



Advancing storage &  
information technology

## Software Defined Storage

---

Mark Carlson, Alan Yoder, Leah Schoeb,  
Don Deel, Carlos Pratt

April, 2014  
**Working DRAFT**



*Publication of this Working Draft for review and comment has been approved by the SNIA TC. This draft represents a “best effort” attempt by the SNIA TC 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/>.*

Copyright © 2014 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners

### 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:

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

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](mailto: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.

### 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/>.



---

## Table of Contents

---

<b>Introduction .....</b>	<b>4</b>
Attributes of Software Defined Storage.....	4
Differentiation of Software Defined Storage.....	4
Necessary Software Defined Storage Functionality.....	5
The User’s View of Software Defined Storage.....	5
The Role of Metadata .....	6
The Software Defined Storage Big Picture .....	8
Metadata in CDMI .....	9
SDS as an integral component of the Software Defined Data Center.....	10
<b>About the SNIA.....</b>	<b>11</b>

---

## List of Figures

---

Figure 1: Traditional, Manual conveyance of Data Requirements .....	6
Figure 2: The addition of Metadata for conveyance of Data Requirements .....	7
Figure 3: The Big Picture of Software Defined Storage .....	8



## Introduction

Software Defined Storage (SDS) has been proposed (ca. 2013) as a new category of storage software products. SDS can be an element within a Software Defined Data Center but can also function as a stand-alone technology. The term Software Defined Storage is a marketing ‘buzzword’ that is a follow-on to the term Software Defined Networking, which was first used to describe an approach in network technology that abstracts various elements of networking and creates an abstraction or virtualized layer in software. There is also work going on to define Software Defined Compute. The software defined approach abstracts and simplifies the management of networks into virtual services. In networking, the control plane and the data plane have been intertwined within the traditional switches that are deployed today, making abstraction and virtualization more difficult to manage in complex virtual environments. Network capabilities are now just catching up with capabilities that have been offered in the storage industry for over a decade. SDS does represent a new evolution for the storage industry for how storage will be managed and deployed in the future.

## Attributes of Software Defined Storage

The following are attributes of SDS that are typically seen in the market:

- May allow customers to “build it themselves,” providing their own commodity hardware to create a solution with the provided software.
- May work with either arbitrary hardware or may also enhance the existing functions of specialized hardware.
- May also enable the scale-out of storage (not just the scale up typical of big storage boxes).
- Nearly always includes the pooling of storage and other resources.
- May allow for the building of the storage and data services “solution” incrementally.
- Incorporates management automation.
- Includes a self service interface for users.
- Includes a form of service level management that allows for the tagging of metadata to drive the type of storage and data services applied. The granularity may be large to start, but is expected to move to a finer grained service level capability over time.
- Allows administrators to set policy for managing the storage and data services.
- Enables the dis-aggregation of storage and data services.

Some analysts and vendors contend that SDS must be hosted on heterogeneous block storage. This is not the SNIA's position, which is platform-independent. The SNIA definition of SDS allows for both proprietary and heterogeneous platforms. What is necessary to meet the SNIA definition is that the platform offers a self-service interface for provisioning and managing virtual instances of itself.

## Differentiation of Software Defined Storage

The aspect of SDS that differentiates it is apparent from how some of the products are deployed. It is that Data Services can be executed either in servers, storage, or both spanning the historical



boundaries of where they execute. This has potential impacts on security and reliability, and may be an interesting revival for Direct Attached Storage (DAS) in some cases.

While SDS builds on the virtualization of the Data Path, SDS is not virtualization alone. The Control Path needs to be abstracted as a service as well. The storage service interface needs to allow the data owner to express requirements on both the data, and the needed service level requirements. A Cloud, a Data Center, a Storage system or a Data Administrator can be used to implement the interface.

### Necessary Software Defined Storage Functionality

Since many storage offerings today have already been abstracted and virtualized, what capabilities should be offered to claim the title of Software Defined Storage?

Software Defined Storage should include:

- **Automation** – Simplified management that reduces the cost of maintaining the storage infrastructure.
- **Standard Interfaces** – APIs for the management, provisioning and maintenance of storage devices and services.
- **Virtualized Data Path** – Block, File and Object interfaces that support applications written to these interfaces.
- **Scalability** – Seamless ability to scale the storage infrastructure without disruption to availability or performance.

Ideally, SDS offerings allow applications and data producers to manage the treatment of their data by the storage infrastructure without the need for intervention from storage administrators, without explicit provisioning operations, and with automatic service level management.

In addition, Data Services should be able to be deployed dynamically and policies should be used to maintain service levels and match the requirements with capabilities. Metadata should be used to:

- Express Requirements
- Control the Data Services
- Express Service Level Capabilities

### The User's View of Software Defined Storage

An application or storage user's view of SDS includes both a data path and a control path. The data path consists of a combination of previously standardized block, file and object interfaces for which applications have been developed, but what about the control path?

Nearly all storage that is currently deployed requires a storage administrator to create virtual storage devices (Block Storage Logical Units, Filesystem Shares, Object Containers) for the application to use. Behind the scenes, the storage administrator is deploying data services for the data that is stored on these devices. In the majority of cases, each data service requires its own administration interface. Changing those data services affects all the data stored on those virtual devices and can become error



prone and fragile. Communication of the requirements for that data is usually out of band of any storage interface, and is made directly to the storage administrator, as is shown in the Figure below:

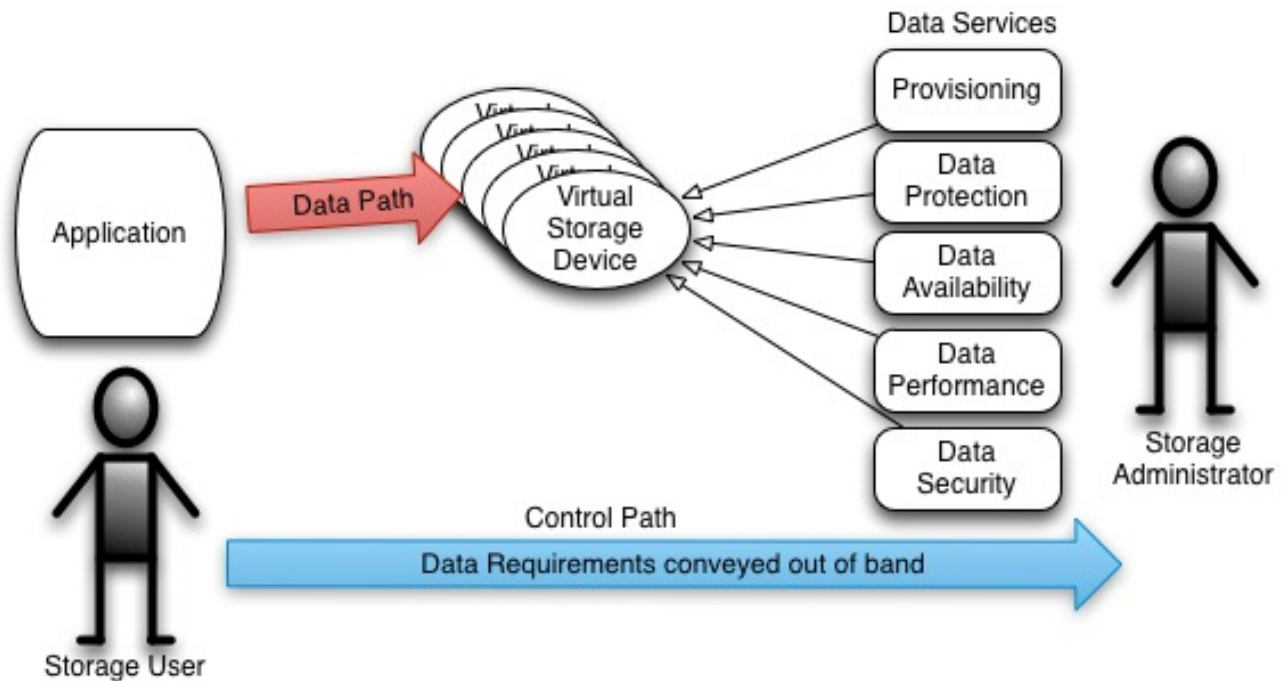


Figure 1: Traditional, Manual conveyance of Data Requirements

As can be seen in the above figure, this storage is not very “software defined”, but more “storage administrator” defined and implemented. This leads to a high total cost of ownership for storage due to a lack of automation.

### The Role of Metadata

In order to introduce automation into the storage infrastructure and reduce the costs due to manual administration, there needs to be a way to convey the data requirements directly to the automation software. The granularity of these requirements needs to at least be at the level of individual virtual storage devices as is common today. To prepare for future automation, however, each data object should be able to convey its own requirements independent of which virtual storage device it resides on. The objects should be grouped and abstracted to let the “user” understand their choices, otherwise the user will need to be a storage expert.

## Software Defined Storage



In order to convey the requirements to the storage system, the application or user needs to mark each file or object with those requirements. Metadata, or “data” about data, is the perfect mechanism for this purpose. By marking the data object with metadata, which documents the requirements, the storage system can address those requirements with the data services as show below:

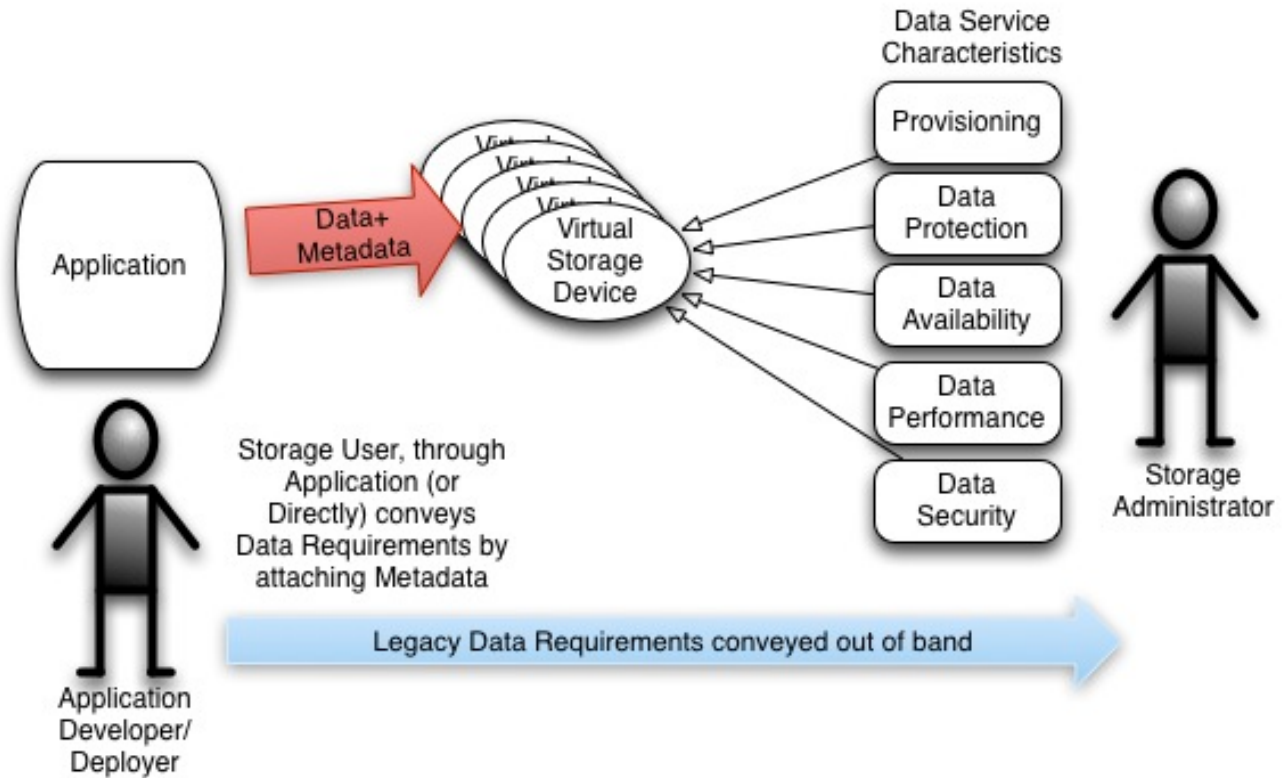


Figure 2: The addition of Metadata for conveyance of Data Requirements

The requirements may still be conveyed out of band of the data path, but automation can eventually obviate this need. With SDS, the Storage Administrator can start to move to higher level tasks such as defining policies, rather than spending time fixing immediate problems that cause service levels to degrade.





## The Software Defined Storage Big Picture

Putting it all together, the following diagram illustrates the concepts behind Software Defined Storage:

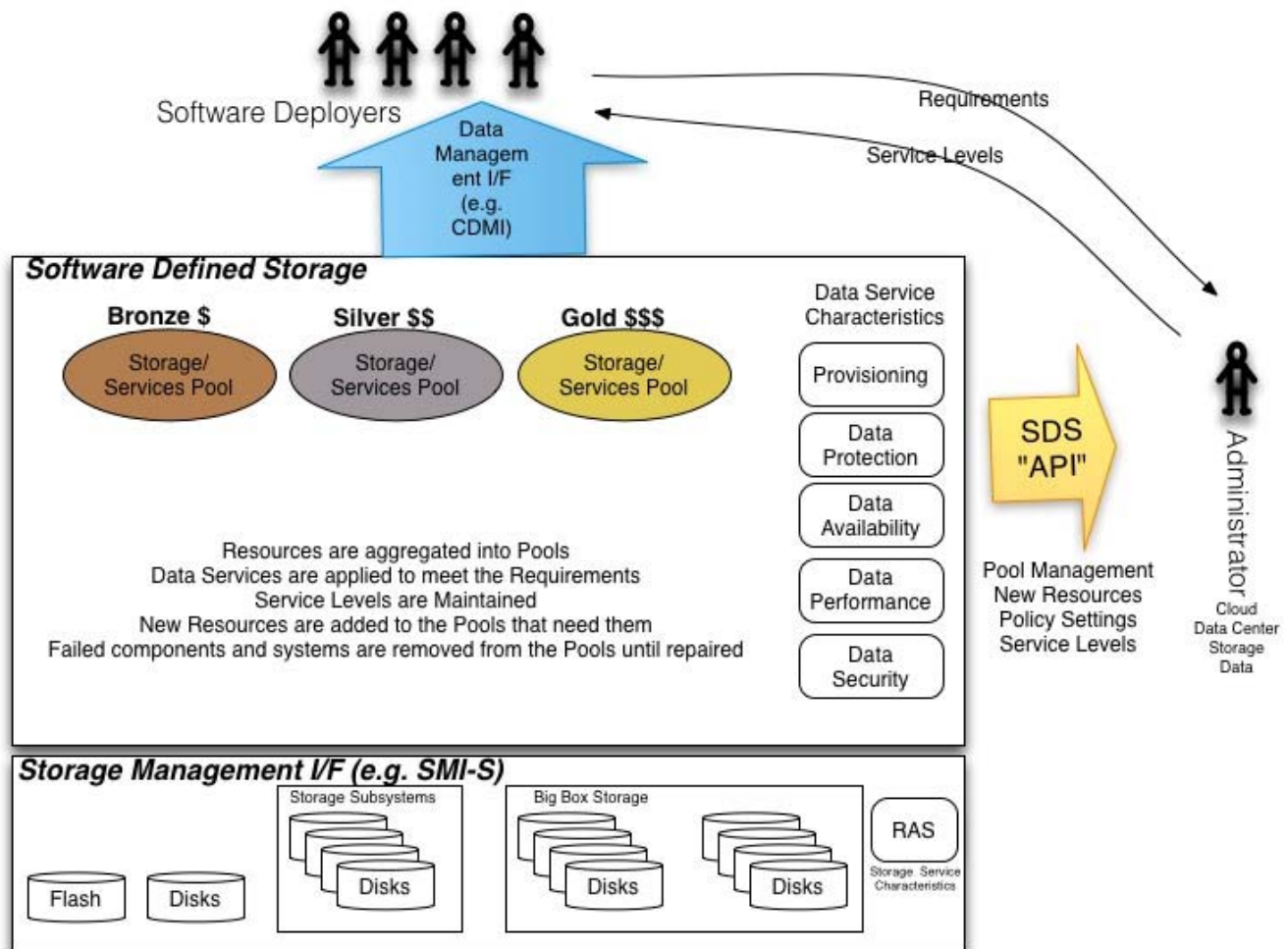


Figure 3: The Big Picture of Software Defined Storage

Software Deployers work through a Data Management interface (such as CDMI) to convey their requirements for the data they own. They also receive the desired service levels through a combination of the SDS solution and the Administrators.

For now, SDS aggregates the resources into Pools. The Data Service characteristics are applied to the data in order to meet the service level requirements and are thus maintained. New resources are added to the Pools that need them and failed components and systems are removed from the Pools until repaired.

SDS prefers a standardized storage management interface (such as SMI-S) in order to automate the management of the storage resources and discover their capabilities for use in various pools. However,





legacy storage management interfaces are common today, and predicting their demise is premature. Additionally there are emerging open source APIs that are becoming a de facto storage management standard, an example is OpenStack Cinder.

Lastly, SDS enables the administrators to work with abstract interfaces that let them manage pools, assign new resources, set up policies and determine service levels.

### Metadata in CDMI

The Cloud Data Management Interface (CDMI) uses many different types of metadata, including HTTP metadata, data system metadata, user metadata, and storage system metadata. To address the requirements of enterprise applications and the data managed by them, this use of metadata allows CDMI to deliver simplicity through a standard interface. CDMI leverages previous SNIA standards such as the eXtensible Access Method (XAM) for metadata on each data element. In particular, XAM has metadata that drives retention data services useful in compliance and eDiscovery.

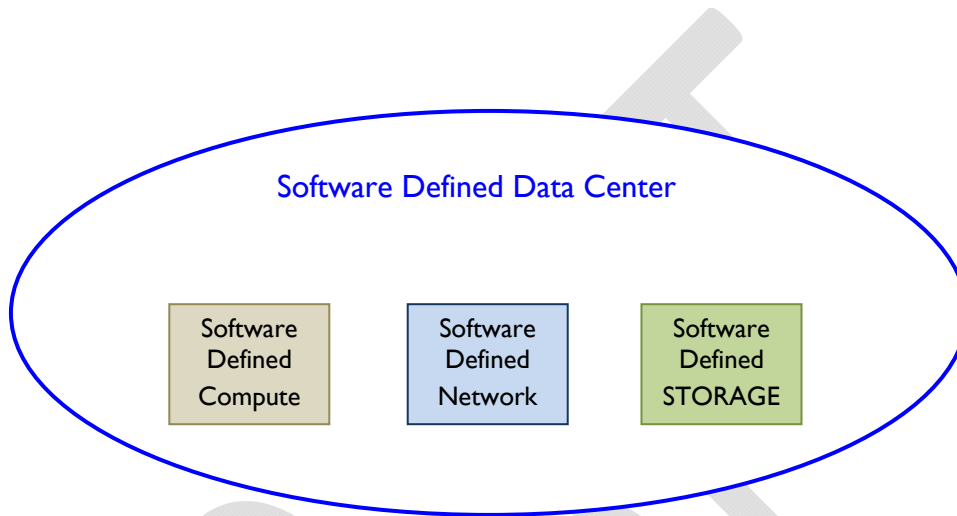
CDMI's use of metadata extends from individual data elements and can apply to containers of data, as well. Thus, any data placed into a container essentially inherits the data system metadata of the container into which it was placed. When creating a new container within an existing container, the new container would similarly inherit the metadata settings of its parent container. Of course, the data system metadata can be overridden at the container or individual data element level, as desired.

The extension of metadata to managing containers, not just data, enables a reduction in the number of paradigms for managing the components of storage – a significant cost savings. By supporting metadata in a cloud storage interface standard and proscribing how the storage and data system metadata is interpreted to meet the requirements of the data, the simplicity required by the cloud storage paradigm is maintained, while still addressing the requirements of enterprise applications and their data.



### SDS as an integral component of the Software Defined Data Center

A question that many systems administrators have is: Where does SDS fit in my data center? A simple answer is to look at the Software Defined Data Center (SDDC) as the brain of the hardware infrastructure that inevitably is behind a cloud or part of a more traditional and older data center. From a high level view SDDC is comprised of three components as shown in the figure below.



The Software Defined Compute is a virtualized computer environment that offers the processing layer of the SDDC. The Software Defined Network provides a less complex environment for its management. The Software Defined Storage offers a less complex method of managing storage. All three are needed to have a well-tuned working software defined data center. In short it can be said that SDS is an integral part of SDDC.



### About the 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](http://www.snia.org).

DRAFT