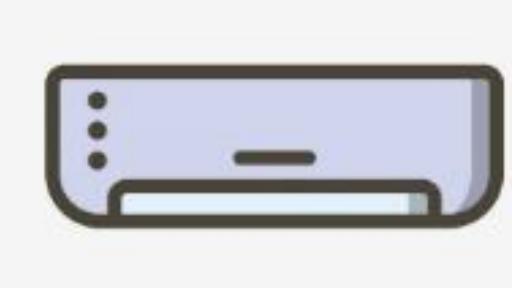
Switch On
Switch Off



AC

Appliance Name: Kitchen AC

Switch On
Switch Off



TV

Appliance Name: Samsung TV

Switch On
Switch Off



Washing Machine

Appliance Name: Washing Machine

Switch On
Switch Off



Introduce Anomaly

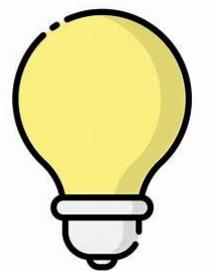
Living Room Light ▾ Introduce Anomaly

© 2024 Appliance Control Panel

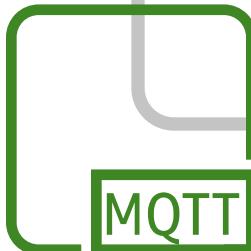
Search

11:02 17-10-2024 ENG US

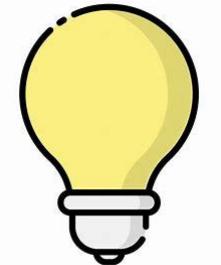
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



IoT Device



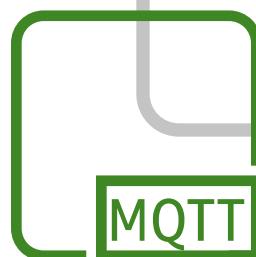
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



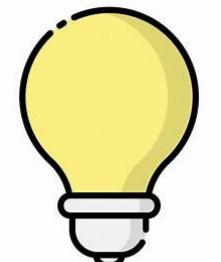
IoT Device



IoT Thing



MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



IoT Device



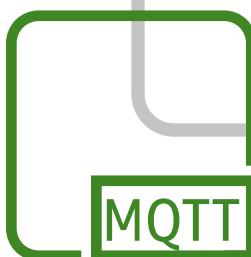
Digital Certificate



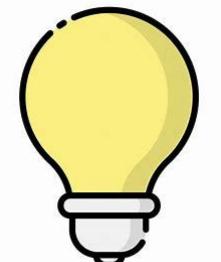
Private Keys



IoT Thing



MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



IoT Device



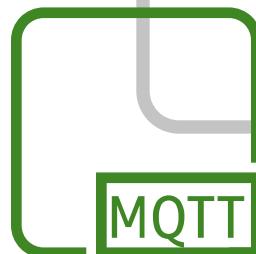
Digital Certificate



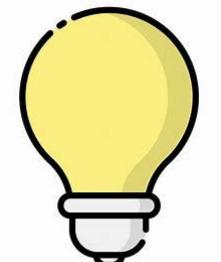
Private Keys



IoT Thing



MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



IoT Device



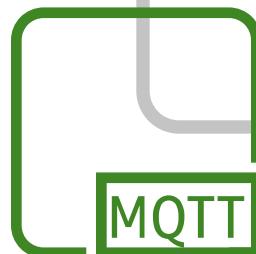
Digital Certificate



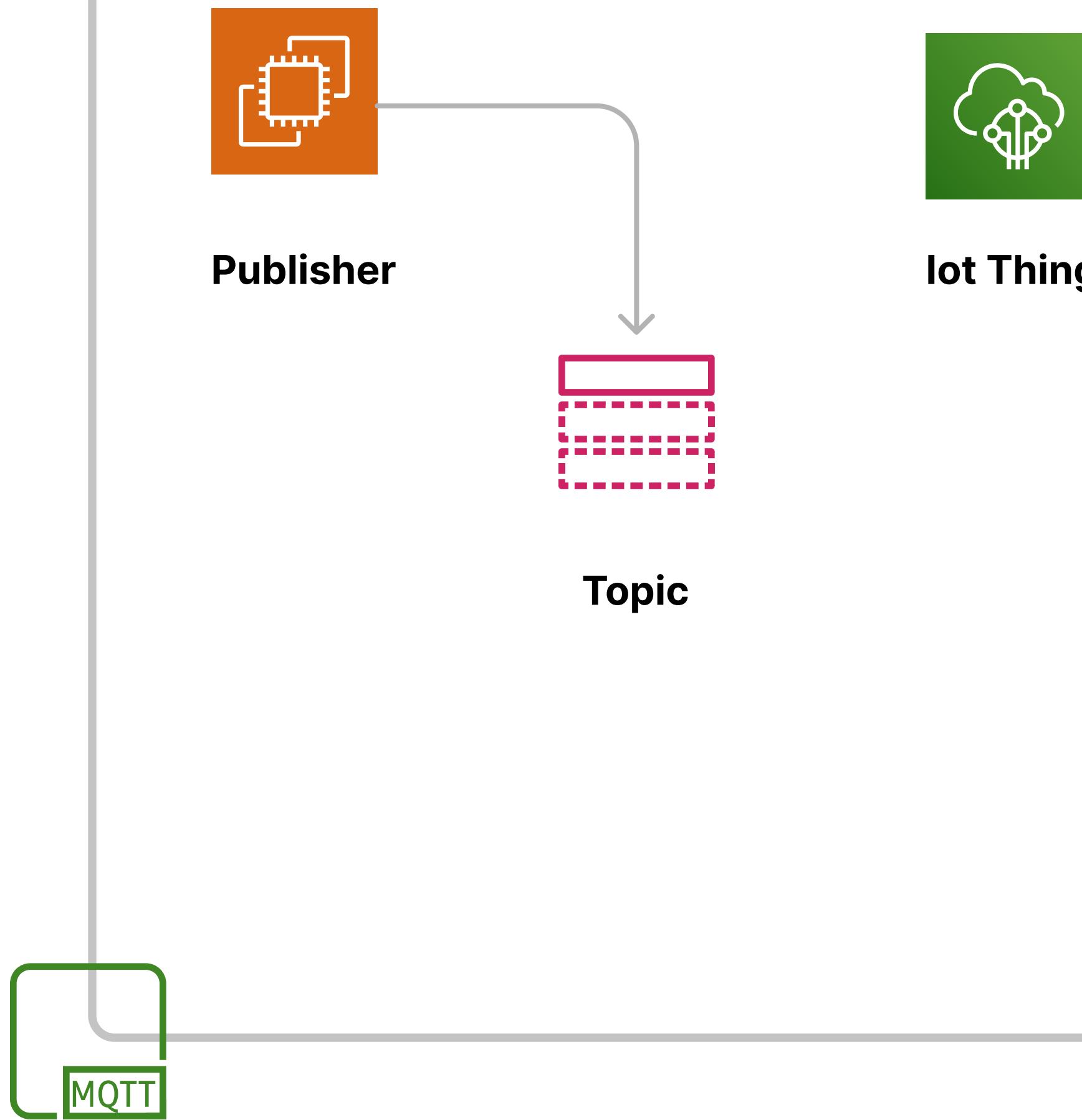
Private Keys



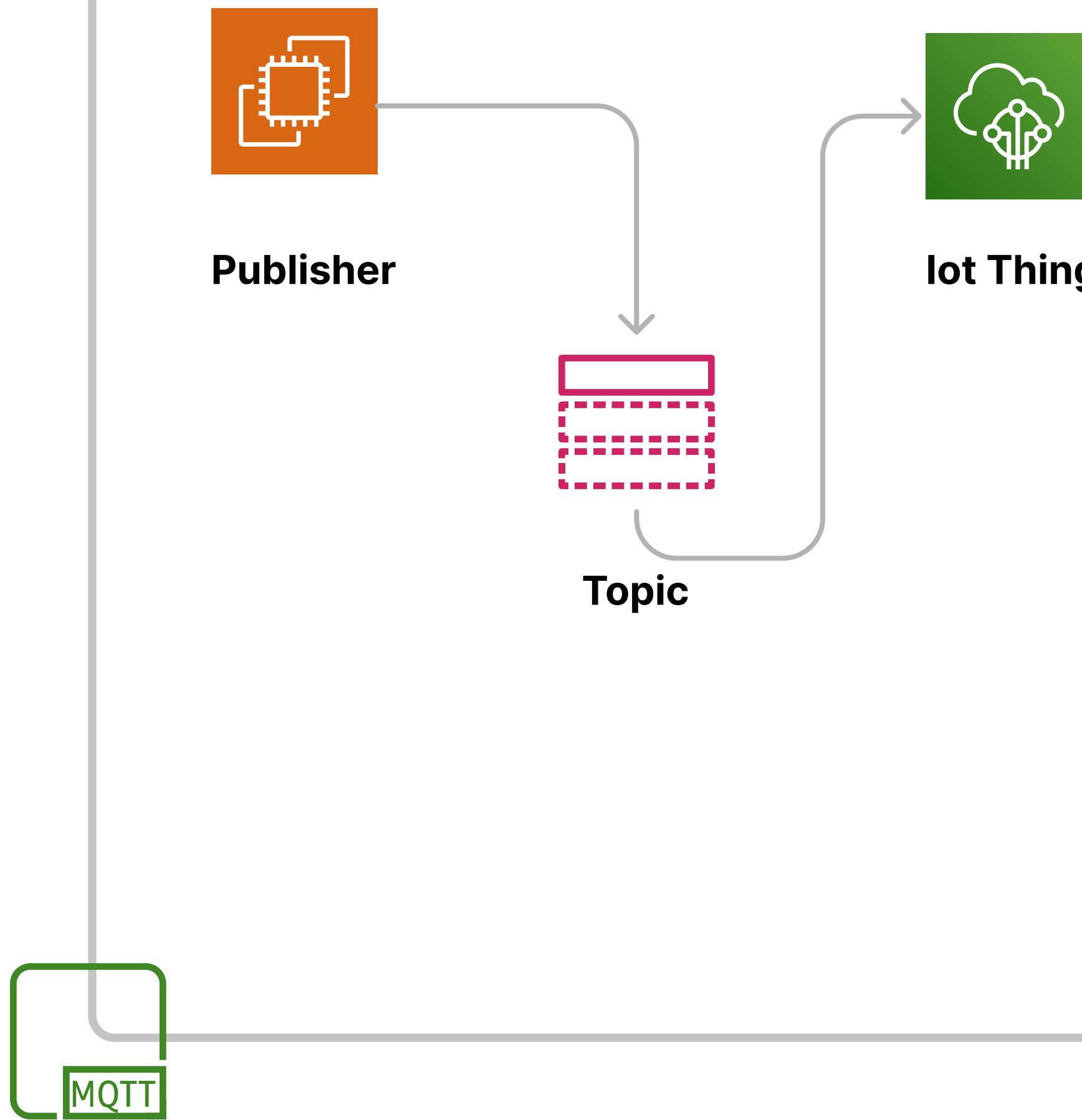
IoT Thing



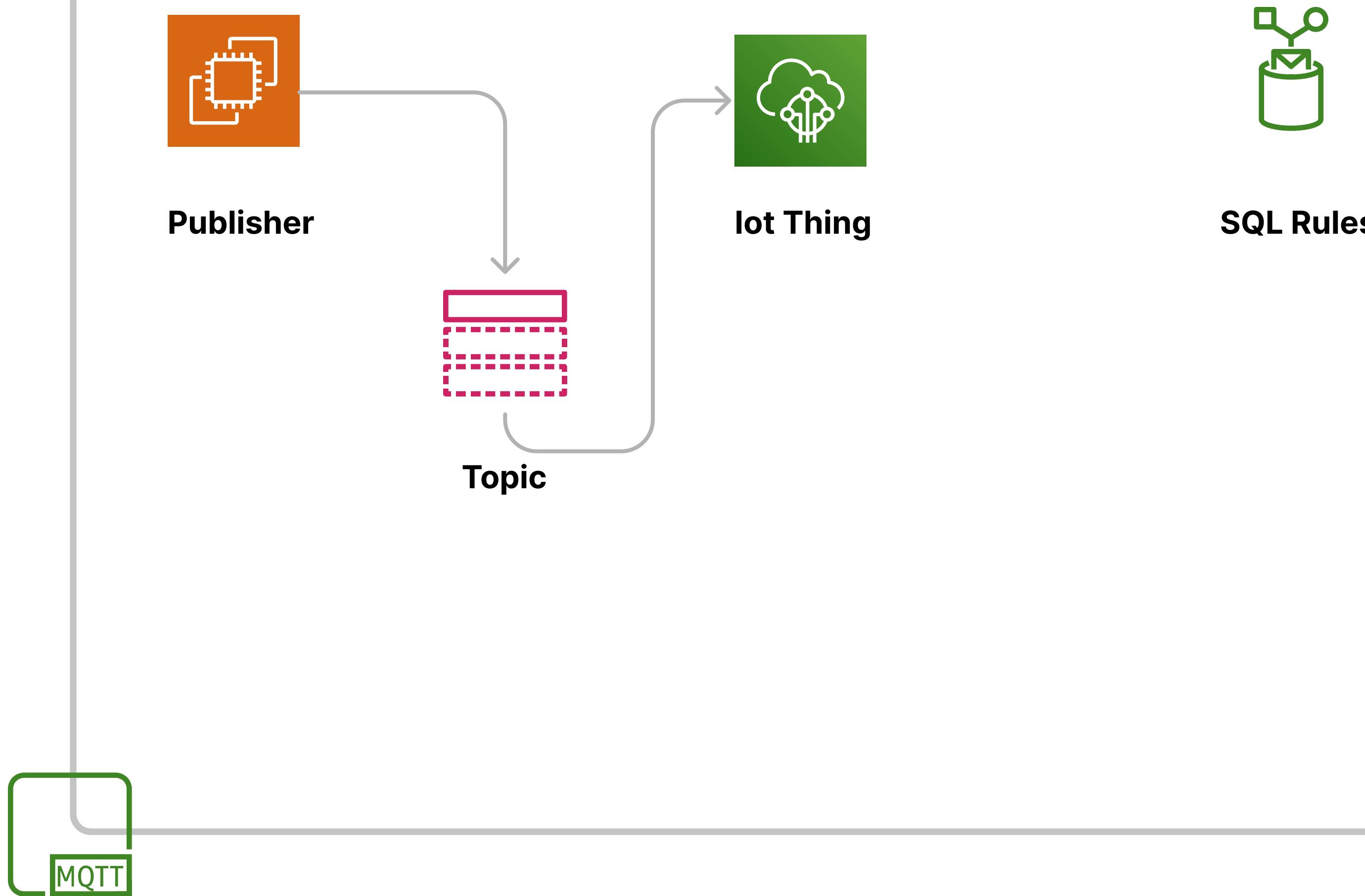
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



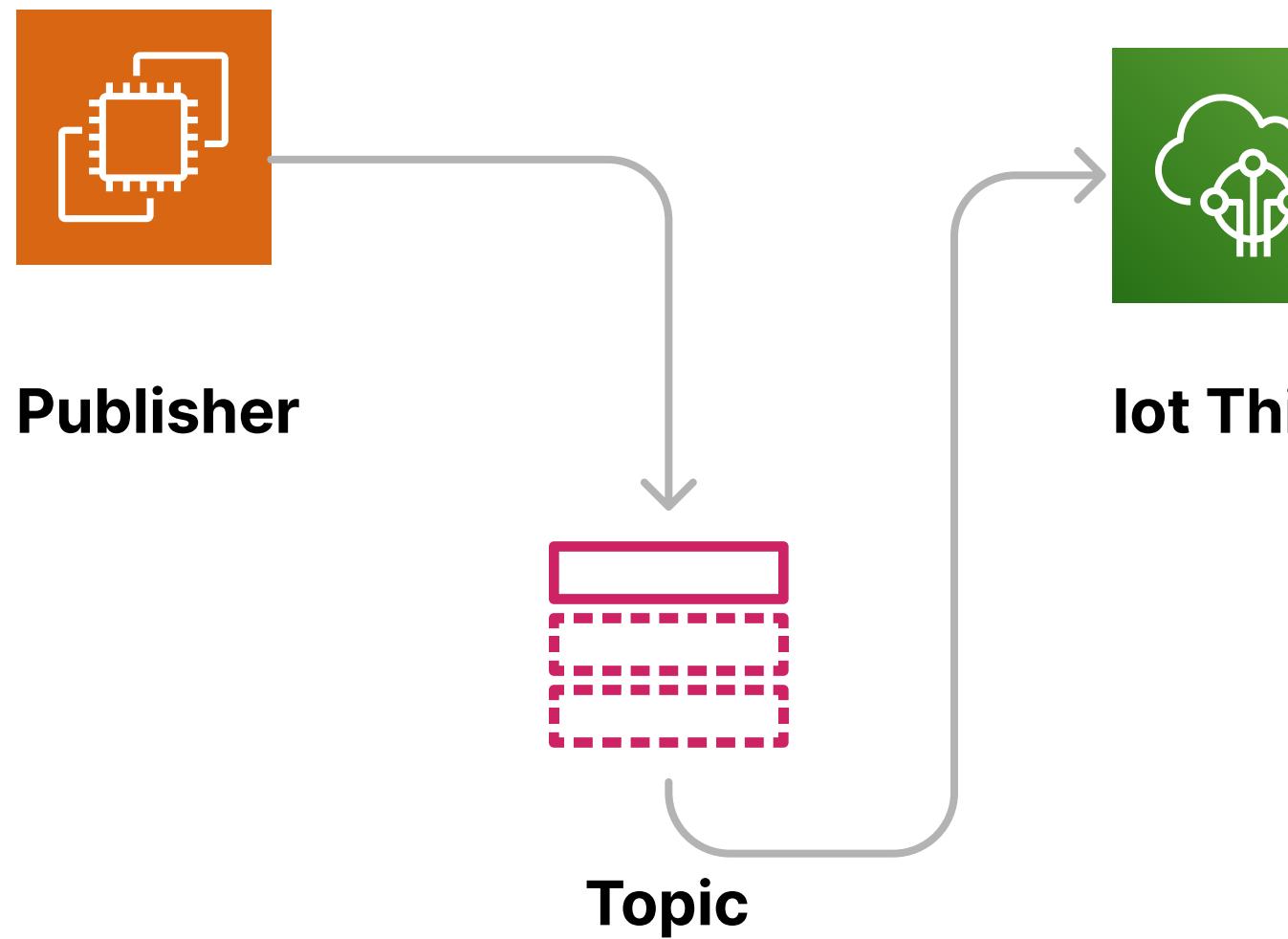
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.

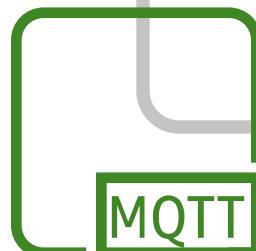


MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.

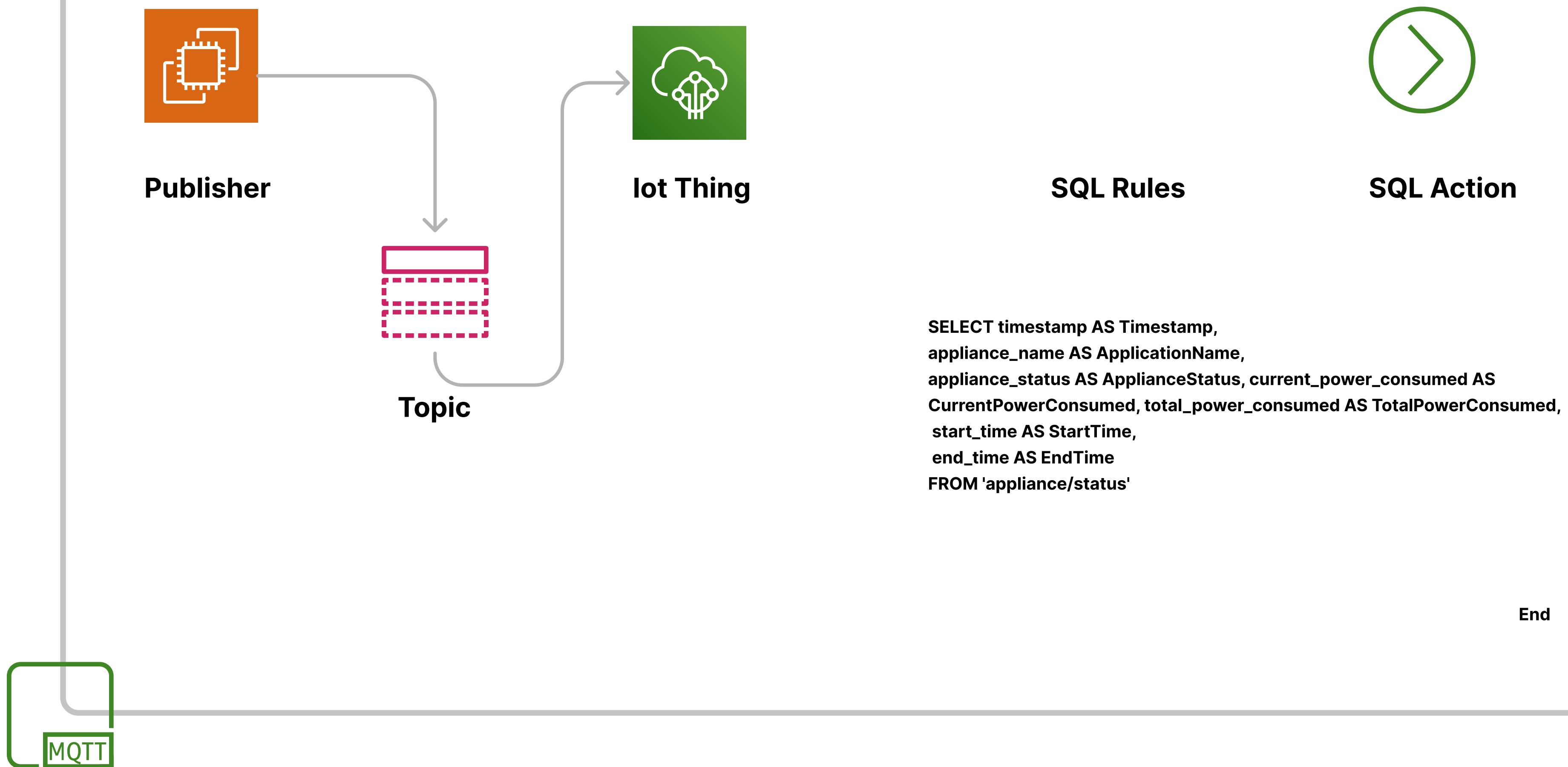


SQL Rules

SQL statement		SQL version
SQL statement	SELECT timestamp AS Timestamp, appliance_name AS ApplicationName, appliance_status AS ApplianceStatus, current_power_consumed AS CurrentPowerConsumed, total_power_consumed AS TotalPowerConsumed, start_time AS StartTime, end_time AS EndTime FROM 'appliance/status'	2016-03-23
<hr/>		
Actions	Error action	Tags
<hr/>		
Actions (1)		
Actions occur when an event is triggered. Actions are executed until all actions are completed or an error occurs. To add or remove actions, you will need to edit the rule.		
<hr/>		
Service	Action	View details
<input type="radio"/> DynamoDBv2	Split message into multiple columns of a DynamoDB table (DynamoDBv2)	

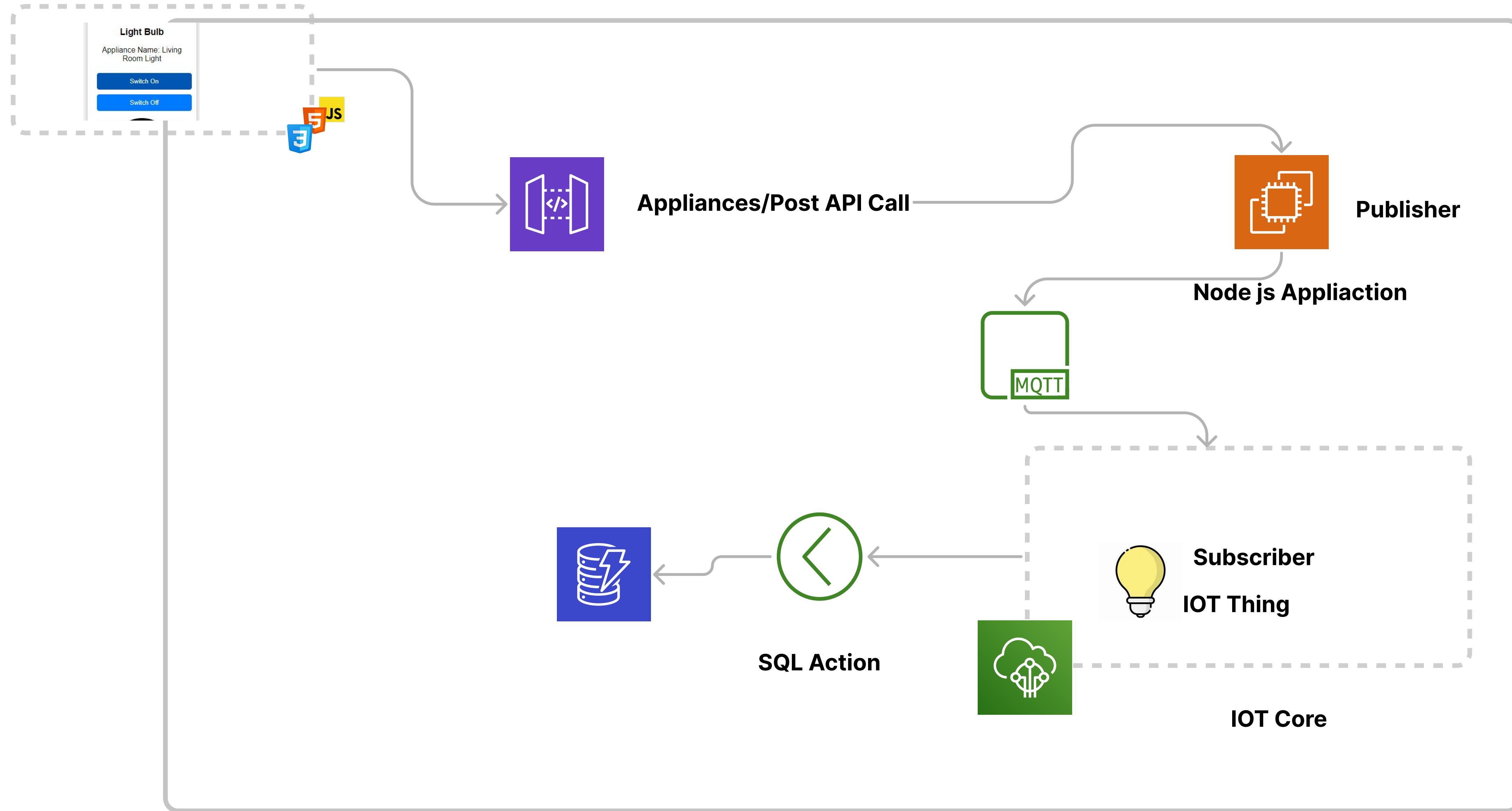


MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) applications. It allows devices to send (publish) and receive (subscribe) messages through a broker, typically over low-bandwidth, high-latency networks.



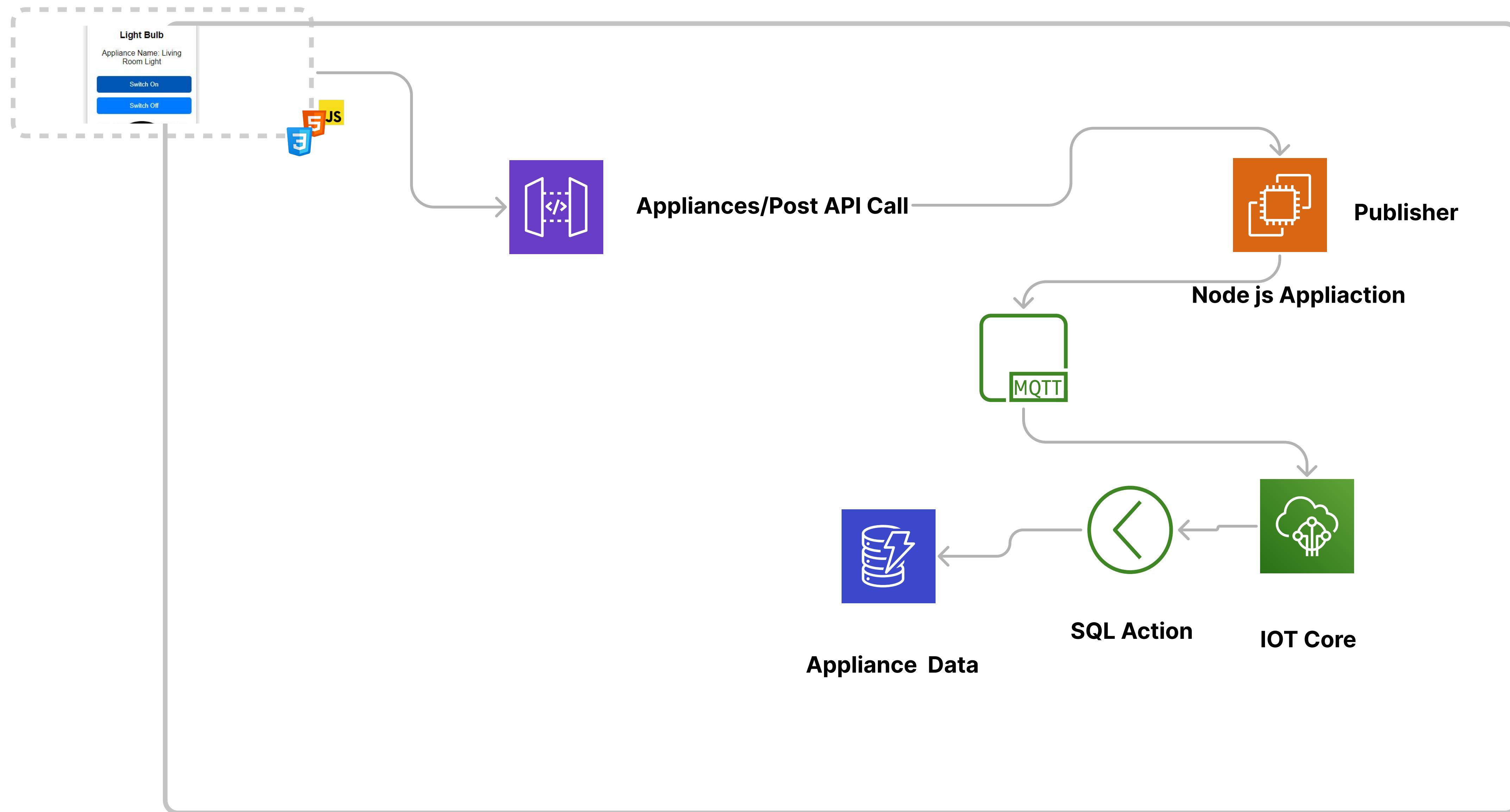
Appliance View Page

Frontend Application

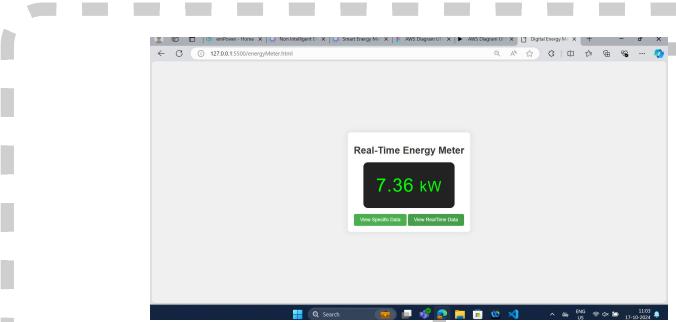


Appliance View Page

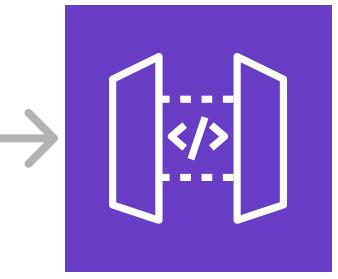
Frontend Application



DashBoard View

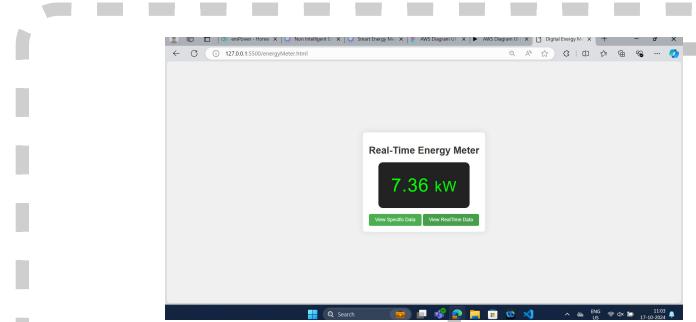
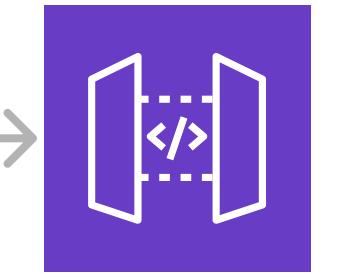


Angular
JS



Frontend Application

Dashboard View

5
JS

Get Graphs/Post API Call

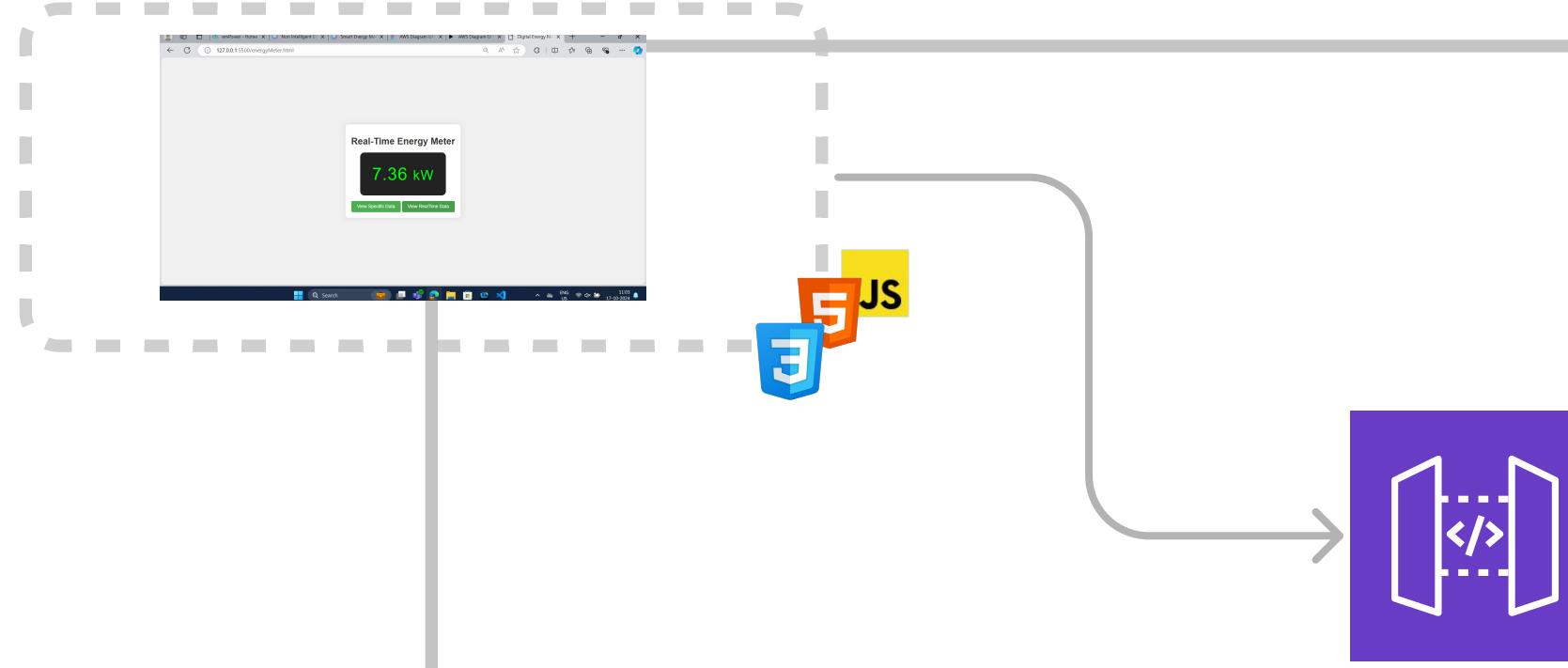
A screenshot of a network traffic capture tool interface, likely Postman or similar, showing a POST request. The "Payload" tab is selected. The "Request Payload" section displays the following JSON data:

```
{startTime: "2024-10-04T12:35", endTime: "2024-10-26T12:35", applianceName: "ALL"}  
applianceName: "ALL"  
endTime: "2024-10-26T12:35"  
startTime: "2024-10-04T12:35"
```

Frontend Application



DashBoard View



Appliances/Post API Call

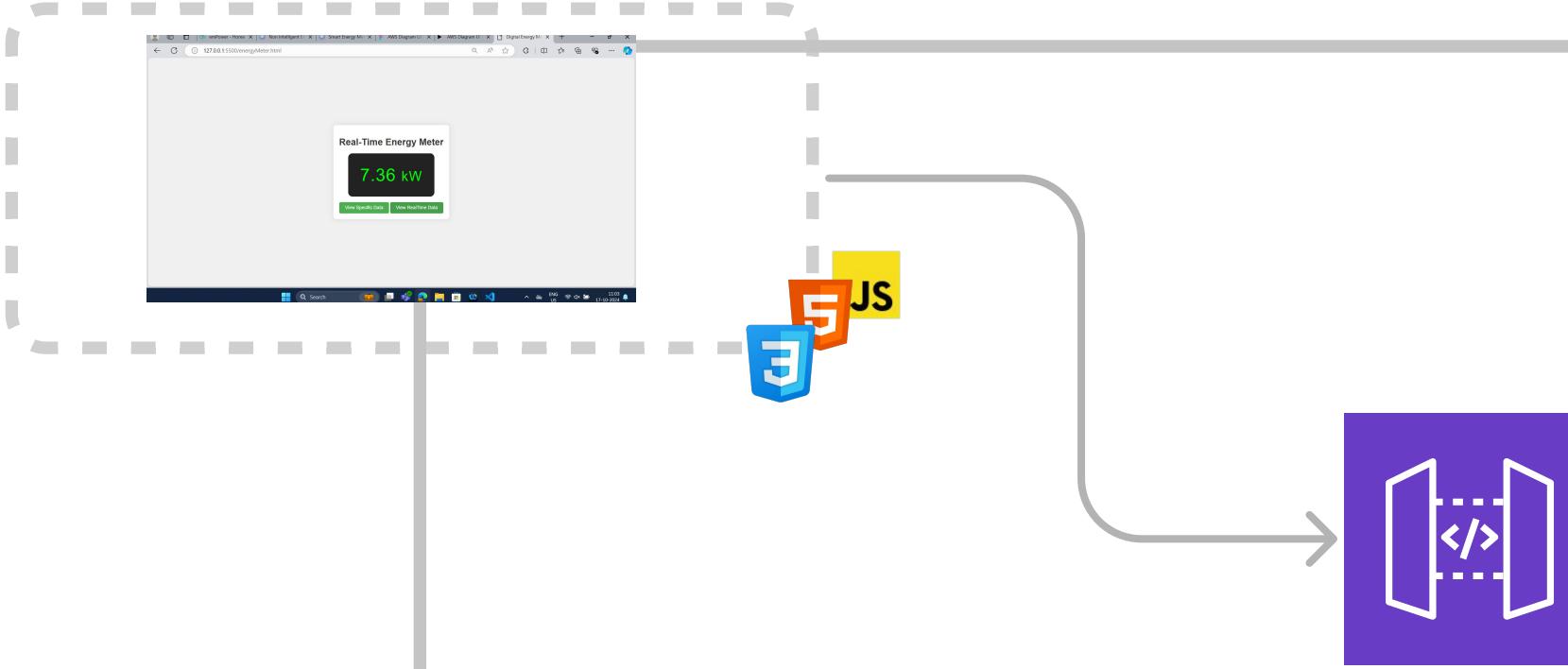


```
def lambda_handler(event, context):
    try:
        # Get the body of the request
        body = json.loads(event['body'])

        # Extract start and end time from the body
        start_time_str = body['startTime']
        end_time_str = body['endTime']
        appliance_name = body['applianceName'] # Extract appliance name
```

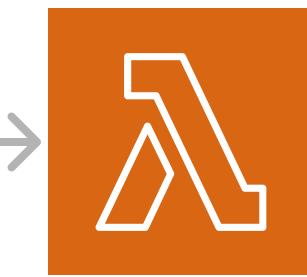
Frontend Application

DashBoard View



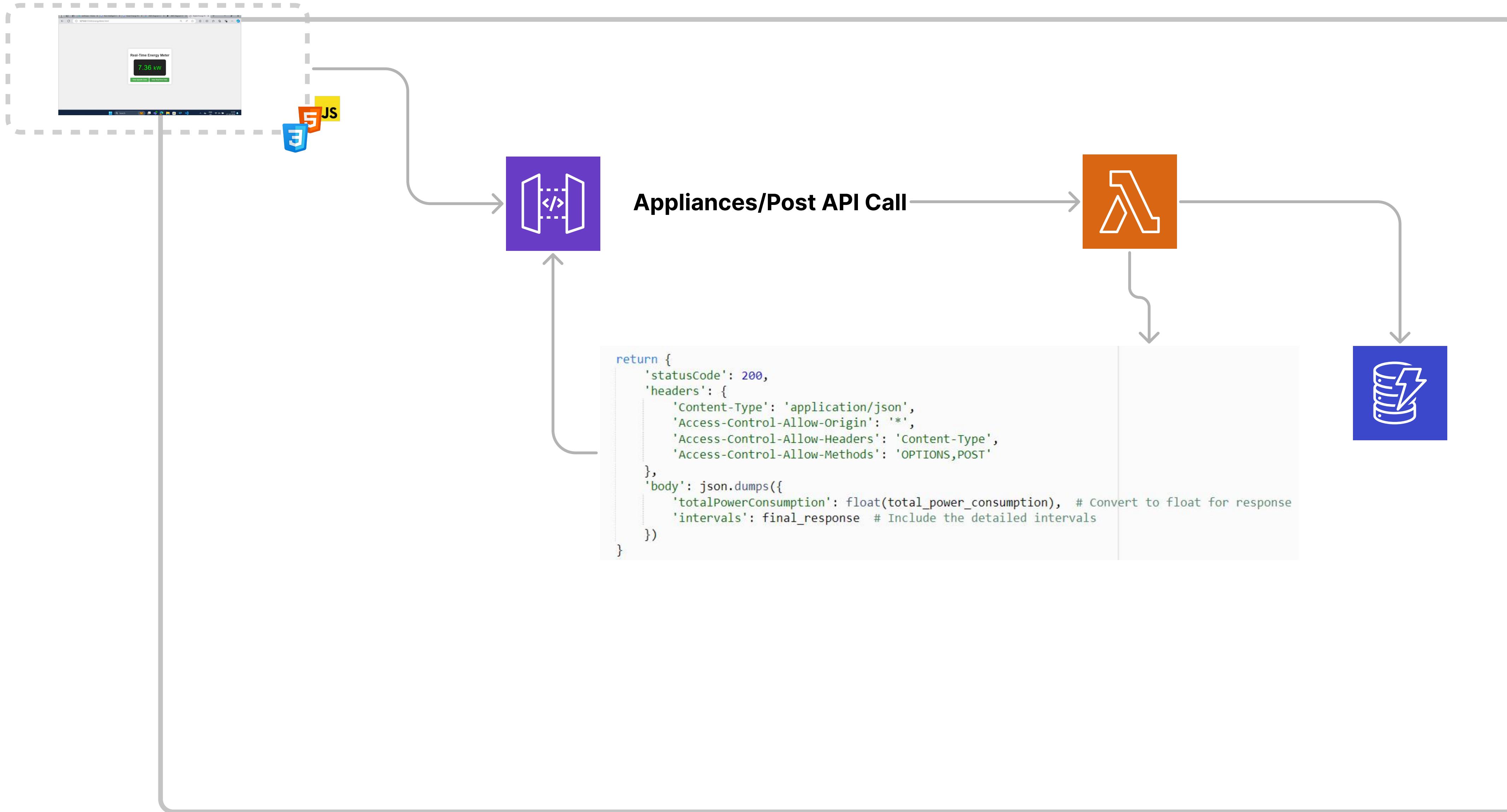
Frontend Application

Appliances/Post API Call



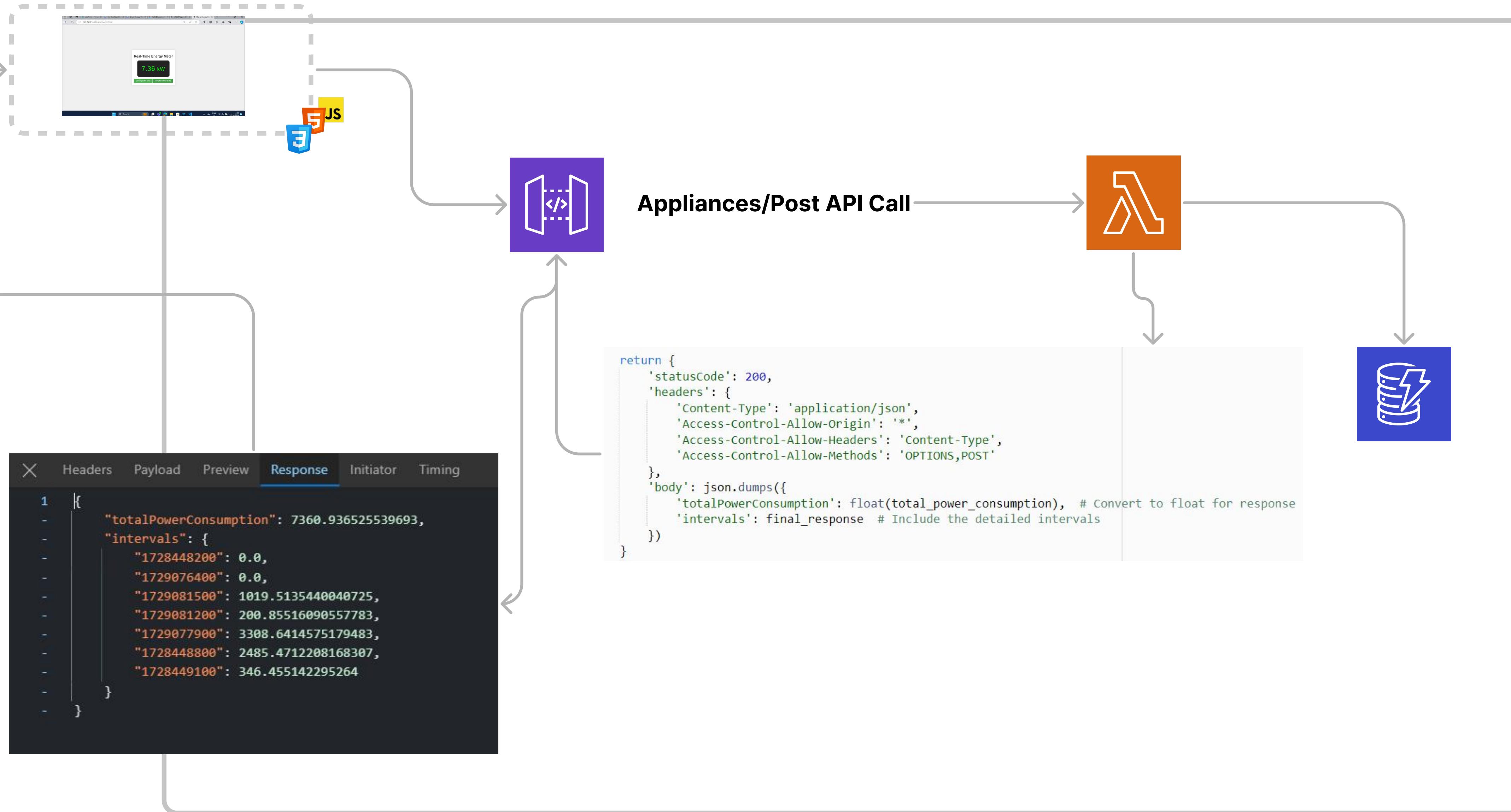
Dashboard View

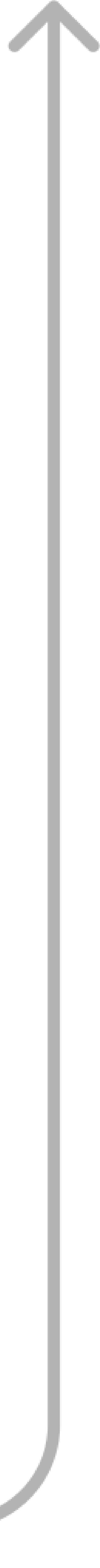
Frontend Application



Dashboard View

Frontend Application





Dashboard

Start Time:

04 - 10 - 2024 12 : 35

End Time:

26 - 10 - 2024 12 : 35

Appliance Name:

ALL

[View Dashboard](#)

Power Consumption Chart

Total Power Consumed (kWh)

Time Interval	Total Power Consumed (kWh)
10:00:00 am	0
10:10:00 am	~2,500
10:15:00 am	~400
4:30:00 pm	0
4:55:00 pm	~3,300
5:50:00 pm	~150
5:55:00 pm	~1,000

Total Power Consumed (kWh)

Time (5-minute intervals)

IOT Devices

Central Processing Unit

DataStorage

Energy Meter

Dashboards

IOT Devices

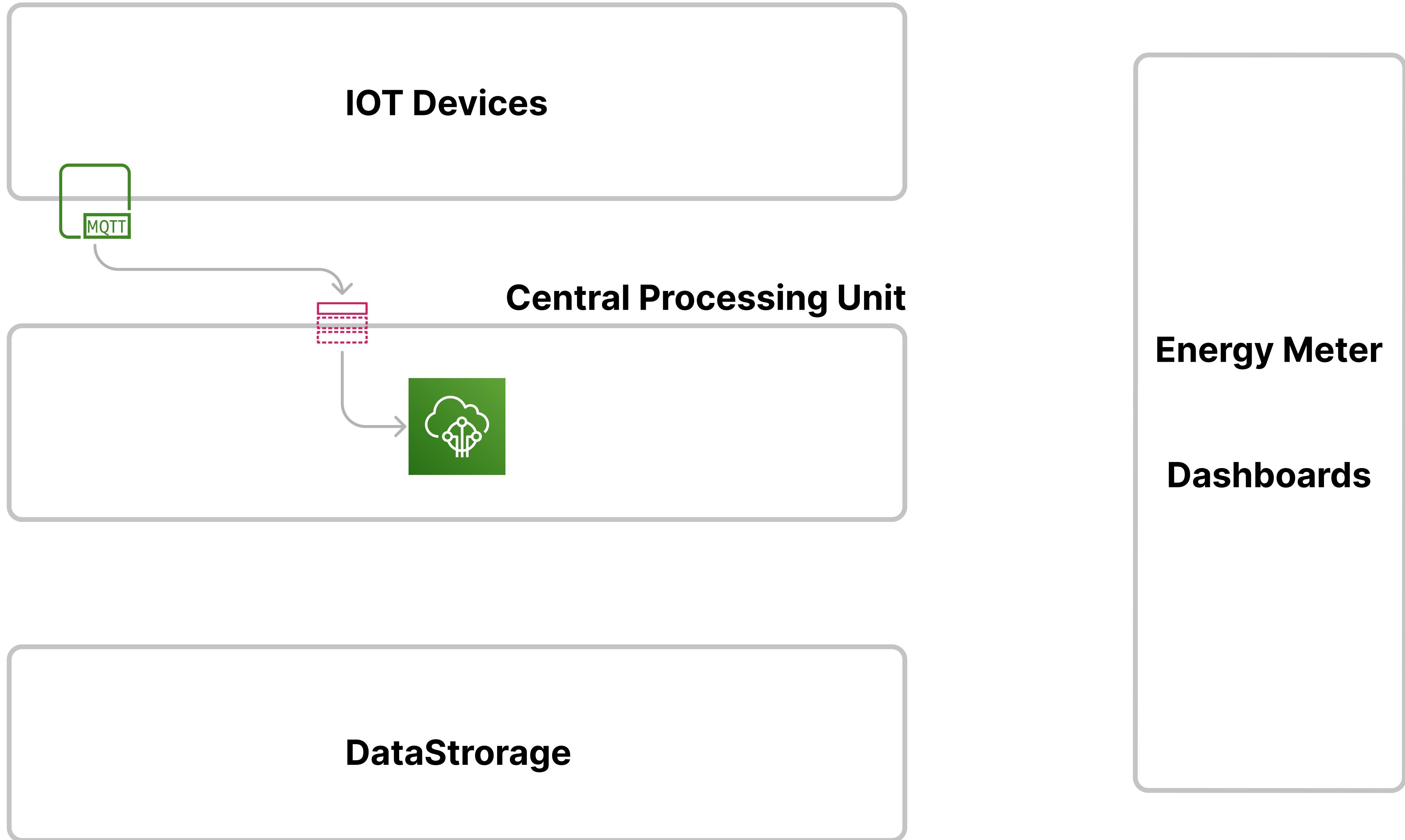
Central Processing Unit

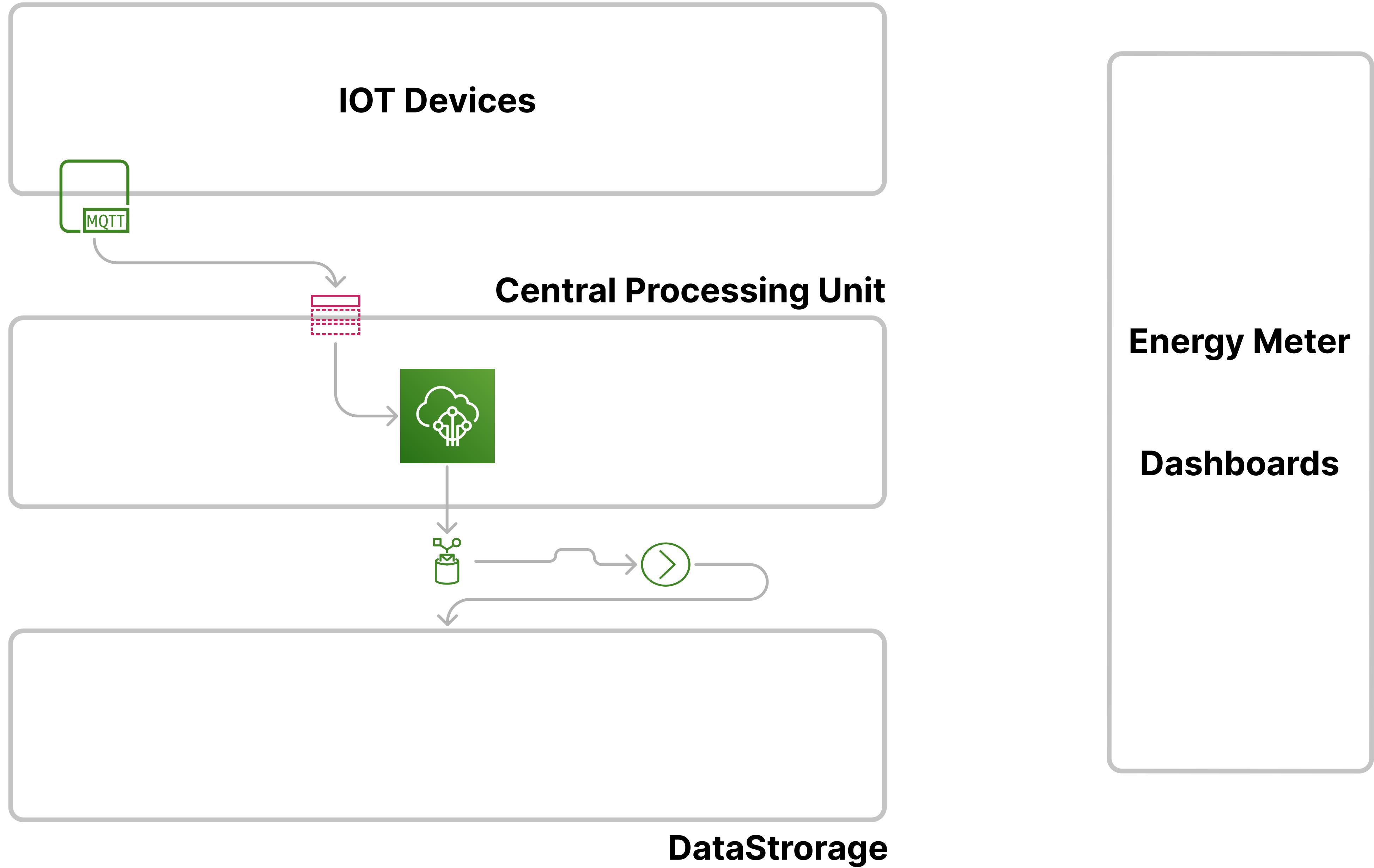


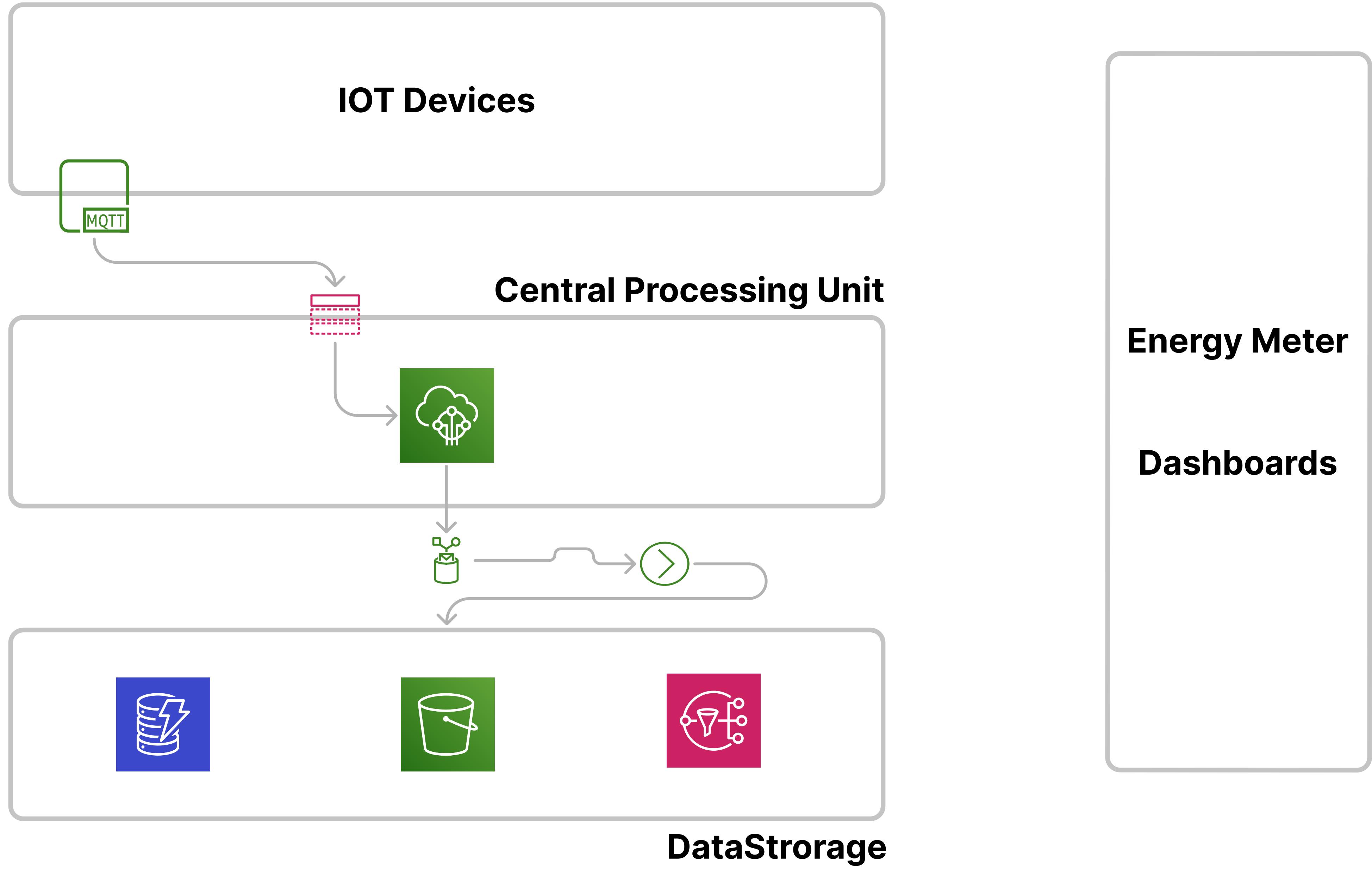
DataStorage

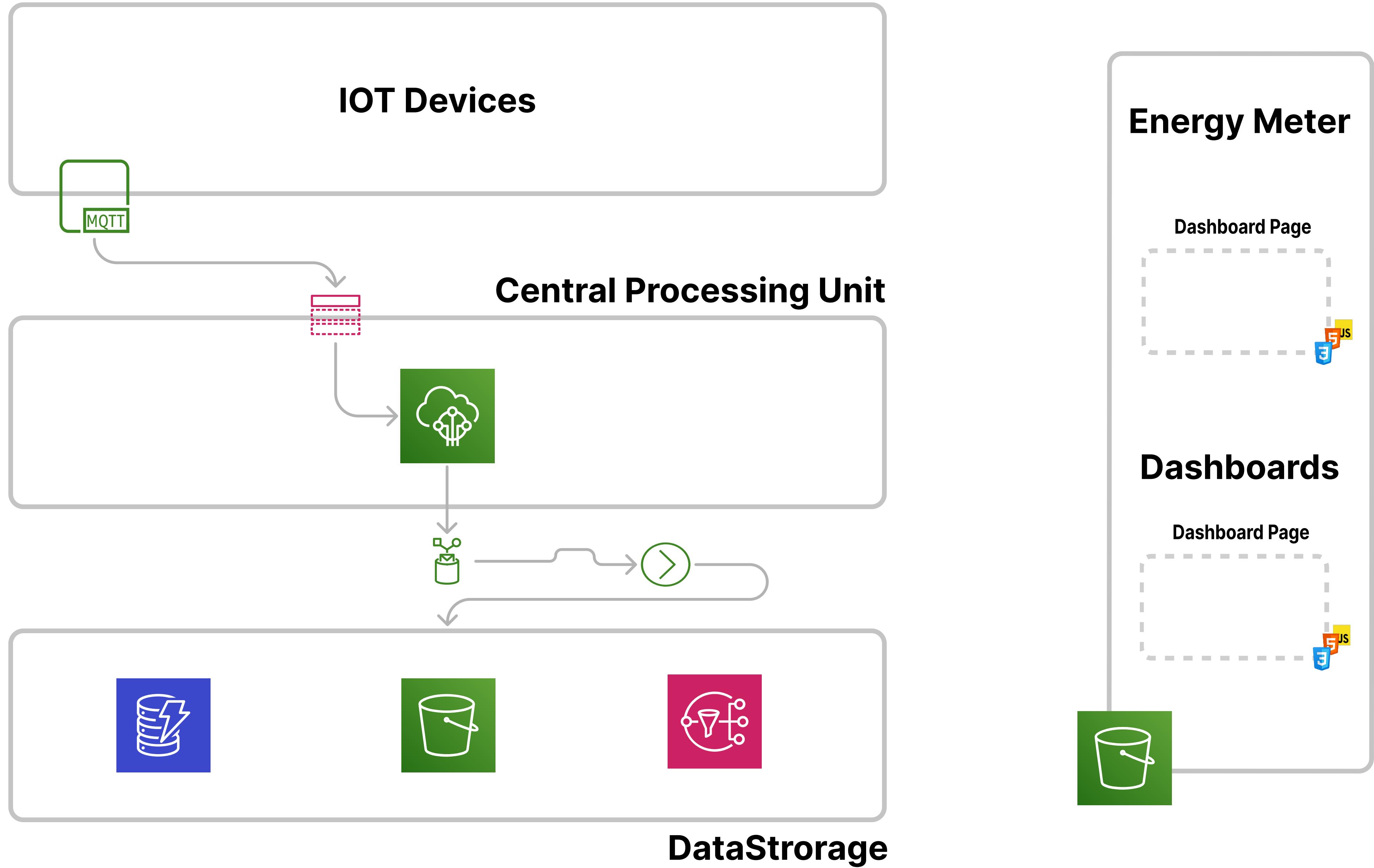
Energy Meter

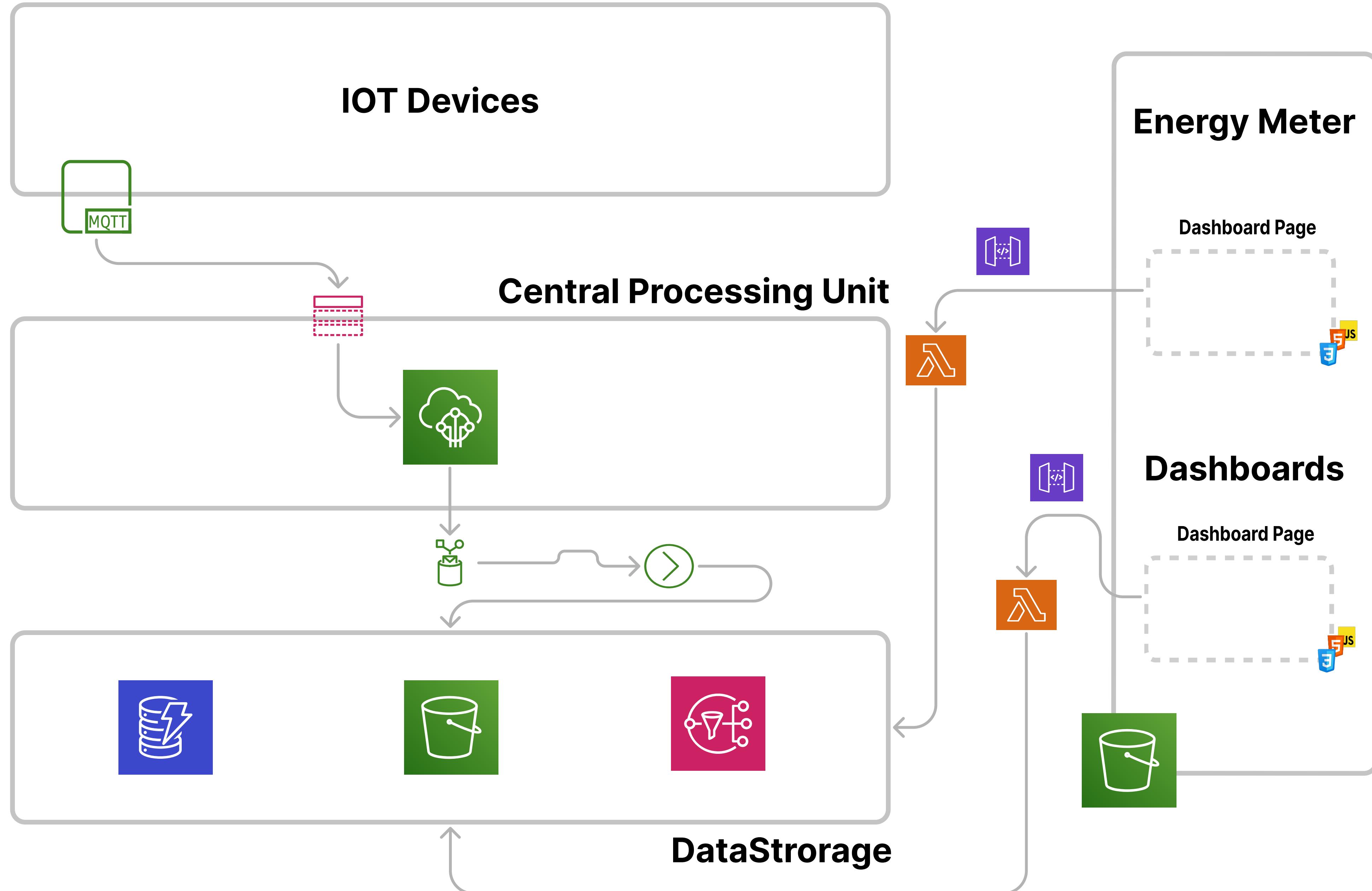
Dashboards

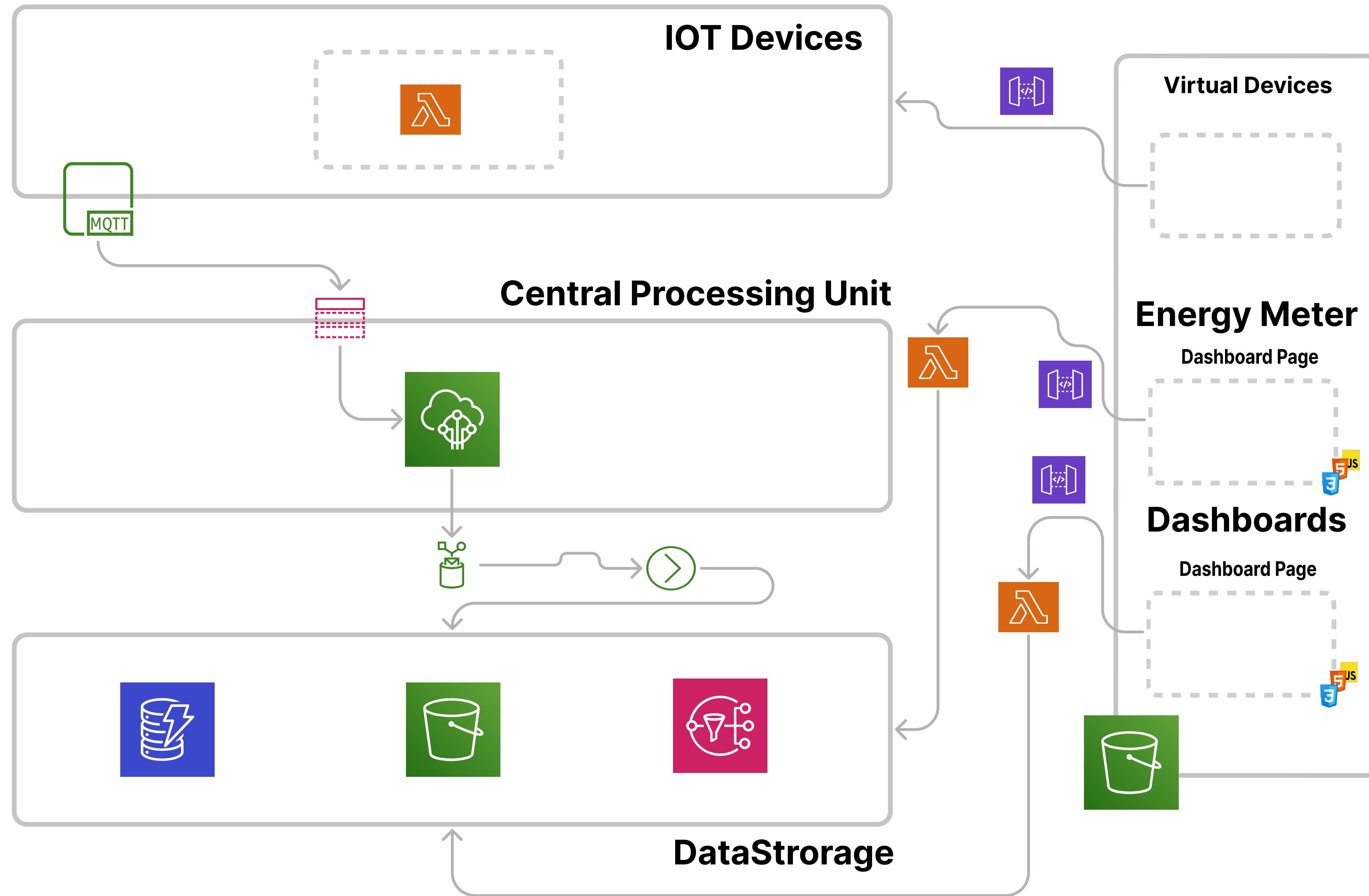


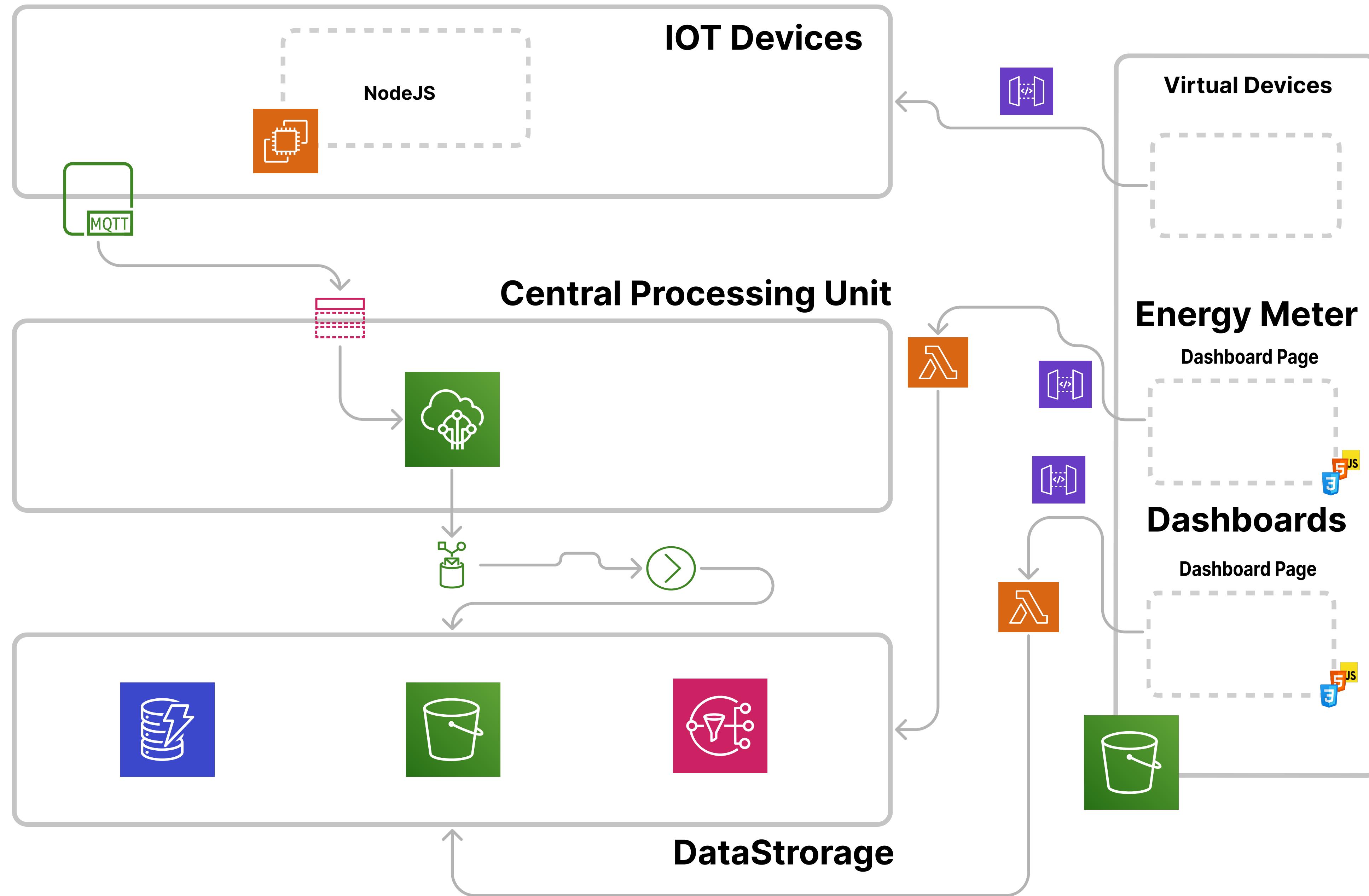


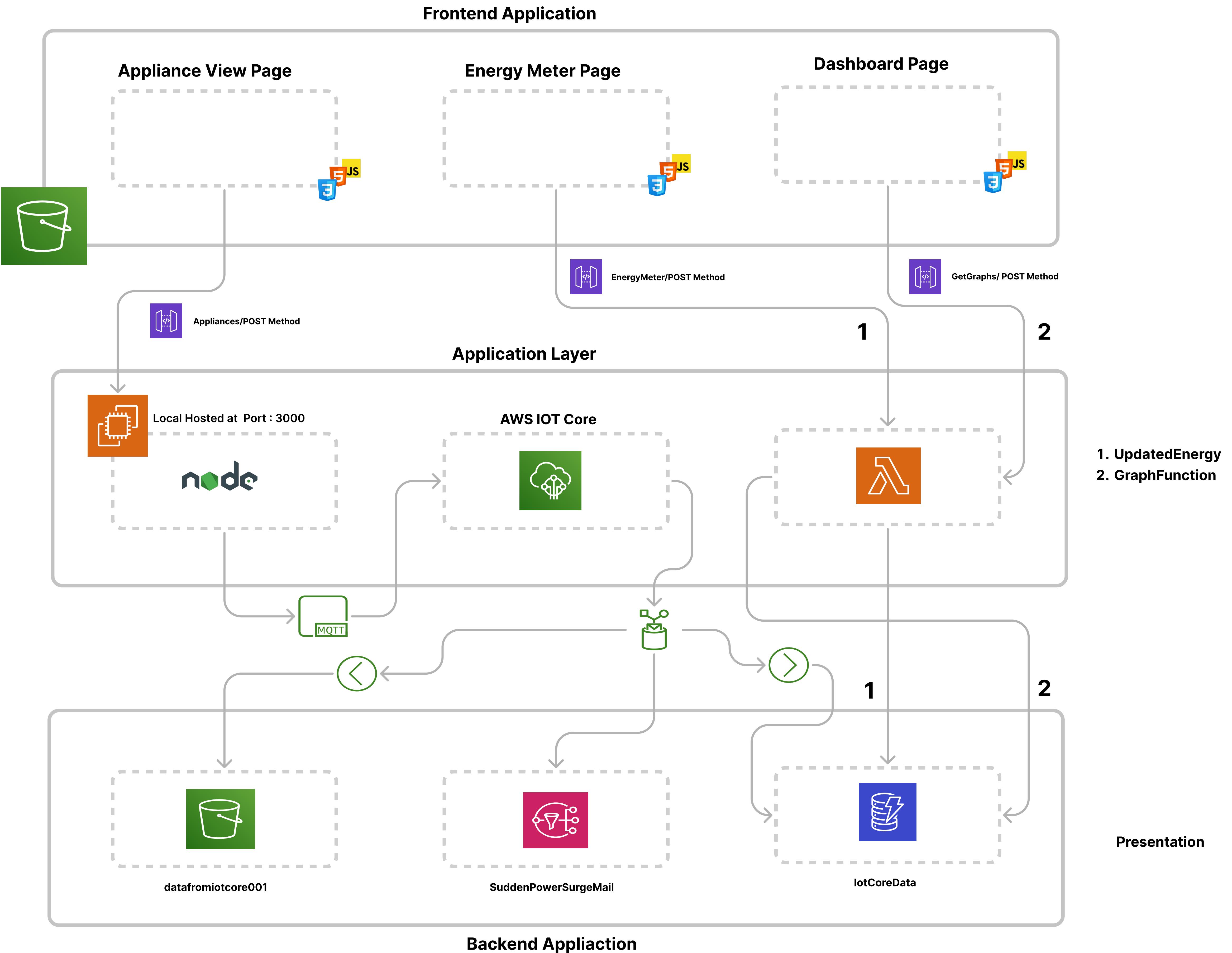


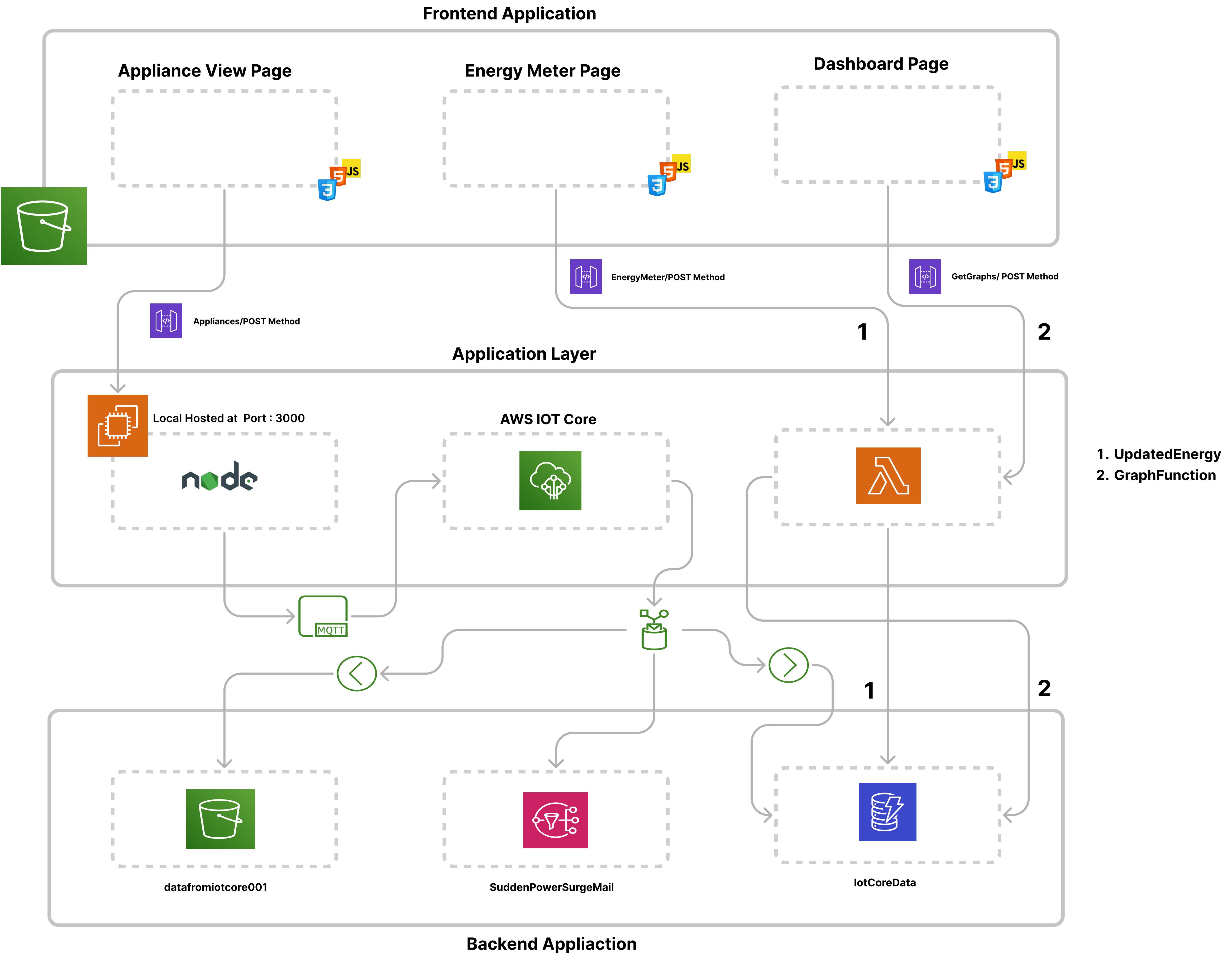


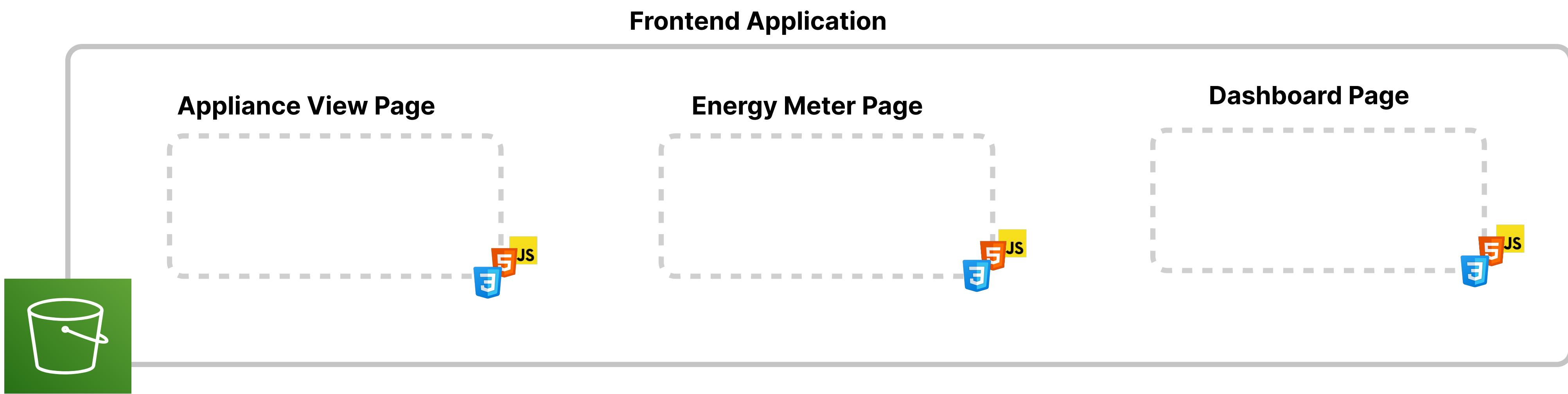












Smart Energy Meter Dashboard



View Appliances

Monitor and control your appliances.



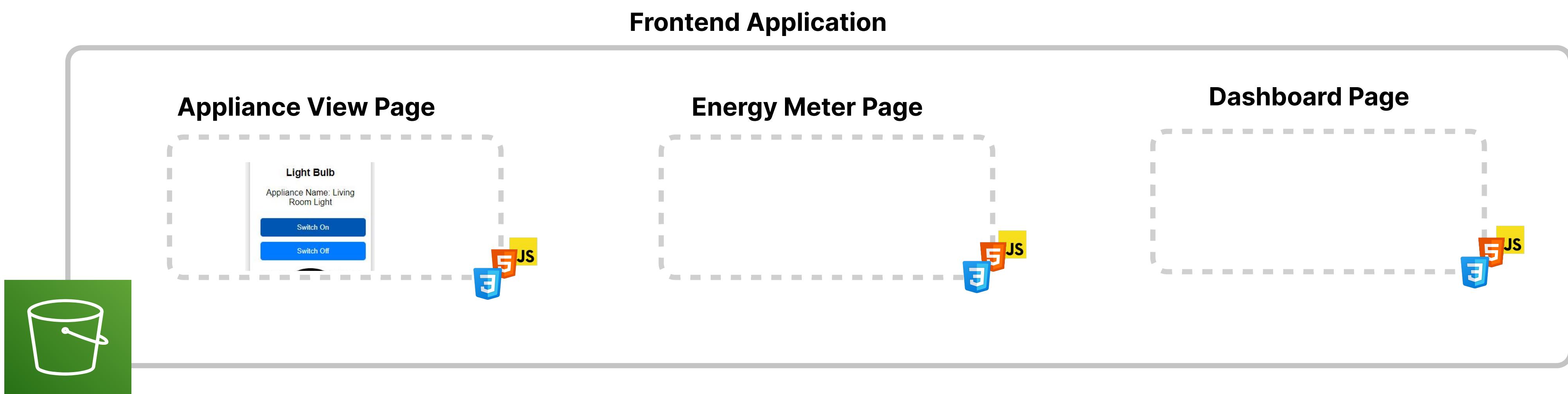
View Energy Meter

Check your energy consumption.



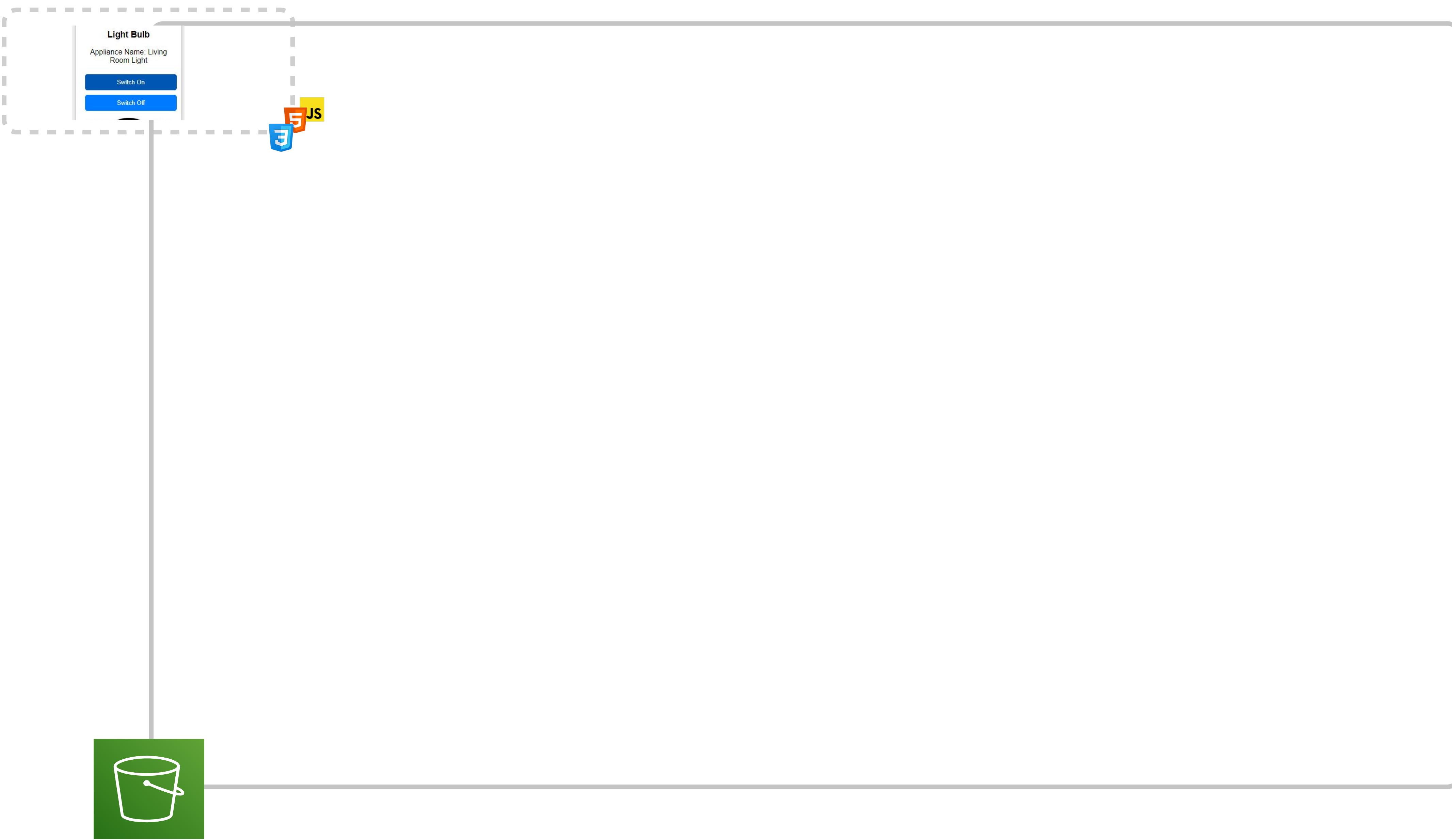
View Dashboard

Get insights on your usage.



Appliance View Page

Frontend Application



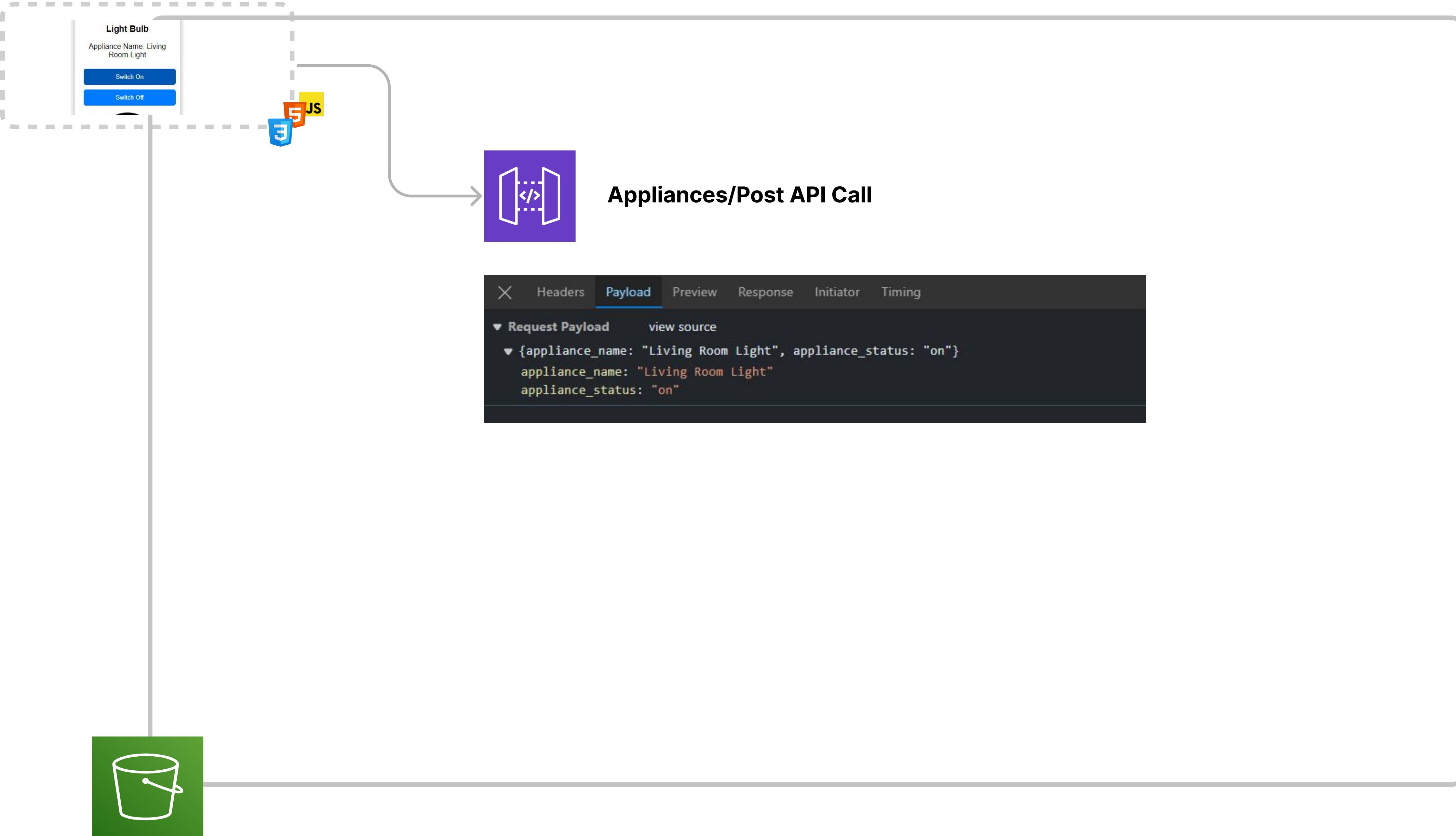
Appliance View Page

Frontend Application



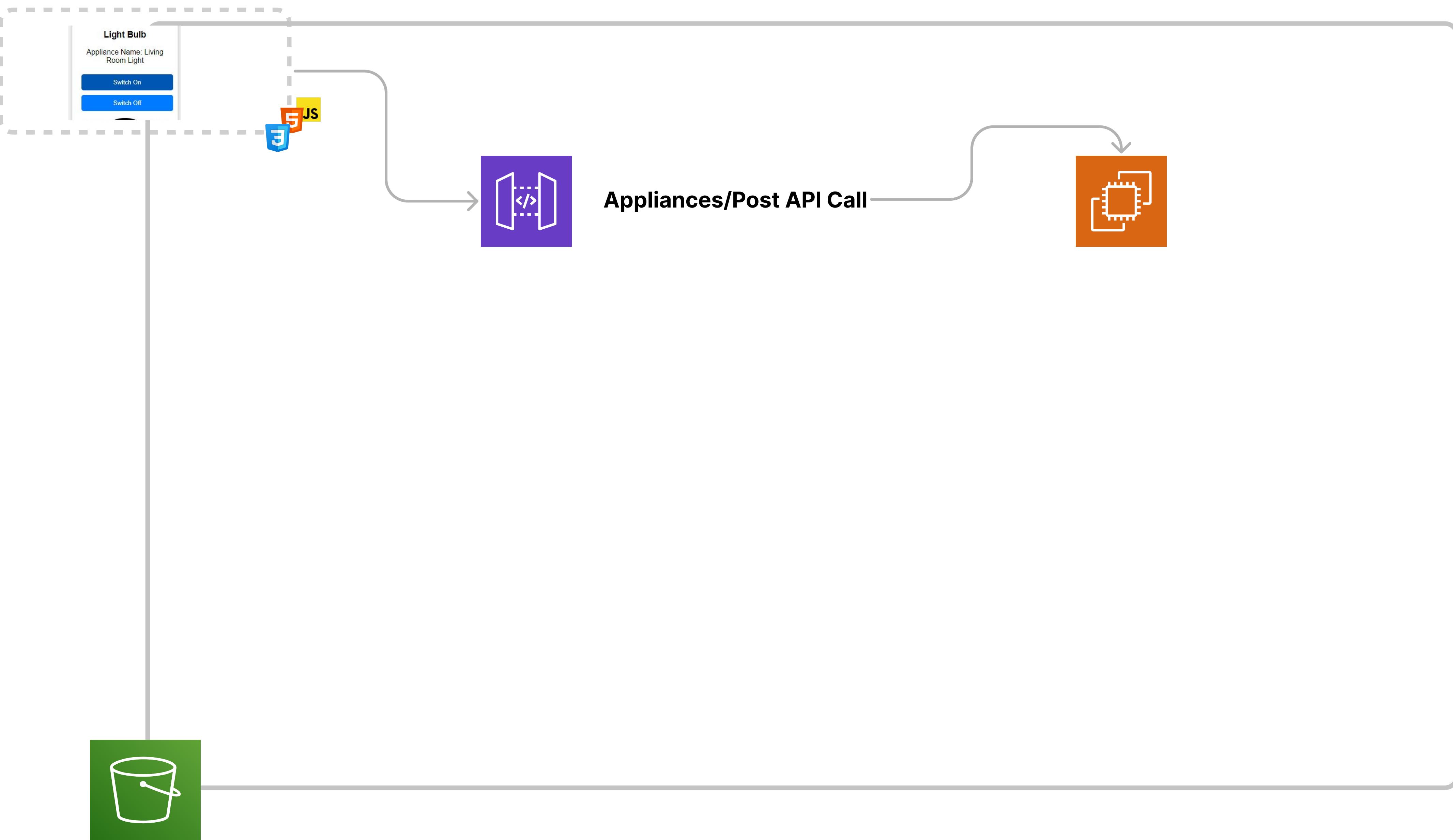
Appliance View Page

Frontend Application



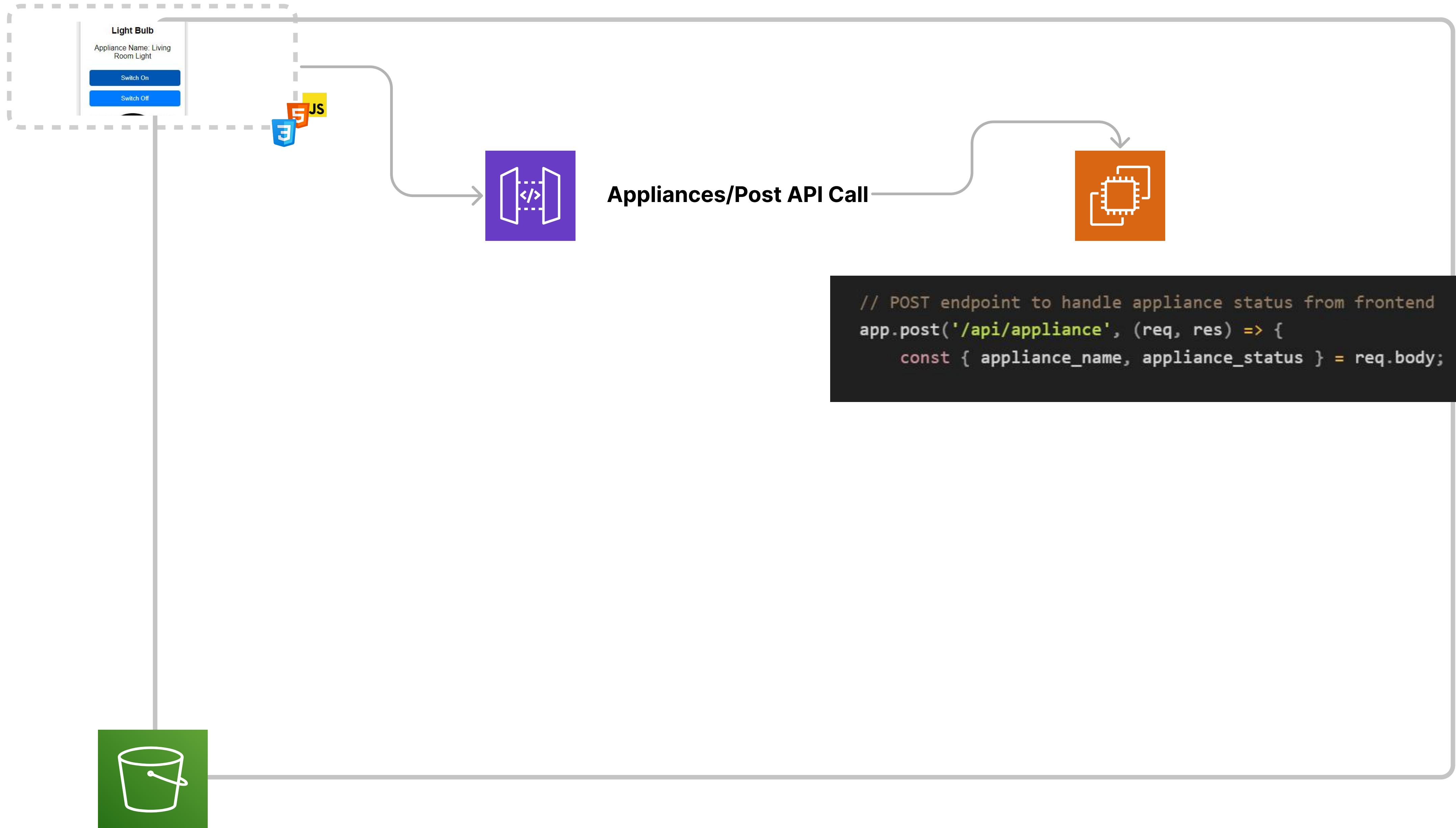
Appliance View Page

Frontend Application



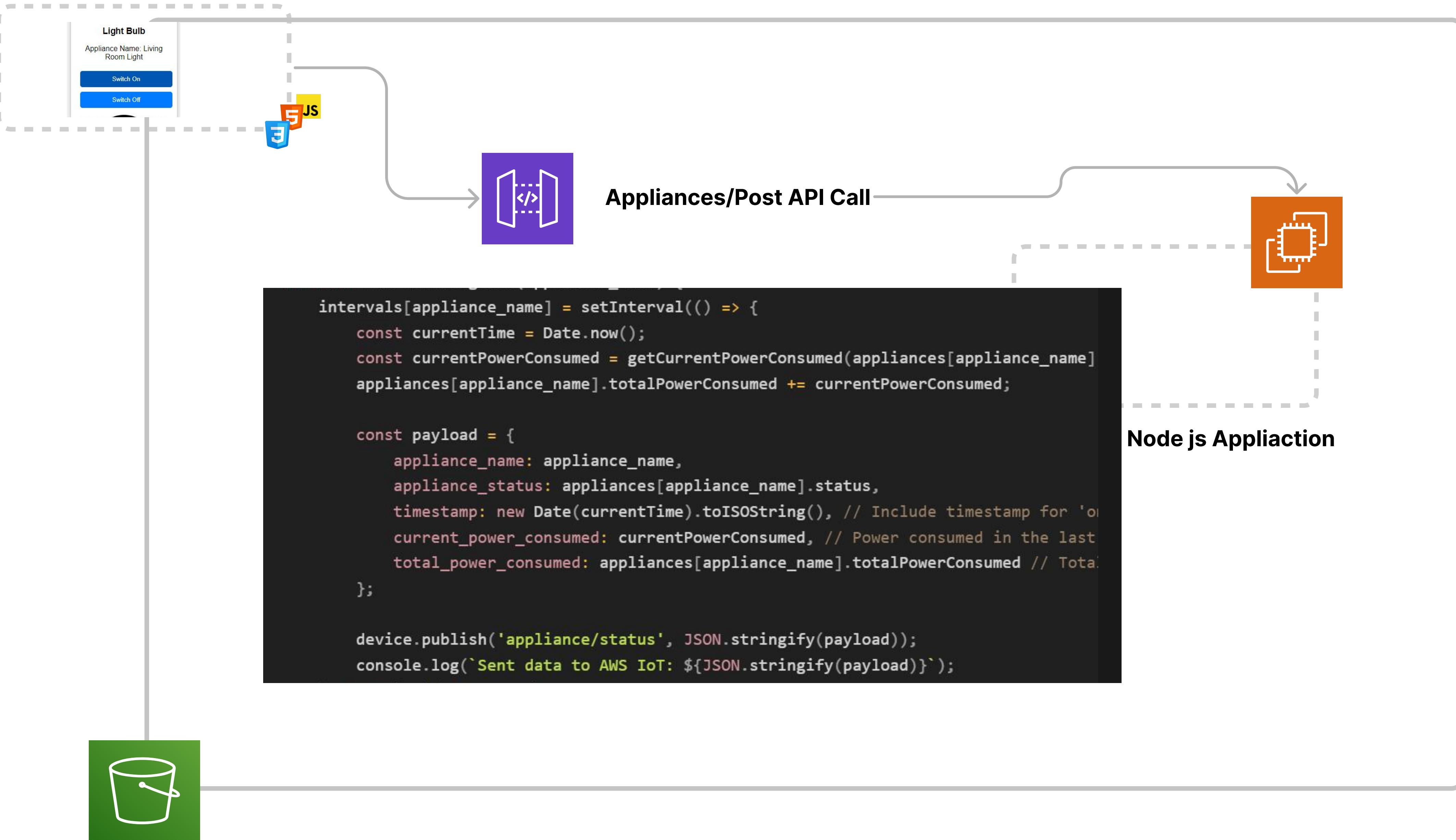
Appliance View Page

Frontend Application



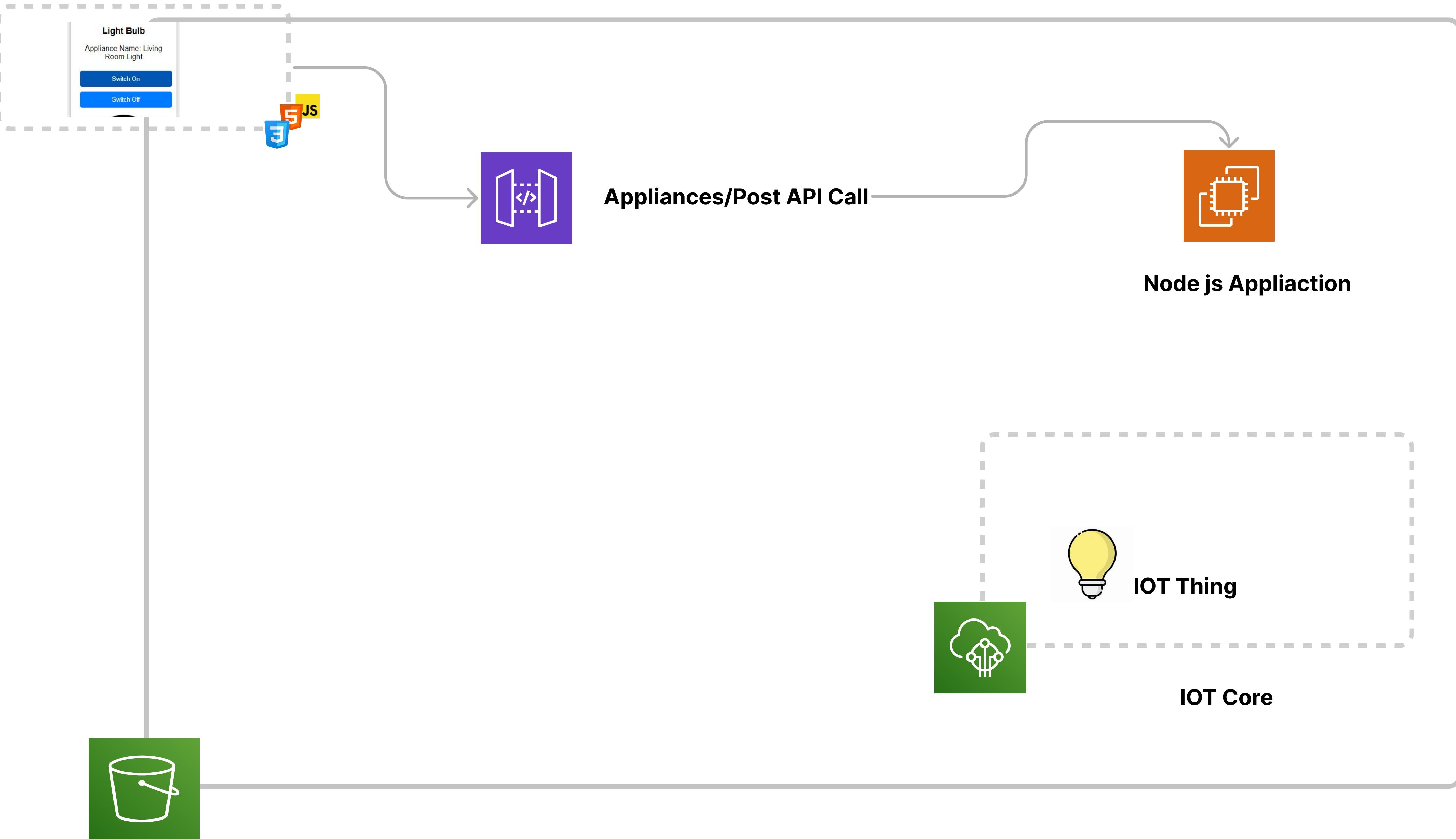
Appliance View Page

Frontend Application



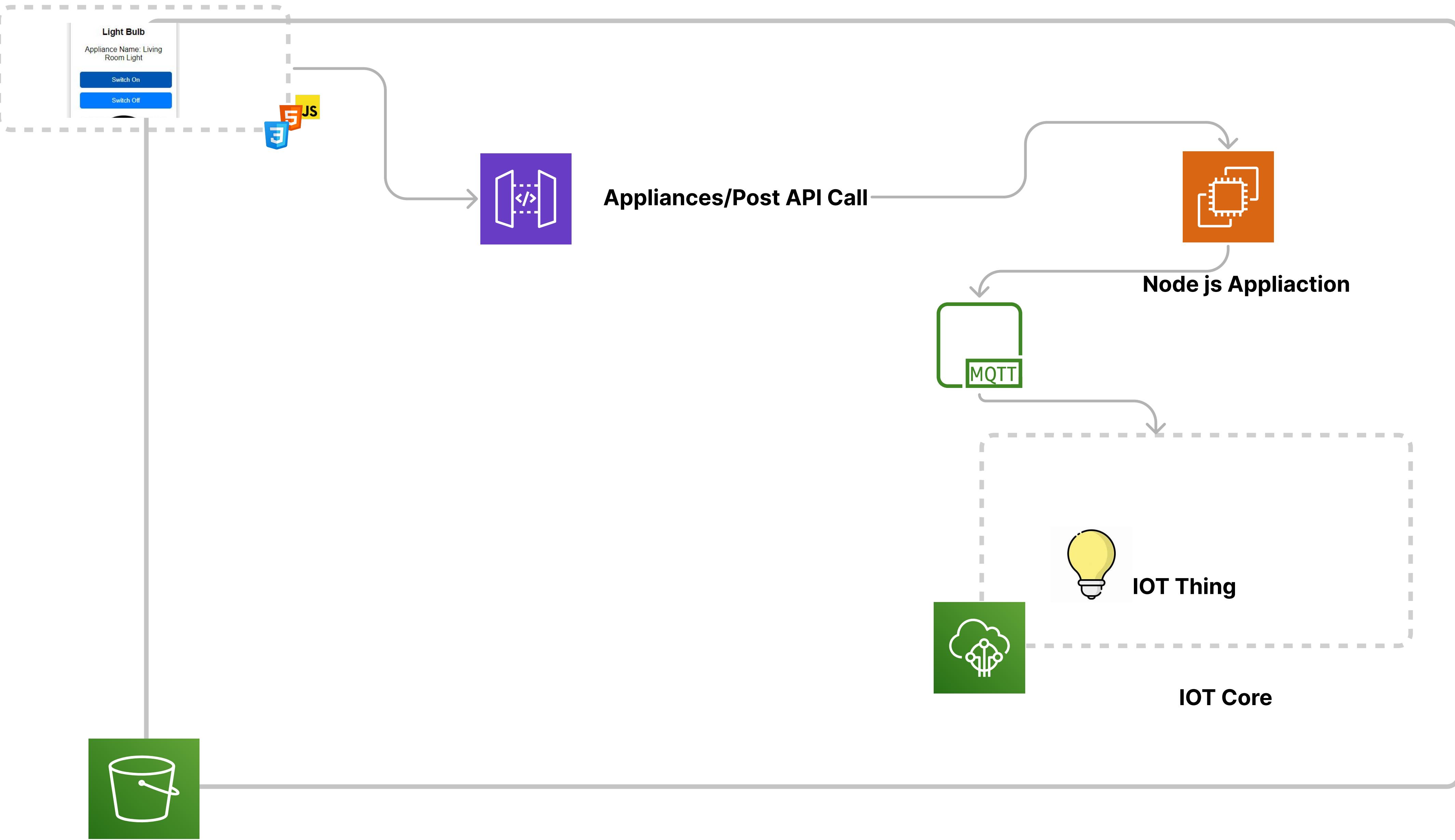
Appliance View Page

Frontend Application



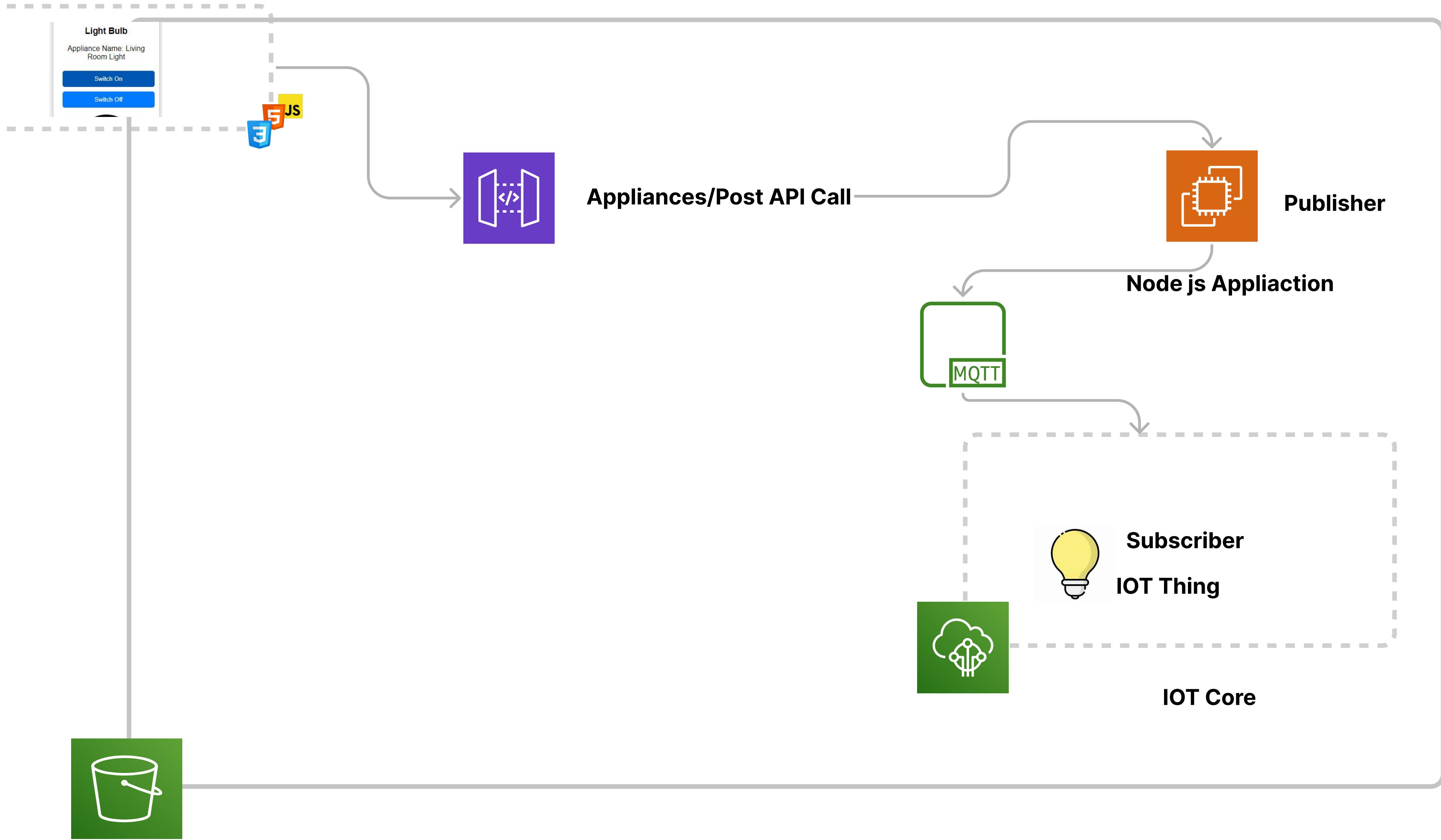
Appliance View Page

Frontend Application



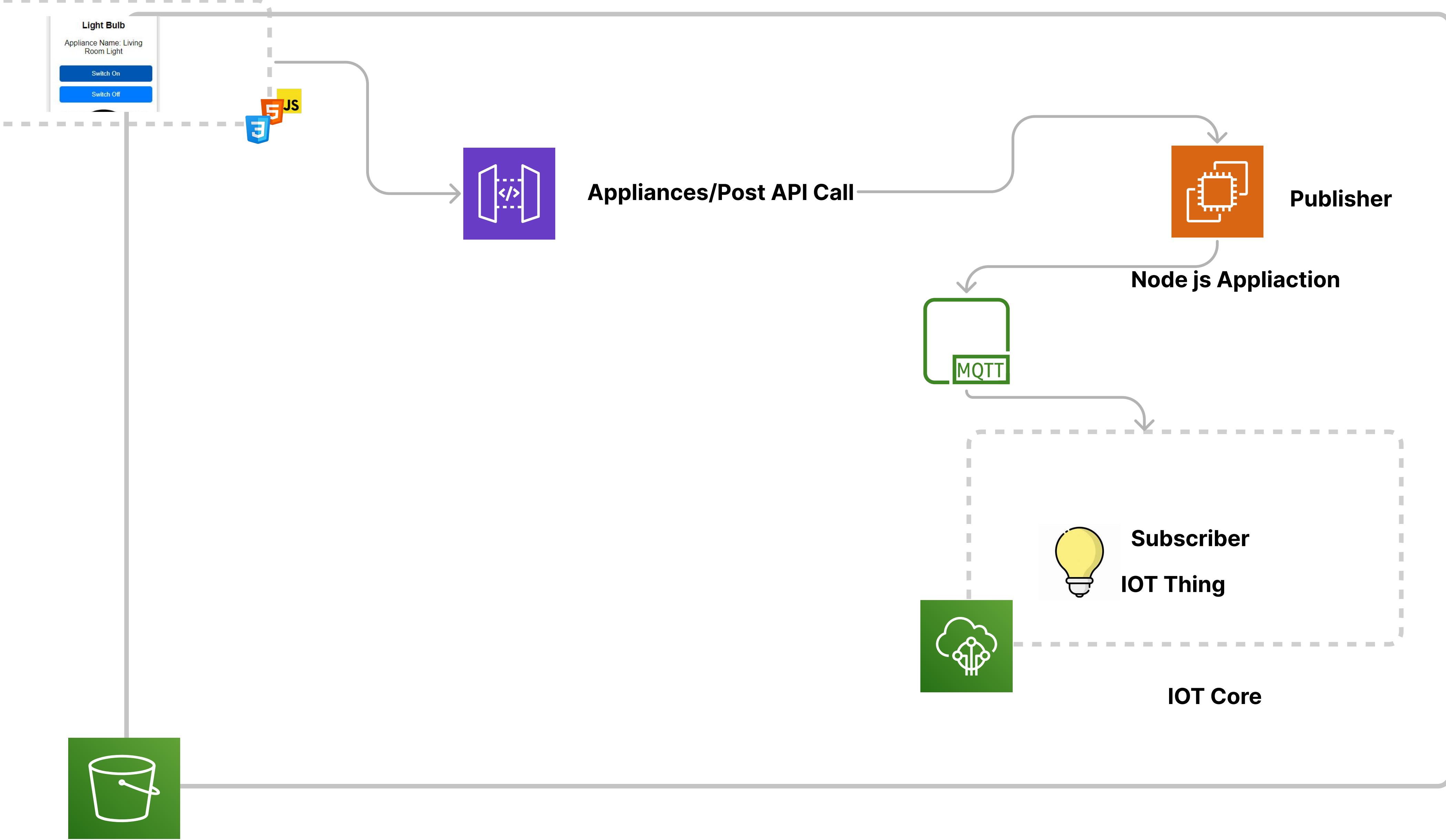
Appliance View Page

Frontend Application



Appliance View Page

Frontend Application



Dashboard

Start Time:

12-09-2024 11:0

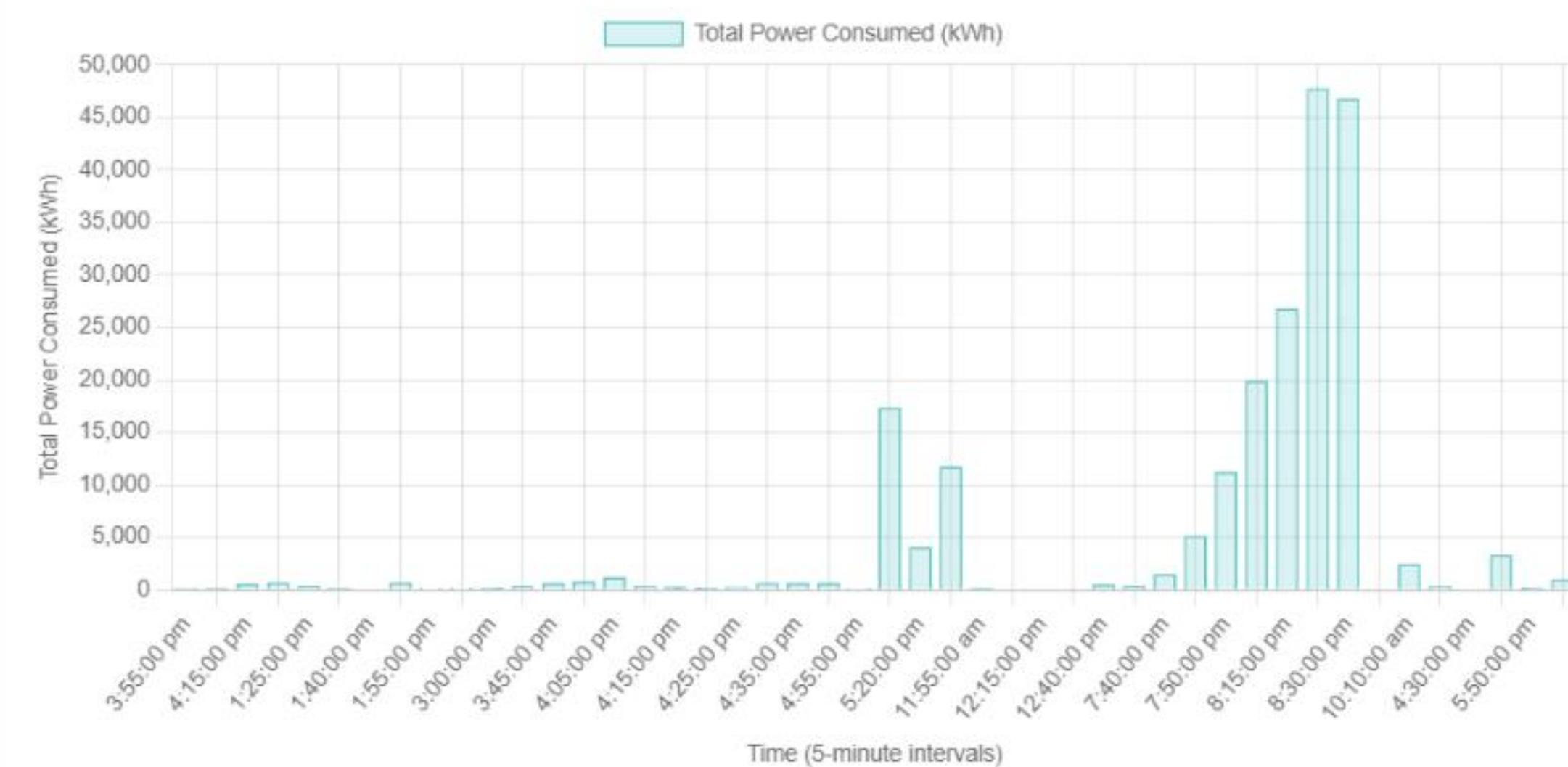
End Time:

Appliance Name

All

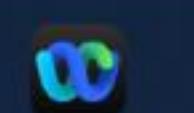
[View Dashboard](#)

Power Consumption Chart

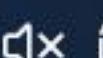




Search



ENG
US



11:03
17-10-2024

Dashboard

Start Time

12-09-2024 11:0

End Time:

Appliance Name

ALL

[View Dashboard](#)

Power Consumption Chart

