



Simple Satellite Overview

OSK v2.5



Simple Sat (SimSat) Objectives



- **SimSat is a fictional spacecraft that provides a reference mission context**
- **Provide a complete application suite illustrating**
 - What apps are required to meet a mission's requirements
 - How they are configured and integrated as a system
- **Provide example scripts**
 - Integration test script
 - Operational script
- **Provide context for training exercises**

This slide deck does not cover OSK or the cFS.
See OSK Quick Start Guide or User's Guide for OSK and cFS descriptions.

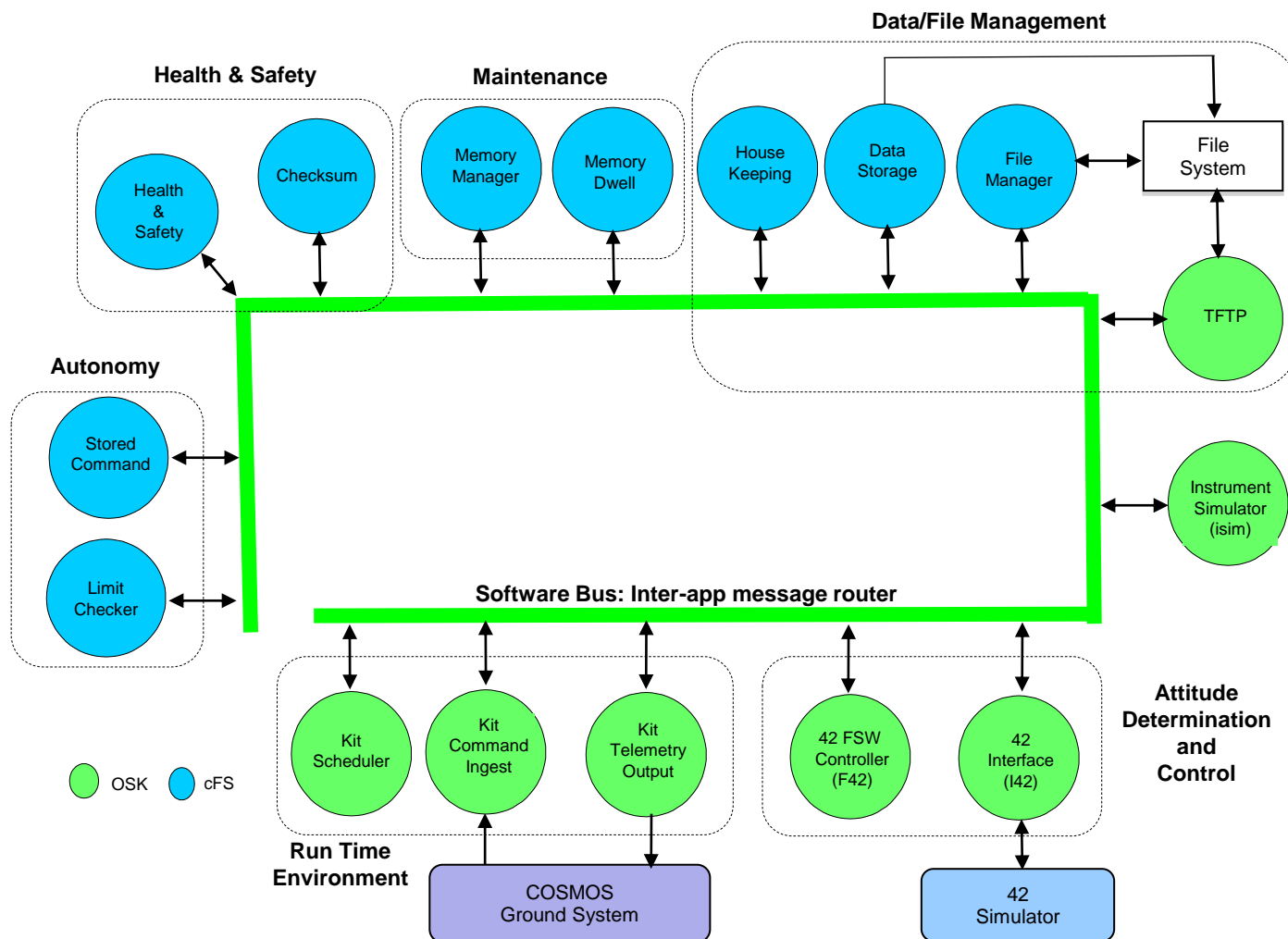
- **Low Earth Orbit (LEO)**
 - 90 minute orbit
 - One 15 minute ground contact per orbit with bi-directional communications
- **One science instrument, iSim**
 - Detector's 1Hz scan produces 10 bytes of data
 - Power on sequence
 - Apply power, wait for instrument warm up (~20s), then enable science
 - Power off sequence
 - Disable power



SimSat Ops Concepts & Requirements



- **Science team requires**
 - A 1Hz auxiliary spacecraft data containing time, attitude, orbit data, and instrument status
 - Start science during a ground contact. Can be automated but ops prefers to monitor instrument health.
- **Ground contact resources/schedule are preplanned**
 - Implies autonomous operations can be loaded on board using stored commands
- **Each pass can either be a low or high downlink rate**
- **FSW must autonomously monitor instrument health and power off the instrument in the event of a fault**





SimSat Applications (1 of 3)



- **The previous slide shows a cFS “bubble” chart where each app is a bubble and they communicate via messages on the software bus.**
 - The blue cFS apps are reusable open source apps that are available on <https://github.com/nasa/xx> where ‘xx’ is the abbreviated app name
 - The green OSK apps were written specifically for OSK
 - The external COSMOS and 42 interfaces use UDP and TCP respectively
- **Apps are designed to perform a dedicated function with clear interfaces and they operate in groups to achieve higher level mission objectives**
- **Runtime Environment Apps**
 - Kit Command Ingest (KIT_CI) receives CCSDS command packets from COSMOS and sends them on the Software Bus
 - Kit Telemetry Output (KIT_TO) reads CCSDS telemetry packets from the Software Bus and sends them to COSMOS
 - Kit Scheduler (KIT_SCH) contains tables that define when to send messages on the Software Bus
 - Apps can use these messages to perform synchronous activities, e.g. sending their housekeeping status packet



SimSat Applications (2 of 3)



- **Data/File Management**

- File Manager (FM) provides a ground interface for performing common directory and file operations
- Data Storage (DS) reads packets from the software bus and writes them to files according to table-defined
- Housekeeping (HK) creates new telemetry packets from pieces of other telemetry packets. The new packets are written to the SB and can be stored and/or telemetered.
- Trivial File Transfer Protocol (TFTP) transfers files between the flight and ground COSMOS. There's an open source CCSDS File Delivery Protocol (CFDP) app that will be added in a future release.

- **Autonomy**

- Limit Checker (LC) monitors one or more telemetry values and start stored command relative time sequences (RTSs) in response to limit violations
- Stored Command (SC) Provides services to execute preloaded, table-defined command sequences at predetermined absolute or relative time intervals



SimSat Applications (3 of 3)



- **Attitude Determination and Control Apps**
 - 42 Interface (I42) manages a TCP/IP connection to 42 and transfers actuators/sensor packets to/from 42
 - 42 FSW (F42) Implements the “ThreeAxisFsw” attitude control algorithm defined in 42
- **Maintenance**
 - Memory Dwell (MD) creates telemetry packets containing contents of memory location specified in dwell tables
 - Memory Manager (MM) provides read/write access to memory
- **Health & Safety**
 - Checksum (CS) monitors checksums across table-defined static code/data regions and reports errors
 - Health & Safety (HS) monitors table-defined application check-in and event messages and reporting errors and/or starting a RTS to address the issue



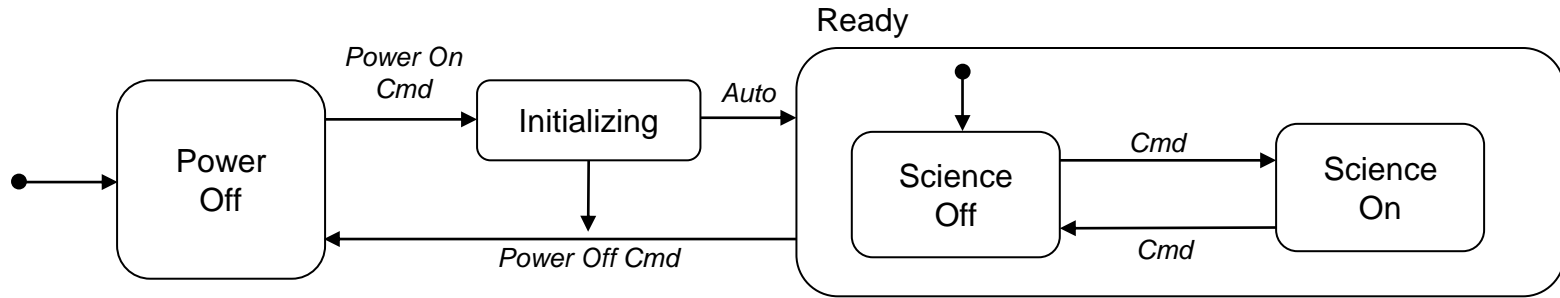
SimSat App Solution (1 of 5)

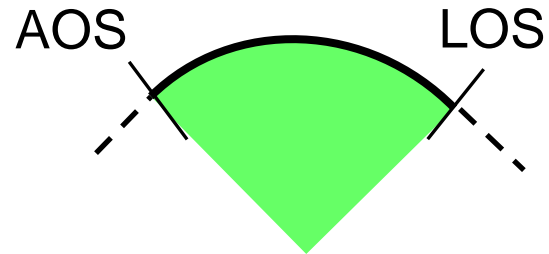


- **iSim App**
 - Simulate science instrument data
 - Creates science data files and moves them to downlink directory
 - Commands
 - Power instrument on/off
 - Start/stop science data
 - Set/clear fault
 - Telemetry
 - Instrument status: Off, Initializing, Ready
 - Science data: Enabled, disabled
 - Fault: True, False
 - Use informational events to trace behavior.



SimSat App Solution (1 of 5)



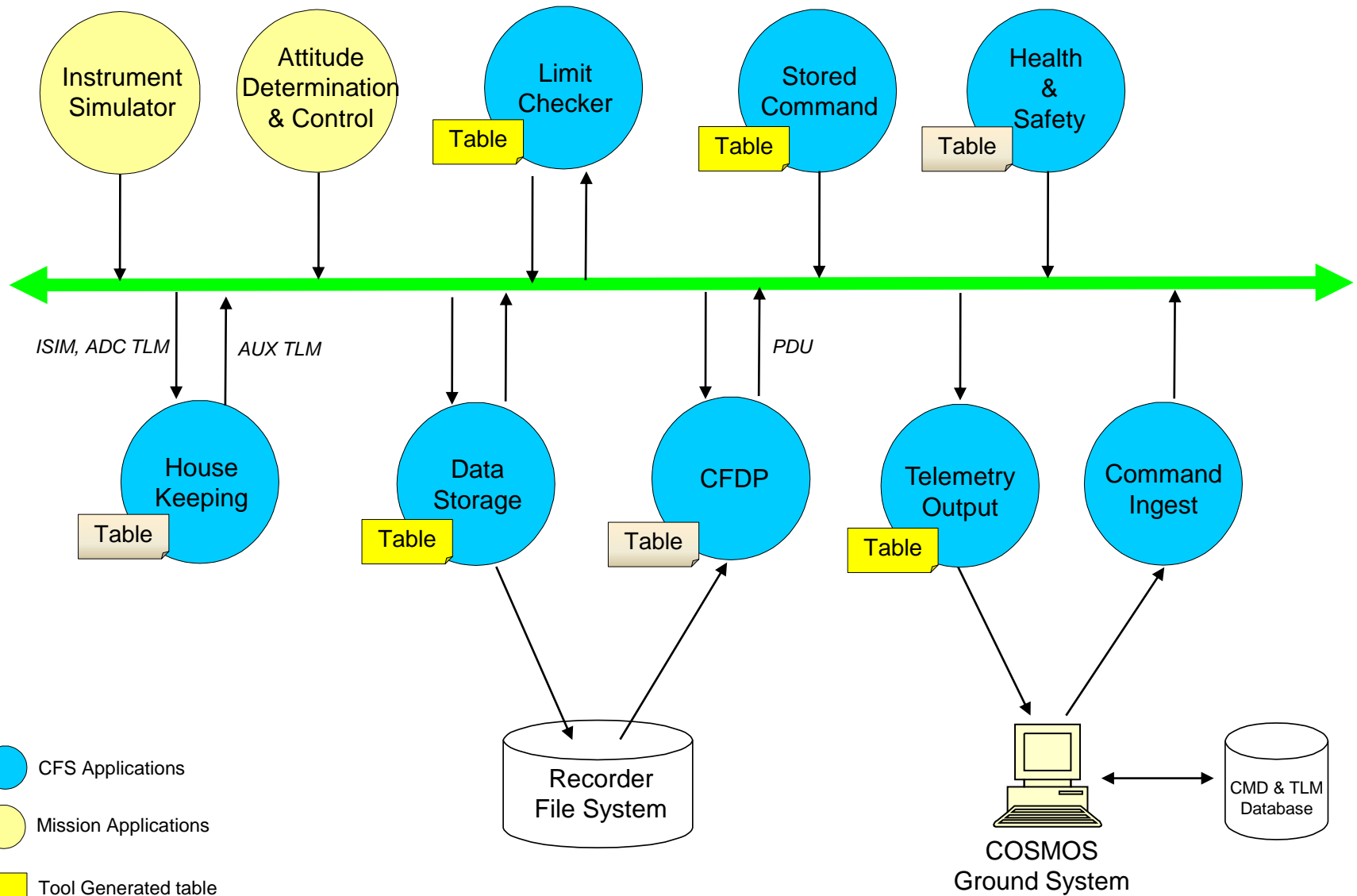


Assess health

Retrieve Data

Upload

- Create and load stored command sequences to run a demo
- Use OpenSatKit exercises to work through an example





SimSat Application Configuration (1 of 4)



- **Housekeeping (HK)**
 - HK Combo packet 1 (0x089C) comes predefined with HK and contains HK data from each cFE app
 - Scheduler Slot 6, Activity 6
 - Create a new auxiliary science data packet using HK combo packet 2 (0x089D) that combines instrument status telemetry and ADC data
 - Scheduler Slot 6, Activity
- **Data Storage (DS)**
 - Configure file & filter tables to create:
 - Event message file: DS filter 6, file 0
 - Auxiliary data file: DS filter 15, file 6



SimSat Application Configuration (2 of 4)



- **Telemetry Output (KIT_TO)**
 - Doesn't support filter tables
 - Create low/high tables that define which packets will be output for each scenario
 - Load low/high rate tables using stored commands
- **Limit Checker**
 - Monitor instrument status for the ready state and start RTS to enable science
 - Monitor instrument for a fault and start RTS to power off instrument if a fault persists for 3 seconds
 - WP #12 – Monitor ISIM fault
 - AP #2 – Start RTS 6 to stop science and power off the instrument



- **Stored Command (SC)**

- Create Relative Time Sequences (RTS) to perform specific operational functions

- RTS Definitions

- 6 - Power off science instrument

- TODO

- Load KIT_TO low rate table
 - Load KIT_TO high rate table
 - Power on science instrument
 - Start science
 - Stop science
 - Start pass
 - End pass

- Absolute Time Sequence (ATS)

- Create an ATS to manage 24 hours of operations
 - For periodic operations such as bSat the duration of an ATS should be much longer than the ATS upload frequency to account for contingencies



- **File Manager (FM) & Trivial File Transfer Protocol (TFTP)**
 - Use FM to perform directory listing of files to downlink
 - Transfer files from flight to ground using TFTP
- **Checksum**
 - Configure checksum to monitor the stored command table checksums
- **CCSDS File Delivery Protocol (CF)**
 - Currently not in the kit
 - CF could significantly change the operational scenarios. Most of the file transfer and onboard file deletion activities could be automated if CF's "hot directory" and Class 2 mode are used



Contact Planning



- **For each ground contact**
 1. Assess health of spacecraft
 - a. Take action if needed
 2. Manage onboard data files
 3. Uplink new ATS if needed



Assess Health of Spacecraft



- **Verify expected spacecraft state**
 - This is mission specific, includes items such as
 - Expected control mode, clear LC flags, etc.
- **Dump, transfer, and display event log**
 - Event log should not fill up with informational events if you're judicious on how you define events. See cFE training module for guidelines
 - Clear log after log transferred to the ground



Manage Onboard Data Files



- 1. Use FM to list directory to a file**
- 2. Transfer directory file to the ground**
- 3. Sort files in priority order**
- 4. Transfer files in priority order**
 - a. Delete each file after successful transfer



Reference

