

デマンドバス システム連携API標準仕様書 ガイダンス



2026年2月
国土交通省 総合政策局 公共交通政策部門 モビリティサービス推進課

目次

1. プロジェクトの概要

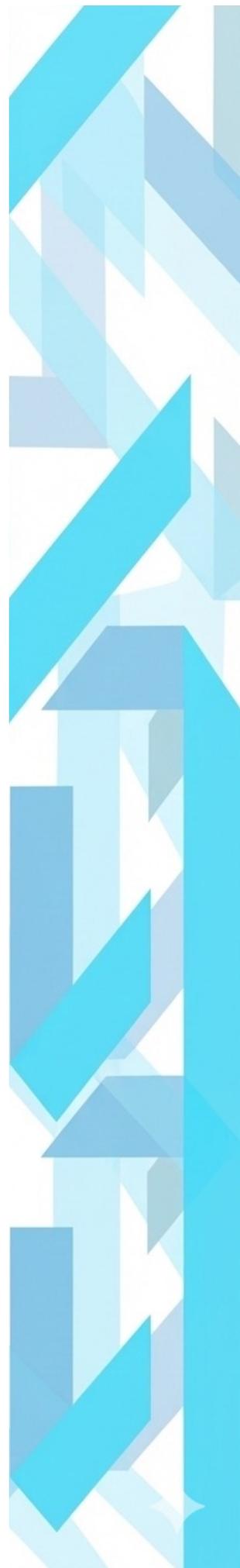
- 1.1. はじめに
- 1.2. デマンドバスについて
- 1.3. デマンドバスにおける現状課題
- 1.4. デマンドバスシステムの
API標準化による課題解決
- 1.5. API標準化のプロセス
- 1.6. API標準化がもたらす便益
- 1.7. 標準ドキュメントの種類

2. 標準ドキュメントの読み方

- 2.1. システムアーキテクチャの読み方
- 2.2. APIの読み方
- 2.3. ERDの読み方
- 2.4. 業務フロー/業務一覧の読み方

3. 標準ドキュメントの解説

- 3.1. 標準化のスコープと概要
- 3.2. システムアーキテクチャの解説
- 3.3. APIの解説
- 3.4. ERDの解説
- 3.5. 業務フロー/業務一覧(抜粋)の解説





1. プロジェクトの概要

- 1.1. はじめに
- 1.2. デマンドバスについて
- 1.3. デマンドバスにおける現状課題
- 1.4. デマンドバスのAPI標準化による課題解決
- 1.5. API標準化のプロセス
- 1.6. API標準化がもたらす便益
- 1.7. 標準ドキュメントの種類



1.1. はじめに

本ガイドスは、国土交通省が推進する地域交通DX推進プロジェクト「COMmmONS (コモンズ)」における、2025年度「デマンドバスシステム連携API標準化プロジェクト」の成果を解説するものです。

プロジェクトの目的

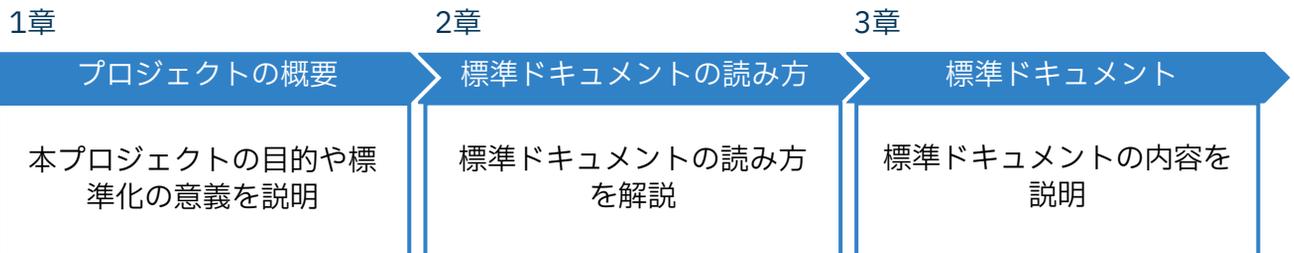
国土交通省では、地域交通DX推進プロジェクト「COMmmONS (コモンズ)」として、2025年度に「デマンドバスシステム連携API標準化プロジェクト」を実施しました。

このプロジェクトは、地域やシステムベンダーごとに仕様が異なるデマンドバスシステムにおいて、MaaSアプリ等との連携時に発生する個別開発の負担を解消し、システム連携の効率化および地域間の相互運用性の確保を図ることを目的としています。

具体的には、異種システム間を疎結合に接続するための「標準API仕様」および「標準業務モデル」を策定することで、アプリ構築や連携にかかる開発工数を削減するとともに、異なる地域のデマンドバスを同一アプリで予約可能にするなど、利用者の利便性向上と持続可能な地域交通サービスの実現を目指します。

本ガイドスの構成

本ガイドスは、プロジェクトの背景と意義を説明する「プロジェクトの概要（第1章）」、技術資料の読み方を解説する「標準ドキュメントの読み方（第2章）」、標準仕様の内容を説明する「標準ドキュメント（第3章）」の3章により構成されています。



本ガイドスの対象読者

本ガイドスは、デマンドバスシステムとMaaSアプリ等の連携を検討されている自治体の交通政策等の担当者様、システムの標準化対応を検討されているデマンドバスシステム事業者様及びMaaSアプリ事業者様等に向けた資料となります。



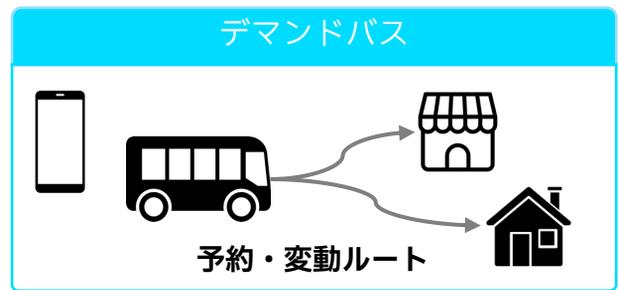
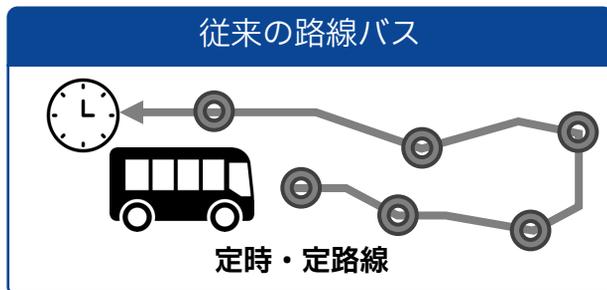
1.2. デマンドバスについて

デマンドバスとは、利用者の「予約」があった場合にのみ運行し、予約状況に応じて運行ルートや時刻を柔軟に変更できる乗合交通サービスです。「デマンド」とは「需要」や「要求」を意味し、利用者の要望に応じて運行することからこの名称で呼ばれます。

主な特徴と運行形態

地域の実情に合わせて多様な運行形態がとられますが、主に以下の特徴を有します。

- 運行の効率化
予約がない場合は運行しないため、乗客のいない状態で走る状態を防ぎ、効率的な運行が可能です。
- 柔軟なルート設定
予約に応じて、特定のエリア内を自由に走行します。
- 乗合（シェア）利用
タクシーとは異なり、原則として他の利用者と同乗して目的地へ向かう「乗合形式」で運用されます。



利用手順

一般的な利用の流れは以下のとおりである。

1. 事前予約： スマートフォンアプリ(または電話)を用いて、利用希望日時と乗降場所を予約する。
2. 配車： 予約内容に基づきルートが設定され、車両が指定場所へ配車される。
3. 乗車/移動： 指定の場所で乗車し、目的地へ移動する。

1. 事前予約



スマホで予約

2. 配車



ルート設定

配車

3. 乗車/移動



予約車両に乗車

導入メリット

利便性の向上

柔軟に乗降位置を設定することが可能であるため、バス停までの移動負担が軽減される。

地域時交通の維持

利用者の少ない地域や時間帯においても、無駄を省いた高効率な運行により、公共交通網を維持することが可能となる。

1.3. デマンドバスにおける現状課題

従来の公共交通の代替としてデマンドバスの導入が進む一方、様々なサービスが提供されているため、サービス間の相互運用が難しくなっています。

システム間の連携に必要な標準的なインターフェースが存在しないため、個別開発に伴う多大な工数負担が、MaaSアプリ等との連携や利便性向上の課題となっています。

解決すべき課題

- 廃線を余儀なくされた路線バスや鉄道の代替としてデマンドバスの導入が進み、新たな交通サービスとしての存在感が増している一方、多くのデマンドバスシステムサービスが参入した結果、様々なアプリやシステムが提供され、地域間の相互運用性が確保されていない状態となっている。
- 一部のMaaSアプリ等ではデマンドバスシステムとの連携が図られ、サービス利用のワンストップ化が実現しているものの、システム間の連携に必要なAPIやデータモデルを開発するためには個別の調整が必要となっている。
- 個別開発による工数負担が大きいことが、デマンドバスシステムとMaaSアプリ等の連携サービスの普及が進まない要因となっている。



利用者

- 私がいつも使うデマンドバスだと、隣町のデマンドバスは使えない、隣町のスーパーへの移動ニーズがあるが、移手段がなく断念している
- 免許返納すると隣町までどうやって行けば良いのか分からない



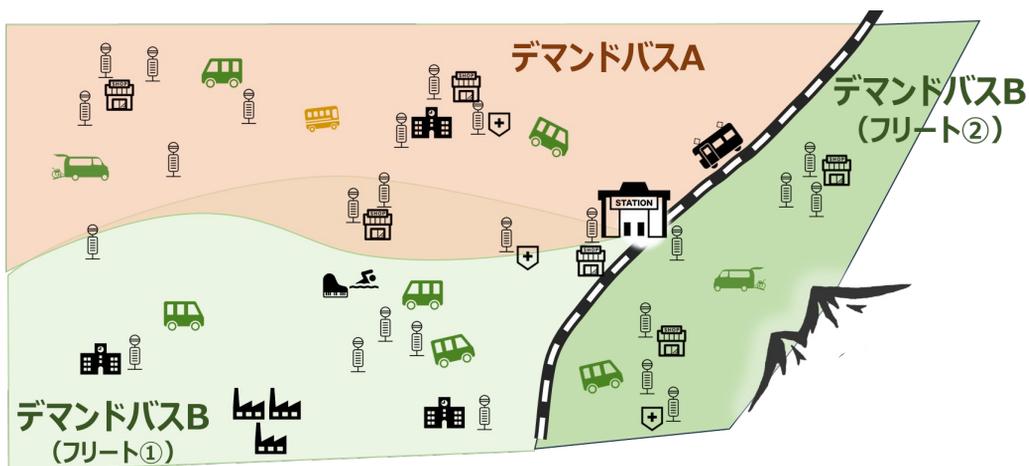
自治体担当者

- 隣町とのデマンドバスと連携し、広域での利便性を上げたいけど、システム提供事業者が違うから連携できない、どうすればいいかわからない
- なんとかシステムを連携させることを検討したいけど、そんな企画をする知見がない



MaaS
アプリ
事業者

- MaaSアプリとして様々なサービスをつなげて利便性を向上したいけど、サービスひとつひとつが違うと仕様理解をするだけでも大変
- 地域の利便性を向上するためにサービスをつなげたいけど、基礎自治体が複数になると費用面がなかなか折り合わない



※フリート：その地域やサービスエリア内でデマンド交通を担う車両群のこと

1.4. デマンドバスシステムのAPI標準化による課題解決

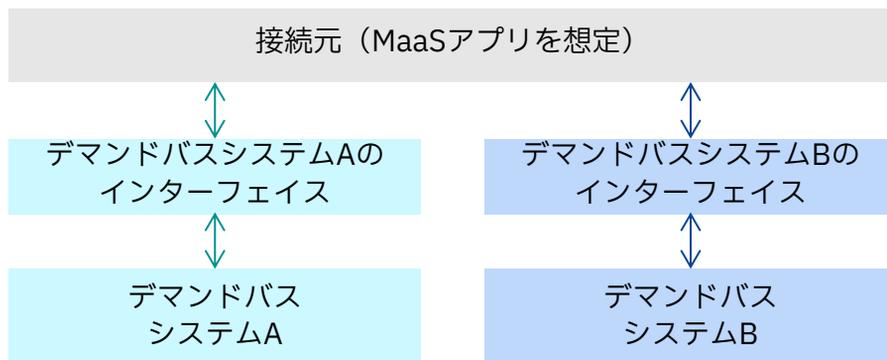
MaaSアプリ等のユーザー向けアプリケーションとデマンドバスシステムが連携し、予約や検索などをワンストップ化するためのシステム連携仕様と、その運用に必要な業務モデルの標準仕様を策定します。

これにより、システム間の連携に必要な開発工数の削減を図るとともに、多様なサービスの相互連携や地域間の相互運用性を確保し、地域交通の利便性向上を実現します。

課題解決のアプローチ

- MaaSアプリとデマンドバスシステムの連携を実現するための、両システムが連携するための標準API仕様を開発する。
- 標準API仕様の可用性を高めるため、疎結合な状態での連携を実現する仕様とする。既存のデマンドバスシステムの業務類型を整理し、これに基づく汎用的なAPI定義を行う。
- 標準API仕様の運用に必要な業務手順を整理し、円滑な導入を促進する。

As-Is



To-Be



1.5. API標準化のプロセス

標準化は以下の3段階で推進します。まず、デマンドバス事業者の既存業務やシステム仕様の現状を調査し把握します。次に、その仕様を基に標準的な業務モデルとAPI仕様を検討します。最後に有用性の検証を通して、コスト削減効果や技術的価値、社会実装の可能性などを踏まえた、API仕様を最終的に策定します。

現状調査

STEP①

関係事業者に対してヒアリングを行い、既存の業務フローやシステム仕様などの現状を調査・分析する。

API仕様検討

STEP②

現状調査で明らかになった要件をもとに、実態に即した実現可能性の高い標準化を検討する。

標準API仕様策定

STEP③

検討したAPIの有用性を確認するため、社会実装の可能性を検証し、その後標準API仕様を策定する。

現状調査対象企業

本プロジェクトにおいて、ヒアリングを実施した企業と選定理由は以下のとおりです。

#	業界	企業名	選定理由
1	MaaS アプリ	トヨタファイナンシャルサービス	日本最大規模のMaaSアプリであるmy routeを実装しており、豊富な事例を持っているため
2		福井県福井市	地域のMaaSアプリであるふくアプリ/ふくいMaaSを地域の民間事業者が実装して展開している先進的な事例を創ってきた地方公共団体のため
3		群馬県	地域のMaaSアプリとして先進的な事例を創ってきたGunMaaSを実装しており、デマンドバスを連携させたいという意向も強かったため
4		東日本旅客鉄道	地域のMaaSアプリとして先進的な事例を創ってきたGunMaaSを実装を担当している会社のため
5		ふくいのデジタル	地域のMaaSアプリであるふくアプリ/ふくいMaaSを地域の民間事業者という立場で実装支えている先進的な事例を創ってきた会社のため
6	デマンドバス	アイシン	チョイソコという名称でデマンドバスシステムの提供を実施しており、トップレベルの実装数を誇るため

現状調査及び標準化の対象

本取り組みにおける調査対象は、業務の根幹となる「業務モデル」およびシステム連携に関わる「API仕様」としている。それぞれの領域における詳細な調査結果を踏まえ、実効性のある標準ドキュメントの作成を進める。

業務モデル

#	調査項目名	主要論点	調査手法	調査アウトプット	獲得されるナレッジ
1	デマンドバスの社会実装事例の調査	•現状、デマンドバスはどのような運行形態で社会実装されているのか？	ドキュメントリサーチ	標準業務一覧 API仕様	デマンドバスシステム連携に関する事業性調査 (デマンドバスの業務モデルの整理、ユースケースを実現する構成例)
2		•APIを介したサービス間連携を想定した場合、業務をどの単位で分けるべきか？			
3		•APIを介したサービス間連携を想定した場合、標準化すべき業務モデルはどのように定義できるか？			
4		•上記3点の主要論点で現状業務とかい離がないか？			
5	MaaSアプリとの連携ユースケース調査	•MaaSアプリが、現状で、実現しているサービス連携にはどのようなものがあるのか？	ヒアリング	標準業務フロー	デマンドバスシステムAPIに関する有用性調査 (デマンドバスと連携するユースケースの調査)
6		•MaaSアプリとデマンドバスを連携させている実例があるか？またあるならどういった連携か？			
7		•MaaSアプリとデマンドバスシステムと連携させたい、という具体的なニーズがあるか？連携させたいと考えるときの課題は何か？			

API仕様

#	調査項目名	主要論点	調査手法	調査アウトプット	獲得されるナレッジ
1	既存のAPI仕様調査	<ul style="list-style-type: none"> •デマンドバスを実現するAPIにはどのようなものが用意されているのか？ •既存のAPIが持つ課題や機能差分は何か？ 	ドキュメントリサーチ	API仕様	デマンドバスAPIに関する技術的なフィジビリティスタディ (デマンドバス業務モデルの整理、デマンドバスAPIのインターフェース仕様)

1.6. API標準化がもたらす便益

本プロジェクトで策定した標準APIの導入により、交通事業者や自治体、システムベンダーのアプリ構築に掛かる工数を低減します。

また、異なるデマンドバス間で同一のUIで予約を実現し、サービス間連携を促進することで、利用者のアクセシビリティを向上します。

実証内容と仮説

● 工数削減の実証：

APIインターフェイスの共通化で、複数サービス連携した**アプリの構築工数が50%低減**される。

➡ 実装担当者はの実装工数に関するヒアリングと、事業者には工数削減期待効果をアンケート

● サービス間連携の実証：

デマンドバスシステムのアクセシビリティ向上として、デマンドバスサービスを**同一アプリから複数予約可能**となる。（3つ以上予約可能となる）

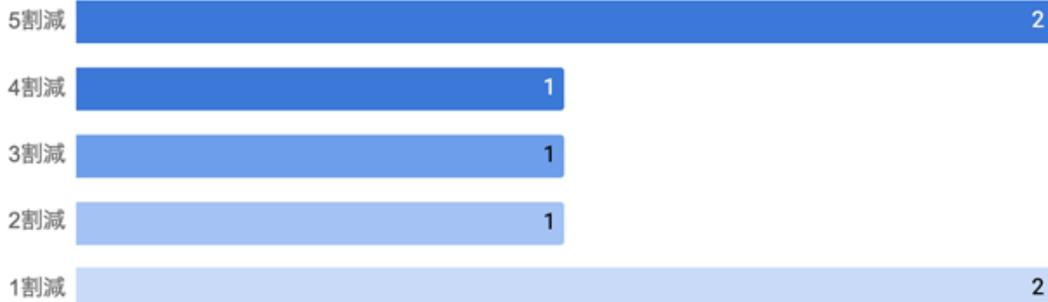
➡ 実際の検証環境を用意し、予約操作を行うことで検証

実証結果と便益

工数削減

標準APIを活用して開発実装した事業者からは、実装工数は半減以下との評価が得られました。

このことから、社会実装に向けてはサービス間連携のハードルを大きく削減可能であると評価できます。



サービス間連携

異なる地方公共団体のデマンドバスサービスを標準APIで呼び出し、同一UIで複数のデマンドバスサービスを予約できることを確認しました。



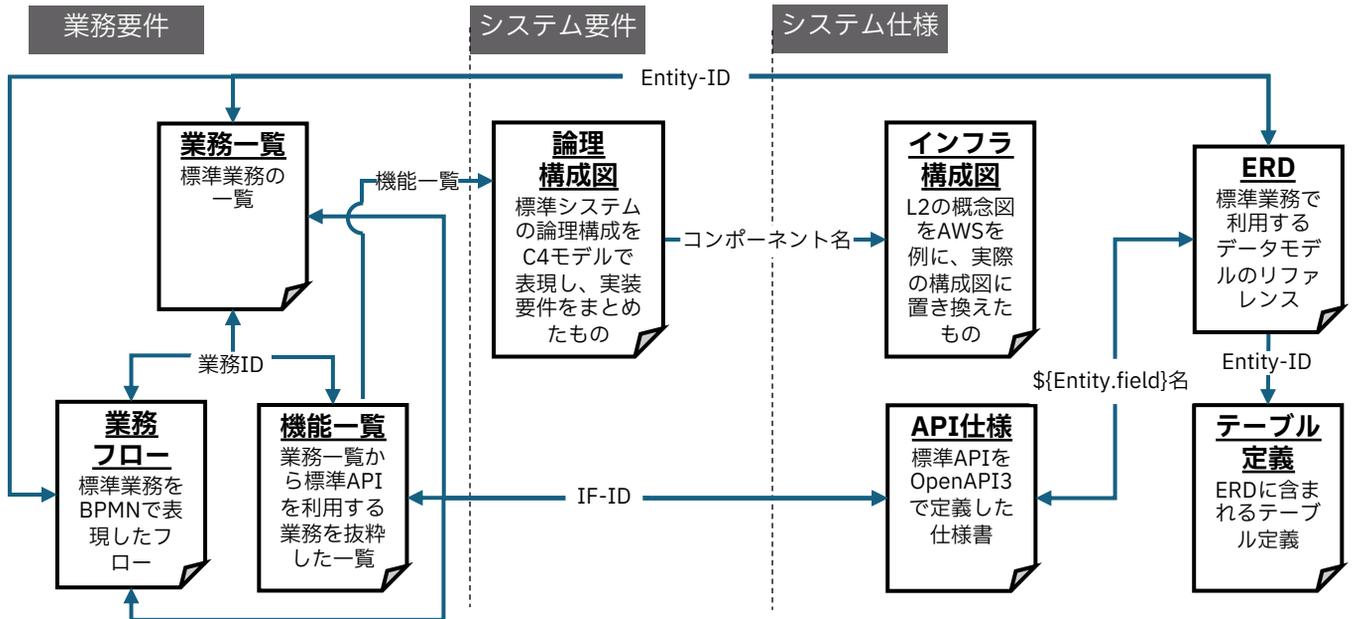
1.7. 標準ドキュメントの種類

標準ドキュメントでは、「業務要件覧」、「システム要件」、「システム仕様」の3種類で定義しています。

まず、システムが実現すべき「業務要件」を定義するのが「業務フロー/業務一覧/機能一覧」です。これを具現化する「システム要件」は、システムの全体像を示す「論理構成図」で表現されます。システム要件をさらに詳細化した「システム仕様」では、具体的なリファレンス実装を示す「インフラ構成図」と、システム間の連携インターフェースを示す「API」、データ構造やエンティティ間の依存関係を示す「ERD」で構成されます。

また各ドキュメントは、業務ID、IF-ID、Entity-IDなどの名称で機能ごとに採番され、各ドキュメント間で紐づけを行っております。

標準ドキュメントの種類



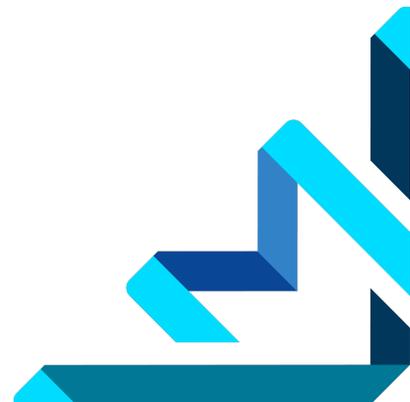
名称	内容	ファイル形式		想定読者
		参照用	Rawデータ	
デマンドバスシステム連携API標準仕様書_ガイダンス	標準化の意図、ビジネス上の意味など、仕様の背景や文脈（コンテキスト）の解説する。「標準ドキュメント」を読むための「前提知識」を提供する。	PDF	PPTX	
デマンドバスシステム連携API標準仕様書_業務一覧	業務フローで扱う業務内容をリスト化し説明する。IDにより他資料との紐づきを管理する	PDF	XLSX	交通事業者 自治体担当者 開発担当者
デマンドバスシステム連携API標準仕様書_業務フロー	抽象度の高い事業機能から、現場の作業手順までを体系化したもの。APIを定義するにあたり、システムが実現すべき業務の流れを説明する。	PDF	drawio	
デマンドバスシステム連携API標準仕様書_機能一覧	業務一覧から標準APIを利用する業務を抜粋した一覧	PDF	XLSX	
デマンドバスシステム連携API標準仕様書_論理構成図	APIを実装するための、システムの全体像を概念的に説明する。	PDF	drawio	開発担当者
デマンドバスシステム連携API標準仕様書_インフラ構成図	論理構成図のシステム部分を実際の実装例を踏まえて詳細化して説明する。	PDF	drawio	
デマンドバスシステム連携API標準仕様書_API	システム間の連携インターフェース仕様を説明する。この仕様を推奨として活用促進する。	HTML	yaml	
デマンドバスシステム連携API標準仕様書_ERD	APIが保持すべきデータの構造を定義する。	PDF	mermaid	
デマンドバスシステム連携API標準仕様書_テーブル定義	ERDで定義されたデータを実際のデータベースに落とし込むための仕様。	PDF	markdown	

※ドキュメントのダウンロードURL(<https://www.mlit.go.jp/commmons/document/003/>)



2. 標準ドキュメントの読み方

- 2.1. システムアーキテクチャの読み方
- 2.2. APIの読み方
- 2.3. ERDの読み方
- 2.4. 業務フロー/業務一覧の読み方



2.1. システムアーキテクチャの読み方

システムアーキテクチャでは、システムの論理構成およびインフラ実装の参照モデルを定義しています。論理構成には「C4モデル」を採用し、全体像から詳細への階層的な可視化を通じて関係者間の共通理解を形成します。また、インフラ構成ではAWS Lambda等を活用したサーバレス構成をリファレンスとして例示し、設計指針を示します。

論理構成図の説明

「Level 1 (System Context)」はビジネス視点で各プレイヤーとシステムの連携関係を示し、「Level 2 (Container)」はそのシステム内部を機能単位(コンテナ)に分解した技術視点の図です。

階層の意味

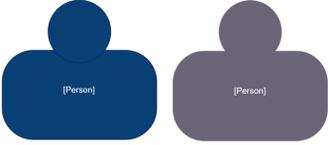
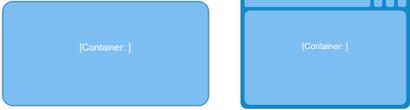
Level 1 (System Context)

- 視点: 誰が (アクター) 、どのシステムと連携するか。
- 目的: プロジェクトの境界線と、外部システムとの関係性を把握する。

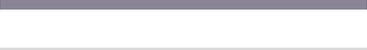
Level 2 (Container)

- 視点: システムの中身がどのような技術要素 (アプリ、API、DB) で構成されているか。
- 目的: ソフトウェアの責務分担とデータの保存場所を把握する。

記号の意味

	人型 (Person): <ul style="list-style-type: none">➤ システムを利用するユーザー、または事業者 (操作の主体)。
	四角 (Software System / Container): <ul style="list-style-type: none">➤ Level 1では: 1つの大きなシステム全体。➤ Level 2では: その中にある「Webアプリ」「APIサーバー」「モバイルアプリ」などの実行単位。
	円筒 (Database): <ul style="list-style-type: none">➤ データが保存される場所。

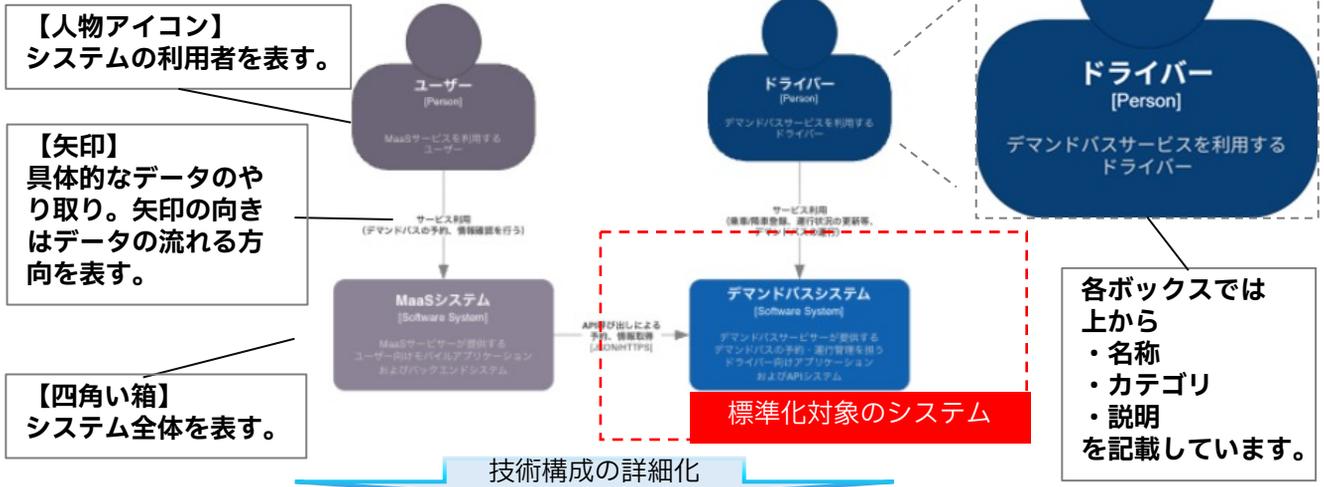
色の意味

	-----➤ 人物(IF標準化対象システム利用)
	-----➤ ソフトウェアシステム(IF標準化対象)
	-----➤ コンテナ(IF標準化対象)
	-----➤ コンテナ(IF標準化対象外)
	-----➤ 人物(IF標準化対象外システム利用)
	-----➤ ソフトウェアシステム(IF標準化対象外)

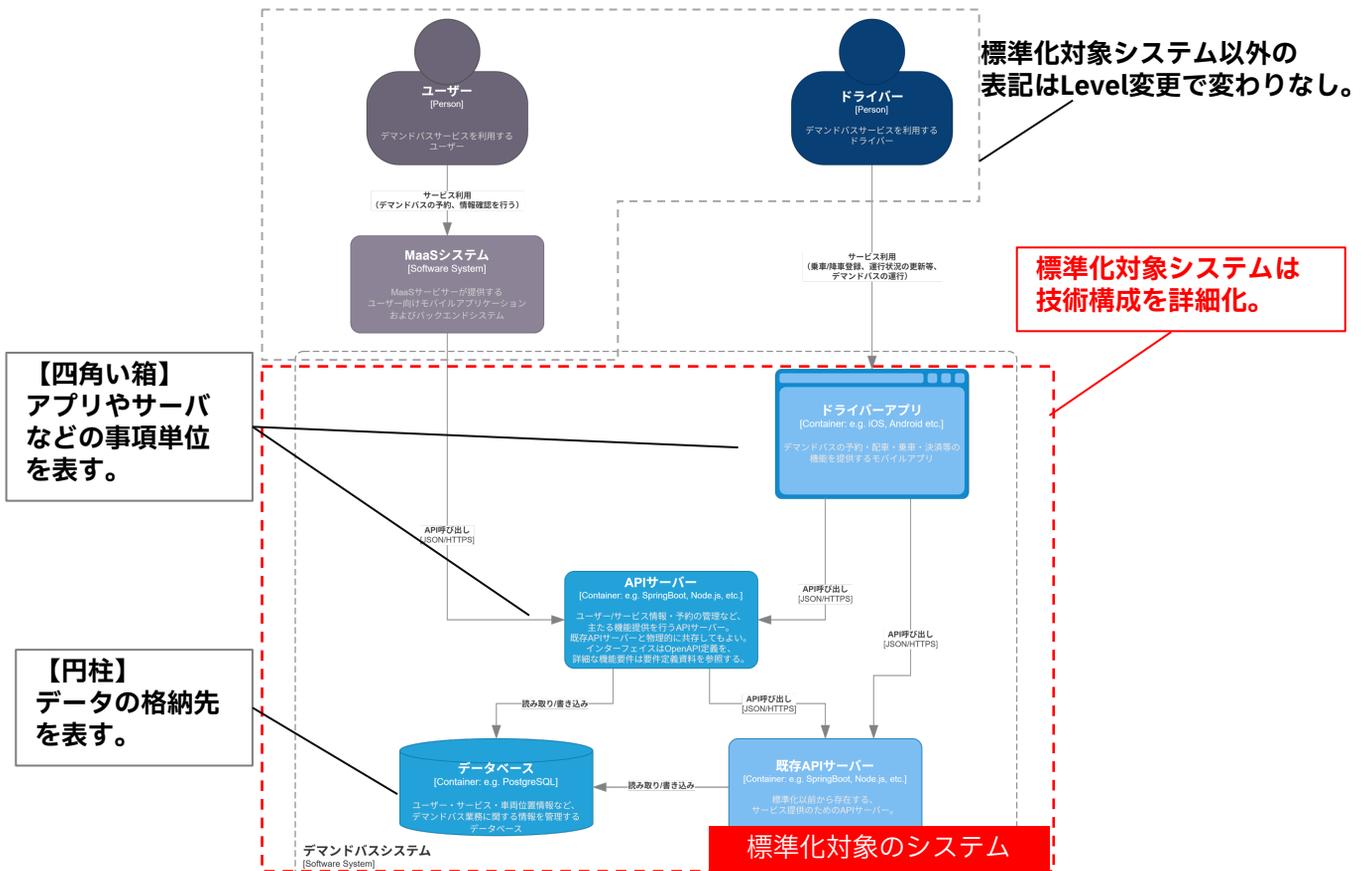
論理構成図の読み方

本図は標準システムアーキテクチャの階層構造を示します。上段（Level1）は利用者（ユーザー・ドライバー）とシステムの関わりや、外部システム（MaaS等）との連携関係を表します。下段（Level2）は標準化対象システム内部をアプリ・API・DB等の機能単位に分解し、具体的なデータの流れや実装の参照モデルを定義しています。

Level 1(System Context)の例



Level 2 (Container)の例

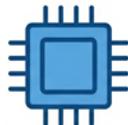


インフラ構成図の説明

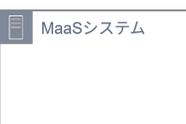
インフラ構成図は関連システムのクラウド構成の実装例を可視化したものです。本書では、広く一般に普及しているクラウドコンピューティングサービスであるAWSを実装例として用いていますが、利用するサービスを限定するものではありません。

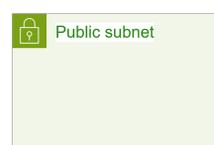
インフラ構成図は、利用者や管理画面からの操作が、セキュリティ層とAPIを経由し、サーバーレス処理とデータベースで処理・保存される流れを示しています。また、インターネット接続可能な領域と閉域を区分けし、さらに専用線接続などで安全性を担保したネットワーク設計全体を俯瞰できます。

システム構成要素

システム構成要素	実装例として扱う技術要素
 <p>セキュリティ</p> <p>ユーザーの認証・認可や通信の暗号化を行い、不正アクセスや脅威からシステムを保護する機能。</p>	  <p>AWS WAF AWS Network Firewall</p>
 <p>コンピュータ</p> <p>アプリやビジネスロジックを実際に動作させ、データを処理するための計算リソース（実行基盤）。</p>	  <p>Amazon API Gateway AWS Lambda</p>
 <p>データベース</p> <p>業務データやシステムの状態を保存し、効率的に検索・更新できるように管理する格納庫。</p>	 <p>Amazon Aurora</p>
 <p>ゲートウェイ</p> <p>外部からのリクエストを一元的に受け付け、適切なコンピュータリソースへ通信を中継・制御するシステムの玄関口。</p>	    <p>Internet Gateway NAT Gateway Direct Connect Gateway Virtual Private Gateway</p>

枠組みの説明

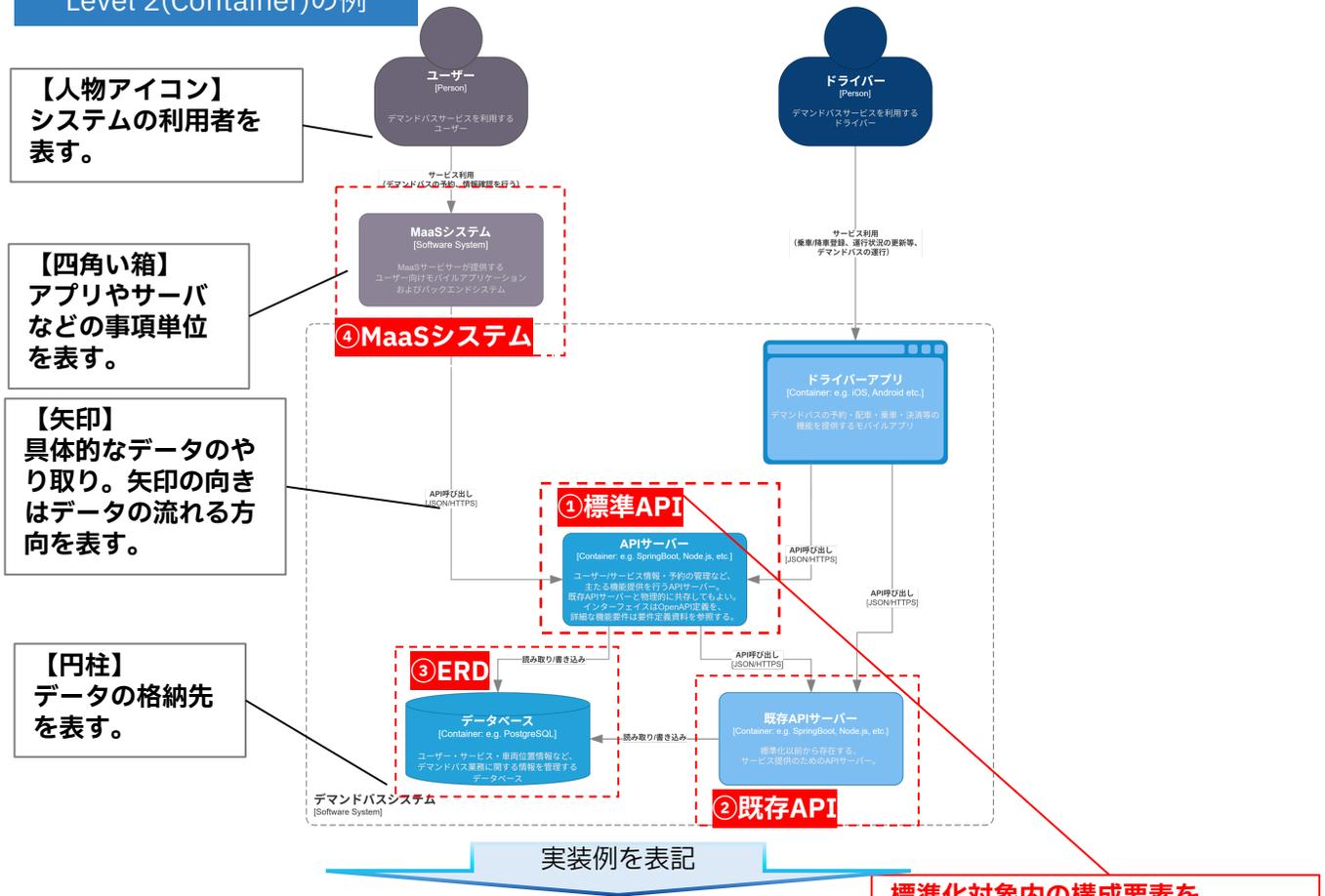
 <p>MaaSシステム</p>	<p>外枠(システム/アカウント): 「MaaSシステム」「認証システム」など、システムごとの管理境界を表します。</p>
 <p>VPC</p>	<p>VPC(紫の枠): Virtual Private Cloud。AWS上に構築された他から独立した仮想ネットワーク空間です。</p>
 <p>東京リージョン</p>	<p>リージョン(緑の点枠): データセンターが集まる物理的な地域です。</p>

 <p>Public subnet</p>	<p>サブネット(色付き四角): ネットワークを用途別に区切った部屋です。</p> <ul style="list-style-type: none"> ➤ Public: インターネットへの出入り口がある区画 (Webサーバーやセキュリティ機器など)。 ➤ Private: インターネットから直接アクセスできない安全な区画 (データベースや処理実行部など)。
 <p>Private subnet</p>	

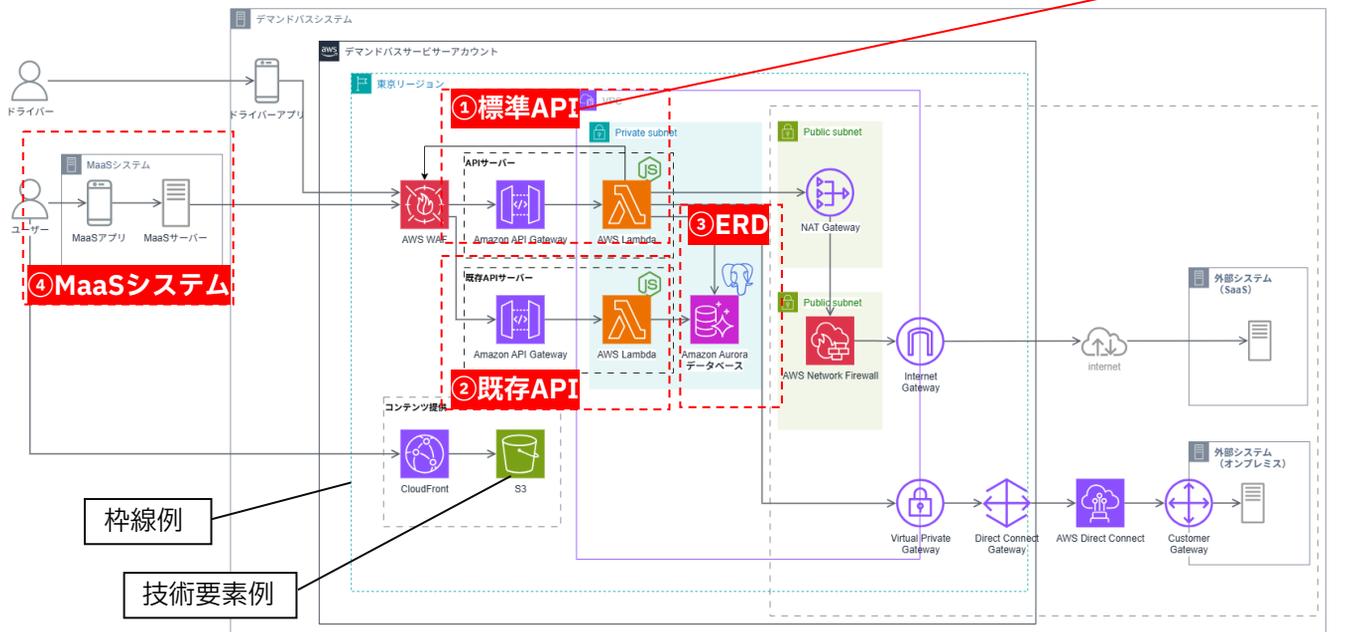
インフラ構成図の読み方

システムの解像度を「論理」から「実装」へと高めるプロセスに着目した図です。前述の論理構成図(Level2)を基礎に、具体的な実装例を用いて詳細化しています。

Level 2(Container)の例



インフラ構成図の例

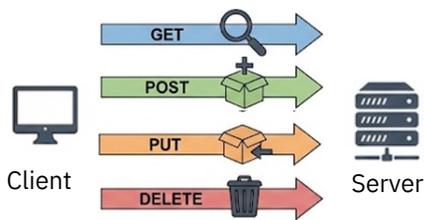


2.2. APIの読み方

API仕様は、RESTful APIのインターフェース定義標準であるOpenAPI 3.0によって記述されています。実装者は本定義に基づき、リソースへのアクセスパス、必須パラメータ、バリデーションルール、およびHTTPステータスコードによるエラーハンドリング仕様を読み解く必要があります。

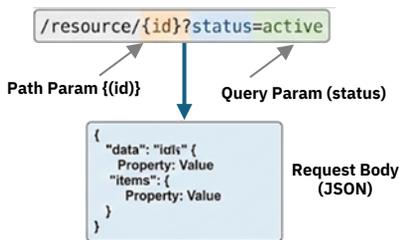
仕様定義の構造

Paths & Operations



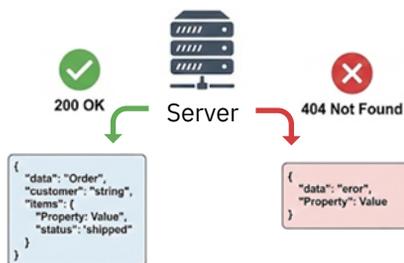
- Paths : サーバー上の「どのデータ」を操作するかを指定するURI(Uniform Resource Identifier)です。
- Operations : 実行する操作タイプです。
 - GET: リソースの取得 (副作用なし)。クエリパラメータでのフィルタリングが主。
 - POST: リソースの新規作成。リクエストボディに作成データを含めます。
 - PUT: リソースの置換・更新。ID指定でリソースの状態を変更します。
 - DELETE: リソースの削除。

Parameters & Request Body



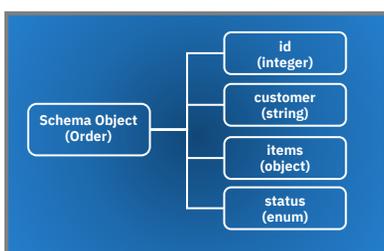
- Parameters (in): パラメータの格納場所を指定します。
 - path: URIの一部 (例: /resource/{id})。リソースの特定に使用。required: trueが必須。
 - query: URIの末尾 (例: ?status=active)。検索条件やソート指定に使用。
 - header: HTTPヘッダー (認証トークン等)。
- Request Body: POST/PUT時に送信するペイロードです。
 - content: メディアタイプを指定 (主に application/json)。
 - required: ボディ自体の必須有無。

Responses



- HTTP Status Codes: 処理結果をコードで分類します。
 - 200 OK / 201 Created: 正常終了。content内に返却データスキーマが定義されます。
 - 400 Bad Request: クライアント側の入力・形式エラー。
 - 401 Unauthorized: 認証失敗・未認証。
 - 404 Not Found: 指定リソースが存在しない。
 - 500 Internal Server Error: サーバー内部エラー。

Components & Schemas



- Data Types:
 - type: データ型 (integer, string, object..)
 - format: 型のフォーマット詳細。
 - enum: 列挙型。指定可能な固定値のリスト。
 - nullable: null値を許容するかどうか。
 - required: オブジェクト内で必須となるプロパティ名のリスト。

APIの読み方

APIを利用する際に参照する仕様書の構成を説明します。左メニューの目次から、中央のパラメータ詳細、右側の実行サンプルまで、どこにどのような情報が記載されているか、画面の見方をポイントを記載しています。

The screenshot shows the API documentation for the endpoint `POST /passengers/{passenger_id}/agreements`. The page is divided into several sections, each highlighted with a red dashed box and a numbered callout:

- 1**: Search bar and navigation menu.
- 2**: API title: 「ユーザーの規約同意を登録」.
- 3**: API summary description.
- 4**: Path parameters section.
- 5**: Request body schema section.
- 6**: Responses section, showing status codes 200 OK, 400 Bad Request, and 500 Internal Server Error.
- 7**: HTTP method and path: `POST /passengers/{passenger_id}/agreements`.
- 8**: Request samples section, showing a JSON payload.
- 9**: Response samples section, showing status codes 200, 400, and 500.

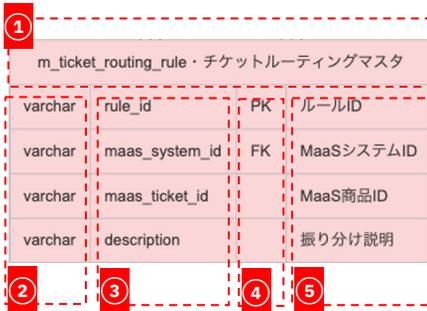
番号	エリア名称	説明
①	目次	APIのリソース（データの種類）やエンドポイントの一覧が表示されています。ここから閲覧したいAPIを選択したり、キーワードで検索したりするためのサイドバーです。
②	APIタイトル	選択しているAPIの機能名（例：「ユーザーの規約同意を登録」）が表示されています。
③	API概要説明	このAPIが具体的にどのような処理を行うか、どのような前提条件（制約事項）があるかといった仕様の詳細が記述されています。
④	パスパラメータ	URLの一部として指定する変数（例：passenger_id）の定義です。必須項目かどうかや、パラメータの説明が書かれています。
⑤	リクエストボディ定義	データ登録や更新時に送るデータの中身（スキーマ）の定義です。データの型（String, Arrayなど）や必須有無が記載されています。
⑥	レスポンス定義	このAPIが返すHTTPステータスコード（200, 400, 500など）の一覧です。クリックすると詳細が開く形式になっていることが多いです。
⑦	HTTPメソッドとパス	実際にリクエストを送る際のHTTPメソッド（GET, POST, PUT, DELETEなど）と、エンドポイントのURLパスが表示されています。
⑧	リクエストサンプル	リクエストを送信する際の具体的なデータの記述例（Payload）です。開発者がコピー＆ペーストして試せるようになっています。
⑨	レスポンスサンプル	処理結果として返ってくるデータ的具体例です。ステータスコード（200や400など）ごとの返却イメージを確認できます。

2.3. ERDの読み方

ERDは、システム開発やデータベース設計において、データの構造と関係性を可視化する図面です。その中で標準的に使われる「IE記法」は、線の端につく記号（鳥の足や丸印など）の組み合わせで、「1対1」「1対多」といったデータの関係（カーディナリティ）を表現します。

テーブル定義とIE記法

テーブル定義

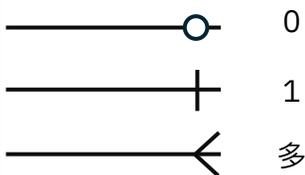


四角い箱は1つのデータ管理単位(テーブル)を表します。左から順に以下の要素で構成されています。

- ① テーブル名(m_ticket_routing_rule)
 - データベース上の物理テーブル名(英字)と、その内容を表す論理名(日本語)が併記されています。
- ② データ型(varchar等)
 - データの種類とサイズ。
- ③ 物理名(user_id等)
 - システム(データベース)上で実際に使われるアルファベットの列名。
- ④ キー(PK, FK)
 - PK(Primary Key/主キー): その行を特定するための唯一のID。重複しない。
 - FK(Foreign Key/外部キー): 他のテーブルと紐付くためのID。
- ⑤ 論理名(ユーザーID等)
 - 人間が理解しやすいように付けられた日本語の項目名。

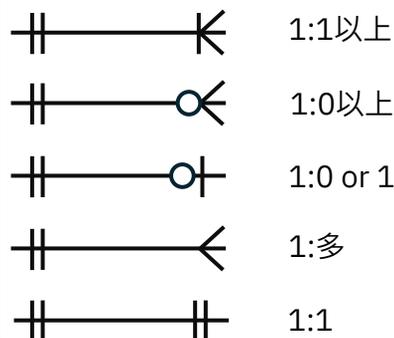
IE記法

基本表記



- 基本となる3つの表記
以下の3つの基本形状の組み合わせで成り立っています。
丸(○) : 0 (ゼロ / 存在しない可能性がある)
縦棒(|) : 1 (イチ / 単一)
鳥の足(<) : 多 (タ / 複数)
- オプションリティを含んだ表記
オプションリティを含んだ表記では、以下の意味を有します。

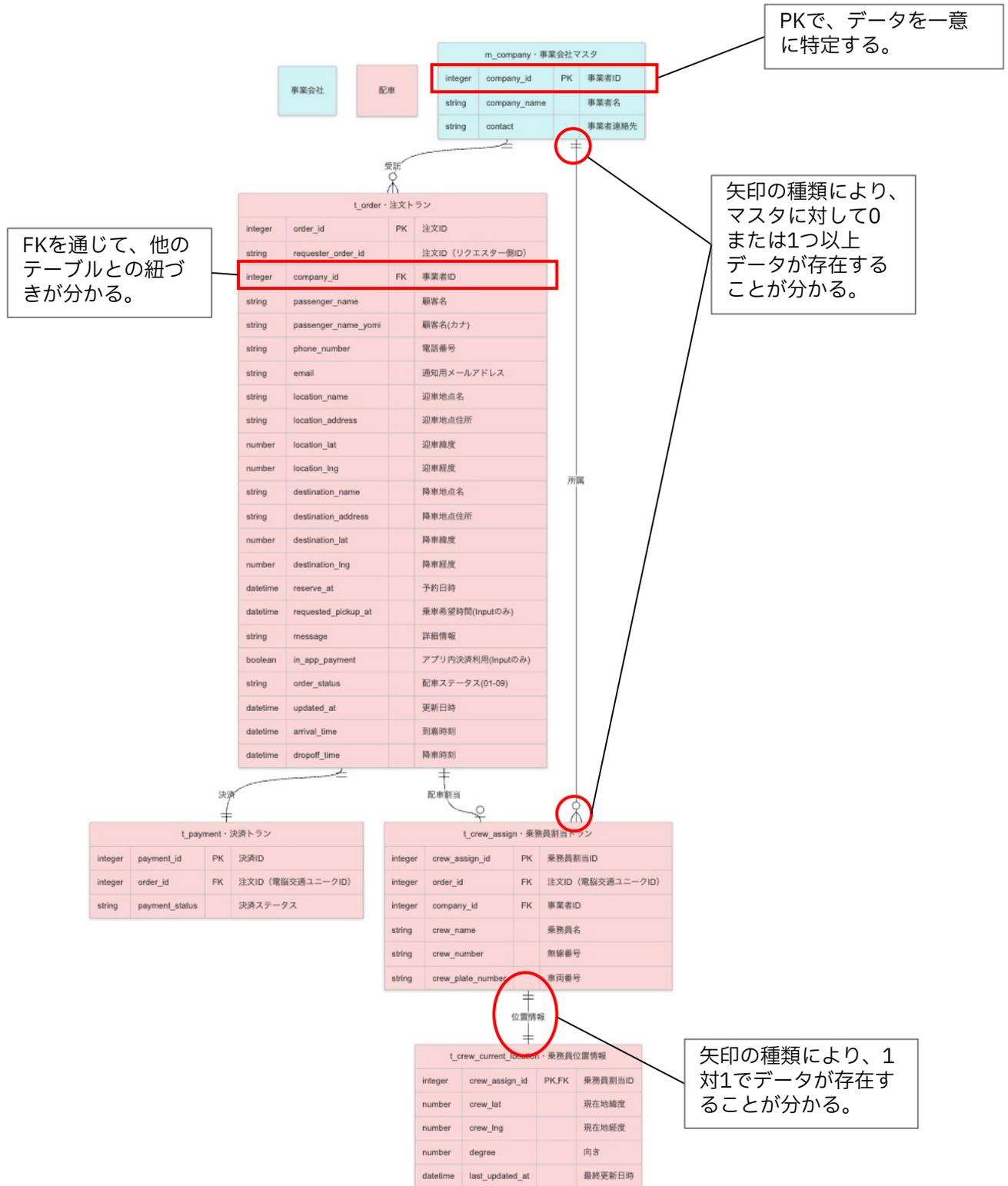
オプション表記



名称	意味・解説
1:1以上	「少なくとも1つは必ず必要で、複数あってもよい」 (例: 注文には、必ず1つ以上の商品明細が必要)
1:0以上	「なくてもよいが、複数あるかもしれない」 (例: 会員は、注文履歴が0件かもしれないし、多数あるかもしれない)
1:0 or 1	「なくてもよいが、あるとしても1つだけ」 (例: 社員に対して、社用車は0台か、割り当てられても1台)
1:多	「多数」 ※これは少し簡略化された表記で、通常は上の「0以上」か「1以上」を明確にして使います。
1:1	「必ず1つだけ存在する」 (例: 注文明細には、必ず1つの商品情報が紐づく)

ERDの読み方

ERDはシステムが扱うデータのつながりを表す設計図です。四角い箱（テーブル）は注文や事業者ごとのデータの棚であり、中に管理項目が定義されています。箱を結ぶ線はデータ同士の関係性（リレーション）を示し、線の端の記号で「1対1」や「1対多」といった結びつきのルールを表現しています。中心となる箱から線を辿ることで、業務における情報の流れや依存関係を読み解くことができます。



2.4. 業務フロー/業務一覧の読み方

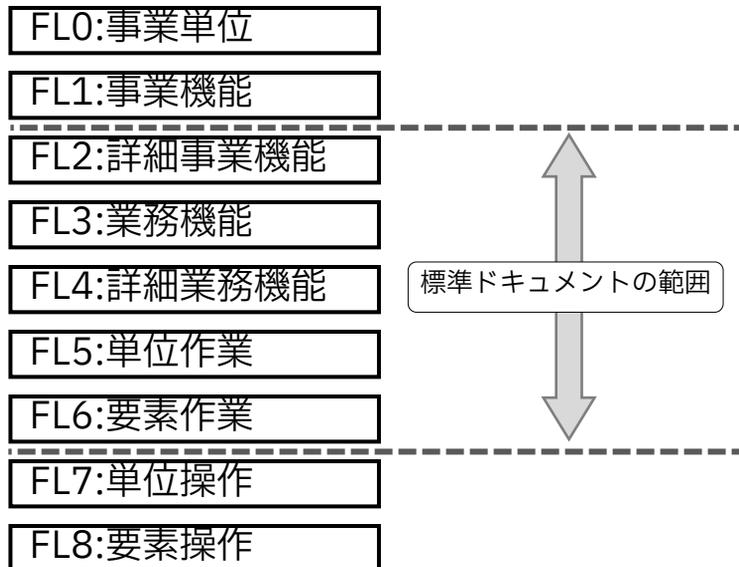
業務フロー/業務一覧は、標準API仕様に基づくシステム実装を運用するための業務モデルの指針です。業務をファンクションレイヤー（以下、FLとする）定義に従って「事業」から「要素作業」まで階層化し、標準的な粒度で整理します。その上で、BPMN 2.0表記法に基づきシステムと人の動きを可視化します。これにより業務要件とAPIの整合性を確保し、システムを運用するための具体的な業務手順を理解することができます。

ファンクションレイヤーによる業務の階層化

業務を階層化して整理する理由は、読み手の役割によって必要な「情報の解像度」が異なるためです。

例えば、全体像を把握したい管理者層には「大まかな業務の流れ」があれば十分ですが、システムを実装する開発者には「具体的な処理手順」が必要です。

本ドキュメントでは、双方の視点をカバーし、ビジネスの全体理解から実際のシステム設計までスムーズにつなげるために、適切な粒度（FL2～FL6）に分けて定義しています。



FLによる業務分解例

機能階層	具体例	想定利用者	利用目的
FL0	自動車保険事業	経営層	グループ全体の事業管理
FL1	個人向け販売	事業責任者	バリューチェーンの構築
FL2	ディーラー経由販売	部門マネージャー	ビジネスモデル・チャネル設計
FL3	見積り依頼～契約	プロセスオーナー	業務サイクルの管理・KPI設定
FL4	見積り依頼・価格提示	チームリーダー	部署間の連携・進捗管理
FL5	見積り依頼受付	実務担当者	自身のタスク・責任範囲の確認
FL6	依頼書開封・チェック	業務改善担当	ツール選定・工数（時間）分析
FL7	依頼書ファイル開く	RPA開発者	自動化手順の設計
FL8	ファイルメニュークリック	RPA開発者	ロボットの具体的な動作指定

業務一覧の読み方

業務一覧は、業務全体を大きな塊（FL2）から具体的な作業手順（FL6）へと段階的に分解し、整理したものです。「左から右へ」視線を動かすことで、業務の解像度が高まる構造になっています。また、APIを利用する業務については、APIのIDを割り振り、APIの資料と対応付けて確認できるようになっています。

FLごとにIDと名称を記載。

No	FL0		FL2		FL3		FL4		FL5		FL6	
	ID	事業単位	ID	詳細事業機能	ID	業務機能	ID	詳細業務機能	ID	単位作業	ID	要素作業
1	M13	QRチケット事業	M13-FL2-010	商品企画								
2	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画						
3	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-010	商品企画立案				
4	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-010	商品企画立案	M13-FL5-010	商品企画検討		
5	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-010	商品企画立案	M13-FL5-010	商品企画検討	M13-FL6-010	商品企画検討
6	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-010	商品企画立案	M13-FL5-020	商品企画承認		
7	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-010	商品企画立案	M13-FL5-020	商品企画承認	M13-FL6-020	商品企画承認
8	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-020	商品企画確認				
9	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-020	商品企画確認	M13-FL5-030	商品企画確認		
10	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-020	商品企画確認	M13-FL5-030	商品企画確認	M13-FL6-030	商品企画確認
11	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-020	商品企画確認	M13-FL5-040	商品企画書受領		
12	M13	QRチケット事業	M13-FL2-010	商品企画	M13-FL3-010	商品企画	M13-FL4-020	商品企画確認	M13-FL5-040	商品企画書受領	M13-FL6-040	商品企画書受領

FL4以下の業務についてアクターを表現。

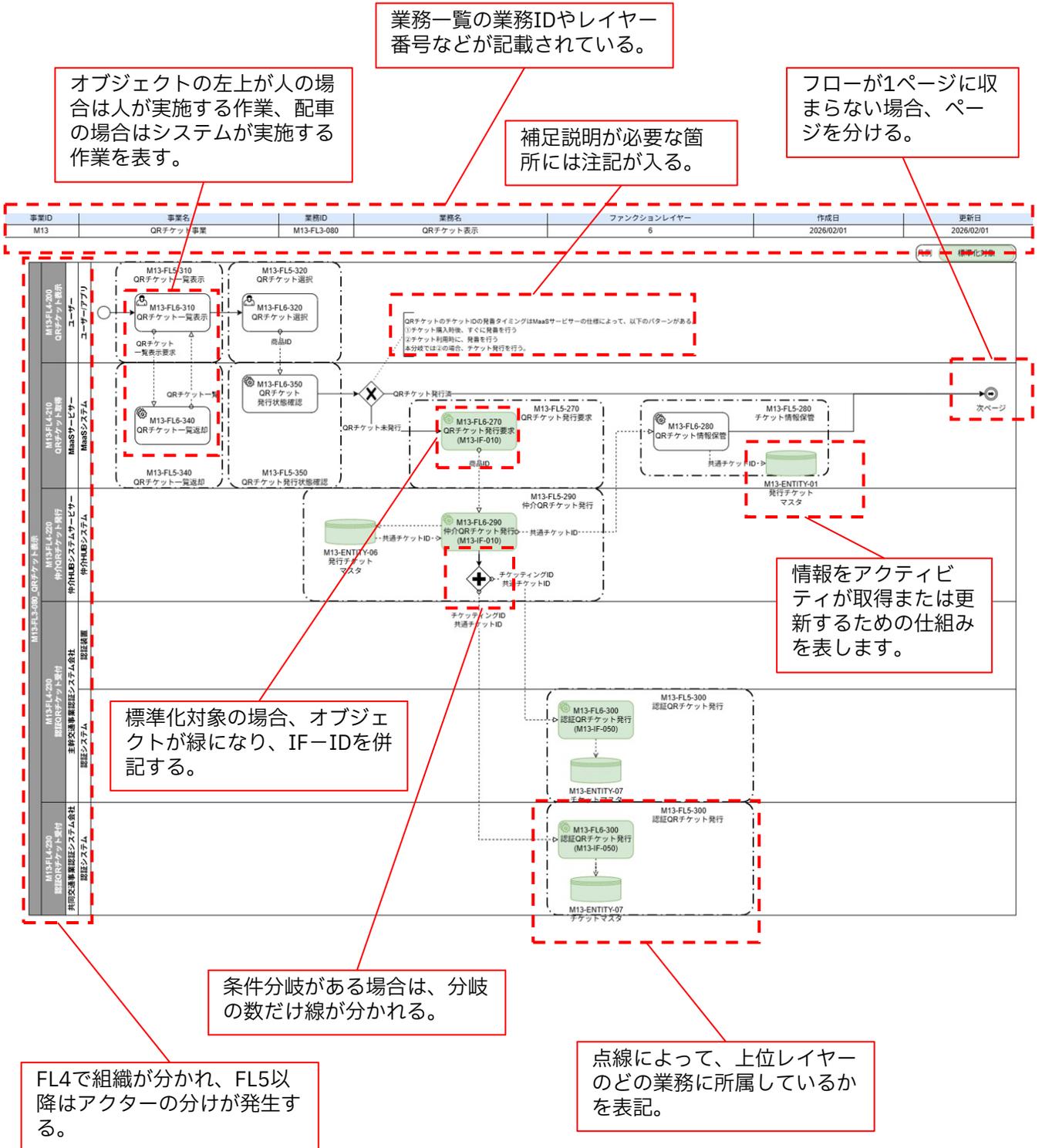
各業務の詳細を説明。

アクター	システム	API		目的・内容
		ID	名称	
				交通事業者、MaaSサービス、認証システム間で共通のQRチケットのサービスを提供するにあたり、商品を決めるための業務。
				共通のQRチケットを利用する事業者間で、商品情報(商品名、販売金額など)を取り決める業務プロセス。
				主管交通事業者が、商品企画を検討し、共同交通事業者と同意を行う。
主管交通事業者				主管交通事業者の規格車が商品企画の検討を行い、共同交通事業者へ確認をしてもらう。
				主管交通事業者の規格車が商品企画の検討を行い、共同交通事業者へ確認をってもらう。
				主管交通事業者の承認者が、担当者から受領した商品企画の承認を行う。
主管交通事業者				主管交通事業者の承認者が、担当者から受領した商品企画の承認を行う。承認した企画書は共同交通事業者へ送信する。
				共同交通事業者が、主管交通事業者から受領した商品企画の確認を行う。
				共同交通事業者の担当者が、主管交通事業者から受け取った商品企画の確認を行い、合意を行う。
主管交通事業者				共同交通事業者の担当者が、主管交通事業者から受け取った商品企画の確認を行い、合意を行う。
				共同交通事業者の担当者が、主管交通事業者から企画書を受領する。
主管交通事業者	○	M13-IF-010	商品企画書受領API	共同交通事業者の担当者が、主管交通事業者から企画書を受領する。

APIを扱う業務について、IDと名称を記載。

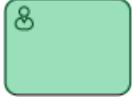
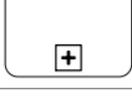
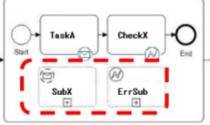
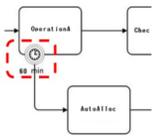
業務フローの読み方

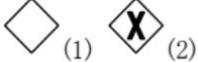
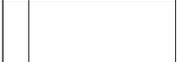
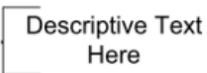
業務フローは、業務一覧で細分化した業務を、階層ごとの流れとしてまとめた資料です。視線を左から右へ動かすことで、大枠の「詳細事業機能 (FL2)」から、詳細な「要素作業 (FL6)」へと段階的に具体化される構造になっています。



BPMN 2.0 に基づく業務フロー記述ルール

本標準ドキュメントでは、「業務一覧」で整理したFL構造を、BPMN2.0のルールに従って図式化します。本標準ドキュメントで使用するBPMN2.0のオブジェクトは以下の種類があります。

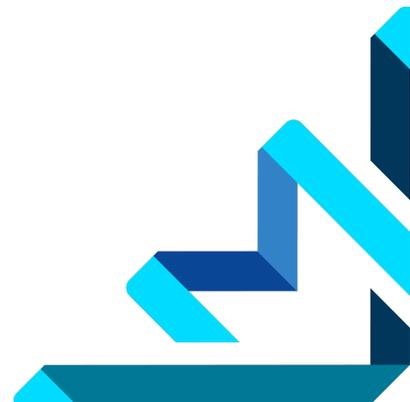
No	分類	図形名称	説明	図形
1	アクティビティ	ユーザータスク (User Task)	ユーザータスクは、BPMNエンジンと連携してユーザーが実行する業務処理を表す。	
2	アクティビティ	緑のユーザータスク (Service Task)	緑のユーザータスクは、サービスタスクの中でも本プロジェクトの標準化対象のものを表す。	
3	アクティビティ	サービスタスク (Service Task)	サービスタスクは、Webサービスのような人間の介入がないアプリケーションによって実行される業務処理を表す。	
4	アクティビティ	緑のサービスタスク (Service Task)	緑のサービスタスクは、サービスタスクの中でも本プロジェクトの標準化対象のものを表す。	
5	アクティビティ・ マーカ	サブプロセス (Sub-Process)	サブプロセスは、親ビジネスプロセスに組み込まれた下位のビジネスプロセスを表す。	
6	アクティビティ・ その他	イベント・サブプロセス (Event Sub-Process)	イベント・サブプロセスはプロセスの枠線を点線で表す(※図中の点線で囲った部分)。 イベント・サブプロセスは、ビジネスプロセス内に配置され、そのビジネスプロセスで発生したイベントをトリガとして実行されるサブプロセスである。	
7	イベント	イベント(イベントタイプなし) (None Event) (Start Event / End Event)	<p>■ Start 名前が示すとおり、開始イベント (Start Event) は、特定のプロセスまたはコレオグラフィーがどこから始まるかを示します。</p> <p>■ End 名前が示すとおり、終了イベント (End Event) は、プロセスまたはコレオグラフィーがどこで終わるかを示します。</p>	
8	イベント	境界イベント (Boundary Event)	<p>タイマー・トリガの境界イベントの例(※図中の点線で囲った部分)。</p> <p>境界イベントは、タスク、サブプロセス及びコールアクティビティ内でメッセージ、エラー、タイマーなどのイベントが発生した場合に、境界イベントに接続するフローにトークンが流れる。</p> <p>■ 中断イベントの場合は、タスク、サブプロセス及びコールアクティビティの実行を中断し、トークンは境界イベントに接続するフローに流れる。</p> <p>■ 非中断イベントの場合は、タスク、サブプロセス及びコールアクティビティの実行を中断せず、後続のフローが継続されると共に、トークンは境界イベントに接続するフローに流れる。</p>	
9	イベント	エラー (Error Event)	<p>エスカレーション・イベント・サブプロセスは、ビジネス・アクティビティが実行上の制約 (時間ベースの締め切りなど) を満たさない場合に、その完了を迅速化するための手段を実行します。</p> <p>エスカレーション開始イベントは、インライン・イベント・サブプロセスをトリガーするためのみに許可されます。</p>	
10	イベント	リンク (Link Event)	エラーイベントは、アクティビティでエラーが発生した場合に処理を中断し、エラー処理に接続するイベントを表す。	

No	分類	図形名称	説明	図形
11	イベント	強制終了 (Terminate Event)	強制終了イベントは、ビジネスプロセスの即時停止を表す。同一ビジネスプロセス内の全ての実行中タスクを強制終了し、当該ビジネスプロセスをただちに終了させる。ただし、サブプロセスで使用されている場合、当該サブプロセスを終了し、親ビジネスプロセスに戻る。	
12	ゲートウェイ	排他ゲートウェイ (Exclusive Gateway)	排他ゲートウェイは、フローが排他的に分岐すること又は分岐したフローが結合することを表す。実行可能モデルにおける排他ゲートウェイは、分岐条件情報の設定条件で評価しフローの分岐を行う。 (1)、(2)のいずれかを使用するかは、製品で対応している図形を使用し、両方に対応している場合は(2)を使用する。	
13	ゲートウェイ	並列ゲートウェイ (Parallel Gateway)	並列ゲートウェイは、フローが並列に分岐しその後のフローで並列処理されていたフローが同期的に結合することを表す。	
14	接続オブジェクト	シーケンスフロー (Sequence Flow)	シーケンスフローは、タスク、イベント、ゲートウェイ間の処理の流れを表す。	
15	接続オブジェクト	方向性の関連、 管理データへの関係 (Directional Association, Directed Data Association)	データの関連付けは、タスクとデータの関連性、タスクと管理データへの関係を図示する。	
16	データ	データオブジェクト (Data Object)	データオブジェクトは、アクティビティの入出力要素として表す。 データオブジェクトは、実行モデルの処理フローに影響しない。	
17	データ	データストア (Data Store)	データストアは、個別DB、共有DBを表す。 データストアは、実行モデルの処理フローに影響しない。	
18	データ	緑のデータストア (Data Store)	緑のデータストアは、サービスタスクの中でも本プロジェクトの標準化対象のものを表す。	
19	スイムレーン	プール (Pool)	プールは、一連のビジネスプロセスを記述する範囲を表す。プールには、一意なビジネスプロセス識別子を指定する。	
20	スイムレーン	レーン (Lanes)	レーンは、タスクを実行するユーザーや部署の範囲を表す。	
21	その他	グループ (Group)	グループは、要素のセットを強調表示する。 制約や規則を課すものではない。	
22	その他	テキスト注釈 (Text Annotation)	テキスト注釈は、プロセス又はその要素に関する補足情報(コメント)を表す。ダイアグラム上の任意の場所に配置することができ、どの要素にもアタッチすることができる。	



3. 標準ドキュメントの解説

- 3.1. 標準化のスコープと概要
- 3.2. システムアーキテクチャの解説
- 3.3. APIの解説
- 3.4. ERDの解説
- 3.5. 業務フロー/業務一覧の解説(一部抜粋)



3.1. 標準化のスコープと概要

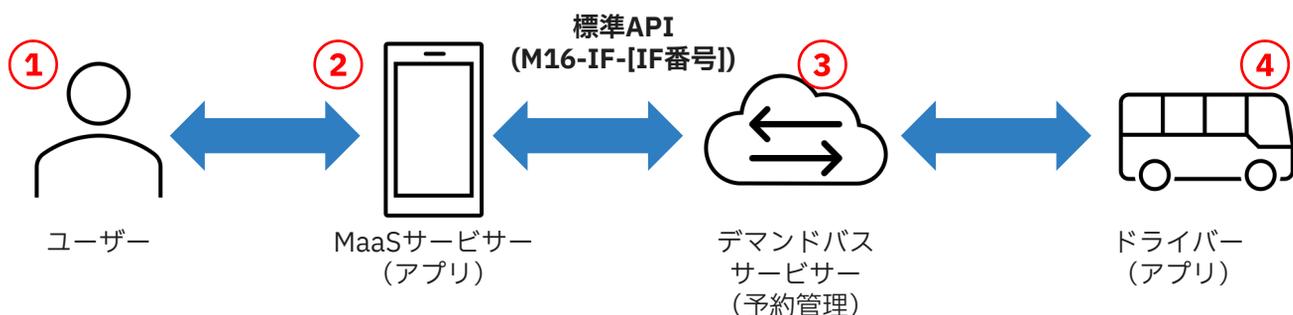
本章で解説する標準化のスコープと概要を解説します。

標準化のスコープ

本プロジェクトでは、MaaSアプリとデマンドバスシステム間におけるシステム連携の効率化と相互運用性の確保を目的とし、「標準API仕様」および「標準業務モデル」を標準ドキュメントとして策定します。

下図は、本プロジェクトが対象とするシステム範囲（境界）と、システム連携に関与する各プレイヤー（アクター）の関係性を示したものです。

MaaSシステムとデマンドバスシステムを疎結合に接続する標準API仕様を定義することで、ベンダーの実装に依存しない利便性の高い交通サービスの実現を目指します。



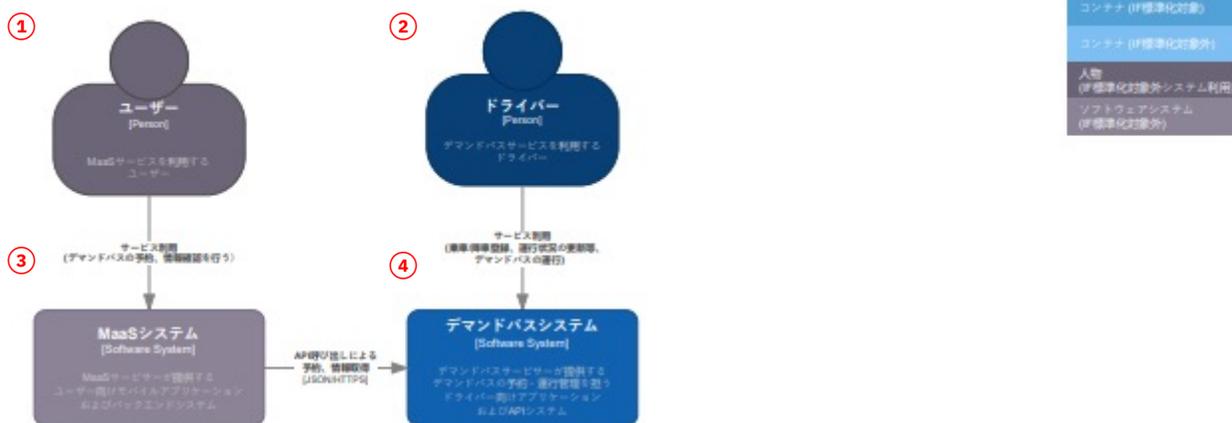
番号	アクター	役割
①	ユーザー	デマンドバスサービスの利用者。 MaaSアプリ（MaaSサービサー提供）を利用して、乗降場所の検索、デマンドバスの予約・変更・キャンセル、および運行状況の確認を行います。
②	MaaSサービサー（アプリ）	ユーザーとの直接的な接点（UI）を提供する事業者システム。 ユーザーからのリクエストを受け付け、標準APIを通じてデマンドバスサービサー（③）へ予約照会や登録依頼を行います。また、ユーザーへ予約結果や運行状況を通知・表示します。
③	デマンドバスサービサー（予約管理）	デマンドバスの運行計画、配車計算、予約台帳を管理する基盤システム。 標準APIを用いてMaaSサービサーからのリクエストを処理し、予約可否の判定や確定情報の登録を行います。 また、ドライバーや車両の運行状況を管理します。
④	ドライバー（アプリ）	デマンドバスの運行を担当する乗務員および車載システム。 デマンドバスシステムから配信される運行指示（送迎ルートや乗客情報）に基づき業務を遂行します。現地での乗車・降車実績や運行状況をシステムへ報告します。

3.2. システムアーキテクチャの解説

システムアーキテクチャのドキュメントを掲載します。

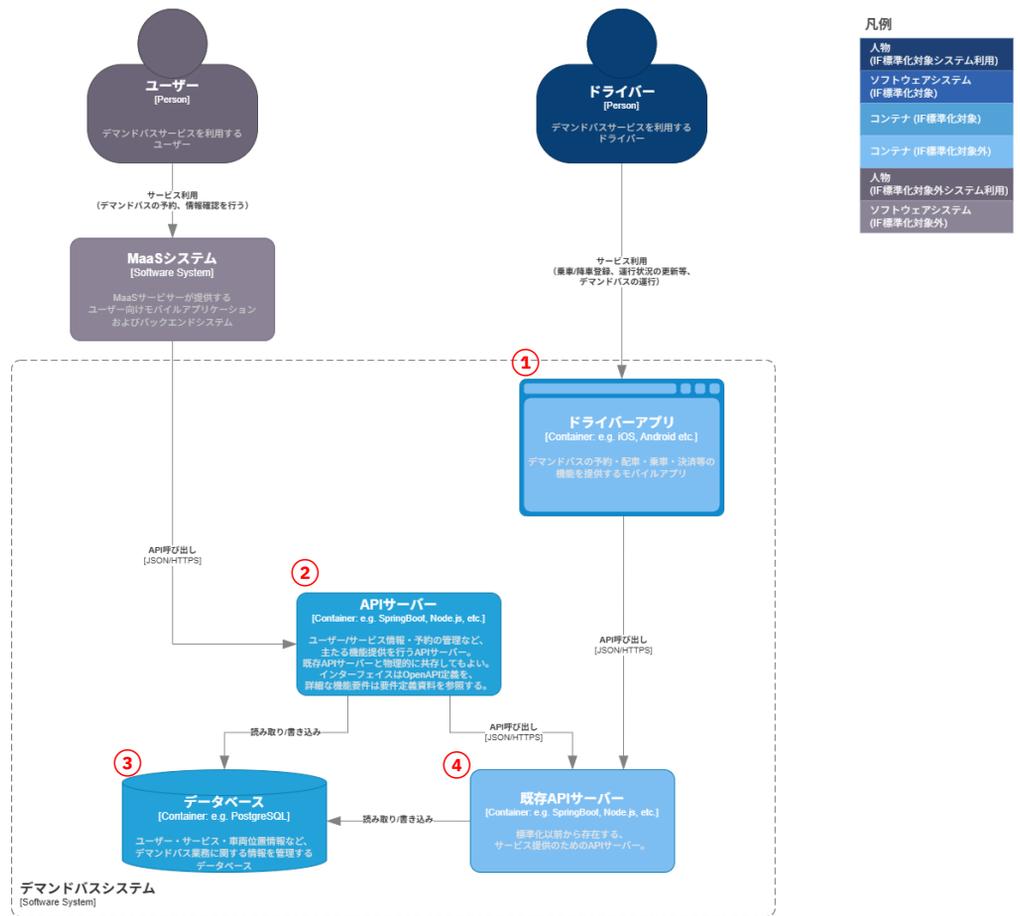
論理構成図

Level1: System Context



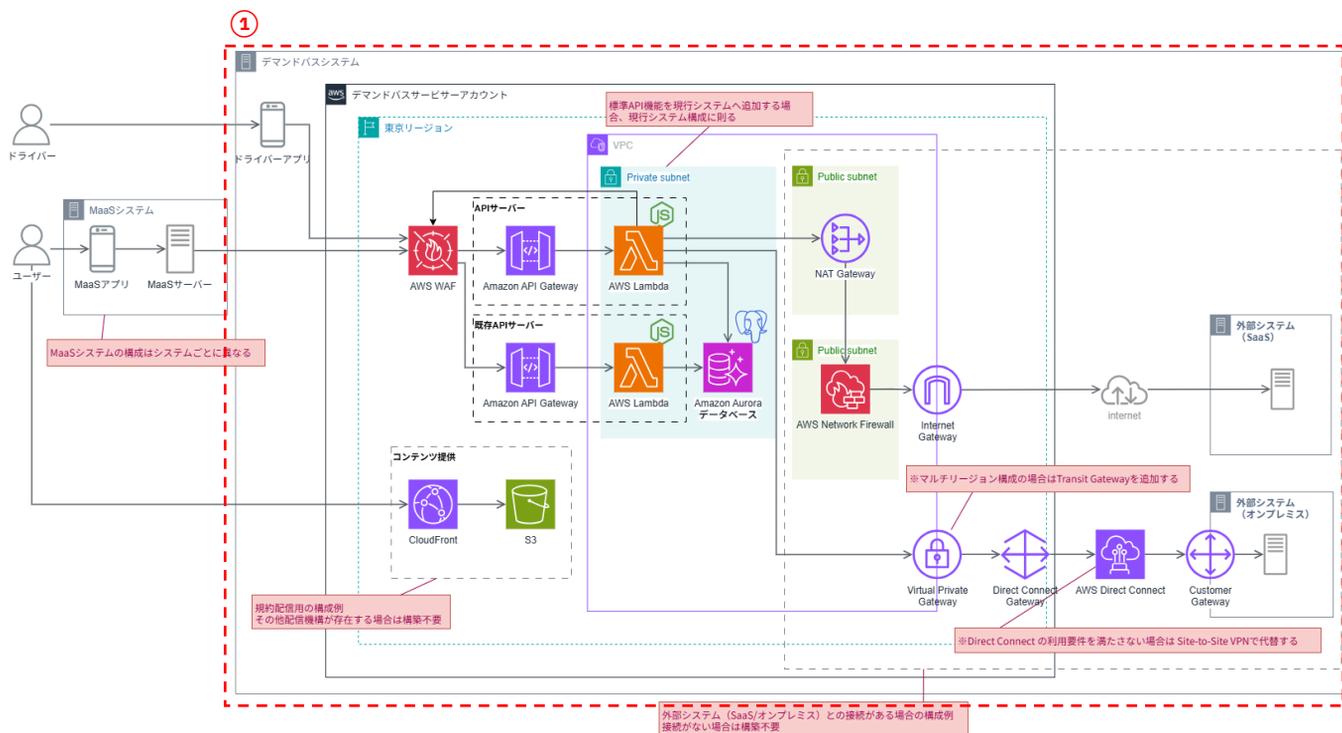
番号	名称	カテゴリ	説明
①	ユーザー	人物 (Person)	MaaSサービスを利用するエンドユーザー。デマンドバスの予約や情報確認を行う主体。
②	ドライバー	人物 (Person)	デマンドバスサービスを運行する担当者。乗降登録や運行状況の更新などを行う主体。
③	MaaSシステム	ソフトウェアシステム	MaaSサービスが提供する、ユーザー向けモバイルアプリケーションおよびバックエンドシステム。
④	デマンドバスシステム	ソフトウェアシステム	デマンドバスの予約・運行管理を担う、ドライバー向けアプリおよびAPIシステム。

Level2: Container



番号	名称	所属システム	説明
①	ドライバーアプリ	デマンドバスシステム	ドライバーが使用するモバイルアプリ (iOS/Android等)。予約・配車・乗車・決済などの機能を提供し、APIサーバーや既存APIサーバーと連携する。
②	APIサーバー	デマンドバスシステム	ユーザー/サービス情報、予約管理などの主要機能を提供するサーバー。インターフェースはOpenAPIで定義され、既存APIサーバーと物理的に併存する場合もある。
③	データベース	デマンドバスシステム	ユーザー、サービス、車両位置情報、デマンドバス業務に関する情報を一元管理するデータストレージ (PostgreSQL等)。
④	既存APIサーバー	デマンドバスシステム	IF標準化以前から存在する、サービス提供のためのAPIサーバー。新設のAPIサーバーやドライバーアプリからの呼び出しに対応する。

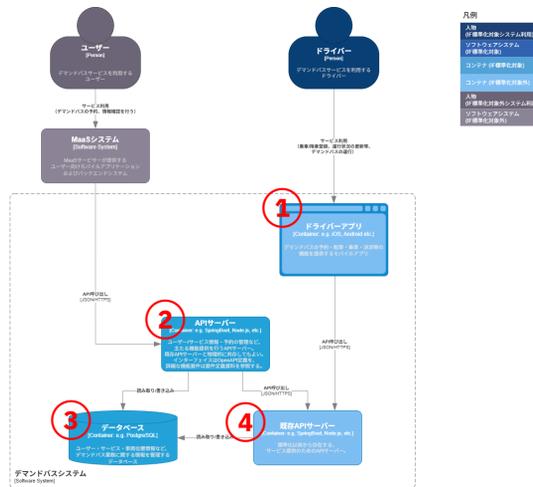
インフラ構成図(実装例)



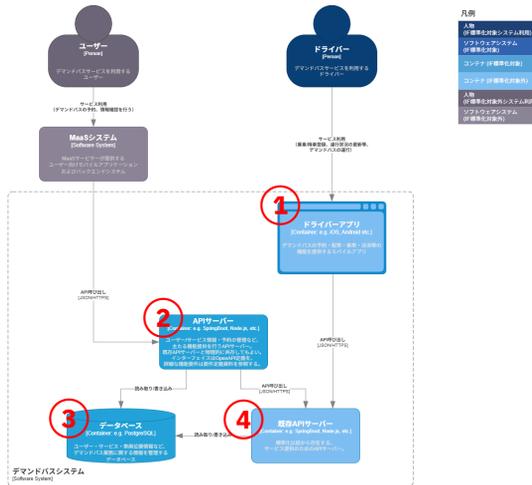
番号	システム領域名	詳細説明
①	デマンドバスシステム	<p>【APIの受付とセキュリティ】</p> <ul style="list-style-type: none"> • AWS WAF: 外部（MaaSシステムやドライバーアプリ）からの不正なリクエストを遮断し、Webアプリケーションを保護します。 • Amazon API Gateway: リクエストの入り口として機能し、認証やトラフィックの制御を行います。 <p>【ビジネスロジックの実行】</p> <ul style="list-style-type: none"> • AWS Lambda: 「APIサーバー」および「既存APIサーバー」の実体であり、予約管理や配車計算などのサーバーレスなプログラム実行を担います。 <p>【データの保存と管理】</p> <ul style="list-style-type: none"> • Amazon Aurora: 「データベース」の実体として、ユーザー情報、サービス情報、車両位置情報などを一元管理します。 <p>【静的コンテンツの配信】</p> <ul style="list-style-type: none"> • Amazon CloudFront & S3: 規約ドキュメントやアプリ用の画像ファイルなど、頻繁に更新されないデータを効率的に配信します。 <p>ネットワーク保護と外部連携</p> <ul style="list-style-type: none"> • Private Subnet / NAT Gateway: データベースなどの重要リソースをインターネットから隔離された環境に配置しつつ、必要な時だけ外部へ通信を確立します。 • AWS Network Firewall: VPC全体のネットワークトラフィックを精査し、セキュリティを強化します。 • Direct Connect / VPN: 外部のSaaSやオンプレミスシステムと、専用線または暗号化された経路で安全に接続します。

3.3. APIの解説

APIの一覧を掲載します。



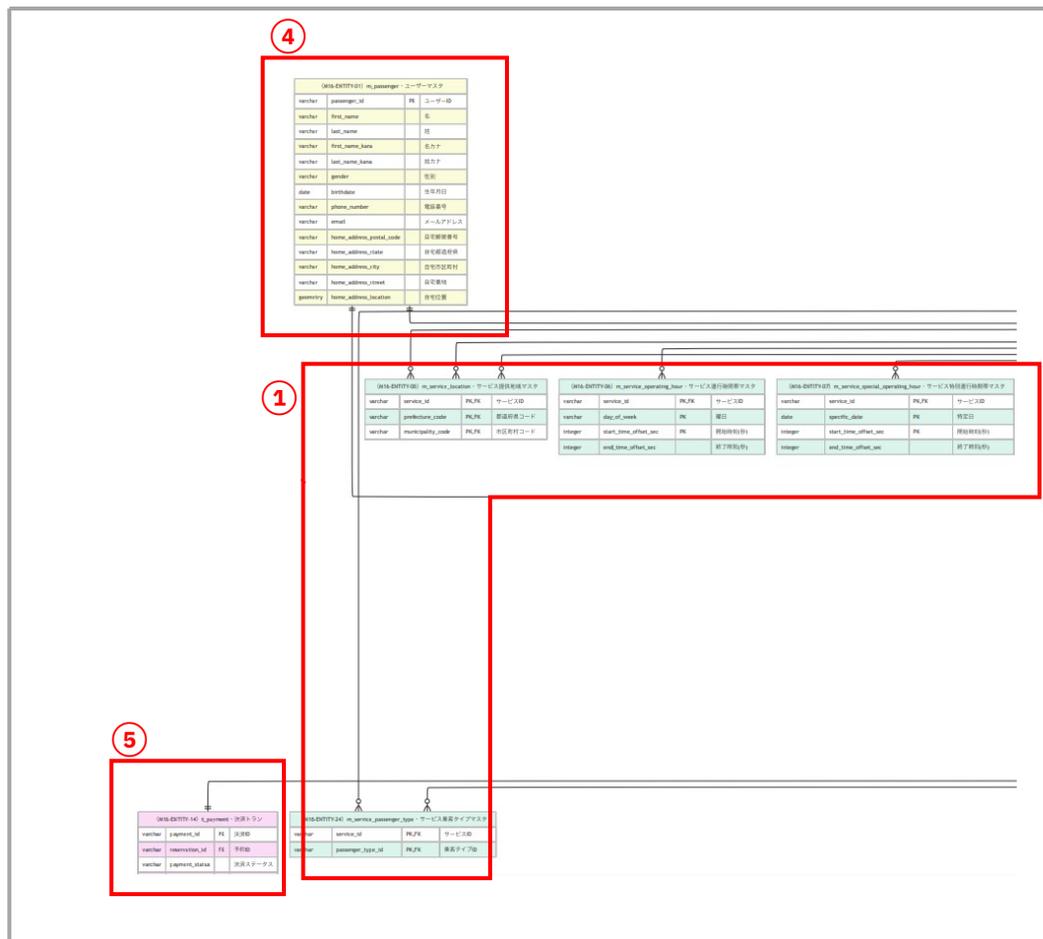
APIサーバー	API名	API使用場所	機能説明	目的
デマンドバスシステム	(M16-IF-010) プラットフォーム規約一覧を取得	MaaSシステム → ②	プラットフォーム全体に適用される各種規約の最新一覧を取得します。	ユーザーが利用開始前に共通規約を確認・同意できるようにするため。
	(M16-IF-020) サービスの概要情報一覧を取得	MaaSシステム → ②	デマンドバスサービスの概要情報を取得します。	利用可能なバスサービスを一覧表示し、ユーザーが選択できるようにするため。
	(M16-IF-030) サービスの詳細情報を取得	MaaSシステム → ②	指定されたサービスIDに対応する、デマンドバスサービスの詳細情報を取得します。	運行エリアや予約ルールなど、特定のサービスの詳細な仕様を表示するため。
	(M16-IF-040) ユーザー情報を新規登録	MaaSシステム → ②	ユーザー情報を新規登録します。	初めてサービスを利用するユーザーのアカウントを作成するため。
	(M16-IF-050) ユーザー情報を取得	MaaSシステム → ②	指定されたユーザーIDに紐づくユーザー情報を取得します。	プロフィール画面などで現在の登録内容を表示・確認するため。
	(M16-IF-060) ユーザー情報を更新	MaaSシステム → ②	指定されたユーザーIDに対して、ユーザー情報を更新します。	氏名や連絡先などのユーザー登録情報を変更するため。
	(M16-IF-070) ユーザー情報を削除 (同意情報含む)	MaaSシステム → ②	指定されたユーザーIDに紐づくユーザー情報を削除します。	退会処理に伴い、個人情報および規約同意状況を完全に抹消するため。
	(M16-IF-080) ユーザーが利用可能なサービスの概要情報一覧を取得	MaaSシステム → ②	指定されたユーザーIDに紐づくユーザーが利用可能なサービス情報を取得します。	ユーザーが利用できる「住民限定」や「登録ユーザー限定」などのサービス情報を取得し、予約資格の確認を行うため。
	(M16-IF-090) ユーザーの規約同意を登録	MaaSシステム → ②	指定されたユーザーIDに対して、利用規約 (terms_id) への新規同意を登録します。	規約更新時などに、ユーザーが内容を承諾したことを記録するため。
	(M16-IF-100) ユーザーの規約同意一覧を取得	MaaSシステム → ②	指定されたユーザーIDが示すユーザーが、過去に同意した利用規約の一覧を取得します。	再同意が必要な規約があるかを確認するため。
	(M16-IF-110) 乗降場所一覧を取得	MaaSシステム → ②	デマンドバスの固定の乗降場所情報を取得します。	地図上やリストから出発地・目的地 (バス停) を選択できるようにするため。



APIサーバー	API名	API使用場所	機能説明	目的
デマンドバスシステム	(M16-IF-120) 予約候補を生成	MaaSシステム → ②	ユーザーが指定した条件をもとに、予約可能な便候補をリアルタイムに生成します。	予約確定前に、実際に乗車可能な時間や運賃をユーザーに提示するため。
	(M16-IF-130) 予約を登録	MaaSシステム → ②	事前の検索によって取得された予約便候補を用いて、予約を登録します。	ユーザーが選択した特定の便で予約を確定させるため。
	(M16-IF-140) ユーザーの予約一覧を取得	MaaSシステム → ②	パスパラメータで指定したユーザーが登録した予約情報を複数件取得します。	ユーザーが自分の過去の利用履歴や今後の予定をアプリで確認するため。
	(M16-IF-160) 予約を更新	MaaSシステム → ② ① → ②	登録済みの予約情報を予約ID指定で更新（人数変更やキャンセルなど）します。	ユーザーによる予約キャンセルや、人数変更などの要望に対応するため。
	(M16-IF-170) 予約の決済情報を取得	MaaSシステム → ② ① → ②	登録済みの予約に対する決済情報を取得します。	運賃の支払状況（受領済みか等）を確認するため。
	(M16-IF-180) 予約の決済情報を登録または更新	① → ②	指定された予約に対して、決済金額および決済ステータスを登録または更新します。	乗車時に確定した運賃の登録や、決済完了の記録を行うため。
	(M16-IF-210) 車両の位置情報を取得	MaaSシステム → ② ① → ②	車両の現在位置情報を取得します。	ユーザーへの接近通知や、地図上でのリアルタイムな車両位置表示のため。
	(M16-IF-220) 運行遅延情報を取得	MaaSシステム → ②	運行遅延情報を取得します。	渋滞や事故による遅延をユーザーに知らせ、利便性を確保するため。

3.4. ERDの解説

ERDのドキュメントを掲載します。



1

[表名] m_service サービス		
varchar	service_id	PK サービスID
varchar	name	名称
varchar	description	説明
varchar	brand_image_url	ブランドイメージURL
varchar	operation_type	操作方法
timestamp	start_datetime	サービス開始日時
timestamp	end_datetime	サービス終了日時
varchar	service_group_id	サービスグループID
boolean	time_specifiable_department	店舗別指定可能フラグ
boolean	time_specifiable_area	店舗別指定可能フラグ
integer	max_advance_reservation_days	最大予約可能日数
integer	min_advance_reservation_months	最低予約月数
integer	cancel_advance_reservation	キャンセル可能
integer	search_advance_reservation	最大検索可能回数
boolean	front_shop_reserveable	店頭予約可能フラグ
boolean	home_reserveable	自宅予約可能フラグ
boolean	custom_location_reserveable	任意地予約可能フラグ

[表名] m_prefecture 都道府県マスタ		
varchar	prefecture_code	都道府県コード
varchar	prefecture_name	都道府県名

[表名] m_municipality 市区町村マスタ		
varchar	municipality_code	市区町村コード
varchar	municipality_name	市区町村名

[表名] m_reserveable_area 予約エリアマスタ		
varchar	name	エリア名称
varchar	description	エリア説明
varchar	year	エリア年度

[表名] m_reservation_department サービス別予約マスタ		
varchar	reservation_id	予約ID
varchar	service_id	サービスID
varchar	title	タイトル
text	content	内容
timestamp	start_datetime	予約可能開始日時
timestamp	end_datetime	予約可能終了日時

[表名] m_reserveable_area サービス別予約エリアマスタ		
varchar	service_id	PK サービスID
varchar	reserveable_area_id	PK エリアID

[表名] m_reservation_area 予約エリアマスタ		
varchar	reservation_area_id	PK 予約エリアID
varchar	service_id	PK サービスID
varchar	from_area_id	PK 出発エリアID
varchar	to_area_id	PK 到着エリアID
boolean	is_allroad	移動可能フラグ

[表名] m_reservation_area 予約エリアマスタ		
varchar	service_id	PK サービスID
varchar	from_area_id	PK 出発エリアID
varchar	to_area_id	PK 到着エリアID
boolean	is_allroad	移動可能フラグ

5

[表名] m_reservation_area 予約エリアマスタ		
varchar	reservation_id	PK 予約ID
varchar	passenger_id	PK ユーザーID
varchar	vehicle_id	PK 車両ID
varchar	status	予約ステータス
integer	fare_total	乗車料合計
integer	max_distance	最大乗車距離
varchar	pickup_type	乗車場所種別
varchar	dropoff_type	降車場所種別
varchar	pickup_area_id	PK 乗車エリアID
varchar	dropoff_area_id	PK 降車エリアID
varchar	pickup_display_name	乗車場所名称
varchar	dropoff_display_name	降車場所名称
geometry	pickup_location	乗車場所座標
geometry	dropoff_location	降車場所座標
timestamp	pickup_datetime	乗車日時
timestamp	dropoff_datetime	降車日時
integer	count	乗車回数

3

[表名] m_vehicle_accessibility_features 車両利用可能マスタ		
varchar	vehicle_id	PK 車両ID
varchar	accessibility_feature_type	PK 機能利用可能種別
integer	count	機能数

1

[表名] m_reservation_passenger_count 予約乗客数マスタ		
varchar	reservation_area_id	PK 予約エリアID
varchar	passenger_type	PK 乗客タイプID
integer	count	人数

[表名] m_reservation_passenger_count 予約乗客数マスタ		
varchar	reservation_area_id	PK 予約エリアID
varchar	passenger_type	PK 乗客タイプID
integer	count	人数

[表名] m_reservation_accessibility_features_count 予約機能利用可能数マスタ		
varchar	reservation_area_id	PK 予約エリアID
varchar	accessibility_feature_type	PK 機能利用可能種別
integer	count	乗車可能機能数(中心乗客)

[表名] m_reservation_accessibility_features_count 予約機能利用可能数マスタ		
varchar	reservation_area_id	PK 予約エリアID
varchar	accessibility_feature_type	PK 機能利用可能種別
integer	subtotal	合計

2

[表名] m_stop 乗降場マスタ		
varchar	stop_id	PK 乗降場ID
varchar	name	名称
varchar	description	説明
geometry	location	乗降場所位置
timestamp	start_datetime	利用開始日時
timestamp	end_datetime	利用終了日時
boolean	is_booking_available	乗車可能フラグ
boolean	is_lighting_available	夜間利用可能フラグ

6

[表名] m_reservation_area 予約エリアマスタ		
varchar	service_id	PK サービスID
varchar	category	カテゴリ
varchar	name	名称
varchar	url	URL
boolean	agreement_required	同意必須フラグ

4

[表名] m_agreement 利用同意マスタ		
varchar	passenger_id	PK ユーザーID
varchar	service_id	PK サービスID
timestamp	agreed_at	同意日時

2

[表名] m_stop_features 乗降場利用可能マスタ		
varchar	stop_id	PK 乗降場ID
varchar	stop_id	PK 乗降場ID
varchar	title	タイトル
varchar	description	説明

3

[表名] m_vehicle 車両マスタ		
varchar	vehicle_id	PK 車両ID
varchar	service_id	PK サービスID
varchar	name	名称
varchar	status	乗車ステータス
varchar	number_plate	ナンバープレート
integer	seats	定員数

[表名] m_stop_booking_features 乗降場利用可能マスタ		
varchar	stop_id	PK 乗降場ID
varchar	search_word	検索ワード

[表名] m_reservation_area サービス別予約マスタ		
varchar	service_id	PK サービスID
varchar	stop_id	PK 乗降場ID

[表名] m_reservation_area サービス別予約マスタ		
varchar	stop_id	PK サービスID
timestamp	start_datetime	PK 予約開始日時
timestamp	end_datetime	PK 予約終了日時
varchar	description	説明

[表名] m_reservation_area サービス別予約マスタ		
varchar	service_id	PK サービスID

[表名] m_vehicle_picture 車両画像マスタ		
varchar	vehicle_image_id	PK 車両画像ID
varchar	vehicle_id	PK 車両ID
varchar	url	URL
varchar	title	タイトル
varchar	description	説明

[表名] m_reservation_area 予約エリアマスタ		
varchar	vehicle_id	PK 車両ID
geometry	location	現在地
timestamp	timestamp	タイムスタンプ

[表名] m_operation_delay 運行遅延マスタ		
varchar	delay_id	PK 遅延ID
varchar	vehicle_id	PK 車両ID
varchar	status	遅延ステータス
varchar	delay_reason	遅延理由コード
varchar	delay_reason_detail	遅延理由詳細
integer	delay_minutes	遅延時間
timestamp	started_at	遅延開始日時
timestamp	closed_at	遅延終了日時
varchar	closed_reason	遅延終了理由
varchar	closed_reason_detail	遅延終了理由詳細

7

- サービス
- 乗降場
- 車両
- ユーザー
- 予約
- 予約
- 遅延

番号	カテゴリ	主なテーブル（エンティティ）	内容・役割の説明
①	サービス関係	<ul style="list-style-type: none"> ・ m_service (サービスマスタ) ・ m_available_passenger_type (利用可能乗客タイプマスタ) ・ m_service_operating_hour (運行時間マスタ) ・ m_reservable_area (予約可能エリアマスタ) 	<p>「どのようなバスサービスを提供するか」を定義する領域です。</p> <p>運行するサービス自体の基本情報、利用可能な乗客区分（大人・子供など）、運行曜日や時間帯、サービス提供エリア（GeoJSON等）などを管理します。</p>
②	乗降場所関係	<ul style="list-style-type: none"> ・ m_stop (乗降場所マスタ) ・ m_service_stop (サービス・乗降場所紐付け) ・ m_stop_suspension (乗降場所休止期間管理) 	<p>「どこで乗り降りできるか」を管理する領域です。</p> <p>固定のバス停（ミーティングポイント）の位置情報や名称、写真、および特定の期間利用できない場合の休止情報を管理します。</p>
③	車両関係	<ul style="list-style-type: none"> ・ m_vehicle (車両マスタ) ・ m_vehicle_accessibility_feature (車両付帯設備マスタ) ・ t_vehicle_location (車両位置情報トラン) 	<p>「どの車両が運行し、どこにいるか」を管理する領域です。</p> <p>車両の定員や車椅子対応などの設備情報（マスタ）とGPS等から送信されるリアルタイムな現在位置情報（トランザクション）を保持します。</p>
④	ユーザー関係	<ul style="list-style-type: none"> ・ m_passenger (ユーザーマスタ) ・ m_passenger_home (自宅位置情報) 	<p>「誰が利用するか」を管理する領域です。</p> <p>MaaSシステムと連携したエンドユーザーの基本情報（ID、氏名、連絡先）や予約時に使用する自宅位置情報などを管理します。</p>
⑤	予約関係	<ul style="list-style-type: none"> ・ t_reservation (予約トラン) ・ t_reservation_passenger_count (予約人数明細) ・ t_payment (決済トラン) 	<p>「いつ・誰が・どの便に乗るか」を管理する中核領域です。</p> <p>予約ごとの乗降場所・時間・ステータス（確定/キャンセル等）、乗車人数内訳および運賃の決済状態をトランザクションとして記録します。</p>
⑥	規約関係	<ul style="list-style-type: none"> ・ m_term (規約マスタ) ・ m_service_term (サービス規約紐付け) ・ m_agreement (同意履歴トラン) 	<p>「利用ルールと同意状況」を管理する領域です。</p> <p>プラットフォームや各サービスごとの利用規約の版数管理と、各ユーザーが「いつ・どの規約に」同意したかの履歴を管理します。</p>
⑦	運用関係	<ul style="list-style-type: none"> ・ t_operation_delay (運行遅延トラン) 	<p>「運行の例外状況」を管理する領域です。</p> <p>車両ごとの遅延発生状況、遅延時間（分）およびその理由（渋滞・事故等）と解消状況を管理し、ユーザーへの通知等の基礎データとなります。</p>

3.5. 業務フロー/業務一覧の解説

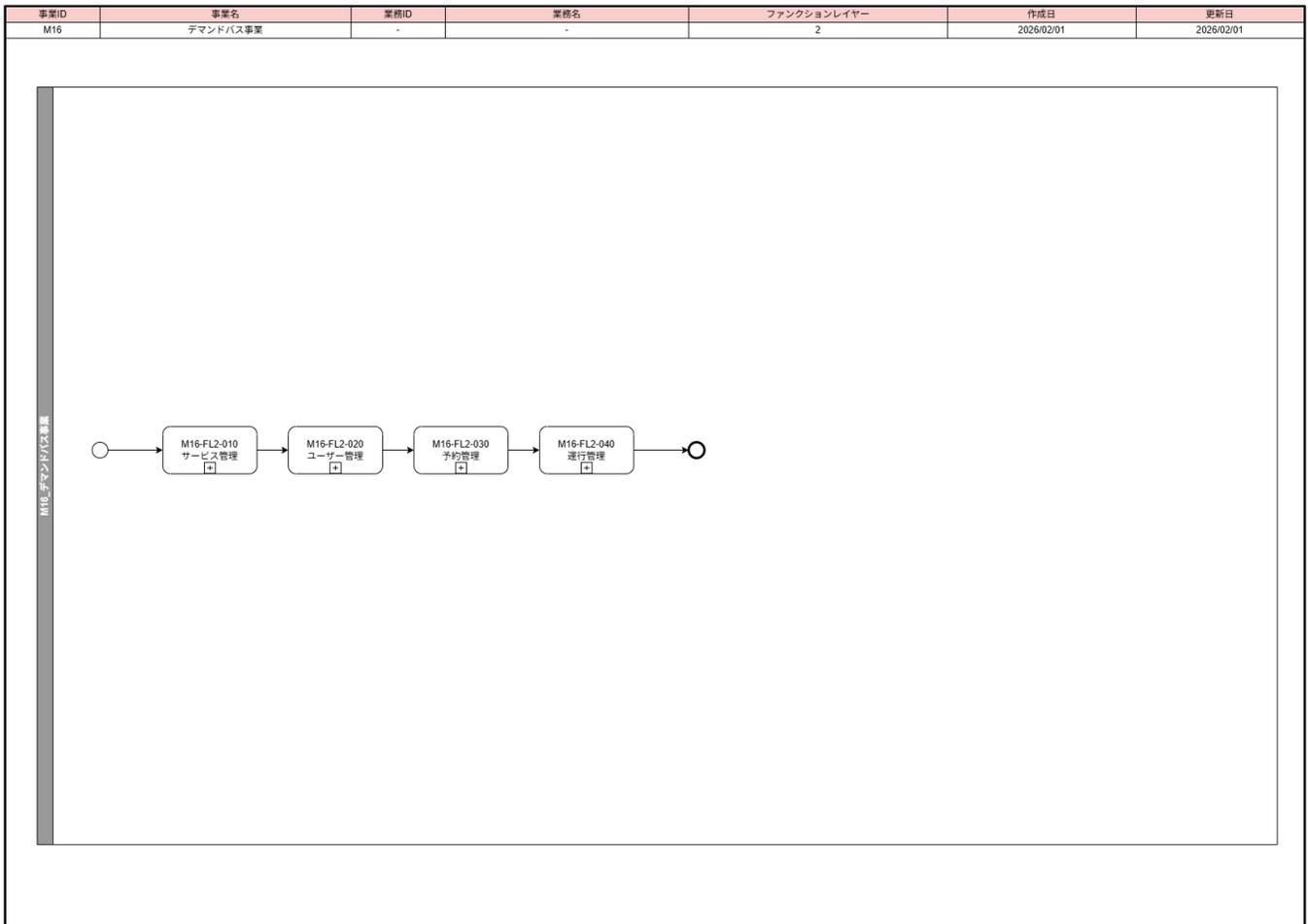
業務フロー/業務一覧のドキュメント(重要箇所)を掲載します。

FL(ファンクションレイヤー)2

FL2フローは、業務全体の概略を最大粒度で把握することを目的とします。

まず、MaaSサービスとデマンドバスサービス間でユーザーへ提供するサービスの管理を行います。次に、ユーザー登録や更新/削除を行う「ユーザー管理」を経て、ユーザーによるデマンドバスの「予約管理」へと移ります。最終的に、デマンドバスの運行を行う「運行管理」が実施されることで、一連のビジネスサイクルが完結する流れとなります。

業務フロー



業務一覧

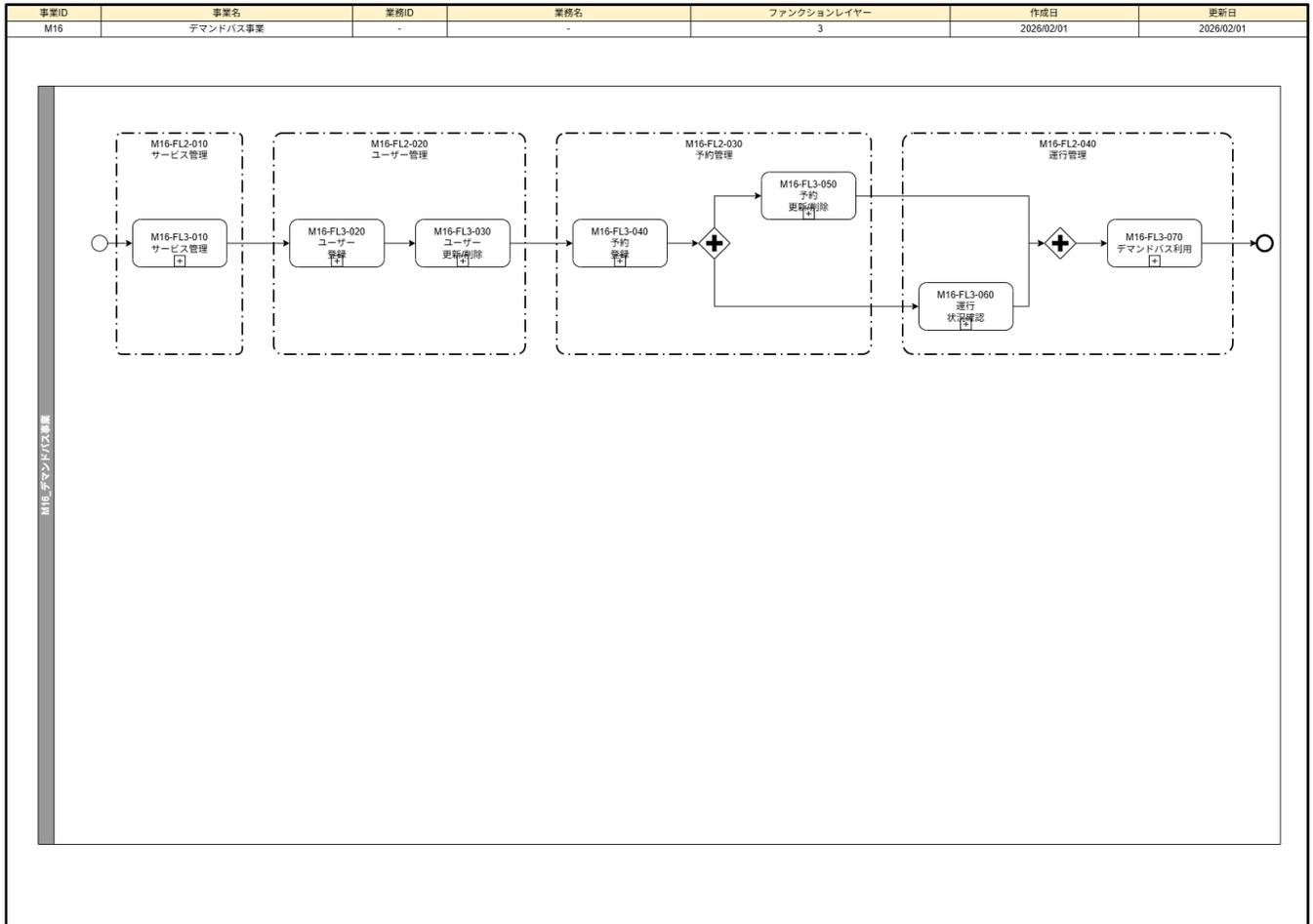
ID	詳細事業機能	内容
M16-FL2-010	サービス管理	MaaSサービスとデマンドバスサービス間で、事前にユーザーが利用するデマンドバスサービス情報(運行エリア、運行時間帯など)を管理する業務。
M16-FL2-020	ユーザー管理	MaaSサービス・デマンドバスサービスを利用するために必要なユーザー情報を管理する業務。
M16-FL2-030	予約管理	デマンドバスサービスの予約、予約の更新/削除を管理する業務。
M16-FL2-040	運行管理	デマンドバスの運行状況、運行を管理する業務。

FL(ファンクションレイヤー)3

FL3フローは、FL2を主要業務群へ分解し、各業務の実行順序を把握することを目的とします。

事前に、MaaSサービサーとデマンドバスサービサー間で運行エリア等のサービス情報を連携し、ユーザーの登録と更新/削除を行います。その後、ユーザーによるデマンドバスの予約登録や更新/削除を扱います。最後に、予約したデマンドバスを利用するにあたり運行状況の確認を行い、デマンドバスに乗車する流れになります。

業務フロー



業務一覧

ID	業務機能	内容
M16-FL3-010	サービス管理	MaaSサービサーとデマンドバスサービサー間で、事前にユーザーが利用するデマンドバスサービス情報(運行エリア、運行時間帯など)を連携する業務。
M16-FL3-020	ユーザー登録	MaaSサービス・デマンドバスサービスを利用するために、ユーザーの新規登録を行う業務プロセス。
M16-FL3-030	ユーザー更新/削除	デマンドバスサービサーに登録されているユーザー情報の更新/削除を行う業務プロセス。
M16-FL3-040	予約登録	ユーザーがデマンドバスを利用するために、予約の登録を行う業務プロセス。
M16-FL3-050	予約更新/削除	デマンドバスサービサーに登録されているデマンドバスの予約内容の変更/削除を行う業務プロセス。
M16-FL3-060	運行状況確認	ユーザーがデマンドバスに乗車するために、デマンドバスの運行状況(車両の位置情報、遅延情報など)を確認を行う業務プロセス。
M16-FL3-070	デマンドバス利用	ユーザーが実際にデマンドバスに乗車し、目的地で降車するまでの業務プロセス。

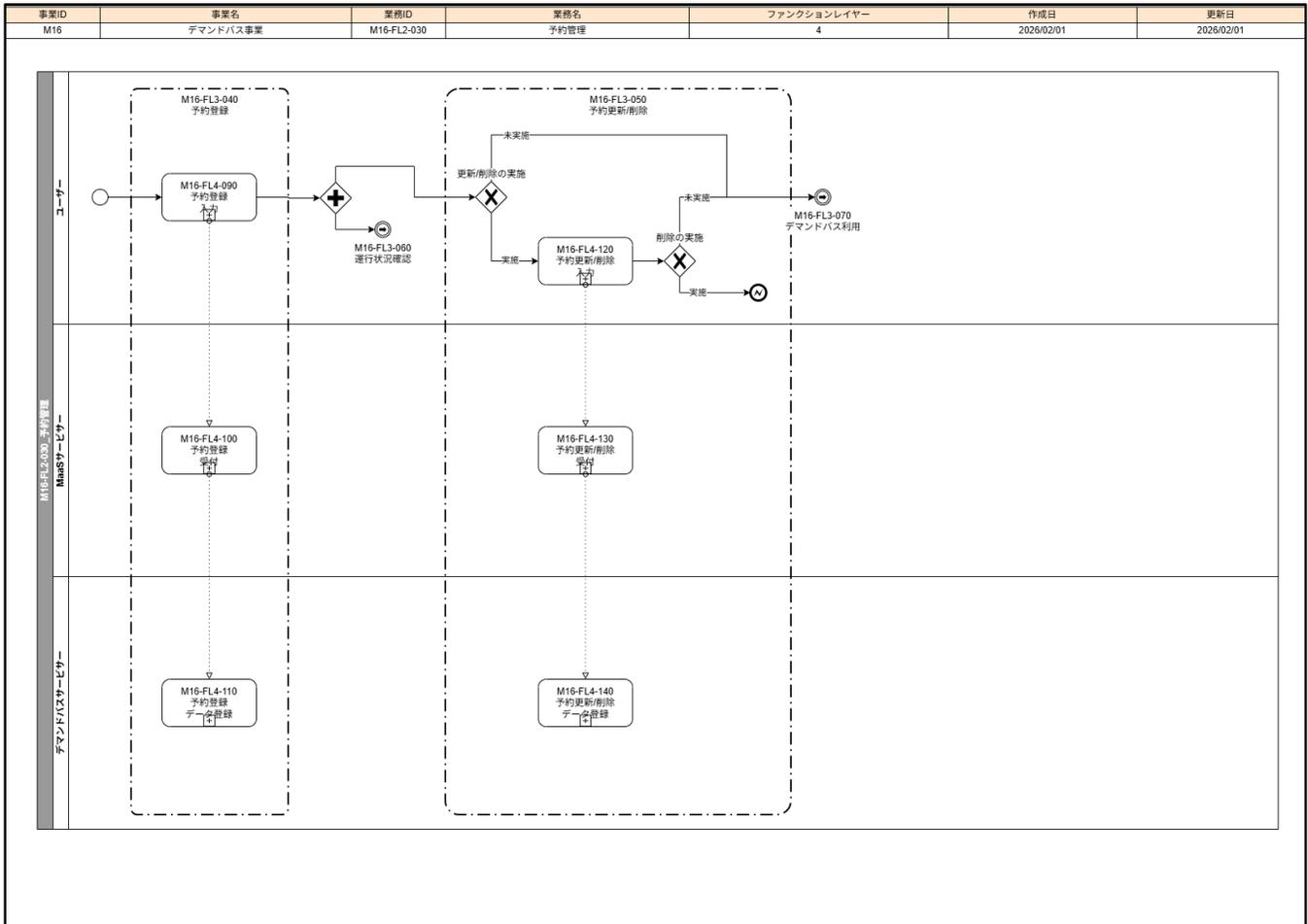
FL(ファンクションレイヤー)4

FL4フローは、FL3のタスクを組織間の受け渡し単位に分解し、各組織の役割を把握します。

ユーザーによる予約入力を起点に、MaaSサービスが予約を受け付け、デマンドバスサービスへ予約情報の連携を行い、予約が確定します。並行して運行状況確認が可能となり、必要に応じてユーザーが更新/削除を行います。その後、デマンドバスの運行状況確認や利用プロセスへと続く流れとなります。

業務フロー

—予約登録の業務を抜粋—



業務一覧

ID	詳細業務機能	内容
M16-FL4-090	予約登録入力	ユーザーが、予約したいデマンドバスを選択し、MaaSサービスへ送信する。
M16-FL4-100	予約登録受付	MaaSサービスが、ユーザーからの予約依頼を受け付け、デマンドバスサービスへ送信する。
M16-FL4-110	予約登録データ登録	デマンドバスサービスが、MaaSサービスからの予約登録要求を受け付け、予約登録を行う。
M16-FL4-120	予約更新/削除入力	ユーザーが、予約情報を照会し、予約の更新/削除を行い、MaaSサービスへ送信する。
M16-FL4-130	予約更新/削除受付	MaaSサービスが、ユーザーからの予約の変更/削除要求を受け付ける。
M16-FL4-140	予約更新/削除データ登録	デマンドバスサービスが、MaaSサービスから予約の更新/削除要求を受け付け、予約の更新/削除を行う。

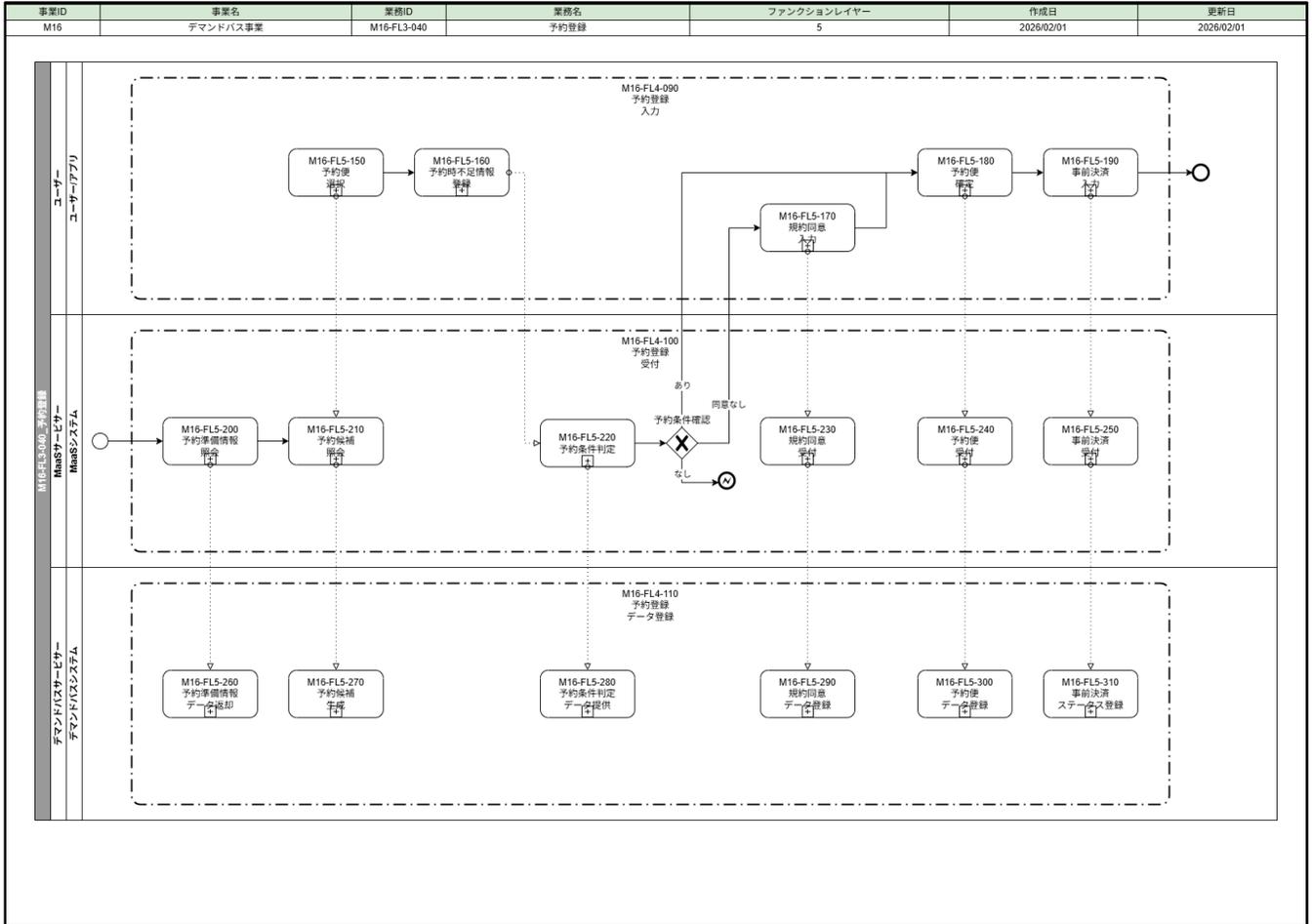
FL(ファンクションレイヤー)5

FL5フローは、組織内のアクターの具体的な業務内容の把握を目的としたものです。

ユーザーが、MaaSアプリから利用したい予約便の選択をすることを起点に、MaaSシステム内でユーザーの予約資格の確認を行います。予約資格を満たしている場合、ユーザーが予約リクエストを送り、予約情報と決済情報をデマンドバスシステムへ連携を行い、予約プロセスが完了となります。

業務フロー

—予約登録の業務を抜粋—



業務一覧

一予約登録の業務を抜粋

ID	単位作業	内容
M16-FL5-150	予約便選択	ユーザーが、アプリ上で乗車条件(乗降場所、乗車時間など)を入力し、レスポンスされた選択肢の中から乗車したいデマンドバスを選択し、予約申請を行う。
M16-FL5-160	予約時不足情報登録	ユーザーが、デマンドバス予約申請時までにユーザー登録が完了していなかった場合、ユーザーの新規登録を行う。
M16-FL5-170	規約同意入力	ユーザーが、デマンドバス予約申請時までに、最新の利用規約に同意できていなかった場合、最新の利用規約の同意を行う。
M16-FL5-180	予約便確定	ユーザーが、利用するデマンドバスの予約便を確定させ、MaaSサービスへ登録申請をする。
M16-FL5-190	事前決済入力	ユーザーが、アプリ上で乗車料金の決済を行う。
M16-FL5-200	予約準備情報照会	MaaSサービスが、デマンドバス予約に必要な情報(最新のサービス情報、乗降場所一覧)を事前にデマンドバスサービスから取得を行う。
M16-FL5-210	予約候補照会	MaaSサービスが、ユーザーから予約条件を受け取り、デマンドバスサービスへ送信し、レスポンスで返ってきた予約候補をユーザーへ返却する。
M16-FL5-220	予約条件判定	MaaSサービスが、ユーザーの利用資格を確認する。
M16-FL5-230	規約同意受付	MaaSサービスが、ユーザーから規約に同意した情報を受け取り、デマンドバスサービスへ送信する。 デマンドバスサービスから、登録結果を受け取り、ユーザーへ送信する。
M16-FL5-240	予約便受付	MaaSサービスが、ユーザーからの予約申請を受け付け、デマンドバスサービスへ送信する。 デマンドバスサービスから予約登録結果を受け取り、ユーザーへ返却する。
M16-FL5-250	事前決済受付	MaaSサービスが、ユーザーからの事前決済を受け付け、決済処理を行う。 決済処理後、決済ステータスをデマンドバスサービスへ送信する。
M16-FL5-260	予約準備情報データ返却	デマンドバスサービスが、MaaSサービスへ、リクエストに基づいたデマンドバスサービス情報や乗降場所一覧を送信する。
M16-FL5-270	予約候補生成	デマンドバスサービスが、検索条件(都道府県、市区町村など)に基づく、予約可能な便の候補を生成し、MaaSサービスへ送信する。
M16-FL5-280	予約条件判定データ提供	デマンドバスサービスが、MaaSサービスへ、予約資格の判定に必要な情報を送信する。
M16-FL5-290	規約同意データ登録	デマンドバスサービスが、MaaSサービスから規約同意状態を取得し、登録を行う。
M16-FL5-300	予約便データ登録	デマンドバスサービスが、MaaSサービスからの予約登録要求を受け付け、予約登録を行う。 MaaSサービスへ登録結果を送信する。
M16-FL5-310	事前決済ステータス登録	デマンドバスサービスが、MaaSサービスから事前決済情報を取得し、登録を行う。 登録結果をMaaSサービスへ送信する。

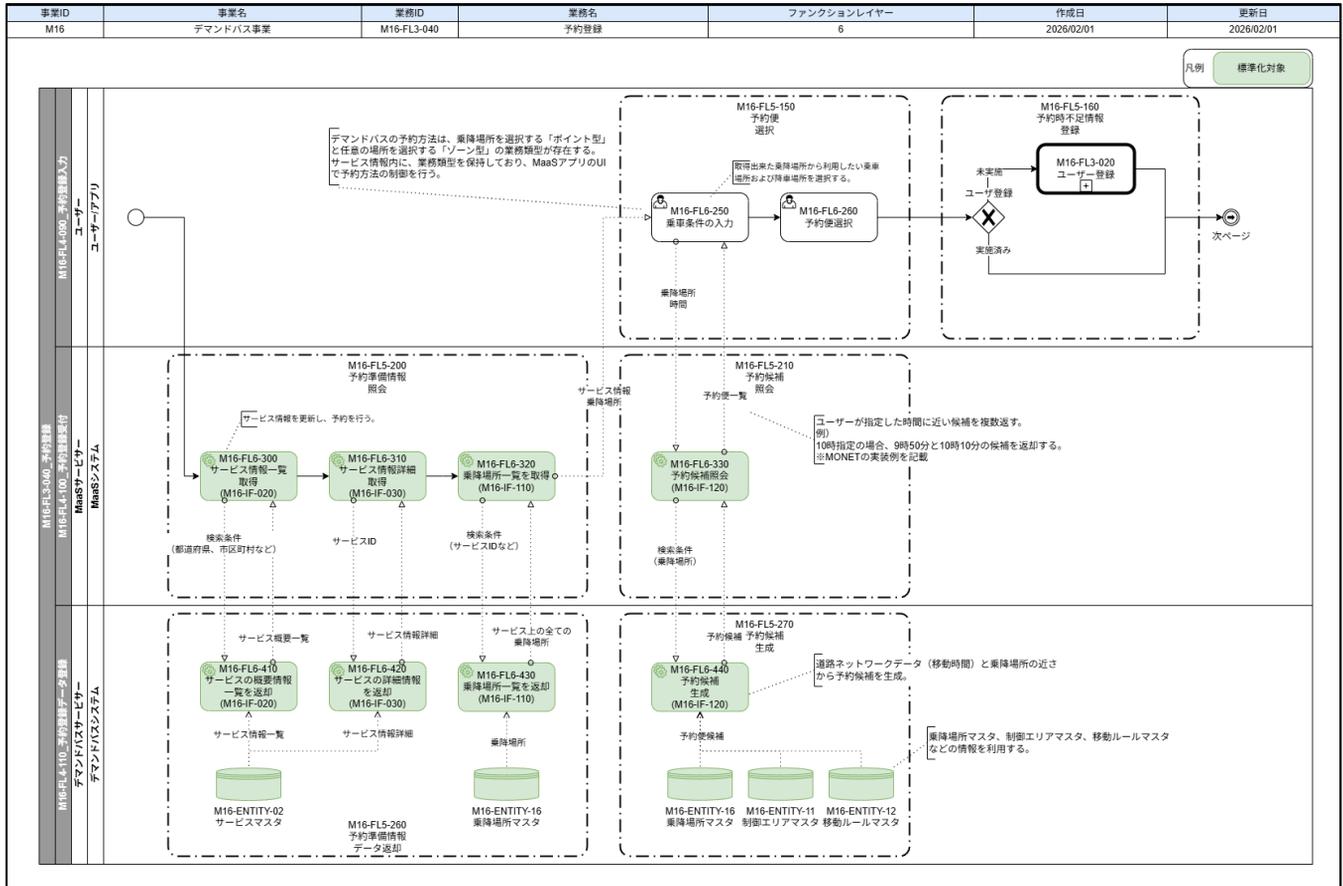
FL(ファンクションレイヤー)6

FL6フローは、手順レベルまで業務を分解し、アクター間の情報伝達の把握を目的とします。

事前に、MaaSシステムは最新のデマンドバスサービスを取得しユーザーへ表示します。ユーザーは乗車条件入力を行い、MaaSシステムがAPIを介して、デマンドバスシステム内で生成された予約候補を受け取り、ユーザーへ返却します。ユーザーが予約便を選択すると、ユーザー未登録の場合は「ユーザー登録」へ誘導され、実施済みの場合は次ページのフローへ続きます。

業務フロー

—予約登録の業務を抜粋—



業務一覧

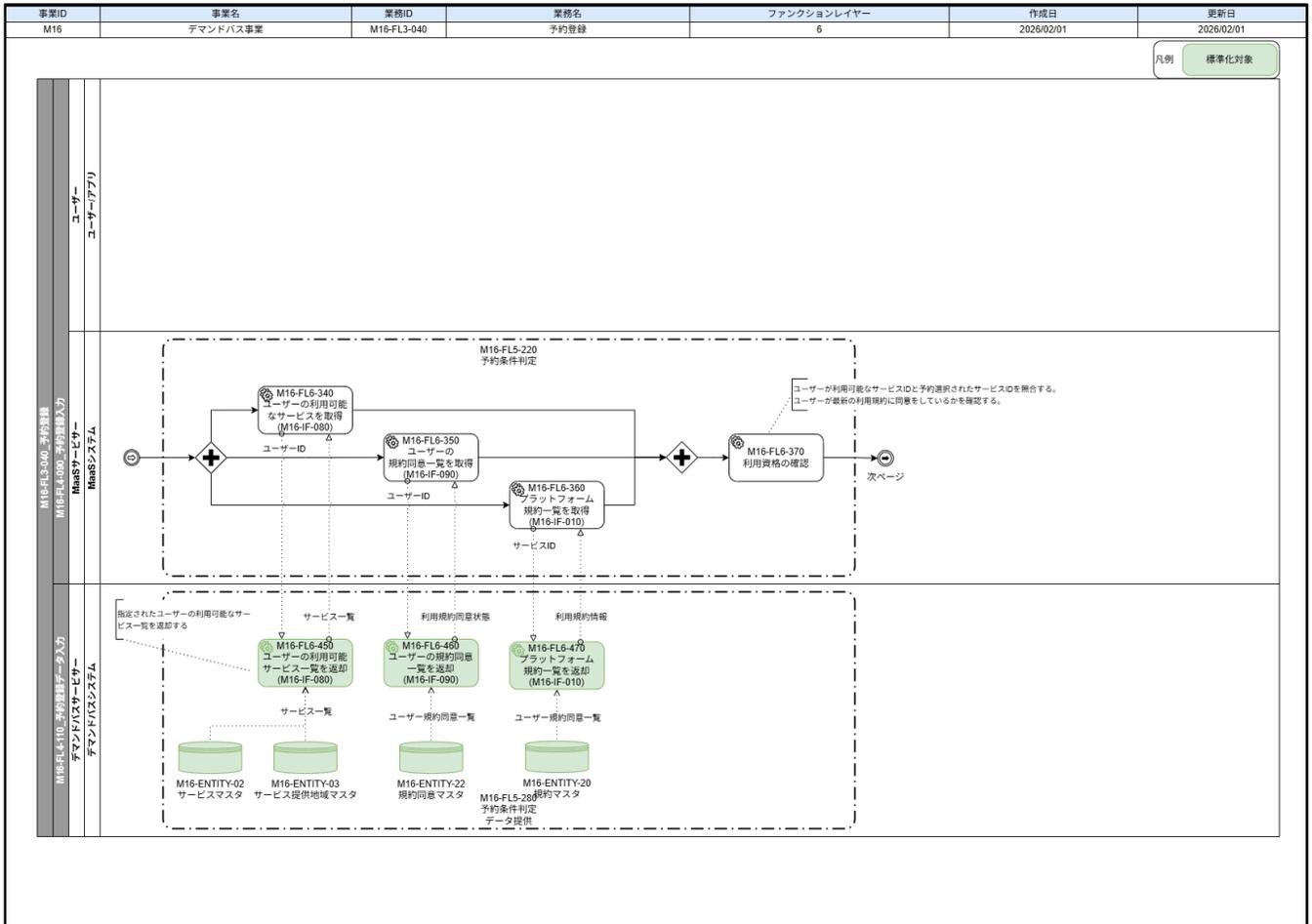
ID	要素作業	内容
M16-FL6-250	乗車条件の入力	ユーザーが、アプリ上で乗車条件(乗降場所、乗車時間など)を入力する。MaaSサービサーへ入力した乗車条件を送信する。
M16-FL6-260	予約便選択	ユーザーが、検索条件を基に提示されたデマンドバスの候補から、利用したいデマンドバスを選択し、予約申請を行う。
M16-FL6-300	サービス情報一覧取得	MaaSサービサーが、デマンドバスサービサーから、検索条件(都道府県、市区町村など)範囲のサービス情報一覧(サービスID、サービス名など)を取得する。
M16-FL6-310	サービス情報詳細取得	MaaSサービサーが、デマンドバスサービサーから、指定したサービスIDのサービス詳細情報(運行エリア、時間など)を取得する。
M16-FL6-320	乗降場所一覧を取得	MaaSサービサーが、デマンドバスサービサーから、サービスIDなどの条件に基づく、乗降場所を取得する。
M16-FL6-330	予約候補照会	MaaSサービサーが、ユーザーから予約条件を受け取り、デマンドバスサービサーへ送信し、レスポンスで返ってきた予約候補をユーザーへ返却する。
M16-FL6-410	サービスの概要情報一覧を返却	デマンドバスサービサーが、MaaSサービサーに、サービス情報一覧(サービスID、サービス名など)を送信する。
M16-FL6-420	サービスの詳細情報を返却	デマンドバスサービサーが、MaaSサービサーに、サービス詳細情報(運行エリア、時間など)を送信する。
M16-FL6-430	乗降場所一覧を返却	デマンドバスサービサーが、MaaSサービサーに、乗降場所一覧を送信する。
M16-FL6-440	予約候補生成	デマンドバスサービサーが、検索条件(都道府県、市区町村など)に基づく、予約可能な便の候補を生成し、MaaSサービサーへ送信する。

FL(ファンクションレイヤー)6

ユーザーによる予約便選択を起点に、MaaSシステムがユーザーの利用可能サービス一覧、規約同意状況、及びプラットフォーム規約一覧の取得を行います。これらの情報を基に、ユーザーの選択したサービスと本ユーザーの利用可能なサービスIDの照合と最新の利用規約への同意状態を確認を行います。利用資格が確認された後、次ページの予約確定プロセスへ続きます。

業務フロー

一予約登録の業務を抜粋



業務一覧

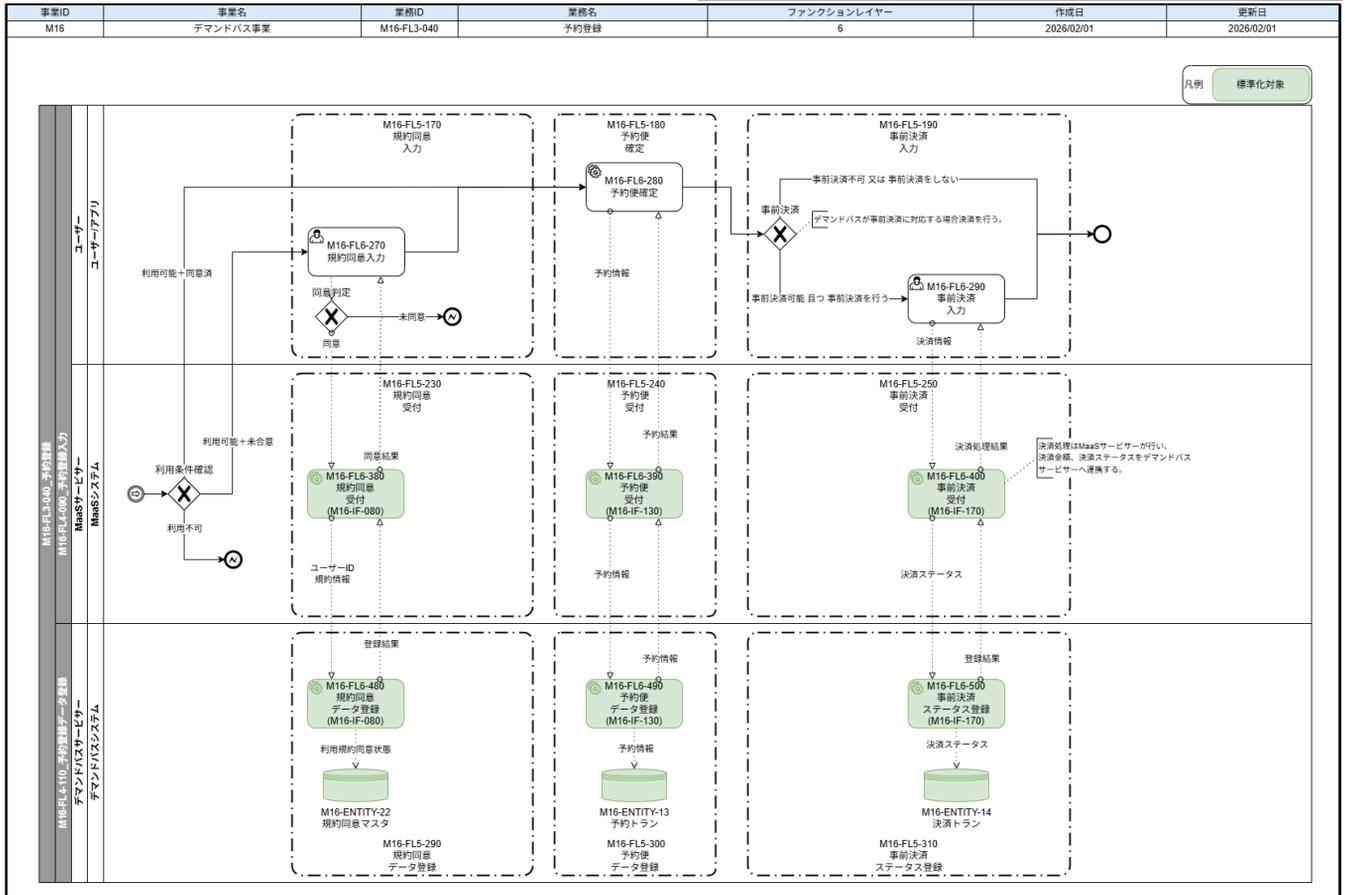
ID	要素作業	内容
M16-FL6-340	ユーザーの利用可能なサービスを取得	MaaSサービサーが、デマンドバスサービサーから、ユーザーIDを基にユーザーが利用可能なサービス一覧を取得する。
M16-FL6-350	ユーザーの規約同意一覧を取得	MaaSサービサーが、デマンドバスサービサーから、ユーザーIDを基に規約同意一覧を取得する。
M16-FL6-360	プラットフォーム規約一覧を取得	MaaSサービサーが、デマンドバスサービサーから、サービス利用に必要な最新のプラットフォームの規約一覧を取得する。
M16-FL6-370	利用資格の確認	MaaSサービサーが、ユーザーが利用可能なサービスか、最新の利用規約に同意できているかの確認を行う。
M16-FL6-450	ユーザーの利用可能サービス一覧を返却	デマンドバスサービサーが、MaaSサービサーへ、指定されたユーザーの利用可能なサービス一覧を送信する。
M16-FL6-460	ユーザーの規約同意一覧を返却	デマンドバスサービサーが、MaaSサービサーへ、指定されたユーザーの規約同意状況の一覧を送信する。
M16-FL6-470	プラットフォーム規約一覧を返却	デマンドバスサービサーが、MaaSサービサーへ、リクエストを基に規約一覧を送信する。

FL(ファンクションレイヤー)6

利用条件の確認後、ユーザーが最新の利用規約に未同意の場合はユーザーが利用規約の同意を行います。利用資格の確認が完了後、ユーザーが、事前決済の要否に応じて決済を行います。最終的に、予約情報と決済情報をMaaSシステムが受け取り、APIを介してデマンドバスシステムへ連携します。デマンドバスシステムがこれら情報を登録し、デマンドバス予約の一連のプロセスが完了します。

業務フロー

—予約登録の業務を抜粋—



業務一覧

ID	要素作業	内容
M16-FL6-270	規約同意入力	ユーザーが、デマンドバス予約申請時までに、最新の利用規約に同意できていなかった場合、最新の利用規約の同意を行う。
M16-FL6-280	予約便確定	ユーザーが、利用するデマンドバスの予約便を確定させ、MaaSサーバーへ登録申請をする。
M16-FL6-290	事前決済入力	ユーザーが、アプリ上で乗車料金の決済を行う。
M16-FL6-380	規約同意受付	MaaSサーバーが、ユーザーから規約に同意した情報を受け取り、デマンドバスサーバーへ送信する。 デマンドバスサーバーから、登録結果を受け取り、ユーザーへ送信する。
M16-FL6-390	予約便受付	MaaSサーバーが、ユーザーからの予約申請を受け付け、デマンドバスサーバーへ送信する。 デマンドバスサーバーから予約登録結果を受け取り、ユーザーへ返却する。
M16-FL6-400	事前決済受付	MaaSサーバーが、ユーザーからの事前決済を受け付け、決済処理を行う。 決済処理後、決済ステータスをデマンドバスサーバーへ送信する。
M16-FL6-480	規約同意データ登録	デマンドバスサーバーが、MaaSサーバーから規約同意状態を取得し、登録を行う。
M16-FL6-490	予約便データ登録	デマンドバスサーバーが、MaaSサーバーからの予約登録要求を受け付け、予約登録を行う。 MaaSサーバーへ登録結果を送信する。
M16-FL6-500	事前決済ステータス登録	デマンドバスサーバーが、MaaSサーバーから事前決済情報を取得し、登録を行う。 登録結果をMaaSサーバーへ送信する。



COMmmONS

by MLIT

デマンドバスシステム連携API標準仕様書 ガイダンス

発行日：2026年2月

発行元：国土交通省 総合政策局 公共交通政策部門 モビリティサービス推進課

受託者：MONET Technologies