Technology Reports

LwM2M　オープンAPI　IoTデバイス管理

IoTデバイスの遠隔制御を実現する汎用プラットフォーム開発

ネットワーク開発部　岡村 健司　山門 彩
黒川 祐一郎

近年，IoTデバイスやIoT関連サービスが普及するにつれ，膨大な数のIoTデバイスの制御・メンテナンスの効率化が課題となっている．例えば，IoTデバイスを管理するシステムのUIがブラウザを使ったWeb UIのみで，多数のIoTデバイスに一括で処理を投入する場合，ユーザは1つひとつブラウザで手作業することが必要であり，非効率的であるため，任意の一括処理を容易に実行可能な仕組み作りが課題である．そこで，IoTデバイスの遠隔管理・制御の操作性や利便性の向上を目的として，「オープンAPI」と「Web UI」を提供するドコモIoTデバイス管理プラットフォームを開発した．これにより，ユーザ環境からプラットフォームへの接続手段やユーザが利用するIoTデバイスの選択肢が増え，IoTデバイスとの接続容易性が向上した．

1. まえがき

ドコモでは以前よりAWS（Amazon Web Services）*1上でIoTデバイス管理プラットフォーム（以下，旧システム）を提供してきた．このシステムではIoTデバイスを，Webブラウザを使いアクセスするWeb UI*2から管理でき，IoTデバイスのファームウェアアップデートも可能であった．しかし，IoTデバイスの需要増加に対応できるシステムのスケーラビリティやWeb UI以外からの管理，IoTデバイスのさらなる省電力化が求められるようになり，このためドコモは既存システムの機能を流用しつつ，それぞれのニーズに応えた新たなシステムを開発した．

また，システムの実現に際し旧システムの構成を，IoT標準仕様であるOMA-LwM2M（Open Mobile Alliance Lightweight Machine to Machine）*3プロトコルを用いてIoTデバイスと通信を行うIoTデバイス収容ノード（vDME（virtualized Device Management Equipment）*4）と，IoTデバイスを管理，制御するインタフェースであるサービス制御ノード（DSP（Device Service Platform）*5）に分離した事で，将来の機能追加やサービスの拡張を柔軟に行え

＊1　AWS：Amazon Web Services社が提供するクラウドコンピューティングサービス．
＊2　Web UI：Webブラウザ上でユーザとコンピュータとの間で情報をやり取りする際の操作画面や操作方法．
＊3　OMA-LwM2M：Open Mobile Allianceによって，IoTデバイス管理を目的として標準規定された通信プロトコル．
＊4　vDME：IoTデバイスを含むさまざまな種類の通信端末を収容し，制御するノード．仮想化基盤上に構築されている．

るようにした．

本稿では，開発したシステムの全体構成，スケーラビリティの実現，vDMEとDSPの機能分担，新たに開発したオープンAPI（Application Programming Interface）*6，および省電力化を実現するための工夫点について解説する．

2. IoTデバイス管理プラットフォームの開発経緯および全体像

ドコモが提供してきた旧システムでは，Web UIをドコモが提供し，ユーザはそこでIoTデバイスへの制御指示やIoTデバイスの管理が可能である．しかし，近年のIoTデバイス数増加に伴い，旧システムでのIoTデバイス収容数の逼迫が課題であった．さらに，機能的な面についても旧システムでは不足しており，新たなニーズへ対応するために機能追加する必要があった．そこで，IoTデバイス収容数の柔軟な拡大を実現するため，ドコモの仮想化技術を使ったプラットフォーム上で動作可能とするシステムを開発し，機能追加についても対応を行った．今回開発したシステムを「ドコモIoTデバイス管理プラットフォーム」（以下，本プラットフォーム）と呼ぶ．

本プラットフォームの開発に際し，旧システムの機能やプログラムを流用することで開発効率を上げ，機能拡充・IoTデバイス収容数の拡大・IoTデバイスとのOMA-LwM2Mプロトコルのパラメータに着目した制御による省電力化を行った．

旧システムと本プラットフォームの全体構成の比較を**図1**に示す．本プラットフォームのシステム構成を，IoTデバイスの管理，制御を行うWeb UIを

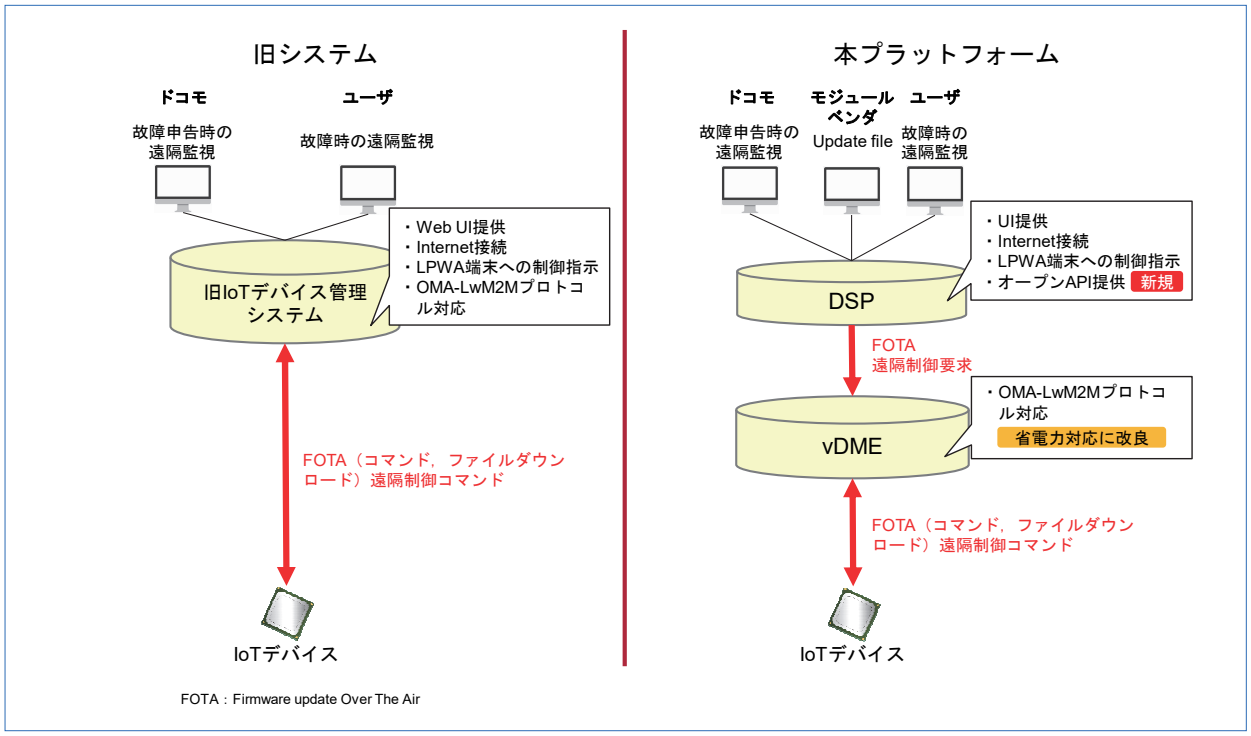

図1　旧システムと本プラットフォームの比較

*5　DSP：IoTデバイスを含むさまざまな種類の通信端末を管理，制御するためのUIを提供するノード．

*6　オープンAPI：アプリケーションから特定のサービスを利用するためにユーザへ公開（オープン）するインタフェース．本稿では，特にRESTful APIを指す．

提供するDSPと，IoTサービスモデルをベースとしたvDMEに機能分担した．DSPは，IoTデバイスを管理，制御するためのWeb UIや同様の管理，制御が可能なオープンAPIを提供する．vDMEは，IoTデバイスとのOMA-LwM2Mのアプリケーション層プロトコルであるCoAP（Constrained Application Protocol）*7やHTTP（HyperText Transfer Protocol）*8/HTTPS（HTTP Secure）*9通信を終端し，IoTデバイスのファームウェアを管理するファイルサーバの役割を担う．本開発により，Web UI（Internet SSL（Secure Socket Layer）*10）とオープンAPIの二通りの方式で本プラットフォームの利用が可能となった．また，ユーザPC・ユーザシステムとの間の通信は認証や暗号化技術を用いてセキュリティを確保した．

本プラットフォームが具備している機能をカテゴリごとに図示した機能マップを**図2**に示す．

3. 収容デバイス増加のための対応

プラットフォームに収容されるIoTデバイスの数は今後も増加が見込まれ，それに対して本プラットフォームの設備増設で対応する必要がある．

そこで，ドコモの仮想化基盤上にプラットフォームを構築することで，需要の増加に対して柔軟にシステム拡張できるスケーリング*11を可能とした．ドコモの仮想化基盤はNFV（Network Functions Virtualisation）基盤*12を用いることで，スケーリングの際に効率化された最小手順で対応できる．さらに，仮想化技術であるオートヒーリング*13機能により，障害時の安定したサービス継続も可能となる．

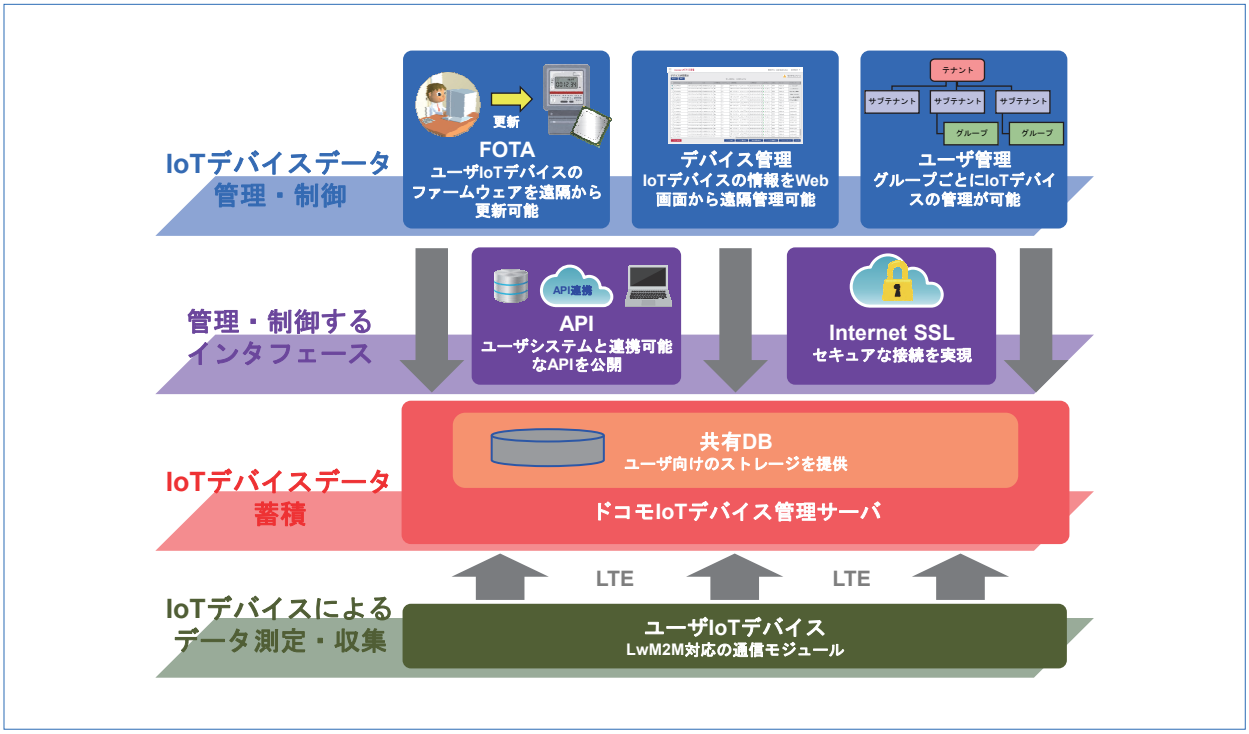

図2　本プラットフォームの機能マップ

＊7　CoAP：IoTデバイスのような省電力・低容量デバイス向けのUDP（User Datagram Protocol）を使ったHTTPのような動作をする通信プロトロル．

＊8　HTTP：WebブラウザとWebサーバの間で，HTML（HyperText Markup Language）などのコンテンツの送受信に用いられる通信プロトコル．

＊9　HTTPS：TLS（Transport Layer Security）プロトコルを用いて，HTTP通信を安全に行う通信手法．なりすまし・中間者攻撃・盗聴などの攻撃を防ぐことができる．

＊10　Internet SSL：主にインターネットを利用してクライアントとサーバとの間で通信を行う際に，通信を暗号化しデータの改ざんを発見することにより，安全に通信を行うためのプロトコル．

＊11　スケーリング：ハードウェアや仮想マシンの負荷状況に応じて通信ソフトウェアとしての処理能力が不足，あるいは余剰になった際に，通信ソフトウェアを構成するVM（Virtual Machine）を増減することにより処理能力を最適化すること．

4. オープンAPI開発

旧システムでは，ユーザのアクセス手段はWeb UIのみだった．そのため，ユーザが必要な機能のAPIを組み合わせたアプリケーションを自システムに実装し，そのアプリケーションを使ってIoTデバイスを管理，制御したいというニーズがあった．本開発では，それができるようにRESTful＊14対応のオープンAPIの実装を行った．オープンAPIはインターネット経由で利用可能な，本プラットフォームユーザに公開されたものである．オープンAPI/Web UIの処理イメージを**図3**に示す．ユーザがオープンAPI/Web UIで本プラットフォームへアクセスすると，DSPはそれを受け付け，vDMEとのインタフェース仕様に従い，vDMEへ指示を渡す．vDMEはDSPから受けた指示をもとにIoTデバイスを制御する．

これにより，オープンAPIを使ってユーザのシステムから直接，IoTデバイス情報を取得したり，ファームウェアアップデートなどを管理，制御したりすることが可能となる．さらに，外部システムと本プラットフォーム間では，認証，暗号化技術を用い，セキュアな通信を通じてオープンAPIの仕様やSDK（Software Development Kit）＊15サンプルをユーザに公開することで，オープンAPI利用の容易化を行った．

5. リアルタイムな通信状況取得と省電力化のバランスが取れる仕組みの実現

OMA-LwM2Mプロトコルの標準規定のパラメータに「LifeTime」がある．LifeTimeとは，IoTデバイスのBindingMode＊16がQueueモード＊17の際，本プラットフォームにRegister（登録）されている時間である．LifeTimeの時間内にIoTデバイスよりアップデート通信を受信しない場合本プラット

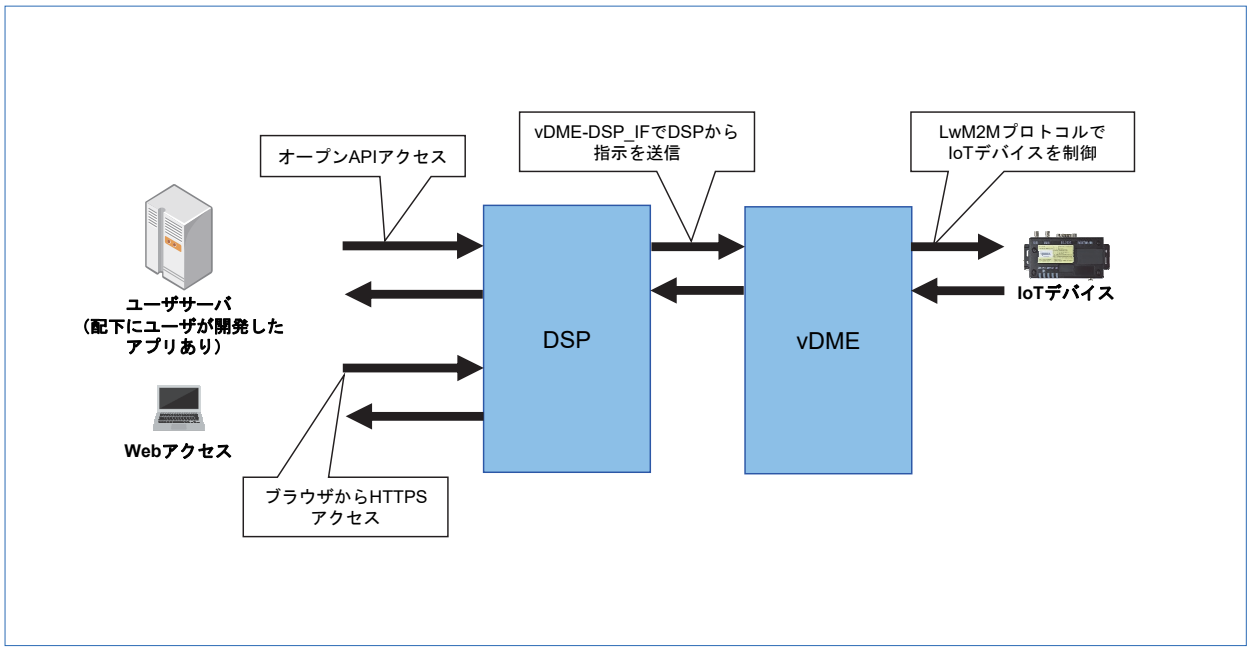

図3　オープンAPI/Web UIの処理イメージ

＊12 NFV基板：通信キャリアのネットワークを仮想化技術により汎用ハードウェア上で実現した基盤．
＊13 ヒーリング：ハードウェア障害や仮想マシン障害が発生した際に，正常なハードウェア上に仮想マシンを移動，または再作成することで通信ソフトウェアとして正常な状態に復旧する手続き．
＊14 RESTful：提供される情報を直接指し示してステートレスに情報を取得／提供する考え方．
＊15 SDK：アプリケーションを作成するときに必要となる，ドキュメント，ツール，ライブラリ，サンプルプログラムなどからなる開発キット．
＊16 BindingMode：OMA-LwM2Mで定義されているIoTデバイスの動作を決定するパラメータ．
＊17 Queueモード：OMA-LwM2Mで定義されているIoTデバイスの動作モードの1つ．

フォームはIoTデバイスのRegisterを解除する．LifeTime値が小さい場合，Register期間が短く，IoTデバイスはアップデートを頻繁に送信するため，消費電力が多くなるが，本プラットフォームはリアルタイムな通信状況を取得することができる．一方，LifeTime値が大きい場合，Register期間が長く，アップデートの送信頻度が低いため省電力化が期待できるが，情報取得のタイムリー性が落ちる．つまり，ユーザのIoTサービス利用目的に応じてLifeTimeの値を設定することで，リアルタイムな通信状況取得と省電力化のバランスを取る必要がある．

本開発では，ユーザごとにLifeTime値を任意のタイミングで設定，変更することにより，柔軟にリアルタイムな通信状況取得と省電力化のバランスが取れる仕組みを実現した．

6. あとがき

本稿では，開発したシステムの全体構成，スケーラビリティの実現，vDMEとDSPの機能分担，新たに開発したオープンAPI，および省電力化を実現するための工夫点について解説した．

今後もIoTデバイス管理に対するニーズを集約し，必要に応じて本プラットフォームに要求される機能追加を進めていく．