The Software Communications Architecture (SCA) was an early product of the Joint Tactical Radio System (JTRS) program, with SCA 0.1 released December 1, 1999.

It was a software-centric vision, with the idea that hundreds of reusable components could be dynamically linked together to form applications and waveforms.

- The SCA became non-domain specific – no radio-specific interfaces were defined
- The SCA documentation encouraged programs and industry to develop standardized application programming interfaces (APIs).
The Need for Tactical Radio Standards

The first waveform was ported to multiple Joint Tactical Radio (JTR) sets and porting effort was 6x expectations.

<table>
<thead>
<tr>
<th>Porting Deficiency</th>
<th>Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of JTR Set APIs</td>
<td>JTRS Standards was established for APIs and standards</td>
</tr>
<tr>
<td>Missing Files and Documents</td>
<td>JTRS Software Standard specified required documents and directory structure</td>
</tr>
<tr>
<td>Missing Functionality</td>
<td>JTRS Software Standard specified required software package content</td>
</tr>
<tr>
<td>Absence of DSP and FPGA Specifications</td>
<td>APIs developed for DSPs and FPGAs</td>
</tr>
<tr>
<td>Waveform Connection Complexity</td>
<td>Developer education and SCA changes</td>
</tr>
<tr>
<td>Security Compliance</td>
<td>Developer education and participation by NSA</td>
</tr>
</tbody>
</table>

- The government had waited expectantly six years for industry, Object Management Group (OMG), Software Defined Radio (SDR) Forum, etc., to bring a set of APIs and standards for SDRs, but it didn’t happen

- Consequently the government had to step in and create the standards
In 2005, radio-specific interfaces had not been defined
The original SCA concept was industry would define them
The Standards team requested guidance and participation from the DoD’s Modular Open Systems Architecture (MOSA) project, then named Open Systems Joint Task Force (OSTJF).

Business model considerations:

1) Data rights and licensing
2) Life cycle sustainment strategy (plug and play)
3) Modular designs with loose coupling
4) Lower development risk through system-level designs
We discovered that simply defining open systems APIs was insufficient.

We had to include design patterns in the APIs to enable:

a) aggregation,
b) least privilege
c) extension
d) explicit enumeration
e) deprecation
Example: Fine Grain Interface – FPGA Portability

The use case itself is very simple – how to achieve portability of Field Programmable Gate Array (FPGA) code.

- When developing the Verilog or other code for the FPGA, designers want to use manufacturer-optimized intellectual property instead of hand-writing a function such as a Fast Fourier Transform (FFT)
- The interfaces for these intellectual property blocks are not the same across manufacturers
- Consequently the FPGA source code must be modified when porting from one manufacturer to another
Elevating the Interface: Modem Hardware Abstraction Layer (MHAL) On Chip Bus (MOCB)

- The API specifies an interface to an MOCB block on the FPGA
- Within the MOCB block, designers can specify their unique interfaces
- The interfaces between the MOCB block and other MOCB blocks are defined
An essential capability of a tactical radio is security services.

Security services are frequently developed independently of the radio set and waveforms, requiring integration between three different groups.

- Radio Security Services is an API that isolates the waveform, radio set, and the security services.
- This API required subject matter experts (SME) and government agency coordination.
- The API decouples waveform porting from the radio-specific security implementations.
- Radio Security Services is the most complex tactical radio API.
Essential to open systems standards is a standards body.

- Product demand – compelling need for standards
- Membership composed of multiple organizations or programs
- Technical and business interests represented
- Funding for communications and configuration management
- Access to subject matter experts (SMEs) as required

**Notes from our standards body:**

- Meeting tempo began as twice monthly, then quarterly, then bi-annually
- Provided additional value with developer education as part of the meetings
- Face-to-face meetings superior to virtual meetings
- Non-attributional meeting notes
- Helpful to have end users represented
Contributors and stakeholders desire comprehensive documentation of processes (including corner cases).

This is time consuming and vexing – standards bodies are not all the same.

A few of the documented principles:

• Need – the API must solve a unique problem not currently addressed with existing APIs
• Validation – the API must be validated with an implementation to demonstrate maturity of the use cases for the API
• Collaboration – the standards body has collective experience and a knowledge base that improves the product
• Emergency Request - Contributors and stakeholders wanted a streamlined process that could accommodate an emergency request
Products and Success of the Standards Body

- 65+ Standards, APIs, and interfaces generated over a 6-year timespan

- Waveform porting substantially reduced
  - Today 50% of porting cost is testing
  - General Purpose Processor (GPP) software ports with little touch-labor
  - FPGA modem software ports acceptably within device families

- Soldier Radio Waveform (SRW) ported to 23+ radio platforms

- 39 Tactical Radio APIs and Standards registered in the DoD IT Standards Registry (DISR)

- JTNC Standards body sustaining and evolving the tactical radio APIs
Would recommend more cautious and judicious use of component-level software communication.

- The hoped-for collection of small reusable software components was never realized

Would consider larger-module components and waveforms

- Fine-granularity of software is not the commercial model

Would define hardware waveform interfaces.

- ASIC versions of WIFI and Bluetooth make software versions a difficult cost and schedule justification
- Imperative to leverage consumer products and technology
- Emerging tactical radio model today is a hybrid of software-defined and hardware implementation
Evolution to Radio Modular Architecture

The tactical radio architecture currently fielded is missing 3 emerging features:

- Support for commercial hardware modules/ASICS
- Support for commercial waveforms
- Support for software defined networking

The Modular Radio Architecture for Tactical Radios

- Not a complete reset, but no longer software defined radio (SDR) only
- Expanded perspective includes higher-level interfaces to reuse commercial technology and components
Observations from Generating the Tactical Radio Standards

- Generating standards is not a straight line – sometimes there are curves and detours
- Balancing between excellence and ‘good enough’ is challenging
- Strong leadership above the standards body is imperative
- There must be sufficient reward for contributors and participants