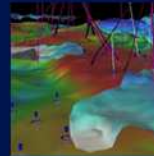


Computer Systems, Inc.  
**MERCURY**

*Challenges Drive Innovation™*



# Component Portability Infrastructure

*Building Waveform-Ready™ SATCOM Terminals*

**Jim Kulp, Consulting Software Architect**

# CPI Provides Waveform Portability



## SCA Compliance

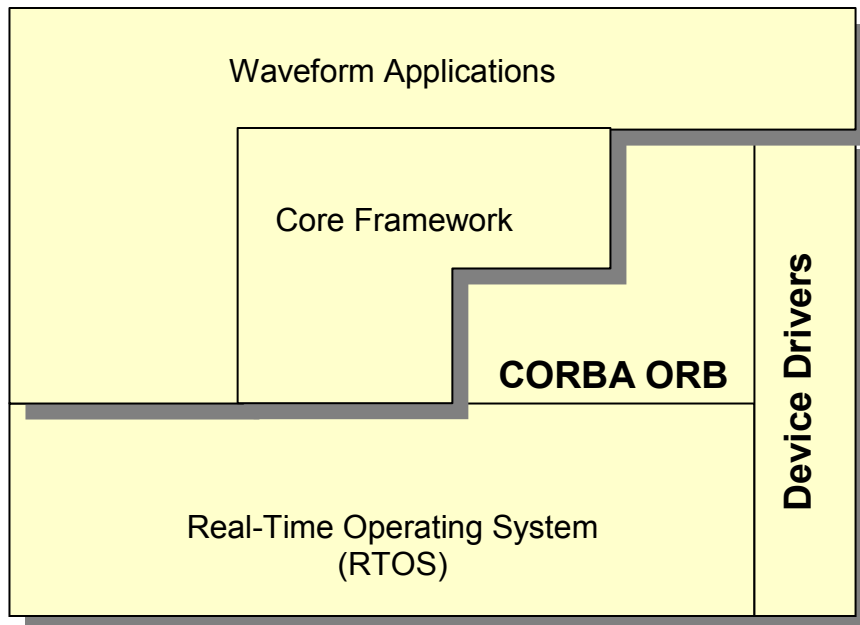
- RTOS
- CORBA ORB
- Core framework



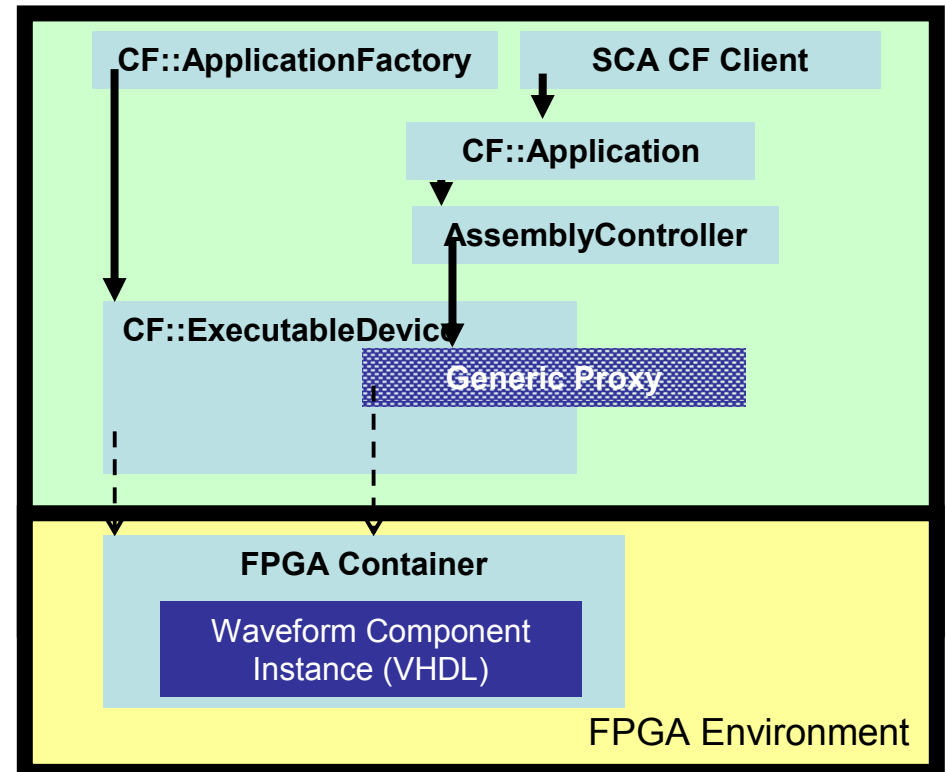
## FPGA Code Portability

- OCP-based Interfaces
- FPGA development environment
- SCA CF agnostic

### Software Communications Architecture



### FPGA Component Implementations



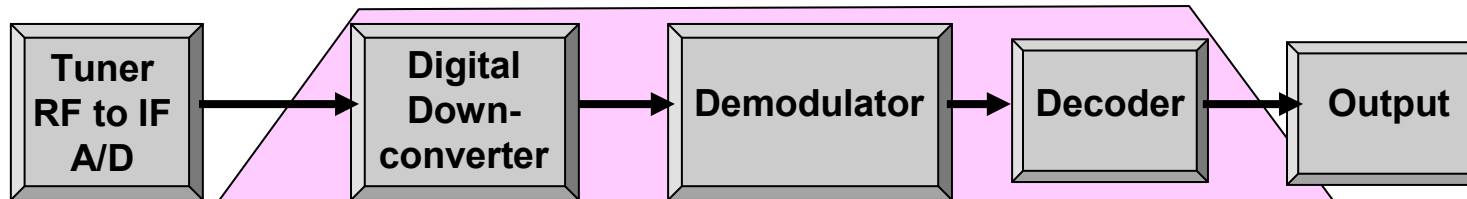
# CPI Enables Waveform-Ready Platforms

- **Built-in control to ensure independence from waveform**
  - Complete control plane functionality built in to CPI
  - SCA compatibility
- **Built-in data transport mechanisms**
  - Supporting high-performance switch fabrics and back planes
- **Built-in memory access**
  - Multiple FPGA memory technologies, DMA
- **Open interfaces to plug-in waveform components**
  - OCP for FPGAs, C/C++ for GPP and DSPs
- **Supports heterogeneous platforms**
  - Supporting GPP, FPGA, DSPs today

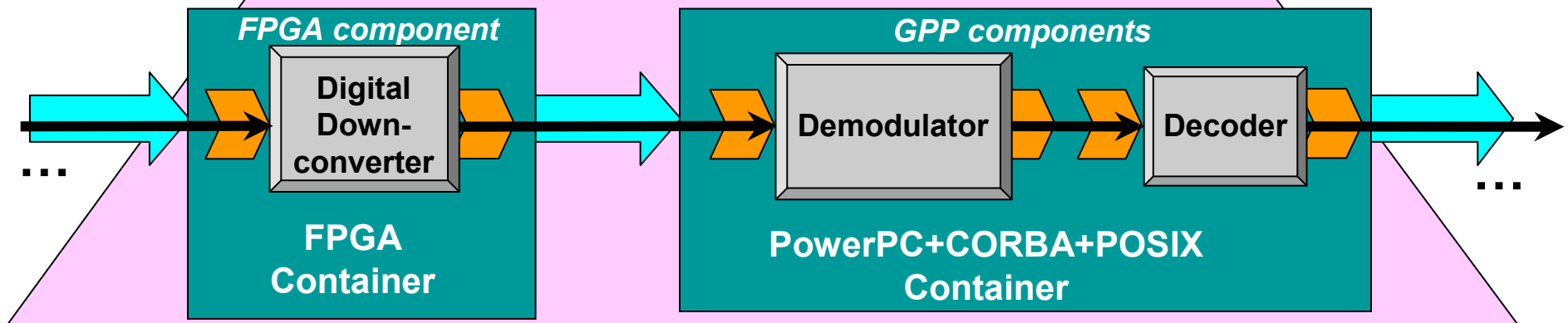
***CPI provides control and data plane infrastructure necessary to plug in 3<sup>rd</sup>-party waveform components***


# CPI Components Execute in Containers

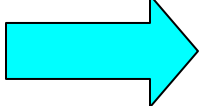
## Simple Receiver



## Component implementations in Containers



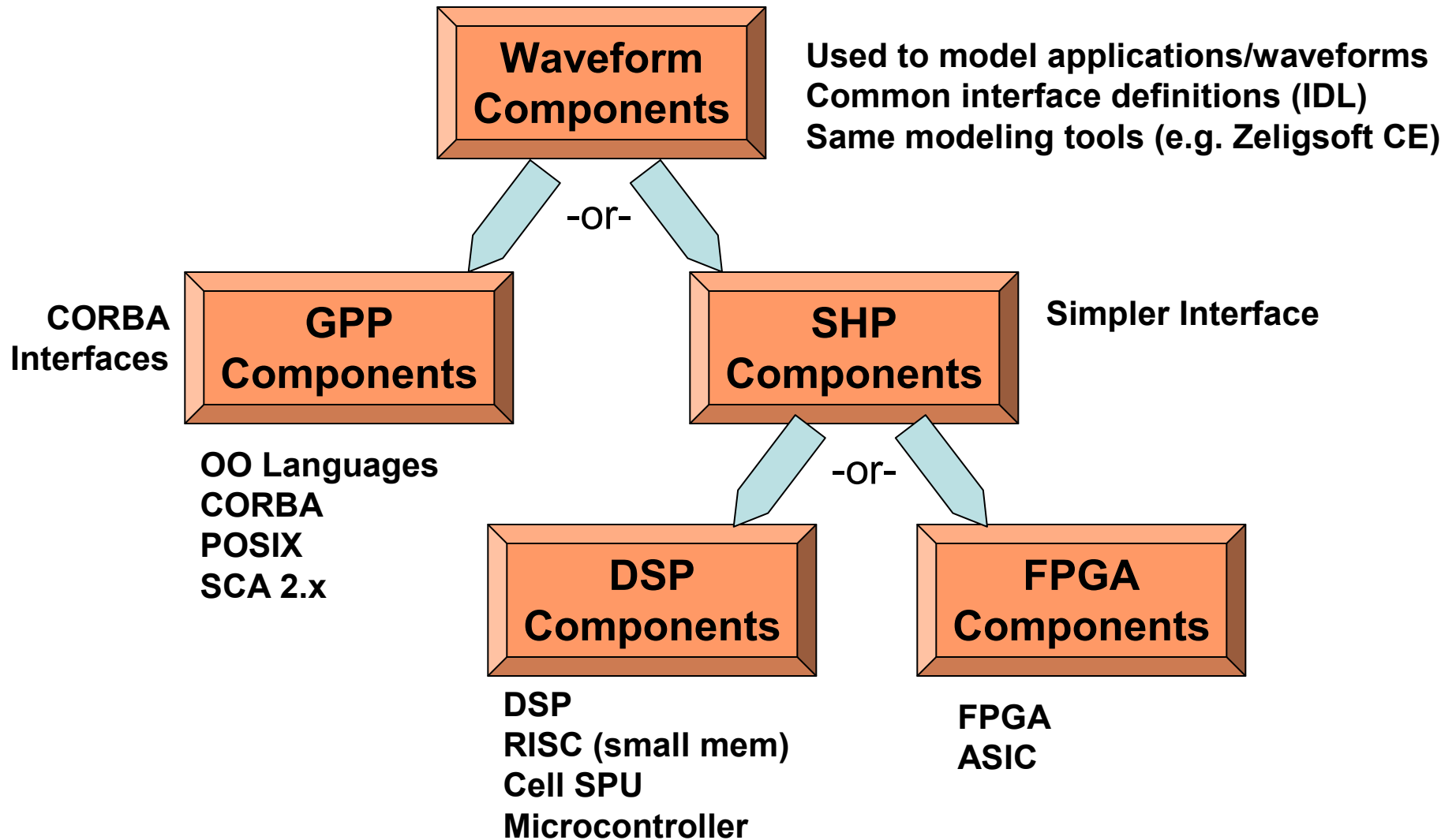
 Components talk to their containers. Important interface for portability of components. APIs used by component authors.

 Containers talk to containers. Important interface for interoperability/plug&play of containers (e.g. boards). Protocols/networks/busses.

 Communication between components, conveyed by their containers

- **Infrastructure to support component-based applications**
  - Control plane and data plane
  - FPGA IP and software/middleware/drivers
- **Targeting heterogeneous system challenges**
  - Setup/control/management and runtime
  - FPGA-centered
  - Peer model with GPP and DSP software
  - Reuse/portability within and across technologies
- **Driven by high-bandwidth, data-link applications**
  - Applications are waveforms
  - Waveforms are assemblies of connected components
  - Most components will be in FPGAs
- **Driven by DoD communications requirements**
  - Open standards and interfaces
  - SDR compatibility
  - Waveform portability
    - DoD owns/provides waveforms to primes
    - Upper limit of component granularity

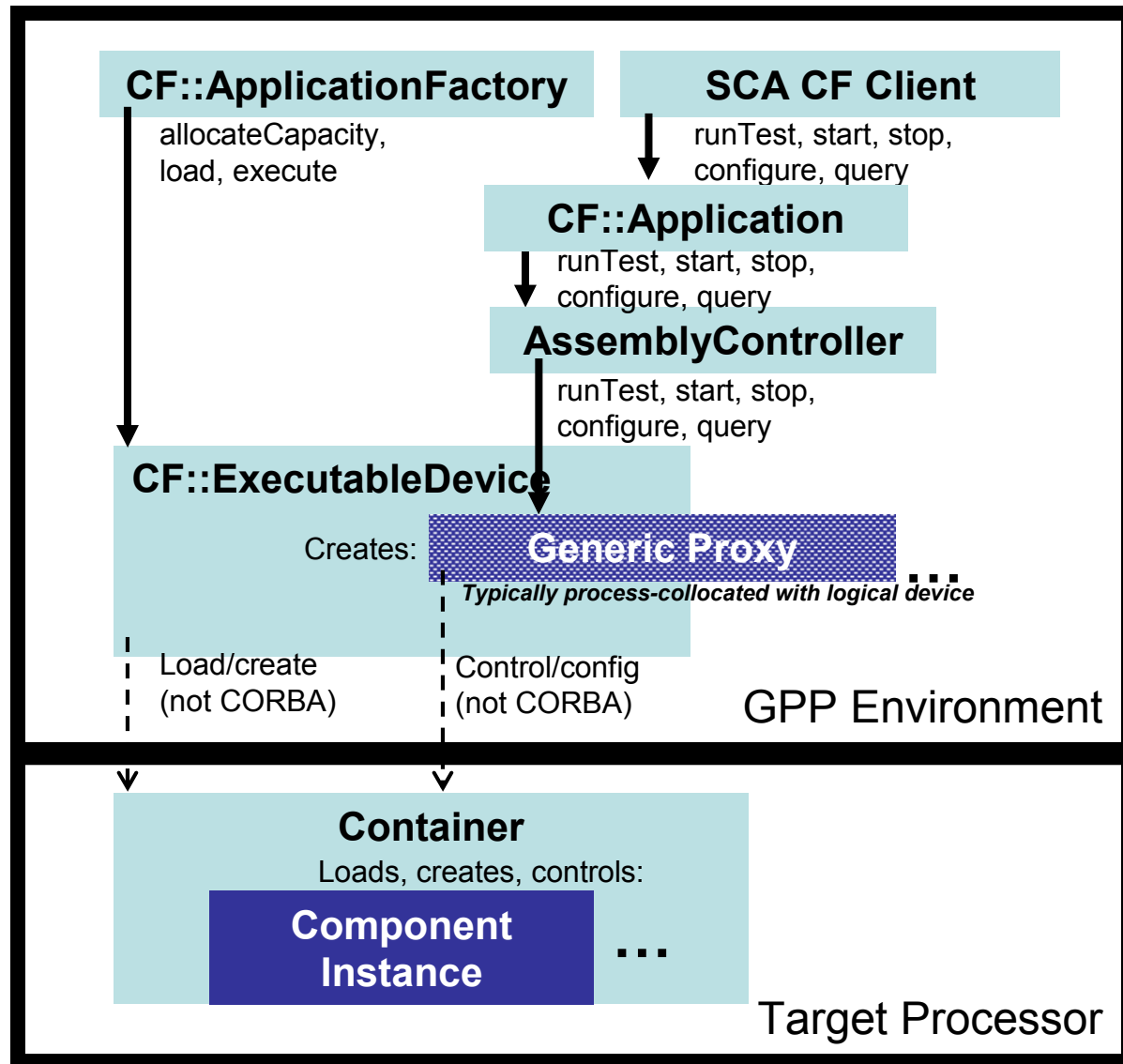
# CPI Supports Multiple Component Types



- **CF:: Resource interface is fixed**
- **Port interfaces use constrained IDL**
- **Request/response messages use derived “struct” from a subset of IDL**
  - Component source code fills/uses struct, sends/received messages
  - No “stub/skeleton” generated code required or desired at this level
  - Simple arrays work fine (fixed or variable)
- **Configuration properties use a derived “struct” from property definition, using same layout rules**

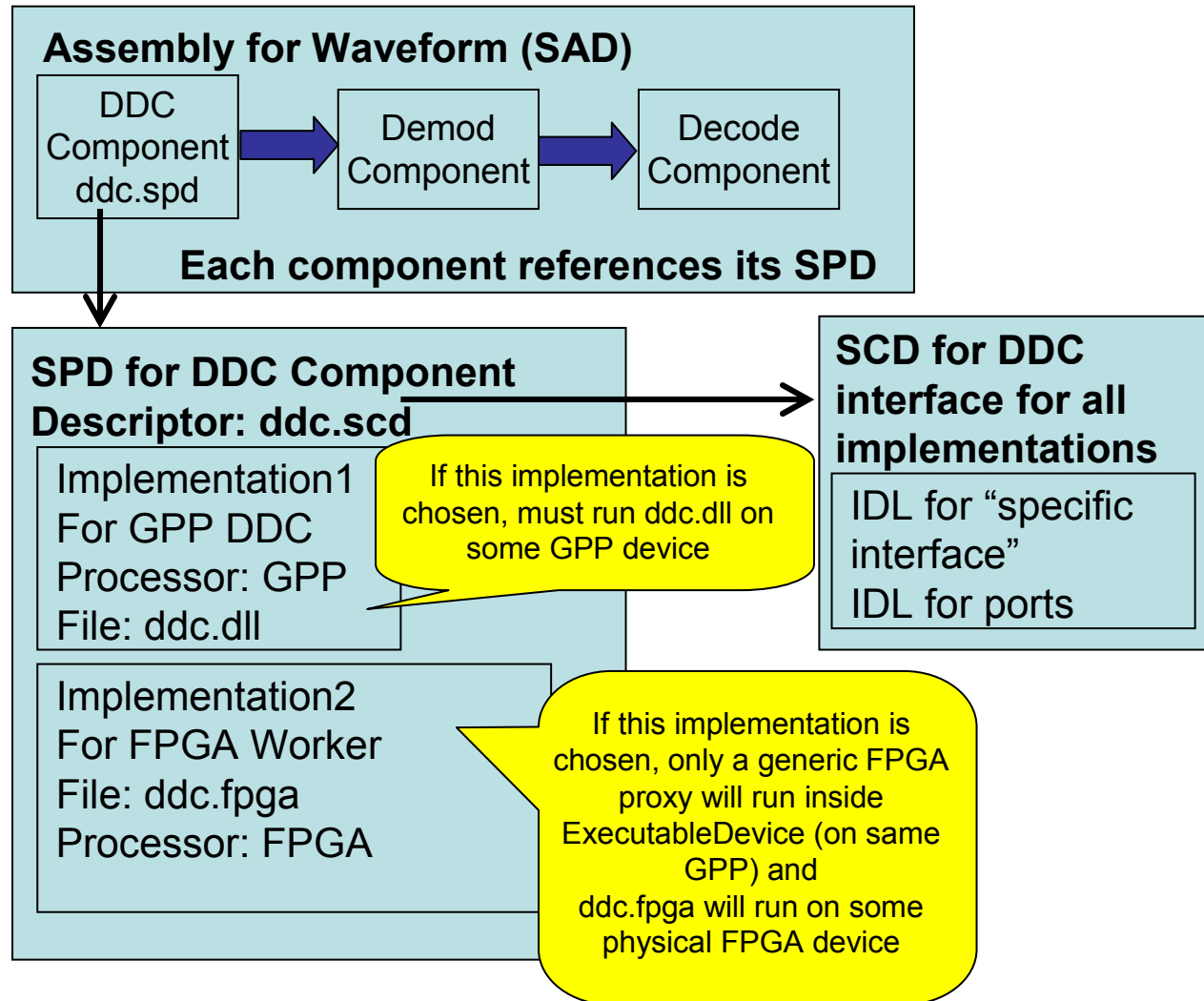
# CPI Component Implementations

SCA  
View

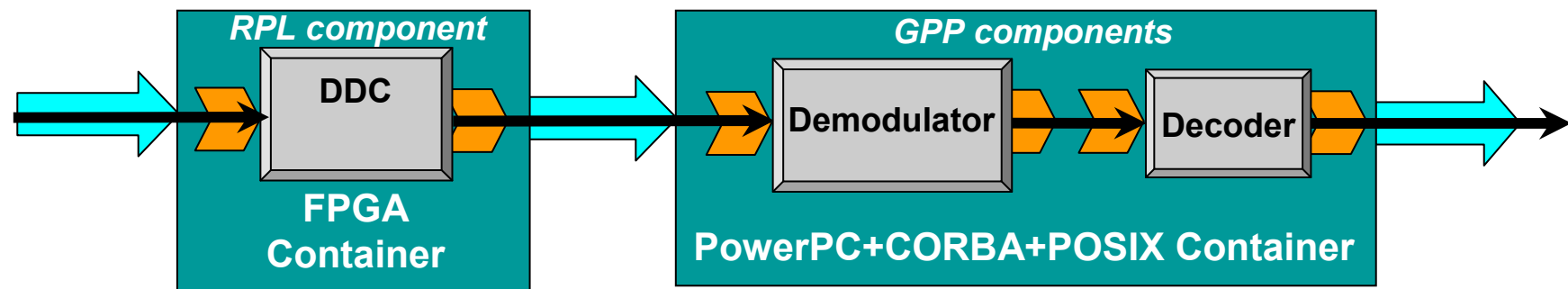


# CPI Component Implementations

SCA  
Metadata  
View



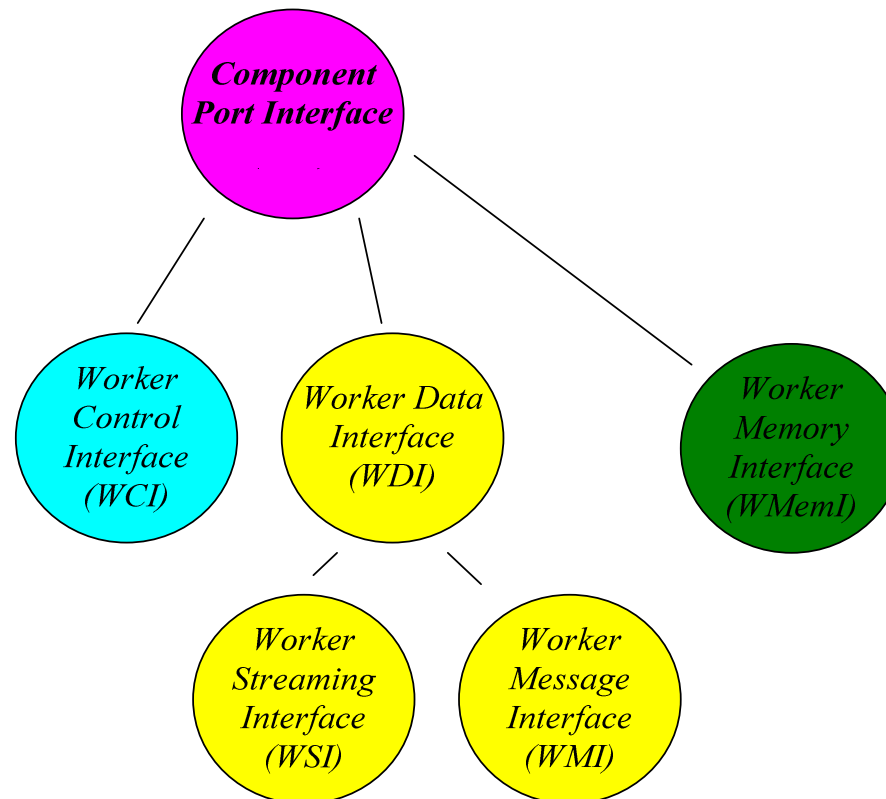
- **Adaptation to GPP ORBs at ORB transport level**
  - ORB transport knows what to do directly at GPP side  
*--- or ---*
  - Container may create CORBA/GIOP messages directly
  - *Choice is up to platform integrator*
  - *No such adaptation is used or needed unless CPI and GPP component implementations are connected*



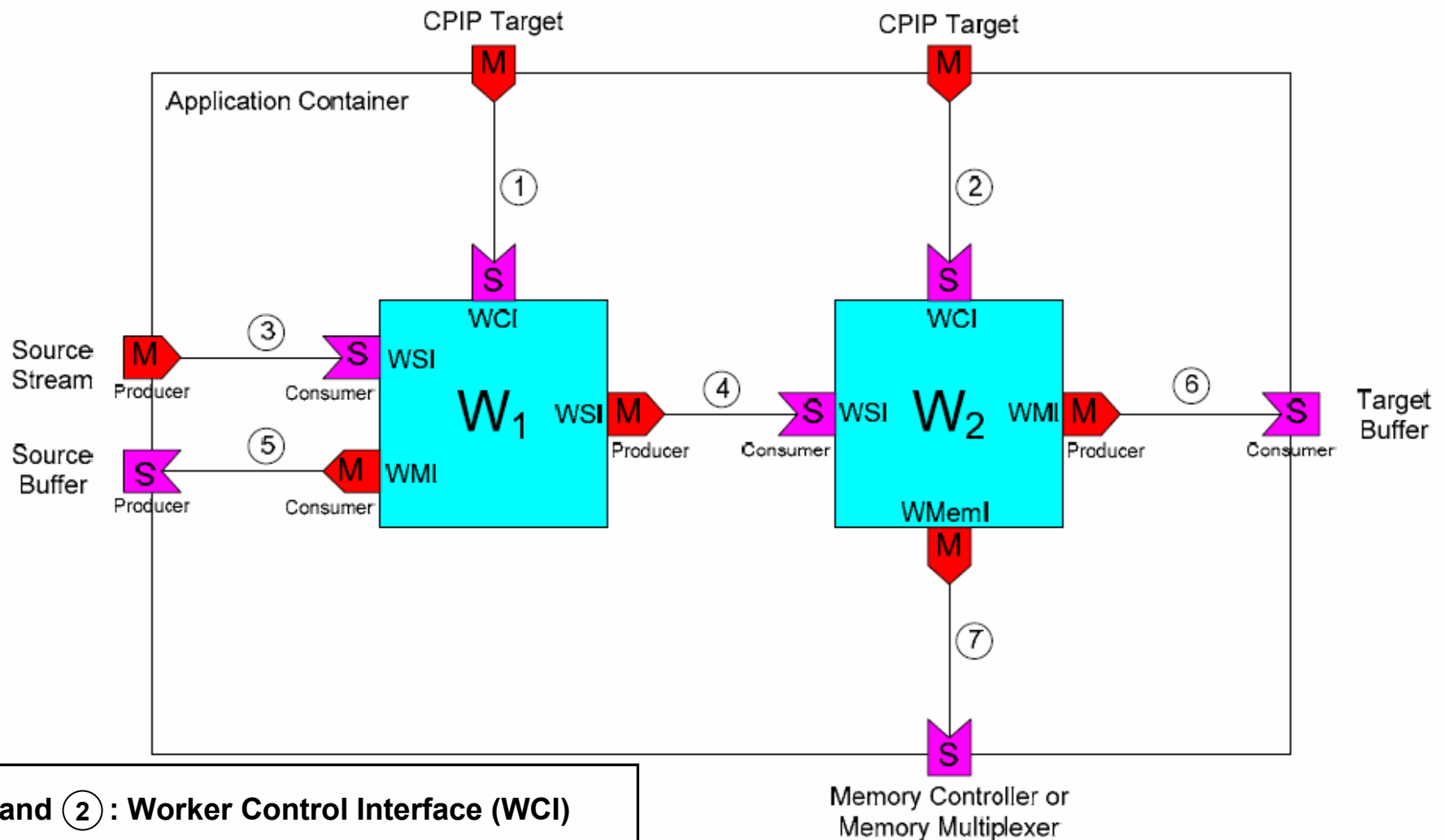
- **Generic proxies, synthesized by SCA logical devices, represent the CF:: Resource interface to world**
- **Proxies acting as “adapters” can do customized translation if “interception overhead” is acceptable**

- **All interfaces defined using OCP**
  - Open standard for how “IP cores” are connected
  - Independent of VHDL vs. Verilog, interface mappings for both
  - Range of performance options
  - Increasing acceptance by users and producers of FPGAs, SoC
- **Management interface**
  - Initialize/start/stop/release/test on one OCP “thread”
  - Configure read/write on second OCP “thread”
- **Inter-component interface**
  - Burst read/write transactions on OCP-port
    - One OCP port per IDL port per direction
  - Implementation chooses master or slave role
  - Implementation chooses FIFO or random access style
- **Local interfaces**
  - Clocks and local memory access (several styles)

- **Component interface – collection of OCP profiles**
  - Fully compliant with OCP 2.2
  - Signal definition and some semantic information
  - Used for control, data, and memory interface patterns



# Example: 2 FPGA Components in a Container



- **Full consistency with SCA's component model**
  - Consistent with concept of IP cores
- **Full interoperability with CORBA**
  - Without introducing any CORBA into FPGA IP
- **Methodology using a mature, SoC/ASIC-tested approach**
  - Consistent with best practices of how IP cores are normally developed
  - Integration (control and data) provided behind the scenes
- **Minimal latency and footprint for control and data plan**
  - Full integration at SCA component level
- **Optimal performance for connections**
  - Inside FPGA
  - Between FPGAs on board (parallel wires)
  - FPGAs across backplane (serial wires)

- **Open and published specification for waveform interfaces**
  - Anyone can implement/wrap/support it – no vendor lock-in
- **Clear separation of control and data plane**
- **CPI based on high-performance FPGA Developer's Kit (FDK)**
  - Shipping for 5 years with lots of maturity and experience
  - State-of-the-art performance from pair of FPGAs in a smart missile to 100 FPGAs across mesh-connected backplane
- **IP kit includes**
  - Support for component environment AND
  - Support for hardware elements surrounding FPGA such as transceivers, memories, fabrics, and DMA