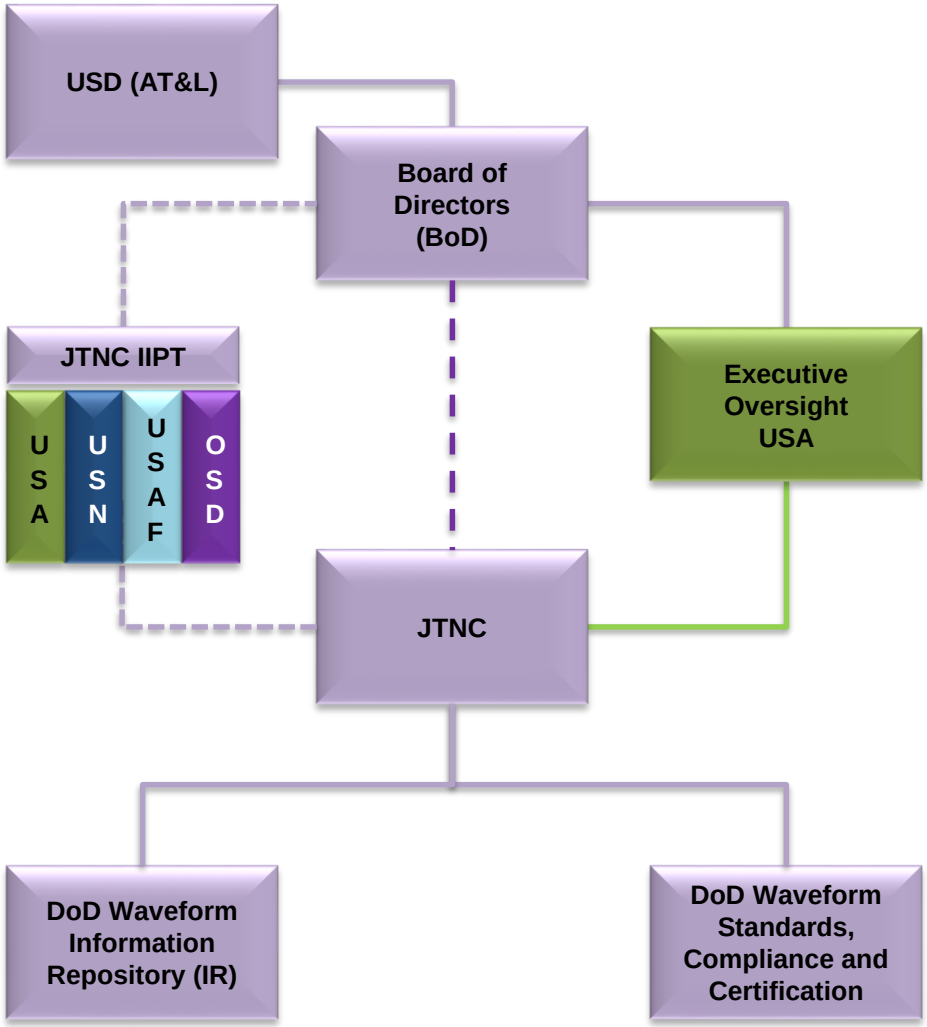
JTNC Standardization Overview

Department of Defense (DoD)
Waveform Standards
23 May 2018

Agenda

- JTNC Overview
- JTNC Activities
- Software Communications Architecture (SCA) Outreach and Advocacy

JTNC Overview



JTNC Chartered Mission

To ensure interoperable, secure, and affordable waveform and wireless communications by recommending standards, conducting compliance and certification analyses in accordance with DoD policies, and maintaining a DoD Waveform Information Repository (IR)

JTNC Chartered Vision

Interoperable, secure, and affordable waveforms and wireless communications in support of Service, Multi-Service and Coalition forces

Core Functions

- **DoD Waveform Standards and Software Communications Architecture (SCA)**
  - Provides a validated open systems reference architecture that separates waveform/network manager from the radio set
  - Permits common waveform software to be deployed across multiple vendor's radio sets
- **DoD Waveform IR Management & Configuration Control**
  - Provides a cyber-hardened, DoD-wide waveform library and controlled access for waveforms and associated network managers, operating environment software, models, architectural standards and Application Program Interfaces (APIs)
  - Protects and distributes artifacts based on legal agreements between government and software developers
- **Technical Analysis of DoD Waveform IR products**
  - Compliance: preliminary characterizations regarding meeting gov't standards for interoperability and security. Assessments facilitate preparation for participation in Service-level test events
  - Certification: comprehensive characterization of Waveform IR products as to whether they meet DoD standards and policies for interoperable and secure joint tactical networking
- **Technical Advisor to JTNC Board of Directors (BoD)**
  - Provide subject matter expertise on waveforms and wireless communications as requested or identified in support of DoD, the Services, Program Offices and stakeholders
  - Support the various DoD agencies overseeing the protection of critical technologies of wireless communications exported under commercial and/or Foreign Military Sales (FMS) and licenses

Collaborative Approach to Open Standards

- **JTNC coordinates with multiple organizations to evolve the SCA & Applications Program Interfaces (APIs) and align with industry standards in support of DoD and coalition efforts**
  - Joint Services
  - Other Government
  - Industry
  - Standards Organizations
  - Non-U.S.

- **SCA 4.1 developed within industry forums**
  - Meetings and website portal open to the general public for collaboration on the SCA

Coordination with industry associations aligns with JTNC's chartered core function to recommend adoption of the SCA and APIs

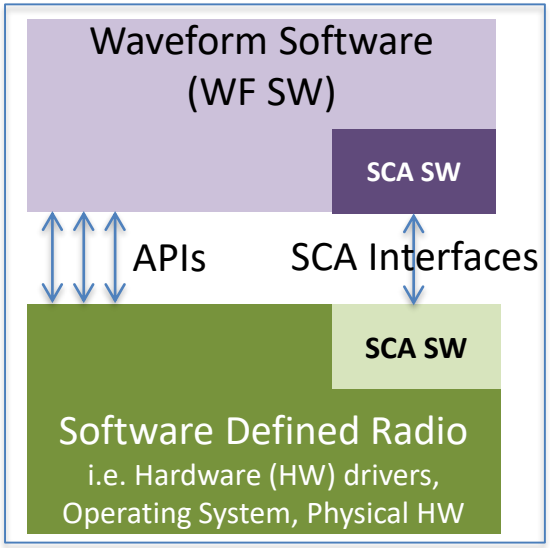
Selected JTNC Activities

- **Standards alignment**
  - Alignment enables DoD systems to conform to the SCA and other frameworks simultaneously - increasing software reuse and cost savings
    - Future Airborne Capability Environment OS profiles
- **Spec adoption**
  - SCA 4.1 Contractual Requirement in $142M Naval Air Systems Command (NAVAIR) Acquisition
- **Wireless Innovation Forum (WInnF) Verification and Validation**
  - DoD Waveform Standards and JTNC Test and Evaluation Lab (JTEL) actively participating in the development of the SCA 4.1 Verification Plan and Test Procedures
- **Technology Investigations**
  - Exploring approach for common Management Information Base
  - Researching usage of platform specific tools and patterns for the Ettus Research Universal Software Radio Peripheral family of products

SCA & APIs Enable Modular Design and Designate Key Interfaces

- **Software Communications Architecture (SCA)**
  - Defines a standard framework for instantiating, configuring, and managing software
- **Tactical Radio Application Programming Interfaces (APIs)**
  - Suite of interface specifications
  - Each defines a key software interface for an abstraction of the underlying product specific software functionality or physical hardware

Example of an SCA & API Enabled Product

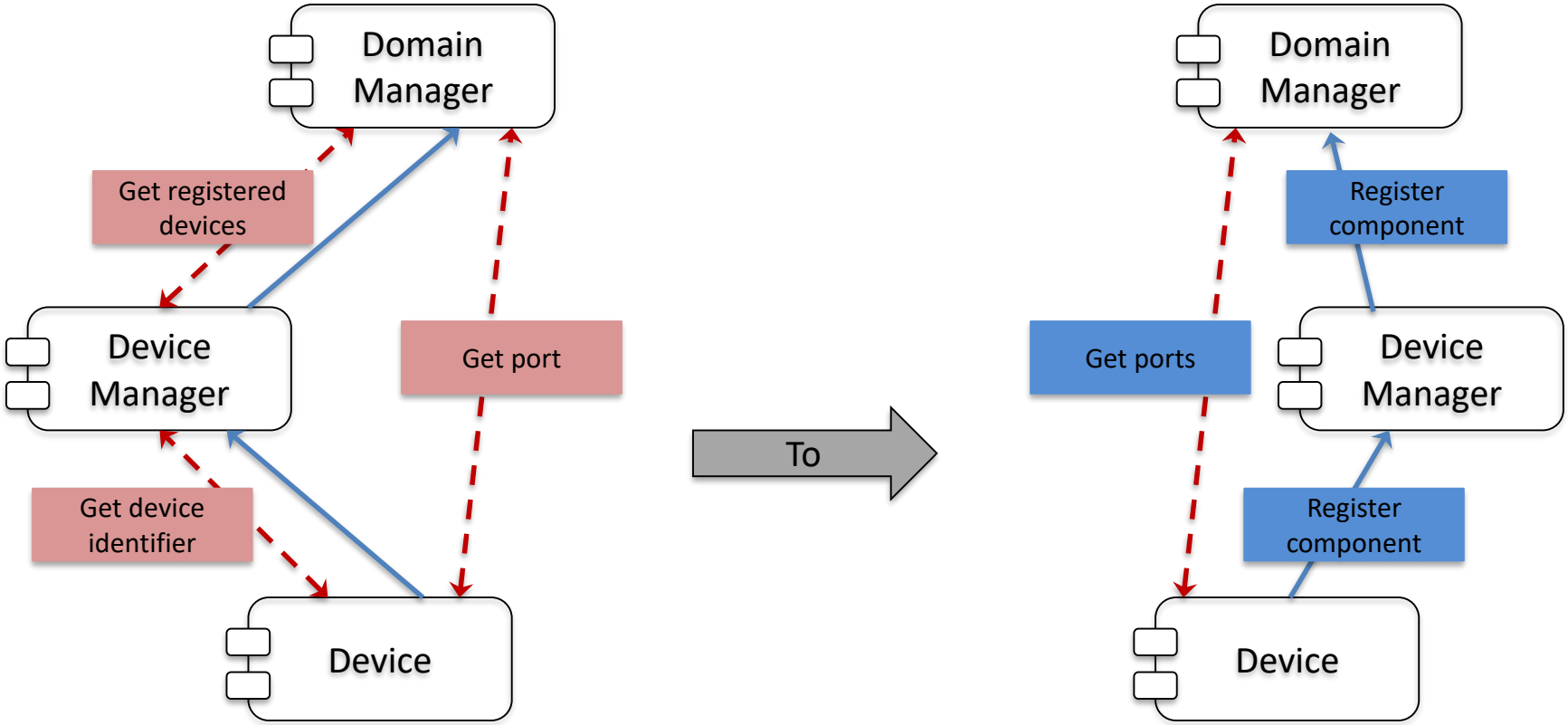


The SCA and APIs improve cyber security and performance, promote competition and reduce lifecycle costs

SCA 4.1 Benefits

- **System security enhancements**
  - Interfaces refactored to use least privilege pattern
  - Push model registration reduces interactions between components
  - Component Registry reduces potential information exfiltration
- **Platform start time reductions**
  - Push model streamlines component initialization and registration
  - Port “aggregation” minimizes calls
- **Platform performance optimizations**
  - Multi core processor support affords opportunities for better utilization of resources
  - ComponentFactory enables more component collocation

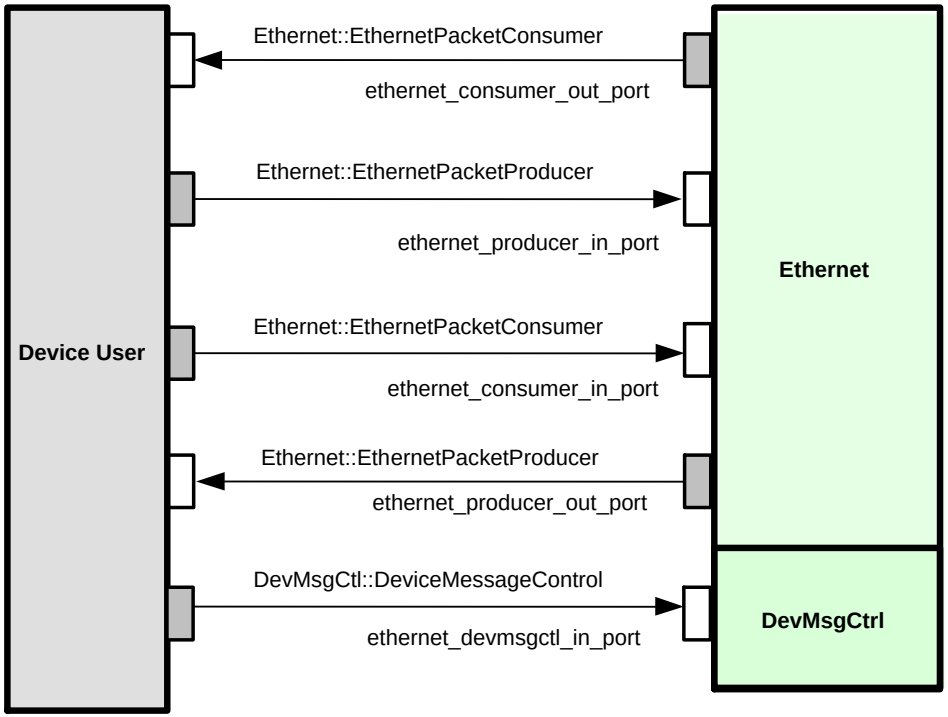
Enhanced Security Example



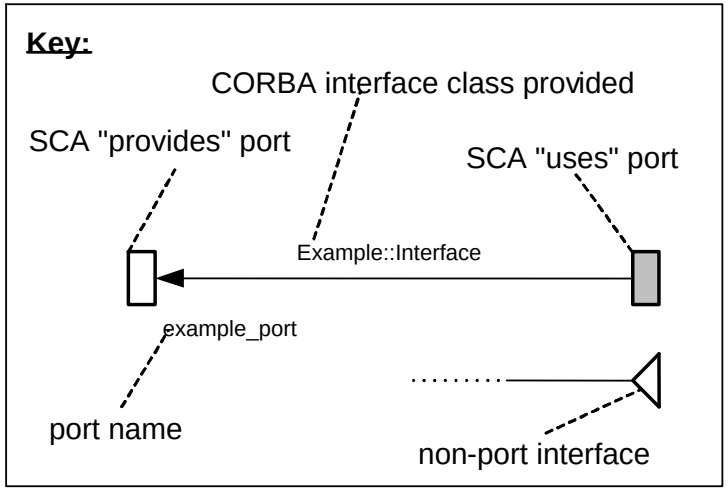
- **SCA 2.2.2 pull model**
  - Requires multiple calls to obtain attributes
  - Requires calls to get each port
  - Application component references are pulled from the Naming Service
- **Each call represents a risk (and requires resource utilization)**
  - Contents could be observed, intercepted or corrupted

- **SCA 4.1 push model**
  - Requires single call to register component
  - Provides alternative strategies for port connectivity
- **Reduces number of calls and consequently exposure**
  - Enhances performance also

Improved Start Time Example



Ethernet Device reference architecture



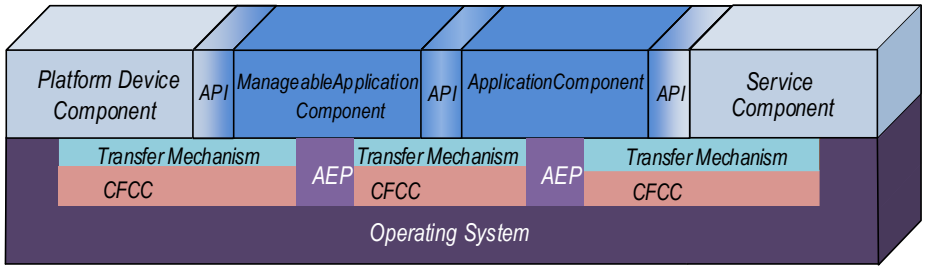
- **SCA 2.2.2 get port**
  - Requires 5 calls to retrieve the ports that establish the bidirectional interface
- **SCA 4.1 port aggregation**
  - Requires 2 calls (1 per direction) to establish connectivity

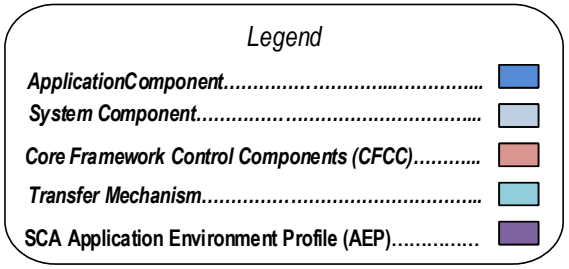
SCA 4.1 Technical Enhancements

- **Optional Composition**
  - Affords opportunity to align components with desired size and capabilities
- **IDL Decomposition**
  - Provides standardized method to exclude interfaces that are not applicable to a component
- **External Application Connectivity**
  - Enables improved system modeling, architecture and evolution
    - "Applications" can be isolated within a security enclave
    - "Applications" can be partitioned into commodity and mission critical sections (e.g. shared across providers)
- **Platform Independent Model**
  - Allows selection of a platform specific technology stack

SCA 4.1 Component Customization

- **Application and Platform Component Enhancements**
  - Component Scalability
    - Allows application, device and service components to tailor (i.e. exclude) capabilities that are not applicable
  - Lightweight AEP and CORBA profiles
    - Constrains application to a subset of the OE capabilities
  - Component Flexibility
    - Allows product architects to define product specific “standard” interfaces



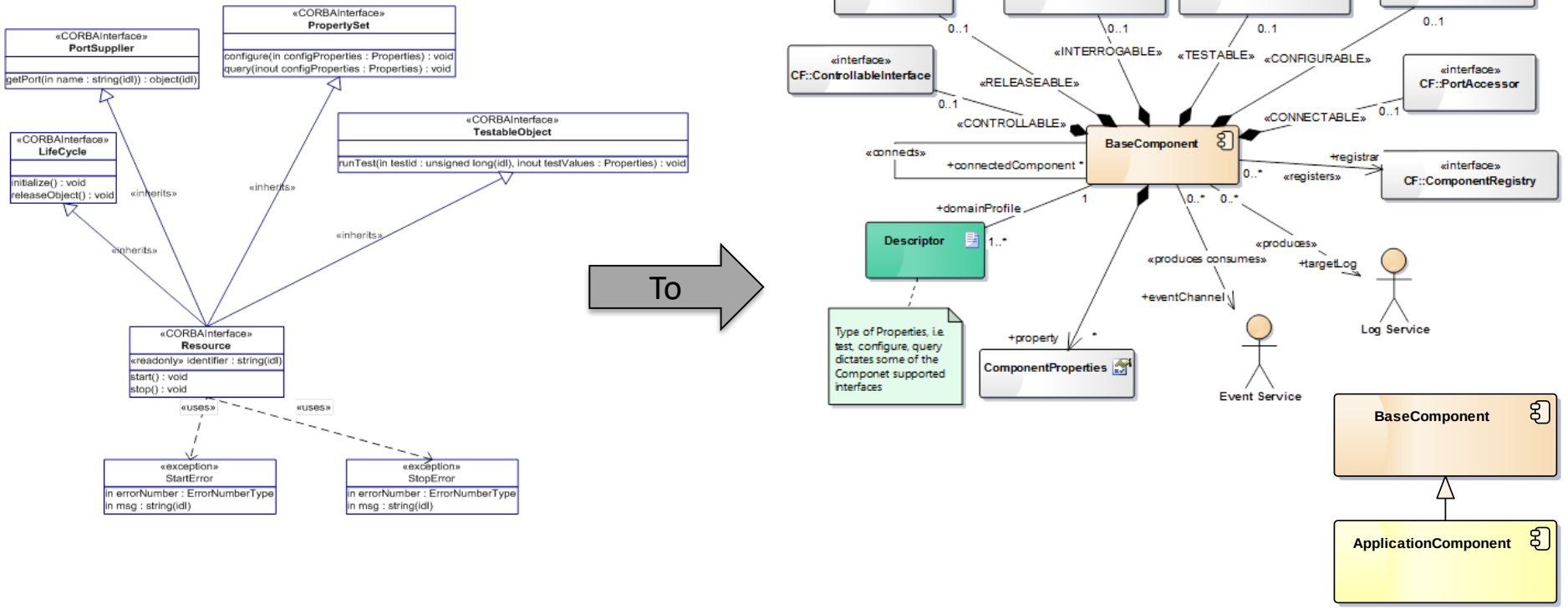


- **Core Framework Enhancements**
  - Platform independent transfer mechanism
    - Gives developer freedom to select mechanism that aligns to product architectural requirements
  - RTOS Profiles
    - Allows developer to provide an operating system that aligns with the needs of the target platform and applications
  - Manager Scalability
    - Allows manager component to tailor (i.e. exclude) capabilities not applicable to a product

SCA 4.1 Migration

- **Unique strategies will be adopted depending on the type of component and the nature of the platform**
- **There are three distinct families of SCA 2.2.2 developers**
  - **Application Developers**
    - Build Waveforms using the Base Application Interfaces
    - Migrate in accordance with the application migration guidance
  - **Device Developers**
    - Build Devices using the Base Device Interfaces
    - Migrate in accordance with the Device migration guidance
  - **Infrastructure Developers**
    - Build Waveforms using the Framework Control Interfaces
    - Migrate in accordance with the Application, ApplicationFactory, DeviceManager or DomainManager migration guidance depending on which components are supported

Application Migration Overview



- **SCA 4.1 restructured to communicate the specification in terms of interfaces and components**
  - The SCA Resource and Device interfaces were removed
    - Developer is responsible for defining interfaces which provide equivalent functionality
- **The following SCA 4.1 compliant interface could be defined which mimics the 2.2.2 Resource (there is a difference in the identifier)**
  - interface BaseResource : LifeCycle, TestableInterface, PropertySet, PortAccessor, ControllableInterface

Migration Considerations

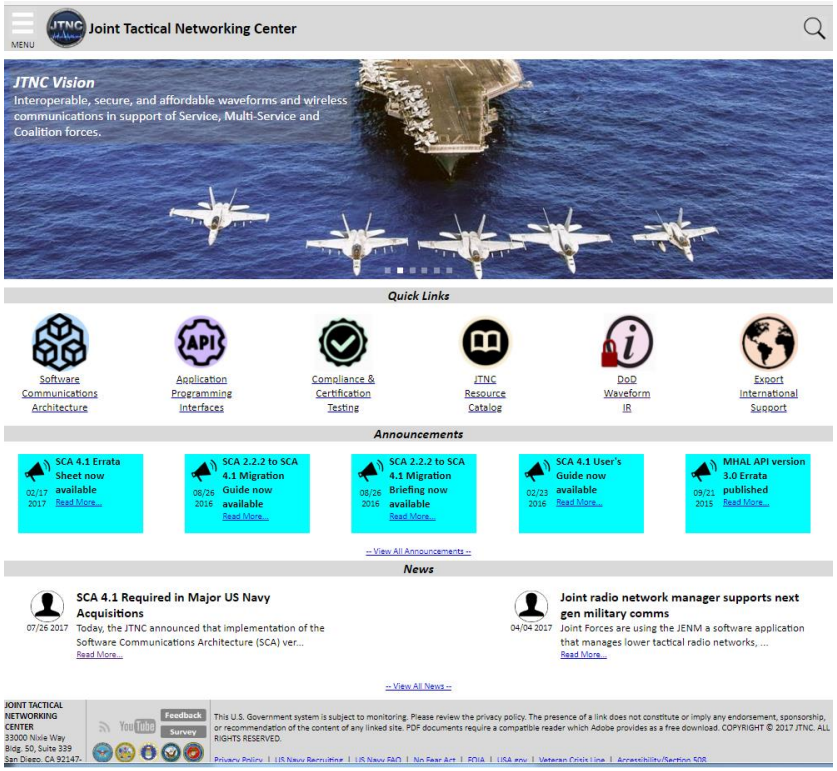
- **Applications – determine what needs to transition to SCA 4.1**
  - Nested applications allow mixture of 2.2.2 and 4.1 components
    - Consider future component evolution (e.g. do they need to become smaller, are they deployed in vulnerable locations)
- **Devices & Services – specification dictates that all devices migrate to SCA 4.1**
  - Could work with Core Framework provider to extend framework to have backwards compatible Platform Components
  - Ultimately, they need to be transitioned
- **Infrastructure (CF) – scope the breadth of the implementation project**
  - Determine need for backwards compatibility support
  - Identify the set of supported Units of Functionality

Migration Implementation Approach

- **Incremental approach allows for efficient use of resources**
  - **Implementations are shared across several component families**
    - Provides opportunity to implement once and disseminate across multiple components
    - ComponentType construct is fundamental to SCA 4.1 and requires a sound design
    - Component “interface” change requires a revised implementation approach
  - **Interfaces can be migrated prior to adding new functionality**
    - Proxies can provide mapping for existing capabilities, new functions can be left empty
  - **Test apparatus can move in parallel with component migrations**
    - Proxy components can translate between SCA 2.2.2 and 4.1 components to allow use of existing assets
  - **Units of functionality can be adjusted after the fact**
    - It may be more expedient to fully map and migrate an interface than to remove and regression test
    - Optimization (removal or addition) can occur in a future increment on top of a stable test platform

Additional Material

- **SCA & APIs**
  - **Standards & Specifications: http://www.public.navy.mil/jtnc_new/Pages/home.aspx**
- **SCA 4.1 Benefits and Features <*New*>**
  - **Paper: http://www.public.navy.mil/jtnc/PapersBriefsReports/SCA_4.1_Features___Benefits%20_v1a.pdf**
- **Detailed SCA 2.2.2 to SCA 4.1 migration brief and companion paper**
  - **Brief: http://www.public.navy.mil/jtnc/PapersBriefsReports/SCA_222_to_41_Migration.pdf**
  - **Paper: http://www.public.navy.mil/jtnc/PapersBriefsReports/SCA_222_to_SCA_41_Migration_Guidev0.1.pdf**
- **Webinar Recording: SCA 2.2.2 to SCA 4.1 Product Migration Guide Briefing and Discussion**
  - **https://youtu.be/5j4prVMiWZg**



Thank you

Contact Info:

DoD Waveform Standards
JTNC_Standadards@navy.mil
http://www.public.navy.mil/jtnc