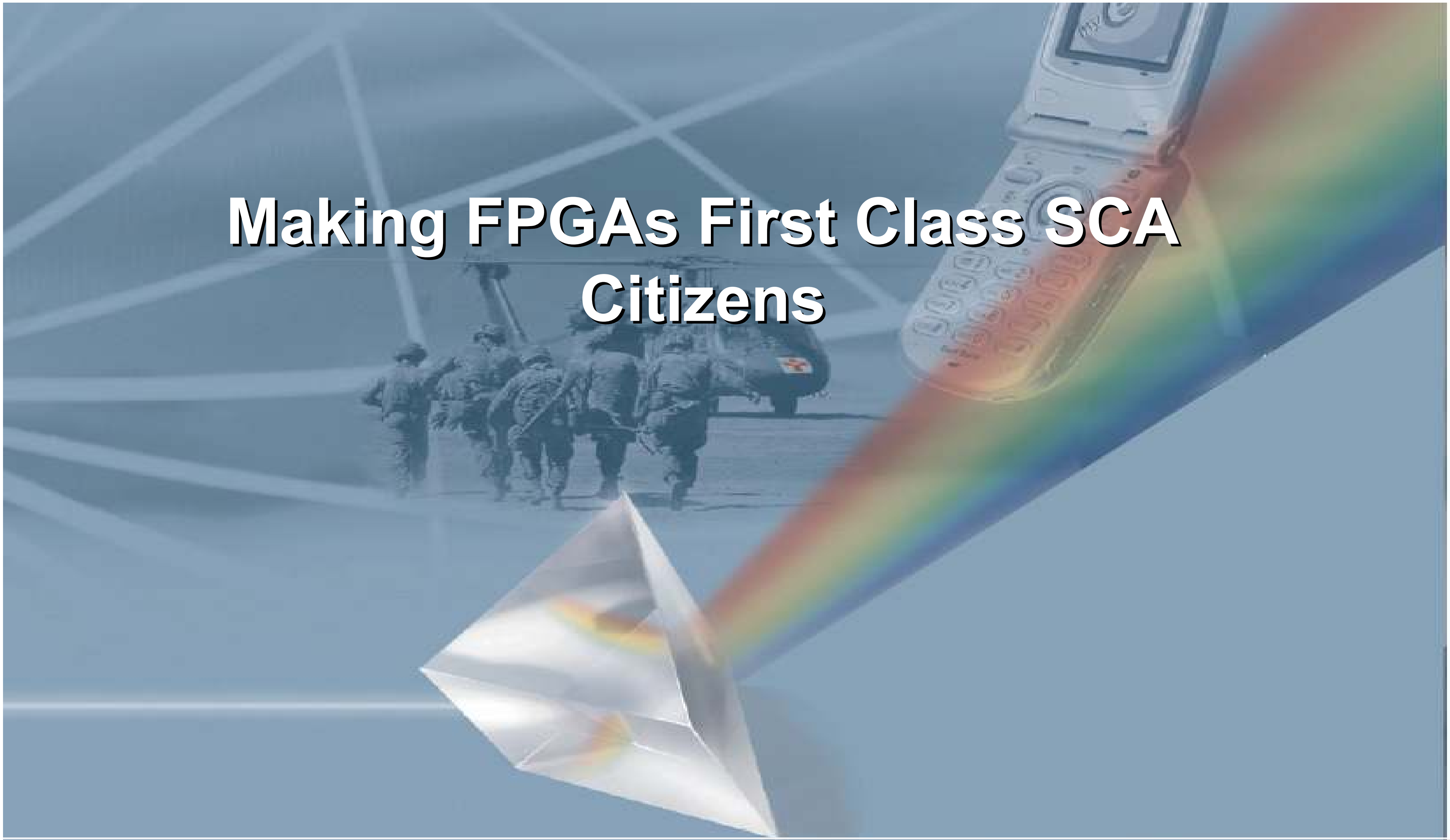
Productivity Tools & Middleware

PrismTech



Making FPGAs First Class SCA Citizens

Spectra Software Defined Radio Products

SDR/SCA Overview

- SDR is gaining ground in military and commercial markets
- SDR adoption is being driven by the SCA
  - The SCA is a mature specification
  - SCA compliance is being required on more and more military platforms
  - Companies using the SCA are starting to report that it does save waveform porting time
- The industry is moving closer to producing the first true software defined radios
- Adoption has been slowed by some preconceptions and misconceptions about the SCA that may no longer be valid

SDR/SCA Misconceptions

- SCA operating environment components too large
  - They take up valuable system resources such as memory
- CORBA is too low in performance
  - The TCP/IP stack adds too much overhead for simple data transfer
- The SCA is complex and difficult (expensive) to learn and utilize
- There is no SCA solution for hardware objects
  - waveform components implemented on silicon devices such as FPGAs and ASICs

We’ve Heard This Before

- These generic problems are similar to issues encountered when moving to a higher level of abstraction
  - Assembly to C
  - Schematic capture to VHDL
- How did C/VHDL overcome these obstacles??
  - A strong driver to push it along
  - Enablers to make it practical
- The Driver was a strong business case based on portability
  - Write C once and run it on many machines
  - Write VHDL once and synthesize it on many platforms
- The enablers were technology breakthroughs

Generic Enablers

- increases in processor performance
- increases in memory size and density
- smaller software footprint
- more efficient software processing tools
  - compilers, synthesizers
- high level tools to remove complexity from user
  - operating system, emulation and debugging environments
  - Simulators, automatic place and route

SDR/SCA Enablers

- Small footprint embedded ORBs for GPPs and DSPs
  - ORBs that require KBs of memory compared to MBs of memory
- Small footprint second generation OE
  - An entire SCA OE middleware suite consisting of ORB, ORB services and core framework require less memory than ORBs of just a few years ago
- Domain specific SDR development tools
  - Allows development at higher levels of abstraction
  - Provides automatic SCA code generation
  - Reduces the complexity of the SCA
- CORBA enablers implemented in hardware
  - CORBA/SCA communications are brought directly onto hardware platforms such as FPGAs and ASICs

A NEW ERA

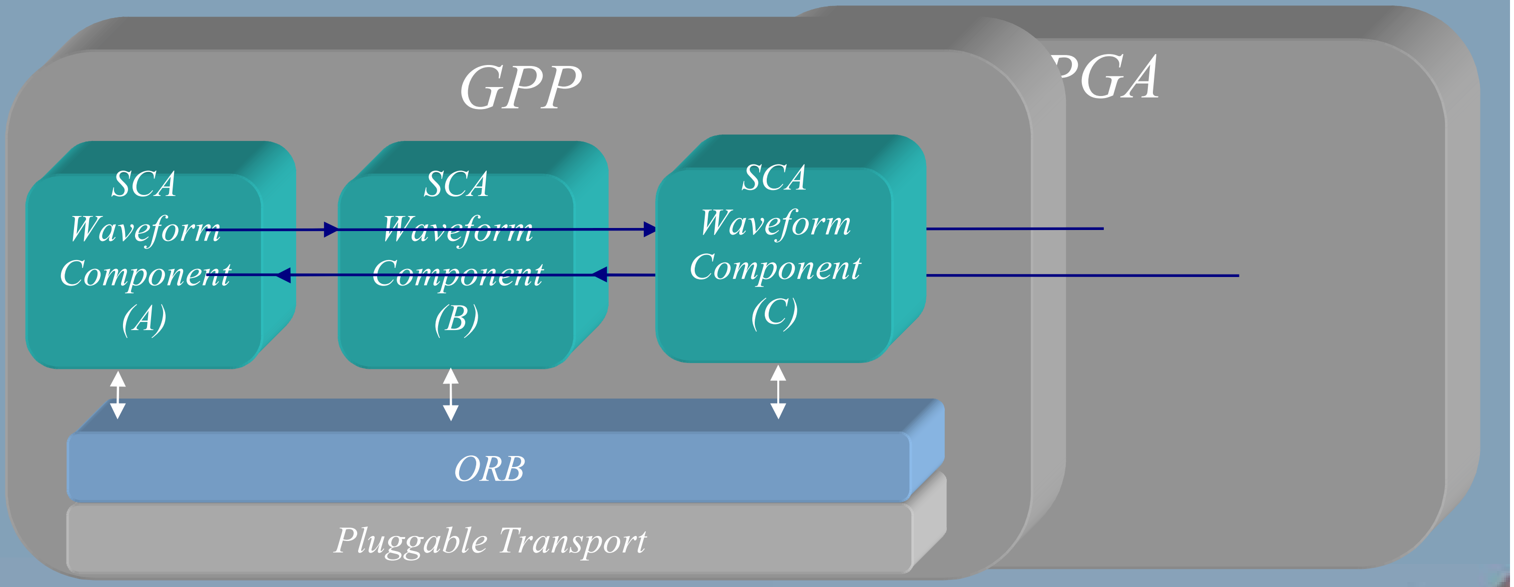
- The strong business case for the SCA and the development of technological enablers have brought SDR to the edge of a new era of industry wide acceptance
- The industry now has the critical innovations needed to allow ubiquitous SDR adoption in military and commercial markets

A Key Enabler

- The goal
  - Realize the cost savings of portability by moving the SCA as close to the antenna as possible.
- Small software footprint allows the SCA to run on embedded GPPs and DSPs
- The next obstacle for the SCA to cross lies in the interface between embedded processors and the digital radio hardware
- A key enabler is needed to bring the SCA to waveform components implemented on silicon devices

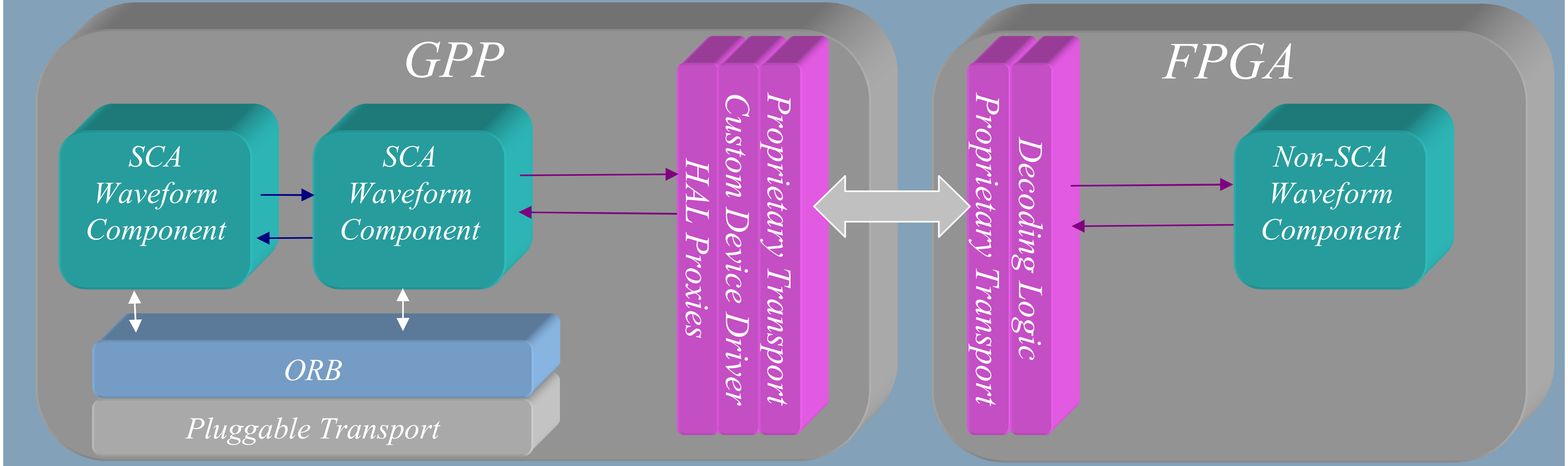
What We Want to Do

- Migrate certain waveform components into FPGA implementations… while still maintaining SCA compatibility


GPP
PGA
SCA Waveform Component (A)
SCA Waveform Component (B)
SCA Waveform Component (C)
ORB
Pluggable Transport

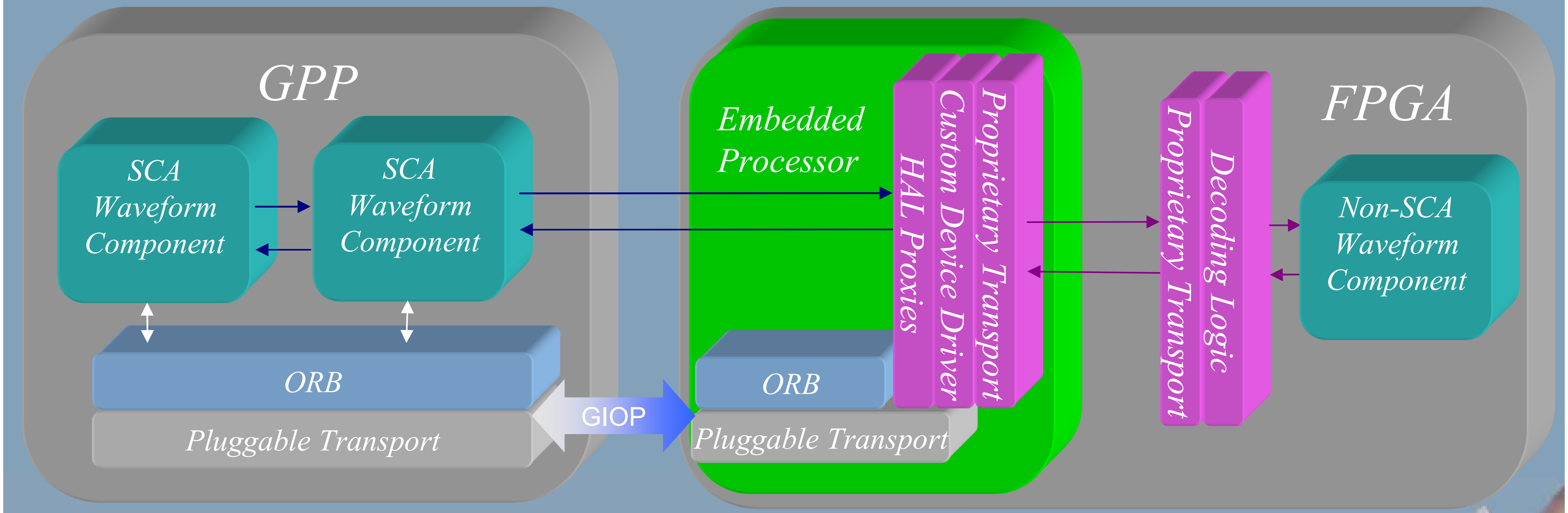
Historical Approaches

- Attempts to implement this resulted in non-SCA compliant components in the FPGA and added complexity for the radio developer



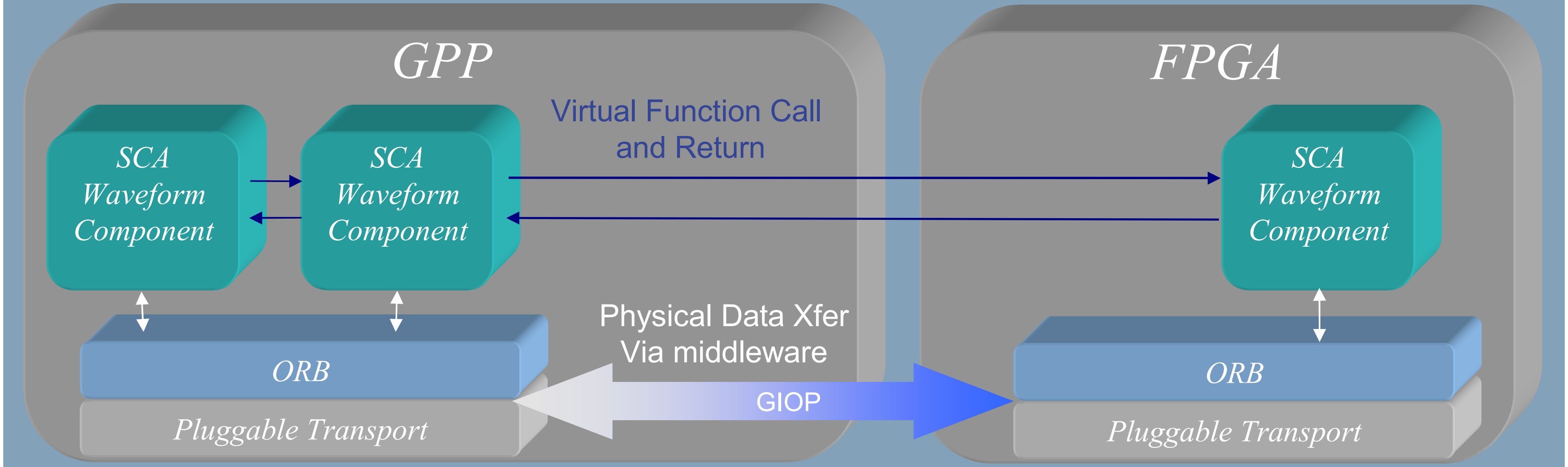
Other Attempts

- Other attempts to implement this have only moved the problem internal to the FPGA. The result is still non-SCA compliant components in the FPGA and a likley *increase* in added transport overhead.



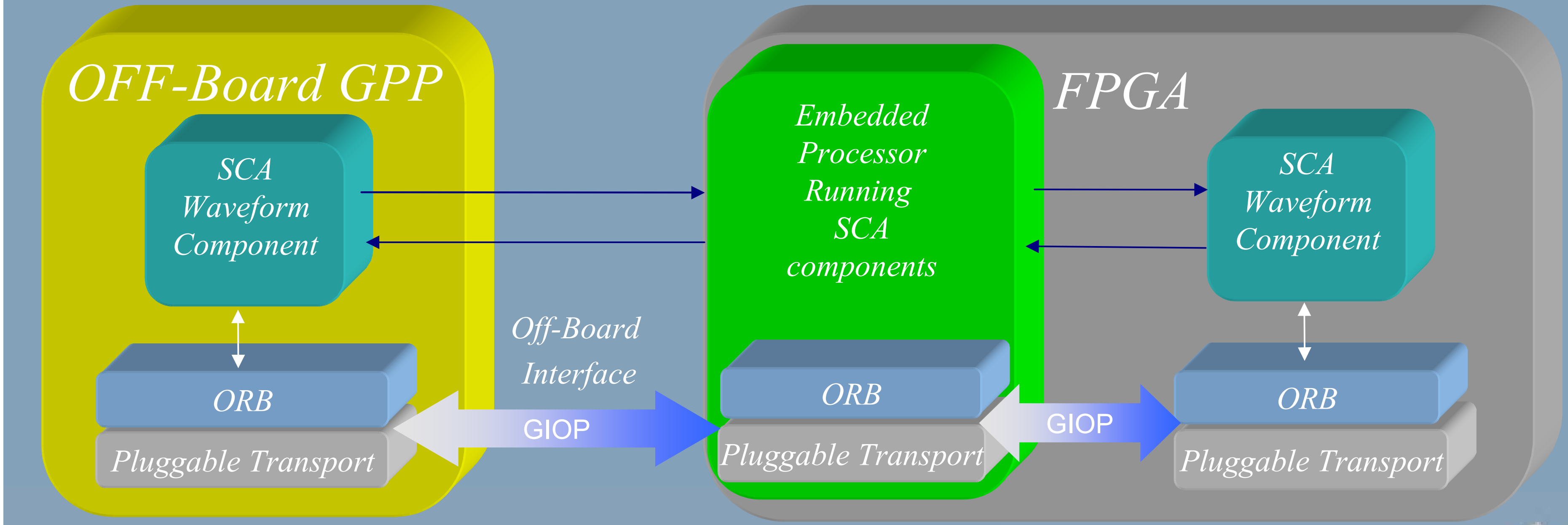
What if…

- Using an ORB, SCA compliance is maintained and overhead is reduced



What about Single Chip Solutions

- For System on a Chip (SoC) solutions, both the embedded processor and off-board processors can communicate, via an ORB with waveform components implemented in hardware; thus, SCA compliance is maintained.

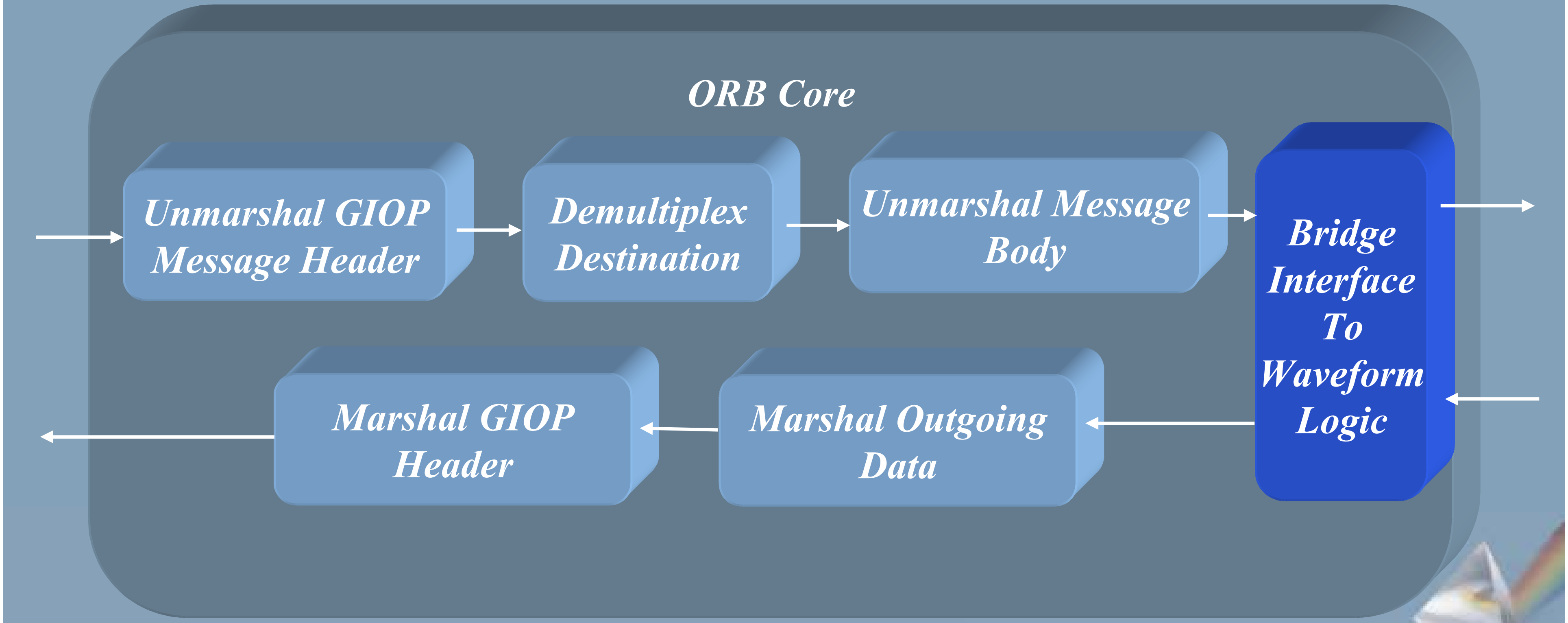


What Does A Hardware ORB need to look like?

- Can’t be as complex as a software ORB
  - No need to implement the entire CORBA standard
  - All data types are not necessary to support SDR development
- Can’t be as large as a software ORB
  - Can’t just take software ORB and turn it into VHDL
  - Must minimize gate count
- Can’t be as slow as a software ORB

What’s In a Hardware ORB?

- It is a Generic Inter-ORB Protocol (GIOP) processor consisting of several different functional blocks



PrismTech’s ICO

- PrismTech has developed a hardware implementation of an ORB
- It is called the Integrated Circuit ORB (ICO)

ICO

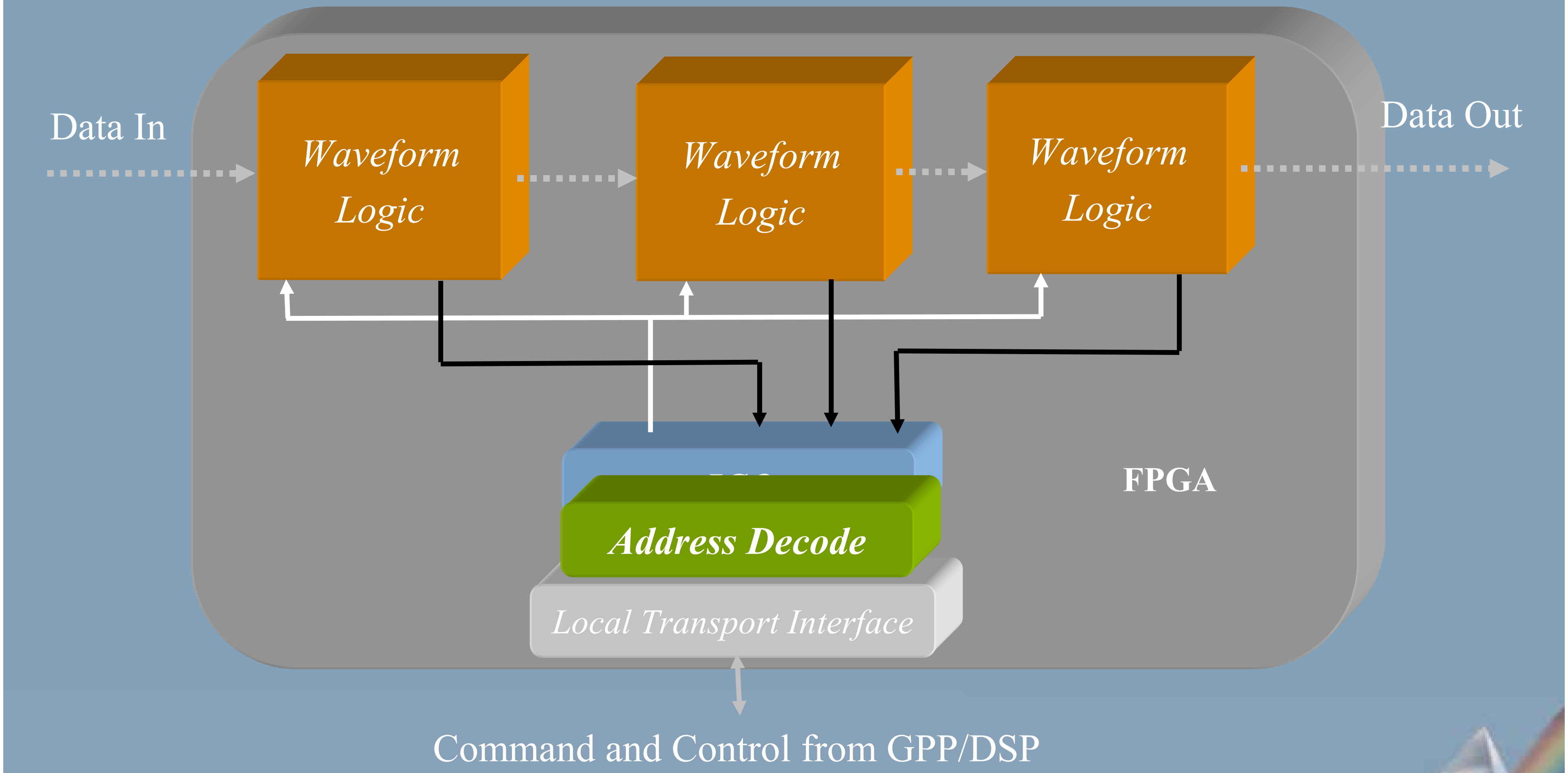
Productivity Tools & Middleware

PrismTech

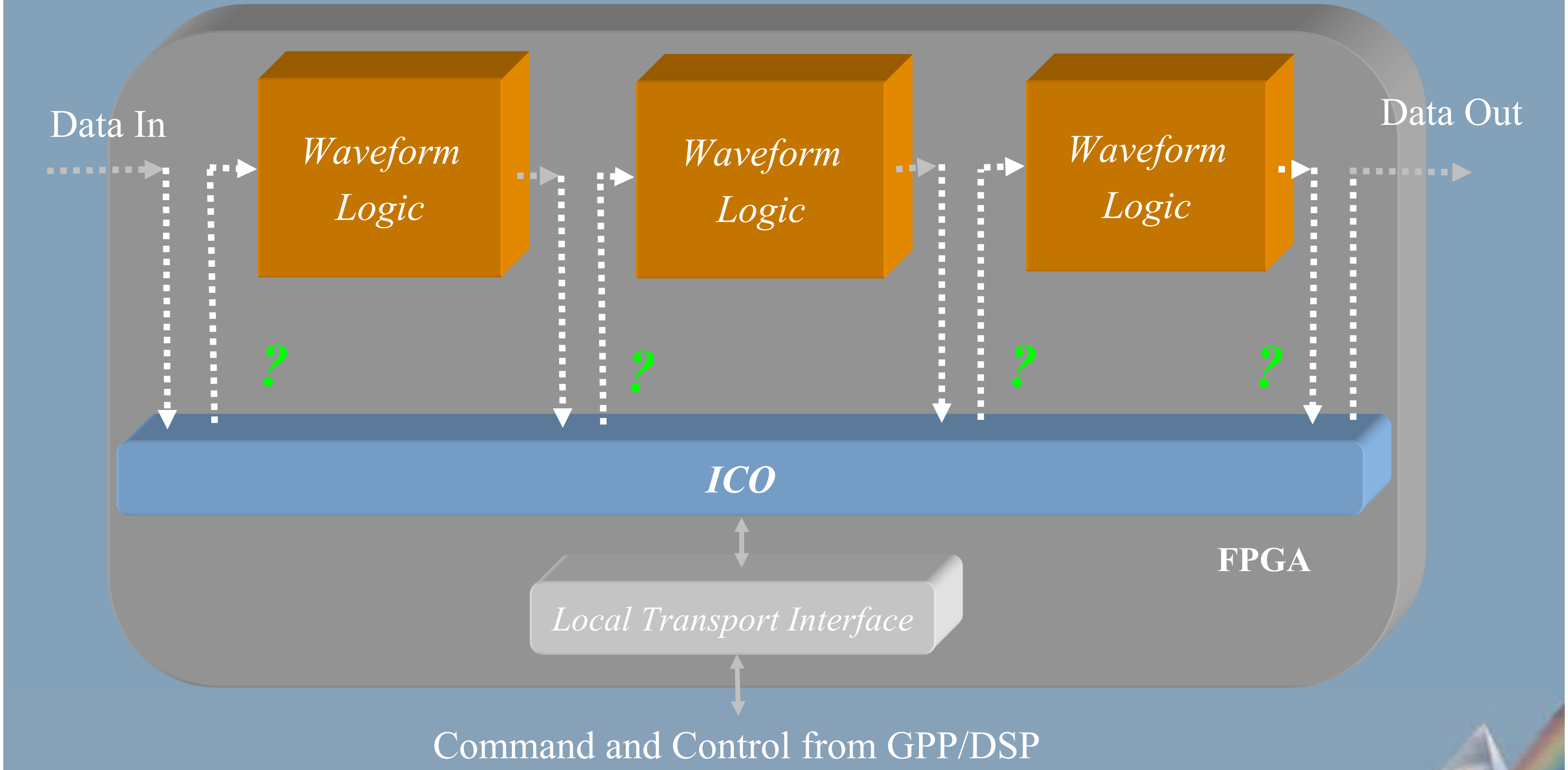
ICO In the FPGA Design

Spectra Software Defined Radio Products

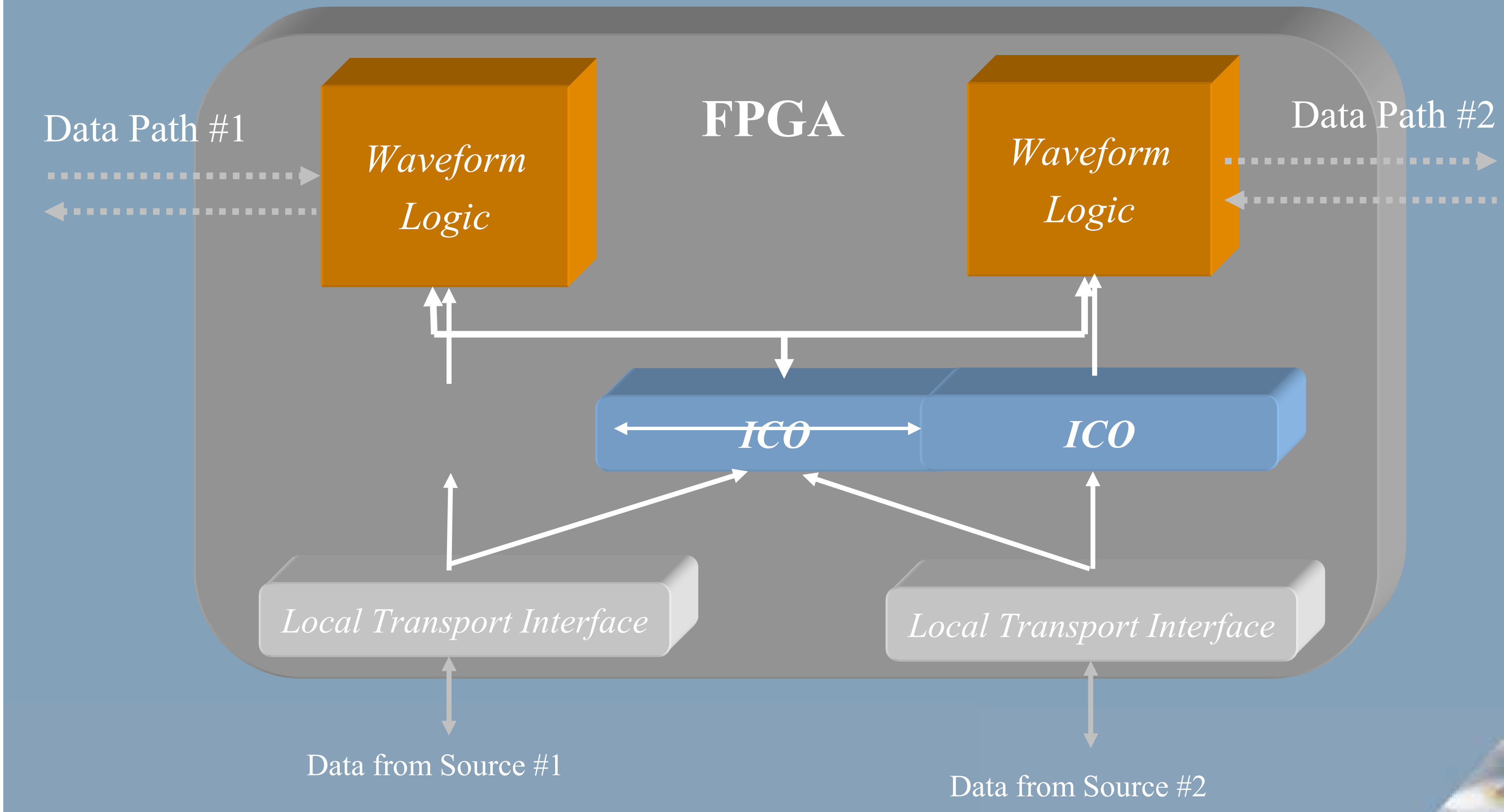
ICO FPGA Data Flow Example



ICO and Inter/Intra Chip Communication



ICO and Parallel Processing



ICO In FPGA Summary

- ICO may be used for inter and intra board communications
- System performance requirements will determine the nature of its deployment in FGPA designs
  - The data speed
  - The burstiness of the data
  - The ultimate portability of the interface
- The system considerations are analogous to Assembly vs. C language choices
  - Assembly is still used where high performance is critical and C is used for portability

Productivity Tools & Middleware

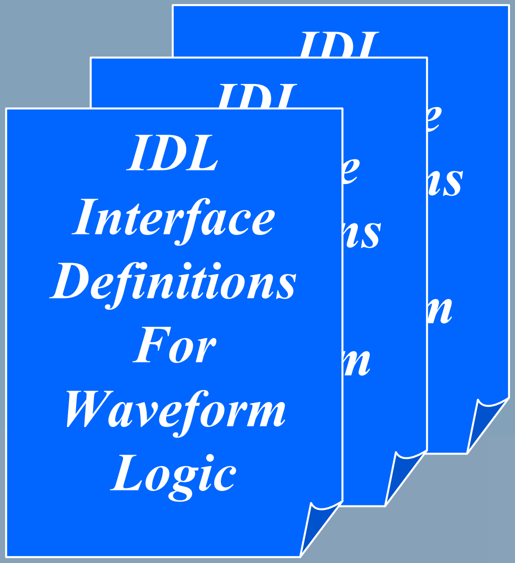
PRISMTECH

The ICO Design Flow

Spectra Software Defined Radio Products

ICO Design Flow

- The radio developer defines component to be implemented in H/W
- IDL definitions of the component interfaces are made

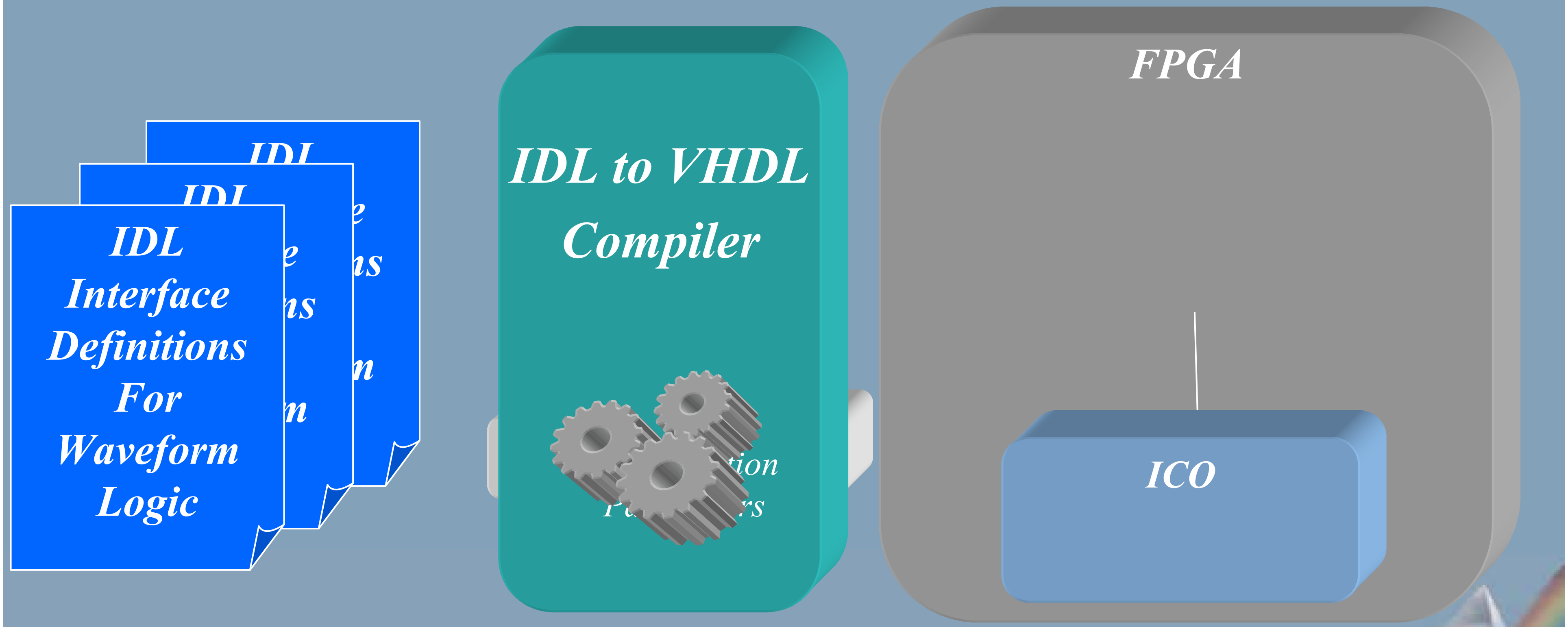


CORBA Data Types Supported by ICO

- CORBA data types supported by ICO
  - Simple Data types
    - Char, octet,
    - Unsigned short
    - Unsigned long
    - Unsigned long long
  - Complex Data types
    - Strings of simple types
    - Sequence of simple types
    - Any of simple type
    - Structures of simple types, strings and anys
- CORBA standard supports additional complex data types not typically needed in FPGA processing.

ICO Design Flow

- The IDL to VHDL compiler generates configuration parameters and a VHDL binding for the ICO



Productivity Tools & Middleware

PRISMTECH

Designing SCA compliant components using ICO

Spectra Software Defined Radio Products

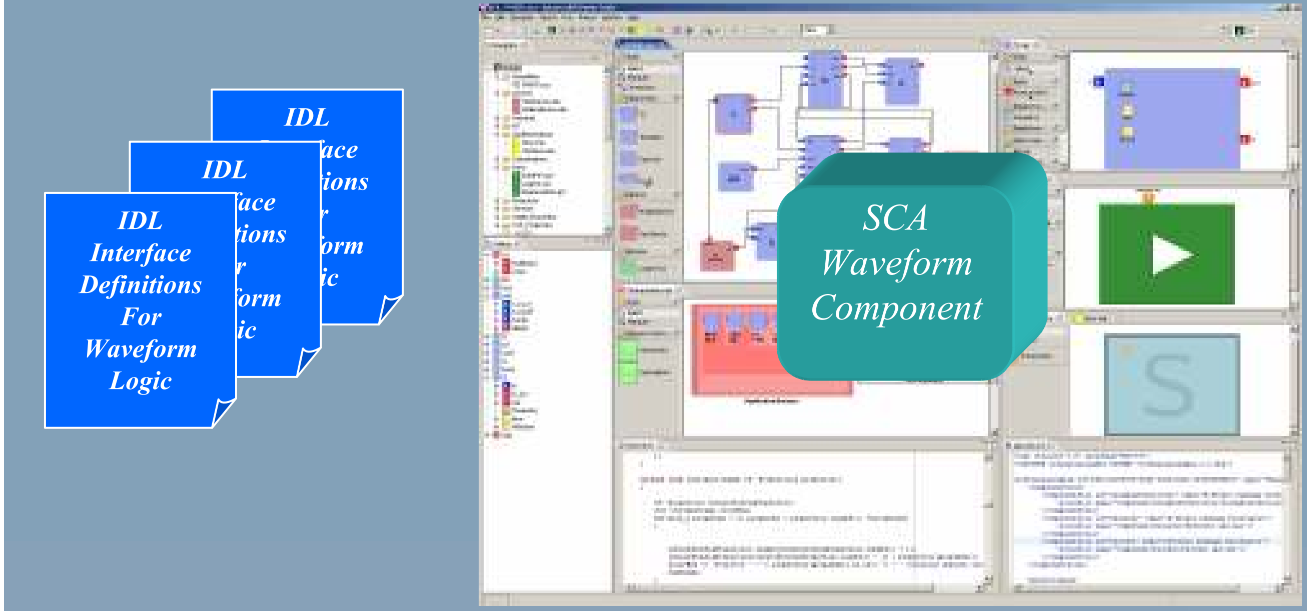
ICO Support for the SCA

- ICO supports the CORBA constructs required for the SCA Resource and Device Interface
  - Testable Object
    - runTest
  - Resource
    - Start, Stop
  - Life Cycle
    - Initialize
    - Release Object
  - Port
    - Connect
    - disconnect
  - Property Set
    - Configure
    - Query
  - Port Supplier
    - Get(port)

SCA Devices

- The SCA executableDevice type is probably not appropriate for FPGAs as there is no code being executed
- The loadableDevice type must initially be handled by an external processor that stands in for ICO
- Reconfigurable FPGAs allow the loadableDevice component to reside on the FPGA and dynamically load other parts of the FPGA

- Modeling tools can be used to generate a VHDL SCA waveform component container



IDL Interface Definitions For Waveform Logic

SCA Waveform Component

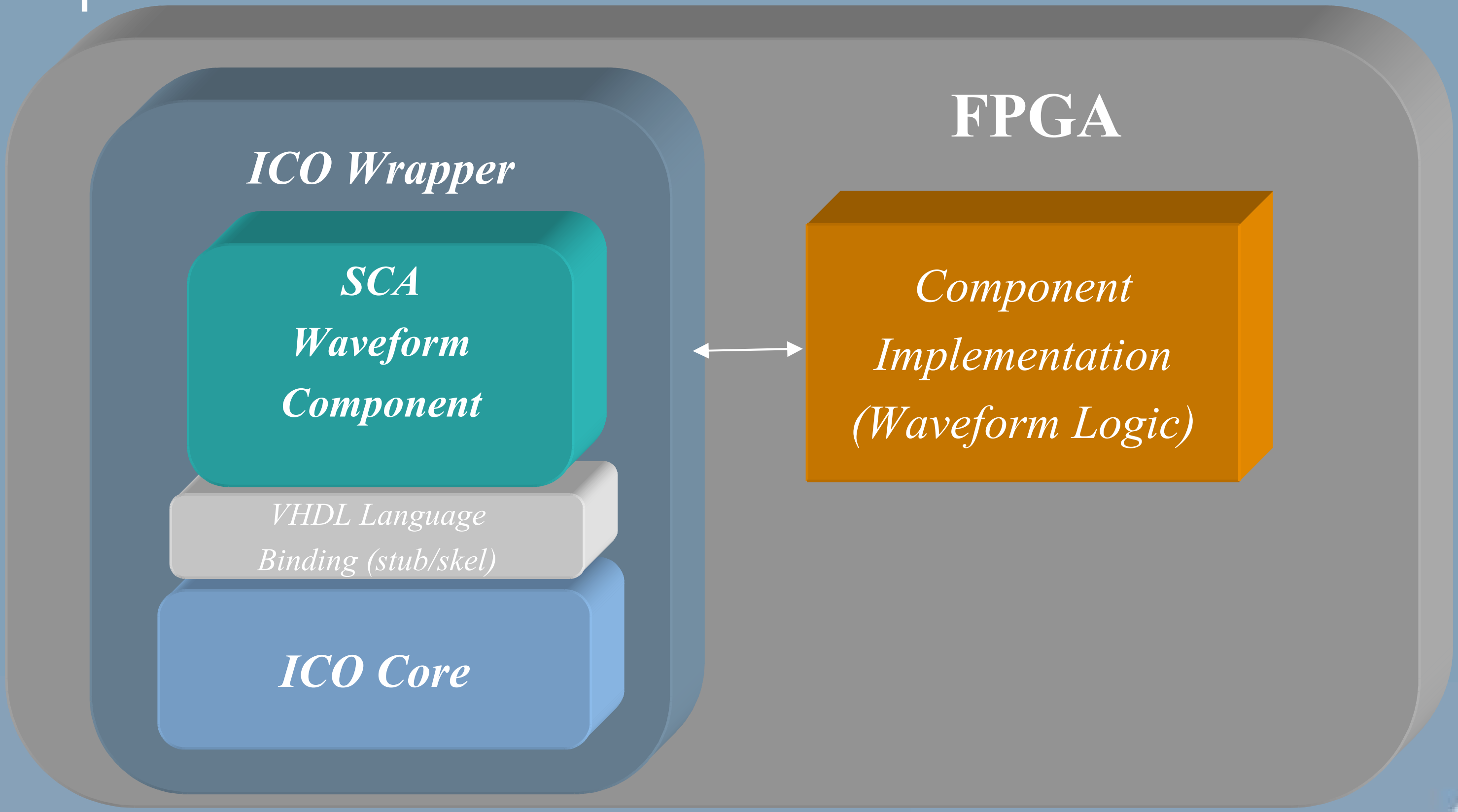
VHDL

Source

Code

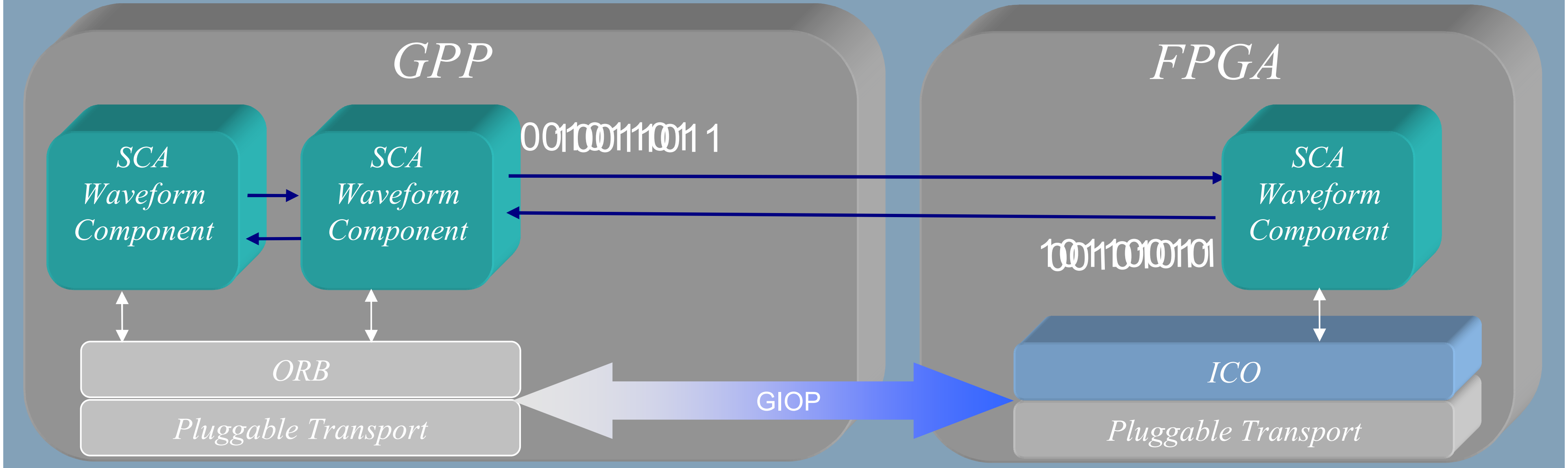
ICO Design Flow

- The embedded waveform logic is then connected to the ICO Wrapper which contains the VHDL SCA waveform component


FPGA
ICO Wrapper
SCA Waveform Component
VHDL Language Binding (stub/skel)
ICO Core
Component Implementation (Waveform Logic)

Using ICO…

- Providing seamless SCA compliance to the FPGA with all location transparency benefits of CORBA



Why use ICO?

- CORBA message processing is executed directly in H/W
  - 100x faster than in S/W
- Eliminates the need for S/W proxies/adapters on GPPs
  - Reduces overhead, latency
  - Increases throughput
- Eliminates the need for complex hardware abstraction layer protocols
- Supports direct access to SCA components running on H/W
  - Modeling tools can auto-generate VHDL delivering SCA "components" in H/W
- Applications in security-related areas where the assurance of large software applications (such as ORBs) is suspect
- Supports vision of SCA architectural consistency across all aspects of the SDR

Contact Info

Fred Humcke
fh@prismtech.com