NASA Data Acquisition System (NDAS) Software Architecture

Dawn Davis, NASA Stennis Space Center
Michael Duncan, ASRC Research and Technology Solutions (ARTS)
Richard Franzl , Lockheed Martin- Stennis Space Center
Wendy Holladay, NASA Stennis Space Center
Peggi Marshall, ASRC Research and Technology Solutions (ARTS)
Jon Morris, Lockheed Martin- Stennis Space Center
Mark Turowski, NASA Stennis Space Center

NDAS Software Project
Overview

- Overview
  - The NDAS Software Project is for the development of common low speed data acquisition system software to support NASA's rocket propulsion testing facilities at John C. Stennis Space Center (SSC), White Sands Test Facility (WSTF), Plum Brook Station (PBS), and Marshall Space Flight Center (MSFC).

- Benefits
  - Creates a uniform, non-proprietary platform to meet goals in supporting propulsion system development
  - Consistency in data from across test locations/centers
  - Modular in design and able to support various test programs independent of the customer and hardware
  - Uniform software will add efficiency to projects as all personnel will be trained on one system

NDAS Software Project
Requirements

- Requirements for NDAS was created by a team comprised of representatives from the four NASA rocket propulsion testing facilities: SSC, WSTF, PBS, and MSFC.
- A review of the concept of operation of each facility was completed as well as trade studies of different data acquisition system software platforms and architectures utilized at places outside of NASA was performed.
- Low Speed Data Acquisition System (LSDAS) is utilized to provide real time display and recording of data. This data includes both analog and discrete measurements including but not limited to transducers, transmitters, thermocouples, test stand status monitoring, and valve commands and positions.
  - NDAS must be able to correctly process data from the sensors and convert the data to engineering units.
- In order to ensure the LSDAS meets performance requirements, "system calibrations" of the LSDAS are required.
  - As a minimum the software will provide capability to perform voltage insertion calibrations, shunt calibrations, and frequency calibrations.

NDAS Software Project
Basic Requirements

- The LSDAS samples at nominal sample rates of 250 samples/second.
  - The software must support this sampling rate and also have the capability of recording data at various recording rates.
  - Must operate 24 hours per day, 7 days a week, acquiring data continuously
- The LSDAS samples at nominal sample rates of 250 samples/second.
  - The software must support this sampling rate and also have the capability of recording data at various recording rates.
- A “roadmap” is used to document the test configuration as well as to configure the hardware utilized by the LSDAS
  - The software must provide the capability to store configuration for the hardware, the sensors, and sensor data, as well as any supplemental information needed by the engineers to operate the system.
  - Reports must also be generated that will assist in maintaining the LSDAS and tracking configuration changes between tests.
- The software should be capable of handling redundancy of hardware.

NDAS Software Project
Challenges

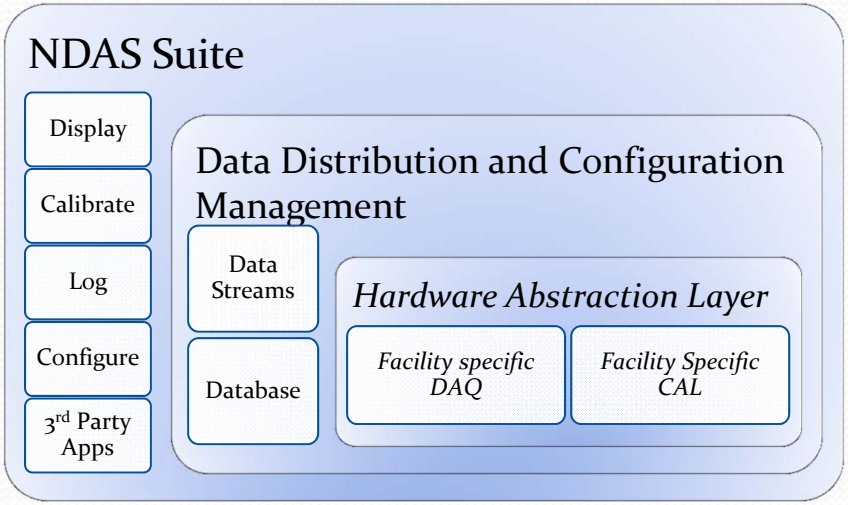
- Each center utilizes different hardware and system configurations for the LSDAS.
- Some systems are aging and may require replacement in the next 10 years.
- The Centers operate the LSDAS differently:
  - Day to day operations: Some Centers operate software 24 hours day, logging data continuously at variable rates. Other Centers operation only required to support test operations.
  - Interfaces: In some instances, the test stand may share portions of the data acquisition system hardware thru patching to other systems such as a facility control system, therefore portions of the software may be safety critical.
  - System Calibrations: The types of calibrations may vary depending on instrumentation types, uncertainty requirements or Center processes.
  - Contents of database: Some Centers maintain the entire test stand configuration, others document the information for the sensor/test configuration required to operate the LSDAS.
- Centers need to have the ability to support customer specific requirements. This may require changes to the software.

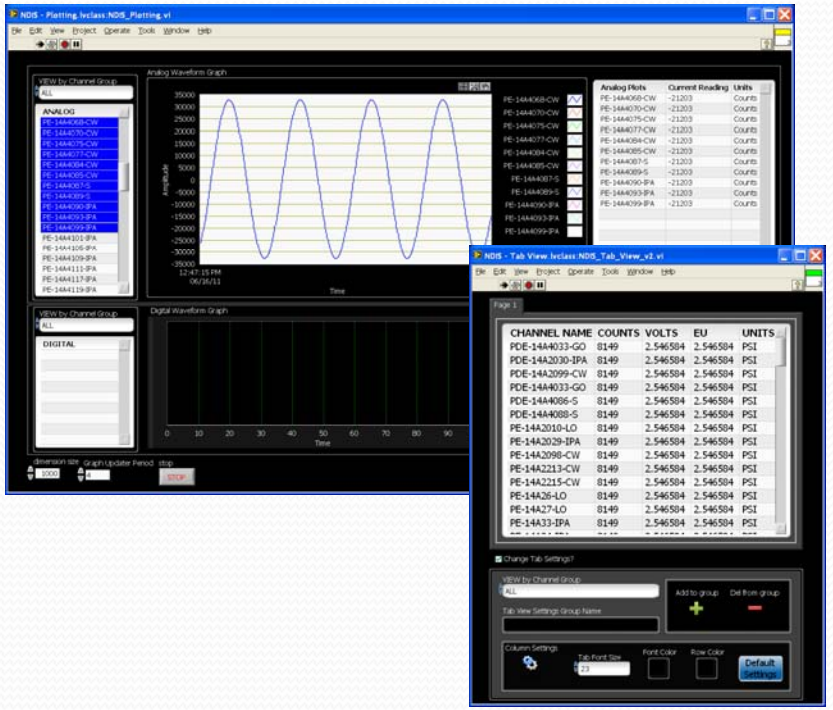
NDAS Software Project
Development Goals

Stennis Space Center

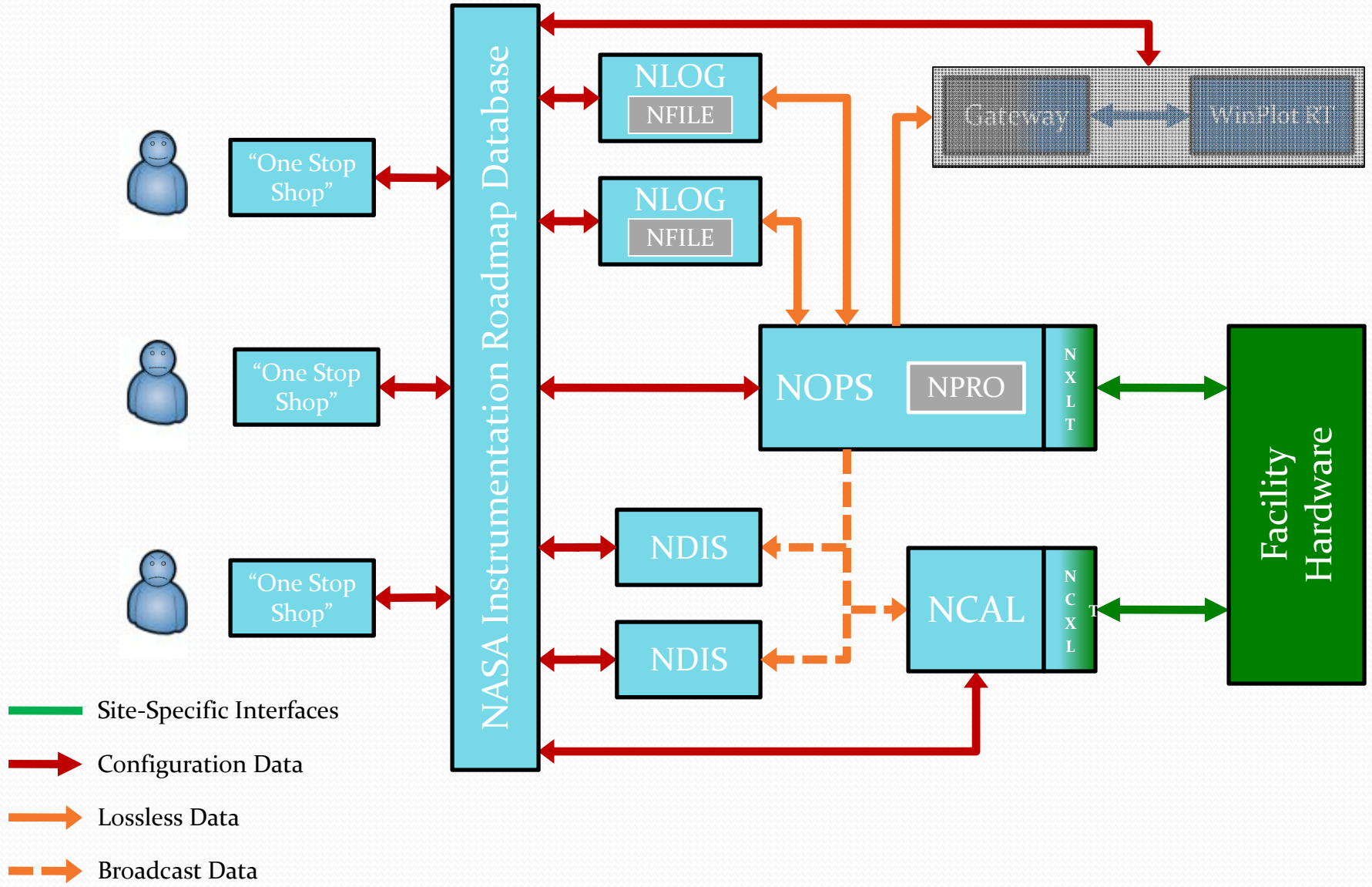
The NDAS project developed an architecture that would provide:

- **Adaptability:** Hardware abstraction layer adaptable to different acquisition systems with minimal effort.
- **Modularity:** Functional areas designed as separate modules to simplify maintenance and life cycle support.
- **Extensibility**: Displays and data output files can be customized via a standardized plug-in architecture.
- **Flexibility:** Innovative hierarchical and self-referential database architecture allows for flexibility to deploy to any facility.
- **Unified System Configuration:** The system, measurements and calibrations are managed and configured within a common user interface.
- **Streamlined Operations:** Run-time processing and analysis minimizes post-test data processing turnaround time.





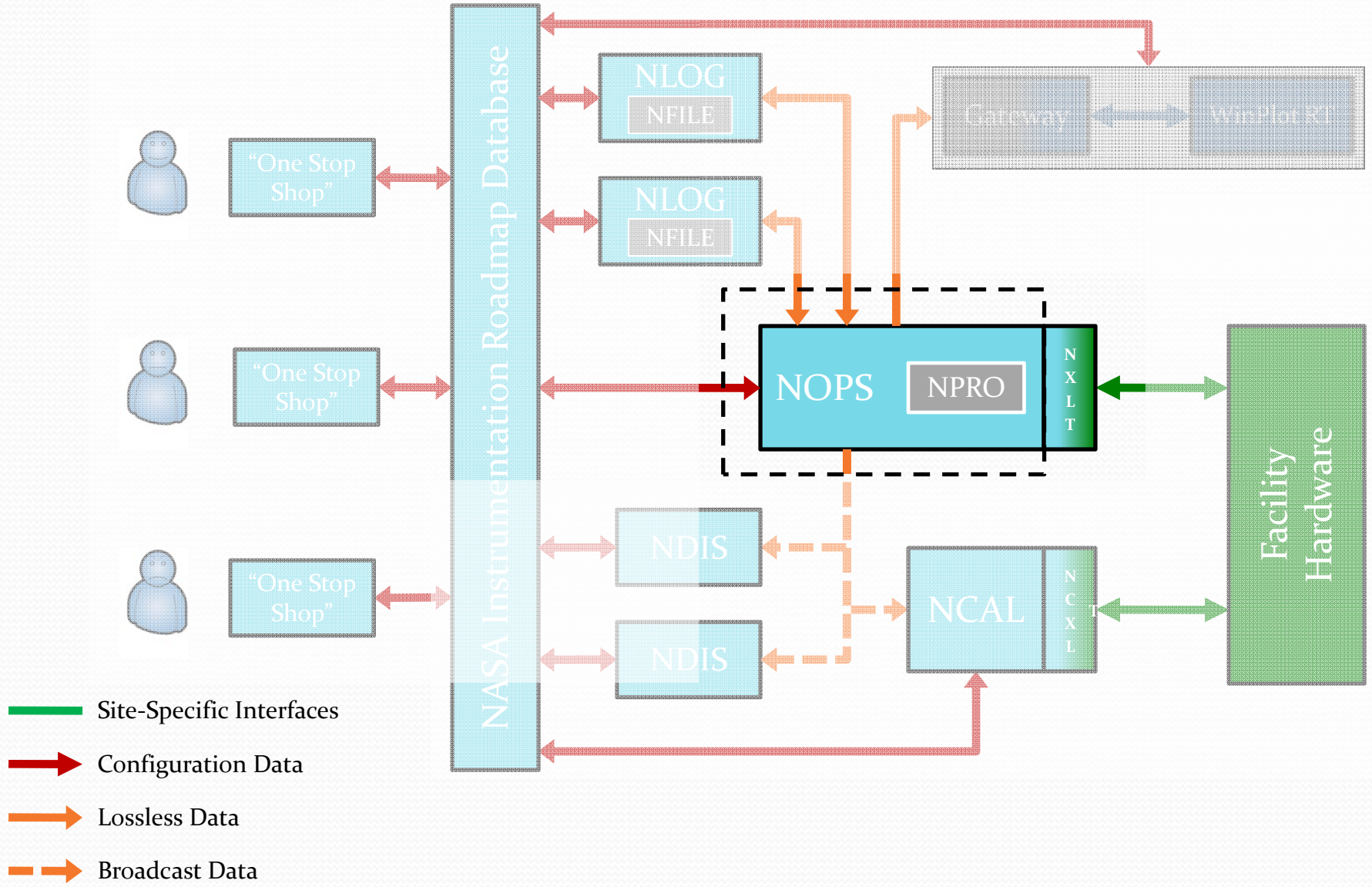
NDAS Software
Overview



NDAS Software Project
Modules

- The adopted architecture divides the software into modules to implement the functionality and system requirements contained in the project's requirements document. It also allows for the flexibility and adaptability necessary in deploying the software at the different centers. The NDAS modules are:
  - **NXLT** (Translation Layer)
  - **NOPS** (DAS Operations)
  - **NCAL** (Calibration)
  - **NDIS** (Display)
  - **NPRO** (Engineering Unit Processing)
  - **NLOG** (Data Logging)
    - **NFILE** (Data File)
  - **NIRD** (NASA Instrumentation Roadmap Database)
  - **NDMS** (Distributed Data Management System)
- The software is developed in Labview. Able to take advantage of the flexibility inherent to Labview.
- Database is developed in Microsoft SQL.

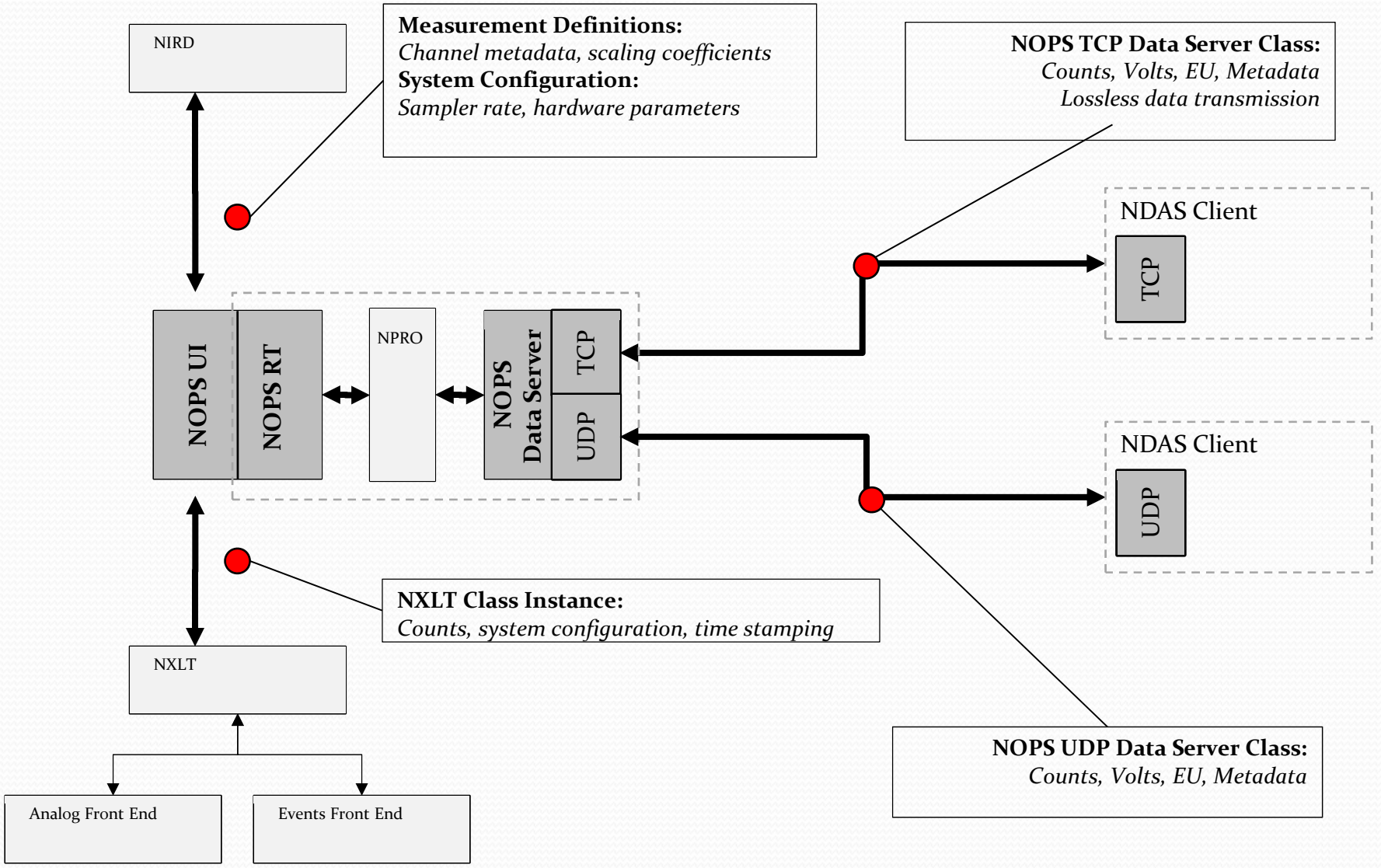
NOPS


"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NASA Instrumentation Roadmap Database
NLOG
NFILE
NLOG
NFILE
Gateway
Workstation
NOPS
NPRO
NXLT
NDIS
NDIS
NCAL
NCXT
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

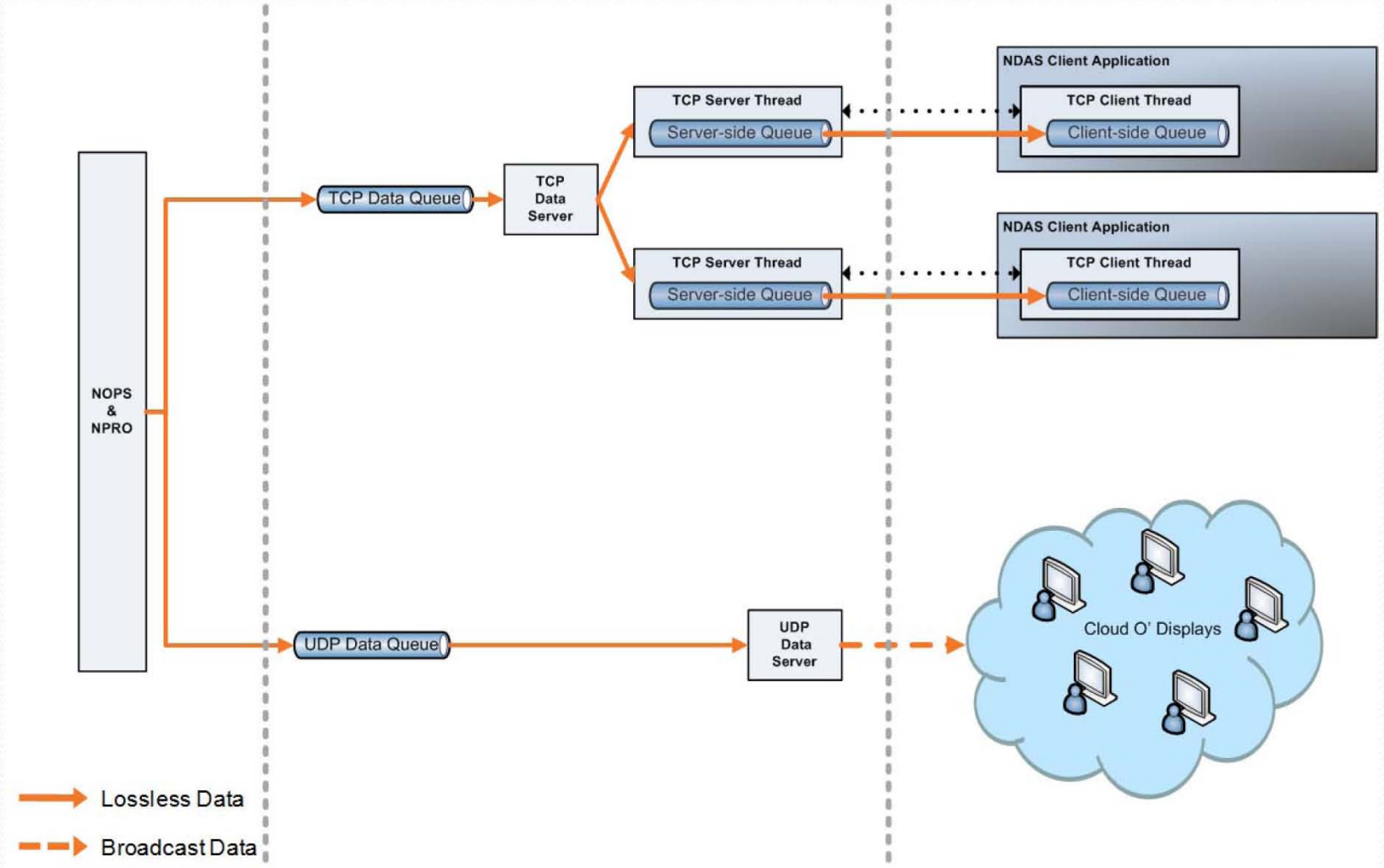
NOPS
Overview

- This module manages the NXLT connection to the acquisition hardware.
- Also manages the data stream connections to the NDAS distributed software elements such as NLOG, NCAL, and NDIS.
- It provides the framework for the NPRO module to perform Engineering Unit conversion on all data at run-time.
- NOPS reports and manages application level errors to the NLOG API.

NOPS
Detailed View



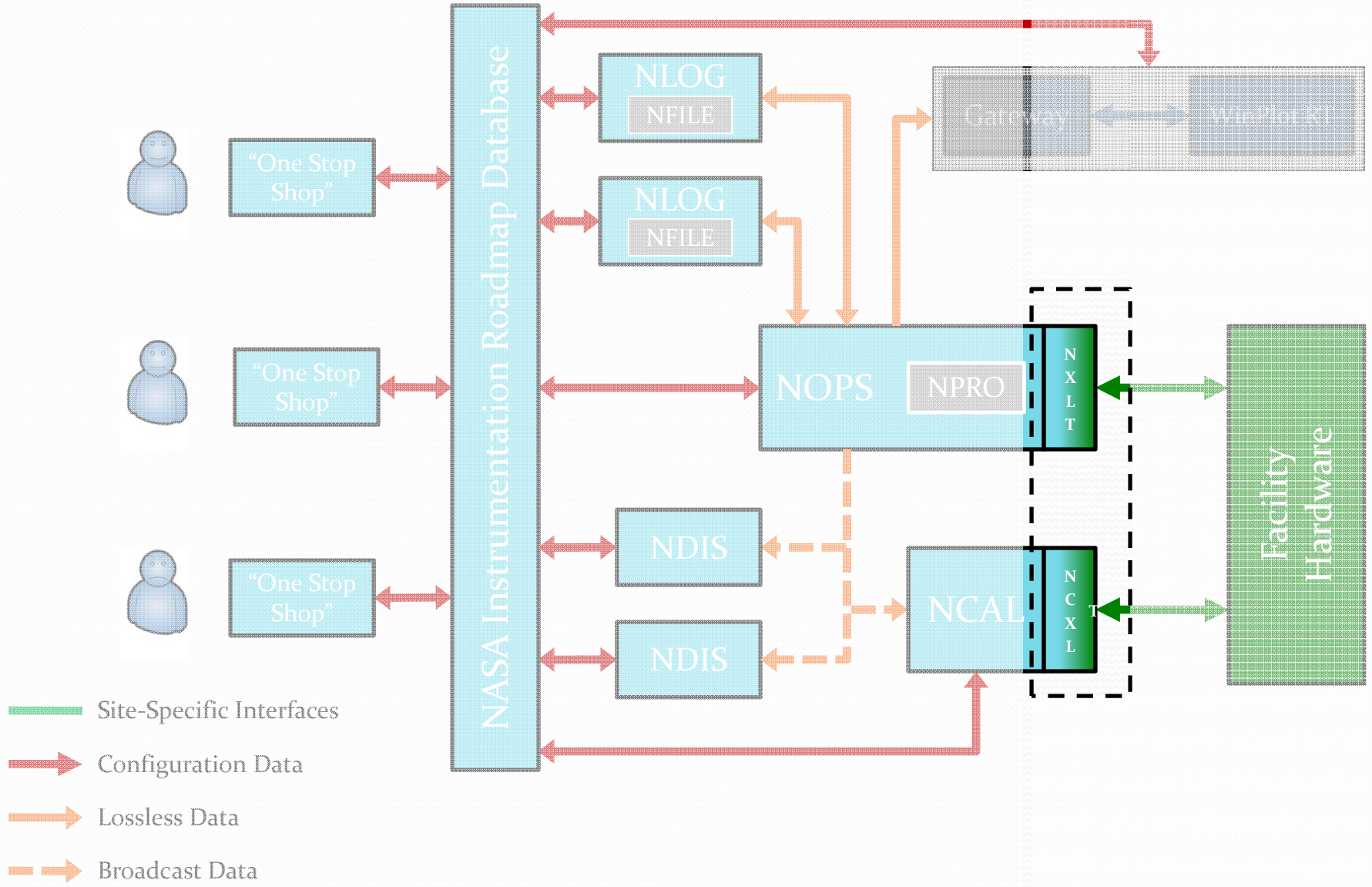
NOPS
Data Servers



NOPS
User Interface

- Implemented using the LabVIEW xCE and AMC Frameworks
- The User Interface communicates with the NOPS RT system over UDP to exchange state and command information.
  - All messages generate a response so that the UI can verify that it was received by the RT system.
- The NOPS UI acts as a pass through to configure the NXLT and NPRO layers.
  - Hardware is configured in the NOPS UI using database and user inputs and then sent to the real-time system.
  - Measurement definitions are validated in the NOPS UI and then sent to the real-time system for execution.
- All configuration changes require the user to verify their actions with a second click on a floating prompt

NXLT and NCXLT



NXLT/NCXLT
Overview

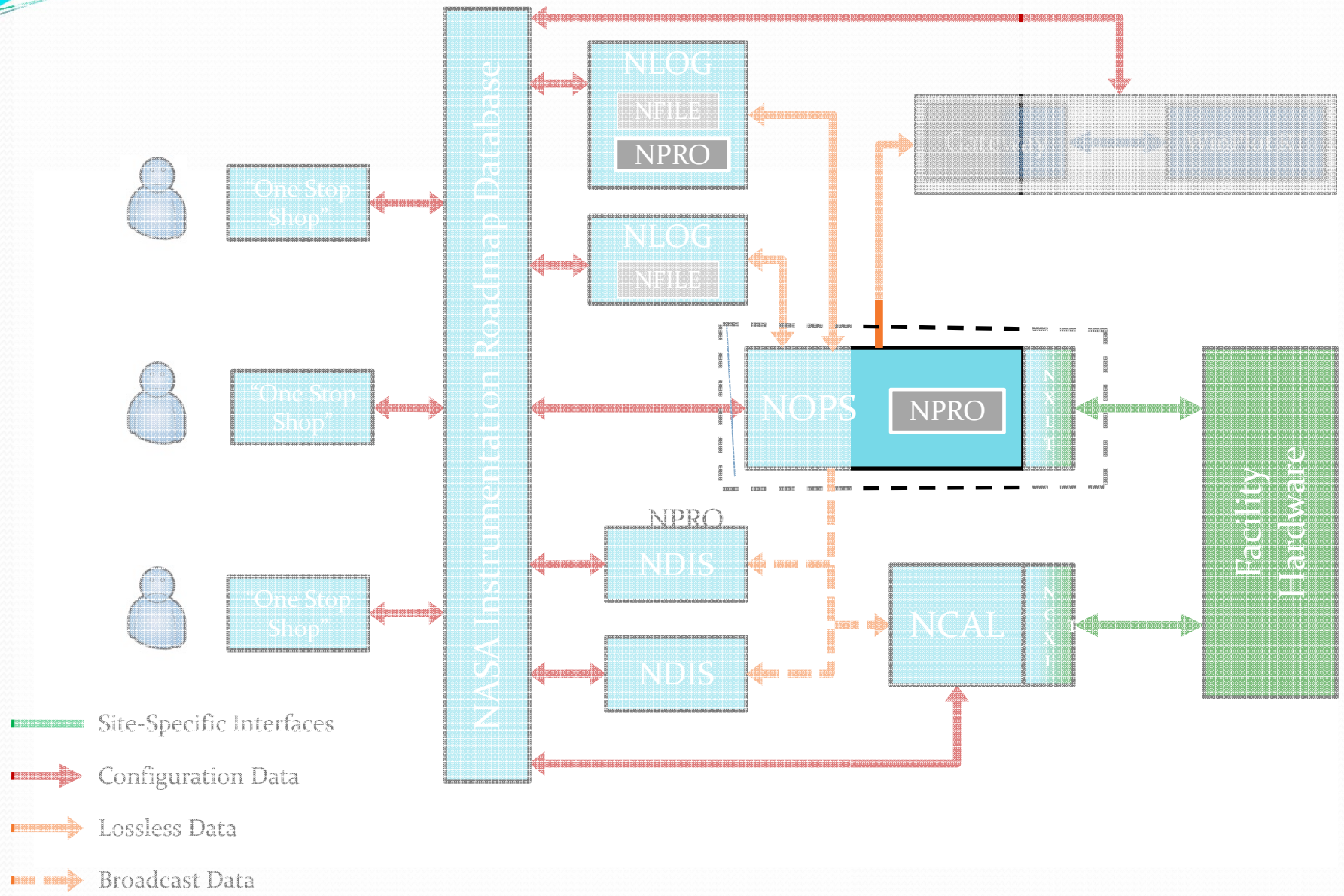
NXLT

- Provides an abstraction layer between NOPS and site specific acquisition hardware
  - Masks the differences in hardware from the application software
- Capable of supporting multiple acquisition front ends simultaneously
- Initialized by NOPS using information stored in the NIRD database
  - Acquisition hardware is configured during system initialization and stored in the NIRD

NCXLT

- Provides an abstraction layer between NCAL and site specific calibration sources and signal conditioners
  - Masks the differences in hardware from the application software
- Initialized by NCAL using information stored in the NIRD database
  - Calibration and signal conditioning hardware is configured during system initialization and stored in the NIRD

NPRO

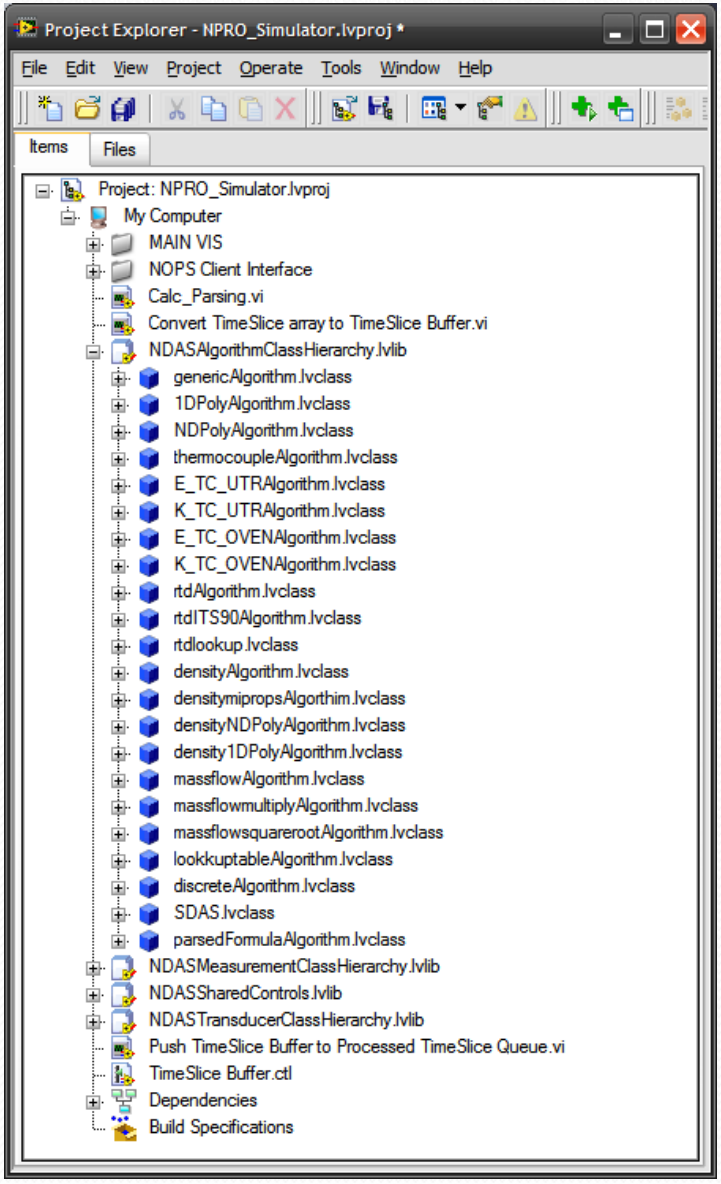

NASA Instrumentation Roadmap Database
NLOG
NFILE
NPRO
NLOG
NFILE
"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NOPS
NPRO
NXIF
NPRO
NDIS
NDIS
NCAL
NCXF
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

NPRO Module
Description

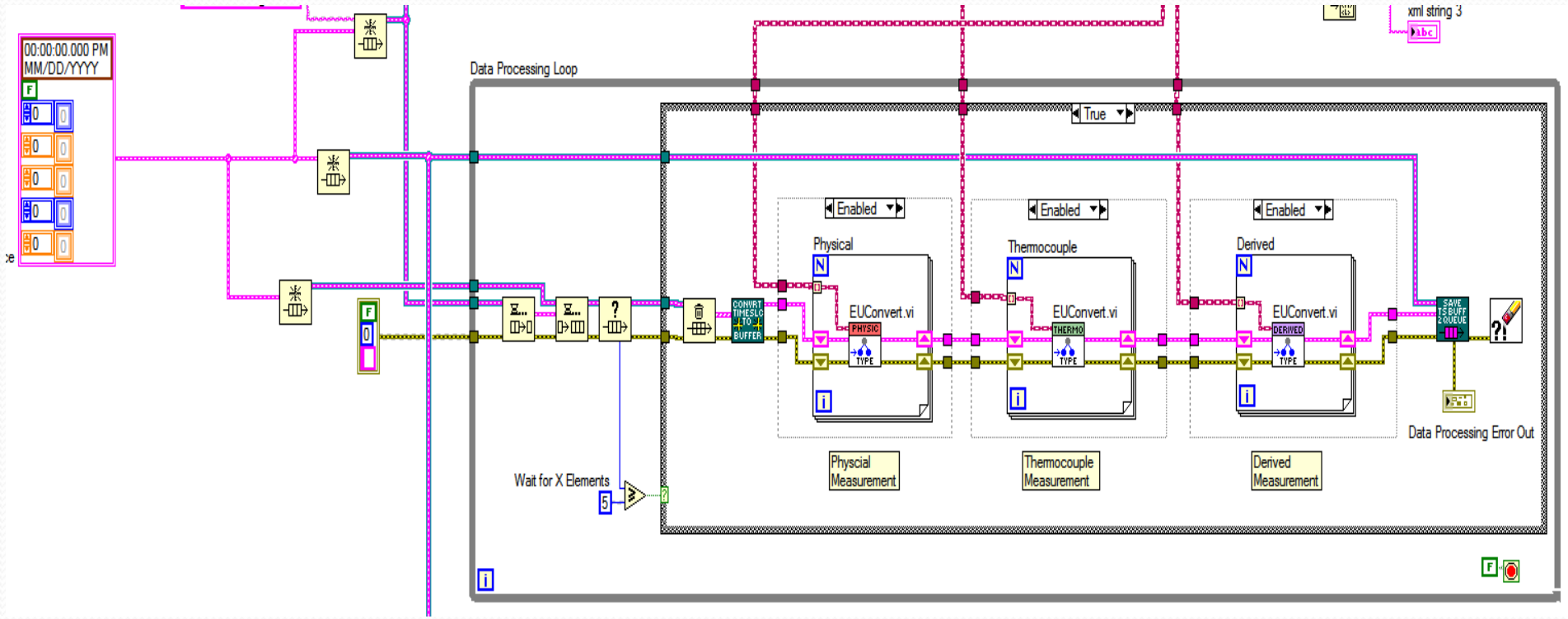
- The NPRO can function as either an API called by NOPS or a stand-alone application that supports the production of Engineering Unit converted data for real-time display and storage. All data required to support EU conversions is housed in the NIRD.
- NPRO provides output of data to support the NFILE module. That data becomes the official processed data reviewed and released to customers. In addition, this data is provided in real-time to NDIS to support displays during test activities.
- NPRO's architecture provides the flexibility to enable expansion to meet customer specific requirements.
- Utilize a class structure to organize the different engineering unit processes.
- NPRO provides Engineering Unit data including but not limited to first order calculation, multi-order, discretes, pulse data, RTDs, thermocouples, density, mass flow, and special calculations. This module will be capable of handling scripts to automate processing.
- NPRO incorporates all existing calculations with the ability to easily insert additional calculation types using Object-Oriented Design and LabVIEW equations parsing via formula nodes.
- Engineering Unit data is generated using NIST traceable techniques, i.e.. ITS90, NIST lookup tables, MiProps.

NPRO
Algorithm Class Hierarchy

Class Structure showing the Algorithm Class

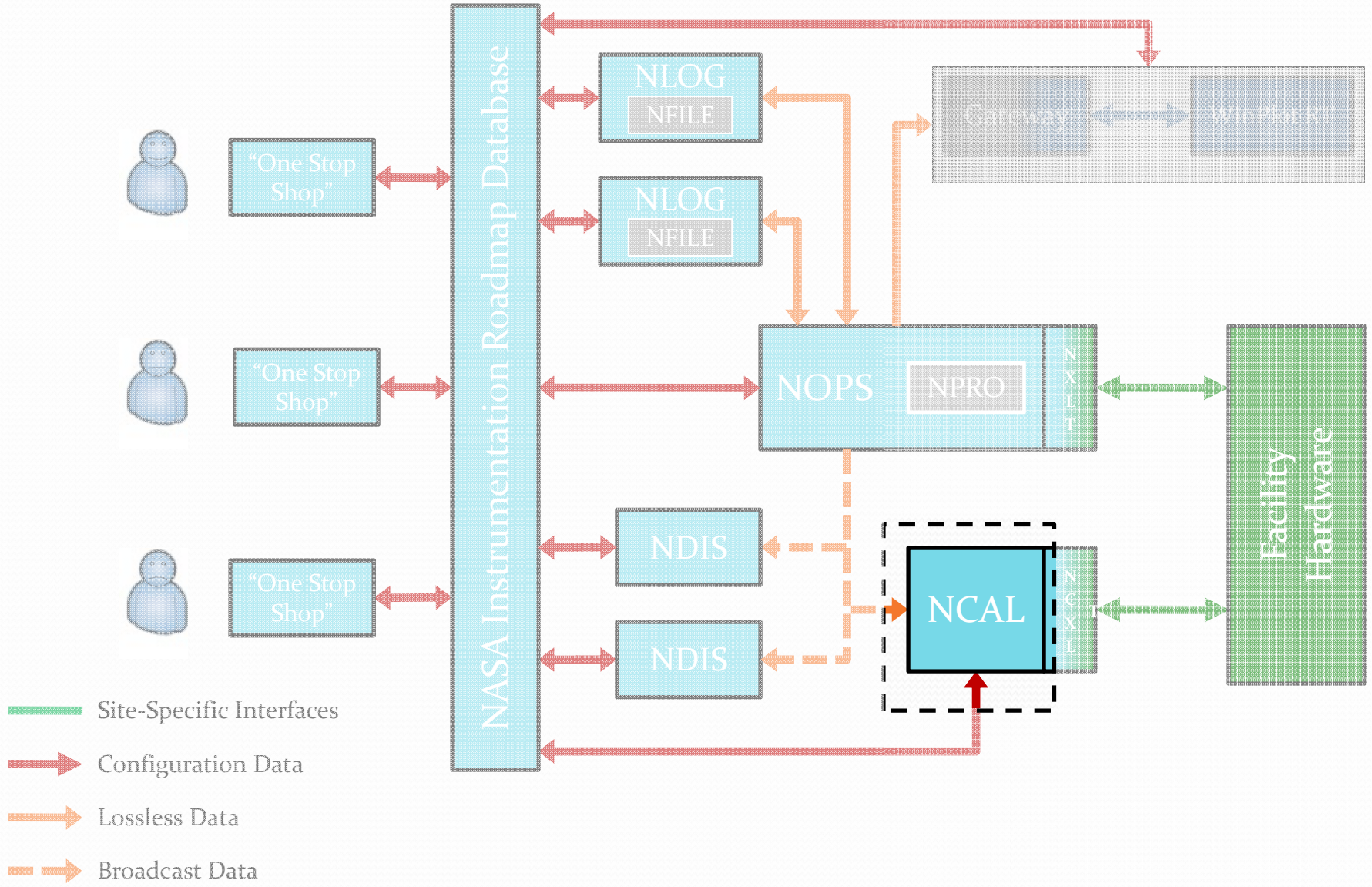


NPRO
Implementation



NPRO Implementation – Consists of a staged approach
Physical ----------> Thermocouples ----------> Derived
EUConvert is a member of Measurement Class and determines which Algorithm is instantiated

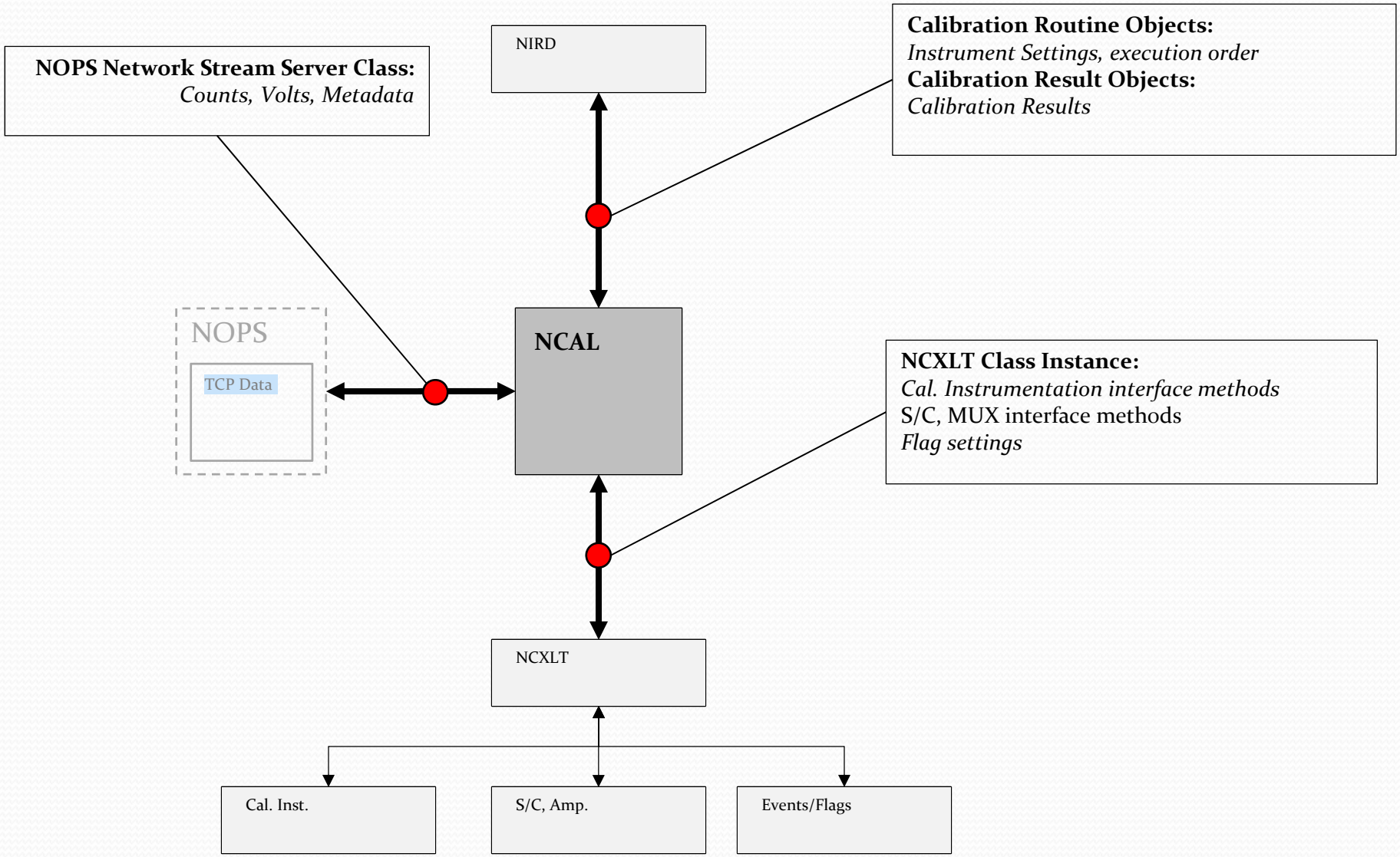
NCAL


"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NASA Instrumentation Roadmap Database
NLOG
NFILE
NLOG
NFILE
Gateway
NOPS
NPRO
NDIS
NDIS
NCAL
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

NCAL

- NCAL performs calibration, linearity/hysteresis, and Measurement System Analysis
  - Modular, object oriented approach to calibration based on fundamental calibration "types"
  - Calibrations are performed based on calibration instructions
  - The architecture allows for flexibility in defining a calibration process that may differ among centers or test programs
- Interfaces with hardware through NCXLT translation layer
  - All access to hardware is high level through NCXLT class instance methods.
- Acquires data through NOPS data stream
  - Reads NOPS network stream server.
- Records and analyzes calibration result data
  - Self-contained with no reliance on separate downstream loggers/processors.
- Interfaces directly with database through NIRD API
  - Retrieves, commits calibration routines and results with full audit control

NCAL


NOPS Network Stream Server Class:
Counts, Volts, Metadata
NIRD
Calibration Routine Objects:
Instrument Settings, execution order
Calibration Result Objects:
Calibration Results
NOPS
TCP Data
NCAL
NCXLT Class Instance:
Cal. Instrumentation interface methods
S/C, MUX interface methods
Flag settings
NCXLT
Cal. Inst.
S/C, Amp.
Events/Flags

NCAL
Basic Calibration Types

- NCAL defines specific calibration “types”. The calibration types are listed below:
  - Voltage (VCAL)
  - Frequency (FCAL)
  - Shunt (SHUNT)
  - Ambient (AMB)
  - Short (SHORT)
  - Programming (PROG) - *special “non-calibration” type used strictly for programming hardware*

NCAL
Calibration Instructions

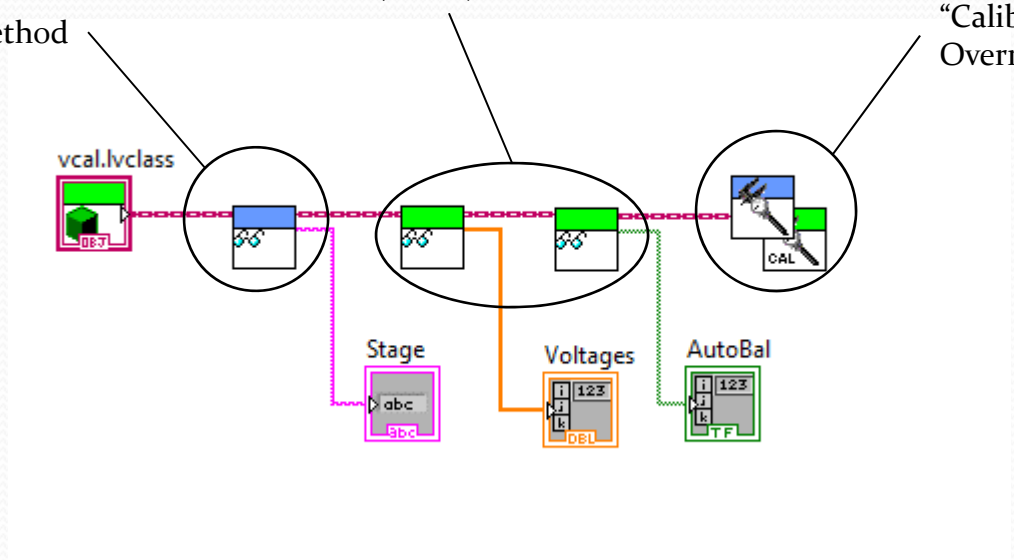
- Calibration "Instructions" encapsulate the information necessary to perform a particular type of calibration on one or more measurement IDs (MSID, channel…).
- Calibration instructions are assigned unique IDs.
- Calibration instructions are stored in the database.
- NCAL defines a general object class for all calibration instructions and object sub-classes for each specific calibration type.
- Calibration instruction classes include a generic string array to provide a mechanism for sending any special commands to hardware through NCXLT.

NCAL
Calibration Instruction Methods

Child (VCAL) Methods

Parent (CAL) Method

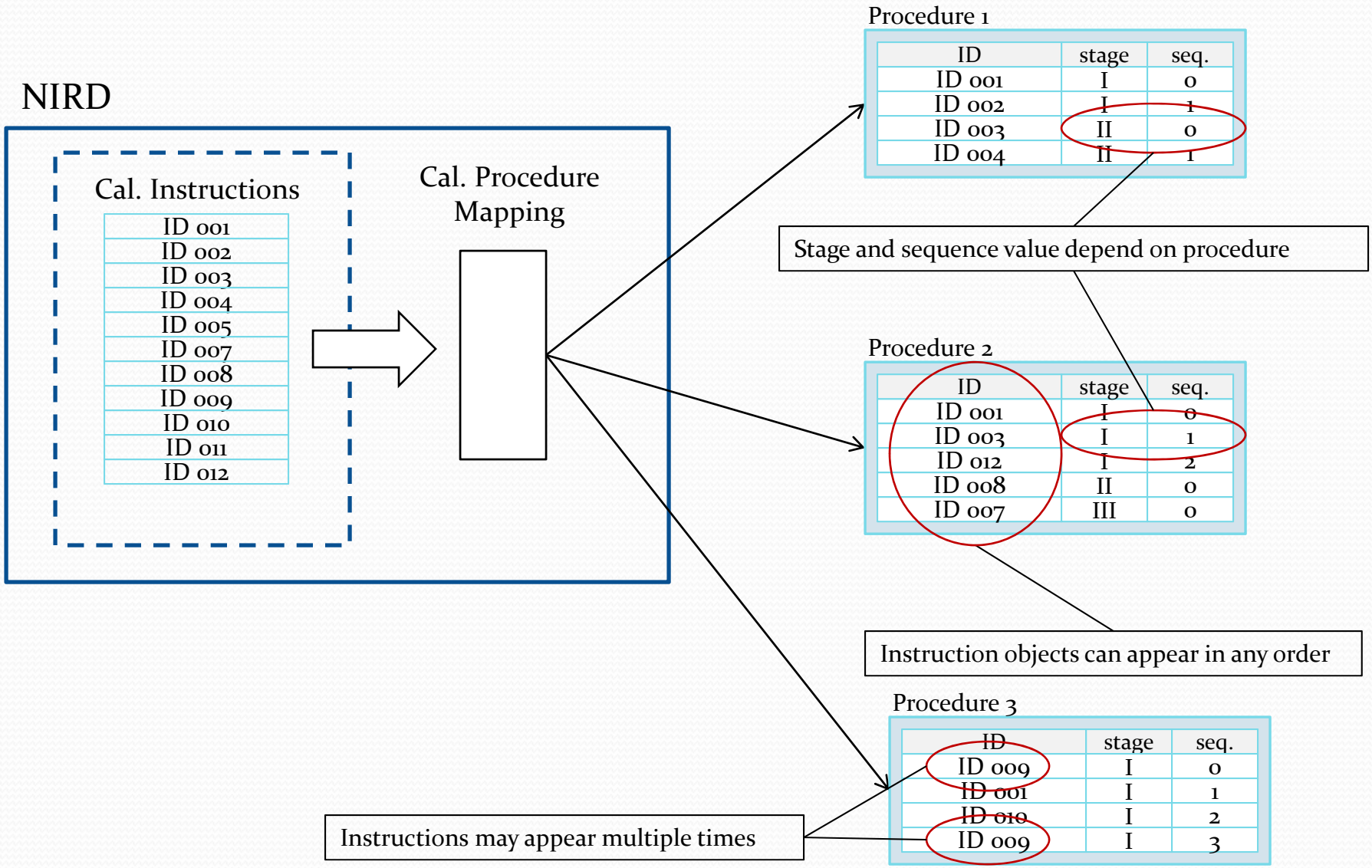
"Calibration" Dynamic Dispatch Override



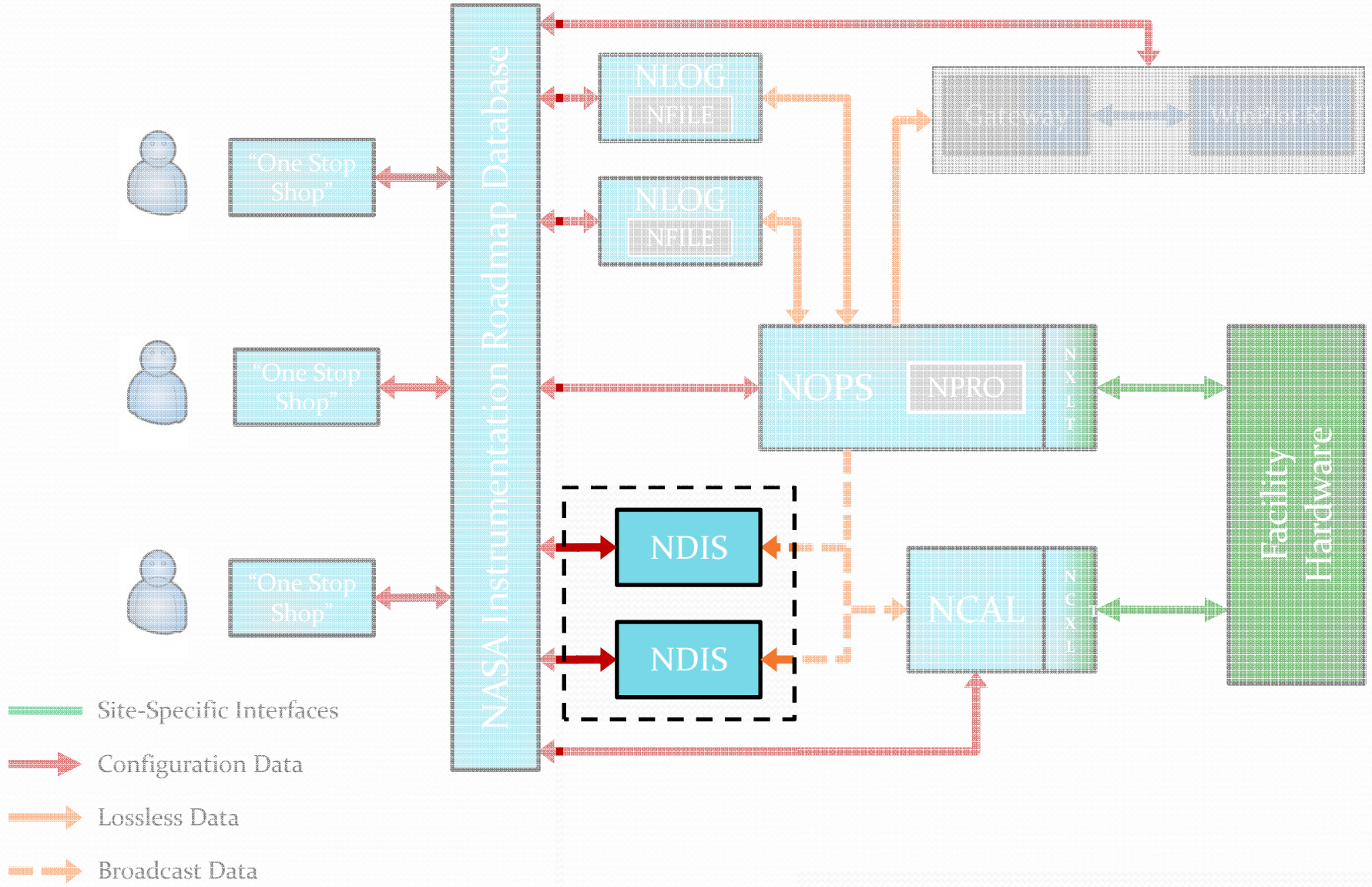
NCAL
Calibration Procedures

- Calibration "Procedures" are collections of calibration instructions executed in a **specified order**.
- Calibration procedures are given arbitrary names.
- The execution order is defined by the "stage" and "sequence" values of the instruction objects.
  - Stage – Order of single group of instructions within a procedure.
  - Sequence – Order of an individual instruction within a stage.
- NIRD maintains the **relationship** between a calibration procedure, its constituent instructions and their execution order.
- NIRD is queried by procedure name.
- NIRD delivers the procedure to NCAL as a group of instruction objects with stage and sequence values accordingly set.
- NCAL executes all or portions of the procedure.

NCAL
Calibration Procedures

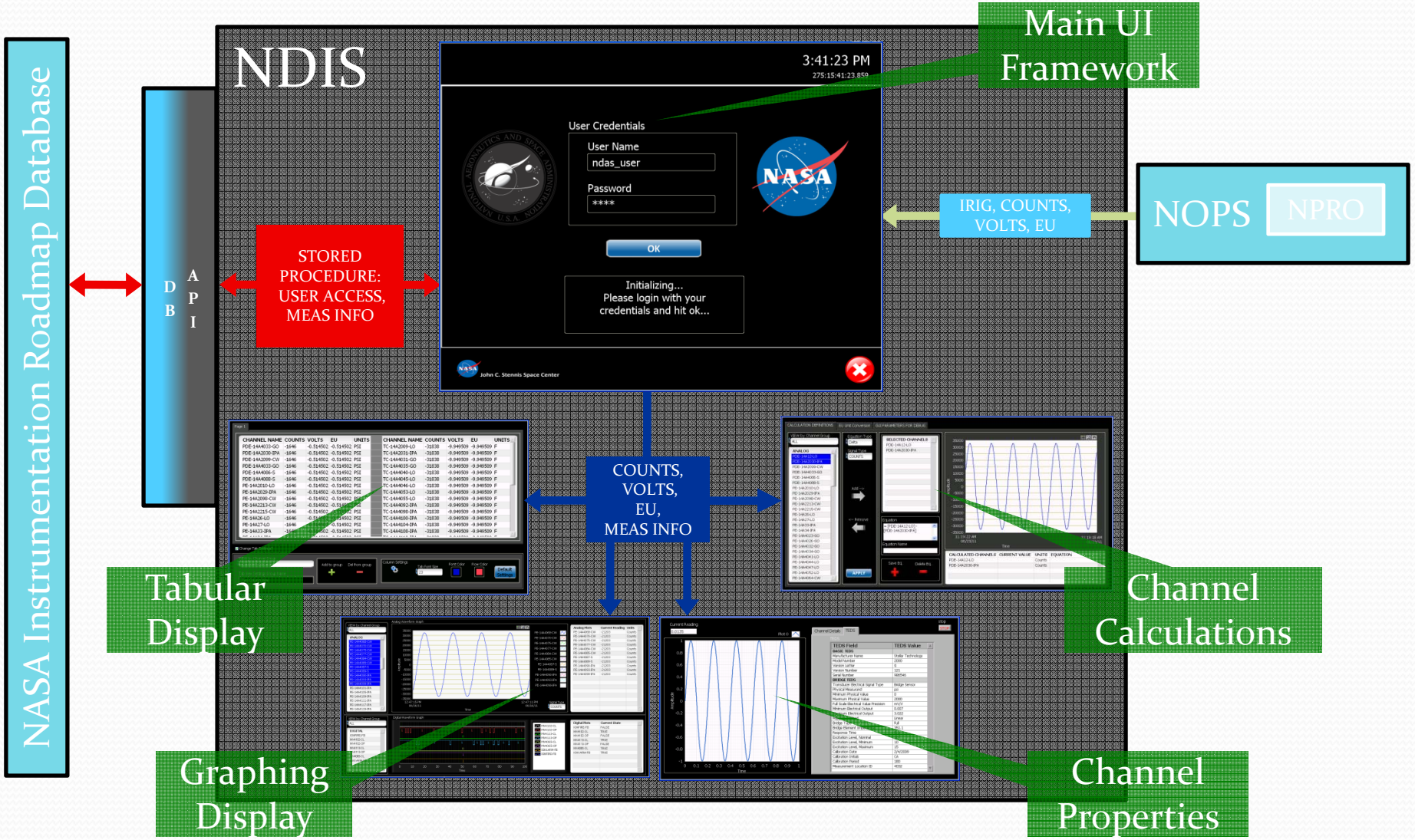


NDIS


"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NASA Instrumentation Roadmap Database
NLOG
NFILE
NLOG
NFILE
NOPS
NPRO
NXTH
NDIS
NDIS
NCAL
NCXH
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

NDIS – Input Diagram

Data - Database, IRIG, Counts, Volts, EU



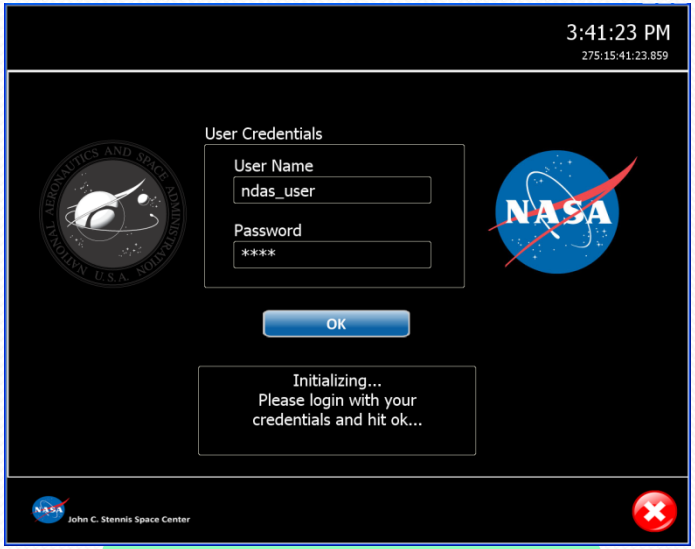
NDIS – User Interface Framework
OO-Based State Machine Producer/Consumer Design

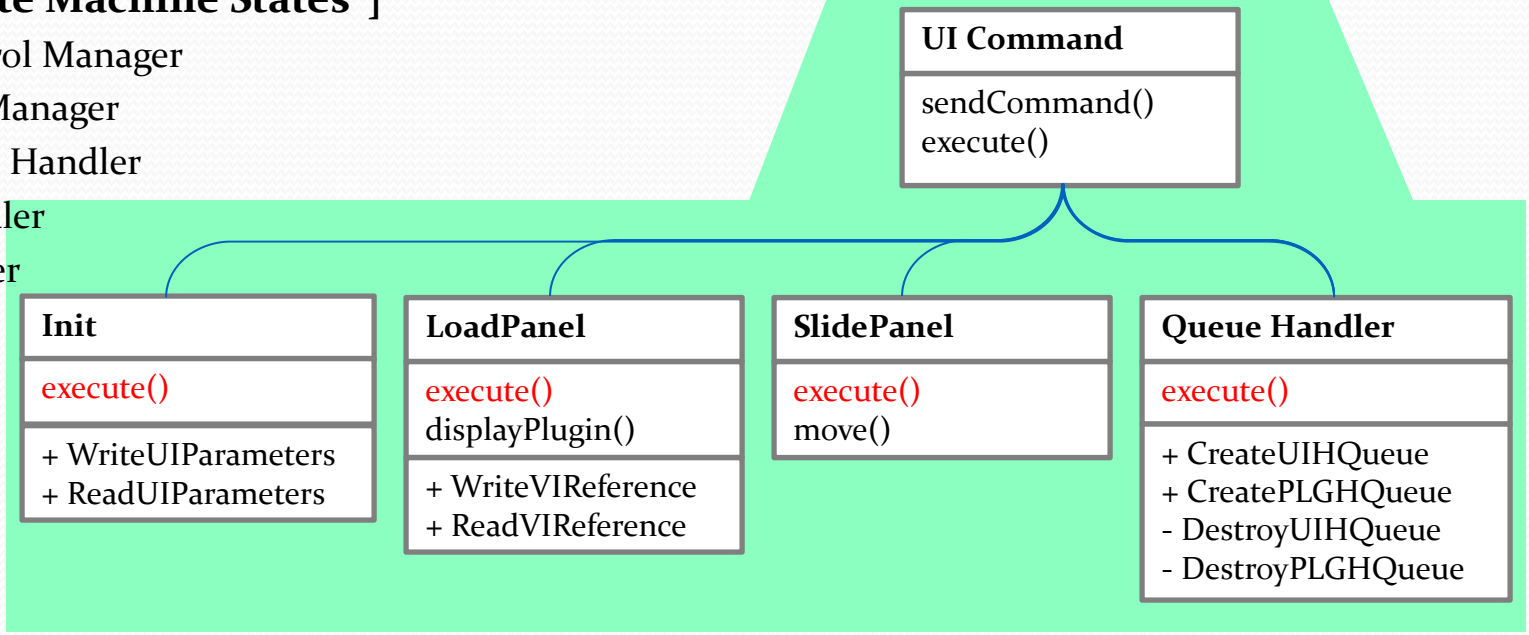
UI Framework – OO State Machine

- Follows recommended NI software design patterns
  - Queued event, OO state machine, consumer/producer
  - Aka 'Chain of Command Pattern'
- Uses **Dynamic Dispatching** to determine (at run-time) which version of the execute method runs
- Execute method acts as a "OO state machine"
- Architecture allows for continuous operation
- portions of the procedure.

Execute () – ["State Machine States"]

- Plugin Control Manager
- UI Control Manager
- DAQ Stream Handler
- Queue Handler
- Error Handler
- DB Handler
- Load Panel
- Slide Panel
- Login
- Logout
- Init
- Exit

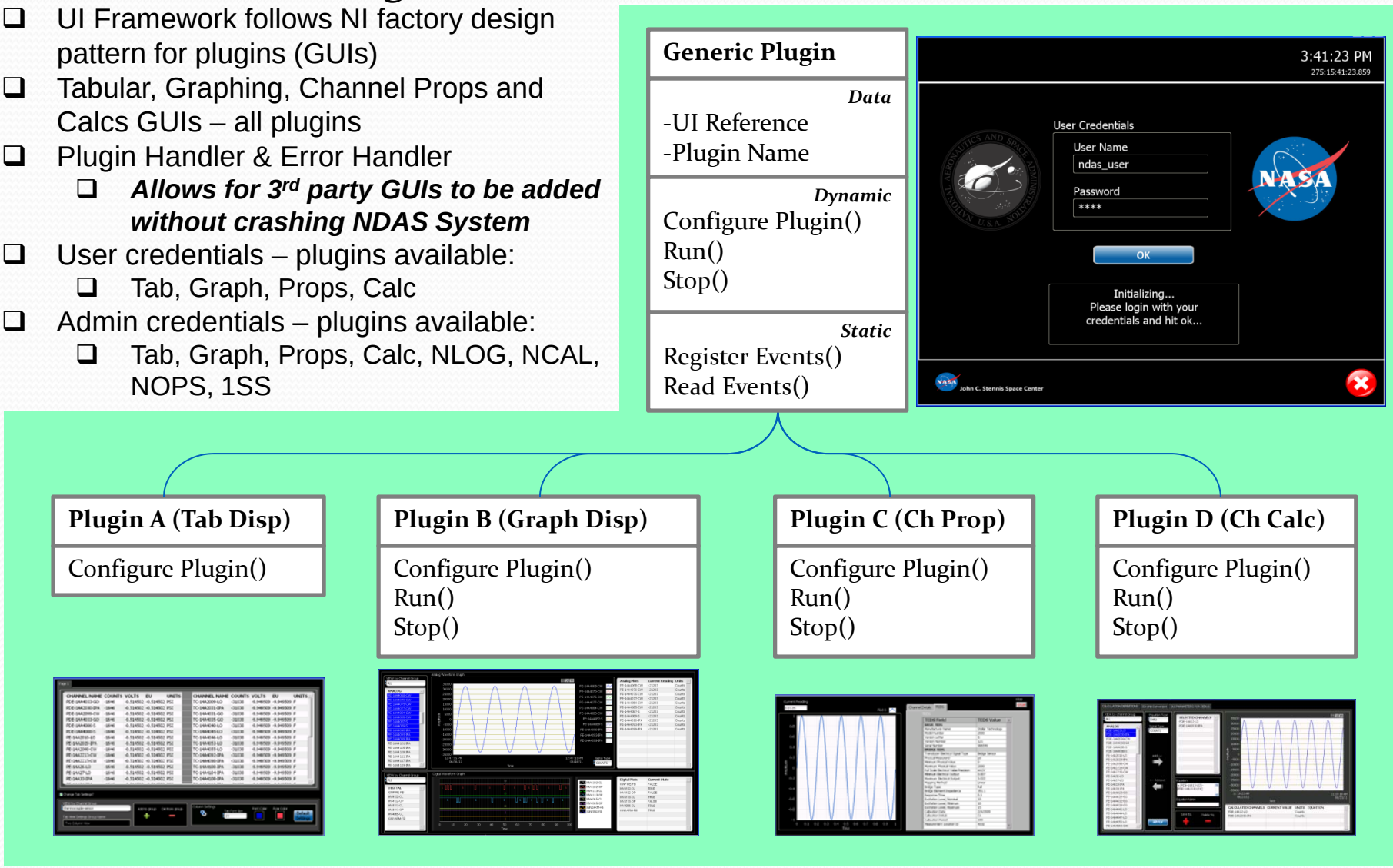




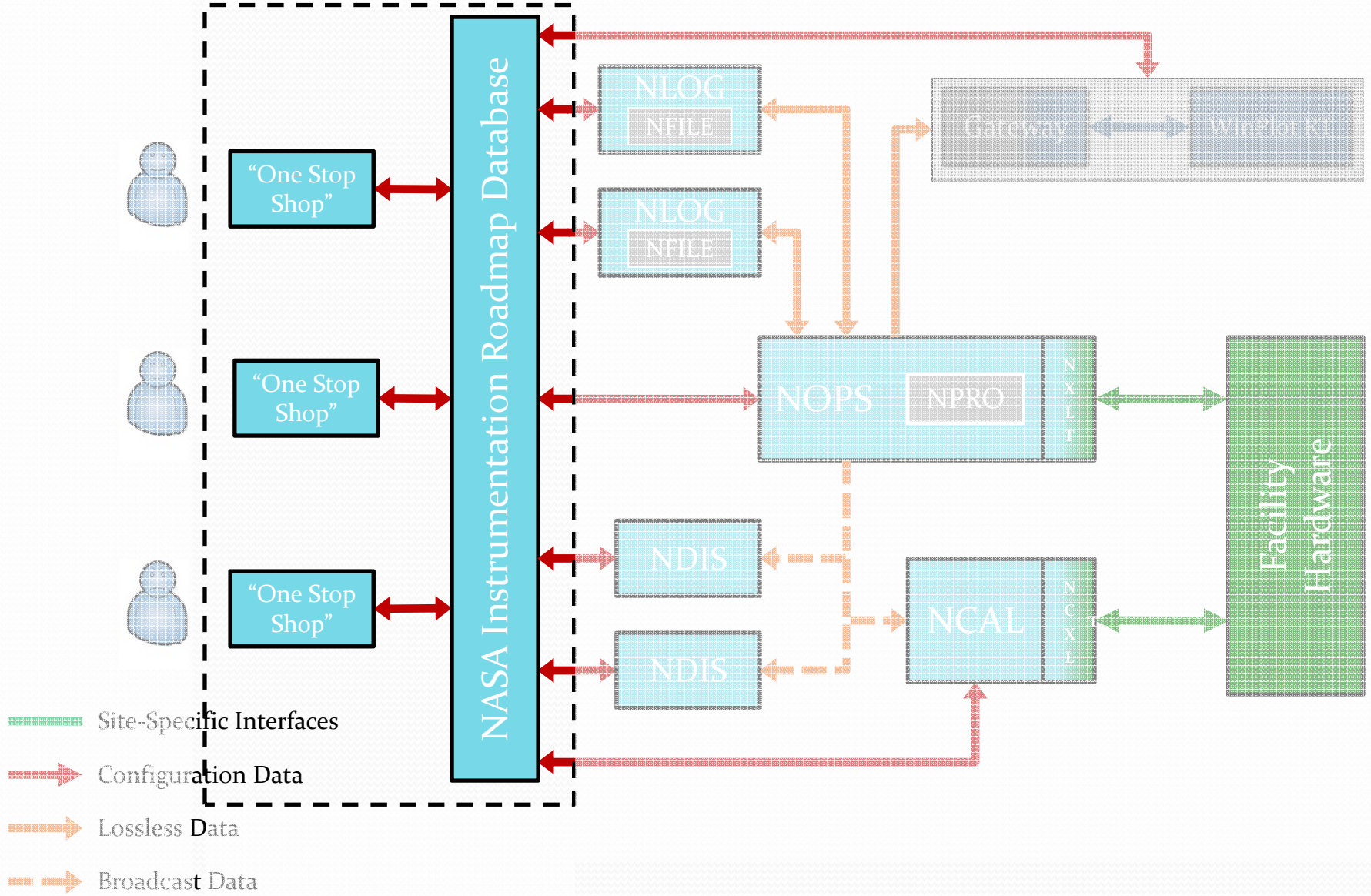
NDIS – User Interface Framework
Plugin GUIs

UI Framework – Plugins

- UI Framework follows NI factory design pattern for plugins (GUIs)
- Tabular, Graphing, Channel Props and Calcs GUIs – all plugins
- Plugin Handler & Error Handler
  - ***Allows for 3rd party GUIs to be added without crashing NDAS System***
- User credentials – plugins available:
  - Tab, Graph, Props, Calc
- Admin credentials – plugins available:
  - Tab, Graph, Props, Calc, NLOG, NCAL, NOPS, 1SS

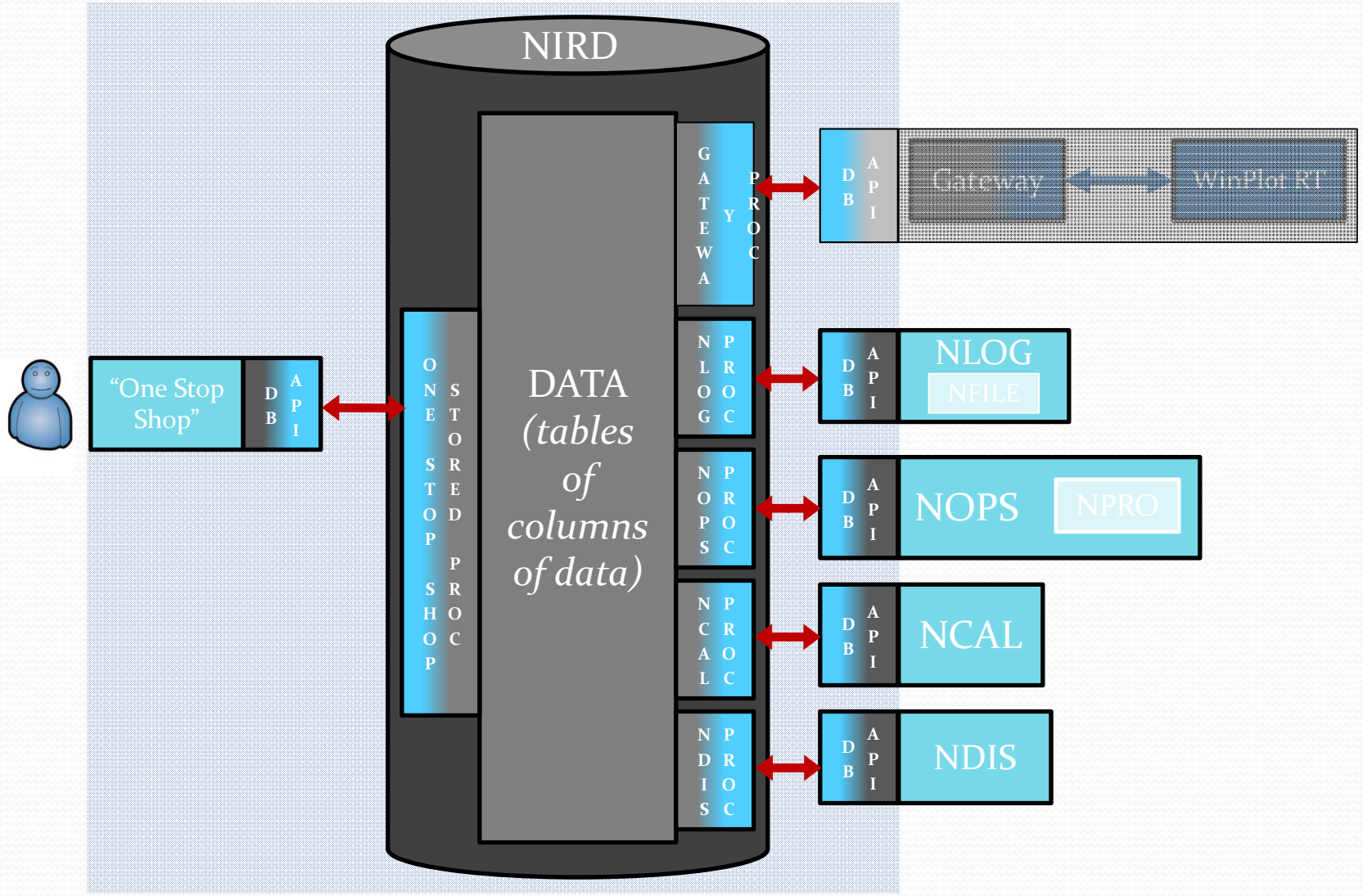


NIRD


"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NASA Instrumentation Roadmap Database
NLOG
NFILE
NLOG
NFILE
NOPS
NPRO
NDIS
NDIS
NCAL
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

Components of NIRD

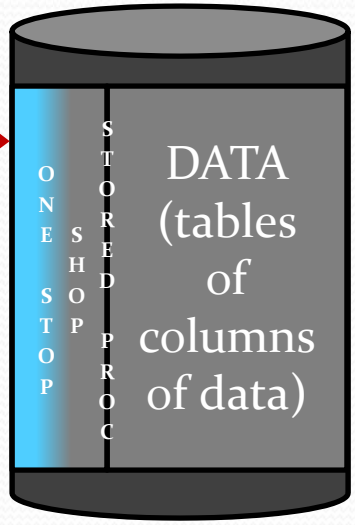
One Stop Shop, DB APIs, Stored Procedures, Data

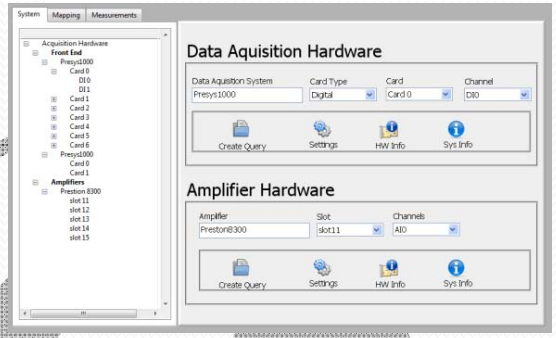


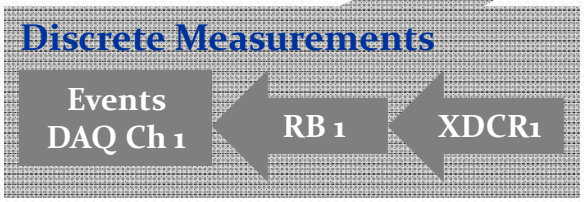
NIRD – *User Interface*
One Stop Shop

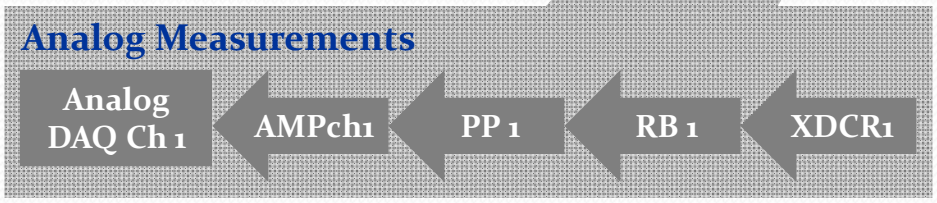
One Stop Shop Capabilities

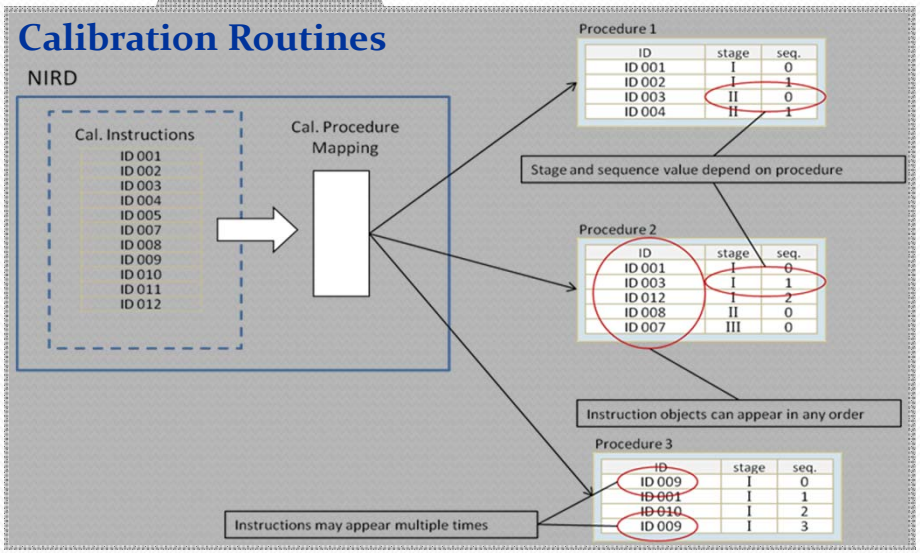
- GUI - main user entry point into NIRD
- Display system HW components
- Setup measurements
- Setup calibration routines
- Create custom database views/reports







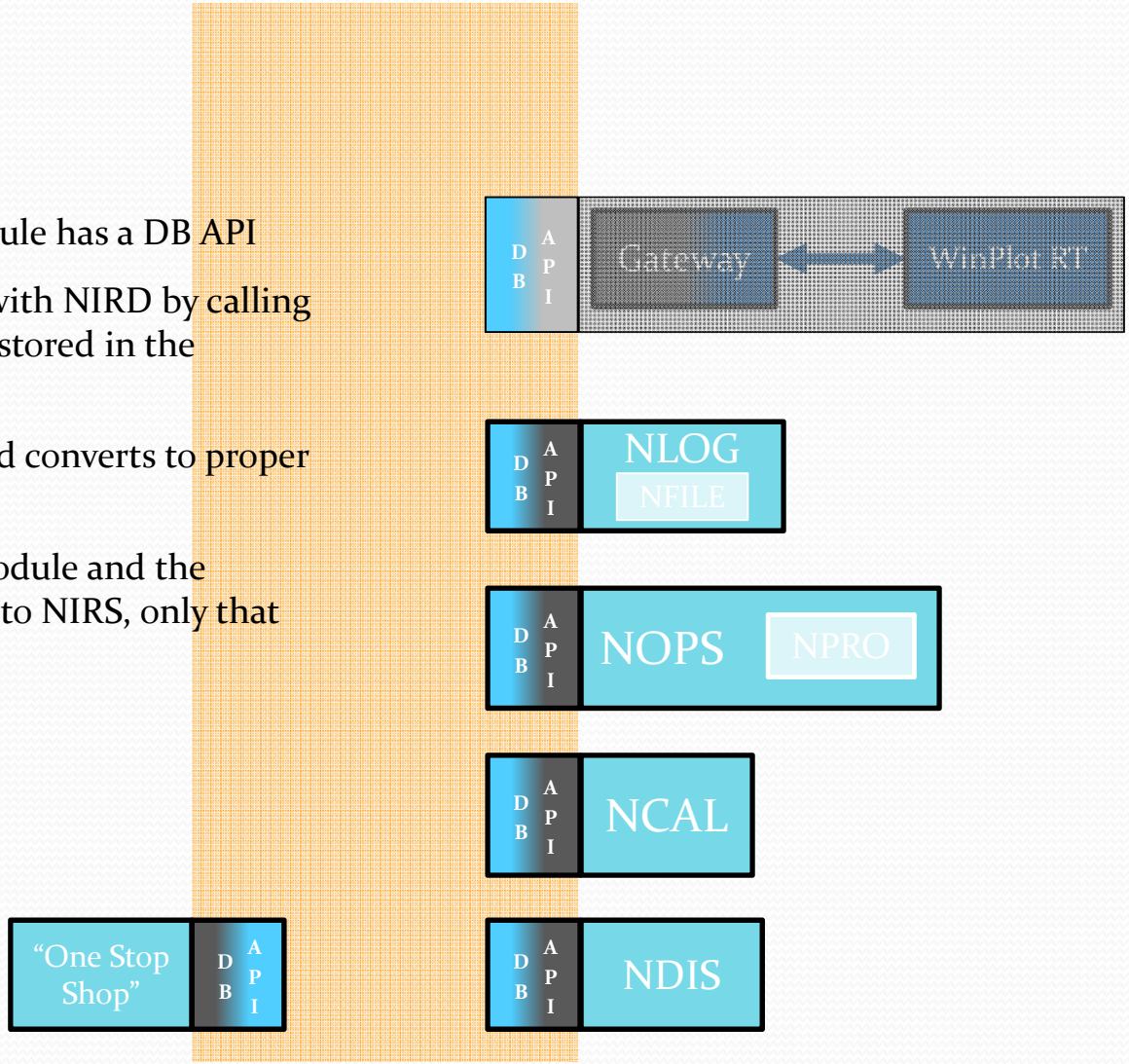




NIRD – *NDAS Module Interface*
NDAS Module Database APIs

LabVIEW Database APIs

- Each NDAS LabVIEW software module has a DB API
- The APIs are used to communicate with NIRD by calling stored procedures/routines that are stored in the database to obtain or set data.
- Translates (parses) database data and converts to proper LabVIEW data types and structures
- If changes are made to a software module and the information that is obtained or sent to NIRS, only that API requires updating.



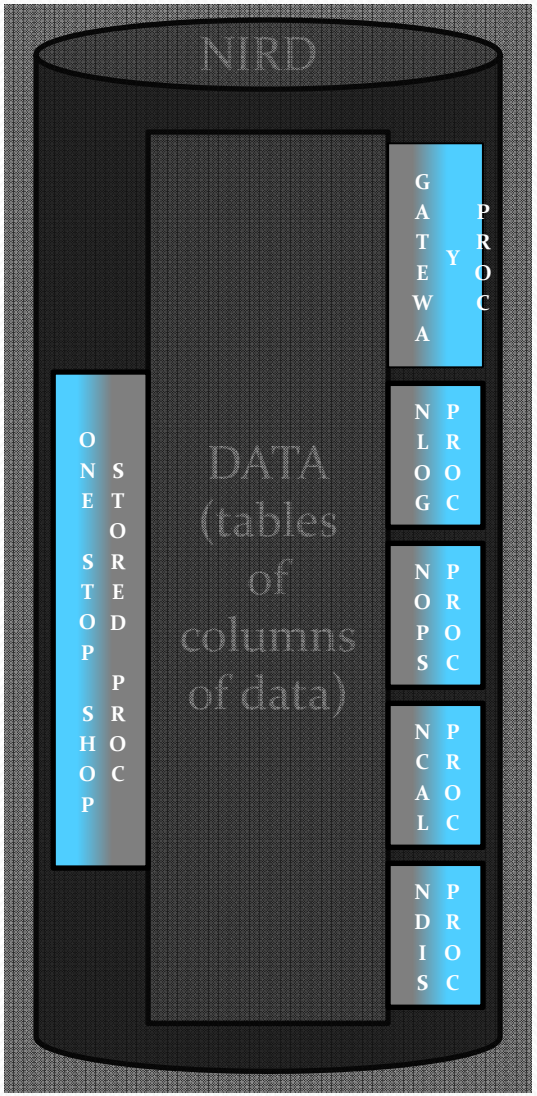
NIRD – *API to Data Interface*
Stored Procedures

Stored Procedures

- The procedures are Written in TSQL
- These procedures/routines used to get/set data in NIRD
- MSSQL allows system tables for storing of stored procedures, so they are stored in NIRD
- Each NDAS module has a set of stored procedures that enables import and export to NIRD.

Benefits of using Stored Procedures

- Keeps work of retrieving data on server side, not on application/client side
- Easier to maintain database
  - All NDAS database maintenance/upgrades can be assigned to a database administrator therefore the Centers do not have to acquire additional personnel to maintain.
  - NDAS non-LabVIEW code is in one area. Database personnel does not need training in LabVIEW.
  - Procedures/routines can be stored with backups or archives.



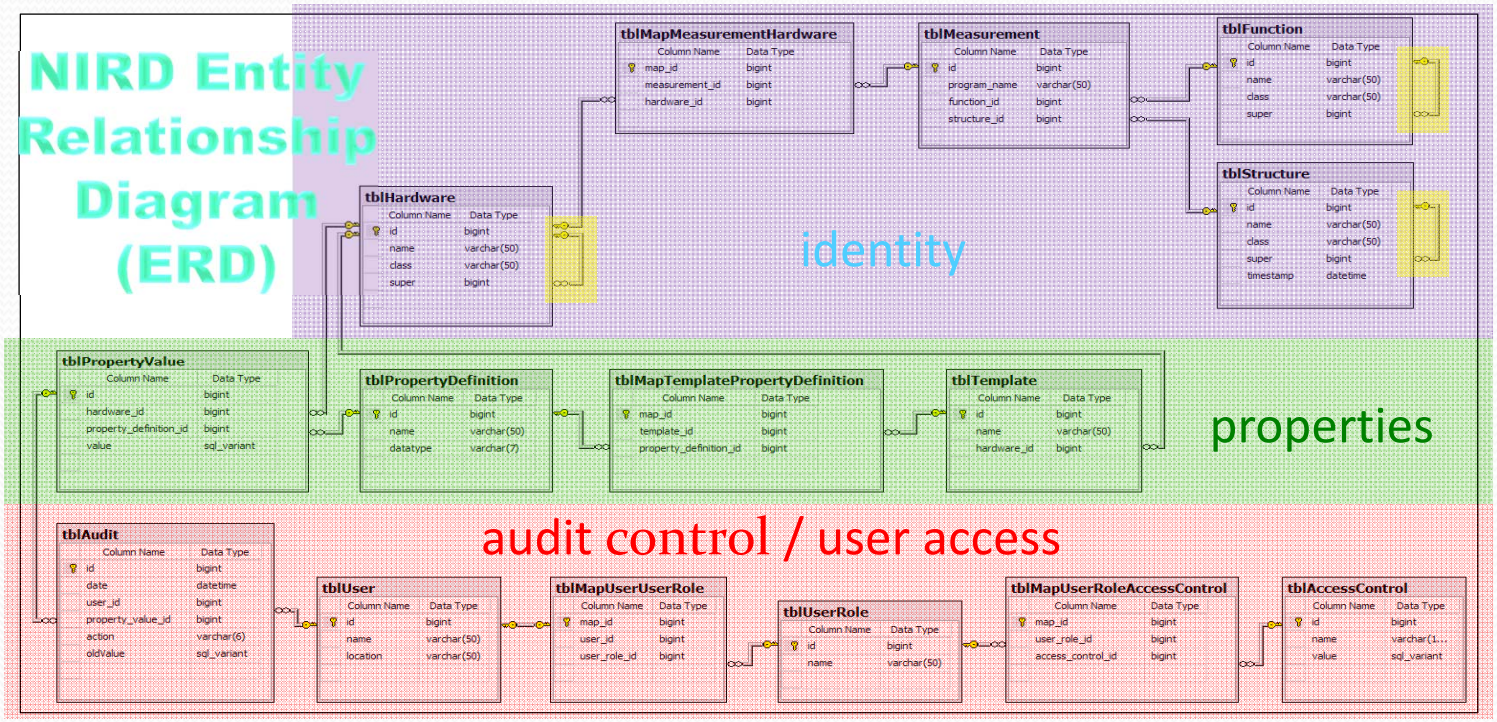
NASA Instrumentation Roadmap Database

NIRD uses a hybrid of database models

Database models:

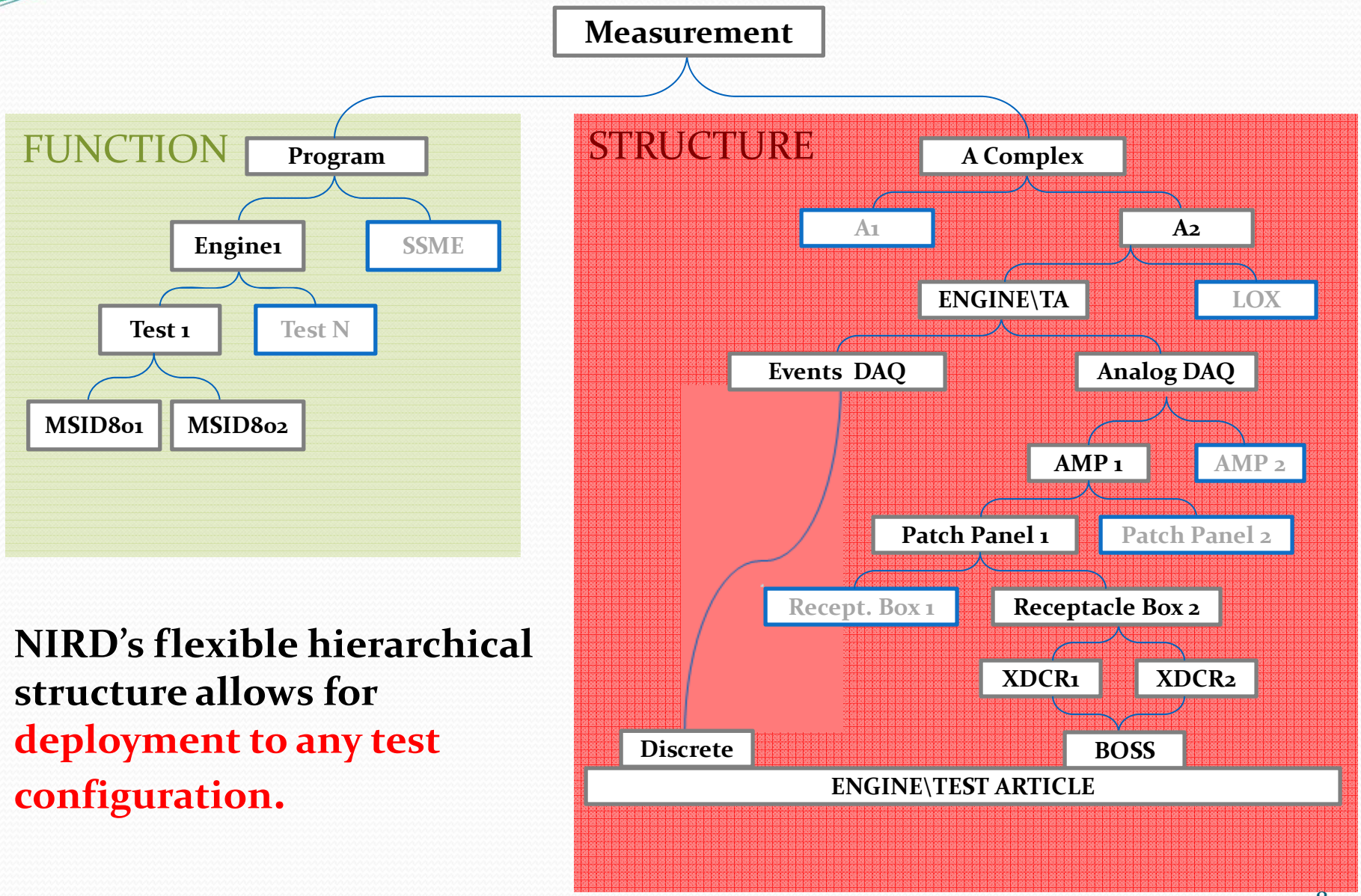
- flat: spreadsheet
- relational: easy to query
- object-relational: highly flexible
- hierarchical: preserves hierarchy in organization
- network: models decentralized nodal systems
- recursive: establishes inheritance
- object-oriented: meshes with programming languages

NASA Instrumentation Roadmap Database



NIRD

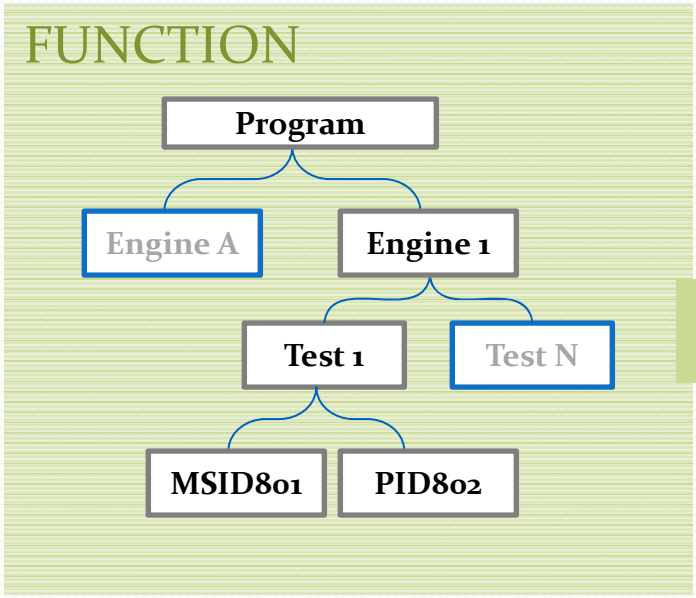
Storing the identity of system components



NIRD's flexible hierarchical structure allows for deployment to any test configuration.

NIRD

Storing the identity of system components - function

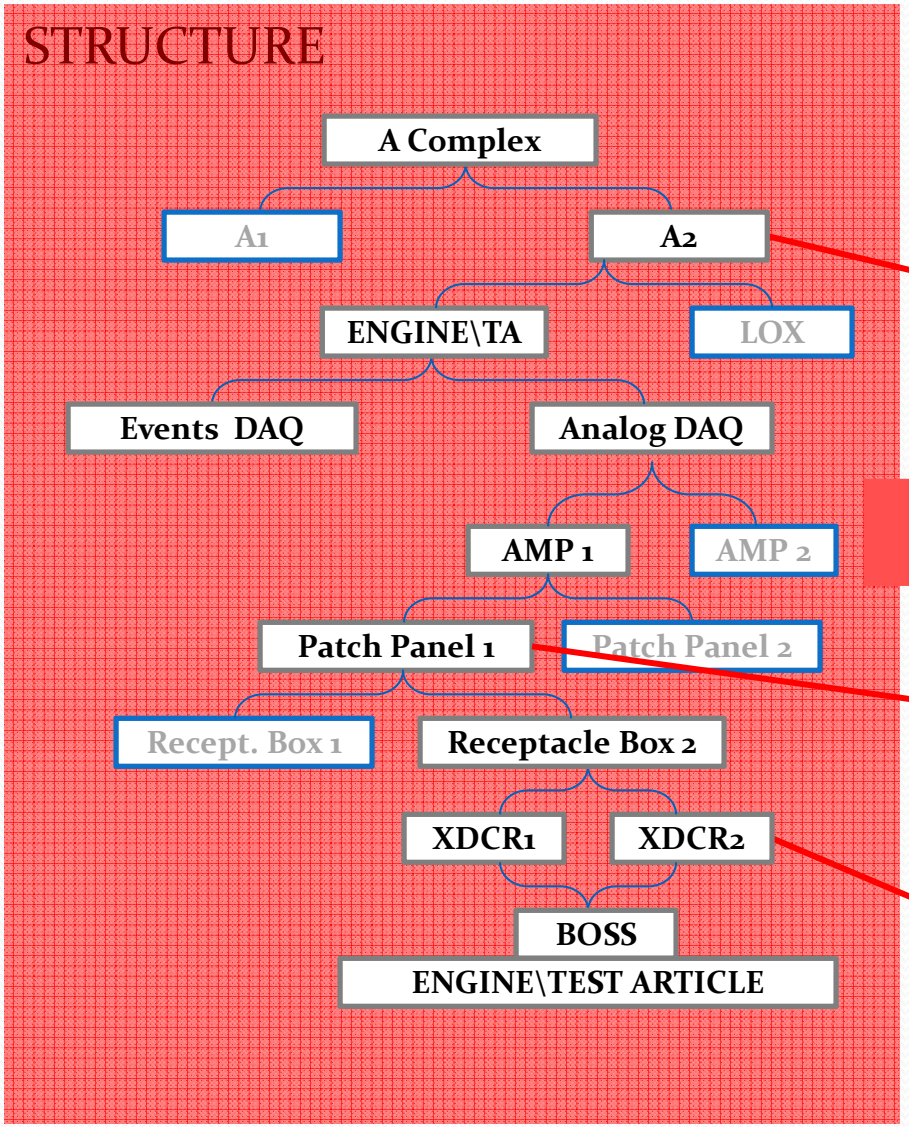


tblFunction

id	name	class	super
0	Engine 1	test program	NULL
1	Test 1	test id	0
2	MSID801	pid	1
3	Engine A	test program	NULL
4	Test N	test id	0
5	MSID801	pid	1
6	MSID802	pid	1
7	(Test E)	(test id)	(3)

NIRD

Storing the identity of system components - structure



tblStructure

id	name	class	super
0	A Complex	facility	NULL
1	A1	stand	0
2	A2	stand	0
3	engine / TA	system	2
4	LOX	system	2
5	Events DAQ	DAQ	3
6	Analog DAQ	DAQ	3
7	AMP 1	amplifier	3
8	AMP 2	amplifier	3
9	Patch Panel 1	patch	7
10	Patch Panel 2	patch	7
11	Receptacle Box 1	remote	9
12	Receptacle Box 2	remote	9
13	XDCR1	transducer	12
14	XDCR2	transducer	12

NIRD

The elegant simplicity of hierarchical metadata

tblFunction

id	name	class	super
0	Engine 1	test program	NULL
1	01A_Eng1	test id	0
2	MSID801	pid	1
3	MSID802	pid	1

tblStructure

id	name	class	super	timestamp
0	A2	facility	NULL	2005-07-24…
1	test article	system	0	2005-07-24…
2	My Daq	DAQ	1	2005-07-24…
3	My amp	amplifier	2	2008-11-13…
4	channel_0	channel	2	2008-11-13…
5	channel_1	channel	2	2008-11-13…

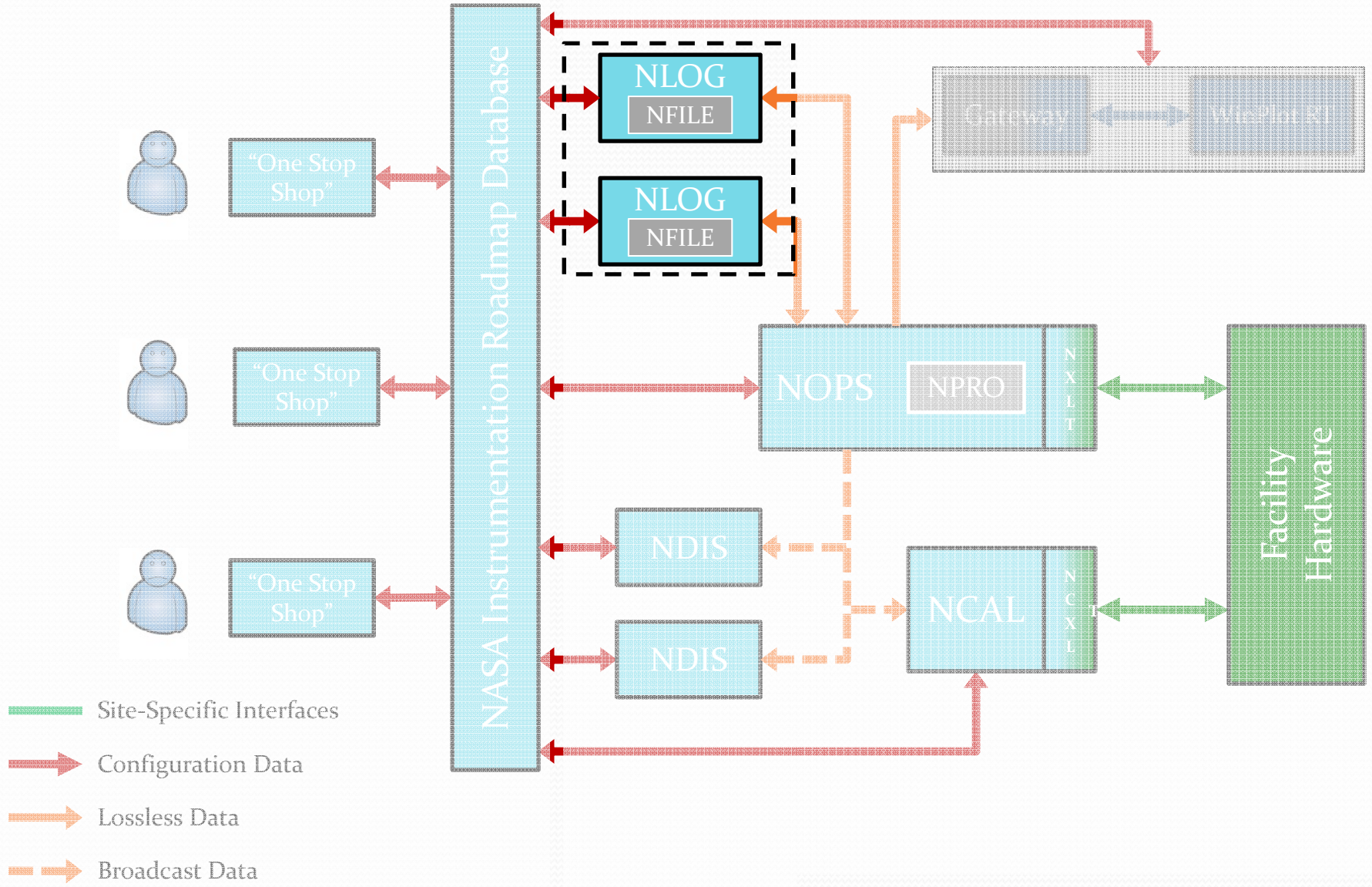
tblMeasurement

id	program_name	function_id	structure_id
0	daq_channel_a	2	4
1	daq_channel_b	3	5

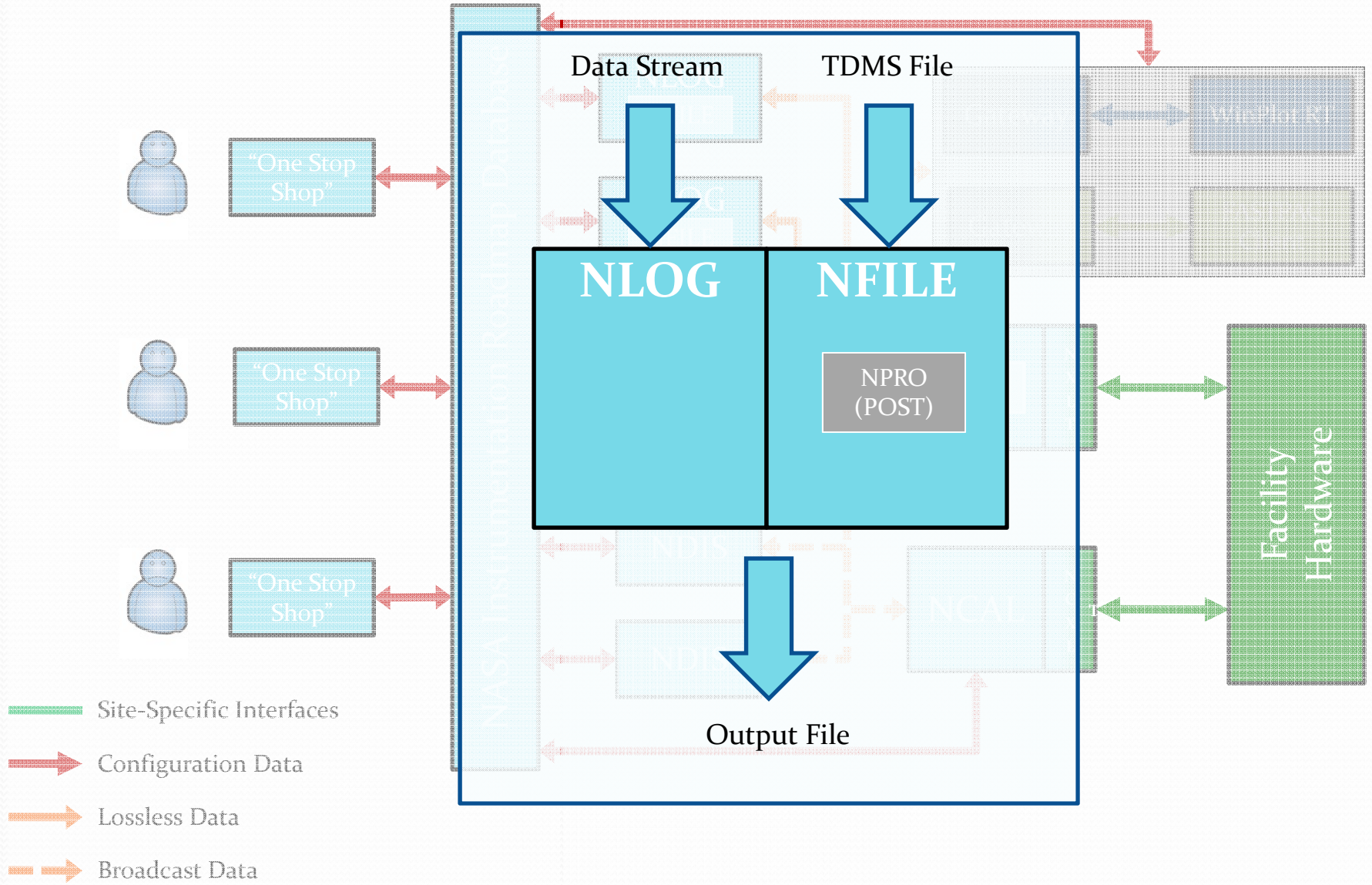
data channel b

- pid: MSID8001
- test id: 01A_Eng1
- test program: Engine 1
- amplifier: Myamp
- DAQ: MyDaq
- system: test article

NLOG-NFILE


NLOG
NFILE
NLOG
NFILE
Gateway
Web Portal
"One Stop Shop"
"One Stop Shop"
"One Stop Shop"
NASA Instrumentation Roadmap Database
NOPS
NPRO
NXI
NDIS
NDIS
NCAL
NXI
Facility Hardware
Site-Specific Interfaces
Configuration Data
Lossless Data
Broadcast Data

NLOG-NFILE
Overview



NLOG-NFILE Modules
Description

- NLOG/NFILE is an application opened by a user or other NDAS modules.
- Logs NOPS stream data into selected format. (TDMS data file, TDMS FIFO buffer directory)
- Converts TDMS files to other file formats. Standard formats in NDAS includes: Matlab, WinPlot and CSV.
- **Easily modifiable to provide customer specific file formats.**

NDAS Software Project
Conclusion

- The NDAS Software Architecture allows adaptability of the software to different hardware architectures through the use of the translation layers.
- The organization of the software into the various functional areas provides the modularity.
- The use of calibration instructions provides the capability to tailor the calibration methods to meet Center specific processes or sensor specific calibration requirements.
- The class structure employed for the engineering unit conversion software provides a supple method to add future measurement types.
- The database structure allows for the flexible hierarchical structure allows for deployment to any test configuration.
- Although the software was specifically developed for the low speed data acquisition system, the adopted architecture may support high speed data acquisition systems with minor modifications further streamlining operations at the Centers.
- NDAS is on schedule to be completed June 2012. It is currently in beta testing operating as the secondary data acquisition system supporting engine testing at SSC.

NDAS Software
Project Team

Project Role	Name	Organization
Project Manger	Mark Hughes	NASA SSC – EA52
Design Lead & Systems Engineer	Dawn Davis	NASA SSC – EA31
Software Developer (NPRO)	Wendy Holladay	NASA SSC – EA31
Software Developer (NCAL)	Michael Duncan	SSC - ARTS
Software Developer (NLOG, NFILE)	Peggi Marshall	SSC - ARTS
Software Developer (NIRD, NDIS)	Richard Franzl	SSC – Lockheed Martin
Software Developer (NXLT, NCXLT, NPRO)	Jon Morris	SSC – Lockheed Martin
Database Developer (NIRD)	Mark Turowski	NASA SSC – EA34
Project Support-Hardware Integration & Test Support	Ryan Nazaretian	USRP/Mississippi State University
Project Support-WinPlot RT	Jason Warren	USRP/Mississippi State University
Project Support-Database Architecture	Harvest Zhang	USRP/Princeton University