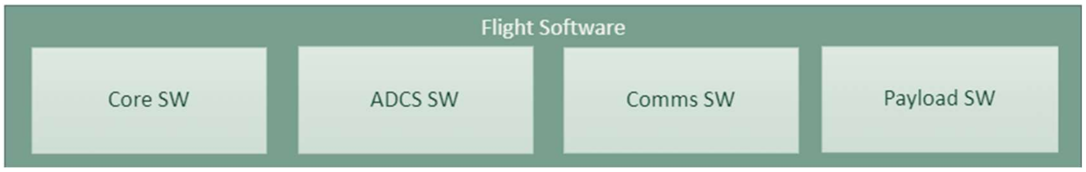
PowerSat FSW Definitions

(a nonexhaustive list)

Last Updated: 7/14/2022

PowerSat Specific FSW Nomenclature:



- **FSW**
  - flight software, all software files on the spacecraft: Core, ADCS, Comms, and Payload SW.
- **Core SW**
  - the files related to the health and status of the spacecraft bus
  - collects data from power and temperature sensors, monitors battery status
  - synchronizes the clock and checks flight images in memory for bit flips due to radiation
- **ADCS SW**
  - the files related to collecting telemetry form gyroscope, accelerometer, magnetometer, and GPS
- **Comms SW**
  - the files needed to uplink commands and downlink data (files, beacon, and telemetry)
  - process commands received from ground
- **Payload SW**
  - the files needed to manage the PowerSat specific payloads: Solar Array from DcubeD and accompanying PowerSat HIPO

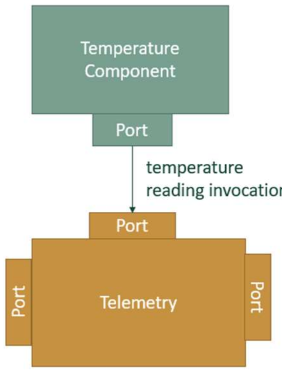
F Prime Definitions:

Additional definitions of F Prime constructs (ports, components, topology) can be found in the user's guide.

Additional definitions of F Prime data constructs (commands, events, telemetry) can be found in the user's guide

Additional definitions of F' provided components can be found here..

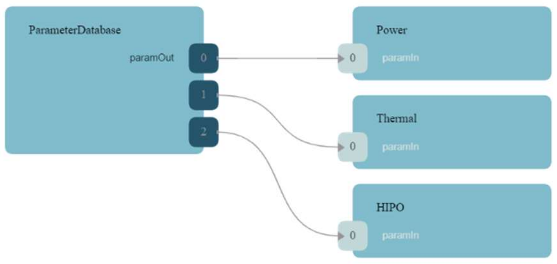
- **Component**
  - Each component is a discrete piece of the overall architecture and should have no code dependency on any other components.
    - **Passive**
      - A passive component has no thread and cannot support asynchronous commands or port invocations.
        - Ex: a temperature component that is activated every 10 seconds to read the temperature from all sensors and report to the Telemetry Database. A simple diagram is shown below.



- **Active**
  - A component has a thread of execution and a queue.
    - Ex: the Health component is always running to ping other components to ensure the processor is not stalled
- **Queued**
  - A queued component has a queue but no thread of execution. It can handle asynchronous commands and port invocations. (Rarely used)
    - Ex: The MathReceiver component from the Math Component Tutorial is given many tasks by MathSender. When activated, MathReceiver performs the math operation and reports the result.

- **Port**
  - The components are connected through ports which are defined prior to creation of the component. Ports have a type and a kind. A port's type is defined by the port invocations' data type. A port's kind defines its directionality (in/out) and can be guarded, and synchronous or asynchronous.
    - **Guarded**
      - The port is limited to a single invocation at a time.
        - Ex: A downlink component can an only downlink one file at a time so the port would be guarded.
    - **Synchronous**
      - The call directly invokes functions without using a queue.
        - Ex: The temperature sensor reads temperatures when activated.
    - **Asynchronous**
      - The call is placed in a queue and dispatched on a thread.
        - Ex: The MathReceiver component has an asynchronous input port where all the math jobs are queued.
- **Topology**

- The graph of interconnected components and ports developed to produce a full functioning software architecture.



- An example from the PowerSat FSW is shown above. The parameter database sends parameters that act as thresholds to the Power, Thermal and HIPO components. For further information about these custom components refer to PS-07-BBM_008 Custom Components Definitions. The numbers are an index of the port for that component. The names of the port connections are also given.

- **Event (EVR)**
  - A type of downlinked data that represents a single event in the system. Events represent the history of the system and are defined per-component. A component can report several events.
    - Ex: The temperature component reports an event each time it reads a temperature from a sensor.
- **Channel**
  - A single value of telemetry read and downlinked. Channels represent the current system state and are stored in the telemetry database.
    - Ex: The temperature component reads a value from a sensor, converts to degrees Celsius, then updates channel in telemetry database with new value.

- **Command**
  - Uplinked data items that instruct the system to perform an action. Commands are unique to an instance of a components. A component can have several commands.
    - Ex. A temperature component has a READ_TEMP command. However there are three different temperature components in your topology. Thus, there are three separate read temperature commands: TEMP_1_READ_TEMP, TEMP_2_READ_TEMP, and TEMP_3_READ_TEMP
  - Commands can be sequenced to automate the execution of many commands in a row.