Communications Research Centre Canada
An Agency of Industry Canada
Centre de recherches sur les communications Canada
Un organisme d'Industrie Canada

# The Importance of APIs for Certification of SCA Radios

**Steve Bernier, M.Sc.,**

**Senior Architect & Project Leader,**

**Advanced Radio Systems,**

**Communications Research Centre Canada**

# Outline

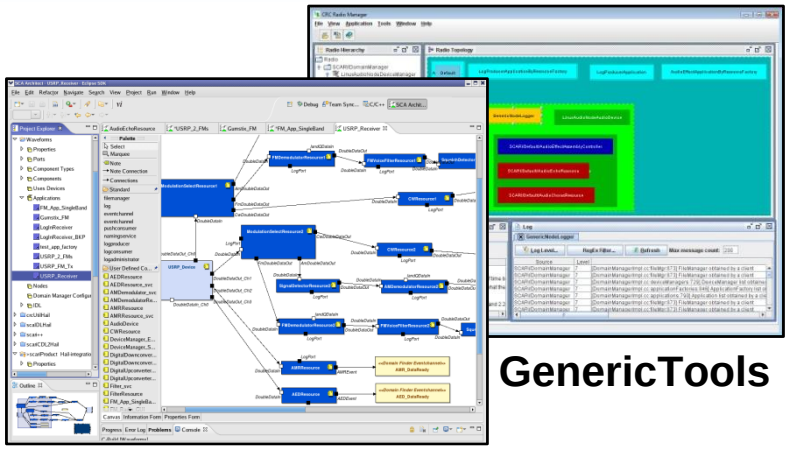
- The Software Communications Architecture (SCA)
- Compliance
  - SCA compliance
  - Domain-specific API compliance
- Conclusion

# The SCA

- **The Software Communications Architecture (SCA) is mainly used for the creation of military Software Defined Radios (SDRs)**
  - The SCA was created for the US DoD Joint Tactical Radio System (JTRS) program
- **However, the SCA standardizes generic features of software defined embedded systems**
  - The installation process for applications
  - The deployment of applications on heterogeneous distributed platforms
  - The control of applications
  - Introspection, health status monitoring

# The SCA

- **Standardizing APIs for common features enables the use of generic tools**



**GenericTools**

# The SCA

- **The SCA also helps make application source code more portable**
  - **Defines a standard for modeling software components and assemblies (Component-Based Development)**
    - Better documentation leads to better portability
  - **Imposes a standard for system calls used in applications (SCA POSIX AEP)**
    - Makes source code more portable across different operating systems
  - **Imposes a standard for communications between software components (CORBA & MHAL)**
    - Developers don't deal with transports directly

# The SCA

- **From a software development perspective, the SCA is a Component-Based Development architecture**

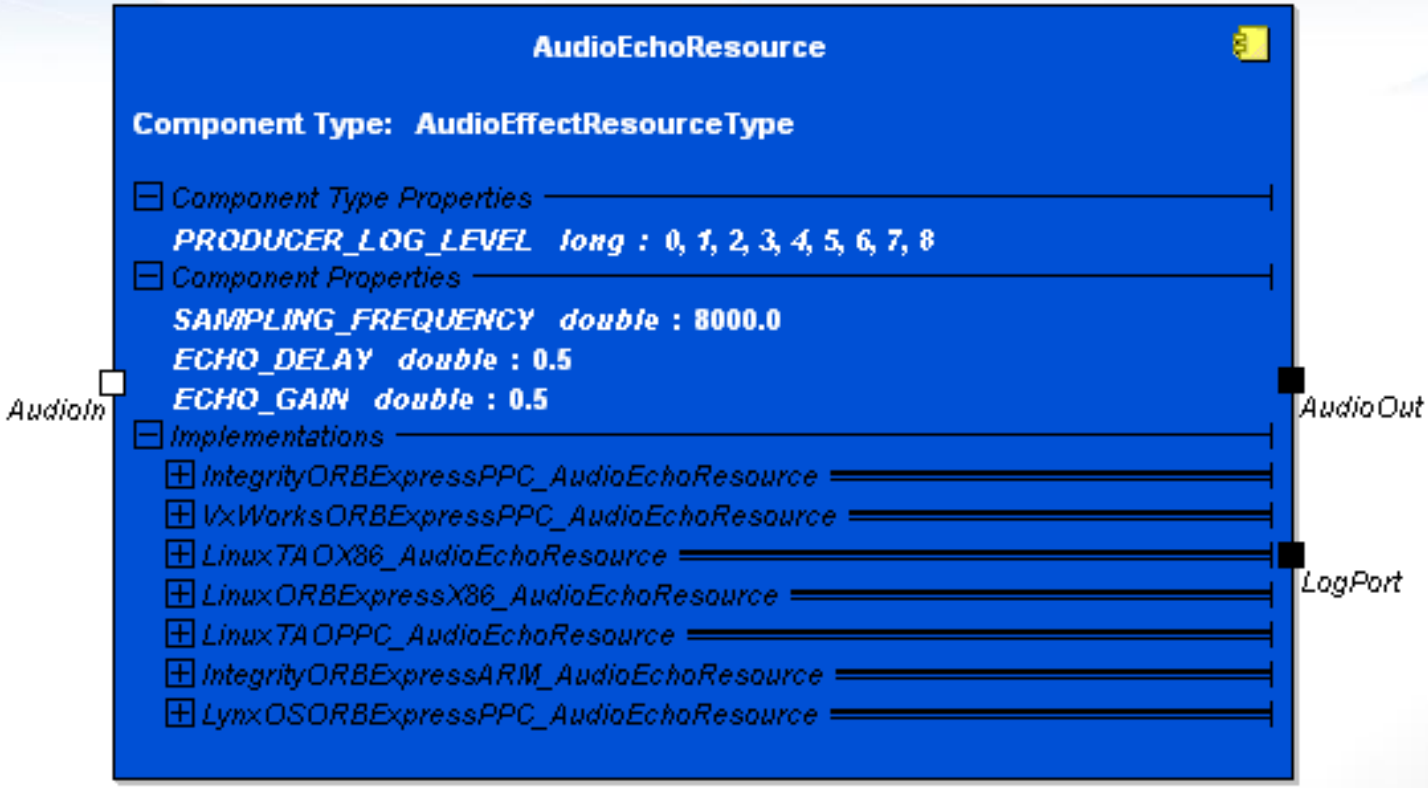
- **What is Component-Based Development ?**
  - **CBD is a development paradigm where the smallest unit of software is a component**
  - **With CBD, an application is 'assembled' using software components much like a board is populated with hardware components**

# The SCA

- **CBD promotes the COTS culture and is enabling the industrialization of software**
- **The goal is to apply the hardware development paradigm to software**
  - Purchase software components from a ‘spec-sheet’ catalog
    - Describe how to influence behavior (config properties)
    - Describe how to interface (ports)
    - Describe resource consumption (capacity properties)
    - Describe resource requirements (capability properties)

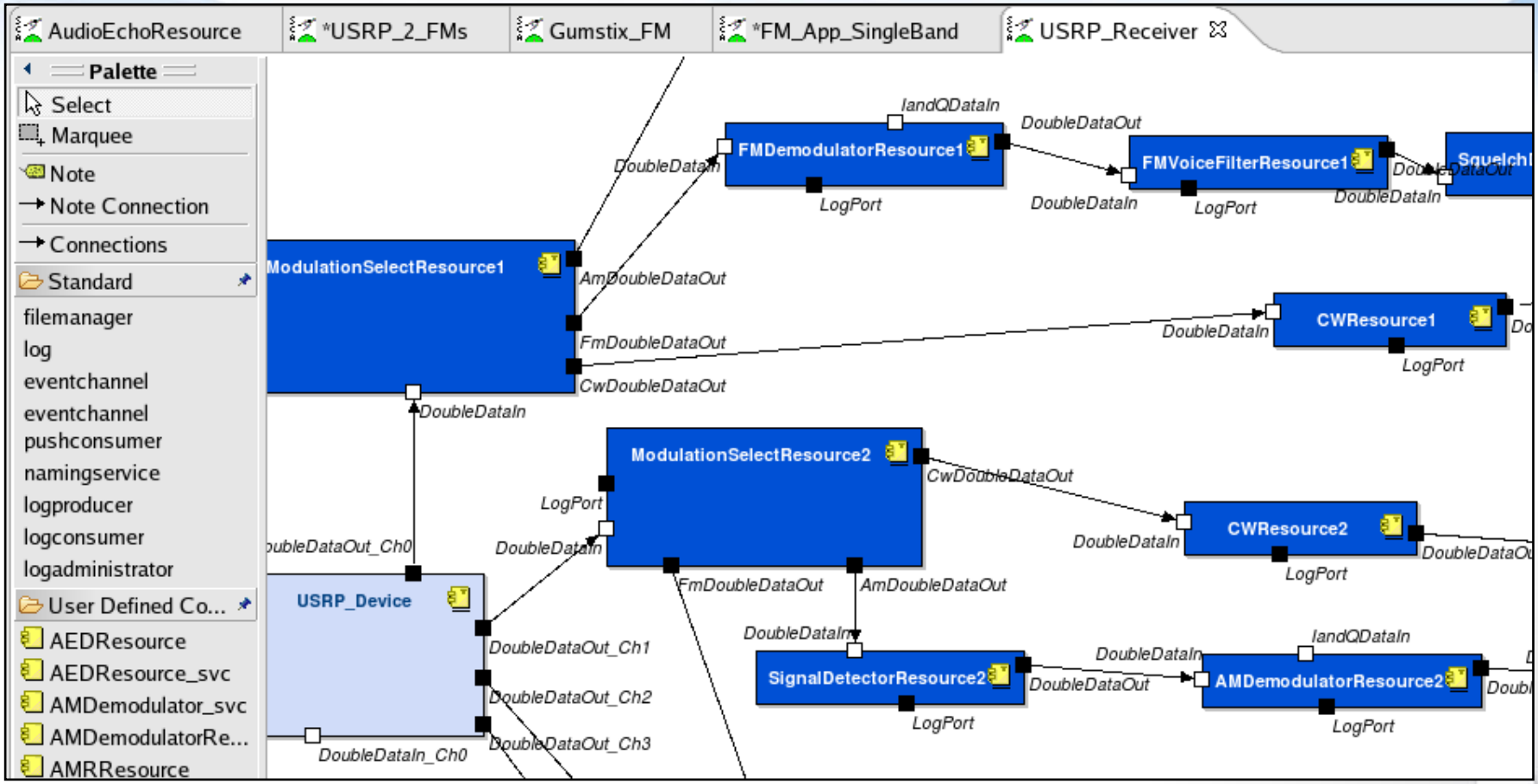
# The SCA

## Graphical representation for a software component model



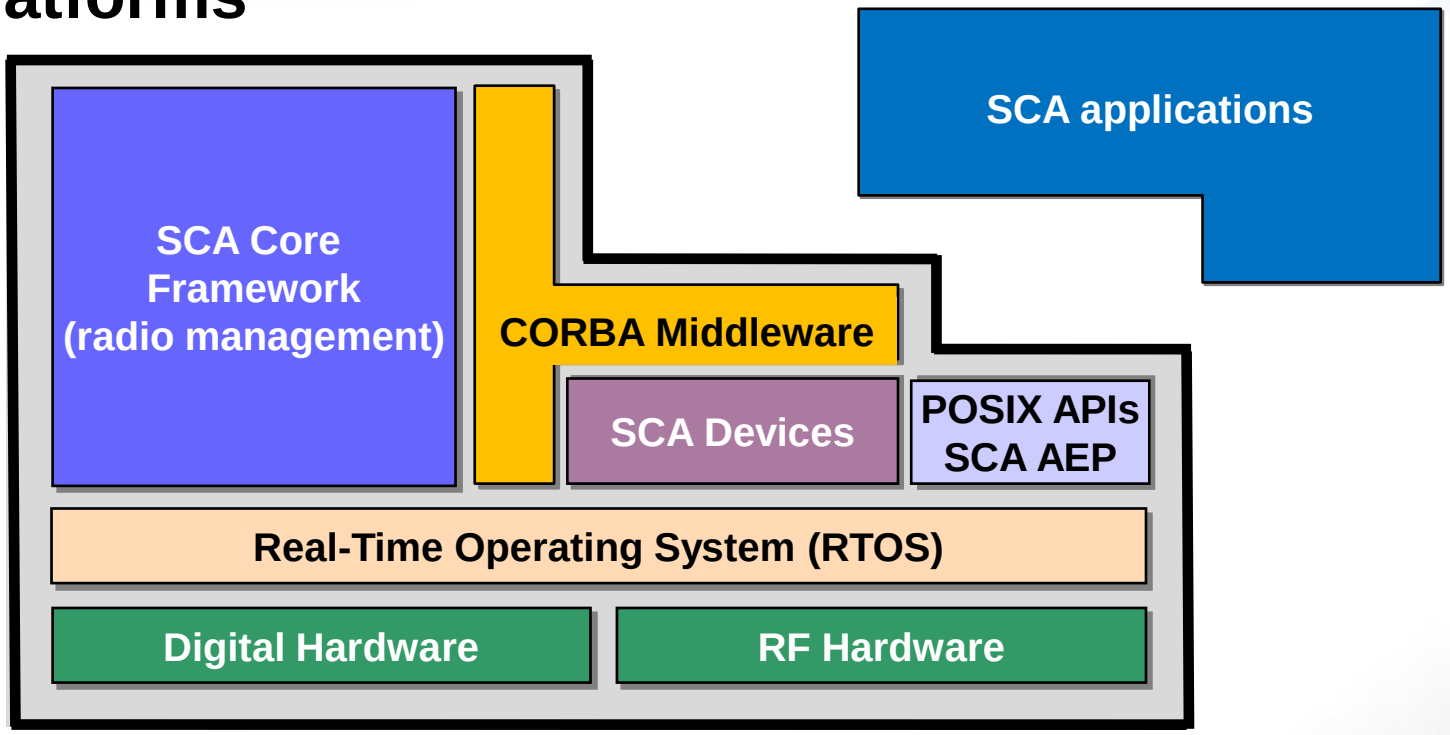
# The SCA

## Graphical representation for an assembly of software components



# The SCA

- The goal of the SCA is to allow applications to be quickly ported across different SCA compliant platforms




SCA platform

# Outline

- The Software Communications Architecture (SCA)
- Compliance
  - SCA compliance
  - Domain-specific API compliance
- Conclusion

# Compliance

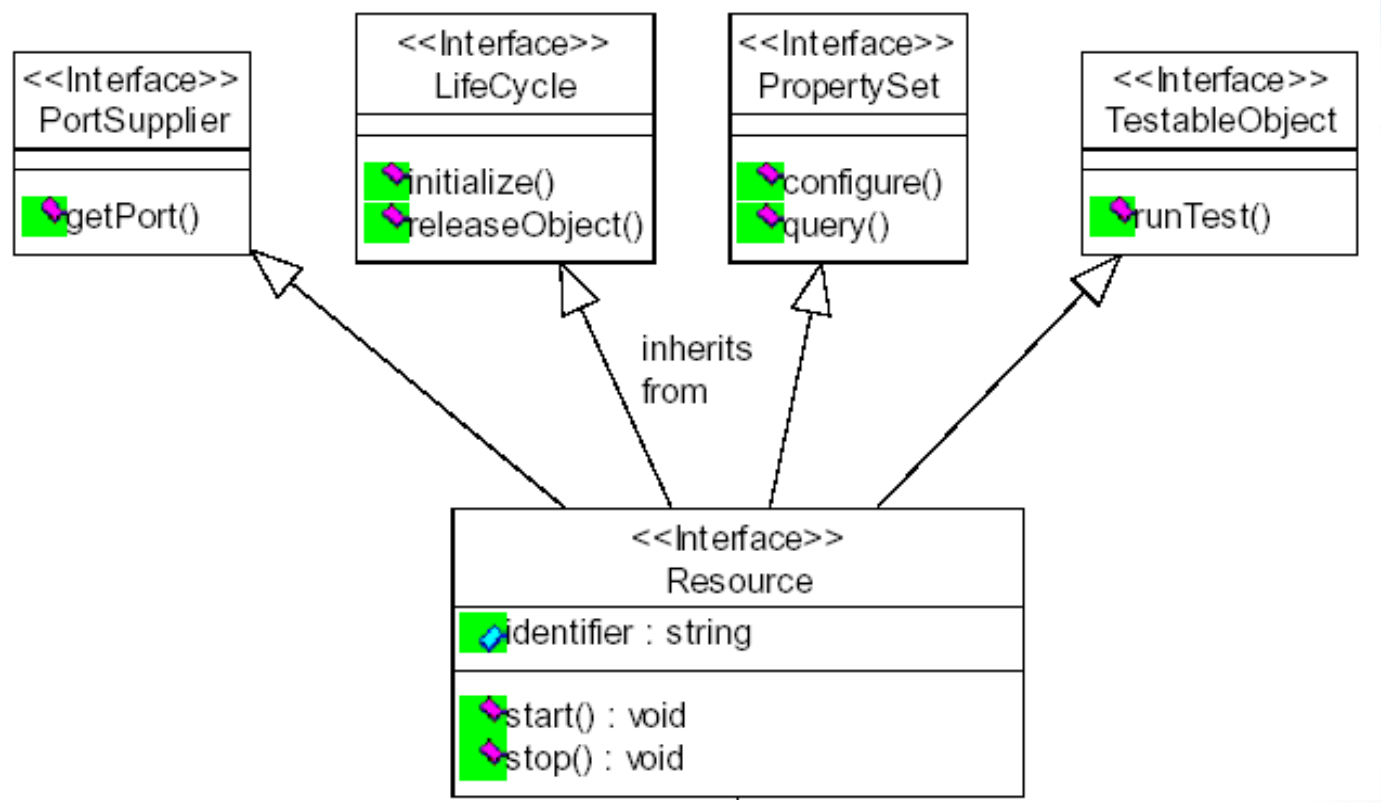
- **The certification of an SCA system can be viewed as a two-step process**
  - **First Step: SCA Compliance**
  - **Second Step: Domain-Specific API compliance**
- **SCA compliance only deals with the "Deployment and Configuration" aspect of software components**
- **Domain-specific API compliance deals with the APIs provided by the SCA Devices of a platform which are used by SCA Applications**

# SCA Compliance

- **The SCA specification document only defines APIs for Deployment and Configuration (D+C)**
  - The D+C is a process by which software is deployed onto processing elements (GPP, DSP, FPGA) of a platform
  - The D+C abstracts all types of processing elements using two type of SCA Devices: LoadableDevice and ExecutableDevice
  - The D+C standardizes how software components are initalized, released, started, stopped, interconnected, configured, queried

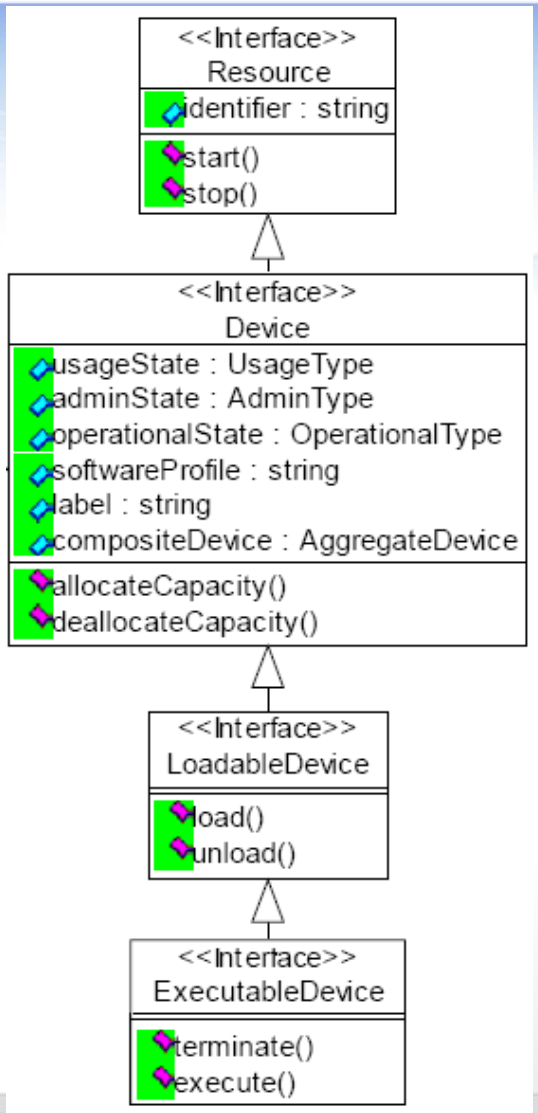
# SCA Compliance

- **SCA D+C Standard APIs for Application components:**



# SCA Compliance

- **SCA D+C Standard APIs for Platform Components:**



# SCA Compliance

- **SCA compliance for an *application* means it has to meet the D+C requirements:**
  - It comes with the proper description files (XML domain profile)
  - It only uses system calls allowed by the SCA POSIX AEP specification
  - It uses minimum CORBA and/or MHAL for communications between software components
  - It meets a number of SCA Core Framework requirements
    - Support standard input arguments, Provide standard properties, etc.

# SCA Compliance

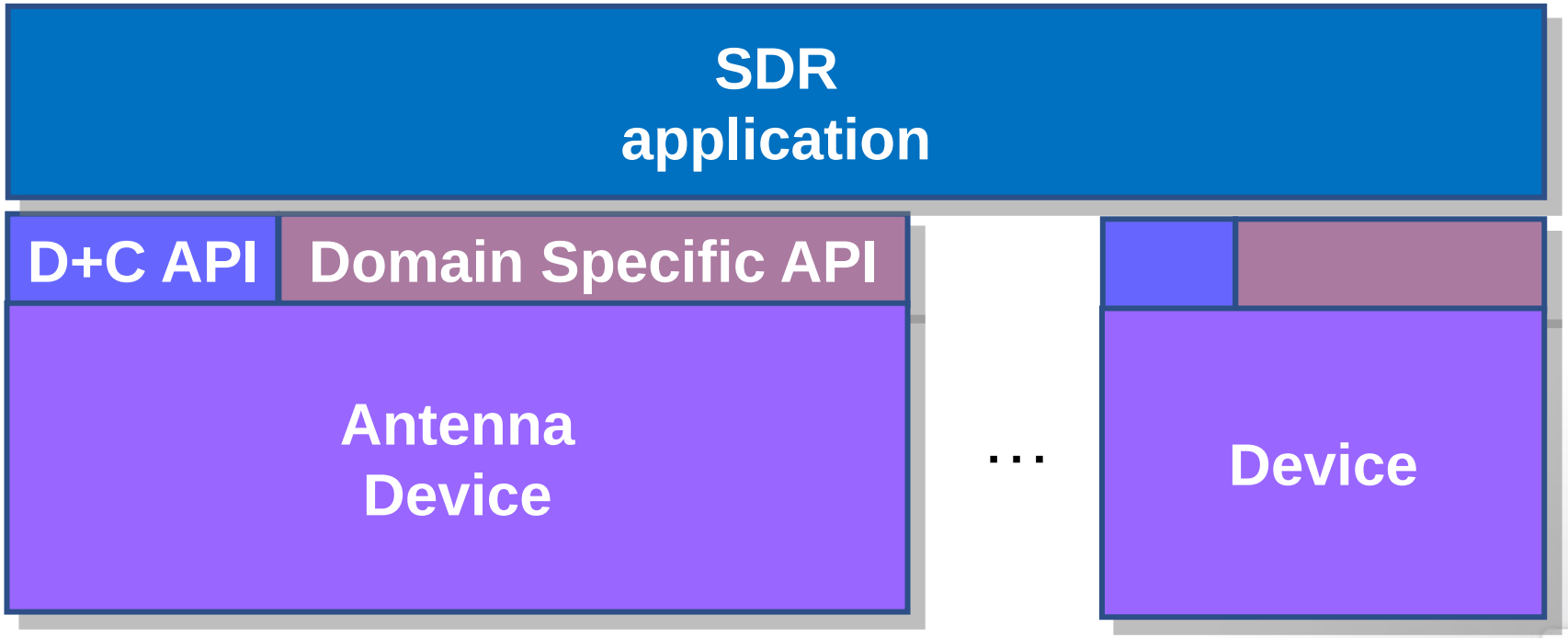
- **SCA compliance for a *platform* means it has to meet the D+C requirements:**
  - It meets the D+C requirements
  - It provides the system calls defined in the SCA AEP POSIX specification
  - It provides minimum CORBA and possibly MHAL support
  - It provides an SCA Core Framework
  - It provides at least one SCA ExecutableDevice which is used to deploy the software components of an application
  - Each SCA Device comes with the proper description files (XML domain profile)

# Outline

- The Software Communications Architecture (SCA)
- Compliance
  - SCA compliance
  - Domain-specific API compliance
- Conclusion

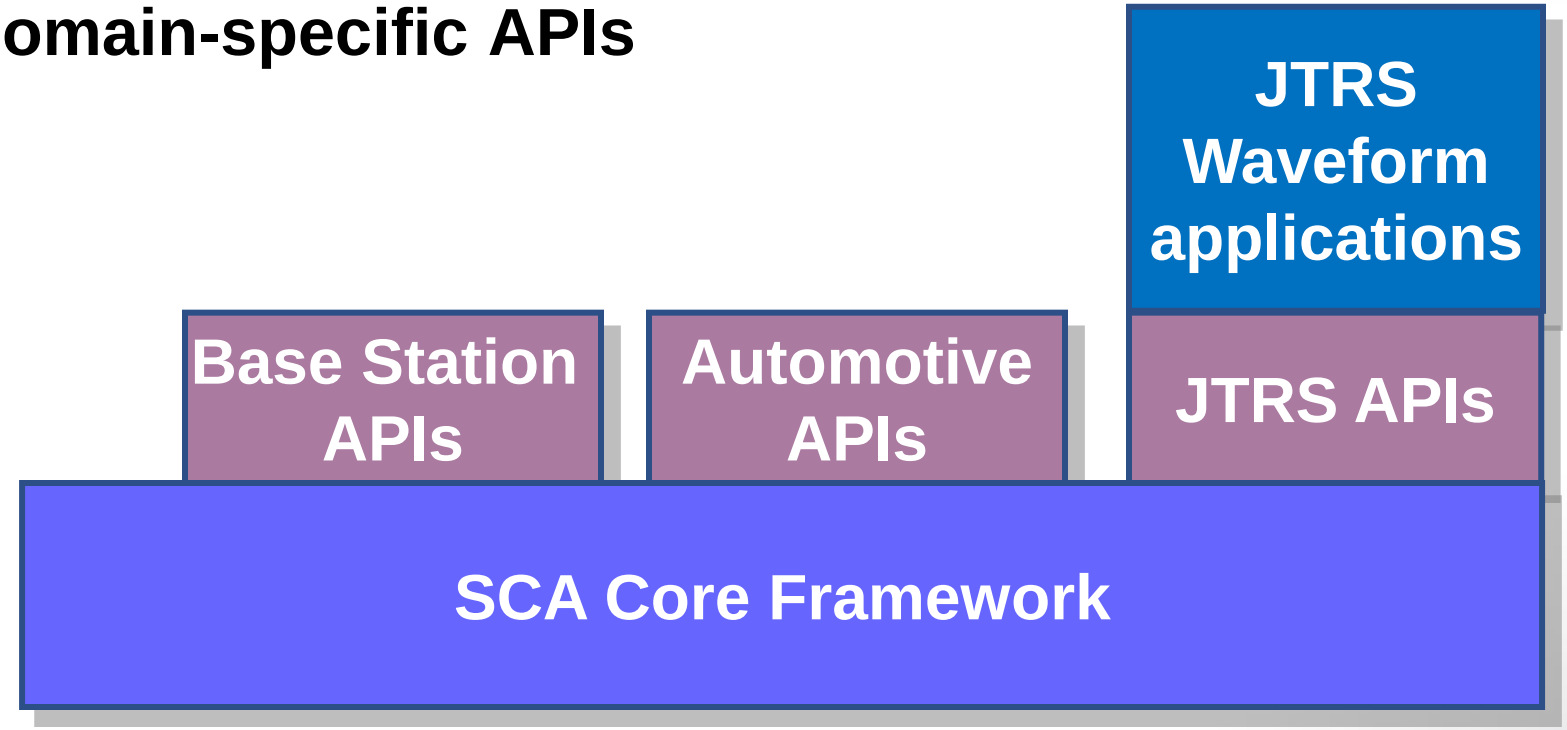
# Domain-Specific API Compliance

- **Domain-specific APIs are essential for application portability**
  - An application cannot easily be ported to platforms that don't provide the required domain-specific APIs



# Domain-Specific API Compliance

- The SCA is independent of the application domain
- Different domains are supported by domain-specific APIs



# Domain-Specific API Compliance

- **The Joint Program Executive Office (JPEO) has released a number of domain-specific APIs:**
  - **The "JPEO JTRS Standards APIs" fall under two category: basic and complex APIs**
  - **The complex APIs are made of basic APIs. Here is the list of the complex APIs:**

| | |
|---|---|
| AudioPortDeviceApi | MhalApi |
| EthernetDeviceApi | SerialPortDeviceApi |
| FrequencyReferenceDeviceApi | TimingServiceApi |
| GpsDeviceApi | VocoderServiceApi |

# Domain-Specific API Compliance

- **The OMG "Software-Based Communication (SBC)" Domain Task Force (DTF) has also released a number of models that can be used to define radio-specific APIs**
  - **Timing, Serial IO, Audio, Antenna, etc.**
- **The SDR Forum "Transceiver Subsytem Interfaces Task Group" is working in a new API**
  - **Transceiver API has been submitted**

# Domain-Specific API Compliance

- **The SDR Forum "Antenna API Task Group" has worked on an set of APIs for different types of Antennas**

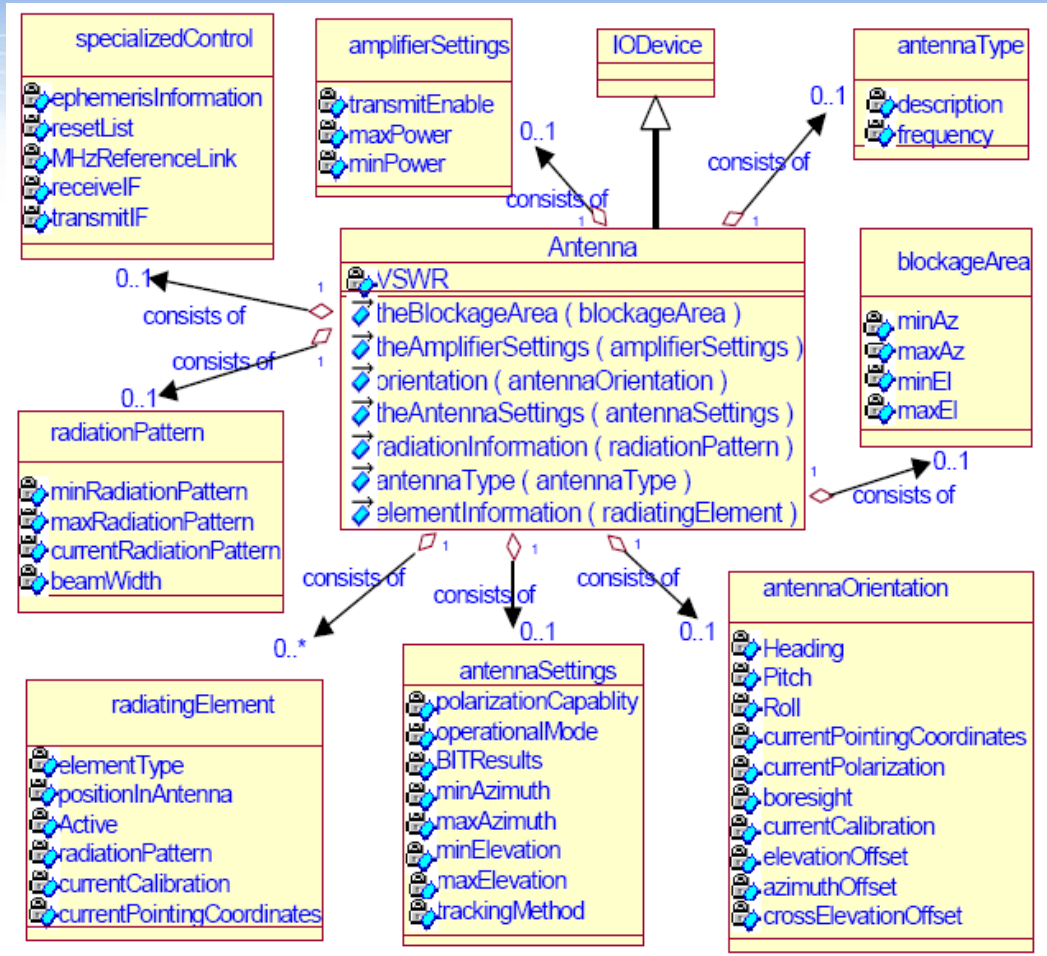



**Figure 14: Antenna Command and Control UML diagram**

# Outline

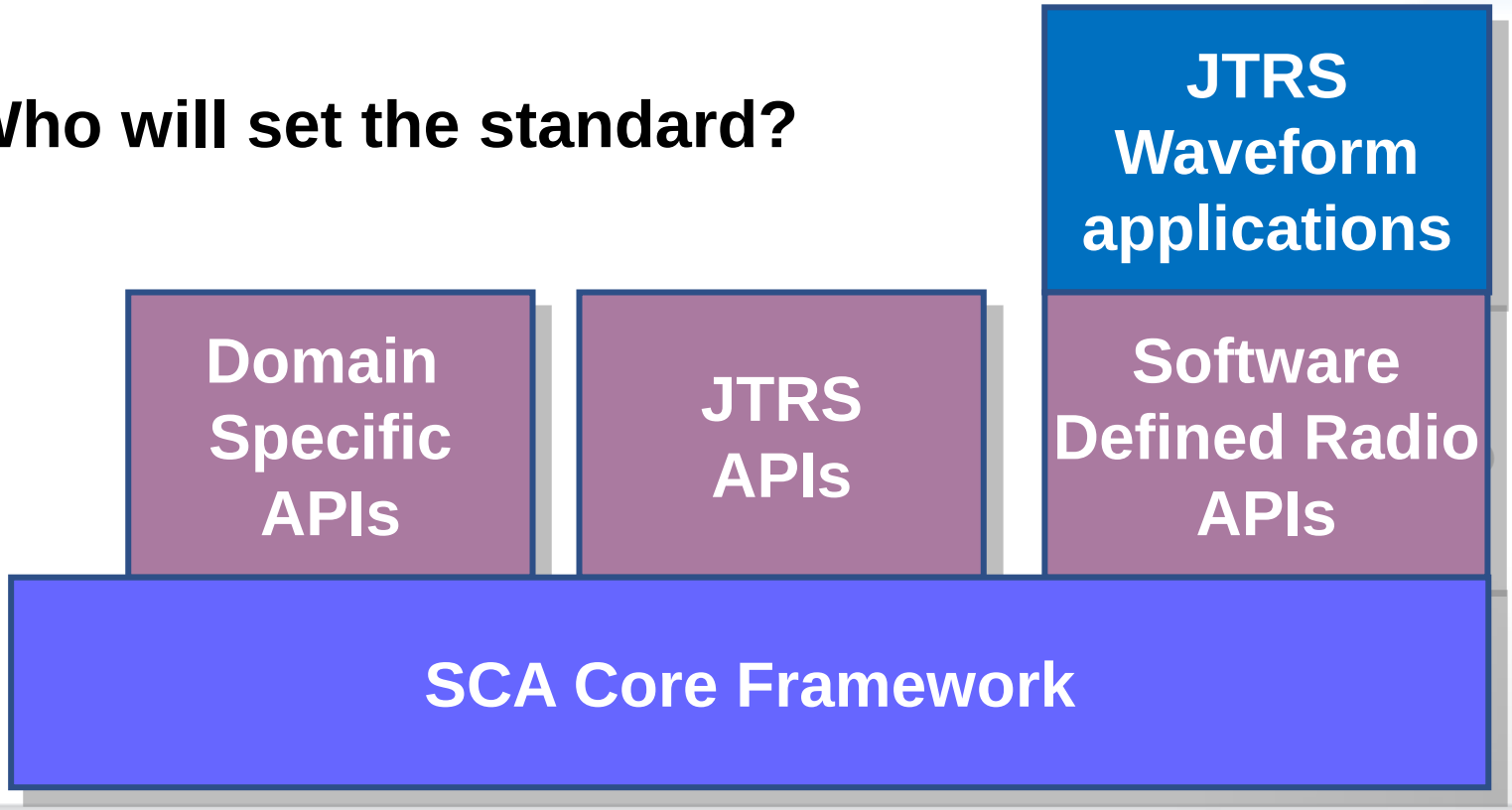
- **The Software Communications Architecture (SCA)**
- **Compliance**
  - SCA compliance
  - Domain-specific API compliance
- **Conclusion**

# Conclusion

- **Compliance is a two step process:**
  - SCA-compliance
  - Domain-specific API compliance
- **SCA compliance is independent of a specific application domain**
  - The JPEO relies on a number of tools (JTAP, WTT, DRP) and manual inspection for SCA-compliance
- **No single organization offers a comprehensive set of radio-specific APIs**
  - Immaturity and lack of APIs leads to the use of proprietary APIs which affects portability

# Conclusion

- **The next big step for Software Defined Radios is the standardization of radio-specific APIs**
  - **Will bring application portability to another level**
- **Who will set the standard?**



# Questions?

**Contact information:** **steve.bernier@crc.ca**
**www.crc.ca/sdr**