Hardware Error Reporting Standardization
for cloud-scale and Edge Infrastructure
Hardware Error Reporting Standardization

Jeff Autor, Distinguished Technologist, HPE
Rama Bhimanadhuni, Principal Engineering Lead, Microsoft
Antonio Hasbun, Principal Engineer, Intel
Problem statement

Cloud and Edge hardware consists of heterogeneous hardware with custom error reporting solutions

Impact:

• Tools for analyzing log contents cannot work across vendor implementations
• Difficulty in ML based error patterns analytics
• Contribute to long lead times for identifying mitigations and Solutions

A modern standard solution is needed for hardware error reporting, storing and transporting to increase the effectiveness of fleet management
Requirements

• Autonomous and near real time *FRU Isolation* for all types of Hardware Failures.

• Need *structured data* that describes hardware errors

• Support *filtering* and *categorization*

• Allow entries to contain an *unbounded* amount of data for deep dive

• Work with *existing* diagnostic *tools* and *APIs*

• *Extendable* to include future hardware error types and expanding scope of error coverage
CPER/Redfish recap

**CPER -> UEFI’s Common Error Record Format**
- Current Industry standard which provides extensive hardware error information
- Encapsulate MCA, AER, CXL and other hardware errors
- Firmware implementations and tools support is available
- It is a highly structured, binary format
- Does not define a network transport for delivering this data to clients

**Redfish -> DMTF’s standard management protocol**
- Defines a data model for event logs
- Easily extendable to include CPER data within the log model
- Provides a publish/subscribe model for event delivery
- Widely adopted for fleet management solutions
How Redfish encapsulates CPER

Any Redfish log entry allows an “Additional Data” URI that can contain a CPER record

Example:

```
"@odata.type": 
"#LogEntry.v1_10_0.LogEntry",
"Id": "1",
"EntryType": "Event",
"Severity": "Critical",
"Created": "2022-03-07T14:44:00Z",
"Message": "A Machine Check Exception (MCE) error has occurred."
"MessageId": "Platform.1.0.MachineCheckException",
"GeneratorId": "UEFI-BIOS",
"DiagnosticDataType": "CPER",
"AdditionalDataURI": "/logs/236912.bin",
"AdditionalDataSizeBytes": 4592,
```
New Redfish Message Registry

- Redfish defines log and event messages in a machine-readable “Message Registry”
- Creating a new registry for “Platform” errors covered by CPER or other common sources of fault reporting (e.g. Machine Check Exception)
- Message registry can be expanded to cover new hardware error types (backwards compatible)
Example message registry (one message)

"CableInserted": {
    "Description": "Indicates that a network cable has been inserted.\",
    "LongDescription": "This message shall be used to indicate that a network cable has been inserted.\",
    "Message": "A network cable has been inserted into network adapter '%1' port '%2'.\",
    "Severity": "OK",
    "NumberOfArgs": 2,
    "ParamTypes": [ "string", "string" ],
    "ArgDescriptions": [
        "The `Id` of the network adapter.\",
        "The `Id` of the network port." ],
    "ArgLongDescriptions": [
        "The value of this argument shall be a string containing ...",
        "The value of this argument shall be a string containing ... " ],
    "Resolution": "Refresh your cached version of the network port ...",
    "ClearingLogic": {
        "ClearsIf": "SameOriginOfCondition",
        "ClearsMessage": [ "CableRemoved" ]
    }
},

OPEN POSSIBILITIES.
Join the group discussion here!

- Requirement document has been contributed (v0.5)
  - Link: here

- Provide feedback on error messages to standardize in the Message Registry
  - Link: https://www.dmtf.org/standards/redfish

- Implement the Redfish “Diagnostic Data” logging support to provide the CPER data to client software in a standard format

- Join the mailing list
  - Link: https://ocp-all.groups.io/g/OCP-HWFaultMgt
Thank you!