共同研究報告書(2024 年度)

研究課題: MQTTRPC によるスマートシティ用サービス基盤の実装

研究担当者　北海道大学大学院情報科学研究院
教授　土橋 宜典

あらまし国が推進するデジタル田園都市国家構想では都市で発生する様々なデータをインターネットインフラで流通させ、それを活用するサービスを実装することで都市運営の効率化、環境負荷低減などを実現することを目指している。IT サービスで強化された都市機能はスマートシティと呼ばれ、その実装が国内外で進められている。その中核となるのがデータ利活用基盤サービス基盤であり、データ交換のプロトコルとして FIWARE が推奨され、多くの自治体でサービス提供が開始されている。公共事業としてスマートシティの情報流通基盤の実装が進む一方で、それを活用したサービスの実装は遅れている。本研究では札幌市が提供するデータ利活用サービス基盤であるさっぽろ圏データ取引市場を例として、具体的なサービスを実装することにより、都市のスマートシティ化について検討した。

キーワード　Society 5.0, スマートシティ, FIWARE, NGSI, IoT,　MQTT,　RPC, ブローカー, 人流計測

1. まえがき

昨年度、本研究助成事業で開発した MQTTRPC 上で実装した IoT サービスは、MQTTRPC の汎用性の高さから、カメラサービス、LBS(位置情報サービス)など多様なサービスを短時間で実装可能なことを示すことができた。この成果を活用し、さっぽろ圏データ取引市場と連携した IoT 情報サービスを実装することにより、スマートシティのプロトタイプサービスの実装を行った。

さっぽろ圏データ取引市場は札幌市およびその近郊で集められるデジタルデータを、それを必要とする利用者に対して配布するサービスを提供する。データは有償のもの、無償のものがあり、決済システムも含まれている。データ取引市場ではサービスは FIWARE により実現されているが、ここにアクセスするためには決済を含めた厳密な認証手続きが必要となる。一方、スマートシティが想定しているサービスは IoT 型であり、アプリケーションも携帯端末や無人機器に組み込まれる場合が多く、そのような端末に決済情報が含まれることのリスクがある

また、FIWARE が取り扱うデータ形式 NGSI-v2 は IoT 型のセンサーが取り扱う小規模データに対してプロトコルオーバヘッドが大きく、不必要にサーバー資源やネットワーク帯域を消費する。そのため、アプリケーションが直接に FIWARE をアクセスするのではなく、データ仲介用の MQTTRPC ノードを経由してデータのやり取りをする形式を提案し、実装した。

2. MQTTRPC の通信モデルと基本プロトコル

2.1. MQTTRPC の構成要素

MQTTRPC は MQTT を下層プロトコルとして、その上に実装する遠隔プログラム実行プロトコルである。MQTTRPC の構成要素は以下に示す 3 要素からなる。

1. MQTT Broker
2. Edge Node
3. Client Node

MQTT Broker は各ノードがメッセージ交換するときに使用する中継機能を提供する。実態は標準的な MQTT Broker (Mosquito)である。Edge Node と Client Node は動作上の違いは無いが、動作環境が異なっている。Edge Node は IoT サービスではセンサーのように遠隔地の無人環境で動作するノードコンピュータを想定している。一方、Client Node は人間が操作するブラウザ画面をイメージしている。クライアントノードはブラウザのように、人間が操作する時だけ一時的に動作するものを想定している。これら 3 要素の関係を図 1 に示す。

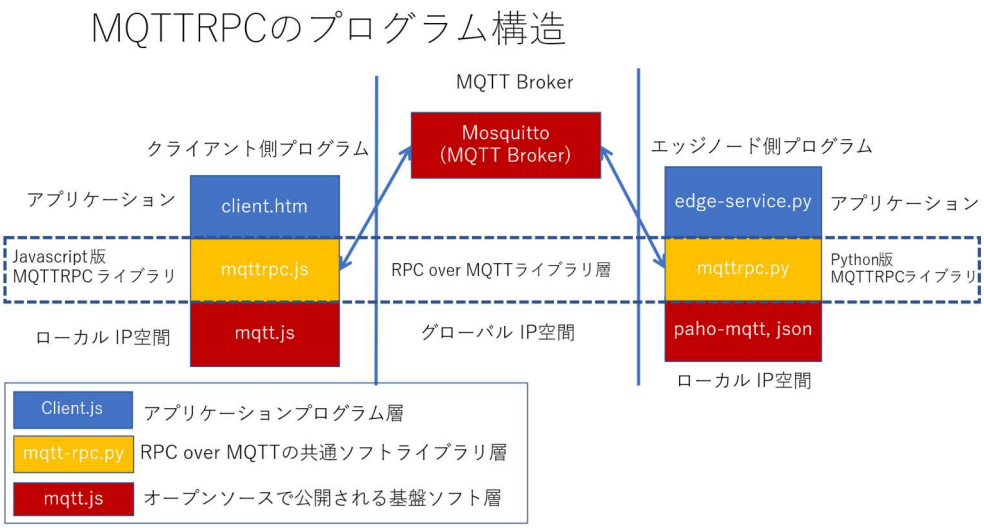

図 1. MQTTRPC の構成要素

2.2. MQTTRPC のメッセージ交換パターン

MQTTRPC はエッジノード上では常駐プロセス（MQTTRPC Daemon）として実行されている。ノード上の MQTTRPC Daemon は重要な動作パラメータとして以下の 4 パラメータを使う。

1. MQTT Broker URI
2. Service Key
3. Service Class
4. Node ID

MQTT Broker URI は各ノードが RPC メッセージを交換する時に経由する MQTT Broker のアクセス情報であり、IP アドレスと TCP ポート番号で定義される。Service Key は MQTT Broker が管理するサービス空間を識別するものである。Service Key を共有するノードに対しては同報通信を行うことができる。Service Key は Major Key と Minor Key の 2 階層で記述される。Major Key やサービス提供者の名称など大枠でのくくりであり、個々のサービス名は Minor Key で指定することを想定している。つまり、一つの Major Key に対して、複数の Minor Key を定義できる。Service Class は Service Key で指定されたサービス空間に存在するノードの種類(Class)を指定する。一般的な応用では”edge”と”client”の 2 種類のクラスが使用される。Node ID は使用する MQTT Broker 内でユニークな ID でノードの識別のために使われる。Node ID は MAC アドレスや CPU ID のようにユニーク性が担保されたものを使うことができる。

2.3. MQTTRPC のトピックパターン

MQTTRPC では MQTT の TOPIC パターンに通信の相手先、通信モードを記述し、メッセージ部に RPC オブジェクトを JSON 形式で記述する。MQTTRPC が発する RPC リクエストは以下の 5 階層の TOPIC パターンを持つ。

major_key / minor_key / service_class / dst_node / src_node

各階層の値（文字列）は'*'を使用することでワイルドカードとして機能させることができる。
例えば、*service_key* = it-prototyping/camera のサービスに含まれる、*src_id* = dc1234567 から *dst_id* = b81234567 に対して RPC メッセージを送る場合は、以下のトピックパターンになる。

it-prototyping / camera / * / b81234567 / dc1234567

これは、１対１の通信パターンである。
もう一つのパターンとして、*service_class* に含まれるノードすべてに同報型の RPC メッセージを送る TOPIC パターンがある。src_id = dc1234567 から service_class = edge であるノードすべてに RPC メッセージを送る場合は以下のトピックパターンになる。

it-prototyping / camera / edge / * / dc1234567

これは service_key と serivice_class が同一である全ノードに対してメッセージを送る通信パターンである。図 2-1, 図 2-2 に通信パターンを図示する。

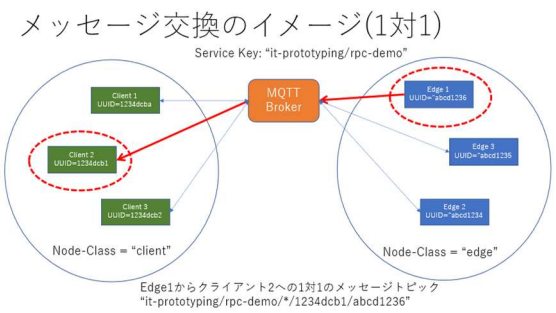

図 2-1 Node ID 指定の通信パターン

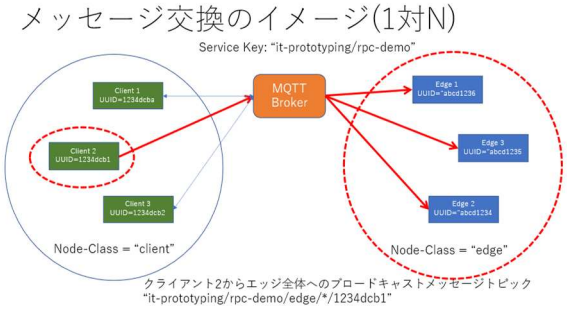

図 2-2 Service Class 指定の通信パターン

3. さっぽろ圏データ取引市場への MQTTRPC の応用

札幌市が運営するさっぽろ圏データ取引市場は NGSI-LV2 REST API 準拠のアクセスプロトコルによりデータ提供を行っている。提供されているデータのカタログ画面を図 3 に示す。データの多くは更新頻度が月単位の静的なものであるが、リアルタイム性があるデータとして札幌市地下歩行空間に設置された人流計測センサーの観測データが 15 分間隔の更新で提供されている。

リアルタイムデータ、特に非同期的に更新が起こるデータの場合は HTTP によるプル型のデータリクエストプロトコルでは頻繁にデータリクエストを繰り返す必要があり公立的ではない。しかし、札幌圏データ取引市場では FIWARE による REST API 準拠のアクセスプロトコルしか提供しないため、これを利用するクライアント端末が多数になると API のエンドポイントへのアクセス回数が端末数に比例することになり、サーバー負荷の増加や API 利用料金が増加するなどの問題がでてくる。

その問題を解決するために、本研究では FIWARE の API エンドポイントとクライアントの間に FIWARE-MWTTRPC 変換のプロキシサーバーを置き、クライアント側を非同期動作させる実装を行った。これにより、プロキシサーバーとクライアントの間の通信は MQTTRPC により非同期化され、クライアントはデータ更新が発生するまで割り込み待ち状態にできる。図 3.2 は FIWARE サーバー、HTTP-MQTTRPC プロキシサーバー、クライアント端末の関係を図示している。

図 3.1　さっぽろ圏データ取引市場で公開されているデータのカタログ

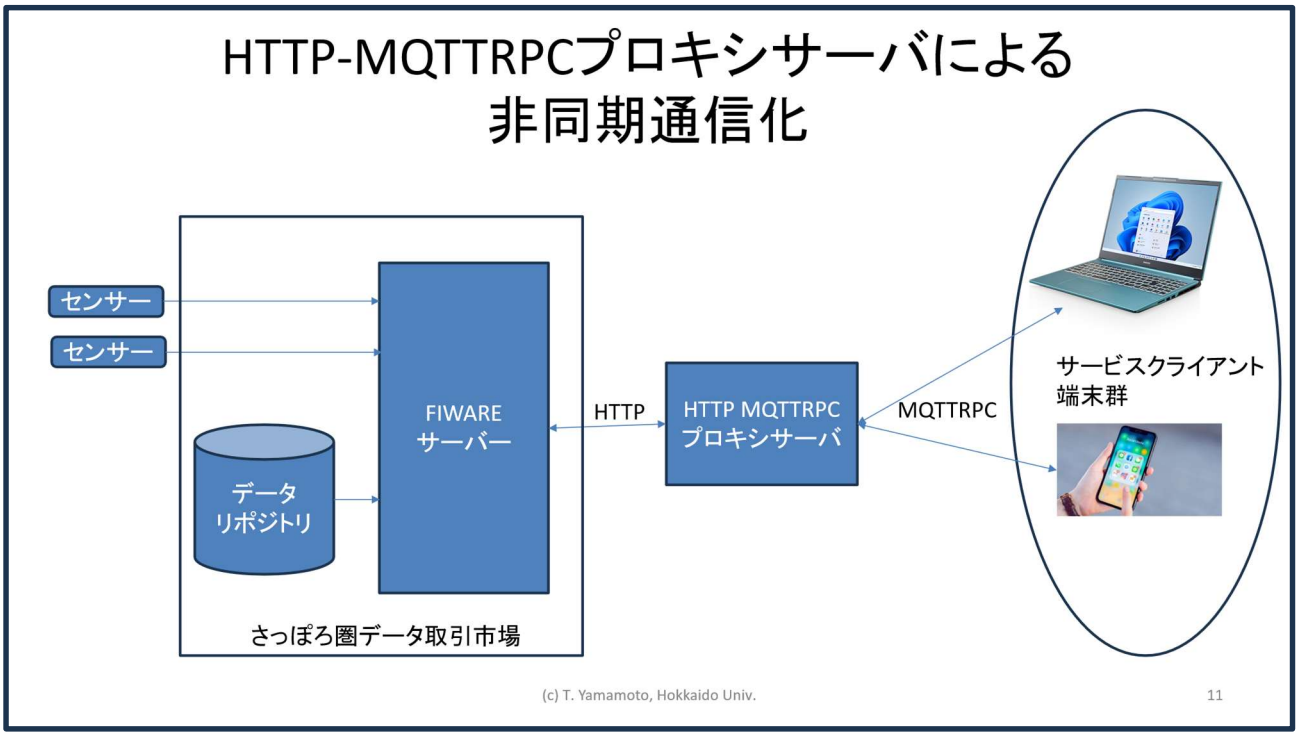

図 3.2 FIWARE サーバー、HTTP–MQTTRPC プロキシサーバー、端末の関係

図 3.3、図 3.4 は MQTTRPC を用いて実装した人流センサーデータの可視化ツールの実装例である。この実装では javascript で記述された MQTTRPC ライブラリを用いてアプリケーションが記述されており、ブラウザ上で動作する。MQTTRPC により非同期化されており、端末では FIWARE でのデータ更新が起こったときのみ画面の更新が実行される。つまり、データ更新の確認はプロキシサーバーが代表して定期的に行い、更新発生時に MQTTRPC がプッシュメッセージを各端末に送信することになる。これにより、端末数が増加しても、FIWARE サーバーへのアクセスは増加せず、データ取引市場に対する負荷が下げられ、従量課金による負担金も軽減できることになる。

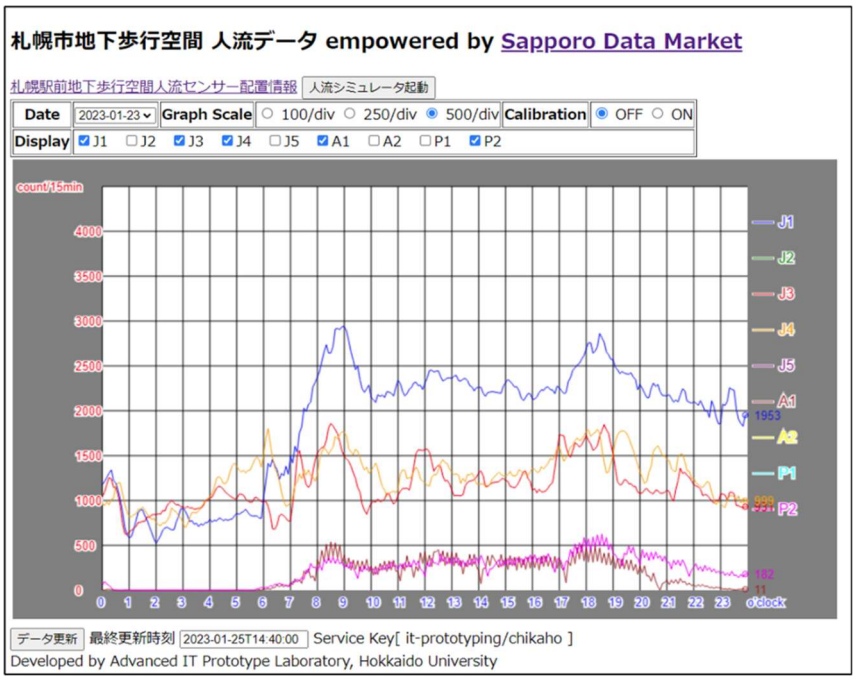

図 3.3 人流データの可視化例 1

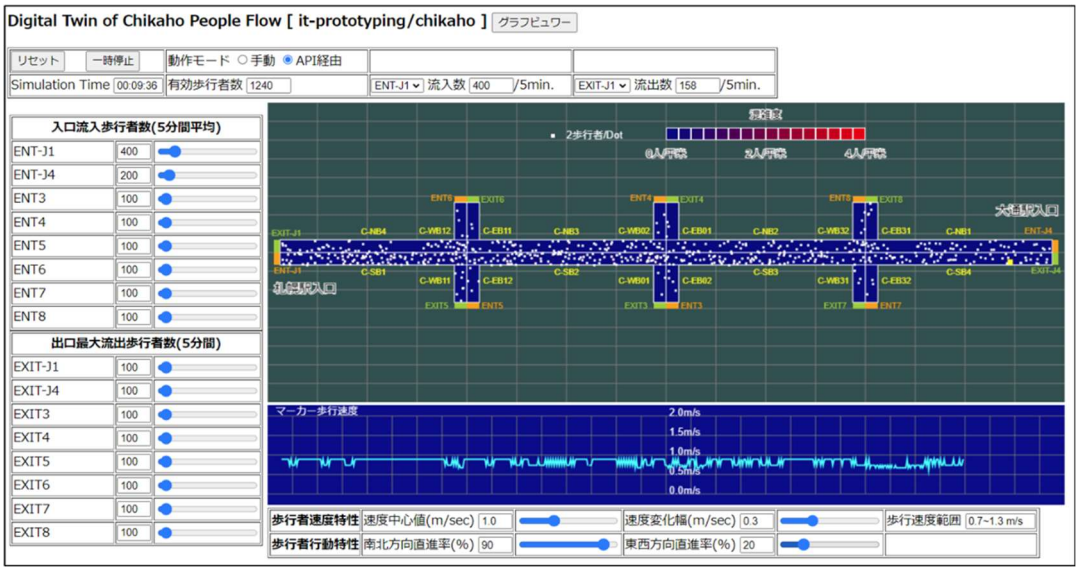

図 3.4 人流データの可視化例2

研究担当者　北海道大学大学院情報科学研究院
教授　土橋 宜典

4. むすび

札幌市が運営するさっぽろ圏データ取引市場が提供するデータを用いて具体的なサービスの具体例を提案し、サービスとして実装した。都市の状況を表現するデータはリアルタイム性があり、かつ非同期であることが多く、現在の標準プロトコルとして推奨されている FIWARE のようなプル型で同期的なアクセスプロトコルを用いるとできることに限界が出てくる。私たちは FIWARE とクライアントの間に HTTP と MQTT のプロトコル変換を行うプロキシサーバーを置くことにより、クライアント側はプロキシサーバーからのプッシュ型データ配信によりデータを受信する構造を MQTT 上に実装した RPC プロトコルである MQTTRPC とその応用例の開発について報告した。MQTT は低機能でシンプルな通信モデルであるが、不安定な通信環境でも安定して動作する特徴を備えている。本研究で開発した応用例はその特徴を生かしたものであり、MQTTRPC で実用レベルのサービスが実現できることを示すことができた。