XILINX
ALL PROGRAMMABLE™

Libmetal and OpenAMP

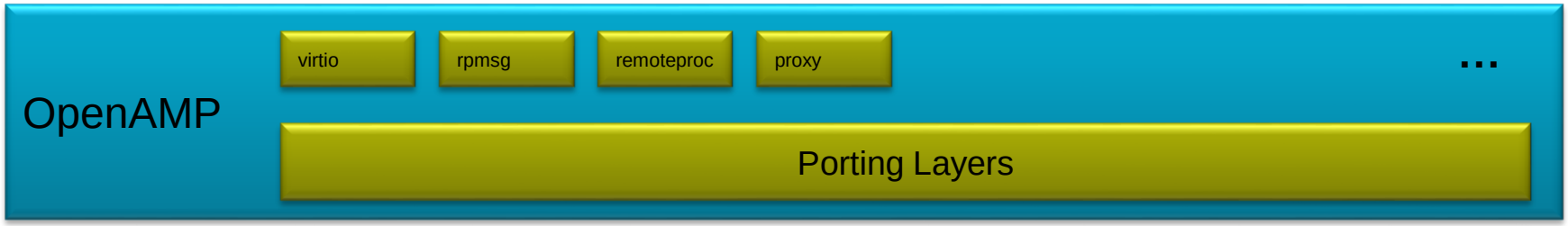
Libmetal Motivation

Why Introduce Libmetal?

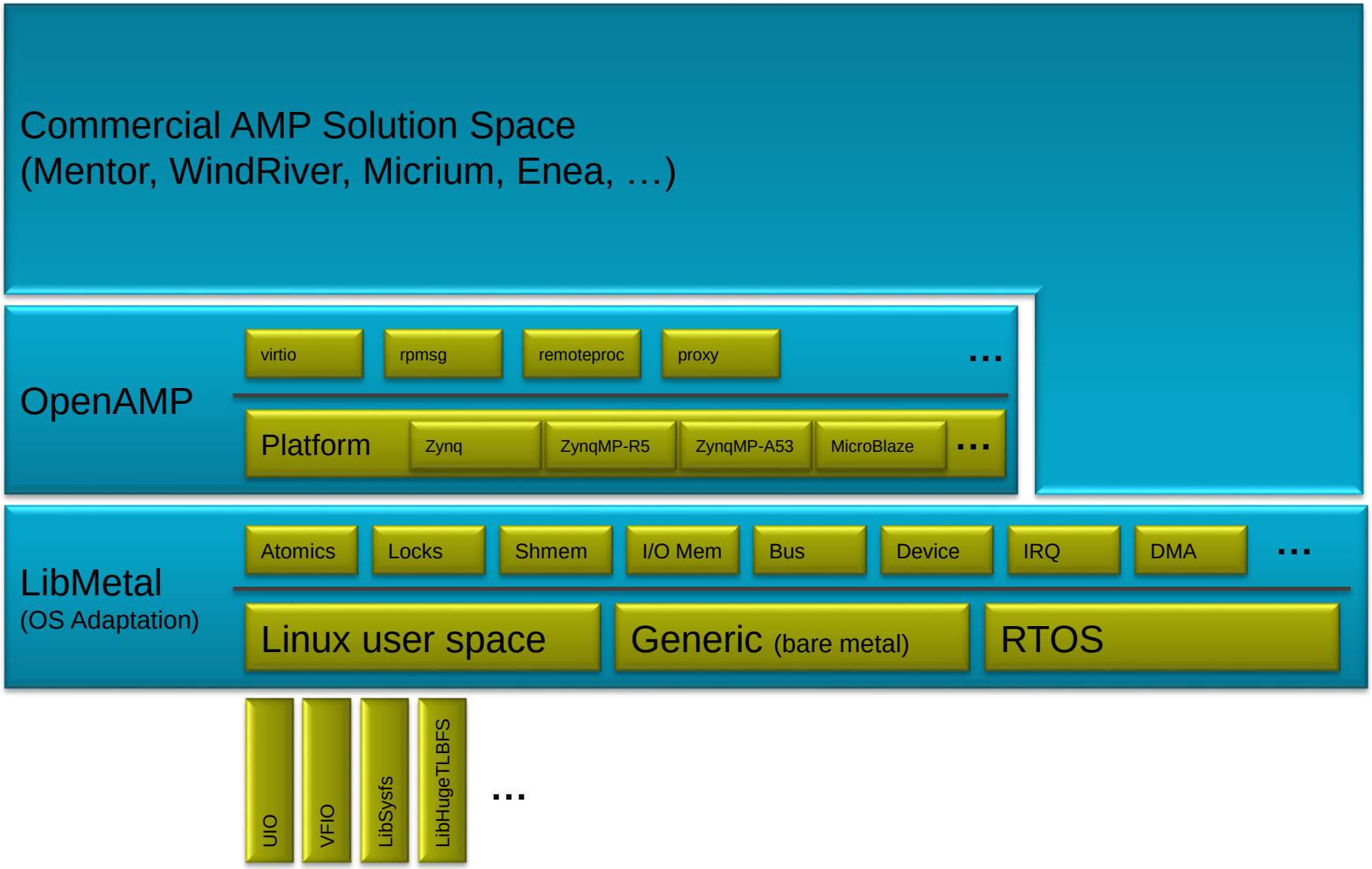
- Remove OS / compiler specific details from OpenAMP repo
  - Make it much easier to port OpenAMP to a new OS and/or HW
    - Minimum implementation supports flat memoryspace RTOS's
    - Optional features for OS's with MMU/process support
  - Avoid forking OpenAMP for different OS's
- Implement OpenAMP libraries in Linux user space
  - It is very hard to upstream rpmsg changes
    - The current rpmsg/remoteproc flow always requires Linux kernel to load the slave and Linux kernel has to be the rpmsg master
    - It is hard to upstream changes to this kernel flow
  - But still remain backwards compatible with Linux kernel implementation
- Light weight AMP solution
  - Libmetal provides device access, interrupt handling and memory request APIs
  - Allows for porting new APIs (e.g. MCAPI) once and get them running on all platforms
  - Users can use libmetal APIs directly to build their specialized AMP solution

How OpenAMP use Libmetal

Current OpenAMP



Evolving OpenAMP on Top of Libmetal



OpenAMP on Linux

- Port rpmsg/virtio to Linux userspace
  - Reduce userspace to kernel space memory copy
  - Allow Linux to run as rpmsg slave
- Enable rpmsg between Linux processes
  - Easier for testing
- Be backwards compatible with rpmsg kernel implementation
  - OpenAMP on RTOS/baremetal will still work with rpmsg/remoteproc in Linux kernel
  - New feature needs to have a new feature bit in virtio specification. And will need to negotiate during rpmsg channel setup.

OpenAMP on Linux User Space