SC23

Denver, CO | i am hpc.

Centralized Composable HPC Management with the OpenFabrics Managment Framework

Phil Cayton (Intel), Michael Aguilar (Sandia Labs), Christian Pinto (IBM Research)

Centralized Composable HPC Management with the OpenFabrics Managment Framework

Michael Aguilar (Sandia Labs), Phil Cayton (Intel), Christian Pinto (IBM Research)

RESDIS23 Workshop---SC23 Supercomputing Conference

November 17, 2023

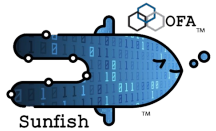
Denver, Colorado

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2023-11721C

https://orcid.org/0000-0001-7060-2742

Integration of BeeGFS on Demand with Composable Disaggregated Infrastructure

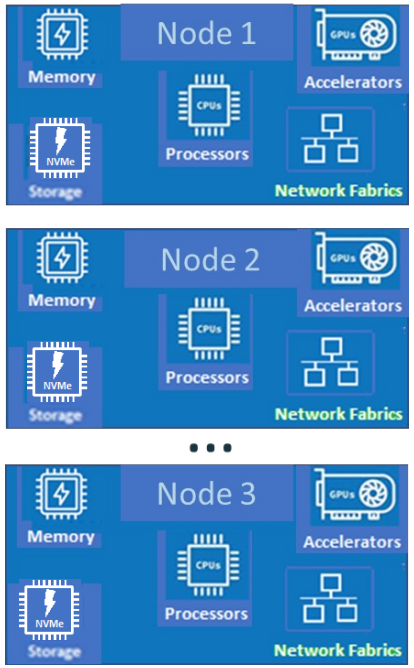
1. Quick overview of Composable Disaggregated Infrastructure (CDI)
2. Design Considerations for a Composability Manager
3. Introducing Sunfish 

4. Sunfish Core Services
5. Sunfish Hardware Agents
6. The Sunfish Composability Management Framework
7. Acknowledgements and Questions

Quick overview of Composable Disaggregated Infrastructure

Traditional compute clusters are created by combining compute servers over network fabrics

- individual compute servers are statically provisioned
- often results in overprovisioning or stranded resources

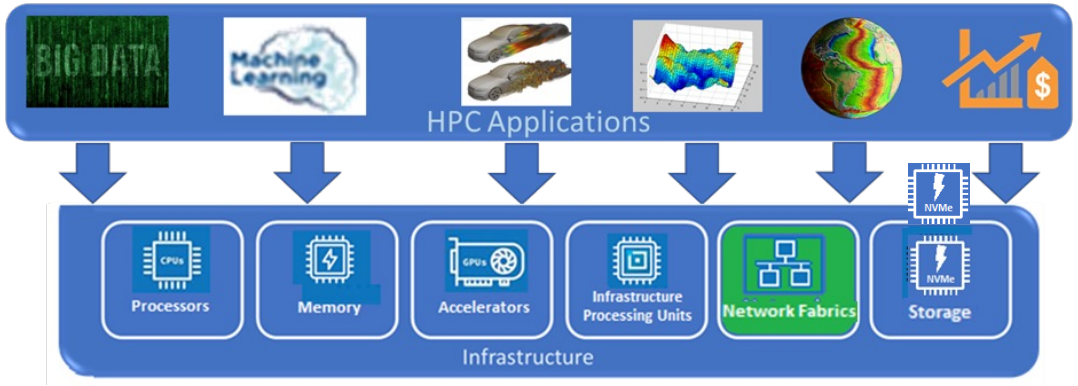


Composable Disaggregated Infrastructure (CDI)

- Computational resources are physically separated over high-speed/low-latency fabrics
- Computational resources are dynamically composable, as needed, into a computer system

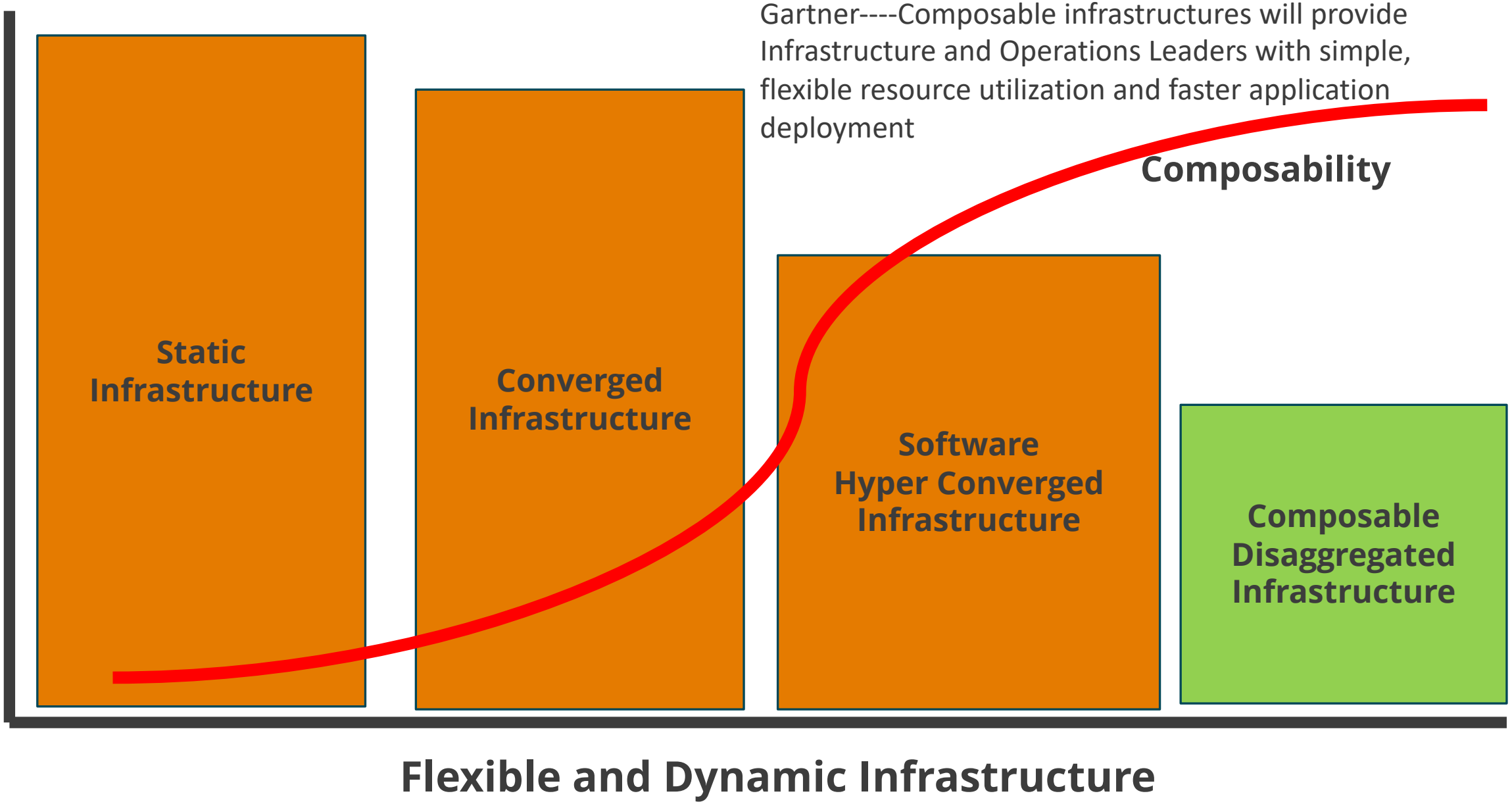
Composable HPC and Enterprise Computing Systems:

- Enable efficient usage of available hardware resources by provisioning it where it is needed
- Mitigate the need for hardware overprovisioning
- Reduce electricity consumption and cooling costs
  - 4% of the World's Energy Consumption in input into datacenters (https://www.energy.gov/eere/buildings/data-centers-and-servers)

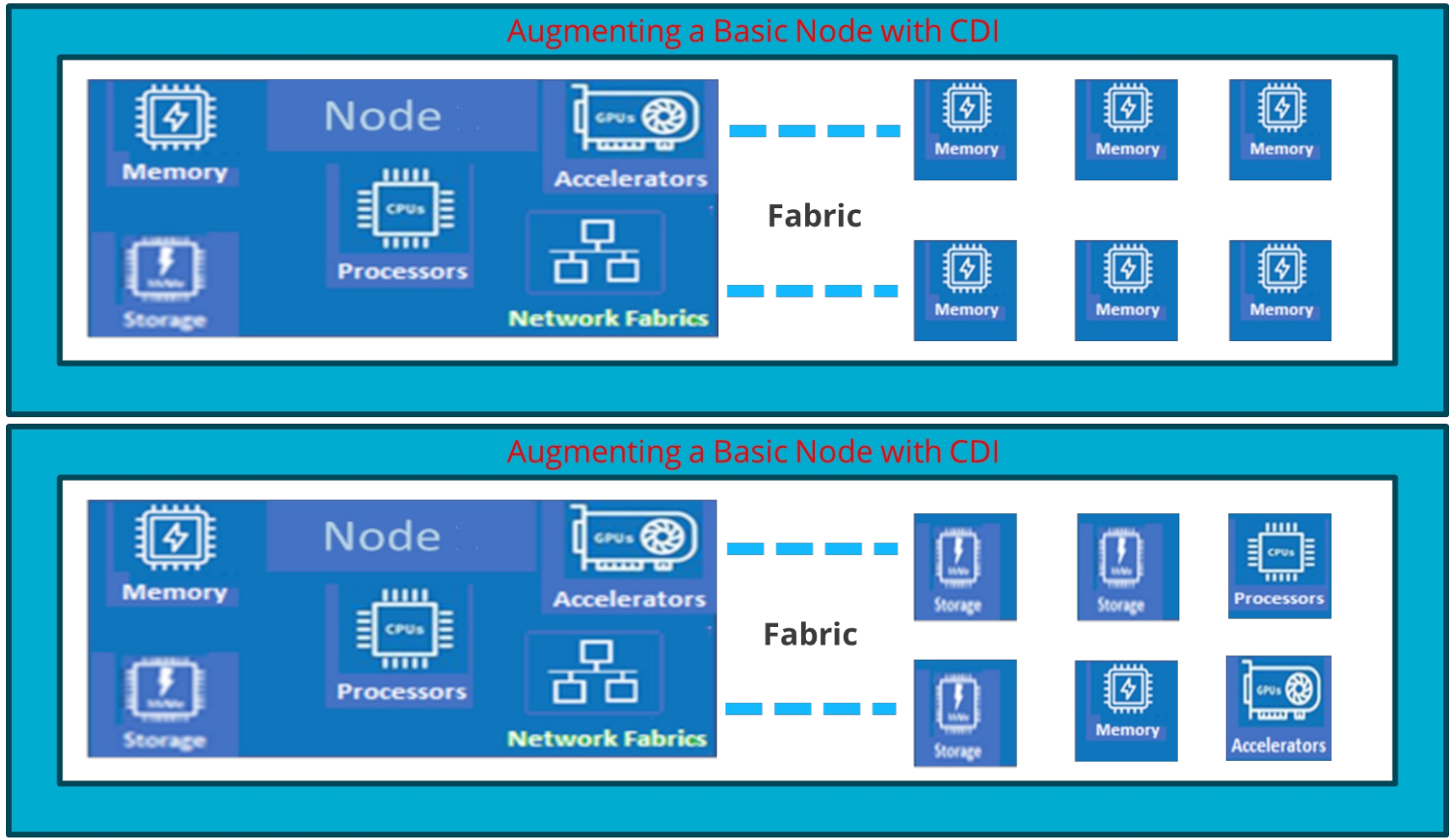


Quick overview of Composable Disaggregated Infrastructure

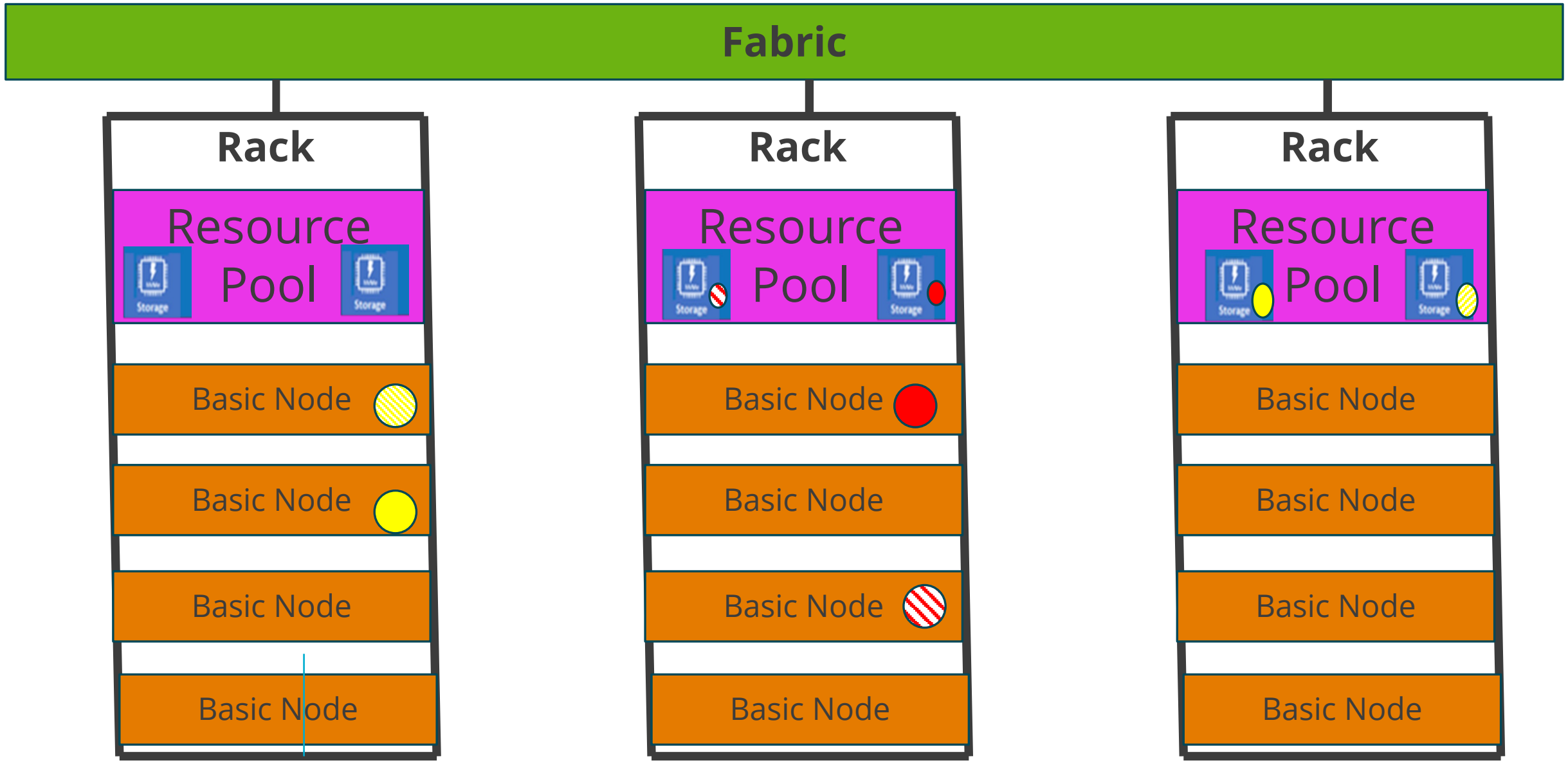
Gartner----Composable infrastructures will provide Infrastructure and Operations Leaders with simple, flexible resource utilization and faster application deployment



Quick overview of Composable Disaggregated Infrastructure

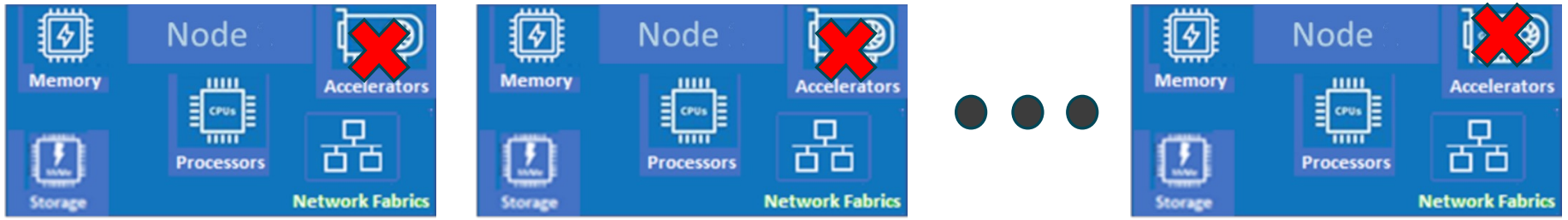


Quick overview of Composable Disaggregated Infrastructure

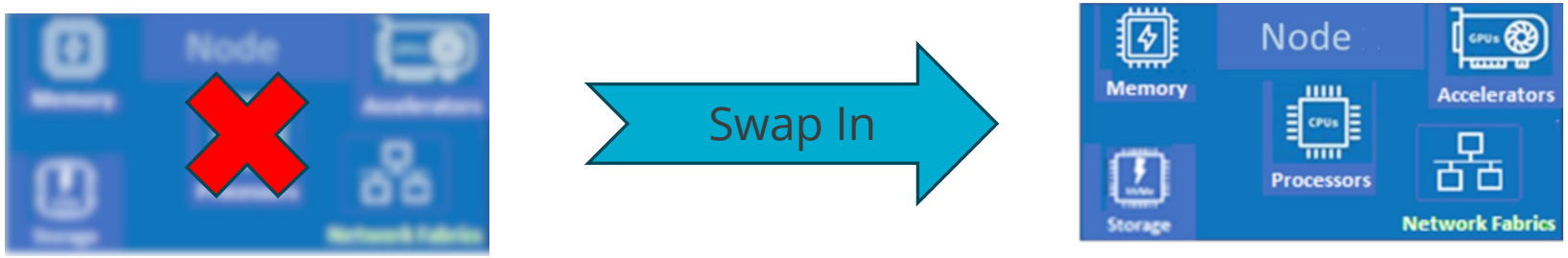


Design Considerations for a Composability Manager

- The larger the HPC system, the greater the potential impact of:
  - Stranded Resources

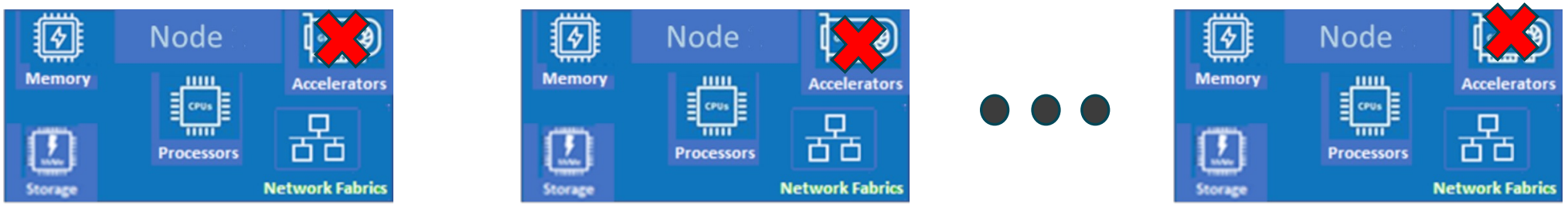


  - Computational Stability



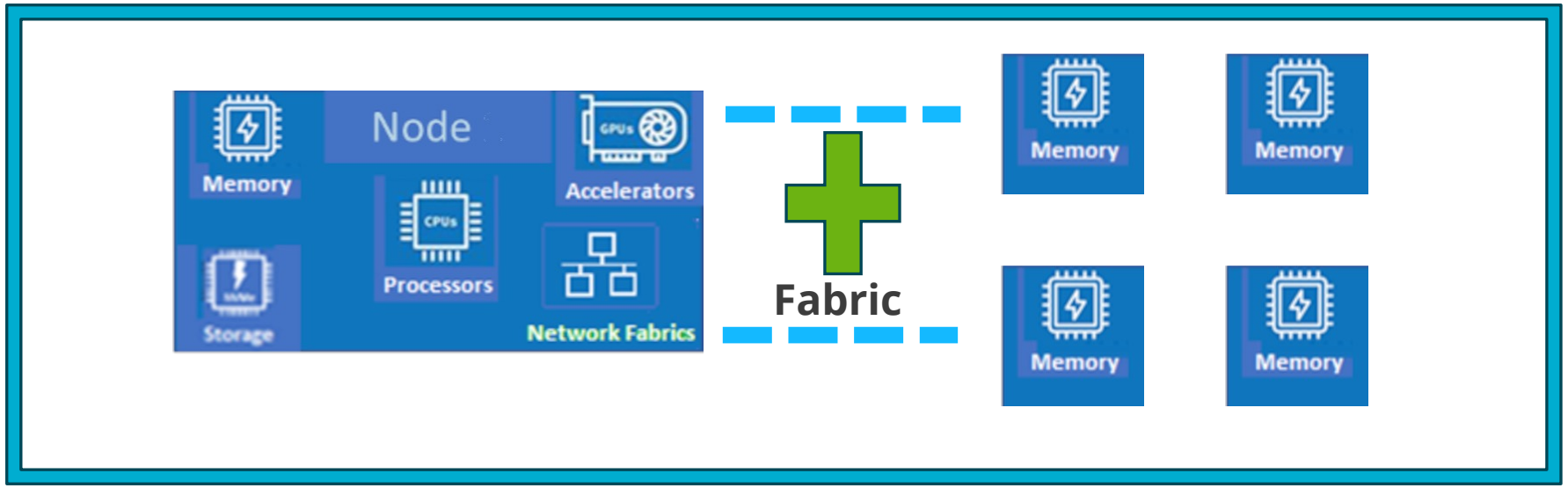
Limitations of current HPC System Architectures

- The larger the HPC system, the greater the potential impact of:
  - Resource limits are fixed. So, we have to ‘steal’ memory resources from our compute nodes to run our Burst-Buffer.
  - Stranded Resources that are using up energy and generating heat
  - Increased monetary resources to build out components to address all possible types of Application Codes that the HPC must support.
  - Hardware failures during the run-time can kill running applications.



Versatility and capabilities for a CDI infrastructure and a Composable Burst-Buffer filesystem What we can do with such a set-up.

Augment the node with memory. We have the memory that we are going to use for our BeeOND Parallel Filesystem.

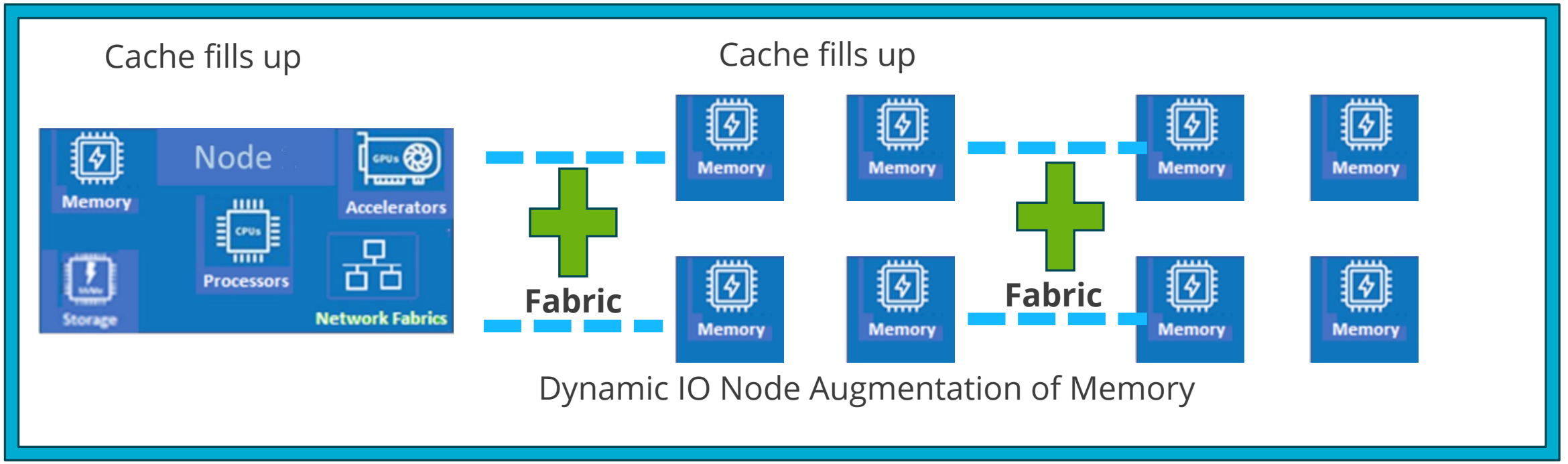


Versatility and capabilities for a CDI infrastructure and a Composable Burst-Buffer filesystem
What we can do with such a set-up.

If we need more IO servers to mitigate load issues, we can compose additional servers and automatically add them into the storage pool. The new OST storage just shows up in the pool.

Versatility and capabilities for a CDI infrastructure and a Composable Burst-Buffer filesystem What we can do with such a set-up.

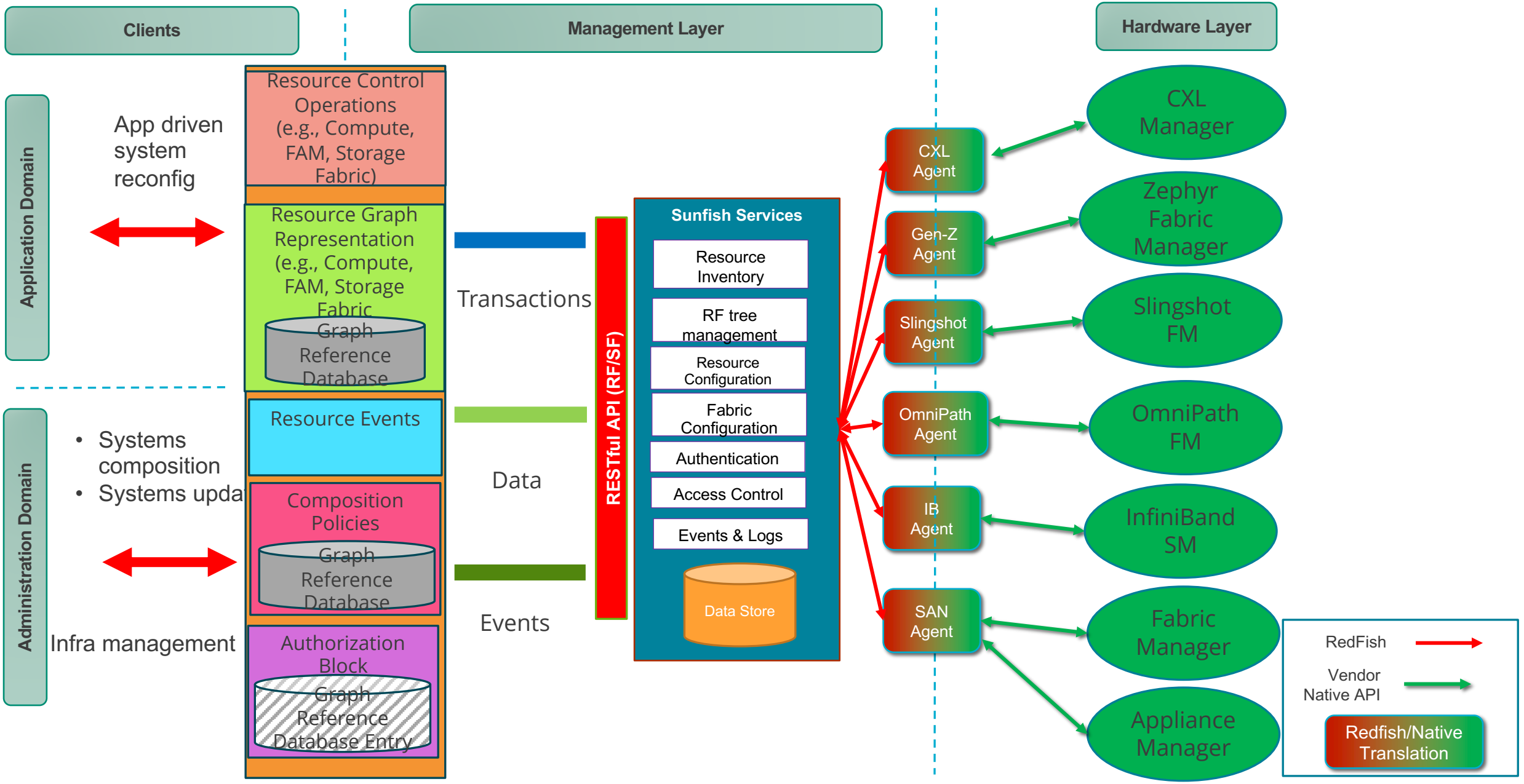
Mitigate IO Cache Thrashing by dynamically adding memory to the IO node
(eg. ZFS Adaptive Replacement Cache, XFS Buffer Cache, etc.) to prevent Virtual Memory Swaps from disk.



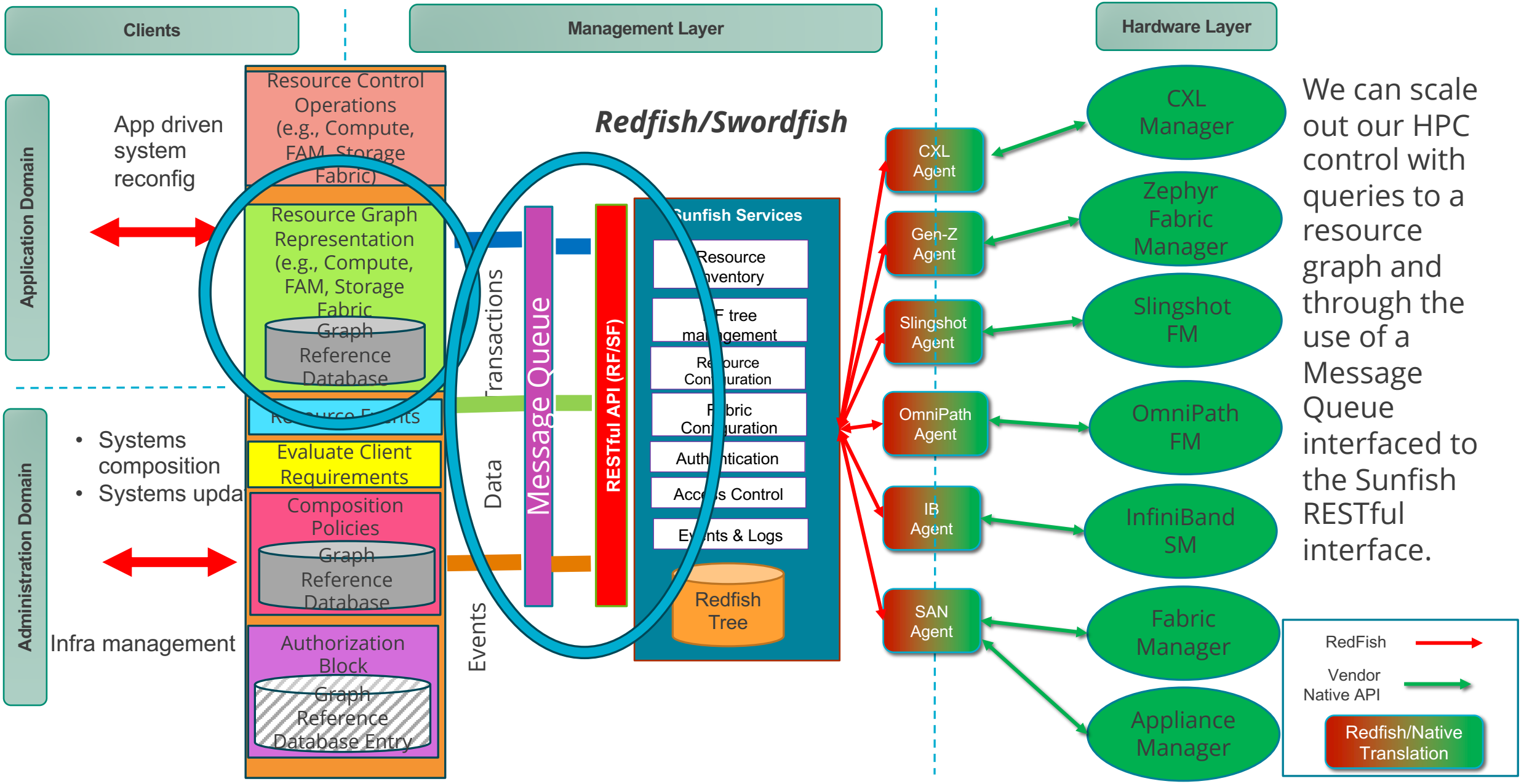
Design Considerations for a Composability Manager

Scaling the control structure To very large HPC systems

Introducing Sunfish

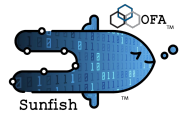


Sunfish

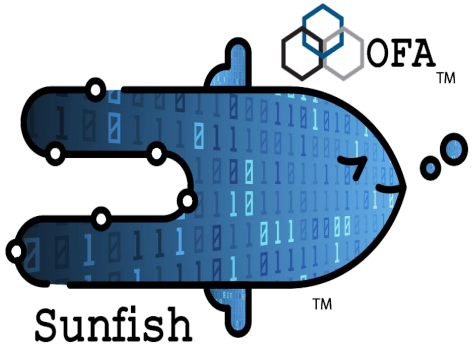


We can scale out our HPC control with queries to a resource graph and through the use of a Message Queue interfaced to the Sunfish RESTful interface.

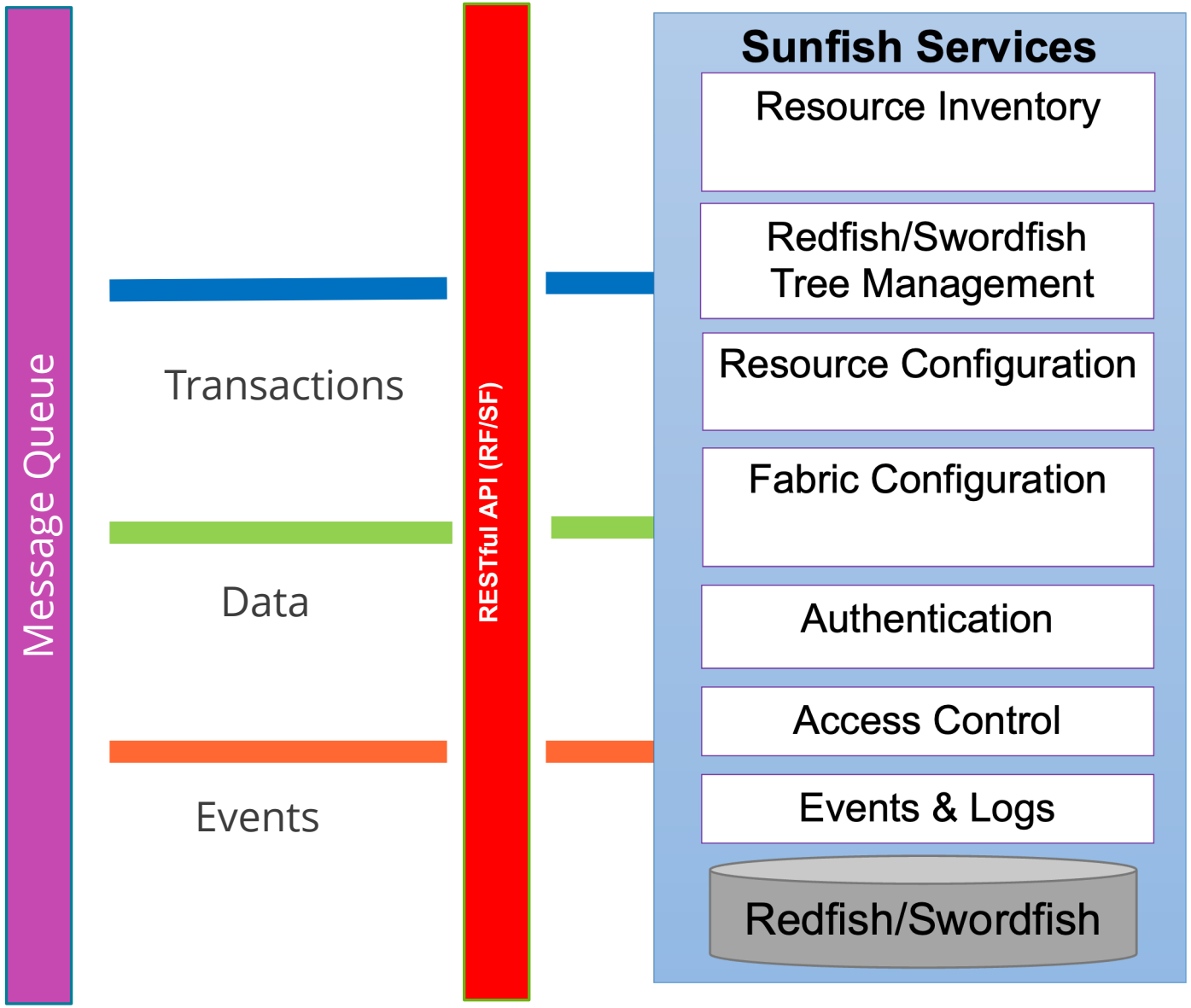
Sunfish Core Services



- Redfish/Swordfish Tree
- RESTful Interface
  - Supports message queues such as RabbitMQ or Apache Kafka for scaling
- Built-In:
  - Authentication
  - Aggregation Support for Components
  - Event Communications and Subscriptions



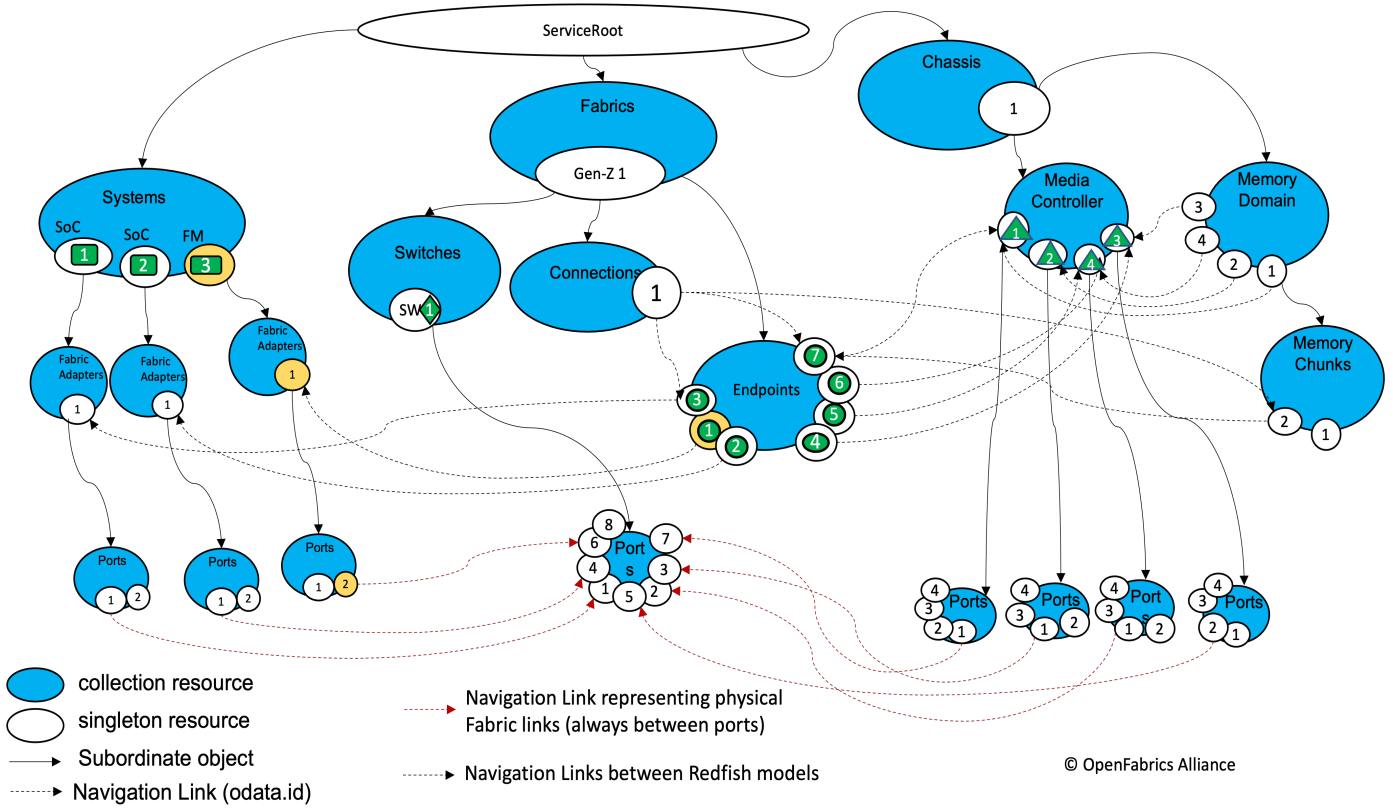
Sunfish Core Services



Sunfish Core Services

Redfish Representation of a Composable Disaggregated Infrastructure
Redfish mapping of a simple HPC system

Simple Gen-Z Linux System Redfish Tree: Physical Objects, Endpoints, and Port linkages

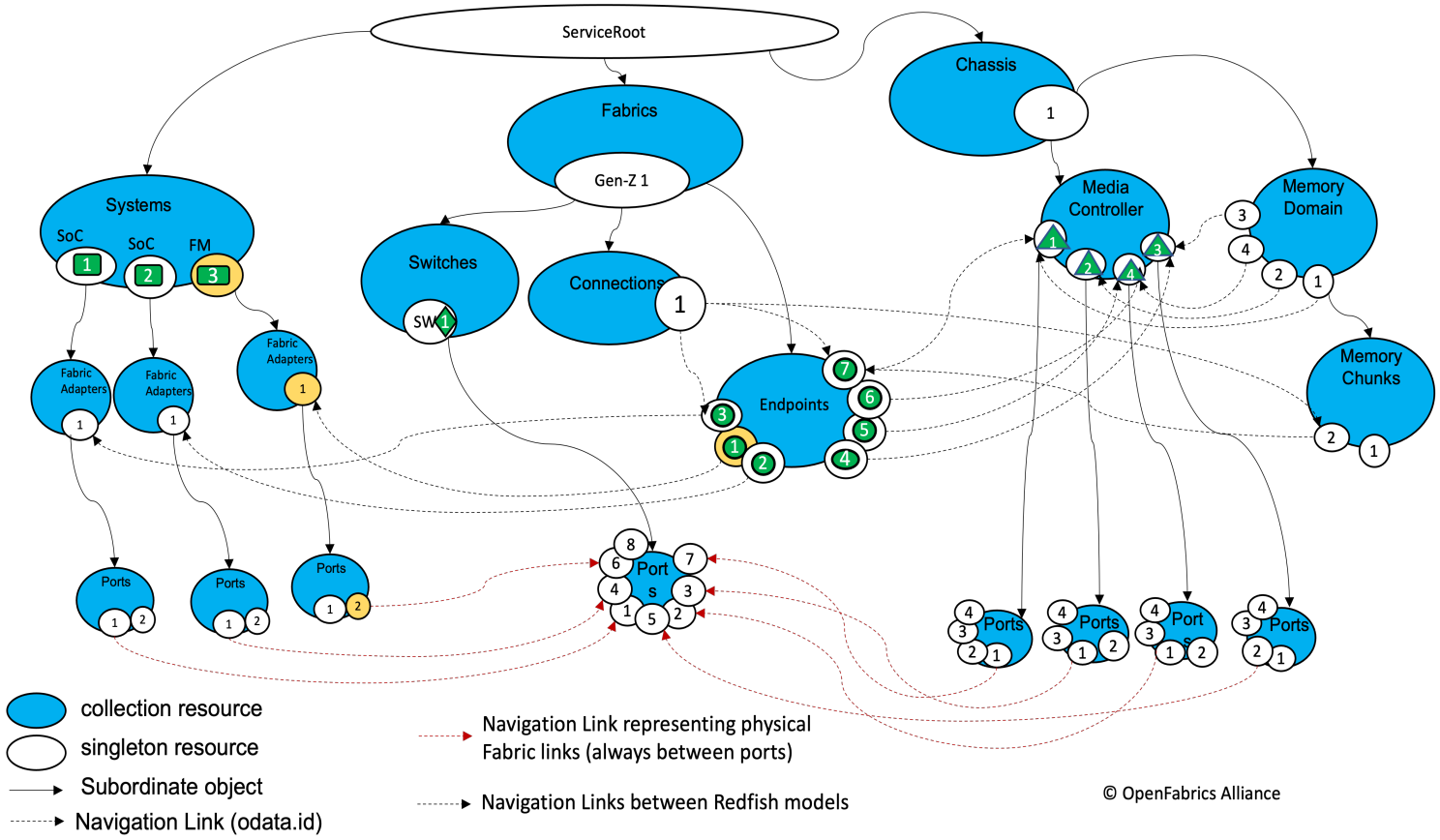



```
curl -X GET -H "Content-Type: application/json"
http://ofmfserv:5000/redfish/v1/Fabrics/NVMeoF/Endpoints/Initiator
1
{
    "@odata.type": "#Endpoint.v1_7_0.Endpoint",
    "Id": "Initiator1",
    "Name": "NVMe-oF Initiator (Host)",
    "EndpointProtocol": "NVMeOverFabrics",
    "Identifiers": [
        {
            "DurableName": "host.corp.com:nvme:nvm-subsys-sn-
4635",
            "DurableNameFormat": "NQN"
        }
    ],
    "ConnectedEntities": [
        {
            "EntityType": "NetworkController",
            "EntityRole": "Initiator"
        }
    ],
    "IPTransportDetails": [
        {
            "TransportProtocol": "Ethernet",
            "IPv4Address": {
                "Address": "10.3.5.205"
            },
            "Port": 13244
        }
    ],
    "Links": {
        "Connections": [
            {
                "@odata.id":
"/redfish/v1/Fabrics/NVMeoF/Connections/1"
            }
```

Sunfish Core Services

Redfish Representation of a Composable Disaggregated Infrastructure
Redfish mapping of a simple HPC system

Simple Gen-Z Linux System Redfish Tree: Physical Objects, Endpoints, and Port linkages

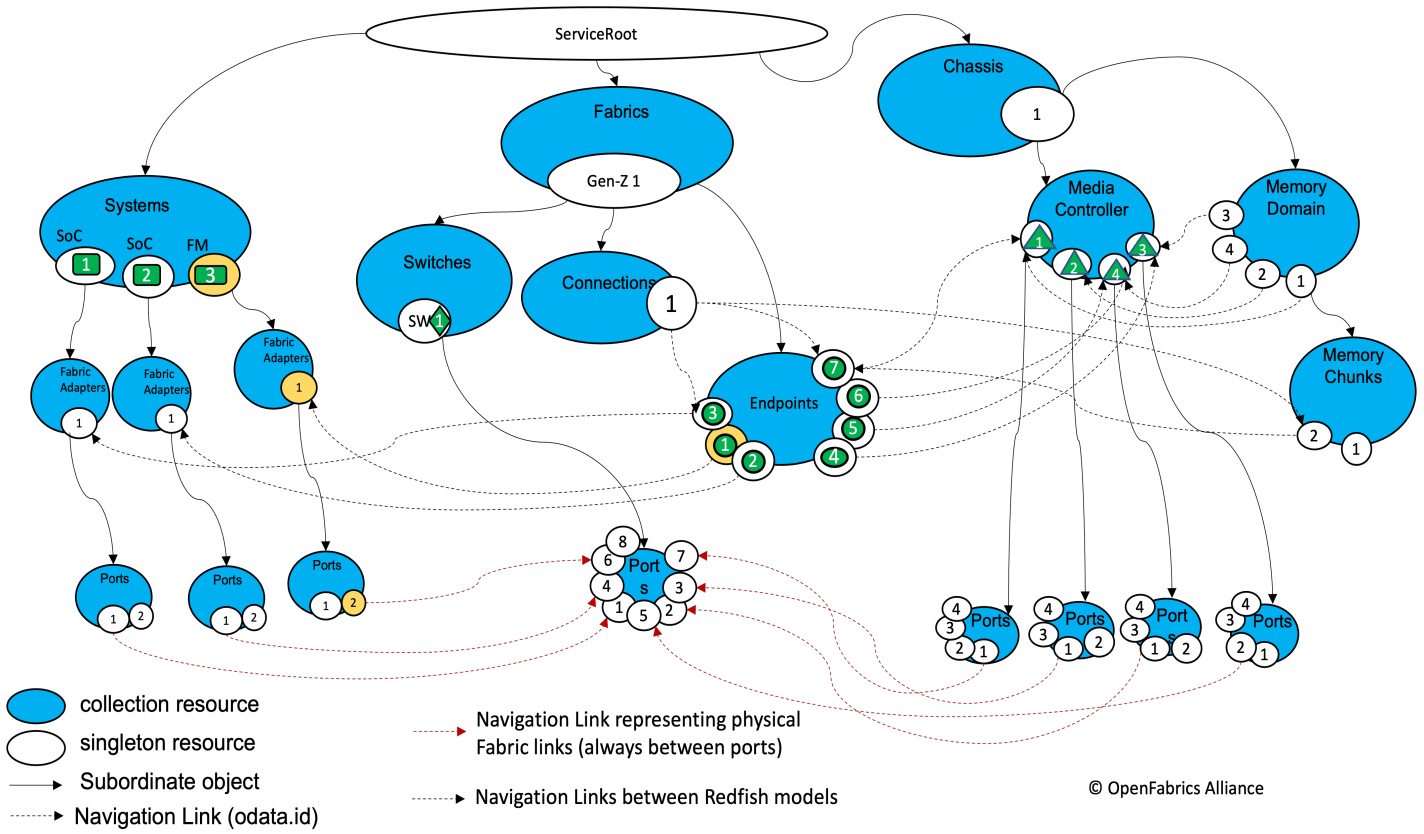



```
curl -X GET -H "Content-Type: application/json"
http://ofmfserv:5000/redfish/v1/Fabrics/NVMeoF/Connec
tions/1
{
    "@odata.type": "#Connection.v1_0_0.Connection",
    "@Redfish.ReleaseStatus": "WorkInProgress",
    "Id": "1",
    "Name": "Host Connection 1",
    "Description": "Connection info for host 1",
    "ConnectionType": "Storage",
    "VolumeInfo": [
        {
            "AccessCapabilities": [
                "Read",
                "Write"
            ],
            "Volume": {
                "@odata.id":
"/redfish/v1/Storage/IPAttachedDrive1/Volumes/SimpleN
amespace"
            }
        },
        {
            "AccessCapabilities": [
                "Read",
                "Write"
            ],
            "Volume": {
                "@odata.id":
"/redfish/v1/Fabrics/NVMeoF/Connections/1"
```

Sunfish Core Services

Redfish Representation of a Composable Disaggregated Infrastructure
Redfish mapping of a simple HPC system

Simple Gen-Z Linux System Redfish Tree: Physical Objects, Endpoints, and Port linkages

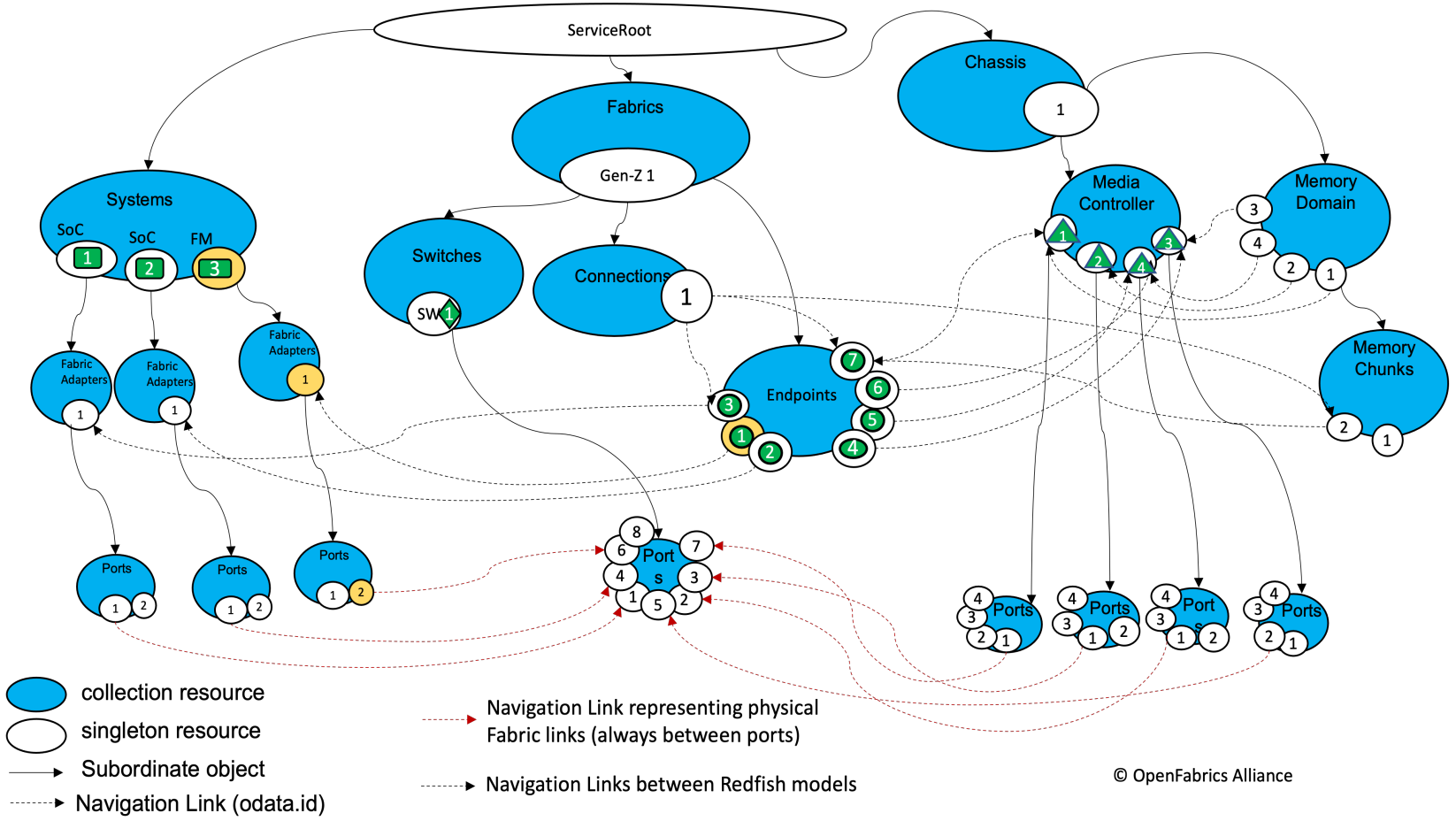


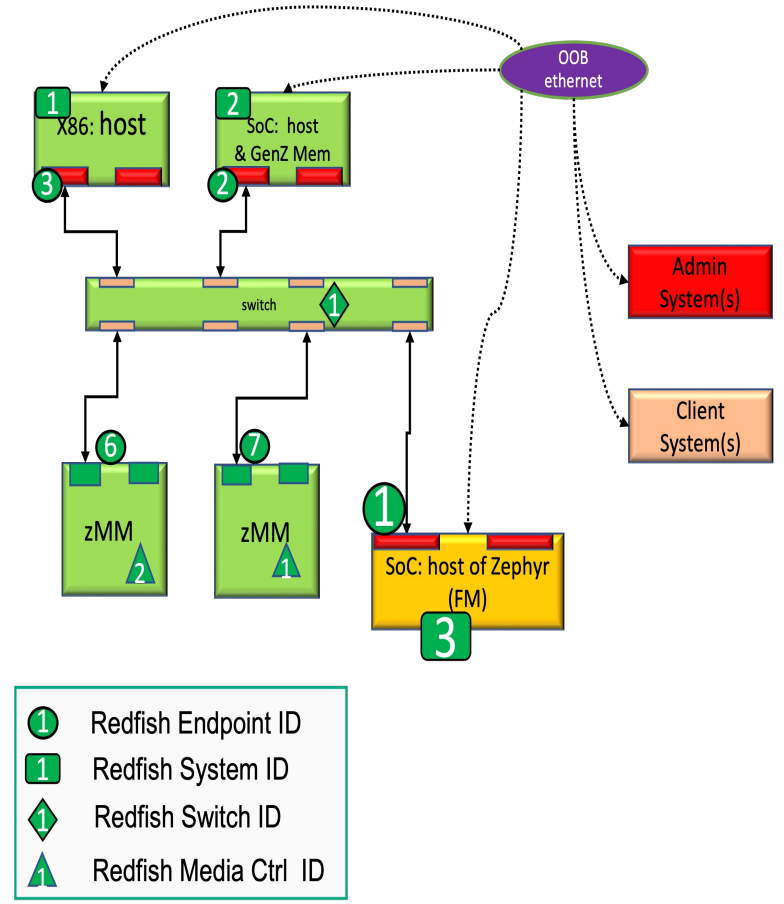

```
$> curl -X GET -H "Content-Type: application/json"
hthp://ofmfserv:5000/redfish/v1/Fabrics
{
    "@odata.type": "#FabricCollection.FabricCollection",
    "Name": "Fabric Collection",
    "Members@odata.count": 2,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Fabrics/NVMeoF"
        },
        {
            "@odata.id": "/redfish/v1/Fabrics/Ethernet"
        }
    ],
    "@odata.id": "/redfish/v1/Fabrics"
}(Swordfish)

curl -X POST -H "Content-Type: application/json" -d
@fabric_connection.json  http://ofmfserv:5000/redfish/v1/Fab
rics/CXL
{
    "@odata.id": "/redfish/v1/Fabrics/CXL",
    "@odata.type": "#Fabric.v1_3_CXL.Fabric",
    "Id": "CXL",
    "Name": "Fabric"
}
```

Sunfish Hardware Agents

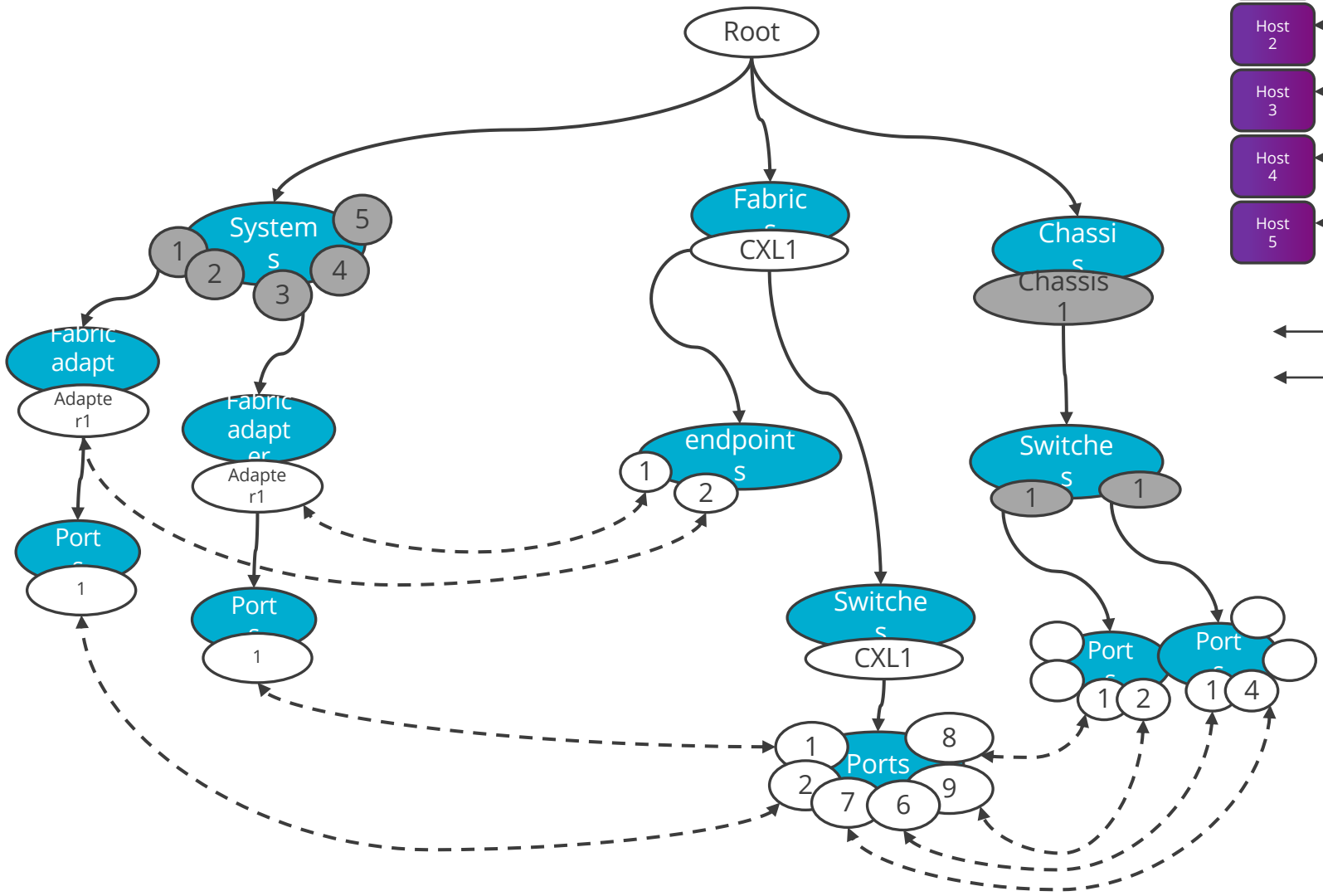
Redfish Representation of a Composable Disaggregated Infrastructure
Redfish mapping of a simple HPC system from Hardware to Redfish for the Sunfish Core

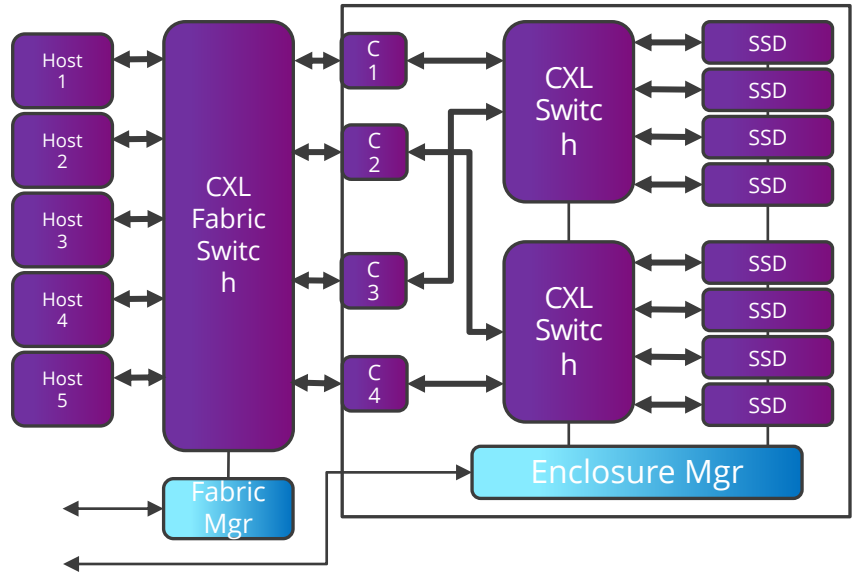
Simple Gen-Z Linux System Redfish Tree: Physical Objects, Endpoints, and Port linkages



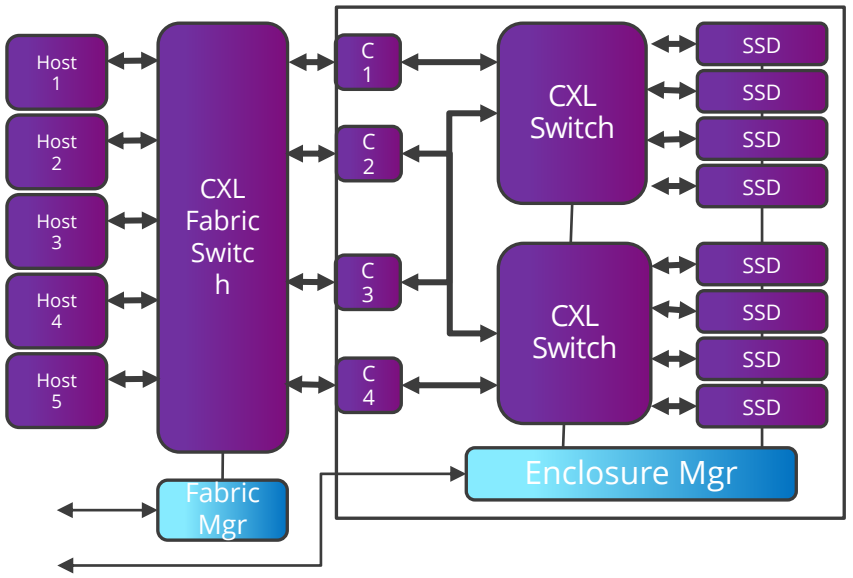


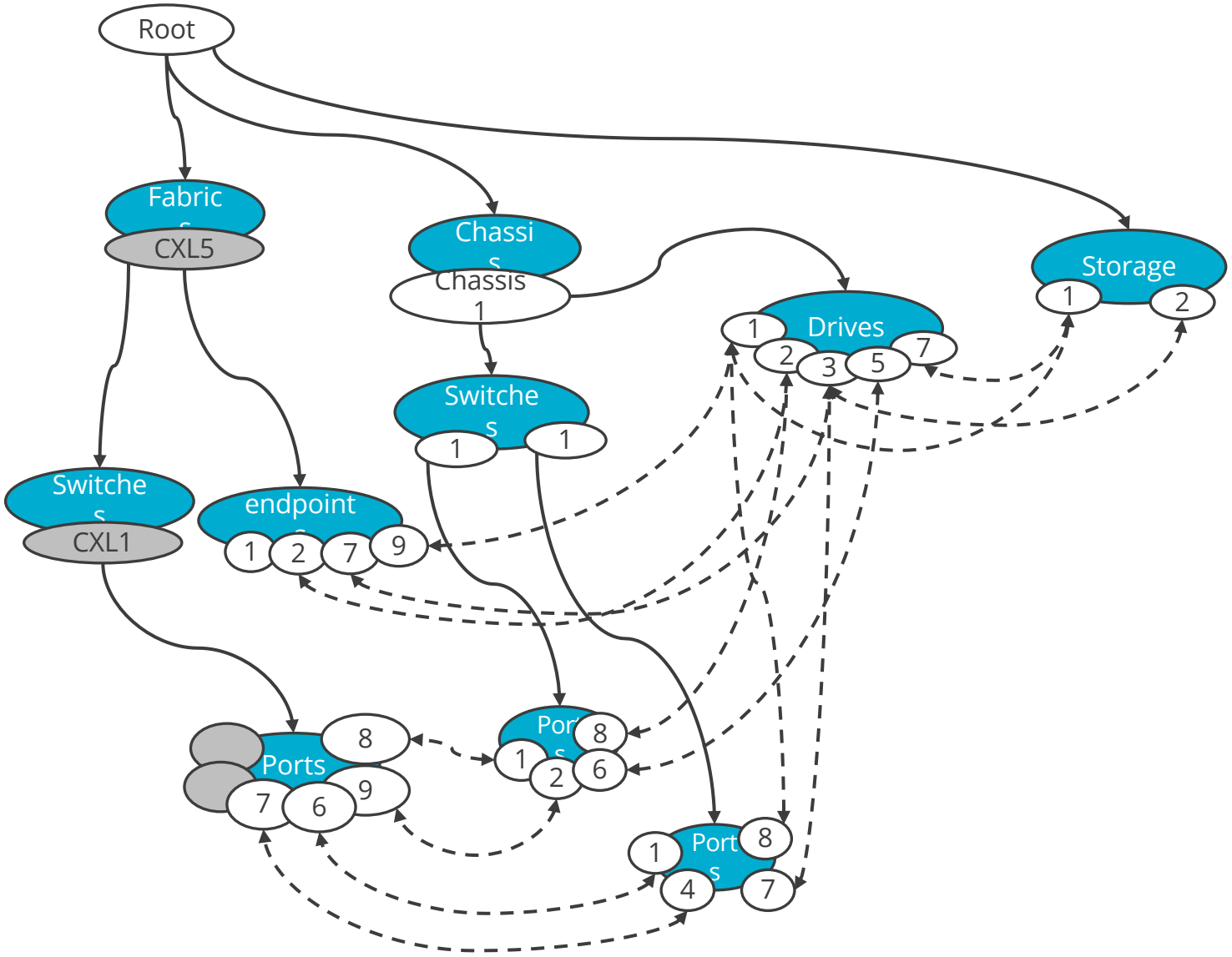
CXL Fabric Manager View



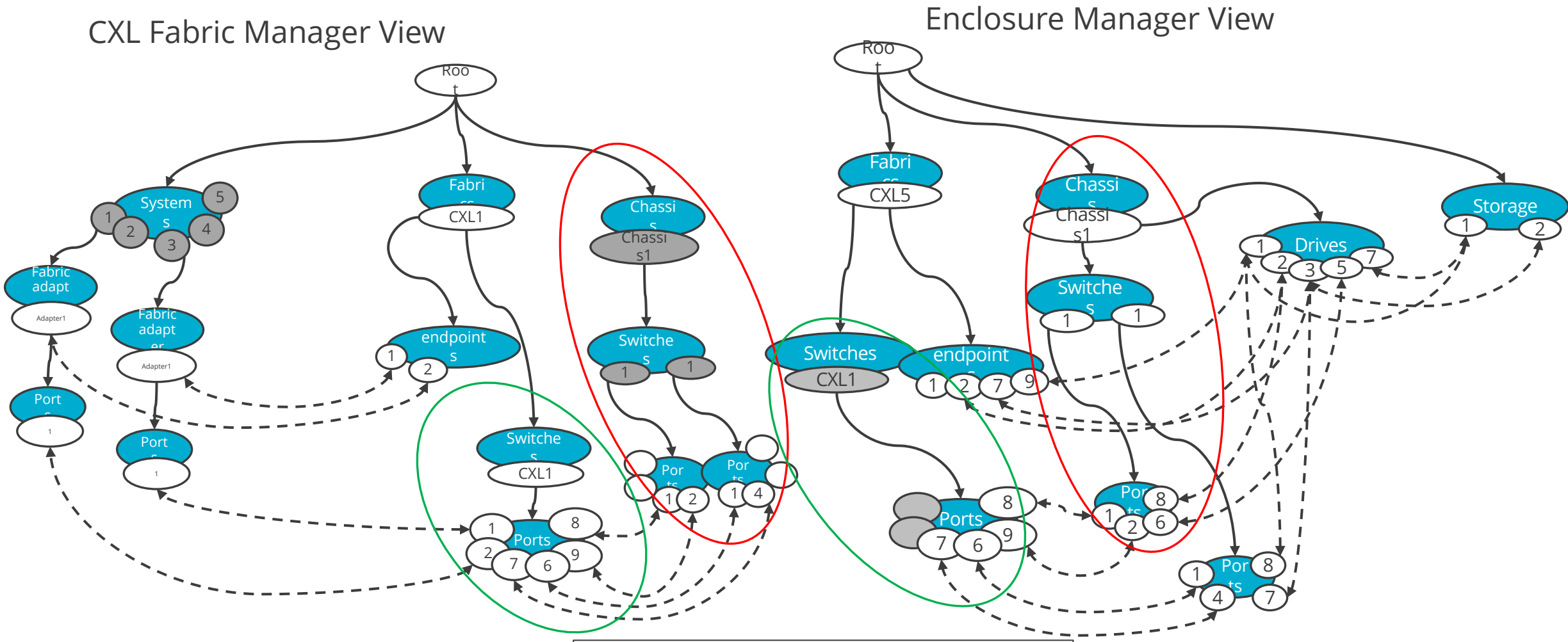


Enclosure Manager View



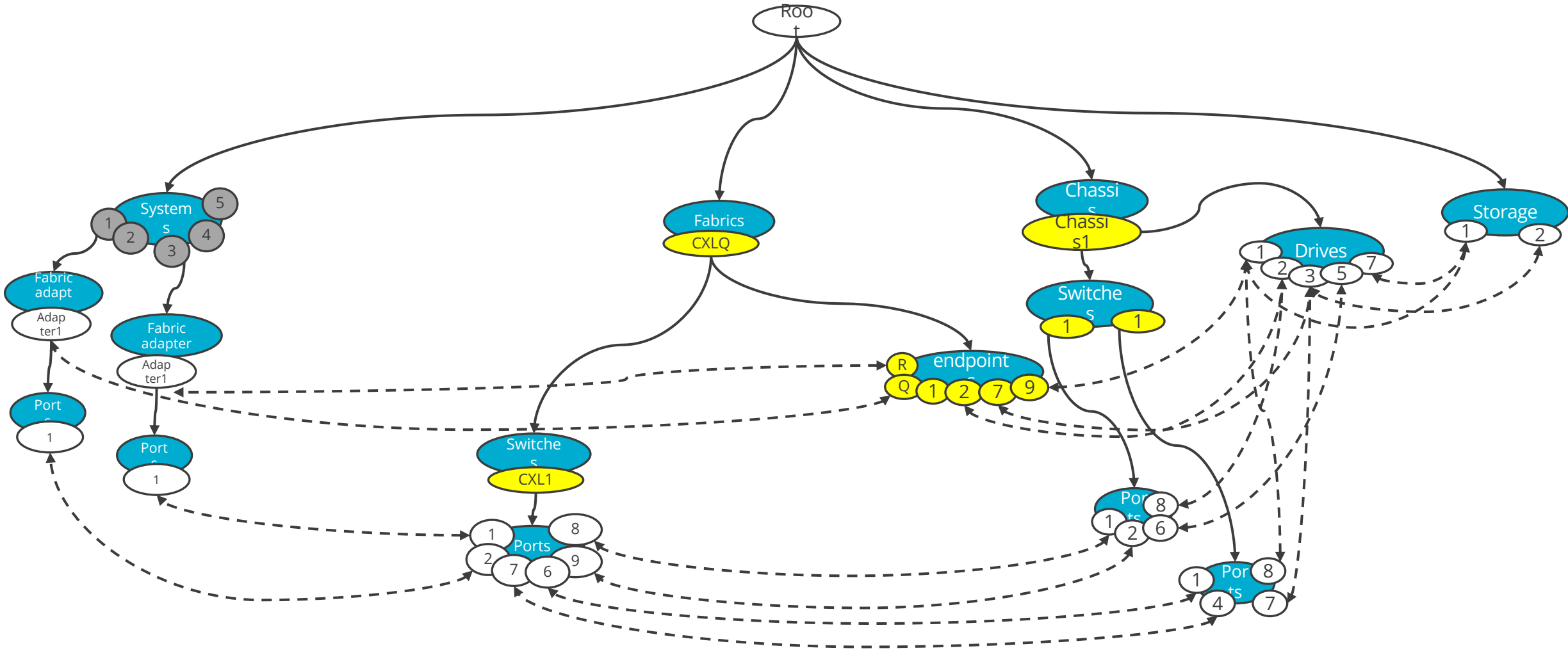


The Two Views Need To Be Merged

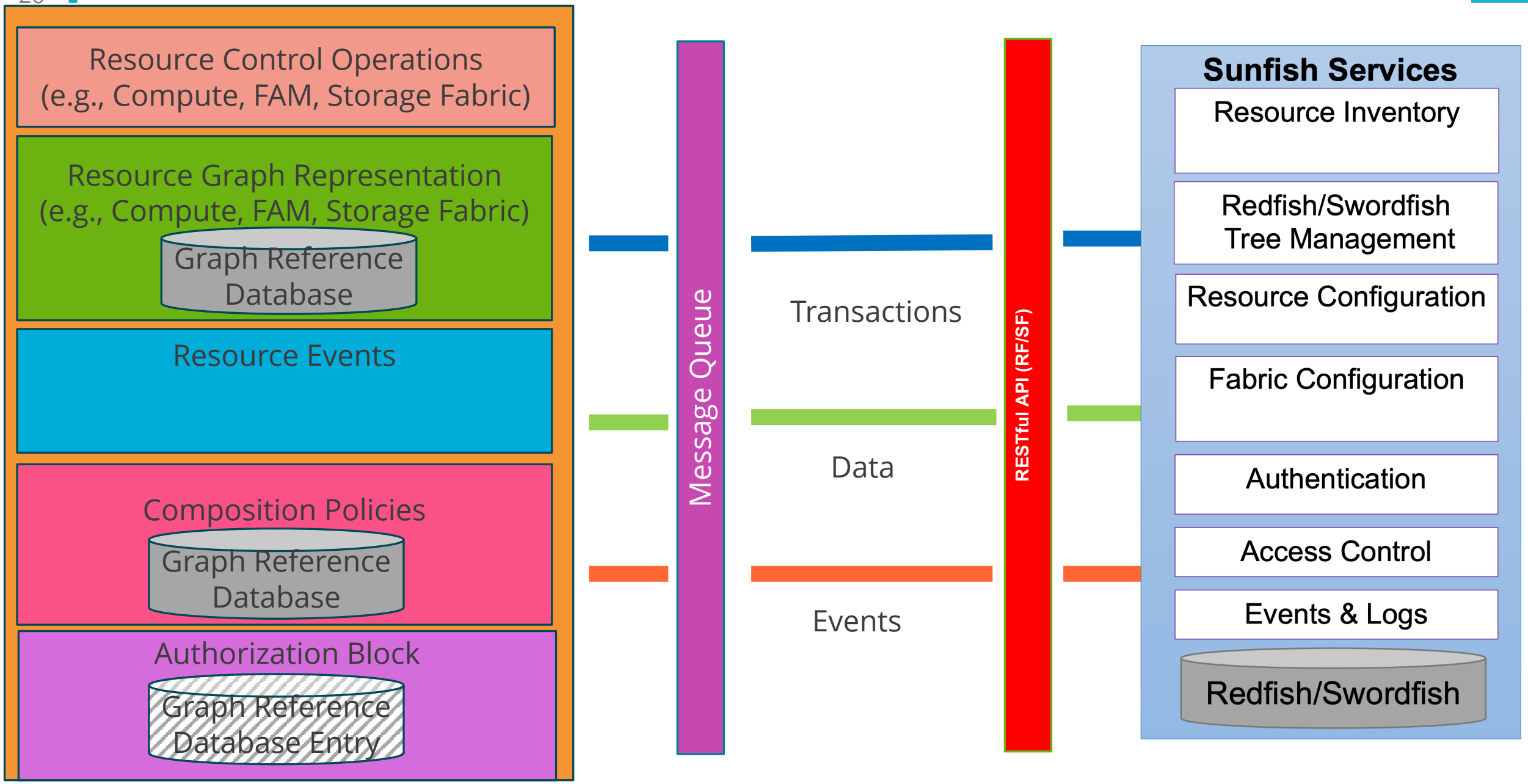


Different fabrics may require different methods to detect boundary links and resolve boundary

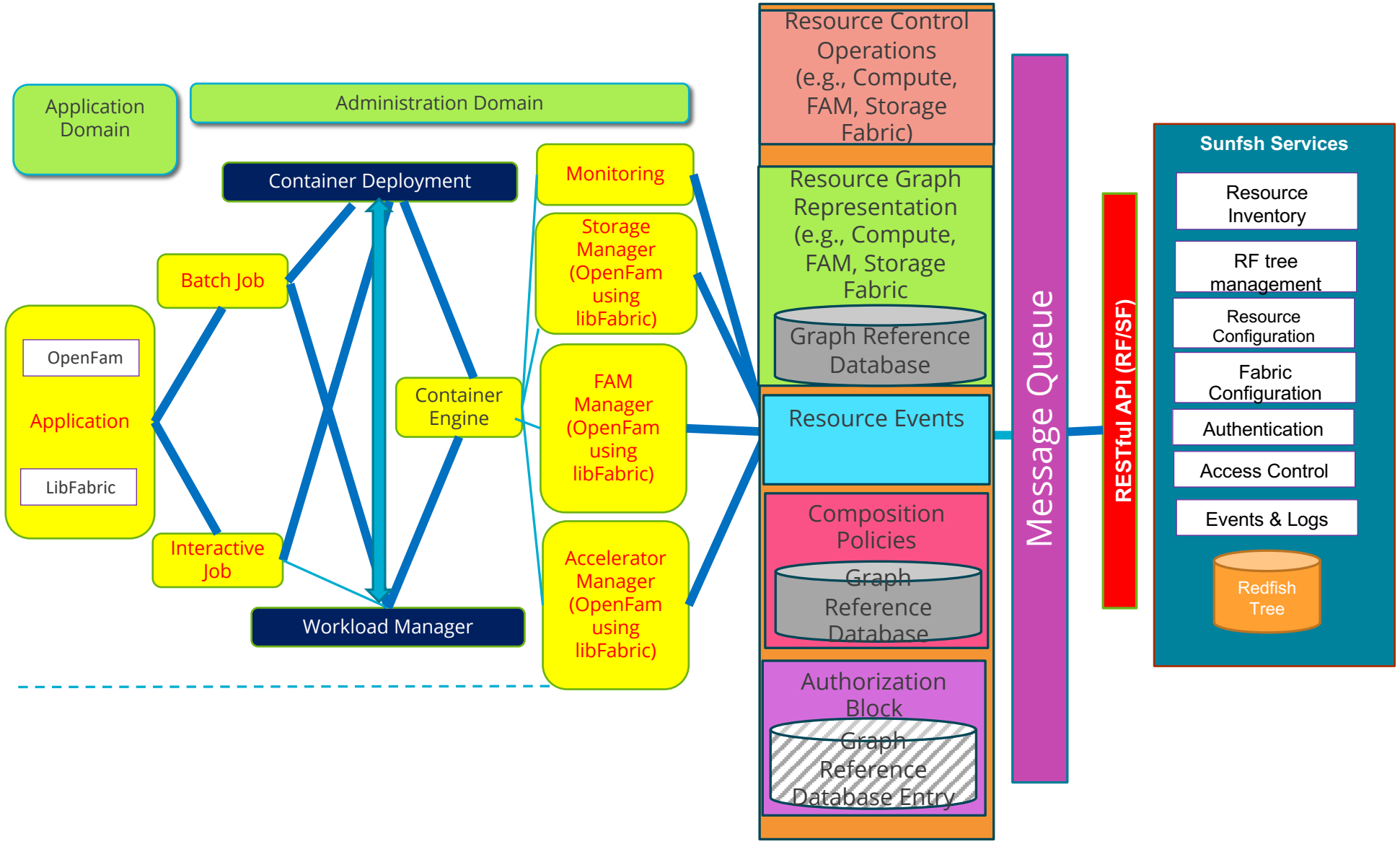
The Two Views After Merging by Sunfish



Sunfish Composability Management Framework



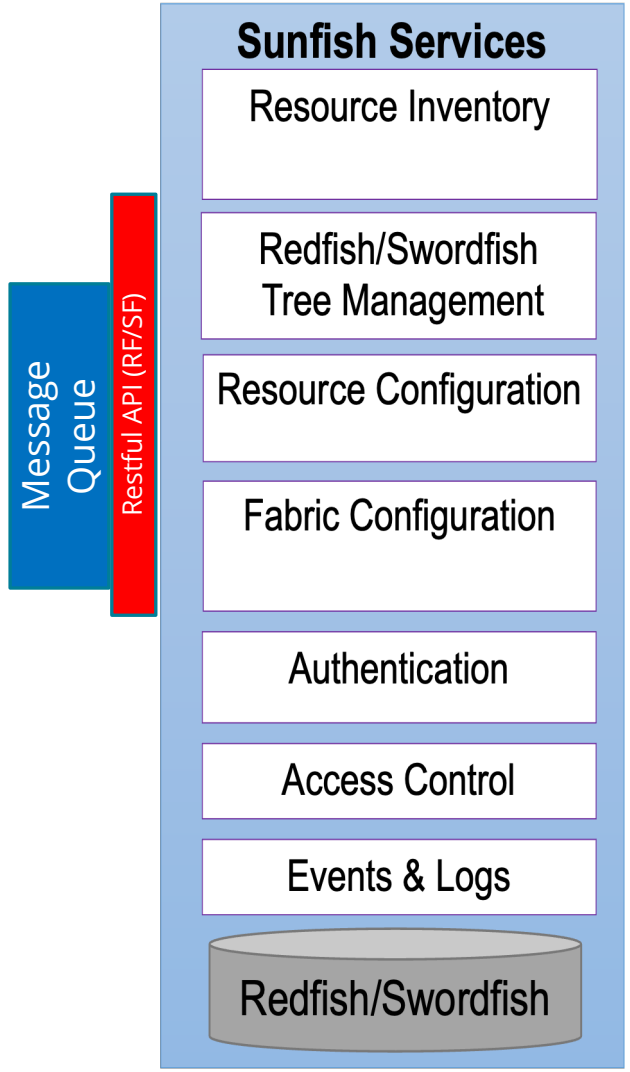
Sunfish Composability Management Framework



Sunfish Composability Management Framework

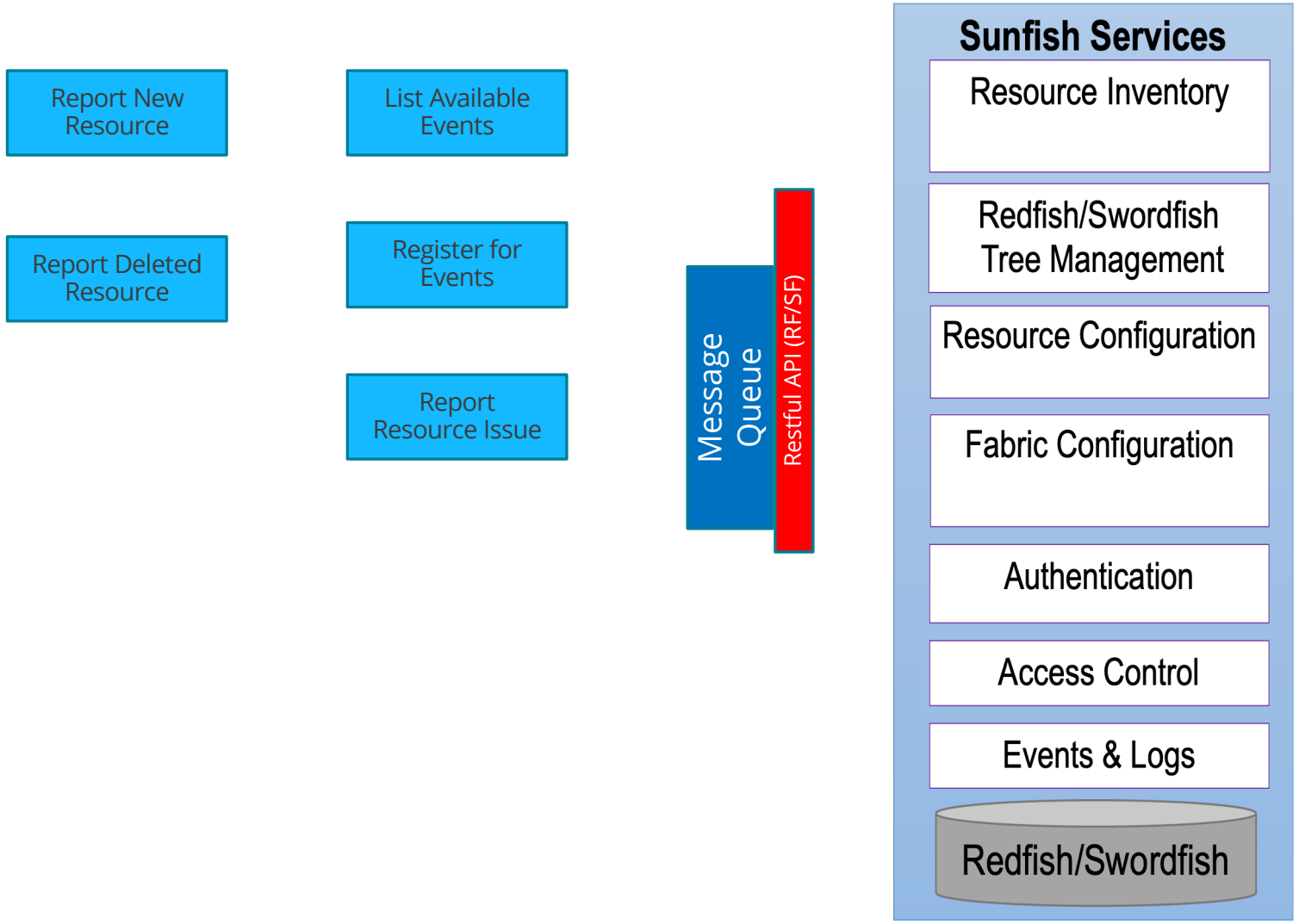
Resource Control Operations

- Register Composer
- Client Request Fulfilled
- Query Resource Features
- Expand a Compute Node
- Show Aggregated Resource
- Gather Resource Security Information
- Query Resource Connections
- Create Resource Namespace
- Delete Resource Namespace
- Query Inventory
- Modify Resource Deta
- Request Resource QoS
- Query Resource Path
- Create System
- Reserve Inventory
- Resource Authorization Modification
- Gather Resource Failover
- Build Resource Path
- Delete System
- Assemble System
- Add Resource to Database
- Release Resource Failover Option
- Modify Resource Path
- Show System
- Disaggregate a Compute Node
- Get Request from Client
- Choose best Resource Failover Options
- Delete Resource Path
- Modify System



Sunfish Composability Management Framework

Resource Events



Sunfish Composability Management Framework

Evaluate and Meet Client Requirements

- Get Server Resource Information
- Correlate Servers with a Reference Variable
- Create Server Category Reference

For Best Server option

- Server Nearest Neighbor Estimate
- Get Server Physical Location
- Availability of Network Link

For best Connection option

- Resource Pool Nearest Neighbor Estimate
- Provide Resource Pool Information
- Ger Resource Pool Physical Location
- Ger Indiividual Resource Characteristics

For best Resource option

- Gather Client Request Frequency of Occurrence
- Define Client Request Expected Statistical Requirement Ordering
- Grab Client Resource Request Numerical Range of Data
- Grab Client Request Similar Resource Items in Transaction Data
- Grab Client Resource Request Transaction Trends
- Create a Decision Tree for Resource Selection
- Resources are missing from Server to Meet Client Request?
- Subspace FINDIT clustering of Resources

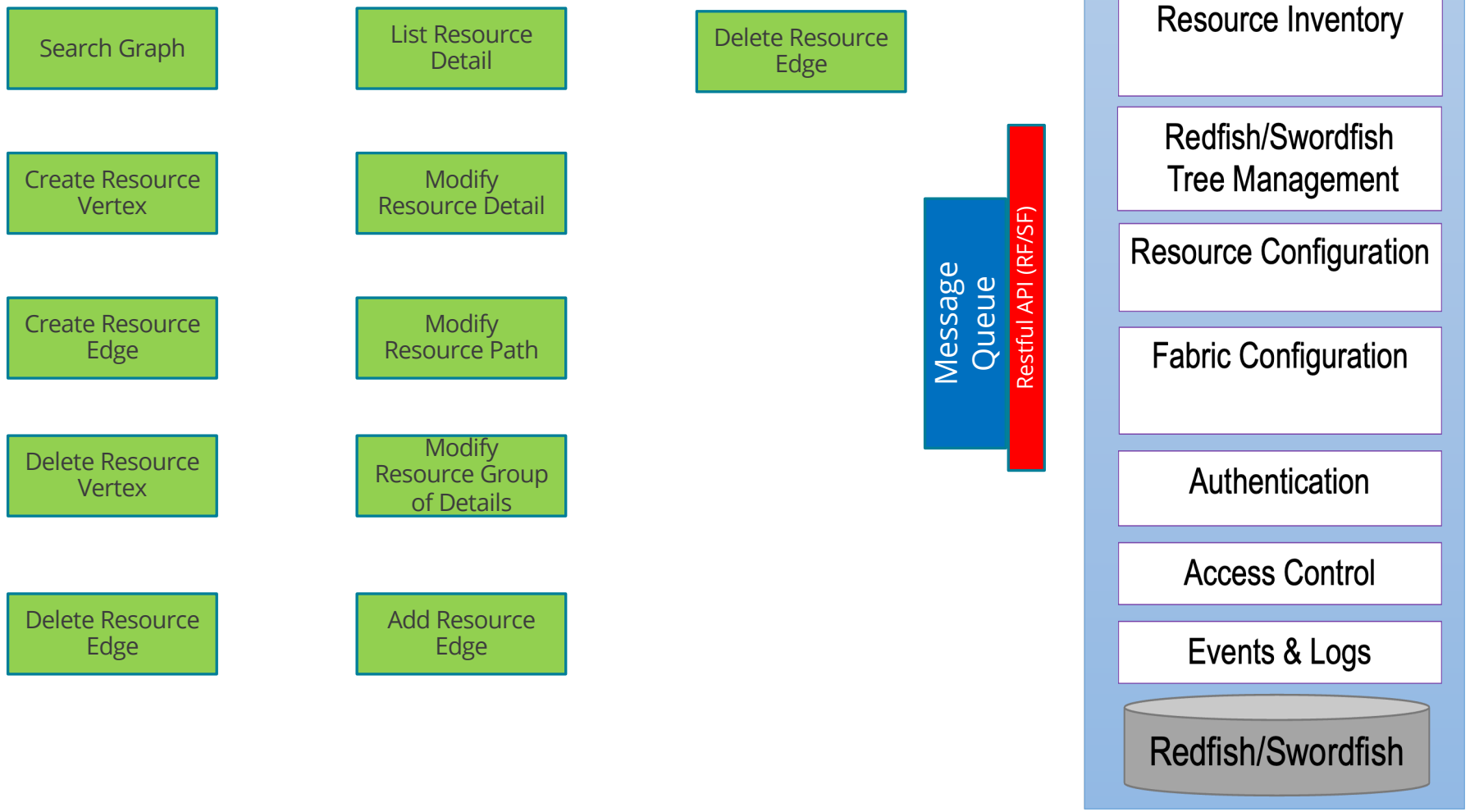
Message Queue

Restful API (RF/SF)

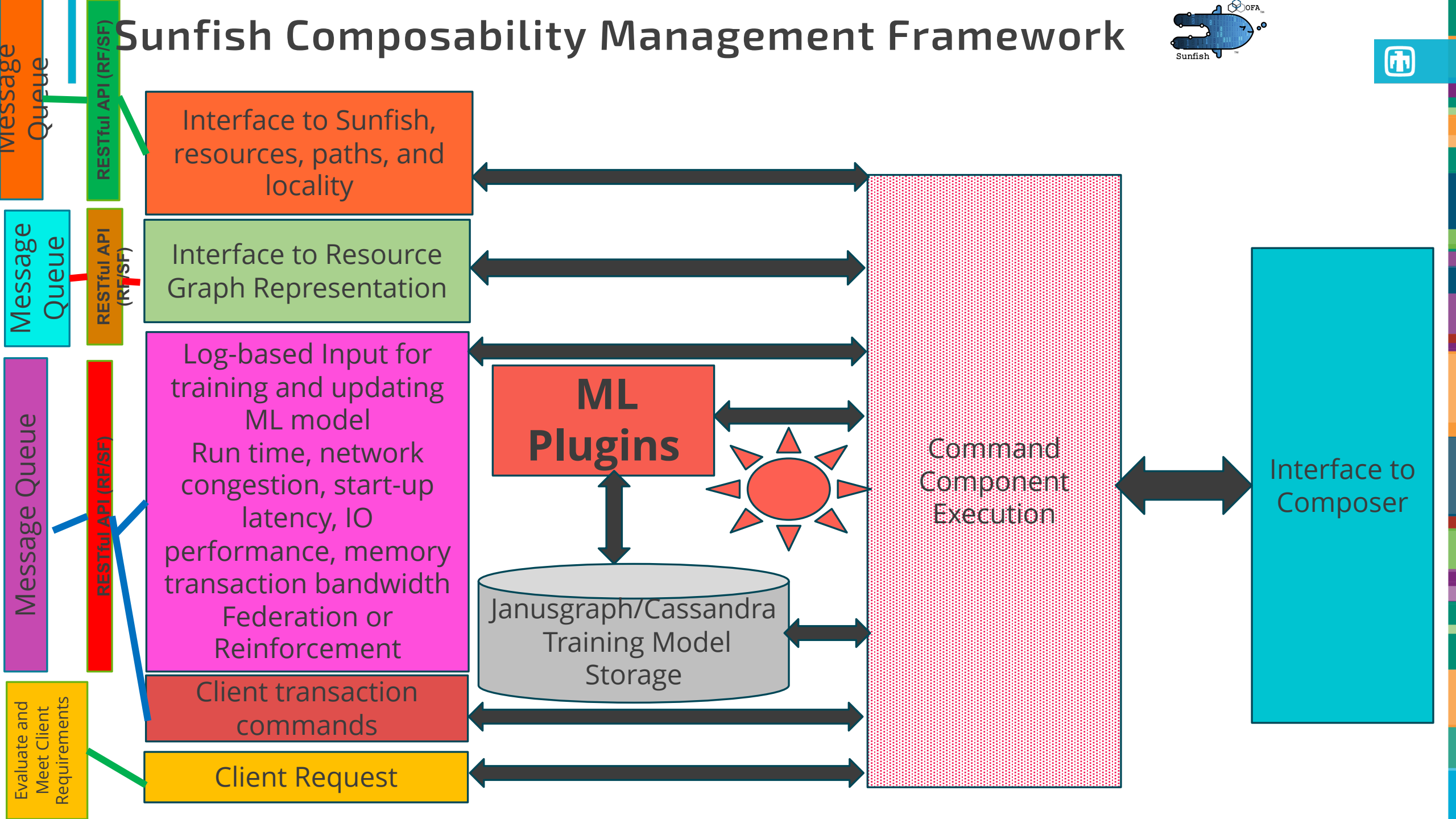
Sunfish Services

- Resource Inventory
- Redfish/Swordfish Tree Management
- Resource Configuration
- Fabric Configuration
- Authentication
- Access Control
- Events & Logs
- Redfish/Swordfish

Sunfish Composability Management Framework



Sunfish Composability Management Framework



Reference Framework for Composition Decisions and Policies

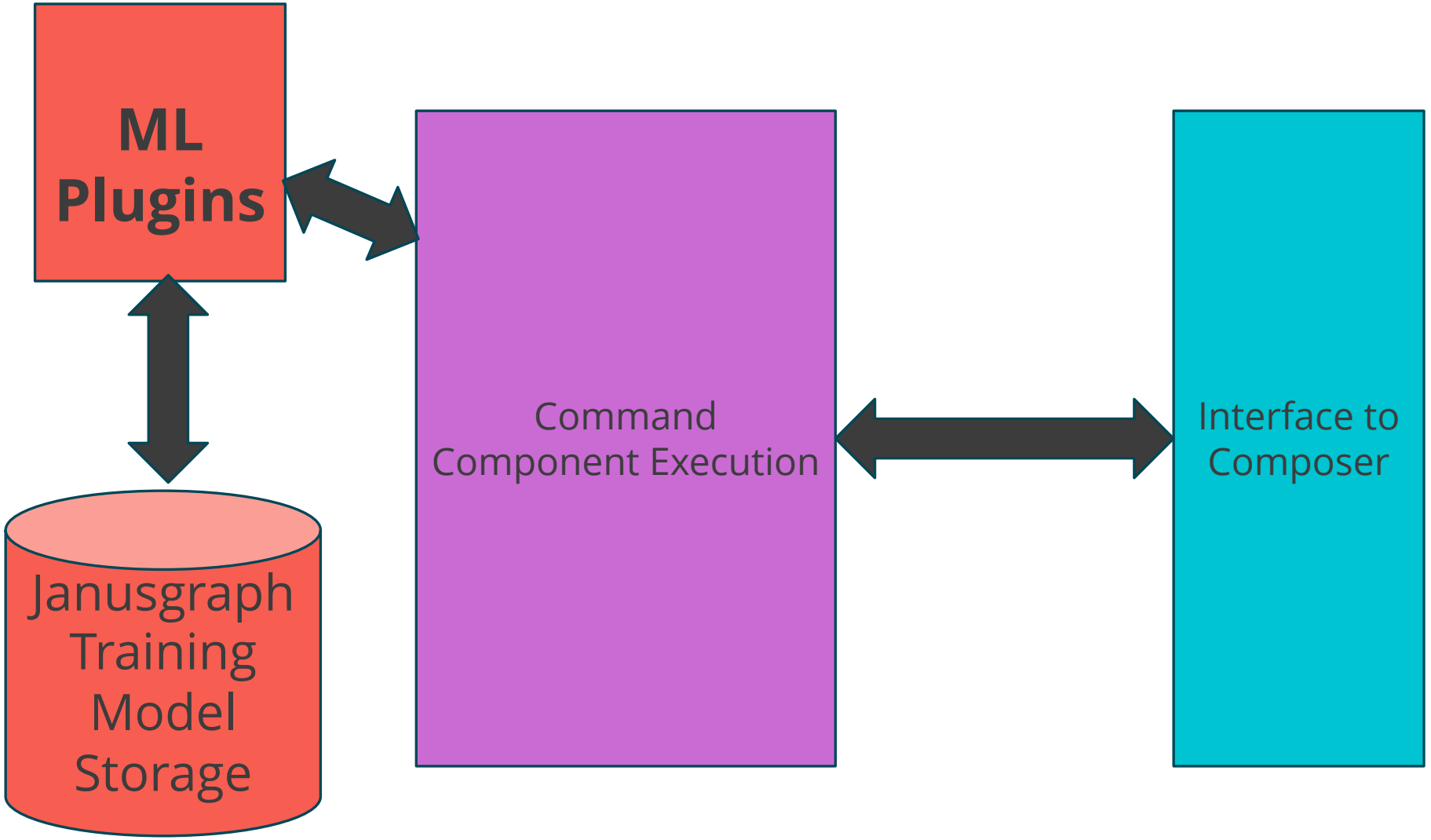
Interface to Sunfish, resources, paths, and locality

Interface to Resource Graph Representation

Log-based Input for training and updating ML model
Run time, network congestion, start-up latency, IO performance, memory transaction bandwidth
Federation or Reinforcement

Client transaction commands

Client Request



Acknowledgements and Questions?

- OpenFabrics Management Framework Working Group
  - Doug Ledford, Phil Cayton, Mike Aguilar, Christian Pinto, Richelle Ahlvers, Russ Herrell, Michele Gazzetti, Jeff Hilland, John Mayfield, Jim Hull, Tracy Spitler, Chris Morrone, Eugene Novak, Dennis Dallesandro, Kurt Bowman, Catherine Appleby, etc.