IP Based Drive Management Specification
Version 0.0.10 rev 0

ABSTRACT: This document describes the management features and functions of a storage device class known as IP Based Drives. It includes a taxonomy covering the scope of involved device capabilities.

Publication of this Working Draft for review and comment has been approved by the Object Drive Technical Work Group (TWG). This draft represents a “best effort” attempt by the Object Drive TWG to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a “work in progress.” Suggestion for revision should be directed to http://www.snia.org/feedback/.
USAGE
The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

1. Any text, diagram, chart, table or definition reproduced shall be reproduced in its entirety with no alteration, and,

2. Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing tcmd@snia.org. Please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

All code fragments, scripts, data tables, and sample code in this SNIA document are made available under the following license:

BSD 3-Clause Software License

Copyright (c) 2014, The Storage Networking Industry Association.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of The Storage Networking Industry Association (SNIA) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
DISCLAIMER
The information contained in this publication is subject to change without notice. The SNIA makes no warranty of any kind with regard to this specification, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this specification.

Suggestions for revisions should be directed to http://www.snia.org/feedback/.

Copyright © 2016 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.
# Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Sections</th>
<th>Originator:</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.1</td>
<td>Dec. 22, 2015</td>
<td>All</td>
<td>Mark Carlson, James Espy</td>
<td>First draft combining contents of an earlier requirements white paper and management spec draft.</td>
</tr>
<tr>
<td>0.002</td>
<td>Dec. 23, 2015</td>
<td>Various spots</td>
<td>James Espy</td>
<td>Updates to contributor list and various editor notes.</td>
</tr>
<tr>
<td>0.003</td>
<td>Jan 18, 2016</td>
<td>Various spots</td>
<td>James Espy</td>
<td>Updates from review comments.</td>
</tr>
<tr>
<td>0.004</td>
<td>Jan 18, 2016</td>
<td>2.1 References 4.3 Acronyms 6 Management</td>
<td>James Espy</td>
<td>Acronyms additions. Section 6 organization update.</td>
</tr>
<tr>
<td>0.005</td>
<td>Feb 8, 2016</td>
<td>7 Use Cases 5.2 Electrical</td>
<td>James Espy</td>
<td>Removed section 7. Modified table in section 5.2.</td>
</tr>
<tr>
<td>0.006</td>
<td>March 14, 2016</td>
<td>4.2 Definitions</td>
<td>James Espy</td>
<td>Modified (added) definitions</td>
</tr>
<tr>
<td>0.007</td>
<td>March 22, 2016</td>
<td>Sections 5 and 6</td>
<td>James Espy</td>
<td>Selected text updates. Pictorial updates. Moved Redfish manageability example to Annex.</td>
</tr>
<tr>
<td>0.008</td>
<td>May 10, 2016</td>
<td></td>
<td>James Espy</td>
<td>Abstract and introduction updates. Table 1 References update; 5.2 updates; 5.5 Taxonomy depiction update; 6 and 7 updates;</td>
</tr>
<tr>
<td>0.009</td>
<td>May 11, 2016</td>
<td></td>
<td>Mark and Jim</td>
<td>Multiple changes of cleanup and change accepting</td>
</tr>
<tr>
<td>0.010</td>
<td>May 24, 2016</td>
<td>Many</td>
<td>James Espy</td>
<td>Consistent use of IP Based Drive throughout document</td>
</tr>
</tbody>
</table>
Intended Audience

This document is intended for use by individuals and companies engaged in developing storage systems utilizing IP Based Drive devices and/or related sub-systems.

Changes to the Specification

Each publication of this specification is uniquely identified by a three-level identifier, comprised of a version number, a release number and an update number. The current identifier for this specification is listed on the title page of this document. Future publications of this specification are subject to specific constraints on the scope of change that is permissible from one publication to the next and the degree of interoperability and backward compatibility that should be assumed between products designed to different publications of this standard. The SNIA has defined three levels of change to a specification:

- Major Revision: A major revision of the specification represents a substantial change to the underlying scope or architecture of the specification. A major revision results in an increase in the version number of the version identifier (e.g., from version 1.x.x to version 2.x.x). There is no assurance of interoperability or backward compatibility between releases with different version numbers.
- Minor Revision: A minor revision of the specification represents a technical change to existing content or an adjustment to the scope of the specification. A minor revision results in an increase in the release number of the specification’s identifier (e.g., from x.1.x to x.2.x). Minor revisions with the same version number preserve interoperability and backward compatibility.
- Update: An update to the specification is limited to minor corrections or clarifications of existing specification content. An update will result in an increase in the third component of the release identifier (e.g., from x.x.1 to x.x.2). Updates with the same version and minor release levels preserve interoperability and backward compatibility.
About SNIA

The Storage Networking Industry Association (SNIA) is a not-for-profit global organization, made up of some 400 member companies spanning virtually the entire storage industry. SNIA’s mission is to lead the storage industry worldwide in developing and promoting standards, technologies, and educational services to empower organizations in the management of information. To this end, the SNIA is uniquely committed to delivering standards, education, and services that will propel open storage networking solutions into the broader market. For additional information, visit the SNIA web site at www.snia.org.

Acknowledgements

The SNIA Object Storage Technical Working Group (TWG), which developed and reviewed this specification, would like to recognize the significant contributions made by the following members:

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark Carlson</td>
<td>Toshiba</td>
</tr>
<tr>
<td>David Slik</td>
<td>NetApp</td>
</tr>
<tr>
<td>James Espy</td>
<td>EMC</td>
</tr>
<tr>
<td>Paul Suhler</td>
<td>WDC</td>
</tr>
<tr>
<td>Bill Martin</td>
<td>Samsung</td>
</tr>
<tr>
<td>Don Heins</td>
<td>Seagate</td>
</tr>
</tbody>
</table>
Contents

1 INTRODUCTION ........................................................................................................... 8
  1.1 OVERVIEW ............................................................................................................ 8

2 NORMATIVE REFERENCES ..................................................................................... 9
  2.1 REFERENCES ....................................................................................................... 9

3 SCOPE ...................................................................................................................... 11
  3.1 DESCRIPTION ...................................................................................................... 11

4 DEFINITIONS, SYMBOLS, ABBREVIATIONS, AND CONVENTIONS ................. 12
  4.1 OVERVIEW .......................................................................................................... 12
  4.2 DEFINITIONS ........................................................................................................ 12
  4.3 ACRONYMS AND ABBREVIATIONS ..................................................................... 12
  4.4 KEYWORDS ......................................................................................................... 13
  4.5 CONVENTIONS ..................................................................................................... 14

5 IP BASED DRIVE CHARACTERISTICS AND REQUIREMENTS .......................... 16
  5.1 PHYSICAL LAYER – FORM FACTOR ..................................................................... 16
  5.2 ELECTRICAL ....................................................................................................... 16
  5.3 LINK LAYER ......................................................................................................... 16
  5.4 ENVIRONMENTAL .............................................................................................. 17
  5.5 TAXONOMY ........................................................................................................ 17

6 IP BASED DRIVE MANAGEMENT ........................................................................ 21
  6.1 OVERVIEW .......................................................................................................... 21
  6.2 ESTABLISH PHYSICAL LINK .............................................................................. 21
  6.3 ASSIGN IP ADDRESS .......................................................................................... 21
  6.4 DISCOVER BASIC SERVICES .............................................................................. 21
  6.5 REDFISH MANAGEABILITY ............................................................................... 22

7 ANNEX .................................................................................................................... 25
  7.1 EXAMPLE OF REDFISH MANAGEABILITY ......................................................... 25
1 Introduction

1.1 Overview

The growing popularity of object based storage has resulted in the development of Ethernet-connected storage devices, herein referred to as IP Based Drives and subsystems supporting object interfaces and in some cases the ability to run localized applications. This document specifies management features of these devices.

A number of storage solutions, as part of open source and other projects, are architected to scale out by incrementally adding and removing functional storage nodes. Example projects include Hadoop HDFS, Ceph, and Swift (OpenStack object storage).

Typical scale-out storage nodes consist of relatively inexpensive enclosures with IP network connectivity, CPU, Memory and Direct Attached Storage (DAS). While inexpensive to deploy, these solutions become harder to manage over time. Power and space requirements of data centers are difficult to meet with this type of solution. IP Based Drive vendors look to enable solutions that better meet data center requirements by re-partitioning solutions into drive-based storage nodes and creating points of interoperability.

Some of the aspects in this document may be met by various designs that fall outside of a typical drive form factor and yet may still interoperate at some level, such as the object protocol. Examples are an interposer that has two different interfaces (e.g. converting Ethernet to SATA) or an entire enclosure comprised of standard storage devices fronted by an object interface processing unit that pools the individual drives.
## 2 Normative References

### 2.1 References

Table 1 lists the standards, specifications and other documents related to this specification.

<table>
<thead>
<tr>
<th>Author/Owner</th>
<th>Title</th>
<th>Revision</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Compute Project</td>
<td>Storage Device with Ethernet Interface</td>
<td>4</td>
<td><a href="http://www.opencompute.org/wiki/storage">http://www.opencompute.org/wiki/storage</a></td>
</tr>
<tr>
<td>IEEE</td>
<td>IEEE 802.3cb</td>
<td></td>
<td><a href="http://www.ieee802.org">http://www.ieee802.org</a></td>
</tr>
<tr>
<td></td>
<td>RFC 4171 - iSNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 791 - IPV4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 2460 - IPV6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMTF</td>
<td>Redfish</td>
<td></td>
<td><a href="http://www.dmtf.org/standards/redfish">http://www.dmtf.org/standards/redfish</a></td>
</tr>
<tr>
<td></td>
<td>RFC 2131 – DHCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 2132 - DHCP Options and BOOTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 1035 - DNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 6762 Multi-case Domain name Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RFC 6763 – mDNS/DNS-SD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAS-3</td>
<td></td>
<td><a href="http://www.t10.org">http://www.t10.org</a></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF 8201 – 2.5” Drive Form Factor Dimensions</td>
<td></td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF 8223 – 2.5” Drive Form Factor with Serial Connector</td>
<td></td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF 8301 – 3.5” Drive Form Factor Dimensions</td>
<td></td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF 8323 – 3.5&quot; Drive Form Factor with Serial Connector</td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF-8601</td>
<td>0.7</td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
</tr>
<tr>
<td>SFF Committee</td>
<td>SFF-9639</td>
<td>1.0</td>
<td><a href="http://www.sffcommittee.org/">http://www.sffcommittee.org/</a></td>
</tr>
</tbody>
</table>
3 Scope

3.1 Description

This document focuses on devices and interfaces within drive (e.g. HDD and SSD) form factors. Larger and smaller form factors and varying physical connectivity are not precluded. Additionally, there is no bias or advocacy toward any particular data persistence technology. IP Based Drive devices can utilize spinning media of virtually any recording technology along with flash and non-volatile solid state technology.

This specification is focused on an in-band management API that can be used for scale-out management of IP Based Drives using the IP protocol.

This document is divided into several sections. The first lists key definitions, conventions, etc. The second describes fundamental attributes of IP Based Drives. The third details management functionality and represents the main content of this specification. The Annex discusses selected topics useful in understanding this document.
4 Definitions, Symbols, Abbreviations, and Conventions

4.1 Overview
The terms and definitions used in this specification are based on those found in the SNIA dictionary (www.snia.org/education/dictionary) and the industry. In cases where the current definitions in the SNIA dictionary conflict with those presented in this document, the definitions in this document shall be assumed to be correct.

4.2 Definitions

key value storage
A type of object storage interface where a key is used to address the associated object.

object
The encapsulation of data and associated meta-data.

object drive
A storage element that directly provides object services.

object service
Object level access to storage.

object storage
A storage device that provides object services. Object storage includes DSaaS.

4.3 Acronyms and Abbreviations

API Application Programming Interface
ACS-4 ATA Command Set-4
ATA Advanced Technology Attachment
CDMI Cloud Data Management Interface
DAS Direct Attached Storage
DHCP Dynamic Host Configuration Protocol
DMTF Distributed Management Task Force
DNS Domain Name Service
FC Fibre Channel
FRU Field-Replaceable Unit
GbE Gigabit Ethernet
HDFS Hadoop Distributed File System
HDD Hard Disk Drive
IB InfiniBand
INCITS InterNational Committee on Information Technology Standards
IP Internet Protocol (Alt. Intellectual Property)
4.4 Keywords

expected
A keyword used to describe the behavior of the hardware or software in the design models presumed by this standard. Other hardware and software design models may also be implemented.

mandatory
A keyword indicating an item that is required to be implemented as defined in this specification to claim compliance with this specification.

may
A keyword that indicates flexibility of choice with no implied preference.

may not
Keywords that indicate flexibility of choice with no implied preference.

obsolete
A keyword indicating that an item was defined in prior revisions to this specification but has been removed from this revision.
optional
A keyword that describes features that are not required to be operational. However, if any optional feature is operational, it shall be implemented as defined in this specification.
prohibited
A keyword used to describe a feature or behavior that is not allowed to be present.
required
A keyword used to describe a behavior that shall be implemented.
reserved
A keyword referring to bits, bytes, words, fields, and code values that are set aside for future standardization.
Note 1 to entry: A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard.
Note 2 to entry: Recipients are not required to check reserved bits, bytes, words or fields for zero values; receipt of reserved code values in defined fields shall be reported as an error.

shall
A keyword indicating a mandatory requirement.
should
A keyword indicating flexibility of choice with a preferred alternative; equivalent to the phrase “it is recommended”.

4.5 Conventions
Certain words and terms used in this specification have a specific meaning beyond their normal English meaning. These words and terms are defined either in 4.2 or in the text where they first appear.
Numbers that are not immediately followed by lower-case b or h are decimal values. Numbers immediately followed by lower-case b (xxb) are binary values. Numbers immediately followed by lower-case h (xxh) are hexadecimal values. Hexadecimal digits that are alphabetic characters are upper case (i.e., ABCDEF, not abcdef). Hexadecimal numbers may be separated into groups of four digits by spaces. If the number is not a multiple of four digits, the first group may have fewer than four digits (e.g., AB CDEF 1234 5678h)
Storage capacities shall be reported in base-10. IO transfer sizes and offsets shall be reported in base-2. The associated units and abbreviations used in this specification are:

- A kilobyte (KB) is equal to 1,000 \( (10^3) \) bytes.
- A megabyte (MB) is equal to 1,000,000 \( (10^6) \) bytes.
• A gigabyte (GB) is equal to 1,000,000,000 (10^9) bytes.
• A terabyte (TB) is equal to 1,000,000,000,000 (10^{12}) bytes.
• A petabyte (PB) is equal to 1,000,000,000,000,000 (10^{15}) bytes.
• An exabyte (EB) is equal to 1,000,000,000,000,000,000 (10^{18}) bytes.
• A kibibyte (KiB) is equal to 2^{10} bytes.
• A mebibyte (MiB) is equal to 2^{20} bytes.
• A gibibyte (GiB) is equal to 2^{30} bytes.
• A tebibyte (TiB) is equal to 2^{40} bytes.
• A pebibyte (PiB) is equal to 2^{50} bytes.
• An exibyte (EiB) is equal to 2^{60} bytes.
5 IP Based Drive Characteristics and Requirements

Items described in this section are not exclusive. Other form factors and connectors may be added as they become available.

5.1 Physical Layer – Form Factor

Drives may be delivered in a number of different standard physical dimensions such as 3.5" or 2.5" form factors as referenced below.

- SFF-8201  2.5" Drive Form Factor Dimensions
- SFF-8223  2.5" Drive Form Factor with Serial Connector
- SFF-8301  3.5" Drive Form Factor Dimensions
- SFF-8323  3.5" Drive Form Factor with Serial Connector

5.2 Electrical

This specification is for Ethernet connected media. Other media are possible and would not be outside the definition of an IP Based Drive.

A standard for the modified pin-out of SFF-8639 (Serial Attachment 12 Gb/s 6X Unshielded Connector) is documented in SFF-9639.

IP Based Drives shall implement the pin out in the SFF 9639 (overlay of 8639) SNIA Ethernet Drive.

The interaction between chassis and drives can be described as in the following informational chart:

<table>
<thead>
<tr>
<th></th>
<th>SNIA Ethernet Drive</th>
<th>OCP Kinetic Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNIA Ethernet Drive chassis</td>
<td>Drive and/or chassis may</td>
<td>I²C on drive, but no guarantee</td>
</tr>
<tr>
<td></td>
<td>implement Power Disable and/or I²C</td>
<td>chassis supports it</td>
</tr>
<tr>
<td>OCP Kinetic Drive chassis</td>
<td>Chassis supports I²C, Drive</td>
<td>I²C mandatory on both</td>
</tr>
<tr>
<td></td>
<td>may support I²C</td>
<td></td>
</tr>
</tbody>
</table>

5.3 Link Layer

An IP Based Drive shall support IPv4 (see RFC 791) and should support IPv6 (see RFC 2460).

An IP Based Drive shall support operation at 1 Gb/s, and may support 2.5 Gb/s, 5 Gb/s, or higher in any combination. If more than one speed is supported, then an IP Based Drive shall perform speed auto-negotiation at power-on (see SFF-8601 and IEEE 802.3cb). The goal of speed negotiation is to achieve the highest speed possible by both sides of a connection.
The SFF 8601 specification defines the speed negotiation process for Ethernet drives, and the LLDP-based port auto-negotiation protocol for Ethernet drives that do not support IEEE 802.3 CL73 auto-negotiation.

**IP Based Drives:**
- Shall implement SFF 8601;
- May implement the OCP I2C management functionality defined in “Storage device with Ethernet Interface” v4 [http://files.opencompute.org/oc/public.php?service=files&t=3f4d2fedef373ae0b29a2b86f0255da1b ]; and
- May implement manual link speed assignment (run at other than negotiated speed).

### 5.4 Environmental

This document specifies a discovery and reporting mechanism to provide information on device environmental characteristics (e.g. power and temperature).

### 5.5 Taxonomy

An IP Based Drive may be defined in many forms. This taxonomy lists several forms with the idea that innovation could expand this list over time.

A tabular taxonomy is shown below.

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>Host Connect</th>
<th>Protocol Examples</th>
<th>Management Defined In…</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key-Value (KV)</strong></td>
<td>GbE</td>
<td>e.g. Kinetic/KV</td>
<td>This standard and the appropriate object protocol documentation</td>
<td>High level KV interface. Simplified host stack. Host sees object interface (drive abstracted). Lossy connectivity (TCP/IP), full routability. Multi-host/client. Failover for availability.</td>
</tr>
<tr>
<td><strong>Pre-Configured In-Storage Compute</strong></td>
<td>GbE</td>
<td>Object/other e.g. Kinetic/Ceph</td>
<td>This standard and the appropriate object protocol documentation</td>
<td>Custom apps (e.g. object) installed at factory. Apps run in an embedded environment. Lossy connectivity (TCP/IP), full routability.</td>
</tr>
<tr>
<td>User-Configured Embedded Compute</td>
<td>GbE</td>
<td>Object/other e.g. Kinetic/Ceph</td>
<td>This standard and the appropriate object protocol documentation</td>
<td>Multi-host/client. Failover for availability.</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----</td>
<td>---------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Interposer (addition to drive)</td>
<td>GbE</td>
<td>Object/other e.g. Kinetic/Ceph</td>
<td>This standard and the appropriate object protocol documentation</td>
<td>Allows standard block drives to be used as Ethernet-connected Object Drives. Allows collections of standard block drives to be virtualized as Ethernet-connected Object Drives. Lossy connectivity (TCP/IP), full routability. Multi-host/client. Failover for availability.</td>
</tr>
</tbody>
</table>

Depictions:

- Traditional Drive
  - SATA/SAS/PCIe
  - ATA/SCSI/NVMe
  - Disk and/or NVM

- Kinetic KV Drive
  - Ethernet
  - Object API
  - Disk and/or NVM
### Pre-configured In-Storage Compute

- **Ethernet**
- **Custom SW (E.g. Object API)**
- **OS + App Containers**
- **Mapping of objects to storage locations**
- **Disk and/or NVM**

### User-configured Embedded Compute

- **Ethernet**
- **Provisionable Software**
- **OS + App Containers**
- **Mapping of objects to storage locations**
- **Disk and/or NVM**
6 IP Based Drive Management
This section describes management functionality at the device level. This can be expanded to include subsystem enclosures and multi-rack configurations.

6.1 Overview
There are two aspects to device discovery and management:
1. Management Discovery;
2. Service Discovery.

Management Discovery is described in this section. Service Discovery is the responsibility of the service (application) operating on the device and is beyond the scope of this document.

Management Discovery has the following structure and sequence further described in this section:
- Establish Physical link (Ethernet, LLDP);
- Assign IP Address (DHCP);
- Discover Basic Services (DNS-Name, NTP-Time);
- Provide Management Services (Redfish).

6.2 Establish Physical Link
It is assumed that a physical Ethernet link has been established using LLDP. See section 5.3

6.3 Assign IP Address
Object Drives shall support IPv4 and shall support IPv6.

If more than one port is implemented, such ports may be configured for redundancy or other purposes.

Link aggregation, if needed may be done through LLDP and as such is not part of this document.

6.4 Discover Basic Services
The discovery of IP Based Drive basic services involves finding its access point on a network and perhaps its position in whatever enclosure it is located. It also involves determining the IP Based Drive’s capabilities. At a high level, if the drive serves up a
native protocol (Key/Value or other) the type and protocol version are important (other lower level capabilities should be discovered through the protocol itself). If the drive is capable of hosting software in a general purpose computing environment, that capability should be discoverable as well. In this case, an interoperable means to load and manage the software on the drive is required.

Redfish references SSDP (Simple Service Discovery Protocol) for discovery. IP Based Drives should use the RedFish optional discovery mechanism.

### 6.5 Redfish Manageability

The following table lists which management properties are covered by the existing Redfish specification:

<table>
<thead>
<tr>
<th>Manageable Property</th>
<th>Description</th>
<th>I2C</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Uniquely identifies the resource within the collection of like resources.</td>
<td>No</td>
<td>Device-assigned</td>
</tr>
<tr>
<td>Name</td>
<td>Human-readable name of the resource or array element.</td>
<td>No</td>
<td>Device-assigned</td>
</tr>
<tr>
<td>SystemType</td>
<td>The type of computer system represented by this resource.</td>
<td>No</td>
<td>This can be used to represent multi-tenancy and partitioning.</td>
</tr>
<tr>
<td>AssetTag</td>
<td>The user definable tag that can be used to track this computer system for inventory or other client purposes</td>
<td>No</td>
<td>Read/write</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>The manufacturer or OEM of this system.</td>
<td>Yes</td>
<td>0x02</td>
</tr>
<tr>
<td>Model</td>
<td>The model number for this system.</td>
<td>Yes</td>
<td>0x03</td>
</tr>
<tr>
<td>SKU</td>
<td>The manufacturer SKU for this system</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>SerialNumber</td>
<td>The serial number for this system</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>PartNumber</td>
<td>The part number for this system</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Provides a description of this resource and is used for commonality in the schema definitions.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>UUID</td>
<td>The universal unique identifier (UUID) for this system</td>
<td>Yes</td>
<td>Also known as World Wide ID</td>
</tr>
<tr>
<td>HostName</td>
<td>The DNS Host Name, without any domain information</td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>State</td>
<td>This indicates the known state of the resource, such as if it is enabled. The following values are defined: Enabled, Disabled, StandbyOffline, StandbySpare, InTest, Starting, Absent</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Health</td>
<td>This represents the health state of this resource in the absence of its dependent resources. The following values are defined: OK, Warning, Critical</td>
<td>No*</td>
<td>Vendor-specific in I2C</td>
</tr>
<tr>
<td>IndicatorLED</td>
<td>The state of the indicator LED, used to identify the system</td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>PowerState</td>
<td>The power state of the device</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>Reset Action</td>
<td>Yes</td>
<td>Ability to trigger a reset.</td>
</tr>
<tr>
<td>FactoryMACAddress</td>
<td>Factory-assigned MAC address</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>MacAddress</td>
<td>User-assigned MAC address</td>
<td>Yes</td>
<td>Read/Write</td>
</tr>
<tr>
<td>SpeedMbps</td>
<td>Speed of Interface</td>
<td>Yes</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Access</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>FullDuplex</td>
<td>Duplex state of the Interface</td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>FQDN</td>
<td>Fully qualified domain name</td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>IPv6DefaultGateway</td>
<td>Default Gateway for IPv6 networking</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>NameServers</td>
<td>List of Name Servers</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>IPv4Addresses</td>
<td>IP Address information for IPv4 addressing</td>
<td>Yes</td>
<td>0x08 0x12 Read/Write</td>
</tr>
<tr>
<td>IPv6Addresses</td>
<td>IP Address information for IPv6 addressing</td>
<td>Yes</td>
<td>0x08 0x12 Read/Write</td>
</tr>
<tr>
<td>VLAN</td>
<td>VLAN information</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature Sensor information</td>
<td>Yes</td>
<td>0x06</td>
</tr>
<tr>
<td>Fan Speed</td>
<td>Fan Speed Information</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>BiosVersion</td>
<td>The version of the system BIOS or primary system firmware.</td>
<td>Yes</td>
<td>0x15</td>
</tr>
</tbody>
</table>
7 Annex

7.1 Example of Redfish Manageability

Redfish represents manageable entities as a series of URLs that are discoverable from a root URL ("/redfish/v1/"). Manageable entities are discoverable in two ways, by a flat listing (systems & managers), or a hierarchical listing (chassis), which contains systems.

Given that IP Based Drives have many of the characteristics of micro-servers, even if running purpose-specific software, the closest match in the currently defined Redfish schemas is the "ComputerSystem" entity.

IP Based Drive Diagram

Here is an example Redfish JSON response for a single hypothetical IP Based Drive chassis:
GET /redfish/v1/Chassis/
{
    "@Redfish.Copyright": "Copyright © 2014-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#ChassisCollection.ChassisCollection",
    "Name": "Chassis Collection",
    "Members@odata.count": 1,
    "Members": [{
        "@odata.id": "/redfish/v1/Chassis/Kinetic"
    }]
}

This indicates that for the managed entity, there is a single chassis, called "Kinetic". This is an arbitrary identifier, and is assigned by the system being managed.

Since everything in Redfish is URL-defined and discoverable, this view (of chassis) includes links to each chassis. Moving to the "/redfish/v1/Chassis/Kinetic" URL gives us:

GET /redfish/v1/Chassis/Kinetic
{
    "@Redfish.Copyright": "Copyright © 2014-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#Chassis/Members/$entity",
    "@odata.id": "/redfish/v1/Chassis/Kinetic",
    "@odata.type": "#Chassis.1.0.0.Chassis",
    "Id": "1",
    "Name": "KineticDriveChassis",
    "ChassisType": "RackMount",
    "AssetTag": "Chicago-45Z-2381",
    "Manufacturer": "KineticSystems",
    "Model": "3500RX",
    "SKU": "8675309",
    "SerialNumber": "437XR1138R2",
    "PartNumber": "224071-J23",
    "IndicatorLED": "Lit",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Thermal": {
        "@odata.id": "/redfish/v1/Chassis/Kinetic/Thermal"
    },
    "Power": {
        "@odata.id": "/redfish/v1/Chassis/Kinetic/Power"
    },
    "Links": {
        "ComputerSystems": [{
            "@odata.id": "/redfish/v1/Systems/892374839"
        }, {
            "@odata.id": "/redfish/v1/Systems/238597323"
        }],
        "ManagedBy": [{
            "@odata.id": "/redfish/v1/Managers/2892374839"
        }]
    }
}
This provides management information related to chassis, and lists each drive installed in the chassis (Systems), and the management controller for the chassis (BMC).

The Systems list for this controller is a subset of the global systems list. In our example, there is only one Chassis, so the Redfish systems listing would contain the same entries:

```
GET /redfish/v1/Systems/
{
   "@Redfish.Copyright": "Copyright © 2014-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
   "@odata.context": "/redfish/v1/$metadata#Systems",
   "@odata.id": "/redfish/v1/Systems",
   "@odata.type": "#ComputerSystemCollection.ComputerSystemCollection",
   "Name": "Computer System Collection",
   "Members@odata.count": 1,
   "Members": [{
      "@odata.id": "/redfish/v1/Systems/892374839"
   }, {
      "@odata.id": "/redfish/v1/Systems/238597323"
   }]
}
```

Since everything in Redfish is URL-defined and discoverable, these two views (of systems) include links to each system. Moving to the "/redfish/v1/Systems/892374839" URL gives us:

```
GET /redfish/v1/Systems/892374839
{
   "@Redfish.Copyright": "Copyright © 2014-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
   "@odata.context": "/redfish/v1/$metadata#Systems/Members/$entity",
   "@odata.id": "/redfish/v1/Systems/892374839",
   "@odata.type": "#ComputerSystem.1.0.0.ComputerSystem",
   "Id": "892374839",
   "Name": "Kinetic483",
   "SystemType": "Physical",
   "AssetTag": "Chicago-45Z-2381",
   "Manufacturer": "Seagate",
   "Model": "ST4000NK0001",
   "SKU": "8675309",
   "SerialNumber": "892374839",
   "PartNumber": "224071-J23",
   "Description": "KineticObjectDrive",
   "UUID": "38947555-7742-3448-3784-823347823834",
}
```
This describes a physical system named "Kinetic483", which represents a single drive. Notable aspects includes links back to the chassis that the drive is installed in, properties for Ethernet connectivity, Simple Storage and Logging, and an example of an OEM property that allows the drive to indicate which slot it is installed in.

As an example, for Ethernet Interfaces:

GET /redfish/v1/Systems/892374839/EthernetInterfaces

{}
Here we see two Ethernet interfaces, representing the primary and secondary interface for the drive.

Looking at the first interface:

GET /redfish/v1/Systems/892374839/EthernetInterfaces/12446A3B0411

```
{
  "@odata.context": "/redfish/v1/$metadata#Systems/Members/892374839/EthernetInterfaces/Members/$entity",
  "@odata.id": "/redfish/v1/Systems/892374839/EthernetInterfaces/12446A3B0411",
  "@odata.type": "#EthernetInterface.1.0.0.EthernetInterface",
  "Id": "1",
  "Name": "EthernetInterface",
  "Description": "SystemNIC1",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "FactoryMacAddress": "12:44:6A:3B:04:11",
  "MacAddress": "12:44:6A:3B:04:11",
  "SpeedMbps": 1000,
  "FullDuplex": true,
  "HostName": "kd483",
  "FQDN": "kd483.example.com",
  "IPv6DefaultGateway": "fe80::3ed9:2bff:fe34:600",
  "NameServers": ["names.seagate.com"],
  "IPv4Addresses": [{{
    "Address": "192.168.0.10",
    "SubnetMask": "255.255.252.0",
    "AddressOrigin": "Static",
    "Gateway": "192.168.0.1"
  }},
  "IPv6Addresses": [{{
    "Address": "fe80::1ec1:deff:fe6f:1e24",
    "PrefixLength": 64,
    "AddressOrigin": "Static",
    "AddressState": "Preferred"
  }}],
  "Links": {
    "VLANs": {
      "@odata.id": "/redfish/v1/Systems/892374839/EthernetInterfaces/12446A3B0411/VLANs"
    },
    "Oem": {}"