Architecture Musing

Cloud/SDN

Embedded/IoT

Traditional

Wrapper Facade

Containers/VM’s

AL (Network, Crypto, Disk, ...)

Intel

Cavium

Freescale

FPGA etc ...
AL Goals

• Decouple the innovation boundary
  • Allow network application vendors to innovate above the line without having to build custom solutions for each HW vendors processor, SoC or NIC.
  • Allow processor, SoC, NIC and Crypto vendors to innovate below the line without having to do something special for each network application vendors implementations.

• Increase Velocity, Reduce Effort and Improve Quality
  • Enable network application vendors the ability to rapidly prototype in a given environment without significant investments in time and engineering effort.
  • Define an abstraction that allows vendors on each side of the boundary to focus on hardening of the pieces they own
  • Network application performance portability
AL Requirements

- Common & Open Interface, supporting:
  - Different Vendor IO architectures
  - Leverage Hardware Offload if available
    - Cryptography and security protocol assist
    - Classification and packet steering
    - Time measurement and event management
  - Platform Diversity
    - Wide range of performance and cost points
    - Embedded dedicated devices and Data Center server environments
- Efficient model for packet movement
  - North/South and East/West
- Hypervisor Agnostic
AL Requirements

• Roles and responsibilities clearly defined
• Clearly defined model/semantics
  • Functional Characteristics (this tends to be covered well)

• Non-functional Characteristics (not so obvious but kill you)
  • Security for deployment (least privilege)
  • Performance (this is what most people concentrate on)
  • Extensibility (how can we extend but not change the interface)
  • Management and instrumentation
  • Configuration, deployment and tuning
Finally (well almost)

- Portability across systems
  - Source Code
  - Performance
- Good Documentation
- Test Framework