



# Arrow Connect Boot Camp #2

Tam Nguyen

Brandon Hall

Denver - December 2017

# Arrow Connect Platform

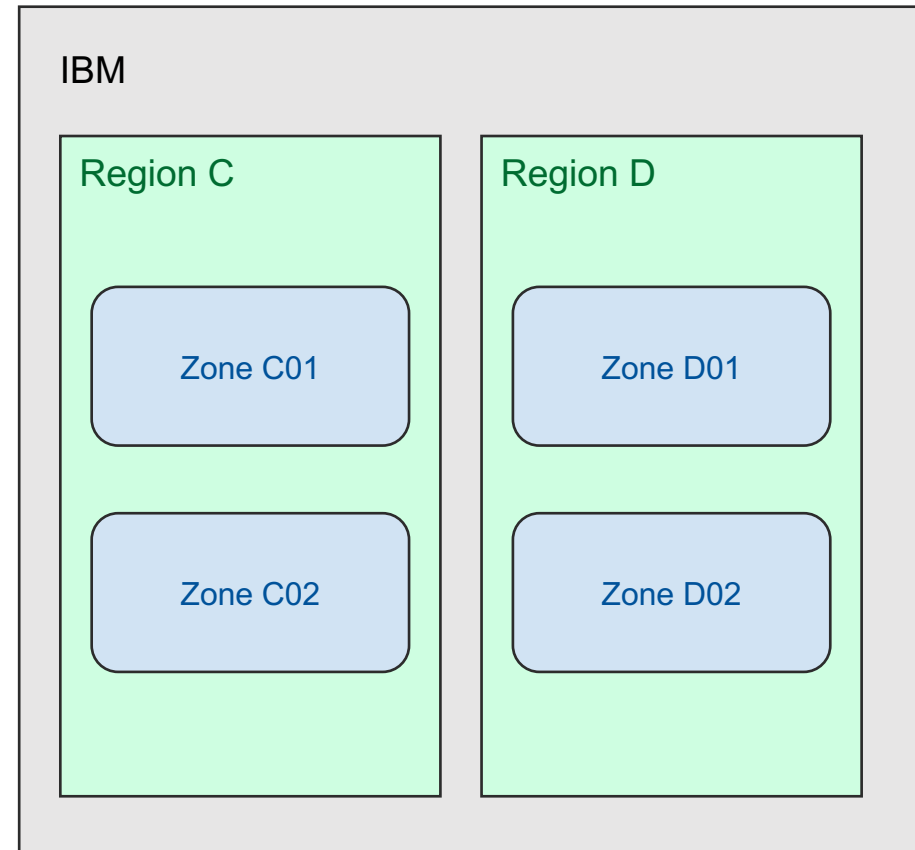
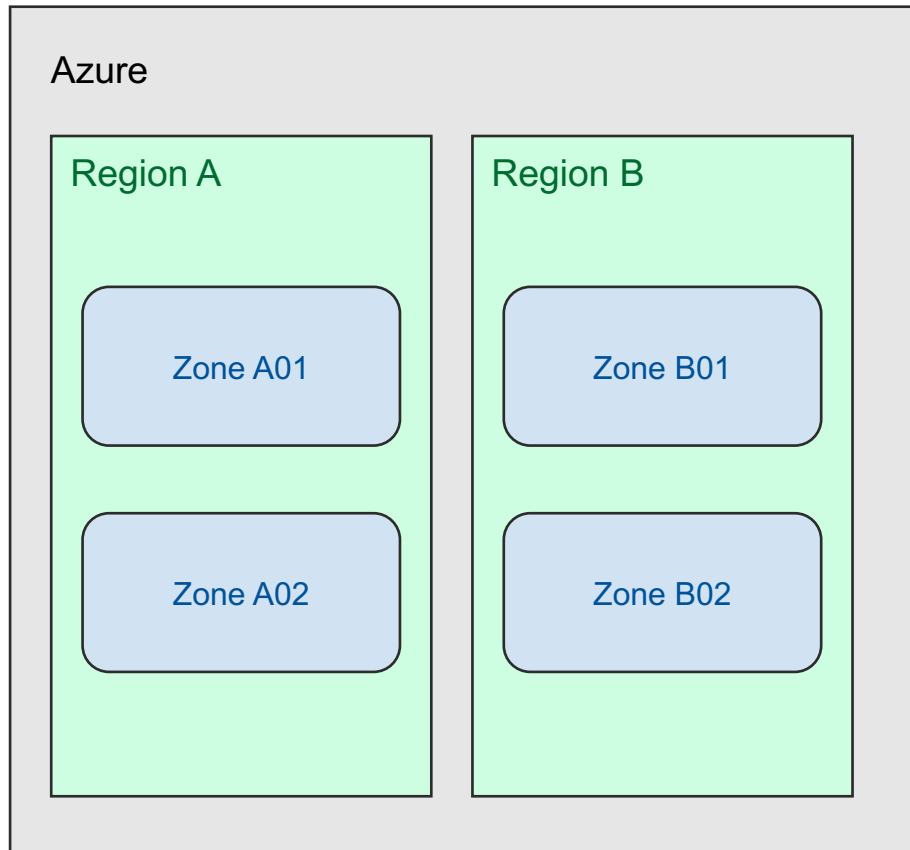
# Arrow Connect Platform

- Arrow Connect is a Platform-as-a-Service (PaaS)
- Arrow Connect runs on Arrow's private cloud that is hosted globally on many public cloud providers such as Microsoft Azure, IBM Bluemix, Amazon Web Services
- Software built on this platform inherits the following features
  - Multi-tenancy (SaaS) – **unlimited nested level**
  - Multiple security schemes (SSO, AD, custom, RBAC, etc.)
  - Encryption at rest and in transit
  - Data encapsulation
  - Distributed Services
  - Highly Available
  - Resilient

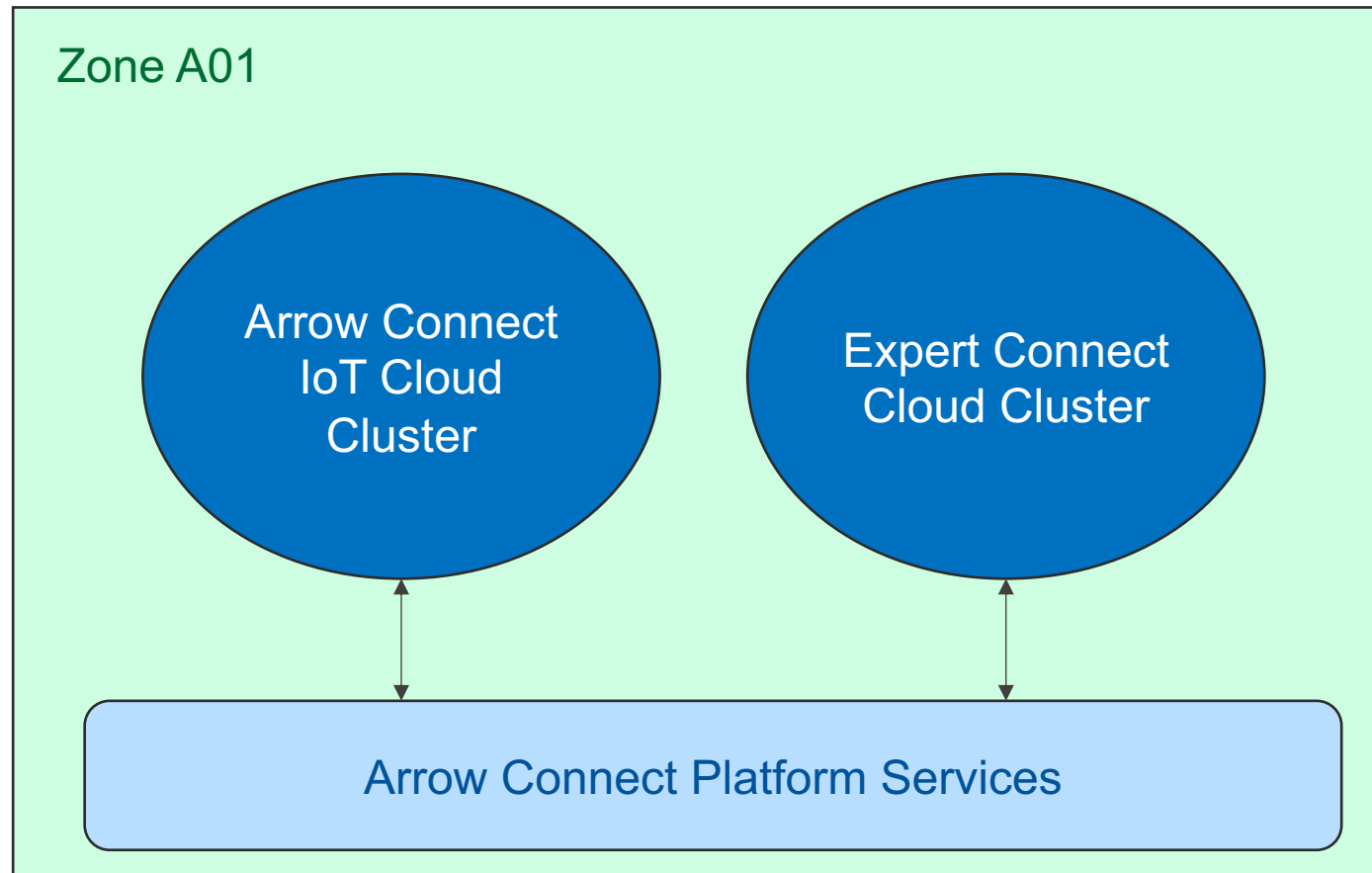
# Arrow Connect Platform Services

- Database (NoSQL, SQL)
- Cache
- Message Queue
- Message Hub
- Security
- Rule Engine
- Audit Log
- File Service
- Reporting

# Arrow Connect Infrastructure (1/2)



## Arrow Connect Infrastructure (2/2)



# Arrow Connect Management Portal (Beta)

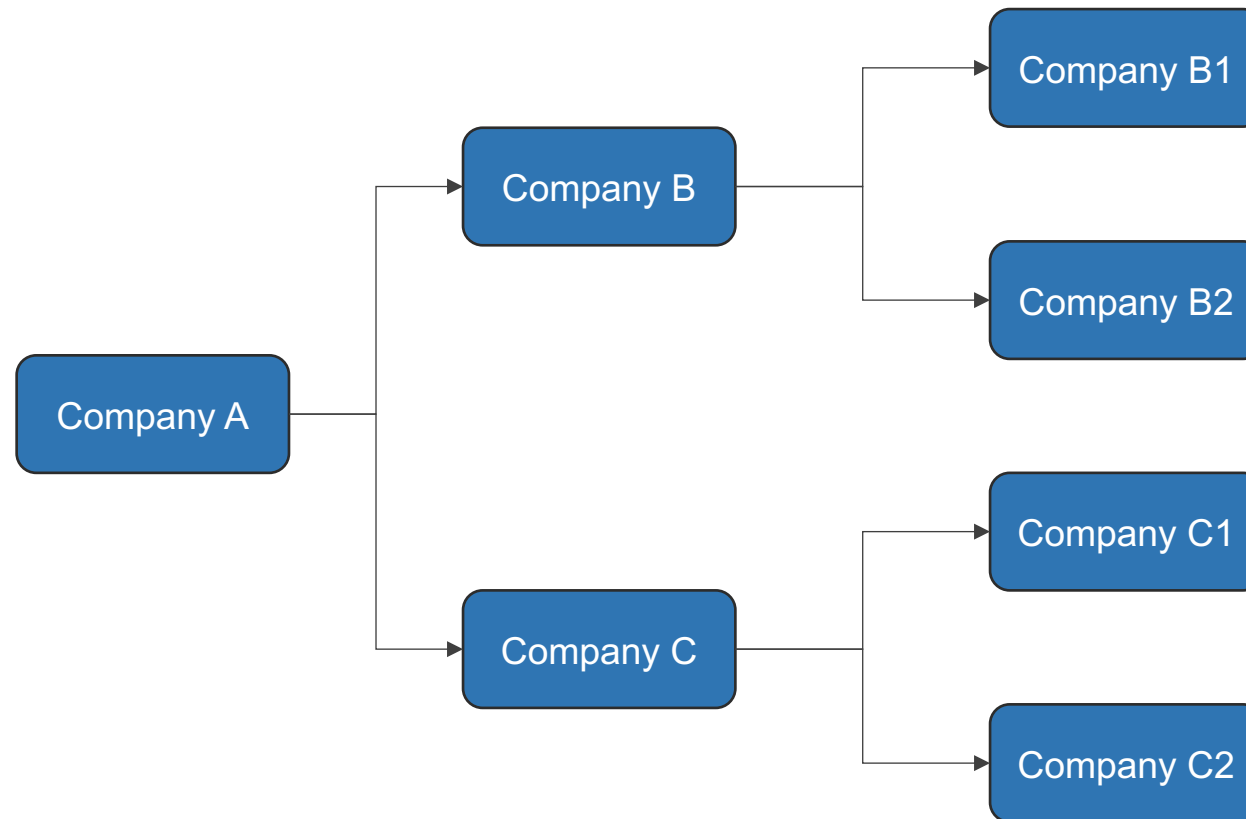
- URL – <https://acs.arrowconnect.io>
- Provides tool to manage the following
  - Tenants and hierarchy
  - Subscriptions
  - Application instances
  - Authentication
  - User
  - Roles
  - Access keys

# Arrow Connect Data Models

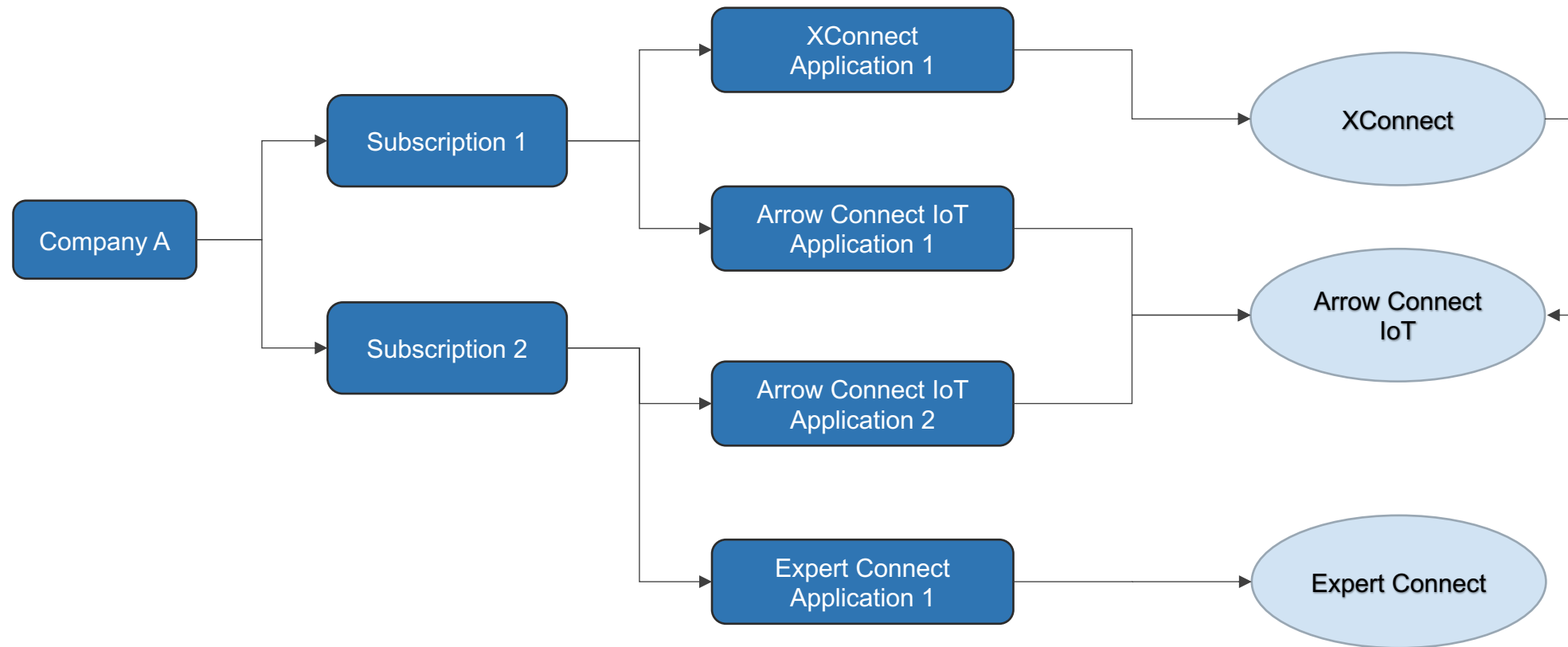
- Company (Tenant)
- Subscription
- Product
- Product Extension
- Application
- User
- Authentication
- Role
- Privilege
- Access Key



# Company (Tenant) Model



# Subscription / Application Models



# User Authentication

- Built-in Authentication
  - Default authentication method
  - Managed by Arrow Connect Platform
- Active Directory (AD)
- Single Sign On (SSO)

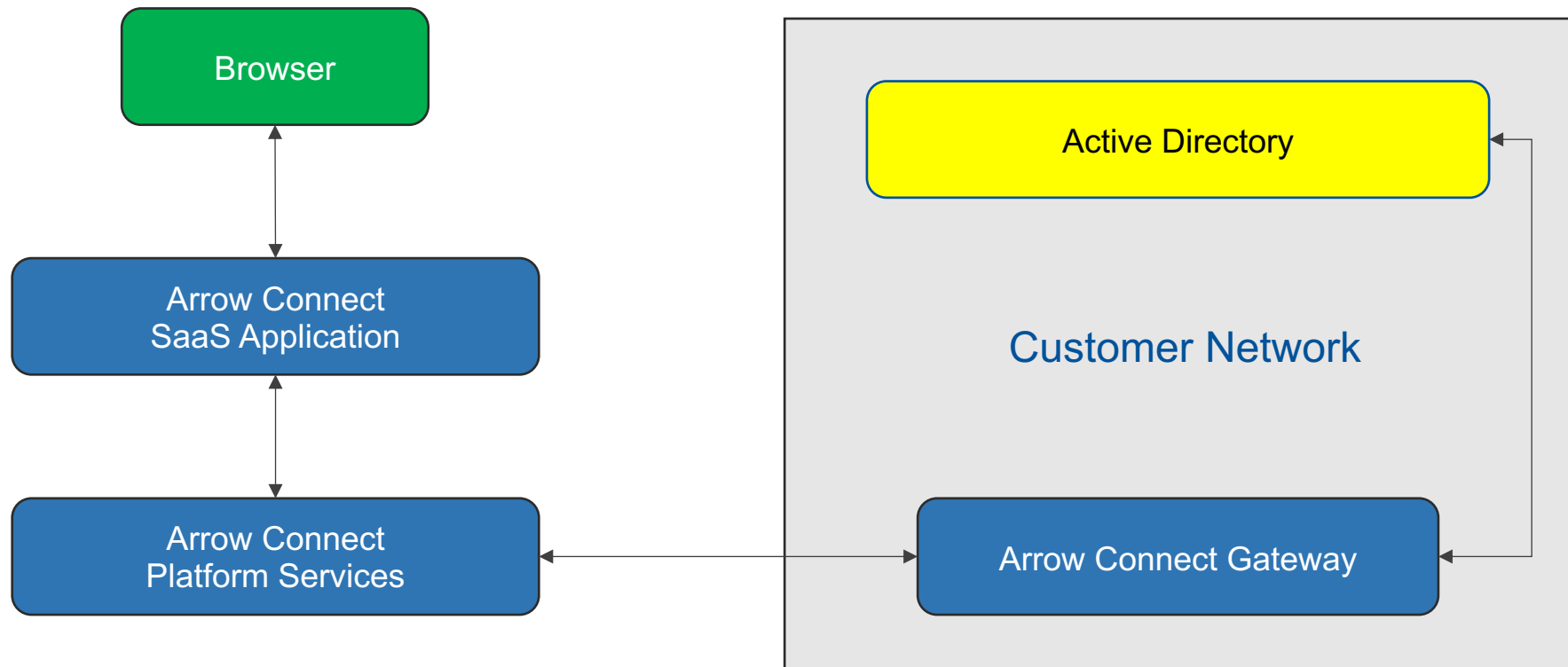
# Built-in Authentication

- Secured Internal Database
- Encrypted at rest
- Password hashing
- Configurable password policy
- Account lock-out policy

# Active Directory Authentication

- AD is usually not accessible outside customer network
- Requires an Arrow Connect Gateway installed within customer network
- Gateway requires 2 outbound ports. No inbound port is needed
  - 443 (HTTPS) – management channel
  - 5672 (AMQP) – data channel

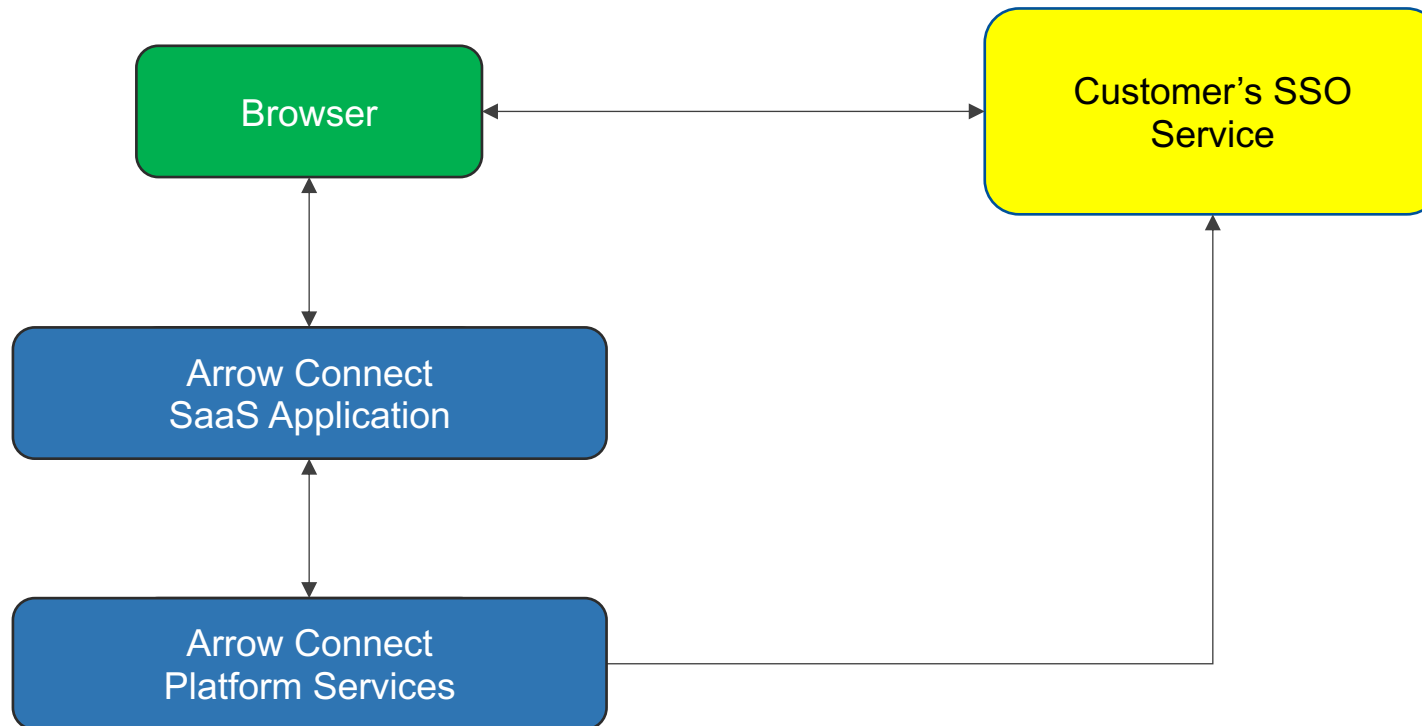
# Active Directory Authentication



# Single Sign On Authentication

- Popular method for Enterprise customer who already has many IT systems both internal and external that need a consolidated authentication mechanism
- Customer SSO service is usually accessible from the internet
- Platform can directly communicate with SSO service without a need of a gateway

# Single Sign On Authentication (SSO)

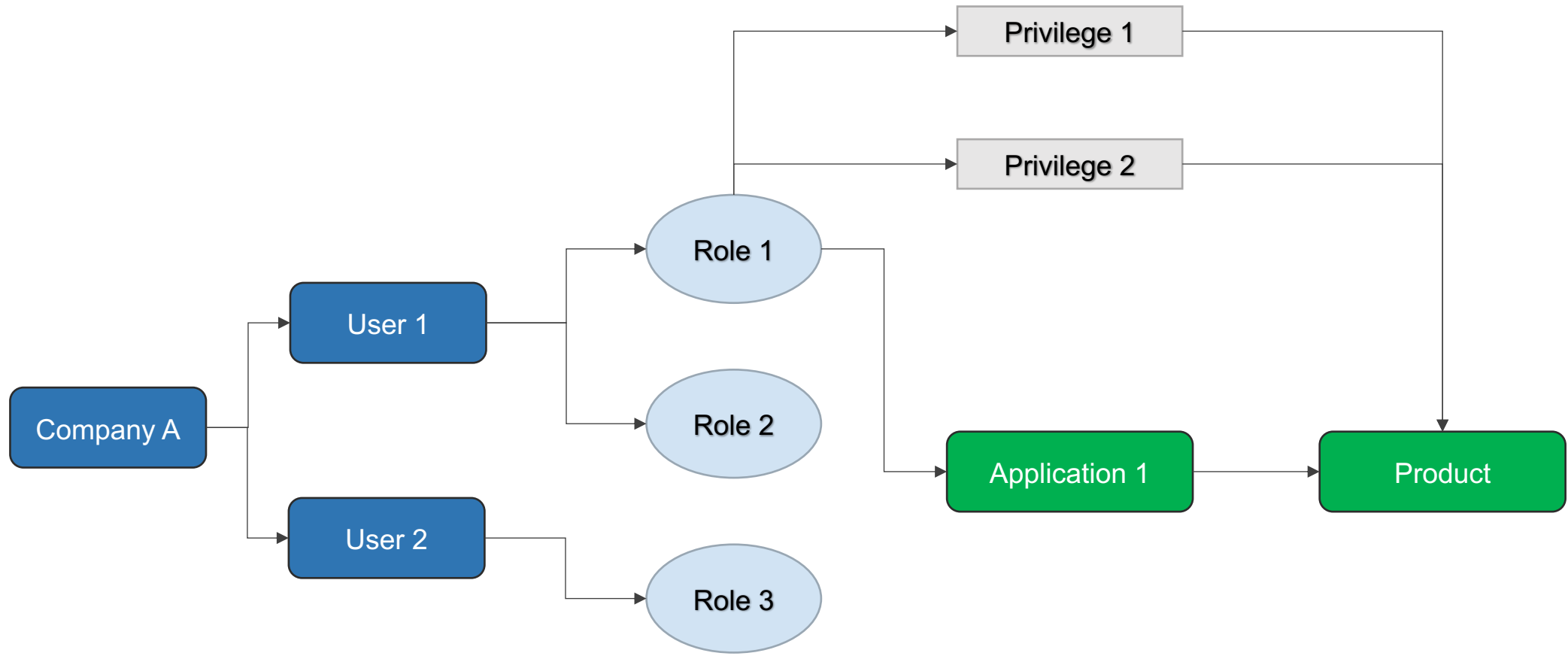




# User Authorization

- Software Product defines a list of privileges that control access to all operations within the software. Privileges are part of the software design and cannot be managed through web portal
- Operations are usually CRUD (CREATE, READ, UPDATE, DELETE)
- Role is defined within each application instance that contains a set of privileges
- Roles are managed by application instance admin
- Roles are assigned to users that give them access to the application
- Users can have many roles from different application instances. Web portal usually allows user to switch between application instances on the fly

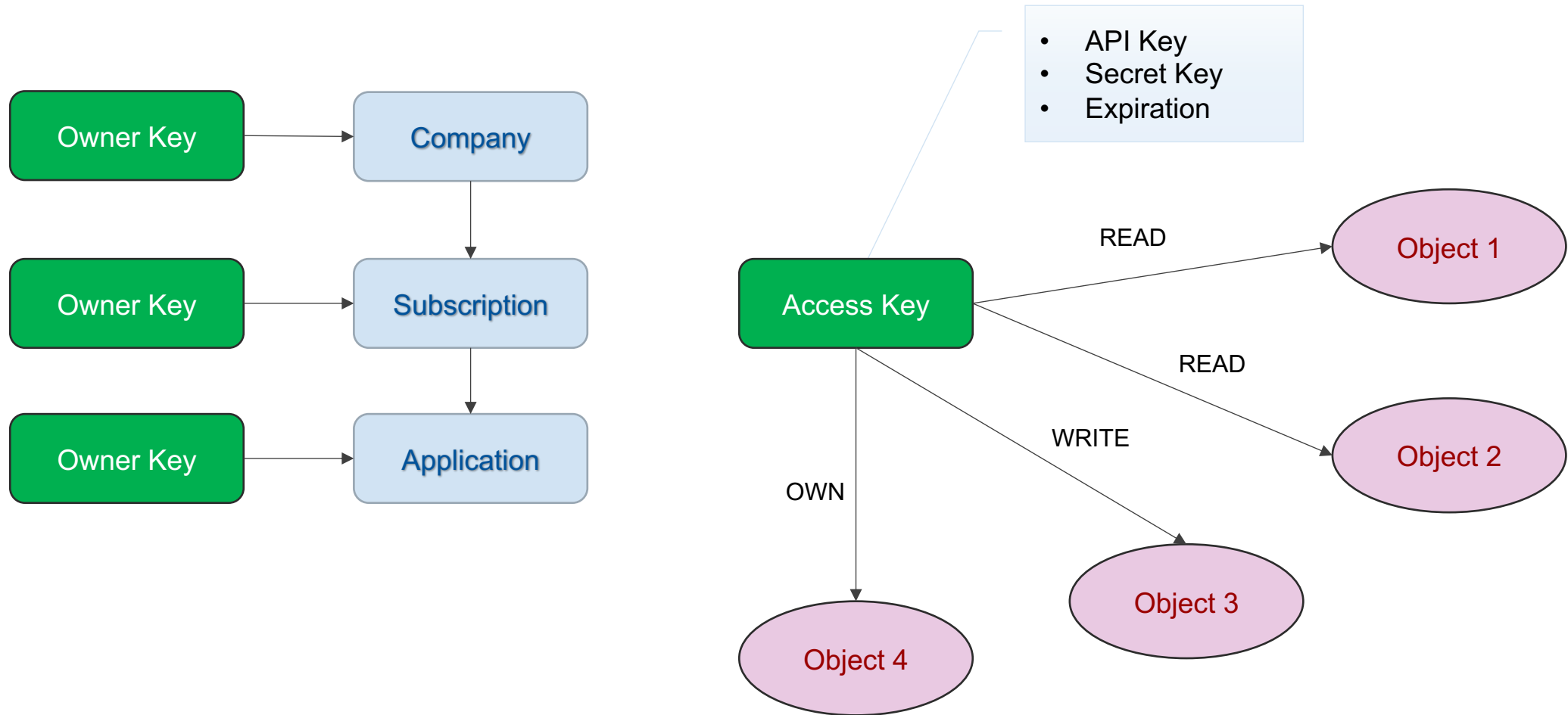
# RBAC Models



# API Authentication and Authorization

- Application access (API integration) is controlled by Access Keys
- Access Key consists of
  - API Key – used as authentication
  - Secret Key – symmetric key, used for digital signature and encryption
  - Expiration Date/Time
  - List of access to resources in the platform
    - Level – READ, WRITE, OWNER (reserved)
    - Platform Resource Identifier (PRI)
- API Request Signing using Access Key is required for production deployment
- Access Keys can be managed in the web portal

# Application Integration – Access Key Model



# API and SDK

- REST API is available for integration with Arrow Connect Platform
- Production Endpoint – <https://acs-api.arrowconnect.io>
- Swagger UI - <https://acs-api.arrowconnect.io/swagger-ui.html>
- Only Java SDK is available at this time. More to be added in the future
- **Note: API is still in ALPHA. Only a small set of API methods are available**

# Arrow Connect Lab

- Web Portal – <https://acs.arrowconnect.io>
- Credentials – hand out
- Browse
  - Tenant
  - Subscription
  - Application
  - User
  - Role
  - Access Key
- Create
  - Subscription
  - User
  - Role
- Configure built-in authentication policy

# Arrow Connect IoT

# Arrow Connect IoT

- Complete IoT Platform to enable customers to build their own end-to-end IoT solutions
- Flexible IoT solutions – OEM, Enterprise, SaaS, etc.
- Different verticals – Smart Building, Smart Lighting, Smart City, Automotive, etc.
- Wide support of OEM hardware
- Wide support of IoT Cloud services
- Divided into 3 main areas
  - Arrow Connect IoT Cloud Platform
  - Arrow Connect IoT Gateway
  - API and SDKs



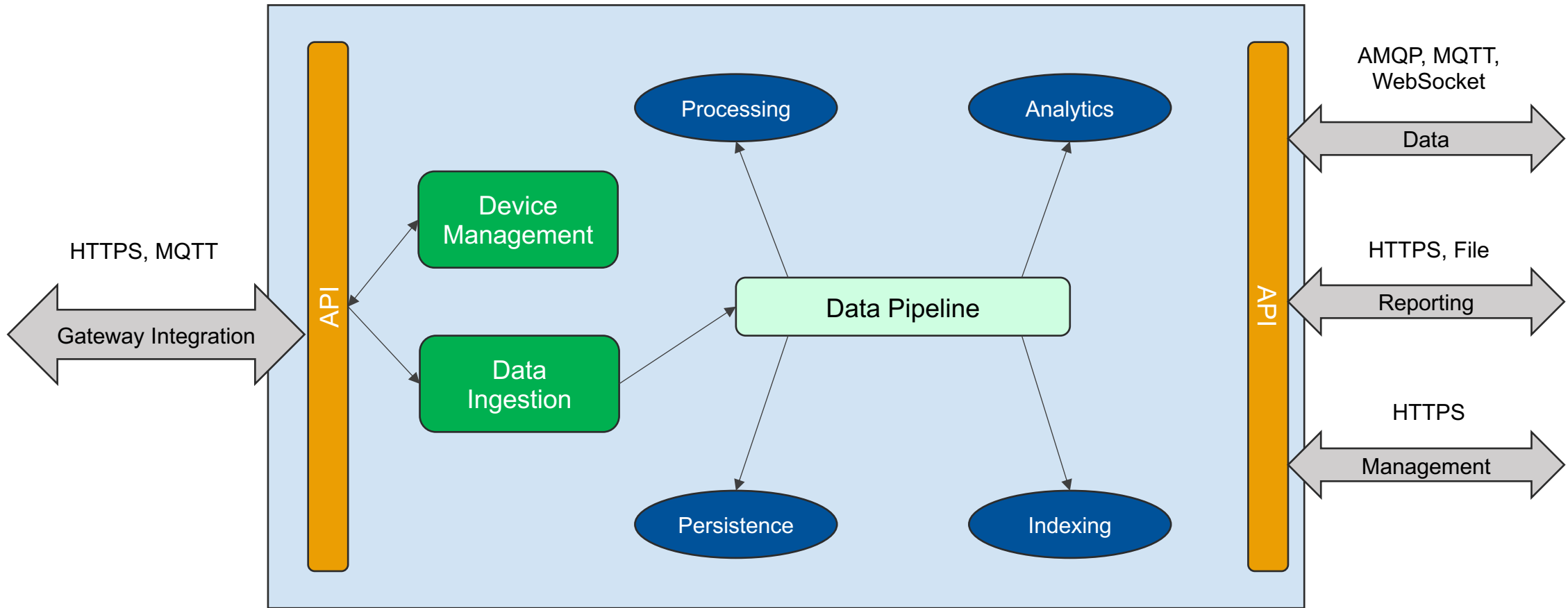
# IoT Cloud Overview (1/2)

- Arrow Connect IoT Cloud is a Software-as-a-Service (SaaS)
- Built on top of Arrow Connect PaaS
- Provides **complete** IoT cloud platform to build IoT solutions
  - Device Management
  - Data Ingestion
  - Software/Firmware Management
  - Data Analytics
  - Stream Processing / Rule Engine
  - API Integration
  - SDKs in many programming languages
  - Management Portal
  - Dynamic Dashboards
  - Reporting

# IoT Cloud Overview (2/2)

- Flexible deployment options
  - Cloud-based Arrow solution (SaaS)
  - On-premise Arrow solution
  - Arrow Connect + IBM Watson IoT
  - Arrow Connect + MS Azure IoT
  - Arrow Connect + AWS IoT
- Product Extensions
  - Add-on third-party products and services
- Out-of-the-box Integration solutions
- Custom Application Development
  - Customer's Intellectual Property (IP)
  - Various hosting options

# High Level Component Diagram



# Data Models

- Gateway
- Device
- Asset Type
- Device State
- Device Action
- Node (Group)
- Node Type
- Telemetry
- Telemetry Definition
- Software Release
- Software Release Schedule
- Test Procedure
- Test Result

# Device

- Device is the main entity in an Arrow Connect IoT deployment
- Device collects telemetry from one or more sensors
- Device has no direct connectivity to the cloud. Cloud communication must be done through a Gateway, although both of them could be physically on the same piece of hardware
- Device could be either actively sending telemetry to Gateway, or Gateway can poll data from Device periodically, depending the technology and type of deployment
- Device is managed by Gateway software

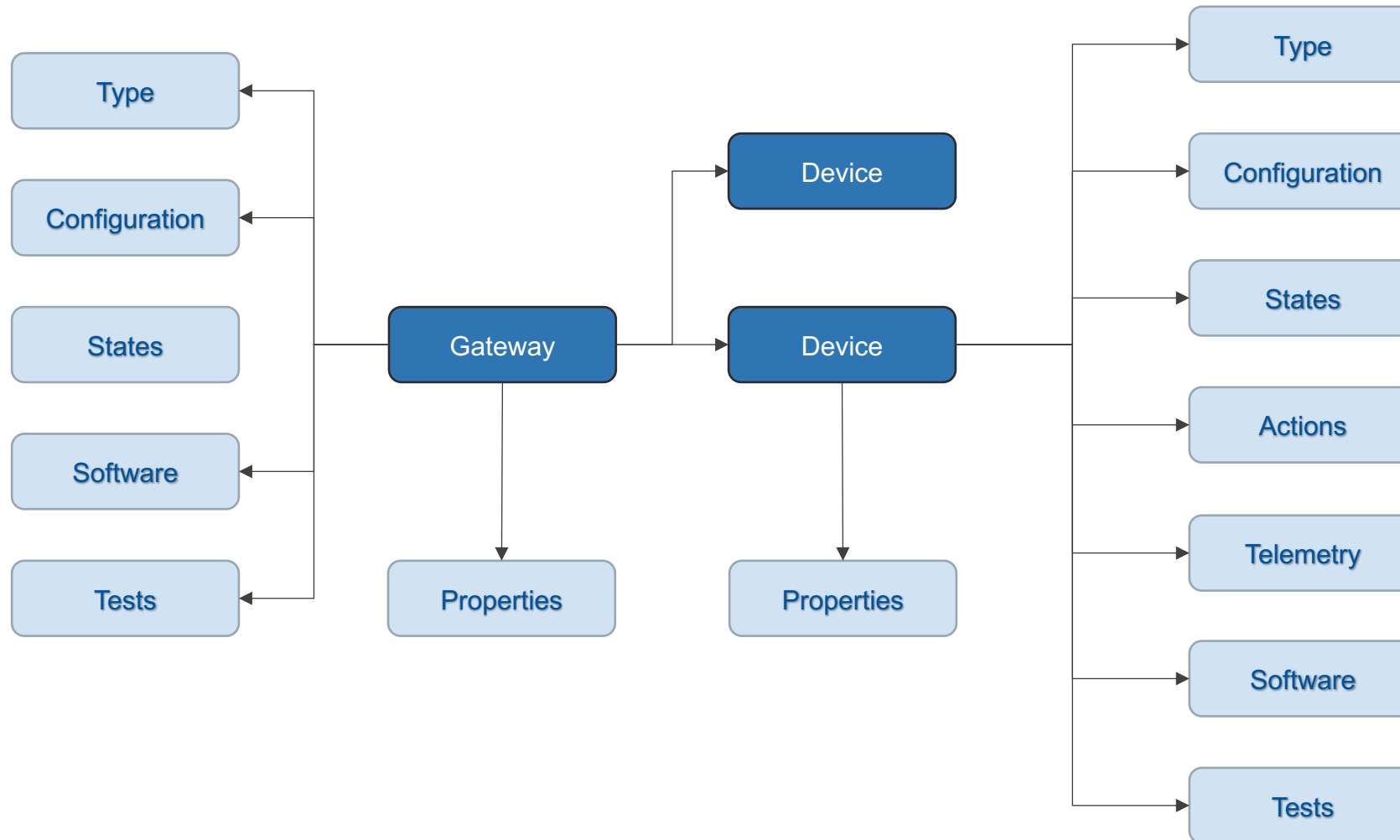
# Gateway

- Gateway serves as a device hub.
- Gateway communicates with the cloud via API
- Gateway manages lifecycle of devices
  - Registration
  - De-registration
  - Send device telemetry to cloud
  - Send cloud command to device
  - Sync device configuration to cloud
  - Restore device configuration from cloud
  - Firmware update

# Asset Type

- Asset is the generic name for either a gateway or a device
- Each asset in the system is linked to an Asset Type that defines its characteristic
- Asset Type name is hard-coded (or detected) by the software running on the physical asset
- Important element of firmware management process
- Device asset type provides the following definition
  - Telemetry
  - State
  - Device action rules (discussed later)
- **Note: Web portal has a button to manually sync telemetry definition when viewing device details. Upon clicking this button, the system will analyze telemetry history of this device and sync the definition to the asset type. This only helps during development cycle, and not production where asset types must be already fully defined**

# Gateway and Device Models





# Telemetry (1/3)

- Payload is a JSON document of key-value pairs

```
{  
    "key1"      : "value1",  
    "key2"      : "value2"  
}
```

- Key format: `<data-type>|<telemetry-name>`
- Value format: string presentation of the value
- Telemetry name must be unique in a payload
- Telemetry name cannot contain special characters. Valid characters are
  - [a-z] [A-Z]
  - [0-9]
  - underscore and dash

## Telemetry (2/3)

Data Type	Description
s	String
i	Integer
f	Floating point
b	Boolean
d	Date
dt	Date and Time
i2	2-tuple of integer values delimited by   character
i3	3-tuple of integer values delimited by   character
f2	2-tuple of float values delimited by   character
f3	3-tuple of float values delimited by   character
bi	Binary value (base64-encoded string)

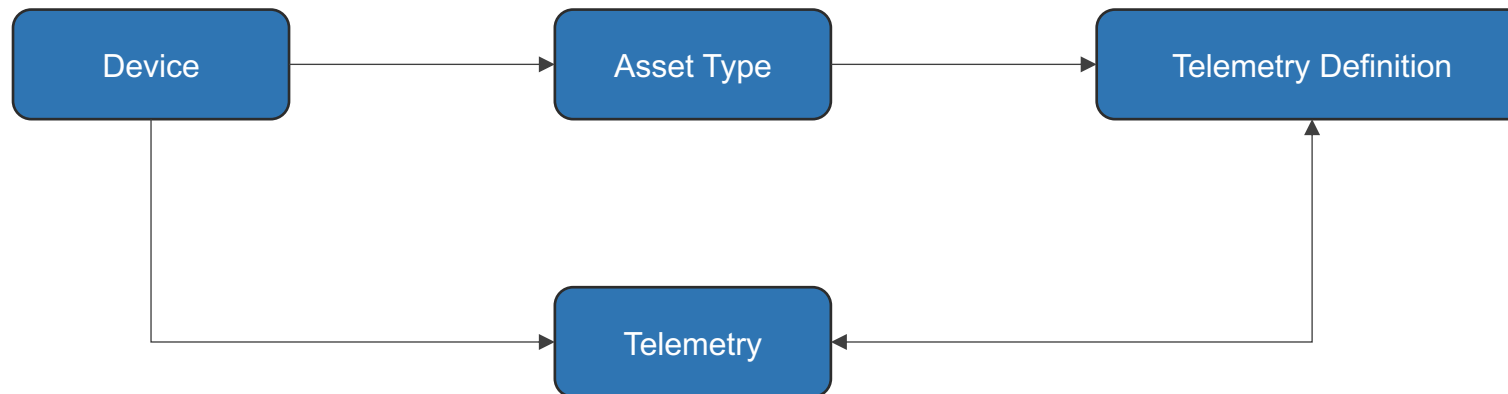
# Telemetry (3/3)

- Example payload

```
{  
  "s|color"      : "red",  
  "i|counter"    : "1246",  
  "f|temperature": "60.5",  
  "b|valid"      : "true",  
  "d|expiredDate": "09/01/2016",  
  "i2|resolution": "1920|1080",  
  "f3|xyz"       : "23.3|24.2|83.1"  
}
```

# Telemetry Definition

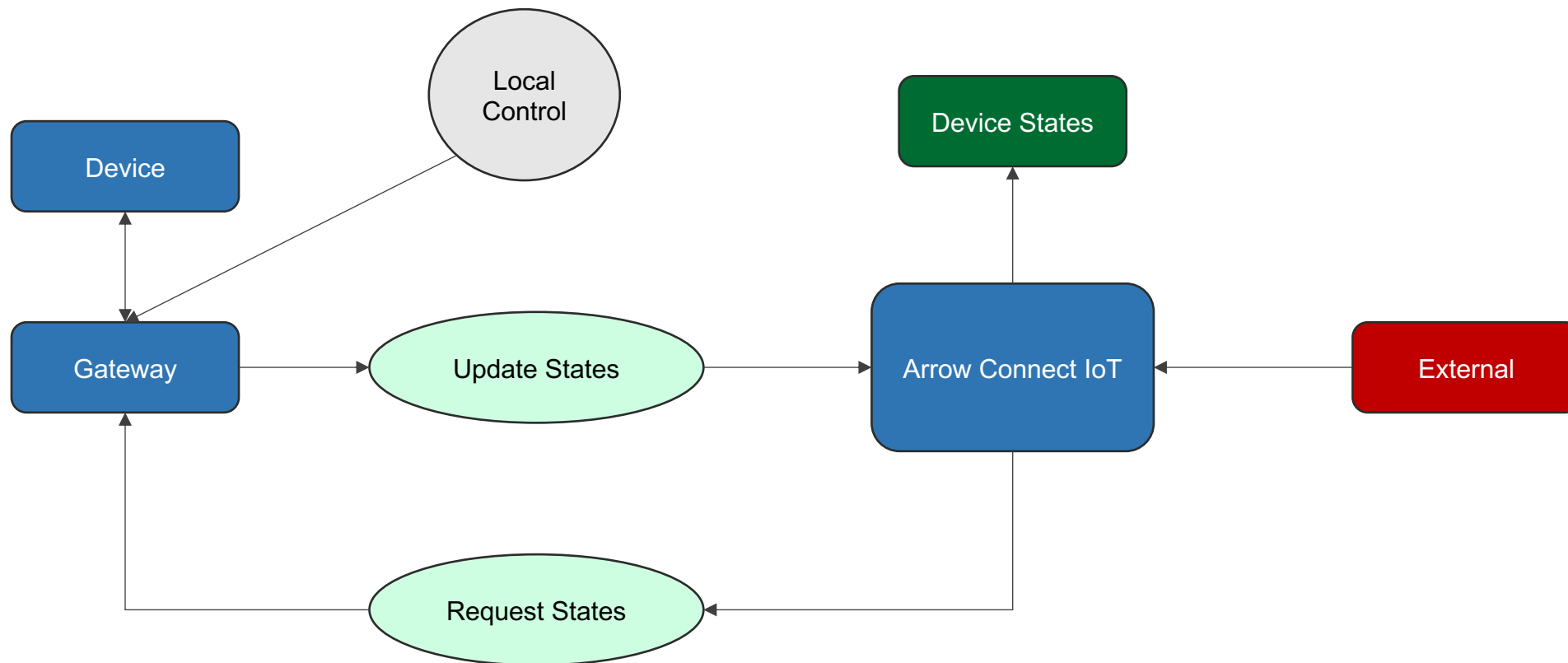
- Name
- Description
- Data type
- Unit of measure
- Additional metadata



# Device State

- States are attributes of the devices that can be controlled remotely. For example, on/off status of an LED, speed of a motor, temperature setting of a thermostat, etc.
- Do not use state for read-only sensor data such as humidity, ambient light, pressure, etc. Define them as telemetry instead
- Settings can also be defined as states
- State definition should be defined in Asset Type similar to telemetry definition
- Platform manages synchronization between the actual states reported by the gateway to the cloud, and the requests of state change from the cloud sent to the gateway and device

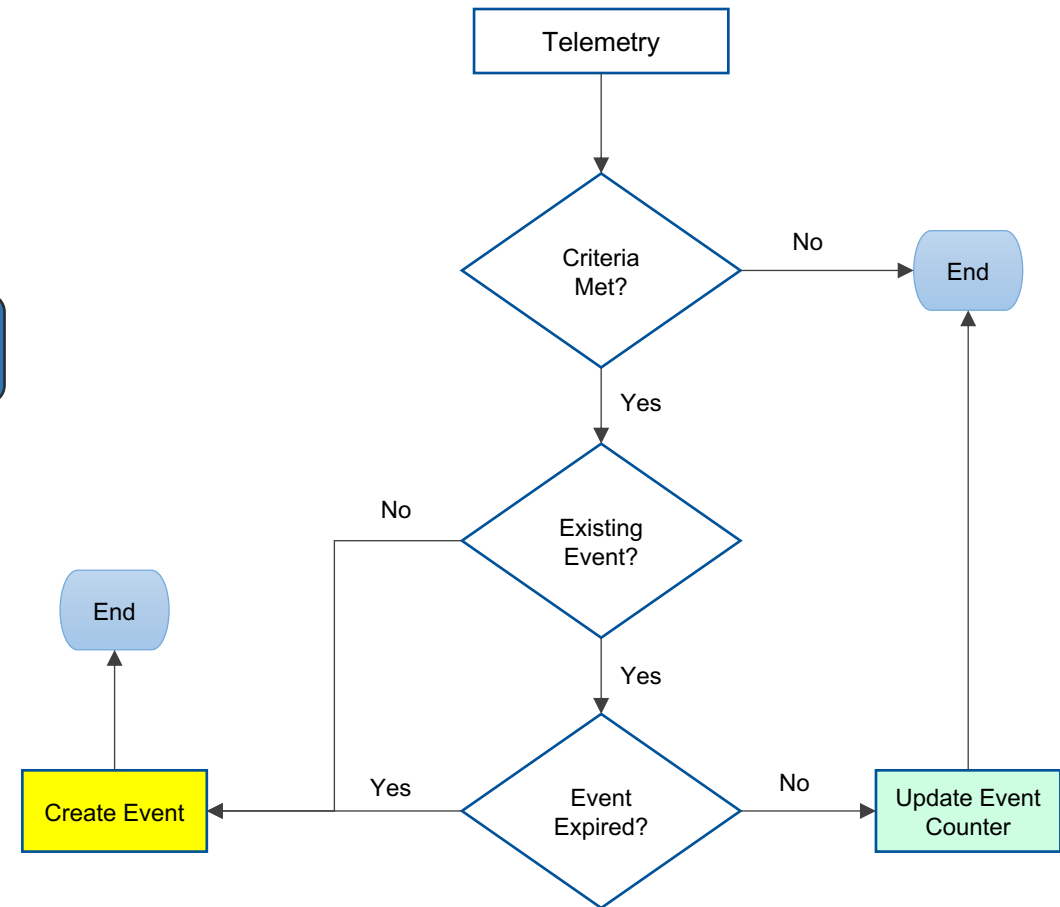
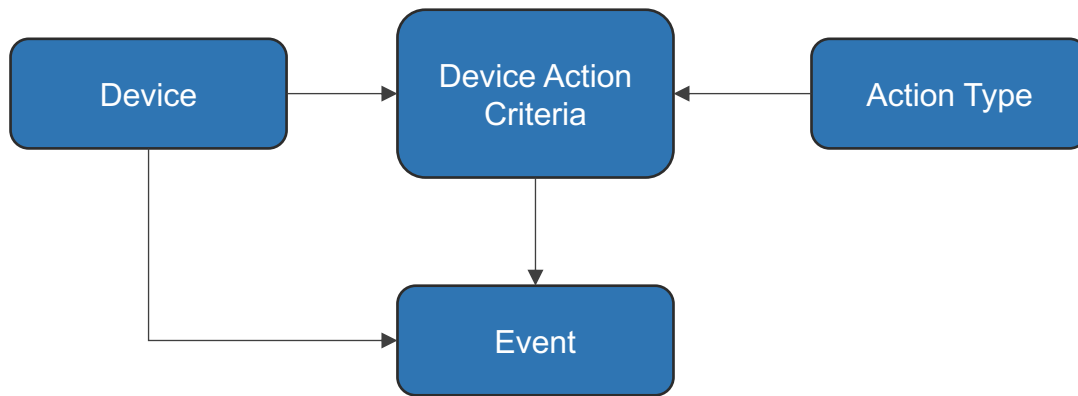
# Device State Models



# Device Action

- The platform has a built-in stream processing engine that analyzes live telemetry stream coming into the cloud. Rules can be set up to monitor the stream and trigger actions when defined criteria are met
- There are 3 action types available at this time
  - Send an email
  - Send a command to the device
  - Integrate with an external REST API via POST
- API integration is powerful since it's generic and can be configured to integrate with any cloud service including our own Arrow Connect IoT API. Some examples of API integration are
  - Send SMS, mobile push notifications, tweets, etc.
  - Create alarms, support tickets, etc.

# Device Action Models

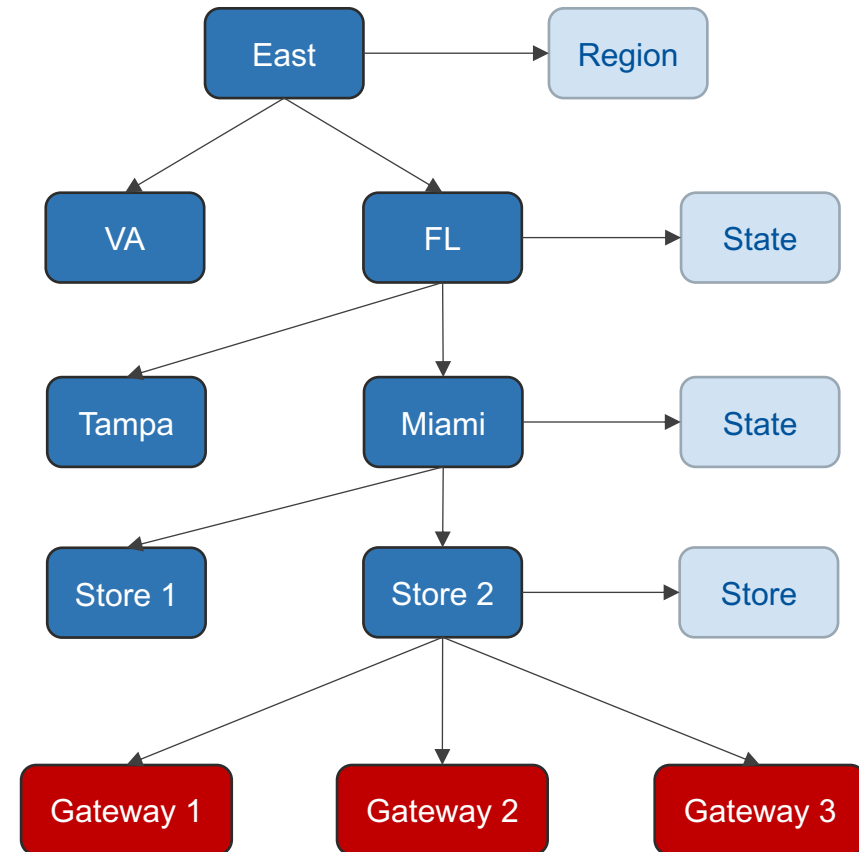
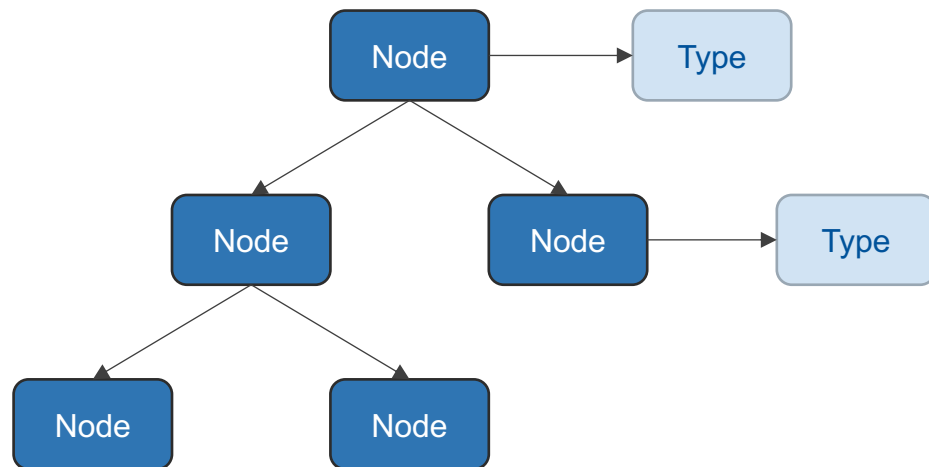




# Gateway and Device Hierarchy

- Gateways and devices can be organized into a tree hierarchy
- There's no limit in the node levels and the number of child nodes in each level
- Node and group are used interchangeably
- Portal does not support any action at node level at this time; however, custom development project can build its own application logic on top of the hierarchy. Future enhancements will support generic actions at the node level such as applying rules, etc.

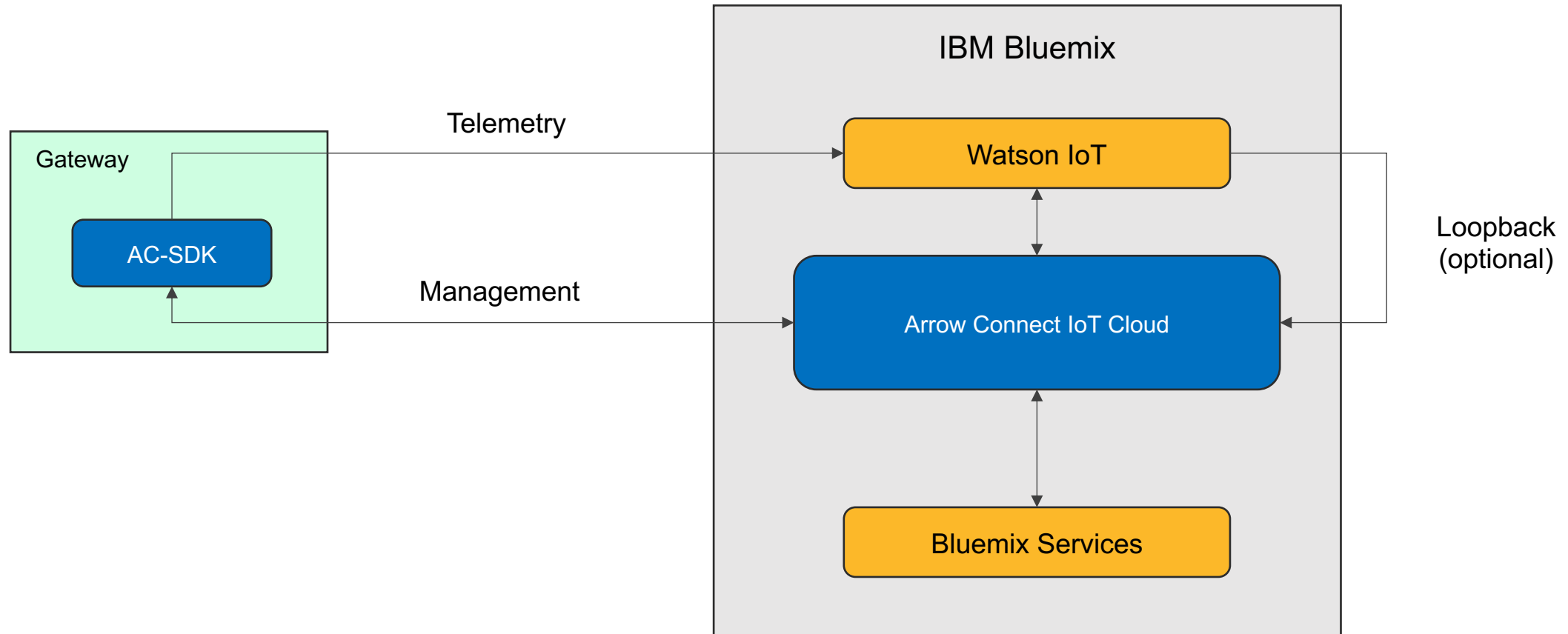
# Node (Group) Model



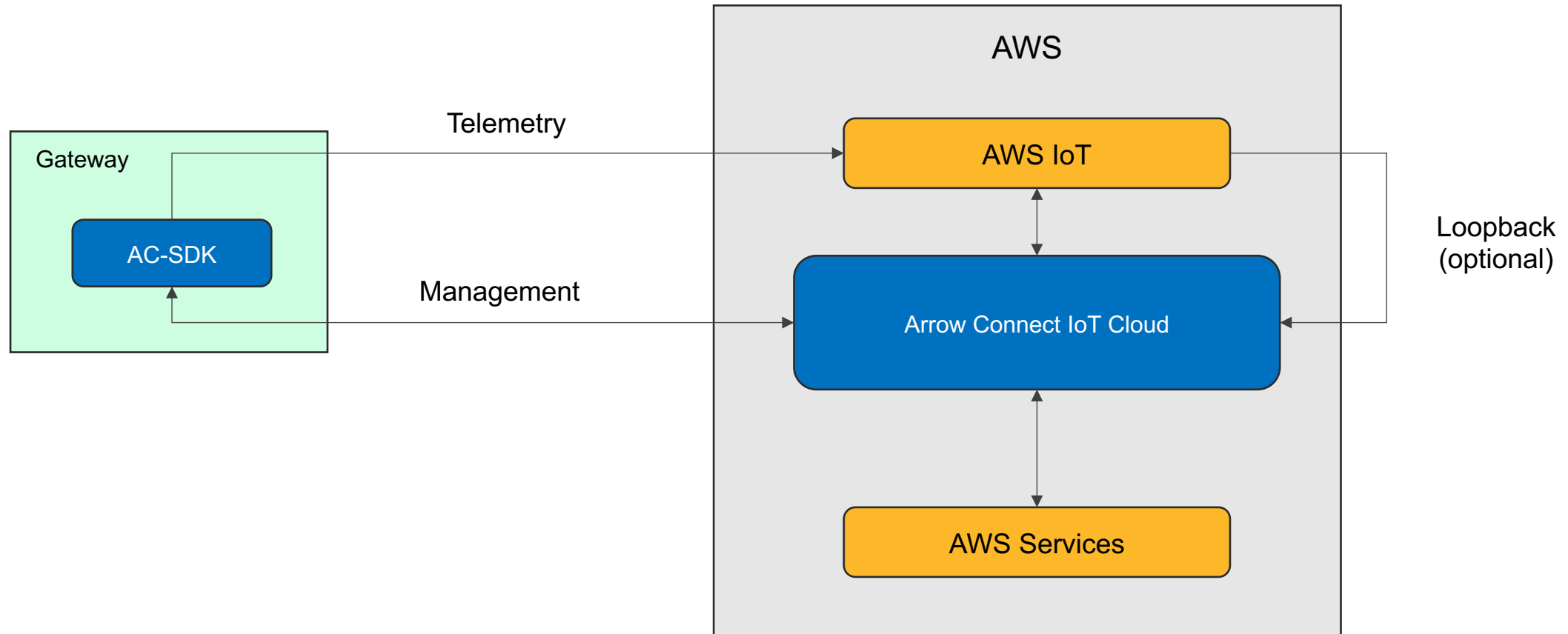
# IoT Platform Provider Integration

- Arrow Connect IoT implements direct back-end integration with major IoT platform providers, currently IBM, Azure, and AWS. This integration in the cloud allows Arrow Connect to provision gateways, devices and security access on the platform provider back-end programmatically
- Platform provider SDKs are bundled with Arrow Connect SDKs to provide seamless integration from the gateway directly into platform provider without any additional development
- Configuration is done through web portal or Admin API. If leveraging our SDK, gateways will automatically download credentials and connection information from Arrow Connect cloud during gateway check-in, no manual configuration is needed at the gateway level unless the customer desires gateway pre-configuration
- Gateway pushes telemetry directly into provider platform, not through Arrow Connect
- Customer has option to turn on telemetry loopback functionality which automatically subscribe to provider telemetry stream and clone the data back into Arrow Connect in real time.

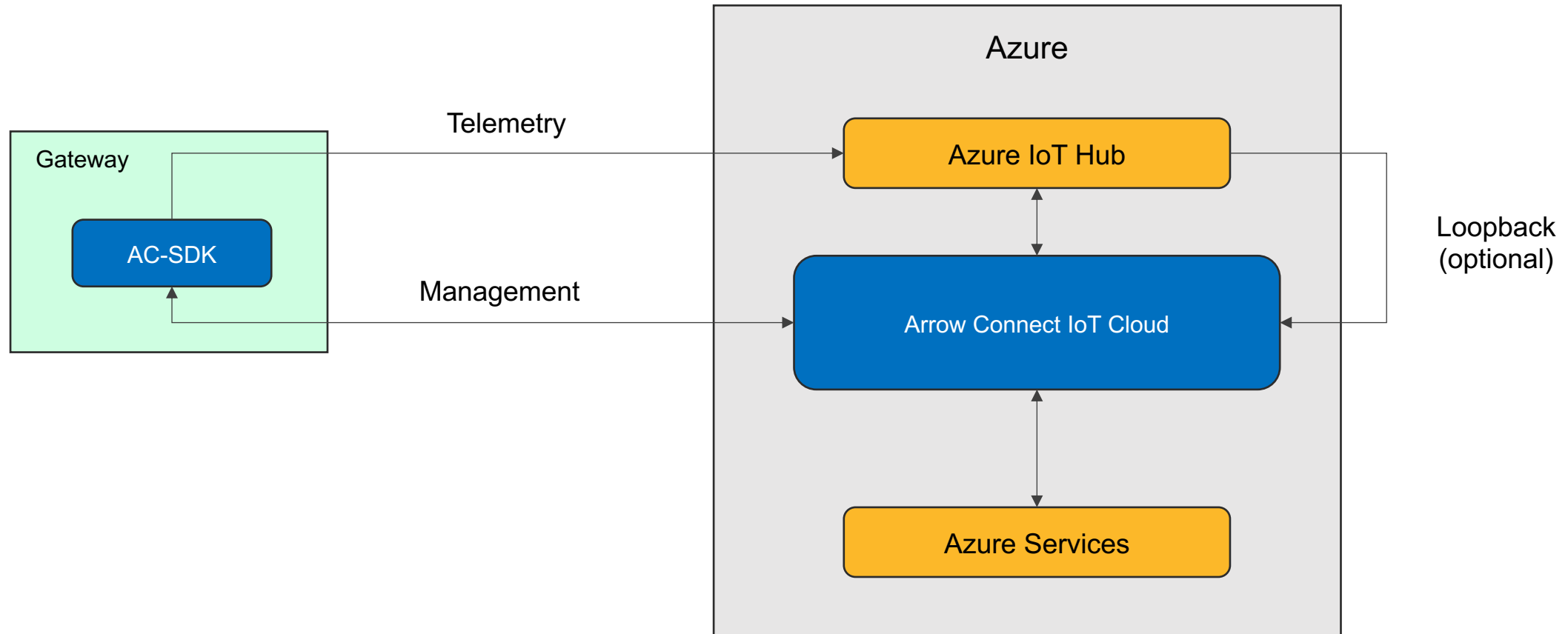
# IBM Watson IoT Integration



# AWS IoT Integration



# Azure IoT Integration



# IoT Platform Provider - Lab

- Create an IBM developer account  
<https://www.ibm.com/account/us-en/signup/register.html>
- Create an IBM Cloud account  
<https://console.bluemix.net/docs/admin/adminpublic.html#signing-up-for-ibm-cloud>
- Add Watson IoT service  
<https://www.ibm.com/internet-of-things/spotlight/watson-iot-platform/trial>
- Create new application instance in Arrow Connect Platform Portal
- Log on to new application instance in Arrow Connect IoT Portal and associate it with the IBM account
- Create new profile in mobile app and start streaming data into IBM Watson IoT

# Firmware Management

- Centralized Firmware repository
- Track both hardware and software versions of gateway and device
- Track historical upgrades
- Track update path and compatibilities
- Perform upgrade by gateway, device, or group (location, type, etc.)
- Track upgrade progress
- Schedule update by gateway, device, or selection
- Update status notification
- Reports



# Firmware Repository

- Web Portal - <https://fwm.arrowconnect.io>
- Management portal for firmware owner of both gateway and device
- Track hardware versions and manufacturers
- Track software versions and compatability matrix between hardware and software versions
- Track upgrade / downgrade path between different software versions
- Secured repository of firmware binary files
- Manage Right-To-Use workflow between firmware owner and asset owner

# Firmware Repository - Lab

- Create hardware manufacturer Apple
- Create hardware iPhone with 2 versions - iPhone 8 and iPhone X
- Create software manufacturer Apple
- Create software iOS with 2 version – iOS 10 and 11
- Map software to different hardware versions and update path

# Firmware Management Process

- Feature requires additional license
- Manage firmware update jobs
- Job includes update requests for one or more devices of different firmware versions that are eligible to receive update of the targeted version. For example, you can schedule devices running firmware version 1.0 and 2.0 together in the same job that has target update of version 3.0 if the upgrade path in firmware repository allows it
- Job could be either on-demand (manual push) or scheduled
- Platform has background process to manage firmware update job. Update requests are sent to gateways asynchronously and the platform keeps track of all update progress. Email notifications can be set up for job status update. Current job status can also be viewed in web portal

# Firmware Management - Lab

- Create 2 devices of type iPhone – 1 iPhone 8 and 1 iPhone X
- Map asset type to firmware repository
- Configure them to run iOS 10
- Schedule update job for both to iOS 11
- Use FOTA simulator to respond to update requests, mark one request succeeded and one failed

# Firmware Manufacture Test Management

- Test Procedure Respository
- Historical test results for gateways and devices
- Full API for integrated manufacturing testing
- Manual and Automated Testing
- Reports

# Arrow Connect IoT - API & SDK

# REST API

- Built on top of Arrow Connect PaaS API layer, this REST API leverages the platform Access Key authentication and authorization mechanism
- API provides full integration into IoT platform. Our gateway software (introduced later) use the same API to communicate with the cloud
- Swagger UI is available to help with development and testing  
<https://api.arrowconnect.io/swagger-ui.html>
- API URL contains the name of the zone and region that the application is deployed
  - For example: URL for zone 02 in region B is <https://api-b02.arrowconnect.io>
  - Default zone (developer zone) is A01 and can be referenced with or without the suffix  
<https://api-a01.arrowconnect.io>  
<https://api.arrowconnect.io>
- API categories
  - Gateway API – gateway to cloud integration
  - Admin API – cloud to cloud integration

# Telemetry API

- Platform supports MQTT for sending telemetry to Arrow Connect cloud. When deployed together with IBM, Microsoft, or AWS integration, MQTT is also their recommended protocol
- REST API is also supported (not recommended)
- For streaming telemetry out of the platform, the following methods are supported
  - WebSocket
  - AMQP
  - MQTT (Q1/2018)
  - HTTP POST (Q1/2018)
- Telemetry querying with pagination via REST API is also supported
- Data extraction tool is available to help with extracting large amount of data for data warehousing or importing into other analytical tools



# Software Development Kits (SDK)

- SDKs are provided and maintained by Arrow for faster and better integration development with the platform. SDK communicates with the platform via REST API
- SDKs are publicly available at <https://github.com/arrow-acis>
- Support many programming languages
  - Java
  - C/C++
  - iOS
  - Android
  - Python (Jan 2018)
  - C# (TBD)
  - Node.js (TBD)
  - PHP (TBD)
  - Perl (TBD)
  - Ruby (TBD)
  - Go (TBD)

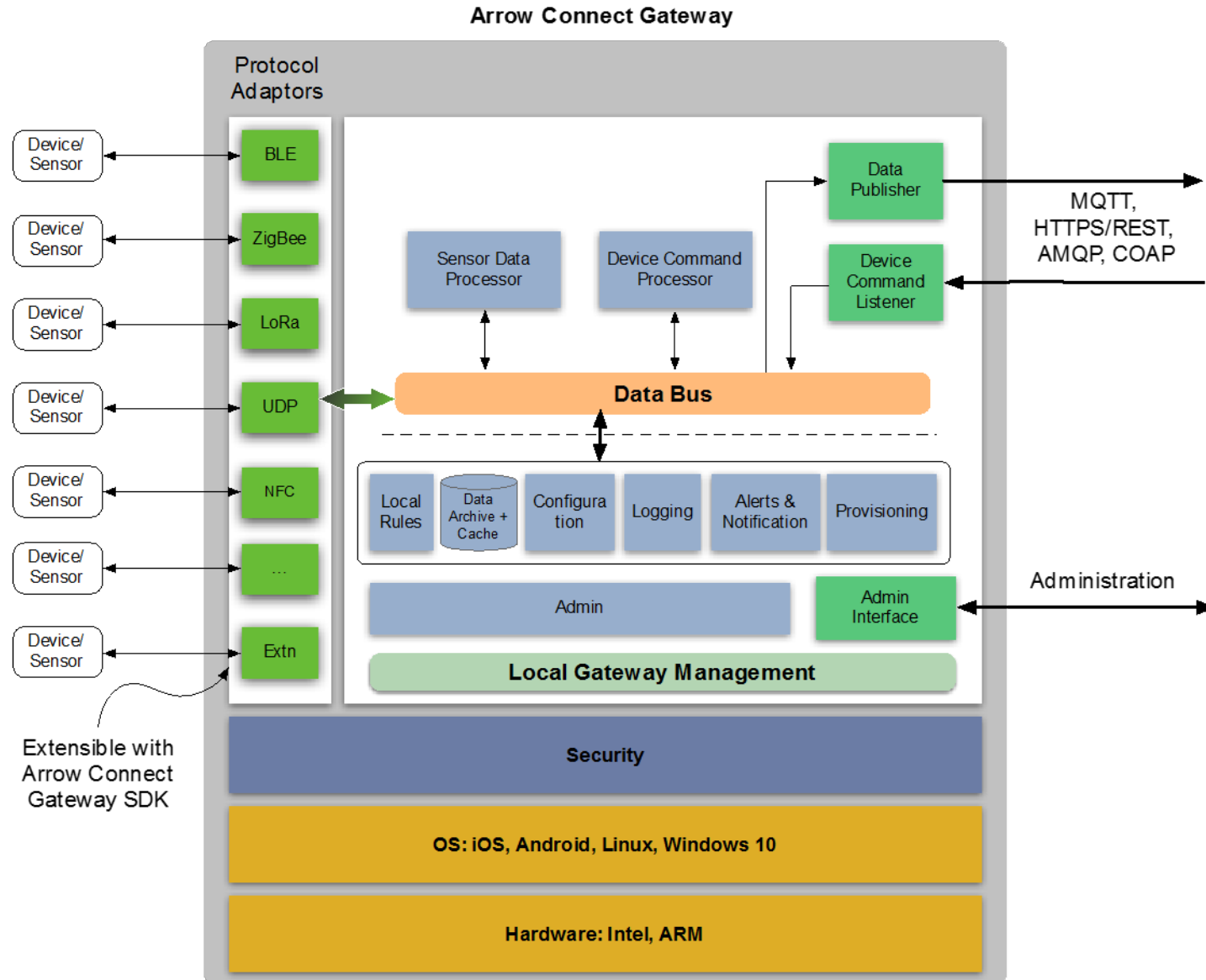
# Embedded SDK

- Complete C-SDK for REST API methods and data integration via MQTT
  - <https://github.com/arrow-acn/acn-sdk-c>
  - ANSI C standard
  - Portable to different compilers
- Reference Firmware
  - <https://github.com/arrow-acn/acn-embedded>
  - STM32 Nucleo, IDW01M1, IKS01A1, ARM mbed
  - ARIS board, Renesas S7, ThreadX RTOS
  - SX-ULPGN (QCA-4010), Silex Xtensa, ThreadX RTOS
  - ST SensorTile, STM32Cube
  - SenseAbility, Cypress PSoC
  - STM32L475 IoT Node Kit
  - OnSemi BB-GEVK, ARM mbed
  - Arrow Cypress Quadro IoT Wi-Fi kit
- Reference firmware fully support FOTA update via Firmware Management Process

# Arrow Connect IoT Gateway

# Arrow Connect IoT Gateway

- ❑ Full-featured gateway software – **Production Ready**
  - Linux
  - Windows
  - iOS
  - Android
- ❑ Modular design for Plug-and-Play (device adapters, databuses, cloud connectors, databases)
- ❑ Many out-of-the-box protocol support
- ❑ Extensible by Arrow and third-party software vendors
- ❑ Local management via web portal - **Optional**
- ❑ Edge analytics
- ❑ Pre-certified for many HW and SW design
- ❑ Multiple configurations from small footprint to full scale gateway solutions
- ❑ Easy integration with third-party gateway architecture



# Mobile Applications

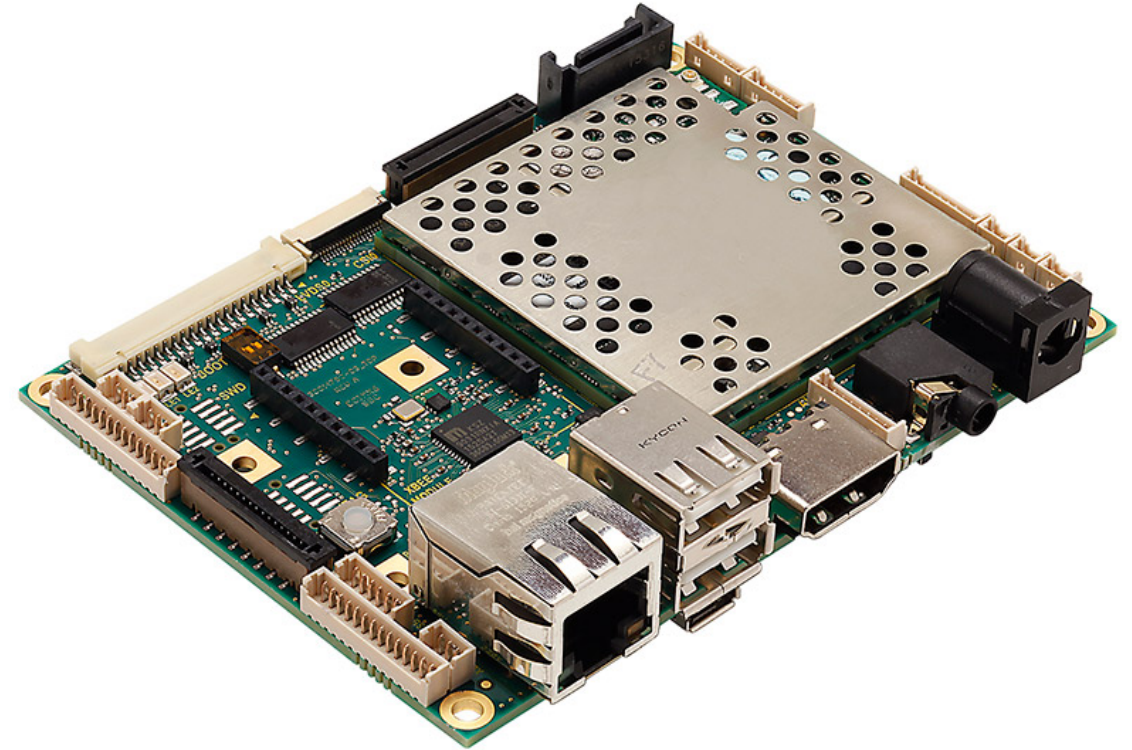
- iOS and Android mobile applications are available in App Store and Google Play. Search for "Arrow Connect Gateway"
- Application functions both as a gateway and a client application
- Support multiple user profiles. Each profile is associated with an application instance. Application instances from different tenants are also allowed
- Support many BLE development kits
- Connect to multiple BLE devices simultaneously
- Manage device actions
- Telemetry summary and historical
- Support GPS

# Mobile Applications - Lab

- Acquire the Application Code of the application instance on Arrow Connect Portal
  - Administration → Settings → Application → Application Code
- Register new Profile or add Profile to existing installation with Application Code
- The mobile phone itself has many internal sensors which is a good demonstration of an IoT device
  - Send telemetry to cloud
  - Control device from cloud
  - Discover other functionalities – dashboard, historical data, configurations, actions

# Digi ConnectCore 6 SBC

- <https://www.digi.com/products/embedded-systems/single-board-computers/connectcore-6-sbc>
- Ultra-compact off-the-shelf SBC
- NXP i.MX6 CPU (Quad, Dual, Lite)
- 4 GB eMMC
- 512MB / 1GB DDR
- Wi-Fi
- Bluetooth
- Xbee Socket
- HD video processing
- Display - HDMI, LVDS, Parallel LCD, MIPI-DSI
- Camera – Parallel, MIPI CSI-2
- CAN Bus
- GPIO/I2C/SPI/UART
- Pre-installed with a custom Yocto image for Arrow Connect Gateway based on Yocto 2.2 and Digi BSP





# Additional Modules

- Digi XBee ZigBee S2C RF module
- Silicon Labs EM357 chipset
- <https://www.digi.com/resources/documentat ion/digidocs/pdfs/90002002.pdf>
- Demonstrate ZigBee functionality

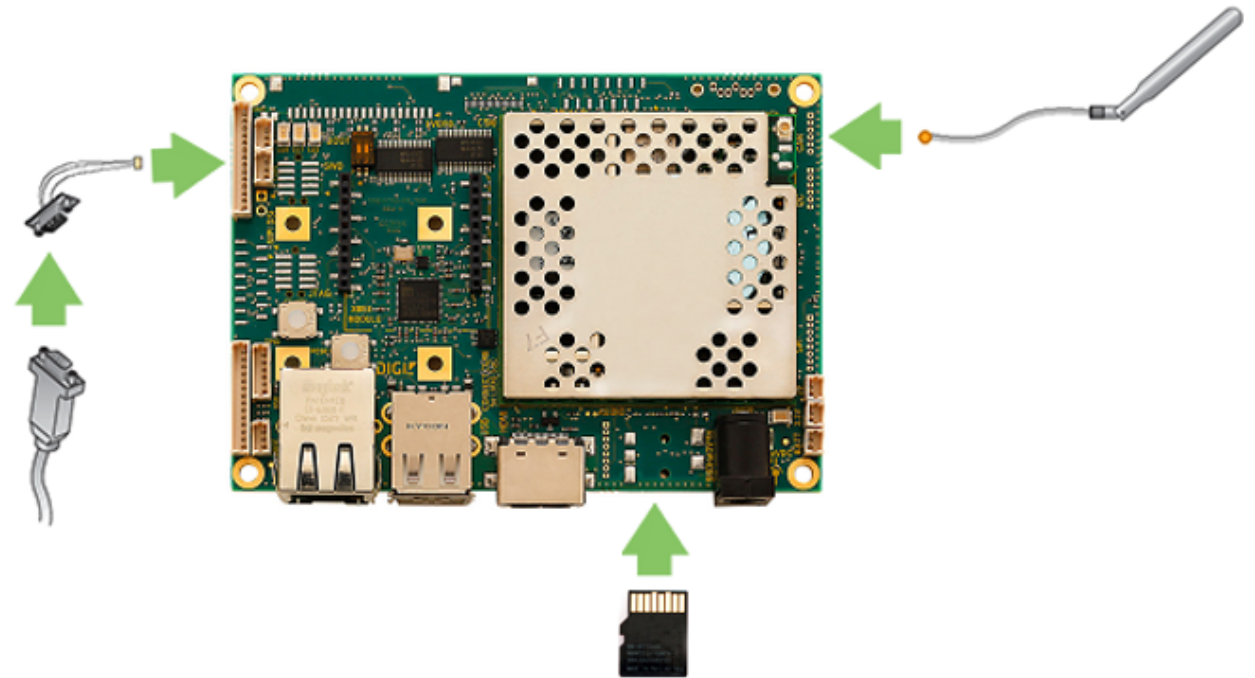


- Silicon Labs Thunderboard Sense
- <https://www.silabs.com/products/development-tools/thunderboard/thunderboard-sense-kit>
- Demonstrate BLE functionality



# Prepare System Console

- Connect serial cable with small connector to SBC (as shown)
- Connect F-F serial cable
- Connect Serial-to-USB cable
- Connect USB end to your laptop
- Connect antenna to SBC (as shown)
- Download [TeraTerm](#)
- Launch TeraTerm and connect to COM port using the settings  
**115200-8-N-1-N**
- Power up SBC
- Serial console should display boot up console output and stop at login prompt
- Log in as **root**, password **root**



# Set up Wi-Fi

- Make sure Wi-Fi device is detected by OS  
`nmcli dev`
- Create and activate Wi-Fi connection. This command can be used to create more than one Wi-Fi connection  
`nmcli dev wifi connect <SSID> password <PASSPHRASE>`
- Connection files can be found under directory  
`/etc/NetworkManager/system-connections`
- To connect to the default Wi-Fi connection manually  
`nmcli dev con wlan0`
- To disconnect Wi-Fi manually  
`nmcli dev dis wlan0`
- More nmcli options can be found here  
<https://manpages.debian.org/stretch/network-manager/nmcli.1.en.html>

# Install Arrow Connect Gateway

- At command prompt, run the following commands

```
mkdir /opt  
cd /opt  
wget https://content.arrowconnect.io/public/selene/latest/ccimx6/install.sh  
sh install.sh
```

- Log on to Arrow Connect Web Portal <https://portal.arrowconnect.io> and look for Access Key with name ApplicationOwnerKey. Copy Selene AES-128 Api Key and Secret Key
- Edit main gateway configuration file using your favorite text editor (vi, nano, etc.) to add the API and Security Keys above

```
nano /opt/selene/config/devices/self.properties
```

- To start gateway software

```
/etc/init.d/selene start
```

- To stop gateway software

```
/etc/init.d/selene stop
```

- To "tail" log file

```
tail -f /opt/selene/log/app.log
```

# Arrow Connect Gateway (Selene)

- Fully loaded, production-ready Arrow Connect gateway developed in Java
- Run on most Linux distros (Debian, Ubuntu, Yocto, mLinux, etc.) and Windows 10 IoT Enterprise
- Tested on many development boards and commercial gateways – Dragonboard, Inforce, ConnectCore 6, Edison, Dell, HP, Advantech, etc.
- Contains Main Gateway Engine and optional Web Portal (available Q1/2018)
- Embedded database – supports H2, SQLite and MySQL
- Data bus for internal messaging and queuing – supports File, Redis, MQTT (Mosquitto) and RabbitMQ
- More than a dozen out-of-the-box PnP device adapters to support different data protocols - BLE, ZigBee, DUST, LoRa, GPIO, I2C, Modbus, etc.

# Selene Directory Structure

- Default installation directory – `/opt/selene`
- `bin/*` – scripts
- `config/devices/*` – configuration files of all bootstrap devices
- `config/devices/self.properties` – gateway configuration file
- `config/log4j2.xml` – log file settings
- `config/selene.properties` – global configuration file
- `config/samples/*` – sample configuration files of supported devices
- `db/*` – database files (for H2 and SQLite)
- `databus/*` – message queue files (for FILE database)
- `lib/*` – application binaries
- `log/*` – log files
- `update/*` – software upgrade files (for firmware OTA)

# Selene on Digi ConnectCore 6 SBC

- The Yocto image on your ConnectCore 6 SBC was built with some of the software components pre-installed to support Selene
  - MySQL is used for Selene database backend
  - Mosquitto (MQTT broker) is used for Selene data bus
  - BlueZ is the BLE software stack to support Thunderboard Sense
- Selene has an Arrow proprietary implementation of ZigBee HA 1.2 stack which leverages Digi Java SDK to communicate with the XBee ZigBee S2C module via UART
- Local configuration files are meant to be bootstrap for an initial Selene installation. After starting Selene for the first time, the configuration files are loaded into the database and will be ignored in the future since the database are kept synchronized with the cloud in both direction. The database can also be managed by Selene web portal (available in Q1/2018). If you make any change to the configuration files, you need to reset the local database
  - `/etc/init.d/selene stop`
  - `/opt/selene/bin/reset-database.sh`

# Bluetooth Low Energy (BLE)

- Bluetooth Low Energy (BLE, Bluetooth Smart) is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries. Compared to Classic Bluetooth, Bluetooth Smart is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range
- 2.4 GHz
- AES-128 encryption
- Low Power Design - Keeping radio off
- Advertising Channels vs Data Channels
- Generic Attribute Profile (GATT)
  - Services - <https://www.bluetooth.com/specifications/gatt/services>
  - Characteristics - <https://www.bluetooth.com/specifications/gatt/characteristics>
  - Descriptors - <https://www.bluetooth.com/specifications/gatt/descriptors>
- Data is exposed as Attributes – handle / type / value
- Selene supports both Bluetooth device pairing and Bluetooth beacons



# Silicon Labs Thunderboard Sense

- Wireless Gecko EFR32 MCU
  - ARM Cortex M4 core, 256 kB RAM, 1024 kb Flash
  - ZigBee and Thread Radio
- Integrated with lots of sensors
  - Relative Humidity and Temperature
  - UV and Ambient Light Sensor
  - Pressure Sensor
  - Indoor Air Quality and Gas Sensor
  - 6-axis Inertial Sensor
  - Hall-effect Sensor
- Misc
  - LEDs
  - Digital Microphone

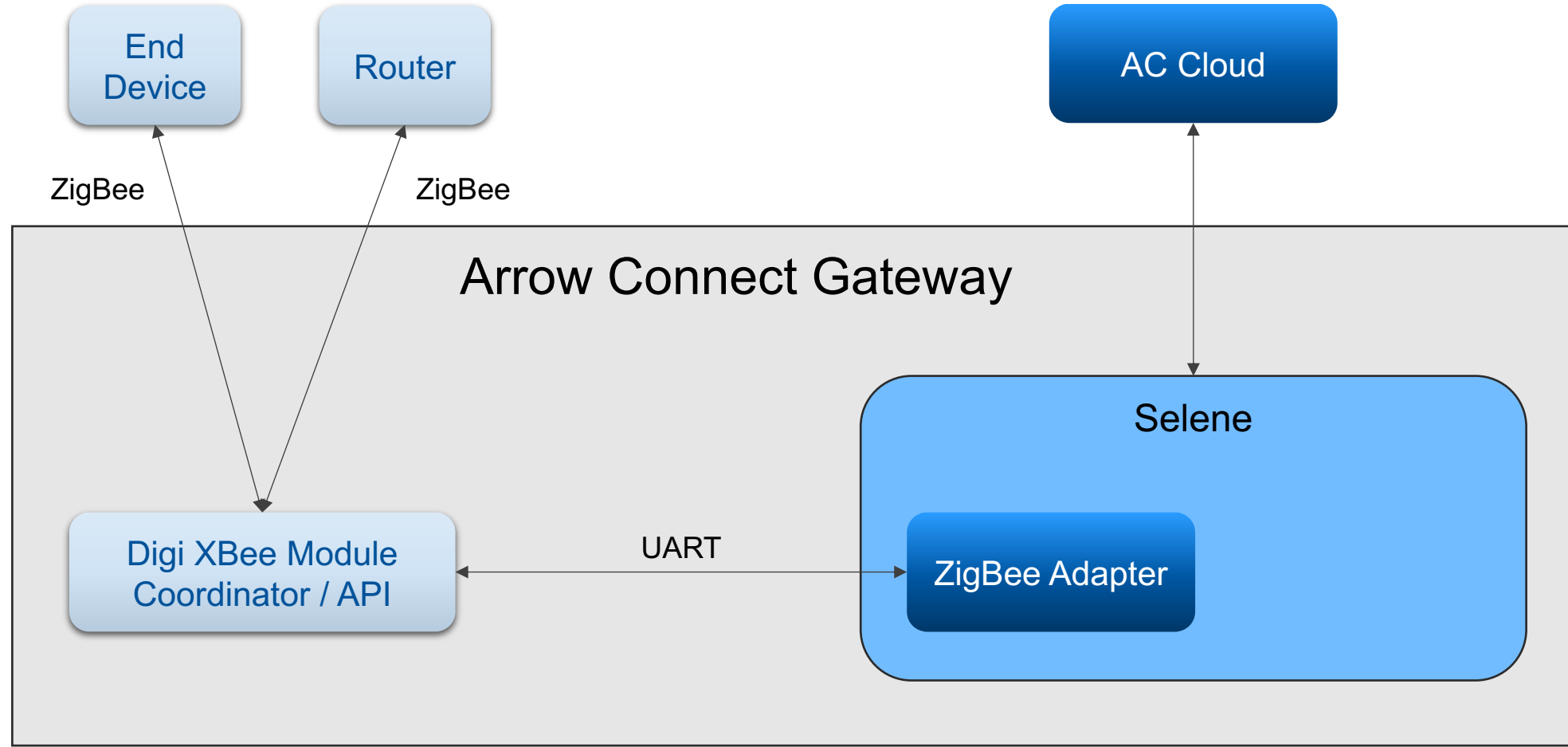
# Set up Thunderboard Sense

- Make sure selene is not running  
`/etc/init.d/selene stop`
- Download device configuration file  
`cd /opt/selene/config/devices`  
`wget https://content.arrowconnect.io/public/training/12-2017/thunderboard-sense.properties`
- Edit file thunderboard-sense.properties to enter your Thunderboard Serial Number and MAC Address  
`name=Thunderboard Sense <serial-number>`  
`uid=thunderboard-sense-<serial-number>`  
`bleAddress=<mac-address>`
- Reset local database  
`/opt/selene/bin/reset-database.sh`
- Start selene  
`/etc/init.d/selene start`
- Wait about 2 minutes for Selene to fully come up. Switch on Thunderboard and notice a green LED is flashing. Push one of the black buttons on the Thunderboard once to start pairing. When the pairing process is complete, the LED will stop flashing, and it's now sending telemetry to Selene
- Look for your Thunderboard Sense device showing up in Arrow Connect Portal

# ZigBee Integration (1/2)

- Use Digi XBee ZigBee S2C module
  - Silicon Labs EM357 SoC with EmberZNET PRO stack
  - IEEE 802.15.4 Radio
  - Mostly 2.4GHz, some sub-GHz
  - Reliable, Interoperable and Low Power
  - 3 types of nodes – coordinator, router, and end devices
  - Supports both Zigbee Mesh and DigiMesh
  - DigiMesh is Digi's proprietary mesh network
- Set up module to run as the coordinator node
  - Default firmware is running DigiMesh
  - Configure firmware to run in bypass mode to support ZigBee Mesh
- Support ZigBee Home Automation 1.2
  - Implement custom ZigBee adapter to fully support HA 1.2 specification  
<http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeehomeautomation/>
  - Provide low-level API to interact with ZigBee network from the cloud

## ZigBee Integration (2/2)



# Set up ZigBee Motion Sensor

- Make sure selene is not currently running

```
/etc/init.d/selene stop
```

- Download device configuration file

```
cd /opt/selene/config/devices
```

```
wget https://content.arrowconnect.io/public/training/12-2017/zigbee-ha.properties
```

- Reset local database

```
/opt/selene/bin/reset-database.sh
```

- Start selene

```
/etc/init.d/selene start
```

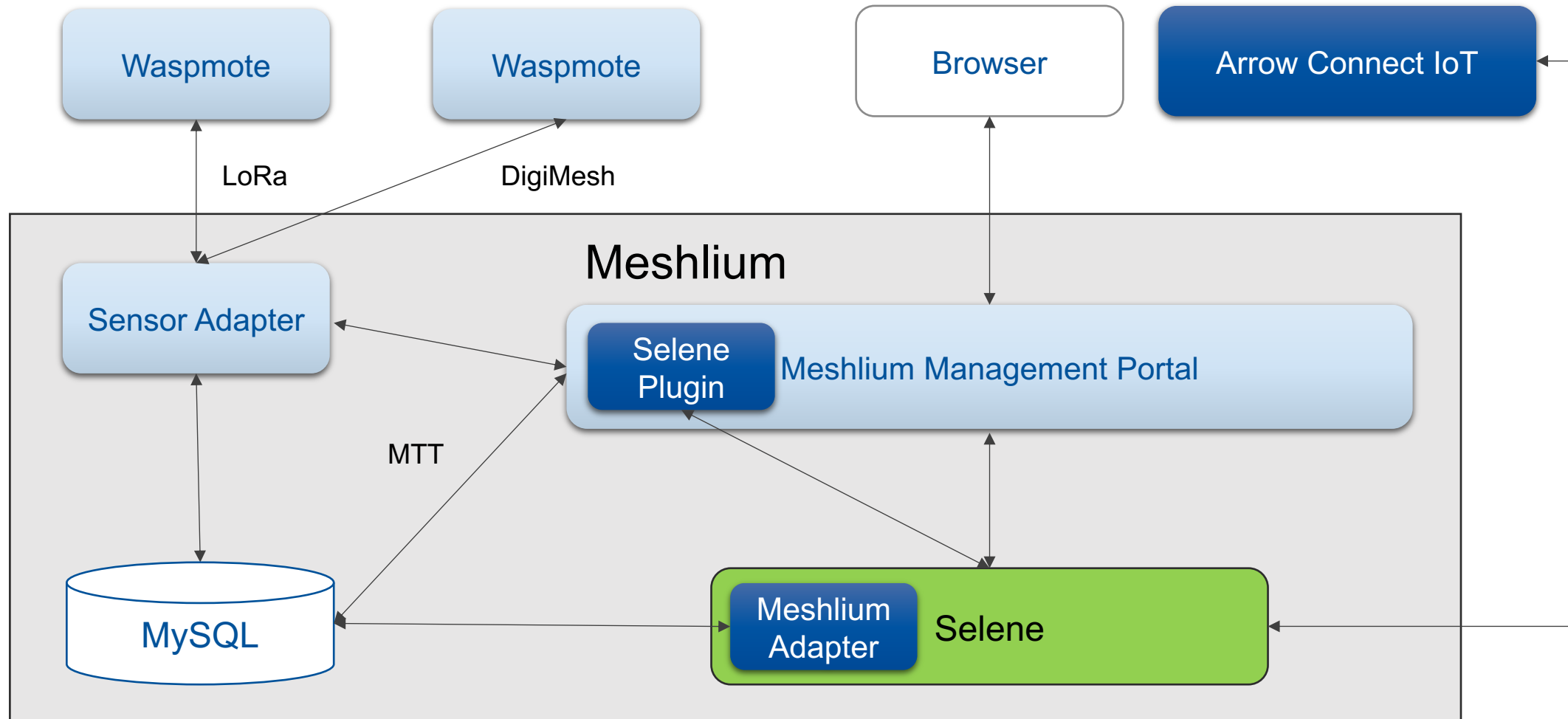
- Look for your Thunderboard Sense device showing up in Arrow Connect Portal

# Libelium Meshlium

- Libelium Meshlium is fully certified for Arrow Connect Gateway. Our software is bundled with the Meshlium Gateway out-of-the-box  
<http://www.libelium.com/arrow-electronics-extends-iot-offering-with-libelium-agreement/>
- Check out developer hub for more information  
<https://developer.arrowconnect.io/docs/libelium-meshlium>



# Meshilum – Selene Integration



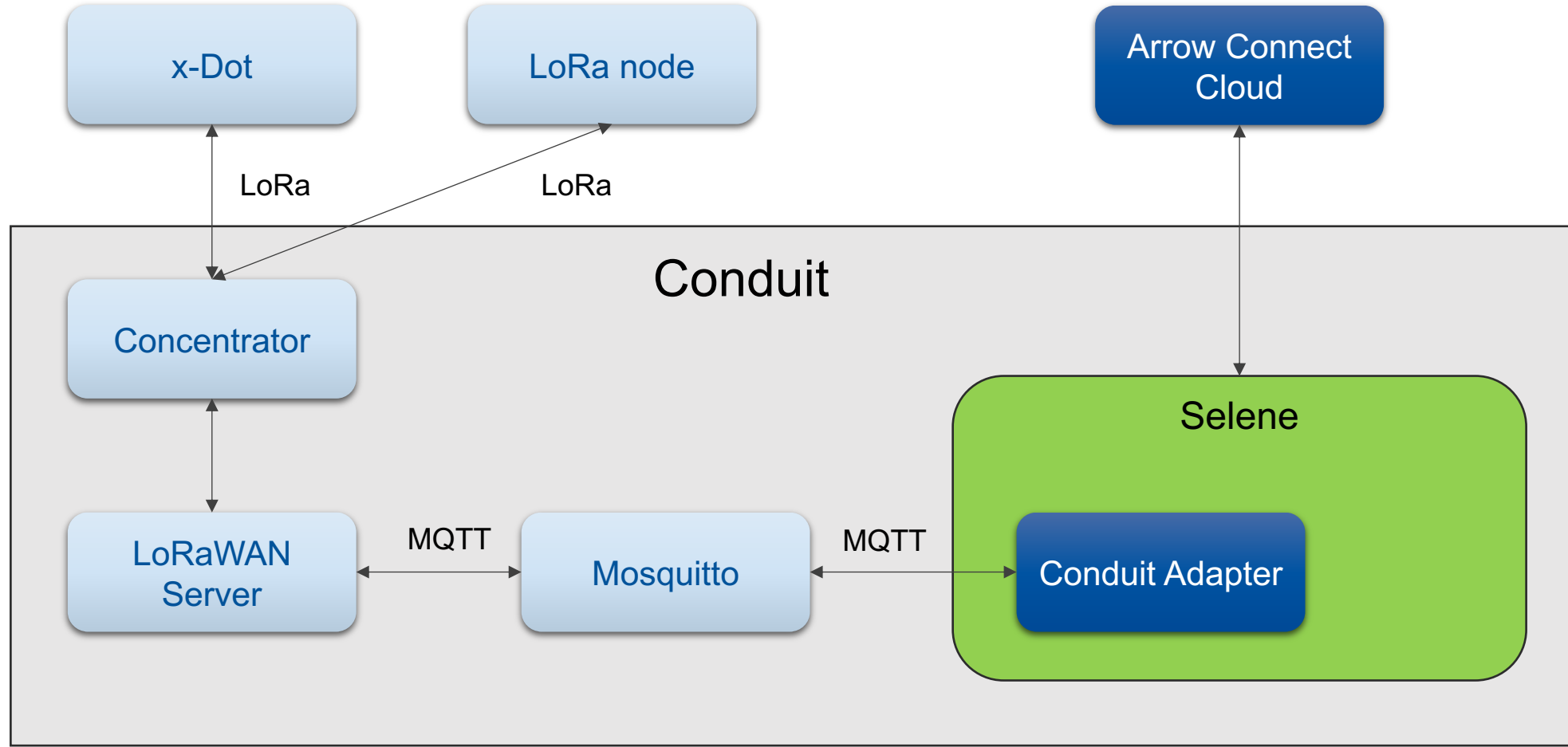
# Multitech Conduit – LoRa and LoRaWAN

- Multitech Conduit is fully certified to run Arrow Connect Gateway to manage LoRa devices using local LoRaWAN server on the Conduit
- Arrow Connect also supports Multitech Conduit with LoRaWAN implementation by leveraging LORIoT.io as a middleware LoRaWAN server in the cloud. We developed a back-end integration that integrates LORIoT.io into Arrow Connect IoT Cloud
- Native support of LoRaWAN implementation in Arrow Connect IoT is planned for Q1/2018

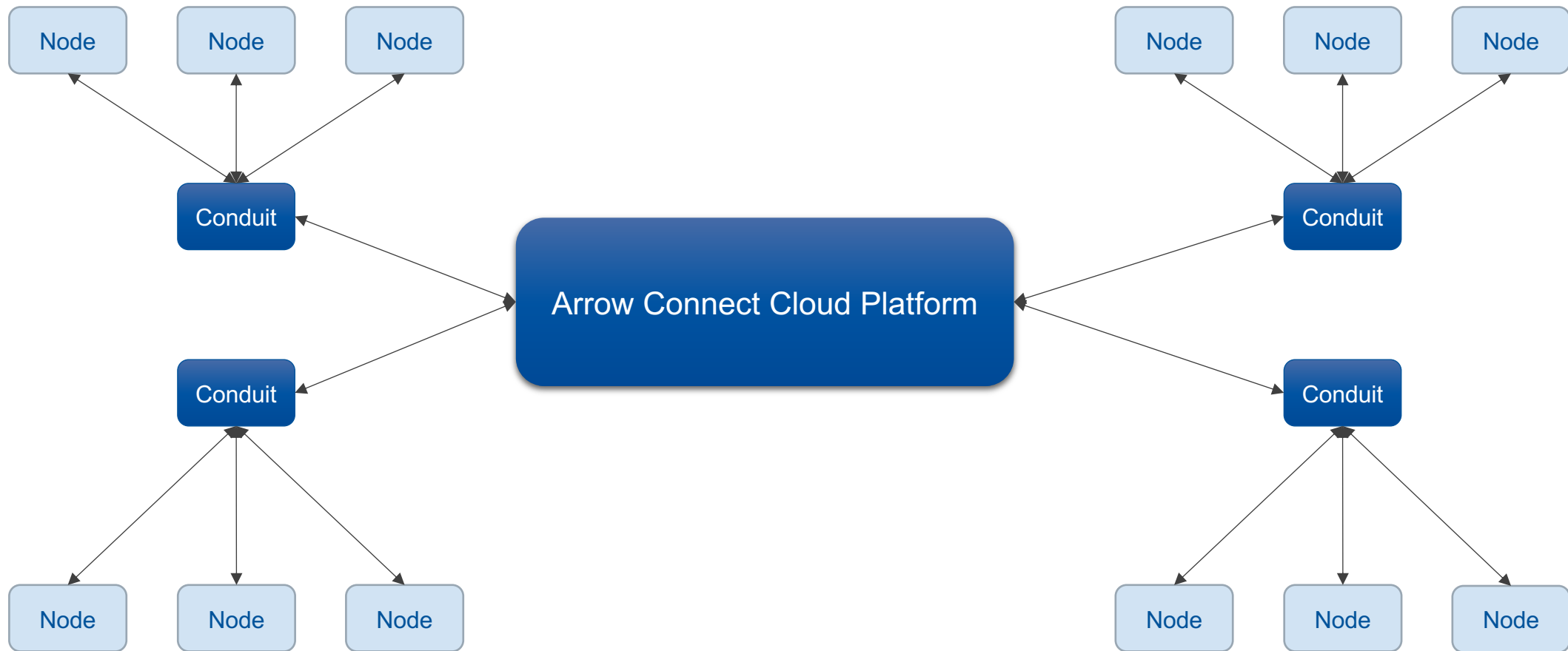




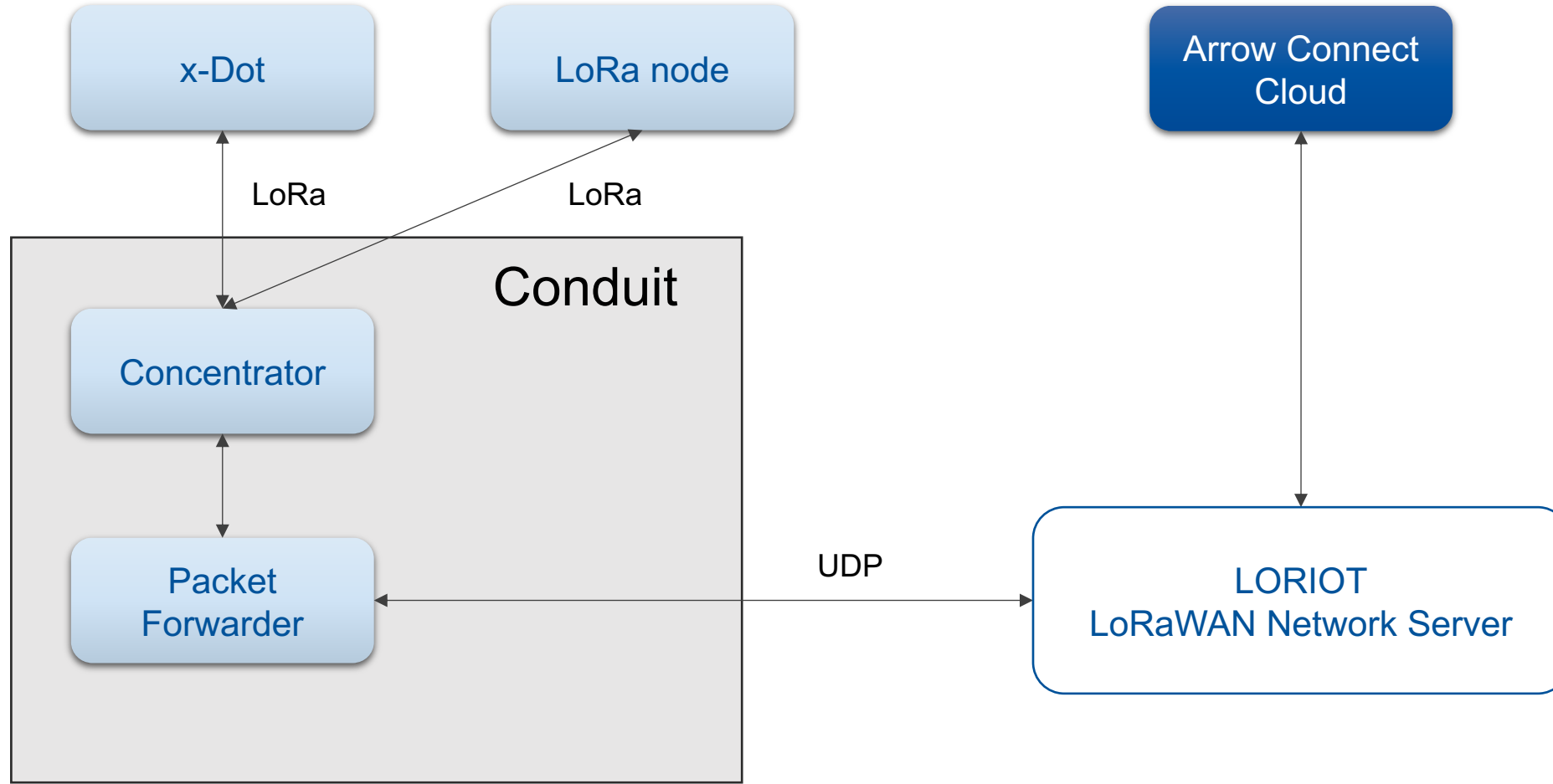
# Conduit – Selene Integration (LoRa) (1/2)



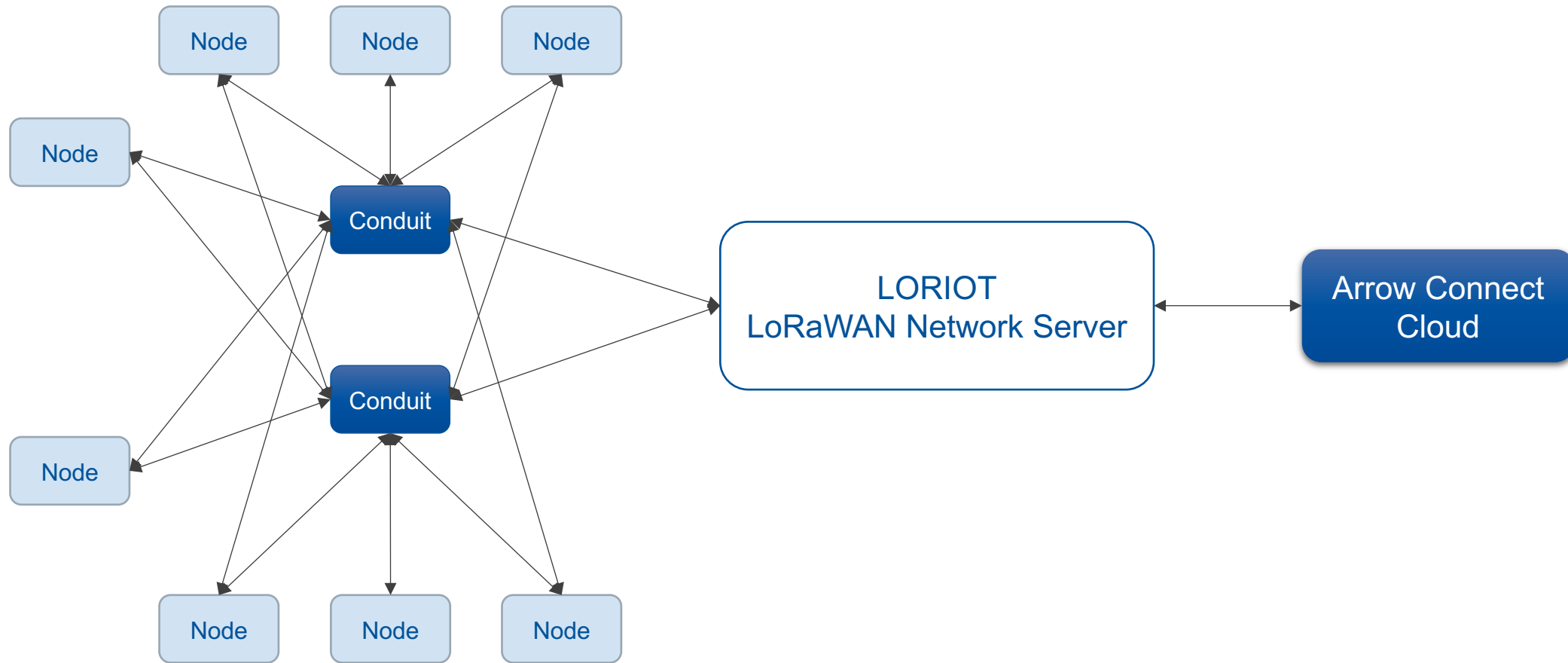
## Conduit – Selene Integration (LoRa) (2/2)



# Conduit – Arrow Connect (LoRaWAN) (1/2)



## Conduit – Arrow Connect (LoRaWAN) (2/2)

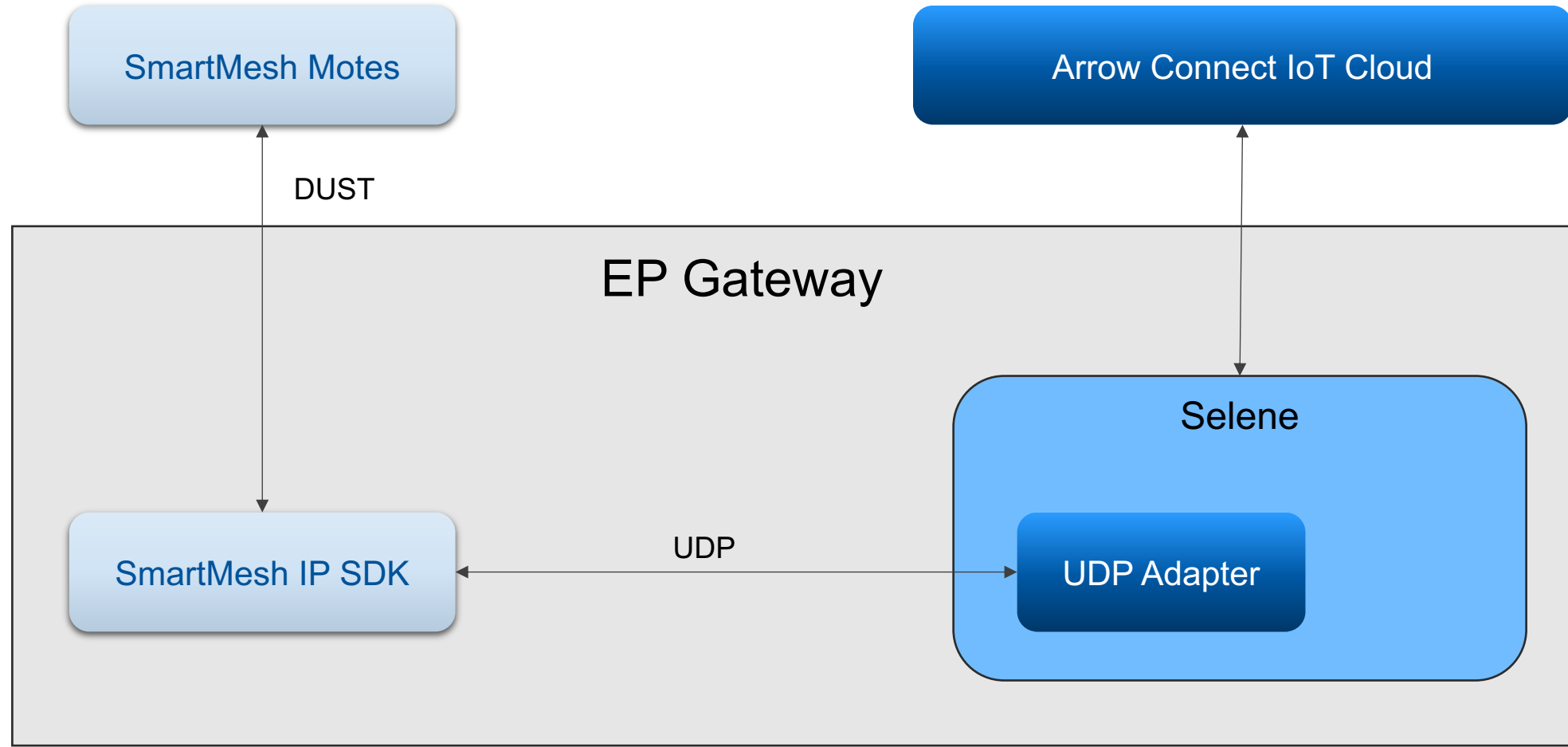


# Embedded Planet – SmartMesh IP

- Embedded Planet Gateway with SmartMesh-IP sensor nodes are fully certified with Arrow Connect Gateway
- Implementation uses Selene's generic UDP device adapter to communicate with the local gateway software stack



# SmartMesh IP – Selene Integration



# Deployment Use Cases

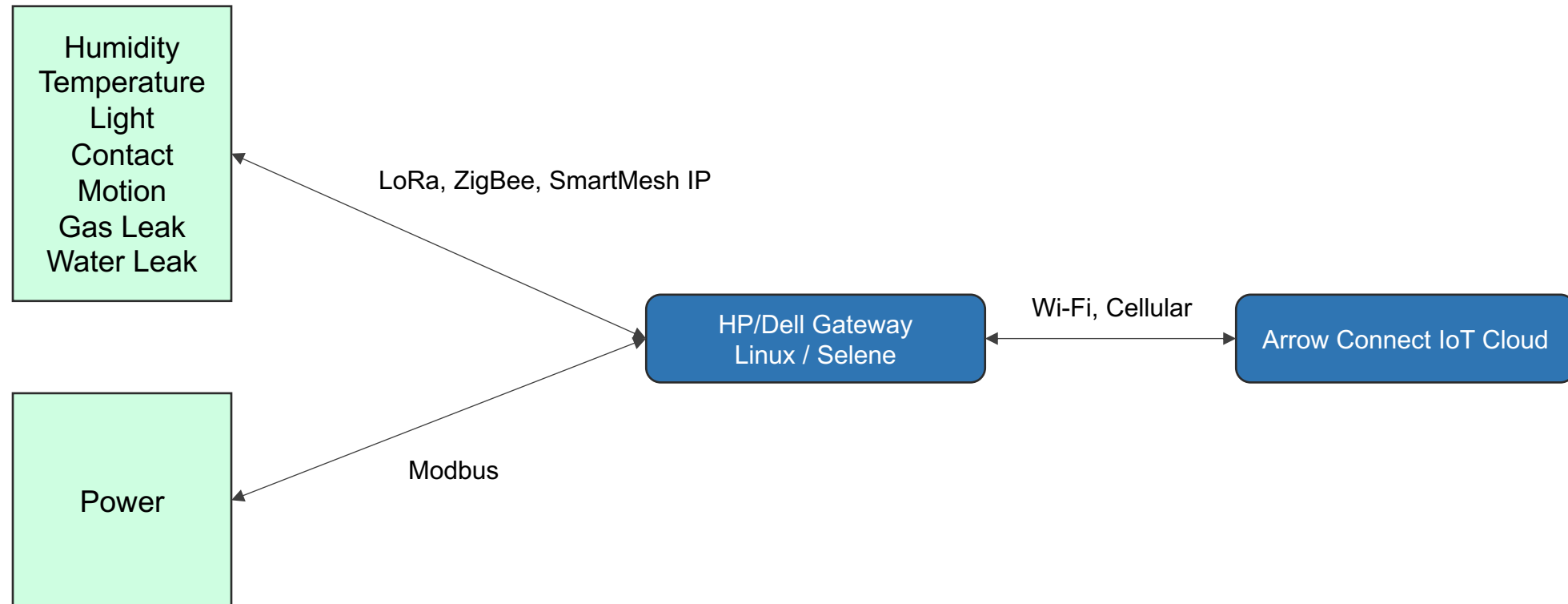
---

# Smart Buildings - Requirements

- Need to cover large areas
- Mesh network is not required
- Mixture of wired and wireless sensors
- Could have large number of sensors and devices
- May require more than one gateway
- Enterprise grade gateways
- Local device and gateway management
- Wi-Fi or Cellular



# Smart Buildings - Suggestions



# Smart Cities - Requirements

- May need to cover very large areas
- Mesh network is often required
- Mostly wireless sensors
- Could have very large number of sensors and devices
- Require many gateways
- Rugged gateways
- Cellular

# Smart Cities - Suggestions

