



INTERNET OF THINGS | COEN 243

DEPARTMENT OF COMPUTER ENGINEERING, SANTA CLARA UNIVERSITY, CALIFORNIA

WEI XIAO, INSTRUCTOR

(WXIAO@SCU.EDU)

SHIVAKUMAR MATHAPATHI, GUEST INSTRUCTOR

(SMATHAPATHI@SCU.EDU)

## Internet of Things

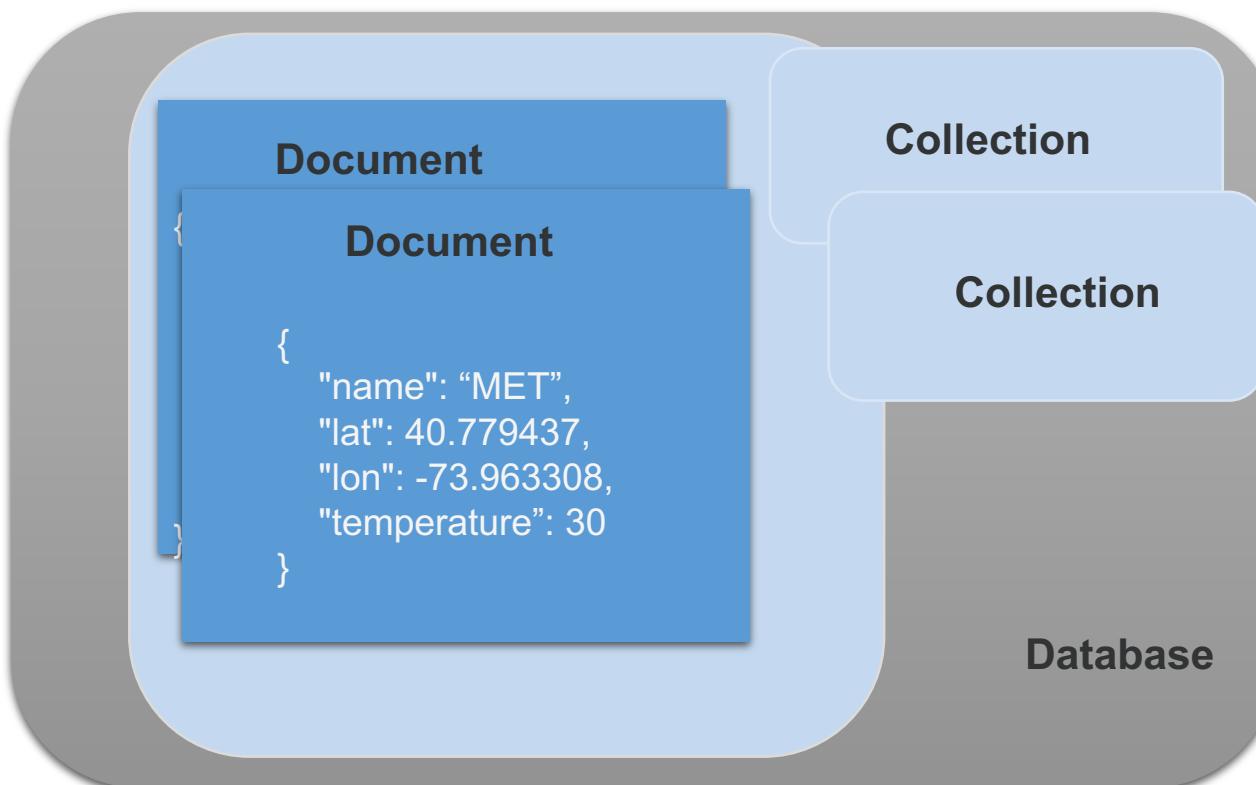
# Database, Cloud And Edge Solutions

## Database Integration

# MongoDB

## Document, Collection, Database

- Open-source NoSQL, non-relational database
- Dynamic Schema



□ Example:

```
{  
    name: "Alan",  
    age: 25,  
    groups: [ "news", "sports"]  
}
```

□ Example:

```
{  
    name: { first: "Alan", last: "Turing" },  
    birth: new Date('Jun 23, 1912'),  
    death: new Date('Jun 07, 1954'),  
    contribs: [ "Turing machine", "Turing test", "Turingery" ],  
    views : NumberLong(1250000)  
}
```

# MongoDB

## BSON Types

Type	Number	Alias	Notes
Double	1	“double”	
String	2	“string”	
Object	3	“object”	
Array	4	“array”	
Binary data	5	“binData”	
Undefined	6	“undefined”	Deprecated.
ObjectId	7	“objectId”	
Boolean	8	“bool”	
Date	9	“date”	
Null	10	“null”	
Regular Expression	11	“regex”	
DBPointer	12	“dbPointer”	Deprecated.
JavaScript	13	“javascript”	
Symbol	14	“symbol”	Deprecated.
JavaScript (with scope)	15	“javascriptWithScope”	
32-bit integer	16	“int”	
Timestamp	17	“timestamp”	
64-bit integer	18	“long”	
Decimal128	19	“decimal”	New in version 3.4.

## MongoDB Shell

- The mongo shell is an interactive JavaScript interface to MongoDB.
- You can use the mongo shell to query and update data as well as perform administrative operations.

# MongoDB

## MongoDB Host

- MongoDB Atlas      <https://www.mongodb.com/cloud/atlas>
- mLab      <https://mlab.com>



# Other DB

- InfluxDB <https://www.influxdata.com/>
- Couchbase <https://www.couchbase.com/>

 *influxdb*



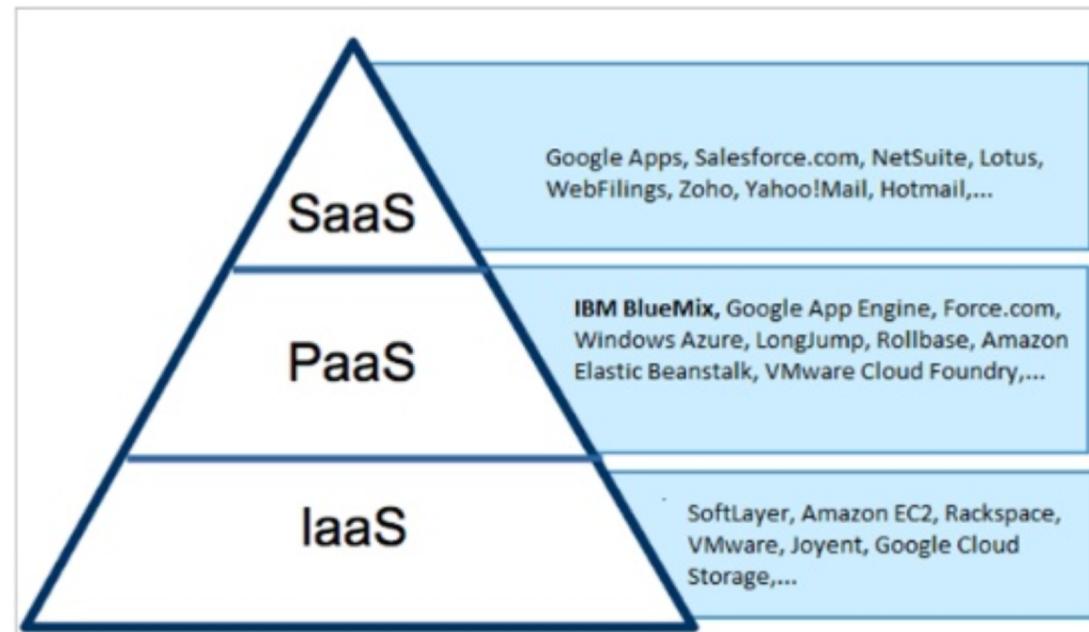
## Cloud Computing

# Cloud Computing

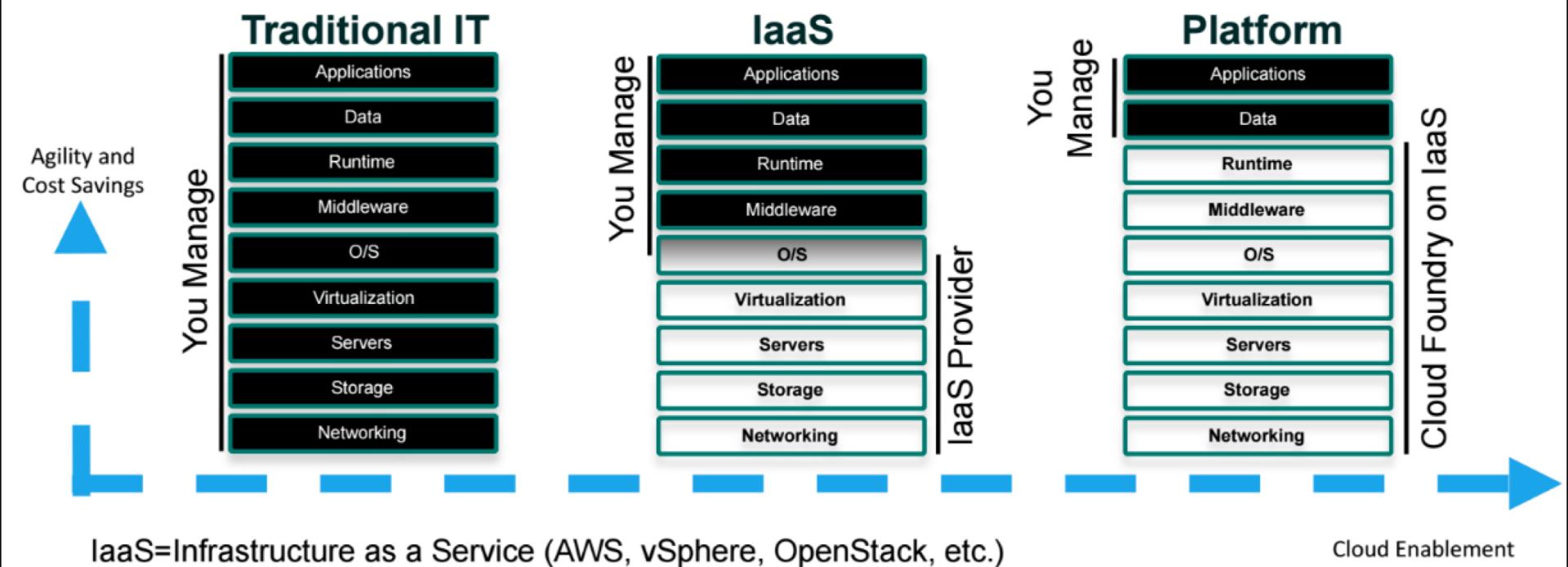
Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources(such as networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction

# Cloud Computing

- **SaaS (Software as a Service)** is something you simply sign up for and generally start using with little or no customization.
- **PaaS (Platform as a Service)** is something which you use to build or run your own applications — e.g. a managed environment for running code or a DB.
- **IaaS (Infrastructure as a Service)** is core IT infrastructure, you either rent a virtual server, or to provide solutions for file storage, load balancing or some other “network type” problem.



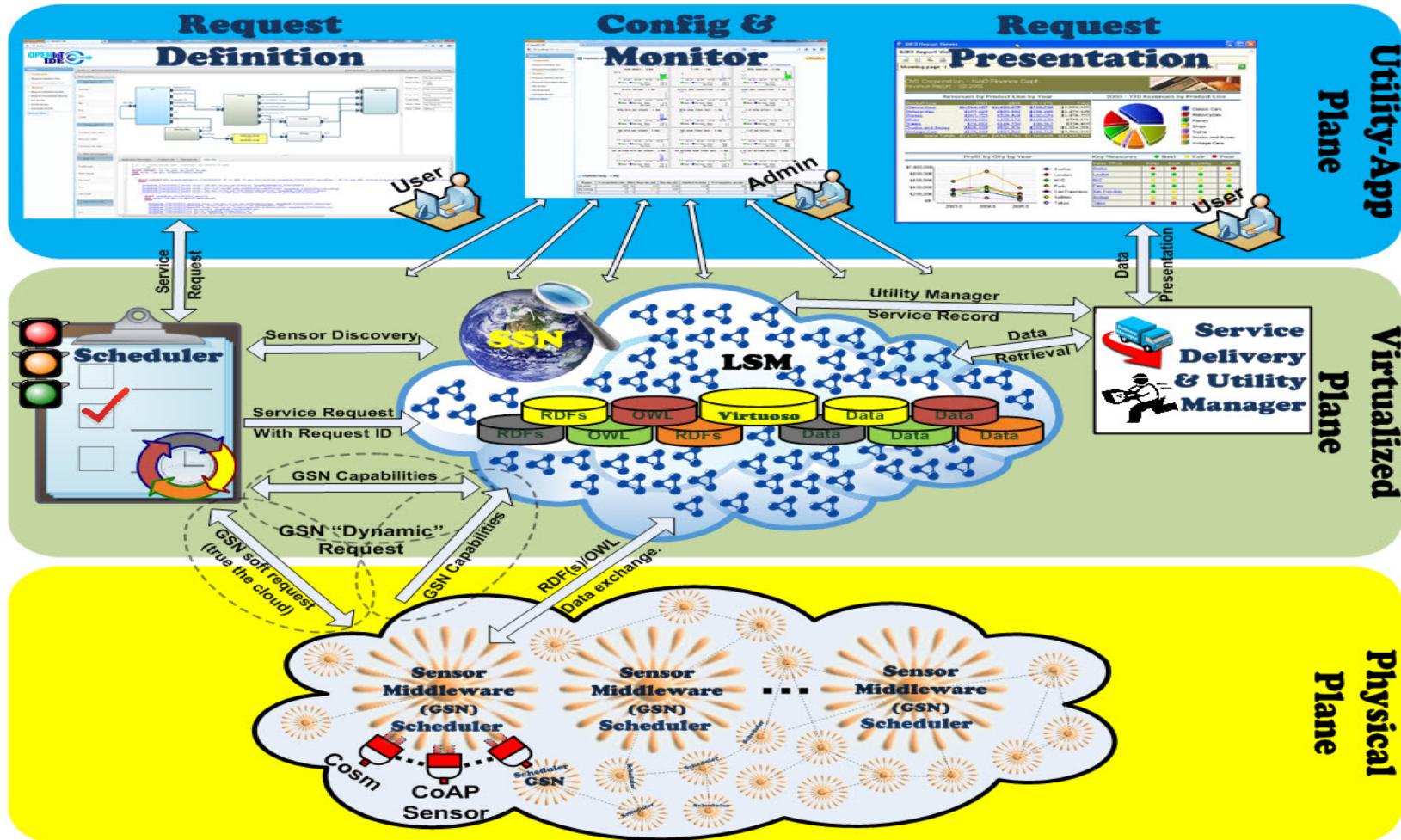
# Cloud Computing



Source: CloudFoundry.com

# Cloud Platforms

## Open IoT



# Cloud Platforms

## AWS IoT

AWS IoT is a fully managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices.



1

Securely connect and manage any physical device across multiple networks and protocols

2

Extract and Filter data from your devices and take action with custom Rules

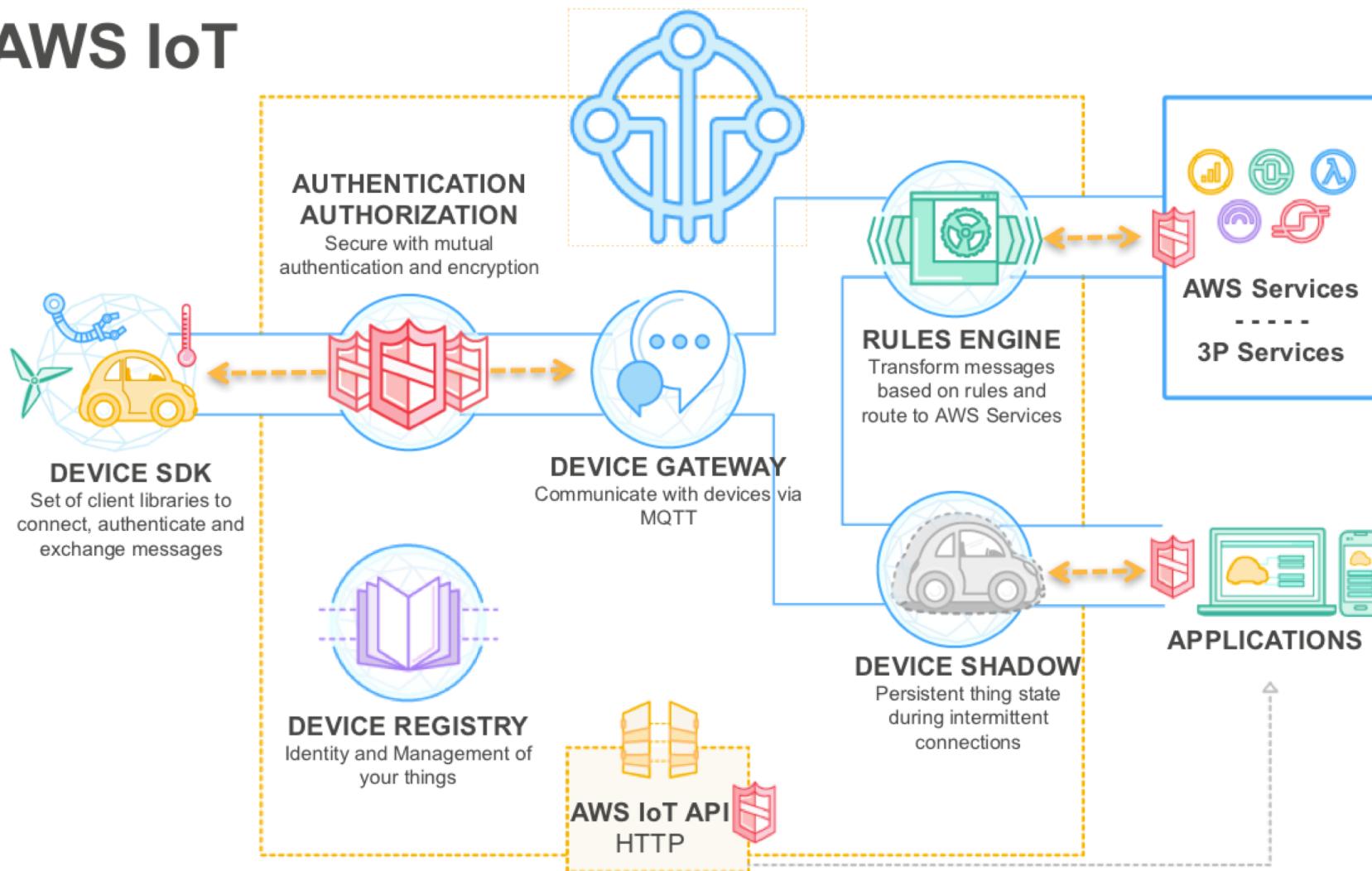
3

Create Web and Mobile Applications that interact with Devices reliably at any time

# Cloud Platforms

## AWS IoT

### AWS IoT



Source: [aws.amazon.com](http://aws.amazon.com)

## Device Gateway

- Standard protocols – MQTT / HTTP
- AWS front-end for devices
- Scalable, long-lived connections
- Secure by default – X509 certs, TLS 1.2

Topic Based Architecture:

Example: scu/computerengineering/coen243

## Device Registry

- Device meta-data

- Example:

```
$ aws --region us-east-1 iot describe-thing --thing-name pump1
{
    "attributes": {
        "MN": "AB776",
        "install_data": "12-29-2016",
        "SN": "2542678921"
    },
    "thingName": "pump1",
    "defaultClientId": "pump1"
}
```

## Device Shadows



- Report its current state to one or multiple shadow
- Retrieve its desired state from shadow

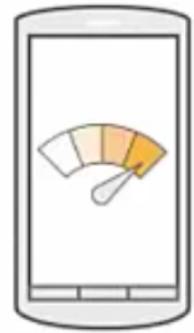


- Shadow reports delta, desired and reported states along with metadata and version



- Set the desired state of a device
- Get the last reported state of the device
- Delete the shadow

## Device Shadows



"engine": "ON"

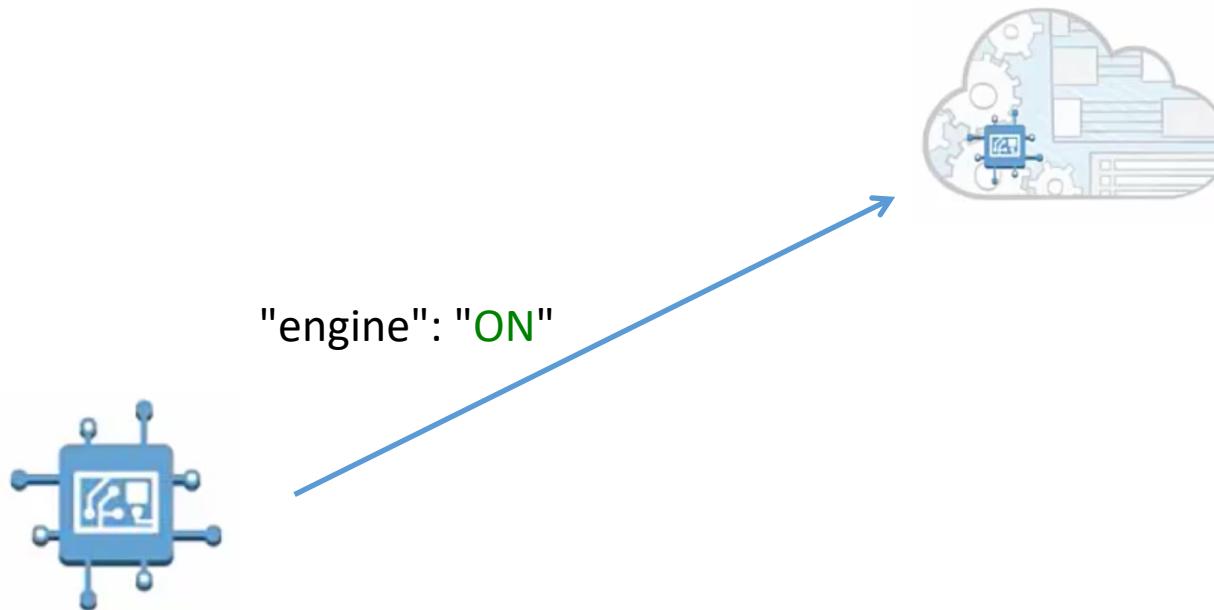
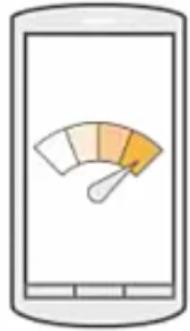


"engine": "OFF"

"engine": "ON"

```
{  
  "state": {  
    "desired": {  
      "engine": "ON",  
      "light": "OFF"  
    }  
    "reported": {  
      "engine": "OFF",  
      "light": "OFF"  
    }  
    "delta": {  
      "engine": "ON",  
    }  
  }  
  "version": 1  
}
```

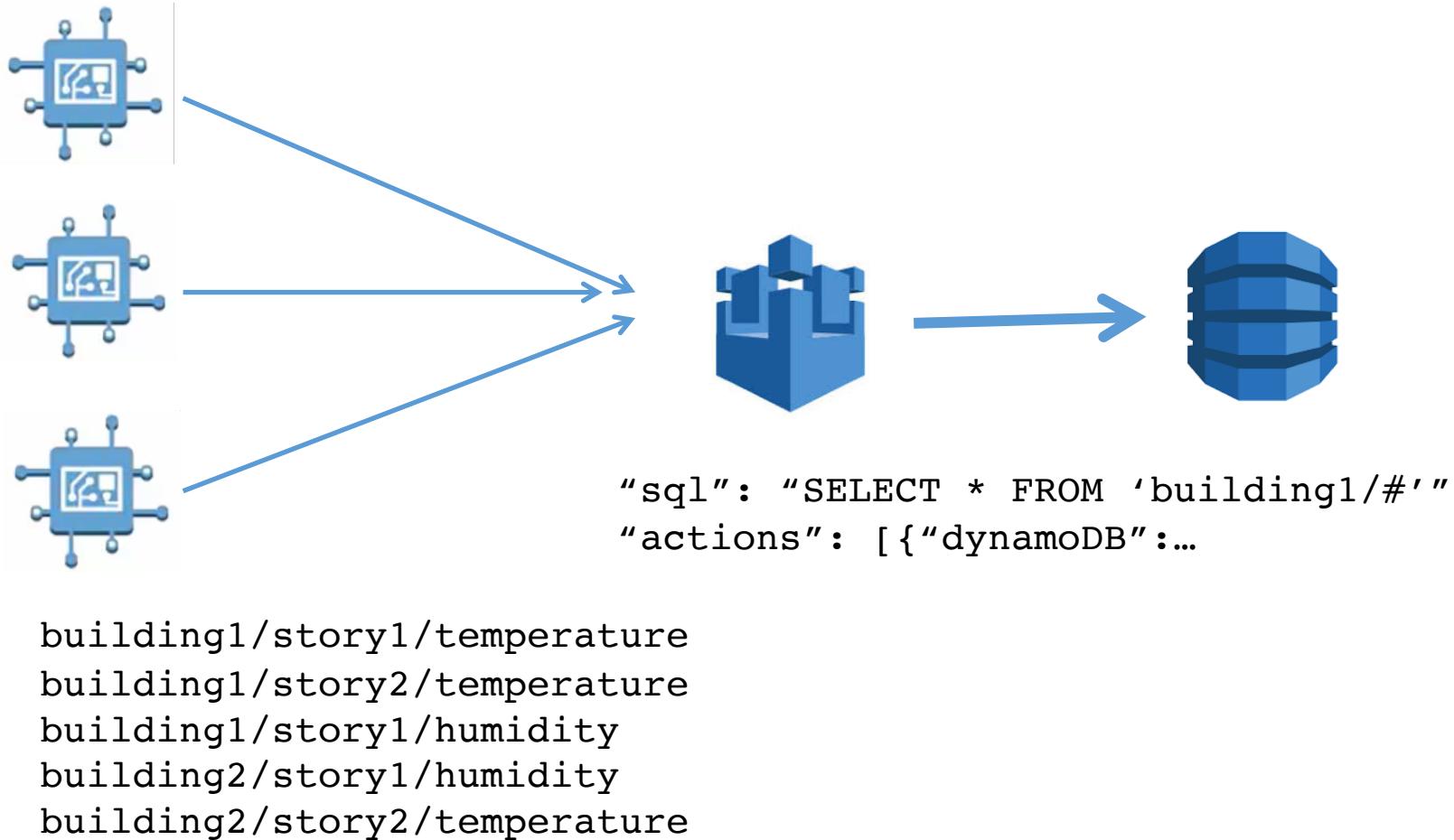
## Device Shadows



```
{  
  "state": {  
    "desired": {  
      "engine": "ON",  
      "light": "OFF"  
    }  
    "reported": {  
      "engine": "ON",  
      "light": "OFF"  
    }  
  }  
  "version": 1  
}
```

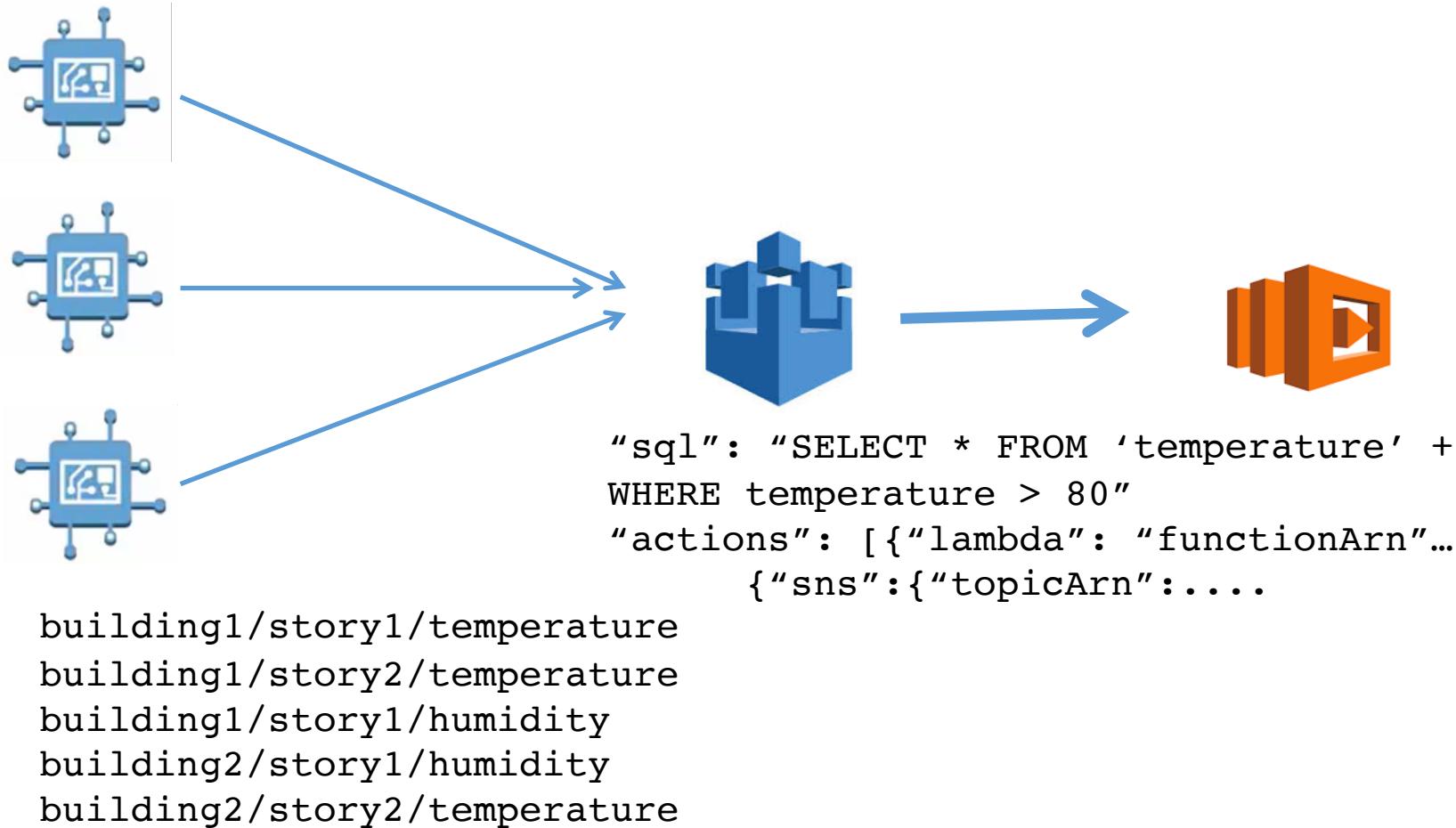
## Rules Engine

- Transform message based on rules and route to AWS Services



## Rules Engine

- Transform message based on rules and route to AWS Services



## Rules Engine

Integration:

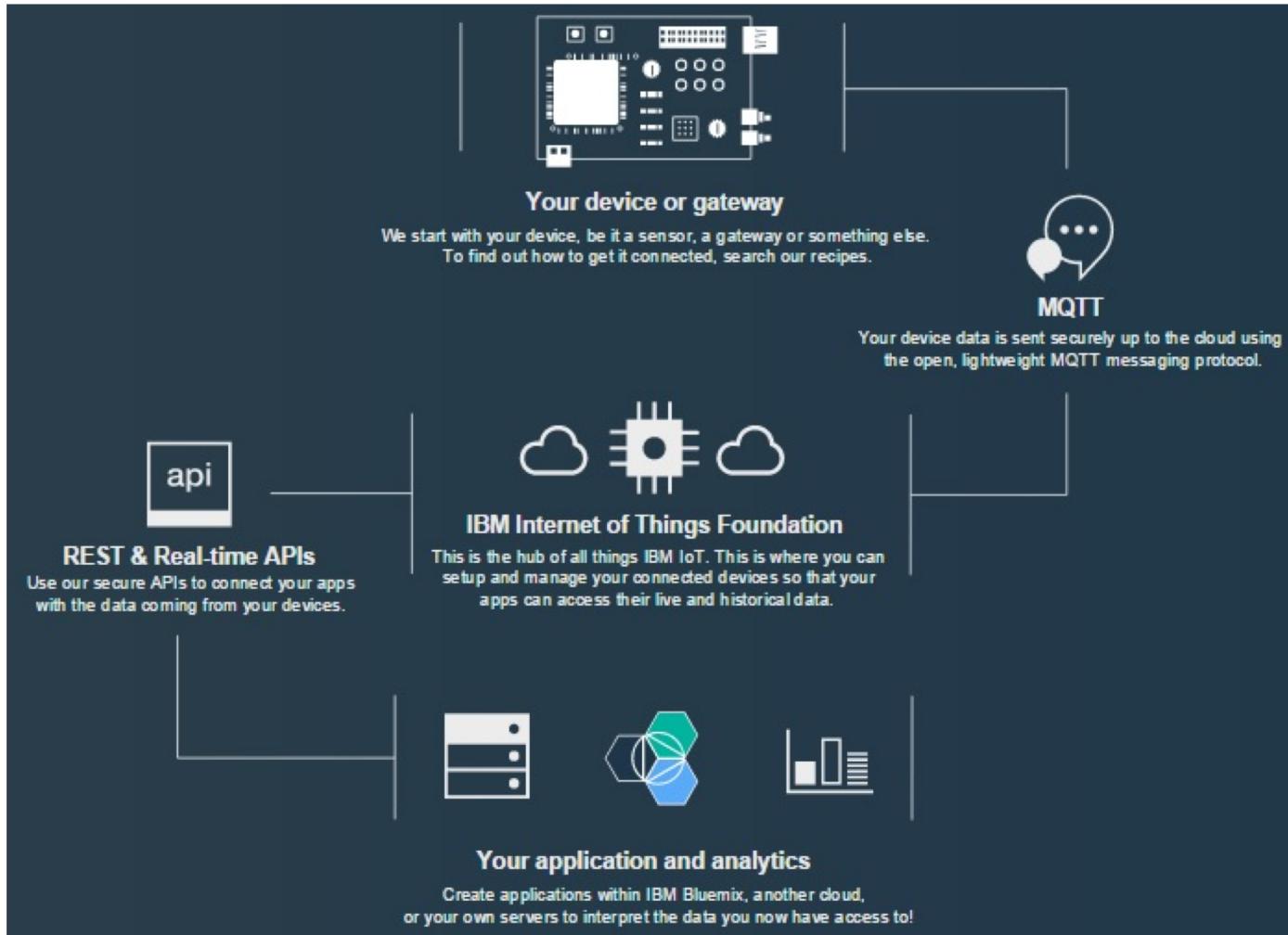
- DynamoDB
- Kinesis
- Lambda
- S3
- SNS
- Firehose
- SQS
- Republish(AWS IoT)

## Device SDKs

- C-SDK: RTOS, MCU
- JS-SDK: Embedded Linux Platforms
- Arduino:
- Mobile SDK: Android and iOS

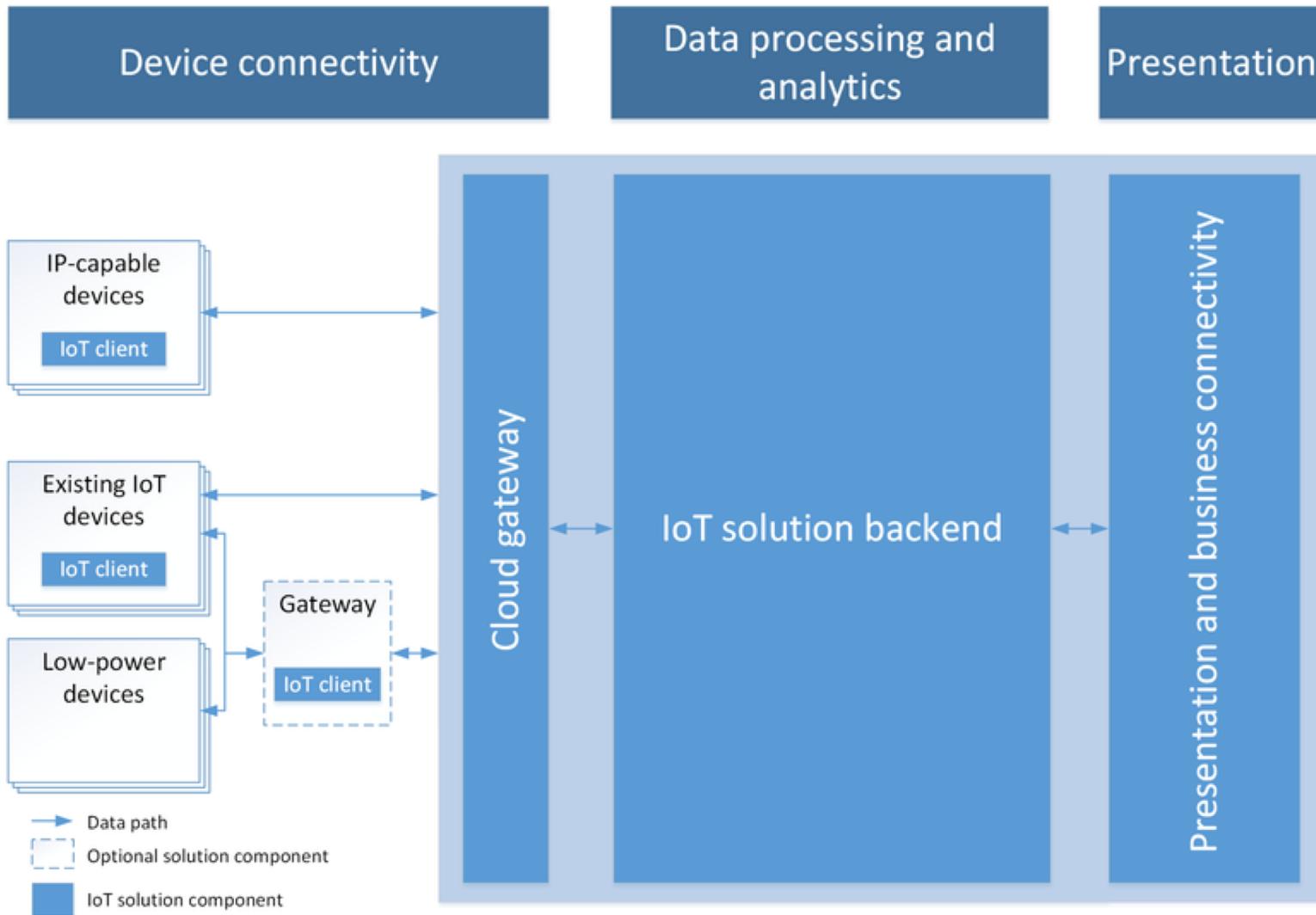
# Cloud Platforms

## IBM Bluemix



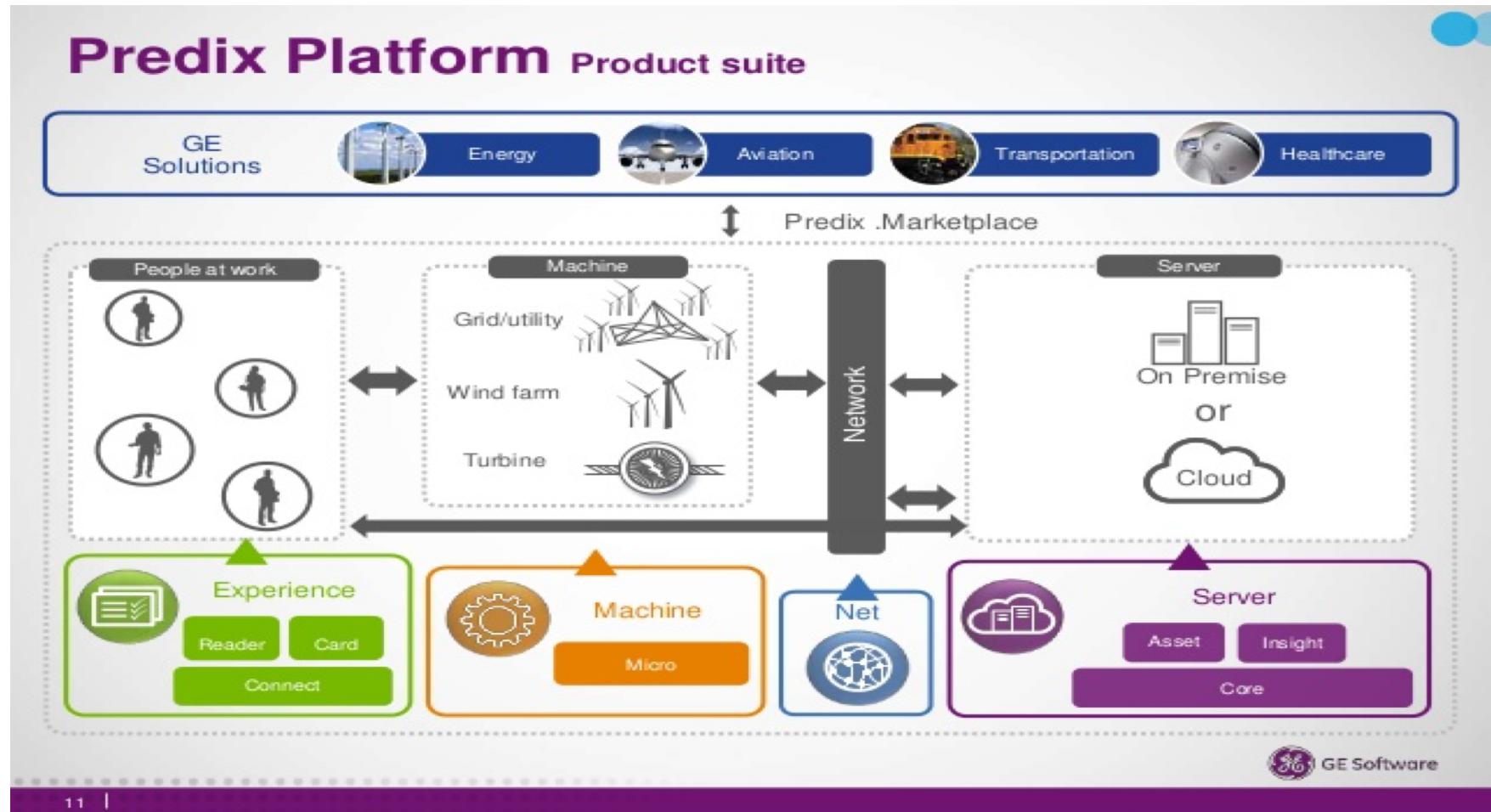
# Cloud Platforms

## Microsoft Azure



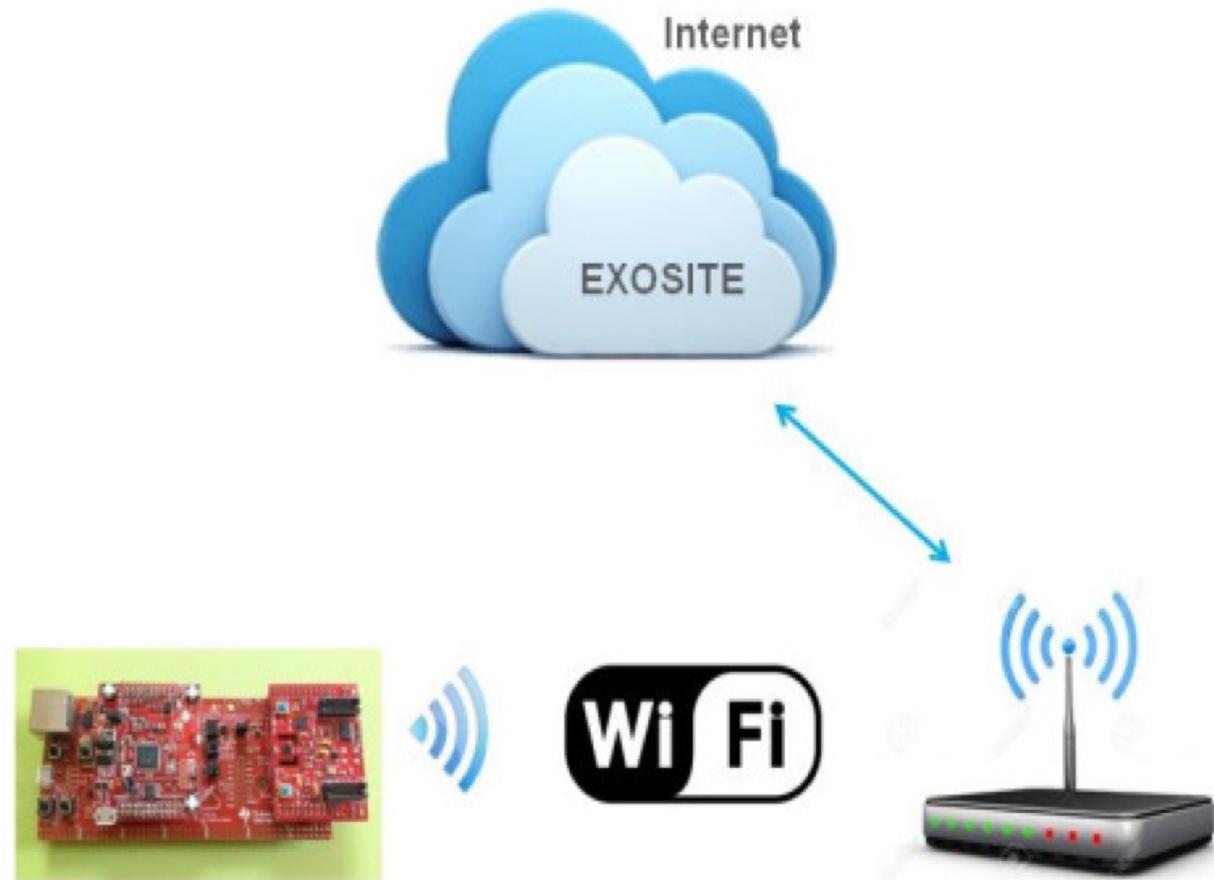
# Cloud Platforms

## GE Predix



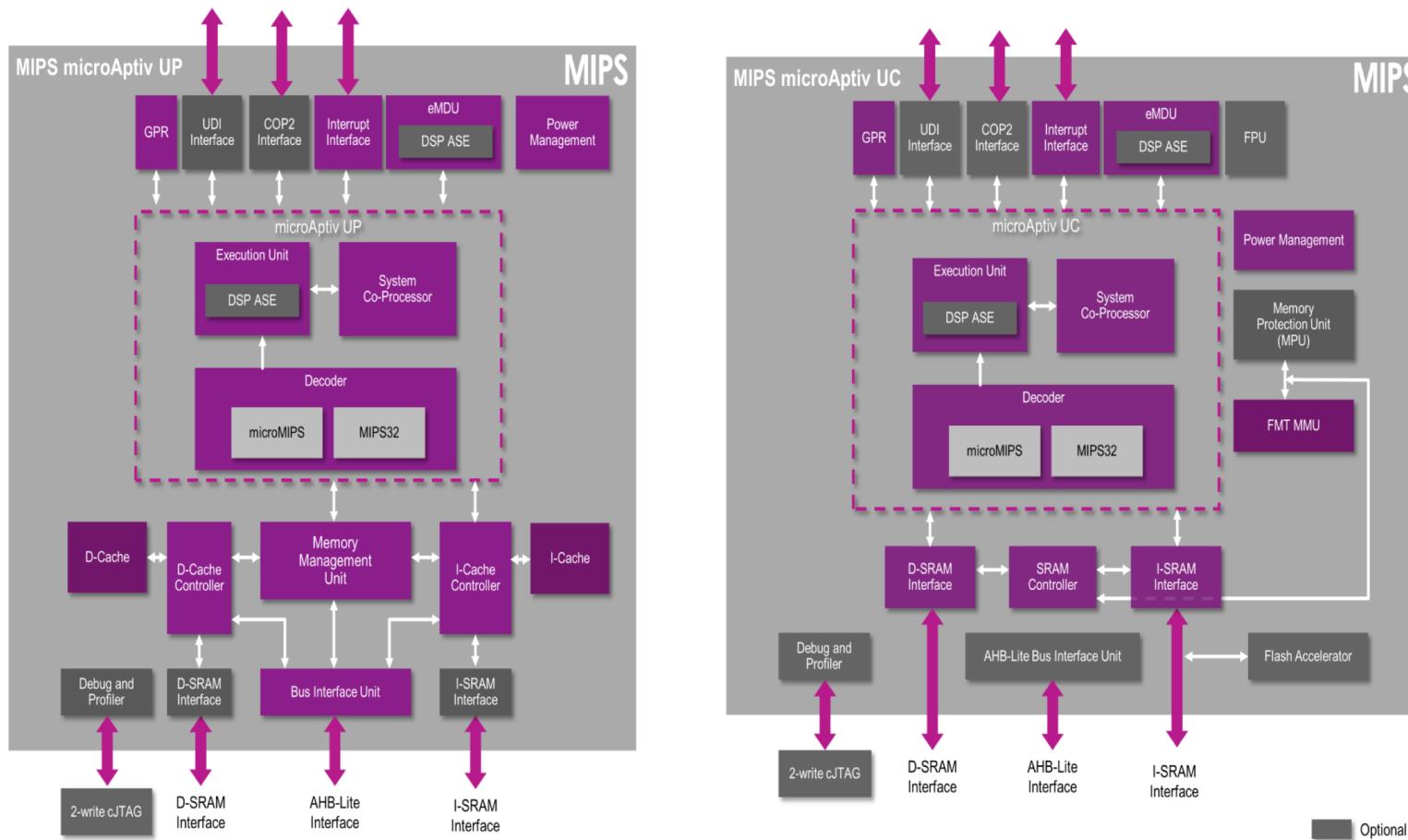
# Cloud Platforms

## Texas Instrument Exosite



# Cloud Platforms

## Samsung ARTIK



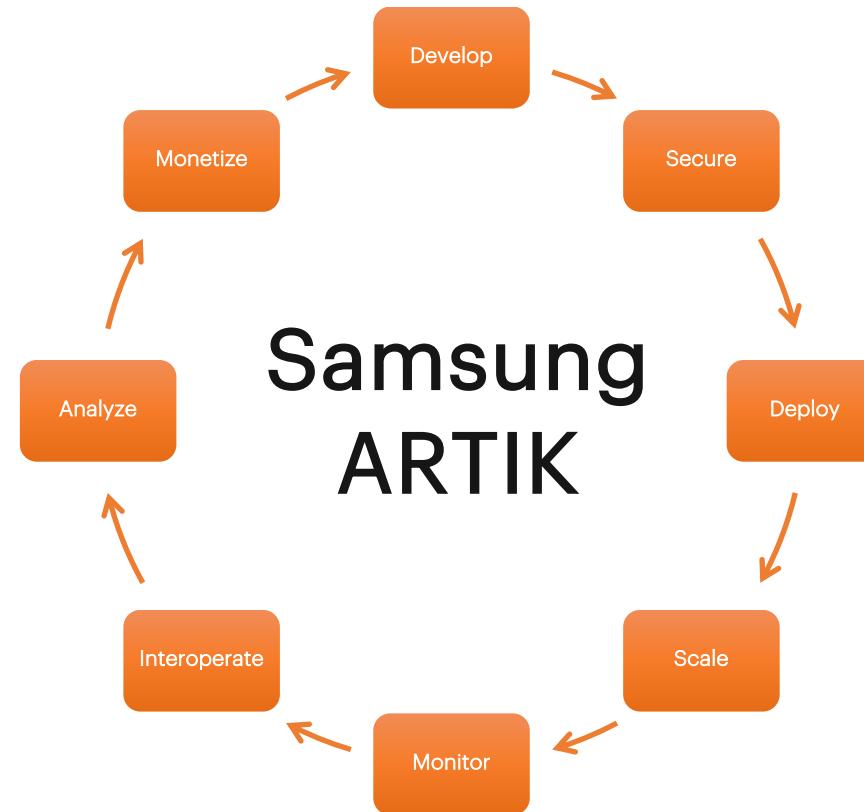
ARTIK makes it  
**easy** to  
manage the

entire IoT  
lifecycle

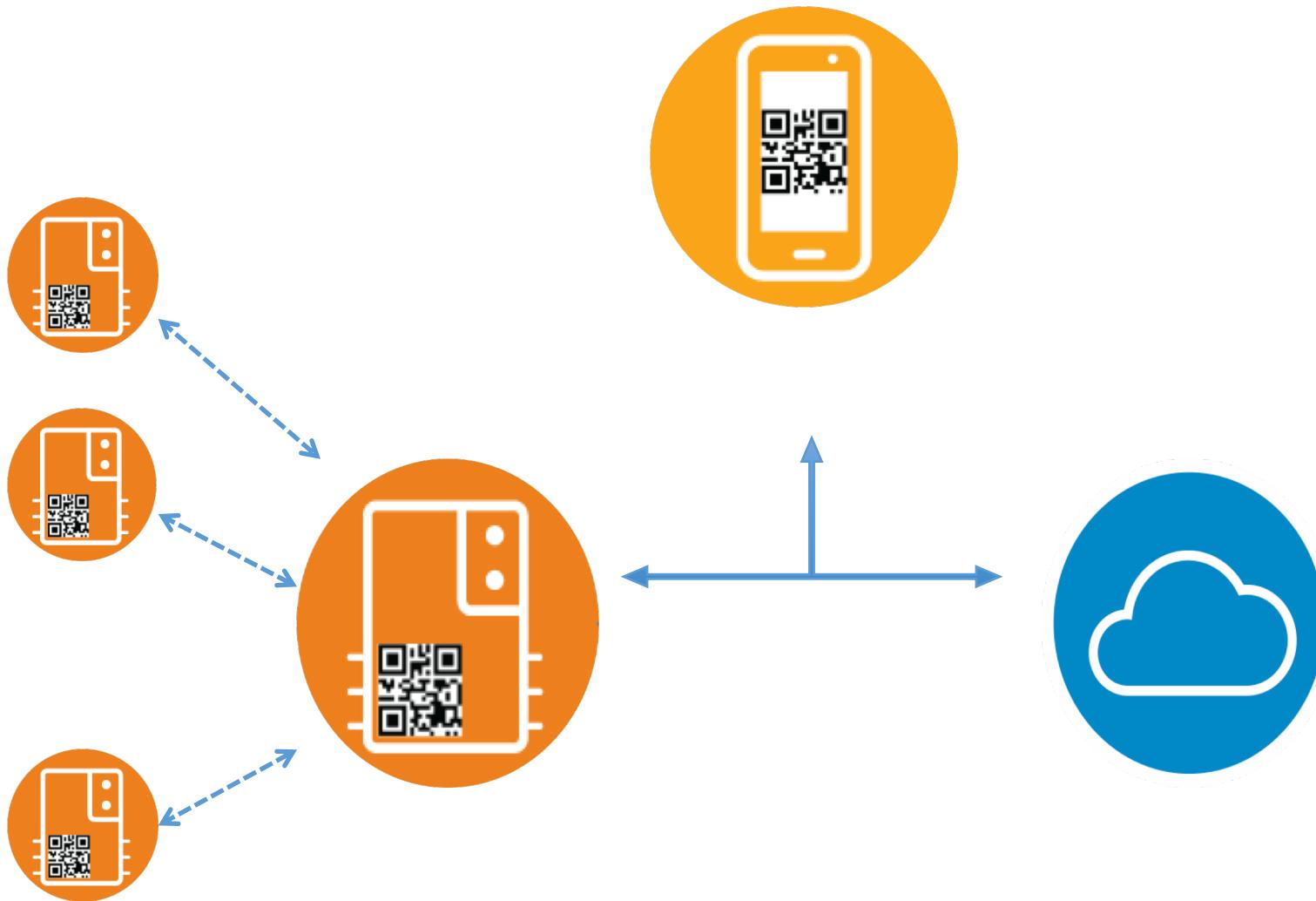
from things

to **secure** and  
interoperable

IoT products and  
solutions



# Easy onboarding



# Device Type, Manifest

ARTIK Device  
Set Up / Manifests + NEW VERSION ▼

▼ V1 CURRENT ✓ VIEW SAMPLE MESSAGE 10/11/2017 11:57

[DOWNLOAD MANIFEST](#)

### Fields

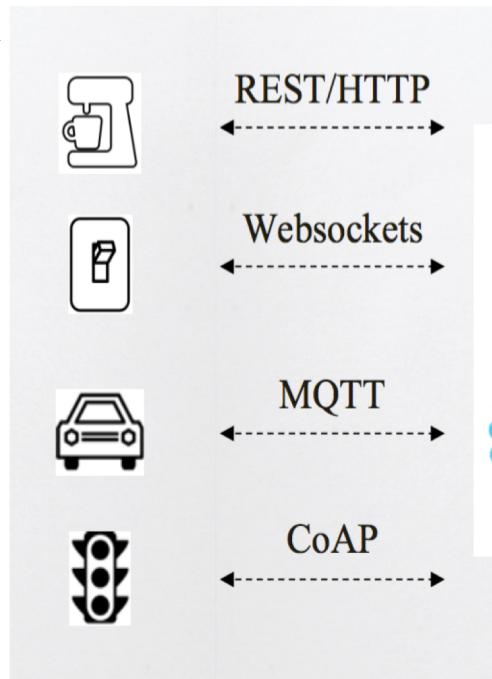
humidity	Double	None
id	String	None
lat	Double	None
long	Double	None
state	Boolean	None
temperature	Double	°C

```
{  
  "name": "temperature",  
  "type": "CUSTOM",  
  "valueClass": "Double",  
  "isCollection": false,  
  "tags": [],  
  "unitSymbol": "°C"  
}
```

# Platform / Cloud Software Development

## Platform Development:

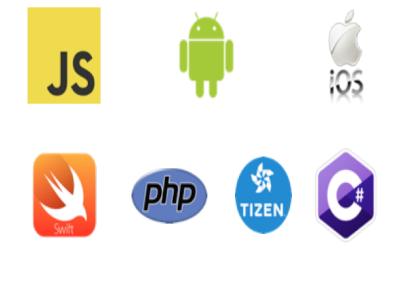
- 05x/5/7: ARTIK SDK / IDE
- Other Options for Gateway devices



## Cloud Development:

Easy to use open APIs

10 Mobile & platform SDKs



Developer Portal

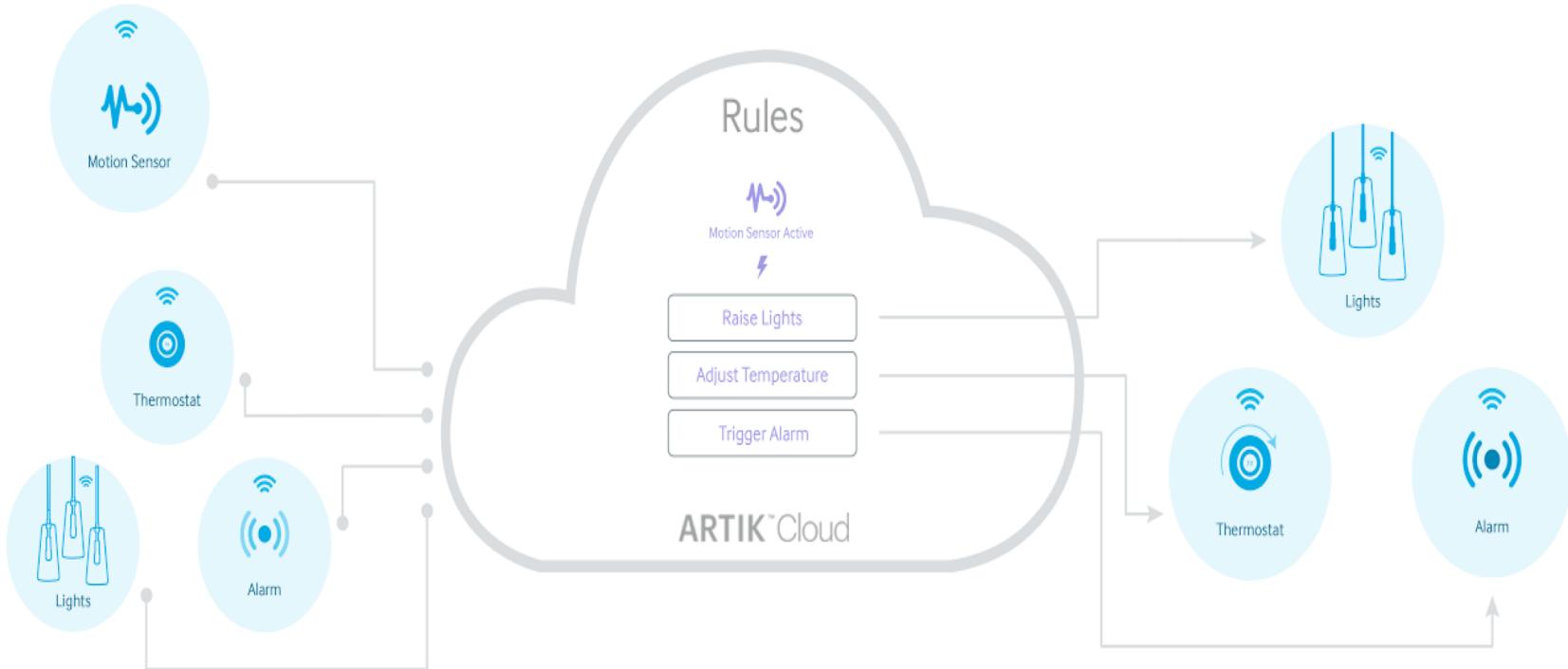
API Console

Device Simulator

# Data Visualization



# Orchestration Engine



# Rules

Choose device activity to monitor

IF

Temp Sensor - SSIC temperature X

is more than

34



F

[ADD DURATION](#)

[NEW CONDITION](#)

This rule can run at any time on any day

Send actions to your devices

THEN

ARTIK LED setOn X

[NEW ACTION](#)

# Fleet Management & Over the Air Update

- Based on Lightweight Machine to Machine(LWM2M)
- Monitors device presence, also properties like firmware version memory usage, battery consumption etc.
- OTA

The screenshot shows a web application for managing device types and performing Over-the-Air (OTA) updates. The left sidebar contains navigation links for Documentation, Community, Device Types, Applications, and a search bar. Under 'DEVICE TYPES', the 'OTA Updates' link is highlighted in blue. The main content area displays a list of known OTA update images for a device type named 'sampledevicetype1'. A red box highlights the 'UPLOAD NEW IMAGE' button. Below the table, it says 'Showing 1 to 1 of 1 OTA Update Images'.

FILE	UPDATE TYPE	VERSION	SIZE	UPLOAD TIME	
InstallationInstructionsforSSI.pdf V1 image file	System/OS	v1	704.06 KB	11/Jan/17 05:50 PM	

SEARCH FOR KNOWN OTA UPDATE IMAGES

sampledevicetype1

Device Management / OTA Update Images

SEARCH

UPLOAD NEW IMAGE

FILE

UPDATE TYPE

VERSION

SIZE

UPLOAD TIME

InstallationInstructionsforSSI.pdf  
V1 image file

System/OS

v1

704.06 KB

11/Jan/17 05:50 PM

DOCUMENTATION

Device Manifest

Platform Basics

Showing 1 to 1 of 1 OTA Update Images

1

- PTC ThingWorx is an application development platform for IoT
- Including Thingworx Foundation, Analytics, Kepware
- AR integration
- Cloud connectors to AWS, Microsoft Azure

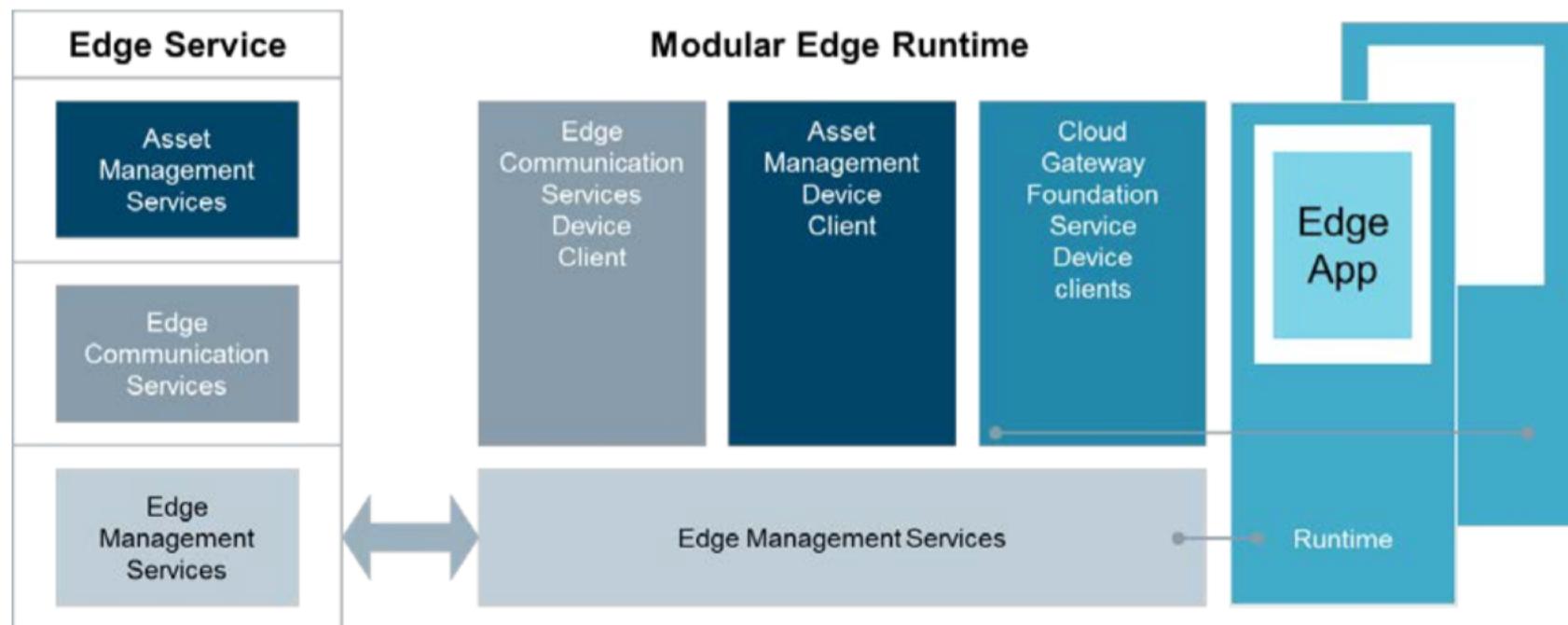


- MindSphere is the cloud-based, open IoT operating system that connects real things to the digital world, and enables powerful industry applications and digital services to drive business success.
- MindSphere is based on CloudFoundry. It is a Platform as a Service (PaaS) enables a rich partner ecosystem to develop and deliver new applications.
- Open standards and interfaces: e.g, OPC Unified Architecture(OPC UA) standard, a M2M communication protocol for industrial automation interoperability.



## MindSphere

## Edge device



## Setting Up AWS IoT

# Cloud Technologies

## Setting Up AWS IoT

1. Open AWS console and click on Sign In to the Console.

The screenshot shows the AWS homepage with a green header bar. At the top right, there is a yellow "Sign In to the Console" button. Below the header, there's a large banner for Amazon Aurora, followed by a section for AWS TechChat and other services like Migrating to AWS and Amazon Polly.

**AWS Aurora**  
Security, availability, and reliability of commercial databases at 1/10th the cost  
[Learn more »](#)

**Manage Your Resources**  
[Sign In to the Console](#)

**AWS Console Mobile App**  
View your resources on iOS and Android devices  
[Download the Mobile App »](#)

**MIGRATING TO AWS**  
Join the live webinars and download the whitepaper  
[https://aws.amazon.com/rds/aurora/start/?sc\\_icchannel=ha&sc\\_icampaign=pac\\_q3...](https://aws.amazon.com/rds/aurora/start/?sc_icchannel=ha&sc_icampaign=pac_q3...)

**DYNAMO DB AUTO SCALING**  
Automatically adjusts capacity in response to request volumes

**AWS TECHCHAT**  
Stay informed of the latest round up of AWS news and announcements.  
[Subscribe to AWS TechChat](#)

**AMAZON POLLY**  
What is text-to-speech?

# Cloud Technologies

## Setting Up AWS IoT

2. If you are a new user then select I am a new user and click on Sign in using our secure server or use the existing credentials.

The screenshot shows the AWS Sign In page at [https://signin.aws.amazon.com/signin?redirect\\_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fstate%3Dhash...](https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fstate%3Dhash...). The page features the AWS logo and a "Sign in" button. Below it, there's a field for the "Email address of your AWS account" containing "xxxxxxxxxxxx@xxxxxx.com". A "Next" button is visible. To the right, a green sidebar promotes "Migrate from Oracle to Amazon Aurora" with an icon of two databases and a "View Project »" button. At the bottom, there's a link to "About Amazon.com Sign In".

3. Then fill up the form and create and verify your account.
4. Sign in to your aws account and click on AWS IoT in the management console.
5. Now start creating thing as follows.

# Cloud Technologies

## Configuring your AWS IoT Service

6. Go to AWS IoT and open up the AWS IoT Dashboard. Now let us create a Thing on the AWS IoT dashboard by clicking on the “Registry” and “Create” button.

The screenshot shows the AWS IoT Things Registry interface. On the left, a sidebar menu is visible with options like Dashboard, Connect, Registry (which is selected and highlighted in blue), Things, Types, Greengrass, Security, Rules, Test, Software, Settings, and Learn. The main content area is titled "Things" and contains a search bar labeled "Search things". Below the search bar are three cards representing existing things:

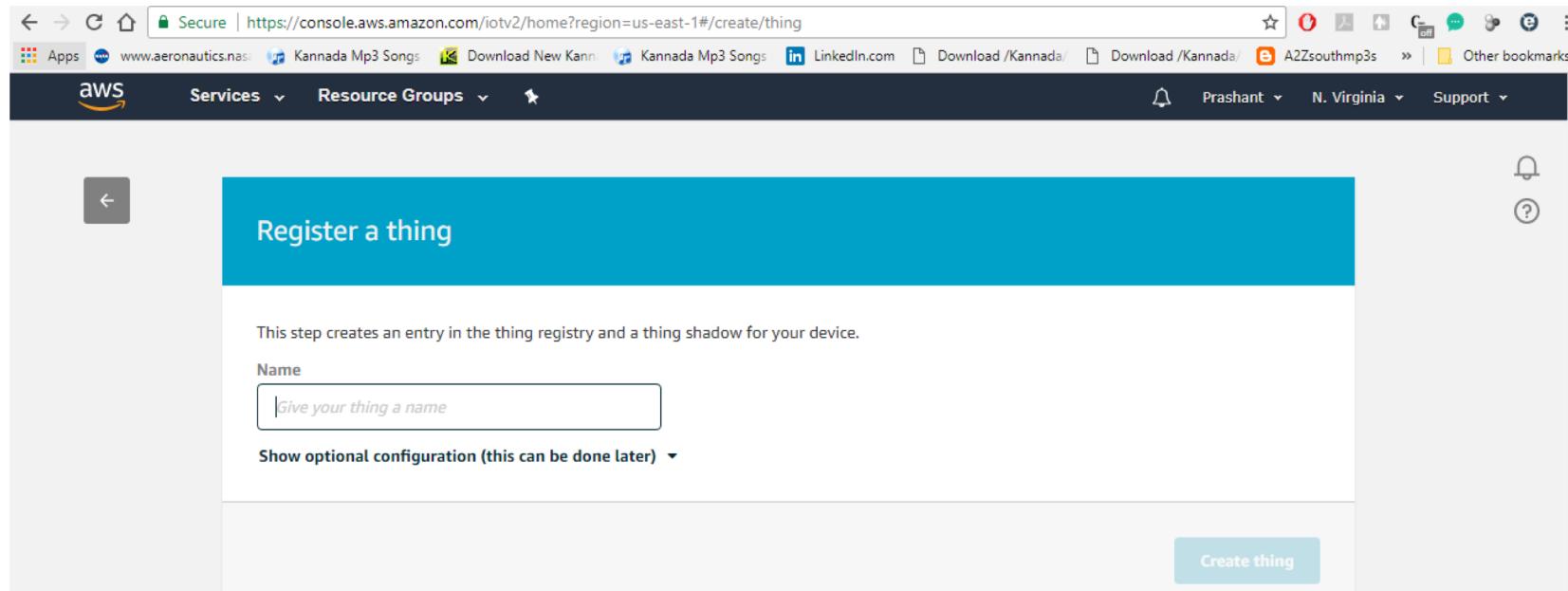
- new\_thing** (NO TYPE)
- blde** (NO TYPE)
- trx\_aws** (NO TYPE)

On the right side of the main content area, there are several buttons: "Create" (highlighted in blue), "Grid View", "List View", and a help icon. The top of the browser window shows the URL <https://console.aws.amazon.com/iotv2/home?region=us-east-1#/thinghub>.

# Cloud Technologies

## Register a Thing

7. Please make a note of this thing-name, we will have to configure the starter kit with this exact same name later on.



# Cloud Technologies

## Register a Thing

8. The thing is now created on the AWS IoT service. Clicking on Thing, will show you the contents of this thing.

The screenshot shows the AWS IoT service interface. At the top, there's a navigation bar with links for 'Secure', 'AWS', 'Services', 'Resource Groups', and user information ('Prashant', 'N. Virginia', 'Support'). Below the navigation is a dark header bar with the text 'THING', 'trx\_aws', 'NO TYPE', and 'Actions'. The main content area has a sidebar on the left with 'Details' sections for 'Security', 'Shadow', 'Interact', and 'Activity'. The 'Type' section shows 'Thing ARN' with the value 'arn:aws:iot:us-east-1:234995935387:thing/trx\_aws' and an 'Edit' button. The 'Attributes' section shows '0 Attributes' and an 'Edit' button. At the bottom, there are links for 'Feedback', 'English (US)', and legal notices: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

# Cloud Technologies

## Shadow

9. Click on “View Thing” and select “Thing name” in AWS IoT page

The screenshot shows the AWS IoT console interface. On the left, the 'Resources' section displays various AWS IoT resources. A resource named 'Led1' is highlighted with a blue border. The main panel on the right provides a detailed view of the selected 'Led1' thing. It shows the last update was 2 minutes ago, and the current shadow status is 'Out of sync'. The shadow version is 5450. The 'Shadow state' pane displays a JSON document representing the current state of the thing. The 'Shadow metadata' pane is currently empty. At the bottom, there are buttons for 'Create a rule' and 'Connect a device'.

https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/thing/Led1

AWS IoT

Resources | MQTT Client | Tutorial | Settings | 0 notifications

Learn more Detail Update shadow Edit ×

Last update 2 minutes ago

Attributes light: 4

Linked certificates Show all

Shadow status Out of sync

Shadow version 5450

Shadow state

```
8  "pb_lambda": 0,
9  "led": 0,
10 "light": 3,
11 "SEN1130P-Temprature": 0,
12 "SEN1130P-Humidity": 66,
13 "occupancy": 1,
14 "temperature": 23,
15 "BM180-Pressure": 109866,
16 "BM180-Temperature": 34.29,
17 "MHZ16-CO2": 518,
18 "MHZ16-T": 29,
19 "Gas-Q9": 773.52,
20 "Water Sensor": 536936724
21 },
22 "dew": 0,
```

Shadow metadata

Create a rule Connect a device

Feedback English

© 2008 - 2016, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy Terms of Use

## Connecting Raspberry Pi to Ubidots

# Components

- Raspberry Pi 3 model B
- 8 GB Micro SD card (Suggested)
- Micro USB power adapter
- Display monitor
- Keyboard
- Mouse

# Raspberry Pi Setup

If you have already done a basic setup of your Raspberry Pi you can jump this section, otherwise follow these steps:

1. Download the Raspberry Pi operating system, we recommend [Raspbian Wheezy](#).
2. Unzip the file and you'll end up with a large IMG file: DON'T COPY AND PASTE THIS FILE INTO YOUR SD CARD, because it won't work. You'll need to write it properly to the SD card by following these steps according to your operating system:
  1. MacOS: <http://ivanx.com/raspberrypi/>
  2. Linux: [http://elinux.org/RPi\\_Easy\\_SD\\_Card\\_Setup#Using\\_Linux\\_.28i\\_ncluding\\_on\\_a\\_Pi.21.29](http://elinux.org/RPi_Easy_SD_Card_Setup#Using_Linux_.28i_ncluding_on_a_Pi.21.29)
  3. Windows: [http://elinux.org/RPi\\_Easy\\_SD\\_Card\\_Setup#Using\\_Windows\\_7\\_or\\_Windows\\_XP](http://elinux.org/RPi_Easy_SD_Card_Setup#Using_Windows_7_or_Windows_XP)
3. Plug a keyboard and a monitor through the HDMI or TV/Analog ports.

# Configuring Raspberry Pi to use Ubidots Python API Client

1. Let's make sure your device is up to date so that it has the latest python tools (be aware that this will take a while):
  - I. *\$ sudo apt-get update*
  - II. *\$ sudo apt-get upgrade*
2. Download the pip installer and install Ubidots' Python library
  - I. *\$ sudo apt-get install python-setuptools*
  - II. *\$ sudo easy\_install pip*
  - III. *\$ sudo pip install ubidots*

# Setup a Test Variable in Ubidots

1. As a [logged in user](#) navigate to the “Data” tab.
2. Create a Data Source by clicking on the orange icon on the right. Then create a variable within that Data Source.
3. Take note of the variable’s ID to which you want to send data. For this example we’ll use a variable with the ID:  
“521d792df91b2816f35c8587”
4. Take note of your [API key](#).

# Send Data to Ubidots

1. Coming back to your Raspberry Pi:
2. Create a directory called “ubidots” where you can put this and future scripts:
  - I. *\$ mkdir ubidots*
3. Create a python script using your favorite text-editor. We’ll use “nano” in this case:
  - I. *\$ cd ubidots*
  - II. *\$ nano ubi-test.py*

# Send Data to Ubidots

Put the following code into the created file. Please note the fields where you should put your API key and your variable ID.

```
from ubidots import ApiClient
import random

#Create an "API" object

api = ApiClient("7fj39fk3044045k89fbh34rsd9823jkfs8323")

#Create a "Variable" object

test_variable = api.get_variable("521d792df91b2816f35c8587")

#Here is where you usually put the code to capture the data, either through your GPIO pins or as a calculation. We'll simply put a random value here:

test_value = random.randint(1,100)

#Write the value to your variable in Ubidots

test_variable.save_value({'value':test_value})
```

# Send Data to Ubidots

Run the code several times to send some random values to the cloud:  
\$ python ubi-test.py

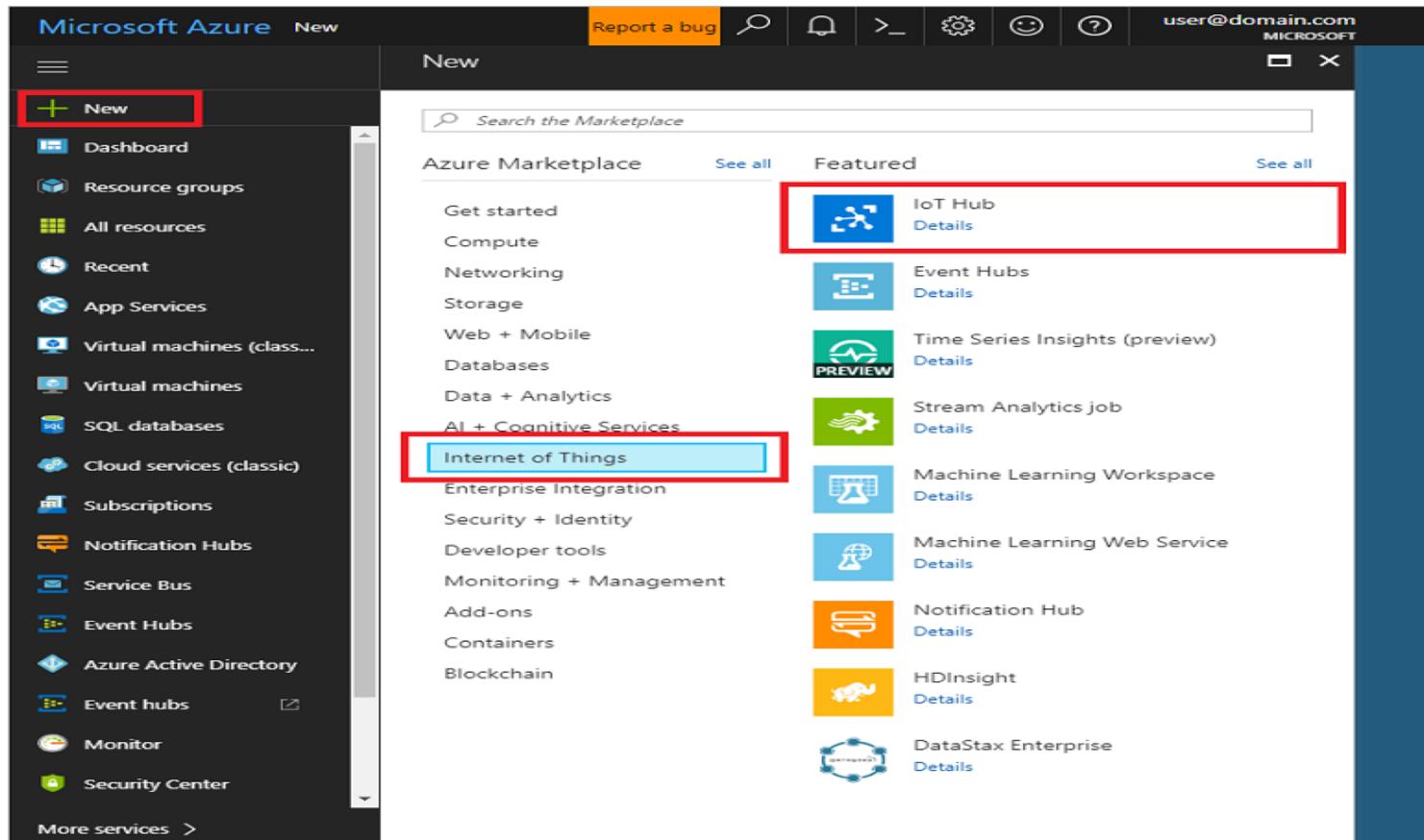
You should now see some data-points in your browser! Once your data is in the cloud, it's all down the hill. You can trigger events according to your data, create insights to share, etc.



## Connect Raspberry Pi to Azure IoT Hub (Node.js)

## Create an IoT hub

1. Sign in to the Azure portal.
2. Select New > Internet of Things > IoT Hub.



# Cloud Technologies

## Create an IoT hub

3. In the IoT hub pane, enter the following information for your IoT hub:

Name: Create a name for your IoT hub. I

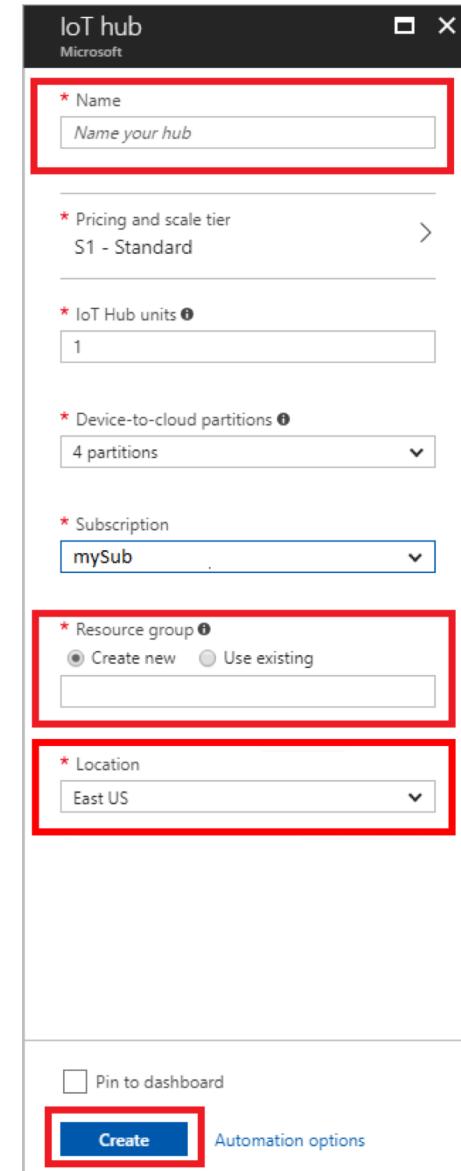
The IoT hub will be publicly discoverable as a DNS endpoint,

Pricing and scale tier: see the [Pricing and scale tier](#).

Resource group: Create a resource group to host the IoT hub [es](#)

Location: Select the closest location to you.

Pin to dashboard: Check this option for easy access



# Cloud Technologies

## Create an IoT hub

4. Click Create. Your IoT hub might take a few minutes to create. You can monitor the progress in the Notifications pane.

5. Click on the dashboard. Make a note of the Hostname, and then click Shared access policies.

The screenshot shows the Azure portal interface for an IoT hub named 'IoTGetStarted'. The left sidebar has a red box around the 'Overview' link. The main content area shows the hub's details under the 'Essentials' tab. A red box highlights the 'Hostname' field, which contains the value 'IoTGetStarted.azure-devices.net'. Below this, the 'Pricing and scale tier' is listed as 'F1 - Free' and 'IoT Hub units' as '1'. At the bottom, there is a summary card with the date '4/5/2017 UTC', the hub name 'IOTGETSTARTED', a circular progress bar showing '0% TOTAL' for devices, and the text 'MESSA...' followed by '0 / 8k'.

# Cloud Technologies

## Create an IoT hub

6. In the Shared access policies pane, click the iothubowner policy, and then copy and make a note of the Connection string of your IoT hub. For more information

The screenshot shows the Azure IoT Hub Shared access policies page. On the left, the navigation menu includes Overview, Activity log, Access control (IAM), Device Explorer, and Shared access policies (which is highlighted with a red box). Under SETTINGS, other options like Pricing and scale, Operations monitoring, IP Filter, and Properties are listed. The main content area displays a table of shared access policies. One row, 'iothubowner', is highlighted with a red box. The 'PERMISSIONS' column for this row lists 'registry write, service connect, device connect, registry read, registry write'. To the right, a detailed view of the 'iothubowner' policy is shown. It has an 'Access policy name' of 'iothubowner' and a 'Permissions' section with four checked boxes: Registry read, Registry write, Service connect, and Device connect. Below this is a 'Shared access keys' section with two keys: Primary key (value: fky+kg960fVX19XDOJ02WjNMPb6DaLhG) and Secondary key (value: bPFekUT+b/QGNdI/B/pYWs4xjnMFpJCOJ). A red box highlights the 'Connection string—primary key' section, which contains the value 'HostName=IoTGetStarted.azure-devices.r' followed by a download icon.

POLICY	PERMISSIONS
iothubowner	registry write, service connect, device connect, registry read, registry write
service	service connect
device	device connect
registryRead	registry read
registryReadWrite	registry write

iothubowner  
IoTGetStarted

Save Discard More

Access policy name  
iothubowner

Permissions

Registry read

Registry write

Service connect

Device connect

Shared access keys

Primary key

fky+kg960fVX19XDOJ02WjNMPb6DaLhG

Secondary key

bPFekUT+b/QGNdI/B/pYWs4xjnMFpJCOJ

Connection string—primary key

HostName=IoTGetStarted.azure-devices.r

# Cloud Technologies

## Register a device in the IoT hub for your device

1. In the [Azure portal](#), open your IoT hub.
2. Click Device Explorer.
3. In the Device Explorer pane, click Add to add a device to your IoT hub.
4. Then do the following:

Device ID: Enter the ID of the new device. Device IDs are case sensitive.

Authentication Type: Select Symmetric Key.

Auto Generate Keys: Select this check box.

Connect device to IoT Hub: Click Enable

# Cloud Technologies

## Register a device in the IoT hub for your device

The screenshot shows the Azure IoT Hub Device Explorer interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), and Device Explorer (which is highlighted with a red box). The main area has a search bar, a toolbar with 'Add' (also highlighted with a red box), 'Columns', 'Refresh', and 'Delete'. A message says: 'You can use this tool to view, create, update, and delete devices on your IoT Hub. You'. Below is a table with columns 'DEVICE ID' and 'STATUS', showing 'No results'. On the right, a modal window titled 'Add Device' is open. It contains fields for 'Device ID' (set to 'new-device'), 'Authentication Type' (set to 'Symmetric Key'), 'Primary Key' (placeholder 'Enter your primary key here'), 'Secondary Key' (placeholder 'Enter your secondary key here'), and 'Auto Generate Keys' (checkbox checked). At the bottom of the modal are 'Enable' and 'Disable' buttons.

- Click Save.
- After the device is created, open the device in the Device Explorer pane.

# Cloud Technologies

## Register a device in the IoT hub for your device

- Make a note of the primary key of the connection string.

The screenshot shows the Azure IoT Hub Device Explorer interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), and Device Explorer (which is selected). The main area has a search bar and buttons for Add, Columns, Refresh, and Delete. A message says, "You can use this tool to view, create, update, and delete devices on your IoT Hub. You can also generate connection strings for your devices here." Below is a table with columns DEVICE ID and STATUS. One row shows a device with ID "new-device" and status "enabled". This row is highlighted with a red box. To the right is a "Device Details" panel for "Device Id: new-device". It shows the Primary key (value: d0qltgHj6U8Wb+3PX5l9ism5elGtJLRTb89) and Secondary key (value: lRibWxq/0Z77553R1XIXIEAryKFJXBZE8nil1). A large red box highlights the "Connection string—primary key" section, which contains the value: HostName=IoTGetStarted.azure-devices.r. Below it is the "Connection string—secondary key" section with the same value. At the bottom of the panel are "Enable" and "Disable" buttons.

## Connecting Raspberry Pi to IBM Bluemix

# Cloud Technologies

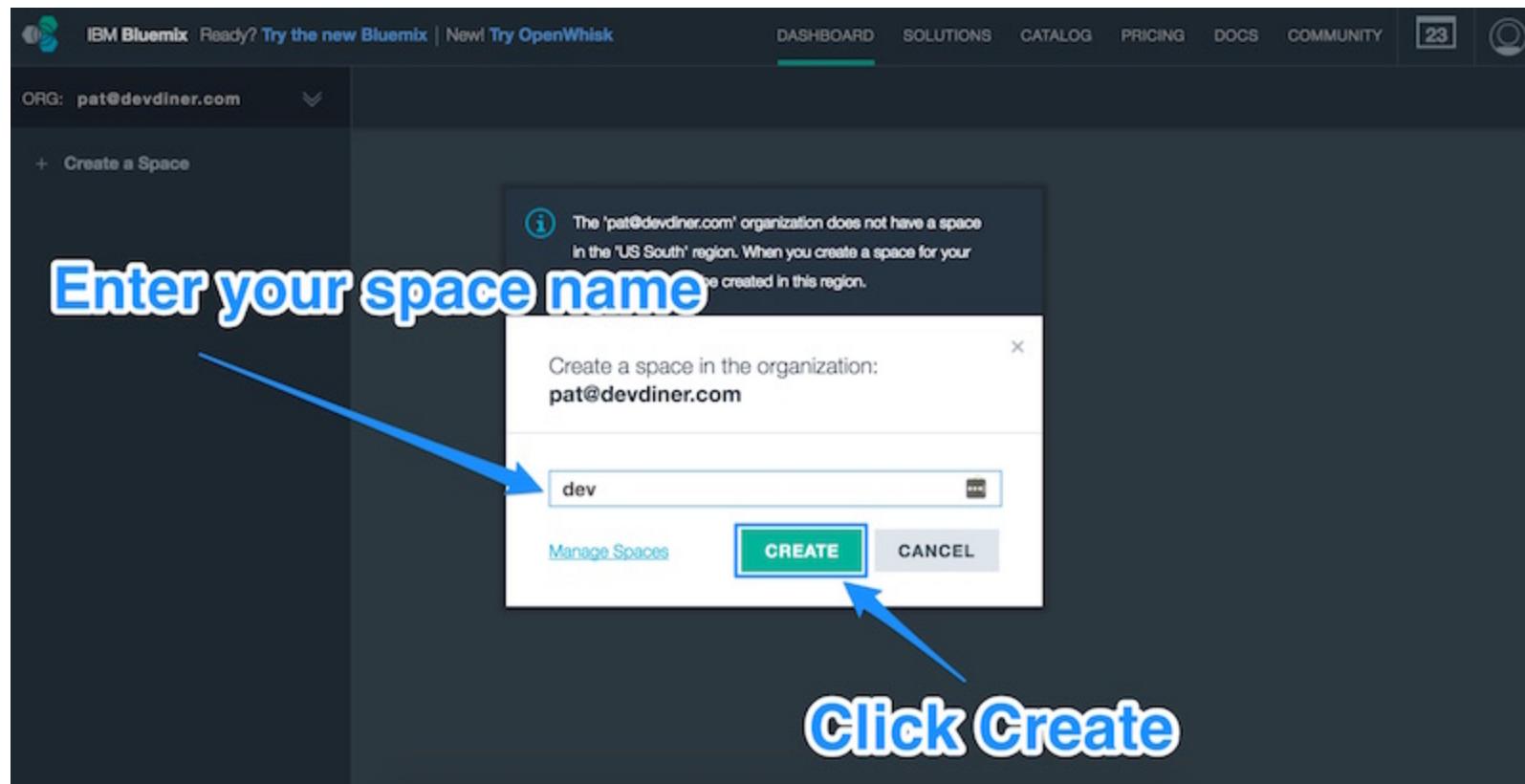
## Connecting Raspberry Pi to IBM Bluemix

- Login to Bluemix - <https://console.ng.bluemix.net/login>

The image shows the IBM Bluemix dashboard. At the top, it says "IBM Bluemix Ready? Try the new Bluemix | New! Try OpenWhisk". The navigation bar includes DASHBOARD, SOLUTIONS, CATALOG, and PF. On the left, there's a sidebar with "ORG: pat@devdiner.com" and a list of resources: "Create a Space", "dev" (selected), "CF APPS (0)", "SERVICES (0)", "CONTAINERS (0)", and "VIRTUAL SERVERS (0)". The main area has a large blue button with the text "Choose your region". Below it are four cards: "Cloud Foundry Apps" (0 B/2 GB Used), "Containers" (0 B/2 GB, 0/0 Public IPs Requested | 0 Used), "Data & Analytics" (WORK WITH DATA), and "Services" (0/10 Used). A blue arrow points from the text "Choose your region" to the "Region" dropdown in the top right corner, which is set to "US South". The top right also shows the user profile "Patrick Catanzariti pat@devdiner.com" with links for Account, Status, and Log Out. Below the profile are sections for "Manage Organizations", "Notifications" (You have no new notifications), and "Support" with options for "Get help" and "Submit an Idea". At the bottom, it says "You haven't created anything yet. Create an app or browse the Bluemix catalog".

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

The screenshot shows the IBM Bluemix dashboard. On the left, there's a sidebar for the organization 'pat@devdiner.com' under the 'dev' space, listing 'CF APPS (0)', 'SERVICES (0)', 'CONTAINERS (0)', and 'VIRTUAL SERVERS (0)'. The main area has several cards: 'Cloud Foundry Apps' (0 B/2 GB Used), 'Containers' (0 B/2 GB, 0/0 Public IPs Requested | 0 Used), 'Virtual Servers' (0 B/0 B | 0/0 Public IPs), 'Data & Analytics' (WORK WITH DATA), and a highlighted card 'Services & APIs' (100+ Services, 0/10 Used). A blue arrow points from the text 'Then we choose "Use Services or APIs"' to the 'USE SERVICES OR APIs' button in the 'Services & APIs' card. A message at the bottom says 'You haven't created anything yet. Create an app or browse the [Bluemix Catalog](#) to get started.'

IBM Bluemix Ready? Try the new Bluemix | New! Try OpenWhisk

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY 23

ORG: pat@devdiner.com

+ Create a Space

dev

- CF APPS (0)
- SERVICES (0)
- CONTAINERS (0)
- VIRTUAL SERVERS (0)

Cloud Foundry Apps  
0 B/2 GB Used

Containers  
0 B/2 GB  
0/0 Public IPs Requested | 0 Used

Virtual Servers  
0 B/0 B | 0/0 Public IPs

CREATE APP START CONTAINERS RUN VIRTUAL SERVERS

Data & Analytics  
WORK WITH DATA

100+ Services & APIs  
0/10 Used

USE SERVICES OR APIs

You haven't created anything yet. Create an app or browse the [Bluemix Catalog](#) to get started.

Then we choose "Use Services or APIs"

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

The screenshot shows the IBM Bluemix Catalog interface. At the top, there is a navigation bar with links for DASHBOARD, SOLUTIONS, CATALOG (which is underlined), PRICING, DOCS, and COMMUNITY. A user icon with '23' notifications is also present. The main area has a dark background with a search bar containing the text 'internet of things platform'. To the left, a sidebar lists various service categories like Watson, Mobile, DevOps, etc., with 'Internet of Things' highlighted by a blue box and an arrow pointing to it from the text 'We find the "Internet of Things Platform"'. In the center, there is a card for the 'Internet of Things Platform' by IBM, featuring a hexagonal icon with a gear inside. Below the card, the text 'A new generation of applications' is visible. To the right of the card, the text 'We find the "Internet of Things Platform"' is displayed in large blue letters, followed by 'Select it!' with another arrow pointing to the card. At the bottom of the catalog page, there is a section titled 'You could also check this box to filter options too, rather than typing' with a checkbox labeled 'Looking for more?'. A note at the very bottom says 'Check out the Bluemix Labs Catalog to try out experimental runtimes and services.'

We find the  
"Internet of Things  
Platform"

Select it!

You could also check this box to filter  
options too, rather than typing

Looking for more?

Check out the Bluemix Labs Catalog to try out experimental runtimes and services.

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

The screenshot shows the IBM Bluemix Catalog interface. At the top, there's a navigation bar with links for 'IBM Bluemix', 'Ready? Try the new Bluemix | New! Try OpenWhisk', 'DASHBOARD', 'SOLUTIONS', 'CATALOG' (which is underlined in blue), 'PRICING', 'DOCS', 'COMMUNITY', a '23' badge, and a user profile icon.

In the main content area, there's a card for the 'Internet of Things Platform' service. The card includes:

- Icon:** A hexagonal icon with a gear inside.
- Name:** Internet of Things Platform
- Author:** IBM
- Type:** Service
- Location:** US South
- Publish Date:** 03/30/2016
- View Docs:** A button to view documentation.

The card also contains a brief description: "The IBM Internet of Things service lets your apps communicate with and consume data collected by your connected devices, sensors, and gateways. Our recipes make it super easy to get devices connected to our Internet of Things cloud. Your apps can then use our real-time and REST APIs to communicate with your devices and consume the data you've set them up to collect."

Below the description are two bullet points:

- **Connect your devices securely to the cloud**
- **Build an app that talks to your devices**

Under each bullet point is a brief explanatory text and a small diagram or screenshot illustrating the process.

To the right of the card, there's a sidebar titled 'Add Service' with fields for 'Space' (set to 'dev'), 'App' (set to 'Leave unbound'), 'Service name' (set to 'Internet of Things Platform-ya'), and a 'Selected Plan' dropdown (set to 'Free'). A large green 'CREATE' button is at the bottom of this sidebar. An arrow points from the text 'Click "Create"' to this button.

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

The screenshot shows the IBM Bluemix interface for the Internet of Things Platform. The top navigation bar includes links for 'IBM Bluemix Ready? Try the new Bluemix | New! Try OpenWhisk', 'DASHBOARD' (which is underlined), 'SOLUTIONS', 'CATALOG', 'PRICING', 'DOCS', 'COMMUNITY', and a notifications icon showing '23'. Below the navigation is a breadcrumb trail 'Back to Dashboard... > Internet of Things Platform-ya'. The main content area is titled 'Internet of Things Platform-ya' and features three main sections: 'Connect your devices', 'Learn how to build', and 'Learn how to extend your app'. A large blue arrow points from the text 'We then click to launch our dashboard' down to a blue-bordered button labeled 'Launch dashboard'.

We then click to launch our dashboard

Launch dashboard

Connect your devices

Use our recipes to find out how to add your devices. We work with partners and have sample connection recipes for many devices.

Launch the Watson IoT Platform dashboard and add your devices by clicking the 'Add Device' button under the 'Devices' tab.

Learn how to build

When you have added your devices, you can come back to Bluemix to start building your app using your real-time and historical device data.

Read the docs to find out how to make the most out of your app.

Go to docs

Learn how to extend your app

Use other Bluemix services to extend your app to start creating a great Internet of Things app.

Here are some of the services you could use:

- Twilio (Third Party)
- Cloudant NoSQL DB (IBM)
- Dash DB (IBM)
- Geospatial Analytics (IBM)
- Time Series Database (IBM)
- IBM Analytics for Hadoop (IBM)

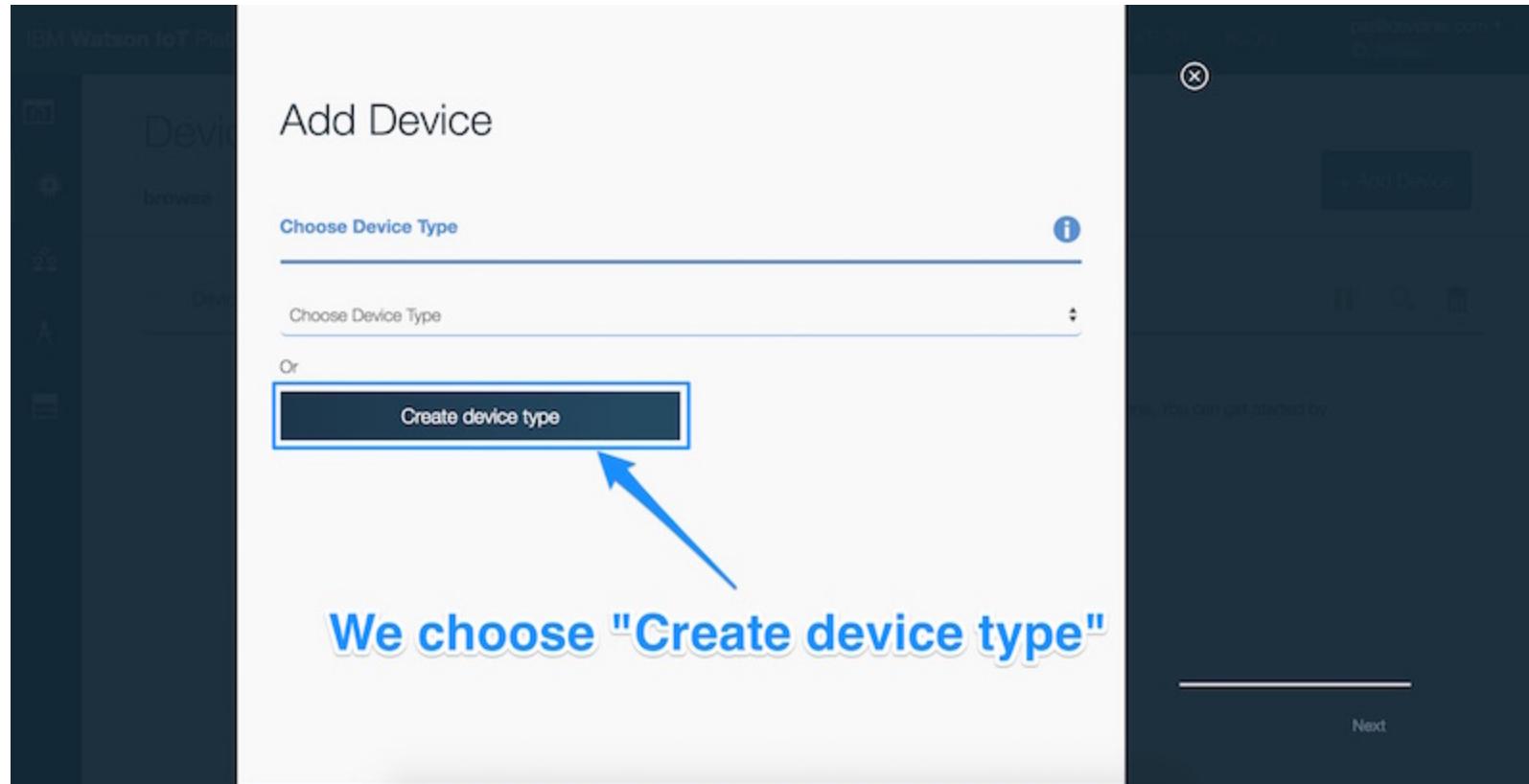
# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

The screenshot shows the IBM Watson IoT Platform Overview page. At the top, there is a navigation bar with links for QUICKSTART, SERVICE STATUS, DOCUMENTATION, and BLOG, along with a user account section. Below the navigation bar, there is a sidebar with icons for Home, Settings, Devices, Analytics, and Data. The main content area is titled "Overview". It features four cards: "DEVICE TYPES" (No devices have been added, with an "Add Device" button highlighted by a blue arrow and the text "We want to add our Raspberry Pi"), "STORAGE" (Storage used today: 0.0 MB, This month: 0.0 MB, Previous month: 0.0 MB), "DATA CONSUMED" (0.0 MB Data traffic consumed today), and "DEVICE" (0 Currently registered). A large blue arrow points from the text "We want to add our Raspberry Pi" towards the "Add Device" button.

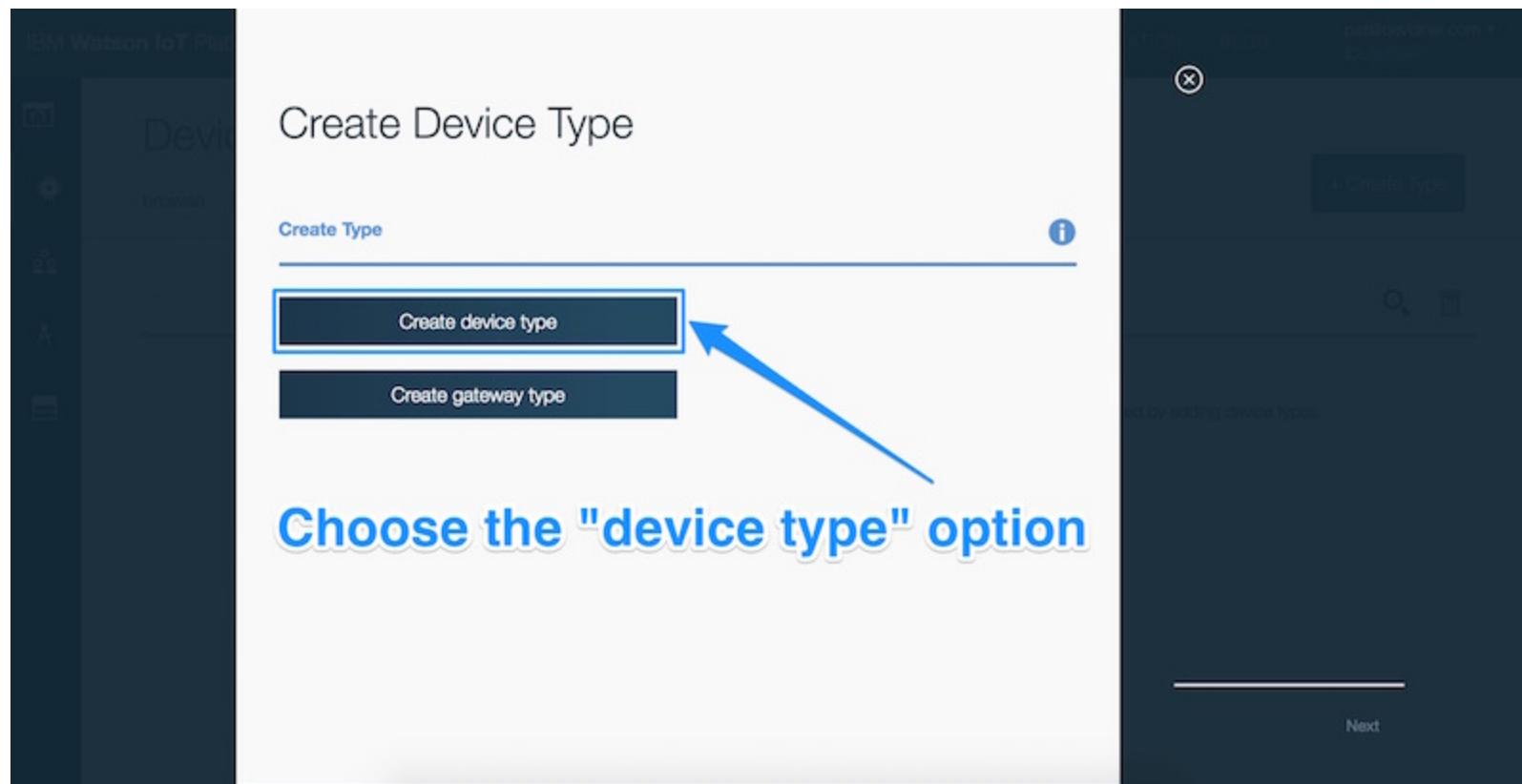
# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

Create Device Type

We enter in a device type name in lowercase separated by commas

Name: the-greatest-pis-in-the-world

Description: My wonderful IBM powered Raspberry Pis

The device type name is used to identify the device type uniquely, using a restricted set of characters to make it suitable for API use.

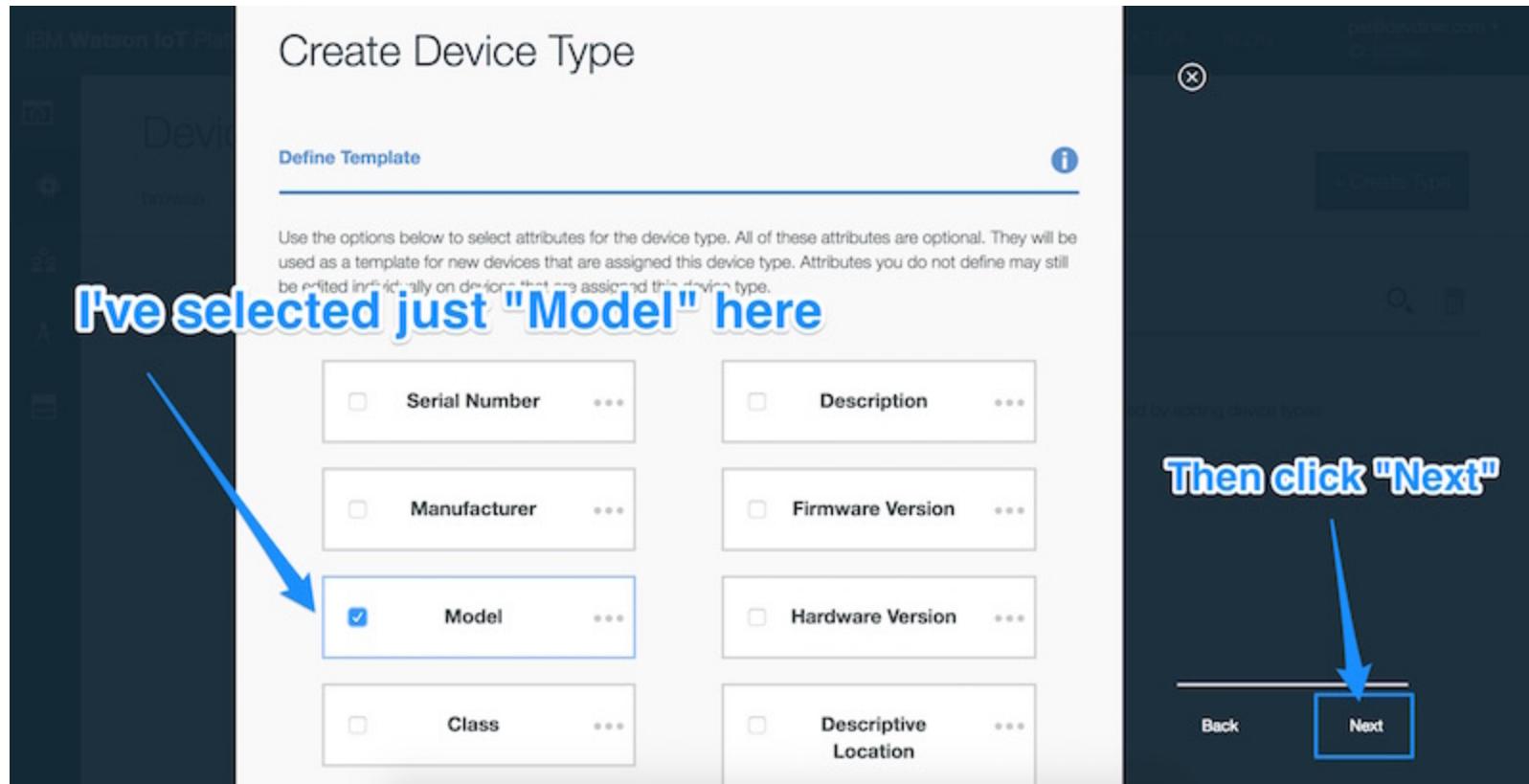
The device type description can be used for a more descriptive way of identifying the device type.

Then enter in a more human readable description

Then click "Next"

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

IBM Watson IoT Platform

Create Device Type

Submit Information

You must now set values for the attributes you have selected for this device type. The values of these attributes will act as a template for new devices that are assigned this device type. You can override these values when adding individual devices.

Model

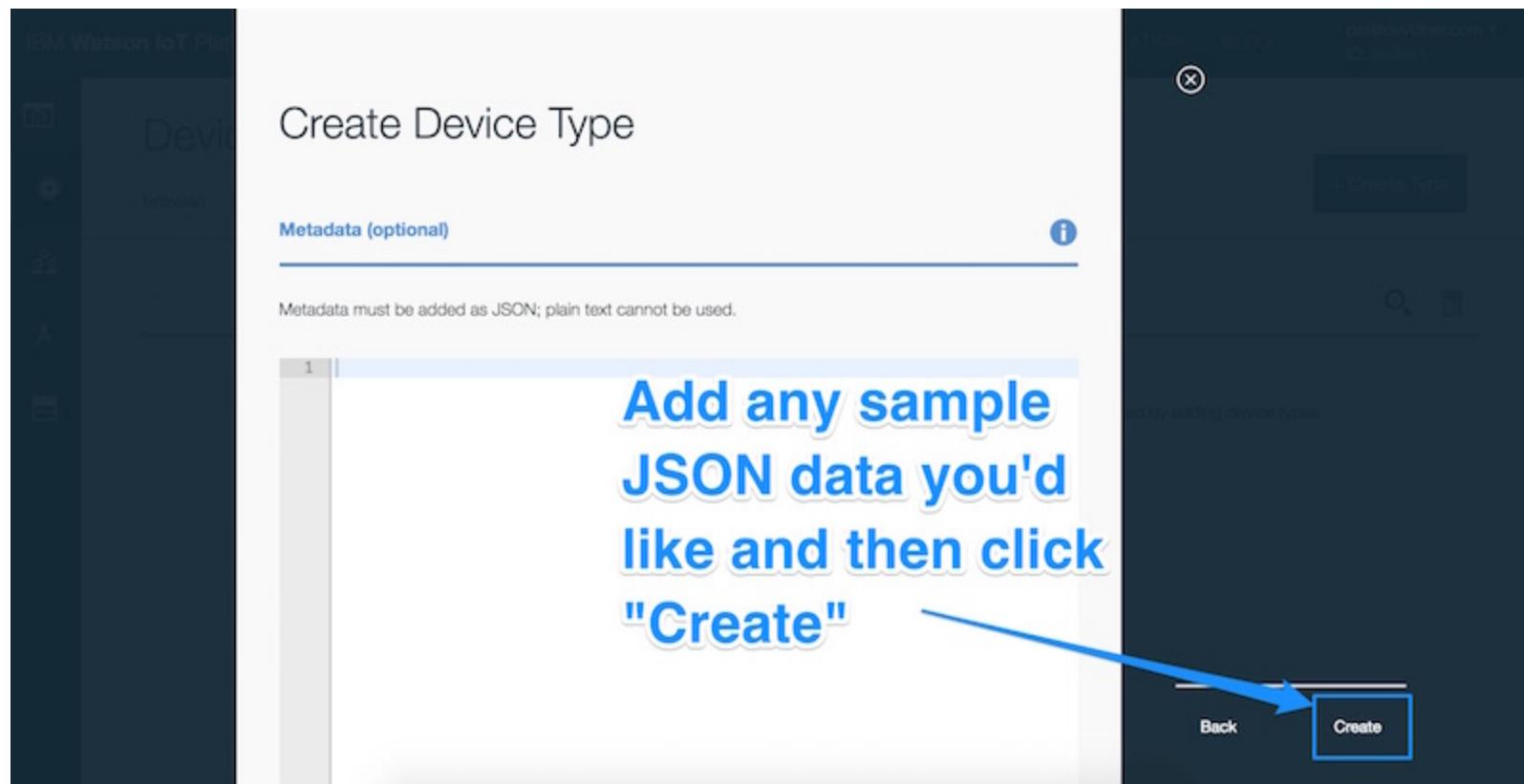
Enter in your device's model name.  
Mine was "Raspberry Pi 3 Model B"

Then click "Next"

Back

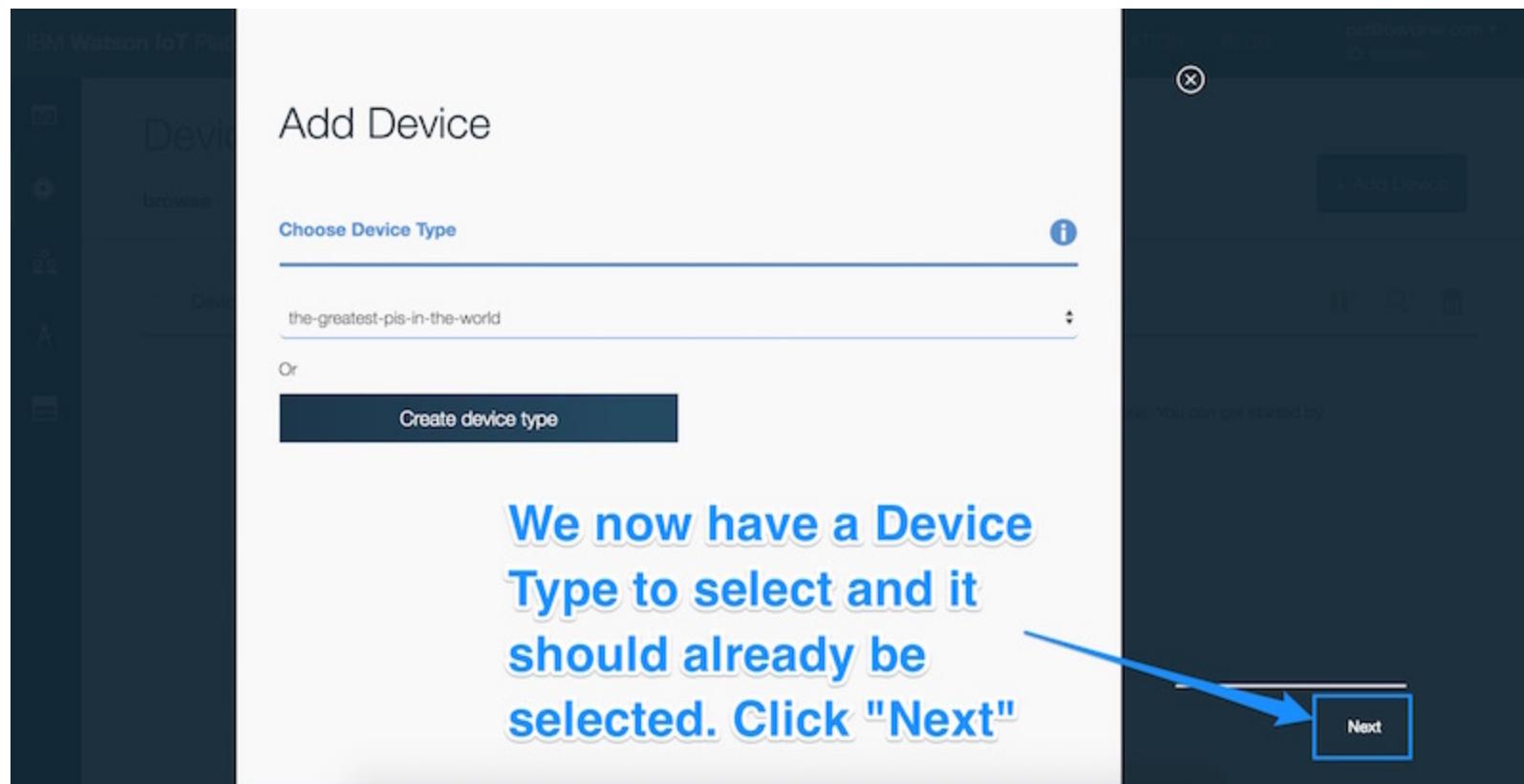
# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



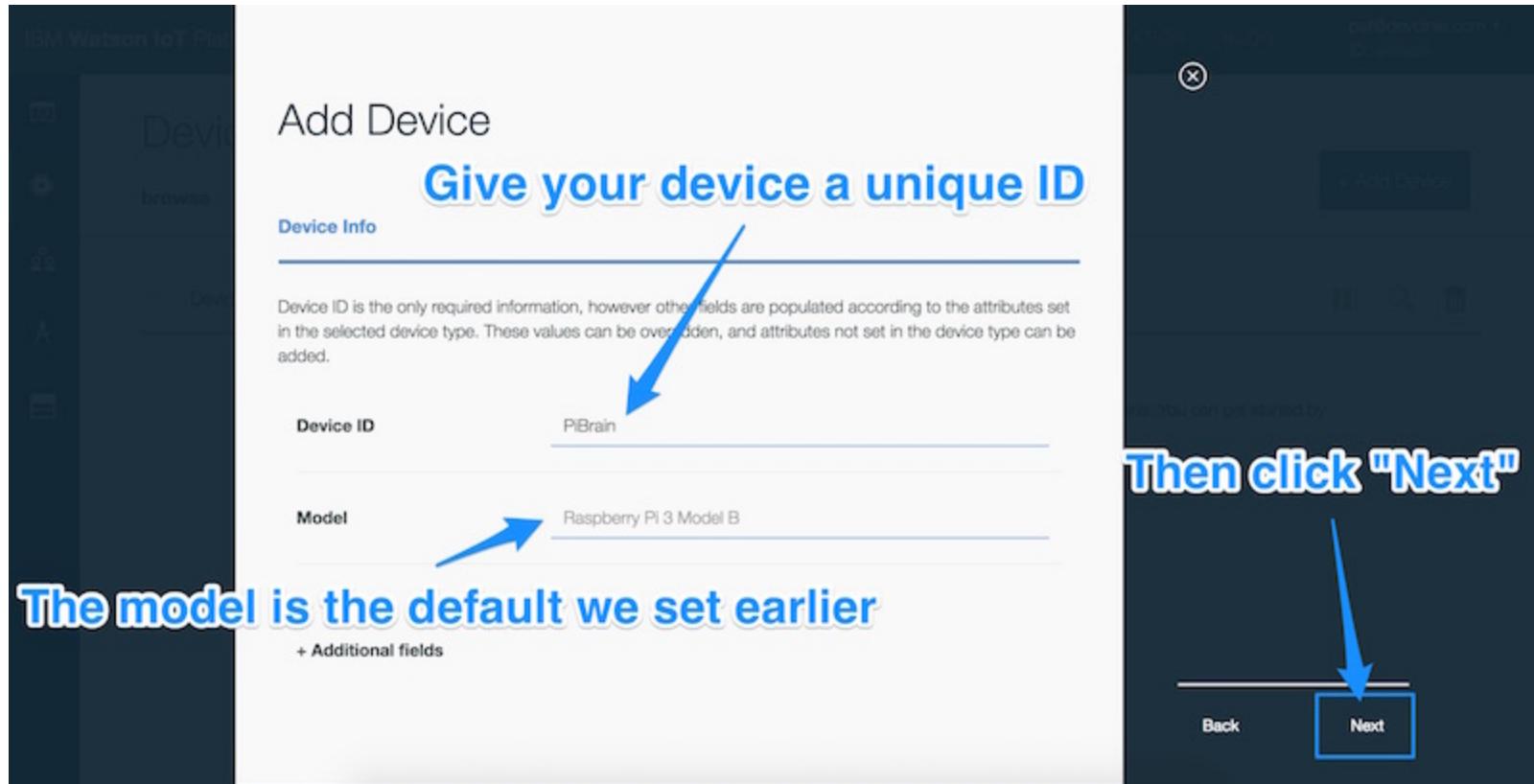
# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



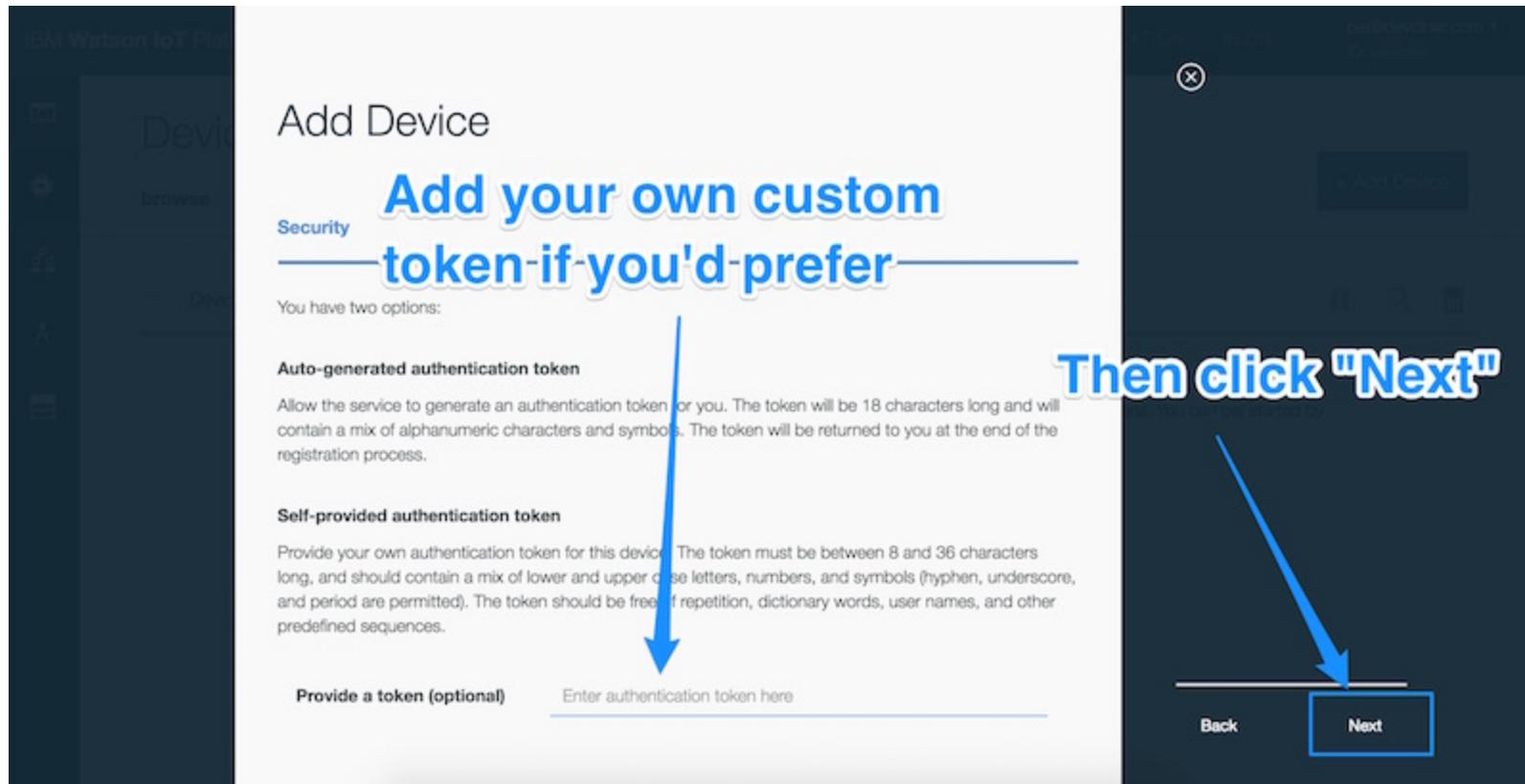
# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix

IBM Watson IoT Platform

### Add Device

**Summary**

Please check that all submitted information for this device is correct before adding this device.

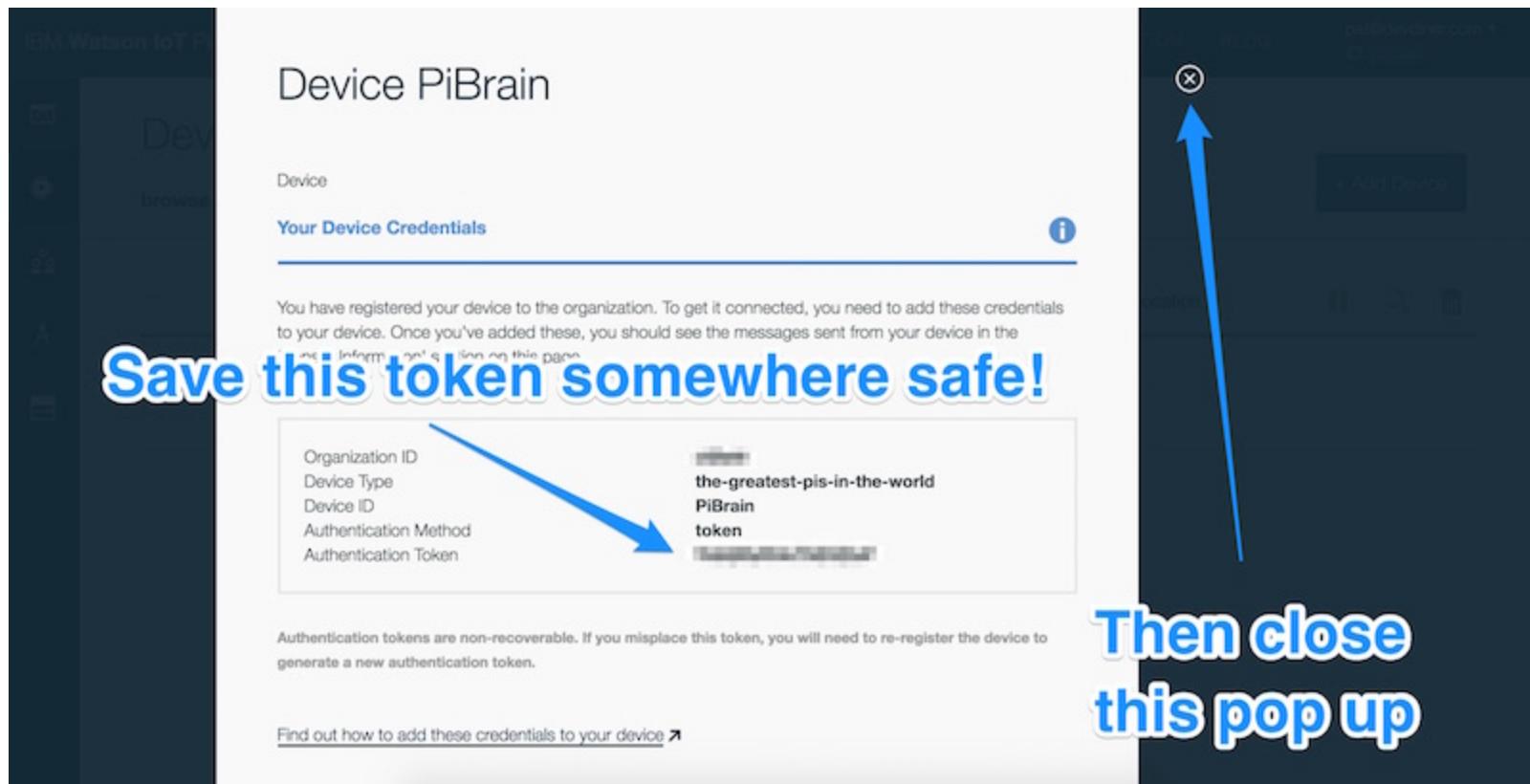
Device Type	the-greatest-pis-in-the-world
Device ID	PiBrain
Serial Number	-
Manufacturer	-
Model	Raspberry Pi 3 Model B
Class	-
Description	-
Firmware Version	-

If everything is correct, click "Add"

Back      Add

# Cloud Technologies

## Connecting Raspberry Pi to IBM Bluemix



## Fog Computing

# What is Fog Computing

- **Fog computing** is a term created by Cisco that refers to extending cloud **computing** to the edge of an enterprise's network. Also known as **Edge Computing** or Fogging

# Fog Computing

## Why the Edge?



### IoT in the Cloud

- Remote monitoring and control
- Merging remote from across multiple IoT devices
- Near infinite compute and storage to train machine and other advanced artificial intelligence tools

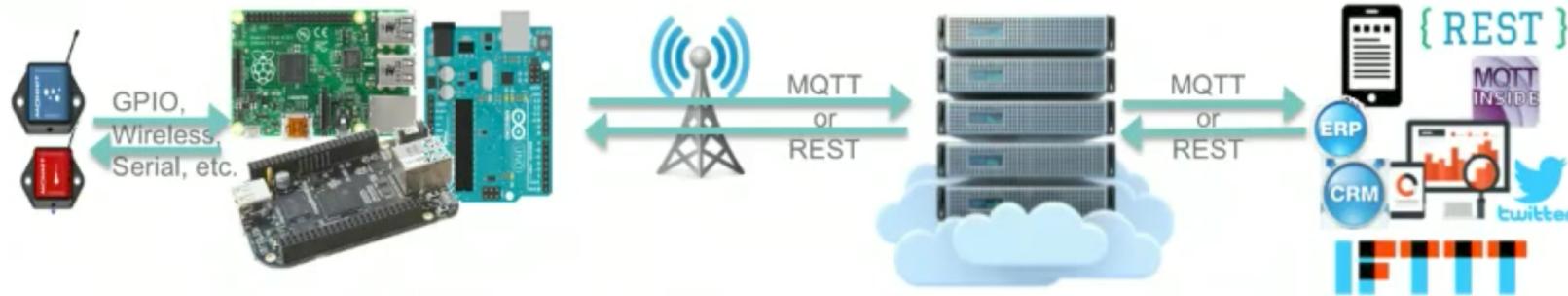
### IoT on the Edge

- Low latency tight control loops require near time-time response
- Public Internet inherently unpredictable
- Privacy of data and protection of IP

# Fog Computing

## Why the Edge?

### Fragmentation



Sensor Protocols	Controller	Network	Standards	Comm. protocol / Device Mgmt
<ul style="list-style-type: none"><li>• ZigBee</li><li>• Z-Wave</li><li>• CANBus</li><li>• MODBus</li><li>• Bluetooth/BLE</li><li>• DECT</li><li>• 6LoWPAN</li><li>• LoRA</li></ul>	<ul style="list-style-type: none"><li>• OSes</li><li>• Compilers</li><li>• Programming languages</li><li>• Processors</li></ul>	<ul style="list-style-type: none"><li>• Cellular</li><li>• Wi-Fi</li><li>• Gigabit Eth</li><li>• Proprietary network (ISM band)</li><li>• Safety spectrum (Non-ISM band)</li></ul>	<ul style="list-style-type: none"><li>• oneM2M</li><li>• Thread</li><li>• AllSeen</li><li>• OPC / OPCUA</li></ul>	<ul style="list-style-type: none"><li>• MQTT</li><li>• LWM2M</li><li>• CoAP</li><li>• AllSeen</li></ul>

# Fog Computing

## IoT Edge Duties

- Hardware and Field Abstraction
  - Sensor Connectivity
  - I/O Access
- Manage Network and Connectivity
  - Wireless, Firewall, VPN
  - Online/Offline mode
- Manage Applications
  - Remote Device Management
  - Remote Start/Stop, Install/Uninstall
- Manage IoT Connectivity
  - e.g, MQTT/CoAP Connections
  - Data Buffering and Retries
  - Provisioning, Credentials and Certificates

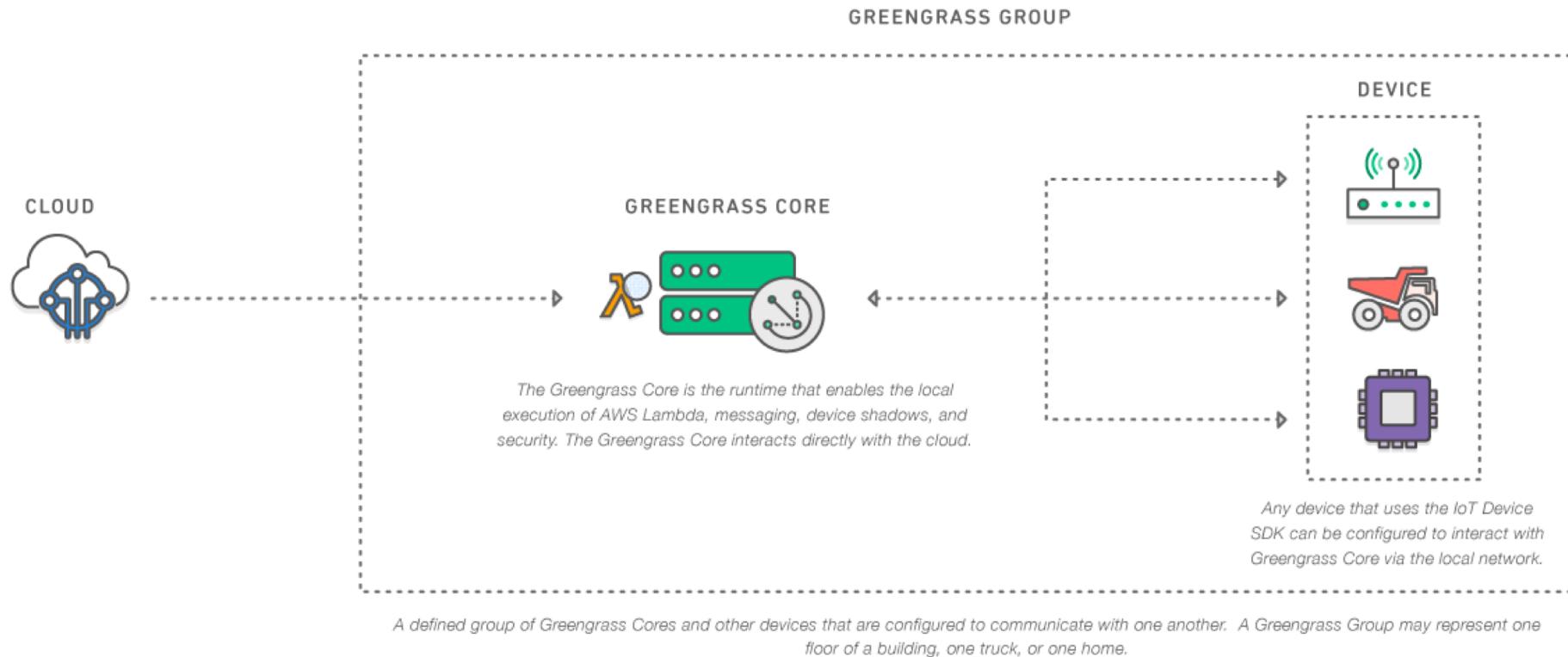
## What is Greengrass?

- An extension of AWS IoT, including Lambda, Broker, same security model, critical shadow functions
- Greengrass Core: The runtime responsible for Lambda execution, messaging, device shadows, security, and for interacting directly with the cloud
- Greengrass Core Group:
- System Requirements: Min Single-Core 1GHz, Min 128MB RAM, x86 and ARM, Linux(Ubuntu or Amazon)
  
- GreenGrass SDK, in C++ at this point

# AWS Greengrass

## Why Greengrass?

- Low latency processing
- Process your data on the Edge before sending to Cloud
- Run completely offline



# AWS Greengrass

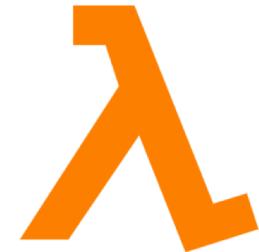
## Components

- Greengrass Core: Greengrass Core is the runtime that enables the local execution of AWS Lambda, messaging, device shadows and security. Core interacts directly with the Cloud.

## Components

### Local Lambda

- Lambdas are event-driven compute functions. With Greengrass, you can write Lambda functions in the cloud and deploy them locally
  - Command and control
  - Offline operation
  - Data filtering & aggregation
  - Iterative learning
- Greengrass runs Lambdas written in Python 2.7
- Invoke Lambda functions with messaging and shadow updates



## Components

### Shadow

- JSON documents that represent state of your devices and Lambdas
- Sync to the cloud or keep them local

### Messaging

- Local MQTT Pub/Sub messaging
- Define subscriptions between publishers and subscribers
- Apply MQTT topic filters

### Security

- Mutual auth, both locally and also with the cloud
- Certificate on your devices can be associated to SigV4 credentials in the cloud

## Use Case – Gas Turbine Filter Maintenance

- Each turbine consists of multiple devices and sensors
- Each turbine has a single Greengrass core
- Lambda functions have enough logic to perform edge machine learning
- Models and Lambda changes can be centrally updated
- Lambda filters data and uses its edge machine learning capabilities to determine when to request maintenance/replacement of filters
- Only required data is sent to AWS IoT

# Microsoft Azure IoT Edge

## Azure IoT Edge

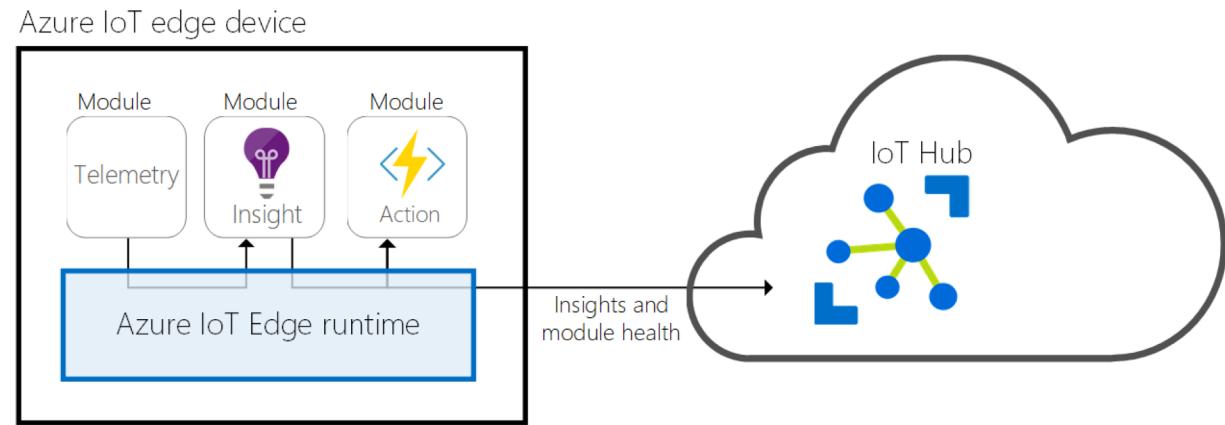
Github Page: <https://github.com/Azure/iot-edge>

# Microsoft Azure IoT Edge

## Azure IoT Edge

Edge Runtime provides fundamental services:

- Manage connectivity and security for devices otherwise isolated from Internet
- Multiplexing capabilities
- Store and forward(Offline)

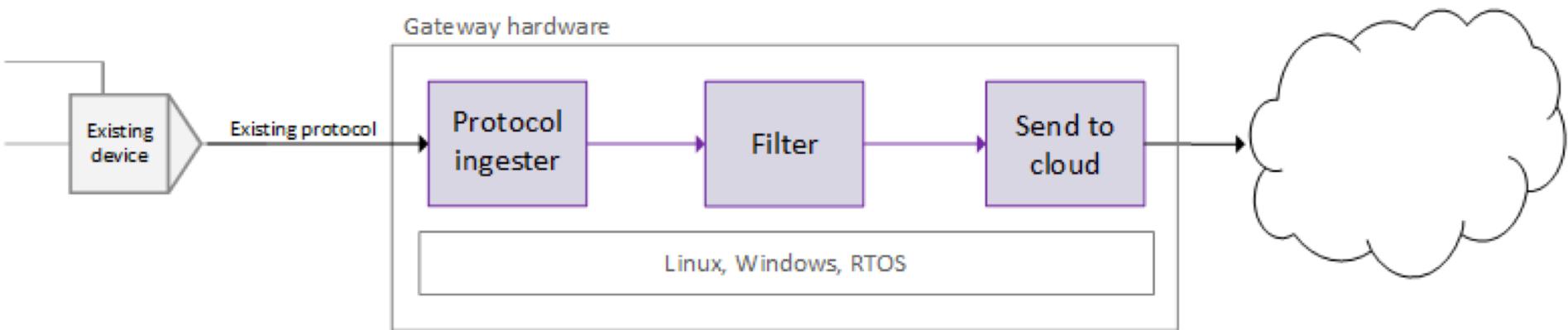


# Microsoft Azure IoT Edge

## Azure IoT Edge

Edge Runtime manages modules:

- Modules add capabilities to the runtime
- Each Module performs an action
- Chain of modules can be thought of as a data processing pipeline, solving an end to end scenario



Connect your existing devices which speak their existing protocols without making expensive changes to them.

Create a module pipeline which accomplishes your specific scenario.

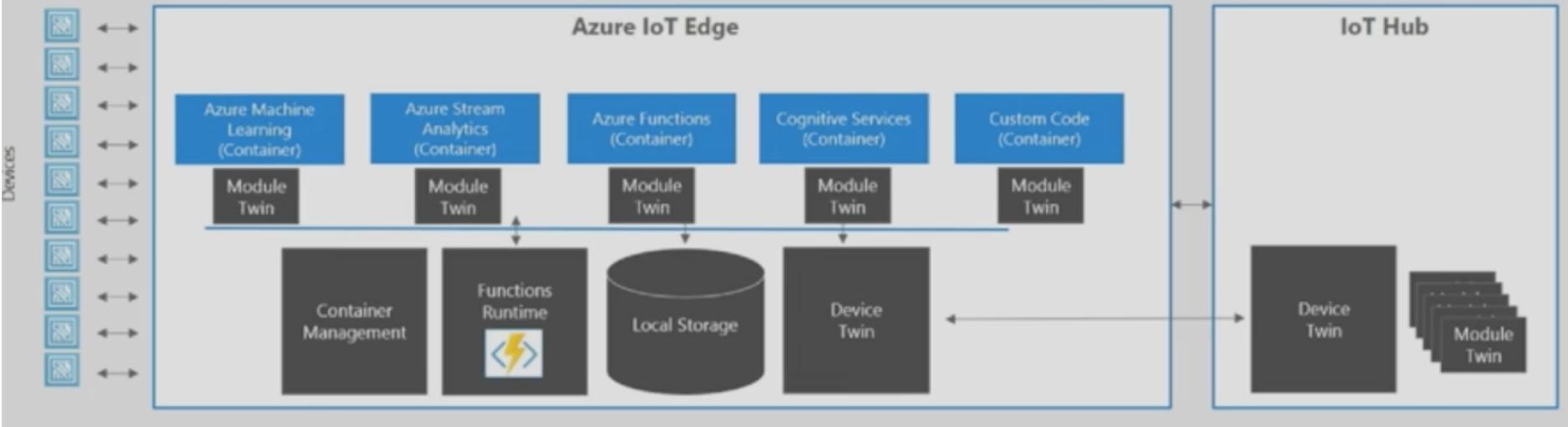
Only send necessary data to the cloud for further analytics, dashboarding, or storage.

- Modules are Docker containers.
- You can create custom modules in the language you choose.

# Microsoft Azure IoT Edge

## Azure IoT Edge

- Container based modules
  - Azure Functions
  - Azure Stream Analytics
  - Azure Machine Learning
  - Cognitive Services
- Offline / Synchronized Device Twins
  - Local Storage
  - Cloud Management & Deployment
  - High Availability / Fault Tolerance
  - Cloud Dev/Test Support



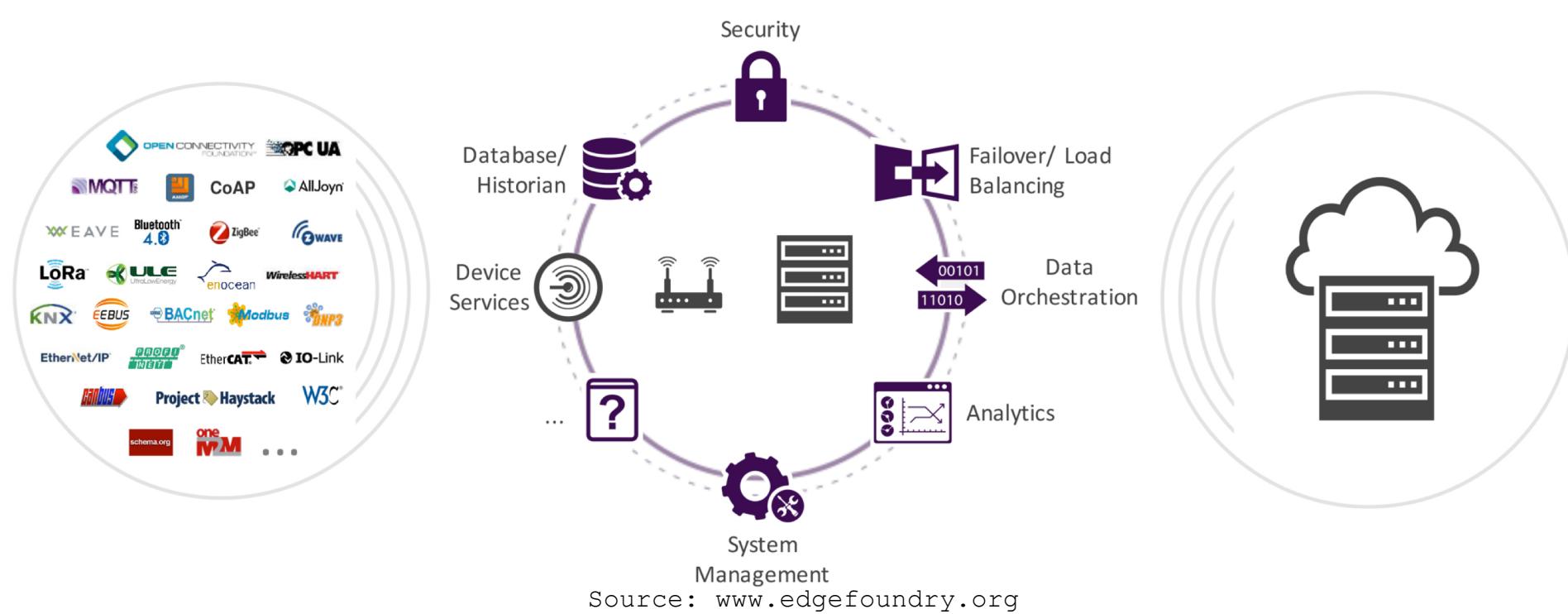
# EdgeX Foundry

## Overview

Any Combination  
of Standards

Interoperable Edge  
Applications

Choice of Backend  
Applications



- EdgeX Foundry™ is a vendor-neutral open source project building a common open framework for IoT edge computing.
- An interoperable framework hosted within a full hardware- and OS-agnostic reference software platform, to enable an ecosystem of plug-and-play components.

## Components

- MongoDB: EdgeX Foundry uses MongoDB as the persistence mechanism for sensor data as well as metadata about the devices/sensors that are connected.
- Java: The open sourced microservices of EdgeX Foundry are written in Java 8.
- Eclipse: EdgeX Foundry micro services were created in Eclipse using Java Maven projects.
- <https://www.edgexfoundry.org/>

# Eclipse Kura

## Open Source

