【注意】 この文書は、<u>2019年11月6日付の W3C勧告候補 [Web of Things (WoT) Thing Description」(原文は英語</u>)を、WoT JP Commuityが翻訳と修正をおこなって公開しているものです。この文書の正式版は、あくまでW3Cのサイト内にある英語版であり、この文書には翻訳上の間違い、あるいは不適切な表現が含まれている可能性がありますのでご注意ください。

W3C

Web of Things (WoT) Thing Description

W3C Candidate Recommendation 6 November 2019

This version:

https://www.w3.org/TR/2019/CR-wot-thing-description-20191106/ Latest published version:

https://www.w3.org/TR/wot-thing-description/

Latest editor's draft:

https://w3c.github.io/wot-thing-description/

Implementation report:

https://w3c.github.io/wot-thing-description/testing/report.html

Previous version:

https://www.w3.org/TR/2019/CR-wot-thing-description-20190516/

Editors:

Sebastian Kaebisch (Siemens AG)

Takuki Kamiya (<u>Fujitsu Laboratories of America</u>) Michael McCool (<u>Intel</u>)

Victor Charpenay (Siemens AG) Matthias Kovatsch (<u>Huawei</u>)

Participate:
GitHub w3c/wot-thing-description

File a bug Commit history

Pull requests

Contributors:

In the GitHub repository

Repository: We are on GitHub

File a bug

Copyright © 2017-2019 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply

摘要

本文書は、WoT(Web of Things) Thing Description (TD)のための形式モデルおよび共通表現を解説するものである。TDは、<u>モノ</u>のメタデータおよびインターフェースを記述し、<u>モノ</u>は、WoT内での対話を提供し、それに参加する物理エンティティまたは仮想エンティティのアブストラクションである。TDは、多様なデバイスの結合、および、多様なアブリの相互運用を可能にする少数のボキャブラリに基づく対話セットを提供する。TDは、デフォルトでJSON-LD 処理も可能なJSON 形式でエンコードされる。後者は、マシン理解可能な方法で<u>モノ</u>に関する知識を表すための効果的な基盤を提供する。TDインスタンスは、モノ自体によって提供されるか、または、モノにリソース制限(例えば、制限されたメモリ空間)があるとき、あるいは、WoT互換性レガシーデバイスのTDによって レトロフィットされているときには、外部で提供される。

本文書のステータス

本項では、発行時点での本文書のステータスを説明する。このため、他文書が本文書よりも新しい場合もある。現行のW3C出版物一覧および本テクニカルレポートの最新版は、 https://www.w3.org/TR/のW3Cテクニカルレポート索引に記載されている。

(本什様書内で黄色表示されている)以下の危険状態の機能は、CR期間中に報告されるインプリメンテーション経験、及び・あるいは、受け取るコメントが不十分であるために削除されることがあ

- 第5.3.3.7項CertSecurityScheme、第5.3.3.9項PublicSecurityScheme、第5.3.3.10項PoPSecurityScheme記載のセキュリティスキームに関連する全項。
- 第5.3.3.11項OAuth25securityScheme内のimplicit, password, clientフローに関するというボキャブラリ用語とアサーション。
- 前述の第5.4項デフォルト値定義内の上記項に関連する全デフォルト値。
- 不完全な書き込み拒否を可能にするwriteallpropertiesのためのビヘイビアアサーション

本文書は、WoTワーキンググループによって推奨候補として発行されたものであり、W3C推奨になることを意図している。

<u>GitHubの問題指摘</u>は本仕様の考察の材料として歓迎される。また、それを<u>public-wot-wg@w3.org(アーカイブ</u>)で我々のメーリングリストにコメントを送ることもできる。

W3Cは、本文書が信頼できるものであると考えられていることを示し、開発者コミュニティによるインプリメンテーションを奨励するために、推奨候補を公開するもので、この推奨候補は、2019年 12月4日以前に提案推奨として更新される予定である。

[・]ググループのインプリメンテーションレポートを参照いただきたい。

推奨候補としての公開はW3Cメンバーによる推奨を意味するものではない。本文書は、常に、更新されたり、他の文書と置き換えられたり、また破棄される可能性もある草案文書であり、"作業中文

本文書は、<u>W3C特許ポリシー</u>に従い運営しているグループが作成したものである。W3Cでは、ワーキンググループの成果物に関係する<u>公的な特許公開リスト</u>を管理しており、そのページには特許開 示にあたっての指示も掲載されている。<u>必須特許請求</u>を含むと信じる特許について実知識を持つ者は、<u>W3C特許ポリシー第6項</u>に従って、その情報を開示しなければならない。

本文書は、2019年3月1日のW3Cプロセス文書に準拠する。

Table of Contents

- 1. 1. Introduction
- 2. 2. Conformance
- 3. 3. Terminology
 4. 4. Namespaces
- 5. 5. TD Information Model
 - 1. 5.1 Overview 2. 5.2 Preliminaries
 - 3. 5.3 Class Definitions
 - 1. 5.3.1 Core Vocabulary Definitions
 - 1. 5.3.1.1 Thing 2. 5.3.1.2 InteractionAffordance
 - 3. 5.3.1.3 Property Affordance
 - 4. 5.3.1.4 ActionAffordance
 - 5. 5.3.1.5 EventAffordance 6. 5.3.1.6 VersionInfo
 - 7. 5.3.1.7 MultiLanguage2. 5.3.2 Data Schema Vocabulary Definitions
 - 1. 5.3.2.1 DataSchema
 - 2. 5.3.2.2 ArraySchema
 - 3. 5.3.2.3 BooleanSchema
 - 4. 5.3.2.4 NumberSchema 5. 5.3.2.5 IntegerSchema
 - 6. 5.3.2.6 ObjectSchema
 - 7. 5.3.2.7 StringSchema

```
3. 5.3.3 Security Vocabulary Definitions
1. 5.3.3.1 SecurityScheme
                                   2. 5.3.3.2 NoSecurityScheme
                                   3. 5.3.3.3 BasicSecurityScheme
                                   4. 5.3.3.4 DigestSecurityScheme
5. 5.3.3.5 APIKeySecurityScheme
                                   6. 5.3.3.6 BearerSecurityScheme
                                   7. 5.3.3.7 CertSecurityScheme
                                         5.3.3.8 PSKSecurityScheme
                                   9. 5.3.3.9 PublicSecurityScheme
                                        5.3.3.10 PoPSecurityScheme
                                  11. 5.3.3.11 OAuth2SecuritySche
                         4. 5.3.4 Hypermedia Controls Vocabulary Definitions
                                   1. 5.3.4.1 Link
2. 5.3.4.2 Form
              3. 5.3.4.3 ExpectedResport
4. 5.4 Default Value Definitions

    6. 6. TD Representation Format
    1. 6.1 Mapping to JSON Types

    6.2 Omitting Default Values
    6.3 Information Model Serialization

    6.3.1 Thing Root Object
    6.3.2 Human-Readable Metadata
    6.3.3 version
                         4. 6.3.4 securityDefinitions and security
                         5. 6.3.5 properties
                         6. 6.3.6 actions
                         7. 6.3.7 events
                        8. 6.3.8 links
                        9. 6.3.9 forms
                       10. 6.3.10 Data Schemas
               4. 6.4 Identification
    7. 7. TD Context Extensions

    7.1 Semantic Annotations
    7.2 Adding Protocol Bindings

   3. 7.3 Adding Security Schemes8. 8. Behavioral Assertions

    8.1 Security Configurations
    8.2 Data Schemas

    8.3 Protocol Bindings
    1. 8.3.1 Protocol Binding based on HTTP

    S.3.2 Other Protocol Bindings
    Security and Privacy Considerations
    Security and Privacy Considerations
    Security and Privacy Considerations
    Security and Privacy Considerations
    Security and Privacy Risk
    Security and Privacy Risk
    Security and Privacy Risk

2. 9.2 Immutable Identifiers Privacy Risk
3. 9.3 Fingerprinting Privacy Risk
4. 9.4 Globally Unique Identifier Privacy Risk
5. 9.5 TD Interception and Tampering Security Risk
6. 9.6 Context Interception and Tampering Security Risk
7. 9.7 Inferencing of Personally Identifiable Information Privacy Risk

10. 10. IANA Considerations
1. 10.1 application/td-json Media Type Registration
2. 10.2 CoAP Content-Format Registration
11. A. Example Thing Description Instances

    A.1 MyLampThing Example with CoAP Protocol Binding
    A.2 MyLightSensor Example with MQTT Protocol Binding

    2. A.2 MyLightSensor Example with MQTT Protocol Bin
    3. A.3 Webhook Event Example
    3. B. JSON Schema for TD Instance Validation
    4. C.1 Thing Description Templates
    5. C.1 Thing Description Template Examples
    6. C.1.1 Thing Description Template: Lamp
    7. C.1.2 Thing Description Template: Buzzer
    8. JSON-LD Context Usage
    9. E. Recent Specification Changes
    1. E.1 Changes from First Candidate Recommendation
    2. E.2 Changes from Third Public Working Draft
    16. F. Acknowledgements
  16. F. Acknowledgements
  17. G. References1. G.1 Normative references
               2. G.2 Informative references
目次
    1. はじめに
   2. 適合性
   3. 用語
    4. 名前空間

 TD情報モデル

              1. 概要
              2 前付け
              3. クラス定義
                         1. 中核ボキャブラリの定義
                                   1. Thing
                                   2. InteractionAffordance
                                   3. PropertyAffordance
                                   4. ActionAffordance
                                   5. EventAffordance
                                    6. VersionInfo
                                  Multilanguage
                         2. データ・スキーマ・ボキャブラリの定義
                                   1. DataSchema
```

8. 5.3.2.8 NullSchema

2. ArraySchema

- 3. BooleanSchema
- 4. NumberSchema
- 5. IntegerSchema
- 6. ObjectSchema
- 7. StringSchema
- 8. NullSchema

3. セキュリティボキャブラリの定義

- 1. SecurityScheme
- 2. NOSecurityScheme
- 3. BasicSecurithScheme
- 4. DigestSecurithScheme
- 5. APIKeySecurituScheme
- 6. BearerSecurityScheme
- 7. CertSecurityScheme
- 8. PSKsecurithScheme
- 9. PubloicSecurityScheme
- 10. PopSecurityScheme
- 11. OAuth2SecurithScheme
- 4. ハイパーメディア制御ボキャブラリの定義
 - 1. Link
 - 2. Form
 - 3. ExpectedResponse
- 1. デフォルト値の定義

1. **TD表現**フォーマット

- 1. JSONタイプへのマッピング
- 2. デフォルト値の省略
- 3. 情報モデルのシリアライズ
 - 1. Thing Root オブジェクト
 - 2. 人間が読み取り可能メタデータ
 - 3. Version
 - 4. securithDefinition to security
 - 5. properties
 - 6. actions
 - 7. events
 - 8. links
 - 9. forms
 - 10. データスキーマ
- 1. ID

1. TDコンテキスト拡張子

- 1. 意味論的注釈
- 2. プロトコルバインディングの追加
- 3. セキュリティ方式の追加

2. ビヘイビアアサーション

- 1. セキュリティ構成
- 2. データスキーマ
- 3. プロトコルバインディング
 - 1. HTTPに基づくプロトコルバインディング
 - 2. その他のプロトコルバインディング

9. セキュリティとプライバシーに関する考慮事項

- 1. プライバシーリスクをデリフェレンスするコンテキスト
- 2. 不変識別子プライバシーリスク
- 3. プライバシーリスクフィンガープリンティング
- 4. グローバル一意識別子プライバシーリスク
- 5. TD傍受とセキュリティリスクの改竄
- 6. コンテキスト傍受とセキュリティリスクの改竄
- 7. 個人情報プライバシーリスクの推測

10. IANAに関する考慮事項

1. application/td+json メディアタイプ登録

2. CoAP Content-Format Registration

A.

- 1. CoAP プロトコルバインディングを使用したMyLampThing の例
- 2. MQTT プロトコルバインディングを使用したMyLightSensor の例
- 3. Webhookイベントの例
- B. TDインスタンス確認のためのJSONスキーマ
- C. TDテンプレート
 - 1. TDテンプレートの例
 - 1. TDテンプレート: ランプ
 - 2. TDテンプレート: ブザー
- D. JSON-LDコンテキスト用法
- E. 最近の仕様変更
 - 1. 第一次推奨候補からの変更点
 - 2. 第三次公開作業ドラフトからの変更点
- F. 謝辞
- G. 参考文献
 - 1. 標準参考文献
 - 2. 参考文献

1. はじめに §

本項は標準ではない。

WoT Thing Description (TD)は、W3C Web of Things (W3CWoT)における中心的なビルディングブロックであり、 $\underline{\mathtt{t}}_{-}$ (ウェブサイトのindex.htmlに酪似する)のエントリポイントとみなすことができる。TDインスタンスには、4つの主要なコンポーネント、すなわち、 $\underline{\mathtt{t}}_{-}$ 自体に関する文字メタデータ、 $\underline{\mathtt{t}}_{-}$ はどのように使用できるかを示す<u>対話アフォーダンス</u>、マシンが理解できるようにするために $\underline{\mathtt{t}}_{-}$ と交換されるデータ<u>スキーマ</u>、および、ウェブ上の他の $\underline{\mathtt{t}}_{-}$ 、または、文書との形式的または非形式的な関係を表すための<u>ウェブリンク</u>で構成されている。

W3C WoTの<u>対話モデル</u>は、3タイプの<u>対話アフォーダンス</u>を定義している。すなわち、プロパティ(<u>PropertyAffordance</u>クラス)は、現在の値を取得する、または、動作ステータスを設定するなどパラメータを感知・制御するために使用することができる。アクション(<u>ActionAffordance</u>クラス)は、物理的な(したがって時間のかかる)プロセスの呼び出しをモデル化するが、既存のブラットフォームのRPCのような呼び出しを抽象化するために使用することもできる。イベント(<u>EventAffordance</u>クラス)は、通知、個別イベント、または値のストリームが非同期で受信側に送信される通信のブッシュモデルに使用される。詳細は[<u>WOT-ARCHITECTURE</u>]参照。

一般に、TDは、URIスキーム [RFC3986](例えば、http、coapなど,[IANA-URI-SCHEMES])、メディアタイプに基づくコンテンツタイプ[RFC2046](例えば、application/son、application/xml、application/cbor、application/exiなど、[IANA-MEDIA-TYPES]、および、セキュリティメカニズム(TDインスタンスのシリアライズは、JSON [RFC8259]に基づいており、ここでは、本仕様書で定義するとおり、JSON名はTDボキャブラリ用語を指す。さらに、TDのJSONシリアライズは、拡張および豊富な意味処理を可能にするために、JSON-LD 1.1[JSON-LD 1.1]のシンタックスを遵守しているようには、ATM 1.5 (JSON-LD 1.1 (

<u>例</u>1は、TDインスタンスであり、MyLampThingというタイトルのランプという<u>モノ</u>を記述するプロパティ、アクション、イベントを持つ<u>対話モデル</u>となっている。

```
Example 1: Thing Description Sample

{
    "@context": https://www.wd.org/2019/wot/ddv1",
    "dit".umdevops:32473-WoTLamp-1234",
    "title": "MyLamp Thing",
    "security/Disaic_sc"],
    "properties";
    "status": {
        "type": "string",
        "forms": [['met": "https://mylamp.example.com/status"]]
    },
}

/*Context: ['met": "https://mylamp.example.com/loggle"]
}

/*Coverhealing": ['metext-thips://mylamp.example.com/loggle"]

/*Coverhealing": ['metext-thi
```

例1:TDの例

```
{
    "@context": https://www.w3.org/2019/wot/td/v1,
    "id": "urn:dev:ops:32473-WoTLamp-1234",
    "title": "MyLampThing",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in":"header"}
    },
    "security": ["basic_sc"],
    "properties": {
        "status": {
            "type": "string",
            "forms": [{"href": "https://mylamp.example.com/status"}]
    }
```

このTDの例からは、タイトルステータスを持つ1つの<u>プロパティアフォーダンス</u>が存在することがわかる。さらに、このプロパティが、(hrefメンバーによってforms構造内でアナウンスされる)URI https://mylamp.example.com/statusでGETメソッドを使って"(安全な形式の)HTTPプロトコルを介してアクセス可能であり、ストリングベースのステータス値を返すということを示す情報が提供されている。GETメソッドの使用は、明示的に述べられていないが、本文書で定義されるデフォルト想定の1つである。

同様に、<u>アクションアフォーダンス</u>が、https://mylamp.example.com/toggle資源上のPOSTメソッドを使って切り替えステータスをトグルするために指定され、ここでもPOSTは、アクションを呼び 出すためのデフォルト想定である。

<u>イベントアフォーダンス</u>により、<u>モノ</u>が非同期メッセージのためのメカニズムを送信できるようになる。ここで、ランプの過熱事象が起こった場合に通知されるサブスクリプションは、 https://mylamp.example.com/oh上の長いボーリングサブプロトコルを有するHTTPを使用することによって取得することができる。

本例は、また、アクセスのためにユーザ名およびパスワードを必要とするbasicセキュリティスキームを指定している。セキュリティスキームには、まず、securityDefinitionで名前が与えられ、次に、セキュリティセクションでその名前を指定することによってアクティブ化されることに留意されたい。HTTPプロトコルとの併用で、本例は、HTTP基本認証の使用を説明している。最上位レベルでの少なくとも1つのセキュリティスキームの指定は必須であり、これによってすべてのリソースに関するデフォルトのアクセス要件が与えられる。しかし、セキュリティスキームは、また、モノレベルで与えられる構成をオーバーライドするフォームレベルで与えられる構成を使ってフォームごとに指定することもでき、きめの細かいアクセス制御の指定を可能にしている。アクセス制御機構が何も使用されていないことを示すために特別なnosecセキュリティスキームを使用することも可能である。追加例が後述される。

TDは、いくつかの名前空間にコンテキスト定義を追加する可能性を提供する。このメカニズムは、正式な知識、例えば、アプリケーションの特定のドメインの論理規則が所与の名前空間で見つけることができる場合には、TDインスタンスのコンテンツに追加の意味を結合するために使用することができる。コンテキスト情報は、formsフィールドで宣言された基礎となる通信プロトコルのいくつかの構成およびビヘイビアの指定するときに役立つ。例2は、SAREF-Smart Appliance Reference Ontology(スマート家電参照オントロジー)- (SMARTM2M)を参照するものとしてプレフィックス sarefを宣言するために、@contextに第2の定義を導入して、例1のTDサンブルを拡張したものである。このIoTオントロジーには、モノの意味論と対話アフォーダンスを持たせる@typeフィールドの値として設定することが出来る意味論上ラベルと解釈される用語が含まれる。下記の例では、モノはsaref:LightSwitch、statusプロパティは saref:OnOffState、toggleアクションは saref:ToggleCommandとラベル付けされる。

```
Example 2: Thing Description with TD Context Extension for semantic annotations
     "@context": [
                        w.w3.org/2019/wot/td/v1".
        { "saref": "https://w3id.org/saref#" }
     "id": "urn:dev:ops:32473-WoTLamp-1234",
     "title": "MyLampThing",
"@type": "saref:LightSwitch",
"securityDefinitions": {"basic_sc": {
        "scheme": "basic"
         "in": "header'
       security": ["basic_sc"],
     "properties": {
    "status": {
        "@type": "saref:OnOffState",
        "type": "string",
        "string",
            "forms": [{
    "href": "https://mylamp.example.com/status"
           }]
     }-
"actions": {
"toggle": {
    "@type": "saref:ToggleCommand",
    "forms": [{
               orms": [{
"href": "https://mylamp.example.com/toggle"
           }]
     "events": {
         "overheating": {
           verneaming: {
   "data": {"type": "string"},
   "forms": [{
        "href": "https://mylamp.example.com/oh"
```

```
例2: 意味論的注釈用のTDコンテキスト拡張子を持つTD
{
    "@context": [
        "https://www.w3.org/2019/wot/td/v1".
        {"saref":"https://w3id.org/saref#"}
    ],
    "id": "urn:dev:ops:32473-WoTLamp-1234",
    "name": "MyLampThing",
    "@type": "saref:LightSwitch",
    "securityDefinitions": {"basic_sc": {
        "scheme": "basic",
```

```
"in": "header
}},
"security": ["basic sc"]
"properties": {
    "status": {
        "type": "string",
        "forms": [{
            "@type": "saref:GetCommand".
            "href": "https://mylamp.example.com/status"
       }]
    }
},
"actions": {
    "toggle": {
        "forms": [{
            "@type": "saref:ToggleCommand",
            "href": "https://mylamp.example.com/toggle"
       }]
},
"events": {
    "overheating": {
        "data": {"type": "string"},
        "forms": [{
            "@type": "saref:NotifyCommand",
            "href": "https://mylamp.example.com/oh"
       }]
   }
}
```

いくつかの@context内の宣言メカニズムは、JSON-LDによって指定される。TD インスタンスは、本仕様のバージョン1.1 [j<u>son-ld11]</u> に準拠している。TDインスタンスは、また、RDFドキュメント として処理もできる。(意味処理の詳細は、付録<u>D.JSON-LDコンテキスト用法</u>と名前空間IRI内の資料(例: <u>https://www.w3.org/2019/wot/td</u>)参照。)

2. 適合性 §

標準ではないとされる項と同様に、本仕様書におけるすべてのオーサリングガイドライン、図、例、および注釈は標準ではない。本仕様内のこのようなものはすべて標準である。

本文書内のキーワード"場合がある"、"しなければならない"、"してはならない"、"推奨される"、"べきである"、及び、"べきではない"は、このように太字で表示されているときにのみB<u>OP 14</u> [RFC2119] [RFC8174]の記述に従い解釈するものとする

TDインスタンスは、それがTDシリアライズに関する第5項TD情報モデル及び第6項TD表現フォーマットで 標準ステートメントに従う場合、本仕様に準拠する。

TDインスタンスを確認するためのJSONスキーマ[JSON-SCHEMA]は、付録B. TDインスタンス確認JSONスキーマ</u>で提供されている。

3. 用語 §

モノ、コンシューマ、TD, 対話モデル、対話アフォーダンス、プロパティ、アクション、イベント、プロトコルバインディング、サービアント、WoTインターフェース、WoTランタイム等の基本 WoT用語は、WoTアーキテクチャ仕様の<u>第3項</u>で定義される [<u>WOTARCHITECTURE</u>]。

また、本仕様では、以下の定義をしている。

TDコンテキスト拡張子

TD情報モデル

制約が適用される事前定義の<u>ボキャブラリ</u>で構築された<u>クラス</u>定義、したがって、これらの<u>ボキャブラリ</u>の意味論を定義する。クラス定義は、通常、<u>シグネチャ(ボキャブラリ用語)</u>及びその<u>シグネチャ</u>内の関数で表現される。<u>TD情報モデル</u>には、また、<u>クラス</u>に対するグローバル関数として定義されるデフォルト値が含まれる。 TDプロセッサ

所与のフォーマットでTDの何らかの内部表現をシリアライズする、また∕あるいは、そのフォーマットからそれをシリアライズ解除することができるシステム。TDプロセッサは、意味的に矛盾す るTD、すなわち、モノックラスのインスタンス関係に対する制約を満たすことができないTDを検出しなければならない。そのために、TDプロセッサは、可能なすべてのデフォルト値が割り当てられているTDの標準フォームを計算することができるかもしれない。TDプロセッサは、通常、WoTランタイムのサブシステムである。TDプロセッサのインブリメンテーションは、(TDドキュメントをシリアライズできる)TDプロデューサのみ、あるいは、(TDドキュメントからシリアライズできる)TDプロデューサのみ、あるいは、(TDドキュメントからシリアライズできる)TDプロデューサのみ、あるいは、(TDドキュメントからシリアライズ解除することができる)TDコンシューマであろう。 をシリアライズできる)TDプロデューサ TDシリアライズ またはTDドキュメント

サービアント間で保存および交換できる<u>TD</u>のテキストまたはバイナリ表現。<u>TDシリアライズ</u>は、ネットワーク上で交換されるときにメディアタイプによって識別される所与の表現フォーマットに従 う。 <u>ID</u>のデフォルト表現フォーマットは、本仕様が定義するようにJSONベースである。 ボキャブラリ

名前空間IRIで識別される<u>ボキャブラリ用語</u>集。

用語 ・ボキャブラリ用語

が開始している。 文字列。用語が<u>ボキャブラリ</u>の一部である場合、すなわち、名前空間IRIとプレフィックスがついている場合、それは<u>ボキャブラリ用語</u>と呼ばれる。読みやすくするために、本書に記載されている<u>ボ キャブラリ用語</u>は、コンパクトな形式で書かれており、完全なIRIとはなっていない。

これらの定義は、第5.2項前付けの中でさらに詳細される。

4. 名前空間 §

本仕様第5項TD情報モデルで定義されている本バージョンTD情報モデルは、以下のIRIによって識別される。

https://www.w3.org/2019/wot/td/v1

URI[<u>RFC3986]</u>でもあるこのIRI<u>[RFC3987]</u>は、<u>JSON-LDコンテキストファイル[son-ld11]</u>を得るためにデリフェレンスすることができ、<u>TDドキュメント</u>内のコンパクトなストリングを完全なIRIベー スの<u>ボキャブラリ用語</u>に拡張することができる。しかしながら、この処理は、JSONベースの<u>TDドキュメント</u>を<u>TDプロセッサ</u>インプリメンテーションのオプション機能であるRDFに変換するときに のみ要求される。

本仕様では、<u>ボキャブラリ用語</u>は常にコンパクトなフォームで表されている。それらの拡張されたフォームは、それらが属する<u>ボキャブラリ</u>の名前空間IRIでアクセスすることができる。これらの名前空間は<u>第5.3項クラス定義</u>の構造に従う。<u>TD情報モデル</u>で使用される各<u>ボキャブラリ</u>は、以下のように、それ自体の名前空間IRIを持つ。

 Vocabulary
 Namespace IRI

 Core
 https://www.w3.org/2019/wot/td#

 Data Schema
 https://www.w3.org/2019/wot/json-schema/

 Security
 https://www.w3.org/2019/wot/security#

 Hypermedia Controls https://www.w3.org/2019/wot/hypermedia#

 ボキャブラリ
 名前空間IRI

 コア
 https://www.w3.org/2019/wot/id#

 データスキーマ
 https://www.w3.org/2019/wot/jsonschema#

 セキュリティ
 https://www.w3.org/2019/wot/security#

 ハイパーメディアコントロール
 https://www.w3.org/2019/wot/hypermedia#

<u>ボキャブラリ</u>は互いに独立している。それらは、他のW3C仕様で再利用・拡張することもある。<u>ボキャブラリ</u>の設計を大きく変更するには、必ず新しい年ベースの名前空間URIの割り当てが必要 になる。<u>TD情報モデル</u>の総体的な一貫性を維持するために、関連するJSON-LDコンテキストファイルは、大きくはない変更、特に、新しい<u>用語</u>の追加を識別するために、あらゆるバージョンが それ自体のURI (v1、v1.1、v2.)を持つようにバージョン化されることに留意されたい。

いくつかの名前空間IRIの<u>ボキャブラリ</u>では大きくはない変更のみをすることができるので、そのコンテンツは、安全なキャッシュか、または、アプリへの組み込みが可能である。名前空間IRIの下で比較的静的なコンテンツを公開することの利点の一つは、制約されたデバイス間で交換されるメッセージのベイロードサイズの最適化である。また、プライベートネットワークから公的に利用可能なボキャブラリにアクセスするデバイスが原因となるプライバシー漏洩を回避する(<u>第</u>9.1項プライバシーリスクをデリフェレンスするコンテキストも参照)。

5. TD情報モデル §

本項では、TD情報モデルについて紹介する。TD情報モデルは、TDおよびそのシリアライズ処理のための概念的な基盤となる。これについては、別途、第<u>6項TD</u>表現フォーマットで説明する。

5.1概要 §

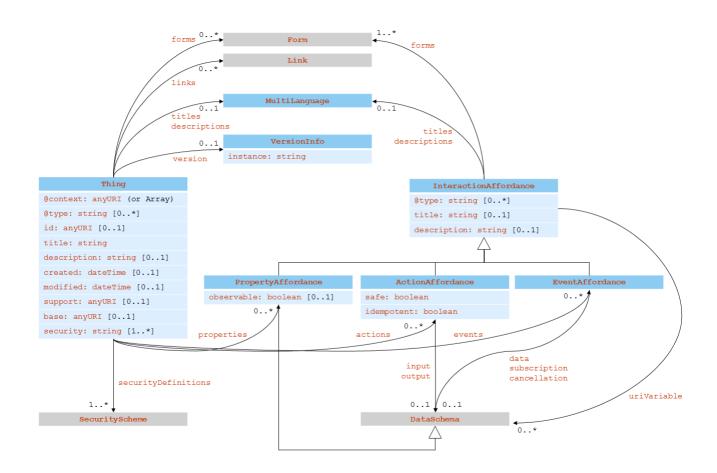
TD情報モデルは、以下の独立した<u>ボキャブラリ</u>で構築される。

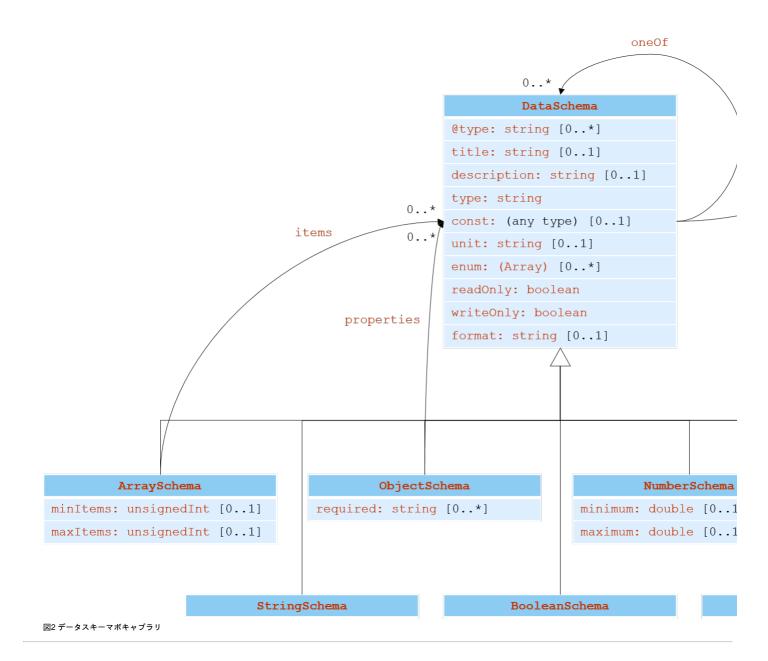
- プロパティ、アクション、イベントアフォーダンス、対話モデルを反映する中核TDボキャブラリ[WOT-ARCHITECTURE]
- JSON スキーマによって定義された用語(のサブセット) を含むデータ・スキーマ・ボキャブラリ[JSONSCHEMA]
- セキュリティメカニズムとその設定要件を識別するWoTセキュリティ<u>ボキャブラリ</u>
- ウェブリンクとフォームを使用してRESTful通信の主な原則をエンコードするハイパーメディア制御ボキャブラリ

これらの<u>ボキャブラリ</u>のそれぞれは、本質的に、従来のオブジェクト指向の意味でオブジェクトとして解釈される、データ構造を構築するために使用することができる<u>用語</u>セットである。オブジェクトは、クラスのインスタンスであり、プロパティを有する。W3C WoTのコンテキストでは、これらは、<u>モノ</u>およびそれらの<u>対話アフォーダンス</u>を示す。オブジェクトの正式定義は、<u>第5.2項前付</u> <u>け</u>に記載されている。<u>TD情報モデル</u>の主要要素は、<u>第5.3項クラス定義</u>の中で提示する。<u>デフォルト値</u>がある場合、TD 内で特定のオブジェクトプロパティを省略してもよい。デフォルトのリストは第5.4項デフォルト値の定義にある。

下記のUML図は、T<u>D情報モデル</u>の概要を示している。これは、すべてのクラスとクラス間に存在する関連付けを表として表示している。矢印が示すようにクラス<u>Thing(モノ)</u>から始まる。読みやすくするために、図は、4つの基本<u>ボキャブラリ</u>のおのおの一個とし、4つの部分に分割されている。

е





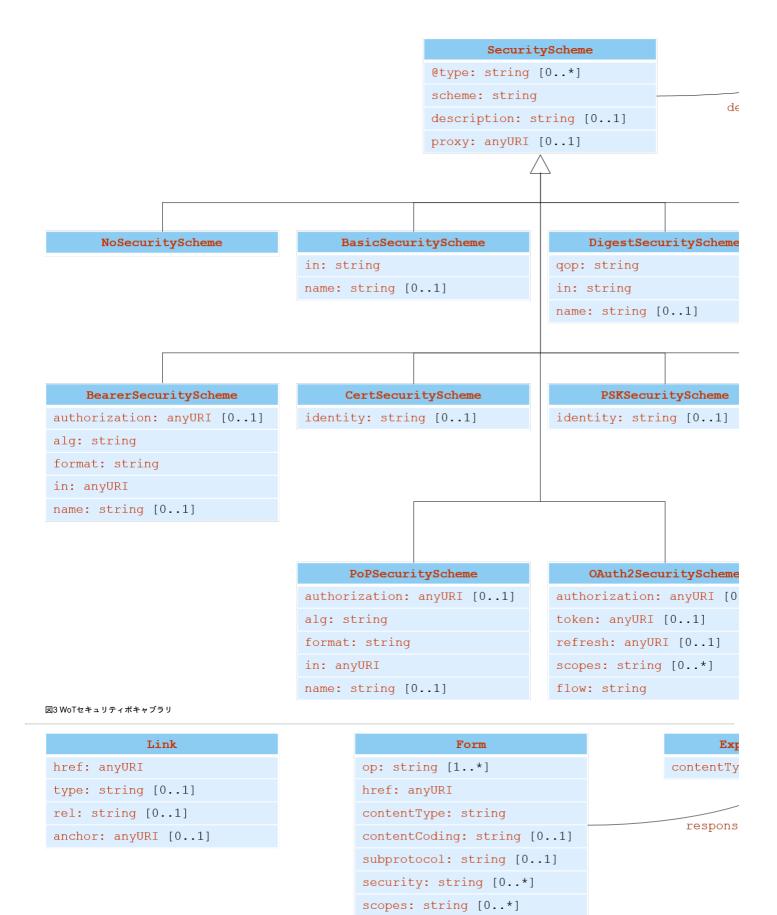


図4 ハイパーメディア制御ボキャブラリ

5.2 前付け 🕯

ッリーベース文書(すなわち、生JSON処理)および豊富な意味論的ウェブツーリング(すなわち、JSON-LD処理)に関する簡易規則の両方による容易な処理が可能なモデルを提供するために、本文書 はそれに応じてTD情報モデルを構築するための以下のような形式的な前付けを定義する。

本項における全ての定義は、セットであり、これは、直感的に、それ自体がセットであり得る要素の集合である。全ての任意に複雑なデータ構造は、セットとして定義することができる。特に、オブジェクトは、以下のように再帰的に定義されるデータ構造である。

- <u>用語</u>は、x + y = y = 0に属している場合も、また、属さない場合もあるx = y = 0である。
- 名前が<u>用語</u>で、値が別の<u>オブジェクト</u>である名前と値のペアのセットも<u>オブジェクト</u>である。

この定義は、<u>オブジェクト</u>が同じ名前で複数の名前-値ベアを含むことを妨げるものではないが、一般に、本仕様では考慮されない。要素が名前として数字のみを有する<u>オブジェクト</u>は、配列と呼ばれる。同様に、名前として(<u>ボキャブラリ</u>に属さない)<u>用語</u>のみの要素を有する<u>オブジェクト</u>は、マップと呼ばれる。<u>マップ</u>内のいくつかの名前-値ベアに現れるすべての名前は、<u>マップ</u>の範囲内において一意であると想定される。

さらに、<u>オブジェクト</u>は、いくつかの<u>クラス</u>のインスタンスとすることができる。<u>ボキャブラリ用語</u>で表される<u>クラス</u>は、まず、シグネチャと呼ばれる<u>ボキャブラリ用語</u>セットで定義される。<u>シ</u>グネチャが空であるクラスは、シンプルタイプと呼ばれる。

<u>クラス</u>の<u>シグネチャ</u>は、<u>クラス</u>をさらに定義する2つの関数、すなわち、**割当関数**および型関数を構築することができる。<u>クラスの割当関数</u>は、入力として<u>クラスのシグネチャのボキャブラリ用語</u>を うけ、true、または、falseのいずれかを出力として返す。直感的には、<u>割当関数</u>は、<u>クラス</u>の例を挙げるときに<u>シグネチャ</u>の要素が必須であるか、または、<u>任意選択</u>であるかを示す。また、<u>クラ</u> スの型関数も、<u>クラスのシグニチャ</u>の<u>ボキャブラリ用語</u>を入力として受け取り、別の<u>クラス</u>を出力として返す。これらの関数は部分的であり、それらのドメインは、定義されている<u>クラス</u>の<u>シグニ</u> チャに限定される。

これらの2つの関数に基づいて、**インスタンスの関係**は、<u>オブジェクト</u>と<u>クラス</u>とからなるペアに対して定義することができる。この関係は満たすべき制約と定義される。すなわち、以下の2つの制約の両方ともが満たされる時に<u>オブジェクト</u>は<u>クラス</u>のインスタンスとなる。

- クラスの割り当て関数がtrueを返す全ての用語について、オブジェクトは、名前としてボキャブラリ用語を有する名前-値ペアを一個含んでいるか
- オブジェクトの何らかの名前-値ペアの中で名前として使用されるクラスのシグネチャ内の全てのボキャブラリ用語に対して、そのペアの値は、与えられたボキャブラリ用語に対するクラスの型関数が返すクラスのインスタンスであるのか

上記の定義によれば、<u>オブジェクト</u>は、その構造にかかわらず、あらゆる<u>単純型</u>のインスタンスである。しかし、<u>インスタンス関係</u>のもう一つの定義が<u>単純型</u>のために導入されている。つまり、<u>オブジェクト</u>が所与の語彙形式(例えば、boolean(ブール)型の場合はtrueとfalseであり、unsignedInt(アンサインドイント)型の場合1、2、3など)を持つ<u>用語</u>である場合、そのオブジェクトは<u>単純タイ</u>プのインスタンスである。

さらに、**パラメータ化クラス**と呼ばれる追加の<u>クラス</u>を汎用<u>マップ</u>および<u>配列</u>構造から導出することができる。<u>オブジェクト</u>が、その全ての名前--値ペアの値がこの<u>クラス</u>のインスタンスであるような<u>マップ</u>である場合、そのオブジェクトは、ある<u>クラス</u>のマップ、即ち、ある<u>クラス</u>でパラメータ化されたマップタイプのインスタンスである。同じことが<u>配列</u>にも当てはまる。

最後に、<u>クラス</u>は、前者のあらゆるインスタンスが後者のインスタンスでもある場合、何らかの他の<u>クラス</u>の**サブクラス**である。

上記のすべての定義に従い、「1D情報モデルは、クラス名(ボキャブラリ用語)、シグニチャ(ボキャブラリ用語のセット)、割当関数、および型関数を含むクラス定義のセットとして理解されるべきである。これらのクラス定義は、第5.3項クラス定義で表にまとめられている。各表とも、割当列内の値「必須」(あるいは、「オブション」)は、割り当て関数が、対応するボキャブラリ用語に対してtrue(あるいは、false)を返すということを示している。

慣例により、<u>単純型</u>は小文字で始まる名前で表される。<u>TD情報モデル</u>は、XMLスキーマ <u>[xmlschema11-2-20120405]</u>で定義されている以下の<u>単純型</u>を参照する。すなわち、string, anyURI, dateTime, integer, unsignedInt, double, booleanである。これらの定義(すなわち、それらの語彙形式の仕様)は、本<u>TD情報モデル</u>の範囲外である。

さらに、TD情報モデルは、<u>ボキャブラリ用語</u>のベアに関するグローバル関数を定義する。この関数は、<u>クラス</u>名および別の<u>ボキャブラリ用語</u>を入力として受け取り、<u>オブジェクト</u>を返す。返された<u>オブジェクト</u>がnull(ヌル)でない場合、それは、入力<u>クラス</u>のインスタンス内の入力<u>ボキャブラリ用語</u>の何らかの割当の<u>デフォルト値</u>を表す。この関数で<u>割当関数</u>に関して上で定義された制約を緩和することが可能となる。すなわち、オブジェクトがすべての必須割当を含む場合、あるいは、<u>デフォルト値</u>が欠けている割当について存在する場合、そのオブジェクトは<u>クラス</u>のインスタンスである。デフォルト値はすべて、表<u>第5.4頁デフォルト値</u>で表の表に記載されている。<u>第5.3項クラス定義</u>の各表では、<u>TD情報モデル</u>の <u>クラス</u>と <u>ボキャブラリ用語</u>の対応する組み合わせに<u>デフォルト</u>値がある場合、割当列には値「デフォルトちり」と書かれている。

ここで使用される形式化は、抽象データ構造としての<u>オブジェクト</u>と、モノのような物理世界オブジェクト間で可能性のある関係を考慮していない。しかし、<u>TD 情報モデル</u>に含まれる全ての<u>ボキャブラリ用語</u>をRDFリソースとして再解釈し、物理世界のより大きなモデルに結合する可能性(オントロジー)に対しては考慮された。この点については、付録D. TDオントロジーで取り扱われる。意味処理の詳細は、付録D.JSON-LDコンテキスト用法と名前空間IRI内の資料(例: https://www.w3.org/2019/wot/td)参照。

5.3クラス定義 §

<u>TD</u>プロセッサは、TDが<u>第5.3.1項中核ボキャブラリ定義、第5.3.2項データ・スキーマ・ボキャブラリ定義、第5.3.3項セキュリティボキャブラリ定義</u>および<u>第5.3.4ハイパーメディア制御ボキャブラリ定義</u>で定義されているすべての<u>クラス</u>に対する<u>クラス</u>インスタンス化制約条件を満足していなければならない。

5.3.1 Core Vocabulary Definitions §

5.3.1.1 Thing 1

メタデータおよびインターフェースがWoTTDによって記述される物理エンティティまたは仮想エンティティのアブストラクションであり、仮想エンティティは1つまたは複数のモノで構成される。

Vocabulary term	Description	Assignmen	t Type
@context	JSON-LD keyword to define short-hand names called terms that are used throughout a TD document.	mandatory	anyURI or Array
@type	JSON-LD keyword to label the object with semantic tags (or types).	optional	string or Array of string
id	Identifier of the Thing in form of a URI [RFC3986] (e.g., stable URI, temporary and mutable URI, URI with local IP address, URN, etc.).	optional	anyURI
title	Provides a human-readable title (e.g., display a text for UI representation) based on a default language.	mandatory	string
titles	Provides multi-language human-readable titles (e.g., display a text for UI representation in different languages).	optional	MultiLanguage
description	Provides additional (human-readable) information based on a default language.	optional	string
descriptions	Can be used to support (human-readable) information in different languages.	optional	MultiLanguage
version	Provides version information.	optional	VersionInfo
created	Provides information when the TD instance was created.	optional	dateTime
modified	Provides information when the TD instance was last modified.	optional	dateTime
support	Provides information about the TD maintainer as URI scheme (e.g., mailto [RFC6068], tel [RFC3966], https).	optional	anyURI
base	Define the base URI that is used for all relative URI references throughout a TD document. In TD instances, all relative URIs are resolved relative to the base URI using the algorithm defined in [RFC3986]. base does not affect the URIs used in @context and the IRIs used within Linked Data [LINKED-DATA] graphs that are relevant when semantic processing is	optional	<u>anyURI</u>
	applied to TD instances.		
properties	All Property-based Interaction Affordances of the Thing.	optional	Map of PropertyAffordance
actions	All Action-based Interaction Affordances of the Thing.	optional	Map of ActionAffordance
events	All Event-based Interaction Affordances of the Thing.	optional	Map of EventAffordance
links	Provides Web links to arbitrary resources that relate to the specified Thing Description.	optional	Array of Link
forms	Set of form hypermedia controls that describe how an operation can be performed. Forms are serializations of Protocol Bindings. In this version of TD, all operations that can be described at the Thing level are concerning how to interact with the Thing's <u>Properties</u> collectively at once.	optional	Array of Form
security	Set of security definition names, chosen from those defined in securityDefinitions. These must all be satisfied for access to resources.	mandatory	string or Array of string
securityDefinitions	Set of named security configurations (definitions only). Not actually applied unless names are used in a security name-value pair.	mandatory	Map of SecurityScheme

ボキャブラリ用語	説明	割当	型
@context (@コンテキスト)	JSON-LD キーワード。TDド キュメント全体で使用される 用語と呼ばれる略記名を定義 する。	必須	anyURIまたは配列
@type (@タイプ)	JSON-LD キーワード。オブ ジェクトに意味タグ(またはタ- プ) をラベル付けする。		string(ストリング)または string(ストリング)の配列
id	URI[<u>RFC3986</u>]形式のモノの識 別子(例: 安定したURI、一 時的で不定なURI, ロー カルIPアドレスを持つUR	必須	anyURI

ı		I、URN等)。		
	title(タイトル)	デフォルト言語で人間が読み 取り可能なタイトルを提供す る(たとえば、UI 表現のテキス ト表示)。	必須	string(ストリング)
	titles(タイトル)	複数言語の人間が読み取り可能なタイトルを提供する(例えば、異なる言語でのUI表現のためのテキスト表示)。	オブション	MultiLanguage (複数言語)
	description(説明)	デフォルト言語で追加の(人間 が読み取り可能な) 情報を提供 する。	オプション	string(ストリング)
	descriptions(説明)	さまざまな言語で(人間が読み取り可能な)情報をサポートするために使用できる。	オプション	MultiLanguage(複数言語)
ı	version(バージョン)	バージョン情報を提供する。	オプション	VersionInfo(バージョン情報)
	created(作成)	TD インスタンスが作成された 時点の情報を提供する。	オプション	dateTime (日時)

modified(変更)	TD インスタンスが最後に変更 された時点の情報を提供す る。	オブション	dateTime(日時)
support(サポート)	TD管理者に関する情報をURIスキーム(例えば、 mailto[<u>RFC6068</u>]、tel [<u>RFC3966</u>]、https)として提供する。		anyURI
base(ベース)	TDドキュメント全体にわたってすべての関連URII 参照に使用さるベースURI を定義する。TDインスタンスでは、すべての関連URIは、 [RFC3986]で定義されているアルゴリズムを使用して、ベースURIに呼応して解読される。 baseは、@contextで使用され	オプション	<u>anyURI</u>
	Daseは、woontextで使用されるURIや意味処理がTDインスタンスに適用されるときに直接的に関連するLinked Data [LINKED-DATA]グラフ内で使用されるIRIに作用しない。		
properties(プロパティ)	モノのすべてのプロバティ ベースの <u>対話アフォーダン</u> <u>ス</u> 。	オプション	PropertyAffordance (プロパ ティアフォーダンス) のマップ
actions(アクション)	モノのすべてのアクション ベースの <u>対話アフォーダン</u> <u>ス</u> 。	オプション	ActionAffordance (アクションフェーダンス) のマップ
events(イベント)	モノのすべてのイベントベース の <u>対話アフォーダンス</u> 。	オプション	EventAffordance (イベント フォーダンス) のマップ
links(リンク)	指定されたTDに関連する任意 のリソースへのウェブリンク を提供する。	オプション	<u>Link</u> (リンク)の配列
forms(フォーム)	操作の実行方法を記述する フォームハイバーメディア制 御のセット。フォームは、ブロトコルバインディングのシリアライズである。本バーションのTDでは、モノレベルで記述することができるすべての動作は、モノのブロバティと一度にひとまとめに対話する方法に関するものである。	オブション	<u>Form</u> (フォーム) の <u>配列</u>

security(セキュリティ)	セキュリティ定義名のセット。セキュリティ定義で定義で定義されている名前から選択される。これら定義名は全て、リッスヘアクセスするために満足されていなければならない。	必須	string(ストリング)また はstring (ストリング)の配 列
securityDefinitions (セキュリティ定義)	名前付きセキュリティ構成セット(定義のみ)。security名前一値ペアで使用しない限り、実際には適用されない。	必須	SecurityScheme (セキュリ ティスキーム) のマップ

@context 名前-値ペアは、anyURI型の場合は直接的に、あるいは配列型の場合は最初の要素として、anyURI https://www.w3.org/2019/wot/td/v1を含んでいなければならない。@contextが配列である場合、anyURI https://www.w3.org/2019/wot/td/v1を含んでいなければならない。@contextが配列である場合、anyURI https://www.w3.org/2019/wot/td/v1の後に、任意の順序でanyURI型またはマップ型の要素が続く可能性もあるが、@context 配列内のすべての名前-値ペアを持つMapは一個のみとすることが推要される。値が、anyURI型の名前空間識別子であり、名前が、その名前空間を表す用語またはプレフィックスである場合、@context配列に含まれるマップに名前-値ペアが入っている場合もある。名前が用語@languageであり、値が[BCP47]で定義されている整形式言語タグ(例えば、en、de-AT、gsw-CH、zh-Hans、zh-Hant-HK、sl-neids)である場合、@context配列に含まれる一個のマップは、TDのデフォルト言語を定義する名前-値ペアを含んでいるべきである。

すべての人間が読み取り可能なテキストストリングの基本方向の計算は、以下の一連の規則によって定義される。

- 言語タグが与えられない場合、基本方向は、CLDR可能性サブタグ[LDML]のような最も強力なヒューリスティックまたは検出アルゴリズムによって推測されるべきである。
- MultiLanguage(複数言語)マップ以外では、基本方向はデフォルト言語の言語タグから推測できるかもしれない。
- MultiLanguage(複数言語)マップの内部では、名前-値ペアの各値の基本方向は、対応する名前で与えられた言語タグから推測されるかもしれない。
- ある言語を異なる基本方向を持つ複数のスクリプトで書くことができる場合、@languageまたはMultiLanguage(複数言語)マップで与えられる対応する言語タグは、適切な基本方向が推論できるように、スクリプトサブタグを含んでいなければならない。例としてAzeriがあり、これは、ラテン語スクリプトを使用する場合はLTR(az-Latnで指定)、アラビア語スクリプトを使用する場合はRTL(az-Arabで指定)と書かれている。

TDプロセッサは、双方向テキストを処理する際に特殊なケースに気づくべきである。TDプロセッサは、ユーザにストリングを提示するとき、特に周囲のテキストに埋め込むとき(例えば、ウェブユーザインターフェース用)、注意して双方向分離を使用するようにすべきである。混合方向テキストは、言語が適切に識別された場合でも、全言語で起こりうる。

TDプロデューサは、単純なユーザエージェントによってうまく表示できるような方法で混合方向ストリングを提供するよう試みるべきである。例えば、RTLストリングがLTRラン(ラテン語スクリプトにおける数字あるいはブランドあるい商用名など)で始まる場合、ストリングの始めにRLM文字を入れるか、または、双方向制御における逆方向ランをラップすれば、適切な表示を支援することができる。

ウェブ上のストリング:言語と方向のメタデータ[string-meta] は、何らかのガイダンスを提供し、双方向テキストを使用する場合の隠れた危険を多数例証している。

properties, actions, events(プロパティ、アクション、および、イベント)配列で明示的に提供される<u>対話アフォーダンス</u>に加えて、<u>モノ</u>は、そのオプションのforms配列でFormインスタンスによって 示されるメタ対話も提供することができる。<u>モノ</u>インスタンスのforms<u>配列</u>がFormインスタンスを含んでいる場合、その名前opに直接または配列内で割り当てられたストリング値は、 readallproperties、writeallproperties、readmultipleproperties、writemultipしepropertiesという操作タイプのうちのいずれかでなければならない。(モノインスタンスのフォーム<u>使用例</u>を参照。)

これらのメタ対話のそれぞれのデータスキーマは、単一のObjectSchemaインスタンス内の各PropertyAffordanceインスタンスのデータスキーマを組み合わせることによって構築され、ObjectSchemaインスタンスのpropertiesマップは、対応するPropertyAffordanceインスタンスの名前によって識別されるPropertyAffordanceの各データスキーマを含んでいる。

特に指定がない限り(例えば、<u>TDコンテキスト拡張子</u>を介して)、readmultipleproperties操作の要求データは、期待されるPropertyAffordanceインスタンス名を含む<u>配列</u>となり、これは、Formインスタンスによって指定されたコンテンツタイプにシリアライズされている。

5.3.1.2 InteractionAffordance

<u>コンシューマ</u>に可能な選択肢を示し、<u>コンシューマ</u>がそのモノとどのように対話することができるかを示唆するモノのメタデータ。潜在的なアフォーダンスには多くの型があるが、W3C WoTでは、 3つの対話アフォーダンス、すなわち、プロパティ、アクション、および、イベントを定義する。

Vocabulary term	Description	Assignment	Туре
@type	ISON-LD keyword to label the object with semantic tags (or types).	optional	string or Array of string
title	Provides a human-readable title (e.g., display a text for UI representation) based on a default language.	optional	string
titles	Provides multi-language human-readable titles (e.g., display a text for UI representation in different languages).	optional	MultiLanguage
description	Provides additional (human-readable) information based on a default language.	optional	string
descriptions (Can be used to support (human-readable) information in different languages.	optional	MultiLanguage
	Set of form hypermedia controls that describe how an operation can be performed. Forms are serializations of Protocol Bindings.	mandatory	Array of Form
uriVariables	Define URI template variables as collection based on DataSchema declarations.	optional	Map of DataSchema

ボキャブラリ用語	説明	割当	型
@type (タイプ)	JSON-LD キーワード。オブ ジェクトに意味タグ(またはタ イプ) をラベル付けする。	オプション	string(ストリング) また はstring(ストリングス)の配列
title(タイトル)	デフォルト言語に基づいて、人間が読み取り可能なタイトルを 提供する(たとえば、UI 表現の テキストを表示)。		string (ストリング)
titles(タイトル)	複数言語の人間が読み取り可能なタイトルを提供する(例えば、異なる言語でのUI表現のためのテキスト表示)。	オプション	MultiLanguage(複数言語)
description(説明)	デフォルト言語で追加の(人間 が読み取り可能な) 情報を提供 する。	オプション	string(ストリング)
descriptions(説明)	さまざまな言語で(人間が読み取り可能な)情報をサポートするために使用できる。	オプション	MultiLanguage (複数言語)
forms (フォーム)	操作の実行方法を記述する フォームハイパーメディア制 個のセット。フォームは、ブ ロトコルバインディングのシ リアライズである。	必須	<u>Form(</u> フォーム)の <u>配列</u>
uriVariables	URI テンプレート変数を、 データスキーマ宣言に基づい てコレクションとして定義す る。	オブション	DataSchema (データスキーマ) の <u>マップ</u>

クラス対話アフォーダンスには、以下のサブクラスがある。

- PropertyAffordance
- ActionAffordance
- EventAffordance

5 3 1 3 Property Affordance

モノの状態を公開する対話アフォーダンス。この状態は検索(読み取り)でき、随意に更新(書き込み)することができる。モノは、変更後に新しい状態をブッシュすることによって、ブロバティを観察 可能にすることを選択することもできる。

Vocabulary
termDescriptionAssignment Type

optional

boolean

observable Proper

A hint that indicates whether Servients hosting the Thing and Intermediaries should provide a Protocol Binding that supports the observeproperty operation for this Property.

ボキャブラリ用語説明割当型Observable(観察可能)モノと仲介者を提供するサーかのolean(ブール)

ピアントが、このプロパティ のobserveproperty(オブザーブ プロパティ) 操作をサポートす るプロトコルバインディング を提供すべきかどうかを示す ヒント。

ヒント。

ote

Property instances are also instances of the class <u>DataSchema</u>. Therefore, it can contain the type, unit, readOnly and <u>writeOnly</u> members, among others.

馆

プロパティインスタンスは、クラスの \underline{r} ータスキーマのインスタンスでもある。

したがって、type、unit、readOnly、writeOnlyメンバーも含むことができる。

PropertyAffordanceは、InteractiveAffordanceクラスおよびDataSchemaクラスのサブクラスである。FormインスタンスがPropertyAffordanceインスタンス内にある場合、opに割り当てられる値は、readproperty、writeproperty、observeproperty、unobserveproperty、または、これらの組み合わせを含む配列のうちの1つでなければならない。

5.3.1.4 ActionAffordance

状態を操作する(例えば、ランプをオンまたはオフする)、または、モノ上のプロセスをトリガする(例えば、次第にランプの明るさを落とす)モノの関数の呼び出しを可能にする対話アフォーダンス。

Vocabulary term	Description	Assignment	т Туре
input	Used to define the input data schema of the Action.	optional	DataSchema
output	Used to define the output data schema of the Action.	optional	DataSchema
safe	Signals if the Action is safe (=true) or not. Used to signal if there is no internal state (cf. resource state) is changed when invoking an Action. In that case responses can be cached as example.	with default	boolean
idempotent	Indicates whether the Action is idempotent (=true) or not. Informs whether the Action can be called repeatedly with the same result, if present, based on the same input.	with default	boolean

ボキャブラリ用語	説明	割当	型
input(入力)	アクションの入力データス キーマを定義するために使 用。	オプション	DataSchema (データスキーマ)
output(出力)	アクションの出力データス キーマを定義するために使 用。	オプション	DataSchema _(データスキーマ)_
		<u>デフォルトあり</u>	

ActionAffordanceは、InteractioniAffordance<u>クラス</u>の<u>サブクラス</u>である。フォームインスタンスがこのActionAffordanceインスタンス内にあるとき、opに割り当てられた値は、invokeaction(アクション呼び出し)で**なければならない。**

5.3.1.5 EventAffordance

イベントソースを記述する対話アフォーダンスであり、非同期的にイベント・データを<u>消費者</u>にプッシュする(例えば、オーバーヒート警報)。

Vocabulary term	Description	Assignment	Type
subscription	Defines data that needs to be passed upon subscription, e.g., filters or message format for setting up Webhooks.	optional	DataSchema
data	Defines the data schema of the Event instance messages pushed by the Thing.	optional	<u>DataSchema</u>
cancellation	Defines any data that needs to be passed to cancel a subscription, e.g., a specific message to remove a	optional	<u>DataSchema</u>

<u>ボキャブラリ用語</u>	説明	割当	型
subscription(サブスクリプション)	サブスクリプション時に渡す 必要があるデータを定義する (ウェブフックを設定するため のフィルタあるいはメッセー ジフォ-マットなど)。	オプション	<u>DataSchema</u> <u>(データスキーマ)</u>
dara(データ)	モノによってプッシュされる イベントインスタンスメッ セージのデータスキーマを定 義する。	オブション	DataSchema (データスキーマ)
cancellation(キャンセル)	サブスクリプションをキャン セルするために渡す必要があ るデータを定義する(ウェブ フックを削除するための特定 のメッセージなど)。	デフォルトあり	DataSchema (データスキーマ)

EventAffordanceは、InteractionAffordance<u>クラス</u>のサ<u>ブクラス</u>である。FormインスタンスがEventAffordanceインスタンス内にあるとき、opに割り当てられた値は、<u>配列</u>内のsubscribeevent、unsubscribeevent、または、両方のうちのいずれかでなければならない。

5.3.1.6 VersionInfo

Vocabulary term

TDドキュメントのバージョン情報を提供するモノのメタデータ。必要に応じて、ファームウェアおよびハードウェアバージョン(TD名前空間外における用語定義)などの追加バージョン情報は、<u>TDコンテキスト拡張子</u>メカニズムを介して拡張することができる。

instance	Provides a version indi instance.	cator of this TD	mandatory	string	
ボキャブラリ用	語	説明	割当	á	型
Instance(インス	.タンス)	このTD インスタンスの ジョンID を提供する。		<u> </u>	string(ストリング)

Assignment Type

VersionInfo<u>クラス</u>のインスタンス内の値は、意味論的バージョニングパターンに従うことが推奨される。ここで、ドットで区切られた3 つの数字のシーケンスは、メジャーバージョン、マイナーバージョン、パッチバージョンをそれぞれ示す。詳細は[<u>SemVer</u>]を参照。

5.3.1.7 MultiLanguage

<u>[BCP47]</u>に記述されている言語タグによって識別される異なる言語で人間が読み取り可能なテキストを提供する<u>マップ</u>。例えば、TDインスタンス内のコンテナの使用法に関しては、<u>第6.3.2項人間が</u> 読み取り可能なメタデータ参照。

MultiLanguage $\overline{{\color{blue} \mathtt{vul}}}$ の各名前は、 $[\underline{\mathtt{BCP47}}]$ で定義されているような言語タグでなければならない。

Description

5.3.2 データ・スキーマ・ボキャブラリ定義 『

データスキーマ定義は、JSON スキーマ[JSON-SCHEMA] によって定義された用語の非常に一般的なサブセットを反映している。TDインスタンス内のデータスキーマ定義は、この定義されたサブセットに限定されておらず、<u>第7項 TDコンテキスト拡張子</u>で解説されている追加用語のための<u>TDコンテキスト拡張子</u>を使用しているJSONスキーマで見られる追加の用語を使用してもよいことに留意されたい。あるいは、これら用語は、<u>TDプロセッサ</u>に意味的に無視される。(意味処理の詳細は、付録D.JSON-LDコンテキスト用法と名前空間IRI内の資料(例:https://www.w3.org/2019/wot/td)参照。)

5.3.2.1 DataSchema

使用されるデータフォーマットを記述するメタデータ。確認のために使用することができる。

Vocabulary term	Description	Assignmer	nt Type
@type	JSON-LD keyword to label the object with semantic tags (or types)	optional	string or Array of string
title	Provides a human-readable title (e.g., display a text for UI representation) based on a default language.	optional	string
titles	Provides multi-language human-readable titles (e.g., display a text for UI representation in different languages).	optional	MultiLanguage
description	Provides additional (human-readable) information based on a default language.	optional	string
descriptions	Can be used to support (human-readable) information in different languages.	optional	MultiLanguage
type	Assignment of JSON-based data types compatible with JSON Schema (one of boolean, integer, number, string, object, array, or null).	optional	string (one of object, array, string, number, integer, boolean, or null)
const	Provides a constant value.	optional	any type

readOnly

writeOnly

Provides unit information that is used, e.g., in international science, engineering, and business. Used to ensure that the data is valid against one of the executed in the array.

Restricted set of values provided as an array.

Boolean value that is a hint to indicate whether a property interaction / value is read only (=true) or not (=false). Boolean value that is a hint to indicate whether a property interaction / value is write only (=true) or not (=false).

optional Array of any type
with default
with default
optional string

optional string Array of DataSchema Type

Allows validation based on a format pattern such as "date-time", "email", "uri", etc. (Also see below.)

ボキャブラリ用語	説明	割当	型
@type (タイプ)	オブジェクトに意味論タグ(または型)をラベル付けするため のJSON-LD キーワード。	オブション	<u>string</u> (ストリング) また は <u>string</u> (ストリング)の <u>配列</u>
title(タイトル)	デフォルト言語に基づいて、人間が読み取り可能なタイトルを 提供する(たとえば、UI 表現の テキストを表示)。		string(ストリング)
titles(タイトル)	複数言語の人間が読み取り可能なタイトルを提供する(例えば、異なる言語でのUI表現のためのテキスト表示)。	オプション	MultiLanguage(複数言語)
description(説明)	デフォルト言語で追加の(人間 が読み取り可能な) 情報を提供 する。	オプション	string(ストリング)
descriptions(説明)	さまざまな言語で(人間が読み 取り可能な)情報をサポートす るために使用できる。	オプション	MultiLanguage (複数言語)
type(型)	操作の実行方法を記述する フォームハイバーメディア制 御のセット。フォームは、ブ ロトコルバインディングのシ リアライズである。	オプション	string (ストリング) (object. array, string, number, integer, boolean, null のいずれか)
Const	URI テンプレート変数を、 データスキーマ宣言に基づい てコレクションとして定義す る。	オプション	任意の型
unit(単位)	国際科学、エンジニアリン グ、ビジネスなどで使用され る単位情報を提供する。	オプション	string (ストリング)
one of (の一つ)	配列内の指定された一つのスキーマに対してデータが有効であることを保証するために使用される。	オブション	<u>DataSchema</u> (データスキーマ) の <u>配列</u>

enum	配列として提供される制限さ れた値のセット。	オプション	任意の型の <u>配列</u>
readOnly(読取り専用)	プロパティ対話/値が読み取り 専用(=true) かそうではないか (=false) を示すヒントとなる ブール値。	<u>デフォルトあり</u>	<u>b</u> oolean(ブール)
writeOnly(書き込み専用)	プロパティ対話/値が書き込み 専用(=true) かそうではないか (=false)を示すヒントである ブール値。	<u>デフォルトあり</u>	<u>b</u> oolean(プール)

クラスDataSchema には、以下のサブクラスがある。

ArraySchema

BooleanSchema

NumberSchema

IntegerSchema

ObjectSchema

StringSchema

NullSchema

フォーマットストリング値は、[JSON-SCHEMA] (特に第7.3項定義フォーマット) で定義されている固定値セットとそれに対応するフォーマットルールから判別できる。サービアントは、format値を 使用して、それに応じて追加の確認を実行.**しても良い**。既知の値セットで見つからない値がformatに割り当てられる場合、そのような確認は成功**すべきである**。

Assignment Type

double(ダブル)

配列型のデータを記述するメタデータ。このサブクラスは、DataSchemaインスタンス内のtypeに割り当てられた値のarrayによって示される。

Vocabulary term	Description	Assignmen	t Type
items	Used to define the characteristics of an arra	y. optional	<u>DataSchema</u> or <u>Array</u> of <u>DataSchema</u>
minItems	Defines the minimum number of items that harray.	nave to be in the optional	unsignedInt
maxItems	Defines the maximum number of items that array.	have to be in the optional	unsignedInt
ボキャブラリ用		割当	型

<u>ボキャブラリ用語</u>	説明	割当	型
Items(項目)	配列の特性を定義するために 使用される。	オプションの配列	<u>DataSchema</u> また は <u>DataSchema</u> の配列
minItems(最小項目)	配列に含めなければならない項 目の最小数を定義する。	オプション	unsignedInt
maxItems(最大項目)	配列に含める必要がある項目 の最大数を定義する。	オプション	unsignedInt

5.3.2.3 BooleanSchema

boolean 型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchemaインスタンス内のtypeに割り当てられた値booleanによって示される。

Vocabulary term

number型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchemaインスタンス内のtypeに割り当てられた値numberによって示される。

maximum	'	, , , ,	or associated number or integer ty	optional	double
ボキャブラリ用	五品	説明	割当	型	
minimum(最小)		最小数値を指定する。関連付けられた数値型または整数型のよ に適用される。		double(ダブル)	

Description

オプション maximum(最大) 最小数値を指定する。関連付 けられた数値型または整数型

のみに適用される。

5.3.2.5 IntegerSchema

integer型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchema インスタンスでtype に割り当てられた値integer によって示される。

ボキャブラリ用語	説明	割当	型
minimum(最小)	最小数値を指定する。関連付け られた数値型または整数型のみ に適用される。		integer(整数)
maximum(最大)	最小数値を指定する。関連付けられた数値型または整数型のみ に適用される。		integer(整数)

object型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchemaインスタンス内のtypeに割り当てられた値objectによって示される。

Vocabulary term	Description	Assignment	Туре
properties	Data schema nested definitions.	optional	Map of DataSchema
required	Defines which members of the object type are mandatory.	optional	Array of string

required	mandatory.		ориона	7 tiray or sting	
ボキャブラリ用	蓋	説明	割当		型
properties(プロル	パティ)	データスキーマのネストされた 定義。	オプション		DataSchemaのマップ
required(要求さ	れる)	オブジェクト型のどのメンバー が必須であるかを定義する。	オプション		<u>string(</u> ストリング)の <u>配列</u>

string型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchemaインスタンス内のtypeに割り当てられた値stringによって示される。

5.3.2.8 NullSchema 5

null型のデータを記述するメタデータ。この<u>サブクラス</u>は、DataSchemaインスタンス内のtypeに割り当てられた値nullによって示される。このサブクラスは、1つの許容可能な値、すなわち、nullの みを記述する。これは、情報として使われる場合、データもnullになれるというoneOf宣言の一部として使用することができる。

5.3.3 セキュリティボキャブラリ定義』

本仕様は、W3C WoTの<u>プロトコルパインディング</u>として適格なプロトコルに直接組み込まれるか、または、これらのプロトコルと組み合わて広く使用されている十分に確立されたセキュリティメカ ニズムから選ばれたメカニズムを提供する。現在のHTTPセキュリティスキームセットは、部分的に<u>OpenAPI 3.0.1</u>に基づいている(<u>[OPENAPI]</u>も参照)。しかしながら本仕様で与えられたHTTPセキュ リティスキーム、<u>ボキャブラリ</u>、構文は、OpenAPIと多くの類似点があるが、それらは互換性を持たない。

5 3 3 1 SecurityScheme

セキュリティメカニズムの構成を記述するメタデータ。名前schemeに割り当てられた値は、第<u>5項 TD情報モデル</u>で定義される標準<u>ボキャブラリ</u>、あるいは、 $\overline{\text{LD}}$ コンテキスト拡張子のいずれかで、 $\overline{\text{LD}}$ に含まれる<u>ボキャブラリ</u>の中で定義され**なければならない。**

Vocabulary term	Description	Assignmer	nt Type
@type	JSON-LD keyword to label the object with semantic tags (or types).	optional	string or Array of string
scheme	Identification of the security mechanism being configured.	mandatory	<u>string</u> (e.g., nosec, basic, cert, digest, bearer, pop, psk, public, oauth2, or apikey)
description	Provides additional (human-readable) information based on a default language.	optional	string
descriptions	Can be used to support (human-readable) information in different languages.	optional	MultiLanguage
proxy	URI of the proxy server this security configuration provides access to. If not given, the corresponding security configuration is for the endpoint	optional	<u>anyURI</u>

ボキャブラリ用語	説明	割当	型
@type (@型)	オブジェクトに意味タグ (またはタイプ) をラベル付けするためのJSON-LD キーワード。	オプション	string(ストリング) またはstring(ストリン グ)の配列
s chema(スキーマ)	構成されているセキュリ ティメカニズムの識別。	オプション	string(ストリング) (nosec, basic,cert,digest,bearer,pop,psk,public,oauth2,or apikey)
description(説明)	デフォルト言語に基づい て、追加の(人間が読み取 り可能な)情報を提供す る。	オプション	string (ストリング)
descriptions(説明)	さまざまな言語の(人間が 読み取り可能な)情報をサポートするために使用で きる。	オプション	MultiLanguage(複数言語)
proxy(プロキシ)	このセキュリティ構成が アクセスを提供するプロ キシーサーバのURI。 指定 されていない場合、対応 するセキュリティ構成は エンドポイント用であ る。	オブション	<u>anyURI</u>

クラスSecuritySchemeは、以下のサブクラスを有する。

- NoSecurityScheme
- BasicSecurityScheme
- DigestSecurityScheme
 APIKeySecurityScheme
- BearerSecurityScheme
- CertSecurityScheme
- PSKSecurityScheme
- PublicSecurityScheme
- PoPSecurityScheme
 OAuth2SecurityScheme
- NoSecurityScheme

BasicSecurityScheme

DigestSecurityScheme

APIKeySecurityScheme

BearerSecurityScheme

<u>CertSecurityScheme</u>

PSKSecurityScheme

<u>PublicSecurityScheme</u>

<u>PoPSecurityScheme</u>

OAuth2SecurityScheme

5.3.3.2 NoSecurityScheme

リソースにアクセスするために必要とされる認証または他のメカニズムがないことを示す<u>ポキャブラリ用語</u>によって識別されるnosec (すなわち、「scheme」:「nosec」)に対応するセキュリティ構

5.3.3.3 BasicSecurityScheme

暗号化されていないユーザ名およびパスワードを使用する<u>ボキャブラリ用語</u>basic(すなわち、「scheme」:「basic」)によって識別される基本認証[<u>RFC7617</u>]セキュリティ構成。本スキームは、例えば、TLSのように機密性を提供する他の何らかのセキュリティメカニズムと共に使用されるべきである。

Vocabulary tern		Assignmen	
in	Specifies the location of security authentication information.	with default	string (one of header, query, body, or cookie)
name	Name for query, header, or cookie parameters.	optional	string

ボキャブラリ用語	説明	割当	型
In(内)		<u>デフォルトあり</u>	string(ストリング)
	セキュリティ認証情報の場所を 指定する。	!	(header,query,body,cookieのい ずれか)
name(名前)	クエリ、ヘッダー、あるいは、 クッキーのパラメータ名。	オプション	string(ストリング)

5.3.3.4 DigestSecurityScheme

<u>ボキャブラリ用語</u>digest (すなわち、「scheme」:「digest」)によって識別されるダイジェストアクセス認証[<u>RFC7616</u>]セキュリティ構成。本スキームは、基本認証に似ているが、中間者攻撃を 回避する機能が追加されている。

Vocabulary term	Description	Assignment	Туре
qop	Quality of protection.	with default	string (one of auth, or auth-int)
in	Specifies the location of security authentication information.		$\underline{\text{string}}$ (one of header, query, body, or cookie)
name	Name for query, header, or cookie parameters.	optional	string

ボキャブラリ用語	説明	割当	型
qop	保護の質。	<u>デフォルトあり</u>	<u>string(ストリング)</u> (authあ るいは authintのいずれか)
in (内)		<u>デフォルトあり</u>	string(ストリング)
	セキュリティ認証情報の場所を 指定する。	Ė	(header,query,body,cookieのい ずれか)
name(名前)	クエリ、ヘッダー、あるいは、 クッキーのパラメータ名。	オプション	string(ストリング)

5.3.3.5 APIKeySecurityScheme

Vocabulary term

<u>ボキャプラリ用語</u>apikey (すなわち、「scheme」:「apikey」)によって識別されるAPIキー認証セキュリティ構成。これは、アクセストークンが不透明であり、標準トークンフォーマットを使用して いない場合に用いる。

Type

Assignment

Assignment

optional <u>anyURI</u>

in	Specifies the location of security authentication information.	with default	string (one of header, query, body, or cookie)
name	Name for query, header, or cookie parameters.	optional	string
ボキャブラリ用	證 説明	割当	型
in (内)	セキュリティ認証情報の場所を	<u>デフォルトあり</u>	string (ストリング) (header,guery,body,cookieのい
	指定する。		(neader,query,body,cookleので ずれか)
name(名前)	クエリ、ヘッダー、あるいは、 クッキーのパラメータ名。	オプション	string(ストリング)

Description

Description

Specifies format of security authentication information.

URI of the authorization server.

Encoding, encryption, or digest algorithm.

5.3.3.6 BearerSecurityScheme

Vocabulary term

authorization

ala

format

ベアラトークンがOAuth2とは別個に使用される状況における<mark>ボキャブラリ用語</mark>bearer(すなわち、「scheme」:「bearer」)によって識別されるベアラトークン[<u>RFC6750</u>]セキュリティ構成。本スキームは、ベアラトークンがOAuth2とは独立して使用される状況のためのものである。oauth2 スキームが指定されている場合は、通常、本スキームを暗示・指定する必要はない。Formatに関しては、値jwtは[<u>RFC7519</u>]への適合性を示し、jwsは[<u>RFC7797</u>]への適合性を示し、cwtは[<u>RFC8392</u>]への適合性を示し、jweは[<u>RFC7516</u>]への適合性を示し、algの値はこれらの標準と整合して解釈される。ベアラトークンのための他のフォーマットおよびアルゴリズムは、ボキャブラリ拡張において指定**もできる。**

with default string (e.g., MD5, ES256, or ES512-256)

with default string (e.g., jwt, cwt, jwe, or jws)

Type

in	Specifies the location o information.	f security authentication	with default	string (one of head cookie)	er, query, body, Or
name	Name for query, heade	r, or cookie parameters.	optional	string	
ボキャブラリ用	語	説明	割当	1	Ҵ
authorization(計	可)	許可サーバのURI。	オプション	<u> </u>	anyURI
alg(アルゴリズ.	۲)	エンコーディング、暗号化、ま たはダイジェストアルゴリズ ム。	<u>デフォルトあり</u>	(string(ストリング) MD5,ES256,ES512-256)
format(フォーマ	ツト)	セキュリティ認証情報のフォー マットを指定する。	<u>デフォルトあり</u>		string(ストリング) jwt,cwt,jwe,jws)
in (内)		セキュリティ認証情報の場所を 指定する。	<u>デフォルトあり</u>	(string(ストリング) header,query,body,cookieのい ずれか)
name(名前)		クエリ、ヘッダー、あるいい は、クッキーのパラメータ名。	オプション	Š	string(ストリング <u>)</u>

5.3.3.7 CertSecurityScheme

本項は危険な状態にある。

<u>ボキャブラリ用語</u>cert (すなわち、「scheme」:「cert」)によって識別される[<u>X509V3</u>]に準拠する証明書ベースの非対称キーセキュリティ構成。

Vocabulary term		Description		Assignment	Туре
	Identifier providing infor confirmation.	mation which can be used for se	election or	optional	string.
ボキャブラリ用	语	説明	割当	型	
identity (ID)		選択または確認に使用できる情報を提供する識別子。	オプション 情	str	ing(<u>ストリング)</u>

5.3.3.8 PSKSecurityScheme \$

<u>ボキャブラリ用語</u>psk (すなわち、「scheme」:「psk」)によって識別される事前共有キー認証セキュリティ構成。

Vocabulary term	Description		Assignment Type
identity	Identifier providing information which car confirmation.	n be used for selection or	optional <u>string</u>
ボキャブラリ用	語説明	割当	型
identity (ID)	選択または確認		string(ストリング)

5.3.3.9 PublicSecurityScheme

This section is at risk.

本項は危険な状態にある。

<u>ボキャブラリ用語</u>public (すなわち、「scheme」:「public」)によって識別される生の公開キー非対称鍵セキュリティ構成。

Vocabulary term	Description	Assignment	Type
identity	Identifier providing information which can be used for selection or confirmation.	optional	string

ボキャブラリ用語	説明	割当	型
identity (ID)	選択または確認に使用できる情報を提供する識別子。	オプション	string(ストリング)

Description

5.3.3.10 PoPSecurityScheme §

This section is at risk.

Vocabulary term

本項は危険な状態にある。

<u>ボキャブラリ用語</u>pop(すなわち、「scheme」:「pop」)によって識別されるPoP(Proof-of-possession)トークン認証セキュリティ構成。jwtは<u>[RFC7519]</u>との適合性を示し、jwsは<u>[RFC7797]</u>との適合性を示し、cwtは<u>[RFC8392]</u>との適合性を示し、jweは<u>[RFC7516]</u>との適合性を示し、algの値はこれらの標準と整合して解釈される。PoPトークンのための他のフォーマットおよびアルゴリズムは、ボキャブラリ拡張において指定**もできる。**

Type

Assignment

rocusaia, j term		2 000pt.o	7.00.g	,,,,,
authorization	URI of the authorization	server.	optional	anyURI
alg	Encoding, encryption, o	r digest algorithm.	with default	string (e.g., MD5, ES256, or ES512-256)
ormat	Specifies format of secu	urity authentication information.	with default	string (e.g., jwt, cwt, jwe, or jws)
n	Specifies the location o information.	f security authentication	with default	string (one of header, query, body, or cookie)
name	Name for query, header	r, or cookie parameters.	optional	string
ボキャブラリ用	ă.	説明	割当	型
authorization(許	可)	許可サーバのURI。	オプション	<u>anyURI</u>
alg(アルゴリズム	()	エンコーディング、暗号化、ま	<u>デフォルトあり</u>	string(ストリング)
		たはダイジェストアルゴリズ ム。		(MD5,ES256,ES512-256)
format(フォーマ	ット)	セキュリティ認証情報のフォー	<u>デフォルトあり</u>	string(ストリング)
		マットを指定する。		(jwt,cwt,jwe,jws)
in (内)			<u>デフォルトあり</u>	string(ストリング)
		セキュリティ認証情報の場所を 指定する。		(header,query,body,cookieの ずれか)
name(名前)		クエリ、ヘッダー、あるいい	オプション	string(ストリング)
		は、クッキーのパラメータ名。		

5.3.3.11 OAuth2SecurityScheme

本項は危険な状態にある。[<=@@@左の文,余分では?]

OAuth2 authentication security configuration for systems conformant with [RFC6749] and [RFC8252], identified by the Vocabulary Term oauth2 (i.e., "scheme": "oauth2"). For the implicit flow authorization MUST be included. For the password and client flows token MUST be included.

<u>ボキャブラリ用語</u>oauth2 (すなわち、「scheme」: 「oauth2」) によって識別される [<u>RFC6749</u>] および [<u>RFC8252</u>] に準拠するシステムのOAuth2認証セキュリティ構成。implicitフローについては、authorizationが含まれなければならない。passwordとclientフローについて、tokenが含まれなければならない。codeフローについては、authorizationとtokenの両方が含まれなければならない。SecuritySchemeにscopeが定義されていない場合、それらは空であるとみなされる。

Vocabulary term	Description	Assignment	t Type
authorization	URI of the authorization server.	optional	anyURI
token	URI of the token server.	optional	anyURI
refresh	URI of the refresh server.	optional	anyURI
scopes	Set of authorization scope identifiers provided as an array. These are provided in tokens returned by an authorization server and associated with forms in order to identify what resources a client may access and how. The values associated with a form should be chosen from those defined in an OAuth2SecurityScheme active on that form.	optional	string or Array of string
flow	Authorization flow.	mandatory	string (one of implicit, password, client, Or code)

ボキャブラリ用語	説明	割当	型
authorization(許可)	許可サーバのURI。	オプション	<u>anyURI</u>
token(トークン)	トークンサーバのURI.	オプション	<u>anyURI</u>
refresh(リフレッシュ	リフレッシュサーバのURI	オプション	<u>anyURI</u>
scopes(範囲)	配列として提供される許可範ィ ID セット。これらは、クラ・ アントがどのリソースにどの うにアクセスできるかによう をかるこれったに関連付けれたトークンとして提供され る。フォームに関連付けられ いる値は、そのフォーム上 クティブな OAuth2SecurityScheme で定 される値の中から選択する必 がある。	よす すてら てア 義	string(ストリング)また はstring(ストリング)の配列
flow(フロー)	許可フロー	必須	<u>string(ストリング)</u> (implicit,password,client,codeの いずれか)

映している。定義された用語は、例えば、Switchというモノによって制御されるLampというものなどの別の<u>モノ</u>との関係を記述するために便用することができる。Formクラスは、<u>モノ(</u>及び他の ウェブリソース)の状態を操作するために新たに導入されたハイパーメディア制御に対応する。

5.3.4.1 Link (リンク) 🐇

リンクは、「リンクコンテキストは、リンクターゲットにリレーション型リソースを有する」という形式のステートメントとみなすことができ、ここでオプションのターゲット属性によって、リ ソースをさらに記述することができる。

Vocabulary term	Description	Assignmen	t Type
href	Target IRI of a link or submission target of a form.	mandatory	anyURI
type	Target attribute providing a hint indicating what the media type [RFC2046] of the result of dereferencing the link should be.	optional	string
rel	A link relation type identifies the semantics of a link.	optional	string
anchor	Overrides the link context (by default the Thing itself identified by its id) with the given URI or IRI.	optional	anyURI

ボキャブラリ用語	説明	割当	型
href	リンクのターゲットIRI、また は、フォームの送信ターゲッ ト。	必須	anyURI
type(型)	リンクをデリフェレンスした結果のメディアタイプ[<u>RFC2046</u>] が何であるべきかを示すヒント を提供するターゲット属性。		string(ストリング)
rel	リンクリレーション型はリンク の意味を識別する。	オプション	string(ストリング)
anchor(アンカー)	所与のURIあるいはIRIを使って、リンクコンテキスト(ラフォールトで、それ自体がそのidにより識別されるモノ)を無効にする。	ŕ	anyURI

5.3.4.2 Form 5

フォームは、「フォームコンテキスト上でオペレーションタイプ操作を実行するために、送信ターゲットへ要求メソッドを要求する」というステートメントとみなすことができ、オプションの フォームフィールドは、必要な要求をさらに記述することができる。TDでは、フォームコンテキストは、Properties、Actions、およびEventsなどの周囲のオプジェクト、または、メタ対話のための モノ自体となる。

Vocabulary term	Description	Assignmen	туре Туре
	Indicates the semantic intention of performing the operation(s) described by the form.		
op	For example, the Property interaction allows get and set operations. The protocol binding may contain a form for the get operation and a different form for the set operation. The op attribute indicates which form is for which and allows the client to select the correct form for the operation required.	with default	string or Array of string (one of readproperty, writeproperty, observeproperty, unobserveproperty, invokeaction, subscribeevent, unsubscribeevent, readallproperties, writeallproperties, readmultipleproperties, or writemultipleproperties)
	op can be assigned one or more interaction verb(s) each representing a semantic intention of an operation.		······································
href	Target IRI of a link or submission target of a form.	mandatory	anyURI
contentType	Assign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type [RFC2046].	with default	string
contentCoding	Content coding values indicate an encoding transformation that has been or can be applied to a representation. Content codings are primarily used to allow a representation to be compressed or otherwise usefully transformed without losing the identity of its underlying media type and without loss of information. Examples of content coding include "gzip", "deflate", etc	optional	string
	Indicates the exact mechanism by which an interaction will be accomplished for a given protocol when there are multiple options.		
subprotocol	For example, for HTTP and Events, it indicates which of several available mechanisms should be used for asynchronous notifications such as long polling (longpoll), WebSub [websub] (websub), Server-Sent Events [eventsource] (sse). Please note that there is no restriction on the subprotocol selection and other mechanisms can also be announced by this subprotocol term.	optional	string (e.g., longpoll, websub, Or sse)
security	Set of security definition names, chosen from those defined in securityDefinitions. These must all be satisfied for access to resources.	optional	string or Array of string
scopes	Set of authorization scope identifiers provided as an array. These are provided in tokens returned by an authorization server and associated with forms in order to identify what resources a client may access and how. The values associated with a form should be chosen from those defined in an OAuth2SecurityScheme active on that form.	optional	string or Array of string
response	This optional term can be used if, e.g., the output communication metadata differ from input metadata (e.g., output contentType differ from the input contentType). The response name contains metadata that is only valid for the response messages.	optional	ExpectedResponse

response messages.			
ボキャブラリ用語	説明	割当	型
ор	フォームで記述された操作を引行する意味・意図を示す。例は、Property対話は、getおは、set動作を可能にする。プロトルパインディングは、フォックはめの別のフォームと、セットオペレーを含むことができる。opaには、ためのアオームがどのフォームがどのフォームがとのフォームがとかを示るかを示るかを示るが要とされる遺形であるとができるようにするとなっています。には複数の対話動詞を割り当てることができる。	えび コ ナ ム た が ー の ト E が 彙 チ	string (ストリング) また はstring (ストリング) の配列 (readproperty、writeproperty、 observeproperty、 unobserveproperty、 invokeaction、 subscribeevent、 unsubscribeevent、 readallproperties、 writeallproperties、 writeallproperties、 writemultiplepropertiesのいず れか)
href	リンクのターゲットIRI、また フォームの送信ターゲット。	は必須	anyURI
contentType (コンテンツタイプ)	メディアタイプ (例: 'text/plain およびメディアタイプ [RFC2046]の潜在パラメータ (例: 'charset=utf-8') に基づい て、コンテンツタイプを割り でる。	,	string(ストリング)
contentCoding(コンテンツコーディング)	、コンテンツコーディング値は、表現に使用されているなりに使用されているなりです。コンテンツコーディングがは、	n 元 プ と な う こ	string(ストリング)

subptotocol(サブプロトコル)	オブション オブション がある場合 に、特定プロトコルに対して対 話が実行されるそのメカニズム を示す。例えば、HTTPおよび イベントの場合、これは、ロン グポーリング(longpoll)、ウェブ サブwebsub、サーバが 送信したイベント[eventsource] (sse)などいくつか利用可能なメカニズムのうちへきかを示す。サ ブプロトコルの選択に制限はな く、他のメカニズムもこのサブ プロトコル用語によってアナウ ンスできることに留意された い。	<u>string(ストリング)</u> (例: longpoll, websub、または sse)
security(セキュリティ)	オプション オプション いる。定義から選択されたセ キュリティ定義名セット。リ ソースへのアクセスのためには これら全てが満たされなければ ならない。	<u>string(ストリング)</u> また は <u>string</u> (ストリング)の <u>配列</u>
scopes(範囲)	オプション 配列として提供される許可範囲 ID セット。これらは、クライア ントがどのリソースにどのよう にアクセスできるかを識別する ために、許可サーバによって返 される、フォームに関連付けら れたトークンとして提供され る。フォームに関連付けられて いる値は、そのフォーム上でア クティブな OAuth2SecurityScheme で定義 される値の中から選択する必要 がある。	<u>string(ストリング)</u> また は <u>string</u> (ストリング)の <u>配列</u>

response(応答) オプション ホオプション田語は、たとえ

本オブション用語は、たとえ ば、以下の場合に使用すること ができる。出力通信メタデータ が入力メタデータとは異なる例 えば、出力コンテンツ型が入力 コンテンツ型とは異なる)。 応答 名には、応答メッセージにのみ 有効なメタデータが含まれる。 ExpectedResponse (期待される応答)

フォームの可能な操作型リストは確定されたものである。本仕様書の本バージョンでは、<u>IWOT- ARCHITECTURE</u>に記述されたWoT対話モデルをインプリメンテーションするために必要な周知の 型のみを含んでいる。本標準の将来バージョンはこのリストを拡張したものになるかもしれない。が、操作型は、サービアントが任意に設定する**べきではない**。

オプションのresponse名前-値ペアは、期待される応答メッセージのメタデータを提供するために使用することができる。中核ボキャブラリでは、そのペアはコンテンツタイプ情報を持つのみであるが、TDコンテキスト拡張子を適用することができる。応答名-値ペアが与えられていない場合、応答のコンテンツタイプは、フォームインスタンスに割り当てられたコンテンツタイプと同じであると仮定しなければならない。ExpectedResponseクラス内のcontentTypeは、デフォルト値を持っていないということに留意されたい。例えば、フォームのコンテンツタイプの値がapplication/xmlである場合、応答のコンテンツタイプの仮定値もapplication/xmlとなる。

いくつかの使用事例では、入出力データは、異なる形式、例えば、JSONを受け入れるが画像を返すアクションで表される事もある。そのような場合、オブションのresponse名前-値ペアは、期待される応答のコンテンツタイプを記述することができる。期待される応答のコンテンツタイプがフォームのコンテンツタイプと異なる場合、フォームインスタンスは、名前responseを持つ名前-値ペアを含んでい**なければならない。**例えば、ActionAffordanceは、その入力データとしてapplication/jsonのみを受け入れることができ、一方、その出力データとしてimage/jpegコンテンツタイプで応答する。その場合、コンテンツタイプは異なり、response名前-値ペアが、応答コンテンツタイプ(image/jpeg)情報を<u>コンシューマ</u>に提供するために使用されなければならない。

contentCodingプロパティとして可能な値は、例えば、 $IANA\ HTTP$ コンテンツコーディングレジストリで見つけることができる。

5.3.4.3 ExpectedResponse §

期待される応答メッセージを記述する通信メタデータ。

Vocabulary term

ContentType

Assign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type

Massign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type (e.g., text/plain) and potential parameters (e.g., te

 ボキャブラリ用語
 説明
 割当
 型

 contentType (コンテンツタイプ)
 必須
 メディアタイプ(例:'text/plain') およびメディアタイプ [RFC2046]の潜在パラメータ (例: 'charset=utf-8') に基づいて、コンテンツタイプを割り当てる。
 マンテンツタイプを割り当てる。

5.4 デフォルト値定義 🖣

TD内の割当が欠落している場合、TDプロセッサは、第5.4項デフォルト値定義内の表に示されているデフォルト値割当に従わなければならない。

下表は、TD情報モデルで定義されているデフォルト値のすべてである。

Class	Vocabulary Term	Defa	ault Value	Comment
Form	contentType	application/json		
DataSchema	readOnly	false		
DataSchema	writeOnly	false		
ActionAffordance	safe	false		
ActionAffordance	idempotent	false		
Form	ор	Array of string with the elen	nents readproperty and	If defined within an instance of PropertyAffordance
Form	op	invokeaction		If defined within an instance of ActionAffordance
Form	ор	subscribeevent		If defined within an instance of EventAffordance
BasicSecurityScheme	in	header		
DigestSecurityScheme	in	header		
BearerSecurityScheme	in	header		
PoPSecurityScheme This feature is at risk.	in	header		
APIKeySecurityScheme	in	query		
DigestSecurityScheme	qop	auth		
BearerSecurityScheme	alg	ES256		
PoPSecurityScheme This feature is at risk.	alg	ES256		
BearerSecurityScheme	format	jwt		
PoPSecurityScheme This feature is at risk.	format	jwt		
		<u>ボキャブラリ</u>	デフォルト値	注釈
<u>クラス</u>		用語		
Form		readOnly	false	

クラス	<u>ボキャブラリ</u>	<u>デフォルト値</u>	注釈
274	用語		
Form	readOnly	false	
DataSchema	writeOnly	false	
DataSchema	safe	false	
ActionAffordance	idempotent	false	
Form	ор	readpropertyおよび writeproperty要素を持つ string(ストリング)の配列。	PropertyAffordanceインスタン ス内で定義されている場合
Form	ор	invokeAction	ActionAffordanceインスタンス 内で定義されている場合
Form	ор	subscribeevent	EventAffordanceインスタンス 内で定義されている場合
BasicSecurityScheme	in	header	
DigestSecurityScheme	in	header	
BearerSecurityScheme	in	header	
PoPSecurityScheme 本機能は危険な状態である。	in	header	
本機能は危険なれたとめる。 APIKeySecurityScheme		query	
	in		

[DigestSecurityScheme	qop	auth
E	3earerSecurityScheme	alg	ES256
F	PoPSecurityScheme	alg	ES256
ā	本機能は危険な状態である。	arg	
E	BearerSecurityScheme	format	jwt
F	PoPSecurityScheme	format	jwt
7	本機能は危険な状態である。	Tomac	

6. TD表現形式 §

WoT Thing Descriptions represent Things and are modeled and structured based on § \$\frac{1}{2}\$. TD Information Model. This section defines a JSON-based representation format for Things, a serialization of instances of the Class Thing defined by the TD Information Model.

WoTTDは、モノを表し、第5項TD情報モデルに基づいてモデル化され、構成されている。本項では、TD情報モデルが定義するクラスThingのインスタンスのシリアライズであるモノのJSONベース

A TD Processor MUST be able to serialize Thing Descriptions into the JSON format [RFC8259] and/or deserialize Thing Descriptions from that format, according to the rules noted in § 6.1 Mapping to JSON Types and § 6.3 Information Model Serialization

 $\underline{\text{TD}}$ プロセッサは、第6.1項 JSONタイプへのマッピングと第6.3項情報モデルシリアライズに記載されている規則に従って、JSONフォーマット[RFC8259]に $\underline{\text{TD}}$ をシリアルライズする、また・あるいは、そのフォーマットから $\underline{\text{TD}}$ をシリアライズ解除することができ**なければならない。**

The JSON serialization of the TD Information Model is aligned with the syntax of JSON-LD 1.1 [json-ld11] in order to streamline semantic evaluation. Hence, the TD representation format can be processed either as raw JSON or with a JSON-LD 1.1 processor (for details about semantic processing, please refer to the documentation under the namespace IRIs, e.g., https://www.w3.org/2019/wot/td).

TD 情報モデルのJSONシリアライズは、意味評価を簡素化するために、JSON-LD 1.1 [sonId11] の構文と整合化されている。従って、TD表現フォーマットは、生のJSONとして、または、第D項RDFへの変換でさらに詳述するように、JSON-LD 1.1プロセッサを用いて処理することができる。意味処理の詳細は、付録D.JSON-LD コンテキスト用法と名前空間IRI内の資料(例: https://www.w3.org/2019/wot/td)参照。

In order to support interoperable internationalization, TDs MUST be serialized according to the requirements defined in Section 8.1 of RFC8259 [RFC8259] for open ecosystems. In summary, this requires the following:

- TDs MUST be encoded using UTF-8 [FC3629].
 Implementations MUST NOT add a byte order mark (U+FEFF) to the beginning of a TD document.
- TD Processors MAY ignore the presence of a byte order mark rather than treating it as an error.

相互運用可能な国際化をサポートするために、TDは、オープンエコシステムのためのRFC8259(RFC8259)の第8項で定義されている要件に従ってシリアライズされなければならない。要約する

- TDはUTF-8 [RFC3629] を使って符号化されなければならない。
- インプリメンテーションは、TDドキュメントの先頭にBOM (byte order mark) (U+FEFF)を追加してはならない。
- TDプロセッサは、BOMをエラーとして扱うのではなく、その存在を無視してもよい。

6.1 JSON タイプへのマッピング §

TD情報モデルは、モデルオブジェクトとJSONタイプ間で容易にマッピングできるように構築される。すべてのクラスはJSONオブジェクトへのマップを例に挙げ、クラスインスタンスの各名前--値 ペアはJSONオブジェクトのメンバーである。

第5.3項クラス定義(すなわち、string、anyURI、dateTime、integer、unsignedInt、double、boolean)で言及されている<u>シンプルタイプ</u>はすべて、下記の規則に従って、基本JSONタイプ(string、number、boolean)へマップする。これらの規則、名前-値ペアに適用される。

- string型またはanyURI型の値は、JSONストリングとしてシリアライズされなければならない。
- dateTime型の値は、[RFC3339]で指定された"日-時"形式でJSONストリングとしてシリライズされなければならない。例には、2019-05-24T13:12:45Z および2015-0711T09:32:26+08:00など がある。dateTime型の値は、オフセットの代わりにUTCタイムゾーンを表すリテラルzを使用すべきである。
- integer型またはunsignedInt型の値は、小数部または指数部のないJSON数としてシリアライズしなければならない。
- double型の値は、JSON番号としてシリアライズしなければならない。
- boolean型の値は、JSONブールとしてシリアライズしなければならない。

TD情報モデルの複合型(すなわち、配列、マップ、および<u>クラス</u>インスタンス)はすべて、以下の規則に従って、構成されたJSON型(配列およびオブジェクト)にマッピングされる。

- 型Arrayの値は、JSON配列としてシリアライズされなければならない。名前・値ペアの各値は、そのペアの数値名で順序付けられたJSON配列の要素とする。
- 型Mapの値は、JSONオブジェクトとしてシリアライズされなければならず、名前-値ペアの各値は、JSONオブジェクトのメンバーとする。 クラスインスタンスはJSONオブジェクトとしてシリアライズされなければならない。これは、個々に第6.3項情報モデルシリアライズで与えられている詳細な規則に従って行われなければな

6.2 デフォルト値の省略 §

TDシリアライズでは、第5.4項デフォルト値の定義の表のようにデフォルト値が定義されているボキャブラリ用語を省略してもよい。

以下の例は、<u>デフォルト値(</u>=チェックがついたチェックボックス)を持つメンバーも含めるためのチェックボックスを持つ<u>例</u>1のTDインスタンスである。これらのメンバーは、TDシリアライズ を単純化するために省略することができる(=チェックがついていないチェックボックス)。<u>TDプロセッサ</u>は、あたかも所与の<u>デフォルト値</u>によって明らかに存在するかのように、全く同じよう にこれらの省略されたメンバーを解釈することに留意されたい。

```
デフォルト値で
```

```
"@context": https://www.w3.org/2019/wot/td/v1,
"id": "urn:dev:ops:32473-WoTLamp-1234",
"title": "MyLampThing",
"securityDefinitions": {
    "basic_sc": {
        "Scheme":"basic",
        "in": "header
},
"security": [
    "basic sc'
],
"properties": {"status": {
        "type": "string", "readOnly" : false, "writeOnly" : false, "forms": [{
                "readproperty", "writeproperty"
            ],
            \hbox{``href'':''} https://mylamp.example.com/status \verb] \ ,
            "contentType": "application/json"
        }]
   }
},
"actions": {
    "toggle": {
        "safe": false,
        "idempotent": false,
        "forms": [{
            "op": "invokeaction",
             「href」:"https://mylamp.example.com/toggle2,
            "contentType": "application/json"
        }]
},
"events": {
    "overheating": {"data": {
            "type": "string",
            "readOnly" : false, "writeOnly" : false
       },
        "forms": [{
            "op": "subscribeevent",
            "href": https://mylamp.example.com/oh,,
             "contentType": "application/json",
            "subprotocol": "longpoll"
       }]
}
```

使用される<u>プロトコルバインディン</u>グに応じて、追加のプロトコル固有の<u>ボキャブラリ用語</u>が適用されることがあるということに留意されたい。これらは、関連する<u>デフォルト値</u>を持っており、したがって、本サブ項で説明されているように省略することもできる。さらなる情報は、<u>第8.3項プロトコルバインディング</u>に記載されている。

6.3 情報モデルのシリアライズ §

6.3.1 モノのルートオブジェクト 『

TDは、型Thingの<u>オブジェクト</u>をルートとするデータ構造である。その代り、TDのJSONシリアライズは、<u>TD</u>情報モデルから構築された構文ツリーのルートであるJSONオブジェクトである。

TDシリアライズのルート要素は、@contextという名前のメンバーと、https://www.w3.org/2019/wot/td/v1と等しいか、それを含むストリングまたは配列型の値を含むJSONオブジェクトで**なければならない。**

一般に、このURIは、本仕様が定義するTD表現フォーマットバージョンを識別するために使用される。JSON-LD 処理[<u>son-ld11]</u> の場合、このURI はTDコンテキストファイルを指定する。配列型の @contextは、<u>TDコンテキスト拡張子</u>を示す(詳細は<u>第7項 T D コンテキスト拡張子</u>を参照)。

Example 4 { "@context": "https://www.w3.org/2019/wot/td/v1", ... }

名前がThingの<u>シグニチャ</u>内の<u>ポキャブラリ用語</u>である場合、Thingのインスタンスの名前-値ペアは、ルートオブジェクトのJSONメンバーとしてシリアライズされな**ければならない。**すべての必須 およびオプションメンバーを含むシリアライズされたルートオブジェクトのTDスニペットを以下のとおり:

```
Example 5: Sample of Thing serializations

{
    "@context": "https://www.w3.org/2019/wot/td/v1",
    "@ype": "Thing",
    "tide: "MyThing",
    "tide: "MyThing",
    "tides: "MyThing",
    "tidesorption": "Human readable information.",
    "description": "Human readable information.",
    "description": (-...),
    "support": "mailtosupport@example.com",
    "version": (-...),
    "created": "2019-6.01T09:12:43.124Z",
    "security Definitions": (-...),
    "security Definitions": (-...),
    "security: -...
    "base": "https://servient.example.com/",
    "properties": (-...),
    "inks": [-...],
    "links": [-...],
    "links": [-...],
    "forms": [-...]
```

<u>クラス</u>Thingのインスタンス内のversion、securityDefinitions、properties、actions、eventsに割当てられたすべての値は、JSONオブジェクトとしてシリアライズされなければならない。

<u>クラス</u>Thingのインスタンス内のlinksとformsに割り当てられた値はすべて、 $\hat{\mathbf{g}}$ 6.3.8項linksと $\hat{\mathbf{g}}$ 6.3.9項formsでの定義のようにJSONオブジェクトを含むJSON配列としてシリアライズされなければならない。

<u>クラス</u>Thingのインスタンス内でsecurityに割り当てられた値は、JSONストリングとして、または、要素がJSONスト

リングのJSON配列としてシリアライズされなければならない。

6.3.2 人間が読み取り可能なメタデータ 『

titleおよびdescriptionという名前のJSONメンバーは、人間が読み取り可能なメタデータを表示するTDドキュメント内で使用される。これらは、TDドキュメントを確認する開発者のためのコメントとして、または、ユーザインターフェースのための表示テキストとして使用することができる。

第5.3.1.1項Thingで定義されているように、人間が読み取り可能なメタデータを表示するために使用される基本テキストの方向は、最初の強力な規則などのヒューリスティックスを使用して推定するか、言語情報から推論することができる。TDドキュメントでは、デフォルト言語は、@context内の@languageに割り当てられた値によって定義され、これは、必要に応じてスクリプトサブタグと共に、ベーステキスト方向を決定するために使用することができる。しかし、人間が読み取り可能なテキストを解釈するとき、各人間が読み取り可能なストリング値は、独立して処理され**なければならない。**言い換えれば、<u>TDプロセッサ</u>は、1つのストリングから別のストリングへの方向を変化させること、または、TD内の他の場所から別のストリングの方向を推論することはできない。

Note

Strings on the Web [STRING-META] suggests both strong-first and language-based inferencing as means to determine the base text direction. Given that the Thing Description format is based on JSON-LD 1.1 [json-ld11], which currently lacks explicit direction metadata, these approaches are currently considered appropriate at the time of this publication. However, if JSON-LD 1.1 adopts support for explicit base direction metadata as recommended by [STRING-META], the Thing Description format should be updated to take advantage of that feature.

ウェブ上のストリング[<u>string-meta</u>]は、基本テキストの方向を決定する手段として、強力な最初の推論、また、言語ベースの推論の両方を示唆している。TDフォーマットがJSON-LD 1.1[<u>json-ld11</u>]に基づいており、これは、現在、明示的な方向メタデータを欠いているので、これらのアプローチは、現在、本公開時点で適切であると考えられている。しかしながら、JSON-LD 1.1が[<u>string-meta</u>]が推奨するような明示的な基本方向メタデータのサポートを採用する場合、TDフォーマットは、その機能を利用するために更新されるべきである。

titleおよびdescriptionを使用するTDスニベットを以下のとおりである。デフォルト言語は、@context 配列内のJSON オブジェクト内の@language メンバーの定義によってen に設定される。

titlesおよびdescriptionsという名前のJSONメンバーは、一個のTDドキュメント内において複数言語で人間が読み取り可能なメタデータを提供するためにTDドキュメント内で使用される。 MultiLanguageマップのすべての名前-値ペアは、JSONオブジェクトのメンバーとしてシリアライズされなければならない。ここで、名前は<u>IBCP47</u>によって定義された周知の言語タグであり、値は、そのタグによって示された言語の人間が読み取り可能なストリングである。詳細は<u>第5.3.1.7項MultiLanguage</u>参照。TDドキュメント内のすべてのMultiLanguageオブジェクトには、同じ言語メンバーセットが含まれるべきである。

異なるレベルでtitlesおよびdescriptions使用するTDスニペットは以下のとおりである。

```
Example 7

(**@context** https://www.wd.org/2019/wot/dv/1*,
**tille**: "My Thing**,
**tof**: "My Thing**,
**t
```

TDインスタンスは、titleおよびdescriptionの使用をtitlesおよびdescriptionsと組み合わせることもできる。titleとtitles、あるいは、descriptionとdescriptionsが同じJSONオブジェクト内に存在する場合、titleおよびdescriptionの値はデフォルトテキストとして見ることができる。titleとtitles、あるいは、descriptionとdescriptionがTDドキュメントに存在する場合、各titleおよびdescriptionメンバーを持っているべきである。デフォルトテキストの言語は、デフォルト言語で示され、これは、通常、TDンスタンスの作成者によって設定される。

もう一のデフォルト言語の設定用法は、HTTPのAccept-Languageへッダーフィールドなどの言語ネゴシエーションメカニズムを介することである。デフォルト言語がネゴシエートされていた場合、ネゴシエーションの結果と返されたコンテンツのデフォルト言語を示すために、@languageメンバーが存在しな**ければならない。**デフォルト言語のネゴシエーションが成功した場合、TDドキュメントは、titlesおよびdescriptionsメンバー内のMultiLanguageオブジェクトよりも慢先して、適切で整合するtitleおよびdescriptionメンバー値を持っているべきである。しかしながら、モノは、そのような動的に生成されたTDをサポートしない、また、(例えば、リソース制約のために)言語ネゴシエーションをサポートしないと選択してもよいことに留意されたい。

633 vorcion

名前がVersionInfoの<u>シグニチャ</u>に含まれる<u>ボキャブラリ用語</u>である場合、VersionInfoのインスタンスの名前·値ペアは、すべて、名前として<u>ボキャブラリ用語</u>を持つJSONメンバーとしてシリアライズされ**なければならない**。

バージョン情報オブジェクトのTDスニペットは以下のとおりである。

```
Example 9
{
...
"version": { "instance": "1.2.1" },
...
}
```

Versionメンバーは、TDコ<u>ンテキスト拡張子</u>に基づく追加のアプリケーションおよび/またはデバイス固有のバージョン情報のコンテナである。詳細については、<u>第7.1項意味論的注釈</u>を参照。

6.3.4 securityDefinitions & security

Thingインスタンスでは、securityDefinitionsに割り当てられる値は、SecuritySchemeのインスタンスの<u>マップ</u>である。SecuritySchemeインスタンスの<u>マップ</u>のすべての名前・値ペアは、<u>マップ</u>をシリアライズした結果のJSONオブジェクトのメンバーとしてシリアライズされ**なければならない。**ペアの名前はJSONストリングとして、SecuritySchemeのインスタンスであるペアの値はJSONオブジェクトとしてシリアライズされ**なければならない。**

SecuritySchemeの中の一つの<u>サブクラス</u>のインスタンスの名前・値ペアすべては、その名前がその<u>サブクラス</u>の<u>シグニチャ</u>またはSecuritySchemeの<u>シグニチャ</u>に含まれる<u>ボキャブラリ用語</u>である場合、SecurityScheme<u>サブクラス</u>のインスタンスを名前として<u>ボキャブラリ用語</u>でシリアライズした結果であるJSONオブジェクトのメンバーとしてシリアライズされ**なければならない**・

以下のTDスニペットは、基本的なユーザ名/パスワード認証をヘッダーで指定する単純なセキュリティ構成である。inに与えられる値は、実際には、デフォルト値(header)であり、省略することもできる。名前付きセキュリティ構成は、securityDefinitionsマップで指定されなければならない。その定義は、securityメンバーにそのJSON名を組み込むことによってアクティブ化されなければならない。そのJSON名は1つの定義だけがアクティブ化された時にストリング型となることができる。

```
Example 10
...
"securityDefinitions": {
  "basic_so": {
    "scheme: "basic",
    "in": "header"
  }
},
"security": "basic_so",
...
```

これは、より複雑な例であり、<u>モノ</u>のベアラトークン認証と組み合わされたプロキシのダイジェスト認証を示すTDスニベットである。digestスキームでは、inの<u>デフォルト値</u>(すなわち、header)は省略されるが、依然として適用される。ユーザ名/パスワードおよびトークンなどの対応するプライベートセキュリティ構成は、正常に対話するために<u>コンシューマ</u>内で構成されなければならないことに留意されたい。複数のセキュリティ定義をアクティブ化すると、securityメンバーは配列となる。

```
Example 11

"security Definitions": {
    'proxy_se': {
        "scheme": 'digest",
        "proxy': 'https://portal.example.com/"
    },
    bearer_se": {
        "in": 'header",
        "scheme": 'bearer",
        "soneme": 'bearer",
        "alg": "ES256",
        "authorization": 'https://servient.example.com:8443"
    }
},
"security": ['proxy_se", "bearer_se"],
...
```

TDでのセキュリティ構成は必須である。セキュリティ定義は少なくとも1つ、モノレベルで(すなわち、TDルートオブジェクトで)security配列を通してアクティブ化されな**ければならない。**この構成 は、<u>モノ</u>と対話するために必要なデフォルトのセキュリティメカニズムと見なすことができる。また、セキュリティ定義は、モノレベルでアクティブ化されたすべての定義を無効にする(すなわち、完全に置き換える)フォームオブジェクトにsecurityメンバーを含めることによって、フォームレベルでのアクティブ化も**できる。**

nosecセキュリティスキームは、セキュリティが必要とされない場合に提供される。<u>モノ</u>の最小セキュリティ構成は、以下の例に示すように、モノレベルでのnosecセキュリティスキームのアクティブ化である。

より複雑な例として、我々が、すべての<u>対話アフォーダンス</u>が、認証が必要とされないものを除いて、基本認証を要求する<u>モノ</u>を持っていると仮定してみよう。statusプロパティおよびtoggleアクションについては、basic認証が要求され、モノレベルで定義される。しかしながら、overheatingイベントに関しては、認証は必要なく、従って、セキュリティ構成は、フォームレベルでオーバーライドされる。

セキュリティ構成は、同じ<u>対話アフォーダンス</u>内の異なるフォームに対して指定することもできる。これは、例えば、HTTP及びCoAP [<u>RFC7252</u>] のような異なるセキュリティメカニズムをサポート する複数のブロトコルをサポートするデバイスに対して要求される。また、代替の認証メカニズムが許可される場合にも有用である。ここで、ブロパティアフォーダンスをアクティブ化する3つの方 法を示すTDスニペットを紹介する。基本認証を使用したHTTPS、ダイジェスト認証を使用したHTTPS、ベアラトークン認証を使用したHTTPである。言い換えれば、複数のフォーム内で異なる セキュリティ構成を使用すれば、「ORI 方式でセキュリティメカニズムを組み合わせられる。対照的に、複数のセキュリティ構成を同じsecurityメンバーに構成すると、それらを「AND」方式で組 み合わせるということになるが、その場合、それらはすべて、<u>対話アフォーダンス</u>のアクティブ化を可能にするために満足される必要があるためである。モノレベルで(デフォルト)構成を1つアク ティブ化することは、依然として、必須であることに留意されたい。

もう一つより複雑な例として、OAuth2はスコープを利用する。トークン内に現れる可能性があり、そのリソース(またはW3C WoTの場合は<u>対話アフォーダンス</u>)へのアクセスを可能にするために、リソース内の対応する識別子と一致しなければならない識別子である。例えば、以下の例では、statusプロパティは、スコープlimitedを含むベアラトークンを使用する<u>コンシューマ</u>が読み取ることができるが、configureアクションは、specialスコープを含むトークンを用いることによって呼び出すことができるのみである。スコープは、ロールと同一ではないが、しばしばロールに関連付けられ、例えば、おそらく、管理ロール内のスコープのみが、「特別な」対話を実行することを許可される。トークンは複数のスコープを持つことができる。本例では、管理者には、おそらく、limitedでsepcialなスコープ両方を持つトークンが発行され、一方、通常のユーザにはlimitedスコープを持つトークンのみが発行される。

6.3.5 properties §

Thingインスタンス内のpropertiesに割り当てられる値は、PropertyAffordanceのインスタンスの<u>マップ</u>である。PropertyAffordanceインスタンスの<u>マップ</u>の名前-値ペアは、すべて、<u>マップ</u>をシリアライズした結果のJSONオブジェクトのメンバーとしてシリアライズされな**ければならない**。ペアの名前はJSONストリングとして、PropertyAffordanceインスタンスであるペアの値はJSONオブジェクトとしてシリアライズされなければならない。

PropertyAffordanceインスタンスの名前・値ペアは、すべて、その名前がPropertyAffordance、InteractionAffordance、あるいは、DataSchemaの<u>シグニチャ</u>(の1つ)に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたPropertyAffordanceインスタンスをシリアライズした結果得られるJSONオブジェクトのメンバーとしてシリアライズされなけ**ればならない。** <u>DataSchema</u>インスタンスのシリアライズの詳細については、<u>第6.3.10</u>項参照データスキーマ参照。

PropertyAffordanceインスタンス内のformsに割り当てられた値は、第6.3.9項formsで定義されているように1つ以上のJSONオブジェクトシリアライズを含むJSON配列としてシリアライズされなければならない。

2つの<u>プロパティ</u>アフォーダンスのスニペットを以下に示す。

Example 16: Sample of Property serializations

```
"properties": {
  "on": {
  "type": "boolean",
  "forms": [...]
},
  "status": {
  "type": "object",
  "properties": {
  "type": "number",
  "minimum": 0.0,
  "maximum": 100.0
},
  "gb": {
  "type": "array",
  "items": {
  "type": "number",
  "minimum": 0,
  "maximum": 255
},
  "minitems": 3,
  "maxltems": 3
},
  "required": ['brightness", "rgb"],
  "forms": [...]
},
```

6.3.6 actions 5

Thingインスタンスでは、actionsに割り当てられる値は、ActionAffordanceのインスタンスの<u>マップ</u>である。ActionAffordanceインスタンスの<u>マップ</u>の名前-値ペアは、すべて、<u>マップ</u>をシリアライズした結果のJSONオブジェクトのメンバーとしてシリアライズされ**なければならない**。ペアの名前はJSONストリングとして、ActionAffordance インスタンスであるペアの値はJSONオブジェクトとしてシリアライズされ**なければならない**。

ActionAffordanceインスタンスの名前・値ペアはすべて、その名前がActionAffordance、あるいは、InteractionAffordanceの<u>シグニチャ(</u>の1つ)に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたActionAffordanceインスタンスをシリアライズした結果得られるJSONオブジェクトのメンバーとしてシリアライズされなければならない。

ActionAffordanceインスタンスでoutputとinputに割り当てられる値は、JSONオブジェクトとしてシリアライズされなければならない。これらは、クラスdataschemaに依存し、そのシリアライズは、第6.3.10項データスキーマで定義される。

ActionAffordanceのインスタンスでformsに割り当てられる値は、第6.3.9項formsで定義されているように1つ以上のJSONオブジェクトシリアライズ含むJSON配列としてシリアライズされなければならない。

アクションアフォーダンスのTDスニペットを以下に示す。

Example 17: Sample of an Action serialization

```
"actions": {
   "fade": {
    "tide": Fade in/out",
    "description": "Smooth fade in and out animation.",
    "input": {
        "type": "object",
        "properties": {
        "trom": {
            "type": "integer",
            "minimum": 0,
            "maximum": 100
        },
        "to": {
            "type": "integer",
            "minimum": 0,
            "maximum": 100
        },
        "duration": {"type": "number"}
      },
      "required": ["to", "duration"],
    },
    "output": {"type": "string"},
    "forms": [...]
},
...
```

6.3.7 events 5

Thingインスタンスでは、eventsに割り当てられる値は、EventAffordanceのインスタンスのマップである。EventAffordanceインスタンスのマップの名前・値ペアは、すべて、マップをシリアライズした結果のJSONオブジェクトのメンバーとしてシリアライズされなければならない。ペアの名前はJSONストリングとして、EventAffordanceインスタンスであるペアの値はJSONオブジェクトとしてシリアライズされ**なければならない。**

EventAffordanceインスタンスの名前・値ペアはすべて、その名前がEventAffordance、あるいは、InteractionAffordanceの<u>シグニチャ(</u>の1つ)に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたEventAffordanceインスタンスをシリアライズした結果得られるJSONオブジェクトのメンバーとしてシリアライズされなければならない。

EventAffordanceインスタンスでsubscription、data、および、cancellationに割り当てられる値は、JSONオブジェクトとしてシリアライズされ**なければならない。**これらは<u>クラスDataSchema</u>に依存し、そのシリアライズは<u>第6.3.10</u>項データスキーマで定義される。

EventAffordanceのインスタンスでformsに割り当てられる値は、第6.3.9項formsで定義されているように1つ以上のJSONオブジェクトシリアライズを含むJSON配列としてシリアライズされなければならない。

イベントオブジェクトのTDスニペットを以下に示す。

Example 18: Sample of an Event serialization

```
"events": {
    "overheated": {
        "data" : {
            "type": "string"
        },
        "forms": [...]
    }
},
```

イベントアフォーダンスは、既存の(例えばWebSub[<u>websub</u>])または顧客向けイベントメカニズム(例えばWebhoks)を採用するために、柔軟に定義されている。このため、所望のメカニズムに 従って、subscriptionおよびcancellationを定義することができる。詳細は<u>[WoT-BindingTemplates]</u>参照。<u>例A.3 Webhook イベント例</u> は、Webhook を説明するためにイベントがどのように subscriptionおよびcancellationを使用できるかを例示している。

6.3.8 links

linkインスタンスの名前-値ペアはすべて、その名前がlinkの<u>シグニチャ</u>に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたlinkインスタンスをシリアライズした結果得られるJSONオブジェクトのメンバーとしてシリアライズされな**ければならない。**

links配列内のリンクオブジェクトのTDスニペットを以下に示す。

Example 19: Sample of a Link serialization

```
...
"inks": [{
    "rel": "controlledBy",
    "hrrel": "https://servient.example.com/things/lampController",
"type": "application/td+json"
}]
```

6.3.9 forms 5

formインスタンスの名前-値ペアはすべて、その名前がFormの<u>シグニチャ</u>に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたFormインスタンスをシリアライズした結果 得られるJSONオブジェクトのメンバーとしてシリアライズされなければならない

必要に応じて、フォームオブジェクトは、プレフィックスで識別されるプロトコル固有のボキャブラリ用語で補足されてもよい。第8.3項プロトコルバインディングも参照。

forms配列内のフォームオブジェクトのTDスニペットを以下に示す。

Example 20: Sample of a Form serialization

```
"forms": [{
    "op": "writeproperty",
    "href": "http://mytemp.example.com:5683/temp",
    "contentType": "application/json",
    "htv:methodName": "POST"
}]
```

hrefには、http://192.168.1.25/left?p=2&d=1のpやdなどのダイナミック変数を含むURIを入れることもできる。その際、URIは、[RFC6570]http://192.168.1.25/left{?p,d}で定義されているようにテンプレートとして定義することができる。

そのような場合、URIテンプレート変数は、JSON名として関連付けられた(一意の)変数名を持つJSONオブジェクトベースのuriVariablesメンバーに集められなければならない。

Formインスタンス中のuriVariablesに割り当てられるマップ中の各値のシリアライズは、<u>クラスDataSchema</u>に依存しな**ければならない。**そのシリアライズは、<u>第6.3.10項データスキーマ</u>で定義され

URI テンプレートとuriVariables を使用したTD スニペットを以下に示す。

```
Example 21

{
    "@context": [
        "https://www.w3.org/2019/wot/tdv1",
        { "eg": "http://www.example.org/ot#" }
    ...
    "actions": {
        "ur/Variables": {
            "p": {"type": "integer", "minimum": 0, "maximum": 16, "@type": "eg:SomeKindOfAngle" },
            """ ' "; "type": "integer", "minimum": 0, "maximum": 1, "@type": "eg:Direction" }
    },
    "forms": [{
        "hreft: "http://192.168.1.25/left[?p.d]",
        "httv:methodName": "GET"
    }]
    },
    ...
},
...
}
```

contentTypeメンバーは、「;文字」で区切られた属性-値ペアとしてメディアタイプパラメータを含むメディアタイプ[RFC2046]を割り当てるために使用される。例:

```
Example 22
...
"contentType" : "text/plain; charset=utf-8",
...
```

いくつかの使用事例では、<u>対話アフォーダンス</u>のフォームメタデータは、要求を記述するだけでなく、期待される応答のためのメタデータも提供する。例えば、アクションtakePhotoは、要求ベイロードのJSON(すなわち、"contentType":"application(json")を使用してカメラのパラメータ設定(アパーチャ優先順位、タイマなど)を送るためのinputスキーマを定義する。このアクションの出力は撮影された写真であり、これは、例えば、JPEGフォーマットで可能となる。そのような場合、responseメンバーは、応答ベイロードの表現フォーマット(例えば、"contentType":"image/jpeg")を示すために使用される。ここでは、コンテンツタイプが表現フォーマットを完全に指定するのでoutputスキーマは必要とされない。

Formインスタンスでresponseに割り当てられる値は、それが存在する場合、JSONオブジェクトでな**ければならない。**応答オブジェクトは、それが存在する場合、<u>ExpectedResponse</u>の<u>クラス</u>定義の中で定義されているcontentTypeメンバーを含んでい**なければならない。**

上記のアクションtakePhotoに基づいて、responseメンバーを持つformスニペットを以下に示す。

formsがトップレベルに存在する場合、それは、<u>モノ</u>が提供するメタ対話を記述するために使用することができる。例えば、操作タイプ「readallproperties」および「writeallproperties」は、<u>コンシューマ</u>がすべてのプロパティを一度に読み取りと書き込みをすることができる<u>モノ</u>とのメタ対話のためのものである。以下の事例では、formsメンバーがTDルートオブジェクトに含まれ、<u>コンシューマ</u>が一つのプロトコルトランザクションで<u>モノ</u>のすべてのプロパティ(すなわち、on, brightness, timer)を読み取るか、または、書き込むために、送信対象https://mylamp.example.com/allpropertiesを使用することができる。

オペレーションタイプwriteallpropertiesの場合、<u>コンシューマ</u>は書き込み可能なプロパティ**すべて**と(新しく)割り当てられた値(例:ペイロード内)を提供することが求められる。さもなければ、<u>モノ</u>は不整合を避けるためにこの呼び出しを拒否してよい。

6.3.10 データスキーマ ¶

<u>DataSchemaクラス</u>で定義されたWoTTDのデータスキーマは、JSONスキーマ用語のサブセット[<u>JSON-SCHEMA</u>] のに基づいている。したがって、<u>モノ</u>とやり取りされるデータを検証するためにTD データスキーマのシリアライズをJSONスキーマバリデーターのインプリメンテーションに直接与えることができる。

 $\vec{r} - \textit{y} - \textit{x} + - \textit{y} - \textit{y}$

割り当てられる値、EventAffordanceインスタンスでsubscription, data, cancellationに割り当てられる値、および、(<u>フォームオブジェクト</u>がURIテンプレートを使用する場合)InteractionAffordance の<u>サブクラス</u>のインスタンスでuriVariablesに割り当てられる値に適用される。

DataSchemaの一つの<u>サブクラス</u>インスタンスの名前-値ペアはすべて、その名前がその<u>サブクラスのシグニチャ</u>、あるいは、DataSchemaの<u>シグニチャ</u>に含まれる<u>ボキャブラリ用語</u>である場合、名前として<u>ボキャブラリ用語</u>のついたDataSchemaの<u>サブクラス</u>インスタンスをシリアライズした結果得られるJSONオブジェクトのメンバーとしてシリアライズされ**なければならない。**

ObjectSchemaインスタンス内のpropertiesに割り当てられる値は、JSONオブジェクトとしてシリアライズされなければならない。

DataSchemaインスタンスのenum、required、および、oneOfに割り当てられる値は、JSON配列としてシリアライズされなければならない。

ArraySchemaのインスタンス内のitemsに割り当てられる値は、JSONオブジェクトまたはJSONオブジェクトを含むJSON配列としてシリアライズされなければならない。

TDスニペットデータスキーマメンバーを以下に示す。周囲のオブジェクトは、データスキーマオブジェクト(例えば、input、output用)又は付加的なメンバーを含むプロパティオブジェクトであって もよいことに留意されたい。

Example 25: Sample of a DataSchema serialization

```
"type": "object",
"properties": {
    "status": {
        "title": "Status",
        "type": "string",
        "enum': "On", "Off", "Error"]
},
    "brightness": {
        "title": "Brightness value",
        "type": "number",
        "minimum": 0.0,
        "maximum": 100.0
},
    "gb": {
        "title": "RGB color value",
        "type": "array",
        "items": {
              "ype": "number",
              "minimum": 255
},
        "minitems": 3,
        "maxtlems": 3,
    }
}...
```

readOnlyおよびwriteOnlyという用語は、読み取り対話(すなわち、プロパティを読み取る時)においてどのデータ項目をやり取りするか、および書き込み対話(すなわち、プロパティを書き込む時)においてどのデータ項目をやりとりするかを知らせるために使用することができる。これは、従来型でない<u>モノ</u>のプロパティが、読み取りおよび書き込みのために異なるデータを示すときの回避策として使用することができ、これは、TDを用いて既存のデバイスまたはサービスを増補するときに発生する可能性がある。

readOnly およびwriteOnly を使用したTD スニペットを以下に示す。

```
Example 26

"properties"; {
    "status"; {
    "description": "Read or write On/Off status.",
    "ype": 'object',
    "properties"; {
    "latestStatus"; {
        "type": 'string",
        "enum": ['On', 'Off],
        "readOnly": true
    },
    "newStatusValue"; {
        "type": string",
        "enum": ['On', 'Off],
        "writeOnly": true
    }
},
    torms: [...]
}

orms: [...]
}
```

statusプロパティが読み込まれると、ペイロード内のlatestStatusメンバーを使用してステータスデータが返される。statusプロパティを更新するには、ペイロード内のnewStatusValueメンバーを介して新しい値が提供されなければならない。

追加機能として、TDインスタンスでは、データスキーマ内のunitメンバーが使用できる。これによって、測定単位をデータアイテムに関連付けることができる。そのストリング値は、自由に選択することができる。しかしながら、周知のボキャブラリで定義されている単位を選択することが推奨される。例については、第7項TDコンテキスト拡張子参照。

6.4 Identification §

TDのJSONベースのシリアライズは、メディアタイプapplication/td+json、または、CoAPコンテンツフォーマットID T.B.D.によって識別される(<u>第</u>10項IANA考慮事項を参照)。

Note: CoAP Content-Format

CoAP-based WoT implementations can use the experimental Content-Format 65100 until the final Content-Format ID has been assigned.

□ 注: CoAPコンテンツフォーマット

CoAPベースのWoTインプリメンテーションでは、正式コンテンツフォーマットIDが割り当てられるまで、試験的なコ~ンテンツフォーマット65100を使用することができる。

7. TD Context Extensions §

This section is non-normative.

本項は標準ではない。

<mark>第 5 項</mark>TD情報モデルの標準<u>ボキャブラリ</u>定義に加えて、WoTTDは、追加の名前空間からコンテキスト知識を追加する機能を提供している。本メカニズムは、TDインスタンスを追加の(例えば、ドメ イン固有の)意味論で強化するために使用することができる。また、将来、追加の<u>プロトコルバインディング</u>や新しいセキュリティスキームをインボートするために使用することもできる。

そのような<u>TDコンテキスト拡張子</u>について、TDは、JSON-LD <u>[ison-ld11]</u>で周知の@contextメカニズムを使用する。<u>TDコンテキスト拡張子</u>を使用する場合、<u>クラス</u>Thingの@context値は、JSON-LD コンテキストファイルを識別するanyURI型の追加要素を持つ配列、あるいは、<u>第5.3.1.1項Thing</u>で定義されている通り、名前空間IRIを含む<u>マップ</u>である。.

第6.1項 JSON型へのマッピングにある複合型のシリアライズ規則は、拡張@context名前-値ペアのシリアライズを定義している。TDコンテキスト拡張子を持つスニペットを以下に示す。

```
Example 27

{
    "@context": [
        "https://www.w3.org/2019/wot/td/v1",
        {
            "eg": "http://example.org/iot#",
            "cov": "http://www.example.org/coap-binding#"
        },
        "https://schema.org/"
    },
    "https://schema.org/"
},
```

7.1 意味論的注釈 5

<u>TDコンテキスト拡張子</u>は、TDインスタンスへの<u>ボキャブラリ用語</u>追加を可能にする。含まれる名前空間が、RDFスキーマまたはOWLによって提供される定義などの<u>クラス</u>定義に基づく場合、それ らを使用して、インスタンスをそのような外部<u>クラス</u>定義に関連付けることによって、TDの<u>クラス</u>インスタンスに意味論的に注釈を付けることができる。これは、@type名前-値ペアに<u>クラス</u>名を割 り当てるか、または、複数の関連付け/注釈の配列値に<u>クラス</u>名を入れて行う。 $\frac{\mathbf{\pi}6.1\bar{\mathbf{u}} \mathbf{JSON} \underline{\mathbf{u}} - \mathbf{v} \mathbf{v} \mathbf{v}}{\mathbf{v}}$ 内のシリアライズ規則に従って、@typeは、JSON ストリングまたはJSON <u>配列</u>としてシリアライズされる。@type は、ノード型を設定するために使用されるJSON-LDキーワード[json-ld11] である。

TDコンテキスト拡張子は、TDの任意の<u>クラス</u>インスタンス内に追加の名前-値ペアおよび明確に定義された値を含めることも可能にする。これらのペアおよび値は、含まれる<u>ボキャブラリ用語</u>で定 義され、それぞれ、対応するJSONオブジェクト内の追加メンバーまたは既存メンバーの値としてシリアライズされる。例として、<u>モノ</u>の追加バージョンメタデータ、または、データアイテムの測 定単位がある。

一例として、以下に示すTDスニベットは、<u>モノ</u>のハードウェアおよびファームウェアのバージョン番号を追加することによってバージョン情報コンテナを拡張し、<u>モノや例2とOM</u>(測定単位オントロジー[<u>RIJGERSBERG</u>]) でも使用されているデータスキーマユニット: <u>SAREF</u>用に外部の<u>ボキャブラリ</u>の値を使用する。これらのボキャブラリは、例として使用されており、家庭自動化領域では特にその他の<u>ボキャブラリ</u>が存在するかもしれない。

多くの場合、<u>TDコンテキスト拡張子</u>は、対話中に(レスポンスのベイロード内で)データ交換により表示される物理世界のオブジェクトの状態情報の意味論的処理を可能にするために使用されるかもしれない。たとえば、RDF内のこの状態情報の意味論的説明は、<u>TDドキュメント</u>に埋め込むことができ、データスキーマの項目は、物理世界のオブジェクトのRFDモデル状態の特定部分の参照を促す注釈を個々につけることができる。

下記のTDスニベットは、ランプの上他ウィを説明するためにSAREFを使用している。<u>SSN</u>(Semantic Sensor Network Ontology. 意味論的センサーネットワークオントロジー)[<u>Vocab-SSN</u>]から取り入れた外部<u>ボキャブラリ用語</u>ssn:forPropertyは、status<u>プロパティ</u>のデータスキーマを物理世界オブジェクトの実際のオン/オフ状態にリンクするために使用されている。

例2では、モノの状態は、それ自体のstatusアフォーダンスによって与えられ、起こりうる状態変化は、toggleアフォーダンスで与えられる。つまり、物理世界オブジェクトの状態は、直接、モノの対話アフォーダンスを提供するということである。この設計はシンプルなケースにおいては十分である。より複雑なケースでは、しかし、複数のアフォーダンスが同じ物理的状態に使用できることもある。上記の例では、fullStatusプロパティが、ランプの状態をより多くの言葉で表現する他の方法を提供している。

7.2 プロトコルバインディングの追加

本項は標準ではない。

TDのTDコ<u>ンテキスト拡張子</u>を使って通信メタデータを補足することができ、または、Formインスタンスを表すJSONオブジェクトにシリアライズされた追加の<u>ボキャブラリ用語</u>で新しい<u>プロトコルバインディング</u>を追加することができる。(第8.3項プロトコルバインディングも参照)。

以下のTDの例では、仮想CoAP<u>プロトコルパインディング</u>を使用する。というのは、このような<mark>プロトコルパインディング</mark>が本仕様書執筆時点では存在しないためである。この<u>TDコンテキスト拡張子</u>は、名前空間例http://www.example.org/coap-binding#を介してアクセス可能なRDF1.0[<u>HTTP-in-RDF10</u>]内のHTTPボキャブラリに類似したCoAP RDFボキャブラリがあると仮定している。補足されたcov:methodNameメンバーは、どのCoAPメソッドコード0.02の場合はPOST、またはCoAPメソッドコード0.07の場合はIPATCH)。

Example 30: Specialization of forms through TD Context Extension

```
"@context": [
    "https://www.w3.org/2019/wot/ta/v i , { "cov": "http://www.example.org/coap-binding#" }
1.
      "brightness": {
    "description": "The current brightness setting",
    "https:":"integer"
         "type": "integer"
"minimum": -64,
"maximum": 64,
          "forms": [{
             "op": "readproperty",
"href": "coap://example.org:61616/api/brightness",
"cov:methodName": "GET"
             {
"op": "writeproperty",
"href": "coap://example.org:61616/api/brightness",
             "cov:methodName": "POST
        }]
```

7.3 セキュリティスキームの追加 🦠

<u>第</u>5.3.3項 セキュリティボキャブラリ定義に含まれていない新しいセキュリティスキームは、<u>TDコンテキスト拡張子</u>メカニズムを使用してインボートすることができる。本例で は、<u>http://www.example.org/ace-security#</u>で名前空間が本例のために定義する[<u>ACE</u>]に基づく仮想ACEセキュリティ方式を使用している。このような追加のセキュリティスキームは、<u>クラスセキュリ <u>ティスキーム</u>の<u>サブクラス</u>でなければならないことに留意されたい。</u>

```
Example 31
               https://www.w3.org/2019/wot/td/v1",
                  "cov": "http://www.example.org/coap-binding#", "ace": "http://www.example.org/ace-security#"
        "securityDefinitions": {
                  "scheme": "ace:ACESecurityScheme",
                  "ace:as": "coaps://as.example.com/token"
"ace:audience": "coaps://rs.example.com"
"ace:scopes": ["limited", "special"],
"ace:cnonce": true
          security": ["ace sc"].
                    "forms": [{
                      Jrins - n
"op": "readproperty",
"href": "coaps://rs.example.com/status",
"contentType": "application/cbor",
"cov:methodName": "GET",
"ace:scopes": ["limited"]
                 }]
              "configure": {
                 "forms": [{
    "op": "invokeaction",
    "treft": "coaps/trs.example.com/configure"
    "contentType": "application/cbor",
    "cov:methodName": "POST",
    "cov:methodName": "POST",
                        "ace:scopes": ["special"]
                 }]
```

<u>第</u>5.3.3項 セキュリティボキャブラリ定義の中で定義されているセキュリティスキーム全てが既にTDコンテキストの一部であり、<u>TDコンテキスト拡張子</u>を使って含める必要はないということに留意

8. ビヘイビアのアサーション §

-ションは、TDの表現または情報モデルとは対照的に、WoTシステムのコンポーネントのビヘイビアに関するものである。しかしながら、TDは記述的であり、特に、前から存在する ネットワークインターフェースを記述するために使用されることもあるということに留意されたい。この場合、そのようなすでに存在するインターフェースのビヘイピアを制約するアサーションを 行うことはできない。代わりに、アサーションは、そのようなインターフェースを正確に表すために、TDに対する制約となると解釈されなければならない。

8.1 セキュリティ構成 5

安全な相互運用を可能にするために、セキュリティ構成はモノの要件を正確に反映しなければならない。

- モノが対話のために特定のアクセスメカニズムを要求する場合、そのメカニズムはTDのセキュリティ構成の中で指定されなければならない。
- モノが対話のために特定のアクセスメカニズムを要求しない場合、そのメカニズムはTDのセキュリティ構成の中で指定されてはならない。

8.2 データスキー

TDで提供されるデータスキーマは、TDで指定された対話の中で記述された<u>モノ</u>が返し、また、受け入れるデータペイロードを正確に表すべきである。一般に、<u>コンシューマ</u>は、WoTTDに与えられ ていないものを生成セず、厳密にデータスキーマに従うべきである。が、WoTTDに明示的に与えられていない<u>モノ</u>からの追加データを受け入れるべきである。一般的に、<u>モノ</u>は、WoTTDによって 記述されるが、<u>コンシューマ</u>は、<u>モノ</u>と対話するときにWoTTDに従わざるを得ない。

- WoTTDに記述されている別のターゲットのモノと対話するときに、<u>コンシューマ</u>として動作する<u>モノ</u>は、その対話で与えられたデータスキーマに従って編成されたデータを生成しな**ければな** らない。
- WoTTDは、各対話によって返され、また、受け入れられたデータを正確に記述しなければならない。
- <u>モノ</u>は、そのWoTTDで与えられたデータスキーマにそのようなデータが記述されていない場合でも、対話から追加データを返して**もよい。**これは、返されるデータ内に追加のプロパティあるいはアイテムがある可能性がある場合に、ObjectSchemaとArraySchema(itemsがDataSchemaの列にある場合)に適用される。これは、「<u>JASON-SCHEMA</u>」内に定義されるとおりに、
- "additioalProperties": true あるいは "additionalItems": trueがであるかのように挙動する。

 別のモノと対話するときに<u>コンシューマ</u>として動作する王ノは、ターゲットのモノのTDで与えられたデータスキーマに記述されていない追加データを確実に受け取らな**ければならない。**これは、返されるデータ内に追加のプロパティあるいはアイテムがある可能性がある場合に、ObjectSchemaとArraySchema(itemsがDataSchemaの列にある場合)に適用される。これは、 「JASON-SCHEMA」内に定義されるとおりに、"additioalProperties": true あるいは "additionalItems": trueがであるかのように挙動する。 別の<u>モノ</u>と対話するときに<u>コンシューマ</u>として動作する<u>モノ</u>は、その<u>モノ</u>のTDで与えられたデータスキーマに記述されていないデータを生成してはならない。
- ・別のモノと対話するときに<u>コンシューマ</u>として動作するモノは、UNLFシブレート、ベースURI、および、ターゲットのモノのTDで与えられるhrefバラン・タに従ってURIを生成しなければな
- WoTTD内のURIテンプレート、ベースURI、およびhrefメンバーは、モノのWoTインターフェースを正確に記述しなければならない。

8.3 プロトコルバインディング 5

<u>プロトコルパインディング</u>は、<u>対話アフォーダンス</u>から、HTTP [RFC7231]、CoAP [RFC7252]、MQTT [MQTT]</u>などの特定のプロトコルの具体的メッセージへのマッピングである。<u>対話アフォーダンスのプロトコルバインディング</u>は、第6.3.9項 formsで定義されているような形式でシリアライズされる。

WoTTD内のすべてのフォームには、hrefメンバーが与える送信ターゲットが入っていなければならない。この送信ターゲットのURIスキームは、<u>モノ</u>がどの<u>プロトコルバインディング</u>を実装しているか[<u>WoTArchitecture</u>]を示す。例えば、ターゲットがhttpまたはhttpsで始まる場合、<u>コンシューマ</u>は、<u>モノ</u>がHTTPベースの<u>プロトコルバインディング</u>を実装していることを推測することができ、フォームインスタンス内のHTTP固有の用語を期待すべきである(<u>第</u>8.3.1項 HTTPベースのプロトコルバインディングを参照)。

- WoTTD内のすべてのフォームは、そのhrefメンバーのURIスキームが示す<u>プロトコルバインディング</u>の要件に従わなければならない。
 WoT TD内のすべてのフォームは、対話内で<u>モノ</u>が受け入れる要求(要求ヘッダーなどがあれば)を正確に記述しなければならない。

8.3.1 HTTPベースのプロトコルバインディング 』

デフォルトに従い、TDは、RDF1.0 [<u>HTTP-in-RDF10</u>]内のHTTPボキャブラリのHTTPRDFボキャブラリ定義を入れることによって、HTTPベースの<u>プロトコルバインディング</u>をサポートする。この ボキャブラリは、<u>http://www.w3.org/2011/http#</u>を指すプレフィックスhtvを使ってTDインスタンス内で直接的に使用することができる。さらに、HTTPベースの<u>プロトコルバインディング</u>の詳細に関 しては、[WOT-BINDING-TEMPLATE]を参照。

HTTPベースの<u>プロトコルバインディング</u>を実装する<u>モノ</u>と対話するために、<u>コンシューマ</u>は、フォームを送信するときにどのHTTPメソッドを使用するかを知っている必要がある。一般的なケース では、TDは、メソッドを示す用語、すなわちhtv:methodNameを明示的に含むことができる。簡潔にするために、HTTPベースの<u>プロトコルバインディング</u>は、各操作タイプの<u>デフォルト値</u>を定義 し、これは、また、<u>モノ</u>が期待するメソッド(例えば、読み取りのためのGET、書き込みのためのPUT)の収束を目的とする。HTTPベースの<u>プロトコルバインディング</u>を表すフォームの中でメソッド が示されていない場合、<u>デフォルト値</u>は下表のように仮定されな**ければならない。**

Vocabulary term	Default value	Context
htv:methodName	GET	Form with operation type readproperty
htv:methodName	PUT	Form with operation type writeproperty
htv:methodName	POST	Form with operation type invokeaction

ボキャブラリ用語	デフォルト値	コンテキスト
htv:methodName	GET	操作タイプreadpropertyを持つのフォーム
htv:methodName	PUT	操作タイプwriteproperty を持つフォーム
htv:methodName	ポスト	操作タイプinvokeaction を持つフォーム

例えば、<u>第1項はじめに</u>の例1では、この形式のオペレーションタイプとHTTP方法は入っていない。以下の<u>デフォルト値</u>は、<u>例1</u>の形式のためであると考えるべきである。

Example 32

```
with default values for Protocol Binding based on HTTP
     "@context": "https://www.w3.org/2019/wot/td/v1", "id": "urn:dev:ops:32473-WoTLamp-1234", "title": "MyLampThing",
      "securityDefinitions": {
          "basic_sc": {
    "scheme": "basic",
    "in": "header"
      },
"security": [
          "basic_sc
       ],
"properties": {
  "status": {
    "type": "string",
    "forms": [
               {
    "href": "https://mylamp.example.com/status"
               }
          "toggle":
             "forms": [
               {
    "href": "https://mylamp.example.com/toggle"
               }
           ]
         }
       "events": {
          "overheating": {
            "data": {
    "type": "string"
              {
    "href": "https://mylamp.example.com/oh",
    "subprotocol": "longpoll"
} 1
               }
```

```
"@context": "https://www.w3.org/2019/wo"id": "urn:dev:ops:32473-WoTLamp-1234" "title": "MyLampThing",
 "securityDefinitions": {
    "basic sc": {
 "security": [
    "basic so
 "properties": {
     "status": {
    "type": "str
    "forms": [
                "op": "readproperty"
                "href": "https://mylamp.example.com/status",
               "htv:methodName": "GET
                "href": "https://mylamp.example.com/status"
"htv:methodName": "PUT"
      1
   }
    "toggle": {
    "forms": [
         "op": "invokeaction",
"href": "https://mylamp.example.com/toggle",
"htv:methodName": "POST"
      ]
 "events": {
              eating": {
        "data":
            "type": "string
         forms": [
               "op": "subscribeevent",
"href": "https://mylamp.example.com/oh",
```

8.3.2 その他のプロトコルバインディング ¶

モノが実装できる<u>プロトコルバインディング</u>の数は制限されていない。他の<u>プロトコルバインディング(</u>例えば、CoAP、MQTT、または、OPC UAのための)は、RDF 1.0[<u>HTTP-in-RDF10</u>]のHTTPボ キャブラリと同様のプロトコル<u>ボキャブラリ</u>、または、<u>デフォルト値</u>定義を含む仕様など別個の文書で標準化される事になっている。このようなプロトコルは、<u>TDコンテキスト拡張</u>メカニズムの使 用によってTDに単純に結合することができる(第7項TDコンテキスト拡張子を参照)。

IoTプラットフォームとエコシステムの説明方法に関する情報に関しては、[WOT-BINDING-TEMPLATE]を参照。

9. セキュリティとプライバシーに関する考慮事項 §

本項は標準ではない。

一般に、WoTシステムを保護するために取られるセキュリティ対策は、システムが直面する可能性がある脅威および攻撃者、ならびに、保護する必要がある資産の価値に依存する。さらに、プライ がシーリスクは、識別可能な人とモノの関連性、および、直接的な情報とそのような関連性から入手できる推測情報に依存することになる。様々な状況に適応させることができる解威モデルを含め、WoTに関するセキュリティおよびブライバシーの考慮事項の詳細な考察は、参考文献[WOT-SECURITY-CONSIDERATIONS]に記載されている。本項では、セキュリティとブライバシーリスク と、WoTTDに直接関連する実行可能な軽減対策についてのみ説明する

WoTTDは、安全なネットワークインターフェースと安全でないネットワークインターフェースの両方を記述することができる。TDが既存のネットワークインターフェースに組み込まれる場合、そ のネットワークインターフェースのセキュリティ状態に変化は期待できない。

WoTTDの使用は、以下の項で挙げられるセキュリティおよびプライバシーリスクを紹介している。各リスク説明の後、いくつかの実行可能な軽減対策を提案する。

9.1 プライバシーリスクをデリフェレンスするコンテキスト 5

JSON-LD [ison-Id11] ドキュメントの@contextメンバーで指定されたボキャブラリファイルのデリフェレンスはプライバシーリスクになりうる。WoTの場合、攻撃者は、そのようなデリフェレンス によって生成されたネットワークトラフィックを観察することができ、特にドメイン固有のボキャブラリが使用される場合、デバイスに関する情報を推論するために、宛先IPアドレスなどのデリ フェレンスのメタデータを使用することができる。これは、たとえ接続が暗号化されていてもリスクとなり、DNSプライバシーリークにつながる。

めに更新には新しいURIの使用が必須であるため、厳密なバージョン制御が必要となる。コンテキストファイルがTD内のメタデータを解釈するシステムにローカルで利用可能になる見込みを高める ことが可能な場合には必ず、周知の標準ボキャブラリファイルを使用する。

9.2 変更不能識別子のプライバシーリスク §

識別子(id)を含んでいるTDは、識別可能な人と関連付けられているモノを説明することができる。このような識別子は、トラッキングなどのさまざまなリスクを呈する。しかし、その識別子も変異することができない場合、デバイスが他の人に譲渡あるいは販売され、既知のIDがその人をトラッキングするために使用されるため、トラッキングリスクは増幅する。

軽減対策: すべての識別子は可変でなければならず、王人のidを更新するメカニズムでなければならない。具体的には、王人のidは、ハードウェアに固定されるべきではない。しかしながら、これは、識別子が 固定されたURIであるというLinked Data(リンクドデータ)の理想と矛盾する。多くの状況では、王人が再初期化される場合、識別子への更新が許容される。この場合、ソフトウェアエンティ ティとして、古い王人が存在しなくなり、新しい王人が作成される。これは、例えば、デバイスが新しい所有者に売られると、十分にトラッキングチェーンを遮断することができる。あるいは、デ バイスの動作状態中により頻繁な変更が望まれる場合、変更が行われたときに識別子の変更を許可されたユーザのみに通知するメカニズムを導入することができる。しかしながら、いくつかのクラ スのデバイス、例えば、医療デバイスは、いくつかの管轄区域において法律によって不変のIDを必要とすることがあるということに留意されたい。この場合、そのような不変の識別子を含むTDなど のファイルへのアクセスを保証するために特別な注意が払われるべきである。また、できる限り、そのような場合、"真に"変異不可能な識別子をTD内で共有しないことが望ましい。

9.3 指紋プライバシーリスク 5

上述したように、TD内のidメンバーは、プライバシーリスクを引き起こしうる。しかしながら、その追跡リスクを軽減するために説明の通りにidが更新されたとしても、指紋を介して、TDを特殊な物理デバイスに関連付け、そこから、指紋を使って識別可能な人にたどり着くことが依然として可能でる。

特定のデバイスインスタンスが指紋で識別できない場合、一連の対話などTD内の情報からデバイスタイプを推測し、医療状態など、識別可能な人に関する個人情報を推測するために使用することが できる。

軽減対策:

非対象があた。 許可されたユーザのみが、<u>モノ</u>のTDへのアクセスを提供されるべきである。また、許可レベルに必要な情報量と使用例のみが提供されるべきである。TDが、たとえば、認証を要求するディレクトリ サービスを通してなど安全で機密性のある経路で許可されたユーザのみに配信できるならば、外部の許可のない人たちは指紋のためのTDへのアクセス権を持たない。 このリスクをさらに軽減す るために、 TTDの特定の使用には不必要な情報は、可能な限り、省くべきである。たとえば、コンシューマがモノの状態を保存しないデバイスへの特別な 接続に関しては、idは省くことができる。コンシューマがその使用のためにある対話が必要でない場合、それを省くことができる。コンシューマが、ある対話を使用する許可を得ていない場合、そ れも省くことができる。コインシュー間が、人間が読み取り可能な情報、たとえば、タイトルや説明を表示する能力を持たない場合、省くことができる。あるいは、ゼロ長ストリングに変えること

9.4 グローバルー意識別子プライバシーリスク §

グローバルー意識別子は、第三者が識別子を知っていることになるので、一元的権限者がこれらを生成/配布する必要がある場合プライバシーリスクを呈する。

TD内のidフィールドは、意識してグローバル一意にする必要はない。一元的登録を必要としない配布方法で適切なIDを生成するために利用できる暗号メカニズムが複数存在する。これらが、通常、 同一識別子を生成する可能性は非常に低い。また、これがシステム設計で考慮する必要がある。たとえば、必要に応じ、同一のIDを検知し、IDの再生成を行う。IDの範囲も、また、グローバルであ る必要はない。あるコンテキストでモノを判別する識別子の使用が好ましい。例としては、家庭あるいは工場内。

9.5 TD傍受と改竄セキュリティリスク 5

コンテキストファイルの傍受および改竄は、ボキャブラリの解釈を変更して攻撃を容易にするために使用され得る。

コンテキストファイルは、認証されたチャネルを介してのみ取得されるということであるが、デリフェレンスされた場合、傍受および変更に対して脆弱であるHTTP URLを使って多く 理想的には、 のコンテキストが表示されるということは注目に値する(かつ、残念なことである)。しかし、コンテキストファイルが変更不能でキャッシュされ、可能な限りデリフェレンスが回避されれば、このリ スクは低減できる。

9.7 個人情報プライバシーリスクの推測 5

たくさんの場所で、ユーザのプライバシーを保護するために、個人情報、すなわち、特定の個人に関連付けることができる情報を処理するための法的要件がある。このような情報は、もちろん、IoT デバイスが直接生成することができる。しかしながら、IoTデバイスの存在及びメタデータ(TDに格納されたデータの種類)は、個人情報を持っているか、又は、推論するために使用することもでき る。この情報は、特定の個人がある種のデバスを所有しているという事実と同じくらい単純ではいてある。そして、その個人に関する追加の推論につなべめである。

軽減対策:

軽減が来た。個人デバイスに関連付けられたTDは、それが個人情報を含んでいるかのように扱う。本原則の適用例として、ユーザ同意の取得方法を考えてみてください。<u>モノ</u>が生成する個人データの使用に対す る同意は、<u>モノ</u>が、データを消費するシステムと組み合わされるときに取得されることが多く、これは、TDがデバイスにアクセスするためにローカルディレクトリまたはTDを消費するシステムに登 録されるときにも頻繁に発生する。この場合、<u>モノ</u>から送信されるデータの使用同意は、<u>モノ</u>のTDにアクセスするための同意と組み合わせることができる。第2の例として、TDが個人情報を含むと 考える場合、個人情報は無期限に保持されるべきではなく、同意が与えられた目的以外に使用されるべきでない。

10. IANAの考慮事項 §

10.1 application/td+json メディアタイプ登録 🦠

Type name: application

Subtype name:

td+ison

Required parameters: None

Optional parameters:

None

Encoding considerations:

See RFC 6839, section 3.1. Security considerations:

See RFC \$8259

Since WoT Thing Description is intended to be a pure data exchange format for Thing metadata, the serialization SHOULD NOT be passed through a code execution mechanism such as JavaScript's evail) function to be parsed. An (invalid) document may contain code that, when executed, could lead to unexpected side effects compromising the security of a system.

WoT Thing Descriptions can be evaluated with a JSON-LD 1.1 processor, which typically follows links to remote contexts (i.e., TD context extensions, see § *\vec{\phi}\$7. TD Context Extensions) automatically, resulting in the transfer of files without the explicit request of the Consumer for each one. If remote contexts are served by third parties, it may allow them to gather usage patterns or similar information leading to privacy concerns. While implementations on resource-constrained devices are expected to perform raw JSON processing (as opposed to JSON-LD processing), implementations in general SHOULD statically cache vetted versions of their supported context extensions and not to follow links to remote contexts. Supported context extensions can be managed through a secure software update mechanism instead.

Context Extensions (see § 17. TD Context Extensions) that are loaded from the Web over non-secure connections, such as HTTP, run the risk of being altered by an attacker such that they may modify the TD Information Model in a way that could compromise security. For this reason, Consumer again SHOULD vet and cache remote contexts before allowing the system to use it.

Given that JSON-LD processing usually includes the substitution of long IRIs [RFC3987] with short terms, WoT Thing Descriptions may expand considerably when processed using a JSON-LD 1.1 processor and, in the worst case, the resulting data might consume all of the recipient's resources. Consumers SHOULD treat any TD metadata with due skepticism.

Interoperability considerations:

See RFC \$825

Rules for processing both conforming and non-conforming content are defined in this specification.

Published specification:

https://w3c.github.io/wot-thing-description

Applications that use this media type:
All participating entities in the W3C Web of Things, that is, Things, Consumers, and Intermediaries as defined in the Web of Things (WoT) Architecture.

Fragment identifier considerations: See RFC 6839, section 3.1.

Additional information:

Magic number(s):

Not Applicable

File extension(s):

.jsontd Macintosh file type code(s):

Person & email address to contact for further information:

Matthias Kovatsch <w3c@kovatsch.net>

Intended usage: COMMON

Restrictions on usage:

None Author(s)

The WoT Thing Description specification is a product of the Web of Things Working Group

Change controller:

W3C

刑夕:

アプリケーション

サブ型名:

td+ison

必須パラメータ:

なし

オプションパラメータ:

なし

エンコーディングに関する考慮事項:

RFC 6839、第3.1項を参照。

セキュリティ考慮事項:

RFC 8259を参照。

WoT TDは、<u>モノ</u>のメタデータの純粋なデータ交換フォーマットのためのものであるため、シリアライズは、構文解析されるJavaScriptのeval()関数などのコード実行メカニズムをすり抜け**てはなら** ない。(無効な)文書は、実行されると、システムのセキュリティを危うくする予想外の副次的影響をもたらす可能性があるコードを含むこともある。

WoTTDはJSON-LD 1.1 プロセッサで評価することができる。JSON-LD 1.1 プロセッサは、通常、自動的にリモートコンテキスト(TDコンテキスト拡張子、第7項TDコンテキスト拡張子を参照)への リンクに従い、各<u>コンシューマ</u>からの明示的なリクエストを得ずに、ファイルが転送される。リモートコンテキストが第3者から提供されると、第3者が、プライバシーの懸念につながる使用バ ターンまたは同様の情報を収集することができるようになる。リソースが制約されたデバイス上のインプリメンテーションには、(JSON-LD処理とは対照的に)未加工のJSON処理実行が期待される が、一般的には、インプリメンテーションは、サポートされているコンテキスト拡張子の精査済みパージョンを静的にキャッシュする**べきであり**、リモートコンテキストへのリンクに従うべき**では** ない。サポートされているコンテキスト拡張子は、その代わり、安全なソフトウェア更新メカニズムで管理することができる。

HTTPなどの安全でない接続でウェブからロードされるコンテキスト拡張子(第7項TDコンテキスト拡張子を参照)には、セキュリティを危険にさらしうる方法で<u>TD情報モデル</u>を変更するように攻撃者 が変更するリスクがある。このため、<u>コンシューマ</u>は、システムがリモートコンテキストを使用できるようにする前に、再度、リモートコンテキストを精査し、キャッシュするべきである。

JSON-LD処理には、通常、長いIRI<u>[RFC3987]</u>を短い用語で置き換えるということを鑑み、WoTTDは、JSON-LD 1.1プロセッサを使用して処理されるときにかなり拡張することがあり、最悪の場合には、結果として得られるデータは、受信者のリソース全てを消費することもある。<u>コンシューマ</u>は、何らかのTDメタデータを相当の疑いを持って扱う**べきである**。

相互運用性の考慮事項:

RFC 8259を参照。

適合コンテンツと非適合コンテンツの両方を処理するための規則は、本仕様で定義される。

公開されている仕様:

https://w3c.github.io/wot-thing-description

本メディアタイプを使用するアプリケーション:

W3C WoT内の全参加エンティティ、すなわち、Web of Things (WoT)アーキテクチャで定義されているモノ、コンシューマ、および仲介者。

フラグメント識別子の考慮事項:

RFC 6839、第3.1項を参照。

追加情報

マジックナンバー:

適用外

ファイル拡張子:

.jsontd

Macintosh ファイルタイプコード:

詳細情報に関する連絡先とメールアドレス:

Matthias Kovatsch <w3c@kovatsch nets

意図されたアプリケーション:

共通

使用上の制約事項:

なし

著者:

WoTTD仕様は、Web of Things Working Groupの成果物である。

変更管理者:

W3C

10.2 CoAPコンテンツフォーマット登録 §

IANAは、Constrained RESTful Environments (CoRE)パラメータ登録 [RFC7252]内のCoAPコンテンツフォーマットのを 割り当てている。WoTTDのコンテンツフォーマットIDは、256から9999までの(t.b.d.)である(IETF レビューまたはIESG承認))。

Media Type application/td+ison

Encoding:

ID:

T.B.D. Reference:

["Web of Things (WoT) Thing Description", May 2019]

メディアタイプ:

application/td+json

エンコーディング:

ID:

TRD

[__Web of Things (WoT)Thing Description] 、2019年5月]

A. TDインスタンスの例 §

太頂は標準ではない。

A.1 CoAP プロトコルバインディングを使用したMyLampThingの例 §

モノの特徴リスト:

- タイトル: MyLampThingコンテンツ拡張子: なし

- 提供されるアフォーダンス:プロパティーつ、アクション一つ、 イベントーつ
- セキュリティ: PSKSecurityScheme (PSKセキュリティスキーム)
 プロトコルバインディング: TLSのCoAP [RFC7252]
 コメント: 第7.2項 プロトコルバインディングの追加を参照

```
Example 33: MyLampThing with CoAP Protocol Binding
      "@context": [
            https://www.w3.org/2019/wot/td/v1",
           cov": "http://www.example.org/coap-binding#
      In "id" "um:dev.ops:32473-WoTLamp-1234",
"title": "MyLampThing",
"description": "MyLampThing uses JSON serialization",
"security Definitions": {"psk_so"; {"scheme": "psk"}},
"security": ["psk_so"],
"security": ["psk_so"],
            "status": {
               "description" : "Shows the current status of the lamp",
               'Gescription': Snows the current status of the is "type": "string",
"forms": [[
"op': 'readproperty",
"hreft": "coaps://mylamp.example.com/status",
"cov:methodName": "GET"
               }]
       }]
  }

*events": {

"overheating": {

"description": "Lamp reaches a critical temperature (overheating)",

"data": {"type": "string"},

"forms": {

"href": "coaps://mylamp.example.com/oh",

"cov:methodName": "GET",

"subprotocol": "cov:observe"

}

}
```

A.2 MQTT プロトコルバインディングを使用したMyLightSensorの例 🖇

モノの特徴リスト:

- タイトル: MyLampSensor
- コンテンツ拡張子: なし
- 提供されるアフォーダンス: イベントーつセキュリティ: なし

- ・ ピューファイ・なし ・ プロトコルバインディング: MQTT MQTT ・ コメント: MQTTクライアントは、アドレス192.168.1.187:1883 の背後で実行されているMQTTブローカーによって、トピック/lightSensor に光センサデータ(数字はテキスト形式でシリアライズ されてい)を頻繁に発行する。

```
Example 34: MyLightSensor with MQTT Protocol Binding
                                   "@context": "https://www.w3.org/2019/wothd/v1",
"title": "MyLightSensor",
"id": "umrdev.ops:32473-WoTLightSensor-1234",
"securityDefinitions": {"nosec_sc": {"scheme": "nosec"}},
"security": ["nosec_sc"],
"events": {
""titletCensor": {
""titletCensor: {
""titletCens
                                                                              "lightSensor": {
                                                                                                             {
    "href": "mqtt://192.168.1.187:1883/lightSensor",
    "contentType": "text/plain"
```

A.3 Webhookイベントの例 5

<u>モノ</u>の特徴リスト:

- タイトル: WebhookThing
 コンテキスト拡張子: HTTP<u>プロトコルバインディング</u>補足(TD コンテキストにすでに含まれているhtv プレフィックス)を使用する。
- 提供されるアフォーダンス: 1イベントセキュリティ: なし

- ・セキュリティ: なし
 プロトコルパインド: HTTP
 コメント: Webhook Thingは、Webhookメカニズムを使用して、最新の温度値を定期的に<u>コンシューマ</u>にプッシュするイベントアフォーダンスtemperatureを提供し、<u>モノ</u>は、<u>コンシューマ</u>が提供するコールパックURICPOST要求を送信する。これを説明するために、subscriptionメンバーは、subscribeeventフォームを使って送信されなければならない書き込み専用パラメータ callbackURLを定義する。読み取り専用パラメータsubscriptionIDは、そのサブスクリプションが返信する。WebhookThingは、dataによって定義されたペイロードを有するこのコールパック URIに定期的にポストする。サブスクライブを解除するには、<u>コンシューマ</u>は、URIテンブレートを利用するunsubscribeeventフォームを送信しなければならない。uniVariablesメンバーは、subscriptionIDストリングを入れるように<u>コンシューマ</u>に通知する。これは、適切な意味注釈を含めるために、<u>TDコンテキスト拡張子</u>を使ってさらに自動化することができる。あるいは、subscriptionと同様にcancellationメンバーを使ったサブスクライブ解除を想定し、これを、サブスクライブ解除するためのペイロードでポスト要求を記述するunsubscribeeventフォームと組み合わせることができる。 合わせることができる。

```
Example 35: Temperature Event with subscription and cancellation
       "@context": "https://www.w3.org/2019/wot/td/v1",
      "@context": "https://www.w3.org/2019/wot/td/v1",
"id": "urndev.pos;32473-Thing-1234",
"title": "WebhookThing",
"description": "Webhook-based Event with subscription and unsubscribe form.",
"security Definitions": {"nosec_sc"; {"scheme": "nosec"}},
"security": ["nosec_sc"],
"security": ["nosec_sc"],
        "events": {
            "temperature": {
               emperature: {
    "description": "Provides periodic temperature value updates.",
    "subscription": {
        "uppe: "object",
        "properties": {
        "allibackURL": {
        "upper: "stription"

                             "type": "string",
"format": "uri",
"description": "Callback URL provided by subscriber for Webhook notifications.",
                              "writeOnly": true
                        },
"subscriptionID": {
-": "etring",
                             ubscription 1. ;
"type": "string",
"description": "Unique subscription ID for cancellation provided by WebhookThing.",
"readOnly": true
               },
'data": {
    "type": "number",
    "description": "Latest temperature value that is sent to the callback URL"
)
                 cancellation": {
                     "properties": {
                         toperuse: .{
"subscriptionID": {
"type: "integer",
"description: "Required subscription ID to cancel subscription.",
"writeOnly": true
                "uriVariables": {
                     "subscriptionID": { "type": "string" }
                      "op": "subscribeevent",
"href": "http://192.168.0.124:8080/events/temp/subscribe",
"contentType": "application/json",
"htv:methodName": "POST"
                        "op": "unsubscribeevent", "href": "http://192.168.0.124:8080/events/temp/{subscriptionID}", "htv:methodName": "DELETE"
```

B. TDインスタンス確認のためのJSONスキーマ §

本項は標準ではない。

以下は、JSONベースのフォーマットでシリアライズされたTDインスタンスを構文的に確認するためのJSONスキーマ[JSON-SCHEMA]文書である。

Note

The Thing Description defined by this document allows for adding external vocabularies by using @context mechanism known from JSON-LD [json-ld11], and the terms in those external vocabularies can be used in addition to the terms defined in §\$\frac{1}{2}\$. TD Information Model. For this reason, the below JSON schema is intentionally non-strict in that regard. You can replace the value of additional Properties schema property true with false in different scopes/levels in order to perform a stricter validation in case no external vocabularies are used.

。 注

本文書によって定義されたTDは、JSON-LD [json-id11]からわかる@contextメカニズムを使用することによって外部ボキャブラリを追加することができる。また、この外部ボキャブラリ中の用語は、第5項TD情報モデルで定義されている用語に加えて使用することができる。このため、下記JSON スキーマは意図的にその点に関して厳密には記載されていない。外部ボキャブラリが使用されていない場合により厳密な確認を実行するために、異なるスコープ/レベルでadditionalPropertiesスキーマプロパティtrue の値をfalseに置き換えることができる。

Note

Please note that some JSON Schema validation tools do not support the $\mbox{\scriptsize iri}$ string format.

· 注

一部のJSON スキーマ確認ツールでは、iriストリングフォーマットをサポートしていないということに注意する。

TDインスタンスを確認するための以下のJSONスキーマは、<u>デフォルト値</u>を持つ用語が存在することを要求しない。したがって、<u>デフォルト値</u>を持つ用語は任意選択である。(<u>第</u>5.4項デフォルト値 定義も参照)

```
"oneOf": [{
"type": "string"
                 "type": "array",
"items": {
    "type": "string"
                }
     ]
, "description": {
    "$ref": "#/definitions/description"
           },
"descriptions": {
    "$ref": "#/definitions/descriptions"
}
           "title": {
    "$ref": "#/definitions/title"
           },
"titles": {
    "$ref": "#/definitions/titles"
          Title ...

"uriVariables": {

"type": "object",

"additionalProperties": {

"$ref": "#/definitions/dataSchema"

\[
\]
         }
"forms": {
"type": "array",
"minitems": 1,
"items": 4,
"$rei": "#/definitions/form_element_property"
}
            }, "observable": {
    "type": "boolean"
           },
"writeOnly": {
  "type": "boolean"
           },
"readOnly": {
  "type": "boolean"
         "'y---

"oneOf": {

"'type": "array",

"items": {

"Sref": "#/definitions/dataSchema"
          },
"unit": {
    "type": "string"
            "enum": {
  "type": "array",
  "minItems": 1,
  "uniqueItems": true
          rormat": {
  "type": "string" },
  "type": {
  "type": {
   "type": string",
   "enum": [
   "boolean",
   "integer",
   "number",
  "string",
  "object",
  "array",
  "null"
           },
"format": {
    "type": "string"
                 ]
           },
"items": {
    "oneOf": [{
        "$ref": "#/definitions/dataSchema"
}
                 "$nc."
},
{
    "type": "array",
    "liems": {
        "$ref": "#/definitions/dataSchema"
    .
           "maxItems": {
  "type": "integer",
  "minimum": 0
           "minItems": {
  "type": "integer",
  "minimum": 0
           },
"minimum": {
  "type": "number"
           },
"maximum": {
  "type": "number"
           "additionalProperties": {
    "additionalProperties": {
        "$ref": "#/definitions/dataSchema"
        .
          "required": {
  "type": "array",
  "items": {
    "type": "string"
    "
      },
"required": [
"forms"
       ],
"additionalProperties": true
},
"descriptions": {
```

```
"$ref": "#/definitions/descriptions"
            },
"title": {
    "$ref": "#/definitions/title"
           },
"titles": {
    "$ref": "#/definitions/titles"
          "$rter..."
},
"uriVariables": {
  "type": "object",
  "additionalProperties": {
    "$ref": "#/definitions/dataSchema"
}
             },
"@type": {
    "$ref": "#/definitions/type_declaration"
}
        "$ren . },
"forms": {
   "type": "array",
   "minitems": 1,
   "items": {
        "$ref": "#/definitions/form_element_action"
        "
            }, "input": {
    "$ref": "#/definitions/dataSchema"
           },
"output": {
    "$ref": "#/definitions/dataSchema"
,
          },
"safe": {
    "type": "boolean"
           },
"idempotent": {
    "type": "boolean"
      },
"required": [
"forms"
        ],
"additionalProperties": true
"event_element": {
  "type": "object",
  "properties": {
    "description": {
        "type": "string"
        "
           'yr-'
},
"descriptions": {
    "$ref": "#/definitions/descriptions"
.
           **-
},
"title": {
    "$ref": "#/definitions/title"
            },
"titles": {
    "$ref": "#/definitions/titles"
          Tire...

}
"uriVariables": {
  "type": "object",
  "additionalProperties": {
  "$rei": "#/definitions/dataSchema"
}
            },
"@type":{
    "$ref": "#/definitions/type_declaration"
,
        "$its...
},
"forms": {
  "type": "array",
  "minitems": {
    "$ref": "#/definitions/form_element_event"
    \
    \"\hat{a} \text{The man}"
             },
"subscription": {
    "$ref": "#/definitions/dataSchema"
           },
"data": {
    "$ref": "#/definitions/dataSchema"
           },
"cancellation": {
    "$ref": "#/definitions/dataSchema"
}
          },
"type": {
    "not": {}
          },
"enum": {
"not": {}
          },
"const": {
"not": {}
      }, 
"required": [
"forms"
       ],
"additionalProperties": true
 "$re-
},
"op": {
"oneOf": [{
    "!ype": "string",
    "enum": [
    "readproperty",
    "writeproperty",
    "unobserveproperty",
    "unobserveproperty"
]
                          "type": "array",
"items": {
  "type": "string",
  "enum": [
  "readproperty",
  "writeproperty",
  "unobserveproperty",
  "unobserveproperty")
              1
            },
"contentType": {
    "type": "string"
```

```
"type": "array",
"items": {
    "type": "string"
             },
"scopes": {
  "type": "array",
  "items": {
      "type": "string"
      ,
          }
},
"subProtocol": {
  "type": "string",
  "enum": [
      "longpoll",
      "websub",
      "sse"
]
             sse"

},
"response":{
  "type": "object",
  "properties":{
  "contentType":{
  "type": "string"
  }
}
       },
"required": [
"href"
        ],
"additionalProperties": true
"adduturia...";
},
"form_element_action"; {
  "type"; "object",
  "properties"; {
  "href"; {
  "$ref": "#/definitions/anyUri"
  ",
}
        "type": "array",
"items": {
  "type": "string",
  "enum": [
        "invokeaction"
                                    ]
                ]
              },
"contentType": {
    "type": "string"
          "type": "array",
"items": {
    "type": "string"
            },
"scopes": {
  "type": "array",
  "items": {
      "type": "string"
      \
            "subProtocol": {
  "type": "string",
  "enum": [
    "longpoll",
    "websub",
    "sse"
           response": {
 "type": "object",
 "properties": {
 "contentType": {
 "contentType": {
 "type": "string"
 }
      },
"required": [
"href"
        ],
"additionalProperties": true
"op": {
"op": {
"ope": "string",
"enum": [
"subscribeevent",
"unsubscribeevent",
             "u..
]
},
{
"type": "array",
"items": {
    "type": "string",
    "enum": [
        "subscribeevent",
        "unsubscribeevent"
               }
             },
"contentType": {
    "type": "string"
          "security": {
    "type": "array",
    "items": {
        "type": "string"
    }
               }, "scopes": {
    """ "array."
```

```
type : array ;
"items": {
"type": "string"
             }
},
"subProtocol": {
  "type": "string",
  "enum": [
     "longpoll",
     "websub",
     "sse"
]
                sse"

},
"response": {
  "type": "object",
  "roontentType": {
     "type": "string"
     }
}
         }, '
"required": [
"href"
          ],
"additionalProperties": true
"adductrics..."
},
"form_element_root": {
    "type": "object",
    "properties": {
    "href": {
        "$ref": "#/definitions/anyUri"
        "$ref": "#/definitions/anyUri"
     "$re.".
},
"op":{
"oneOf":[{
"type": "string",
"enum":[
"readallproperties",
"writeallproperties",
"readmultipleproperties",
"writemultipleproperties",

1
                                   "type": "array",
"items": {
"type": "string",
"enum": [
"readallproperties",
"writeallproperties",
"writemultipleproperties",
"writemultipleproperties",
                                          ]
                    1
                },
"contentType": {
    "type": "string"
             "security": {
    "type": "array",
    "items": {
        "type": "string"
        "
              }, '
"scopes": {
  "type": "array",
  "items": {
      "type": "string"
      \
              "subProtocol": {
  "type": "string",
  "enum": [
    "longpoll",
    "websub",
    "sse"
             "response": {
  "type": "object",
  "properties": {
    "contentType": {
    "type": "string"
    }
                }
        },
"required": [
"href"
          ],
"additionalProperties": true
  },
"description": {
    "type": "string"
},
"title": {
    "type": "string"
    "ons": {
 },
"descriptions": {
    "type": "object"
},
"titles": {
    "type": "object"
    "ma": {
"Uppe": "object",
"properties": {
    "bype": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
        ...
"
               , "description": {
    "$ref": "#/definitions/description"
               **Title": {

"$ref": "#/definitions/title"
               "descriptions": {
    "$ref": "#/definitions/descriptions"
}
               **Titles": {
    "$ref": "#/definitions/titles"
    .
               },
"writeOnly": {
  "type": "boolean"
               ;,
"readOnly": {
  "type": "boolean"
                },
"oneOf": {
```

```
"type": "array",
"items": {
    "$ref": "#/definitions/dataSchema"
            }, '
"unit": {
    "type": "string"
             "type": "string"
},
"enum": {
  "type": "array",
  "minItems": 1,
  "uniqueItems": true
             },
"format": {
    "type": "string"
.
             "type": "string"
},
"const": {},
"type": {
  "type": "string",
  "enum": [
  "boolean",
  "integer",
  "number",
  "string",
  "object",
  "array",
  "null"
}
                   ]
            , "maxItems": {
  "type": "integer",
  "minimum": 0
            , "minItems": {
  "type": "integer",
  "minimum": 0
             },
"minimum": {
  "type": "number"

           },
"maximum": {
  "type": "number"
            },
"properties": {
    "additionalProperties": {
        "$ref": "#/definitions/dataSchema"
}
           ;
"required": {
"type": "array",
"items": {
"type": "string"
}
},
"link_element": {
  "type": "object",
  "properties": {
    "anchor": {
        "$ref": "#/definitions/anyUri"
        .
}
           ***:

},

"href": {

    "$ref": "#/definitions/anyUri"

}
           },
"rel": {
    "type": "string"
           },
"type": {
    "type": "string"
       },
"required": [
"href"
        ],
"additionalProperties": true
"aduno..."

},

"securityScheme": {
  "oneOf": [{
        "ype": "object",
        "properties": {
        "@type": "#/definitions/type_declaration"
        \
            \"*fef": "#/definitions/type_declaration"
                      , "description": {
    "$ref": "#/definitions/description"
}
                      "descriptions": {
    "$ref": "#/definitions/descriptions"
}
                      },
"proxy": {
    "$ref": "#/definitions/anyUri"
                    "scheme": {

"type": "string",

"enum": [

"nosec"
                            ]
                },
"required": [
"scheme"
                   "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
                      , "description": {
    "$ref": "#/definitions/description"
                      "descriptions": {
    "$ref": "#/definitions/descriptions"
}
                      **Proxy": {

"$ref": "#/definitions/anyUri"
```

```
"scheme": {
  "type": "string",
  "enum": [
    "basic"
            ]
            n": {
"type": "string",
"enum": [
    "header",
    "query",
    "body",
    "cookie"
           ]
    },
"name": {
    "type": "string"
;
"required": [
"scheme"
  "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
     , "description": {
    "$ref": "#/definitions/description"
     and...
};
"descriptions": {
    "$ref": "#/definitions/descriptions"
}
     },
"proxy": {
    "$ref": "#/definitions/anyUri"
}
    ***.

},

"scheme": {

"type": "string",

"enum": [

"cert"
           ]
    },
"identity": {
    "type": "string"
}, 'required": [
   "scheme"
  ]
 "type": "object",
"properties": {
    "@type": {
        "sref": "#/definitions/type_declaration"
        .
     },
"description": {
    "$ret": "#/definitions/description"
,
     , "descriptions": {
    "$ref": "#/definitions/descriptions"
     },
"proxy": {
    "$ref": "#/definitions/anyUri"
}
    "scheme": {
  "type": "string",
  "enum": [
  "digest"
            ]
 ,

"qop": {

"type": "string",

"enum": [

"auth",

"auth-int"
            ]
    },
"name": {
    "type": "string"
}, '
"required": [
"scheme"
  "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration" |
     "descriptions": {
    "$ref": "#/definitions/descriptions"
}
     "proxy": {
    "$ref": "#/definitions/anyUri"
    "scheme": {
  "type": "string",
  "enum": [
  "bearer"
      },
"authorization": {
    "$ref": "#/definitions/anyUri"
      "$ref": "#/definiti
},
"alg": {
"type": "string",
"enum": [
"MD5",
"ES256",
"ES512-256"
    },
"format": {
  "type": "string",
```

```
"enum": [
   "jwt",
   "jwe",
   "jws"
   ]
},
"in": {
"type": "string",
"enum": [
"header",
"query",
"body",
"cookie"
            },
"name": {
    "type": "string"
 }, "required": [
   "scheme"
      "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
             }, "description": {
    "sref": "#/definitions/description"
             },
"descriptions": {
    "$ref": "#/definitions/descriptions"
}
             "proxy": {
    "$ref": "#/definitions/anyUri"
               "scheme": {

"scheme": {

"type": "string",

"enum": [

"psk"

]

}
            },
"identity": {
  "type": "string"
 }, "required": [
   "scheme"
      "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
        "
             },
"description": {
    "$ref": "#/definitions/description"
             and...
];
"descriptions": {
    "$ref": "#/definitions/descriptions"
             },
"proxy": {
    "$ref": "#/definitions/anyUri"
          "scheme": {

"type": "string",

"enum": [

"public"
                             ]
            },
"identity": {
    "type": "string"
 },
"required": [
"scheme"
    "type": "object",
"properties": {
    "@type": {
        "sref": "#/definitions/type_declaration"
        "
             }, "description": {
    "sref": "#/definitions/description"
             and the state of t
             },
"proxy": {
    "$ref": "#/definitions/anyUri"
         *"--.
},
"scheme": {
  "type": "string",
  "enum": [
      "oauth2"
             ,
"authorization": {
"$ref": "#/definitions/anyUri"
,
             "token": {
    "$ref": "#/definitions/anyUri"
             "refresh": {
    "$ref": "#/definitions/anyUri"
          "scopes": {
    "type": "array",
    "items": {
        "type": "string"
    "
      }
},
"flow": {
"type": "string",
"enum": [
"implicit",
"password",
"client",
"code"
         }
"required": [
"scheme",
"flow"
```

```
"type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration"
                        }, "description": {
    "$ref": "#/definitions/description"
                        },
"descriptions": {
    "$ref": "#/definitions/descriptions"
,
                        },
"proxy": {
    "$ref": "#/definitions/anyUri"
                      ***.
},
"scheme": {
    "type": "string",
    "enum": [
        "apikey"
                 ]

"in": {

"type": "string",

"enum": [

"header",

"query",

"body",

"cookie"
                       }, "name": {
    "type": "string"
                  },
"required": [
"scheme"
                     "type": "object",
"properties": {
    "@type": {
        "$ref": "#/definitions/type_declaration'
                        },
"description": {
    ""$ref": "#/definitions/description"
                        , "descriptions": {
    "$ref": "#/definitions/descriptions"
                        },
"proxy": {
    "$ref": "#/definitions/anyUri"
                      "scheme": {
    "type": "string",
    "enum": [
        "pop"
                        , "authorization": {
    "$ref": "#/definitions/anyUri"
    1
                     "format": {

"type": "string",

"enum": [

"jwt",

"jwe",

"jws"

1
                         ]
},
"alg": {
"type": "string",
"enum": [
"MD5",
"ES256",
"ES512-256"
                  ]
},
"in": {
"type": "string",
"enum": [
"header",
"query",
"body",
"cookie"
                       },
"name": {
    "type": "string"
                   },
"required": [
"scheme"
        ]
  anyUri": {

"anyUri": {

"type": "string",

"format": "iri-reference"
},
"type": "object",
"properties": {
    "id": {
        "type": "string",
        "format": "uri"
}
   },
"title": {
    "$ref": "#/definitions/title"
  },
"titles": {
    "$ref": "#/definitions/titles"

  },
"actions": {
    "type": "object",
    "additionalProperties": {
        "$ref": "#/definitions/action_element"
    ,
}
 },
"events":{
"type": "object",
"additionalProperties":{
"$ref": "#/definitions/event element"
```

```
"description": {
    "$ref": "#/definitions/description":
    descriptions:: {
        "$ref": "#/definitions/descriptions'
    },
"version": {

"type": "object",
"properties": {

"instance": {

"type": "string"
          }
        "required": |
       ]
},
"links": {
    "type": "array",
    "items": {
        "$ref": "#/de
            "$ref": "#/definitions/link element"
       }
 },
"forms": {
    "type": "array",
    "minItems": 1,
    "": "ms": {
            "$ref": "#/definitions/form element root"
       }
   },
"base": {
    "$ref": "#/definitions/anyUri"
    "securityDefinitions": {
        "type": "object"
       "minProperties": 1,
"additionalProperties": {
    "$ref": "#/definitions/securitySche
 },
"support": {
    "$ref": "#/definitions/anyUri"
 },
"created": {
  "type": "string"
    "modified": {
    "type": "string"
    "security": {
  "type": "array",
  "minItems": 1,
  "items": {
             "type": "string
       }
  },
"@type": {
    "$ref": "#/definitions/type_declaration"
 },
"@context": {
    ""$ref": "#/definitions/thing-context".
"required": [
    "title"
    "security",

"securityDefinitions",

"@context"
"additionalProperties": true
```

C. TDテンプレート §

本項は標準ではない。

TDテンプレートは、<u>モノ</u>のクラスの記述であり、クラウドサーバによる何千ものデバイスの共通処理を可能にするために、<u>モノ</u>のグループ全体で共有されるプロバティ、アクション、イベント、および共通メタデータを記述しており、<u>モノ</u>それぞれで利用するものではない。TDテンプレートは、<u>第5項TD情報モデル</u>からの同じ中核ボキャブラリおよび情報モデルを使用する。

TDテンプレートを使って次のことが可能になる。

- クラウドサービスによる複数の<u>モノ</u>の管理
- ・ まだ開発されていないデバイス(<u>下</u>)のシミュレーション・ 共通の<u>干</u>ノモデルを共有する異なるメーカーのデバイス間で共通のアプリ
- 複数のモデルを<u>モノ</u>と組み合わせる

TDテンプレートは、デバイスとのインターフェース及び可能な対話(プロパティ、アクション、イベント)の論理的記述であるが、シリアル番号、GPS位置、セキュリティ情報、具体的なプロトコル エンドポイントなどデバイス固有の情報を含まない。

TDテンプレートは、特定のエンドポイントへの<u>プロトコルバインディング</u>を含まず、特定のセキュリティメカニズムを定義していないので、フォームおよびセキュリティ定義とセキュリティキーは 存在してはならない。

同じTDテンプレートは、複数のベンダの複数のモノによって実装することができる。モノは複数のTDテンプレートを実装し、追加のメタデータ(ベンダ、位置、セキュリティ)を定義し、具体的なプロトコルへのバインディングを定義することができる。共通のモノに組み合わされている異なるTDテンプレートからのプロパティ、アクション、イベントの間の衝突を回避するために、これらの識 別子はすべて、 $\pm \angle$ の中で唯一無二でなければならない。

あるクラスのデバイスの共通のTDテンプレートは、ベンダを超えてアプリを書くことを可能にし、アプリ開発者にとってより魅力的な市場を生み出す。具体的なTDは、複数のTDテンプレートを実 装することができ、したがって、機能ブロックを結合されたデバイスに集約することができる。

クラウドベンダーのビジネスモデルは、通常、何千もの同一のデバイス管理することで構築される。同一TDテンプレートのデバイスすべては、クラウドアプリにより同じ方法で管理することができる。インターフェースとインスタンスが別々に扱われる場合、複数のシミュレートされたデバイスを作成することは容易である。

しかし、TDテンプレートは、いくつかの任意のボキャブラリ用語と必須の<u>ボキャブラリ用語</u>が存在しないTDのサブセットであるので、TDと同じ方法・同じフォーマットでシリアライズすることが できる。TDテンプレートインスタンスは、いくつかの必須用語が欠落しているため、TDインスタンスと同じ方法で確認することができないことに留意されたい。

C.1 TDテンプレートの例 §

本項では、ランプのTDテンプレートとブザーのTDテンプレートを紹介する。

C.1.1 TDテンプレート: ランプ 』

```
Example 36: MyLampThingTemplate serialized in JSON
                        "@context": ["https://www.w3.org/2019/wot/td/v1"],
                   "@context: [ ntgs:/www.w3.org/2019/wot/a/v1 ],
"@type": "ThingTemplate",
"title": "Lamp Thing Description Template",
"description": "Lamp Thing Description Template",
"properties:"
"status": {
"stat
                                                       "description" : "current status of the lamp (on|off)",
                                                       "type"
                                                       "readOnly": true
                        "actions": {
                                         "toggle":
                                                       "description" : "Turn the lamp on or off
                        "events": {
                                     "overheating": {
    "description" : "Lamp reaches a critical temperature (overheating)",
    "data": {"type": "string"}
```

C.1.2 TDテンプレート: ブザー 🜗

```
Example 37: MyBuzzerThingTemplate serialized in JSON
    "@context": ["https://www.w3.org/2019/wot/td/v1"],
     actions": {
"buzz": {
        "description": "buzz for 10 seconds"
```

D. JSON-LDコンテキスト用法 §

本項は標準ではない。

現行の仕様書は、<u>TD情報モデル</u>を異なる<u>ボキャブラリ</u>に関する一連の制約、すなわち、<u>ボキャブラリ用語</u>として取り入れている。本項では、これらの制約の機会読み取り可能定義が、TDドキュメン トの必須@contextを使用して以下にクライアントアプリに結合されたのかを簡単に説明する。

TDドキュメントからTD情報モデルへのアクセスは、2段階で行われる。まず、クライアントは、IRIへのJSONストリングからのマッピングを検索する。気尾のマッピングは、後述のとおりJSON-IDコインと INDICATE CAPE NO. 2 CA No. A EXPIRICITABLES ます、フライアントは、IHIへWJSUNAトリンクからのマッピンクを検索する。気尾のマッピングは、後述のとおりJSON-LDコインテキストを定義されている。そこで、クライアントは、それらをデレファランスしてIRI上で定義されている制約にアクセスすることができる。制約はクライアントプログラムが簡単に解読できるIRDF形式で論理原理として定義されている。

第5項TD情報モデル内で参照されているボキャ <u>ブラリ用語</u>はすべて、TDドキュメント内で(コンパクトな)JSONストリングとしてシリアライズされている。しかしながら、これら用語のはそれぞ お、第一のLINKED Data原理 (LINKED-DATA) どおり、完全なINIで降に識別できる。JSONキーからIRIへのマッピングは、TDの@context値が示すものである。たとえば、https://www.w3.org/2019/wot/td/v1のファイルは、以下のマッピングを含んでいる(さらに存在する中で):

```
properties → https://www.w3.org/2019/wot/td#hasPropertyAffordance
object → https://www.w3.org/2019/wot/json-schema#ObjectSchema
basic
        → https://www.w3.org/2019/wot/security#BasicSecurityScheme
href
         → https://www.w3.org/2019/wot/hypermedia#hasTarget
```

このJSONファイryはJSON-LD 1.1シンテックス[<u>JSON-LD11</u>]に続くものである。多数のJSON-LDライブラリが自動的にTDの@contextを処理し、その中のJSONストリングすべてを拡張することが できる。

TDの<u>ボキャブラリ用語</u>すべてがIRIに拡張されると、第二ステップは、その<u>ボキャブラリ用語</u>を参照する<u>TD情報</u>も出るのフラグメントを取得するためにこのIRIをデレファランスする。たとえば、

https://www.w3.org/2019/wot/json-schema#ObjectSchema</mark>デレファランスすると、ObjectSchemaという用語は<u>クラス</u>であり、もっと正確にするとDataSchemaのサブクラスであるというRDFドキュ メントとなる。このような理論原理は、さまざまな複雑な形式を使ったRDFで表現される。サブクラス関係は、RDF Schema原理「<u>RDF-SCHEMA</u>」として表現される。さらに、これら原理は、さ ま座万形式でシリアライズされることがある。ここでは、Turtle形式 <u>(TURTL)</u> でシリアライズされている。

```
<a href="https://www.w3.org/2019/wot/json-schema#ObjectSchema">https://www.w3.org/2019/wot/json-schema#ObjectSchema</a>
<a href="https://www.w3.org/2019/wot/json-schema#ObjectSchema">https://www.w3.org/2019/wot/json-schema#ObjectSchema</a>
   rdfs:subClassOf <https://www.w3.org/2019/wot/json-schema#DataSchema>
```

·で、ユーザエージェントが内容交渉をしない場合、人間が読み取り可能なHTML資料がRDFドキュメントの代わりに返される。内容を交渉するためには、クライアントは、自身のリクエ スト内にHTTPヘッダーAccept: text/turtleを入れなければならない。

E. 最近の仕様変更 §

E.1 第1次勧告候補からの変更点 5

- 一般
 - 。 <u>TDプロセッサ</u>との明確化された定義と、すべてのインプリメンテーションが正確に想定するように、インプリメンテーションはTDプロデューサあるいはTDコンシューマであるという関
 - 連テキスト(インブリメンテーションに影響なし) モノidの明確にされた一意性と可能な値。特に、ローカルIPアドレスを持つURIあるいは一時的で変異するURIのようなプライバシー保護する値

 - 新されたレファランス味論注釈の見直しテキスト
 - 。 バグ修正のために更新された<mark>例</mark>2
 - nameは、第5.3.1.1Thingクラス定義への前回変更を反映するためにtitleへ改名された。
 @typeは、formsから改名された。
 - 。 バグ修正のために更新された例33
 - opは、デフォルト値を無効にするためにstatusプロパティのformに追加された。
 - ・ ["readproperty", "writeproperty"]のデフォルト値は両方がGET法にしてしまったであろう。ここは意図的ではなかった。
- 小さな編集上の見直しと修正
- 用語法
 - 。 <u>第3項用語法</u>標準ではない。
- TD情報モデル
 - 1870年 とグル ・ <u>第5.3.1.1Thing</u>で、<u>モノ</u>の<u>ボキャブラリ用語</u>idは、オプションとなった。また、idの一意性の表現は、必要がないため削除された。

 - 第23.5.1 TIMINg い 2...
 図1のラベル位置調整
 アサーション・強調部分でない大文字MAYを修正。(データスキーマのTDコンテキスト拡張子;インプリメンテーションに影響なし)
 図2"JSONスキーマボキャブラリ"のタイトルは、"データスキーマボキャブラリ"に変更。
 - 。図2"JSONスキーマボキャブラリ"のタイトルは、"データスキーマボキャブラリ"に変更。 <u>OAuth2認証キュリティ構成のボキャブラリ用語</u>は、現在、デフォルト値はない。よって、<u>第5.4項デフォルト値定義</u>から削除された。
- TD表現形式
- ・ TDコンテキスト拡張子
- - 。見直しされた例28と例29
 - モノを拡張クラスで意味論的に注釈するためにsaref:TemperatureSensorを使用する。

- 例中で整合性を取るためにプロパティを注釈するためにSAREFを使用する。
- より新しいontology-of-units-of measure(om)を使用する
 <u>例2</u>との設計上の違いを明確にするために<u>例29</u>を更新した。

- ビヘイビアアサーション
 第8.2項Data Schemasはデータスキーマに説明されていないにもかかわらず、モノが追加データを返し、コンシューマが追加データを受け取らなければならないときの詳細な状況を説明 している。
 - 第8.3.1項HTTPベースのプロトコルバインディングは、"HTTPプロトコルバインディング" から改名された。
- セキュリティとブライバシーに関する考慮事項

 一意でなくオプションのモノのボキャブラリ用語はと整合させるために第9項セキュリティとプライバシーに関する考慮事項を更新。
 使用例に必要のないときに口をフィルタリングするといった軽減対策を追加。
- デバイスタイプを特徴で識別するといったリスクを追加。 グローバル一意識別子リスクを追加し、これが不要であり、配布された暗号メカニズムあるいはローカルな範囲内のidを使用することができるということを指摘している。
- 付録
 - 付録C TDテンプレートは、"モノテンプレート"から改名。
 - <u>「付録D JOON-LDコンテキスト用法</u>は、RDF内のTD情報検索のための2段階ステップを説明している。 WoT Architecture仕様 [<u>WOT-ARCHITECTURE</u>] 書参照は、有益である。

E.2 第3次公開草案からの変更点 5

第3次公開作業ドラフトからの変更点は、Candidate Recommendation推奨候補内で説明されている。

F.謝辞 §

編集者は、御寄稿、ガイダンスおよび専門知識を御提供頂いたMichael Koster, Michael Lagally, Kazuyuki Ashimura, Ege Korkan, Daniel Peintner, Toru Kawaguchi, María Poveda, Dave Raggeti Kunihiko Toumura, Takeshi Yamada, Ben Francis, Manu Sporny, Klaus Hartke, Addison Phillips, Jose M. Cantera, Tomoaki Mizushima, Soumya Kanti Datta and Benjamin Klotzに謝意を表します。

また、本文書改良を可能にしたサポート、技術的入力、および提案を頂いたW3Cスタッフ及びW3C WoTインタレストグループ (WoT IG)およびワーキンググループ (WoT WG)の他のすべての現役お よび前参加者に対し謝意を表します。

G. References §

G.1 Normative references 5

[BCP47]

Tags for Identifying Languages. A. Phillips; M. Davis. IETF. September 2009. IETF Best Current Practice. URL: https://tools.ietf.org/html/bcp47

[eventsource]

Server-Sent Events. Ian Hickson. W3C. 3 February 2015. W3C Recommendation. URL: https://www.w3.org/TR/eventsource/ [RFC2046]

Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed; N. Borenstein. IETF. November 1996. Draft Standard. URL: https://tools.ietf.org/html/rfc2046

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997. Best Current Practice. URL: https://tools.ietf.org/html/rfc2119

[RFC3339]

Date and Time on the Internet: Timestamps. G. Klyne; C. Newman. IETF. July 2002. Proposed Standard. URL: https://tools.ietf.org/html/rfc3339 [RFC3629]

<u>sformation format of ISO 10646</u>. F. Yergeau. IETF. November 2003. Internet Standard. URL: https://tools.ietf.org/html/rfc3629

[RFC3986]

ource Identifier (URI): Generic Syntax. T. Berners-Lee; R. Fielding; L. Masinter. IETF. January 2005. Internet Standard. URL: https://tools.ietf.org/html/rfc3986 [RFC3987] d Resource Identifiers (IRIs). M. Duerst; M. Suignard. IETF. January 2005. Proposed Standard. URL: https://tools.ietf.org/html/rfc3987

IRFC65701

te. J. Gregorio; R. Fielding; M. Hadley; M. Nottingham; D. Orchard. IETF. March 2012. Proposed Standard. URL: https://tools.ietf.org/html/rfc6570

[BEC6749]

The OAuth 2.0 Authorization Framework. D. Hardt, Ed., IETF. October 2012. Proposed Standard. URL: https://tools.ietf.org/html/rfc6749 [RFC6750]

The OAuth 2.0 Authorization Framework: Bearer Token Usage. M. Jones; D. Hardt. IETF. October 2012. Proposed Standard. URL: https://tools.ietf.org/html/rfc6750

[RFC7252] The Constrained Application Protocol (CoAP). Z. Shelby; K. Hartke; C. Bormann. IETF. June 2014. Proposed Standard. URL: https://tools.ietf.org/html/rfc7252

[RFC7516] JSON Web Encryption (JWE). M. Jones; J. Hildebrand. IETF. May 2015. Proposed Standard. URL: https://tools.ietf.org/html/rfc7516

[RFC7519]

JSON Web Token (JWT). M. Jones; J. Bradley; N. Sakimura. IETF. May 2015. Proposed Standard. URL: https://tools.ietf.org/html/rfc7519

[RFC7616]

HTTP Diaest Access Authentication. R. Shekh-Yusef, Ed.; D. Ahrens; S. Bremer. IETF. September 2015. Proposed Standard. URL: https://httpwg.org/specs/rfc7616.html

The 'Basic' HTTP Authentication Scheme. J. Reschke. IETF. September 2015. Proposed Standard. URL: https://httpwg.org/specs/rfc7617.html [RFC7797]

JSON Web Signature (JWS) Unencoded Payload Option. M. Jones. IETF. February 2016. Proposed Standard. URL: https://tools.ietf.org/html/rfc7797

[RFC8174] Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. B. Leiba. IETF. May 2017. Best Current Practice. URL: https://tools.ietf.org/html/rfc8174 [RFC8252]

OAuth 2.0 for Native Apps. W. Denniss; J. Bradley. IETF. October 2017. Best Current Practice. URL: https://tools.ietf.org/html/rfc8252

[RFC8259]

The JavaScript Object Notation (JSON) Data Interchange Format. T. Bray, Ed. IETF. December 2017. Internet Standard. URL: https://tools.ietf.org/html/rfc8259

[RFC8288]

Web Linking. M. Nottingham. IETF. October 2017. Proposed Standard. URL: https://httpwg.org/specs/rfc8288.html [RFC8392]

CBOR Web Token (CWT). M. Jones; E. Wahlstroem; S. Erdtman; H. Tschofenig. IETF. May 2018. Proposed Standard. URL: https://tools.ietf.org/html/rfc8392

[websub]

WebSub. Julien Genestoux; Aaron Parecki. W3C. 23 January 2018. W3C Recommendation. URL: https://www.w3.org/TR/websub/ [X509V3]

TTU-T Recommendation X.509 version 3 (1997). "Information Technology - Open Systems Interconnection - The Directory Authentication Framework" VISO/IEC 9594-8:1997.. ITU.

[XMLSCHEMA11-2-20120405]

W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. David Peterson; Sandy Gao; Ashok Malhotra; Michael Sperberg-McQueen; Henry Thompson; Paul V. Biron et al. W3C. 5 April 2012. W3C Recommendation. URL: https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405

G.2 Informative references §

[ACE]

ntication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth). L. Seitz; G. Selander; E. Wahlstroem; S. Erdtman; H. Tschofenig. IETF. 27 March Authentication and Authorization for Constrained Environments (ACE) using the 2019. Internet-Draft. URL: https://tools.ietf.org/html/draft-ietf-ace-oauth-authz-24

[HTTP-in-RDF10]

HTTP Vocabulary in RDF 1.0. Johannes Koch; Carlos A. Velasco; Philip Ackermann. W3C. 2 February 2017. W3C Note. URL: https://www.w3.org/TR/HTTP-in-RDF10/

[IANA-MEDIA-TYPES]

Media Types. IANA. URL: https://www.iana.org/assignments/media-types/[IANA-URI-SCHEMES]

. Uniform Resource Identifier (URI) Schemes. IANA. URL: https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml [JSON-LD11]

SON-LD 1.1. Gregg Kellogg; Pierre-Antoine Champin. W3C. 9 September 2019. W3C Working Draft. URL: https://www.w3.org/TR/json-ld11/

[JSON-SCHEMA]

JSON Schema Validation: A Vocabulary for Structural Validation of JSON. Austin Wright; Henry Andrews; Geraint Luft. IETF. 19 March 2018. Internet-Draft. URL: https://tools.ietf.org/html/drafthandrews-ison-schema-validation-01 [LDML]

Unicode Technical Standard #35: Unicode Locale Data Markup Language (LDML), Mark Davis: CLDR Contributors, URL: https://unicode.org/reports/tr35/ [LINKED-DATA]

Linked Data Design Issues. Tim Berners-Lee. W3C. 27 July 2006. W3C-Internal Document. URL: https://www.w3.org/DesignIssues/LinkedData.html

[MQTT]

MOTT Version 3.1.1, Andrew Banks; Rahul Gupta, OASIS, 10 December 2015, OASIS Standard Incorporating Approved Errata 01, URL; http://docs.oasis-open.org/mgtt/mgtt/v3.1.1/mgtt-v3.1.1.html [OPENAPI]

OpenAPI Sr. cification: Version 3.0.1. Darrel Miller: Jason Harmon: Jeremy Whitlock; Kris Hahn: Marsh Gardiner; Mike Ralphson: Rob Dolin: Ron Ratovsky; Tony Tam, OpenAPI Initiative, Linux Foundation. 7 December 2017. URL: https://swagger.io/specification/

[RDF-SCHEMA]

RDF Schema 1.1. Dan Brickley; Ramanathan Guha. W3C. 25 February 2014. W3C Recommendation. URL: https://www.w3.org/TR/rdf-schema/

[RFC3966]

The tel URI for Telephone Numbers. H. Schulzrinne. IETF. December 2004. Proposed Standard. URL: https://tools.ietf.org/html/rfc3966

[RFC6068]

The 'mailto' URI Scheme. M. Duerst; L. Masinter; J. Zawinski. IETF. October 2010. Proposed Standard. URL: https://tools.ietf.org/html/rfc6068

Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. R. Fielding, Ed.; J. Reschke, Ed., IETF. June 2014. Proposed Standard. URL: https://httpwg.org/specs/rfc7231.html [RIJGERSBERG]

Ontology of Units of Measure and Related Concepts. Hajo Rijgersberg; Mark van Assem; Jan Top. Semantic Web journal, IOS Press. 2013. URL: http://www.semantic-web-journal.net/content/ontology-units-measure-and-related-concepts

[SEMVER]

Semantic Versioning 2.0.0. Tom Preston-Werner. 26 December 2017. URL: https://semver.org/

[SMARTM2M]

ETSI TS 103 264 V2.1.1 (2017-03): SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping. ETSI. March 2017. Published. URL:

http://www.etsi.org/deliver/etsi_ts/103200_103299/103264/02.01.01_60/ts_103264v020101p.pdf

[string-meta]

Strings on the Web: Language and Direction Metadata. Addison Phillips; Richard Ishida. W3C. 11 June 2019. W3C Working Draft. URL: https://www.w3.org/TR/string-meta/[TURTLE]

RDF 1.1 Turtle. Eric Prud'hommeaux; Gavin Carothers. W3C. 25 February 2014. W3C Recommendation. URL: https://www.w3.org/TR/turtle/

[VOCAB-SSN]

Semantic Sensor Network Ontology. Armin Haller; Krzysztof Janowicz; Simon Cox; Danh Le Phuoc; Kerry Taylor; Maxime Lefrançois. W3C. 19 October 2017. W3C Recommendation. URL: https://www.w3.org/TR/vocab-ssn/ [WOT-ARCHITECTURE]

Web of Things (WoT) Architecture. Matthias Kovatsch; Ryuichi Matsukura; Michael Lagally; Toru Kawaguchi; Kunihiko Toumura; Kazuo Kajimoto. W3C Candidate Recommendation. 6 November 2019. URL: https://www.w3.org/TR/wot-architecture/

[WOT-BINDING-TEMPLATES]

Web of Things (WoT) Protocol Binding Templates. Michael Koster. W3C. 5 April 2018. W3C Note. URL: https://www.w3.org/TR/wot-binding-templates/

[WOT-SECURITY-GUIDELINES]

Web of Things (WoT) Security and Privacy Guidelines. Michael McCool; Elena Reshetova. W3C Note. 6 November 2019. URL: https://www.w3.org/TR/wot-security/

İ