



WHITE PAPER: INBAND MANAGEMENT AGENTS

Authors : George Kola, Murugasamy Nachimuthu, Panos Christeas,
Chukwunenye Nnebe, Changho Choi (OCP FTI - Cloud Service Model (CSM)
Working Group)

Executive Summary

Inband Management agent (IMA) is a process that runs on the same component it is monitoring (e.g. host CPU of server) and provides management functionality. Management functions include telemetry, monitoring and management of operation (e.g. switching operation mode, restarting a processor, power on-off, configuration, error handling, etc). Currently, there is no standardized design of an in-band agent that offers telemetry and monitoring services in Redfish format and is abstracted and isolated from the host operating system in a way to satisfy the requirements and constraints of datacenter usage.

We are addressing this gap and by creating an open design, we enable the following

1. **Shareability:** This will allow vendors to provide agents for their hardware and enable organizations to utilize it instead of trying to build it in-house. Further, organizations can utilize existing agents.
2. **Serviceability:** This monitoring data can be shared with vendors and enable replacements of failing, failed and about to fail components. This will help develop a service model for hardware components.
3. **Security:** This would enable securely sharing just the needed monitoring data without any ability to compromise or expose user data. This eliminates potential issues with proprietary inband tools that may accidentally access user data.
4. **Address missing functionality:** Current management specifications cover out of band use-cases only. There are multiple use cases where the telemetry/management functionality is either inband-only or needs inband to augment out-of-band functionality.
5. **Support high frequency/high bandwidth use cases:** Inband access is required to support high frequency and high bandwidth use cases and live debug use cases where it is feasible.
6. We will be upstreaming the additional schemas to DMTF

Table of Contents

Introduction	4
1 Overview	5
1.1 Terminology	5
What is an inband agent ?	5
What about Smart NIC ?	6
What is an inband management agent ?	6
1.2 What is the problem we are trying to solve ?	6
Shareability	6
Serviceability	6
Security	6
Address missing functionality	7
2 Inband Management Use Cases	7
2.1 Inband only access	8
2.2 High bandwidth requirement	8
2.3 Latency Sensitive Diagnostics/High processing power requirement	8
2.4 Coordination with Host OS	8
2.4 Augment Out-of-Band/BMC	9
2.5 Security Use case	9
Security key exchange with firmware. Agent can be used to provision security keys with the device. For example an inband agent can set up the security keys with SSD firmware and HDD firmware.	9
Coordination with Host OS	9
3 Inband Management Agent (IMA) Requirements	10
4 Inband Management Agent (IMA) Architecture	10
We propose the following IMA architecture	10
5 Inband and Out Of Band Aggregation	12
6	12
7 Conclusion	12
8 Glossary	12
9 References	12
10 License	13
11 About Open Compute Foundation	14
12 Appendix A. [Title]	15

Introduction

Inband Management agent (IMA) is a process that runs on the same component it is monitoring (e.g. host CPU of server) and provides management functionality. Management functions include telemetry, monitoring and management of operation (e.g. switching operation mode, restarting a processor, power on-off, configuration, etc). Currently, there is no standardized design of an in-band agent that offers telemetry and monitoring services in Redfish format and is abstracted and isolated from the host operating system in a way to satisfy the requirements and constraints of datacenter usage.

We are addressing this gap and by creating an open design, we enable the following

1. **Shareability:** This will allow vendors to provide agents for their hardware and enable organizations to utilize it instead of trying to build it in-house. Further, organizations can utilize existing agents.
2. **Serviceability:** This monitoring data can be shared with vendors and enable replacements of failing, failed and about to fail components. This will help develop a service model for hardware components.
3. **Security:** This would enable securely sharing just the needed monitoring data without any ability to compromise or expose user data. This eliminates potential issues with proprietary inband tools that may accidentally access user data.
4. **Address missing functionality:** Current management specifications cover out of band use-cases only. There are multiple use cases where the telemetry/management functionality is either inband-only or needs inband to augment out-of-band functionality.
5. **Support high frequency/high bandwidth use cases:** Inband access is required to support high frequency and high bandwidth use cases and live debug use cases where it is feasible.

1 Overview

We first define the terminology and then describe the problem we are trying to solve.

1.1 Terminology

What is an inband agent ?

If an agent/process runs on the same component it is monitoring utilizing the component CPU and memory, it is considered as an inband agent. For example, if an agent runs on the host CPU of the server and monitors the server, it is inband. Similarly, an inband agent may run on a smart NIC and monitor the smart NIC.

What is an out-of-band agent ?

If an agent/process runs on a complementary hardware to the component being monitored typically on a separate controller like baseband management controller (BMC), it is considered as out-of-band.

What about Smart NIC ?

If an agent runs on the smart NIC CPU and monitors the smart NIC, it is an inband agent. If the agent running on smart NIC is used to monitor a separate tray or host memory, it is out-of-band.

What is an inband management agent ?

Inband Management agent (IMA) is a process that runs on the same component it is monitoring (e.g. host CPU of server) and provides management functionality. Management functions include telemetry, monitoring and management of operation (e.g. switching operation mode, restarting a processor, power on-off, configuration, error handling, etc).

1.2 What is the problem we are trying to solve ?

Today there is no open design for inband monitoring/diagnostics, telemetry and management agent that speaks Redfish. Furthermore, some of the monitoring, telemetry and management are available only out-of-band. We are addressing this gap by creating an open design for inband management agent

Creating this open standard enables the following:

Shareability

This will allow vendors to provide agents for their hardware and enable organizations to utilize it instead of trying to build it in-house. Further, organizations can utilize existing agents.

Serviceability

This monitoring data can be shared with vendors and enable replacements of failing, failed and about to fail components. This will help develop a service model for hardware components.

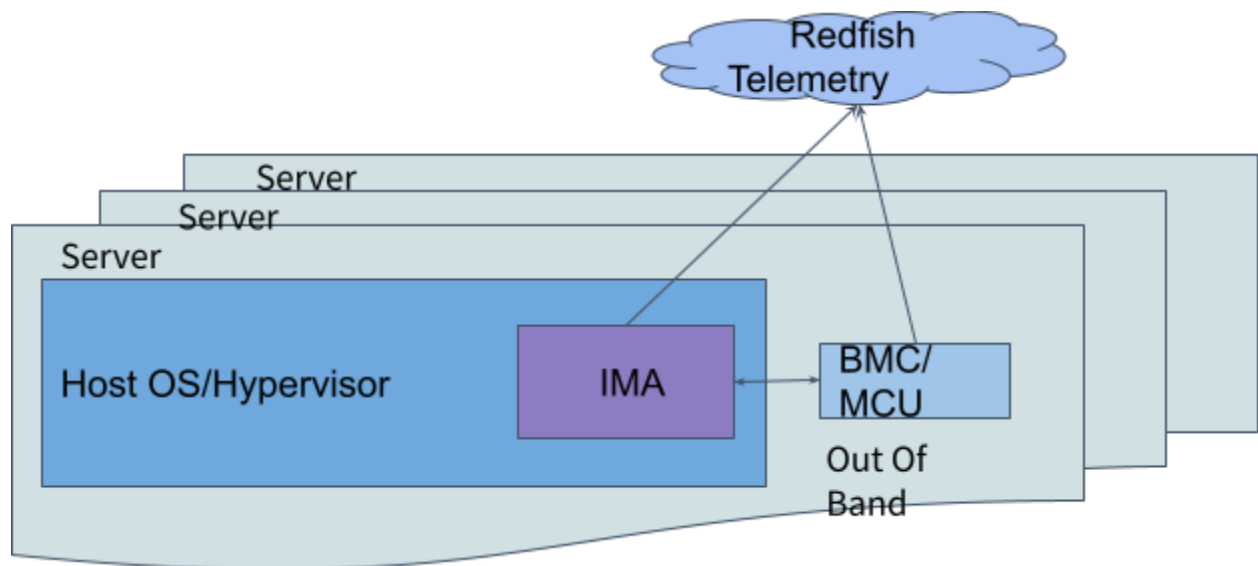
Security

This would enable securely sharing just the needed monitoring data without any ability to compromise or expose user data.

This eliminates potential issues with proprietary inband tools that may accidentally access user data.

Address missing functionality

Current management specifications cover out of band use-cases only. There are multiple use cases where the telemetry/management functionality is either inband-only or needs inband to augment out-of-band functionality. Inband is required to support high frequency and high bandwidth use cases and live debug use cases.



2 Inband Management Use Cases

There are a number of inband use cases namely

2.1 Inband only access

The full device functionality is accessible only inband due to lack of out-of-band access or limited out-of-band access. Examples include the following

- Hard Disks and disks with SATA & SAS interfaces.
- Memory error reporting
- PCIe errors
- PCIe device
 - Typically PCIe devices are only accessible from inband.
 - There is a CPU vendor specific mechanism to get some PCIe info (Platform Environment Control Interface (PECI) for Intel, Advanced Platform Management Link (APML) for AMD) but actual access to PCIe devices is not possible.

2.2 High bandwidth requirement

Some functionality may need higher bandwidth that may not be available on the out-of-band path. Examples include real-time debugging, fetching device logs, latency sensitive firmware updates, etc.

- i2c bus bandwidth is limited. i3c when adopted would reduce this issue but it may be inefficient to service models depending on data size .

2.3 Latency Sensitive Diagnostics/High processing power requirement

Certain diagnostics may need to be run on reboot and reboot have a short window and those diagnostics need high bandwidth/processing power to be able to meet the latency requirement.

For example, row hammer vulnerability that needs host access and latency sensitive diagnostics information.

2.4 Coordination with Host OS

Firmware updates may need coordination with the host OS/hypervisor to stop other access to the device (e.g. via device driver unload) and then update the firmware and finally reload the device driver and enable access to the device.

Further, this may involve coordination with higher level system (e.g. draining the workload running on the device and making the device not-schedulable before update and then marking it as available after update)

2.4 Augment Out-of-Band/BMC

- IMA can be used to pipe data from BMC. For example, BMC may not have network connectivity and an inband agent can be used to provide connectivity to BMC.
- An inband agent can provide authorized access to BMC.
- Inband Agent can export a higher level interface to a simple microcontroller and augment its functionality (e.g. convert fixed point microcontroller data to floating point)
- Inband agents can combine both inband and out-of-band to provide a unified functionality. For example, on storage devices there may be a need to combine kernel information like partition/volume that is accessible only inband with the hardware information available out-of-band and get an unified view from the inband agent. Another use case is that on some machines HDD write protect and reset pin may be accessible only out-of-band and for firmware update, we would need to toggle write protect, update the firmware, toggle write protect and then reset the HDD for firmware change to take effect.
- memory error logging (inband memory error logs passed on to out-of-band). Can be extended to other error logging that is accessible inband only.

2.5 Security Use case

Security key exchange with firmware. Agent can be used to provision security keys with the device. For example an inband agent can set up the security keys with SSD firmware and HDD firmware.

Coordination with Host OS

For example, authorized access to data.

- Disks can be locked.
- Access may be limited till explicit access is requested and granted
- Disks are then unlocked and
- Authorized access
- When the access is completed or with a timeout, disks are locked again

IMA can serve as a Root of Trust component

Root trust for device

Root of trust for software update

3 Inband Management Agent (IMA) Requirements

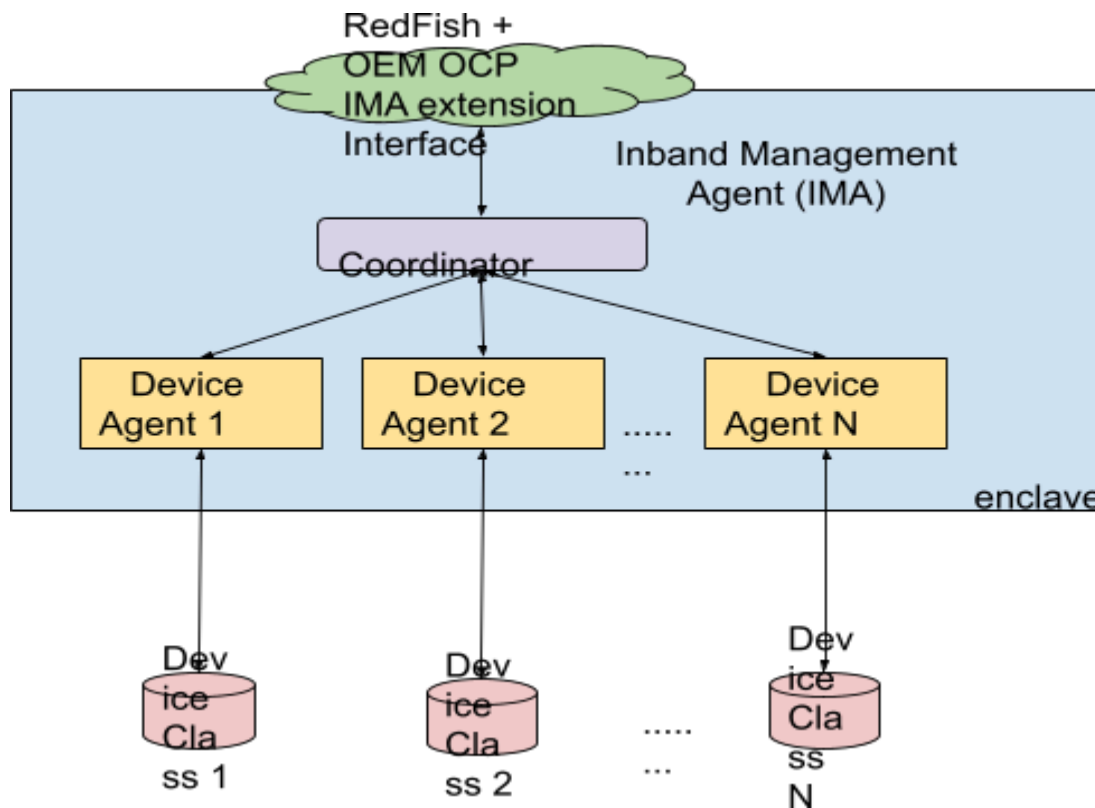
1. Each Inband Management Agent (IMA) should support a predefined class of devices. We do not want configurable IMA that can support any device as that may allow for accidentally accessing user-data or accessing data from another vendor device. Agents will have to explicitly specify the device class/type they can support management for.
2. The architecture should ensure that Vendor A agent does not access data from Vendor B device.
3. The architecture should allow agent discovery i.e. find the agent that can handle a detected device.
4. The architecture should allow extensibility. For example, if a new device supports a few additional functionality compared
5. The architecture should allow controlling when the agent is activated/deactivated. For example, we want the agent to be activated only when a hardware device is found and not activated otherwise.

4 Inband Management Agent (IMA) Architecture

We propose the following IMA architecture

- a. There is a central co-ordinator that provides common functionality like PCIe scans in a secure manner
- b. There is a set of Inband Management Agents with each Agent supporting a predefined set of device classes/types specified in a manifest.
- c. There is inspection of Manifest of each agent to ensure that it just supports the appropriate device classes (e.g. vendor restriction, device class restriction).
 - i. For example, SSD vendor A Agent can only access information about SSD vendor A SSDs.
 - ii. This is similar to how Android/iOS apps specify what access they have to the camera/location/address book etc.

- d. An administrator can determine the set of agents to be allowed on a machine.
 - i. This can be done statically apriori to determine the agents allowed and access they have.
- e. This allowed set of agents are registered with the co-ordinator.
- f. The co-ordinator at startup and on device plugin will check for available devices and figure out the appropriate agents to activate. There is an agent priority order such that an agent with highest priority will be activated for a given device. This functionality is used to ensure specialization i.e. a vendor can give a specialized agent for a specific device while a generic agent can handle the rest of the device class of the vendor.
- g. Agents can rely on the coordinator for common functionality like reading PCIe config space etc.
- h. The coordinator exports a Redfish API. Incoming Redfish API requests are routed to the appropriate agent and the telemetry is fetched and reported.
- i. The co-ordinator functionality allows an agent to depend on another agent securely. We do not want agents to talk to each other but sometimes, there may be a need to allow certain agent functionality to be exported to be used by other agents.



The figure above shows an example. The co-ordinator detects the different device classes and activates the appropriate inband management agents and the agents manage their device and

communicate with the co-ordinator and the coordinator exposes a Redfish API to manage the devices and export their telemetry. Agents run inside the enclave.

5 Inband and Out Of Band Aggregation

6 Fault Management use case

7 Conclusion

In this white paper we made the case for need of an open design of Inband Management Agent and made that design available.

8 Glossary

Term: definition.

9 References

1. Books
2. Web resources

10 License

There are 2 types of license under which a document can be submitted. Please provide one only.

- Creative Commons - This is the suggested language.

OCP encourages participants to share their proposals, specifications and designs with the community. This is to promote openness and encourage continuous and open feedback. It is important to remember that by providing feedback for any such documents, whether in written or verbal form, that the contributor or the contributor's organization grants OCP and its members irrevocable right to use this feedback for any purpose without any further obligation.

It is acknowledged that any such documentation and any ancillary materials that are provided to OCP in connection with this document, including without limitation any white papers, articles, photographs, studies, diagrams, contact information (together, "Materials") are made available under the Creative Commons Attribution-ShareAlike 4.0 International License found here:

<https://creativecommons.org/licenses/by-sa/4.0/>, or any later version, and without limiting the foregoing, OCP may make the Materials available under such terms.

As a contributor to this document, all members represent that they have the authority to grant the rights and licenses herein. They further represent and warrant that the Materials do not and will not violate the copyrights or misappropriate the trade secret rights of any third party, including without limitation rights in intellectual property. The contributor(s) also represent that, to the extent the Materials include materials protected by copyright or trade secret rights that are owned or created by any third-party, they have obtained permission for its use consistent with the foregoing. They will provide OCP evidence of such permission upon OCP's request. This document and any "Materials" are published on the respective project's wiki page and are open to the public in accordance with OCP's Bylaws and IP Policy. This can be found at <http://www.opencompute.org/participate/legal-documents/>. If you have any questions please contact OCP.

Footer:



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

- Copyright License - This is the suggested language.

© Copyright [Company Name][Year] All rights reserved.

If your company has different copyright language, please insert your copyright information here.



OCP will need a signed [Copyright License Agreement](#) so we can publish this on our website. Please insure you have signed this agreement.

11 About Open Compute Foundation

At the core of the Open Compute Project (OCP) is its Community of hyperscale data center operators, joined by telecom and colocation providers and enterprise IT users, working with vendors to develop open innovations that, when embedded in product are deployed from the cloud to the edge. The OCP Foundation is responsible for fostering and serving the OCP Community to meet the market and shape the future, taking hyperscale led innovations to everyone. Meeting the market is accomplished through open designs and best practices, and with data center facility and IT equipment embedding OCP Community-developed innovations for efficiency, at-scale operations and sustainability. Shaping the future includes investing in strategic initiatives that prepare the IT ecosystem for major changes, such as AI & ML, optics, advanced cooling techniques, and composable silicon. Learn more at www.opencompute.org.

12 Appendix A. [Title]