

2020年度 データ連携基盤技術報告書

スーパーシティのデータ連携基盤に関する調査業務 要約版

目次

1. 調査結果：データ連携基盤の全体像	1-3
1-1. Society 5.0 リファレンスアーキテクチャとの関係性	1-3
1-2. データ連携基盤に必要な構成要素の導出（仮説設定）	1-5
1-3. 本報告書におけるデータ連携基盤の全体像	1-7
2. 調査結果（ア）データ仲介機能	2-8
2-1. ブローカー調査	2-8
2-1-1. ユースケース調査	2-8
2-1-2. リファレンスモデル・製品事例調査	2-9
2-1-3. 調査結果からの考察	2-9
2-2. ブローカー評価	2-10
2-2-1. 評価対象のブローカー選定	2-10
2-2-2. ブローカー評価結果	2-11
2-2-3. 調査結果からの考察	2-11
2-3. ブローカー要件（案）	2-13
2-3-1. 機能要件	2-14
2-3-2. 非機能要件	2-14
2-3-3. ブローカーにて取り扱うデータ種別	2-15
2-3-4. 導入時の方針	2-16
3. 調査結果（イ）①API 共通ルール/標準仕様	3-17
3-1. API 設計/開発・公開・運用プロセス	3-17
3-2. API 利用規約テンプレート	3-18
3-3. API 標準仕様案	3-19
4. 調査結果（イ）②API カタログ/開発者ポータル	4-1
4-1. 概要・目的	4-1
4-2. 仕様検討結果	4-2
4-2-1. 仕様検討結果	4-2
5. 調査結果（ウ）データモデルの方針	5-7
5-1. データモデルの役割	5-8
5-1-1. 一般論	5-8
5-1-2. データモデルと相互運用性	5-8
5-1-3. スーパーシティにおけるデータモデルの役割	5-9
5-1-4. データモデルとデータカタログ	5-9
5-2. 推奨データモデル概要	5-10
5-2-1. 推奨データモデルの整理原則	5-10
5-2-2. 検討対象・検討方法	5-10
5-2-3. 基本データの共通記述方法	5-10
5-2-4. 推奨データモデル	5-10
5-2-5. データカタログのデータモデル定義	5-10
5-3. 今後のデータモデル確立へ向けた課題	5-11

表 1-1-1 データ連携基盤と都市 OS

参考文献	名称	説明
スーパーシティ/スマートシティの相互運用性の確保等に関する検討会 最終報告書	データ連携基盤	ビルディングブロック方式を用いて構成され、公開された API を通じてデータの集積や配信を行う機能。 データ連携基盤は、スーパーシティ/スマートシティの相互運用性の確保等に関する検討会 最終報告書における、データ連携層・データ層に該当する。
スマートシティリファレンスアーキテクチャ	都市 OS	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ことを実現する IT システムの総称。 都市 OS は、スマートシティリファレンスアーキテクチャにおける、機能層・データ層・データ連携層に該当する。

各参考文献を基に、スーパーシティにおけるデータ連携基盤のレイヤーを、図 1-1-2 に整理した。各参考文献ではそれぞれのアーキテクチャ視点が異なるため一部レイヤーが交差している。例えば、「相互運用性の確保等に関する検討会」における機能層は、個々のサービスやサービス群について言及しているが、スマートシティリファレンスアーキテクチャにおける機能層は、サービスと連携するために必要な機能（API や認証等）について言及しているといった点である。

データ連携基盤は、「相互運用性の確保等に関する検討会」における、データ層、データ連携層が中心となるが、データや API を公開するにあたり、スマートシティリファレンスアーキテクチャにおける機能層も一部必要と考える。

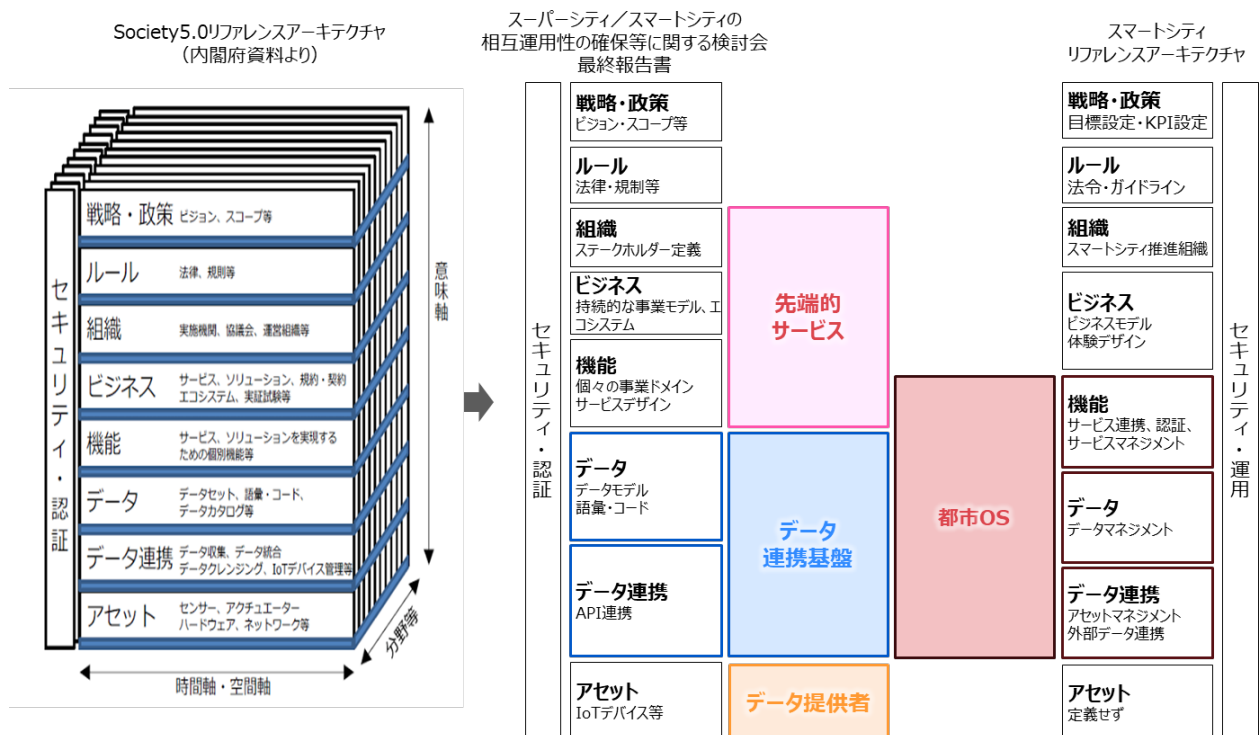


図 1-1-2 Society 5.0 リファレンスアーキテクチャとの関係性

1-2. データ連携基盤に必要となる構成要素の導出（仮説設定）

データ連携基盤に必要となる構成要素の導出にあたり、データ連携方式を整理した上で、データ連携基盤に求められる要件と、それを満たす機能について、「スーパーシティ/スマートシティの相互運用性の確保等に関する検討会 最終報告書」に記載されているサービス及び相互運用性の観点を中心に整理する。各観点で導出した構成要素については、本報告書（ア）～（ウ）の調査結果を踏まえ、1-3章にて本報告書におけるデータ連携基盤の全体像を示す。

(1) データ連携方式

データ連携基盤におけるデータ連携方式を、「スーパーシティ/スマートシティの相互運用性の確保等に関する検討会 最終報告書」に記載されているデータ仲介の分類（データ蓄積方式、データ分散方式）から整理した。

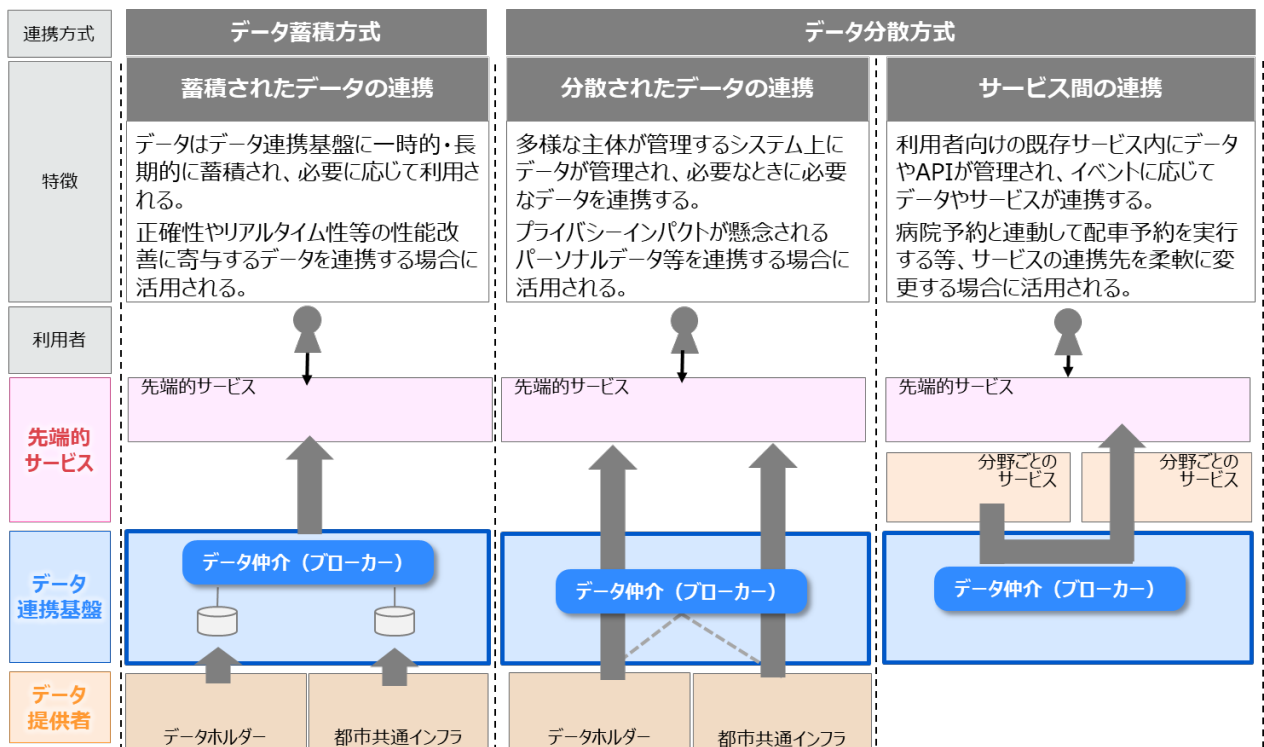


図 1-2-1 データ連携方式

図 1-2-1 で整理した各連携方式を実現するためにデータ連携基盤に必要となる事項について、相互運用性の確保及び先端的サービスの観点から導出した結果を記載する。導出過程の概要を次項(2)、(3)、(4)にて示す。導出過程の詳細及び先端的サービスの例については「データ連携基盤技術報告書」を参照頂きたい。

(2) 相互運用性の確保に必要となる事項

スーパーシティにおけるデータ連携基盤は、公開された API を通じてデータの集積や配信を行う機能を提供するものである。データ連携基盤の API は、誰もがデータ連携基盤に接続でき、かつサービス間のデータ連携により複数分野を跨った便益を提供可能とするために、開発者ポータル（API カタログ）上で公開することで相互運用性を確保する必要がある。「スーパーシティ/スマートシティの相互運用性の確保等に関する検討会 最終報告書」に記載されている相互運用性の確保に必要となる事項から、データ連携基盤に必要となる機能は以下と考える。

表 1-2-1 相互運用性の確保に必要となる機能（案）

相互運用性の確保に必要となる事項	機能
<ul style="list-style-type: none"> ブローカー機能等を用いて、様々な主体が提供するデータを集約・変換・配信する 原則としてデータ分散方式とする 	データ仲介（ブローカー）
<ul style="list-style-type: none"> API はオープン API とする 	オープン API
<ul style="list-style-type: none"> API の情報をまとめた API カタログを実装する 	開発者ポータルサイト（API カタログ）

(3) 単一自治体エリアで実装されるサービスに必要となる事項

先端的サービスの例における単一エリアのユースケースを実現する方法を検討する。単一自治体エリアで実装されるサービスの例にある、後期高齢者の通院対策を図る A 市の構想を抽出し、サービス利用時に必要なデータ連携の要件と、データ連携基盤が提供すべき機能を整理した。

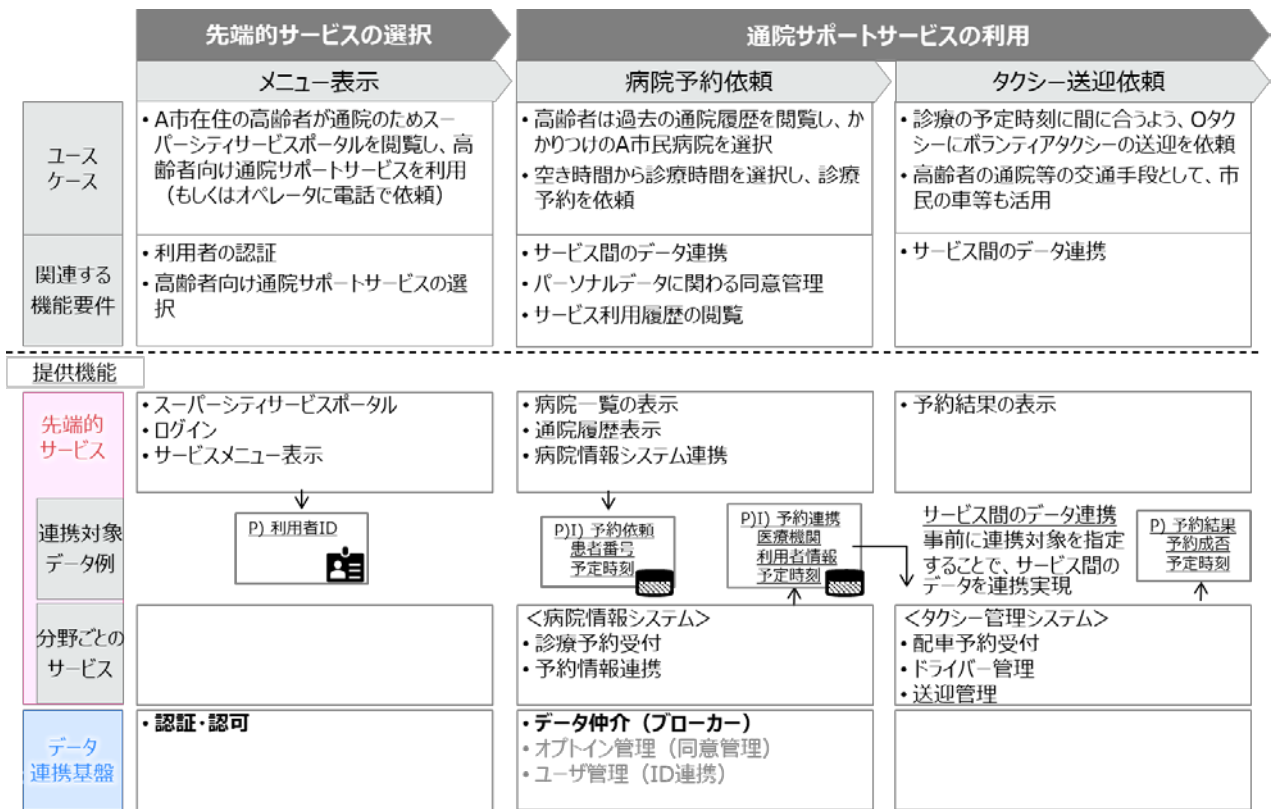


図 1-2-2 後期高齢者の通院対策を図る A 市の構想（サービス利用時）

図 1-2-2 で整理した結果より、単一自治体エリアで実装されるサービスを実現するために関連する機能要件と、データ連携基盤に必要となる機能は以下と考える。

表 1-2-2 単一自治体エリアで実装されるサービスに必要となる機能（案）

関連する機能要件	説明	機能
<ul style="list-style-type: none"> 利用者の認証 	連携する複数のサービスに対する利用	認証・認可

	者の認証を連携する	
・ サービス間の連携	複数のサービス間のデータやサービスを連携する	データ仲介（ブローカー）
・ パーソナルデータの同意管理	利用者が同意の上で、第三者にパーソナルデータの公開範囲を指定する	オプトイン管理（同意管理）

(4) 複数自治体エリアを跨って実装されるサービスに必要な事項

各都市のデータ連携基盤にて、データ仲介（ブローカー）、API 共通ルール、推奨データモデルが共通的に具備されることで、複数自治体エリアを跨って実装されるサービスにおいても、都市間のサービス連携やデータ連携が容易になると考える。

一方、連携協定等を結んだ都市同士（1:1、1:N）の連携は可能となるが、将来的に全国の都市が相互に連携し、全国を跨いだ都市間（N:N）のデータ連携・共有を実現するためには、より高度な技術要素が必要になると考える。全国を跨いだ都市間連携に必要な今後の課題について示す。

(a) 都市間連携における今後の課題

都市間連携における今後の課題として、全国を跨いだデータの連携・共有のために、特に各都市のデータ所在管理、都市間の厳格な API 仕様の管理、ユニークな ID 管理が必要になると考える。

各機能の詳細については本報告書では記載対象外ではあるが、今後検討が必要となる事項のため、執筆時点の仮説について記載する。

(ア) 各都市のデータ所在管理

分散型データ連携を実現する場合、データのアクセス先（所在）の管理が必要となるが、連携対象となる都市が増大した場合、その対象となるデータのアクセス先の肥大化や、アクセス先の更新に関わる運用が煩雑になる可能性がある。そのため、全国を跨いだ都市全体のデータ所在を管理・制御する仕組みや、直接連携していない都市同士を論理的に連携する仕組みが必要になると考える。

(イ) 都市間の共通ルール

API 共通ルールを相互に実装することで、都市間が望めば互いに接続できる状態になるが、データ連携基盤を導入する都市数が増大した場合、コストや構築時期の関係により API 共通ルールを順守できないことも考えられる。中央主導で指針となる API 仕様のサンプルの提示や、API の実装を容易にする開発ツールを提供、データモデルの継続的な整備等、今回策定した API 共通ルールを各エリアで採用するための仕組みが必要になると考える。

(ウ) ユニークな ID 管理

スマートシティリファレンスアーキテクチャでは、異なる都市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 管理をする必要があると言及されている。一方、スーパーシティにおいては、データの ID 管理に加えて、データの分散管理を前提とした都市間連携の実現には、利用者、組織、サービス、アセット（デバイス）についてもユニークな ID で管理する必要がある。これにより様々な連携対象からデータ、サービス、個人の特特定や、利用者や組織が地域をまたいだ場合でもそれぞれ関連付けすることが可能となる。

1-3. 本報告書におけるデータ連携基盤の全体像

各地域で整備されるデータ連携基盤においては、相互運用性の確保の事項で示したデータ仲介（ブロー

カー)、オープン API、開発者ポータルサイトを最低限必要な機能として整備する必要があると考える。その他の機能は、各地域が解決する課題に応じた先端的サービスのユースケースに従い、ビルディングブロック方式で必要となる機能を整備する必要がある。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 の機能を参考とし、データ連携基盤を整備する際の全体像と必要となる構成要素を図 1-3-1 に示す。各機能の説明については、スマートシティリファレンスアーキテクチャを参照頂きたい。

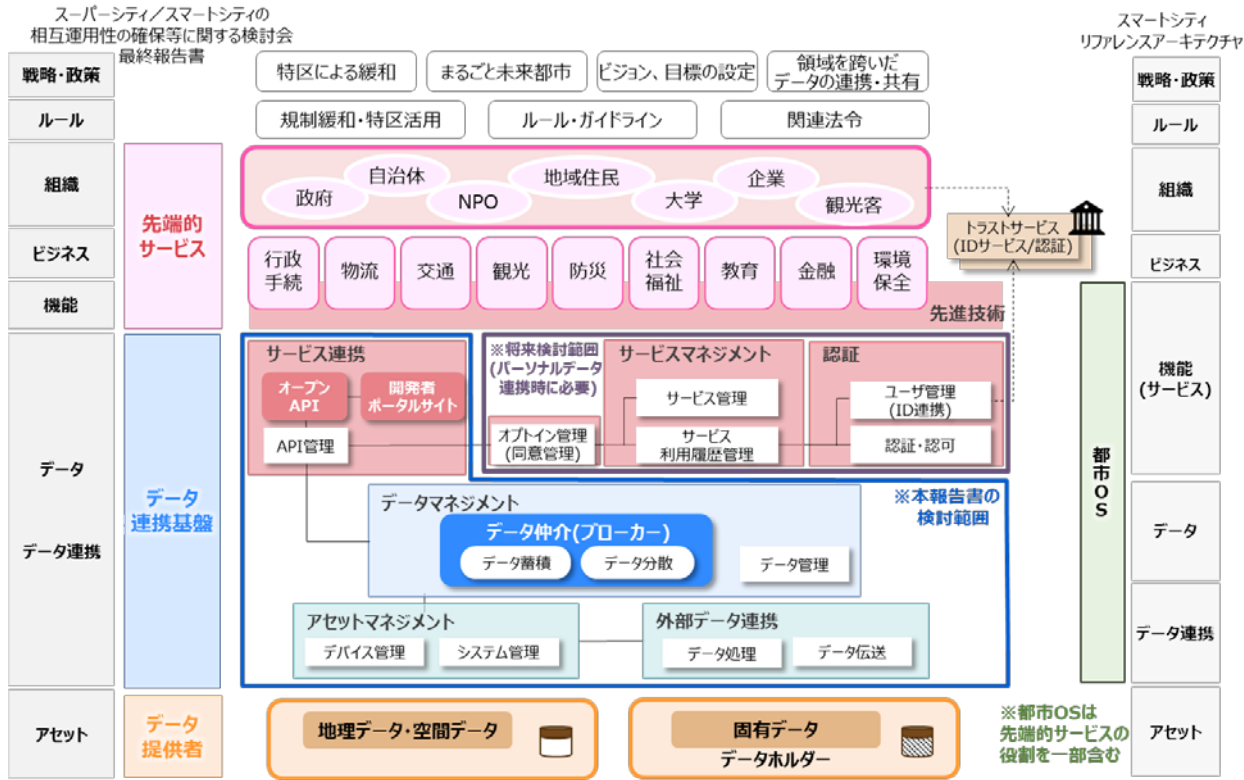


図 1-3-1 データ連携基盤の全体像

本報告書では、データ連携基盤を整備するにあたり、必須機能となる「データ仲介（ブローカー）」、「オープン API」、「開発者ポータルサイト」、について、それぞれ、2 調査結果（ア）データ仲介機能、3 調査結果（イ）①API 共通ルール/標準仕様、4 調査結果（イ）②API カタログ/開発者ポータルに仕様案を提示している。

また、パーソナルデータの連携に必要な事項（オプトイン管理、ユーザ管理等）や、都市間連携のために必要となる事項（各都市のデータ所在管理、都市間の共通ルール、ユニークな ID 管理等）については、相互運用性の確保のため、今後検討が必要になると考える。

2. 調査結果（ア）データ仲介機能

2-1. ブローカー調査

汎用的かつ共通的なブローカー仕様の整備に向け、先端的サービスのユースケース、および各種団体が公開するリファレンスモデルや製品について事例を選定し、各事例についての調査を実施する。

2-1-1. ユースケース調査

ユースケース調査では、データ連携基盤を利用するサービス側からの視点で、先端的な技術の活用、都市への普及、データ利活用の促進等の観点で、幅広くユースケースを調査し、汎用性の高いブローカー仕様案の検討、蓄積すべきデータの検討等に活用する。

調査にあたり、先端的なユースケースのうち、複数のデータを組み合わせることで実現、かつ都市への具体的な適用事例が見られる事例を選定し、各事例について、ブローカー要件（案）を検討する上で考慮すべき現状の想定課題から抽出した調査観点に沿って調査を行う。

2-1-2. リファレンスモデル・製品事例調査

リファレンスモデル・製品事例調査では、データ連携基盤自体の観点で、類似する事例からデータ連携基盤やブローカーの目指すべき姿を検討するために、各種団体が公開するリファレンスモデルや製品事例について国内外の取り組みを調査する。

調査にあたり、直近3年（2018～2020年）で新しい機能や具体的な都市への適用事例が見られ、かつデータ仲介機能に該当する技術情報が開示されているものに絞り、各事例について、ブローカー要件（案）を検討する上で考慮すべき現状の想定課題から抽出した調査観点に沿って調査を行う。

2-1-3. 調査結果からの考察

本調査ではデータ連携において、データ連携基盤からデータ利用者側を Northbound、データ連携基盤からデータ提供者側を Southbound と分けてまとめられていることより、データ連携基盤におけるデータ連携の目的が「データ利活用」と「データ収集」の2側面があり、要件が異なることがわかった。ブローカーにおいても、求められる機能が異なることより、データ連携目的を分けて整理する。「データ利活用」を目的とする場面とは、データ利用者が必要なタイミングでデータを取得するケースである。「データ収集」を目的とする場面とは、データ利用者がデータ発生時にデータを取得するケースである。

下記に調査結果に基づきブローカーとして想定課題に関する解決策を記載する。

表 2-1-1 調査結果からの想定課題解決策

No.	課題	説明	調査結果からの想定課題の解決策
1	データ分析	データを一元的に蓄積しないデータ分散型のシステムであっても、接続するシステム全体を1つのビッグデータとみなし AI 分析等を活用するためにはどのような方法があるか。	<ul style="list-style-type: none"> データ利活用 データ利用者からの参照リクエストをデータ提供者へ転送する（リクエスト転送）方法が必要。 データ収集 データ提供者が送信したデータを必要としているデータ利用者へ通知する方法（Pub/Sub）が必要。
2	汎用性	データ分散型を前提にしつつ、国際的にも汎用性が高いデータ連携基盤を作るにはどのようなモデルとすればよいか。	<ul style="list-style-type: none"> データ利活用 REST などの標準的な API の利用を推奨。 データ収集 既存アセットとの連携を考慮し、多様な接続方式をサポートすることを推奨。 データ利活用/データ収集 標準的なデータモデルや認証方式の利用を推奨
3	蓄積すべきデータ	データ連携に係る接続の正確性とスピードを重視するため、データ連携基盤に最低限蓄積すべき静的データ（地図情報等）はどのようなものがあるか。	<p>調査では取り扱うデータに関してデータ種別ごとに調査</p> <ul style="list-style-type: none"> メタデータ ブローカーではなく、データ連携基盤のデータカタログなどの機能にて管理すべき。 静的データ ブローカーではテキスト形式のデータ取り扱い可能。 データ連携基盤としては、テキスト形式、地理空間データ、バイナ

No.	課題	説明	調査結果からの想定課題の解決策
			<p>リデータを取り扱えるべき。</p> <ul style="list-style-type: none"> 動的データ ブローカーではテキスト形式のデータ取り扱い可能。 データ連携基盤としては、テキスト形式、動画データを取り扱えるべき。 パーソナルデータ データ連携基盤として取り扱えるべきであるが、ブローカーに管理機能等を具備する必要はない。取り扱う際は、リスクアセスメント及びデータのセキュリティ対策が必要。
4	既存システムとの連携	交通、行政等の既存の稼働中のシステムに接続する場合、これらのシステム及びデータ連携基盤そのものに大きな影響や負担を与えることなく正確にデータ連携するには、どのような方法があるか。	<ul style="list-style-type: none"> データ利活用 データ連携基盤と接続するシステムに対して、標準的な API を提供し、メッセージ（処理・手続き）を要求する方法が必要。 データ収集 データ発生時に利用者へ通知するイベントドリブン方式が必要。 データ利活用/データ収集 システムとデータ連携基盤のインターフェース相互変換が必要。
5	コスト	運用性・保守性等を加味した上で、ランニングコストを低減させるためにはどのような方法があるか。	ライセンス費用が無償、カスタマイズの自由度、情報の検索性などの点で OSS 利用によるコストメリットがある。ただし OSS は無保証であるため、社会実装として安定運用を実現するためには保守体制の整備又はベンダサポート等の考慮が必要。
6	スケーラビリティ	都市間連携・分野間連携の連携数が将来的に増加した場合においても、リアルタイム性等のブローカー性能を維持可能なように拡張性を保つにはどのような方法があるか。	調査結果からの解決策はなし。 ブローカー評価（2-2 章）にて検討する。
7	パーソナルデータの利活用	地域課題を解決するために、プライバシーリスクを軽減しつつ必要最小限のデータ連携のもとで最大限の価値を発揮するためにはどのような方法があるか。	パーソナルデータを取り扱う場合は、データ連携基盤にて、本人同意管理や匿名化等の機能が必要。また信頼されたサービス間のみでのセキュアなデータ交換するなど、セキュリティ対策を実施した上で利用が必要。

2-2. ブローカー評価

本項では、ブローカー要件を抽出するため、ブローカーの評価を実施する。

評価を行うブローカーを選定し、選定された各ブローカーに対し実機での評価（ユースケース評価、性能評価）を実施する。

2-2-1. 評価対象のブローカー選定

本調査では、一般的なブローカー製品を洗い出し、評価に利用するブローカーを選定するために、異なるブローカー技術に関してはブローカーの基本機能であるデータ仲介における「データ分散方式」にフォーカスし、3つのパターンに分類する。ブローカー技術の比較のため、分類した仲介パターンを網羅できるようにパターンごとにブローカー製品 1 種類を選定する。また、今回の調査事業の目的はブローカー技術の比較評価及び課題抽出/改善方針検討であるため、ある程度自由に試行錯誤可能なものが望ましいため、評価には OSS として仕様が公開されているものから選定する（ブローカーとして OSS が必ずしも

好ましいというわけではない。

表 2-2-1 データ分散方式のパターンと選定ブローカー製品

パターン	方式	説明	選定製品/OSS 名
パターン A	同期方式 (Pull 型)	データ利用者のリクエストの即座に返信する(同期方式)特徴を持つ。 データ提供者の所在を隠蔽して透過的アクセスできるものや、直接アクセスするものなど製品によって実装の違いがある。	FIWARE Orion X-Road
パターン B	非同期方式 (Pull 型)	一般的に市場に存在するブローカー製品の主流な方式(Pub/Sub 型モデル)。データ利用者とデータ提供者のリクエスト/返信のタイミングが異なる(非同期方式)特徴を持つ。 データ利用者のタイミングでデータを取得する。	Apache Kafka
パターン C	非同期方式 (Push 型)	パターン B と同様に、データ利用者とデータ提供者のリクエスト/返信のタイミングが異なる(非同期方式)特徴を持つ。 データ提供者のタイミングでデータ利用者へ通知を行う。	FIWARE Orion

2-2-2. ブローカー評価結果

今回の評価内容としては、スーパーシティ選定エリアでの実装を想定した「ユースケース評価」と、データ仲介のパターン別の「性能評価」を行い、ユースケース(機能・運用)と性能の両面から課題を抽出する。ブローカー評価の評価内容について以下にまとめる。

表 2-2-2 ブローカー評価内容

評価分類	説明
ユースケース評価	ブローカー要件(案)を検討する上で考慮すべき現状の想定課題から抽出したユースケース評価の評価観点に沿って、各ブローカー製品の機能確認を行う。
性能評価	「スケーラビリティ」の想定課題に関して、各ブローカー製品の処理性能の傾向について確認を行う。
スケーラビリティ評価	ブローカーがスケールアップ/アウトした際の性能傾向を確認する。 ①スケールアップ評価：ブローカーサーバ 1 台に対してベースとなるサーバスペックから CPU とメモリを変動させて性能測定を行う。 ②スケールアウト評価：ブローカーサーバスペックを固定し、ブローカーサーバの構成台数を変動させて性能測定を行う。
データ仲介パターン評価	以下のデータ仲介パターンにおける性能傾向を確認する。 ①データ蓄積方式：ブローカーにデータ蓄積した際の性能影響確認 ②データ分散方式 ・単一ブローカーによる単一分野連携：データ分散方式の最小構成の性能測定(限界性能) ・単一ブローカーによる単一分野連携 ※認証処理あり： ブローカー外の認証処理のオーバーヘッドを要因とした性能影響確認 ・単一ブローカーによる複数分野連携：データ連携の分野数増による影響確認 ・複数ブローカーによる単一分野連携：ブローカーの多段連携数による影響確認

2-2-3. 調査結果からの考察

下記に評価結果に基づきブローカーとして想定課題に関する総括を記載する。

(a) ユースケース評価

表 2-2-3 ユースケース評価結果考察

No.	課題	検証方針	考察及び課題
1	データ分析	各種サービスに対して物理的なデータの所在を隠蔽し、統合的なデータアクセスを実現する方法を検証する。	データ提供者の持つデータの ID と所在情報 (URL) を登録し紐付けることで複数のデータ提供者の持つデータを一元的に管理することが可能となる。ただし、ブローカーが保持する情報は原則として所在情報のみであり、ブローカー自体がデータの条件等による検索を実施することは難しい。条件付き検索等の機能はデータ提供者側の API 仕様に依存する。
3	蓄積すべきデータ	ブローカー製品でどのようなデータが取り扱うことができるかを検証する。	全てのブローカー製品がテキスト形式に加えバイナリ形式のデータ仲介が可能であった。ただしバイナリデータをテキスト形式へエンコードする手法が主であり、データサイズが大きい場合は NW 負荷やクライアント負荷が高くなることが懸念される。サイズが大きい場合はストリーム配信が可能なオンラインストレージ等への蓄積が望ましいと考えられる。
4	既存システムとの連携	連携先システム・ブローカー間の適切な機能分担を検証する。また、利用者と他システムとの仲介という観点で、利用者への通知や、連携先システムへの機能呼び出し等メッセージの仲介をイベントドリブンで実施できるかを検討する。	ブローカー製品自体には既存システムのインターフェースに合わせてプロトコルやデータ形式を変換する機能を具備していないケースが主である。ETL 等を併用することでインターフェース変換機能を実装するのが望ましい。
5	コスト	運用保守における属人化/特定ベンダ依存を極力排除した、ブローカーの運用保守作業の効率化を可能とする機能/構成となっていることを検証する。	今回検証した全てのブローカー製品がパブリッククラウド上に構築可能ことが確認できた。パブリッククラウドの機能を活用することで、運用性・保守性を向上しつつランニングコストを効率化させることができる。 一方、ブローカー製品自体の管理機能が UI で提供されるケースは限定的であり、API 提供が主流である。運用性・保守性を考慮すると UI で管理可能なダッシュボード等を実装するのが望ましい。
6	スケーラビリティ	スケールアップ/アウトによる性能向上傾向確認及び、連携する都市数・分野数、他機能が性能に与える影響範囲の検証。	スケールアップ/アウトすることで性能が向上することが確認できたが、ブローカーのように数多くアクセスを分散して受け渡す場合はスケールアウトの方がボトルネックが発生しにくく相性も良い。
7	パーソナルデータの利活用	ブローカー製品がどのようにプライバシーリスクを軽減することが可能かの検証。	ブローカー製品はデータ (メッセージ) を仲介する機能に特化しており、オプトインや匿名化のようなパーソナルデータに特化した機能は具備しておらず、別機能との連携が必須である。X-Road は事例からも機微なデータを取り扱うことが想定されているが、高度なセキュリティ技術に加え接続サービスを認可制にすることで制限された範囲でのデータ相互利用を実現している。

(b) 性能評価

(ア) スケーラビリティ評価

ブローカーのスケールアップとスケールアウトに関して、結論としては、どのブローカー技術においてもスケールアウトの方が効率的であると考えます。

今回の性能測定では、ブローカーへのリソース影響として CPU、メモリの使用率、DiskI/O を確認したが、それらがボトルネックになっていることは確認できなかった。サーバのスケールアップに関しては、拡張できるリソースが CPU やメモリに限られ、その上限についてもサーバ環境に依存する。一方、スケールアウトについては、台数を増加することによりブローカー1 台に対する処理の競合が少なくなり、スループットの向上が見込める。また、ブローカー台数を増やすことで可用性強化（冗長化）にも効果的である。

(イ) データ仲介パターン評価

今回の評価では、認証処理追加によるオーバーヘッドのボトルネックよりブローカー性能が先に来たため、認証処理を要因としたスループットの低下は確認できなかったが、認証やデータ変換の処理内容によっては全体の性能に影響する可能性もあるため、データ仲介の間に機能追加を行う場合はその影響を考慮する必要があると考える。

また、今回の評価では、データ仲介における、連携分野数の影響、及び都市間連携を想定した多段連携数による影響を評価した。連携分野数における評価については、全体のリクエスト数（データ量）は変えずに単純な分野数の増加によるブローカーへの影響を評価したが、スーパーシティで想定する連携分野数（5 分野）では問題にはならなかった。多段連携数による影響については、同期側のブローカーにおいて、段階数が増えたことにより、スループットの変化はなかったが、TAT が伸びていることを確認した。

データ仲介パターンによる各ブローカー技術への影響を以下にまとめる。

表 2-2-4 ブローカー技術ごとの仲介パターンによる影響

機能	パターン A (同期方式 Pull 型)	パターン B (非同期方式 Pull 型)	パターン C (非同期方式 Push 型)
周辺機能による影響	製品単体の機能では実現不可であるデータ変換等を他コンポーネントと組み合わせて実現する場合は影響を考慮する必要がある。	製品単体の機能では実現不可であるデータ変換等を他コンポーネントと組み合わせて実現する場合は影響を考慮する必要がある。	製品単体の機能では実現不可であるデータ変換等を他コンポーネントと組み合わせて実現する場合は影響を考慮する必要がある。
連携分野数による影響	スーパーシティで想定する連携分野数（5 分野）では問題にはならない。	スーパーシティで想定する連携分野数（5 分野）では問題にはならない。	スーパーシティで想定する連携分野数（5 分野）では問題にはならない。
多段連携数による影響	段階数が増えたことにより、スループットの変化はなかったが、TAT が伸びていることを確認した。	非同期方式 Pull 型のブローカー処理は、ブローカーの段階数による性能影響は発生しにくい処理方式である。	非同期方式 Push 型のブローカー処理は、ブローカーの段階数による性能影響は発生しにくい処理方式である。

2-3. ブローカー要件（案）

本項では、ブローカーの機能要件及び非機能要件の策定、ブローカーにて取り扱うデータ種別の整理、エリアへの導入する際の方針について述べる。

2-3-1. 機能要件

ブローカーとは、様々な主体が提供するデータを集約し、適切な処理を経た上で公開する仕組みである。データ連携の目的として、「データ利活用」と「データ収集」の2種類があり、サービス側の用途に応じて使い分ける。それぞれ求められる機能が異なるため、各機能についてデータ連携目的を分けて整理する。

上記のとおり、抽出した要件を機能一覧にまとめる（表 2-3-1）。必須/推奨については、ブローカーが具備しなければデータ連携基盤の原則であるデータ分散方式でのデータ利活用を実現できない機能を必須、他サービス等の連携で代替可能と考えられる機能を推奨とする。

表 2-3-1 機能一覧

データ連携目的	要件名	分類	説明	必須	推奨
データ利活用	データ参照	データ分散	データ参照の要求を受け、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蔽することができること		○
		データ蓄積	データ参照の要求を受け、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	イベント処理	サービス呼び出しの要求を受け、外部サービスの処理を実行し結果を返却可能なこと (例：交通サービスでタクシーを予約)		○
	API仕様	API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	○	
	データ変換	データ変換	外部サービスへの接続時、接続先サービスのインターフェースに合わせたデータ変換が可能なこと		○
データ収集	データ更新	イベント処理	データ提供者からデータを受け、必要なサービスへデータを送信できること		○
			データ送信時、リアルタイムにデータの分析・変換・加工処理等が可能なこと		○
		データ蓄積	データ提供者からデータを受け、データストア機能に蓄積可能なこと		○
	API仕様	API仕様	多種多様なアセットからのデータ収集を想定し、標準APIに限らず様々な接続方式に対応可能なこと（MQTT等）		○

必須：本機能がないとデータ分散方式でのデータ利活用が実現できないもの（コア機能）

推奨：あるのが望ましいが、他サービス等との連携で代替可能なもの

2-3-2. 非機能要件

想定課題を基に、運用・保守性、性能・拡張性、セキュリティという観点で、ブローカーの非機能要件を整理する。

表 2-3-2 非機能一覧

分類	要件名	概要	必須	推奨
運用・保守性	サポート体制	サポート体制が整備されていること 社会実装として安定運用を実現するためには保守体制の整備 又はベンダサポート等の考慮が必要	○	
	運用管理 I/F	UI での運用管理機能があること		○
	情報開示 (ドキュメント整備)	保守作業の効率化のため、構築・運用・利用に関する情報が 入手可能であること		○
性能・拡張性	性能	・ 5 分野以上の先端的サービス間のデータ連携ができること ・ 2 段以上のデータ仲介ができること (接続する他都市のブローカーを経由して他都市の先端的 サービスへアクセスする想定)		○
	リソース拡張	利用状況に応じた柔軟なリソース拡張が行えること (スケールアップ・スケールアウトなど) ※スケールアウトの実現を推奨	○	

セキュリティについて、ブローカーの要件としては定義しない。パーソナルデータを取り扱う際、同意管理・匿名化等のプライバシー保護に関する機能が必要である。ただし、データ連携基盤内のブローカーではない別の機能にて実現することが妥当である。

2-3-3. ブローカーにて取り扱うデータ種別

ブローカーにて取り扱うデータの種別を整理し、各種データがデータ分散方式又はデータ蓄積方式のどちらでの管理が適しているかを整理する。データ管理主体がデータを保持すべきという考え方で、データ種別問わず原則不可とする。ブローカー調査から、データ連携基盤において取り扱うデータのデータ種別を抽出し、ブローカー評価から、ブローカーにおいて取り扱うデータのデータ種別を選択する。

表 2-3-3 ブローカーにて取り扱うデータ種別

	メタデータ	データ本体					
	静的/動的	静的			動的		
データ形式	—	テキスト	地理空間	バイナリ	テキスト	動画 (ストリー ム)	
代表的な データ例	所有者情報	避難所情報	地形図	観光地写真	水位 センサー 情報	河川 監視画像	行動履歴
データ 分散方式	対象外 (※1)	必須	対象外 (※2)	必須	必須	対象外 (※3)	必須
データ 蓄積方式	対象外 (※1)	原則不可 (※4)	対象外 (※2)	原則不可 (※4)	原則不可 (※4)	対象外 (※3)	原則不可 (※4)

必須：スーパーシティのブローカー機能として取り扱い必須

原則不可：原則として取り扱い不可ではあるが、特定の条件下でのみ取り扱い可であるもの

対象外：ブローカー機能の対象外

(※1)：データカタログ等の専用機能で管理されるため、ブローカー機能の対象外

(※2)：GIS等の専用機能で管理されるため、ブローカー機能の対象外

- バイナリデータとしてデータファイル自体を取り扱うことは可能
- (※3) : VMS 等の専用機能で管理されるため、ブローカー機能の対象外
- (※4) : データ分散方式を原則とするが、データ連携基盤運営者が保持するデータであればデータ蓄積方式も可能とする

データ分散方式とデータ蓄積方式はトレードオフの関係になっているため、上記分析を参考に導入するエリアによって判断する。データ連携基盤は、データ分散方式のデータ管理を原則としている。また、データ蓄積方式は、蓄積したデータを管理するコストやデータ漏洩のリスクなどが増大するというデメリットもある。しかし、データ分散方式に比べデータ蓄積方式の方がレスポンスが早い、該当エリアにおいて蓄積する価値があるデータなどのメリットも考えられる。

2-3-4. 導入時の方針

あるエリアにデータ連携基盤を構築するにあたり、ブローカーを導入する際の方針について整理する。上述の要件を全て満たしたブローカーを用意することができれば良いが、現実的には製品仕様上の制約や導入又は運用の費用との兼ね合い上、困難である。ケース（用途）別の機能を整理することで、導入時のブローカーの実装すべき機能を取捨選択できるようにすることが目的である。

(1) ケースの例

データ連携基盤又はブローカーが導入される際に、解決すべき課題によって、必要な機能は異なる。以下のような3つのケースにおいて、必要な機能を明確にする。

- ケース1 : ブローカーを経由して各サービスからデータを取得する（データ利活用）
- ケース2 : ブローカーを経由して各サービスへデータを送信する（データ収集）
- ケース3 : データ連携基盤にデータを蓄積する（データ蓄積）

(2) 用途別の必要機能

ケースによって、必要となるブローカーの機能を抽出する。

表 2-3-4 用途別機能要件

データ連携目的	要件名	分類	説明	ケース別要件 (○ : 対象、— : 対象外)		
				ケース1	ケース2	ケース3
データ利活用	データ参照	データ分散	データ参照の要求を受付け、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蔽することができること	○	—	—
	データ蓄積	データ参照の要求を受付け、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	イベント処理	サービス呼び出しの要求を受付け、外部サービスの処理を実行し結果を返却可能なこと (例 : 交通サービスでタクシーを予約)	○	—	—
	API仕様	API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	○	—	○

データ連携目的	要件名	分類	説明	ケース別要件 (○：対象、－：対象外)		
				ケース1	ケース2	ケース3
			なこと			
	データ変換	データ変換	外部サービスへの接続時、接続先サービスのインターフェースに合わせたデータ変換が可能なこと	○	－	－
データ収集	データ更新	イベント処理	データ提供者からデータを受付け、必要なサービスへデータを送信できること	－	○	－
			データ送信時、リアルタイムにデータの分析・変換・加工処理等が可能なこと	－	○	－
	データ蓄積	データ提供者からデータを受付け、データストア機能に蓄積可能なこと	－	－	○	
	API仕様	API仕様	多種多様なアセットからのデータ収集を想定し、標準APIに限らず様々な接続方式に対応可能なこと（MQTT等）	－	○	○

導入するエリアによって、解決すべき課題や導入後の利用方針によって、適した組み合わせでブローカーを実装することが求められる。上記以外にも、同意管理・匿名化等のプライバシー保護に関する機能や、データを蓄積する目的や手段などは、必要可否を含めて、導入時に検討する必要がある。このように、ブローカーとしての機能面の拡張性という意味で、他機能と組み合わせられるようにビルディングブロックのアーキテクチャが望ましい。

3. 調査結果（イ）①API 共通ルール/標準仕様

3-1. API 設計/開発・公開・運用プロセス

スーパーシティに具備されるAPIの設計/開発・公開・運用のプロセスを表3-1-1に示す。各プロセスの詳細については、「データ連携基盤技術報告書」5-1. API 設計/開発・公開・運用プロセスを参照。スーパーシティでは、様々なAPIが多数のサービスに利用され、サービスやシステムがAPIで相互に接続・連携するような形態が想定されるため、APIの再利用性と互換性の確保が重要である。今後スーパーシティの整備に向けて、多数のAPIが新規に開発・運用されることを想定し、設計/開発の段階からその望ましいプロセスを示すことで、APIの品質の均質化と安定したAPIの提供を推進することが本節の目的である。

表 3-1-1 API の設計/開発・公開・運用プロセス

大プロセス	中プロセス	小プロセス
1. 設計/開発プロセス	(1) 計画：既存APIの有無を調査し、関係者と連携しながら開発方針を策定する。	(a) 既存APIの調査
		(b) ステークホルダとの連携
		(c) 開発手法の選択
		(d) APIの整備をサポートするツールの導入検討
	(2) 要件定義：サービス機能に合わせて必要となるAPI機能を定義する。	(a) 機能要件の検討
		(b) 非機能要件の検討
	(3) 設計：OpenAPI仕様に準拠したインタ	(a) インターフェースとデータモデルの設計

大プロセス	中プロセス	小プロセス
	一フェースとデータ項目を設計する。	(b)セキュリティの設計
		(c)モック API サーバによる OpenAPI 仕様のテスト
	(4)開発:策定した仕様に準拠するインターフェースとビジネスロジックを開発する。	(a)OpenAPI 仕様からの雛形プログラム生成
		(b)ビジネスロジックの実装
	(5)テスト:サービス向けの公開を前提とした観点で成果物をテストする。	(a)テスト計画
(b)OpenAPI 仕様の準拠性の観点		
(c)API 機能の観点		
(d)非機能の観点		
2. 公開プロセス	(1)API の公開と廃止:開発した API の提供を開始又は終了する。	(a)API のデプロイと CI/CD の活用
		(b)API のバージョン管理
		(c)API の廃止
	(2)ドキュメントの公開:OpenAPI 仕様ドキュメントと利用規約を公開する。	(a)API ドキュメントの公開
(b)API 利用規約の公開		
3. 運用プロセス	(1)監視とモニタリング:API ログ、稼働状況、性能、セキュリティの観点で監視する。	(a)API ログの収集と分析
		(b)稼働状況と性能のモニタリング
		(c)セキュリティ監視
	(2)障害対応:障害の検知、原因調査、復旧対応を行う。	(a)障害検知
		(b)原因調査
		(c)復旧作業
	(3)サポート対応:API 利用者からの Q&A 対応や、フィードバックを受けて改善を検討する。	(a)サポート窓口による対応
		(b)開発者コミュニティによる対応
		(c)フィードバックの収集・対応

3-2. API 利用規約テンプレート

本節では、API 利用規約のテンプレートについて述べる。想定読者は API の提供により先端的区域データ活用事業活動を支援するデータ連携基盤整備事業者である。

API 利用規約テンプレートが対象とする API と、API を提供するデータ連携基盤整備事業者、API を利用する先端的サービス事業者やデータ連携基盤整備事業者との関係を下図に示す。

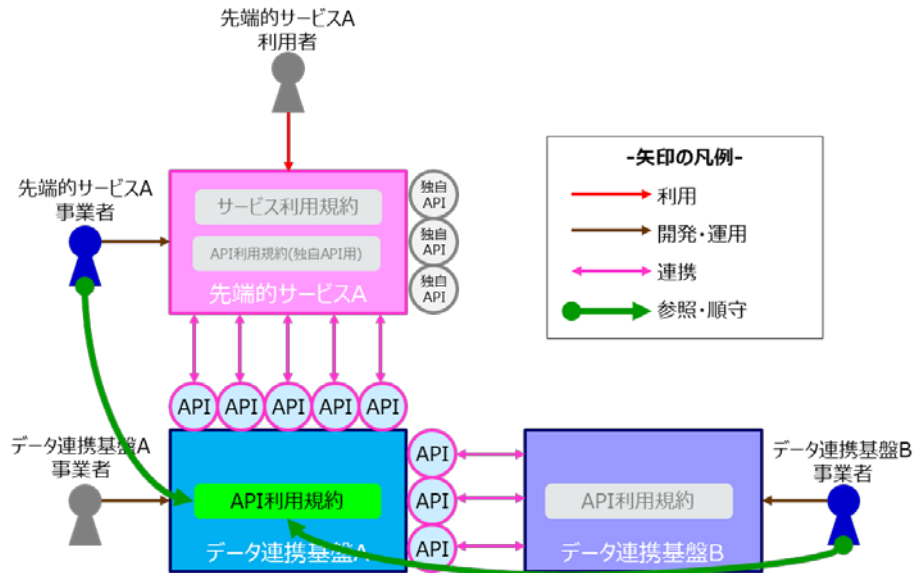


図 3-2-1 API 利用規約テンプレートの位置付け

API 利用規約テンプレートは、データ連携基盤整備事業者から、データ連携基盤が提供する API を利用して先端的サービスを開発する先端的サービス事業者や、API を利用して新たな API を開発する別のデータ連携基盤整備事業者に提供される API 利用規約を対象としたものであり、先端的サービス事業者が開発する先端的サービスの利用者に向けた利用規約や、先端的サービス事業者が独自に提供する API の利用規約については本書の対象外とする。また、API 利用規約テンプレートは、特定の単一 API に適用されるものでなく、複数の API に対して一括して適用されることを想定している。

API 利用規約テンプレートの各条文タイトルを以下の四角枠の中に記載した。API 利用規約テンプレートの条文サンプルと解説については「データ連携基盤技術報告書」5-2. API 利用規約テンプレートを参照。

第1条 目的	第8条 禁止事項	第15条 個人情報の取扱
第2条 定義	第9条 利用解除	第16条 規約の変更
第3条 登録	第10条 権利の帰属・利用	第17条 提供の終了
第4条 API 認証情報の発行及び管理	第11条 免責	第18条 連絡/通知
第5条 本 API の提供条件	第12条 補償/賠償	第19条 権利義務等の譲渡禁止
第6条 料金	第13条 反社会的勢力の排除	第20条 分離可能性
第7条 API 利用者の義務・責任	第14条 秘密保持	第21条 準拠法及び管轄裁判所

3-3. API 標準仕様案

本節では、API 提供に際し共通化すべき標準仕様案を表 3-3-1 に取りまとめた。API の相互運用性を確保する上で重要なもの、セキュリティ観点で特に遵守すべきと考えられるもの、調査した複数のガイドラインで定められており一般的なルールと認められるものについて **(必須)** とし、それらより重要度が低く、API を整備するプラットフォームや開発フレームワーク等の制約によって採用可否が変わり得るものについて **(推奨)** とした。

各要件の詳細については、「データ連携基盤技術報告書」5-3. API 標準仕様案を参照。

表 3-3-1 API 標準仕様案

大分類	小分類	API が準拠すべき標準的な要件
1. 基本となる考え方	(1)API の通信プロトコル	(必須) API と API 利用者 (サービス) を接続する通信路を暗号化する。
		(推奨) API と API 利用者 (サービス) を接続するプロトコルを HTTPS とする。
	(2)API を呼び出すインターフェース	(必須) API のインターフェースを OpenAPI 仕様に準拠した設計とする。
		(必須) OpenAPI 仕様のファイル名は openapi. json もしくは openapi. yaml ファイルとする。 (推奨) API のインターフェースは、Pragmatic REST に基づく設計とする。
	(3)API で取り扱うデータモデルとフォーマット	(推奨) API が取り扱うデータモデルとして、業界標準の形式を採用する。
		(推奨) API が取り扱うリクエスト/レスポンスボディフォーマットを JSON 形式とする。
		(推奨) 属性値のフォーマットを、国際標準やデファクトスタンダード等に従って設定する。
	(4)URL リソース名と属性名の命名規則	(必須) データモデルと属性名として、半角英数字、ハイフン、及びアンダースコアのみを使用する。
		(推奨) API で取り扱うデータモデル名と属性名として、標準的なアメリカ英語の名詞を使用する。
		(推奨) API で取り扱うデータモデル名と属性名として、スネークケースを使用する。
		(推奨) URL 中のリソース名を複数形で記載する。
		(推奨) プログラミング言語でよく利用されるキーワードとの競合を避けた名前を使用する。
		(推奨) 一般的によく知られる名称については略語を使用する。
		(推奨) 属性名には前置詞を使わない短くシンプルな名前を使用し、形容詞を名詞の前に記載する。
		(必須) 属性値が数値となる場合は、属性名に単位を含める。
		(必須) 範囲を表す属性名は、属性名の先頭を「start」又は「end」とする。
(必須) 属性値が配列で表現される場合、その属性名を複数形とする。		
(必須) 商品数などを表す属性名には「count」を含める。		
2. API リクエストの標準	(1) リクエスト URL	(必須) URL を RFC 3986 に準拠し、RFC 6570 に示される URI Template 記法で表現できる形式とする。
		(必須) API のバージョン (v1、v2 等) を URL に含める。
		(推奨) API のバージョンを上げた場合に下位互換性を保つ設計とする。
		(推奨) 地域が用意する全ての API でドメイン名を統一する。
		(必須) URL に API を表す「api」を含める。
		(推奨) サブドメイン名に API 提供を表す「api」を含める。
		(必須) API の検証利用目的のための環境を用意する。
		(推奨) API の検証利用目的のためのサブドメイン名を「api-sandbox」とする。
		(必須) 地域の開発者ポータル URL には「developer」を含める。
		(推奨) 地域の開発者ポータルのサブドメイン名を「developer」とする。
		(必須) 取り扱うデータ種が異なる場合は URL を分割する。
		(必須) 計算や翻訳などの処理を実行する API の URL を名詞ではなく動詞とする。

大分類	小分類	API が準拠すべき標準的な要件
		(必須) データ種の階層構造を表現する URL パスを設計する。
		(必須) データ種の ID にはデータベースのシーケンス番号を指定しない。
		(必須) データ種の ID には URL で使用可能な ASCII 文字を使用する。
		(推奨) UUID などデータ種と関連しない値を ID として使用する。
		(必須) URL 末尾のスラッシュ有無で API の動作を変更しない。
		(必須) URL には機密情報を含まない設計とする。
	(2) クエリパラメータ	(必須) クエリパラメータを指定しなくても API を利用できる設計とする。
		(推奨) クエリパラメータ名は、Web API で標準的に利用されるものを採用する。
		(必須) クエリパラメータ名は半角英数字で始め、全て小文字で記載する。
		(必須) 1 つのクエリパラメータに複数の値を指定する際には、カンマ「,」を用いる。
		(必須) 複数の値を指定するクエリパラメータ名を複数形、それ以外を単数形で記載する。
		(推奨) GET リクエストにおいて、返却データが 10 件以上になる場合、「limit」もしくは「since」「until」パラメータを指定し、返却件数を制限する。
		(推奨) レスポンスデータ項目として 10 件以上の項目を含む場合、「fields」パラメータを指定し、要求する項目のみ返却する。
		(推奨) リクエストデータに外字を含まない。
		(必須) API のデータ操作内容により、HTTP メソッドを使い分ける。
	(3) リクエストメソッド	(必須) API のデータ操作内容により、HTTP メソッドを使い分ける。
	(4) リクエストヘッダ	(推奨) 必要に応じて標準的なリクエストヘッダを設定する。
	(5) リクエストボディ	(推奨) リクエストボディを設定する場合は、JSON 形式を採用する。
(推奨) テキストデータの文字コードは UTF-8 を使用する。		
3. API レスポンスの標準	(1) ステータスコード	(必須) API の実行成否に関わらず、その結果を表す適切なステータスコードを返却する。
		(必須) GET リクエストが成功した場合、通常は 200 OK を返す。
		(必須) GET リクエストで 0 個のリソースを取得した場合は 200 OK を返す。
		(必須) GET リクエストでリソースの取得ができなかった場合は 404 Not Found を返す。
		(必須) POST リクエストが新しいリソースの作成に成功した場合、201 Created を返す。
		(必須) PUT リクエストが既存リソースの更新に成功した場合、200 OK 又は 204 No Content を返す。
		(必須) PATCH リクエストが既存リソースの更新に成功した場合、200 OK 又は 204 No Content を返す。
		(必須) DELETE リクエストが対象リソースの削除に成功した場合、204 No Content を返す。
		(必須) DELETE リクエストの対象リソースが削除済みの場合、204 No Content を返す。

大分類	小分類	API が準拠すべき標準的な要件
	(2) レスポンスヘッダ	(推奨) 必要に応じて標準的なレスポンスヘッダを設定する。
	(3) レスポンスボディ	(推奨) レスポンスボディを返却する場合は、JSON 形式を採用する。
		(推奨) テキストデータの文字コードには UTF-8 を使用する。
		(必須) API のレスポンスボディに外字を使用しない。
		(必須) 日本語 (全角文字) については『JIS X 0213』に定義される JIS 第 1 水準～JIS 第 4 水準を利用する。
		(推奨) API に返却するデータについて、必要に応じて文字コードの変換、文字の縮退を行う。
		(必須) レスポンスデータに複数行テキストが含まれる場合、改行コードには LF を使用する。
		(必須) JSON で 0 個のリソースを返却する場合、その属性値を NULL ではなく空配列 ([]) とする。
		(必須) JSON の属性値にテキストを保持する場合、タブや改行などの特殊文字をエスケープする。
		(推奨) 処理ステータス、処理ログ、ドキュメント、パラメータ、データ件数、実行結果データ、画面遷移などのメタデータを API 実行結果に含める。
(推奨) エラーレスポンス時として、エラー内容を示すレスポンスボディを返却する。		
4. 認証方式の標準	(分類なし)	(推奨) API のクライアント認証では API Key を使用した認証方法を選択する。
	(分類なし)	(推奨) API のユーザ認証では OpenID Connect を使用する。
	(分類なし)	(必須) 個人情報など個人同意が必要となる情報の流通を行う場合はオプトイン管理を実装する。
5. システム構成	(分類なし)	(推奨) API ゲートウェイを利用し、API の共通機能を取りまとめて提供する。

本節で取りまとめた API 標準仕様案は OpenAPI 仕様 3 系をベースとしているが、近年では下表に挙げるような通信規格/仕様も用途に合わせて採用されている。今後は、これらの通信規格/仕様の観点から本 API 標準仕様案の拡張を検討し、様々な先端的サービスの可能性を広げていくことが考えられる。また、OpenAPI 仕様も継続的にバージョンアップされているため、本 API 標準仕様は定期的に更新していくことが望まれる。

表 3-3-2 今後 API 標準仕様で検討すべき通信規格/仕様

通信規格/仕様	概要
gRPC ²	gRPC は、Google 社が開発を開始したリモートプロシージャコールシステムで、独自のインターフェース記述言語を利用して様々なプログラミング言語に対応したリモートプロシージャコールの組み合わせを実現する。HTTP/2 プロト

² <https://grpc.io/>

通信規格/仕様	概要
	<p>コールを利用しているためストリーム型の通信をサポートし、マイクロサービスアーキテクチャにおける接続サービスや、多人数参加型のチャットサービスにおけるインターフェースに利用することができる。</p>
GraphQL ³	<p>GraphQL は、Facebook 社が中心となって開発された API 向けのクエリ言語及びその実装である。クライアントアプリケーションが呼び出すエンドポイントが単一でありながら、様々なクエリを投げることでバックエンドAPI から必要なデータ項目を取り出して必要なデータの形式に変換してフロントに返すことができる仕組みを提供する。クライアントアプリケーションが自らの要望に合わせて得られるデータ項目を決定できるというメリットがある。</p>
Webhook	<p>コールバック URL を設定することで Web ページ又は Web アプリケーションの動作を拡張又は変更する方法のこと。特に標準などはないが 2007 年に Jeff Lindsay によって言及されたプログラミング用語⁴である。</p>
WebRTC ⁵	<p>WebRTC は、Web Real-Time Communication の略で、クライアントアプリケーションにシンプルな API 経由でリアルタイム通信を提供するオープンソースのプロジェクトである。ブラウザ間で直接ピア・ツー・ピア通信を行い、プラグインやネイティブアプリケーションのインストールなしで、ボイスチャット、ビデオチャット、ファイル共有などを実装できる。</p>
AsyncAPI ⁶	<p>AsyncAPI は、WebSocket や MQTT といった非同期プロトコルをサポートする API 仕様である。AsyncAPI Initiative によって仕様案策定が進められている。OpenAPI 仕様を拡張した形で仕様案策定が進められており、OpenAPI Initiative との交流も活発になされている。</p>

³ <https://graphql.org/>

⁴ <https://web.archive.org/web/20180630220036/http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/>

⁵ <https://webrtc.org/>

⁶ <https://www.asyncapi.com/>

4. 調査結果（イ）②API カタログ/開発者ポータル

4-1. 概要・目的

データ連携基盤が実際に構築され利活用される際、実装されている機能の利用方法や各種基準ドキュメント、API の利用方法、活用事例等を公開するための場が必要となる。

API の情報公開や開発者支援するための情報提供を行うポータルサイトの必要性については、既に他の事業の中でも検討がなされている。例えば、API テクニカルガイドブック⁷では、「API の利用を促進するためには、提供している API が広く開発者等に認知され、開発を支援するドキュメントやテスト環境等が用意されていることが重要」といった記載が存在し、公開すべき項目や告知方法等が検討されている。

また、スーパーシティ構想⁸では「スーパーシティでは各取組で実装される API に関して、情報の見つけやすさを向上させ、公開されている様々な API への接続をより容易とするために、API に関するメタデータやデベロッパーサイトの情報をまとめたカタログサイトの実装を行います。」という記載がなされており、各エリアで実装される API についてはカタログサイト上で集約・公開するものであると定義されている。さらに、各取り組みで実装される API の収集・公開だけでなく、「自団体のウェブサイト、あるいは信頼性の高いソースコードリポジトリ等、利用する技術者にわかりやすい場所においてデベロッパーサイト（開発者サイト）を作成し、そこで API に関する情報を、技術者にわかりやすい場所と形式で公開しなくてはなりません。」との記載から、各エリアにおけるデベロッパーサイト構築の必要性と、そこで提供される API に関するカタログサイトとの連携・住み分けの必要性が示唆されている。

さらに、各スーパーシティの取り組みにおける API を集約・公開するカタログサイトと、各エリアで構築されるデベロッパーサイトの連携、及びエリア同士の情報連携を実現するため、各エリアのデベロッパーサイトの構築においては連携を前提とした規格や品質の均一化が図られる必要がある。これについては、データ連携基盤活用のために最低限具備すべき機能を、中央から各エリアに対して提言することが望ましい。

上記を踏まえ、本章では、エリア横断で API の共通ルール・標準仕様等について中央から情報を公開するカタログサイトを「API カタログ」、そこからリンクされ、各エリア独自の API 詳細情報を公開したり、先端的サービスの開発に寄与する各種機能を各エリアの開発者に対して提供する場を「開発者ポータル」と定義する。

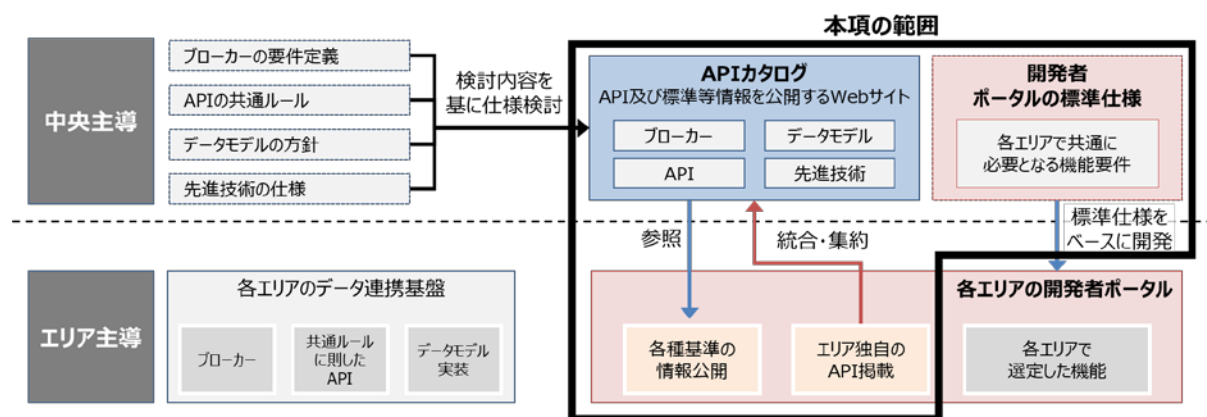


図 4-1-1 API カタログと開発者ポータルの本項における検討範囲

⁷ 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

⁸ 「スーパーシティ」構想について

<https://www.kantei.go.jp/jp/singi/tiiki/kokusentoc/supercity/supercity.pdf>

本章では、スーパーシティ推進において必要となる API カタログ・開発者ポータルが具備すべき機能を調査するとともに、中央とエリアの連携実現の観点から、中央が担うべき役割と各エリアで提供されるべき役割を整理することで API カタログと開発者ポータルのあるべき姿を詳細化する。

本章に記載の報告内容は、スーパーシティの取り組み主体が、各エリアに対して準拠させるべき基準や共通に実装させるべき機能といった内容を検討する際のインプットのひとつとして活用されることを想定している。また各エリアにおいては、本章を参照し、構築すべき開発者ポータルと全体における連携イメージ、及び必要となる準備事項等をより具体的にイメージしてもらうとともに、必ず実装されるべき機能とエリアの特色に合わせて独自で検討の余地がある機能とを検討する際の指針のひとつとなることを目的とする。

なお本章は、中央が整備すべき機能と各エリアで整備されるべき機能の住み分けに関する提言を主たる目的とし、詳細な実装方式や利用技術、及びそれらを基に検討されるべき非機能要件については検討に留めるものとする。また、各エリアで実装されるべき機能に関しても、各エリアの地域色の差を鑑み、必ずしも全国共通で必要ではないと想定される機能については、機能の概要レベルの提言に留めるものとし、詳細は各エリアが個別で実装検討すべきものとして位置付けている。

4-2. 仕様検討結果

4-2-1. 仕様検討結果

API カタログと開発者ポータルに関する基礎調査・仕様検討、ヒアリング調査を通して、各仕様の整理を行った。基礎調査・仕様検討、ヒアリング調査の内容および仕様検討結果の詳細については、「データ連携基盤技術報告書」6章を参照。

(1) API カタログの機能一覧案

API カタログの機能一覧案の仕様検討結果について表 4-2-1 に示す。

表 4-2-1 API カタログの機能検討結果

No.	画面名	機能名	機能概要	必須
1	全画面共通	メニュー機能	API カタログの全画面共通で、画面上部にサイトの各メニューへアクセスできるリンクを表示する。	○
2	サイト トップ	更新情報お知らせ	API カタログにて管理しているコンテンツ（ガイドラインや利用規約等）の更新情報を表示する。 また、API カタログに掲載されているエリアに追加があった場合も、この画面で表示される想定。ただし、各エリアの API の追加/更新/削除については表示しない。	○
3	サイト トップ	障害・メンテナンス 情報表示	API カタログの障害情報やメンテナンス時の計画停止情報等を表示する。	—
4	利用ガイド	利用規約公開	API カタログで各エリアに対して公開される情報（ガイドライン、API 仕様、標準データモデル等）に関するライセンス形態、禁止事項、免責事項を表示する。 API カタログに掲載される、各エリアの API の利用規約については、掲載元である各エリアの開発者ポータルを参照することとし、それに関する免責の記載が必要な場合はこの画面に表示する。	○

No.	画面名	機能名	機能概要	必須
5	利用ガイド	利用手順公開	<p>API カタログの概要、操作手順を表示する。API カタログでできることや、公開されている情報等をわかりやすく表示し、各メニューへリンクさせる。</p> <p>また、API カタログの将来機能として、ユーザ情報登録機能等が実装される場合は、登録フロー等の情報が掲載されるべきと想定する。</p> <p>ただし、API カタログに掲載される、各 API の利用手順については、各エリアの開発者ポータルを参照することとする。</p>	○
6	利用ガイド	エリア状況確認	<p>データ連携基盤を実装しているエリアの一覧を表示し、当該エリアの開発者ポータルへのリンクを掲載する。</p> <p>また、データ連携基盤の稼働状況（構築中、稼働中、稼働日等）や、稼働しているサービスの状況を表示できることが望ましい。元となる情報については、各エリア側から申告機能を将来実装することを想定する。</p>	○
7	利用ガイド	関連サイトリンク表示	<p>関連するサイトへのリンクを表示する。</p> <p>(1) 関連する省庁のリンク 各省庁にて実績のあるサービスや API 情報へのリンク</p> <p>(2) 基準・技術情報へのリンク データ連携基盤構築、開発者ポータルやサービス開発に関連する技術情報や開発ツール等</p> <p>(3) 内閣府のスーパーシティ関連情報へのリンク</p>	—
8	API 公開・提供	API 一覧表示	<p>各エリアで提供している API の概要情報を一覧形式で表示する。概要情報として連携されるべき項目は「API 名称、保有機能の概要、提供データに関する説明（データ更新日、提供元、データ内容、更新タイミング、その通知方法）」等が想定される。</p> <p>また、主たる想定利用者である先端的サービス事業者がサービス開発に必要な情報を迅速に入手できるよう検索機能は必須とし、検索可能項目についても、エリア、分野、データモデルの形式等複数の項目から横断的に検索できることが望ましい。</p> <p>なお、API 詳細仕様等については、API 提供元へのリンク先を併せて掲載し、そちらで参照される想定。</p>	○
9	API 公開・提供	API 情報収集	<p>各エリアの API 情報を自動で連携するための方法として将来的に検討が必要となる機能。</p> <p>中央から各エリアの情報を吸い上げる Pull 型と、各エリアから情報を登録してもらう Push 型での実装方式が考えられる。</p> <p>詳細な実装方式は将来的に検討が必要となるものの、Pull 型の場合は各エリアのデータ連携基盤上に API 概要や利活用状況といった情報を格納してもらい、中央からそれらの情報へアクセスする方式が考えられる。</p> <p>また、Push 型の場合は API カタログの画面を通じて登録してもらう方法や、API の公開や変更と同時に、自動で API カタログへの更新用 API がキックされる機能を各エリアの開発者ポータル</p>	—

No.	画面名	機能名	機能概要	必須
			ルやデータ連携基盤上に共通実装してもらうこと等が想定される。	
10	API 公開・提供	API 統計情報表示	API カタログで公開している各エリアの API について、API カタログ上で頻繁に参照されているものや、各エリアのデータ連携基盤において特に利活用されている API の情報を表示する。 各エリアの利活用状況については、個別に連携・登録される必要があるが、API カタログにおける被参照情報の収集にあたっては、Google アナリティクス等のアクセス統計ツールが活用できると想定する。 当該統計情報はアクセス負荷情報の把握にも利活用される想定。	—
11	開発者向け情報提供	ドキュメント公開	先進技術情報、データ連携基盤構築や開発者ポータル開発ガイドライン等のドキュメントを表示、ダウンロードを可能とする。 PDF や HTML 形式での公開も可能ではあるが、GitLab、GitHub、Note 等を活用することで、版管理や版ごとの変更差分を表示できる等のメリットが存在し、更新履歴や更新部分を説明するための資料作成等の手間を削減することが可能となる。	○
12	開発者向け情報提供	ツールキット提供	各エリアがデータ連携基盤構築や開発者ポータルを構築する際に活用できるテンプレート・ツールキット・SDK 等を公開する。 公開にあたっては API カタログ上に直接ファイルを公開する方式に加え、GitLab、GitHub、Docker 等の外部ツールを提供内容や提供目的に応じて使い分けることが想定される。	○
13	開発者向け情報提供	開発者コミュニティ	開発者間のやり取りを実現する。開発者間で開発上の課題を共有・解決することで、問合せ対応の負荷軽減や改善点の発見、ナレッジの蓄積を行う。 外部のチームコラボレーションツールを利活用することを想定しており、API カタログ上にはそれらへのリンクや利活用方法が掲載されることを想定している。現段階で本機能に活用できると想定するチームコラボレーションツールの例は以下のとおり。 (Zendesk、phpMyFAQ、GitLab、GitHub、Slack、Teams)	—
14	FAQ 一覧	FAQ	開発者コミュニティ、又は問合せ回答でよく受ける質問・回答例を整理、表示する機能。	—
15	問合せ/回答	各種ガイドラインや API カタログ自体に関する 問合せ/回答	先進技術情報や、データ連携基盤構築ガイドライン、開発者ポータルガイドライン等に関して個別に問合せを受付ける画面。 問合せ対応負荷を軽減するため、受付時にプルダウンメニューからカテゴリを選択する方式の実装や、FAQ への誘導がなされることが望ましい。	—

(2) 開発者ポータルの機能一覧案

開発者ポータルの機能一覧案の仕様検討結果について表 4-2-2 に示す。

表 4-2-2 開発者ポータル機能検討結果

No.	画面名	機能名	機能概要	必須
1	全画面共通	メニュー機能	開発者ポータルの全画面共通で、画面上部にサイトの各メニューへアクセスできるリンクを表示する。	○
2	サイトトップ	更新情報お知らせ	開発者ポータルにて管理しているコンテンツ(各 API の詳細情報、簡易検証環境、利用規約等)の更新情報を表示する。	○
3	サイトトップ	障害・メンテナンス情報表示	当該エリアのデータ連携基盤や開発者ポータルの障害情報やメンテナンス時の計画停止情報等を表示する。 将来機能として検討されている当該エリアのサービス稼働状況が実装されている場合は、併せて表示する。	○
4	利用ガイド	利用規約公開	開発者ポータル上で公開される情報(各 API の詳細情報、オープンデータ、簡易検証環境・開発者コミュニティといった開発支援ツール)に関するライセンス形態、禁止事項、免責事項を表示する。	○
5	利用ガイド	利用手順公開	開発者ポータルの概要、操作手順を表示する。開発者ポータルでできることや、公開されている情報等をわかりやすく表示し、各メニューへリンクさせる。 また、開発者ポータルの将来機能としてユーザ情報登録機能が実装される場合は、登録フロー等の情報が掲載されるべきと想定する。	○
6	利用ガイド	関連サイトリンク表示	関連するサイトへのリンクを表示する。 (1) 基準・技術情報へのリンク データ連携基盤構築、開発者ポータルやサービス開発に関連する技術情報や開発ツール等 (2) オープンデータ技術情報 オープンデータの公開開始日や拡張子等、メタデータ(カタログデータ、組織/人等)、推奨データセット (3) API カタログへのリンク	—
7	API 仕様公開・提供	API 一覧表示	各エリアのデータ連携基盤で実装されている API の情報を一覧表示する。SwaggerUI や ReDoc 等の機能を使い OpenAPI 仕様ドキュメントを公開する想定。 また、主たる想定利用者であるサービス事業者が、求める情報にリーチしやすいよう、キーワード検索機能やスーパーシティの分野ごとに入り口を分けること等が有効と想定する。	○
8	API 仕様公開・提供	API 詳細	各 API の詳細な仕様を公開する。 SwaggerUI や ReDoc 等のツールを使い OpenAPI 仕様ドキュメントを公開する想定。	○

No.	画面名	機能名	機能概要	必須
9	API 仕様公開・提供	API 開発評価環境 提供 (簡易検証環境)	先端的サービス事業者等に対して、サービス開発評価向けに API を簡易に検証するための環境を提供する。 実装される場合の方式は以下のいずれかを想定。 (1)検証用のデータ連携基盤を構築してそこに対してリクエストを送ってもらう。 (2)モックアップのような形で、定型的なレスポンスを返す小規模な検証環境を構築する。 なお、費用対効果の面で全エリア必須機能として実装されるにはハードルが高いため、必ずしも本機能を実装せずとも、サンプルプログラムの提供やマニュアルの充実化等でも代替可能。	—
10	API 仕様公開・提供	活用事例	各エリアで、データ連携基盤上の API を活用したサービス・ソリューションの事例をモデルケースとして資料化し、他事業者参照頂くことでデータ連携基盤の利活用が推進され、新たなサービスの参画も促進される。 当該機能では、それらの活用事例の資料を公開する。 また、各地域単位で実施されるコミュニティイベントや事業者向けイベント（アイデアソン・ハッカソン・各種勉強会等）の資料等もここで公開されることが望ましい。	—
11	API 仕様公開・提供	API 登録	当該エリアのデータ連携基盤独自で開発した API の登録やメンテナンスを行う。API の活用事例を表示する。 当該エリアのデータ連携基盤で開発した API 情報の登録・更新・削除を行う。 OpenAPI 仕様ドキュメントが登録される想定。API 情報の登録にあたっては、ユーザ登録等の機能を通じて、承認されたユーザのみが登録できることが望ましい。	—
12	開発者コミュニティ	情報交換機能	開発者間のやり取りを実現する。開発者間で開発上の課題を共有・解決することで、問合せ対応の負荷軽減や改善点の発見、ナレッジの蓄積を行う。 外部のチームコラボレーションツールを利活用することを想定しており、開発者ポータル上にはそれらへのリンクや利活用方法が掲載されることを想定している。 現段階で本機能に活用できると想定するチームコラボレーションツールの例は以下のとおり。 (Zendesk、phpMyFAQ、GitLab、GitHub、Slack、Teams)	○
13	FAQ	FAQ 一覧	よく受ける質問・回答例を整理、表示する機能。 開発者コミュニティでも同様の役割は担えるため必須機能とはせず、各エリアにおけるコミュニティの在り方やユーザ管理の方法に応じて全ユーザが参照すべき内容がある場合は個別実装される想定。	—

API カタログと開発者ポータルの基本的な考え方のイメージを図 4-2-1 に示す。

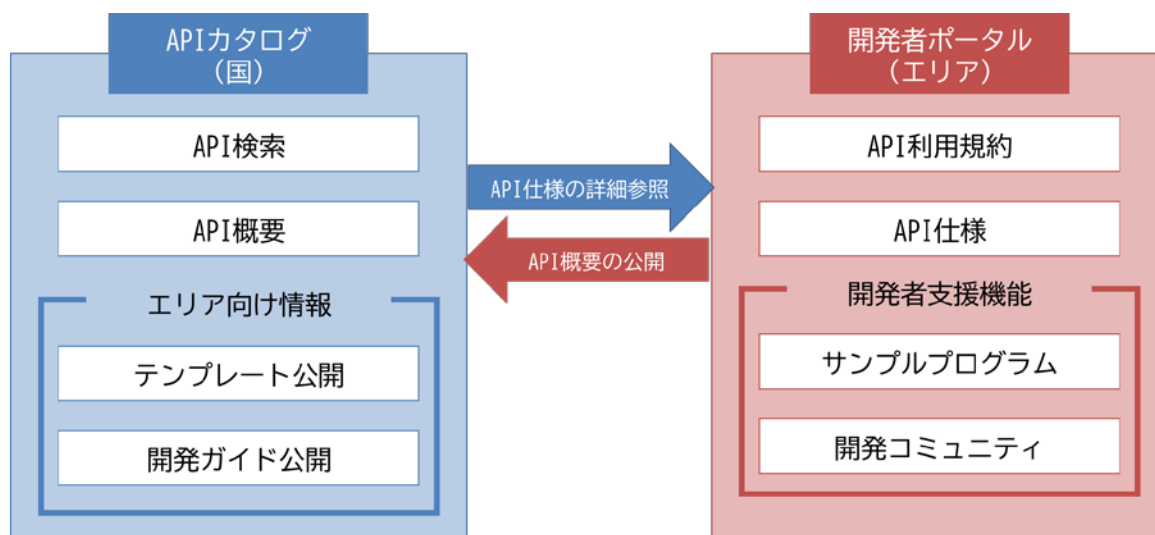


図 4-2-1 API カタログと開発者ポータルイメージ

また、API カタログと開発者ポータルの役割は表 4-2-3 のとおりである。

表 4-2-3 API カタログ/開発者ポータルの役割

対象	主体	目的
API カタログ	国	<ul style="list-style-type: none"> 先端的サービスの開発の際に、各エリアのデータ連携基盤で提供される API やサービスについての情報が得られること。 エリアがデータ連携基盤を構築・運用する際に、他エリアの事例や情報を入手できること。 各エリアにおけるデータ連携基盤・スーパーシティの実現を促進するために、各エリアがデータ連携基盤及び開発者ポータルの構築に必要なドキュメント・ツール、API 標準仕様等の情報を公開すること。
開発者ポータル	エリア	<ul style="list-style-type: none"> データ連携基盤の利用者が、データ連携基盤を用いた先端的サービスを円滑に開発できること。 データ連携基盤で実装されている API の詳細情報や利用規約・手順等のドキュメント公開、開発者間で問題解決を図るためのコミュニティ機能等、基盤を有効に利活用できること。

5. 調査結果（ウ）データモデルの方針

本章では、スーパーシティにおいて様々なサービスを実現する上で用いられるデータの項目、形式、構造、複数のデータ間の関係性等を示すデータモデル⁹について、異なるスマートシティ間での相互運用性を確保することを目的にその取り扱いや作成の方針を示している。ただし、既に独自のデータモデルを使

⁹ データモデルには、あるデータと他のデータの関係やデータの構造を表現する定義も含まれる。よって本書のデータモデルを参照することは、データの構造や意味を人間やデータを処理する機械に伝達し、相互運用性を確保するための手段の意味を含む。なお、様々なデータモデルの記述レベルのうち、「データ項目名」とその「データ項目の説明」までのシンプルなデータ項目定義を対象とする。

用している場合においては、必ずしも本章で示すデータモデルに変換することを強制するものではなく、本章の内容は、あくまで様々な利用場面を想定してデータモデルの在り方を検討した結果を例示するものであり、スーパーシティ、スマートシティにおけるあらゆる利用場面に適用できる完全性を保証するものではない。

本章において扱うデータモデルの対象範囲は、特定のデータ形式を対象とせず、地理空間データの基盤となるデータを対象とし、地理空間データについては別途調査を行った。なお、地理空間データ以外の「動的・静的データ」「パーソナルデータ」については、現時点で公式な定義がなされていないため、今後の検討対象とした。なお、データモデルを構成するデータ項目は、国内外の主要なスマートシティのデータモデルの中で共通的に使われている項目、国内でニーズの高い項目を、有識者ヒアリングを踏まえて選定したものである。

本データモデルは、スマートシティにおける各種サービスで用いる地物（ちぶつ）と移動体を中心に検討し、将来、本データモデルを行政情報等へと展開を図れるよう改版を予定している。本データモデルを提示するにあたり、可能な範囲で既存の参照情報を用いることにより、スーパーシティやスマートシティ関係者のデータモデルの設計や活用の実装が容易になるようにした。しかし、実際に使用する際は、本データモデルを各サービスにおいて、使用方法および制約条件等、当該サービスの全体像から整理をすることが前提である。

5-1. データモデルの役割

5-1-1. 一般論

データモデルは、複数のデータ項目を関連付けてグループ化・構造化し、各データの値の意味を誰もが正確に解釈できるようにする役割を持つ。さらに、対象となる現実世界をデータとして抽象化し、その関係や構造を特定の表現形式で記述し、人間にわかりやすくかつコンピュータ処理にも適したモデルが望ましいとされる。また、有益な情報として、第三者がデータをサービスや事業等に活用する場合は、データをシステム間で連携する場合には、そのデータの持つ意味と併せて、データの型（数値型、文字列型、または「あり、なし」を示す識別子等）、桁数、単位等を定義し、当該定義の情報を開示・共有する必要がある。データのコンセンサスを得るためには、データの値に限らず、当該データの意味・型・桁数・単位等コンセンサスを得るために必要な情報を定義したデータモデルを組にして相手に伝える必要がある。

5-1-2. データモデルと相互運用性

本書において、データモデルを用いた相互運用性の考え方として、システムや組織を超えてデータを活用するステークホルダがビジネス目的や費用対効果を考慮し合意しやすい手段を前提に、社内システムや特定の業界などステークホルダが固定されている場合に精緻なレベルでデータの意味を共有しやすい「①個別にすり合わせ」、新たなデータセットの整備や EC サイトで定型フォームへ購入情報を登録等、値に制約をかけられる場合にデータの質を確保しやすい「②モデルを統一」、各システムのデータモデルの定義を変更せずに参照モデルの相違点を明らかにすることでデータの相互運用性を確保しようとする「③参照モデルを共有する」に整理した（図 5-1-1）。従来のシステム連携やデータ連携では一般的に当該手法が採用されてきたと考えられるが、多様なデータ源で生成・処理されるデータを連携させるスーパーシティにおいては、自由度の低さが課題になる。

①個別にすり合わせ

②モデルを統一

③参照モデルを共有する

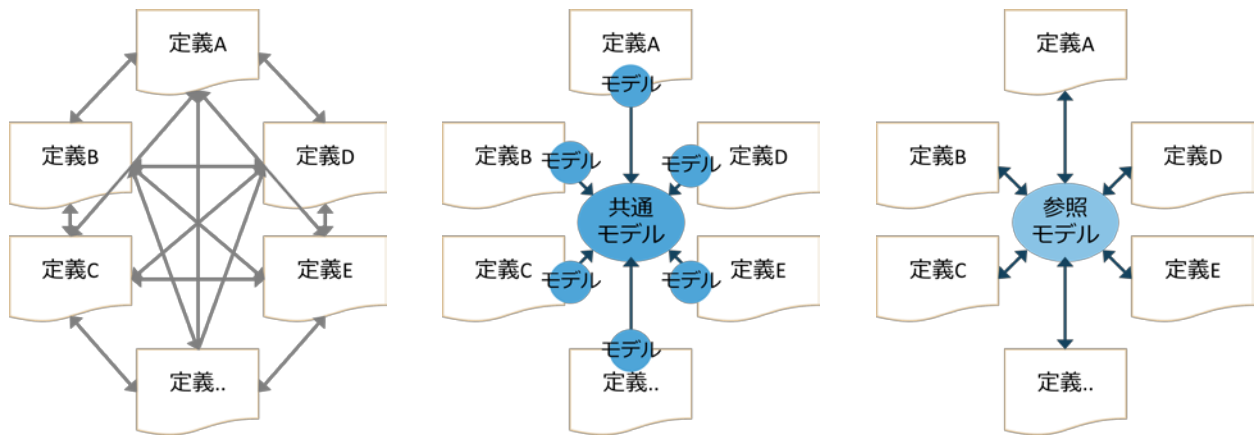


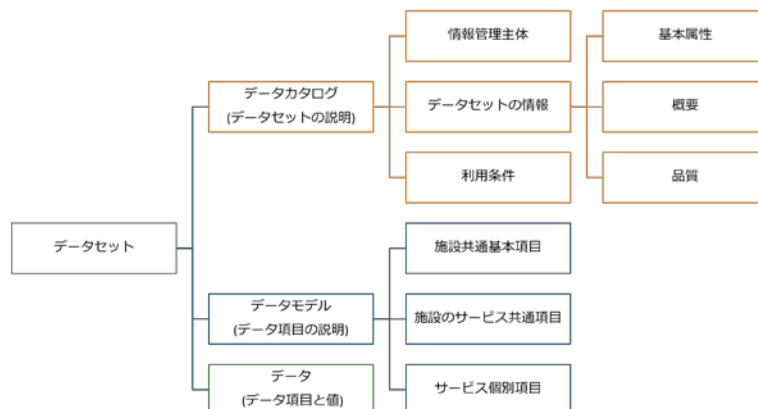
図 5-1-1 相互運用性の考え方

5-1-3. スーパーシティにおけるデータモデルの役割

スーパーシティのデータは、特定の自治体の 1 つの情報システムの中だけに閉じて利用されるのではなく、他の自治体や企業・団体、居住者、来街者、さらには海外の政府・自治体や企業・団体が管理する情報システムに受け渡され、異なる情報源から得られる複数のデータを統合、集計、加工等、様々な用途に使われる。そのため、当該データを扱う全ての情報システムが、各データの形式や意味を正しく識別できるように、一定の共通ルールを定め、各システムが当該ルールを認識しておく必要がある。加えて、土地、建物、店舗、土木構造物、機械、車両、人、エネルギー、催事イベント等、管理対象の種類によって取り扱うデータの形式や構造が異なる。システム間でデータを受け渡して利活用するためには、各データがどのような形式や構造で記述されているのかを全てのシステムが共通して識別できる必要がある一方で、技術の進化や事業モデルの変遷、法規制の改正等によって様々なデータ利活用のユースケースが発生することも想定される。そのため、スーパーシティ、スマートシティのデータ連携基盤を実際に設計・構築・運用する際には、既存のデータモデルに適合させるだけでなく、将来にわたり継続的に、新しいデータモデルの追加や複数のデータモデルを階層的に構造化する組み合わせ方の追加変更の対応ができるようになるべく柔軟で拡張性のある技術や方式を採用することが望まれる。

5-1-4. データモデルとデータカタログ

スーパーシティでデータ連携される全てのデータセットは、データセット提供者(情報管理主体)とデータセットの情報、利用条件といったデータセットを説明するメタデータすなわちデータカタログの記載内容(オレンジ色の枠)、データ項目に関するメタデータすなわちデータモデル(青色の枠)を付与される必要がある。(ユースケース等の詳細は本書参照)



5-2. 推奨データモデル概要

5-2-1. 推奨データモデルの整理原則

本書において、データモデルを(1)参照モデルとしての整備、(2)既存データモデルの活用、(3)スモールスタートと成果の可視化、(4)グローバル標準との整合性、(5)イノベーションのための成長の仕組み、(6)多様な用途に永く使える仕組みの原則に基づいて整理した。なお、本データモデルは、完全性を目指すものではなく、継続的な改善の対象とすることを原則としている。

5-2-2. 検討対象・検討方法

データモデルの検討対象として、国内外の主要スマートシティで共通的に使われているデータモデル、国内でニーズの高いデータモデルを対象に調査・取りまとめを行った。推奨データモデル検討方法は、**エラー! 参照元が見つかりません。**を対象に、実在するデータモデルのデータ項目名及びその説明の定義を検討、整理して記載した。データモデルの検討手順は、対象となるデータのテーマに関して、推奨データセット、IMI 共通語彙基盤、国土数値情報及び GTFS、Schema.org などの既存規格で示されているデータモデルを抽出して行った。なお、スマートシティ間、スーパーシティ間で相互運用性を確保することが前提であるため、推奨データモデルの最小構成には極力沿うこととした。

5-2-3. 基本データの共通記述方法

全てのメタデータにおいて、共通的なデータモデル、データモデルを構成する要素の値の共通記述方法を本書にて示した。項目は、文字、外国語表記・ピクトグラム、日時、経度・緯度、住所等、連絡先、センサーデバイスである。(詳細は本書参照)

5-2-4. 推奨データモデル

スーパーシティ、スマートシティでニーズが高いと考えられるデータのデータモデルを相互運用性を確保するため、5-2-1 推奨データモデルの整理原則にしたがって、可能な範囲で既存のデータモデルを参照し、本書にて示した。示した項目は、今後の検討課題を含めて、土地、建物、施設、出入口、設備、道路、その他の建物、交通、イベント、センサーデータ、建物内・地下街、地下埋設物、移動オブジェクト、自然、緊急情報、行政情報である。(詳細は本書参照)

5-2-5. データカタログのデータモデル定義

データやデータセットが生成された目的と異なる用途でのデータ利活用が期待されるスーパーシティでは、極力標準的なメタデータで記述されたデータカタログが求められる。このデータモデルは、スーパーシティにおいてサービスを提供する事業者が、必要なデータの要件をデータ連携基盤等を通じて要望し、要件に近いデータセットを保有する者が要望に応じるといった用途にも活用され得る。

本書では、スーパーシティのデータ連携に必要と考えられるメタデータを「情報の管理主体」、データセットの基本的な属性を記載するためのデータモデルである「データセットの基本属性」、データセットに含まれるデータの概要を記載するデータモデルである「データセットの概要」、データセットを入手する前に必要な情報である「データの品質」、データセットの利用に必要な契約ポリシー、利用期間や利用範囲、有償/無償の区分や支払条件等の利用条件を示すための「データセットの利用条件」に抜粋して説明を加えた。(詳細は本書参照)

5-3. 今後のデータモデル確立へ向けた課題

本来データをシステム間で共有・連携させる際は、全国で統一されたデータモデル定義が運用されており、全てのシステムがその定義を参照してデータを送受信できる状態が理想である。また、データ連携において実際に必要なデータやデータ連携方法は、実現すべきサービスやスーパーシティごとのビジョン、また技術環境等によって異なり、さらにそれらの成熟に伴って変化を続けていくため、政府、自治体、アーキテクト、事業者等が協調しながらデータモデルの改善を重ねていけるよう、効果や課題の共有と議論の場を持続的に運用することが求められる。