Project Objectives

Use the JMSG_DEMO app to learn how Basecamp's JSON Message(JMSG) Facility is used to communicate between the Software Bus and either a UDP or MQTT communications link. JMSG_DEMO serves an example for how developers can create an app that uses the Facility for external communications. User apps define SB messages as normal and also two small functions that assist with the translation between command/telemetry SB binary messages and JSON text messages.

These slides are used in the Space Step's JSON Message (JMSG) Demo Project at

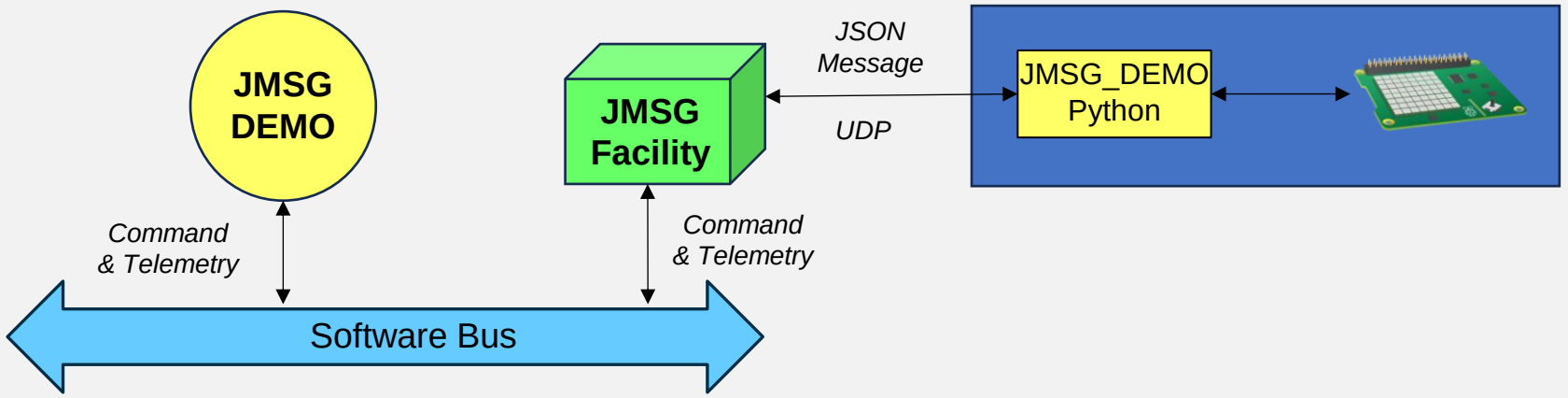
https://spacesteps.c[illegible]/basecamp-jmsg-demo/

JSON Message (JMSG) Communication Motivation

- cFS Basecamp educational projects often require communicating with local hardware components and with remote command and telemetry systems
- Creating external cFS interfaces can often be code intensive and time consuming which can detract from the purpose of the educational project unless the project is about the interface itself
- Basecamp's app store includes a suite of libraries and apps that work together called the ***JMSG Facility*** that simplifies communicating with systems external to the cFS target
- Basecamp's solution must be orders of magnitude simpler than implementing project-specific communications solutions otherwise complexity has merely shifted from developing a custom solution to developing with Basecamp's JMSG Facility
- The next two slides describe use cases that drove the development of the JMSG Facility

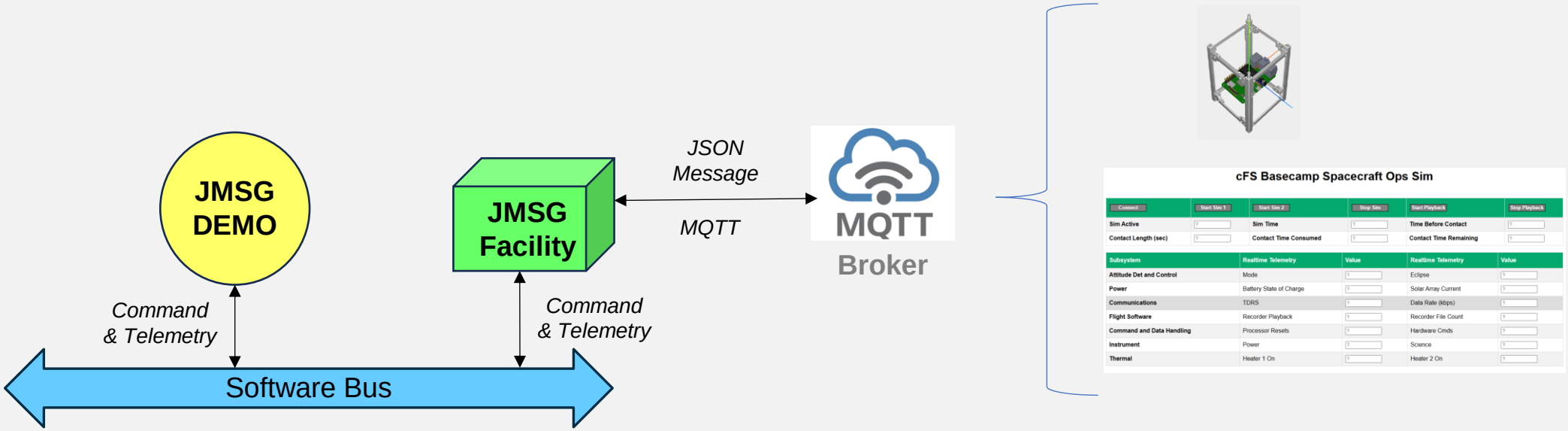
Use Case 1: Interfacing with Python

(Raspberry Pi Hardware Interface)



- In general, this is an interface to execute Python scripts. Currently, the most common use case is on a Raspberry Pi (RPI) to interface with hardware addon components
- RPI Python libraires simplify the task of interfacing with hardware components. Also, the libraries are updated with new RPI processor & hardware addon versions simplifying maintenance
- The JMSG Facility simplifies the task of creating apps that communicate with RPI hardware components
- The JMSG_DEMO app and Python script serves as examples that can be modified for your needs

Use Case 2: Remote MQTT Communications



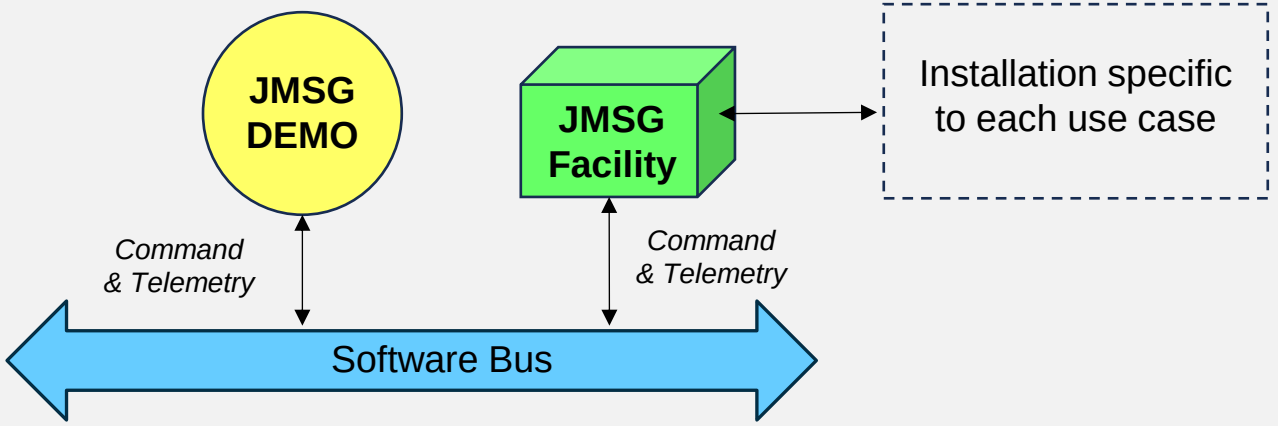
- MQTT is the de facto standard for the Internet of Things (IoT) communications
- Providing a bridge between the Software Bus and an MQTT Broker opens up a wide range of possibilities

cFS Target Installation

cFS Target Installation

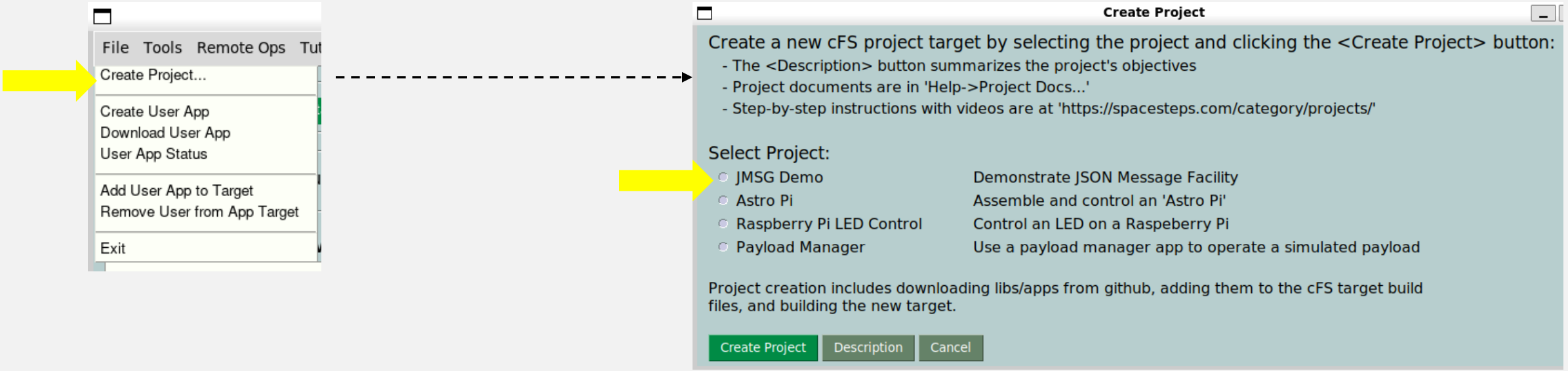
- The section describes how to install the JMSG Demo project cFS target and what configuration tables need to be tuned for each use case scenario
  - It does not cover installing and running software external to the cFS target. These topics are covered in each use case instructions

- JMSG communication concepts are introduced to provide context for understanding how the configuration tables are used

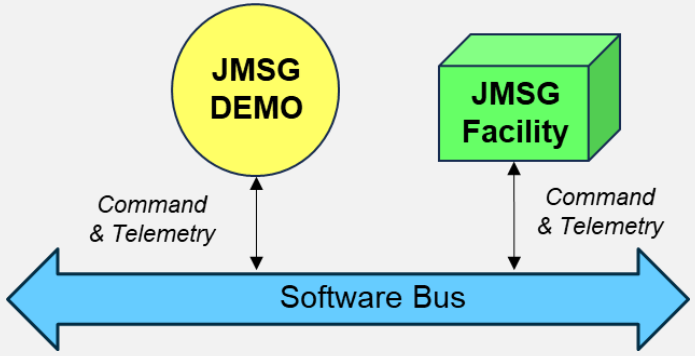


cFS Target Installation

- Create JMSG Demo project using the Create Project tool



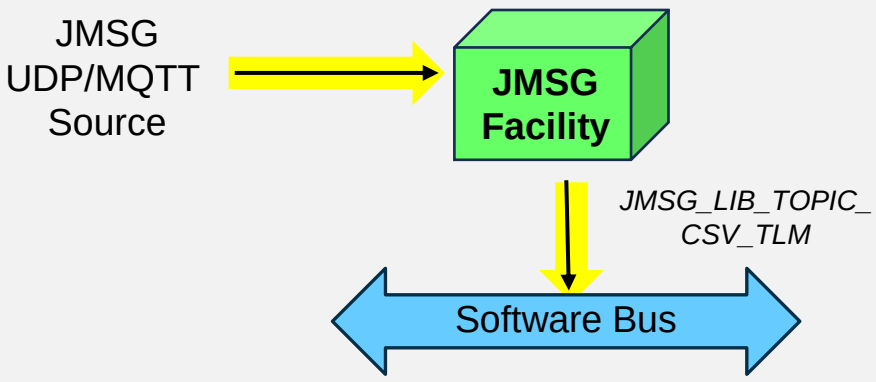
- This creates a new cFS target that contains the JMSG demo app and the JMSG Facility library and apps



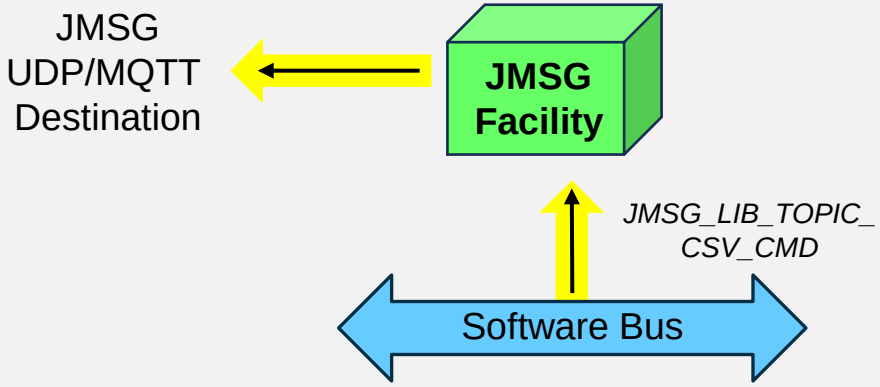
JMSG Communication Terminology

- Messages are identified by *Topic* identifiers
  - cFS binary messages and their topic identifiers are defined in Electronic Data Sheets (EDS)
  - JMSGs use “topic names” as identifiers that follow the MQTT “topic” standard (details on later slide)
- The JMSG Facility translate between text JMSGs and binary Software Bus messages
- JMSG Facility topic translation roles are defined from the SB perspective
  - *Publish:* Ingest JMSGs and translate them to binary messages thar are published on the SB
  - *Subscribe:* Subscribe to and ingest SB messages and translate them to JMSGs that are sent over UDP or MQTT
- Typically, commands are defined with a publish role and telemetry with a subscribe role with the
  - The JMSG_LIB EDS topic identifier names are shown below

Publish Topic



Subscribe Topic



JMSG Configuration Tables[1]

JMSG Topic Table
- cpu1_jmsg_topics.json
- Define the topics that are recognized and processed by the JMSG Facility

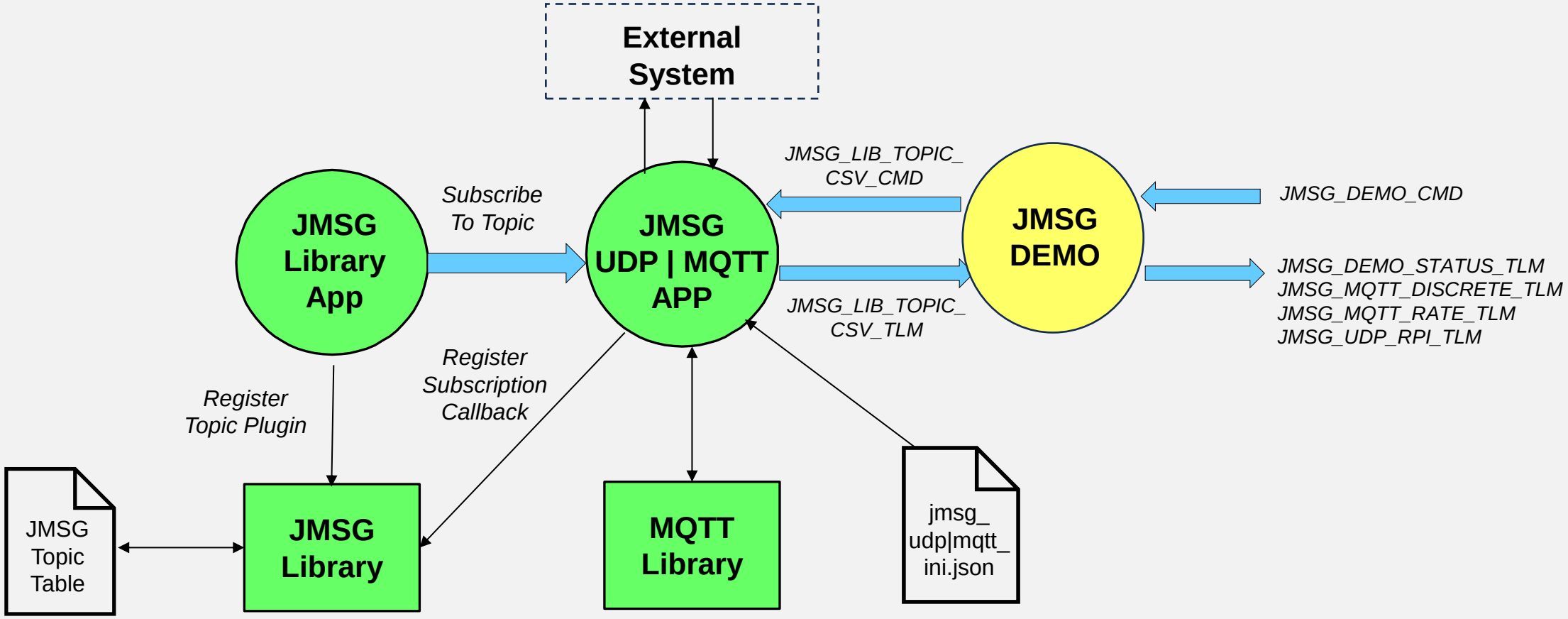
JMSG UDP App Initialization Table
- cpu1_jmsg_udp_ini.json
- Defines UDP IP address and ports

JMSG MQTT App Initialization Table
- cpu1_jmsg_mqtt_ini.json
- Defines MQTT broker URL and port
- Defines MQTT performance parameters, defaults suitable for demos but they may need tuning for projects

The table configuration details are provided after the next few slides describe the JMSG Facility components

1. Tables are located in cfs-basecamp/cfe-eds-framework/basecamp_defs/

JMSG Facility Data Flow

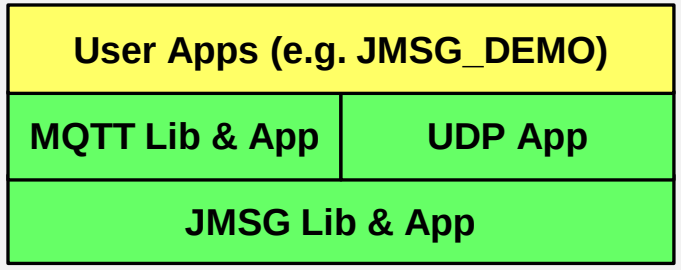


JMSG Facility Data Flow

- **The detailed data flow is not required to do the exercises in this project; however, a deeper understanding allows the following:**

  - ✓ Understand the purpose of table configurations and which components own the tables
  - ✓ Monitor the message flows through the system which helps with understanding the system and troubleshooting

- **The JMSG Facility System function in three layers described on the next slides**



JMSG Library and App

JMSG Library
- cpu1_jmsg_topics.json
- Define the topics that are recognized and processed by the JMSG Facility

JMSG Library App
- cpu1_jmsg_udp_ini.json
- Defines UDP IP address and ports

JMSG MQTT Library
- cpu1_jmsg_udp_ini.json

JMSG MQTT APP
- cpu1_jmsg_mqtt_ini.json
- Defines MQTT broker URL and port
- Defines MQTT performance parameters, defaults suitable for demos but they may need tuning for projects

JMSG UDP APP
- cpu1_jmsg_mqtt_ini.json

1. Tables are located in cfs-basecamp/cfe-eds-framework/basecamp_defs/

JMSG MQTT Library and App

JMSG Library
- cpu1_jmsg_topics.json
- Define the topics that are recognized and processed by the JMSG Facility

JMSG Library App
- cpu1_jmsg_udp_ini.json
- Defines UDP IP address and ports

JMSG MQTT Library
- cpu1_jmsg_udp_ini.json

JMSG MQTT APP
- cpu1_jmsg_mqtt_ini.json
- Defines MQTT broker URL and port
- Defines MQTT performance parameters, defaults suitable for demos but they may need tuning for projects

JMSG UDP APP
- cpu1_jmsg_mqtt_ini.json

1. Tables are located in cfs-basecamp/cfe-eds-framework/basecamp_defs/

JMSG UDP App

JMSG Library
- cpu1_jmsg_topics.json
- Define the topics that are recognized and processed by the JMSG Facility

JMSG Library App
- cpu1_jmsg_udp_ini.json
- Defines UDP IP address and ports

JMSG MQTT Library
- cpu1_jmsg_udp_ini.json

JMSG MQTT APP
- cpu1_jmsg_mqtt_ini.json
- Defines MQTT broker URL and port
- Defines MQTT performance parameters, defaults suitable for demos but they may need tuning for projects

JMSG UDP APP
- cpu1_jmsg_mqtt_ini.json

1. Tables are located in cfs-basecamp/cfe-eds-framework/basecamp_defs/

JMSG DEMO App

JMSG Library
- cpu1_jmsg_topics.json
- Define the topics that are recognized and processed by the JMSG Facility

JMSG Library App
- cpu1_jmsg_udp_ini.json
- Defines UDP IP address and ports

JMSG MQTT Library
- cpu1_jmsg_udp_ini.json

JMSG MQTT APP
- cpu1_jmsg_mqtt_ini.json
- Defines MQTT broker URL and port
- Defines MQTT performance parameters, defaults suitable for demos but they may need tuning for projects

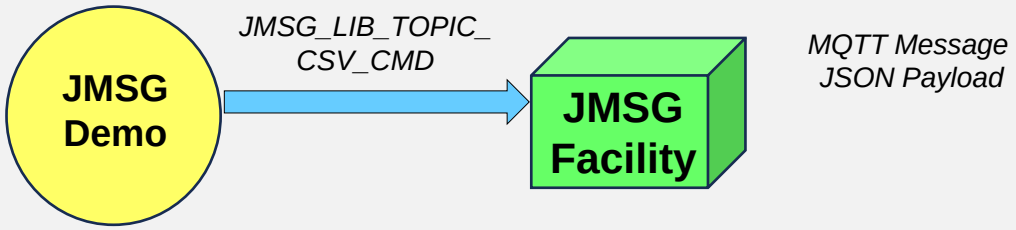
JMSG UDP APP
- cpu1_jmsg_mqtt_ini.json

1. Tables are located in cfs-basecamp/cfe-eds-framework/basecamp_defs/

JMSG Topic Table

Topic Table

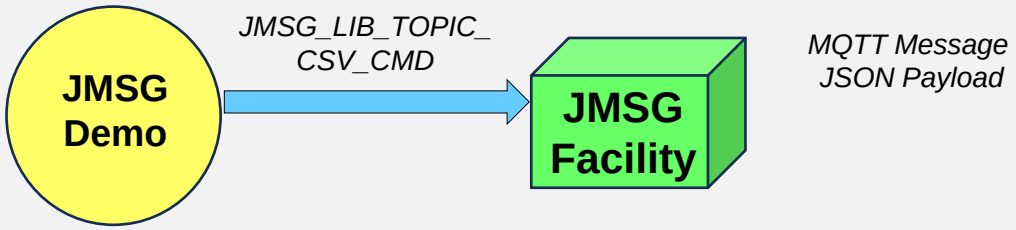
- Define the topics that are recognized and processed by the JMSG Facility



JMSG_UDP Initialization Table

Topic Table

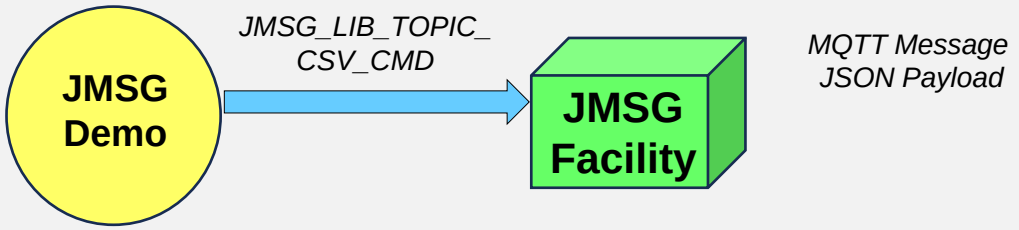
- Define the topics that are recognized and processed by the JMSG Facility



JMSG_MQTT Initialization Table

Topic Table

- Define the topics that are recognized and processed by the JMSG Facility



JMSG Message Definitions

Communications Mechanism	Messages	Message Definition
Software Bus	JMSG_LIB_TOPIC_SCRIPT_CMD_TOPICID JMSG_LIB_TOPIC_CSV_CMD_TOPICID[1] JMSG_LIB_TOPIC_CSV_TLM_TOPICID[1]	jmsg_lib.xml EDS definitions
UDP	"topic name"[2], JSON Payload	jmsg_topic_csv_cmd.c JSON Payload matches EDS definitions[1]
MQTT	"topic name" [2], JSON Payload	jmsg_topic_csv_tlm.c JSON Payload matches EDS definitions[1]

1. Comma Separated Variables (CSV) strings are used for command and telemetry parameters
2. Topic name follows the MQTT "topic" standard which is a string used to identify and route messages
   - Topics consists of one or more levels separated by a forward slash
   - For example, myhome/roundfloor/livingroom/temperature
   - Basecamp's UDP topic name appends a colon to the MQTT topic

Interfacing with Python

Raspberry Pi Hardware Interface

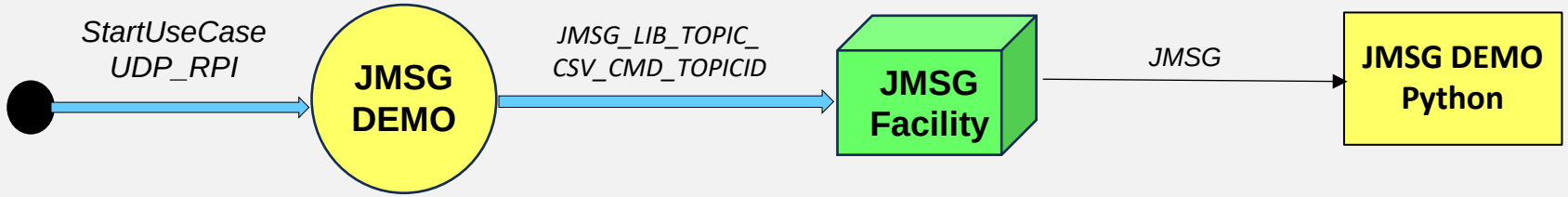
Start JMSG Demo Python

2. Start the cFS

Remote MQTT Communications

Connecting the cFS to the Internet of Things (IoT)

Built-in Command and Telemetry Plugins



basecamp/csv/cmd:{"name": "UDP RPI", "parameters": "rate-x,0.17453,rate-y,0.00000,rate-z,0.00000,lux,8"}

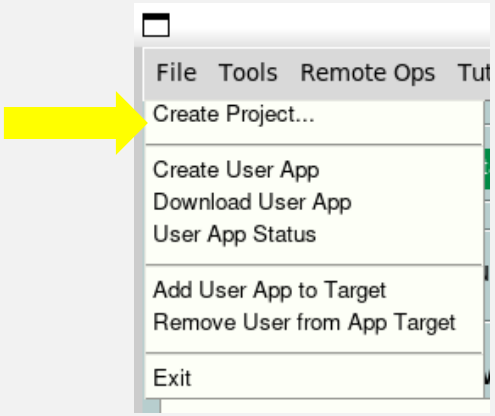
Start Event

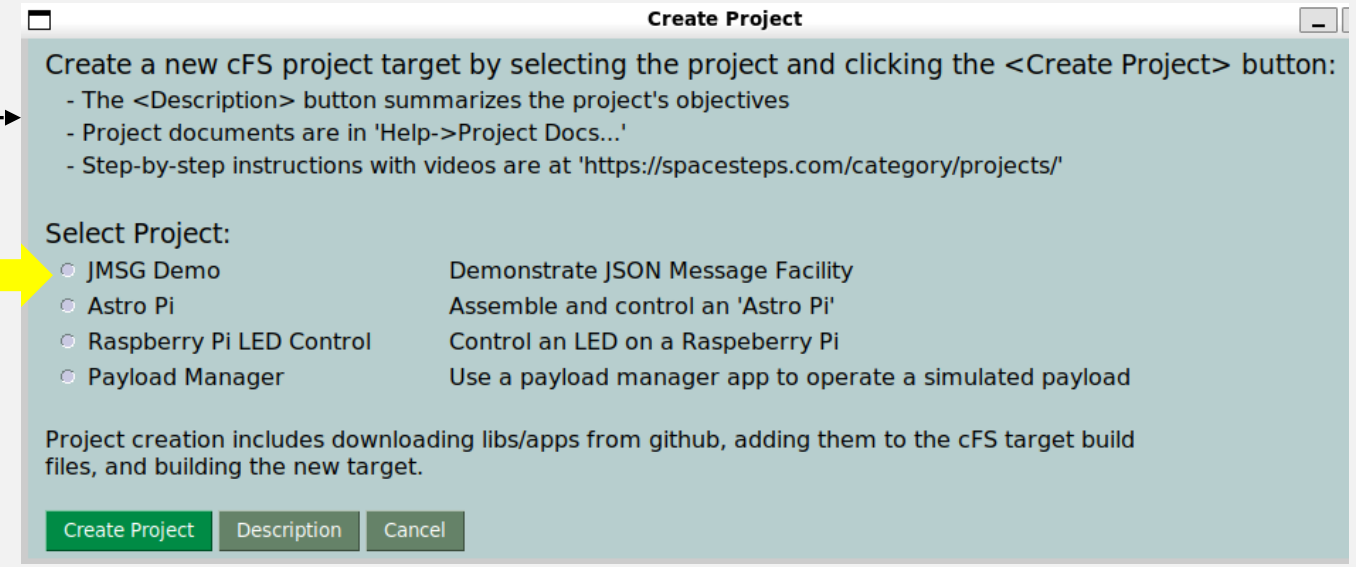
Software Bus Message

Internet

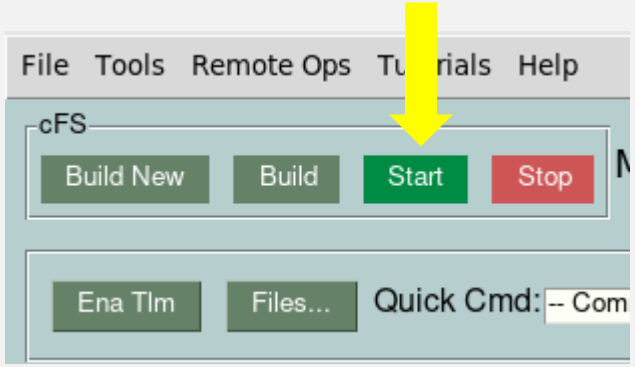
Python Script

1. Create JMSG Demo project using the Create Project tool

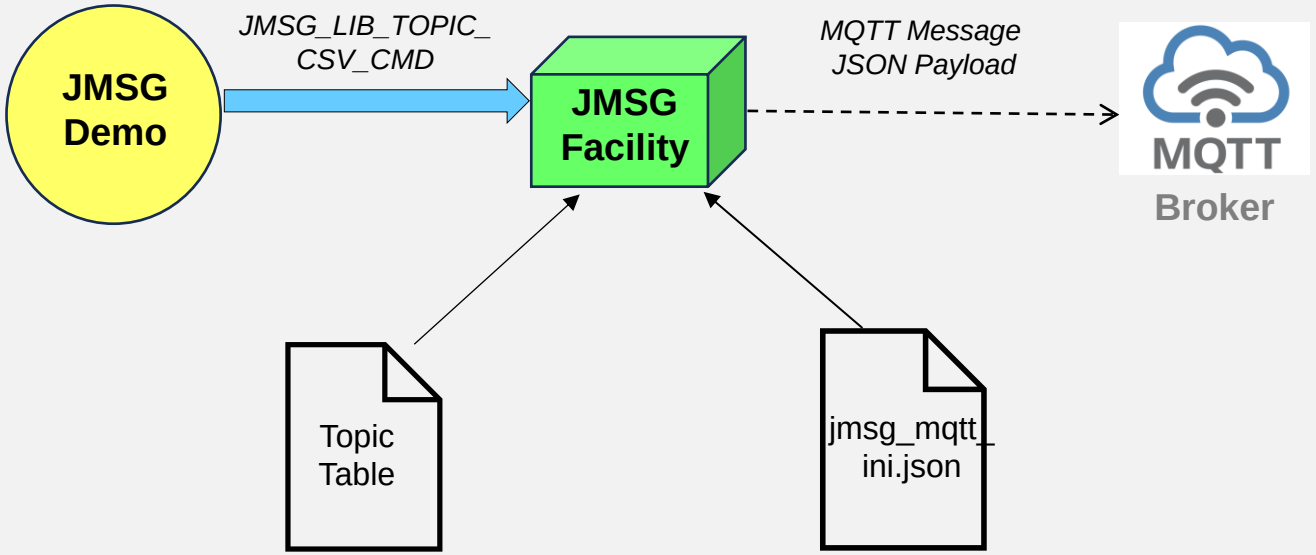




2. Start the cFS



Use Case: Send MQTT Discrete/Rate

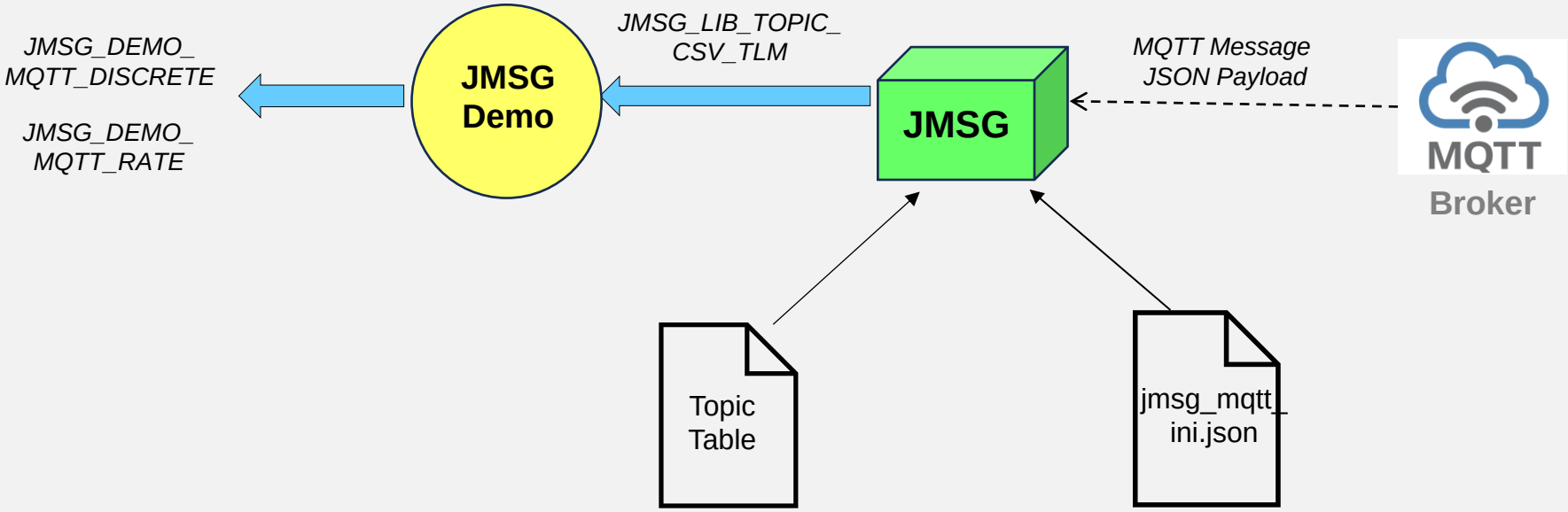


JMSG Represents a combination of the JSMG library and JMSG apps

Use Case: Control Remote Target

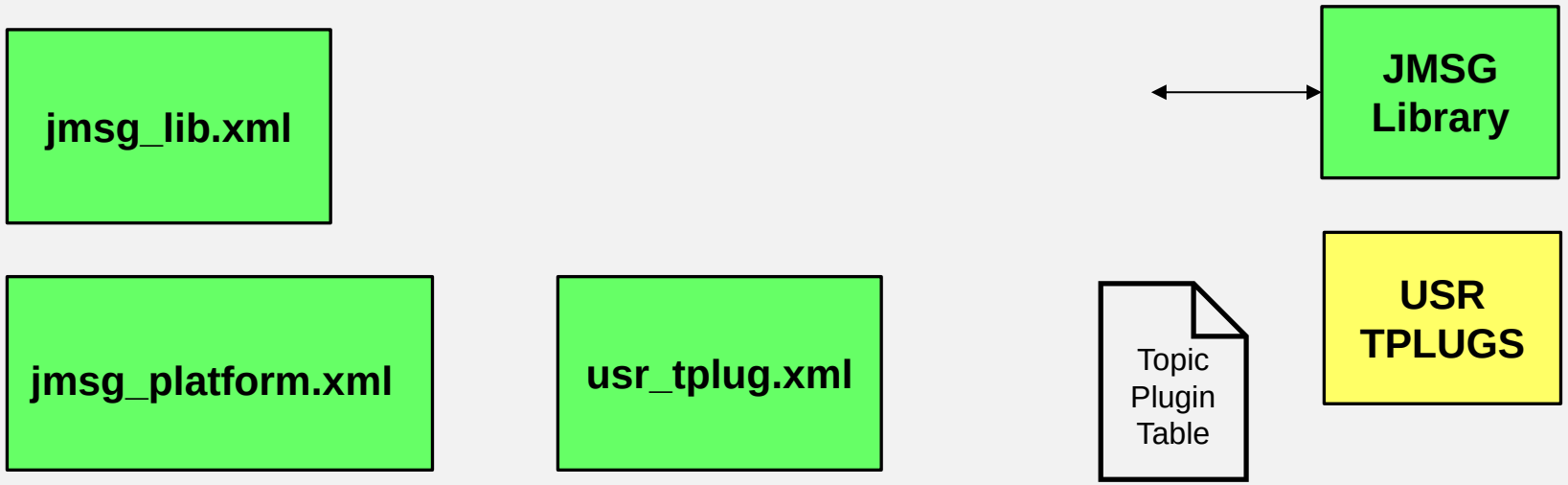
- Controlling remote cFS targets is often helpful in demonstrations and educational environments. MQTT is an internet communication standard that supports JSON message payloads. Free MQTT broker servers are available so no-cost solutions can be created.

Use Case: Receive MQTT Discrete/Rate



JMSG Represents a combination of the JSMG library and JMSG apps

JMSG EDS Definitions



JMSG Architectural Components

JMSG_LIB

Plugins data visibility trade. Keep adding plugins very simple and data should not need to be telemetered. This should help tooling as well.

TODO: Add circular dependency checks. E.g. JMSG_APP requires MQTT&UDP command topic IDs

The architecture intentionally decouples the network protocol from the plugin message format

The callback structure requires new plugins to be compiled into the library. A message-based design would allow

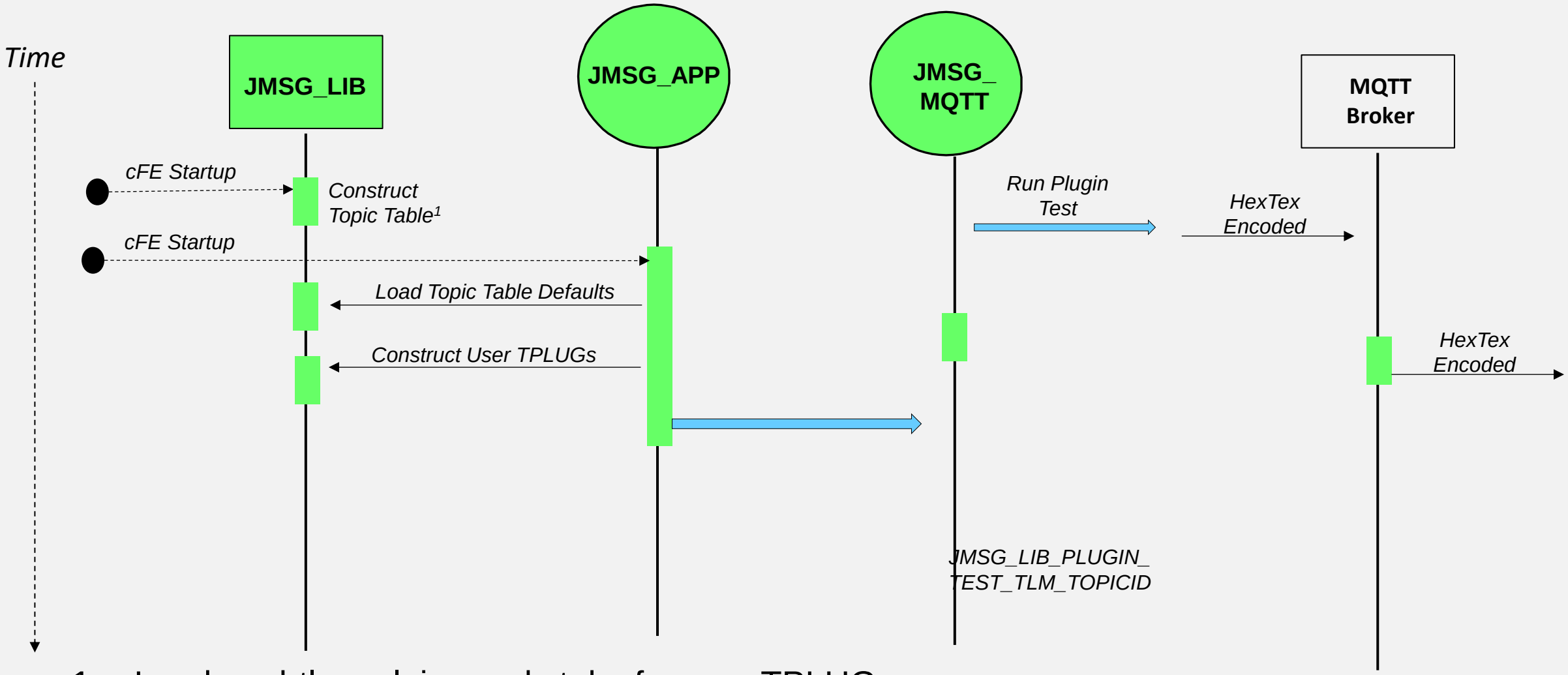
Debug:

If see stub function event then User IDs of constructed TPLUG does not agree with topic table entry

JSON format differences

Publish can happen without the correct plugin as long as the network protocol is the same.

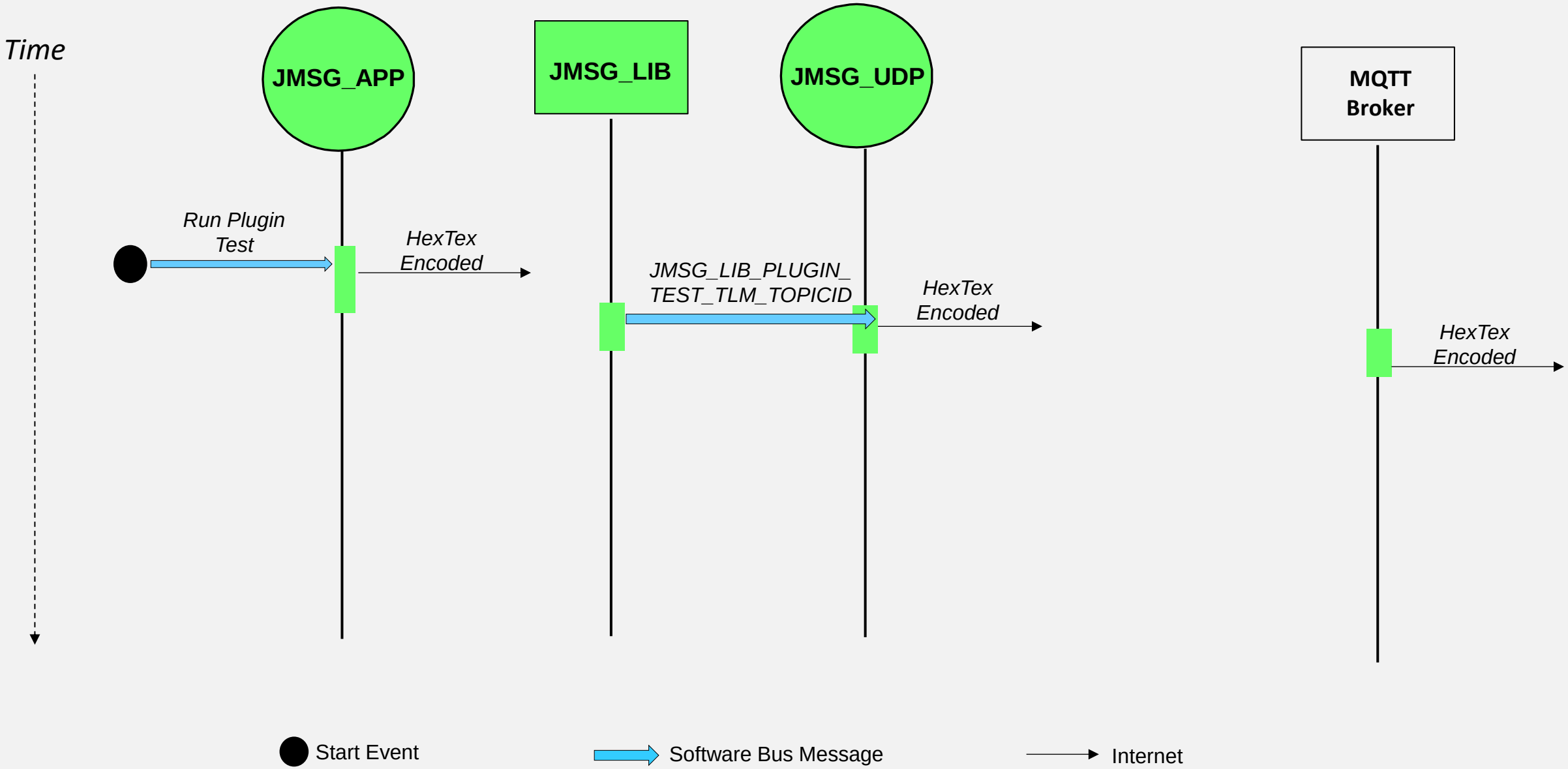
Topic Plugin Construction & Subscriptions



1. Load cmd-tlm pulgins and stubs for user TPLUGs
2.

Start Event Software Bus Message Internet

Built-in Command and Telemetry Plugins



JMSG Library & App Overview

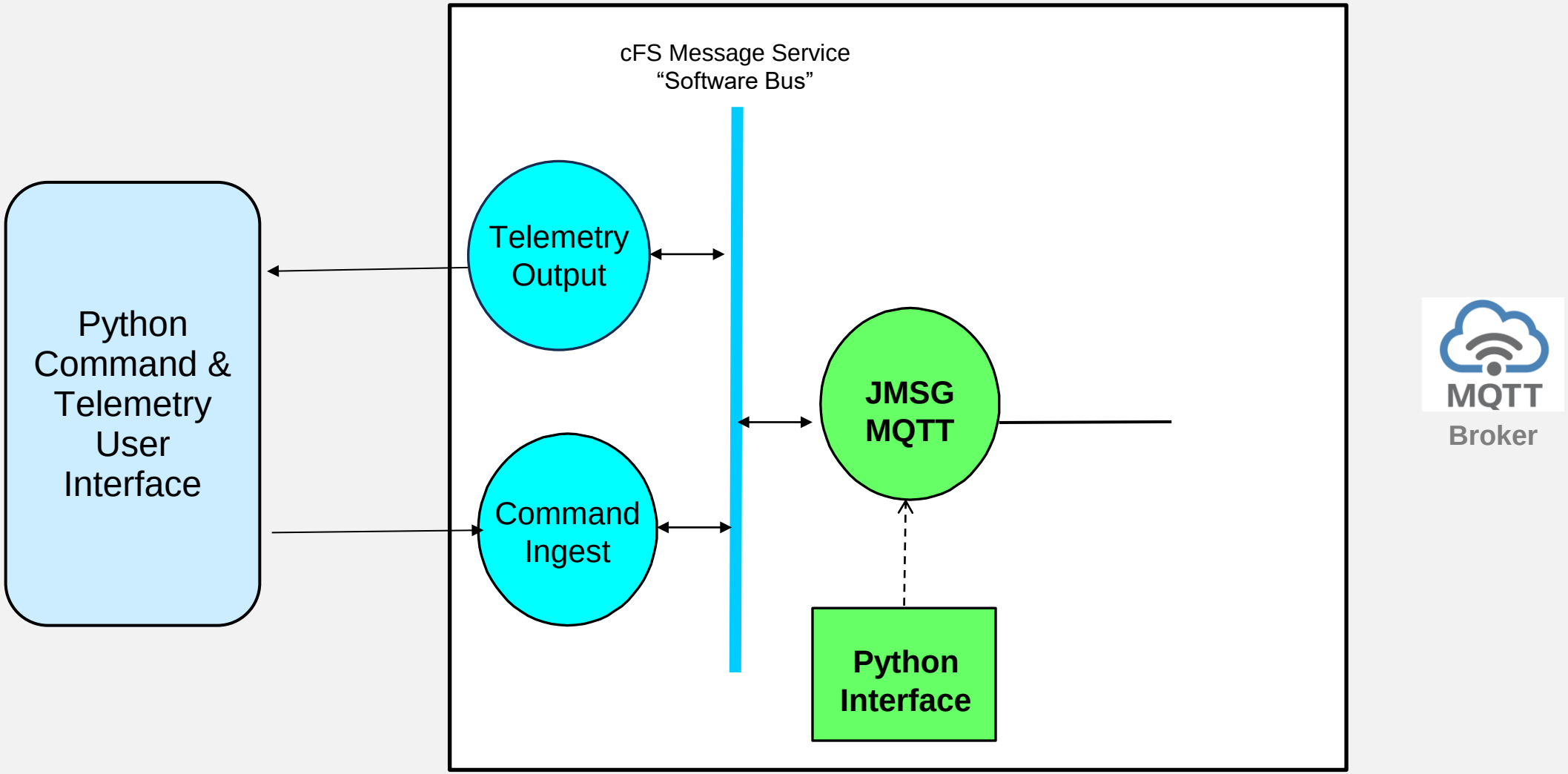
- Library
  - Create Topic Table
  - Provides API to manage the table
  - Defines
- App

- Commands
- Telemetry

Local JMSG Interface

Built-in Command and Telemetry Plugins

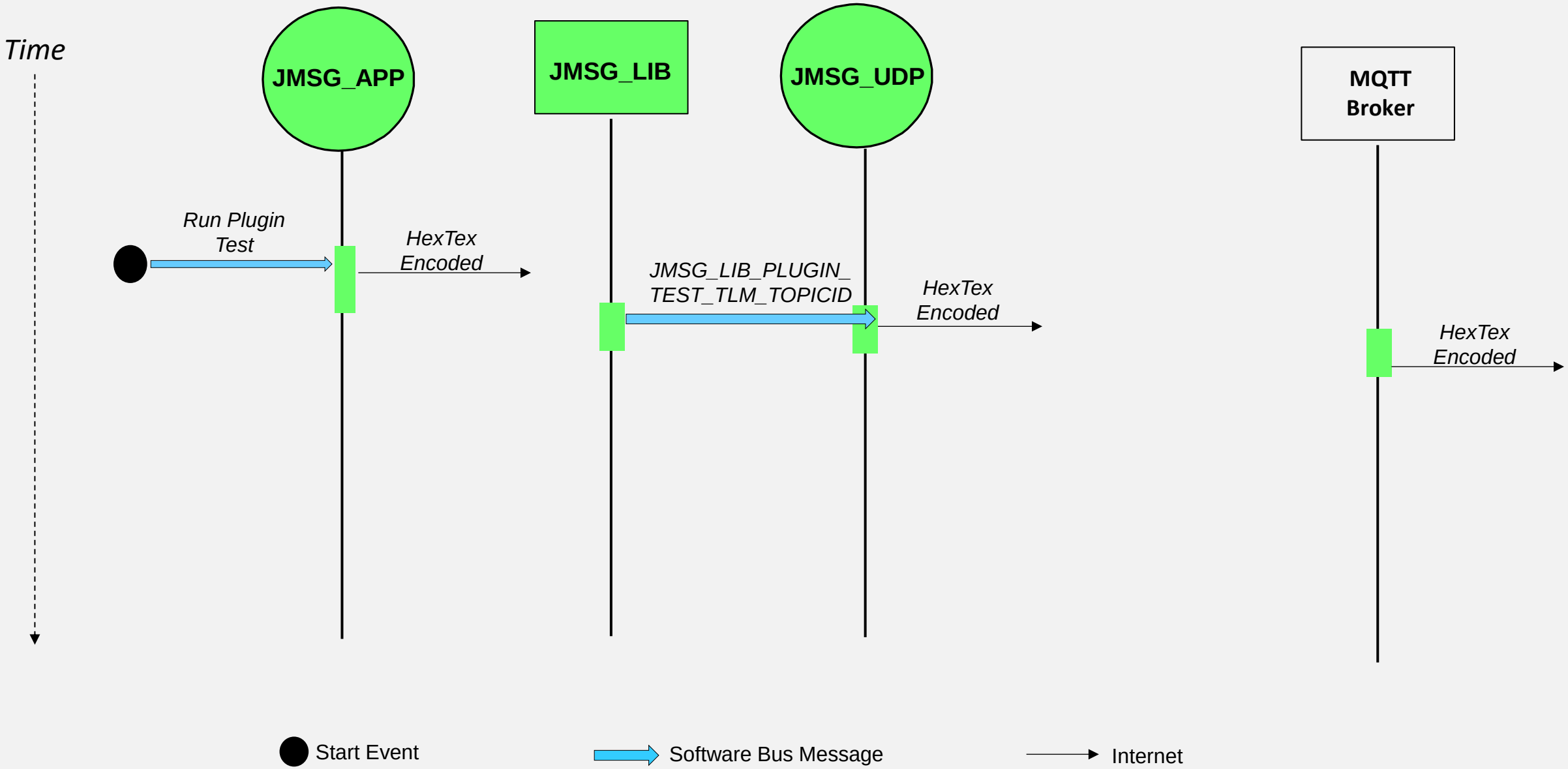
Space Steps MQTT Rate Demo



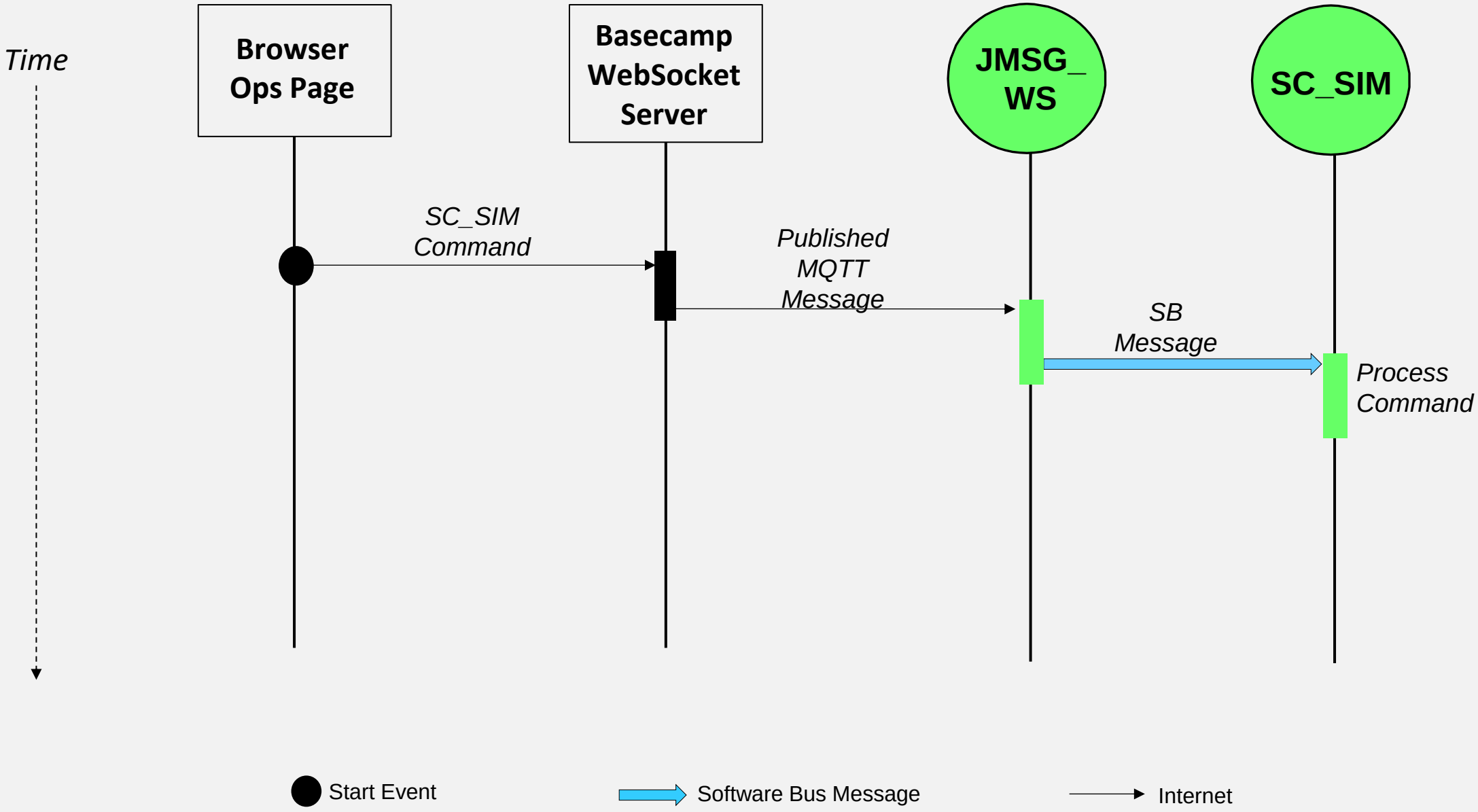
Space Steps MQTT Rate Demo

1. JMSG_MQTT: jmsg_mqtt_ini.json
   - "MQTT_BROKER_PORT": 1883
   - "MQTT_BROKER_ADDRESS": "broker.emqx.io",
2. JMSG_LIB: jmsg_topics.json
   - "TPLUG_WEB_RATE_PLUGIN_TLM_TOPICID": "cfe-4",
   - "name": "basecamp/demo/sc/rate“
   - "cfe-4": 0
   - “cfe": 0
   - "protocol": "mqtt“
   - "sb-role": “pub“
   - "enabled": “true“

Built-in Command and Telemetry Plugins



Spacecraft Ops: Commands



Remote JMSG Interface