

# タクシー配車システム 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

## 文書管理情報

項目	内容
文書名称	タクシー配車システム連携API標準仕様書ガイダンス
文書ID	commmmons_doc_001
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
最新版数	v1.0
制定年月	2026年2月
適用範囲	配車アプリと配車管理システム間での、配車依頼の連携や 配車状況・車両位置情報の同期
備考	本仕様書は2025年度の地域交通DX推進プロジェクト「COMmmmons」における「モビリティ・データ標準化プロジェクト」により作成されました。

## 改訂履歴

版数	年月日	改訂内容
v1.0	2026/02/13	初版制定

# 目次

## 1. プロジェクトの概要

- 1.1. はじめに
- 1.2. タクシー配車について
- 1.3. タクシー配車における現状課題
- 1.4. タクシー配車システム連携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/一覧の解説(一部抜粋)
- 3.6. ユースケース





# 1. プロジェクトの概要

---

- 1.1. はじめに
  - 1.2. タクシー配車について
  - 1.3. タクシー配車における現状課題
  - 1.4. タクシー配車の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「タクシー配車システム連携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

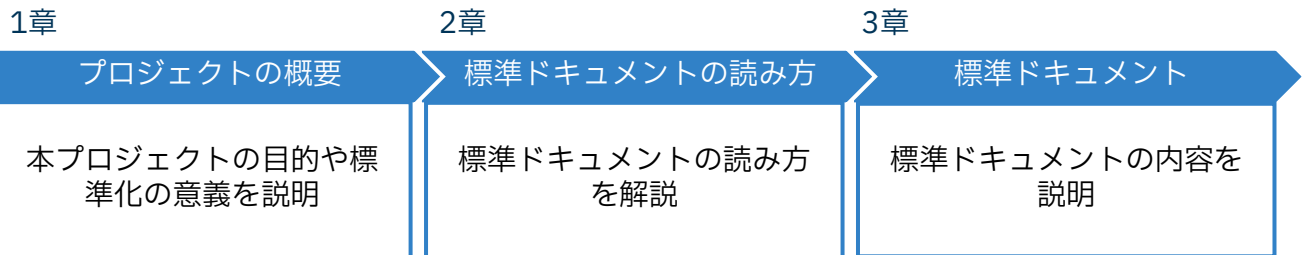
国土交通省では、地域交通DX推進プロジェクト「COMmmONS (コモンズ)」として、2025年度に「タクシー配車アプリ連携API標準化プロジェクト」を実施しました。

このプロジェクトは、現状バラバラに存在する配車アプリ（タクシー利用者向けアプリ。配車リクエストをスマートフォンアプリ上で可能とするものをいう。）と配車管理システム（タクシー事業者向けシステム。アプリや電話などからリクエストを受け付け、配車室のオペレーションによりリクエストに応じた配車を差配するものをいう。）間の標準的な連携インターフェースを開発することを目的としています。

具体的には、標準APIの提供により、既存の配車アプリや地域アプリ、WEBサービスなど様々な「出面」と配車システムとの「N対N」の接続を容易にすることで、「出面」がアクセス可能な車両数を増加させ、ユーザーが利用するアプリに関わらず共通の配車リソースの利用を可能とするとともに、配車アプリと配車管理システムが連携するための個別開発工数を低減させることを目指します。

### 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、配車アプリと配車管理システムの連携を検討される交通事業者様やシステムベンダーの開発担当者様、自治体の交通政策等の担当者様に向けた資料となります。



## 1.2. タクシー配車について

タクシー配車とは、利用者の指定した場所へタクシー車両を呼び出し、乗車できるサービスです。路上で空車のタクシーを探して手を挙げる「流し」の利用とは異なり、指定場所へ車両を呼ぶことができる点が特徴です。主に「電話」または「アプリ」を通じて依頼を行います。「アプリ」を用いた配車の中にも、配車室で配車するパターンと、アプリ側で配車するパターンがあります。

### 配車方法の種類と特徴

#### • 電話配車

乗客は電話により配車リクエストを行う。リクエストを受注した配車室オペレーターは、無線等を通じて差配車両を決定。

#### • 配車管理システムによる配車(配車室で差配)

乗客は電話または配車アプリにより配車リクエストを行う。リクエストを受注した配車室オペレーターは、配車管理システムを通じて差配車両を決定。

#### • アプリ配車(アプリで差配)

乗客は配車アプリにより配車リクエストを行う。リクエストはアプリ内の差配ロジックにより差配され、ドライバーズアプリを通じて差配車両を決定。

### 利用手順

#### 電話配車

##### 1. 電話によるリクエスト

乗客が配車依頼



##### 2. 配車受付

オペレーターが  
依頼受付



##### 3. 車両選択

オペレーターが  
車両選択



##### 4. 差配完了

オペレーターから  
差配完了連絡



##### 5. 迎車



#### 配車管理システムによる配車 (配車室で差配)

##### 1. アプリ又は電話による リクエスト

乗客が配車依頼



##### 2. 配車受付

オペレーターが  
配車管理システム  
を用いて依頼受付



##### 3. 車両選択

オペレーターが  
配車管理システム  
を用いて車両選択



##### 4. 差配完了

オペレーターから  
差配完了連絡



##### 5. 迎車



#### アプリ配車 (アプリで差配)

##### 1. アプリによるリクエスト

乗客が配車依頼



##### 2. マッチング

アプリのロジック  
が配車先を選定



##### 3. 差配提案

アプリが  
ドライバーに  
差配提案



##### 4. 提案受領

ドライバーが  
差配提案受領



##### 5. 迎車



### 1.3. タクシー配車における現状課題

特に地方部においては、「お客様」への配慮や乗務員負担の平準化等の観点から、タクシー配車を配車室の直接オペレーションによって行うニーズがあります。

他方、配車アプリや地域アプリの配車リクエスト機能などの普及が進みつつある昨今、電話とアプリという二つの経路で配車室がリクエストを受注するモデルも出現しています。

#### 解決すべき課題

地域のタクシー事業者が様々な配車アプリや配車管理システムを導入しつつあるなか、配車アプリ-配車管理システムが垂直統合されているケースでは、ユーザーが地域のタクシー車両へアクセスするためには、電話によるリクエストと同様に、様々なアプリやサービスを使い分ける必要があり、地域の輸送資源へのアクセシビリティに課題が生じます。

他方、配車アプリや配車管理システムはビジネスベースで提供される競争領域のプロダクトであり、各地域やタクシー事業者は自らのニーズや運用にマッチした最適なサービスを選定する立場にあります。

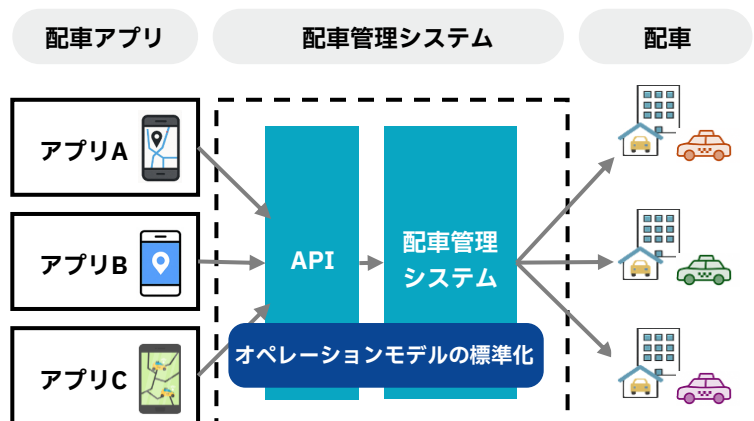
そこで、地域で様々な配車アプリや配車管理システムを併存させつつ、輸送資源へのアクセシビリティを最大化するため、配車アプリ-配車管理システムの間を連携させるためのシステムインターフェースの標準化が必要となります。

これにより、両システム間の連携コストを低減しつつ、ユーザーがアクセスできる輸送資源を最大化し、需要と供給のマッチングの最適化を図ることが可能となります。

#### 配車アプリと配車システムの連携が進んでいない



#### 配車アプリ-配車管理システムの間を連携させるためのシステムインターフェースを標準化



地域交通の確保と効率化のためには  
事業者間のシステムやオペレーション連携、標準化が不可欠

## 1.4. タクシー配車のAPI標準化による課題解決

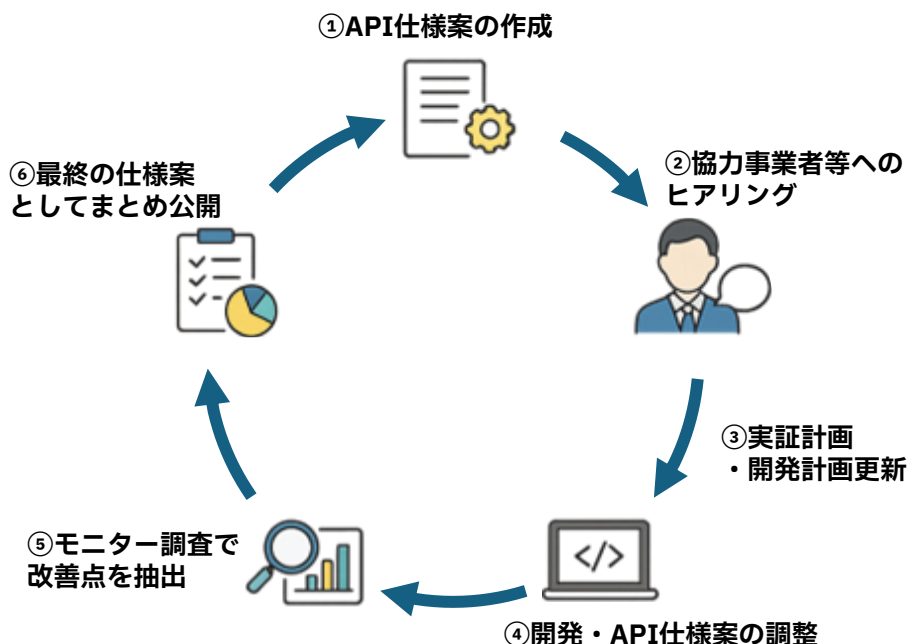
「タクシー配車アプリ連携API標準化プロジェクト」では、既存の配車アプリ及び配車管理システムの仕様調査を行ったうえで、配車アプリと配車システム間の連携を円滑化するためのシステム間連携インターフェースおよび業務連携モデルの標準仕様を策定しました。

標準仕様の策定に当たっては、交通事業者や関連システムベンダーへのヒアリングを通じてシステム仕様やシステム運用の課題を整理し、標準仕様案へ反映しました。また、策定した標準仕様案の実証運用を行い、課題のフィードバックを得たうえで、最終的な仕様案として取りまとめました。

### 課題解決のアプローチ

#### 【標準I/F仕様】

1. 配車アプリ/配車システム間のAPIや業務フローのうち標準的な範囲の定義や、API仕様(データ項目/業務フロー)の暫定案を作成
2. 配車システムベンダー、配車アプリベンダー、タクシー事業者等にヒアリングし、以下を調査
  - ・アプリの運用課題の整理
  - ・各障壁のクリティカル度合い、メジャー度合い
  - ・各ベンダー製品の配車アプリ-システム間のAPI仕様の標準化の課題の整理
3. 調査結果を踏まえて、解決施策を検討し、実証計画・システム開発計画を更新
4. 必要な開発・API仕様案の調整
5. 複数のタクシー事業者で活用し、運用に合うか定性評価、改善点を抽出
6. 改善点を反映してI/F仕様案としてとりまとめ





## 1.5. API標準化のプロセス

本プロジェクトは以下の3つのステップで実施しました。

まず、「現状調査」で関係各社の既存業務やシステム仕様を把握します。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、最終的な標準仕様案を策定しました。

### 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。

### 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API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	配車管理システムベンダー	電脳交通	自社運営での知見を有しているため
2	タクシー事業者	万代タクシー(株)	運用課題などの知見を有している。配車アプリの導入にも積極的な姿勢であるため
3		タクシーステーション新潟	
4		アイシステム合同会社	立ち上げ時期が適切な見込みのため
5		山形共同配車(山交ハイヤー・観光タクシー)	立ち上げ時期が適切な見込みのため
6	アプリベンダー	Uber Japan(株)	連携済みかつグローバル知見がある
7		DiDiモビリティジャパン(株)	協力体制が築けているため
8		S.RIDE(株)	協力体制が築けているため
9		GO(株)	業界シェア上位であるため
10		(株)NearMe	相乗り・乗合の観点での連携模索が可能
11		(株)アイシン	協力体制が築けているため
12		Newmo(株)	関西圏でのタクシー配車DX化取組に積極的なため
13	業界団体	高松市タクシー協会	ローカル運用の適切なサンプルとして参考情報を得る
14	システムベンダー	(株)デジタルテクノロジー四国	ローカルアプリとの連携検討をするため
15		VAL研究所	ローカルアプリとの連携検討をするため
16	配車管理システムベンダー	(株)システムオリジン	長年の配車SYS供給の知見があるため
17		モバイルクリエイイト(株)	
18		新潟通信機(株)	新鋭としての配車SYS供給の知見があるため
19	ミックウェア		

## 現状調査及び標準化の対象

本プロジェクトにおける調査対象として、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」を定義しました。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ました。

### 標準業務モデル

#	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	業務一覧・業務モデル/フローの作成	電話/配車システム/ドライバーアプリを利用した単独での配車業務フロー	ヒアリングドキュメントリサーチ	<ul style="list-style-type: none"> <li>標準業務一覧</li> <li>標準業務フロー</li> </ul>	<ul style="list-style-type: none"> <li>タクシー配車オペレーションに関する事業性調査</li> <li>タクシー配車オペレーションに関する技術的なフィジビリティスタディー</li> <li>タクシー配車オペレーションのシステムアーキテクチャ</li> </ul>
2	業務一覧・業務モデル/フローの作成	配車導入との業務差分			
3	業務一覧・業務モデル/フローの作成	ビジネス要件ほか立上に必要な事項			

### 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	API仕様の作成	機能モジュール単位のシステム・アーキテクチャは何か？	ヒアリングドキュメントリサーチ	API仕様	<ul style="list-style-type: none"> <li>タクシー配車システム連携APIに関する事業性調査</li> <li>タクシー配車システム連携APIに関する技術的なフィジビリティスタディー</li> <li>タクシー配車システム連携APIシステムアーキテクチャ</li> <li>配車アプリと配車システムの連携の為にAPI仕様案に関する有用性調査結果</li> </ul>
2	API仕様の作成	配車アプリ、I/F、配車システムそれぞれが保持する機能は何か？			
3	API仕様の作成	対象範囲のAPIのデータ項目・データ型・Request Body・Response Body 他 Open APIの必要項目は？			

## 1.6. API標準化がもたらす便益

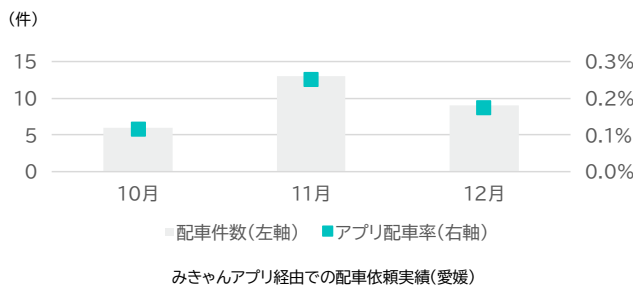
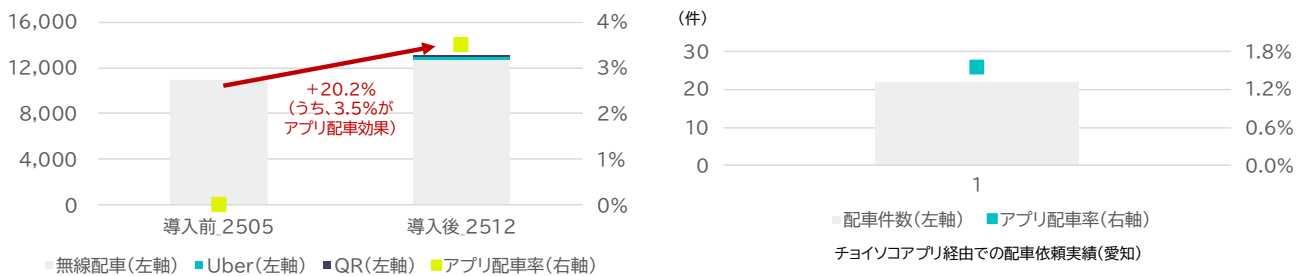
「タクシー配車システム連携API」とこれを運用するための標準業務モデルの標準仕様を策定することで、ユーザーの移動体験の良化とタクシー事業者の経営効率向上を同時に実現することを目指します。

### 実証仮説

- 標準APIに対応するタクシー配車アプリが増えることで、地方自治体が運営するサービスからの配車依頼が可能になり、タクシー配車アプリへの配車依頼が増加する。
- 標準APIに対応するタクシー配車アプリが増え、配車依頼が増加した結果タクシー事業者への配車依頼も増加が見込まれる。

### 実証の内容・結果と便益

- 実証実験を通じ、標準APIを介したアプリ導入によって、タクシー事業者における総配車件数の増大を確認しました。
  - 新潟県では、万代タクシーがUber・QR配車導入前（2025年5月）と比較し20.2%の配車件数増加（2025年12月）が確認されました。うち、3.5%がアプリ導入による配車件数増加効果となります。
  - 愛知県では、地方自治体が運営するサービスを介し、22件/月（連携タクシー事業者の月間配車件数の1.6%）のタクシー依頼が確認されました。
  - 愛媛県では、地方自治体が運営するサービスを介し、9.3件/月（連携タクシー事業者の月間配車件数の0.2%）のタクシー依頼が確認されました。



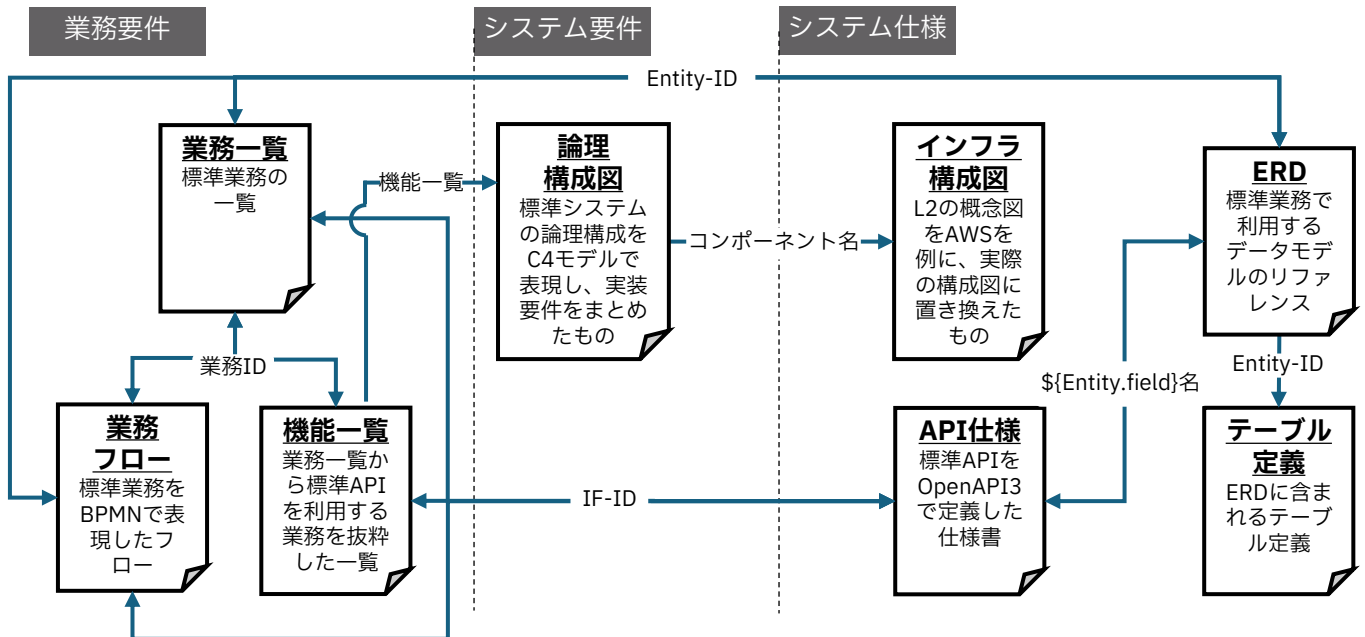
## 1.7. 標準ドキュメントの種類

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	
タクシー配車システム連携API標準仕様書_ガイダンス	現状の課題や標準化の意義など、プロジェクトの背景やねらいを解説する。「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	
タクシー配車システム連携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
タクシー配車システム連携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	
タクシー配車システム連携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	
タクシー配車システム連携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
タクシー配車システム連携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	
タクシー配車システム連携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	
タクシー配車システム連携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	
タクシー配車システム連携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/001/>)



## 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仕様の読み方
- 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ー事項単位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

	<b>人型 (Person):</b> <ul style="list-style-type: none"><li>➤ システムを利用するユーザー、または事業者(操作の主体)。</li></ul>
	<b>四角 (Software System / Container):</b> <ul style="list-style-type: none"><li>➤ Level 1では: 1つの大きなシステム全体。</li><li>➤ 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。</li></ul>
	<b>円筒 (Database):</b> <ul style="list-style-type: none"><li>➤ データが保存される場所。</li></ul>

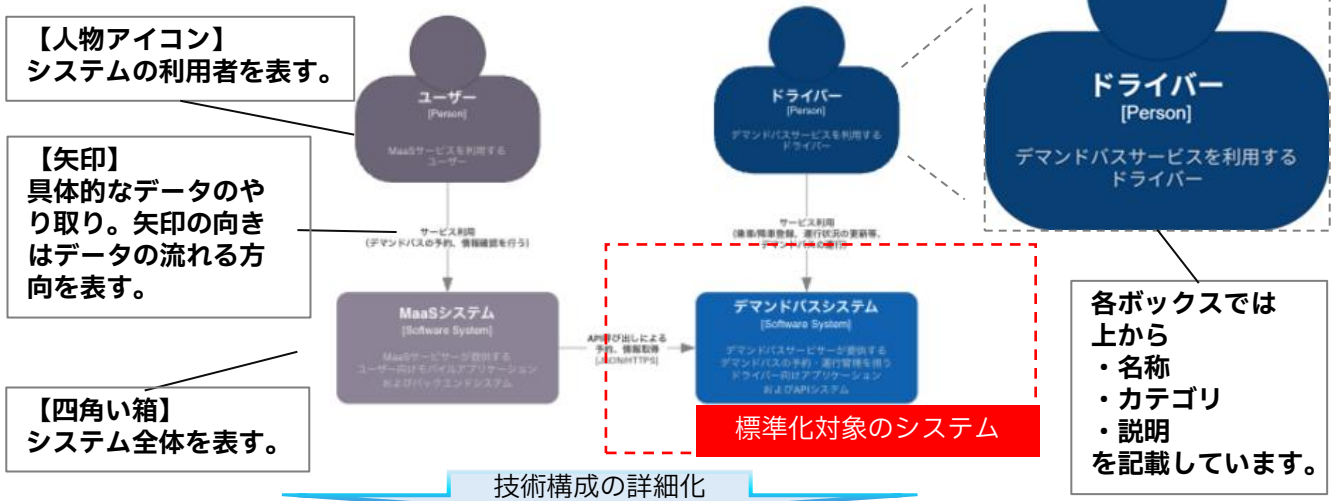
#### 色の意味

	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)

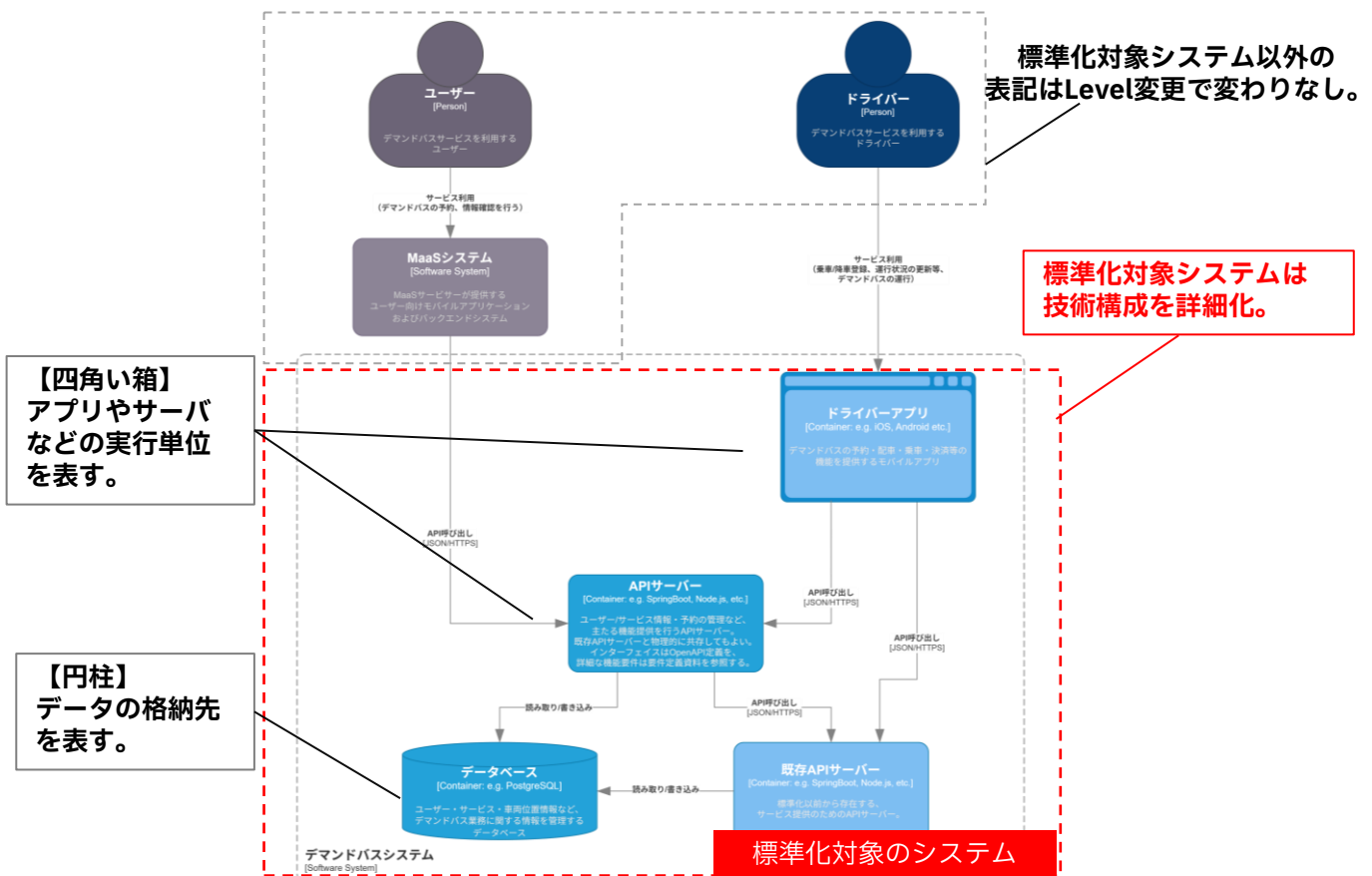
## 論理構成図の読み方

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（レベル1）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（レベル2）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。

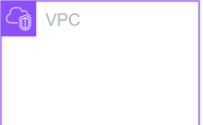
インフラ構成図は、利用者や管理画面からの操作が、セキュリティ層とAPIを經由し、サーバーレス処理とデータベースで処理・保存される流れを示しています。また、インターネット接続可能な領域と閉域を区分けし、さらに専用線接続などで安全性を担保したネットワーク設計全体を俯瞰できます。

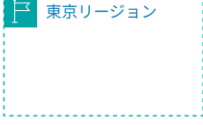
### システム構成要素

		実装例として扱う技術要素			
 <p><b>セキュリティ</b></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</p>	 <p>AWS Network Firewall</p>		
 <p><b>コンピュータ</b></p>	<p>アプリケーション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</p>	 <p>AWS Lambda</p>		
 <p><b>データベース</b></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><b>ゲートウェイ</b></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</p>	 <p>NAT Gateway</p>	 <p>Direct Connect Gateway</p>	 <p>Virtual Private Gateway</p>

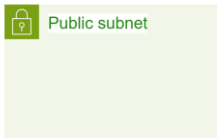
### 枠組みの説明

 MaaSシステム  
 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。

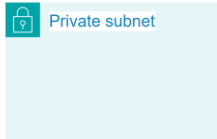
 VPC  
 VPC(紫の枠): Virtual Private Cloud。AWS上に構築された他から独立した仮想ネットワーク空間です。

 東京リージョン  
 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。

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。

 Public subnet

➤ 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 subnet

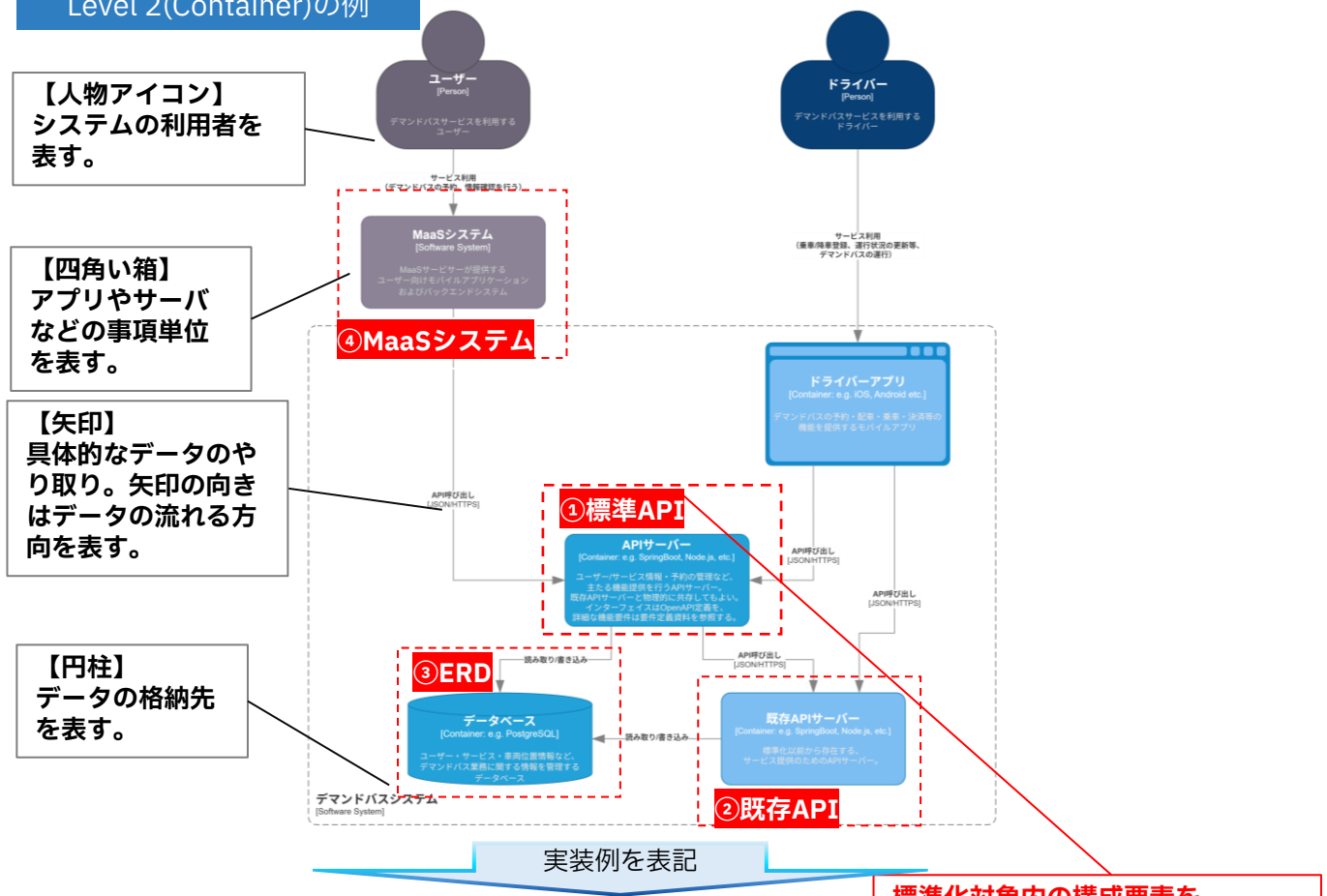
➤ Private: インターネットから直接アクセスできない安全な区画（データベースや処理実行部など）。



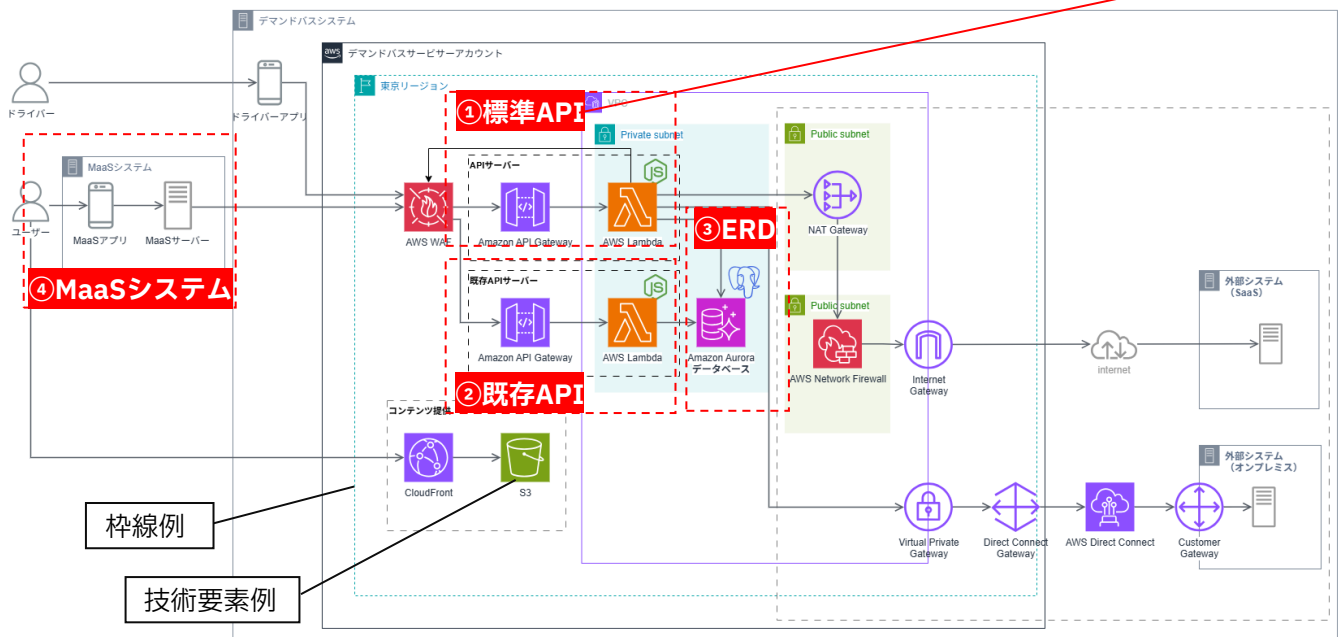
## インフラ構成図の読み方

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

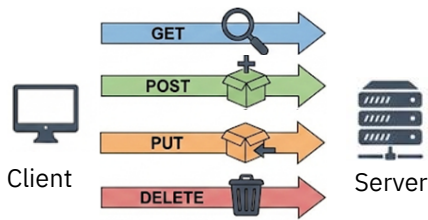


## 2.2. APIの読み方

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。

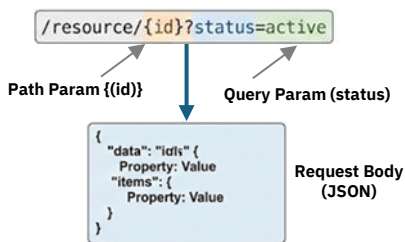
### 仕様定義の構造

#### Paths & Operations



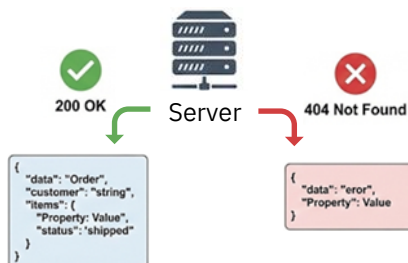
- Paths : サーバー上の「どのデータ」を操作するかを指定するURI(Uniform Resource Identifier)です。
- Operations : 実行する操作タイプです。
  - GET: リソースの取得 (副作用なし)。クエリパラメータでのフィルタリングが主。
  - POST: リソースの新規作成。リクエストボディに作成データを含めます。
  - PUT: リソースの置換・更新。ID指定でリソースの状態を変更します。
  - DELETE: リソースの削除。

#### Parameters & Request Body



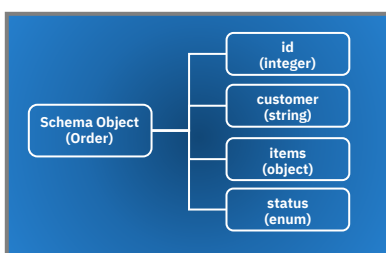
- Parameters (in): パラメータの格納場所を指定します。
  - path: URIの一部 (例: `/resource/{id}`)。リソースの特定に使用。required: trueが必須。
  - query: URIの末尾 (例: `?status=active`)。検索条件やソート指定に使用。
  - header: HTTPヘッダー (認証トークン等)。
- Request Body: POST/PUT時に送信するペイロードです。
  - content: メディアタイプを指定 (主に `application/json`)。
  - required: ボディ自体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

#### 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仕様の読み方

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に記載しています。

The screenshot shows the API documentation for the endpoint 'ユーザーの規約同意を登録'. 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 text explaining the endpoint's purpose and requirements.
- 4**: Path parameters section showing 'passenger\_id' as a required string.
- 5**: Request body schema section showing 'agreements' as a required array of objects.
- 6**: Responses section listing status codes: 200 OK, 400 Bad Request, and 500 Internal Server Error.
- 7**: HTTP method and URL path section showing 'POST /passengers/{passenger\_id}/agreements'.
- 8**: Request samples section showing a JSON payload for the 'agreements' array.
- 9**: Response samples section showing the expected JSON response structure.

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の具体例です。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

① m_ticket_routing_rule ・ チケットルーティングマスタ			
varchar	rule_id	PK	ルールID
varchar	maas_system_id	FK	MaaSシステムID
varchar	maas_ticket_id		MaaS商品ID
varchar	description		振り分け説明

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

—○	0
—+	1
—<	多

- 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。

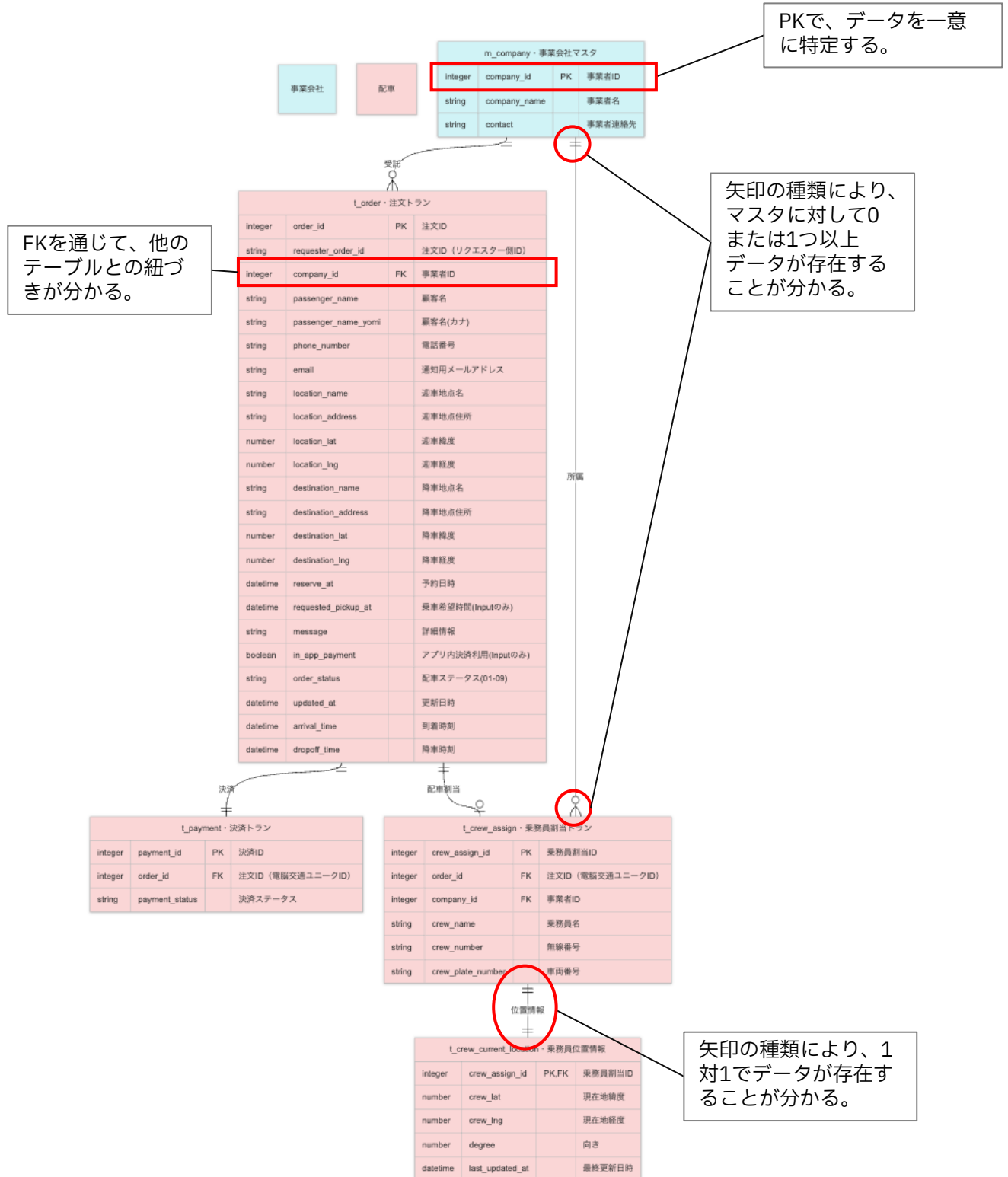
##### オプション表記

—<	1:1以上
—○<	1:0以上
—○+	1:0 or 1
—<	1:多
—	1:1

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

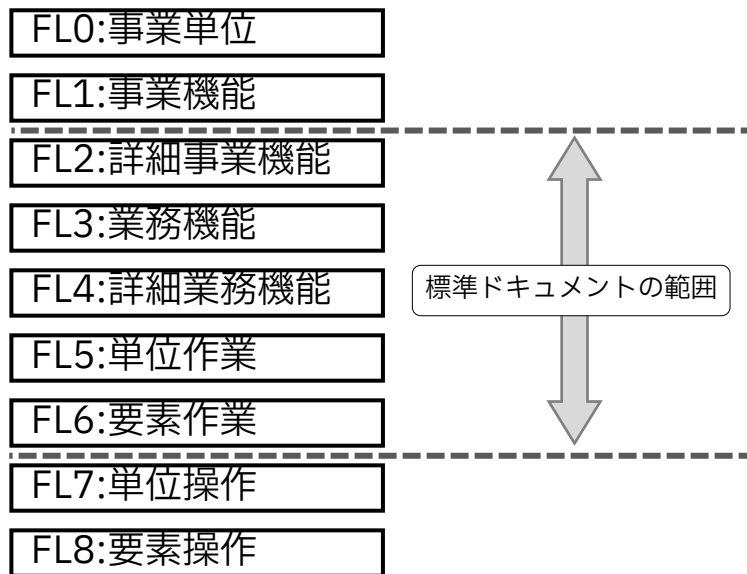
業務フロー/業務一覧は、標準API仕様に基づくシステム実装を運用するための業務モデルの指針です。業務をファンクションレイヤー（以下、FLとする）定義に従って「事業」から「要素作業」まで階層化し、標準的な粒度で整理します。その上で、BPMN 2.0表記法に基づきシステムと人の動きを可視化します。これにより業務要件とAPIの整合性を確保し、システムを運用するための具体的な業務手順を理解することができます。

### 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。

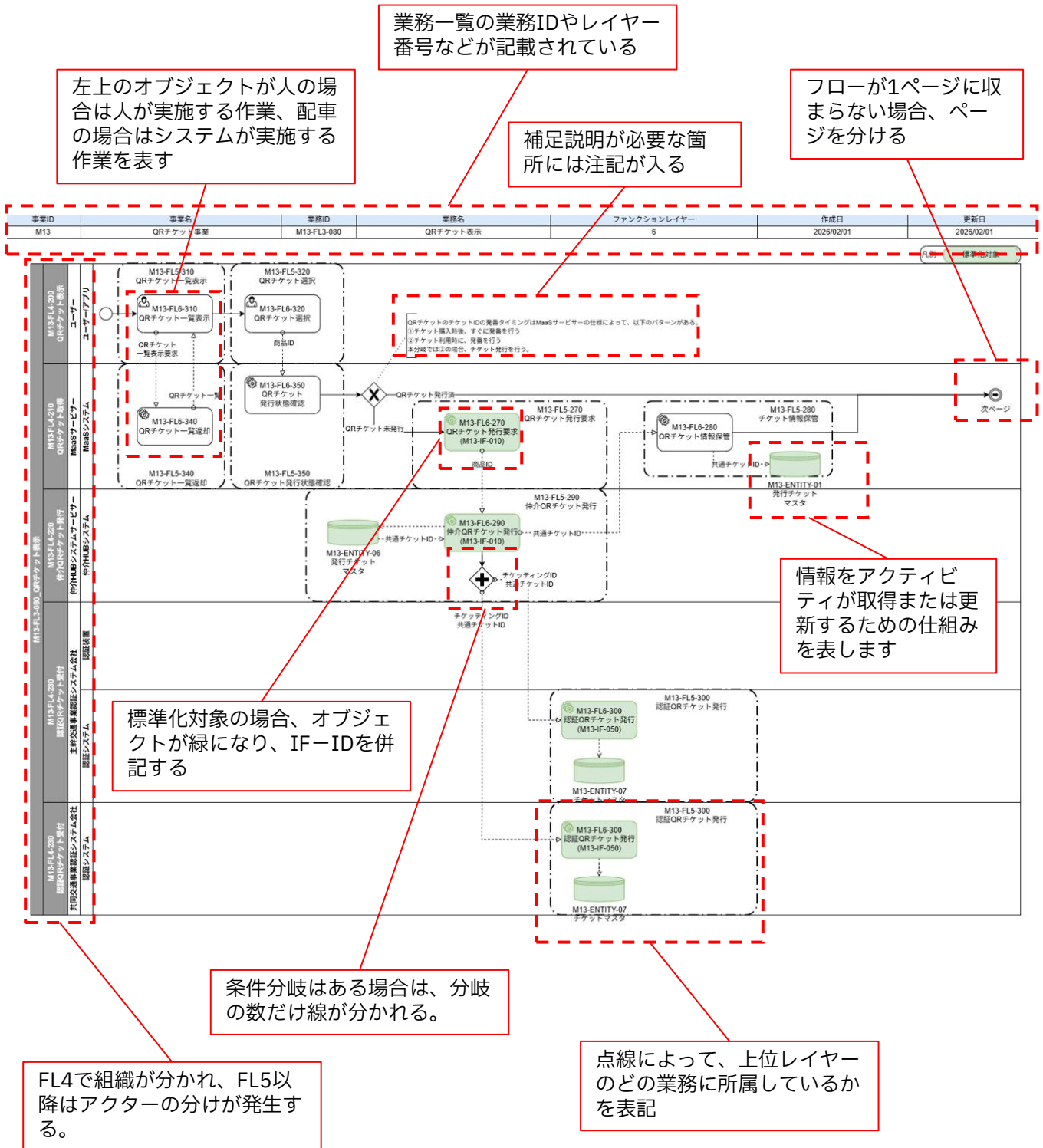
アクター	システム	API		目的・内容
		ID	名称	
				交通事業者、MaaSサービス、認証システム間で共通のQRチケットのサービスを提供するにあたり、商品を決めるための業務。 共通のQRチケットを利用する事業者間で、商品情報(商品名、販売金額など)を取り決める業務プロセス。
主管交通事業者				主管交通事業者が、商品企画を検討し、共同交通事業者と同意を行う。 主管交通事業者の規格事が商品企画の検討を行い、共同交通事業者へ確認をしてもらう。 主管交通事業者の規格事が商品企画の検討を行い、共同交通事業者へ確認をしてもらう。
主管交通事業者				主管交通事業者の承認者が、担当者から受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


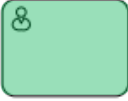

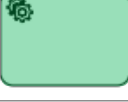

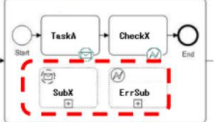

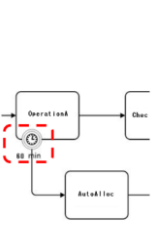


視線を左から右へ動かすことで、大枠の「詳細事業機能（FL2）」から、詳細な「要素作業（FL6）」へと段階的に具体化される構造になっています。





## 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)



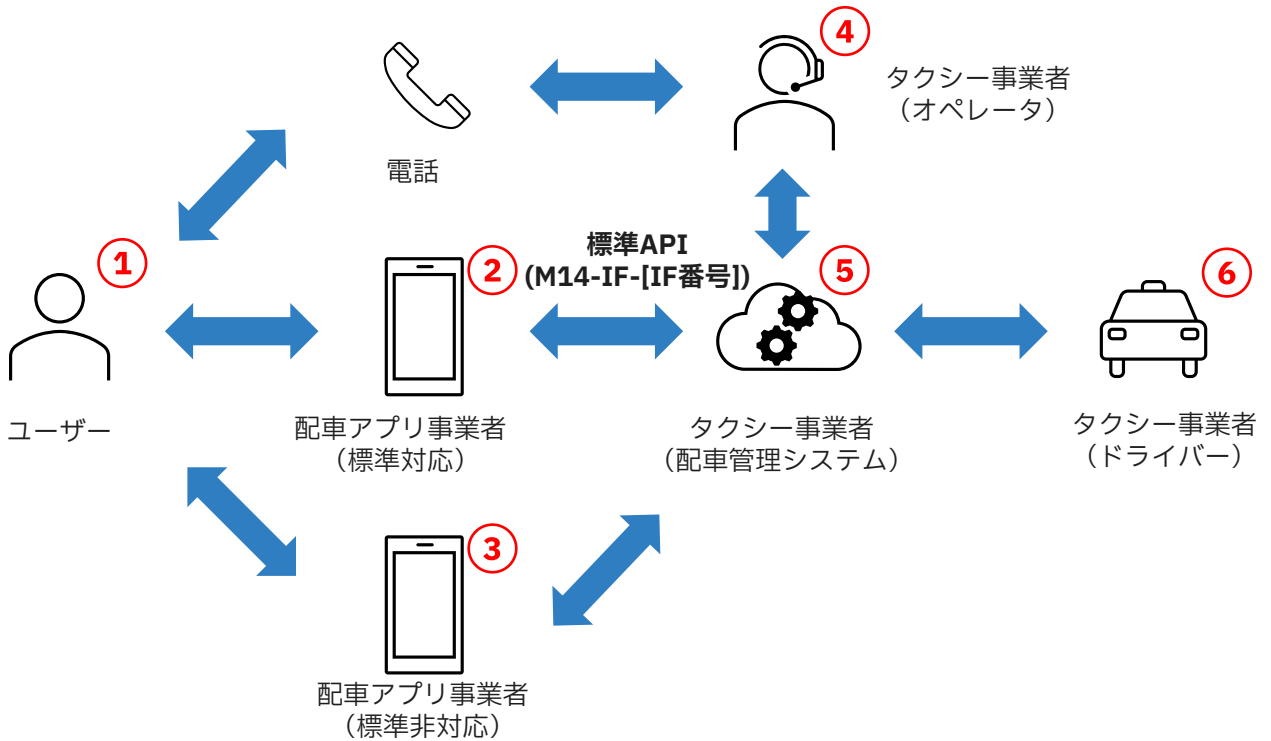
### 3.1 標準化の範囲と概要

本章で解説する標準化の範囲と概要を解説します。

#### 標準化の範囲

本プロジェクトでは、配車アプリとタクシー配車管理システム間におけるデータ連携の効率化と相互運用性の確保を目的とし、「タクシー配車システム連携API仕様」及び「標準業務モデル」を標準仕様として定義しています。

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、配車・運行管理業務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。



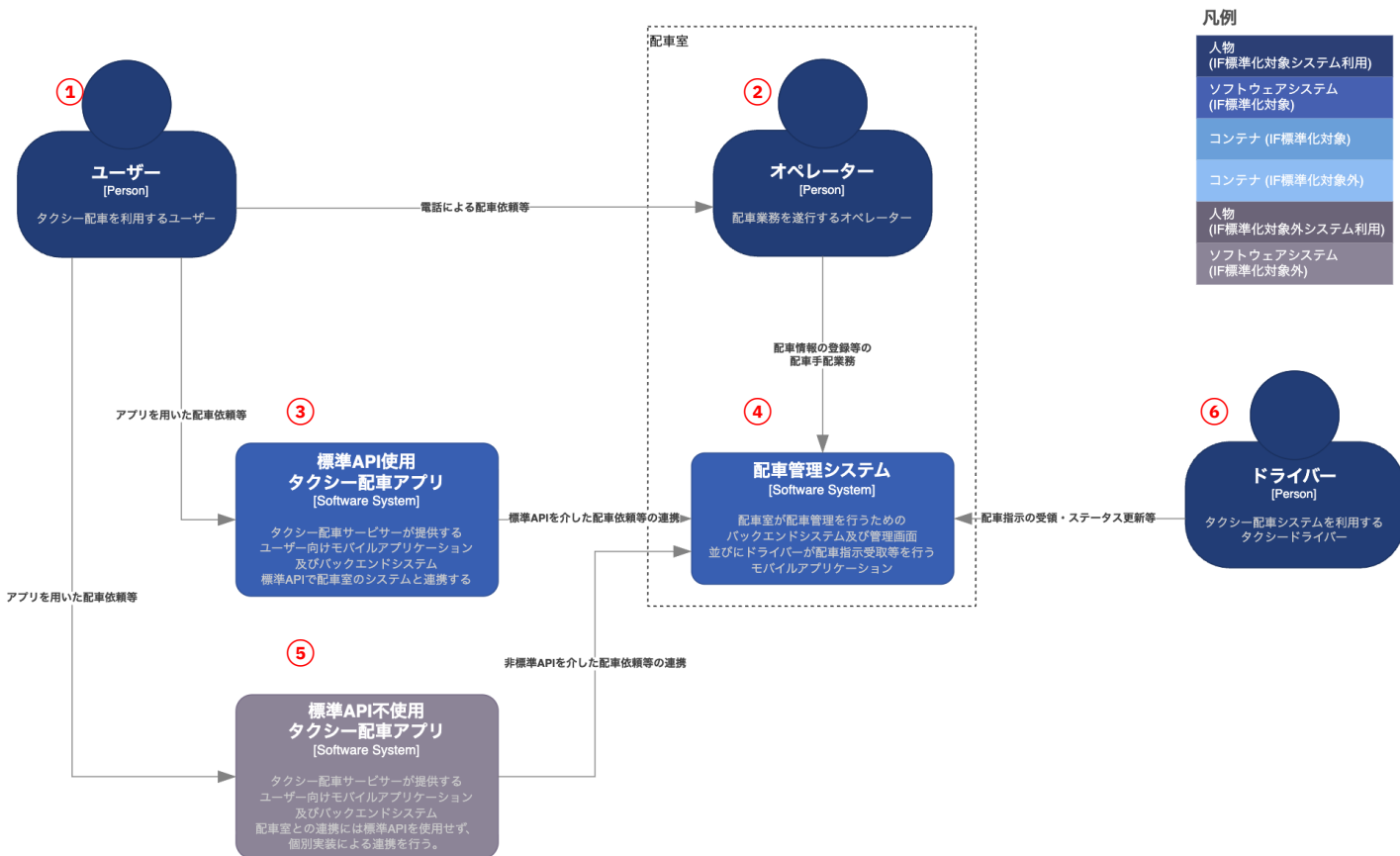
番号	アクター	役割
①	ユーザー	タクシー配車サービスの利用者。 配車アプリ（標準/非標準）や電話を通じて、タクシーの配車依頼、乗車、決済を行うエンドユーザー。
②	配車アプリ事業者 （標準対応）	標準APIを用いて連携する配車アプリ提供者。 「標準API」に準拠し、タクシー配車管理システムに対して配車依頼やステータス同期を行うアプリおよびバックエンドシステムを提供する。
③	配車アプリ事業者 （標準非対応）	独自仕様で配車管理システムと連携するアプリ提供者。 個別のAPI実装によってタクシー配車管理システムに対してステータス同期等を行う。 配車差配は独自ロジックで行うため、配車室のオペレーションを介さない。（※標準化の直接対象外だが、システム接続のエコシステムには含まれる）
④	タクシー事業者 （オペレータ）	配車業務の担当者。 電話及び配車アプリ（標準対応）からのリクエストの受付、配車不能時の対応、車両への指示などを行う。
⑤	タクシー事業者 （配車管理システム）	配車差配を支援するシステム。 配車依頼の受付、車両の引当（自動/手動）、ドライバーへの指示送信、運行ステータスの管理を行うバックエンドシステムおよびデータベース。
⑥	タクシー事業者 （ドライバー）	旅客運送を担う乗務員。 無線又はドライバーズアプリ等を通じて配車指示を受領し、迎車・実車・支払・降車などの実業務およびステータス更新を行う。

## 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を解説します。

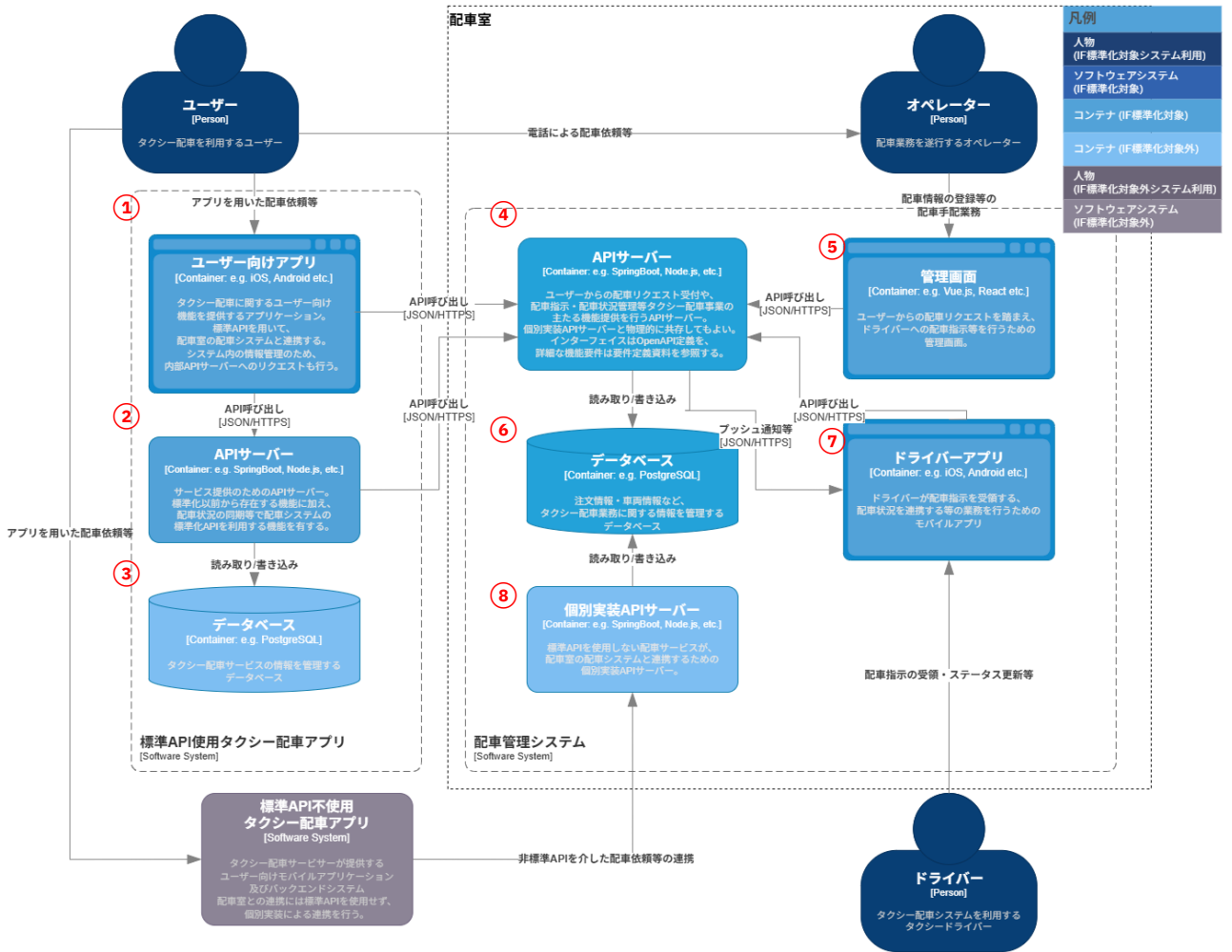
### 論理構成図

#### Level1: System Context



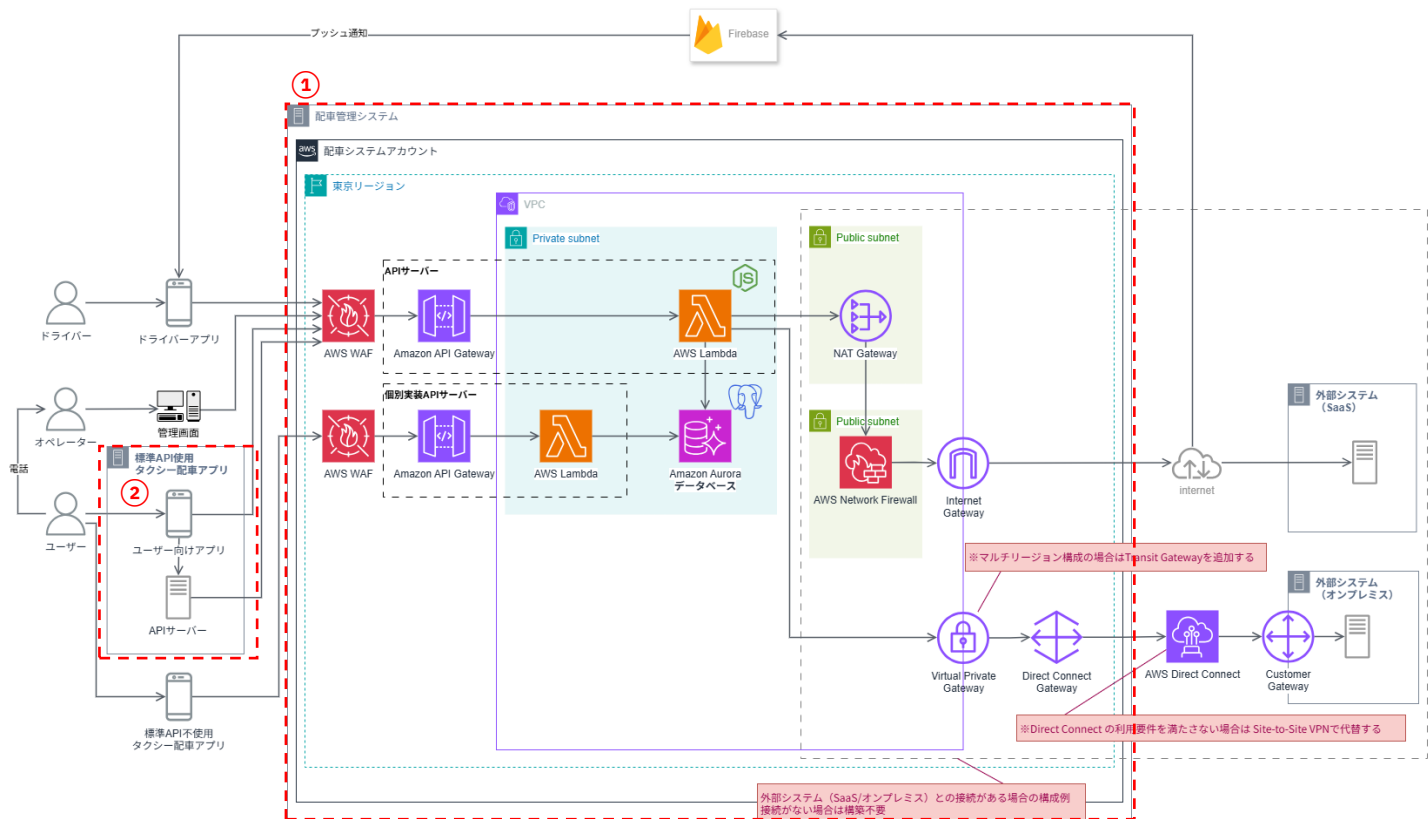
番号	名称	カテゴリ	説明
①	ユーザー	Person	タクシー配車を利用するエンドユーザー。アプリまたは電話で配車依頼を行います。
②	オペレーター	Person	配車室にて配車業務を遂行する担当者。電話又はアプリの依頼を受け、システムへ登録を行います。
③	標準API使用 タクシー配車アプリ	Software System	ユーザー向けアプリ及びバックエンド。標準APIを介して配車室のシステム (④) と連携します。
④	配車管理システム	Software System	配車管理用のバックエンド、管理画面及びドライバーズアプリを含む、配車室の中核システム。
⑤	標準API不使用 タクシー配車アプリ	Software System	ユーザー向けアプリ及びバックエンド。独自ロジックにより配車差配を行う。配車管理システム (④) とは情報連携のみを行う。
⑥	ドライバー	Person	タクシー配車管理システムを利用する運転手。配車指示の受領やステータス更新を行います。

# Level2: Container



番号	名称	所属システム	説明
①	ユーザー向けアプリ	標準API使用 タクシー配車アプリ	ユーザーに配車機能を提供するiOS/Android等のアプリ。標準APIを用いて配車管理システムと連携します。
②	APIサーバー	標準API使用 タクシー配車アプリ	サービス提供のためのサーバー。標準APIを利用して配車状況の同期等を行う機能を持ちます。
③	データベース	標準API使用 タクシー配車アプリ	タクシー配車サービスの情報を管理するPostgreSQL等のデータベース。
④	APIサーバー	配車管理システム	配車リクエスト受付や配車指示など、配車事業の主機能を提供するサーバー。
⑤	管理画面	配車管理システム	オペレーターが配車リクエストを確認し、ドライバーへ配車指示等を行うためのVue.js/React等の画面。
⑥	データベース	配車管理システム	注文情報や車両情報など、タクシー配車業務に関する情報を管理するデータベース。
⑦	ドライバーアプリ	配車管理システム	ドライバーが配車指示を受領し、配車状況を更新するためのiOS/Android等のアプリ。
⑧	個別実装APIサーバー	配車管理システム	標準APIを使用しない配車サービスが、配車管理システムと連携するための個別実装サーバー。

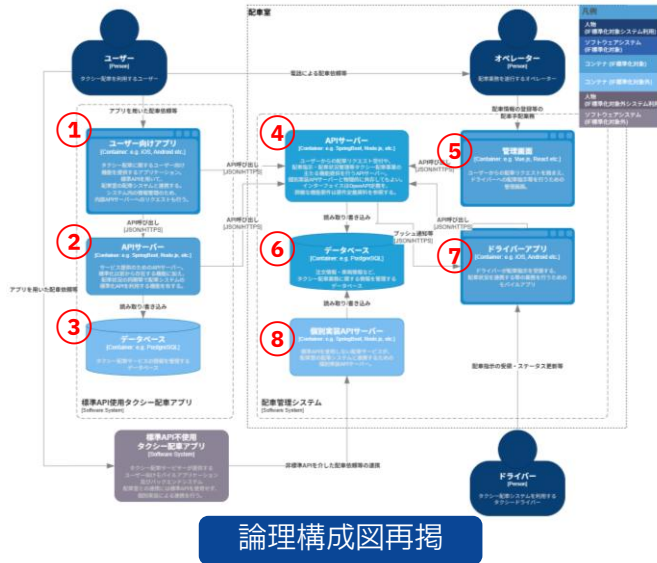
# インフラ構成図(実装例)



番号	システム領域名	詳細説明
①	配車管理システム	<ul style="list-style-type: none"> <li>リクエストの受付と検証: 外部（アプリや管理画面）からの通信をAWS WAFで受けて不正アクセスを遮断し、Amazon API Gatewayが窓口となって適切なAPIエンドポイントへ振り分けます。</li> <li>ビジネスロジックの実行（サーバーレス）: AWS Lambdaが起動し、Node.js等のプログラムによって「配車可能な車両の検索」や「予約確定処理」などの計算処理をオンデマンドで実行します。</li> <li>データの管理: 全ての配車ステータス、車両位置、ユーザー情報はAmazon Aurora（データベース）に保存・更新されます。</li> <li>プッシュ通知のトリガー: Lambdaでの処理結果（配車確定など）に基づき、Firebaseを経由してドライバーやユーザーの端末へリアルタイム通知を送る指示を出します。</li> </ul>
②	標準API使用 タクシー配車アプリ	<ul style="list-style-type: none"> <li>ユーザー操作の変換: ユーザーがモバイルアプリで行った「タクシーを呼ぶ」という操作を、システムが理解できるデータ形式（JSON）に変換します。</li> <li>標準APIによる連携: 独自に構築されたAPIサーバーを介して、中央の配車システム（①）が定めた「標準API」のルールに則って通信を行います。これにより、異なる事業者のアプリからでも同じ仕組みで配車依頼が可能になります。</li> <li>状態の同期: 中央システム（①）から返ってきた配車確定情報や車両の現在地データを受け取り、アプリ上の地図や画面表示を最新の状態に更新します。</li> </ul>

### 3.3. API仕様の解説

API仕様の一覧を掲載します。



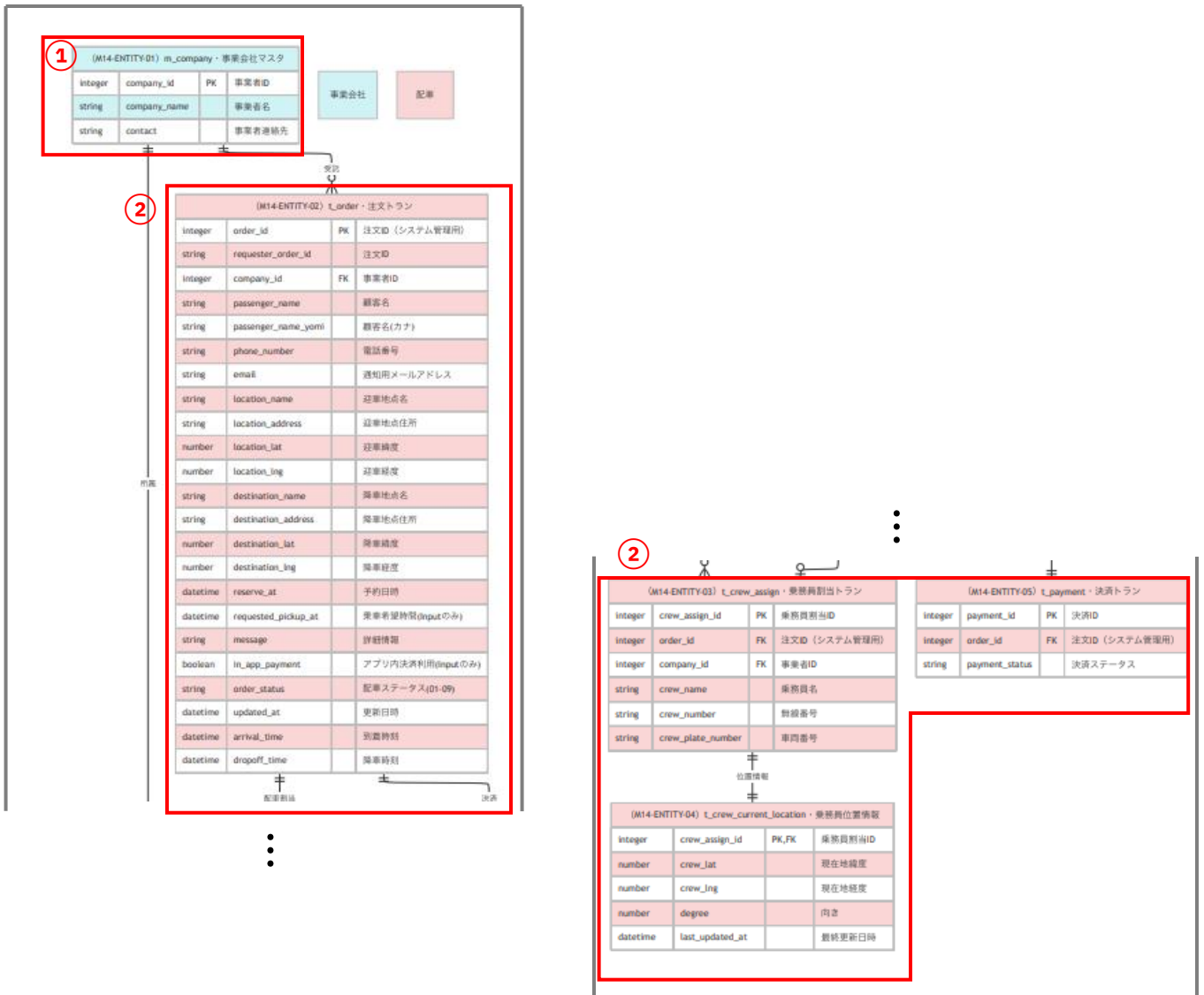
論理構成図再掲

APIサーバー	API名	API使用場所	機能説明	目的
タクシー配車	(M14-IF-010) 配車依頼	① → ④ ② → ④	ユーザーの現在地（緯度・経度）、事業者ID、電話番号、予約日時などの情報を送信し、配車をリクエストする。	<b>配車リクエストの自動登録</b> 電話を介さずアプリから直接、または外部サービス経由で配車室システムに注文情報を即時連携するため。
	(M14-IF-020) 配車情報複数取得	① → ④ ② → ④	複数の配車依頼ID (order_id) をカンマ区切りで指定し、それぞれの最新ステータスや車両位置情報を一括で取得する。	<b>複数注文の一括ステータス管理</b> ユーザーの利用履歴一覧や管理画面において、複数の注文状況を効率的に同期し、最新情報を表示するため。
	(M14-IF-030) 配車情報単一取得	① → ④ ② → ④	指定した1件の配車依頼IDに基づき、配車ステータス（迎車中、到着、実車中など）や乗務員情報、車両の現在地（緯度・経度・向き）を取得する。	<b>リアルタイムな車両情報の提供</b> 配車確定後に車両の接近状況をアプリの地図上に表示し、ユーザーが車両を見つけやすくするUXを実現するため。
	(M14-IF-040) 配車依頼キャンセル	① → ④ ② → ④	指定した配車依頼IDの注文を取り消す。	<b>キャンセル処理の同期化</b> ユーザー都合によるキャンセルを即座に配車室およびドライバーへ通知し、無駄な迎車走行を防止するため。



### 3.4. ERDの解説

ERDを掲載します。



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	事業会社・マスタ管理	・ m_company (事業会社マスタ)	「誰がサービスを提供するか」を管理する領域です。 タクシー事業者の名称や連絡先など、システムの利用主体となる事業者情報を保持します。これは配車依頼を受け付ける際の紐付けの起点となります。
②	配車・実行管理	・ t_order (注文トラン) ・ t_crew_assign (乗務員割当トラン) ・ t_crew_current_location (乗務員位置情報) ・ t_payment (決済トラン)	「いつ・どこで・誰が配車を遂行するか」を管理する領域です。 ユーザーからの注文(場所、時間、ステータス)を中核に、実際に担当するドライバーの割り当てやリアルタイムな車両位置、および決済状況を管理します。注文から完了、決済までの一連の業務フローを記録する動的なデータ群です。

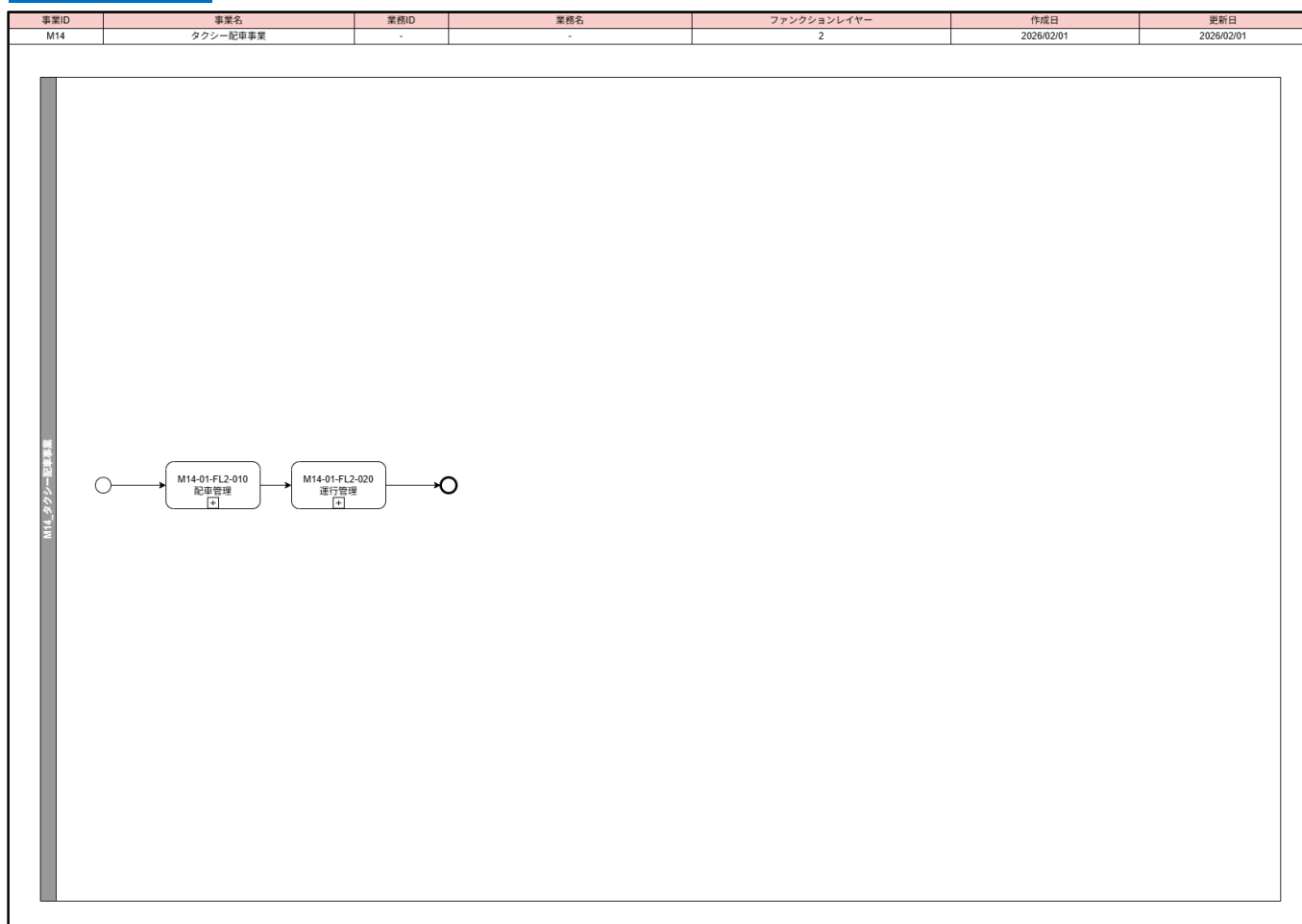
### 3.5. 業務フロー/業務一覧の解説

業務フロー/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

この業務フローは、タクシー配車業務の全体像を定義したものです。主に「配車管理」と「運行管理」の2つの主要な業務機能で構成されています。配車管理ではユーザーからの配車依頼を受け付けて車両を割り当て、運行管理では実際の車両走行や乗降状況を管理します。これらが連動することで、効率的な配車サービスを実現する仕組みです。

#### 業務フロー



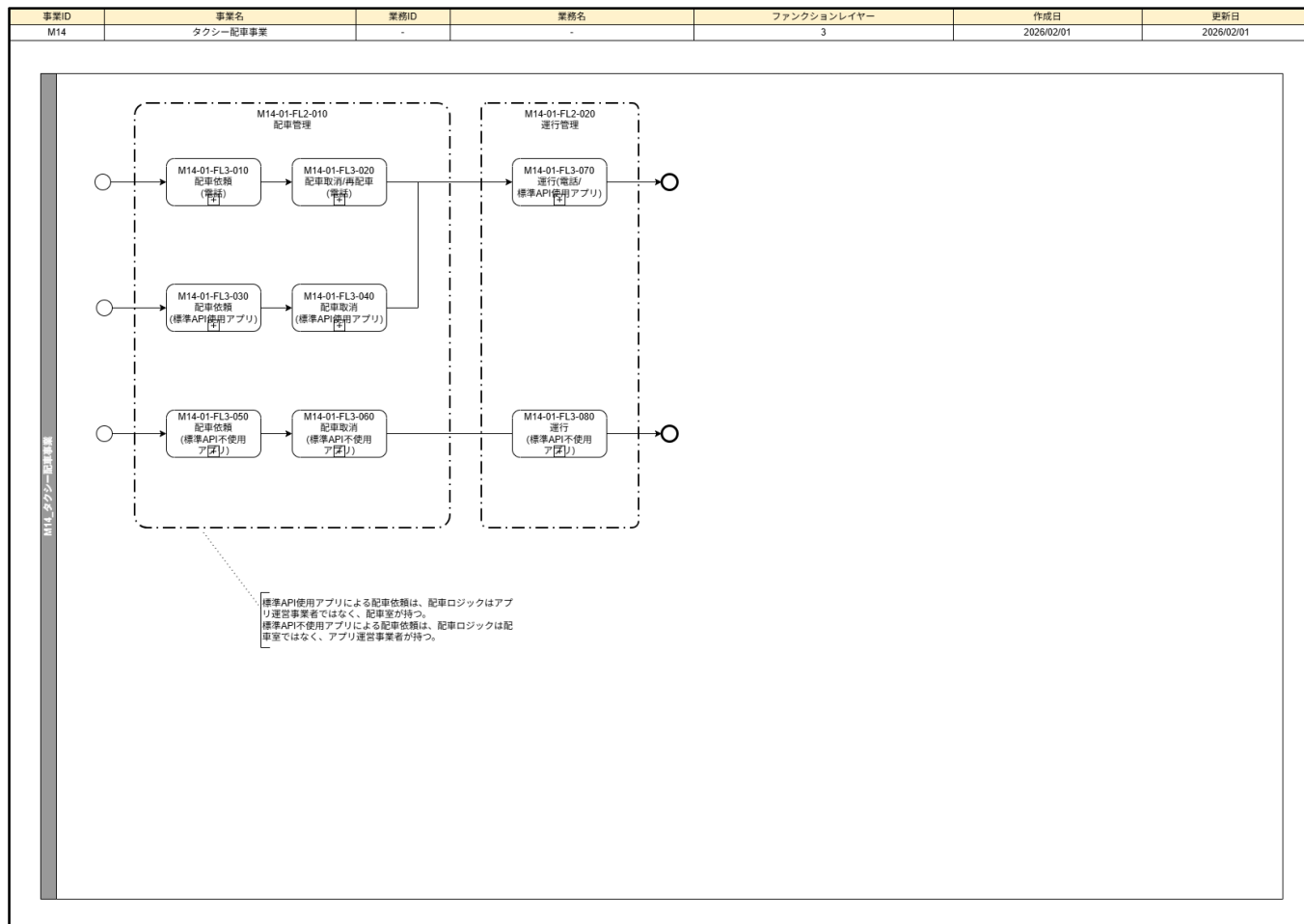
#### 業務一覧

ID	詳細事業機能	内容
M14-01-FL2-010	配車管理	配車室運営事業者とタクシードライバーが、乗客からの配車を管理する業務。
M14-01-FL2-020	運行管理	タクシードライバーが、乗客を目的地へ送迎する業務。

## FL(ファンクションレイヤー)3

この業務フローは、タクシー配車業務における「配車管理」と「運行管理」の各フェーズを詳細化したものです。配車管理では、電話や標準API対応・非対応アプリなど、多様なチャネルからの配車依頼や取消に対応する仕組みが定義されています。続く運行管理では、アプリ等を通じた運行状況の管理がなされます。特徴として、標準APIの利用有無により配車ロジックの保持主体が異なる点が明記されており、柔軟なシステム連携を想定した構造となっています。

### 業務フロー



### 業務一覧

ID	業務機能	内容
M14-01-FL3-010	配車依頼(電話)	電話を利用した配車依頼を管理する業務。
M14-01-FL3-020	配車取消/再配車(電話)	電話を利用した配車取消/再配置依頼を管理する業務。
M14-01-FL3-030	配車依頼(標準API使用アプリ)	標準APIを使用しているアプリを利用した配車依頼を管理する作業。
M14-01-FL3-040	配車取消(標準API使用アプリ)	標準APIを使用しているアプリを利用した配車取消を管理する作業。
M14-01-FL3-050	配車依頼(標準API不使用アプリ)	標準APIを使用していないアプリを利用した配車依頼を管理する作業。
M14-01-FL3-060	配車取消(標準API不使用アプリ)	標準APIを使用していないアプリを利用した配車取消依頼を管理する作業。
M14-01-FL3-070	運行(電話/標準API使用アプリ)	タクシードライバーが、乗客を目的地へ送迎する業務。(標準APIを使用しているアプリによる予約の場合)
M14-01-FL3-080	運行(標準API不使用アプリ)	タクシードライバーが、乗客を目的地へ送迎する業務。(標準APIを使用していないアプリによる予約の場合)

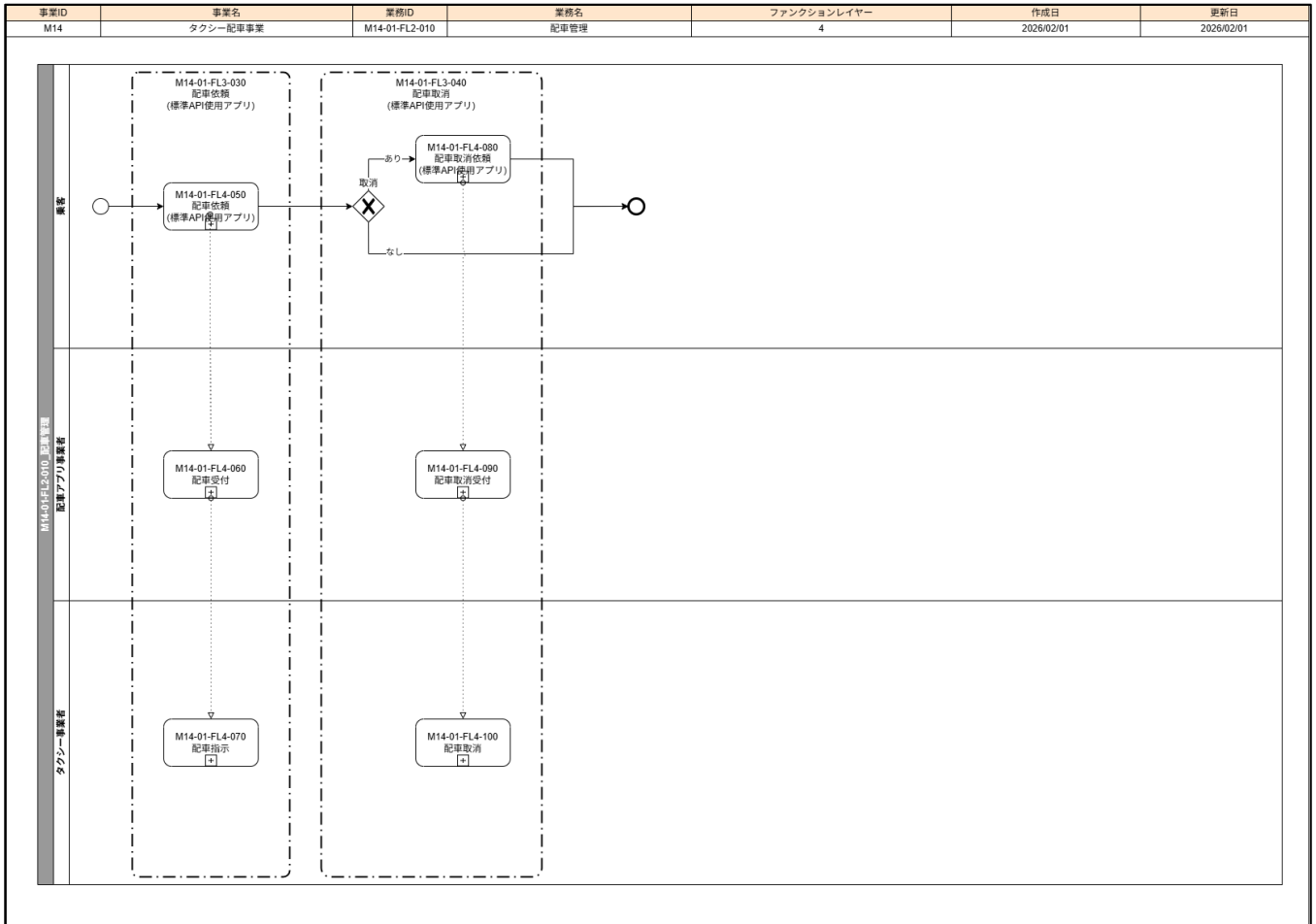
## FL(ファンクションレイヤー)4

この業務フローは、タクシー配車業務における配車管理を定義したものです。

ユーザーがアプリで配車を依頼すると、その要求はまずアプリ事業者のシステムで受け付けられ、そこからタクシー事業者の配車室へと伝達されます。配車室では、車両の割り当てを行う「配車指示」を出し、タクシー車両側でその指示を受領することで配車が確定します。アプリ事業者が受け付けたリクエストは配車室を介してやり取りされることを表しています。

### 業務フロー

### —配車依頼(標準API利用アプリ)抜粋—



### 業務一覧

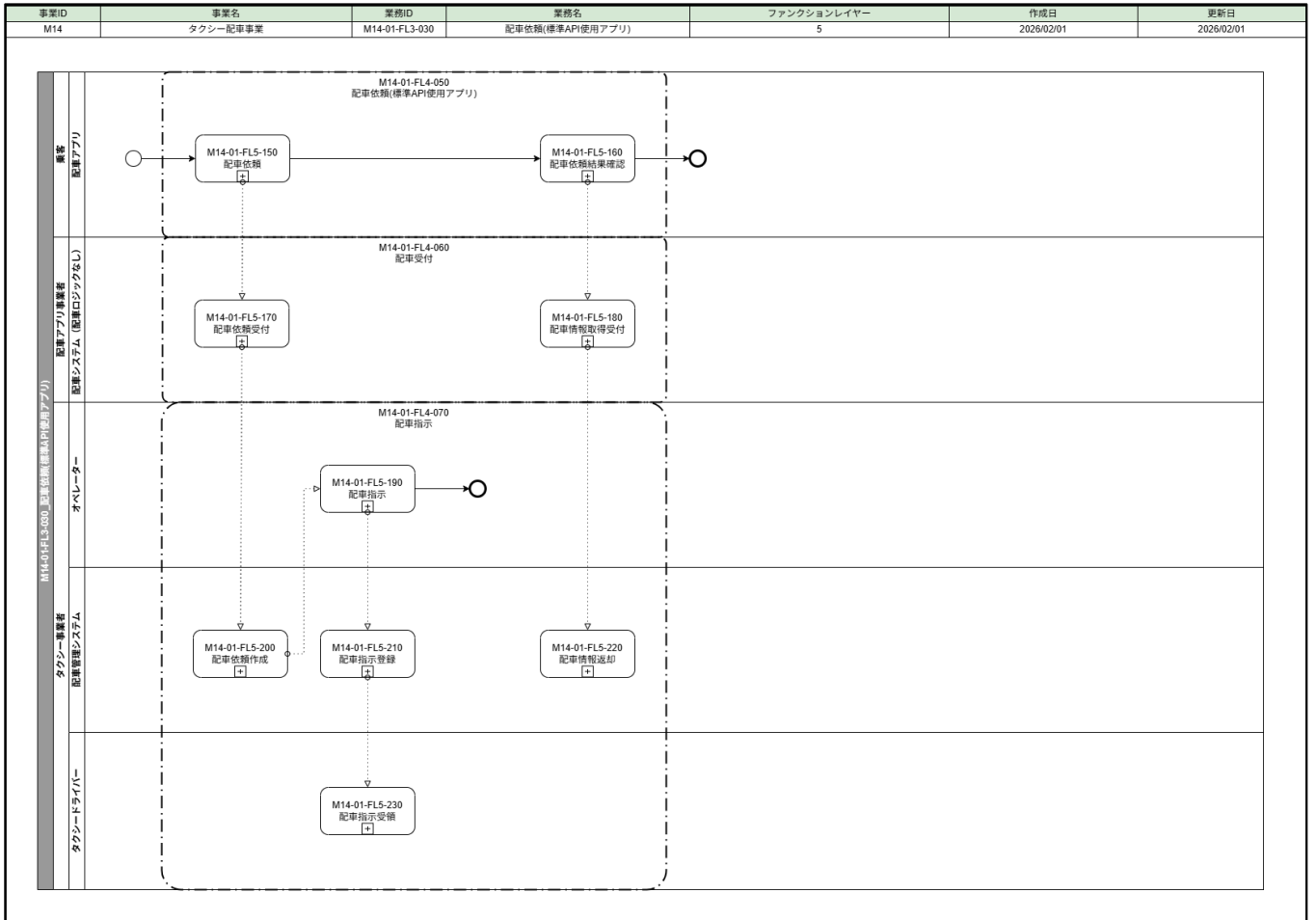
ID	詳細業務機能	内容
M14-01-FL4-050	配車依頼 (標準API使用アプリ)	乗客が、標準APIを使用するアプリから、配車依頼をする作業。
M14-01-FL4-060	配車受付	配車アプリ事業者が、乗客からの配車依頼を受け付けて、配車室へ配車依頼する業務。
M14-01-FL4-070	配車指示	配車室が、配車システムからの依頼を管理して、タクシードライバーへ配車指示する業務。
M14-01-FL4-080	配車取消依頼 (標準API使用アプリ)	乗客が、標準APIを使用したアプリを用いて配車室へ配車取消依頼をする業務。
M14-01-FL4-090	配車取消受付	配車アプリ事業者が、乗客からの標準APIを使用したアプリによる配車取消依頼を受け付ける業務。
M14-01-FL4-100	配車取消	配車室運営事業者が、乗客からの標準APIを使用したアプリによる配車取消を行う業務。

## FL(ファンクションレイヤー)5

この業務フローは、タクシー配車業務における「標準APIを使用するアプリ」からの配車依頼プロセスを詳細に定義したものです。ユーザーがアプリで依頼を行うと、情報は「アプリ事業者の配車アプリ」を經由して「タクシー事業者の配車室」へと伝達されます。配車室のオペレーターが配車管理システムにより内容を確認して配車指示を出し、ドライバーがそれを受領することで予約が確定します。システムと人の判断（オペレーター）の双方によりリクエストを処理します。

### 業務フロー

### ー配車依頼(標準API利用アプリ)抜粋ー



### 業務一覧

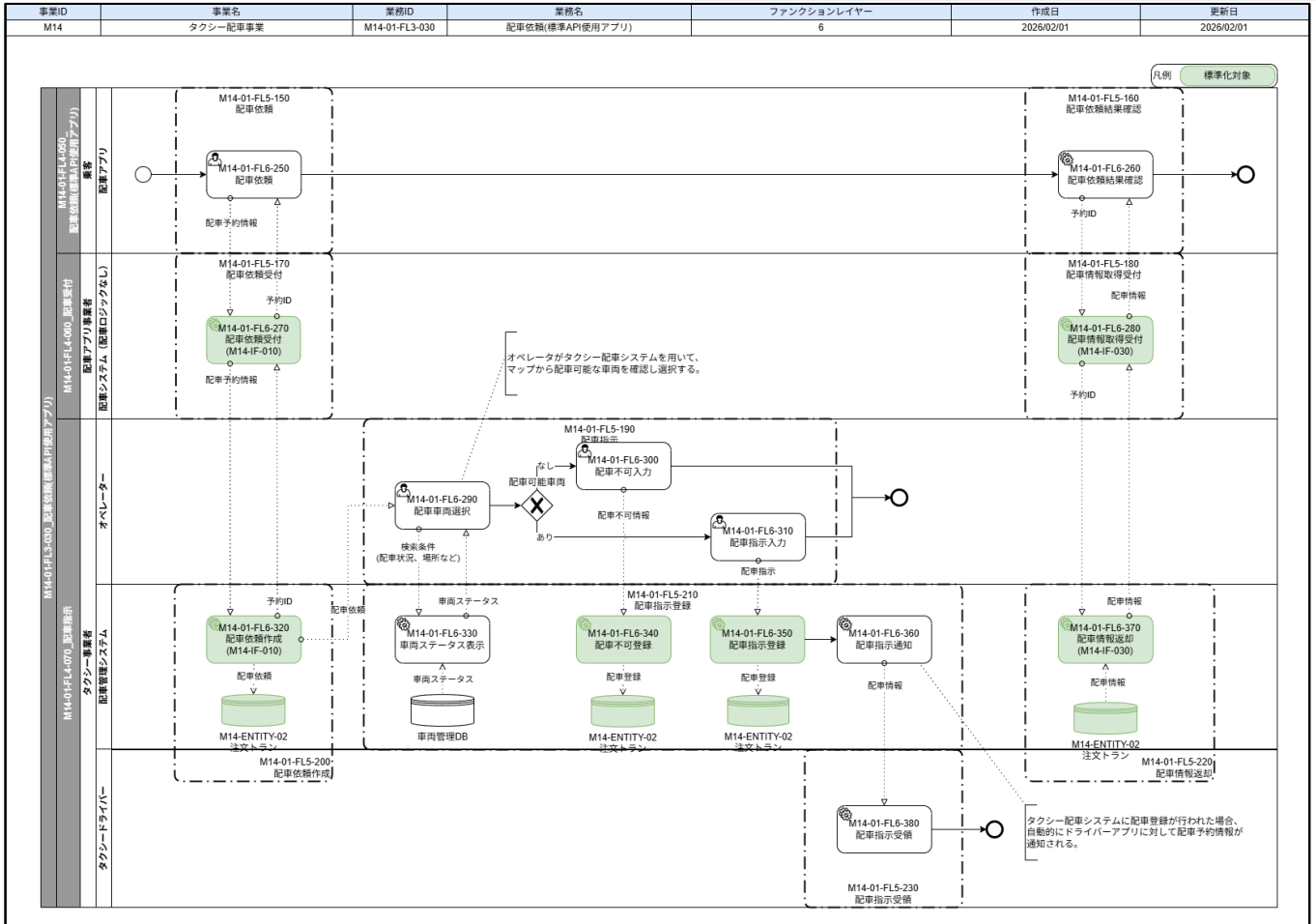
ID	単位作業	内容
M14-01-FL5-150	配車依頼	乗客が、標準APIを使用するアプリから、配車アプリ事業者へ配車依頼をする作業。
M14-01-FL5-160	配車依頼結果確認	乗客が、配車依頼の結果を確認する作業。
M14-01-FL5-170	配車依頼受付	配車アプリ事業者の配車システムが、乗客からの配車依頼を受け付けて、配車室へ配車依頼する業務。
M14-01-FL5-180	配車情報取得受付	配車アプリ事業者の配車システムが、乗客からの配車情報取得依頼を受領し、配車システムに対して取得依頼を行い、返答を乗客に伝達する業務。
M14-01-FL5-190	配車指示	配車室オペレーターが、乗客からの配車依頼を確認し、配車車両を決定し、登録を行う業務。
M14-01-FL5-200	配車依頼作成	配車システムが、乗客から配車依頼を受領し、注文トランへ配車依頼情報を入力する業務。
M14-01-FL5-210	配車指示登録	配車システムが、配車室オペレーターからの配車指示を登録し、ドライバーへ通知を行う業務。
M14-01-FL5-220	配車情報返却	配車システムが、配車アプリからの配車情報取得依頼を受けて、指定車両の配車状況を注文トランから取得し、配車アプリに対して返答する業務。
M14-01-FL5-230	配車指示受領	タクシードライバーが、配車室からの配車指示を受領する業務。

# FL(ファンクションレイヤー)6

この業務フローは、タクシー配車業務における「配車依頼」の具体的なシステム処理を詳細に定義したものです。乗客がアプリから配車を依頼すると、そのリクエストはアプリ事業者のシステムを経由して配車室へと伝達されます。配車室のオペレーターは、配車管理システム上で車両ステータスを確認しながら最適な車両を選択し、配車指示を入力します。これらのシステム間の情報のやり取りには、標準APIが用いられています。

## 業務フロー

## —配車依頼(標準API利用アプリ)抜粋—



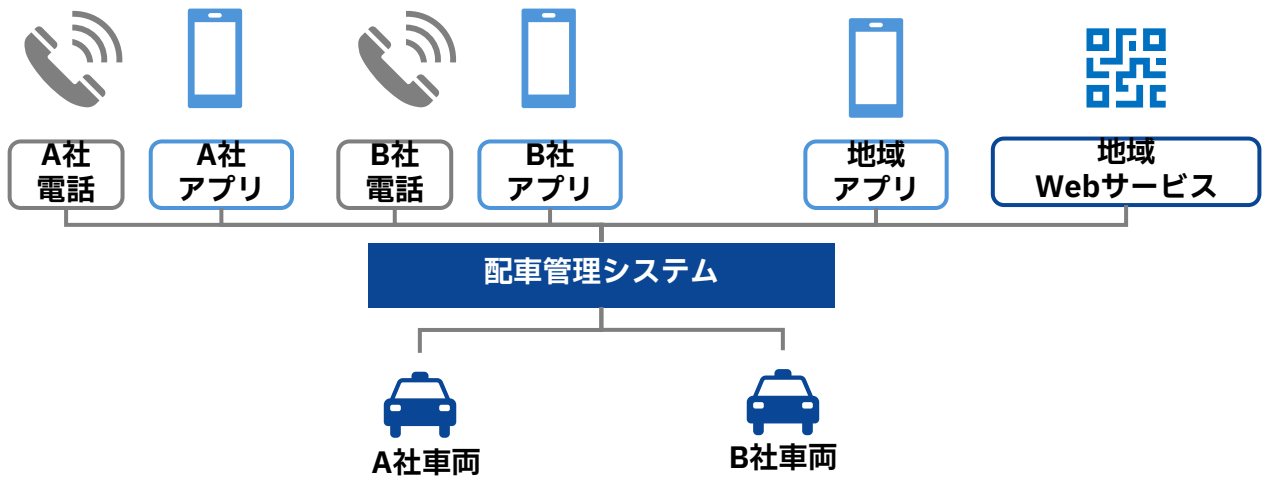
## 業務一覧

ID	要素作業	内容
M14-01-FL6-250	配車依頼	乗客が、標準APIを使用するアプリから、配車アプリ事業者へ配車依頼をする作業。
M14-01-FL6-260	配車依頼結果確認	乗客が、配車依頼の結果を確認する作業。
M14-01-FL6-270	配車依頼受付	配車アプリ事業者の配車システムが、乗客からの配車依頼を受け付けて、配車室へ配車依頼する業務。
M14-01-FL6-280	配車情報取得受付	配車アプリ事業者の配車システムが、乗客からの配車情報取得依頼を受領し、配車システムに対して取得依頼を行い、返答を乗客に伝達する業務。
M14-01-FL6-290	配車車両選択	配車室オペレーターが、配車システムを確認して、配車車両を決定する業務。
M14-01-FL6-300	配車不可入力	配車室オペレーターが、乗客からの配車依頼を受け付けられない車両がなかった場合に、配車システムに対して、配車不可入力を行う業務。
M14-01-FL6-310	配車指示入力	配車室オペレーターが、配車システムに対して、乗客の送迎を行うタクシー車両を指示する業務。
M14-01-FL6-320	配車依頼作成	配車システムが、乗客から配車依頼を受領し、注文トランへ配車依頼情報を入力する業務。
M14-01-FL6-330	車両ステータス表示	配車システムが、配車室に対して、管理車両のステータスの表示を行う業務。
M14-01-FL6-340	配車不可登録	配車システムが、配車室オペレーターからの配車不可入力を受けて、配車不可情報を注文トランに登録する業務。
M14-01-FL6-350	配車指示登録	配車システムが、配車室オペレーターからの配車指示を受けて、配車登録情報を注文トランに登録する業務。
M14-01-FL6-360	配車指示通知	配車システムが、配車室オペレーターからの配車指示を受けて、配車指示をタクシードライバーに対して行う業務。
M14-01-FL6-370	配車情報返却	配車システムが、配車アプリからの配車情報取得依頼を受けて、指定車両の配車状況を注文トランから取得し、配車アプリに対して返答する業務。
M14-01-FL6-380	配車指示受領	タクシードライバーが、配車室からの配車指示を受領する業務。

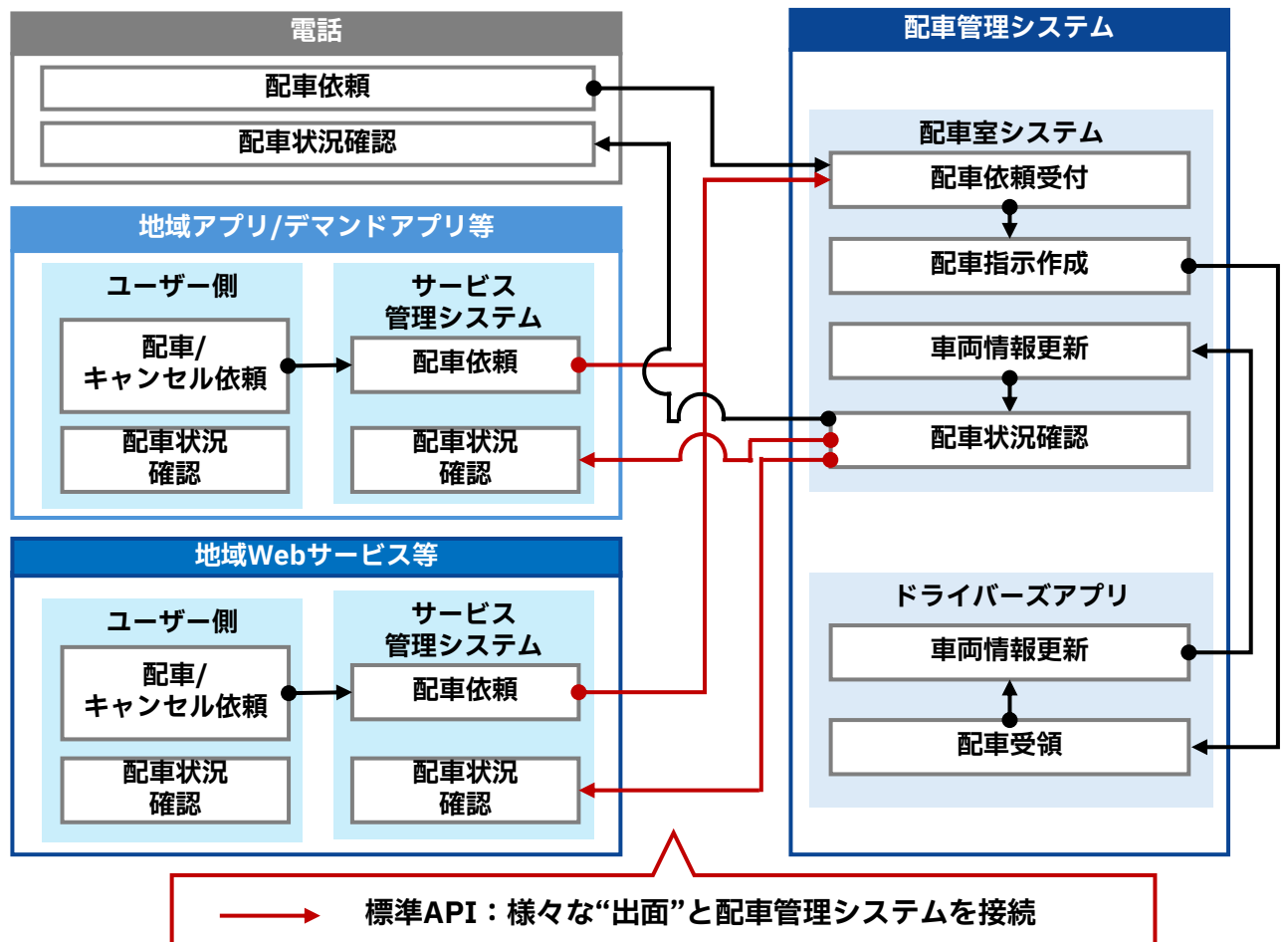
### 3.6. ユースケース

標準APIのユースケースとして、複数のタクシー事業者や地域アプリを「配車管理システム」へ集約する際の共通のインターフェースとして利用することが想定されます。これにより、電話やアプリといった多様な受付窓口から入る配車依頼を一元管理し、各社の車両へ効率的に指示を送ることが可能です。標準APIの導入により個別開発の不要な低コストの実装が実現し、地域の移動利便性と運行効率を同時に向上できます。

#### 配車室のイメージ



#### 配車室のシステムにおける標準APIの活用





**COMmmONS**

by MLIT

タクシー配車システム連携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：株式会社電脳交通