



Considerations for the Next Revision of STRS

Presented By: Joe Downey

Authors: Sandra K. Johnson,
Louis M. Handler, and Janette C. Briones
of NASA Glenn Research Center

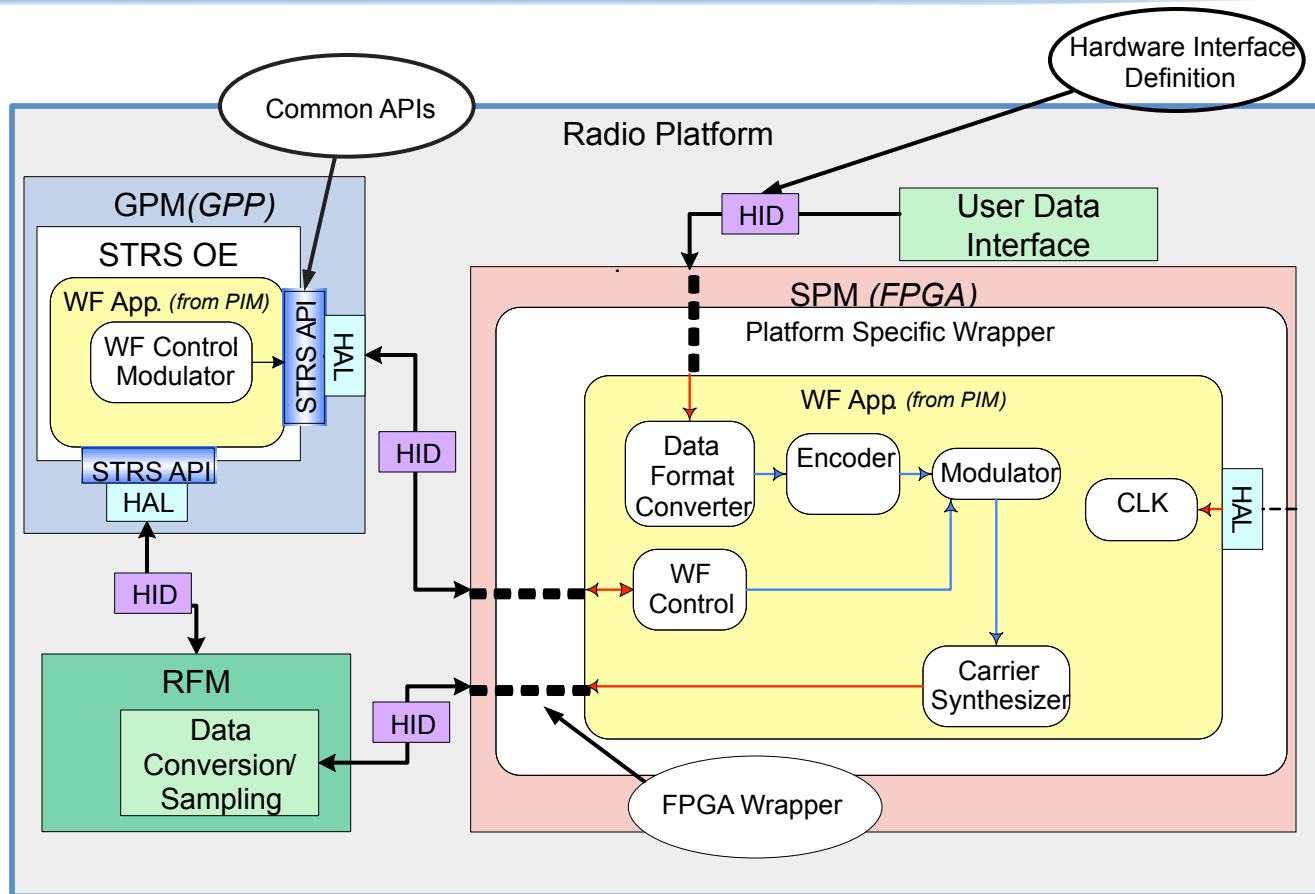


Agenda

- STRS Overview
- Tenets and Benefits of STRS
- Considered Updates
 - Clarification for STRS Devices
 - Handle Names and IDs
 - STRS Timing
 - Operating Environment Information
 - SDRs without GPMs
 - Cognitive Radio
 - Acquisition Guidance
 - Standardized Platform Services
- Future Work
- STRS Repository Current and Future Submittals



What is STRS?

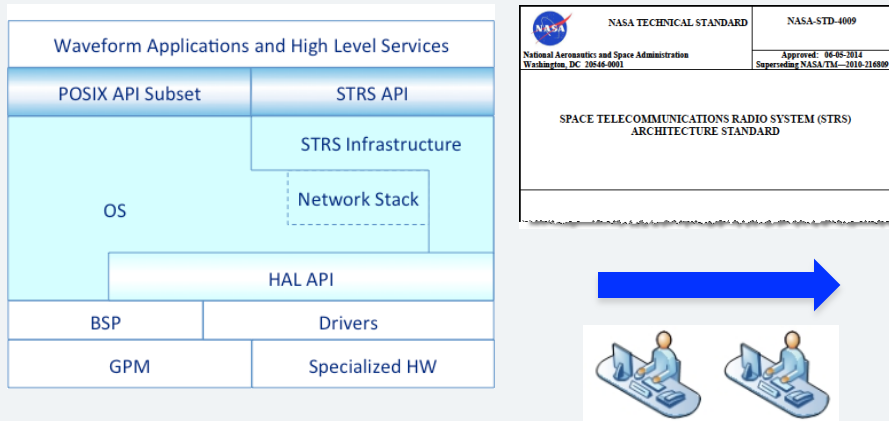


- Space Telecommunication Radio System
- STRS focus: constrained space environment
- STRS architecture: APIs and open HID and HAL



Tenets and Benefits of STRS

STRS architecture and Standard



Reduces effort to port applications, especially configuration and control portion. VHDL/Verilog porting effort may still be significant depending on platform differences. Development guidance (e.g. modularity, documentation) supports reuse.

Reduces vendor dependence allowing application development by third party on SDR platform.

Enables parallel development of platform and application, reducing development time.

STRS Repository

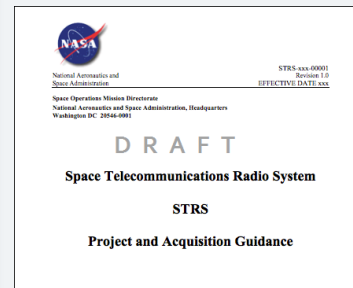


Pre-existing code is available from multiple potential sources.

Sharing of documentation across projects reduces effort and improves documents and procedures.

Knowledge Retention across projects improves quality.

Guidance for SDR Acquisitions

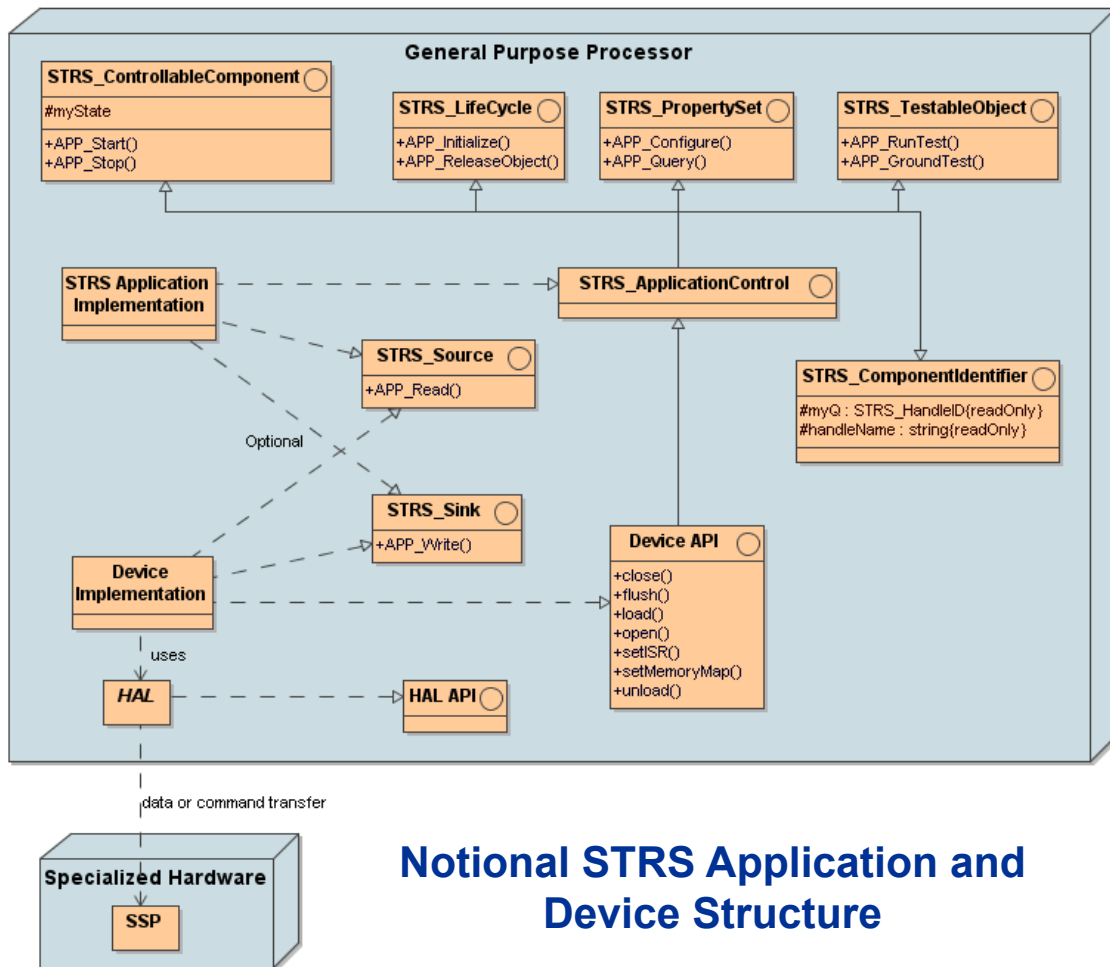


Guidance on unique SDR/STRS requirements, development, test approach, and data rights is provided.

Knowledge gained on the unique aspects of procuring SDR platforms and applications can be shared among projects.



Issue: Use of STRS Devices



Notional STRS Application and Device Structure

STRS Device Definition

- Extension of an STRS application with a set of unique (non-STRS standard) APIs
- Proxy for the data and/or control path to the actual hardware.

Discussion:

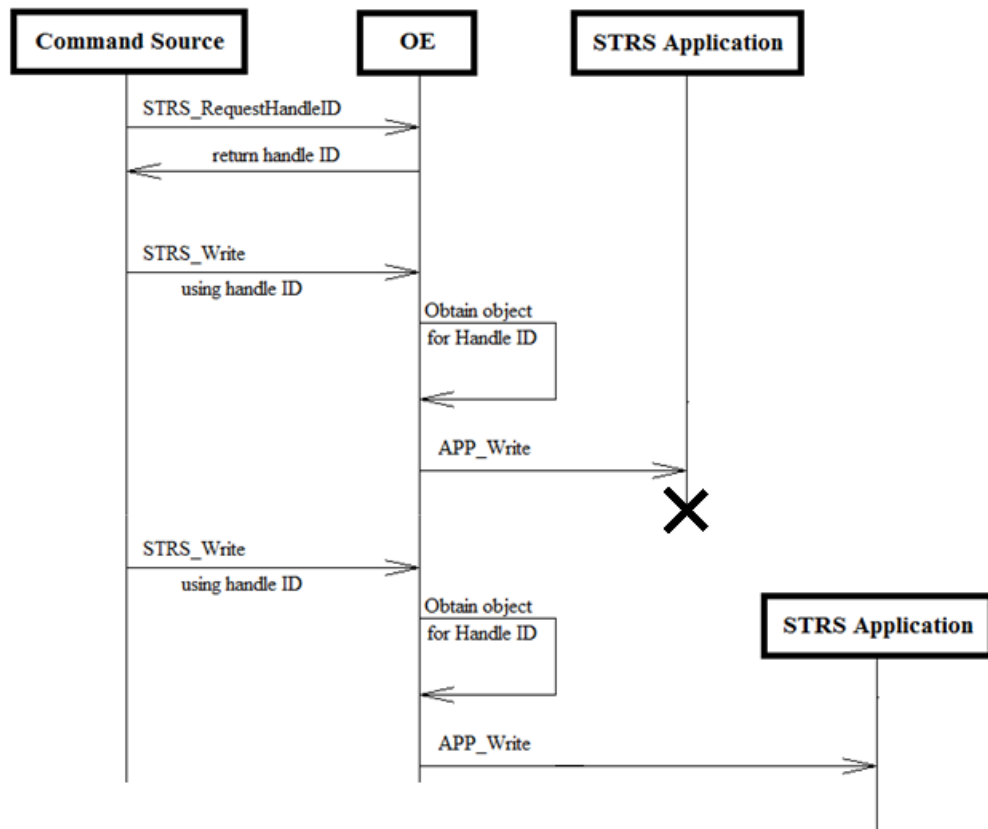
- Suggestion from commenters: Use standard device APIs to enable device portability across platforms.
- Device portability is not an STRS objective. It may limit platform option and add complexity that might affect performance.

Resolution:

- Add definition and detail in the supporting STRS Handbook, NASA-HDBK-4009, with suggestions that may aid device portability.
- No changes to the STRS architecture.



Issue: Uniqueness and Valid Timeframe of Handle Names and IDs



Handle Names and ID

- Defined with `STRS_InstantiateApp()`

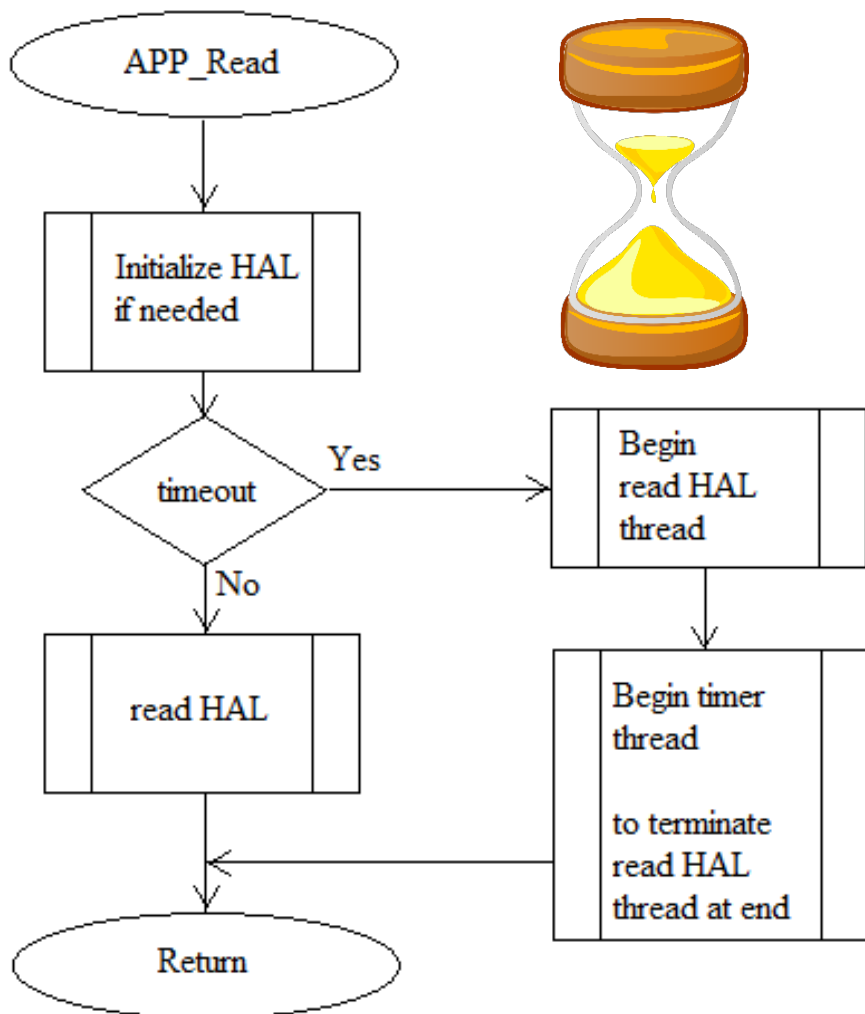
Discussion:

- Potential problem: reuse of handle IDs for different handle names for different purposes.

Resolution:

- Description field for `STRS_InstantiateApp()` updated to add: *Configuration file must specify handle names and it must be unique.*
- Section added to STRS Handbook to address suggestion Handle Name time concerns and determination of Handle ID.

Issue: Standardizing Timing Services using STRS Timing APIs



STRS time APIs

- Platform provider determines format and use of time contents needed for time stamps and event coordination.
- Must be defined by the mission.

Discussion:

- Questions arose concerning epoch, timestamps of messages, and use of dependent and independent timers.

Resolution:

- A new integer types will be added to allow for varying lengths for STRS_Seconds and STRS_Nanoseconds with guidance to prevent rollover for missions with a long operational lifetime.
- Recommendations added STRS Handbook about the use of the STRS time APIs and time conversion tool availability.



Issue: Ability of Application to Obtain Operating Environment Information

OE information required by some applications

- Application may need current information about the STRS Operating Environment such as version numbers, active applications, resources used and free, and faults.

Discussion:

- STRS_Query can be used, but without standard handle name and ID for OE, application not portable.

Resolution:

- New requirement to be added to require a standard handle name and handle ID available to obtain run-time details about the OE with STRS_Query.



Issue: Use of STRS in SDRs without General-Purpose Processing Modules

Small SDR platforms rely on external system for general purpose processing

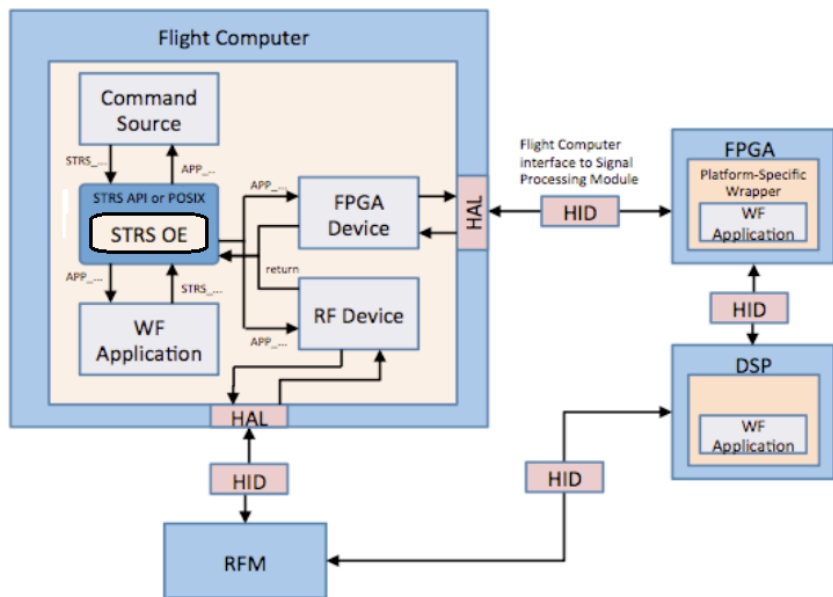
- STRS requires GPM for STRS API implementation.

Discussion:

- Compliant and non-compliant approaches to add STRS functionality suggested.

Resolution:

- This issue will continue to be scrutinized as digital signal processing hardware advances.



Option 1: Distribute functionality

- Compliant with current version of STRS
- Implement STRS OE on the flight computer.
- Adds burden to flight computer and multiple vendor integration.

Option 2: Control the SPM using non-STRS methods

- Increases application portability effort.
- Continues to offer advantages over non-STRS SDRs due to other STRS requirements (HAL, test applications, third party development).

Option 3: Control the SPM via the Application

- SPM respond independently by parsing bits from command stream.
- Same pros/cons as Option 2.



Issue: Implementation of STRS-compliant Cognitive Radio

Cognitive capabilities to be added to future SDRs at NASA

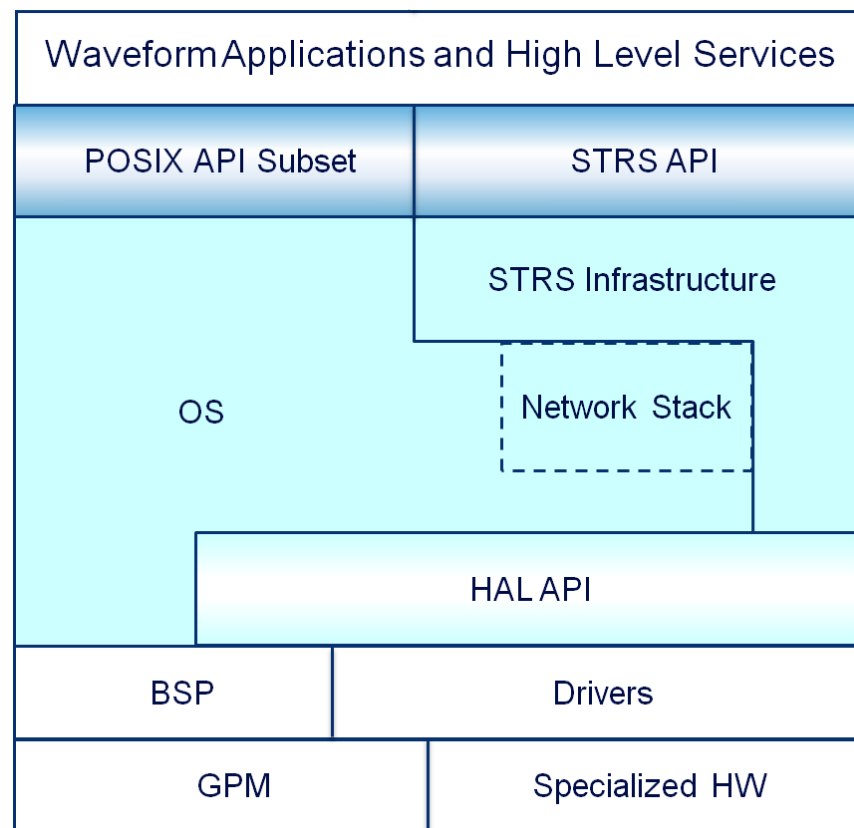
- SDRs ideal platform for implementing autonomous decision making ability.
- STRS has appropriate methods to implement cognition.

Discussion:

- Current STRS architecture adds an “adapter” layer between the application layers and the cognitive engine.

Resolution:

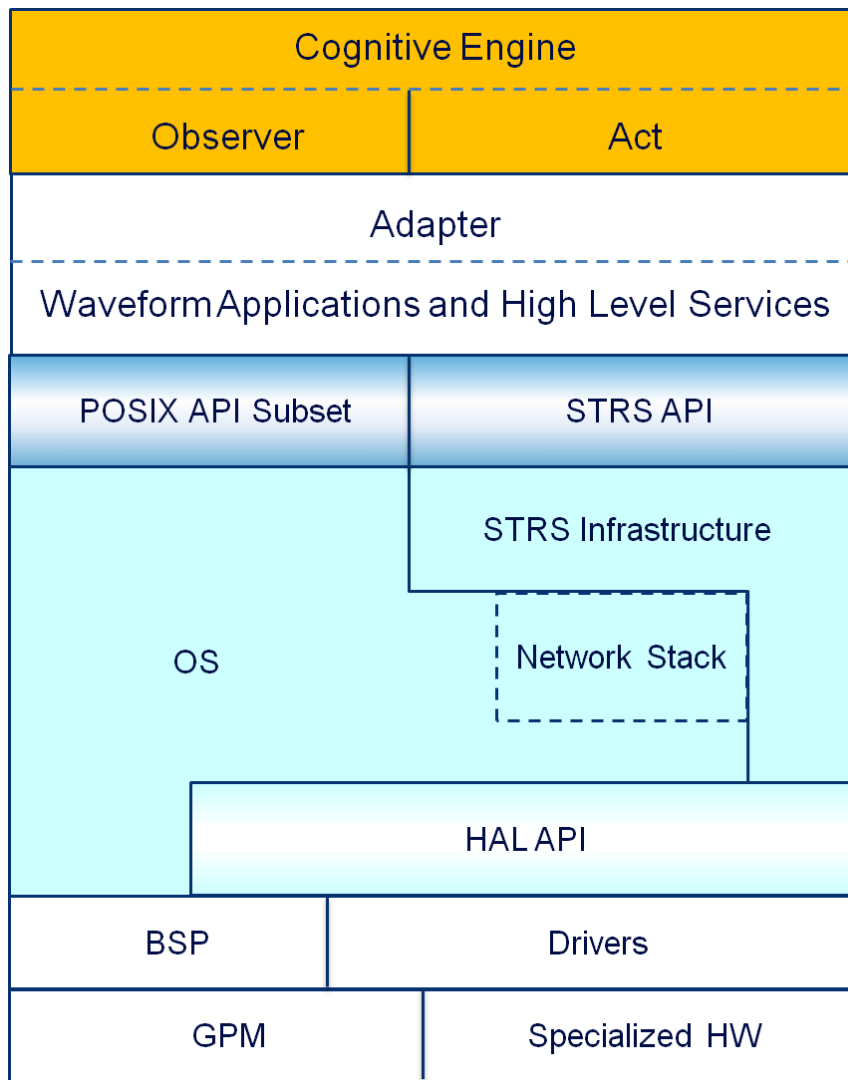
- Concept is currently being studied in a prototype system.





Cognitive Radio

- The “adapter” layer in the STRS architecture informs the cognitive engine about the radio environment.
- The cognitive engine can learn autonomously from experience and take appropriate actions to adapt the radio operating characteristics.
- Optimize performance under adverse conditions such as
 - mitigating the effects of unplanned interference,
 - maximize the data throughput,
 - reconfiguring due to propagation effects





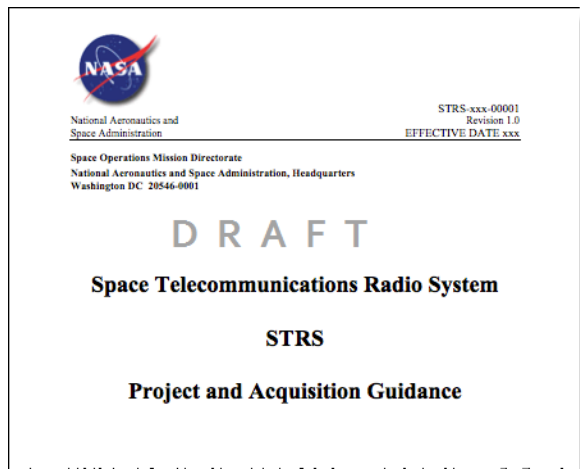
Issue: Capturing Acquisition Guidance

Mission-specific Guidance

- Some requirements are based on mission needs - not part of the STRS Architecture Standard.
- Solution: Write the "STRS Project and Acquisition Guidance" document.

Examples of Content Scope

- Data rights (OE, wrapper, application, documentation, source code).
- Required documentation for SDR procurements.
- STRS-specific needs for projects (training, compliance, development systems).
- Flexible command and telemetry structure.
- Verification and Validation considerations.
- Test waveforms.





Future Work

- Complete updates to identified documents and obtain concurrence. Expected to be completed in FY17.
- Any suggested software changes will be coded and the NASA GRC's reference implementation will be changed accordingly to assure accuracy.
- Continue to capture Lessons Learned for future updates.
- Infusion to promote STRS for use in future flight hardware missions.



STRS Website
<https://strs.grc.nasa.gov>



STRS Repository Current and Future Submittals

	Developer	Waveform/Component Description	Data Rights	Submittal Year
Complete or near complete	GRC	✓ STRS Compliance Tools	GPR	FY14
	GRC	✓ GRC GSFC TDRSS (GGT) Waveform for JPL SDR	GPR	FY14, FY15, FY16
	GRC	Harris SDR Capture Test Waveform	GPR	FY15
	JPL	✓ Short Blocklength LDPC Codes for SDRs	GPR	FY15
	JPL	✓ General Purpose LDPC Codes for SDRs	GPR	FY15
	GRC	✓ Reconfigurable Bandwidth-Efficient Transmit Waveform for High-rate Telemetry	GPR	FY15/FY16
	JPL	✓ OE & Test Waveform Applications	GPR	FY15/FY16
	GRC	✓ STRS Reference Implementation	GPR	FY16
	GD	Baseline S-band TDRSS waveform software	Limited	FY14
	Harris	Baseline Ka-band TDRSS waveform software	Limited	FY14
In development	GRC	SNR estimator component	GPR	FY16
	GRC	ACM component	GPR	FY16
	JPL	~15 components and 10 waveforms	GPR	FY15-FY18
	Comm Largo*	Scintillation-Hardened GPS waveform	Open	FY16
	GRC	VCM component	GPR	FY16
	GRC	DVB-S2 waveform	GPR	FY16-FY18
	Virginia Tech	Signal Classification and Interference Mitigation component	GPR	FY16/17
	Worcester Polytechnic	Intelligent Media Access Protocol component	GPR	FY16/17
	GRC	Cognitive Engine on an SDR	GPR	FY17

GPR = Government Purpose Rights

* Additional funding for development needed

14

✓ Requested