SCA 4.1: a promising future for the SDR eco-system

Antoine SCHINDLER, Adrien DUPREZ, Eric SALIBA, Eric NICOLLET

WInnComm 2015 SCA 4.1 Workshop – 26 mar 2015 – San Diego, CA

Corporate Profile & SCA Experience

Thales

Radio Communications Products Business Line

A world leader with global market presence
1B€ Revenues





SDR TACTICAL
COMS
COMMS EW
SATCOM
NAVAL COMMS
COMS, NAV, IFF

CONTACT

CONTACT Programme
The largest European SDR programme using SCA

ESSOR

ESSOR Programme

MBITR2
2ch HH
JEM

MBITR 2

FLEXNET
1st International SDR/SCA with Multi-WF Porting experiences

SVFuA Programme
Porting SEM WF into SVFuA

Worldwide experience of SDR and SCA Stds

- Thales in **France**, with International Product Line and France Contact Programme development
- Thales in **Germany**, with participation in German SVFuA program
- Thales in the **US** (TDSI), delivering radios to US DoD

Leading Innovation in SDR Products and Systems

- Actively supporting the standardization efforts, including SCA
- Developing tools to support cooperation based on SDR standards

Thales committed to SCA for Waveform Portability Business Models in SDR Platforms

Introduction

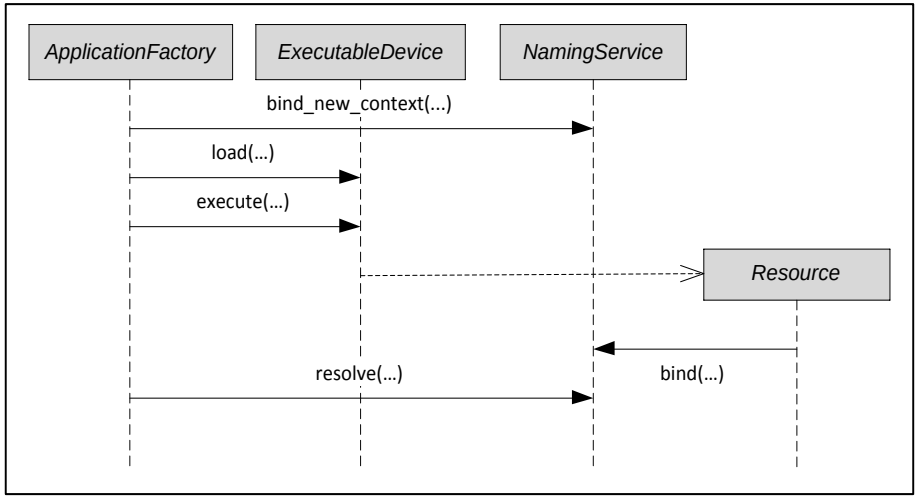
- **Draft SCA 4.1 released beginning of Feb 2015**
- **A leaner standard mature enough to become the new reference**
  - Almost 10 years after 2.2.2 publication
  - Taking advantage of extensive usage experience
- **Aims of the article is to evaluate the anticipated benefits regarding**
  - Deployment performance
  - Waveform portability
  - Optimization of development costs
  - Security
  - Scalability
- **Considerations are provided based on**
  - Analogies with results of earlier prototyping for SCA 2.2.2 SWaP optimization
  - General expertise for secure SDR solutions development
  - Long-standing involvement in support of a Standards-based SDR ecosystem
  - Leadership position in development of 2 WInnF Standards used by SCA 4.1

Deployment performance – Component loading

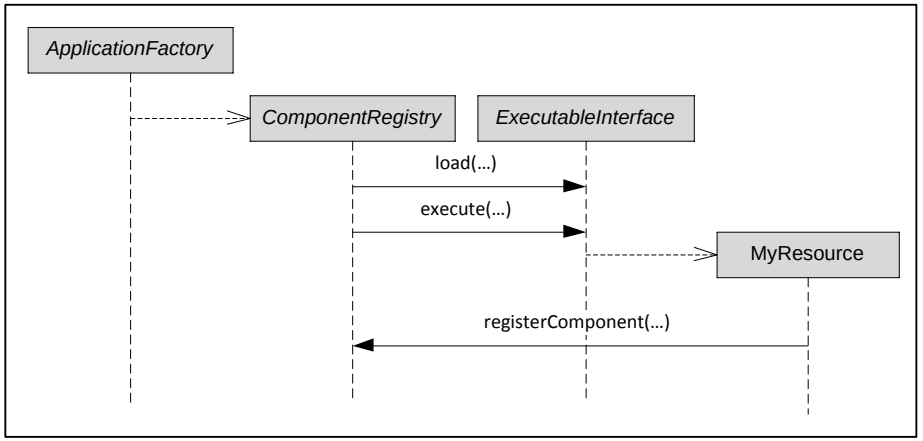
- Most important in boot time and application instantiation is spent in file system accesses
- Reduction of the size of loaded binaries is therefore helpful in reducing boot time
  - Especially if retrieved from an encrypted file system
- SCA 2.2.2 requires WF components to implement all operations of the PropertySet interface, with associated behavior (e.g. exceptions), even if not used
- The “Optional inheritance” of SCA 4.1 enables to avoid implementation of unused operations
- For instance: a simple DeviceComponent can now implement only 1 operation instead of 7 attributes + 10 operations

Deployment performance – Execution and registration

- **CORBA Naming Service is not used by SCA 4.1**
- **Replaced by « push model » registration approach**
- **Components now directly registring to its manager instead of using Naming Service as intermediate**
- **Deployement time savings expected thanks to**
  - No wait time for the Component to be registered
  - To time spent in loading, execution and initialization of the Naming Service



SCA 2.2.2 execution and registration



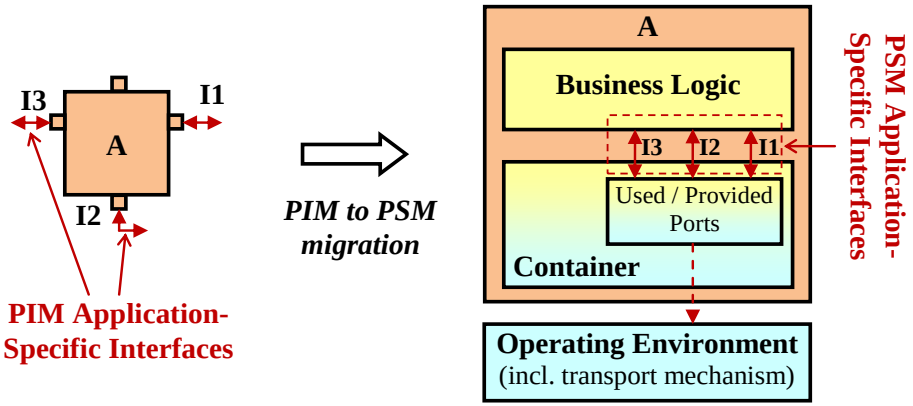
SCA 4.1 execution and registration

Deployement performance – Component deployment

- **« Push model » avoids some XML profiles to be parsed twice**
  - A Device Manager parses the XML of a Component it manages
  - It stores the associated information
  - It directly delivers the associated information to the Domain Manager when performing its own registration
  - While according to SCA 2.2.2 the same information would have been parsed again
- **« Push model » reduces the number of interactions for connection of components**
  - 3 operation calls required in SCA 2.2.2, involving the interface *PortSupplier*
  - 2 operation calls required in SCA 4.1, using the interface *PortAccessor*
  - The associated saving is to be multiplied by the number of components and connection among components

Portability – PIM IDL Profiles (1)

- **A restructured Appendix E**
  - Appendix E renamed into “Model Driven Support Technologies”
  - Addition of new Appendix E-1 “Application Interface Definition Language Platform Independent Model Profiles”
  - Fully endorsing the WInnF Standard “IDL Profiles for Platform-Independent Modeling of SDR Applications”
- **Brings essential value for better portability of SDR Applications**
- **Support of PIM to PSM migration**



  - Highly consistent with ESSOR reports on its methodology for WF developments

Portability – PIM IDL Profiles (2)

Portability benefits

- Clear separation between the Business Logic and the Container
  - Business Logic developed independently from platform assumptions
  - Enables to select CORBA or alternate transport mechanism
- PIM models of SDR Application enable more consistent designs
  - SDR Application porting conducted on a component-wise approach
  - No need for modification of the functional behavior during porting
- Choice of programming language for Business Logic independent from the PIM
  - Flexibility in programming options
  - Decisive in PHY Layers, where a same PIM component can be developed for DSP or FPGA while maintaining the overall consistency

Portability – Extension towards DSP and FPGA (1)

- **A consistent basis for GPP Component Model**
  - Connectivity: Minimum CORBA mandated by SCA 2.2.2, PIM + CORBA PSM in SCA 4.1,
  - Operating System: POSIX AEP until SCA 4.0, “Full” POSIX AEP Profile in 4.1
  - Reconfiguration Support: interface *Resource* of SCA 2.2.2, equivalent optionally inherited interfaces of SCA 4.1
- **Extension in SCA 4.1 enable to address DSP and FPGA OEs**
- **Connectivity: flexible and consistent approach**
  - PIM Level: the ULw PIM IDL Profile provides optimal solution for DSP and FPGA
    - Leveraging WInnF Standard content, and ESSOR contribution “IDL Profile for DSP and FPGA”
  - PSM Level: clear allowance for usage of an unlimited set of Connectiviy mechanisms
    - Standard integrated solutions: CORBA ; standard raw solutions: MHAL Connectivity / MOCB
    - Proprietary / Technology-dependent solution possible

Portability – Extension towards DSP and FPGA (2)

- **Mature POSIX-based AEPs**
  - WInnF Standard "Lw and ULw AEPs for Resource Constrained Processors" are essentially endorsed by SCA 4.1 Annex B
  - Brought important improvements to previous achievements for DSP RTOS access
  - Leveraging contributions from ESSOR on "ESSOR DSP AEP"
- **The remaining gap: Reconfiguration Support**
- **Portability benefits**
  - Enables DSP and FPGA environments to be consistently addressed
  - Significantly expands the boundaries of what SCA can bring to future products, while SCA 2.2.2 proved to be de facto limited to GPP environments
  - Enables significant increase of the proportion of SDR applications being designed with high degree of portability
  - Enables conformant SDR platforms to more easily host SDR Components in their DSP / FPGA processors

Optimized Applications Development Costs

- **Backwards compatibility**
  - An optional UoF (Unit of Functionality) for the Core Framework
  - Allows a SCA 4.1 framework to remain capable to manage SCA 2.2.2 components
  - Opens for adoption of SCA 4.1 on new platforms without adding the costs, risks and schedule barriers associated to simultaneous migration of SDR Applications
  - End-term perspective is SCA 4.1-only solutions
- **Leaner SDR Applications development cycles**
  - Earlier defects detection: one acknowledge benefit of PIM/PSM development paradigm
  - Easier introduction of code generation: in addition to possible usage of CORBA, usage of alternate code generation solutions is made possible, enabling to take advantage of platform-specific solutions
  - Simplification of the test phases
    - Reduction in the number of requirements
    - Rephrasing to facilitate usage of static code analysis tools

Security

- **Improvement in equipment integrity and software assurance**
- **SCA 4.1 components registrations are done in a single transaction**
  - In SCA 2.2.2, Naming Service was used, enabling a component to sniff and use any object reference in the same Naming Context
  - Naming Service was a vulnerability deserving additional protection mechanisms
- **SCA 4.1 imposes restrictions to information discovery**
  - The only Domain Manager interface now available to Components is the interface enabling registration
  - Other services of the Domain Manager are accessed based on a need-to-know approach
- **Static port connection enables platform stability and control**
- **Conditional inheritance limits complexity and "attack surface" in only providing the required interfaces**
- **All previous features are identified as helpful in perspective of security evaluations and accreditations**

Scalability (1)

- **SCA 2.2.2 was one-size-fits-all, insufficiently taking into account the diversity of possible usage contexts**
- **SCA 4.1 provides significant scalability improvements**
- **Core Framework Profiles**
  - Lightweight, Medium and Full profiles defined for Core Framework capability
  - Optimize the capability / constraints ratio of deployment and configuration
- **Connectivity/CORBA Neutrality**
  - Mandate for CORBA usage was an important barrier for adoption
  - CORBA-neutrality provided by PIM description removes this barrier
  - Maintaining past investments and performance of CORBA-based solutions

Scalability (2)

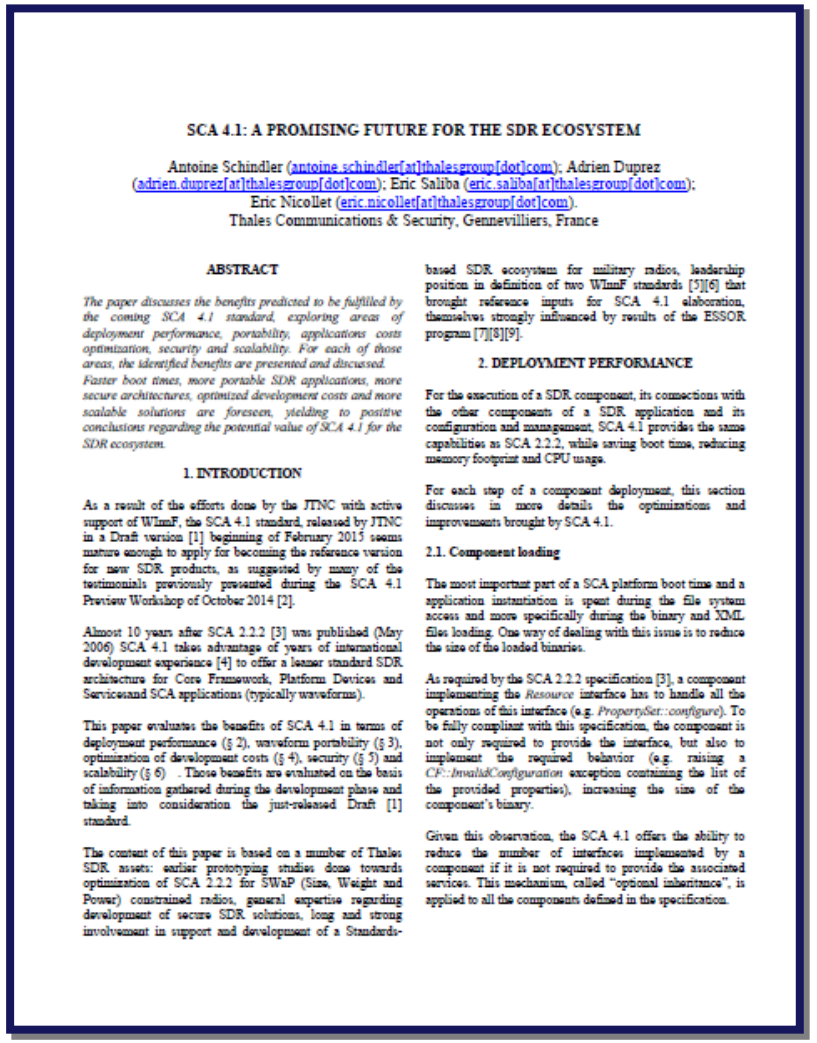
- **Optional inheritance**
  - Enables components to be tailored to their strict needs
  - Facilitates adoption in avoiding to get familiar with not needed interfaces
- **Nested Applications**
  - Enables to make application composed of other applications
  - Eliminated the combinatory explosion of systems with multiple options for different segments
  - Facilitates integration of new capabilities within systems
  - SCA 4.1 more suited to handling of complex systems than SCA 2.2.2

Conclusions

- **Various areas in which SCA 4.1 is expected to deliver key value**
  - Faster boot times, thanks to simplification of components deployment mechanisms
  - More portable SDR Applications, thanks to support of PIM/PSM designed paradigms coupled to high flexibility in choice of implementation options
  - More secure architectures, thanks to suppression of the most vulnerable parts of the previous architecture and avoidance of non-required interfaces
  - Optimized development costs, thanks to leaner overall architecture and simplification and automation of the testing phases
  - More scalable solutions, enabling to better adapt the designs to requirements and platform constraints
- **SCA 4.1 has the potential to bring decisive value to bring forward the SDR ecosystem**
  - More efficient for existing areas where SCA is used
  - Making SCA much more attractive for new adopters
- **Implementation reports will enable to move the knowledge basis on SC 4.1 benefits from previsions to implementation results**

Getting more accurate…

- This presentation reflects a fully written article
- Both the presentation and the article are available within conference proceedings
- Zip file with all 14 references available upon demand (ask presenter)



SCA 4.1: A PROMISING FUTURE FOR THE SDR ECOSYSTEM

Antoine Schindler (antoine.schindler[at]thalesgroup[dot]com); Adrien Duprez (adrien.duprez[at]thalesgroup[dot]com); Eric Saliba (eric.saliba[at]thalesgroup[dot]com); Eric Nicollet (eric.nicollet[at]thalesgroup[dot]com).
Thales Communications & Security, Gennevilliers, France

ABSTRACT

The paper discusses the benefits predicted to be fulfilled by the coming SCA 4.1 standard, exploring areas of deployment performance, portability, applications costs optimization, security and scalability. For each of those areas, the identified benefits are presented and discussed. Faster boot times, more portable SDR applications, more secure architectures, optimized development costs and more scalable solutions are foreseen, yielding to positive conclusions regarding the potential value of SCA 4.1 for the SDR ecosystem.

1. INTRODUCTION

As a result of the efforts done by the JTNC with active support of WInnF, the SCA 4.1 standard, released by JTNC in a Draft version [1] beginning of February 2015 seems mature enough to apply for becoming the reference version for new SDR products, as suggested by many of the testimonials previously presented during the SCA 4.1 Preview Workshop of October 2014 [2].

Almost 10 years after SCA 2.2.2 [3] was published (May 2006) SCA 4.1 takes advantage of years of international development experience [4] to offer a leaner standard SDR architecture for Core Framework, Platform Devices and Servicesand SCA applications (typically waveforms).

This paper evaluates the benefits of SCA 4.1 in terms of deployment performance (§ 2), waveform portability (§ 3), optimization of development costs (§ 4), security (§ 5) and scalability (§ 6) . Those benefits are evaluated on the basis of information gathered during the development phase and taking into consideration the just-released Draft [1] standard.

The content of this paper is based on a number of Thales SDR assets: earlier prototyping studies done towards optimization of SCA 2.2.2 for SWaP (Size, Weight and Power) constrained radios, general expertise regarding development of secure SDR solutions, long and strong involvement in support and development of a Standards-based SDR ecosystem for military radios, leadership position in definition of two WInnF standards [5][6] that brought reference inputs for SCA 4.1 elaboration, themselves strongly influenced by results of the ESSOR program [7][8][9].

2. DEPLOYMENT PERFORMANCE

For the execution of a SDR component, its connections with the other components of a SDR application and its configuration and management, SCA 4.1 provides the same capabilities as SCA 2.2.2, while saving boot time, reducing memory footprint and CPU usage.

For each step of a component deployment, this section discusses in more details the optimizations and improvements brought by SCA 4.1.

2.1. Component loading

The most important part of a SCA platform boot time and a application instantiation is spent during the file system access and more specifically during the binary and XML files loading. One way of dealing with this issue is to reduce the size of the loaded binaries.

As required by the SCA 2.2.2 specification [3], a component implementing the *Resource* interface has to handle all the operations of this interface (e.g. *PropertySet::configure*). To be fully compliant with this specification, the component is not only required to provide the interface, but also to implement the required behavior (e.g. raising a *CF::InvalidConfiguration* exception containing the list of the provided properties), increasing the size of the component's binary.

Given this observation, the SCA 4.1 offers the ability to reduce the number of interfaces implemented by a component if it is not required to provide the associated services. This mechanism, called "optional inheritance", is applied to all the components defined in the specification.

End of the presentation

Thank you for your attention

Contact: eric.nicollet@thalesgroup.com

Full 7 pages article is available in conference proceedings