

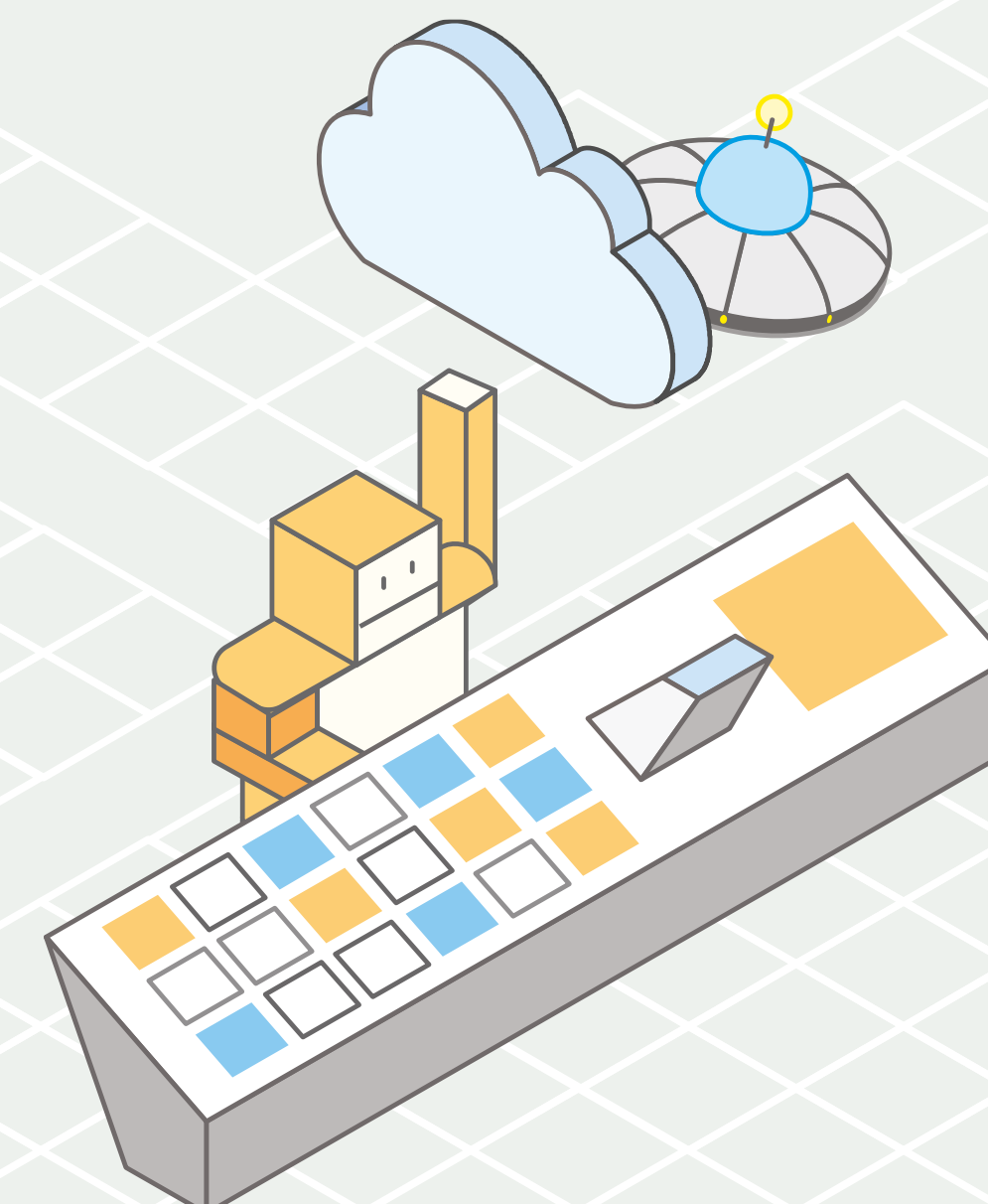


Pop-up Loft  
**LONDON**

# What is happening at the Edge?

Danilo Poccia, Technical Evangelist

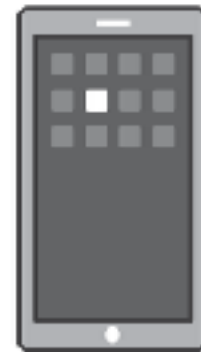
 @danilop

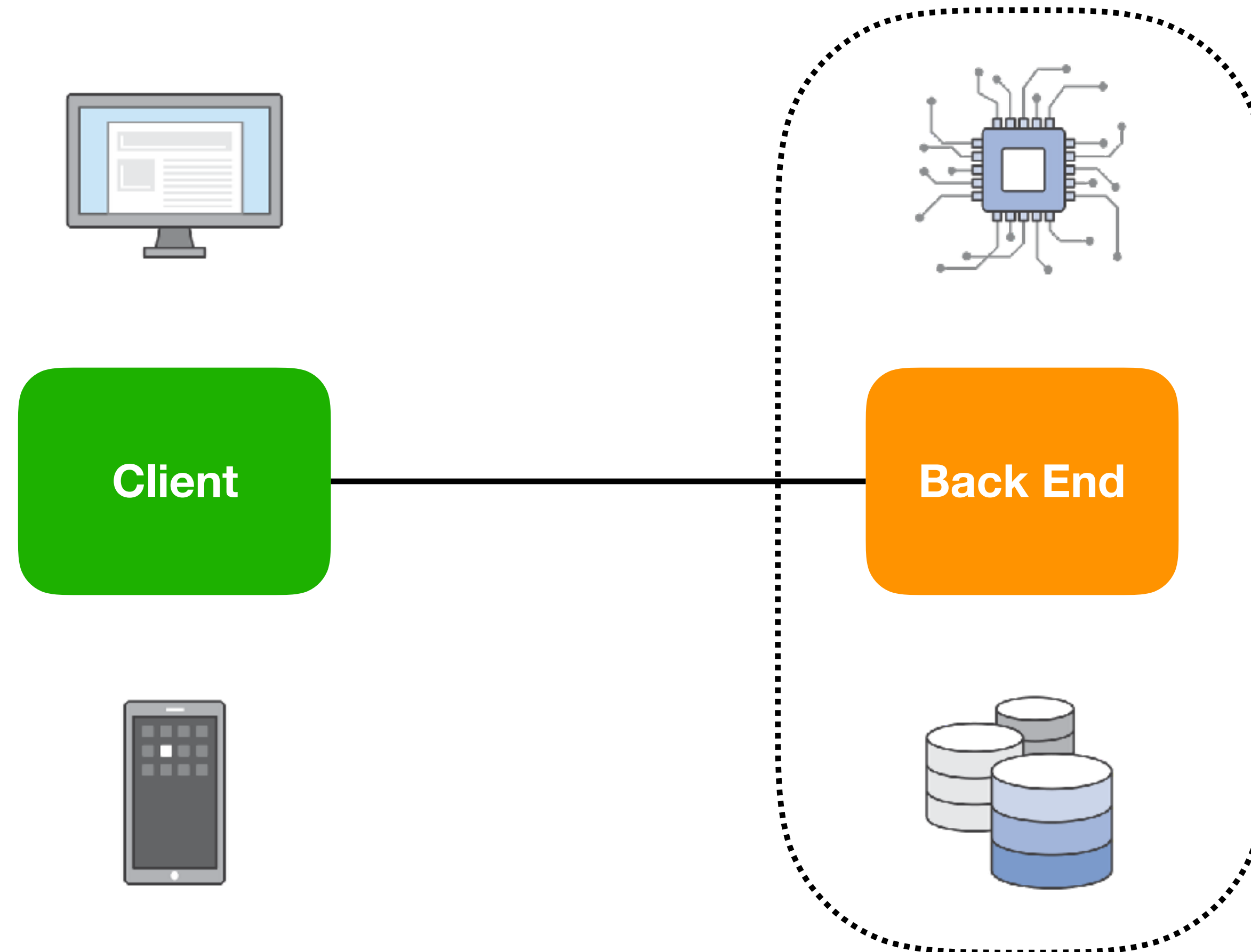


**How are IT architectures evolving?**

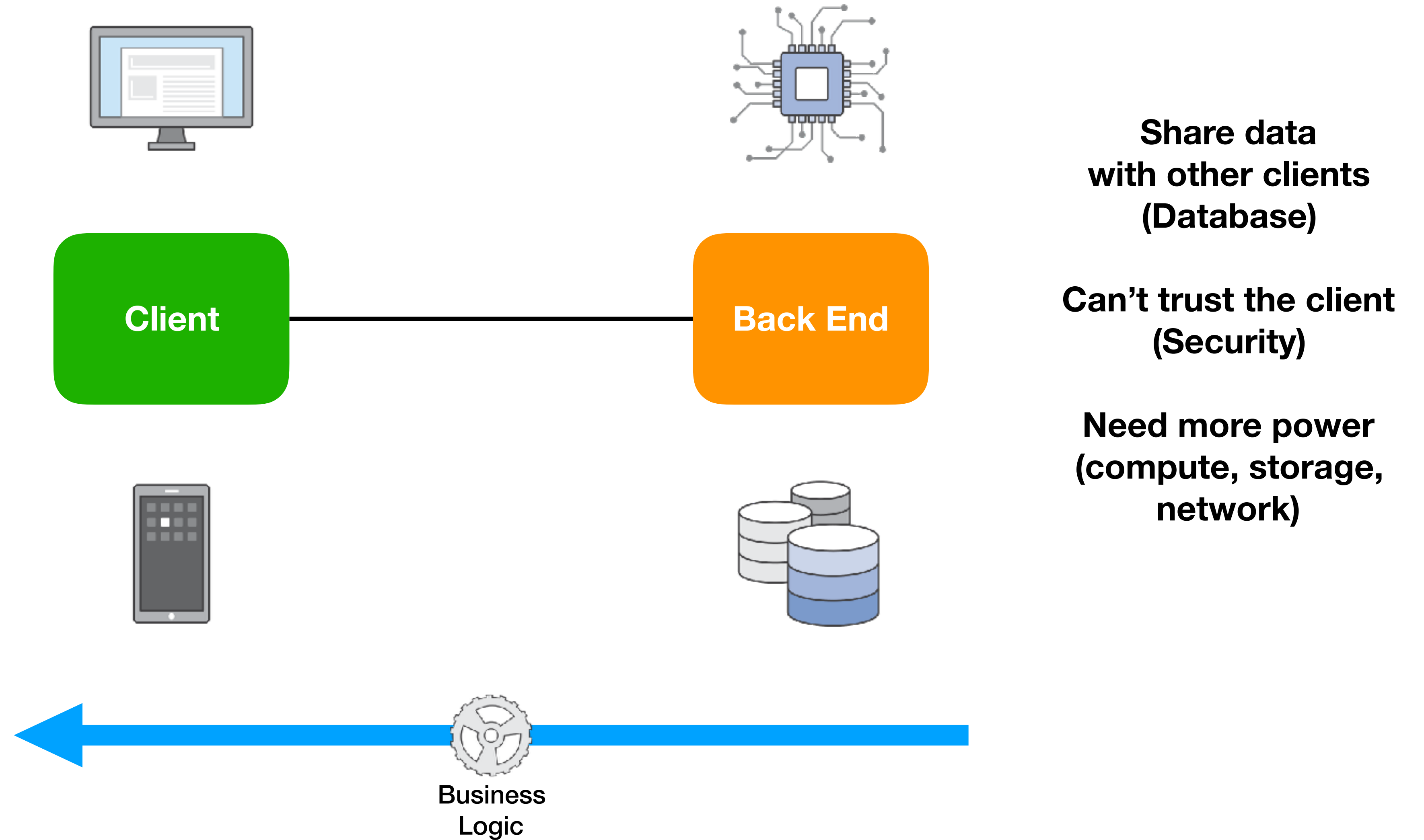


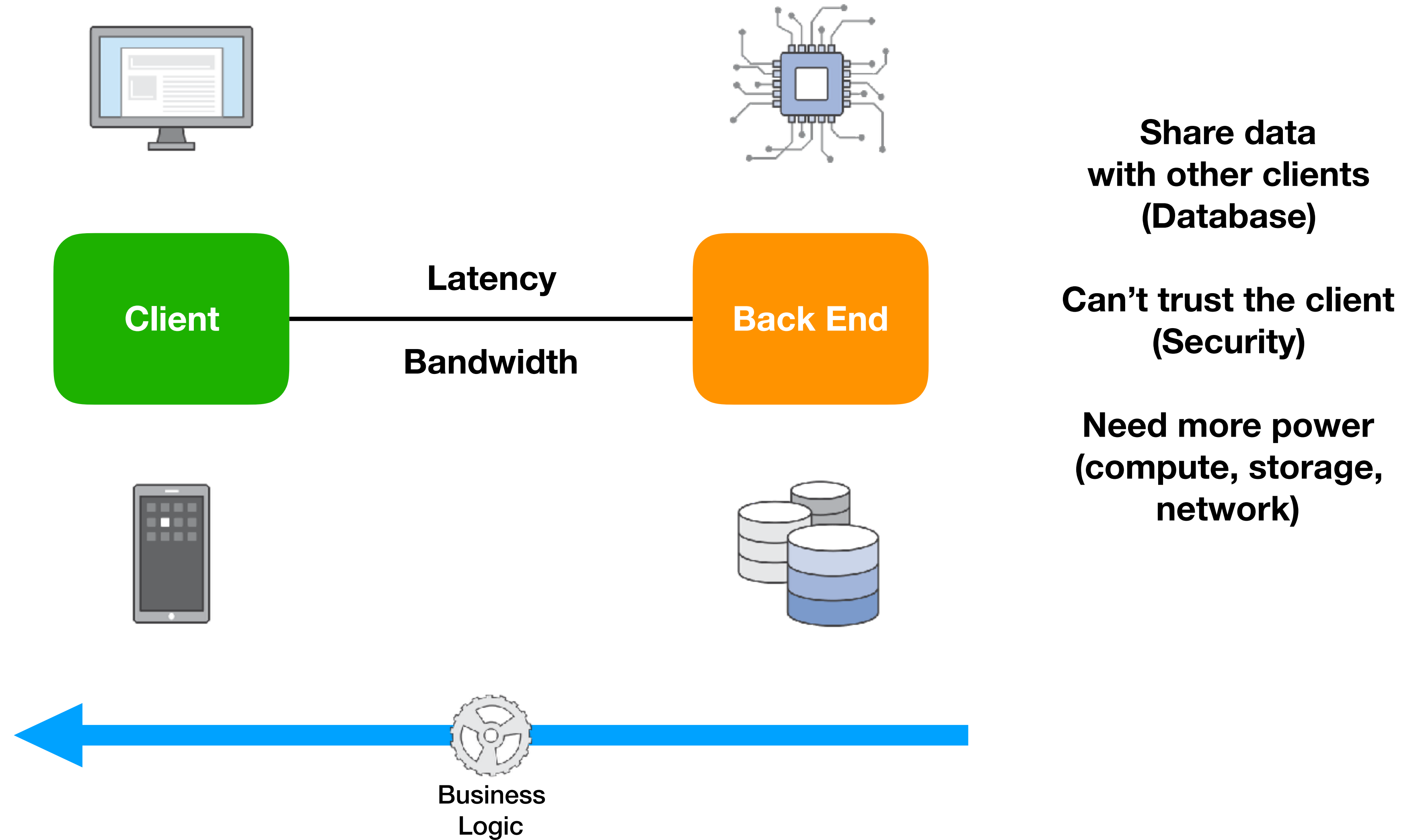
**Client**



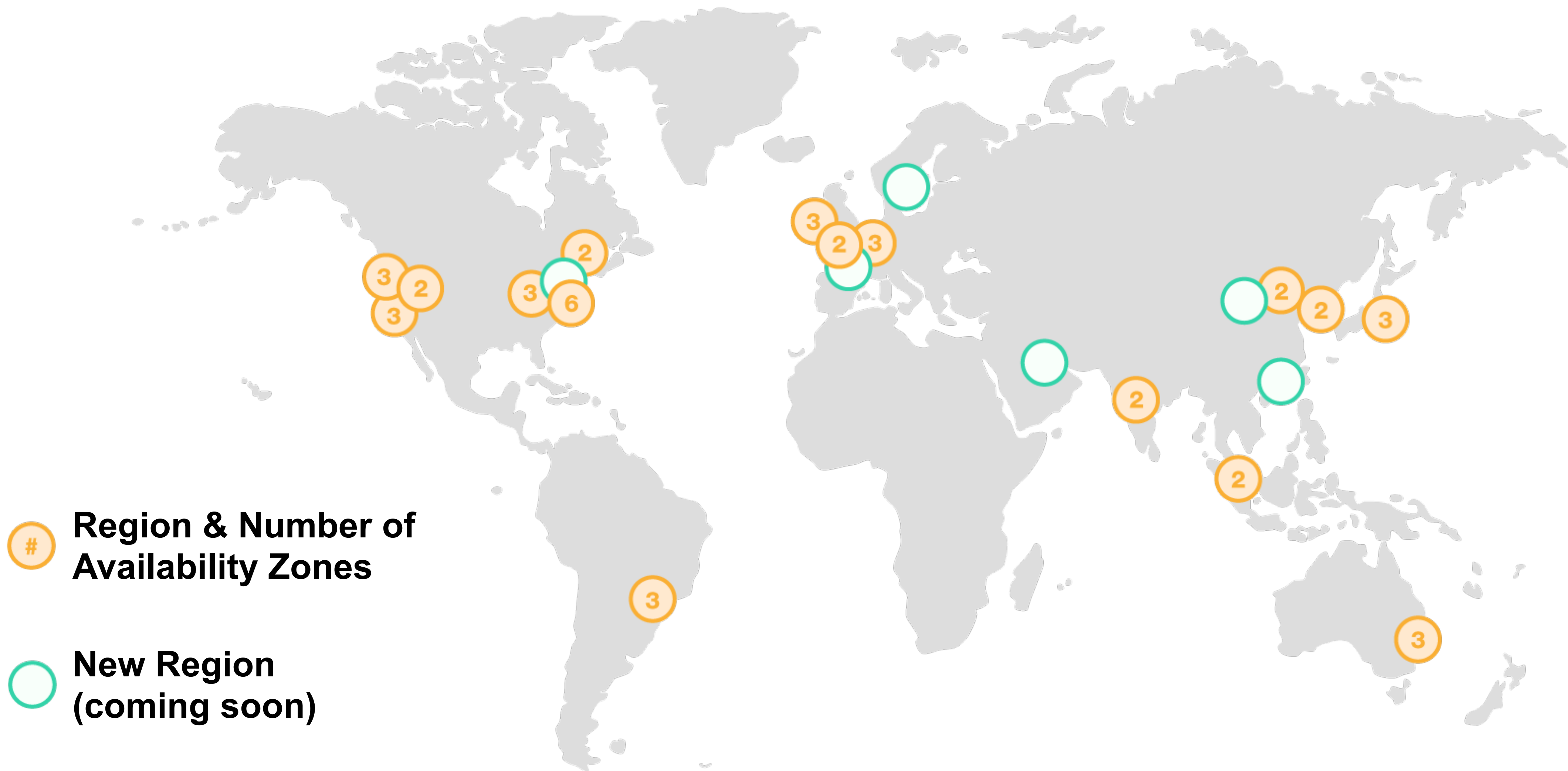


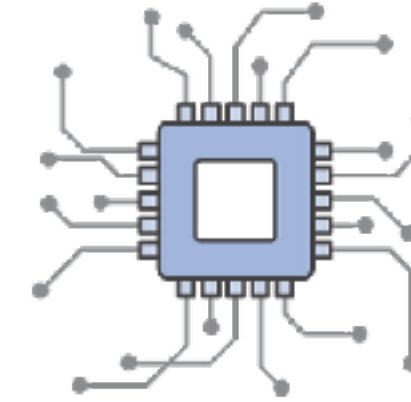
- Share data with other clients (Database)**
- Can't trust the client (Security)**
- Need more power (compute, storage, network)**





# AWS Global Infrastructure

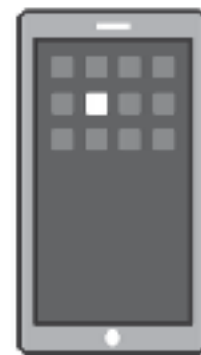




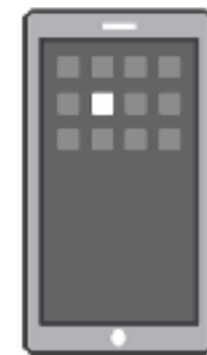
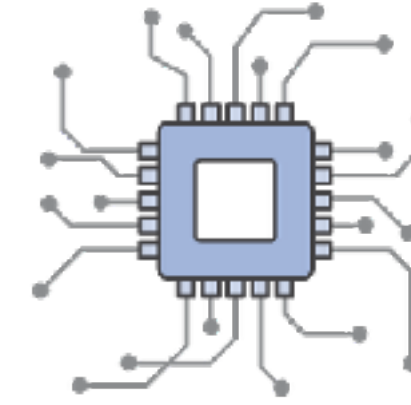
**Client**

**Latency**  
**Bandwidth**

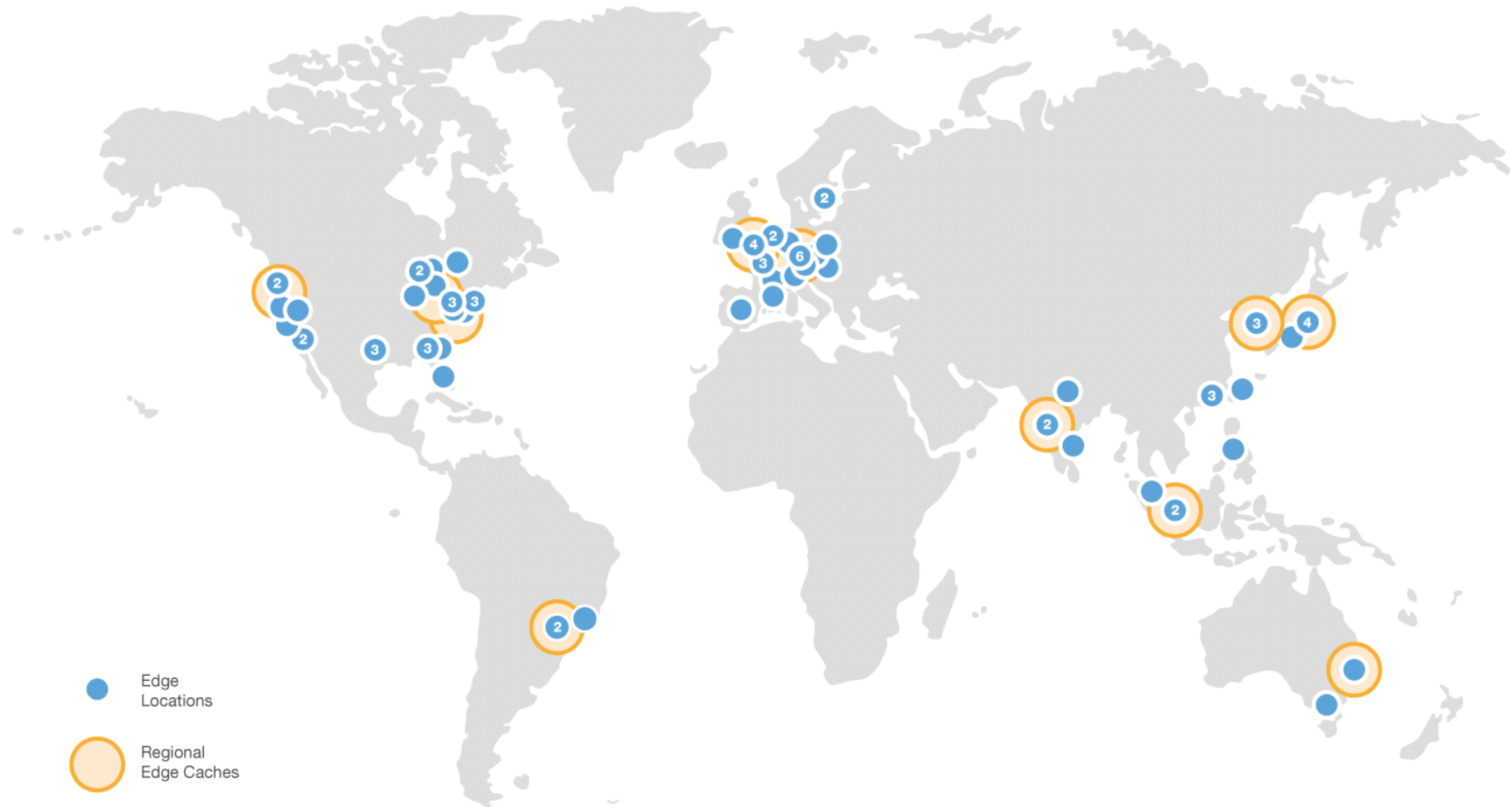
**Back End**

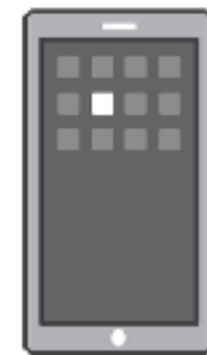
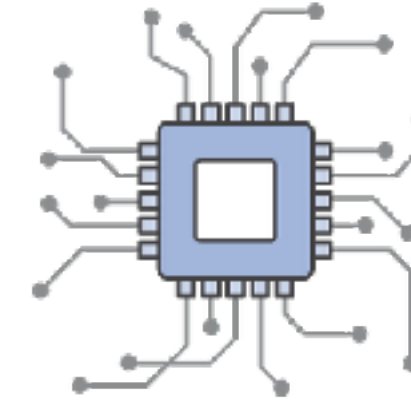


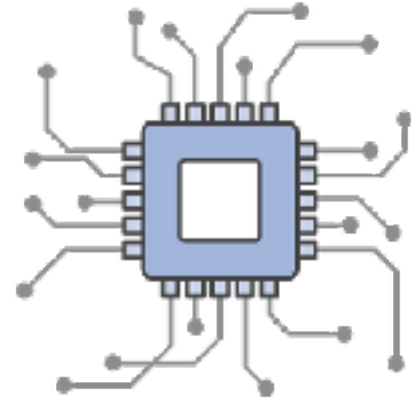


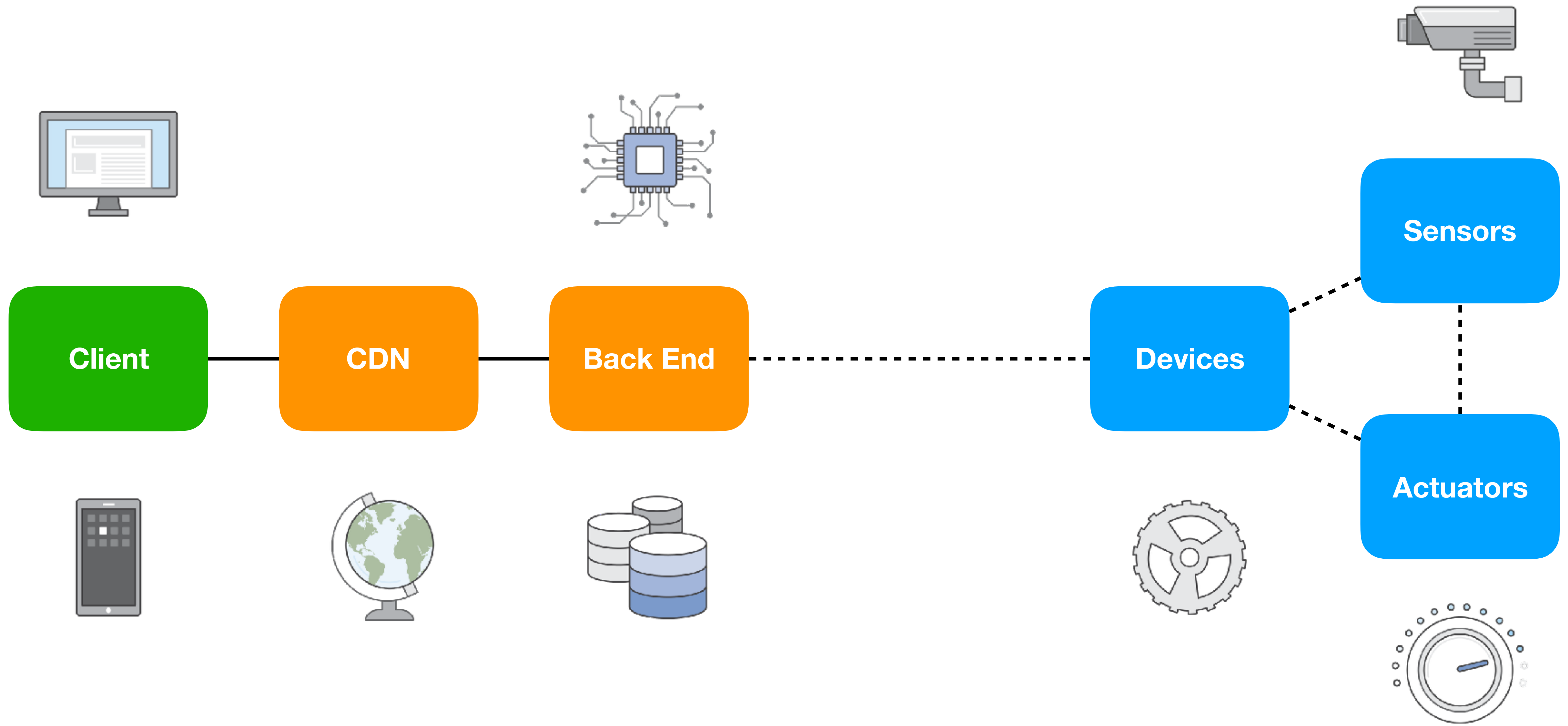


# Amazon CloudFront Edge Infrastructure

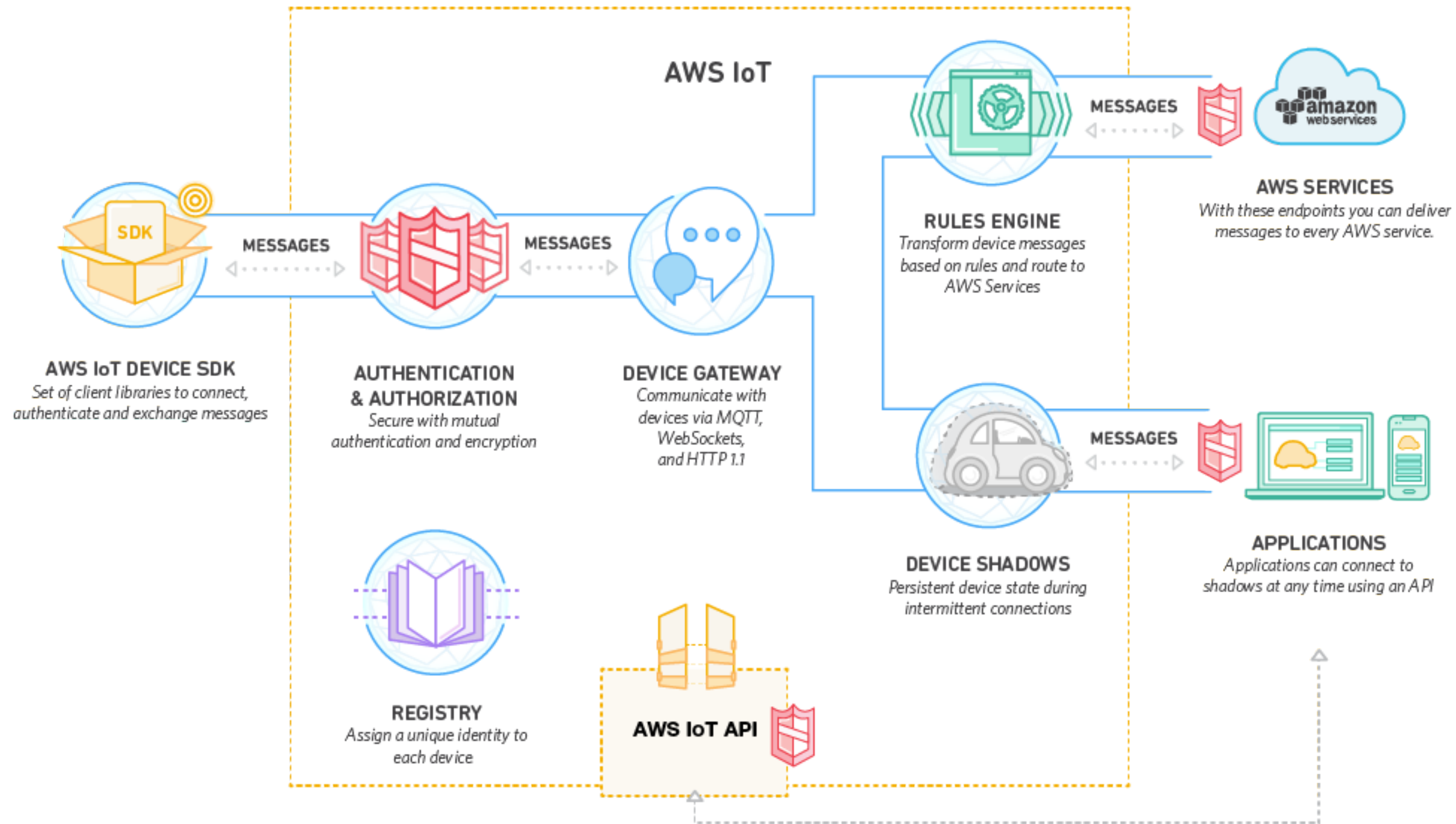




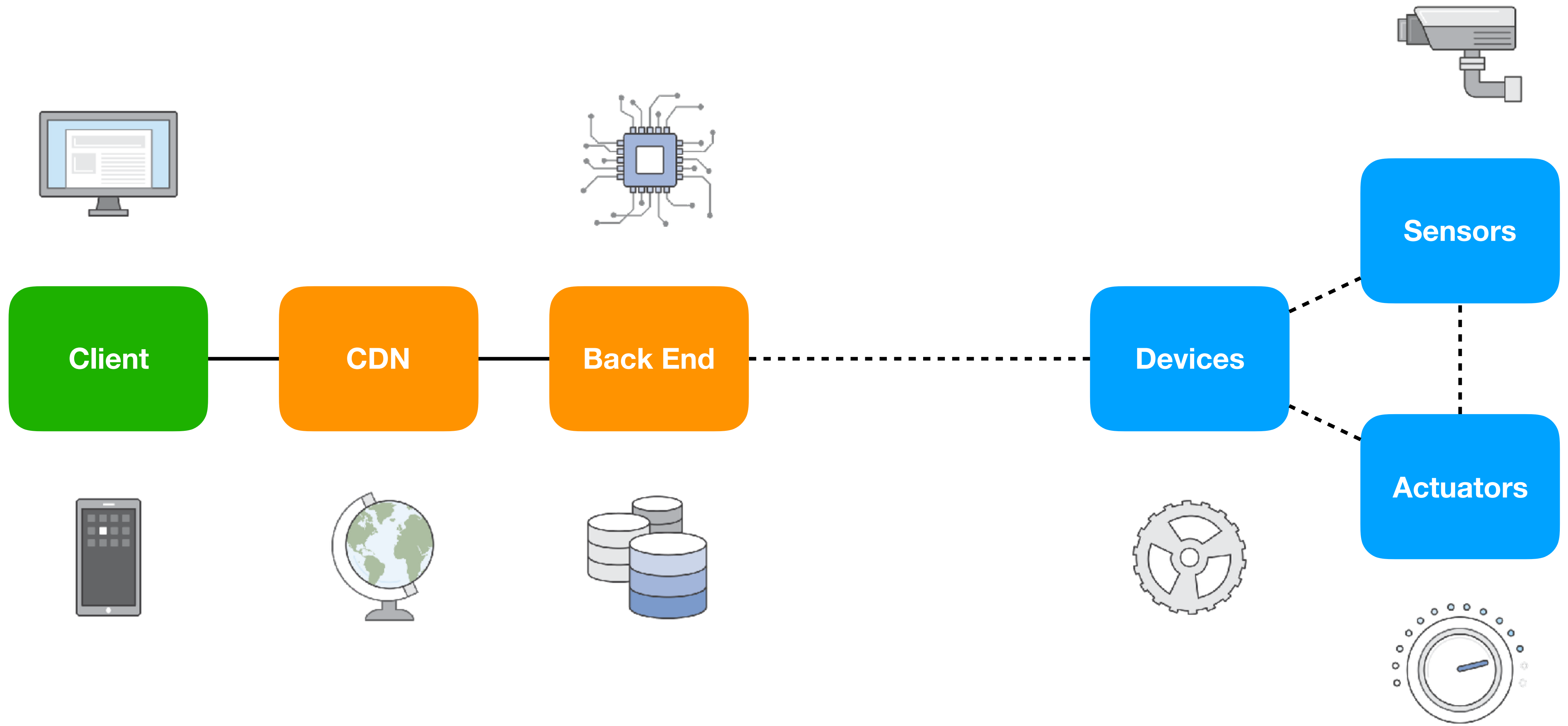


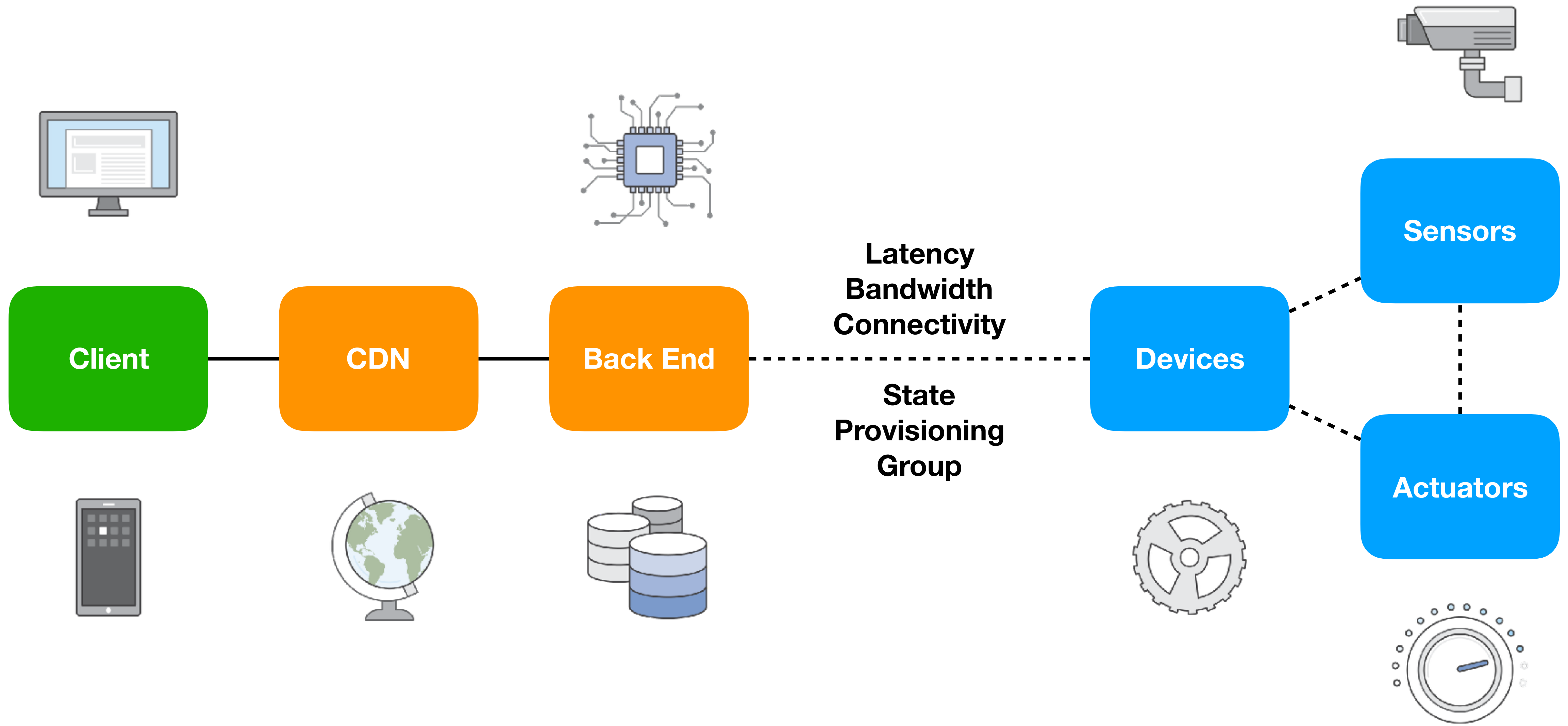


# AWS IoT

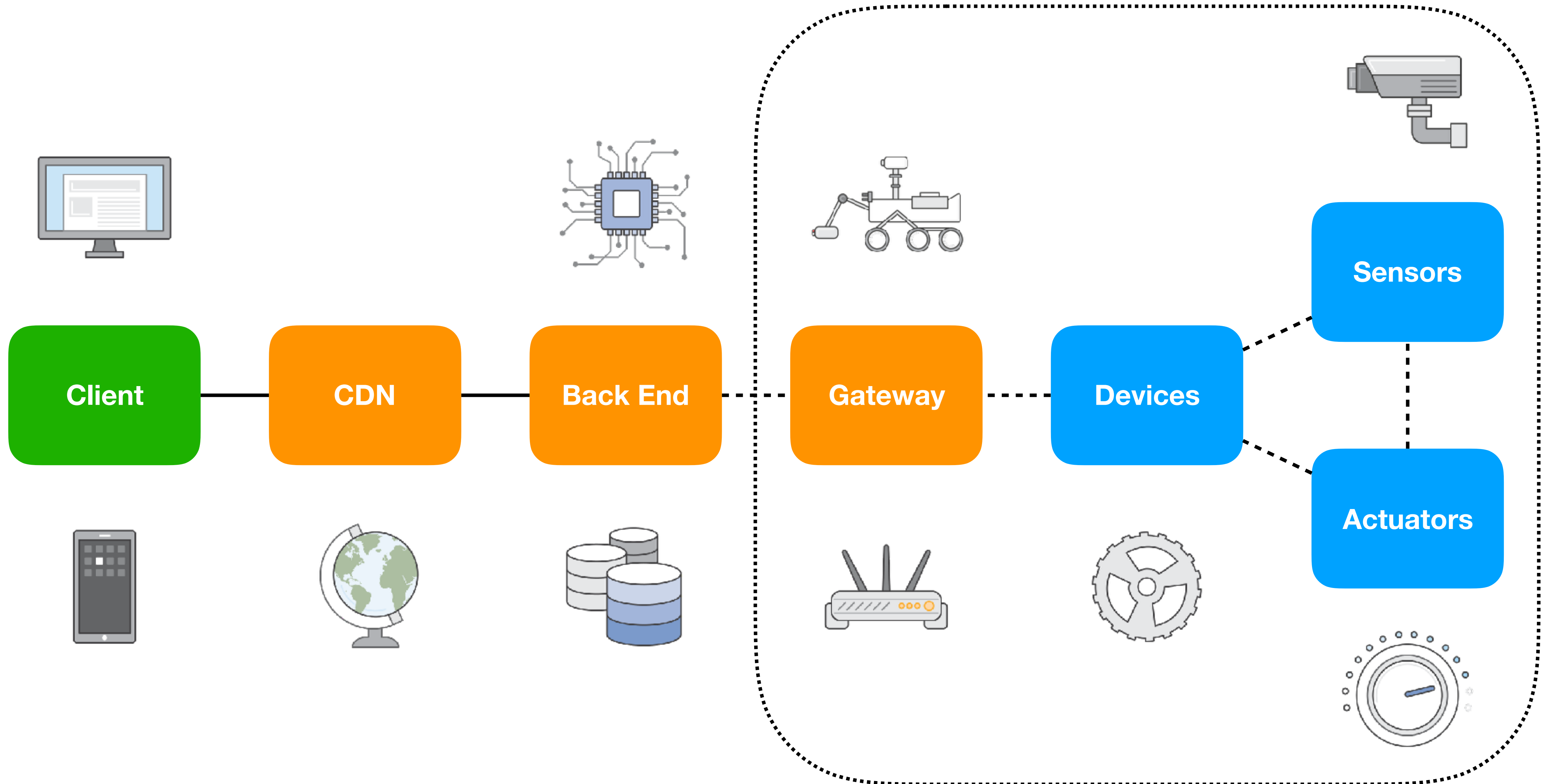




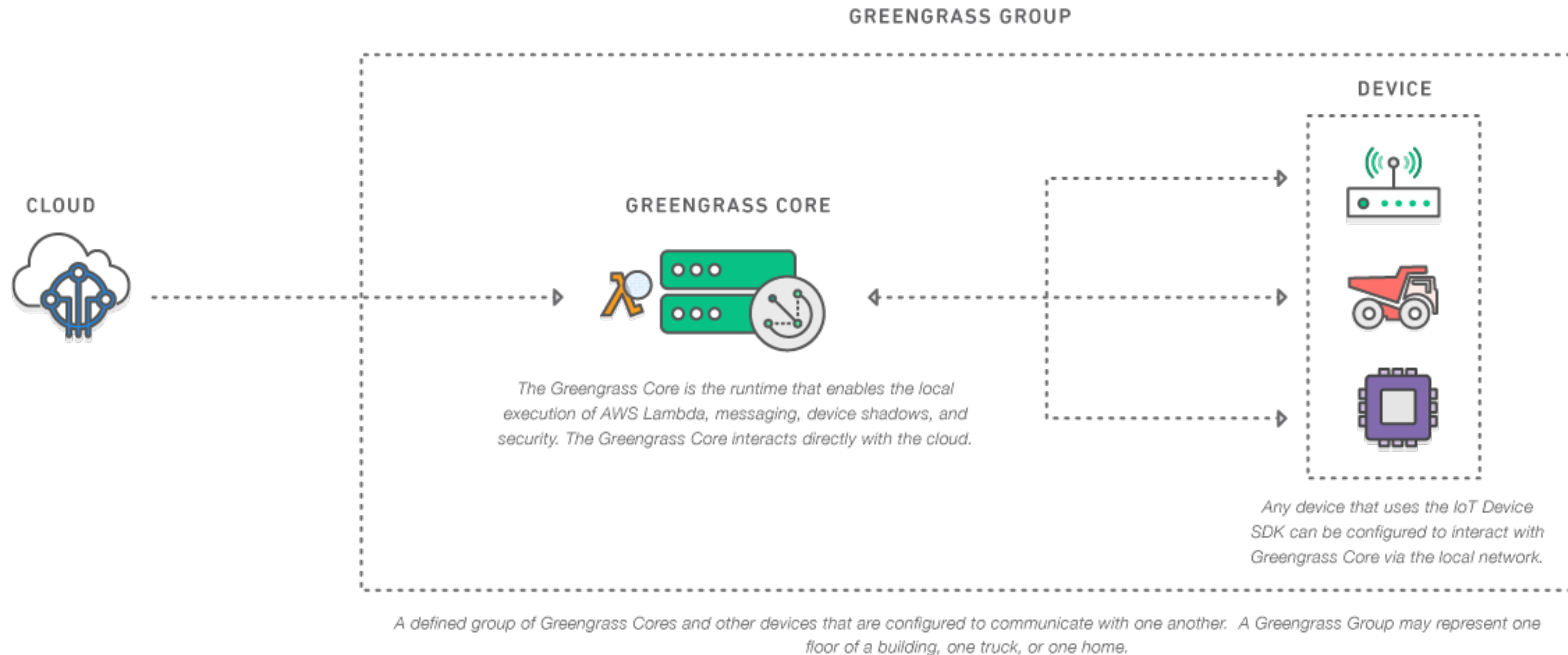




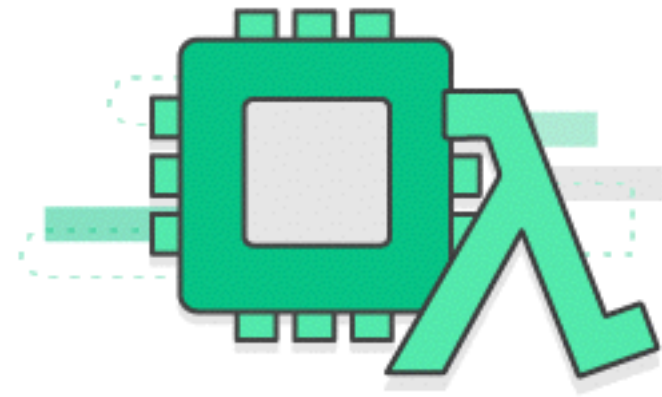




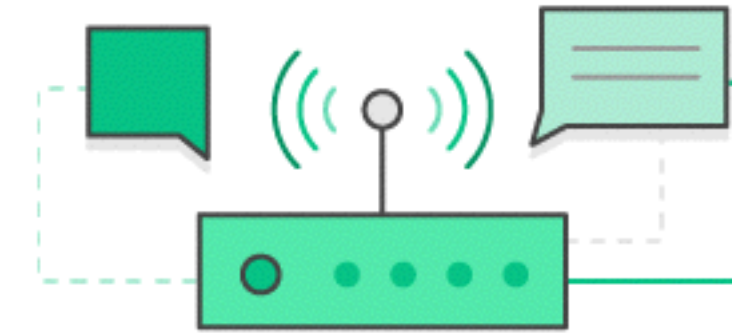
# AWS Greengrass



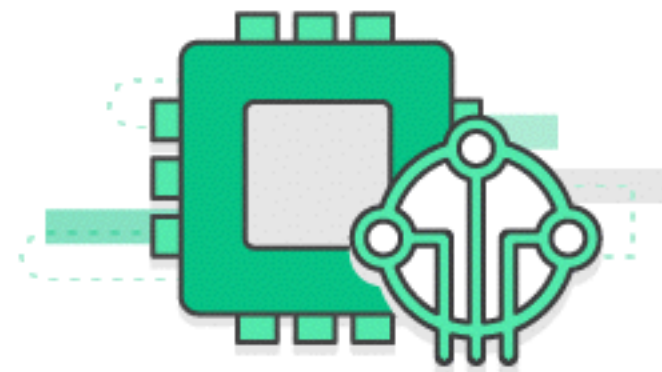
# Greengrass Features



**Local Support for AWS  
Lambda**



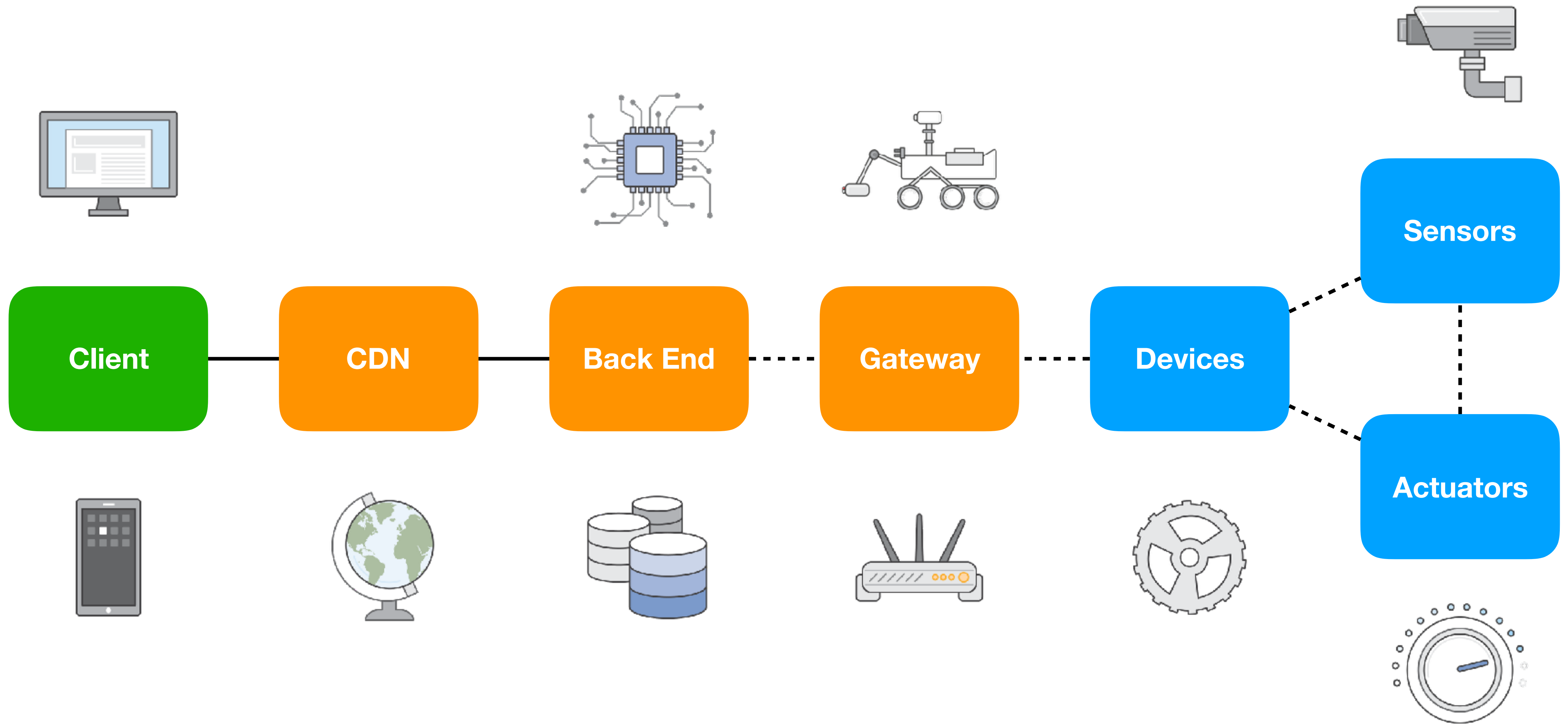
**Local Messaging**



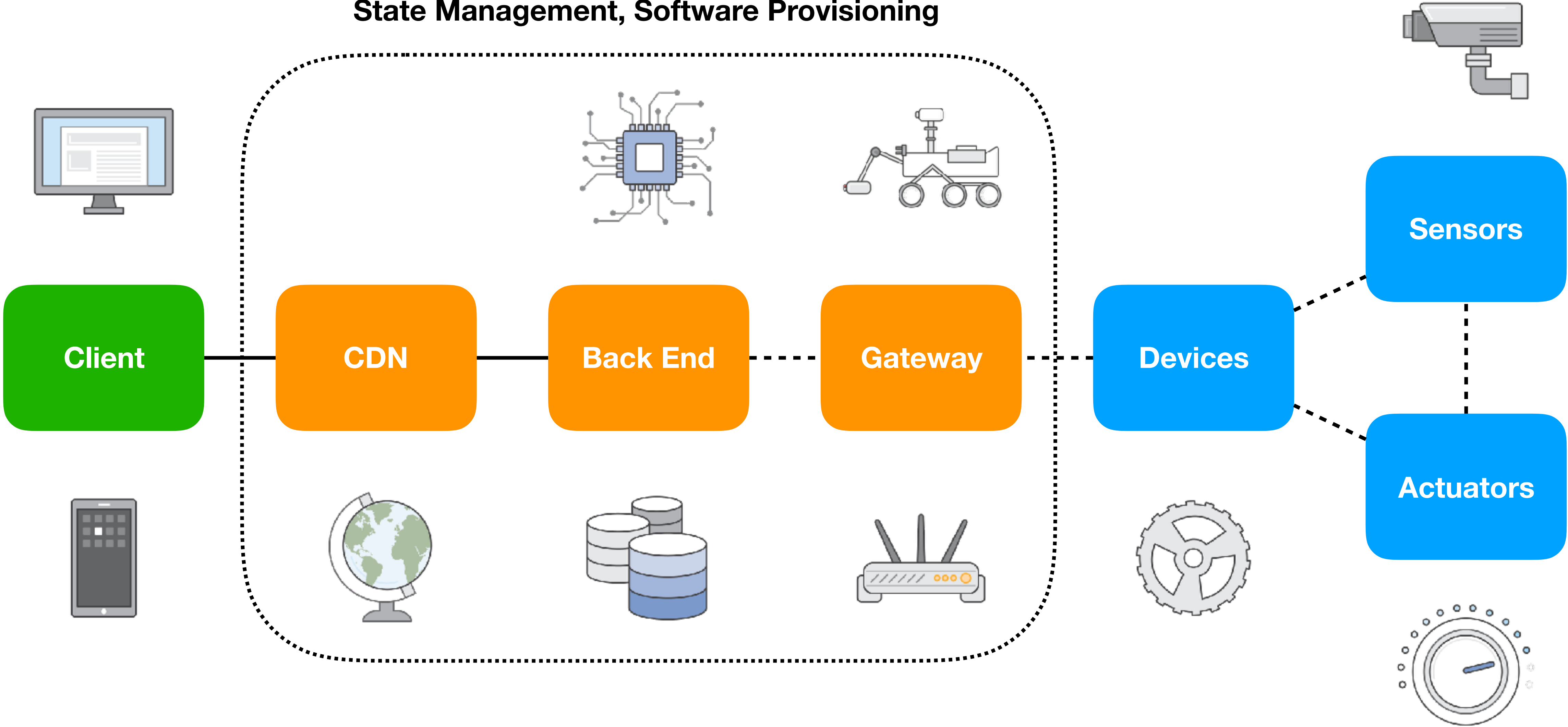
**Local Support for AWS IoT  
Devices Shadows**



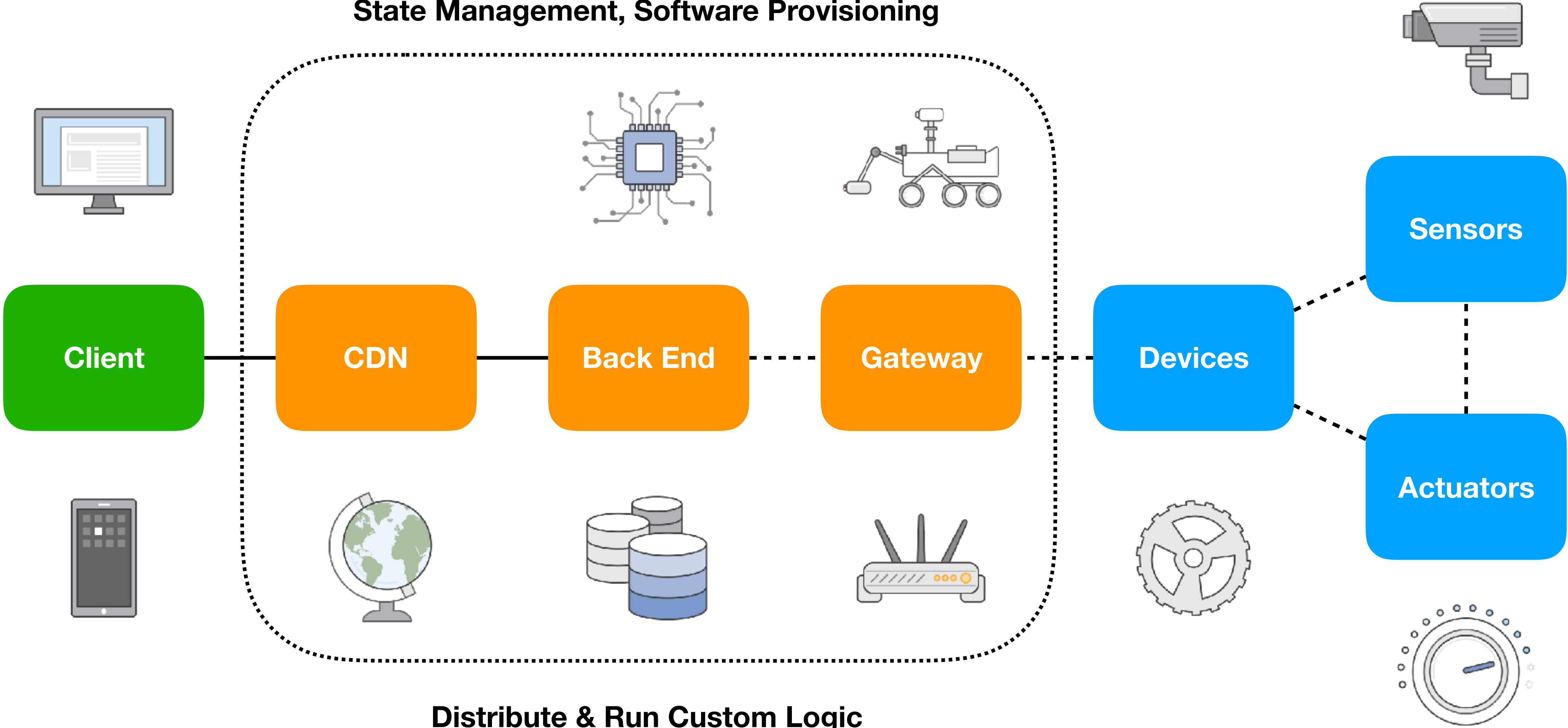
**Hardware Ecosystem**



Optimize Latency, Bandwidth, Connectivity,  
State Management, Software Provisioning

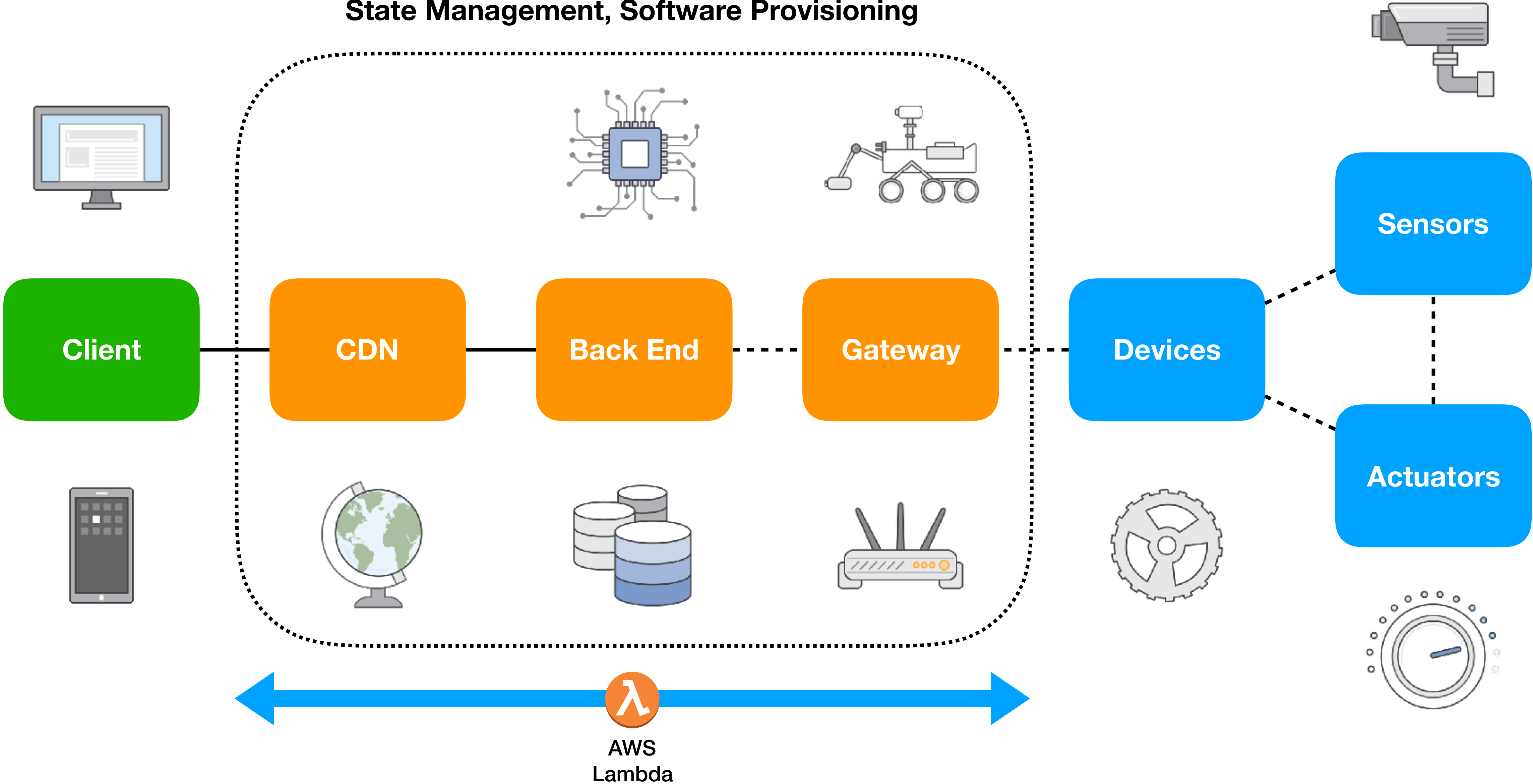


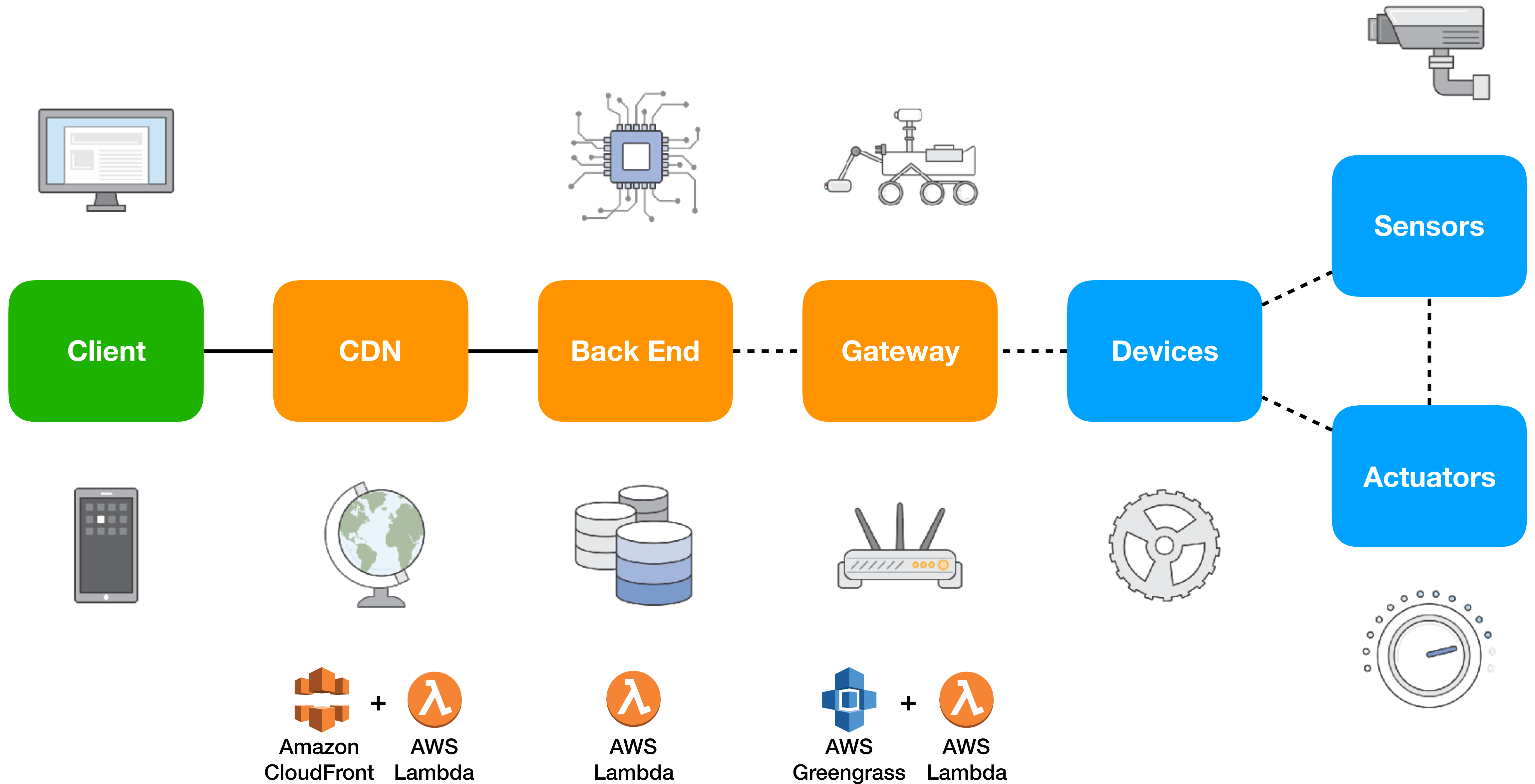
Optimize Latency, Bandwidth, Connectivity,  
State Management, Software Provisioning



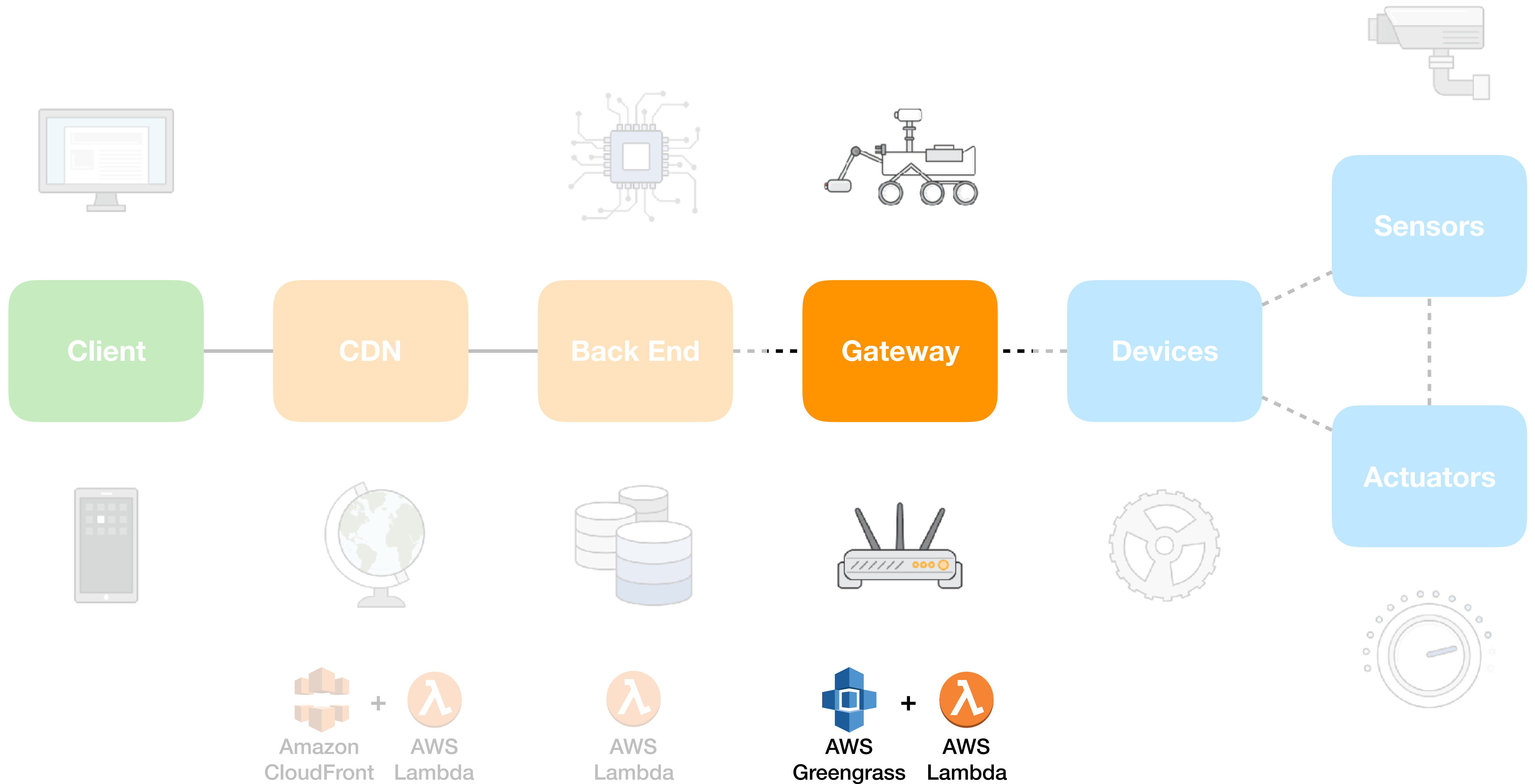


Optimize Latency, Bandwidth, Connectivity,  
State Management, Software Provisioning









# Greengrass Example Functions

## Long-Lived Functions (running indefinitely)

```
import greengrasssdk
import platform
from threading import Timer
import time

# Creating a greengrass core sdk client
client = greengrasssdk.client('iot-data')

my_platform = platform.platform()

def greengrass_hello_world_run():
    if not my_platform:
        client.publish(topic='hello/world',
                       payload='Hello world! Sent from Greengrass Core.')
    else:
        client.publish(topic='hello/world',
                       payload='Hello world! Sent from Greengrass Core running on platform: {}'.format(my_platform))
    # Asynchronously schedule this function to be run again in 5 seconds
    Timer(5, greengrass_hello_world_run).start()

# Execute the function above
greengrass_hello_world_run()

# This is a dummy handler and will not be invoked
def function_handler(event, context):
    return
```

# Greengrass Example Functions

## On-Demand Functions

```
import greengrasssdk
import platform
import json

# Creating a greengrass core sdk client
client = greengrasssdk.client('iot-data')

# Retrieving platform information to send from Greengrass Core
my_platform = platform.platform()

def function_handler(event, context):
    print("Received event: " + json.dumps(event, indent=2))
    message = {}
    message['debug'] = 'I just received this event: {}'.format(json.dumps(event, indent=2))
    if not my_platform:
        message['text'] = 'Hello World! Sent from Greengrass Core.'
    else:
        message['text'] = 'Hello World! Sent from Greengrass Core running on platform: {}'.format(my_platform)
    client.publish(topic='hello/world', payload=json.dumps(message))
    return
```

# Greengrass Demo

# Greengrass Benefits



**Respond to Local Events  
in Near Real-time**



**Operate Offline**



**Secure  
Communication**



**Simplified Devices  
Programming**



**Reduce the Cost of  
Running IoT Applications**

# AWS Snowball Edge

100TB + Greengrass Core ( $\approx$  EC2 m4.4xlarge instance)





# Snowball Edge Use Cases

## Embedded Applications



**PHILIPS**

*“Snowball Edge enables us to extend the innovative capabilities of HealthSuite, our cloud-enabled connected health ecosystem of devices, applications and digital tools supported by AWS, even when there is no network support.”*

—Dale Wiggins,  
Business Leader, HealthSuite digital platform,  
Philips

# Snowball Edge Use Cases

Remote Locations for data collection and analysis



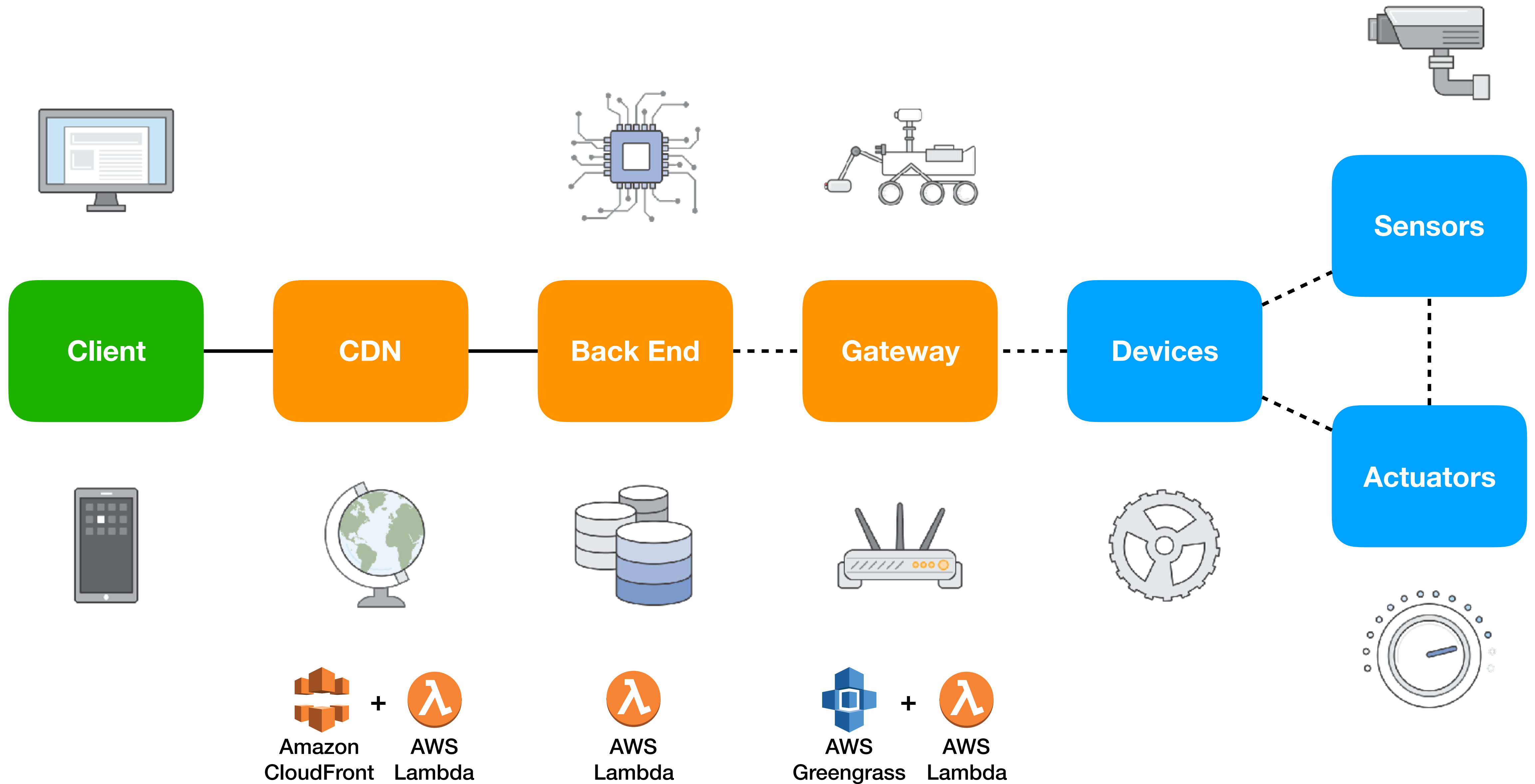
**Oregon State**  
UNIVERSITY

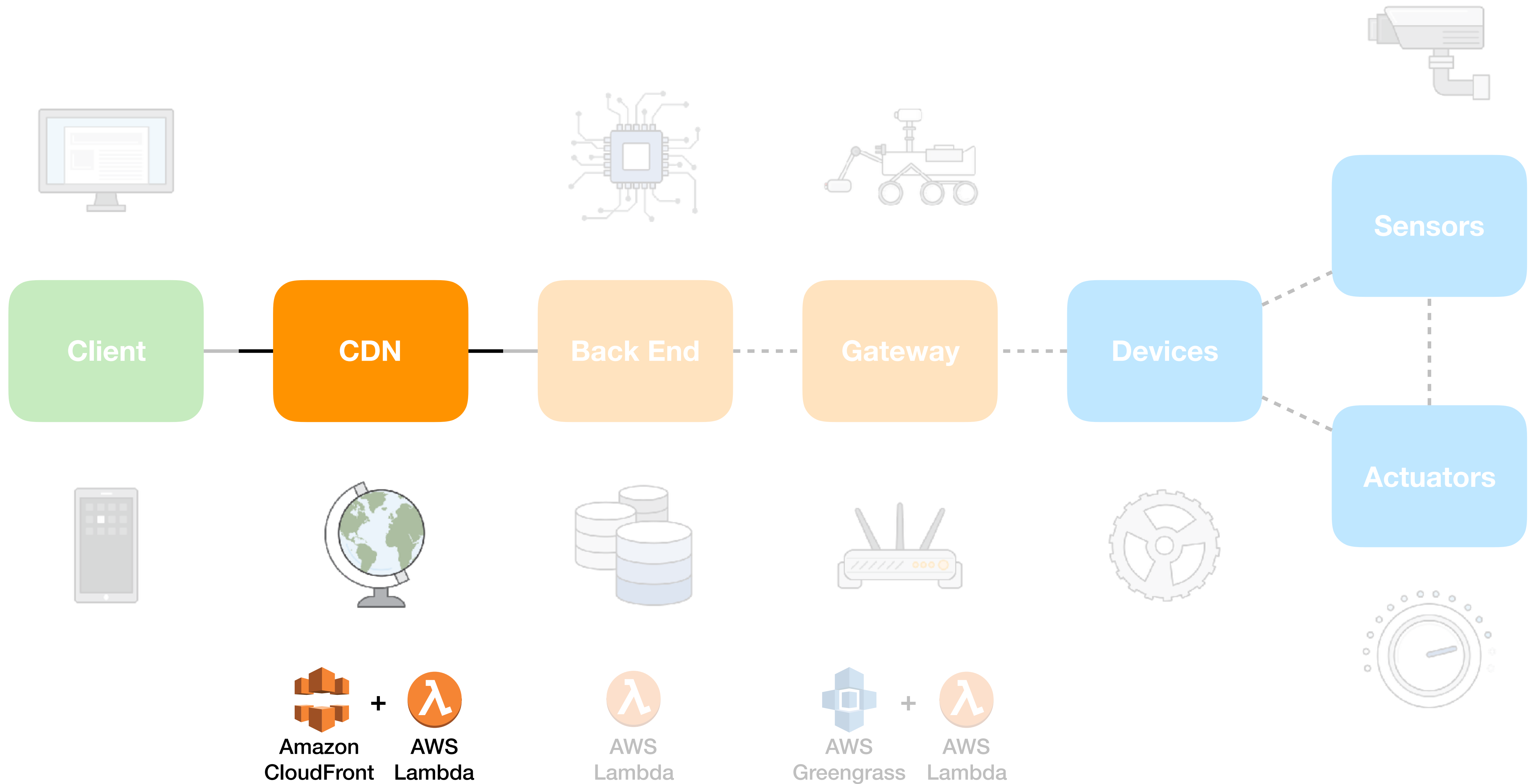
**Hatfield Marine  
Science Center**

*“With AWS Snowball Edge, we can now collect 100 TB of data with no intermediate steps, and we can also analyze the images immediately using the onboard compute capabilities.”*

— Bob Cowen,  
Director of Hatfield Marine Research Center,  
Oregon State University



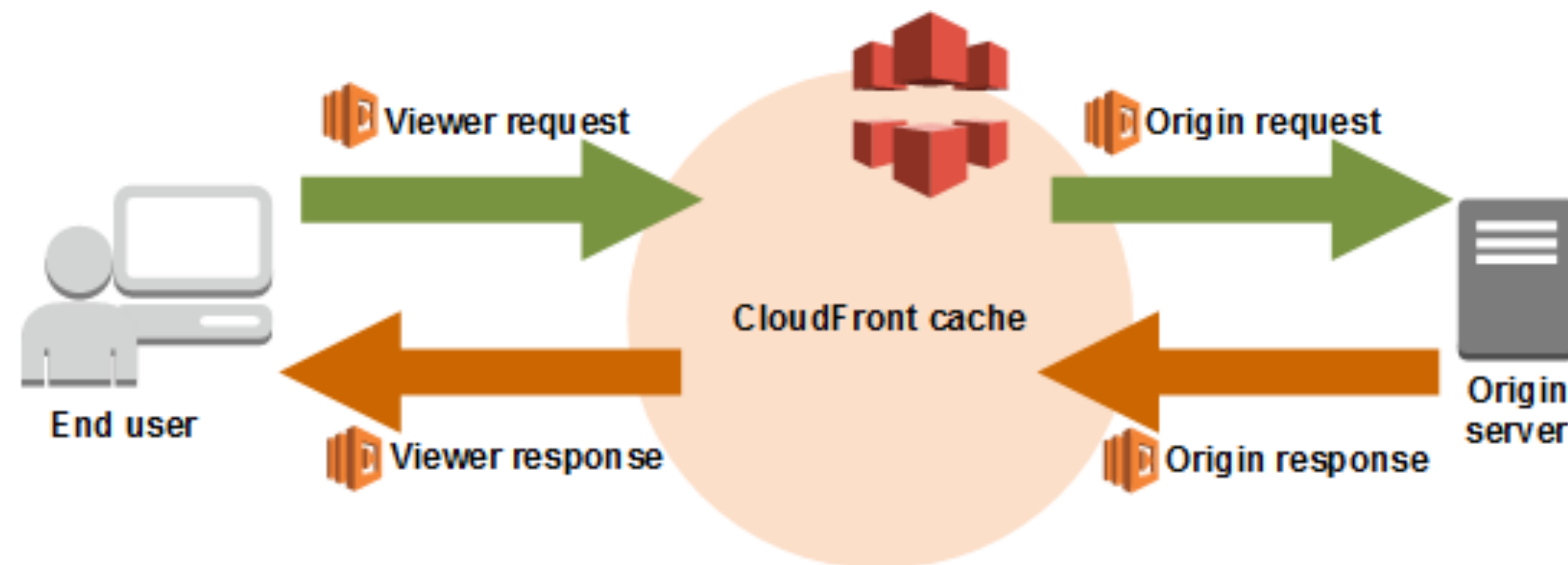




# AWS Lambda@Edge

Take serverless to your users

## Events / Triggers



Customize your content delivery while reducing load on your origin

# Lambda@Edge Event Structure

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionId": "EDFDVBD6EXAMPLE"
        },
        "request": {
          "clientIp": "2001:0db8:85a3:0:0:8a2e:0370:7334",
          "method": "GET",
          "uri": "/picture.jpg",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "curl/7.51.0"
              }
            ]
          }
        }
      }
    }
  ]
}
```

## Request Event

<http://amzn.to/2hunJuA>

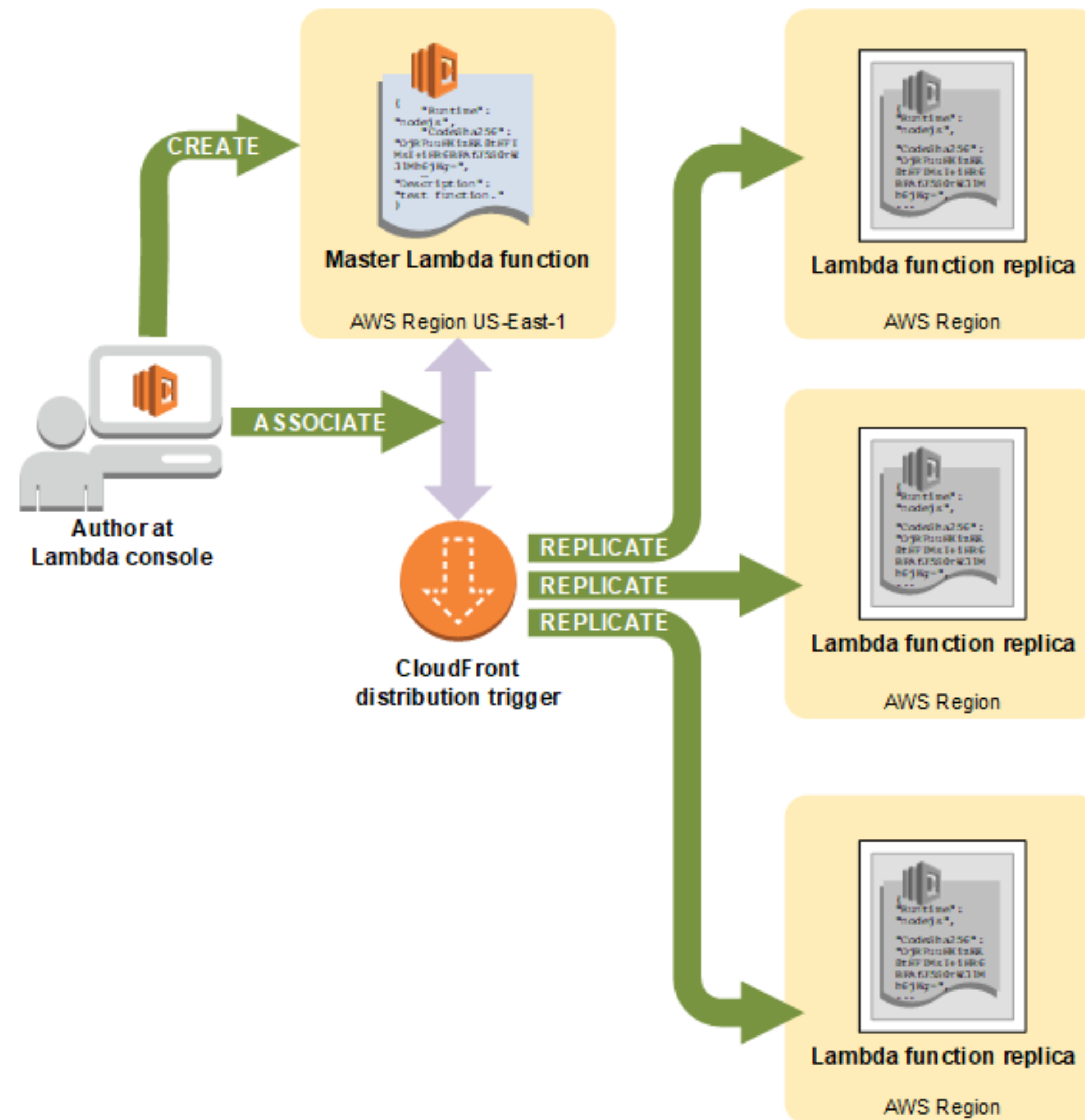
```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionId": "EDFDVBD6EXAMPLE"
        },
        "request": {
          "clientIp": "2001:0db8:85a3:0:0:8a2e:0370:7334",
          "method": "GET",
          "uri": "/picture.jpg",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "curl/7.18.1"
              }
            ]
          }
        },
        "response": {
          "status": "200",
          "statusDescription": "OK",
          "headers": {
            "server": [
              {
                "key": "Server",
                "value": "MyCustomOrigin"
              }
            ],
            "set-cookie": [
              {
                "key": "Set-Cookie",
                "value": "theme=light"
              },
              {
                "key": "Set-Cookie",
                "value": "sessionToken=abc123; Expires=Wed, 09 Jun 2021 10:18:14 GMT"
              }
            ]
          }
        }
      }
    }
  ]
}
```

## Response Event

# Creating Lambda@Edge Functions

Lambda replicates  
the function to  
AWS Regions and  
CloudFront edge  
locations around the  
globe

# Replicas can't be edited or deleted



# Lambda@Edge Example Functions

## cloudfront-http-redirect

```
'use strict';

exports.handler = (event, context, callback) => {
  /*
   * Generate HTTP redirect response with 302 status code and Location header.
   */
  const response = {
    status: '302',
    statusDescription: 'Found',
    headers: {
      location: [{
        key: 'Location',
        value: 'http://docs.aws.amazon.com/lambda/latest/dg/lambda-edge.html',
      }],
    },
  };
  callback(null, response);
};
```

# Lambda@Edge Example Functions

cloudfront-response-generation

```
'use strict';

exports.handler = (event, context, callback) => {
  /*
   * Generate HTTP response using 200 status code with a simple body.
   */
  const response = {
    status: '200',
    statusDescription: 'OK',
    headers: {
      vary: [{
        key: 'Vary',
        value: '*',
      }],
      'last-modified': [{
        key: 'Last-Modified',
        value: '2017-01-13',
      }],
    },
    body: 'Example body generated by Lambda@Edge function.',
  };

  callback(null, response);
};
```



# Lambda@Edge Example Functions

## A/B Testing

<http://amzn.to/2hoarw4>

```
'use strict';

exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const headers = request.headers;

  if (request.uri !== '/experiment-pixel.jpg') {
    // do not process if this is not an A-B test request
    callback(null, request);
    return;
  }

  const cookieExperimentA = 'X-Experiment-Name=A';
  const cookieExperimentB = 'X-Experiment-Name=B';
  const pathExperimentA = '/experiment-group/control-pixel.jpg';
  const pathExperimentB = '/experiment-group/treatment-pixel.jpg';

  let experimentUri;
  if (headers.cookie) {
    for (let i = 0; i < headers.cookie.length; i++) {
      if (headers.cookie[i].value.indexOf(cookieExperimentA) >= 0) {
        console.log('Experiment A cookie found');
        experimentUri = pathExperimentA;
        break;
      } else if (headers.cookie[i].value.indexOf(cookieExperimentB) >= 0) {
        console.log('Experiment B cookie found');
        experimentUri = pathExperimentB;
        break;
      }
    }
  }

  if (!experimentUri) {
    console.log('Experiment cookie has not been found. Throwing dice...');
    if (Math.random() < 0.75) {
      experimentUri = pathExperimentA;
    } else {
      experimentUri = pathExperimentB;
    }
  }

  request.uri = experimentUri;
  console.log(`Request uri set to "${request.uri}"`);
  callback(null, request);
};
```

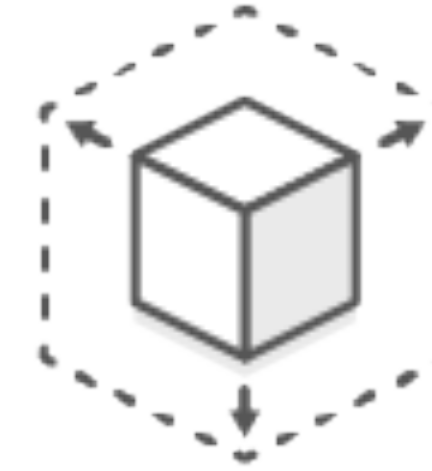


# **Lambda@Edge Demo**

# Lambda@Edge Benefits



**Global**



**Continuous Scaling**



**Reduce Origin Load**



**Subsecond Metering**

# Lambda@Edge Use Cases

**Access Control at the Edge**

**URL Rewrites**

# Lambda@Edge Use Cases

**Remote Network Calls**

**Response Generation At Viewer Request**

**Access Control at the Edge**

**URL Rewrites**

# Lambda@Edge Use Cases

**Highly Personalized Websites**

**A/B Testing**

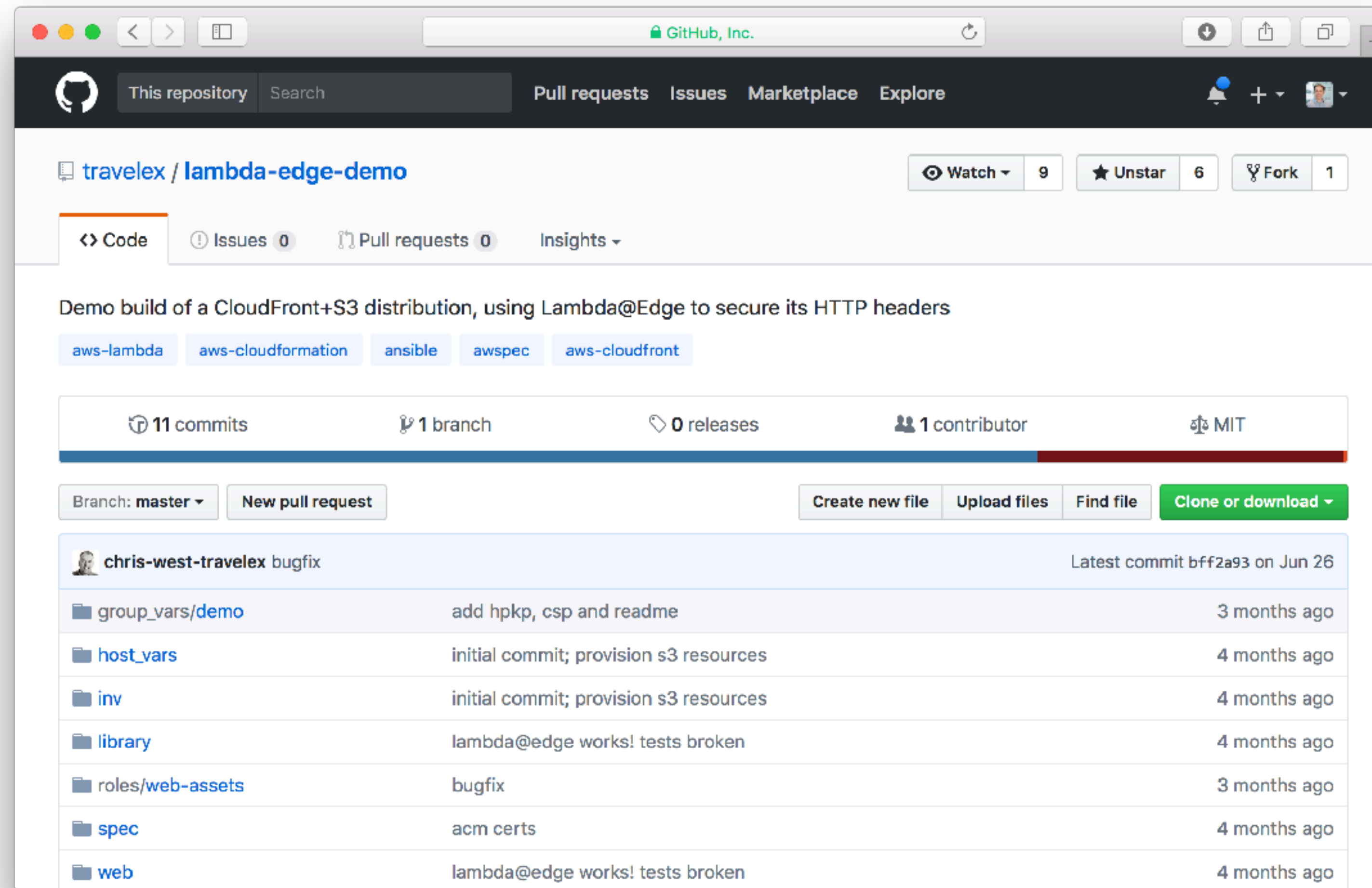
**Remote Network Calls**

**Response Generation At Viewer Request**

**Access Control at the Edge**

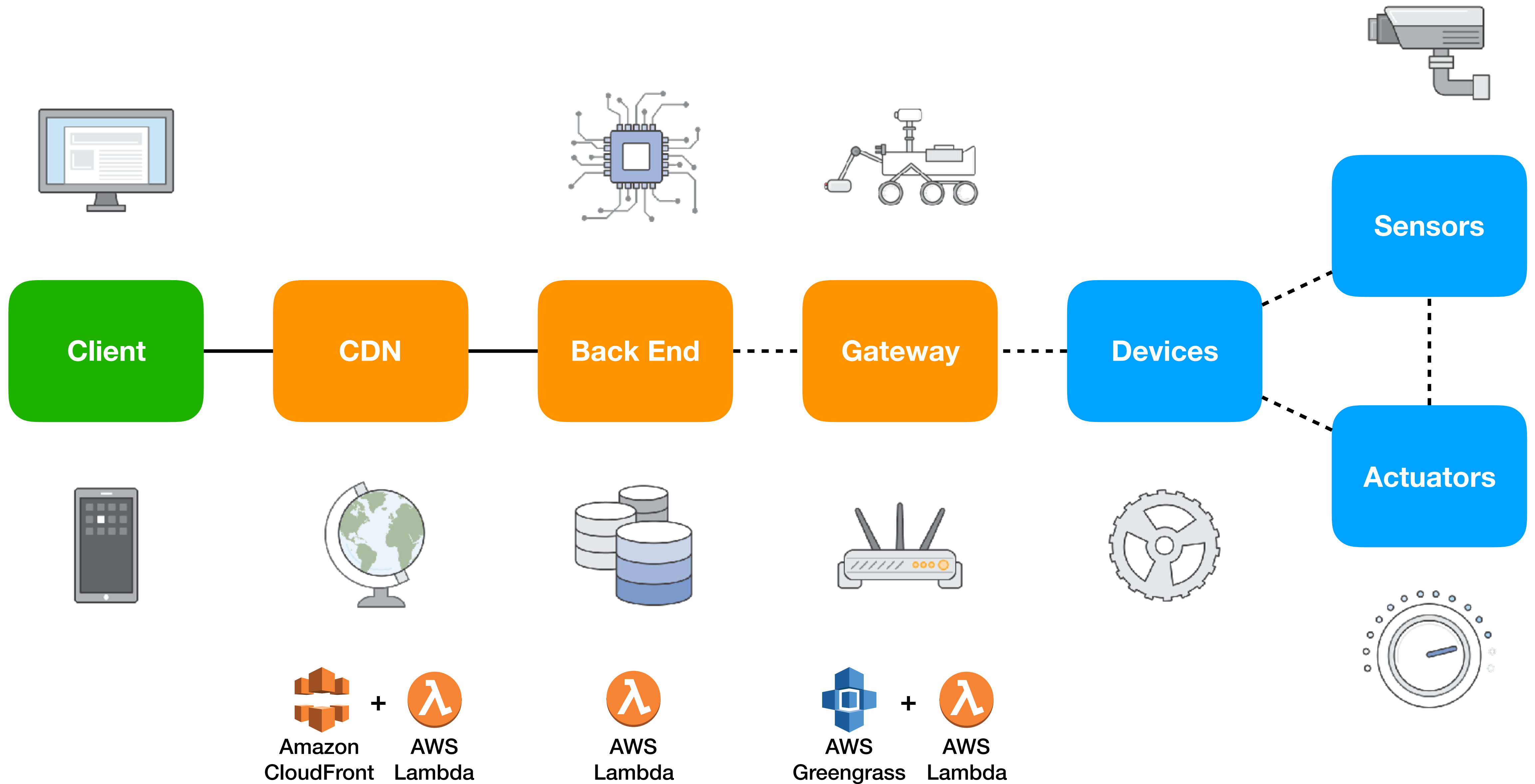
**URL Rewrites**

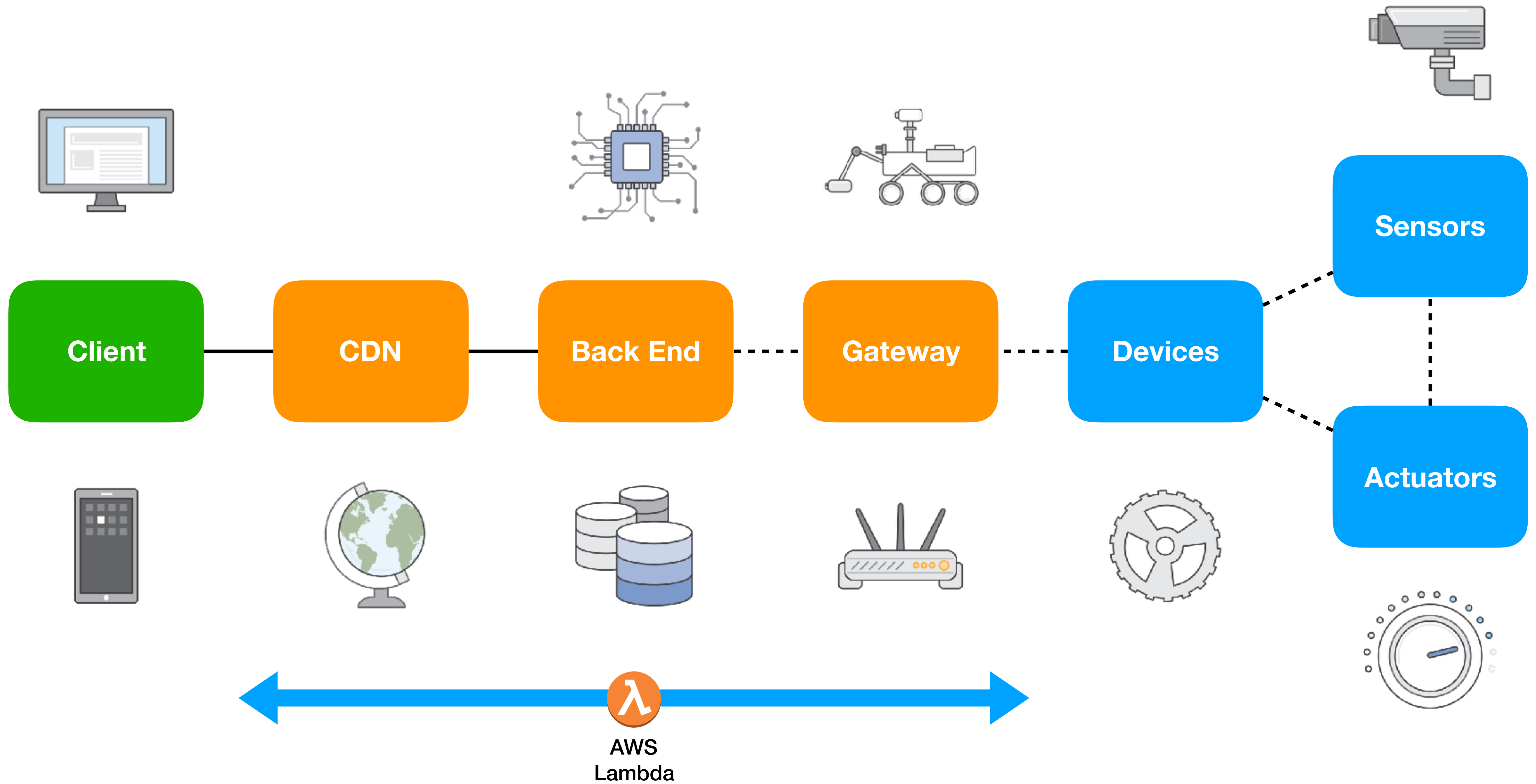
# Lambda@Edge Use Cases



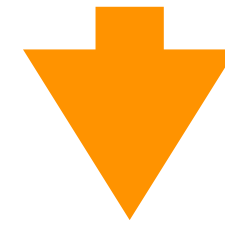
*Demo build of a CloudFront+S3 distribution, using Lambda@Edge to secure its HTTP headers*



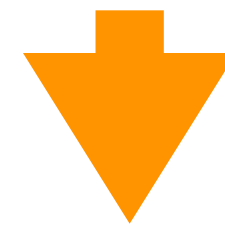




Optimize your IT Architectures at the Edge



Push Business Logic Closer to Users & Devices



Use the Same Serverless Programming Model



Create a Better User Experience for Your Customers



Pop-up Loft  
**LONDON**

# Thank You

 @danilop

