TTC技術レポート
Technical Report

TR-1076
IoTエリアネットワーク
情報モデルの概説

Overview of Information Model
for IoT Area Network

第1.00版

2019年6月4日制定

一般社団法人
情報通信技術委員会
THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、一般社団法人情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を一般社団法人情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

目　次

1　はじめに 5
2　各種情報モデルのまとめ 5
3.　各情報モデルの概要 6
3.1　oneM2M 6
3.1.1　概要 6
3.1.2　情報モデルの一例 8
3.2　W3C 8
3.2.1　概要 8
3.2.2　詳細 9
3.3　OCF 10
3.3.1　概要 10
3.3.2　詳細 11
3.4　Echonet 13
3.4.1　概要 13
3.4.2　詳細 14
3.5　IPSO 15
3.5.1　概要 15
3.5.2　詳細 16
3.6　ZigBee 18
3.6.1　概要 18
3.6.2　詳細 19
3.7　IIC 21
3.7.1　概要 21
3.7.2　詳細 21
3.8　OPC-UA 26
3.8.1　概要 26
3.8.2　詳細 27
4.　まとめ 32

＜参考＞

１．　国際勧告等との関連

本技術レポートに関する国際勧告は本文中に記載している。

２．　改版の履歴

版数	制定日	改版内容
第1.0版	2019年6月4日	制定

３．参照文章

主に、本文内に記載されたドキュメントを参照したが代表的な文書を下記に示す。

３．１　規準参照文書

[1]　oneM2M TS-0023-V2.0.0, Home Appliances Information Model and Mapping

[2]　TS-M2M-0023v2.0.2　oneM2M 技術仕様書　家電機器の共通デバイス管理モデル

[3]　Home Gateway Initiative Smart Device Template
https://github.com/Homegateway/SmartDeviceTemplate/tree/7c890b69d9764e341ef1768c5a0e7d53a47cff5c

３．２　非規準参照文書

[TTC標準 JJ-300.30]　「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作概要」, 2017, 一般社団法人情報通信技術委員会

[TR-1066]　TTC技術レポートTR-1066 「橋梁モニタリングのための低消費電力無線通信方式のガイドライン」, 2017, 一般社団法人情報通信技術委員会

[oneM2M]　http://www.onem2m.org/

４．技術レポート作成部門

第１．０版　：　IoTエリアネットワーク専門委員会　（SWG3604）

1　はじめに

本概説は、IoTエリアネットワーク情報モデルの概要書であり、デバイス側とサービス側のデータを伝達する仕組みである、情報モデル（データーモデルとも言う）について各種の分野で検討されているモデルについて調査したものである。今後IoTに関する通信制御はますます増大するものと考えられ、対象となる分野は家庭内のみならず、街レベルや電力制御や社会インフラ等への制御として変遷・拡大を続けてゆくものと考えられる。そのため、これらに係る各種フォーラム等の情報モデルを調査し、関連情報・プロトコル・データ情報等を整理し今後の方向性も合わせて検討したものである。

具体的には、情報モデルとはデバイスの機能・情報を表現するためのモデルでデバイス管理モデルとも呼ばれている。図1-1 にoneM2M等で述べられているイメージ図を示す。

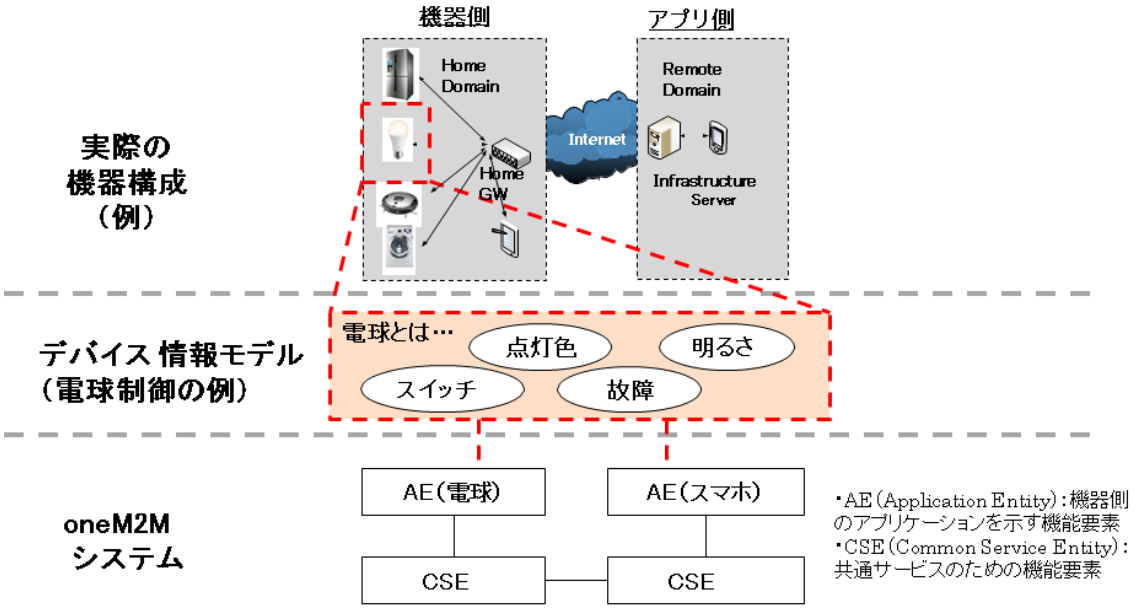


図 1-1　情報モデルイメージ図

2　各種情報モデルのまとめ

情報モデルは各所で検討されており、それぞれ名称や構造が異なっている。具体的な下記検討機関を調査・検討してとりまとめた。

- ◆ OneM2M、W3C、WoT、IIC、OCF(旧 OIC)
- ◆ ECHONET Lite、IPSO、ZigBee、OPC-UA、YANG、TR-181、KNX

これらを、アーキテクチャ、データプロトコル、情報（データ）モデルに分類した比較表を表2-1 に示す。表にあるように大分類としてテレコム関係と一般オープンソース、その他に分類した。調査ターゲットしては表に示すように、赤丸で囲んだテレコム関係と一般オープンソースを対象範囲とし、黒丸で囲んだIETF、IEC、BBFは対象外とした。

次章以降にこれらの詳細について記載する。

表 2-1　各種情報モデルのまとめ

大分類	標準化団体	標準規定範囲		
		アーキテクチャ	データープロトコル	情報(データ)モデル
テレコム	oneM2M	TS-001	TS-004	TS-0023(SDT、ホームアプライアンス)
	SG13	Y.2070	規定範囲外、もしくは検討中	対象範囲
	SG20	Y.4113		
	IIC	IIRA		
	W3C	WoT Architecture	WoT Binding Template	WoT Thing Description
オープンソース	OCF(OIC)	OIC Core	OIC Core	OIC Core Smart Home
	Echonet	規定範囲外	Echonet Lite	オブジェクト詳細規定
	KNX		KNX	Application Description
	IPSO		OMA LWM2M	Smart Object
	ZigBee		ZigBee IP/Pro	ZigBee Cluster Library
その他	IETF		NETCONF(RFC6241)	YANG(RFC6020) トランスポートはNETCONF
	IEC	対象外	Echonet、KNX	CIM(スマグリ関係:IEC TC57、61850他)
	BBF		TR-069	TR-181(トランスポートはTR-069)

3.　各情報モデルの概要

表2-1にある各種情報モデルの概要を、以下に記載する。

3.1　oneM2M

3.1.1　概要

oneM2M は、欧米日中韓の標準化団体（7 団体：ETSI、ATIS、TIA、ARIB、TTC、CCSA、TTA）が協力し，2012 年に結成した M2M に関するグローバル標準仕様策定プロジェクトである。　情報モデルに関しては oneM2M がホームアプライアンスの抽象情報モデル策定に適用した、HGI（Home Gateway Initiative）で規定された記述様式 SDT（Smart Device Template）3.0 を利用した TS-0023 「Home Appliances Information Model and Mapping」がある。 図 3-1 にその概要を示す。

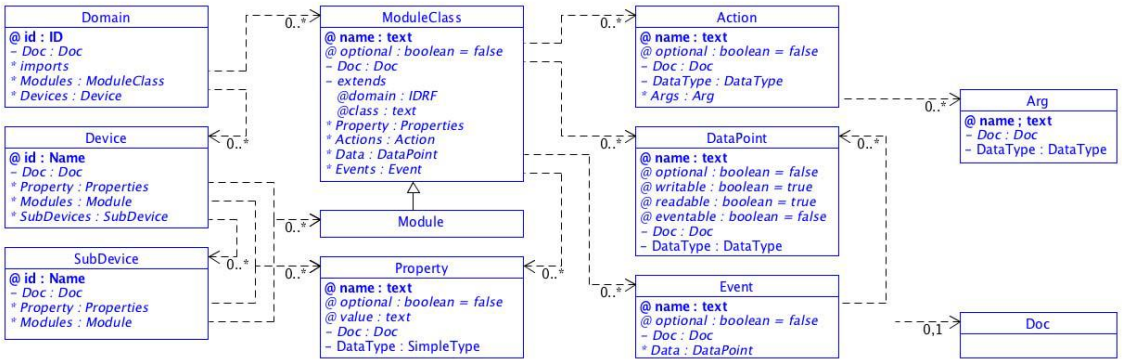


図 3-1　oneM2M の SDT

（出典：oneM2M TS-0023 出典：TS-M2M-0023v2.0.2
「oneM2M 技術仕様書家電機器の共通デバイス管理モデル」Figure 5.2.1-1 より引用
（http://www.ttc.or.jp/jp/document_list/pdf/j/TS/TS-M2M-0023v2.0.2.pdf））

図にあるように SDT の構成要素としては、以下の要素で構成するものとする。

(1) ドメイン（Domain）

- ネームスペースを示し、規定するモジュールやデバイスを一意に参照できる
- テクノロジーやセグメントの異なる SDT テンプレートへのラベル付けが出来る
- これにより、異なる詳細構造を持つ複数のテンプレートが利用できる

(2) デバイス（Device）

- モデル化するセンサ、アクチュエータなどの基本的な実体を示す
- Sub Device を定義して階層化することが出来る
- 0 または複数の Sub Device および Module と Property で構成される

(3) サブデバイス（Sub Device）

- Device に組み込まれた、個別の機能を持った実体を示す
- Sub Device を追加することで Device の機能を変化させる場合などに用いる
- Sub Device は 0 または複数の Module と Property で構成される

(4) モジュールクラス（Module Class）

- 一つのサービスを示す
- 複数の Device 間で再利用ができる
- 0 または複数の Action、DataPoint、Event、Property から構成される

(5) モジュール（Module）

- Module Class から生成するサブクラスである
- Device または Sub Device に特有のサービスを表す
- Device または Sub Device に包含される

(6) アクション（Action）

- Device のオペレーションを表す
- Module に包含される

(7) データポイント（DataPoint）

- Device が持つ機能の 1 つの態様を表す
- ステートレスなオペレーション（読み、書き、または両方）が可能である
- Module に包含される

(8) イベント（Events）

- Device に特定の事象が発生した時にそれを公開する
- Module に包含される

(9) プロパティ（Property）

- Device または Module に任意の追加情報を付加するために用いる
- Device または Module の機能に直接関係しない情報を示す（ベンダー名、シリアル番号等）
- Device または Module に包含される

(10) ドキュメンテーション（Doc)

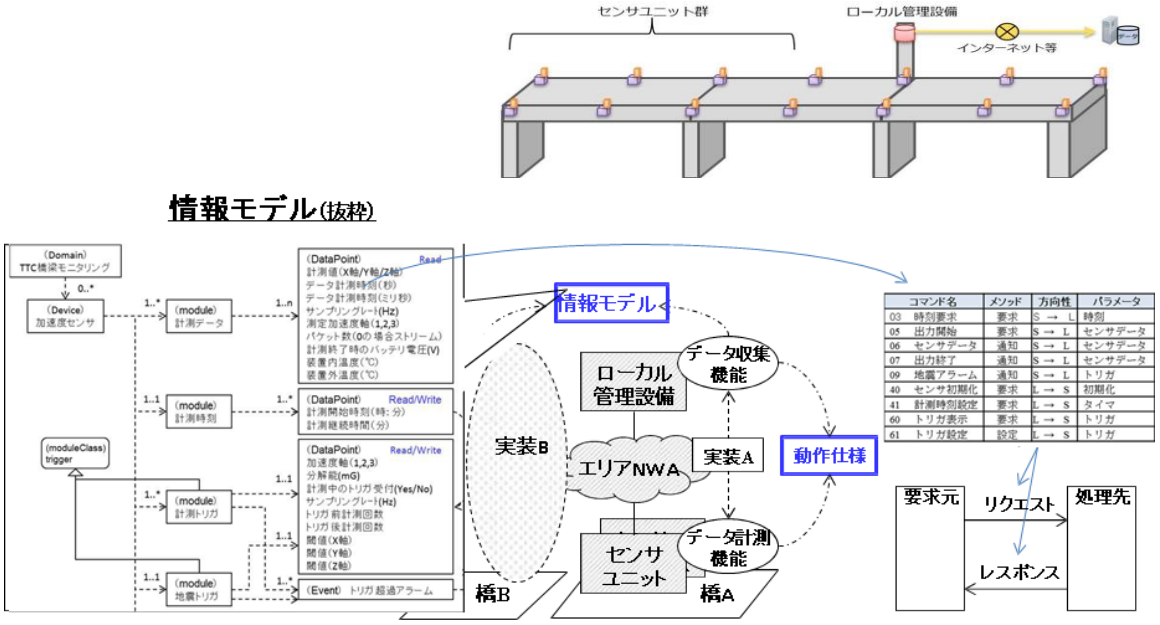
- 補足説明を記載する
- Device や Module の機能には影響しない
- 全ての構成要素に含まれる

3.1.2　情報モデルの一例

OneM2MのSDTを参考にした情報モデルの一例として、TTC技術レポートTR-1066　「橋梁モニタリングのための低消費電力無線通信方式ガイドライン」とTTC標準 JJ-300.30 「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作概要」を、以下に記載する。

加速度センサの情報モデルは、TR-1066 に記載の橋梁モニタリングシステムにおける「センサユニット」をアプリケーションで取り扱う際に必要となる情報の要件、関係性、制約、規則、操作などを SDT に従って記述したものである。

本標準では、情報モデルの対象とするドメインを「橋梁モニタリング」とし、そこで定義するデバイスを「加速度センサ」とし、これに関わるモジュールクラスを規定したもので、図3-2 にその概要を示す。



図出典：TTC標準 JJ-300.30「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作」

その他参考資料:TR-1066「橋梁モニタリングのための低消費電力無線通信方式ガイドライン」

図 3-2　橋梁モニタリング用加速度センサの 情報モデル例

3.2　W3C

3.2.1　概要

W3Cとは「World Wide Web Consortium」の略称で、Web技術の標準化を行う非営利団体の名称である。W3Cは1994年に創設され、Webで使用される技術を標準化し、よりスムーズな開発や品質向上を目標に活動が続けられている。W3Cが勧告しているWWW関連の規格は「Web標準」と呼ばれ、Web制作の現場でもHTMLやCSS、DOMに関する仕様としてサービスの品質を決める大きな役割を果たしている。インターネット通信では互換性が重要視されており、Web標準規格に準拠することはこの「互換性を保証する」ことにつながるとされている。

その中で情報モデルに係るものとしてWoT Architectureがあり、物のデータモデルとBinding Templateを用いて連携するWoT Thing Description（以下TDと略す）と言う情報モデルがある。TDは、Linked Data ボキャブラリとIoTシステムの機能APIとの間のギャップを埋める構造化データで、それは“物のためのHTML”と見ることができる。TDは、Thingの一般的なメタデータと、Thingの相互作用、データモデル、通信、およびセキュリティメカニズムに関するメタデータを提供している。

通常、TDは、WoTが明示的な拡張ポイントを提供するドメイン固有のメタデータを使用している。ただし、ドメイン固有の語彙は、W3Cの標準化活動の範囲外である。図3-3 にW3C WoT Thing Descriptionのデータモデルを示す。

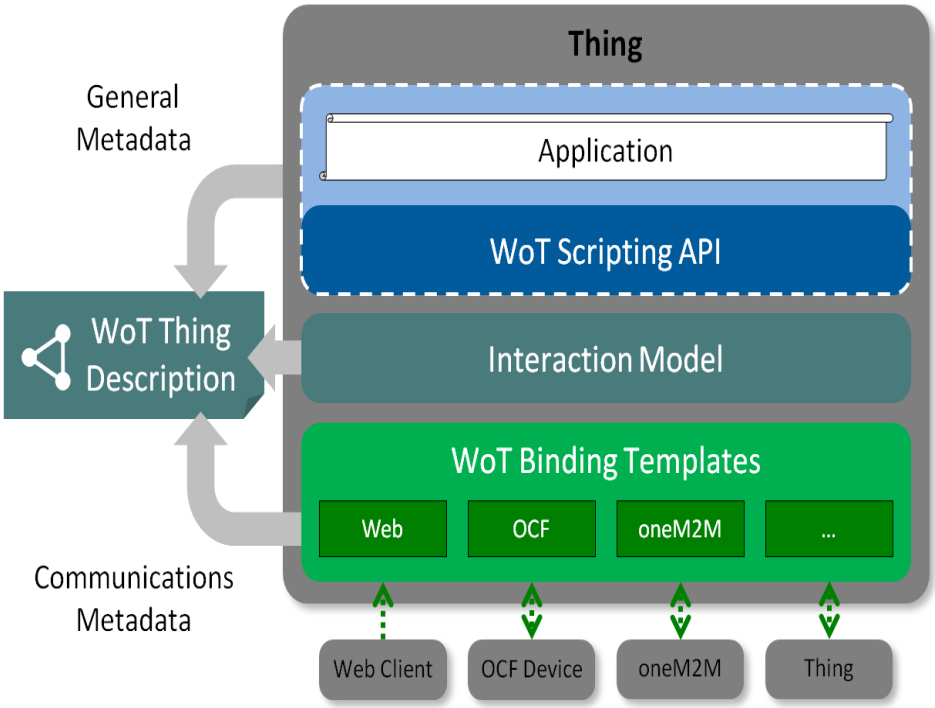


図 3-3　W3C WoT Thing Description のデータモデル

出典：　W3Cサイトより引用：The W3C Web of Things (WoT)、
「Figure 10 Conceptional Architecture of the WoT Building Blocks」より
（http://www.w3.org/TR/wot-architecture/）

さらにTDはユーザデータ、運用データについては、多くのプロトコルでEntityに対するプロパティとその値として管理される。 TDは、Thing自体のセマンティックメタデータ、WoTのプロパティ、アクション、およびイベントを含む narrow-waistの相互作用モデル、データモデルを機械で理解可能にするセマンティックスキーマ、およびWebリンクのフィーチャ間の関係を表現するフィーチャで構成される。

3.2.2　詳細

情報モデルの特徴としては、プロトコル、セキュリティ等の情報を含み、インタラクションモデルとして、property、action、event を定義している。具体的には TD はユーザデータ、運用データについては、多くのプロトコルで Entity に対するプロパティとその値として管理される。 TD は、Thing 自体のセマンティックメタデータ、WoT のプロパティ、アクション、およびイベントを含む narrow-waist の相互作用モデル、データモデルを機械で理解可能にするセマンティックスキーマ、および Web リンクのフィーチャ間の関係を表現するフィーチャで構成される。

図 3-4 にその概要を示す。

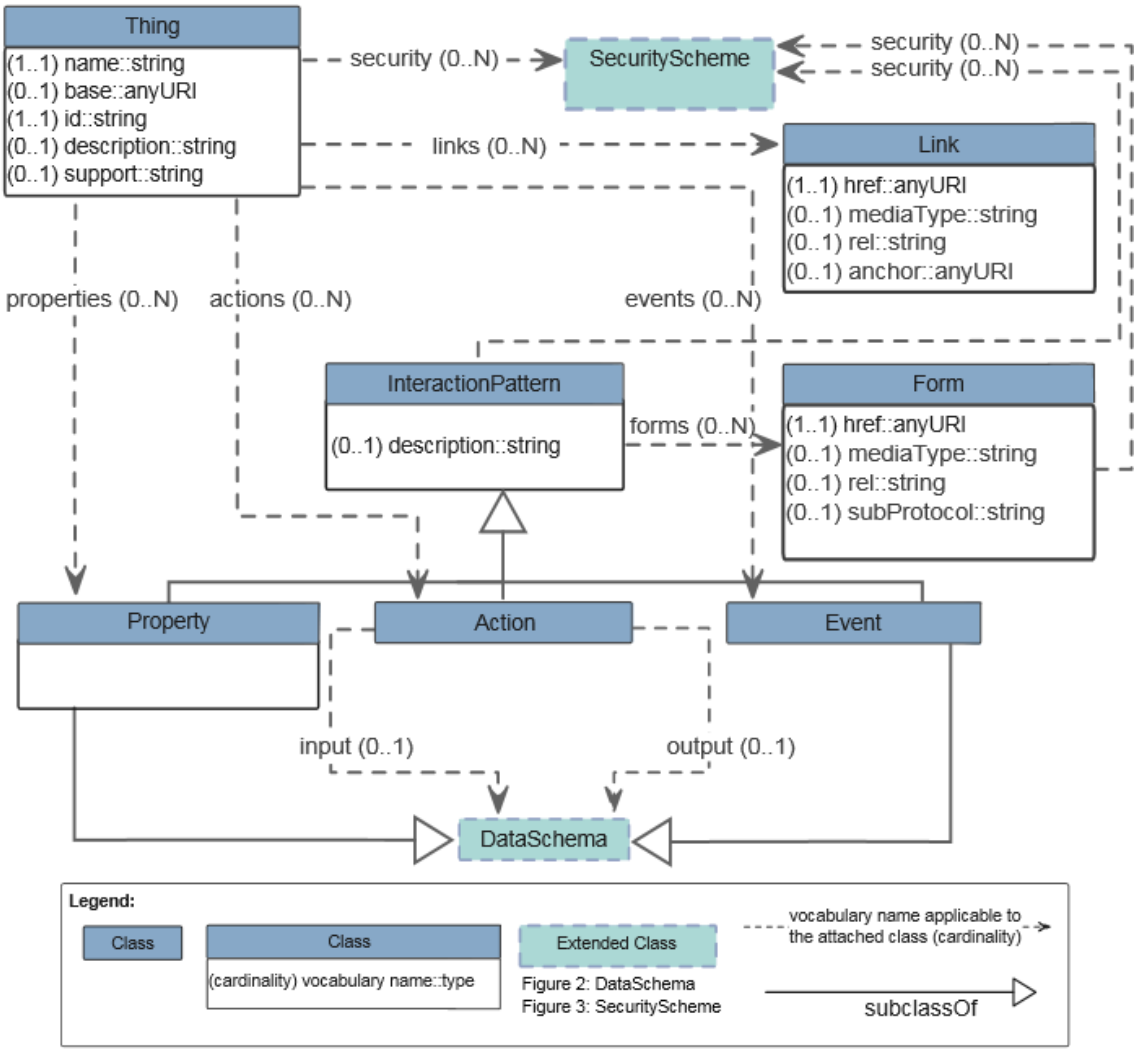


図 3-4　W3C WoT Thing Description のデータモデル

出典：　W3Cサイトより引用、Web of Things (WoT) Thing Description、Figure 1 TD core modelより（http://www.w3.org/TR/2018/WD-wot-thing-description-20181021/#changes）

3.3　OCF

3.3.1　概要

OCF（Open Connectivity Foundation）は2016年2月にできた新しい組織である。OCFはインテルやサムスン、デルなどが中心となって2014年7月に設立したOpen Interconnect Consortium（OIC）と、2013年2月にQualcommが中心となって設立したAllseen Allianceの主力メンバーであるマイクロソフト、クアルコム、エレクトロラックスの3社がボードメンバーとして参加して組織された。OCFの2016年12月のプレスリリースによれば、新たに68社が加入して会員数が300社以上になったとのことである。図3-5 に統合の流れを記載する。

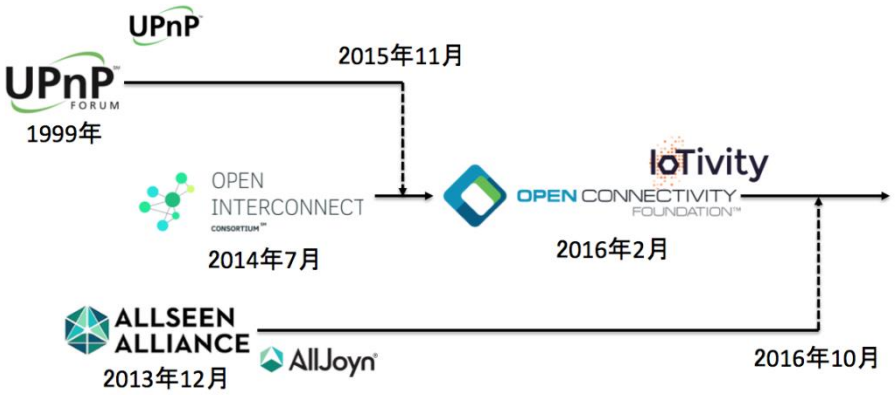


図 3-5　OCF の変遷

出典： Thech Blog（フューチャーアーキテクト開発者ブログ）図1より引用

（https://future-architect.github.io/articles/20161209/,）

3.3.2　詳細

OIC Core Smart Home は Samsung、Intel などが中心となり設立した Open Connectivity Foundation（OCF）のデータモデルで詳細は以下の通りである。図 3-6 に詳細を示す。

- OIC は OS、トランスポート、ベンダーに依存せず全てのモノをつなぐための IoT ソフトウェアフレームワーク
  - ✓ 現在は OCF（Open Connectivity Foundation）で OCF = OIC + UPnP Forum
- データモデルとデータプロトコル（メッセージング）を明確に区別し
  - ✓ データモデルは RESTful リソースモデルで、Create, Read, Update, Delete, Notify（CRUDN）で通信
  - ✓ データプロトコルは、CoAP/HTTP/MQTT 等で CRUDN を実現
  - ✓ リソースモデルは、物理デバイスの抽象表現

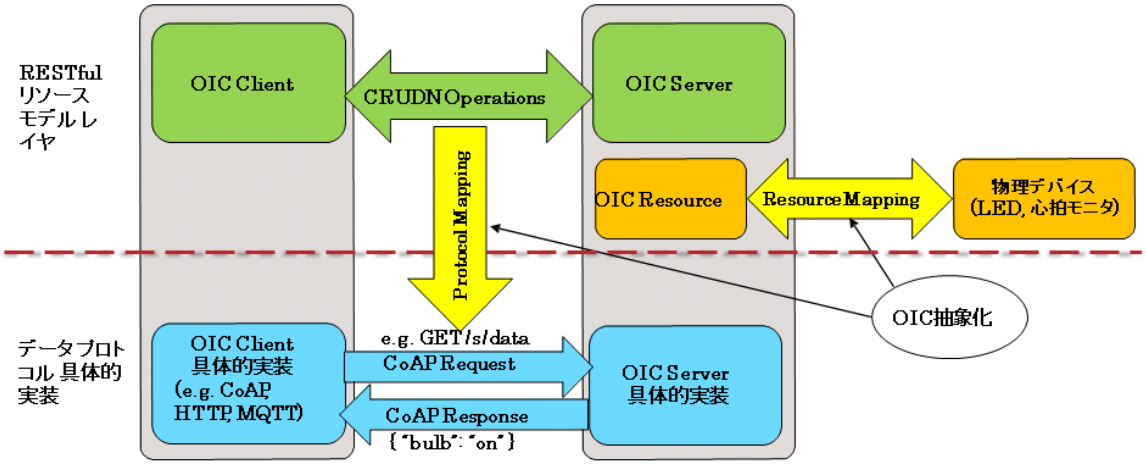


図 3-6　OCF のデータモデル

出典：OCF_Core_Specification_v2.0.0、Figure 1: Architecture - concepts を和訳化

（https://openconnectivity.org/specs/OCF_Core_Specification_v2.0.0.pdf

最新版は V2.0.1）

(1)　プロトコル構成

下図にOCFのプロトコル構成を示す。OCFでは、機器間で通信するための機能ブロックとして図3-7 のようなプロトコル構成を取っている。CBOR:は「The Concise Binary Object Representation」の略でIETF RFC 7049で定義されており、設計目標が、非常に小さいコードサイズ、かなり小さいメッセージサイズ、およびバージョンネゴシエーションを必要としない拡張性の可能性を含むデータフォーマットで、JSONデータモデルに基づく簡潔なバイナリオブジェクト表現である。

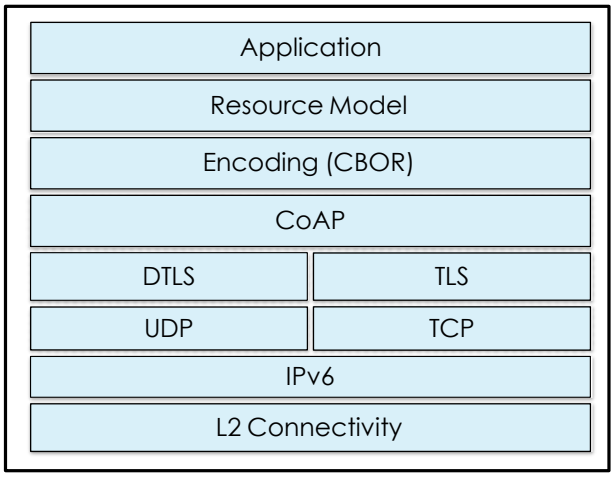


図 3-7　OCF Protocol Stack

出典：OCF Specification Introduction and Overview

https://openconnectivity.org/specs/OCF_2.0_Specification_Overview.pdf

(2)　具体的なデバイス制御

具体的には機器タイプ毎に共通する操作をResourceとして下記の様に定義している。

・一般的な操作について、色々な機器に対して定義を用意

・対応機器はこれを元に動作・応答を実装

・メーカー独自操作等はオプションで可能

・Resourceとして値の型、範囲、Read-only、collectionなどのプロパティを用意

下記に具体的なデバイスタイプの一例を表3-1に示す。

表 3-1　必須リソースを含む、デバイスタイプの一例

Device Name (informative)	Device Type ("rt") (Normative)	Required Resource name	Required Resource Type
Active Speaker	oic.d.speaker	Binary Switch	oic.r.switch.binary
		Audio Controls	oic.r.audio
Air Conditioner	oic.d.airconditioner	Binary Switch	oic.r.switch.binary
		Temperature	oic.r.temperature
Air Purifier	oic.d.airpurifier	Binary Switch	oic.r.switch.binary
Air Quality Monitor	oic.d.airqualitymonitor	Air Quality Collection	oic.r.airqualitycollection

出典：(株)ピクセラ社資料より、

https://www.slideshare.net/PIXELAcorporation/what-is-ocf

さらに詳細なデバイスの制御例として電球の制御について図3-8に記載する。

・サンプル概要：バイナリスイッチ、明るさリソースを備えたスマートライトデバイス

・機種[ドメイン定義]：ライトデバイス（oic.d.light）

・関連資料：必須コアリソース：oic / res、oic / p、oic / d

・必須セキュリティリソース（図示せず）

・機器固有リソース：バイナリスイッチ（oic.r.switch.binary）、

・この例では、他のオプションのリソースを公開している。

Brightnessリソース（oic.r.light.brightness）

Device Title	Device Type	Associated Resource Type	M/O
Light	oic.d.light	oic/res (oic.wk.res)	M
		oic/p (oic.wk.p)	M
		oic/d (oic.d.light)	M
		Binary switch (oic.r.switch.binary)	M
		Brightness (oic.r.light.brightness)	O

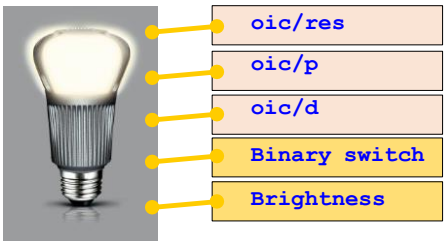


図 3-8　Example: Smart light device

出典：OCF Specification Introduction and Overview

https://openconnectivity.org/specs/OCF_2.0_Specification_Overview.pdf

3.4　Echonet

3.4.1　概要

ECHONET コンソーシアムは、白物家電、住宅設備機器を中心としたホームネットワークシステム実現に向けて 1997 年から活動を行っている団体である。2018 年 10 月現在、幹事会員 7 社、幹事準会員 42 社、一般会員 165 社、一般準会員 36 社から構成されている。

ECHONET と ECHONET Lite があり後者が新しい規格である。ECHONET Lite は、センサ類、白物家電、設備系機器など省リソースの機器を IoT 化し、エネルギーマネジメントやリモートメンテナンスなどのサービスなどを実現するための通信仕様で、 通信仕様や各機器の制御コマンドを共通仕様とすることで、マルチベンダ環境でのシステム構築を実現しており、下記特徴がある。

・各種既存の標準伝送メディアの利用が可能

・家庭・中小ビル・店舗向け等、広範囲な機器に対応

・重点機器のさらなる相互接続性の向上

・クラウド上のサービスとの連携が可能

ECHONET Lite の規格類の構成を図 3-9 に示す。

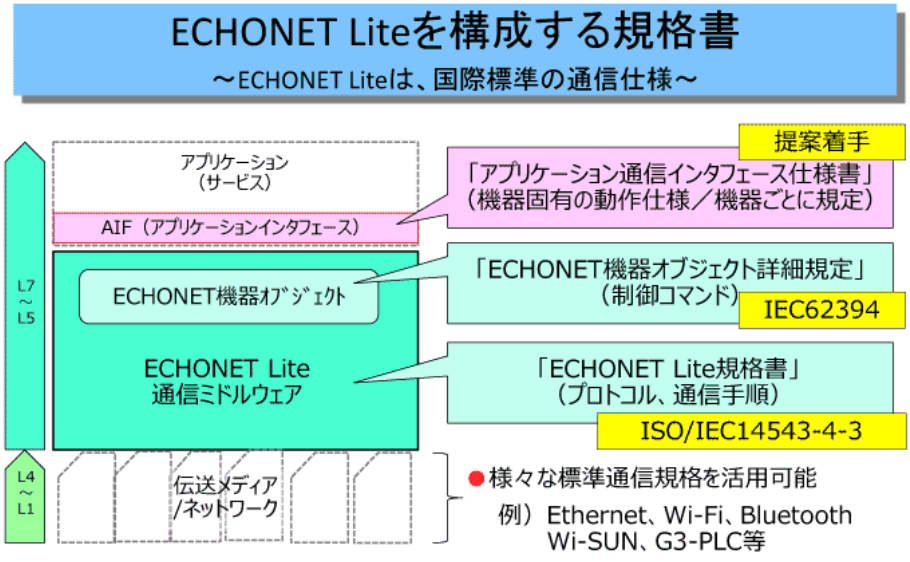


図 3-9　ECHONET Lite の規格類の構成

（出典：ECHONETホームページ,「ECHONET Lite規格の特長と概要」より
（https://echonet.jp/about/features/）

3.4.2　詳細

情報モデルに関連する規定としては、ECHONET オブジェクト詳細規定があり、下記の特徴がある。

- 住宅設備のデータモデルと通信方式
- 2000 年に ECHONET 規格
  - ✓ 家電等のデータモデル作成、Property と Event はあるが Action の考えがない
  - ✓ 通信媒体毎の接続方法を規定
- 2011 年に ECHONET Lite 規格(IP ベース)
  - ✓ IP を基本として、データプロトコルを簡略化、媒体依存がなくなる
  - ✓ データモデルは住宅設備を中心に 80 種類定義

下記、表 3-2、図 3-10 にデータモデル例とデータモデルとデータプロトコルの構成を示す。

表 3-2　データモデル例

プロパティ名称	EPC	プロパティ内容 値域(10 進表記)	データ型	データサイズ	単位	アクセスルール	必須	状変時アナウンス	備考
動作状態	0x80	ON／OFF の状態を示す。 ＯＮ＝0x30，ＯＦＦ＝0x31	unsigned char	1 Byte	－	Set Get	○ ○	○	
節電動作設定	0x8F	機器の節電動作を設定し、状態を取得する。 節電動作中=0x41 通常動作中=0x42	unsigned char	1 Byte	－	Set/ Get	○	○	
運転モード設定	0xB0	自動／冷房／暖房／除湿／送風／その他の運転モードを設定し、設定状態を取得する。 順番に以下のコードが対応。	unsigned char	1 Byte	－	Set/ Get	○	○	

家庭用エアコンのデータモデル(一部)

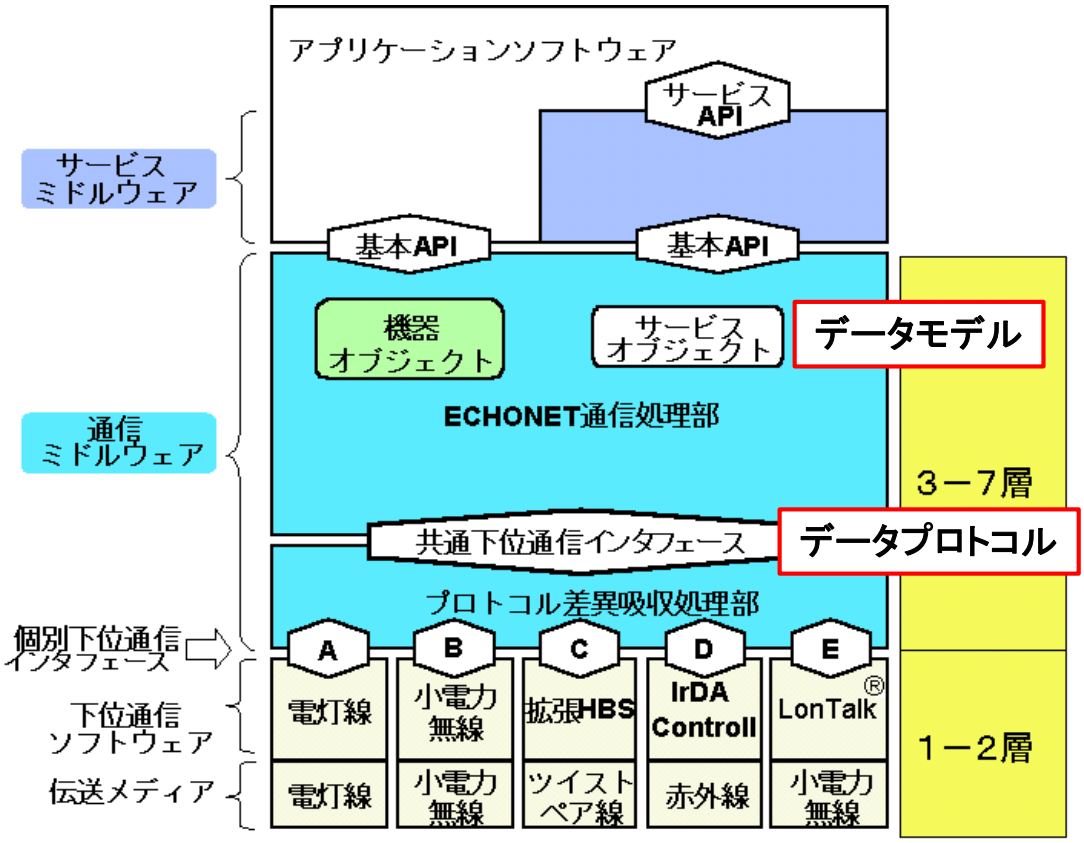


図 3-10　データモデル例

（出典：ECHONETホームページより編集、https://echonet.jp）

上段：APPENDIX ECHONET機器オブジェクト詳細規定 Release K、
３．２．１ 家庭用エアコンクラス規定より抜粋、加筆
（https://echonet.jp/wp/wp-content/uploads/pdf/General/Standard/Release/Release_K_jp/Appendix_Release_K.pdf）

下段：ECHONET（エコーネット）の技術動向、図3より抜粋、加筆
（http://www.sharp.co.jp/corporate/rd/journal-82/pdf/82-08.pdf）

3.5　IPSO

3.5.1　概要

IPSOは2008年9月に設立されたアライアンス（IP for Smart Objects Alliance）で、目的は“Smart Object”をIP（Internet Protocol）ベースで構築。設立はArch Rock、 Atmel、Cimetrics、Cisco、Duke Energy、Dust Networksなど、テクノロジー、通信、エネルギーの各企業25社で行われ、エネルギー、消費者、ヘルスケア、および産業用アプリケーションにおけるIPネットワークデバイスを提唱している。

IPSOスマートオブジェクトは

- スマートオブジェクトアライアンス（IPSO）用 IP によって開発
- IoT デバイスおよびアプリケーション間のセマンティック相互運用性
- LWM2M オブジェクトモデルに基づく、再利用可能なオブジェクト ID とリソース ID
- LWM2M アドレス指定、データタイプ、およびコンテンツフォーマットをサポートできる、さまざまなトランスポートプロトコル（CoAP、HTTP、MQTT）で使用可能
- 単純なオブジェクト用の Basic Starter Pack が 2014 年に公開

- Basic Objects は単純なセンサとアクチュエータから構成

3.5.2　詳細

IPSO Smart Objectsの構成を図3-11に示す。

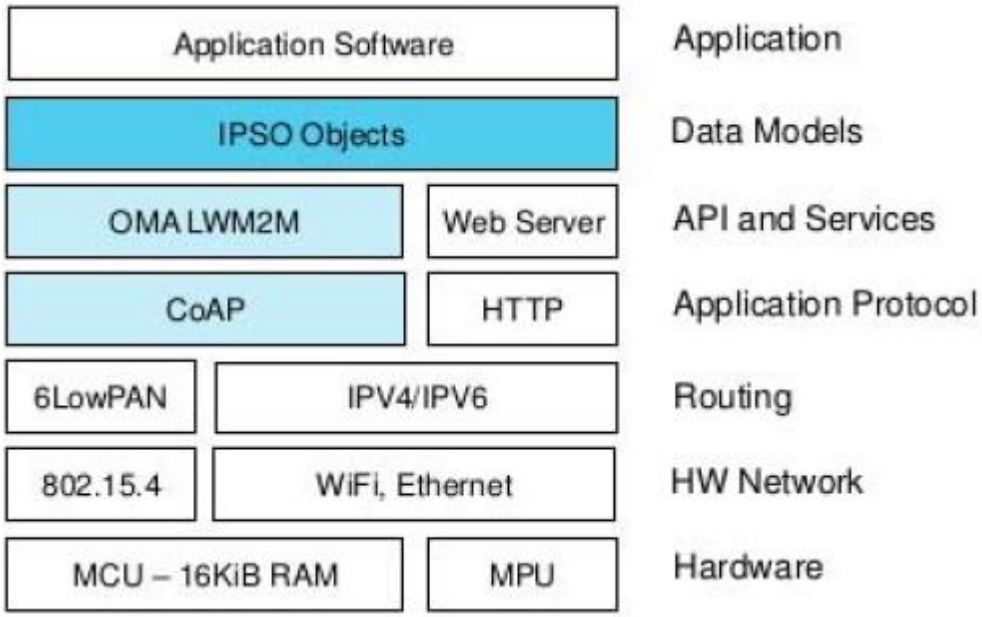


図 3-11　IPSO の７層モデル

出典：　IPSO Smart Object for IoT、P3より

（https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot）

以下のような特徴がある。

◆　›組み込みデバイスへの RESTful Web サービス
- エンタープライズ SOA 統合を容易にする
- アプリケーション開発を容易にする Web プログラミング

◆　HTTP または CoAP が可能
- Observe など、制約のある環境での CoAP（および EXI）
- HTTP-CoAP インターワーキング

◆　リソースビューをサポート
- Web リソース - Web リンク
- セマンティックアノテーションとシンプルなプロファイルが望ましい
- XML - EXI、JSON

◆　「オープン」とは発見を示す
- 出版と検索

◆　レガシーデバイスのラッピングが可能

図3-12に、温度管理をモデルとした制御概念図を示す。

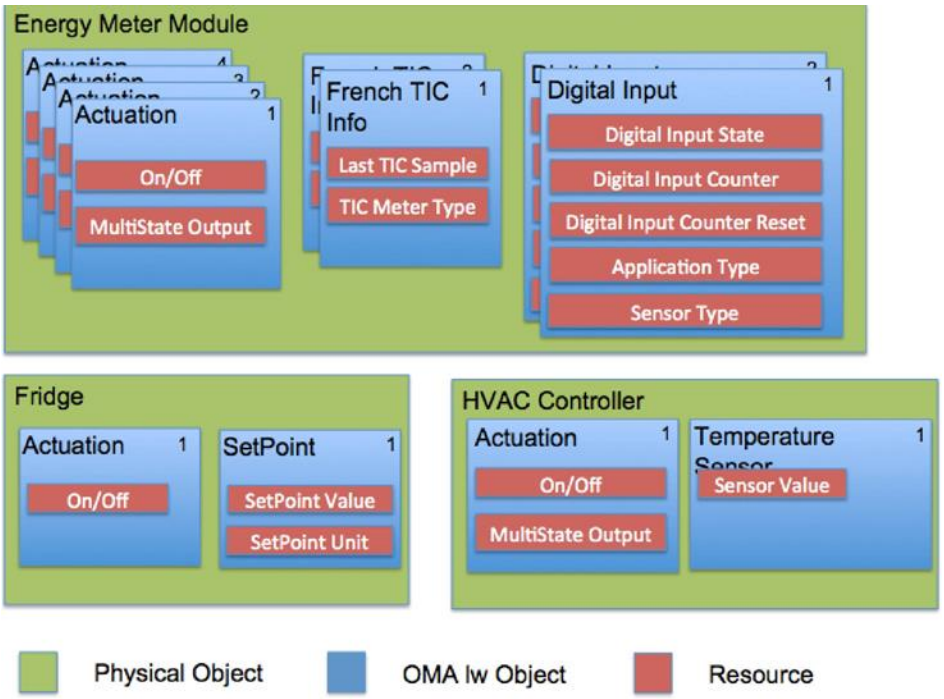


図 3-12　IPSO スマートオブジェクト概念図

出典：　IPSO Smart Object for IoT、P11より

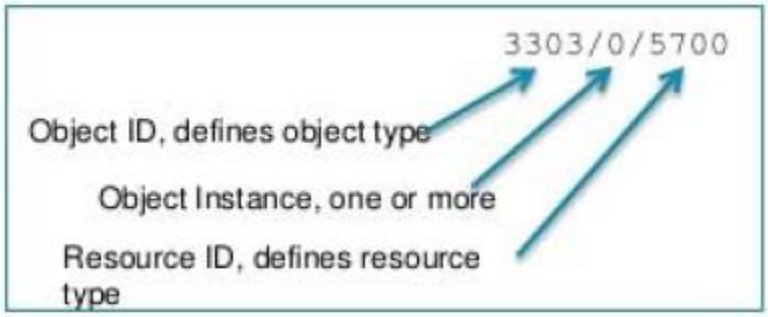
（https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot）

また、具体的な制御のデータモデルとしたIPSO BAROMETER OBJECTの制御例を図3-13に示す。

◆ URI テンプレートを使った REST API
- オブジェクト
- オブジェクトインスタンス
- リソース
- （リソースインスタンス）

◆ 再利用可能なリソースとオブジェクト ID
- 概念の共通定義
- セマンティック用語へのマッピング 温度、現在の価値
- ID は OMNA に登録されています

◆ サーバー上のパス階層に埋め込むことができる
- 一例：/home/weather/3303/0/5700



Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min Measured Value	5601	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value measured by the sensor since power ON or reset
Max Measured Value	5602	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value measured by the sensor since power ON or reset

図 3-13　IPSO BAROMETER OBJECT 制御の一例

出典：　IPSO Smart Object for IoT、P4,5より

（https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot）

3.6　ZigBee

3.6.1　概要

ZigBee は、近距離無線ネットワークの世界標準規格の一つであり、2017 年 9 月で 37 ケ国、400 社以上がメンバー、2,200 以上の認定商品がある。

技術的には、信頼性のある、低消費電力・低コストの無線通信として2001年からZigBee Allianceにて研究が進められてきた。サービスプロバイダー、インストーラ、小売業者に適した技術で、電池駆動可能な超小型機器への実装に向いている。

基礎部分の（電気的な）仕様は IEEE 802.15.4 として規格化されている。論理層以上の機器間の通信プロトコルについては「ZigBee Alliance」が仕様の策定を行っている。なお、ZigBee の名称はミツバチ（Bee）がジグザグに飛び回る行動にちなんで名付けられたとされている。

3.6.2　詳細

ZigBeeは幅広いスマートホームデバイスのための言語であり、企業はホームモニタリング、エネルギー管理、冷暖房、セキュリティ、およびコンビニエンスデバイスの統合エコシステムを提供できる。

ZigBeeがカバーする範囲は、OSI参照モデルのネットワーク層以上の部分で、物理層／MAC層についてはIEEE802.15.4を採用している。　ZigBeeはPAN（Personal Area Network）に分類されるが、ネットワークトポロジーとして、スター、ツリー（木構造)、メッシュをサポートすることで市場の様々な要求に応えることができる。

具体的なデータモデルとしては、ZigBee Cluster Libraryがある。ZigBeeでは「プロファイル」という形で交換するデータの形式と解釈を細かく定義して運用している。パケット毎にそのパケットがどのプロファイル用なのかということを明示して送信する。（図3-14）

具体的なプロファイルは各種の仕様を持っており、それらの例の一部を以下に記載する。

図 3-14　ZigBee の７層モデル

出典：　ZigBee入門「データ送受信」より引用

（http://www.skyley.com/wiki/index.php?ZigBee%E5%85%A5%E9%96%80）

具体的な制御対象例は以下のとおりである。

- Home Automation 1.2
- Smart Energy 1.1b
- Smart Energy 1.2
- Smart Energy 1.2a
- Smart Energy 1.2b
- Telecommunication Services 1.0
- Health Care 1.0
- RF4CE – Remote Control 1.0
- RF4CE – Input Device 1.0
- Remote Control 2.0
- Light Link 1.0
- IP 1.0

- Commercial Building Automation 1.0
- Gateway 1.0
- Green Power 1.0 (Optional battery-less remote control feature of Zigbee 2012)

一例としてのON/OFFの制御モデルを図3-15に示す。

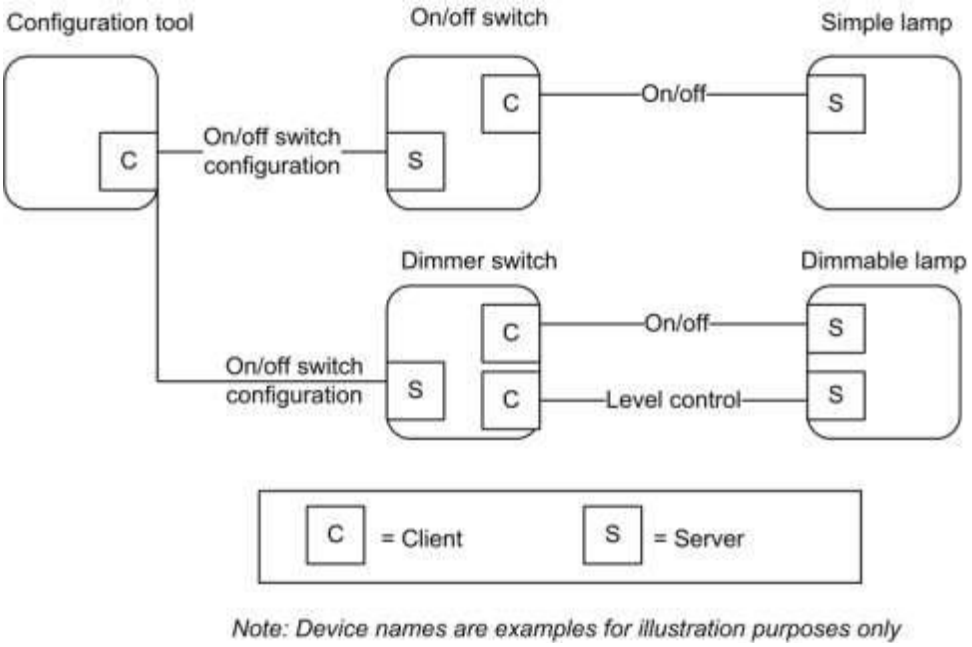


図 3-15 Typical Usage of On/Off and Level Control Clusters

出典： ZigBee Cluster Library Specification Revision 6
Figure 3-2. Typical Usage of On/Off and Level Control Clustersより引用
（http://www.zigbee.org/~zigbeeor/wp-content/uploads/2014/10/07-5123-06-zigbee-cluster-library-specification.pdf）

表 3-3　制御コマンド例

ID	Name	Description
0x0006	On/Off	Attributes and commands for switching devices between 'On' and 'Off' states.
0x0007	On/Off Switch Configuration	Attributes and commands for configuring on/off switching devices
0x0008	Level Control	Attributes and commands for controlling a characteristic of devices that can be set to a level between fully 'On' and fully 'Off'.

出典：　ZigBee Cluster Library Specification Revision 6,
Table 3-3. On/Off and Level Control Clustersより引用
（http://www.zigbee.org/~zigbeeor/wp-content/uploads/2014/10/07-5123-06-zigbee-cluster-library-specification.pdf）

3.7　IIC

3.7.1　概要

IIC（Industrial Internet Consortium）は 2014 年 3 月に米国企業 5 社（AT&T、Cisco、General Electric、IBM、Intel）により Industrial IoT（IIoT）の実現のために設立されたコンソーシアムである。

IIC が定めた共通のアーキテクチャやフレームワークをガイドラインとして使用しており、一例として IIC Connectivity Framework がありシステム間のデータ交換が可能になる。

3.7.2　詳細

コネクティビティは、機能ドメイン内の参加者間、システム内の機能ドメイン間、システム間のデータ交換を可能にする。 交換されるデータには、センサーの更新、イベント、アラーム、ステータスの変更、コマンド、および構成の更新が含まれている。 接続性は、図 3-16 に示すように、Industrial Internet Reference Architecture1 で定義されている機能ドメイン間のクロスカット機能で、接続性は、インダストリアルインターネット参照アーキテクチャにおける分野横断的な機能である。 これは、機能ドメイン内（管理、運用、情報、アプリケーション、ビジネス）の参加者間でデータを交換する機能を提供する。

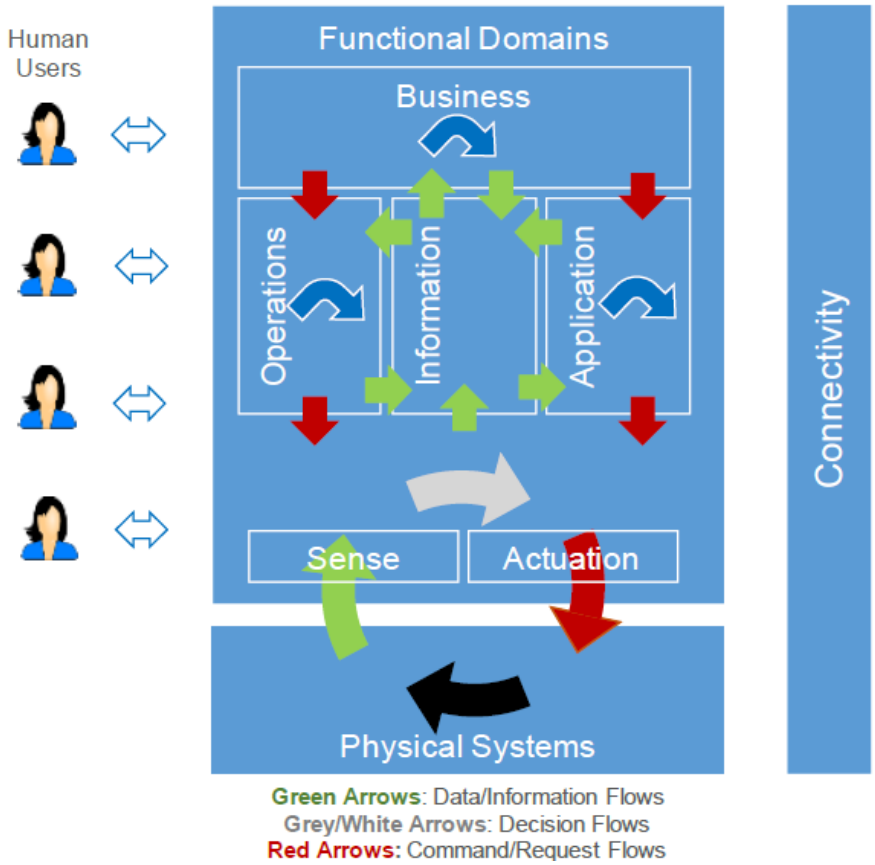


図 3-16　Industrial Internet Reference Architecture

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Figure1-1 Industrial Internet Reference Architecture より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.01_PB_20180228.pdf ）

IIoT（Industrial Internet of Things）接続スタックモデルを図 3-17 に示す。IIoT 接続スタックモデルと、Industrial Internet Reference Architecture（IIRA）3 のクロスカット機能としての接続範囲を示している。 接続機能は、IIoT システム内の機能ドメイン内の機能ドメイン間のデータ共有メカニズムを提供している。

図 3-16 は、IIoT 接続スタックモデル、および Industrial Internet Reference Architecture（IIRA）内の分野横断的機能としての接続の範囲を示している。 接続機能は、機能ドメイン内の参加者間および IIoT システム内の機能ドメイン間でのデータ共有メカニズムを提供する。The data services framework in the distributed data interoperability and management function は、コネクティビティフレームワークによって提供される foundation for semantic interoperability により、インダストリアルインターネット参照アーキテクチャ（IIRA）1 の動的な構成およびコーディネーション機能の semantic interoperability の基礎を提供する。

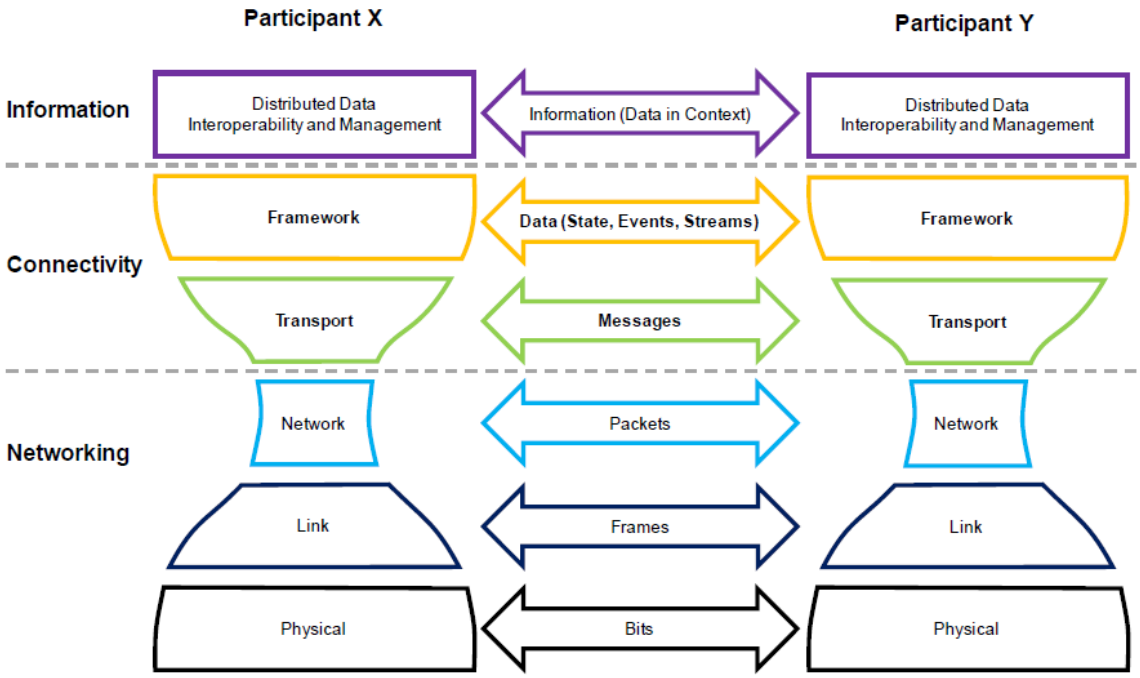


図 3-17　IIoT（Industrial Internet of Things）接続スタックモデル

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Figure 2-1: Industrial Internet Connectivity Stack Model. より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.01_PB_20180228.pdf）

この IIoT 接続機能レイヤの役割と範囲を表 3-4 に示す。

表 3-4　Role and scope of the Connectivity functional layers

IIoT Connectivity Stack Model	Correspondence to OSI Model (ISO/IEC 7498)	Correspondence to Internet Model (RFC 1122)	Correspondence to Levels of Conceptual Interoperability
Framework Layer	7. Application 6. Presentation 5. Session	Application Layer	Syntactic Interoperability: Structured data types shared between endpoints. Introduces a common structure to share data; i.e., a common data structure is shared. On this level, a common protocol is used to exchange data; the structure of the data exchanged is unambiguously defined.
Transport Layer	4. Transport	Transport Layer	Technical Interoperability: Bits and Bytes shared between endpoints, using an unambiguously defined communication protocol.
Network	3. Network	Internet Layer	Packets shared between endpoints that may not be on the same physical link. Packets are routed between physical links by a “network router”.
Link	2. Data Link	Link Layer	Digital Frames shared between endpoints on a shared substratum (link).
Physical	1. Physical		Analog signal modulation between endpoints on a shared substratum.

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Table 2-1: Role and scope of the Connectivity functional layers.より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf）.

次に機能ドメイン間のやり取りの例を図 3-18 に示す。異なる機能ドメイン間で横方向での相互運用性によるコミュニケーションの目標を実現するには、図に示すように、各業界・標準間で、標準化されたコアゲートウェイで、各コアを接続することにより実現する。

各コア接続規格には、他のすべてのコア規格に対する標準化されたゲートウェイが必要で、コア規格が追加されるたびに、複雑さと相互運用性の課題が高まる。 設計をいくつかのコア接続規格に限定することで、機能ドメイン全体で IIoT システムのニーズをカバーし、業界間での水平方向の相互運用性の目標を達成している。

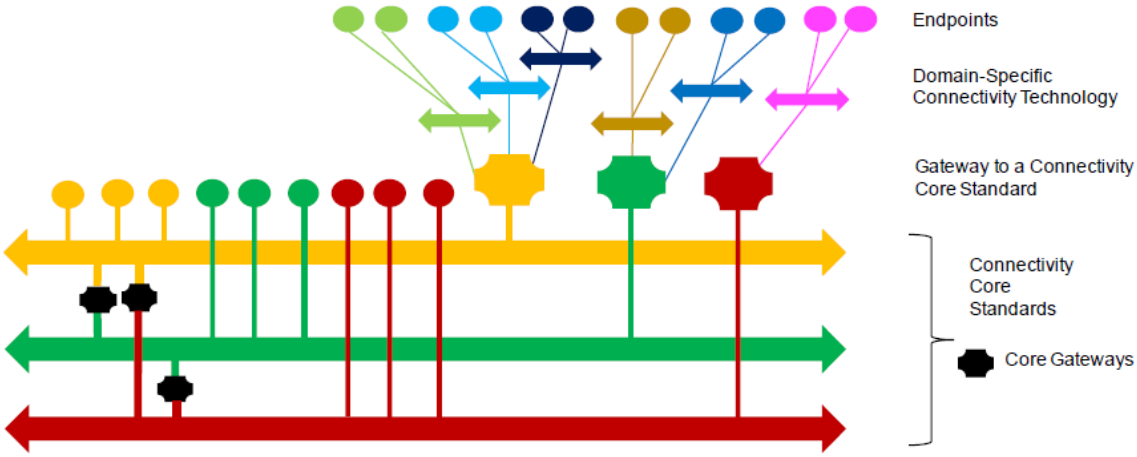


図 3-18　standardized Core Gateway

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Figure 3-4: Each core connectivity standard より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.01_PB_20180228.pdf）

図 3-19 は、IIoT 接続フレームワークとトランスポートをまとめたもので、汎用的な使用のためにとらわれない方法で作成された接続フレームワークを示している。 点線で囲まれたボックスは、複数の業界・標準にまたがって適用可能な特定のアプリケーションに焦点を当てて、特定業界の業界で起きた接続基準を示している。 いくつかの接続フレームワークは、独自のトランスポートプロトコル（例えば、DDS、OPC-UA：次項目参照）を定義しており、これらのフレームワークは、フレームワークとトランスポート層ボックスとの間に隙間なく示されている。 その他のもの（Web サービス、oneM2M など）は、汎用転送プロトコルに依存している。 網（IP）および下位層（有線および無線）も参考として示している 。

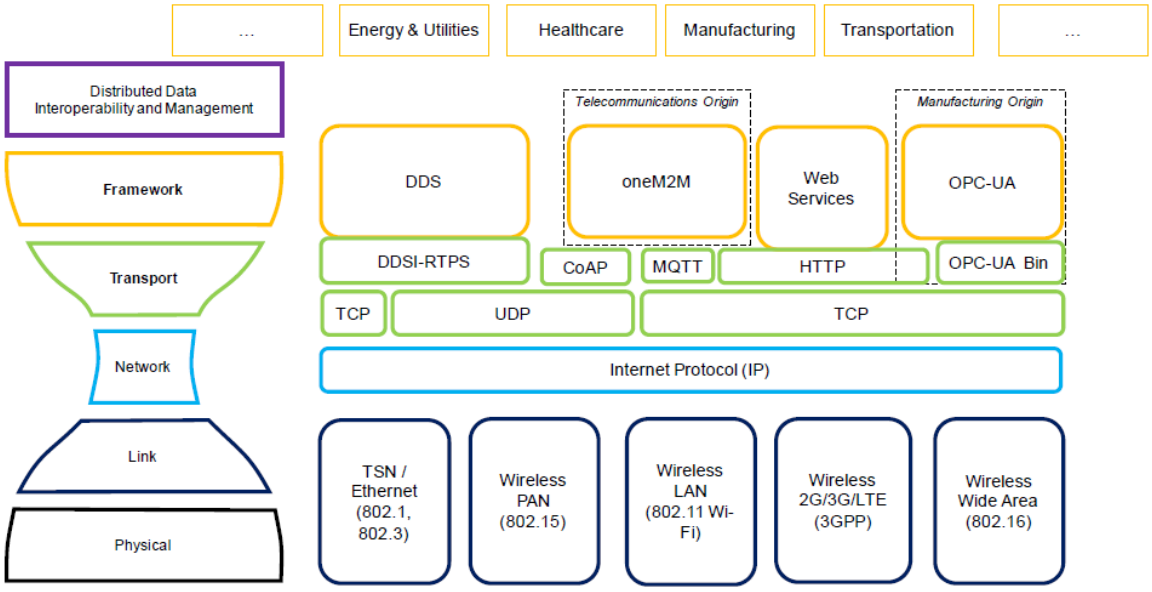


図 3-19　The prominent IIoT connectivity frameworks and transports

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Figure 7-1: IIoT connectivity standards. Dotted より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf）

主要な Connectivity framework 機能には、データリソースモデル、パブリッシュ - サブスクライブおよび要求 - 応答データ交換パターン、データのサービス品質、データセキュリティ、およびプログラミング API が含まれる。 これらを図 3-20 に示すと同時に、主要機能について以下に記載する。

◆　DATA RESOURCE MODEL

Connectivity framework は、時間とともに状態を変えることができるデータオブジェクト 1 を表す方法を提供する。 データオブジェクトは、プログラミング言語のように、構造化されたフィールドの集まりで、それは階層的であり得、そして静的または動的に型付けされ得る。

Connectivity framework は、参加者の間でデータオブジェクトへの変更を配布する。さまざまなアプリケーション分野や業界のデータモデルは、通常、接続フレームワークによって提供される抽象データオブジェクトにマッピングされる。

◆　ID AND ADDRESSING

Connectivity framework は、各データオブジェクトを識別してアドレス指定する手段を提供する。 ID は、データオブジェクトをアドレス指定し、データオブジェクト表現内のフィールドを読み書きするために使用される。

・データオブジェクト表現内の明示的な ID フィールド

・データオブジェクト表現で特別にマークされたフィールドに基づく暗黙の ID

・デバイス、アプリケーション、またはネットワークエンドポイントの名前空間内の統一リソース識別子（URI）。

◆　DATA TYPE SYSTEM

構文上の相互運用性を保証するために、コネクティビティスタックはデータの構文を記述する方法を提供する。データ型は、データの解釈に課せられる構文上の制約で、暗黙のうちに（例えばコ

ード内で）または明示的にデータ型を共有またはマッピングすることなくシステムを接続することは不可能である。

Connectivity framework は、プログラミング環境においてデータオブジェクトを構造として表現するため、およびワイヤ上で通信されるデータをフォーマットするためのデータ型システムを提供する。データ型システムは、静的型プログラミング言語（例えば C、C++、C#または Java）に見られるデータ型のようにオブジェクト指向型でもよいし、または動的型プログラミング言語に見られる動的データ型（例えばC、JavaScript、Python、Lua）でも良い。

データ型システムは、データ型の進化を管理する手段を提供する必要がある。これには、バージョン間のバージョニングおよび割り当て可能性ルールが含まれるため、新しいバージョンのデータ型を使用しているアプリケーションは、古いバージョンのデータ型を使用しているアプリケーションと可能な限り最大限に通信できる。

データ型システムはまた、通信中（移動中）および格納中（静止中）の直列化データフォーマット、ならびにプログラミング言語表現から直列化フォーマットに直列化する動作、およびプログラミング言語表現に逆シリアル化する動作を定義する。

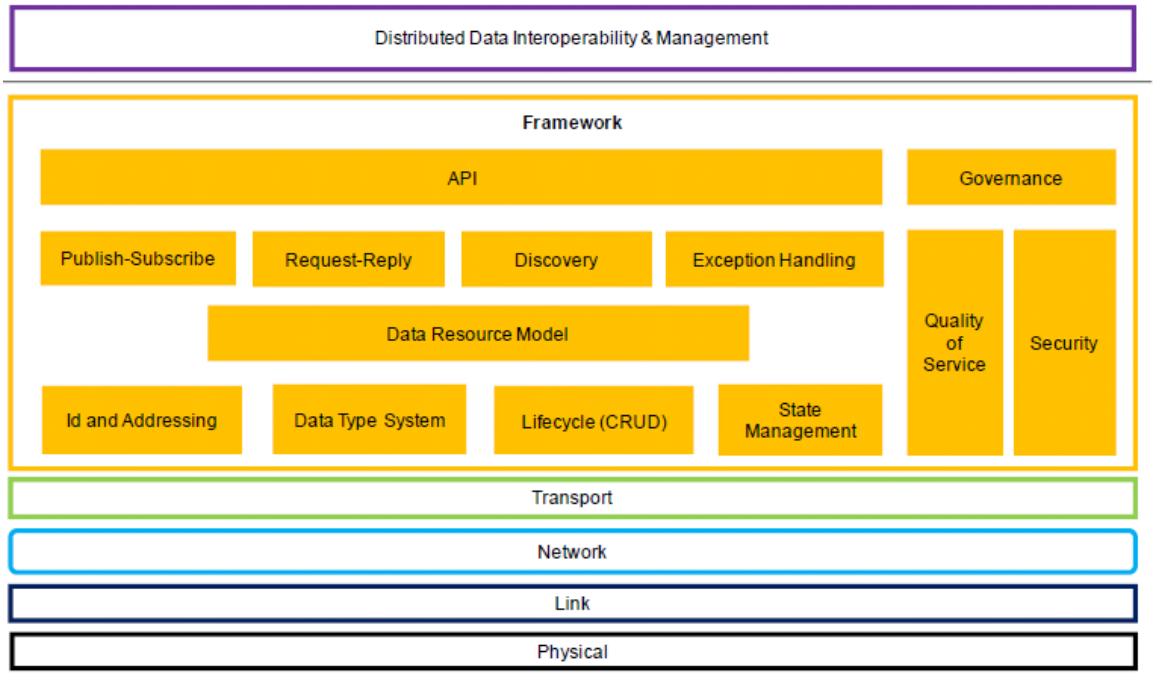


図 3-20　Connectivity framework layer functions

出典：　The Industrial Internet of Things Volume G5: Connectivity Framework
Figure 4-1: Connectivity framework layer functions.より
（https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf）

3.8　OPC-UA

3.8.1　概要

OPC ホームページ（https://jp.opcfoundation.org/about/what-is-opc/）によると、OPC は、産業オートメーション分野やその他業界における、安全で信頼性あるデータ交換を目的とした相互運用を行うための標準規格である。

OPCはプラットフォームから独立し、多くの製造ベンダーのデバイス間で、シームレスな情報の流れを確保し、OPC Foundationは、この標準規格の開発と維持に責任を負っている。また、OPC標準規格は、業界の製造ベンダー、エンドユーザー、そしてソフトウェア開発者が開発した一連の仕様であり、これら仕様は、リアルタイムデータのアクセス、アラームとイベントの監視、履歴データやその他アプリケーションのアクセスを含み、クライアントとサーバー間に加え、サーバーとサーバー間のインターフェースを定義している。

この標準規格は、1996年に初めて発表され、当初、OPC標準規格は、Microsoft® Windows®オペレーティング・ システムに限られていたが、現在、 OPC Classic として知られるそれら仕様は、製造、建設、オートメーション、石油・ガス、再生可能エネルギー、ユーティリティ等、数々の業界で広く適応されている。

その後、次世代のOPC 技術として、 OPC UA（統合アーキテクチャ）は安全で信頼性あるプラットフォーム非依存の相互運用性に向け大きく進化し、工場やプロセス制御装置から大規模な上位情報系システムへ情報転送をするために設計された。

3.8.2　詳細

OPC-UA（IEC 62541、OPC Unified Architecture:統合アーキテクチャ）は次世代の OPC 技術として、安全で信頼性あるプラットフォーム非依存の相互運用性に向け大きく進化したものである。

以下、OPC ホームページからの抜粋を記載する。

OPC UA は、工場やプロセス制御装置から大規模な上位情報系システムへ情報転送をするために設計されており、Industrie 4.0 の通信の情報交換として重要な役割を持っている。図 3-21 に OPC UA 階層モデルを示す。

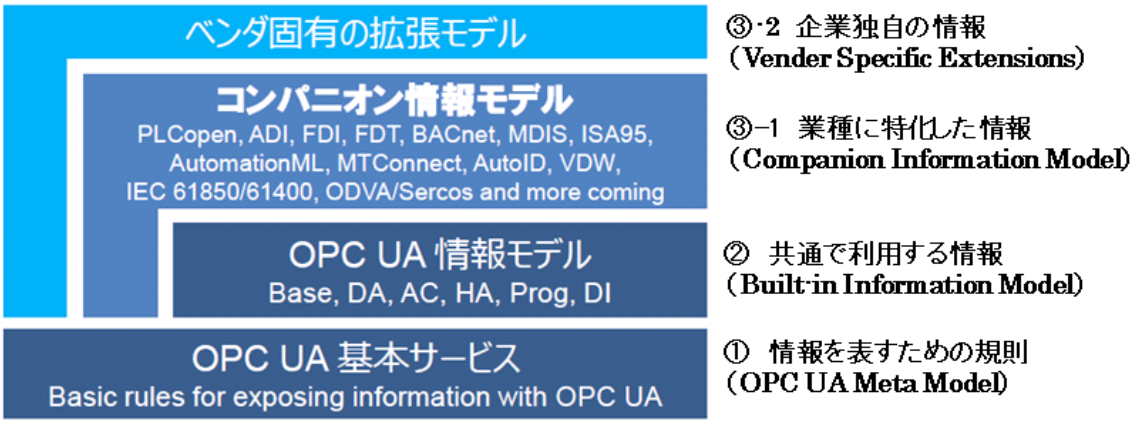


図 3-21　OPC UA 階層モデル

出典：　OPC UA コラボレーションン取組とそのユーザー価値
第一のカギ：伝えるための枠組み=情報モデルより
（https://jp.opcfoundation.org/wp-content/uploads/sites/2/2017/12/OPCDay2017_07_collaboration.pdf）

図中の①~③は具体的には、以下の意味を示す。

① モデル定義基盤（Meta Model）の確立

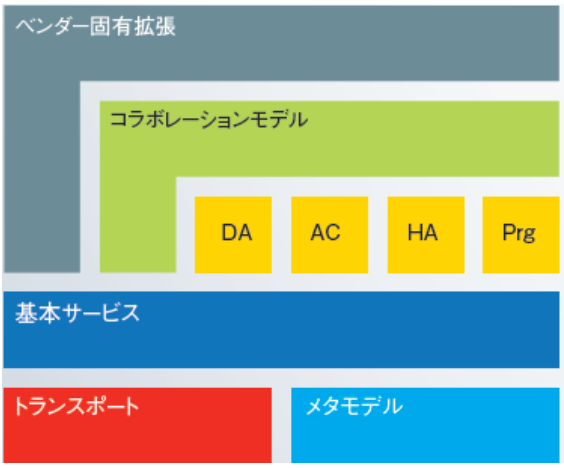
② 基本デル構成要素（Built-in Information Model）の定義

③ 拡張モデルの拡充

-1 協調領域（Companion Information Model）

-2　競争領域（Vender Specific Extensions）

OPC UA 階層モデル別の見方をすると、図 3-22、3-23 に示す構成となる。



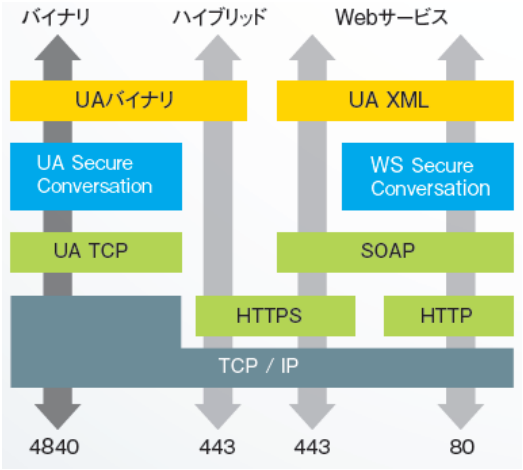


図 3-22　OPC UA 階層モデル　　**図 3-23　OPC UA トランスポート・プロファイル**

出典：OPC ホームページ、OPC UA 技術の詳細より

https://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industrie4-and-IoT-JP-v3.pdf

具体的には、以下の構成となる。

- ◆ トランスポート – OPC UA アプリケーション間のデータ交換機構に対応：異なる要求事項に対応する様々なトランスポート・プロトコル（速度とスループットへの最適化 = UA TCP と UA バイナリ、ファイアーウォール親和性 = HTTP+ SOAP）。
- ◆ メタモデル – OPC UA で情報モデルを発行するためのルールと基本コンポーネントを指定：様々な基本ノードと基本タイプも含む。
- ◆ サービス – 情報を提供するサーバーと情報を使用するクライアント間のインターフェースを構成：情報モデルは階層化アプローチに従っている。

以下、図中の具体的な記載について記載する。

＜汎用 OPC UA 情報モデル＞

OPC UA　階層モデルに示した情報モデルは、一般的な有効情報（例えば、アラームやオートメーションデータ）のモデルとして、すでに OPC UA によって定義されている。一般定義をさらに特化したその他の情報モデルはここから派生する。そのため、一般モデルに対してプログラムされたクライアントは、ある程度特化したモデルも処理することができる。

①　データアクセス（DA）

データアクセス、略して DA は、リアルタイムデータ（すなわち、基本的な産業用または業務用プロセスデータの現在の状態と挙動を表すデータ）のモデリングを記述する。これにはアナログ・ディスクリート変数、工学単位、品質コードの定義が含まれる。データの出所は、センサー、コントローラ、ポジションエンコーダーなどで、これらをデバイスの I/O に直接接続することも、リモートデバイスの直列接続とフィールドバスで接続することも可能である。

② アラーム＆コンディション（AC）

この情報モデルは、状態（ダイアログ、アラーム）の処理方法を定義する。状態の変化は、イベントを起こし、クライアントはこうしたイベントを登録し、利用可能な関連値の中から、イベントレポート（例えば、メッセージテキスト、ACK 挙動）に含めて受け取りたいものを選択することができる。

③ ヒストリアクセス(HA)

HA により、バリアブルとイベントのヒストリカルデータにアクセスでき、これらのデータの読み取り、書き込み、変更が可能である。データはデータベース、アーカイブ、その他のストレージシステムに配置することができ、幅広い集計処理機能により、サーバーで前処理を行うことができる。

④ プログラム（PRG）

「プログラム」は、バッチ処理の演算や操作のような複雑なタスクを表す。各プログラムはステートマシンで示され、状態遷移は、クライアントに対してメッセージを起こす。

＜統合アドレス空間モデル＞

次にオブジェクトの記載方法の例について記載する。オブジェクトモデルは、生産データ、アラーム、イベント、履歴データをひとつの OPC UA サーバーに統合可能にしている。これにより、例えば、温度測定機器を温度値、アラーム・パラメーター、対応するアラーム限度と共にオブジェクトとして示すことができる。図 3-23 にその例を記載する。

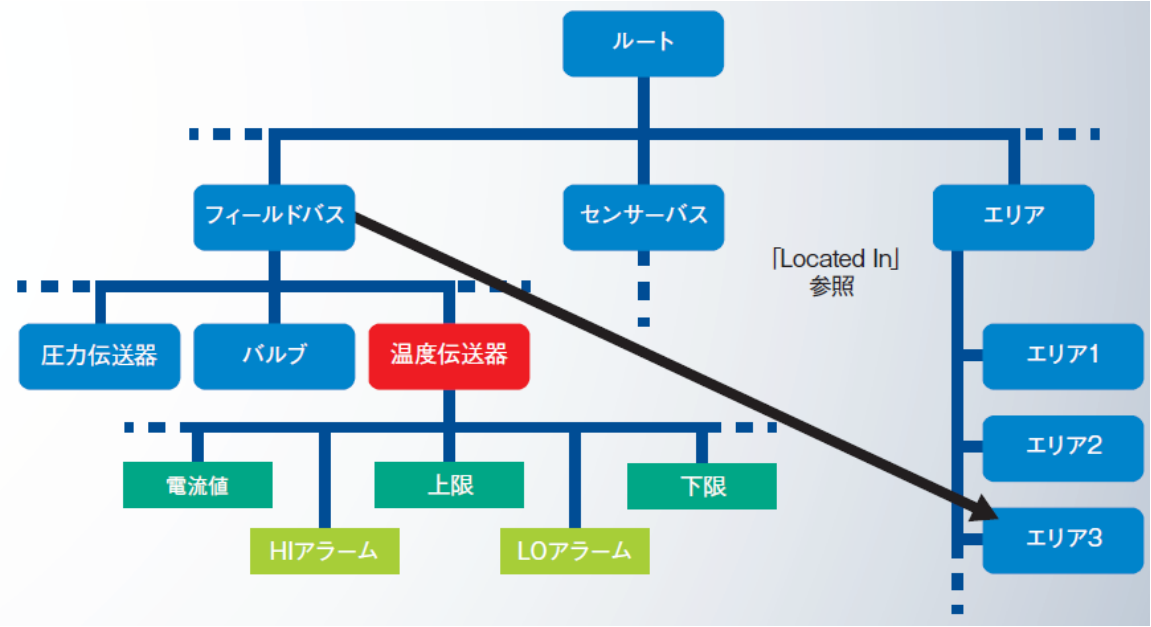


図 3-23 OPC UA 一貫性のあるアドレス空間

出典： OPC ホームページ、統合アドレス空間モデルより、
(https://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industrie4-and-IoT-JP-v3.pdf)

OPC UA は異なるアドレス空間とサービスを統合および標準化し、OPC UA アプリケーションはひとつのインターフェースだけでナビゲーションできるようになる。

OPC UA のアドレス空間は、階層的に構成され、クライアントとサーバーの相互運用性を促進する。最上位レベルはサーバー用に標準化され、アドレス空間のすべてのノードには階層経由で到達すること

ができ、ノード間に相互参照を付けられ、アドレス空間が結束したノードのネットワークを形成する。

OPC UA のアドレス空間には、インスタンス（インスタンス空間）だけでなく、インスタンスタイプ（タイプ空間）も含まれている。

OPC UA は、Industrie 4.0 の情報レベルの相互運用性に関する要求事項を満たす、成熟した規格である。OPC UA は、包括的な情報モデル（What）を公開し、個別に開発されたアプリケーション間で複雑なデータを交換するために、プロトコルとサービス（How）を提供する。

＜テクノロジー固有の情報モデル＞

制御／オートメーションを扱う標準化委員会は、テクノロジー固有の情報モデルを作成している。例えば、IEC61804 (EDDL)、ISA SP 103（フィールドデバイスツール）、ISA-S88、ISA-S95、IEC-TC57-CIM などがある。これらは、特定の知識分野の単位、関係、ワークフローの記述を標準化しているため、重要な仕様である。

OPC Foundation は、新しい規格をゼロから開発するため、他の組織と意欲的に協力しており、これらの組織の情報モデルを OPC UA（コンパニオン規格）にマッピングするルールは、共同作業部会で規定されている。図 3-24 に IEC62541: OPC UA 仕様を示す。

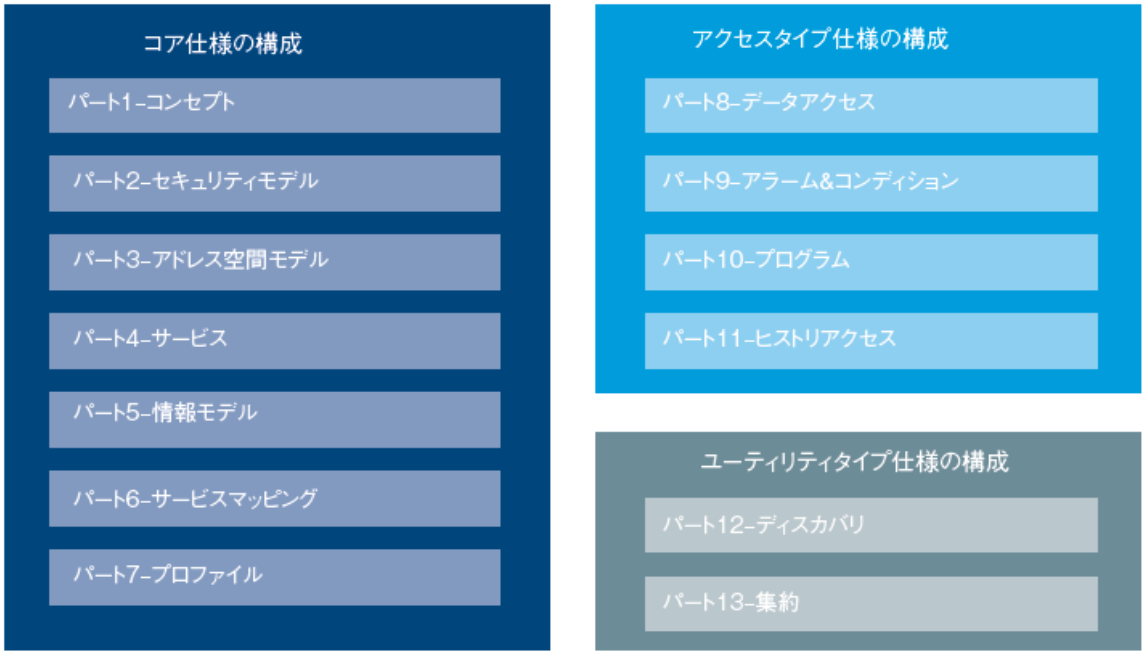


図 3-24　IEC62541: OPC UA 仕様

出典：OPC ホームページより、
ttps://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industrie4-and-IoT-JP-v3.pdf

現在作成中のコンパニオン規格 は以下の通りである。

- OPC UA for Devices (IEC 62541-100)
- OPC UA for Analyser Devices
- OPC UA for Field Device Integration

- OPC UA for Programmable Controllersbased on IEC61131-3
- OPC UA for Enterprise and Control Systems based on ISA 95
- OPC UA for Machine Tool Connectivity (MTConnect)
- OPC UA for AutoID (AIM)
- OPC UA for BACnet (Building Automation)
-

＜テクノロジー固有の情報モデル＞

その他の取り組みと具体的なデータモデルの例について、以下に記載する。

- コラボレーションの取り組み
  - OPC が適用 される 市場・業界団体と協力し適用するための仕様（コンパニオン仕様
  - 適用事例（ユースケース）や適用のためガイドラインを創出
- 工場自動化分野:
  - PLCopen (PLC)
  - FDT-Group (ソフトウェアインターフェース)
  - PROFIBUS, EtherCAT, CC-Link, IO-Link など(産業用ネットワーク)
  - OMAC (包装機)
  - EUROMAP (成形機)
- ビルオートメーション:
  - BACnet (ビルオートメーション)
- オイル・ガス:
  - MDIS (MCS-DCS 間通信)
- IoT、セキュリティ技術:
  - OpenFog（IoT）
  - M2M Alliance (セキュリティ)

さらに、5 件のコンパニオン仕様・ガイドラインが会員に公開中。(2017/12/10 現在)

＜具体的な情報モデル＞

OPC UA は、図 3-25 にあるように、関連する型システムを含む一般的なオブジェクトモデルを定義している。 このデータモデルに加えて、すべての物理システムを OPC UA に準拠するモデルに変換して OPC UA サーバーで表現する方法を記述するための規則が定義されている。 このメタモデルを使用して、あらゆる種類のデバイス、機能、およびシステム情報を記述できる。 基本型システムは、オブジェクト間の関係、いわゆる参照、および多重継承をサポートしている。 したがって、それは現代のオブジェクト指向プログラミング言語と比較することができる。 基本モデルは、参照型とデータ型だけでなく、オブジェクト型と変数型も提供する。 このモデルに基づいて、OPC UA はメタデータとセマンティクスを含むあらゆる種類のデータを表現できる。

◆ UA モデル

OPC UA データモデルは、UA 情報モデルの基盤を形成する。 特殊なモデルであるため、データアクセス、アラームと状態、履歴アクセス、プログラムなどの特定の機能を追加することで基盤を拡張します。

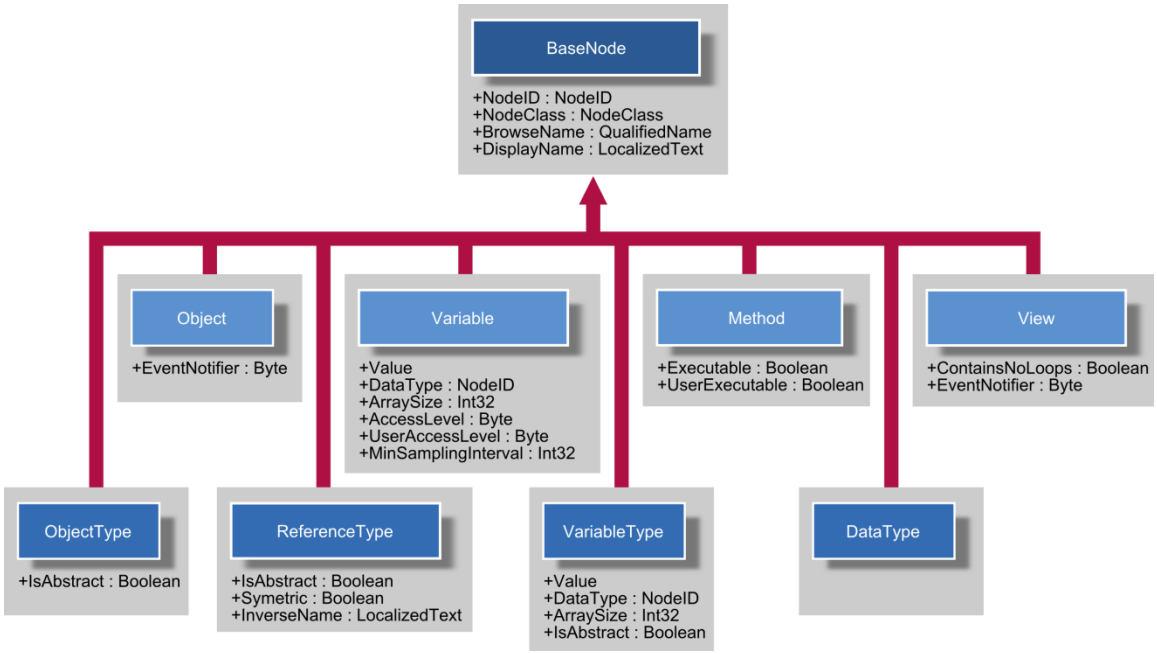


図 3-25　IEC62541: OPC UA 情報モデル

出典：ascolab 社資料より、

（http://www.ascolab.com/en/technology-unified-architecture/meta-model.html）

4.　まとめ

以上をまとめると、下記の分類となると考えられる。

- ◆　方向性１：　OPC-UA の例の様に情報モデルを階層的に定義
- ◆　方向性２：　W3C の例のように物のデータモデルと Binding　Template をもちいて連携し、アクション、およびイベントを含む相互作用モデルを構成⇒Protocol Binding のような汎用的な考え
- ◆　方向性３：　OCF　（OIC Core Smart Home）の例、データモデルとデータプロトコル（メッセージング）を明確に区別

  Echonet オブジェクト規定、IPSO Smart Objects も同様（データの構成、記述は別途）

  ⇒３種のデータモデル、Property/Action/Event や Set/Get/Observe など
- ◆　方向性４：　IIC Connectivity Framework のように複数の業界・標準にまたがって適用可能なように個別のモデルを規定し Core-Gateway を設けて相互連携、方向性は OPC-UA も同じ

いづれにしても、今後台頭してくる様々な情報モデルに対して調査を継続して、機会を狙って整理して行くなどの見直しが必要と考える。