



SDC 

STORAGE DEVELOPER CONFERENCE

SNIA  SANTA CLARA, 2017

SwordflshJS

- A Swordflsh JS Library

**Vinod Eswaraprasad, Sowmya B
Wipro Technologies**

What we want to talk ?

- ❑ Manageability at HyperScale
- ❑ Redfish and Swordfish – For Better Health
- ❑ A look inside Swordfish – Usage Model
- ❑ The Reusable Recipe – JS Library
- ❑ Easy SRM Dev - Demo
- ❑ Questions/Comments ?



Manageability for HyperScale



Digital Infrastructure == Hyper Scale

- Digital Infrastructure of today is large set of common hardware.
- Current Infrastructure management suffers with scale
 - Performance
 - Reliability
 - Security
- Modelling difficulty in a multi-vendor environment
- Non-standard tools and frameworks



Solving the Web-scale Manageability Problems

- Web-scale is best managed by Web interface based protocol
- Less Chatty
 - More information in fewer transactions
- Common APIs – Restful
- Internet standards and tool chains
 - Language Support
- Simplify the manageability protocol



What Are The Choices?

- ❑ Well-known protocol – Common CRUD semantics
- ❑ Make the discovery easy
- ❑ Primarily Out-of-band (Host interface optional)
- ❑ Easily Extensible
- ❑ Supports – Compute, Network and Storage



Redfish and Swordfish



DMTF Redfish - Basics

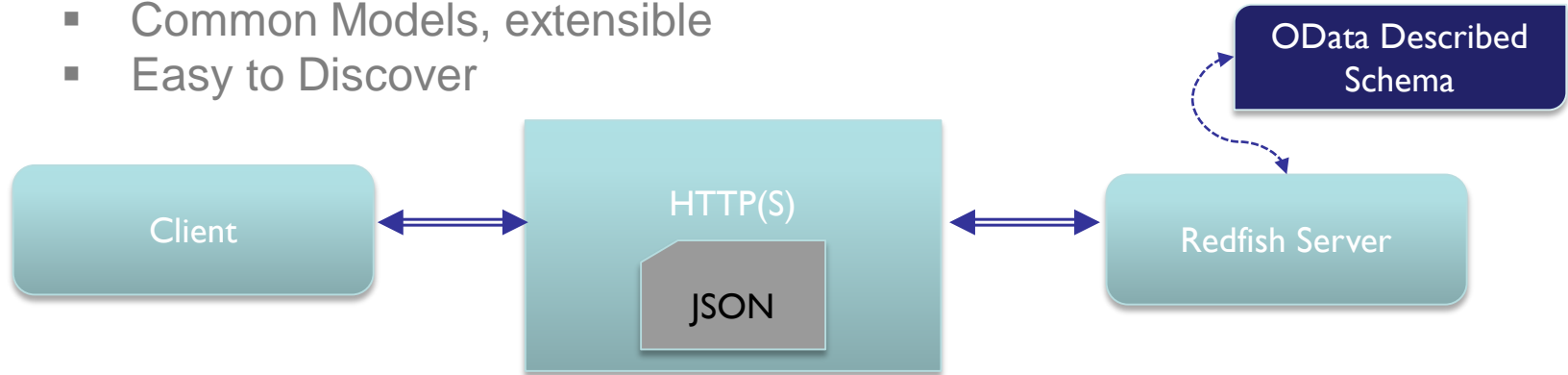
Rest + HTTP(S) + JSON

- ❑ HTTP methods are used as protocol for common CRUD operations
- ❑ A Redfish interface shall be exposed through a web service endpoint
- ❑ Hypermedia API with a small set of defined URIs



Redfish – Hypermedia Based Protocol

- Protocols and a core set of data models and behaviors for the management of systems
- Redfish Interface
 - Restful
- Redfish Models
 - Common Models, extensible
 - Easy to Discover



✓ Secure

✓ Scalable

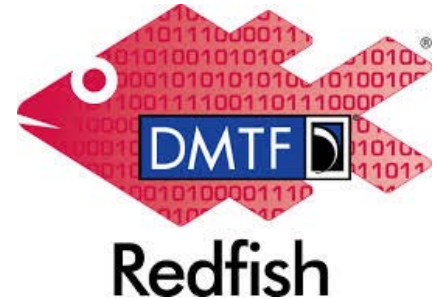
✓ Extensible

✓ Commonly Used



Redfish - Features

- OData convention
 - Resources modelled using OData, and translated to JSON
- Model Orientated
 - No dependency between Model and Protocol; can change
- Sync and Async operations
 - Time consuming tasks at the server side
- Event support
 - Time Critical State Change or Errors
- Actions Support
 - Like Reset operation

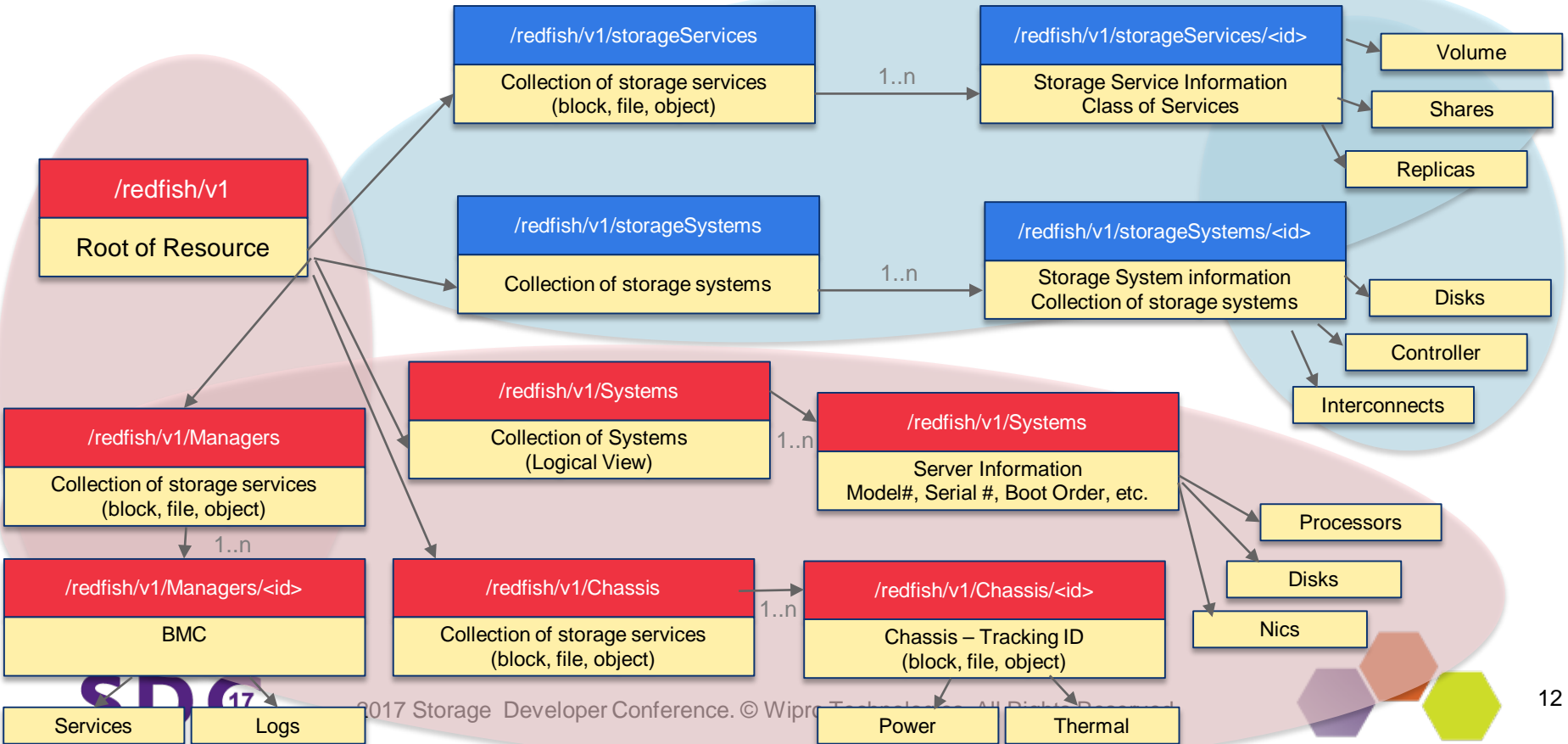


SNIA Swordfish – Storage Model over Redfish

- ❑ Extension to Redfish to support Storage
- ❑ Model for Scalable storage and associated data services
- ❑ Storage Services
 - ❑ snapshots, replication, mapping and masking, and provisioning
- ❑ Wide Range of Storage
 - ❑ Small Object Drive – to- RAID arrays – File Server – Converged Systems, Hyper converged Cloud scale storage



Swordfish – Data Model Overview



Reusable JS Library for clients?



SRM Using Swordfish

Business Goals to storage specific actions and requirements

- ❑ Common Storage Resource Management Tasks
 - ❑ Configuration and provisioning
 - ❑ Resource Monitoring
 - ❑ Event and log management
 - ❑ Performance assessment
 - ❑ Diagnostics, Fault detection and remediation
 - ❑ Accounting and resource consumption

The Management Application should talk Swordfish....

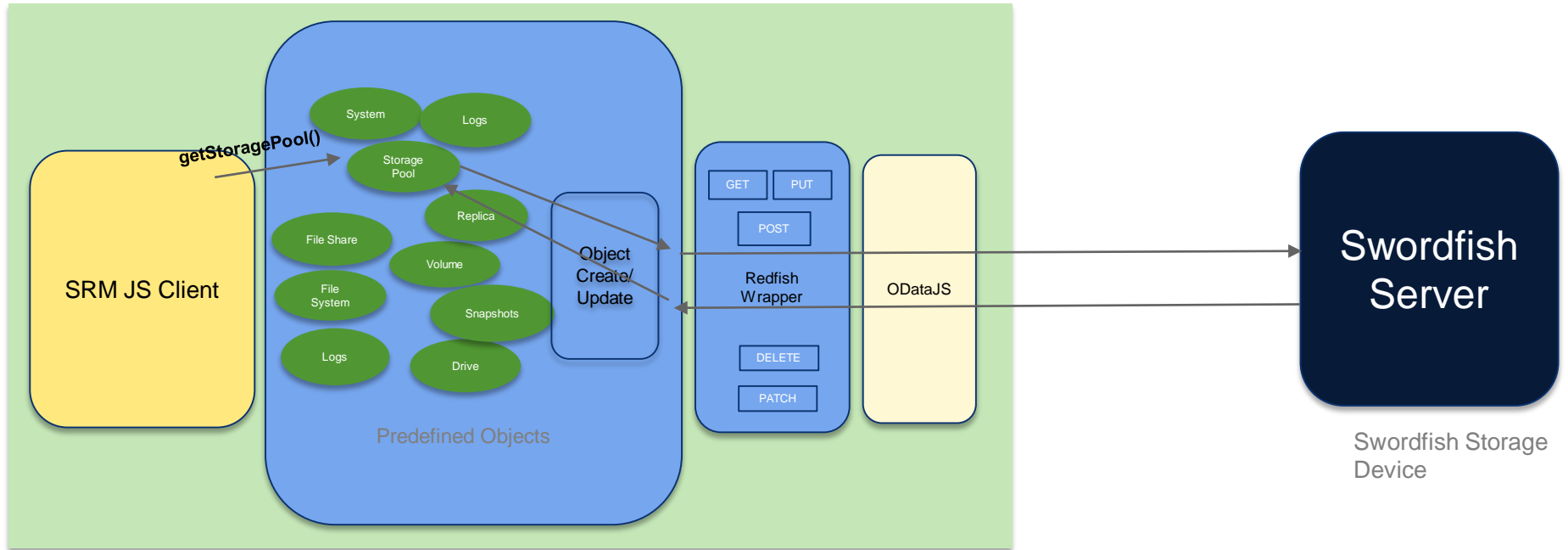


Reusable Objects - Framework

- ❑ Goal of the SwordfishJS
 - ❑ Provides an easy way to access redfish/swordfish resources within JS Clients
 - ❑ Set of JS APIs that wraps
 - ❑ GET, PATCH, PUT, POST and DELETE Operations
 - ❑ Provide a of pre-defined set of JSON objects - directly used by the Application
- ❑ Abstracts the complexity of the protocol from application developers
- ❑ Leverage existing JS modules
 - ❑ ODataJS



Swordfish JS - Operation



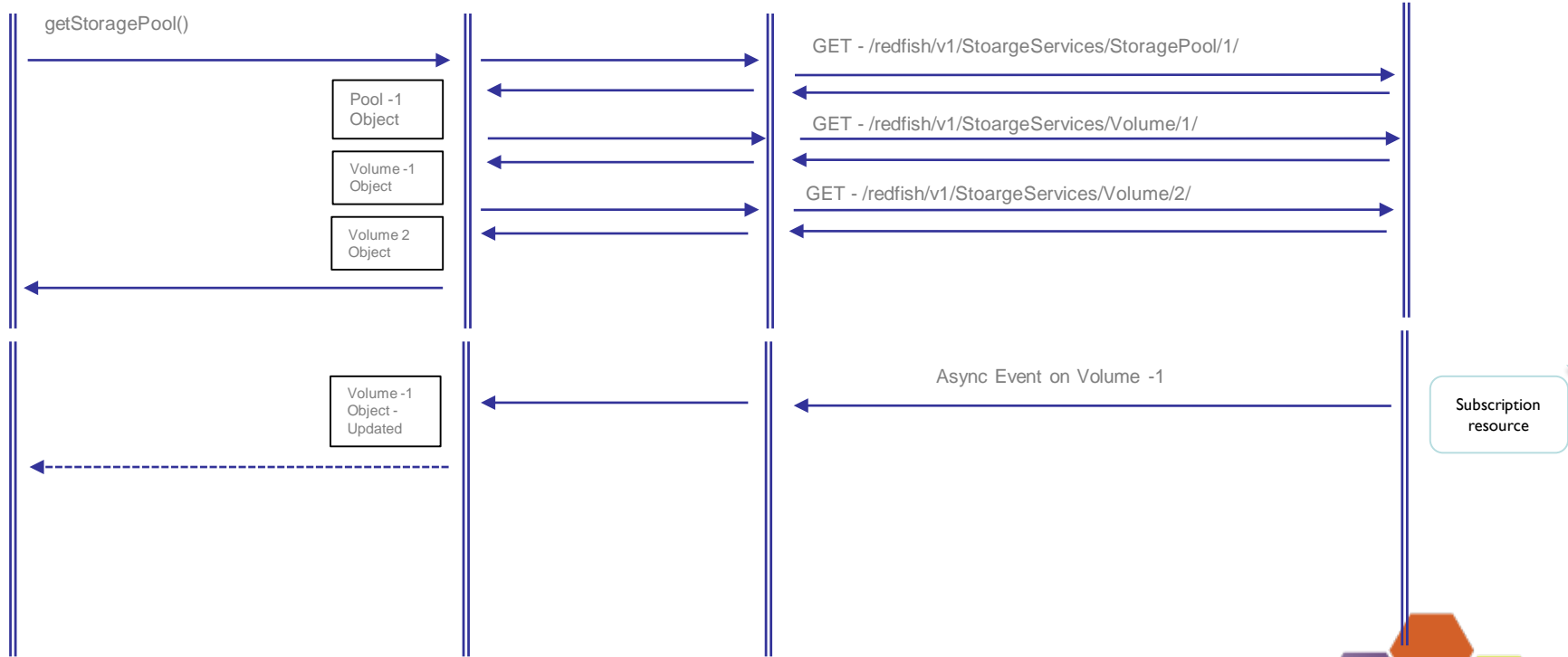
Storage Object and Swordfish - Interactions

JS Application

SwordfishJS-Library

Redfish Wrapper

Swordfish Server



The JS Storage Resource – Sample Layout

| Storage System | GetStorageSystem() |
|---|----------------------|
| Name/Description IP address Status Model Serial # Firmware Version | ResetStorageSystem() |

| Snapshot | GetSnapshot() |
|---|------------------|
| Name/Description | CreateSnapshot() |
| Status | DeleteSnapshot() |
| Capacity Allocated Capacity Remaining Capacity Provisioning Type | UpdateSnapshot() |

| File Share | GetFileShare() |
|---|--|
| Name/Description Size Status Protocol Access Path | CreateFileShare () DeleteFileShare () UpdateFileShare () |

| Storage Pool | GetStoragePool() |
|---|---------------------|
| Name/Description | CreateStoragePool() |
| Status | DeleteStoragePool() |
| Capacity Allocated Capacity Remaining Capacity Allocated Volumes Drives | UpdateStoragePool() |

| Drives | GetDrives() |
|---|-------------|
| Name Size Status Location Media Type Speed Serial Number Volumes | SetDrives() |

| File System | GetFileSystem() |
|--|--|
| Name | CreateFileSystem() |
| Capacity Allocated Capacity Remaining Capacity Shares | DeleteFileSystem() UpdateFileSystem() |

| Volume | GetVolume() |
|---|----------------|
| Name/Description | CreateVolume() |
| Status | DeleteVolume() |
| Capacity Allocated Capacity Remaining Capacity Provisioning Type | UpdateVolume() |

| Replica Info | GetReplicaInfo() |
|--|------------------|
| Replica Role Source Target Progress Status Replica Type Replica State | |

| Log Entry | GetLogEntry() |
|--|-------------------|
| Log Entry Code Log Entry Type Log Entry Message ID Message Args Message | CreateLogEntry () |



What we have today and way forward ?

- ❑ Fully compliant Redfish Wrapper
- ❑ JS Object wrapper (with GET/PUT/POST/DELETE) APIs
 - ❑ System
 - ❑ Storage Pool
 - ❑ Volume
 - ❑ Drives
 - ❑ File Share
 - ❑ File system
 - ❑ Snapshot
 - ❑ Log
- ❑ Support Event based Object State Update
- ❑ Support Actions on Objects



Reusable JS – Easy SRM Demo



The SwordfishJS Usage – Setup and Demo

Setup

- ❑ Demonstration of the SwordfishJS usage in sample Management Application
- ❑ Ability to quickly develop management actions
 - ❑ Sample Grommet JS Application
 - ❑ Swordfish Mockup schema and objects
 - ❑ Nginx webserver

Demo

- Storage System Status
 - Pools and Volume Information
- Pools and Volume Data gathering
 - Utilization
 - Health
- Volume Creation
- Event Handling



Learnings and Shortcomings...

- Mapping high level storage resources to the Swordfish Schema
 - Aggregation
- Discovery process by navigating the GET response from Service Root
- Handling ASYNC operations to update object status – special case
- No direct way to identify snapshot volumes
- Unavailability of performance statistics data in the current swordfish data model
- Very less Diagnostic actions support



Questions ?



Thank You.

