

## イベント・バスを用いたスマート・メーター・システムの構築

玉村 亘 川井 秀之 岡田 高幸 早瀬 道朗 堀場 隆文

## Building Smart Meter System with Event Bus

Wataru Tamamura, Hideyuki Kawai, Takayuki Okada, Michio Hayase and Takafumi Horiba

電力会社を中心としたスマート・グリッド実現のため、スマート・メーター・システムの構築が求められている。しかし多数のメーター側の組み込みアプリケーションとそれらを管理するセンター側のアプリケーションの連携においては、ネットワーク環境や通信形態の多様性の観点から、従来の Point to Point 技術を組み合わせたアプローチでは実装上の困難を伴う。本論文では、イベント・バスを活用することでそれらの課題に適應するスマート・メーター・システムを構築できることを示す。イベント・バスを用いて、多数のメーター側とセンター側のアプリケーションを、通信の負荷を増やさずに相互に連携可能な仕組みを実現できた。さらに、実用レベルで要求されるより大規模なシステムにイベント・バスを適用する際の課題について考察した。

The Building of Smart Meter systems is necessary to establish a Smart Grid system for electric power companies. However, in the application integration of many meters and host applications, the point-to-point communication approach results in implementation difficulties because of the diversity of the network environments and communication patterns. This paper introduces the use of the Event Bus to solve problems that appear in the application integration of the Smart Meter system infrastructure. We were able to build an application integration mechanism using the Event Bus to interconnect meter applications and host applications without any increase in network traffic. We also present considerations regarding the large-scale implementation of Smart Meter systems.

Key Words & Phrases : スマート・メーター, アプリケーション連携, イベント・バス, Publish/Subscribe, MQTT  
Smart Meter, Application Integration, Event Bus, Publish/Subscribe, MQTT

## 1. はじめに

電力会社を中心としたスマート・グリッド実現のため、スマート・メーター・システムの構築が求められている [1]。そのためには多数存在する各種メーター上の組み込みアプリケーションとそれらを管理するセンター側のアプリケーションが相互に連携できる仕組みが必要となる。

スマート・メーター関連の研究として業務面からの要件策定が各種標準化団体により進められている [1] ほか、アプリケーション連携の観点においてもデバイス側とセンター側双方からの研究が進められている。デバイス側からのアプローチとして家庭内・屋外などの機器の存在場所を基準とした分類分け、および場所ごとに利用可能な通信手段の整理などが行われている [2]。センター側からのアプローチとして必要な技術要素を体系化したソリューション・アーキテクチャーが存在する [3] ほか、

大量データを高速に処理するためのストリーム・コンピューティングに関する研究が進められている [4]。

しかし、メーター側のアプリケーション（電力使用量の検針やブレーカーの入／切操作などを行う）とセンター側のアプリケーション（検針データの収集やメーターへの命令などを行う）を連携する仕組みは十分に研究されているとはいえない。多数のメーター側とセンター側のアプリケーションを連携させるには、ネットワーク環境や通信形態の多様性の観点から、従来の Point to Point (P2P) 技術を組み合わせたアプローチでは実装上の困難を伴う。

本論文ではイベント・バスを活用してアプリケーション連携基盤を構築する手法が、電力会社におけるスマート・メーター・システムに有効であることを示す。さらにより大規模なシステムにイベント・バスを適用する際の課題について考察する。

まず 2 章でスマート・メーター・システムにおけるアプリケーション連携基盤という対象領域の特性について述べ、従来アプローチでの解決が困難であることを示す。3 章ではイベント・バスをアプリケーション連携基盤として

提出日:2009年5月11日 再提出日:2009年12月14日

用いたスマート・メーター・システムの実装について述べる。4章にてイベント・バス適用の効果について述べ、5章で大規模適用に向けた考察、6章で総括を行う。

## 2. スマート・メーター・システムにおけるアプリケーション連携基盤

### 2.1 スマート・メーター・システムとその特性

スマート・メーター・システムの全体像を図1に示す。ここでは、Field Area Network（以下、FAN）、Wide Area Network（以下、WAN）、Enterpriseの3つの領域に分け、それぞれの領域でのメーター、コンセントレーター、センターの特性について述べる。

メーターはFAN領域に存在し、電力データ（使用量、メーター制御命令など電力業務にかかわるデータ）を受発信する能力を持つ。将来、メーターは各家庭だけでなく電気自動車や太陽光発電装置などにも設置されると考えられるが、各家庭に設置されるとしても、その数は日本においては数千万の規模となる。さらに、新築物件の増加などにより追加が頻繁に発生する。

コンセントレーターはWAN領域に存在し、メーターとセンター間を流れる電力データを中継するための機器で、物理的な設置場所は電柱上などになる。設置場所など物理的な制約により高可用構成を取れず、落雷や台風による全損時は取り替える必要がある。メーターとコンセントレーターとの接続関係は固定の場合もあれば、電気自動車などの移動により動的に接続関係が変わる場合もある。なお、メーターとコンセントレーターは低スペックの組み込み機器とみなせる。

センターはEnterprise領域に存在し、メーターから発信された電力データの受信やメーターに対する命令送

信機能を持つ。構成要素としてメーター管理業務を行うMeter Data Management System（以下、MDMS）、およびWAN側とのインターフェースとなり電力データをEnterprise領域内の各種業務システムに仲介するためのシステム連携サーバーが必要となる。

### 2.2 システムに求められる要件

スマート・メーター・システムにおけるアプリケーション連携基盤は、Enterprise領域では一般的でない多数の要件に対応する必要がある。ここでは大きく2つのカテゴリーに分けて整理する。

#### A) 異なる領域の連携

FAN/WAN/Enterpriseの異なる領域にまたがった連携の仕組みが必要となる。

FAN領域では高速PLCや光ファイバーなどを用いたブロードバンドTCP/IP通信ですべての網を統一できず、特定小電力無線やZigBee、低速PLCなど[5]のように、狭帯域で不安定な非TCP/IP通信も用いられる可能性がある。WAN領域のコンセントレーターは、FAN領域の非IPネットワークとEnterprise領域のIPネットワークをつなぐゲートウェイとして、領域分割のために必須のコンポーネントである。コンセントレーターの多段構成も取り得るため、コンセントレーターが直接メーターやセンターにつながっていない状態でも、全体として正しくデータを届けられるマルチホップ連携の仕組みが必要となる。

#### B) サーバー駆動による処理

スマート・メーター・システムにおいては、センター側からのトリガーによりメーターを操作する必要がある。また、メーターが新たに設置されたり移動されたりするため、サーバー側でメーターを管理できる仕組みが必要となる。さらに個別メーターへの命令配信、特定のメーター群への一斉命令配信など柔軟な通信形態が求められるほか、メーター遠隔制御のためのリアルタイム性も必要となる。

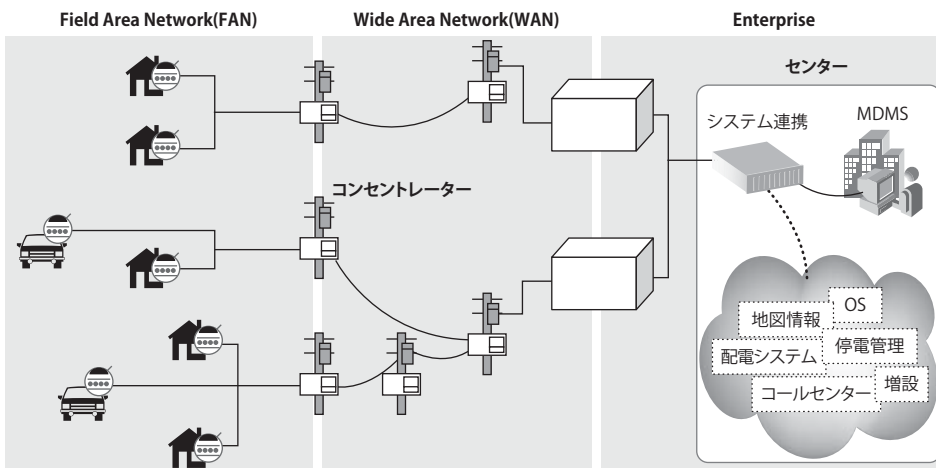


図1. スマート・メーター・システムの全体像

### 2.3 従来アプローチの課題

ここではクライアント・サーバーやEnterprise Application Integration (EAI)、Enterprise Service Bus（以下、ESB）[6]などEnterprise領域で一般的に用いられる、P2P技術を組み合わせる多様なアプリケー

ションを連携する方式を従来アプローチと考え、スマート・メーター・システムに適用したときの課題を整理する。

### A) 異なる領域の連携への対応

FAN/WAN 領域に対しては、TCP/IP を前提とした、しかもヘッダーサイズが大きい（数百バイトあるなど）Enterprise 領域の技術だけで連携基盤を統一できず、デバイス側の技術も組み合わせる必要がある。

また、直接の接続先情報とは別に最終あて先までの経路を管理する仕組みが必要となる。異なる P2P 技術を組み合わせたマルチホップ環境では、End to End で通用するあて先管理の仕組みを独自に作り、それをすべてのコンセントレーターが理解する必要がある。最終あて先への経路選択・経路情報管理テーブルの更新作業など複雑な処理を、信頼性および機器性能の低いコンセントレーターが担う必要がある。

### B) サーバー駆動による処理への対応

サーバー駆動処理を行うためにはメーターが常時稼働してセンターからの接続を受け付ける必要があり、バッテリー駆動の組み込み機器などへの適用は現実的でない。センター側から接続を試みて失敗したときに初めてメーターの不在を検知するため、移動を繰り返すメーターへの対応は難しい。ブロードキャストを用いて探す方法も考えられるが、コンセントレーター間で不必要なネットワークトラフィックを大量に発生させ、帯域を枯渇させる致命傷になりかねない。

## 3. イベント・バスを用いたアプリケーション連携基盤の構築

### 3.1 イベント・バスの役割と構成要素

電力使用量の変化や障害の発生など、状態の変化（イベントの発生）をきっかけに処理を開始するモデルをイベント駆動処理モデルと呼ぶ [7]。単体では意味のないイベントを時系列で捕らえたり、異なるイベントを複数組み合わせることで、隠れていたビジネス的な価値を新たに創造する可能性を持った処理モデルである。

イベント駆動処理は図 2 に示すように、イベントを生成するイベント・プロデューサー、イベントを受信するイベント・コンシューマー、イベントの加工を行うイベント・

バスで構成される。

イベント・バスは単一または複数のイベント・ブローカーにより構成され、イベント・ブローカー同士はブリッジにより、クライアントとブローカーはイベント・チャンネルにより接続される。

個々のイベント・ブローカーは、以下のようなイベント処理を行う。

- 特定条件に基づくほかのブローカーへの転送
- フォーマット変換、複数イベントの集約・分割、妥当性検査や暗号複合化
- クライアントの接続・切断、複数イベントのパターン解析などに基づく新規イベント生成

イベント・ブローカーのうち、イベント・バス全体のあて先制御やほかのブローカーが理解できるイベント形式への変換などバス内の中核的な処理を行うものを EEB (Enterprise Event Broker) と呼ぶ。

図 2 のようなモデルを実現するため、通信には Publish/Subscribe (以下、PubSub) プロトコルの利用が想定される。PubSub のプログラミング・モデルを図 3 に示す。

送信者を Publisher, 受信者を Subscriber と呼ぶ。最低限必要な API は Connect (以下、Con), Disconnect (以下、Disc), Publish (以下、Pub), Subscribe (以下、Sub), Unsubscribe (以下、Unsub) の 5 つである。Publisher は Con にブローカーのアドレスを指定して接続した後、トピック\*1 と送信すべきデータ本体を Pub の際に指定して送信する。Subscriber はブローカーに接続後、あらかじめ受信したいトピックを指定して Sub しておく。ブローカーは該当トピックのメッセージを受け取ると、適切な Subscriber に対してメッセージを送信する。送受信が終わると、Disc によりブローカーとの接続を終了する。

上記の通り、Publisher と Subscriber はブローカーを介した疎結合の関係にあり、ほかの送受信者のアドレスや参加者の数などは意識しない。トピック設計次第で、個別配信・一斉配信・一括データ収集などさまざまな通信

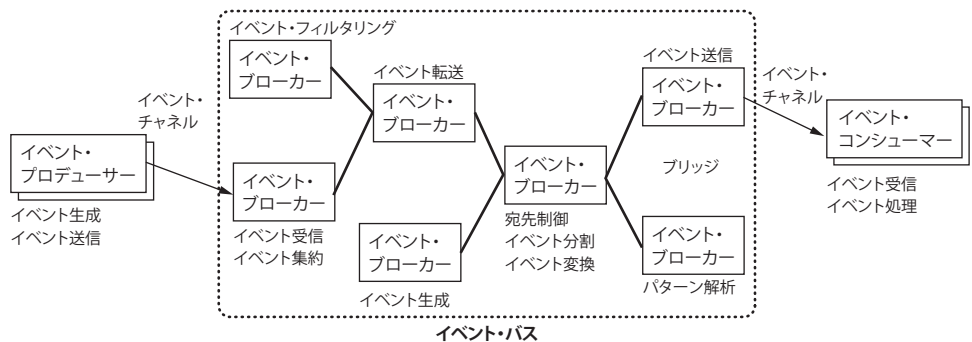


図 2. イベント駆動処理の全体像



が可能となる。

PubSub プロトコルを実現する仕様として、組み込み系ではMQ Telemetry Transport（以下、MQTT）や MQTT for Sensor Networks（以下、MQTTS）[8]、Enterprise 領域では JMS [9]などの利用が想定される。ブローカー同士は PubSub Bridge [10] を用いることで連携することができる。

### 3.2 イベント・バスを用いたスマート・メーター・システムの実装

電力業務において求められるスマート・メーター・システムのアプリケーション連携基盤を、イベント・バスを用いて実装した。イベント・プロデューサーあるいはイベント・コンシューマーに相当するのはメーターおよびセンター側アプリケーションである。メーターは送信処理ではイベント・プロデューサーに、受信処理ではイベント・コンシューマーとなるなど、双方の役割は固定されない。コンセントレーターはイベント・ブローカーとして、センター側のシステム

連携サーバーは EEB としてイベント・バスを構成する。

使用したソフトウェアを表 1、システム構成図を図 4 に示す。

メーターから EEB までは軽量 PubSub プロトコルとして実績のある MQTT V3 を選定した。位置情報はトピックに持たせることで、コンセントレーターには単純なトピック・ベースの転送処理のみ行わせることとした。よって最低限必要な機能を持つ MQTT 実装として、メーターには IA92 [8]、コンセントレーターには RSMB（Really Small Message Broker）[8] を使用している。EEB には MQTT-JMS プロトコル変換、トピック・ベース・ルーティング、DB 連携、固定長 -JSON<sup>\*2</sup> メッセージ変換など多様な機能が求められるため WebSphere<sup>®</sup> Message Broker（以下、WMB）[8] を使用している。

EEB と MDMS、および MDMS とブラウザ間にも PubSub Bridge によるイベント・バスを構成し、電力使用量やメーター稼働状態などをリアルタイムにブラウザ上に表示、およびブラウザからの命令をメーターがイベントとして処理可能なシステムを構築した。EEB と MDMS

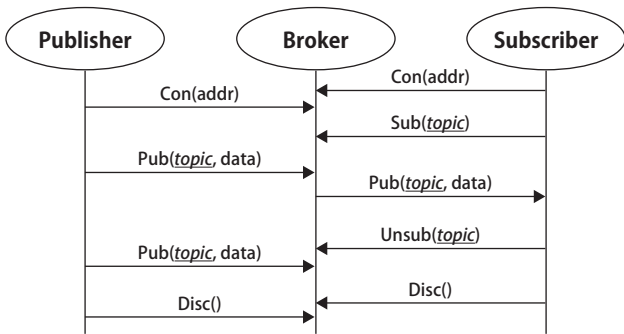


図 3. PubSub プログラミング・モデル

表 1. 使用したソフトウェア

	ソフトウェア	通信プロトコル
メーター	Java™ implementation of WebSphere MQ Telemetry transport (IA92) V1.45	MQTT
コンセントレーター	Really Small Message Broker (RSMB) V1.0.3	MQTT
EEB	WebSphere Message Broker (WMB) V6.1	MQTT・JMS
MDMS	WebSphere Application Server (WAS) V7.0	JMS

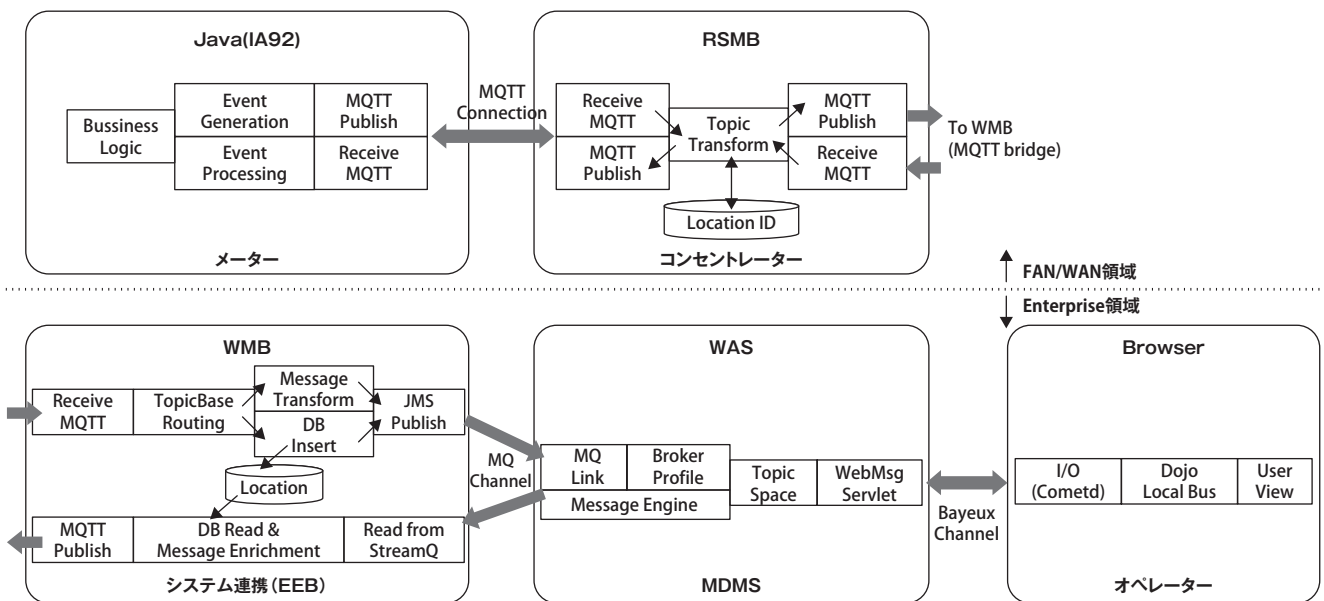


図 4. システム構成図

間には WebSphere Application Server (以下, WAS) V7 から実装された MQ ネットワークとの PubSub Bridge 機能 [11], MDMS とブラウザー間には WAS Feature Pack for Web2.0 の Web メッセージング機能 [12] を用いることで RSMB と WMB, WAS を用いてイベント・バスを構成し, メーターからブラウザーまでの End to End でのイベント処理が可能となっている。

データの流れはメーター・センター間で双方向である。センター側へのデータ送信では, メーターからの Pub や切断イベントをコンセントレーターが検知する。コンセントレーターは PubSub Bridge 定義を元に転送すべきデータかどうかを判断し, EEB に転送する。センターに届いたデータは Sub しているアプリケーションに届けられる。メーターが初めてコンセントレーターに接続するか, 以前と違うコンセントレーターに再接続した際には自身のクライアント ID を含む位置情報変更通知を Pub する。コンセントレーターは自身の ID を位置情報として付加し上流に転送する。センターはクライアント ID と位置情報 (コンセントレーター ID のリスト) を key-value 型の位置情報管理テーブルに格納する。メーター位置の特定には, この位置情報を用いる。センター側アプリケーションはあて先クライアント ID を指定して Pub する。EEB は位置情報管理テーブルを参照し, 位置情報を付加して該当のコンセントレーターに転送する。コンセントレーターは転送する際に自身の ID を位置情報から削除し, 下流の Subscriber に転送する。

上記の実装により実現した対象業務は以下の通りである。考えられる業務はその他多数存在するが, 想定されるデータの流れという観点で, スマート・メーター・システムにおけるイベント・バスの活用方法としては網羅できていると考えてよい。

- 自動検針: メーターの電力使用量を定期的にセンターへ送信する。あるいはセンターから要求があった場合にも送信する。
- ロード・プロファイル: センターからの命令を受け, 6 時間分の電力使用量の推移 (30 分値) をセンターへ送信する。
- 停止停解: センターからの命令により電力の供給を停止したり停止解除したりする。ブレーカーの切/入に相当する。停止時は自動検針も停止する。
- 不点検知: 停電あるいはメーターが故障したことをセンターに通知する。

### 3.3 イベント・バス適用時の考慮点

イベント・バスを適用するに当たって考慮が必要であった点について記述する。

## 1. PubSub Bridge 間の仕様の整合性

WMB から Pub する際に利用可能な Message service domain (以下, Msd) と WAS 側で認識できる Msd がマッチしないため, WAS 側で `json_bytes` と認識される。一方 Web メッセージング・クライアントが処理できるのは `json_text` か `json_object` であるため, メッセージが破棄されブラウザーに表示できない。そのため, WMB 内で `json_text` を送信するため Compute ノード内で以下のようなコーディングを行った。

```
SET OutputRoot.BLOB.UnknownParserName = 'json_text';
SET OutputRoot.BLOB.BLOB = ASBITSTREAM(InputBody, ...);
```

上記のような方法で整合性を確保することで, FAN/WAN/Enterprise というそれぞれ特性の異なる領域を PubSub Bridge により連結し, マルチホップでの双方向アプリケーション連携を実現することができた。

## 2. データの確実な送達

MQTT では QoS<sup>\*31</sup> 以上のメッセージを用いることで転送保証されるが, RSMB V1.0.3 では初回起動後, 接続先ブローカーに接続できていない状態では PubSub Bridge は有効になっていない。この状態で Pub すると, persistence パラメーターが有効であっても QoS1 以上のメッセージがロストする。これを解決するには RSMB 起動時に接続先ブローカーを内部的に登録しておくような仕様変更が必要となる。位置情報登録は確実にセンターに届く必要があるため, 今回はメーターがセンターからの応答を受け取るまで Pub を QoS0 で再送することとした。これにより, 組み込みブローカーの実装に依存せず, ネットワーク負荷を抑え, かつ転送直前のコンセントレーター全損といった特殊ケースへの対応も含めてデータの確実な送達という要件を満たすことができた。

## 4. イベント・バス適用の効果

イベント・バスを用いて, 多数のメーター側とセンター側のアプリケーションを, 通信の負荷を増やさずに相互に連携可能な仕組みを実現できた。なお, スマート・メーター・システムの特性には, 以下のように対応できる。

### A) 異なる領域の連携への対応

コンセントレーターは PubSub Bridge の定義に従って隣接したブローカーにデータを転送するだけで, 上流・下流の双方向に正しくイベントを送達できる。あて先管理をセンター側で行うことでコンセントレーターに負荷をかけた構成が可能となる。

また MQTT, MQTTS, JMS, MQ といったさまざまな実装を使用して PubSub という共通のモデルを利用できる。よって非 IP 環境の MQTTS クライアントからの Pub をセンター側の Subscriber が JMS で受け取るようなシステムが実現可能となる。メーターの増加は新規位置登録で、移動は切断イベントとしてセンターが検知できる。

### B) サーバー駆動による処理への対応

メーターはコンセントレーターに対して Sub を発行しておくことでセンターからのデータを受け取る。例えば MQTTS の sleeping client<sup>\*4</sup> 機能を使用すれば、メーターが休眠状態から復帰したタイミングでデータを受け取ることもできる。つまりメーターがサーバーとして振る舞う必要がないにもかかわらず、センターはクライアントのように振る舞うことができる。また PubSub の特性を生かして特定の位置情報を持つメーターに対するデータの一括収集や一斉命令配信などが容易に行える。データは Subscriber にのみ届くため、ブロードキャストのように不要なネットワーク・トラフィックが発生することもない。

## 5. 大規模適用に際しての考察

ここでは、大規模なスマート・メーター・システムへの適用に際して考慮すべきポイントを整理する。

### 5.1 障害に対する考え方

FAN/WAN 領域で発生したデータは、できるだけバックアップや高可用構成などの面で信頼性の高いセンター側で処理をする方が、データロストの可能性を減少させることができる。つまりメーターおよびコンセントレーターの信頼性に過度に依存しない方がよいと考える。具体的にはコンセントレーター内に業務データを保持する retain publication<sup>\*5</sup> や persistence 機能に依存したシステム設計を行わないことが推奨される。メッセージ再送によるデータの送達保証はその一例である。

電力業界においては通信不能（メーターは稼働中）と不点（通電せずメーターは非稼働）を区別する必要があり、現在は不点検知において MQTT の Last Will Testament（以下、LWT）機能<sup>\*6</sup>を使用しているが、厳密にはメーターにバッテリーが必要となる。通電停止の検知と不点イベントの発信はバッテリー電源を使用して行う。メーターにバッテリーを搭載できない場合は LWT を検知後、一定時間メッセージが来ない時

に不点とみなすなどのロジックが必要になる。

### 5.2 セキュリティー

MQTT V3 の仕様自体はセキュリティーの概念を持たない。よって V3.1 以降の認証機能や MQTT over SSL 機能、あるいは IPSec や stunnel のような SSL トンネリング・サービスの利用、アプリケーション・レベルでのメッセージ暗号化による対応などが必要となる。MQTT のセキュリティーに関して整理された文献 [13] がある。異なる PubSub Bridge を連携する場合にはトランスポート・セキュリティーの仕組みも異なる可能性があり、アプリケーション・レベルでの End to End のデータ暗号化なども考慮する必要がある。

### 5.3 アプリケーション設計

イベント・バスの位置付けを図 5 に示す。メーター側とセンター側双方のアプリケーションは、それぞれイベント・バスとの通信のみを意識すればよくなるため、アプリケーション間の通信は特定のハードウェアあるいはネットワーク技術の制約を受けない。独自の通信手段にしか対応しない HAN（Home Area Network）デバイスはゲートウェイ機能を持つメーター経由、またサービス志向のセンター側アプリケーションは ESB 経由でイベント・バスに接続する。アプリケーションの開発者が個々の実装技術や API を意識せずに開発するには、図 5 にあるようなイベント・フレームワークの開発が必要となる。

また PubSub を用いたシステム連携においてはトピック設計が極めて重要となる。電力メーター以外のデバイスを扱うなどの環境変化を見越してトピックを設計することで追加要件に対応しやすくなる。適切な設計により PubSub Bridge 間転送、つまりネットワーク・トラフィックも抑制できる。相対的に複雑なコンテンツ・ベースではなくトピック・ベースで転送制御することで、データ解析負荷

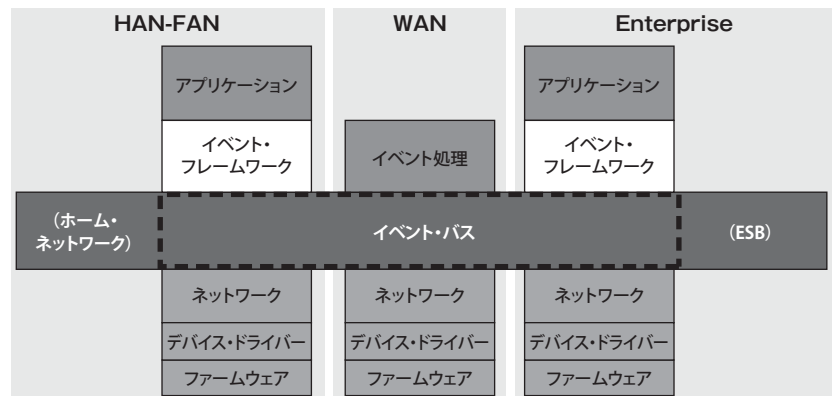


図 5. イベント・バスの位置付け



も抑えることができる。

現在はイベント 1 件に対してアクションを起動する 1:1 あるいは 1:N 形態となっている。特定期間内の異常な使用状況を検知してあらかじめ登録した連絡先に通知するようなサービスを提供する場合、複数のイベント情報を踏まえてアクションを起動する N:1 あるいは N:M 形態の処理が必要となる。このような処理は EEB とは別に BEP (Business Event Processing) [7] エンジンを組み合わせ、役割分担の明確化による負荷の軽減・生産性の向上を図ることが推奨される。

#### 5.4 スケーラビリティ・パフォーマンス

多数の WAN 側からのリクエストを処理する EEB の負荷増大が主要な考慮点となる。

まずはスケールアウトにより対応できるかを見極める。複製ブローカー [14] による可用性の向上などを合わせて検討する。ブローカーが複数になる場合、集合 [14] によるブローカーごとのクライアント数削減を行う。コンセントレーターごとに固定の接続先リストを持たせると柔軟性は損なわれるが動的に接続先を変える必要がない場合には必要十分である。接続先情報を返す専用の割り振りサーバーを配置する場合には障害発生時の問い合わせ集中を考慮する。

HTTP の負荷分散などで一般的に用いられるロード・バランサーは、MQTT や JMS といったプロトコルを明示的にサポートしていないことが考えられる。その場合プロジェクト・リスクでの稼働検証となるほか、MQTT の clean session 機能<sup>\*7</sup>をはじめ TCP より上位の固有の機能が負荷分散装置を介して使えないことでサーバー駆動が実現できなくなるなどの問題が出る可能性があり、適用に際しては慎重に検討する必要がある。

イベント・ブローカーを多段構成にすることで、同時接続数は大幅に減らすことができる。ただし多段にするブローカーが組み込み機器で稼働する場合、高可用構成を取れないため単一障害点になる。また EEB を挿入すると高可用構成を取れるが、コスト面での負担が大きくなる。

位置情報管理テーブルその他アプリケーション DB にはインメモリー DB や分散オブジェクト・キャッシュ技術などデータ・アクセスの高速化技術 [15] を用いる必要がある。また、スループットを抑えたい場合、コンセントレーターで複数メーターのデータを集約し、送信回数を減らして一回あたりの送信データ量を増やす集約の仕組みが有効なケースもある。ただしこの場合は EEB でのデータ解析の負荷が高まるため注意が必要である。また、コンセントレーターのストレージにデータを一旦保持する形

態となるため、コンセントレーターの信頼性が低いケースでは推奨できない。

#### 5.5 実装の提供

組み込みブローカーの選定においては、必要なリソースが少なく稼働プラットフォームの制約が少ない C ベースの実装を選ぶか、JVM の移植が必要となるが多機能な Java ベースの実装を選ぶか、を考慮する必要がある。なお現時点では MQTTS を実装した製品は発表されていない。ベンダーからの OEM やサービスの一環としての提供のほか、新しいハードウェアや通信プロトコルにいち早く対応するためにメーター・メーカーが独自実装するケースも考えられる。この際、相互接続に関する問題を避けるため、MQTT や MQTTS などの通信仕様が公開された仕様を用いるべきである。メーターやコンセントレーターの提供者がこの原則に従うことで、システム全体としてイベント・バスを機能させることができる。

#### 6. おわりに

本論文ではイベント・バスを活用してアプリケーション連携基盤を構築する手法が、スマート・メーター・システムに有効であることを示した。また、実際の電力業務において求められるアプリケーション連携基盤を実装し、通信の負荷を増やさずに相互に連携可能な仕組みを実現できた。さらに、実用レベルで要求されるより大規模なスマート・メーター・システムへの適用に際して考慮すべき点を明確にした。今後、本論文で検討した内容を基に大規模なスマート・メーター・システムにイベント・バスを適用し、その効果を検証していきたい。

謝辞

当論文にて実装したシステムの実プロジェクトへの展開に際しては日本アイ・ビー・エム・サービス株式会社 (ISC-J) の皆様にご尽力いただきました。MQTT や MQTTS の最新技術情報を Hursley Software Development の皆様にご提供いただいたほか、関連研究

※ 1 データの特性を表す文字列  
 ※ 2 テキストベースの軽量データフォーマット  
 ※ 3 0 は最大 1 回送達、1 は最低 1 回送達、2 は確実に 1 回送達。  
 ※ 4 休眠中のクライアント宛のデータをブローカーが保持しておく機能  
 ※ 5 トピックに対する最新のパブリケーションをブローカー内に保存しておく機能  
 ※ 6 ブローカーがクライアントの切断を検知したときにメッセージを Publish する機能  
 ※ 7 前回の接続情報の保持を選択できる機能

および事例情報はグローバルの Energy & Utilities 関係者から、製品情報はソフトウェア事業各ブランドのテクニカル・セールスの皆様から提供いただきました。ここにあらためて深謝いたします。

## 参考文献

- [1] 市川類：米国におけるスマート・グリッドの産業構造と標準化を巡る最近の動向，ニューヨークだより 2009 年 7 月臨時増刊号，情報処理推進機構。
- [2] Erich W. Gunther: "An Overview of Smart Grid Standards", OpenSG Users Group (2009.02).
- [3] 水谷 浩二：環境を考慮した社会インフラとしての電力・ガス業界イノベーション，ProVISION No.55, pp.61-62 (2007).
- [4] Alan Hopkins: "Event stream processing with WebSphere eXtreme Scale", [http://www.ibm.com/developerworks/websphere/techjournal/0809\\_hopkins/0809\\_hopkins.html](http://www.ibm.com/developerworks/websphere/techjournal/0809_hopkins/0809_hopkins.html) (2008.09).
- [5] 阪田史郎："将来の NGN を支えるユビキタスネットワーク技術"，第 4 回 明日を拓く高度情報通信シンポジウム，千葉大学大学院 (2007.12).
- [6] Martin Keen: "Patterns: Implementing an SOA Using an Enterprise Service Bus", IBM Redbooks, ISBN 0738490008 (2004.07).
- [7] 星島洋一："SOA を加速するイベント処理技術の価値"，渋谷テクニカルナイト・セミナー (2008.11).
- [8] MQTT.org: <http://mqtt.org/> (2008.08).
- [9] Java Message Service Documentation, <http://java.sun.com/products/jms/docs.html> (2002.03).
- [10] Matthew Roberts: "Bridge for linking two publish/subscribe message brokers", United States Patent 7389325 (2008.06).
- [11] WebSphere MQ ネットワークとのパブリッシュおよびサブスクライブ：[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.pmc.doc/concepts/cjc0005\\_.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.pmc.doc/concepts/cjc0005_.html)
- [12] 須江信洋，田中孝清："Web Messaging"，WAS Feature Pack for Web2.0 Workshop (2008.04).
- [13] Andreas Wirth: "Securing the MQTT Publish-Subscribe Protocol"，University of Zurich (2006.07).
- [14] パブリッシュ／サブスクライブ・トポロジー：[http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r1m0/topic/com.ibm.etools.mft.doc/aq01190\\_.htm](http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r1m0/topic/com.ibm.etools.mft.doc/aq01190_.htm)
- [15] インメモリ技術を利用した超高速トランザクションへの対応：[http://www.ibm.com/developerworks/jp/websphere/library/wxd/wxs\\_soliddb/index.html](http://www.ibm.com/developerworks/jp/websphere/library/wxd/wxs_soliddb/index.html) (2009.06).



日本アイ・ビー・エム株式会社  
ソフトウェア事業  
西日本ソフトウェア・テクニカル・セールス  
アドバイザー IT スペシャリスト

玉村 亘 Wataru Tamamura

### [プロフィール]

主に MQ ファミリー製品群の提案支援に従事。2005 年から西日本のお客様に対する WebSphere 製品群の提案支援を担当。システム連携をキーワードに，SOA (Service Oriented Architecture) やスマート・メーターなどのソリューションを活用したお客様の課題解決をサポートしている。



日本アイ・ビー・エム株式会社  
未来価値創造事業  
スマートエネルギーソリューション部長

川井 秀之 Hideyuki Kawai

### [プロフィール]

企業間取引 (EDI) システム開発や B2B, EC など電子商取引のソリューション開発のリーダーを経て，2005 年より電力ガス IUN ソリューション・リーダー，2009 年より未来価値創造事業にて新エネルギーも含めたエネルギー・ソリューションを担当。



IBM ビジネスコンサルティング  
サービス株式会社  
戦略コンサルティングサービス  
シニアマネージングコンサルタント

岡田 高幸 Takayuki Okada

### [プロフィール]

IBM 製造開発統括部門に入社後，製品開発，製造，サービス・ビジネスを経て，IBCS に参画。半導体，組み込み製品，基板製造，ソフトウェア開発，設計サービスなどテクノロジーにかかわる幅広い開発プロジェクトとプロジェクトマネジメントを経験している。電子機器，通信，半導体，電力などに対して，テクノロジーにかかわるコンサルティング・サービスを提供。専門分野は，半導体，半導体開発，回路基板，システム開発，および組み込み系ソフトウェア。



日本アイ・ビー・エム・サービス株式会社  
GBS 事業部  
第二アプリケーションサービス本部  
部長  
アドバイザー PM

早瀬 道朗 Michio Hayase

### [プロフィール]

1994 年，日本 IBM 東海情報ソリューションに入社，以来，中部地区製造業のお客様を中心に SE, PM としてデリバリーを担当。2008 年からは，日本 IBM サービスのライン・マネージャーとして，公共・公益のお客様を中心としたデリバリー部門を担当。



日本アイ・ビー・エム・サービス株式会社  
GBS 事業部  
第二アプリケーションサービス本部  
IT アーキテクト

堀場 隆文 Takafumi Horiba

### [プロフィール]

2001 年に日本アイ・ビー・エム・サービスに入社。入社以来，WebSphere Application Server を利用した Web システムの構築に従事。その後，Web のフレームワークの設計・開発および，技術支援に従事する。