



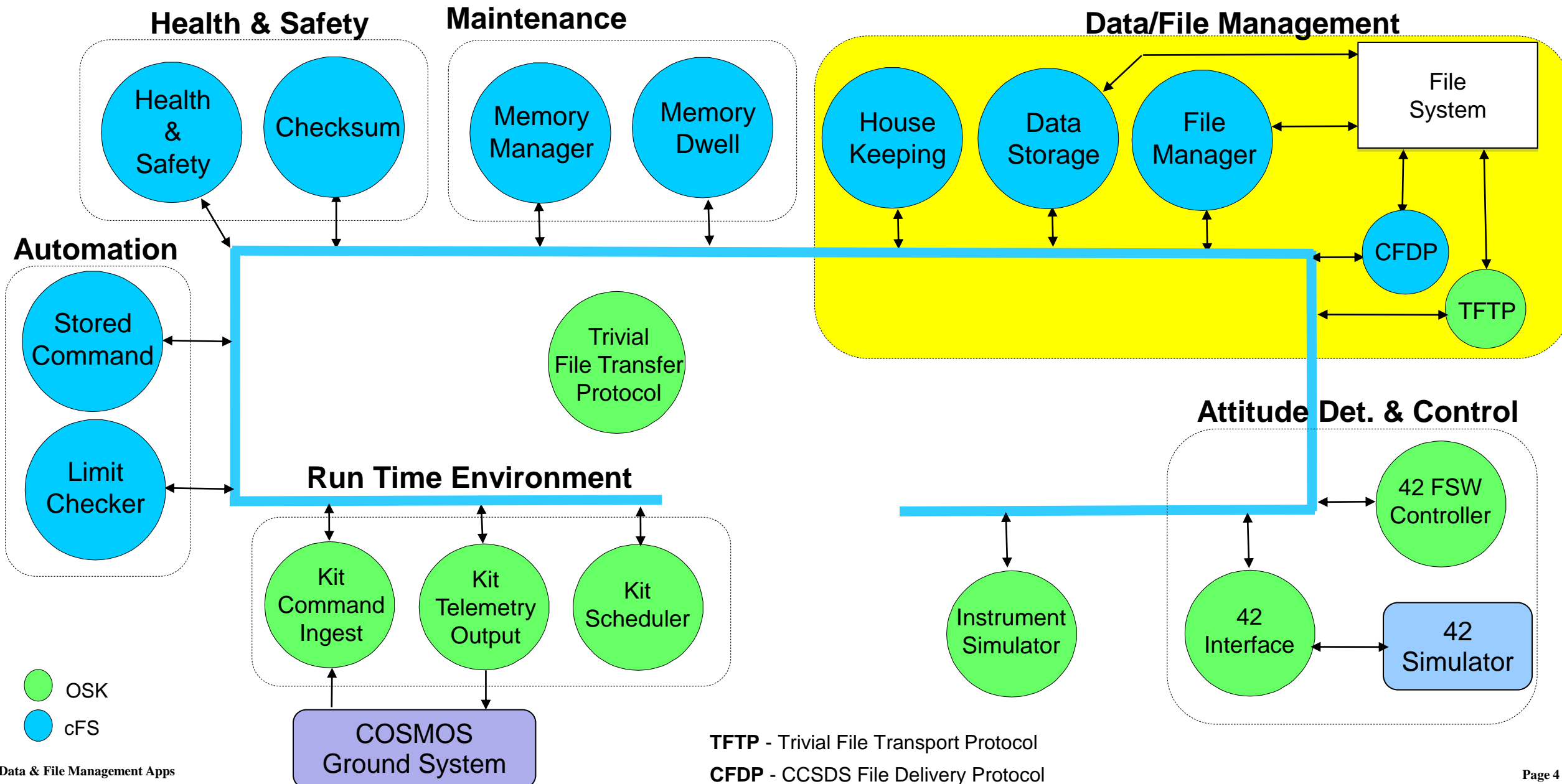
Core Flight System (cFS) Training

**Community Apps:
File & Data Management**

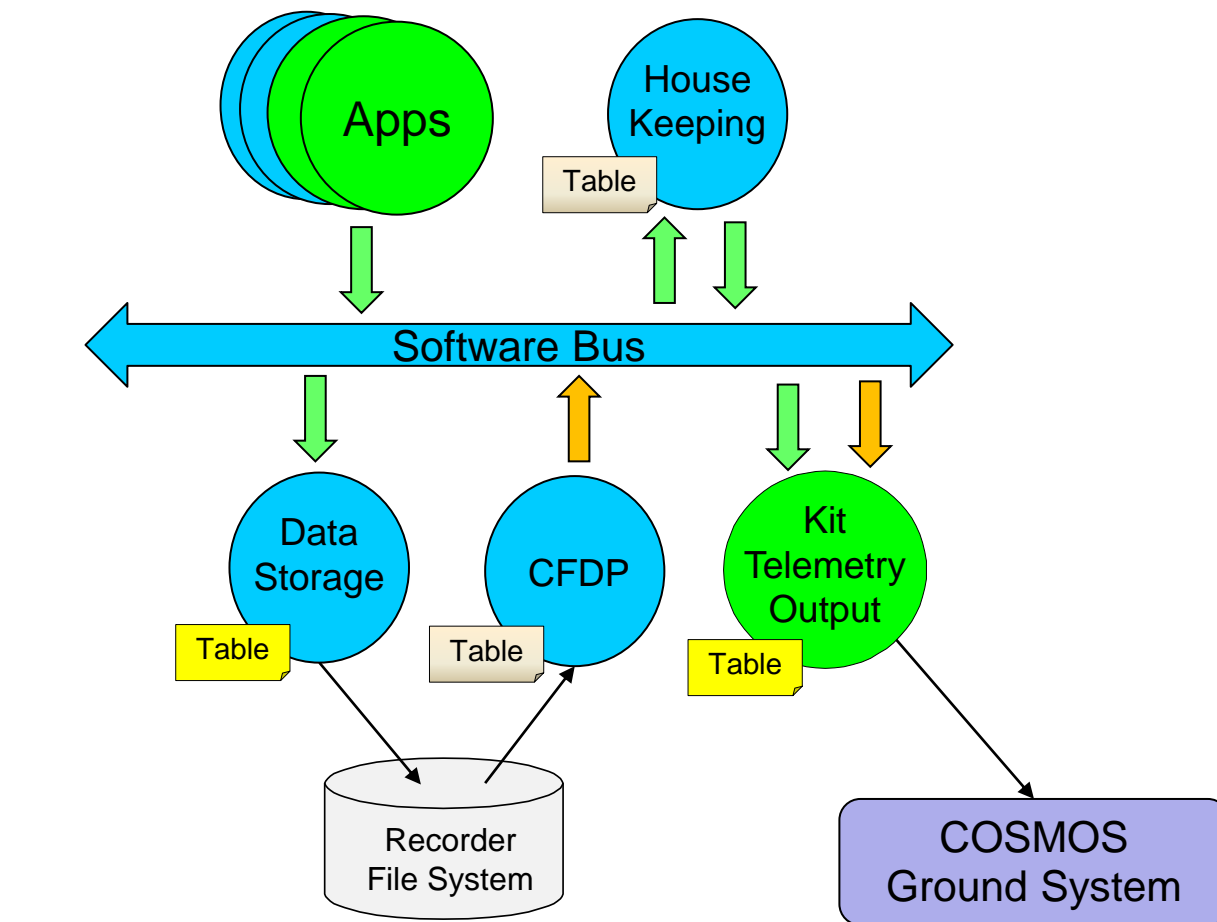
- **These are draft slides released with OSK v2.6**
- **An introductory training video was released with v2.5 and FM video with v2.6 so these slides are mature**
- **The remainder of the slide deck contains slides that were collected from existing material for each app. This material will be matured as each detailed video is created**



Data-File Management Application Overview



- **File Manager (FM)**
 - Provides commands for manipulating directories and files
 - Users obtain information about directories, files and file systems by requesting one time telemetry packets or dumping information to a file
- **Housekeeping (HK)**
 - Combines subsets of multiple source packets from any app into a new packet
 - New packets are generated when an “Output Pkt” command is received
- **Data Storage (DS)**
 - Receives messages from the software bus and writes them to files
 - Table parameters determine which messages are stored in which files
- **File Transfer**
 - **CFDP** implements the CCSDS File Delivery Protocol
 - Sophisticated protocol designed for space applications with many features such as prioritized downlink queues and automatic onboard file deletion after file transfers
 - **TFTP** implements the Trivial File Transfer Protocol
 - Simple lockstep protocol originally designed network remote host booting scenarios

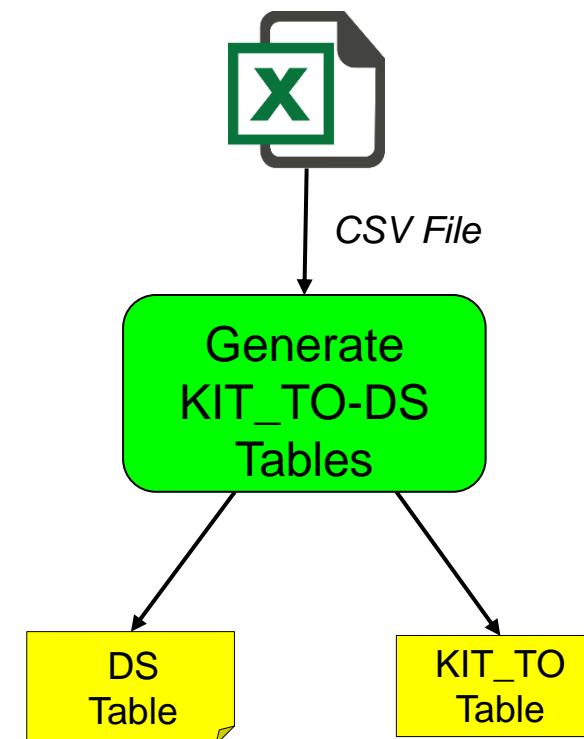


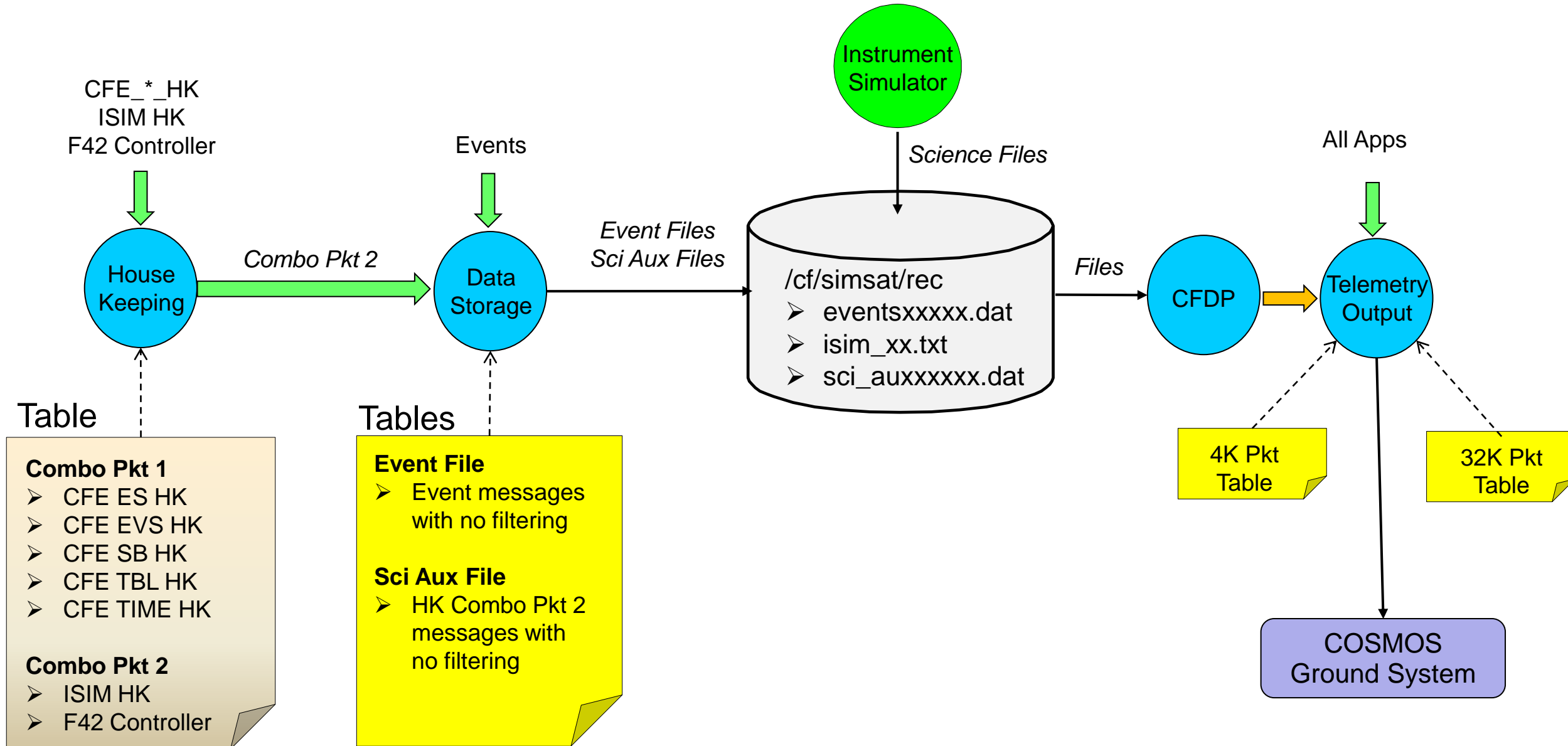
Real-time data
 Playback data

Tool-generated table

CFDP = CCSDS File Delivery Protocol (CFDP)

Spreadsheet used to analyze & compute telemetry storage & real-time rates for mission operational scenarios/modes

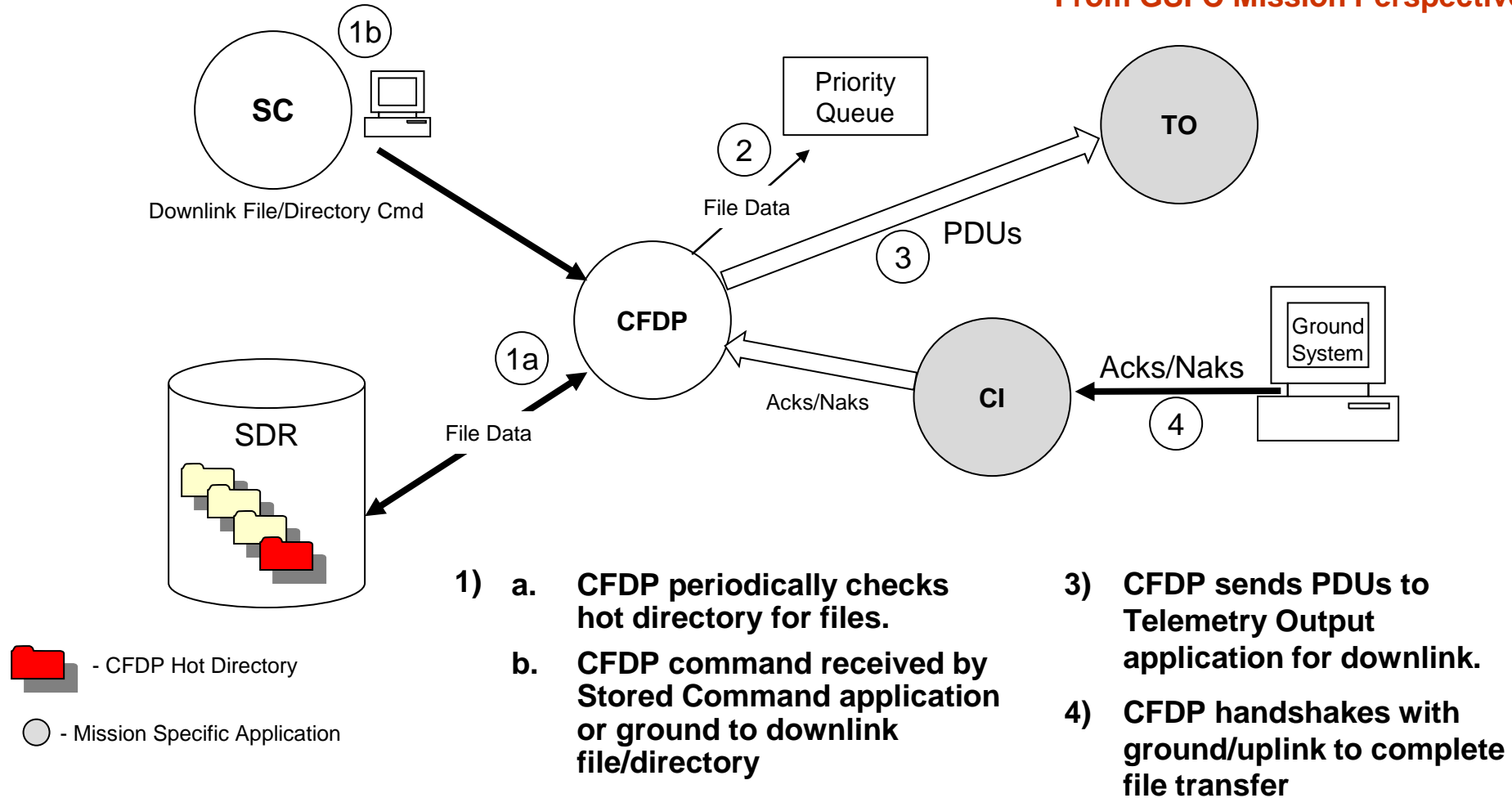




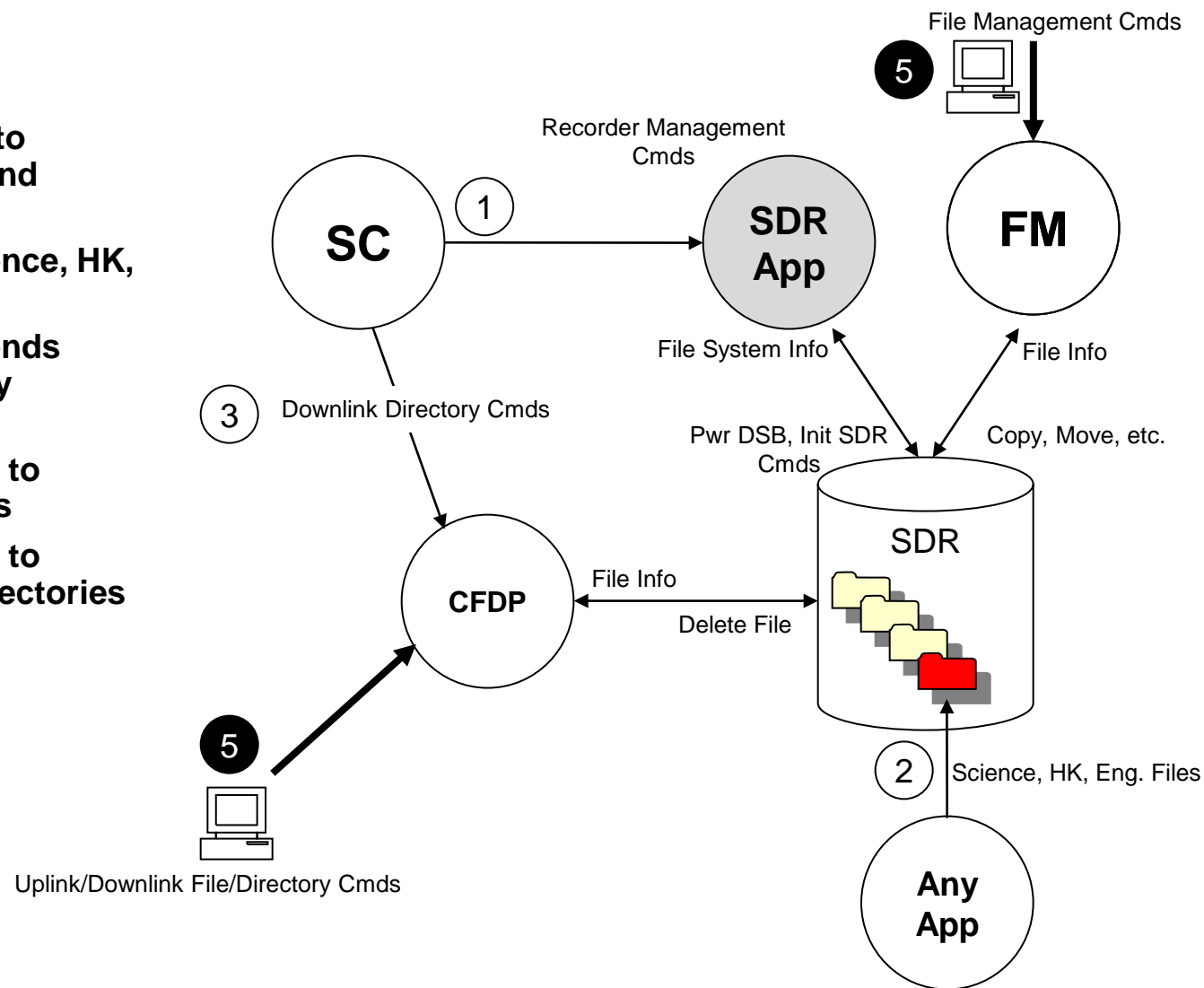


Operational Scenarios

* From GSFC Mission Perspective



- 1) **Stored commands sent to initialize file system(s) and create partitions**
- 2) **Applications create Science, HK, and/or Engineering files**
- 3) **SC (typically via ATS) sends CFDP downlink directory commands**
- 4) **Ground commands sent to uplink and downlink files**
- 5) **Ground commands sent to manage the files and directories in the file system(s).**

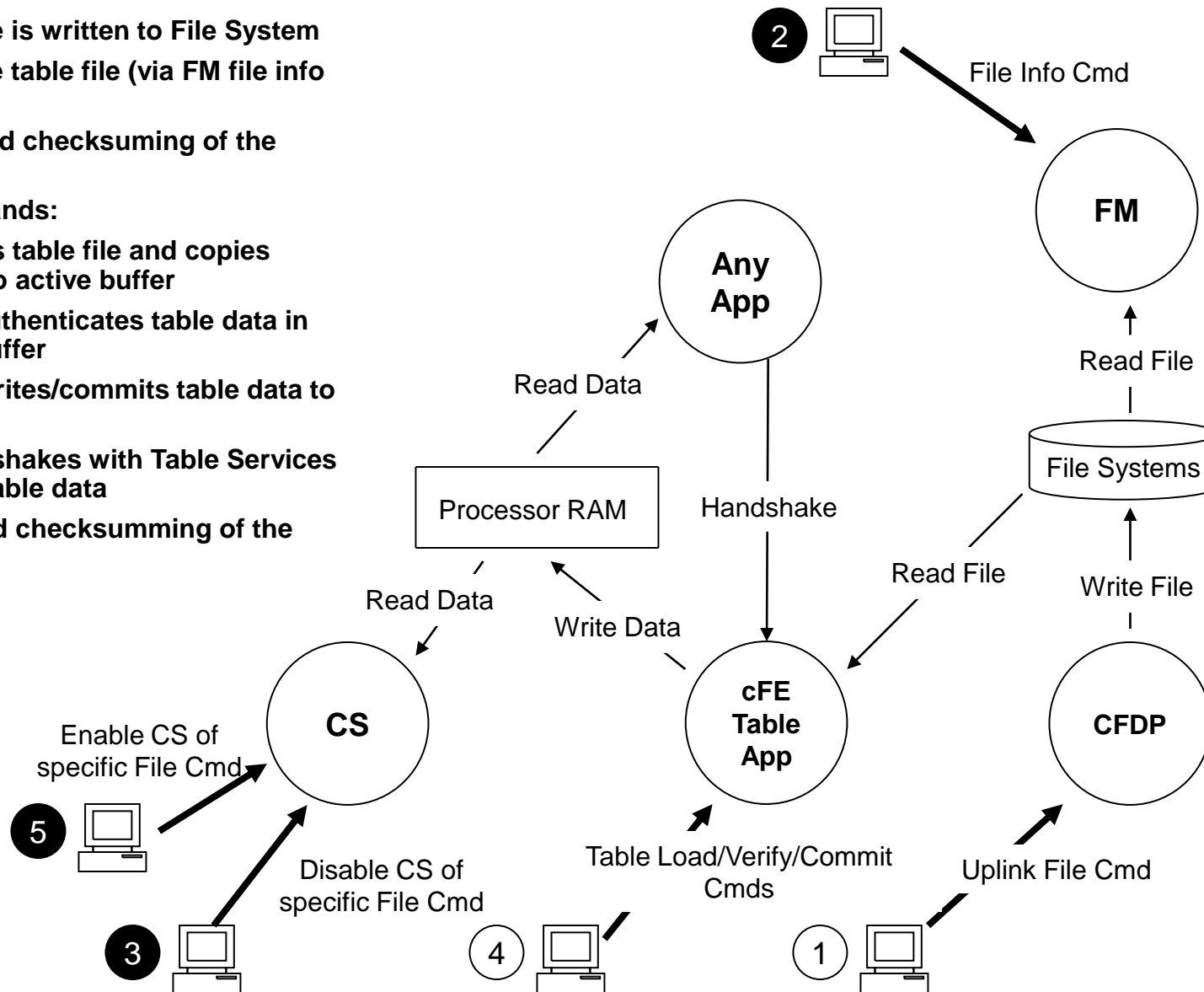


- CFDP Hot Directory

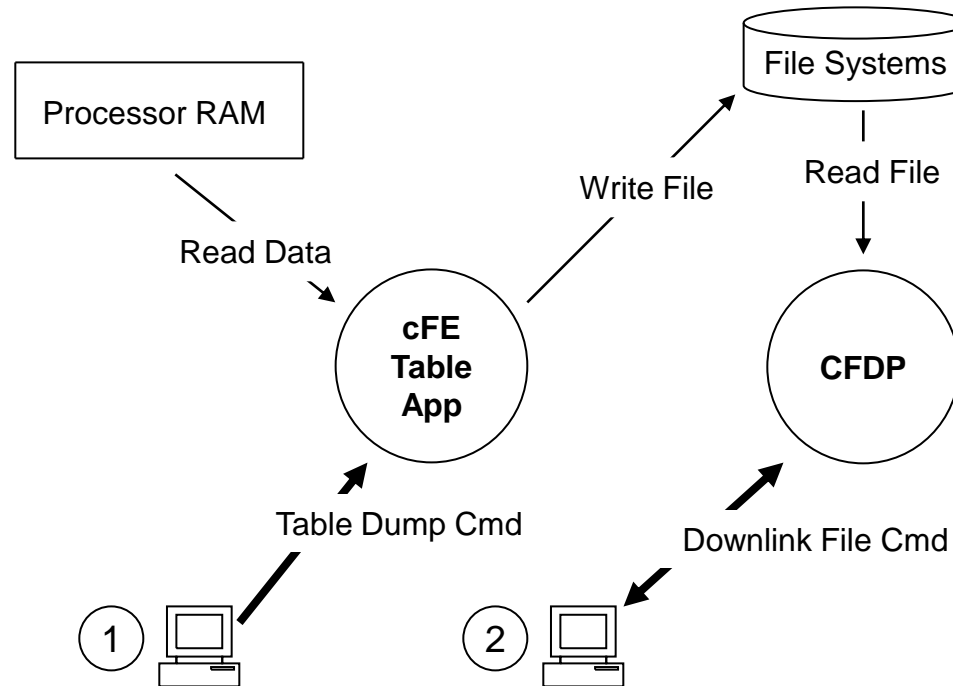
- Mission Specific Application

- Optional Step

- 1) Uplink table – table is written to File System
- 2) Optionally CRC the table file (via FM file info command)
- 3) Disable background checksumming of the table
- 4) Send Table commands:
 - Load – reads table file and copies contents into active buffer
 - Validate – authenticates table data in the active buffer
 - Activate – writes/commits table data to RAM
- Application handshakes with Table Services to read updated table data
- 5) Enable background checksumming of the table



- 1) **Send Table dump command – table file is written to File System**
- 2) **Downlink file – table is written to ground File System.**

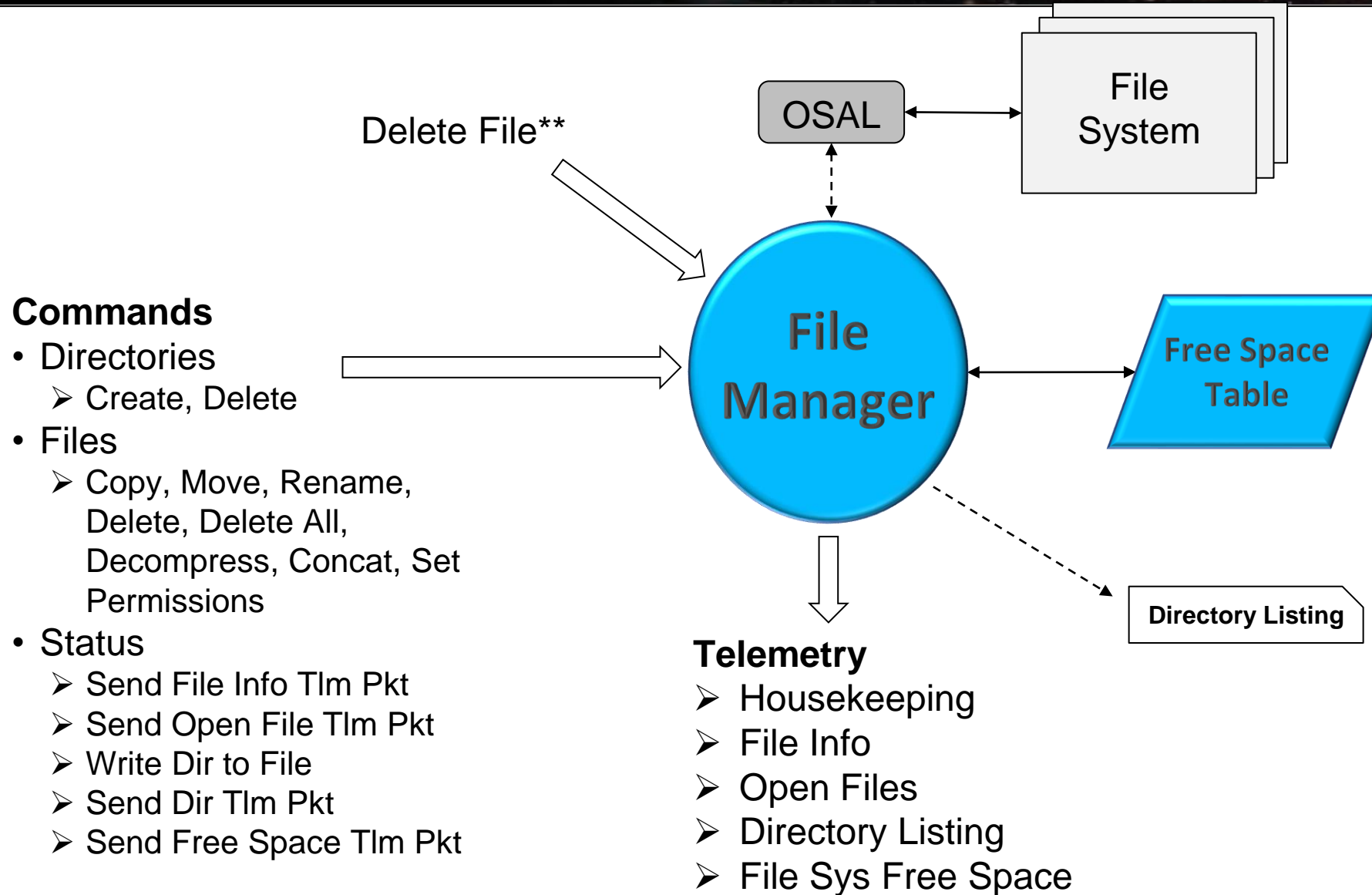




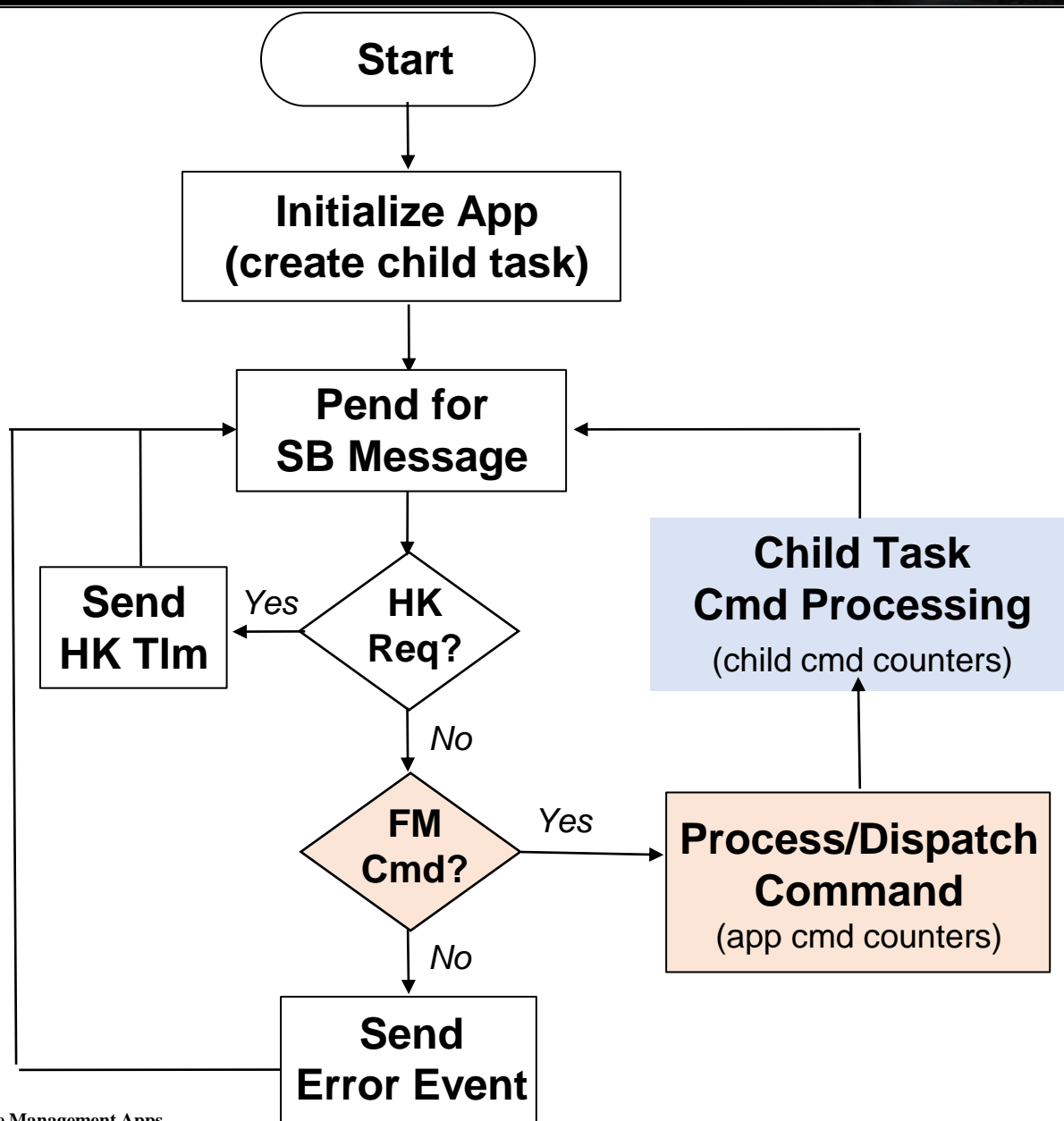
File Manager

<https://github.com/nasa/FM>

- **Provide an interface for managing files and directories**
 - Primary interface is the ground but receiving commands from other apps must also be taken into consideration
- **Meet the specific needs of the spacecraft file management environment**
 - Once a spacecraft is operational, directory structures are typically fairly static
 - The file system 'clients' are based on the spacecraft's interfaces (data producers/consumers) that don't change
 - Deterministic and efficient file system performance is often required
 - Working over a command and telemetry space link limits real-time file system management
 - File transfer protocols like CCSDS File Delivery Protocol (CFDP) offer options to automatically delete a file once a file transfer has completed successfully
- **Operate in the cFS environment**
 - All directory and file commands use the cFS' Operating System Abstraction Layer (OSAL) to access the file system
 - Consistent behavior depends on the underlying OS
 - Executive Service's shell command provides a 'backdoor'



** Onboard command that doesn't affect ground command counters



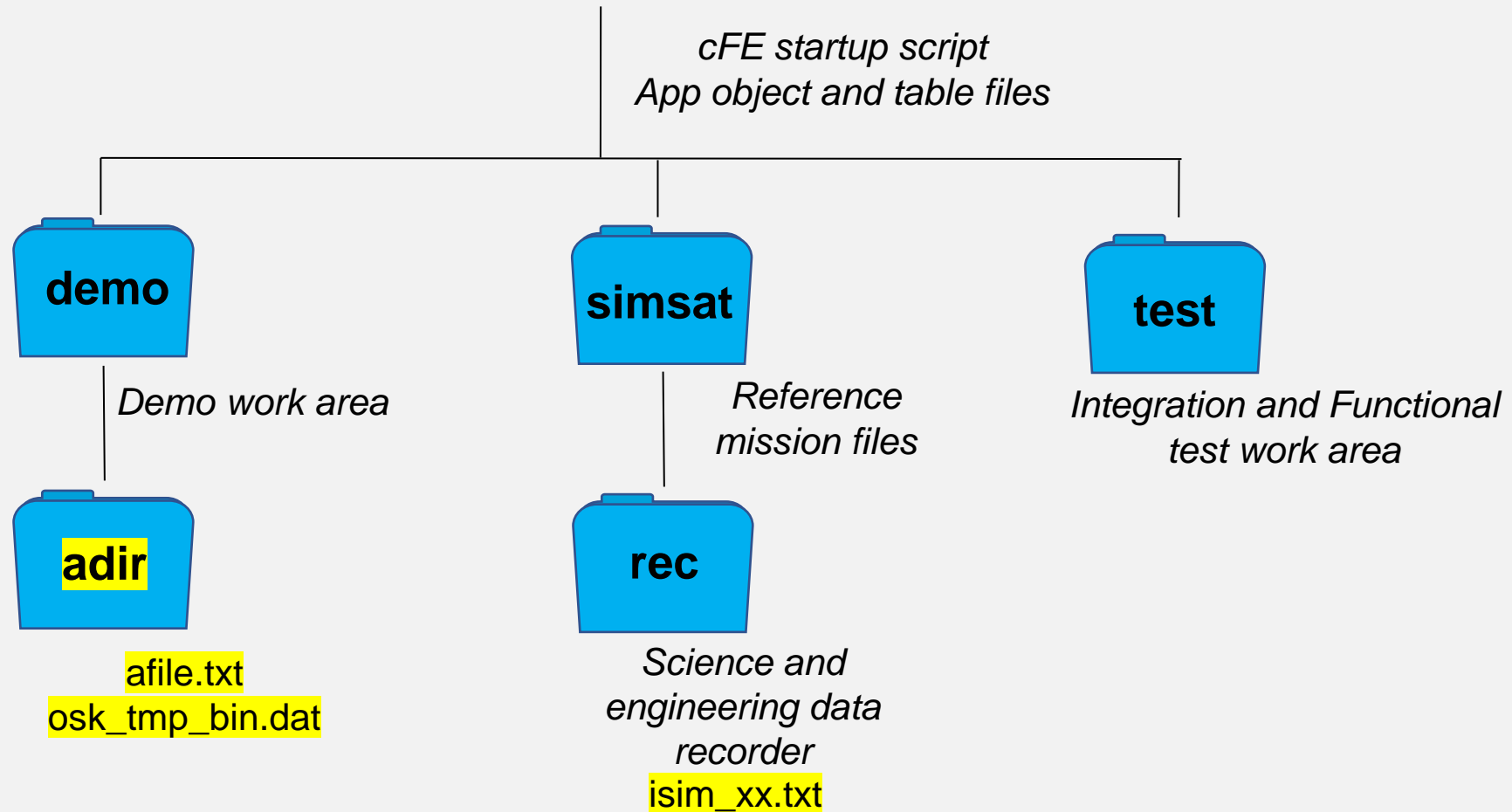
Child Task

- Separate thread of execution that shares main app's memory space
- Implements all command functions except
 - Get File System Free Space
 - Set Free Space Table State
 - Send Open File Telemetry Pkt
- Tune performance using build-time configuration parameters

Command Processing

- Main app processing validates command input parameters
 - Increments valid command count if valid inputs and successful command dispatch to child task
- Child task performs command function and increments command counters if successful

~/OpenSatKit-master/cfs/build/exe/cpu1/cf



Highlighted directory and files used in FM demos

Data Structure Definitions	Description	External Dependencies
FM_DIR_LIST_FILE_ENTRIES	Max directory entries written to a file	Table Manger binary file definition
FM_DIR_LIST_PKT_ENTRIES	Max directory entries listed in the directory telemetry packet	Telemetry packet definition
FM_TABLE_ENTRY_COUNT	Max FreeSpace table entries	Table Manger binary file definition
OS_MAX_PATH_LEN	OSAL definition used in multiple full path filename definitions	C&T pkt and binary file definitions
OS_MAX_NUM_OPEN_FILES	OSAL definition used in max open file telemetry listing	Telemetry packet definition

Child Task Definitions	Description
FM_CHILD_TASK_PRIORITY	Execution priority for child task
FM_CHILD_QUEUE_DEPTH	Max number of commands that can be queued to child task
FM_CHILD_FILE_BLOCK_SIZE	Size of each block read/written from/to files
FM_CHILD_FILE_LOOP_COUNT	Number of file blocks read/written between task sleeps
FM_CHILD_FILE_SLEEP_MS	Duration of child task's sleep between file block reads/writes
FM_CHILD_STAT_SLEEP_FILECOUNT	Number of file status inquires between task sleeps
FM_CHILD_STAT_SLEEP_MS	Duration of child task's sleep between file status inquiries



FM Commands (1 of 2)



Command	Description
Noop	Increments the Command Accepted Counter and sends a debug event message
Reset Command Counters	Initializes the following FM counters to 0: Command Rejected Counter, Command Accepted Counter
File Copy	Copies the command-specified file to the command-specified destination file or directory
File Move	Moves the command-specified file to the command-specified destination file or directory
Rename File	Renames the command-specified file to the command-specified file
Delete File	Deletes the command-specified file, if and only if, the file is closed
Delete All Files	Deletes all files in the command-specified directory, if and only if, the files are closed.
Decompress File	Decompresses the command-specified file creating the command-specified destination file
Concatenate Files	Concatenates the command-specified source files creating the command-specified destination file
File Information	Creates and sends a software bus message containing the file size, last modification time, and file status (Open, Closed) of a given file, if and only if, the file exists



FM Commands (2 of 2)



Command	Description
List Open Files	Creates and sends a software bus message containing the number of open files, the name/path of each open file, and application identifier associated with each open file
Create Directory	Creates the command-specified directory
Delete Directory	Removes the command-specified directory, if and only if, the command-specified directory is empty
Directory Listing via File	Writes to a file the complete listing of the command-specified directory
Directory Listing via Message	Creates and sends a software bus message containing the contents of a directory (up to <PLATFORM_DEFINED> filenames, starting at the command-specified offset)

Telemetry Point	Description
FileStatus	Status indicating whether the file is Open or Closed
CRC_Computed	Flag indicating if a CRC was computed on the command specified file
<OPTIONAL> CRC	Computed CRC of file contents
FileSize	Size of file in bytes
LastModifiedTime	System time the file was last modified
Filename	Echo of command specified filename

- CRC ground tool provided



FM Open File Listing Telemetry Message



Telemetry Point	Description
NumOpenFiles	Number of open files in the FSW system
FileNames[1..n] where n = <PLATFORM_DEFINED> FM_MAX_OPEN_FILE_LIST_MSG_FILES	Names of open files in the FSW system
AppNames[1..n] where n = <PLATFORM_DEFINED> FM_MAX_OPEN_FILE_LIST_MSG_FILES	Names of applications that have files open in the FSW system



Directory Listing Telemetry Message



Telemetry Point	Description
DirSize	Directory size in bytes
DirOffset	Echo of command specified directory offset
TotalFiles	Total number of files contained in the command specified directory
FileSizes[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Sizes of the files contained within the command-specified directory starting at the command specified offset
FileLastModTimes[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Last modification times of the files contained within the command-specified directory starting at the command specified offset
DirName	Echo of command specified directory name
FileNames[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Names of files contained within the command-specified directory starting at the command-specified offset

- **File Format**
 - Binary
- **File Content**
 - cFE file header
 - Header length
 - Spacecraft ID
 - Processor ID
 - Application ID
 - Creation Time (seconds and subseconds)
 - File Description
 - Echo of command-specified directory name
 - Directory size in bytes
 - Total number of files in the directory
 - For each file contained in the directory:
 - File Name
 - File Size
 - Last Modification Time



Open Files Telemetry Message



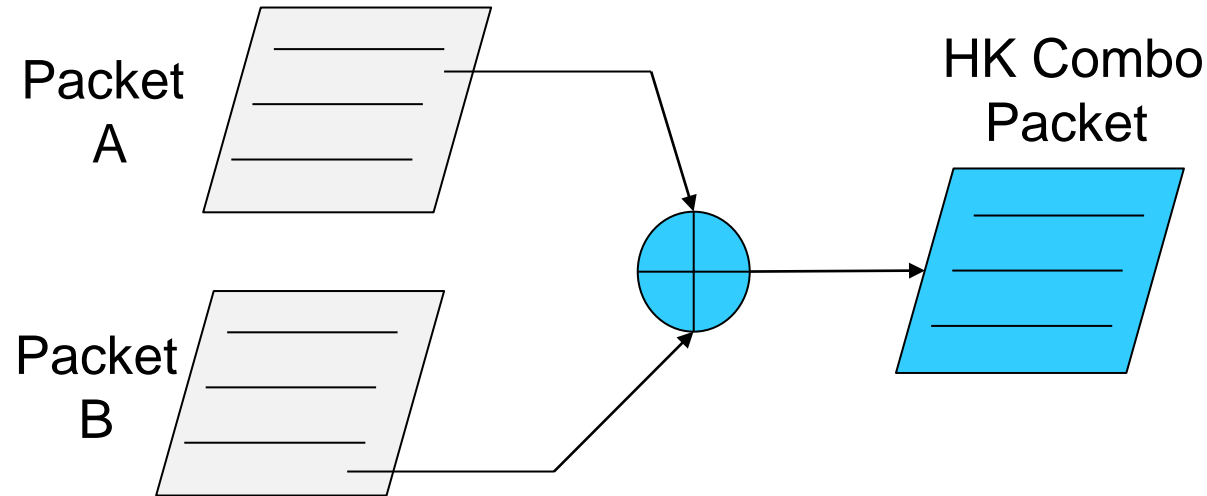
Telemetry Point	Description
CommandCounter	Number of rejected commands
CommandErrCounter	Number of accepted commands
NumOpenFiles	Number of open files in the entire FSW system
BlockSize[1..n]	Block size of drive n
NumBlocks[1..n] where n = <MISSION_DEFINED> FMMaxNumDevices	Number of available blocks on drive n



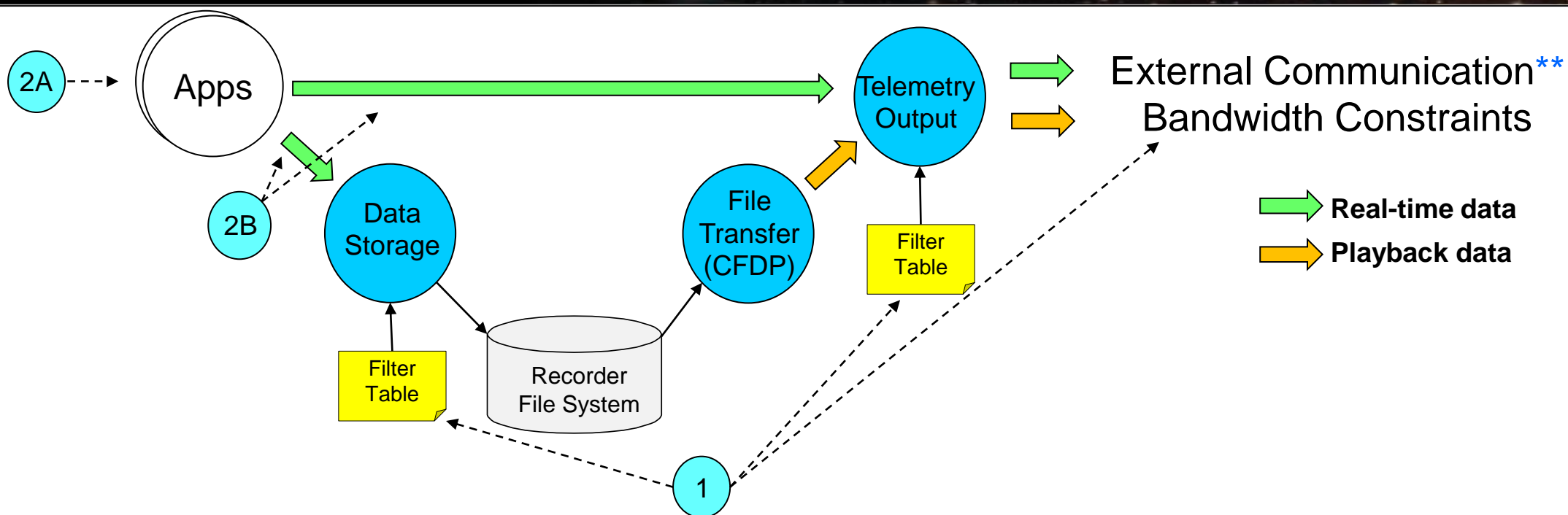
Housekeeping

<https://github.com/nasa/HK>

- **Provide a mechanism to help optimize telemetry storage and downlink utilization**
- **Allow the telemetry stream to be configured from an operational perspective of what rate each data item needs to be received**
 - Often convenient to think in terms of fast, medium and slow
- **Decouple app functional packet designs from mission specific telemetry requirements**
 - Apps designs can group logically related data into packets



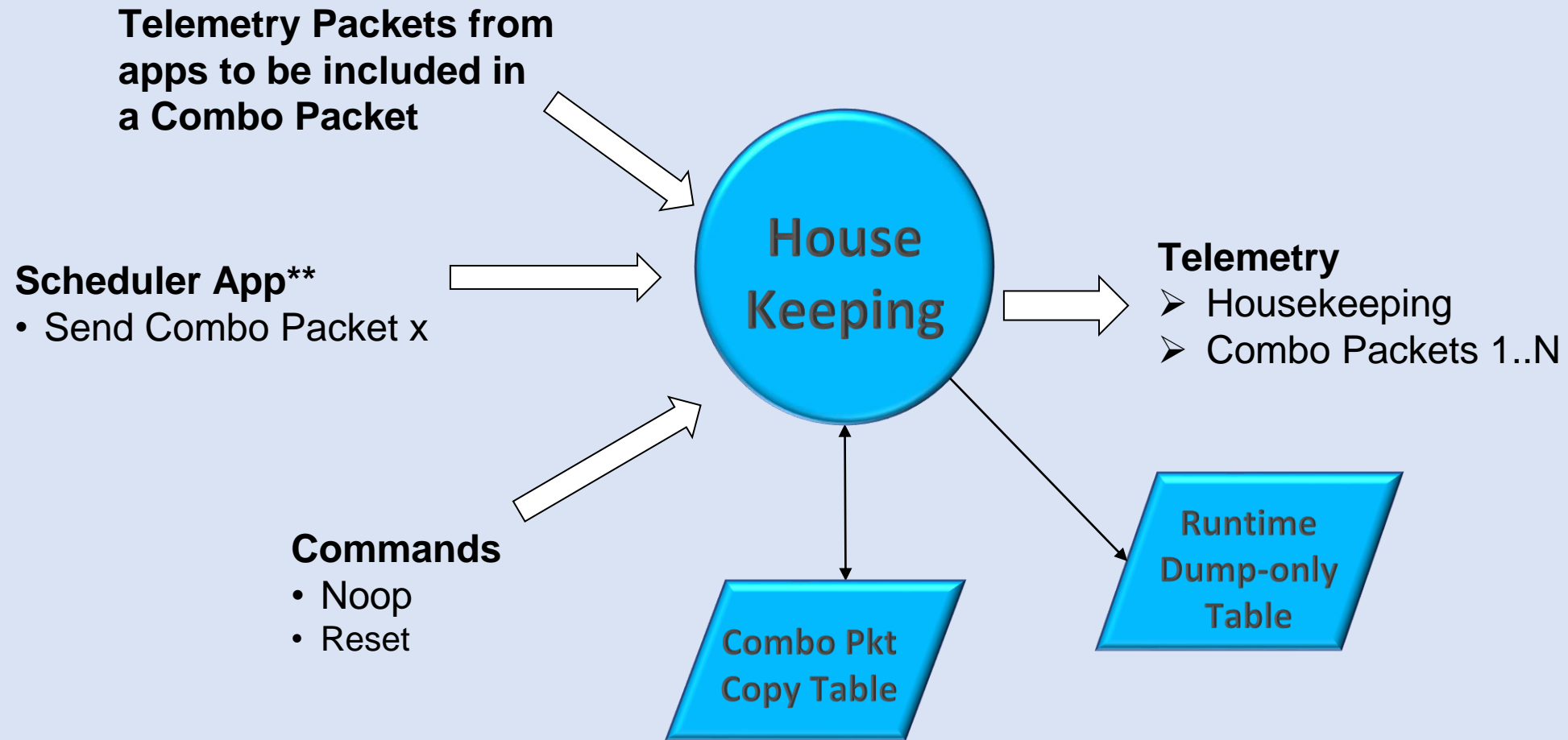
- **HK achieves objectives by combining portions of telemetry packets from different apps into new “combo” telemetry packets**
 - Combo packets reduce packet header overhead to help telemetry optimization
- **Combo packet content drivers**
 - Common desired downlink rate (fast, medium, slow) for each data item
 - Related functional data, for example science auxiliary data (instrument state, spacecraft attitude, etc.)
- **Combo packets can be defined at compile-time or runtime**
 - Runtime configuration supports in-orbit maintenance



System Design Process

1. Analyze downlink, configure DS and TO filter tables
2. If filter tables can't accommodate bandwidth constraints, then two options:
 - A. Redesign packets generated by apps
 - B. Use HK app to create new "combo" packets that will meet bandwidth constraints

**Typically, this means downlink to the ground, but it could be to any external system and Telemetry Output app may be replaced another app such as SB Network



**Typically, from Scheduler App but not required to be

- Copy table defines source packets sections from any app that are copied into a new “combo” packet

```

hk_copy_table_entry_t      HK_CopyTable[HK_COPY_TABLE_ENTRIES] =
{
    /*          inputMid      inputOffset  outputMid      outputOffset  numBytes*/

    /*  0 */ { CFE_ES_HK_TLM_MID,    12,    HK_COMBINED_PKT1_MID,    12,        4, },
    /*  1 */ { CFE_EVS_HK_TLM_MID,   12,    HK_COMBINED_PKT1_MID,    16,        4, },
    /*  2 */ { CFE_SB_HK_TLM_MID,    12,    HK_COMBINED_PKT1_MID,    20,        4, },
    /*  3 */ { CFE_TBL_HK_TLM_MID,   12,    HK_COMBINED_PKT1_MID,    24,        4, },
    /*  4 */ { CFE_TIME_HK_TLM_MID,  12,    HK_COMBINED_PKT1_MID,    28,        4, },

    /*  5 */ { ISIM_TLM_HK_MID,      18,    HK_COMBINED_PKT2_MID,    12,        4, },
    /*  6 */ { F42_HK_TLM_MID,       22,    HK_COMBINED_PKT2_MID,    16,        2, },
    /*  7 */ { F42_CTRL_TLM_MID,     12,    HK_COMBINED_PKT2_MID,    18,       12, },
    /*  8 */ { F42_CTRL_TLM_MID,     36,    HK_COMBINED_PKT2_MID,    30,       28, },

```

- When copy table is loaded HK
 - Subscribes to each input MID
 - Allocates memory for output packets
- New combo packet generated when a “Request Combo Pkt” command is received
 - Usually sent by the scheduler app
 - Increment “missed data counter” If all input data not received when request received

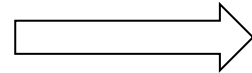


SimpleSat Default Configuration



CFE_ES_HK_TLM_MID,
CFE_EVS_HK_TLM_MID,
CFE_SB_HK_TLM_MID,
CFE_TBL_HK_TLM_MID,
CFE_TIME_HK_TLM_MID,

ISIM_TLM_HK_MID,
F42_HK_TLM_MID,
F42_CTRL_TLM_MID,
F42_CTRL_TLM_MID,





Runtime Dump-only Table

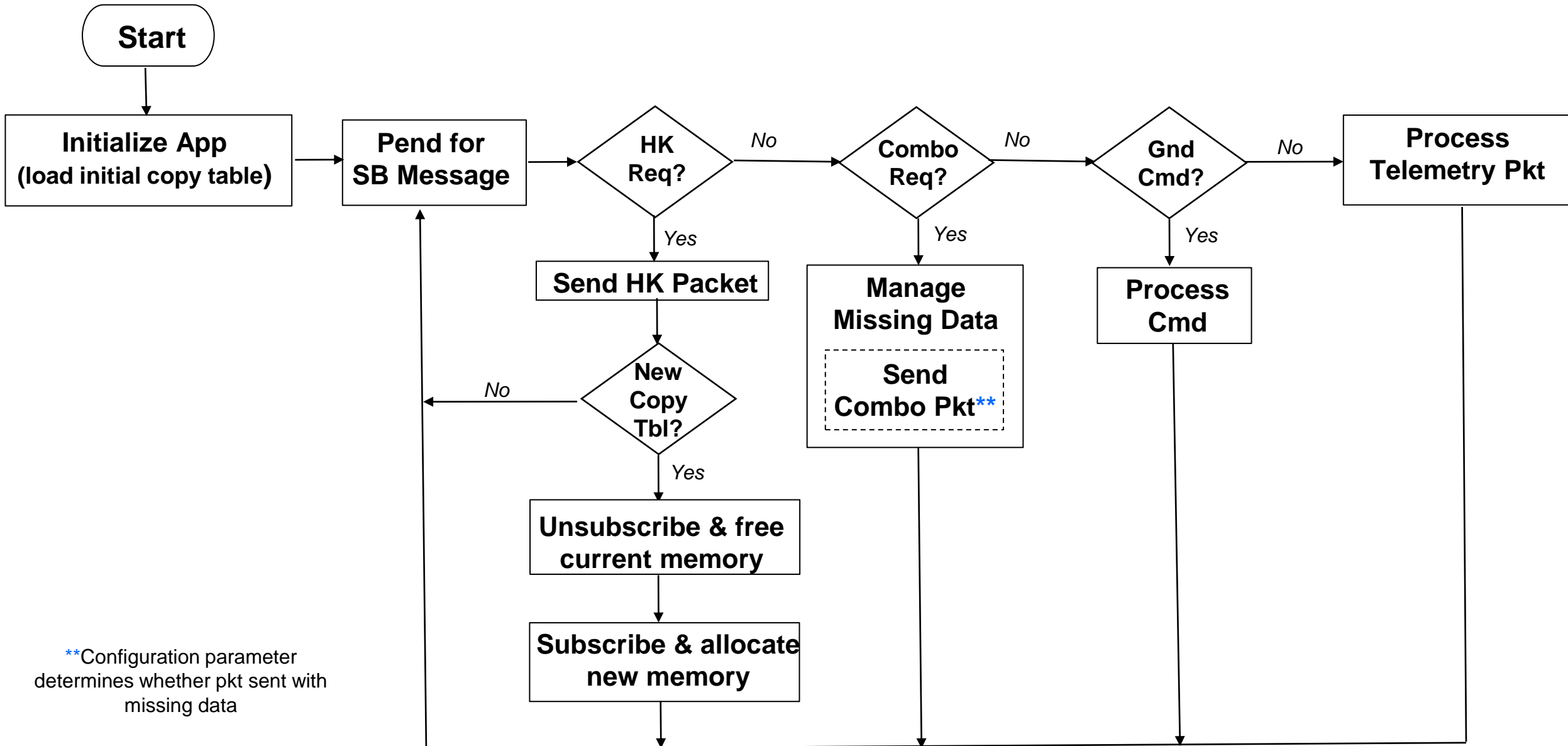


- TBD

InMid Subscribed	InMid Present	OutMid Address
Yes	No	0x42004589
Yes	No	0x42004589
Yes	No	0x420046FE
No	No	NULL

○
○
○

Yes	No	0x42004589
No	No	NULL
No	No	NULL
No	No	NULL



Data Structure Definitions	Description	External Dependencies
HK_COPY_TABLE_ENTRIES	Number of entries in the copy and runtime tables	Table Manger binary file definition

Definitions	Description
HK_PIPE_DEPTH	Depth of input pipe used for all HK input packets
HK_DISCARD_INCOMPLETE_COMBO	If true (1) only send combo packet upon request if all input data has been updated If false (0) always send combo packet when requested
HK_NUM_BYTES_IN_MEM_POOL	Maximum total size of all combo packets. Note 168 bytes is needed for memory pool overhead.

Parameter	Description	Default Value
Pipe Depth	Depth of HK command pipe	40
# of Copy Table Entries	Number of elements in the HK copy table to process	128
# bytes in memory pool	Number of bytes to allocate in the HK memory 6144 pool (needed for the HK output packets)	6144
Default HK Copy Table Name	--	CopyTable
Default HK Runtime Table Name	--	RuntimeTable
Default HK Copy Table Filename	--	/cf/apps/hk_cpy_tbl.tbl
Mission Revision	Mission-level revision number	0



HK Commands



Command	Description
No-Op	Increments the HK Command Accepted Counter and sends an info event message
Reset Counters	Initializes the following counters to 0: <ul style="list-style-type: none">- Command Counter- Command Error Counter- Output Messages Sent- Missing Data Counter



HK Housekeeping Telemetry Message



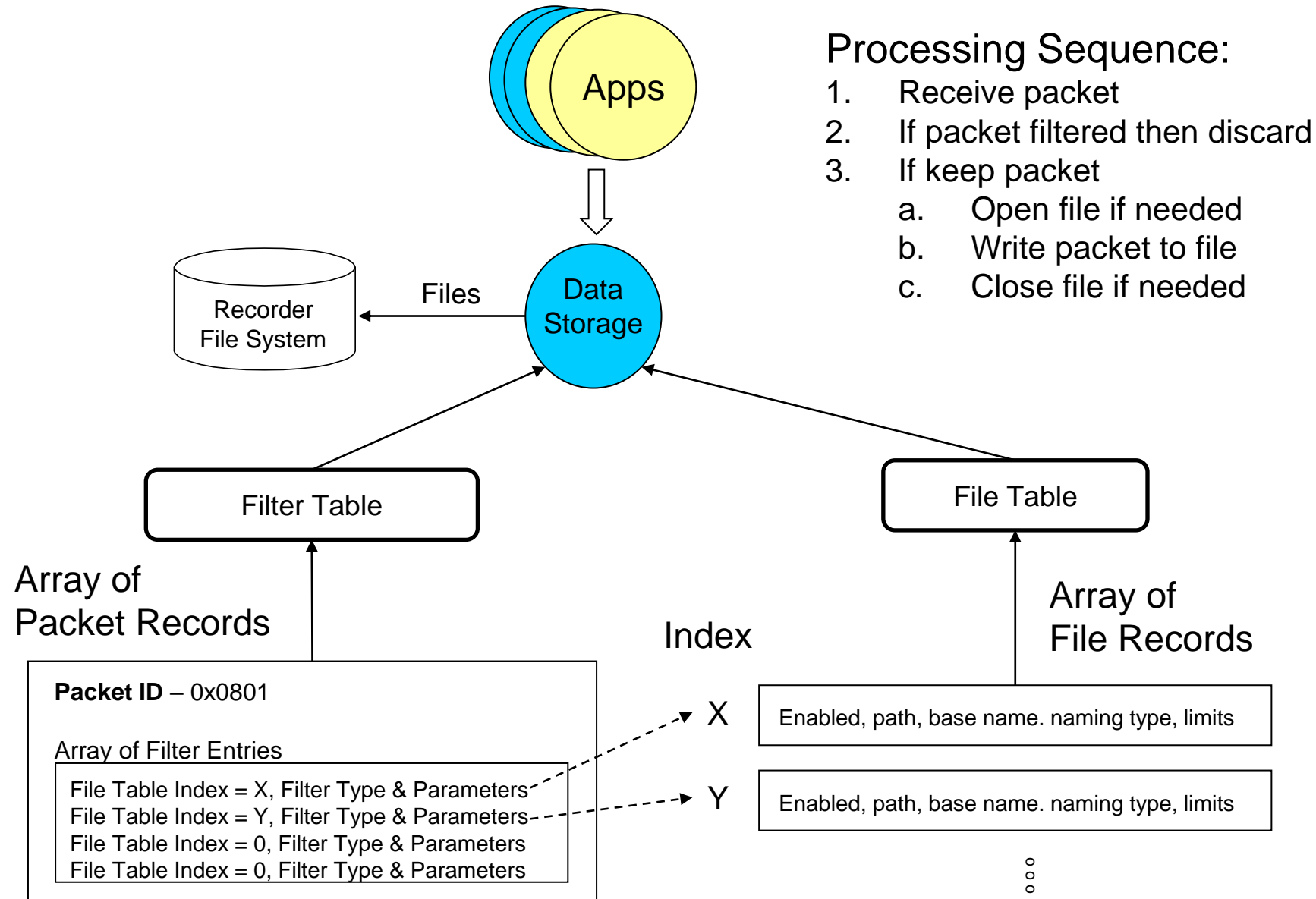
Telemetry Point	Description
Command Counter	Number of accepted commands
Command Error Counter	Number of rejected commands
Output Messages Sent	Number of output messages sent
Missing Data Counter	Number of times missing data from other apps was detected
Memory Pool Handle	Used to get memory pool statistics. The memory pools is used to allocate memory for output messages.



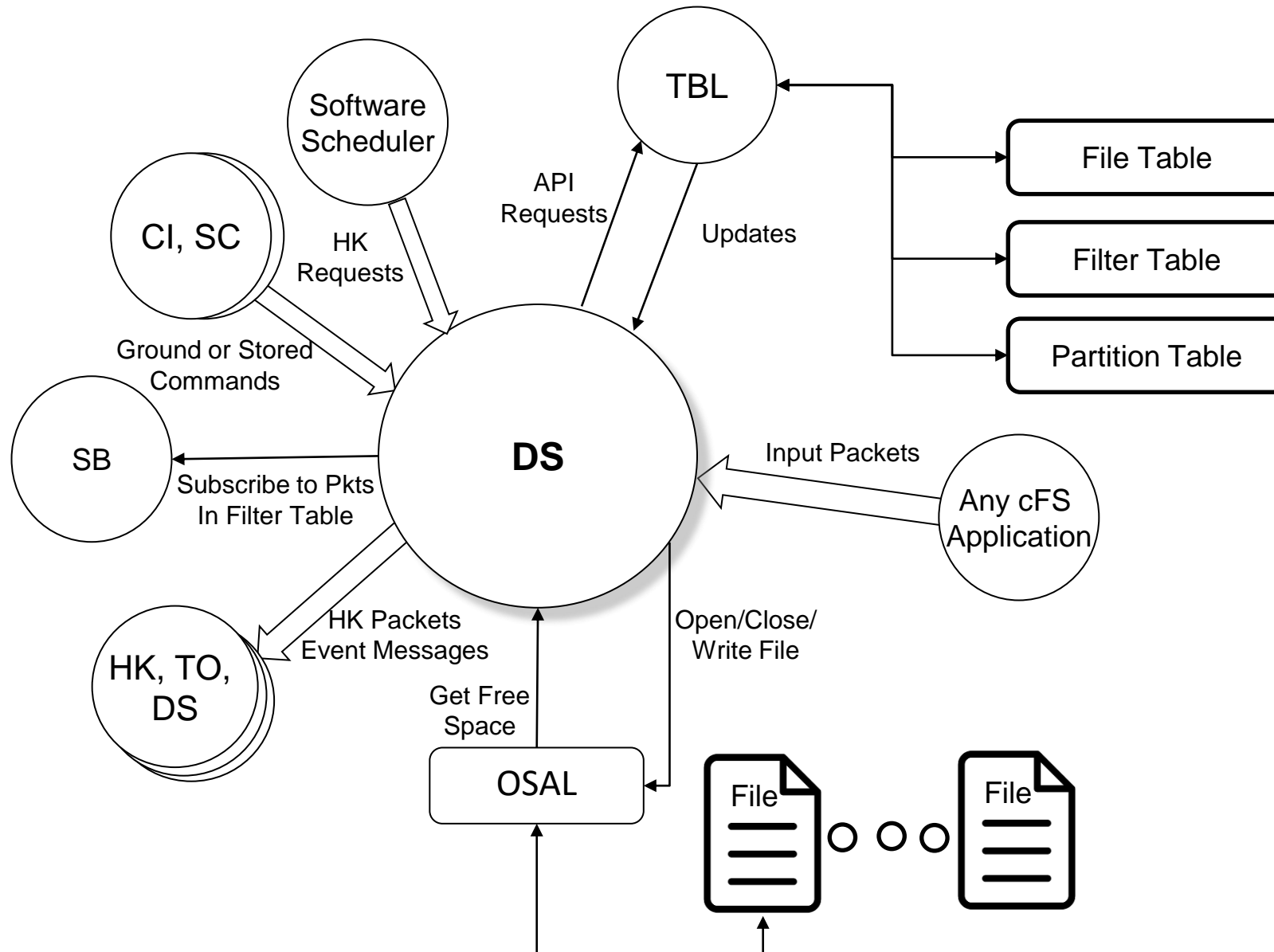
Data Storage

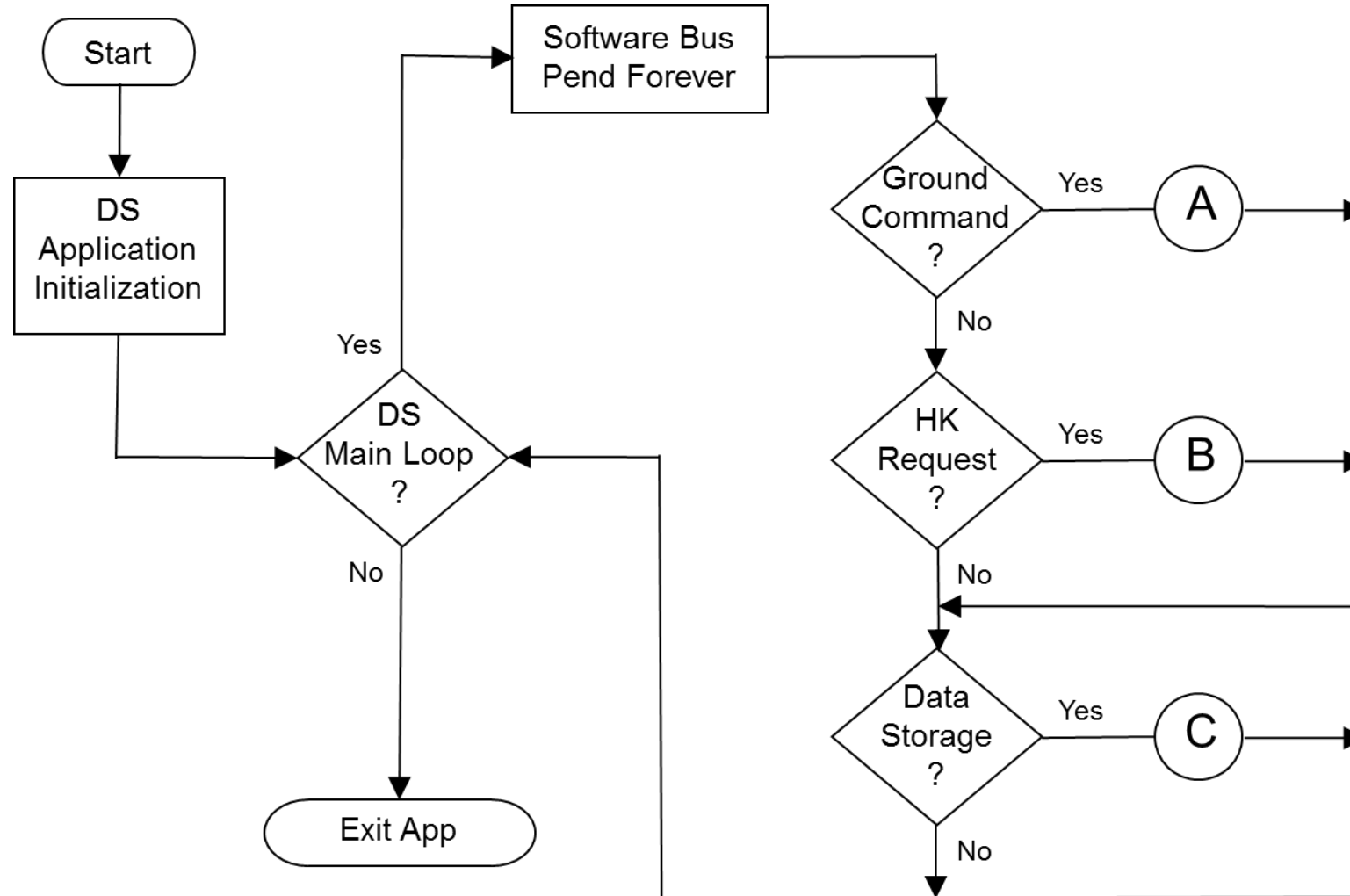
<https://github.com/nasa/DS>

- **DS receives messages from the software bus and writes them to files**
 - Messages to be stored in each file are specified in tables
 - provides time and sequence based filtering of message packet
 - File closure limit may be size-based or time-based (table defined)
- **DS uses two tables**
 - Filter Table - one entry per message id
 - File Table - one entry per file base name
- **DS has no download or playback capabilities**



DS Context Diagram







- **Time-Based Files:**

- Filename = Basename + YYYYDDHHMMSS + extension
- Time in filename is the time the file was created.
- Files are created when the first input pkt is received
- File Table tells DS how long (in seconds) the file should be open
 - File closed when time reached or reset occurs
- Next file created when next input pkt received



- **Size-Based Files:**
 - Filename = Basename + 8 Digit Sequence + extension
 - Sequence count in filename starts at zero after a power-on reset.
 - If input packet would cause file size to be > table defined “max file size”, then
 - Current open file is closed
 - File sequence count is incremented
 - New file created and input packet is written

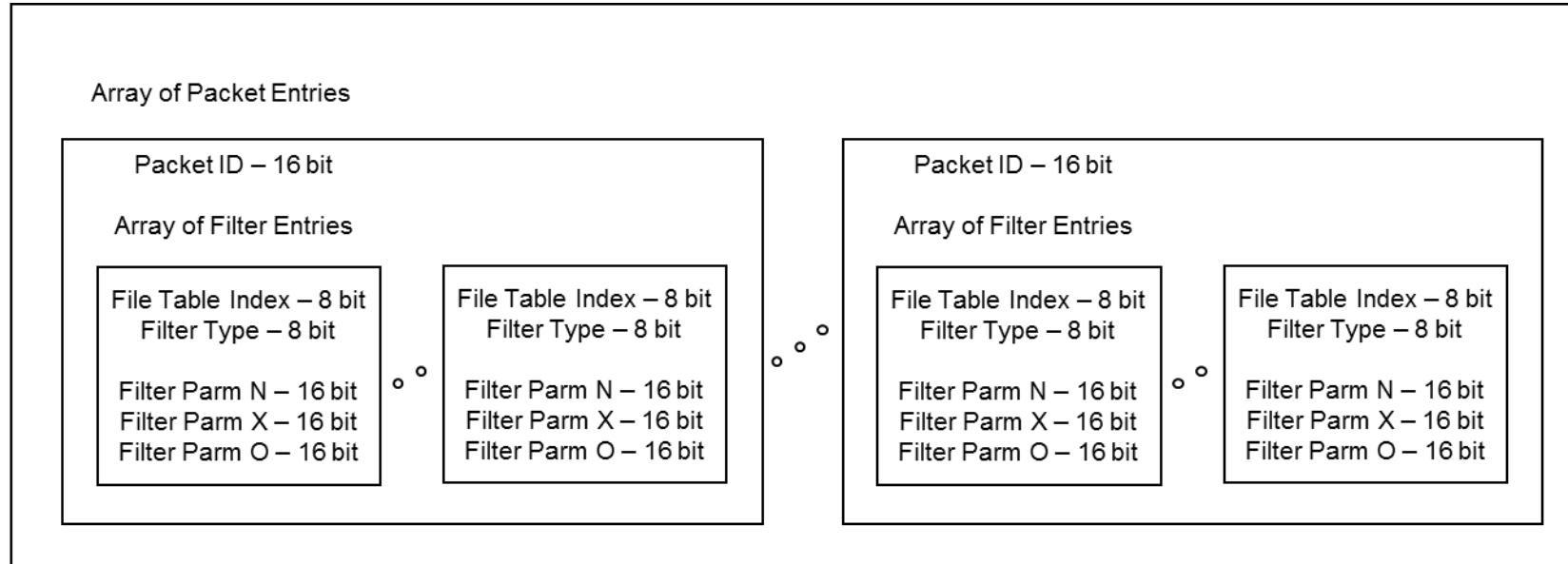


Table Size Configuration Parameters:

DS_PACKETS_IN_FILTER_TABLE: default value = 256

DS_FILTERS_PER_PACKET: default value = 4

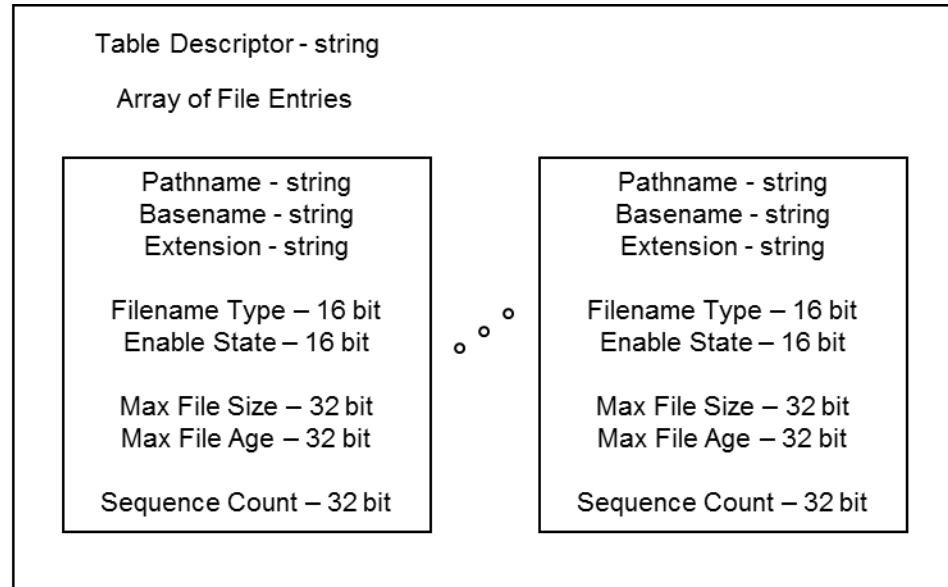
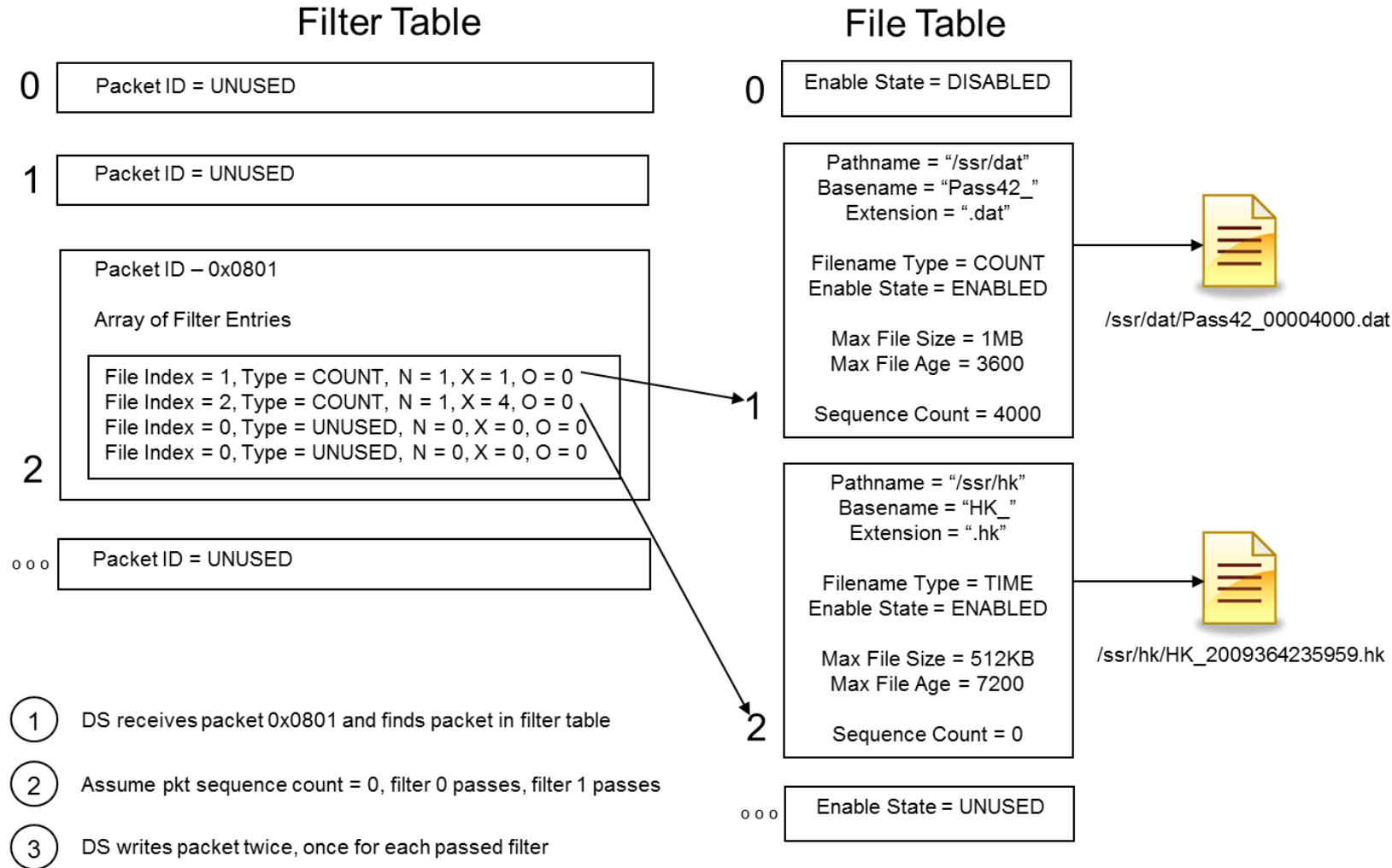


Table Size Configuration Parameters:

DS_DEST_FILE_CNT: default value = 16



Configuration Parameter	Description	Default Value
DS_DESTINATION_TBL_NAME	Logical name for the Destination File Table	"FILE_TBL"
DS_DEF_DEST_FILENAME	Default table filename — loaded at startup	"/cf/appsids_file_tbl.tbl"
DS_DEST_FILE_CNT	Number of file entries in Destination File Table	16
DS_PATHNAME_BUFSIZE	Size of pathname buffer in cmds, tlm, tables	OS_MAX_PATH_LEN (64)
DS_BASENAME_BUFSIZE	Size of basename buffer in cmds, tlm, tables	OS_MAX_PATH_LEN (64)
DS_EXTENSION_BUFSIZE	Size of extension buffer in cmds, tlm, tables	8
DS_FILTER_TBL_NAME	Logical name for the Packet Filter Table	"FILTER_TBL"
DS_DEF_FILTER_FILENAME	Default table filename — loaded at startup	"/cf/apps/ds_filter_tbl.tbl"
DS_PACKETS_IN_FILTER_TABLE	Number of packet entries in Packet Filter Table	256
DS_FILTERS_PER_PACKET	Number of filters per packet table entry	4

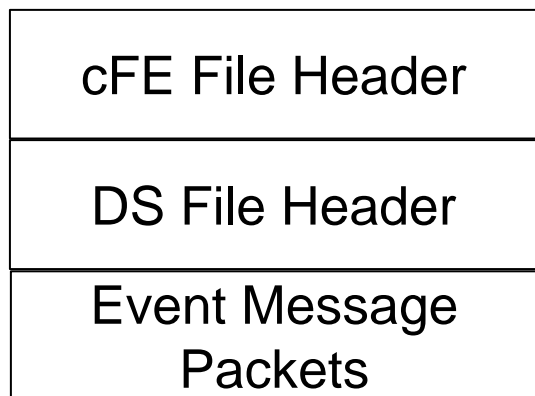
DS Configuration Parameters - 2

Configuration Parameter	Description	Default Value
DS_SEQUENCE_DIGITS	Number of digits in sequence portion of filename	8
DS_MAX_SEQUENCE_COUNT	Max filename sequence count before rollover	99999999
DS_TOTAL_FNAME_BUFSIZE	Size of buffer to contain fully qualified filename	OS_MAX_PATH_LEN (64)
DS_FILE_HDR_SUBTYPE	Common cFE file header subtype identifier for DS files	12345
DS_FILE_HDR_DESCRIPTION	Descriptive text for DS file secondary header	"DS data storage file"
DS_FILE_MIN_SIZE_LIMIT	Smallest amount that may be set for file max size limit	1024 (bytes)
DS_FILE_MIN_AGE_LIMIT	Smallest amount that may be set for file max age limit	60 (seconds)
DS_APP_PIPE_NAME	Logical name for DS application input pipe	"DS _CMD _PIPE"
DS_APP_PIPE_DEPTH	Size of DS application input pipe	256 (packets)
DS_MAKE_TABLES_CRITICAL	If "1", cFE Table Services will store DS tables in CDS	0
DS_SECS_PER_HK_CYCLE	DS measures file age by counting HK cycles	5 (seconds)

Command	Description
No-op	General DS aliveness test — verifies command handler and event generation
Reset Counters	Reset DS application housekeeping telemetry counters
Set Enable State For Packet Processor	Set enable/disable state for data storage packet processor
Set Destination File For Packet Filter	Modify packet filter table entry — set destination file
Set Filter Type For Packet Filter	Modify packet filter table entry — set filter type (sequence count vs time)
Set Filter Parms For Packet Filter	Modify packet filter table entry — set filter parms (N, X, 0)
Set Filename Type For Destination File	Modify destination file table entry — set filename type (sequence count vs time)
Set Enable State For Destination File	Modify destination file table entry — set enable/disable state
Set Path Portion of Destination Filename	Modify destination file table entry — set path portion of filename (string)
Set Base Portion of Destination Filename	Modify destination file table entry — set base portion of filename (string)
Set Extension Portion of Destination Filename	Modify destination file table entry — set extension portion of filename (string)
Set Max File Size For Destination File	Modify destination file table entry — set max file size (bytes)
Set Max File Age For Destination File	Modify destination file table entry — set max file age (seconds)
Set Filename Sequence Count For Destination File	Modify destination file table entry — set filename sequence counter value
Close Destination File	Close data storage file, file re-opened when next packet written to file

Telemetry Point	Description
Commands Accepted Counter	Number of successful ground commands (includes commands from on board sources)
Commands Rejected Counter	Number of commands with process errors
Disabled Packet Counter	Packets not processed because the packet processor was disabled
Ignored Packet Counter	Packets not processed because tables were not loaded or the packet was not in the filter table
Filtered Packet Counter	Packets processed that did not pass any filter tests (includes disabled destination files)
Passed Packet Counter	Packets processed that passed at least one filter test
File Write Counter	Total number of successful file writes
File Write Error Counter	Total number of file write errors
File Update Counter	Number of files with secondary header successfully updated prior to being closed
File Update Error Counter	Number of errors trying to update the secondary file header
Destination Table Load Counter	Number of times that cFE Table Services signaled new destination file table data
Destination Table Error Counter	Number of times that cFE Table Services signaled no destination file table data was available
Filter Table Load Counter	Number of times that cFE Table Services signaled new packet filter table data
Filter Table Error Counter	Number of times that cFE Table Services signaled no packet filter table data was available
Packet Processor Enable State	Current enable/disable state for the data storage packet processor
Array data (per destination file)	
-- File age, size, rate of growth	File age in seconds, file size in bytes, file growth rate in bytes per second
-- Filename sequence count	Filename sequence counter (this value will be used when the next file is created)
-- Enable state, open state, filename	File enable state, file open state, current filename (if open)

Example DS File Format Containing Event Messages



Each packet prepended with
CCSDS Telemetry Packet header

OpenSatKit/cfs/cfe/fsw/cfe-core/src/inc/cfe_fs_extern_typedefs.h

```

typedef struct
{
    uint32  ContentType;           /**< \brief Identifies the content type (= 'cFE1'=0x63464531) */
    uint32  SubType;              /**< \brief Type of \c ContentType, if necessary */
                                   /**< Standard SubType definitions can be found
                                   \link #CFE_FS_SubType_ES_ERLOG here \endlink */
    uint32  Length;               /**< \brief Length of primary header */
    uint32  SpacecraftID;         /**< \brief Spacecraft that generated the file */
    uint32  ProcessorID;         /**< \brief Processor that generated the file */
    uint32  ApplicationID;        /**< \brief Application that generated the file */

    uint32  TimeSeconds;          /**< \brief File creation timestamp (seconds) */
    uint32  TimeSubSeconds;       /**< \brief File creation timestamp (sub-seconds) */

    char     Description[CFE_FS_HDR_DESC_MAX_LEN]; /**< \brief File description */
} CFE_FS_Header_t;
  
```

OpenSatKit/cfs/cfe/fsw/cfe-core/src/inc/cfe_evs_msg.h

```

typedef struct {
    char     AppName[CFE_MISSION_MAX_API_LEN];
    uint16   EventID;
    uint16   EventType;
    uint32   SpacecraftID;
    uint32   ProcessorID;
} CFE_EVS_PacketID_t;

typedef struct {
    CFE_EVS_PacketID_t  PacketID;
    char                Message[CFE_MISSION_EVS_MAX_MESSAGE_LENGTH];
    uint8               Spare1;
    uint8               Spare2;
} CFE_EVS_LongEventTlm_Payload_t;
  
```

OpenSatKit/cfs/apps/ds/fsw/src/ds_file.h

```

typedef struct {
    uint32  CloseSeconds;
    uint32  CloseSubsecs;
    uint16  FileTableIndex;
    uint16  FileNameType;
    char     FileName[DS_TOTAL_FNAME_BUFSIZE];
} DS_FileHeader_t;
  
```

This file is parsed by Table Manager using the following file
cosmos/config/tools/table_manager/ds_event_log.txt

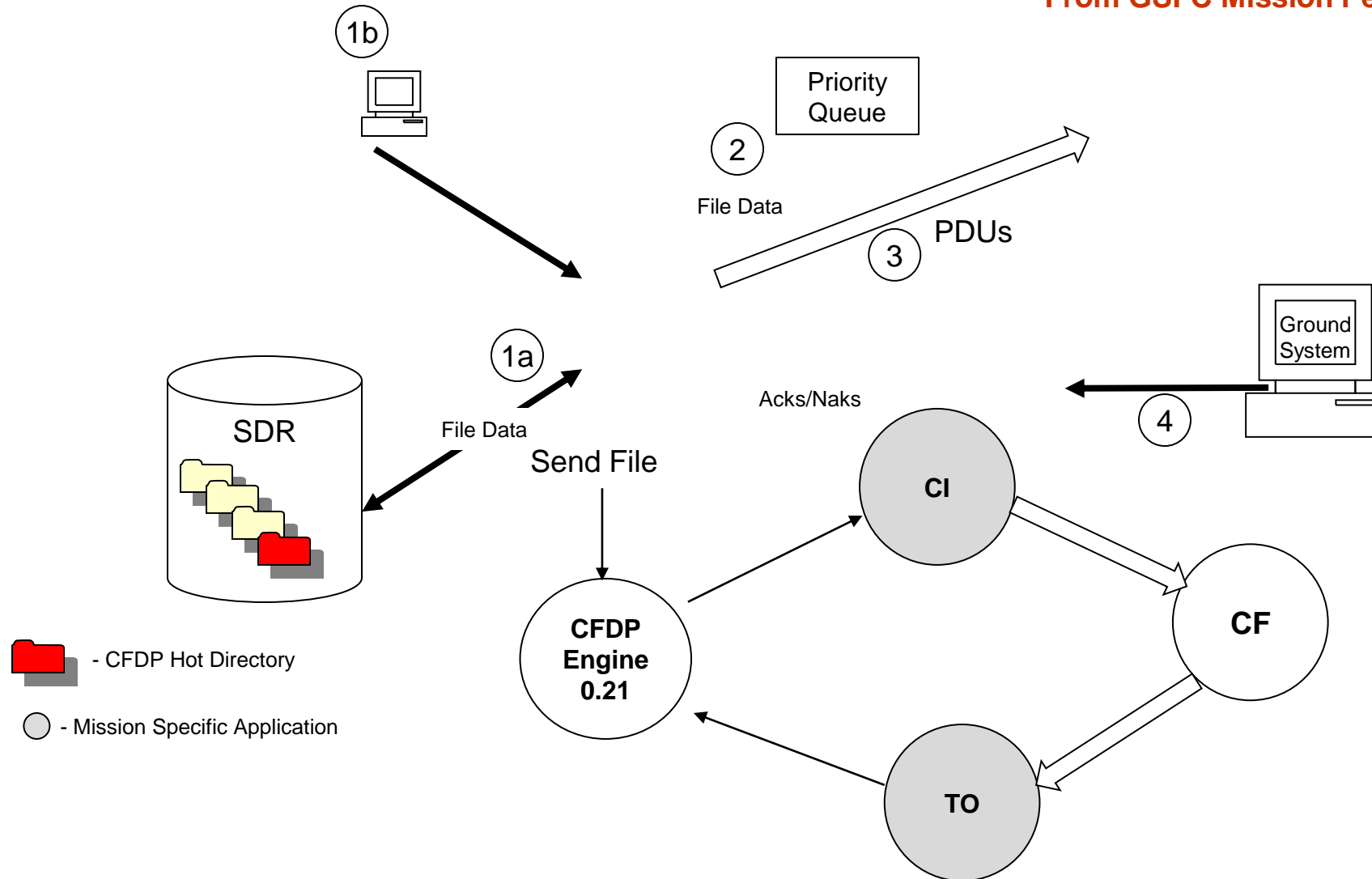


CCSDS File Delivery Protocol

<https://github.com/nasa/CF>



*** From GSFC Mission Perspective**

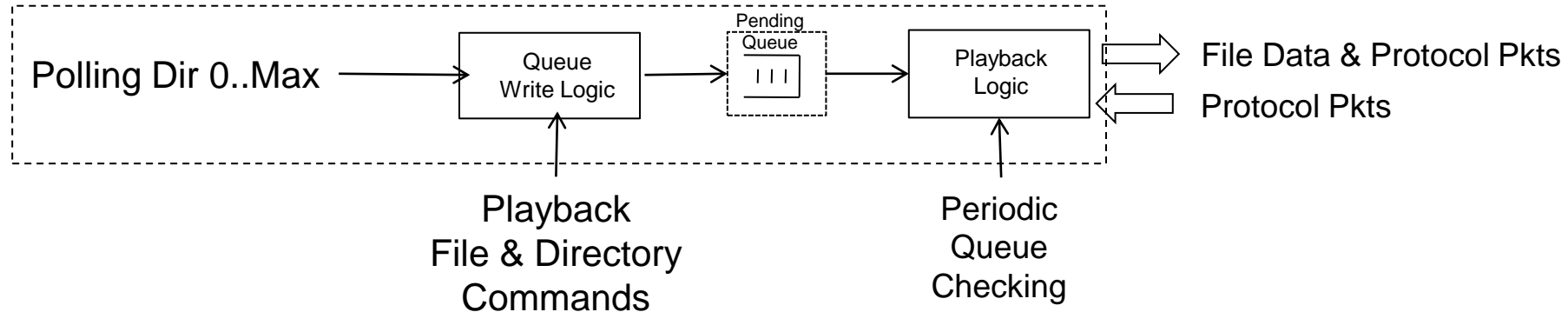




- **Transmits and receives files to and from the ground**
 - Typically interfaces to ground through Command Ingest and Telemetry Output applications
- **Implements the CCSDS File Delivery Protocol (CFDP) Class 1 (unreliable) and Class 2 (reliable)**
- **Files are processed sequentially**

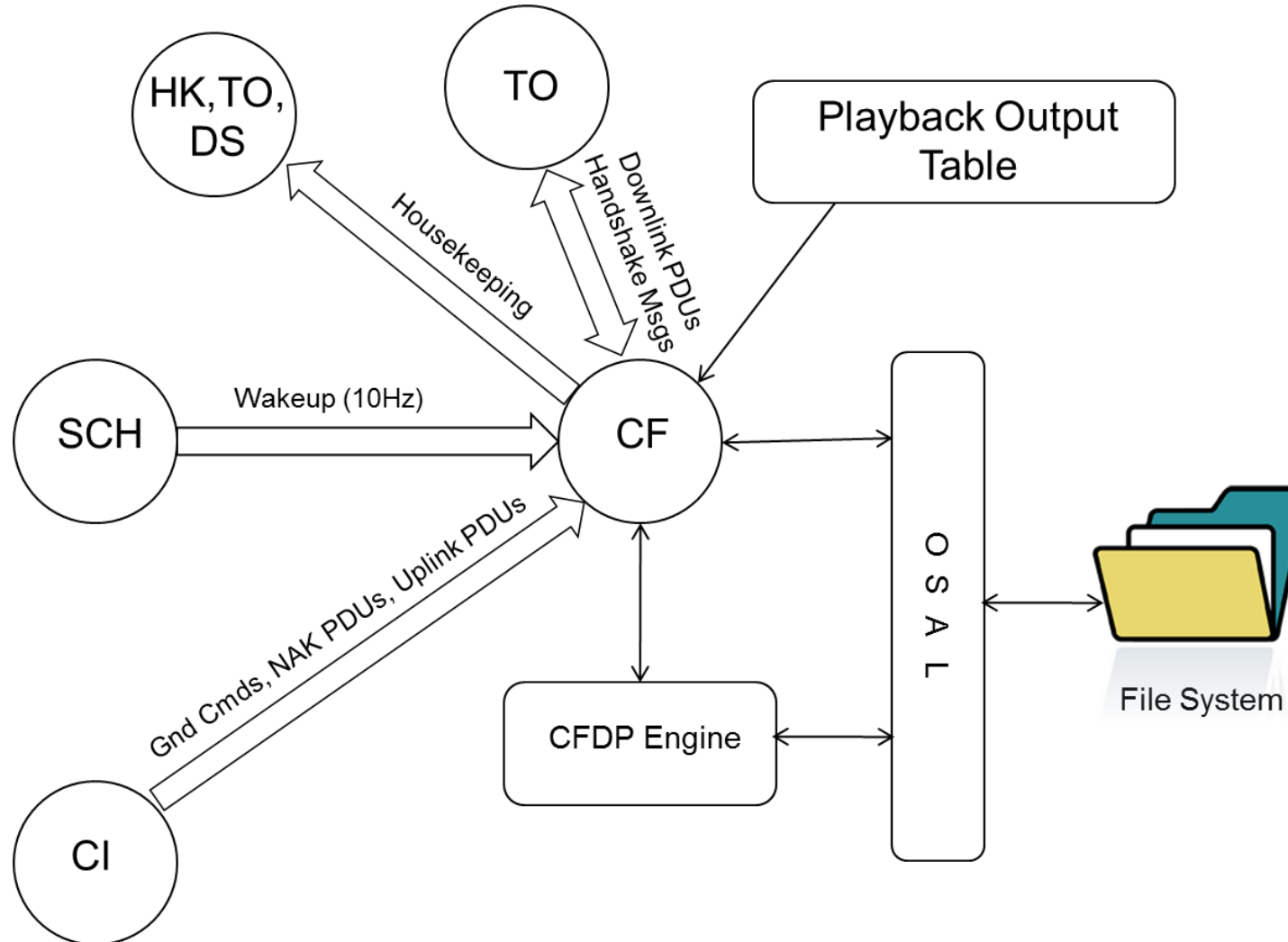
Playback Overview

CF Queue Processing - Duplicated For Every Channel



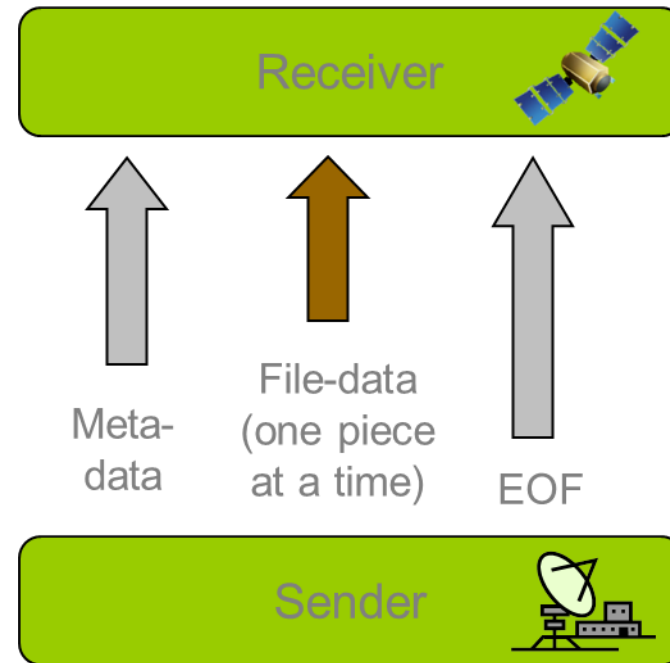
- **Files are queued for playback one of three ways**
 - Playback File command
 - Playback Directory command
 - By way of the Polling Directory
- **Playback pending queue has customized sorting.**
 - Each time a file (or group of files in the case of polling directories) is placed on the queue, the custom call-back routine is executed.
- **Each channel gets a configurable number of polling directories.**
 - When enabled, polling directories are polled for files periodically.
 - Any files found in a polling directory get placed on the playback pending queue immediately.

CF Context Diagram

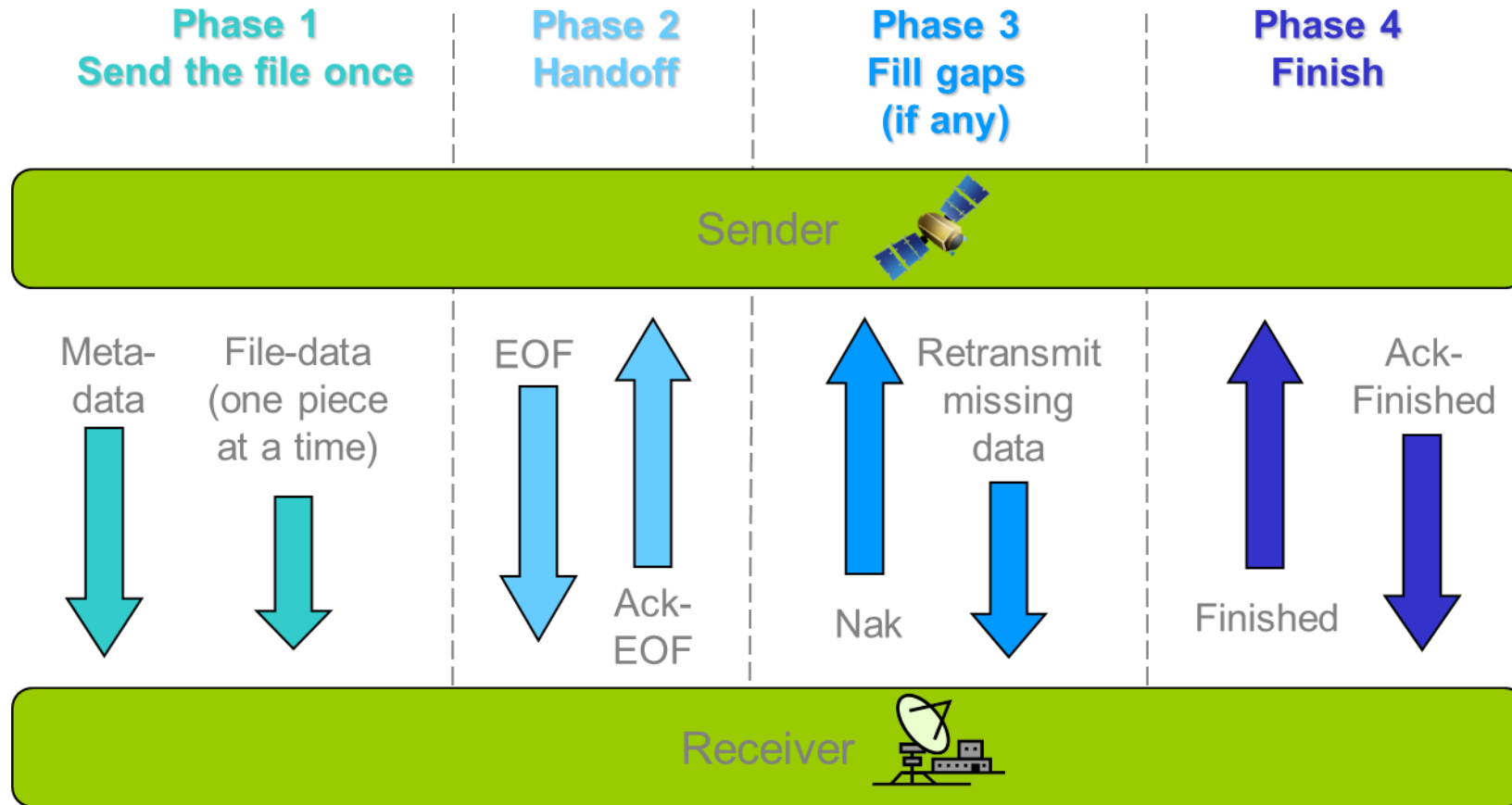




- **Provides compile-time and run-time configuration parameters**
- **Packages file data and protocol messages in PDUs as defined in CCSDS Blue Book CCSDS 727.0-B-4**
- **Capable of transmitting and receiving class 1(Unreliable) or class 2(Reliable) transfers**
- **Handles simultaneous transactions**
- **Provides “Indications” to inform the CF application of significant occurrences**
- **Receives “Put Requests” to start file transfer**
- **Verifies file checksum for class 1 and 2**

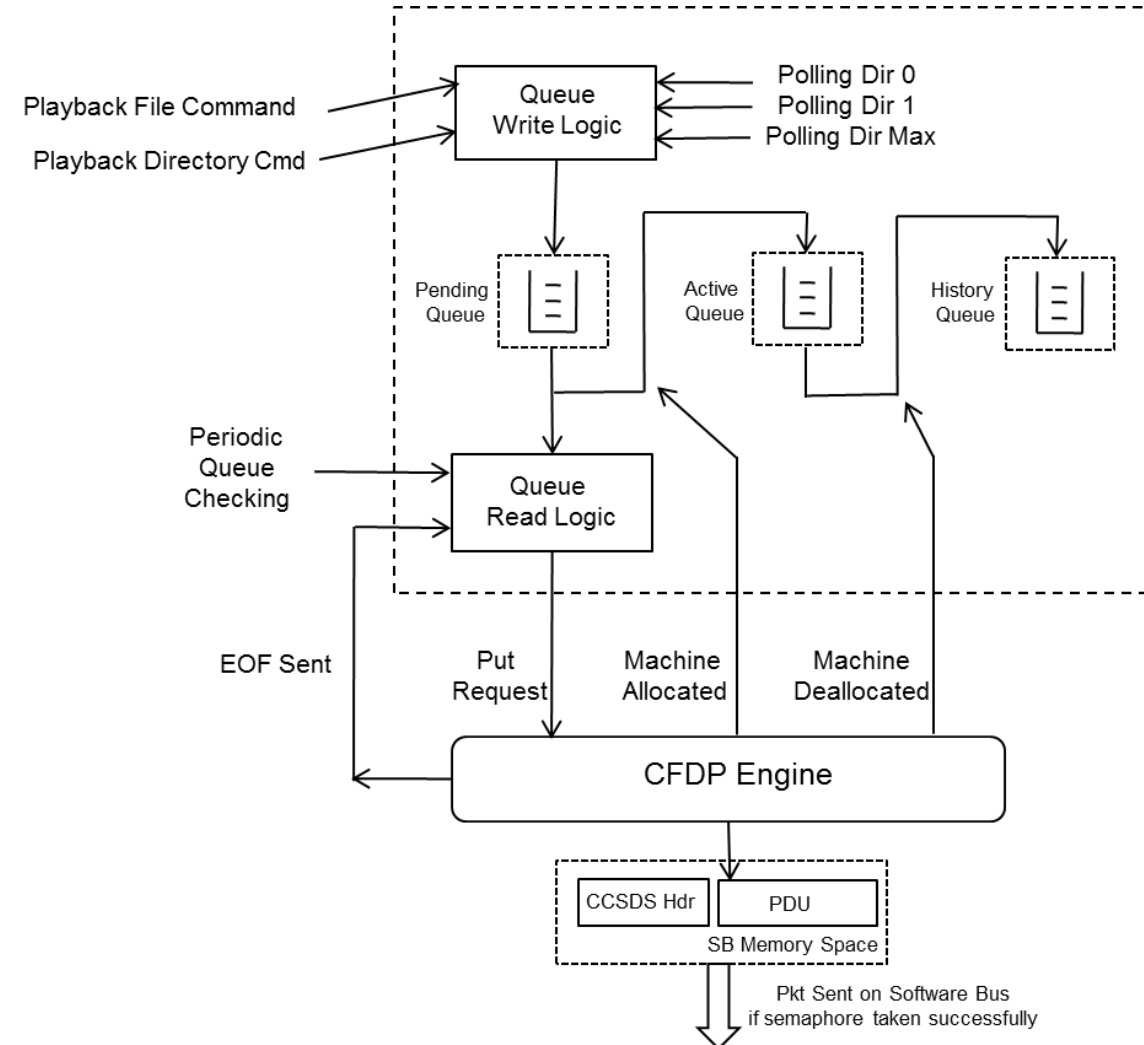


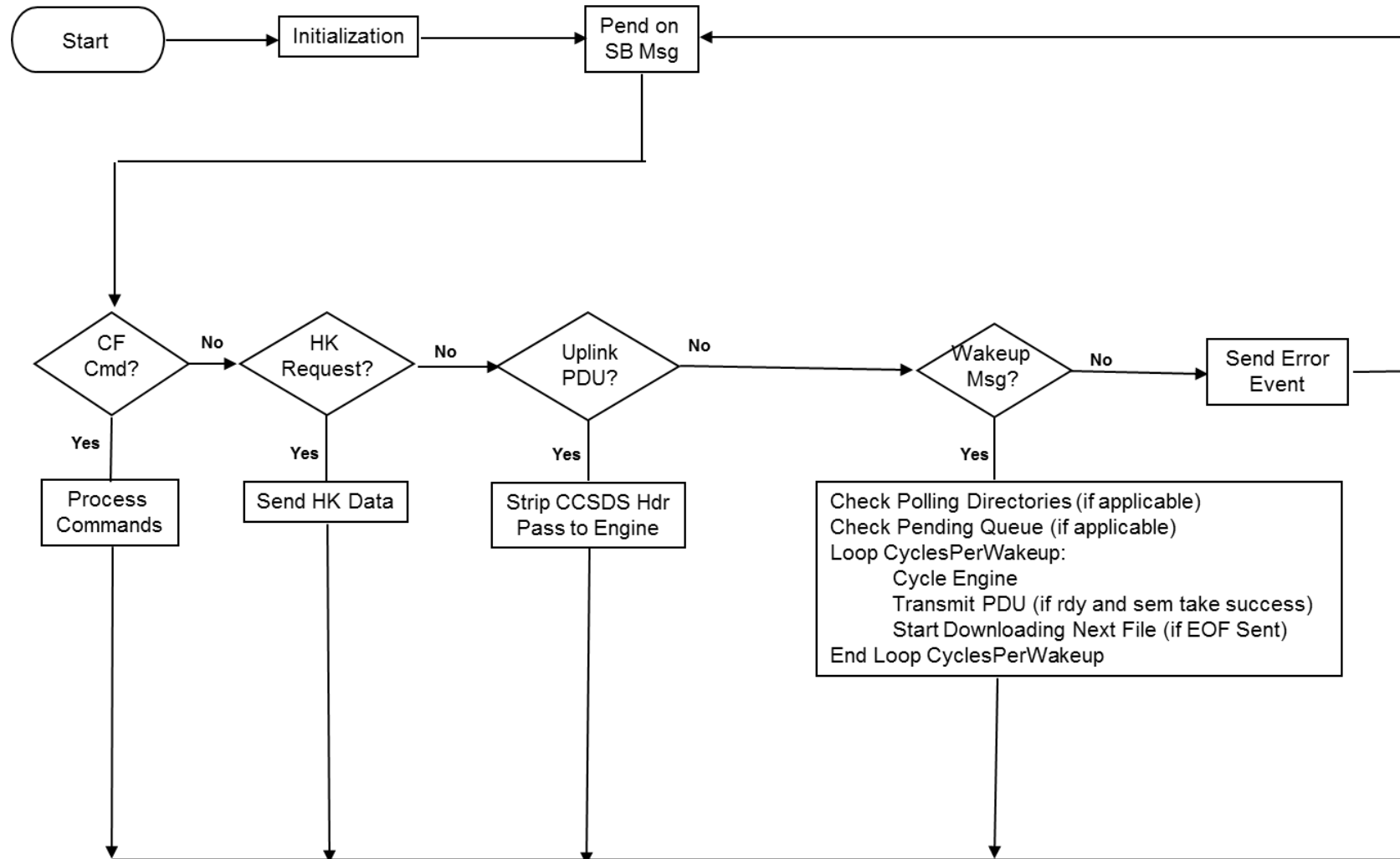
CF Class 2 Playback Example





- **Subscribes to Uplink PDUs. Msgld is specified in the configuration table**
 - CF application does not do much processing in the uplink direction. The engine does most of the work.
- **Receives PDUs wrapped in CCSDS packet from Command Ingest application (CI).**
- **Strips away the CCSDS header and passes the raw PDU to the engine.**
- **Keeps track of uplinked files through the use of an active queue and a history queue**







CF Compile Time Configuration



- **Max Simultaneous Transactions**
- **Pipe Name and Depth**
- **Max File Data in Playback PDU**
- **Max File Data in Uplink PDU**
- **Engine Temp File Prefix**
- **Configuration Table**
- **Configuration Table Filename**
- **Max Restricted Directories**
- **Max Playback Channels**
- **Max Polling Directories Per Channel**
- **Memory Pool Bytes**
- **Default Queue Info Filename**

Command	Description	Parameters
Noop	Increment cmd counter, display CF version number	None
Reset Counters	Reset one or all – cmd, fault, uplink, downlink counters	None
Playback File	Adds file to playback pending queue.	class, channel preserve, priority, SrcFilename, DstFilename
Playback Directory	Adds files in a given directory to playback pending queue.	class, channel preserve, priority, SrcDir, DstDir
Purge Queue	Purges the playback pending queue	channel
Write Queue	Writes the queue contents to a file.	type (uplink, playback), channel, queue, path/filename
Write Active Trans	Writes the transaction information (for all active transactions) to a file.	path/filename
Enable/Disable Dequeue	Enable/Disable Dequeue of playback pending queue.	channel
Dequeue Node	Dequeue a file on the pending or history queue.	type (uplink, playback), channel, queue, path/filename
Set Engine MIB Param	Set the engine configuration parameter specified in the command.	The configuration parameter to be set, value
Get Engine MIB Param	Display the given engine parameter in an event.	The configuration parameter to be displayed
Dump Config Params	Displays entire configuration contents via tlm packet. Includes run-time and compile-time.	None

Command	Description	Parameters
Suspend Transaction	Pauses timers and counters for a single (or all) transaction(s)	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename or "all"
Resume Transaction	Resume timers and counters for a single (or all) transaction(s)	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename or "all"
Cancel Transaction	Cancels a single (or all) transaction(s)	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename or "all"
Abandon Transaction	Abandons a single (or all) transactions(s)	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename or "all"
Freeze	Pauses timers and counters for all transactions	None
Thaw	Resumes timers and counters for all transactions	None
Enable/Disable Polling Directory	Enable or disable polling directory	Channel, poll, directory number (get number from config table)
Set Poll Directory Param	Change class, priority, preserve, SrcPath, or DstPath of given polling directory	Channel, poll directory number, class, priority, preserve, source pathname, destination pathname
Send Transaction Diag	Send diagnostic packet for a single transaction	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename



CF Commands – 3



Command	Description	Parameters
Kickstart	Start the transmission of the next file on the pending queue	Channel
Quick Status	Display high level status of the specified transaction	String formatted as SrcEntityId_TransSeqNum (ex.0.24_3) or filename
GiveTake	Adjust the handshake semaphore in the unexpected case that the semaphore value lost or gained a count when viewed during idle time	Channel
Auto Suspend Enable	enable or disable auto suspend mode	0 to disable and 1 to enable



CF Housekeeping Telemetry Message - 1



Telemetry Point	Description
Command Counter	Commands executed successfully
Command Error Counter	Commands that failed to execute
Memory In Use	Number of queue node bytes in use
Peak Memory In Use	Peak queue node bytes in use
Max Memory Needed	Memory needed if all queues were full
Memory Allocated	Memory allocated for queue nodes
Queue Nodes Allocated	Number of queue nodes currently allocated
Queue Nodes Released	Number of queue nodes returned to heap
Num Uplink PDUs Received	Number of uplink PDUs received
Num Files Uplinked Successfully	Number of uplink transactions succeeded
Num files failed uplink	Number of uplink transactions failed
Num uplink files in progress	Number of uplink transactions in progress
Last file uplinked	Filename of last file uplinked
Positive ACK Limit Counter	Number of ack timeout faults
Keep Alive Limit Counter	Number of keep alive faults
Invalid Transmission Mode Counter	Number of Inval transmission mode faults
FileStore Rejection Counter	Number of filestore rejection faults
File Checksum Failure Counter	Number of checksum failure faults
Filesize Error Counter	Number of filesize error faults



CF Housekeeping Telemetry Message - 2



Telemetry Point	Description
NAK Limit Counter	Number of NAK limit faults
Inactivity Counter	Number of inactivity faults
Invalid File Structure Counter	Number of invalid file structure faults
Suspend Request Counter	Number of suspend requests
Cancel Request Counter	Memory cancel requests
Flight Entity ID	Flight Entity ID
Frozen/Thawed Status	Transactions frozen or thawed
Machines Allocated by Engine	Number of machines allocated
Machines Deallocated	Number of machines deallocated
Frozen Partners	Any partners frozen — yes/no
Active Playback Files	Number of active playback files
Active Uplink Files	Number of active uplink files
Total Files Sent	Number of playback files sent
Total Files Received	Number of uplink files received
Total Transactions Frozen	Number of transactions frozen
Total Transactions Suspended	Number of transactions suspended
Total unsuccessful files sent	Number of unsuccessful playback files
Total unsuccessful files received	Number of unsuccessful uplink files



CF Housekeeping Telemetry Message - 3



Telemetry Point	Description
PDU's Sent	Number of PDUs sent
Files Sent	Number of files sent
Files Sent successfully	Number of files sent successfully
Files Sent unsuccessfully	Number of files sent unsuccessfully
Files on Pending Queue	Number of files on pending queue
Files on Active queue	Number of files on active queue
Files on History queue	Number of files on history queue
Naks Received	Number of NAKs received
Dequeue Enable State	Pending queue, dequeue state
Poll Directory Enable State	One flag for each polling directory
Red Light Counter	Number of times TO has denied the request to send a pdu
Green Light Counter	Number of times TO has accepted the request to send a pdu
Poll Directory Check Counter	Number of times poll directories checked
Pending Queue Checked Counter	Number of times pending queue checked



Packet Management Table Generation Tool

Data Storage
Telemetry Output

1. **Use a tool, typically a spread sheet, to analyze packet management strategies for different mission scenarios**
 - Determine which packets need to be stored for potential playback and which packets need to be downlinked in real-time and the rates for each of these activities
 - Iterative process that may involve redefining packets or using the housekeeping app to create new packets containing data specific for a particular mission scenario's needs
 - The analysis tool must support the generation of the table tool's input file requirements
2. **Prepare inputs for the table generation tool**
 - Export the analysis data to a Comma-Separated Value (CSV) text file
 - Define configuration parameters in the JSON input file to produce the desired table files
3. **Generate telemetry output and data storage tables**
 - Run the table generation tool
 - The table tool JSON configurations determine output file names and locations

- **For each packet (APID)**
 - Define SB subscription parameters: Priority, Reliability, Buffer Limit
 - Compute bits/sec based on packet production rate
 - Asynchronous packets are produced on demand so bandwidth must be allocated for asynchronous telemetry supported in each mode
 - Packet filtering parameters are defined for each mission mode/scenario

45

	32K Nominal	4K Safehold
Total Filter Table Bits/Sec (Sync):	13,518.0	3,558.2
Computed Telemetry Rate (Bits/sec):	13,522.4	3,559.3
Available Bandwidth (Bits/Sec):	32,000	4,000
Bandwidth Utilized (%):	42.3%	89.0%

KEYWORD_ROW	APID Name	APID (Dec)	APID (Hex)	SB Priority	SB Reliability	SB Buf Lim	Packet Size (Bytes)	Production Interval (sec)	Packet Type	Nominal Bits/Sec	Filter Type	Remainder Limit (N)	Filter (X)	Offset (O)	Effective Interval (sec)	Filtered Bits/Sec	Filter Type	Remainder Limit (N)	Filter (X)	Offset (O)	Effective Interval (sec)	Filtered Bits/Sec
KEYWORD_ROW	APID_NAME	APID		PRIORITY	RELIABILITY	BUF_LIM			PIPE		F1_TYPE	F1_N	F1_X	F1_O			F2_TYPE	F2_N	F2_X	F2_O		
5	CFE_TIME_HK_TLM_MID	5	0x0805	0	0	4	44	4.0	SYNC	88.0	COUNT	1	1	0	0.25	88.0	COUNT	1	4	0	0.00	22.0
6	CFE_TIME_DIAG_TLM_MID	6	0x0806	0	0	4	212	0.0	ASYN	0.0	COUNT	1	1	0	0.00	0.0		0	0	0	0.00	0.0
7	Spare	7	0x0807					0.0		0.0		0	0	0	0.00	0.0		0	0	0	0.00	0.0
8	CFE_EVS_EVENT_MSG_MID	8	0x0808	0	0	40	164	300.0	ASYN	4.4	COUNT	1	1	0	0.00	4.4	COUNT	1	4	0	0.00	1.1

- **For each packet (APID)**
 - Define SB subscription parameters: Priority, Reliability, Buffer Limit
 - Compute bits/sec based on packet production rate
 - Asynchronous packets are produced on demand so bandwidth must be allocated for asynchronous telemetry supported in each mode
 - Packet filtering parameters are defined for each mission mode/scenario

							ds_nom_filter							
							Unfiltered Generation Rate (kbps)							303.44
							Filtered Generation Rate (kbps)							11.83
							Target Generation Rate (kbps)							12.43
							Storage Full Time (Hours):							75.67
							Percent of goal (3 day goal = 72 Hours)							105.09%
	APID Name	APID (Dec)	APID (Hex)	Packet Size (Bytes)	Packet Interval (Hz)	Nom. Bits/Sec	Filter Type	N	X	O	Dest. File ID	Filtered Packet Interval (sec)	Filtered Bits/Sec	
KEYWORD_ROW	APID_NAME	APID					F1_TYPE	F1_N	F1_X	F1_O	F1_FILE			
1	CFE_EVS_HK_TLM_MID	1	0x0801	284	0.25	568.0	COUNT	0	1	0	1	0.00	0.0	
2	CFE_ES_HK_TLM_MID	2	0x0802	156	0.25	312.0	COUNT	0	1	0	1	0.00	0.0	
3	CFE_SB_HK_TLM_MID	3	0x0803	40	0.25	80.0	COUNT	0	1	0	1	0.00	0.0	



Prepare Data for Table Generation Tool



- **For each packet (APID)**
 - Define SB subscription parameters: Priority, Reliability, Buffer Limit
 - Compute bits/sec based on packet production rate
 - Asynchronous packets are produced on demand so bandwidth must be allocated for asynchronous telemetry supported in each mode
 - Packet filtering parameters are defined for each mission mode/scenario



Generate Tables



- **TBD**