Document Identifier: DSP-IS0501

Date: 2014-10-23

Version: 1.0.1c

Software Defined Data Center (SDDC) Definition A White Paper from the OSDDC Incubator

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, it may still change, perhaps profoundly. This document is available for public review and comment.

Provide any comments through the DMTF Feedback Portal: http://www.dmtf.org/standards/feedback

Document Type: White Paper

Document Status: Work in Progress - Not a DMTF Standard

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Software Defined* Data Center (SDDC) Definition (DSP-IS0501) was prepared by the Open Software Defined Data Center (OSDDC) Incubator.

The goal of the OSDDC Incubator is to develop SDDC use cases, reference architectures and requirements based on real world customer requirements. Based on these inputs the Incubator will develop a set of whitepapers and set of recommendations for industry standardization for the SDDC.

The work coming out of this incubator will result in:

1) A clear definition and scope of the SDDC concept.
2) New work items to existing chartered working groups.
3) Expanded scope to existing chartered groups
4) Creation of new working groups, if needed.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

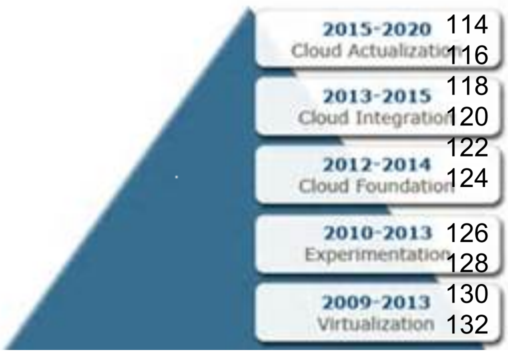
- Ali, Ghazanfar - ZTE Corporation
- Black, David - EMC
- Bumpus, Winston - VMWare, Inc.
- Carlson, Mark – DMTF Fellow
- Dolin, Rob - Microsoft Corporation
- Khasnabish, Bhumip – ZTE
- Leung, John - Intel
- McDonald, Alex - NetApp
- Ronco, Enrico - Telecom Italia
- Snelling, David - Fujitsu
- Shah, Hemal - Broadcom
- Wells, Eric - Hitachi, Ltd.
- Wheeler, Jeff - Huawei
- Zhdankin, Alex - Cisco

Software Defined Data Center (SDDC) Definition

1 Executive summary

1.1 Introduction

The virtualization and cloud industry continue their evolution with the most recent settling point being the 'Software Defined Data Center (SDDC)'.



While the SDDC is an evolutionary result of virtualization and cloud computing technologies, the term itself (SDDC) was only coined recently. The reader should find it interesting that the initial definition did not declare the emergence of the 'Software Defined *Cloud*' but rather the 'Software Defined *Data Center*'.

To date, the SDDC has been defined in many ways. The following examples are a few of the more prevalent (and realistic) definitions gleaned from a large number of resources used for this paper:

> *"A Software Defined Data Center (SDDC) is a data storage facility in which all elements of the infrastructure – networking, storage, CPU and security – are virtualized and delivered as a service. Deployment, provisioning, configuration and the operation, monitoring and automation of the entire infrastructure is abstracted from hardware and implemented in software."*

Another:

> *"SDDC is the phrase used to refer to a data center where the entire infrastructure is virtualized and delivered as a service."*

Regardless of the definition, it is clear that the move to the SDDC is the major technology shift of this decade. While other definitions have been proposed by various vendors and standards development organizations (SDOs), they all have similar, if not identical, intent or wording. Very few definitions of an SDDC actually offer any substantial or comprehensive information that a person seeking to understand just what exactly an SDDC is would find useful.

The balance of this paper will present evidence that there is a major difference between cloud computing and an SDDC and that each is a separate collection of technologies, products, and services.

1.2 SDDC definition

Software Defined Data Center (SDDC): a pool of compute, network, storage and other resources that can be dynamically discovered, provisioned and configured based on workloads.

SDDC provides a programmatic abstraction that enables policy-driven orchestration of workloads as well as measurement and management of resources consumed.

SDDC is comprised of a set of features that include:

a. A pool of compute, network, storage and other resources
b. Discovery of resource capabilities
c. Automated provisioning of logical resources based on workload requirements
d. Measurement and management and of resources consumed
e. Policy-driven orchestration of resources to meet SLOs of the workloads

2 SDDC technology and functionality

SDDC incorporates and is heavily dependent upon the use of topologies that abstract, pool, and automate the use of the virtualized resources. Virtualization technologies can be thought of as a commodity, or common resources when integrated and used by SDDC. The focus on industry standardized management models and application programming interfaces (APIs) provides this level of abstraction. Various vendors and SDOs are championing their respective offerings into the new SDDC community.

Core SDDC features and functionalities include:

- Abstraction of compute, network, and storage resources
- Virtualization of network resources and services
- Image automation and library support for templates
- Topology automation and standardization
- Virtualization of object, block and file storage
- Topology centric services for traditional ‘edge’ features like security, IDS / HIDS, AAA, Firewall, Load balancing and so on

The SDDC should be:

- Standardized - at the API and functional model aspects initially
- Holistic - by using the abstractions from the hardware layer provided by the SDDC functionality
- Adaptive - with elasticity being more directed and rooted by and in the Business logic
- Automated - in provisioning, configuration, operational and run-time management aspects

2.1 SDDC virtualization, Cloud and relationships

Virtualization is central to the SDDC and is necessary but not sufficient. The three major building blocks that virtualization delivers are: network virtualization, compute virtualization, and storage virtualization. Software defined builds upon virtualization and provides an abstracted functionality.

There are three primary components to Virtualization that carry over to the SDDC:

1. Storage Virtualization – enables the pooling of physical storage facilities and devices from various physical networked devices into what appears to be a single storage pool managed by a centralized management service/console.
2. Compute Virtualization (or server virtualization) - incorporates the masking, or abstracting of the underlying collection of physical server resources from the end user/consumer. This concept includes the abstracting of the number and identity of physical servers, associated processors, memory and operating systems. The abstraction allows the complexity of the underlying infrastructure to be hidden from the user/consumer though this complexity is still required to be managed by someone, most likely the provider.
3. Network Virtualization - represents the most difficult of all areas contributing to the SDDC solutions. The virtualization of network resources combines the available network resources (services, bandwidth, LAN, WAN, VLANs, Security, etc.) into a resource pool that provides subsets of the whole to virtual machines as the physical networks provide these to physical

servers. The use of network virtualization in Cloud and SDDC is lagging the other two primary areas largely due to the complexity, vendor proprietary technologies, various standards and methods in place today in physical network environments.

Control of the SDDC is automated by software. Management of the SDDC is different than management of the physical Data Center. A business logic layer is required to integrate and translate application requirements, SLAs, policies, and other legacy considerations.

SDDC differs from Cloud and Virtualization in these ways:

- SDDC is not defined nor is it focused on a standardized IT solution. Aspects of Cloud and Virtualization are standardized with cross-SDO work driving them as well. Only the DMTF currently has a focus on SDDC as an SDO. Various consortia and forums are beginning to discuss the needs for a standardized approach but there are none of these in a position of creating or driving an SDDC to a national or international standard and specification.
- SDDC builds upon the successes of Server Virtualization, broadening the individual components of the Data Center (DC) that have been virtualized, and envisioning a unified control console/management solution.
- Cloud is a relatively new IT operational model (and marketing model) focusing on the delivery and consumption of IT Services. Even the underlying complexities of the physical and virtualized environments are abstracted from the consumer (as in PaaS and SaaS today).
- SDDC extends this operational model by further refining and expanding upon the three traditional delivery models of cloud computing; that is, infrastructure, platform, and software as a service (IaaS, PaaS and SaaS respectively).

SDDC does not simplify the complexity or management of the physical DC environment.

- The physical Data Center (pDC) will still be the major underlying component for any virtualized, Cloud or SDDC solution, regardless of vendor. The pDC will still be required as the basis for the virtualized and Cloud services. The provider, carrier or intermediary will still have all of the complexity of managing and operating the pDC as they do today. An SDDC, however, may enable more efficient usage of pDC.
- Many of the improvements brought about by the focus on Cloud and SDDC are actually taking place in the physical data center infrastructure like data center fabric.
- The physical hardware underlying the SDDC and Cloud is becoming ‘commoditized’ by processor and network equipment manufacturers that allow for faster and simpler Cloud and SDDC environments that can be managed by centralized tools.

The use of IaaS, PaaS and SaaS has led to a need for greater operational efficiencies and a more abstract management software layer than can be provided by Cloud.

- De facto vendors in Cloud are looking to provide SDDC with a greater scope than the scope of services that can be delivered by Cloud.
- SDDC does indeed compete in the traditional sense with PaaS and SaaS and will do more so as consumers adopt further Private and Hybrid Clouds.
- Cloud cannot deliver on the promise of full mobility and BYOD (bring-your-own-device), whereas SDDC can for any enterprise consumer.

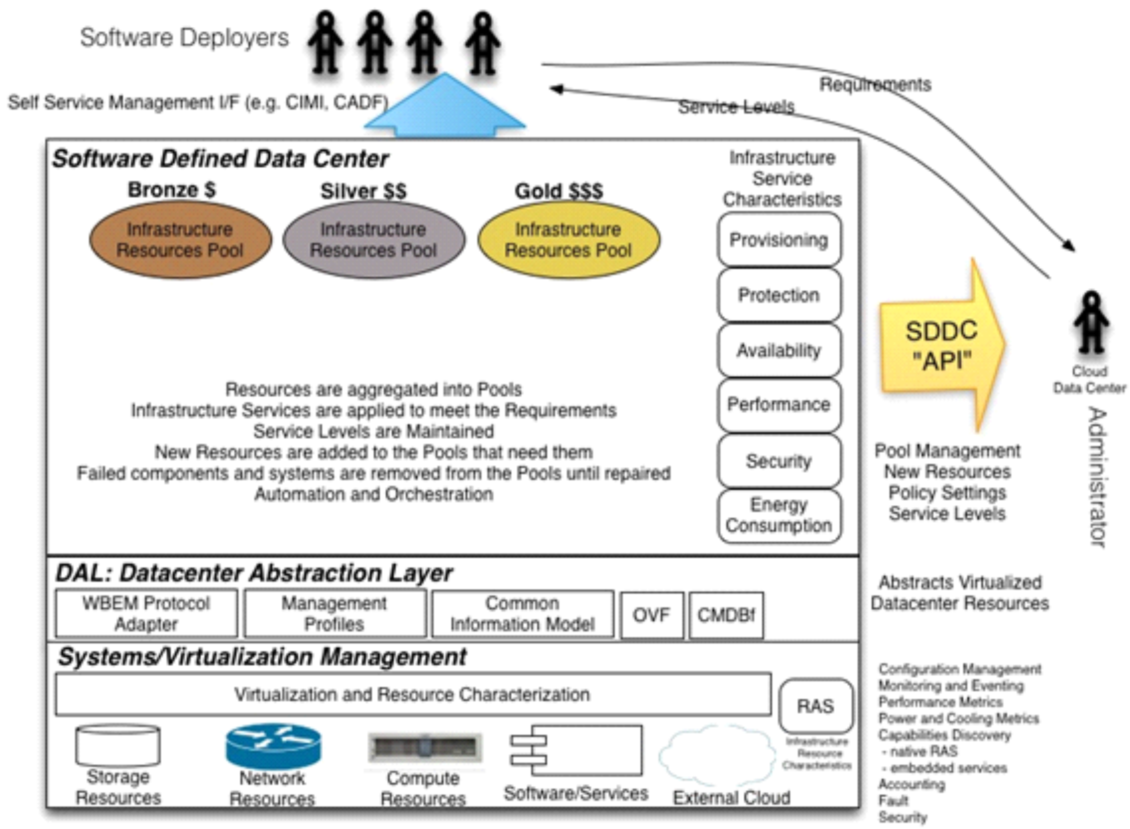


Figure 1 – Software Defined Data Center architecture

An SDDC architecture defines data center resources in terms of software. Specifically, it releases compute, network, and storage from hardware limitations and increases service agility. This architecture can be considered an evolution from server virtualization to complete virtualization of the data center.

2.2 Server virtualization

Server virtualization releases CPU and memory from the limitations of underlying physical hardware. As a standard infrastructure technology, server virtualization is the basis of the SDDC, which extends the same principles to all infrastructure services.

2.3 Software Defined Network

In a Software Defined Network (SDN), the network control plane is moved from the switch to the software running on a server. This move improves programmability, efficiency, and extensibility. SDN is to date the most developed and understood software-defined technology.., Thereforethis paper does not delve into the details of this software defined component.

2.4 Software Defined Storage

Software Defined Storage (SDS) is an emerging ecosystem of products and requires further discussion here. . This software should make visible all physical and virtual resources and enables programmability and automated provisioning based on consumption or need. SDS separates the control plane from the data plane and dynamically leverages heterogeneity of storage to respond to changing workload demands. The SDS enables the publishing of storage service catalogs and enables resources to be provisioned on-demand and consumed according to policy.

In many respects, SDS is more about packaging and how IT users think about and design data centers. Storage has been largely software defined for more than a decade: the vast majority of storage features have been designed and delivered as software components within a specific storage-optimized environment.

The SNIA definition(need a reference to SNIA whitepaper here) 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.

2.4.1 Necessary Software Defined Storage functionality

Because 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

2.5 Data center abstraction layer

Data centers are complex as they contain a wide variety of devices (compute, storage, networks, power management, etc.) often from multiple vendors and are often managed by using vendor proprietary solutions.. There are today standards and mechanisms for managing all the devices and we call this the Data center Abstraction Layer (DAL). The DAL provides a set of standards to abstract this complexity:

- Increased cost.
    - Increased people cost due to added complexity that result in the need to spend more on training. As a result the IT budget shifts from vendorsto system integrators and in-house staff.
    - Management applications and skills need to be updated every time a new device/vendor is brought in.
- Higher operational cost due to inconsistent management technologies, standards, and different security/application models.

- Higher chance for errors and downtime due to the inconsistencies listed above, which impact the ability to automate.

Less Agility:

- – Fewer choices in hardware due to high cost of entry for new independent hardware vendors (IHVs) to compete with existing proprietary ecosystems.
- – Onboarding a new device requires updating management applications and processes, which reduces the agility in onboarding new devices and vendors.
- – Inconsistent management technologies results in a complex and tightly coupled data center architecture. Any change in one layer or one element often requires changes in multiple other layers/elements. This complexity results in an environment where change cannot be done rapidly.
- – Hinders the ability to manage the fabric and the data center as a single entity. Becomes hard to orchestrate change across heterogeneous environment.

The DAL name was inspired by HAL (the Hardware Abstraction Layer). Twenty years ago, the industry got together to solve a very common problem: “How do we abstract the hardware layer from the application and services that the OS provides?”

The idea was to define the elements that should be abstracted, and then develop the necessary protocols and standards to manage and interact with these elements. After new elements plug in to HAL, the OS layer would know how to deal with them. The HAL provided a consistent interface for the operating system and applications to interface with the hardware devices without worrying about which provider the devices came from. This solution reduced the overall cost of PCs and also provided great agility/choice in selection of hardware devices.

The HAL is the right abstraction when working with a single PC or a single server. Thinking around the same lines as HAL, we should do the same thing with the data center. We should abstract the elements in the data center and make them available as a set of standards resources to the software defined layers in the SDDC. DAL in essence is “abstracting the underlying resources in the context of a data center”.

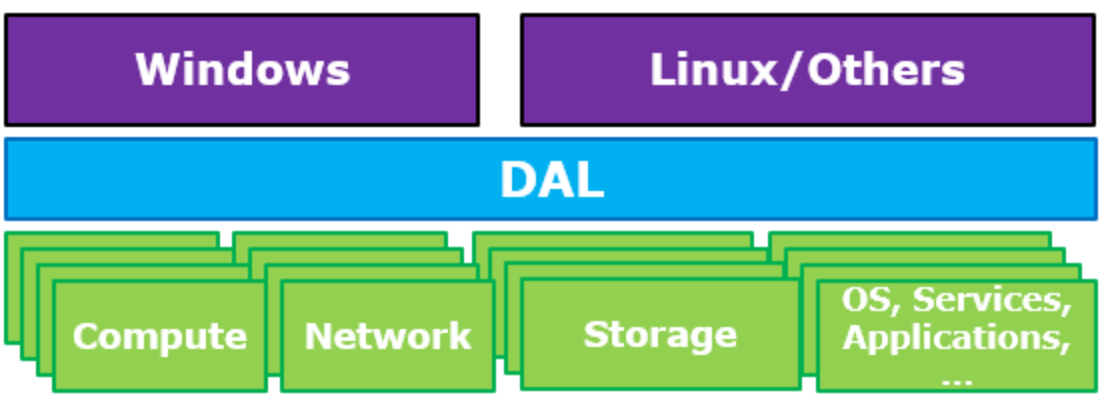


Figure 2 – Data center Abstraction Layer

The DAL approach enables

- devices to participate in data center management by implementing standard interfaces,

- higher level management applications to manage devices in a data center in a consistent manner (using DMTF standards based protocol (such as WS-MAN) and a consistent model (such as CIM)) and without requiring any device-specific changes in management applications.

3 Barriers to SDDC adoption

There are many barriers to the adoption of SDDC in the current virtualization and Cloud industry by providers, brokers, and consumers. A few of the key ones are listed in this clause.

3.1 General challenges

There is a complete industry built around management solutions for existing data centers including certifications (CCIE, MCSE, etc.,) and compliance/conformance solutions. The existing base of custom and complex management software on both the provider and consumer sites presents barriers to moving into SDDC.

- Industry and global standards for physical DCs that extend to the equipment penetration point of the consumer. It is very difficult to parse this responsibility if the logical/virtual/SDDC topology does not align up with the physical or virtual.
- Specialized hardware costs and deployments are understood to add value.such as solutions like Fiber Channel for SANs. Providers are not willing to abandon their current infrastructure components if they do not natively support SDDC for the promise of market share or revenue that might be a long time in coming.
- The necessary isolation of workloads, users, and services, as well as logical and virtual devices provided by today's current implementations. This level of intelligence and service will be abstracted out to the software layer if the SDDC pundits have their way. The likelihood of this happening quickly is not a viable assumption.
- Support for multi-tenancy to the hardware level. Because SDDC will not have a control or management plane that affects/effects the hardware level, SDDC will struggle to establish and maintain the level of isolation and security that an existing pDC afford today.

There is not a one-for-one mapping of the features and functionality provided by the virtualization and Cloud domains into the SDDC domain. SDDC is not lockstep marching with Cloud, but diverges at even the initial stages. Cloud computing did not require a serious look at applications or software re-engineering but SDDC does if the SDDC is to be used optimally.

3.2 Applications/services and SDDC

One of the more difficult functional challenges that SDDC is inheriting from Cloud is the area of 'Applications'. The main areas of difficulty challenging users, vendors, and providers in respect to applications and SDDC are as follows:

- **Mobility** - The introduction of application mobility by the Cloud. Applications are moved between systems, hosts, racks, chassis, pods, sites, and geographies with their Virtual Machine context in order to provide application and resource scaling and elasticity. In order to address these issues the SDDC providers and consumers will have to:
  - modify the underlying application code by directly adding the capabilities for state management across the physical and virtual resources; or
  - provide synthetic socket calls that directly intercept the applications communications with the SDDC and redirect to appropriate code allowing the necessary resources and services for applications mobility; or
  - add 'shims' or proxy layers between the applications and the stock/standardized socket calls that the applications use. These shims or proxies will filter the appropriate information

to and from the applications and underlying virtual resources to provide fundamental applications mobility in an SDDC environment.

- **Common APIs** - The lack of common application to SDDC or application to Cloud APIs. Most APIs coming from SDOs today are focused on fundamental IaaS enablement and management, and are not providing application to SDDC capabilities.
- **Interoperability and Federation** - The inability today for an application to seamlessly operate across multiple Cloud or SDDC environments to use necessary resources from each. In order to accomplish this feature today in Clouds the provider must implement a wide range of applications and management solutions.
- **Standardization** - The lack of standardized means to provide for application creation and use of ‘mashups’ in the Cloud or SDDC environments. In order to run natively in any SDDC the application will have to be more of a composite of other applications than a silo of a single fork or tree of code.

While the use of SDDC is supposed to free up the application layer from the hardware layer, the SDDC does introduce both new and complex functionality for the application layer.

3.3 Authorization and authentication requirements

In this section we discuss authorization and authentication requirements for SDDC. The following are some of the major topics.

- Data, content, and media authenticity: Association and identification of data to its owner (user, enterprise consumer, service provider, location, etc.) and access privileges.
- Role-based and privilege-based access to video surveillance content and alarm notifications.
- Perimeter security of the virtualized data center operations and real-time insight into security issues to the provider and to the enterprises using their services.
- Business-hours-based security monitoring of provider assets.
- Control for customers during self service - ability for customers to maintain effective control of their workloads even though the protection mechanisms and even the locations of workloads may not be known to customers.
- Protection of virtual machines, network traffic, actual/residual data, and other resources of a tenant against unauthorized access by another tenant.

3.4 Privacy and security requirements

In this section we discuss privacy and security requirements for SDDC. The main concern here is the management of the life cycle of data, including data privacy and security while in use, in motion, or at rest within a virtualized infrastructure environment.

- Data while in use: (a) Isolation of data while in use by the computing resources, and (b) Management of the data usage based on access privileges of the users, enterprise consumer, and service providers.
- Data in motion: Restriction of the data transmission across geographical boundaries based on government regulations or enterprise policies and configurations defined during self-service setup.

3.5 Data at rest (monitoring and management): (a) Data isolation in a multi-tenant environment to protect against side attack (across tenants) or admin attacks; (b) Data migration managed as defined by enterprise/government

policies; (c) Deletion, loss/leakage, and location of data.Audit, verification, and regulatory requirements

In this clause, we discuss audit, verification, and regulatory (both domestic and international) requirements for SDDC. The following points need consideration beyond the traditional requirements:

- Governance, risk, and compliance: (a) Clear certification and accreditation guidelines; (b) Clear e-discovery guidelines; (c) Virtualization audit assurance and log sensitivity management; (d) Need for clarity on how the NIST SP 800-53-style control guides (http://csrc.nist.gov/publications/nistpubs/800-53A-rev1/sp800-53A-rev1-final.pdf) can work in virtualized environment; (e) Need of clear guidelines for privacy, and lawful interception in the virtualized service environment.
- Backup and recovery of information (import/export across multiple service providers).
- Business continuity and disaster recovery: How to maintain continuity of operations by having redundancy: (a) within the same provider, and (b) across multiple service providers?

4 Standards activity

4.1 DMTF standards work

DMTF standards enable effective management of IT environments through well-defined interfaces that collectively deliver complete management capabilities. DMTF standard interfaces are critical to enabling interoperability among multi-vendor IT infrastructures, and systems and network management including cloud computing, virtualization, desktop, network, servers and storage.

Some of the key DMTF standards and initiatives under development that will enable the new SDDC paradigm are described below.

4.1.1 Open SDDC Incubator

The DMTF is the only SDO currently that is focusing on developing initial management models for the SDDC marketplace. The DMTF recently launched its ‘SDDC Incubator’ with the charter of directing all future work in the DMTF for SDDC.

4.1.2 Virtualization Management

DMTF’s Virtualization Management (VMAN) initiative includes a set of specifications and profiles that address the management life cycle of a heterogeneous virtualized environment.

4.1.3 Cloud Management

Technologies like cloud computing and virtualization are rapidly being adopted by enterprise IT managers to better deliver services to their customers, lower IT costs, and improve operational efficiencies.

DMTF's Cloud Management Initiative is focused on developing interoperable cloud infrastructure management standards and promoting adoption of those standards in the industry. The work of DMTF working groups promoted by the Cloud Management Initiative is focused on achieving interoperable cloud infrastructure management between cloud service providers and their consumers and developers.

Cloud Infrastructure Management Interface (CIMI)

CIMI is a self-service interface for infrastructure clouds, allowing users to dynamically provision, configure, and administer their cloud usage with a high-level interface that greatly simplifies cloud systems management. The specification standardizes interactions between cloud environments to achieve interoperable cloud infrastructure management between service providers and their consumers and developers, enabling users to manage their cloud infrastructure use easily and without complexity.

Open Virtualization Format (OVF)

The OVF specification provides a standard format for packaging and describing virtual machines and applications for deployment across heterogeneous virtualization platforms, OVF was adopted by the American National Standards Institute in August 2010.[4] OVF was adopted as an International Standard in August 2011 by the Joint Technical Committee 1 (JTC 1) of the International Organization for Standardization (ISO), and the International Electrotechnical Commission (IEC).[1] In January 2013, DMTF released the second version of the standard, OVF 2.0, which applies to emerging cloud use cases and provides important developments from OVF 1.0 including improved network configuration support and package encryption capabilities for safe delivery.

Web-Based Enterprise Management (WBEM)

WBEM defines protocols for the interaction between systems management infrastructure components implementing the Common Information Model (CIM), and is a major component of the DAL, The CIM Schema is a conceptual schema that defines how the managed elements in an IT environment (for instance computers or storage area networks) are represented as a common set of objects and relationships between them. CIM is extensible in order to allow product specific extensions to the common definition of these managed elements. CIM uses a model based upon UML to define the CIM Schema. CIM is the basis for most of the other DMTF standards.

Configuration Management Database Federation (CMDBf)

CMDBf facilitates the sharing of information between configuration management databases (CMDBs) and other management data repositories (MDRs). The CMDBf standard enables organizations to federate and access information from complex, multi-vendor infrastructures, simplifying the process of managing related configuration data stored in multiple CMDBs and MDRs.

Systems Management Architecture for Server Hardware (SMASH)

DMTF's SMASH standards are a suite of specifications that deliver architectural semantics, industry standard protocols and profiles to unify the management of the data center. The SMASH Server Management (SM) Command Line Protocol (CLP) specification enables simple and intuitive management of heterogeneous servers in the data center. SMASH takes full advantage of the DMTF's Web Services for Management (WS-Management) specification - delivering standards-based Web services management for server environments. Both provide server management independent of machine state, operating system state, server system topology or access method, facilitating local and remote management of server hardware. SMASH also includes the SM Managed Element Addressing Specification, SM CLP-to-CIM Mapping Specification, SM CLP Discovery Specification, SM Profiles, as well as a SM CLP Architecture White Paper.

4.2 Other related work

Standards-related work in the SDDC arena is still new and work in other SDOs mainly focused on SDN, not SDDC. It is important to look at emerging standards from other SDOs and how they may be relevant to SDDC. Some of these are listed below.

4.2.1 OASIS - Cloud Application Management for Platforms (CAMP)

The OASIS CAMP advances an interoperable protocol that cloud implementers can use to package and deploy their applications. CAMP defines interfaces for self-service provisioning, monitoring, and control. Based on REST, CAMP is expected to foster an ecosystem of common tools, plug-ins, libraries, and frameworks, which will allow vendors to offer greater value-add.

Common CAMP use cases include:

- moving on-premises applications to the cloud (private or public)
- redeploying applications across cloud platforms from multiple vendors

4.2.2 OASIS - Topology and Orchestration Specification for Cloud Applications (TOSCA)

The TOSCA TC substantially enhances the portability of cloud applications and the IT services that comprise them running on complex software and hardware infrastructure. The IT application and service level of abstraction in TOSCA will also provide essential support to the continued evolution of cloud computing. For example, TOSCA would enable essential application and service life cycle management support, e.g., deployment, scaling, patching, etc., in Software Defined Environments (SDE), such as Software Defined Data Centers (SDDC) and Software Defined Networks (SDN).

TOSCA facilitates this goal by enabling the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behavior of these services (e.g., deploy, patch, shutdown) independent of the supplier creating the service, and any particular cloud provider or hosting technology. TOSCA enables the association of that higher-level operational behavior with cloud infrastructure management.

TOSCA models integrate the collective knowledge of application and infrastructure experts, and enable the expression of application requirements independently from IaaS- and PaaS-style platform capabilities. Thus, TOSCA enables an ecosystem where cloud service providers can compete and differentiate to add value to applications in a software defined environment.

These capabilities greatly facilitate much higher levels of cloud service/solution portability, the continuous delivery of applications (DevOps) across their life cycle without lock-in, including:

- Portable deployment to any compliant cloud
- Easier migration of existing applications to the cloud
- Flexible selection and movement of applications between different cloud providers and cloud platform technologies
- Dynamic, multi-cloud provider applications

4.2.3 SNIA - Cloud Data Management Interface (CDMI)

The SNIA Cloud Data Management Interface (CDMI) is an ISO/IEC standard that enables cloud solution vendors to meet the growing need of interoperability for data stored in the cloud. The CDMI standard is applicable to all types of clouds – private, public, and hybrid. There are currently more than 20 products that meet the CDMI specification.

CDMI provides end users with the ability to control the destiny of their data and ensure hassle-free data access, data protection, and data migration from one cloud service to another.

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.

4.2.4 ETSI/ISG – Network Function Virtualization (NFV)

The first use case of ETSI/ISG NFV discusses NFV Infrastructure as a Service (NFVIaaS), which may have a lot of similarity with SDDC. The NFVI includes compute, networking, and storage infrastructure in virtualized forms. NFVIaaS calls for combining and interconnecting network as a service (NaaS), and other compute/storage Infrastructure as a Service (IaaS) in order to provide virtual network function (VNF) to the network administrators. The VNFs from different administrative domains can be interconnected and clustered for developing an end-to-end service. The NFV use case document is available at the following URL:

http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf.

4.2.5 IETF/IRTF

There are a few IETF and IRTF working/research groups (WGs/RGs) and drafts that discuss Virtual Data Center (VDC). The concept of VDC and the service that can be offered by using VDC are very similar to the SDDC concept that we discuss here in this paper.

The NVO3 (Network Virtualization Overlays/Over-Layer-3) Working Group (WG) focuses on developing interoperable solution for traffic isolation, address independence, and virtual machine (VM) migration in Data Center Virtual Private Network (DCVPN).

DCVPN is defined as a VPN that is viable across a scaling range of a few thousand VMs to several million VMs running on more than 100,000 physical servers. DCVPN supports several million endpoints and hundreds of thousands of VPNs within a single administrative domain. Further details about IETF NVO3 activities can be found at http://datatracker.ietf.org/wg/nvo3/charter/.

The SCIM (System for Cross-domain Identity Management) WG is developing the core schema and interfaces based on HTTP and REST for creating, reading, searching, modifying, and deleting user identities and identity-related objects across administrative domains.

Initial focus areas of the SCIM WG are developing a core schema definition, a set of operations for creation, modification, and deletion of users, schema discovery, read and search, bulk operations, and mapping between the inetOrgPerson LDAP object class (RFC 2798) and the SCIM schema. Further details on IETF SCIM activities can be found at http://datatracker.ietf.org/wg/scim/charter/.

The SDN (Software-Defined Networking) Research Group (RG) is currently focusing on developing definition and taxonomy for SDN. Future work may include a study of model scalability and applicability, multi-layer programmability and feedback control system, network description languages, abstractions, interfaces and compilers, and security-related aspects of SDN. Further details about IRTF SDN activities can be found at https://irtf.org/sdnrg.

4.2.6 Open Networking Foundation (ONF)

ONF has developed a southbound interface (SBI; south of the controller) called OpenFlow™ in order to enable remote programming of the flow forwarding.

Currently ONF is focusing on Software Defined Networking (SDN) related issues especially the concepts, frameworks, and architecture.

The network segmentation, multi-path multi-tenancy support, and security-related activities of the Forwarding Abstraction WG, Northbound Interface (NBI) WG, Configuration and Management WG, Layer 4-7 Services DG, and Security DG may be very helpful for open SDDCs and their interconnections.

4.2.7 Open DayLight (ODL)

ODL focuses on control and programmability of the abstracted network functions and entities. The objective is to develop northbound interfaces (NBIs) for gathering network intelligence including performing analytics, and then use the controller to orchestrate adaptive new rules throughout the network for efficient automated operations. Detailed technical overview of ODL initiatives is available at http://www.opendaylight.org/project/technical-overview.

ODL supports OpenFlow and other protocols as SBIs, and released Base (Enterprise), Virtualization, and Service Provider editions of the software packages (http://www.opendaylight.org/software).

4.2.8 Open Data Center Alliance (ODCA)

ODCA initiatives and activities are focused on developing open, interoperable solutions for secure cloud federation, automation of cloud infrastructure, common management, and transparency of cloud service delivery.

5 Conclusion

To realize an SDDC, data center resources, such as compute, network, and storage, are expressed as software. They also need to have certain characteristics, such as multi-tenancy; rapid resource provisioning; elastic scaling; policy-driven resource management; shared infrastructure; instrumentation; and self-service, accounting, and auditing. This ultimately entails a programmable infrastructure that enables valuable resources to be automatically cataloged, commissioned, decommissioned, repurposed, and repositioned.

6 References

S. Karavettil et al, “Security Framework for Virtualized Data Center Services, IETF discussion draft (http://tools.ietf.org/id/draft-karavettil-vdcs-security-framework-05.txt), June 2013.\

Add a reference to SNIA SDDS

7 Glossary

Table 1 – Glossary of terms

Acronym or Phrase	Definition	Explanation
AAA	Authentication, Authorization, and Auditing	
API	Application Programming Interface	
Block storage		

Acronym or Phrase	Definition	Explanation
BYOD	Bring Your Own Device	
Cloud	Cloud Computing	
Fiber Channel		
File storage		
Firewall		The three major areas of concern in system security
IaaS	Infrastructure as a Service	An interface used by an application program to request services. The term API is usually used to denote interfaces between applications and the software components that compose the operating environment (e.g., operating system, file system, volume manager, device drivers, etc.) Source: http://www.snia.org/education/dictionary/a
IDS	Intrusion Detection System	Storage organized and allocated in blocks of fixed size.
HIDS	Host Intrusion Detection Systems	The policy of permitting employees to bring personally owned mobile devices (laptops, tablets, and smart phones) to their workplace, and to use those devices to access privileged company information and applications Source: http://en.wikipedia.org/wiki/Bring_your_own_device
LAN	Local Area Network	
Load Balancing		A high-speed LAN technology, most commonly used for SAN's.
Metadata		
NAS	Network Attached Storage	A device, often implemented in software, to control data flows between two or more networks. Firewalls typically reject network traffic that does not originate from trusted address and/or ports and thus provides a degree of isolation between networks.

Acronym or Phrase	Definition	Explanation
Object storage		
PaaS	Platform as a Service	A system used to detect unauthorized access to resources.
pDC	Physical Data Center	An IDS specifically designed to protect host systems.
SaaS	Software as a Service	
SAN	Storage Area Network	A mechanism used to distribute demands for resources amongst those available. Usually used in reference to processing resources but may be applied to any resource.
SDDC	Software Defined Data Center	
SDN	Software Defined Network	
SDO	Standards Development Organization	
SDS	Software Defined Storage	
Virtual Appliance		
VLAN	Virtual LAN	
WAN	Wide area network	A storage system consisting of storage elements, storage devices, computer systems, and/or appliances, plus all control software, communicating over a network. Source: http://www.snia.org/education/dictionary/s#storage_area_network
Copyright		
	SNIA	http://www.snia.org/education/dictionary/s
	Wikipedia	Creative Commons Attribution-Sharealike 3.0 Unported License

ANNEX A
(informative)

Change log

Date	Author	Comments
2014-03-07	Hemal Shah	Initial draft
2014-04-03	Winston Bumpus	Added DMTF Standards
2014-06-13	Working Session	Merged updates and comments – Draft 9
2014-06-19	Bhumip Khasnabish	Added requirements and SDO overviews
2014-06-24	Eric Wells	Glossary & formatting
2014-09-29	Working Session	Edits for 1.0.1c