REEKOH

Pipeline Design and Messaging Protocol Specification

AVEVA System Platform / IBM Maximo Integration

Version:

1.0

Change Record

Date	Author	Version	Change reference
28/4/20	Aaron Watkins	1.0	Initial Version

Contents

High Level Description

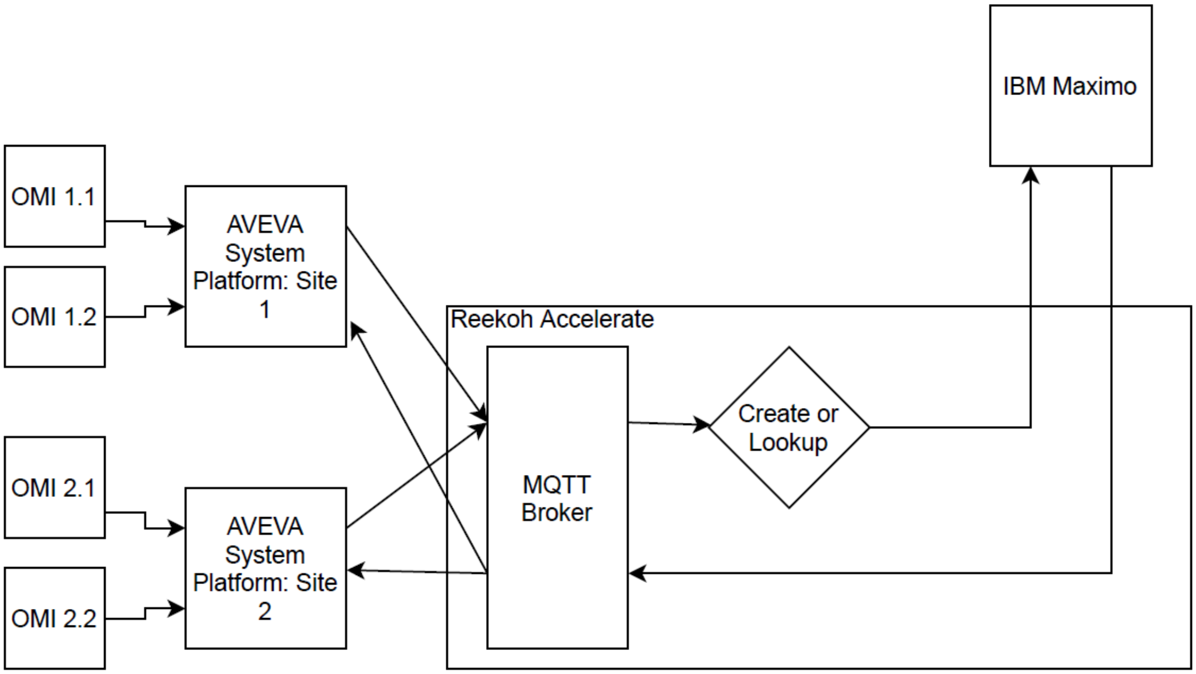
As part of the Solution Ready series integration solutions being offered by Reekoh, we are demonstrating capability to integrate AVEVA System Platform, HMIs or OMI applications to IBM Maximo, which is a work order management system commonly used by AVEVA customers.

There are three scenarios that we desire to cater for in this solution:

1. As a user I want to be able to see if a piece of equipment has any outstanding maintenance work orders
2. As a user I want to be able to look at the previous maintenance tasks undertaken on a piece of equipment
3. As a user I want to be able to initiate a work order from the process control (HMI) system.

Each one of these originates a request from AVEVA and ultimately returns a response to AVEVA.

Context Diagram

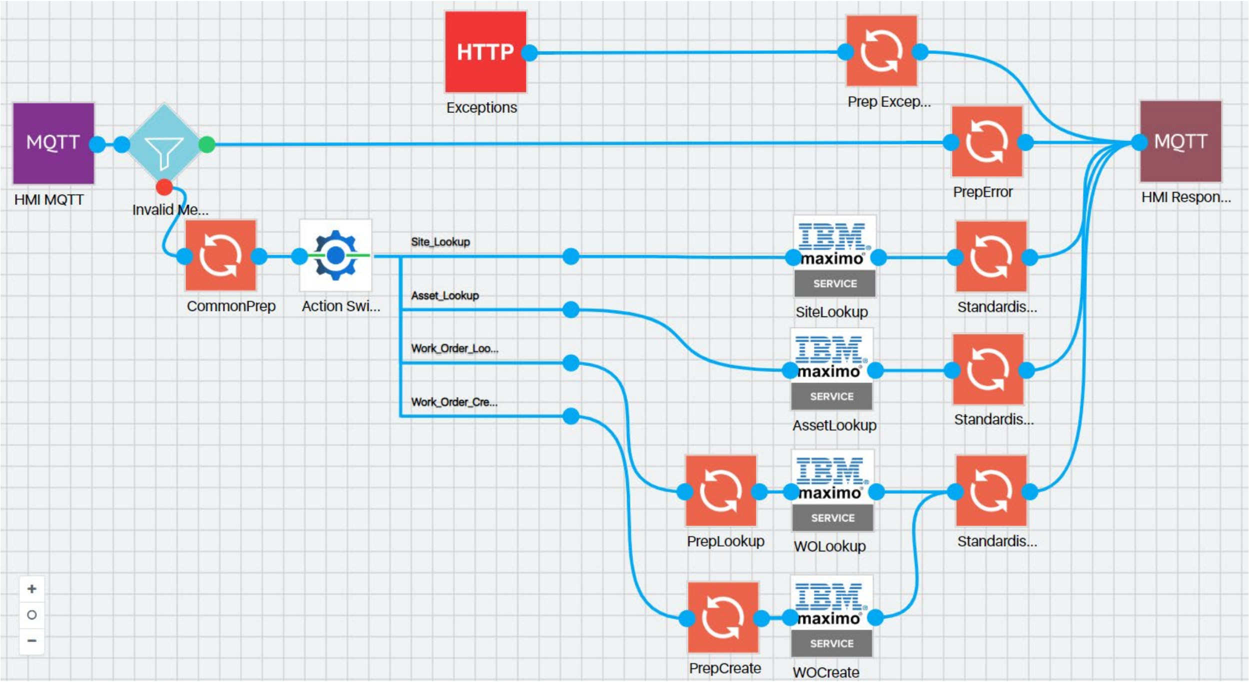


Multiple OMI apps may connect **directly** through AVEVA System Platform instances to Reekoh Accelerate™ via an MQTT Broker. Having said that, assuming the application is capable of acting as an MQTT Client, direct connection to Reekoh Accelerate™ is possible. The transmitted message will indicate the desired action to be performed on IBM Maximo. The result of that action will be published back to the MQTT Broker, allowing the end application to get the results.

Each OMI app will need to be registered as a device within Reekoh Accelerate™ allowing validation of clients, while also permitting the separation of messaging channels for each application.

Pipeline Details

Screenshot



MQTT Gateway

The MQTT Gateway is going to be used for two purposes – data ingress and also as a channel for data egress, facilitating the request/response nature of the pipeline.

The topic structure will be defined like such:

- Data Ingress: data/in/
- Data Egress: data/out/<device> and data/out/errors

The device is identified through the Client ID.

Relevant configuration will be:

- Data Topic: data/in
- Allowed Topics: data/out/#
- Username/Password: <relevant to customer>
- Disable Gateway Client ID Validation: false
- Identify Device through: Client ID
- Type of Data: device

Note – device data is only used as part of the validation of OMI devices and to validate outbound topics. Having validated the device, one can be assured that the response communication channel can be identified. The MQTT Gateway can also be configured with IP Address Range whitelisting if desired.

Messaging Interface

Inbound Messaging

Messages are to be transmitted in JSON format. All keys are specified in lower case. The basic structure is defined as per the following table

Key	Value Type	Sample Value	Description
id	Any Integer or String	28 "abc" "CL1_R1" "fred_R1"	A value that identifies the request to the originating system. Can be used to correlate responses to requests
action	String. Limited values	"create" or "lookup"	What action is desired on the target system. Note – not all actions are permissible with all resource types
asset	String	"11300"	Optional in some contexts
resource	String. Limited values	"wo", "asset" or "site"	The resource in IBM Maximo we wish to interact with: Assets, Work Orders or Sites
params	Object of keys and values. Permitted configuration dependent on combination of action and resources		

Permitted combinations of actions and resources:

- lookup/site: Get a list of sites. Asset is not permitted
- lookup/asset: Get a list of assets. Asset is optional
- create/wo: Create a new work order for an asset
- lookup/wo: Get a list of work orders for an asset

Invalid combinations of actions and resources will result in an error being returned.

Site Lookup

Site lookup lets you obtain a list of valid site identifiers

Asset should not be specified. No parameters are supported.

Sample Request

```
{
    "id": 28,
    "action": "lookup",
    "resource": "site",
    "asset": "",
    "params": {}
}
```

Asset Lookup

Asset lookup has two modes of operation:

- Lookup details on a specific asset
- Search for assets

When asset is specified, it is suggested that params not be specified beyond a site (given that an asset with a given name is unique only within the context of a given site) as otherwise, there will be a requirement for the asset to meet all search requirements.

Parameters

Key	Value Type	Sample Value	Description
site	String	"texas", "bedford"	Optional. A value that identifies the relevant site. Valid options can be inspected through use of the lookup/site function
location	String	"HWSTOCK"	Optional. Where the asset is located at a given site.
prefix	String	"TR"	Optional. When searching for an asset, finds an asset with a given prefix
type	String. Limited values	"IT", "GEN", "DIESELENG"	Optional. Instance specific asset type
startAt	Integer	0	Optional. Pagination control. Defaults to the value of zero
maxItems	Integer	10	Optional. Pagination control. Defaults to the value of five

Sample Request

```
{
    "id": 28,
    "action": "lookup",
    "resource": "asset",
    "asset": "11300",
    "params": {
        "site": "bedford"
    }
}

{
    "id": 28,
    "action": "lookup",
    "resource": "asset",
    "params": {
        "prefix": "TR",
        "type": "IT"
    }
}
```

Work Order Lookup

Work Order lookup allows you to search for work orders associated with a given asset and other, optional, search criteria.

Parameters

Key	Value Type	Sample Value	Description
site	String	"texas", "bedford"	Optional. A value that identifies the relevant site. Valid options can be inspected through use of the lookup/site function
status	String. Limited values	"OPEN", "CLOSE"	Optional. Whether interested in open or closed work orders.
workType	String		Option. Type of Work required. Values are implementation dependent
startCreateDate	Date (ISO Format)	2020-05-01	Optional. Look for work orders created after a date
endCreateDate	Date (ISO Format)	2020-05-02	Optional. If specified, *startCreateDate* is required, which must be an earlier date. Use to create a date range for work order creation.

AS AN ALTERNATIVE, DATE SEARCHES CAN USE OFFSETS

maxOffset	Integer	10	Optional. Look for work orders created in the last x number of days.
minOffset	Integer	5	Optional. If specified, *maxOffset* is required and *minOffset* must be less than *maxOffset*. Use to create a date range for work order creation.

startStatusDate	Date (ISO Format)	2020-05-01	Optional. Look for work orders with a status change after a date.
endStatusDate	Date (ISO Format)	2020-05-02	Optional. If specified, *startStatusDate* is required, which must be an earlier date. Use to create a date range for work order status change.

AS AN ALTERNATIVE, DATE SEARCHES CAN USE OFFSETS

maxStatusOffset	Integer	10	Optional. Look for work orders with a status change in the last x number of days.
minStatusOffset	Integer	5	Optional. If specified, *maxStatusOffset* is required and *minStatusOffset* must be less than *maxStatusOffset*. Use to create a date range for work order status change.

startAt	Integer	0	Optional. Pagination control. Defaults to the value of zero
maxItems	Integer	10	Optional. Pagination control. Defaults to the value of five

Sample Request

```
{
    "id": 28,
    "action": "lookup",
    "resource": "wo",
    "asset": "11300",
    "params": {
        "site": "bedford",
        "maxOffset": 10,
        "minOffset": 5
    }
}
```

Work Order Creation

Work Order creation is tied to a specific asset:

Parameters

Key	Value Type	Sample Value	Description
site	String	"texas", "bedford"	Required. A value that identifies the relevant site. Valid options can be inspected through use of the lookup/site function
description	String. Limited values	"My Work Order"	A literal description of the nature of the work order

Sample Request

```
{
    "id": 28,
    "action": "create",
    "resource": "wo",
    "asset": "11300",
    "params": {
        "site": "bedford",
        "description": "My Work Order"
    }
}
```

Outbound Messaging

Messages are to be transmitted in JSON format. All keys are specified in lower case. The basic structure is defined as per the following table

Key	Value Type	Sample Value	Description
id	Copied from inbound message		
action	Copied from inbound message		
asset	Copied from inbound message		
resource	Copied from inbound message		
success	boolean	true	Whether the action worked
message	string	"Invalid action"	A helpful message

pagination	object	`{}`
		OR
		`{`
		`"startAt": 0,`
		`"maxItems": 10,`
		`"total": 23`
		`}`
			Details about how much information has been returned

results	array	Results dependent on action and resource	

Site Lookup

For sites, each individual site contained within the *results* will conform to the following schema:

Key	Value Type	Description
site	String	Site Identifier
description	String	Description of the asset
extras	Object	A key value pairing of additional system specific fields eg. { "langcode": "EN", "contact": "SHANEP" }

Sample Response

```
{
  "id": 28,
  "action": "lookup",
  "resource": "site",
  "asset": "",
  "results": [{
    "site": "MCLEAN",
    "description": "McLean IT Operations Center",
    "extras": {
      "CONTACT": "SHANEP",
      "LANGCODE": "EN"
    }
  }],
  "success": true,
  "message": "",
  "pagination": {
    "startAt": 0,
    "maxItems": 20,
    "total": 20
  }
}
```

Asset Lookup

For assets, each individual asset contained within the *results* will conform to the following schema:

Key	Value Type	Description
asset	String	The asset's identifier. This should be used to satisfy various requests *asset* field
site	String	Where the asset is located
location	String	Where the asset is located
description	String	Description of the asset
type	String	The asset type. Null if unspecified
extras	Object	A key value pairing of additional system specific fields

		eg. { "status": "operating", "purchaseprice": 1225.0 }

Sample Response

```
{
  "id": 28,
  "action": "lookup",
  "resource": "asset",
  "success": true,
  "pagination": {
    "startAt": 0,
    "maxItems": 2,
    "total": 78
  },
  "message": "",
  "results": [{
    "asset": "11300",
    "site": "bedford",
    "location": "ENG-001",
    "type": "IT",
    "description": "A terminal",
    "extras": {
      "status": "operating",
      "purchaseprice": 1225.0
    }
  }, {
    "asset": "TR432",
    "site": " bedford",
    "type": "IT",
    "description": "A terminal",
    "extras": {
      "status": "offline",
      "purchaseprice": 1225.0
    }
  }]
}
```

Work Order Lookup

For work orders, each individual work order contained within the *results* will conform to the following schema:

Key	Value Type	Description
wo	String	The work order’s identifier
actualStartDate	Date (ISO Format)	Actual date at which the work started
actualFinishDate	Date (ISO Format)	Actual date at which the work finished
asset	String	The asset’s identifier. Should match overarching *asset*
assetName	String	The asset’s name
assetGroup	String	Unit or process to which the asset belongs
assignedTo	String	The individual assigned the work order
createDate	Date (ISO Format)	Date the work order was created
createTime	Time (ISO Format)	Time the work order was created
description	String	Description of the work order
discipline	String	Specific discipline of work being conducted
priority	String	Action priority
requestedBy	String	The individual requesting the work order
requestedTeam	String	Operational team of individual requesting work order
site	String	Unique site to which the asset belongs
status	String	The current status of the work order
statusDate	Date (ISO Format)	Date at which the current status changed
system	String	Process area to which the asset belongs
targetStartDate	Date (ISO Format)	Date at which the work should have started
targetFinishDate	Date (ISO Format)	Date at which the work should have finished
workType	String	Type of work being performed (values are system dependent)
extras	Object	A key value pairing of additional system specific fields eg. { "esthours": 4 }

Empty fields will contain an empty string.

Sample Response

```
{
  "id": 28,
  "action": "lookup",
  "resource": "wo",
  "success": true,
  "pagination": {
    "startAt": 0,
    "maxItems": 2,
    "total": 78
  },
  "message": "",
  "results": [{
    "wo": "1399",
    "actualStartDate": "2020-05-25",
    "actualFinishDate": "",
    "asset": "11300",
    "assetName": "A terminal",
    "assetGroup": "ENG-001",
    "assignedTo": "jbloggs",
    "createDate": "2020-05-01",
    "createTime": "20:03:28",
    "description": "Replace keyboard",
    "discipline": "Electrical",
    "priority": "P1",
    "requestedBy": "jsmith",
    "requestedTeam": "",
    "site": "bedford",
    "status": "WAPPR",
    "statusDate": "2020-05-01",
    "system": "",
    "targetStartDate": "2020-05-05",
    "targetFinishDate": "2020-05-15",
    "extras": {
      "esthours": 1
    }
  }, {
    "wo": "1401",
    "actualStartDate": "2020-05-25",
    "actualFinishDate": "",
    "asset": "TR431",
    "assetName": "A terminal",
    "assetGroup": "ENG-001",
    "assignedTo": "jbloggs",
    "createDate": "2020-05-01",
    "createTime": "20:03:28",
    "description": "Fix broken terminal",
    "discipline": "Electrical",
    "priority": "P1",
    "requestedBy": "jsmith",
    "requestedTeam": "",
    "site": "bedford",
    "status": "WAPPR",
    "statusDate": "2020-05-01",
    "system": "",
    "targetStartDate": "2020-05-05",
    "targetFinishDate": "2020-05-15",
    "extras": {
      "esthours": 5
    }
  }]
}
```

Work Order Creation

Details of the work order created will be returned in the same format as for orders looked up, except that there will only be a single work order.

```
{
  "id": 28,
  "action": "create",
  "resource": "wo",
  "success": true,
  "pagination": {},
  "message": "",
  "results": [{
    "wo": "1402",
    "actualStartDate": "2020-05-25",
    "actualFinishDate": "",
    "asset": "TR431",
    "assetName": "A terminal",
    "assetGroup": "ENG-001",
    "assignedTo": "jbloggs",
    "createDate": "2020-05-01",
    "createTime": "20:03:28",
    "description": "Fix broken terminal",
    "discipline": "Electrical",
    "priority": "P1",
    "requestedBy": "jsmith",
    "requestedTeam": "",
    "site": "bedford",
    "status": "WAPPR",
    "statusDate": "2020-05-01",
    "system": "",
    "targetStartDate": "2020-05-05",
    "targetFinishDate": "2020-05-15",
    "extras": {
      "esthours": 5
    }
  }]
}
```

Errors

No results will be returned.

Sample Response

```
{
  "id": 28,
  "action": "invalid",
  "resource": "wo",
  "asset": "11300",
  "success": false,
  "pagination": {}
  "message": "Invalid Action",
  "results": []
}
```

Filter

The filter splits the flow depending on whether the request contains errors or not.

Error Handling Converter

This converter runs through the range of potential errors and produces a useful validation error message for transmission back to the originating device through the device response channel and in the agreed message format.

Common Prep Converter

This converter performs a range of common actions related to the IBM Maximo interface. Such actions include:

- Defaulting pagination
- Cleansing parameters. Parameters passed in with an empty string are removed so that downstream IBM Maximo service plugins work correctly. Note – the IBM Maximo service plugin supports specifying all potentially relevant search fields and provided the field isn't present in the inbound message, they'll be disregarded.
- Requesting responses in compact form where possible, minimising data transmitted between Reekoh and IBM Maximo

Rules Engine

The Rules Engine is utilised as a switch to despatch requests down either the Site Lookup, Asset Lookup, Work Order Lookup or Work Order Creation workflows.

Site Lookup IBM Maximo Service

The Site Lookup IBM Maximo service creates a search to be executed against IBM Maximo to get all possible site information. This will hit a URL that's similar to:

> GET: http://BASE_URL/maxrest/rest/mbo/site/?

The data returned from this call is a list of IBM Maximo data objects.

The plugin is configured with a HTTP Exception Logger so that unhandled exceptions are caught and passed to the Exceptions HTTP Gateway.

Site Egress Converter

The Site Egress converter changes the results into the form specified for data output. Specific field mappings performed:

IBM Maximo Field	Work Order Result Field	Notes
SITEID	*site*	
DESCRIPTION	*description*	
Others	Wrapped into extras, with the key lowercased	

Any missing fields from the mapping are inserted as an empty string.

Asset Lookup IBM Maximo Service

The Asset Lookup IBM Maximo service takes the original parameters and creates a search to be executed against IBM Maximo for either a specific asset (form the Device ID) or based on other criteria. This will be hitting a URL that's similar to:

GET: http://BASE_URL/maxrest/rest/mbo/asset/?

To support both asset number exact matching and asset prefix searching, two separate search fields are used referencing the IBM Maximo *assetnum* field, one of which performs exact matching (with the ~eq~ operator). The other uses the ~sw~ operator to support prefix based searching.

The data returned from this call is a list of IBM Maximo data objects.

The plugin is configured with a HTTP Exception Logger so that unhandled exceptions are caught and passed to the Exceptions HTTP Gateway.

Asset Egress Converter

The Asset Egress converter changes the results into the form specified for data output. Specific field mappings performed:

IBM Maximo Field	Work Order Result Field	Notes
SITEID	*site*	
DESCRIPTION	*description*	
ASSETNUM	*asset*	
LOCATION	*location*	

ASSETTYPE	*type*	
Others	Wrapped into extras, with the key lowercased	

Any missing fields from the mapping are inserted as an empty string.

Work Order Creation Converter

The Work Order Creation converter supplies default values for Work Order Creation and can be removed in if those defaults are not desired.

Work Order Creation IBM Maximo Service

The Work Order Creation IBM Maximo service takes the data encoded in the original parameters section and creates a new work order in IBM Maximo. Note – the asset number will be aligned to the original device ID. This will be hitting a URL that's similar to:

POST: http://BASE_URL/maxrest/rest/mbo/workorder

The data that's returned from this call is the IBM Maximo data object, as may be contained in a lookup. The plugin is configured with a HTTP Exception Logger so that unhandled exceptions are caught and passed to the Exceptions HTTP Gateway.

Work Order Lookup Converter

The lookup converter supplies default values and performs any changes to the payload format for looking up a Work Order in IBM Maximo.

Some specific actions performed:

- IBM Maximo supports searching for a *status* match (through the ~eq~ operator) or a lack of a match (through the ~neq~ operator). When we want to search for OPEN work orders (implemented as 'not CLOSED'), the converter removes the *status* field from *params*, so that we don't search for a match and create a transitory field called *negstatus*.
- Gives preference to the use of *start/endCreateDate* over *max/minOffset*.
- Gives preference to the use of *start/endStatusDate* over *max/minStatusOffset*.
- Translates dates/offsets to the Date Time fields required by IBM Maximo
- Configures preferred result order (by *statusDate* descending)

Work Order Lookup IBM Maximo Service

The Work Order Lookup IBM Maximo service takes the original parameters and creates a search to be executed against IBM Maximo for that specific asset (from the device ID). This will be hitting a URL that's similar to:

GET: http://BASE_URL/maxrest/rest/mbo/workorder/?

Closed Work Orders use the search string: *status=~eq~CLOSE*. Open Work Orders use the search string: *status=~neq~CLOSE*. To support both operators, different search parameters are configured (*status* and *negstatus*).

The data returned from this call is a list of IBM Maximo data objects.

The plugin is configured with a HTTP Exception Logger so that unhandled exceptions are caught and passed to the Exceptions HTTP Gateway.

Work Order Egress Converter

The Work Order Egress converter changes the results into the form specified for data output. It normalises the single instance returned from the work order creation with the multiple instances returned from the work order lookup and then performs data transformation. Specific field mappings performed:

IBM Maximo Field	Work Order Result Field	Notes
WONUM	*wo*	
SITEID	*site*	
REPORTDATE	*createDate*	Only the date component is extracted
REPORTDATE	*createTime*	Only the time component is extracted
DESCRIPTION	*description*	
STATUS	*status*	
WORKTYPE	*workType*	
REPORTEDBY	*requestedBy*	
WOPRIORITY	*priority*	
ASSETNUM	*asset*	
LOCATION	*assetGroup*	
STATUSDATE	*statusDate*	Only the date component is extracted
TARGETSTARTDATE	*targetStartDate*	Only the date component is extracted
TARGETCOMPDATE	*targetFinishDate*	Only the date component is extracted
SCHEDSTART	*actualStartDate*	Only the date component is extracted
SCHEDFINISH	*actualFinishDate*	Only the date component is extracted
Others	Wrapped into extras, with the key lowercased	

Any missing fields from the mapping are inserted as an empty string.

MQTT Connector

This connector is configured to point back to the MQTT Gateway at the data entry but targeting the data egress MQTT topics.

HTTP Gateway

Throughout the pipeline, the IBM Maximo Service plugins have been configured with HTTP Exception Loggers so that any unexpected communication errors won't be just logged to the console but can be captured and fed back to the external application. This gateway plugin receives those inbound messages.

Exception Handling Converter

This plugin receives the exception message and converts it into the form specified for data egress. It also specifies that the resulting exception should be transmitted back to the client on the data/out/errors topic.