Emerging Scalable Storage Management Functionality: What’s New in 2019

Richelle Ahlvers
Broadcom Inc.
Disclaimer

- The information in this presentation represents a snapshot of work in progress within SNIA
- This information is subject to change without notice.
- For additional information, see the SNIA website: www.snia.org/swordfish
Abstract

- By now, you have a good understanding of SNIA Swordfish™ and how it extends the DMTF Redfish® specification to manage storage equipment and services. This presentation covers what’s new and how the specification has evolved in the last year.

- Developed by the Storage Networking Industry Association (SNIA), SNIA Swordfish™ is an extension of the DMTF Redfish specification to provide a unified approach for the management of storage equipment and services in converged, hyper-converged, hyperscale and cloud infrastructure environments, making it easier for IT administrators and DevOps to integrate scalable solutions into their data centers.
Swordfish Specs and Technical Timeline

2016: v1.0.0 Released: Block and File functionality with Class of Service Interface
2017: v1.0.2 – 1.0.5: Enhancements, etc..
2018:
  - v1.0.6: Introduction of two StorageSystem models, updated on-demand replica models
  - v1.0.7 Swordfish WIP Release: Enhanced Spare Capacity Management, Rebuild Management, Volume types, YAML schema support
  - Spare Management White Paper

2019:
  - v1.1.0 - Swordfish Features and Profiles WIP release
    - Closed “gap” between Redfish Storage model and Swordfish
    - Enhanced features and functionality requested to support scalability in direct-attach use cases
    - Updated Swordfish mockups: swordfishmockups.com

Future Functionality
- Full NVMe Enablement: Functionality alignment across DMTF, NVMExpress/NVMe-MI and SNIA for NVMe use cases
- Storage-specific security roles
- Object Storage
- Enhanced profile support for SNIA Alliance partner organizations
Other Expanded Functionality

- Added Consistency Groups
- Added Replication controls to Volumes and Consistency Groups
  - AssignReplicaTarget
  - CreateReplicaTarget
  - RemoveReplicaRelationship
  - ResumeReplication
  - ReverseReplicationRelationship
  - SplitReplication
  - SuspendReplication
- Expanded Volume schema
- User’s Guide significantly expanded
  - Use cases covering RF, SF and SF with CoS
    - Volume creation, Storage Group creation, Replication management
  - Sorted by Feature
Lifecycle: From Definition to Implementations

1. Describe
2. Prescribe
3. Implement
4. Test Conformance

- Redfish
- OpenStack
- OpenBMC
- UEFI
- SNIA
- SMI
- Storage Management
- openSDS
- GitHub
Starting with Redfish: Simple Storage

Volumes are in Collections off of the Storage resource, drives are in arrays off of the storage resource, and optionally, the Chassis.

2019 Storage Developer Conference. © Broadcom Inc. All Rights Reserved.
Adding Swordfish...

Swordfish schema (StoragePools, StorageGroups, ConsistencyGroups, etc) attach to storage.

Once Pools exist, move where Volumes are created from VolumeCollection to StoragePool Collection. Also, manage media (drives) from pools.
Supporting Class Of Service

Class of service shows up at Service Root; underlying resources can be pulled from the extensions added to Storage or directly.
Swordfish Capabilities

- Advertised using “SupportedFeatures” (Features)
  - Features are high-level descriptions of functionality that an implementation advertises that it currently supports
  - Profiles are detailed descriptions that describe down to the individual property level what functionality is required in order to advertise features

- **Block storage**
  - Provisioning with optional class of service control
    - Resource provisioning from disk, volume, pool, and persistent memory
  - Volume Mapping and Masking
  - Local and Remote Replication
  - Capacity and health metrics
  - Performance metrics

- **File system storage**
  - Adds File System and File Share
  - Leverages all other concepts – provisioning with class of service, replication, Solution level connectivity
  - Fabric connect, host connect
    - Endpoint abstraction
  - **Additional content**
    - Object drive storage
Primary Swordfish Elements

- Volume: Block addressable storage.
- Filesystem: File-addressable storage.
- StoragePools: Storage capacity that can be used to produce volumes or other storage pools.
- StorageGroup: A set of volumes and endpoints that are managed as a group for mapping and masking.
- ConsistencyGroup: A set of volumes that are treated by an application or set of applications as a single entity.
- Fileshare: A shared set of files with a common directory structure that is exported for use by remote systems.

Optional Elements:

- ClassOfService: A choice of utility or warranty offered to customers by a service. Defined by selecting from available LinesOfService.
- StorageService: Represents a service that provides ClassOfService based provisioning, management, and monitoring for logical storage and associated resources.
Features and Profiles: Specifying Required Functionality
Swordfish Features and Profiles

- **Features**
  - High-level descriptions of functionality that an implementation advertises that it currently supports

- **Profiles**
  - Profiles are detailed descriptions that describe down to the individual property level what functionality is required in order to advertise features

- **Client / Storage Profile**
  - Selected combination of Swordfish profiles

- **The Profile Files**
  - JSON formatted files
  - Specifies required Redfish/Swordfish interface elements (resources, properties, values)

---

2019 Storage Developer Conference. © Broadcom Inc. All Rights Reserved.
Swordfish Profiles Inheritance Hierarchy

- Swordfish Discovery
  - Event Notification
  - IO Performance
    - EnergyStar for Storage
  - Swordfish File Provisioning
    - Local Replication
    - File Capacity Management
    - Mapping and Masking
  - Swordfish Block Provisioning
    - Remote Replication
    - Block Capacity Management
  - Swordfish Class Of Service Discovery
    - CoS Local Replication
    - CoS Remote Replication
Implementations Advertise Supported Features

- Accessed from FeaturesRegistry
  - Accessible at the ServiceRoot
- Set of SupportedFeatures that contains:
  - Feature Name
  - Description
  - Version
  - Corresponding Profile Definition
"@Redfish.Copyright": "Copyright 2019 SNIA. All rights reserved.",
"@odata.type": "#FeaturesRegistry.v1_0_0.FeaturesRegistry",
"Id": "SwordfishFeatures.1.0.2",
"Name": "Swordfish Features Registry",
"Language": "en",
"Description": "This registry defines the Swordfish features.",
"RegistryPrefix": "SwordfishFeatures",
"RegistryVersion": "1.0.2",
"OwningEntity": "SNIA",
"Features": {
  "SNIA.Swordfish.Discovery": {
    "Description": "Supports discovery of resources in a Swordfish system.",
    "Version": "1.0.1",
    "CorrespondingProfileDefinition": "SwordfishDiscovery.json"
  },
  "SNIA.Swordfish.EventNotification": {
    "Description": "Supports the Swordfish Event Notification Feature.",
    "Version": "1.0.1",
    "CorrespondingProfileDefinition": "SwordfishEventNotification.json"
  },
  "SNIA.Swordfish.Block.IOPerformance": {
    "Description": "Supports the Swordfish Block IOPerformance Feature.",
    "Version": "1.0.1",
    "CorrespondingProfileDefinition": "SwordfishIOPerformance.json"
  },
...
## Profile Definition Deep-Dive: Swordfish Discovery - Requirements

<table>
<thead>
<tr>
<th>Requirement Area</th>
<th>Detail</th>
</tr>
</thead>
</table>
| Required behaviors:      | “Get” on all system object implemented  
                           | “Get” on all object properties implemented                                                                                             |
| Required properties:     | ComputerSystem -> HostingRoles property must be set to StorageServer.  
                           | Volumes Collection in Storage.  
                           | Storage Pools Collection in Storage.  
                           | Must have a Collection for storage media (either Drives or in an attached CapacitySources structure).  
                           | StorageSystem Collection in ServiceRoot, at least one member.                                                                         |
Profile Definition Deep-Dive: Swordfish Discovery – Profile Headers

```json
{
   "SchemaDefinition": "/RedfishInteroperabilityProfile.v1_3_0.json",
   "ProfileName": "SwordfishDiscovery",
   "ProfileVersion": "1.0.1",
   "OwningEntity": "SNIA.org",
   "Purpose": "Define requirements for discovery of resources and relationships that are common to all Swordfish confromant implementations.",
   "ContactInfo": "SNIA.org",
   "RequiredProfiles": {},
   "Registries": {
      "Features": {
         "MinVersion": "1.0.2",
         "Repository": "https://redfish.dmtf.org/profiles/swordfish",
         "SupportedFeatures": {
            "SNIA.Swordfish.Discovery": {}
         }
      }
   }
}
```
Profile Definition Deep-Dive: Swordfish Discovery – Set `HostingRoles` Property

```json
"Resources": {
  "ServiceRoot": {
    "PropertyRequirements": {
      "StorageSystems": {},
      "Systems": {
        "ComputerSystem": {
          "PropertyRequirements": {
            "HostingRoles": {
              "ConditionalRequirements": [{
                "Purpose": "Swordfish implementations must specify a hosting role type of StorageServer.",
                "CompareValues": "StorageServer"
              }]
          }
        }
      }
    }
  }
}
```
Profile Definition Deep-Dive: Swordfish Discovery – Requirements for Volumes, StoragePools, Media Collections

"Resources": {
  "ServiceRoot": {
    "PropertyRequirements": {
      "StorageSystems": {},
      "Systems": {
        "ComputerSystem": {
          ...
          "Storage": {
            "Volumes": {
              "PropertyRequirements": {
                "Members": {
                  "MinCount": 0
                }
              }
            },
            "StoragePools": {
              "PropertyRequirements": {
                "Members": {
                  "MinCount": 0
                }
              }
            }
          }
        }
      }
    }
  }
}

"Drives": {
  "ReadRequirement": "Recommended",
  "ConditionalRequirements": [{
    "Purpose": "ConditionalRequirements: The implementation must choose at least one selected media container (appropriate for the system's selected media type) and instantiate a collection for it. This may be instantiated as a CapacitySources structure to feed a StoragePool, FileSystem, or Volume. When drives are used, the Drives Collection in either Storage or, when implementing the ClassOfService Feature, StorageServices shall be implemented.",
    "CompareType": "Absent",
    "ReadRequirement": "Mandatory",
    "Comparison": "AnyOf",
    "CompareValues": ["StoragePool/CapacitySource", "FileSystem/CapacitySource", "Volume/CapacitySource"]
  }]
}
...

Profile Definition Deep-Dive:
Swordfish Discovery – General Requirements

```
"StorageSystemCollection": {
    "PropertyRequirements": {
        "Members": {
            "MinCount": 1
        }
    },

"StoragePoolCollection": {
    "PropertyRequirements": {
        "Members": {
            "MinCount": 0
        }
    },

    "VolumeCollection": {
        "PropertyRequirements": {
            "Members": {
                "MinCount": 0
            }
        }
    }
```

...
Swordfish Info: www.snia.org/swordfish

- Resources
  - Specifications
  - User’s Guide
  - GitHub for Swordfish Tools
  - Practical Guide
  - Other Documentation
- Swordfish Mockups Site
  - ISC and HSC configurations
  - Block vs. file configurations
  - Small and large configurations
- Education/Community
  - Whitepapers, Presentations
  - YouTube shorts & Webinars
- Participate
  - Join SNIA and the SSM TWG
  - Implement
Open Source Tools and Infrastructure Development

- Available: [http://github.com/snia](http://github.com/snia)
  - Swordfish Emulator Extensions
    - Extends the Redfish emulator – adds all Swordfish schema (behave like dynamic objects)
  - Basic Swordfish Web client
    - Discover, display and edit Swordfish services
  - DataDog and Power BMI Client Sample Dashboards
    - Sample implementations show integration concepts with sample code:
      - PowerBI: Point-in-time dashboard; Datadog: Data trending dashboard
  - NEW! Swordfish Powershell Toolkit
    - Powershell toolkit integration for Windows and Linux
How to Participate: Shaping the Standard

- Find pointers to the latest technical content:
  - [http://snia.org/swordfish](http://snia.org/swordfish)
  - [http://www.snia.org/publicreview#swordfish](http://www.snia.org/publicreview#swordfish)
- Join the SSM TWG
  - By Joining the SNIA and SSM TWG, you can shape the standard: [https://members.snia.org/apps/org/workgroup/ssmtwg](https://members.snia.org/apps/org/workgroup/ssmtwg)
- Through the SNIA feedback portal, providing feedback on “Work In Progress”
  - As the group produces “Works In Progress”, you can provide feedback at [http://www.snia.org/feedback](http://www.snia.org/feedback)
Q&A