



Powering Netcentricity

The Software Communications Architecture (SCA) and FPGAs

Meeting the challenges of integrating FPGA application components using the SCA

May 26th 2011 – Andrew Foster, Spectra Product Manager

1. Introduction
 1. FPGA User in SDR
 2. SCA Objectives
2. SCA device model and MHAL
3. CORBA Everywhere
4. Spectra IP Core
 1. Architecture
 2. IDL to VHDL mapping
5. FPGA Design with ICO
6. ICO CORBA Profile
7. Standardization
8. Summary

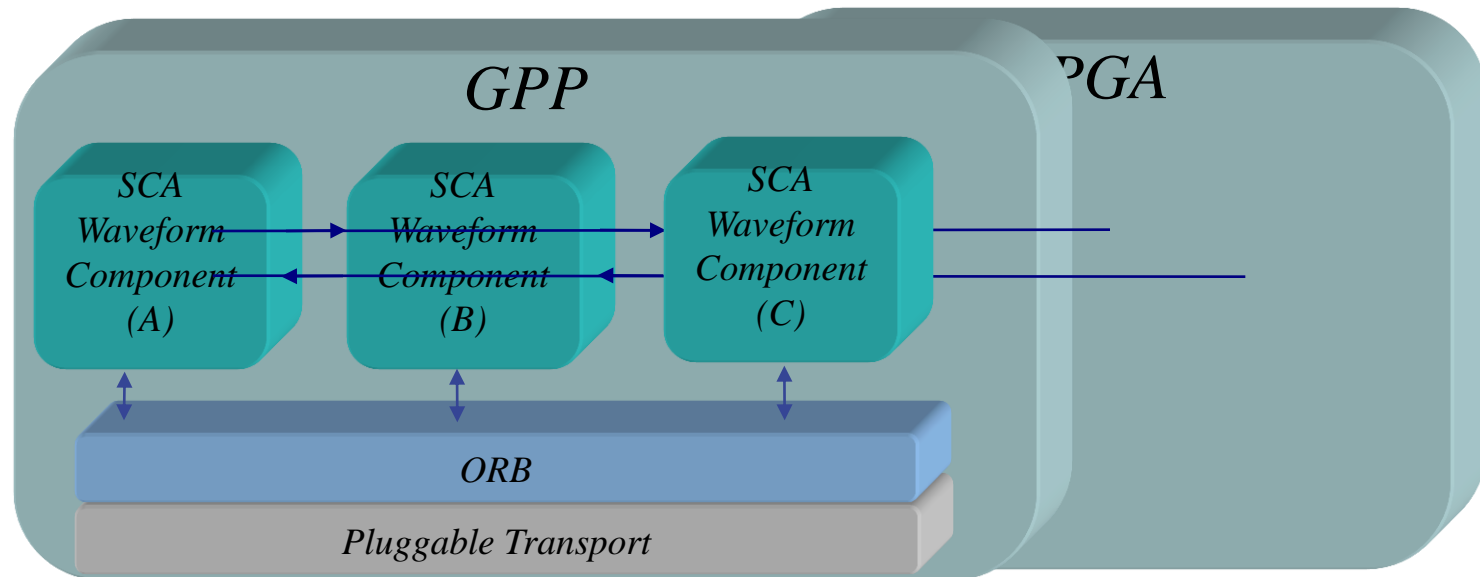


Performance Critical Middleware

Introduction

- ▶ FPGAs are a key enabling technology in SDR
- ▶ Typically used to perform IF up/down conversion and crypto functions
- ▶ Today's generation of more powerful FPGAs can also support DSP baseband and GPP processing tasks
- ▶ Key benefits of FPGAs in SDR include:
 1. Reduced time to market for new applications when compared to ASICs
 2. Easily re-configurable
 3. Hardware parallelism can exceed DSP processing power
 4. NRE costs of custom ASIC design far exceeds FPGA solutions
 5. More reliable than software
 6. Lower power solution possible

- ▶ To seamlessly integrate waveform logic running on the FPGA... while still maintaining SCA compliance





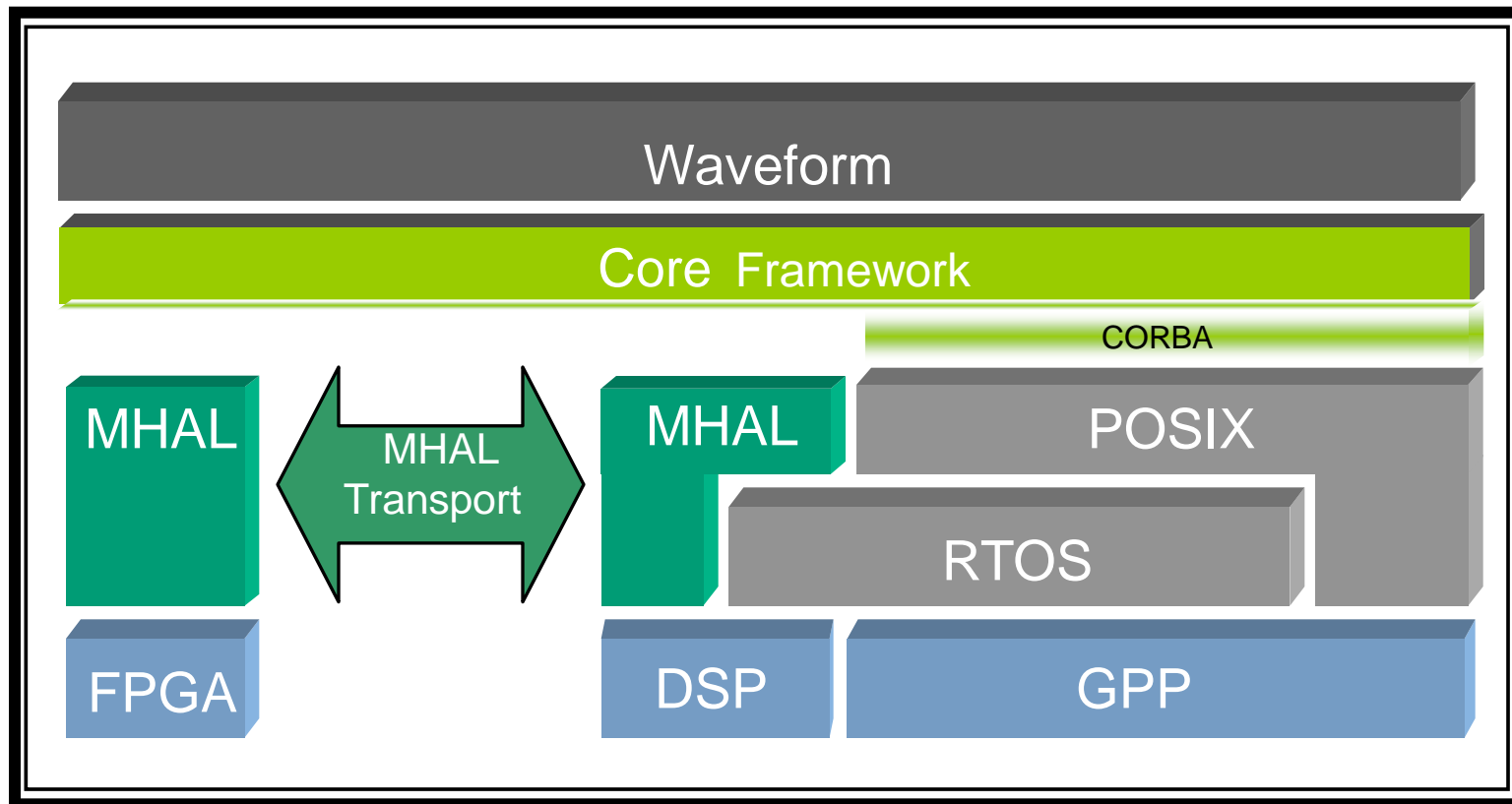
Performance Critical Middleware

SCA Device Model and MHAL

- ▶ “*..waveforms shall use the MHAL Communications Service for all data and control flowing between software components residing in different Computational Elements where at least one CE does not support CORBA...*”

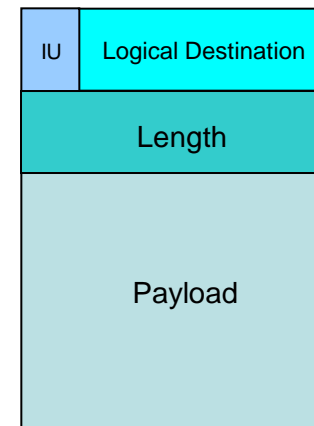
*Reference: Joint Tactical Radio System (JTRS) Standard Modem
Hardware Abstraction Layer Application Program Interface (API)
Version: 2.11.1, 02 May 2007*

- ▶ PrismTech interpretation....CORBA Everywhere would be optimal, if this were available.....
- ▶ This is about to change – next revision of the SCA “SCA Next” will formally adopt a CORBA profile for DSPs and work ongoing for a lightweight CORBA profile for FPGAs



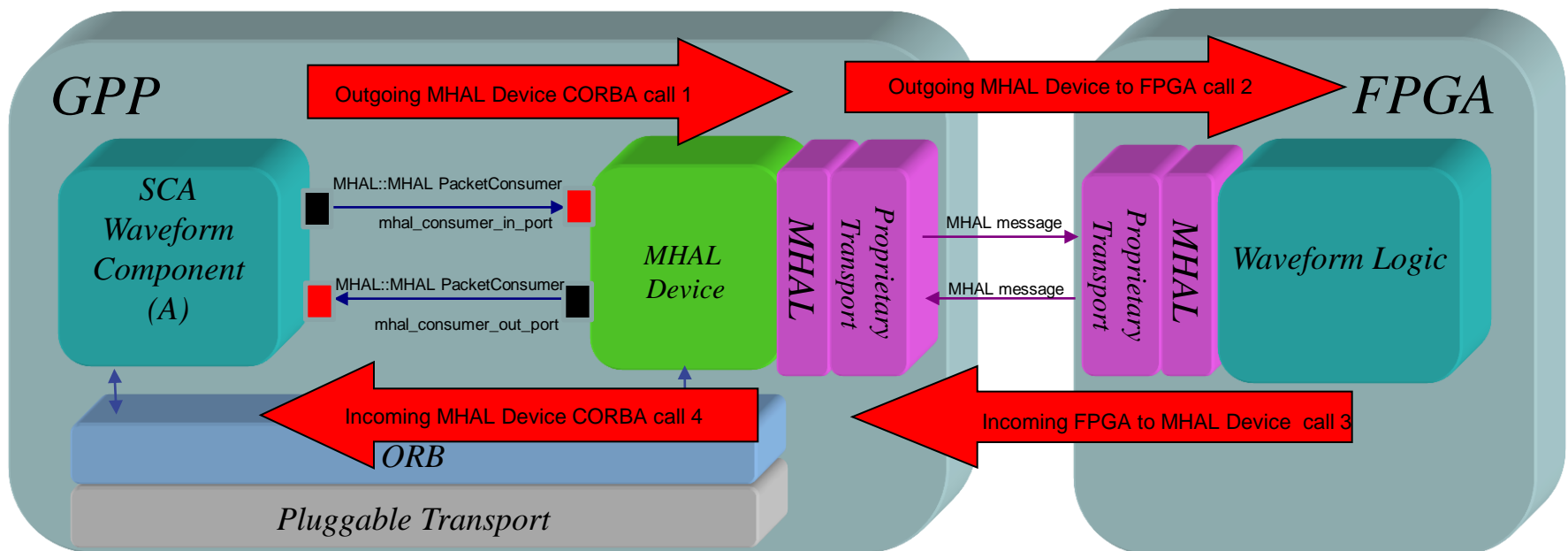
MHAL approach provides a degree of portability, however, the format and content of messages sent to the MHAL components is not standardised and must be written by each waveform developer

- ▶ MHAL has been adopted and standardised by the JTRS program to move data to and from modem hardware
- ▶ MHAL interfaces are used for command, control and data messages
- ▶ Offers an alternative to CORBA when dealing with processor and bus technologies with no off the shelf CORBA support
- ▶ MHAL On Chip Bus (MOCB) standard provide a set of interfaces to support a shared memory architecture
- ▶ Issues:
 - ▶ Interface between components is defined as a simple stream
 - ▶ The “on the wire” definition of the protocol is left to each developer
 - ▶ Interface semantics are captured in the protocol messages that travel over the stream
 - ▶ In order to isolate an assembly waveform component from the MHAL message oriented interface an “adaptor” or proxy” is often used



Standard
MHAL Message
Structure

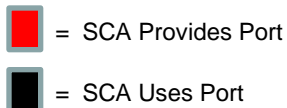
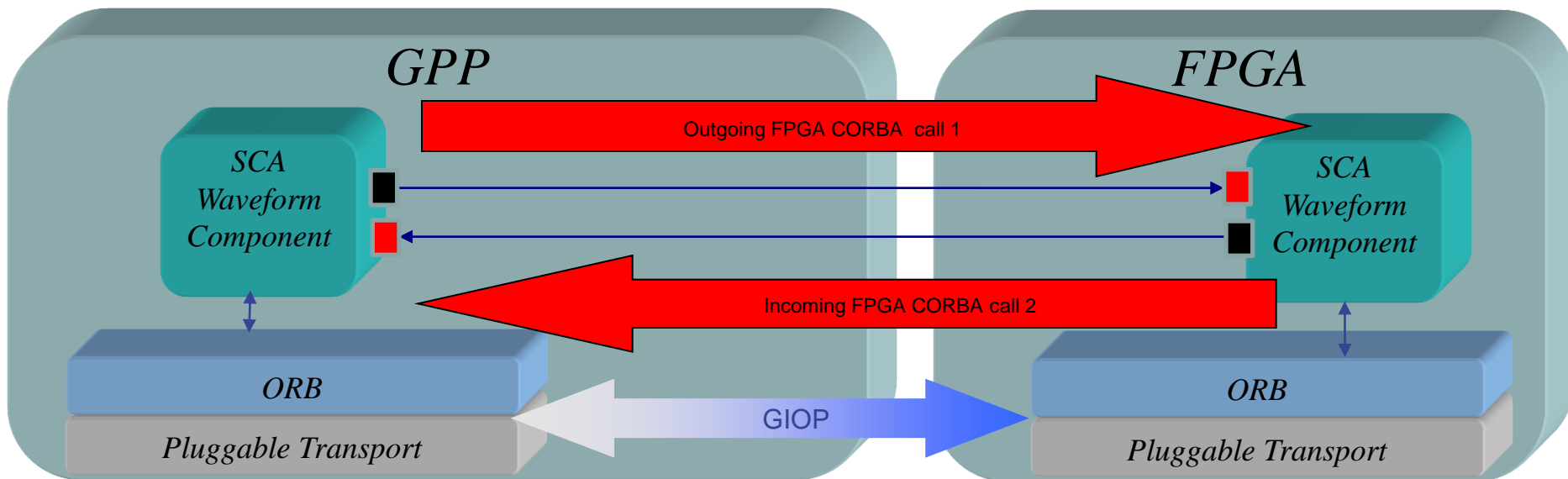
- Attempts to implement MHAL have resulted in added complexity for the radio developer and increased overall system latency
- Using MHAL Device approach requires a double call hop for both outgoing (sink) and incoming (source) calls – 4 calls instead of optimal two

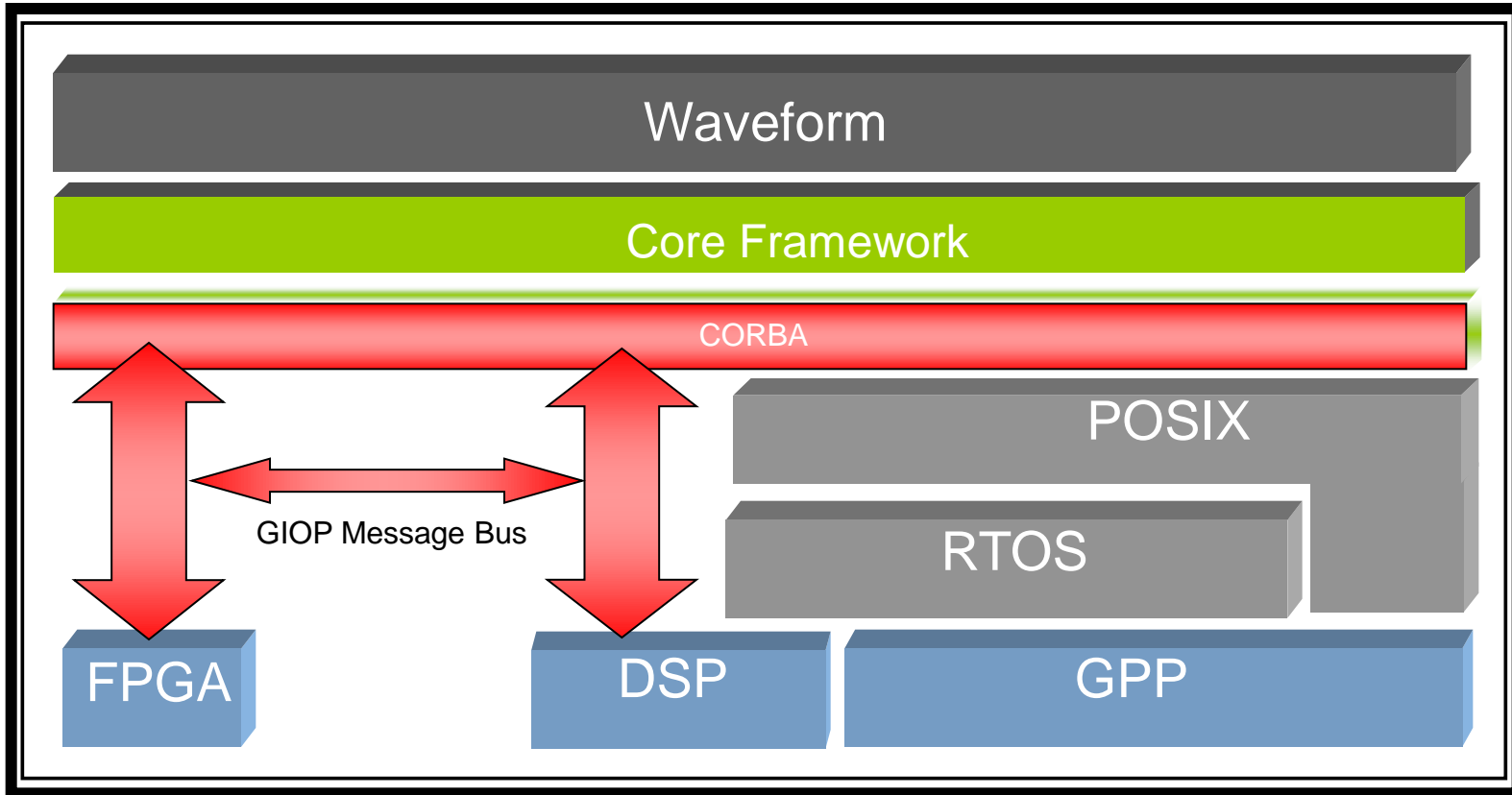


What if...

11

- ▶ Using an ORB, SCA compliance is maintained and overhead is reduced
- ▶ Now only requires single direct call for both outgoing and incoming messages – 2 calls instead of previous 4



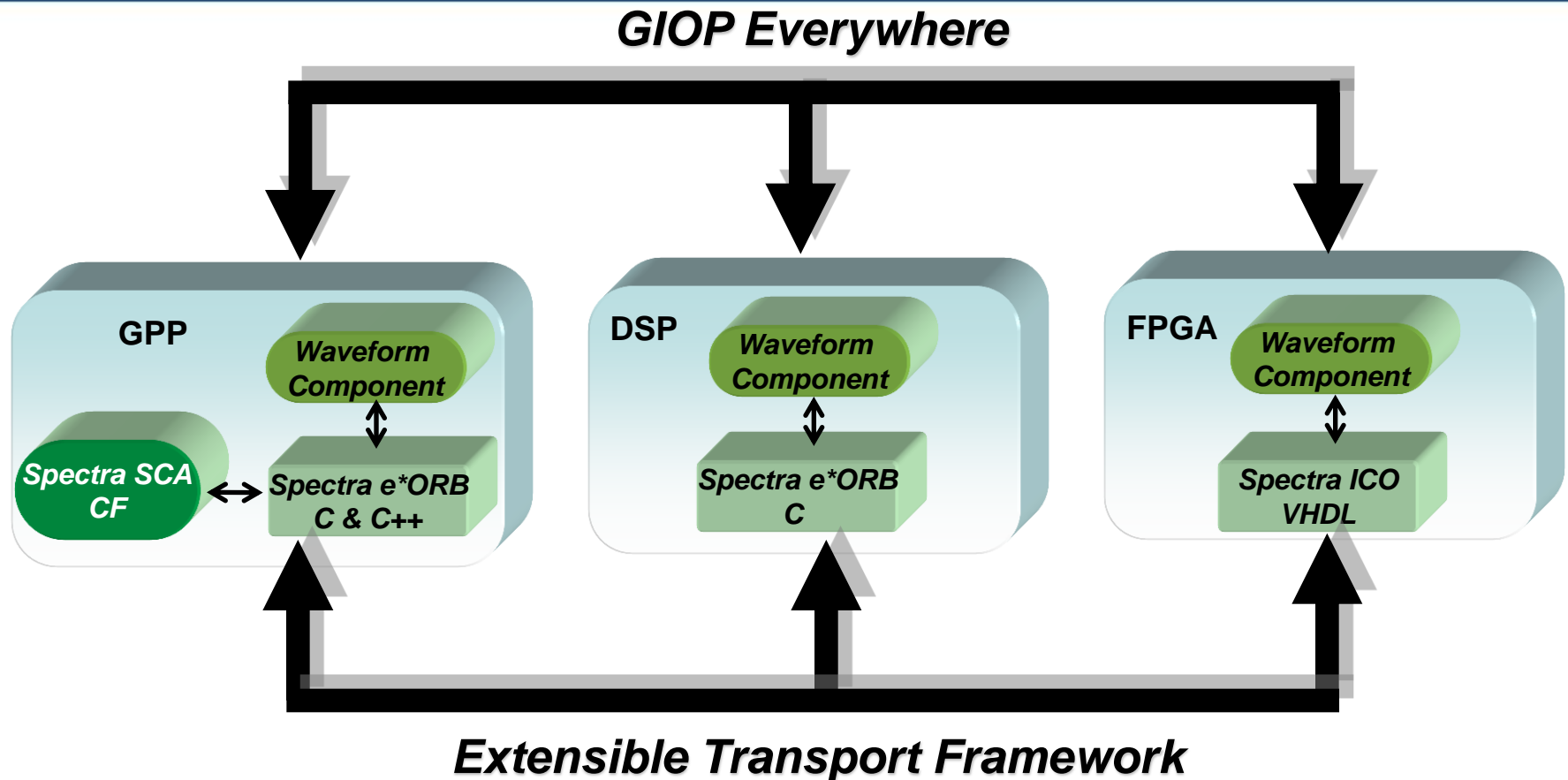


Standardized CORBA interfaces across signal processing chain



Performance Critical Middleware

CORBA Everywhere



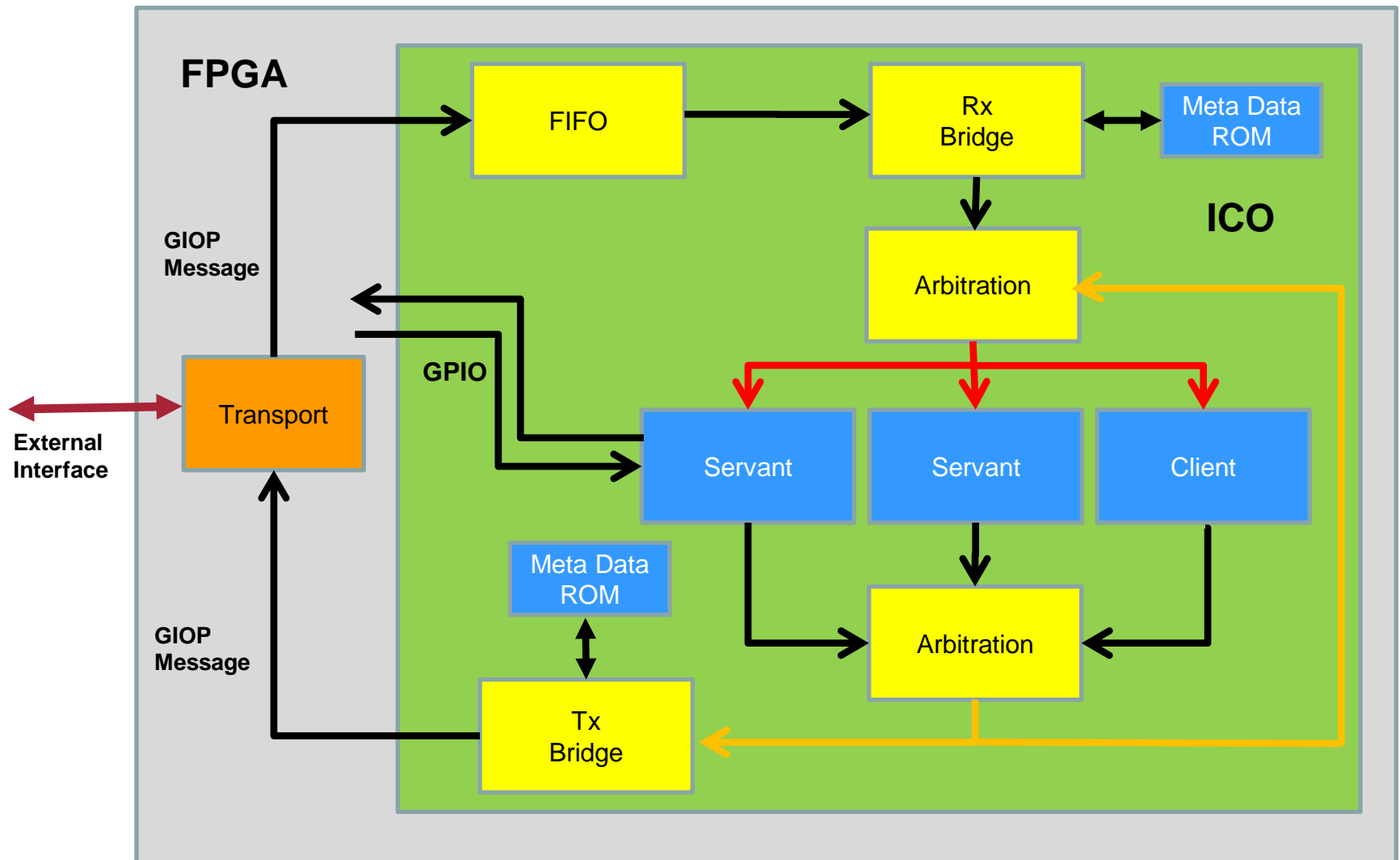
By leveraging CORBA a **standards-based**, high-performance, low-footprint, fully-interoperable COTS middleware solution that can be deployed across multiple processor types, including GPP, DSP, & FPGA environments

- ▶ Once all SDR processors (GPP, DSP and FPGA) are CORBA enabled, a number of potential benefits can be realised:
 - ▶ Reduce overall system complexity and improve time-to-market for new waveform applications and also legacy waveform porting
 - ▶ Support waveform component location transparency making it much easier to re-locate waveform components across processors
 - ▶ Eliminate the need for proprietary communication protocols reducing complexity and improving waveform portability
 - ▶ Remove the need to use adaptor patterns in combination MHAL, therefore reducing communication latency and improving throughput



Performance Critical Middleware

Spectra IP Core ORB (ICO)



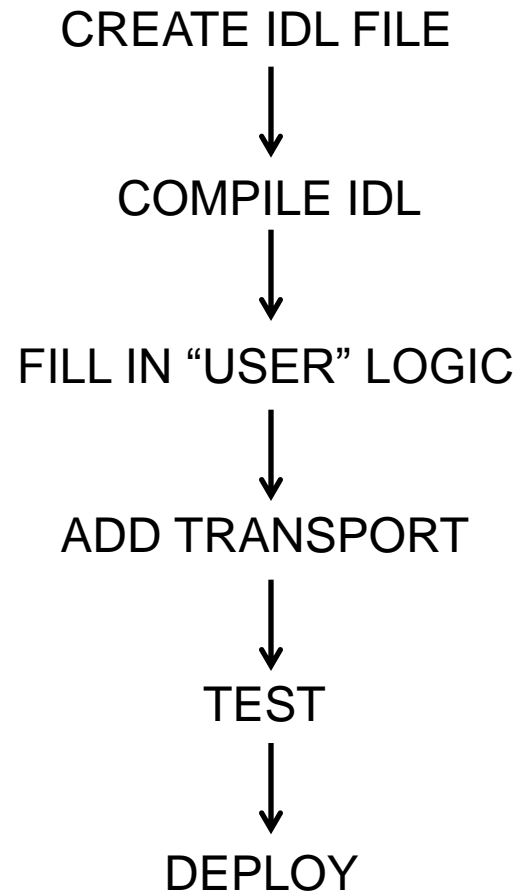
- ▶ Direct mapping of CORBA primitive types to VHDL
- ▶ The mapping requires that a bus-based architecture is used
- ▶ The bus must support the concept of data and addressing
- ▶ The mapping defines a protocol called Bus Interoperability Protocol (BIOP)
- ▶ GIOP can be converted to BIOP and vice-versa

- ▶ Bus-based communication mechanism
- ▶ The protocol is designed to support requests and replies between hardware entities
- ▶ Data exchanged in the form of messages
- ▶ Three message types – Request, One-way Request and Reply
- ▶ Messages consist of a header, and optionally message data
- ▶ Request/reply data is placed on the bus with an address offset from the target entity's base address
- ▶ The offset is a constant generated according to the IDL-VHDL language mapping
- ▶ Operation parameter and reply data passed in GIOP CDR encoding order



Performance Critical Middleware

FPGA Design with ICO



```
module AnalogDigital
{
    interface DAC
    {
        void send_data(in unsigned longval);
    };

    interface ADC
    {
        long read_data ();
    };
};
```


Example servant –DAC Interface

23

```
case AddressOffset(adr_i(AddressBusLow'range)) is
  when ADC_read_data_request =>
    v.request_id := unsigned(dat_i);

  when ADC_read_data_replyaddr =>
    v.reply_addr := dat_i(AddressBusHigh'range);

  when ADC_read_data_request_end =>
    case r.reply is
      when normal =>
        v.state := ADC_read_data_reply_state;
      when others =>
        v.state := request_state;
    end case;

  when DAC_send_data_request =>
    v.request_id := unsigned(dat_i);

  when DAC_send_data_replyaddr =>
    v.reply_addr := dat_i(AddressBusHigh'range);

  when DAC_send_data_val =>
    -- Modify the following line as needed.
    null;
  when DAC_send_data_request_end =>
    case r.reply is
      when normal =>
        v.state := DAC_send_data_reply_state;
      when others =>
        v.state := request_state;
    end case;

  when others =>
    null;
end case;
```

```
when ADC_read_data_return_state =>
    v.address := mkaddr(r.reply_addr, ADC_read_data_return);
    if bus_grant = '1' then
        v.state := ADC_read_data_reply_end_state;
        -- Modify the following line as needed.
        v.data := (others => '0');
        v.we := '1';
    else
        null;
    end if;
end process;
```



Performance Critical Middleware

ICO CORBA Profile



- ▶ ICO v2.0 provides support for the following key features:
 - ▶ Supports GIOP version 1.0 protocol
 - ▶ Processes incoming CORBA requests
 - ▶ One way operations
 - ▶ Two way operations
 - ▶ Support for CORBA clients and servers
 - ▶ Clients can be internal to the FPGA written in VHDL or external to FPGA(e.g., on a GPP or DSP) implemented by a conventional software application
 - ▶ Servants implemented on FPGA in VHDL
 - ▶ No arbitrary restriction on the number of clients and servers that can be supported on the FPGA

▶ IDL compiler support

- ▶ Supports IDL to VHDL language mapping and will auto generate VHDL equivalent of CORBA stubs and skeletons allowing ICO to be easily connected to servants implementing waveform logic
 - ▶ Based on CORBA 3 grammar, but only supporting a subset of data types and constructs
 - ▶ Supports main IDL keywords - Import, Module, Interface, Forward, Constant, Attribute
 - ▶ Supports CORBA directional parameter types- Void, Return, In, Out, InOut
 - ▶ Simple data types - Char, Octet, Boolean, Short, Unsigned Short, Long, Unsigned Long, Long Long, Unsigned Long Long, String
 - ▶ Enumerated Types
 - ▶ Complex data types — Struct, Sequence
 - ▶ CORBA exceptions support
 - ▶ User exceptions
 - ▶ System Exceptions

- ▶ Pluggable and open transport interface allows user-defined custom transports to be plugged into ICO
- ▶ Written in pure VHDL and completely portable across FPGA devices
- ▶ Available on both Altera and Xilinx FPGAs, including:

FPGA Model	Board	Tool Chain
Altera Stratix II	Altera Eval	Modelsim ALTERA 6.5b + Quartus II v9.1
Altera Stratix III	Altera Eval	Modelsim ALTERA 6.5b + Quartus II v9.1
Altera Stratix IV	Altera Eval	Modelsim ALTERA 6.5b + Quartus II v9.1
Altera Cyclone II	Altera Eval	Modelsim ALTERA 6.5b + Quartus II v9.1
Altera Cycone III	Altera Eval	Modelsim ALTERA 6.5b + Quartus II v9.1
Xilinx Spartan 3	Xilinx Eval	ISE 13
Xilinx Spartan 6	Monsoon	ISE 13
Xilinx Virtex Pro IV	Pro 4600	Modelsim Xilinx Edition III + ISE 12

 = available now
 = planned

- ▶ SCA 2.2.2 assumes that an FPGA is a non-CORBA capable Computational Element (CE)
- ▶ SCA Next will standardize a CORBA profile for DSP (Lightweight profile based on CORBA/e) reflecting availability of lightweight ORB implementations for DSPs
- ▶ Through the SCA Next working group at the WINNF the ESSOR consortium have recommended that SCA Next formalises IDL subset for use with FPGAs
- ▶ Unlikely that additional SCA standardization for an FPGA CORBA profile until will happen without OMG standardisation first
- ▶ PrismTech with support from our ICO customers are in discussions with the OMG on new areas of FPGA CORBA standardization, specifically:
 - ▶ An IDL to VHDL language mapping – describing the subset of IDL language that is supported and its mapping to VHDL + BIOP
 - ▶ A CORBA profile for FPGA – describing the subset of CORBA functionality that should be supported by a compliant implementation

- ▶ MHAL supports goal of SCA application portability but with limitations
 - ▶ MHAL forces hardware engineers focus on low level message format/protocol details
 - ▶ Routing messages via MHAL Devices adds additional latency into communications path
- ▶ ICO supports direct access to SCA components running on H/W
- ▶ ICO enables vision of SCA architectural consistency across all aspects of the SDR
- ▶ ICO eliminates the need for MHAL improving application portability
- ▶ ICO eliminates the need for S/W proxies/adapters (i.e., MHAL Device) - reducing call latency
- ▶ ICO helps reduce time to market for new applications by simplifying FPGA integration task

Thank you for Listening

31

- ▶ For additional information on PrismTech's Spectra products and services:
 - ▶ E-mail:
 - ▶ info@prismtech.com
 - ▶ Website:
 - ▶ www.prismtech.com/spectra
 - ▶ Today's Presenter:
 - ▶ Andrew Foster, Spectra Product Manager
 - ▶ Email: awf@prismtech.com

Thank You