



Enabling Data Center Management with DMTF Redfish

Message Registry Opportunities and Technical
Challenges

Hari Ramachandran (Microsoft)
Chitkala Sethuraman (Microsoft)

October 2025

Copyright © 2025 DMTF



Redfish

www.dmtf.org

Redfish: The Manageability Standard for Hyperscalers

- Hyperscale data centers require a unified, vendor-neutral standard for scalable management
- Redfish delivers interoperability, automation, and consistency across diverse hardware and vendors
- The Message Registry mechanism is central to Redfish's value for hyperscalers.

Registry Name	Version
AutomationNodeEvent	1.0.0
Environmental	1.2.0
JobEvent	1.1.0
Platform	1.3.0
Power	1.0.2
StorageDevice	1.5.0
Telemetry	1.1.0
Update	1.2.0
Base	1.21.0
AccountSecurity	1.0.0
HeartbeatEvent	1.1.0
ResourceEvent	1.4.2
ServiceCommunications	1.0.0

"Redfish's registry ecosystem enables hyperscalers to standardize alerting, automate diagnostics, and scale event management across thousands of nodes."

What is the Redfish Message Registry?

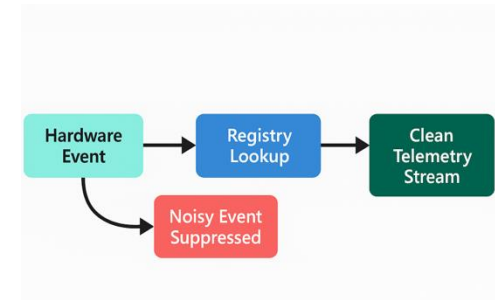
- **Definition:**
 - The Message Registry is a standardized framework that defines machine-readable alert messages used in Redfish-based hardware management
- **Purpose & Value:**
 - Enables consistent interpretation of alerts across vendors and platforms
 - Supports automation of fault detection and diagnostics
 - Provides a foundation for scalable event subscription models
- **How It Works:**
 - Registry entries define the structure and meaning of alerts
 - Standardized messaging simplifies integration and interoperability
 - Registry-driven filtering allows suppression of noisy or redundant events



The Message Registry is central to Redfish's eventing model, enabling automation, cross-vendor consistency, and scalable management in hyperscale data centers

Benefits of Redfish Message Registry

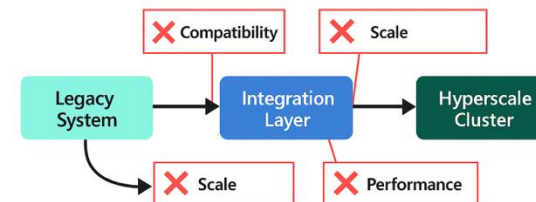
- **Standardized Alert Definitions**
 - Ensures consistent interpretation of events and alerts across vendors and platforms.
 - Reduces ambiguity and manual troubleshooting.
- **Automation & Diagnostics**
 - Enables automated fault detection workflows.
 - Supports predictive diagnostics and rapid response to hardware issues.
- **Registry-Driven Event Filtering**
 - Suppresses noisy or redundant events (e.g., test and housekeeping alerts).
 - Improves telemetry quality and operational efficiency.
- **Interoperability & Integration**
 - Promotes seamless integration across diverse systems.
 - Simplifies management of hardware at scale.
- **Scalable Event Subscription Models**
 - Foundation for managing thousands of event subscriptions efficiently.
 - Prefix-based subscriptions reduce operational overhead.



Redfish Message Registry empowers hyperscalers to automate hardware management, improve operational efficiency, and ensure consistent, reliable event handling across massive data center environments

Integration Challenges in Hyperscale Environments

- **Legacy Infrastructure Compatibility**
 - Many existing systems lack native Redfish support.
 - Integration requires bridging legacy protocols (e.g., IPMI) with Redfish.
- **Phased Adoption & Planning**
 - Migrating to Redfish demands careful planning and staged rollouts.
 - Training and change management are critical for successful integration.
- **Operational Scale**
 - Managing thousands of Redfish subscriptions across hyperscale clusters.
 - Event storms during hardware updates (e.g., GPU firmware pushes) can overwhelm systems.
- **Performance Bottlenecks**
 - RESTful operations may encounter latency and throughput issues at scale.
 - Efficient event delivery and polling mechanisms are needed.
- **Interoperability**
 - Ensuring seamless operation with diverse hardware and vendor implementations.



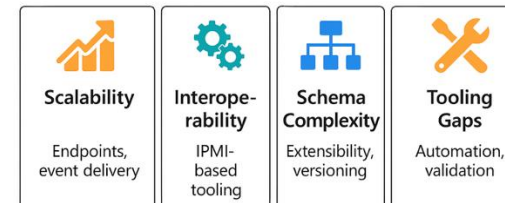
Integrating Redfish into hyperscale environments is not just a technical upgrade—it's a transformation that requires bridging legacy systems, scaling operations, and overcoming performance bottlenecks

Adoption Challenges for Redfish in Hyperscale Data Centers

- **Scalability of Endpoints**
 - Managing Redfish across thousands of nodes can strain infrastructure.
 - Event delivery and polling performance may degrade at scale.
- **Interoperability with Legacy Tooling**
 - Many environments still rely on IPMI-based or proprietary management tools.
 - Ensuring seamless operation between Redfish and legacy systems is complex.
- **Schema Complexity & Extensibility**
 - Redfish's schema may not easily accommodate custom hardware configurations.
 - Extending the schema for new device types or telemetry can be challenging.
- **Automated Versioning & Compatibility**
 - Keeping schemas and implementations up-to-date across vendors and platforms.
 - Automated compatibility checks are needed for reliable operations.
- **Telemetry Integration & Data Volume**
 - Integrating Redfish eventing with existing telemetry pipelines.
 - Managing and processing large volumes of event data efficiently.
- **Tooling Gaps**
 - Lack of robust automation, validation, and compliance tools for Redfish.
 - Friction in deploying and operating Redfish at scale.

Adopting Redfish at hyperscale is more than just technical integration—it requires overcoming challenges in scalability, interoperability, schema evolution, and operational tooling.

Adoption Challenges



Strategic Solutions for Hyperscale Redfish Adoption

- **Federated Redfish Service Architecture**
 - Deploy Redfish services in a federated model to support large-scale, distributed environments.
 - Improves scalability and management across thousands of endpoints.
- **Role-Based Access Control at Scale**
 - Implement granular, scalable authentication and authorization.
 - Ensures secure operations in hyperscale data centers.
- **Event Service Optimization**
 - Enhance Redfish Event Service for high-volume, hyperscale scenarios.
 - Reduce event storms and improve delivery efficiency.
- **Custom Schema Overlays**
 - Develop schema extensions tailored for hyperscale hardware profiles.
 - Maintain interoperability while supporting diverse device types.
- **Enhanced Telemetry & Real-Time Monitoring**
 - Improve message registry and event capabilities for robust telemetry.
 - Enable real-time monitoring and diagnostics.
- **Firmware Lifecycle Management**
 - Streamline firmware updates and configuration control for thousands of devices.
 - Reduce operational overhead and risk.
- **Secure, Scalable Access Control**
 - Strengthen authentication mechanisms for large-scale environments.
 - Support role-based and attribute-based access models.

Strategic enhancements—such as federated architectures, optimized event services, and custom schema overlays—are essential for scaling Redfish adoption and unlocking its full potential in hyperscale environments.

Call to Action

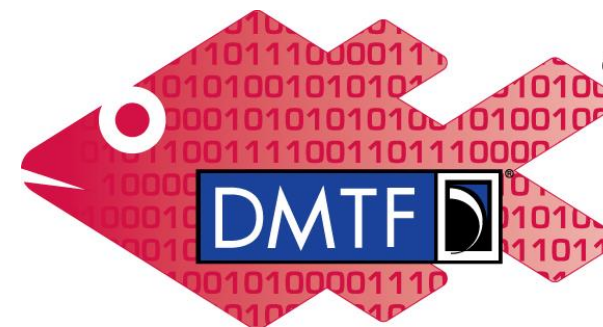
- **Engage:** Join Redfish standards and working groups.
- **Collaborate:** Share your hyperscale requirements and challenges.
- **Adopt:** Pilot Redfish solutions in your data centers.
- **Partner:** Work with standards to shape future enhancements.

Let's build scalable, secure, and interoperable data centers—together.

Thank you!

**For more information,
visit us online at
dmtf.org**

**Visit the
Redfish Developers Hub at
redfish.dmtf.org**



Redfish