



An Open Source FPGA Infrastructure for Heterogeneous Component-Based Systems and Applications:

OpenCPI - Open Component Portability Infrastructure

*SDR Forum FPGA Workshop
September 17th, 2009*

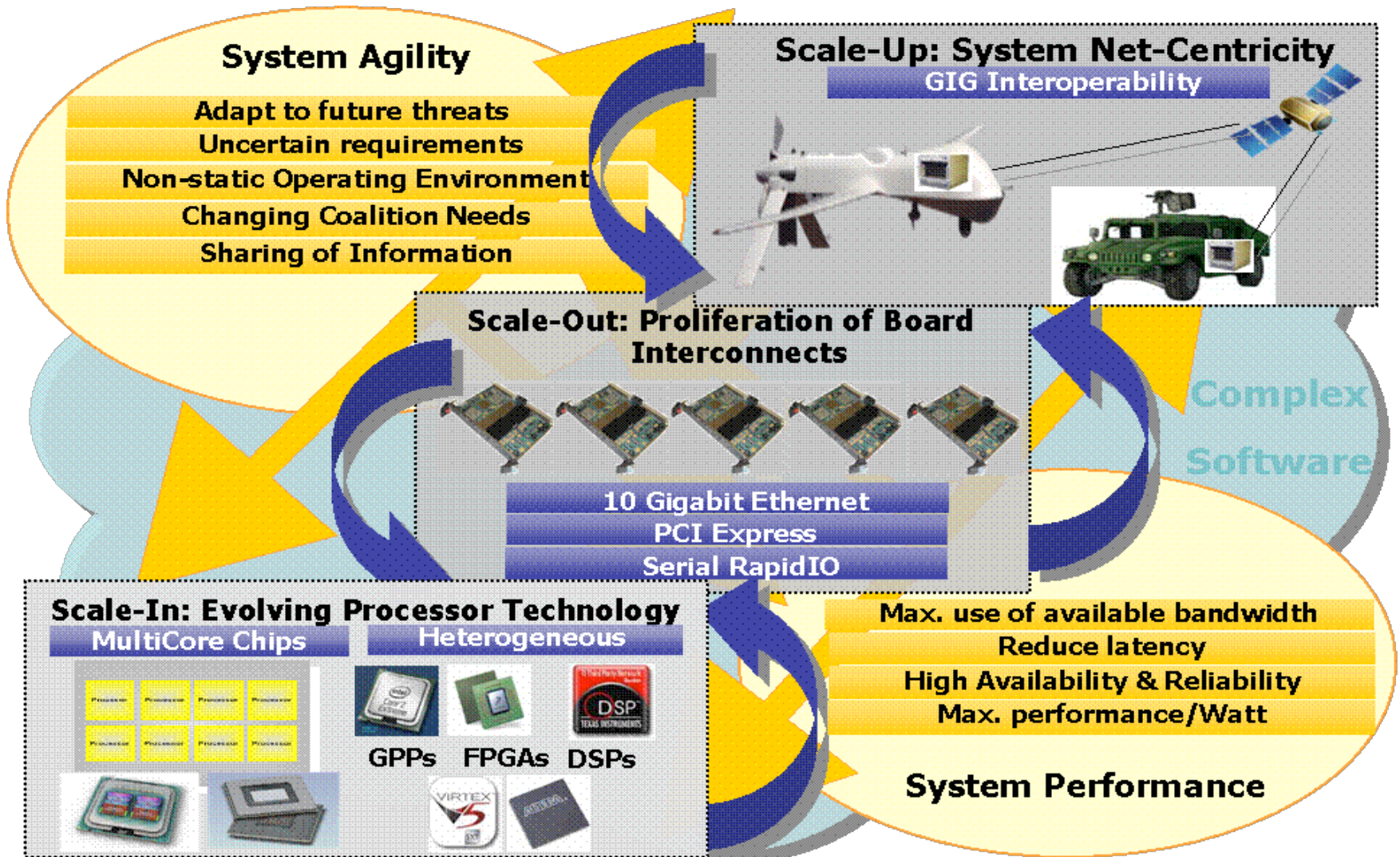
Jim Kulp

jkulp@mercfed.com

Shep Siegel

shepard.siegel@atomicrules.com

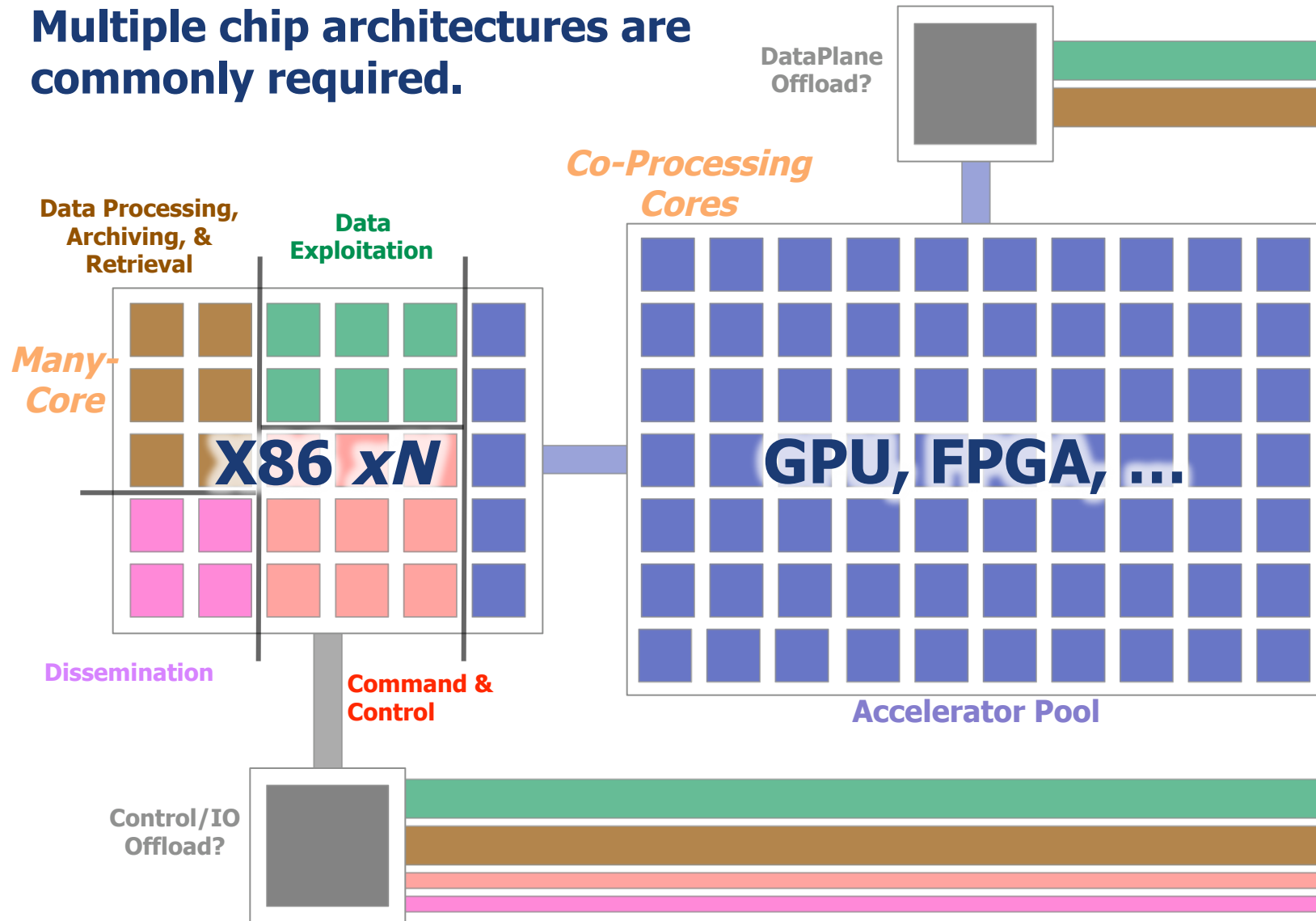
Mission Needs & Technology drive Software Complexity



Changing mission requirements combined with technology proliferation lead to more complex & costly development/integration/maintenance

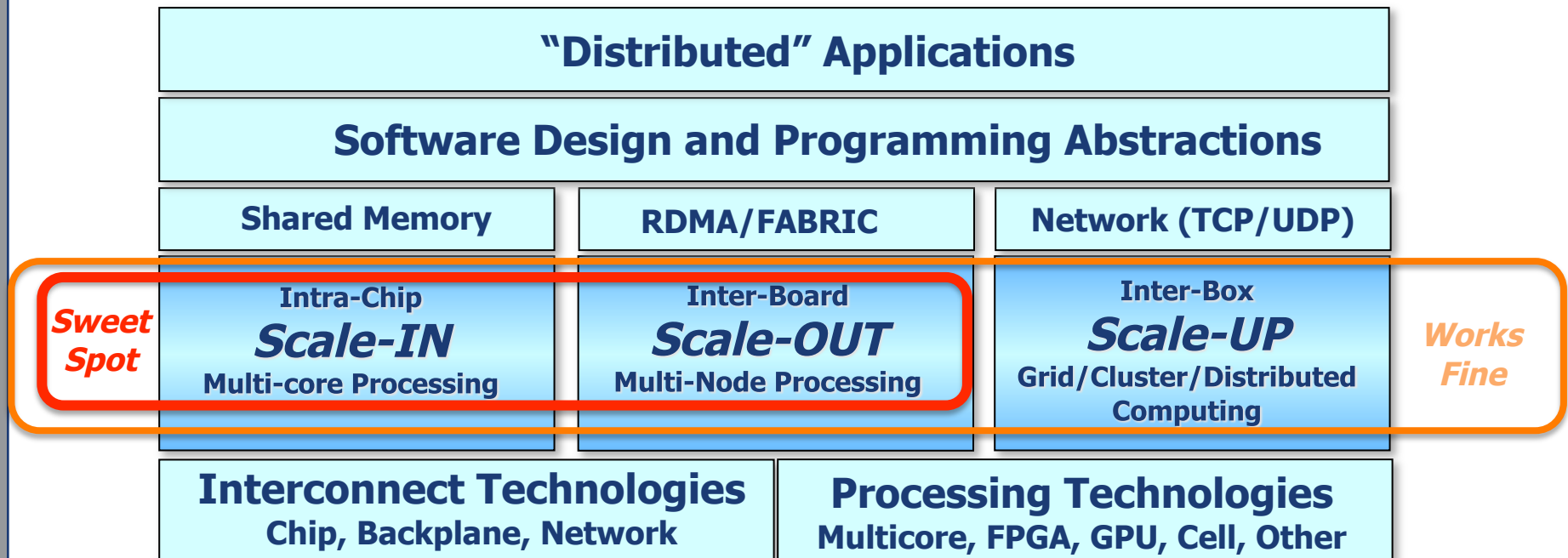
Complexity in processing architectures

Multiple chip architectures are commonly required.



Scalability Challenge

Various scaling options available, with different performance/latency tradeoffs



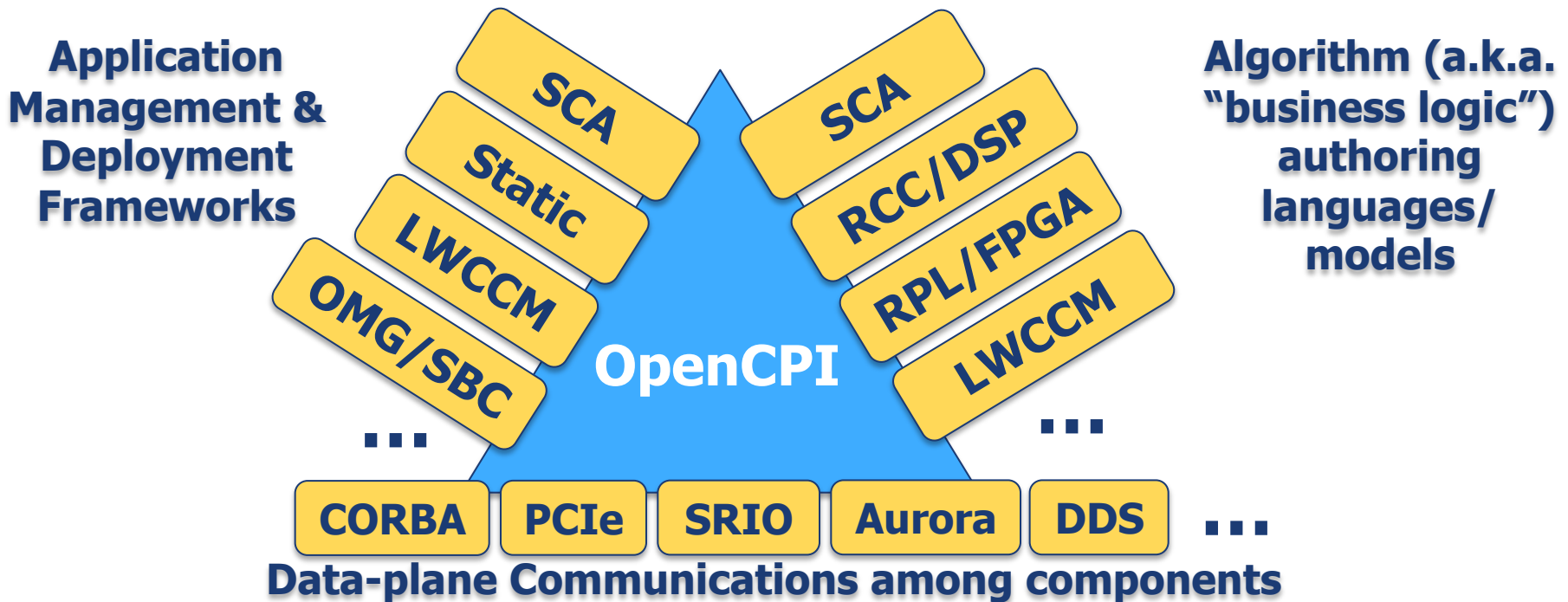
OpenCPI: Open-Source Component Portability Infrastructure

- Embedded framework for Component-Based Development (CBD)
 - focused on close-to-the-metal technologies
- For *heterogeneous, component-based, embedded* applications
 - FPGA, GPP, DSP, GPU, Multicore processing, as peer technologies
 - Intra-chip/interchip/interboard/network interconnects in applications
 - The reality of complex, SWAP-sensitive, deployed applications
 - Within a system (interoperating) or across a tech refresh



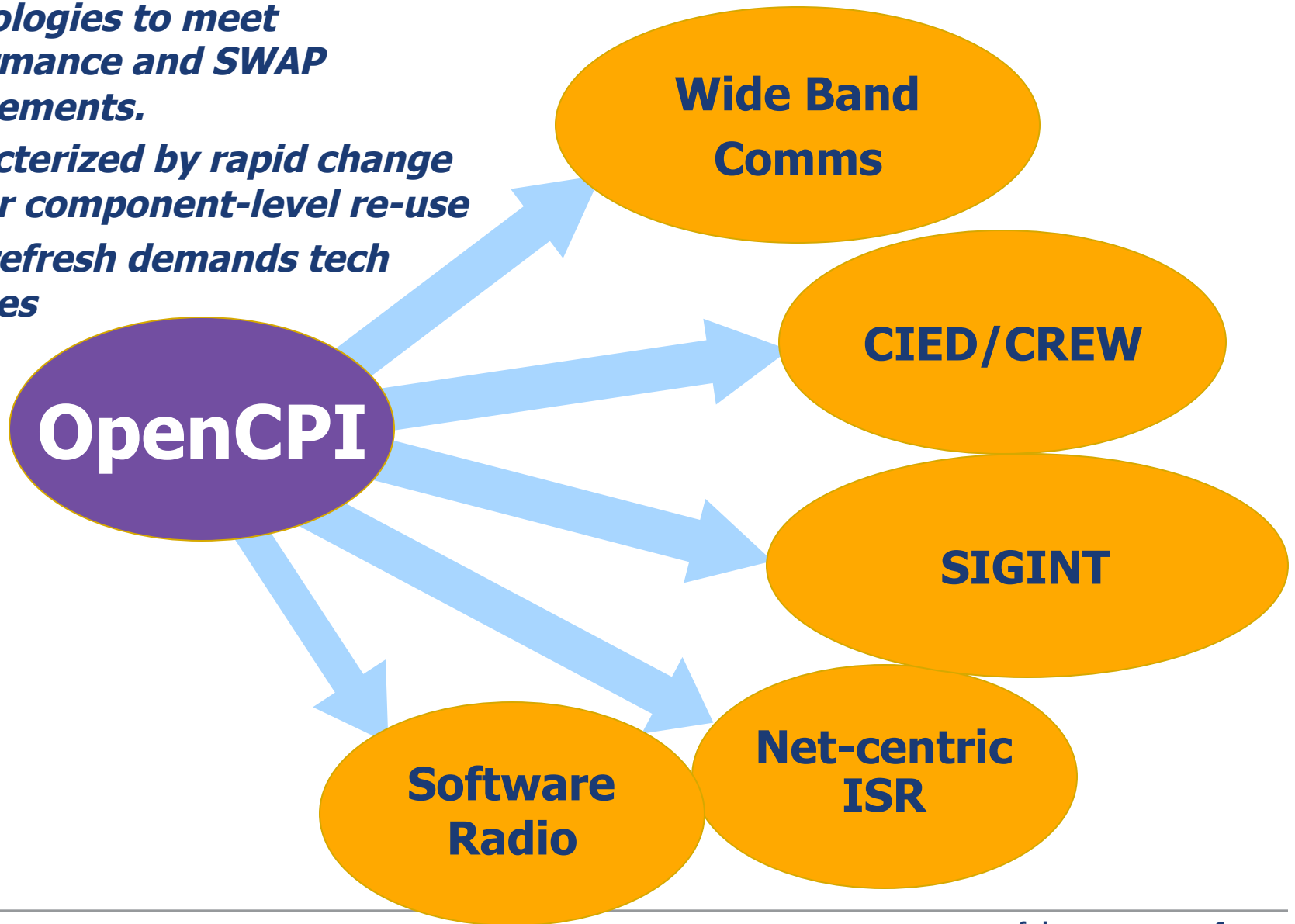
OpenCPI is...

- Different/multiple component authoring models:
 - How to write app code as components for different technologies
 - Build/make/debug mechanisms and low-level tools for each model
- Execution environment code for each model
 - The “container” that runs and controls the components on a processor/node
- Transport mechanisms/drivers/IP so components can talk to each other.

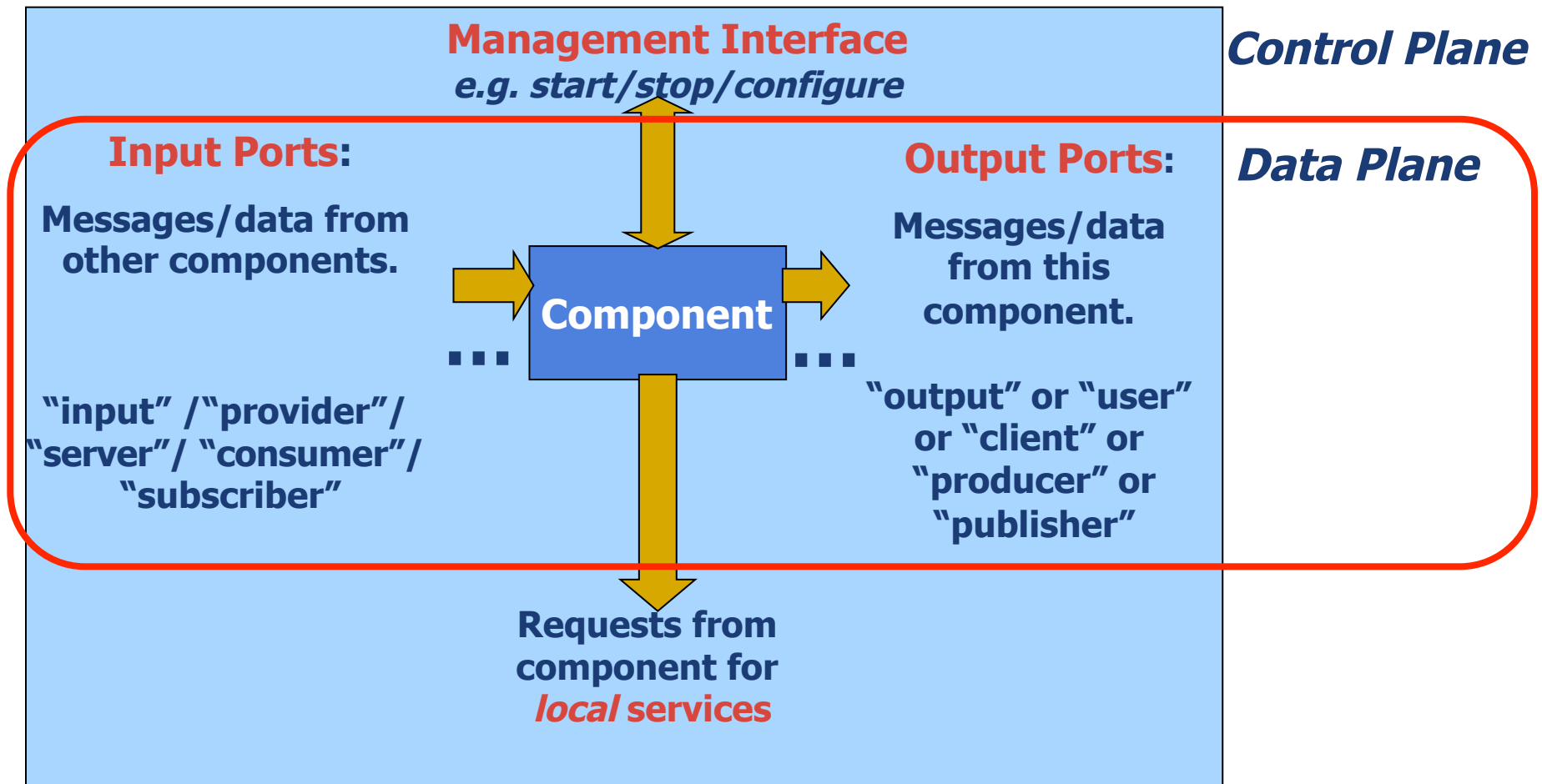


Where OpenCPI can be (and is being) used:

- *Application areas using advanced technologies to meet performance and SWAP requirements.*
- *Characterized by rapid change and/or component-level re-use*
- *Tech refresh demands tech changes*

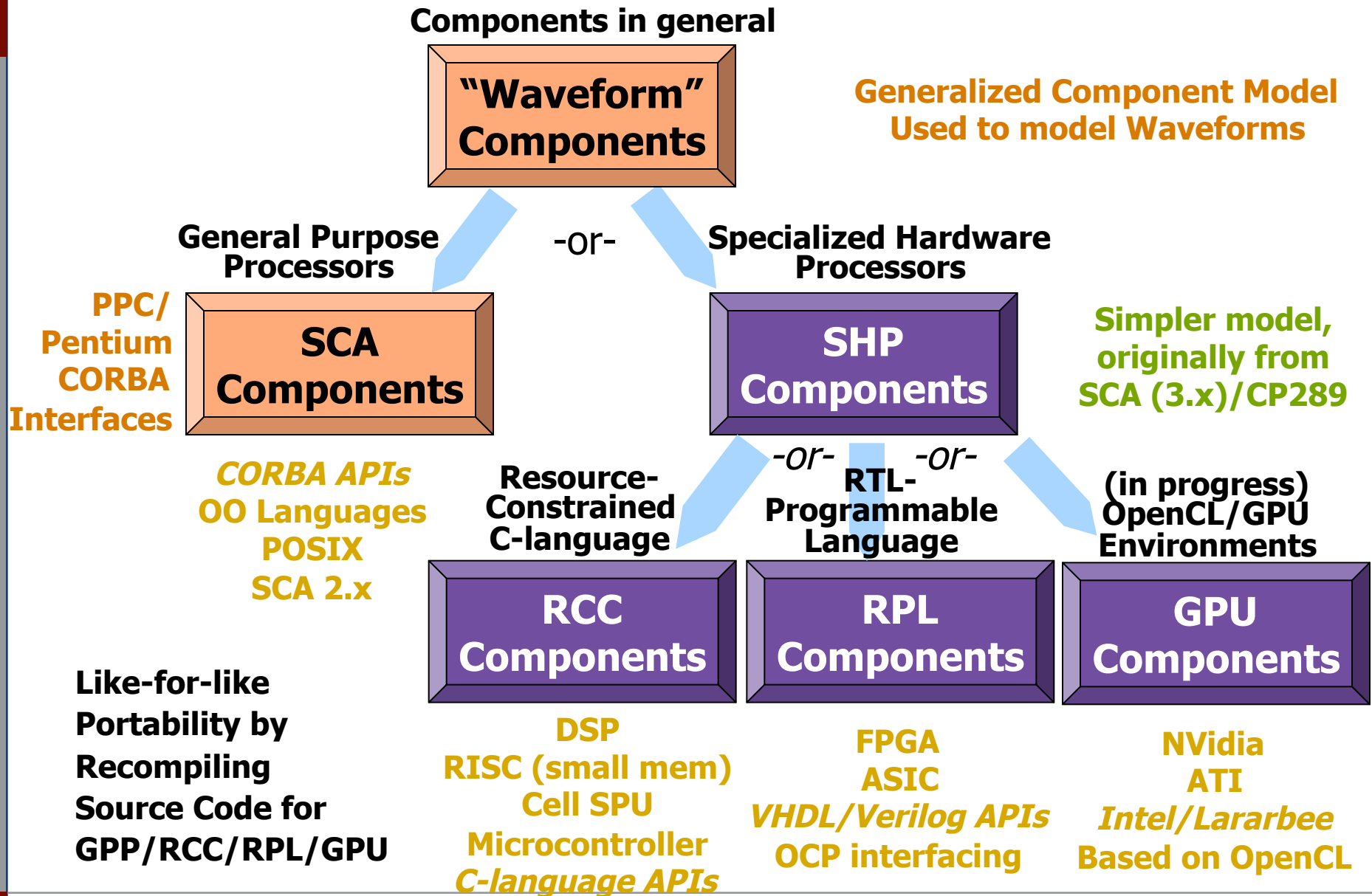


SW/Firmware Component Authoring Model Pattern



- **Models for all technologies must *interoperate* in a system**
- ***Simple enough for very low-level implementations***
- ***Designed to be compliant (as subset/profile) with other frameworks***

Authoring models for Component Implementations

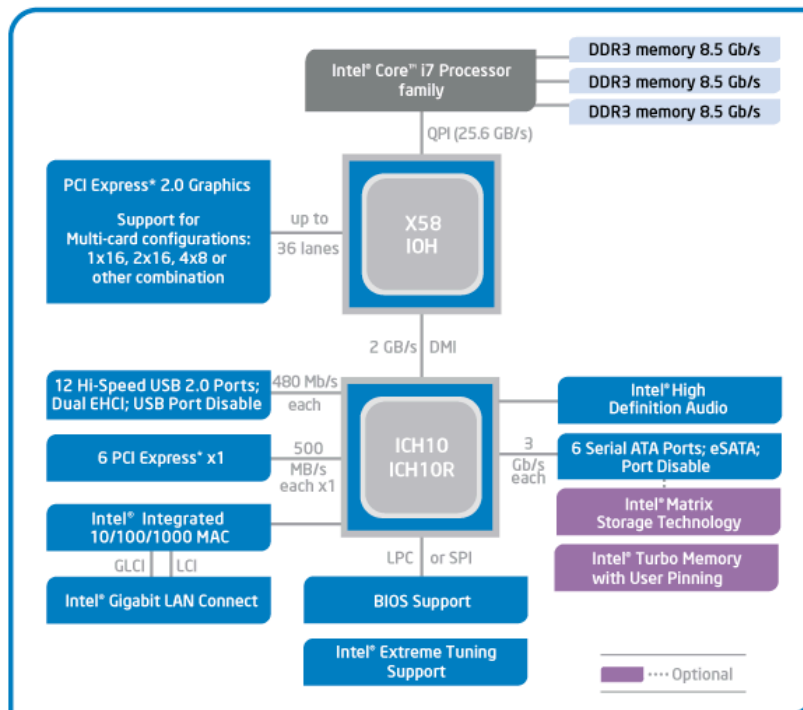


Key facts, OpenCPI relating to the DoD SCA.

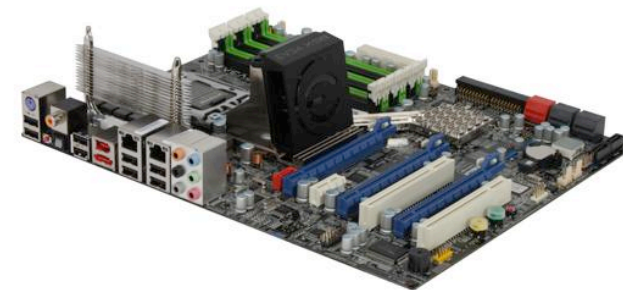
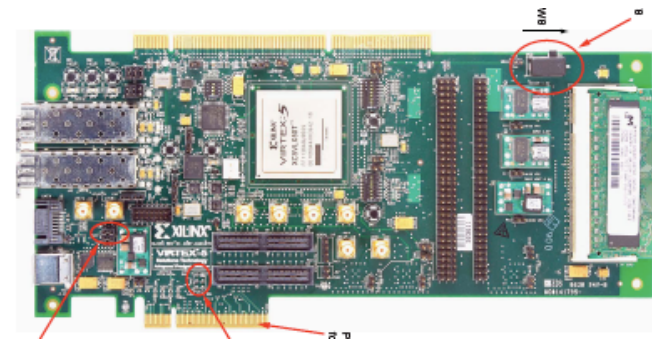
- OpenCPI is not a replacement for a core framework
 - Has CF::ExecutableDevice interface for all containers, including SCA
- Non SCA/CORBA/POSIX components (DSP/FPGA/GPU etc.):
 - Coexist with runtime interoperability
 - Are transparent to CF, assembly controllers, waveform components
 - Can use same metadata (SPD/SCD/PRF + IDL)
 - Can use same modeling tools (we have used SCARI and Zeligsoft)
 - Use IDL restrictions from CORBA/E micro profile
 - Configuration/test properties supported, with max sizes for sequences and strings (the IDL max size feature missing from SCA PRF).
 - CORBA only used when talking to C++/SCA/CORBA components.
- OSS system uses OSSIE CF, OMNI ORB, Linux
 - Delivered to programs using Harris/SCARI CF, OIS/TAO ORB, VxWorks-DKM/Linux

Modern, low-cost, powerful reference system

- Latest desktop technology that supports high speed slots for other technologies
- 3 high speed slots for other processors: GPU, FPGA, Cell
- Cheap! Box w/FPGA board + *tools* = ~\$5K.



Intel X58 Express Chipset Block Diagram



OpenCPI Reference Platform & Tools: Goals

- Easy to get started.
- Easy to buy – low cost of admission.
- Easy to experiment – hours, not weeks to see results.
- Easy to leverage work from others.
- Easy to mix cutting edge processing technologies (multi-core CPUs, GPGPUs, FPGAs, etc.)
- Turn on, install all SW, modify FPGA “hello world” and run < 2hr.
 - FPGA edit/build/reload/run ~20 minutes.

OSS Licensing for OpenCPI: Context

Context: Real Time Embedded Software Community

- NetFPGA: BSD-style license
- VSIPL: BSD-style license
- OpenAIS: BSD-style license
- RTLinuxFree (WindRiver): GPL + Closed Version
- OpenSAF: LGPL v2.1
- Qt: LGPL

Context: The Defense Community

- Some Defense/Intelligence Community contractors/primes have prohibitions about using the GPL
- Potential users want to mix with existing/new IP
- Some potential contributions may be classified (for classified reuse)

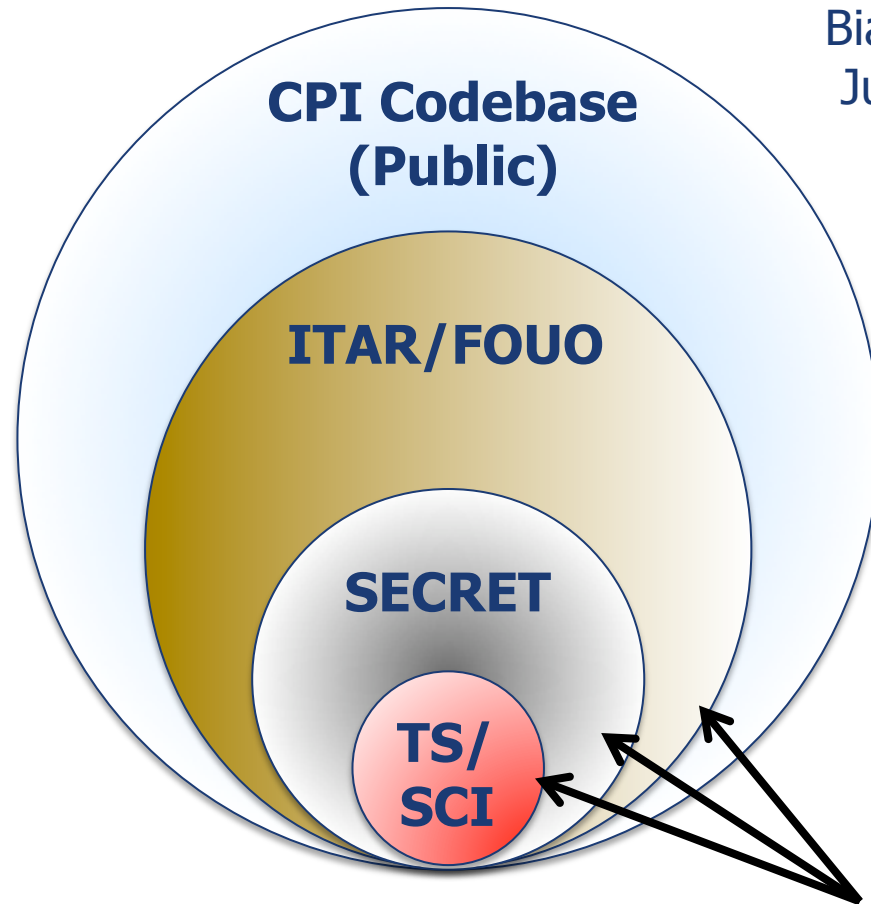
Context: Different OpenCPI contributors:

- Some contributions may be classified (for classified reuse)
- Some platform support modules may be proprietary

Conclusion: OpenCPI uses LGPL 3.0, like Jboss/Mozilla/OpenOffice

Development in Constrained Environments

More specialized application code



Bias to make open, Justify exceptions

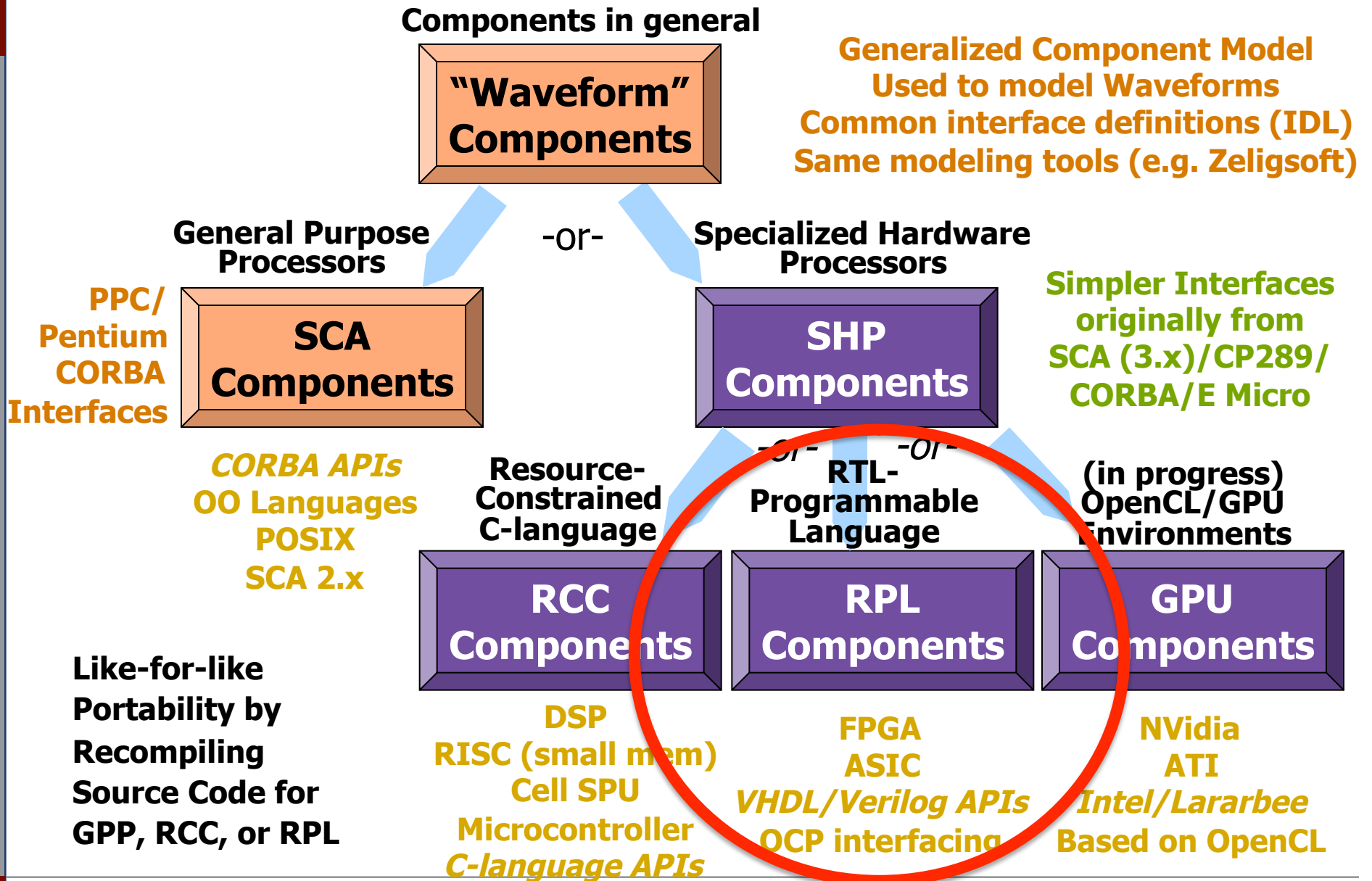


Studied, Future setup

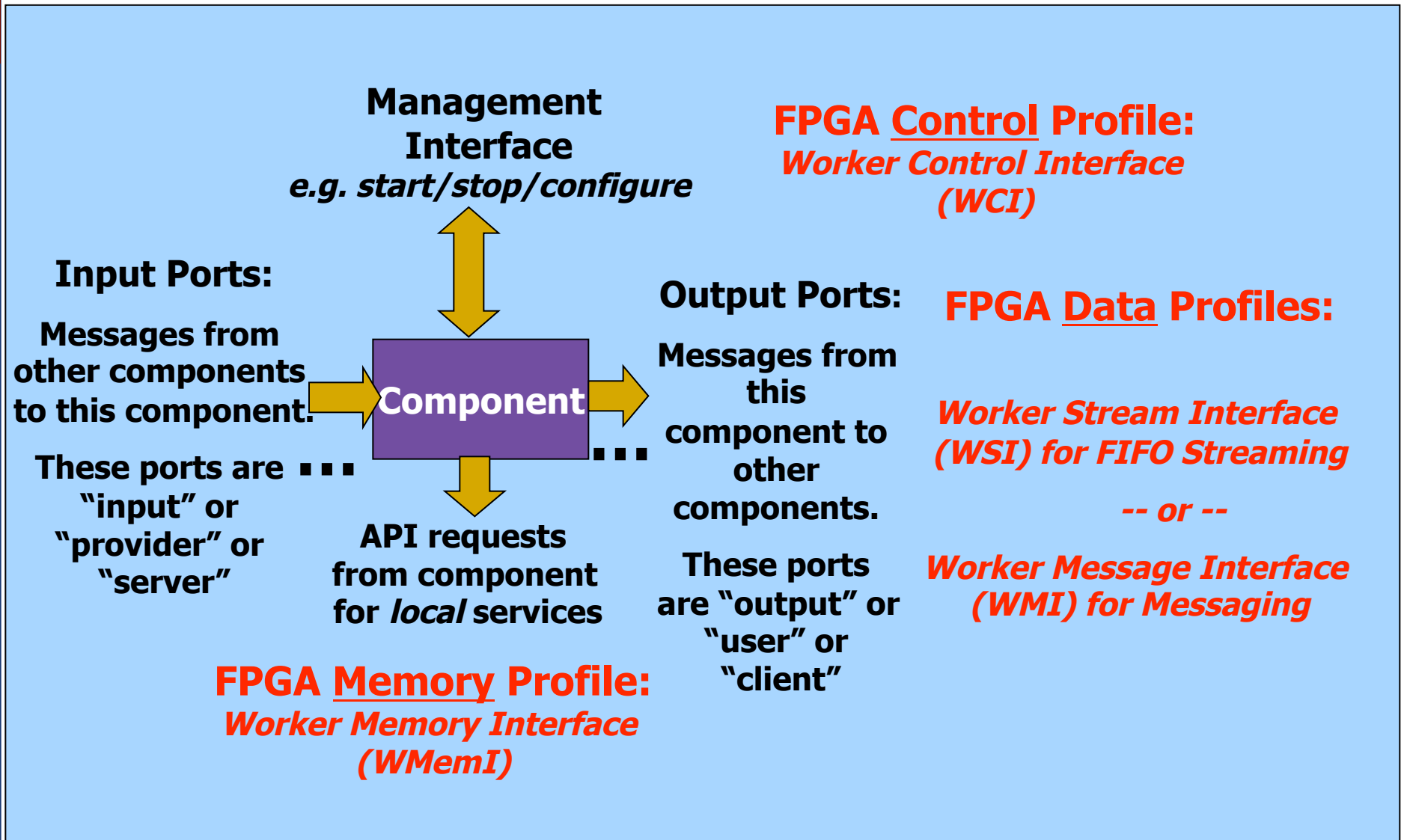
OpenCPI Status

- Initial OSS prototype posted on site
- Appropriate licensing established
- Squeaky clean non-proprietary
 - Some new/immature pieces that replaced non-OSS modules
- Ready soon for community participation
- SCA Adaptation with Omni ORB and Ossie CF.
 - Core code used on programs with SCARI and Harris.
- Site established and various subsystems selected
 - www.opencpi.org and subdomains:
 - opencpi.org/wiki
 - opencpi.org/forum
 - opencpi.org/blog
 - <http://trac.opencpi.org> - code roadmap and bug tracking
 - <http://svn.opencpi.org> - code repository

FPGA Component Authoring Model & Infrastructure



FPGA model via modestly parameterized interfaces

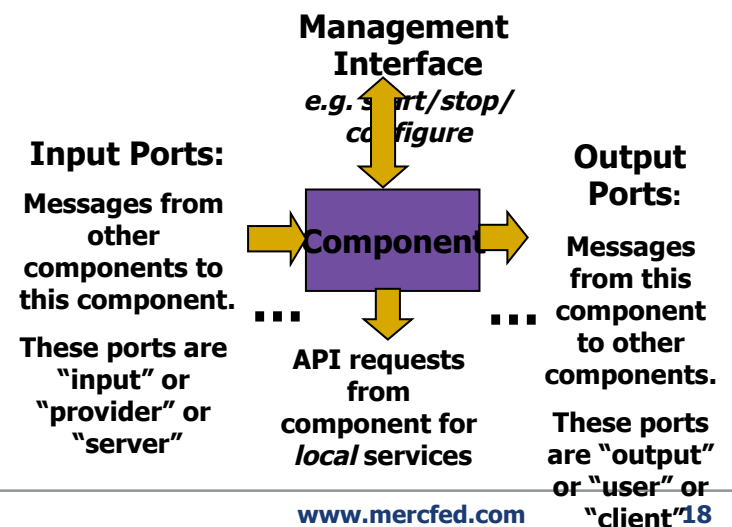


Interfaces achieved via OCP profiles

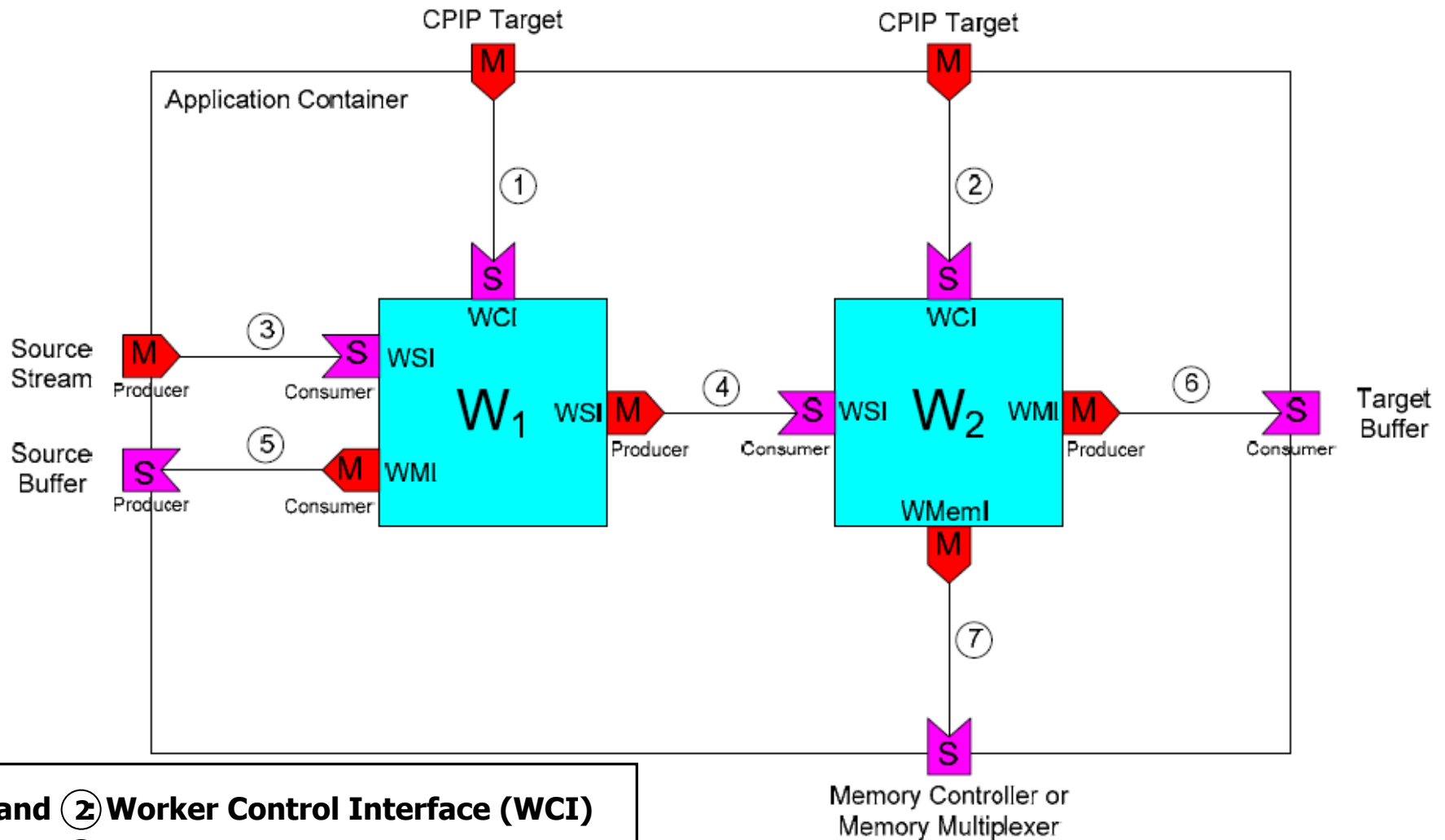
- All interfaces are defined using *carefully profiled* OCP
- Open Core Protocol:
 - A standard for defining how “IP Cores” are connected
 - A broad range of clearly defined interface choices
 - Profiling is key to appropriate use
- Language independent
 - VHDL vs. Verilog vs. BluespecSV etc., doesn’t matter
- FPGA “Components” are natural “IP Cores”.
- OCP specified by several DoD programs

Profiles for the FPGA ("RPL") Component Model

- **Worker Control Interface (WCI):**
 - Provides control and configuration: the control plane
 - Supports the SCA's control model
- **Worker Stream Interface (WSI):**
 - Provides unidirectional FIFO streaming with flow control.
 - Producer interfaces are masters, consumer interfaces are slaves: *glueless*
- **Worker Message Interface (WMI):**
 - Enables producing/consuming data from buffers with random addressing and/or buffer reuse.
- **Worker Memory Interface (WMemI):**
 - Provides access to local memory:
 - SRAM/DRAM/BRAM.
 - *Not for communication.*

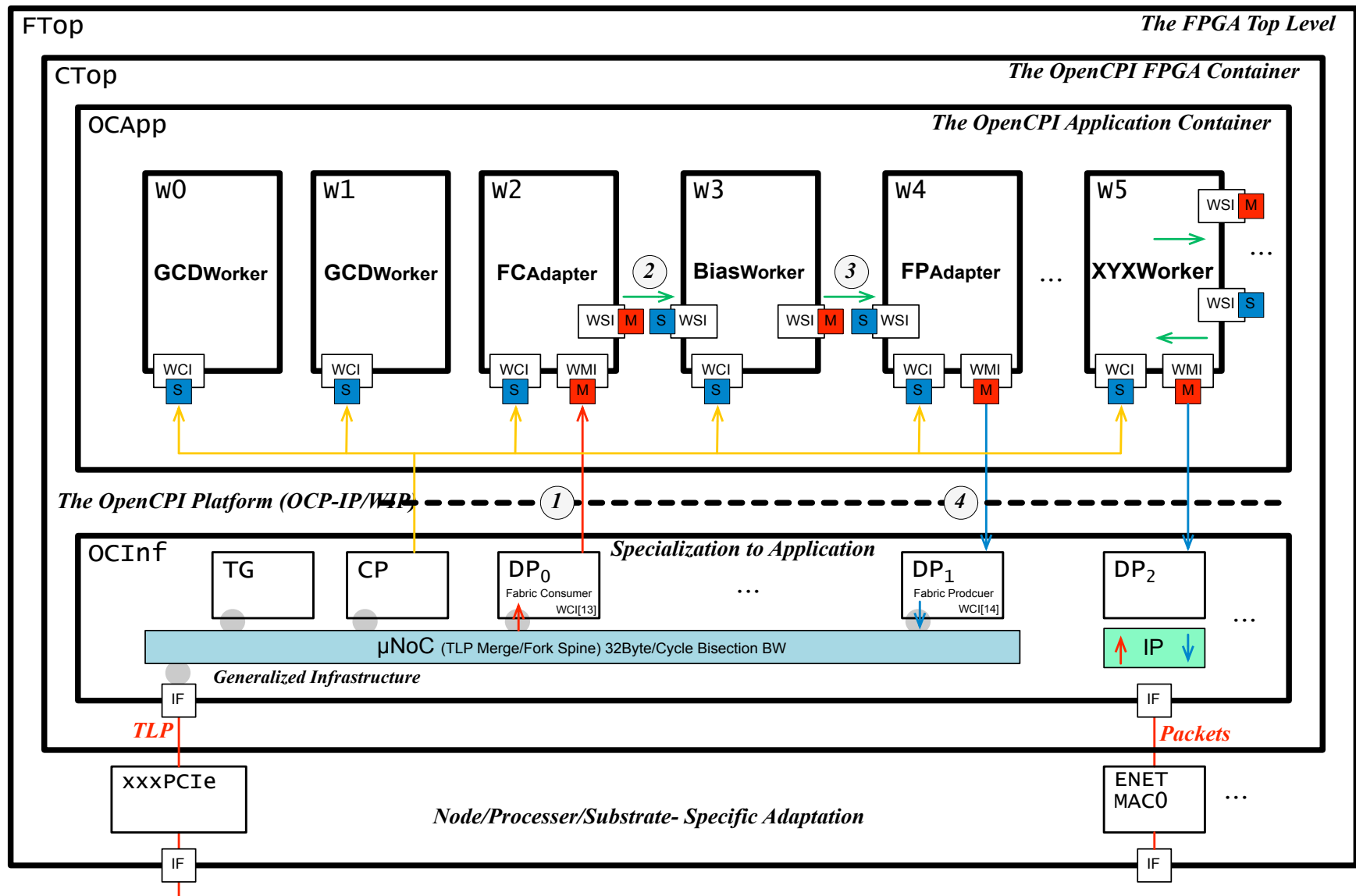


Example: Two FPGA Components in a Container



- ① and ② Worker Control Interface (WCI)
- ③ and ④ Worker Stream Interface (WSI)
- ⑤ and ⑥ Worker Message Interface (WMI)
- ⑦ Worker Memory Interface (WMemI)

New & Open FPGA On-chip architecture for OpenCPI



New & Open FPGA On-chip architecture for OpenCPI

- Clearly defined roles for different blocks:
 - *Application functions*, e.g. algorithms, waveform components
 - *Adapters*: to locally connect app blocks written with different profile choices
 - *Interconnect blocks* allowing off-chip comms with other components in other processors *of any type*
 - *IO blocks* allowing I/O to sensors or other systems
 - i.e., *not* other components
- Layered design dramatically improves productivity
 - E.g., Focused/fast simulation, rapid development loop

Conclusions

- CBD can be effectively applied to embedded systems
 - The SCA started the train down the track
- Heterogeneous systems are a fact of life
- Multiple compatible and interoperable models are required
- OpenCPI embodies this tradeoff:
 - Carefully select a core model (~~ a simplified SCA model)
 - Allow different mappings to different technologies
 - Implement a low-level container and transport architecture
 - Don't dictate control system or transport technology
- Software Radio is one of many application areas for embedded component systems

