

STORAGE DEVELOPER CONFERENCE



*BY Developers FOR Developers*

# Sunfish Open Source Management for CDI

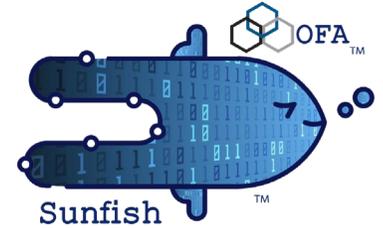
Sunfish, Redfish, and Swordfish

Russ Herrell, Hewlett Packard Enterprise

9/18/23

# Agenda

- Sunfish Overview
  - Redfish, Swordfish, and Sunfish, oh Why?
- Sunfish and Agent Communications
  - How they link up
  - Event driven conversations
- Sunfish / Agent Start Up Demo
- Sunfish and Multiple Agents
  - A simple example of a tough problem
  - The role of Sunfish in the solution
- Status of Sunfish
- Summary and Wrap-up

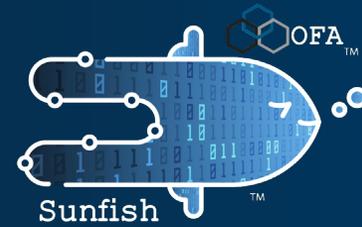




# Sunfish Overview

What is Sunfish?

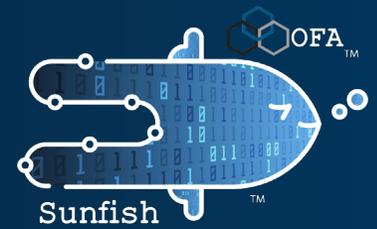
# Composable Disaggregated Infrastructure (CDI)



- CDI enables assigning *pools* of resources to consumers
  - Started with disaggregated storage
  - Moving towards disaggregated memory and disaggregated accelerators
  - Assigned resources may be private to or shared among consumers
- CDI requires one or more interconnecting *fabrics*
  - Disaggregated storage is already supported on several fabrics
    - For example: Infiniband, Ethernet, PCIe and FiberChannel
  - Disaggregated memory requires a memory semantic fabric
- CDI needs to avoid disaggregated management stacks
  - Disaggregated resources come with independent management tools

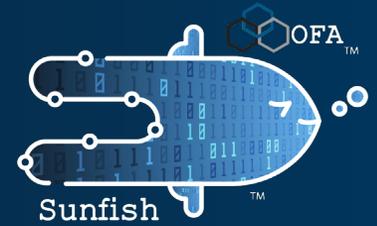
Sunfish provides a framework for wrangling the multitude of independent management tools behind a single, consistent, standards-based API

# Sunfish



- Composable disaggregated infrastructures provide a promising solution to addressing the provisioning and computational efficiency limitations, as well as hardware and operating costs, of integrated, siloed systems. But how do we solve these problems in an open, standards-based way?
- The Sunfish project, a collaboration between DMTF, SNIA, the OFA, and the CXL™ Consortium to provide elements of the overall solution, with Redfish® and SNIA Swordfish™ manageability providing the standards-based interface.
- Sunfish is designed to configure fabric interconnects and manage composable disaggregated resources in dynamic High Performance Computing (HPC) infrastructures using client-friendly abstractions.
- This presentation will highlight an open, standards-based approach to composable resource management for disaggregated infrastructures through Sunfish and provide a demonstration of the Sunfish reference implementation.

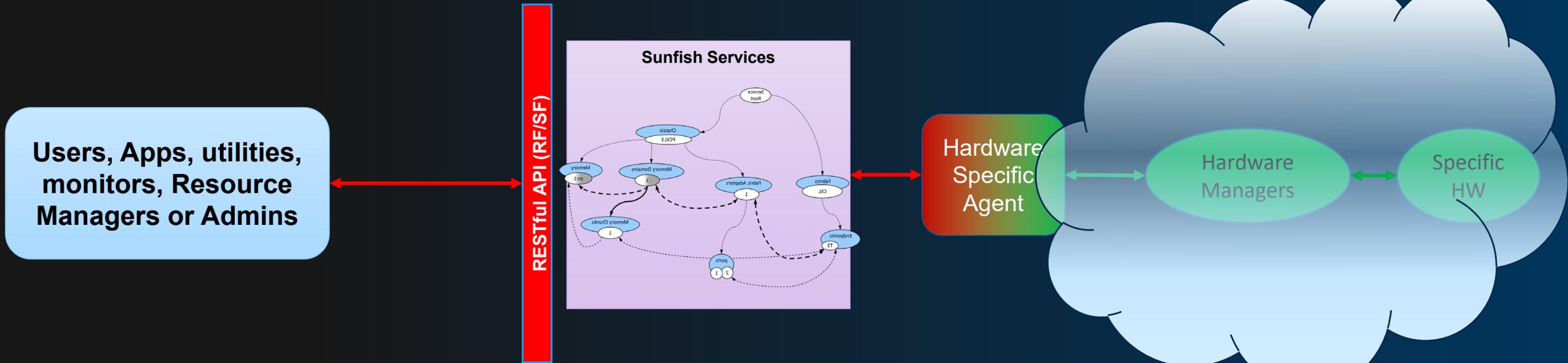
# The Sunfish Objective in Visual Form



Sunfish Clients see abstracted Fabric Attached Resource objects

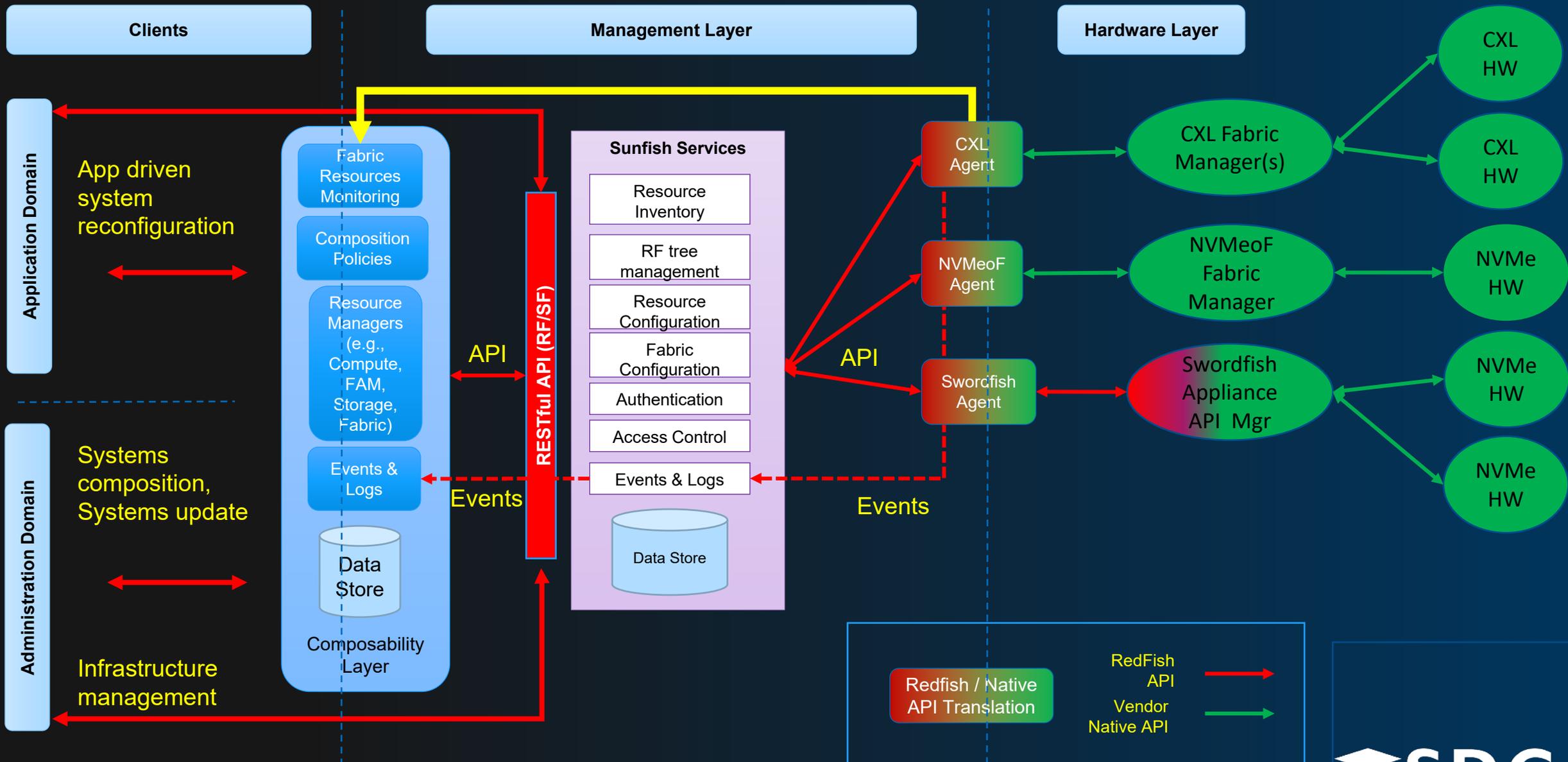
Sunfish Services manages the Redfish models of all resources from multiple hardware Agents

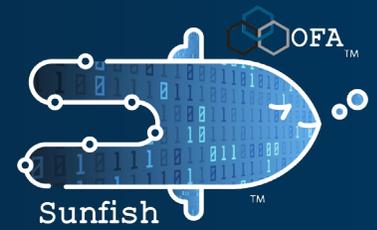
Sunfish Agents hide the hardware specifics by creating appropriate Redfish models of resources



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

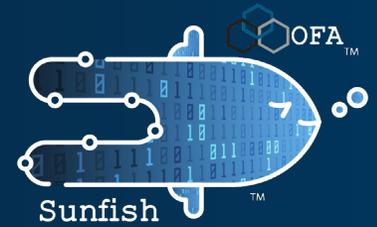




# Sunfish - Agent Communications

How does Sunfish communicate with a fabric Agent / Manager?

# Sunfish and Agent Communication



- Sunfish communications with its Agents are Event Driven
- Agents are resource aggregators that present Sunfish with a Redfish / Swordfish model of all resources which they manage and/or model
- Agents send Events to Sunfish
  - To start the Sunfish – Agent interface
  - To alert Sunfish of the presence of new resources
  - To alert Sunfish of health and status changes in existing resources
  - To inform Sunfish of Events received from resources the Agent manages
- Sunfish registers for Events associated with resources
- Sunfish sends Redfish and Swordfish API calls to Agents
  - To query the latest status, health, or configuration of the Agents' resources
  - To request changes in state or configuration of Agents' resources
  - To Create or Destroy logical Redfish and Swordfish objects which impact Agents' resources
- Goal is to keep the Sunfish resource models current without 'polling'

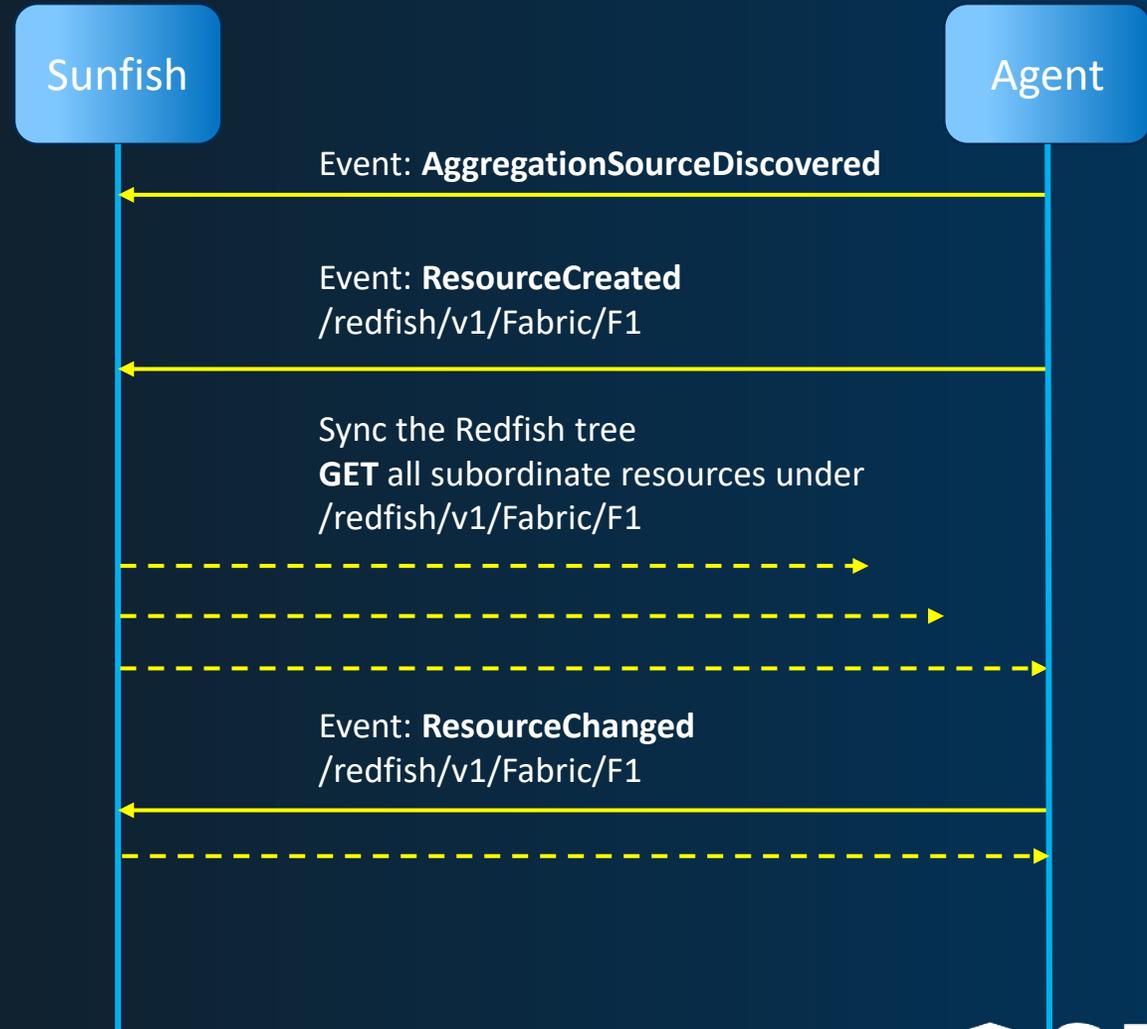
# Sunfish and Agent Event-driven Communication

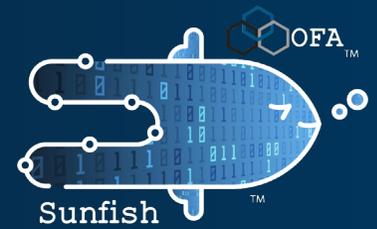


- Agent issues Event to Sunfish registering itself as an AggregationSource
  - Sunfish creates the appropriate objects in its resource tree and acknowledges the Event
- Agent issues Event to Sunfish reporting a new resource (a new Fabric)
- Sunfish responds with a recursiveFetch of all subordinate and linked resources under the newly created resource

## If anything changes

- Agent sends Event to Sunfish alerting of a change in a resource
- Sunfish responds with a fresh query of the resource





# Sunfish Agent Registration Demo

Michele Gazzetti  
IBM Research Europe - Ireland



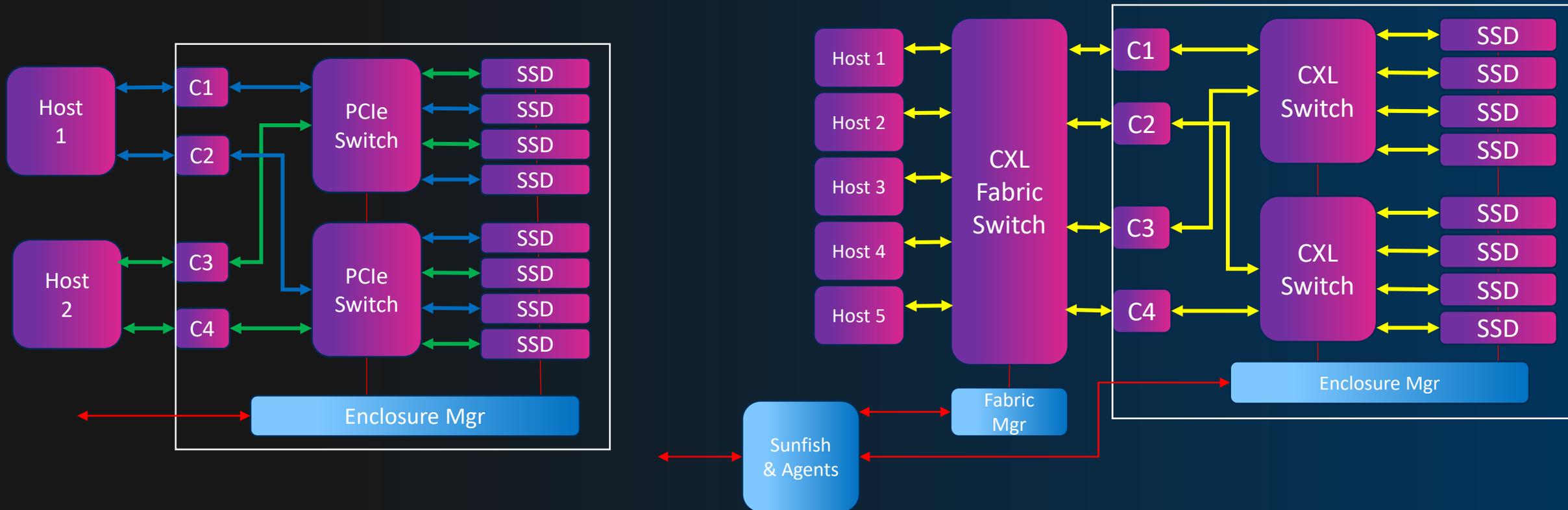




# Sunfish and Multiple Agents

How does Sunfish make things scale easier?

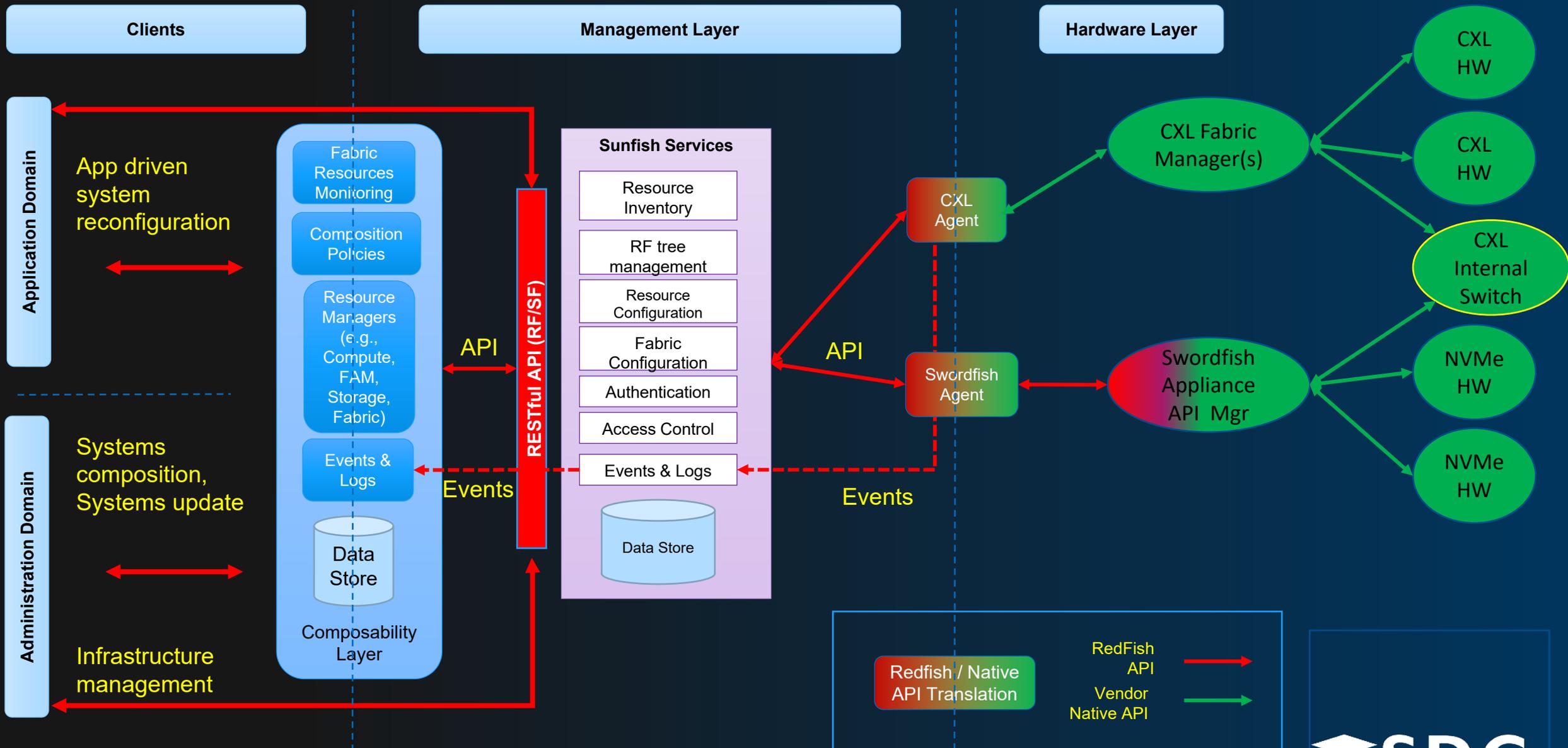
# Why Sunfish?: A Simple Disaggregated Infrastructure Example



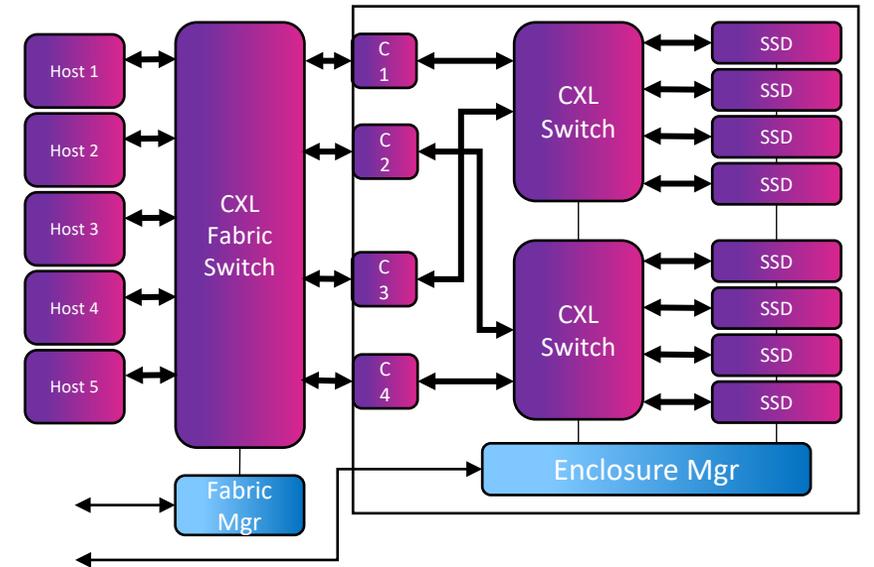
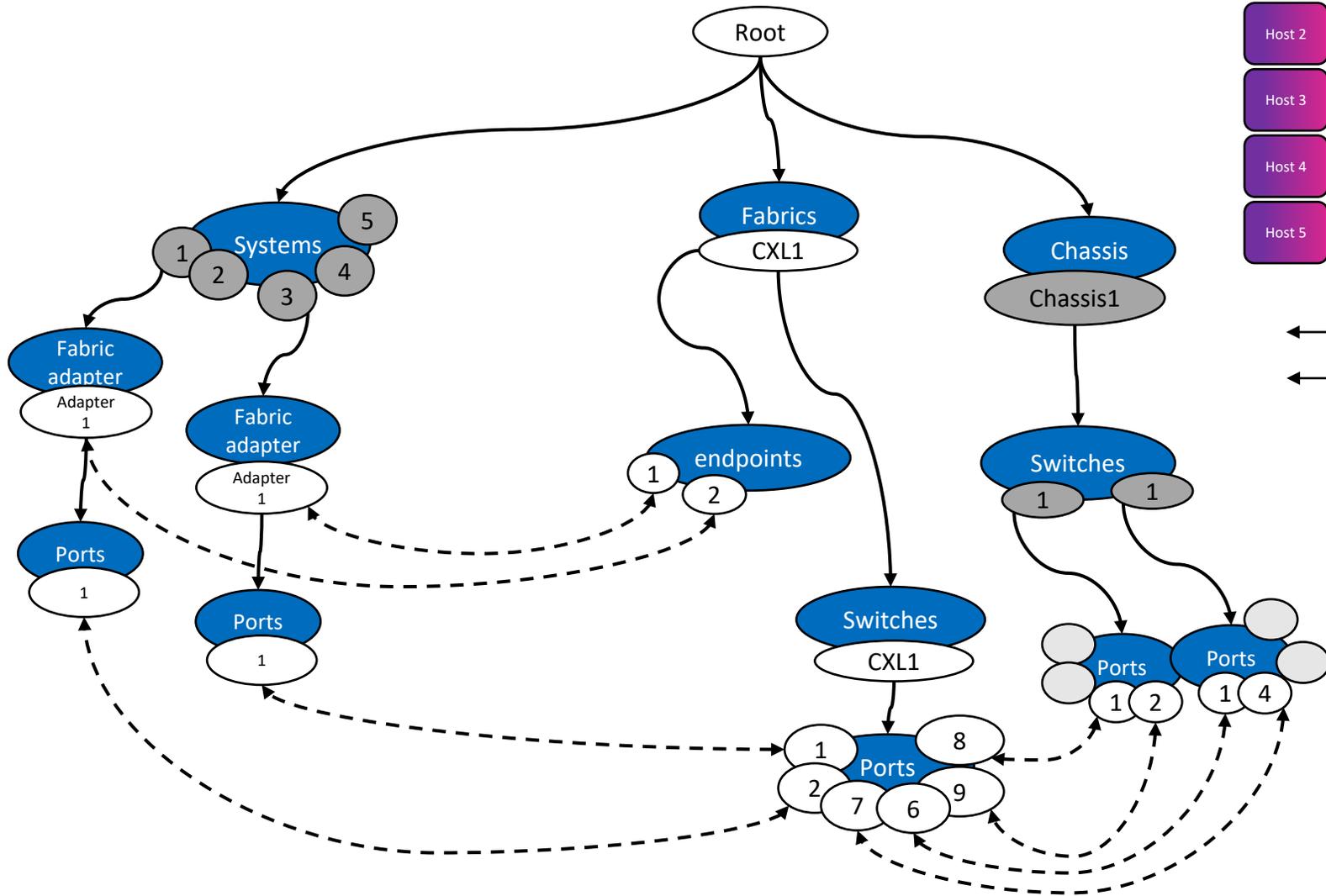
- From a shared JBOD

- To a shareable pool of NVMe Capacity

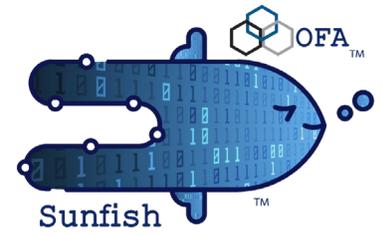
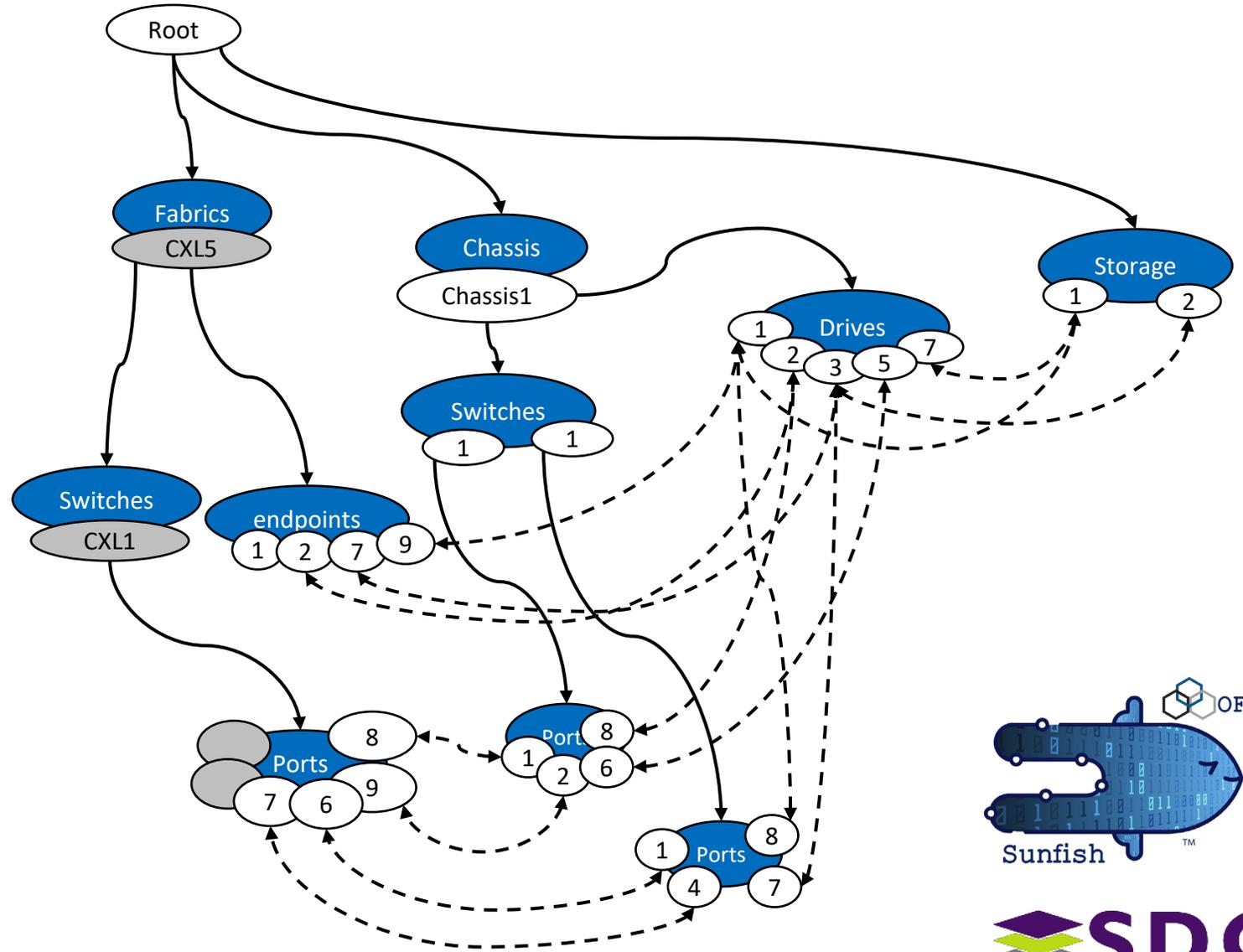
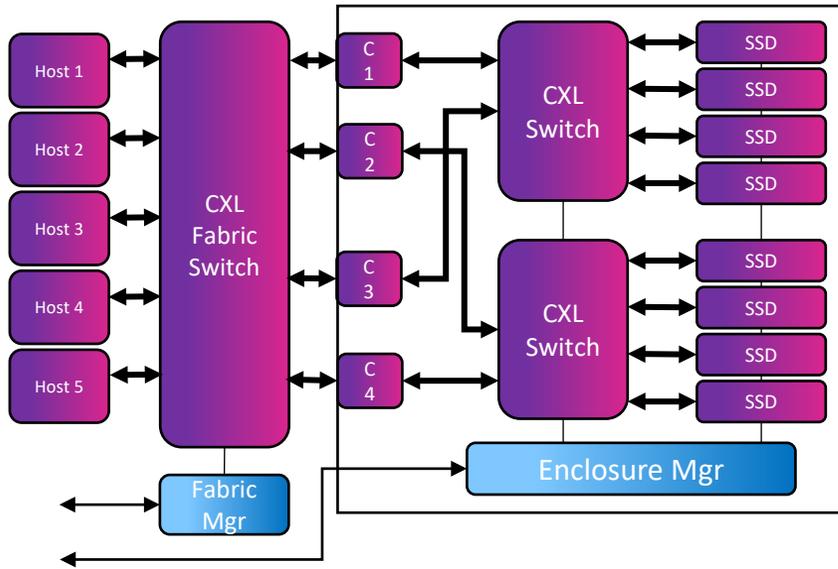
# The Sunfish Open Fabric Management Framework



# CXL Fabric Manager View

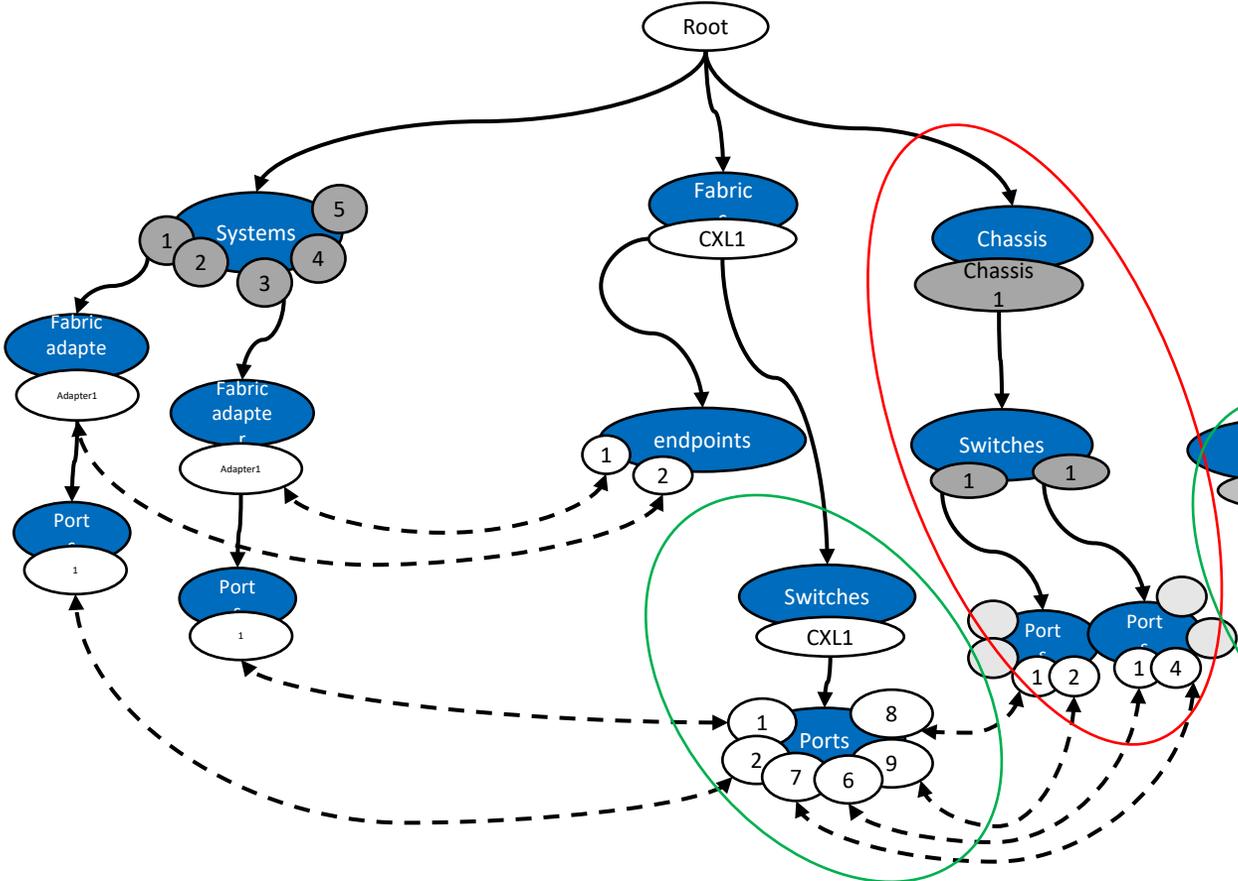


# Enclosure Manager View

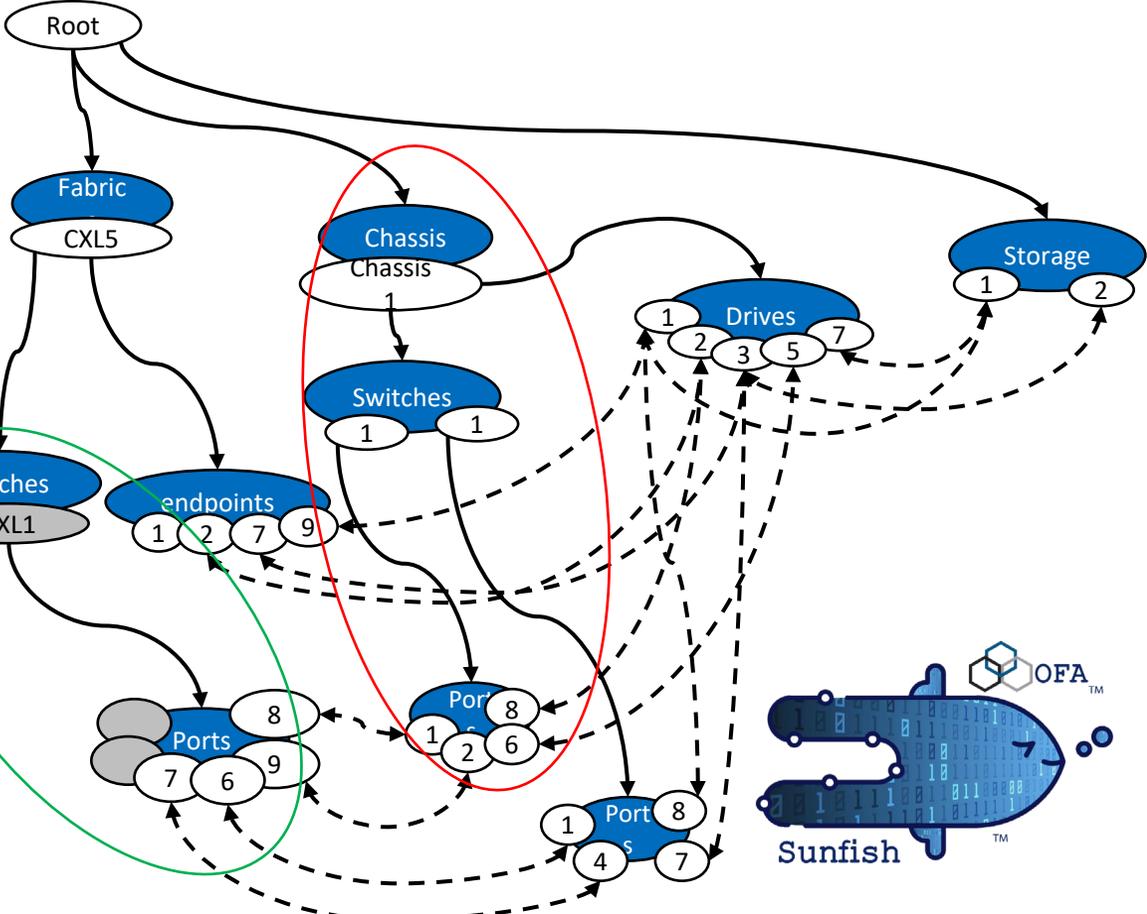


# The Two Views Need To Be Merged

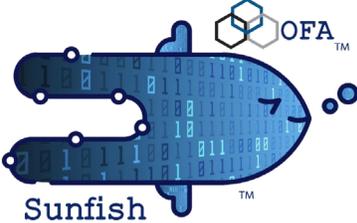
CXL Fabric Manager View



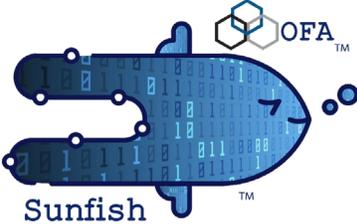
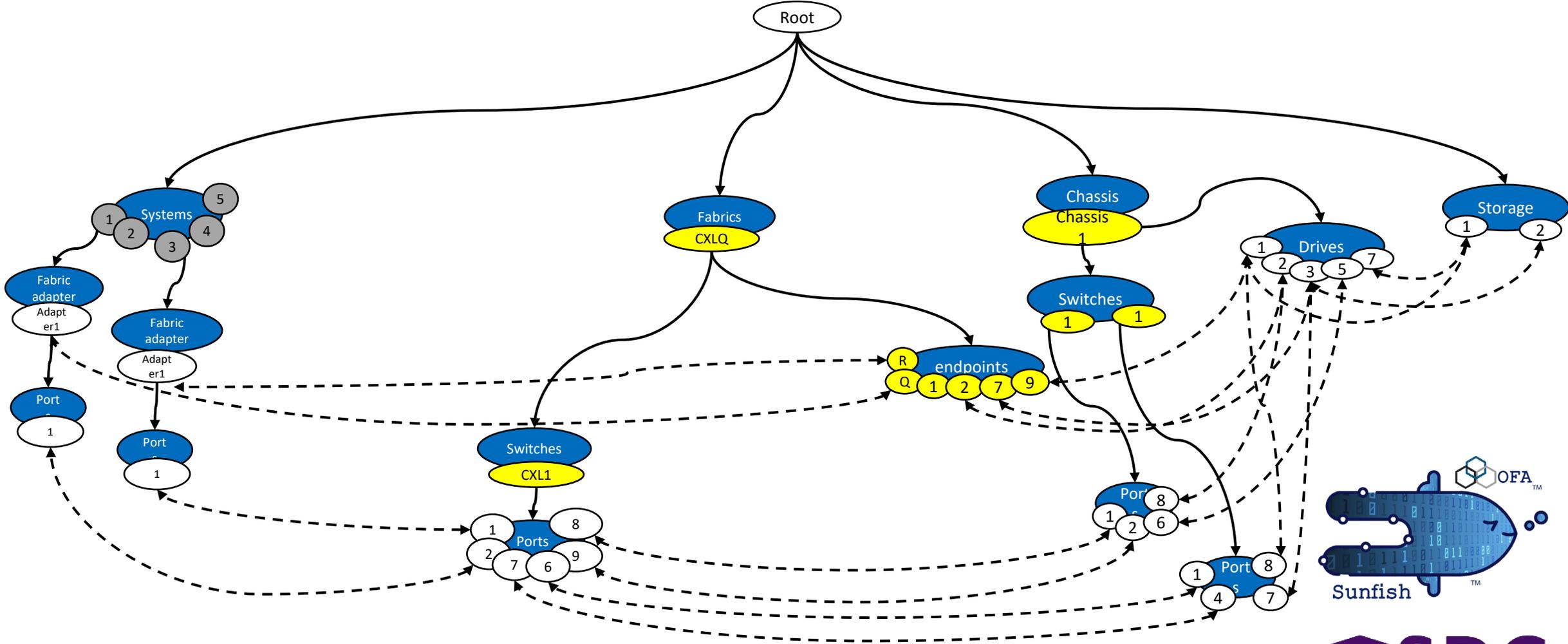
Enclosure Manager View



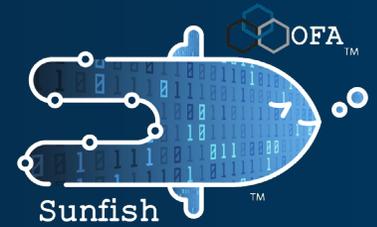
Different fabrics may require different methods to detect boundary links and resolve boundary component mergers



# The Two Views After Merging by Sunfish

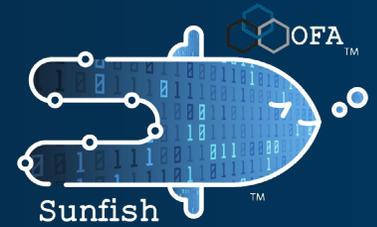


# Status of Sunfish



- Working through common use cases and tasks
  - Sunfish – Agent start up
  - Client discovery of resources via Sunfish inquiries
  - Binding of resources to consumers (e.g. creating Connections)
  - Documenting modeling requirements for Agents (e.g. how to model FAM)
- Soliciting new use cases and tasks, new fabrics, and new resources
  - If you have something your business needs modeled and managed, talk to us
- Generating reference code implementations of Sunfish framework components
  - Agent templates and reference Agents for specific fabric managers
  - Sunfish API Services and Sunfish Event Services
  - Sunfish Composability Clients and utilities
- Working on Redfish and Swordfish schema
  - Recent modeling recommendations for FAM

# Summary and Wrap-up

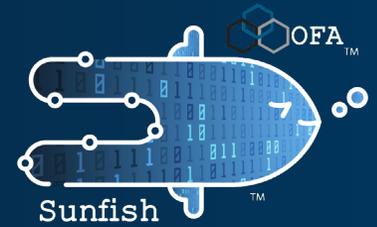


- The Sunfish Open Fabric Management Framework is intended to present one comprehensive, well-defined view of fabric composable resources
- The OFA Sunfish workgroup is collaborating with DMTF, SNIA, and the CXL Consortium
- Reference code implementations of many of the framework components will be rolled out over time as required features are architected, coded, and validated, and as related standards become available

We encourage you to:

- Review the Sunfish project collateral
  - OFA Sunfish website: <https://www.openfabrics.org/openfabrics-management-framework/>
  - Sunfish github site: <https://github.com/OFMFWG>
- Join the OFA Sunfish effort and contribute use cases, solutions and code examples

# Thank You



## ■ Sunfish Working Group Members

- Atul Sandur (AMD)
- Christian Pinto (IBM)
- Doug Ledford (Redhat) Co-chair
- Elham Hojati (Intel)
- Jeff Hilland (HPE)
- Jim Hull (Intelliprop)
- John Mayfield (HPE)
- Michele Gazzetti (IBM)
- Michael Aguilar (Sandia) Co-chair
- Phil Cayton (Intel)
- Richelle Ahlvers (Intel)
- Rajalaxmi Angadi (Intel)
- Russ Herrell (HPE)
- Tracy Spitler (Intelliprop)

## ■ Sunfish Working Group Contact Info

- Doug Ledford [dledford@redhat.com](mailto:dledford@redhat.com), Michael Aguilar [mjaguil@sandia.gov](mailto:mjaguil@sandia.gov)

## ■ Speaker Contact Info

- Russ Herrell, [russ.herrell@hpe.com](mailto:russ.herrell@hpe.com)