PRISMTECH

Performance Critical Middleware

Using Model Driven Development to Implement SCA v4.0 Compliant DSP and FPGA Based Applications

Andrew Foster, Product Manager, PrismTech Corporation

SDR-WInnComm, January 9th 2012

Agenda

- Introduction to SCA 4.0
- SCA 4.0 CORBA PSM
- SCA 4.0 Middleware
- SCA 4.0 Model Driven Development
  - Addressing the key challenges
  - Application modelling for GPP, DSP and FPGA
  - Hybrid models – split component model
  - Device Modelling
  - Model driven testing

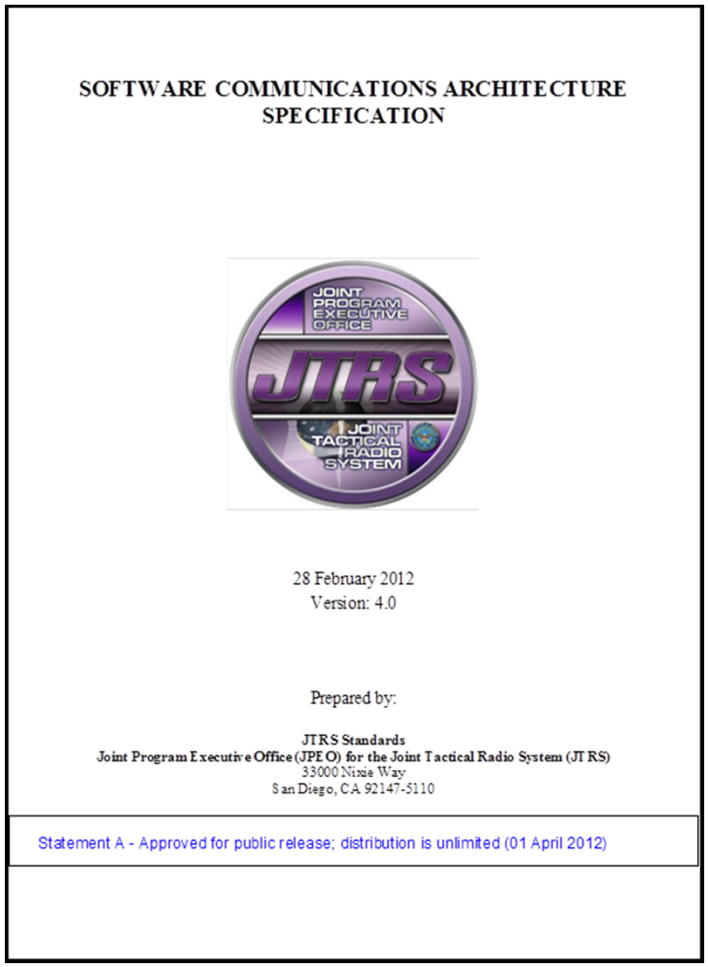
PRISMTECH

PRISMTECH

Performance Critical Middleware

Introduction to SCA 4.0

Background



SOFTWARE COMMUNICATIONS ARCHITECTURE
SPECIFICATION

28 February 2012
Version: 4.0

Prepared by:

JTRS Standards
Joint Program Executive Office (JPEO) for the Joint Tactical Radio System (JTRS)
33000 Nixie Way
San Diego, CA 92147-5110

Statement A - Approved for public release; distribution is unlimited (01 April 2012)

- The Software Communications Architecture (SCA) has remained largely unchanged since 2001 when v2.2 was released and the Joint Tactical Radio System (JTRS) program started.
- SCA 4.0, approved February 2012, represents a radical shift in the approach to specifying the architecture, design and implementation of a software defined radio (SDR).

PRISMTECH

SCA 4.0 Approach

- The SCA 4.0 specification has been developed following the Model Driven Architecture (MDA) approach
- The base specification has been developed as a Platform Independent Model (PIM)
- Appendices define transfer mechanisms to provide co-located or distributed client/server operations.
- Currently the only transfer mechanism indentified in a Appendix E is based on the Common Object Request Broker Architecture (CORBA)
- Appendix E-1 defines the CORBA PSM for use with SCA 4.0

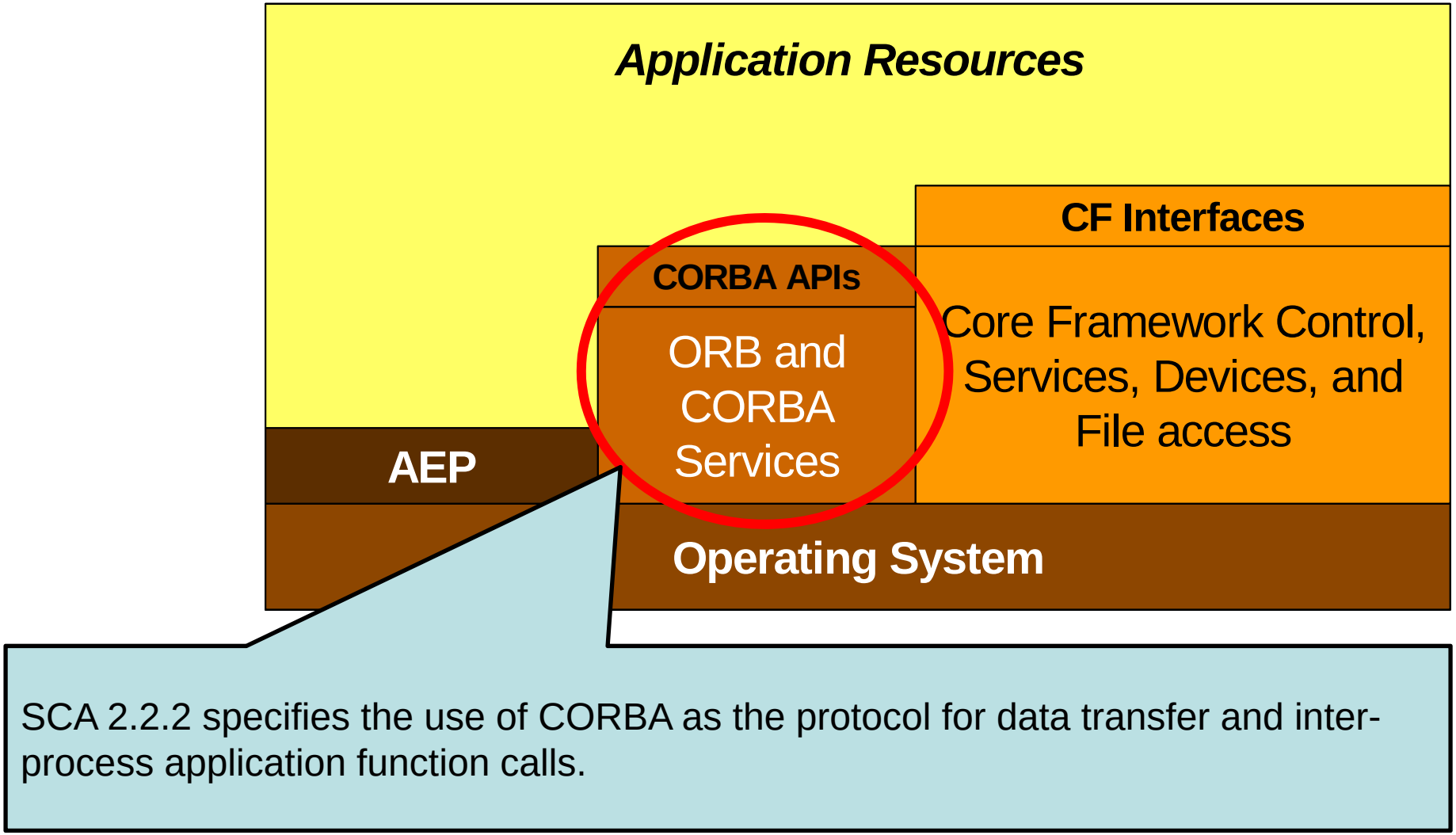
SCA v4.0

- **JPEO Objectives**
  - **Reduce development resources**
    - Budget
    - Schedule
  - **Reduce test and certification time**
    - Reduce number of requirements
    - Increase use of automated testing
  - **Improve performance**
    - Reduce boot up latency
  - **Reduce memory footprint**
  - **Technology refresh**
  - **Incorporate lessons learned**
  - **Backwards compatibility of applications is an overarching tenet**

SCA v4.0 Key Changes

- Lightweight Components
- Component Registration
- Deployment Optimizations
- Profiles and SCA Conformance
- Platform Independent Model

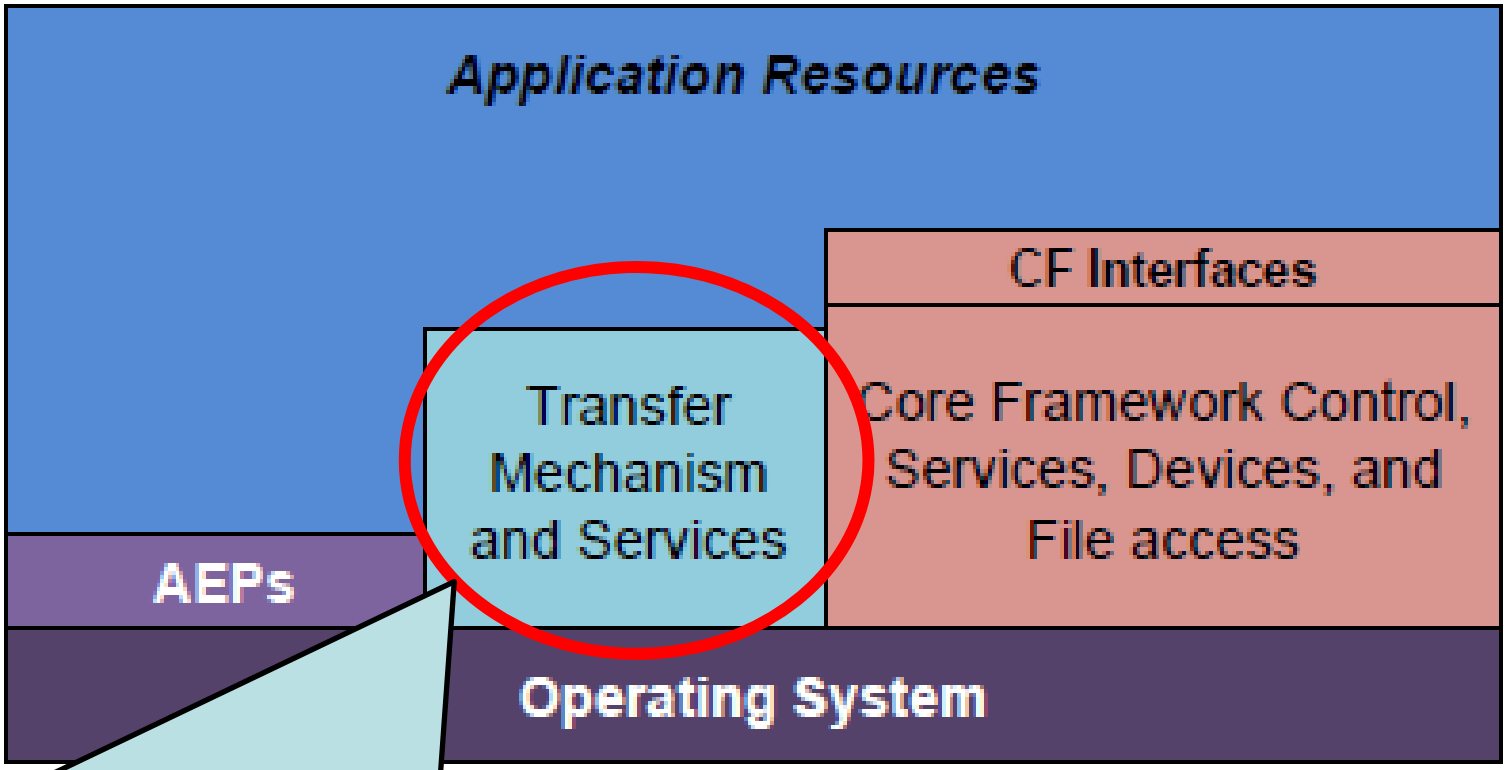
SCA 2.2.2 OE and AEP



SCA 2.2.2 specifies the use of CORBA as the protocol for data transfer and inter-process application function calls.

PRISMTECH

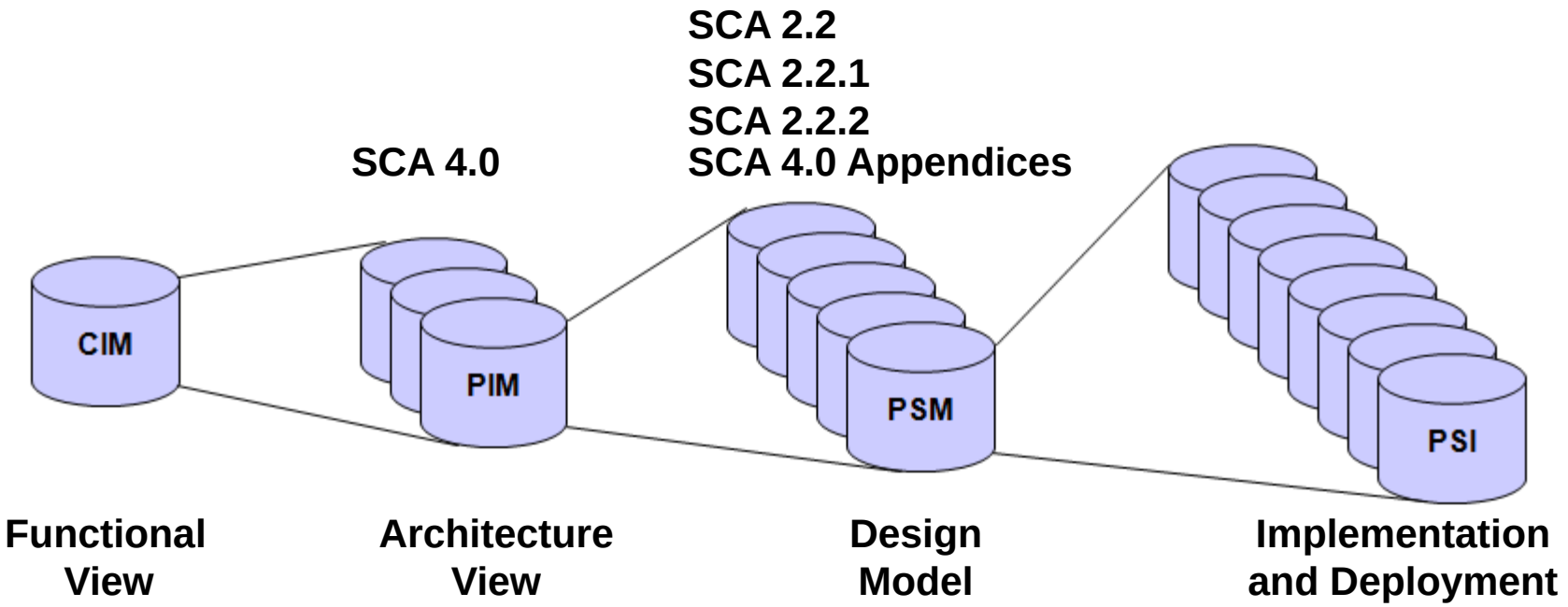
SCA 4.0 OE and AEP



SCA products can be realized using a variety of transports and technologies (e.g. CORBA, C++, SOAP, Data Distribution Service (DDS), MHAL Communication Service, etc.). (excerpt from Appendix E)

PRISMTECH

Model Driven Architecture Views



CIM – Computationally Independent Model
PIM – Platform Independent Model
PSM – Platform Specific Model
PSI – Platform Specific Implementation

PRISMTECH

PSM Appendices

- E Transports and Technologies
- E.1 – Common Object Request Broker Architecture (CORBA)
  - Full – Provides features for general platforms and applications
  - Lightweight – Provides minimal features for highly constrained resources
  - Ultra-Lightweight – Essential capabilities supported by FPGAs
- E.2 – C++
- E.3 – OMG Interface Definition Language

PRISMTECH

SCA 4.0 CORBA PSM

Appendix E-1 CORBA PSM

- Three CORBA profiles based on CORBA/e with additional features from RT CORBA
- The SCA CORBA profiles are characterized as follows:
  1. SCA Full CORBA (Full) Profile – is the Full CORBA profile and is targeted for applications hosted on GPPs
  2. SCA Lightweight CORBA (LW) Profile – is more constrained than the SCA Full CORBA Profile and is targeted towards environments with limited computing support (e.g. DSPs)
  3. SCA Ultra-Lightweight CORBA (ULW) Profile – is more constrained than the SCA Lightweight CORBA Profile and is specifically intended for processing elements with even more limited computing support (e.g., DSPs & FPGAs)

Appendix E-1 CORBA PSM

- Each profile characterizes the IDL features allowed for definition of interfaces between application components.
- The LW Profile narrows the IDL feature set in order to limit the processing overhead caused by a number of types in the Full Profile.
- The ULW Profile narrows the constructs even further to accommodate typical limitations of DSP and FPGA environments.
- The shared IDL foundation of the profiles facilitates portability not only between platforms, but also across processing elements and transfer mechanisms.
- Component portability may be enhanced if the IDL from more constrained profiles are used when defining application interfaces targeted for components deployed within less constrained processing elements.

Full Profile (Highlights)

- Based on CORBA/e Compact Profile
- IDL data types - boolean, octet, short, unsigned short, long, unsigned long, enum, float, double, long double, long long, unsigned long long, char, string, unions, arrays, struct, sequence, object
- Minimum CORBA POA
- Restricted Any data type
- SYNC_SCOPE_POLICY
- RT CORBA – including PriorityModelPolicy, PriorityBandedConnectionPolicy, ServerProtocolPolicy and RT Thread Pools
- COS Event – PushSupplier and PushConsumer
- ORB_init() parameters - allows Root POA to be created with non default parameters

PRISMTECH

Lightweight Profile (Highlights)

- Based on CORBA/e Micro Profile
- IDL data types - boolean, octet, short, unsigned short, long, unsigned long, enum, float, double, long double, long long, unsigned long long, char, string, unions, arrays, struct, sequence, object
- **Any data type not allowed**
- Root only POA
- ORB_init() parameters - allows Root POA to be created with non default parameters

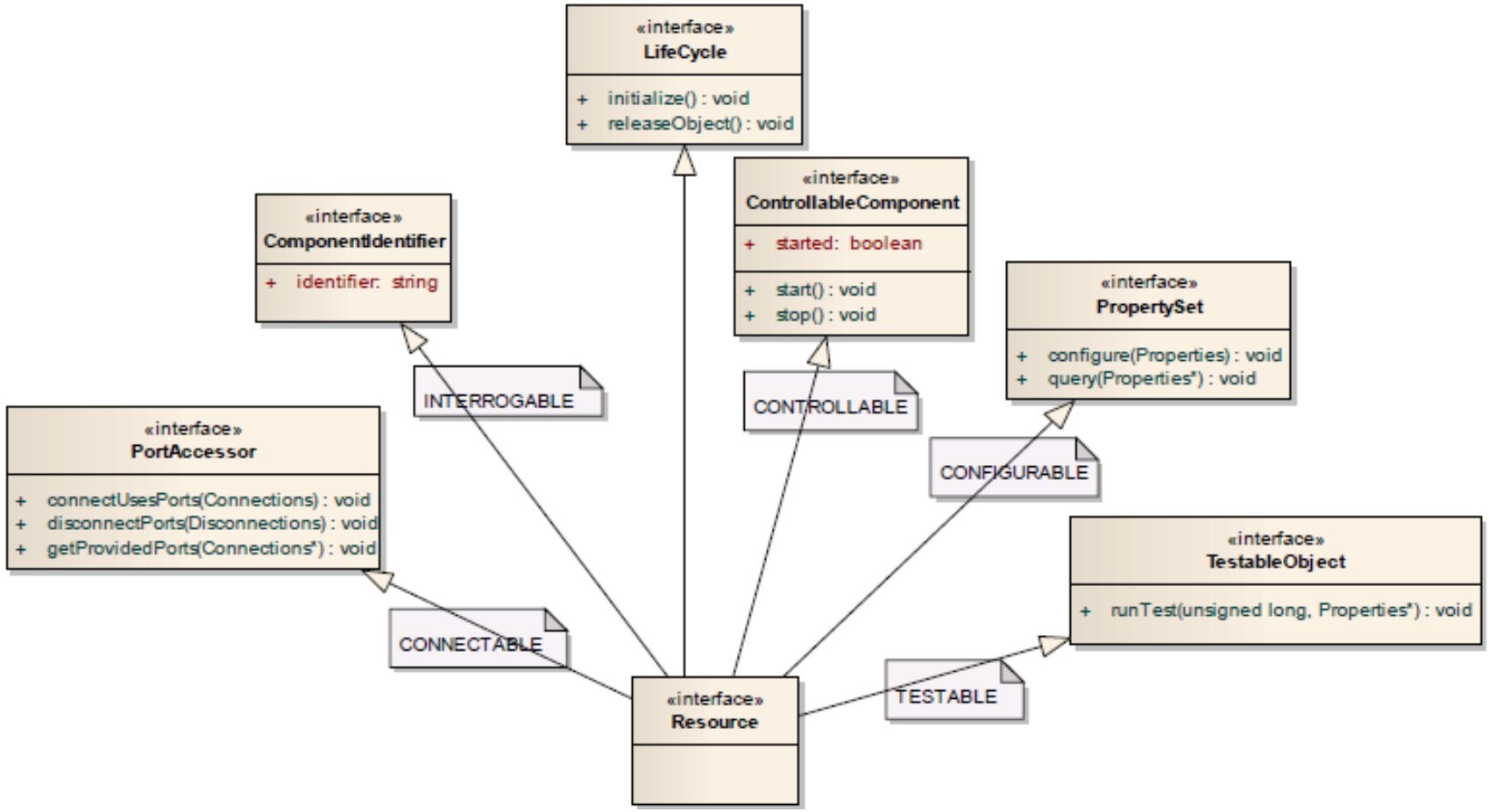
Ultra-Lightweight Profile (Highlights)

- ULW profile only standardises the subset of IDL a CORBA capable FPGA can use
- **String, Any or Object data types not allowed**
- ORB internal details are unspecified and could be implemented in a number of ways, for example:
  1. Software ORB – using a processor core embedded on the FPGA
  2. Hardware ORB – key functions of an ORB implemented as an IP core

IDL basic data types	`Short`
	`Long`
	`unsigned short`
	`unsigned long`
	`Boolean`
	`Octet`
IDL complex data types	`struct` (restricted to supported basic data types)
	`sequence` (restricted to supported basic data types)
	`Enum`
IDL keywords	`Module`
	`Interface`
	`In`
	`Out`
	`Inout`
	`Void`
	`Typedef`
	`oneway`
Return value	`Return` values of a basic data type to be supported

PRISMTECH

SCA 4.0 Resource Interface



PRISMTECH

Resource Interfaces Supported by Each CORBA Profile

Resource Interfaces	Optional Category	Attributes & Operations	Full Profile	Lightweight Profile	Ultra-Lightweight Profile
LifeCycle	Mandatory	initialize():void releaseObject(): void	✓	✓	✓
ComponentIdentifier	INTERROBABLE	identifier: string	✓	✓	×**
PortAccessor	CONNECTABLE	connectUsesPorts(Connections): void disconnectPorts(Connections): void getProvidedPorts(Connections): void	✓	✓	×***
ControllableComponent	CONTROLLABLE	started: boolean start(): void stop(): void	✓	✓	✓
PropertySet	CONFIGURABLE	configure(Properties): void query(Properties): void	✓	×*	×
TestableObject	TESTABLE	runTest(unsigned long, Properties): void	✓	×*	×

* Requires support for Any data type
** Requires support for String data type
** Requires support for Object data type

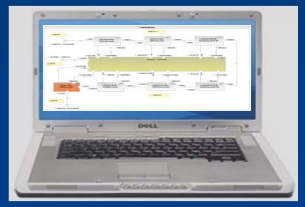
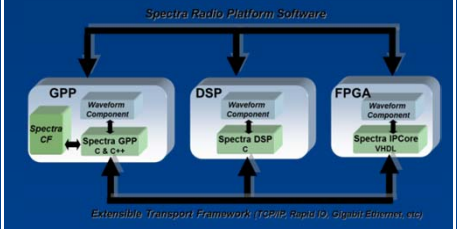
PRISMTECH

PRISMTECH

Performance Critical Middleware

SCA 4.0 Middleware

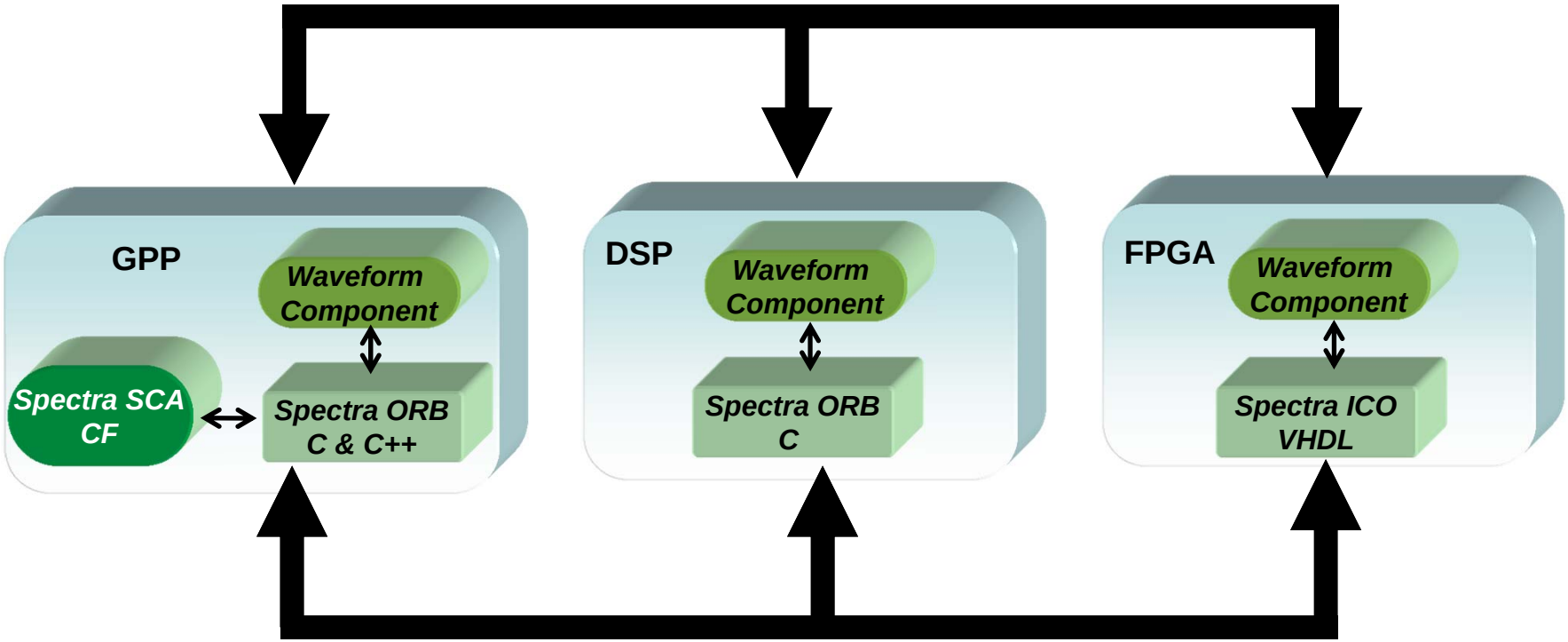
Spectra SDR Product Suite

Spectra CX	Spectra CF	Spectra CDB	Spectra DTP
Spectra CX is a model-driven development tool that enables SDR and waveform software to be rapidly developed, integrated and tested. Spectra CX allows radio platform details to be packaged and delivered to distributed independent development groups using the tools to hide platform complexities and IP as needed.	Spectra Core Framework (CF) is a high-performance, ultra low footprint, radio management framework providing advanced capabilities and extensibilities for multiple software radio architectures.	Spectra Common Data Bus (CDB) is a high-performance, integrated data bus providing a unified data exchange protocol and format. Spectra CDB supports a wide range of General Purpose Processor (GPP), Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) processing elements.	Spectra DTP is an SDR development and test platform that supports waveform design and implementation for military, homeland security and commercial SDRs. Spectra DTP is an optimized small form-factor platform with low power consumption that enables the development, testing and deployment of waveforms.
			

PRISMTECH

Spectra Common Data Bus (CDB) – "SCA Everywhere"

SCA Everywhere



Extensible Transport Framework (TCP/IP, Rapid IO, Gigabyte Ethernet etc.)

Standards based, high performance, low footprint, fully interoperable COTS SCA middleware solution that can be deployed across multiple processor types, including GPP, DSP and FPGA environments

PRISMTECH

Spectra Common Data Bus (CDB)

- Spectra Common Data Bus (CDB) is a fully integrated and optimized Software Defined Radio (SDR) middleware stack
- Spectra Common Data Bus (CDB), runs across a wide range of General Purpose Processor (GPP), Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) processing elements
- Spectra CDB includes the following:
  - Spectra ORB
    - C++ ORB (for GPP)
    - C ORB (for GPP and DSP)
  - Spectra Lightweight Services
    - Spectra Lightweight Naming Service
    - Spectra Lightweight Event Service
    - Spectra Lightweight Log Service
  - Spectra IP Core ORB (ICO) for FPGA and ASIC

PRISMTECH

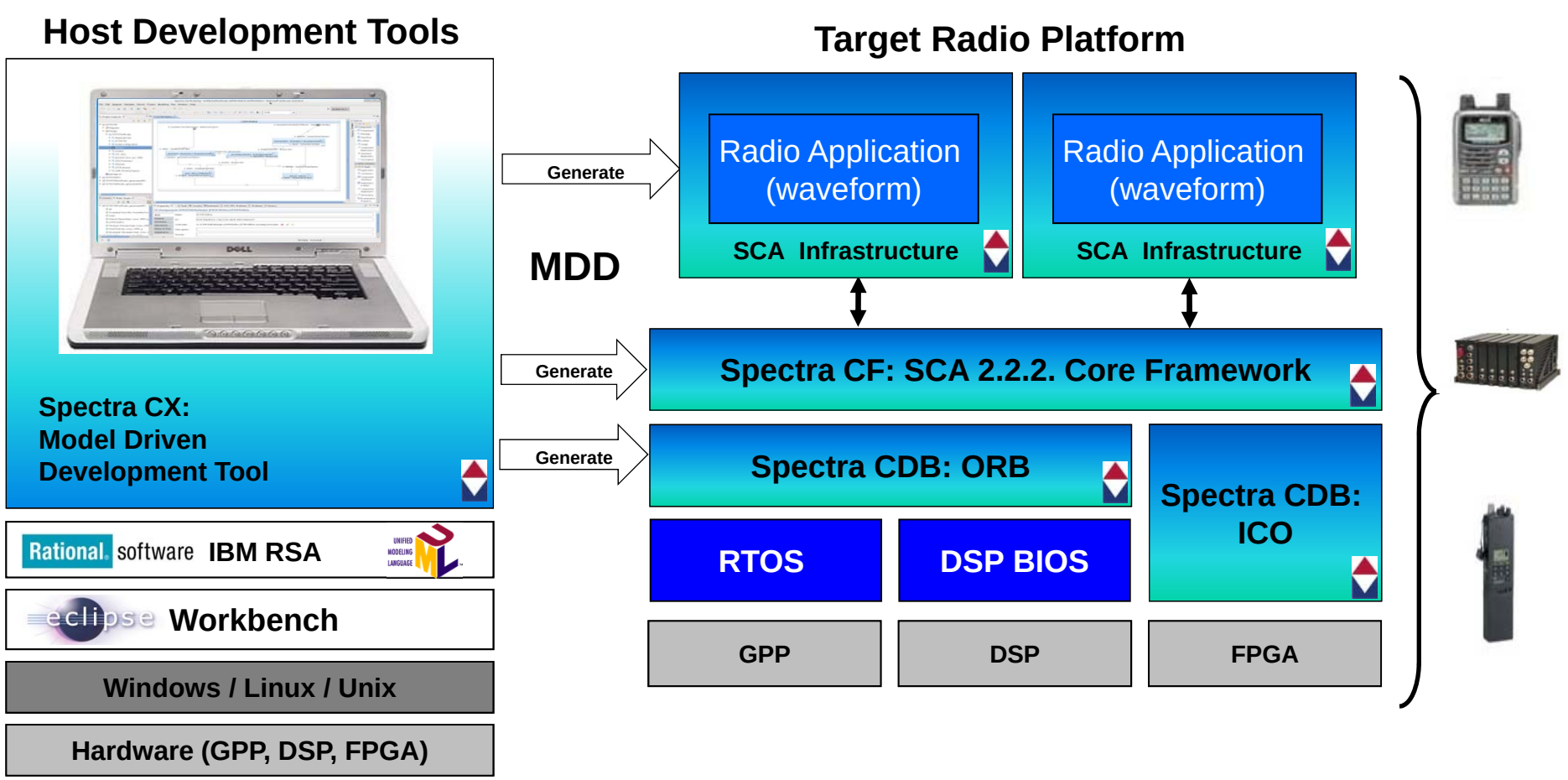
Spectra CDB SCA 4.0 Roadmap

- Spectra CDB suite of CORBA ORBs will be fully SCA 4.0 compliant by early 2013
- Spectra ORB v2.0
  - Spectra ORB C++ Edition will provide support for SCA 4 Full Profile
  - Spectra ORB C Edition will provide support for SCA 4 Full and Lightweight Profiles
- Spectra ICO v2.3
  - ICO already supports SCA 4 Ultra-Lightweight Profile
    - ULW profile defines a subset of the functionality and data types that ICO can actually support

PRISMTECH

SCA 4.0 Model Driven Development

Spectra Model Driven SCA Development



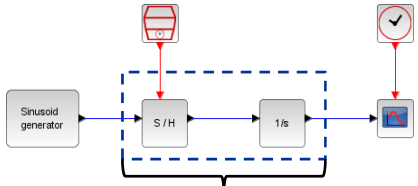
End-to-End: Model, Validate, Generate, Develop, Build, Test, Deploy

PRISMTECH

Spectra CX in the SDR WF Development Flow

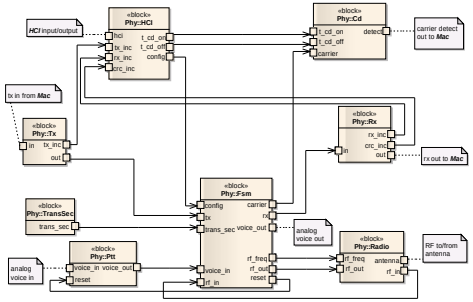
Modeling and simulation of core waveform algorithmic blocks in **MATLAB, SIMULINK, Mathematica, System View** etc.

Functional WF Block Model

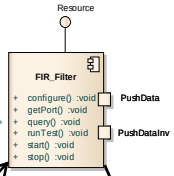


Spectra CX

Fully specified application, deployment constraints, interconnections and protocols

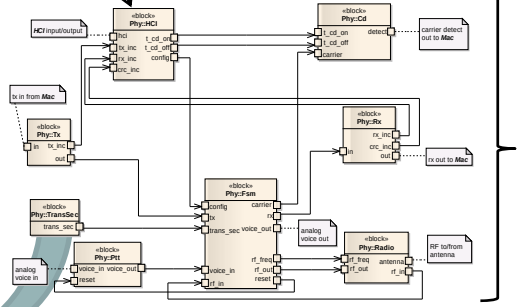


Selected Blocks Map to:



Software Component

Component Library



Collection of Components form a Composite Component or Application

Target platform capabilities and constraints

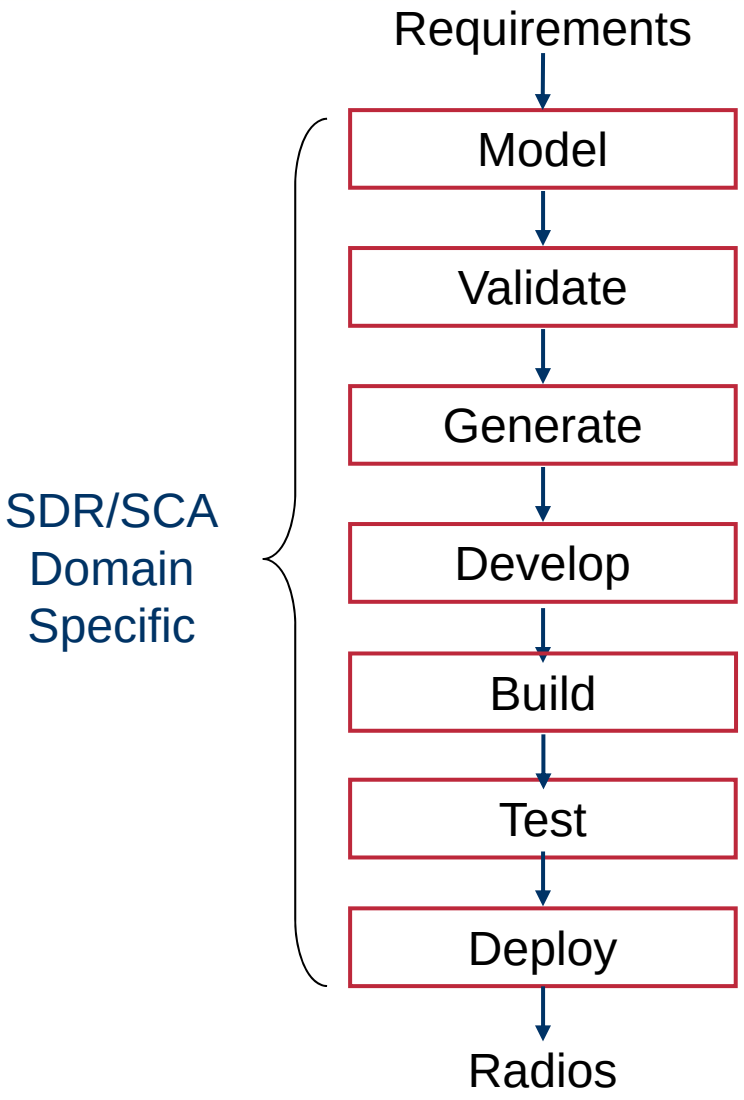
Deployment Criteria and Protocol is specified

Deployment Engine

Waveform application is deployed on target platform

PRISMTECH

Spectra CX Radio Development



Use UML2 to capture the requirements of the system

Components, Applications, Devices, Platforms, and deployments of Waveforms on target platforms

Validate the correctness of designs, plans, and imported artifacts against standard

Correct-by-construction generation of the descriptors (XML), SCA structural code, and makefiles

Behavior code using Eclipse IDE coding tools or 3rd party UML, Block Diagram, and State Chart design tools

Build configuration management for the Target Platforms

Automated unit testing of components and subsystems of an application (waveform)

Runtime monitor allows users to deploy and inspect an application in real-time

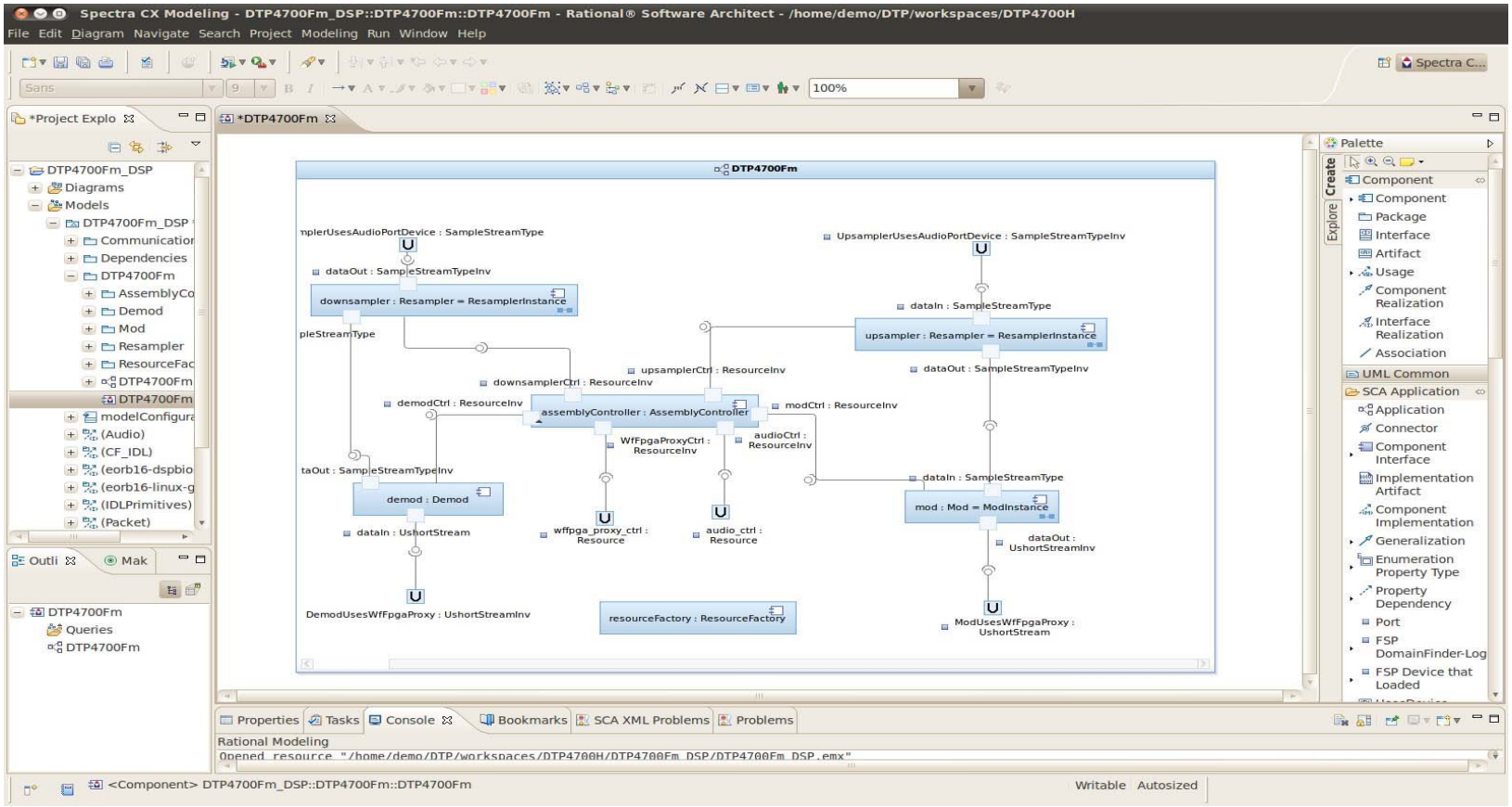
XML descriptors (sad, dcd, dmd, spd, scd, prf), IDL, and Binary files (executable, loadable)

PRISMTECH

SCA 4.0 Key Challenges

- SCA 4.0 is not backwardly compatible – how do I migrate my SCA 2.2.2 applications or platforms to SCA 4.0 ?
- How can I build SDRs that leverage the benefits of Lightweight Components based on the optional UOFs that SCA 4.0 has introduced ?
- Optionality has benefits but it also introduces complexity and the possibility for inconsistencies and errors. How can this be managed ?
- How can I leverage the new CORBA PSM and deploy SCA components across GPP-DSP-FPGA processing elements ?
- How do we ensure that we can leverage the benefits of the SCA 4.0 PIM/PSM separation as new SDR technologies emerge (e.g. new non CORBA middleware) ?

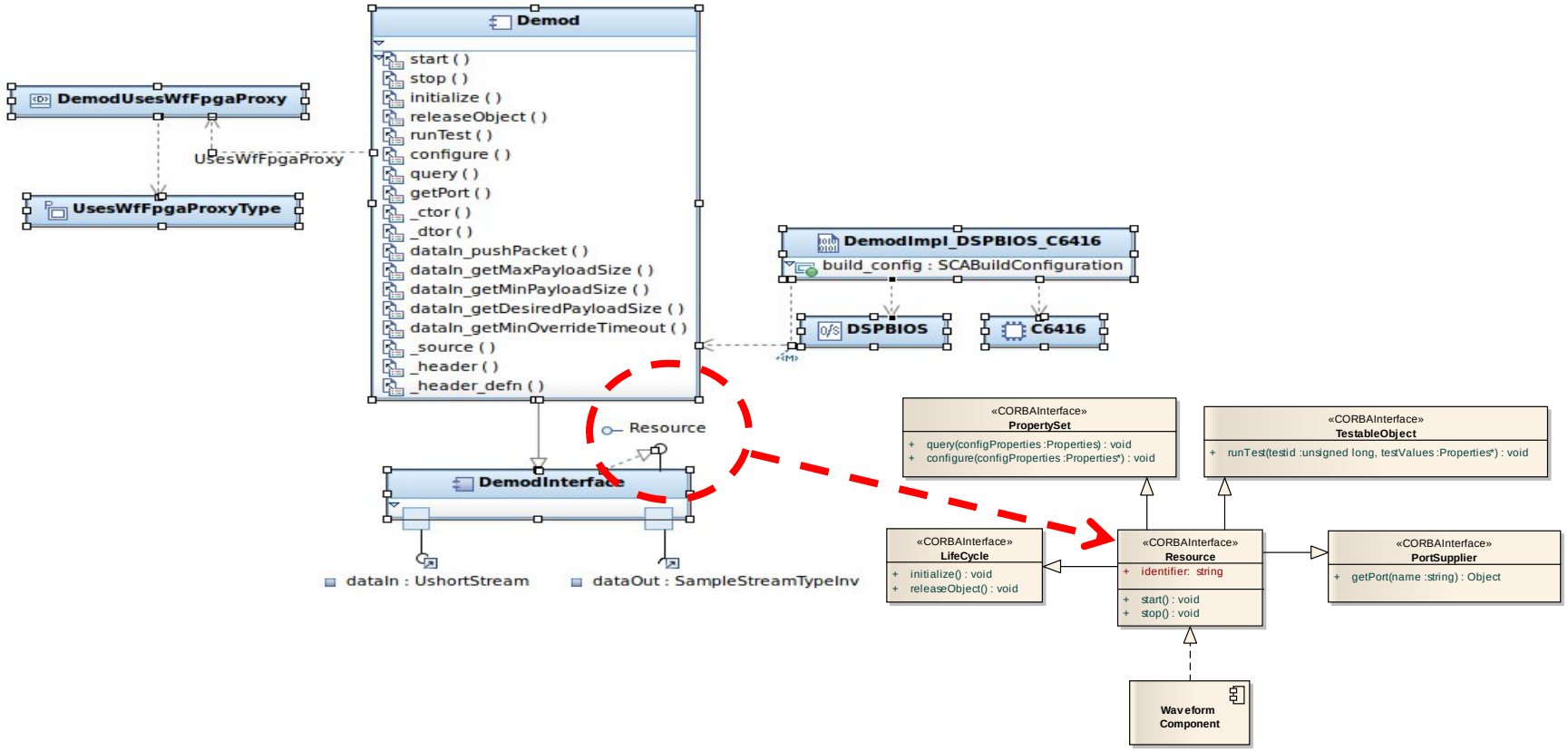
SCA 2.2.2 Model Driven Development



- Spectra CX currently supports SCA 2.2.2 application and platform development

PRISMTECH

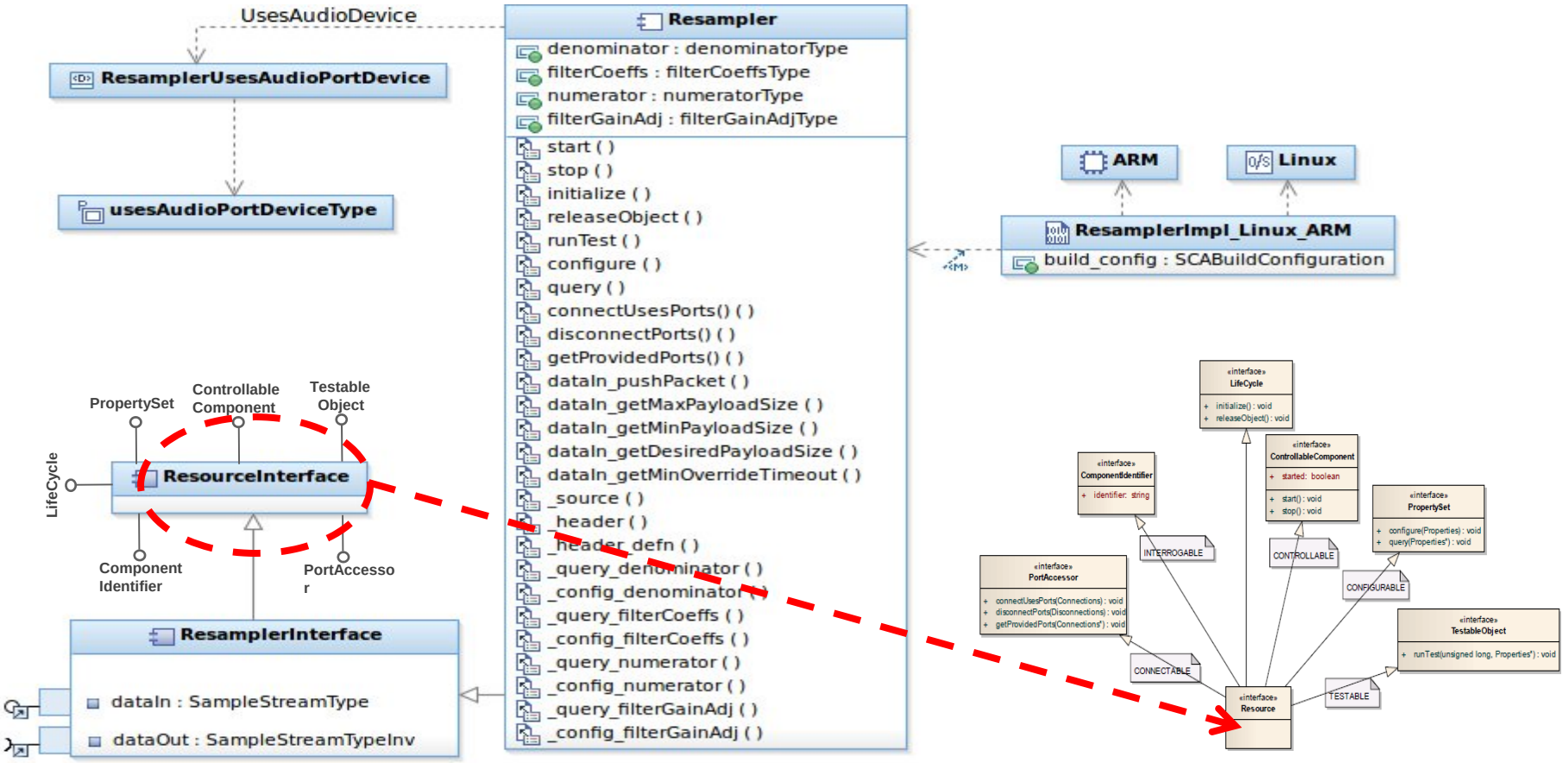
SCA 2.2.2 MDD for GPP & DSP



- Spectra CX supports the creation of application components that can be deployed on a GPP (C or C++) but also on a DSP (C)
- A DSP component must implement the full SCA 2.2.2 Resource interface

PRISMTECH

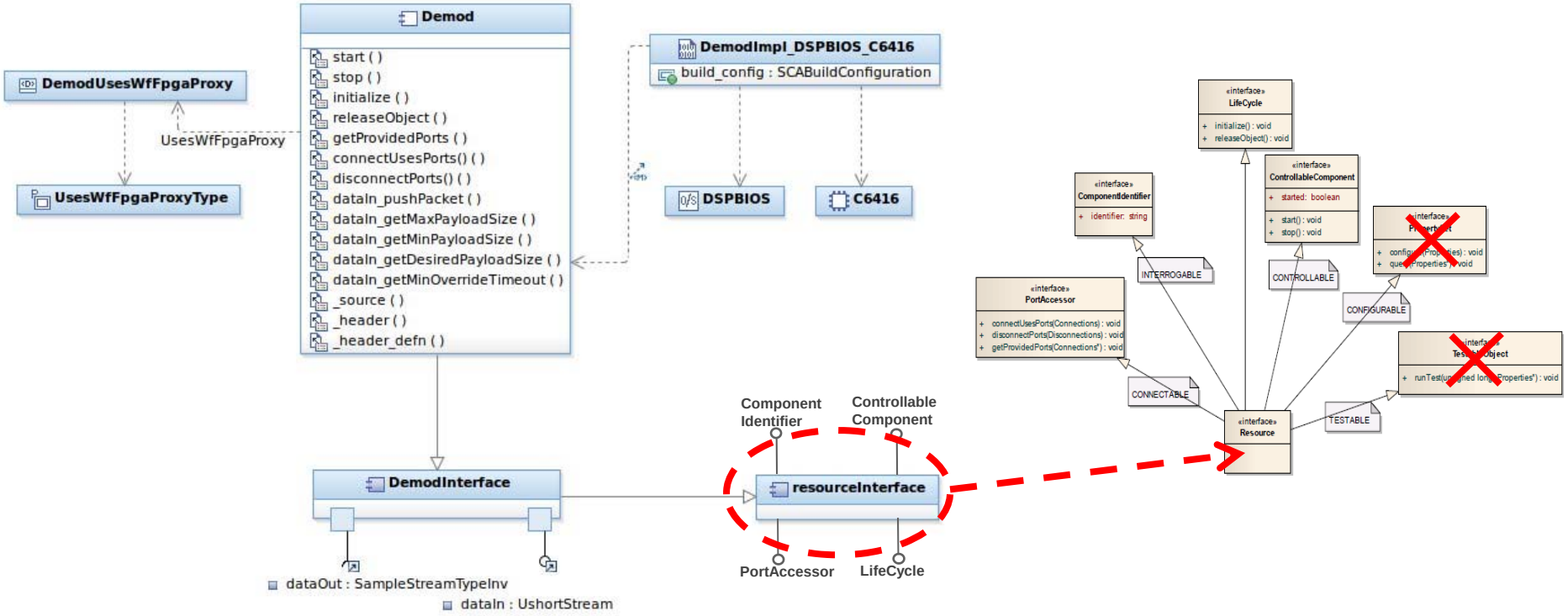
SCA 4.0 MDD for the GPP



- In the future Spectra CX will be extended to support SCA 4.0 Resource components including support for modelling of optional interfaces
- Example above provides illustration of an SCA 4.0 Resource component implementing all optional interfaces and targeted at a GPP

PRISMTECH

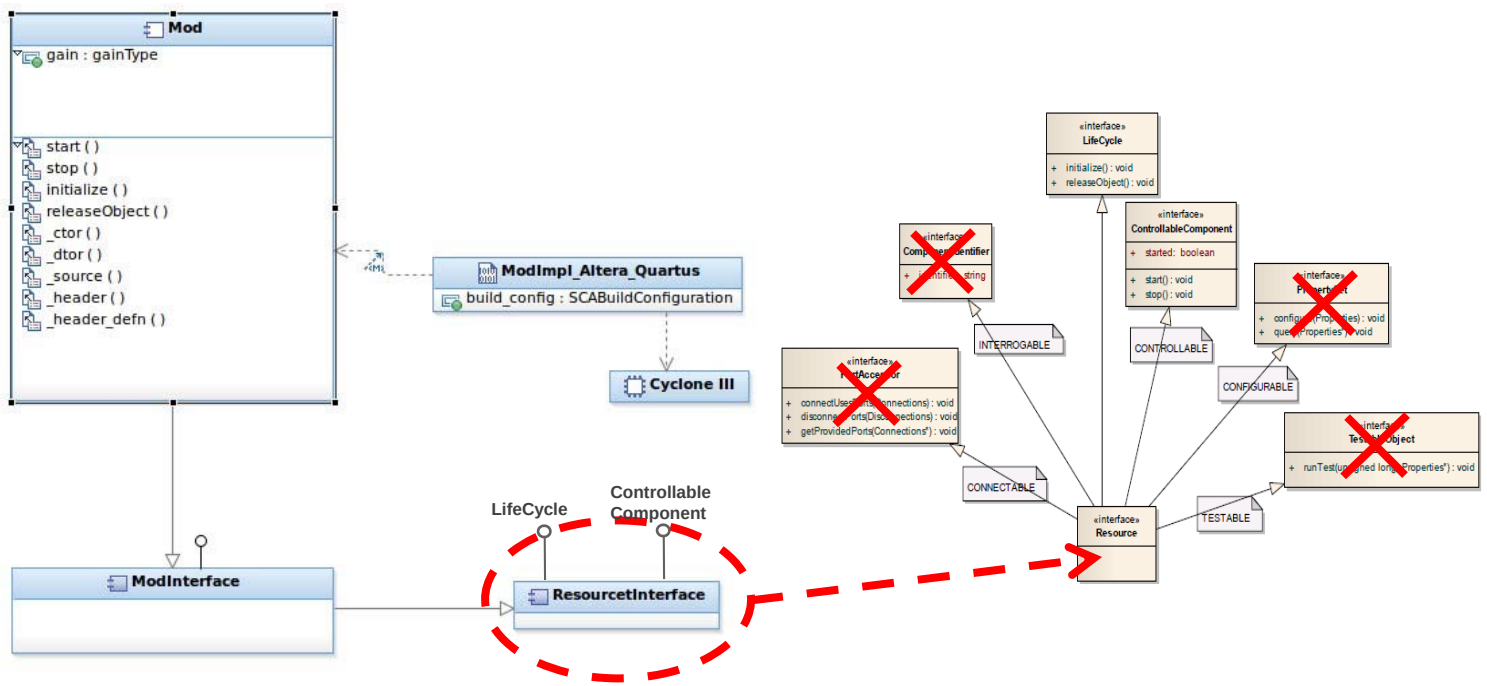
SCA 4.0 MDD for the DSP



- Example above provides illustration of an Resource component implementing optional interfaces and targeted at a DSP enabled with SCA 4.0 CORBA middleware
- For an SCA 4.0 Resource component that will be deployed on the FPGA due to restrictions in the Lightweight Profile the optional PropertySet and TestableObject interfaces cannot be supported
- The tool can be use to validate such constraints at design time instead of running into problems at generation, build or even runtime

PRISMTECH

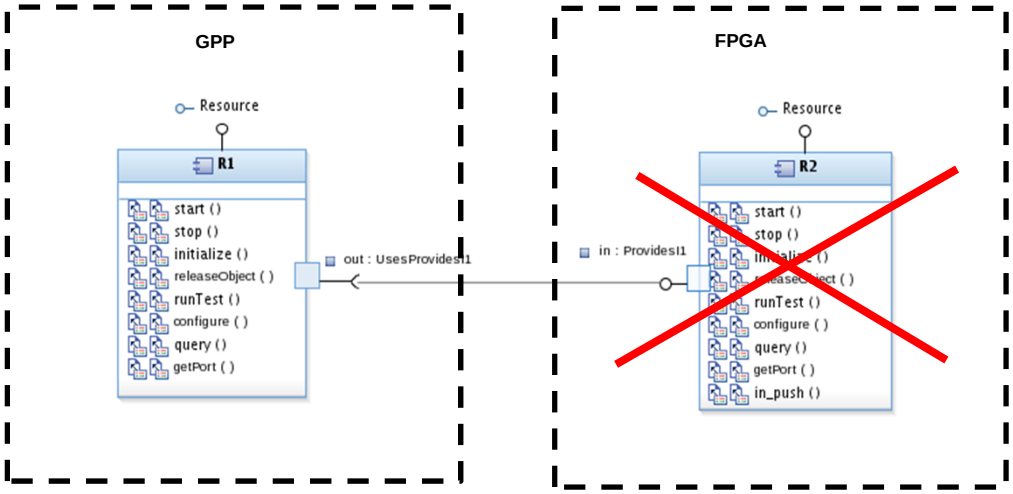
SCA 4.0 MDD for the FPGA



- Example above provides illustration of an SCA 4.0 Resource component implementing optional interfaces and targeted at a FPGA enabled with SCA 4.0 CORBA middleware
- For an SCA 4.0 Resource component that will be deployed on the DSP due to restrictions in the Ultra- lightweight Profile the optional ComponentIdentifier, PropertySet,TestableObject and PortAccessor interfaces cannot be supported
- The tool can be use to validate such constraints at design time instead of running into problems at generation, build or even runtime enabled with SCA 4.0 CORBA middleware

PRISMTECH

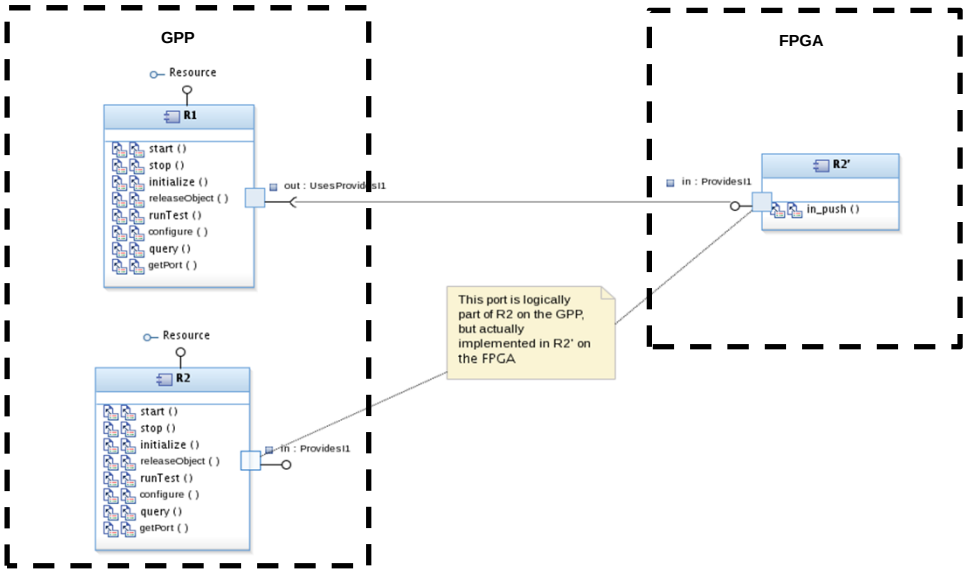
Split Component Modelling (1)



- Lightweight and Ultra-lightweight profiles restrict the interfaces that a Resource component deployed on the DSP or FPGA can support.
- Creating a component with ports on the FPGA (as shown above) or that can support configure/query on the DSP is not possible when using the currently defined SCA 4.0 CORBA profiles

PRISMTECH

Split Component Modelling (2)



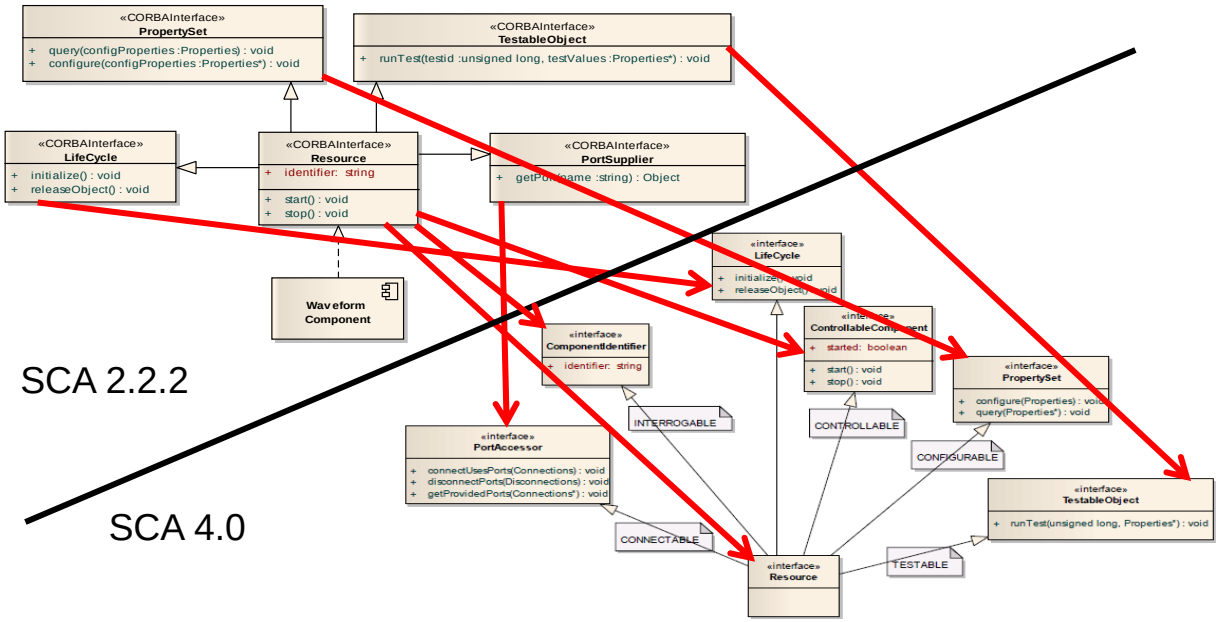
- One possible approach is to host the parts of a Resource interface that cannot be support ed on a DSP or FPGA on the GPP
- The system view is of a single logical component , however the physical implementation of this component is split across the GPP and the DSP or FPGA
- In the example above connection establishment between R1 and R2 is via the getPort() operation on R2 GPP component. However the physical connection is made to a object reference for the R2' component on the FPGA implementing the in_push() operation
- Modelling tools such as Spectra CX can be used to model and auto-generate the code required to implement this split component behaviour

PRISMTECH

SCA 2.2.2 to SCA 4.0 Migration

- A significant issue that must be addressed is that SCA 4.0 does not support backwards compatibility with SCA 2.2.2 application and platform components as originally envisaged
- An SCA 4.0 Core Framework will not be able to deploy an SCA 2.2.2 waveform
- Significant parts of the Resource and Device interfaces such as the connection APIs have changed between SCA 2.2.2 and 4.0
- Manually migrating SCA 2.2.2 application or platform components to SCA 4.0 will be an expensive and time consuming process
- This is major area where MDD approaches can bring significant benefits by helping automate much of the migration process

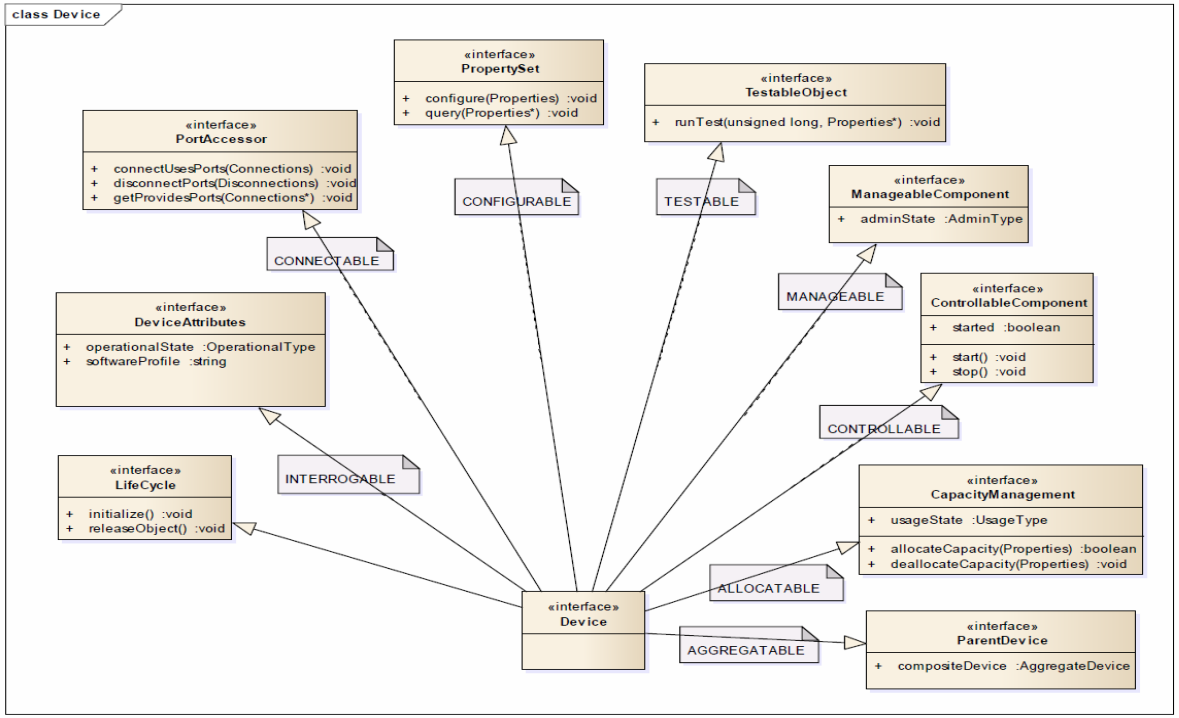
Automating the SCA 2.2.2 to SCA 4.0 Migration Process



- MDD tools such as Spectra CX will be able to auto-generate an SCA 4.0 model from a SCA 2.2.2 model using a set of mapping rules
- The MDD tools will generate the SCA 4.0 component container code (including XML, source code, make files) based on the target PSM technologies
- If the business code is also being maintained as part of the SCA 2.2.2 model as is possible with tool such Spectra CX then it can also be automatically migrated into the new SCA 4.0 model
- If the business code is being maintained independently (e.g. library includes) then these references can be automatically migrated into the SCA 4.0 model

PRISMTECH

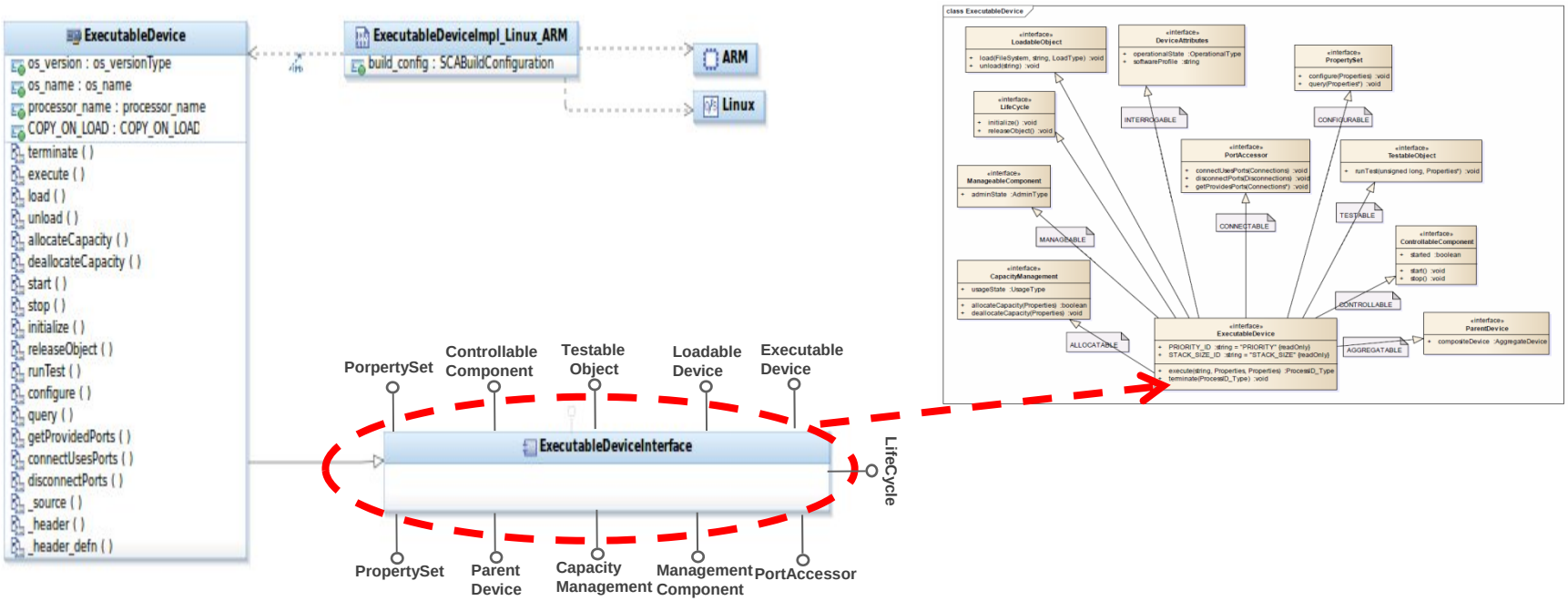
SCA 4.0 Devices



- SCA 4.0 Devices also support the concept of optional interfaces
- Only mandatory interface is LifeCycle
- A LoadableDevice component also inherits the LoadableDevice interface
- An ExecuatbleDevice component also inherits both LoadableDevice and ExecutableDevice interfaces

PRISMTECH

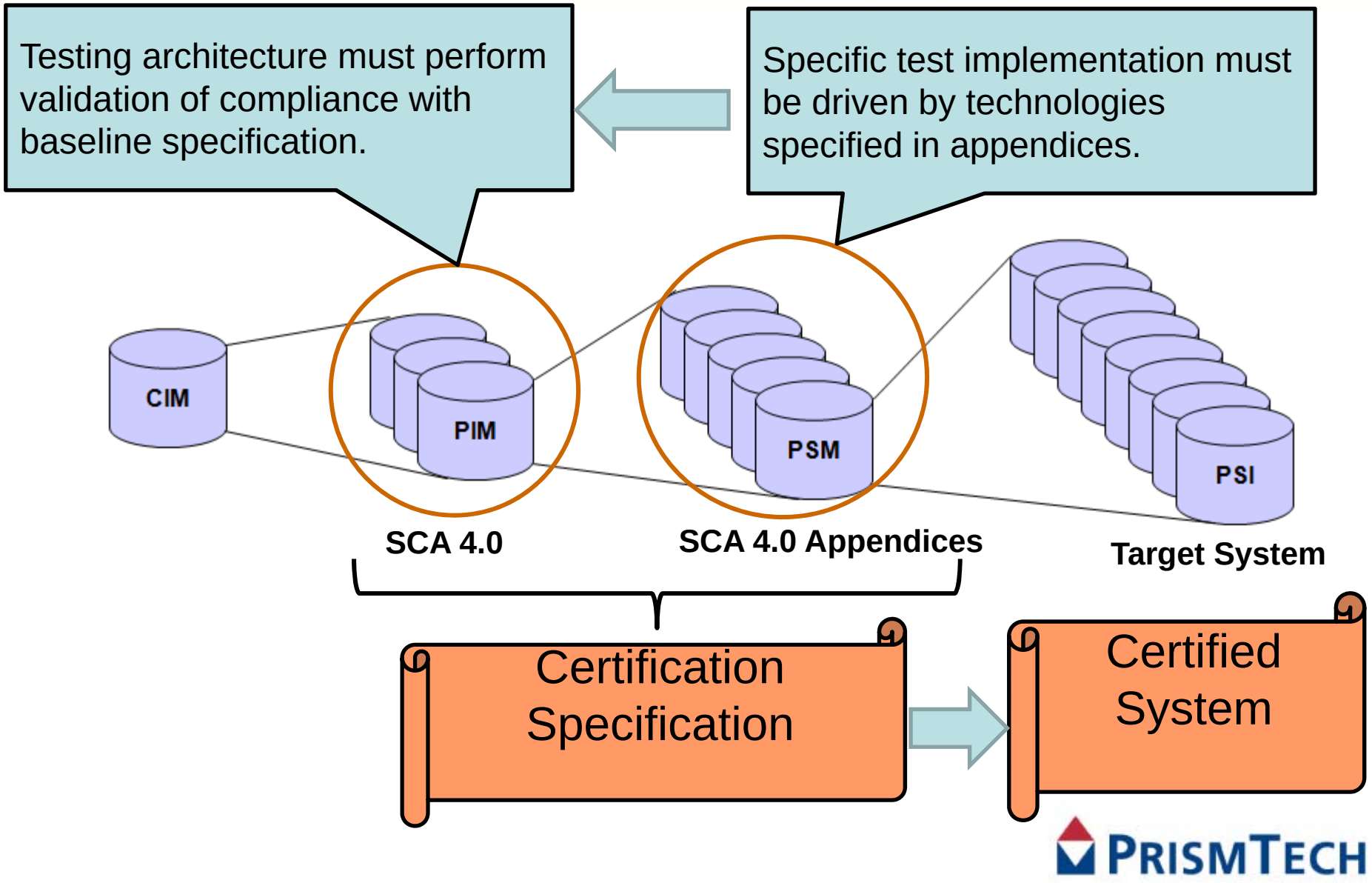
SCA 4.0 Device Modelling



- In the future Spectra CX will be extended to support SCA 4.0 Devices components including support for modelling of optional interfaces
- Example above provides illustration of an SCA 4.0 ExecuatbleDevice component implementing all optional interfaces and representing a GPP processor

PRISMTECH

SCA 4.0 Model Driven Testing



Summary

- SCA 4.0 introduces major changes to the standard
- Interfaces now defined specified as a PIM that can be mapped to different PSMs
- Standard aligns with a Model Driven Development approach to developing SDRs
- New CORBA PSM extends SCA support for DSPs and FPGAs
- Standard introduces new challenges that SDR developers will face, including:
  - How to leverage previous SCA investments – migrating from SCA 2.x to 4.x ?
  - How to leverage the benefits of more sophisticated PSM technologies ?
  - How to manage the extra complexity that having many more choices adds ?
  - How to test and certify SCA systems based on different interface profiles and underlying PSM technologies ?
- Model Driven Development and advanced tooling will be key to the successful adoption of SCA 4.0

PRISMTECH

Additional Information

- Andrew Foster e-mail:
  - awf@prismtech.com
- Web Site:
  - www.prismtech.com/spectra
- Or contact your PrismTech account manager

PRISMTECH

Thank You

PRISMTECH