

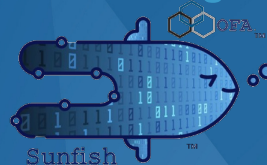


2024 OFA Virtual Workshop

MANAGING COMPOSABLE DISAGGREGATED INFRASTRUCTURE WITH OFA SUNFISH

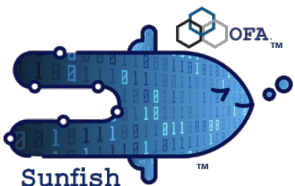
Christian Pinto

Staff Research Scientist, IBM Research Europe
Co-chair, OpenFabrics Alliance Management Framework Workgroup



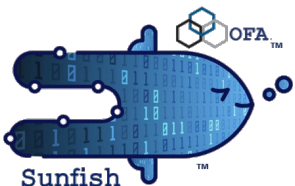
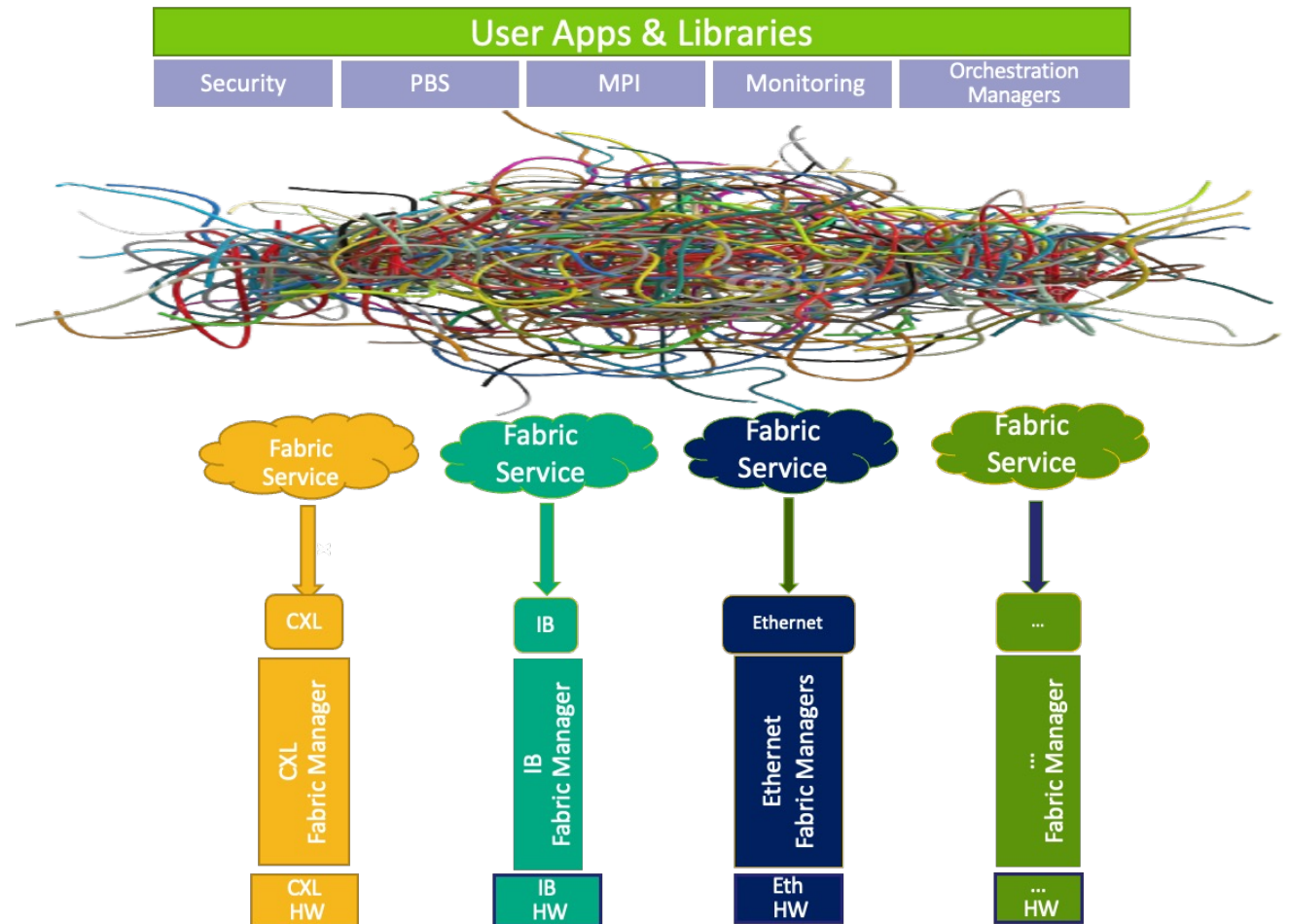
CONTRIBUTORS

- **Michele Gazzetti (IBM Research Europe)**
- **Phil Cayton (Intel)**
- **Russ Herrell (Hewlett Packard Enterprise)**
- **Michael Aguilar (Sandia National Labs)**
- **Brian Pan (H3 Platform)**
- **Ziyan Zhuang (H3 Platform)**
- **Jin Hase (Fsas Technologies Inc.)**
- **Naoki Oguchi (Fsas Technologies Inc.)**

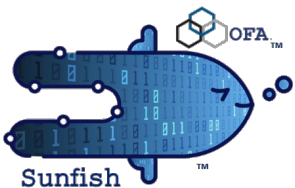
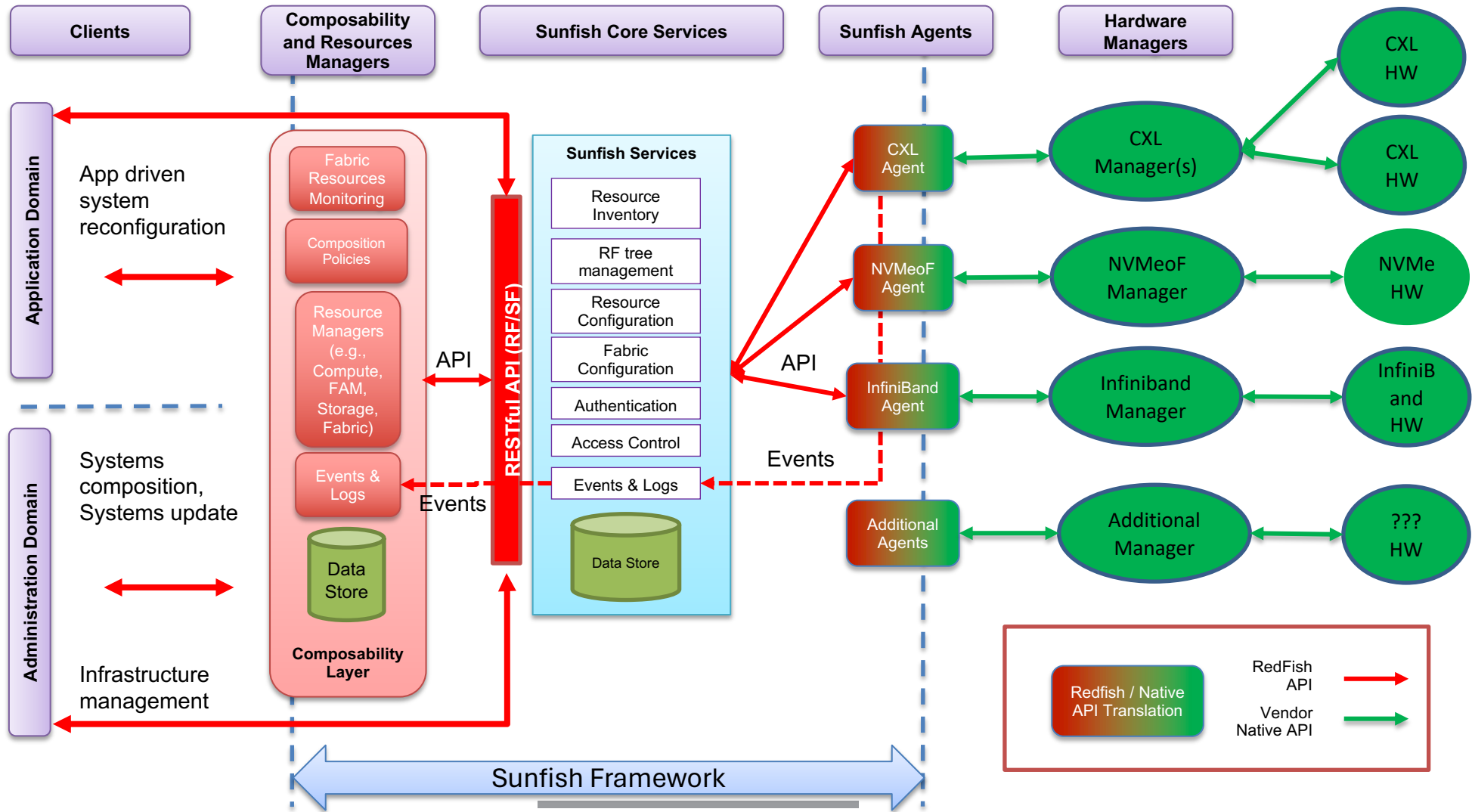


OFA SUNFISH

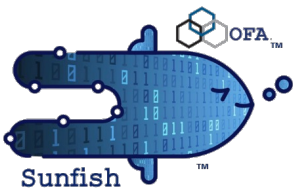
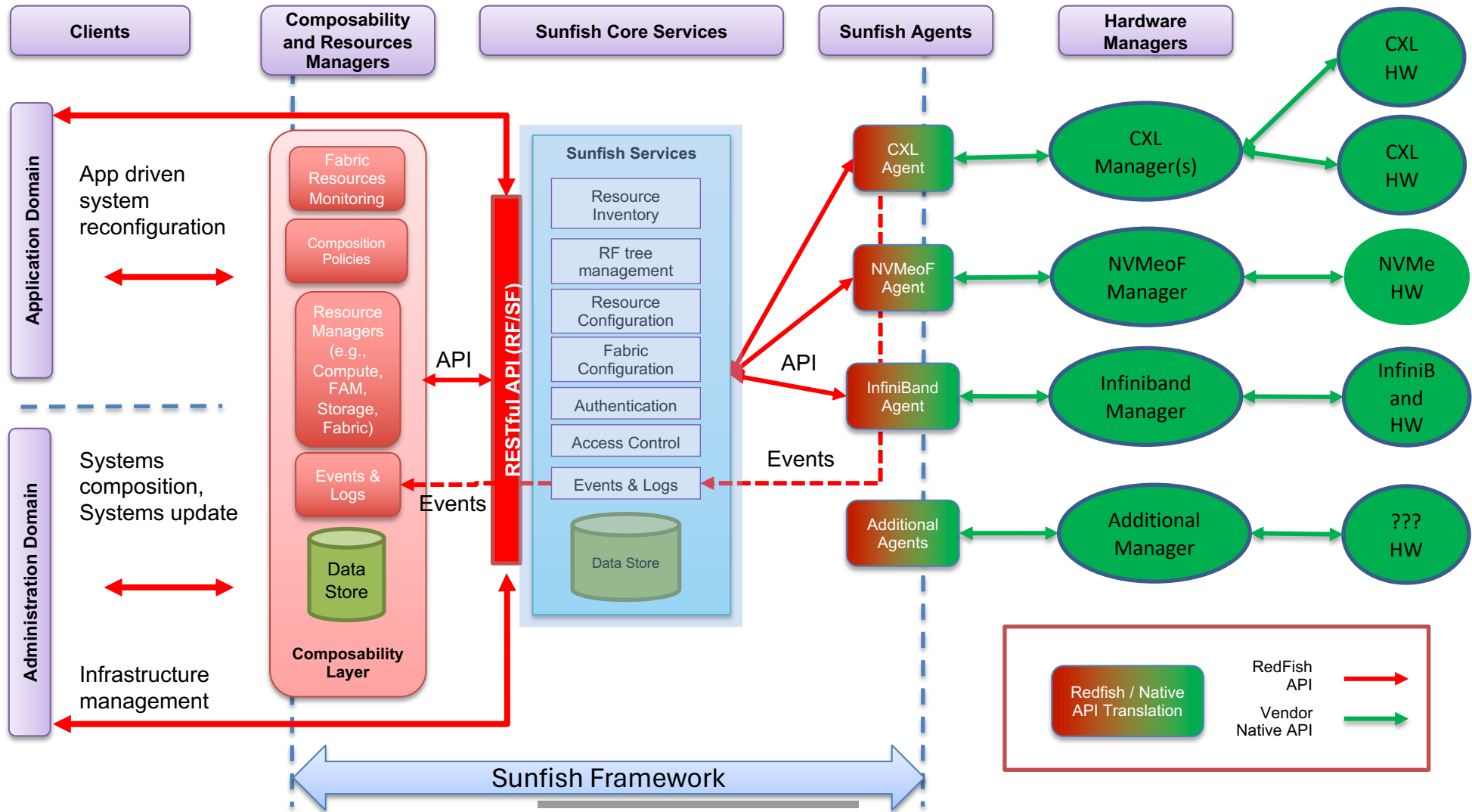
- **Network(fabric)-disaggregated infrastructure becoming the state-of-the-art**
- **No common fabric manager interface or fabric model available to link applications with remote resources**
- **Administrators asked to manage an increasing heterogenous fabrics infrastructure**
- **Difficult to automate because different fabrics require different optimizations**



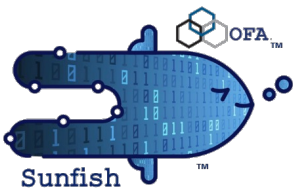
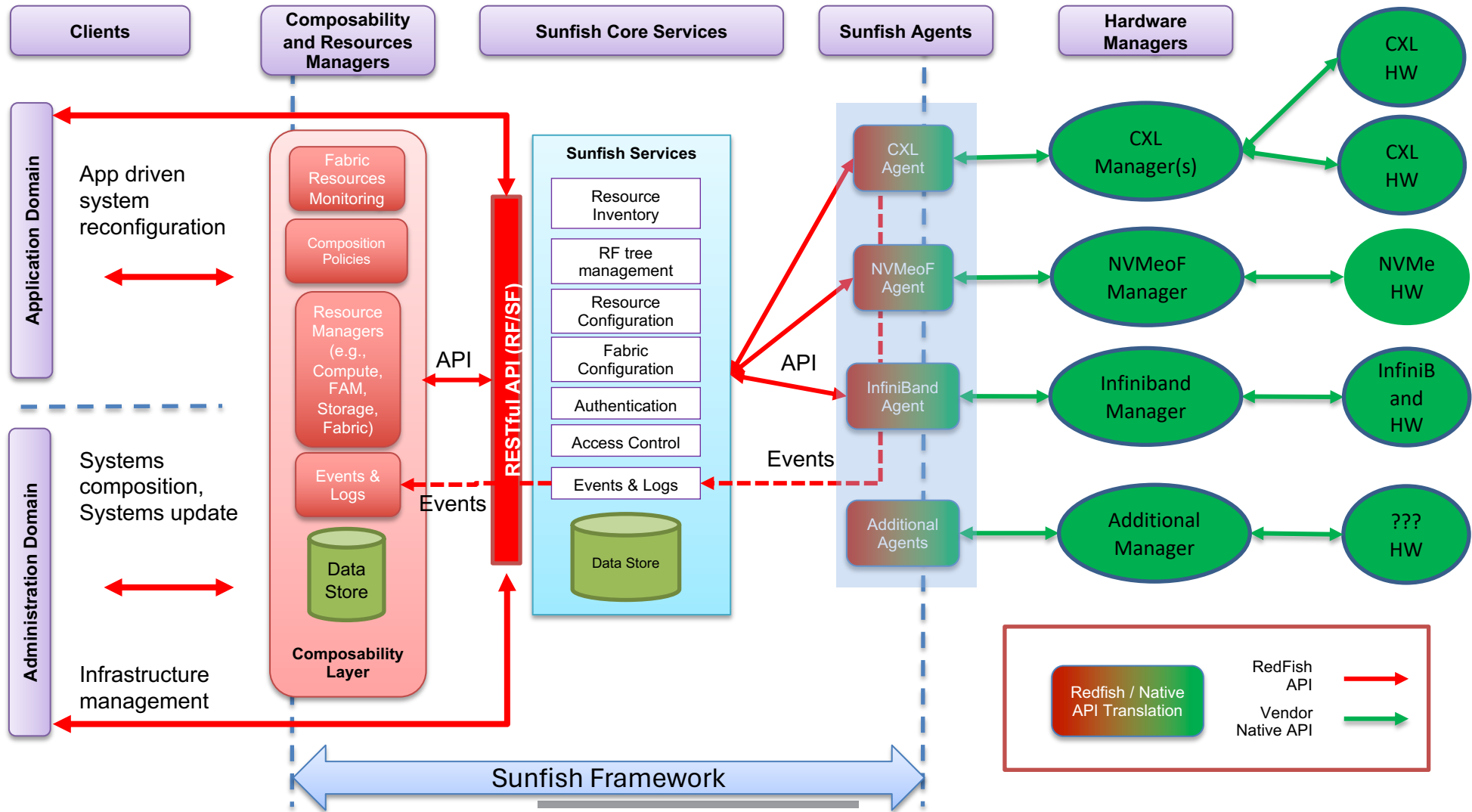
OUR PROMISE



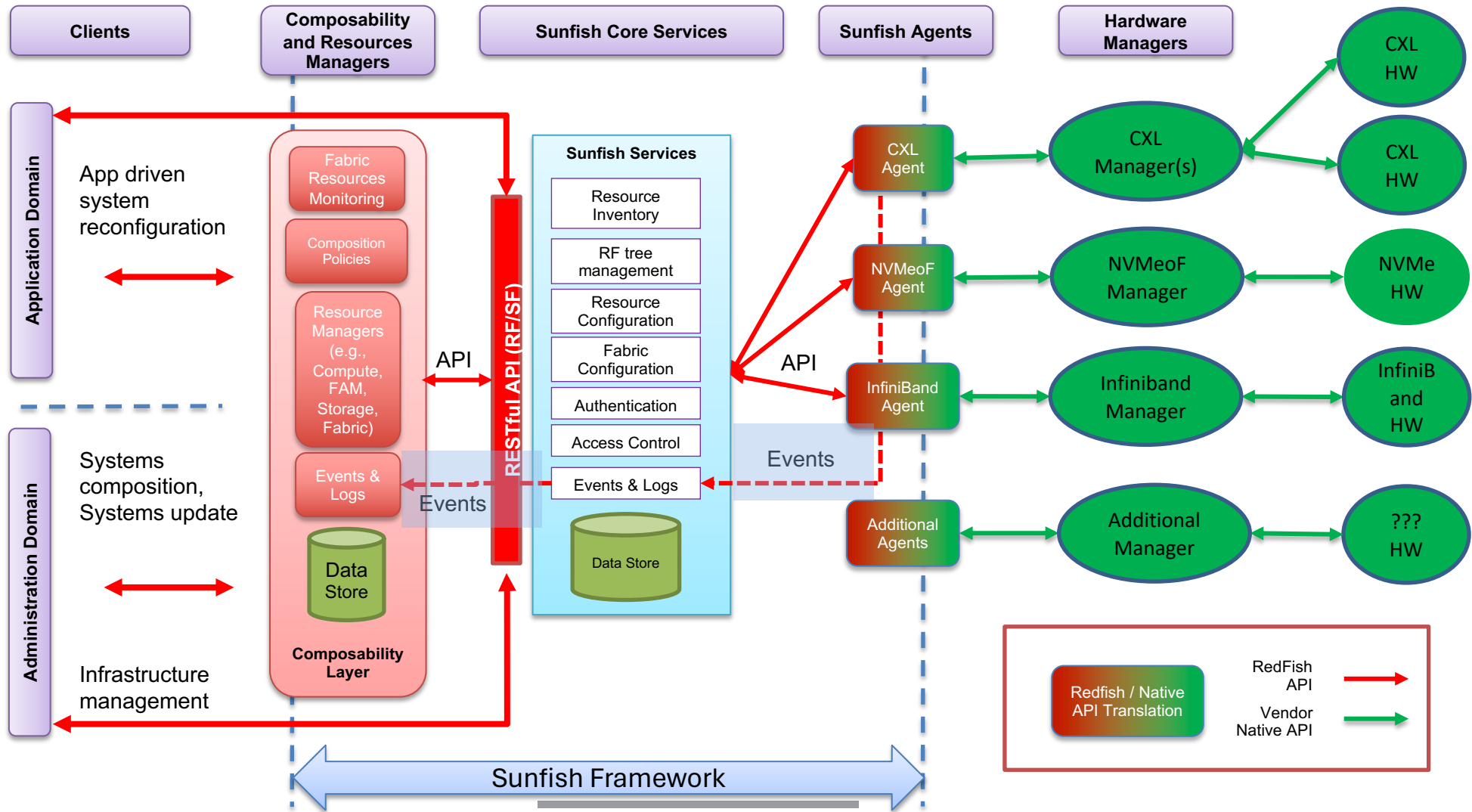
OUR FOCUS SO FAR



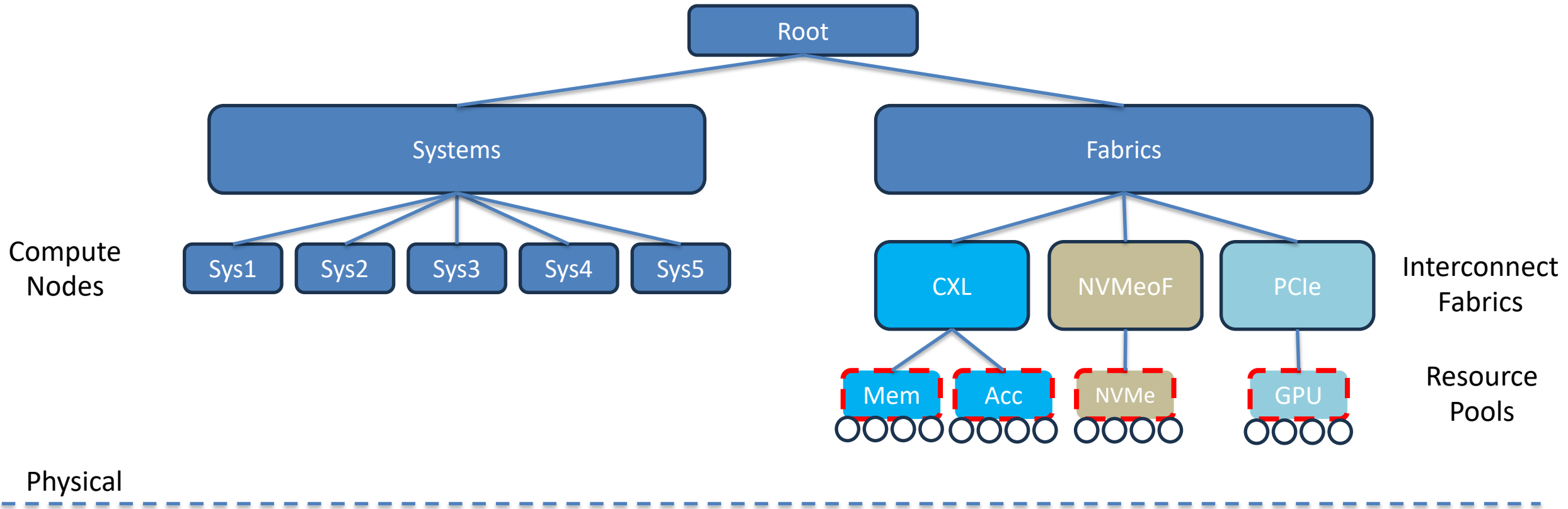
OUR FOCUS SO FAR



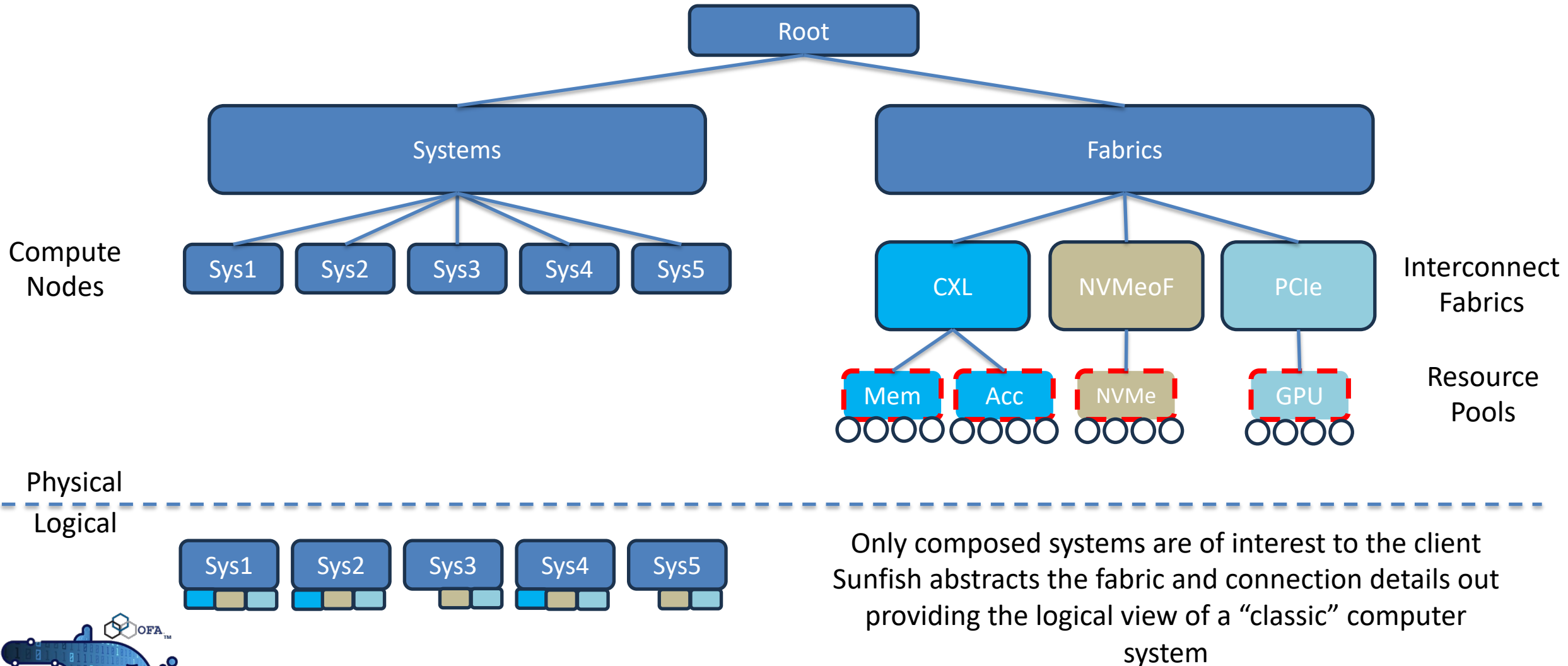
OUR FOCUS SO FAR



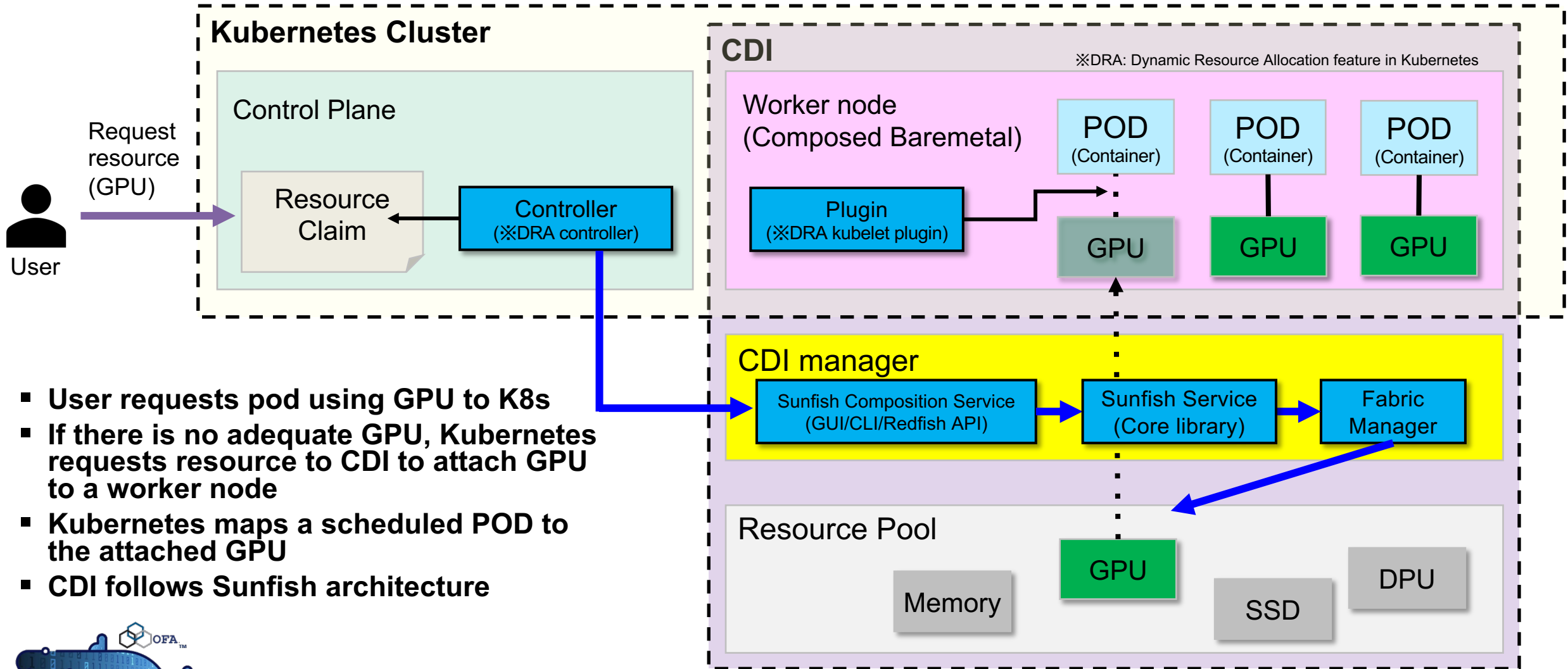
WHY SUNFISH?



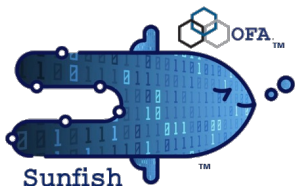
WHY SUNFISH?



KUBERNETES COMBINED WITH CDI



- User requests pod using GPU to K8s
- If there is no adequate GPU, Kubernetes requests resource to CDI to attach GPU to a worker node
- Kubernetes maps a scheduled POD to the attached GPU
- CDI follows Sunfish architecture

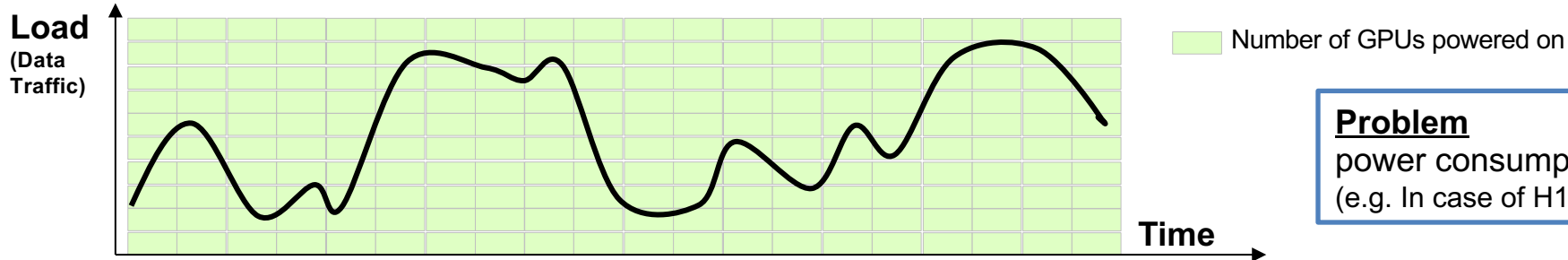


CDI USE CASE

Power saving by adjusting the number of GPUs in the base station

Conventional operation

The maximum number of GPUs are always running assuming maximum load. In the example below, 10 GPUs are always powered on

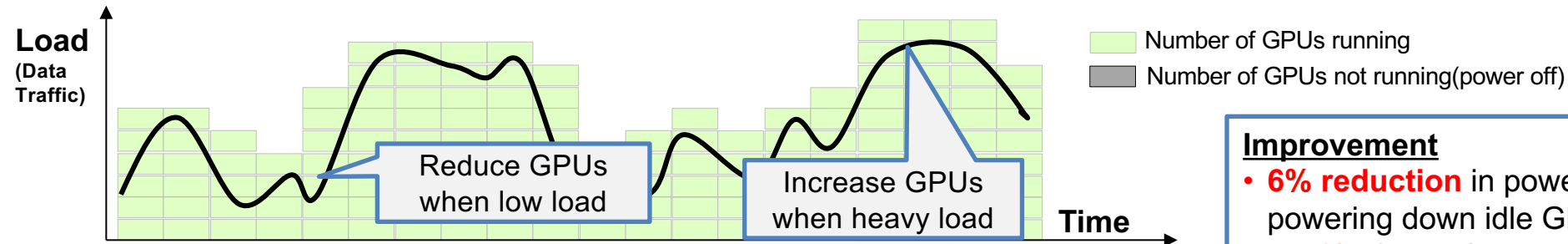


Problem

power consumption is constantly high (e.g. In case of H100, **49W at idle per GPU**)

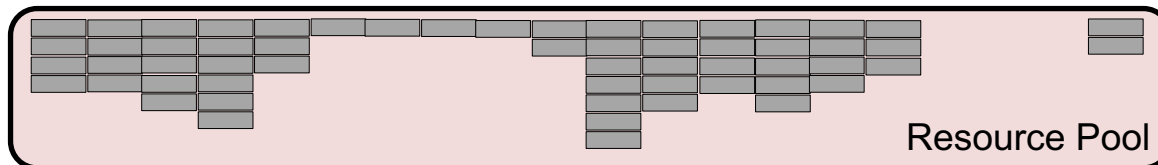
Solution by CDI

CDI automatically increases/decreases the number of GPUs based on load. Unused GPUs are returned to the resource pool to power down and save power



Improvement

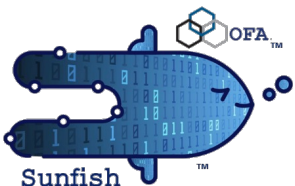
- **6% reduction** in power consumption by powering down idle GPUs
- **~25%** of total GPU cycles are then available for other workloads



SUNFISH IS FINALLY OUT!

The OFMF Workgroup is happy to announce the first official release of the Sunfish Framework

- Official documentation
- Reference software implementation

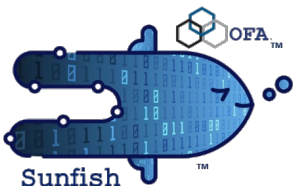


DOCUMENTATION

- **Requirements and normative references for implementing a fully compliant Sunfish Framework, Hardware Agent and Client**
 - Sunfish framework components design and interactions description
 - Interactions between Sunfish and Hardware Agents
 - Hardware Agents lifecycle management (registration, failover, etc.)
 - Redfish/Swordfish schema objects adopted
 - Additions to Redfish schema
 - Sunfish specific Redfish modeling requirements (e.g., CXL Fabric Attached Memory)

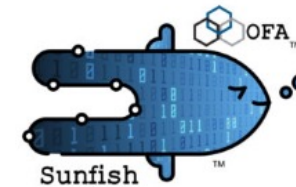


[Documentation download](#)



Sunfish Framework Documentation

v0.3



Sunfish OpenFabrics Management Framework for Composable Disaggregated Infrastructures

Version 0.3

ABSTRACT: Sunfish is designed for managing composable disaggregated resources over multiple fabrics using a central repository and an open-source API and toolset. Sunfish is designed for manipulating connected hardware resources using client-friendly RESTful abstractions and configuring fabric interconnects so that datacenter and AI workloads can be linked with available resources over dynamic fabric infrastructures.

The Sunfish OpenFabrics Management Framework API defines a RESTful interface and a standardized data model to provide data structures to help simplify the development of composable distributed, disaggregated, computer architectures. Sunfish contains abstract data structures that represent computer system resources, available network fabric components and management, current resource operational conditions, and abstracted representations of composed disaggregated computing systems.

Last Updated 04/23/2024

USAGE

Copyright (c) 2024 OpenFabrics Alliance (OFA). All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

The OpenFabrics Alliance hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

1. Any text, diagram, chart, table or definition reproduced must be reproduced in its entirety with no alteration, and,
2. Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced must acknowledge the OFA copyright on that material, and must credit the OFA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, or any portion thereof, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to the OFA.

REFERENCE SW IMPLEMENTATION

▪ Reference Sunfish Core Library

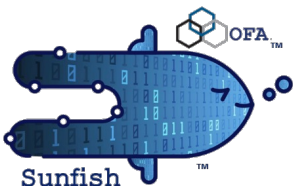
- Code: https://github.com/OpenFabrics/sunfish_library_reference
- Implements the Sunfish core services as a python library
 - RedFish tree management
 - Interactions with Hardware Agents
 - Events brokerage

▪ Reference Sunfish Server

- Code: https://github.com/OpenFabrics/sunfish_server_reference
- RESTful API for the Sunfish core library

▪ Reference Sunfish Hardware Agent

- Work on CXL Hardware Agent in progress
- Agent API to Sunfish Server being developed for CXL FAM
- Agent backend being developed for CXL fabric mock-ups



THE SUNFISH COMMUNITY

OPENFABRICS ALLIANCE

OAK RIDGE National Laboratory

intel.

Red Hat

CORNELIS NETWORKS

SNIA

Sandia National Laboratories

IBM

H3

IntelliProp

DMTF

Lawrence Livermore National Laboratory

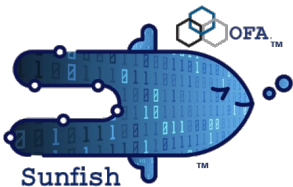
Hewlett Packard Enterprise

Fsas Technologies

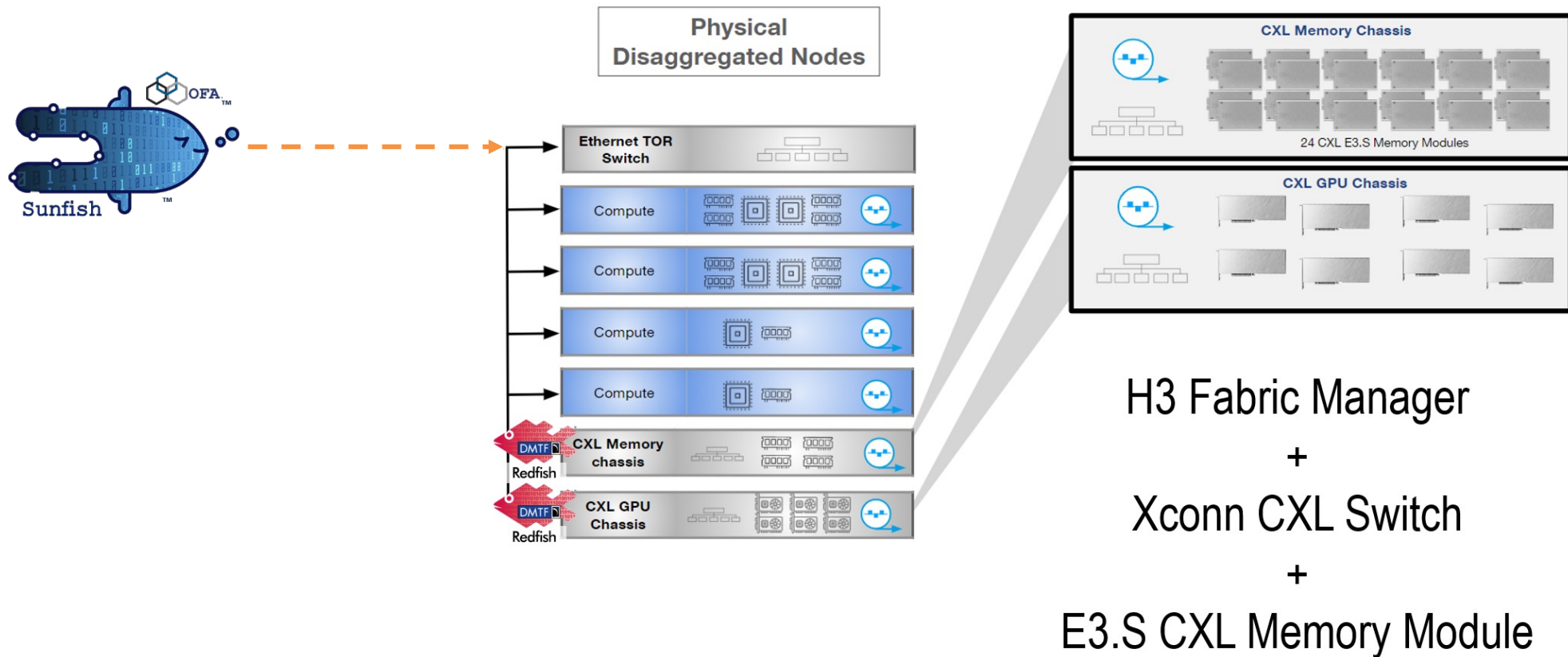
CXL Compute Express Link

DELL Technologies

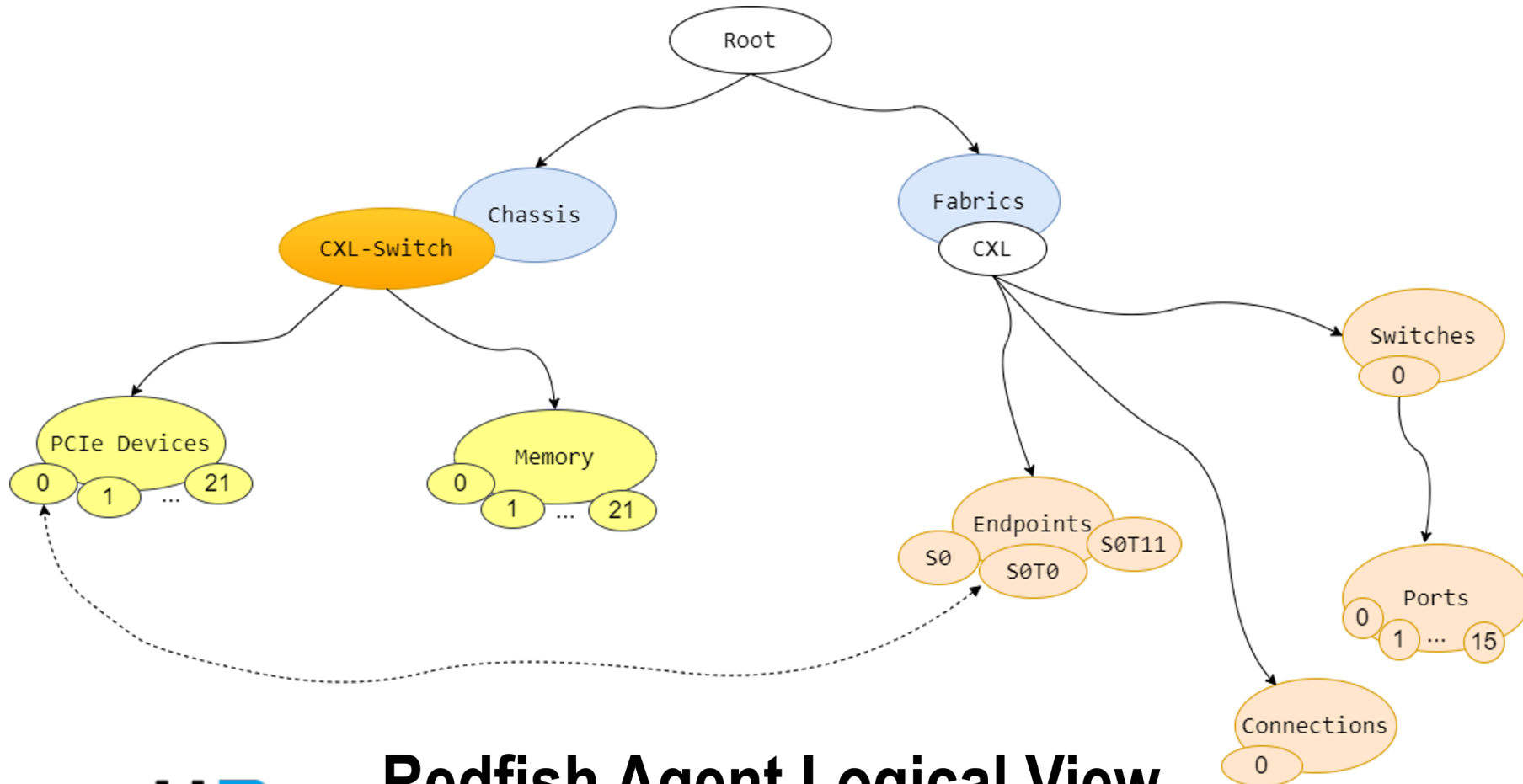
scalemem



FIRST HARDWARE AGENT FOR CXL MEMORY



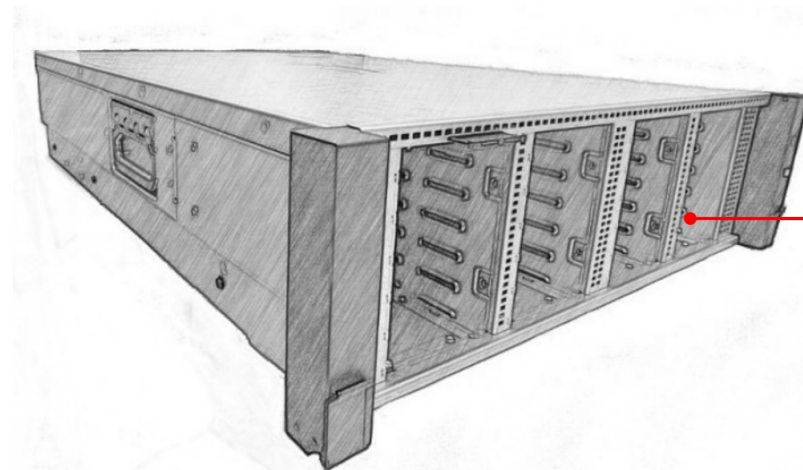
FIRST HARDWARE AGENT FOR CXL MEMORY



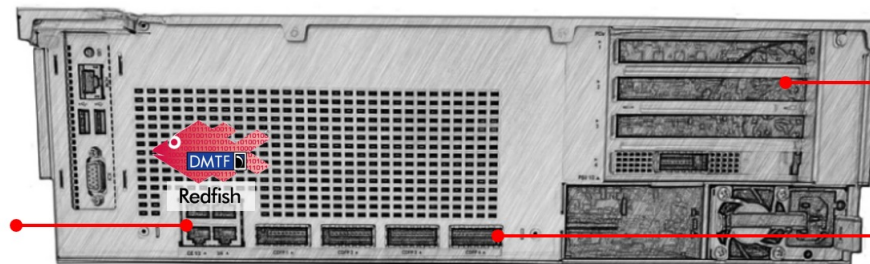
Redfish Agent Logical View

FIRST HARDWARE AGENT FOR CXL MEMORY

CXL Memory Solution



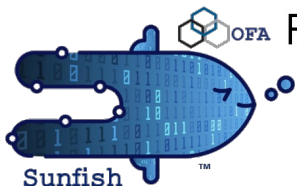
24 E3.S 2T Memory



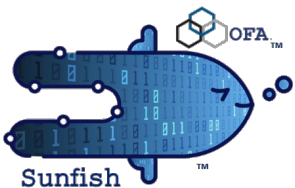
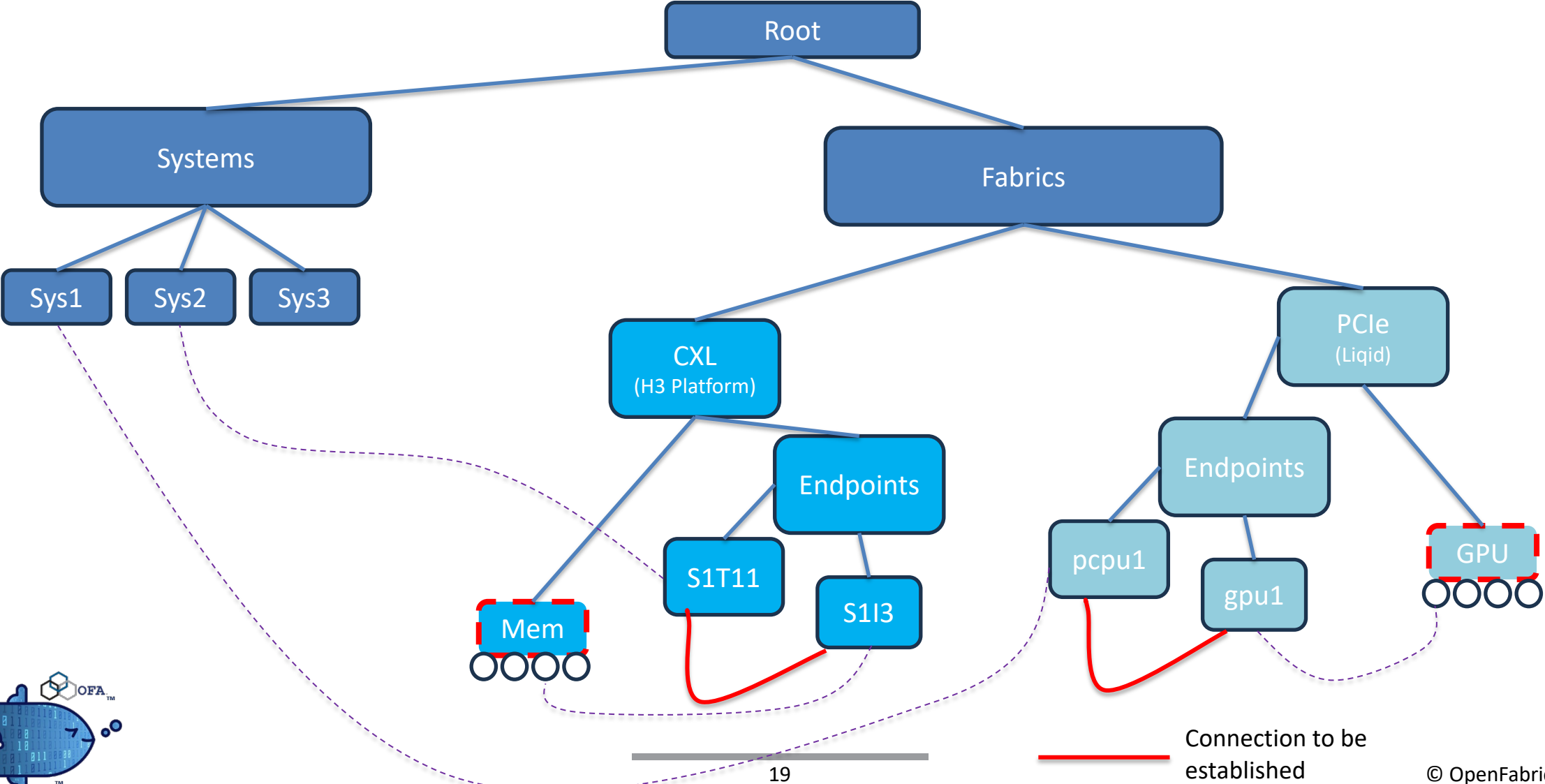
Expandable CXL ports

CXL host ports

Management port of mCPU for Redfish agent



DEMONSTRATION



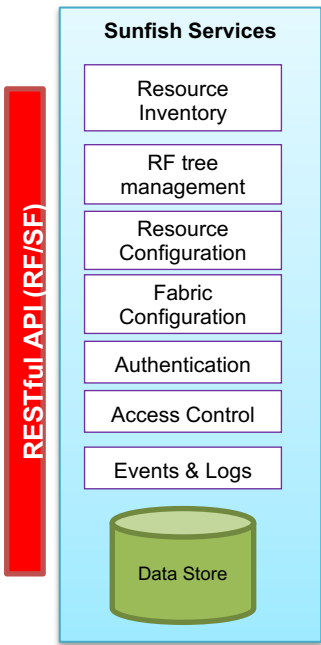
DEMONSTRATION

Clients

Sunfish Core Services

Sunfish Agents

Hardware Managers



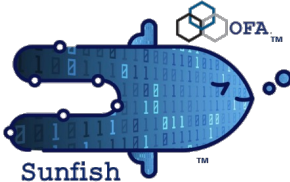
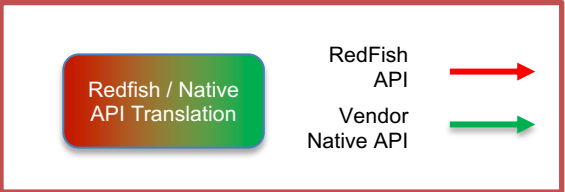
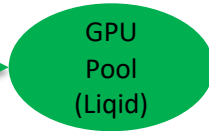
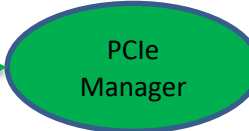
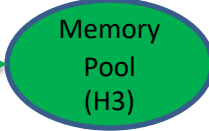
1. Registration Event

2. Agent tree crawling

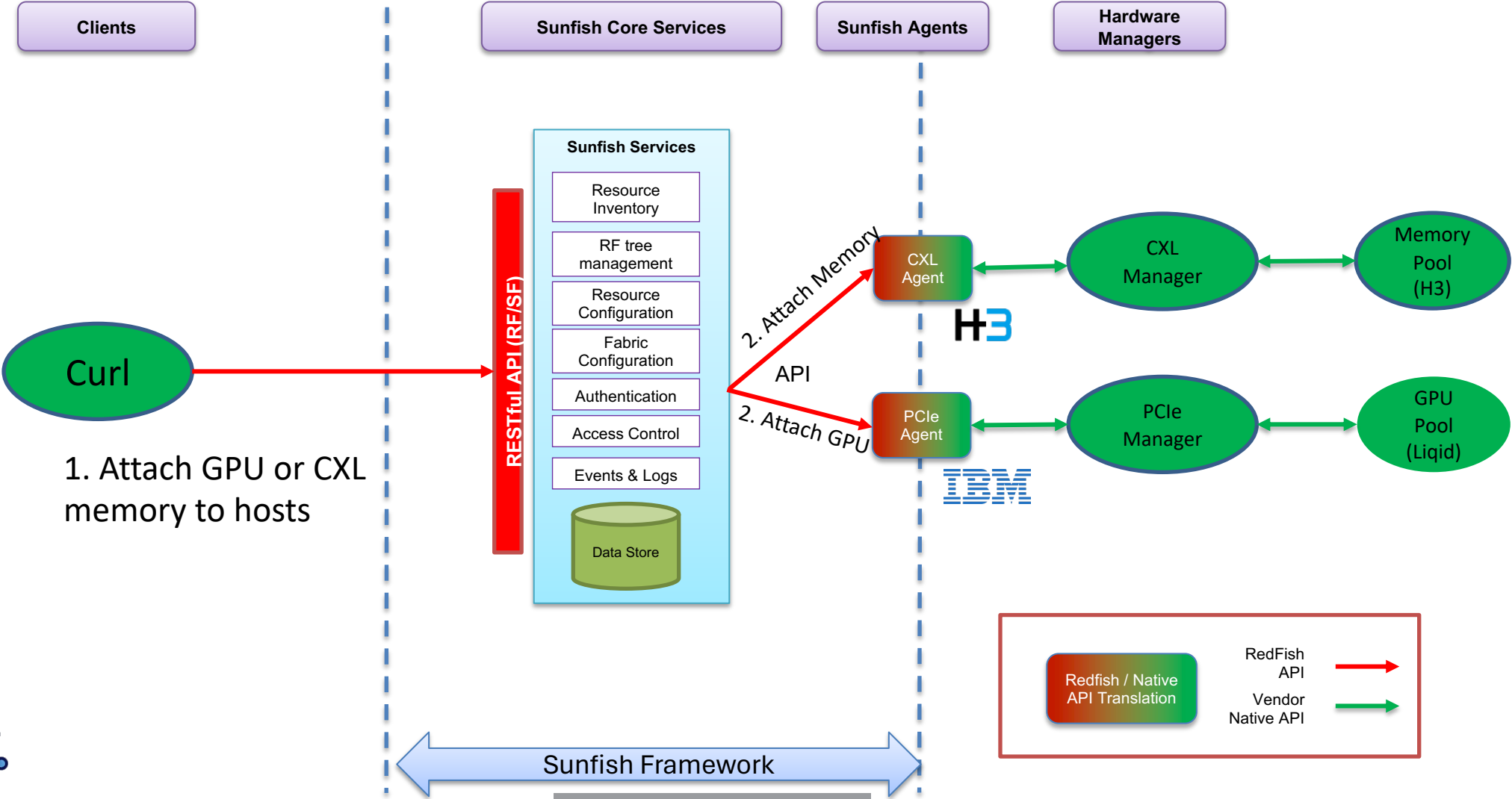
1. Registration Event

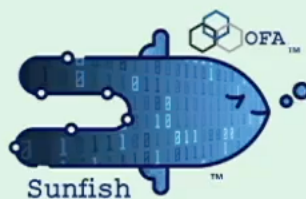
CXL Agent

PCIe Agent



DEMONSTRATION





Sunfish

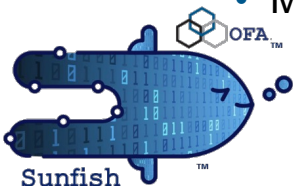
`/redfish/v1/`

```
{
  "@odata.id": /redfish/v1/,
  "@odata.type": "#ServiceRoot.v1_14_0.ServiceRoot",
  "AggregationServiceService": {
    "@odata.id": /redfish/v1/AggregationService/
  },
  "CompositionService": {
    "@odata.id": /redfish/v1/CompositionService/
  },
  "Id": "RootService",
  "Name": "Root Service",
  "RedfishVersion": "1.14.0",
  "Systems": {
    "@odata.id": /redfish/v1/Systems/
  }
}
```


CONCLUSIONS AND NEXT STEPS

- **The Sunfish community is rapidly growing, and we are targeting further hardware vendors for creating an ecosystem of agents.**
- **Focus on integrating with clients (e.g., Kubernetes, Flux, etc.) to demonstrate the value of a single API approach.**
- **Sunfish will be at SC'24 in Atlanta, GA**

- **Join the community:**
 - Contributions welcome:
 - Workload managers integration
 - Parallel computing libraries integration
 - More agents for real disaggregated hardware products
 - How to join
 - Meeting weekly on Fridays @7am Pacific Time
 - https://www.openfabrics.org/my-calendar/#mc_calendar_05_2802-calendar-details-my-calendar
 - Join the Mailing list:
 - <https://lists.openfabrics.org/mailman/listinfo/ofmfwg>
 - Reach out for information
 - Christian Pinto: christian.pinto@ibm.com
 - Michael Aguilar: mjaguil@sandia.gov





2024 OFA Virtual Workshop

THANK YOU

Christian Pinto

IBM Research Europe

