STORAGE DEVELOPER CONFERENCE
SDC 23
BY Developers FOR Developers

Survey of Redfish Open-source Implementations

John Leung, Intel, Principal Engineer

DMTF and Redfish Service Implementations

- DMTF develops interfaces standards which expose manageability (e.g. Redfish)
- DMTF policy is to be implementation neutral - thus encouraging innovation below the interface. (DMTF Alliance Partners may not have such a policy)
- Redfish service implementations exists as firmware and as hosted software agents

Implementation	Language	Auto-generated using	Repo owner
OpenBMC	C++ (firmware)		Linux Foundation
Swordfish Emulator	Python (SW agent)	mockup, CSDL schema	SNIA
Redfish Service Framework	Java	mockup, OpenAPI schema	PICMG/ASU
PSME	C++ (SW agent)		OCP
Device Manager	GoLang (SW agent)		OCP

OpenBMC Project

- A Linux Foundation project whose goal is to produce a customizable, open-source firmware stack for Baseboard Management Controllers (BMCs).
  - A BMC is a specialized controller embedded on the baseboard which may be operational when the rest of the baseboard is not (out-of-band)
- Features:
  - Uses the Yocto Project as the underlying building and distribution generation
  - Uses D-Bus as an inter-process communication (IPC)
  - Includes a web application for interacting with the firmware stack (WebUI)

https://openbmc.org (LF in 2018)
https://github.com/openbmc/openbmc/wiki

Software Agent Implementation - **Python**

- Redfish Emulator (2016)
  - Simple simulators: Mockup-Server (get) and Profile-Simulator (patch)
  - Emulator envisioned for rapidly prototyping of new interface behavior
  - Includes code generator for emulator stubs
  - Cloud Foundry deployable
- Swordfish Emulator (2018)
  - SNIA extends modelling any resource
  - Used to prototype managing NVMe-over-Fiber
- OpenFabrics Mgmt Frmwk (2021)
  - Base from the framework is being developed
- NVMe Redfish Service on Linux, RHEL (2023)

github.com/dmtf/Redfish-Interface-Emulator
github.com/snia/Swordfish-API-Emulator

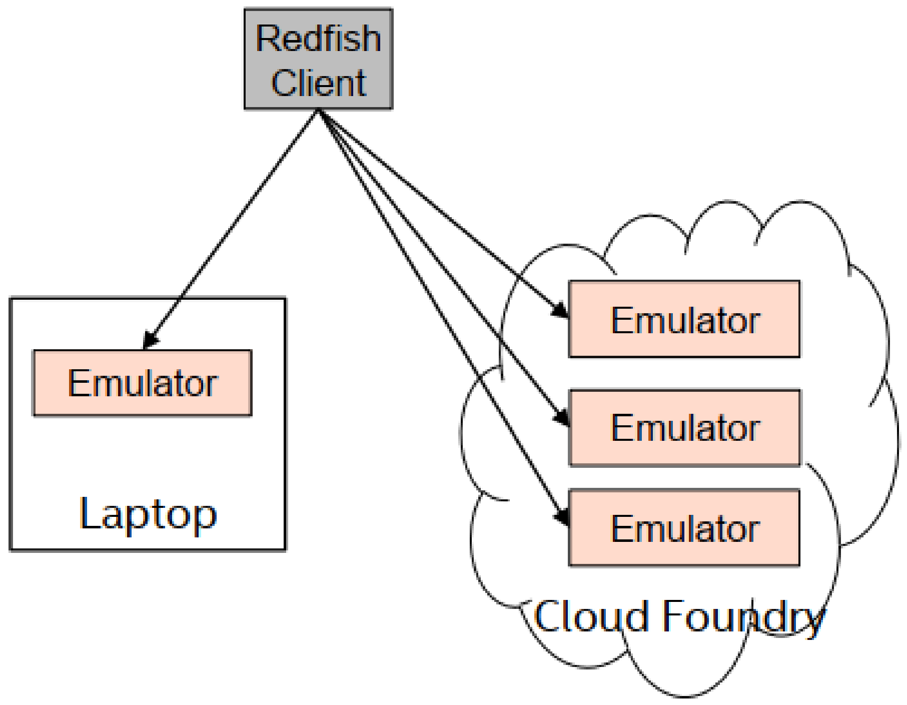
Swordfish Service Capabilities

Redfish Emulator

- Multiple methods of instantiation - hosted as standalone, in a cloud foundry, or as docker container
- Dynamic emulation of schema objects (circa 2016)
- Auto-generate Python code from mockups

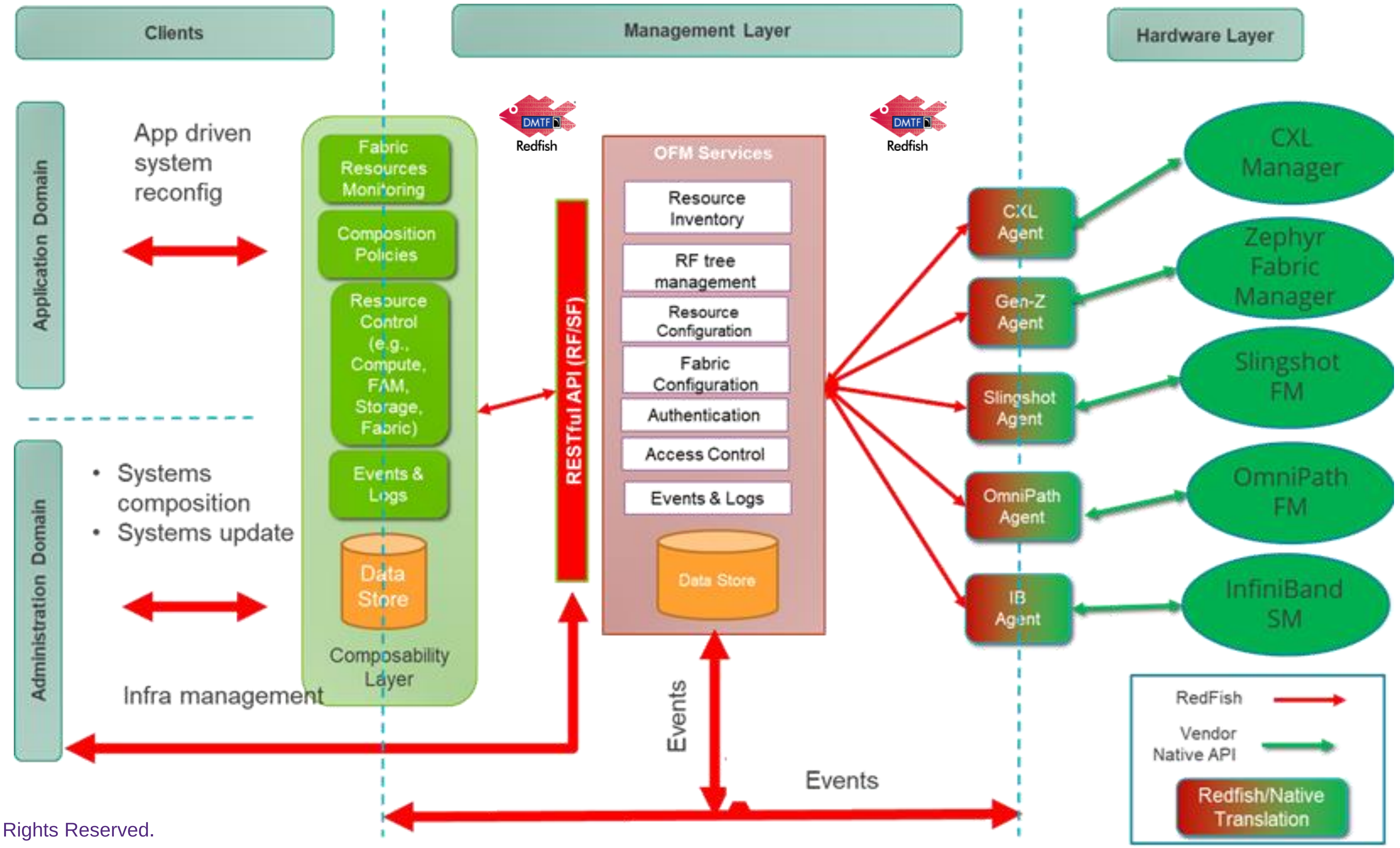
Swordfish Emulator

- Dynamic emulation of **all** schema objects in **all** URI locations
- Auto-generate Python code from any valid Redfish conformant schema
- Supports basic auth, sessions, certificates
- Supports Redfish services: Account (Admin only), Event (WIP), Session
- Service targeted to pass SNIA conformance tests



OpenFabrics Management Framework

https://www.openfabrics.org/openfabrics-management-framework



NVMe Redfish Service on Redhat RHEL Linux

- Start with Swordfish emulator framework
  - RedfishF/Swordfish structure: service, metadata
  - Core services: account service, session service, event service
  - Stubs and template for all object types
  - Persistent database
- Added
  - Startup / discovery
  - Fills in stubs for relevant objects (removes all unneeded objects)
- Availability
  - In the process of making public

Software Agent Implementation - Java

- PICMG is extending the Redfish model to support Industrial IoT
  - "PICMG Announces Significant Progress of IoT.X Family of Sensor Data Modeling and Abstraction Specifications"
- Redfish Service Framework
  - Functional Redfish Service generated from Redfish mockups and schema (OpenAPI)
  - Static behavior is auto-generated. Special behaviors like actions are stubbed out - when one implements the server, the actions can be coded
  - Implemented by ASU students so that we could play with customizing the dynamic features for our new objects.

https://www.picmg.org/picmg-announces-significant-progress-of-iot-x-family-of-sensor-data-modeling-and-abstraction-specifications/

Software Agent Implementation - **C++**

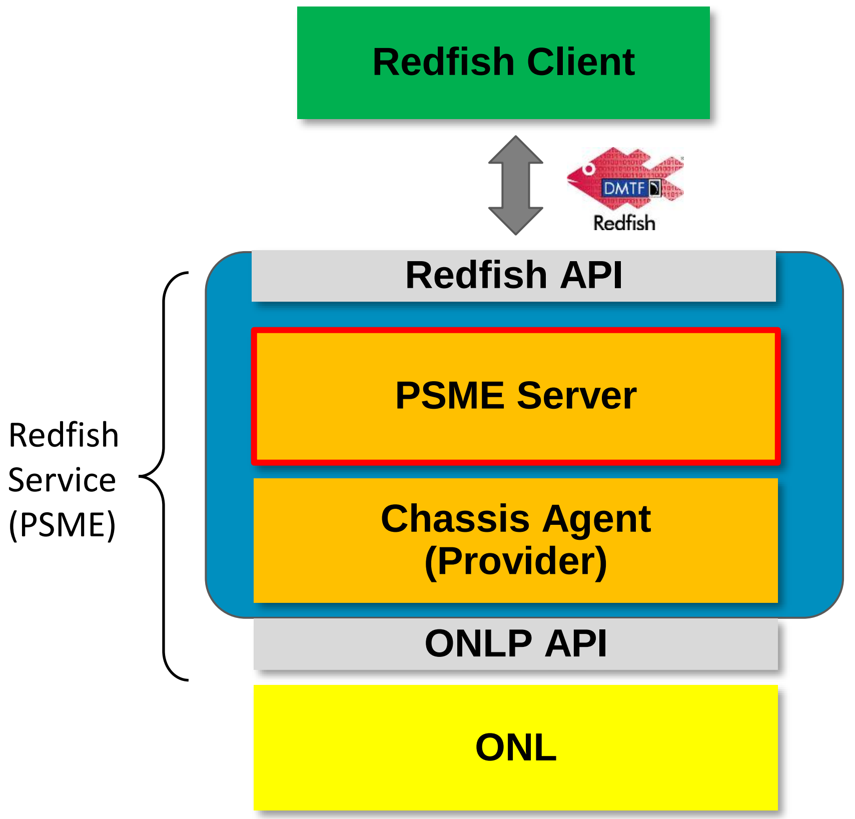
- OCP Rack Scale PSME (2019)[1]
  - Contribute by Intel (Pooled System Management Engine)

- OCP Device Manager PSME (2021)[2]
  - Contributed by Edgecore Networks (2021) - along with Device Manager
  - Celestica fixes build and fixes issues with baseline profile conformance (2022)
  - Celestica contributed a design spec for a platform which would fulfill the manageability requirements via the software agent

[1]github.com/opencomputeproject/Rack-Manager
[2]github.com/opencomputeproject/HWMgmt-DeviceMgr-PSME
3DS1000 ECS 1Gb Enterprise Switch

PSME Redfish Service

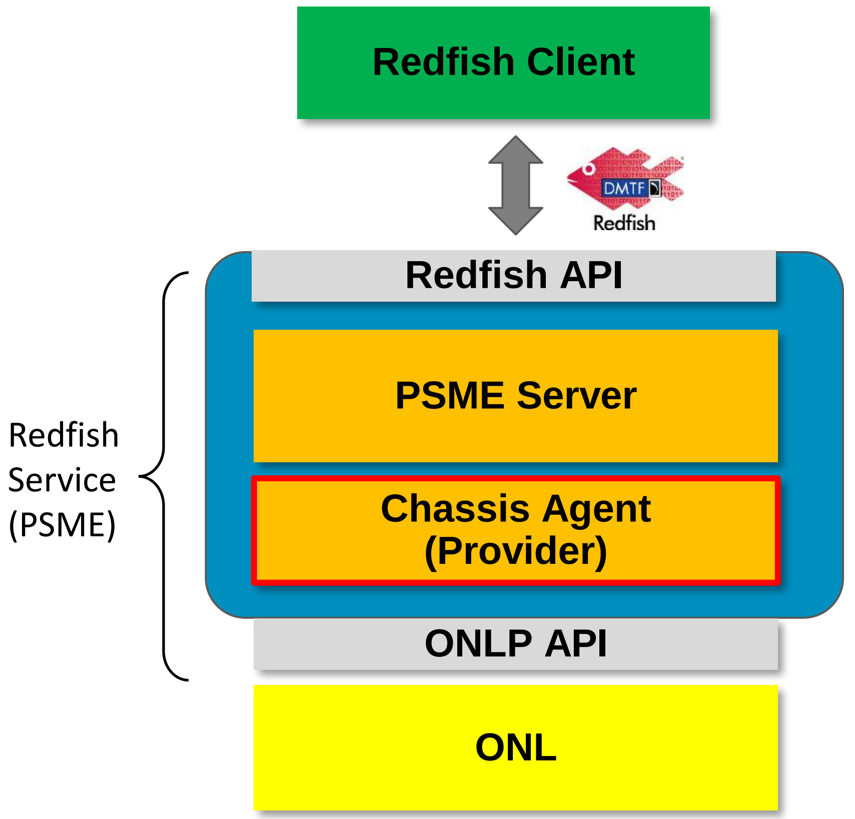
- PSME executes on ONL (Open Network Linux)[1]
  - Compose of PSME Server and Chassis Agent
- The PSME Server supports the Redfish interface and model
  - Requests information from and invokes actions on resources
- The PSME Server supports the Event Service
  - Redfish Clients may launch an event listener and subscribe for events to be sent to that event listener
  - Supports subscriptions for the ResourceAdd, ResourceRemove, and Alert type events



[1]https://github.com/opencomputeproject/OpenNetworkLinux

Chassis Agent

- Gathers peripheral information about thermal/fan/PSU/port transceiver statistics through ONLP API[1]
- Sends to PSME Server
- While gathering peripheral information, agent will check for posted event, and send these events to the PSME Server.



[1]**Open Network Linux Platform APIs** provide a common, consistent abstraction interface for accessing important platform assets such as SFPs, PSUs, Fans, Thermals, LEDs, and ONIE storage devices.

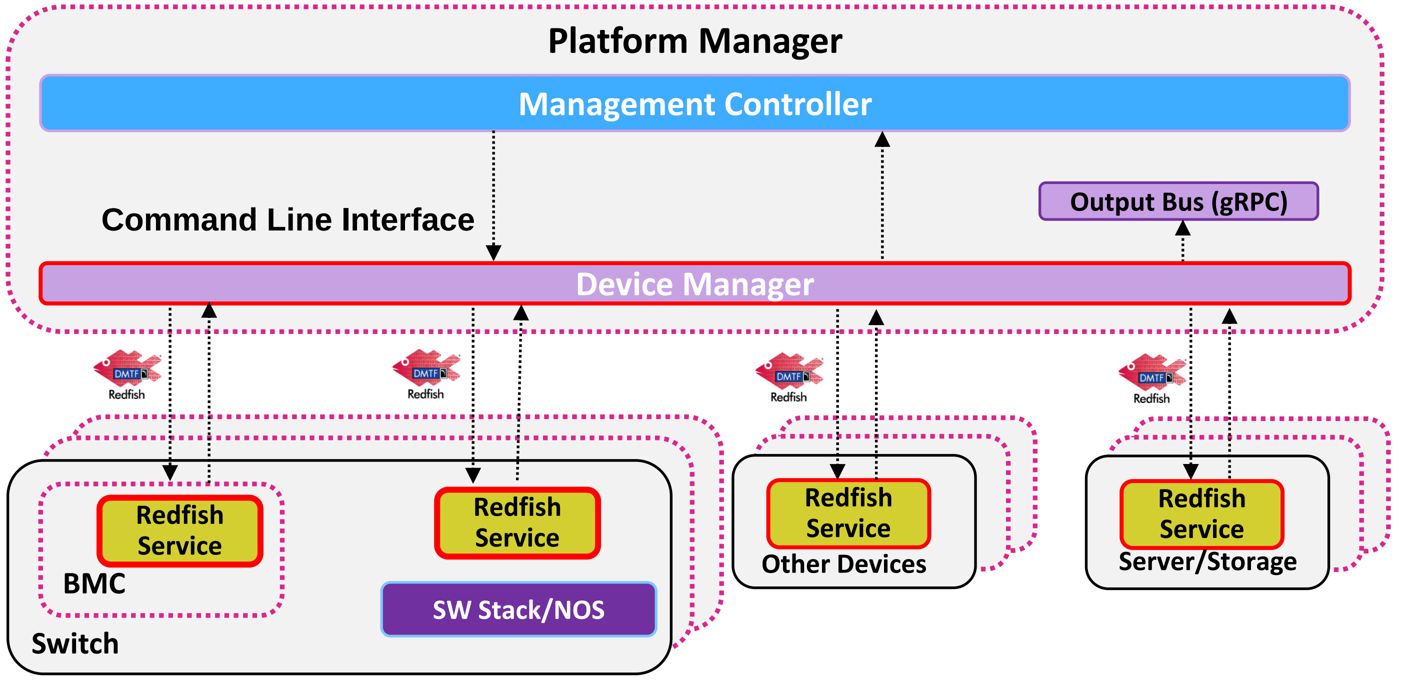
Software Agent Implementation - **GoLang**

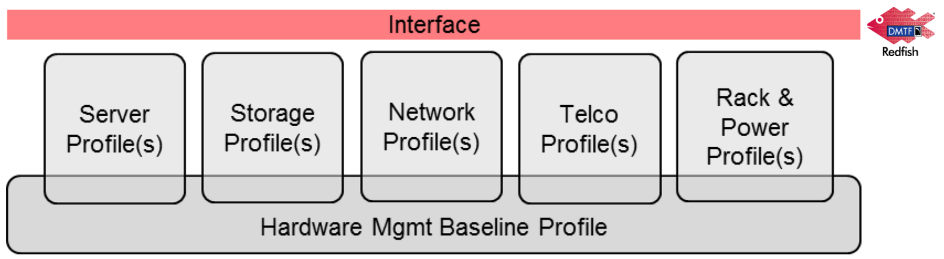
- **OCP Device Manager**
  - EdgeCore Networks (2021)
    - Contribution to OCP - bundled with Edge-core's PSME
    - Build of source was problematic
  - Intel (2022-2023)
    - Cleaned up the build
    - Added Redfish northbound interface
- **Linux Foundation ODIM (Open Distributed Infrastructure Mgmt)**

https://github.com/opencomputeproject/HWMgmt-DeviceMgr-DeviceManager
https://odim.io

Device Manager initial contribution

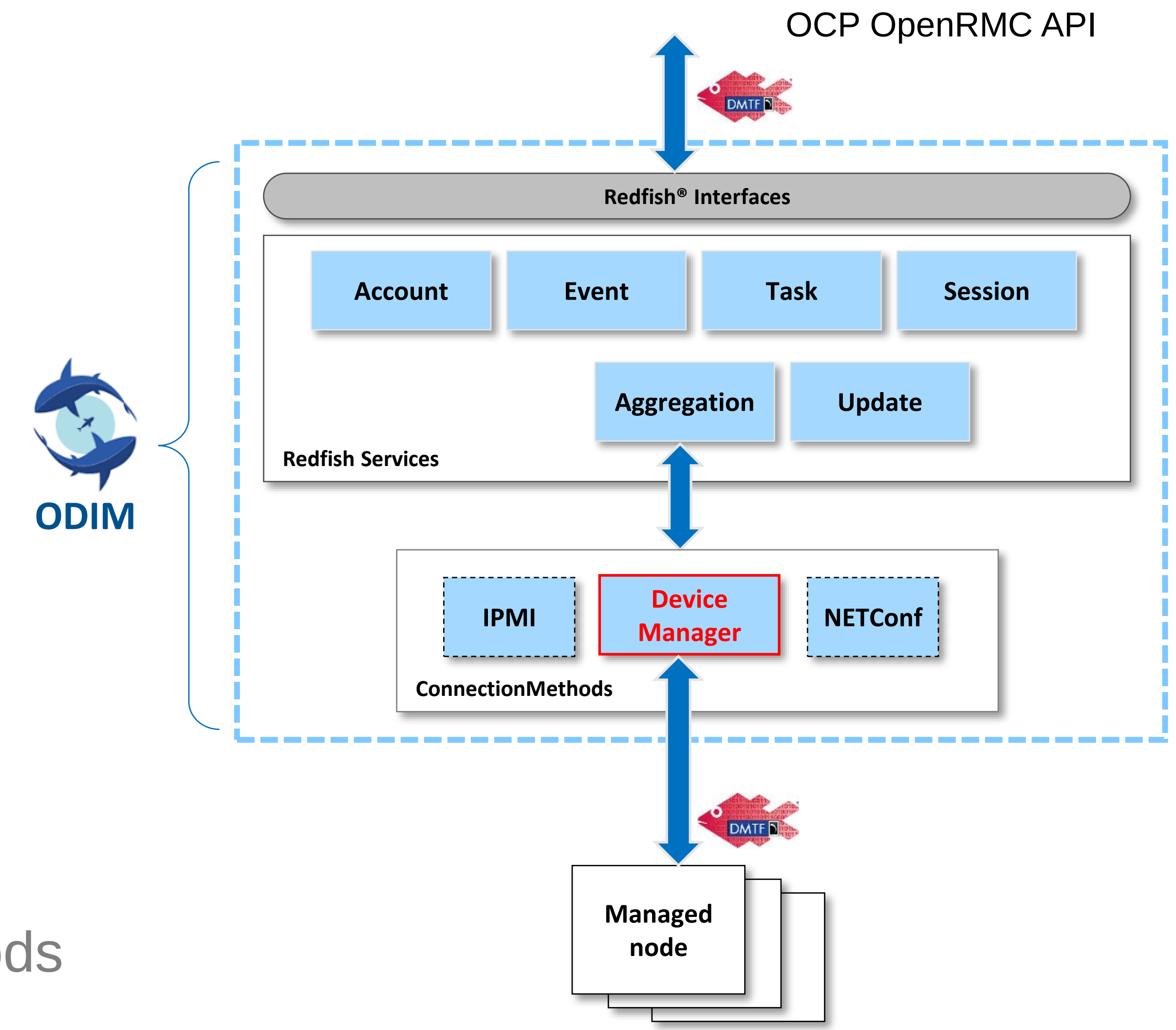
- Developed to manage platforms hosting Redfish Service(s) conformant to the OCP Baseline profile
- Exposed a command line interface





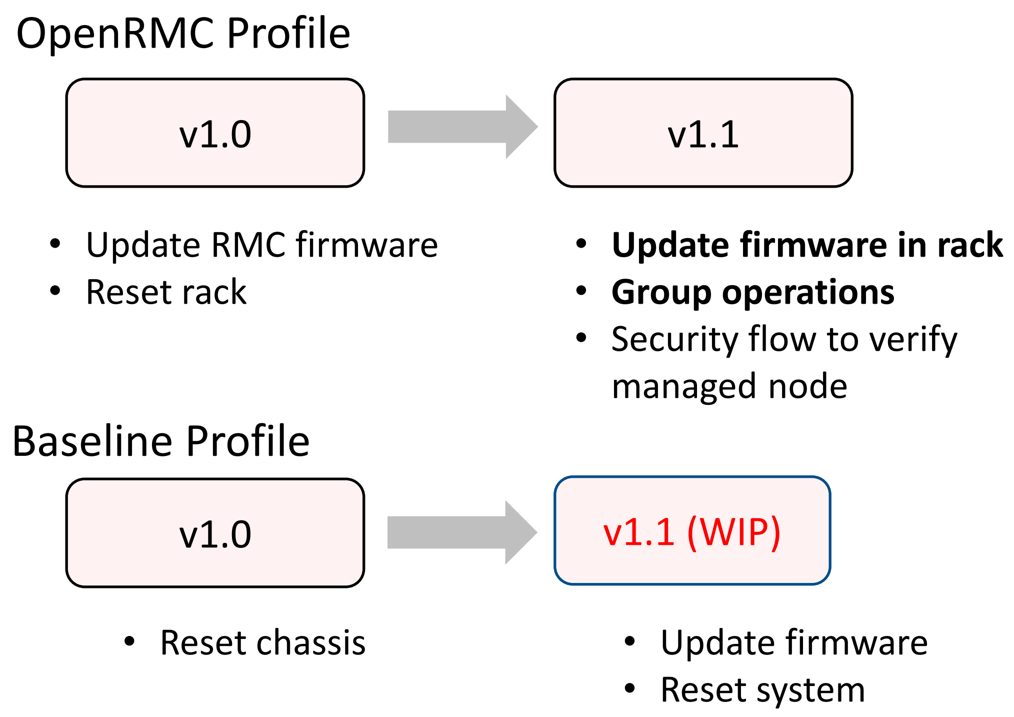
Adding Redfish interface for Device Manager

- OCP rack manager API (OpenRMC) required a northbound Redfish interface
- ODIM had a Redfish interface implementation for managing a distributed infrastructure (GoLang)
  - Account, Event, Task, and Session base services
  - Aggregation (group operations) and Update services (firmware)
  - Plugin architecture for connection methods



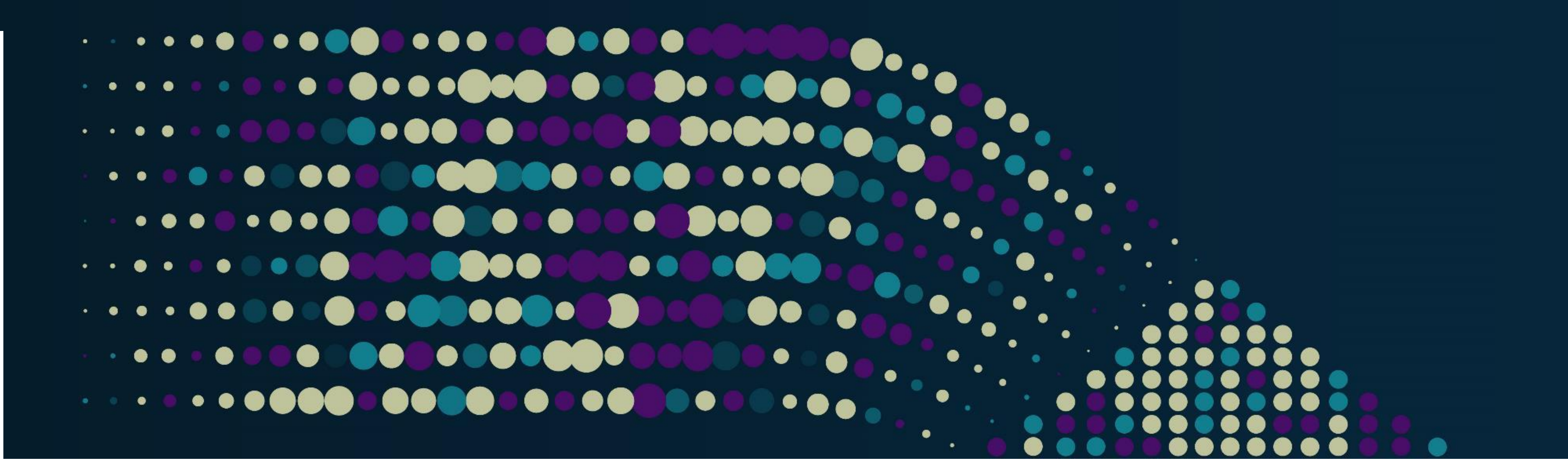
Next steps for Device Manager

- Conform to OpenRMC v1.0, then v1.1[1]
- Conform to Redfish Conformance Suite
  - Service-Validator
  - Protocol-Validator
- Support future revisions of OpenRMC rack manager interface be specified in the OpenRMC-DM subproject[2]



[1]Contributed OpenRMC specifications - https://www.opencompute.org/contributions?query=openrmc%20usage
[2]OpenRMC-DM subproject - https://www.opencompute.org/projects/openrmc-dm

Summary



Please take a moment to rate this session.

Your feedback is important to us.