

第1版	: 初版発行
第2版 (2022/12/26版)	: 修正 (p.3-9追加)
第3版 (2023/2/2版)	: 修正 (p.33追加、誤記修正)

# パーソナルデータ連携モジュール 説明資料

2023年2月2日

日本電気株式会社

# 目次

## 1. コンセプト

1. パーソナルデータ連携モジュールの概要
2. パーソナルデータ流通の概観
  - ・ step1：サービス定義
  - ・ step2：同意
  - ・ step3：蓄積
  - ・ step4：共有

## 2. アーキテクチャ

1. 論理構成
  - ・ システム構成
  - ・ PxR-Block詳細
2. 物理構成（例）

## 3. アクター構成

1. アクター定義
2. アクター関係

## 4. 機能

1. 流通制御SP向け機能
2. 領域運営SP向け機能
3. アプリケーションP向け機能
4. 個人向け機能

## 5. 運用

1. 運用フロー一覧
2. 運用フロー詳細

## 6. 技術

1. 個人認証
2. 組織認証
3. カタログ管理
4. データ管理

### <別紙>

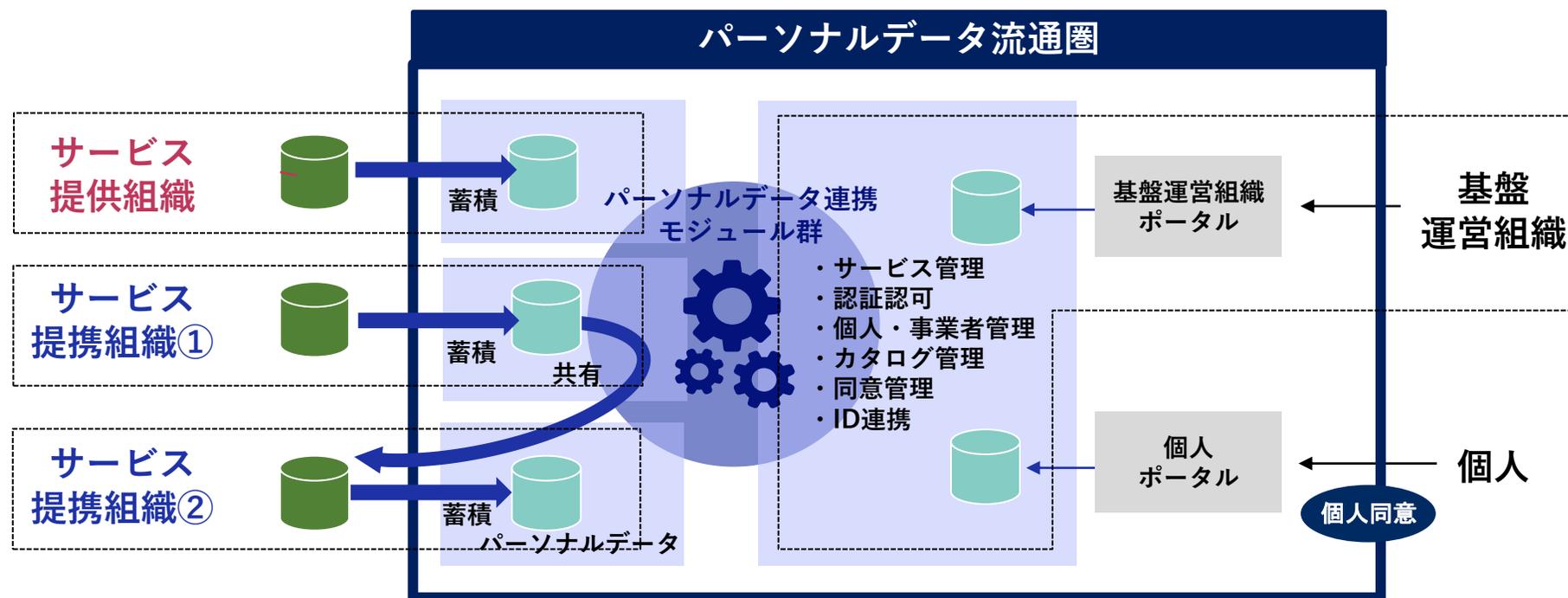
- ・ パーソナルデータ連携モジュール\_公開APIリスト
- ・ パーソナルデータ連携モジュール\_機能要件テンプレート

# 1. コンセプト

---

# 1-1. パーソナルデータ連携モジュールの概要

パーソナルデータ連携モジュールとは、サービス事業者が保有しているパーソナルデータを、個人同意に基づき組織をまたがって共有できる、トラスト層/データ連携層の機能群。



- ・蓄積：個人の同意に基づき、パーソナルデータを他組織からアクセス可能な状態にすること
- ・共有：個人の同意に基づき、他組織が蓄積したパーソナルデータにアクセスすること

## 1-2. パーソナルデータ流通の概観

4つのstepを通じて、各アクターとパーソナルデータ連携モジュール群（API群）が連携し協調動作することでパーソナルデータ流通圏を形成。

### step1 サービス定義

複数のアプリを連携させるサービスを定義する

### step2 同意

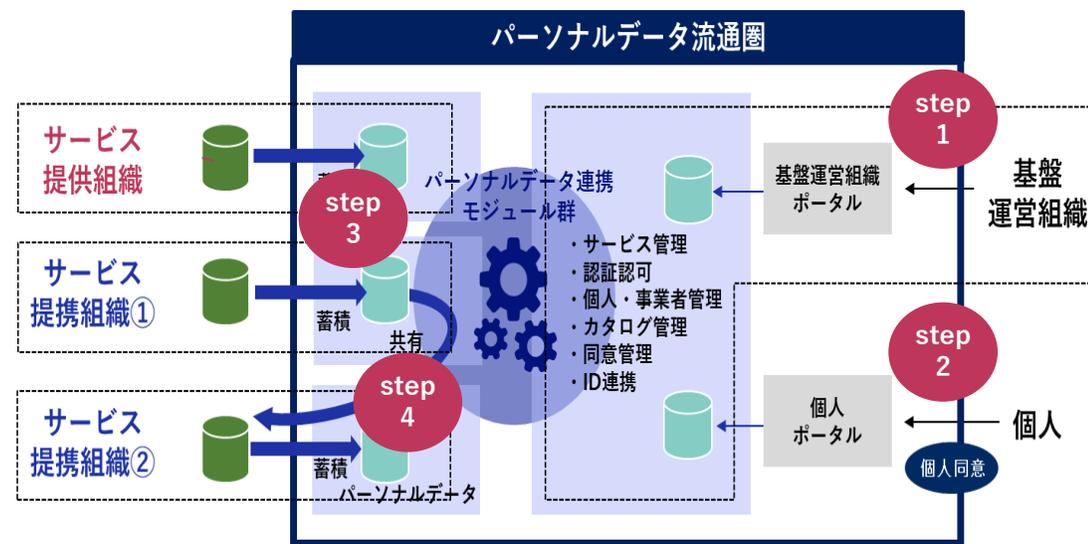
定義したサービスに対する個人同意を取得する

### step3 蓄積

アプリが収集したパーソナルデータのうち、蓄積同意を取得したデータのみ対応するPxRテナントに蓄積する

### step4 共有

アプリが別のアプリにより収集、蓄積されたパーソナルデータを取得する



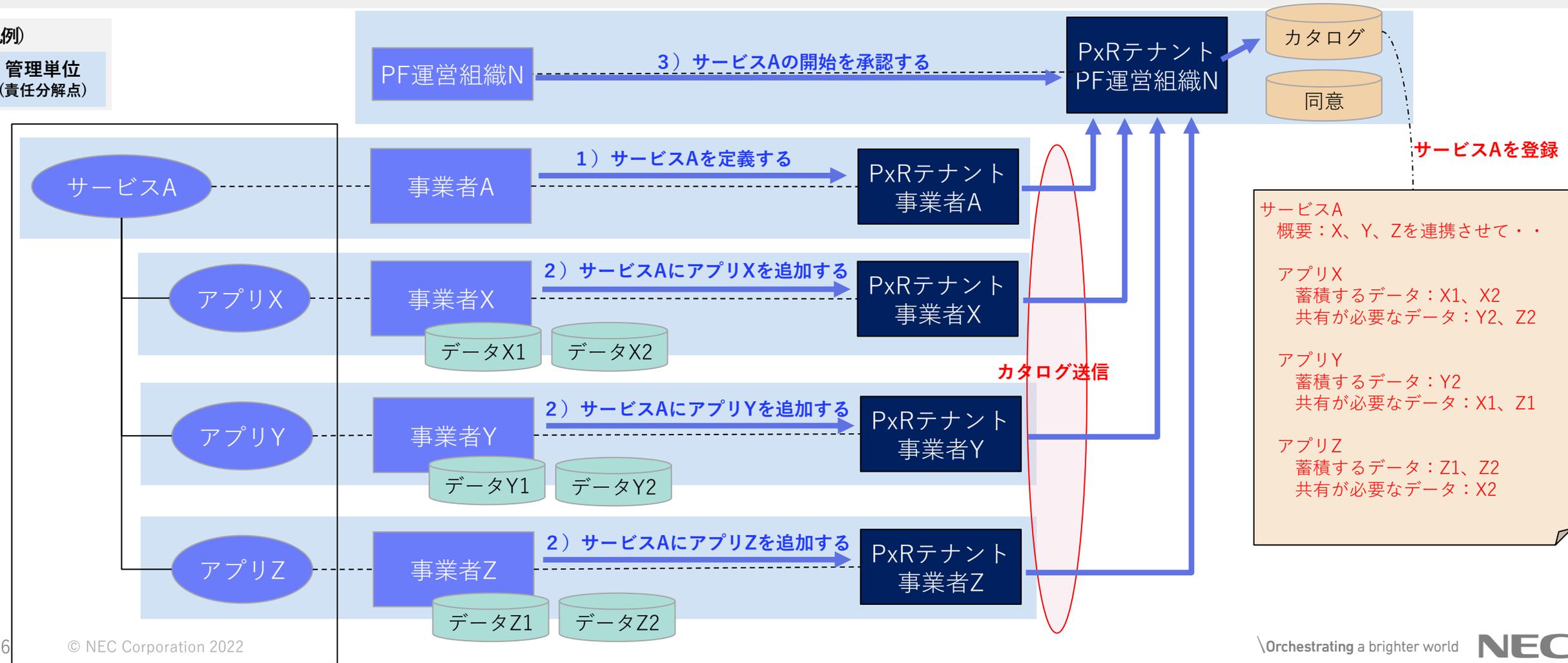
# 1-2. パーソナルデータ流通の概観

事例：3つのアプリを連携させる（step1：サービス定義）

step1：サービス定義では、複数のアプリを連携させるサービスを定義。サービス定義にアプリを連携させる目的を明記することで、個人がデータ連携の目的を理解することを容易にすることが可能。

凡例

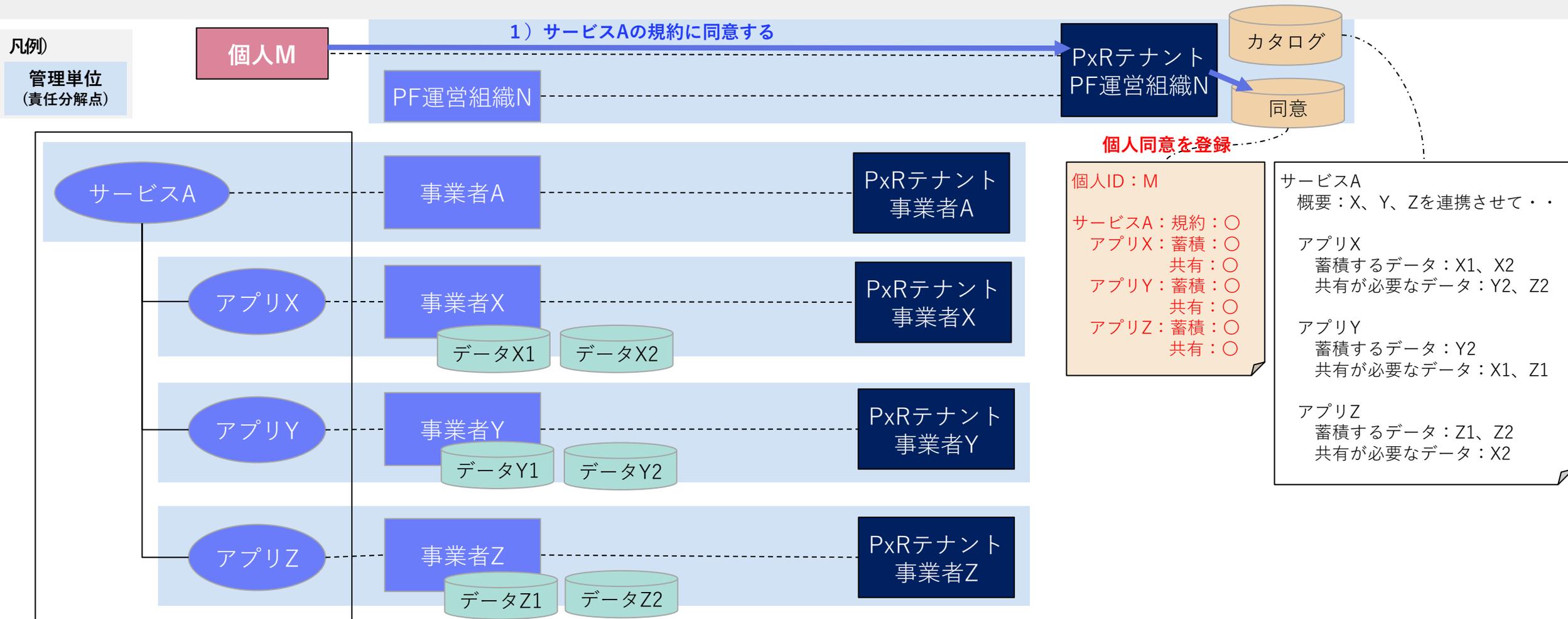
管理単位  
(責任分解点)



# 1-2. パーソナルデータ流通の概観

事例：3つのアプリを連携させる（step2：同意）

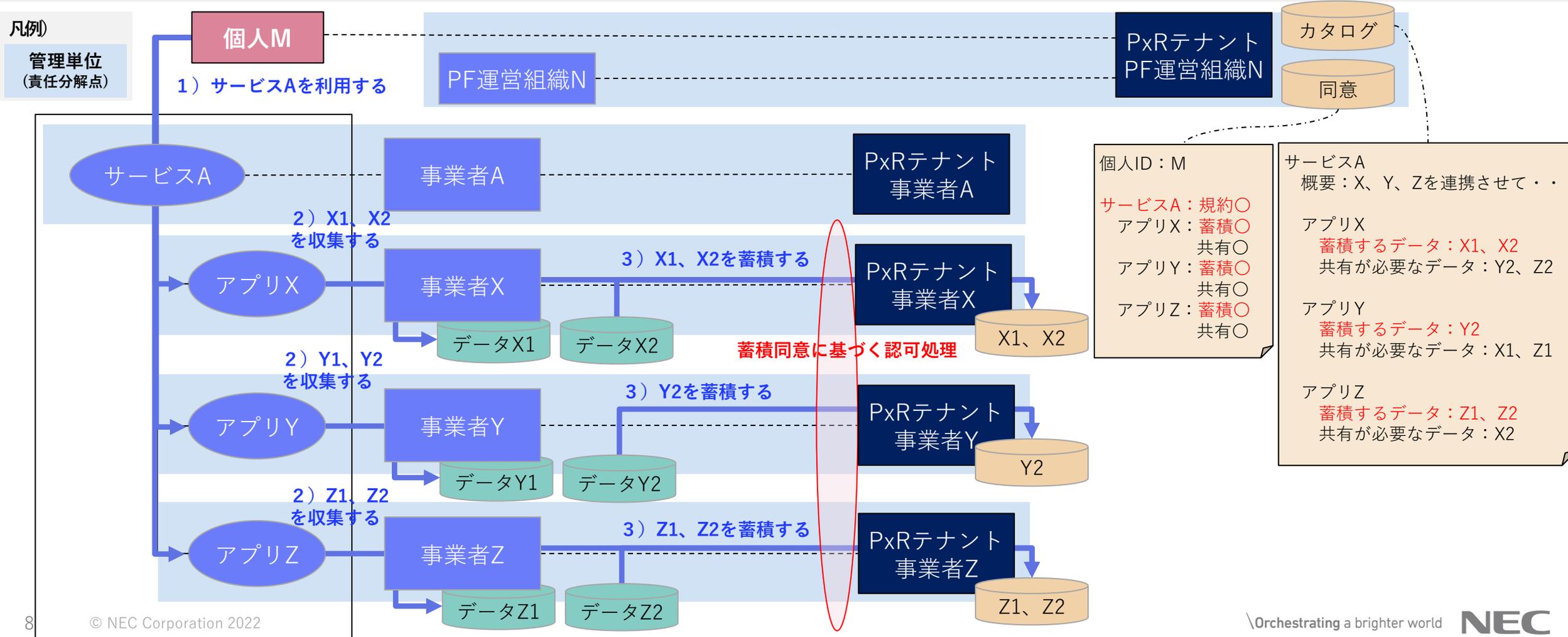
step2：同意では、サービスに対する個人同意を取得します。個人同意の取得をPF運営組織がサービス横断的に行うことで、個人の同意管理UXをサービス横断的に統一し、パーソナルデータ流通に対する安心感を高める。



# 1-2. パーソナルデータ流通の概観

事例：3つのアプリを連携させる（step3：蓄積）

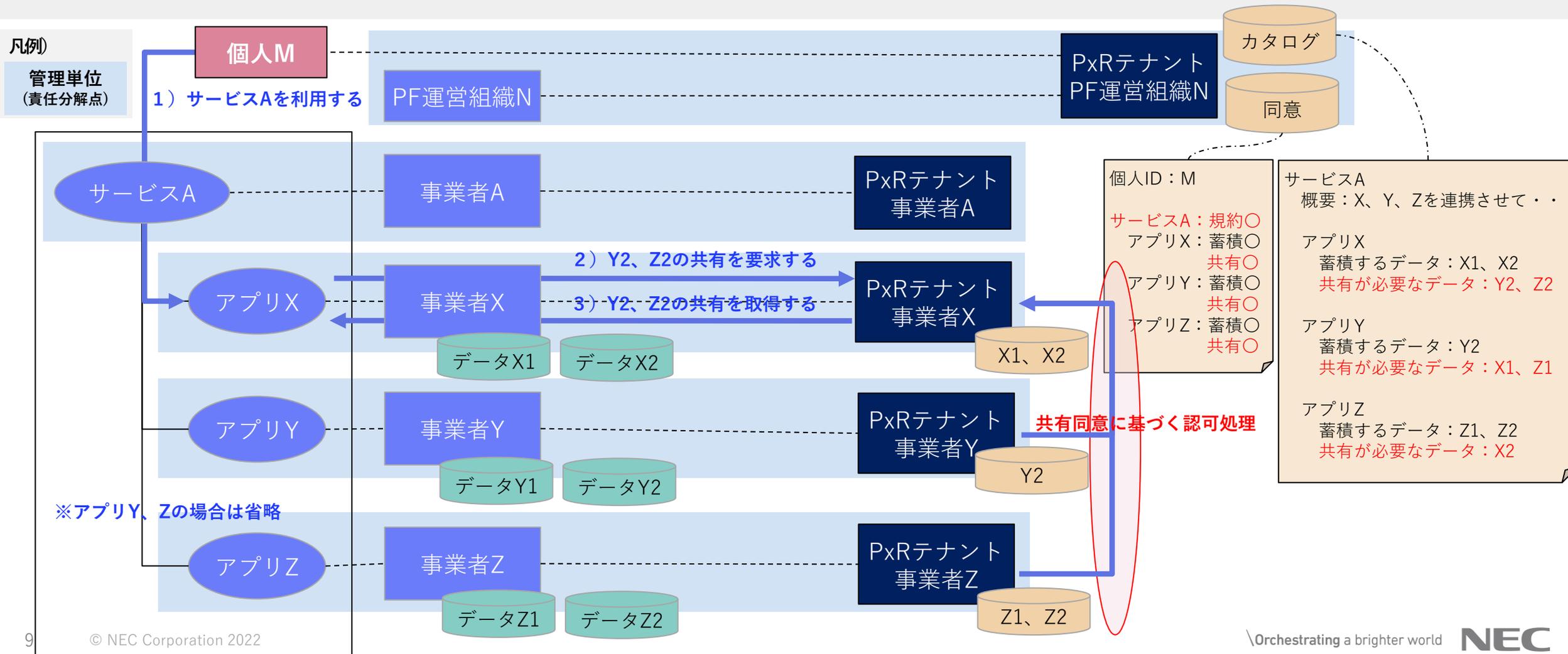
step3：蓄積では、アプリが収集したパーソナルデータの内、蓄積同意を取得したデータのみを対応するPxRテナントに蓄積する。テナント内の各事業者用の環境（Block）は、各事業者で管理可能であり、蓄積データの分散管理が実現される。



# 1-2. パーソナルデータ流通の概観

事例：3つのアプリを連携させる（step4：共有）

step4：共有では、アプリが他のアプリが蓄積したパーソナルデータを取得する。

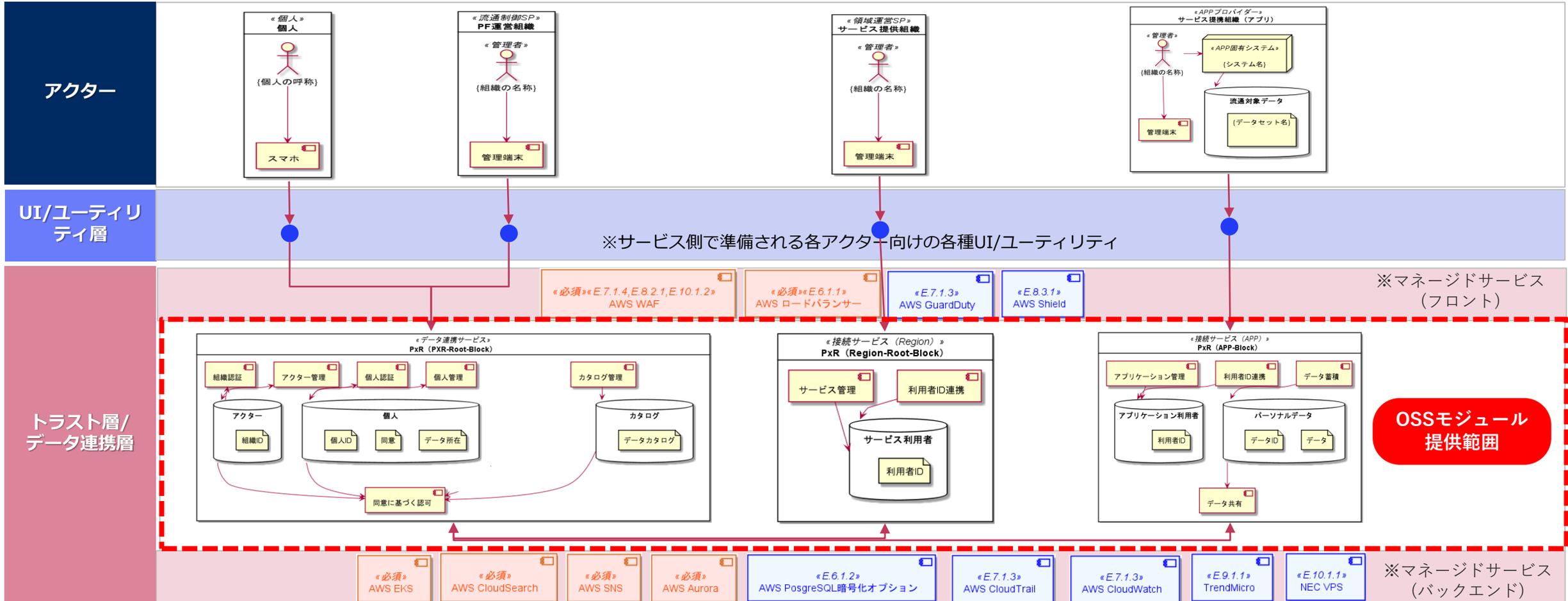


## 2. アーキテクチャ

---

# 2-1. 論理構成 システム構成

各アクターに提供するシステム要素を**PxR-Block**と呼び、これらが協調動作することでPxRエコシステムが形成される。個人や各組織は各種ポータルシステム等を通じてPxR-Blockを利用することで、容易に連携が可能。



## 2-1. 論理構成

### PxR-Block詳細（提供機能）

◆ 各PxR-Blockが提供する機能は下記の通り

No.	提供機能（PxR-Service）			PxR-Block		
	名称	和名	説明	PxR-Root-Block	Region-Root-Block	APP-Block
1	operator	オペレーター	オペレーター(運営メンバー、個人)のアカウント管理と認証を行います	○	○	○
2	catalog	カタログ	My-Condition-Dataカタログの管理を行います	○		
3	catalog-update	カタログ更新	My-Condition-Dataカタログのうち、承認処理等追加で特殊な手順を踏むカタログの管理を行います	○		
4	pxr-block-proxy	PxR-Block-Proxy	PxR-Blockのサービスへ各種メソッドでProxyを行います	○	○	○
5	notification	通知	オペレーターへ通知や承認要求を行います	○	○	○
6	book-manage	My-Condition-Book管理	My-Condition-Bookの管理を行います	○		
7	book-operate	My-Condition-Book運用	My-Condition-Bookへデータの蓄積と共有を行います		○	○
8	identity-verify	本人性確認	本人性の確認を行います	○		
9	ctoken-ledger	CToken台帳	CToken台帳の管理を行います	○		
10	local-ctoken	Local-CToken	Local-Ctoken-Storeの更新とCToken台帳への差分送信を行います			○
11	certification-authority	認証局	証明書の生成や取得、検証を行います	○		
12	certificate-manage	証明書管理	証明書の保存を行います	○	○	○
13	access-control-manage	アクセス制御管理	API間のアクセス設定とデータ操作定義を確認し、APIトークンの生成指示を行います	○		
14	access-control	アクセス制御	APIトークンの生成と照合を行います	○	○	○
15	binary-manage	バイナリ管理	バイナリファイルのアップロード、ダウンロードを行います			○

## 2-1. 論理構成

### PxR-Block詳細（公開API一覧）

#### ◆ 各PxR-Blockが提供する公開APIは下記別紙を参照のこと

- 「パーソナルデータ連携モジュール\_公開APIリスト」

#### ◆ ソフトウェア構成

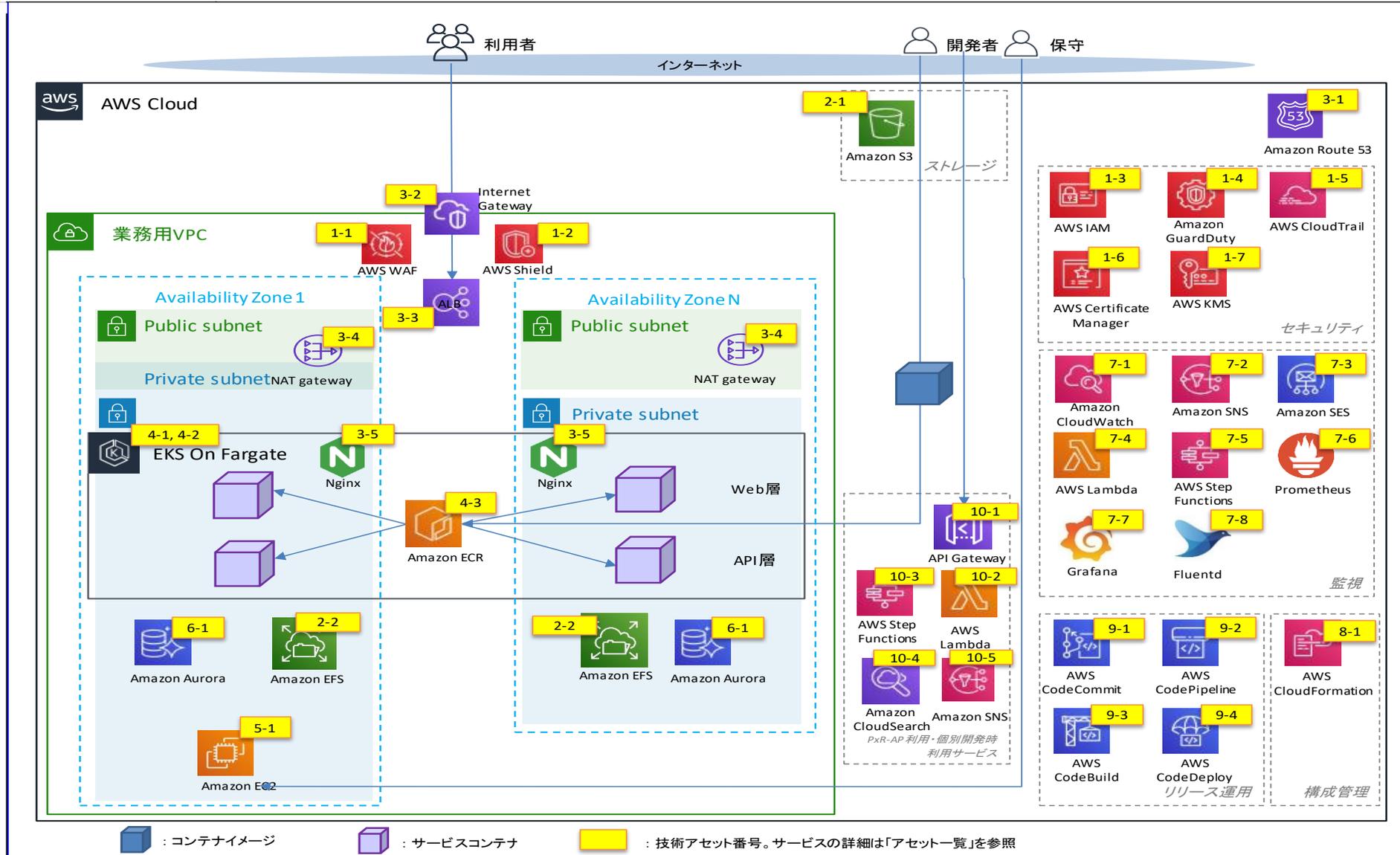
- 上記API機能を構成するソフトウェアは下記の通り

カテゴリ	ソフトウェア名	補足
プログラミング言語	TypeScript	
ORMマッパー	TypeORM	TypeScript向けのORMマッパー
	Express	Node.jsのMVCフレームワーク
プログラム実行環境	node.js	サーバーサイドJavaScript環境
Webサーバー	nginx	フロント配置（リバースプロキシ→node.js）
監視ツール（参考）	Prometheus	リソース監視ソフトウェア
	Grafana	ログ・データ可視化ツール
	Fluentd	ログ収集管理ツール

# 2-2. 物理構成 (例)

## 物理構成図

※AWSでの物理構成例です



## 2-2. 物理構成 (例) 利用アセット一覧

※AWSでの物理構成例です

凡例：AWS「●」はマネージドサービスを示す

大項番	サービス名 (ユースケース)	小項番	サービス/製品名	機能概要	製品区分	
					AWS	他
1	セキュリティ	1	AWS WAF	悪意のあるウェブトラフィックのフィルターサービス PxRサービスへの不正アクセスをブロック、監視を行う	○	
		2	AWS Shield	DDoS保護サービス	●	
		3	AWS Identity & Access Management (IAM)	サービスとリソースへのアクセスを安全に管理するサービス 管理者、開発者、保守担当者がAWSマネジメントコンソールへのアクセスに使用する	○	
		4	Amazon GuardDuty	マネージド型脅威検出サービス	●	
		5	AWS CloudTrail	AWS アカウントのガバナンス、コンプライアンス、運用監査、リスク監査を行うためのサービス	○	
		6	AWS Certificate Manager	SSL/TLS証明書のプロビジョニング、管理、およびデプロイメントサービス	●	
		7	AWS Key Management Service	マネージド型の暗号化キー作成と管理	●	
2	ストレージ	1	Amazon Simple Storage Service (S3)	クラウド内のスケーラブルなストレージ ドキュメントのアップロード、各種ログの保存領域として使用する	○	
		2	Amazon Elastic File System (EFS)	Amazon EFS (EC2 用フルマネージド型ファイルシステム) Fargate上でのファイル格納に使用する	●	
3	ネットワーク	1	Amazon Route 53	スケーラブルなドメインネームシステム (DNS)	○	
		2	Internet Gateway (IGW)	VPC コンポーネントの1つ VPC とインターネットとの間の通信を可能にする	○	
		3	Elastic Load Balancing (ALB)	ロードバランサー 複数のターゲットにわたる着信トラフィックの分配を行う	○	
		4	NAT Gateway	VPC コンポーネントの1つ PrivateSubnetに配置するサービスコンテナからのPublicSubnet 経由でのインターネット通信を解決する	○	
		5	Nginx	Webサーバ、リバースプロキシ NginxをFargateコンテナとし実行する		○
4	コンテナサービス	1	Amazon Elastic Kubernetes Service (EKS)	フルマネージド型の Kubernetes サービス PxRサービスをKubernetesアプリケーションとして実行、スケールする	●	
		2	AWS Fargate	コンテナ向けサーバーレスコンピューティングエンジン 各サービスをpodとして実行する	○	
		3	Amazon Elastic Container Registry (ECR)	コンテナイメージを簡単に保存、管理、デプロイできる完全マネージド型のコンテナレジストリ	●	
5	仮想サーバ	1	Amazon EC2	クラウド内の仮想サーバ 踏み台サーバとして、EKS on Fargateへの保守用、機能設定、バージョンアップ、AWS設定など保守用に使用する	○	
6	データベース	1	Amazon Aurora (Amazon RDS)	高性能マネージドリレーショナルデータベース PxR-APで使用する設定情報、ユーザー情報、蓄積データを管理する	●	

大項番	サービス名 (ユースケース)	小項番	サービス/製品名	機能概要	製品区分	
					AWS	他
7	監視	1	Amazon CloudWatch	リソースとアプリケーションのモニタリングを行うサービス PxRサービスの監視機能として使用する	●	
		2	Amazon Simple Notification Service (SNS)	フルマネージド型pub/subメッセージング、SMS、Eメール、およびモバイルプッシュ通知 監視通知、およびPxRサービスからの通知として使用する	●	
		3	Amazon Simple Email Service (SES)	柔軟でスケーラブルなメールサービス Grafanaからの監視通知として使用する	○	
		4	AWS Lambda	イベント発生時にコードを実行 監視用ログイベントとして使用し、CloudWatchログに出力する	○	
		5	AWS Step Functions	AWS Lambda関数やAWSの複数のサービスを使用してワークフローを構築できるサービス	○	
		6	Prometheus	リソース監視ソフトウェア		○
		7	Grafana	ログ・データ可視化ツール		○
		8	Fluentd	ログ収集管理ツール PxR-AP (pod) からのログ収集に使用する		○
8	構成管理	1	AWS CloudFormation	テンプレートを使ったリソースの作成と管理を行うサービス 環境構築時のテンプレート自動構築実行に使用する		○
9	リリース運用	1	AWS CodeCommit	マネージド型のソース管理サービス プライベートGitリポジトリでのコードの保存に使用する	●	
		2	AWS CodePipeline	継続的デリバリーを使用したソフトウェアのリリース PxRサービスのVerUpを自動化する	●	
		3	AWS CodeBuild	ソースコードをコンパイルし、テストを実行し、デプロイ可能なソフトウェアパッケージを作成する	●	
		4	AWS CodeDeploy	コードデプロイの自動化	●	
10	PxR-AP利用・個別開発時利用サービス	1	API Gateway	クライアントとバックエンドサービス群の間に置かれる API 管理ツール API Gatewayコンソールを使用して、共有トリガー機能開発などでのAPI管理、デプロイ、監視を行う	●	
		2	AWS Lambda	イベント発生時にコードを実行 共有トリガー機能開発などでのAPI (Lambda関数) を実行する		○
		3	AWS Step Functions	AWS Lambda関数やAWSの複数のサービスを使用してワークフローを構築できるサービス 個別開発で作成したAPI (Lambda 関数) を応答性サーバーレスアプリケーションとして統合し、実行するために使用する		○
		4	Amazon CloudSearch	マネージド型検索サービス PxR-APのカタログサービスが内部的に使用する	●	
		5	Amazon Simple Notification Service (SNS)	フルマネージド型pub/subメッセージング、SMS、Eメール、およびモバイルプッシュ通知 2段階認証などSNS通知を行うために使用する	●	

### 3. アクター構成

---

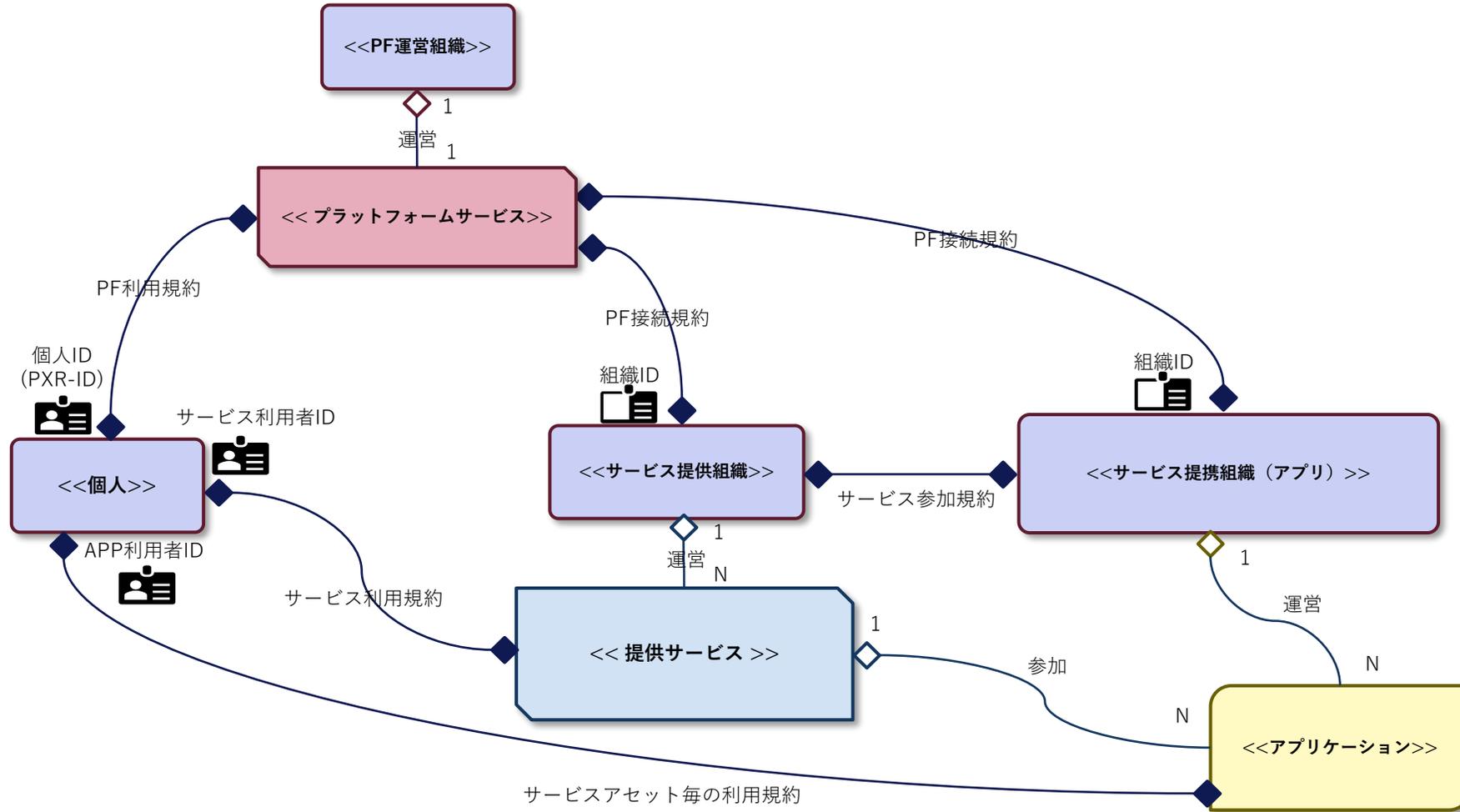
## 3-1. アクター定義

### ◆ PxRエコシステムに登場するアクターは以下の通り

No	アクター	説明	対象Block		
			PXR-Root-Block	Region-Root-Block	APP-Block
1	個人	プラットフォーム (PxR) およびサービスを利用する個人	○		
2	PF運営組織	プラットフォーム (PxR) を運営する組織。PF接続組織を認定、管理する	○		
3	PF接続組織	プラットフォーム (PxR) に接続する組織		○	○
4	サービス提供組織	サービスを提供するPF接続組織		○	
5	サービス提携組織	サービス提供組織と提携し、サービスへ機能を提供するPF接続組織			○
6	サービス提携組織(アプリ)	サービスへアプリケーションにより機能を提供するサービス提携組織			○
7	サービス提携組織(データ保有)	サービスが利用するパーソナルデータを収集し、プラットフォーム (PxR) 上で管理するサービス提携組織			○
8	サービス提携組織(データ利用)	サービスを提供するため、プラットフォーム (PxR) 上で管理されたパーソナルデータを利用するサービス提携組織			○

## 3-2. アクター関係

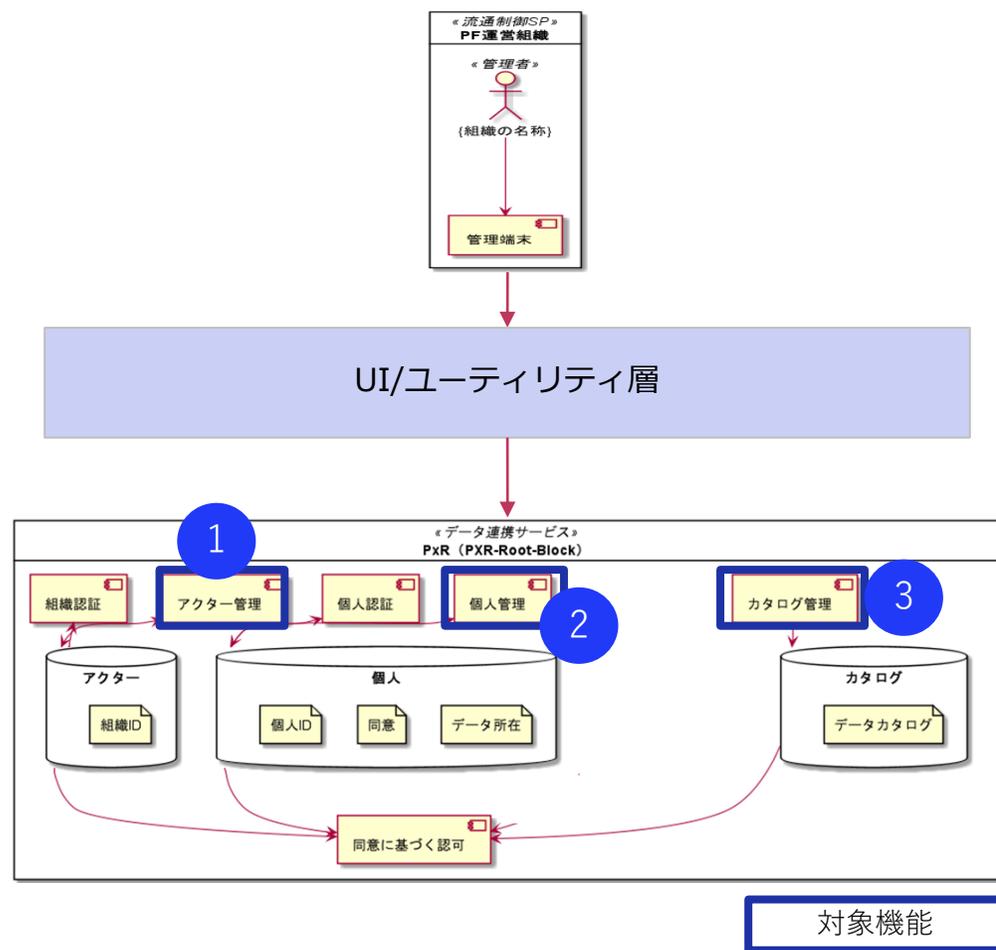
- ◆ 各アクターは下記のトラスト関係を構築する



## 4. 機能

---

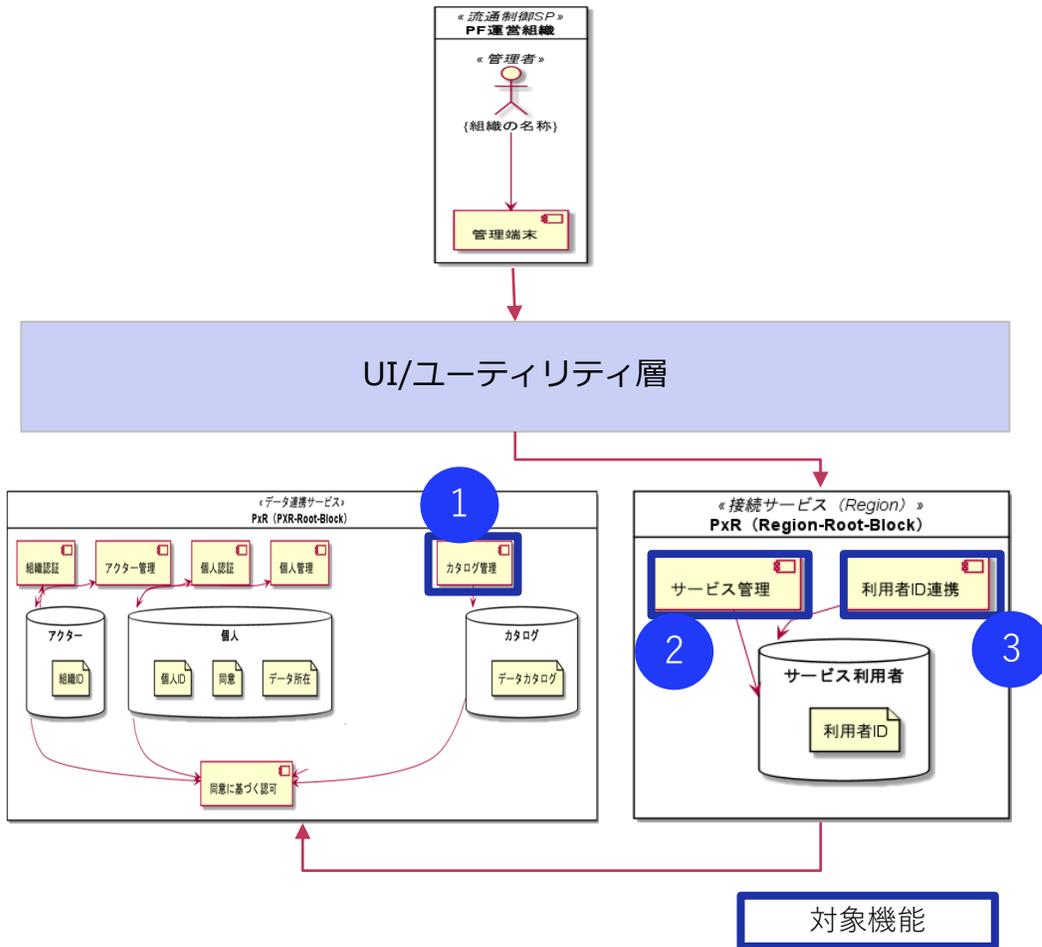
# 4-1. 流通制御SP向け機能



機能		機能概要
1	アクター管理	各アクターの作成からRegionへの参加、離脱等の申請～認定承認等のアクターのライフサイクルに関わるオペレーションを行う為の機能群。 アクター認定時にCL証明書を発行し各組織へ配布することで組織認証を実現する。
2	個人管理	PxR内で個人を一意に特定するID(PXR-ID)を発行し、認証情報を含む個人の基礎情報やデータ蓄積/共有に関わる規約への同意状態、MCB(※1)の開設～削除までのライフサイクルに関わるオペレーションを管理する機能群。
3	カタログ管理	MCD(※2)データカタログ定義に対するメンテナンスやバージョンアップにおける変更申請～承認、各アクターとの同意といったMCDデータカタログに関わるオペレーション全般を行う機能群

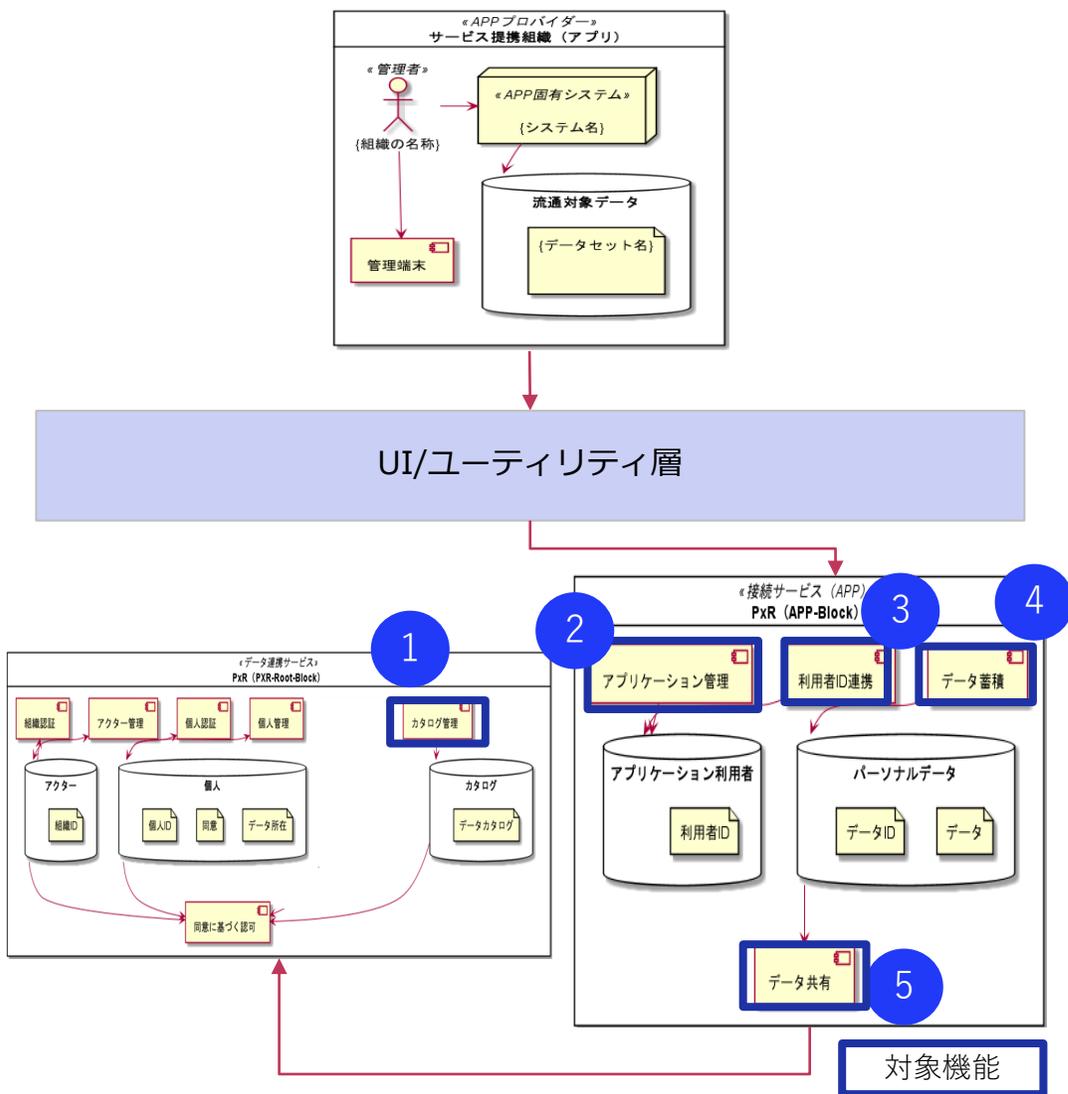
- (※1) My-Condition-Book  
個人が保持する、My-Condition-Data(※2)の集合体
- (※2) My-Condition-Data  
個人の状態を表すデータであり、PxRシステムに蓄積されるパーソナルデータを指す

## 4-2. 領域運営SP向け機能



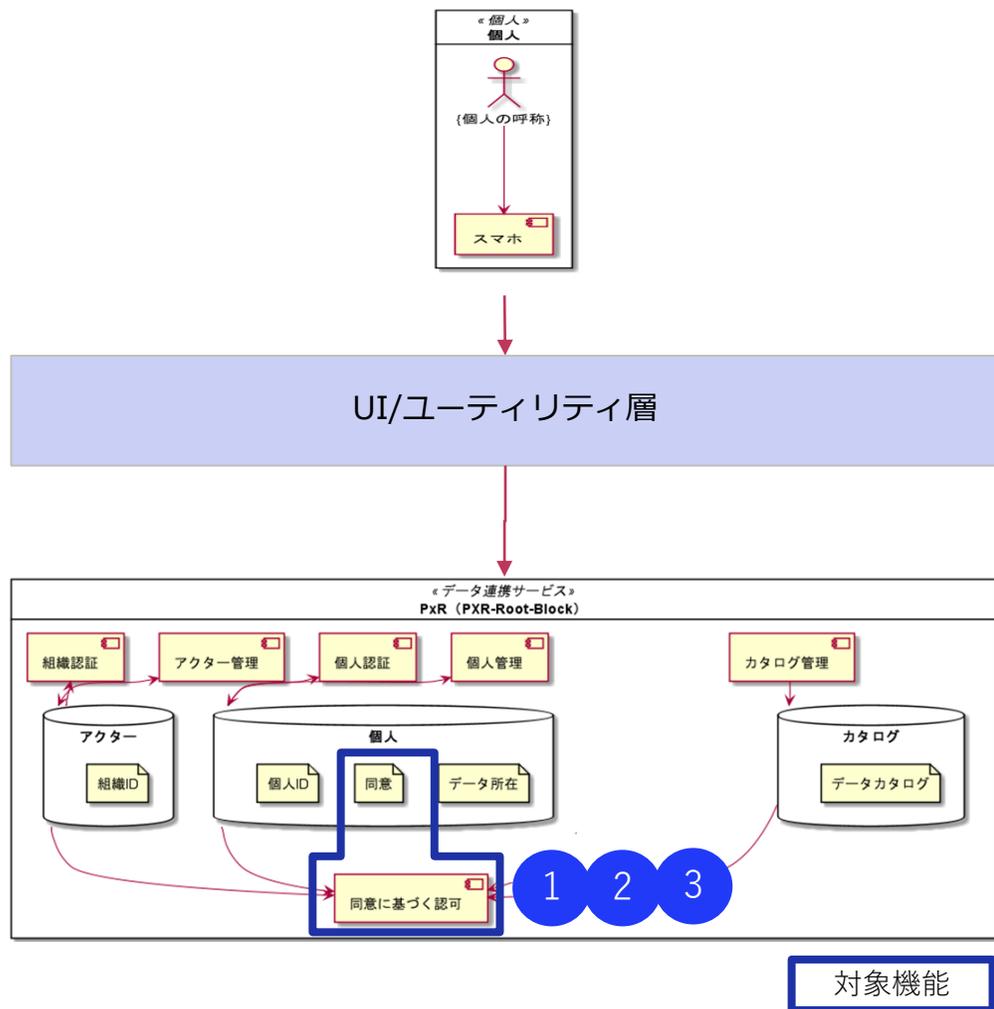
機能		機能概要
1	カタログ管理	流通制御SPから委任された管理対象Regionに関わるサービスの基本情報や設定を管理する (サービスに関わる設定情報カタログの管理)
2	サービス管理	提供するサービス情報や当該サービスに参加するアプリケーション及び利用する個人との関連性や参加申請～承認といったサービス構成要素とその状態を管理する機能群
3	利用者ID連携管理	PxR-IDと個人が該当Region内で利用する利用者IDの紐付けを管理する。

## 4-3. アプリケーションP向け機能



機能		機能概要
1	カタログ管理	当該アプリケーションPが提供するアプリケーションが蓄積するデータカタログ定義のメンテナンスを行う機能群。
2	アプリケーション管理	当該アプリケーションPが提供するアプリケーションにおける基本情報や蓄積/共有に関する定義をメンテナンスする機能群
3	利用者ID連携管理	PxR-IDと個人が該当アプリケーション内で利用する利用者IDの紐付けを管理する。
4	データ蓄積	個人がアプリケーションを利用することで取得されるデータを受信し、パーソナルデータとして蓄積する機能。 同意管理(3-4. 個人向け機能参照)が保持する蓄積同意状態を元に蓄積可否の制御が行われる
5	データ共有	データ蓄積機能にて蓄積されたパーソナルデータを他アプリケーションやサービスへ共有する機能。 同意管理(3-4. 個人向け機能参照)が保持する共有同意状態を元に共有可否の制御が行われる

## 4-4. 個人向け機能



機能		機能概要
1	利用者ID連携 (同意管理)	個人管理で払い出される個人ID (PXR-ID) と当該個人が利用するサービス、アプリケーションにおける利用者IDの関連性を管理する機能。 ※3-2. 領域運営SP向け機能、3-3. アプリケーションP向け機能における利用者ID連携機能と連動し、個人による同意判断に従い連携状態が保持される
2	蓄積同意管理	個人がアプリケーションを利用することで生成されるMCDデータをPXRに蓄積する際の同意状態を管理する機能。個人自らの同意した場合に限り蓄積が行われる。またカタログ管理機能によるデータ定義の変更が発生した場合の再同意プロセスも本機能が提供する。
3	共有同意管理	蓄積されたパーソナルデータを他サービスやアプリケーションに共有する際の同意状態を管理する機能。個人自らが同意した対象データ、共有先に限り共有が行われる。またカタログ管理機能によるデータ定義の変更が発生した場合の再同意プロセスも本機能が提供する。

## 5. 運用

---

## 5-1. 運用フロー一覧

◆ 下記にPxRエコシステムの運用フローを示す

フェーズ	フロー番号	フロー内容
準備	0.1	PF運営組織がプラットフォーム（PxR）の提供を開始する
	0.2	PF接続組織がプラットフォーム（PxR）に接続する
	0.3	サービス提供組織がサービス（Region）を開始する
利用	1	個人がプラットフォーム（PxR）の利用を開始する
	2	個人がサービス（Region）の利用を開始する
	3.1	サービス提携組織がパーソナルデータを利用する
	3.2	個人が自身のサービス（Region）利用状況を把握、管理する
終了	4	個人がサービス（Region）の利用を終了する
	6	サービス提供組織が、サービスの提供を終了する
	7	個人がプラットフォーム（PxR）の利用を終了する
	9	PF運営組織が、プラットフォーム（PxR）の提供を終了する

## 5-2. 運用フロー詳細

◆ 運用フローの詳細は下記に示す別紙「機能要件テンプレート\_別紙」を参照

フロー番号	フロー内容	参照詳細資料
0.1	PF運営組織がプラットフォーム（PxR）の提供を開始する	
0.2	PF接続組織がプラットフォーム（PxR）に接続する	機能要件テンプレート_別紙_0章
0.3	サービス提供組織がサービス（Region）を開始する	
1	個人がプラットフォーム（PxR）の利用を開始する	機能要件テンプレート_別紙_1章
2	個人がサービス（Region）の利用を開始する	機能要件テンプレート_別紙_2章
3.1	サービス提携組織がパーソナルデータを利用する	機能要件テンプレート_別紙_3章
3.2	個人が自身のサービス（Region）利用状況を把握、管理する	
4	個人がサービス（Region）の利用を終了する	機能要件テンプレート_別紙_4章
6	サービス提供組織が、サービスの提供を終了する	機能要件テンプレート_別紙_6章
7	個人がプラットフォーム（PxR）の利用を終了する	機能要件テンプレート_別紙_7章
9	PF運営組織が、プラットフォーム（PxR）の提供を終了する	機能要件テンプレート_別紙_9章

## 6. 技術

---

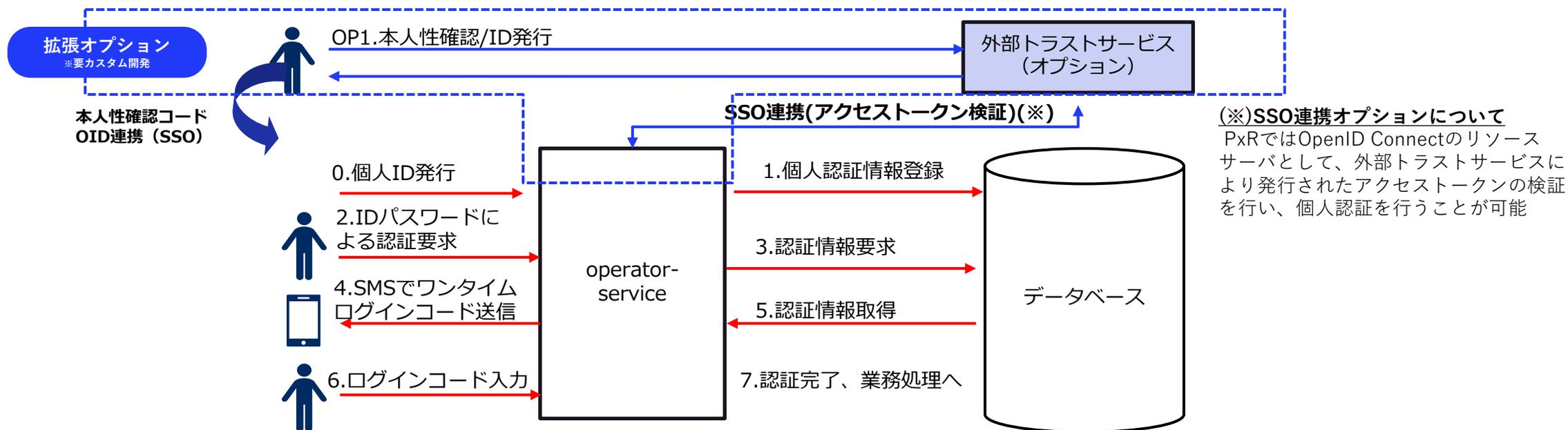
## 6-1. 個人認証

### ◆ 標準仕様

- PXR-Serviceの1つであるoperator-serviceにてオペレーター(運営メンバー、個人)のアカウント管理と認証を行う
- オペレータサービスは、なりすましなど本人以外からの認証要求でないことを確認するためにIDとパスワードに加えSMS認証による2要素認証を用いる

### ◆ 拡張オプション ※標準対象外(カスタム開発が必要となる前提)

- 外部トラストサービスと連携することで、より強固かつシームレスな個人認証を実現することが可能  
(例)本人確認：eKYC、認証サービス：OpenID系SSO連携)

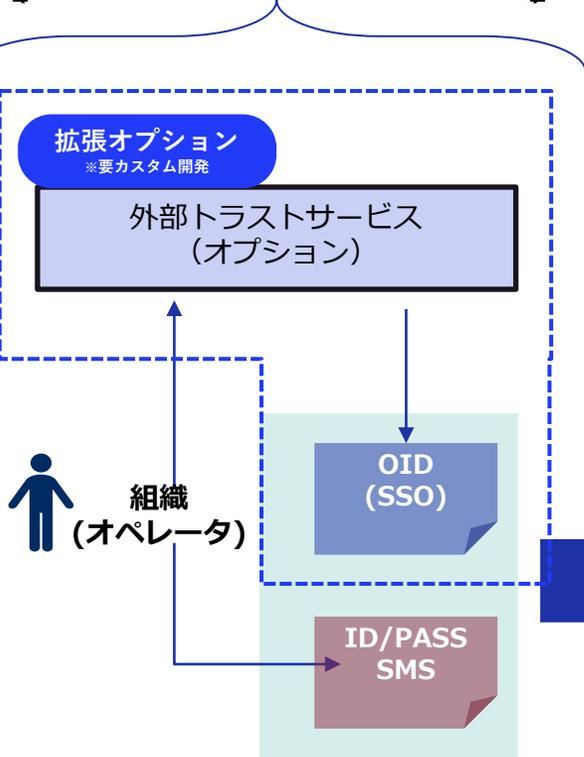


## 6-2. 組織認証

- ◆ オペレータのID/PASS認証 + CL証明書による組織認証を標準とする  
※トラストサービス連携によるOIDC連携 (SSO) もカスタム開発により実現可能
- ◆ 組織間認証はログイン認証をトリガとしたAPIトークンにより認証認可制御を行う

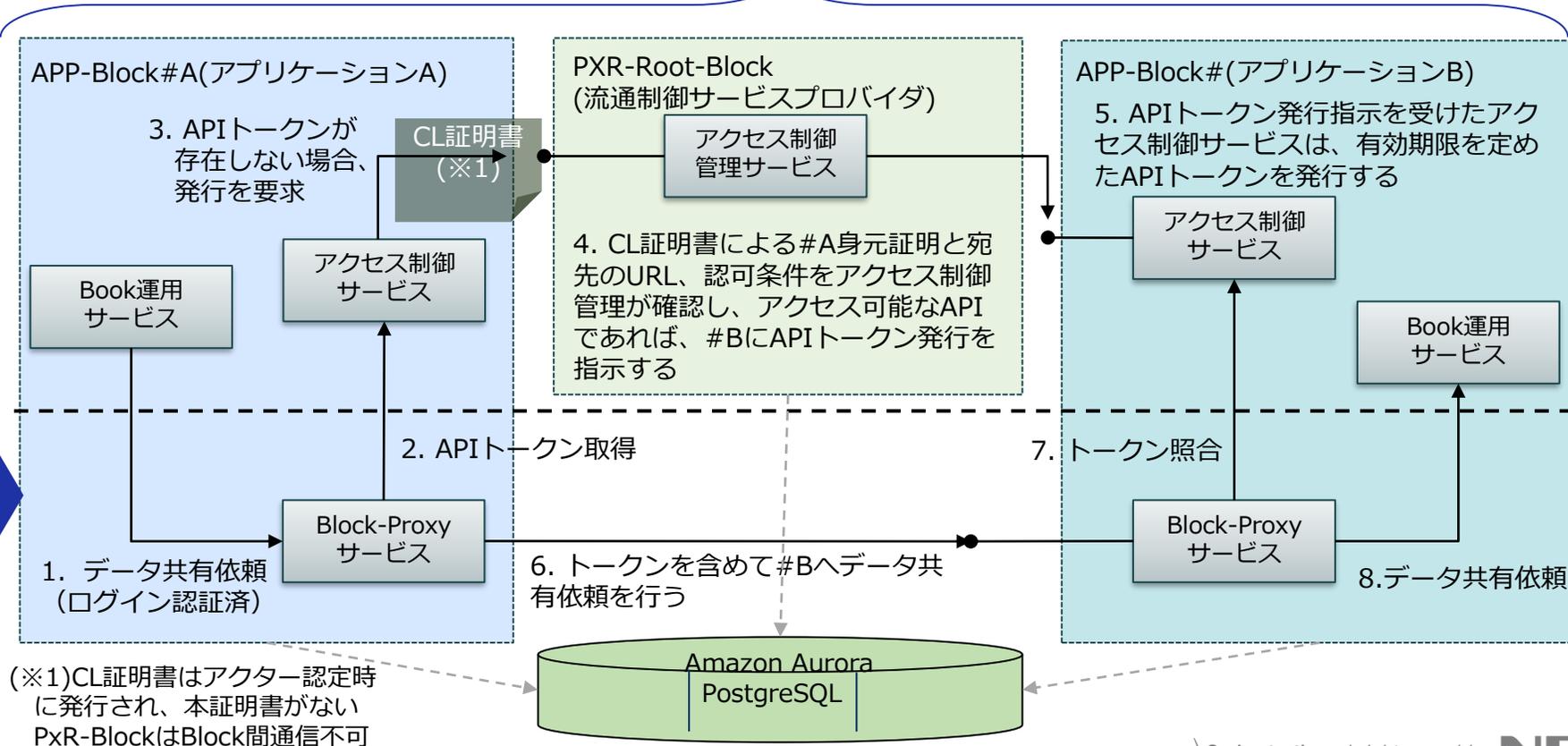
### 組織認証

(PxR-Block利用時の利用者認証方式)



### 組織間認証(データ共有時の例)

(PxR-Block間通信時のBlock認証方式)

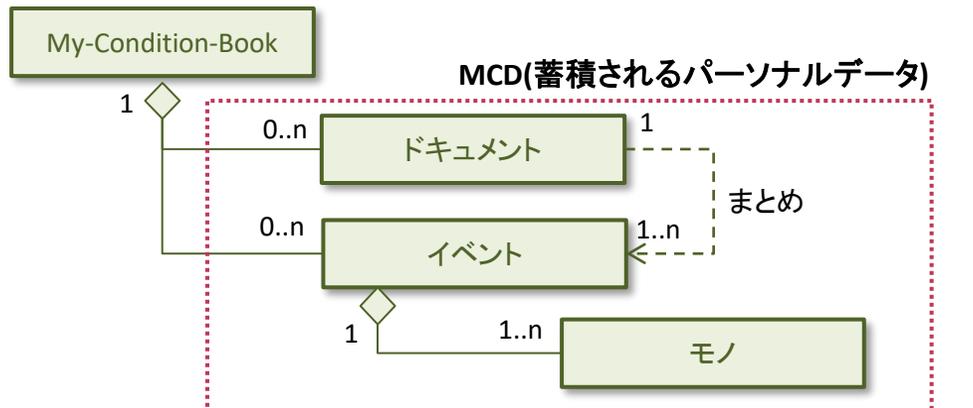


## 6-3. カタログ管理

### ◆ カタログの基本構造と定義方法

- いずれも標準的なテキストベースの構造データ表現フォーマットであるJSON形式で定義される
- カタログテンプレートを作成し、カタログ管理機能が提供する公開APIを利用することでPxRシステム内に投入する

#### データカタログ (MCD)



- My-Condition-Book：個人の基本属性とドキュメントおよびイベントを格納  
例) 個人ID,氏名,性別,生年月日,住所,etc
- ドキュメント：個人が体験するシーンに紐づくイベントのまとめり+アドバイスなどのテキスト  
例) 本日の診察レポート
- イベント：個人が体験するシーンを個人の基本動作に分解したもの  
例) 診断、検体検査、負荷検査
- モノ： イベントの中で記録、測定、評価される項目  
例) 診断病名、検査項目-値

#### MCDメタデータ

##### ■ MCDに対するメタ情報を管理

- ドキュメント、イベント、モノ定義(名前空間、基礎情報等)

名前	所属の名称	導入時の形式	備考
document	ドキュメント	ドキュメント	
event	イベント	イベント	
item	モノ	モノ	

##### ■ 注意事項(他製品とのカタログ連携について)

- CKAN等、他製品とのカタログ連携、共有は標準機能外です。よって、連携を行う場合は、上記公開APIを経由したPxRシステムと他製品間の連携部分をカスタム開発する必要があります。

(カスタム開発例)

PxR側のカタログをAPI経由で抽出し、データ整形した上でCKAN側へ投入する等の連携処理実装

# 6-3. カタログ管理

## ◆ (参考) カタログ定義シート～カタログ定義ファイル

### ①メタデータ定義

カタログ (ext)

カタログ名	カタログ名称	個人向けの説明文	継承元
xxxxxx			
/actor		アクター定義	
/pxr-root [XXXX部門]		流通制御サービスプロバイダー TBD	/model/actor/pxr-root/流通制御サービスプロバイダー
/region-root [XXXX部門]		領域運営サービスプロバイダー TBD	/model/actor/region-root/領域運営サービスプロバイダー
/region [XXXXサービス]		リージョン TBD	/model/actor/region-root/region/Region
/app [アプリケーションAA]		アプリケーションプロバイダー TBD	/model/actor/app/アプリケーションプロバイダー
/actor_9999999			
/application app_sample		アプリケーション TBD	/model/actor/app/application/アプリケーション
/store app_sample蓄積定義		蓄積機能 アプリサンプル蓄積定義	/model/actor/app/store/アプリケーションが蓄積可能なデータ
/share app_sample共有定義		共有機能 アプリサンプル共有定義	/model/actor/app/share/アプリケーションが提供する状態共有機能
/notification app_sample蓄積イベント通知		蓄積イベント通知 アプリサンプル蓄積イベント通知	/model/actor/app/store/notification/蓄積イベント通知

PxR基盤でのベースモデルのうち  
各項目のベースとなるものを記載

XXX定義				
名前空間	ext/actor/app/xxx			
名称	アプリケーションAAA			
説明	[]			
コード	[自動採番]			
バージョン	1			
定義	プロパティ名	型情報	プロパティ値	補足
	application	code[]	(下記アプリケーション定義の情報)	アプリケーション定義の配列(初期値設定)
	breakaway_flg	boolean	false	離脱フラグ(使用しない)
	category	code[]	(設定済み)	本定義が所属するカテゴリ(初期値設定)
	information-site	string		情報サイト(使用しない)
	main-block	code	(設定済み)	アクター参加時に割り当てられたPxR-Block(初期値設定)
	other-block	code[]		他アクターから引き継いだPxR-Blockの配列(使用しない)
	region-alliance	code[]	(設定済み)	参加している領域運営サービスプロバイダーのコード配列(初期値設定)
	statement	string[]		組織ステートメント(使用しない)
	status	at	(設定済み)	設定した時刻(設定済み)
		by	(設定済み)	設定した組織のコード(設定済み)
		status	(設定済み)	設定の履歴(設定済み)

### ②各カタログ詳細定義(項目、型)

### ③カタログ定義ファイル (JSON形式ファイル)

```
{
  "catalogItem": {
    "ns": "catalog/ext/nec-healthcare-consortium/document/actor_1001548",
    "name": "診察レポート",
    "_code": {
      "_value": 1001613,
      "_ver": 2
    },
    "inherit": {
      "_value": 52,
      "_ver": 1
    },
    "description": {
      "title": null,
      "section": [
        {
          "title": "診察レポート",
          "content": [
            {
              "sentence": "診察レポート"
            }
          ]
        }
      ]
    }
  }
},
```

公開API  
(カタログ管理サービス)

オブジェクト  
(JSON)形式

# 6-4. データ管理

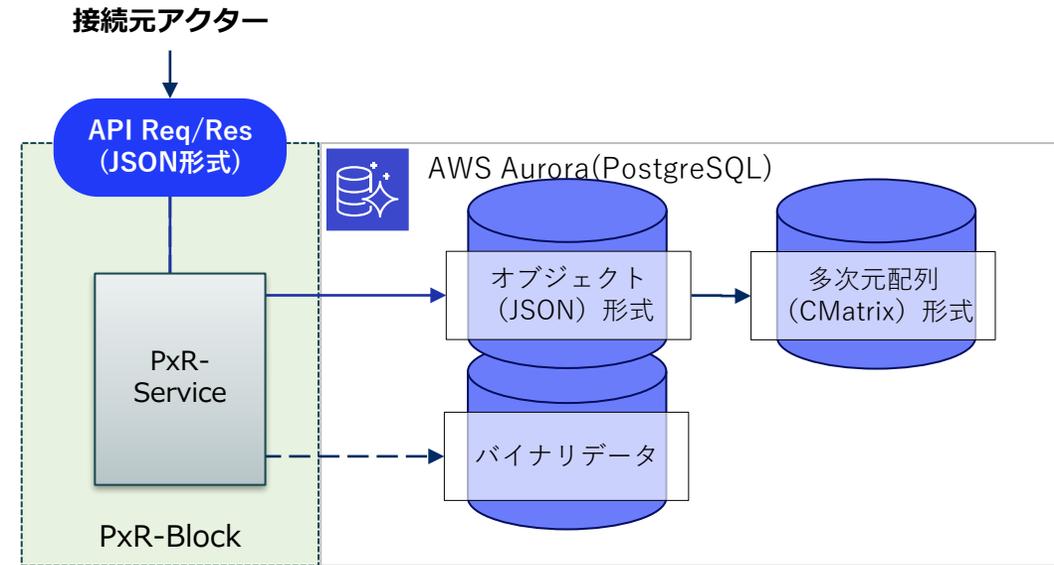
## データベース方式

### ◆ データベース前提条件

PxR-PFとしての要件と根拠	実績のある製品またはサービス
PostgreSQL互換製品であること 機能面 <ul style="list-style-type: none"> <li>・スキーマによる論理分離が可能で、PxR-BlockとPxR-Serviceの二段階分離が可能であること</li> <li>・PxR-AP仕様によりWrapperがPostgreSQL対応のみであるため</li> </ul> 可搬性 <ul style="list-style-type: none"> <li>・広く用いられているOSS</li> </ul>	 AWS Aurora(PostgreSQL)

### ◆ 基本構造・方式

- オブジェクト形式、CMatrix形式、バイナリ形式の3つの形式でデータを格納する
- 全てのDB操作(CRUD)は公開されるAPI経由で行い、データはJSON形式で扱う  
※APIは「1章 (PxR説明資料-別紙)公開APIリスト」を参照
- 登録APIが受け付けたオブジェクトと、オブジェクトから生成されるCMatrix (多次元配列) をDBに格納する。
- バイナリデータがある場合は別DBに格納する。(オブジェクト/Cmatrix DBにはバイナリデータと連携するID情報のみ格納する)

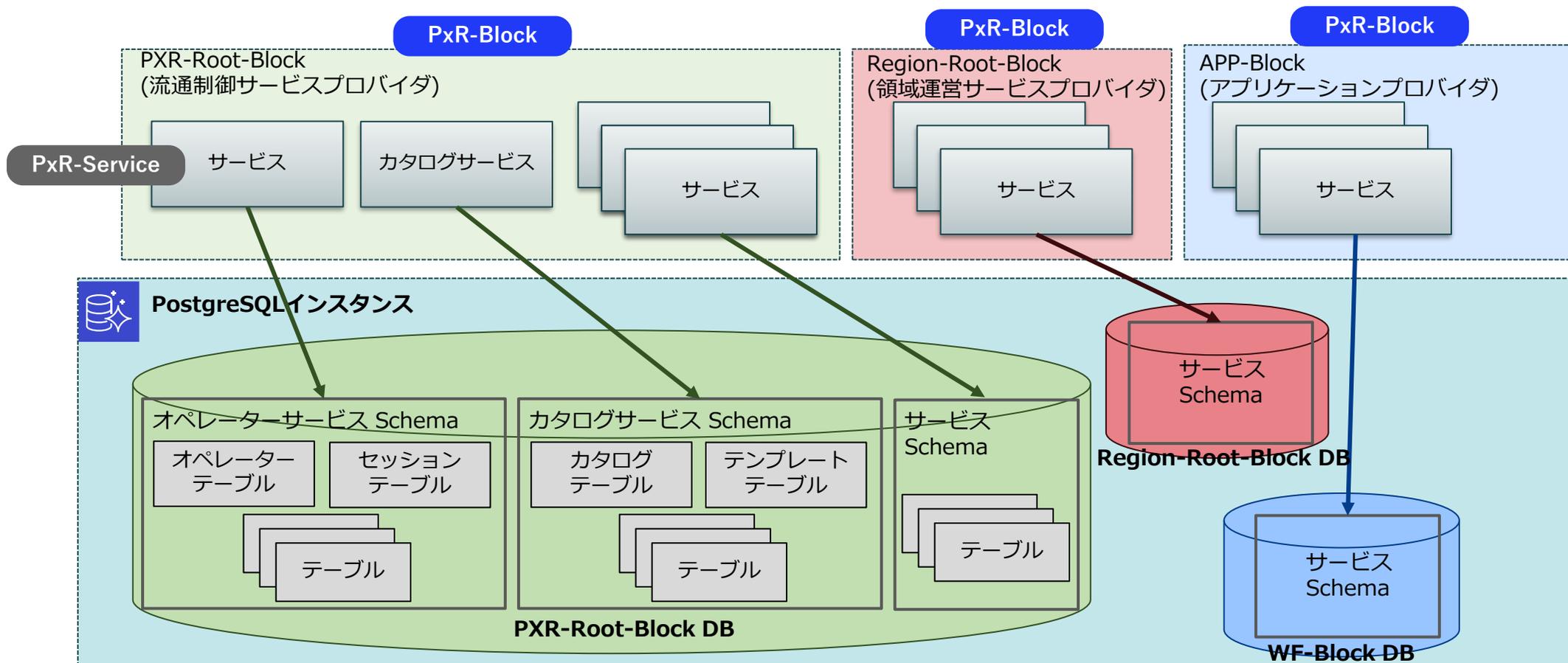


## 6-4. データ管理

### PxR-Block、PxR-ServiceとDBの対応

◆ PxR-Service毎でスキーマを分け、アクセス可能なエンティティを制限することで責任範囲を明確化

- PostgreSQLの1インスタンス内でPxR-Block毎に1つのデータベースを使用する
- PxR-Block毎に使用するデータベースでは、PxR-Service毎にそれぞれスキーマで区切られており、PxR-Service内のAPIはそのスキーマに作成されたテーブルを利用する



\ Orchestrating a brighter world

**NEC**