WInnF
&
SCA 4.1

Steve Bernier, NordiaSoft
WInnComm-Europe, May 2019

Agenda

About the WInnF and SCA

Highlights on SCA 4.1 support activities

Highlights on SCA 4.1 certification

Highlights on SCA 4.1 Core Features

Conclusions

About the WInnF and SCA

WInnF SDR Standards

JTNC-developed standards hosted by the WInnF

- SCA versions 2.2.2 and 4.1
- JTNC APIs

WInnF-developed Standards

- Transceiver Next (= API + Properties)
- International Radio Security Services (IRSS) API
- Lightweight and Ultra Lightweight Application Environment Profiles - (U)LW AEPs
- Time Service

SCA & PAI Issues Collection Form is available on WInnF website to collect usage feedbacks

Highlights on Support activities

SCA 4.1

Released SCA Support Activities

- SCA 4.1 Backwards Compatibility and Multi Core support
- SCA 4.1 AEP profiles
- SCA 4.1 Compliance Verification
- SCA 4.1 Applications Verification Plan
- SCA 4.1 Applications Verification Procedures

A consistent suite of standards for SCA 4.1 compliance verification emerges

- Openly elaborated within WInnF
- With direct involvement of a breadth of MoD and Industries stakeholders, including the JTNC

Transceiver Next

PIM Specification (main document)

- Released Jul 2017, outcome of Transceiver Next project
- Technology-agnostic API with extensive behaviors
- Extensive set of standard Properties for portability
- Harmonized views across EU and US / CAN stakeholders

Native C++ and FPGA PSMs

- Finalized 2018
- Enabling hybrid heterogenous developments
- Extensive specification of header files for easier usage

SCA PSM

- In finalization
- Extending coverage to mainstream SCA

Time Services Facility

Project name: Hamonized Time Services

- Target work product: WInnF Time Services Facility

Building on the success of Transceiver Next

- Harmonization
- PIM / PSM specification approach
- Standard Properties specification

Main focus

- Leveraging US, ESSOR and GE backgrounds plus industry experience on Timing Service (« what time is it? »)
- Filling existing gap on Timer capabilities (« wake me up!»)
- Fixing overlaps between API and AEP

Highlights on SCA 4.1 Certification

SCA Certification - Introduction

- **To get the benefits of the standard, some level of verification must be performed ensure compliancy to the standard**
- **For SCA v2.2.2, the JTRS Test Application (JTAP) and R-Check SCA were developed**
  - R-Check SCA is commercially available from Reservoir Lab
  - JTAP is limited to US DoD acquisition program
- **For SCA v4.1, the WInnF, at the request of the US DoD JTNC program, has define the test procedures for certification testing**
  - WG 1: Extract & document every requirement of SCA v4.1
  - WG 2: Define plan and test procedures for SCAv4.1 applications

SCA v4.1 Requirements

- **482 requirements were identified in the SCAv4.1**
- **Grouped in three different categories**

1. **"OE" category:**
    - Requirements for the software that comes with the platform.
    - Includes requirements for:
        - operating system and the SCA Core Framework, which can launch and control an SCA application.
        - communications layer between the various software components
        - standard APIs used by SCA applications to interact with devices such as the GPS, audio, network, serial ports, and the like.

2. **"AP" category:**
    - Requirements for the software that constitutes an SCA application.
    - Includes requirements for:
        - how an application needs to be implemented in order to be portable to other radio platforms.

SCA v4.1 Requirements

3. **"Both" category:**
   - Requirements that apply to both the Operating Environment software and the Application software.

Category	Number of Requirements
Operating Environment Only	396
Application Only	22
Operating Environment and Application	64
Total:	482

Requirements List

Table 1 SCA 4.1 Requirements Allocation, Objectives and Verification Criteria

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA1	The OE and related file systems shall support a maximum filename length of 40 characters and a maximum pathname length of 1024 characters.	OE	3.1.1	Establishes the maximum filename and pathname length that a file system is required to support.	1. OE and related file systems accept filenames of 40 or fewer characters. 2. OE and related file systems accept pathnames of 1024 or fewer characters.
SCA451	The OE shall provide the functions and options designated as mandatory by a profile defined in Appendix B.	OE	3.1.1	Provide a standard set of Operating System functions and standard C library functions	Verify each of the mandatory functions listed in Appendix B is available in the selected profile.
SCA452	The OE shall provide a transfer mechanism that, at a minimum, provides the features specified in Appendix E for the specific platform technology implemented.	OE	3.1.2	Specifies a standard set of transport mechanisms provided by the OE	As specified by the transfer mechanism PSM listed in Appendix E, Section 9.2, for the platform technology and its profile
SCA453	The log service shall conform to the OMG Lightweight Log Service Specification [1].	OE	3.1.2.1	Establishes requirements for a SCA Log Service.	Logging functionality conforms to the OMG Lightweight Log Service Specification.
SCA3	The OE shall provide two standard event channels: Incoming Domain Management and Outgoing Domain Management.	OE	3.1.2.2.1	To ensure the OE provides standard event channels.	The OE provides the Incoming Domain Management and Outgoing Domain Management event channels (i.e. "IDM_Channel" and "ODM_Channel").

Test Procedures

- **WInnF WG focused on Test Procedures for the 86 Application related requirements**
  - 22 requirements for Application only
  - 64 requirements for Application and OE
- **Eight groups have been identified, namely:**
  - ApplicationComponent
  - ApplicationComponentFactoryComponent
  - ApplicationControllerComponent
  - AssemblyComponent
  - BaseComponent
  - BaseFactoryComponent
  - ManageableApplicationComponent
  - Operating System

Test Procedures - Example

Table 14: Steps to execute Test Procedure for SCA156

Step	Action	Expected Result
1	Locate the <assemblycontroller> element within the SAD.	The <assemblycontroller> element is found within the SAD.
2	Retrieve the id of the next componentinstantion or assemblyinstantiation reference within the <assemblycontroller> element.	The id of the next instantiation reference is obtained or the verification will terminate.
3	Locate in the SAD a <componentinstantiation> element or an <assemblyinstantiation> element with a value of the ID attribute equal to the id retrieved in step 2.	A matching <componentinstantiation> element or <assemblyinstantiation> element is found and go to step 2.

Highlights on SCA 4.1 Core Features

SCA 4.1 Highlights

Component scalability

- Allow component developers to choose whether or not to implement some of the standard sub-component interface. The scalability will also be used to support the different profiles of the specification

Scalability of the manager components

- Allow developers to choose whether or not to implement all of the manager interfaces. The manager scalability will also be used to support the different profiles of the specification

Minimal Ultra-Lightweight AEP definition

- Provides minimal ULw specification with optional grouping to extend capability

SCA 4.1 Highlights

Major Improvement in boot time via *push registration* of SCA *Devices*

- Registration of *Devices* and *Services* is performed in one call using digested information
- With SCAv2.2.2, registration of each Device to a *DomainManager* requires a minimum of 19 interactions
  - Requires the copy of XML files for every Device which can be slow

Test Scenario	Standard Registration	One call Registration	Improvement
Linux Desktop, 1 *Device*	0.56 sec	0.19 sec	~ 66%
Linux Desktop, 4 *Devices*	1.53 sec	0.24 sec	~ 84%
LynxOS PPC405, 1 *Device*	0.86 sec	0.13 sec	~ 85%
LynxOS PPC405, 4 *Devices*	2.33 sec	0.22 sec	~ 91%

SCA 4.1 Highlights

Support for process-space collocation and core assignment

- Major footprint savings
- Major speed improvements

Major footprint Savings:

- SCA components are made of SCA interface implementation, CORBA stubs and skeletons, and OS system calls
- Same kind of SCA components are largely made of the same pieces
- Up to 70% of the pieces are the same for all SCA Resources
- Process Space collocation provides significant footprint savings
- Footprint savings translate in faster boot time with slow file systems

SCA 4.1 Highlights

Major Speed Improvements:

- Process Space Collocation performance measurements*

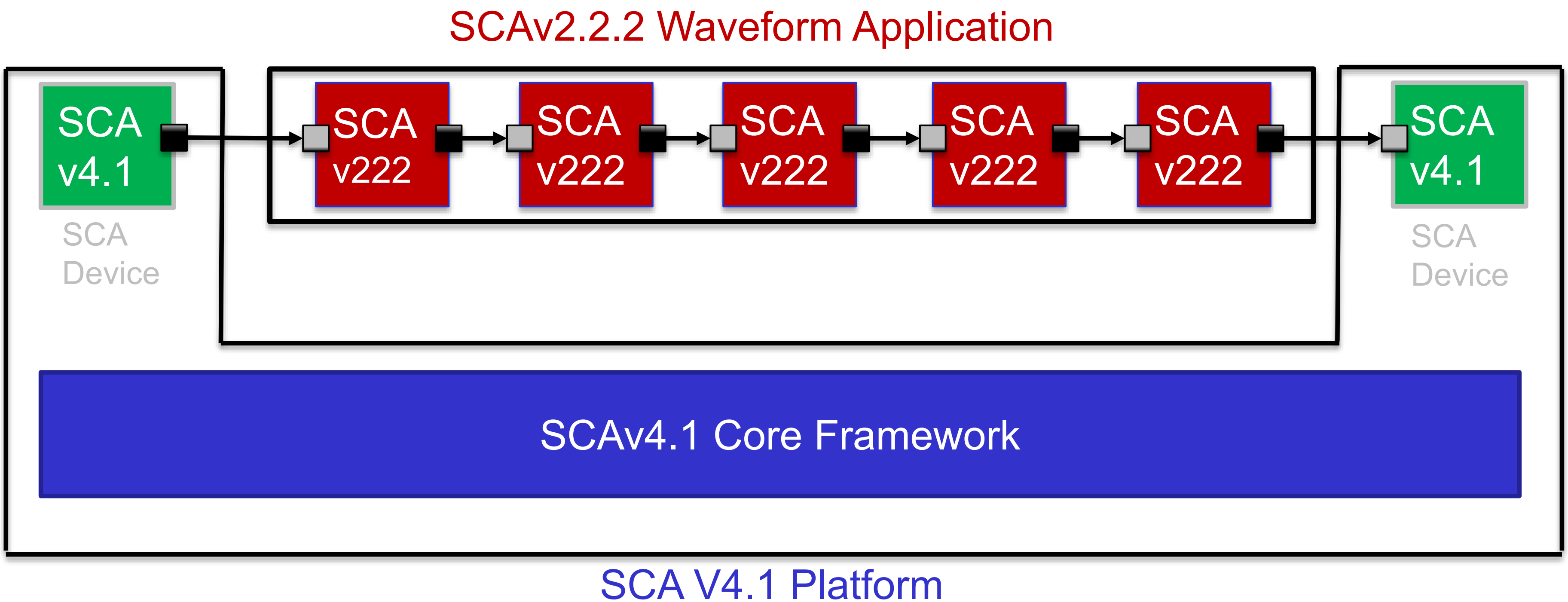
	Double Sequence		Octet Sequence	
Number of Elements in the sequence	1024	2048	1024	2048
Average Round Trip Time for PPC405/INTEGRITY using TCP/IP (usec) Using separate process spaces	3334	7272	1428	1767
Average Round Trip Time for PPC405/INTEGRITY using INTCONN (usec) Using separate process spaces	2215	4728	1042	1273
Average Round Trip Time for PPC405/INTEGRITY using direct method invocation thanks to process space colocation (usec)	244	492	155	231

* The following table presents metrics gathered using ORBexpress while running a ping test using different pluggable transports. Does not require changing a single line of source code to switch transport

SCA 4.1 Highlights

SCAv4 is backwards compatible to SCAv2.2.2

- Launch SCAv2.2.2 applications on a SCAv4.1 platform



Conclusions

Conclusions

The SDS Committee actively works and delivers

- Suite of specifications supporting SCA 4.1 verification
- Facility standards
  - Transceiver Facility V2
  - Time Services Facility
- Energy Management API

Some new challenges

- Furthering a SDR Standards Usage and Certification agenda
- Encourage prototyping of developed Standards
- Supporting NATO in implementation of its Waveform Policy

SCAv4.1 provides better performances and is fully backwards compatible with SCA 2.2.2 applications

Thank you for your attention
Questions?

<u>Contact:</u>

Steve.Bernier[at]NordiaSoft.com