



# Power API and Redfish: Standardizing Power Measurement and Control for HPC

Torsten Wilde (HPE), Ryan Grant (Sandia NL),  
Jeff Autor (HPE), Todd Rosedahl (IBM), Steve  
Martin (Cray), Barry Rountree (LLNL)

BoF: Power API and Redfish: Standardizing Power Measurement & Control for HPC - SC18

SC18 Dallas, Texas

<https://eehpcwg.llnl.gov/>

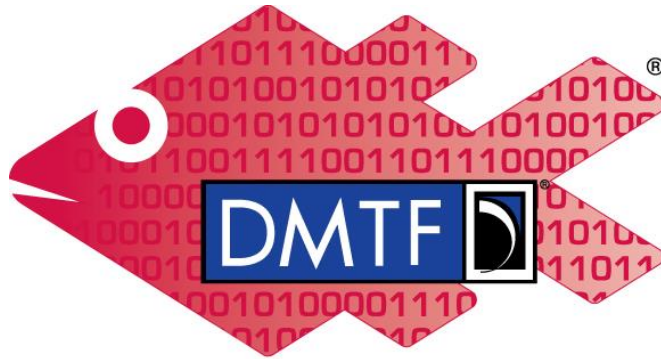
# Agenda

- **Short overview of Redfish and PowerAPI**
- **Guided discussion**

# How do Redfish and PowerAPI fit together

- **Redfish** – Vendor agnostic software defined management interface to access cluster hardware information (monitoring, control) in a standardized way (IPMI-over-LAN replacement)
- Defines interfaces and data representations (REST-full interface: get , post, delete, and patch)
- Designed for 1s or slower data access (out-of-band management network)
- Definition community driven
- Underlying implementation vendor specific
- **PowerAPI** – Vendor agnostic API to access cluster power information (monitoring, control) in a standardized way (facility concerns down to low level software/hardware interfaces, e.g. get application power consumption from scheduler)
- Defines function calls and data representations related to system wide power and energy management
- Designed for high speed access in ms (in-band and out-of-band)
- Definition community driven
- Underlying implementation vendor specific
- Power API implementation could use Redfish to access hardware power measurements

# Redfish



# Redfish

# What is Redfish?

- Industry Standard Software Defined Management for Converged, Hybrid IT
  - HTTPS in JSON format based on OData v4
  - Schema-backed but human-readable
  - Equally usable by Apps, GUIs and Scripts
  - Extensible, Secure, Interoperable
- Version 1 focused on Servers
  - A secure, multi-node capable replacement for IPMI-over-LAN
  - Represent full server category: Rackmount, Blades, HPC, Racks, Future
  - Intended to meet OCP Remote Machine Management requirement
- Expand scope over time to rest of IT infrastructure
  - Additional features coming out approximately every 4 months
  - Working with SNIA to cover more advanced Storage (Swordfish)
  - Working with The Green Grid & ASHRAE to cover Facilities (Power/Cooling)
  - Work with the IETF to cover some level of Ethernet Switching

# Redfish 2018 Release 2

- Redfish Specification v1.6.0
  - NEW OpenAPI v3.0 Support
  - NEW Telemetry Streaming & Eventing
  - Improved Event Subscription methods
- Redfish Specification v1.5.1
  - Errata release with numerous clarifications
- Redfish Schema Bundle 2018.2
  - **DSP8010\_2018.2.ZIP** - contains all Redfish schemas
- Redfish Registries 2018.1
  - NEW 'Task', 'Resource' registries
  - Updated 'Base', 'Privilege' registries
- **13 NEW Schemas v1.0.0**
  - TelemetryService, MetricDefinition, MetricReportDefinition, MetricReport, Triggers
  - JobService, Job, PCIeSlots, (plus 5 Resource Collections)
- **29 Updated Schemas (highlights)**
  - Schedule v1.2.0 – Contribution from SNIA.org with minor additions
  - EventDestination v1.4.0, Event v1.3.0, MessageRegistry v1.2.0 – Improved subscriptions
  - Resource v1.7.0 – Added 'ContactInfo' array to commonly-used Location object

# Redfish for OpenAPI™ Specification v3.0

- OpenAPI Specification
  - A community-driven open specification from the OpenAPI Initiative (OAI), The Linux Foundation® Collaborative Project
  - Describes API services in a YAML-format definition document
  - Rich ecosystem of tools for developers and end users
- Supported by Redfish Specification v1.6.0
  - Added support for OpenAPI schema files (YAML)
  - Requires use of standardized URIs
    - Fixed URIs for all Redfish resources
    - Construct path to Collection member using “Id”
- Redfish Schema (DSP8010) bundle 2018.2
  - Now includes OpenAPI YAML files
  - Along with JSON Schema and CSDL files

# DCIM Work in Progress v0.8 release

- Download [DSP-IS0005.zip](#) from [www.dmtf.org/standards/redfish](http://www.dmtf.org/standards/redfish)
- Focused on architectural direction and common features of the model
  - General-purpose Sensor model
  - Ability to reflect “sensor readings” elsewhere in the model
  - Handling of alarm conditions where no additional data is available
- Rendered these into one example with schema and mockup
  - Rack-based Power Distribution Unit (RackPDU)
  - Schemas and mockups
- Other DCIM equipment schemas were not updated
  - Desire for feedback on the architectural features as shown in RackPDU
  - All schemas will be refactored in the next release to match the direction



# PowerAPI



# What is the Power API?

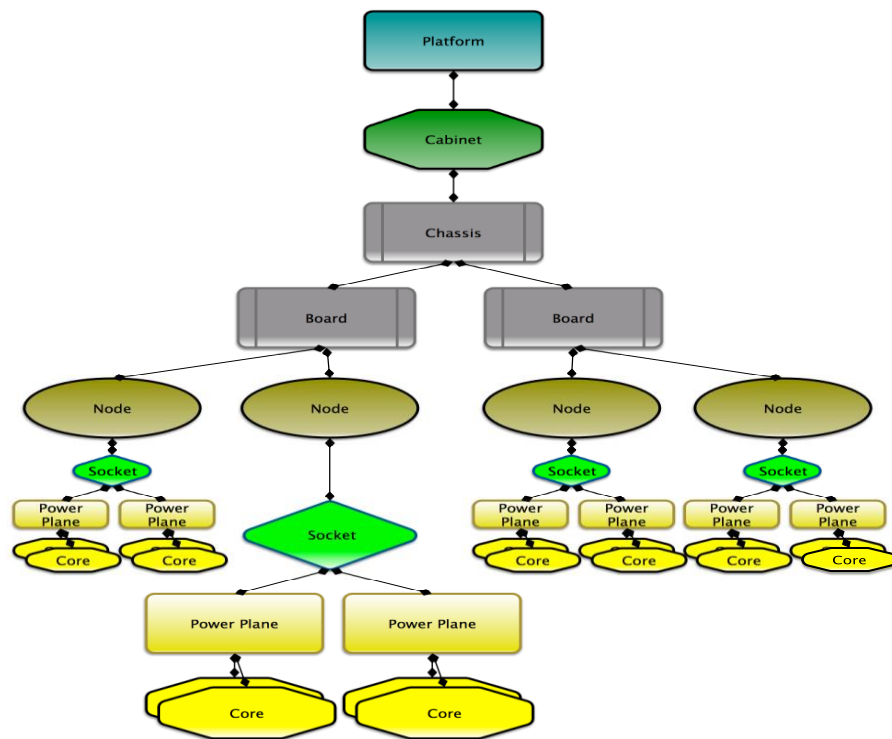
- The Power API is a comprehensive system software API for interfacing with power measurement and control hardware
- Designed to be comprehensive across many different levels of a data center
- Many different actors can interface with a single API to perform several different roles
- Encompasses facility level concerns down to low level software/hardware interfaces

# System Description

Presents a navigable view of the system's hardware components

- Can extend to custom object types
- Can be heterogeneous

```
typedef enum {  
    PWR_OBJ_PLATFORM = 0,  
    PWR_OBJ_CABINET,  
    PWR_OBJ_CHASSIS,  
    PWR_OBJ_BOARD,  
    PWR_OBJ_NODE,  
    PWR_OBJ_SOCKET,  
    PWR_OBJ_CORE,  
    PWR_OBJ_POWER_PLANE,  
    PWR_OBJ_MEM,  
    PWR_OBJ_NIC,  
    PWR_NUM_OBJ_TYPES,  
    /* */  
    PWR_OBJ_INVALID = -1,  
    PWR_OBJ_NOT_SPECIFIED = -2,  
} PWR_ObjType;
```

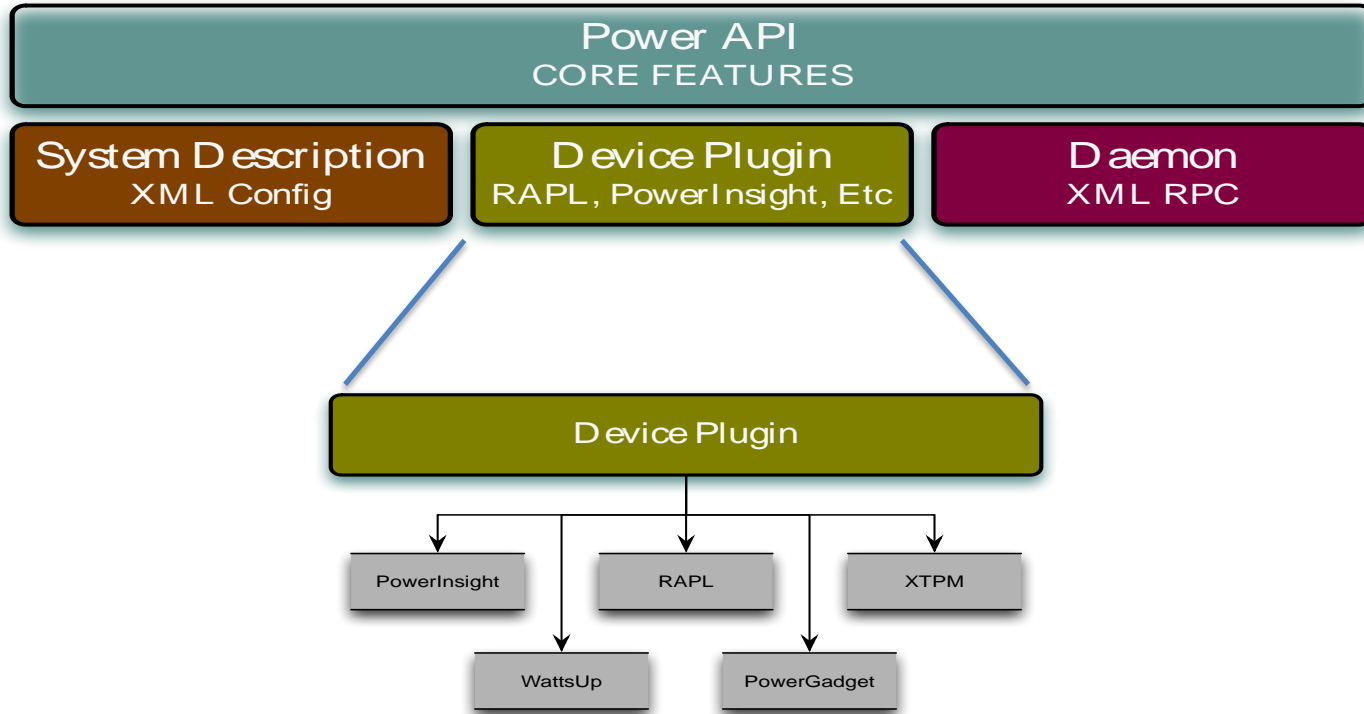


Example System Description

# Common Functionality

- **Navigation** across and **grouping** of **objects** in the system
- **Attributes** (e.g., power cap, voltage) for the objects can be accessed depending on **role** (e.g., user, app, OS, admin)
- **Getters/setters** enable basic measurement and control for the exposed object attributes
- **Metadata** interface provides information about quality, frequency, and other characteristics of measurement/control
- **Statistics** interface gathers data on one or more attributes for an object or group of objects over time

# Reference Implementation

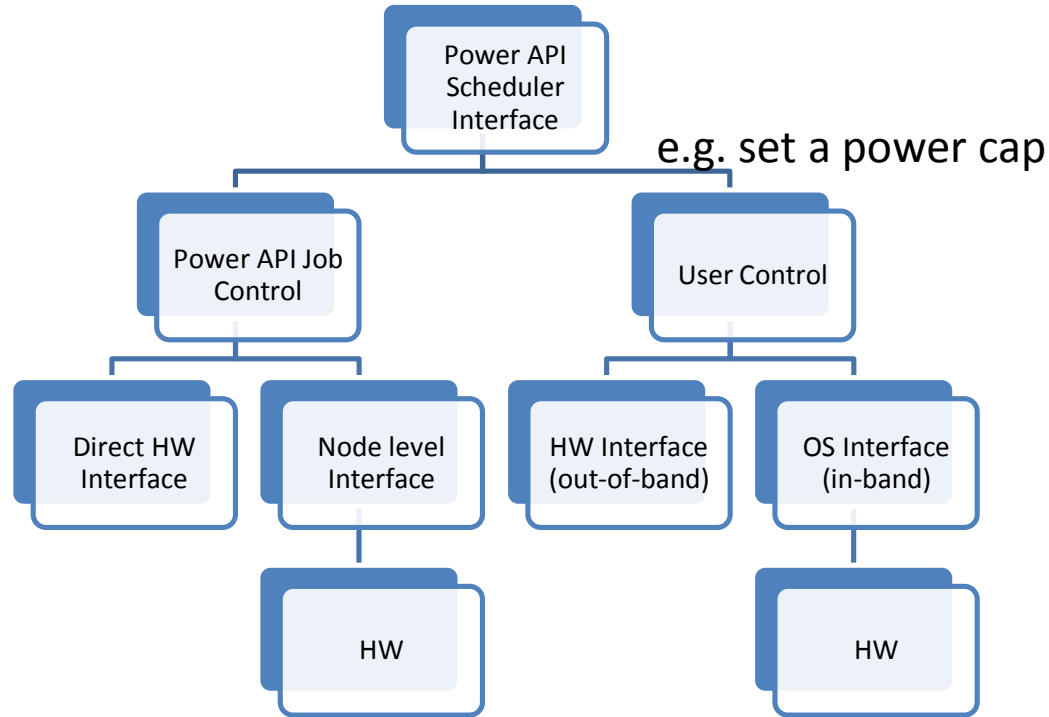


Available online and open source: <http://github.com/pwrapi>

# New advances in the Power API

- Community model
  - New Specifications Document
  - Open meetings
  - Multi-institution involvement
- New convenience functions
  - Streamline writing code with object name length queries
- New reporting functions
  - Allow multiple statistics to be gathered in a single object (forthcoming)
- New notification methods between Interface hierarchy
  - Upcoming

# Power API – Portable Interfaces





**Ryan Grant**

Principal Member of Technical Staff  
Sandia National Laboratories

Chair of the Power API Specification Committee

Working on: Power API Spec, Community Reference  
Implementation





# **Steven J. Martin**

**Staff Engineer, Hardware Management Group  
Cray Inc.**

**Active member Power API Specification Committee**

**Cray HPC Power API development**

**Redfish enablement on next generation Cray systems**



**Todd J. Rosedahl**  
STSM, Power Systems  
IBM Inc.

Active member Power API Specification Committee  
Redfish DMTF – IBM representative  
Redfish enablement on IBM systems