May 21-22

Mark Swick

Systems Architect

marks@rt.com

Your systems.
Working as one.

How Data Distribution Services (DDS) Brings Interoperability to Future Airborne Capability Environment (FACE) Certified Conformant Systems

Mark Swick, Systems Architect, RTI

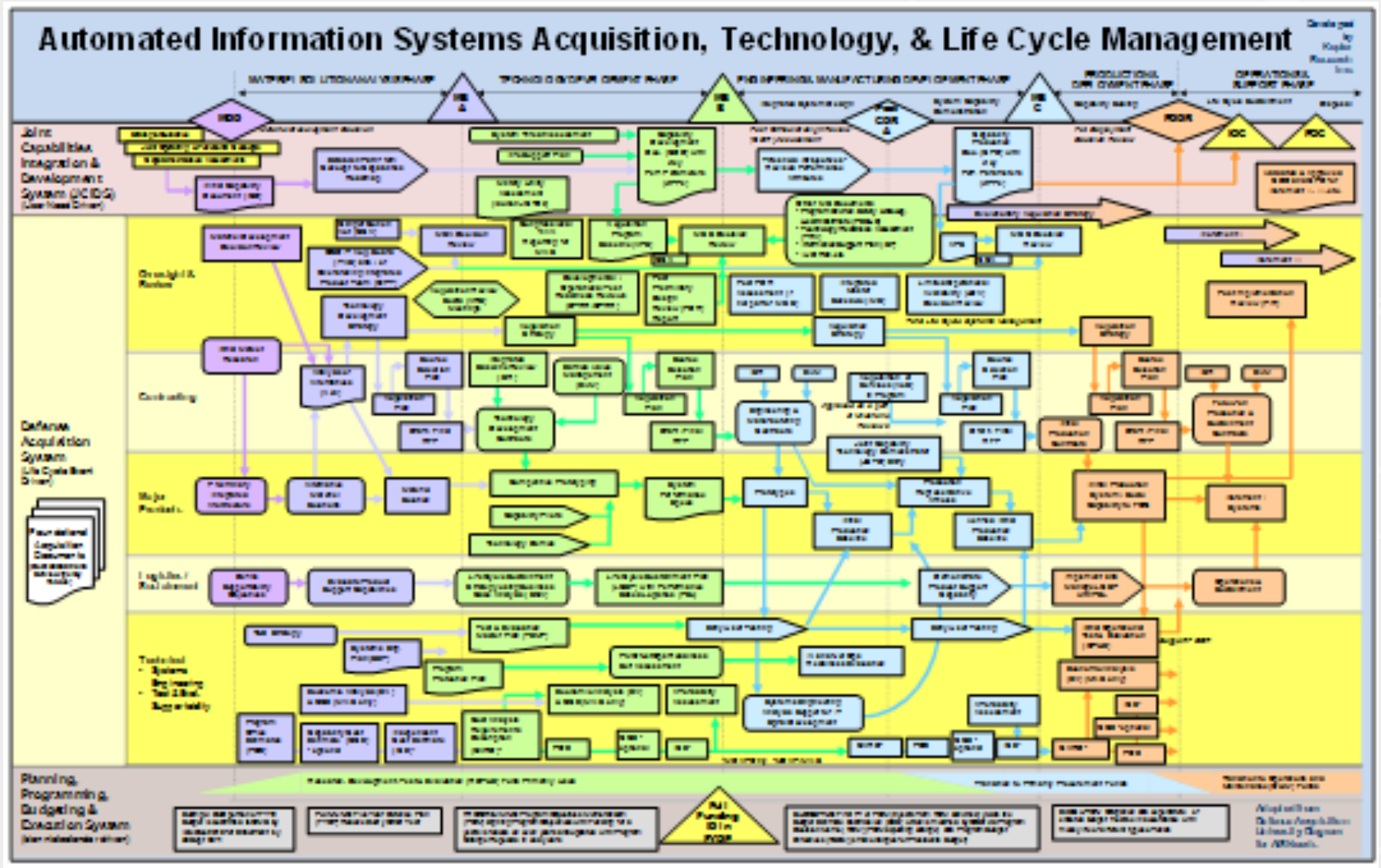
Introduction

Future Airborne Capability Environment (FACE™)

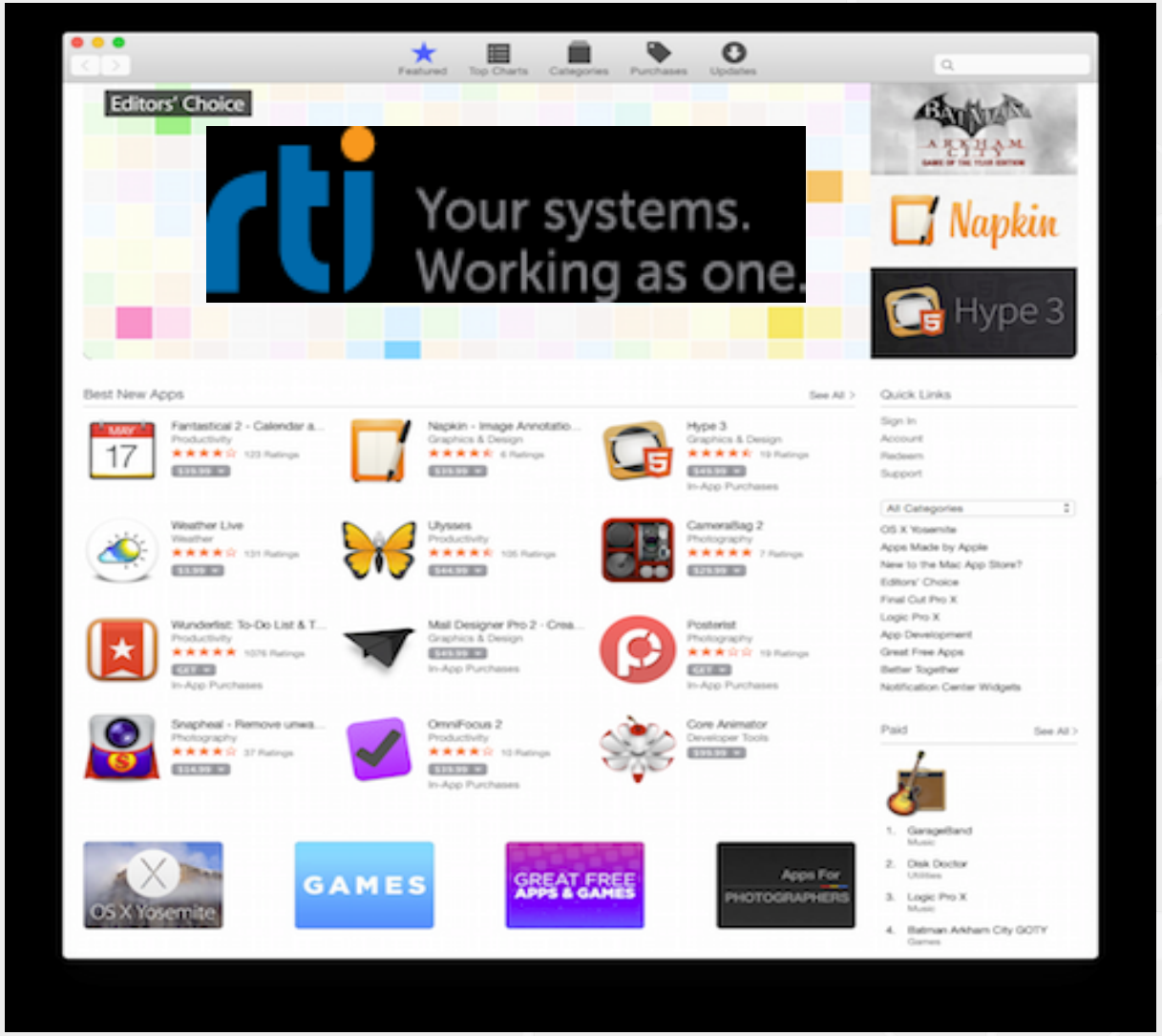
https://www.opengroup.us/face

The Government Needs a New Kind of Helicopter

How the US DoD Buys IT for a New Kind of Helicopter



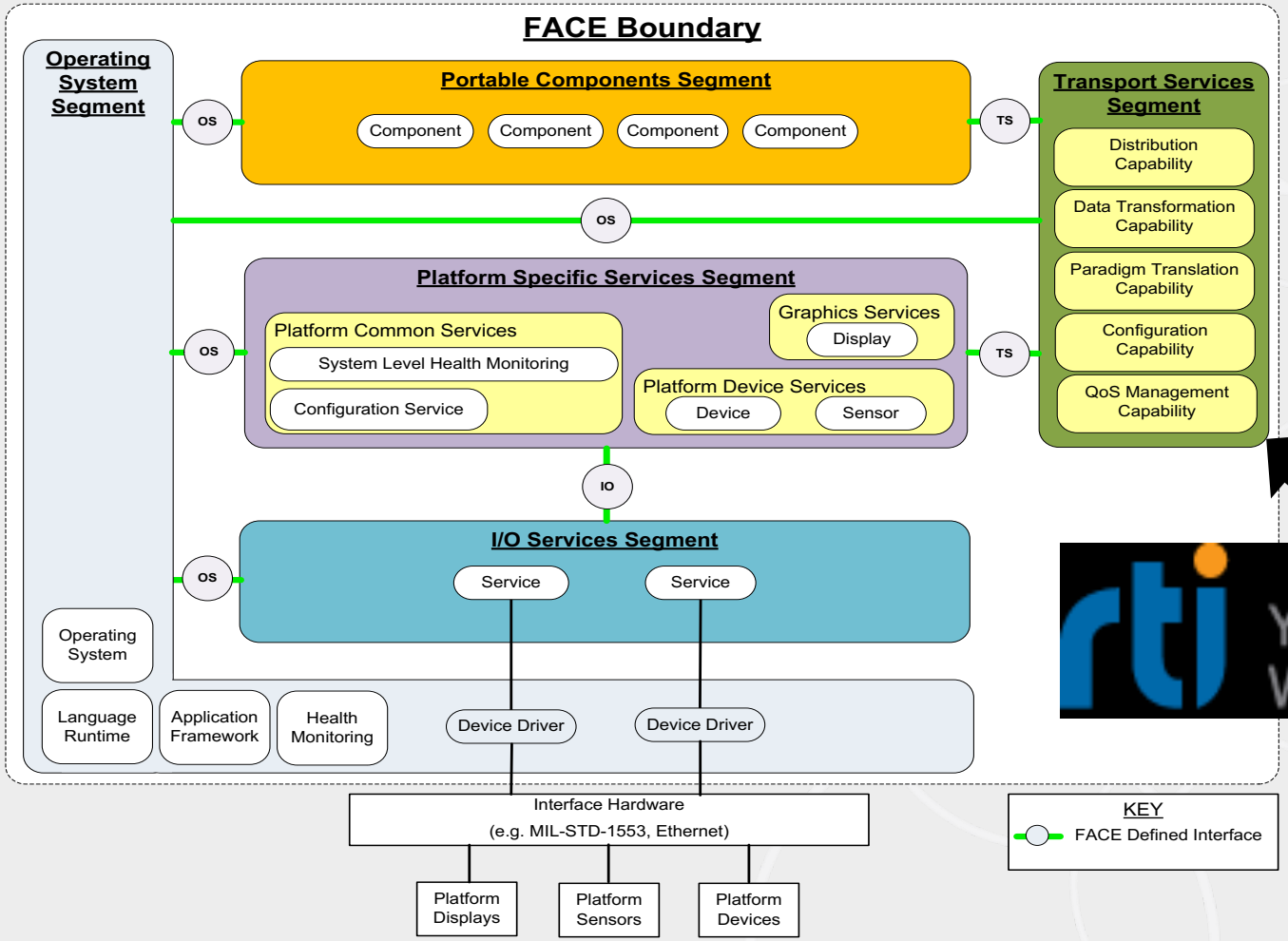
How the Government *wants* to Buy IT for the New Helicopter



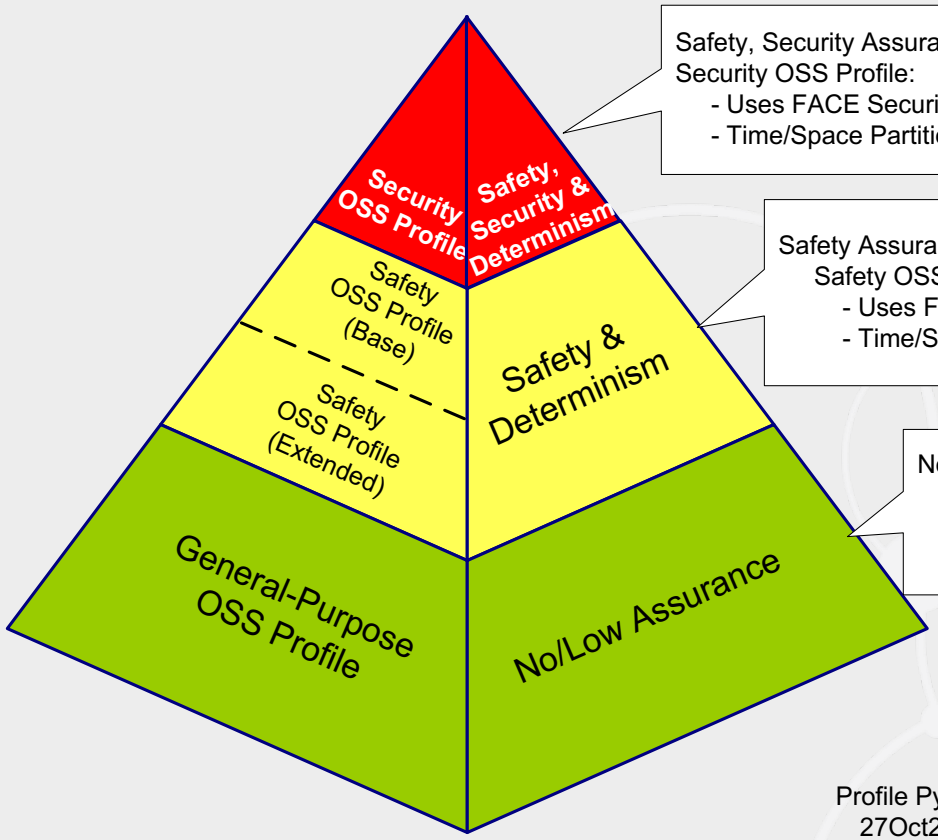
An Avionics Internet of Things

The Open Group FACE™ Consortium

FACE TSS Implementation with Connext DDS



FACE Conformance



Safety, Security Assurance & Determinism
Security OSS Profile:
- Uses FACE Security Interfaces
- Time/Space Partitioning Required

RTI Connext Micro/Cert

Safety Assurance & Determinism
Safety OSS Profile:
- Uses FACE Safety Interfaces
- Time/Space Partitioning Recommended

No/Low Assurance
General-Purpose OSS Profile:
- Uses FACE General-Purpose Interfaces
- Time/Space Partitioning Optional

RTI Connext PRO

Profile Pyramid
27Oct2015

FACE Certification Overview

Acknowledgement by the FACE Consortium that a FACE segment complies with the FACE Technical Standard

FACE Segment Development

- Implement the TSS
- Reqs, Design, Tests

FACE Verification

- 4 Verification Authorities
  - Army VA
  - Navy VA
  - Tucson Embedded
  - Certon (Cyient)

FACE Certification

- Upload info to Registry
- Receive certificate
- Only certifies FACE conformance

FACE
Future Airborne Capability Environment

THE Open GROUP

Conformance Certificate

Certificate number: **15441947**
Certificate date: **December 7, 2018**

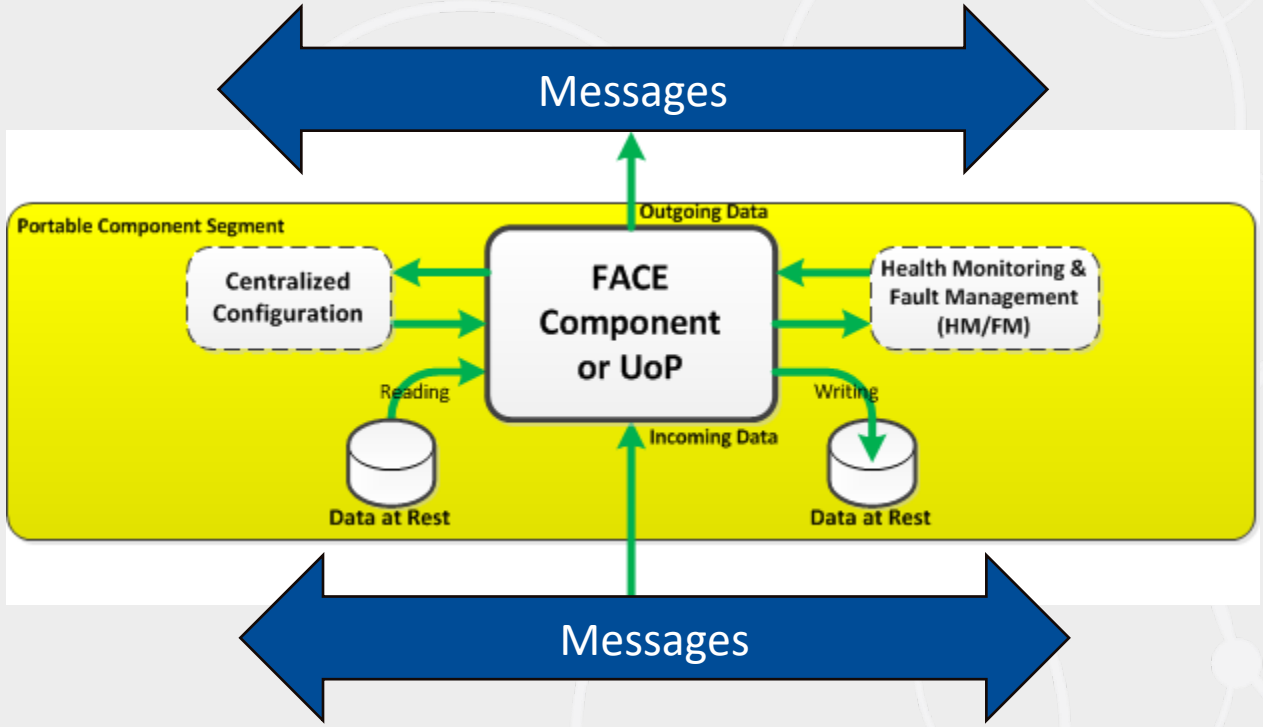
The following UoC has been certified FACE Conformant.

Software Supplier	RTI
UoC Name	RTI Connext TSS
Conformant Variants	n/a
UoC Part Number	Multiple, including 8632-20-23-00
UoC Version	2.1
FACE Segment	Transport Services
FACE Profile	Safety Base Sub-profile
FACE Technical Standard	Edition 2.1
FACE Verification Authority	TES-SAVI

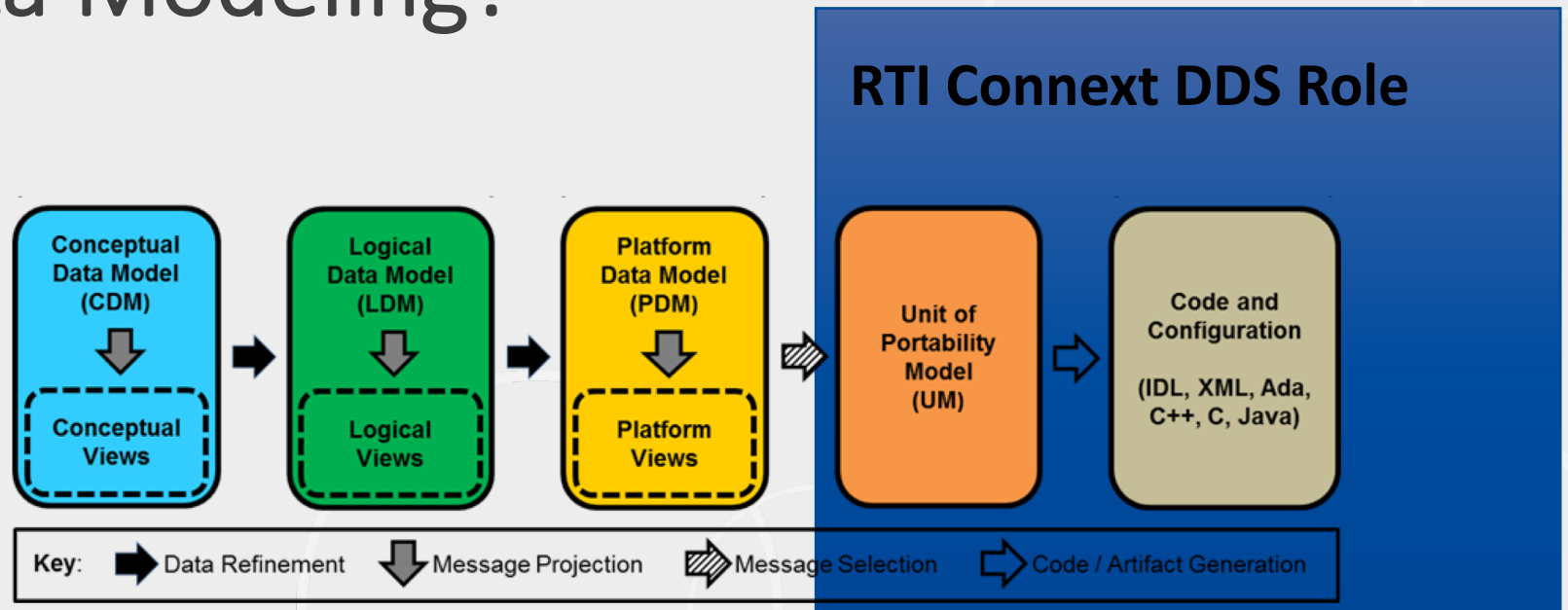
FACE Certification Authority

Data Model

“Guided by existing standards and methodologies, the Data Model Subcommittee shall focus on the definition of configuration data, messages and requirements for transient and persistent data models and meta data in the FACE architecture.”



Why Data Modeling?



Things → ***Measurements*** → ***Views*** → ***Messages***

Does Anybody Really Know How High it is?

Does Anybody Really Care?

Why RTI for FACE?

- Dramatically reduce development and deployment risk
- Proven technology in the solution space
- Tools provide ability to test, debug, monitor and record
- Life-cycle cost reduction/predictability
- Provides Interoperability

Same as for all customers

Why RTI Connext DDS for FACE Solutions?

- Legacy Messaging

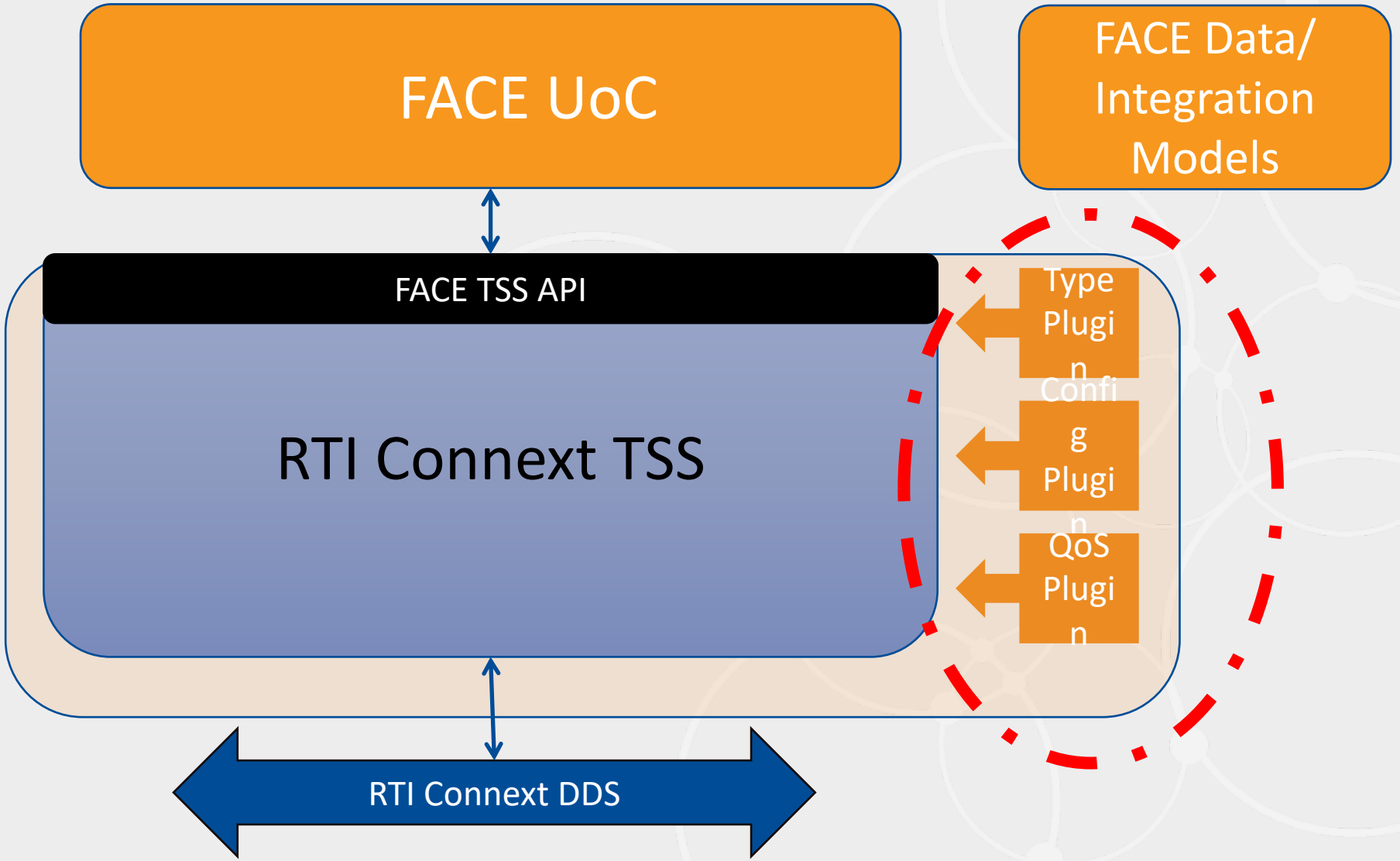
- RTI Connext DDS

Conversations about Things

Conversations about Things

- FACE Data Model implementations describe semantics of ***Things***
- They nor the FACE Technical Standard APIs describe the semantics of the ***Conversation***
- RTI Connext DDS fills the void

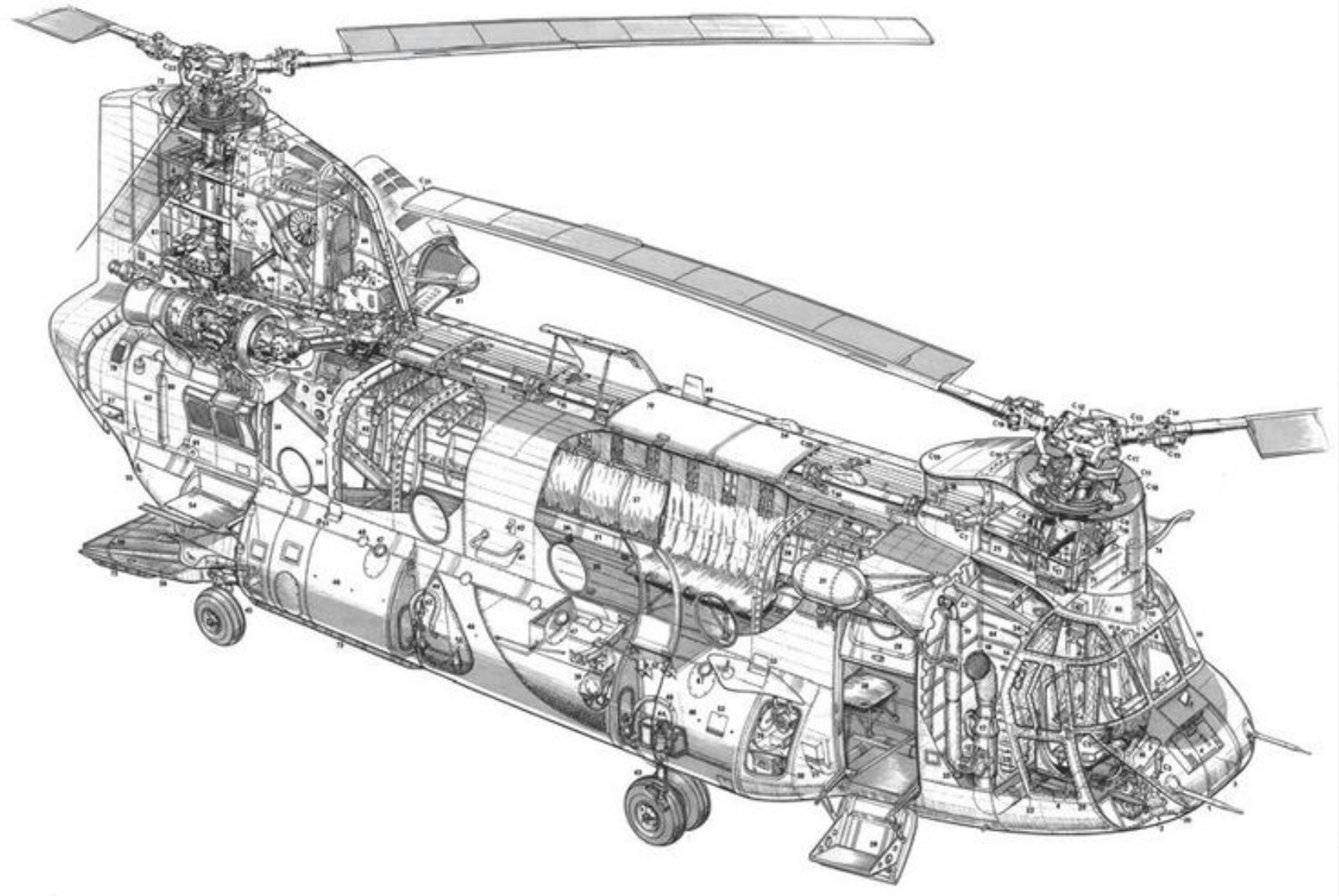
Underspecified Areas



How Many Conversations?

- RTI Connext DDS supports **thousands** of Conversations
- For **hundreds** of participants
- About **hundreds of thousands** of Things
- With **Governance and Security** for each Conversation
- With every Thing **identified** and managed

This Many Conversations



How do we do it?

- Avionics requires the same features as other business verticals
- ***Same Conversations***, about different ***Things***
- Only the scale and determinism requirements differ for avionics

One Conversation at a time

FACE Technical Interchange Meeting, September 2018 :

Rapid Integration Framework

https://www.opengroup.org/sites/default/files/images/Rapid_Integration_Framework_Presentation_AMRDEC.pdf

Crew Mission Station

- The Crew Mission Station (CMS) was initiated by the UH-60 Program Office to:
  - Add Situational Awareness for the Crew Chief on the UH-60 Blackhawk
  - Provide a means to deploy new capabilities as rapidly as possible
  - Produce a government owned open systems architecture
  - Promote independence for the system integrator

Rapid Integration Framework

- Based on the Objective Architecture for CMS
- Defines an Open Systems Architecture for a Core System
- Includes Hosted Capability definitions as extensions to the architecture
- A Rapid Integration Platform:
  - Includes the Systems Architecture as well as the resulting system for a specific implementation of the RIF
    - CMS is a Rapid Integration Platform

Demonstration Objectives

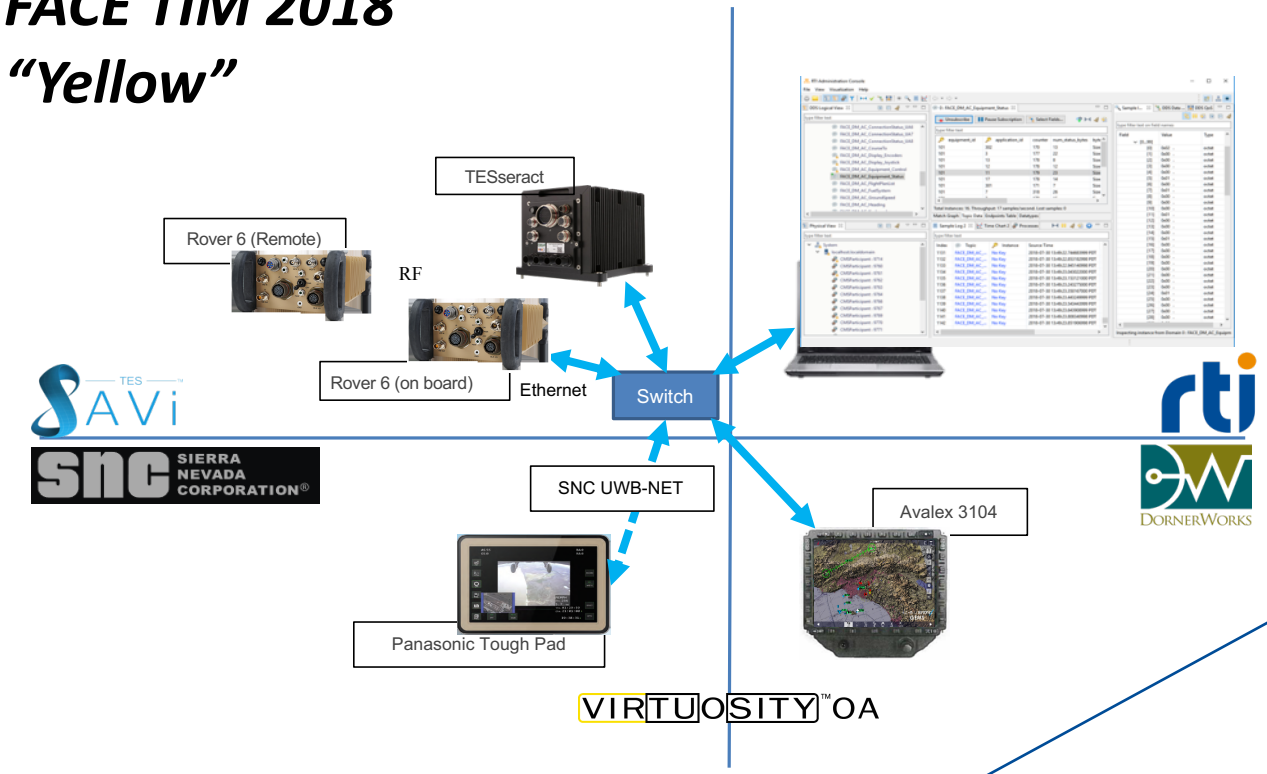
- Rapidly integrate new hardware and software to support technology refresh, i.e. replace suppliers' products
- Provide government managed architecture for Industry to extend and show integration and interoperability of new/innovative capabilities
- Enable Industry to port and reuse software and artifacts across platforms
- Enable learning and outreach for PEO AVN

This architecture and these components are not bound to any single vendor or product.

Demonstrations

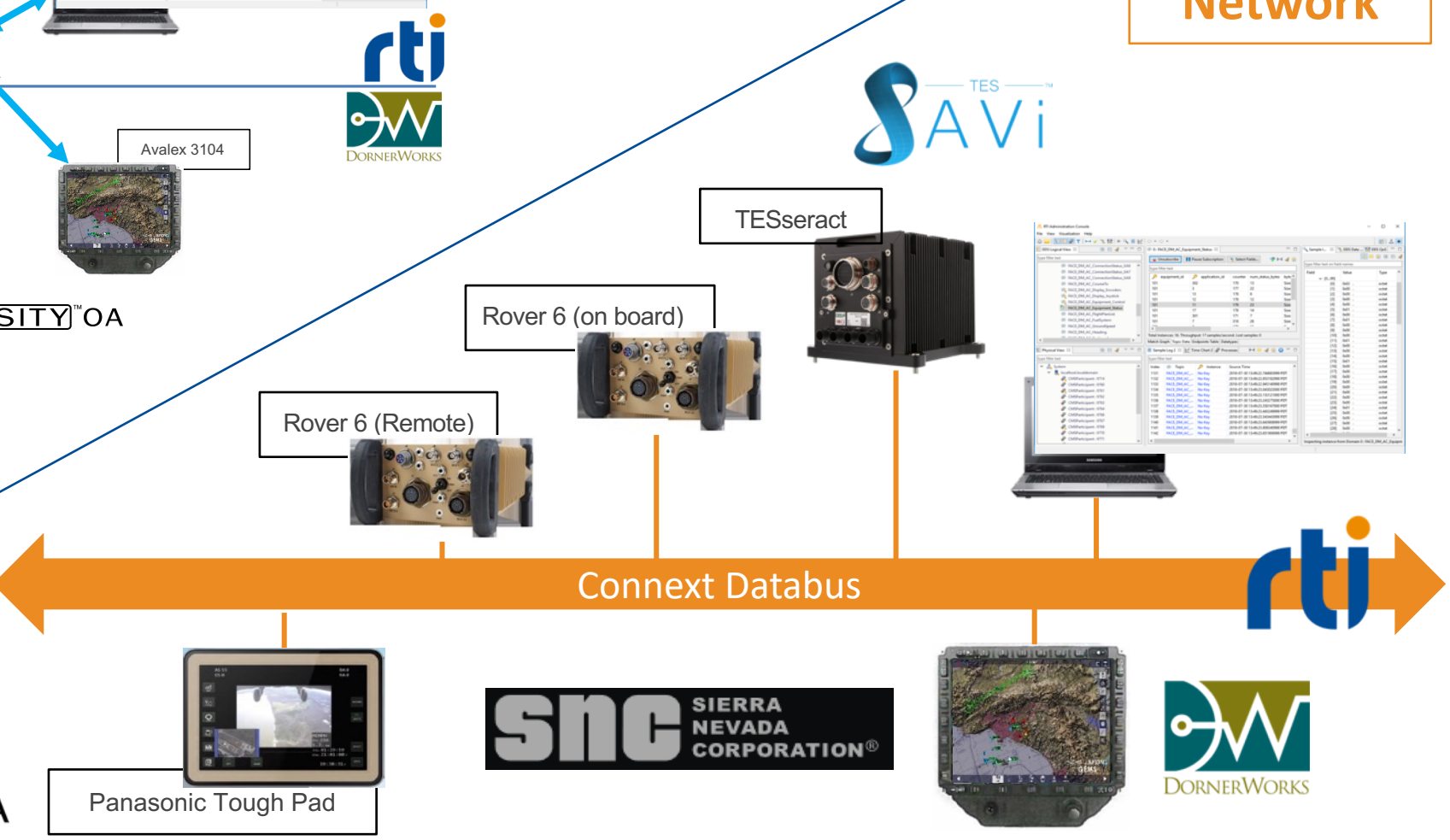
- Rapid Integration
  - Envisioned 6 months prior
  - Kicked off 3 months prior
  - 20 participating organizations
- Demonstrations include 5 alternatives : **White, Green, Yellow, Blue and Red**)
  - Red demo did not use DDS in the TSS implementation
  - software redeployed to other operating systems and hardware • alternative hardware
  - alternative software
  - additional capabilities

FACE TIM 2018
"Yellow"



Physical Network

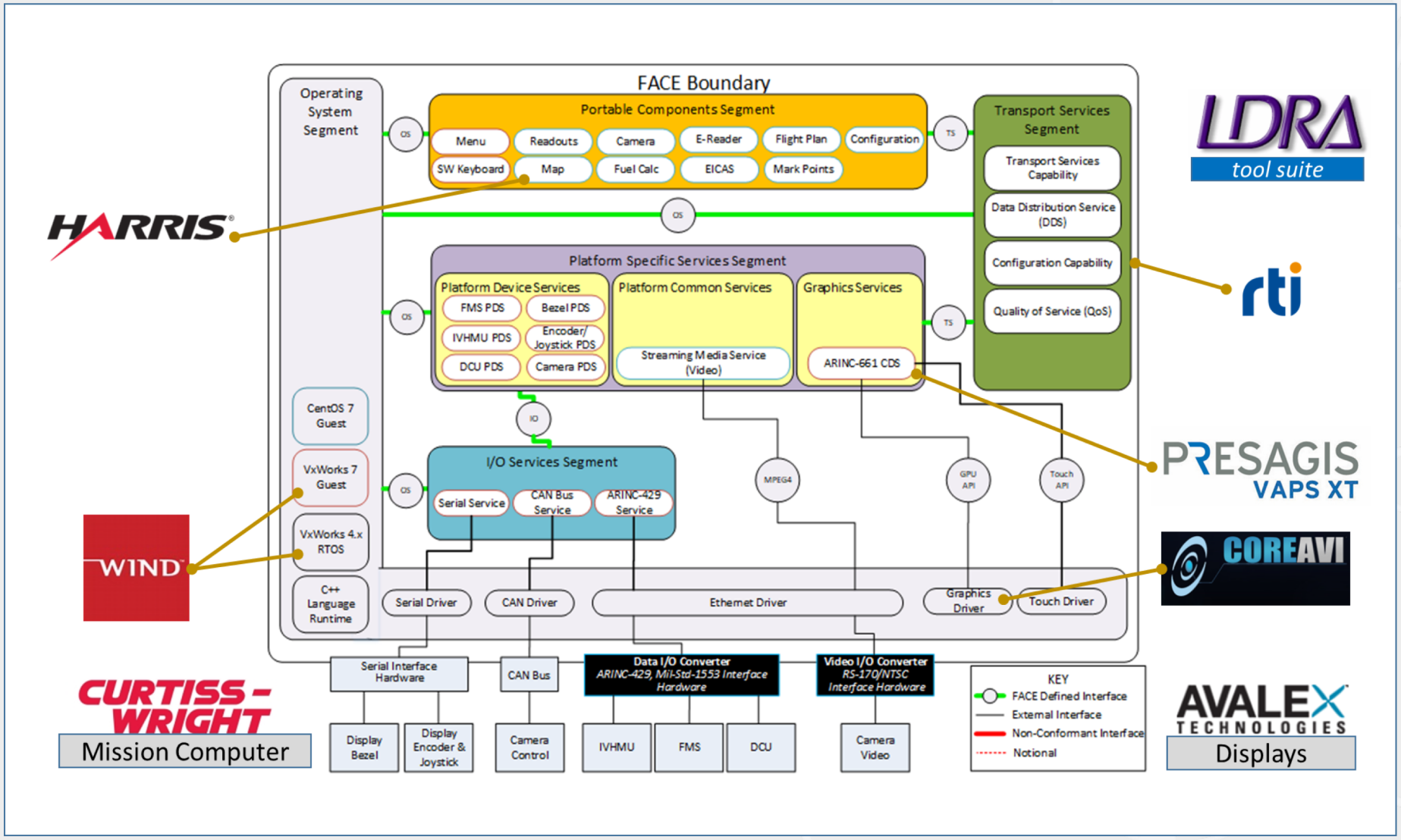
Logical Network



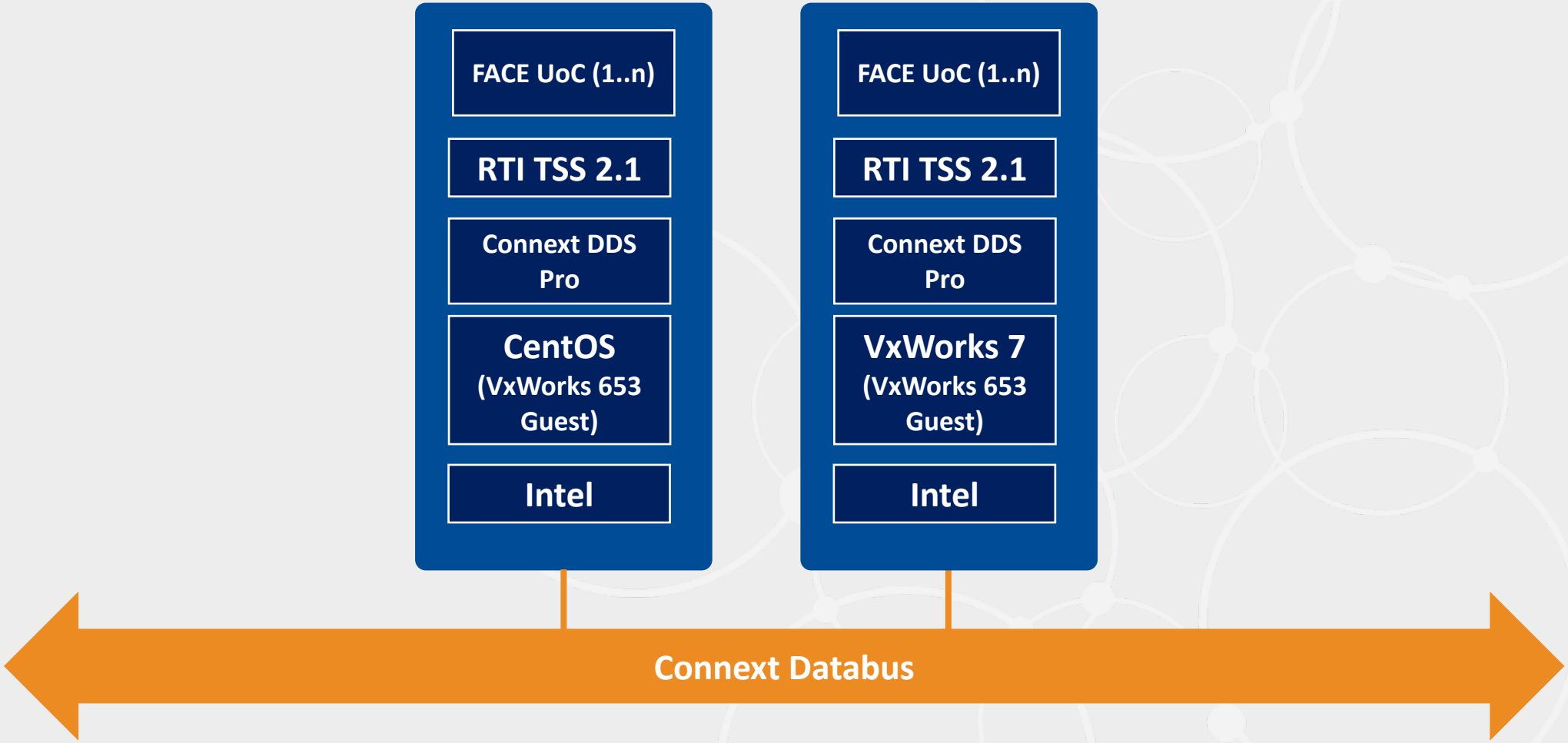
Realistic Complexity

- Numbers Approximate
  - 20 Processes
  - 20 Domain Participants
  - Ethernet networks
  - 70 Topics
  - 100 DataWriters
  - 130 DataReaders

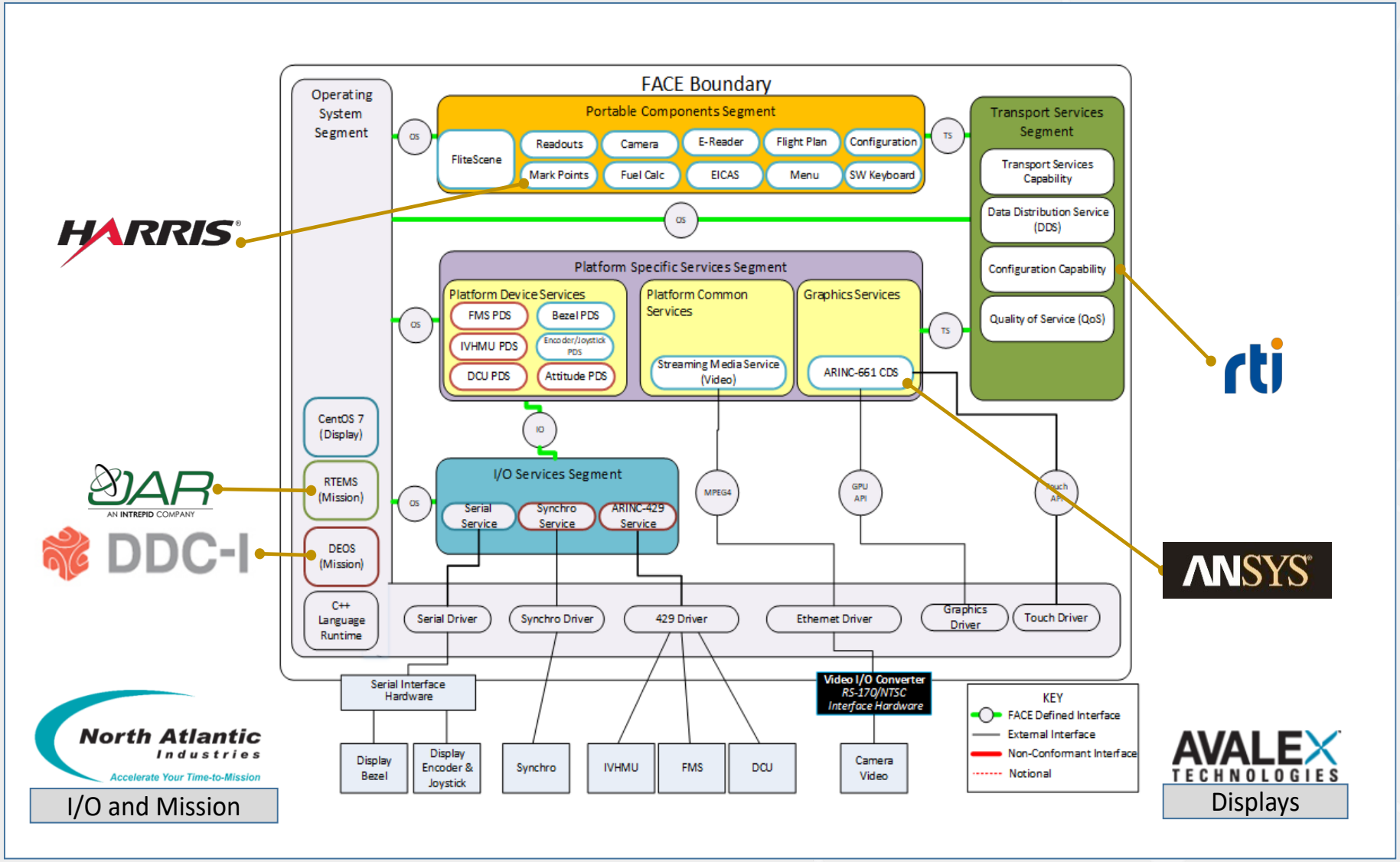
White FACE View



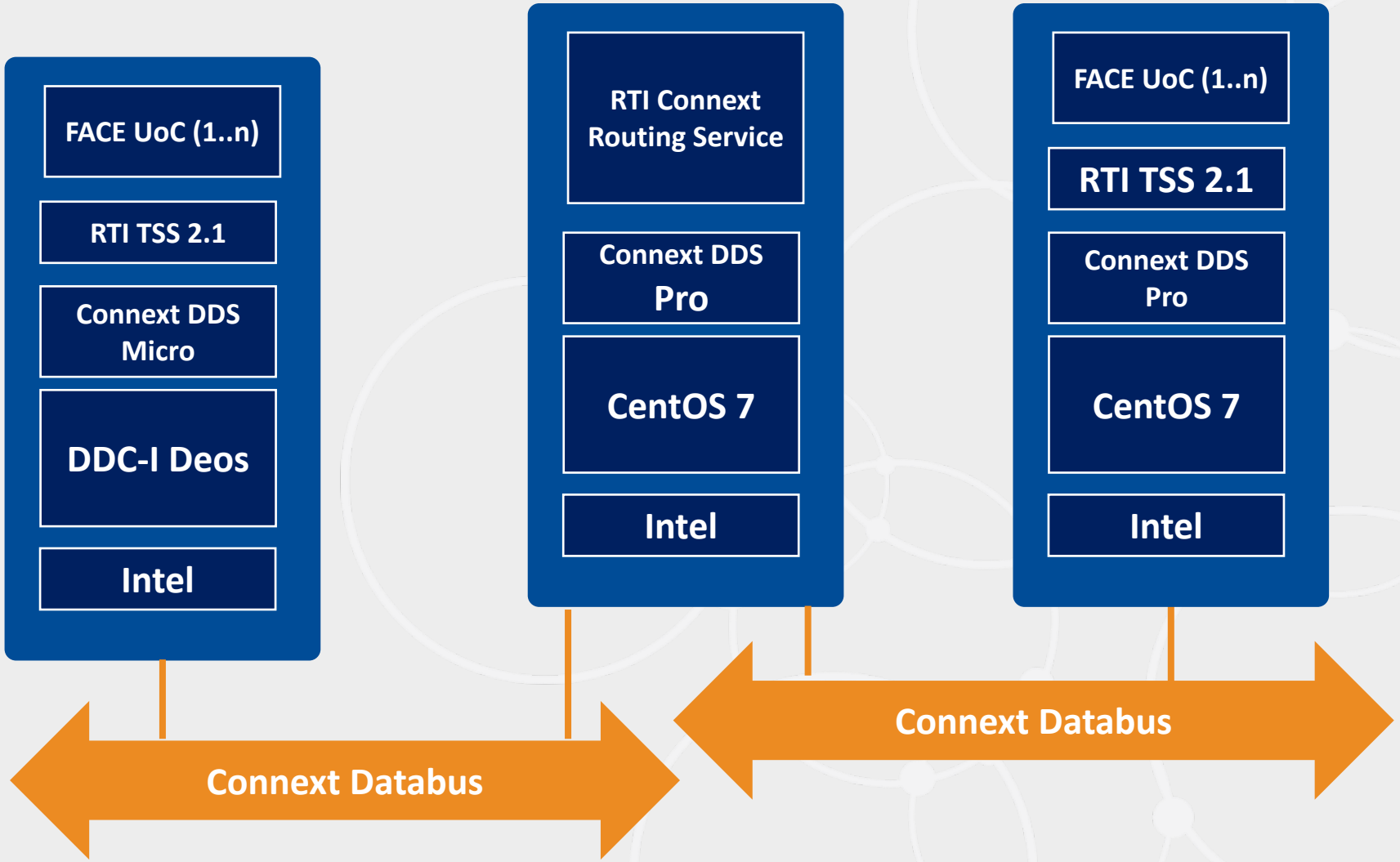
“White” Demonstration Stack



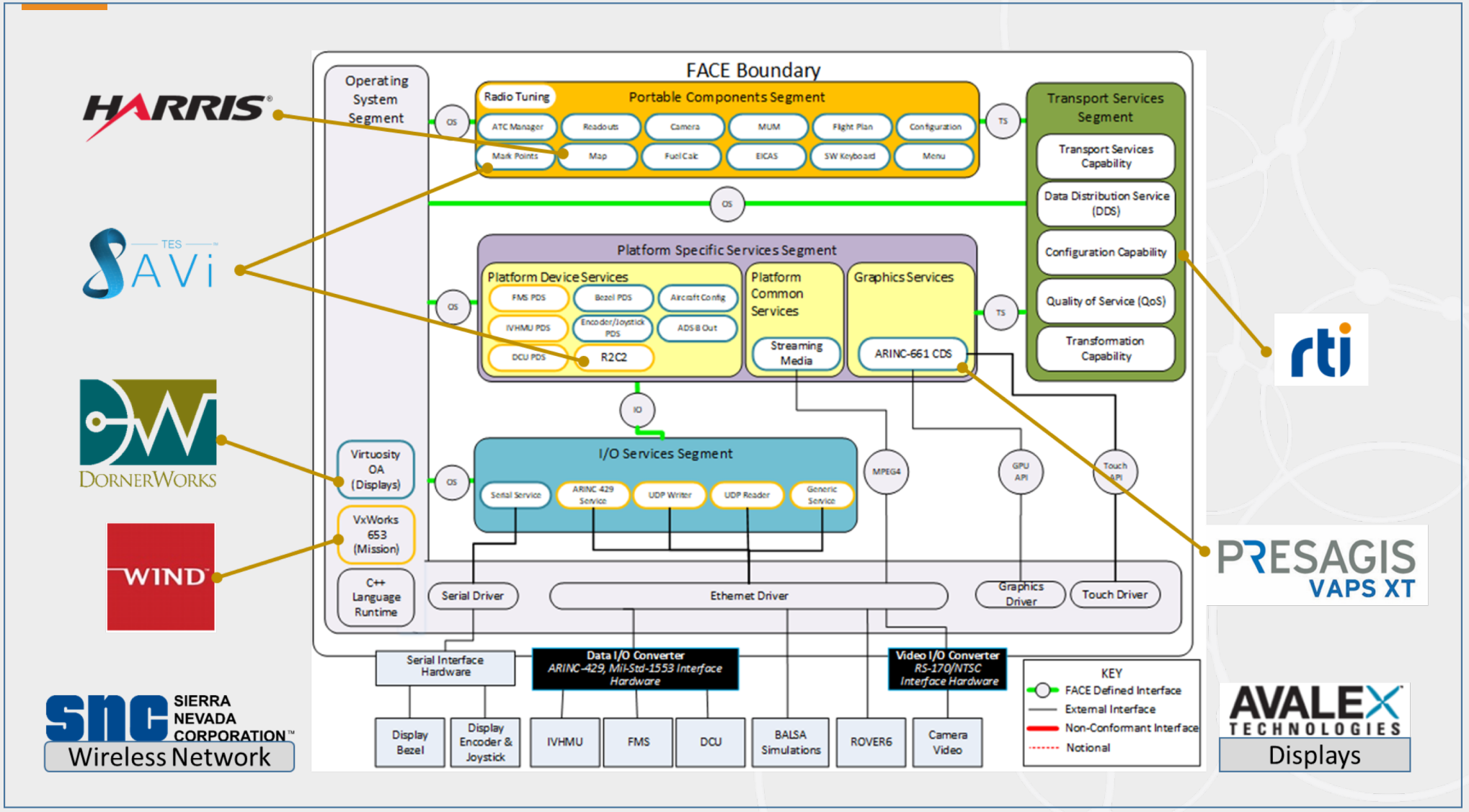
Green FACE View



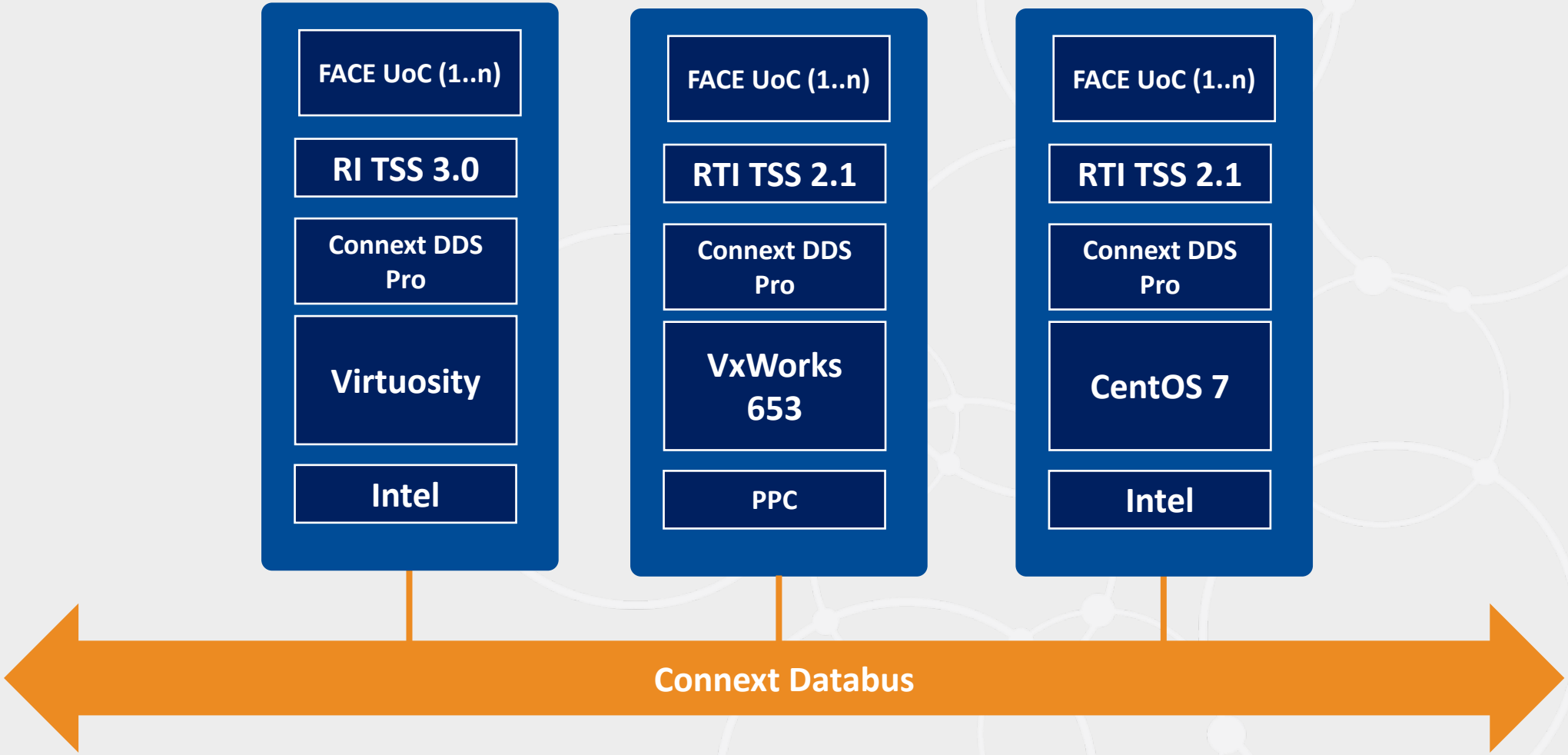
“Green” Demonstration Stack



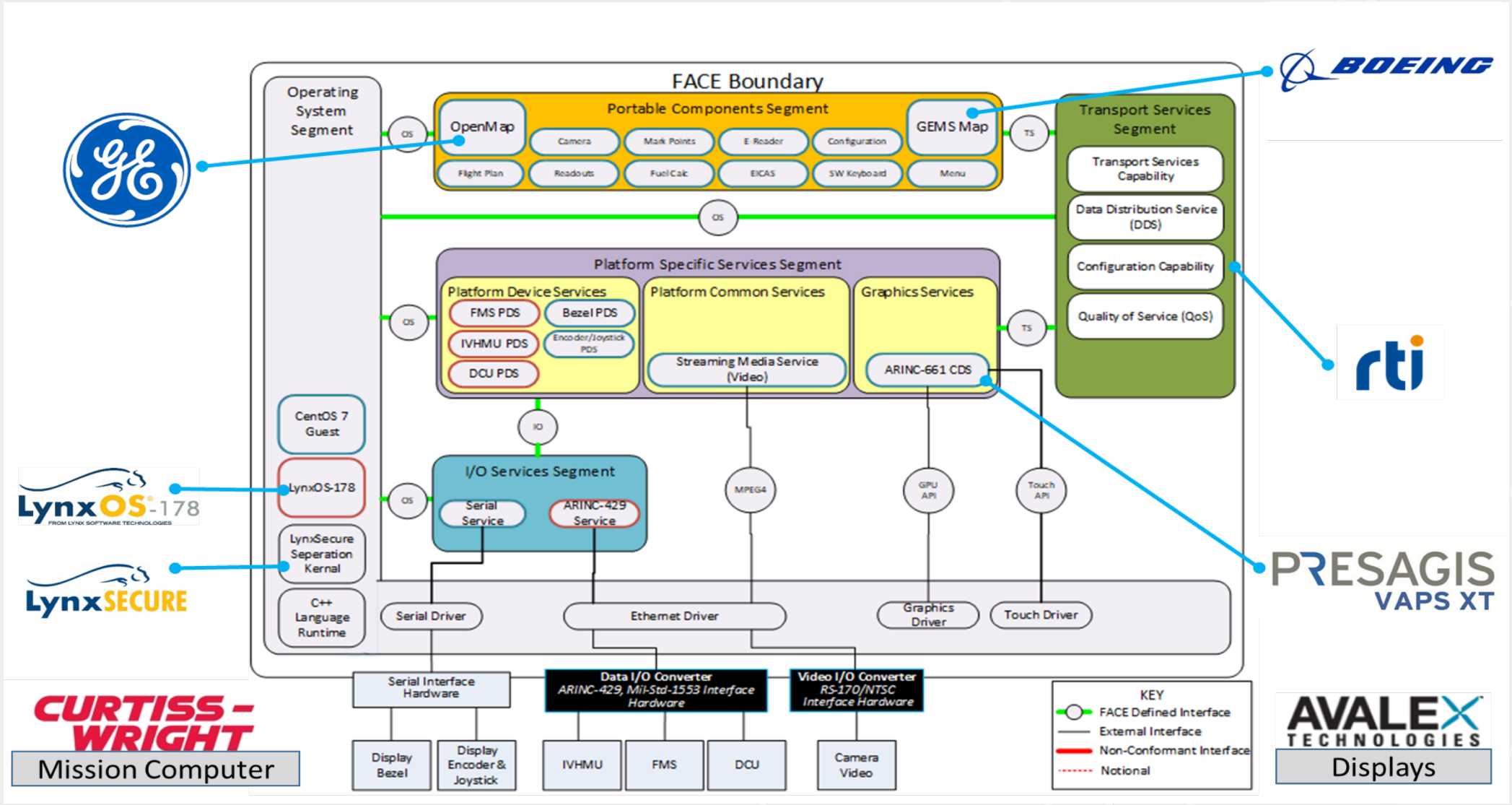
Yellow FACE View



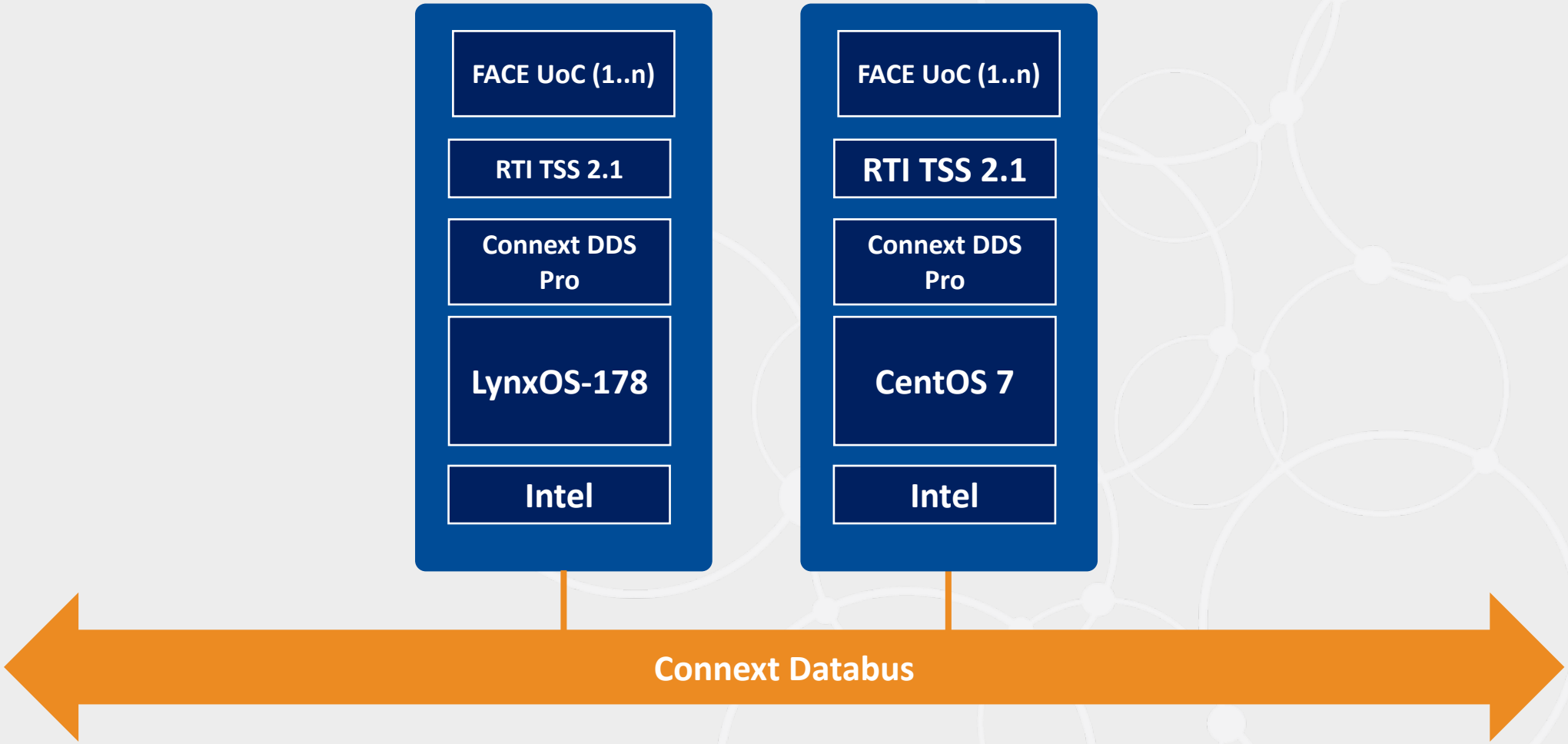
“Yellow” Demonstration Stack



Blue FACE View



“Blue” Demonstration Stack



Prerequisites for Rapid Integration

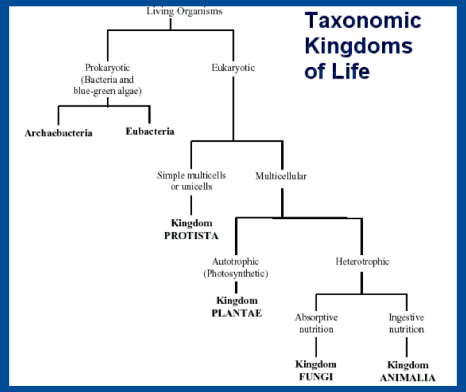
Meeting the Need

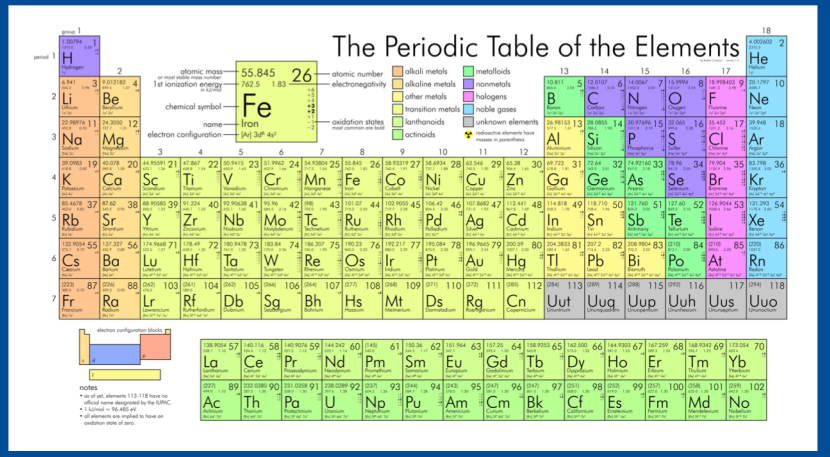
- Interoperable Data Model
- Portable Software Components (APIs)
- Communication Interoperability between OS/HW platforms
- Configurable communication Quality of Service (behavior)
- Availability of OS/HW platforms
- Backwards Compatibility
- Integration and Analysis tools

Model

A model is anything used in any way to represent something else

Data Model





A data model is a representation that describes the data about the things that exist in your domain

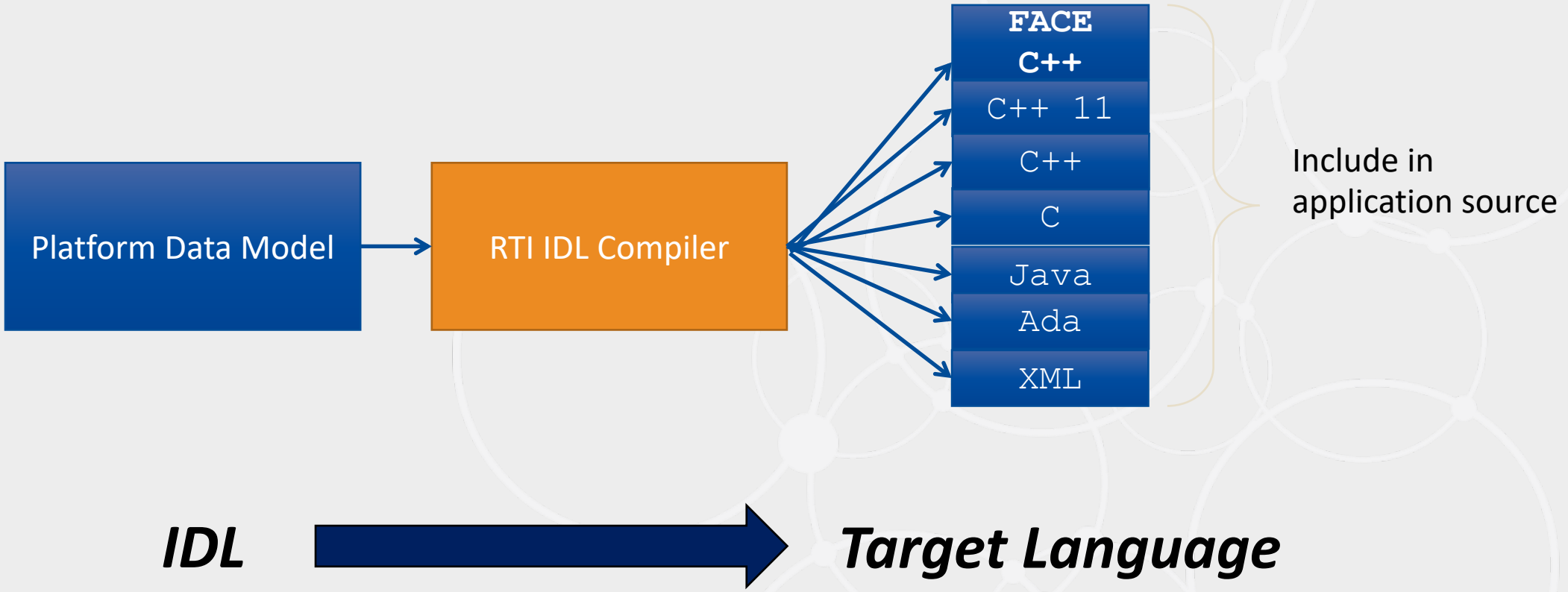
Model and Implementation

- Model provides the Context and Semantics
  - Containment and relationships
  - May not necessarily be in the messages (data at rest)
- Messages can be compact
  - Use the model for context
  - 'Know' the association between a command and a status
- Using machine readable context
  - Can generate the system appropriate mediation

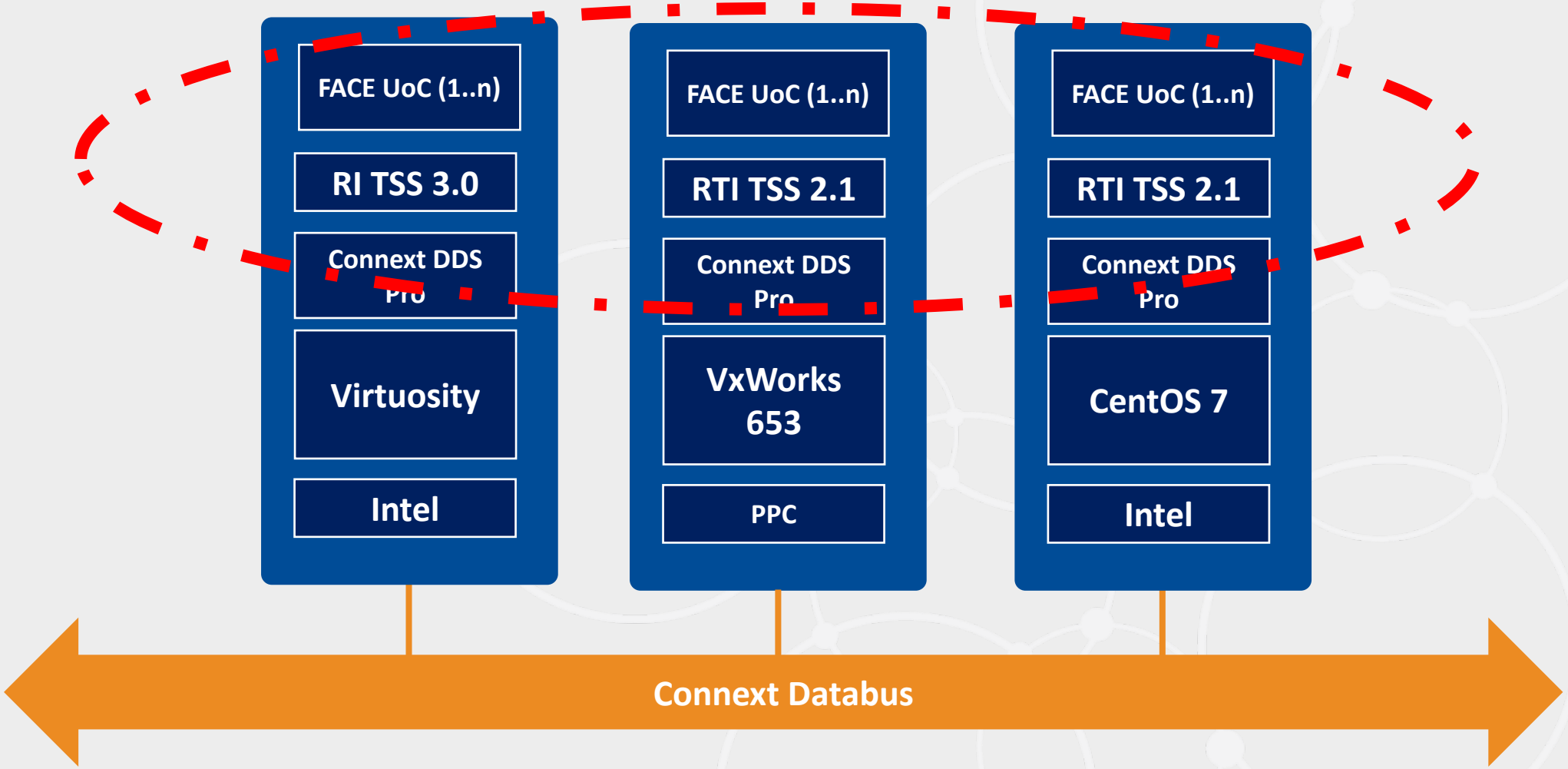
DDS Natively Supports Interoperable Data Models

- DDS messages are strongly typed
- OMG IDL basis for native DDS Data Model schema
  - XML, XSD, also supported
  - Apps use target code generated by RTI's IDL compiler
- DDS natively understands data
  - Type safety
  - Heterogeneous interoperability (languages, CPUs)
  - Wire efficiency (minimizes metadata)
  - Enables middleware-level filtering (including at source)
  - Eases integration (explicit interfaces)

Code Generation



“Yellow” Demonstration Stack



APIs and Portability

TSS Type Specific API

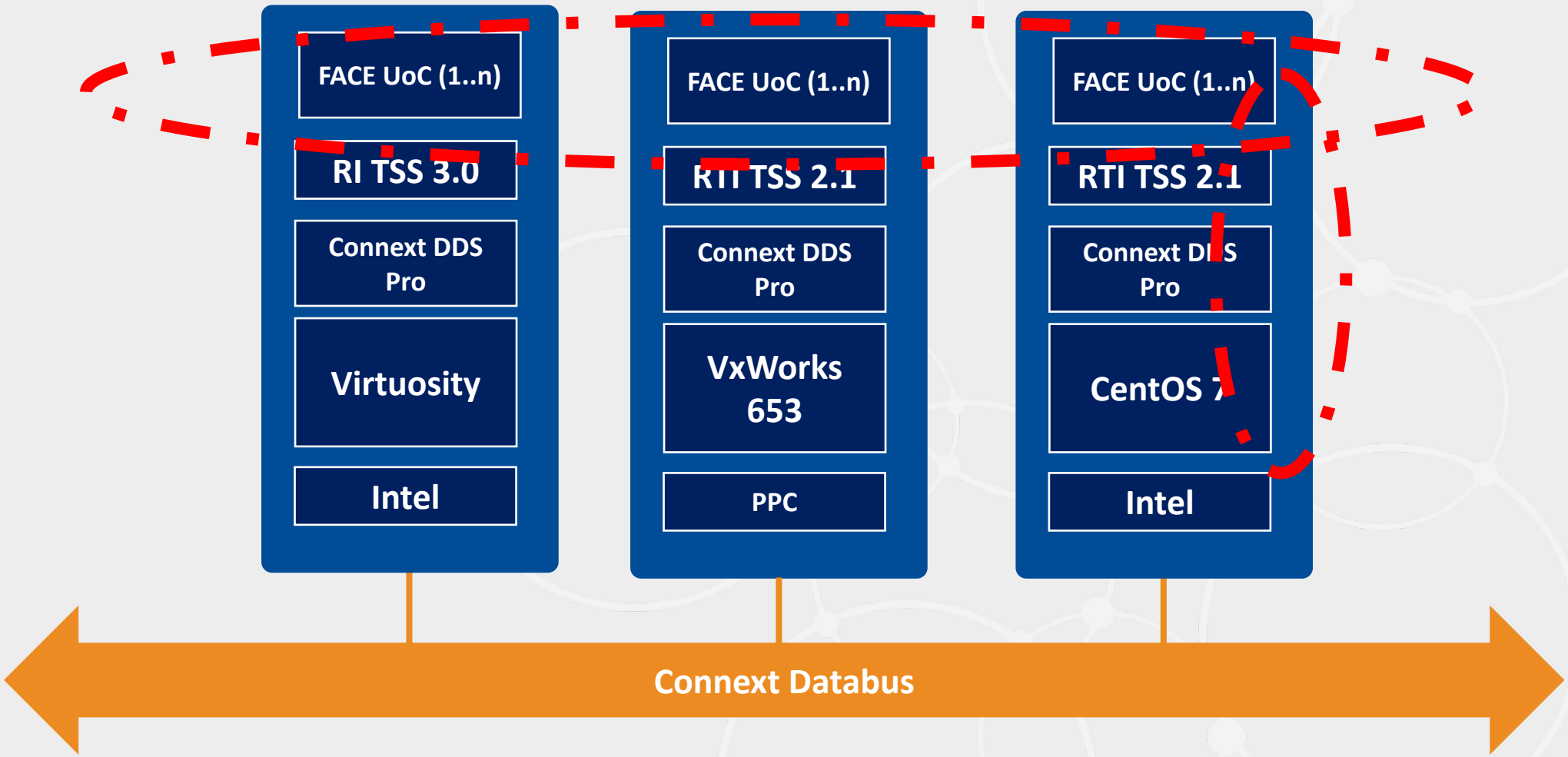
- C++ API
  - **FACE::TS::Initialize(…)**
  - **FACE::TS::Create_Connection(…)**
  - **FACE::TS::Receive_Message(…)**
  - **FACE::TS::Send_Message(…)**
  - **FACE::TS::Register_Callback(…)**
  - **FACE::TS::Unregister_Callback(…)**
  - **FACE::TS::Get_Connection_Parameters(…)**
  - **FACE::TS::Destroy_Connection(…)**

  - **FACE::Read_Callback::send_event(…)**

TSS Implementation

- FACE “connections” map to DDS Topics
  - If direction is “source:” DataWriter
  - If direction is “destination:” DataReader
  - If direction is “both:” DataWriter & DataReader
- One Domain Participant per Domain
  - Participants are reused for Publishers, Subscribers
- RTI TSS Implementation utilizes “plugins”
  - “Stub” Code generated
  - Customization completed by integrator

“Yellow” Demonstration Stack



Enabling RTI Connext Features

- Three of the stacks never before used a TSS
- In order to meet 3-week development cycle, RTI had to be early in providing TSS functionality
- FACE Portability TSS API allowed parallel development
- Portability of RTI Connext DDS TSS and the DDS standard was essential to quick porting of new platforms

RTI Connext DDS Portability

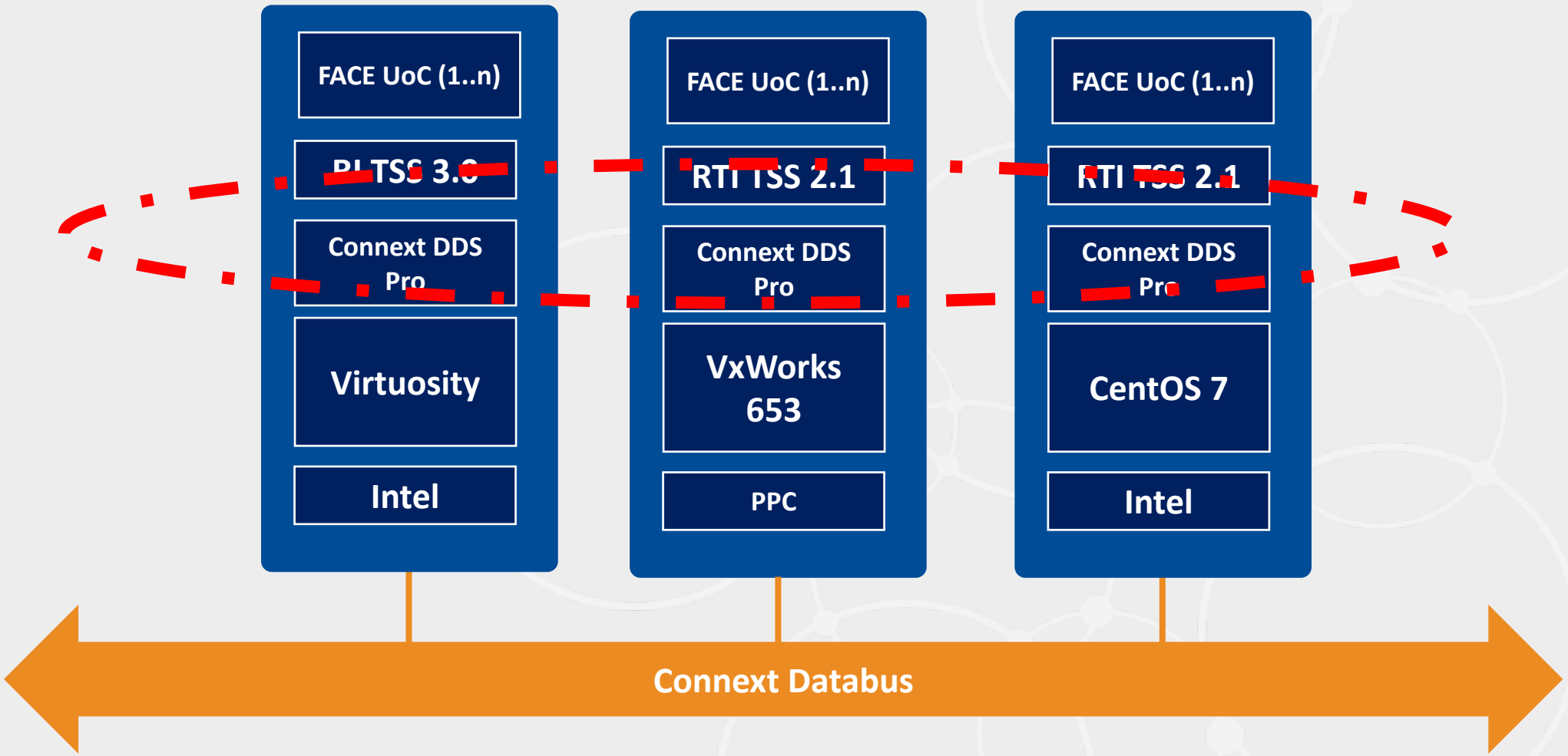
- Essential to Rapid Development of TSS dependent Units of Conformance
  - RTI Connext DDS Professional runs on dozens of platforms
  - Used existing libraries for LynxOS, VxWorks, and CentOS
  - RTI Connext DDS Professional ported to Virtuosity
  - RTI Connext Micro ported to Deos
  - No code changes to run in 653 Guest Partition
- RTI Connext DDS development committed to use of Open Standards

Enabling Quality of Service

Communication Quality of Service

- Shaping Network Traffic and Behavior
  - Streaming Data Design Pattern Used
  - Reliable State Data Pattern Used
  - Reliable Command Pattern Used
- Implemented using FACE TSS Plug-ins
  - QoS is per-topic/per-connection
  - Organized into “Profiles”
  - XML for Connext Professional, Modified plugin-ins for Micro

“Yellow” Demonstration Stack



Proven, Deployed Design Patterns

- Many to one
- One to Many
- Commands
- State Data
- Objective-State

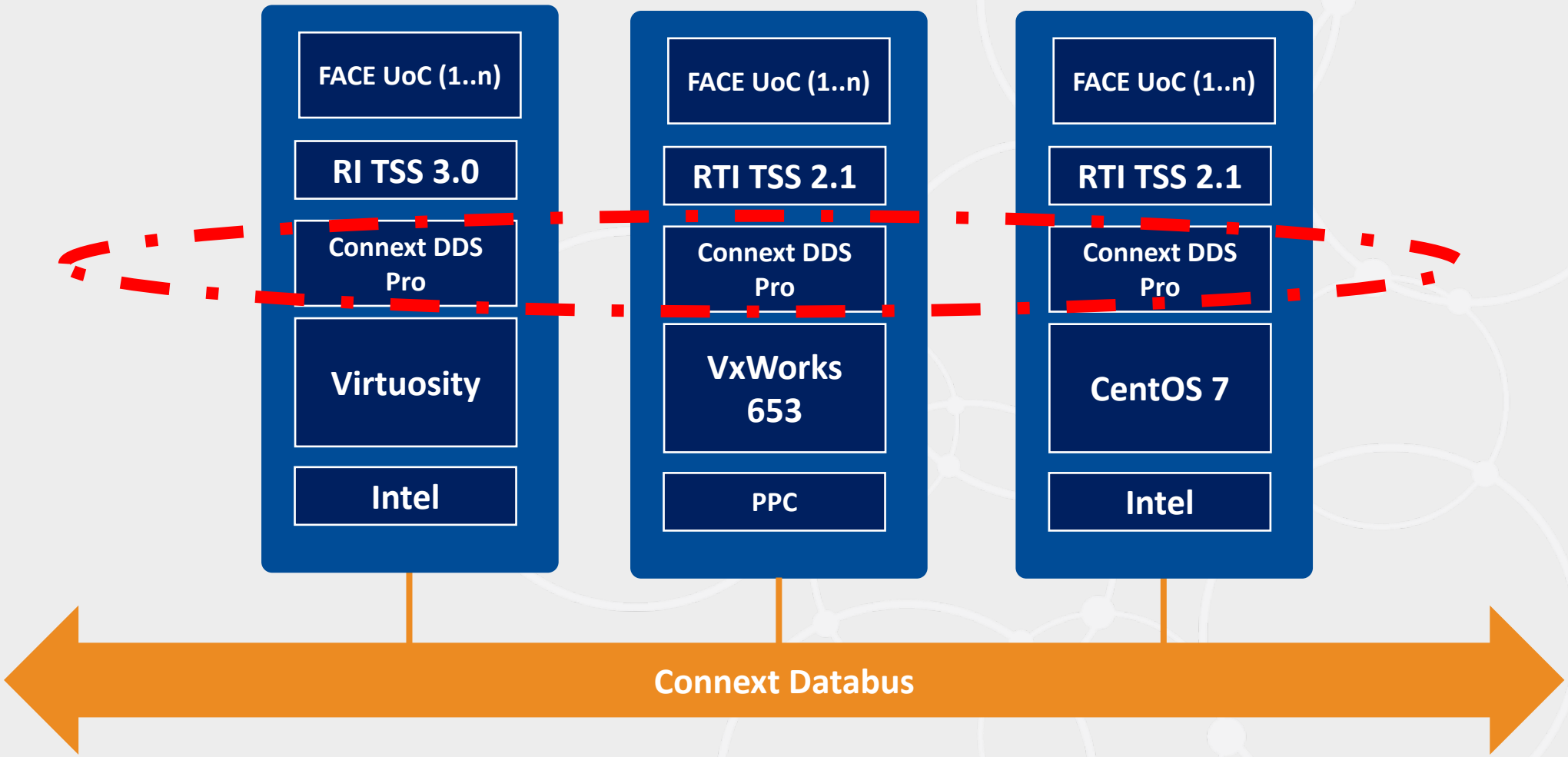
Off the Shelf Profiles

Enabling Network Interoperabilty

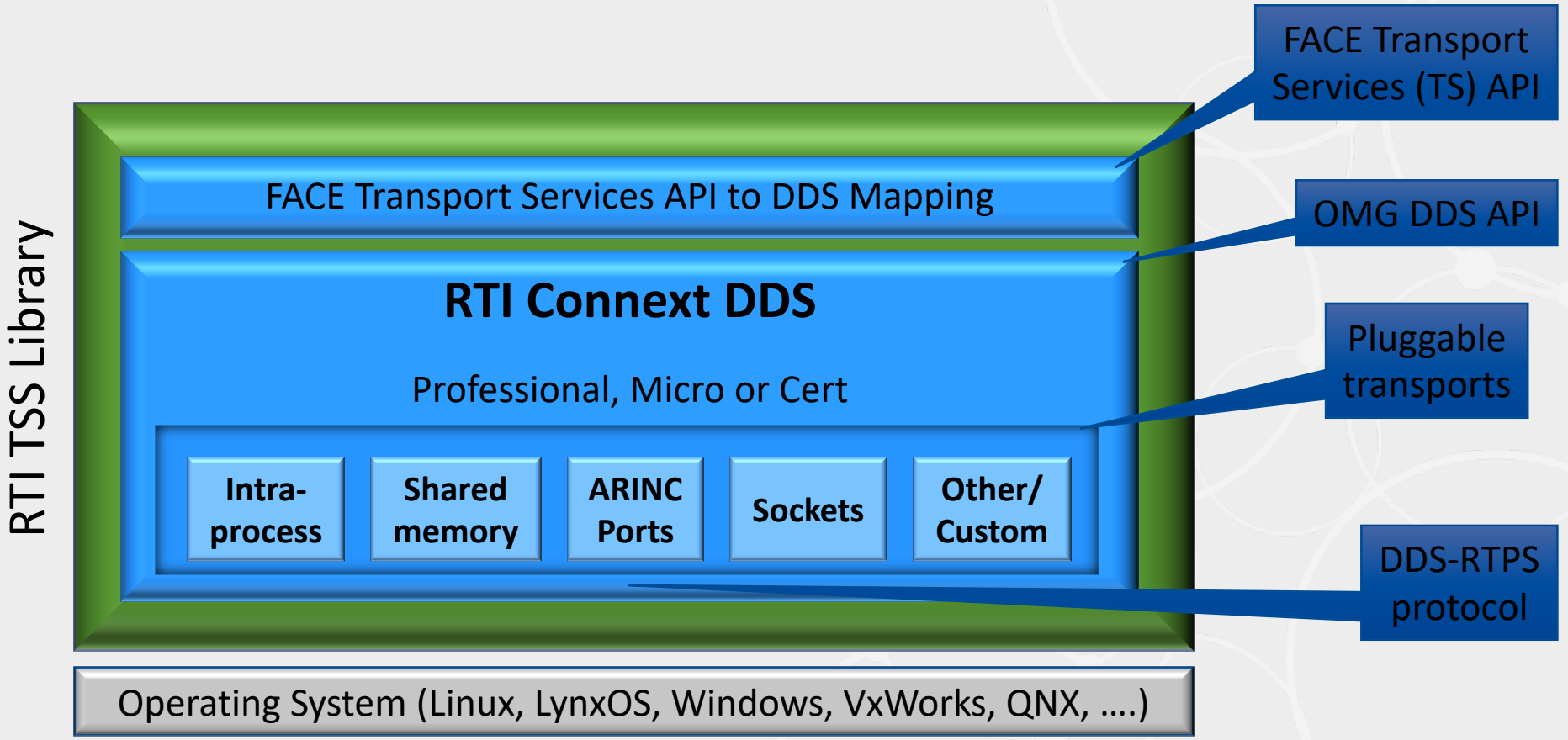
Real-Time Publish-Subscribe (RTPS)

- Open, interoperable wire protocol
- FACE standards do not specify serialization form or wire protocol
- Big-endian (PPC)/Little-endian(Intel) conversions handled by DDS
- Any DDS implementation using RTPS may interoperate

“Yellow” Demonstration Stack



Built on Standard and Open Interfaces

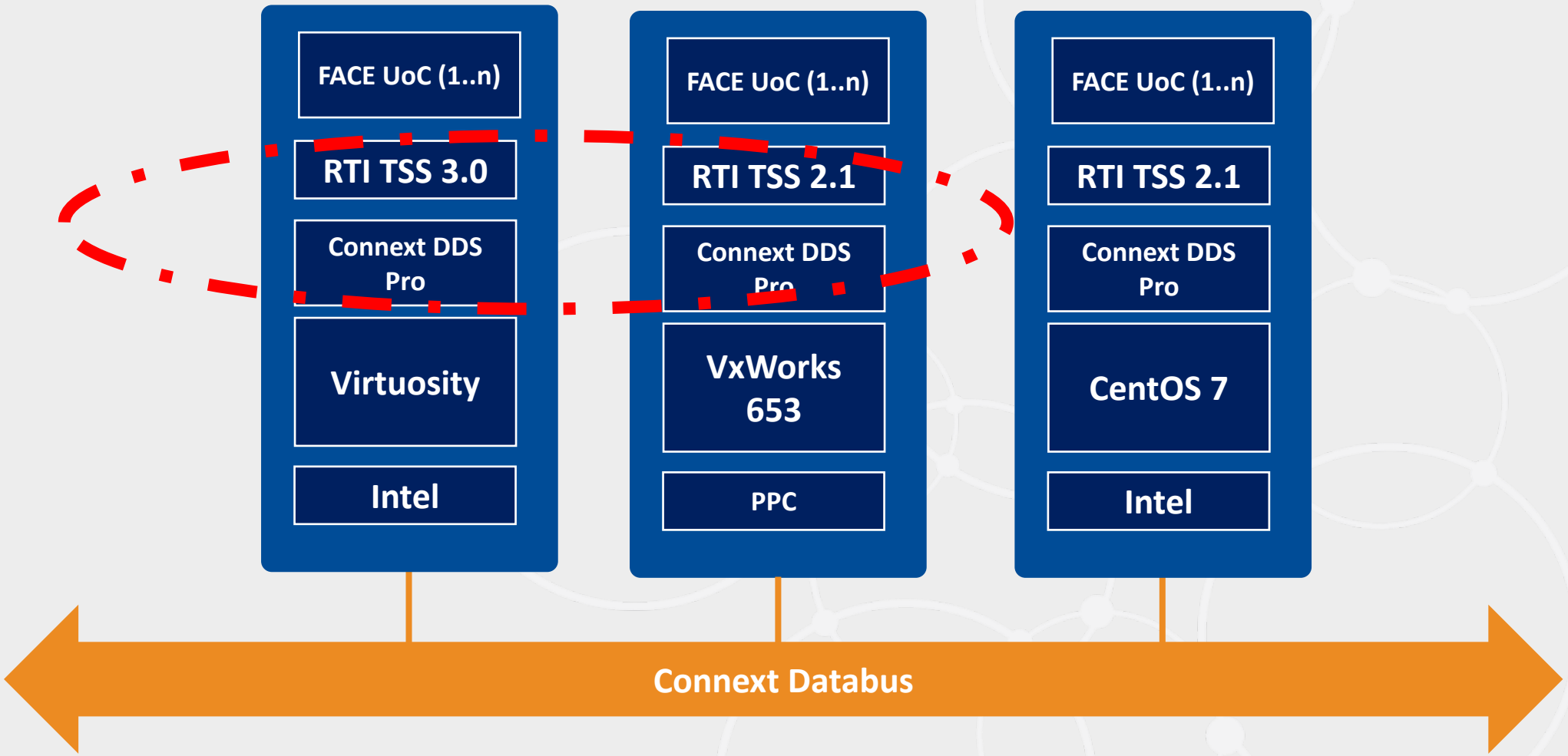


Enabling Backwards Compatibility

Backwards Comapatibility for FACE

- Feature of RTI Connext TSS
  - Design of TSS adds no unique data to message representations
  - Not Required by FACE Technical Standards

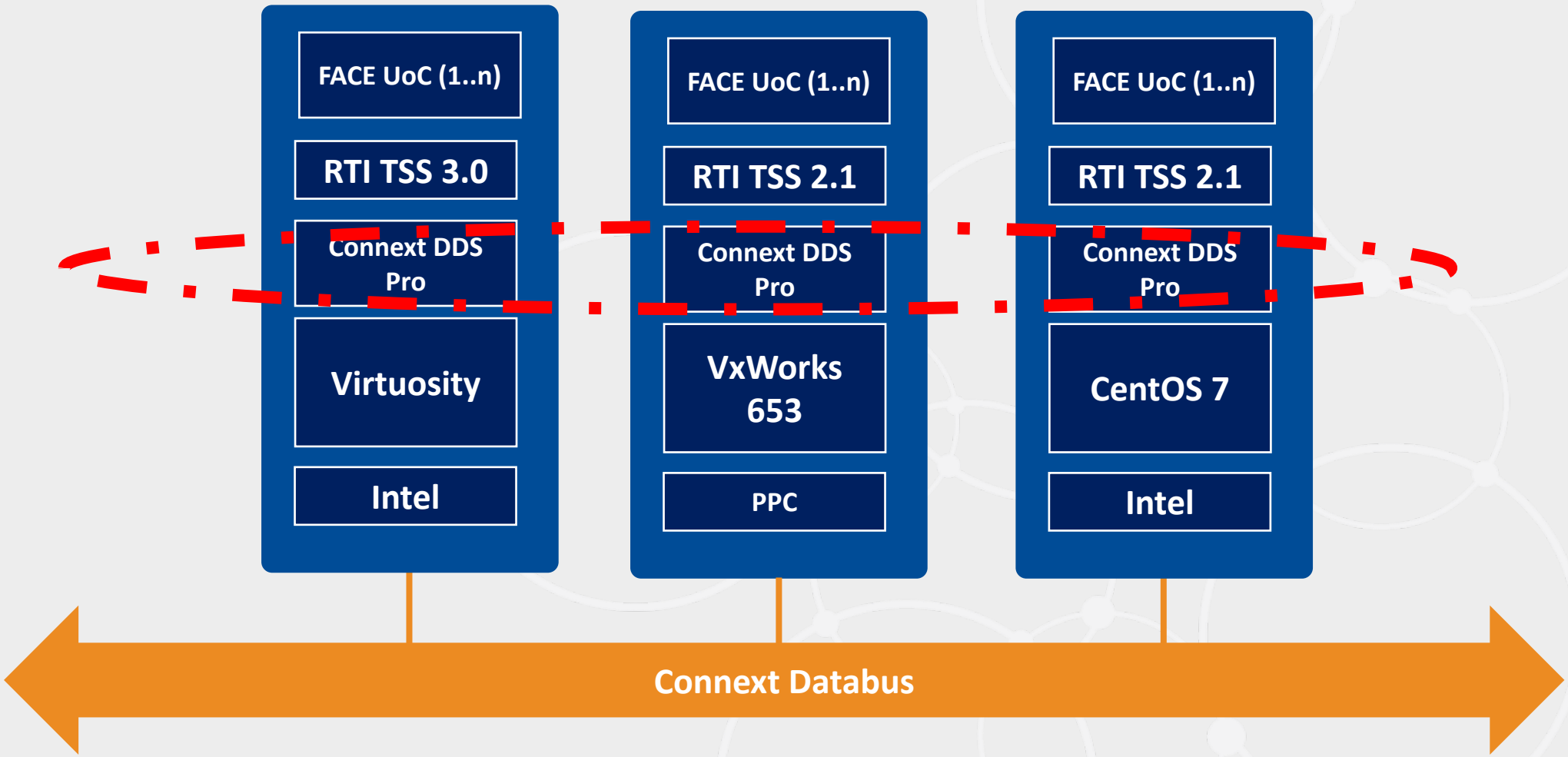
“Yellow” Demonstration Stack



Backwards Comapatibility for FACE

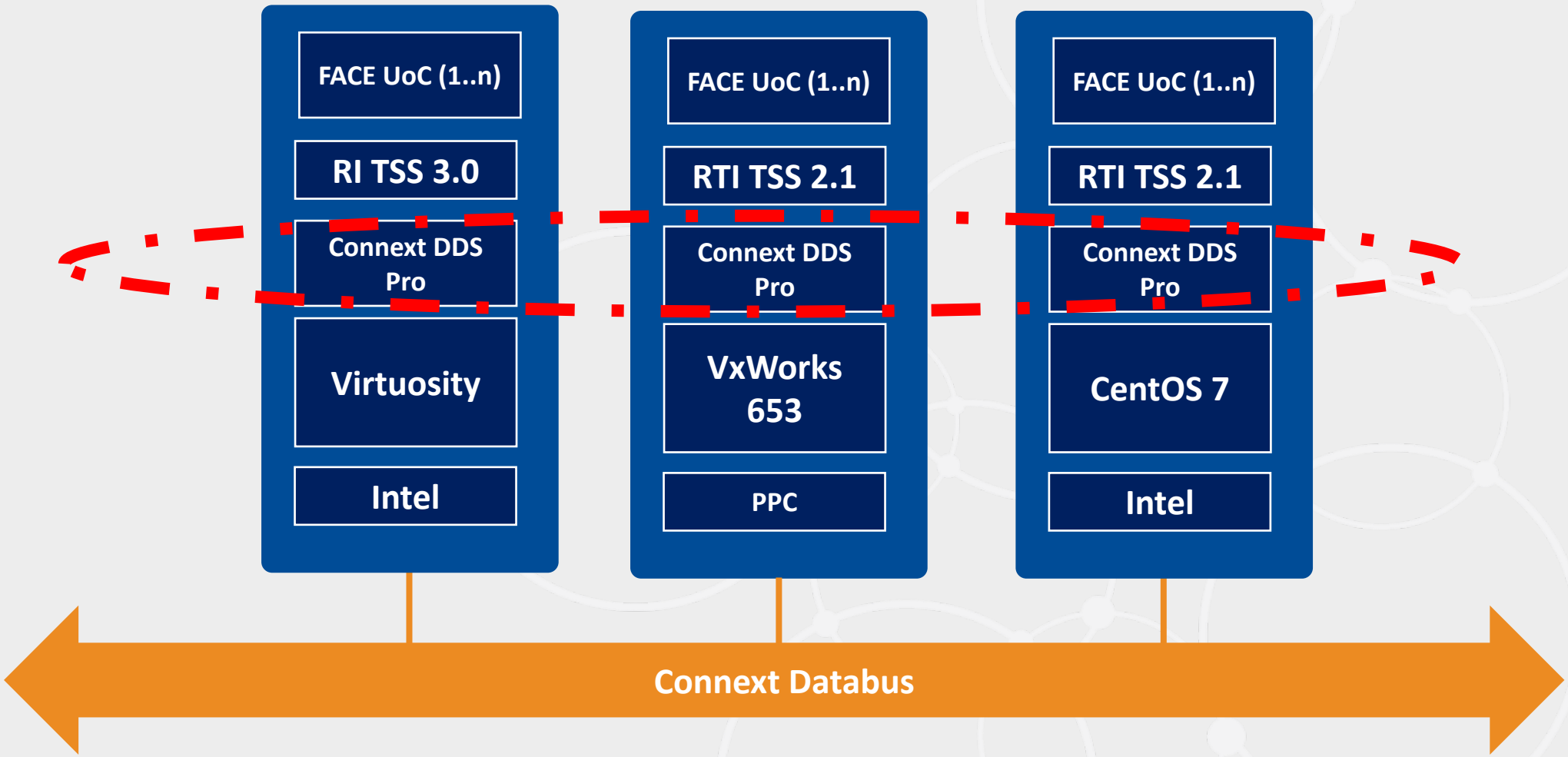
- RTI Connext DDS core is backwards compatible
  - Backwards compatible to 4.x versions (10 years!)
  - Enables incremental upgrade

“Yellow” Demonstration Stack

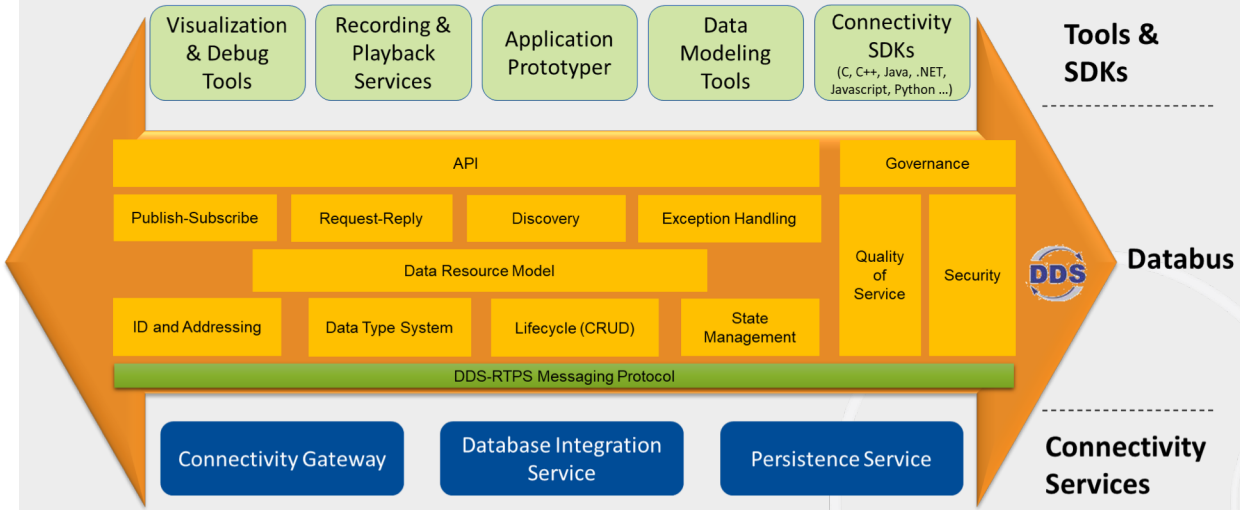


Enabling Integration, Analysis and Debug

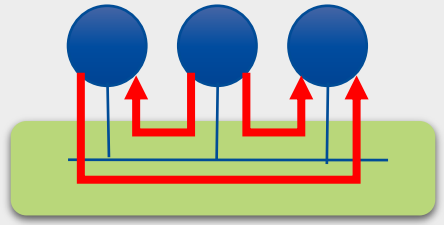
“Yellow” Demonstration Stack



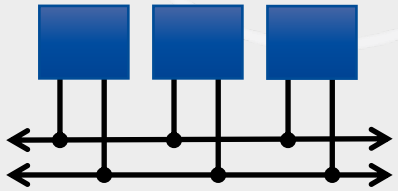
RTI Connext DDS Ecosystem



Databus



Security



Safety Cert

Tools

Services

Utilities

Enabling Rapid Integration

Meeting the Need

- Interoperable Data Model – **FACE,DDS**
- Portable Software Components (APIs) – **FACE, DDS**
- Communication Interoperability between OS/HW platforms - **DDS**
- Configurable communication Quality of Service (behavior) - **DDS**
- Availability of OS/HW platforms – **RTI, DDS**
- Backwards Compatibility – **RTI, DDS**
- Integration and Analysis tools – **RTI, DDS**

Stay Connected

rti.com
Free trial of Connext DDS

rtisoftware

@rti_software

connextpodcast

@rti_software

rti.com/blog