SwordfIshJS

- A SwordfIsh JS Library

25 May 2017

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

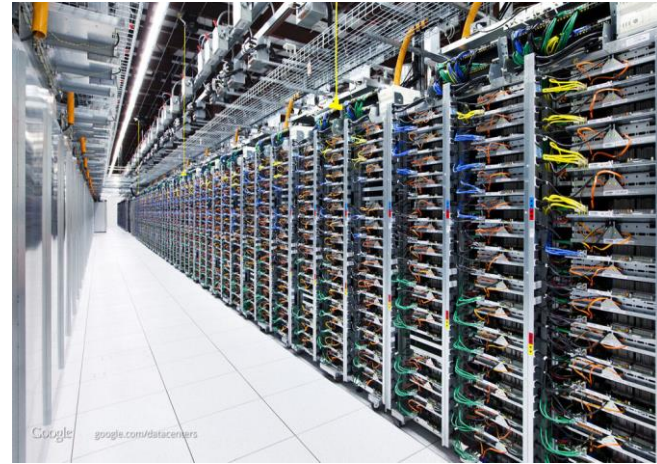- Easy SRM Dev - Demo

- 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 <u>Web interface based</u> 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

OData Described Schema

HTTP(S)

JSON

Client

Redfish Server

✓ 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
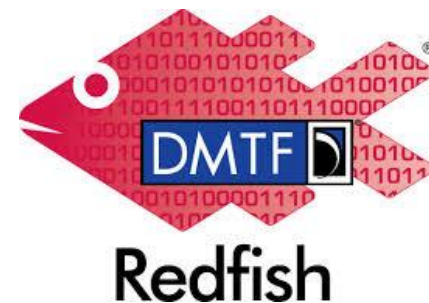
SNIA Swordfish – Storage Model over Redfish

- 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
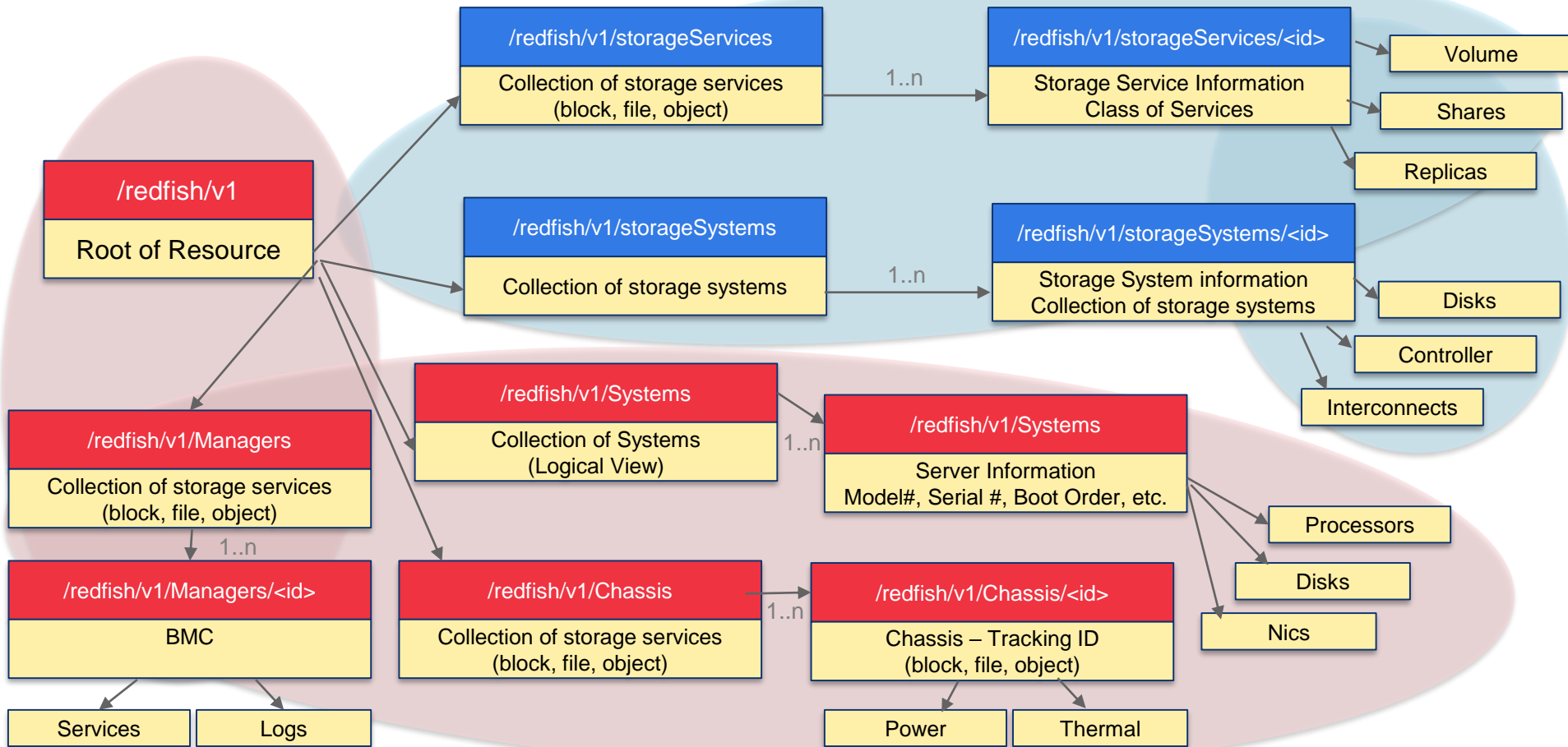
Reusable JS Library for clients?

SRM Using Swordfish

Business Goals to storage specific actions and requirements
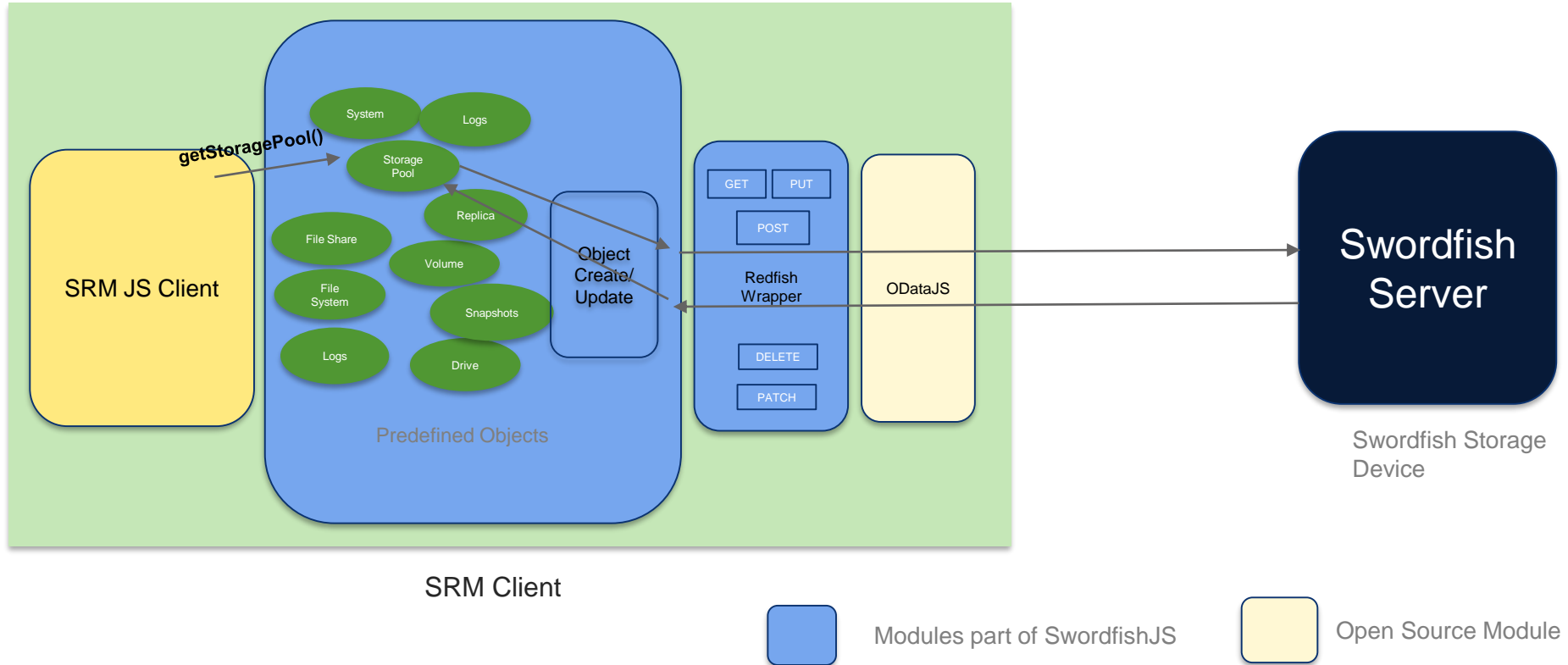
- Common Storage Resource Management Tasks
    - Configuration and provisioning
    - Resource Monitoring
    - Event and log management
    - Performance assessment
    - Diagnostics, Fault detection and remediation
    - Accounting and resource consumption

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

getStoragePool()

SRM JS Client

System | Logs

Storage Pool

Replica

File Share

Volume

File System

Snapshots

Logs | Drive

Object Create/ Update

Predefined Objects

Redfish Wrapper

GET | PUT

POST

DELETE

PATCH

ODataJS

Swordfish Server

Swordfish Storage Device

SRM Client

Modules part of SwordfishJS

Open Source Module

Storage Object and Swordfish - Interactions

JS Application   SwordfishJS-Library   Redfish Wrapper   Swordfish Server

getStoragePool()

GET - /redfish/v1/StoargeServices/StoragePool/1/

Pool -1
Object

GET - /redfish/v1/StoargeServices/Volume/1/

Volume -1
Object

GET - /redfish/v1/StoargeServices/Volume/2/

Volume 2
Object

Volume -1
Object -
Updated

Async Event on Volume -1

Subscription
resource

The JS Storage Resource – Sample Layout

Storage System
Name/Description
IP address
Status
Model
Serial #
Firmware Version

- GetStorageSystem()
- ResetStorageSystem()

Snapshot
Name/Description
Status
Capacity
Allocated Capacity
Remaining Capacity
Provisioning Type

- GetSnapshot()
- CreateSnapshot()
- DeleteSnapshot()
- UpdateSnapshot()

File Share
Name/Description
Size
Status
Protocol
Access Path

- GetFileShare()
- CreateFileShare ()
- DeleteFileShare ()
- UpdateFileShare ()

Storage Pool
Name/Description
Status
Capacity
Allocated Capacity
Remaining Capacity
Allocated Volumes
Drives

- GetStoragePool()
- CreateStoragePool()
- DeleteStoragePool()
- UpdateStoragePool()

Drives
Name
Size
Status
Location
Media Type
Speed
Serial Number
Volumes

- GetDrives()
- SetDrives()

File System
Name
Capacity
Allocated Capacity
Remaining Capacity
Shares

- GetFileSystem()
- CreateFileSystem()
- DeleteFileSystem()
- UpdateFileSystem()

Volume
Name/Description
Status
Capacity
Allocated Capacity
Remaining Capacity
Provisioning Type

- GetVolume()
- CreateVolume()
- DeleteVolume()
- UpdateVolumel()

Replica Info
Replica Role
Source
Target
Progress Status
Replica Type
Replica State

- GetReplicaInfo()

Log Entry
Log Entry Code
Log Entry Type
Log Entry
Message ID
Message Args
Message

- GetLogEntry()
- 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.

Vinod.eswar@wipro.com