Open Domain-Specific Architecture: Protocols to PHYs

Dave Kehlet
Sam Fuller
Jaideep Dastidar
Imran Ahmed

Legal Information

This presentation contains the general insights and opinions of Intel Corporation (“Intel”). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Statements in this document that refer to Intel’s plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel’s results and plans is included in Intel’s SEC filings, including the annual report on Form 10-K.

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

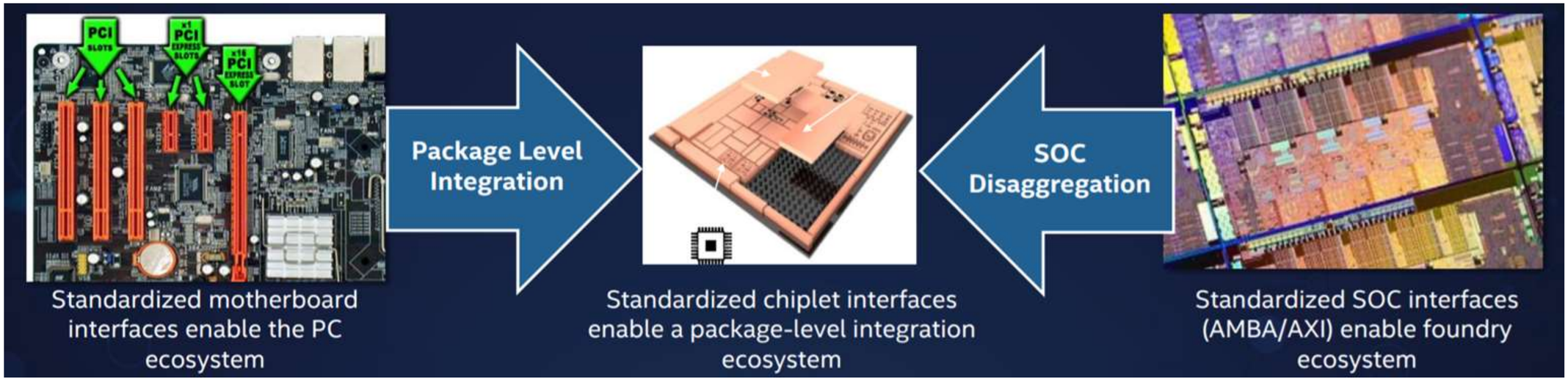
Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

† Tests measure performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Contributors and Collaborators

Imran Ahmed – AnalogX
Jaideep Dastidar – Xilinx
Kevin Drucker – Facebook
Sam Fuller, NXP
Quinn Jacobson – Achronix
Peter Jenkins – Avera Semi
Brian Kahne – NXP
David Kehlet – Intel
Mark Kuemerle – Avera Semi
Paul Mattos – Avera Semi
Dimitry Melts – Facebook
Gary Miller – NXP
Millind Mittal – Xilinx
Mats Myrberg – Microsoft
Bapi Vinnakota – OCP

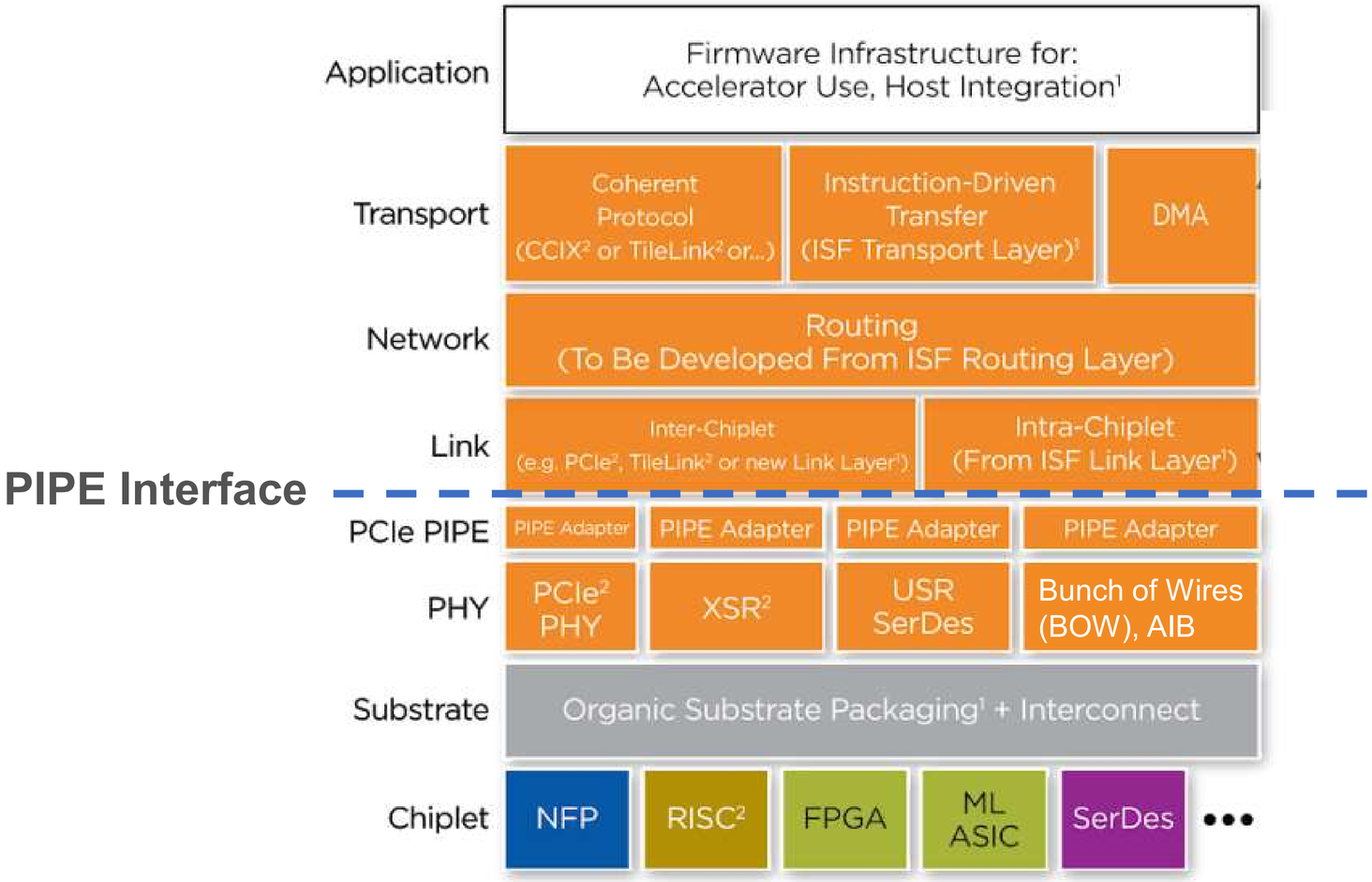
Two Broad Use Cases



Package Level Integration **SOC Disaggregation**

Use cases drive protocols!

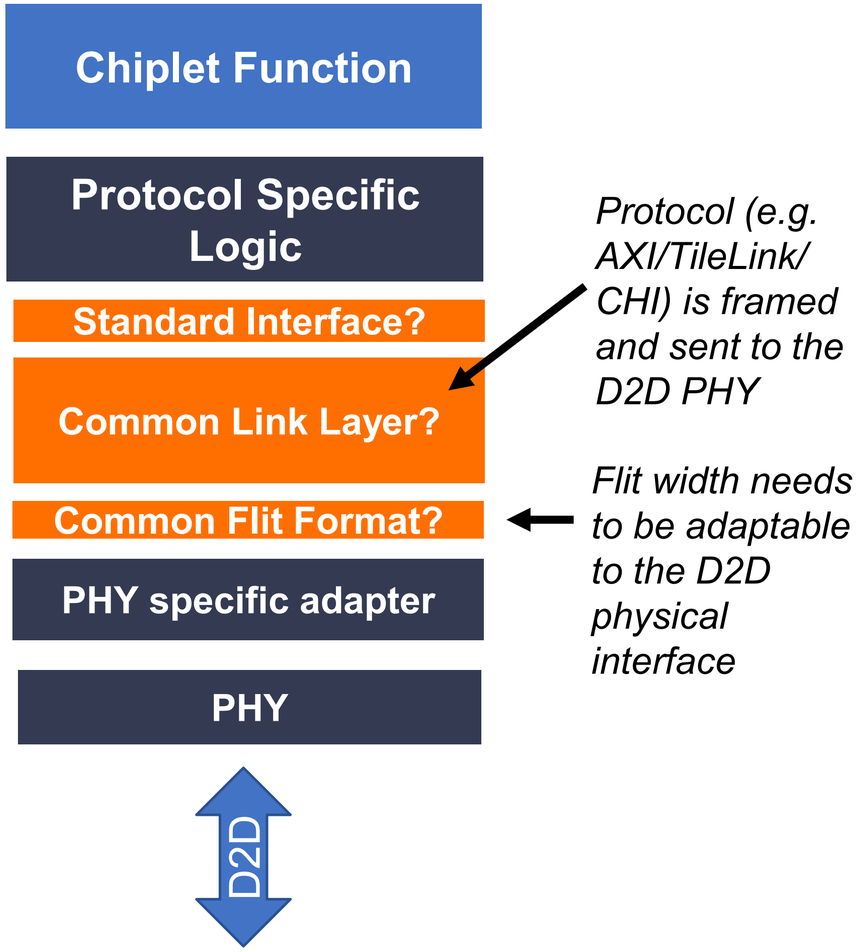
ODSA Protocol Stack – 9/2019



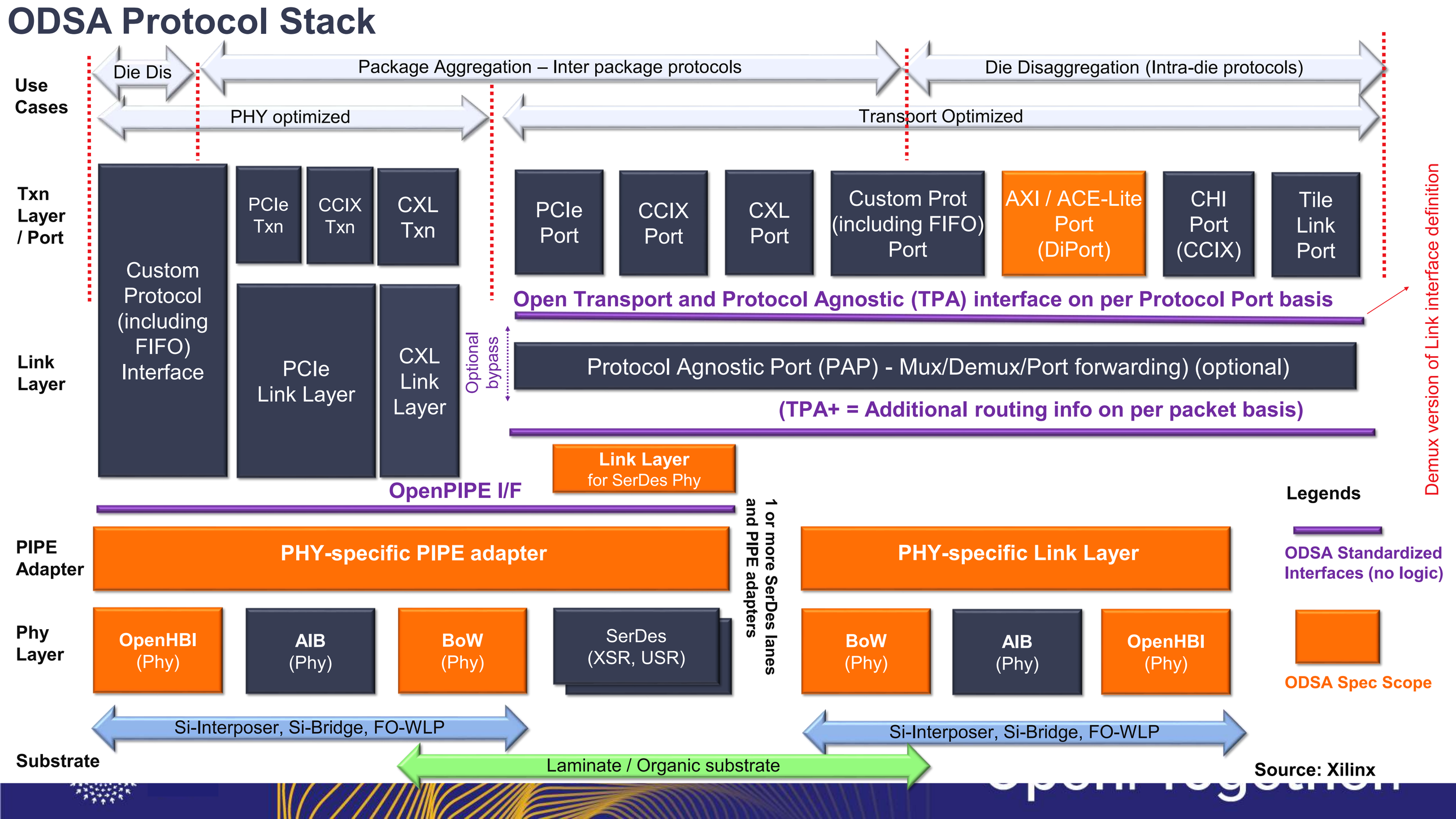
Common Link Layer Exploration

Link Layer

- Can we find a useful standard link layer interface?
- Typical link layer services are:
  - –Framing (delimit a bundle of die-to-die transfer)
  - –Flow control
  - –Error correction, detection
- Link layer interface upwards (toward application)
- Link layer definition downwards (to PHY)
  - –Flit (Flow Control Unit, a stream unit) format

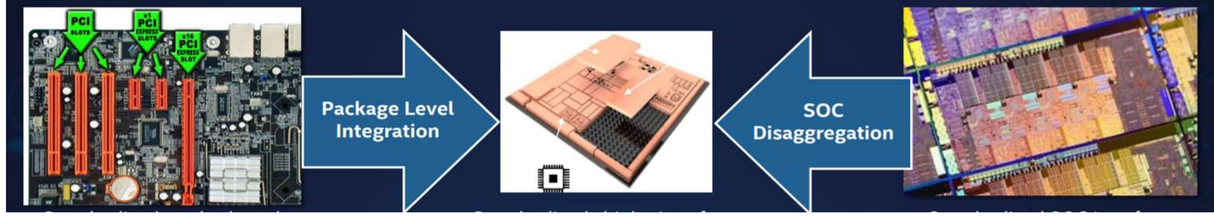


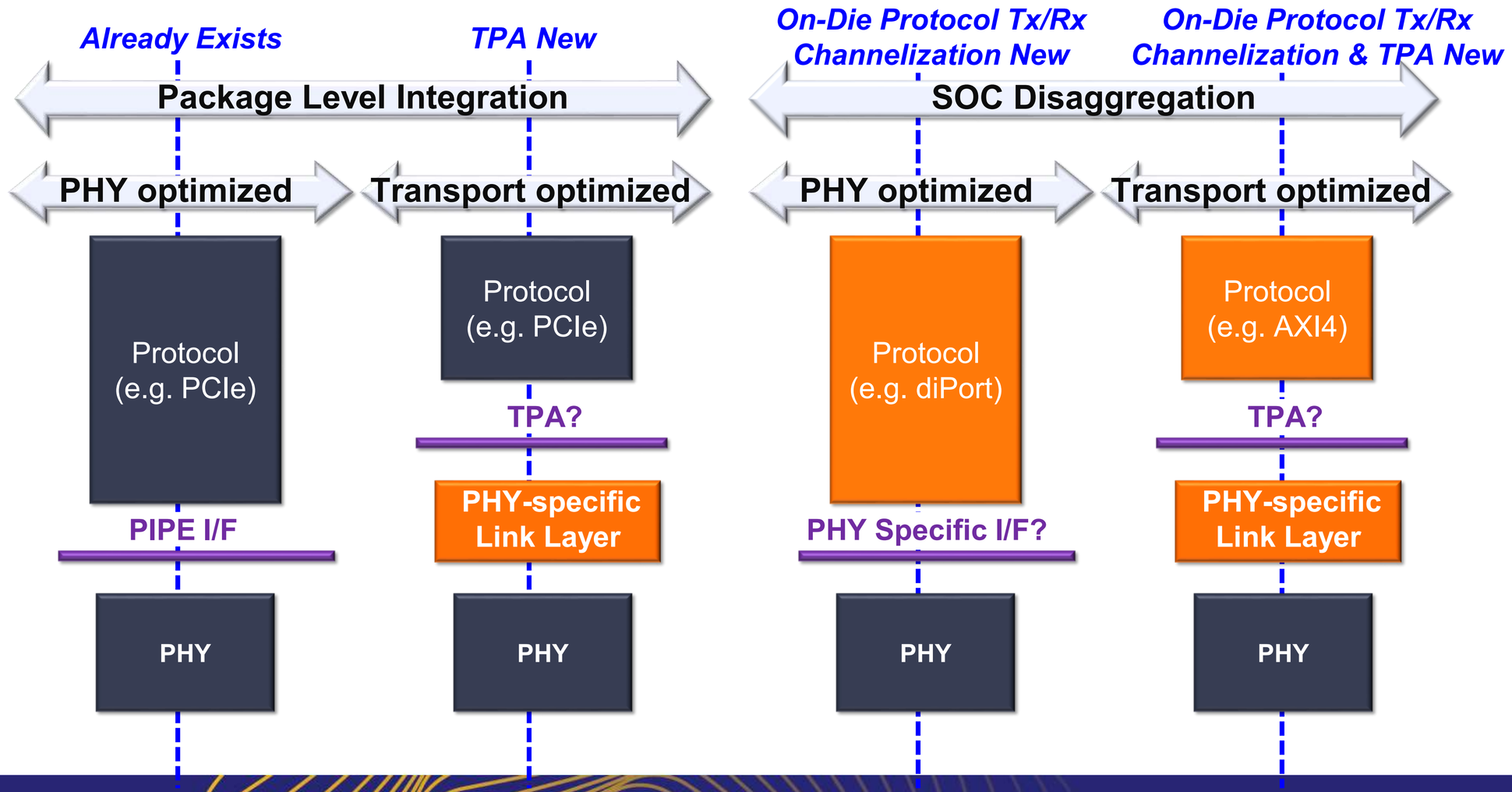
ODSA Protocol Stack



Source: Xilinx

ODSA Protocol Stack





Coming Up

DiePort Overview, Sam Fuller

- Adapts the AXI4/ACE-Lite application interface to a 2-channel die-to-die PHY

Transport and Protocol Agnostic Interface, Jaideep Dastidar

- Encapsulates and sends/receives multi-protocol packets between chiplets

PIPE Adapter, Imran Ahmed

- Mapping the industry standard PHY Interface for PCI Express (PIPE) to your own PHY

NXP’S DIPORT DIE TO DIE TECHNOLOGY

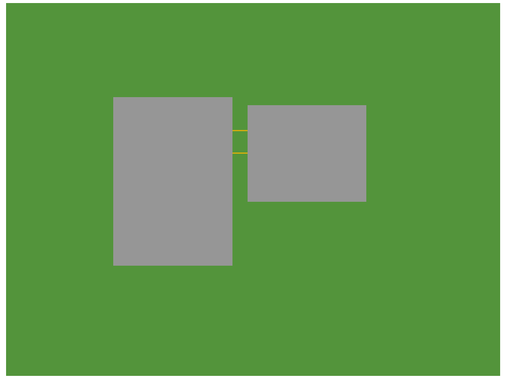
TECHNICAL INTRODUCTION TO ODSA

SAM FULLER,
GARY MILLER, BRIAN KAHNE

18.DEC.19

SECURE CONNECTIONS
FOR A SMARTER WORLD

Goals of the DiPort Technology

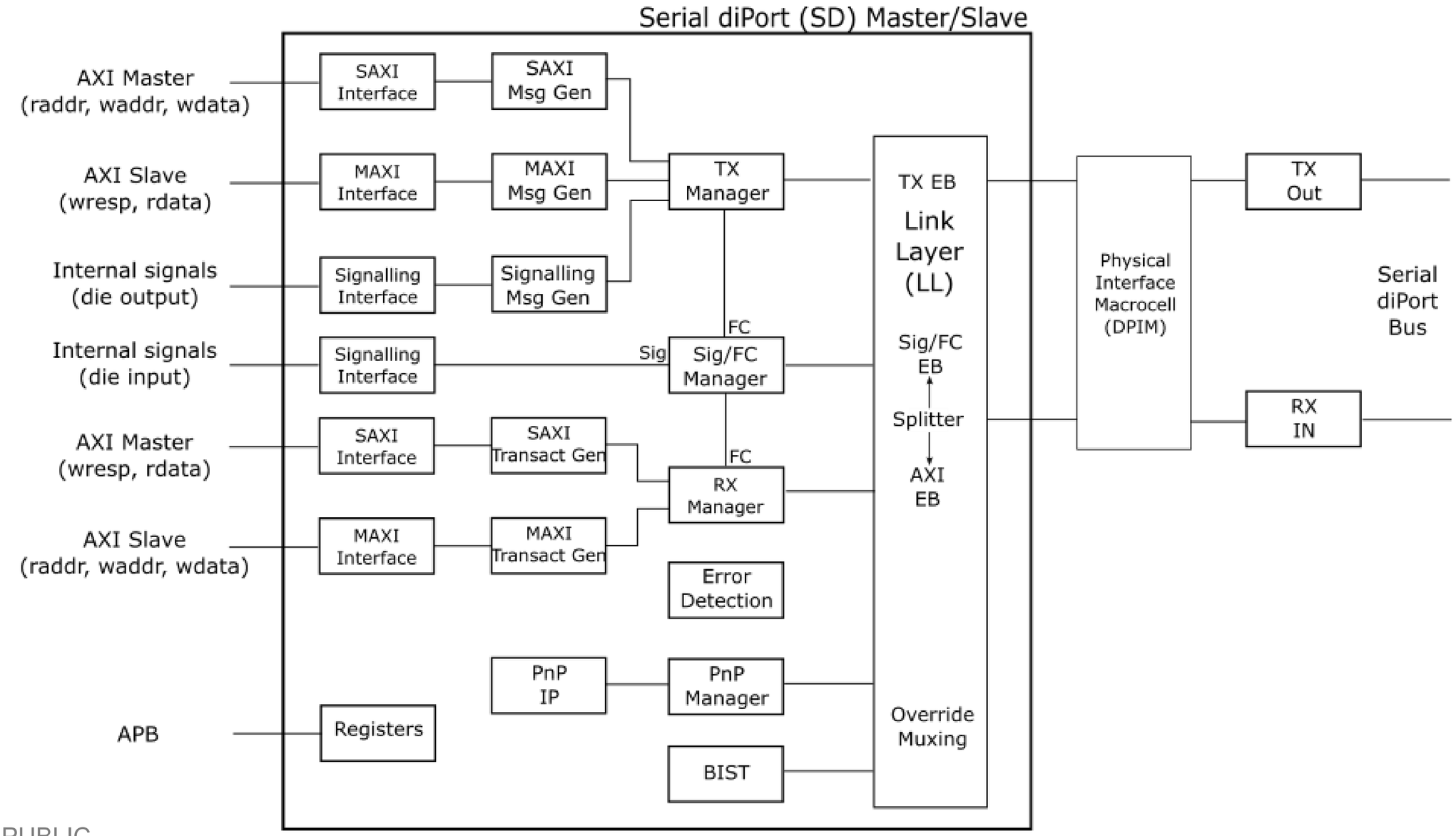


- Optimized for die-to-die communication
  - Latency, bandwidth, power, and cost
- Automatic integration of a unified memory map at start-up
- Connected die appears to software as on-die resources
- Interoperability of mixed technology die
- Critical system information and messages are automatically shared across die
- Quality of service and decoupling of latency
- Low-cost and scalable SiP interface for use with low density SiP assembly technologies
- Solve start-up, reset, frequency change, and errors for SiPs (e.g., power up/down, localized/catastrophic errors)
- Error detection, prevention and functional safety
- Robust test support
- Optional Plug-and-Play if needed for a generalized solution

Serial diPort General Features

- Provides a virtual interconnection of AXI bus between two die
- Supports signaling of hundreds of system states, messages, etc. between two die
  - Automatic replication on other die
  - Handles all clock-domain crossing for scalar and vectors
- Error detection, corruption prevention and resiliency (e.g., functional safety)
- Optimized for AXI channels protocol
  - Tracks AXI state flow with a minimal amount of AXI attributes transferred between die
  - Minimal packetizing delays for both single and burst transactions
  - Pipelining for many simultaneous reads and writes for improved latency and bandwidth
  - Expandable for ACE-Lite (or ACE) support
  - Virtual-linked/hardware-synchronized AXI bus channels between die

Architectural Overview



Overview of Messages

- Message types
  - AXI, Flow Control (FC) and Signaling (SIG)
- Message packing
  - Packed for back-to-back AXI messages
  - Packed for back-to-back SIG/FC messages
  - Close 1 clock when transitioning
- Message ID and CRC
  - 6-bit header for all messages
  - 8-bit CRC for AXI and 6-bit for FC and SIG

AXI/FC Message	# of Bytes
Write Address 32	10
Write Data 8 bits	6
Write Data 16 bits	6
Write Data 32 bits	8
Write Data 64 bits	12
Write Data Last 64 bits	12
Write Data Response	4
Read Address 32	10
Read Address 48	10
Read Data 8	6
Read Data 16	6
Read Data 32	8
Flow Control	4
Idle	2

All AXI/FC messages provide flow control

SIG Messages	# of Bytes
Signal write	6
Signal read	6

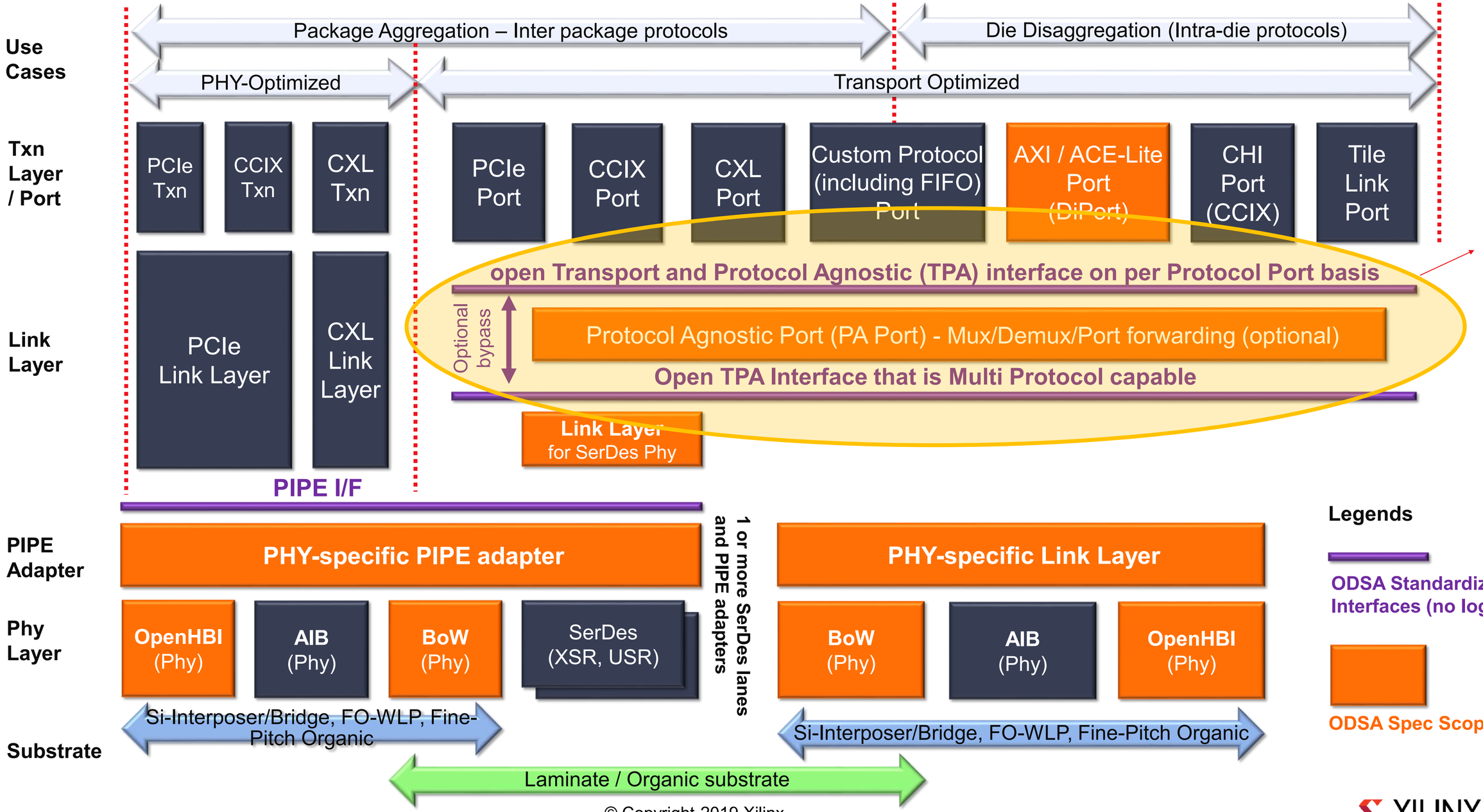
SECURE CONNECTIONS
FOR A SMARTER WORLD

Transport and Protocol Agnostic (TPA) Chiplet Interface

ODSA Workshop
Dec 18, 2019

XILINX.

ODSA Protocol Stack



Chiplet IP ecosystem

CCIX Port with TPA Port

DiePort with TPA Port

Custom Protocol with TPA Port

Tile Link with TPA Port

Chiplet IP Marketplace

TPA Port to Serial PHY + DLL

TPA Port to Parallel I/F

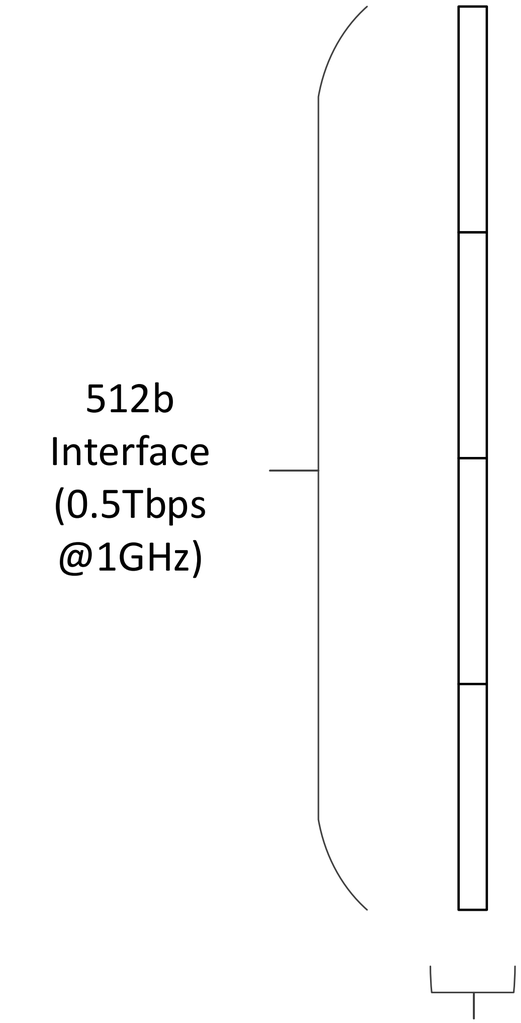
XILINX.

What the proposal covers

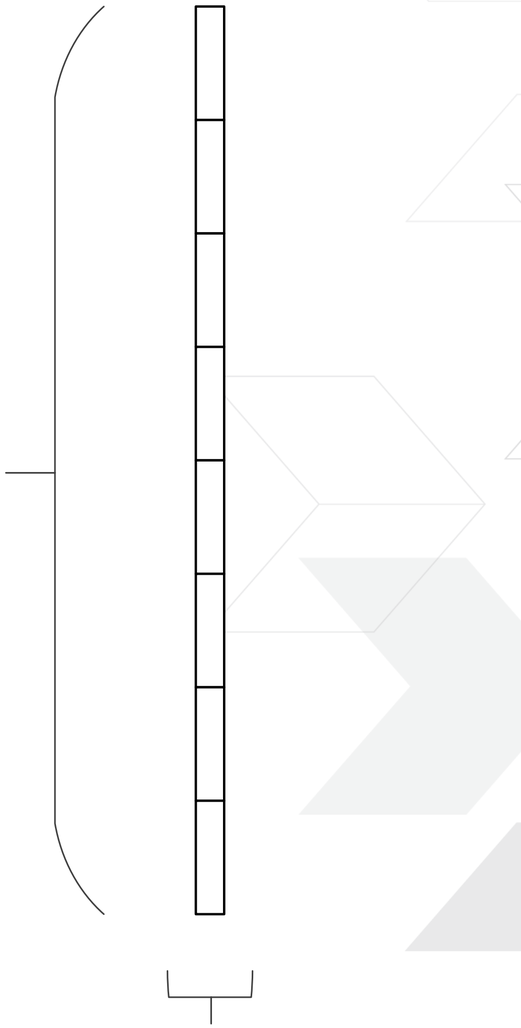
- **On-chip interface + framing between Protocol's Transaction Layer and Data Link Layer of Chip-to-Chip Interface**
- **On-chip interface + framing is the same whether Serial or Parallel Chip-to-Chip PHY and Scales across Protocol Types**
- **Standard bandwidth quanta for Interface width:**
  - Common Interface Protocol constructs at 0.5Tbps, 1Tbps, and 2Tbps
    - Normalized to 1GHz: 512b, 1024b, and 2048b interface
- **Interface definition sanity check against Protocols:**
  - AXI, PCIe, CCIX, CXL
    - PCIe, CCIX are already packetized so can map to interface framing
    - AXI, CXL are not packetized formats – ODSA can create one or consider proposals for packetization – e.g. DiePort

TPA Framing for Packetized Protocols

512b Interface with Max 4 Start/End Markers per cycle

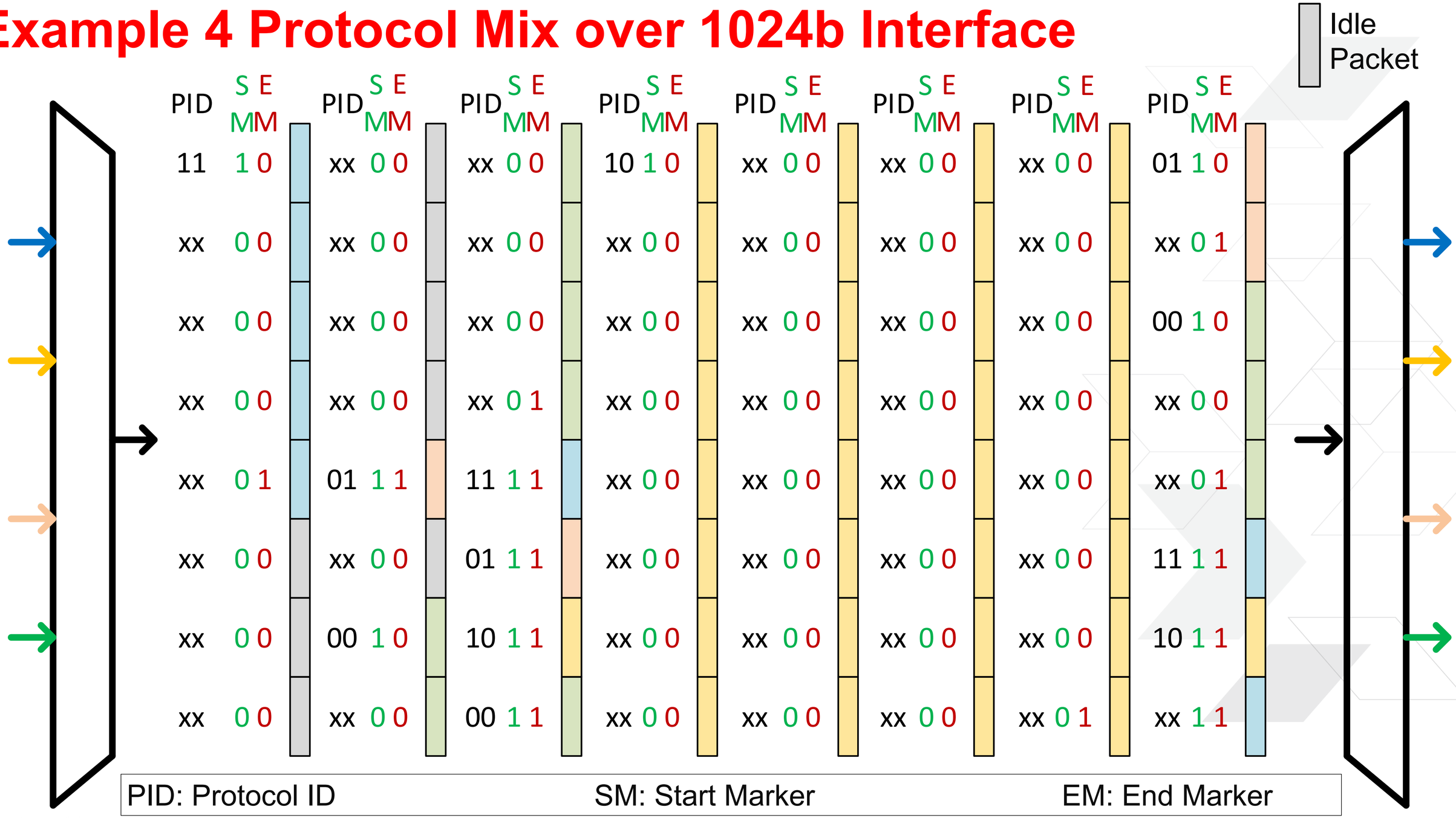


1024b Interface with Max 8 Start/End Markers per cycle

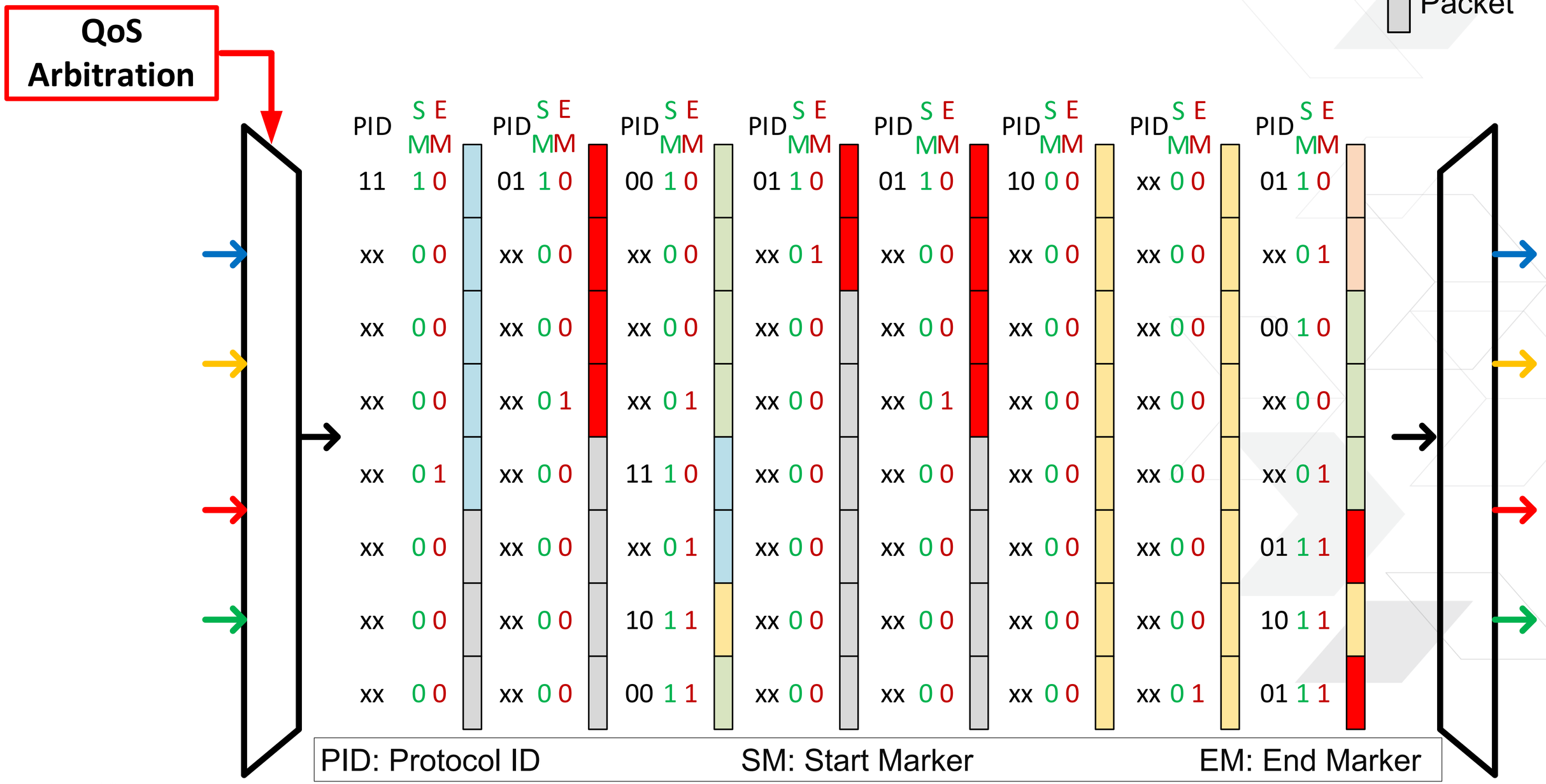


XILINX.

Example 4 Protocol Mix over 1024b Interface

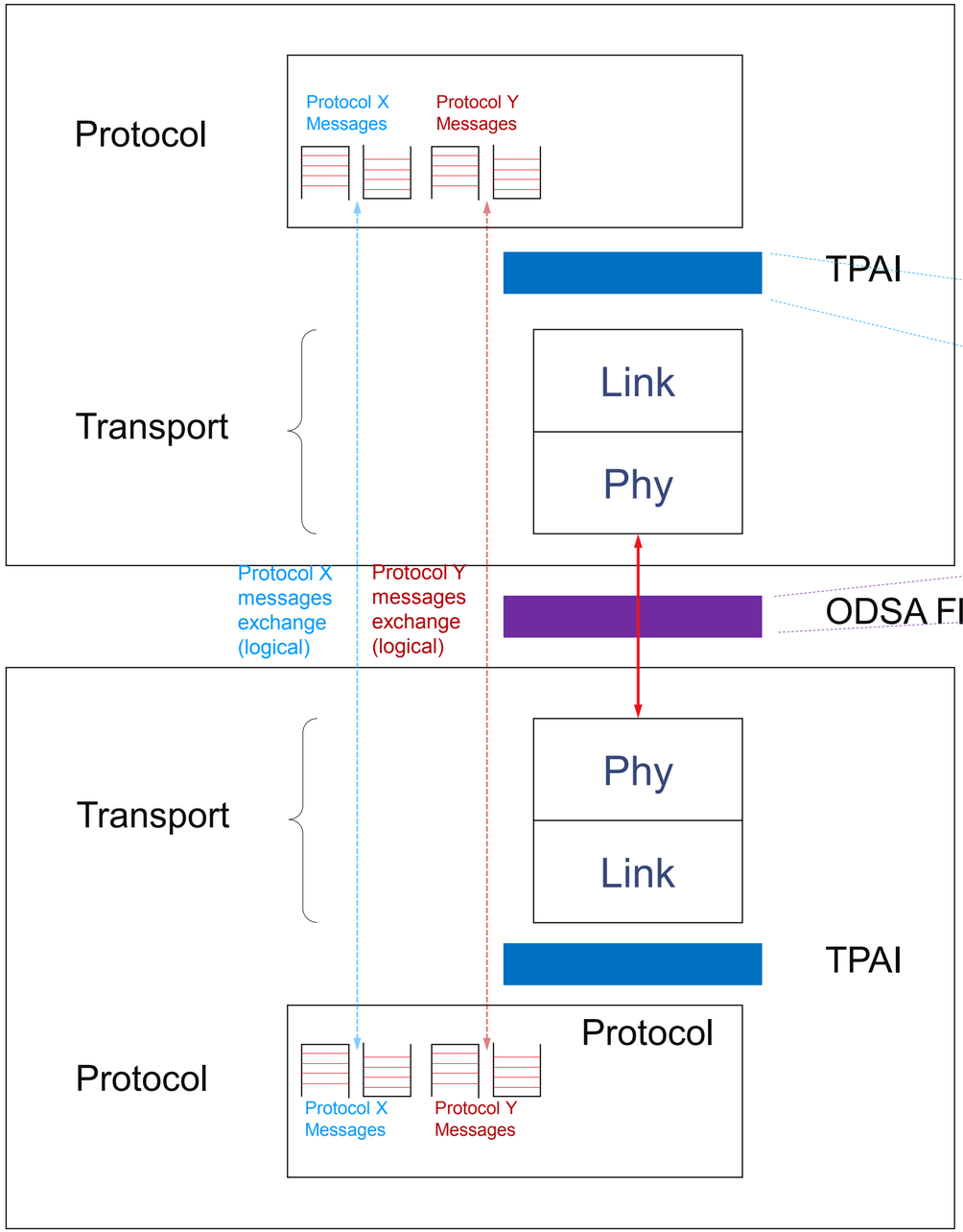


TPA with QoS Arbitration for Low Latency Protocol

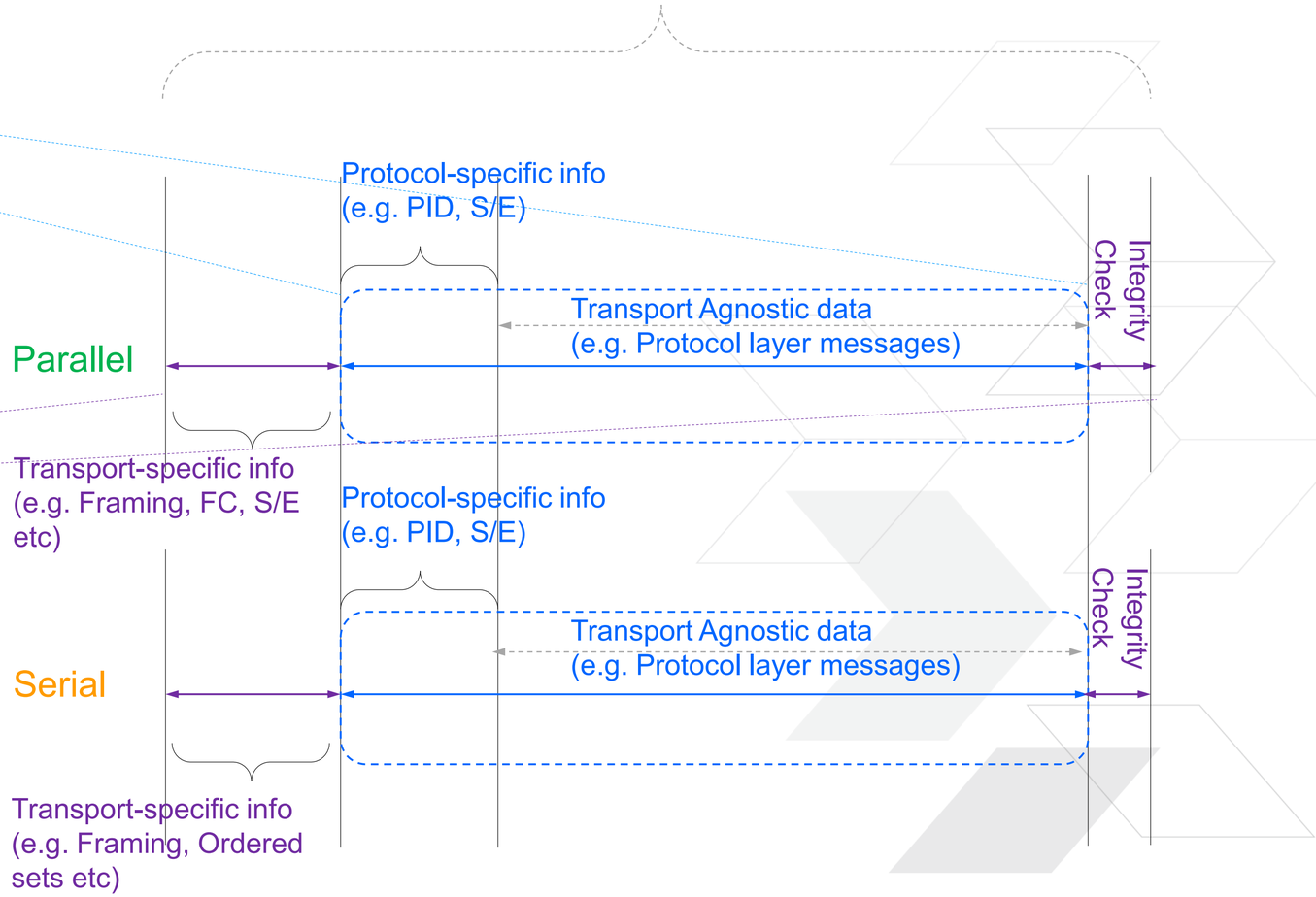


XILINX.

For lower levels, extend TPA with PHY framing and EDC



Logical View of the TPA and ODSA Flits information
These info can be packed into the Serial lanes / Parallel wires accordingly. :



XILINX.

PIPE Adapter for Chiplet PHYs

ODSA Project Workshop

December 18, 2019

Imran Ahmed

Bapi Vinnakota

Consume. Collaborate. Contribute.

PIPE adapter group contributors and collaborators

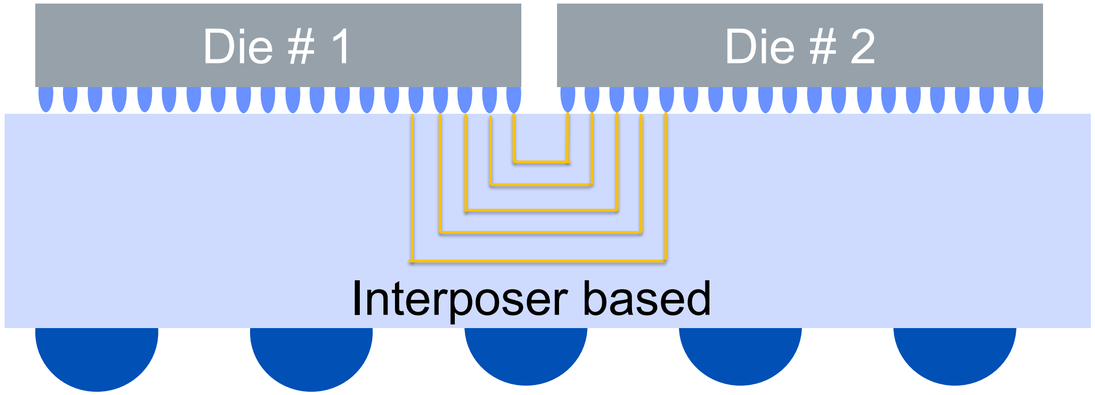
- Imran Ahmed - AnalogX
- Halil Cirit – Facebook
- Ramin Farjad – Aquantia
- Brian Holden – Kandou Bus
- Rita Horner – Synopsys
- David Kehlet – Intel
- Mark Kuemerle – Global Foundries
- Paul Mattos – Global Foundries
- Bapi Vinnakota – Netronome
- Robert Wang – AnalogX
- Jerrold Wheeler – Synopsys

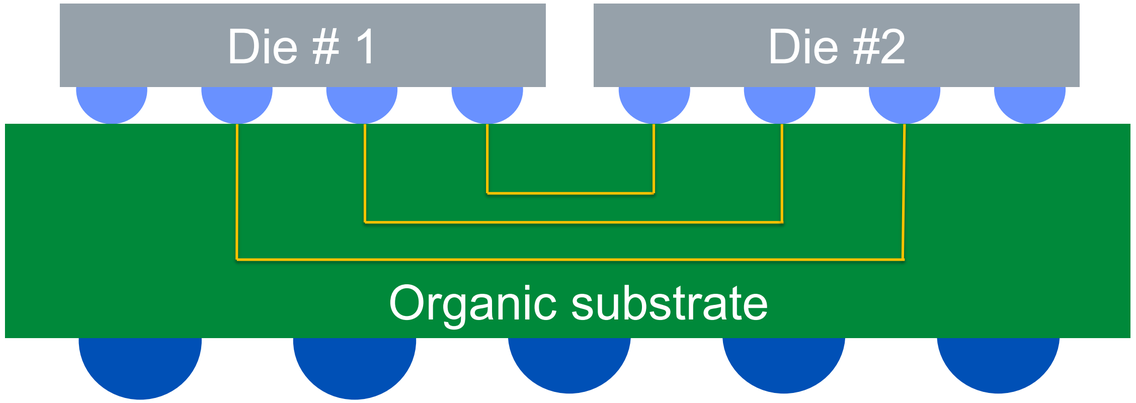
Overview

1. Why do we need a PIPE interface for Chiplets?
2. How to implement a PIPE interface for Chiplets
3. Progress on first draft
4. Next steps

Which Chiplet PHY to use?

- Choice of optimal PHY depends on application and physical interconnect



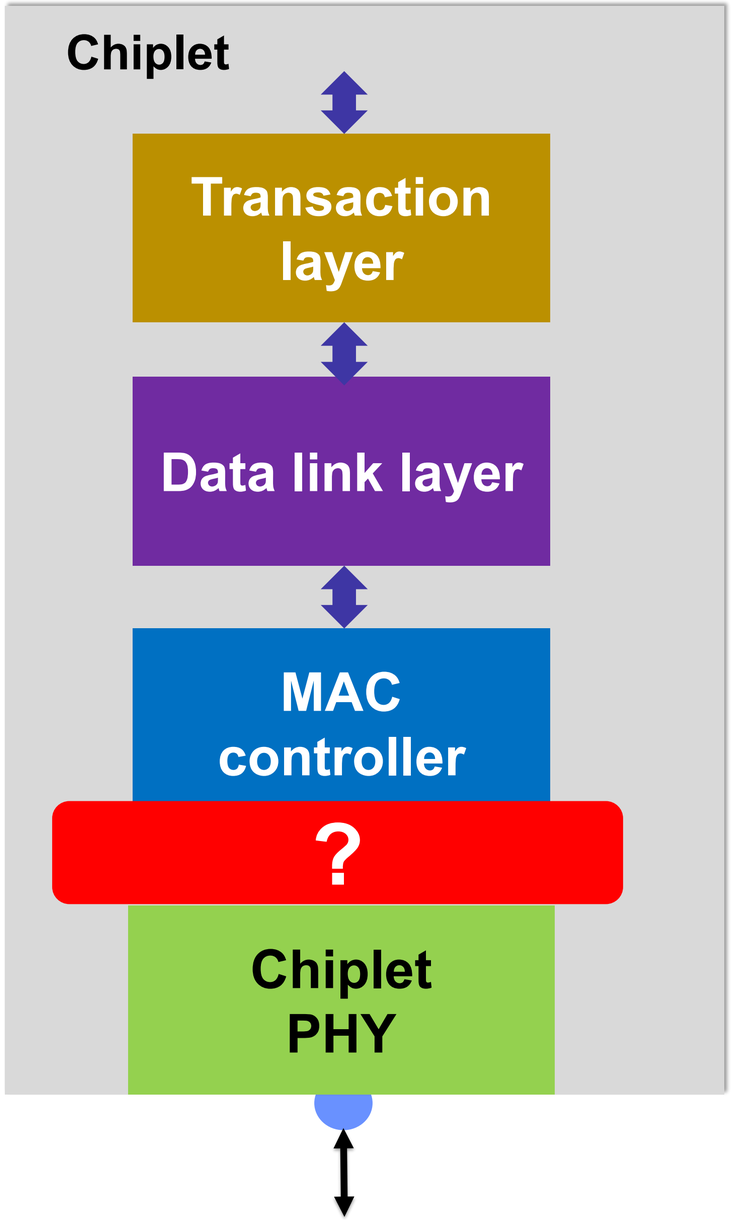


- Higher pin count, parallel bus
- E.g. PHYs: AIB, BoW
- Lower speed bus
- Higher cost packaging

- Lower pin count, serial bus
- E.g. PHYs: AXDieIO, CNRZ-5
- Higher speed bus
- Lower cost packaging

- No universal PHY solution to connect chiplets
- **An open Chiplet ecosystem needs to accommodate different PHYs**

How to connect the PHY to upper layers in a chiplet?

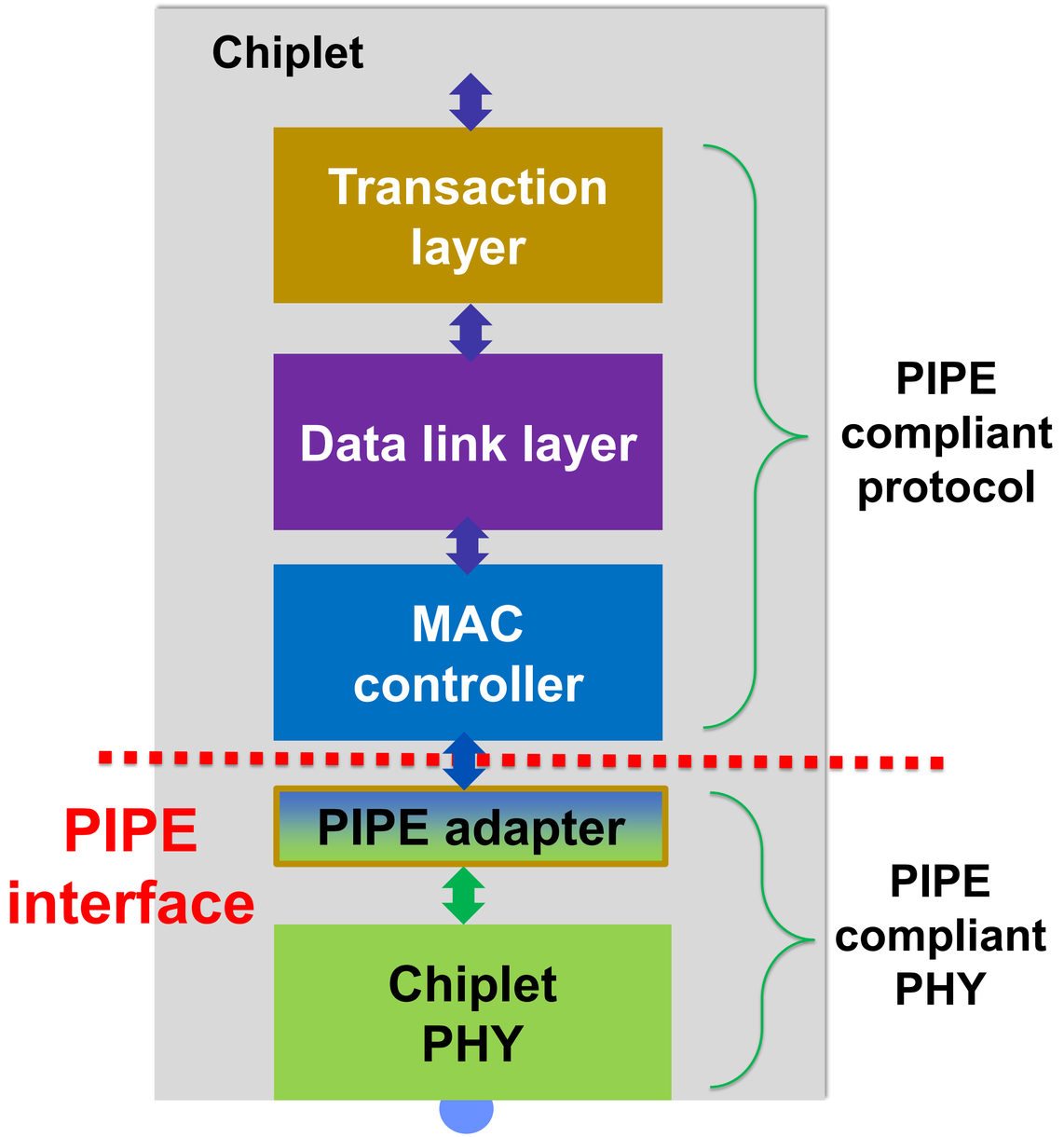


- Given there are many choices of chiplet PHYs how do we efficiently connect the PHY to the upper layers?

1. **Fully custom MAC/PHY interface**
   - Custom design, one-off
   - Cannot mix/match PHYs

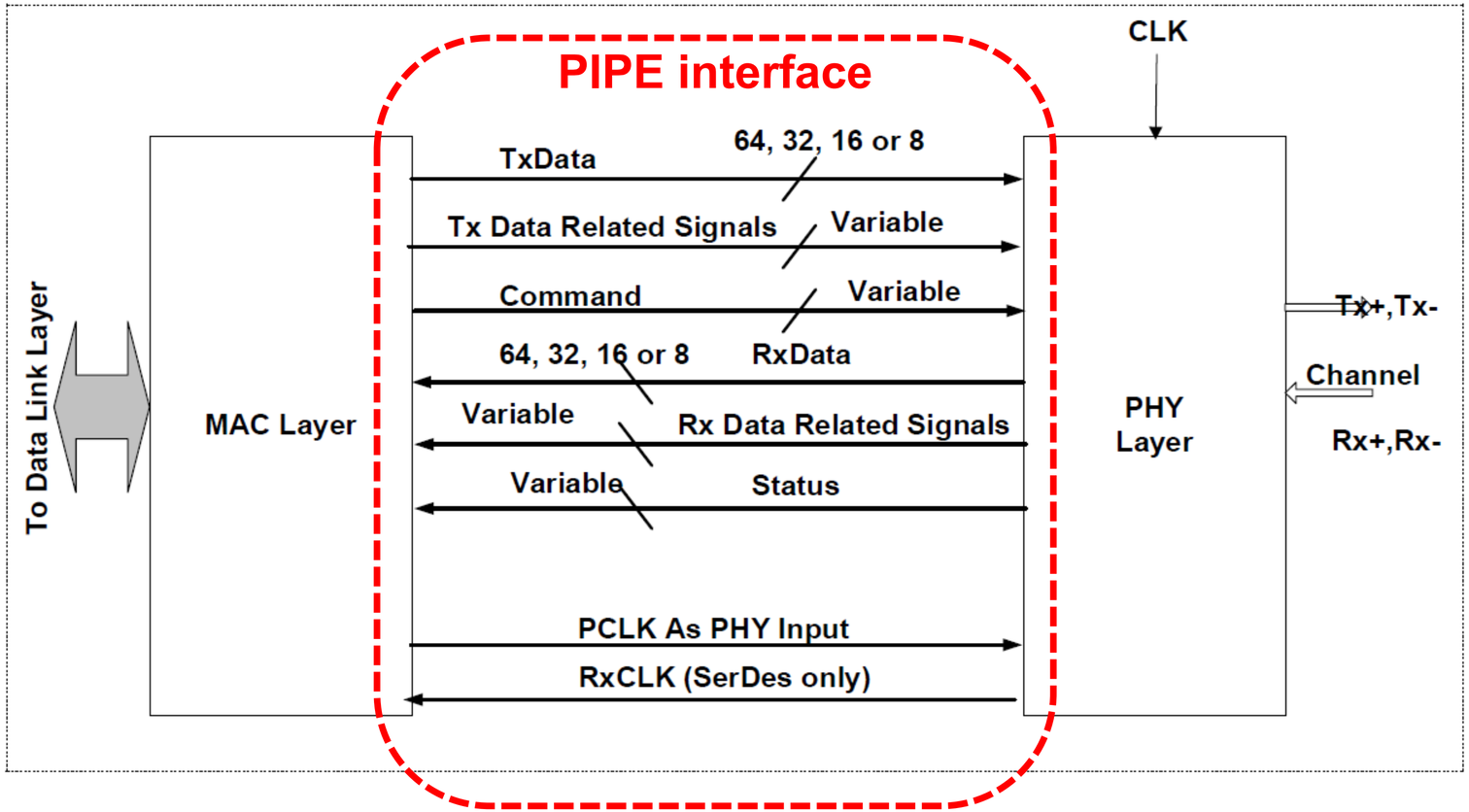
2. **Common MAC/PHY interface**
   - Enables interoperability, ecosystem
   - PIPE a good example of a common MAC/PHY interface already used extensively

Use PIPE to interface PHY to upper layers



- PIPE defines interface between MAC controller and PHY
  - Establishes connectivity and features to be implemented on either side of the interface
- Enables PHY to be designed independently of MAC and vice-versa
- PIPE compliant with **PCIe, SATA, USB, DP, Converged IO, CXL, CCIX,**
- PIPE compliant IP readily available

Adapting PIPE for Chiplets



- Some features of PIPE supported protocols may not be needed for chiplets
- Provide guidance on how to adapt PIPE for Chiplets while optimizing for **lower reach, power and latency**

How to implement PIPE for Chiplets

- PIPE specification categorized into following options:

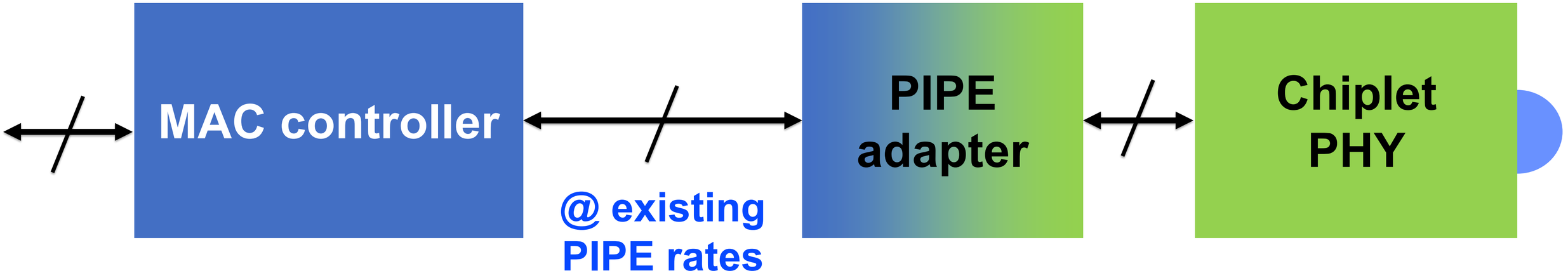
1. Mandatory

- Any requirement that is essential to PIPE functional correctness is mandatory
- **I.e. do not break the controller, don't break the PIPE, don't break the firmware**

2. Optional / Not necessary

- Chiplet PHY can assert and behave as though it met the PIPE requirement even if it does not implement PIPE feature
- Chiplet PHY can spoof/mimic PIPE behavior to preserve upper layers
- To MAC and upper layers it looks like PIPE is satisfied/preserved, thus can use off-the-shelf PIPE compliant upper layer IP

First draft will target PIPE compliant rates



- To take advantage of existing IP first draft supports PIPE compliant rates e.g. PCIe gen 5 (32Gbps) between MAC and PHY
- Future releases to support non-standard rates and configurations

Next steps

- Good progress on converging to a public release document
- Working document accommodates different PHY topologies
- Building first draft release of document by OCP Global Summit
- Looking for further feedback especially from the controller community

Join our weekly meetings
Tuesday mornings 9AM PST!