CubeSat Flight Software Workshop

Ground Data System (GDS) Ground Support Equipment (GSE)

Leonard J. Reder
reder@jpl.nasa.gov
5 June 2019

Jet Propulsion Laboratory
California Institute of Technology

Ground Data System (GDS)/Ground Support Equipment (GSE) Fundamental Concept

- ***GDS – "Is an ops sort of thing..."***

RF Command & Telemetry Data Channel

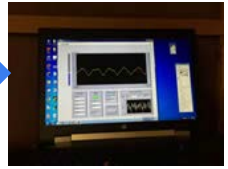
User

GDS

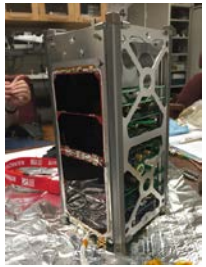
Command & Telemetry Processing/Storage/Visualization

Radio

- ***GSE – "Is a lab sort of thing…"***

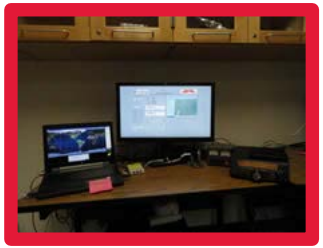


Simulations (Solar/Battery/Load)



CSUNSat1 with JPL developed battery

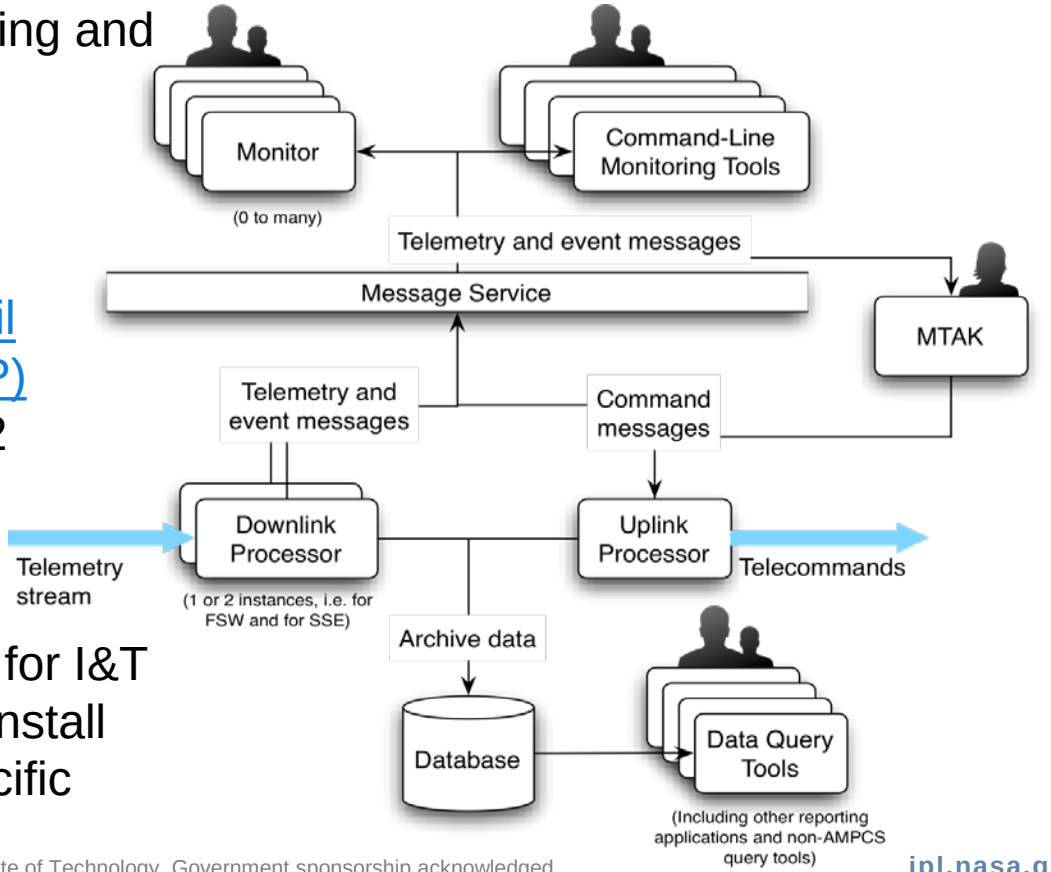
GDS



Command & Telemetry Processing/Storage/Visualization

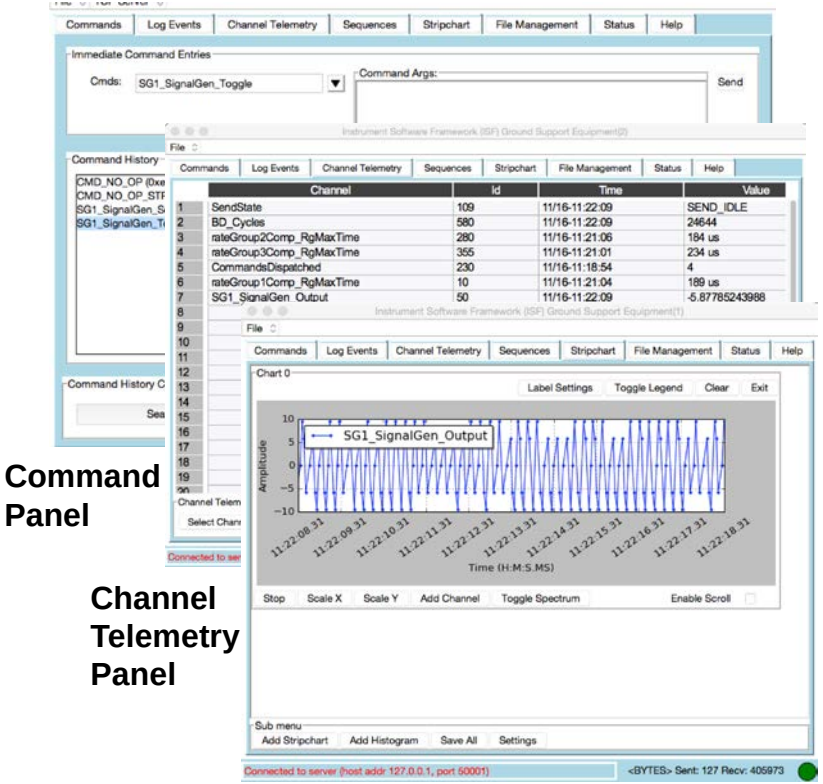
Enterprise Ground Data System (JPL AMPCS)

- AMMOS Mission Data Processing and Control System (AMPCS)
- Architecture Figure from "Cost-Effective Telemetry and Command Ground Systems Automation Strategy for the Soil Moisture Active Passive (SMAP) Mission", Josh Choi, AIAA 2012
- Key thoughts:
  - Based on software bus
  - Uses a full up database
  - MTAK is a Python adapter for I&T
  - Involved to configure and install
    - Requires mission specific adaptations



F’ Ground Support Equipment (GSE) Capability

- ***Out-of-the-box ready to use on Linux, Mac OSX, or Windows without any mission specific tailoring.***
- Enables integration testing and quick-look telemetry monitoring
  - ✓ ***Integration test API and logging***
  - ✓ Immediate commanding
  - ✓ Event and telemetry tables
  - ✓ Sequence assembly and execution
  - ✓ File uplink/downlink
  - ✓ Stripcharts and histograms
- Lightweight portable ground support system that is ***not an Enterprise Class GDS*** such as AMPCS

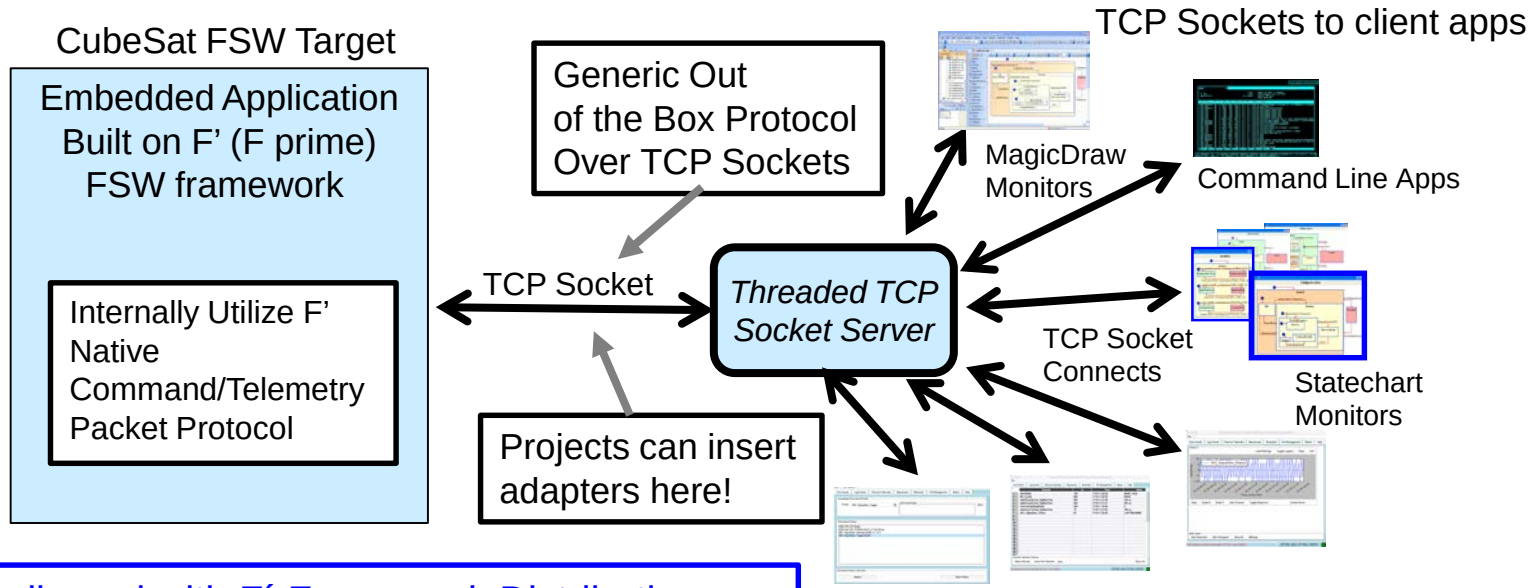


Command Panel

Channel Telemetry Panel

Plotting (Stripcharts, Histograms, etc.) Panel

F’ Ground Support Equipment (GSE) Architecture

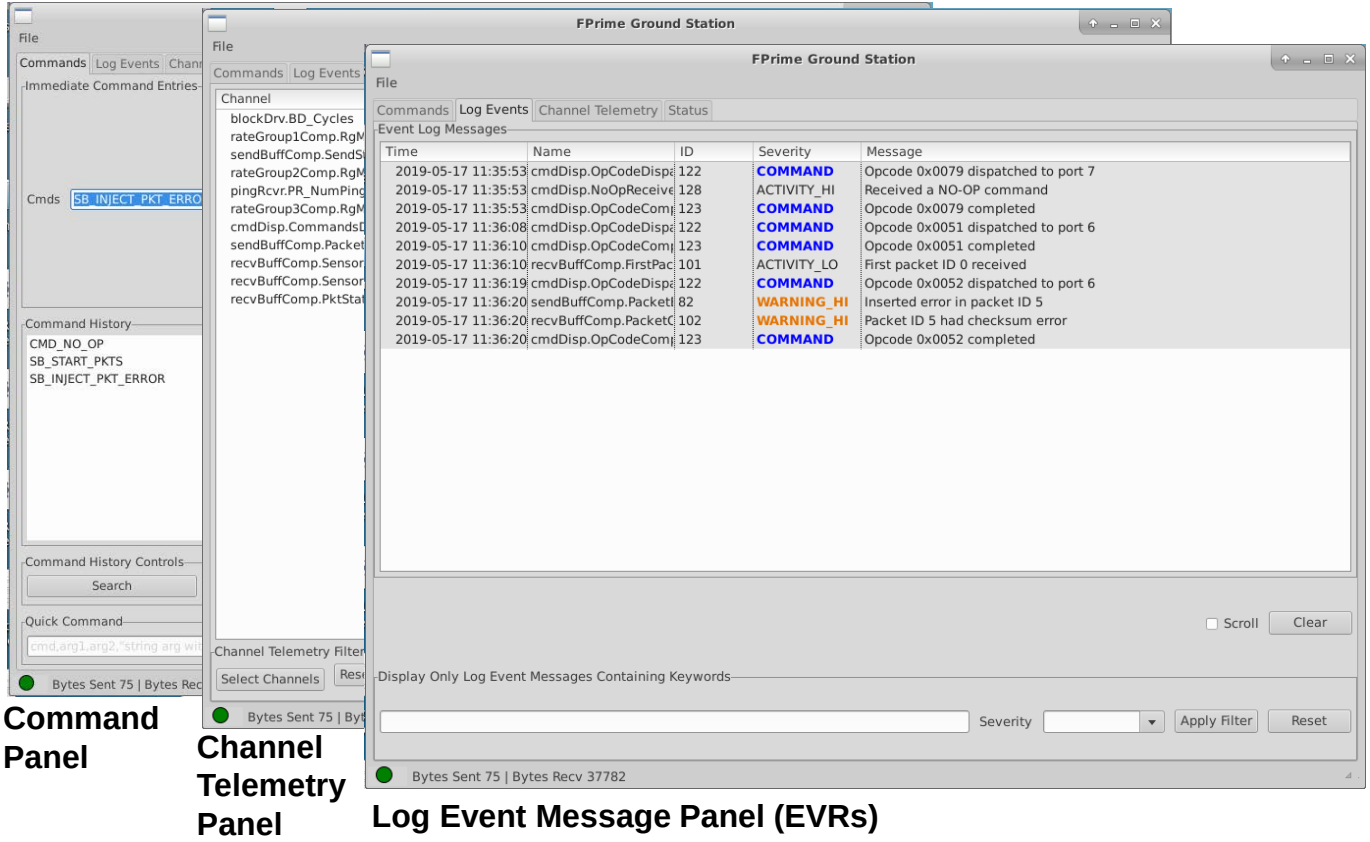


Delivered with F´ Framework Distribution:

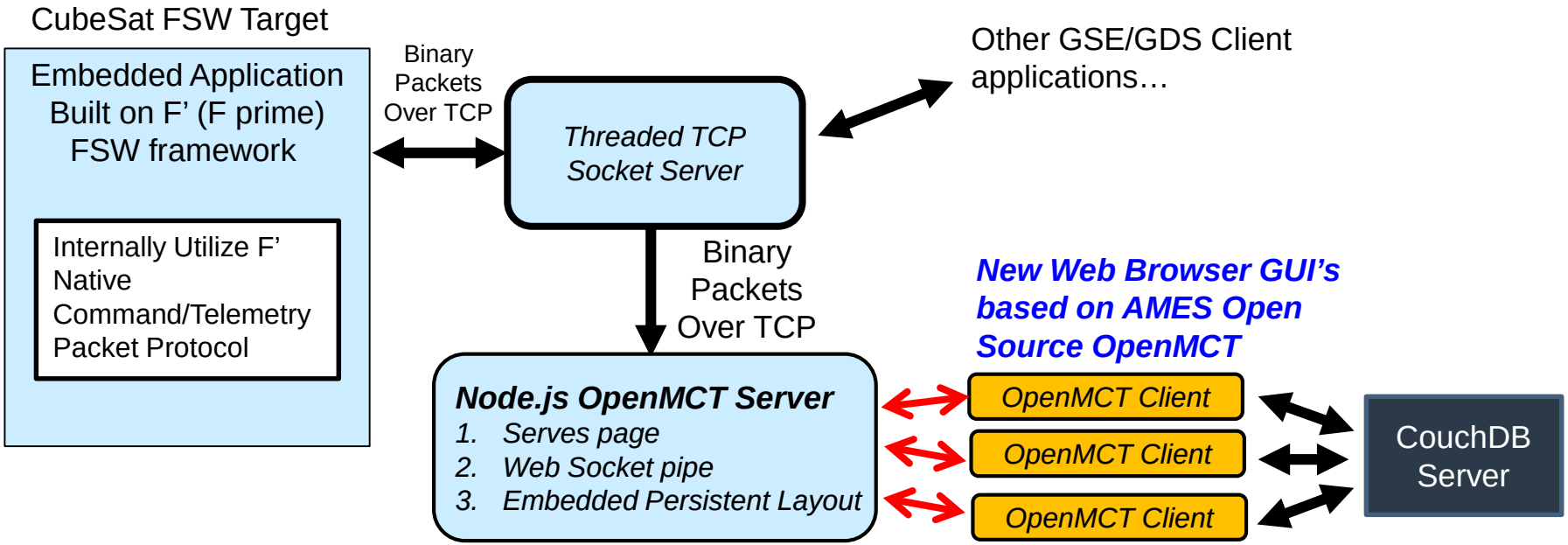
- Threaded TCP Socket Server
- gse.py Graphical User Interface
- Integration Test Framework API
- Example command line scripts built on API

New F’ Ground Data System (GDS) Client GUI

- Look is identical to legacy GSE Client GUI
- Built on WxPython for improved widgets
- Stripcharting & Histograms to be added
- Replaces GSE Client GUI for Python 3



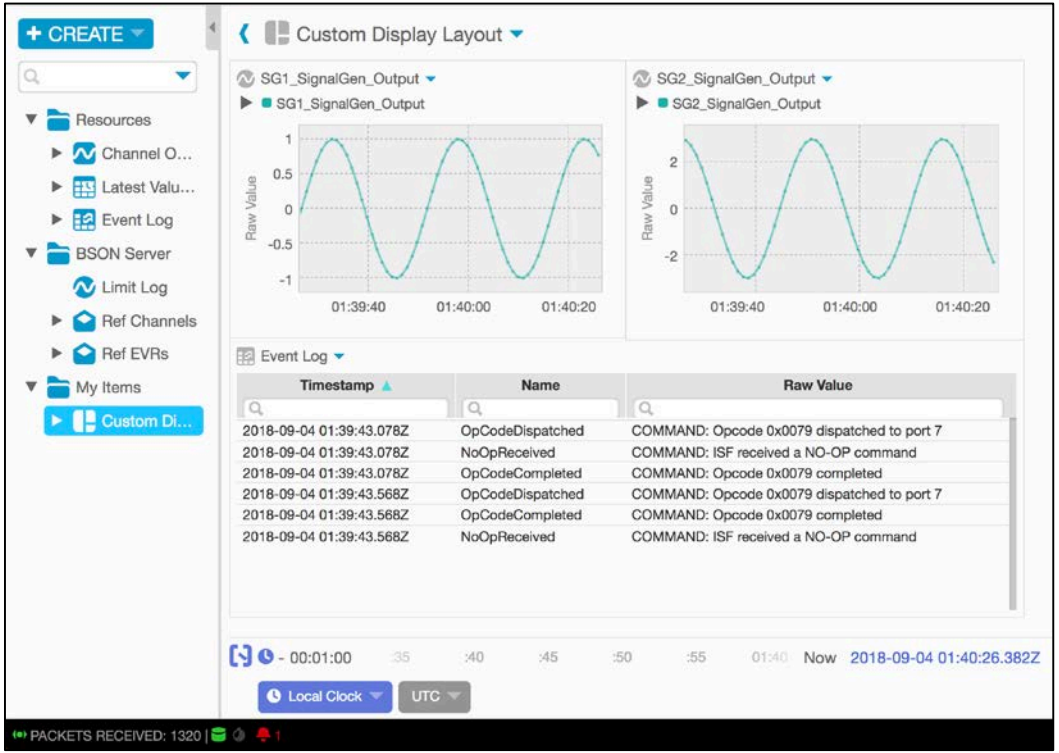
F’ OpenMCT Telemetry Monitoring and Browsing Capability (Architecture)



- Node.js server converts F′ binary to BSON packets
- F´ XML used to generate JSON dictionaries

F’ OpenMCT Telemetry Monitoring and Browsing Capability (Capability)

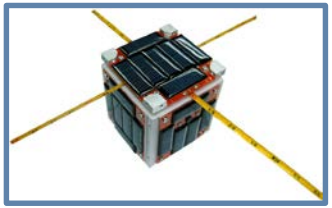
- Open Mission Control Technologies (OpenMCT) open source web framework for visualization of telemetry (see https://github.com/nasa/openmct)
- Node.js Telemetry Server
  - Consumes streaming F´ data
  - Stores telemetry histories using LevelDb database
  - Serves web application content
  - Embedded LevelDb layout persistence
- Client web application
  - Real-time and historical plotting
  - User-customizable layouts
  - Server status indicator in client



F´ Native Protocol & Radio Adaptations (Part 1 Protocol)

- Native message packets assumed within framework
  - Described by descriptors
    - 0 => Commands
    - 1 => Event Log Messages (a. k. a. EVR is Event Report)
    - 2 => Channel Telemetry
    - 3 & 4 => Uplink/Downlink Entities (e.g. RAM Data Product, RAM or DISK Files, etc.)
    - Possible to define others with packets in future
- Often packets (especially simple telemetry) are wrapped with larger packets defined to gain efficiency in transport – Not yet supported in F´
  - Sequences are constructed from commands
- Various protocols used to communicate with radios – None yet supported in F´
  - Consultative Committee for Space Data Systems (CCSDS)
  - AX.25 (Amateur X.25) often used for UHF Ham Radio point-to-point packets

F´ Native Protocol & Radio Adaptations (Part 2 Adapters)



Cubsat with tape measure antenna

RF Channel

RADIO

TCP, UDP or ZeroMQ Connections

Physical Hardware Adapters

Protocol Translation Adapter

Threaded TCP Socket Server (Single Target TCP Socket Connection)

Clients

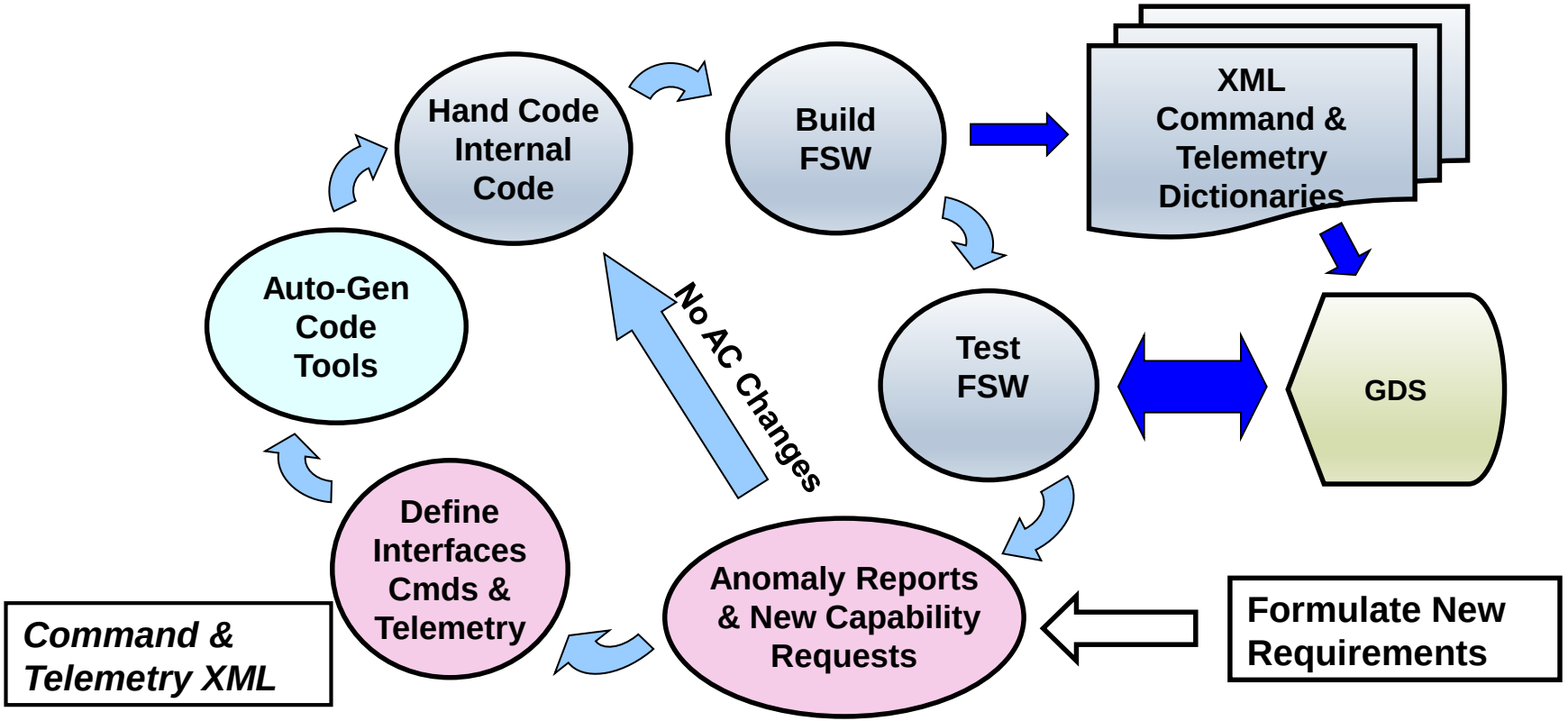
Physical Connection
USB 2.0 / 3.0
RS-422
RJ-11 Ethernet
Others

Projects insert specific adapters to support h/w interfaces and protocol translations (e.g. CCSDS for NASA/DOD, X.25 for University Ham Radio)

Fight / Ground Dictionary Concept

- Ground Dictionaries are the means by which FSW and GDS know how to talk with each other
  - An interface contract between FSW and GDS and visa versa
  - Often represented as flat files that are shared (e.g. XML, JSON, etc.)
  - A database approach such as the web application Dictionary Management System (DMS) (NPO-49751) at JPL is sometimes used as well
- Information shared
  - Commands => opcodes, arguments, mnemonics
  - Event Log Messages => ID's, argument values, formatting, severity
  - Telemetry Channels => ID's, value type, formatting
  - Parameters => ID's, name, type
  - Serializables => Name, data members name/type pairs

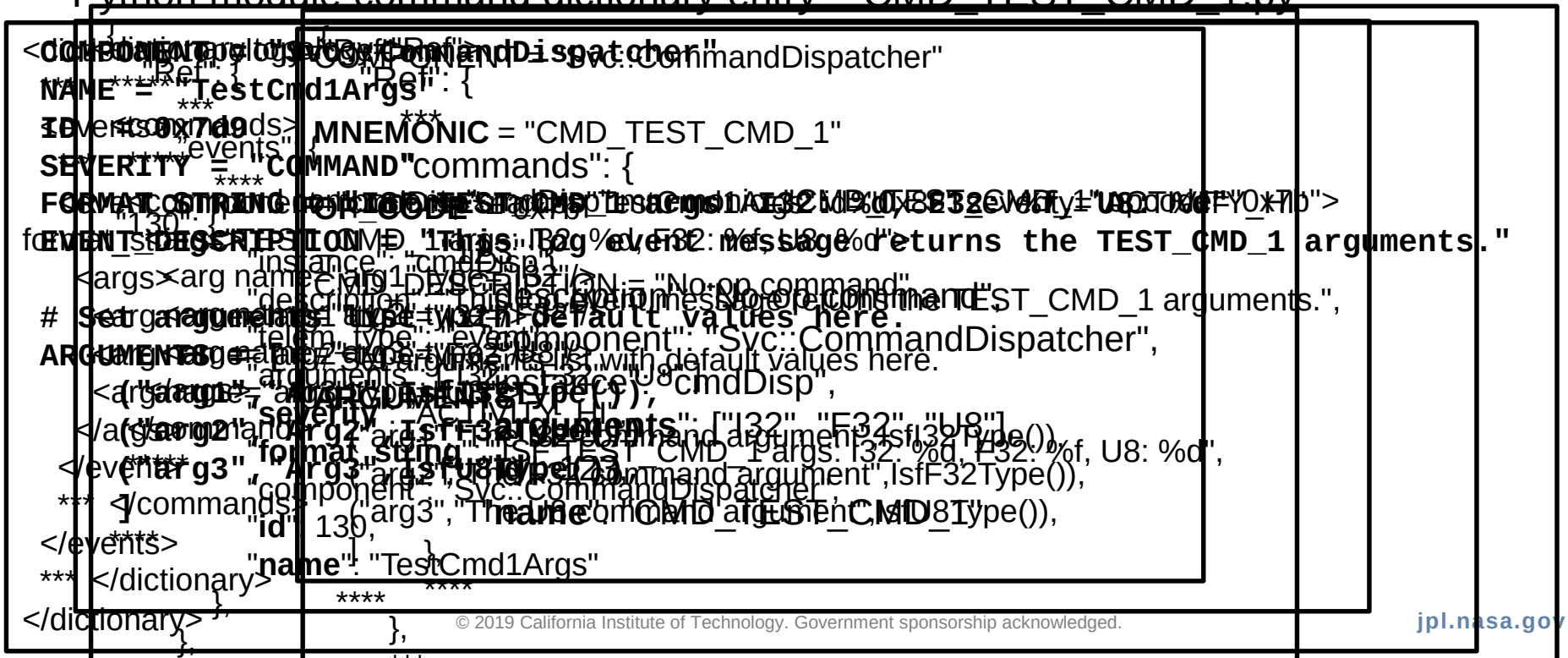
Flight / Ground Dictionaries In A Process Context



Extracted from CL#07-2539

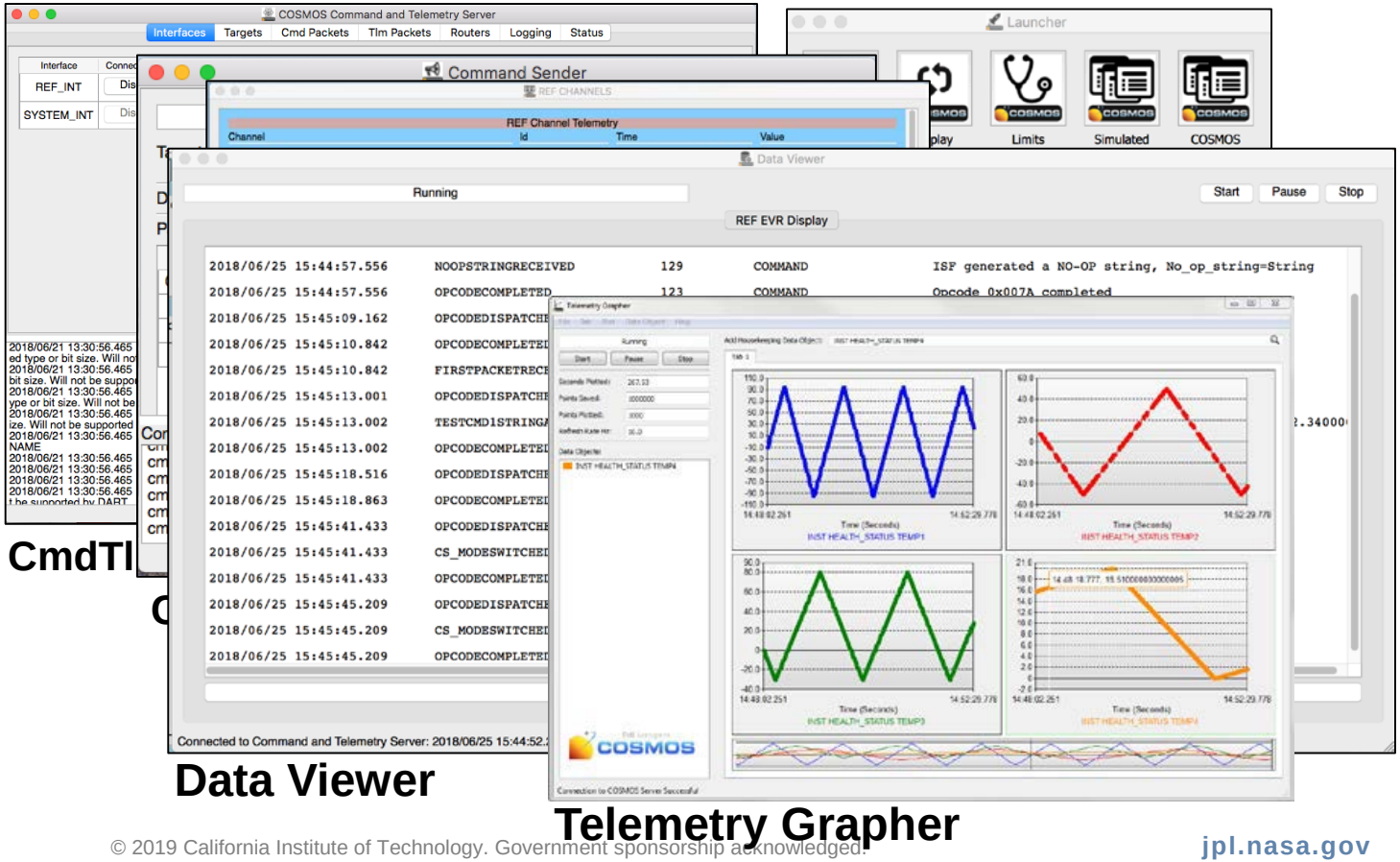
F´ Flight / Ground Dictionary Examples

- F´ utilizes Python Modules, various XML file formats, and JSON



F´ COSMOS GSE Adaptation & Alternative (Capability)

- COSMOS is an open source GSE (a.k.a. GDS) (see https://cosmosrb.com)
- Multi-target capability
- Configurable for both hardware i/f and software protocols
- F´ XML converted to COSMOS text files for configuration
- Over 15 client apps
- Ruby test script API
- Python API
- Telemetry history browsing capability



F´ GDS Summary

- Intention is to provide F´ user community an out-of-the-box ready to use GDS solution that can run on Linux, Mac OSX, or Windows without any mission specific tailoring
  - User defines mission specific dictionaries (commands & telemetry)
  - Dictionaries are automatically generated from topology model
- GDS provides both an end-user GUI tool and an integration test API
  - Work is going on to improve the capabilities
- Other GDS solutions such as COSMOS can be connected to F´ developed systems
- ***Contribute!!!***
  - *We are interested in your improvements, new GUIs, new client tools, new servers, or just ideas*

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov