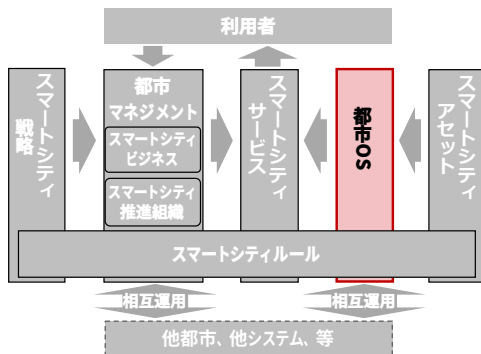


7. 都市 OS

7.1 都市 OS の概要



ここでは都市 OS の概要として、都市 OS の特徴や構成について述べる。

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 の導出方法を図 7.1-1 に示す。都市 OS は、日本のスマートシティの実現に向けた課題から都市 OS の特徴を整理し、そこから「Society 5.0」に基づく日本のスマートシティの考え方や海外スマートシティアーキテクチャ事例を参考に設計された。

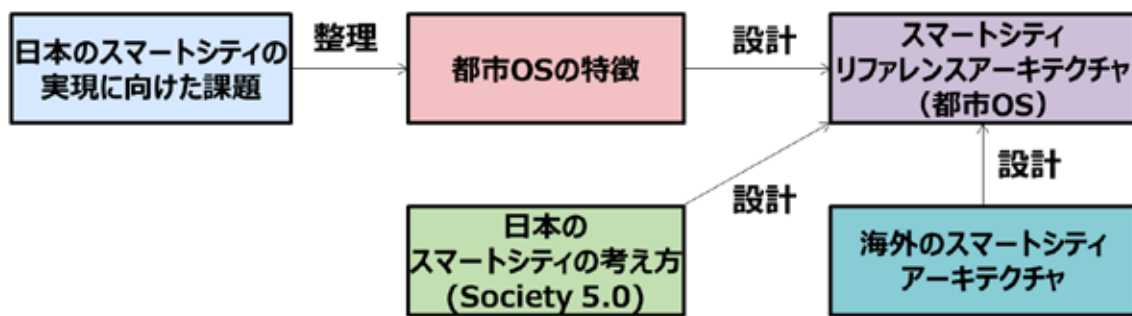


図 7.1-1 スマートシティリファレンスアーキテクチャにおける都市 OS の導出方法

7.1.1 都市 OS の特徴

都市 OS の特徴を図 7.1-2 に示す。日本のスマートシティの実現に向けた課題として、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、の三つがある。①サービスの再利用・横展開において、従来は分野や組織ごとに個別特化したシステムとなっており、そのため他地域への再利用や横展開が困難であるという課題がある。②分野間データ利活用において、従来のサービスは分野や組織ごとにデータが独立しているため、分野間を横断した新サービスの構築が困難という課題がある。③拡張性の低さにおいて、従来の個別特化したシステムでは、機能拡張によるコストや労力が大きくなり、継続的かつ容易にサービスを進化できないという課題がある。

これらの日本のスマートシティの実現に向けた課題への対策として、①相互運用（つながる）、②データ流通（ながれる）、③拡張容易（つづけられる）を都市 OS の特徴として設計する。

- ① 相互運用（つながる）とは、地域内外のサービス連携や各地域における成果の横展開を可能にし、つながりやすくするための仕掛けである。共通的な機能や標準的なインタフェースを具備し、外部に公開する仕組みが必要である。
- ② データ流通（ながれる）とは、分野や組織の壁を越え地域内外の様々なデータを、ひとつの共有された論理的なデータのように見せ、地域内外でデータがながれやすくするための仕掛けである。異種データ（都市 OS 内外の多種多様なデータ）を都市 OS が仲介する仕組みが必要である。
- ③ 拡張容易（つづけられる）とは、地域が解決する課題や目指すべき将来像、及び、スマートシティリファレンスアーキテクチャの更新に合わせ、機能拡張や更新を容易にするといった、都市 OS が継続的に維持・発展しつづけられるための仕掛けである。機能間の疎結合なシステム構築により、必要な機能のみを拡張・更新する仕組みが必要である。

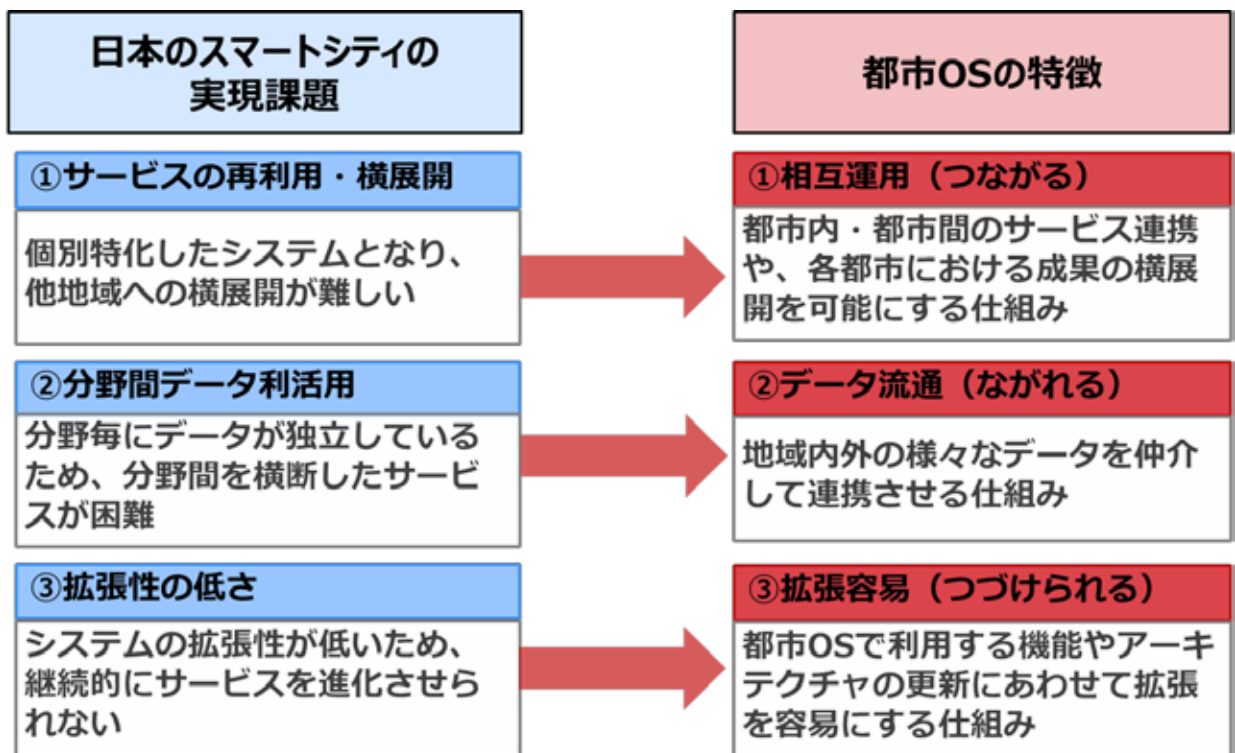


図 7.1-2 都市 OS の特徴

7.1.1.1 相互運用（つながる）

日本の各地域は、図 7.1-3 に示す通り、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。

この相互運用を担保するには、都市 OS が提供する各種機能（API、データ）に対し、以下の二つの方針を取り入れることで、望めば互いに接続できる状態にすることが重要である。

- ① 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
- ② 多様な主体がアクセス可能なよう外部に公開する仕掛け

都市 OS は「7.3.2 都市 OS が提供する API 及びインタフェース」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。詳細は、「7.3.1.3 都市 OS の相互運用」を参照いただきたい。

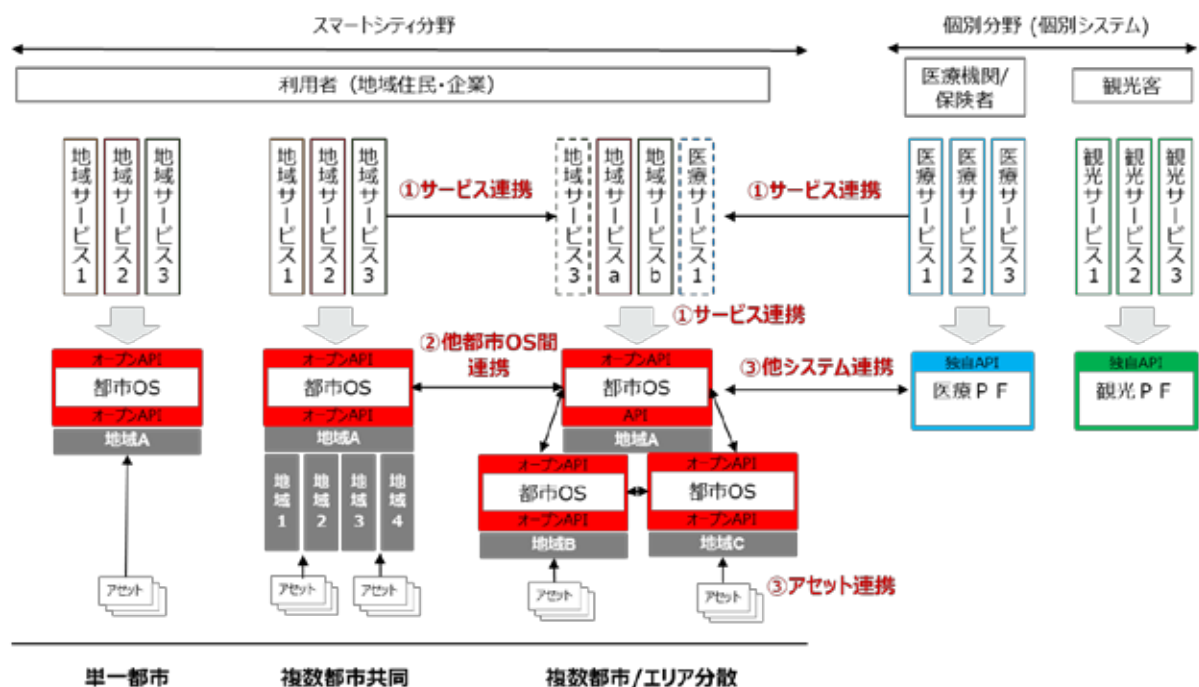


図 7.1-3 都市 OS の相互運用イメージ

都市 OS の相互運用は、①サービス連携、②都市 OS 間連携、③アセット連携/他システム連携の三つがあり、相互運用におけるメリットを表 7.1-1 に示す。

表 7.1-1 相互運用におけるメリット

図との対応	メリット
① サービス連携	都市 OS が提供する API を定めることで、都市 OS とスマートシティサービスが相互運用し、特定の都市 OS だけでなく他都市 OS へ、スマートシティサービスの移植が容易となる。
② 都市 OS 間連携	都市 OS 同士が連携し、都市 OS を通じて他都市 OS のデータを利用できるようにすることで、地域をまたいだ広域でのサービスの提供、地域間のデータ共有による住民の利便性の向上、さらには、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。
③ アセット連携/他システム連携	地域内の多様なスマートシティアセットや、他システムが保持するデータを、都市 OS を介して共有することで、組織やシステムの壁を越えた分野横断型のサービスが提供される。

7.1.1.2 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スマートシティ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」もしくは「Broker」と呼び、①多種多様なデータの取り扱い、②都市 OS 内外のデータを仲介する必要がある。

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表 7.1-2、図 7.1-4 に示す。

表 7.1-2 データ種別の例

項番	データ種別	説明
1	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。
2	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。
3	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、ログデータ、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。
4	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。
5	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。

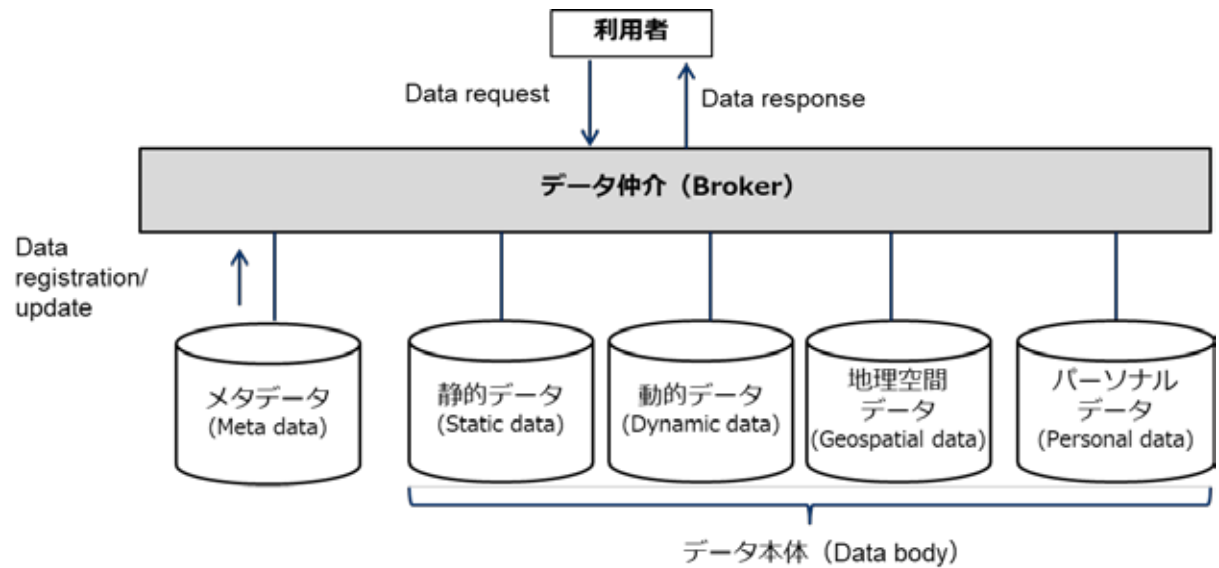


図 7.1-4 都市 OS が扱うデータ種別とデータ仲介

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。利用者は、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。

表 7.1-3 データの仲介方式の分類

仲介方式	説明
データ蓄積方式	都市 OS にデータを蓄積し、一元的に管理する。
データ分散方式	都市 OS にはデータを蓄積せず、都市 OS は分散されたデータの所在情報を管理する。都市 OS は所在情報を利用し、利用者からのデータアクセスに対して、データの仲介をする。

都市 OS のデータ仲介 (Broker) による連携イメージを図 7.1-5 に示す。

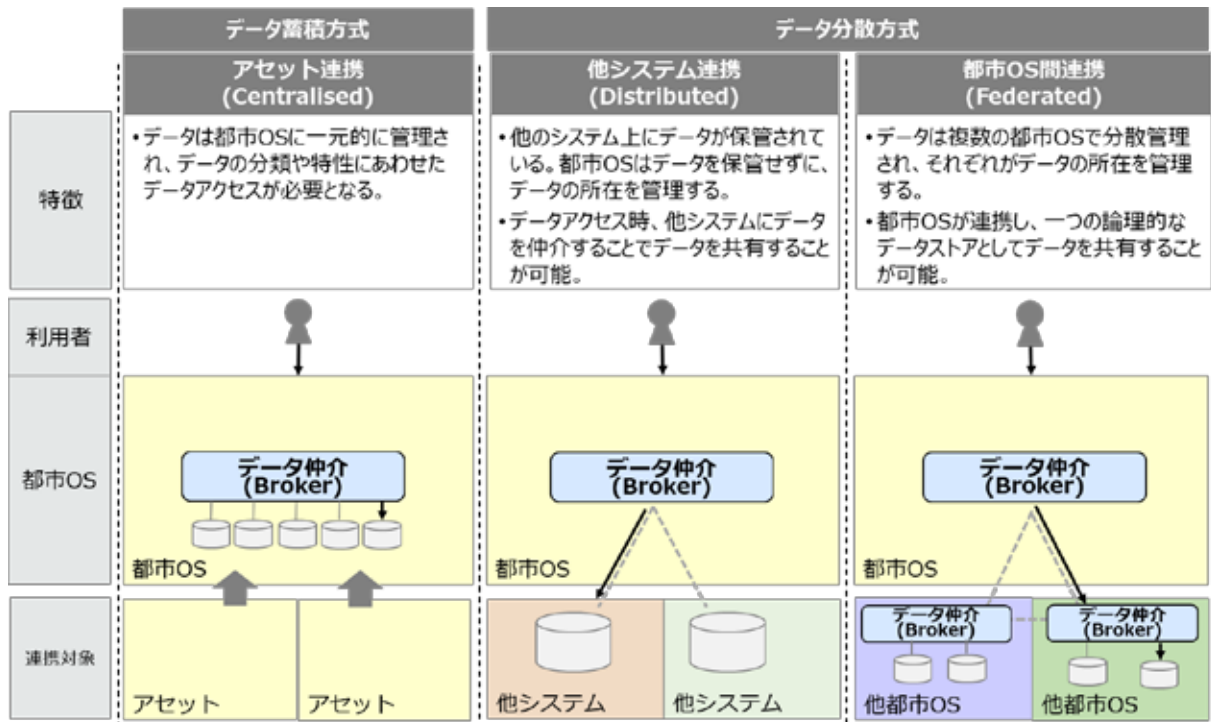


図 7.1-5 都市 OS の連携方法によるデータ仲介例

このようなデータ仲介により、分散された多種多様なデータを集約し活用する上でのメリットとして、都市デジタルツインによる現実世界の仮想化が一例として挙げられる。都市デジタルツインとは、都市空間情報と IoT によるリアルタイム情報を軸とした都市活動・環境情報から構成され、モビリティ、観光、防災、インフラ維持管理、環境、エネルギー、イノベーション創発等、多岐にわたるスマートシティサービス分野において、リアルタイムに都市の状況を可視化し、エネルギー、人流、交通流のシミュレーション、全体最適化、予測、データ駆動型意思決定支援等、都市空間上における高度な情報処理を可能にするデジタル環境と定義可能である（参照：「付録 D 都市デジタル化の動向」）。

近年の都市デジタルツイン化の動向から、構成要素である都市空間情報と都市活動・環境情報は図 7.1-6 に示すように分類できる。

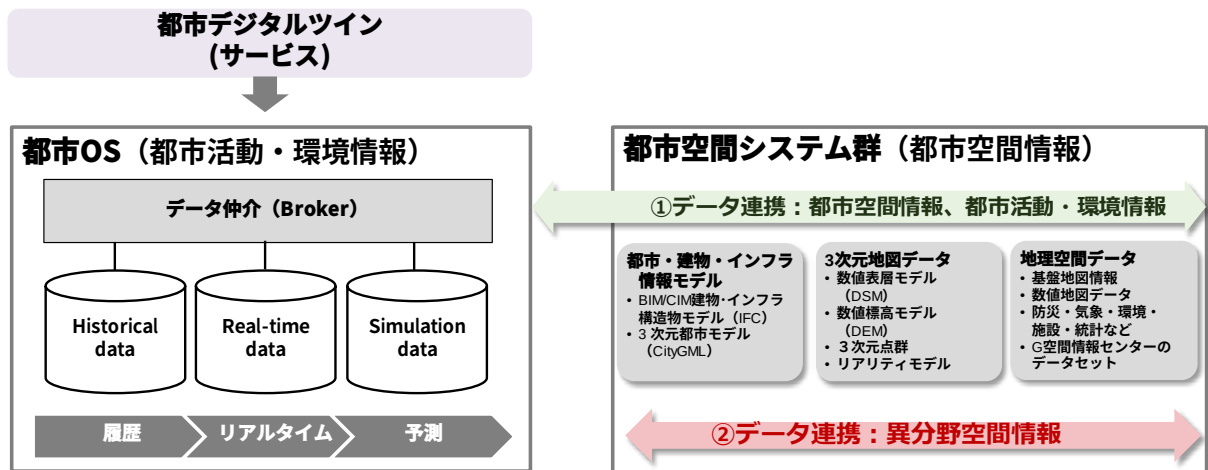


図 7.1-6 都市デジタルツイン

Ⅰ 都市 OS (都市活動・環境情報)

- リアルタイム（動的データ）：位置情報が付加された IoT、ソーシャルメディアデータ等。
- 履歴（静的データ）：リアルタイムデータに位置と時間の情報が付加されたデータ等。
- 予測（静的データ）：人流、交通流、風・光・音環境、防災等のシミュレーション結果データ。

Ⅰ 都市空間システム群（都市空間情報）

- 都市・建物・インフラ情報モデル：都市空間、建物、インフラ構造物の属性情報を含む 3 次元情報モデル。データモデルとして CityGML³²、IFC (Industry Foundation Classes)³³等が活用されてきている。
- 3 次元地図データ：数値表層モデル、数値地形モデル、3 次元点群、3 次元幾何形状中心のリアリティモデル。
- 地理空間データ：基盤地図データ、数値地図データ及び、防災、気象、環境、施設、統計分野の各種地理空間データ。G 空間情報センター³⁴からアクセス可能なデータセットが該当。

³² CityGML <https://www.ogc.org/standards/citygml>

³³ IFC <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>

³⁴ G 空間情報センター https://www.geospatial.jp/gp_front/

7.1.1.3 拡張容易（つづけられる）

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変更等を、視覚的な操作で容易に組み替えることが望ましい。

また、都市 OS が特定ベンダに依存しないよう、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 7.1-7 に示す。

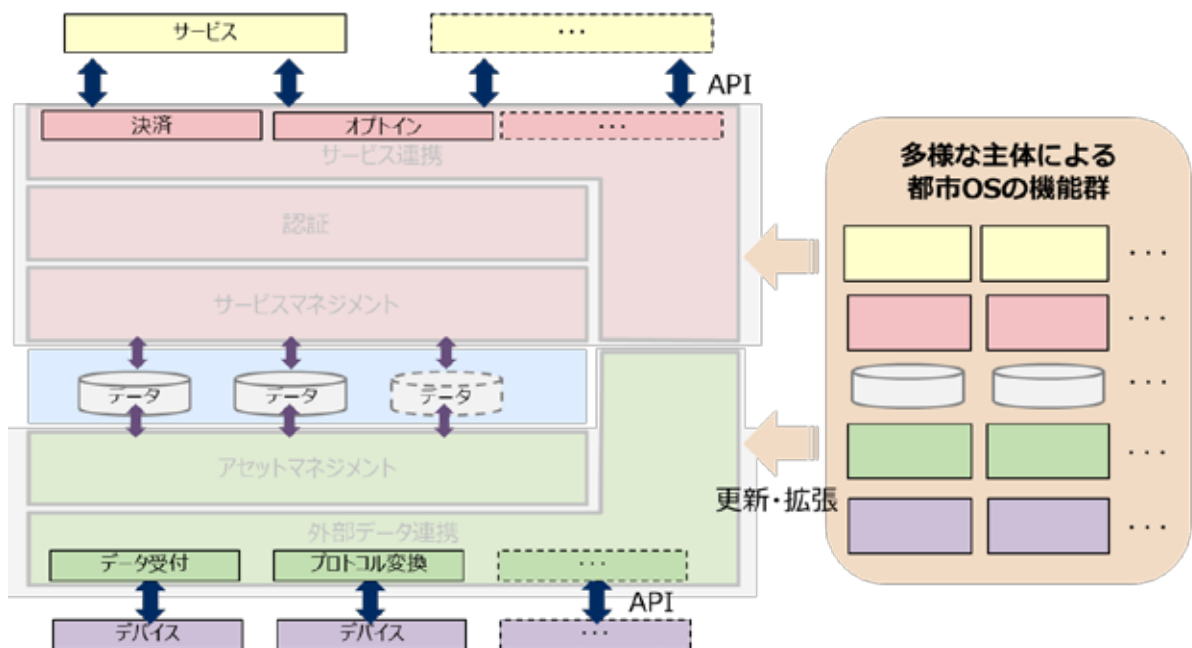


図 7.1-7 ビルディングブロック方式による構築例

7.1.2 都市 OS のアーキテクチャ構成要素の導出

都市 OS の特徴や国内外のスマートシティの考え方を踏まえ、都市 OS のアーキテクチャの要素を導出する。

7.1.2.1 日本のスマートシティの考え方

都市 OS のアーキテクチャを検討するにあたって、第 5 期科学技術基本計画にて提唱された Society 5.0 の定義を参照した。内閣府の科学技術政策ページには次の記述がある。

『Society 5.0 で実現する社会は、IoT（Internet of Things）で全ての人とモノがつながり、様々な知識や情報が共有され、今までにない新たな価値を生み出すことで、これらの課題や困難を克服します。』

『Society 5.0 は、サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより実現します。』

これらを拠り所とし、スマートシティのサイバー空間とフィジカル空間における人とモノを定義することから検討を行った。

スマートシティにおける人とは、フィジカル空間においてスマートシティの取組を行う地域に関わりを持つ人そのものであるが、サイバー空間においてはスマートシティの利用者であると定義した。また、スマートシティにおけるモノとは、フィジカル空間ではその地域にある全てを指すが、サイバー空間においてはデータの送受信源となり得るデバイス（センサやモバイルデバイス等）と定義した。それに加え、スマートシティのサイバー空間においては、ユーザがデバイスからの送受信データを利活用するための仕組みをサービスと定義した。これはフィジカル空間においてはサービス提供者や他システムに位置する。

スマートシティにおけるフィジカル/サイバー空間と人/モノの関係

	フィジカル空間	サイバー空間
人	地域に関わる人そのもの	ユーザ
モノ	地域に存在する全てのモノ	デバイス
	事業者/他システム	サービス

図 7.1-8 スマートシティにおけるフィジカル/サイバー空間と人/モノの関係

また、スマートシティのサイバー空間における人の行動履歴として、ユーザがどのデバイスを使ってどのサービスを使用したかという関係に時系列情報や位置情報を付与した情報であるサービス利用履歴を定義した。そして、サイバー空間におけるこれらのユーザ・デバイス・サービス・サービス利用履歴を管理する枠組みを都市 OS として定義した。

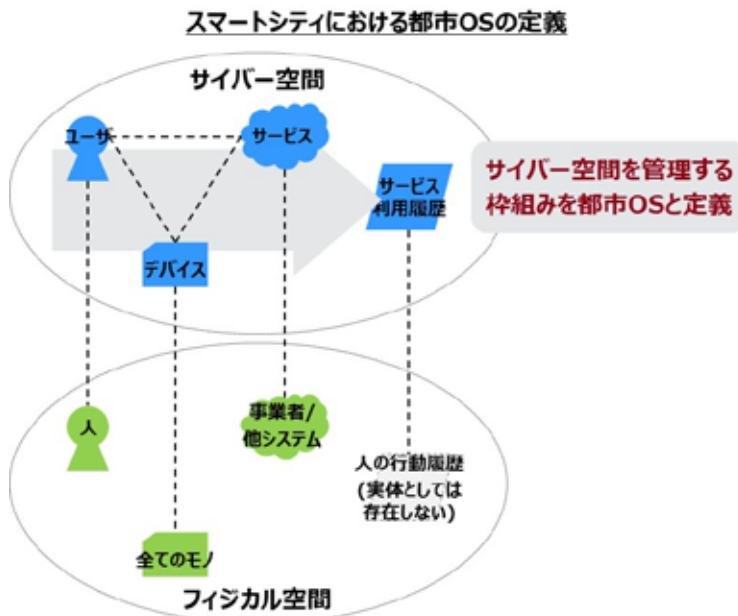


図 7.1-9 都市 OS の定義

次に、利用者・デバイス・サービス・サービス利用履歴を管理する枠組みである都市 OS をアーキテクチャの構成要素として以下の八つに分解した。

- ① 利用者を管理するためのアーキテクチャ要素として、ユーザ ID の管理及び認証・認可を担う認証を定義。
- ② デバイスは、Society 5.0 の考え方ではアセットとして括られるため、その管理のためのアセットマネジメントを定義。
- ③ アセットからのデータ送受信には、新旧の様々なテクノロジーの活用や分野間連携が求められるため、外部データ連携を定義。
- ④ サービスについては、サービス自体の管理とともに、サービス利用履歴を記録する要素としてサービスマネジメントを定義。
- ⑤ サービスと都市 OS の API 接続を担保するためにサービス連携を定義。
- ⑥ 認証・アセットマネジメント・外部データ連携・サービスマネジメント・サービス連携の 5 要素から生まれる、サービス利用履歴含む全てのデータを保持するデータマネジメントを定義。

- ② この六つのアーキテクチャ要素に加えて、通常のシステム運用で必須となる、運用とセキュリティを定義。

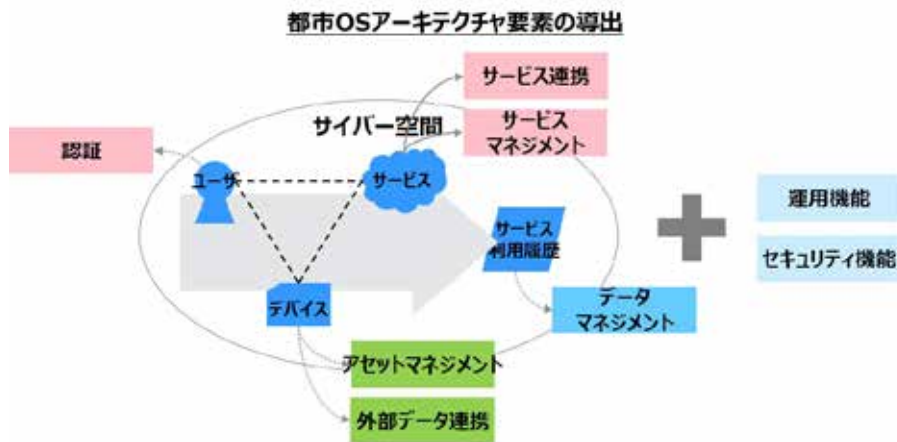


図 7.1-10 都市 OS アーキテクチャ要素の導出

7.1.2.2 都市 OS の特徴とスマートシティリファレンスアーキテクチャとの関係

都市 OS の三つの特徴と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 との関係を図 7.1-11 に示す。都市 OS の三つの特徴を以下の通り反映した。

- ① 特徴①相互運用（つながる）からのアーキテクチャへの反映ポイントは、相互運用のベースになる認証、サービス連携とサービスのマネジメント機能、スマートシティアセット（外部データ連携）とアセットのマネジメント機能である。
- ② 特徴②データ流通（ながれる）からのアーキテクチャの反映ポイントは、外部データからの仲介を実現するデータマネジメント機能と外部データ連携機能である。
- ③ 特徴③拡張容易（つづけられる）からのアーキテクチャの反映ポイントは、先に述べた六つの要素に加え、セキュリティと運用機能である。都市 OS の特徴をアーキテクチャの関連する要素に反映させることで、各要素の位置づけが整理できる。

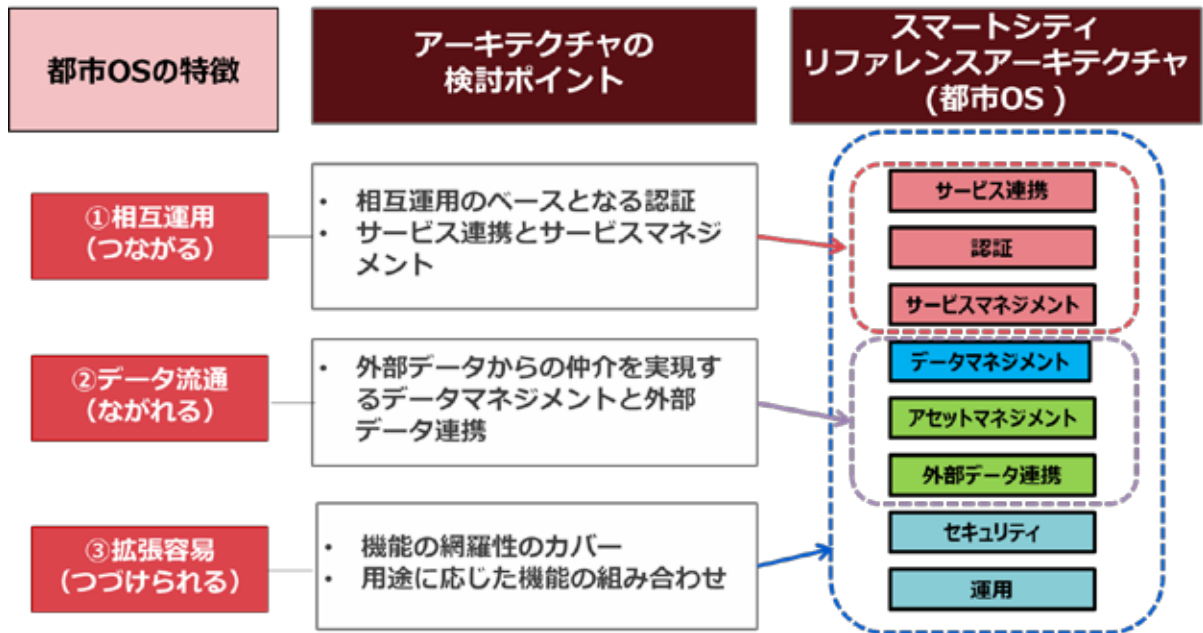


図 7.1-11 都市 OS の特徴とスマートシティリファレンスアーキテクチャ (都市 OS) の関係

7.1.2.3 海外スマートシティアーキテクチャの参考ポイント

都市 OS の設計において、海外の主要なスマートシティアーキテクチャを参考にした。各アーキテクチャの概要その参考ポイントを表 7.1-4 に示す。各アーキテクチャの詳細については「付録 C 海外のスマートシティアーキテクチャ」を参照いただきたい。

表 7.1-4 海外スマートシティアーキテクチャの参考ポイント

アーキテクチャ	概要	参考ポイント	関連章
Synchronicity ³⁵	スマートシティに関する欧州の IoT パイロットであり、現在 20 都市が参加した大規模な取組。	- 都市 OS の各機能群における構成要素とその定義 - 最小限相互運用メカニズム (Minimal Interoperability Mechanisms、MIMs) における、API、データモデルの考え方 - 認証系 API、データマネジメント系 API - アーキテクチャ維持組織の機能	7.2 都市 OS の機能説明 7.3 外部連携 9.1.1 アーキテクチャの維持発展を可能とする各種取組
FIWARE ³⁶	FI-PPP が次世代インターネット技術における欧州の競争力強化と、社会・公共分野のスマートアプリケーション開発を支援するために、開発した基盤ソフトウェア。	- 都市 OS の各機能群における構成要素とその定義 - 認証系 API、データマネジメント系 API	7.2 都市 OS の機能説明 7.3 外部連携
X-Road ³⁷	エストニア政府が整備した安全なデータ交換のためのプラットフォーム。	アーキテクチャ維持組織の機能	9.1.1 アーキテクチャの維持発展を可能とする各種取組
IndiaStack ³⁸	インド政府が生体認証技術を活用した個人を一意に識別する番号として Aadhaar を開発、Aadhaar を活用するデジタルインフラとしての API 群 (e-KYC、e-Sign 等) を含めた総称。	個人に関する認証 (個人認証)	7.2.2 認証

³⁵ <https://synchronicity-iot.eu/>³⁶ <https://www.fiware.org/>³⁷ <https://x-road.global/>³⁸ <https://www.indiastack.org/>

アーキテクチャ	概要	参考ポイント	関連章
IES-City ³⁹	NIST (National Institute of Standards and Technology; 米国国立標準技術研究所) が主導して定めたコンセンサスフレームワーク。	相互運用ポイントである Pivot al Points of Interoperability (PPI) の考え方	7.3 外部連携

7.1.2.4 都市 OS のスマートシティリファレンスアーキテクチャ

先に説明した、国内外のスマートシティの考え方や都市 OS の特徴を踏まえ、導出したスマートシティリファレンスアーキテクチャの都市 OS の構成要素を説明する。都市 OS は、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）や他システム・他都市 OS から収集したデータを保管し、都市 OS 内外のサービスへ連携する役割を担うことから八つの要素から構成される。都市 OS の全体像を図 7.1-12 に示す。

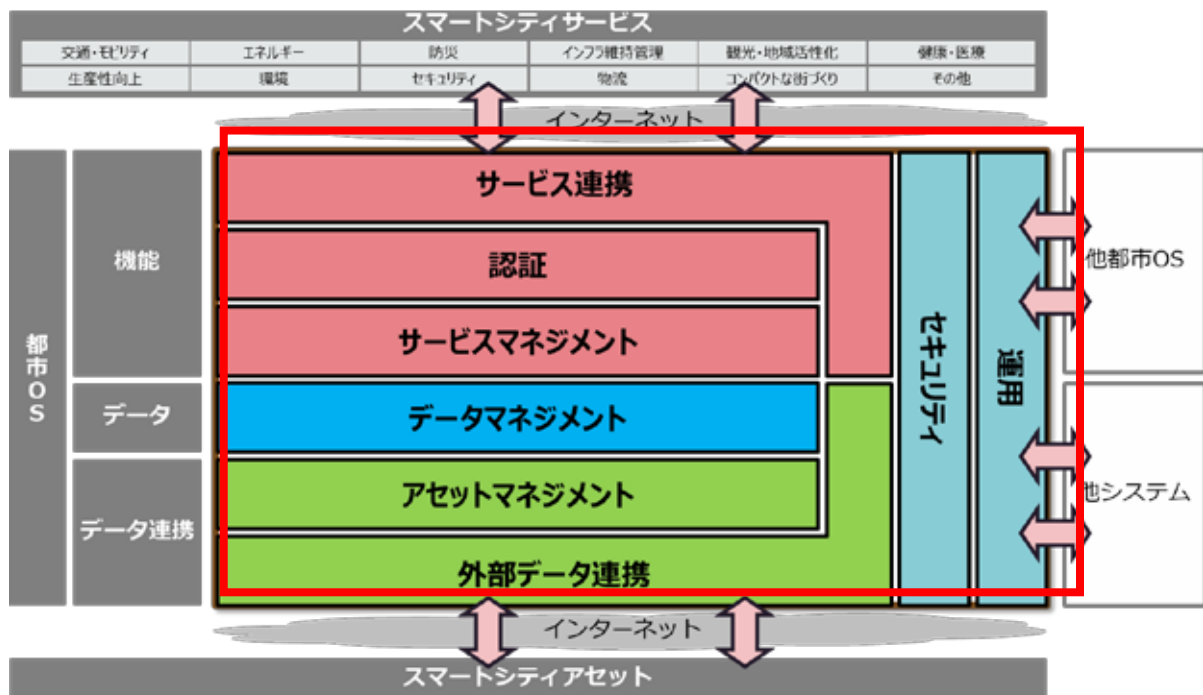


図 7.1-12 都市 OS の全体像

図 7.1-12 の赤枠内に示した八つの機能群の定義を表 7.1-5 に示す。

³⁹ <https://pages.nist.gov/smartcitiesarchitecture/>

表 7.1-5 都市 OS の機能群概要

構成層	機能群	定義	参照章
機能 (サービス)	サービス連携	都市 OS 上で動作する各種サービスと連携する機能や API を提供。共通サービスやオープン API を提供し、API 管理や都市 OS 間連携の機能を持つ。	7.2.1
	認証	利用者、または、スマートシティサービス、他都市 OS に対して、用途に応じた認証方法を提供。認証・認可やユーザ管理の機能を持つ。	7.2.2
	サービス マネジメント	都市 OS 上で動作するスマートシティサービスを管理する機能を提供。サービス管理やサービス利用履歴管理の機能を持つ。	7.2.3
データ	データ マネジメント	都市 OS に保存・蓄積するデータの管理や、地域内外に分散されたデータを仲介する機能を提供。データ仲介やデータ管理の機能を持つ。	7.2.4
データ連携	アセット マネジメント	都市 OS と連携するスマートシティアセットや他システムの管理と、スマートシティアセットへの制御を実行する機能を提供。デバイス管理やシステム管理の機能を持つ。	7.2.5
	外部データ 連携	スマートシティアセット、または、他システムとのインタフェースを管理し、データフォーマットやプロトコル差異を吸収する機能を提供。データ処理やデータ伝送の機能を持つ。	7.2.6
共通機能	セキュリティ	都市 OS の内外部の脅威から都市 OS を防御するために必要な機能を提供。	7.2.7
	運用	都市 OS の IT システム運用に必要なシステム管理機能や管理プロセスを提供。	0

特に、都市 OS との連携におけるインタフェースの考え方については、「7.3 外部連携」を参照いただきたい。

7.2 都市 OS の機能説明

都市 OS は図 7.2-1 に示すように、「サービス連携」、「認証」、「サービスマネジメント」、「データマネジメント」、「アセットマネジメント」、「外部データ連携」の 6 種類の機能群に、各機能群共通の「セキュリティ」、「運用」の 2 種類の機能群を加えた全 8 機能群から構成される。

それぞれの機能群には、個別機能をひとまとまりとした機能ブロックが含まれる。その機能ブロックの説明と、個別機能の要件を次節より説明する。なお、地域が解決する課題や目指すべき将来像に応じて、必要となる各要件を取捨選択し、都市 OS が個別機能を具備する必要がある。なお、「付録 A 都市 OS の要件一覧」に再掲し、都市 OS ユースケースの分類例に対し、どの要件を選択したかについても例として示す。

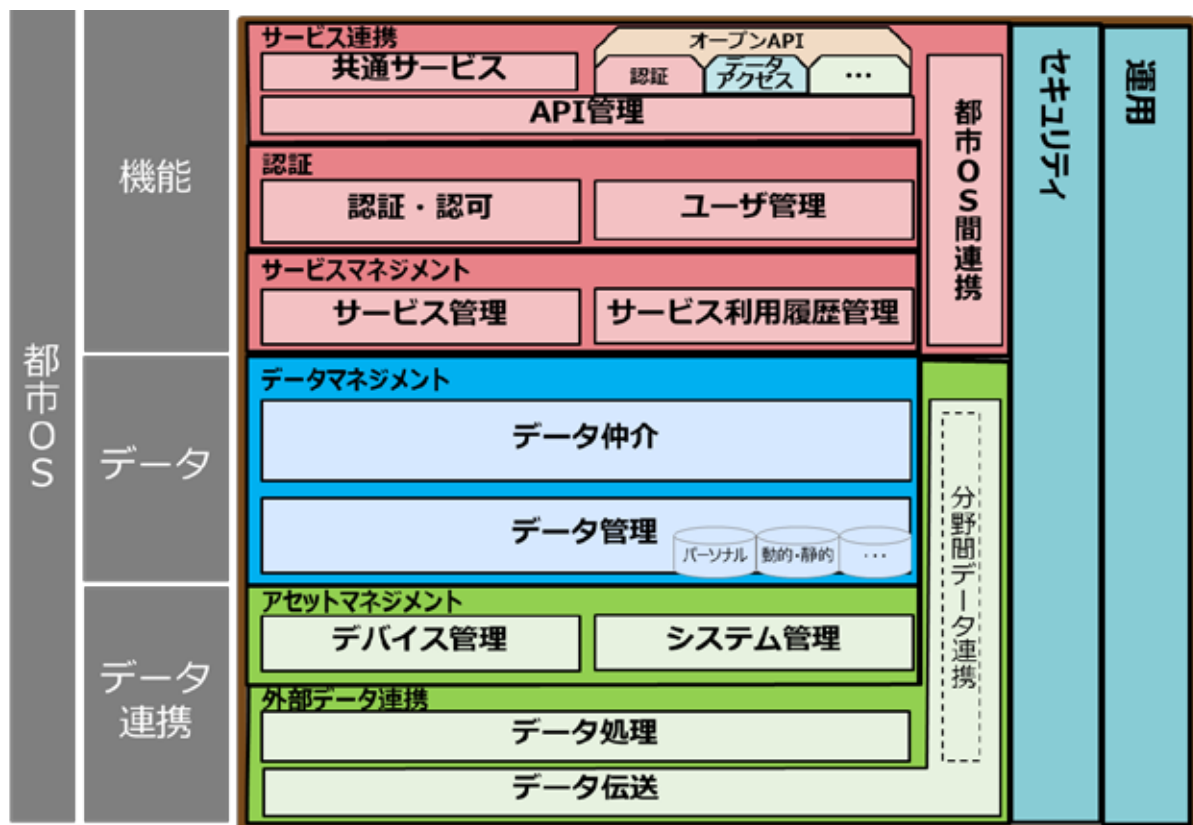


図 7.2-1 スマートシティリファレンスアーキテクチャにおける都市 OS の詳細

7.2.1 サービス連携

サービス連携は、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を提供する。サービス連携は、「共通サービス」、「オープン API」、「API 管理」、「都市 OS 間連携」の機能ブロックで構成される。

7.2.1.1 サービス連携の機能ブロック説明

(1) 共通サービス

共通サービスは、地域や分野を問わず共通的に使われるサービスを、協調領域として都市 OS が提供する機能である。主にユーザインタフェースを通じて提供され、例えば、都市 OS 利用者向けの開発ポータルサイトや、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サイト等が挙げられる。

(2) オープン API

オープン API は、都市 OS が提供する API を、他の主体が利用するために外部公開された接続仕様である。API 管理と連携し外部公開され、サービス連携や都市 OS 間連携に必要な認証やデータアクセス等が挙げられる。オープン API は、外部データ連携におけるスマートシティアセットや他システムとの連携でも活用される場合がある。

(3) API 管理

API 管理は、都市 OS が提供する各種機能を API として管理・公開するものである。管理対象とする API は、外部向けに公開される API だけではなく、都市 OS の内部でのみ利用される API も対象とする。API を管理・公開するための API ライフサイクル管理や、API の流量を制御するための API ゲートウェイ等が挙げられる。

(4) 都市 OS 間連携

都市 OS 間連携は、都市 OS の各種機能やデータ等を都市 OS 間で連携するための機能である。利用者が他の都市に移動した際にも同一の資格情報を利用可能とする認証連携や、都市 OS 間でデータを共有するためのデータ連携が挙げられる。

7.2.1.2 サービス連携の機能要件

共通サービスにおける機能要件を表 7.2-1 に示す。

表 7.2-1 共通サービスの機能要件

項番	個別機能	説明
1	開発ポータルサイト	都市 OS 利用者向けに、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イト	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、配信等を行う機能を提供できること。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	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を提供できること。
4	コンテンツ管理	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、配信等を行う機能を提供できること。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	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を提供すること。
6	オプトイン管理	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を提供できること。
7	可視化・分析ダッシュボード	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を提供できること。 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。

オープン API の機能要件については、「7.3 外部連携」を参照いただきたい。

特に「7.3.2.1 認証系 API」や「7.3.2.2 データマネジメント系 API」が都市 OS 間連携において必要となる。

API 管理における機能要件を表 7.2-2 に示す。

表 7.2-2 API 管理の機能要件

項番	個別機能	説明
1	API ライフサイクル管理	都市 OS 上の API のライフサイクル（登録、参照、変更、削除）を管理できること。都市 OS 上に API のエンドポイントを提供し、利用者が API を利用可能となる。
2	API ゲートウェイ	API の使用量制限やネットワーク速度制限、複数 API の集約等を実行する。

都市 OS 間連携における機能要件を表 7.2-3 に示す。

表 7.2-3 都市 OS 間連携の機能要件

項番	個別機能	説明
1	認証連携	他の都市 OS と連携し、他の都市 OS 利用者の認証情報を基に、利用者からの認証要求に対応できること。 「7.3.2.1 認証系 API」に記載した機能要件に従い都市間の認証連携を実現可能な機能提供を行うことが望ましい。
2	データ仲介	他の都市 OS と連携し、利用者に他の都市 OS のデータを提供できること。 なお、本機能は、「7.2.4.1（1）データ仲介」で定義しているものと同等である。「7.3.2.2 データマネジメント系 API」に記載した機能要件に従い API を公開・連携することで、都市 OS 間のデータ連携が可能となる。

7.2.2 認証

認証は、都市 OS の利用者や、都市 OS と連携するアプリケーションや他システムに対して、都市 OS が用途に応じて認証方法を提供する機能群である。認証は、「認証・認可」、「ユーザ管理」の機能ブロックで構成される。

7.2.2.1 認証の機能ブロック説明

(1) 認証・認可

認証・認可は、大きく認証機能と認可機能の二つに大分される。

認証機能は、都市 OS の利用者が誰であることを識別する機能である。本機能による利用者の識別と、認可機能による利用権限確認により都市 OS の各種機能の利用可否を判断することができる。都市 OS におけるユーザは利用者やアプリケーション等、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フェースを提供することが求められる。

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である。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。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。

(2) ユーザ管理

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 の利用者は都市 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。

7.2.2.2 認証の機能要件

認証・認可における機能要件を表 7.2-4 に示す。

表 7.2-4 認証・認可の機能要件

項番	個別機能	説明
1	認証	「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、アカウントを特定できること。
2	認可	「ユーザ管理」と連携し、アカウントに紐づくロールやポリシーを基に、都市 OS の各種機能や管理するデータの利用範囲を許可・制限できること。
3	個人認証	パーソナルデータを利用する場合、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、セキュアに本人を特定できること。 ※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。
4	シングルサインオン	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できること。 利用者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事が望ましい。

ユーザ管理における機能要件を表 7.2-5 に示す。

表 7.2-5 ユーザ管理の機能要件

項番	個別機能	説明
1	アカウント管理	利用者を特定の 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）を管理できること。
2	ロール管理	利用者が所属するグループ（利用者、管理者等）を定義するロールを管理できること。
3	ポリシー管理	アカウントやロール別に、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。

7.2.3 サービスマネジメント

サービスマネジメントは、都市 OS と連携するスマートシティサービスを管理し、適切に運用するための機能群である。サービスマネジメントは、「サービス管理」、「サービス利用履歴管理」の機能ブロックで構成される。

7.2.3.1 サービスマネジメントの機能ブロック説明

(1) サービス管理

サービス管理は、都市 OS と連携するサービスを管理・公開する機能である。サービスの提供者は、都市 OS 運用者かサービス提供者かは問わない。

また、利用者とスマートシティサービスの紐づけを管理し、利用者はサブスクリプションを基にサービスの利用形態を変更することが可能となる。

(2) サービス利用履歴管理

サービス利用履歴管理は、利用者による都市 OS が連携するスマートシティサービスの利用状況を蓄積する機能である。蓄積したサービス利用履歴は、利用者によるオプトインを前提として、利用者個人の興味関心に沿った最適なパーソナライズドサービスを提供する等に活用することができる。

7.2.3.2 サービスマネジメントの機能要件

サービス管理における機能要件を表 7.2-6 に示す。

表 7.2-6 サービス管理の機能要件

項番	個別機能	説明
1	サービスライフサイクル管理	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）を管理できること。 都市 OS が管理するサービスの一覧は、「サービス連携」と連携し、利用者に公開されることが望ましい。
2	サブスクリプション管理	利用者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）を管理できること。

サービス履歴管理における機能要件を表 7.2-7 に示す。

表 7.2-7 サービス履歴管理の機能要件

項番	個別機能	説明
1	利用履歴管理	利用者の同意のもと、利用者による都市 OS やスマートシティサービスの利用履歴の蓄積・公開する機能を提供すること。

7.2.4 データマネジメント

データマネジメントは、都市 OS に保存・蓄積するデータの管理、及び単一都市・複数都市や他システムに分散されたデータを仲介する機能を提供する機能群である。データマネジメントは、「データ仲介」、「データ管理」の機能ブロックで構成される。

7.2.4.1 データマネジメントの機能ブロック説明

(1) データ仲介

データ仲介機能は、都市 OS 内外に点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である。都市 OS 内に蓄積されるデータ、及び、他都市 OS や他システムに分散されるデータに対し、利用者は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。

(2) データ管理

データ管理機能は、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・蓄積する機能である。

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 にて管理する必要がある。これにより地域をまたいだ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。

7.2.4.2 データマネジメントの機能要件

データ仲介における機能要件を表 7.2-8 に示す。

表 7.2-8 データ仲介の機能要件

項番	個別機能	説明
1	データ蓄積	都市 OS が管理するデータに対し、「データ管理」と連携しデータを処理（登録・参照・更新・削除）できること。
2	データ分散	他都市 OS や他システムに分散するデータに対し、データを仲介（登録・参照・更新・削除）できること。
3	イベント処理	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できること。 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を提供可能となる。

データ管理における機能要件を表 7.2-9 に示す。

表 7.2-9 データ管理の機能要件

項番	個別機能	説明
1	データストア	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、適切に蓄積・活用できること。 データの分類として、パーソナルデータやリアルタイムデータ等がある。リアルタイムデータ等の連続したデータを時系列で確認できるよう履歴を管理できることが望ましい。
2	ユニーク ID 管理	都市 OS が管理するデータそれぞれにユニークな ID を管理し、地域をまたいだ様々なデータの中から一つのデータを特定可能とする仕組みを提供する。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。

7.2.5 アセットマネジメント

アセットマネジメントとは、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は、「デバイス管理」、「システム管理」の機能ブロックで構成される。

7.2.5.1 アセットマネジメントの機能ブロック説明

(1) デバイス管理

デバイス管理は、都市 OS に接続する IoT デバイス等の状態を管理・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である。本機能により、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。

(2) システム管理

システム管理は、都市 OS に接続する他システムの認証情報、接続情報、契約情報等の他システムに接続するために必要な情報を管理するための機能、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ための機能である。本機能により、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。

7.2.5.2 アセットマネジメントの機能要件

(1) デバイス管理

デバイス管理における機能要件を表 7.2-10 に示す。

表 7.2-10 デバイス管理の機能要件

項番	個別機能	説明
1	デバイスライフサイクル登録	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）を管理できること。
2	デバイス状態管理	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、公開できること。
3	デバイス制御（アクチュエーション）	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できること。
4	デバイス監視	接続されているデバイスの死活状況を監視、もしくは、デバイスから送信される障害のイベントの監視ができること。
5	デバイス認証	事前に登録されたデバイスのみアクセスを許可することができること。

(2) システム管理

システム管理における機能要件を表 7.2-11 に示す。

表 7.2-11 システム管理の機能要件

項番	個別機能	説明
1	システムライフサイクル登録	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）を管理できること。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	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、公開できること。

7.2.6 外部データ連携

外部データ連携は、スマートシティアセットや他システムとのインタフェースを管理し、データモデルやプロトコルの差異を吸収する機能群である。外部データ連携は、「データ処理」、「データ伝送」の機能ブロックで構成される。また、「データ伝送」は、「分野間データ連携（分野間データ連携コネクタ）」と呼ばれる、分野間やシステム間を横断したデータ連携の概念も含む。

7.2.6.1 外部データ連携の機能ブロック説明

(1) データ処理

データ処理は、スマートシティアセットや他システムから収集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である。本機能により収集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、分野間のデータ利活用が実現できる。データ処理機能は、スマートシティアセットと他システムで求める要件に差異はない。

(2) データ伝送

データ伝送は、接続プロトコルの差異を吸収する機能である。複数のプロトコルでの接続をサポートし、スマートシティアセットや他システムからの様々な接続要件に対応する。本機能で接続プロトコルの差異を吸収することで、上位層の標準インタフェースでデータを連携することができる。データ伝送機能は、スマートシティアセットと他システムでは求める要件が異なる。

スマートシティアセットとの接続は、デバイスへの制御指示を送信できる必要があるため、双方向通信をサポートするプロトコルが求められる。またハードウェアや通信環境の制約から軽量な通信方式が望ましい。

他システムとの接続は、対象が既存の IT システムであることが多く、独自インタフェースを採用していることが多い。独自インタフェースに対応するため、様々な接続方式に汎用的に対応できる仕組みが求められる。また他の都市 OS とデータ相互利用を実現するための仕組み（収集データの転送、データ取得要求の転送等）が必要である。

(3) 分野間データ連携（分野間データ連携コネクタ）

現在、都市 OS が、多種・多様な他システムとデータを受信・送信する場合、実際にデータを連携するまでに、他システムとの仕様検討や調整等の作業が他システムの数・種類だけ必要であり、煩雑で工数を要している。この課題を解決するため、都市 OS と、接続を行う他システムの双方に対し分野間データ連携コネクタを用いることにより、統一的な仕組みで容易に相互連携が実現され、スムーズなデータ連携が可能となる。

分野間データ連携コネクタは、現在研究開発が進められているプロジェクト⁴⁰であり、このコネクタでは、分散されたデータの検索、データ交換制御、データ交換履歴に関する機能等が将来的に提供される予定となっている。特にデータ交換履歴の機能では、単なるデータ交換ログを記録するのではなく、ブロックチェーンの技術を用いて、データ品質における原本性が担保される予定である。都市 OS においても、外部データ連携機能の開発を容易にする観点で、分野間データ連携コネクタと接続するための機能を将来的に具備することが望ましい。

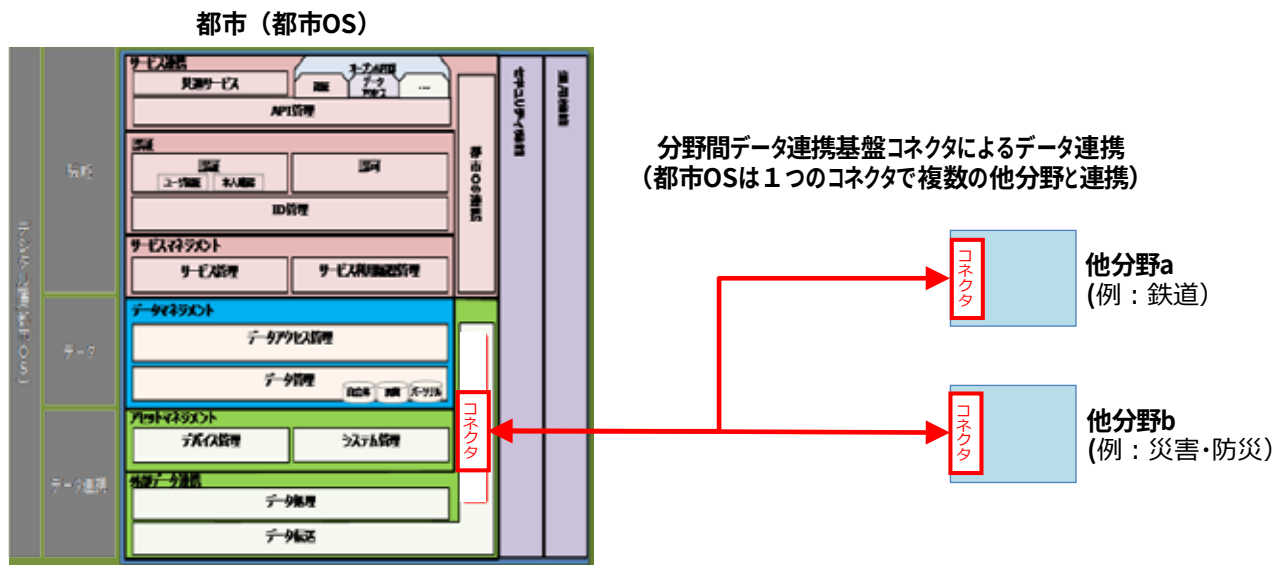


図 7.2-2 分野間データ連携基盤コネクタによる他分野とのデータ連携イメージ

7.2.6.2 外部データ連携の機能要件

データ処理における機能要件を表 7.2-12 に示す。

表 7.2-12 データ処理の機能要件

項番	個別機能	説明
1	データ変換	外部から取得したデータを都市 OS が扱える形式に変換できること。変換対象は、語彙や、形式、項目等が存在するが、取り扱うデータにより変換対象が異なる。
2	データ受付 (キューイング)	都市 OS にデータを蓄積するため、データアクセス（登録・参照）を受け付けること。連携対象は、スマートシティアセットや、他システム等があげられる。

⁴⁰ 戦略的イノベーション創造プログラム（SIP）第2期／ビッグデータ・AIを活用したサイバー空間基盤技術 分野・組織を超えたデータ活用とサービス提供を実現する基盤の研究

https://www.nedo.go.jp/activities/ZZJP2_100126.html

3	データ取得 (クローリング)	定期的に他システムを巡回し、データを取得できること。
4	データ補完	リアルタイムデータ等で欠損したデータを補完し、データ品質を向上できること。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。

データ伝送における機能要件を表 7.2-13 に示す。

表 7.2-13 データ伝送の機能要件

項番	個別機能	説明
1	プロトコル変換	地域に展開するスマートシティアセットや他システムと接続するため、一般的な通信プロトコルから都市 OS が対応する通信プロトコルに変換できること。
2	分野間データ検索	都市 OS 外に分散されたデータを、データの概要情報（カタログ）を基に検索できること。 ※将来、分野間データ連携コネクタとの接続に活用される。
3	分野間データ交換制御	都市 OS、他システムの双方の取り決めによりデータの利用権限を判断し、データのアクセス範囲を制御できること。 ※将来、分野間データ連携コネクタとの接続に活用される。
4	分野間データ交換記録	トレーサビリティによるデータ品質向上のため、都市 OS と他システムの双方で連携したデータの交換履歴を記録できること。 ※将来、分野間データ連携コネクタとの接続に活用される。

7.2.7 セキュリティ

セキュリティは、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。

7.2.7.1 セキュリティの機能ブロック説明

都市 OS のセキュリティ対策としては、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）の二つに大分される。

技術的対策には、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として定義される。

管理的対策には、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が含まれ、都市 OS の運用・管理に必要な事項として定義される。

7.2.7.2 セキュリティの機能要件

技術的対策として、都市 OS は表 7.2-14 に示す個別機能を具備する。

表 7.2-14 技術的対策の機能要件

項番	個別機能	説明
1	認証	都市 OS に接続する利用者、スマートシティサービス、他都市 OS、他システム、IoT デバイス等に対して正しい接続相手であるかを検証し、アクセス権限を与える機能を提供すること。なお、本機能は、以下の章で定義しているものと同等である。 ・「7.2.2 認証」 利用者、スマートシティサービス、他都市 OS に対する認証 ・「7.2.5 アセットマネジメント」 スマートシティアセットに対するデバイス認証
2	暗号化	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こと。
3	不正アクセス防止	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を提供すること。ファイアウォール機能とも呼ぶ。

4	不正アクセス検知／ 遮断機能	不正アクセス防止機能では対応できない、DoS 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、遮断する機能を提供すること。
---	-------------------	--

それぞれのセキュリティ機能と都市 OS の各種機能ブロックとの対応関係を図 7.2-3 に示す。

セキュリティ機能 機能ブロック		認証	暗号化	不正アクセス 防止	不正アクセス 検知/遮断
機能	サービス連携	—	○ (通信の暗号化)	○	○
	認証	○ (認証にて定義)	—	—	—
	サービス マネジメント	—	—	—	—
データ	データ マネジメント	—	○ (データの暗号化)	—	—
データ 連携	アセット マネジメント	○ (アセットに対する 認証)	—	—	—
	外部データ 連携	—	○ (通信の暗号化)	○	○

図 7.2-3 セキュリティ機能と都市 OS の各機能ブロックとの関係

また、管理的対策として、都市 OS の管理・運用においては表 7.2-15 に示す要件を満たす必要がある。

表 7.2-15 管理的対策の機能要件

項番	個別機能	説明
1	脆弱性管理	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こと。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こと。
2	ログ管理	都市 OS が行う通信や処理に関するログを取得すること。取得したログは、証拠保全のために一定期間保存しておくこと。

また、都市 OS はクラウド上のプラットフォームとして実現されるため、考慮すべき一般的事項として総務省が公開する「クラウドサービス提供における情報セキュリティ対策ガイドライン（第 2 版）」⁴¹も併せて参照し、技術的対策、管理的対策ともに実施することが望まれる。

⁴¹ クラウドサービス提供における情報セキュリティ対策ガイドライン（第 2 版）
https://www.soumu.go.jp/main_content/000566969.pdf

7.2.8 運用

運用は、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。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。

7.2.8.1 運用の機能ブロック説明

(1) システム管理

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非機能である。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。

(2) 管理プロセス

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を規定する。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。

7.2.8.2 運用の機能・非機能要件

システム管理における機能要件を表 7.2-16 に示す。

表 7.2-16 システム管理の機能要件

項番	個別機能	説明
1	拡張容易性	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仕掛けを提供すること。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	可用性	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こと。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、利用者への影響を最小化することが重要となる。

管理プロセスにおける要件を表 7.2-17 に示す。

表 7.2-17 管理プロセスの要件

項番	個別プロセス	説明
1	都市 OS 企画・開発管理	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・開発を行うこと。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	サービス移行管理	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・管理を行うこと。
3	システム運用管理	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こと。

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」⁴²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。

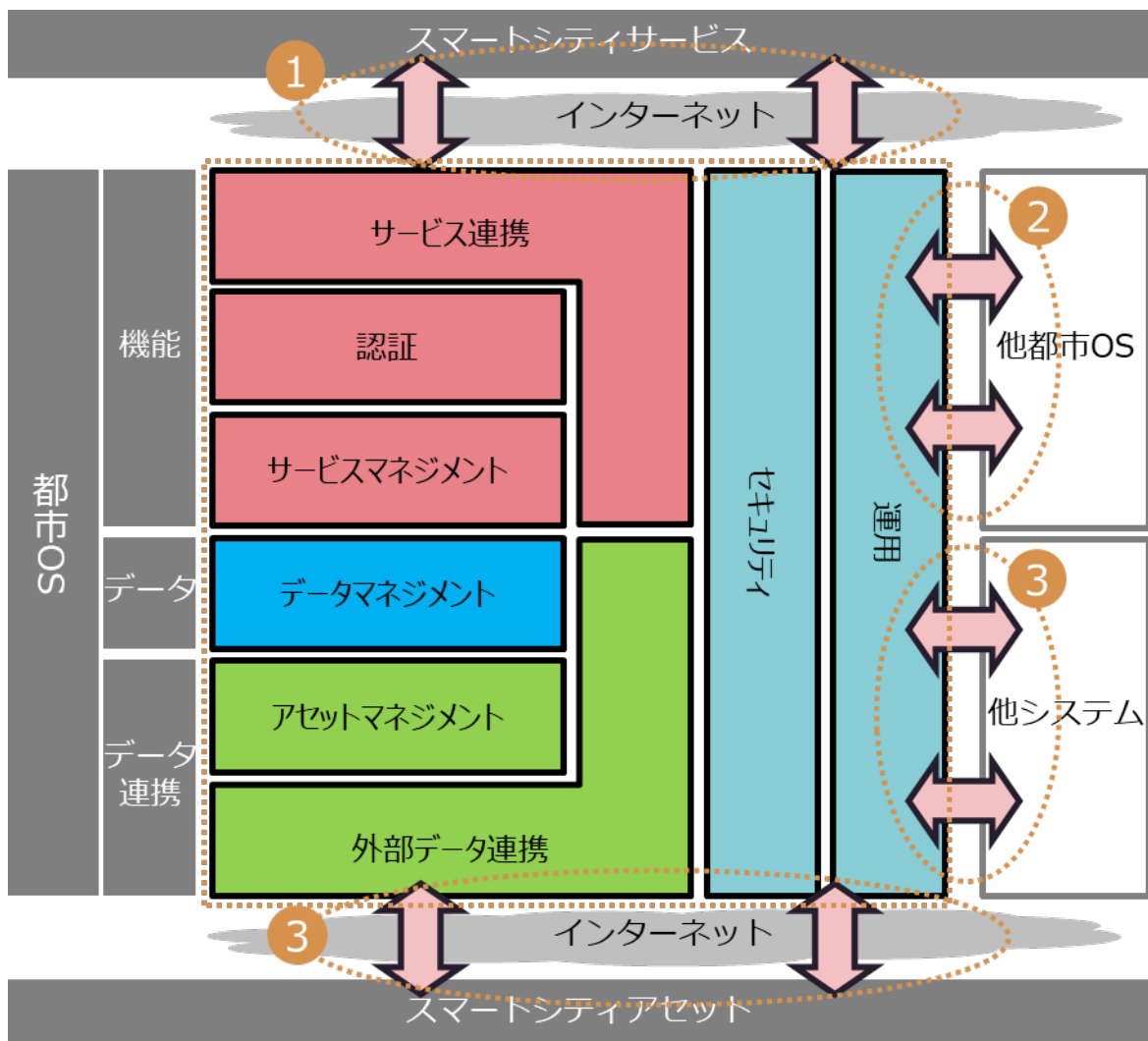
⁴² 「非機能要求グレード」 <https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>

7.3 外部連携

7.3.1 都市 OS の外部連携と API の考え方

7.3.1.1 外部連携方式

都市 OS の外部連携は、図 7.3-1 に示す通り、①サービス連携、②都市 OS 間連携、③アセット/他システム連携をサポートする。連携方法としては、API による連携、及び、機能としての連携がある。スマートシティアセットと他システムについては、「8 スマートシティアセットと他システム」を参照いただきたい。



項番	連携方式	説明	連携ポイント
1	サービス連携	都市 OS 上で動作するスマートシティサービスに対し各種機能を提供するインタフェースや共通サービスを公開	サービス間連携に必要なデータ・通信方式・認証と API の定義
2	都市 OS 間連携	他の都市 OS が公開するスマートシティサービスやデータとの連携	他都市 OS との連携に必要なデータ・通信方式・認証及び API の定義
3	アセット連携/ 他システム連携	多様なスマートシティアセットや、他システムに対するデータの収集や仲介	スマートシティアセット及び他システムとの連携に必要なデータ・通信方式とインタフェースの定義

図 7.3-1 都市 OS における連携ポイント

都市 OS がサービス連携・都市 OS 間連携・アセット連携/他システム連携を可能とするため、API 及びデータの相互運用が必要になる。海外の相互運用を参考に、都市 OS の相互運用を示す。

7.3.1.2 海外の相互運用

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。

(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⁴³。行政機関のために作られたフレームワークであるが、スマートシティにも参考となる概念である。

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。

⁴³ European Interoperability Framework (EIF)

https://ec.europa.eu/isa2/sites/isa/files/eif_brochure_final.pdf



図 7.3-2 European Interoperability Framework (EIF)

(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⁴⁴。

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 7.3-1 の3レベルに加え、Helsinki 提案の Personal Data Management, Amsterdam 提案の Fair AI が MIMs に加わるようになった。

表 7.3-1 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	Marketplace API	It exposes functionalities such as catalogue management,	Basic Data Marketplace Enablers(SC-D2.4)	TM Forum API

⁴⁴ <https://www.youtube.com/watch?v=Dkq8X0K-iwY>

Transactions Management MIM		ordering management, revenue management, Service Level Agreements (SLA), License management etc. Complemented by marketplaces for hardware and services.		
-----------------------------	--	---	--	--

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・都市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 7.3-2 に SynchroniCity における相互運用ポイントの API⁴⁵について記載する。

表 7.3-2 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。

⁴⁵ <https://synchronicityiot.docs.apiary.io/>

図 7.3-3 に共通データモデルの例⁴⁶を示す。

Vertical	Data Model	Description	Original Source	Approval Status
Environment	AirQualityObserved	It represents an observation of air quality conditions at a certain place and time	GSMA	Approved
Environment	NoiseLevelObserved	It represents an observation of those parameters that estimate noise pressure levels at a certain place and time	FIWARE	Approved
PointOfInterest	PointOfInterest	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

Air Quality Observed

Note: The latest version of this Data Model can be found at <https://github.com/synchronicity-iot/synchronicity-data-models/blob/master/Environment>

Description

An observation of air quality conditions at a certain place and time. This data model has been developed with mobile operators and the GSMA.

Data Model

A JSON Schema corresponding to this data model can be found here:

- ID: Unique identifier.
- Type: Entity type. It must be equal to: `AirQualityObserved`.
- Description: (Specify the URI to information about the provider of this information)
 - Attribute type: Property, URI.
 - Optional
- Last updated: (Last update timestamp of this entity)
 - Attribute type: Property, `LastTime`.
 - Read Only: Automatically generated.
- Data Model: Entity's relation hierarchy.

Examples

Normalized Example

Normalized NGSI response

```

{
  "id": "http://synchronicity-iot.com/2018-05-01/2018-05-01",
  "type": "AirQualityObserved",
  "dataSource": {
    "value": "http://2018-05-01/2018-05-01/2018-05-01"
  },
  "location": {
    "value": "London"
  },
  "temperature": {
    "value": 22.5
  },
  "humidity": {
    "value": 70
  },
  "pressure": {
    "value": 1013
  },
  "lastUpdated": {
    "value": "2018-05-01T12:00:00Z"
  }
}

```

LD Example

Sample uses the NGSI-LD representation

```

{
  "id": "http://synchronicity-iot.com/2018-05-01/2018-05-01/2018-05-01",
  "type": "AirQualityObserved",
  "dataSource": {
    "value": "http://2018-05-01/2018-05-01/2018-05-01"
  },
  "location": {
    "value": "London"
  },
  "temperature": {
    "value": 22.5
  },
  "humidity": {
    "value": 70
  },
  "pressure": {
    "value": 1013
  },
  "lastUpdated": {
    "value": "2018-05-01T12:00:00Z"
  }
}

```

図 7.3-3 SynchroniCity におけるデータモデルの標準化例

(3) Pivotal Points of Interoperability (PPI)⁴⁷

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 7.3-4 に示すように策定している。

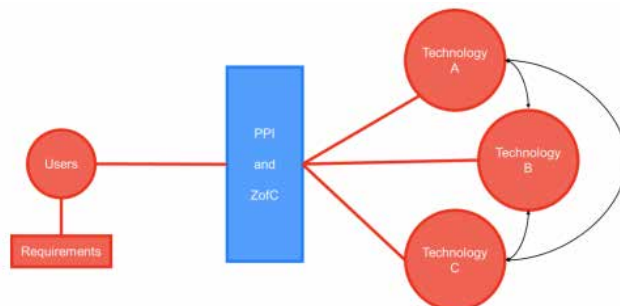


Figure 1: IES-City Framework Structure

図 7.3-4 IES-City Framework⁴⁸

⁴⁶ <https://gitlab.com/synchronicity-iot/synchronicity-data-models>

⁴⁷ NIST A Consensus Framework for Smart City Architectures IES-City Framework
https://s3.amazonaws.com/nist-sgcps/smartcityframework/files/ies-city_framework/IES-CityFramework_Version_1_0_20180930.pdf

⁴⁸ https://s3.amazonaws.com/nist-sgcps/smartcityframework/files/ies-city_framework/IES-CityFramework_Version_1_0_20180930.pdf P7

このような比較検討を通じて、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 7.3-5 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 REST API 等が挙げられている。

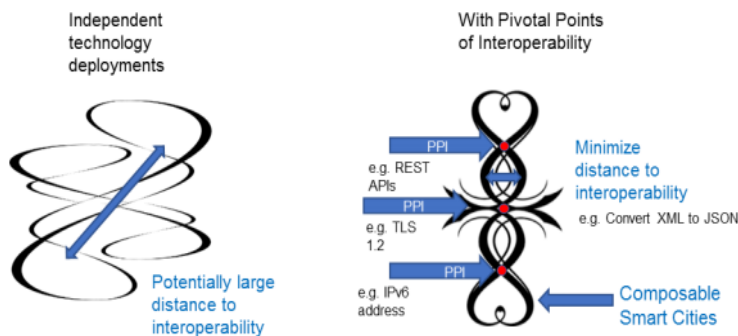


Figure 16: PPI Reduces Distance to Interoperability

図 7.3-5 Pivotal Points of Interoperability⁴⁹

7.3.1.3 都市 OS の相互運用

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 7.3-6 に示す通りに定める。

⁴⁹ https://s3.amazonaws.com/nist-sgcps/smartcityframework/files/ies-city_framework/IES-CityFramework_Version_1_0_20180930.pdf P71

Inter Operability	構成要素	選択肢（例）
Semantic	語彙体系 (型、コード等)	• 共通語彙基盤 • データ・カタログ語彙(DCAT) • Schema.org, RDFS 等
	データ項目	• 政府CIOポータル • FIWARE/SynchroniCity • Open311, GSMA, DATEX II 等
	データ構造	• Schema.org • NGSI/NGSI-LD • RDF+OWL 等
	API仕様	• OAuth2.0/OpenIDConnect • NGSI/NGSI-LD • SPARQL, OData, SQL 等
	APIモデル	• REST/RESTful • GraphQL 等
	データ形式	• JSON/JSON-LD, XML, CSV • Database(RDB, NoSQL) 等
Technical	通信プロトコル	• HTTP/HTTPS • MQTT, CoAP 等
	トランスポート	• TCP, UDP
	インターネット	• IP
	ネットワーク インターフェース	• WWAN, LPWAN, WLAN

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 7.3-6 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、「7.3.2 都市 OS が提供する API 及びインタフェース」を参照いただきたい。

- Y 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。
- Y 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。
- Y 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。
- Y 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。

上記の検討方針を基に、都市 OS の実装方針を以下に記載する。

(1) 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、都市 OS が提供する API を参考に、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 を実装することになる。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。

都市 OS が提供する API は、利用者にとって使いやすい方式で提供する必要があり、WebAPI を採用する。都市 OS における WebAPI は、REST の API モデルの元、HTTPS プロトコルと標準的なデータフォーマットに基づいて、URI を指定しリソースを操作するものである。レスポンスデータは、機械判読しやすいよう JSON のデータ形式を利用する。他の技術的に考慮すべき事項

については、内閣官房情報通信技術（IT）総合戦略室 API テクニカルガイドブック⁵⁰を参照いただきたい。

データモデルに関しては、内閣官房情報通信技術（IT）総合戦略室が推進する推奨データセット⁵¹、共通語彙基盤⁵²やデジタル・ガバメント標準ガイドライン群⁵³にあるデータ標準群やコード一覧が活用できる。日付・時刻、住所、地理座標の表記方法、PoI（Point of Interest）コードや、府省が公表しているコード等が参照できる。これらのデータモデルは、Web でのデファクトスタンダードとしている schema.org⁵⁴も参照にして推進しているが、前出のデータモデルで不足するデータは schema.org を参照すること考えられる。さらに、地理空間やインフラ情報は、国土交通省が推進する国土交通データプラットフォーム⁵⁵の提示するデータモデルについても参照することが有効である。

（2）多様な主体がアクセス可能なよう外部に公開する仕掛け

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。

（a）機能の公開

- ・オープン API（接続仕様、利用方法等）
- ・オープンデータ（データ項目、データ構造等のデータモデル）
- ・メタデータ（カタログデータ、組織/人等）
- ・利用規約（ライセンス、禁止事項、免責事項等）

（b）開発・評価環境の提供

- ・開発ポータル（カタログ、コンソール）
- ・サンプルコード、ライブラリ
- ・サンドボックス

（c）情報交換の環境

⁵⁰ 出典：内閣官房情報通信技術（IT）総合戦略室 API テクニカルガイドブック

https://cio.go.jp/sites/default/files/uploads/documents/1020_api_tecnical_guidebook.pdf

⁵¹ 出典：内閣官房情報通信技術（IT）総合戦略室が推進する推奨データセット

<https://cio.go.jp/policy-opendata>

⁵² <https://imi.go.jp/goi/common/>

⁵³ <https://cio.go.jp/guides>

⁵⁴ <https://schema.org/>

⁵⁵ <https://www5.cao.go.jp/keizai-shimon/kaigi/special/reform/wg6/20191105/pdf/shiryu2.pdf>

・コミュニティ

API やデータモデルのより詳細な公開方法は、内閣官房情報通信技術（IT）総合戦略室が発行する API 導入実践ガイドブック⁵⁶を参照いただきたい。

7.3.2 都市 OS が提供する API 及びインタフェース

7.3.2.1 認証系 API

（1）接続に関する要件（API）

OAuth2.0⁵⁷と OpenID Connect⁵⁸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表 7.3-3 に定義する。また、具体的な実装例を補足資料に記載する。

表 7.3-3 認証系 API の機能要件

項番	個別機能	説明
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。事前に設定された利用者の権限に応じ、利用範囲が制限される。 ※OAuth を活用することを推奨する。
2	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。
3	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。

（2）標準規格

認証に関わる標準規格について説明する。

⁵⁶ 内閣官房情報通信技術（IT）総合戦略室が発行する API 導入実践ガイドブック
https://cio.go.jp/sites/default/files/uploads/documents/1019_api_guidebook.pdf

⁵⁷ <https://oauth.net/2/>

⁵⁸ <https://openid.net/connect/>

(a) 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)。

(b) 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。

(c) SAML⁵⁹ (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。

7.3.2.2 データマネジメント系 API

「7.1.1.2 データ流通 (ながれる)」にて示した通り、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) 接続に関する要件 (API)

表 7.3-4 データアクセス系 API の機能要件

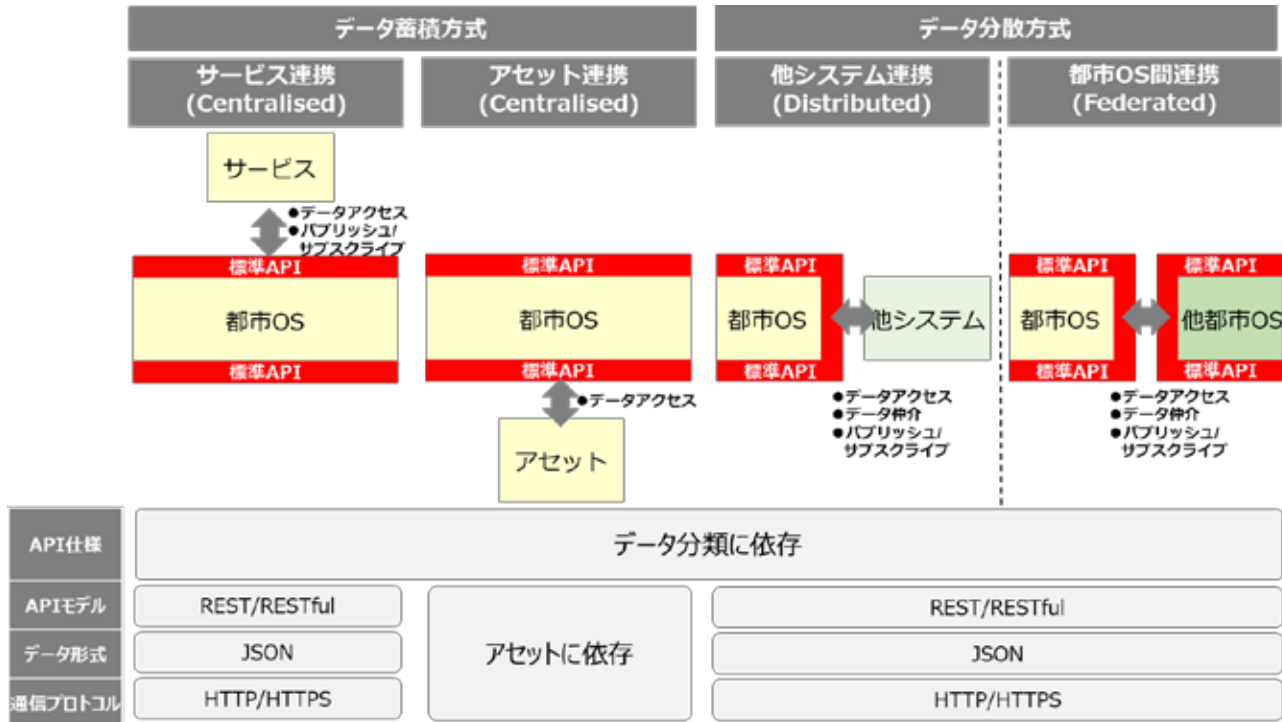
項番	個別機能	説明
1	データアクセス	都市 OS のデータマネジメントと連携し、データのライフサイクル（登録、参照、変更、削除）を管理するための API を提供できること。
2	パブリッシュ/ サブスクライブ	都市 OS が保管するデータに変更が生じた際に、リアルタイムに変更内容を通知先に送信するための API を提供できること。 また、通知内容（条件や通知先等）のライフサイクル（登録、参照、変更、削除）を管理するための API についても提供できること。
3	データ仲介	分散するデータに対し、その所在のライフサイクル（登録、参照、変更、削除）を管理するための API を提供できること。
4	パーソナルデータ （要考慮個人情報）授受	パーソナルデータ（要考慮個人情報）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。

⁵⁹ <https://www.oasis-open.org/committees/download.php/56776/sstc-saml-core-errata-2.0-wd-07.pdf>

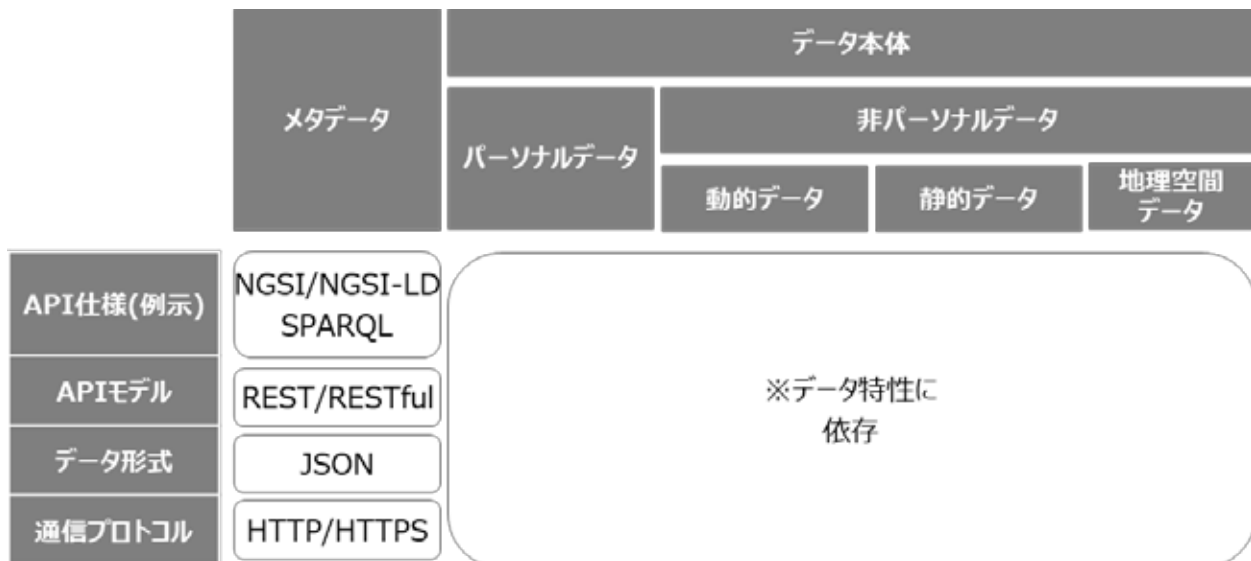
(2) 標準規格

データアクセスに関わる標準規格について説明する。連携方式別や、データ分類別に標準規格を例示するが、都市 OS が管理するデータの特性により実装は異なる。都市 OS が取り扱う API 仕様やデータモデルは、メタデータとして外部に公開し、利用者はデータの特性に合わせたアクセスをする必要がある。ユースケースによる具体的な実装例を APPENDIX に記載する。

(a) 連携方式別



(b) データ分類別



(ア) NGSI⁶⁰/NGSI-LD⁶¹

Next Generation Service Interfaces の略。Open Mobile Alliance により標準化され、以降もアップデートを重ね、最新版の NGSI-LD (Linked Data) は ETSI (欧州電気通信標準化機構) により公開されている。実世界の物理オブジェクトを、一意となる識別子や属性、関連する付加情報を含めたコンテキストとして、標準化されたデータモデルで管理する。データの所在を問い合わせるインタフェース (NGSI-9)、データ本体を問い合わせるインタフェース (NGSI-10) が FIWARE では採用されており、分野や組織を横断したデータ連携を推進している。

(イ) SPARQL⁶²

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ー等のその他のパターンを指定可能。

(ウ) REST/RESTful⁶³

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。

✓ ステートレスなクライアント/サーバプロトコル

✓ 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

✓ リソースを一意に識別する「汎用的な構文」

✓ 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。

⁶⁰ NGSI <https://www.openmobilealliance.org/release/NGSI/>

⁶¹ NGSI-LD https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.02.02_60/gs_CIM009v010202p.pdf

⁶² SPARQL <https://www.w3.org/TR/sparql11-overview/>

⁶³ Representational State Transfer (REST)
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

(エ) JSON⁶⁴

JavaScript Object Notation (JSON; ジェイソン) は、テキストベースの軽量な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ある。

(オ) HTTP/HTTPS⁶⁵

Hypertext Transfer Protocol (ハイパーテキスト・トランスファー・プロトコル、略称 HTTP) とは、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 の略称であり、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。

⁶⁴ JavaScript Object Notation (JSON) <https://tools.ietf.org/html/rfc8259>

⁶⁵ HTTP <https://www.w3.org/Protocols/>

7.3.2.3 サービス連携インタフェース

(1) 接続に関する要件

サービス連携においては、利用者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。

表 7.3-5 サービス連携 API の機能要件

項番	個別機能	説明
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 として公開する機能を提供する。
2	地域ポイント管理	利用者に紐づく地域ポイントの加算・減算・照会処理等を実行する機能を提供する。
3	オプトイン管理	都市 OS の利用者が、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する。提供する情報の種別まで管理することが望ましい。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することが望ましい。
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・検索処理を実行する。 ※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

7.3.2.4 アセット/他システム連携インタフェース

(1) 接続に関する要件

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。

表 7.3-6 スマートシティアセット/他システム連携におけるインタフェースの機能要件

項番	個別機能	説明
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、データアクセス系 API 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。4G/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。

(2) 標準規格

外部データ連携に関わる標準規格について、アセット連携と他システム連携に分けて説明する。

表 7.3-7 アセット連携/他システム連携で活用する標準規格の例

Inter Operability	構成要素	スマートシティアセット連携	他システム連携
Semantic	語彙体系 (型、コード等)	※スマートシティアセット依存	※システム依存
	データ項目		
	データ構造		
	API 仕様		
	API モデル		REST/RESTful 等
	データ形式		CSV, JSON, XML, WMS, Shape File, GeoJSON, 地理院タイル形式, XLSX(MS-EXCEL), 筆ポリゴン, NetCDF 等
Technical	通信プロトコル	HTTP/HTTPS MQTT, CoAP 等	HTTP/HTTPS,FTP/SFTP, SMTP(email),PubSub, PubSubPubPub,XMPP 等
	トランスポート	TCP, UDP	TCP, UDP
	インターネット	IP	IP
	ネットワーク インタフェース	WWAN(4G/5G) Bluetooth/BLE LPWAN(LoRa, SIGFOX) WLAN(Wi-Fi)	WWAN