OFA Sunfish: New Applications for Distributed Storage with SNIA Swordfish®

SDC25

17 September 2025

Nathan Hanford, Computer Scientist

Prepared by LLNL under Contract DE-AC52-07NA27344.

What is Sunfish?

- Sunfish is an open-source standard reference framework for dynamic fabric and device reconfiguration and management, geared towards Composable Disaggregated Infrastructure (CDI) for both Enterprise and HPC applications.
- Sunfish interfaces with Redfish, Swordfish, CXL, and more APIs for device and fabric management.

The Disconnect

- Totally proprietary fabric and storage managers…
  - 👎 have dashboards that no one looks at
  - 👎 don’t work with other vendors’ products
  - 👎 become a maintenance nightmare
  - 😎 keep developers employed
- However, this can’t continue
  - Multi-tenant security is becoming critical
  - Lack of a standardized management framework is limiting customer adoption of new technologies:
    - Persistent memory
    - Fabric-attached memory
    - Near-node storage
    - Custom AI accelerators
    - Secure network and storage enclaves

How does Sunfish help?

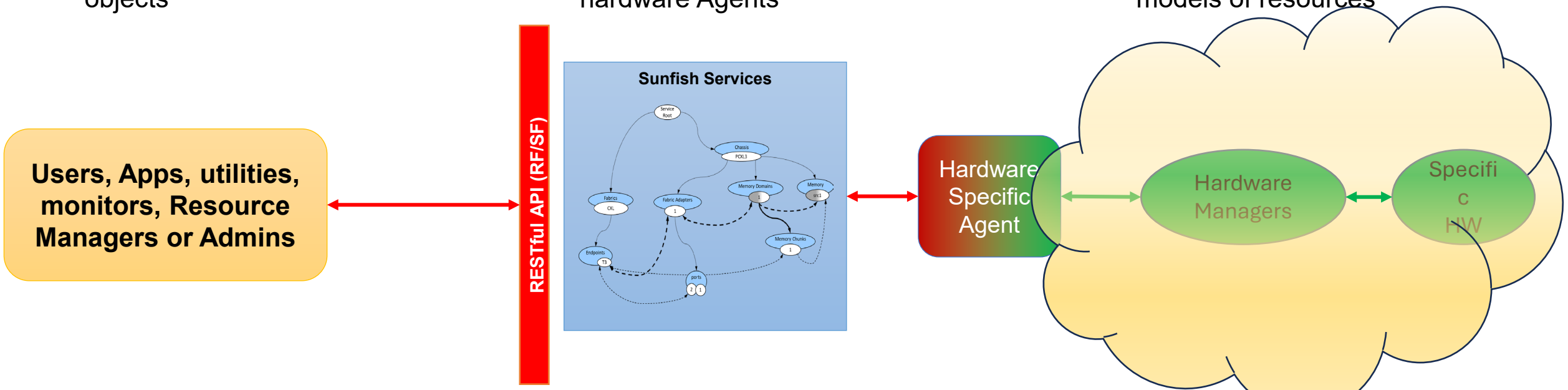
- Sunfish provides an open-source reference framework for a multi-device management solution.
- Your hardware team can provide a simple stable Redfish or Swordfish API, for example.
- Your software team can then implement an agent-based management solution that integrates seamlessly into your customer's datacenter.
- Metrics don't have to end in a dashboard. Standardization makes these data accessible to workload management software, and even the application itself.

The Sunfish Objective

Sunfish Clients see abstracted Fabric Attached Resource objects

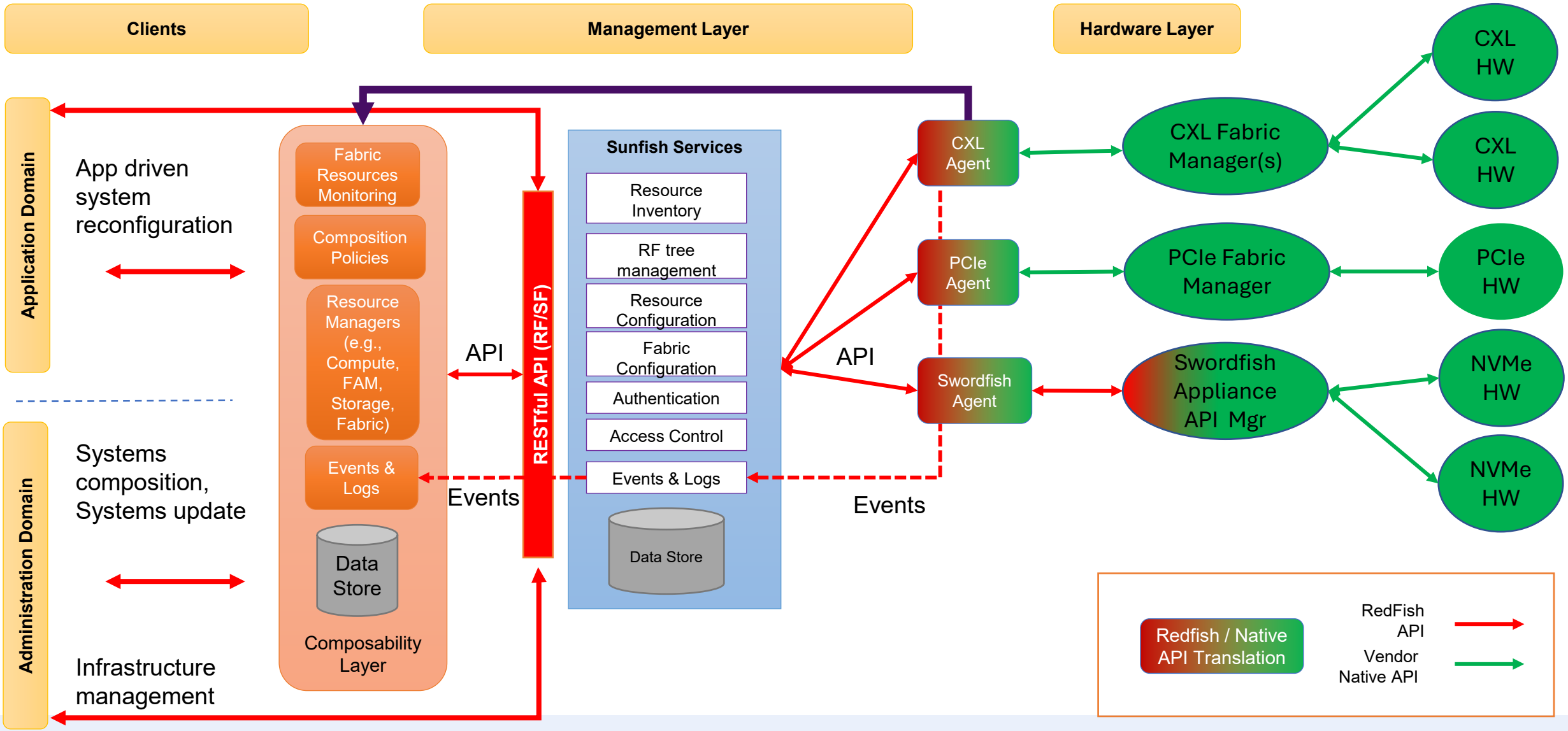
Sunfish Services manages the Redfish models of all resources from multiple hardware Agents

Sunfish Agents hide the hardware specifics by creating appropriate Redfish models of resources



Sunfish defines the policies that Agents follow when creating resource models so that Clients know how to interpret and manipulate them

The Sunfish Open Fabric Management Framework



Core Components

- Sunfish Library Reference
  - Provides a framework for event handling, etc.
  - Operates Database Backend
- Sunfish Agent and Server Reference
  - A Flask-based RESTful agent and server implementation to test the core library.

Sunfish-Compatible Hardware Agents

- Many different hardware manufacturers can easily create agents for Sunfish.
- Sunfish is based on Redfish, so agent creation is straightforward for other Redfish-based implementations.
- Our alpha reference agent/server emulator is available on Github here: https://github.com/OpenFabrics/sunfish_server_reference

Modularity and Scalability

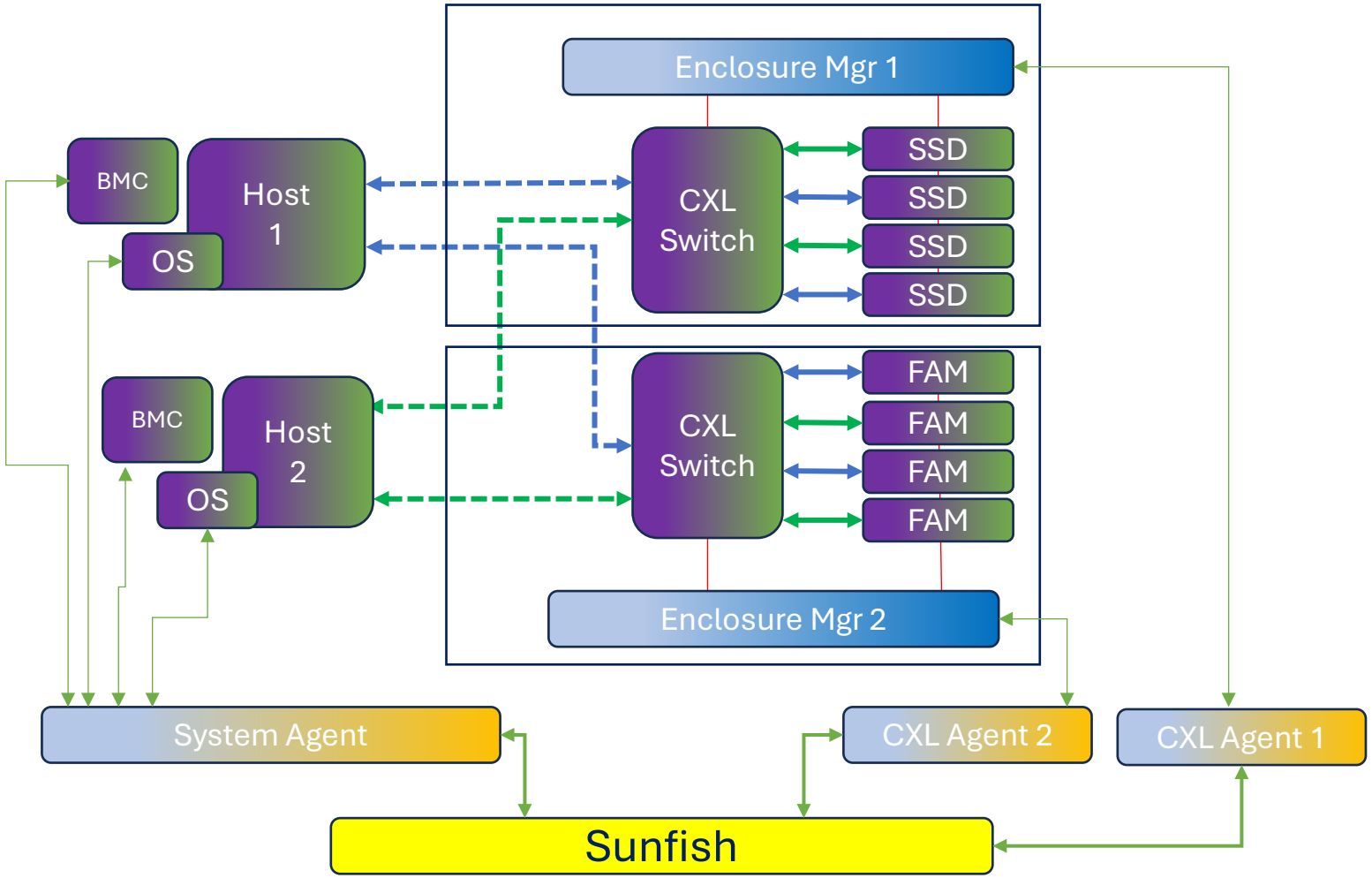
- If need be, custom agents can be created for your application
- Plugins can even be dynamically imported directly into the core library
- But the best option is to just implement to an existing standard API such as Redfish or Swordfish.

Merging Multiple Agent Resource Trees

- Another example of a problem you don’t want to solve
- Sunfish enables integration of heterogeneous fabrics with different resource description frameworks.
- The two important problems to solve when working with multiple Agents:
  - Sunfish must detect and resolve Redfish URI namespace conflicts
  - Sunfish must detect and resolve Boundary Component duplicates

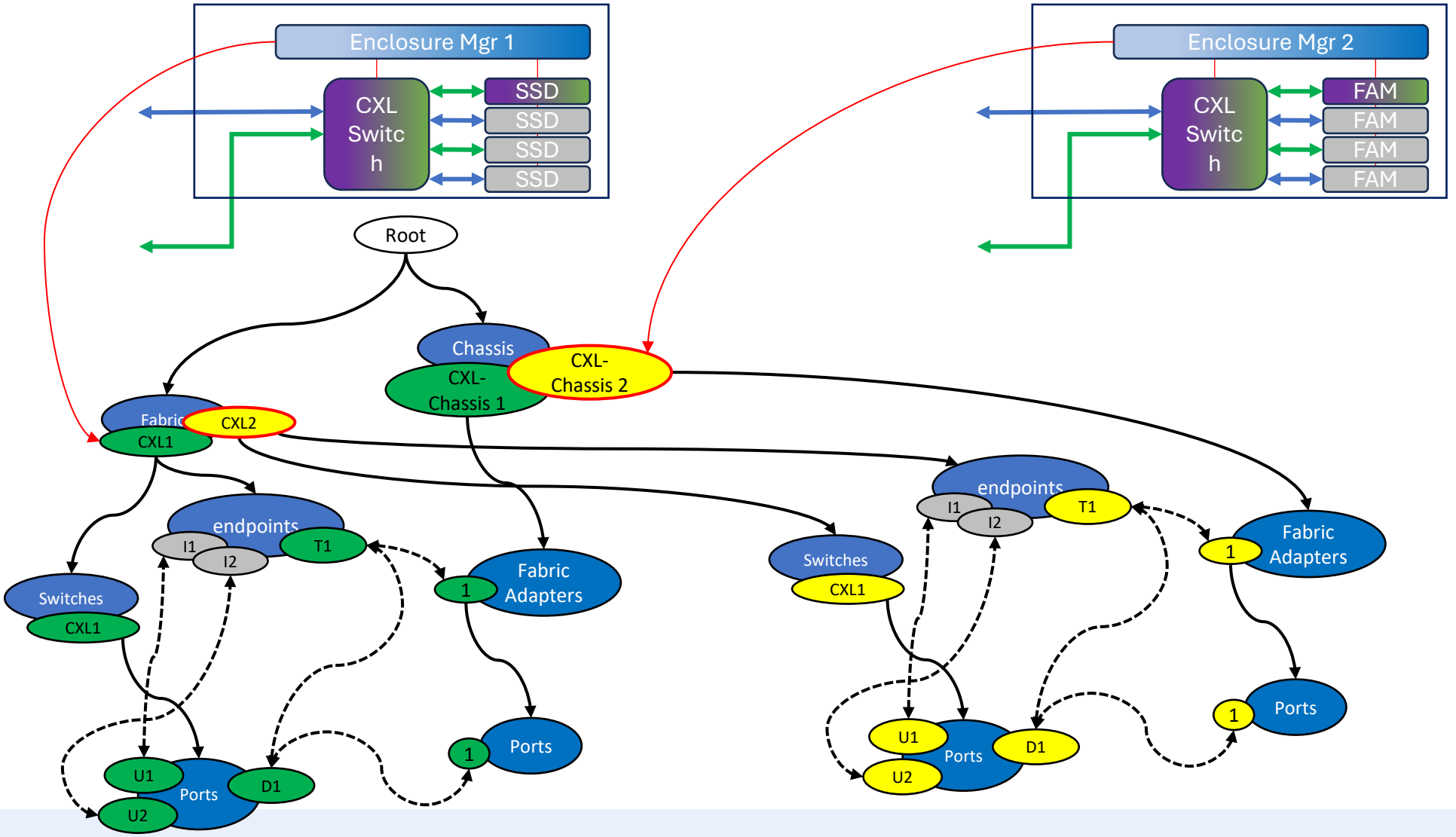
Managing Multiple Agents

- Sunfish Aggregates resources from multiple hardware managers
- Hosts have BMCs and OS
- Storage and FAM Appliances have Enclosure Managers
- Each of these hardware manager / Agent stacks has their own management domain and Redfish URI namespace
- In this topology, each CXL Enclosure is an independent CXL Fabric
  - No Switch-to-Switch links
  - Single level switch topology
- Sunfish Service eliminates duplicate URIs that may occur because the two Enclosure Managers may both use the same Redfish URIs
- Sunfish Service also needs to merge the Redfish URI namespaces of the System Agent and the two CXL Agents
- Dotted links are Boundary Links that cross hardware manager domains
  - E.g. links between a Host and an enclosure switch



A Merged Redfish Model of 2 Fabric Appliances

- Merging multiple Redfish URI namespaces is relatively easy if the actual hardware / logical instances are independent
- Sunfish keeps track of all URIs it translates from a given Agent's URI namespace to the URI used in the Sunfish URI namespace
- Sunfish must validate that the Fabrics and Chassis involved are completely independent
- Every physical device on every fabric needs to have a hardware ID that is unique within the scope of the Sunfish Services
- Sunfish keeps a list of all hardware IDs and searches for duplicates as each new resource is inserted into the database



Why Sunfish for Storage?

- Disaggregated storage and persistent memory is becoming increasingly popular in HPC and AI datacenters for performance reasons.
- Complex in-storage compute processes, such as data cleaning, are best managed by dedicated hardware.
- Speed and throughput requirements are outpacing traditional storage array capabilities.

Composable Storage Infrastructure

- Distributed storage can be located closer to the compute nodes and provide rich services.
- Example: *Rabbits* in LLNL's El Capitan supercomputer provide ephemeral and persistent Lustre filesystems to compute nodes located within a rack.

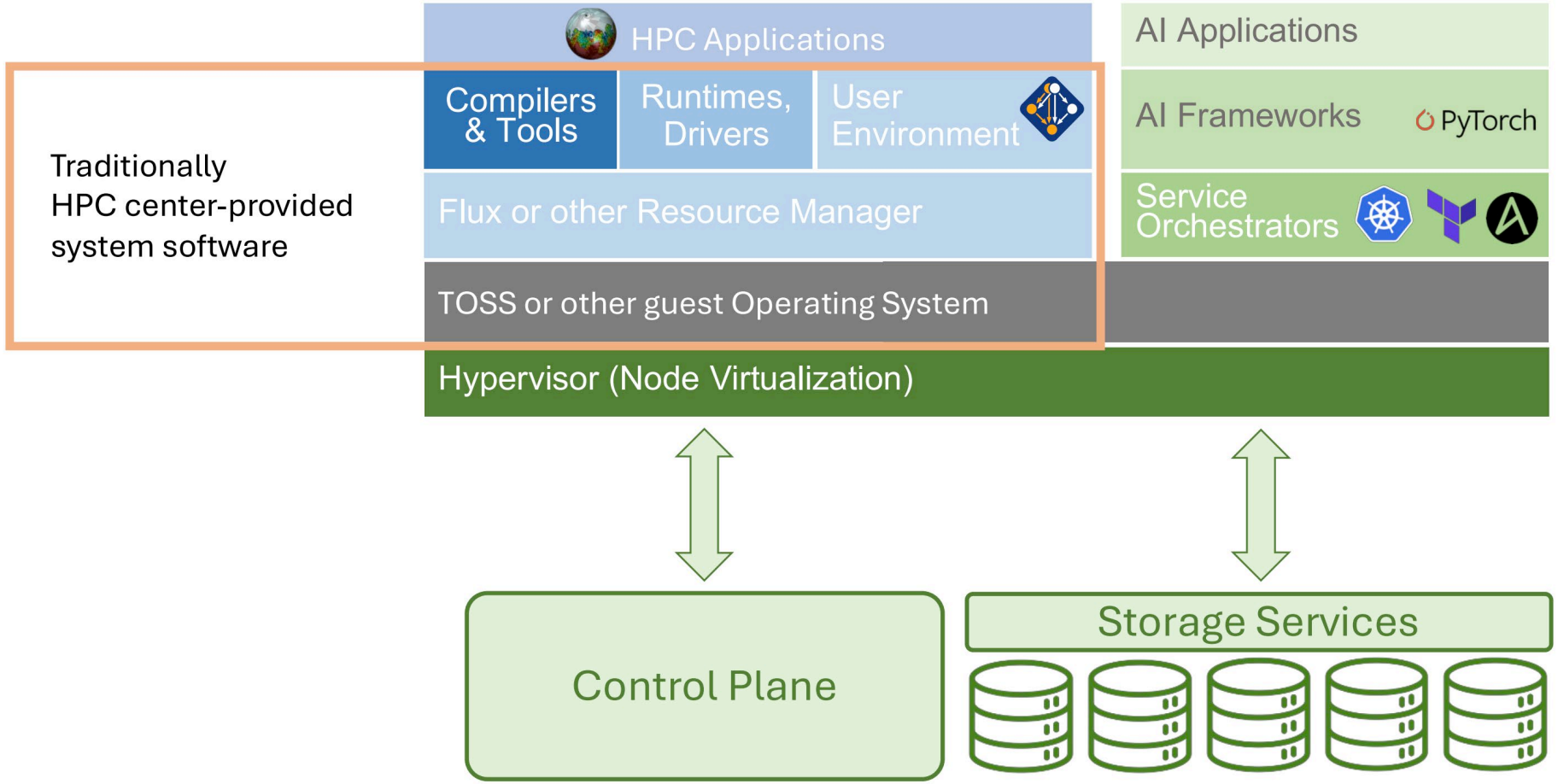
AI/ML Applications

- AI/ML Needs
  - Robust hardware support for information encryption and separation
  - Secure AI/ML Accelerator Appliance or Cluster online partitioning
- AI Accelerator appliance manufacturers can add agents to Sunfish to
  - Report resource availability
  - Report cluster status
  - Dynamically provision and link accelerator resources to applications
  - Report metrics back to applications
- AI/ML workloads can not exist in a vacuum.
  - Secure access to data
  - Co-processing with existing applications

Workload Isolation

- Multi-tenancy on shared hardware resources is rapidly becoming the norm.
- Containers provide some isolation, but lack rich control of hardware security features.
  - i.e. VNI and VXLAN provisioning
  - Secure storage and scratch-space provisioning
- Sunfish can be used to configure robust hardware security features across vendors

Example: HPC Center of the Future



Integration with the Flux Workload Manager

- Flux is an HPC workload manager from LLNL similar to Slurm but hierarchical in nature
- Supports multi-constraint scheduling with plugins
- Naturally, we are working on a standard API for scheduling heterogeneous storage and AI accelerator resources

Cross-Domain Integration

- “People don’t know what they want until you show it to them.”
- Enterprise and HPC customers have been caught by surprise in the past several years realizing they need more than an “AI/ML story.”
- Public models are not going to work in many Enterprise and HPC applications for both technical and regulatory reasons.
- Adopting open management standards like Sunfish puts solutions providers and customers in the best possible position.
  - Maximum interoperability
  - Minimum solution delivery time
  - Minimum maintenance cost

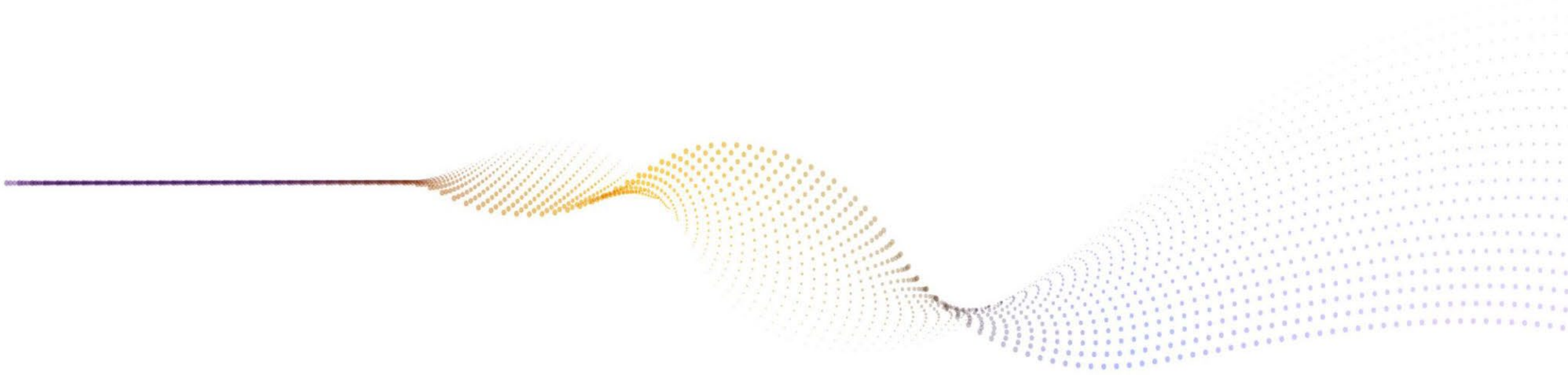
What you can do

- Check out OFA Sunfish
- Choose an open-source management API (i.e. Swordfish, Redfish, etc.), and implement to it in earnest.
  - Represent hardware and features accurately
  - Avoid breaking changes

Questions?

Read the docs here!
https://github.com/OpenFabrics/sunfish_docs

Join our development here!
https://www.openfabrics.org/openfabrics-management-framework/



Thank you for attending!

Please remember to rate this session. You get access the presentations at
http://sniadeveloper.org/conference