Accelerating GFD RTL Verification: A Lightweight Host-to-GFD Framework

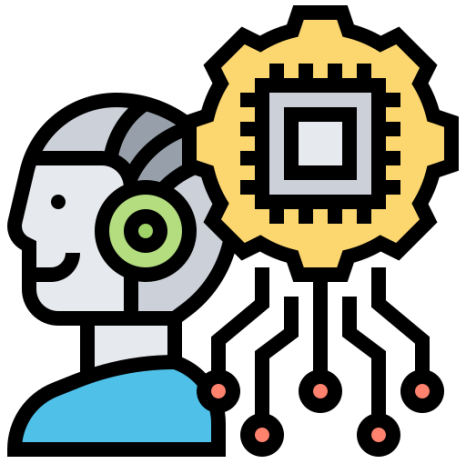
Presented by: Wade Chen, Sr. Verification Engineer & Project Leader of the Avery CXL VIP Team, Siemens

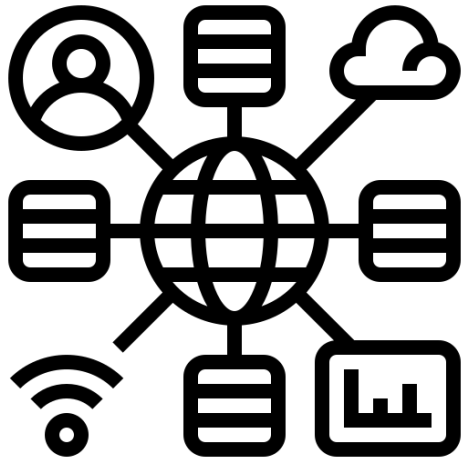
Agenda

- Introduction
  - Background of CXL Fabric and GFD
  - The motivation of GFD verification
  - Technical Overview
- Problem Statement
  - The challenges of verifying RTL GFD
- Proposed Framework
- Conclusion

Background

- Requirements for the high amount of computation and memory needs in many popular fields:
  - Machine Learning/AI.
  - High-performance Computing (HPC).
  - Big Data Analytics.

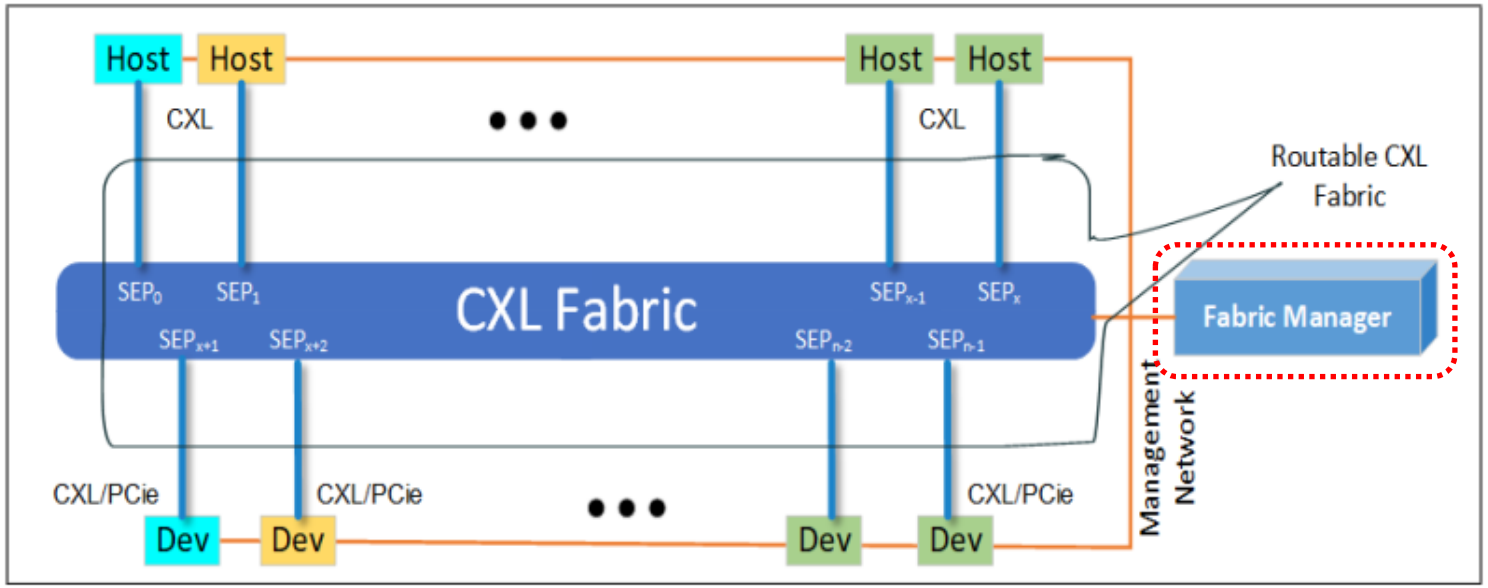




CXL Fabric Architecture

- New features to scale from a node to a rack-level interconnect:
  - Expand the size of CXL fabric using Port Based Routing.
  - Enable support for GFDs and peer communication.
  - A Fabric Manager (FM) connects to the CXL Fabric with an out-of-band management network.

Figure 7-25. High-level CXL Fabric Diagram



Global Fabric-Attached Memory Device (GFD)

- GFD features scalable memory pool, making it highly suitable for Machine Learning/AI and HPC.
- Accessible by all hosts and peer devices within a CXL fabric:
  - Direct CXL P2P access by UIO and CXL.mem.
  - Memory Pooling.

Figure 7-26. ML Accelerator Use Case

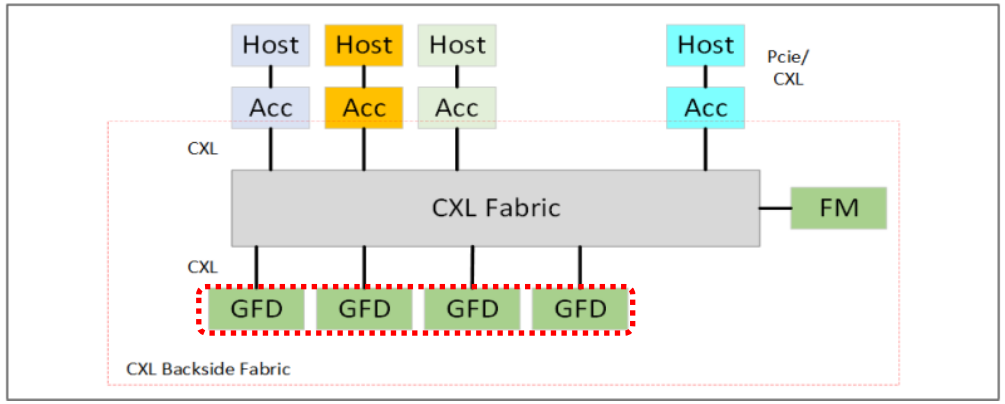
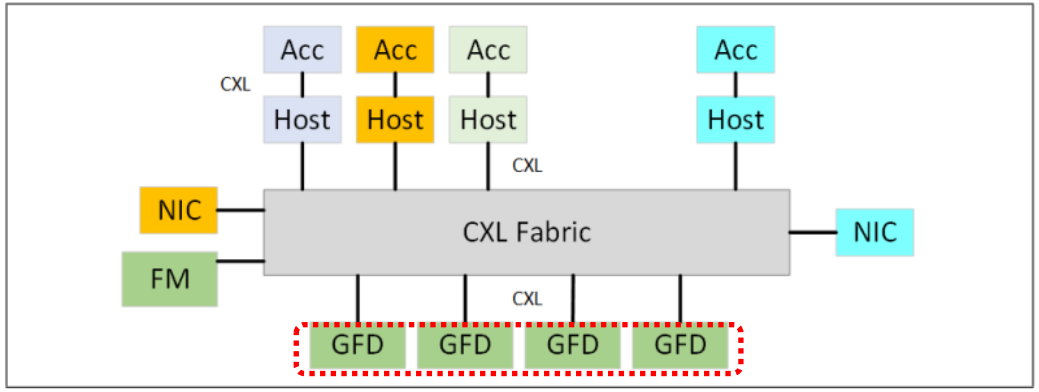


Figure 7-27. HPC/Analytics Use Case



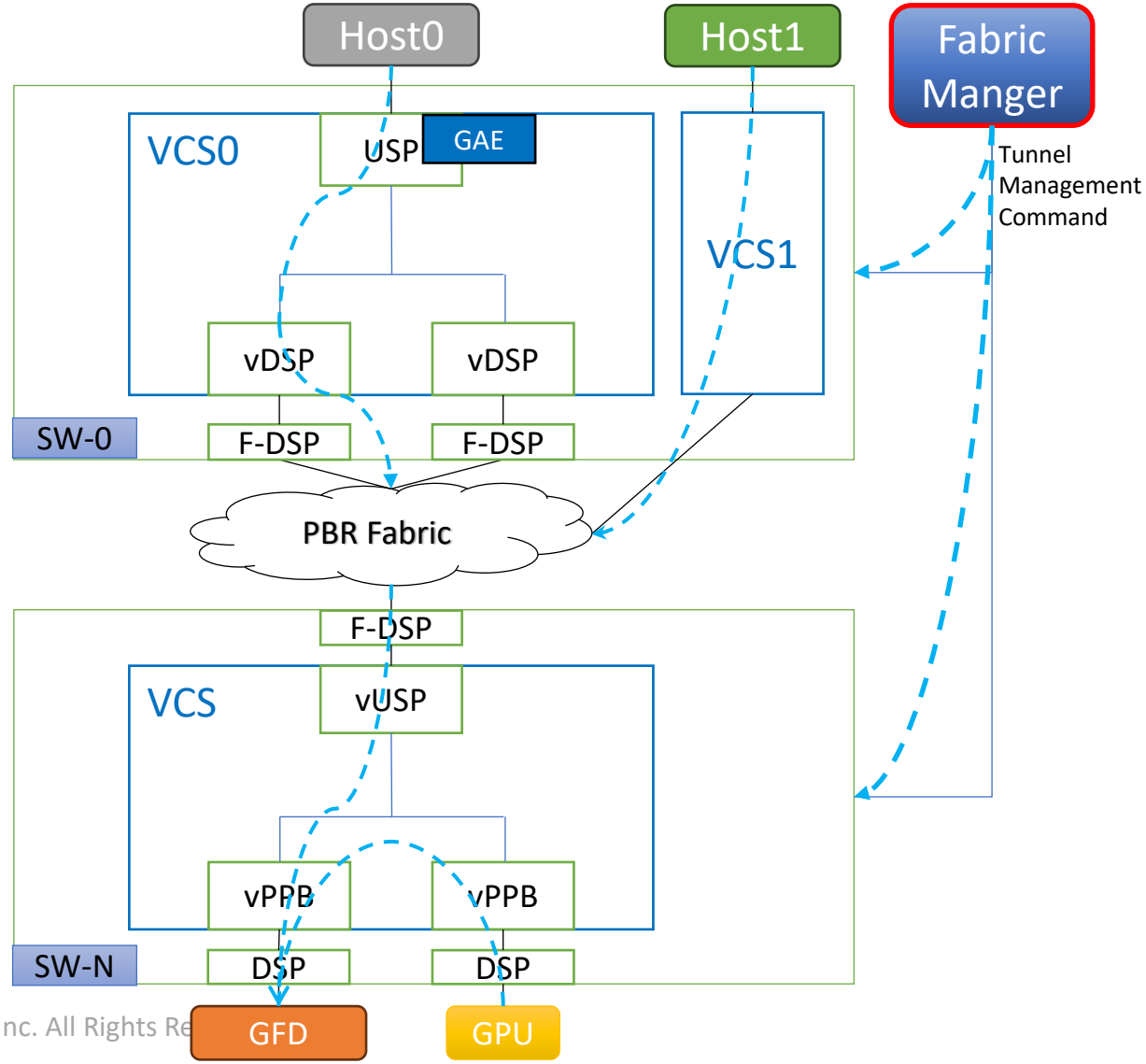
Motivation

- A **Multiple Logical Device (MLD)** also features scalable pool by virtualizing the physical device into Logical Devices (LDs) and a Fabric Manger LD (FMLD).
- A comparison between a GFD and an MLD is presented in the form of a table.
- **The focus of this presentation is to explore an efficient method for verifying GFD.**

Feature or Attribute	MLD (LD-FAM Device)	GFD
Configuration Space	Independent configuration space per LD	Single configuration space
DPA Space	Independent DPA space per LD	Single DPA space
Number of supported hosts	16 max	1000s architecturally; 100s more realistic
Interleave Ways (IW)	1/2/4/8/16 plus 3/6/12	2–256 in powers of 2
Link Connection	Normal CXL Switch or PBR Switch	Limited to PBR Switch

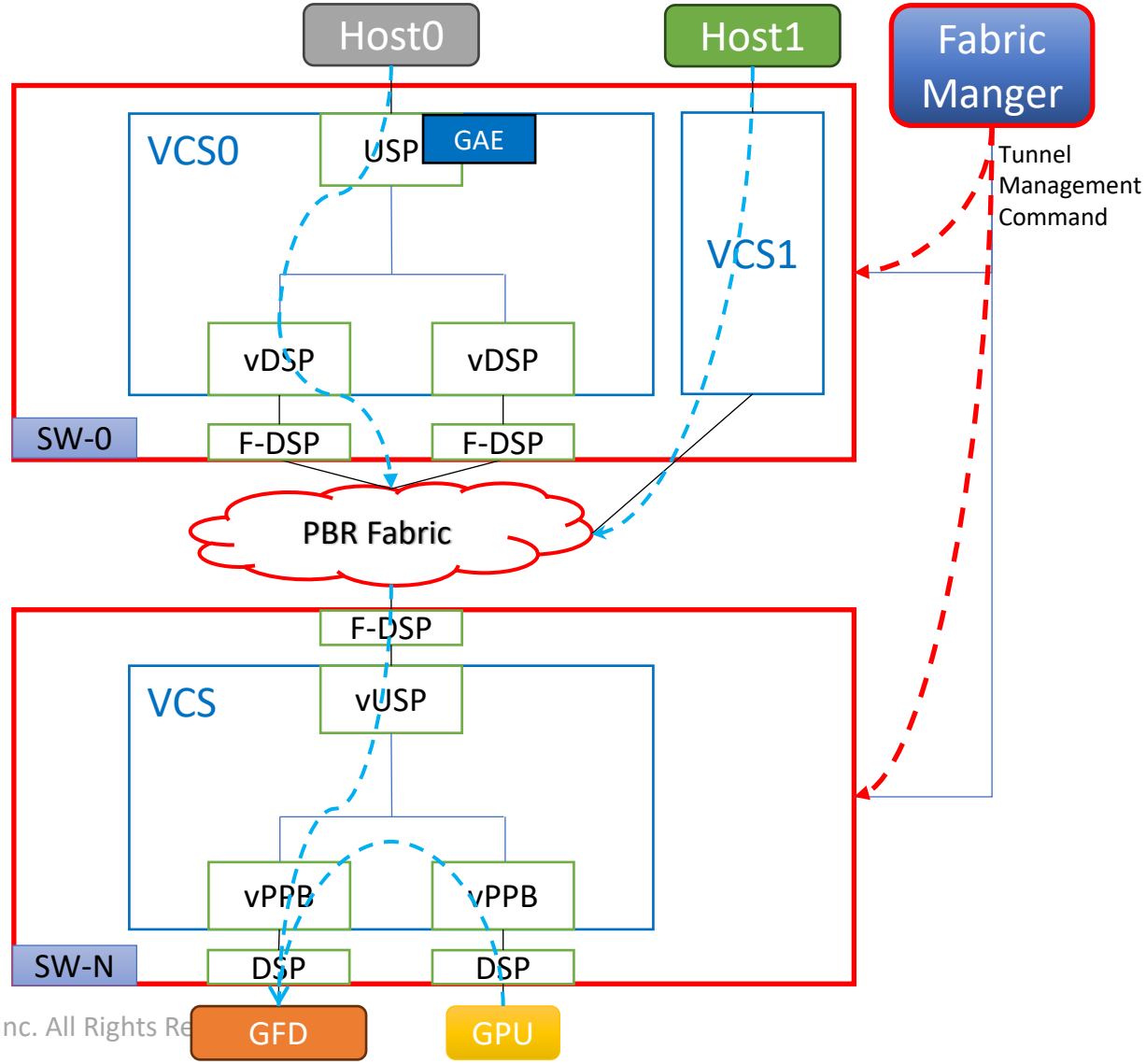
GFD Discovery & Configuration

- The necessary configuration for routing a request to a specific GFD in the CXL:
  - PBR Switch Discovery and Configuration.
  - FM configures the GFD and the Global Memory Access Endpoint (GAE).
  - Host Software performs enumeration and configures the GAE, if it exists.



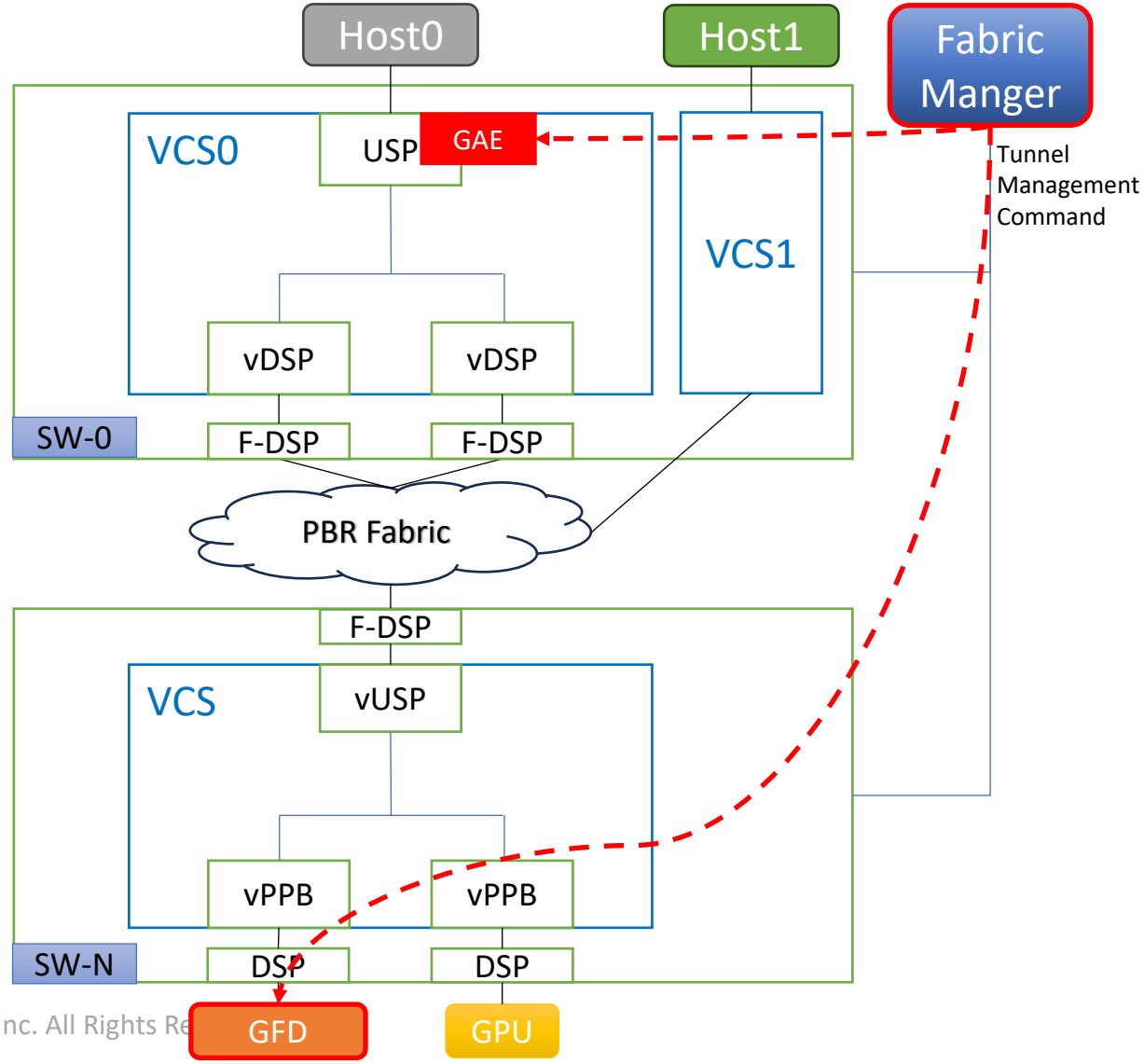
PBR Switch Discovery and Configuration

- CCI Commands:
  - Identify Switch Device.
  - Get Physical Port State.
  - Identify PBR Switch.
  - Bind vPPB and Unbind vPPB.
  - Fabric Crawl Out.
  - Get PBR Link Partner Info.
  - Configure PID Assignment.
  - Configure PID Binding.
  - Set DRT.
  - Set LDST/IDT Configuration.
  - Etc.



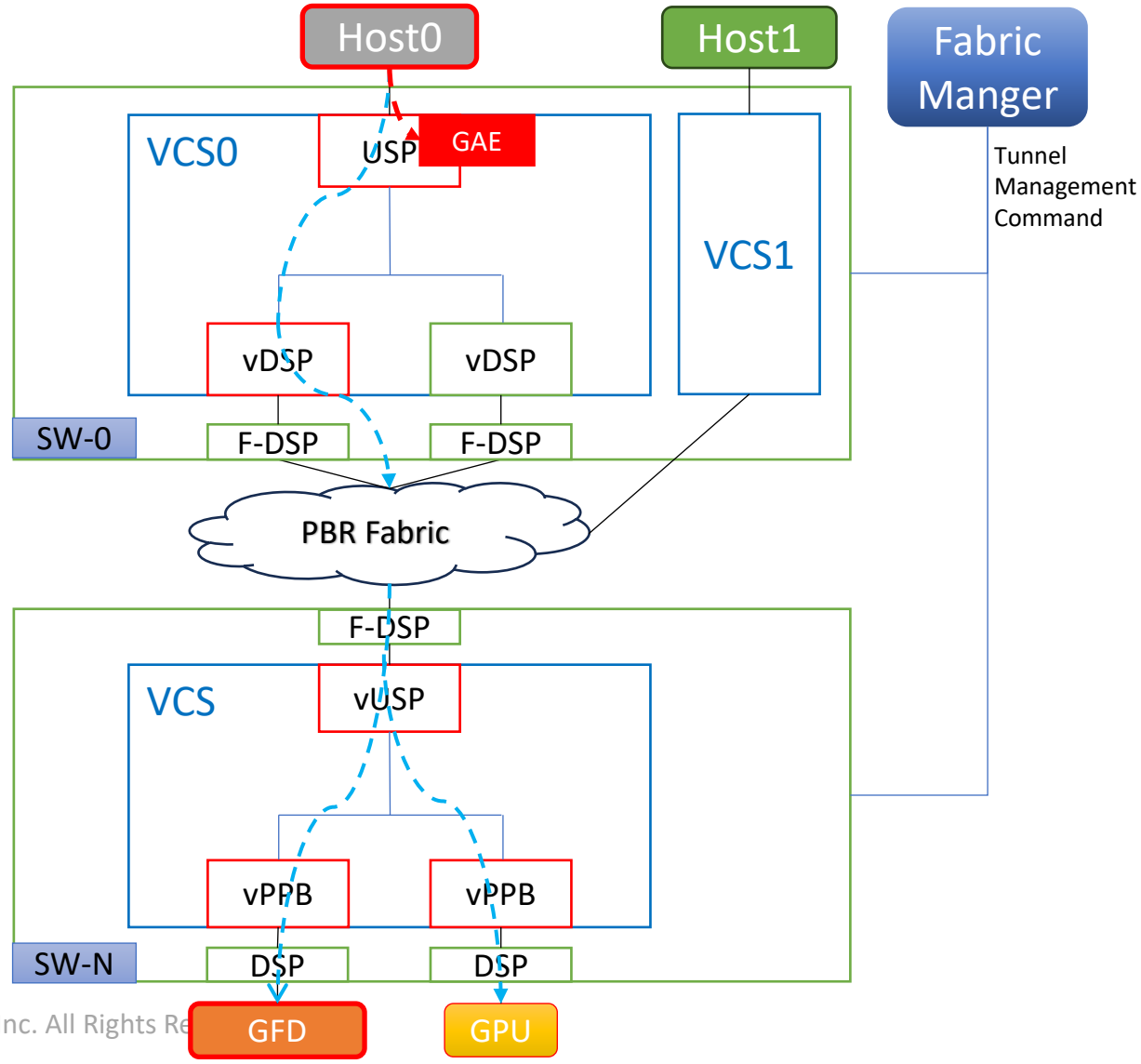
FM Configures GAE and GFD

- CCI Commands:
  - To GAE:
    - Configure VCS PID Access.
    - Set VendPrefixL0 State.
    - Etc.
  - To GFD:
    - Identify GFD.
    - Set GFD DC Region Configuration.
    - Set GFD DMP Configuration.
    - GFD Dynamic Capacity Add.
    - Set GFD SAT Entry.
    - Set GDT Configuration.
    - Etc.



Host Performs Enumeration and Configures GAE

- CCI Commands:
  - Identify GAE command.
  - Set FAST/IDT Configuration.
  - Set FAST Segment Entries.
  - Set IDT DPID Entries.
  - Proxy GFD Management. Command.
  - Etc.

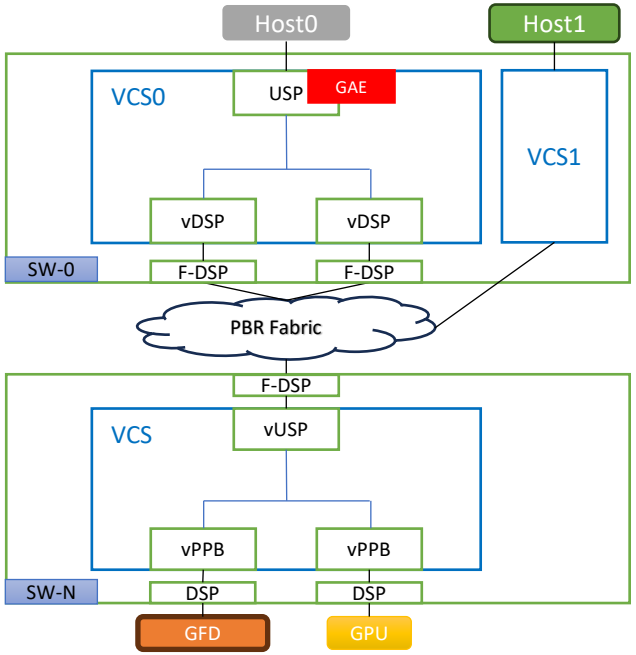


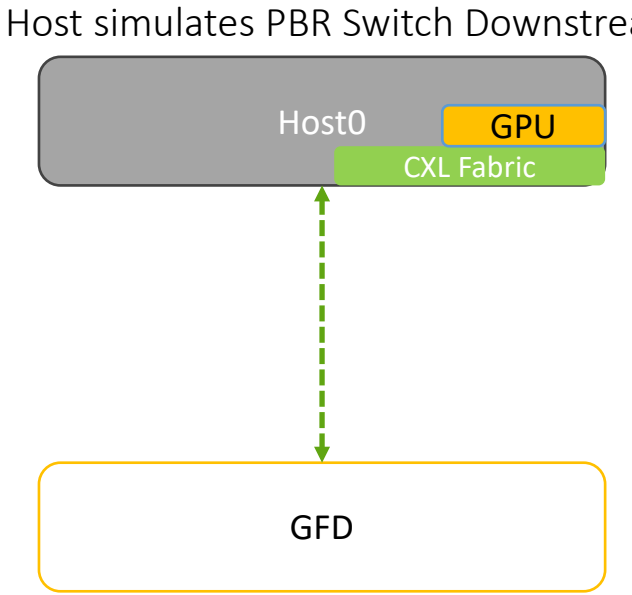
Challenges of Verifying Hardware GFD

- Extremely High Complexity:
  - Numerous CCI Commands.
  - High effort to build a full testbench.
- Long Simulation Time Overhead:
  - All components need to link-up.
  - Large volumes of configuration traffic.
- How to verify essential hardware features in GFD with easier and more efficient testbench?
  - Solution Insight:
    - The SW/FW steps are not difficult to relate to a GFD RTL verification.
    - Reduce time to wait for all components to be ready for configuration.

A Lightweight Host-to-GFD Framework

- Host connects directly to RTL GFD (no CXL fabric switch in between):
  - Host simulates PBR Switch as Switch Downstream Port and GPU.
    - Bypass the configuration and discovery steps of Switch, GAE and GFD, and the routing mechanism inside CXL Fabric.
  - Require GFD design-specific sequences or RTL-customized APIs developed by users for direct configuration of RTL GFD.

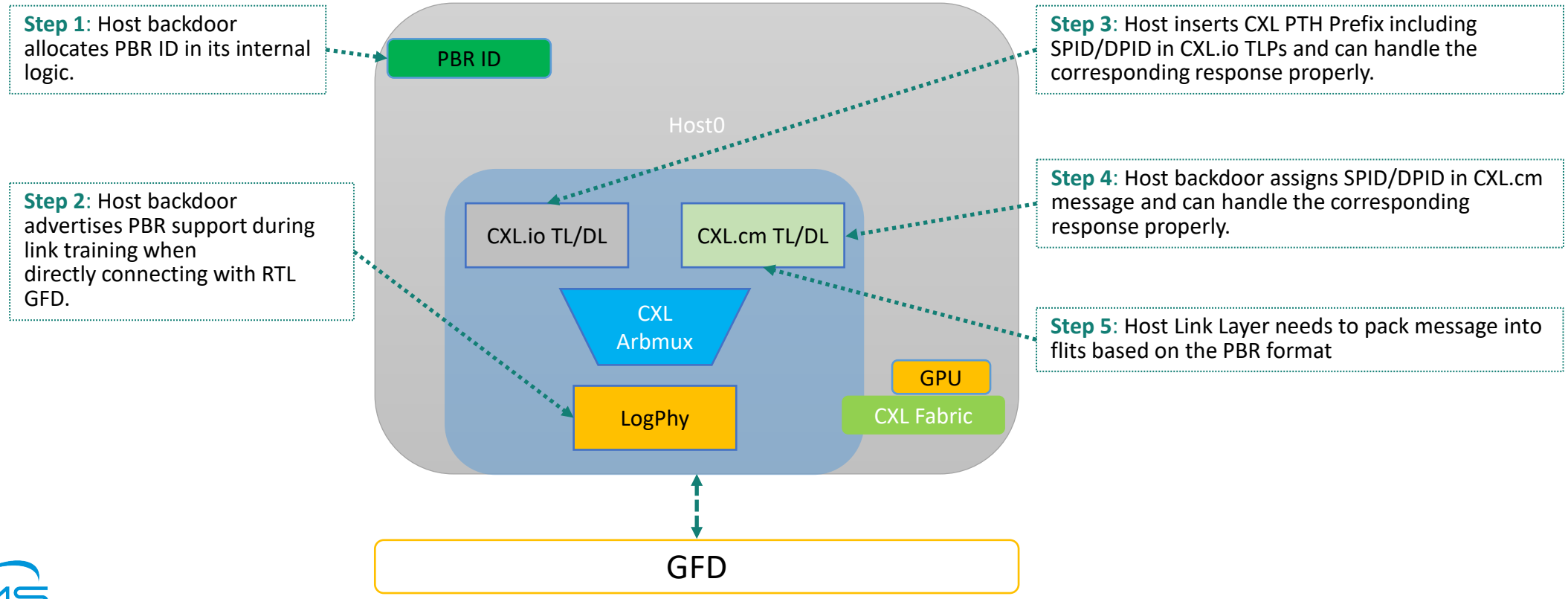




Specific sequence or APIs is required to be developed in DUT for configuration

Host Simulates PBR Switch

- Purpose:
  - Bypass the configuration and discovery steps of Switch, GAE and GFD, and also the routing mechanism inside CXL Fabric.



GFD Design-Specific Sequences or RTL APIs

- Purpose:
  - Finish GFD Configuration without FM CCI Commands from Switch
- Solutions:
  - Require GFD design-specific sequences or RTL-customized APIs developed by users to backdoor configure the RTL GFD.
    - Host may support to send all FM CCI Commands to develop configuration sequences.

CXL Solutions

- Siemens provides a light-weight verification solution based on the **methodology illustrated in this presentation.**
- Host BFM capabilities:
  - Provide backdoor knobs and callbacks to enable direct communication with RTL GFD.
  - Support all CCI Commands generation.
- Testsuite Reuse:
  - Leverage an off-the-shelf testsuite originally designed for a Type3 Device.
  - Add GFD-specific tests for extended coverage.

Conclusion

- **Introduction**: the basic concept of CXL Fabric and GFD.
- **Challenges**: Verifying RTL GFD with a full CXL Fabric setup is extremely complex:
  - Require to include full bring-up of CXL PBR Switch, FM, GAE, and GFD.
  - Require numerous software/firmware configuration steps.
  - Such a setup leads to long simulation time, high effort, and testbench complexity.
- **Siemens Solution**: A light-weight Host-to-GFD framework to bypass full fabric setup.
  - Reduce configuration overhead and simulation time.
  - Enable baseline verification of essential RTL GFD features.
  - Leverage an off-the-shelf testsuite and add GFD-specific tests.

Thank you!