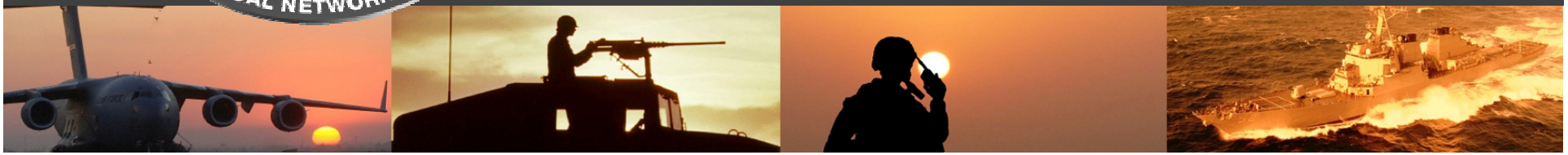
PEO C3T
PROGRAM EXECUTIVE OFFICE COMMAND CONTROL COMMUNICATIONS-TACTICAL

SCA 4.1 New Features



JTNC Standards
18 FEB 2015

PEO C3T
PROGRAM EXECUTIVE OFFICE COMMAND CONTROL COMMUNICATIONS-TACTICAL

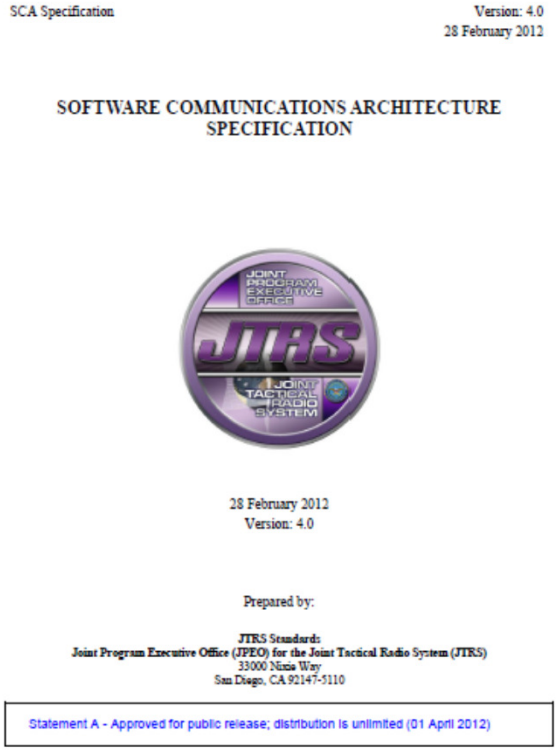
Background

JTNC

- The Software Communications Architecture (SCA) 4.1 Draft was completed on 31 DEC 2014 / approved for public release 26 JAN 2015
- The Wireless Innovation Forum (WInnF) SCA working groups contributed a wealth of valuable technical material that was incorporated within the specification
- JTNC wants to maintain the collaborative relationship with WInnF to develop future specification releases
- The JTNC Standards SCA working group integrated the WInnF and other technical proposals to produce the draft
- SCA 4.1 is scheduled for finalization in JUN 2015

Overview

- **SCA 4.1 preserves that significant technical features of SCA 4.0**
  - **Reduced boot times via Application Push Registration**
  - **Improved Information Assurance through expanded use of the least privilege pattern**
  - **Reduced overall life cycle costs facilitated by component profiles and units of functionality**
  - **Better support for model driven development with the component model**
  - **Extended support for alternate operating environments with the "CORBA neutral" Platform Independent representation**



SCA Specification

Version: 4.0
28 February 2012

SOFTWARE COMMUNICATIONS ARCHITECTURE
SPECIFICATION

28 February 2012
Version: 4.0

Prepared by:

JTRS Standards
Joint Program Executive Office (JPEO) for the Joint Tactical Radio System (JTRS)
33000 Nixie Way
San Diego, CA 92147-5110

Statement A - Approved for public release; distribution is unlimited (01 April 2012)

SCA 4.1 Structural Changes

- **APPENDIX B: SCA APPLICATION ENVIRONMENT PROFILES**
  - Incorporates a new Ultra Lightweight profile
  - Provides the detailed listing of Standard Library functions within a new attachment
- **APPENDIX E - PLATFORM SPECIFIC MODEL (PSM) - TRANSFER MECHANISMS AND ENABLING TECHNOLOGIES**
  - Renames appendix
  - Moves Object Management Group Interface Definition Language (IDL) Platform Independent Model (PIM) within Appendix E
  - Introduces new Appendix E-1 for Application Interface PIM
  - Revises Appendix E-3 to include multiple, language specific Platform Specific Models (PSM)

WInnF Proposals

- Proposal for Backwards Compatibility of SCA Applications [Use Case 1.1]
- Proposal for Scalable Components
- Scalable Manager Components
- Lightweight (Lw) and Ultra Lightweight (ULw) POSIX Application Environment Profiles (AEPs) for Resource Constrained Processors
- Interface Definition Language (IDL) Profiles for Platform-Independent Modeling of SDR Applications
- Proposal for a Naming Convention
- Proposal for SCA 4.1 Push Registration - Allocation Properties
- Proposal for SCA 4.1 Application Mixture Backwards Compatible UOF

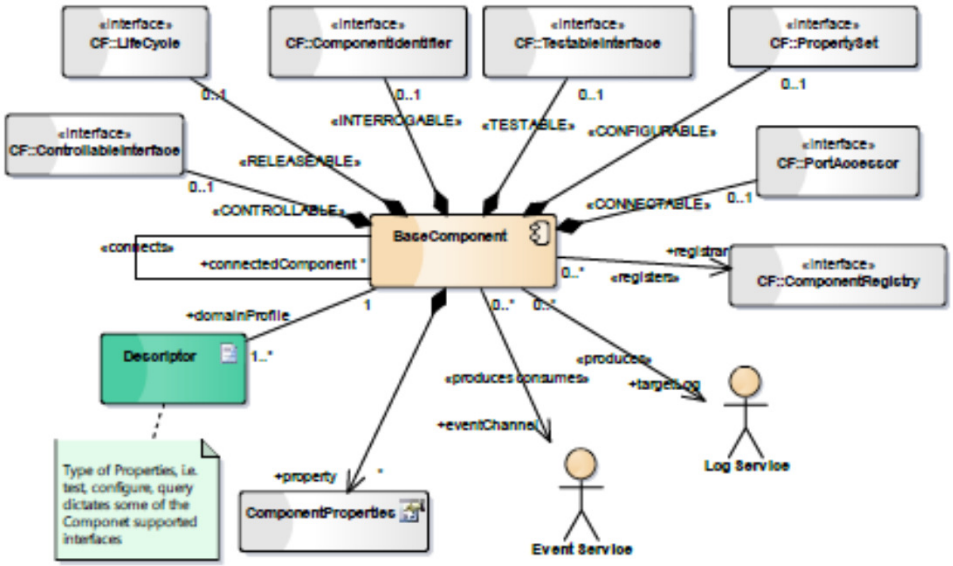
WIRELESS INNOVATION FORUM
SDR FORUM VERSION 2.0 WIRELESSINNOVATION.ORG

Backwards Compatibility

- Addresses primary SCA 4.0 comment from industry
- Preserves investment in SCA 2.2.2 compliant products
- SCA 4.1 includes an optional capability that allows a 4.1 framework to manage 2.2.2 application components
  - Core idea is that the 4.1 Core Framework will be able to manage self-contained 2.2.2 applications
  - Much of the realization of the capability is implementation specific
  - Any necessary usage guidance will be incorporated within the User's Guide
- Requirements attempt to align with the dynamic nature of the Core Framework
  - Seeking suggestions regarding how to best achieve this objective

Scalable Components

- SCA 4.1 preserves the SCA 4.0 capability that allows a system developer to eliminate requirements that are not applicable for a product line
- SCA 4.1 uses the WInnF proposed solution to achieve component scalability
  - Scalability is achieved through component level aggregations that mandate interface inheritance
  - Minimal changes were made to the requirements changes to ensure testability
  - Any necessary usage guidance will be incorporated within the SCA User's Guide
    - Consider introducing text which explains this approach relative to optional inheritance and provides strategies to minimize across the wire object size



Scalable Managers

- SCA 4.1 uses the WInnF proposed solution to achieve Manager scalability
  - Removal of the ManagerRegistry interface necessitated several changes to the UML model (ComponentRegistry functionality was expanded to handle all registration)
  - Minimal changes were made to the requirements changes to ensure testability
  - Any necessary usage guidance will be incorporated within the User's Guide
- The DeviceManager Interface was removed, which impacted the integration of several proposals

Lw & ULw AEPs

- Better aligns AEP to the specific needs of application developers
- SCA 4.1 references the WInnF specification
  - Relies on referenced specification to provide rationale
  - Preserves legacy conformance language (an area for future consideration)
- Maintains reference to 2009 POSIX specification
  - Available Real-time Operating System (RTOS) implementations / compliance will dictate the referenced version in 4.1 final
- Defines Lightweight (Lw) and Ultra Lightweight (Ulw) profiles
  - ULw – focused on minimizing the size of the platform, so it contains the minimal number of required operations (= WInnF Base Profile)
  - LW – provides a relatively full featured RTOS, yet smaller than the full AEP – includes union of WInnF group A & B operations that are a subset of a Future Airborne Capability Environment (FACE) Safety Base profile
- Preserves operations recommended by WInnF
  - Currently working one issue related to a misinterpretation of WInnF specification

IDL Profiles for Platform Independent Modeling

- Provides guidance to product developer which will allow them to implement highly portable interfaces
- Appendix E-1 contains information, equivalent to that which exists within the WInnF specification
  - Relies on the WInnF specification to provide rationale
  - Preserves legacy conformance language (an area for future consideration)
  - Includes Any type in the Full profile
    - Issue has been worked to make the corresponding change within the WInnF specification

Naming Conventions - Interfaces

- Improves specification readability
- SCA 4.1 contains several renamed interfaces
  - Preserves historical awareness and minimizes code change
  - Concentrates changes within interfaces introduced in SCA 4.0
- High level naming convention = everything without component in the name is an interface

SCA 4.0 Name	WInnF Proposed Name	SCA 4.1 Name
ComponentFactory	ComponentFactory	ComponentFactory
ComponentManager	ComponentManager	ComponentManager
ComponentIdentifier	IdentifiableInterface	ComponentIdentifier
PortAccessor	ConnectableInterface	PortAccessor
LifeCycle	InitializableInterface	LifeCycle
TestableObject	TestableInterface	TestableInterface
PropertySet	ConfigurableInterface	PropertySet
ControllableComponent	StartableInterface	ControllableInterface
Resource		N/A
Application	ApplicationManager	ApplicationManager
ApplicationDeploymentData	ApplicationDeploymentAttributes	ApplicationDeploymentAttributes
ApplicationFactory	ApplicationFactory	ApplicationFactory
DomainManager	DomainManager	DomainManager
DomainInstallation	ApplicationInstallation	DomainInstallation
DeviceManager	DeviceManager	N/A
DeviceManagerAttributes	DeviceManagerAttributes	DeviceManagerAttributes
ComponentRegistry	ComponentRegistry	ComponentRegistry
FullComponentRegistry	FullComponentRegistry	FullComponentRegistry
EventChannelRegistry	EventChannelRegistry	EventChannelRegistry
ManagerRegistry	ManagerRegistry	N/A
FullManagerRegistry	FullManagerRegistry	N/A
ManagerRelease	ReleasableManagerInterface	ReleasableManager
Device		N/A
ManageableComponent	AdministrableInterface	AdministratableInterface
CapacityManagement	AllocatableInterface	CapacityManagement
DeviceAttributes	DeviceAttributes	DeviceAttributes
ParentDevice	ChildInterface, ComposableInterface	AggregateDeviceAttributes
LoadableDevice		N/A
LoadableObject	LoadableInterface	LoadableInterface
ExecutableDevice	ExecutableInterface	ExecutableInterface
AggregateDevice	ParentInterface, AggregatableInterface	AggregateDevice
File	File	File
FileSystem	FileSystem	FileSystem
FileManager	FileManager	FileManager

Naming Conventions - Components

- **SCA 4.1 component names align with the WInnF proposals**
- **Exceptions are with constructs that were removed from the specification**

SCA 4.0 Name	WInnF Proposed Name	SCA 4.1 Name
ComponentBase	BaseComponent	BaseComponent
ComponentFactoryComponent	BaseFactoryComponent	BaseFactoryComponent
ComponentManagerComponent	??	ComponentManagerComponent
ResourceComponent		N/A
ApplicationResourceComponent	ManageableApplicationComponent	ManageableApplicationComponent
AssemblyControllerComponent	ApplicationControllerComponent	ApplicationControllerComponent
ApplicationComponent	ApplicationComponent	ApplicationComponent
ApplicationComponentFactoryComponent	ApplicationComponentFactoryComponent	ApplicationComponentFactoryComponent
AssemblyComponent	??	AssemblyComponent
ApplicationFactoryComponent	ApplicationFactoryComponent	ApplicationFactoryComponent
ApplicationManagerComponent	ApplicationManagerComponent	ApplicationManagerComponent
DomainManagerComponent	DomainManagerComponent	DomainManagerComponent
DeviceManagerComponent	DeviceManagerComponent	DeviceManagerComponent
ComponentBaseDevice	BaseDeviceComponent	N/A
DeviceComponent	DeviceComponent	DeviceComponent
LoadableDeviceComponent	LoadableDeviceComponent	LoadableDeviceComponent
ExecutableDeviceComponent	ExecutableDeviceComponent	ExecutableDeviceComponent
AggregateDeviceComponent	AggregateDeviceComponent	AggregateDeviceComponent
FileComponent	FileComponent	FileComponent
FileSystemComponent	FileSystemComponent	FileSystemComponent
FileManagerComponent	FileManagerComponent	FileManagerComponent
PlatformComponent	BasePlatformComponent	BasePlatformComponent
PlatformComponentFactoryComponent	PlatformComponentFactoryComponent	PlatformComponentFactoryComponent
ServiceComponent	ServiceComponent	ServiceComponent
CF_ServiceComponent	ManageableServiceComponent	ManageableServiceComponent

Push Registration

```
string   identifier;
string   managerIdentifier;
string   profile;
ComponentEnumType type;
Object componentObject;
CF:StringSequence supportedInterfaces;
CF::Ports providesPorts;
PropertySet     specializedInfo; // component specific
```

Removed from the proposed componentType struct

Refactored, to build upon SCA 4.0 style definitions

Revised structure and type of proposed Specialized types

- Provides more explicit identification
- Provides easy way of identifying when no info is provided
- Provides way of identifying specialized info when more than one set is provided

- **Provides framework to expand push registration capability beyond applications**
- **Provides a solution to the SCA 4.0 late registration problem**

Push Registration (cont)


```
struct AllocationPropertyType
{
  string id;
  StringSequence values;
  PropertyActionType action;
  PropertyType type;
};

typedef sequence < AllocationPropertyType >
AllocationProperties;

struct PlatformComponentInfo
{
  AllocationProperties allocationProperties;
};

struct ManagerInfo
{
  FileSystem fileSys;
  Components registeredComponents;
};
```

Removed component type specific wrapper

Renamed to eliminate component specific specialized definitions

Application Mixture

- Allows for an incremental evolution of SCA 2.2.2 compliant applications
- The JTNC Standards SCA working group and WInnF SCA working group should collaborate to discuss this proposal
- Plan is to adjudicate prior to the publication of the SCA 4.1 final specification
  - Need to come to consensus on need for this feature and preferred implementation approach
  - Agreement on Backwards Compatibility is a prerequisite for this feature
- Need to provide comprehensive usage guidance in User's Guide

PEO C3T
PROGRAM EXECUTIVE OFFICE COMMAND CONTROL COMMUNICATIONS-TACTICAL

SCA 4.1 Path Forward

JTNC

- SCA 4.1 is scheduled for finalization in JUN 2015
- Submit SCA 4.1 to DoD IT Standards Registry (DISR) as an Emerging Standard in late 2015

SCA Candidate Features

- DSP and FPGA OE
- Errors and Exceptions
- Capacity/performance model
- DSP
- FPGA
- Zero copy
- Two way performance (formerly known as One Ways)
- Static versus dynamic
- Flow connections
- Optimized protocols

Questions?

Acronyms

AEP	Application Environment Profiles
DISR	DoD IT Standards Registry
DSP	Digital Signal Processor
FACE	Future Airborne Capability Environment
FPGA	Field-programmable Gate Array
IDL	Interface Definition Language
LW	Lightweight
POSIX	Portable Operating System Interface
PSM	Platform Specific Model
RT	Real Time
RTOS	Real Time Operating System
SCA	Software Communications Architecture
SDR	Software Defined Radio
Ulw	Ultra Lightweight
UOF	Units of Functionality
WInnF	Wireless Innovation Forum