

# 7. 都市 OS

## 7.1. 都市 OS の概要

都市 OS は、各分野に分断されていたデータを、統合して運用できるようにする統合運用システムのことを指す。これまで国内の「データ連携基盤」については、分野毎の「データ連携基盤」がデジタル田園都市国家交付金等で実装が進められてきたが、都市 OS は分野毎（各サイロ）にあるものではなく、分野間のデータを統合して運用できるようにするものである点が、これまで進められてきた国内の「データ連携基盤」とは異なる点である。

スマートシティにおける都市 OS は、ITU-T Y.4600 の「スマートシティでのデジタルツイン導入に際して要求される機能群」（図 9.5-1）に準拠したものを想定している。ただし一足飛びに都市の 3D モデルを作ることを意味していない。これは詳しくは 9.5.スマートシティ領域でのデジタルツイン国際標準化で述べるが、国内においては SCRA 別冊「地理空間データ連携基盤」にて示した分野間を横断したデータ重ね合わせを地図上（仮想空間）で実現することから開始し、3D 化は空間 ID を用いてこれを高さ方向に拡大したものを想定している。ここでは簡単にまずデジタルツインの機能と都市 OS（地理空間データ連携基盤を含む）の役割の対応を表 7.1-1 にまとめておく。

表 7.1-1 デジタルツインの機能と都市 OS の役割

デジタルツイン（Y.4600 の機能）	都市 OS（地理空間データ連携基盤を含む）の役割
データ収集 (Data Acquisition)	FIWARE 等の IoT Agent の機能等による直接的な動的データ取得 外部の API から取得する間接的な動的データ取得 外部の API から取得する地理空間データ情報取得 外部の API から取得する静的データ情報取得
データ管理・統合 (Data Management and Integration)	分野間のデータの統一フォーマット化（静的データ、動的データ） 統合監視機能
モデル管理 (Model Management)	地理空間データ連携基盤による地図モデルの管理 地図の自動更新、地図の時系列管理（過去と未来）
データ解析 (Data Processing and Analytics)	AI・機械学習を活用し、分析エンジンとしてのデータ処理
可視化 (Visualization)	地理空間データ連携基盤における分野横断したデータの重ね合わせ 地理空間データ連携基盤を活用した市民への情報共有（アプリや Web） 地理空間データ連携基盤の API 提供 (SCRA では別冊 SCRA「地理空間データ連携基盤」にて仕様を公開)
セキュリティ・プライバシー (Security and Privacy)	アクセス制御や個人情報保護 SSI 等
相互運用性 (Interoperability)	分野間・都市間のデータ連携基盤間、都市 OS 間接続と相互運用 (NGSI-LD、DCAT-AP、SSI)

上記の通り都市 OS は、通常のデータ連携基盤と異なり、個別のビジネスサービスに依存しない機能要件があることが分かる。都市 OS の役割は、「都市のデータの統合基盤」としての機能であり、デジタルツインの要素が求められる。特に SCRA 別冊「地理空間データ連携基盤」に公開した仕様にあるとおり、分野を横断した標準的なデジタル地図基盤である地理空間データ連携基盤は、都市 OS にとって必須となる。まずこの地理空間データ連携基盤上に分野間データを連携させ、分野間を横断したデータの見える化をすることが、まずはスマートシティ実現の第一歩である。

これまでは、都市 OS とは「スマートシティを実現しようとする地域が共通的に活用する機能が集約され、スマート

シティで導入する様々な分野のサービスの導入を容易にさせる」とされてきた。しかしながら様々な分野のサイロ化した機能毎のビジネスサービスのために都市OSを整備することはオーバースペックであるし、都市OSをわざわざ活用しなければならないシステム要件上の根拠が希薄である。

スマートシティは分野間・都市間を横断し、これまでの分野間・都市間のはざまにある人の課題を発見し、この発見に基づいた課題にフォーカスして対策を検討し、都市経営におけるEBPMを実現するものである。これを実現するのが都市OSである。よって課題に対する対策はデジタルや新技術を活用したものとは限らない。しかし分野間データを活用した課題発見手法を用いて発見した課題であるために、その課題に対する対策による効果測定は、課題発見時のデータを活用して可能となるのである。こういう点において、都市OSはエコシステムであると言える。

上記の環境や運用方法が整った上で、課題発見手法に用いたリアルタイムデータを活用した課題解決のイノベーションは、この時に起こると言うことは言うまでもない。つまり都市OSは、既にある商材を持ち込みたいベンダーには不要であるが、データを用いた課題発見とその解決方法についてイノベーションを創出する地域にとってはエコシステムでもあると言える。

以下は比較の為、第二版までの都市OSの要件を掲載しておく。都市OSの要件は、本来は以下のような状態を表すものではなく、社会に対して必要とされる機能とは何かの観点で都市OSの要件を記載していくべきである。

「都市OSの役割は、スマートシティの運営やスマートシティサービスに求められる、代表的な3つの特徴「(1) 相互運用(つながる)、(2) データ流通(ながれる)、(3) 拡張容易(機能を広げられる)」を実現するために有効となる基本的開発環境及び、運用環境を提供する」(第二版までの都市OSの要件)

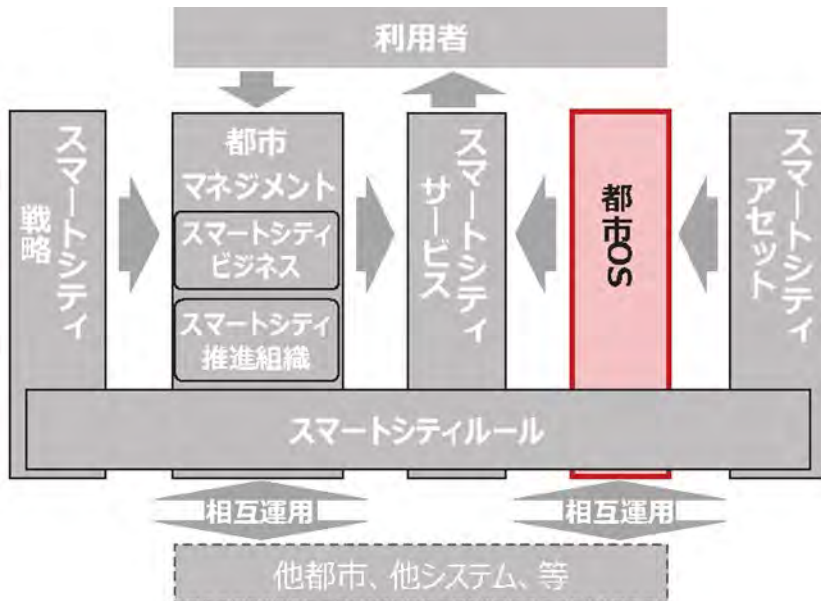


図 7.1-1 都市 OS の位置付け (従来のスマートシティ)

「都市 OS」は、「スマートシティアセット」や「他都市、システム等」から収集・取得したデータを、「スマートシティルール」に十分留意しながら、地域課題解決や地域ビジョンの実現を目的とした「スマートシティサービス」が「利用者」に対して提供されるように機能する(図 7.1-1)とされてきたが、この利用者は、ベンダーが決めた顧客としての利用者ではなく、本来は住民にとって透明性の高いプロセスで決定された利用者でなくてはならず、優先して解決すべき課題から順序だてて実行していくべきであり、最も収益を上げられる利用者という観点からのスマートシティサービスの検討に対しては警戒が必要である。

#### 7.1.1.1. 日本のスマートシティが解決すべき課題

日本のスマートシティが解決すべき課題として、以下の五つがあげられる。

##### 1) スマートシティサービスとデータ連携基盤のサイロ化

分野や組織ごとにサイロ化したスマートシティサービスや、分野毎のデータ連携基盤が乱立し、住民が個別にアクセスして利用していた。このため、本来のスマートシティが解決すべき課題である分野間を跨いだ人の課題については対処がされてこなかった。特にベンダーにとってはオープンデータが価値あるものと捉えられず、個人情報や困り込むビジネスモデルをスマートシティサービスとする傾向になり、サービスだけではなくデータ連携基盤自体のベンダーロックインも拡大し、分野間データ連携が更に困難となる状況を生んでいた。

##### 2) スマートシティで扱う課題のすり替え問題

従来の国内のスマートシティのサービス実装はベンダー志向で実装が進んでいた。スマートシティが求める分野間のはざまにある課題を発見する分野間データ連携の仕組みは構築されず、各地域で優先的に解決すべき課題の発見が置き去りになり、データに基づかずに課題設定が行われ、その利用者を想定したベンダーの収益モデルが優先して横展開する形態となっていた。人口減少がすすむにも関わらず、各地域でユーザが拡大することを想定した横展開ビジネスモデルが乱立し、国全体としての市場規模を度外視した無理筋な事業戦略が拡大した。

##### 3) デジタルツインのユースケース、ビジョンの不在

平成 31 年 3 月 13 日 Society 5.0 実現加速（スマートシティ）タスクフォース合意において「スマートシティとは、先進的技術の活用により、都市や地域の機能やサービスを効率化・高度化し、各種の課題の解決を図るとともに、快適性や利便性を含めた新たな価値を創出する取組であり、Society 5.0 の先行的な実現の場である。」とされ、Society 5.0「サイバー空間とフィジカル空間を高度に融合」したデジタルツイン環境を整える方針が打ち出され、洪水の浸水区域の予想などのユースケースが生まれたが、それ以外のデジタルツインの目的やユースケースが不在となり、デジタルツインの具体的戦略・ビジョンが見えなかった。

##### 4) 都市 OS の具体的なユースケースについての定義の不在とベンダーロックイン

都市 OS の役割は、「相互運用」「データ流通」「拡張容易」とされたが、具体的になんのための「相互運用」であり、「データ流通」であり、「拡張容易」であるかが明確ではなく、「何に役に立つのか」が明記されてこなかった。都市 OS 構築はスマートシティ事業の交付金当の採択の条件として記載されるために、各自治体が導入されるも導入後は使用されていないケースが多くみられ、都市 OS は「拡張容易」とされたがベンダーロックインされ、マルチベンダーによるメンテナンスができない状況の中で実際には「拡張容易性」の確保が困難となっていた。

##### 5) データ流通のプロトコルの定義の不在

データ連携基盤の目指すところは次世代インターネットであったが、通信基盤としての各レイヤーの要件やプロトコルの定義などがされてこなかった。都市 OS 自体がスタンドアロンで動く想定がされており、相互接続のためのルール（コード）の標準化がされてこなかった。

#### 7.1.1.2. 解決策

上記問題に対して以下の三つの解決策を提唱するものである。

##### 1) スマートシティサービスとデータ連携基盤のサイロ化への対策

スマートシティは分野間のはざまにある人の課題の解決をするものであり、各分野のデータを統一した地理空間上で可視化し、分野間データの連携によって発見された分野間の課題を住民と共有し、優先して解決すべき課題を住民に対して透明性高いプロセスにて、住民が参画しつつ選定し、その課題にフォーカスした解決策を、同様に透明性高いプロセスで決定する仕組みを都市 OS の要件とする。そのため、分野間を横断したデータを地理空間データ連携基盤上で可視化させることを都市 OS 導入前の要件とする。

## 2) スマートシティで扱う課題のすり替え問題への対策

スマートシティは分野間のはざまにある人の課題の解決をするものであり、各分野のデータを統一した地理空間上で可視化し、分野間データの連携によって発見された分野間の課題を住民と共有し、優先して解決すべき課題を住民に対して透明性高いプロセスにて住民が参画しつつ選定し、その課題にフォーカスした解決策を同様に透明性高いプロセスで決定する仕組みを都市 OS の要件とする。そのため、地理空間データ連携基盤上で分野間横断したデータ連携により課題が発見されたことを、スマートシティで解決すべき課題の要件とする。また分野間データ連携により発見された課題であるから、課題解決方法の導入後の効果検証についても、同じデータ取得方法を用い KPI 管理が都市 OS で自動的にできるものであることをスマートシティで解決すべき課題の要件とする。

## 3) デジタルツインのユースケース、ビジョンの不在への対策

スマートシティは「サイバー空間とフィジカル空間を高度に融合」したデジタルツイン環境を整え、Society 5.0 の先行的な実現の場であることをふまえ、内閣府が定義した「地理空間データ連携基盤」を実装し、分野間を横断したデータの可視化が地理空間データ連携基盤上でされていることを要件とする。各データ連携は API での連携により取得するものとし、地図データへのデータの埋め込みは数値 PNG を活用するものとする。

地理空間データ連携基盤で連携するデータは標準データとして、国土交通省の国土数値情報、不動産情報ライブラリ、国土地理院の地理院地図ベクター、逆ジオコーダー、DEM（標高）、林野庁の DMS（樹冠高）等が挙げられ、その仕様は SCRA 別冊「地理空間データ連携基盤」の最新版を常に参照するものとする。

動的データに関しては、EU の SynchroniCity や、DCAT-AP 等で実績のあるスマートシティに係る標準的な動的データの他、ジオテクノロジー社のトリマ等で見られるサンプリングデータの活用とアンケートによるウェルビーイングに関する人間の幸福度に関するデータの活用等の手法との連携も期待される。

上記等の分野間のデータ連携からサポートを開始し、住民との共有の為に 2D の地理空間データ連携基盤（別冊 SCRA「地理空間データ連携基盤」）を活用した動的データと静的データの見える化から整備を始めていることがスマートシティの要件となる。特に樹冠高のデータはドローン航路の検討の運輸分野や GX 分野など複数の分野で活用可能である。

## 4) 都市 OS の具体的なユースケースについての定義の不在とベンダーロックインへの対策

都市 OS は、地理空間データ連携基盤にてデジタルツイン環境が構築されていることを導入の前提とし、地理空間データ連携基盤で可視化された動的データや静的データについての AI 等による分析機能や KPI 管理機能を都市 OS にて提供する。特に動的データに関しては NGSI-LD による都市間連携における Broker 機能を具備するものとし、SynchroniCity 等を参考に標準的なデータカタログに準拠した動的データの相互接続を具備しているものとする。スマートシティ事業として導入されるサービス等は、SCRA 別冊にて定義した「地理空間データ連携基盤」上にて分野間データ連携したデータにより、分野間を横断した課題発見がされているものであるとし、地理空間データ連携基盤で可視化したデータを用いて KPI 管理が都市 OS でなされるものであるとし、都市 OS 上にてビルディングブロック方式にて都市 OS 機能としてマルチベンダーにて追加がされるものとする。都市 OS の GIS 機能は、別冊 SCRA「地理空間データ連携基盤」の要件を満たすものとし、地図データは腹持ちせず API 連携にて常に新しいデータが呼び出される仕様とする。

また都市 OS は地理空間データ連携基盤上に連携されたデータについて統合監視機能を具備することが望ましい。統合監視機能は各動的データについて、閾値を超えたものについて通知する機能となる。この統合監視機能は自治体の危機管理用に少なくとも 3 つの画面①発生中警報表示画面、②警報履歴画面、③ログ画面を有する。発生中警報表示画面は通知並びに警報を表示し、警報にはレベル（information、warning、fatal 等）を設定することが可能とする。また②の警報履歴画面は履歴検索機能を具備するものとする。③のログ機能についてもログの検索機能を具備するものとする。また各警報データに関しては、警報と同時に対策についてのマニュアルの個所を自動的に表示させる機能を具備させても良い。これらの警報レベルや警報内容、対応内容については、地理空間データ連携基盤の仕様に合わせて API を公開しなければならない。住民にむけた発報については SNS の



みに限定するようなことなく、マルチメディアでの対応とし、取り残される人が無いようにしなければならない。特に視覚障がい者、聴覚障害者、各種障害者、高齢者、未就学児、児童への対応について、ユーザ本人の意見を十分に反映させる必要がある。

また都市 OS はセンサーだけではなく、住民からの課題申請を登録できる機能を具備し、透明性高く住民と課題の共有、意見交換、課題の選定ができることを要件とする。パーソナルデータに関しては、完全分散管理するものとし、EU の S S I に準拠する完全分散型の方式とし、ブロックチェーン技術を用いたウォレット等によりデータの所有権を個人が完全に保有する方式を要件とする。特に個人データを扱う際は、クラウドサービスを提供する当該国の法令を調査の上、当該国の法令により個人データを当該国が取得できる場合等の場合、そのクラウドサービスは利用しないなどの対策を講じることを要件とする。

都市 OS は FIWARE 等の OSS を前提とし、運用や改修についてマルチベンダー対応ができることを要件とする。都市 OS 並びに導入された都市 OS 機能や、都市 OS にビルディングブロックで追加された機能について、要件定義書、基本設計書、詳細設計書、マニュアル、ソースコードについて開示し、他のベンダーが改修、流用可能とすることを要件とする。

### 5) データ流通のプロトコルの定義の不在

都市 OS の目指すところは次世代インターネット環境の構築であり、都市 OS がスタンドアロンで動作している状態は、従来のワークステーションがスタンドアロンで動作している状態と同じであると例えられる。都市 OS 間が次世代インターネット上で常に相互に API 連携して動作している状態を実現することにより、広域データ利活用の環境を整えることができ、広域の時系列の動的データが自エリアの予測に影響する等、自エリアの動的データが広域に影響する動的データである交通量や河川の水位などの標準データについて、EU の SynchroniCity を参考に NGSI-LD で相互接続できることを要件とする。

その上で、従来の要件についても、以下の図 7.1-2 を参考に考慮することが望ましい。

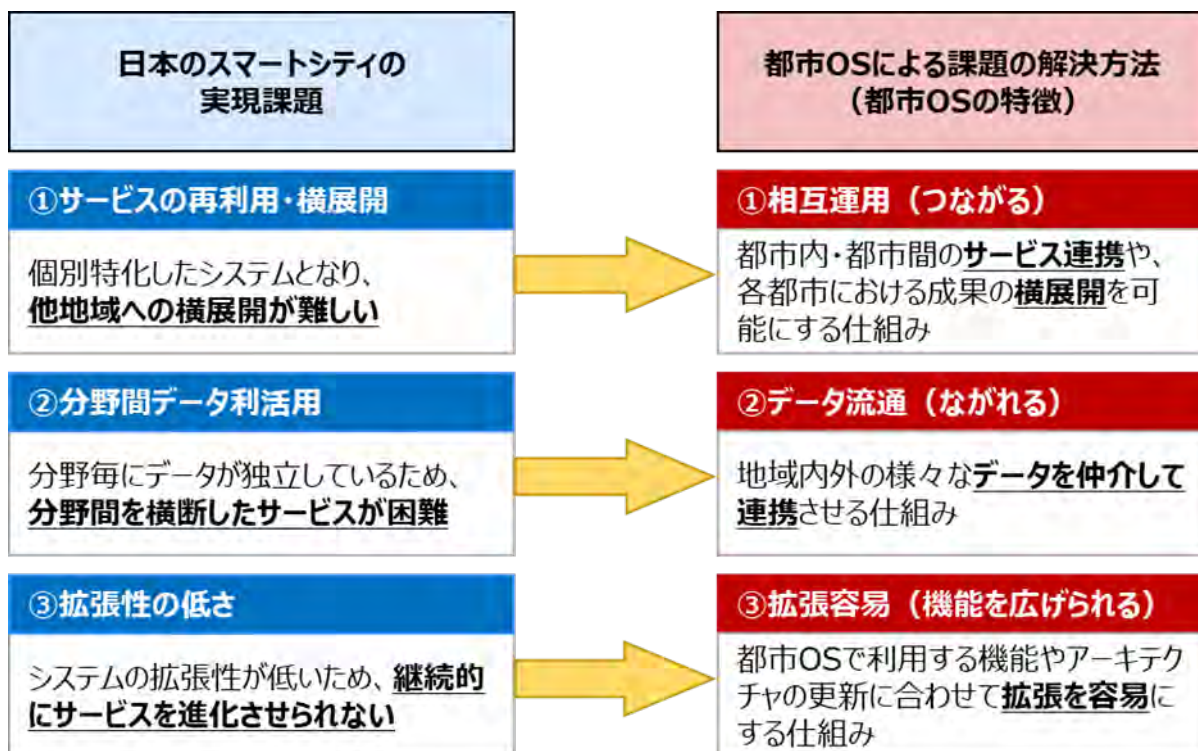


図 7.1-2 日本のスマートシティの課題と都市 OS による解決

### 7.1.1.3. その他の要件の詳細説明

#### 1) 相互運用（つながる）

日本の各地域は、単一都市・複数都市共同・複数都市/エリア分散等の様々な連携形態がある。どの連携形態においても成果の横展開を可能にするためには、都市 OS 間の相互運用が必要になる。都市 OS における相互運用とは、都市 OS が提供する API やデータが、同一形式あるいは機械的な変換により、各種スマートシティサービスや他都市 OS との連携が実現される状態のことである。

#### (1) 相互運用のための各種機能に対する方針

この相互運用を担保するには、以下の①～②の方針に則し、都市 OS の各種機能（API、データ）を実装することで、望めば互いに接続できる状態にすることが重要である。

##### (i) 標準化団体が定めた API やデータモデル等を積極的に採用

都市 OS は、各地域が解決する課題や目指すべき将来像に応じて、機能要件を満たす API を実装する。この際、標準化団体が定めた API 仕様やデータモデル等を積極的に採用することで相互運用が容易になる。

##### (ii) 多様な主体がアクセス可能なよう外部に公開する仕組み

都市 OS が実装した API やデータモデルは、開発ポータル等を用い、オープン API やオープンデータとして外部に公開し、都市 OS 利用者にとって使いやすい環境を整備する。都市 OS 利用者は公開されたオープン API やオープンデータを参照し、同一形式あるいは機械的な変換により、外部連携を実現することになる。

#### (2) 相互運用のメリット

都市 OS は「7.3.1 オープン API に用いられる個々の標準規格やシステムの事例」に示す機能要件や標準規格を参照して実装されることで、各連携形態において共通的に活用できるようになる。各種機能（API、データ）は、オープン API やオープンデータ（もしくは、関係者のみがアクセス可能なクローズド API やクローズドデータ）といった形で提供される。都市 OS の相互運用は以下の①～③の3つの利点がある。（図 7.1-3 参照）

##### (i) サービス連携

都市 OS が API を提供することで、都市 OS とスマートシティサービスの相互運用や、サービスとサービスの相互運用、都市・地域間のサービスの相互運用が可能になる。更に、構築したサービスを、特定の都市 OS だけでなく他都市 OS へ移植することも容易となる。

（例）サービス A からサービス B の機能を利用する連携。オープン API 機能によって、サービス間の機能連携が可能になる。

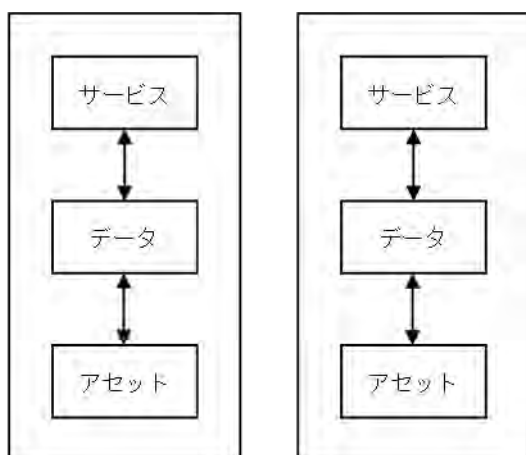
##### (ii) データ連携

都市 OS を通じて都市・地域内や都市・地域間でデータを流通させ、利用できるようにすることで、都市・地域内のサービスの質の向上都市・地域をまたいだ広域でのサービスの提供、住民の利便性の向上、さらに地域の特性を分析することによる地域に根差した新たなビジネス及び産業の創出への貢献も期待される。詳細は、次項の「(2) データ流通」を参照。

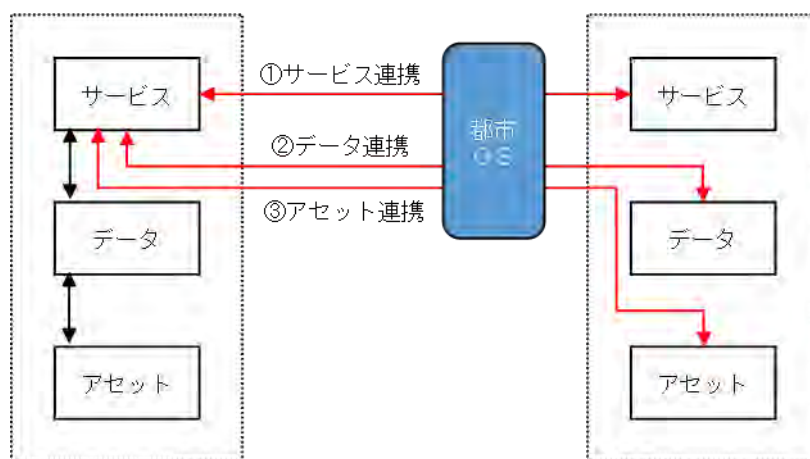
（例）サービス A によって用いられている都市 OS に格納されているデータを、別のサービス B から利用できるようにする連携。データが都市 OS に格納されている場合は（データ蓄積型）、サービス B が都市 OS にアクセスすれば連携できる。データが都市 OS には格納されておらずサービス A のシステムに分散して格納されている場合は（データ分散型）、サービス B が都市 OS のデータ仲介機能にアクセスすることで、サービス A のデータが API 経由で取得されて、サービス B に提供される。

##### (iii) アセット連携/他システム連携

都市 OS がオープン API 機能を提供することで、地域内の多様なスマートシティアセットや、他システムが連携し、組織やシステムの壁を越えた都市・地域や分野を横断したサービスが提供可能になる。



これまでの日本のスマートシティ  
相互運用しない／つながらない



都市OSを用いて相互運用する／つながる

図 7.1-3 日本のスマートシティの相互運用性の課題と都市 OS による解決

(例) サービス A のために設置され都市 OS が管理しているアセット (デバイス) を、別のサービス B から利用できるようにするための連携。この連携には、2 通りの方法がある。第 1 に、アセットから取得したデータを利活用するための連携である。これは、上記のデータ連携の仕組みで実現できる。第 2 に、アセットのデバイスの連携である。これは、アセットにオープン API 機能を備えることによって、他のサービスから利用できるようになる。サービス連携と同じ仕組み実現できる。

## 2) データ流通 (ながれる)

スマートシティにおいて、分野や組織の壁を越え、課題解決のために分野横断型のサービスが提供される必要があり、その実現のため都市 OS は異種データ (都市 OS 内外の多種多様なデータ) を流通させる機能を持つ必要がある。この異種データの流通を実現するための機能を「データ仲介機能」もしくは「Broker」と呼び、( i ) 多種多様なデータの取り扱い、( ii ) 都市 OS 内外のデータを仲介する必要がある。なお、ここでいうデータ流通には API 等を通じたデータへのアクセスも含む。ファイル転送ではなくデータへのアクセスにすることでデータの流通量を減らし、最新の情報を確実に入手することができる。

## (1) 多種多様なデータの取り扱い

都市 OS が取り扱うデータには、特性の異なる様々なデータ種別が存在し、地域が解決する課題に応じたデータを、その特性に従い適切に管理する必要がある。都市 OS が取り扱うデータ種別の例を以下に示す。

### (i) メタデータ

データ本体（静的データ、動的データ、パーソナルデータ等）を効率的に管理したり検索したりするために、データ本体についてのデータモデル（データ項目や形式）や属性情報等を記述した付帯データ。データの分類として、コンテキストデータ（データの関連付け）や、データカタログが挙げられる。メタデータを通じてデータ本体を検索し、各種データ種別の特性に応じアクセスするといった運用がなされる。

### (ii) 静的データ

更新頻度が比較的少なく、長期間保存、参照されるデータ。データの分類として、統計データ、分析データ、履歴データ、文書データ等がある。オープンデータ化すべきものが多く、オープンデータの取り扱いに関するルールに従い、主に自治体が保有データをオープンデータとして公開し活用促進に取り組むことが必要となる。

### (iii) 動的データ

更新頻度が高く、リアルタイムに生成される時間軸に連続したデータ。データの分類として、センサデータ、履歴データ、人流データ、ストリーミングデータ等がある。リアルタイムに変わるデータであるため、時刻情報の付与や履歴管理等が必要となる。

### (iv) 地理空間データ

空間上の特定の地点または区域の位置に関する情報（位置情報）を持つデータ。位置に関連づけられた様々な事象に関する情報も含まれる。データの分類として、地形図、空中写真、衛星画像等の地理空間情報、建物や土木構造物に関連する BIM(Building Information Modeling) データ、CIM(Construction Information Modeling) データ等がある。2 次元の地図情報と地理空間データは、緯度経度情報がマッチングキーである。

### (v) パーソナルデータ

個人情報に加え、個人情報との境界が曖昧なものを含み、個人の属性情報、移動・行動・購買履歴、ウェアラブル機器等から収集されたデータあるいは加工された情報等個人と関係性が見出される広範囲のデータ。データの分類として、要配慮個人情報、個人情報、匿名加工情報がある。パーソナルデータの取り扱いに関するルールに従い、プライバシー保護と高度なセキュリティ対策が必要となる。パーソナルデータは、個々人からの許諾によって取得可能な場合もあり、パーソナライズされた最適な情報を得ることができる。

これらの多様なデータは、技術的な性質や、社会における求められる取り扱い方に違いがあることから、データへのアクセス方法や操作方法がばらばらになってしまい、多様なデータを利用したサービスを構築することは困難になる。そこで、図 7.1-4 のように、データ仲介機能を提供することによって、共通の API スタイルで、それらのデータを扱う（生成・閲覧・更新・削除）ことを可能にし、多様なデータを利用したサービスの構築を容易化できる。



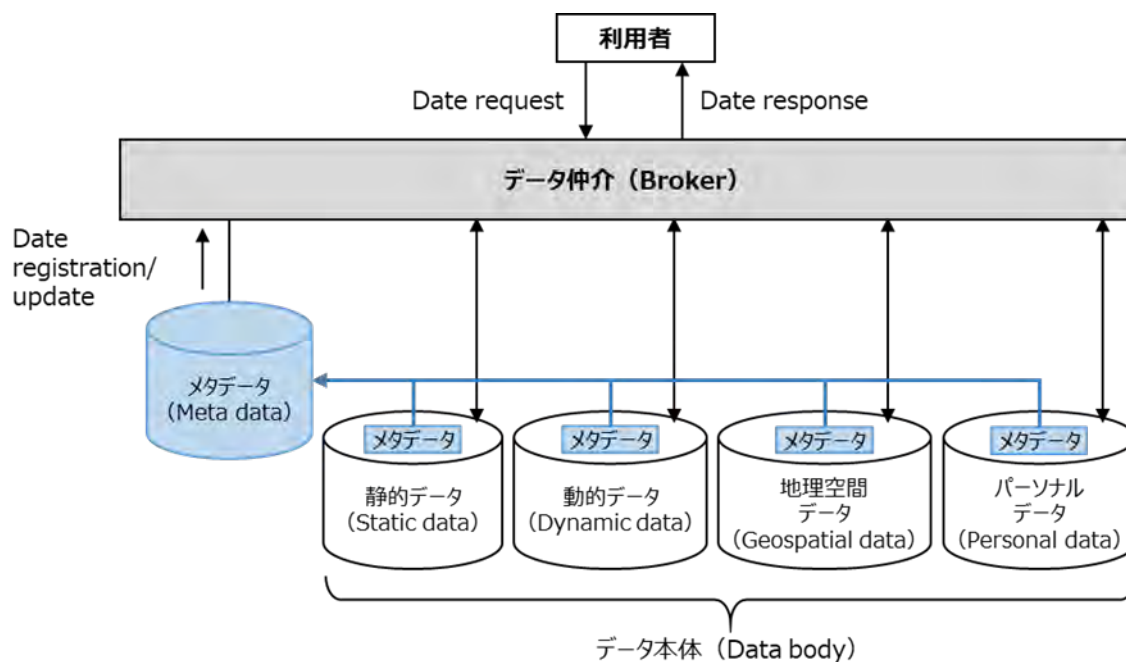


図 7.1-4 都市 OS が扱う異種データとデータ仲介

## (2) 都市 OS 内外のデータを仲介

データ仲介 (Broker) は、都市 OS 内外に点在するデータも仲介する必要がある。この仲介は、データ蓄積とデータ分散の二つの方式に分類される (図 7.1-5)。ユーザは、どちらの仲介方式かを区別することなく、データアクセスが可能となる。

### (i) データ蓄積方式 (集中型)

都市 OS にデータを蓄積し、一元的に管理する。

### (ii) データ分散方式 (分散型)

都市 OS に実体データは存在せず、都市 OS は分散されたデータの所在情報 (URL 等のエンドポイント情報) を管理する。都市 OS は所在情報を利用し、ユーザからのデータアクセスに対して、データの仲介をする。

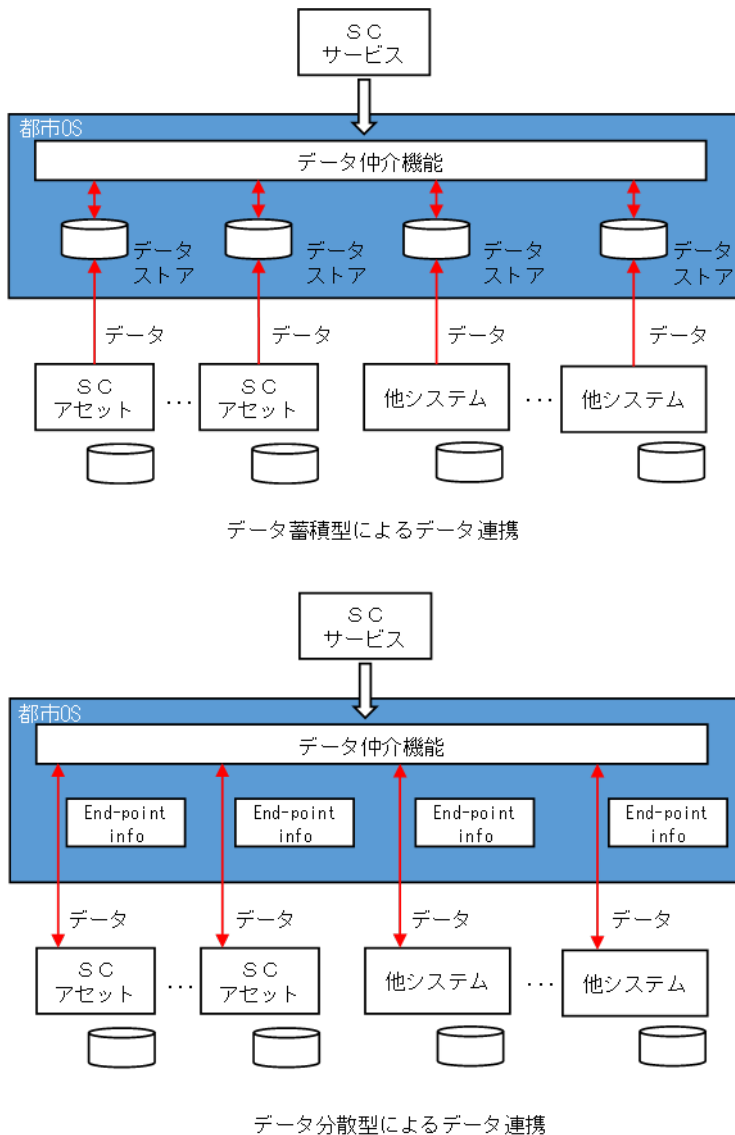


図 7.1-5 データ蓄積方式とデータ分散方式によるデータ連携

### 3) 拡張容易（機能を上げられる）

スマートシティにおいて、地域が解決する課題や目指すべき将来像に応じ、将来の機能追加や更新を継続的に  
行える必要があり、例えばビルディングブロック方式といった機能の組み換えを柔軟に対応できる仕掛けを持つ必要  
がある。

ビルディングブロック方式とは、機能群の中から必要な機能を取捨選択し積み重ねることで、疎結合なシステム  
構築を行う方式のことである。ある程度まとまった機能のかたまりをビルディングブロックと呼ぶが、ビルディングブロック間  
のやり取りを API として統一もしくは公開するマイクロサービスによる構築が、都市 OS としての相互運用できるため  
望ましい。これにより、他のビルディングブロックやサービスへの影響を最小限とした更新が可能となり、保守性が向上  
する。さらに、最初はスモールスタートでの構築にとどめ、地域が解決する課題や目指すべき将来像に応じて少しずつ  
機能拡張していくことも可能となる。将来的には、連携するスマートシティサービスやスマートシティアセットの追加や、  
提供する各種機能の入れ替え及び変更等を、視覚的な操作で容易に組み替えできることが実現可能となり、結  
果的にベンダーロックインを回避することにも繋がる。

また、都市 OS が特定ベンダーに依存しないよう、なるべくオープンソースを使うことで、より透明性を持たせること  
が望ましい。その中で、ビルディングブロック間の各種 API のオープン化を推進し相互運用性を確保することで、多様  
な主体が都市 OS の一部として参画することも可能となり、都市 OS 及び地域の発展にもつながることになる。ただ

し、地域の発展に伴い都市 OS も拡張することにより、複雑化・ブラックボックス化等のシステム課題が顕在化する場合もある。将来的に拡張を容易にして相互運用できるよう、機能設計や運用を考慮することが望まれる。ビルディングブロック方式による構築例を図 7.1-6 に示す。

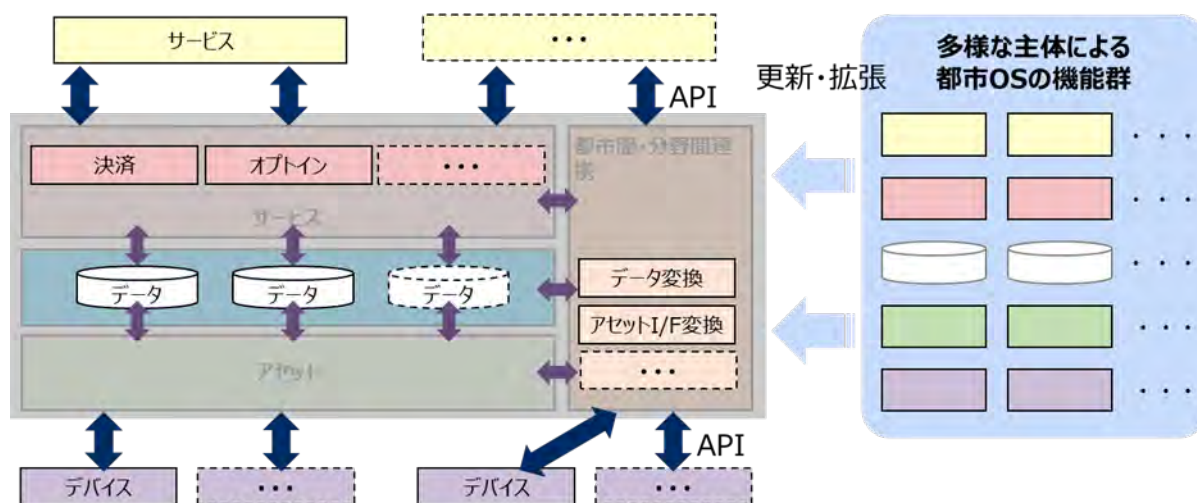


図 7.1-6 ビルディングブロック方式による IT システム構築例

## 7.1.2. 都市・地域「内」連携と都市・地域「間」連携

スマートシティは日本の各地で多く取り組まれており、都市・地域を超えて相互運用（つながる）・データ流通（ながれる）・拡張容易（機能を広げられる）を実現するために連携することは重要である。

都市・地域内では、共通したガバナンスのもとで、スマートシティアーキテクチャのなかに、上記で述べたように都市 OS を導入することで実現することが可能である。それによって、都市 OS が提供する、オープン API 機能やデータ仲介機能（ブローカー）を用いることによってサービス・データ・アセットの連携が実現できる。

一方、都市・地域を超えた連携は、都市 OS がカバーするエリアを超えた外部との連携である。連携先の外部とは、共通のガバナンスを持たず、都市 OS の有無、都市 OS が導入されていたとしてもその種類、サービスやアセットの種類、データの種類の種類、採用している技術標準等がまちまちである可能性がある。

もちろん、スマートシティのサービス提供者やアセット管理者などの個々のステークホルダーが、個別に連携することもできるが、都市・地域内に多くあるサービスやアセット、データを個別に連携することは効率的ではない。そこで、こうした都市・地域間連携に必要な機能に関しても、都市 OS などによる支援があることが望ましい。そこで、都市・地域間で、サービス・データ・アセットを連携する方式は、主に 4 つに分類される。（図 7.1-7）。

### 7.1.2.1. 個別連携型（都市 OS 支援なし）

個別連携型は特に都市 OS 等による支援がなく、個々のステークホルダーが、他の都市が提供するスマートシティに関する機能を直接利用する方法である。連携の数が少ないときは可能であるが、数が多くなると効率的ではないし、セキュリティ上も脆弱になる恐れがある。

### 7.1.2.2. 都市 OS 共有型（集中型連携）

広域の複数の都市・地域で、共通の都市 OS を共有する密結合型の連携方式である。同じ都市 OS を利用するため、都市・地域内連携と同じ方法で都市・地域間連携も実現できる。データ蓄積型によるデータ連携方式と同様の方式である。ただし、これを実現するためには、広域の複数の都市・地域間で、密なガバナンス上の連携が

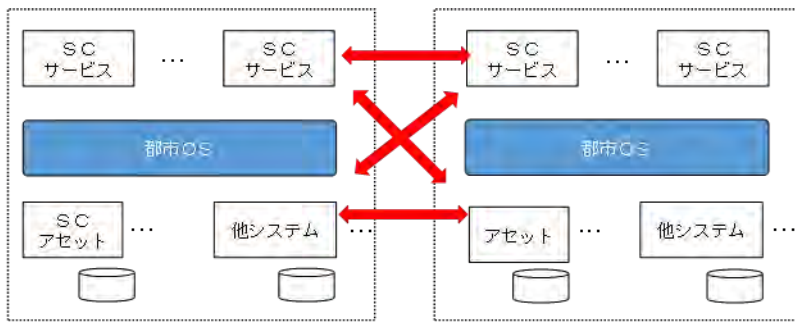
不可欠である。例えば、都道府県（または、広域自治体連合の枠組み）で共通の都市 OS を導入し、個々の基礎自治体がそれを共通に用いて、サービスやアセットを構築・運営することが挙げられる。

#### **7.1.2.3. 都市 OS 連携型（分散型連携）**

個々の都市・地域で導入した個々の都市 OS との間で API を介した疎結合型の連携方式である。共通のガバナンスを持たないことによる各種変換機能や連携機能は、各都市 OS が提供するもののいずれかを利用するか、それらも都市 OS 間で連携することが必要である。データ分散型によるデータ連携方式を、都市・地域間の連携に拡張した方式の 1 つであると位置づけられる。例えば、都市・地域内連携に用いられる、データ仲介機能（ブローカー）やオープン API 機能を、都市・地域外にも適用することで実現できる。

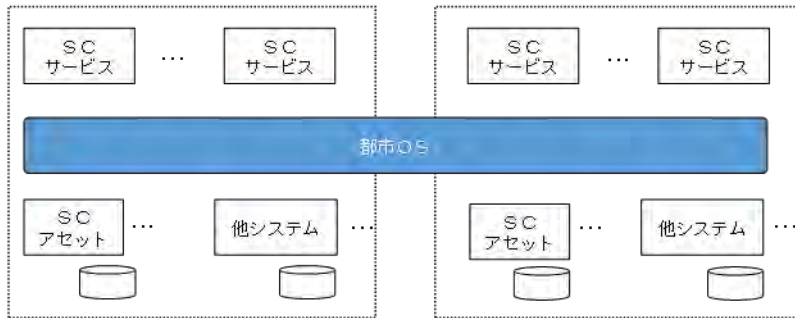
#### **7.1.2.4. 連携基盤利用型（連邦型連携）**

都市・地域間／分野間連携基盤（以下、連携基盤）がハブとなり、個々の都市・地域で導入した個々の都市 OS がこのハブと接続することで、間接的に都市・地域間が連携される方式である。共通のガバナンスを持たないことによる各種変換機能や連携機能は、ハブとなる連携基盤が提供し、それを、各都市・地域の都市 OS が利用する。データ分散型によるデータ連携方式を、都市・地域間の連携に拡張した方式の 1 つであると位置づけられる。



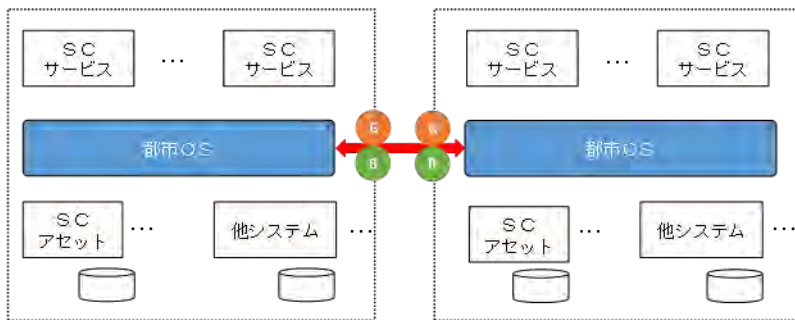
都市・地域A 都市・地域B

① 個別連携型（都市OS支援なし）



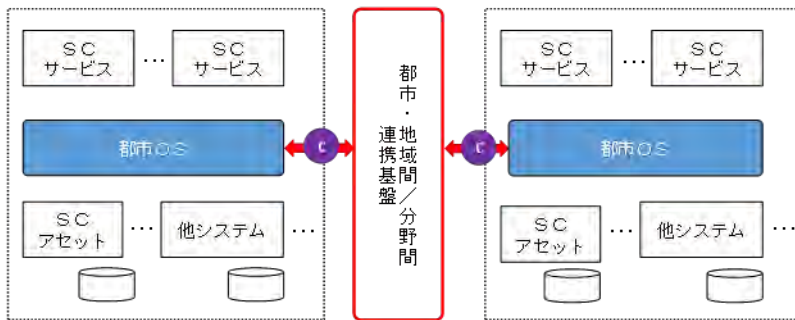
都市・地域A 都市・地域B

② 都市OS共有型（集中型連携）



都市・地域A 都市・地域B

③ 都市OS連携型（分散型連携）



都市・地域A 都市・地域B

④ 連携基盤利用型（連邦型連携）

- オープンAPI機能（APIゲートウェイ）
 ● データ仲介機能（ブローカー）
 ● コネクタ

図 7.1-7 都市・地域間連携、分野間連携の方式



### 7.1.3. 都市や地域の実情にあわせた都市 OS の実装のパターンの例

都市 OS には、スマートシティの実現にあたって共通的に活用する機能が集約されている。そのため、都市または地域がスマートシティに取り組むに際しては、都市 OS を、基本的開発環境及び運用環境として導入・活用することが望ましい。

他方、スマートシティは、全国各地域において一意に取組分野の多少や取組エリア範囲の大小が定まるものではない。また、同じ地域であっても、取組の時間軸によってこれらは可変である。そのため、都市 OS については地域のその時点での実状に即する形で導入することが望ましい。ここでは、地域の実状に応じた都市 OS の実装パターンとして3つの例<sup>62</sup>を提示する。

具体的には、「単一都市・地域内で、パーソナルデータは活用しないスマートシティ」（例 A）、「単一都市・地域内で、パーソナルデータを活用するスマートシティ」（例 B）、「複数の都市・地域に跨って取り組むスマートシティ」（例 C）が挙げられる。なお、例 C は、「複数の都市・地域で個別に都市 OS を活用しているが、それらをブローカーまたはコネクタで連携することで相互にシナジーを効かせたスマートシティサービスを実装するケース」（都市 OS 間連携）と、「複数の都市・地域で同一の都市 OS を共同利用することで相互にシナジーを効かせたスマートシティサービスを実装するケース」（都市 OS 共同利用）の2つに大別される。

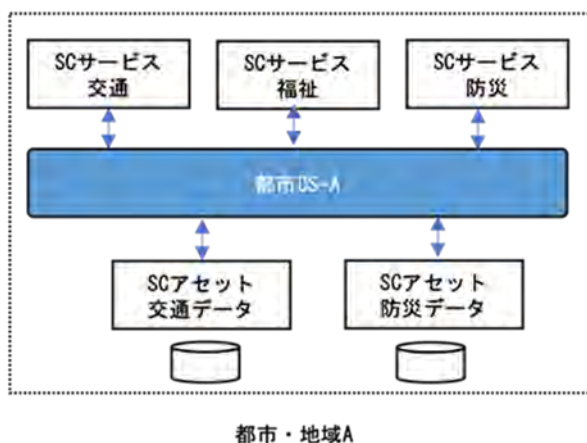
#### 例 A: 地域内データ連携（パーソナルデータを含まない）

##### 概要：

一つの都市または地域が、単一の都市 OS 上で、複数分野の資産やサービスを連携させて、複数のサービスを提供する（図 7.1-8）。

##### 防災におけるユースケース例：

都市・地域 A 内で、交通データと防災データを連携させることによって、交通や福祉、防災等のスマートシティサービスに活用する。



例A：地域内データ連携（パーソナルデータを含まない）

図 7.1-8 例 A: 地域内データ連携（パーソナルデータを含まない）

<sup>62</sup> 実装パターンとは、都市 OS 実装の例示であり、地域の実情に応じたその他の取組や、例 B や C から実装を開始することや、例 A を当面のゴールとして実装を進めること等を否定するものではない。

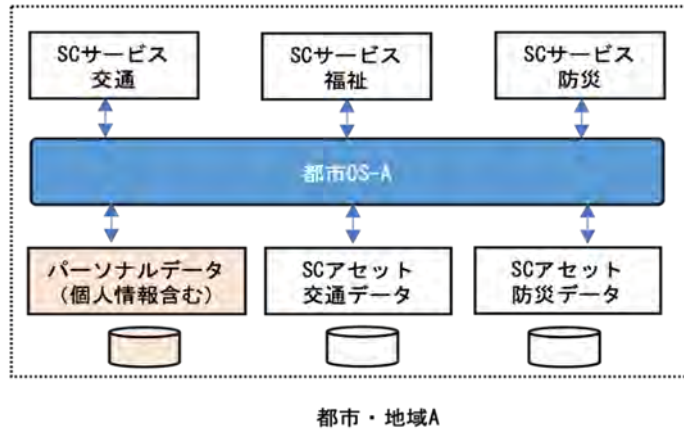
## 例 B: 地域内データ連携 (パーソナルデータを含む)

### 概要:

一つの都市または地域が、単一の都市 OS 上で、パーソナルデータを含む、複数分野のアセットやサービスを連携させて、複数のパーソナライズされたサービスを提供する (図 7.1-9)。

### 防災におけるユースケース例:

都市・地域 A 内で、防災データとパーソナルデータを連携させることによって、避難行動要支援者へのアラート発動等のパーソナライズされたスマートシティサービスの提供が可能になる。



例B: 地域内データ連携 (パーソナルデータを含む)

図 7.1-9 例 B: 地域内データ連携 (パーソナルデータを含む)

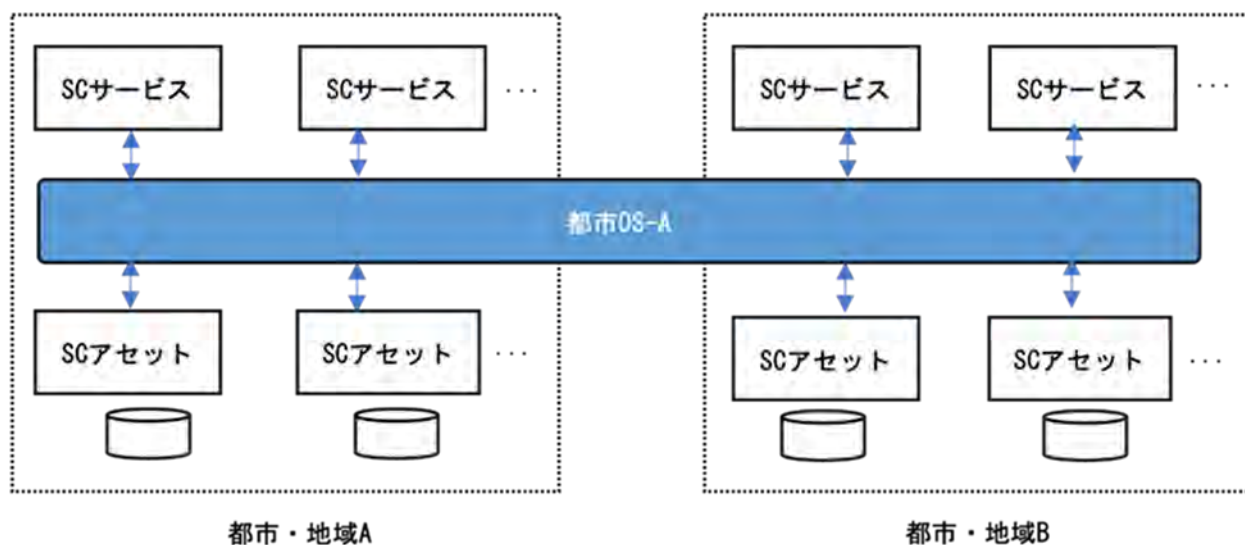
## 例 C: 地域間データ連携

### 概要:

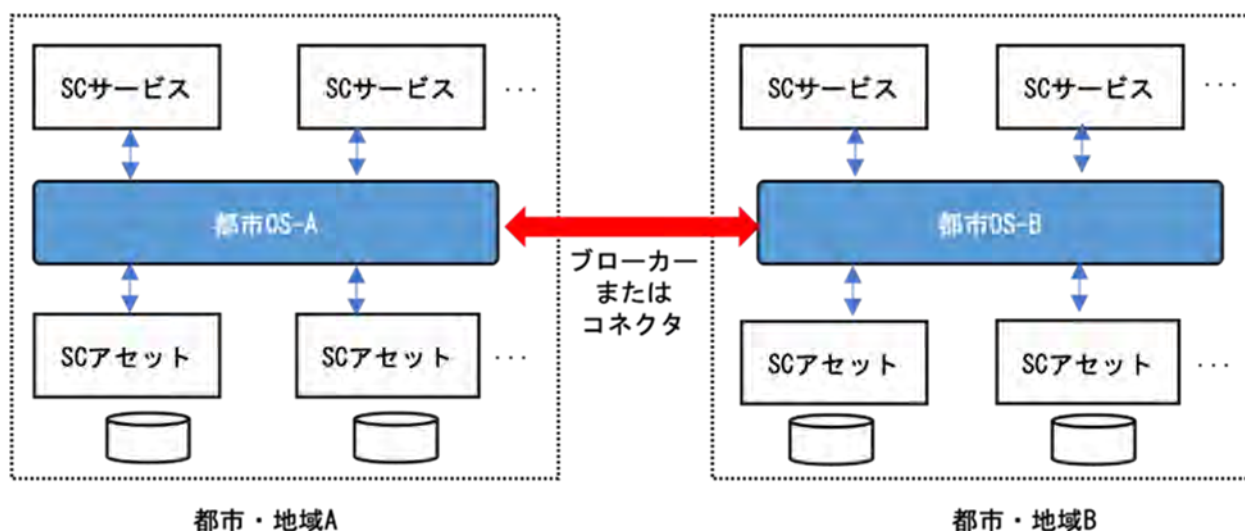
複数の都市または地域が、都市 OS を介して連携し、各都市・地域のサービスやアセットを相互活用したサービスを提供する。都市 OS を共同利用する集中型方式と、都市 OS 間をブローカーやコネクタを用いて連携させる分散型方式がある（図 7.1-10）。

### 防災におけるユースケース例:

下線の上流に位置する都市・地域 A の防災データ（雨量や水位データ）と、下流に位置する都市・地域 B の交通データ（人流・交通流データ）を連携させることによって、都市・地域 B の地区別により精度の高い避難指示が可能になる。



例C-1: 都市OS共同利用



例C-2: 都市OS間連携

図 7.1-10 例 C: 地域間データ連携

## 7.1.4. スマートシティの成長に合わせた都市 OS の成長ステップの例

前項の例 C の事例として、広域自治体の事例を参考に、スマートシティの成長にあわせて都市 OS やスマートシティサービス等が成長していくステップの例を図 7.1-11 に示す。

### ステップ0（当初）

A 都市が所有するデータを、データコンバータを介してデジタル庁の推奨する自治体標準オープンデータセットで定義されている項目列に変換し、ダッシュボード上でオープンデータを提供する。この時点では都市 OS は導入されていない。

### ステップ1（現在）

広域自治体が傘下の市町をとり纏めて、複数都市が所有するデータを、データコンバータを介してデジタル庁の推奨する自治体標準オープンデータセットで定義されている項目列に変換し、都市 OS が提供するダッシュボード上でオープンデータを提供する（地域間連携）。

### ステップ2（将来発展）

広域自治体が傘下の市町をとり纏めて、複数都市において都市 OS を共同利用し、複数のスマートシティサービスを提供する（分野間連携 + 地域間連携）。

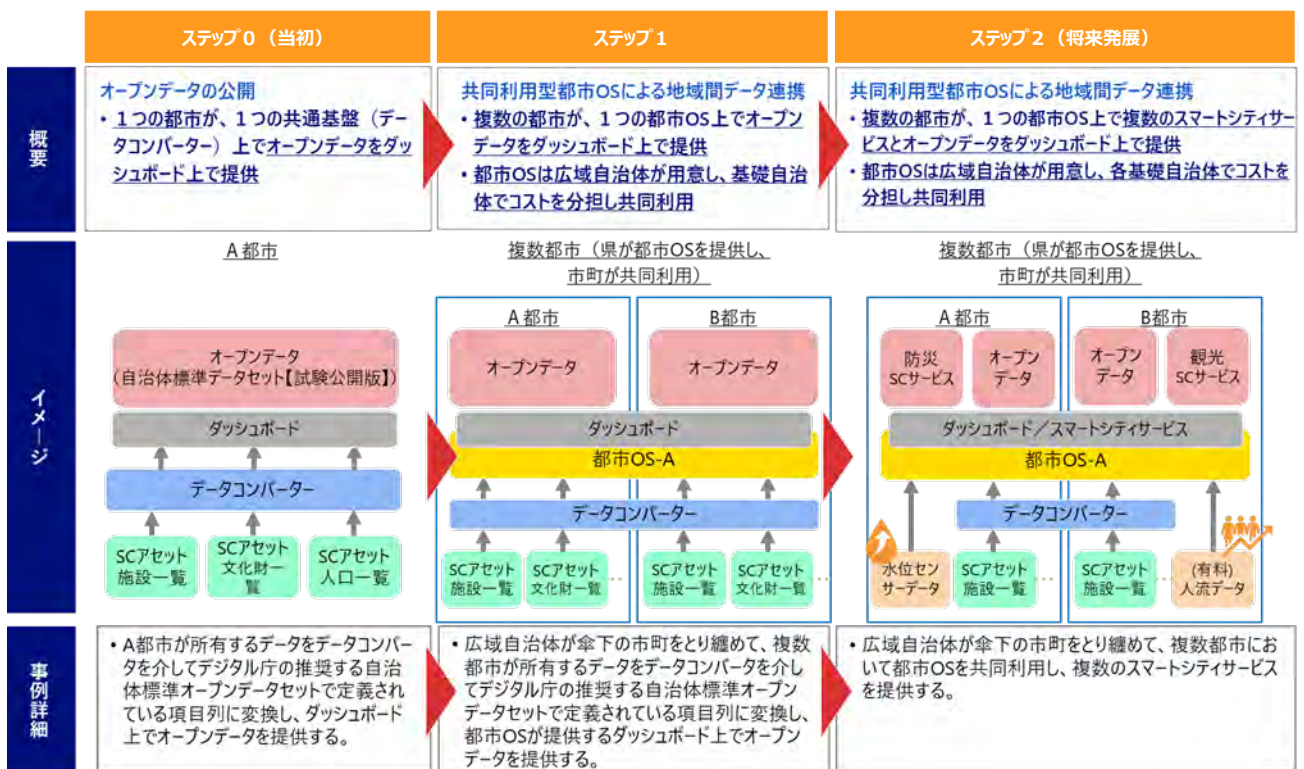


図 7.1-11 地域の実状にあわせた、都市 OS の実装パターンの例示

ステップ1の最新動向として、デジタル庁は「共同利用ガイドブック<sup>63</sup>」を公開し、広域自治体による基盤の共同利用を推進する方針を明らかにした。このガイドブックの本編4章とAppendix5章には、図中におけるステップ2に相当する分野横断サービスの実現に関するプロセスについても記載されている。

## 7.2. 都市 OS の機能

---

### 7.2.1. 都市 OS の構成要素（機能群）

都市 OS の構成要素は、第三版までは従来の第二版を踏襲したデータ流通に関するマネジメント機能群の記載を残すが、第四版以降はデジタルツイン（Y.4600）の構成に基づき、都市マネージメント機能としての都市 OS の姿に再構築を行うものとする。以下は従来の第二版の踏襲分として記載をする。

都市 OS は、都市サービスや都市のアセット（地域で活用している各種デバイスやシステム等）、他システム・他都市 OS から収集したデータを保管し、都市内外の都市サービスと連携する役割を担っている。日本のスマートシティに対する考え方と求める都市 OS の特徴を踏まえ、また、海外のスマートシティアーキテクチャについても参照のうえ、本スマートシティリファレンスアーキテクチャにおける都市 OS の構成要素（機能群）を整理した。整理の結果、都市 OS は「サービスマネジメント機能群」、「データマネジメント機能群」、「アセットマネジメント機能群」の3種類の基本機能群に、「運用支援機能群」、「セキュリティ機能群」、「地域内連携機能群」、「地域間・分野間連携機能群」の4種類の共通機能群を加えた、7つの**機能群**から構成される。各機能群は、**個別機能**をひとまとまりとした一つ以上の機能ブロックから構成される。都市 OS の機能の全体像を図 7.2-1 に、また都市 OS の7つの機能群の定義を以下に示す。

---

<sup>63</sup> [本編\\_データ連携基盤の共同利用ガイドブック](#)



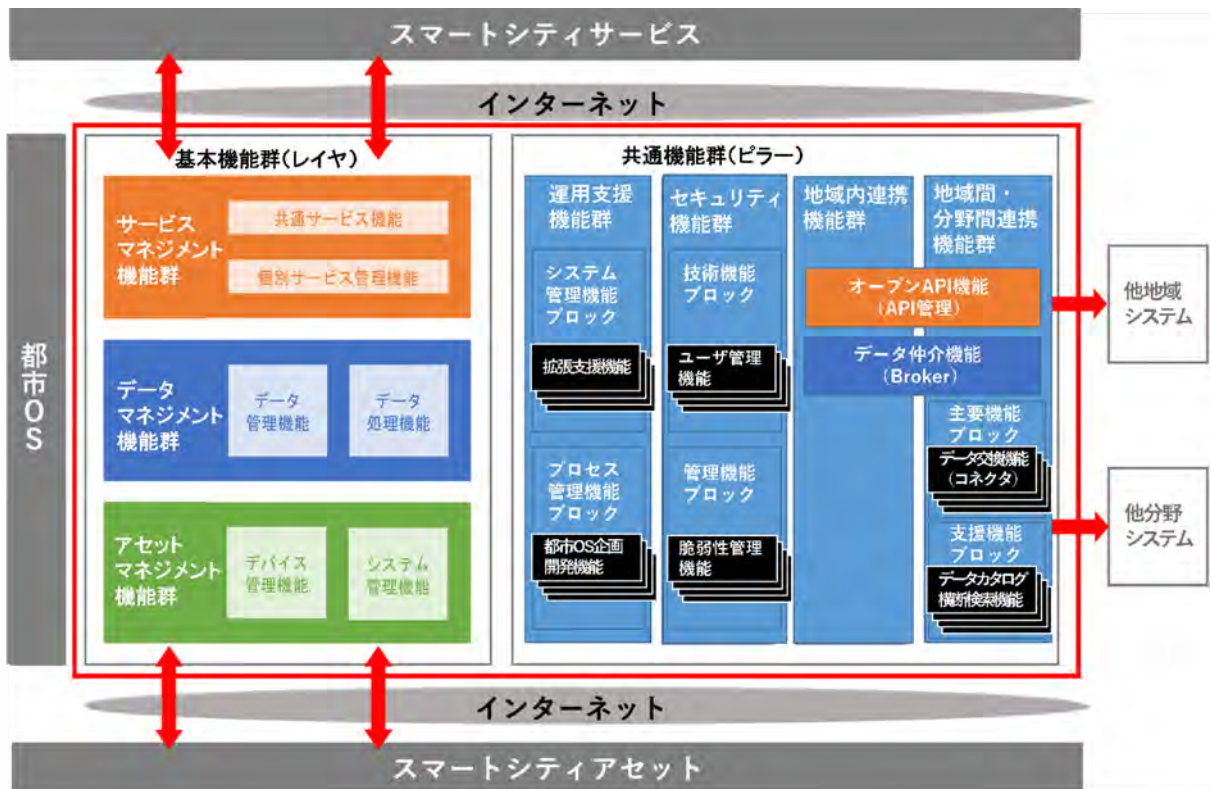


図 7.2-1 スマートシティリファレンスアーキテクチャにおける都市 OS の機能の全体像

### 7.2.1.1. 基本機能群 (レイヤ)

#### 1) サービスマネジメント機能群

サービスマネジメント機能群は、都市 OS 上で動作する各種スマートシティサービスが、都市 OS や他のスマートシティサービスと連携するための機能群である。

##### (1) 共通サービス機能ブロック

共通サービス機能ブロックは、地域や分野を問わず共通的に使われ、協調領域として都市 OS が提供する複数のサービスの機能を提供する。

##### (2) 個別サービス管理機能ブロック

個別サービス管理機能ブロックは、都市 OS と連携するスマートシティサービスを管理し、適切に運用することを支援する機能を提供する。

#### 2) データマネジメント機能群

データマネジメント機能群は、都市 OS に都市データを保存・蓄積・管理し、地域内連携支援機能や地域間・分野間連携支援機能と連携して、単一都市・複数都市や、他システムに分散されたデータの仲介を実現する機能群である。

##### (1) データ管理機能ブロック

データ管理機能ブロックは、サービス連携や、外部データ連携を通じて収集したデータを、分類や形式を問わず保存・蓄積する機能を提供する。

##### (2) データ処理機能ブロック

データ処理機能ブロックは、(スマートシティ) サービスや (スマートシティ) アセットから取得したデータの差異を吸収し、標準データモデル等へ変換する機能を提供する。

#### 3) アセットマネジメント機能群

アセットマネジメント機能群は、データの収集、及び、接続するスマートシティアセットや他システムに登録・削除

等の管理と、スマートシティアセットへの制御を実行する機能群である。

#### (1) デバイス管理機能ブロック

**デバイス管理機能ブロック**は、都市 OS に接続するデバイス（アセット）の状態を管理・監視し、システム管理者がデバイスの接続異常等を検出可能にするための機能を提供する。

#### (2) システム管理機能ブロック

**システム管理機能ブロック**は、都市 OS に接続する他システム（アセット）の認証情報、接続情報、契約情報等、システムに接続するために必要な情報を管理する機能、及び、システム管理者が他システムとのデータ連携状態や接続状態等の状態を管理する機能を提供する。

### 7.2.1.2. 共通機能群（ピラー）

#### 1) 運用支援機能群

**運用支援機能群**は、都市 OS の維持・発展に必要なシステム管理や管理プロセスを提供する機能群である。

##### (1) システム管理機能ブロック

**システム管理機能ブロック**は、都市 OS が想定外の要因によって停止することなく、各種機能を安定して提供するため、都市 OS を維持管理するための機能を提供する。

##### (2) 個別サービス管理機能ブロック

**プロセス管理機能ブロック**は、都市 OS の高度化に向けた取組を継続的に進めるために、必要なプロセスやルール等の規定を支援する機能を提供する。

#### 2) セキュリティ機能群

**セキュリティ機能群**は、都市 OS の外部/内部の脅威から都市 OS を防御するために必要な機能を提供する機能群である。

##### (1) 技術機能ブロック

**技術機能ブロック**は、技術的対策に含まれる認証、暗号化、不正アクセス防止、不正アクセス検知・遮断技術等が含まれ、都市 OS として具備すべき機能を提供する。

##### (2) 管理機能ブロック

**管理機能ブロック**は、管理的対策に含まれる、教育、ルール整備等の人的対策、脆弱性管理、セキュリティ監査等の組織的対策、入退場管理、破壊防止等の物理的対策から構成され、都市 OS の運用・管理に必要な機能を提供する。

#### 3) 地域内連携機能群

**地域内連携支援機能群**は、同一都市・地域内のスマートシティにおける、サービス連携やデータ連携、アセット／他システム連携を実現するための機能群である。

##### (1) オープン API 機能ブロック

**オープン API 機能ブロック**は、ビルディングブロック型で構築されたスマートシティのシステムにおいて、分散して稼働している都市 OS や（スマートシティ）サービス、（スマートシティ）アセット、等の各ブロックが提供する API を仲介するために、API をオープン化する機能を提供する。

##### (2) データ仲介機能（ブローカー）ブロック

**データ仲介機能ブロック**は、都市 OS 内外に存在するデータの所在情報を管理することで、データアクセスを仲介する機能を提供する。

#### 4) 地域間・分野間連携機能群

**地域間・分野間連携機能群**は、他の都市・地域のスマートシティや、他の分野のサービスやデータと連携するための機能群である。

**(1) 主要機能ブロック**

**主要機能ブロック**は地域間・分野間で連携する際の主要なサービスを提供する複数の機能を提供する。

**(2) 支援機能ブロック**

**支援機能ブロック**は主要機能ブロックが効率的・効果的に実現されるために提供する複数の機能を提供する。

次項以降、機能群毎に、各機能ブロックと個別機能の概要を説明する。なお、都市 OS の個別機能は以降で紹介するように非常に多岐に亘るが、各地域が都市 OS を導入するに際しては、当該地域が解決する課題や目指すべき将来像を踏まえたスマートシティ像を念頭に置き、必要な都市 OS の個別機能を取捨選択して実装すれば良い。

## 7.2.2. スマートシティの課題解決策と各機能群との関係

第 **エラー! 参照元が見つかりません。** 節では、日本のスマートシティの課題として、(1) サービスの再利用・横展開、(2) 分野間データ利活用、(3) 拡張性の低さ、の3つを挙げた。それぞれの課題の解決策として、(1) 相互運用（つながる）、(2) データ流通（ながれる）、(3) 拡張容易（機能を上げられる）、の3つを挙げ、それらを実現する手段として都市 OS の導入を述べた。

本節では、これらの課題・解決策と、スマートシティアーキテクチャの各機能群および各機能ブロックとの関係を説明する。

### 7.2.2.1. 相互運用（つながる）

相互運用（つながる）を実現する主要な機能群は、地域内連携機能群と地域間・分野間連携機能群である。相互運用の対象となる（つながられる）側では、提供する機能に API を備えることで、同一都市・地域内の多様なサービスや機能から、また都市・地域外や他の分野の多様なサービスや機能からも API を介して相互運用することが可能になる。この相互運用において、セキュリティやトラストを実現するためには、セキュリティ機能群の役割が不可欠である。

### 7.2.2.2. データ流通（ながれる）

データ流通（ながれる）を実現する主要な機能群は、データマネジメント機能群と、地域内連携機能群、地域間・分野間連携機能群である。スマートシティで扱うデータのうち、都市 OS に集約できるものに関しては、データマネジメント機能群によって蓄積・保存・管理される。これを、各種サービスやアセットがこのデータマネジメント機能群を利用することによって、都市 OS 内のデータ流通を実現することができる。一方、都市 OS に集約できないデータについては、同一都市・地域内に存在する場合は、地域内連携機能群によって、また、他の都市・地域や、他の分野のデータの場合は、地域間・分野間連携機能群によって、API を経由してデータ流通を実現することができる。このデータ流通において、セキュリティやトラストを実現するためには、セキュリティ機能群の役割が不可欠である。

### 7.2.2.3. 拡張容易（機能を上げられる）

拡張容易（機能を上げられる）は、都市 OS を用いてスマートシティサービスを構築すること、そのものによって実現される。第一に、上記のような各種連携機能群によって、サービスや機能を連携させることで、新しい機能を構築して拡張することが容易になる。第二に、スマートシティで共通的に利用されるサービスや機能は、すでに都市 OS の運用支援機能群や共通サービス機能ブロックによって提供されていることから、各サービス提供者やアセット提供者は、それらを開発・提供する必要はないため、新しいサービスの開始や新しいアセットの提供が小さい労力で可能になる。第三に、機能を上げた後にも、不具合や障害を招くことなく稼働できるように、個別サービス管理機能ブロックや、アセットマネジメント機能群がその支援機能を提供している。

## 7.2.3. サービスマネジメント機能群

サービスマネジメント機能群は、都市 OS 上で動作する各種スマートシティサービスが、都市 OS や他のスマートシティサービスと連携するための機能群である。サービスマネジメント機能群は、「**共通サービス機能ブロック**」と「**個別サービス管理機能ブロック**」から構成される。

### 7.2.3.1. 共通サービス機能ブロック

共通サービス機能ブロックは、地域や分野を問わず共通的に使われ、協調領域として都市 OS が提供する複数のサービス機能を提供する。提供される各サービスは、通常ユーザインタフェースを備えており、具体例として、都市 OS ユーザ向けの開発ポータルや、住民と自治体がつながるための双方向コミュニケーションポータル、等がある。

#### 1) 開発ポータルサービス

都市 OS ユーザ向けに、API やデータの検索・仕様の開示が可能なカタログ機能、及び、API を評価可能なコンソール機能等を提供できること。

#### 2) 双方向コミュニケーションポータルサービス

住民や自治体向けに、地域に関連するサービスや情報を集約、配信等を行う機能。住民と自治体、及び、住民とスマートシティサービスをつなぎ、双方向にコミュニケーションが可能な機能を提供し、課題解決や利便性・品質向上に活用されることが望ましい。

#### 3) パーソナライズサービス

住民の志向に沿ったスマートシティサービスを提供するため、住民それぞれが興味を持つ事柄に類する記事の表示優先順位を上げる機能。

#### 4) コンテンツ管理サービス

自治体が提供するポータルサイトやホームページ等に掲載するコンテンツの制作、配信等を行う機能。イベント開催、メール配信等の効果測定を行うためにキャンペーン管理機能を有することが望ましい。

#### 5) 地域ポイント管理サービス

地域課題に対する住民の参加を牽引・維持することを目的とし、地域ごとの独自ポイントサービスを展開・管理するための機能。

#### 6) オプトイン管理サービス

住民が個人の判断で、都市 OS 運用者、及び、サービス提供者に、個人のパーソナルデータの公開範囲を指定するための機能。

#### 7) 可視化・分析ダッシュボードサービス

住民や自治体が地域課題の解決を目的とし、都市 OS 内外のデータと連携し、都市の状況を可視化・分析可能なダッシュボード機能。戦略で設定した KGI/KPI に紐づく分析等、施策に対する効果測定ができることが望ましい。

### 7.2.3.2. 個別サービス管理機能ブロック

個別サービス管理機能ブロックは、都市 OS と連携するスマートシティサービスを管理し、適切に運用することを支援する機能を提供する。個別サービス管理機能ブロックは、以下の複数の機能から構成される。

#### 1) サービスライフサイクル管理機能

都市 OS と連携するスマートシティサービスのライフサイクル（登録、参照、変更、削除）を管理する機能。

#### 2) サブスクリプション管理機能

ユーザが利用できるスマートシティサービスに対して、サブスクリプションの状態（利用の開始終了、利用権限の



設定変更)を管理する機能。

### 3) サービス履歴管理機能

スマートシティサービスの利用の履歴を蓄積し、レポート作成を行う機能。

## 7.2.4. データマネジメント機能群

**データマネジメント機能群**は、都市 OS に都市データを保存・蓄積・管理し、地域内連携支援機能や地域間・分野間連携支援機能と連携して、単一都市・複数都市や、他システムに分散されたデータの仲介を実現する機能群である。データマネジメント機能群は、**データ管理機能ブロック**と**データ処理機能ブロック**から構成される。**データ管理機能ブロック**は、取得したデータを都市 OS に保存・蓄積する複数の機能から構成される。**データ処理機能ブロック**は、取得したデータを都市 OS が扱える形式に処理する機能から構成される。なお、都市 OS は、特性（分類、形式、意味、頻度、量、等）が異なる様々なデータを管理する必要がある。特に、パーソナルデータを取り扱う場合は、改竄不可なデータ蓄積及び授受管理が必要である。

### 7.2.4.1. データ管理機能ブロック

データ管理機能ブロックは、サービス連携や、外部データ連携を通じて収集したデータを、分類や形式を問わず保存・蓄積する機能を提供する。都市 OS は、様々なデータ分類に対して、特性（多様性、頻度、量）が異なる様々なデータを管理する必要がある。特に、パーソナルデータを取り扱う場合は、改竄不可なデータ蓄積及び授受管理を実現する機能を提供する。

また、異なる都市 OS 間でデータを相互運用するためには、データに対しグローバルでユニークな ID を使って管理する必要がある。これにより、都市・地域を超えた様々なデータの中から一つのデータを特定することが可能となる。

#### 1) データストア機能

特性（多様性、頻度、量）が異なる様々なデータに対し、地域が解決する課題に必要なデータを、データの特性に応じて適切に蓄積・活用する機能。データストア機能の API を通じてデータアクセス（登録・参照・変更・削除）を受け付ける。データの分類としては、メタデータや静的データ、動的データ（リアルタイムデータ）、地理空間データ、パーソナルデータなどがある。動的データにおいては、連続したデータを時系列に参照できることが望ましい。

#### 2) イベント処理機能

都市 OS に格納するデータが更新されると、関連するサービス等にデータ更新を通知する機能。

#### 3) データ取得（クローリング）機能

定期的にスマートシティアセットや他システム(Web サイトのオープンデータ)等を巡回し、データを取得する機能。

#### 4) データ ID 管理機能

都市 OS が管理するデータ付与するためのユニーク ID を管理する機能。これによって、地域をまたいだ様々なデータの中から一つのデータが特定可能となる。ユニーク ID は、グローバルで一意となる必要があり、地域ドメイン等を活用することを推奨する。

### 7.2.4.2. データ処理機能ブロック

**データ処理機能ブロック**は、（スマートシティ）サービスや（スマートシティ）アセットから取得したデータの差異を吸収し、標準データモデル等へ変換する機能を提供する。データ処理機能ブロックによって、取得したデータを様々な用途で汎用的に利用することが可能となり、多様なスマートシティサービス間でのデータ利活用が促進される。

また、都市 OS 経由で収集した学習データを用いた AI 開発が広がることを想定した場合、4.3.2 節で述べた通り、AI システムの開発には一定の品質を満たすデータを用いることが求められる。データ処理機能ブロックの持つ

機能は、データの品質を担保するうえで必要な機能である。

#### 1) データ変換機能

取得したデータを都市 OS が扱う標準形式に変換する機能。変換対象は、語彙（意味）やデータ形式、項目、等がある。

#### 2) データ補完機能

リアルタイムデータ等で欠損したデータを補完し、データ品質を向上する機能。データの補完方法は様々な方法があり、目的に応じた補完方法を選択できることが望ましい。

### 7.2.5. アセットマネジメント機能群

アセットマネジメント機能群は、データの収集、及び、接続するスマートシティアセットや他システムの登録・削除等の管理と、スマートシティアセットへの制御を実行する機能群である。アセットマネジメント機能群は、「デバイス管理機能ブロック」と「システム管理機能ブロック」から構成される。

#### 7.2.5.1. デバイス管理機能ブロック

デバイス管理機能ブロックは、都市 OS に接続するデバイス（アセット）の状態を管理・監視し、システム管理者がデバイスの接続異常等を検出可能にするための機能を提供する。本機能ブロックにより、実世界に点在するデバイスをリモートで一元的に管理・監視することが可能となる。また、再起動やファームウェア更新等デバイスに対する制御指示を送信することで、デバイスの制御（アクチュエーション）やメンテナンス等のリモート管理が実施可能となる。

##### 1) デバイスライフサイクル登録機能

デバイス情報（デバイス ID や、固有の MAC アドレス等）のライフサイクル（登録、参照、変更、削除）を管理する機能。

##### 2) デバイス状態管理機能

登録済のデバイスに対して、デバイスの状態（稼働状況や機器情報等）を管理、公開する機能。

##### 3) デバイス制御（アクチュエーション）機能

接続されているデバイスの再起動やデバイスの動作変更等、デバイスの制御を行うためのコマンドを送信する機能。

##### 4) デバイス監視機能

接続されているデバイスの死活状況や、デバイスから送信される障害のイベントを監視する機能。

##### 5) デバイス認証機能

適切な IoT デバイスのみにシステム接続を許可するネットワーク認証の機能。

#### 7.2.5.2. システム管理機能ブロック

システム管理機能ブロックは、都市 OS に接続する他システム（アセット）の認証情報、接続情報、契約情報等、システムに接続するために必要な情報を管理する機能、及び、システム管理者が他システムとのデータ連携状態や接続状態等の状態を管理する機能を提供する。本機能ブロックにより、都市 OS に接続する様々なシステムとのデータ収集・仲介とその状態管理が可能となる。

##### 1) システムライフサイクル登録機能

都市 OS と連携する他システムの連携情報のライフサイクル（登録、参照、変更、削除）を管理する機能。他システムには認証が必要な場合も多く、認証方式やその資格情報についても管理できることが望ましい。

## 2) システム状態管理機能

登録済の他システムに対して、他システムとの接続状態（稼働状況や、機器情報等）を管理、公開する機能。

## 7.2.6. 運用支援機能群

**運用支援機能群**は、都市 OS の維持・発展に必要なシステム管理やプロセス管理を支援する機能群である。システム管理や管理プロセスは、都市 OS を新規導入後も都市 OS 自体を継続的に進化させるために不可欠なものである。これらにより、都市 OS が共通サービスや各種機能を拡充することで、多様なスマートシティサービスを受け入れ、都市 OS の継続的な維持・発展につながるものとなる。運用機能群は、「**システム管理機能ブロック**」と「**プロセス管理機能ブロック**」から構成される。

### 7.2.6.1. システム管理機能ブロック

**システム管理機能ブロック**は、都市 OS が想定外の要因によって停止することなく、各種機能を安定して提供するため、都市 OS を維持管理するための機能を提供する。迅速に障害を検知・復旧するための可用性の確保や、疎結合なシステム構築による拡張容易性の確保等が挙げられる。

#### 1) 拡張支援機能

地域が解決する課題や目指すべき将来像に応じ、将来の機能追加や更新が継続的かつ容易に行える機能。ビルディングブロック方式といった疎結合なシステム構築により、機能の組み換えを柔軟に対応できることが望ましい。

#### 2) ロバスト運用支援機能

スマートシティサービスの可用性を向上させるための機能。都市 OS が障害発生時、都市 OS が可能な限り停止することなく稼働し続ける仕掛けを提供する。都市 OS のサービスレベルを定義し、障害の迅速な検知・復旧、冗長化等により、ユーザへの影響を最小化することが重要となる。

### 7.2.6.2. プロセス管理機能ブロック

プロセス管理機能ブロックは、都市 OS の高度化に向けた取組を継続的に進めるために、必要なプロセスやルール等の規定を支援する機能を提供する。スマートシティサービスや都市 OS の本番稼働に向けたサービス移行管理や、都市 OS の品質を確保するためのシステム運用管理が挙げられる。

#### 1) 都市 OS 企画・開発支援機能

地域の発展等によるサービスの拡大に伴って、都市 OS の各種機能の拡張企画・開発を行う機能。企画に基づき、新規共通サービスや新規機能の導入の計画策定や、要件定義・設計・開発・テスト・移行の工程を管理する。従来のウォーターフォール型の開発だけでなく、共通サービス・各種機能の迅速な立ち上げを実現するため、アジャイル型の開発プロセスを採用することが望ましい。

#### 2) サービス移行支援機能

スマートシティサービスや都市 OS の各種機能を本番稼働する際、スマートシティサービス・各種機能の提供準備と移行計画の策定・管理を行う機能。

#### 3) システム運用管理支援機能

都市 OS におけるシステム運用（変更管理・構成管理・インシデント管理・運用サービス管理・キャパシティ管理等）の管理ツールやプロセスを定義する機能。

また、求められるサービスレベルに応じて、最適なシステム構成を構築するため、IPA/ISEC（独立行政法人情報処理推進機構 セキュリティセンター）が公開する「非機能要求グレード」を活用する。各都市 OS 上のスマート

シティサービスの重要性に応じて、都市 OS の非機能要件を定義し、それを満たすシステム構成とすることが望まれる。

## 7.2.7. セキュリティ機能群

**セキュリティ機能群**は、都市 OS の外部/内部の脅威から都市 OS を防御するために必要な機能を提供する機能群である。なお、スマートシティにおけるセキュリティ機能に関しては、総務省の「スマートシティ・セキュリティガイドライン（第 2.0 版）2021 年 6 月」<sup>64</sup>を参照することが望ましい。

都市 OS のセキュリティ対策は、(1) 技術的対策、(2) 管理的対策（人的対策・組織的対策・物理的（環境的）対策を含む）の二つに大分され、本セキュリティ機能群は上記の対策に応じて、「技術機能ブロック」と「管理機能ブロック」から構成される。

### 7.2.7.1. 技術機能ブロック

技術機能ブロックは、技術的対策に含まれる認証、暗号化、不正アクセス防止、不正アクセス検知・遮断技術等が含まれ、都市 OS として具備すべき機能を提供する。

#### 1) ユーザ管理機能

ユーザ管理は、都市 OS の管理するユーザを一元的に管理する機能である。これにより、都市 OS のユーザは都市 OS 上に実装される様々なアプリケーションを同一のユーザ情報でアクセスすることが可能になる。ユーザ管理には、ユーザを特定する ID に関連づけ、認証情報（パスワードや証明書等）や属性情報（姓名、所属等）の管理と、ID のライフサイクル管理を可能とする。

##### (1) アカウント管理機能

ユーザを特定の ID に関連づけ、認証情報（パスワード）や属性情報（姓名、組織等）の管理と、ID のライフサイクル（登録、参照、変更、削除）を管理する機能。

##### (2) ロール管理機能

ユーザが所属するグループ（ユーザ、管理者等）を定義するロールを管理する機能。

#### 2) ユーザ認証機能

認証機能は、都市 OS のユーザが誰であることを識別する機能。本機能によるユーザの識別と、認可機能による利用権限やアクセス権限の確認により都市 OS やスマートシティの IT システムが提供する各種機能の利用可否を判断できる。都市 OS におけるユーザはユーザやアプリケーションソフトウェア<sup>65</sup>等、様々なケースがあり、それぞれに対する適切な認証方式を提供できる必要がある。

また都市 OS では多様な主体が本認証機能を利用しスマートシティサービスを実装する。本認証機能は外部のスマートシティサービスに対しセキュアかつ利用しやすいインターフェースを提供することが求められる。

##### (1) 認証機能

ユーザ管理に保存された資格情報（ユーザ ID・パスワードや、生体情報等）を用いてユーザの真正性を証明し、アカウントを特定する機能。セキュリティレベルを向上させるため、多要素認証（生体認証、マイナンバーカード等の組み合わせ、等）により、よりセキュアに本人を認証できることが望ましい。

※ 個人の認証については、都市 OS に実装されず、個別のサービスごとに有するケースもあり得る。

<sup>64</sup> [https://www.soumu.go.jp/main\\_content/000757799.pdf](https://www.soumu.go.jp/main_content/000757799.pdf)

<sup>65</sup> M2M: Machine to Machine アプリケーション

## (2) シングルサインオン機能

都市 OS と連携する複数のサービスに対する認証を一元的に管理し、シングルサインオンを実現する機能。ユーザが一度だけ認証することで、都市 OS と連携するスマートシティサービスそれぞれ個別に認証する必要がなくなり、ワンストップサービスの実現につながることが望ましい。

## 3) オブジェクト認証機能

都市 OS に接続するスマートシティサービス、アセット（デバイス）、他都市 OS、他システム、等の正当性を検証する機能。

## 4) 認可機能

認可機能は、都市 OS の管理するデータやサービスに関する利用権限や利用期限を、ユーザやロールごとに設定・判断を可能にする機能。これにより、不用意なデータアクセスやサービス利用を防ぐことが可能となる。アカウントやロール別 34F に、都市 OS にアクセスする範囲や権限を定義する制御ポリシーを管理できること。データアクセスの認可においては、データの公開範囲とアクセス期間の認可を可能にする。またサービス利用の認可においては、サービスの利用範囲と、サービスの有効期限の認可を可能にする。

※ アクセス制御（Access Control）、Usage Control、等と呼ばれる機能を含む。

## 5) 暗号機能

都市 OS が行う通信（都市 OS 内の通信及び都市 OS 外との通信）及び、都市 OS が管理するデータに対して、それぞれの秘匿性に応じ適切なセキュリティ暗号化を行う機能。

## 6) 不正アクセス防止機能（ファイアウォール、等）

都市 OS が行う通信に対して、許可されていない通信（不正な IP アドレスやポート番号を持つパケット等）をブロックする機能。

## 7) 不正アクセス検知／遮断機能（IDS:Intrusion Detection System、等）

不正アクセス防止機能では対応できない、DoS（Denial of Services）攻撃やアプリケーション層の脆弱性を突く攻撃等を検知し、遮断する機能。

### 7.2.7.2. 管理機能ブロック

管理機能ブロックは、管理的対策に含まれる、教育、ルール整備等の人的対策、脆弱性管理、セキュリティ監査等の組織的対策、入退場管理、破壊防止等の物理的対策から構成され、都市 OS の運用・管理に必要な機能を提供する。管理機能ブロックは、都市 OS を構成するソフトウェアが提供する機能ではなく、都市 OS を扱う運用者の活動によって実現される機能から構成される。

#### 1) 脆弱性管理機能

都市 OS を構成するソフトウェアに関しては、その脆弱性に関する情報を収集し、随時パッチ適用等によりその対策を行う機能。また、都市 OS に対して定期的に脆弱性診断を行い、その結果に基づいて対策を実施する機能。

#### 2) 履歴管理機能

都市 OS が行う通信や処理に関する履歴を取得すること。取得した履歴は、証拠保全のために一定期間保存する機能。

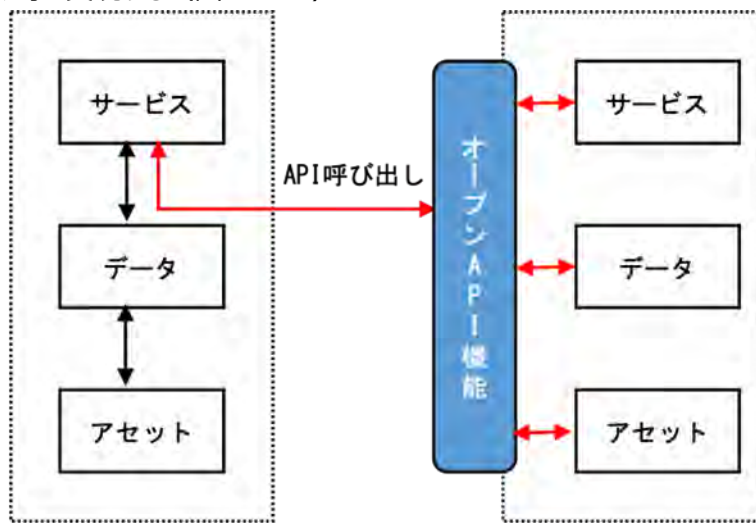


## 7.2.8. 地域内連携支援機能群

地域内連携支援機能群は、同一都市・地域内のスマートシティにおける、サービス連携やデータ連携、アセット／他システム連携を実現するための機能群である。地域内連携支援機能群は、オープン API 機能ブロックとデータ仲介機能（ブローカー）ブロックから構成される。

### 7.2.8.1. オープン API 機能ブロック

オープン API 機能ブロックは、ビルディングブロック型で構築されたスマートシティのシステムにおいて、分散して稼働している都市 OS や（スマートシティ）サービス、（スマートシティ）アセット、等の各ブロックが提供する API を仲介するために、API をオープン化する機能を提供する。都市内や地域内の IT システムに対して、連携に必要な以下に示す機能を付加した API を公開し利用可能にすることで、サービス連携やデータ連携、アセット連携、認証連携、等を実現する（図 7.2-2）。



オープンAPI機能によるサービス連携、データ連携、アセット連携

図 7.2-2 オープン API 機能を用いた、サービス連携とアセット連携の例

#### 1) API 提供機能

API を管理・公開する機能。受付けた API 命令に応じて、関連する他の機能を実行する。

#### 2) 認証機能

API キー等により API を実行した主体の正当性を確認する機能。通常、「セキュリティ機能群」と連携して実現される。

#### 3) ルーティング機能

受付けた API 命令を、他のコンポーネントや機能の API に転送する機能。

#### 4) ライフサイクル管理機能

オープン API 機能ブロックが管理する API を登録・参照・変更・削除する機能。

#### 5) バージョン管理機能

オープン API 機能ブロックが提供する API や、呼び出す他のコンポーネントや関連機能等の複数のバージョンを管理する機能

#### 6) アクセス制限機能

アクセス可能な API やデータを制限する機能。

## 7) レート制限機能

単位時間あたりの API の実行回数を制限する機能。

## 8) アクセス分析機能

API のアクセス数を分析し統計情報化する機能。

## 9) アラート監視機能

受け付けた API 命令を実行する際に発生したアラートを検出する機能。

※ デジタル庁の調査において選定された「API ゲートウェイ」の推奨モジュール等が、当該機能ブロックを構成する機能を実現している。

### 7.2.8.2. データ仲介機能（ブローカー）ブロック

データ仲介機能（ブローカー）ブロックは、都市 OS 内外に散在するデータの所在情報を管理することで、データアクセスを仲介する機能を提供する。異種データの違いを吸収するための共通の API スタイルを提供する他、他システムに分散して存在し、都市 OS が蓄積・保存していないデータに対し、同一インタフェースで透過的にアクセスする機能を提供する（図 7.2-3）。以下の複数の機能から構成される。

#### 1) 異種データ連携機能

都市 OS のデータマネジメント機能ブロックが提供するデータストア機能に蓄積された、様々なデータ（異種データ）に対して、データの種類によらず共通の API スタイルを提供する機能。これによって、複数の異種データを利用する（スマートシティ）サービスの構築を容易化する。

#### 2) データ分散機能

（スマートシティ）サービス等からの、都市 OS 外に分散して管理されているデータに対する登録・参照・更新・削除等のアクセス要求を仲介する機能。

#### 3) イベント処理機能

都市 OS が仲介するデータに対し、事前に定義されたシナリオに従いリアルタイムに処理を実施する機能。これにより、都市 OS 内外に流通するデータの分析・変換・加工処理や、社会状況の変化に伴うアクセス権限の変更等、ダイナミックでかつ柔軟に機能が切り替わる仕掛けが提供できる。

※ 実際に流通しているブローカーのソフトウェアは、データ仲介機能ブロックと、データ管理ブロックに含まれる機能（データストア機能、イベント処理機能）を提供しているものがある（デジタル庁の調査における推奨モジュールである FIWARE Orion、等）が、ここでは機能上の違いから、2つの異なる機能群に分かれて記述されている。また、個人情報を取り扱わないブローカー（非パーソナル）の推奨モジュール（NGSI v2 FIWARE Orion）と、個人情報を取り扱うブローカー（パーソナル）の推奨モジュール（パーソナルデータ連携モジュール）とでは、異なるブローカーモジュールが提供されている。

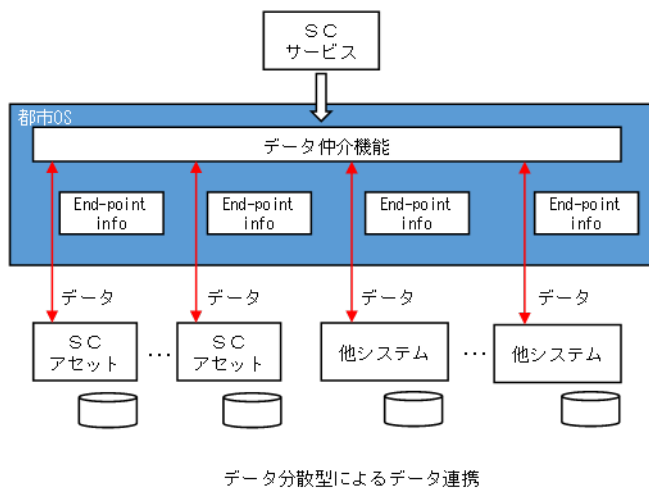
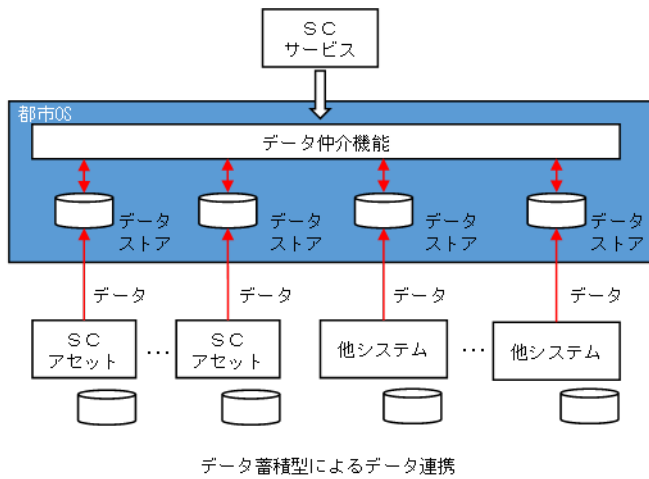


図 7.2-3 データ仲介機能（ブローカー）を用いた分散型データ連携（図 7.1-5 と共通）

## 7.2.9. 地域間・分野間連携機能群

**地域間・分野間連携機能群**は、他の都市・地域のスマートシティや、他の分野のサービスやデータと連携するための機能群である。都市・地域や分野を超えた連携の実現には、基本的には地域内連携と同じように、オープン API 機能やデータ仲介機能（ブローカー）が必要になるが、都市・地域間・分野間連携の場合には、地域内の連携とは異なり、主体に共通のガバナンスが存在しない。したがって、ユーザ管理や認証管理、採用する技術体系（データ形式や通信プロトコル、アセットに用いるデバイス、等）、都市 OS 等が、異なることを想定しなければならない。地域間・分野間連携では、これらの相違を吸収するために、オープン API 機能及びデータ仲介機能（ブローカー）に加えて、以下の機能が必要である。なお、これらの機能は、各都市・地域の都市 OS と、連携基盤が連携して提供することも可能である。どの機能をどちらが提供するかは、いくつかの方式がある（図 7.7 参照）。地域間・分野間連携機能群は、「**主要機能ブロック**」と「**支援機能ブロック**」から構成される。

### 7.2.9.1. 主要機能ブロック

**主要機能ブロック**は地域間・分野間で連携する際の主要なサービスを提供する複数の機能を提供する。

#### 1) データ交換機能（コネクタ機能）

地域間・分野間で、データを授受する機能。API 管理機能や認証機能、認可機能など、他の機能と連携して、地域や分野を超えてトラストを備えたデータ授受を実現する。これらの機能を提供するソフトウェアコンポーネントは「コ

ネクタ」と呼ばれることが一般的である。

**2) API 変換機能**

各都市 OS が提供する API の違いを吸収するために、API を変換する機能。

**3) アカウント管理機能**

地域間・分野間で連携して、データを授受する主体を特定可能なアカウントを管理する機能。

**4) 認証機能**

地域間・分野間で連携して、データを授受する主体の実在性・正当性を確認する機能。

**5) 認可機能**

認証機能によって実在性・正当性が確認された主体に対して、データへの適切なアクセス制御を行う機能。

**6) 証明書登録機能**

データ交換機能を担うソフトウェアモジュール（コネクタ）の真正性を証明するための証明書を登録する機能。

**7) ロケーション管理機能**

データ交換機能を担うソフトウェアモジュール（コネクタ）のネットワーク上の所在を管理するレゾルバ機能、及び交換対象であるデータのネットワーク上の所在を管理するレゾルバ機能。

**8) 来歴管理機能**

データの授受来歴を管理する機能。

**9) 公開機能**

データ交換機能を担うソフトウェアモジュール（コネクタのソフトウェア）のバージョンを管理し、それを公開する機能。

**10) ポータル機能**

地域間・分野間データ連携に関連するサービス群や機能群の一覧とそのエンドポイントを掲載したポータルを一般公開し、データ交換の状況をモニタリングする機能。

**7.2.9.2. 支援機能ブロック**

**支援機能ブロック**は主要機能ブロックが効率的・効果的に実現されるために提供する複数の機能を提供する。

**1) データカタログ横断検索機能**

交換されるデータに関するカタログを、地域間・分野間で横断して検索し、データの発見を支援する機能。

**2) データカタログ作成機能**

交換されるデータに関するカタログの作成ツールを提供し、データの広告・公開を支援する機能。

**3) データ取引市場機能**

契約に基づいてデータの提供と便益の交換に関する契約及びその履行状況を管理する機能。

**4) 語彙管理機能**

交換されるデータの含まれる語彙を登録、保管、及び公開し、分野を超えたデータ活用を支援する機能。

※ 地域間・分野間連携支援機能ブロックを実現した基盤システムに関しては、現在開発が進められているプロジェクト「DATA-EX 分野間データ連携基盤技術」（分野を超えてデータの発見と利用を可能とするプラットフォーム）がある。<sup>a</sup>

※ コネクタと同様の機能を、オープン API 機能ブロックとデータ仲介機能（ブローカー）を組み合わせることもできる。

<sup>a</sup>DATA-EX : <https://data-society-alliance.org/data-ex/>

## 7.2.10. 都市 OS 導入のためのモジュール選定と構成例

都市 OS の導入・構築では、様々なソフトウェアモジュールを選定して組み合わせ方を設計し、システム全体を構成する。候補となるソフトウェアモジュールは商用製品や OSS のものなど様々である。ビルディングブロック方式でシステムを構成するにあたり、設計者は部品となるモジュールに関する基礎的な知識と、それらの組み合わせ方を知る必要がある。

社団法人 データ社会推進協議会（以下、DSA）は、エリア・データ連携基盤の機能要件のコアとなる部品を推奨モジュールとして定め、利用方法や構成例の解説など普及促進を行っている（図 7.2-4）。具体的には、API ゲートウェイ機能を持つ Kong Gateway、非パーソナルデータのブローカー機能を持つ FIWARE Orion、パーソナルデータのブローカー機能を持つパーソナルデータ連携モジュールを推奨している（図 7.2-5）<sup>66</sup>。

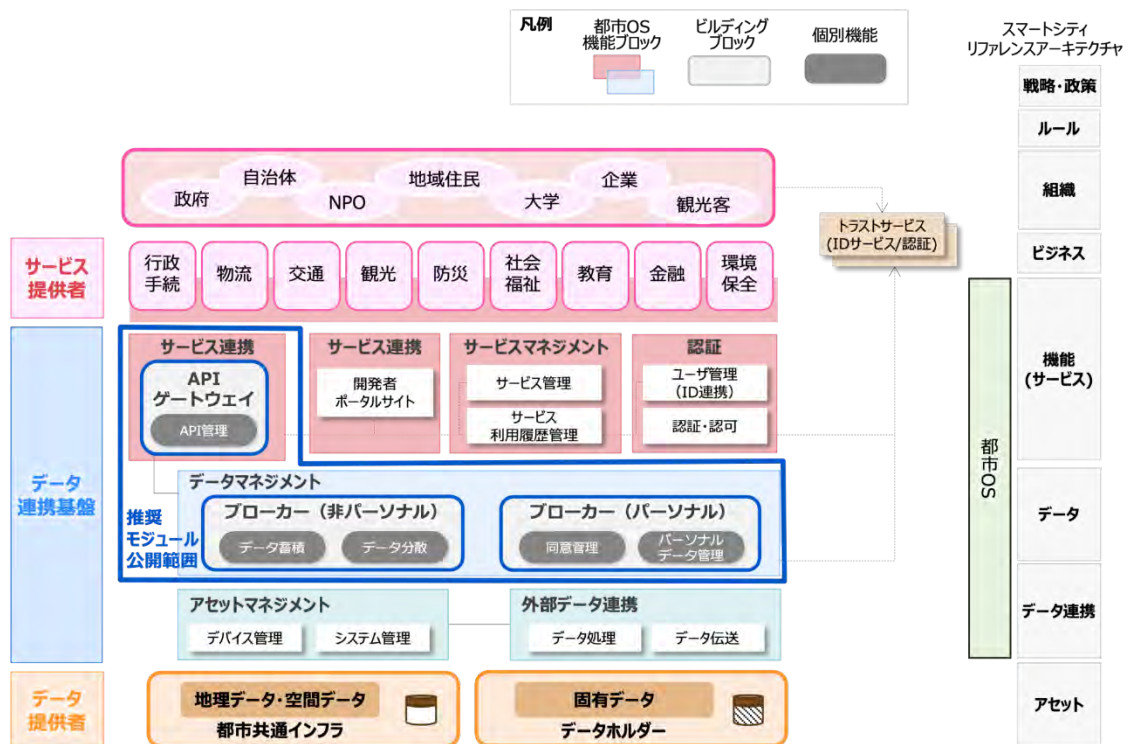


図 7.2-4 エリア・データ連携基盤の全体像（生活用データ連携調査研究の報告書より一部修正して引用）

表4 推奨モジュールのソフトウェア概要

推奨モジュール	OSS名	ライセンサー	ライセンス	URL
APIゲートウェイ	Kong Gateway	Kong Inc.	Apache License 2.0	<a href="https://github.com/Kong/kong">https://github.com/Kong/kong</a>
ブローカー (非パーソナル)	NGSI v2 FIWARE Orion	Telefonica	AGPL v3.0	<a href="https://github.com/telefonicaid/fiware-orion">https://github.com/telefonicaid/fiware-orion</a>
ブローカー (パーソナル)	パーソナル データ連携 モジュール	日本電気 株式会社	MIT License	<概要> <a href="https://data-society-alliance.org/data-ex/area-data/module/manual/">https://data-society-alliance.org/data-ex/area-data/module/manual/</a> <OSSコミュニティ> GitHub : <a href="https://github.com/Personal-Data-Linkage-Module">https://github.com/Personal-Data-Linkage-Module</a> Slack : <a href="https://www.code4japan.org/activity/community">https://www.code4japan.org/activity/community</a> (チャンネル名: #proj-cityos_auth)

図 7.2-5 推奨モジュールのソフトウェア概要

<sup>66</sup> 推奨モジュールの概要 | 一般社団法人データ社会推進協議会(DSA)



先に述べたように、システムの構築には推奨モジュールだけでなく周辺のモジュールも含めた組み合わせ方を知る必要がある。DSA は、推奨モジュール使用した基本的な構成例について、を教育資料として公開している（図 7.2-6）。

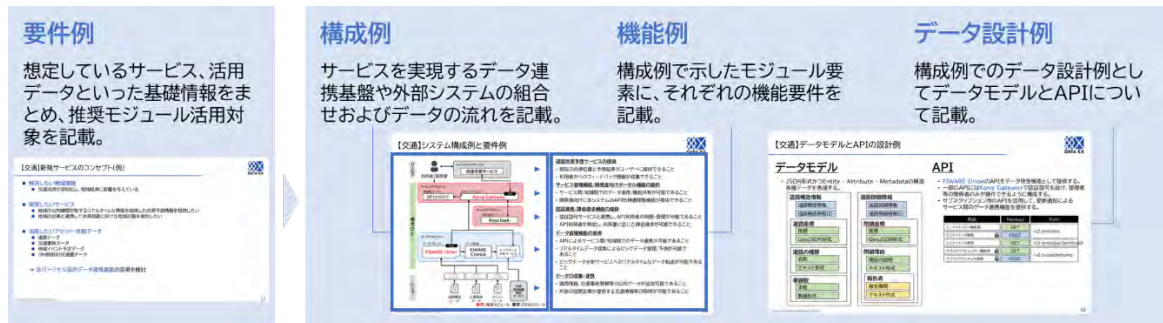


図 7.2-6 教育資料の流れ

この資料では、地域の課題解決にあったサービスとその要件を仮定し、推奨モジュールと周辺モジュールを汲み合わせ、分野別のサービスを構成した場合の例を記載している（図 7.2-7）。導入するサービスの要件により、必要となる機能や連携するアセットの対象も異なるため、適合する機能を持つモジュールについて複数のサービス分野を例に構成例を示している。本書で示した各機能群の機能を持つ具体的なモジュールの名称まで明らかにすることで、具体的なモジュールの組みあわせを検討する際の補助資料として活用することができる。

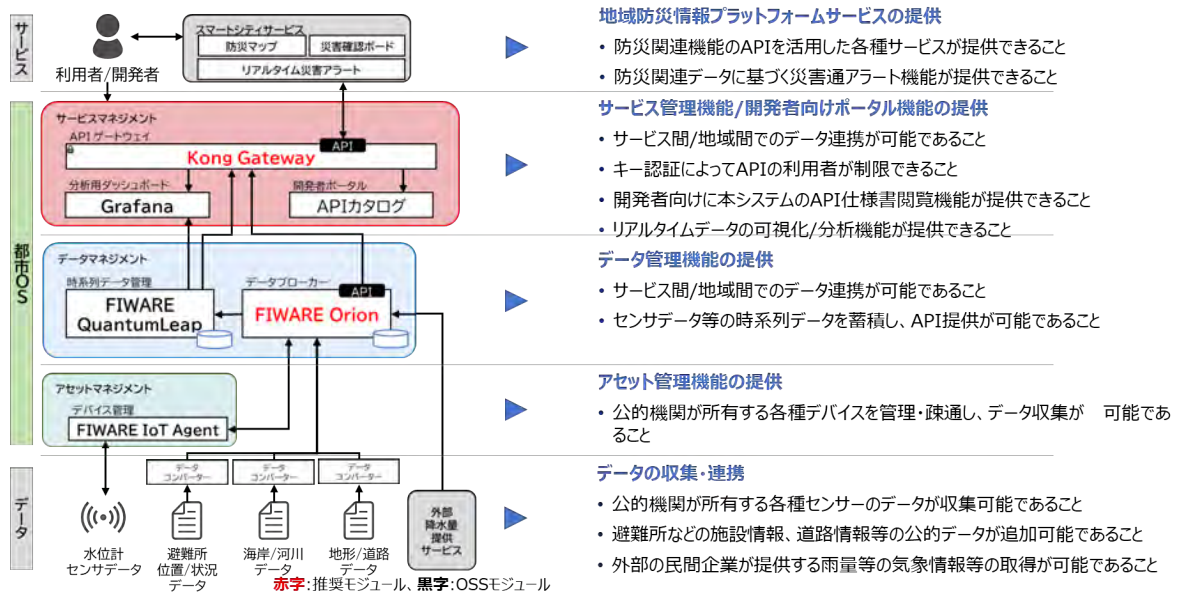
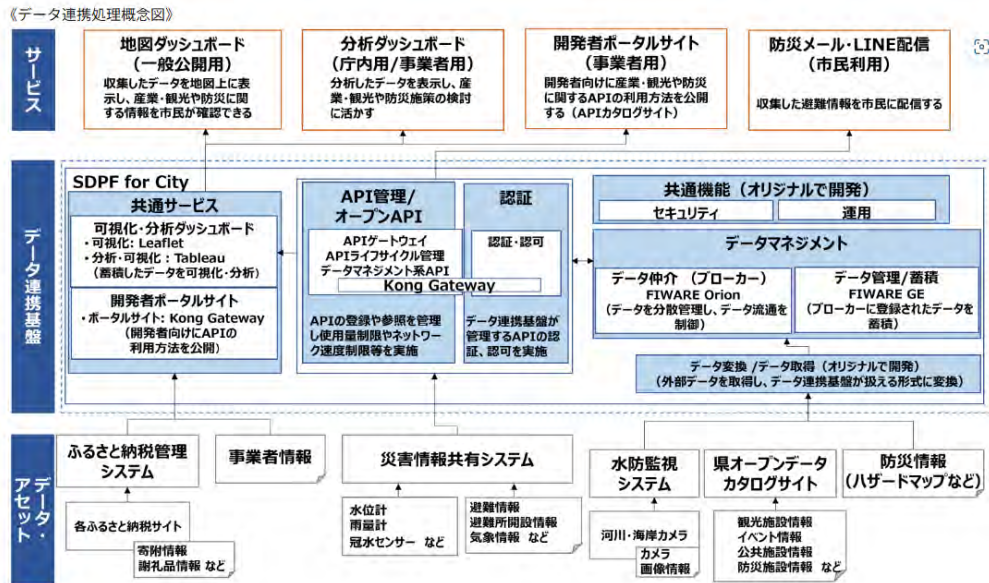


図 7.2-7 推奨モジュールによる構成例

このほかにも、DSA は実際のシステムの構成を図式化したデータ連携処理概念図をホームページに掲載している（図 7.2-8）<sup>67</sup>。これらのコンテンツを参考としていただきたい。

<sup>67</sup> 引用 [データ連携基盤における推奨モジュールの普及状況と導入事例 | 一般社団法人データ社会推進協議会 \(DSA\)](#)



《サービスとして利用している分野》

防災、観光・地域活性化

《事業内容》

静岡県焼津市は水産加工業の売り上げ拡大と防災・減災意識の向上を図っています。

《推奨モジュールの利用事例》

推奨モジュールのブローカー(非パーソナル)を使用して、ふるさと納税管理システムに関する情報、事業者情報、災害情報共有システムのデータ、水防監視システムに関する情報、県オープンデータカタログサイトで公開されている情報、防災情報(ハザードマップなど)を下記のサービスにデータ連携しています。

- ・ 収集した産業、観光、防災に関するデータを地図に表示するダッシュボード
- ・ 分析した産業、観光、防災に関するデータを表示するダッシュボード
- ・ 開発者向けの産業、観光、防災に関するAPIカタログサイト
- ・ 収集した避難情報を市民に配信する防災メール・LINE配信サービス

図 7.2-8 データ連携処理概念図

当該資料中において、非パーソナルデータのシステムについては、推奨モジュールである FIWARE Orion と組み合わせやすい FIWARE GE (Generic Enablers) を活用する構成を例示している。FIWARE GE は中核となる Orion を中心とし、コンテキスト処理や分析・可視化を担うもの、API 認可やコンテキスト情報の可視化などを担うもの、IoT デバイスなどとのインタフェースを担うものなど様々なものが存在する。これらについては FIWARE コミュニティによる情報発信サイト<sup>68</sup>に概要やチュートリアルが公開されている(一例: 図 7.2-9、図 7.2-10)<sup>69</sup>。これらのモジュールの選定の際には、OSS 使用におけるリスクを認識する必要がある。構成例として掲載されているものであっても OSS モジュールとしての更新頻度が少ない場合は、使用実績に乏しい、公開されている情報が陳腐化している、といった可能性もある。最新の動向を調査したうえで、リスクの少ない OSS 選定をすることが重要である。

68 [FIWARE Catalogue | Let's FIWARE](#)

69 [FIWARE 概要 と GEs の紹介 | Let's FIWARE](#)



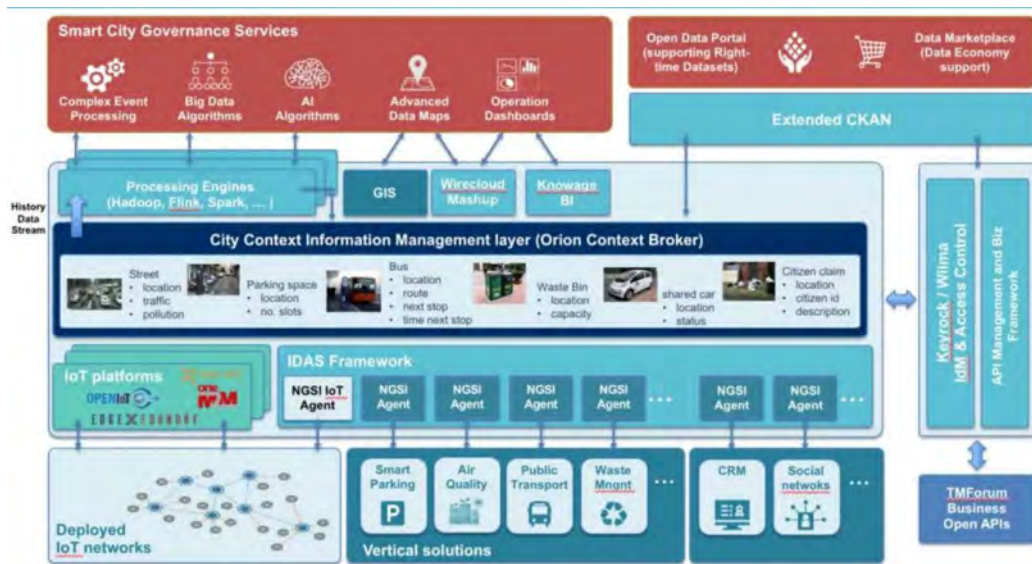


図 7.2-9 FIWARE GE の概要

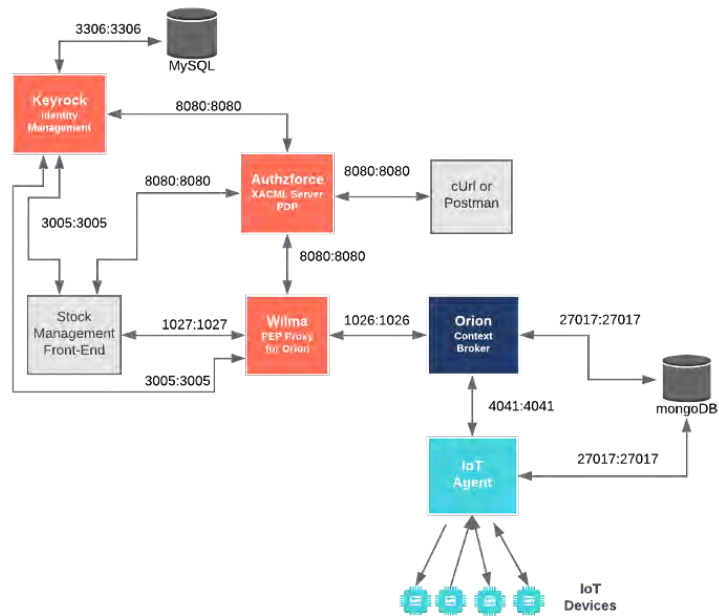


図 7.2-10 Wilma によるアクセス保護と Keyrock による ID 管理を用いた構成例<sup>70</sup>

<sup>70</sup> [ルールセット・ベースの権限 - FIWARE Step-by-Step for NGSI-v2](#)

## 7.3. 地域間・分野間連携方法の事例

### 7.3.1. オープン API に用いられる個々の標準規格やシステムの事例

#### 7.3.1.1. 認証系 API

OAuth2.0 と OpenID Connect を併用（OIDC の仕様に基づき OAuth の認可要求を行う）することが望ましい。外部連携で利用するための機能要件を以下定義する。

##### 1) 認証・認可

ID 管理に保存された資格情報（ID・パスワードや、生体情報等）を用いて検証、及び、アクセストークンの払い出しや失効を行えること。事前に設定されたユーザの権限に応じ、利用範囲が制限される。

※ OAuth を活用することを推奨する。

##### 2) 個人認証

パーソナルデータを活用する場合といった高いセキュリティが求められる認証に対しては、生体認証やマイナンバーカードを併用した多要素認証等、個人を特定するための認証方法を提供すること。

※ 個人の認証については、都市 OS に実装されず、個別のサービスごとに有するケースもあり得る。

##### 3) 属性取得

認証されたユーザの属性情報を取得できること。

※ OpenID Connect を活用することを推奨する。

認証に関わる標準規格について説明する。

##### ・ OAuth (Open Authorization)

サードパーティーアプリケーションによる HTTP サービスへの限定的なアクセスを可能にする認可フレームワーク。最新の標準は、2012 年に RFC として発行された OAuth 2.0 である (RFC6749、RFC6750)。

##### ・ Open ID Connect

OAuth 2.0 プロトコルの上にシンプルなアイデンティティレイヤーを付与したもの。

##### ・ SAML (Security Assertion Markup Language)

異なるインターネットドメイン間でユーザ認証、シングルサインオンを行うための XML をベースにした標準規格である。2002 年に策定され、2005 年にはバージョン 2.0 となっている。

#### 7.3.1.2. サービス連携系 API

サービス連携においては、ユーザの利便性が向上する共通的な機能をユーザインタフェースや API として提供する。標準規格として例示できるものは少なく、将来的に拡充が必要となる。

##### 1) サービス連携（決済等）

都市 OS 上のサービスが保持する API を、都市 OS 上の API として公開できること。

##### 2) 地域ポイント管理

ユーザに紐づく地域ポイントの加算・減算・照会処理等を実行できること。

##### 3) オプトイン管理

都市 OS のユーザが、自身のユーザ情報をどのサービスに対して提供するか、そのオプトイン／オプトアウトを管理できること。提供する情報の種別まで管理できることが望ましい。パーソナル情報授受と連動し、オプトイン／オプトアウトの履歴を管理できることが望ましい。

#### 4) カタログ管理

開発ポータルサイト内のカタログ機能に保管されたメタデータ（データカタログ）の登録・取得・検索処理を実行できること<sup>71</sup>。

##### 7.3.1.3. データ管理系 API

都市 OS は特性が異なる様々なデータを管理する。都市 OS の外部連携では、管理するデータの特性を最大限に活かしたデータマネジメントやデータアクセスが重要である。

###### 1) データアクセス API

都市 OS のデータマネジメントと連携し、データのライフサイクル（登録、参照、変更、削除）を管理するための API を提供できること。

###### 2) パブリッシュ/サブスクライブ（pub/sub）API

都市 OS が保管するデータに変更が生じた際に、リアルタイムに変更内容を通知先に送信するための API を提供できること。また、通知内容（条件や通知先等）のライフサイクル（登録、参照、変更、削除）を管理するための API についても提供できること。

###### 3) データ仲介 API

分散するデータに対し、その所在のライフサイクル（登録、参照、変更、削除）を管理するための API を提供できること。

###### 4) パーソナルデータ（要配慮個人情報）授受 API

パーソナルデータ（要配慮個人情報）をスマートシティサービスや他都市 OS に共有する場合に本機能を提供すること。パーソナルデータの提供には必ず事前に本人確認を行う。本人確認方法には、デバイス認証・生体認証・マイナンバーカード認証等での多要素認証を行う。また、データ提供期間及びデータ提供先を限定する機能を提供する。データ提供時にはその履歴を必ず保存する。

データ管理に関する API の規格などについて説明する。

###### ・ NGSI

NGSI は、“Next Generation Service Interfaces” の略。Open Mobile Alliance により標準化されたオープンな国際標準規格の API である。NGSI では物理オブジェクトをエンティティ、属性、メタデータの 3 つの形式で表現する。

FIWARE では、OMA が策定した NGSI 仕様から派生した FIWARE NGSI を使用している。2024 年現在、FIWARE NGSI のバージョンは NGSI v1、NGSI v2、NGSI-LD の 3 バージョンがあり、NGSI v2 が普及している。最新版は NGSI-LD（Linked Data）であり、ETSI（欧州電気通信標準化機構）により公開されている。NGSI-LD では、データのグラフ構造が表現可能となり、エンティティとリレーションシップとして表される。NGSI v2 と NGSI LD の比較については、表 7.3-3 に示す。

###### ・ JSON

JSON は、“JavaScript Object Notation” の略。軽量なテキストベースで、言語に依存しないデータ記述言語の一つ。人間にとって読み書きが容易で、マシンにとっても簡単にパースや生成を行える形式で

<sup>71</sup> 参考：総務省発行データ流通プラットフォーム間の連携を実現するための基本的事項

[https://www.soumu.go.jp/main\\_content/000483319.pdf](https://www.soumu.go.jp/main_content/000483319.pdf)

ある。

・ SPARQL

SPARQL は“SPARQL Protocol and RDF Query Language” の略。W3C が標準化した RDF 問い合わせ言語の一つ。クエリの基本的なパターンである論理積や論理和を始め、文字列操作やフィルタ一等のその他のパターンを指定可能。

・ HTTP/HTTPS

HTTP は、“Hypertext Transfer Protocol”（ハイパーテキスト・トランスファー・プロトコル）の略。Web ブラウザが Web サーバと通信する際に主として使用する通信プロトコルである。HTTPS は、“Hypertext Transfer Protocol Secure” の略。SSL/TLS プロトコルによって提供されるセキュアな接続の上で HTTP 通信を行うことである。

・ REST/RESTful

REST は、“Representational state transfer” の略。ウェブのような分散ハイパーメディアシステムのためのソフトウェアアーキテクチャのスタイルの一つである。主として以下の四つの設計原則の項目から構成される。

- 1 ステートレスなクライアント/サーバプロトコル
- 2 全ての情報（リソース）に適用できる「よく定義された操作」のセット
- 3 リソースを一意に識別する「汎用的な構文」
- 4 アプリケーションの情報と状態遷移の両方を扱うことができる「ハイパーメディアの使用」

この「REST の原則」に従って実装されている Web システムの HTTP での呼び出しインタフェースのことをしばしば「RESTful」と呼ぶ。

#### 7.3.1.4. アセット連携系／他システム連携系 API

都市 OS が連携するスマートシティアセットや他システムには、異なるデータフォーマットやインタフェース、通信方式や通信プロトコルに合わせた連携が必要である。

##### 1) 片方向通信

汎用的な片方向通信プロトコル（HTTP/HTTPS）による、データアクセスを可能とすること。

※ データアクセスについては、データアクセス API（データ管理系 API の中）を参照いただきたい。

##### 2) 双方向通信

汎用的な双方向通信プロトコル（MQTT、WebSocket 等）による、スマートシティアセットのデータアクセスや、スマートシティアセットへのアクチュエーションを可能とすること。

##### 3) ネットワーク

インタフェーススマートシティアセットと連携するためのネットワークは、解決する課題や、接続する機器の仕様により特性（通信距離、通信速度、消費電力等）が異なる。G/5G 等の広域ネットワーク（WAN）だけでなく、LPWA 等の IoT/M2M の通信に利用されている省電力かつ広域利用可能なネットワーク（LPWAN）も活用すべきである。

#### 7.3.1.5. ブローカー

住民等に提供される各種サービスは、サービス提供者、データ提供者、データ連携基盤の連携によって実現される。持続的なサービス提供及び発展に向けては、サービス、データ、データ連携基盤の間の相互運用性の確保が重要であり、それを実現する上でコアとなる部品が「API ゲートウェイ」「ブローカー」であり、デジタル庁が推奨モジュールとして公開している。

ブローカーは大きく、非パーソナルとパーソナルに分かれる。

### 1) ブローカー（非パーソナル）

ブローカー（非パーソナル）は、個人に紐づかないデータを蓄積または分散管理し、サービス間の非パーソナルデータの流通を制御するものである。

ブローカー（非パーソナル）の推奨モジュール要件<sup>72</sup>は表 7.3-1 の通りである。

**表 7.3-1 ブローカー（非パーソナル）の推奨モジュール要件**

データ連携目的	分類	機能	説明	必須	推奨
データ利活用	データ参照	①データ分散	データ参照の要求を受け付け、外部サービスが保持するデータを返却可能なこと	●	
			データ利用者に対してデータの所在を隠ぺいすることができること		●
		②データ蓄積	データ参照の要求を受け付け、データストア機能に蓄積されたデータを返却可能なこと		●
	サービス呼び出し	③イベント処理	サービス呼び出しの要求を受け付け、外部サービスの処理を実行し結果を返却可能なこと		●
	API仕様	④API仕様	データ利活用の利便性を考慮し、標準ルールに沿ったAPI（REST等）を提供可能なこと	●	
データ収集	データ更新	⑤イベント処理	データ提供者からデータを受け付け、必要なサービスへデータを送信できること		●
		⑥データ蓄積	データ提供者からデータを受け付け、データストア機能に蓄積可能なこと		●

デジタル庁の調査において選定されたブローカー（非パーソナル）の推奨モジュールは「NGSI v2 FIWARE Orion（ライセンス：Telefonica）」である。このNGSI v2 FIWARE Orionの代表的な機能<sup>73</sup>は表 7.3-2の通りであり、上記、ブローカー（非パーソナル）の推奨モジュール要件を満たしている。

**表 7.3-2 NGSI v2 FIWARE Orionの代表的な機能**

機能	概要
データの登録	データを蓄積する機能。
データの参照	データを参照する機能。RESTAPIを用いて、データの所在を意識することなくデータを参照することができる。
データの更新	蓄積したデータを更新する機能。
非同期通知	データが更新された場合にあらかじめ登録された通知先へデータ更新通知を送信する機能。
リクエスト転送	リクエストを外部のデータ提供元へ転送する機能。Orionは、事前に登録されたデータ所在情報をもとに、データ提供者にデータを問い合わせる。

<sup>72</sup> 引用：[推奨モジュールの概要 | 一般社団法人データ社会推進協議会\(DSA\)](#)

<sup>73</sup> 引用：FIWARE Orion 利用手順書（1.1.0版）：（一般社団法人データ社会推進協議会）

表 7.3-3 NGSI v2 と NGSI LD の比較

項目	NGSI v2	NGSI LD
データモデル	OMA Context Information Model に基づき コンテキストエンティティ (Context Entity) , コンテキスト属性 (Context Attribute) , コンテキストメタデータ (Context Metadata) を主要な三要素として表現する。JSON で表現。	JSON-LD (Linked Data をシリアライズするための JSON ベースのフォーマット) を用いて表現する。RDF を基底クラスとしている。
データの関係性	エンティティや属性間の関係 (Relationship) を明示的に表現する仕様なし。	エンティティや属性間の関係 (Relationship) を明示的に表現する仕様あり。
リンクのサポート	なし	リンクドデータとしてエンティティ間の関係を明示的に定義可能
エンティティ ID の形式	単純な文字列 (通常は識別子) (例: "id": "device1")	URI (統一リソース識別子) 形式で表現 (例: "id": "urn:ngsi-ld:Device:00123")
エンティティの属性	属性はエンティティのプロパティとなる情報を、名前と値の対にして定義、リスト化する。属性に型やメタデータの付与も可能。 (例: temperature)	属性はエンティティのプロパティとなる情報 (Property) もしくは関係性 (Relationship) の二種類に大別され、各々定義、リスト化する。(例: temperature が Property タイプ)
属性の型	属性に定義される情報はプロパティのみであるため、型はそのデータの種別を表し、事前定義された既定のもの (String, Number, Boolean など) を使用する。	属性にはプロパティと関係性の複数情報を定義できるため、型には Property もしくは Relationship のいずれかを明示する。プロパティの場合は更にデータの種別を表す型の定義が可能で、関係性の場合はエンティティの関係性を示す型を URI で指定可能。
クエリ言語	Simple クエリ言語, Geographical クエリ	NGSI-LD クエリ言語, NGSI-LD Geo クエリ言語, NGSI-LD Temporal クエリ言語
分野間データ連携	データ構造がシンプルで使いやすい一方、分野間や都市間で連携するデータ表現が複雑になりがちで大規模構成に不向き。	複雑なデータリンクや意味論的關係を扱う場合に適しており、分野間や都市間で連携するデータ表現に向いているが、設計がやや複雑であり比較的難易度が高い。
準拠している規格	<ul style="list-style-type: none"> <li>FIWARE 独自規格</li> <li>JSON の拡張サブセット</li> </ul>	<ul style="list-style-type: none"> <li>W3C の RDF を継承したサブクラス</li> <li>JSON-LD の拡張サブセット</li> </ul>
代表的な OSS プロカー	FIWARE Orion	Orion-LD, Scorpio, Stellio

## 2) ブローカー (パーソナル)

ブローカー (パーソナル) は、個人に紐づくデータを蓄積または分散管理し、データの提供元組織の許諾及び本

人同意に基づき、サービス間のパーソナルデータの流通を制御するものである。

ブローカー（パーソナル）の推奨モジュール要件<sup>74</sup>は表 7.3-4 の通りである。

**表 7.3-4 ブローカー（パーソナル）の推奨モジュール要件**

ステークホルダー	分類	機能	説明	必須	推奨
個人	同意管理	①蓄積同意	個人が「サービス提携組織（アプリ）による蓄積」に対する自身の同意状態を、「データセット種別」の認可粒度で管理できること	●	
		②共有同意	個人が「サービス提携組織（アプリ）による共有」に対する自身の同意状態を、「データセット種別」の認可粒度で管理できること	●	
			個人が「サービス提携組織（アプリ）による共有」に対する自身の同意状態を、「データ実体」の認可粒度で管理できること		●
	パーソナルデータ管理	③パーソナルデータ所在管理	個人が蓄積された自身のパーソナルデータの所在を把握できること	●	
			個人が蓄積された自身のパーソナルデータの内容をデータ実体単位で確認できること		●
		④アクセスログ	個人が蓄積された自身のパーソナルデータの共有状況を把握できること	●	
サービス提供組織	ブローカー（非パーソナル）同等機能	⑤API仕様	表 2 に準ずる	●	
		⑥データ分散	表 2 に準ずる	●	
		⑦データ蓄積	表 2 に準ずる	●	
	先端的サービス管理	⑧先端的サービス定義	サービス提供組織が先端的サービスを定義できること	●	
サービス提携組織	サービスアセット管理	⑨サービスアセット定義	サービス提携組織がサービスアセットを定義できること	●	

デジタル庁の調査において選定されたブローカー（パーソナル）の推奨モジュールは「パーソナルデータ連携モジュール（ライセンサー：日本電気）」である。

### 7.3.1.6. コネクタ

コネクタとは、都市や地域、分野を超えて、同一のガバナンス体系がない主体間において、データ仲介を実現する機能を提供する。したがって、コネクタ機能では、ユーザ管理や認証管理、採用する技術体系（データ形式や通信プロトコル、アセットに用いるデバイス、等）、異なる都市 OS・データ連携基盤等間のデータを授受するために必

<sup>74</sup> 引用：[推奨モジュールの概要 | 一般社団法人データ社会推進協議会\(DSA\)](#)



要な機能を提供する。

異なる都市 OS・データ連携基盤等間の相違を吸収するために、認証連携機能や API 変換機能、データ変換機能、語彙管理機能、ロケーション管理機能、データ収集機能、公開機能、等の機能と連携する必要がある。例として、IDSA<sup>75</sup>や CADDE、DATA-EX のコネクタ（Connector）、等のソフトウェアモジュールがコネクタ機能を実装している。

## 7.3.2. 海外における連携プラットフォームの事例

プラットフォームを連携する方法として、システムを共通化する方法と、システムは共通化せずに相互運用性（Interoperability）のためのルールを策定し、各システムがそのルールに準拠することで連携する方法がある。地域や分野を超えてシステムを共通化することは困難であることから、相互運用性を確保する方法による連携が一般的である。

本節では、海外の相互運用において、欧州での行政機関の相互運用のための概念（EIF）や、全体アーキテクチャにおける技術的にクリティカルな部分のみを共通化するという考え方（MIMs、PPI）を参考に示す。

### 7.3.2.1. European Interoperability Framework (EIF)

欧州では行政機関の相互運用のための概念を European Interoperability Framework (EIF) にて提言している。行政機関のために作られたフレームワークであったが、近年（2022 年 1 月）スマートシティ・プロジェクトとの融合が図られている（図 7.3-1）。

特に都市 OS として重要なのは、「Semantic Interoperability」としてデータの流通に関連するデータモデル（データ、構造、項目等）や、「Technical Interoperability」として技術的に接続するための通信プロトコルである。

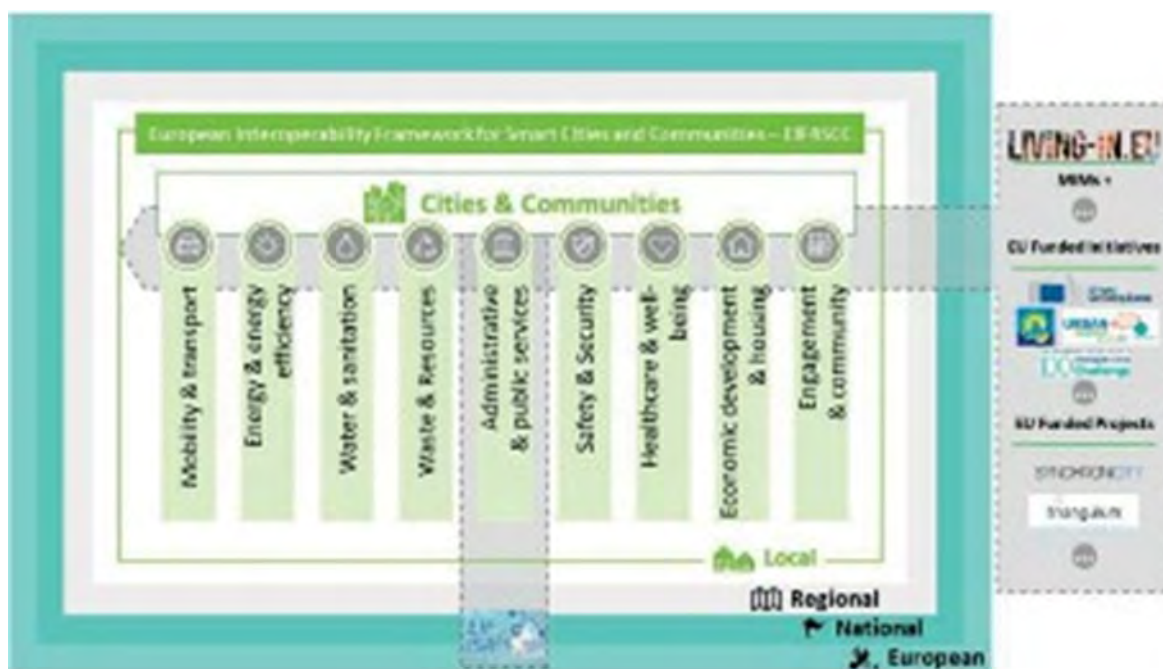


図 7.3-1 EIF for Smart City & community

<sup>75</sup> International Data Spaces Association (IDSA) <https://internationaldataspaces.org>

### 7.3.2.2. Minimal Interoperability Mechanisms (MIMs)

欧州を中心とするスマートシティの国際団体である Open & Agile Smart Cities (OASC) は最小限相互運用メカニズム (Minimal Interoperability Mechanisms, MIMs) の考え方を提唱している。Open & Agile Smart City(OASC) が主催する年次会合 Connected Smart City Conference 2020 (1/23、@ Paris) では、表 7.3-5 の 3レベルに加え、Helsinki 提案の Personal Data Management, Amsterdam 提案の Fair AI が MIMs に加わることになった。

特に、この最小限相互運用メカニズムの考え方に基づいて策定されている SynchroniCity は、相互運用ポイントと呼ばれる主要インタフェースに関する要件をアーキテクチャに定めている。相互運用ポイントは、各都市のアプリケーションやシステムと SynchroniCity フレームワークとを接続するメイン・インタフェースであり、分野・地域間におけるデータの提供と利用に関する API、及び共通データモデルに関する仕様とガイドラインから構成されている (表 7.3-6、図 7.3-2)。

**表 7.3-5 最小限相互運用メカニズム (Minimal Interoperability Mechanisms, MIMs)**

MIM	Point	Description	References	Related Standards & Baselines
OASC Context Information Management MIM	Context Info Management API	This API allow to access to real-time context information from the different cities.	Reference Architecture for IoT-Enabled Smart Cities [SC-D2.10]	ETSI NGSI-LD prelim API, OMA NGSI, ITU-T SG20'/FG-DPM'
OASC Data Models MIM	Shared Data Models	Guidelines and catalogue of common data models in different verticals to enable interoperability for applications and systems among different cities.	Guidelines for the definition of OASC Shared Data Models [SC-D2.2], Catalogue of SASC Shared Data Models for Smart City domains [SC-D2.3]	[FIWARE, GSMA, Schema.org, SAREF, Synchronicity RZ+ partner data models]
OASC Ecosystem Transactions Management MIM	Marketplace API	It exposes functionalities such as catalogue management, ordering management, revenue management, Service Level Agreements (SLA), License management etc. Complemented by marketplaces for hardware and services.	Basic Data Marketplace Enablers (SC-D2.4)	TM Forum API

表 7.3-6 SynchroniCity における API

項番	API	説明
1	IoT Management	様々なIoTデバイスが持つ多様な標準やプロトコルの違いを吸収し、互換性を確保した上で利用できるようにする。
2	Context Data Management	アーキテクチャのコアであり、様々なIoTデバイスやデータソースからのコンテキスト情報を管理し、統一的なアプローチとインタフェースを提供する。
3	Data Storage Management	様々なデータストレージに格納された多種多様なデータへの統一的なアクセスとデータ管理、及びデータ品質の保証に関する機能（データクレンジングやデータ品質をチェックするツール等）を提供する。
4	IoT Data Marketplace	様々なデータを取引するためのシステム（データ取引市場）を実装し、アセットカタログ、発注、収益・顧客・SLA・ライセンス管理等の機能を提供する。
5	Security, Privacy and Governance	アーキテクチャ内のデータ、及びプラットフォームサービスに関する全てのセキュリティをカバーし、ID 管理の他、機密性、認証、認可、完全性、否認防止、アクセス制御等の重要なセキュリティ機能を提供する。

Vertical	Data Model	Description	Original Source	Approval Status
Environment	<a href="#">AirQualityObserved</a>	It represents an observation of air quality conditions at a certain place and time	GSMA	Approved
Environment	<a href="#">NoiseLevelObserved</a>	It represents an observation of those parameters that estimate noise pressure levels at a certain place and time	FIWARE	Approved
PointOfInterest	<a href="#">PointOfInterest</a>	A harmonised geographic description of a Point of Interest	GSMA updated by SynchroniCity	Approved

対象データ項目や表記などを詳細に規定

※一部を記載

The image shows a screenshot of the 'Air Quality Observed' data model specification page. It includes a description of the model, a list of data model properties (like @id, type, @context, @timestamp, @created), and three examples: a 'Normalized Example' in JSON, a 'key-value pairs Example' in JSON, and an 'LD Example' in JSON-LD format.

図 7.3-2 SynchroniCity におけるデータモデルの標準化例

### 7.3.2.3. Pivotal Points of Interoperability (PPI)

米国の国立標準技術研究所（National Institute of Standards and Technology、NIST）は、効果的かつ強力なスマートシティソリューションを実現する上で、種々の IoT プラットフォーム関係者への呼びかけを行い、アーキテクチャの比較検討に関する国際的な共同作業（IES-City Framework）を行い、アーキテクチャ上の特徴に関する合意形成のフレームワークを図 7.3-3 に示すように策定している。

このような比較検討を通して、Pivotal Points of Interoperability (PPI) と、その集合体である Zones of

Concern (ZoC) という考え方が形成されている。それらは、IoT システム間の相互運用性確保のためには必ずしもシステム全体の仕様を合わせる必要はなく、クリティカルな領域において適切と思われる共通技術を採用すればよいという考え方である。PPI の例として図 7.3-4 に示すように、データ源からデータを上げる連携部分 (Southbound Interface) における IPv6 アドレス、データ管理・連携層から上のアプリにデータを提供する連携部分 (Northbound Interface) における RESTAPI 等が挙げられている。

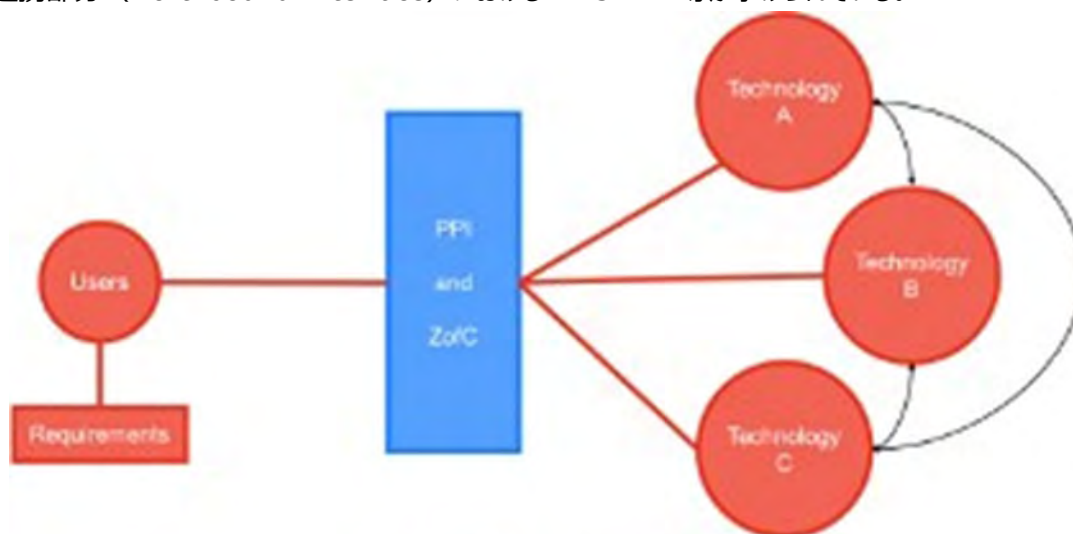


Figure 1: IES-City Framework Structure

図 7.3-3 IES-City Framework

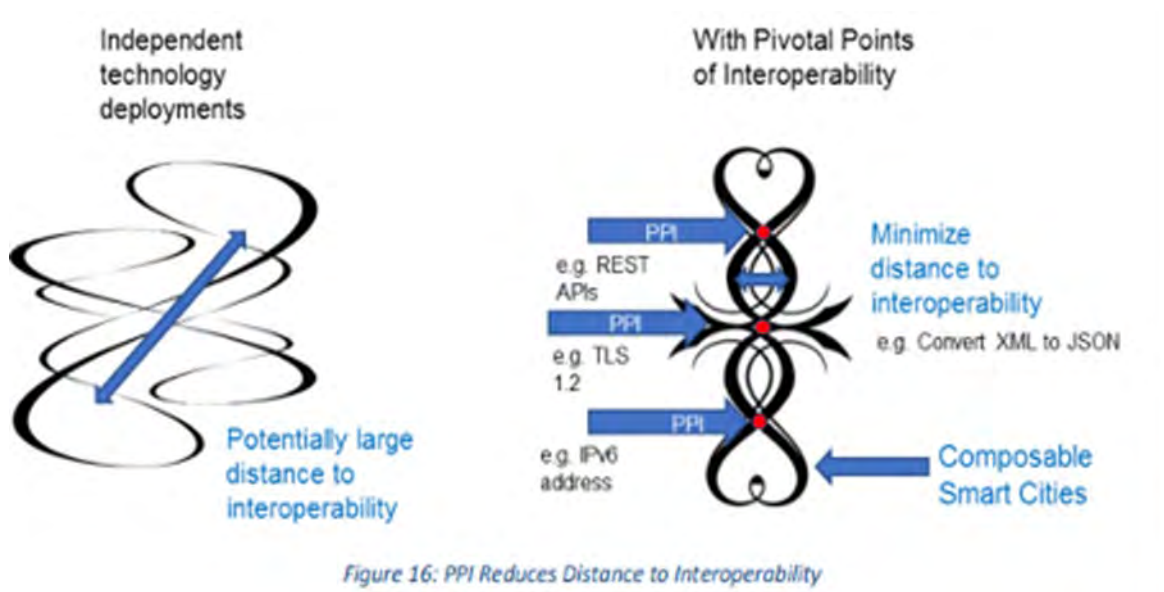


Figure 16: PPI Reduces Distance to Interoperability

図 7.3-4 Pivotal Points of Interoperability

### 7.3.3. 海外における都市 OS の連携

欧州での相互運用の考え方を基に、都市 OS の外部連携における相互運用についても「Semantic Interoperability」、及び、「Technical Interoperability」の二つを一致させる必要があると考える。相互運用の構成要素と、都市 OS が提供する API とデータモデルの関係を図 7.3-5 に示すとおりに定める。



Inter Operability	構成要素	選択肢 (例)
Semantic	語彙体系 (型、コード等)	<ul style="list-style-type: none"> <li>•共通語彙基盤</li> <li>•データ・カタログ語彙(DCAT)</li> <li>•Schema.org, RDFS 等</li> </ul>
	データ項目	<ul style="list-style-type: none"> <li>•政府CIOポータル</li> <li>•FIWARE/SynchroniCity</li> <li>•Open311, GSMA, DATEX II 等</li> </ul>
	データ構造	<ul style="list-style-type: none"> <li>•Schema.org</li> <li>•NGSI/NGSI-LD</li> <li>•RDF+OWL 等</li> </ul>
	API仕様	<ul style="list-style-type: none"> <li>•OAuth2.0/OpenIDConnect</li> <li>•NGSI/NGSI-LD</li> <li>•SPARQL, OData, SQL 等</li> </ul>
	APIモデル	<ul style="list-style-type: none"> <li>•REST/RESTful</li> <li>•GraphQL 等</li> </ul>
	データ形式	<ul style="list-style-type: none"> <li>•JSON/JSON-LD, XML, CSV</li> <li>•Database(RDB, NoSQL) 等</li> </ul>
Technical	通信プロトコル	<ul style="list-style-type: none"> <li>•HTTP/HTTPS</li> <li>•MQTT, CoAP 等</li> </ul>
	トランスポート	<ul style="list-style-type: none"> <li>•TCP, UDP</li> </ul>
	インターネット	<ul style="list-style-type: none"> <li>•IP</li> </ul>
	ネットワーク インターフェース	<ul style="list-style-type: none"> <li>•WWAN, LPWAN, WLAN</li> </ul>

### データモデル

分野や地域を横断してデータを流通するために共通化されたデータ仕様

### 都市OSが提供するAPI

都市OS間連携、サービス連携、アセット/他システム連携にて、共通的に活用される接続仕様

図 7.3-5 都市 OS の API とデータモデルの検討方針

なお、都市 OS が提供する API は以下の方針で記載している。詳細は、「7.3.1 オープン API に用いられる個々の標準規格やシステムの事例」を参照いただきたい。

- 1 API は外部連携に必要となる機能要件と、標準規格となる API 仕様、要件分類を示す。
- 2 API は、標準化団体によって標準仕様を定めているものを採用することが望ましい。
- 3 都市 OS 間で流通するデータは、今後のデータモデルの標準化に追従できることが望ましい。
- 4 API は、スマートシティリファレンスアーキテクチャの継続的な維持・発展により変更となる場合がある。

## 7.4. 都市 OS とオープンソース

---

オープンソースは、都市 OS を構成する多くのビルディングブロックに採用されている重要なソフトウェア形態である。しかし、オープンソースについては「無料」「無保証」などといった側面のみが強調され、誤解されていることも少なくない。正しく理解し、適切に活用することができれば、都市 OS の透明性や拡張性を高め、スマートシティの運営やサービスの発展に大きく寄与する。地域が開発したビルディングブロックやサービスをオープンソースとして公開すれば、コミュニティによる改良や新技術の取り込みが促進され、持続可能かつ柔軟性の高いスマートシティの構築につながる。

本節では、都市 OS にオープンソースを導入することで得られる利点や、地域で開発した成果をオープンソースとして提供する際のポイント、そしてライセンス管理・コンプライアンスに関する考え方について整理する。

### 7.4.1. オープンソースの利点

本節では、オープンソースを活用することで都市 OS にどのようなメリットがもたらされるのか、代表的な観点を示す。

#### 7.4.1.1. デジタル公共財としてのオープンソース

##### 1) 公共インフラとしての信頼性と効率性を高める

オープンソースは、オープンデータと並ぶ「デジタル公共財」であり、都市 OS のような公共インフラストラクチャを支えるソフトウェアとして有用である。ソースコードを公開し、誰でも利用・検証できるため、透明性や拡張性を確保しやすく、公的サービスの信頼性と効率性を高めることが可能となる。

##### 2) 再利用性と新サービスの派生を促進

地域独自に開発した機能やサービスをオープンソース化すれば、他の地域や組織が再利用し、新しいサービスを派生させることも容易である。結果として、スマートシティ全体の発展や、地域間連携が促進される効果が期待できる。

#### 7.4.1.2. バグ対策・改善の迅速化

##### 1) コミュニティによる継続的なメンテナンス

安定したオープンソースは、世界中の開発者やユーザコミュニティがバグやセキュリティ脆弱性を迅速に発見・修正する。自治体や企業が単独で保守する場合に比べ、トラブル対応のスピードが向上しやすい利点がある。

##### 2) 最新技術の取り込みが容易

オープンソースは日々アップデートが行われ、先進的な技術がいち早く導入される場合が多い。都市 OS を常に最新・高機能な状態に保つためにも、オープンソースの活用は効果的である。

#### 7.4.1.3. 透明性とセキュリティ

##### 1) コードの公開による検証性の高さ

ソースコードが公開されているため、セキュリティ面や品質面を専門家や第三者が検証可能である。公共インフラのシステムにおいては、こうした透明性が信頼性を高める大きな要素となる。

##### 2) 改変の自由度の高さ

独自のセキュリティ要件や機能追加が必要になった場合でも、ソースコードを改良できるため、都市や地域ごとの要望に柔軟に対応できる。ベンダー依存が軽減され、機能拡張も容易になる。

#### 7.4.1.4. ロックイン回避とコスト削減

##### 1) ベンダーロックインの回避

ソースコードがクローズドの場合、特定ベンダーに依存しがちである。一方、オープンソースであれば他ベンダーベンダーコミュニティへの引き継ぎも容易となり、長期的なコスト削減や柔軟な運用体制の確保につながる。

##### 2) 共同開発や保守の負担分散

自治体同士や企業、大学などでコストをシェアし合いながら、共同で開発・保守を行うケースもある。結果として、公共財としての成熟が進みやすくなる。

### 7.4.2. オープンソースの提供

本節では、地域が独自に開発した成果をオープンソースとして公開する意義や、逆にオープンソース化しない場合に生じるリスク、そしてコミュニティと連携して責任を共有する仕組みを解説する。

#### 7.4.2.1. 地域発のビルディングブロックをオープンソース化する意義

##### 1) 透明性の確保と改善サイクルの促進

公開されたソースコードは誰でも閲覧・利用が可能になり、多方面からの改良アイデアや修正パッチが集まりやすくなる。地域のスマートシティサービスを広く共有し、さらなるイノベーションを喚起する効果がある。

##### 2) 「税金で作った資産を公開してよいのか」への一石

国内では「税金で作ったソフトウェアを無料で公開してよいのか」という声もある一方、公共財として公開することで、不要な二重投資を避けたり、広域標準化を促進したりできるメリットが大きい。

#### 7.4.2.2. オープンソース化しない場合のリスク

##### 1) 囲い込みによるメンテナンス不足

独自仕様でクローズドに開発されると、メンテナンスや技術継承が限定的になり、新技術への対応が遅れやすくなる。結果としてサービス品質が低下する恐れがある。

##### 2) 改造・機能追加のたびに発注コストが増大

特定ベンダーや独自システムに閉じている場合、新機能を追加するごとに大きな開発費が発生し、予算やスケジュールに制約がかかりやすくなる。

##### 3) ベンダーロックインによる柔軟性の低下

システムの技術やライセンスが閉じているほど、新たにほかのベンダーや技術基盤へ移行する際の障壁が高くなり、結果的に長期運用コストが大幅に膨らむ可能性がある。

#### 7.4.2.3. コミュニティと責任を共有する仕組み

##### 1) コミュニティで育てる文化

「オープンソース化すると責任が持てない」と懸念する声があるが、実際にはオープンソースには「コミュニティで育てる」という文化が根付いている。すべての責任を一者のみが負うのではなく、開発者や利用者が協働して問題を解決していく仕組みがあるため、責任を適切に分担しながら改善を進めることが可能となる。こうしたコミュニティによる責任共有の利点として、品質維持や機能向上が継続的に行われやすい点が挙げられる。特に公共性の高いシステムでは、複数のステークホルダーが連携してアップデートに対応することで、安全性と柔軟性を同時に確保しやすくなる。



## 2) 引き継ぎや人材育成の容易さ

ソースコードやドキュメントが公開されているため、担当者が異動・退職してもプロジェクトの継続がしやすい。新たなメンバーを育成する際にも、オープンな情報が整備されていれば学習コストが低減する。

### 7.4.3. オープンソースのライセンス

本節では、都市 OS のような公共性の高いシステムにおいて留意すべき代表的なライセンスの種類と、スマートシティ構築におけるライセンス選択・管理の考え方を整理する。

#### 7.4.3.1. 代表的なライセンスと特徴

##### 1) MIT License / Apache License

再利用や商用利用の自由度が高いライセンスであり、比較的シンプルなため、都市 OS の実装でも採用例が多い。

##### 2) GPL / LGPL

コピーレフトの考え方を含み、改変や再頒布にも同様のライセンスを課す場合がある。公共性が高いソフトウェアと相性が良い面もあるが、導入時には十分な理解が必要である。

#### 7.4.3.2. スマートシティ構築におけるライセンス選択の考え方

##### 1) 相互運用性と再利用のしやすさを重視

他システムとの連携や、広域的なサービス連携を見据え、利用制限の少ないライセンスを選ぶことが多い。

##### 2) 公共セクター特有の課題への配慮

例えば自治体の調達規則や契約条件によって、ライセンス選定に制約が生じることがあるため、事前に確認が必要となる。

#### 7.4.3.3. ライセンス管理・コンプライアンス

##### 1) 組み合わせによるライセンス衝突リスク

都市 OS で利用されるコンポーネントが多いほど、ライセンスの互換性や表示義務の有無が複雑化し、衝突が生じる可能性が高まる。たとえば GPL ライセンスを含むモジュールと商用ライセンスの組み合わせでは、条件を正しく守らなければ配布時に違反が発生する恐れがある。こうしたリスクを回避するには、導入段階からライセンスの相互依存関係を整理し、必要に応じて専門的な助言を得ることが望ましい。

##### 2) メンテナンスとセキュリティ対応

オープンソースのライセンスであっても、日々公開される修正やパッチに適切に追従できる体制が不可欠である。特に公共サービスではセキュリティリスクを最小化する運用が求められる。