Joint Program Executive Office
Joint Tactical Radio System

SCA Next – An Adaptive Architecture



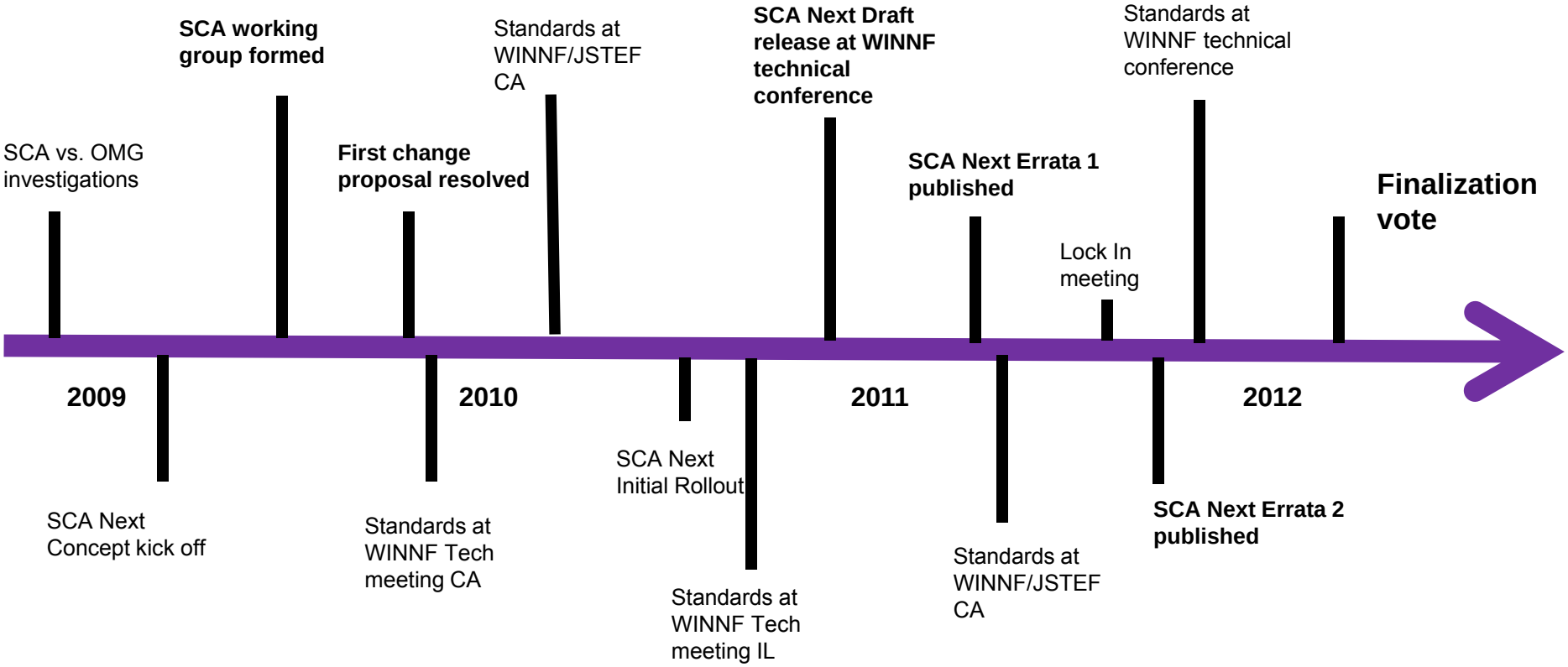

28 Feb 2012

JPEO JTRS

Welcome

- JPEO JTRS will hold a formal vote today on the SCA specification to determine whether the candidate specification will be adopted as the new SCA version.
- SCA "Candidate Release" 27-Dec-2012
  - Available via SCA public website: http://www.public.navy.mil/jpeojtrs/sca
- If approved today a new version of the SCA will be released to JPEO Public Affairs Office and posted to the SCA public website before 31-Mar-2012.
  - SCA Version: 4.0, 28-February-2012
  - Incorporates adjudication of comments received before 28-Jan-2012
- All comments adjudicated and no major issues identified

SCA Next Timeline

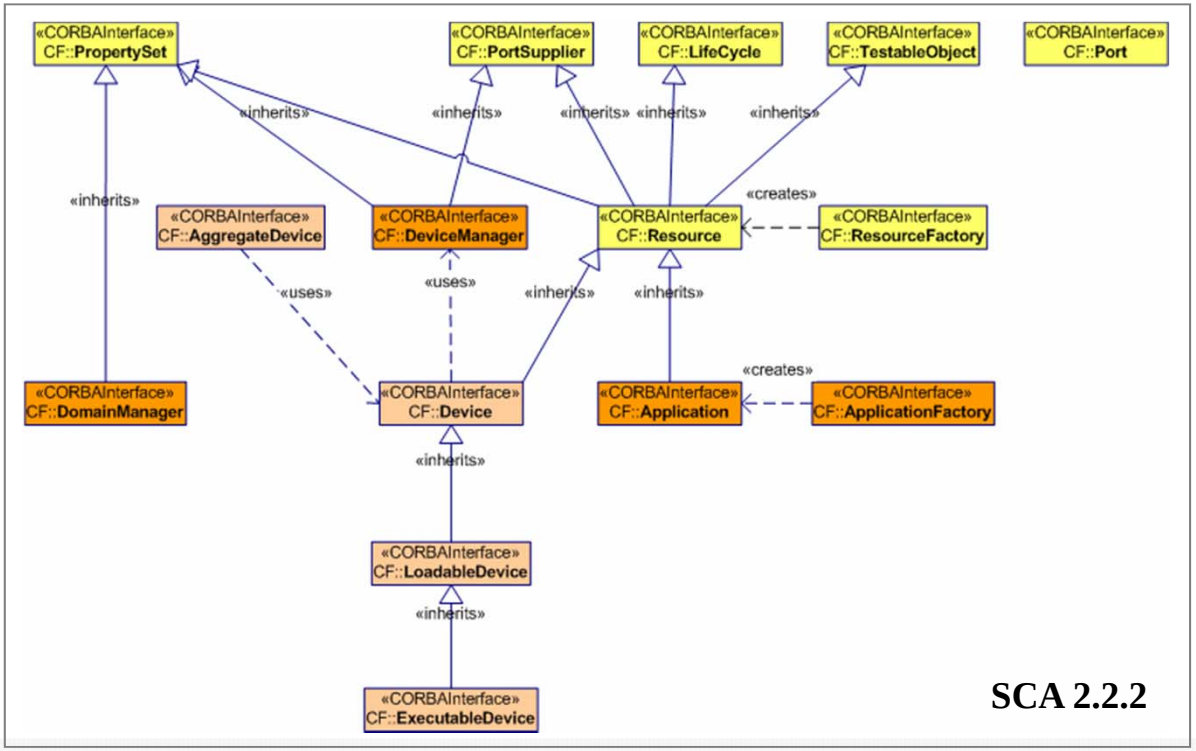


SCA Next is the byproduct of a dedicated team and extensive collaboration

Early Rationale for SCA Next

- While there have been no major deficiencies identified, optimizations have been identified
- Technology has changed dramatically since the inception of the SCA
- With the breadth of potential SCA based target platforms and applications we're looking for ways to broaden its applicability

FINAL / 15 May 2006
Version 2.2.2



SCA 2.2.2

2009 Proposed Revision Topics

Bold Items were considered during the specification process

- SCA ubiquity
  - **SCA PIM**
- Technology Refresh
  - **CORBA/e**
  - **CORBA Services**
  - Retirement of dtd files
  - **Integration of OMG PIM and PSM into SCA**
- Policy Modifications
  - Distribution of XMI files
  - **Incorporation of Appendix D requirements (evaluated, not incorporated)**
  - **Introduction of new compliance profiles**
  - **Definition of multiple SCA profiles**
  - Future of the MHAL
- SCA model modifications
  - SCA Extension expansion for devices
  - **Removal of file operations (evaluated, not incorporated)**
  - **Expansion of set up and tear down semantics (evaluated, not incorporated)**
  - Inclusion of Set management capabilities
  - **Formalization of external port connectivity**
  - **Naming Service / Domain Finder integration**
  - **Expansion of the descriptors to allow for nested applications**

What is the SCA?
This is What SCA Critics have said:

- A frame… …ware Def… Radios
- Goal: Provid… …
  - Deployment
  - Mana…ement
  - Conn… …
  - …
  - …n embedded, …nication systems.
- Addi… …izing softwar… applicati…
  - Portability - of wavefo… platforms
  - Reusability - of software ac… …orm…
  - Scalability - of ha…ware and so…ware
  - …rcial protocols and …ial acceptance

through… pr…

SCA exponentially increases development costs

SCA is too difficult to learn

I can do it …. without SCA

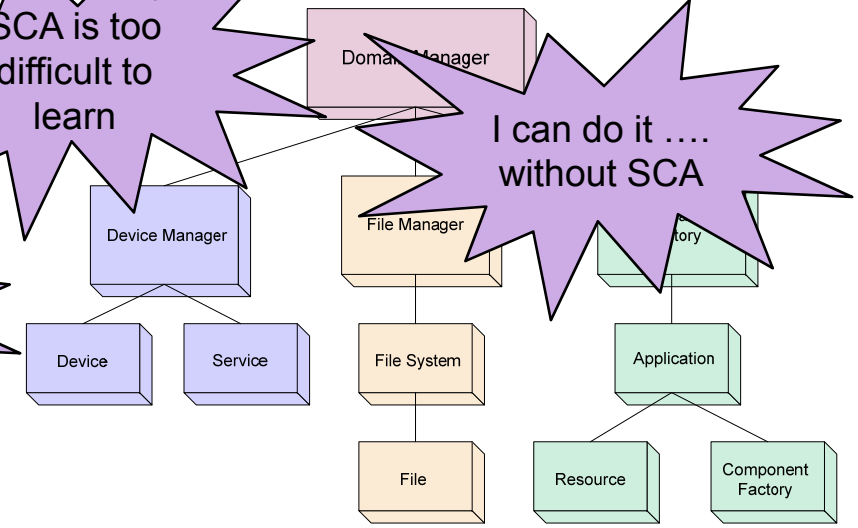
SCA is GPP centered

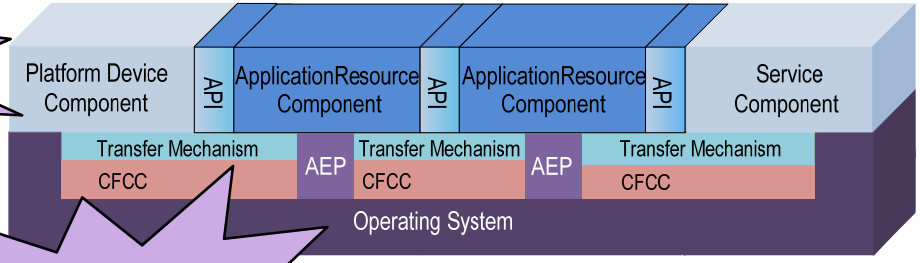
SCA can't meet "my" performance numbers

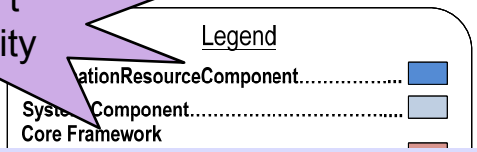
CORBA destroys my architecture

SCA is too big/power hungry … for a handheld

SCA doesn't help portability



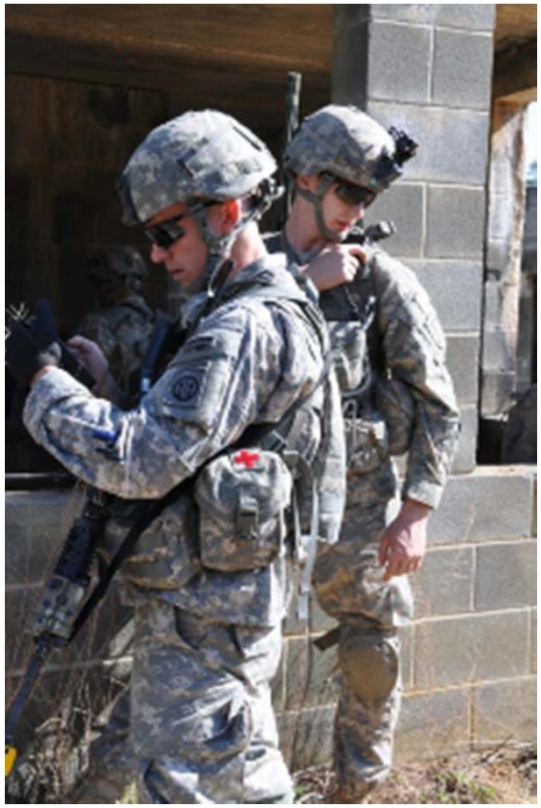




JTRS standards (developed with industry) promote flexibility/scaling and faster development without sacrificing the advantages of open technology.

SCA Next Benefits for Radio Sets – Focus on Long Term Viability

- **Longer Battery Life**
  Software components can be smaller and consume less processing resources. Right-sized middleware and feature sets further reduce processing requirements.
- **Faster Boot Times**
  New registration model reduces communication and even permits booting an image of precompiled, pre-connected waveforms.
- **Better Information/Assurance**
  Design patterns and strategies incorporate security awareness.
- **Lower Cost Radios**
  Potential for reduced number of processors and memory. Fewer requirements may result in less software development, maintenance and testing.
- **Faster Software Development**
  Component model improvements allow better exploitation of development processes and commercial development tools.
- **Innovation to the Warfighter**
  The scalability, extensibility, and flexibility promote creative solutions for a breadth of target platforms. Open standards maintain portability and interoperability of waveforms, permitting radio developers to focus on the radios, not on the waveforms.
- **Waveform/Net Centric Focus**
  SCA provides waveforms and networking with interconnection to presentation/user software and platforms such as Android and iPhone.
- **Supports a Wide Variety or Target Platforms**
  Provides a balance of standardization and radio/mission tailoring.



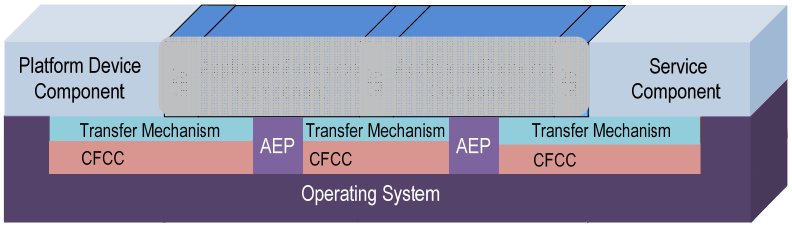
Lower Cost Radios – SCA OE Profiles (New feature of SCA-Next)

- **SCA Lightweight Profile**
  - Suited for radio platforms where the hardware modules have a static configuration.
  - Provides a minimum set of functionality which is applicable for resource (e.g. SWAP) constrained platforms.
- **SCA Medium Profile**
  - Suited for radio platforms with plug-and-play but not removable hardware modules.
  - Still rather lightweight but it introduces a configurable, dynamic aspect.
  - The most flexible platform in that it provides the lightest weight implementation that supports the legacy SCA deployment model.
- **SCA Full Profile**
  - Suited for radio platforms with removable, plug-and-play hardware modules.
  - Provides the full breadth of SCA deployment and management capabilities
  - Aligned to support prime power, multi-channel sets

SCA Next allows platform vendors to select which features are supported to meet their program's mission/schedule/cost without impacting portability or interoperability

Innovation to the Warfighter – Adapting an OE Profile

- Align your design and mission by using optional Units of Functionality



SCA Next Operating Environment (OE)
- less the shaded portion

SCA Next OE Profiles

SCA Next OE Optional Units of Functionality

Full 269

- Management Un-registration
- Management Releasable

Medium 259

- Management Registration

Lightweight 250

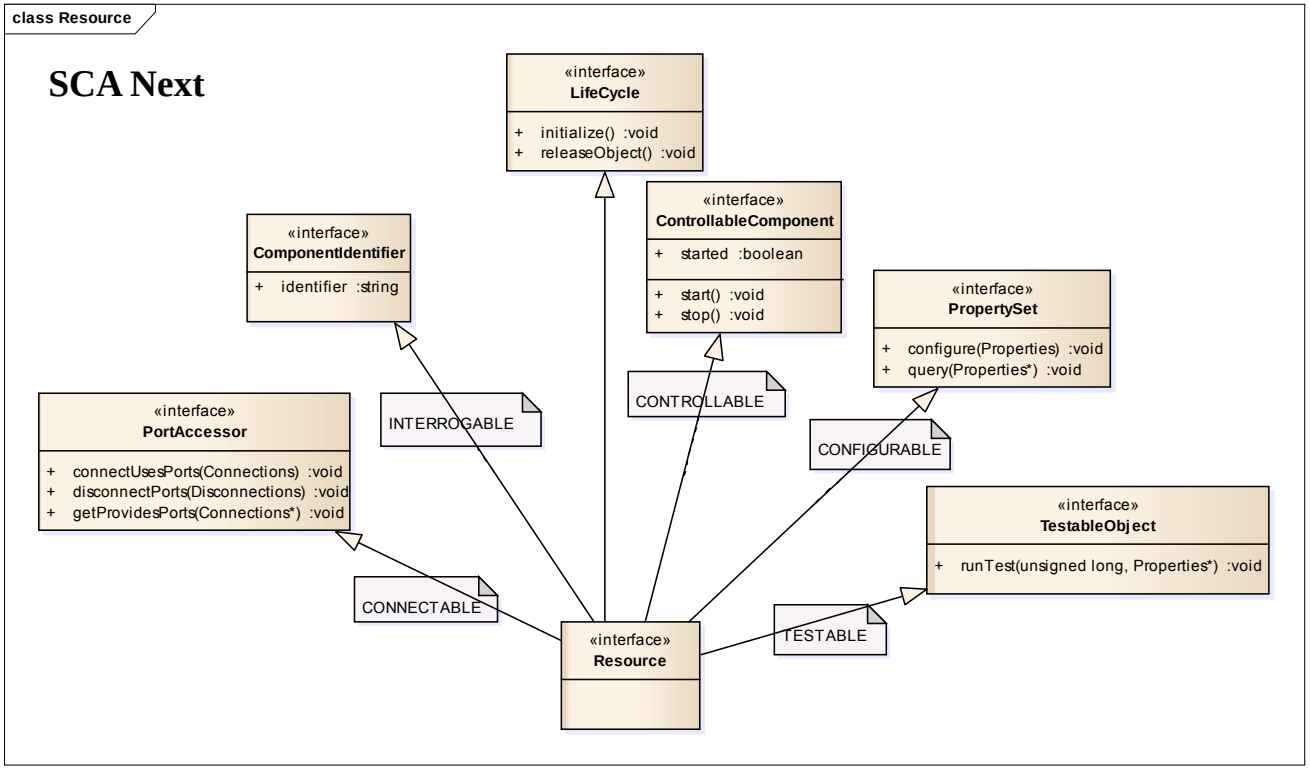
- AEP Provider
- Deployment

Optional + up to 226

- CORBA Provider
- Channel Extension
- Event Channel
- Log Capable
- Application Installable
- Log Producer
- Nested Deployment
- PlatformComponentFactoryDeployment

Key:
Estimated # of associated requirements to implement

Lower Cost Radios – Lightweight Components in SCA Next



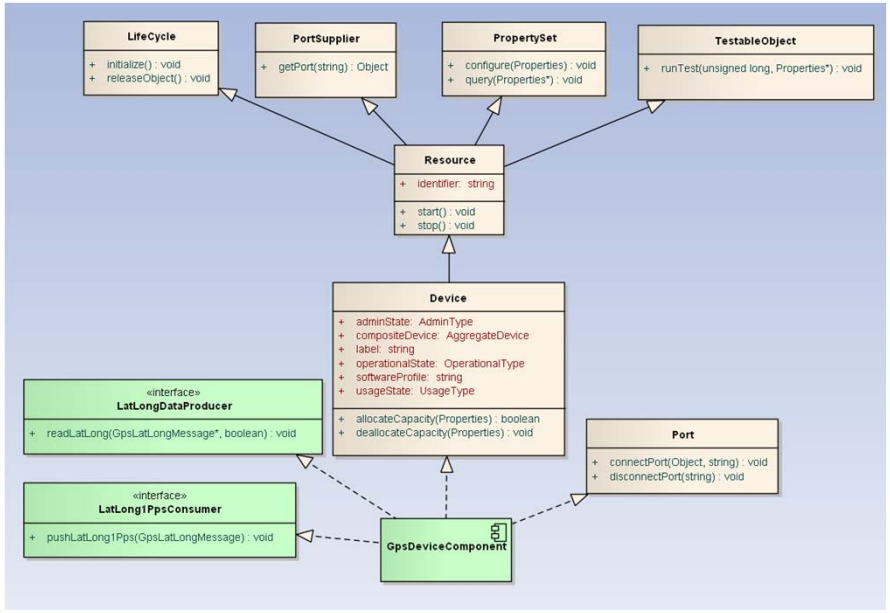
- Earlier versions of the SCA had a one-size fits-all model
- An SCA 2.2.2 CF::Resource had to implement all of its inherited interfaces whether or not they were needed

SCA Next permits tailoring inherited interfaces, which reduces software development and maintenance

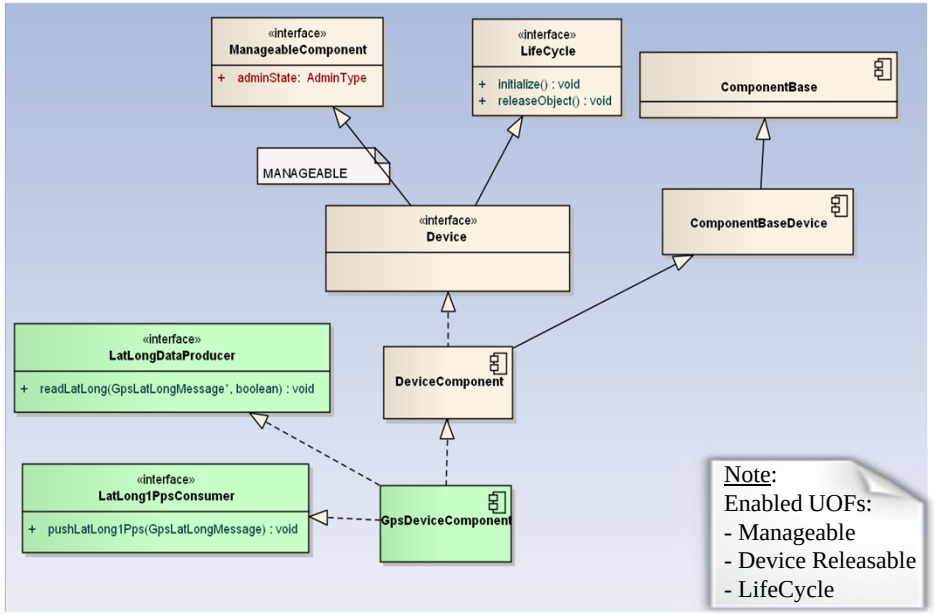
Example OE Component
SCA 2.2.2 vs. SCA Next

For a comparison, a GPS Device was designed for SCA 2.2.2 and SCA Next

SCA 2.2.2



SCA Next

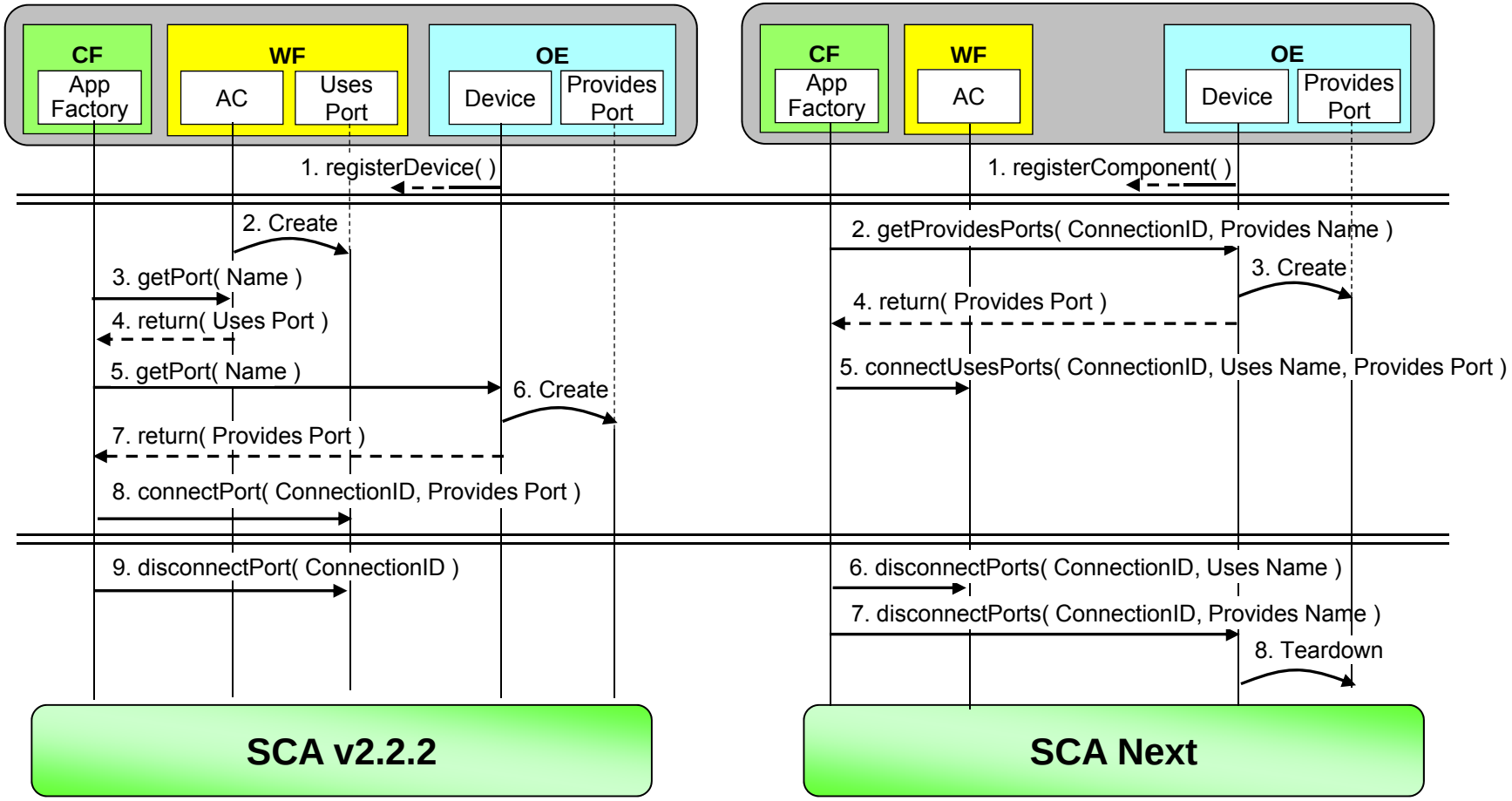


Number of ‘shall’s for the component

SCA 2.2.2	**69**
SCA Next	25

SCA Next reduces software development life cycle costs by providing a standardized approach to remove requirements

Faster Boot Times – SCA Next Port Connection Improvements



SCA Next reduces the waveform startup activity by almost 50 percent (Boot-time is proportional)
SCA Next permits connections to be defined at build time (Reduces startup and security issues)

Information Assurance Enhancements in SCA Next

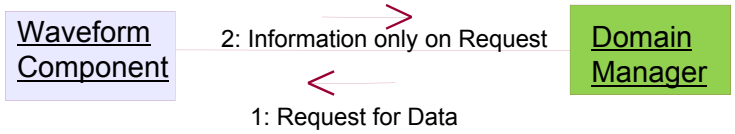
The SCA was originally developed with a 'pull' design pattern instead of a 'push' model.

SCA Next embraces the 'push' model, eliminating the possibility of clients requesting information they should not have.

The original SCA maintained lists that could be queried (or exploited) by other software components.
SCA Next does not publish these lists which could be inappropriately used.

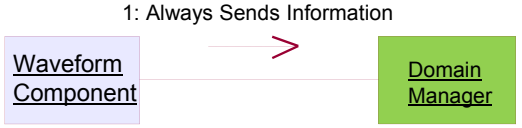
SCA Next deletes the Naming Service.

SCA 2.2.2



unauthorized objects can make information requests

SCA Next



Information is sent directly to the domain without having to authenticate the request.

DomainManager

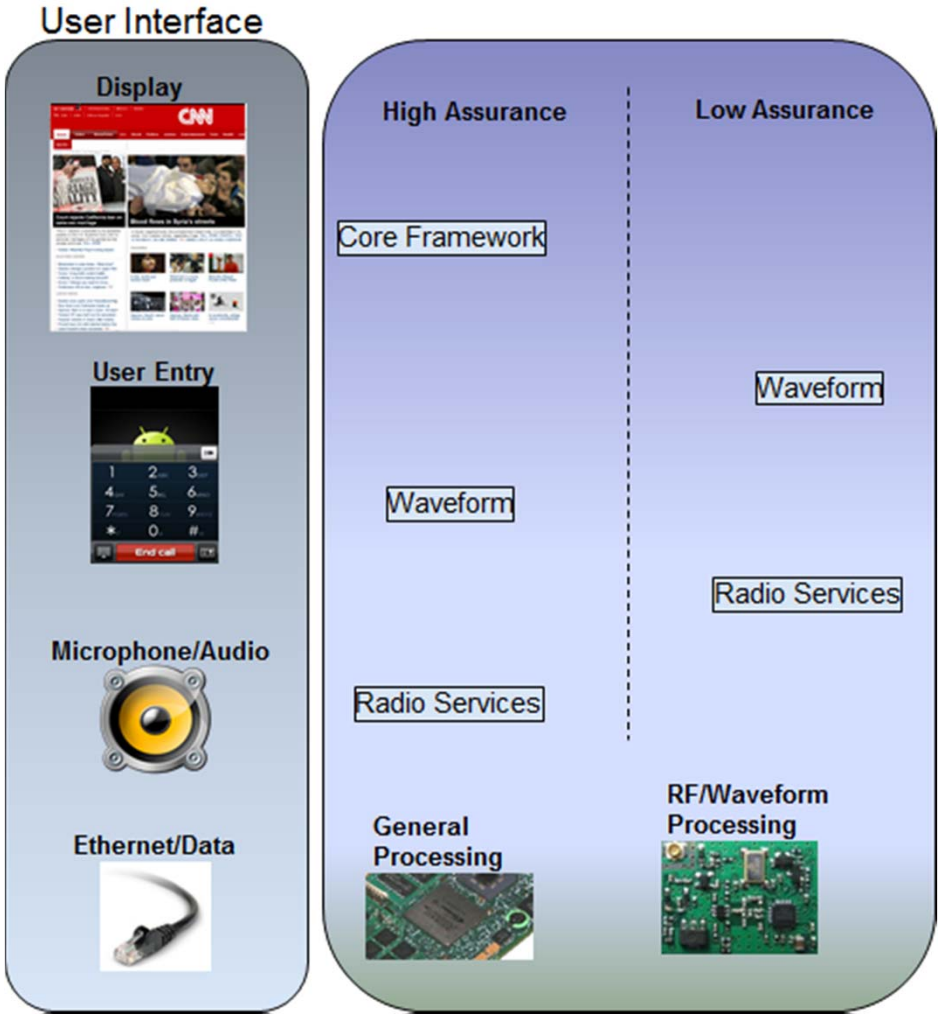
- + applicationFactories: ApplicationFactorySequence
- + applications: ApplicationSequence
- + deviceManagers: DeviceManagerSequence
- + domainManagerProfile: string
- + fileMgr: FileManager
- + identifier: string

- + installApplication(string) : void
- + registerDevice(Device, DeviceManager) : void
- + registerDeviceManager(DeviceManager) : void
- + registerService(Object, DeviceManager, string) : void
- + registerWithEventChannel(Object, string, string) : void
- + uninstallApplication(string) : void
- + unregisterDevice(Device) : void
- + unregisterDeviceManager(DeviceManager) : void
- + unregisterFromEventChannel(string, string) : void
- + unregisterService(Object, string) : void

SCA Next incorporates security awareness

Innovation to the Warfighter – CORBA Neutrality in SCA Next

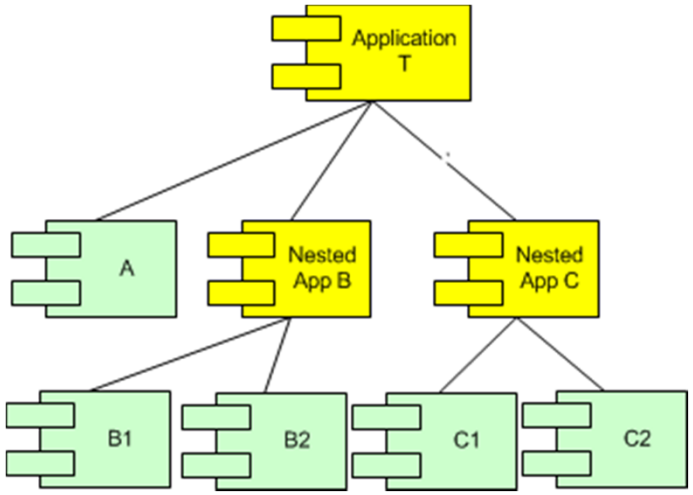
- JTRS has successfully used CORBA in tactical radios – resulting in products that can be fielded today
- There are other technologies that have been developed since the original SCA, such as the Android RPC.
- SCA Next permits other middleware, including simply using C++ pointers where distributed processing is not required.



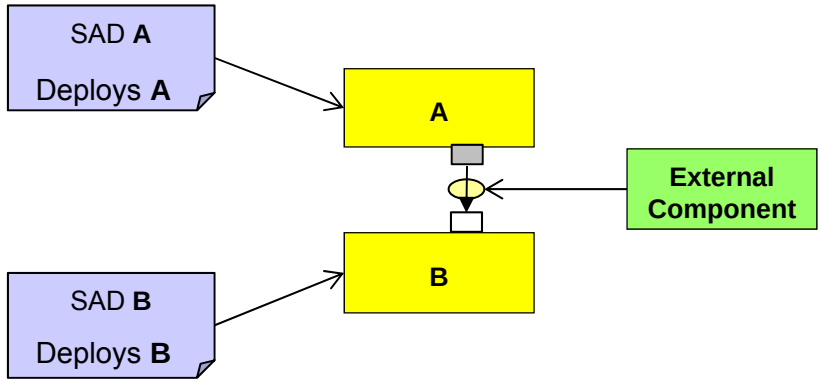
SCA Next can be combined with multiple technologies to allow a single-processor product – resulting in cost and battery advantages for the warfighter.

Innovation to the Warfighter – Application Connectivity in SCA Next

SCA Next Nested Applications



SCA Next Inter-Application Connectivity



a) Unbundles larger waveforms into smaller components
b) Potentially distributes deployment responsibilities

a) Promotes application level collaboration
b) Improved connectivity to an external application or service

Enhancements improve portability, and connectivity to external applications, such as an Android presentation layer

What's Next?

- SCA Next – Future Directions
  - Expand the SCA model to provide better **DSP** / **FPGA** / etc. support
  - Collaborate with providers to develop additional Platform Specific Models (e.g. **native C++ mapping**)
  - Collaborate with industry, academia or others to promote the development of **Open Source SCA Products**
  - Revise the **SCA User's Guide** to provide more information regarding use of the framework

Thanks to our Contributors

Questions?