


# SOIS Architecture and use of Electronic Data Sheets

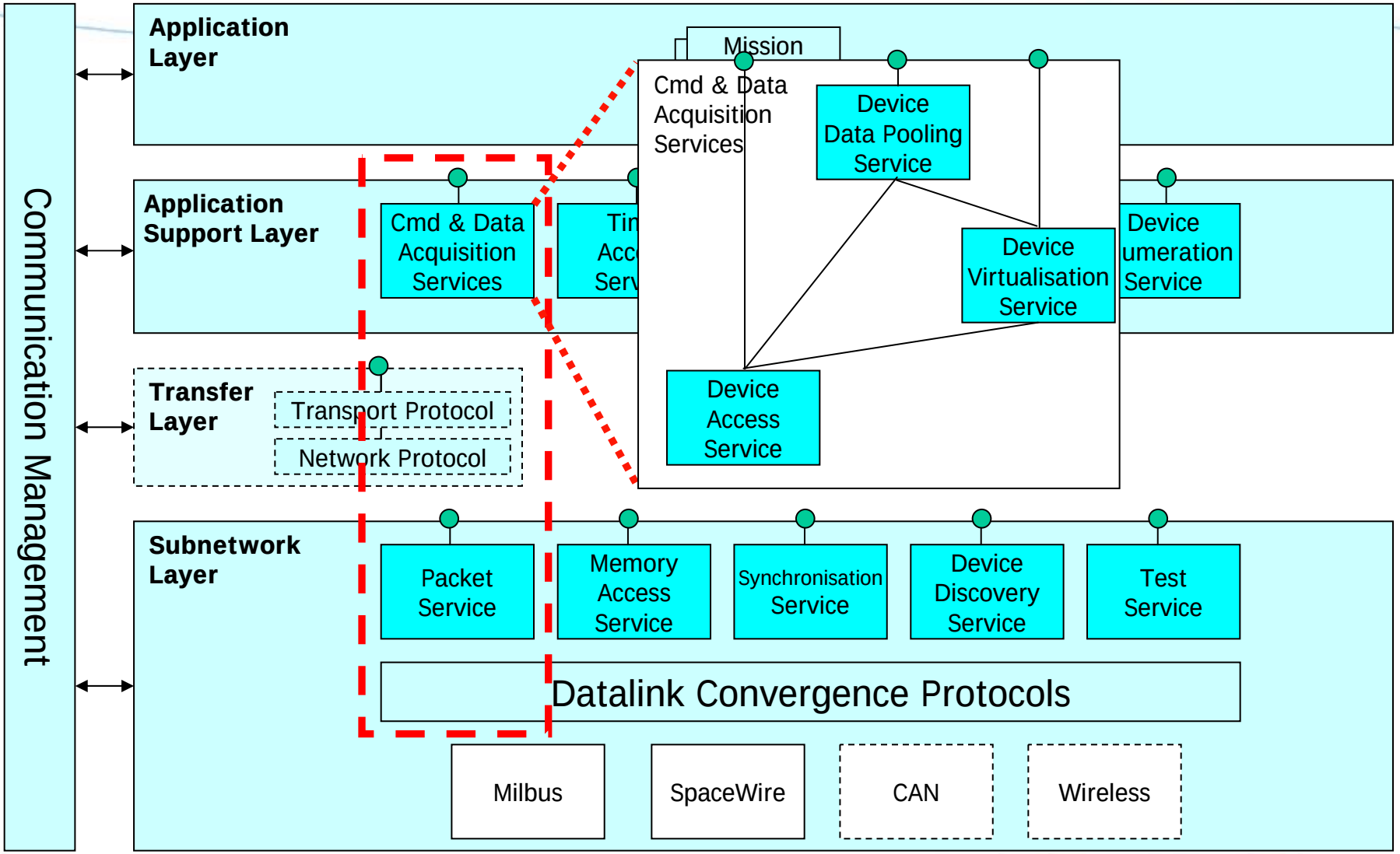
**Stuart Fowell, SciSys**

**ADCSS 2011, 25th October 2011**

# Overview

- Location in SOIS Architecture
  - SOIS Plug-and-Play
- Current approach to accessing Sensors/Actuators
- Device Virtualisation
  - Concept: Interfaces, Protocols and Services
  - Device Data Pooling
  - Worked Example
- Electronic Data Sheets
  - Concept
  - SOIS Plug-and-Play Architecture
  - CCSDS Approach to Standardisation
  - EDS Technologies
  - EDS Structure
  - Worked Example: Development Process with EDS
- Interactions with SOIS-SAFI
- Outlook

# Location in SOIS Architecture



# SOIS Plug-and-Play

- SOIS taking a broad definition of plug-and-play, encompassing design-time activities as well as “run-time” activities
- Goals
  - **Interoperability** – Application and device portability through isolation of the two, permitting flexibility and innovation in both
  - **Adaptivity** – Allow systems to adapt to change, probably more during the development process
  - **Rapidity** – Shorter development times, assisting design, implementation, integration and testing
- Use Cases
  - **Rapid Spacecraft Development** – Development Tool Chain
  - **Automated Integration & Test** – Adaptivity of EGSE
  - **Dynamic Fault Recovery** – Reconfiguration aspects of FDIR
  - **Dynamic Device Migration** – human and robotic missions
- Not talking about subnetwork device discovery today...

# Current Approach to accessing to Sensors/Actuators

- Information for accessing AOCS devices currently provided in ICDs and Basic Software Device Drivers
  - Paper document, not standardised form
- What is the impact of how its delivered today?
  - Little is standard
  - Not all information provided
  - Discrepancies between ICD and actual reality?
  - Difficult to check for discrepancies
- How much device-specific functionality is built into applications?
  - Device Drivers, Equipment Management software function
  - How much changes between missions?
    - Affecting devices, applications, device drivers etc
- Recognised by Primes as a problem
  - E.g. Astrium looking to use spacecraft databases as a start to auto-coding configuration of system
  - Not standardised across industry
- SAVOIR-SAIF addressing electrical interface but doesn't cover functional interfaces
- How are SOIS proposing to solve it?
  - **Device Virtualisation and Electronic Data Sheets**

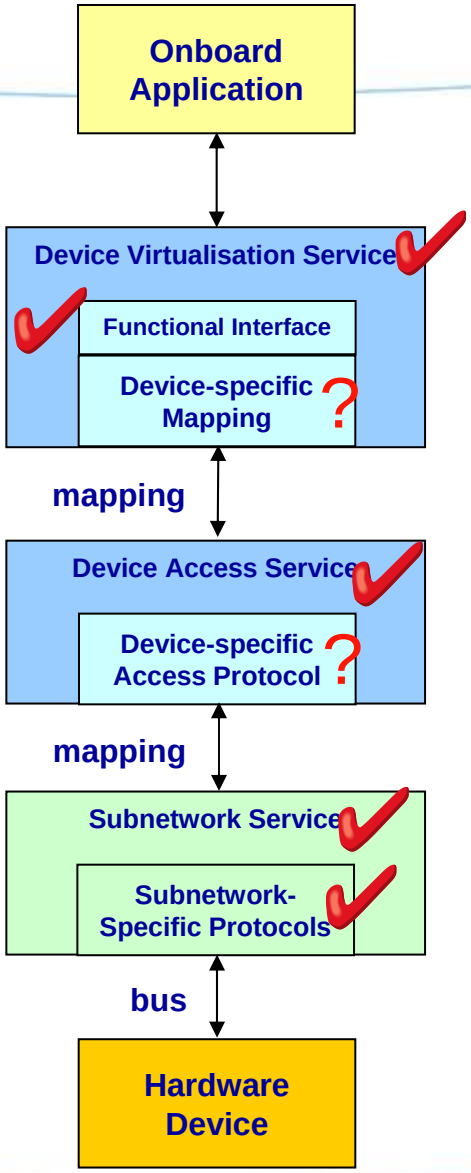
# Device Virtualisation – Concept

- What is Device Virtualisation?
  - Applications interface with devices at a **Functional Interface**
    - Mapped onto access protocols
    - Not directly using protocols or device representation of commands and data values
    - Aim is for single **standard Functional Interface per device type**

- What does Device Virtualisation do for you?
  - Swappable devices = reuseable applications
  - Standard access protocols = reuseable devices

- BUT only works if Functional Interface isolates Applications from device-specific characteristics
  - What if the differing characteristics of the different devices affect the algorithms of the applications?
  - Can they be input parameters to configure e.g. control loop?
  - Can't substitute for system engineering

# Interfaces, Protocols and Services

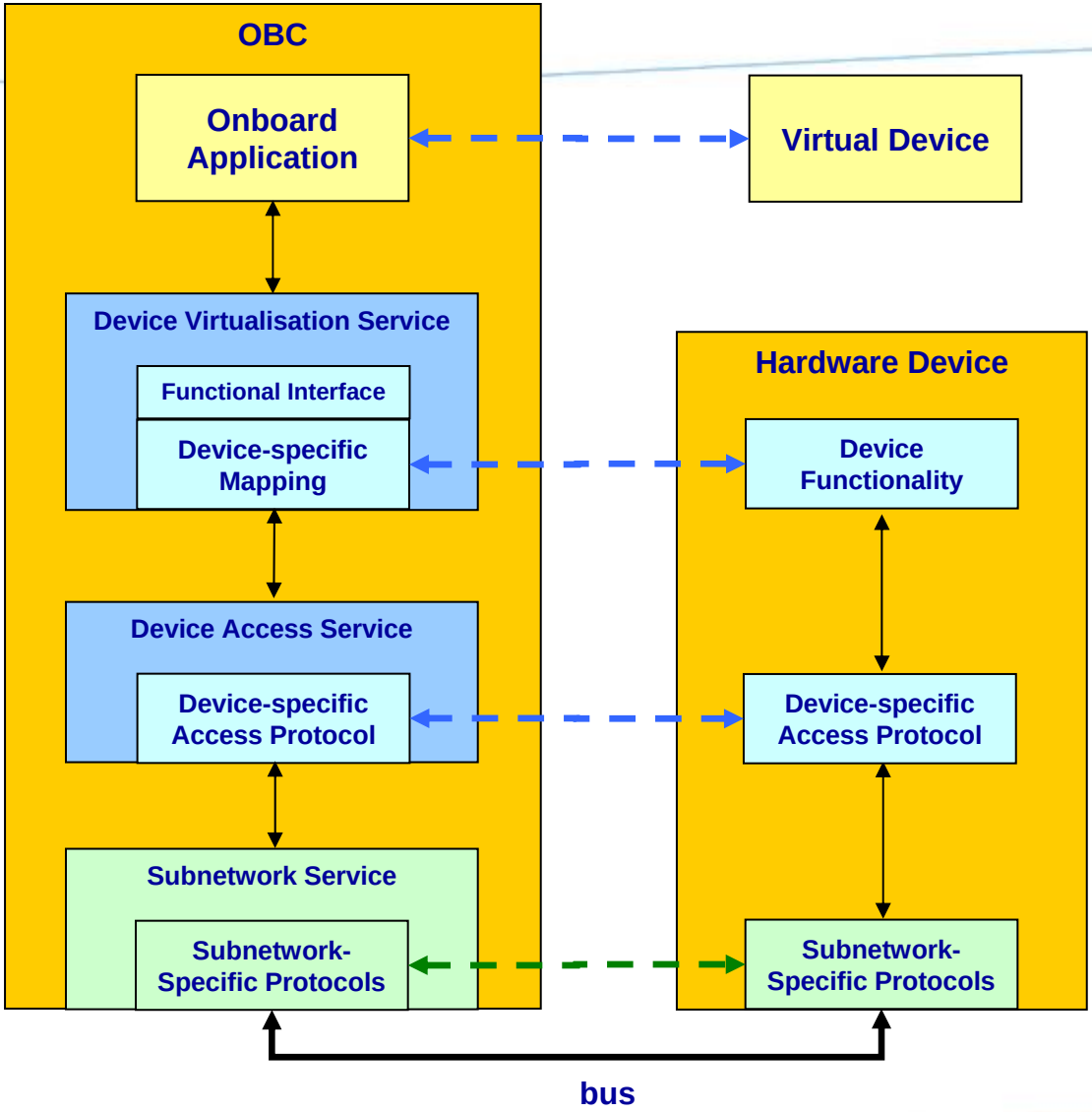



- Subnetwork-specific protocol
  - How to transfer data to/from device across subnetwork
  - QoS: ack, retransmit, priority etc
- Device-specific Access Protocol
  - How to command and acquire raw data for specific devices using subnetwork-specific protocols
  - State machine
- Device-specific Mapping
  - How the functional interface is mapped onto the device-specific access protocols
  - Type conversions, operations, state-machine
- SOIS Services
- Standardisation:
  - Trade-off what/where to standardise?
- Virtualisation adds some overhead
  - DAS can be directly accessed at a cost of tying application to particular device
  - If devices move towards directly supporting the functional interface, the mapping becomes NULL

# Device Data Pooling

- SOIS Device Data Pooling Service maintains an image of the states of a number of devices' values
- User applications access state of a device's value in data pool without having to generate explicit data acquisition request to actual device
  - History of samples maintained together with QoS associated with each sample
- DDPS periodically acquires data from devices at determined sampling rate or caches data from devices that asynchronously generate samples
  - Samples either raw (DAS) or functional interface (DVS)
- Avoids repetition of multiple users performing same acquisitions on devices
- Provides a synchronisation point for multiple acquisitions e.g. synchronised to underlying subnetwork using the SOIS Subnetwork Synchronisation Service
- What it doesn't provide
  - Software parameter data pool
  - PUS HKTM parameter packets

# Worked Example: Accessing a Virtualised Device



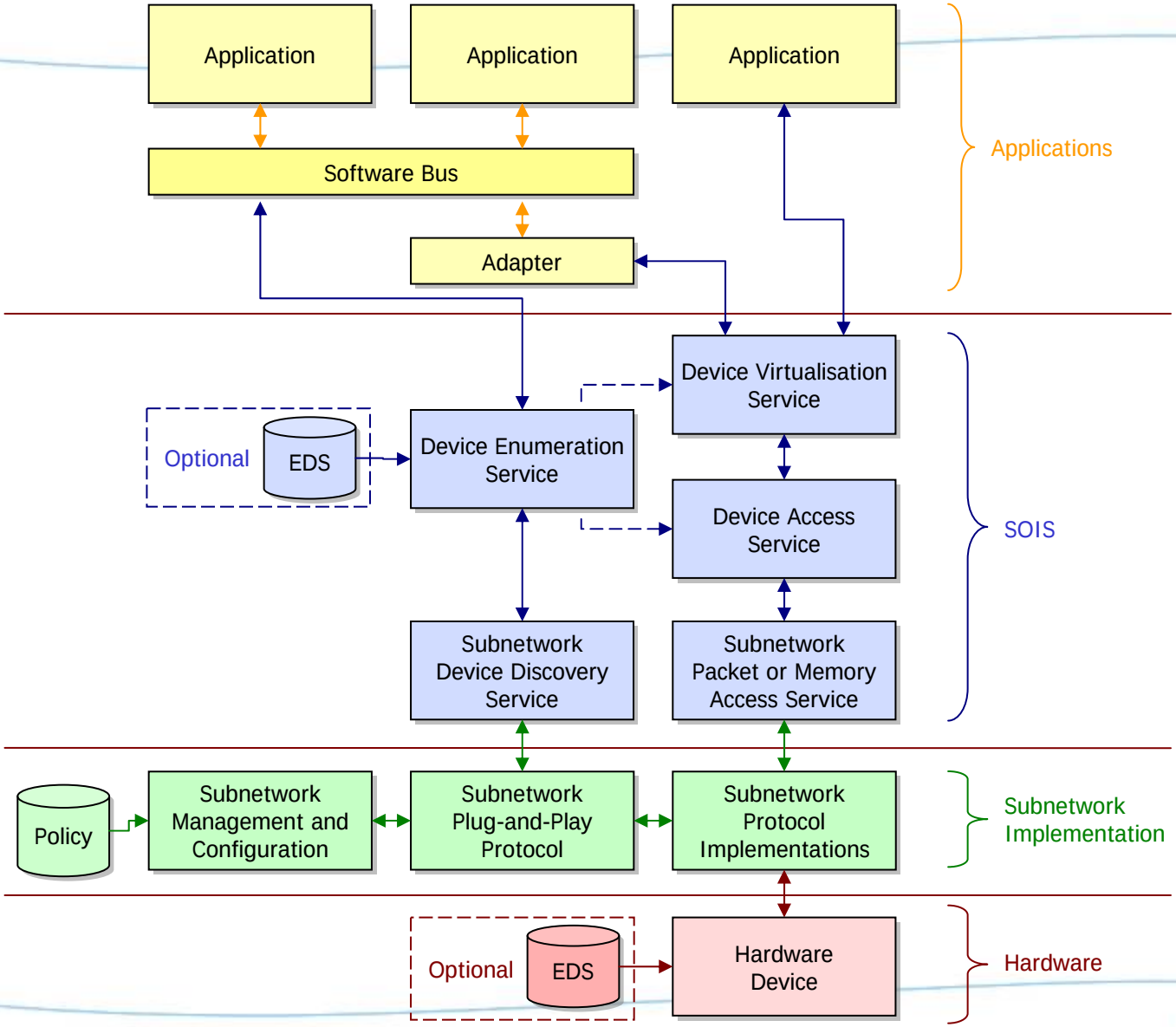
# Electronic Data Sheets

- What are Electronic Data Sheets (EDS)?
  - Electronic specification of information normally found in ICDs
    - But also covers generic Functional Interface

- Doesn't have to be used but provides:
  - Uniformity of information
  - Inputs for tools:
    - Automatic coding/configuration of device drivers (SOIS)
    - Simulation of devices
    - Test tool for devices
    - Automatic import into Spacecraft database
    - Etc
  - Online dynamic discovery of device capabilities – yeah, right! ☺
  - Enabler for standardisation of functional interfaces across device types

- What does EDS do for you?
  - System Engineer
    - Define requirements on interfaces
    - Allows conformance testing of devices to ensure they meet interfaces
    - Configuration support during integration of spacecraft
  - Application Developer
    - Allows conversion of functional interface into API, software model, etc
    - Automatic generation of device driver
  - Device Manufacturer
    - Drives testing of device interface
    - Avoids discrepancies between ICD and device, ICD and device driver

# SOIS Plug-and-Play Architecture



# CCSDS Approach

- CCSDS SOIS Area
  - ESA, NASA, UKSA & associates

- Looking to standardise EDS
  - EDS format standard
  - Extensible Dictionary of Terms standard
  - Template EDSs covering Functional Interfaces for device types (open issue)
  - Not a straightforward thing to do!

- What needs to be in an EDS?
  - Understand what we are defining
    - Structure, elements and referencing
      - Types, interfaces and state machines
  - Taking inputs from AIAA/AFRL SPA and SAVOIR-SAFI

- Currently
  - Defining use cases, deriving requirements
  - Evaluating existing EDS technologies
    - IEEE 1451 TEDS
    - AIAA/AFRL SPA xTEDS
    - CANopen
    - OMG/CCSDS XTCE
  - Establishing EDS structure & roadmap for prototyping/adopting technologies

# EDS Technologies (1/2)

- IEEE 1451 TEDS
  - Suite of standards which defines access to “smart transducers”
  - Each transducer connected to intermediate Transducer Interface Module (TIM) that acts as its representative to rest of the system
  - Complete TIM interface to a transducer defined by 1451.0 base specification TEDS and 1451.X network technology-specific TEDS
  - TEDS have variety of formats, inc. text, binary and (subset in) XML
  - Widely used in other domains, candidate for “spin-in”

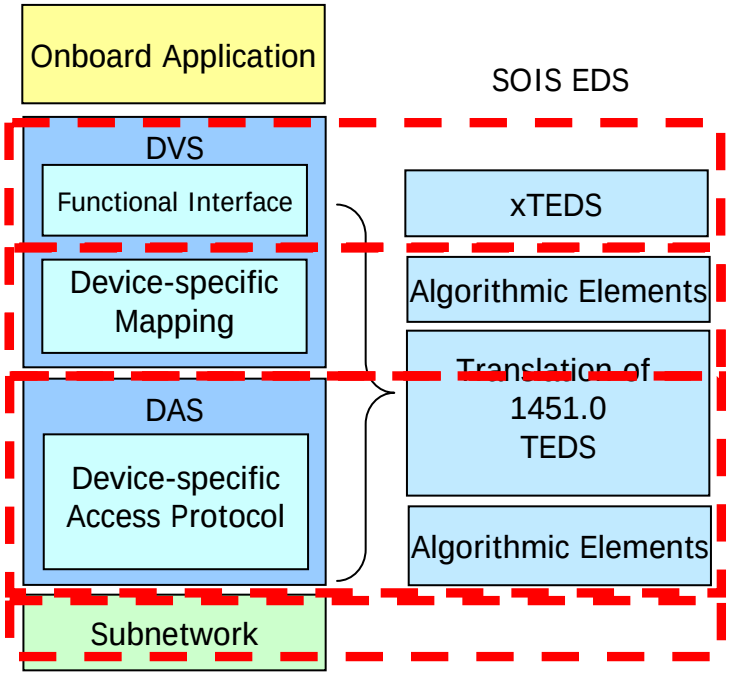
- AIAA/AFRL SPA xTEDS
  - Space Plug-and-Play Architecture (SPA) over-arching architecture for s/c and components for rapid assembly and integration, from AFRL for Operationally Responsive Space programme
  - Published in draft form by AIAA, covering architecture, electrical, coms and mechanical specs
  - Comms in SPA is message passing, with network-connected intelligent devices communicating in identical manner to software applications
    - Applications and devices communicate at same level & are effecitvely interchangeable
  - Message exchanges follow number of pre-defined patterns and transported over network by predefined protocols
  - Valid messages defined in an EDS that comes with product (software or device), may be stored with device or in central database of spacecraft
  - Common Data Dictionary defined semantic terms
  - Publish-subscribe
  - EDS inspired by IEEE 1451 TEDS and in XML format (xML) TEDS => xTEDS

# EDS Technologies (2/2)

- CANopen
  - Provides functionality equivalent to network & above OSI layers through basic CAN message
    - 8 data bytes + 11-bit identifier
    - Protocols of CANopen use identifier for device & function identifier
      - Service Data Object (SDO) protocol transfer of data objects between nodes through get/set iof objects in remote object dictionary in a client/server model
      - Process Data Object (PDO) protocol broadcast of data objects, mapped to object dictionary entries
  - Object Dictionary core to CANopen
    - Each entry has index, symbolic object name, descriptive name, data type, access rights and mandatory/optional flag
    - Basic and composite data types
  - Device Object Dictionary
    - Conform to base specification of CANopen
    - Device Profiles (fuller object dictionaries) specified for device types
    - Device conforms to one or more device profiles
  - Object Dictionary defined as an EDS, originally in text and now in XML (XDD)

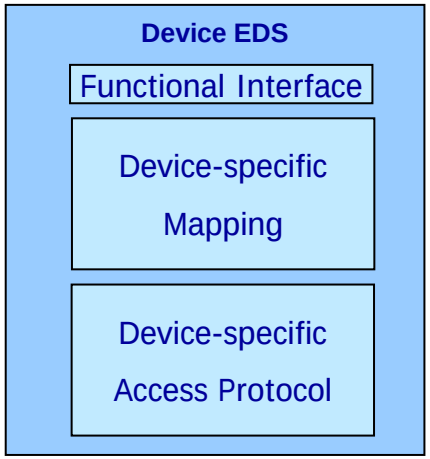
- OMG/CCSDS XTCE
  - XML Telemetric and Command Exchange (XTCE) defines communications data format and protocol in XML
  - Aimed at allowing information for monitoring & control of spacecraft to be defined in XTCE file
  - Includes:
    - variable packet formats, with elements conditional on packet contents
    - Simple processing/conversion algorithms or referencing external algorithm implementations
  - Not strictly a datasheet technology but has many interesting characteristics

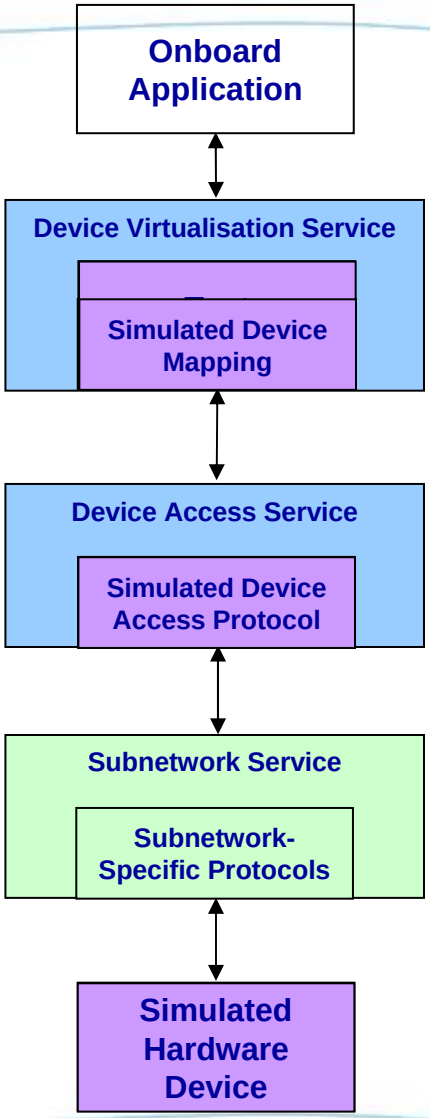
# EDS Structure




- EDS elements mapping to DVS and DAS data
  - Candidate technologies
  - Functional Interface
  - Device-specific Mapping
  - Device-specific Access Protocol
- Who provides/uses what?
  - System Engineer
  - Application Developer
  - Device Manufacturer
  - Device Driver Developer (no longer exists?)

# Worked Example: Development Process with EDS




1. System Specification
    1. Application, Bus and Device Functional Specification
    2. SOIS, Device Virtualisation and EDS Technology Selection
    3. Device Functional Interface Selection
2. Component Development:
    i. Device development & testing against EDS
    ii. Application development & testing against Functional Interface
3. Progressive integration:
    i. Auto-generation of device drivers & integration with device
    ii. Integration with application
4. Run-time

# Interactions with SAVOIR-SAFI

- ETSI SSDHI original attempt to define generic functional interface
  - Inspired by SOIS
- SOIS provides Framework
  - Specification of Functional Interface and mapping onto Device-specific Access Protocol for each device
    - Can be captured in an EDS
- SAVOIR-SAFI
  - Define ***generic*** Functional Interface for device types
    - Sensors: Start tracker, Gyro, Sun sensor
    - Actuators: Reaction wheel, Magneto torquer
  - Check mapping to generic Functional Interface to specific devices' Device-specific Access Protocols
- Allowing SOIS to provide
  - Specification of template EDS for device types

# Outlook

- Standard Electronic Data Sheets
  - Need to consolidate structure and candidate technologies
  - Prototyping leading to drafting of standard (2013?)

- Standard Functional Interface for device types
  - SAVOIR-SAFI output (2012?)
  - Associated SOIS EDS per device class (2013?)