SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

SNIA Swordfish™ Overview and Deep Dive

Richelle Ahlvers
SNIA SSM TWG Chair

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

- 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.

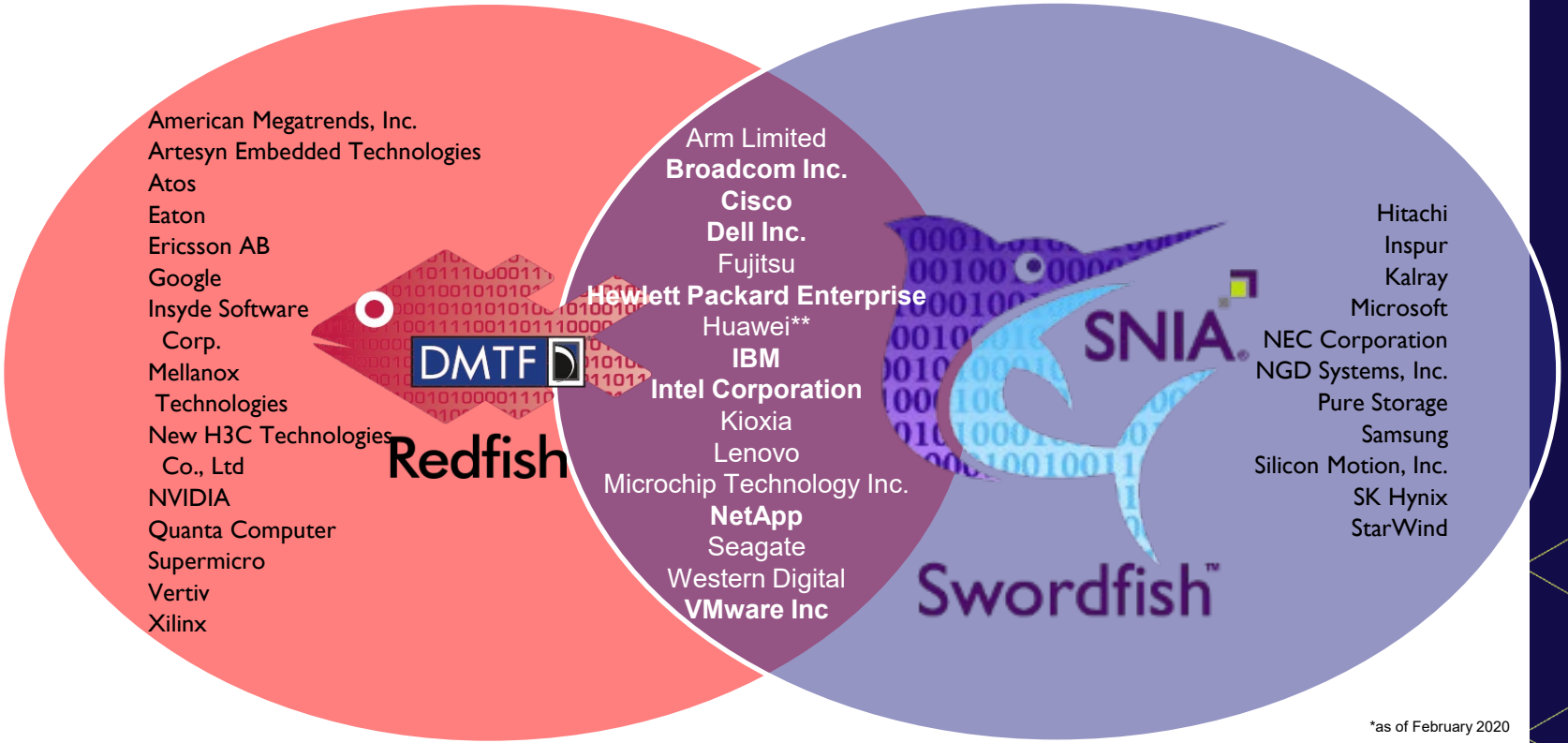
- This session will present an overview of the SNIA Swordfish specification, and will show how Swordfish takes and extends the Redfish specification to deliver the Swordfish storage model. It will also cover the drivers for the SNIA Swordfish approach, as well as providing a comprehensive overview of the functionality included in the Swordfish specification.

The SNIA Swordfish™ Approach

SDC 20

- Develop the management model from the point-of-view of what a client needs to accomplish and to provide information that the client needs
- Cover block, file, and object storage
- Extend traditional storage domain coverage to include converged environments (covering servers, storage and fabric together)
- Provide the option for implementation to utilize Class of Service (intent or service level) based provisioning, management, and monitoring
- Implement the Swordfish API as an ***extension*** of the Redfish API
- Build using DMTF's Redfish technologies
  - RESTful interface over HTTPS in JSON format based on OData v4

Who is Developing Redfish and Swordfish*?

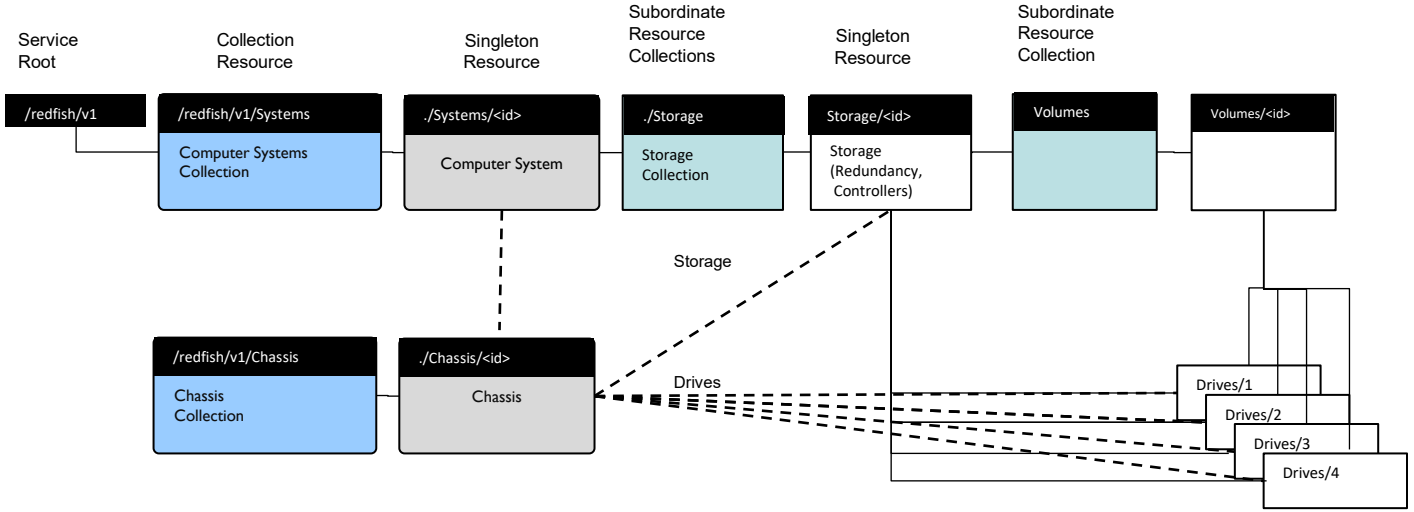


*as of February 2020

** Membership suspended

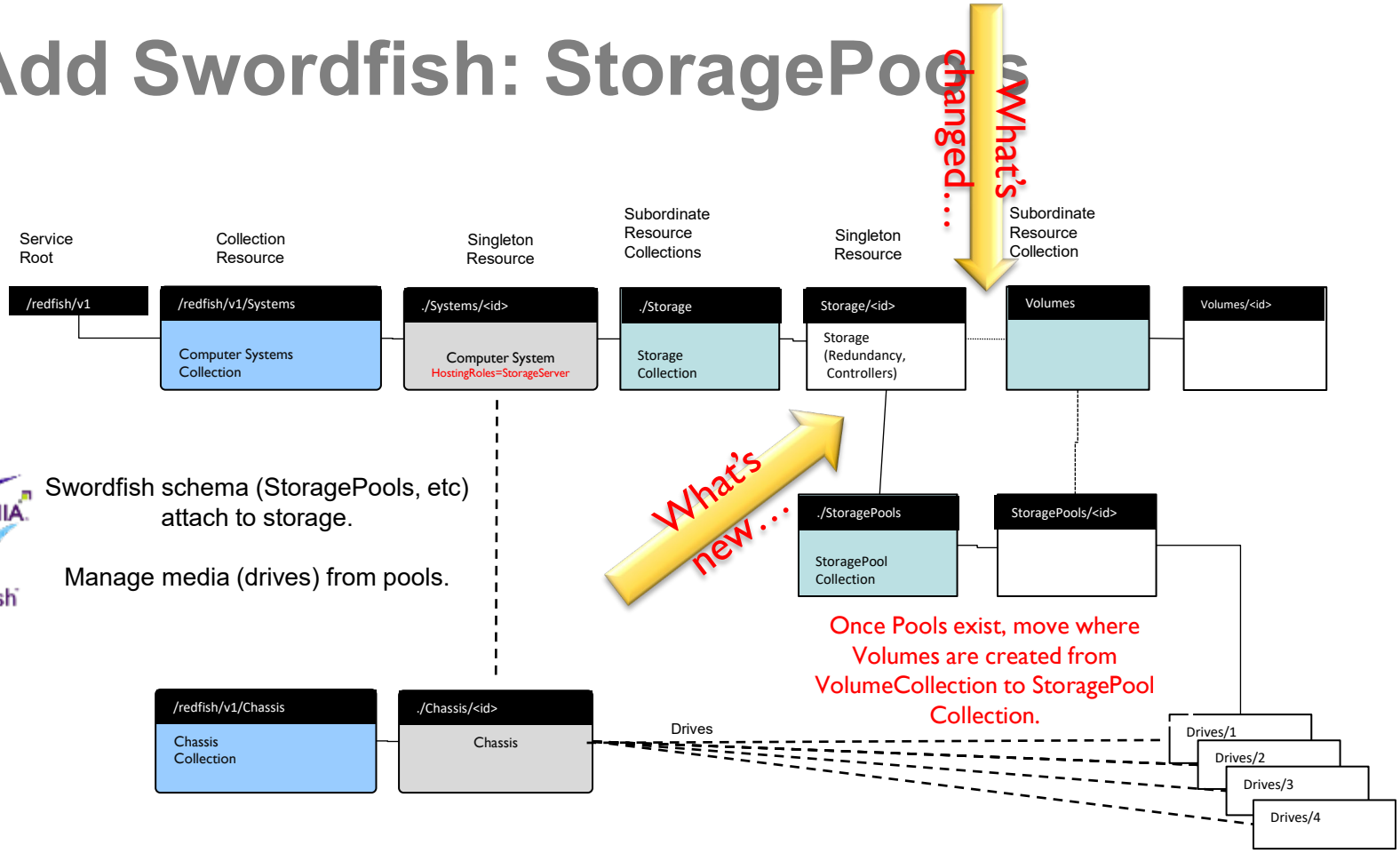
Swordfish: Extending the Redfish REST Model

Starting with Redfish: Simple Storage

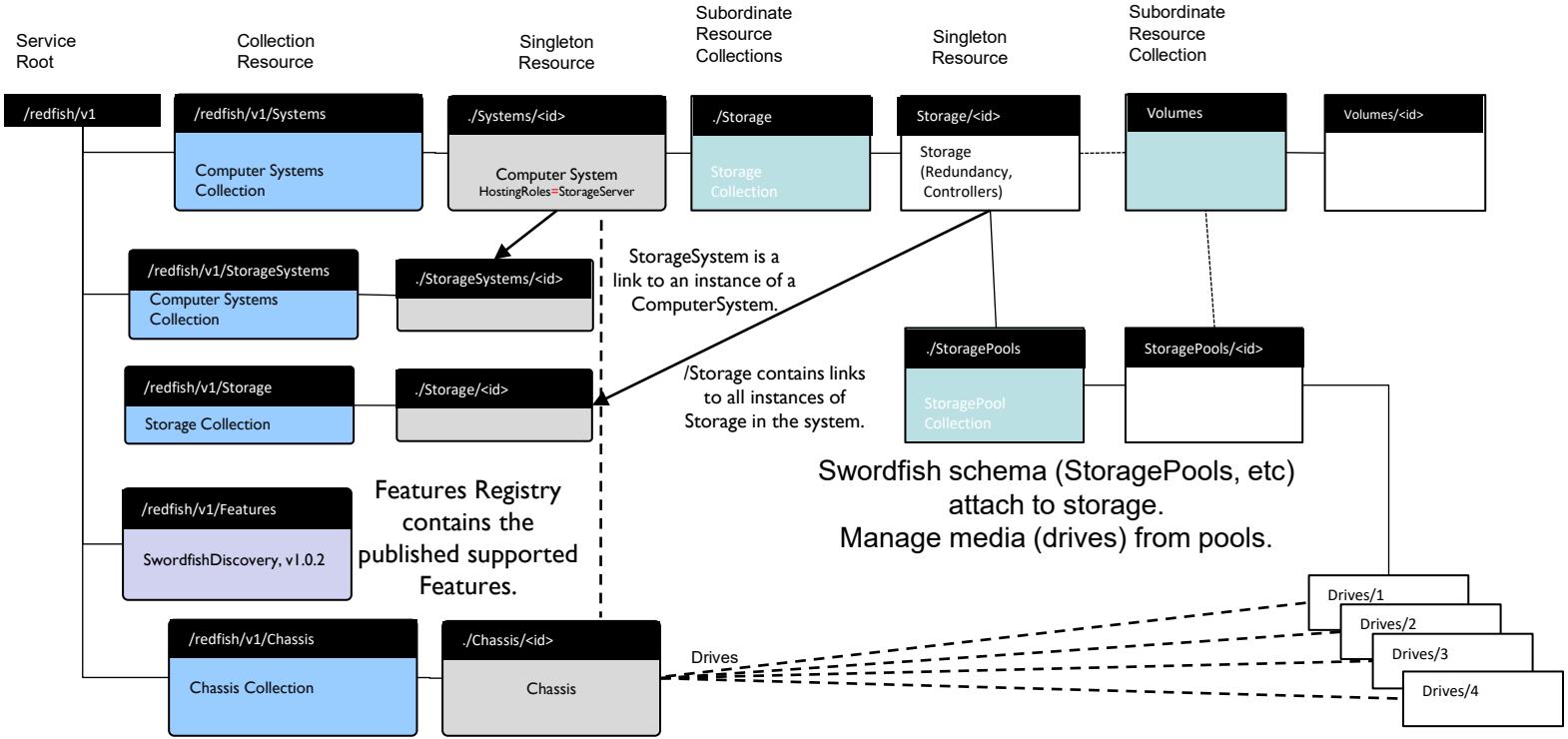


Volumes are in Collections off of the Storage resource, drives are in arrays off of the storage resource, and optionally, the Chassis.

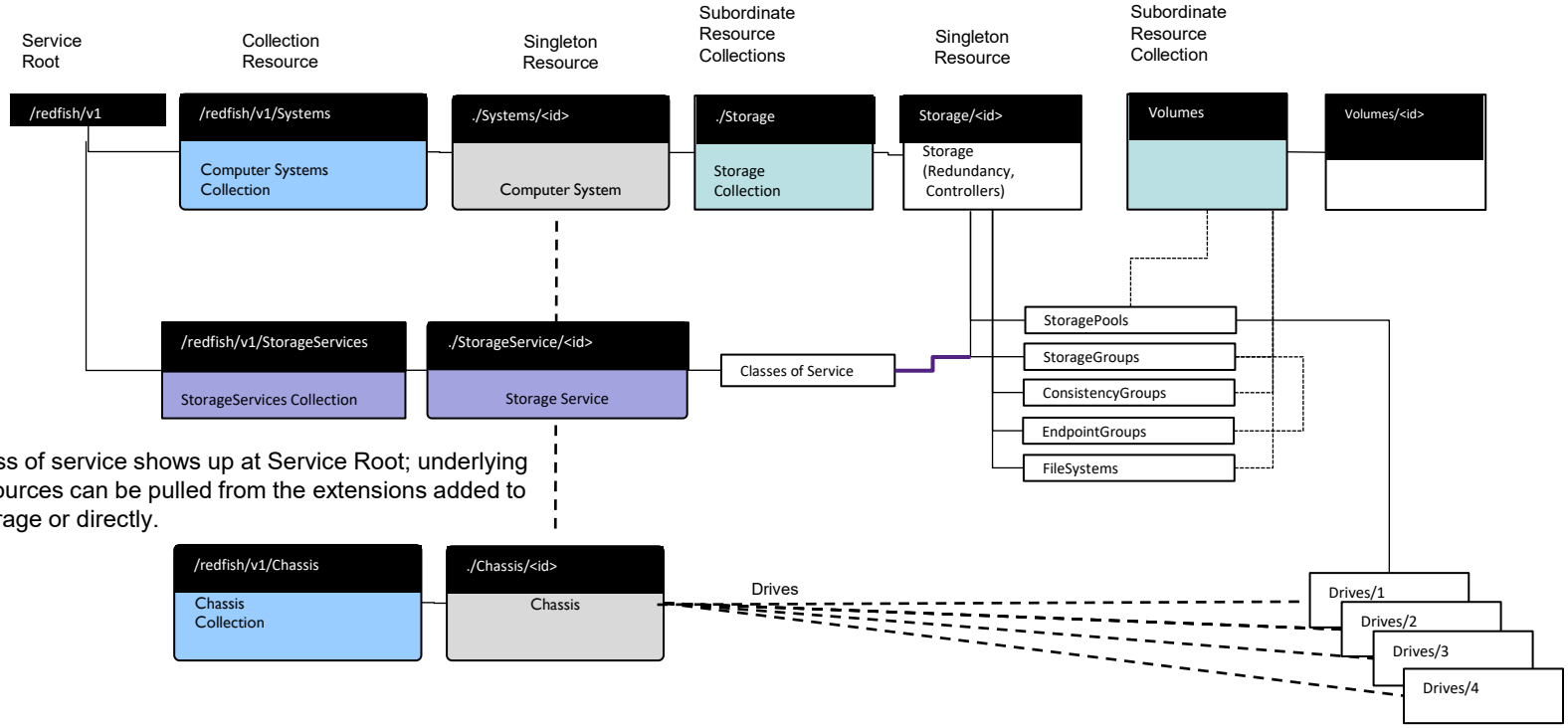
Add Swordfish: StoragePools



Example:
Minimum Swordfish – Integrated Configuration



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 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

Swordfish Elements

- Primary Elements
  - Volume: Block 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.
  - Filesystem: File-addressable storage.
  - Fileshare: A shared set of files with a common directory structure that is exported for use by remote systems.
  - Media: Drives / Memory – objects that provide storage capacity typically in replaceable units
- 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.

Swordfish: Walking the Model

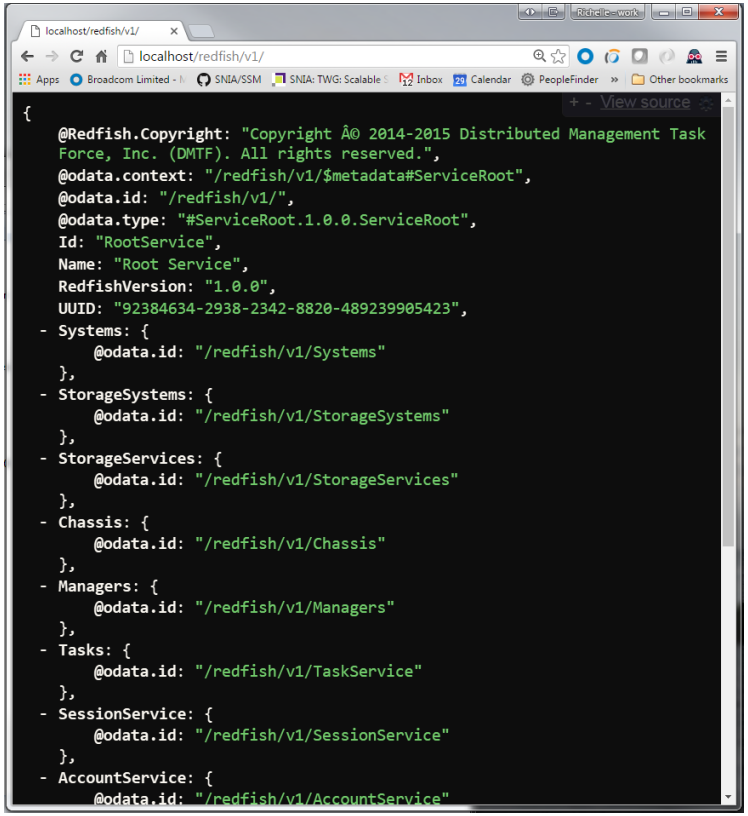
See Example Swordfish Configurations

- As a work tool, the Technical Work Group (TWG) works with “mockups” (snapshots of a state in time) of different types of systems
- Published at http://swordfishmockups.com (/redfish/v1/)

Note: Mockups are representations of implementations, not normative

Overview of Swordfish Hierarchy

- Explore the Swordfish data model to see potential / typical implementation
- Navigate through the model to learn about, and see, various resources
- SNIA mockups show examples of:
  - block storage system permutations - attached to servers, standalone (e.g., an external array), supporting replication, with class of service
  - file server with multiple file shares



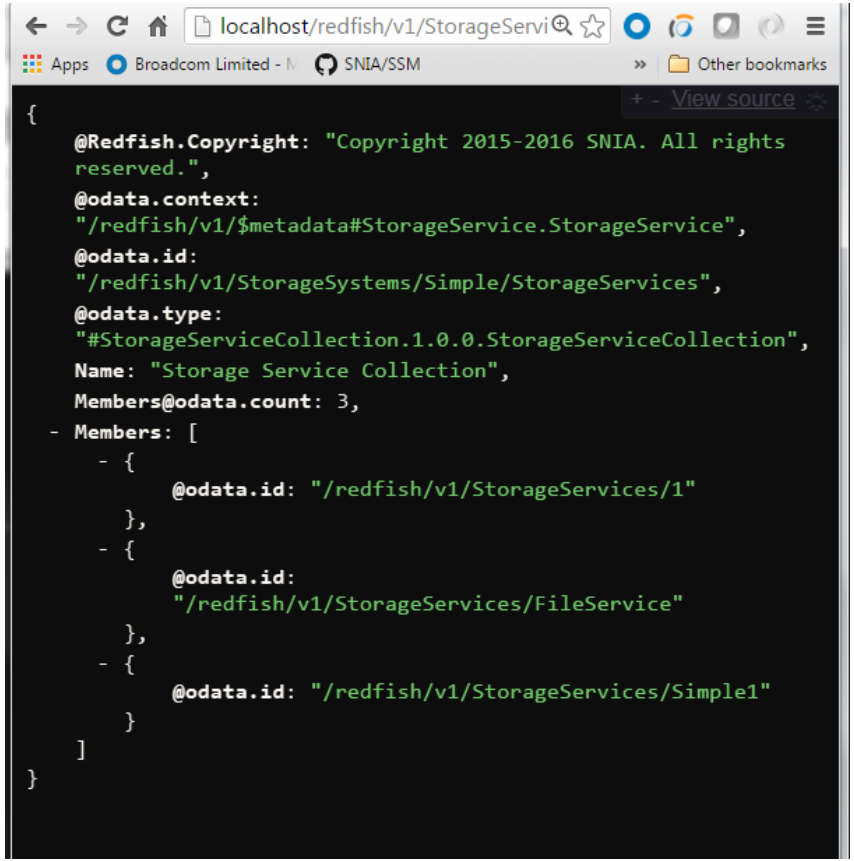
Navigating through the Mockups…

- Select the …./redfish/v1/StorageStorageServices link to see the “Collection” of Storage Services
- Click the “…/StorageServices/Simple” link to see the details of the Simple mockup

“…/StorageServices/1” to see the details of the complex storage service mockup

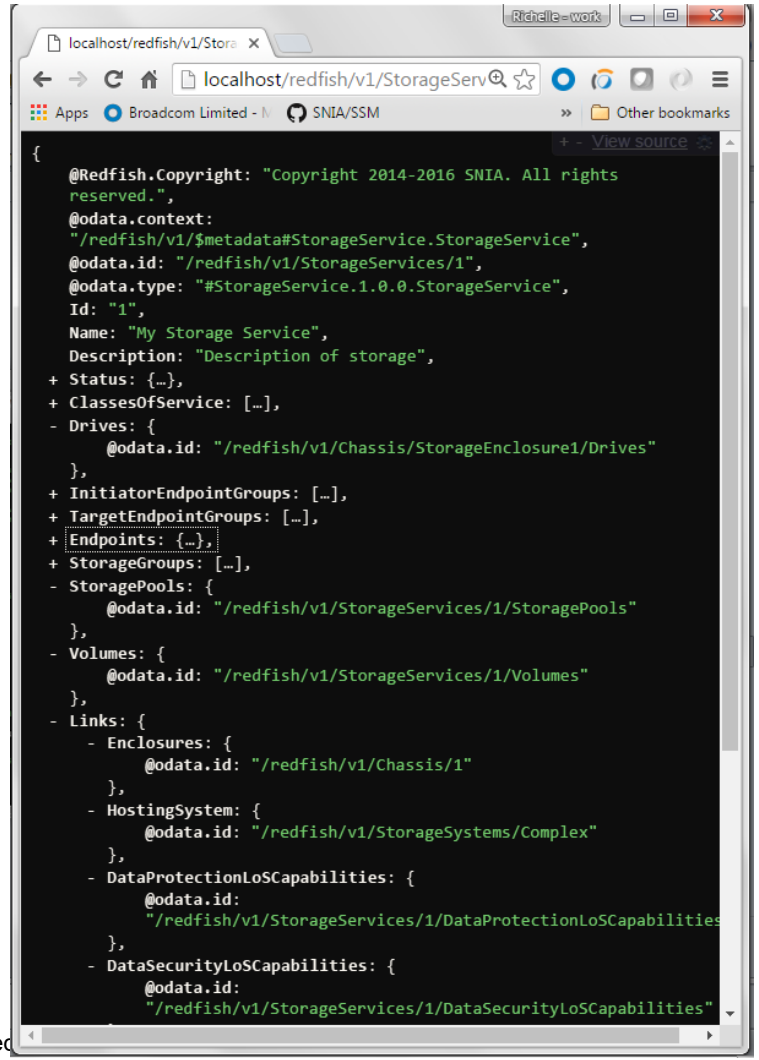
“…/StorageServices/FileService” to see the filesystem mockup

“…/StorageServices/ISC” to see the ISC mockup (look for links to the hosting system)



What's in a Storage Service? (Block)

- Available Classes Of Service
  - Lines of Service that are used to compose the Classes of Service
- Volumes
- Pools
- Groups
- Endpoints
- …
- Pointers to related resources (system, chassis,..)



What's in a Storage Service? (File)

Same structure:

- Available Classes Of Service
- ***File systems***
- Pools
- Groups
- Endpoints
- …
- Pointers to related resources (system, chassis, **block service** or drives)

Swordfish: Sample Use Case – Finding Capacity Information

Using Swordfish: Get Volume Capacity Information

Traverse the Service Root to find the selected volume and get its Capacity information:

- Read the Service Root resource
- Read the link to the Storage Collection
- Pick a Storage System
- Read the link to the Volume Collection
- Pick desired Volume:
  - Collect the Capacity information
  - Look at the consumed vs. allocated capacity information

Swordfish Volume Capacity Step 1: Read the Service Root

(Step 2: Read the link to the Storage Collection)

```
GET /redfish/v1/ HTTP/1.1

HTTP/1.1 200 OK
{
    "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
    "@odata.id": "/redfish/v1/",
    "@odata.type": "#ServiceRoot.v1_0_0.ServiceRoot",
    "Id": "RootService",
...
    "Storage": {"@odata.id": "/redfish/v1/Storage" },
    "StorageServices": {"@odata.id": "/redfish/v1/StorageServices" },
    "Chassis": {"@odata.id": "/redfish/v1/Chassis" },
...
    "Links": {
        "Sessions": {"@odata.id": "/redfish/v1/SessionService/Sessions" }
    },
}
```

Swordfish Volume Capacity Step 3: Pick a Storage System

```
GET /redfish/v1/StorageSystems HTTP/1.1

HTTP/1.1 200 OK
{
    "@odata.context": "/redfish/v1/$metadata#StorageCollection.StorageCollection",
    "@odata.id": "/redfish/v1/Storage",
    "@odata.type": "# StorageCollection.v1_0_0.StorageCollection",
    "Name": "Storage Collection",
    "Members@odata.count": 3,
    "Members": [
        { "@odata.id": "/redfish/v1/Storage/1" },
        { "@odata.id": "/redfish/v1/Storage/2" },
        { "@odata.id": "/redfish/v1/Storage/Simple1" }
    ]
}
```

Swordfish Volume Capacity Step 4: Read the Storage Entity and Find the Volumes Collection

```
GET /redfish/v1/Storage/1 HTTP/1.1

HTTP/1.1 200 OK
{
    "@odata.context": "/redfish/v1/$metadata#Storage.Storage",
    "@odata.id": "/redfish/v1/Storage/1",
    "@odata.type": "#Storage.v1_3_0.Storage",
    "Id": "1",
    "Name": "My Storage Controller",
...
    "Volumes": {
        "Members": [ { "@odata.id": "/redfish/v1/Storage/1/Volumes" } ]
    },
    "Drives": { … },
    "Links": { }
    ...
```

Swordfish Volume Capacity Step 5: Pick Desired Volume

```
GET /redfish/v1/Storage/1/Volumes HTTP/1.1

HTTP/1.1 200 OK
{
...
    "Name": "Volumes",
    "Members@odata.count": 6,
    "Members": [
        { "@odata.id":
          /redfish/v1/Storage/1/Volumes/6100123487654567610012348765456 7" },
        { "@odata.id":
          "/redfish/v1/1/Storage/1/Volumes/654567654567610012348761001234 87" },
        { "@odata.id": "/redfish/v1/Storage/1/Volumes/Volumes/3" },
        { "@odata.id": "/redfish/v1/Storage/1/Volumes/Volumes/4" },
        { "@odata.id": "/redfish/v1/Storage/1/Volumes/Volumes/5" },
        { "@odata.id": "/redfish/v1/Storage/1/Volumes/Volumes/6" }
    ]
}
```

Swordfish Volume Capacity Step 6: Look at Capacity Information

```
GET /redfish/v1/Storage/1/Volumes/6100123487654567610012348765456761001234876545676 HTTP/1.1

HTTP/1.1 200 OK
{
...
    "Id": "610012348765456761001234876545676",
...
 "Capacity": {
        "Data": {
            "ConsumedBytes": 0,
            "AllocatedBytes": 10737418240,
            "GuaranteedBytes": 536870912,
            "ProvisionedBytes": 1099511627776
        },
        "Metadata": {
            ...
        },
        "Snapshot": {
            ...
       }
    }
}
```

Swordfish Configurations

Swordfish Configurations

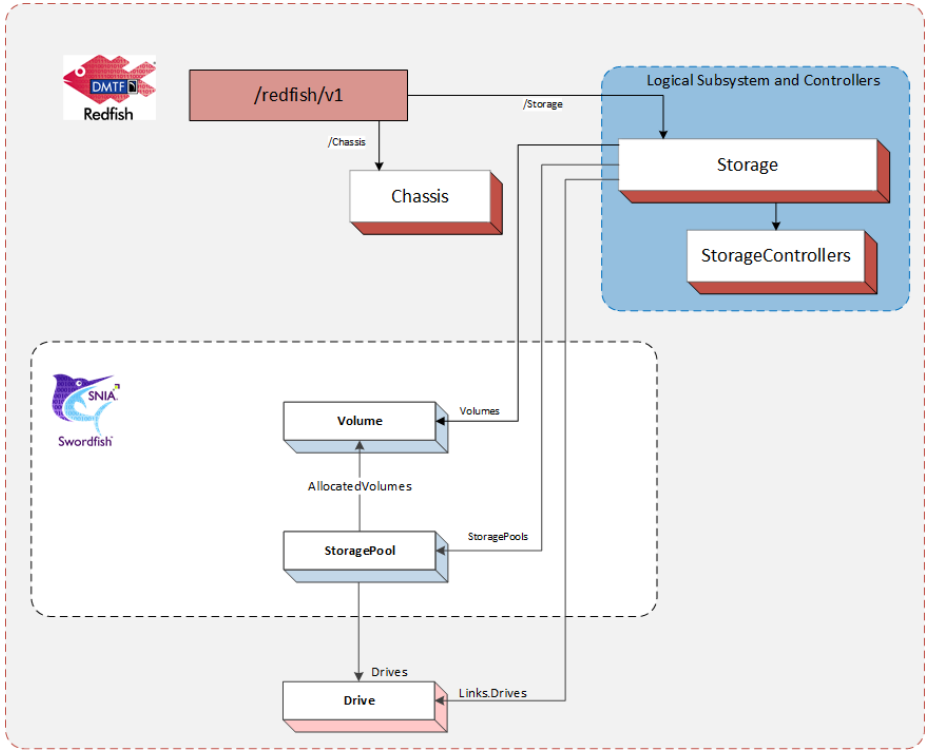
- Due to large variation in potential implementations, there are multiple defined Swordfish Configuration options:

- Standalone Swordfish Configuration
  - Use for external storage devices, shared storage (e.g., fabric attached)
- Integrated Swordfish Configuration
  - Use when storage device is attached to a (single) computer system

- Service-based Configurations
  - Hosted Service Configuration
    - Use when the storage device both implements service-based features, and is hosted by a storage resource within a computer system (e.g., attached to a computer system, or a software-defined storage implementation)
  - Standalone Service Configuration
    - Use for external or fabric attach storage systems that implement service-based features

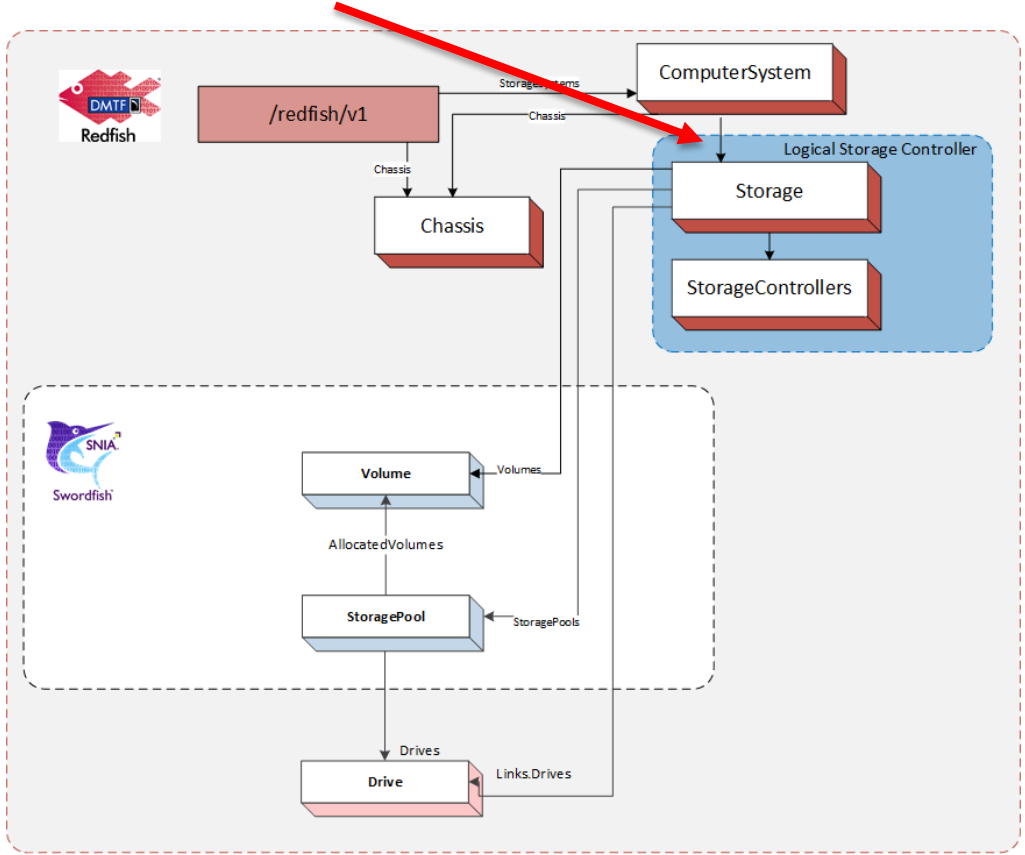
Standalone Swordfish Configuration

- The storage device is instantiated as a Logical Subsystem and Controllers using Storage and StorageControllers in /redfish/v1/Storage
  - Capacity is represented using StoragePools – collections of media
  - Volumes (or FileSystems / FileShares) are created from StoragePools
- Clients can always find Swordfish storage instances in /redfish/v1/Storage



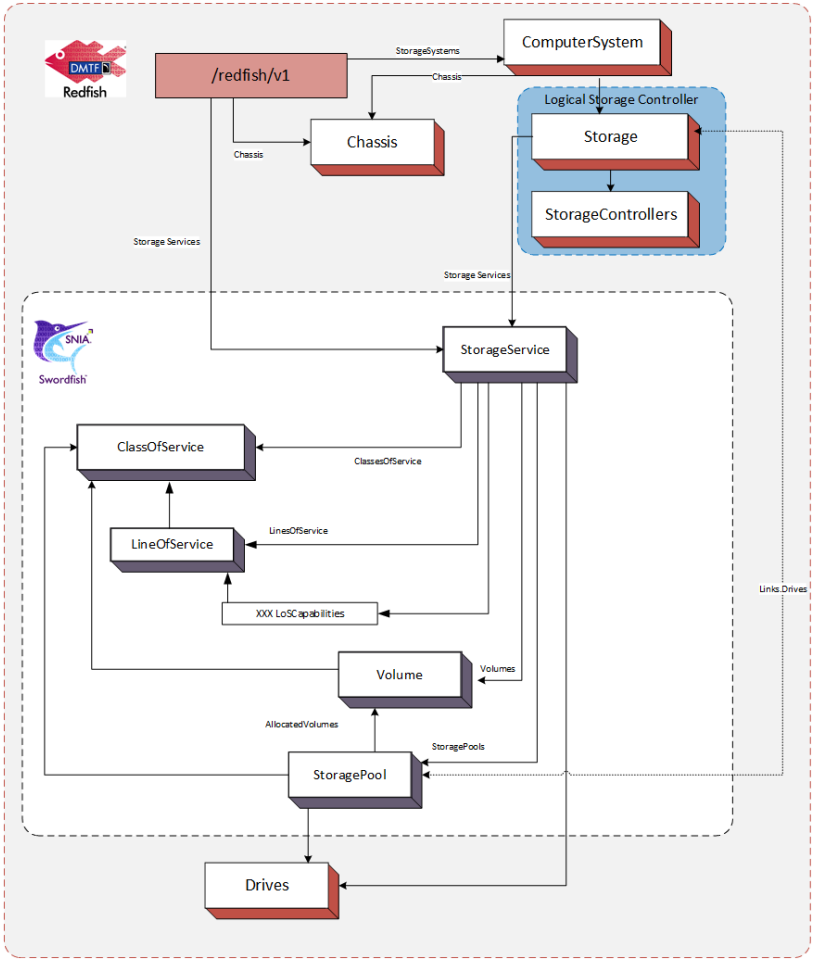
Integrated Swordfish Configuration

- The storage device is instantiated as a Logical Subsystem and Controllers using Storage and StorageControllers attached to a ComputerSystem
  - A reference link to this Storage instance is provided in /redfish/v1/Storage
  - The same sub-objects are instantiated / available, regardless of the type of configuration
- Clients can always find Swordfish storage instances in /redfish/v1/Storage



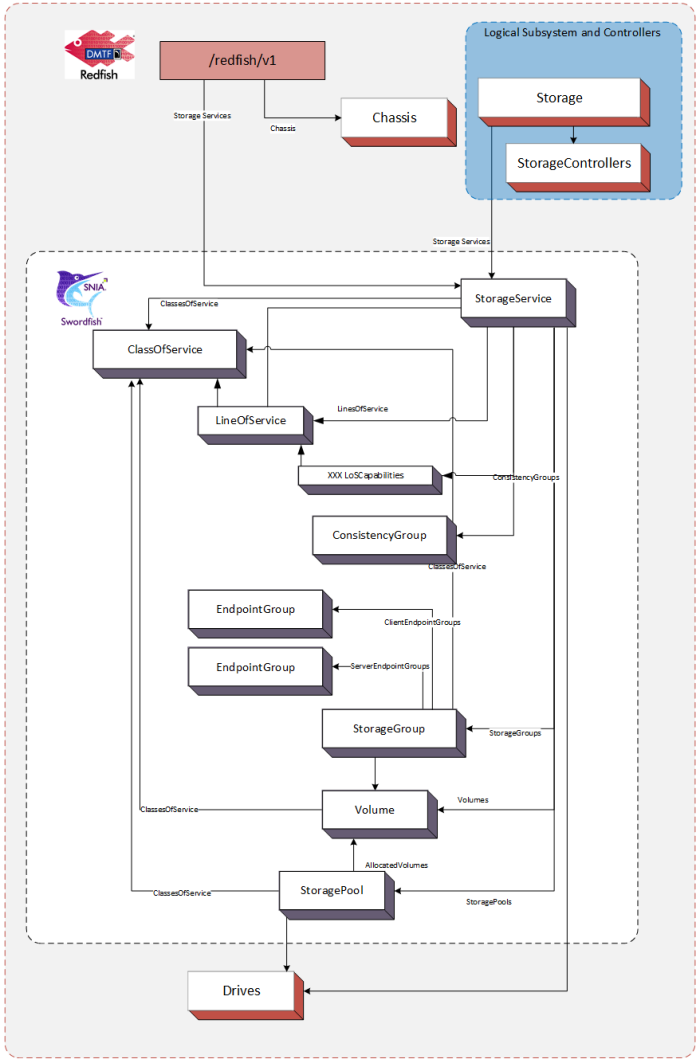
Hosted Service Configuration

- The storage device both implements service-based features, and is hosted by a storage resource within a computer system
- Clients can find all Storage Service instances in /redfish/v1/StorageServices
  - /redfish/v1/Storage can still be used to locate all instances of storage systems
  - May have a m:n (usually m:1) relationship



Standalone Service Configuration

- The storage device both implements service-based features, and is instantiated in /redfish/v1/Storage
  - /redfish/v1/Storage is still used to locate all instances of storage systems
- Clients can find all Storage Service instances in /redfish/v1/StorageServices
  - May have a m:n (usually m:1) relationship



Developing with Swordfish: Multiple Options Available

- Schema available in multiple formats:
  - XML - CSDL
  - JSON
  - YAML / OpenAPI
  - RDE Dictionaries – used as embedded schema information for RDE implementations

Note: If there are any discovered discrepancies, the XML version shall be used as reference. Other versions are generated from the XML version.

Implementing Swordfish: Schema Options

Quick Compare: (XML / CSDL) Same Object, Different Schema Format

```
<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Example">
  <Annotation Term="Redfish.OwningEntity" String="SNIA"/>
  <EntityType Name="Example" BaseType="Resource.v1_0_0.Resource" Abstract="true">
     <Annotation Term="OData.Description" String="An example entity type with example property."/>
     <Annotation Term="OData.LongDescription" String="An example entity type with an example property."/>
   </EntityType>
</Schema>

<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Example.v1_0_0">
  <Annotation Term="Redfish.OwningEntity" String="SNIA"/>
  <Annotation Term="Redfish.Release" String="WIP v1.2.1"/>
  <EntityType Name="Example" BaseType="Example.Example">
     <Annotation Term="OData.Description" String="An example entity type with example property."/>
     <Annotation Term="OData.LongDescription" String="An example entity type with example property."/>
     <Property Name="IsExample" Type="Edm.Boolean">
        <Annotation Term="OData.Permissions" EnumMember="OData.Permission/Read"/>
        <Annotation Term="OData.Description" String="This value is true when the property is set to true."/>
       <Annotation Term="OData.LongDescription" String="The value of this property shall be set to true when the
property is set to true."/>
     </Property>
   </EntityType>
</Schema>
```

Quick Compare: (JSON)
Same Object, Different Schema Format

```
{
   "$id": "http://redfish.dmtf.org/schemas/swordfish/v1/Example.v1_0_0.json",
   "$ref": "#/definitions/Example",
   "$schema": "http://redfish.dmtf.org/schemas/v1/redfish-schema-v1.json",
   "copyright": "Copyright 2016-2019 Storage Networking Industry Association
(SNIA), USA. All rights reserved. For the full SNIA copyright policy, see
http://www.snia.org/about/corporate_info/copyright",
   "definitions": {
      "Example": {
         "additionalProperties": false,
         "description": "An example entity type with example property.",
         "longDescription": "An example entity type with example property.",
         …
            }
         },
         "properties": {
          …
            "IsExample": {
               "description": "This value is true when the property is set to true.",
               "longDescription": "The value of this property shall be set to true
when the property is set to true..",
               "readonly": true,
               "type": [
                  "boolean",
                  "null"
               ]
            },
         …
            "required": [
               "@odata.id",
               "@odata.type",
               "Id",
               "Name"
            ],
            "type": "object"
      }
   },
   "owningEntity": "SNIA",
   "release": "WIP v1.2.1",
   "title": "#Example.v1_0_0.Example"
}
```

Quick Compare: (YAML)
Same Object, Different Schema Format

```
components:
  schemas:
    Example_v1_0_0_Example:
      additionalProperties: false
      description: An example entity type with example property.
      properties:
        ….
        IsExample:
          description: This value is true when the property is set to true.
          nullable: true
          readOnly: true
          type: boolean
          x-longDescription: The value of this property shall be set to true when
            the property is set to true..
      …
      required:
      - '@odata.id'
      - '@odata.type'
      - Id
      - Name
      type: object
```

```
x-longDescription: An example entity type with example property.
    x-patternProperties:
      ^([a-zA-Z_][a-zA-Z0-9_]*)?@(odata|Redfish|Message)\.[a-zA-Z_][a-zA-Z0-
9_]*$:
        description: This property shall specify a valid odata or Redfish property.
title: '#Example.v1_0_0.Example'
x-copyright: Copyright 2016-2019 Storage Networking Industry Association
(SNIA), USA.
  All rights reserved. For the full SNIA copyright policy, see
http://www.snia.org/about/corporate_info/copyright
x-owningEntity: SNIA
x-release: WIP v1.2.1
```

Swordfish Resources

Swordfish Info: www.snia.org/swordfish

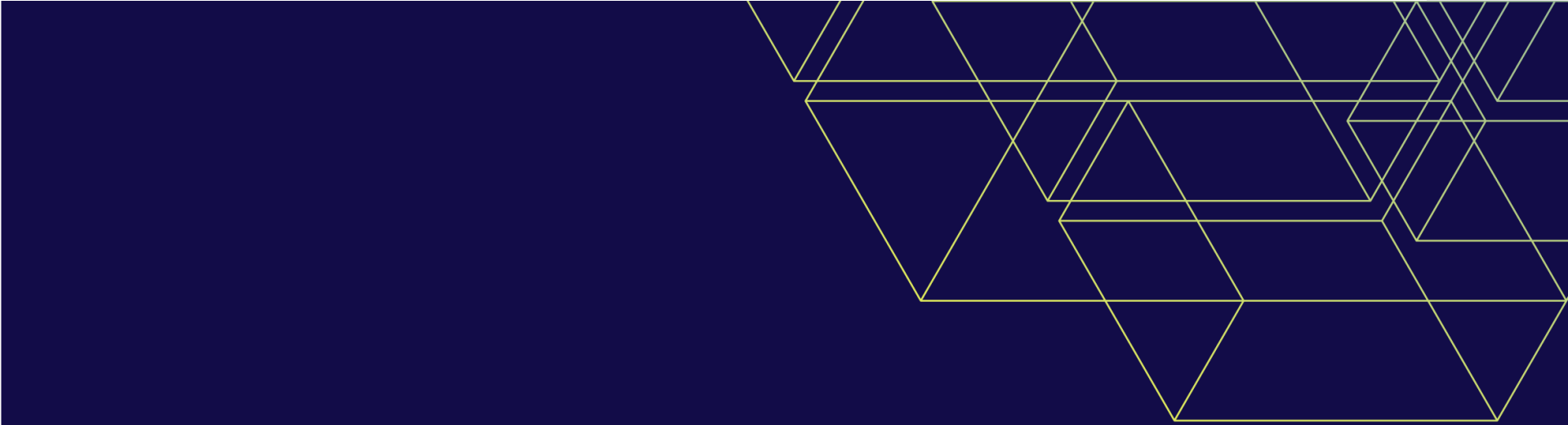
- Resources
  - Specifications
  - User's Guide
  - GitHub for Swordfish Tools
  - Practical Guide
  - Other Documentation
- Swordfish Mockups Site
  - Traditional configs:
  - Service-based
  - Block vs file configurations
  - Small and large configurations
  - NVMe and NVMe-oF
- Education/Community
  - Whitepapers, Presentations
  - YouTube shorts & Webinars
- Participate
  - Join SNIA and the SSM TWG
  - Implement
- Conformance Program

Open Source Tools and Infrastructure Development

- Available: 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
  - Swordfish Powershell Toolkit
    - Powershell toolkit integration for Windows and Linux

Thank you for watching

- SNIA Swordfish™ Standards
  - Schemas, Specs, Mockups, Users Guide, Practical Guide & more https://www.snia.org/swordfish
- Redfish / Swordfish Specification Forum
  - This is where you can ask and answer questions about Redfish and Swordfish
  - http://swordfishforum.com/
- Scalable Storage Management (SSM) TWG
  - Technical Work Group that defines Swordfish
  - Influence the next generation of the Swordfish standard
  - Join SNIA and participate: https://www.snia.org/member_com/join-SNIA
- Join the SNIA Storage Management Initiative
  - Unifies the storage industry to develop and standardize interoperable storage management technologies
  - https://www.snia.org/forums/smi/about/join



Please take a moment
to rate this session.

Your feedback matters to us.