



---

Delivering Heterogeneous Accelerators for the Data Center and Edge

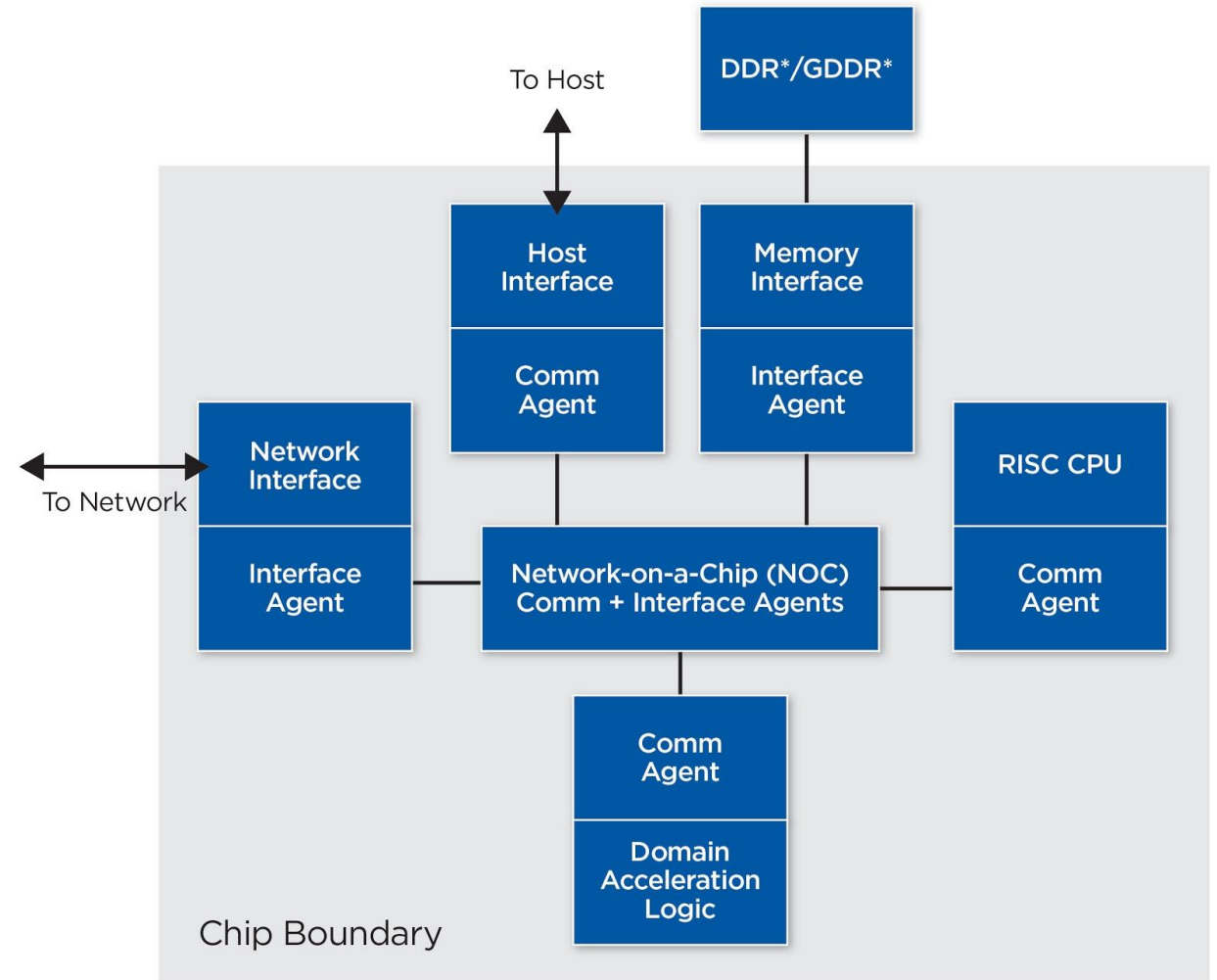
## **ODSA: White Paper, Reference Architecture and Fabric**

*Bapi Vinnakota*



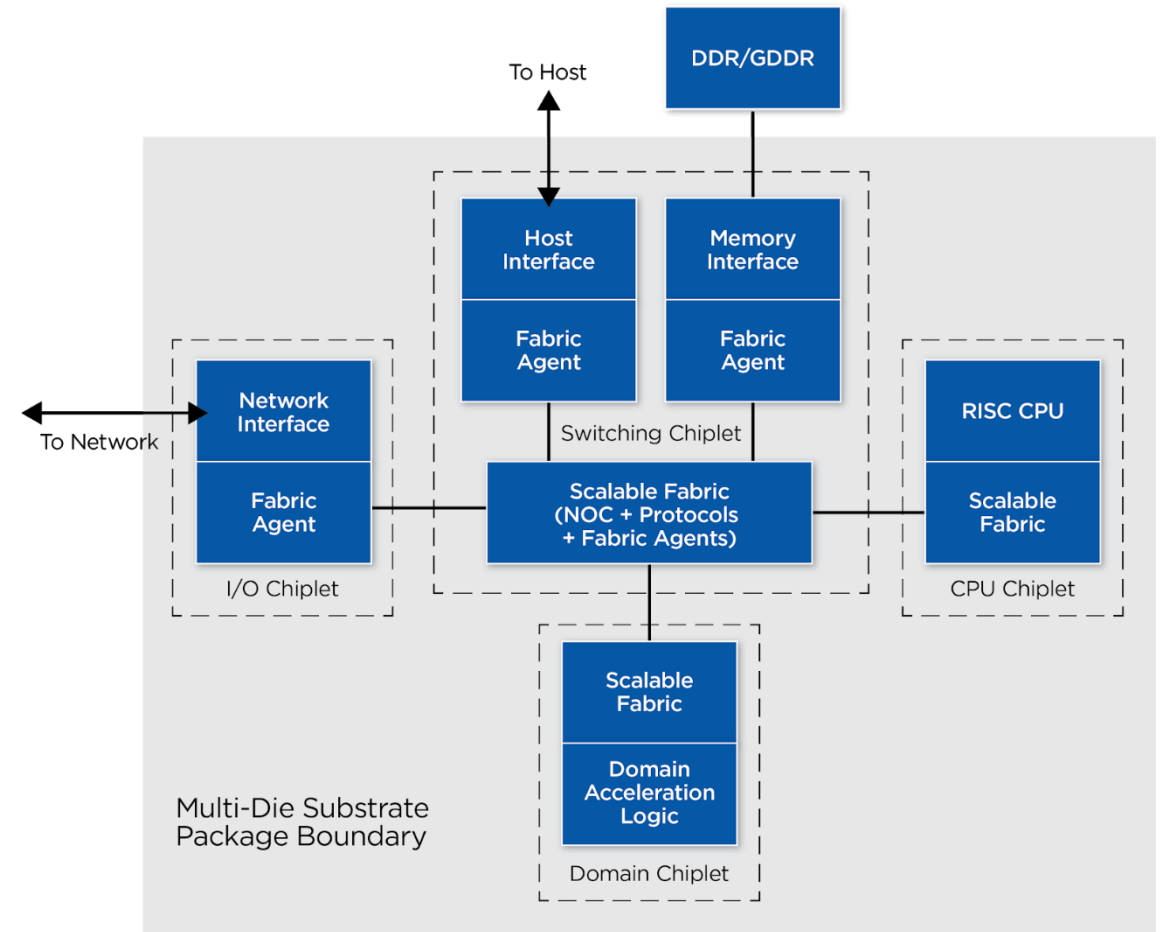
# DOMAIN-SPECIFIC ACCELERATORS

- Host-attached programmable logic optimized for an application domain
  - Tensorflow, Netrnome NFP, Crypto, IoT,...
- Domain-specific accelerators contain lots of generic logic ~35-45% of silicon area, development time
  - Network, Host, Memory Interfaces
  - General-purpose CPUs
  - SRAM, interconnect
  - Domain-specific logic works in coordination with host and/or CPU SW
- Ideally
  - Investment in a DSA should be limited to the domain-acceleration logic
- In reality
  - Buy IP for the “non-core” parts, spend \$\$’s test and integration



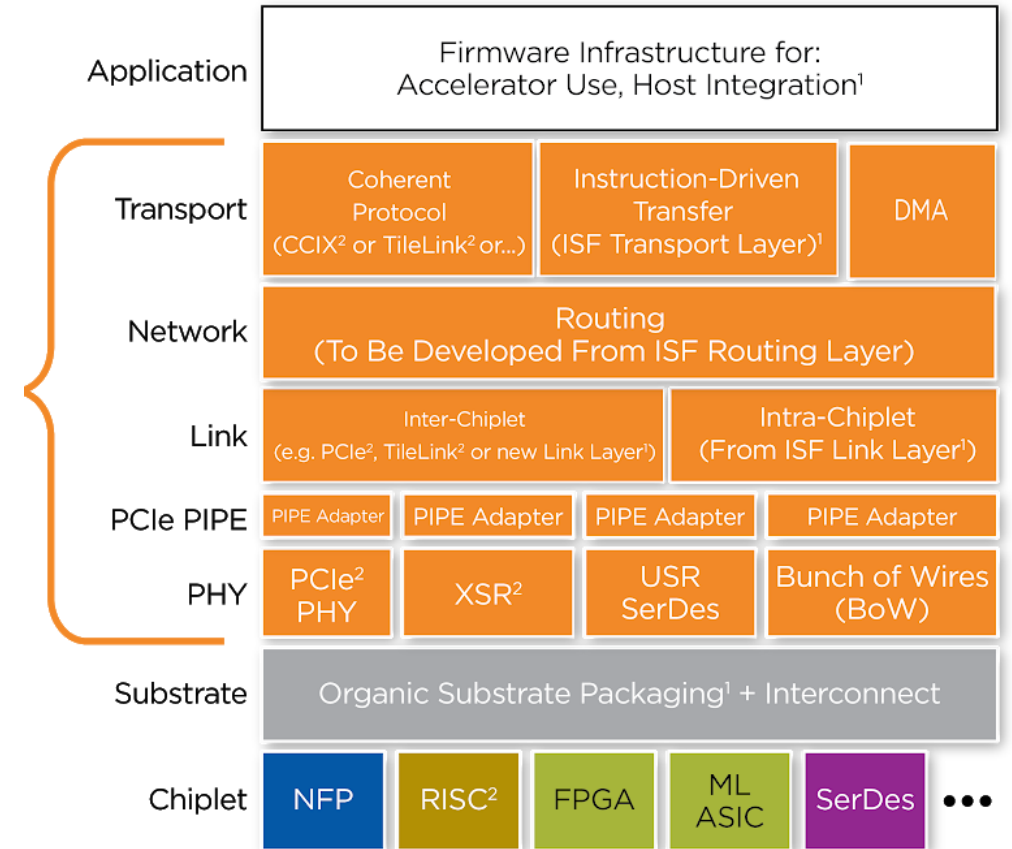
# MULTI-CHIPLET REFERENCE ARCHITECTURE FOR DSA

- With this architecture
  - Build for a new domain with new domain acceleration logic
  - Reuse chiplets instead of IP
- Also addresses connectivity issue
  - Who do I connect to when I build a chiplet
- How do make this work?
  - What is the architectural interface – memory transaction
  - No clear choice at the PHY layer. A design may use multiple PHYs



# THE ODSA STACK

- Memory is the architecture interface
  - Coherence over a small area
  - Non-coherent transport over a larger area
- Inter Chiplet
  - PHY: PIPE interface to abstract the PHY layer, multiple PHYs
  - Link: Reuse existing Link layer – which one?
- Intra Chiplet
  - PHY: Simple BoW. PIPE-like abstraction?
  - Link: Reuse Netronome ISF
- Common
  - Network: Route read/write across chiplets
  - Transport
    - ▶ Cache-coherent protocol. Multiple choices. Which one?
    - ▶ Non-coherent transport
      - Classic DMA
      - Netronome ISF transport layer



<sup>1</sup> New Open IP/Specification

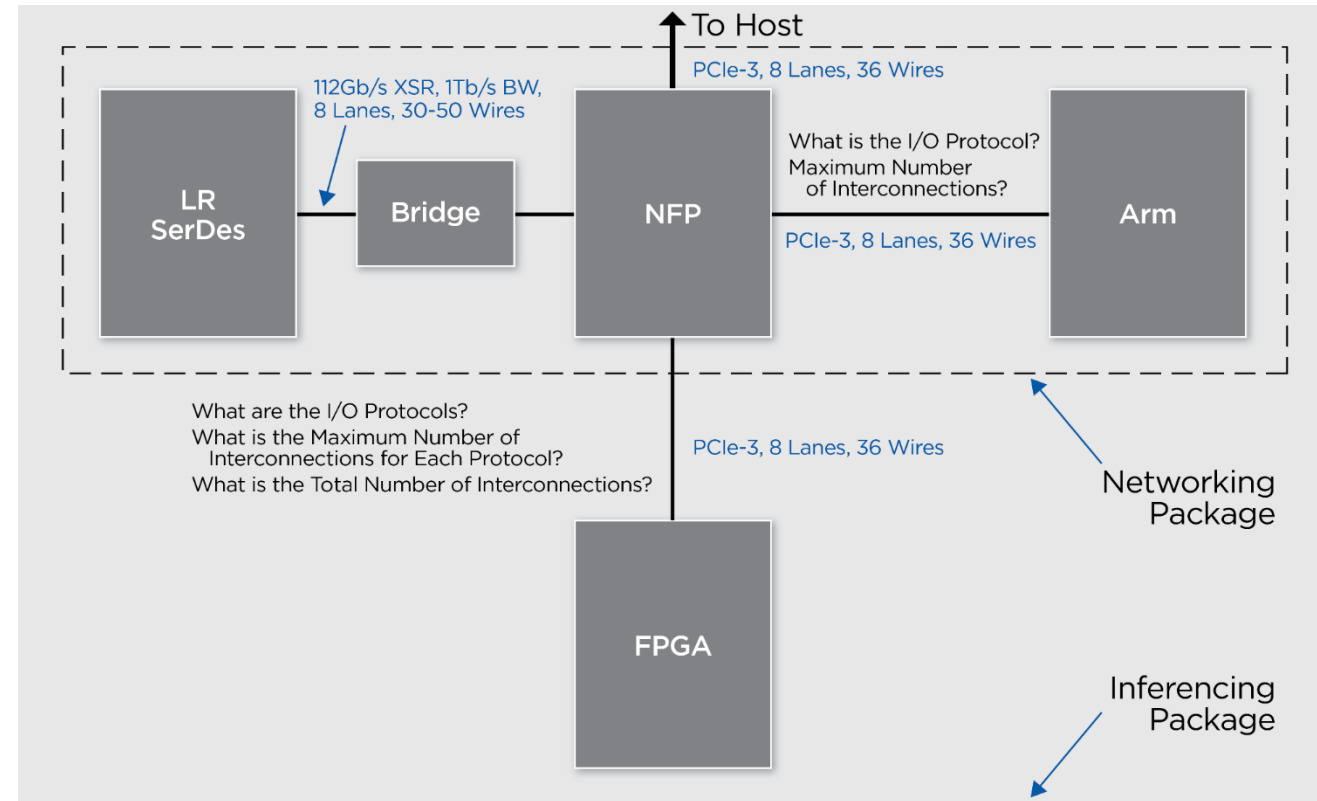
<sup>2</sup> Existing Open Standard

Source: ODSA



# BEYOND THE ARCHITECTURAL INTERFACE

- Implementation
  - Can we demo the architecture. Iron out operations, test
  - What is the effort involved in developing chiplets for the reference architecture
- Important non-technical issues
  - IP rights
  - Workflow, tools
  - Assembly and test
  - Finding an open organization
- Agenda builds off white paper
  - Level set (this session)
  - PoC with today's silicon
  - Building silicon for the reference architecture
  - Business model/open org discussion
- Aim to make tangible progress toward building something useful



# ODSA WHITE PAPER

- Broad overview of the space
  - Needs to be refined to a 1.0 document, Additional participants welcome
- Motivation for chiplets
  - Do we need any more?
- Technology proof points focused on the PHY layer and substrate
  - PHY: Multiple options Serial/Parallel, proof points from Alphawave (newer), Aquantia, Intel, Kandou
  - Substrate: Organic substrates, fiber to the package
  - Fabric: Cache coherence protocols (CCIX, Open CAPI), scalable async fabric from Netronome
- To make chiplets work, they need to behave like they are on the same chip
  - Chiplets need an architectural interface (stolen from Gabe Loh from AMD), not just the PHY
- Generic architectural interfaces are challenging, if not impossible
  - May be possible for a narrower scope

