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 **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

OData Described Schema
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

- **/redfish/v1**
  - Root of Resource
  - Collection of storage services (block, file, object)

- **/redfish/v1/storageServices**
  - Collection of storage services (block, file, object)
  - Storage Service Information
  - Class of Services

- **/redfish/v1/storageSystems**
  - Collection of storage systems
  - Storage System information

- **/redfish/v1/Managers**
  - Collection of storage services (block, file, object)

- **/redfish/v1/Managers/<id>**
  - BMC
  - Services
  - Logs

- **/redfish/v1/Systems**
  - Collection of Systems (Logical View)
  - Server Information
  - Model#, Serial #, Boot Order, etc.

- **/redfish/v1/Chassis**
  - Collection of storage services (block, file, object)
  - Chassis – Tracking ID
  - (block, file, object)

- **/redfish/v1/Chassis/<id>**
  - Power
  - Thermal

- **/redfish/v1/Managers**
  - BMC

- **/redfish/v1/Managers/<id>**
  - BMC

- **/redfish/v1/Systems**
  - Collection of Systems (Logical View)
  - Server Information
  - Model#, Serial #, Boot Order, etc.

- **/redfish/v1/Chassis**
  - Collection of storage services (block, file, object)

- **/redfish/v1/Chassis/<id>**
  - Power
  - Thermal
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

SRM JS Client

getStoragePool()

SRM Client

Object Create/Update

Predefined Objects

SRM JS Client

Swordfish Server

Swordfish Storage Device

Open Source Module

Modules part of SwordfishJS

GET
PUT
POST
DELETE
PATCH

Redfish Wrapper

ODataJS

System
Logs
Storage Pool
Replica
File Share
Volume
Snapshots
Drive
File System
Logs

SRM JS Client

Swordfish Server

Swordfish Storage Device

Open Source Module

Modules part of SwordfishJS
Storage Object and Swordfish - Interactions

JS Application

getStoragePool()

SwordfishJS-Library

Pool -1 Object
Volume -1 Object
Volume 2 Object

Redfish Wrapper

GET - /redfish/v1/StoargeServices/StoragePool/1/
GET - /redfish/v1/StoargeServices/Volume/1/
GET - /redfish/v1/StoargeServices/Volume/2/

Swordfish Server

Async Event on Volume -1

Subscription resource
### The JS Storage Resource – Sample Layout

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

- **ResetStorageSystem()**

#### Storage Pool
- **GetStoragePool()**
  - **Name/Description**
  - **Status**
  - **Capacity**
  - **Allocated Capacity**
  - **Remaining Capacity**

- **CreateStoragePool()**
- **DeleteStoragePool()**
- **UpdateStoragePool()**

#### Volume
- **GetVolume()**
  - **Name/Description**
  - **Status**
  - **Capacity**
  - **Allocated Capacity**
  - **Remaining Capacity**
  - **Provisioning Type**

- **CreateVolume()**
- **DeleteVolume()**
- **UpdateVolume()**

#### Drives
- **GetDrives()**
  - **Name**
  - **Size**
  - **Status**
  - **Location**
  - **Media Type**
  - **Speed**
  - **Serial Number**

#### Snapshot
- **GetSnapshot()**
  - **Name/Description**
  - **Status**
  - **Capacity**
  - **Allocated Capacity**
  - **Remaining Capacity**
  - **Provisioning Type**

- **CreateSnapshot()**
- **DeleteSnapshot()**
- **UpdateSnapshot()**

#### File Share
- **GetFileShare()**
  - **Name/Description**
  - **Size**
  - **Status**
  - **Protocol**
  - **Access Path**

- **CreateFileShare()**
- **DeleteFileShare()**
- **UpdateFileShare()**

#### File System
- **GetFileSystem()**
  - **Name**
  - **Capacity**
  - **Allocated Capacity**
  - **Remaining Capacity**
  - **Shares**

- **CreateFileSystem()**
- **DeleteFileSystem()**
- **UpdateFileSystem()**

#### 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**

- **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