Disrupting the Infrastructure Paradigm

Tim Michels

Distinguished Engineer

Agenda Today

- WHAT IS A DPU?
- HOW CAN THE DPU DISRUPT PRIVATE CLOUD INFRASTRUCTURE?
- WHAT MIGHT A DPU ENABLED DATACENTER LOOK LIKE?
- A COMMUNITY IS NEEDED TO MAKE THIS HAPPEN

DPU Terms and Semantics for Today's Presentation

Some companies and analysts say DPU, others say IPU, or even xPU. This presentation is using the DPU term generically and inclusively as defined in the presentation.

The presentation treats the DPU as a complete system. While it may be anchored by a single powerful SoC, the overall DPU system as discussed is delivered and consumed as an add-in card.

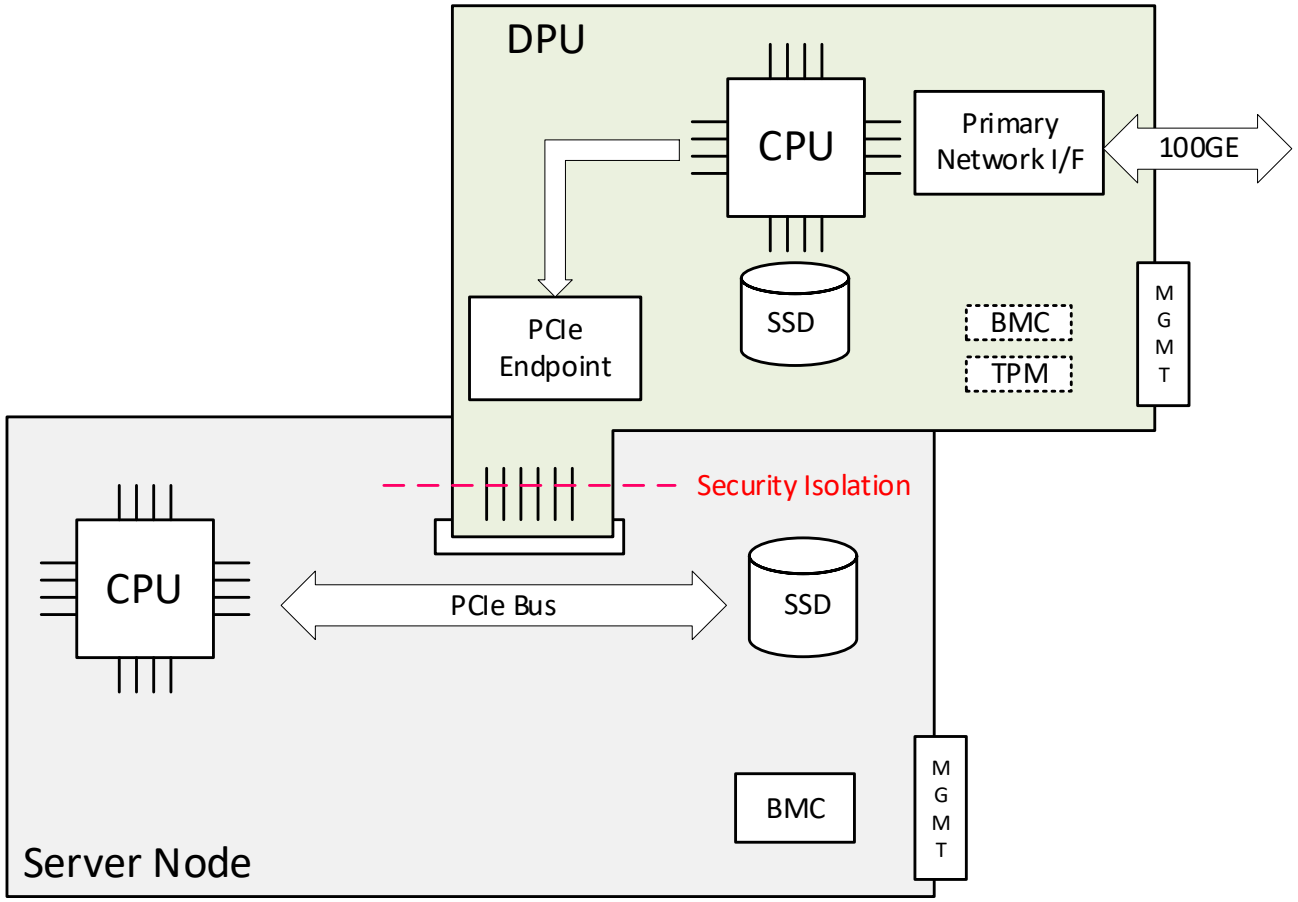
This excludes use cases where a DPU "device" is deeply embedded in a bespoke HW appliance.

Note that this presentation is an **F5 perspective!**

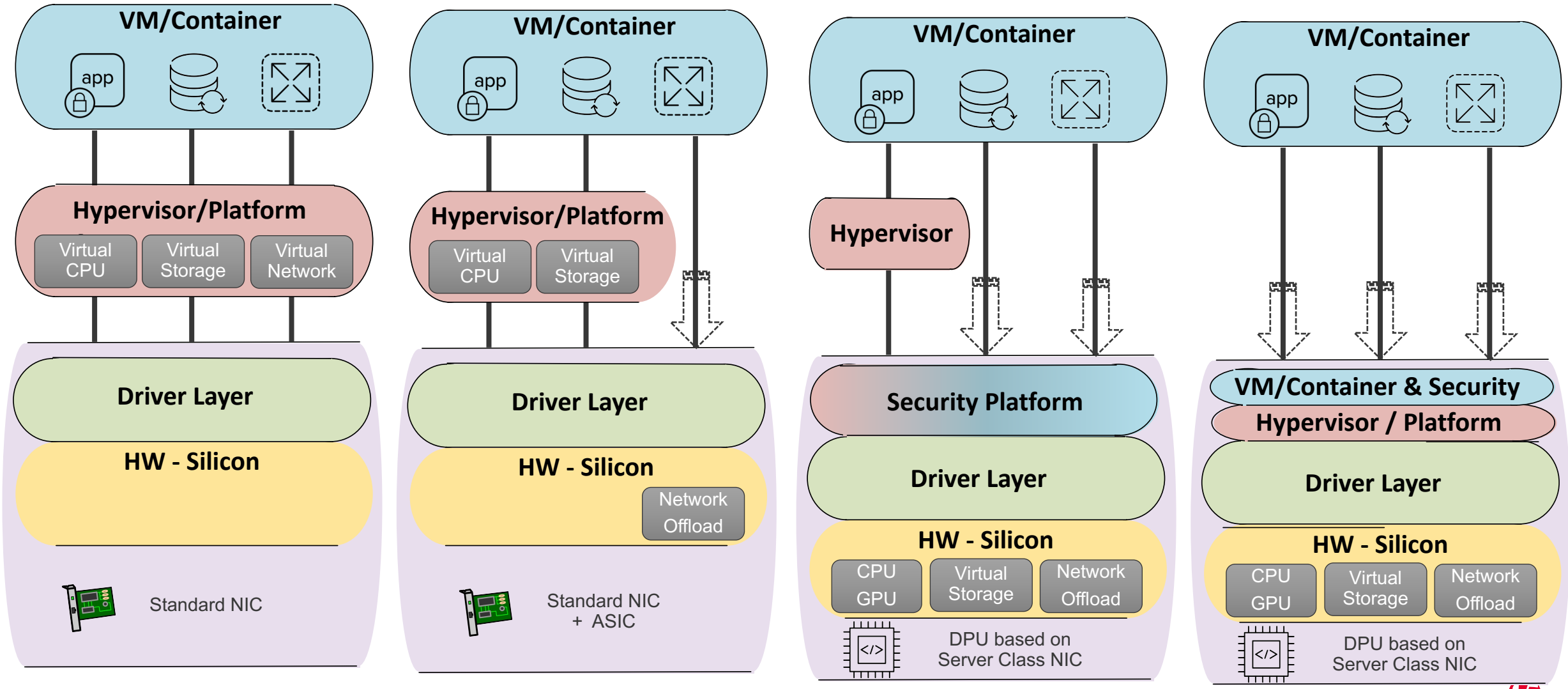
The DPU as a “Server” plugged into the Server

DPU DEFINING TRAITS

- General purpose CPU with significant compute capabilities
- Boots to a general-purpose OS like Linux
- Mix of domain specific HW accelerators
- Strict security isolation from hosting system
- High performance network interface
- Unique Identity on primary network interface
- Independent out-of-band management capability
- Capable of hosting complete infrastructure services



Evolution of the Infrastructure Stack leads to DPUs



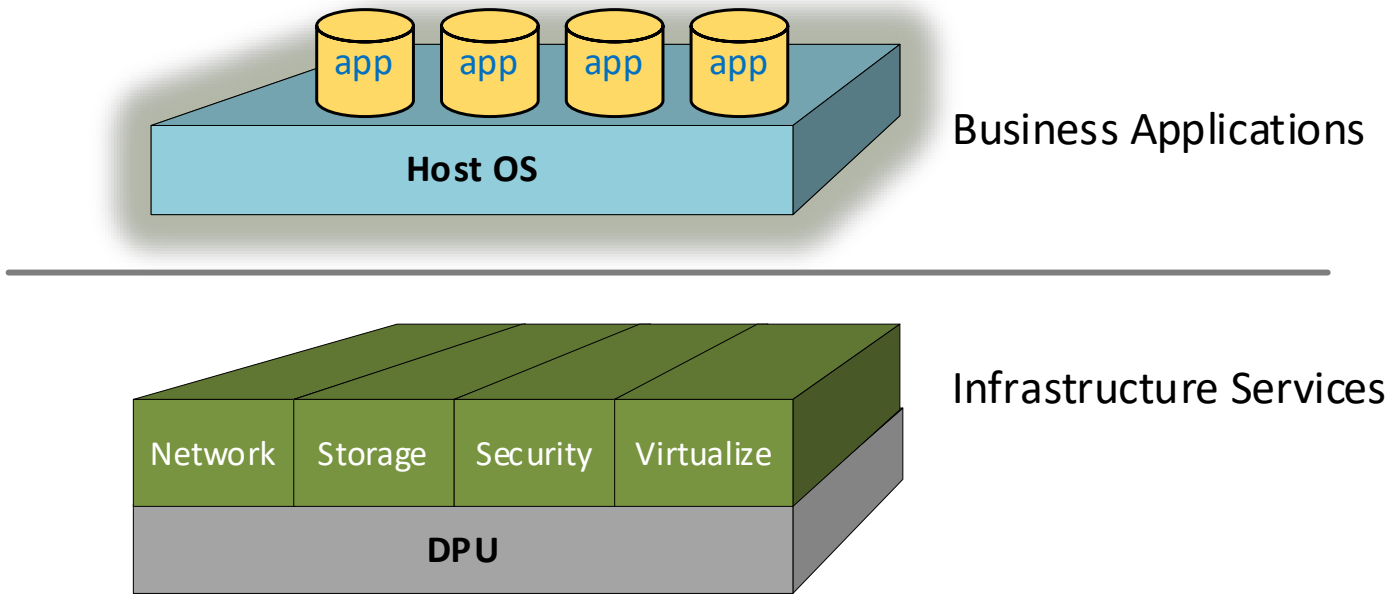
Separating Business Apps from Infrastructure

- **Business Apps** run on the Node
- **Infrastructure Apps** are Services running on the DPU
  - ❑ Network
  - ❑ Storage
  - ❑ Security
  - ❑ Virtualization
- Why move Infrastructure off the node?

  - "30% of CPU cores are being used for datacenter infrastructure needs"

  -"It would take 125 cores to run all the Security, Network, and Storage offloads at 125Gbps"

  Jensen Huang, NVIDIA CEO, @ 2020 GTC Keynote



DPU Disruption Depends on Use Model

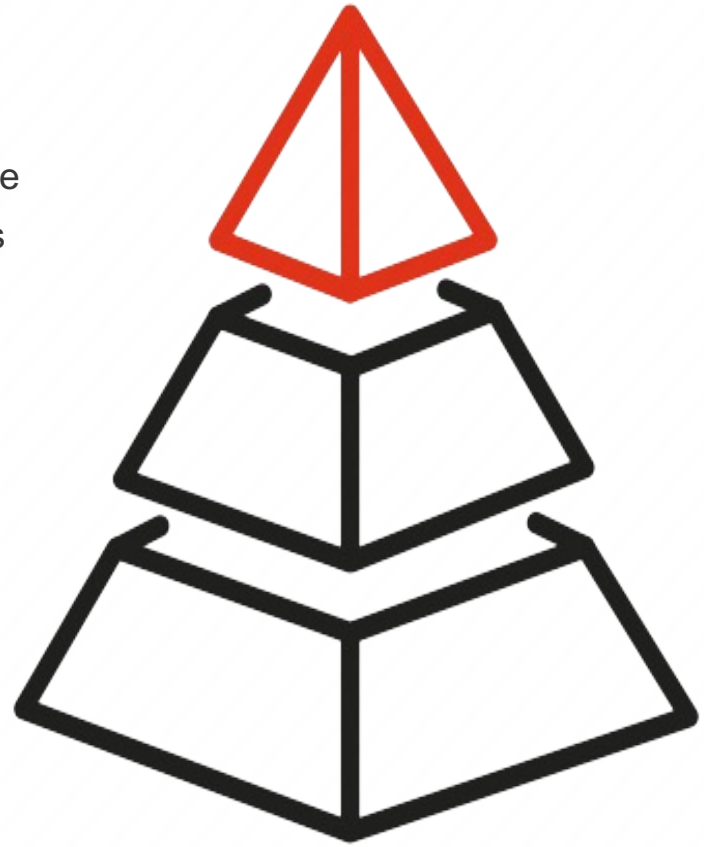
A DPU isn't disruptive when……

- It provides only in-band HW offloads tightly coupled to the server hosted application(s) – SmartNIC model
- DPU compute is only used for HW offload exception processing or to run the HW offload control plane
- It provides only a closed suite of host accessed services defined by the DPU vendor with no ability to deploy 3rd party ISV services

A DPU is disruptive when……

- it is a *hosting platform* for stand alone infrastructure services. These can be deployed in-band per app, between apps (mesh), or multi-app.
- it hosts infrastructure services that provide network, storage, and security abstractions to hide advanced topologies from the applications – SDN Overlays, Mesh, NVMeoF, Network HSM – "Infrastructure Abstraction"
- it is a *hosting platform* for multi-application infrastructure services – L4 FW, DDOS, Bot Detection, API Gateway, etc..
- It is a *hosting platform* for arbitrary 3rd party ISV and customer provided infrastructure services

DPUs are the NEXT evolution of Infrastructure Services



First Generation – **Physical Appliances**

- Discrete boxes, often per service type, deployed across the network hierarchy
- Required independent rack space, network interconnects, and vendor specific knowledge to manage
- Led to inflexible deployments with vendor lock in, forklift upgrades, and high cap-ex and op-ex costs

Second Generation – **Virtualized Appliances**

- Appliance functionality deployed as VM's or container pods
- Deployed stand alone on COTS servers or co-resident as the same node as the application
- Solved some problems of physical appliances but introduced new ones……
- Led to management and control conflicts between application teams and infrastructure teams
- Noisy neighbors and low security isolation barriers

Third Generation – **DPU Hosted Services**

- Does not consume rack space or network interconnects – shared with the host server
- Does not compete for cores, memory, or storage resources with the application workloads – runs on the DPU HW
- Combines the best aspects of the 1st and 2nd generations, while avoiding the pitfalls
- A new, preferred, landing zone for infrastructure services

DPU Enabled Data Center of the Future

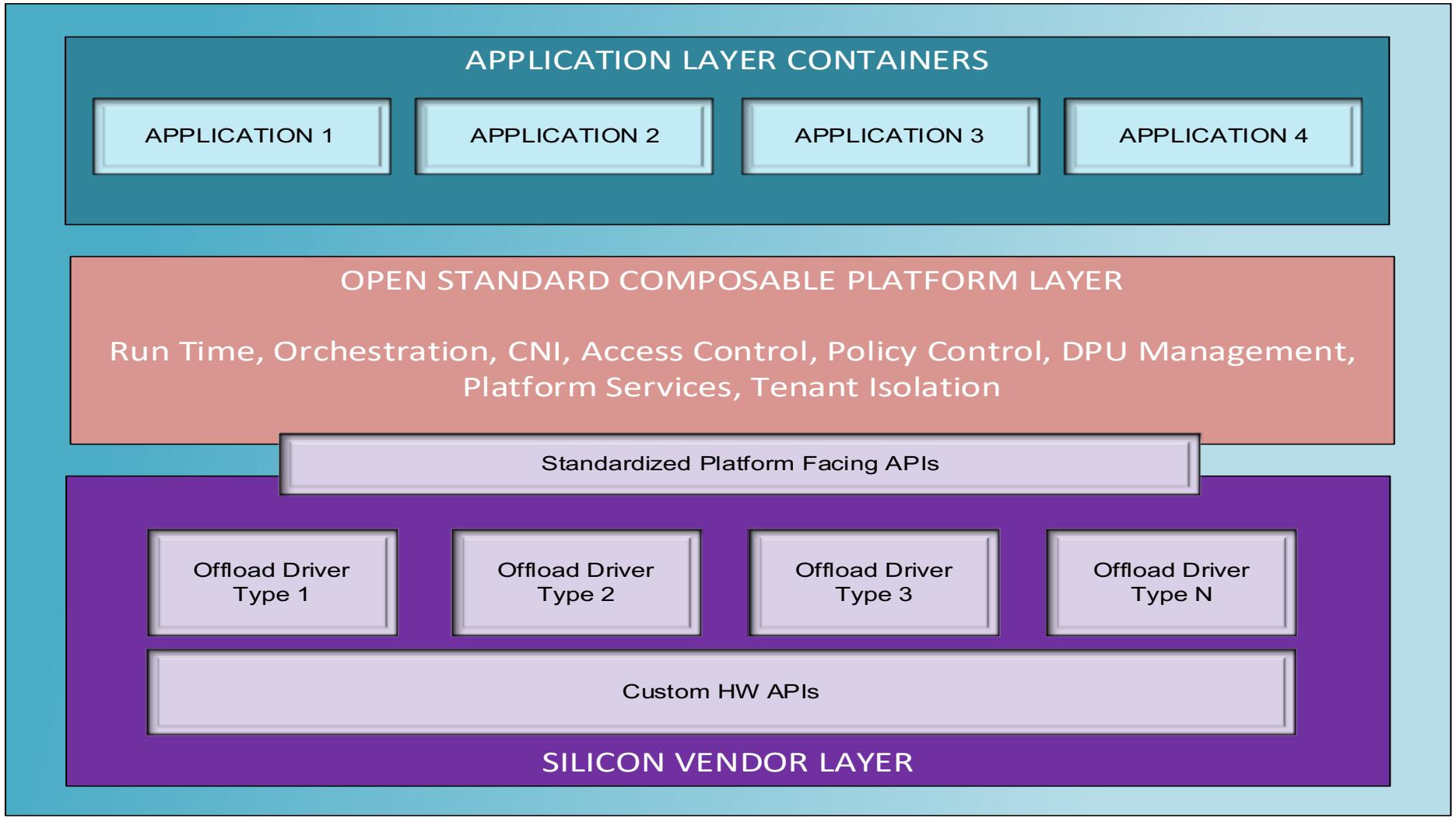
RAINBOWS, BUTTERFLIES, AND UNICORNS

- On-board and provision a heterogeneous DPU fleet at scale
  - Multiple vendors and HW generations with common provisioning methods
  - Image to a common DPU SW stack
  - Provisioned DPUs are made discoverable by Orchestration systems

- Orchestrate and Deploy Infrastructure Services - Dynamically
  - A rich set of multi-vendor ISV provided, and end user created services
  - Any service can run on any DPU
  - Service deployments can be tied into application automation tool chains
  - Service deployments can be automated for scale out and scale in
  - Each DPU may host one or many separate services

- Common Telemetry and Visibility
  - See DPU availability, health, and utilization with common reporting mechanisms and APIs

A SW Stack with an Open Platform Layer is Needed!



Two paths are in front of us ……

WALLED GARDEN OR OPEN COMMUNITY-BASED ECOSYSTEM

Walled Garden

- Faster to get to market and easier to support
- Overall tighter integration
- Single Vendor lock-in
- Limited and selective innovation over time
- Balkanized marketplace
- Limits overall transformative power of DPU based infrastructure

Open Community

- Slower to evolve and reach critical mass in the industry
- Harder to support – needs system integrators
- Requires sustained contributions from a full spectrum of companies and risk-taking early adopters
- Inclusive of the efforts across many Open Source projects
- A rich eco-system of Silicon, PaaS, and ISV vendors will drive innovation and flexibility
- Use cases driven by members not the market TAM and ROI of top vendors

Diamond Bluff Community Goals

- Create a **community-driven** standards-based open ecosystem for DPU/IPU-like technologies
  - Vendor independent – Silicon, System, PaaS, ISV, and End User collaboration
- Create a **vendor agnostic** framework and architecture for DPU/IPU-based software stacks
  - System Integration framework
  - Lifecycle management
  - Telemetry and visibility
- Re-use or define a set of **common APIs**
- Provide implementation examples to **validate** the architecture and API's

A CALL TO ACTION

F5 asks for your help in building this community!

Thank You