Abstract

This document provides a guideline for spacecraft platform, subsystem, and component (including payload) developers for integrating plug-and-play characteristics into spacecraft structures, avionics, and hardware and software components to promote their rapid integration. This guideline will be used as the foundation for Space Plug-and-play Architecture (SPA) standards. It is a work in progress and will be updated periodically as the projected SPA standards evolve.
# Contents

Figures .................................................................................................................................................................. iv
Tables ................................................................................................................................................................... v
Foreword ............................................................................................................................................................... vi
Introduction .......................................................................................................................................................... viii

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scope</td>
<td>1</td>
</tr>
<tr>
<td>2. Tailoring</td>
<td>1</td>
</tr>
<tr>
<td>3. Applicable Documents</td>
<td>1</td>
</tr>
<tr>
<td>3.1 Normative References</td>
<td>1</td>
</tr>
<tr>
<td>4. Vocabulary</td>
<td>1</td>
</tr>
<tr>
<td>4.1 Acronyms and Abbreviated Terms</td>
<td>1</td>
</tr>
<tr>
<td>4.2 Terms and Definitions</td>
<td>3</td>
</tr>
<tr>
<td>5. Architecture</td>
<td>4</td>
</tr>
<tr>
<td>5.1 SPA Inside a General Space Architecture</td>
<td>4</td>
</tr>
<tr>
<td>5.2 SPA Inside</td>
<td>4</td>
</tr>
<tr>
<td>6.1 Primary SPA Goal: Eliminating Barriers to Rapid Satellite Deployment</td>
<td>7</td>
</tr>
<tr>
<td>6.2 SPA Core Concept and Essential Services</td>
<td>7</td>
</tr>
<tr>
<td>6.3 SPA Basic Capabilities</td>
<td>9</td>
</tr>
<tr>
<td>6.3.1 Standardized Physical Interfaces</td>
<td>9</td>
</tr>
<tr>
<td>6.3.2 Self-Organizing Networks</td>
<td>10</td>
</tr>
<tr>
<td>6.3.3 Communication Through Standard Messages</td>
<td>10</td>
</tr>
<tr>
<td>6.3.4 Query Services</td>
<td>10</td>
</tr>
<tr>
<td>6.3.5 SPA-X Interfaces</td>
<td>10</td>
</tr>
<tr>
<td>6.3.6 Self-Describing Components</td>
<td>10</td>
</tr>
<tr>
<td>7. SPA Implementation</td>
<td>10</td>
</tr>
<tr>
<td>7.1 Overview</td>
<td>10</td>
</tr>
<tr>
<td>7.2 Spacecraft Software Services From SPA</td>
<td>11</td>
</tr>
<tr>
<td>7.2.1 Configuration Caching</td>
<td>12</td>
</tr>
<tr>
<td>7.2.2 Configuration Modes</td>
<td>12</td>
</tr>
<tr>
<td>7.2.3 Adding or Updating Software</td>
<td>12</td>
</tr>
<tr>
<td>7.3 SPA Networking</td>
<td>12</td>
</tr>
<tr>
<td>7.4 SPA Components (Devices and Applications)</td>
<td>14</td>
</tr>
<tr>
<td>7.4.1 Component Capabilities</td>
<td>14</td>
</tr>
<tr>
<td>7.4.2 Non-SPA Components Made-To-Be SPA Compliant (Adding ASIMs)</td>
<td>14</td>
</tr>
<tr>
<td>8. Example SPA Implementation</td>
<td>16</td>
</tr>
</tbody>
</table>
8.1 Turning on a SPA System: What Happens When.............................................................................. 16
8.1.1 Register ................................................................................................................................. 16
8.1.2 Find........................................................................................................................................ 16
8.1.3 Access ..................................................................................................................................... 18
8.2 Panel Concept .............................................................................................................................. 20
9 SPA Tools ......................................................................................................................................... 21
9.1 Design Tools ................................................................................................................................. 21
9.1.1 Pushbutton Toolflow .............................................................................................................. 21
9.1.2 xTEDS Simulator .................................................................................................................... 21
9.1.3 SPA Device Part Numbers-ID for Use With Design Tools?................................................. 21
9.2 Test Tools ..................................................................................................................................... 23
9.2.1 Hardware-in-the-Loop Simulation and Test Bypass............................................................... 23
9.2.2 Flight Software-in-the-Loop.................................................................................................... 25
10 Summary of SPA Standards ........................................................................................................ 27
10.1 General SPA Standards............................................................................................................. 27
10.1.1 SPA System Capability Standard ....................................................................................... 27
10.1.2 SPA Ontology Standard ....................................................................................................... 27
10.1.3 SPA Logical Interface Standard ......................................................................................... 28
10.1.4 SPA Networking Standard .................................................................................................. 29
10.1.5 SPA Timing Standard ......................................................................................................... 30
10.1.6 SPA Physical Interface Standard ....................................................................................... 30
10.1.7 SPA Power Standard ......................................................................................................... 31
10.1.8 SPA Test Bypass Extension .............................................................................................. 31
10.2 SPA Application-Specific Standards ....................................................................................... 31
10.2.1 SPA Subnetwork Adaptation Standards (SPA-X) ............................................................ 31
10.2.2 SPA Subnetwork Adaptation Standard ............................................................................. 31
10.2.3 SPA-U Subnetwork Adaptation Standard ......................................................................... 32
Annex A Reference Architectures and Models .................................................................................. 33
A.1 The Data-Centric Plug-and-Play Approach.............................................................................. 33
A.2 CCSDS SOIS Model .................................................................................................................. 35

Figures
Figure 1 – Spacecraft bus architecture (systems view) with SPA overlay.............................................. 6
Figure 2 – SPA core concepts.............................................................................................................. 7
Figure 3 – The SPA Logical Model.................................................................................................... 9
Figure 4 – The SPA Services Manager............................................................................................ 12
Figure 5 – A SPA standard has been defined for each of the SPA system interfaces........................ 13
Figure 6 – The ASIM interfaces the device to the SPA-X network
Figure 7 – SPA network phases of operation
Figure 8 – The component discovery sequence
Figure 9 – Device registration sequence
Figure 10 – Message sequence to identify the Component Addressing Service
Figure 11 – Messaging sequence for assignment of logical address blocks
Figure 12 – A structural panel concept
Figure 13 – SPA port number format
Figure 14 – The Test Bypass Interface Network
Figure 15 – Test-bypass is implemented with a dual ported register file
Figure 16 – The Flight Software-in-the-Loop typical configuration using two PCs
Figure 17 – Component Data Capabilities and Component Network Capabilities
Figure 18 – The SPA spacecraft single-point grounding approach
Figure A.1 – SPA/OSI Layer Model correspondence
Figure A.2 – An OSI/SPA mapping
Figure A.3 – SPA architecture
Figure A.4 – SOIS path between application and device protocol stacks
Figure A.5 – Protocol stacks for SPA applications on the left; SPA ASIMs on the right

Tables
Table 1 – Letter code as it corresponds to the document
Table 2 – Letter code for bus voltage
Table 3 – Letter code for mechanical and thermal devices
Table 4 – Letter code for connectors and cabling
Table 5 – Letter code for test bypass interfaces
Foreword

The desire to quickly and reliably assemble spacecraft has been a challenge since the 1960s. In the 1990s the international computer market noted a similar need to quickly and reliably assemble computers and computer accessories. The invention of Plug-and-Play (PnP) capabilities is now assumed for any modern terrestrial computer system. PnP capability is defined in various publicly available technical standards.

It is the purpose of the AIAA to capture, in this Space Plug-and-Play Architecture (SPA) Guidebook and associated technical standards, technical approaches to adapt the various computer PnP capabilities to small spacecraft and the space environment.

This Guidebook provides a general description of a data centric spacecraft model used to form the on-board PnP network. Various views are used to clearly indicate how this works. A common ontology is described to allow for a profile specific Common Data Dictionary (CDD) so that a stable set of terms may exist. Interfaces between devices are described to simplify the implementation of PnP at the device level. Finally, those PnP protocols identified to date are generally described, as are the adaptations needed for space application.

The detailed requirements for each of these topics are listed in the respective AIAA SPA standards listed below.

- SPA Networking Standard
- SPA Logical Interface Standard
- SPA Physical Interface Standard
- SPA 28V Power Service Standard
- SPA System Timing Standard
- SPA Ontology Standard
- SPA Test Bypass Extension Standard
- SPA SpaceWire (SPA-s) Subnet Adaptation Standard
- SPA System Capability Standard

At the time of approval, the members of the AIAA SPA Standards Committee were:

Fred Slane, Chair  Space Infrastructure Foundation
Jeanette Arrigo   Sierra Nevada Corporation
Scott Cannon   Utah State University
Ken Center   PnP Innovations
Don Fronterhouse*  PnP Innovations
Rod Green   Design Group
Jane Hansen   HRP Systems
Doug Harris   Operationally Responsive Space Office
Paul Jaffe   Naval Research Laboratory
Stanley Kennedy*  Comtech Aero-Astro
The above consensus body approved this document in Month 201X.

The AIAA Standards Executive Council (VP-Standards Name, Chairman) accepted the document for publication in Month 201X.

The AIAA Standards Procedures dictate that all approved Standards, Recommended Practices, and Guides are advisory only. Their use by anyone engaged in industry or trade is entirely voluntary. There is no agreement to adhere to any AIAA standards publication and no commitment to conform to or be guided by standards reports. In formulating, revising, and approving standards publications, the committees on standards will not consider patents that may apply to the subject matter. Prospective users of the publications are responsible for protecting themselves against liability for infringement of patents or copyright or both.

* Alternate CoS Participant
Introduction

This Guidebook provides the introduction to the SPA standard set. The SPA effort is a response to the need for reduced design, fabrication, integration, and test schedules (and therefore related engineering costs) for small spacecraft. The primary goal of SPA is to enable completion of all satellite development phases in days instead of months and years.

With the current technical and standards base, it is common to allocate months in a small satellite development schedule just for integration. This allocation is often repeated recursively at lower levels of decomposition of a large space platform.

Under SPA, computer-negotiated interfaces permit the elements of a complex system to transparently contribute information that accelerates the integration process by reducing or eliminating error-prone human interpretation. Electronic self-configuration/self-organization allows for rapid space vehicle construction. Additionally, the placement of sensors and actuators on the spacecraft is not restricted to specific, predetermined locations. In the terrestrial electronics industry, this capability is called “Plug-and-Play” (PnP). The approach fully supports an à la carte method of constructing arbitrarily complex arrangements of virtually any sensor or actuator type. Self-configuration/self-organization makes the network not only easy to expand and modify, but also robust to component failure from either natural causes or from deliberate attack.

The expected impact of Plug-and-Play goes beyond spacecraft manufacturing to increased manufacturing rates for satellite bus components. Through the production of scores or hundreds of units the economies of scale and the amortization of Non-Recurring-Engineering costs, including iterative design, testing and certification, can fundamentally alter the profitability of satellite fabrication and integration. The result will be faster turns of satellite orders at a lower delivered price and a better profit margin to the manufacturer.

As the SPA concept advances, the set of internalized transport protocols will grow. The initial content of this guidebook will focus on standardization of ontology, the use of xTEDS to establish component (hardware and software) communications interfaces, the logical flow of SPA messages, functions of the SPA network, and the employment of existing data transport standards to form plug-and-play information interfaces. SPA standards complete the architecture with inclusion of physical interfaces.
1 Scope

This Guidebook provides an overview for spacecraft platform (system), subsystem, and component (including payload) developers with spacecraft plug-and-play architectures to promote rapid design, fabrication, integration, and test. Included is an introduction to SPA, providing an informative reference for the uninitiated reader. It also includes a summary of the SPA standards. The standard user is directed to the SPA standards for detailed requirements. In cases where material in this document differs from a SPA standard, the standard will take precedence.

2 Tailoring

Tailoring is a process by which individual requirements or specifications, standards, and related documents are evaluated and made applicable to a specific program or project by selection, and in some exceptional cases, modification and addition of requirements in the standards. When viewed from the perspective of a specific program or project context, the requirements defined in the SPA standards may be tailored to match the actual requirements of the particular program or project. Tailoring of requirements shall be undertaken in consultation with affected stakeholders, including the procuring authority where applicable.

3 Applicable Documents

The following documents contain provisions which, through reference in this text, constitute provisions of the SPA standards. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

3.1 Normative References

- W3C XML 1.0 Extensible Markup Language
- W3C XML Schema Part 1 XML Schema: Structures
- W3C XML Schema Part 2 XML Schema: Data types
- CCSDS 660.0-B-1 XML Telemetric and Command Exchange (XTCE)
- CCSDS 850.0-G-1 Spacecraft Onboard Interface Services – Informational Report
- ECSS-E-ST-50-12C SpaceWire – Links, Nodes, Routers and Networks, July 2008
- IEEE 1451 Standards family Standard for a Smart Transducer Interface for Sensors and Actuators

4 Vocabulary

4.1 Acronyms and Abbreviated Terms

- 6DoF 6-degrees-of-freedom
- AIAA American Institute of Aeronautics and Astronautics
- APT advanced plug-and-play technology
- ASIM Appliqué Sensor Interface Module
- ASME American Society of Mechanical Engineers
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTM</td>
<td>American Society of Testing and Materials</td>
</tr>
<tr>
<td>AWG</td>
<td>American wire gauge</td>
</tr>
<tr>
<td>C&amp;DH</td>
<td>command and data handling</td>
</tr>
<tr>
<td>CAD</td>
<td>computer aided design</td>
</tr>
<tr>
<td>CAS</td>
<td>component addressing service</td>
</tr>
<tr>
<td>CCSDS</td>
<td>Consultative Committee for Space Data Systems</td>
</tr>
<tr>
<td>CDD</td>
<td>Common Data Dictionary</td>
</tr>
<tr>
<td>CG</td>
<td>center of gravity</td>
</tr>
<tr>
<td>DNC</td>
<td>does not care</td>
</tr>
<tr>
<td>DOD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DPM</td>
<td>dual ported memory</td>
</tr>
<tr>
<td>EDA</td>
<td>electronic design automation</td>
</tr>
<tr>
<td>EOP</td>
<td>end-of-packet</td>
</tr>
<tr>
<td>EP</td>
<td>endpoint</td>
</tr>
<tr>
<td>GPS</td>
<td>global positioning system</td>
</tr>
<tr>
<td>GSE</td>
<td>ground support equipment</td>
</tr>
<tr>
<td>HWIL</td>
<td>hardware-in-the-loop</td>
</tr>
<tr>
<td>HWILS</td>
<td>hardware-in-the-loop simulations</td>
</tr>
<tr>
<td>Hz</td>
<td>hertz</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>ICD</td>
<td>interface control document</td>
</tr>
<tr>
<td>ID</td>
<td>identification</td>
</tr>
<tr>
<td>IGES</td>
<td>initial graphics exchange specification</td>
</tr>
<tr>
<td>MKS</td>
<td>meters, kilograms, seconds</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>OMG</td>
<td>object management group</td>
</tr>
<tr>
<td>OS</td>
<td>operating system</td>
</tr>
<tr>
<td>OSI</td>
<td>open system interconnect</td>
</tr>
<tr>
<td>PC</td>
<td>personal computer</td>
</tr>
<tr>
<td>PnP</td>
<td>plug-and-play</td>
</tr>
<tr>
<td>PPS</td>
<td>pulse-per-second</td>
</tr>
<tr>
<td>QoS</td>
<td>quality of service</td>
</tr>
<tr>
<td>SDA</td>
<td>satellite design automation</td>
</tr>
<tr>
<td>SDS</td>
<td>spacecraft data systems</td>
</tr>
<tr>
<td>SM-L</td>
<td>SPA manager for the SPA local interconnects</td>
</tr>
<tr>
<td>SM-s</td>
<td>SPA manager for SpaceWire protocol subnet</td>
</tr>
</tbody>
</table>
SM-x SPA manager for a generic subnetwork.
SOIS spacecraft onboard interface services
SPA Space Plug-n-Play Architecture
SPA-L SPA local interconnect
SPA-S SPA SpaceWire subnet
SPA-U SPA USB subnet
SPA-X SPA based on generic transfer standard
STEP standard for the exchange of project model data
TAT time-at-tone
TB test bypass
TBI test bypass interface
UDP user datagram protocol
uint<n> unsigned integer, n bits
USB universal serial bus
UUID universally unique identifier
XML eXtensible markup language
xTEDS extensible transducer electronic data sheet

4.2 Terms and Definitions
For the purpose of this document, the following terms and definitions apply.

**Appliqué sensor interface module**
a small microcontroller circuit card that provides a logical and physical interface from a non SPA compliant device to the SPA network. The ASIM contains information about the device and its functions and translates between device’s native command and data interface to the standard interface as defined in the xTEDS.

**Application**
software package that can be used in a SPA system

**Component addressing service**
provides blocks of logical addresses to each SPA manager requesting one during topology discovery or component registration; it stores the logical address block, UUID, and physical address of each manager

**Component**
hardware or software that can be used in a SPA system

**Device**
hardware item that could be used in a SPA system

**Extensible transducer electronic data sheet**
machine-readable representation of data provided, messages accepted, services provided, and physical characteristics (e.g., size, weight, and power) for a specific component within a plug-and-play network; xTEDS is conceptually based on the IEEE 1451 standards family
Plug and play
ability to connect a device to the larger system and have the two work together with little or no set-up required (e.g., automated system recognition and data exchange)

Registers
a set of 256 8-byte registers in dual ported memory

Router
a device on a SPA network with multiple ports that may be attached to either another SPA router or a component endpoint

SPA application
a software SPA component

SPA component
an endpoint whose interface conforms to the SPA standards and does not connect to another SPA object via a different port; SPA components are both devices (hardware) and applications (software)

SPA device
a hardware SPA component

SPA lookup service
responsible for accepting component registration and providing data source route information for components requesting a particular type of service (or returning an acknowledgment if the service is not available)

SPA manager
responsible for performing discovery for a particular subnet. It maps incoming packets to the correct SPA endpoint on the subnet, encapsulating the SPA packet with the correct protocol header. In the reverse direction it removes the protocol header and possibly adds a new header conforming to the subnet the packet is about to enter. It is also responsible for topology discovery and reporting within the subnet.

SPA network
an addressable and routable physically connected infrastructure composed of standard SPA transports for the purpose of transporting SPA messages between SPA endpoints and SPA gateways; the SPA network is made available as a SPA service to SPA components through a standard interface

Test bypass host
a test computer that recognizes, configures, and drives the test bypass routers and ASIMs while in a test configuration

Universal serial bus
familiar serial bus used by personal computers, which supports automated enumeration and plug-and-play

NOTE See [http://www.usb.org](http://www.usb.org)

5 Architecture

5.1 SPA Inside a General Space Architecture

SPA sits on top of existing space architecture. SPA uses a federated approach to integrate with existing space architecture in areas such as launch systems and ground systems—those systems are not affected by SPA. In the area of spacecraft systems SPA only affects internal spacecraft aspects; SPA does not affect spacecraft external interfaces. In addition, SPA affects design, fabrication, integration and
all aspects of test in a spacecraft lifecycle (see Figure 1). SPA is not intended to affect spacecraft mission operations or end of life, but it can have a positive impact on spacecraft health through the operational phase.

Because SPA does not address spacecraft external interfaces, SPA systems, services, data, and standards are constrained to describe spacecraft subsystems, assemblies, and components at interfaces and the interfaces themselves. It is important to note that SPA sits on top of existing space architecture even within the spacecraft. This replication of the federated approach mentioned previously allows SPA to take advantage of general advances in space systems development without the need to recreate SPA at each turn. Partially for this reason there is an expected connection between general spacecraft architecture and SPA that is not necessarily well defined from a standards perspective. There has been discussion on developing SPA Application Program Interfaces (APIs) to more clearly constrain the connection from general spacecraft architecture to SPA, but as of this publication there is no clear consensus on exactly what such a standard should require.

There are computer/software architecture models that have been applied to space systems. Annex A discusses the OSI model and the CCSDS SOIS model.

5.2 SPA Inside

Different views present themselves within SPA, although the primary view is data-centric. SPA clearly describes specific viewpoints under the architecture. The SPA standards contain requirements primarily related to the following views:

- **System View**
  - Physical Interfaces, including mechanical and electrical interfaces are described in the Physical Interfaces Standard (AIAA S-133-4-201X)
  - Testing capabilities for SPA compliant spacecraft components, used primarily during spacecraft integration, are described in the Test Bypass Standard (AIAA S-133-8-201X)

- **Services View**
  - Power is described in the Power Service Standard (AIAA S-133-5-201X)
  - Timing is described in the System Timing Standard (AIAA S-133-6-201X)
  - Spacecraft internal computer network establishment and form are described in the Networking Standard (AIAA S-133-2-201X)

- **Data-Centric View**
  - The necessary data and data structure for components is described in the Ontology Standard (AIAA S-133-7-201X)
  - Spacecraft design adaptation of existing space or computer technology protocols are described in the Subnet Adaptation Standards (e.g., SpaceWire Subnet Adaptation; AIAA S-133-9-201X)
  - Spacecraft internal computer network traffic structure and interaction is described in the Logical Interface Standard (AIAA S-133-3-201X)
As an example, the SPA Logical Interfaces Standard defines data structure as message content and format between networking managers and components in the logical interface. Message sequences describe how data flows strictly among the networking managers. The SPA Networking Standard describes some of the network manager to component from a services perspective. Both views are necessary to establish complete requirements for such capabilities as component detection and component registration. Therefore, care should be taken to understand the complete set of requirements in the context of systems, services, and data.

SPA Standards are structured in a federated fashion. The following example outlines future innovations that could replace the network and transport layers of the protocol stack. The SPA Logical Interface Standard may be viewed as a layer in the protocol stack between the application and network layers. The SPA Networking Standard may be viewed as a description of a way to implement the network and transport layers. The documents are structured due to a possibility that the network and transport layers could be replaced in the future without affecting the logical interface. This may be an enabler in the development of standardized SPA application programming interfaces (APIs).

In Section 5.1 of this guide, SPA is defined as having an explicit dependence on spacecraft computational capabilities such as might be provided by a computer operating system (OS) or discrete services which effectively fill that function. The SPA compliant bus subarchitecture can be federated and/or distributed; balancing the positive or negative aspects of each is left to the spacecraft designer. This means that under this dated rendition of SPA standards, compliant components can be used on any SPA compliant spacecraft, but each spacecraft designer will still need to create his/her own SPA compliant spacecraft control electronics suite.

The remainder of this guidebook discusses core concepts of SPA in some detail and how those concepts
are translated into systems, services, and data-centric constructs. These systems, services, and data-centric constructs are further described in a sample (exemplar) implementation at the spacecraft systems level, subsystems level, and component level. SPA relevant tools that may be used in the design, fabrication, integration, or test phases are also discussed. Finally, the SPA standards are described.

6 SPA Goals, Concepts, Principles, and Structure

6.1 Primary SPA Goal: Eliminating Barriers to Rapid Satellite Deployment

SPA, without standards, can be implemented within any small organization or small collection of organizations. The underlying concept that plug-and-play (PnP) technology from the terrestrial computer industry can be adapted to space is not a huge technical challenge. SPA implementation across a broad base requires standards for certain things to allow: broad interoperability with diverse component providers, a cost-effective interface for component providers to build to, and rapid configuration/reconfiguration as the spacecraft is integrated from compliant components. The SPA standards provide a unified plug-and-play methodology to facilitate the rapid deployment of space systems using modular components. A space system typically involves numerous independent component providers. The SPA standards foster an understanding between component providers that if their component is SPA-compliant they can expect full interoperability on a SPA system. PnP also enables the application of drag-and-drop satellite design and other design automation concepts that allow the satellite designer to rapidly design to mission requirements, simulate on-orbit operations, adjust the design as necessary, and verify the design before ever assembling the satellite. Along with the application of PnP technologies, tools that support rapid design and testing of SPA satellites are a critical component to achieving the goals of rapid satellite design, fabrication, integration, and test.

6.2 SPA Core Concept and Essential Services

The core concept of SPA is that components (where a component is a data producing or consuming node in the networked system) register their capabilities with a Lookup Service when they are added to the system (see Figure 2). Once this information is captured, any component with a data need may query the Lookup Service for available sources and receive matches to that query. Subsequently, that component may contact any or all matching components directly to subscribe to data that it provides or utilize its data services.

![Figure 2 – SPA core concepts](image-url)
There are essential services that follow directly from SPA core concepts. They are listed here:

Component Detection – In order to be able to form self-organizing networks, a SPA system must be able to detect any time hardware or software is added to the system.

Component Registration – Once the system detects that a component has been added to the network, a mechanism automatically registers the component with the network. Registration includes relaying to the network the data the component produces and, optionally, the data that it will consume.

Component Self-Identification – SPA components must provide information about their functions to the system.

Command/Response Messages – SPA components send command messages to initiate an action. If appropriate, an initiated component sends a response message with the requested data. A SPA component’s xTEDS defines the command and response messages supported by the component.

Publish/Subscribe Messages – SPA components publish periodic data through predefined data messages. SPA components subscribe to data messages as needed. These subscription data messages are defined in the component’s eXtensible Transducer Electronic Data Sheet (xTEDS).

Identification of Component Failure – SPA systems detect when a component with an active subscription is unable to provide the requested data so that an alternate data source can be sought.

System Monitoring of Component Status – A SPA system monitors the status of components (fully operational, dependency failures, component failure, etc.).

The data-centric approach of SPA leads to an abstraction of data interfaces to components, regardless of whether a component is a hardware device or a software application. This virtualization allows systems to be assembled rapidly without specific configuration or tailoring of the constituent parts. A SPA Logical model is included in Figure 3 showing logical interactions under SPA. “Other SPA Services” are needed in a SPA system but may be provided by non-SPA systems, such as an OS.
6.3 SPA Basic Capabilities

Most of the purpose in the engineering of interfaces is to provide a means to handle the "care and feeding" of hardware and software components, and to relay information between them. In principle, it should be possible to organize and automate (plug and play) the interfacing process to provide access to the core pieces of information or services in each device or software application. To develop the SPA standards, a number of SPA basic capabilities were established and are described in the subordinate sections below. The basics for SPA follow from essential services and core concepts of a well-defined SPA system.

6.3.1 Standardized Physical Interfaces

Devices in the SPA network employ well-defined hardware (electrical, mechanical, and signaling) standards to achieve an interface with connective integrity. SPA devices are physically mountable on a compliant SPA structure (reference SPA Physical Interface, 28V Power Service, System Timing, SpaceWire Subnet Adaptation, and Test Bypass Extension standards). Although not listed in the essential services that follow from the core concepts, two additional services are fundamental to the spacecraft.

- System Common Time – A SPA system provides common time to the system components.
- Standard Mechanical and Electrical Interfaces – SPA systems provide standard mechanical and electrical interfaces for hardware components.
6.3.2 **Self-Organizing Networks**

SPA network addressing and routing tables are self-configured. The SPA network is created dynamically as devices are introduced to it. Any compatible SPA-X device can connect to a same type SPA-X network in any available location. As the satellite is integrated, unique SPA addresses are assigned. SPA can support a heterogeneous network consisting of at least one SPA-Local (SPA-L) network, a Subnetwork Manager (SM-L) running on a SPA-L processing node and any number of SPA subnetworks (SPA-X), each with a SPA-X Subnetwork Manager (SM-X).

6.3.3 **Communication Through Standard Messages**

SPA network uses a self-discovery process with standard-format, predefined messages. In SPA systems there is little distinction between a hardware device that supports an interface and a software application that does the same. Standard messages are used to exchange data and information among components as the system grows and functions. The complete set of SPA messages are defined in the SPA Logical Interface Standard (AIAA S-133-3-201X).

6.3.4 **Query Services**

The xTEDS enables SPA components to coordinate, share data, find resources, and provide resources and services without knowing the physical location and structure of other system components. After SPA components register they can query the Lookup Service for their data consumption needs. These queries are structured by data “kind” or standard “interface” with “qualifiers” that provide additional context to focus the search. A consumer component can subscribe to messages of providers that meet their search criteria.

6.3.5 **SPA-X Interfaces**

Several pre-existing data communications protocols, such as Universal Serial Bus (USB) and SpaceWire, can be used in SPA systems. A generic designation of such is SPA-X. Extensions to the base protocol may be necessary to accommodate the real-time embedded systems and harsh operational environments onboard a satellite. Each specific SPA-X transport standard derives its name from the adapted data transport standard, such as SPA-U (using USB) and SPA-S (using SpaceWire). Only same type devices conforming to the selected SPA-X Subnet Standard are recognized on a SPA-X network.

6.3.6 **Self-Describing Components**

SPA components use an XML byte string called an Extensible Transducer Electronic Data Sheet (xTEDS) in order to self-describe their interfaces. Descriptions of data products within data messages are constructed from a profile dictionary of standard terms (a Common Data Dictionary; CDD). Every device (e.g., sensors, actuators, processors, hubs, and routers) and every software application needs an xTEDS to function on the SPA network as anything more than a consumer of data.

7 **SPA Implementation**

7.1 **Overview**

A SPA system (a small spacecraft) is built from SPA subsystems and SPA components. A spacecraft designer will integrate SPA into the system solution to mission requirements. This solution will include components in the form of various sensors and emitters, attitude and propulsion capabilities, heaters or coolers, power systems, and all the things that are part of a spacecraft. Of critical importance is the computer system or systems that let the spacecraft work. The software (programs and data), which may be an Operating System (OS), manages the hardware and provides common services for the execution of applications. Common services may include support to hardware functions such as input and output and memory allocation.

The SPA spacecraft designer will use a SPA network for the primary spacecraft computer system.
Depending on subsystem complexity it may be necessary to integrate one or more subsystem networks under the primary network. All the spacecraft components, both hardware devices and software applications, will necessarily be SPA compliant or made-to-be SPA compliant (e.g., by use of an ASIM) as they interface to SPA networks or SPA subnetworks.

Rapid implementation of SPA will require predesign of the following:

- A complete spacecraft software service set compliant with
  - The SPA Networking Standard
  - The SPA Logical Interface Standard
  - At least one SPA Subnet Adaptation Standard (e.g., SPA SpaceWire Subnet Adaptation Standard)
  - The SPA Ontology Standard
  - The SPA System Timing Standard

- System and subsystem networks compliant with
  - The same standards as the spacecraft software service set
  - The SPA Physical Interface Standard

- Devices compliant with (or made-to-be compliant with)
  - The SPA SpaceWire Subnet Adaptation Standard for the system or subsystem each is on
  - The SPA Ontology Standard
  - The SPA System Timing Standard
  - The SPA Physical Interface Standard
  - The SPA 28V Power Service Standard
  - The SPA Test Bypass Extension Standard

- Applications compliant with (or made-to-be compliant with)
  - The SPA Ontology Standard
  - The SPA System Timing Standard

7.2 Spacecraft Software Services From SPA

Selection or development of the SPA compliant spacecraft level system software is left to the system developer (this is discussed in Section 5). The collection of services, network/subnetwork managers, and applications within SPA is generally referred to the SPA Services Manager (SSM). An SSM may be the operating systems on a spacecraft or part of an operating system on a spacecraft. One possible example of that SSM implementation is given in Figure 4.

One possible instantiation of an SSM may eventually be an open source implementation of the SPA standards.

In addition to implementing the baseline functionality required in order to comply with the SPA standards, the SSM also implements functionality considered to be best practice or desirable.
7.2.1 Configuration Caching

The SPA Services Manager (SSM) provides a method to store system topology and registration data previously derived from the standard SPA self-discovery and self-configuration process.

7.2.2 Configuration Modes

The SPA Services Manager (SSM) provides two configuration modes. The default mode is for the SPA Services Manager (SSM) system to start-up and go through the standard SPA self-discovery and self-configuration process. The second mode is to bypass the standard SPA self-discovery and self-configuration process and configure the system using cached configuration data. It is anticipated that the ability to cache a known system configuration will be used to provide highly desirable SPA functionality, but these standards do not provide guidance on when, how, or why the configuration cache is executed. Possibilities include decreasing system startup time, system verification and diagnostics, or deploying services for access control and security.

7.2.3 Adding or Updating Software

The SSM provides a method for adding new or updated software modules to a deployed system.

7.3 SPA Networking

SPA is a networked data exchange model. One of the premises of SPA is that there is no distinction between a hardware device that supports an interface and a software application that does the same. The SPA networking topology supports a heterogeneous network consisting of at least one SPA Local (SPA-L) Network Manager running on a SPA processing node and any number of SPA subnet (SPA-X) managers as shown in Figure 4. A SPA standard has been defined for each of the SPA system interfaces. The network configuration shown simply serves as an informative reference for the reader to develop a clear understanding of the way in which the required features of the SPA network are envisioned to interact. SPA-X managers are used to bridge the differing SPA networks. Examples of SPA-X managers include SPA-S (SpaceWire), SPA-U (USB), SPA-1 (I2C), and SPA-E (Ethernet).

Unique SPA addresses are assigned by a SPA Component Addressing Service (CAS). The Component Addressing Service (CAS) shown on the left of Figure 5 provides a logical address block upon request from a subnet manager. A SPA network uses logical addressing so the addresses in that block may only
be assigned to components residing on that subnet. Only one CAS may be active in a particular SPA network at any time. The CAS process must exist on the SPA-L interconnect, and it is directly accessible by all SPA subnet managers (SM-x) in the SPA network after the SM-L has delivered its address.

SPA network topology is determined through a self-discovery process. SPA network addressing and routing tables are self-configured. A local SPA interconnect protocol (SPA-L) is used to connect applications on a processing node of the SPA network. The SPA-L is controlled by a SPA manager for local SPA interconnects (SM-L) process, just as a physical subnet would be controlled by a SPA subnet manager (SM-x) process or device—the "-x" is a generic notation for any subnet interface standard such as SpaceWire, I2C, USB, and so forth. The SM-L process receives an address block for each process on the node, just as the SPA SpaceWire subnet manager (SM-s) receives an address block for each endpoint on a SpaceWire network.

A SPA subnet manager (SM-x) is most commonly a protocol-specific SPA process on a SPA-L, which acts as a translation device for a particular protocol. A SPA subnet manager should

- Know the path route to each endpoint in its subnet.
- Know the processes currently active in each endpoint and have requested a logical address from the Component Addressing Service (CAS) for them per the SPA Networking Standard.
- Know the path route to each SPA subnet manager residing within the subnet, know the logical address of all other core components (other SPA subnet managers, Component Addressing Service, data managers in the SPA network), and notify the SPA Lookup Service of each component needing registration.

An endpoint (EP) is simply a SPA component that does not connect to another SPA object via a different port. A SpaceWire connected endpoint should have a routing table containing the path route to at least one manager in the subnet. The ability to directly address other endpoints and SpaceWire subnet manager in the subnet may be implemented, but it is not required because a SpaceWire subnet manager can re-encapsulate any SPA message to reach a local target based on its routing tables.

A SpaceWire subnet manager, for example, accepts packets received from SPA-L and removes the protocol-specific header. Then it performs a lookup of the SPA destination address in its local lookup
table and encapsulates the packet in a SpaceWire header/End of Packet (EOP), with the correct destination or logical SpaceWire route to reach that endpoint or adjacent manager. In the outgoing direction it removes the SpaceWire header from the outgoing packet and adds a protocol header as indicated in its outgoing lookup table.

In a hardware implementation, a SPA subnet manager (SM-x) might act as a simple bridging device to move packets from one protocol subnet to another. In this case the SPA subnet manager (SM-x) would be required to request addresses from the Component Addressing Service for items in both subnet, and translate packets moving from one domain to another. It is important to note that it is normal practice for multiple SPA subnet managers to coexist on the same subnet, independently performing topology discovery.

The SPA Lookup Service accepts component registrations of SPA components providing services and provides the ID for that service to other components that request them. The SPA Lookup Service process must be instantiated on a SPA-L interconnect, and it can be directly accessed by SPA subnet manager nodes after it has been discovered and its address has been distributed.

7.4 SPA Components (Devices and Applications)

7.4.1 Component Capabilities

SPA compliant components

- Run the application to send and receive transport protocol messages
- Run the device segment of the SSM software and send and receive SSM messages
- Store the device xTEDS
- Serve as the final arbiter of the device health and status
- Are powered exclusively by system power to the device
- Use the single-point ground to the device
- Interface with the system one pulse per second (PPS) time synchronization for the device and provide an internal oscillator for time synchronization
- Provide built-in test functions
- Use test-bypass infrastructure for direct connection to hardware-in-the-loop simulations (HWILS).

7.4.2 Non-SPA Components Made-To-Be SPA Compliant (Adding ASIMs)

Non-SPA compliant devices will need an interface module (an Appliqué Sensor Interface Module, or ASIM) for physical and computational connectivity.

The Appliqué Sensor Interface Module (ASIM), with a SPA-X connector, provides the encapsulated wiring and software translations needed to interface a legacy hardware device to a SPA-X port. The ASIM is a very important implementer of the hidden complexity concept discussed previously. The ASIM bridges the specifics of the device behavior model to the interfaces described in the SPA-X standard. The device xTEDS will be stored in the ASIM (see Figure 6). It is envisioned that the use of ASIMs will be phased out as manufacturers develop fully SPA compliant hardware devices.
Figure 6 – The ASIM interfaces the device to the SPA-X network

The ASIM establishes the common data interface for SPA devices by providing the necessary hardware and software (see Figure 6). Network connections are indicated by the arrows to the right of the SPA-X interface. Several variations of SPA-X ASIMs have been developed. ASIMs provide the following functions:

- Run the application to send and receive transport protocol messages
- Run the device segment of the SSM software and send and receive SSM messages
- Store the device xTEDS
- Convert device-specific data into xTEDS data messages
- Convert xTEDS commands into device-specific commands
- Serve as the final arbiter of the device health and status
- Provide system power to the device
- Provide a single-point ground to the device.

The ASIM may also provide:

- Interface with the system one pulse per second (PPS) time synchronization for the device and provide an internal oscillator for time synchronization
- Built-in test functions for the device
- Test-bypass infrastructure for the direct connection of SPA devices to hardware-in-the-loop simulations (HWILS).
8  Example SPA Implementation

8.1  Turning on a SPA System: What Happens When

The four main stages of SPA network operation are shown below in Figure 7. These four phases correspond to the SPA core concepts of register, find, and access.

![Figure 7 – SPA network phases of operation](image)

8.1.1  Register

The Register concept is executed by topology discovery and component registration.

Network topology discovery is the first action of the SPA system. This discovery is automated for all SPA compliant networks and allows the system to proceed to component discovery.

The component discovery process occurs after completion of the network topology discovery as described in the SPA Networking Standard. Upon completion of network topology discovery, the SPA Lookup Service probes discovered components to complete the component registration process. The SPAProbeReply message contains the CUUID and XUUID for the registering component and receipt of the ID completes the discovery process.

8.1.2  Find

The find concept is executed by subscription processing (see Figure 8).

The SPA message family supports the ability for a SPA component to query for desired data, command providers, and service providers. The responses from queried components indicate matching components that provide the requested data. Three key query capabilities are provided: (a) the ability to query for an item based on its name or qualifiers, (b) the ability to request metadata on a specific variable, and (c) the ability to retrieve an entire xTEDS.
Once a query has revealed a suitable source, it is up to the consumer component to establish a subscription with a provider. That is, establish a connection to the data source. This can be brokered by the SPA Lookup Service, which tracks all subscriptions so that it can notify all consumers if a provider is cancelled intentionally or cancelled due to unresponsiveness. Or, the consumer can send a subscription request directly to the producer component. Subscriptions can be cancelled either by the consumer component or the producing component. Subscriptions can also be maintained over defined time periods by defining a “lease” period. After components have discovered each other and/or subscribed to data, they are ready to exchange application-specific information (Intercomponent Data Exchange) as described in Figure 9.

Figure 8 – The component discovery sequence

Once a query has revealed a suitable source, it is up to the consumer component to establish a subscription with a provider. That is, establish a connection to the data source. This can be brokered by the SPA Lookup Service, which tracks all subscriptions so that it can notify all consumers if a provider is cancelled intentionally or cancelled due to unresponsiveness. Or, the consumer can send a subscription request directly to the producer component. Subscriptions can be cancelled either by the consumer component or the producing component. Subscriptions can also be maintained over defined time periods by defining a “lease” period. After components have discovered each other and/or subscribed to data, they are ready to exchange application-specific information (Intercomponent Data Exchange) as described in Figure 9.

Figure 9 – Device registration sequence

The SPA message family supports the ability for managers on the SPA-L interconnect (SM-L) to query to see if they host the Component Addressing Service (CAS). This message sequence is illustrated in Figure 10. If the SPARedReady exchange completes successfully, then the Component Addressing Service (CAS) returns its own logical address to be used in future transactions.
The Component Addressing Service (CAS) sends the SPAReddy message to a known port to query the readiness of the SM-L.

If SM-L is ready, it responds with a SPAReddy reply. If the SM-L is not ready, the CAS will eventually time out of the request. Because the SPA-L is a lossy protocol, the CAS may need to retry several times to complete the transaction.

Figure 10 – Message sequence to identify the Component Addressing Service

The SM-L next begins probing the local interconnect to determine the address block size required for local components and components accessible through the SPA subnet manager (SM-x) present on the interconnect. Logical address blocks are assigned to the SPA managers per the message sequence described in Figure 11. Using a similar methodology, the remaining component network capabilities shown in Figure 11 are executed. The precise message formats are provided in the SPA Interface Standard.

### 8.1.3 Access

The SPA network supports active and passive Health and Status monitoring to report issues or faults within the network. With active monitoring the SPA Lookup Service sends a message to each component that it is monitoring, and expects a reply within an allotted time. Passive monitoring requires a master to configure components to send a message to the master at a periodic rate. This is achieved through the use of the SPAProbeRequest and SPAProbeReply messages. Fields within these messages not only aid in discovery, but also allow the SPA Lookup Service to monitor the basic health of a component.

A graphic summary of Register, Find, and Access in SPA message sequence (data-centric description), with reference to a topology dependent discovery process (a service) is shown in Figure 11.
1. Before topology discovery the SPA subnet manager (SM-x) periodically transmits SPAReady messages to announce its presence.

2. After learning the location of the CAS, the SM-L responds to the SM-X with a SPAReady message. This ends the periodic broadcasts of SPAReady messages from SPA subnet manager.

3. If the SPA subnet manager has not already done so, it performs protocol specific topology discovery on its subnet to learn the number of components, their types and UUID, and the locations of any additional SPA subnet manager (SM-x).

4. The SM-L requests a block of addresses for itself from the CAS using the SPARqstAddrBlock message.

5. Component Addressing Service assigns an address block with the SPAAssignAddrBlock message.

6. SM-L requests an address block from the CAS for each SPA subnet manager (SM-x) it discovered on its local interconnect.

7. Component Addressing Service responds with a SPAAssignAddrBlock message.

8. SM-L applies the blocks with the SPAAssignAddrBlock message to the SPA subnet manager.

9. The SM-L also distributes the routes to any known core components to the SM-x on its interconnect with a SPADistributeRoute message for each item.

10. Each SM-x may now request an address block for SM-x that it discovered on its subnet during discovery with a SPARqstAddrBlock command.

11. The address blocks are assigned by the CAS with a SPAAssignAddrBlock command.

12. The requesting SM-x will distribute the blocks with a SPAAssignAddrBlock command to the target SPA subnet manager within its subnet.

13. Each SPA subnet manager also propagates the routes to core components it is aware of with a SPADistributeRoute message.

Figure 11 – Messaging sequence for assignment of logical address blocks
8.2 Panel Concept

In practice, a physical framework has to exist to which the appropriate hardware and software can be rapidly added. (Although it is possible to connect a SPA device directly to a host, the simplistic network formed fails to reap the benefits of PnP.) A standardized structural panel connected to a host processor or network is one approach envisioned for the spacecraft bus with the built-in ability to support plug-and-play devices, such as sensors and actuators. The SPA panel networking device, the ASIM, the SPA connector, and the SPA port are defined by the SPA-X standard for a specific transport standard protocol such as SPA-U for the USB standard, or SPA-S for the SpaceWire Subnet Adaptation standard. The defined standards include mechanisms to provide power and to enable system synchronization and system testing. These are uniquely defined for each SPA Subnet Adaptation Standard. In some styles of transport interface, it is necessary to include devices that facilitate network scaling.

Examples include routers (SPA-S), hubs (SPA-U), and bridges (which negotiate between two different SPA subnetworks). Often these devices include their own captive endpoints, or they may be integrated into endpoints (e.g., a SPA-U device might contain its own hub).

Figure 12 provides a visual example of this structural panel concept, which can be used with any standard transport interface protocol (e.g., USB, SpaceWire). The structural panel contains the networking device, such as a router or a hub, and harnessing to prepositioned ports that together allow the panel to be a node with multiple endpoints on the spacecraft network. The sensor or actuator, through the ASIM and SPA connector, can be connected to any SPA-X port on the spacecraft. One may take a number of the panels just described and connect them in a box or whatever shape is physically possible to form a spacecraft bus.

![Figure 12 – A structural panel concept](image)

A SPA-X endpoint device (e.g., a sensor, processor, or actuator) could be connected to any compatible port on any SPA-X network or subnetwork. For example, a spacecraft structure might have a number of pre-integrated SPA-U ports in various locations. It should be possible to plug a SPA-U device in any location on the network and, if needs dictate, to relocate the same device to another port without altering software or hardware in the overall spacecraft.
9 SPA Tools

9.1 Design Tools

9.1.1 Pushbutton Toolflow

The “push-button toolflow” concept is a metaphor of design in which a user is guided through a decision process using computer-automated “wizards” that lead to a constructible specification. A web-based consumer product purchase, such as the approach used by a computer manufacturer's website, is a well-known example. Users often start with a rough concept of a computer they wish to purchase, and logical options and alternatives are presented as the user works through the website (i.e., the toolflow), resulting in a constructible computer. As the combinatorial space of commodity options is large, it is possible that a user could create a unique configuration, even though all components presented through the tool are essentially commodity items. In electronic design automation (EDA), push-button toolflows are also employed, though much more user expertise and interaction is required. The sentiment of a push-button toolflow is very resonant in the “six-day spacecraft”. Indeed, it is difficult to imagine how the concept can succeed without it. The idea of exploiting computer-aided design (CAD) in creating a spacecraft is referred to as satellite design automation (SDA), which, coupled with principles of push-button toolflow, expresses a vision of a comprehensive set of open-architecture tools in which SPA components become library components, and third-party wizards and “add-ons” can be intermingled.

Push-button toolflow is being investigated and developed by a SPA Technical Committee working group.

9.1.2 xTEDS Simulator

This simulator is a specific case of an XML simulator (a common XML verification tool) that allows the emulation of devices and sensors in a SPA compliant system. The xTEDS Simulator acts like the device; registering its xTEDS and appropriately responding to subscriptions, commands, and service requests. As such, a SPA application can be tested and debugged without the physical devices.

This is intended for preliminary testing and debugging prior to traveling to the assembly site—where more complete integration testing and bypass would be available.

An example of an xTEDS Simulator is that developed by the US Air Force Research Laboratory (AFRL Space Vehicles Directorate community of interest. Information for accessing Revision 1.1.0 of the AFRL xTEDS Simulator and the associated user's guide is provided at https://pnpsoftware.sdl.usu.edu. Look under the Download/SDM tab. Documentation is under the Documentation/SDM tab. This latest release now runs from a much simpler and faster Linux or Windows GUI and allows data files to be used for simulated devices.

9.1.3 SPA Device Part Numbers—ID for Use With Design Tools?

Any rapid production facility or capability will need to automatically track the interface requirements for SPA components on hand (stock). Rapid design will require some manner of automatically recognizing component interfaces capable of meeting mission design requirements. For these reasons a unique numbering system for SPA parts makes sense. Notionally, a SPA part number will have the following format as shown in Figure 13:
Figure 13 – SPA part number format

The underlined letters are placeholders for variable values, whereas the “SPA” prefix will apply to all components. The following sections, corresponding to the numbering in the diagram, provide additional information on each segment of the SPA part number.

9.1.3.1 Compliance With SPA Standards

The use of the “SPA” prefix on a part number indicates the part is SPA compliant.

9.1.3.2 Subnetwork Standards

The various subnetwork standards describe the software, protocols, and hardware requirements of components as they apply to specific subnetwork types. These variations include the functionality of subnetwork adapter software and subnet specific messages (either as they are translated from SPA messages or as used for network enumeration). For more information, see the specific standard for the subnetwork type as located by the letter code (Table 1).

Table 1 – Letter code as it corresponds to document

<table>
<thead>
<tr>
<th>Letter Code</th>
<th>Standard Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>SpaceWire Subnetwork Adaptation</td>
</tr>
<tr>
<td>U</td>
<td>Universal Serial Bus Subnetwork Adaptation</td>
</tr>
<tr>
<td>E</td>
<td>Ethernet Subnetwork Adaptation</td>
</tr>
<tr>
<td>I</td>
<td>I2C-bus Subnetwork Adaptation</td>
</tr>
<tr>
<td>O</td>
<td>Fiber Optic Subnetwork Adaptation</td>
</tr>
</tbody>
</table>

NOTE If a component is being created that will support multiple subnet types, list the letter codes for all of the supported network types separated by slashes (/). For example a component supporting both SpaceWire and USB will begin its part number with SPA-S/U. Ordering of the letter code doesn’t matter, but it is recommended that it flow from left to right similar to the order of the list above proceeding top to bottom.

9.1.3.3 Power, Grounding, and Bonding Standards

The various power, grounding, and bonding standards incorporated the specifications for the power bus as well as the recommended “best practices” for grounding and bonding within space systems. There are currently two acceptable bus voltages (see Table 2) based on the needs of the various spacecraft designs and missions these standards are intended to be applied to.
Table 2 – Code for bus voltages

<table>
<thead>
<tr>
<th>Code</th>
<th>Standard Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5V Power, Grounding and Bonding</td>
</tr>
<tr>
<td>28</td>
<td>28V Power, Grounding and Bonding</td>
</tr>
</tbody>
</table>

NOTE The 5V standard has not yet been created and the name and/or content are subject to change until published.

9.1.3.4 Mechanical and Thermal Standards

The specifications for mechanical and thermal standards describe the mechanical requirements for devices for interfacing with a plug-and-play chassis. Again, there are currently two recognized variations based on the needs of the missions and spacecraft these standards are intended for (see Table 3).

Table 3 – Code for mechanical and thermal devices

<table>
<thead>
<tr>
<th>Code</th>
<th>Standard Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mechanical Interface with a 1cm Grid</td>
</tr>
<tr>
<td>5</td>
<td>Mechanical Interface with a 5cm Grid</td>
</tr>
</tbody>
</table>

NOTE The 1 cm grid standard has not yet been created and the name and/or content are subject to change until published.

9.1.3.5 Connector and Cabling Standards

The standards for connectors and cabling describe the requirements for accepted SPA cables, connectors, and wire harnessing. There are currently two connectors used in practice: a 28-pin micro-D used for SPA-S systems (Type A), and a 15-pin micro-D used for SPA-U systems (Type B). Table 4 shows the letter code for each.

Table 4 – Letter code for connectors and cabling

<table>
<thead>
<tr>
<th>Letter Code</th>
<th>Standard Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Interface Connector and Cabling (28-pin)</td>
</tr>
<tr>
<td>B</td>
<td>Interface Connector and Cabling (15-pin)</td>
</tr>
</tbody>
</table>

NOTE The Type B (15-pin) connector standard has not yet been created and the name and/or content are subject to change until published.

9.1.3.6 Test Bypass Interface Standard

The Test ByPass Extension Standard (AIAA S-133-8-201X) describes the requirements for the test bypass secondary network as implemented on a spacecraft. There are two major variations: either a parallel network (hardware-in-the-loop information not running over the network defined by the specifications listed in Section 1.2 of this document) or an in-band network (information is being transported over the networks that are listed in Section 1.2).

Table 5 – Letter code for test bypass interfaces

<table>
<thead>
<tr>
<th>Letter Code</th>
<th>Standard Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/N</td>
<td>In-Band Test Bypass</td>
</tr>
<tr>
<td>P/N</td>
<td>Parallel Network Test Bypass</td>
</tr>
</tbody>
</table>
NOTE Currently there has only been a Parallel Network (P/N) actually implemented. The In-band (I/N) letter code is a placeholder for a potential future specification and the name and/or content of the In-band (I/N) standard are subject to change until published.

9.1.3.7 Example

The part number format described here should be used by both component suppliers as well as customers. An example of a part number as it might be used is presented here: SPA-S-28-5-A-P/N.

9.2 Test Tools

9.2.1 Hardware-in-the-Loop Simulation and Test Bypass

Test Bypass provides the ability for SPA components or entire SPA systems to be rapidly tested without the need to disturb the flight physical configuration. The Test Bypass (TB) network shown in Figure 14 is separate from the spacecraft SPA data network. Because it is a separate network, the Test Bypass (TB) network does not interfere with the function of the spacecraft SPA network. Physical interface from devices into the Test Bypass (TB) network is accomplished through the standard SPA device connector. The Test Bypass (TB) network is powered from external Ground Support Equipment (GSE) to further isolate it from the SPA system under test. Isolating the system under test from the test apparatus removes uncertainties induced by unexpected interactions between the system and the testing apparatus, such as excessive network loading, processor bandwidth, data latency, and message prioritization. The actual spacecraft system timing and latencies are exactly as they will be during operations. Multiple Test Bypass Controllers interface with the GSE simulation to provide closed loop functional testing. On-orbit, the Test Bypass (TB) infrastructure is unpowered, and therefore, unable to interfere with spacecraft operations. A detailed description of Test Bypass functionality, commands, responses, and protocols can be found in the SPA Test Bypass Extension Standard.

Figure 14 – The Test Bypass Interface Network

In a Test Bypass network, simulated data is injected directly into SPA network devices (endpoints) such as star trackers, inertial measurement units, and other sensors through the Test Bypass controller. SPA devices are also able to accept commands from the Test Bypass (TB) controller. In this way, Test Bypass provides the capability to rapidly perform functional tests on assembled systems for verification of SPA network functionality, computing resource margins, and flight software performance. Test Bypass is also
used to validate on-orbit system performance based on spacecraft reaction to injected sensor inputs. Through Test Bypass, the performance of rapidly assembled spacecraft configurations is validated without the need for dedicated test beds and without the need to disturb the spacecraft physical configuration that may have already been qualified for flight and subjected to workmanship verification testing.

A powerful concept in SPA, though not directly related to “plug-and-play”, involves the ability to support the injection of synthetic information or instrumentation of data from components. In the simple example of a SPA thermister, it is possible to substitute the ambient temperature values generated by the thermister in the integration facility with a desired control value representative of the space operational environment in a way that does not perturb the existing SPA network. When coupled with a simulation infrastructure, the test-bypass facility of a network of SPA devices becomes an in situ hardware-in-the-loop system.

The philosophy behind test-bypass is that rapid system development benefits from (if it does not in fact require) the improved ability to dry-run part or all of the system and to expose hooks to improve testing and debugging in the event of inevitable irregularities that are likely to occur in development.

Producing a working spacecraft in days instead of months can only be achieved with efficient, high-confidence processes for testing components and systems at various levels of integration. Just as one can never hope to script a complex software program without test and debug facilities in compilers, a system-wide test and debug infrastructure is also appropriate for SPA. The idea of test bypass emerged as an adjunct feature of SPA, not required, but highly desirable for building a complex SPA system.

As the name implies, test bypass provides a means to suppress or bypass the inherent phenomenology of a device, using synthetic representations of the data provided either manually or through a simulation. For example, a SPA thermister could be tested by heating the device and reading the output. Alternately, if one could intercept the thermister signal and replace the actual value with a pre-selected value, then this capability can simplify testing. Test bypass codifies the mechanism for doing this across all variables expressed in the xTEDS of a device.

The basic functional diagram of test bypass, as shown in Figure 15, has been implemented in most ASIM designs and SPA devices to date. In this base implementation, a dual-ported register file is employed, one port being accessible from the ASIM’s resident processor, the other port being accessible by the test bypass interface (TBI). The implementation requires a test bypass engine state machine (which implements the test bypass function with commands sent through the Test Bypass Interface (TBI) along with a time synchronization state machine (which is synchronized through the SPA 1 Hz external interface). As the register file is written, individual registers are time-stamped through the time synchronization state machine. Phenomenology is suppressed by masking processor access to parts of the register file in which the masking bit is set. The masking is performed through the Test Bypass Interface (TBI) using the test bypass engine state machine. The same state machine has the ability to “engage” the test bypass mode. During this mode, the masking operations take effect, meaning that processor writes to the register file are ignored on the masked registers.

9.2.2 Flight Software-in-the-Loop

The Flight Software-in-the-Loop environment provides the capability to develop and test flight software prior to its integration into the spacecraft. It consists of a 6-degree-of-freedom (6DoF) simulation environment, ground software, and flight software components all running in a networked PC environment. The 6DoF simulation environment provides an extensible simulation of the spacecraft hardware components using “virtual ASIMs” (vASIMs) of reaction wheels, torque rods, coarse sun sensors, star trackers, batteries, solar arrays, and so forth and their interaction with the space environment (orbit, Earth atmosphere, solar radiation, etc.). The 6DoF simulation provides physics-based models of components that have been calibrated as much as possible to match performance data of the actual components, and includes effects such as occlusion of sensors by other components. The 6DoF simulation provides scenarios and anomaly injection into components via the Test Bypass Interface (TBI)
and the capability of logging detailed data on the simulated environment and components. The ground software provides the capability of commanding the flight software using the same interfaces as would be used in the actual satellite and also provides the capability to log telemetry from the software and emulated hardware.

Figure 15 – Test-Bypass is implemented with a dual-ported register file

The flight software running on the PC includes the SSM components (SPA Lookup Service, Network Manager, etc.), the activity application which prioritizes the activities and specific activity applications (Deploy, Charge Batteries, Take Image, Telemetry Handler, and other applications controlling experiments), subsystem controllers (which control spacecraft components), utility applications (which perform computations that support activity agents and subsystem controllers) and general applications (such as image processing).

Figure 16 illustrates the typical configuration. One PC runs the ground software and uses an Ethernet connection to send commands to and receive telemetry from a second PC that is hosting both the 6DoF simulation and the flight software.
10 Summary of SPA Standards

The effort to develop SPA has resulted in a set of standards to provide normative references for constructing hardware and software capable of interfacing with a SPA satellite system. In this stage of SPA development, the following SPA standards have been drafted. They are grouped under two categories: General standards that apply to all SPA systems and components, and application-specific Standards that allow for more varied applications of SPA to support a wider variety of needs. SPA General Standards

10.1 General SPA Standards

10.1.1 SPA Networking Standard

The SPA Networking Standard (AIAA S-133-2-201X) defines normative requirements for network topology discovery, routing, component registration, and subscription processing. It also develops the reader’s understanding of these concepts through the use of informative process sequence diagrams related to the network configuration.

The Network Standard also establishes the overall SPA network methodology, the approach to abstraction of unique transport details, and methods of communicating across multiple similar and dissimilar networks. This standard defines the minimum requirements for the components in a SPA network for the functions of network topology discovery, routing table construction and distribution, packet routing, gateway operations, and dynamic reconfiguration. It also specifies what quality of service guarantees must be provided within a SPA network, and the network security requirements.

10.1.2 SPA Logical Interface Standard

The SPA Logical Interface Standard (AIAA S-133-3-201X) describes the high-level capabilities provided by components within a SPA network. The messages, protocols, and interactions a standard SPA component will use to participate in the SPA network are defined. This standard provides the complete list of standard SPA messages; the message structure defines the exact format of the entire message and its fields as well as describes how the fields are used.
The standard message header, extended header, and footer are defined in the first section of AIAA S-133-3-201X. The messages and protocols that components must abide by to find one another and exchange data are defined in the second and third sections. These capabilities are enumerated in Figure 17. The sequence of events, commands, responses and message formats required to accomplish each of the functions shown in the figure are defined. Messages and protocols related to managing Quality of Service needs (QOS). The services provided in this section of the Standard are informative as not all of the concepts have been sufficiently tested to support a normative reference.

This document does not attempt to describe how the messages are transported from one component to the other. This document and its messages are agnostic to message routing, message delivery, or the network topology. Those details are documented in the SPA Networking Standard (AIAA S-133-2-201X).

Figure 17. Component Data Capabilities and Component Network Capabilities

A method of providing component identification is also provided. Each component has a Component Universally Unique Identifier (CUUID). No two components in any SPA system share the same CUUID. The CUUID is used by the SPA services, and by application or mission specific software to determine specific device type, manufacture date, component position, and other component specific characteristics. Its small fixed size of 128 bits makes it ideal for quick comparison and identification in a database. An algorithm has been defined such that two component vendors can independently generate the CUUID with an extremely low probability of duplication.

10.1.3 SPA Physical Interface Standard

The SPA Physical Interface Standards (AIAA S-133-4-201X) details the mechanical, thermal, and electrical connector interface requirements for SPA hardware components on a SPA-compliant spacecraft. Reporting requirements for mechanical and thermal design data, such as mass, CG, envelope, radiator and heater locations, and so forth are specified.

The standard mechanical interface is a bolted connection to a regularly spaced grid of threaded holes. The grid spacing and fastener size are specified for each particular SPA standard.

The SPA electrical connector interface consists of one or more connectors that contain provisions for power, data, a timing synchronization pulse, grounding connections, and (if specified) a Test Bypass data interface. The connector type and pin assignments are described, along with definitions of connector gender and mechanical mounting. Requirements are provided for the associated cabling, including
details of shielding, shield termination, insulation and cable impedance.

Thermal control of SPA devices is accomplished by the rejection of dissipated power to the mounting surface (conduction) or to the surrounding environment (radiation). The spacecraft provides a conductive interface for the SPA device, however, the device designer may choose other approaches such as heat rejection to space via radiators.

10.1.4 SPA Power Standard

The SPA 28V Power Service Standard (AIAA S-133-5-201X) explains the means by which power service will be supplied to SPA system components.

The voltage reference system (VRS) associated with the spacecraft power service follows single-point ground (SPG) power architecture. The SPA 28V Power Service Standard establishes specifications regarding the quality of the power service such as voltage ripple, transients, and interruptions.

The scope of that document is limited to the interface between a SPA-equipped spacecraft and SPA-compliant device. This interface is implemented at a physical SPA endpoint connector, specified in the SPA Physical Interface Standard, which contains details of the connector type, pin assignments and wiring harnesses. Details of the design of a specific spacecraft electrical power subsystem, including power sources such as solar arrays, and power storage devices, such as batteries, are not relevant to the SPA power interface described in this document.

The SPA power service is regulated 28V to 34V power provided to SPA endpoint components through the connectors specified in the Physical Interface Standard in three current range classes. Battery over-voltage and under-voltage limiting is provided by the power system to control the battery state of charge and to manage spacecraft load. Specifications for main bus voltage ripple, current ripple, bus impedance, and transients are provided. Hard and soft over-current protection is provided by the main bus power service. Device in-rush current limits and over-current fault conditions are described for each of the endpoint power levels.

Consistent with Electrical Power Systems for Unmanned Spacecraft (AIAA S-122-2007) the SPA voltage reference system for is a single point ground (SPG) configured as indicated in Figure 18. With this approach, the nominal +28V battery bus voltage return is tied to the spacecraft structure only at the negative battery terminal. When multiple batteries are used, the primary battery is used to establish the single point ground (SPG). All additional battery negative terminals are wired together. All +28V return lines flow through the power distribution modules and instrument SPA interface in order to supply a single return path for primary SPA component power. Secondary instrument power is referenced to both the instrument housing and spacecraft structure. Note that this is a voltage reference only. The minimum current flowing through instrument chassis returns and into the spacecraft structure is detailed in the SPA 28V Power Service Standard.

The single point ground (SPG) of the SPA System is located at the negative terminal of the primary SPA battery. When multiple batteries are used, the battery nearest to the power source (i.e., solar array or solar array regulator), shall be selected as the primary battery. The single point ground (SPG) is rigidly mounted to the spacecraft structure with a specified maximum impedance between the negative battery terminal and spacecraft structure. All SPA primary power circuits and components are isolated from the chassis structure and secondary power and signal circuits. Resistance values for secondary power returns are specified for surfaces used for the purpose of establishing a chassis ground.
10.1.5 SPA Timing Standard

The SPA System Timing Standard (AIAA S-133-6-201X) establishes a common method for synchronizing time across all SPA devices, processors, and applications through distribution of a time-at-tone (TAT) message and synchronization pulses.

The SPA message family supports a time-at-tone function. The SPA System Timing Standard describes the SPATimeAtTone message received before a Sync Pulse. The format of the TAT message is specified in the SPA Logical Interface Standard or a SPA Subnet Adaptation Standard TAT message can be defined, depending on the implementation for a specific system. The message and pulse are sent to all components. The message contains two different clocks for use at the component, depending on need type. The system clock contains the system time in seconds and microseconds as dictated by the System Clock Epoch Type. The System Clock can make large jumps and even go in the reverse direction. The Monotonic Clock defines an arbitrary time that only increments by 1 for every sync pulse that is sent.

Discussions on availability, latency, jitter, and drift provide general guidelines on system timing requirements which are intended to meet the needs of most systems. Systems requiring greater timing accuracy than described herein may be implemented within the Space Plug-and-Play Architecture with enhanced or modified timing provisions as necessary.

Annexes A is informational and provides information on timing disruptions, time source priority, and master clock source considerations, with descriptions of timing implementation within a reference model.

10.1.6 SPA Ontology Standard

SPA system ontology (AIAA S-133-7-201X) revolves around the extensible Transducer Electronic Data Sheet (xTEDS). Every hardware device or software application used within a SPA system must have an associated self-describing electronic data sheet that fully explains the component (device or application) to other components in the system. The xTEDS contains descriptions of all component-specific commands accepted, variables produced, and data messages that can be delivered by the component. It fully describes the services or data provided by the component and represents the complete protocol for accessing these services or data. This standard establishes the format and allowable terminology for the xTEDS required for inclusion with each component of a SPA satellite system to define the specific data interfaces to the component. The xTEDS uses XML to provide a schema-controlled language for the data sheet that allows users to define their own elements in accordance with the SPA xTEDS Schema and the W3C XML Schema.
The xTEDS uses the eXtensible Markup Language (XML) to provide a schema-controlled language for the data sheet that allows users to define their own elements in accordance with the SPA-developed xTEDS Schema and the W3C XML Schema. All xTEDS prepared for SPA implementations must be validated for conformance with the SPA xTEDS Schema and the XML Schema using some form of a “validating XML parser.”

To simplify xTEDS development, xTEDS interface templates may be employed for commonly-used interfaces, such as device power and device safety. The SPA Ontology Standard provides guidance in developing SPA compliant xTEDS and provides details on available templates.

The xTEDS identifier (XUUID) provides a unique identification of each xTEDS within the network. The XUUID is used to uniquely identify the entire xTEDS stream using a variation on a look-up table. This allows the use of the XUUID to cache xTEDS and prevent full exchange of xTEDS at system start up or discovery.

A common set of terms has been defined that are shared by all SPA applications to allow for the creation of xTEDS that are understood and accessed by components throughout the system. Descriptions of data products within data messages conform to the structure defined in the Ontology Standard and rely on profile specific terms defined under a Common Data Dictionary (CDD). Terms used in the Common Data Dictionary (CDD) must be easily recognized by the system developers, unique for each variable type, and non-duplicating. A parser used to validate the SPA xTEDS against the xTEDS schema should also test the xTEDS against the CDD to ensure only registered terms are used.

Annex A of the xTEDS Ontology Standard (AIAA S-133-7-201X) provides format and access details on the electronically-maintained Common Data Dictionary (CDD) for SPA. Device-specific names and their meanings, as well as the list of common commands, are contained in the Common Data Dictionary (CDD). The parser used to validate the SPA xTEDS against the xTEDS schema also tests the xTEDS against the Common Data Dictionary (CDD) to ensure only registered terms are used. This document is application/domain-specific.

Access information for the SPA xTEDS Schema, the XML Schema, and a validating XML parser is provided in this document.

10.1.7 SPA Test Bypass Extension

The SPA Test Bypass Extension Standard (AIAA S-133-8-201X) establishes how to implement optional test bypass functionality in SPA components to support component-level and integrated system test activities. Test bypass is the mechanism by which simulated test data may be injected into the running SPA system (or by which operational data may be extracted from various test points within the system during integration and test). This standard is applicable to all SPA devices and systems that support a bypass function for testing purposes. This document also contains the messaging protocol for test bypass.

10.1.8 SPA System Capability Standard

The SPA System Capabilities Standard (AIAA S-133-10-201X) correlates SPA core concepts, SPA system services, and the basic capabilities that are required of a SPA system (Section 5) to specific requirements derived directly from SPA core concepts. Each requirement in the capabilities document is mapped to the SPA standards where they are discussed in detail.

10.2 SPA Application-Specific Standards

10.2.1 SPA Subnetwork Adaptation Standards (SPA-X)

10.2.2 SPA-S Subnetwork Adaptation Standard

Based on the SpaceWire data transport standard, as augmented by the SpaceWire Protocol ID standard and the Plug-and-Play Protocol ID, the SPA Spacewire Subnet Adaptation Standard (AIAA S-133-9-201X)
specifies the required physical interface, with signal characteristics, for a SPA-s device.

SpaceWire is a point to point “switch fabric” transport consisting of routers and nodes. A “path routing” addressing scheme is used for SPA applications to pass packets between nodes on the network. The SpaceWire Adaptation Standard describes the implementation of SPA network features on a SpaceWire subnetwork. The physical details of SpaceWire related to signal levels, harnessing, etc. are expressed in the SpaceWire standard document ECSS-E-50-12C. This document does not discuss physical details of SpaceWire expressed in the SpaceWire standard document.

This document specifies the means by which the SPA features of networked component registration, and message routing to endpoints on a SpaceWire network are facilitated. Accomplishing this requires low-level messaging on the SpaceWire network to provide the “convergence functions” to allow the common SPA messages to be transported.

The role of a SPA subnet manager (SM-x) implemented for any transport protocol is to abstract the peculiarities of the subnetwork away, so that SPA core components and clients of those components need not know specific details to target components in the subnet for communication. To date, SPA common messaging has been bridged to several transport flavors, including USB, SpaceWire, and I2C. Research is in progress to develop the standard for Optical transport as well.

In SPA, a subnetwork manager is a construct that bridges to a protocol other than that natively supported by SPA messaging. SPA supports a native packet format and addressing protocol which is independent of the low level networks on which it is transported. In order for the SPA addressing services, data manager, and other subnet managers to access a SpaceWire subnet, there must be a broker that allows the component discovery and messaging communication functions to occur despite the fact that they are not natively supported—this is the role of a SPA SpaceWire subnet manager (SM-s).

The fundamental roles of a SPA SpaceWire subnet manager (SM-s) are described briefly in the following bullets and in greater detail within this specification document.

- Discovering all nodes on the network and their associated SPA components
- Performing any work required to configure the network infrastructure to support addressing
- Requesting to register the components of the subnet with the SPA Lookup Service on the SPA Network
- Tallying required addresses, probing other managers on the subnet for their totals, and requesting address blocks to meet those needs. It also stores assigned logical addresses for SPA components on the subnet, and can partition the address block provided and assign them to other managers.
- Distributing information (if necessary) to allow all endpoints on the subnet to route to any other endpoint on the subnet
- Mapping a SPA message received at SPA SpaceWire subnet manager (SM-s) from the SPA-L side to an endpoint on the subnet
- Releasing messages arriving from the subnet onto the SPA-L in the proper form for local routing
- Providing bridging capability when two or more adapters are installed on a host. This is the ability to pass messages from one subnet to another, regardless of the type of transport.

10.2.3 SPA-U Subnetwork Standard

Based on the USB data transport standard, a SPA-U standard specifies the required physical interface, with signal characteristics, for a SPA-U device and the Appliqué Sensor Interface data transfer protocol developed for low-data-rate devices. A SPA-U document is not included in the original SPA standards set however, a draft working version has been reasonably mature for some time. Community interest in seeing this standard published will drive future availability of a formal standard for SPA-U.
Annex A  Reference Architectures and Models

There are two external reference architectures relevant to SPA and its data-centric framework. These are the Open Systems Interconnection seven-layer model and the Consultative Committee for Space Data Systems (CCSDS) Reference Architecture for Space Data Systems (RASDS). Both contain specific abstractions and are created from particular viewpoints. SPA is constructed from a data-centric viewpoint. Therefore, it is not possible to directly map from OSI or RASDS to SPA. It is possible to compare model layers of each to SPA and obtain valuable information, which gives insight into SPA, resulting SPA standards and SPA requirements.

A.1  The Data-Centric Plug-and-Play Approach

This part about architecture starts at the OSI model and tries to illustrate a thread from OSI to SPA in spite of their different views and viewpoints. Two comparisons are given, each with strengths and imperfections in presentation (see Figure A.1).

![Figure A.1 – SPA/OSI Layer Model correspondence](image-url)
The layers in the reference SPA model are:

1. **Mission** – The mission layer addresses the system implemented by an enterprise to achieve some specific purpose. It supports end user interactions with the system without requiring the end users to know the internal details of the system. This is the highest layer in the model.

2. **Function** – The function layer addresses the functions needed by the mission system to achieve the goals of the mission (enterprise).

3. **Application** – The application layer addresses the activities performed by components in the system to address the functional needs of the mission system.

4. **Services** – The services layer addresses the activities or resources required by a set of applications to address the needs of the mission system applications.

5. **Protocol** – The protocol layer addresses the methods and formats for data exchange between applications and services within the mission system.

6. **Device** – The device layer addresses the need to provide a set of physical units to provide and support the upper layers of the model.

Because the above computing functions are commonly found on most spacecraft, there is extensive knowledge and a large experience base in the spacecraft community to address the engineering aspects of performing these functions. Unfortunately, the manner or process in which these functions can be implemented and even the vocabulary used or associated with these functions can vary significantly from organization to organization. There is no space industry equivalent of the Open System Interconnect (OSI) seven-layer stack model that addresses these other required spacecraft functions.

The space layered model above appears similar to the Open System Interconnect (OSI) Model developed in the telecommunications industry (and adapted by the CCSDS community). The OSI model is the seven-layer stack representation that was defined in the 1970s and is often used to partition and describe the architecture for communication systems. Although this particular version has limitations (e.g., TCP/IP does not map cleanly into this particular stack architecture), it is a common and useful tool used throughout the industry as a common model description. This model is also used in the space vehicle community by CCSDS where it has been applied to describe a model for space–ground and space–space communications.

Figure A.2 does a good job of mapping the SPA concepts in the SPA standards to the OSI layers. Mission and bus/payload support software are really not distinct from each other from a SPA perspective. They may have a logical hierarchy based on the data interfaces that they present.
### OSI Model and SPA Model: Comparison Two

<table>
<thead>
<tr>
<th>OSI</th>
<th>SPA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td><strong>Devices</strong></td>
</tr>
<tr>
<td>(Presentation and Session are transparent in SPA)</td>
<td>e. g. Momentum Wheels, Solar Panels, Propulsion Units, Star Trackers</td>
</tr>
<tr>
<td><strong>Transport</strong></td>
<td><strong>SPA Messaging</strong></td>
</tr>
<tr>
<td><strong>Networking</strong></td>
<td><strong>SPA Subnet Managers</strong></td>
</tr>
<tr>
<td><strong>Data Link</strong></td>
<td><strong>SPA-X Protocols</strong></td>
</tr>
<tr>
<td>Physical Addressing.</td>
<td>SPA-S, SPA-U, etc.</td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td><strong>I²C, USB, SpaceWire Sockets</strong></td>
</tr>
<tr>
<td>Media, signal and binary transmission</td>
<td></td>
</tr>
</tbody>
</table>

Figure A.2 – An OSI/SPA mapping

### A.2 CCSDS SOIS Model

To deal with spacecraft internal communications rather than space–ground and space–space communications, CCSDS generated the Report Concerning Space Data System Standards, Spacecraft Onboard Interface Services Informational Report (CCSDS 850.0-G-1), Green Book (June 2007). The Spacecraft Onboard Information Systems (SOIS) functions used for a currently theoretical and general plug and play capability for SOIS are included in Figure A.3. Further definition of the SPA architecture
shows the similarities and differences in architectural approach. Note the inclusion of both Transport and Network Protocols within a unique Transfer Layer.

Figure A.3 – SPA graphic

Another relevant part of SOIS architecture appears in diagrams like the one in Figure A. 4, which shows the internal structure of the SOIS Device and Data Access Service in the left stack. The black line at the bottom of the diagram represents a network connecting applications to devices. There are two important locations for interfaces, labeled “Application Interface” and “Manufacturer’s Interface” in the figure. The manufacturer’s interface is specific to each device, while the application interface could be standardized to represent a small set of ideal instruments. Representing idealized instruments at the application interface decouples application software from hardware choices. The concept of ideal instruments corresponds to the concept of interfaces in xTEDS.
The diagram in Figure A.5 shows the interfaces in SPA described by xTEDS, and the figure also repositions SOIS functions, so one can see the functions in terms familiar to SOIS designers, without revealing the details of SPA implementations. The SOIS DSAP stands for “Device Specific Access Protocol”, which corresponds to the concept of the device-specific part of an ASIM. The SOIS DVS stands for “Device Virtualization Service”, and it corresponds to the concept of the SPA part of an ASIM.

An important distinction between SOIS architecture and SPA becomes apparent when comparing these
two figures. SPA distributes the application interface, rather than the manufacturer’s interface. SOIS and traditional architectures distribute the manufacturer’s interface. This choice is one of the defining characteristics of SPA and is represented in the SPA Logical Interface document (AIAA S-133-3-201X). It is important to note that in the SPA protocol stacks shown in Figure A.5 services and transport layers exist but are not shown as they have no specific SPA implementation.

There is a potential to implement a traditional software architecture that is logically compatible with SPA, although not plug-and-play-compatible. The potential implementation matches at the logical interface, because it uses xTEDS to describe components at that interface. The SOIS group agreed to use xTEDS for description of the application interface.

The full-SPA implementation is significant because it breaks up the monolithic traditional architecture into portable components that plug and play through an abstract interface defined by xTEDS.