



文書識別番号: DSP-IS0501

日付: 2015/11/23

バージョン: 1.0.0

1  
2  
3  
4

5 **ソフトウェア・デファインド・データ・センター(SDDC)の定**  
6 **義**  
7 **OSDDC インキュベーターによるホワイト・ペーパー**

- 8 廃止文書: なし  
9 文書の種類: 参考仕様  
10 文書の位置付け: 発行  
11 文書の言語: ja-JP

参考仕様は DMTF 標準ではなく、変更されることがある。

12

13 著作権情報

14 Copyright © 2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

15 重要:この仕様は標準ではない。調査および情報文書であり、業界のフィードバックを得るために作成されたもの  
16 である。必ずしも、DMTF の見解やそのすべてのメンバーの見解が反映されているわけではない。この成果物の  
17 全部または一部は、将来の標準で情報として考慮される場合がある。

18 DMTF は、企業やシステムの管理および相互運用性を推進することに力を注いでいる、業界のメンバーから成る  
19 非営利団体である。メンバー、およびメンバー以外でも、出典を正しく表示することを条件に、DMTF の仕様と文  
20 書を複製することができる。DMTF の仕様は時折改定されることがあるため、特定のバージョンおよび公開日に、  
21 常に注意を払う必要がある。

22 本標準または標準案の特定の要素を実装することは、仮特許権を含む第三者の特許権(本書では「特許権」  
23 と呼ぶ)の対象となることもある。DMTF は本標準のユーザーに対し、上記権利の存在について何ら表明するも  
24 のではなく、上記第三者の特許権、特許権者または主張者の、いずれかまたはすべてを認識、公開、または特定  
25 する責任を負わない。また、上記権利、特許権者、主張者の不完全または不正確な特定、公開に対しても責任を  
26 負わない。DMTF は、いかなる相手に対して、いかなる方法または環境、またいかなる法論理においても、上記  
27 の第三者特許権を認識、公開、または特定しないことに対し何ら責任を負わず、上記第三者の標準に関する信  
28 頼性、またはその製品、プロトコル、試験方法論に組み込まれた標準に関しても何ら責任を負わない。DMTF は、  
29 上記標準の実装が知見できるか否かにかかわらず、上記標準を実装するいかなる相手に対しても、また、いかな  
30 る特許権者または主張者に対しても、何ら責任を負わない。また、DMTF は、公開後に標準が撤回または修正さ  
31 れることにより生じるコストや損失に対し何ら責任を負わず、また、標準を実装するいかなる相手からも、上記実  
32 装に対して特許権者が起こす、いずれかまたはすべての侵害の主張から何ら損害を受けず、免責されるものと  
33 する。

34 第三者が保有する特許権であって、DMTF 標準の実装に関連するかまたは影響を与える可能性があるの特許  
35 権者が考え、すでに DMTF に通知済みである特許権に関する情報については、サイト  
36 <http://www.dmtf.org/about/policies/disclosures.php> を参照のこと。

37 この文書の標準言語は英語である。他の言語への翻訳を許可する。

## 目次

39	序文 5	
40	1 要旨	6
41	1.1 はじめに	6
42	1.2 SDDC の定義	6
43	2 ユースケース	6
44	2.1 インフラストラクチャー・アズ・ア・サービス (IaaS)	7
45	2.1.1 関係者	7
46	2.1.2 ユースケース	8
47	2.2 ソフトウェア・アズ・ア・サービス (SaaS)	8
48	2.2.1 関係者	8
49	2.2.2 ユースケース	9
50	3 SDDC の技術と機能	10
51	3.1 SDDC、仮想化、およびクラウドどうしの関係	10
52	4 SDDC のアーキテクチャー	10
53	4.1 サーバーの仮想化	12
54	4.2 ソフトウェア・定義・ネットワーク	12
55	4.3 ソフトウェア・定義・ストレージ	12
56	4.3.1 ソフトウェア・定義・ストレージに必要な機能	12
57	4.4 データ・センター抽象化レイヤー	13
58	4.5 信頼の境界とマルチテナント分離の要件	13
59	5 標準に関連する活動	14
60	5.1 DMTF	14
61	オープンな SDDC インキュベーター	14
62	クラウド管理イニシアチブ	14
63	ネットワーク管理イニシアチブ	14
64	仮想化管理イニシアチブ	14
65	5.1.1 クラウド・インフラストラクチャー管理インターフェース (CIMI)	15
66	5.1.2 オープン仮想化フォーマット (OVF)	15
67	5.1.3 Web ベースのエンタープライズ環境の管理 (WBEM)	15
68	5.1.4 共通情報モデル (CIM)	15
69	5.1.5 構成管理データベース連携 (CMDBf)	15
70	5.1.6 サーバー・ハードウェア向けシステム管理アーキテクチャー (SMASH)	15
71	5.1.7 Redfish API	16
72	5.2 OASIS	16
73	5.2.1 Cloud Application Management for Platforms (CAMP)	16
74	5.2.2 Topology and Orchestration Specification for Cloud Applications (TOSCA)	16
75	5.3 SNIA	17
76	5.3.1 Cloud Data Management Interface (CDMI)	17
77	5.3.2 Storage Management Initiative	17
78	5.4 その他の SDO	18
79	5.4.1 ETSI/ISG - Network Function Virtualization (NFV)	18
80	5.4.2 IETF/IRTF	18
81	5.4.3 Open Networking Foundation (ONF)	18
82	5.4.4 Open DayLight (ODL)	19
83	5.4.5 Open Data Center Alliance (ODCA)	19
84	6 標準間の相違 - 何が欠落しているのか	19
85	6.1 メトリクスの標準	19
86	6.2 アプリケーションおよび作業負荷の管理	19
87	6.3 ポリシーおよびサービス・レベル	19

88	7	結論 .....	20
89	8	参照資料 .....	20
90	9	用語集 .....	20
91			
92		<b>図の一覧</b>	
93		図 1 - SDDC での IaaS のユースケース .....	7
94		図 2 - SDDC での SaaS のユースケース .....	8
95		図 3 - SDDC のアーキテクチャー .....	11
96		図 4 - データ・センター抽象化レイヤー .....	13
97			
98		<b>表の一覧</b>	
99		表 1 - 用語集 .....	20
100			

## 序文

101

102 ソフトウェア・デファインド・データ・センター(SDDC)の定義(DSP-IS0501)はオープン・ソフトウェア・デファインド・  
103 データ・センター(OSDDC)インキュベーター(Open Software Defined Data Center Incubator)により編纂され  
104 た。

105 OSDDC インキュベーターの目標は、実際の顧客の要求に基づいて SDDC のユースケース、参照アーキテク  
106 チャー、および要件を作成することである。これらの情報に基づき、インキュベーターは各種の関連ホワイト・ペー  
107 パーおよび SDDC の業界標準化に向けた各種推奨事項を作成する。

108 インキュベーターによるこれらの作業から、以下の成果が得られる。

- 109 1. SDDC の概念の明確な定義と適用範囲
- 110 2. 既存の公認作業部会に提示する新たな作業項目
- 111 3. 既存の公認部会に提示する拡張された適用範囲
- 112 4. 新たな作業部会を必要とする場合はその新設

113 DMTF は、企業やシステムの管理および相互運用性を推進することに力を注いでいる、業界のメンバーから成る  
114 非営利団体である。DMTF の詳細については、<http://www.dmtf.org> を参照のこと。

## 謝辞

116 DMTF は、この文書の作成に貢献した以下の諸氏に感謝するものである。

- 117 • Ali, Ghazanfar (ZTE Corporation)
- 118 • Black, David (EMC)
- 119 • Bumpus, Winston (VMware, Inc.)
- 120 • Carlson, Mark (Toshiba America Electronic Components, Inc.)
- 121 • Dolin, Rob (Microsoft Corporation)
- 122 • Fayed, Wassim (Microsoft Corporation)
- 123 • Khasnabish, Bhumi (ZTE)
- 124 • Leung, John (Intel)
- 125 • McDonald, Alex (NetApp)
- 126 • Ronco, Enrico (Telecom Italia)
- 127 • Snelling, David (Fujitsu)
- 128 • Shah, Hemal (Broadcom)
- 129 • Wells, Eric (Hitachi, Ltd.)
- 130 • Wheeler, Jeff (Huawei)
- 131 • Zhdankin, Alex (Cisco)

132

# 133 ソフトウェア・デファインド・データ・センター(SDDC)の定義

## 134 1 要旨

### 135 1.1 はじめに

136 ソフトウェア・デファインド・データ・センター(SDDC)は、仮想化技術とクラウド・コンピューティング技術の進化に  
137 よる成果である。これまでに、さまざまな方向性で SDDC という用語が定義されてきた。このホワイト・ペーパーに  
138 使用した多数の資料に見られる定義のうち、頻繁に使用されている(また、現実的な)いくつかの定義例を以下に  
139 示す。

140 「ソフトウェア・デファインド・データ・センター(SDDC)とは、インフラストラクチャーを構成するすべての要素  
141 (ネットワーキング、ストレージ、CPU およびセキュリティ)を仮想化し、サービスとして提供するデータ・スト  
142 レージ施設である。このインフラストラクチャー全体のデプロイメント、プロビジョニング、構成、動作、監視お  
143 よび自動化は、ハードウェアから抽象化され、ソフトウェアに実装される。」(Forrester)

144 別の例:

145 「SDDC とは、インフラストラクチャー全体を仮想化して、サービスとして提供するデータ・センターを指す語句  
146 である。」(VMware)

147 SDDC への移行が、大きな技術変化であることは明らかである。これらの他にもさまざまなベンダーから定義が  
148 提案されているが、いずれも類似の趣旨によるものである。

149 本書の目的は、SDDC のユースケースと定義の概要を述べ、複数の標準の間に存在する相違を明らかにし、  
150 SDDC のさまざまな実装のアーキテクチャーを示すことにある。

### 151 1.2 SDDC の定義

152 ソフトウェア・デファインド・データ・センター(Software Defined Data Center; SDDC): 論理計算機能、ネットワー  
153 ク、ストレージなどのリソースをプログラムで抽象化し、ソフトウェアとして表現したもの。これらのリソースは、作業  
154 負荷の要件に応じて動的に検出、プロビジョニング、および構成される。このような方法で、SDDC は作業負荷の  
155 ポリシー主導型オーケストレーションおよびリソース利用量の数値化と管理を実現する。

156 SDDC は、以下をはじめとする各種機能で構成される。

- 157 • 論理計算機能、ネットワーク、ストレージなどのリソース
- 158 • リソースの検出機能
- 159 • 作業負荷の要件に基づいた論理リソースの自動プロビジョニング
- 160 • リソース利用量の数値化および管理
- 161 • 作業負荷のサービス要件への適合を目的とした、リソースのポリシー主導型オーケストレーション

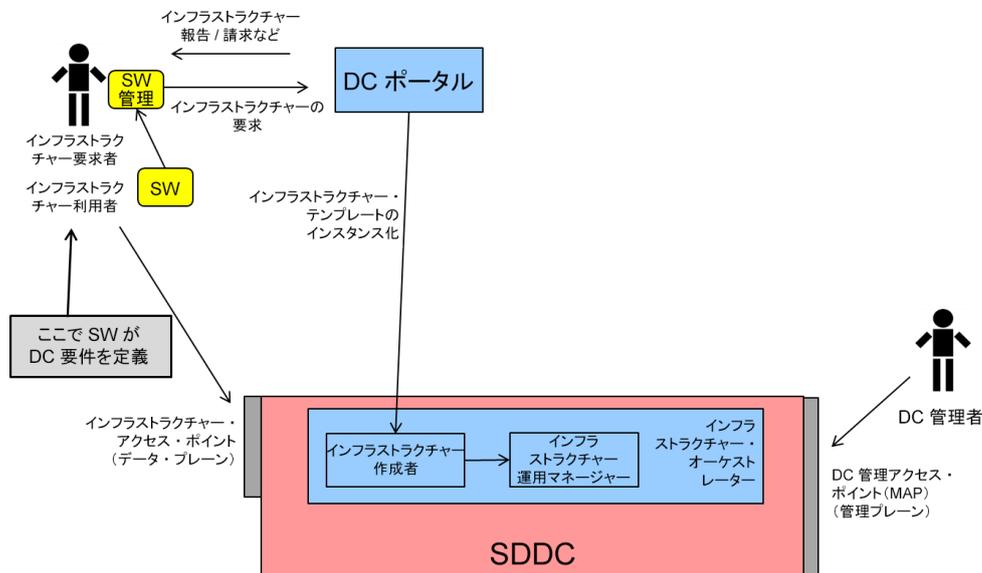
## 162 2 ユースケース

163 この項では、インフラストラクチャー・アズ・ア・サービス(IaaS)やソフトウェア・アズ・ア・サービス(SaaS)など、  
164 SDDC によって提供可能なさまざまなサービスのユースケースについて説明する。

## 165 2.1 インフラストラクチャー・アズ・ア・サービス (IaaS)

166 IaaS では、作業負荷の実行を望む顧客がデータ・センターを利用し、インフラストラクチャーをホストする。インフ  
167 ラストラクチャーが利用可能になると、顧客は必要なソフトウェアおよびコンテンツやデータをインストールして作  
168 業負荷を実行する。

169 図 1 に、ソフトウェア・デファインド・データ・センターを基盤とした IaaS 環境で発生する相互作用を示す。



170

171

図 1 - SDCC での IaaS のユースケース

### 172 2.1.1 関係者

173 このユースケースにおける関係者は、顧客および IaaS データ・センター (DC) 管理者の 2 者である。顧客は、I  
174 ンフラストラクチャー要求者とインフラストラクチャー利用者という 2 つの側面を持つ。

175 インフラストラクチャー要求者は、以下のタスクを実行する。

- 176 • 特定の計算/ストレージ・トポロジーで実行する作業負荷で構成したアプリケーションの設計
- 177 • 特定の作業負荷要件を備えたインフラストラクチャーの要求
- 178 • インフラストラクチャー (ファームウェアや BIOS など) の検証
- 179 • そのインフラストラクチャーの規模の増減の要求
- 180 • 利用量レポートと請求書の受領

181 インフラストラクチャー利用者は、以下のタスクを実行する。

- 182 • OS およびアプリケーションのインストールとコンテンツの提供
- 183 • 作業負荷の実行開始

184 IaaS DC 管理者は、以下のタスクを実行する。

- 185 • データ・センター内の電源と空調の監視
- 186 • データ・センターへのプラットフォーム/リソースの追加 (または交換)
- 187 • リソースの消耗 (または余剰) に関する通知の受領
- 188 • 棚卸しの実施 (アカウント、SW ライセンスなど)
- 189 • セキュリティ監査の実施 (またはセキュリティ請負会社が実施)
- 190 • 電圧低下の可能性に関する通知の受領
- 191 • プラットフォーム・ファームウェアの更新 (セキュリティなど)

## 192 2.1.2 ユースケース

193 作業負荷を把握しているのはインフラストラクチャー要求者である。

194 上の図で、このフローは以下のように進行する。

- 195 1. インフラストラクチャー要求者は作業負荷(WL)を調査し、要求するインフラストラクチャーを決定する。
- 196 2. インフラストラクチャー要求者は、特定のサービス要件を満足するインフラストラクチャーをサービス・
- 197 ポータルから要求する。
- 198 3. サービス・ポータルは、インフラストラクチャー・オーケストレーターに対して、インフラストラクチャーのイ
- 199 ンスタンス化を要求する。
- 200 4. インフラストラクチャー・オーケストレーターは、指定されたインフラストラクチャーをインスタンス化する。
- 201 5. インフラストラクチャーは、指定したインフラストラクチャーを稼働させる。この時点で、インフラストラク
- 202 チャーは運用フェーズに移行し、インフラストラクチャー運用マネージャーによって管理される。
- 203 6. インフラストラクチャー利用者は、稼働を開始したインフラストラクチャーを利用できるようになる。

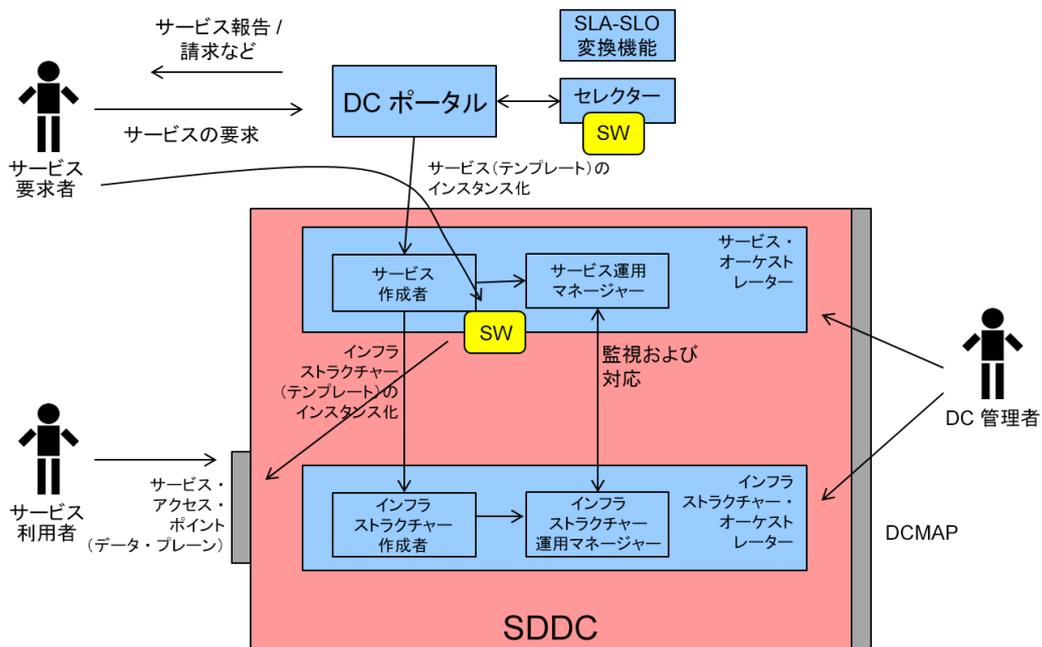
## 204 2.2 ソフトウェア・アズ・ア・サービス(SaaS)

205 SaaS では、サービスのインスタンス化を望む顧客がデータ・センターを利用してそのサービスをホストする。その

206 サービスは、SaaS 顧客とは別のサービス利用者が利用することもある。サービスがインスタンス化された後、そ

207 のサービスが有効になって利用できる状態になる前に、サービス要求者が追加のコンテンツを用意することが必

208 要な場合もある。



209

210

図 2 - SDCC での SaaS のユースケース

211 図 2 に、ソフトウェア・デファインド・データ・センターを基盤とした SaaS 環境で発生する相互作用を示す。

## 212 2.2.1 関係者

213 関係者は、サービス要求者、サービス利用者、および SaaS DC 管理者の 3 者である。

214 サービス要求者は、サービスのインスタンス化を必要として以下のタスクを実行する。

- 215       • 特定のサービス要件を備えたサービスの要求
- 216       • サービスの監視
- 217       • 運用サービスのサービス要件の変更
- 218       • サービス規模の拡張または縮小の要求
- 219       • 別のサービス提供者へのサービス移行の要求
- 220       • サービス終了の要求

221 サービス利用者は、以下のタスクを実行する。

- 222       • サービスの利用

223 SaaS DC 管理者は、以下のタスクを実行する。

- 224       • サービスの監視
- 225       • データ・センター内の電源と空調の監視
- 226       • データ・センターへのプラットフォーム/リソースの追加(または交換)
- 227       • リソースの枯渇(または余剰)に関する通知の受領
- 228       • 棚卸しの実施(アカウント、SW ライセンスなど)
- 229       • セキュリティ監査の実施(またはセキュリティ請負会社が実施)
- 230       • 電圧低下の可能性に関する通知の受領
- 231       • 新しいサービスのステージング/テスト
- 232       • プラットフォーム・ファームウェアの更新(セキュリティなど)

### 233 2.2.2 ユースケース

234 サービス・インフラストラクチャーを定義する作業負荷は、DC サービス・ポータルと呼ばれる。上の図で、このフ  
235 ローは以下のように進行する。

- 236       1. サービス要求者は、特定のサービス要件を満足するインフラストラクチャーをサービス・ポータルから要  
237       求する。
- 238       2. 複数のサービス・テンプレートが利用可能である場合、サービス・ポータルまたはサービス要求者は、そ  
239       の中から特定のサービス・テンプレートを選択できる。
- 240       3. サービス・ポータルは、サービス・オーケストレーターに対し、目的とするサービスのインスタンス化を要  
241       求する。
- 242       4. サービス作成者は、インフラストラクチャー・オーケストレーターに対し、目的とするインフラストラク  
243       チャーのインスタンス化を要求する。
- 244       5. インフラストラクチャーがインスタンス化された後、サービス作成者は OS、アプリケーション、およびコン  
245       テンツをインストールし、それに応じて必要な構成を実行する。
- 246       6. 最後に、サービス作成者がサービスを稼働し、サービス利用者はサービスを利用できるようになる。
- 247       7. この時点で、インフラストラクチャーとサービスの両方が運用フェーズに移行し、それぞれの運用マネー  
248       ジャーによって管理される。

### 249 3 SDDC の技術と機能

250 SDDC は、仮想化したリソースの使用を抽象化および自動化し、必要に応じてプールするトポロジーを組み込み、  
251 重点的に使用している。SDDC に統合して使用する場合の仮想化技術は、共通リソースと考えることができる。  
252 業界標準の管理モデルとアプリケーション・プログラミング・インターフェース(API)に重点を置くことで、このレベ  
253 ルの抽象化が得られる。さまざまなベンダーおよび SDO が、それぞれの商品を新しい SDDC コミュニティに  
254 提供するべく努力している。

255 SDDC は、以下をはじめとする各種機能で構成される。

- 256 1. 論理的な計算、ネットワーク、ストレージ、その他のリソース
- 257 2. リソース性能の検出機能
- 258 3. 処理の要件に基づいた論理リソースの自動プロビジョニング
- 259 4. リソース利用量の数値化および管理
- 260 5. 処理のサービス要件への適合を目的とした、リソースのポリシー主導型オーケストレーション

261 その他の SDDC の特長および機能として、以下のものが挙げられる。

- 262 • トポロジーの自動化
- 263 • セキュリティ(認証、許可、監査)、侵入検知システム(IDS)、侵入防止システム(IPS)、ファイア  
264 ウォール

265 SDDC には、以下の要素が必要である。

- 266 • 標準化: API および機能モデル
- 267 • 総体性: システム規模の抽象化
- 268 • 適応性: 処理を主体とした柔軟性
- 269 • 自動化: プロビジョニング、構成、および実行時管理

#### 270 3.1 SDDC、仮想化、およびクラウドどうしの関係

271 仮想化は SDDC の中心となる技術であるが、それだけでは十分とはいえない。仮想化が提供する 3 つの主要な  
272 構成要素は、計算、ストレージ、およびネットワークである。

- 273 1. 計算の仮想化: 基盤となる物理サーバー・リソースの集合で実現できる計算リソースの抽象化。この概  
274 念には、物理サーバー、プロセッサ、およびメモリの数、種類、および識別情報の抽象化が含まれ  
275 る。コンテナなどの他の技術も使用できる。
- 276 2. ストレージの仮想化: 基盤となる物理ストレージ・リソースおよび論理ストレージ・リソースで実現できるス  
277 トレージ・リソースの抽象化。この概念には、物理ディスクの数、種類、および識別情報の抽象化が含ま  
278 れる。
- 279 3. ネットワークの仮想化: 基盤となる物理リソースおよび論理リソースを使用して実現できるネットワーク・  
280 リソースの抽象化。この概念には、物理メディア、接続、およびプロトコルの数、種類、および識別情報  
281 の抽象化が含まれる。

### 282 4 SDDC のアーキテクチャー

283 標準の API を使用し、仮想化技術に基づいて実現する SDDC 自動化により、ソフトウェアに必要なリソースを正  
284 確にプロビジョニングし、それらのリソース上にソフトウェアを配備できるようになる。これは、図 3 の 2 つの最下  
285 位レイヤーである、データ・センター抽象化レイヤー(Data center Abstraction Layer; DAL)および仮想化とリ  
286 ソースの特性化レイヤーに示されている。この自動化の目的は、配備されるソフトウェアの要件を解釈し、それら  
287 の要件を満たすようにリソースを構成することである。今日では、これらの要件がアウトオブバンドで管理者に伝

288 達されることが一般的である。この場合は、管理者がこれらの要件を解釈する必要がある、なお、API を通じてこ  
 289 れらの要件を伝達することもできる。その API の実装によってこれらの要件を解釈し、本来は管理者が手動で実  
 290 施する必要がある作業を自動化する。図 3 に示す細い黒色の矢印は、要件が手動で管理者に伝達され、管理  
 291 者による解釈の結果がサービス・レベルとして手動により、アウトオブバンドで返されることを示している。管理者  
 292 は応答として、ソフトウェアで必要とするサービス・レベルに適合するリソースを提供する。青い矢印は、セルフ  
 293 サービス管理インターフェースを表している。このインターフェースには、計算、ストレージ、およびネットワークの  
 294 要件をインバンドで伝達する機能を備えた要素が組み込まれているため、手動でのアウトオブバンドによる要件  
 295 伝達が不要になる。これは、DMTF CIMI などのインターフェースとの相違点として認識されてきた。リソースを必  
 296 要とするさまざまな処理について要件を抽象化し、インターフェースにメタデータとして追加する必要がある。

297 短期的には、多くの場合、図の最上位の領域にプロビジョニング、保護、可用性、パフォーマンス、セキュリティ、  
 298 およびエネルギー消費量として示されるインフラストラクチャー・サービス特性を、粗い粒度で仮想リソースに実装  
 299 し、状況によっては物理リソースに実装する。したがって、リソース自体は細かい粒度の制御で仮想化およびプロ  
 300 ビジョン(個々の作業負荷の粒度でのプロビジョニング)されていても、これらの特性を提供するサービスが  
 301 同じ粒度では仮想化もプロビジョニングもされていないこともある。この状況に対処するには、粗い粒度のサービ  
 302 スと同じ粒度で構成およびプロビジョニングしたリソースのプールを最上位の領域に組み込む。リソース・プール  
 303 はさまざまな理由で利用する手法であり、同じように構成した未使用のリソースとプロビジョニング済みリソースの  
 304 両方を収めている。この図では、わかりやすいようにサンプル・プール名を使用しているが、さまざまな構成の  
 305 プールが多数存在する場合もある。管理者が要件を手動で解釈する場合は、このようにしてそれらの要件に最適  
 306 なリソース構成を備えたプールを選択できる。同様な機能を備えた自動化ソフトウェアが存在し、セルフサービス・  
 307 インターフェースを介して要件を受け取る場合は、そのソフトウェアで要件を解釈し、アルゴリズムを使用して適切  
 308 なプールを選択できる。このリソース・プール技法は一時的な手法であり、インフラストラクチャー・サービスがより  
 309 細かい粒度で動作できるようになった時点で撤廃する必要がある。

310 SDDC は、個別に仮想化したコンポーネントからデータ・センター全体に仮想化の範囲を拡張することにより、仮  
 311 想化技術に基づいて構築するものであり、統合した制御および管理ソリューションを目的とするものである。

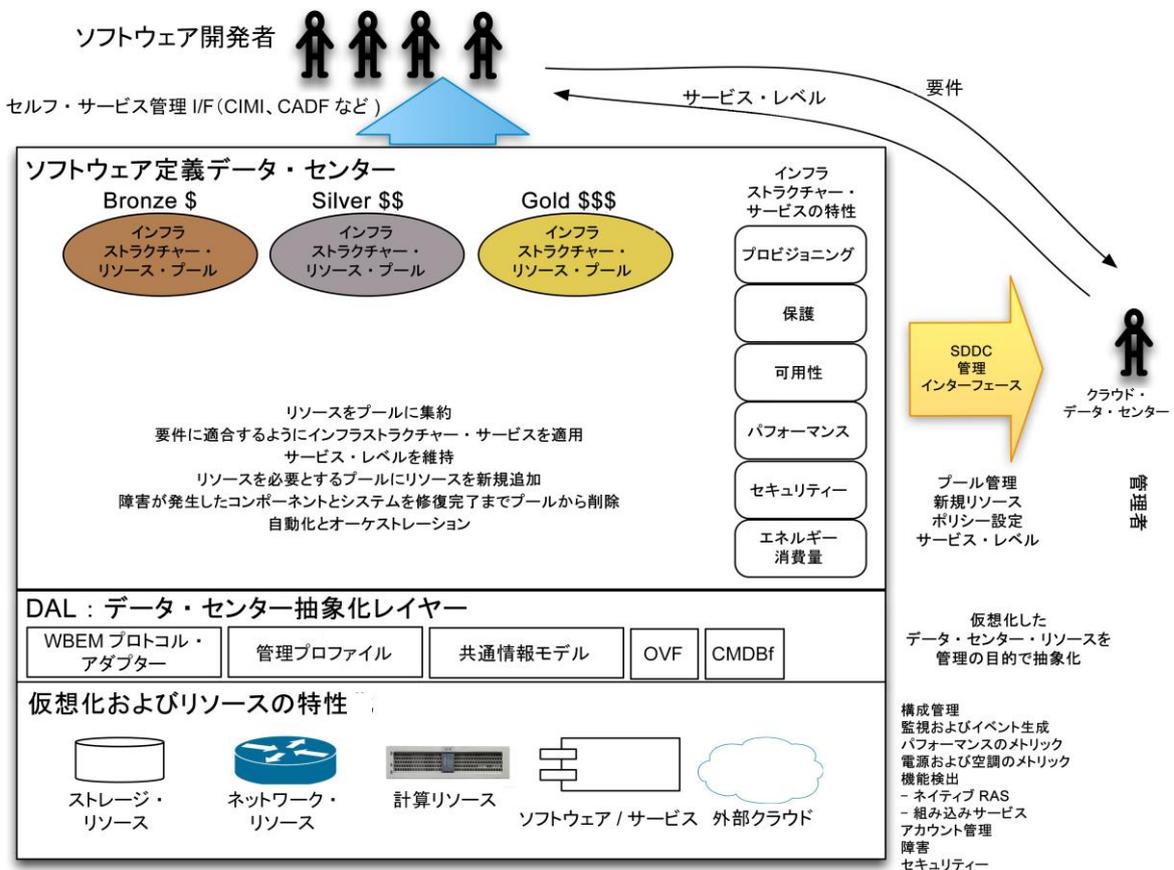


図 3 - SDDC のアーキテクチャー

312  
313

314 図 3 に、SDCC のすべての要素を示す。この SDCC アーキテクチャーには、ソフトウェアに基づくサービスを備  
315 えたデータ・センター・リソースが定義されている。DAL レイヤーは、計算、ネットワーク、およびストレージの各リ  
316 ソースの抽象化を提供する。これらのリソースを、作業負荷の要件に従って仮想化および構成する。

317 DAL は基盤となるリソースを統一性と一貫性の下で抽象化したものであり、SDCC 管理自動化ソフトウェアで使  
318 用できる標準化インターフェースと共通モデルを提供する。

#### 319 4.1 サーバーの仮想化

320 サーバーの仮想化により、基盤となる物理ハードウェアの制限から CPU とメモリーを解放できる。標準のインフ  
321 ラストラクチャー技術としてのサーバーの仮想化は SDCC の基礎である。サーバーの仮想化と同じ原則をすべ  
322 てのインフラストラクチャー・サービスに拡張したものが SDCC である。

#### 323 4.2 ソフトウェア・デファインド・ネットワーク

324 ソフトウェア・デファインド・ネットワーク(SDN)では、これまではスイッチが担っていたネットワーク制御プレーンの  
325 役割がサーバー上で実行するソフトウェアに移行する。この移行により、プログラムで処理できる可能性、効率お  
326 よび拡張性が向上する。SDN は、ソフトウェア・デファインド技術の中でも現時点で開発と理解が最も進んでいる  
327 ものである。したがって、本書では、このソフトウェア・デファインド・コンポーネントについての詳しい説明は省略す  
328 る。

#### 329 4.3 ソフトウェア・デファインド・ストレージ

330 ソフトウェア・デファインド・ストレージ(SDS)は最近になって現れた製品エコシステムであることから、ここでは詳  
331 しい説明が必要になる。このソフトウェアでは、すべての物理リソースと仮想リソースを視覚化し、使用量やニー  
332 ズに基づいてプログラム可能化とプロビジョニングの自動化を実現する必要がある。SDS では、制御プレーンを  
333 データ・プレーンから切り離し、異種混合のストレージを動的に活用して処理要求の変化に対応する。SDS を使  
334 用することで、ストレージ・サービス・カタログの公開が可能になり、オンデマンドでリソースをプロビジョニングし、  
335 ポリシーに従って利用できるようになる。

336 多くの面で、SDS は単なるパッケージ化や IT ユーザのデータ・センターの認識および設計法を超えたものである。  
337 ストレージは、10 年以上にわたって多くの部分でソフトウェア・デファインドになってきている。特定のストレージに  
338 最適化された環境内で、ストレージ機能の大部分がソフトウェア・コンポーネントとして設計され、提供されている。

339 SDS に関する Storage Networking Industry Association (SNIA) の定義は、ベンダ固有のプラットフォームと異  
340 種混合のプラットフォームの両方を考慮している。この SNIA の定義を満足するには、プラットフォームの仮想イ  
341 ンスタンスのプロビジョニングおよび管理のためのセルフサービス・インターフェースをそのプラットフォーム自身  
342 が提供する必要がある。

##### 343 4.3.1 ソフトウェア・デファインド・ストレージに必要な機能

344 現在の多くのストレージ製品はすでに抽象化および仮想化されているので、ここではそのようなストレージ製品が  
345 ソフトウェア・デファインド・ストレージとなるうえで必要な機能について検討する。

346 ソフトウェア・デファインド・ストレージは以下の要件を備えている必要がある。

- 347 • **自動化:** ストレージ・インフラストラクチャーの保守コスト低減に効果的な簡素化した管理
- 348 • **標準インターフェース:** ストレージ・デバイスとストレージ・サービスの管理、プロビジョニング、および保  
349 守を目的とした各種 API
- 350 • **仮想化データ・パス:** アプリケーションからの書き込み先として機能するブロック・インターフェース、ファ  
351 イル・インターフェースおよびオブジェクト・インターフェース
- 352 • **スケーラビリティ:** 可用性やパフォーマンスを損なわずに、ストレージ・インフラストラクチャーの規模を  
353 シームレスに変更できる機能

354 理想には、SDS 製品とは、ストレージ・インフラストラクチャーによるデータの取り扱いを、アプリケーションとデー  
 355 タ作成者が管理できるようにするものである。この管理は、ストレージ管理者による介入操作や明示的なプロビ  
 356 ジョニング操作を必要とせず、自動的なサービス・レベル管理を伴うものとする必要がある。さらに、データ・サー  
 357 ビスを動的に配備できること、サービス・レベルを維持し、性能を要件に合致させるようポリシーを使用することが  
 358 望ましい。この際、メタデータは以下の目的で使用される。

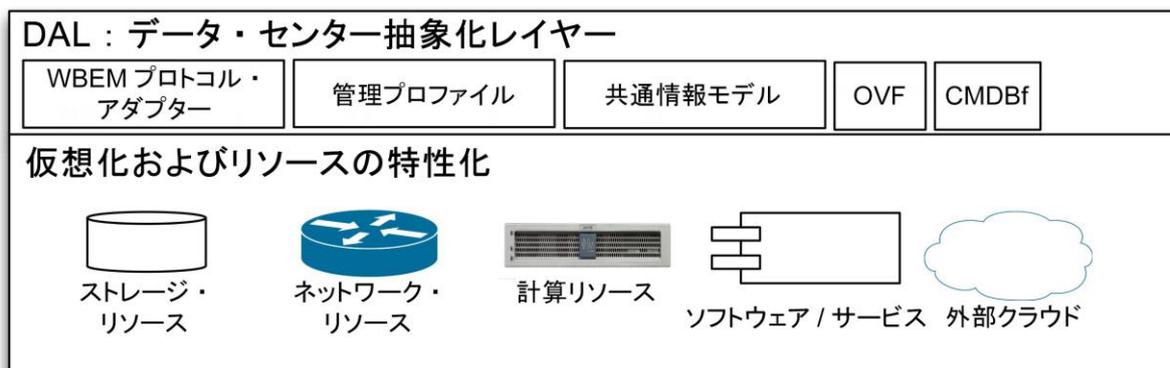
- 359       • 要件の記述
- 360       • データ・サービスの制御
- 361       • サービス・レベル機能の記述

#### 362 4.4 データ・センター抽象化レイヤー

363 データ・センター抽象化レイヤー(DAL)は、データ・センターにある仮想リソースと物理リソースを統一性と一貫性  
 364 の下で抽象化したものである。DAL では、ハードウェア抽象化レイヤー(HAL)の概念をデータ・センター全体に拡  
 365 張している。

366 HAL が開発される前まで、オペレーティングシステムとアプリケーションは PC アーキテクチャーのハードウェア  
 367 が提供する特定の機能に依存していた。HAL は、標準プロトコルを採用することで、このようなハードウェアの多  
 368 様性をオペレーティングシステムとアプリケーションから切り離すことができる抽象化インターフェースを実現した。

369 同様に、DAL はデータ・センターでの計算、ネットワーク、ストレージ、およびソフトウェアの各リソースの多様性を  
 370 抽象化し、それらを SDDC の中で標準化したリソースとして表す。



371

372 図 4 - データ・センター抽象化レイヤー

373 DAL によって実現できる点は以下のとおりである。

- 374       • SDDC の管理レイヤーを使用して、一貫性のある方法でリソースを管理する。
- 375       • 管理レイヤーやアプリケーション・レイヤーの変更を必要とせずに、新しいリソースを導入する。
- 376       SDDC により、リソースの効率と使用率が向上する。

#### 377 4.5 信頼の境界とマルチテナント分離の要件

378 図 3(SDDC のアーキテクチャー)に示すように、一般的な SDDC 実装では、仮想化された計算、ネットワーク、  
 379 ストレージなどのリソースが、同じ組み合わせの物理デバイスでホストされている複数のテナントで共有される。

380  
 381 したがって、これらのテナントの間に明示的な信頼の境界を設定することで、競合することが多いテナント間の適  
 382 切な分離を維持することが必須となる。適切な分離がなければ、ポリシー、セキュリティ、および自動化に関連  
 383 する情報が損なわれ、その結果、適切に機能しているテナントの収益が失われる可能性がある。  
 384

385 アプリケーション、サービス、および管理の観点から、ニーズが複数の物理的な場所にある複数の物理デバイス  
386 にまたがる場合でも、テナント固有のリソースおよびサービス品質(回復性)をはじめとするそれらの構成をサ  
387 ポートする必要がある。これは、テナント固有の組み込みの許可および認証を使用することで達成する必要があ  
388 る。

389  
390 信頼の境界は、物理的、論理的、アドレス空間、ドメイン、およびトポロジーの各セグメンテーションの境界、ピアと  
391 ルーティングのプロファイルなどを使用して構築できる。

392  
393 また、スプーフィングなどの攻撃を防止するために、テナントの識別情報、認証情報、サービスとリソースの利用  
394 契約などを定期的に監視してログに記録し、それらを頻繁に検証および更新できるようにしておく必要もある。

## 395 5 標準に関連する活動

396 DMTF は SDDC のモデルの開発に注力している現時点で唯一の SDO であるが、それに関連する作業に取り  
397 組んでいる組織は DMTF 以外にも数多く存在する。他の SDO の作業の重点は主に SDN および SDS に置か  
398 れているが、新たな標準とそれらが SDDC にどのように関連しているのかを注視することは重要である。

### 399 5.1 DMTF

400 DMTF 標準は、高い完成度の各種管理機能をまとめて提供する明確なインターフェースを通じ、IT 環境の効果  
401 的な管理を可能にする。複数ベンダーの IT インフラストラクチャーどうしの相互運用性、およびクラウド・コン  
402 ピューティング、仮想化、デスクトップ、ネットワーク、サーバー、ストレージなどのシステムとネットワークの管理を  
403 実現するうえで DMTF の標準インターフェースが不可欠である。

404 新たな SDDC パラダイムを実現すると考えられる主な DMTF 標準と策定中の先行標準(イニシアチブ)のいくつ  
405 かについて以下で説明する。

#### 406 407 オープン SDDC インキュベーター

408 DMTF は、SDDC 市場向け初期管理モデルの開発に重点を置いている SDO としては現時点で唯一の存在で  
409 ある。最近、DMTF は、SDDC に向けて DMTF で今後進めるすべての作業を管轄するという趣旨の下で  
410 「SDDC インキュベーター」を設置した。

#### 411 クラウド管理イニシアチブ

412 DMTF のクラウド管理イニシアチブは、クラウド・サービス提供者と、その顧客および開発者との間で相互運用可  
413 能なクラウド・インフラストラクチャー管理を促進することに重点を置いている。このイニシアチブの作業部会は、こ  
414 の相互運用性を実現する目的でオープン標準を開発している。

#### 415 ネットワーク管理イニシアチブ

416 DMTF のネットワーク管理イニシアチブ(NETMAN)は、物理ネットワーク、仮想ネットワーク、アプリケーション中  
417 心型ネットワーク、およびソフトウェア・デファインド・ネットワークの管理を目的とした各種標準を統合している。  
418 NETMAN イニシアチブの目的は、従来のデータ・センター、クラウド・インフラストラクチャー、NFV 環境、および  
419 SDDC エコシステムにわたってネットワーク管理を横断的に統合することである。

#### 420 仮想化管理イニシアチブ

421 DMTF の仮想化管理(VMAN)イニシアチブでは、異種混合仮想環境の管理ライフサイクルを扱う仕様とプロファ  
422 イルを取り上げている。

### 423 5.1.1 クラウド・インフラストラクチャー管理インターフェース (CIMI)

424 CIMI は、インフラストラクチャー・クラウド向けの上位レベルでセルフサービス型のインターフェースである。CIMI  
425 を使用することで、クラウド・システムの管理を大幅に簡素化して、クラウド使用量の動的なプロビジョニング、構  
426 成、および管理が可能になる。この仕様では、JSON と XML を使用して、クラウド環境間の相互作用を標準化し  
427 ている。これにより、相互運用可能なクラウド・インフラストラクチャー管理を実現できる。

428 CIMI は、[国際標準化機構 \(ISO\)](#) および [国際電気標準会議 \(IEC\)](#) の第 1 合同技術委員会 (JTC 1: Joint  
429 Technical Committee 1) により、2015 年 3 月に国際規格として採用された。

430 現在策定中である CIMI 仕様の第 2 版では、機能強化したネットワーク・モデルおよびマルチクラウド・シナリオと  
431 クラウド間シナリオのモデル化により旧バージョンを拡張している。

### 432 5.1.2 オープン仮想化フォーマット (OVF)

433 [OVF](#) 仕様では、複数の異種混合仮想化プラットフォームにわたって配備する仮想マシンとアプリケーションをパッ  
434 ケージ化および記述する標準フォーマットを規定している。OVF は、2010 年 8 月に、[米国規格協会 \(ANSI:](#)  
435 [American National Standards Institute\)](#) によって採択された。また、[国際標準化機構 \(ISO\)](#) と [国際電気標準会](#)  
436 [議 \(IEC\)](#) の第 1 合同技術委員会 (JTC 1: Joint Technical Committee 1) により、2011 年 8 月に国際規格として  
437 採択された。2013 年 1 月に、DMTF は、この標準の第 2 版である OVF 2.0 を発表した。OVF 2.0 は、新たな動  
438 きとなっているクラウドのユースケースを対象としており、ネットワーク構成サポートの向上、安全な配信のための  
439 パッケージ暗号化機能など、OVF 1.0 からの重要な拡張を取り入れている。

### 440 5.1.3 Web ベースのエンタープライズ環境の管理 (WBEM)

441 Web ベースのエンタープライズ環境の管理 (WBEM) は、リソースを検出、アクセス、および操作する方法を定義  
442 し、異種の技術とプラットフォームにまたがるデータ交換を促進する一連の仕様である。

443 [WBEM](#) は、共通情報モデル (CIM) を実装するシステム管理インフラストラクチャー・コンポーネント間での相互作  
444 用のためのプロトコルを定義しており、DAL の主要な構成要素である。

### 445 5.1.4 共通情報モデル (CIM)

446 CIM スキーマは [概念的スキーマ](#) であり、IT 環境の管理対象要素を、オブジェクトと関連の共通的な集合として表  
447 現する方法を定義している。CIM は、製品固有の拡張機能をこのような管理対象要素の共通定義として記述で  
448 きるように拡張可能である。CIM では、[UML](#) に基づくモデルを使用して CIM スキーマを定義する。CIM は大半  
449 の DMTF 標準の基盤である。

### 450 5.1.5 構成管理データベース連携 (CMDBf)

451 [CMDBf](#) は、構成管理データベース (CMDB) と他の管理データ・リポジトリ (MDR) との間で情報の円滑な共有  
452 を実現する。CMDBf 標準により、複雑なマルチベンダー・インフラストラクチャーにある情報の連携を図ってその  
453 情報にアクセスできるようになり、複数の CMDB と MDR に格納した相互に関連性のある構成データの管理プロ  
454 セスを簡素化できる。

### 455 5.1.6 サーバー・ハードウェア向けシステム管理アーキテクチャー (SMASH)

456 DMTF の SMASH 標準は、アーキテクチャーの意味体系、業界標準プロトコル、およびプロファイルを規定して  
457 データ・センターの管理を一元化することを目標とした仕様である。SMASH サーバー管理 (SM) コマンド・ライン・  
458 プロトコル (CLP) 仕様により、データ・センターでの異種混在サーバー群の容易で直感的な管理が可能になる。  
459 SMASH は、DMTF の Web Services for Management (WS-Management) 仕様を活用して、サーバー環境向  
460 けに標準ベースの Web サービス管理を提供する。どちらの仕様も、マシンの状態、オペレーティングシステムの  
461 状態、サーバー・システムのトポロジー、アクセス方法のいずれにも依存しないサーバー管理を提供し、サー  
462 ーバー・ハードウェアのローカル管理およびリモート管理を容易にする。SMASH では、SM Managed Element  
463 Addressing Specification (SM 管理対象要素アドレス仕様)、SM CLP-to-CIM Mapping Specification (SM  
464 CLP-CIM マッピング仕様)、SM CLP Discovery Specification (SM CLP ディスカバリー仕様)、SM プロファイル、  
465 SM CLP アーキテクチャー・ホワイト・ペーパーなども扱っている。

### 466 5.1.7 Redfish API

467 今日のデータ・センターのスケラビリティは、水平型のスケールアウト・ソリューションで実現するようになっている。  
468 多くの場合、こうしたソリューションでは多数の低価格サーバーを使用している。スケールアウト・ハードウェア  
469 の使用モデルは、従来のエンタープライズ・プラットフォームの使用モデルとは根本的に異なっており、新しい手  
470 法での管理が必要である。

471 DMTF の Redfish API は、スケラブルなプラットフォーム・ハードウェアの簡潔で先進的でありながら安全な管  
472 理を求めるエンド・ユーザーの要望に応えるべく設計されたオープンな業界標準仕様およびスキーマである。  
473 Redfish API では、RESTful インターフェースを規定し、JSON および OData を使用して既存のツール・チェーン  
474 の中でソリューションを統合できるようにしている。

## 475 5.2 OASIS

476 OASIS(Organization for the Advancement of Structured Information Standards)は、情報形式標準の製品  
477 非依存な標準を推進することを目的とする非営利の国際コンソーシアムである。

### 478 5.2.1 Cloud Application Management for Platforms(CAMP)

479 OASIS CAMP は、クラウドの実装でアプリケーションをパッケージ化して配備するために使用できる相互運用可  
480 能なプロトコルを推進している。CAMP では、セルフサービスによるプロビジョニング、監視および制御のための  
481 インターフェースを定義している。REST に基づいている CAMP では、共通のツール、プラグイン、ライブラリーお  
482 よびフレームワークで構成するエコシステムの発展が期待でき、ベンダーは CAMP によって高い付加価値を提  
483 供できるようになる。

484 一般的な CAMP のユースケースは以下のとおりである。

- 485 • オンプレミスからクラウド(プライベートまたはパブリック)へのアプリケーションの移動
- 486 • 複数ベンダーのクラウド・プラットフォームにわたるアプリケーションの再配備

### 487 5.2.2 Topology and Orchestration Specification for Cloud Applications(TOSCA)

488 TOSCA TC により、ソフトウェアとハードウェアで複雑に構成したインフラストラクチャーで実行するクラウド・ア  
489 プリケーションと IT サービスの可搬性が大幅に向上する。IT アプリケーションと IT サービスについて TOSCA で規  
490 定している抽象化レベルにより、クラウド・コンピューティングの継続的な発展に不可欠なサポートについても知る  
491 ことができる。たとえば、TOSCA を使用することで、ソフトウェア・デファインド・データ・センター(SDDC)やソフト  
492 ウェア・デファインド・ネットワーク(SDN)などのソフトウェア・デファインド環境(SDE)に不可欠なアプリケーション  
493 とサービスのライフサイクル管理(配備、スケールリング、パッチ適用など)をサポートできるようになる。

494 この目標の実現を容易にするうえで、TOSCA では、アプリケーションとインフラストラクチャーのクラウド・サービ  
495 ス、サービス構成要素間の関連性、およびそれらサービスの運用動作(配備、パッチ適用、シャットダウンなど)を、  
496 サービス作成元に依存せず、特定のクラウド提供者やホスティング技術にも依存せずに、相互に適用できる形態  
497 で規定している。TOSCA では、このような上位レベルの運用動作をクラウド・インフラストラクチャー管理に関連  
498 付けできるようにしている。

499 TOSCA のモデルは、アプリケーションとインフラストラクチャーの専門知識の集合を有機的に統合したものであり、  
500 IaaS や PaaS といった形式のプラットフォーム機能に依存しない形態でアプリケーション要件を記述している。こ  
501 のことから、クラウド・サービス提供者が競争力を発揮して差別化を実現し、ソフトウェア・デファインド環境でア  
502 プリケーションに付加価値を生み出すことができるエコシステムが TOSCA によって実現する。

503 これらの機能により、きわめて高水準なクラウド・サービス/ソリューションの可搬性実現と、ライフサイクル全体を  
504 通じてロックインのないアプリケーションの継続的な提供(DevOps)が著しく容易になり、具体的には以下の状況  
505 が得られる。

- 506 • 対応可能なクラウドへの移植によるデプロイメント
- 507 • クラウドへの既存アプリケーションの容易な移行

- 508       •   さまざまなクラウド提供者とクラウド・プラットフォーム技術にわたる柔軟なアプリケーションの選択と移  
509       動
- 510       •   複数クラウド提供者による動的なアプリケーション

### 511   **5.3   SNIA**

512   Storage Networking Industry Association(SNIA)の使命は、「各種の標準、技術、および教育サービスの開発  
513   および促進において世界のストレージ業界をリードし、組織での情報管理の向上を図ること」である。

514   この目標に向けて、SNIA は多数の仕様を作成している。それらのうち、特に SDCC に関連するものは以下のと  
515   おりである。

#### 516   **5.3.1   Cloud Data Management Interface(CDMI)**

517   SNIA の Cloud Data Management Interface(以下 CDMI)は、クラウドに保管したデータの相互運用性に対して  
518   拡大するニーズにクラウド・ソリューションのベンダーが対応できるようにする ISO/IEC 標準である。CDMI 標準  
519   は、すべての種類のクラウド(プライベート、パブリック、およびハイブリッド)に適用できる。現時点で、20 を超える  
520   製品が CDMI 仕様を満たしている。

521   CDMI により、エンド・ユーザーはそれぞれのデータの動向を制御でき、データ・アクセス、データ保護、およびクラ  
522   ウド・サービス間でのデータ移行を合理化できる。

#### 523   **CDMI のメタデータ**

524   CDMI ではさまざまな種類のメタデータを使用しており、これには HTTP メタデータ、データ・システム・メタデータ、  
525   ユーザー・メタデータ、ストレージ・システム・メタデータなどがある。CDMI では、エンタープライズ・アプリケーシ  
526   ョンの要件およびそのアプリケーションで管理するデータの要件に対応するために、このようなメタデータを使用し  
527   て標準インターフェースによる簡素な操作性を実現している。CDMI は、eXtensible Access Method(以下 XAM)  
528   をはじめとする各データ要素のメタデータに関する従来の SNIA 標準の有効性を強化している。特に、XAM が定  
529   義するメタデータは、コンプライアンスと電子証拠開示に有効なデータ保存サービスの普及に有効である。

530   CDMI によるメタデータの使用は、個別のデータ要素に留まらず、データのコンテナにも同様に適用できる。その  
531   ため、コンテナに置いたデータでは、そのデータが配置されていたコンテナのデータ・システム・メタデータを実質  
532   的に継承する。既存のコンテナの中に新しいコンテナを作成すると、新しいコンテナは親コンテナのメタデータの  
533   設定を上記同様に継承する。当然のことながら、このデータ・システム・メタデータは、コンテナ・レベルまたは個別  
534   のデータ要素レベルで必要に応じて別のデータに置き換えることができる。

535   メタデータをデータだけでなくコンテナの管理にまで広げることで、ストレージの構成要素を管理するためのパラ  
536   ダイムの数を削減できるので、大幅なコスト節減につながる。クラウド・ストレージ・インターフェース標準でメタデー  
537   タをサポートすること、およびデータの要件を満たすためにストレージとデータ・システムのメタデータをどのように解  
538   釈するかを規定することにより、エンタープライズ・アプリケーションとそのデータの要件に対応すると同時に、クラ  
539   ウド・ストレージ・パラダイムで必要とされる単純性を維持できる。

#### 540   **5.3.2   Storage Management Initiative**

541   SNIA の Storage Management Initiative(SMI)は、国際標準である Storage Management Initiative  
542   Specification(SMI-S)を技術作業部会が共同開発する際の指針となる業界ニーズを収集し、優先付けする。  
543   SMI-S は、500 を超える製品に実装されている。

#### 544   **SMI-S 技術仕様**

545   SMI-S は、ストレージ管理機能を、IT 環境の日常タスクを処理する一連の共通ツールへと標準化および合理化  
546   する。SMI-S は、当初はストレージ・デバイスの属性とプロパティを特定するための基盤を提供していたが、現在  
547   では検出、セキュリティ、仮想化、パフォーマンス、障害報告などのサービスも扱っている。

548   SMI-S では、異種混合ストレージ・エリア・ネットワーク(SAN)の相互運用可能な管理の手法を定義する。また、  
549   SMI-S 準拠の CIM サーバーで WBEM クライアントが利用できる情報を規定する。さらに、SAN 内部および

550 SAN を介してデバイスを管理するうえでの特定の要件をサポートするように設計した、オブジェクト指向で XML  
551 とメッセージングを基本とするインターフェースを規定する。SMI-S の最新の公開版は SMI-S V1.6.1 SNIA  
552 Technical Position である。

553 SMI-S は、DMTF の WBEM 仕様および CIM 仕様を採用している。

## 554 5.4 その他の SDO

### 555 5.4.1 ETSI/ISG—Network Function Virtualization (NFV)

556 ETSI/ISG NFV の最初のユースケースでは NFVlaaS(以下 NFV Infrastructure as a Service)について説明し  
557 ているが、これには SDDC との類似性が多数見られる。NFVlaaS は、仮想化した形式の計算、ネットワーキング  
558 およびストレージのインフラストラクチャーで構成されている。NFVlaaS は、ネットワークの結合と相互接続を  
559 サービス(NaaS)として要求し、その他の計算とストレージのインフラストラクチャーをサービス(laaS)として要求  
560 することで、ネットワーク管理者に仮想ネットワーク機能(VNF)を提供する。さまざまな管理可能ドメインにある  
561 VNF は、エンドツーエンド・サービスを作成するために相互接続およびクラスター化できる。NFV のユースケース  
562 に関するドキュメントは、次の URL から入手できる。

563 [http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/001/01.01.01\\_60/gs\\_NFV001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf).

### 564 5.4.2 IETF/IRTF

565 仮想データ・センター(VDC)を検討している [IETF](#) および [IRTF](#) の作業部会/研究部会(WG/RG)がいくつか存在  
566 し、VDC に関する標準の原案もいくつか作成されている。VDC の概念および VDC を使用することで提供できる  
567 サービスの概念は、本書で説明している SDDC の概念と酷似している。

568 NVO3(Network Virtualization Overlays/Over-Layer-3)作業部会(WG)は、Data Center Virtual Private  
569 Network(DCVPN)におけるトラフィックの分離、アドレスの独立性、仮想マシン(VM)の移行で相互運用可能なソ  
570 リューションの開発に重点を置いている。

571 DCVPN は、数千台の VM から 10 万台以上の物理サーバー上で稼動する数百万台の VM に及ぶ規模の範囲  
572 で実現できる VPN であると定義されている。DCVPN は、単一の管理可能ドメインの中で数百万のエンドポイン  
573 トと数十万の VPN をサポートする。IETF NVO3 に関する活動の詳細については、  
574 <http://datatracker.ietf.org/wg/nvo3/charter/>を参照。

575 SCIM(System for Cross-domain Identity Management) WG は、複数の管理可能ドメインにわたってユー  
576 ザーID と ID 関連のオブジェクトを作成、読み取り、検索、変更および削除するコア・スキーマとインターフェース  
577 を HTTP と REST に基づいて開発している。

578 SCIM WG が当初に重視した領域は、コア・スキーマ定義、ユーザーの作成、変更および削除を実行する操作群、  
579 スキーマの検出、読み取りと検索、各種の一括操作、inetOrgPerson LDAP オブジェクト・クラス(RFC 2798)と  
580 SCIM スキーマとのマッピングの開発である。IETF SCIM に関する活動の詳細については、  
581 <http://datatracker.ietf.org/wg/scim/charter/>を参照。

582 SDN(ソフトウェア・デファインド・ネットワーキング)研究グループ(RG)は、現在、SDN の定義と分類の作成に重  
583 点を置いている。今後の作業として、モデルのスケラビリティと適用可能性、マルチレイヤーのプログラム可能  
584 性とフィードバック制御システム、ネットワーク記述言語、抽象化、インターフェースとコンパイラー、SDN のセキュ  
585 リティー関連特性などの研究が挙げられる。IRTF SDN に関する活動の詳細については、<https://irtf.org/sdnrg>  
586 を参照。

### 587 5.4.3 Open Networking Foundation(ONF)

588 [ONF](#) は、フロー転送のリモート・プログラミングを可能にするために、OpenFlow™というサウスバウンドインター  
589 フェース(下位側インタフェース:southbound interface、以下 SBI)を開発した。

590 現在の ONF は、ソフトウェア・デファインド・ネットワーキング(SDN)に関連する案件、特に概念、フレームワーク  
591 およびアーキテクチャーに重点を置いている。

592 Forwarding Abstraction WG、Northbound Interface (NBI) WG、Configuration and Management WG、Layer  
593 4-7 Services DG、および Security DG で扱っているネットワークのセグメンテーション、マルチパスおよびマルチ  
594 テナンスのサポート、およびセキュリティー関連の活動は、オープン SDDC およびその相互接続にきわめて有  
595 効であると考えられる。

#### 596 5.4.4 Open DayLight(ODL)

597 [ODL](#) は、抽象化したネットワークの機能とエンティティーの制御とプログラム化の可能性に重点を置いている。そ  
598 の目的は、効率的な自動化運用を目指して、パフォーマンス分析などのネットワーク・インテリジェンスを収集する  
599 ノースバウンドインターフェース(上位側インターフェース:northbound interface、以下 NBI)を開発し、コントロー  
600 ラーを使用してネットワーク全体で適用性のある新しいルールを構成することにある。ODL イニシアチブの詳細  
601 な技術概要については、<http://www.opendaylight.org/project/technical-overview> を参照。

602 ODL は、OpenFlow などのプロトコルを SBI としてサポートしている。また、このソフトウェア・パッケージの Base  
603 (Enterprise) エディション、Virtualization エディション、および Service Provider エディションを公開している  
604 (<http://www.opendaylight.org/software>)。

#### 605 5.4.5 Open Data Center Alliance(ODCA)

606 [ODCA](#) のイニシアチブと活動は、安全なクラウド・フェデレーション、クラウド・インフラストラクチャーの自動化、共  
607 通管理、およびクラウド・サービス配信の透明性を実現するオープンで相互運用可能なソリューションの開発に重  
608 点を置いている。

## 609 6 標準間の相違 - 何が欠落しているのか

610 これまで、ソフトウェア・デファインド・データ・センターの概念、さまざまなユースケース、およびアーキテクチャーを  
611 分析し、各種標準に関する現在の活動を列挙してきた。その結果、いくつかの技術には明確に定義された標準が  
612 現時点でまだ存在しないことがわかる。ここでは、真に標準に基づく SDDC ソリューションを実現するために調査  
613 と策定を必要とする主要な標準を洗い出す。

### 614 6.1 メトリクスの標準

615 現在、インフラストラクチャーのリソース使用量およびそれらのリソース上でホストする関連のアプリケーションと  
616 サービスのリソース使用量を報告および調整できるようにする標準的なメトリクスが存在しないように見える。処  
617 理で必要とするインフラストラクチャーを処理側で自己管理できるようになると、一連の標準的なメトリクスを開発  
618 する必要が生じることは明らかである。リソース要件とリソース使用量の両方を特定するための真に標準といえる  
619 測定単位はまだ確立されていない。

### 620 6.2 アプリケーションおよび作業負荷の管理

621 アプリケーションおよび処理の要件を数値化するには、一層の取り組みが必要である。作業負荷のデプロイメント  
622 要件については、DMTF のオープン仮想フォーマット(OVF)の規定など、一定の取り組みがなされているものの、  
623 処理とアプリケーションが配備された後、自動構成とオートスケーリングを実現するために多大な取り組みが必要  
624 となる。また、新たな動きとなっているコンテナ化されたアプリケーションの要件を標準化し、ソフトウェア・デファイ  
625 ンド・リソースを動的に作成および削除できるようにするための取り組みも別途必要であることがわかっている。

### 626 6.3 ポリシーおよびサービス・レベル

627 このレベルの自動化を推進するには、標準化したポリシー管理の確立、さらには契約上のサービス・レベル合意  
628 (SLA)に基づいて設定するサービス・レベル目標(SLO)を規定する標準の確立に向けた多大な取り組みが必要  
629 である。これまで、IEC/JTC1 SC38 などの組織によってポリシー用語および標準化したサービス・レベル管理で  
630 の取り組みがなされているが、前述の標準的な尺度など、ポリシーベースのサービス・レベル管理を扱う標準の  
631 作成および普及に向けたさらなる取り組みが必要である。

632 **7 結論**

633 SDCC を実現するには、データ・センターにある計算、ネットワーク、ストレージなどのリソースをソフトウェアとして  
 634 表現する。これらのリソースには一定の特性が必要であり、その特性として、マルチテナント性、リソースの迅速  
 635 なプロビジョニング、柔軟性のあるスケーリング、ポリシー主導型のリソース管理、共有インフラストラクチャー、計  
 636 測、セルフサービス、アカウントिंग、監査などがある。最終的には、リソースのカタログ化、リソースの引き渡し  
 637 と返却、再利用および再配置の自動化を目指してプログラム可能なインフラストラクチャーが必然となる。

638 **8 参照資料**

639 S. Karavettil 他『Security Framework for Virtualized Data Center Services』、IETF discussion draft  
 640 (<http://tools.ietf.org/id/draft-karavettil-vdcs-security-framework-05.txt>)、2013 年 6 月

641 Alan G. Yoder 他『SNIA 2015 Dictionary, Storage Networking Industry Association』  
 642 ([http://www.snia.org/sites/default/files/SNIADictionaryV2015-1\\_0.pdf](http://www.snia.org/sites/default/files/SNIADictionaryV2015-1_0.pdf))、2015 年 3 月

643 SNIA Technical Community: Software Defined Storage (<http://www.snia.org/sds>)

644 **仕様**

645 DMTF: DSP0263、Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based  
 646 Protocol, version 1.1.0、2013 年 10 月 25 日  
 647 [http://dmf.org/sites/default/files/standards/documents/DSP0263\\_1.1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP0263_1.1.0.pdf)

648 DMTF: DSP0243、Open Virtualization Format Specification、version 2.1.0、2014 年 1 月 23 日  
 649 [http://dmf.org/sites/default/files/standards/documents/DSP0243\\_2.1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP0243_2.1.0.pdf)

650 SNIA: SNIA Technical Position: Cloud Data Management Interface (CDMI)、v1.1.1、2015 年 3 月 19 日  
 651 [http://www.snia.org/sites/default/files/CDMI\\_Spec\\_v1.1.1.pdf](http://www.snia.org/sites/default/files/CDMI_Spec_v1.1.1.pdf)

652 SNIA: SNIA Technical Position: Storage Management Initiative Specification (SMI-S) v1.6.1 rev 5、2014  
 653 年 12 月 17 日  
 654 <http://www.snia.org/sites/default/files/SMI-Sv1.6.1r5.zip>

655 **9 用語集**

656

表 1 - 用語集

頭字語または語句	定義	説明
AAA	認証、認可、および監査 (Authentication, Authorization, and Auditing)	システム・セキュリティにおける三大関心領域。

頭字語または語句	定義	説明
API	アプリケーション・プログラミング・インターフェース (Application Programming Interface)	サービスを要求するためにアプリケーションが使用するインターフェース。APIという用語は、アプリケーションと運用環境(オペレーティングシステム、ファイル・システム、ボリューム・マネージャー、デバイス・ドライバーなど)を構成するソフトウェア・コンポーネント間のインターフェースを示すために使用することが普通である。 出典： <a href="http://www.snia.org/education/dictionary/a">http://www.snia.org/education/dictionary/a</a>
ブロック・ストレージ		固定サイズのブロックに編成して割り当てたストレージ。
BYOD	個人所有デバイスの持ち込み (Bring Your Own Device)	従業員が個人所有のモバイル機器(ラップトップ、タブレット、およびスマートフォン)を職場に持ち込み、それらの機器を使用して、アクセス制限された会社の情報やアプリケーションにアクセスすることを許可するポリシー。 出典： <a href="http://en.wikipedia.org/wiki/Bring_your_own_device">http://en.wikipedia.org/wiki/Bring_your_own_device</a>
クラウド	クラウド・コンピューティング	ローカルで利用する機能とは対照的に、インターネット・プロトコルを介してアクセスするリモート・サーバーに基づくコンピューティング機能。
ファイバー・チャネル		主に SAN で使用する高速 LAN 技術。
ファイアウォール		2つ以上のネットワーク間のデータ・フローを制御するデバイスであり、多くの場合はソフトウェアで実装する。ファイアウォールは、信頼されているアドレスやポート以外からのネットワーク・トラフィックを拒否することで、ネットワークどうしを一定の程度で分離することが普通である。
IaaS	インフラストラクチャー・アズ・ア・サービス (Infrastructure as a Service)	IT インフラストラクチャー配信モデルの1つであり、ネットワーク・プロトコルを通じてリソースをサービスとして提供する。 IaaS では、リソースをプロビジョニングおよび管理するためのインターフェースも提供することが普通である。
IDS	侵入検知システム (Intrusion Detection System)	リソースへの無許可アクセスを検出するために使用するシステム
HIDS	ホスト型侵入検知システム (Host Intrusion Detection Systems)	特にホスト・システムの保護を目的として設計されている IDS。

頭字語または語句	定義	説明
LAN	ローカル・エリア・ネットワーク(Local Area Network)	対象範囲を限定した小規模なネットワーク。インターネットのような大規模なネットワークにLANを接続することもできる。
ロード・バランシング		リソースに対する要求を使用可能なリソース間に分散するために使用するメカニズム。通常は処理リソースを対象として使用するが、あらゆるリソースに適用することもできる。
メタデータ		メタデータは「データに関するデータ」であり、構造的メタデータと記述的メタデータの2種類が存在する。構造的メタデータは、データのコンテナに関するデータである。記述的メタデータは、アプリケーション・データの内容に関するものである。
NAS	ネットワーク接続ストレージ(Network Attached Storage)	ネットワークに接続し、コンピューター・システムにファイル・アクセス・サービスを提供するストレージ・デバイスを指す用語。 これらのデバイスは、ファイル・サービスを実装する1つのエンジンおよびデータを格納する1つまたは複数のデバイスで構成することが普通である。 出典： <a href="http://www.snia.org/education/dictionary/n#network_attached_storage">http://www.snia.org/education/dictionary/n#network_attached_storage</a>
NFV	ネットワーク機能の仮想化(Network Function Virtualization)	ルータやファイアウォールをはじめとする専用のネットワーク・アプライアンスを、汎用サーバー上で実行するソフトウェア・アプリケーションで置き換える概念。
オブジェクト・ストレージ		自己完結的データとして編成され格納されたストレージ。
PaaS	プラットフォーム・アズ・ア・サービス(Platform as a Service)	基盤となるインフラストラクチャーをカプセル化し、アプリケーションの開発、実行、および管理を、ネットワーク・プロトコルを介して簡素化するデリバリーモデル。
pDC	物理データ・センター(Physical Data Center)	

頭字語または語句	定義	説明
REST	Representational State Transfer	<p>スケーラブルな Web サービスを作成するためのガイドラインおよびベスト・プラクティスで構成したソフトウェア・アーキテクチャーの形態。REST は、維持可能な高性能アーキテクチャーを実現する分散ハイパーメディア・システムのコンポーネントを設計する際に適用する各種の制約を整理してまとめたものである。</p> <p>出典：  <a href="https://en.wikipedia.org/wiki/Representational_state_transfer">https://en.wikipedia.org/wiki/Representational_state_transfer</a></p>
SaaS	ソフトウェア・アズ・ア・サービス (Software as a Service)	ネットワーク・プロトコルを通じてソフトウェア・アプリケーションをサービスとして提供する配信モデル。
SAN	ストレージ・エリア・ネットワーク (Storage Area Network)	<p>コンピューター・システムとストレージ・デバイス間およびストレージ・デバイスどうしでのデータ転送を主要な目的とするネットワーク。</p> <p>出典：  <a href="http://www.snia.org/education/dictionary/s#storage_area_network">http://www.snia.org/education/dictionary/s#storage_area_network</a></p>
SDDC	ソフトウェア・デファインド・データ・センター (Software Defined Data Center)	本書を参照。
SDN	ソフトウェア・デファインド・ネットワーク (Software Defined Network)	<p>ネットワークコントロールプレーンをフォワーディングプレーンから物理的に分離し、制御プレーンで複数のデバイスを制御する技術。</p> <p>出典：  <a href="https://www.opennetworking.org/sdn-resources/sdn-definition">https://www.opennetworking.org/sdn-resources/sdn-definition</a></p>
SDO	標準策定団体 (Standards Development Organization)	
SDS	ソフトウェア・デファインド・ストレージ (Software Defined Storage)	<p>サービス管理インターフェースを備えた仮想化ストレージ。</p> <p>SDS には、サービス管理インターフェースを通じて指定される要件を満たすために適用可能なデータ・サービス特性を備えたストレージのプールが用意されている。</p> <p>出典：  <a href="http://www.snia.org/education/dictionary/s#software_defined_storage">http://www.snia.org/education/dictionary/s#software_defined_storage</a></p>

頭字語または語句	定義	説明
仮想アプライアンス		特定の種類の仮想マシンで動作するために必要な OS 機能で事前構成したソフトウェア・アプリケーション。必要最低限の機能を備えていることが普通である。 多くの場合、IaaS および SaaS によるシステム・アーキテクチャーでサービスを提供するために仮想アプライアンスを使用する。
VLAN	仮想 LAN (Virtual LAN)	仮想化したローカル・エリア・ネットワーク。
WAN	広域ネットワーク (Wide area network)	

657  
658  
659  
660  
661

## ANNEX A

(参考情報)

### 変更履歴

バージョン	日付	コメント
1.0.0	2015-11-23	

662