



XILINX

ALL PROGRAMMABLE™

OpenAMP Kick Off

Agenda

- Overview of OpenAMP Governance: 30 Minutes
- OpenAMP vs OpenAMP MCA Working Group: 15 Minutes
- Introduction to OpenAMP project and current capabilities: 15 Minutes
- Overview of Identified Gaps in OpenAMP and Evolving OpenAMP: 25Minutes
- NXP OpenAMP modification: 15 Minutes
- What's in the Next Release: 5Minutes
- Total Meeting Time: 120 Minutes

OpenAMP Maintenance

➤ OpenAMP Maintenance:

- <http://openamp.github.io/docs/CONTRIBUTING.md>

➤ Notes:

- OpenAMP next release: 2016.04

➤ Gaps:

- OpenAMP and Linux kernel upstream
 - There is currently no plan to line up with Linux kernel
 - There is an maintainer ship gap about the rpmsg/remoteproc in kernel needs to pick up.

What is OpenAMP?

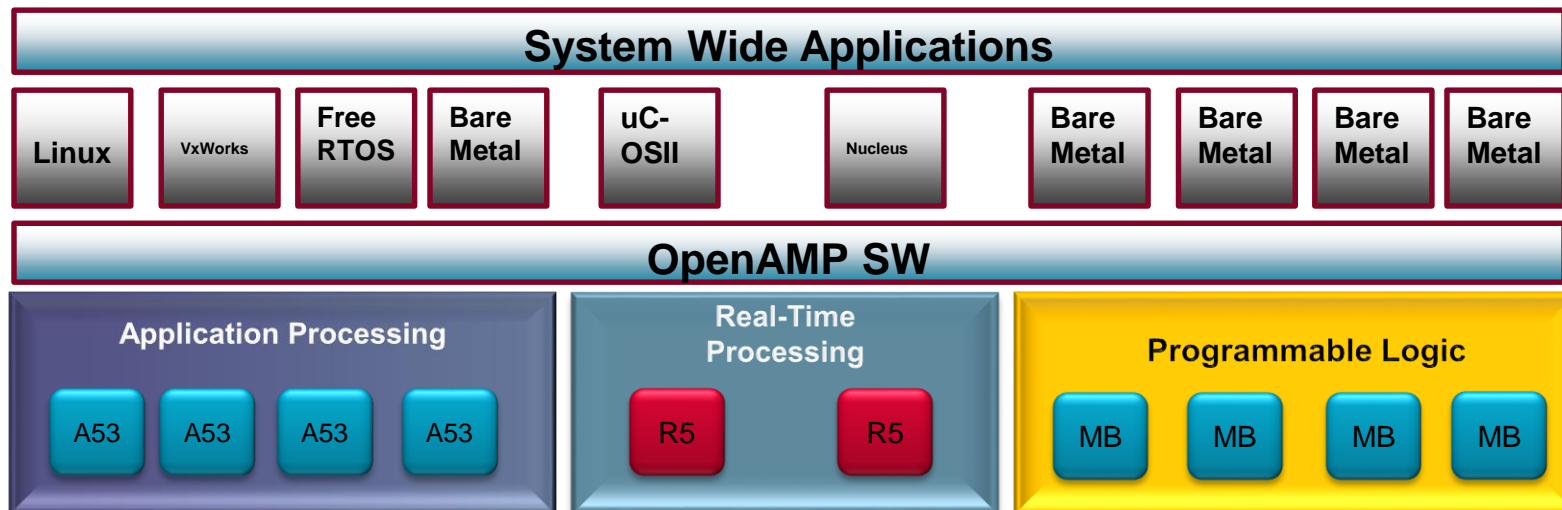
➤ An Open Source project

- Today driven by Mentor Graphics and Xilinx
 - Also used by Micrium, NXP
- Based on existing open source technologies in Linux
- Clean-room implementation for non-Linux kernel code

➤ OpenAMP today

- Multiple licenses (GPL for Linux kernel, BSD for rest)
- Shared memory protocol – virtio
- Lifecycle APIs – remoteproc
- Messaging – rpmsg
- Proxy technologies to emulate Linux processes
- More in future

OpenAMP Software Simplifies Deployment of Heterogeneous Systems



➤ Provides a Layer for Applications

- Standard API's that allow applications to be ported across processors and operating systems

➤ System Development

- Provides a wide range of capabilities needed to deploy applications across asymmetric computing elements

➤ Inter-OS & Inter Processor Communication

- Send messages back and forth

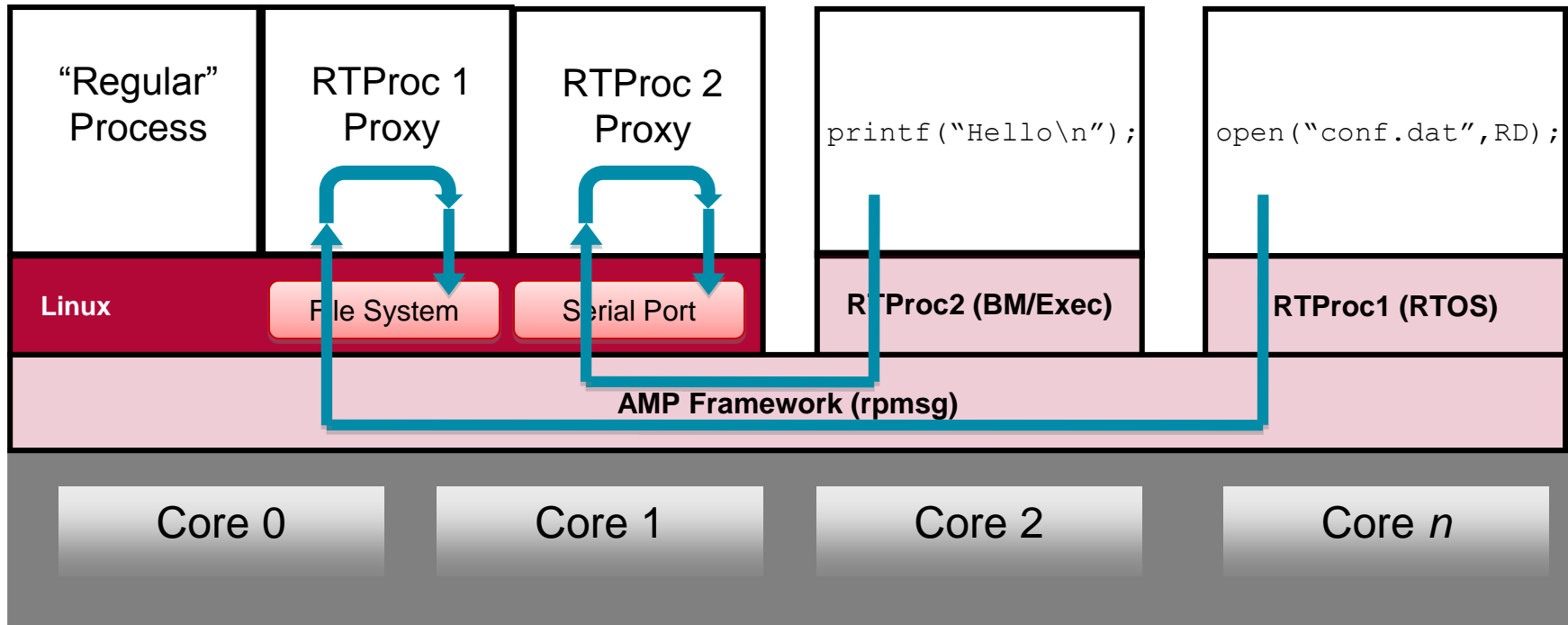
➤ OS Management

- Provides booting/rebooting of processors

RT Process – Make AMP Look Like Linux

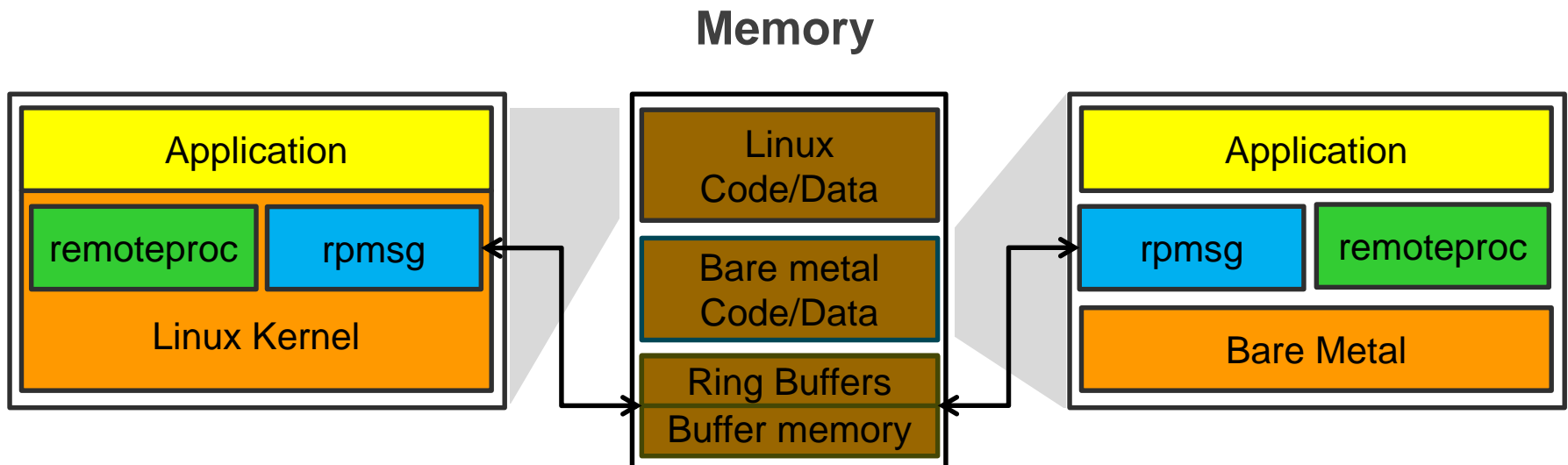
➤ Use standard Linux process concept to manage other OS

- Compile application meant to run on RTOS linking to RTOS and OpenAMP libraries
- Put application executable (ELF file) in the Linux file system
- Boot Linux as normal on main processors
- Start application through special OpenAMP launch program, runs in supervisor mode
- Handles select system(i.e. open/close/read/write/...) calls through RPC
- Terminate RT process like any other process



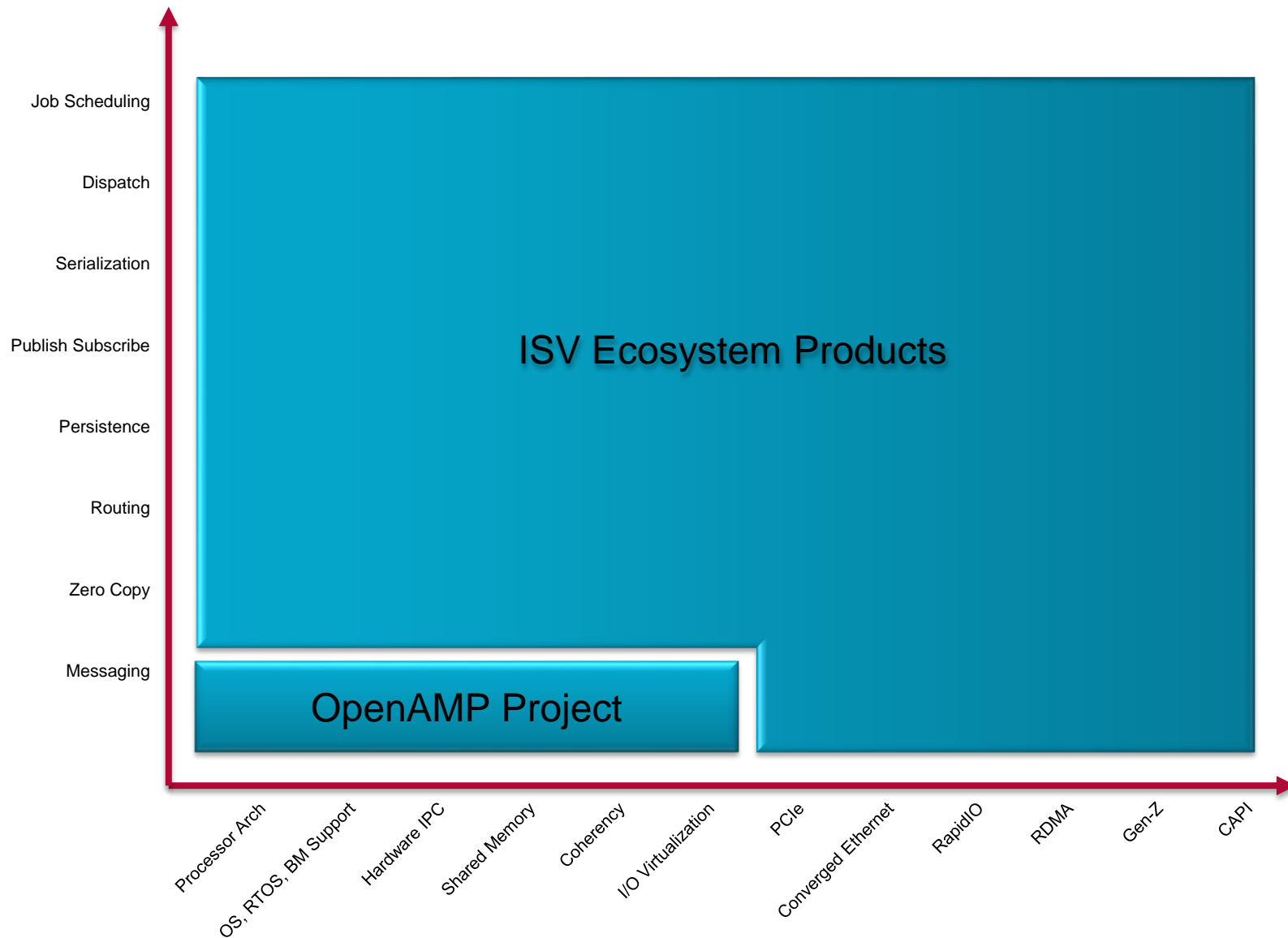
Current OpenAMP Structure

- **Management** and **messaging** frameworks are two key components
- The **management** framework is named **remoteproc**
- The **messaging** framework is named **rpmsg**



OpenAMP Gaps

Landscape Overview



OpenAMP Needs

➤ Maximal Leverage of Hardware Capabilities

- Processor capabilities
- IPI, mailbox, DMA data movers
- Variety of Coherent and Non-Coherent Interconnects

➤ Flexibility Across Broad Range of Use Cases

- Life-cycle and Discovery Options (Master-Slave, Registered Peer-to-Peer, ...)
- Messaging and add-ons (RPC, Pub-Sub, etc.)
- Filesystem, Block Device, Network interfaces, and more...

➤ Consistency and Portability

- Across threads and processes (Linux, RTOS, and Baremetal)
- Across Xen/KVM guests (Linux, RTOS, and Baremetal)
- Across processors (A53, R5, MicroBlaze)
- Across interconnects (AXI, PCIe, and more)

Current OpenAMP Design Gaps

➤ **Targets a narrow problem domain**

- Strictly master-slave, no peer-to-peer; Linux is master only
- Linux support oriented towards kernel clients; not user-space applications
- Remoteproc/rpmsg only; cannot support MCAPI, XenBus/XenStore, etc.

➤ **Lack of consistent API**

- Raw device/ioctl interface in user-space Linux
- BareMetal interfaces do not carry into user-space Linux

➤ **Structural problems**

- Zynq/ZynqMP code splattered all over; includes standalone library code, etc.
- Not modular/composable
- Not self-test capable

Evolving OpenAMP

➤ Inclusive, not exclusive

- Provide a toolkit, not enforce a policy
- Support multiple usage models: master-slave, hierarchical, peer-to-peer, ...
- Support multiple APIs, Protocols, and Platform architectures

➤ Focus on user-space applications, not kernel

- Use existing kernel infrastructure for user-space I/O; follow design patterns of DPDK/ODP/...
- Avoid need for pushing new kernel ABIs upstream (rtpmsg device/sockets/...)

➤ Support linux-generic target

- User-space inter process/thread comms using VirtIO (SHM rings, futex for kick/notify)
- Replaces Zynq/ZynqMP as primary target; allow removal of extraneous baremetal code from OpenAMP
- Necessary for self-contained self-tests within the OpenAMP code base

➤ Composable components

- Build out reusable and pluggable components with well-defined class based interfaces
- Components configured and selected through Autoconf/Cmake like mechanisms??

➤ How to manage shared devices. E.g. timer, interrupt controller, and so on

- Will need to be put in the documentation on this constraint

OpenAMP Components

APIs

- MCAPI, MRAPI, MTAPI, ...
- Baremetal RPMSG APIs
- DPDK PMD, ODP PktIO, ...
- TCF? CF? Others?
- ...

Protocols

- Virtio Protocols (Rpmsg, block, plan9, serial, VirGL, ...)
- Registry Protocols (SHM, DBUS, ...)
- Other Protocols (TCF?) ...

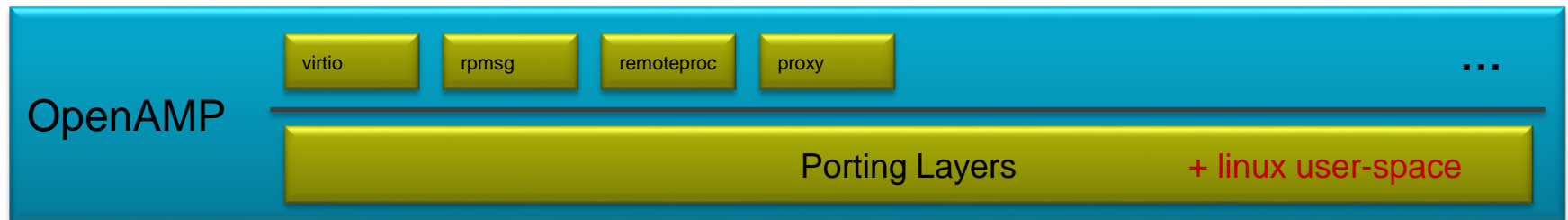
Channels

- Memory rings (SPSC, SPMC, MPSC, MPMP)
- Virtio
- Unix sockets; TCP/IP (ZMQ? Other?)
- ...

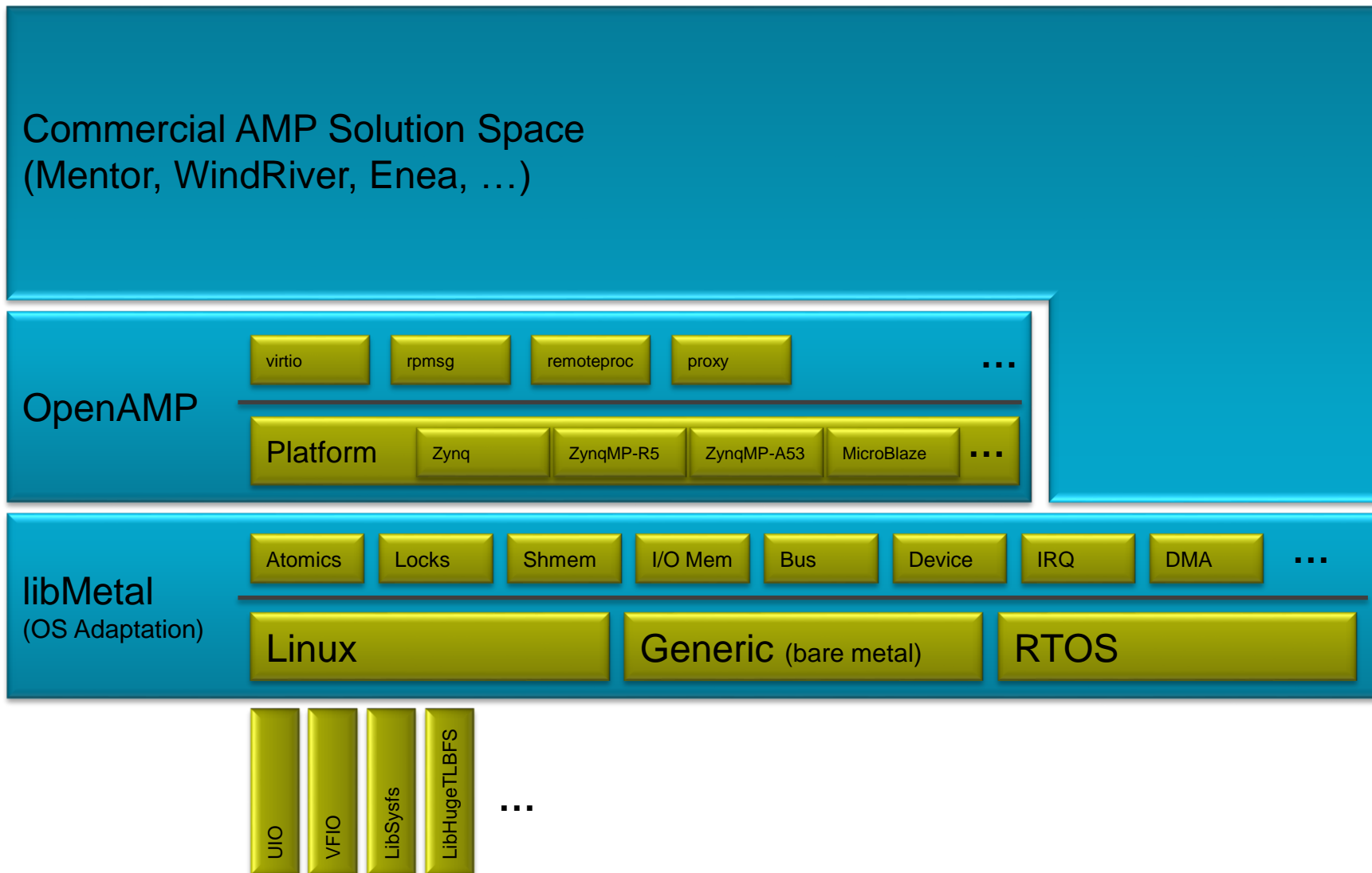
Platform Support

- Shared Memory (Raw physical memory, Linux-SHM. hugepages, CMA, dma-buf, ...)
- Events/Notifications (pipe, futex, WFE/SEV, Mailbox/IPI, ...)
- Control (power mgmt., heartbeat, ...)

Evolving OpenAMP Architecture (Available)



Evolving OpenAMP Architecture (Next)



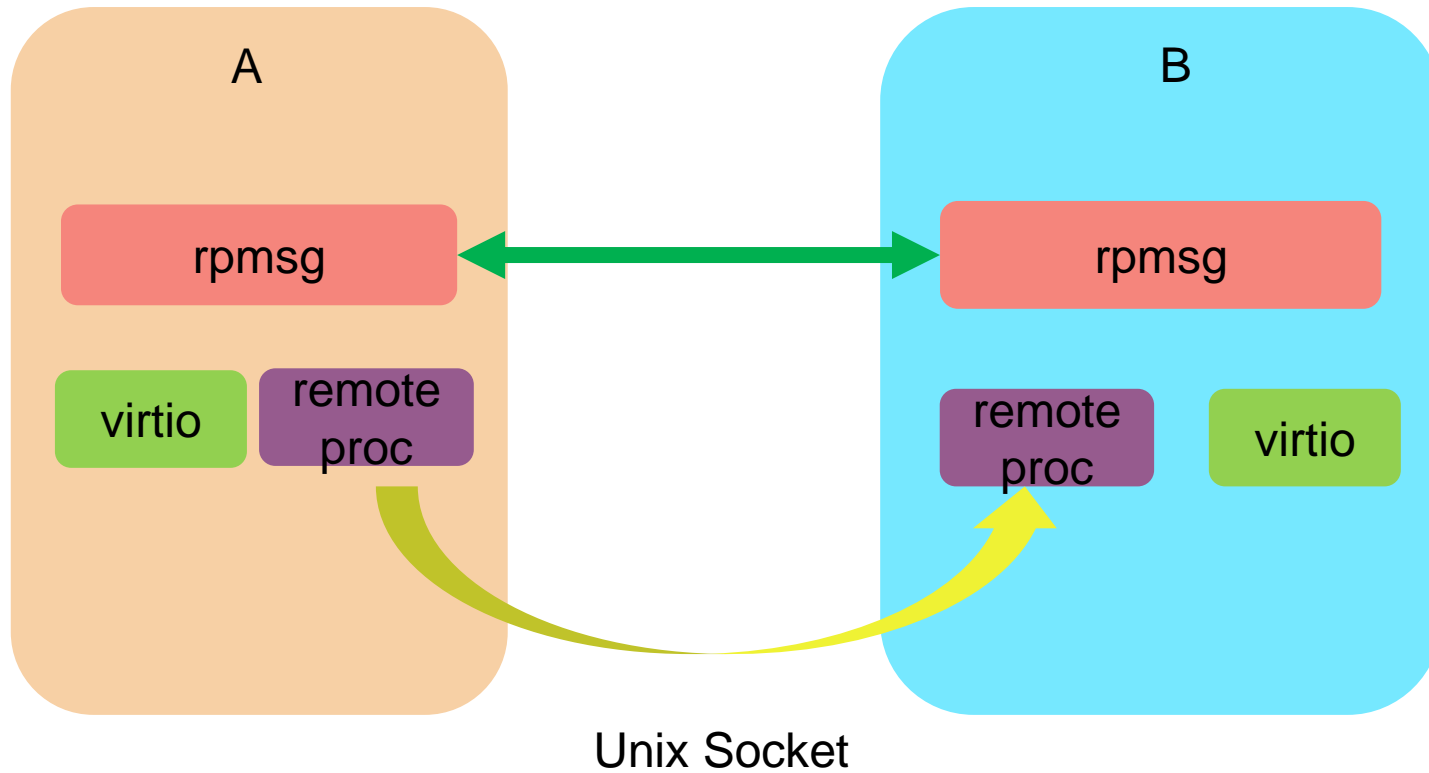
What's going into Next Release (By the end of April)

- **Release Branch will be created by March, only bug fixes will go in.**
- **OpenAMP source code restruct**
 - Obsolete what's not belong to OpenAMP library, such as the OpenAMP app dependent libraries.
 - Clean Zynq/ZynqMP dependent code from the library
 - Easier for non-master-slave model
 - A more centralized place for platform specific implementation
 - Easier to add support for more compilers
- **OpenAMP on Linux Userspace (Not in)**
 - Easier for Linux user applications to use OpenAMP
 - Easier for Users to try OpenAMP on Linux
 - One step forward to application focus, don't need to worry if the communication parties on the same system or not
 - Easier for upstreaming
- **LibMetal (Not in)**
 - Legal review
- **Test suites?**
 - The existing basic applications demos: function test suite, echo_test, matrix multiplication, and rpc demo
 - Welcome contributions
- Support flat binary format in remoteproc
- Linux kernel changes: sysfs for messaging
- Post features for Oct release by the end of March

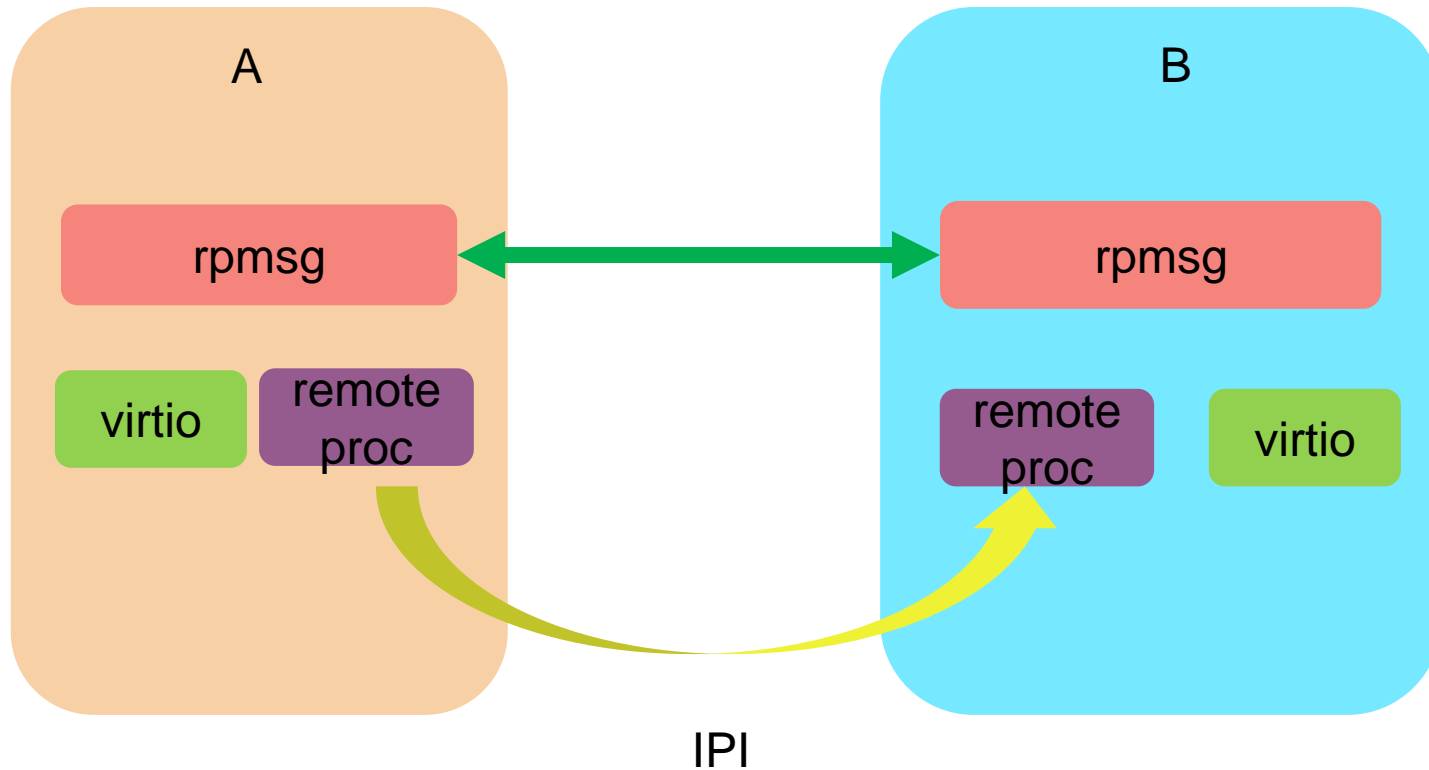
OpenAMP Roadmap

- Post features for Oct release by the end of March
- Features in roadmap:
 - Libmetal (2016.10)
 - Linux userspace support (2016.10)
 - 64bit address space support
 - Support flat binary format in remoteproc
 - Linux kernel changes: sysfs for messaging

OpenAMP over Linux processes



OpenAMP Between Linux process and Another processor



NXP OpenAMP Modification

➤ Notes:

- NXP introduces introruce queueing rpmsg APIs
- Another option from Cyril: in the callback, can transfer buffer ownership to application