# IBAS: Mappings and Constraints

Overview

There are four primary components used in a system using IoT Bridge. These are:

- Edge Gateways publishing Sparkplug based MQTT data
  - Our examples will use Ignition+MQTT Transmission for the Edge Gateway.
- An MQTT Server
  - This could be AWS IoT Core or any other v3.1.1 compliant MQTT Server
- IoT Bridge for AWS SiteWise (IBAS)
- AWS IoT SiteWise
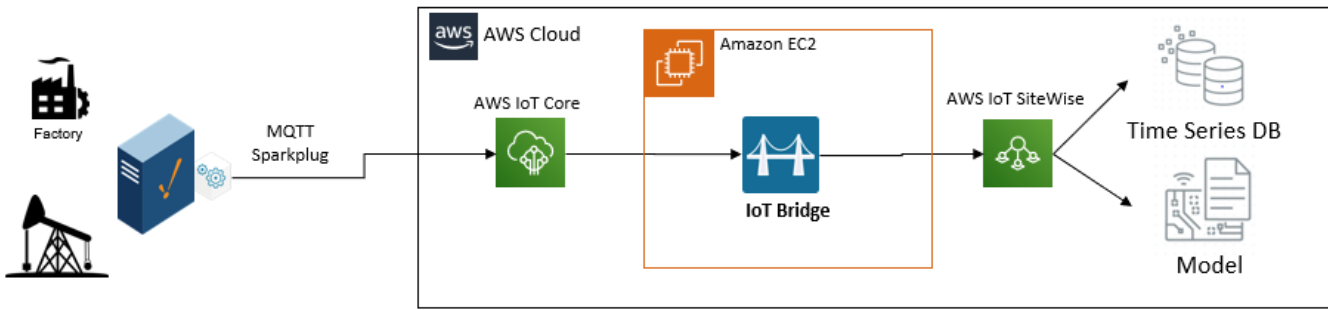
There are some translations that need to take place between the components and these platforms have different capabilities that result in some limitations in how an Edge Gateway can be configured. For example, UDTs in Ignition support more data types that AWS IoT SiteWise does. As a result, use of data types in Ignition must be limited to those supported by AWS IoT SiteWise in order for data to properly show up and be updated in AWS SiteWise.

Mappings

Ignition and AWS IoT SiteWise use different terminology for similar constructs. Those definitions are below.

- An Ignition 'UDT Definition' is mapped to a SiteWise 'Asset Model'.
- An Ignition 'UDT Instance' is mapped to a SiteWise 'Asset'.
- Ignition 'UDT parameters' are mapped to 'SiteWise Attributes'.
- Ignition 'UDT member tags' are mapped to 'SiteWise Measurements'.
- Transform, Metric, and Hierarchy definitions in AWS SiteWise are not used by IB.

Constraints and Limitations

Supported Data Types

AWS IoT SiteWise only supports Booleans, Integers, Doubles, and Strings. As a result, this is the list of data types in Ignition that are supported for consumption into AWS IoT SiteWise

- Boolean
- Byte
  - Will be up-casted to an Integer in SiteWise
- Short
  - Will be up-casted to an Integer in SiteWise
- Integer
- Float
  - Will be up-casted to an Double in SiteWise
- Double
- String

Any non-supported data type will be converted to a String in SiteWise and the measurement value for any instance will show an error with a message similar to "*Long is not a valid datatype*"

User Defined Types (UDTs)

- SiteWise Asset Models have a limit of 200 Attributes and/or Measurements. Therefore UDTs with more than 200 Parameters or member tags will result in the Model not being created in SiteWise.
- UDT Tag and Property names that result in creating SiteWise Measurements and Attributes name, you must follow the regex expression pattern
  - ^[a-zA-Z0-9_-]+$ (for more information on regex expression patterns go to https://regexr.com)
- UDT parameter 'Integers' in Ignition are actually Long values. As a result, UDT parameters of type Integer are not supported at this time.

Nested/Hierarchical User Defined Types (UDTs)

These are not supported currently in IoT Bridge. AWS SiteWise allows Asset Models to have one or more child Asset Models. Ignition supports UDTs having children UDTs as well as UDTs referencing a parent UDT. However, these hierarchical relationships will not be maintained by IB. Support for simple hierarchies as supported by AWS IoT SiteWise will be added in a future version of IoT Bridge.

AWS IoT SiteWise Limits and Quotas

AWS IoT SiteWise has limits and quotas on many different aspects of asset model counts, asset counts, number or attributes per model, etc. For details on those limits see this document.

MQTT Server

- AWS IoT Core has a maximum message size limited to 128 kiloBytes. This limitation can be overcome by using the Cirrus Link Chariot MQTT Server.
- AWS IoT Core does not support use of Sparkplug's 'Primary Host ID' feature. This limitation can be overcome by using the Cirrus Link Chariot MQTT Server.