経済産業省

MOSIP解説書

2021年3月

はじめに

本解説書の目的

本書は、インド国内のデジタルIDプログラム (India Stack) をベースに、eガバメント (電子政府) のあらゆるデジタル公共サービス提供メカニズムの基礎的なビルディング・ブロックを形作るデジタル身分証明(デジタルID)プラットフォームを、第三国展開用にパッケージ化したMOSIP (Modular Open Source Identity Platform) に関して解説したものであり、国内のテクノロジー企業が、第三国におけるオープンソースを活用したデジタルIDプラットフォームの構築およびそれによって利用可能になるであろうと想定される金融、ヘルスケアサービス等を展開していく上での参考となることを目的としている

本解説書の構成と想定読者

第1章 エグゼクティブサマリ:
マネジメント層向けに、MOSIPの設計思想/コンセプト、アーキテクチャおよびその主要な構成要素となる生体認証やプライバシー＆セキュリティ等に関して、その概要を解説

第2章 MOSIP Deep Dive:
技術者向けに、エグゼクティブサマリで示した内容について、実装レベルでの詳細を解説

その他

本解説書は、2020年12月時点でMOSIPウェブサイト (https://www.mosip.io) およびGitHub (https://github.com/mosip/documentation) 上に掲載されている資料をもとに作成したものであり、最新の情報に関しては、当該ウェブサイトを参照

解説書目次

第1章 エグゼクティブサマリ

第2章 MOSIP Deep Dive

第3章 ご参考

第1章
エグゼクティブサマリ

1.1 MOSIP誕生の背景・目的

概要

MOSIPとは？
インド国内のデジタルIDプログラム（India Stack）のノウハウを海外展開するために、コアテクノロジーをオープンソース化したPF

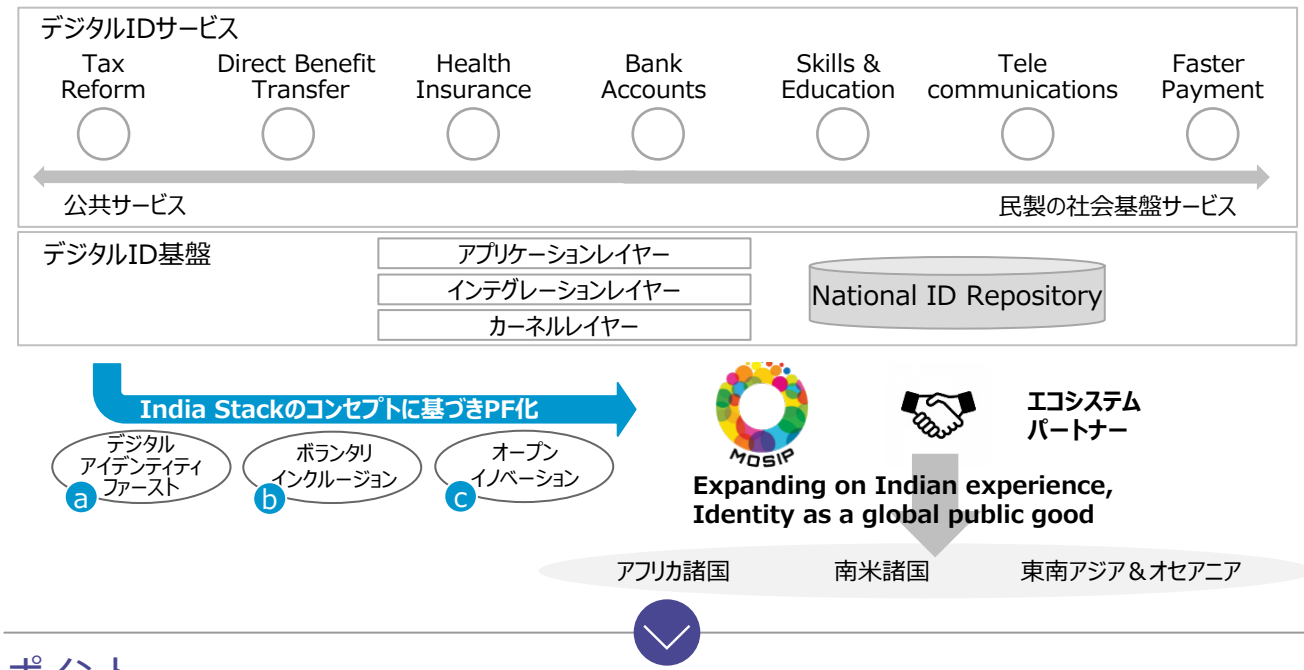
個人のID情報をデジタル化し公共財として流通させる事で、キャッシュレス決済や通信サービスに加えて、助成金移転、租税など公共性の高いサービスを国民全体に浸透させることが狙い

MOSIPの設計思想／コンセプト
MOSIPはIndia Stackの成功要因である以下の設計思想／コンセプトに基づき、設計されている

ⓐ デジタル・アイデンティティ・ファースト
生体認証による簡便なID管理
ⓑ ボランタリ・インクルージョン
国民の目を引くサービスで広範囲な層のID登録を促進
ⓒ オープン・イノベーション
システム開発、生体デバイス、セキュリティ管理、法制度等、各分野に強みを持つ複数プレイヤーによる運営

詳細

India Stack



ポイント

- **世界中の個人のアイデンティティを公共財化することがミッション**
  MOSIPの根底にあるビジョンは、世界各地で未だUndefinedな数十億のIDをデジタル化し、公共財として流通させること

- **成功体験の輸出でデジタルサービス振興**
  MOSIPは政策、技術、法制度、各種サービス事業を組み合わせたインドの成功体験を海外展開するための基盤である

  官民一体となった海外諸国でのデジタルサービス振興のオポチュニティとなる。
  （お金、情報、人財の還流、その結果としての租税、社会保障、信用創造）
  SIや認証デバイス提供等、MOSIP導入に伴う付帯需要も見込む

  各プレイヤーはMOSIPの設計思想／コンセプトを理解し、エコシステムに参画する意義を明確にすることが重要

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

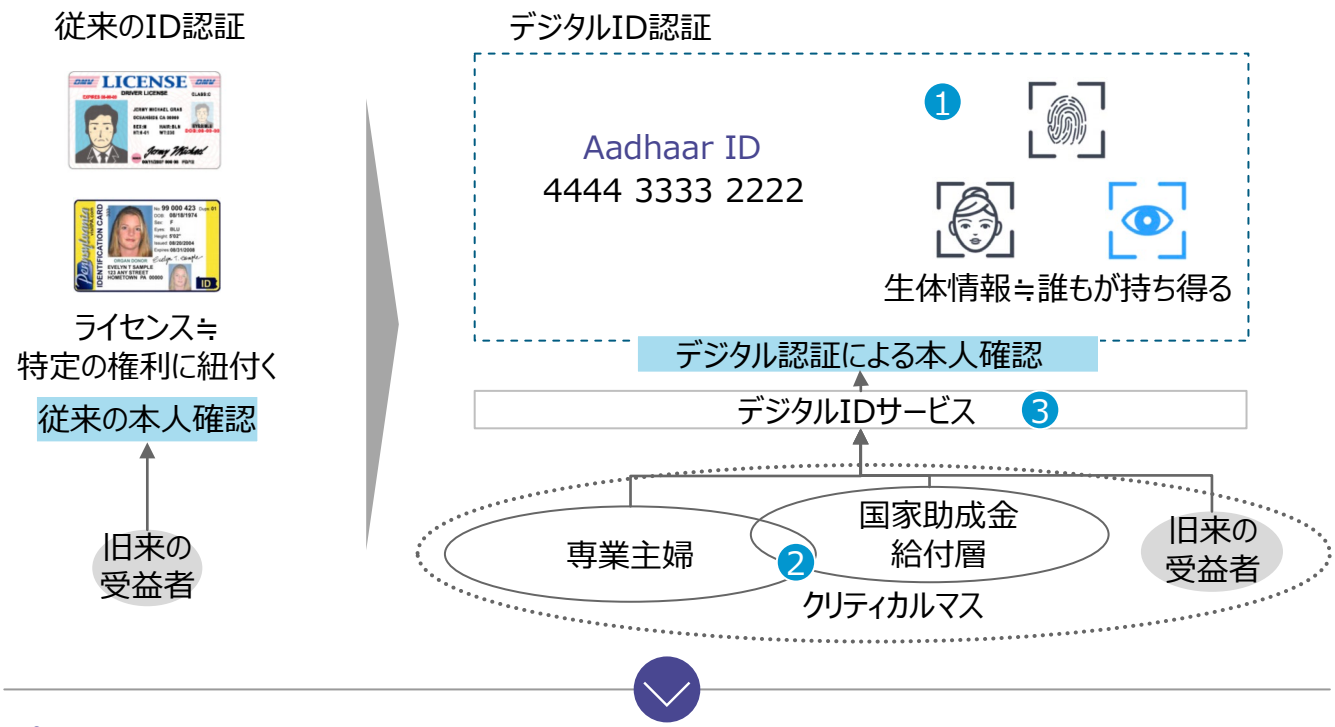
1.2-a デジタル・アイデンティティ・ファースト

概要

インドでは、Aadhaar番号に紐づけられた**個人が誰か？を低コストで簡便に識別する**仕組みを採用

1. 識別に生体情報を用いることで、認証情報登録／本人確認の手間・コストを下げ、より裾野の広い市民層を取り込む。運転免許等、特定の権利・受益に紐づく従来のIDとは独立した管理
2. 国家レベルでの金融/通信サービスの浸透、それによる租税効果を達成するためのクリティカルマスをオーガニックに顕在化させることが狙い
3. 各デジタルIDサービスに共通のKYC機能を提供。各サービス事業者は個別に運転免許証等による本人確認機能を作り込む必要がない

詳細



ポイント

- **インド式のデジタルIDシステムを成立させるKSFは、"生体認証の精度"** であり、この領域で参画するプレイヤーにはデファクトとなるオポチュニティが存在
  - センサー技術や特徴量データ抽出・分析技術等
- **アイデンティティ・ファースト（個人の識別）のモチベーションを明確化**することが重要
  →政府主導でクリティカルマスを顕在化させる意義は？
  →浸透させたい具体的なデジタルサービスは何か？
- **従来のIDとは独立したデジタルIDをデファクト化し、政府が各サービス事業者に提供している**
  - 旧来の受益者層よりもより広い市民層のデジタルIDの取り込みに成功している
  - 政府が発行するIDを利用することで、デジタルIDサービス間のユーザの一意性が高まる

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.2-b ボランタリ･インクルージョン

概要

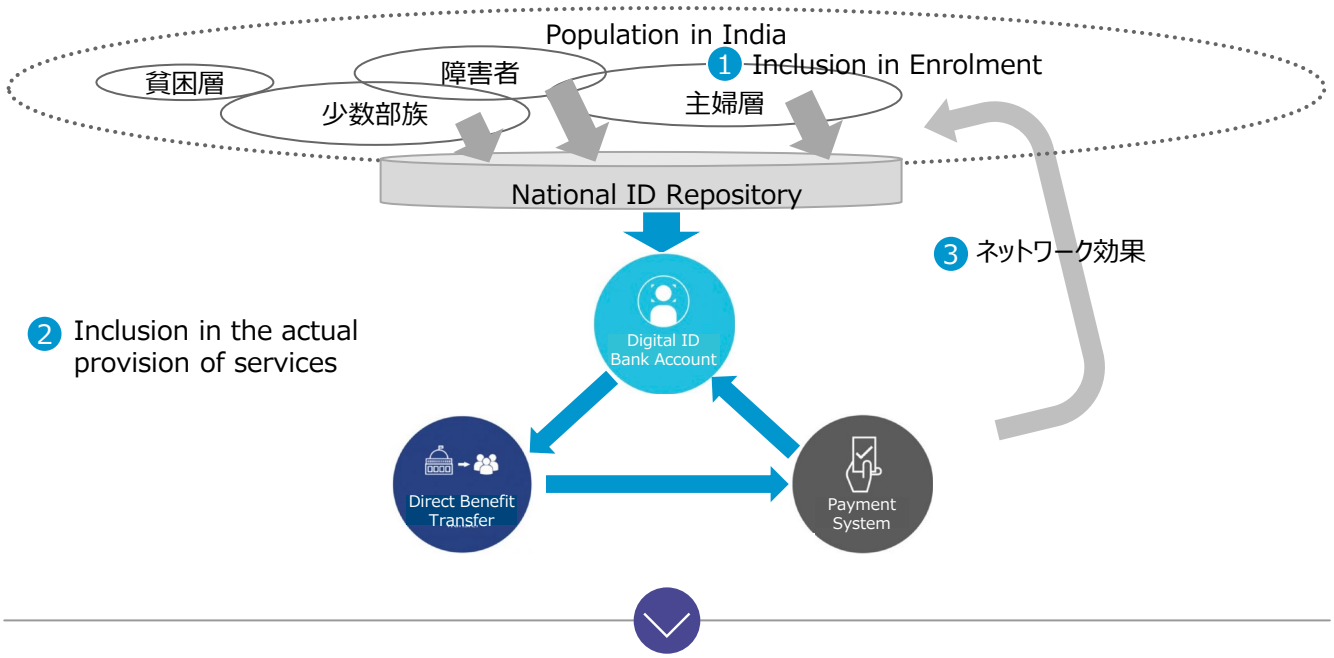
インドでは、民族や階級の垣根を越えて、広く遍く個人IDのInclusion(参画)を促進することをゴールにしている

1 **Inclusion in Enrolment**
クリティカルマスとなり得る層の参画をVoluntaryベースで促進
- 貧困層／少数部族／トランスジェンダー／ホームレス
- 旧来は世帯単位だった主婦層SIMカードの普及が促進
- 障害者や特定職務（炭鉱従事者等）

2 **Inclusion in the actual provision of services**
個人認証をベースとしたサービス利活用を促進し、IDのアクティブ率を上げる

3 **ネットワーク効果**
「皆が利用しているから私も利用したい」という付加価値を生み出し、Inclusionが加速する

詳細



ポイント

- **社会的弱者の取り込みによる新たな市場ターゲットの創出／経済活動の促進**がMOSIP導入のゴールとなっている
- 取組みを成功させるためには、個人認証をベースとしたキラーサービスの導入が必須。**Indian Stackでは補助金供給や決済サービス導入によりお金が還流する仕組みが導入された**
- National IDは**政府主導の取り組みだが、Inclusionには強制力は持たせず市場原理に基づいたオーガニックなID蓄積を指向**している
- **インド全人口に対するカバレッジが高まり、以下の課題が顕在化している**
  1. Social Exclusion（実態としてMandatory Inclusionが必要）
     デジタルIDが発行されていない女性が食糧補助金の利用を拒否される
  2. 個人データの所有権／許諾
     国家によるCensorshipへの恐れや過去の民間企業の不正利用等

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

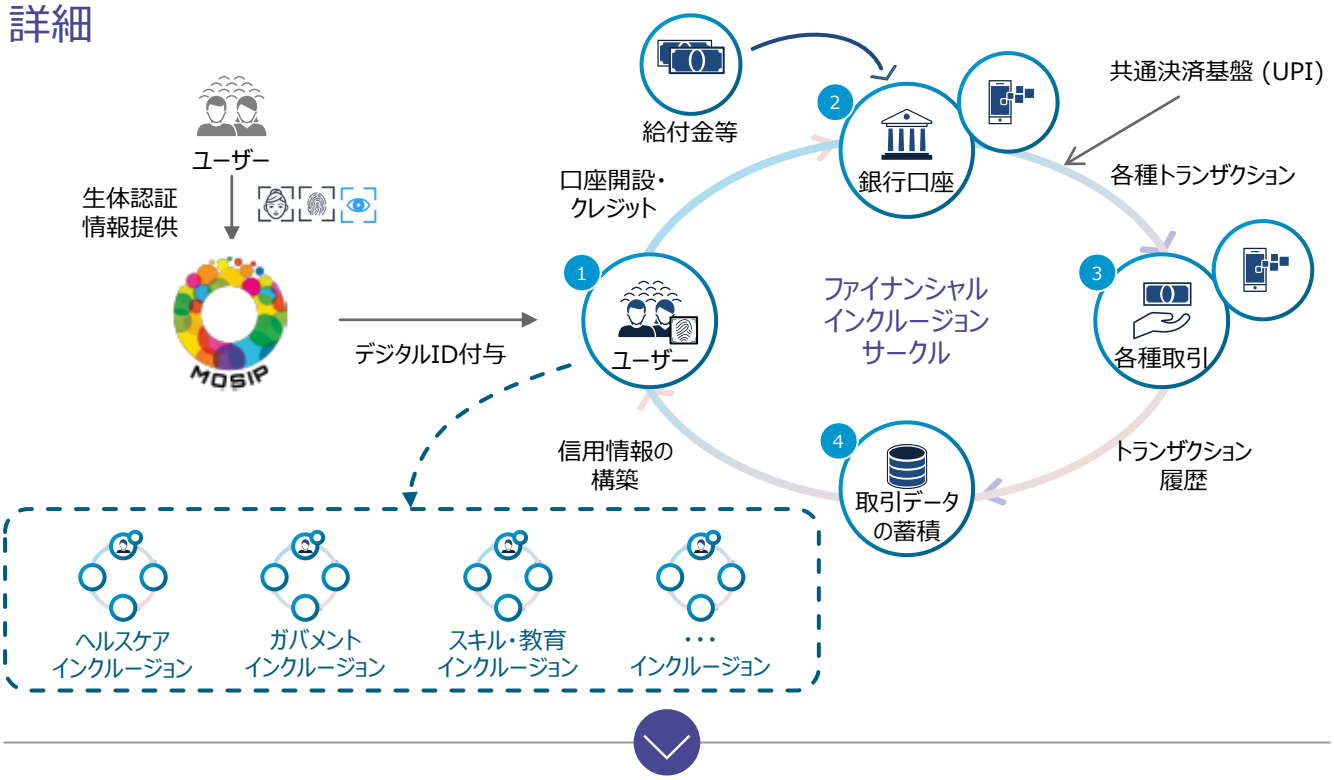
（付録）MOSIPを起点としたインクルージョンサークル

概要

国民にデジタルIDが付与されることで、様々な経済的トランザクションが発生し、国の経済を活性化すると同時に、個人も信用情報を構築し、更なる経済循環のネットワーク効果を生み出す

1. ユーザーがデジタルIDを得ることで、あらゆる場面での本人確認・公的認証が可能に
2. それをもとに、銀行口座の開設やヘルスケア通信サービスの利用が可能に
3. それにより、様々な経済活動に参加することが可能に
4. 経済活動の参加による様々なトランザクション履歴を蓄積することで、個人の信用情報の構築や、ビジネスへの活用が可能に

詳細



ポイント

- **デジタルIDを登録することによって、国民は何ができるようになるのかを明確にする**
  - MOSIPの先にある様々なサービスや利便性を訴求することで、国民にその価値をアピール
- **多種多様なインクルージョンサークルの構築**
  - MOSIPを導入し、デジタルIDを登録することで利用することが可能になるサービス/ソリューションや手続きを、同時並行で構築することで、デジタルIDの価値を早期に最大化
- **インクルージョンから得られるデータの利活用**
  - インクルージョンから得たデータをセキュリティを担保しながら共有することで、新たなサービスやソリューション等の構築・提供に利活用することで、ビジネスの活性化に繋げる

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.2-c オープンイノベーション／エコシステム

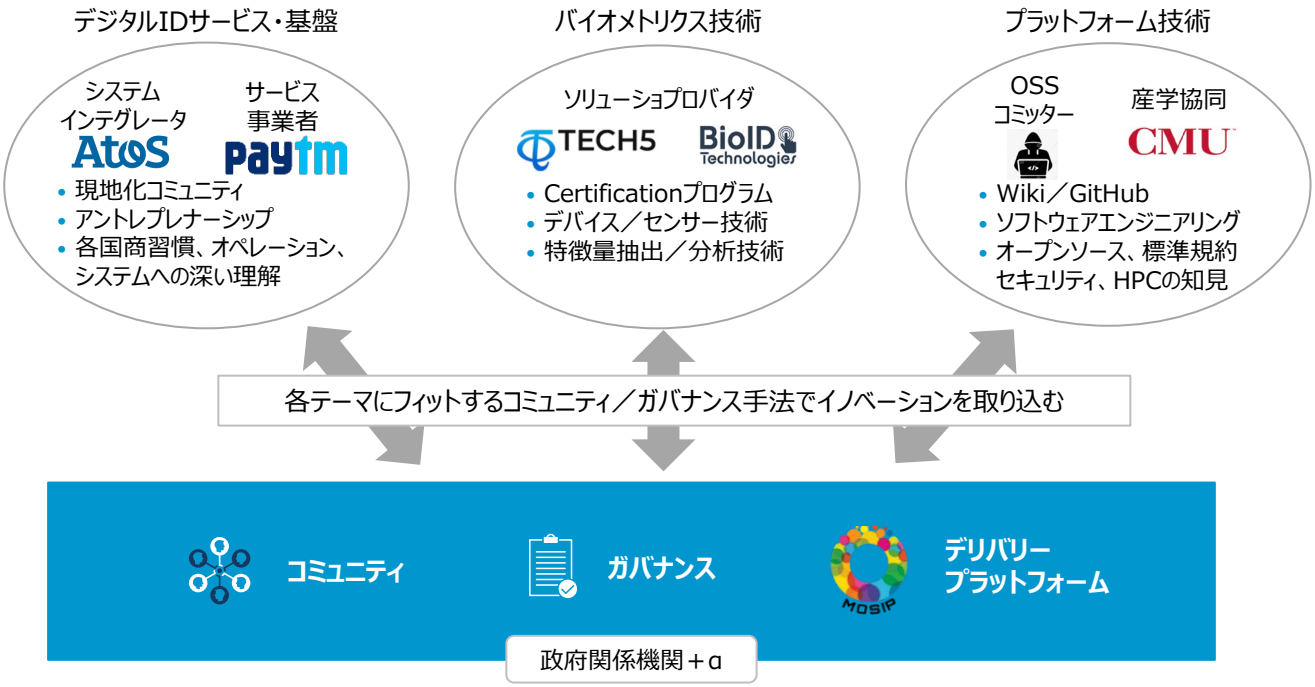
概要

MOSIPはシステム開発、生体認証デバイス、セキュリティ管理、法制度等、各分野に強みを持つ複数プレイヤー協同での運営を前提としている

対象テーマは大きく以下の 3 領域があり、各テーマ毎にコミュニティ／ガバナンス手法を使い分けて、イノーベーションを取り込んでいる

1. デジタルIDサービス・基盤
2. バイオメトリクス技術
3. プラットフォーム技術

詳細



ポイント

- **各国に合わせたエコシステム検討**
  - 各国の事情（母国語、技術者リソース、輸出入規制等）を鑑みたエコシステム検討が重要
  - その上で日本の民間企業／学術機関の立ち位置、参画意義が明確になる
- **コミュニティ／ガバナンス運営者のリーダーシップがKSF**
  - エコシステムを管理・運営するためのガバナンスモデルとリーダーシップが、エコシステム成功のカギとなる

Source: National Open Digital Ecosystem WhitepaperをベースにBCGにて整理

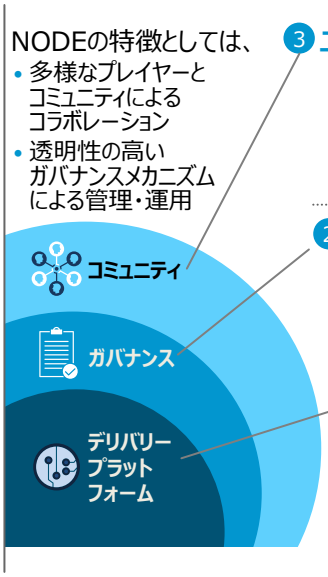
（付録）ナショナル・オープン・デジタル・エコシステム（NODE）

概要

ナショナル・オープン・デジタル・エコシステム（NODE）は3つのレイヤーを基本構造とし、オープンで相互運用可能なデジタルプラットフォームとして、官民の枠を超えたシームレスなサービス提供を実現することを目的としている。

1. サービスとソリューションの提供を容易にする公共のデジタルインフラストラクチャ
2. エコシステムを運営するための規則とルール、そしてそのルールを管理するためのガバナンス体制
3. 共に価値を創造するユーザーとビルダーの共同コミュニティ

詳細



3 ユーザーとサービス提供者のコミュニティ
- エンドユーザーがフィードバックを提供（カスタマー評価、クラウドソーシングによるアイデア）
- パブリックおよびプライベートの開発者/企業が、コアデリバリプラットフォームの上に新しいソリューションを構築

2 エコシステム・ガバナンス
- プラットフォーム・オーナーシップ
- アクセスコントロール
- プライバシー&セキュリティ
- ポリシー&スタンダード
- 人財&ケイパビリティ

1 パブリック・デジタル・インフラストラクチャ
- デジタルIDスタック（例：MOSIP）
- オープンAPI（例：UPI）
- 認証・認可（例：バイオメトリクス）
- データプラットフォーム（例：Hadoop）

ポイント

- **MOSIPの導入だけは、デジタルIDプラットフォームの構築にすぎない**
  - デジタルIDの仕組みの周りに、誰がどんなサービスを構築すべきか等、全体を俯瞰した「あるべき姿」を明確にする必要がある
- **「あるべき姿」の実現に向けて、全体を統括するガバナンスとリーダーシップが必要**
  - エコシステムを管理・運営するためのガバナンスモデルとリーダーシップが、エコシステム成功のカギとなる
    →フィリピンでは、PhilSysがその機能を十分に果たせていないが、モロッコでは、コンサルファームがその役割をカバーすることで、進捗に差が出ている

Source: National Open Digital Ecosystem WhitepaperをベースにBCGにて整理

2.1 アーキテクチャの基本方針

各国導入に向けて、MOSIPは大きく以下の２つをアーキテクチャの基本方針としている

1. デジタルIDサービスを構成する各プレイヤーのノウハウ／IPを取り込み、エコシステム運営をサポートすること
2. India Stackの実運用で培った非機能面のベストプラクティスをMOSIPに実装し、ノウハウを横展開すること

基本方針		内容	
1 エコシステム運営をサポートするアーキテクチャ	MOSIPをプラットフォームとして、強みを持つプレイヤーのノウハウやIPを取り込み、各国エコシステム運営を促進すること	a Modular アーキテクチャ／APIアプローチ	**各国展開時のカスタマイズ／インテグレーションを容易にする仕組み** • MOSIPのコア機能を広く遍く利用してもらうためにAPI仕様を標準化 • ID登録や認証・認可など、業務・サービスのリファレンス実装を用意
		b オープンソース／オープンスタンダード	**オープンソースの積極活用によるソフトウェアイノベーションの取り込み** • HadoopやDockerに代表されるデファクトOSSツールを採用 • MOSIP実装自体もオープンソース化し、技術革新を促進
		c 外部システムインテグレーション	**市場競争力のある外部システムによる機能拡張を容易にする仕組み** • メール通知や郵便配送など、コモディティ機能の外出し • 生体認証に代表されるIPを持つ外部機能との連携I/F
2 デジタルID運用のベストプラクティスを横展開	India Stackのシステム実運用で培った非機能系のベストプラクティスを各国へ横展開すること	d 国民IDレベルの性能拡張性と可用性	**十数億ID規模の登録／認証処理の拡張性、可用性を担保** • 超並列処理を想定したデータ処理基盤を採用 • Public Cloudの積極活用
		e プライバシー／セキュリティ管理	**デジタルIDのプライバシー／セキュリティ管理機能を網羅的に提供** • 秘匿性／一意性を担保する仕組み • ユーザ許諾や監査等の管理機能

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.2 アーキテクチャ全体構造

概要

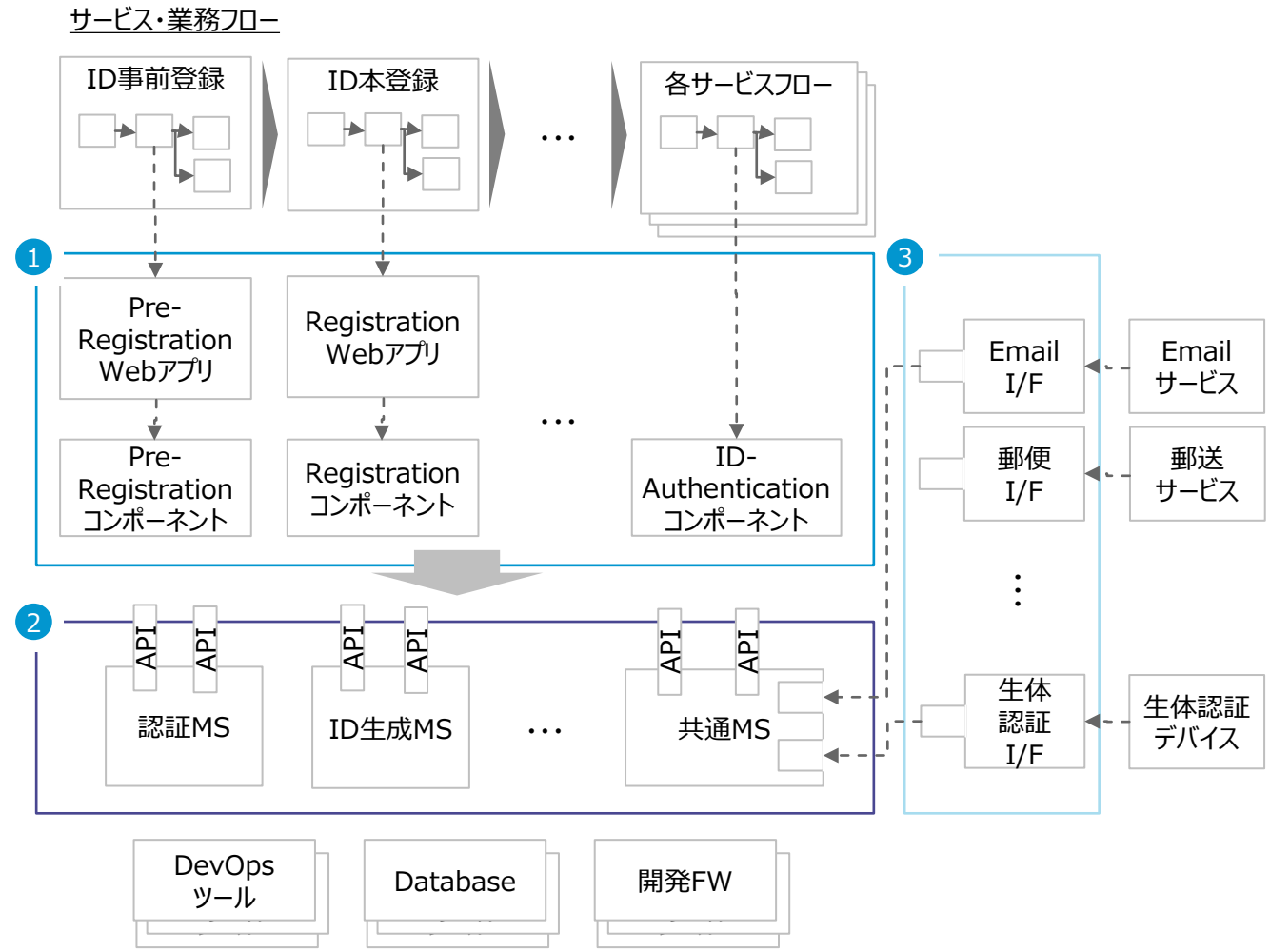
MOSIPのアーキテクチャは大きく３つのレイヤーで構成される。

1. **アプリケーションレイヤー**
   ID登録／認証、サービス利用開始時の本人確認、各種管理機能のアプリケーションの雛形
   各国の手続き業務や認証フローに合わせたカスタマイズが前提

2. **カーネル＆データレイヤー**
   MOSIPのコアとなる機能の集合体。各国アプリケーションに対してAPIを通じて機能提供する
   各国カスタマイズは想定しておらず、機能追加／改修のバージョンはMOSIP運営者が一元管理

3. **インテグレーションレイヤー**
   生体認証デバイス連携、メールサービス連携等、MOSIP外のサービス・デバイスとの連携インターフェースを規定
   市場競争力のある外部サービスをMOSIPに取り込んだ方が合理的な実装部分を外出し

構造イメージ



Source: MOSIPの公開ドキュメントをベースにBCGにて整理

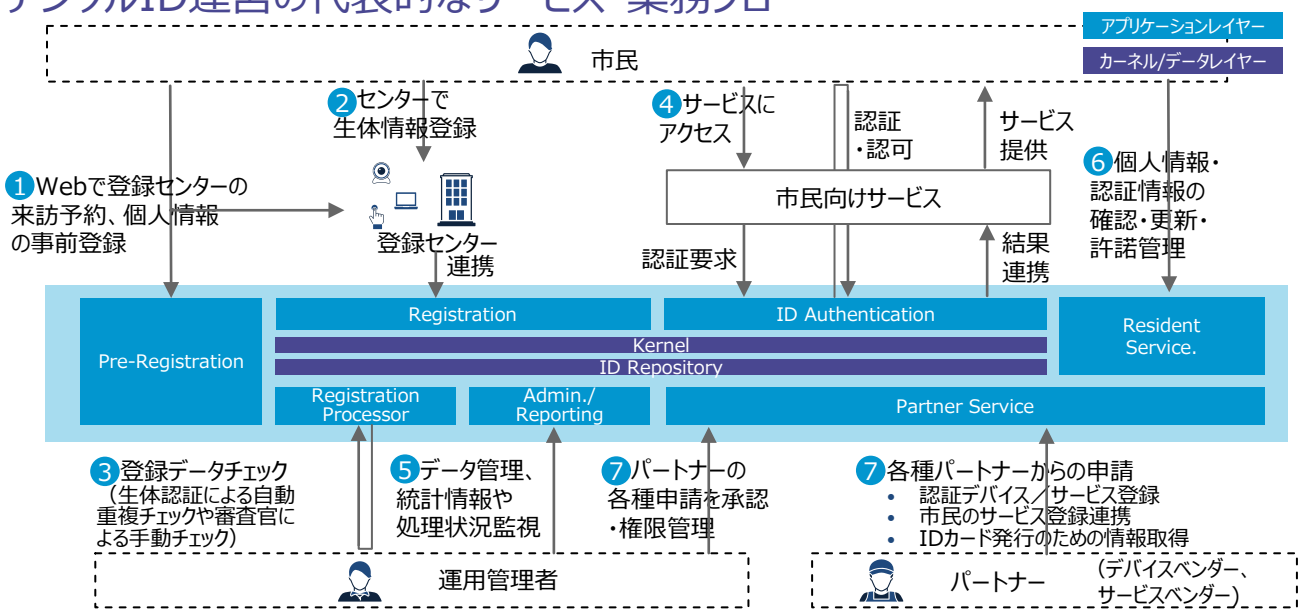
2.3 ①アプリケーション・レイヤー

概要

デジタルID運営に必要な以下のアプリケーションモジュールを提供

❶ Pre-Registration
Webによる事前登録・登録センター来訪予約
❷ Registration
登録センターでの個人情報・生体情報登録
❸ Registration Processor
自動/手動による登録データの形式や重複チェック
❹ ID Authentication
ID関連づいたサービスからの要求による利用者のID認証
❺ Administration/Reporting
システムのマスターデータ管理や統計データの監視・運用
❻ Resident Service
市民セルフサービス
❼ Partner Management
MOSIPのエコシステムのパートナー向けサービス

デジタルID運営の代表的なサービス・業務フロー



ポイント

- **市民向けのフロント機能、管理者やパートナー向けのバックオフィス機能が業務単位でモジュール化されている**
  - 市民向けはID登録や各サービス利用時の認証・認可機能、バックオフィス向けはID登録情報のチェックやPFへの各サービスの組込申請など

- **要件定義時にカスタマイズを前提としたフィット＆ギャップ分析が必要**
  - 導入各国の業務・サービス要件とのフィット＆ギャップ分析による各モジュールカスタマイズが前提
  - MOSIPのアプリケーション実装はAPIの利用方法を規定したリファレンスの意味合いが強く、現地化には相応のコストがかかる（泥臭いオペレーション商習慣、システム的な制約事項への対応など）

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

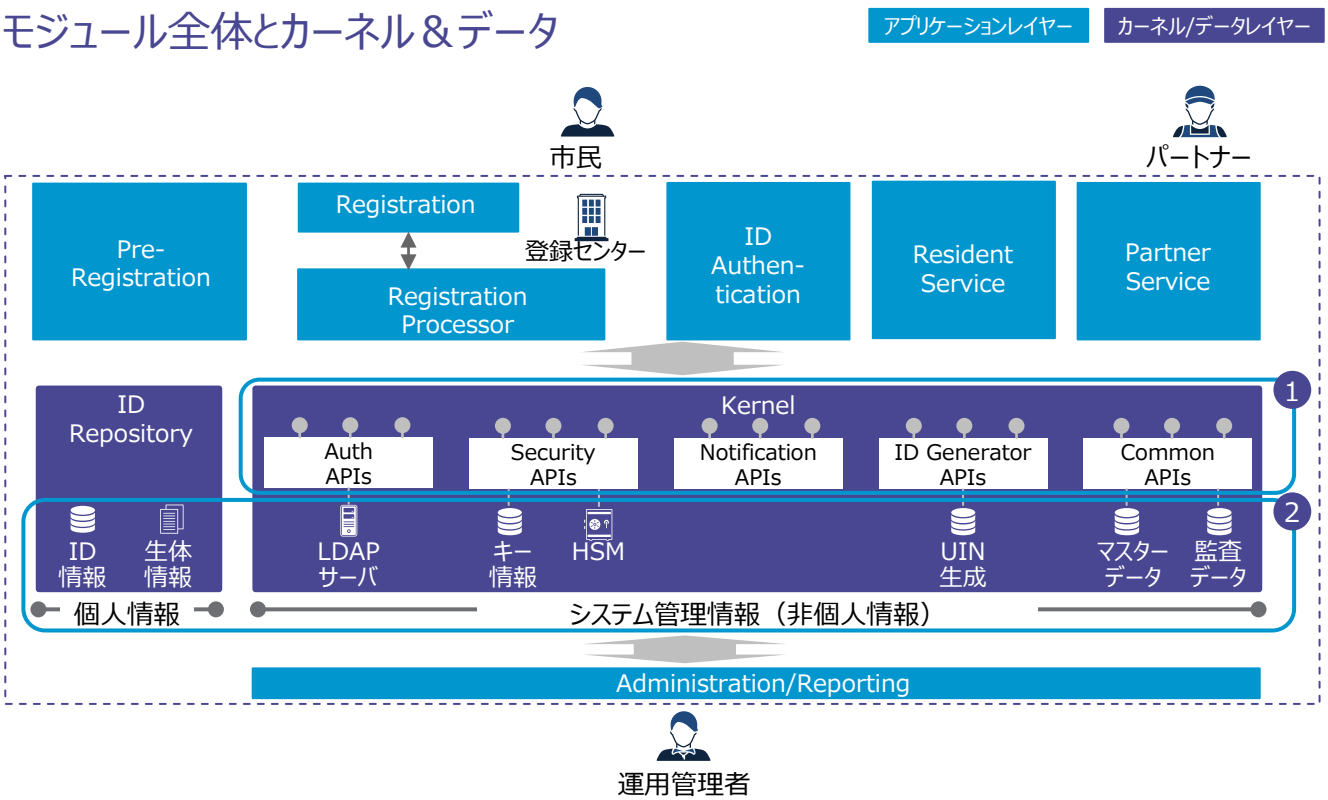
2.3 ②カーネル&データレイヤー

概要

アプリケーションを支える以下のコア機能をカーネルとしてマイクロサービス化・API化し提供

- **Auth APIs**
  認証・認可
  ワンタイムパスワード認証
- **Security APIs**
  キー管理
  暗号管理
  ライセンスキー管理
  署名管理
- **Notification APIs**
  SMS通知
  Eメール通知
- **ID Generator APIs**
  UIN生成
  RID生成
  トークン生成
- **Common APIs**
  監査機能
  データ同期機能
  申請者種別取得機能
  ワンタイムパスワード管理
  登録センターAPIs

モジュール全体とカーネル＆データ



ポイント

❶ API／マイクロサービスアプローチ

- カーネル機能はマイクロサービス化されており、他モジュール及び外部とのインターフェースはAPIで提供され疎結合化
- 各コア機能の実装コードと必要データがマイクロサービスとして独立・分離しているため、開発チーム運営やセキュリティ管理・性能を考慮した機能・データの物理配置の柔軟性を向上

❷ 秘匿性の高いデータアーキテクチャ

- ユーザー情報など機密性の高い個人情報をKernelから分離し、ID Repositoryで集中管理暗号化により秘匿性を向上
- 暗号化に利用する秘密鍵などはHSMなどで堅牢に管理

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.3 ③インテグレーションレイヤー

概要

以下の機能については外部コンポーネントを取り込む前提で連携方針や接続インターフェースが規定されている

- ABIS
- バイオメトリクスSDK
- ウィルススキャン機能
- IAM（権限管理）
- HSM
- 郵便サービス
- eメール/SMSゲートウェイ

詳細

生体認証など高度な専門的な知識を必要とするもの
- ABIS:重複排除のため生体認証タイプに応じたABIS製品を選択
- バイオメトリクスSDK:生体認証タイプに応じて生体認証ベンダーがMOSIPで定義されたBiometrics SDK APIに準拠する形で提供

高度でセキュアなシステム環境を維持するために必要なもの
- ウィルススキャン機能：専門ベンダーが提供しているコモディティ化された製品・サービスを活用
- IAM（権限管理）:利用しているインフラ環境（クラウド）やサービスに合わせた運用アカウント管理を実施
- HSM:暗号鍵など機密性の高いデータを厳密に管理するため、専門ベンダーが提供するHSMなどを活用

ローカライズまたはコモディティ化された機能
- 郵便サービス：導入国で提供されている郵送サービスを活用
- eメール/SMSゲートウェイ:経済性や導入容易性を踏まえ、利用可能な通知サービスを選択

ポイント

- **コモディティ化された機能については経済的な外部モジュールを利用**
  コモディティ化している機能については独自の仕様を作りこまず、経済性・独自仕様による寡占排除を踏まえ、ハードウェア及びソフトウェア・サービスを選択可能

- **高度な専門性を必要とするものは外部パートナーとの連携を想定**
  高度な専門性および先進性を必要とする生体認証に関する機能については、将来的な機能拡張を見越した柔軟な設計とし、専門ベンダーなど外部パートナーとの協業することを前提とする

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.4 ①バイオメトリクス

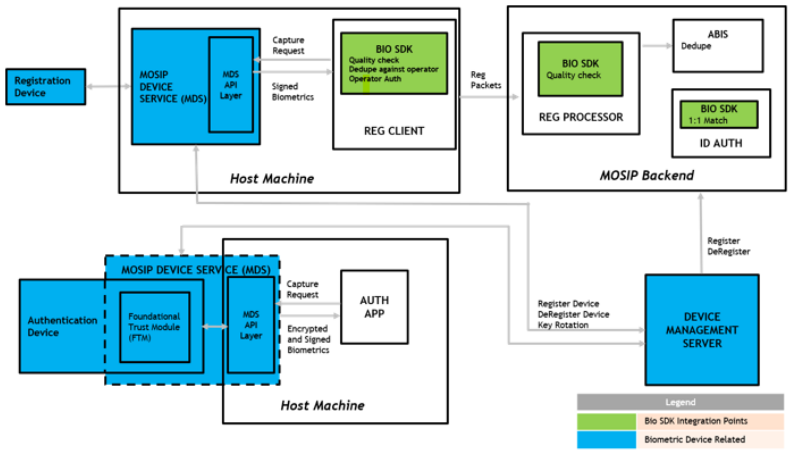
概要

プラットフォームとしてのMOSIPはバイオメトリクスを扱う機能を内蔵しておらず、外部コンポーネントやサブシステムを扱うためのフォーマット、標準、インタフェースのみを定義

バイオメトリクスを扱う機能としては主に以下の2つ
- 品質チェック（重複排除）
- ID Authentication

詳細

バイオメトリクスに関するアーキテクチャ：



MOSIPではバイオメトリクスSDK/APIの仕様を規定
→ フォーマット、標準、インタフェースのみが定義され、**SDK自体は提供されない**

品質チェック（重複排除）

住民に一意のIDを提供することは、IDプラットフォームの主要機能の一つであると位置づけられており、外部コンポーネントであるABIS製品による重複排除を実施。生体認証タイプごとに製品を選択可能
→1対N認証で利用。ABISでの1:1認証は推奨していない

ID Authentication

標準では以下の3タイプの生体認証タイプに対応
- 指紋（1:10認証）/虹彩（1:2認証）/顔（1:1認証）

MOSIPで規定されているMOSIP Device Service(MDS)の仕様の下、生体認証デバイスの製造者、開発者がMOSIP準拠デバイスを開発

ポイント

- **バイオメトリクスSDK自体は生体認証デバイスベンダーが開発する必要がある**
  MOSIPではバイオメトリクスSDK自体は提供されないため、MOSIPで定義されている仕様に則ってに、生体認証デバイスベンダーがSDKを開発する必要ある

- **用途（品質チェック・ID認証）や生体認証タイプによって求められる技術要件は異なり、外部コンポーネントに依存**
  1:1認証と1:N認証では判定速度・判定精度など求められる技術要件が異なり、生体認証タイプ（指紋/虹彩/顔）によって特徴量・照合ロジックも異なるため、それぞれの技術要素に応じた専門ベンダーに依存

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.4 ②プライバシー＆セキュリティ

概要

MOSIPではIDプラットフォームとしてプライバシー＆セキュリティは最優先事項。
アーキテクチャの基本方針に基づき、MOSIPで実装されている機能としては以下

- データベースの暗号化
- 登録データの暗号化
- 鍵管理
- 運用者向け認証

詳細

代表的なMOSIPのセキュリティデザインの特徴
データのアクセスを全てAPI経由とすることで以下のようなセキュリティを実現

- データベースに保存されたデータへの直接アクセスは禁止
- 管理者は実際のデータを表示することなくデータを管理
- IDのスワッピングなどの悪意ある改ざんを防ぐ
- データのプライバシーを確保するためにすべてのアクセスが管理される
- すべてのAPIはレート制限をサポートしており、デジタル署名されている
- すべてのネットワークチャネルは安全ではない前提でチャネル通信を暗号化
- すべての通信データはデジタル署名される

主なセキュリティ機能

- データベースの暗号化
  - MOSIPの暗号化にはデータベースに組み込まれたメカニズムを使用せず、DBに保存されるすべての機密データはDBの外部のアプリケーションレイヤーで暗号化/復号化
- 登録データの暗号化
  - 登録クライアントを使用してすべての個人情報と生体情報を収集し、暗号化される。暗号化に使う鍵はHSMで生成される
- 鍵管理
  - MOSIPではAES鍵とRSA鍵が使用される
- 運用者向け認証
  - 事前登録Webアプリ、管理Webアプリ、住民サービスポータル、登録クライアントでの登録スタッフ・運用者向けの認証を想定

ポイント

- **データへのアクセスはすべてAPI経由に制限**
  全てのデータアクセスをAPI経由とすることでモジュラリティとセキュリティを両立

- **セキュリティ機能を継続的に改善**
  オープンソースプロジェクトとして、MOSIPではコラボレーションとコミュニティの貢献を通じてセキュリティ機能を継続的に改善し、新規開発機能を導入し続けることを狙う
  また、さまざまなセキュリティツールを使用して、継続的にセキュリティを評価をし、脆弱性を発見・対処

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.4 ③基盤インフラストラクチャ

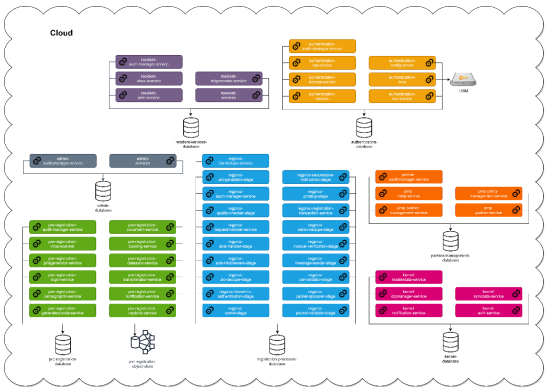
概要

MOSIPでは可用性、信頼性、拡張性、セキュリティ、回復力、管理性を考慮し、セルベースアーキテクチャを採用
処理性能とハードウェアのプロビジョニングを線形にスケーリングさせるため、セルベースアーキテクチャを推奨

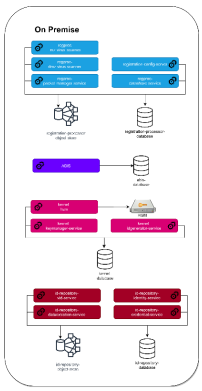
また、クラウドとオンプレミスの両方のメリットを活用するハイブリッドなインフラ構成を想定

詳細

セルベースアーキテクチャ：
MOSIPでは、ハードウェアとソフトウェアをセルに固定する (閉じ込める) セルアーキテクチャが推奨され、
入出力の処理能力のベンチマークはセル単位で行われる。
セルを複製し、ロードバランサーでトラフィックを分散させることで、本番環境の処理能力をスケール可能とする



クラウドでは迅速なデプロイが可能で管理も簡単なため、
スケーラビリティが求められるセルを配置



オンプレミスではローカル環境で
厳格に保持・
管理する必要がある
データ・ポリシーを扱う

ポイント

- **マイクロサービスアーキテクチャを一歩具体化した形で提供**
  マイクロサービス、データベース、ストレージクラスターなどの複数のコンポーネントが複雑に連携し合って実行されている
  複雑なシステムは構築するのに手間がかかるため、MOSIPではセルアーキテクチャとしてアーキテクチャ構想を具体化した形で提供

- **特定のインフラ環境に依存せず、導入可能**
  導入国の要件（登録者数やIT予算）やインフラ環境に応じて導入国が自ら構築するインフラ環境を選択可能。
  ただし、MOSIPとしてはアプリケーションレイヤーに関してはスケーラビリティの観点からクラウドでの構築を推奨

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

2.5 各技術分野の標準化について

レイヤー	アーキテクチャ方針	標準化対象	標準化の状況と各国展開時のポイント
アプリケーションレイヤー	a	業務・サービスフロー／API連携フロー	**デジタルID運営に必要なアプリケーションが標準提供されるが、各国導入にはカスタマイズが必要** • ID登録、認証・認可、各種バックオフィス業務向けのアプリケーション機能がモジュール提供されている。 • あくまでもMOSIPのAPIの利用方法を規定したリファレンス実装の位置付けであり、各国導入においては一定規模のカスタマイズが想定される。
カーネル&データレイヤー	Modularアーキテクチャ／APIアプローチ	APIインターフェース、APIプロトコル	**MOSIP上でのサービス間連携は標準化されているが、他PFとの連携を想定した標準規格には未準拠** • ID登録、認証・認可処理に必要な機能及び関連する管理機能をAPI化し、エコシステムに開示。オープンなプロトコル（REST）を採用し、システム間接続・データ通信方法を標準化 • あくまでもMOSIP上に載るサービス間の認証連携のみを想定した独自実装であり、Open ID ConnectやOAuth2.0といった異なるPF間の認証・認可に関する標準規格には未準拠 • 今後、異なるPF上のサービスとの認証連携を想定し、標準規格に準拠する可能性はある
	b	利用ソフトウェア(開発FW、DBMS等)／MOSIPコア機能	**カーネル&データレイヤーの実装は全てオープンソース化されており、商用ツールの導入余地はない** • 開発フレームワークやDevOps／テスティングツール、データベース等、カーネル実装に必要となるツールやミドルウェアにデファクトスタンダードなオープンソースソフトウェアを積極活用している • カーネルレベルのコア機能に商用のプロプライエタリな実装が入り込む余地はなく、デファクトスタンダードをベースに産学共同やOSSコミッターの専門家中心にセキュリティ技術やHPCの実装改善が継続される
インテグレーションレイヤー	c	通知、郵送機能インターフェース	**コモディティ機能はMOSIP内に実装せず、外部サービス連携のインターフェースを標準化／開示** • 郵便配送や電子メール・SMSによるコミュニケーション等、現地サービス活用が想定される部分はあえて実装せず、MOSIPフレームワークの外部実装箇所としてI/Fのみを標準化・開示している • 必要機能ではあるがコモディティであり、付加価値を追求するコア機能から除外する部分
	外部システムインテグレーション	生体認証／重複排除インターフェース、デバイススペック	**IPインテンシブでオープン化できない要素技術についてはパートナーシップを前提とした連携方針を標準化** • 生体認証や一部のセキュリティ管理機能については、IPを持つ商用ベンダーツールをMOSIPのPFに取り込むためのI/Fやデバイスのハードウェアスペックが標準化されている • カーネルレベルのコア機能としてオープン化できない技術分野は商用ツールベンダーのビジネス機会となる

第2章
MOSIP Deep Dive

1.
設計思想/コンセプト

1.1.1 エンゲージメントの原則

原則

インクルージョン： 普遍的な対象範囲と アクセシビリティ	1. 個人の出生から死亡まで普遍的にシステムの対象とする。差別はしない 2. アクセスおよび利用の障壁や情報およびテクノロジーの利用格差を取り除く
デザイン: 堅牢性、安全性、 応答性、持続可能性	3. 堅牢性の高い (一意で安全、正確な) IDシステムを構築する 4. さまざまなユーザーのニーズに対応した相互運用可能なプラットフォームを作り上げる 5. オープンスタンダードを利用してベンダーおよびテクノロジーに縛られないようにする 6. システム設計を通じてユーザーのプライバシーとユーザーによる管理を保護する 7. アクセス性を犠牲にせずに、財務的および運用上の持続可能性を保持できるよう計画する
ガバナンス: プライバシーとユーザーの 権利を保護して信頼を 構築	8. 包括的な法律および規則の枠組みを通じてデータのプライバシー、セキュリティ、ユーザーの権利を保護する 9. 明確な制度的義務および説明責任を確立する 10. 苦情について独立した監視と裁定を行うことで、法に基づく信頼の枠組みを強化する

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

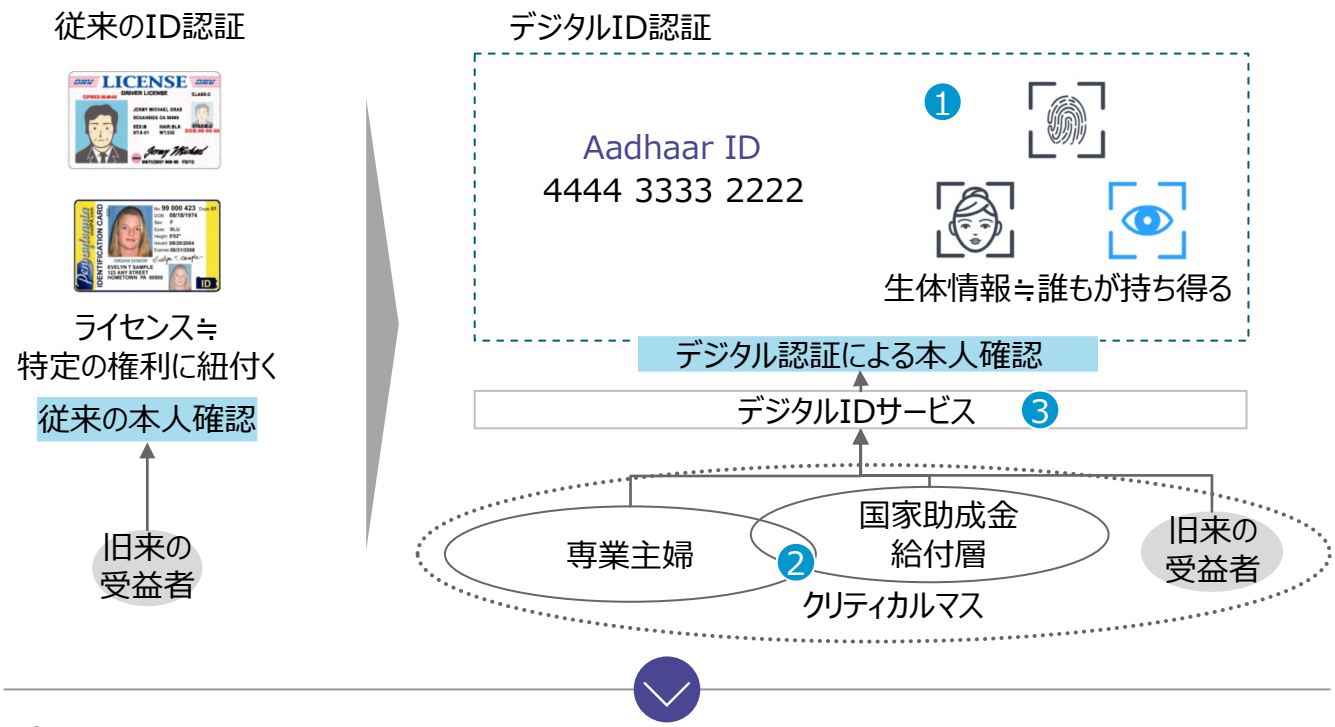
(再掲) 1.1.2 デジタル・アイデンティティ・ファースト

概要

インドでは、Aadhaar番号に紐づけられた**個人が誰か？を低コストで簡便に識別する**仕組みを採用

1. 識別に生体情報を用いることで、認証情報登録／本人確認の手間・コストを下げ、より裾野の広い市民層を取り込む。運転免許等、特定の権利・受益に紐づく従来のIDとは独立した管理
2. 国家レベルでの金融/通信サービスの浸透、それによる租税効果を達成するためのクリティカルマスをオーガニックに顕在化させることが狙い
3. 各デジタルIDサービスに共通のKYC機能を提供。各サービス事業者は個別に運転免許証等による本人確認機能を作り込む必要がない

詳細



ポイント

- **インド式のデジタルIDシステムを成立させるKSFは、"生体認証の精度"** であり、この領域で参画するプレイヤーにはデファクトとなるオポチュニティが存在
  - センサー技術や特徴量データ抽出・分析技術等
- **アイデンティティ・ファースト（個人の識別）のモチベーションを明確化**することが重要
  →政府主導でクリティカルマスを顕在化させる意義は？
  →浸透させたい具体的なデジタルサービスは何か？
- **従来のIDとは独立したデジタルIDをデファクト化し、政府が各サービス事業者に提供している**
  - 旧来の受益者層よりもより広い市民層のデジタルIDの取り込みに成功している
  - 政府が発行するIDを利用することで、デジタルIDサービス間のユーザの一意性が高まる

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

(再掲) 1.1.3 ボランタリ・インクルージョン

概要

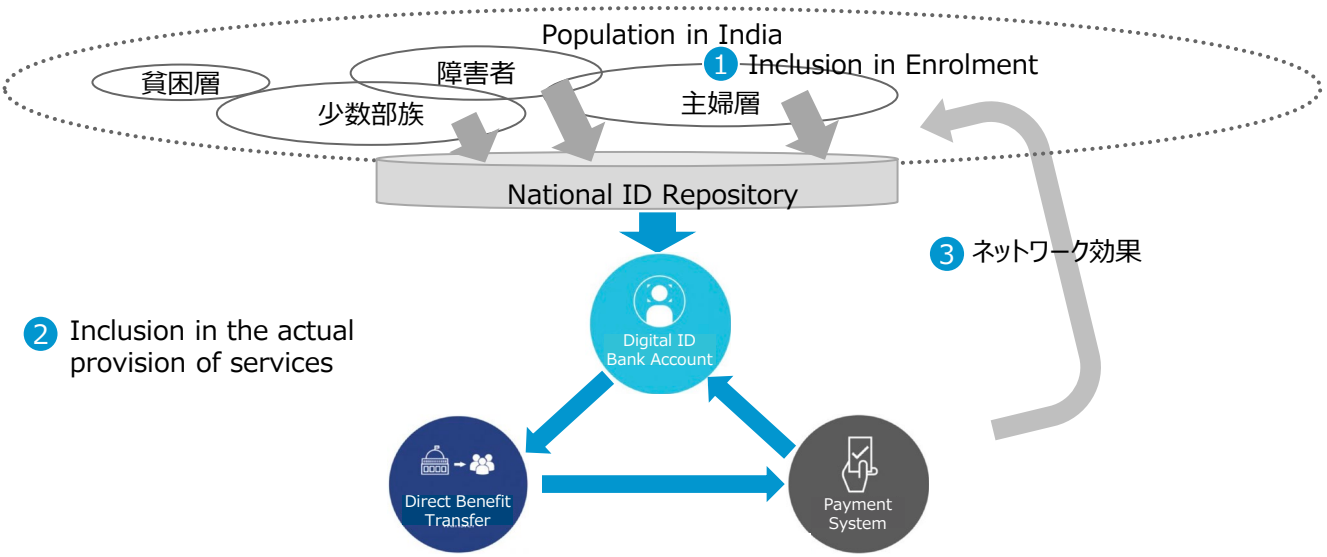
インドでは、民族や階級の垣根を越えて、広く遍く個人IDのInclusion(参画)を促進することをゴールにしている

1 **Inclusion in Enrolment**
クリティカルマスとなり得る層の参画をVoluntaryベースで促進
- 貧困層／少数部族／トランスジェンダー／ホームレス
- 旧来は世帯単位だった主婦層SIMカードの普及が促進
- 障害者や特定職務（炭鉱従事者等）

2 **Inclusion in the actual provision of services**
個人認証をベースとしたサービス利活用を促進し、IDのアクティブ率を上げる

3 **ネットワーク効果**
「皆が利用しているから私も利用したい」という付加価値を生み出し、Inclusionが加速する

詳細



ポイント

- **社会的弱者の取り込みによる新たな市場ターゲットの創出／経済活動の促進**がMOSIP導入のゴールとなっている
- 取組みを成功させるためには、個人認証をベースとしたキラーサービスの導入が必須。**Indian Stackでは補助金供給や決済サービス導入によりお金が還流する仕組みが導入された**
- National IDは**政府主導の取り組みだが、Inclusionには強制力は持たせず市場原理に基づいたオーガニックなID蓄積を指向**している
- **インド全人口に対するカバレッジが高まり、以下の課題が顕在化している**
  1. Social Exclusion（実態としてMandatory Inclusionが必要）
     デジタルIDが発行されていない女性が食糧補助金の利用を拒否される
  2. 個人データの所有権／許諾
     国家によるCensorshipへの恐れや過去の民間企業の不正利用等

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

(再掲) 1.1.4 オープンイノベーション／エコシステム

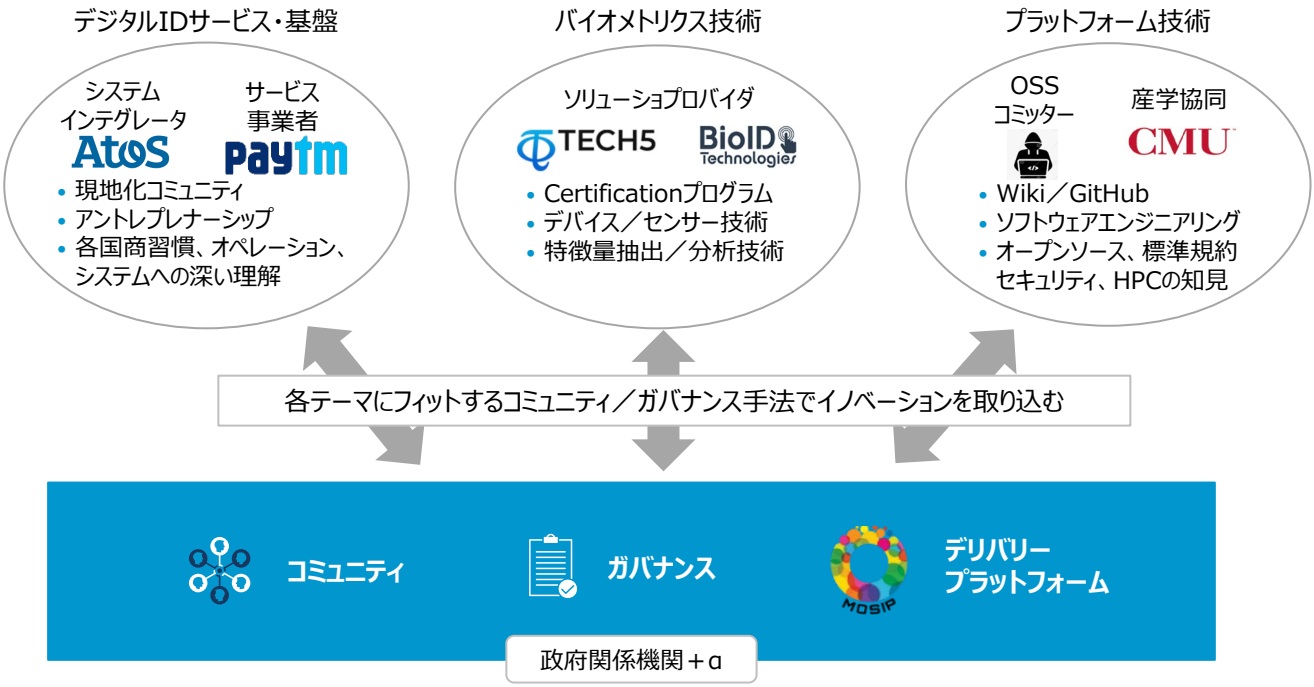
概要

MOSIPはシステム開発、生体認証デバイス、セキュリティ管理、法制度等、各分野に強みを持つ複数プレイヤー協同での運営を前提としている

対象テーマは大きく以下の 3 領域があり、各テーマ毎にコミュニティ／ガバナンス手法を使い分けて、イノーベーションを取り込んでいる

1. デジタルIDサービス・基盤
2. バイオメトリクス技術
3. プラットフォーム技術

詳細



ポイント

- **各国に合わせたエコシステム検討**
  - 各国の事情（母国語、技術者リソース、輸出入規制等）を鑑みたエコシステム検討が重要
  - その上で日本の民間企業／学術機関の立ち位置、参画意義が明確になる
- **コミュニティ／ガバナンス運営者のリーダーシップがKSF**
  - エコシステムを管理・運営するためのガバナンスモデルとリーダーシップが、エコシステム成功のカギとなる

Source: National Open Digital Ecosystem WhitepaperをベースにBCGにて整理

1.1.5 オープンソース・プラットフォーム

オープンなAPI、オープン標準、オープンソースでベンダーの囲い込みを防止し、相互運用性を確保

MOSIPの実装を成功させるには、国ごとに最も関連性の高いMOSIPモジュールを選択して組み込めることが極めて重要である。さらに、デジタルIDを福祉給付金やその他のサービスに使用するには、住民登録、人口登録、その他の機能登録との相互運用性が欠かせない。これを促進するために、MOSIPはAPIファーストのアプローチをとっている。これは、組織の目標や開発者のニーズに沿ったAPIを設計し、プラグアンドプレイを可能にして、効果的にカスタマイズしたソリューションを構築する設計手法である。

オープンで相互運用性のあるものにする

MOSIPでは「オープン・テクノロジー」を利用してベンダーの囲い込みを防いでいる。これにより、デジタルIDシステムが規制やテクノロジーの変化に対応でき、プラットフォームのアジリティが高い。APIファーストのアプローチに従うことで、相互運用性が確保される。（たとえば、MOSIPを実装する国々では、システムを既存のデータベースと統合して、その国の法律に従って収集されるID情報を追加できる。）

国の要件に合わせたモジュール選択

MOSIPはばらばらのモジュールになっており、これらのモジュールを国のニーズに応じて組み合わせ一意なIDシステムを作成できる。これにより、ソリューションを簡単にカスタマイズでき、リエンジニアリングの必要がなくなる。この機能は多大な柔軟性を提供し、技術をゼロから開発するコストと時間が不要になるため、各国にとって特に価値が高い。

疎結合にして、拡張性を高め、結び付ける

MOSIPは、それぞれがIDの発行、更新、認証、ユーザーコントロールなど、複数の機能を実現するいくつかのモジュールを提供している。（たとえば、フィリピンでは現在、データを収集して個人にIDを発行できるようにするID発行モジュールを使用した国家IDシステムを構築中である。）

プライバシーとセキュリティを一体化

MOSIPはユーザーがデータの所有者であり、最大限の透明性を確保することで、ユーザーのIDと個人情報を保護するよう設計されている。また、プラットフォームは不正アクセスから確実に保護されている。組み込まれている基本的な設計要素は、データ暗号化、セキュアAPI経由のIDリポジトリへのアクセス、ユーザー同意フレームワーク、デジタル署名、データの暗号認証、データの匿名化、レイヤー構造のテクノロジー・アーキテクチャである。また、一度だけ使用する限定されたデータへの取り消し可能なアクセスを許可する「仮想ID」や、利用状況の通知やリアルタイムデータを表示して透明性を提供する機能、認証やeKYCをロックする機能、オフライン認証などの高度な機能も組み込まれている。

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.2.1 MOSIP組織体制

事務局

IIITB
(International Institute of Information Technology Bangalore)

資金調達担当

BILL & MELINDA GATES foundation

TATA TRUSTS

OMIDYAR NETWORK

技術委員会

Sanjay Jain, Volunteer, iSPIRT

Prof. Chandrashekhar Ramanathan, IIITB

Prof B Thangaraju, IIITB

Satish Mohan, dhiway

執行委員会

Prof. S Sadagopan, IIITB

Prof S Rajagopalan, IIITB

CV Madhukar, Omidyar Network

Himanshu Nagpal, Bill & Melinda Gates Foundation

Shloka Nath, Tata Trusts

Sharad Sharma, iSPIRT

Sanjay Anandaram, iSPIRT

Prof. Amit Prakash, IIITB

Anuj Gangwal, Tata Trusts

Liv Marte Nordhaug, Norad

国際アドバイザリーグループ

Joseph Atick, ID4 Africa

Alan Gelb, Centre for Global Development

Michiel Van der Veen, European Association for Biometrics

Tomicah Tilleman, Director of the Blockchain Trust Accelerator

Adam Cooper, World Bank

Sudarshan Sen, Reserve Bank of India

Edward Duffus, Plan International

Andrew Hopkins, UNHCR

Jean Philbert Nsengimana, SmartAfrica

Anuchit Anuchitanukul, Thailand

Vyjayanti Desai, ID4D, World Bank

Eileen Donahoe, GDPI, Stanford University

Dame Wendy Hall, Web Science Trust

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.2.2 パートナーシップ

ユースケース・レイヤー:
IDと結び付いた導入国固有のサービス

基本IDとサービスの結び付き
- 財務、医療、健康等のサービスと協力して一意な基本IDを導入

システムインテグレーター・レイヤー:
導入各国のカスタマイズ

導入国のニーズに合わせたカスタマイズ
- 必要な場合に追加モジュール、コンフィギュレーション、セキュリティを開発するシステムインテグレーター(ベンダー)
- MOSIPによって生み出され、育成された商用サービスプロバイダーがシステムインテグレーターにプラットフォームのメンテナンスサポートを提供

コアテクノロジー・レイヤー:
MOSIPプラットフォーム

IDシステムの基本アーキテクチャ
- モジュール型
- 国によらない
- ベンダーに依存しない
- 堅牢性とセキュリティを確保

MOSIPでは …

- オープンソースライセンスの下でMOSIPカーネルを提供
- 包括的ドキュメンテーション
- カーネルの5年間の機能拡張、サポート、メンテナンス
- MOSIPテクノロジーに関するトレーニングおよび教育
- 認証サービス・プロバイダー・プログラム:
  国レベルでのMOSIP実装経験を有するサービスプロバイダーを審査し、認証してグループ化

MOSIPパートナー

- カスタマイズおよびシステム統合サービス
- 国レベルの技術コンサルティング
- MOSIPを基礎とした公開サービスおよび非公開サービス提供のためのソリューション構築
- 国およびその他のユーザー組織向けサポート、メンテナンス、機能拡張サービス

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.2.3 MOSIPコミュニティ

インクルージョンのための設計

MOSIPは個人につながる最後の一歩を提供することで、IDシステムを普及しやすくする。オンラインとオフラインの両方で動作するオプションを提供し、ネットワークに要求される負荷を低減する。たとえばオフラインでの認証は、ネットワークやインフラストラクチャーが脆弱であるなど、特にリソースが限られた環境で有効である。

テクノロジーの可能性を広げる学術専門家とのパートナーシップとオープンソースのコミュニティ・レビューを通じた参加型設計

国家インフラストラクチャーとして欠かせないIDシステムの役割に鑑み、セキュリティの拡張機能はMOSIPの最重要事項の1つである。したがって、MOSIPは機械学習（ML）や人工知能（AI）で専門知識を持つ米国（USA）や英国（UK）の最高レベルの教育機関と連携することで、技術力を高めている。また、公共部門や国際的な開発機関の専門家で構成される国際諮問委員会も任命されている。MOSIPは定期的に専門家の意見を聞く場を設け、テクノロジーやプラットフォームの実装面に関するフィードバックを求めている。

参加型設計とエンドユーザーの関与を促進	MOSIPは、技術研究機関の専門家とデジタルIDシステムを実装している国の専門家の両方を活用し、プラットフォームの技術と実装能力に関するフィードバックを求めている。さらに、ソースコードを公開することで、MOSIPは開発者、試験担当者、ユーザーなどのコミュニティによるレビューを促進して、プログラムの堅牢性を確保している。（たとえば、MOSIPはオペレーティングシステム間の互換性の問題や、同意やオプトアウトのフレームワークの強化などMOSIPのロードマップに対する改善要望などについて、GitHub上でフィードバックを受け取る。）
民間企業のエコシステムを育むことで共同開発を行い、関連するサービスを提供	MOSIPはあらゆる国に合わせて設定およびカスタマイズができるよう設計されている。したがって、システム・インテグレーターのコミュニティを作り、MOSIP上にその国固有のシステム（文書管理システム、ABIS、GPS、ロケーション、IDカード印刷を含む）を構築して、国家デジタルIDシステムを実装している。MOSIPのパートナーシップ・アプローチは2つのモデルに従う。1つ目のモデルでは、MOSIP上でシステムを組み込めるサービス提供者に研修を施し、MOSIPが入念に精査する。エコシステムを作り出すため、MOSIPはIDソリューション・エンジニア、生体認証やデバイスのベンダーを招いてMOSIPとの統合を実習するワークショップを実施している。2つ目のモデルでは、各国が独自にベンダーを調達し、MOSIPがコアテクノロジーのトレーニングと教育、メンテナンスといったサービスを行う。これらのモデルにより、各国はデジタルIDシステムを開発する上での最大の難関（技術力など）を柔軟に乗り越えることができる。今後、MOSIPは統合プロセスの自動化を計画している。これにより、ベンダーは試験結果をMOSIPとオンラインで共有でき、MOSIPはそれに基づいて独立に試験を行って、統合がうまくいったことを検証することができる。
イノベーターのネットワークを醸成する	MOSIPはベンダーを招いてMOSIP上にシステムを構築するワークショップやイベントを実施している。ベンダーはMOSIPのインテグレーションを正しく行う能力を有するシステム・インテグレーターのコミュニティを精査してトレーニングを施す。（たとえば、4つの生体認証ソフトウェア会社が自社のABISとMOSIPを統合している。これらのシステムは登録時に取得される指、光彩、顔などの生体認証画像の品質を評価するために使用できる。）
公共と民間が手を組むことで実装を支援	MOSIPは公共部門と民間企業が協力して実装する。実装に関与するステークホルダー（ベンダー、政府、民間パートナー）のそれぞれについて、責任を定義する包括的構造が描かれている。また、MOSIPはデジタルIDシステムを実装するシステム・インテグレーター向けの包括的なマニュアルを定義しており、特に実装チームに必要な能力やスキルが明確にされている。ただし、実装する国に能力が不足していたために、MOSIPがギャップを埋めなければならない事例もあった。たとえば、国の背景に合わせてコアプラットフォームをカスタマイズするなどである。これは、一般には民間パートナーが国の担当者に企業価値を示すために行うタスクである。円滑な実装のためにMOSIPは政府や営利企業のエコシステムから独立しているため、このことはMOSIPに課題を提示した。

Source: MOSIPの公開ドキュメントをベースにBCGにて整理

1.3 MOSIP導入にあたってのメリット（1/2）

デザインの基本方針	プラットフォームのアーキテクチャを決定しようとする際には、最新のテクノロジーとプラクティスを検討しがちである。しかし、基盤IDシステムのような極めて重要なプラットフォームでは、目的に沿った、周囲のエコシステムをサポートするテクノロジーを選ぶのが堅実である。本セクションでは、MOSIPのテクノロジーの選択につながる主要因について検討する[1)]
成熟度	MOSIPには、導入国のソフトウェア開発者のエコシステムが、既存の知識を活用して簡単に導入できるテクノロジーを選択すべきである。MOSIP開発チームは長期にわたって使用およびデプロイが可能なテクノロジーを選んでいる。
パフォーマンス	基盤IDシステムは、人口規模に合った合理的なパフォーマンスを提供する必要がある。単位時間あたりのIDサービスのリクエストをスムーズに処理し（スループット）、ユーザー体験を低下させることなく個々のリクエストに迅速にレスポンスを返す（応答時間）ことが求められる。全国民に一意なIDを正しく発行すること、ID照合を無限に低い誤り率で行うことのどちらにも高次の正確性が必要である。経年変化、環境条件、および開発ペース等の外的要因に対して、システムは安定で障害耐力を有するものでなければならない。
スケーラビリティ	システムはユーザー数が多くなっても予測的に動作を継続できなければならない。大容量かつ高速で多様なデータ（しかも時間とともに増大し続ける）を高い信頼性をもって処理できる必要がある。ソフトウェアまたはハードウェアのボトルネックは熟知され、スケーラビリティの制限とならないこと。帯域幅が制限されている環境でも良好に動作しなければならない。
セキュリティ	認証されていないアクセスや不正使用を防止できなければならない。攻撃に対する回復力と、セキュリティ違反や攻撃から復旧する機能を有すること。安全な通信チャネルを使用すること。システム内のすべてのアクセスおよびトランザクションが監査可能であること。
手頃な価格	システムは経済的であること。つまり、ハードウェアとソフトウェアはコスト効果が高いものでなければならない。そのため、MOSIP構築チームは汎用ハードウェアベースのインフラストラクチャを活用する。
発展性	テクノロジーおよび規制のポリシーは時間とともに進化する。システムのアーキテクチャは、この進化を多大な再設計なしに簡単に取り入れられるものでなければならない。
オープンスタンダード、オープンソース	銀行、通信、その他サービス業で広く使用されるように、また他のサービスにシームレスに統合できるようにするには、オープンスタンダードの利用が重要である。アーキテクチャはアプリケーション、プログラム言語、およびプラットフォームを問わないものでなければならない。たとえば、XMLまたはJSONをドキュメント交換フォーマットに使用する、HTTP over REST等のオープンプロトコルを使用する等である。ソフトウェアは公開され、ドキュメント化され、アプリケーション・プログラミング・インターフェース（API）によりアクセスできなければならない。プラットフォームではオープンプロトコルとAPIを効果的に使用してプラグインが動作し、拡張可能でなければならない。今日、ソフトウェアテクノロジー分野ではオープンソースを中心に多くのイノベーションが生まれている。Hadoop、Docker、およびLinuxはオープン・ソース・テクノロジーのエコシステムの一例である。オープンソースプラットフォーム化することで、基盤IDシステムは開発者、テスター、その他の専門家の大きなコミュニティに開かれたものとなり、コミュニティによって継続的にフィードバックされ、新規機能やバグ修正への積極的な貢献が得られるようになる。オープンソースのプロジェクトにおける導入や進化のペースのほうが独自の実装よりも速いテクノロジーには多くの例がある。MOSIPアプローチの最大の利点は、導入国が世界中の専門家の構築した「ベストプラクティス」のシステムを得られることである。強固なオープンソースコミュニティによって、セキュリティ、品質、柔軟性、相互運用性、およびサポートが確実なものとなる。
モジュラリティ	プラットフォームにはメインの機能が1つなければならない。エコシステムの革新を成し遂げる鍵は、オープンなインターフェースを使用して機能とインテグレーションを分けておくことである。これにより柔軟性ももたらされ、政府機関はプラットフォームの各局面を個別に進化させることができる。導入国は希望するデザインに適したモジュールを選択しやすくなる。

1. Technology Landscape for Digital Identification（デジタルIDに関する技術的見通し）(http://pubdocs.worldbank.org/en/199411519691370495/ID4DTechnologyLandscape.pdf)

1.3 MOSIP導入にあたってのメリット（2/2）

プライバシー・バイ・デザイン	プラットフォームは安全なものとなるよう設計され、プライバシーを徹底的にサポートするものでなければならない。システム内のすべてのトランザクションおよびレコードデータは、デジタル署名される。共通APIはアクセス制御、監査、（暗号化による）機密性、（署名による）一貫性を確保する。データへの直接のアクセスはできず、システムのエントリポイントから、十分な認証レベルを有するオープンAPIを通じてのみアクセスできる。機密データ（個人識別情報等）は、ストレージ内でも通信中でも機密性が守られなければならない。すべてのサービスおよびデータへのアクセスは、監査証跡として記録される。
人口規模のデザイン	プラットフォームは人口規模、つまり何億人という住民に対応して動作しなければならない。スケーリングの従来のアプローチは、垂直的、つまり処理能力とストレージを積み上げていくものだった。これではコストがかかるだけでなく、システムの全体的なスケーリング能力にも限界があった。したがって、システムが水平的にスケーリングできることが重要になる。これは、システムのコンポーネントをクラウド対応とし、展開や置換が簡単な汎用ハードウェアを活用することで実現できる。
進化を促すミニマリスティックなアプローチ	デザインはシンプルで必要最小限であるべきである。論理レイヤーと機能境界は明確に分離し、明確に記述されたAPIのみを使用してその間の通信を行う。システムのデザインは、進化できるもの、つまり、新しい機能を段階的に構築しつつ、プラットフォームがそれを迅速に取り入れることができるものでなければならない。すべてのコンポーネントは独立に取り替えることができ、拡張できなければならない。疎結合のマイクロサービス型アプローチがこの原則を体現している。
設定の自由度とカスタマイズ性	住民固有のアイデンティティを構成する属性等、導入国にはそれぞれ固有の要件があるため、IDプラットフォームは一律のアプローチをとることができない。MOSIPは開発工数を増やすことなくこれらの多様なニーズの解決を支援する。導入国は理想的にはドキュメントの揃ったオープンソースの「カーネル」を設定し、使用するモジュールを取捨選択して、0からの技術開発にかかる時間とコストを節約し、ソリューションとしてのIDシステムを構築できる。

2.
アーキテクチャ

2.1.1 アーキテクチャの基本方針

ベンダー囲い込みを防ぐ

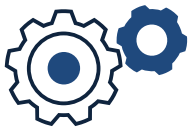
MOSIPは、**知的財産権で保護された、または商用のライセンスフレームワークを使用してはならない。**必要不可欠とみなされる場合、そのコンポーネントは**カプセル化**して必要に応じて置き換えられるようにしておかなければならない

オープン標準を使用する

MOSIPは、**オープン標準**を使用してその機能を開示しておかなければならない(テクノロジーの囲い込みを防ぐため)

水平拡張が可能

変化する負荷要件に対応するため、MOSIPの各コンポーネントは**独立してスケーラブル (スケールアウト)** でなければならない



コモディティコンピューティング

MOSIPでは、**一般的なコンピューティングハードウェアおよびソフトウェア**を使用してプラットフォームを構築しなければならない

データプライバシーの確保

データは**送受信時も保存時も暗号化されなければならない。**すべての**リクエスト**は**認証**および**認可**されなければならない。MOSIPにおける**個人識別データの プライバシー**は絶対に守られなければならない

プラットフォームベースのアプローチ

プラットフォームベースのアプローチに従わなければならないので、すべての共通機能は再利用可能なコンポーネントおよびフレームワークとして共通レイヤーにまとめられる

APIアプローチ

MOSIPは**APIファーストのアプローチ**に従う必要があり、REST原則に従ったサービスとして企業に開示される

管理性

システム内のすべてのイベントの**監査および監視**が可能であること、プラットフォームの全機能が**試験**できること、およびプラットフォームが簡単に**アップグレード**できること、といった**管理性**原則に従わなければならない

疎結合

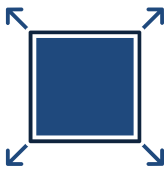
MOSIPコンポーネントは導入国の要件どおりに**構成**してIDソリューションを構築できるよう、**疎結合**でなければならない

国際化をサポート要

MOSIPには、**国際化をサポート**する機能がなければならない

耐障害性

ソリューション全体が耐障害性を備えたものになるよう、MOSIPのすべてのモジュールには回復力がなければならない



拡張可能

MOSIPの主要なサブシステムは、**拡張できるよう**設計すること。たとえば、外部システムを統合して指紋データに対応させる場合、簡単に可能でなければならない

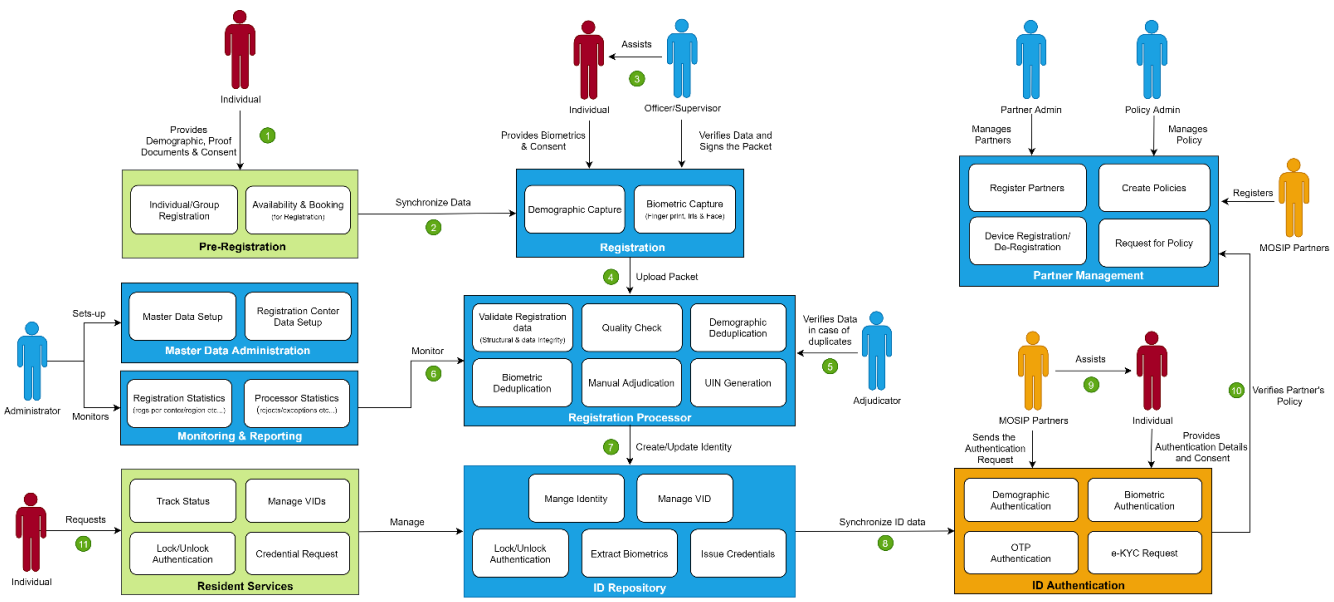
2.1.2 Modularアーキテクチャ（1/2）

1. 機能アーキテクチャ

MOSIPの観点では、IDシステムにはいくつかのアクターが含まれる

- すべての中心である**個人 (Individual)** または**住民 (Resident)**。システムでは、個人や住民の身元情報が扱われる
- ID発行者の代表である**担当者 (Officer)**。担当者はいくつかの専門的な役割に分かれている
  - 登録時に個人を補助する**オペレーター (Operator)**
  - 登録における例外事例を検証し、テストする**監督者 (Supervisor)**
  - ID発行プロセスにおいて手作業で個人データの検証や比較を行う**調整者 (Adjudicator)**
  - 特別な申請の監査またはフォレンジック分析を行う **監査者 (Auditor)**
  - システムの設定データを管理するスーパーユーザー、**管理者 (Administrator)**
- MOSIPと相互に作用するサードパーティのサービスまたはアプリケーションを担当する**パートナー(Partner)**。専門家のパートナーがエコシステムを構成する
  - **依存パーティー (Relying Party)** とは、IDシステムに依存してビジネス取引を行う**認証パートナー (Authentication Partner)** である給付金支給のための社会的スキームや、口座開設のための銀行などが該当す
  - **認証プロバイダー (Credential Provider)** は、認証関連を扱う印刷サービス事業者である
  - **デバイスプロバイダー (Device Provider)** は、生体認証デバイスを提供するパートナーである
  - **FTMプロバイダー (FTM Provider)** は、デバイスの基幹モジュールを提供するパートナーである
  - **パートナーアプリケーション (Partner Application)** とは、MOSIPに依存するシステム、またはMOSIPが依存するシステムである。CRVSシステム、パスポートや運転免許証などの機能IDシステムが該当する

IDシステム (ID System) とは、MOSIPによって統合され、IDが相互運用可能となるシステムである。下記の図に、各アクターとMOSIPの機能アーキテクチャを示す。



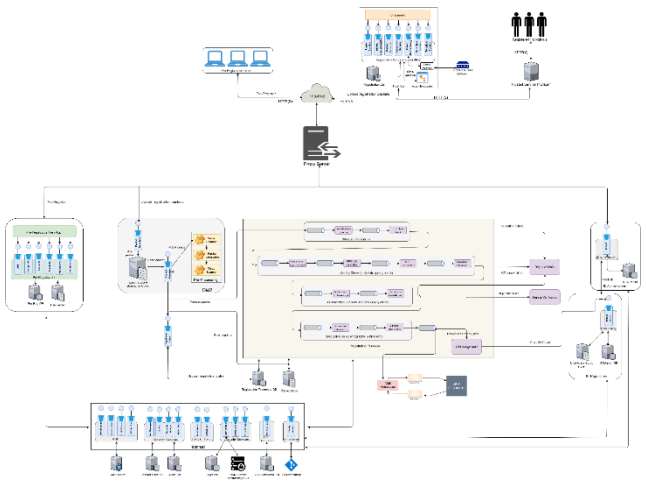
2.1.2 Modularアーキテクチャ（2/2）

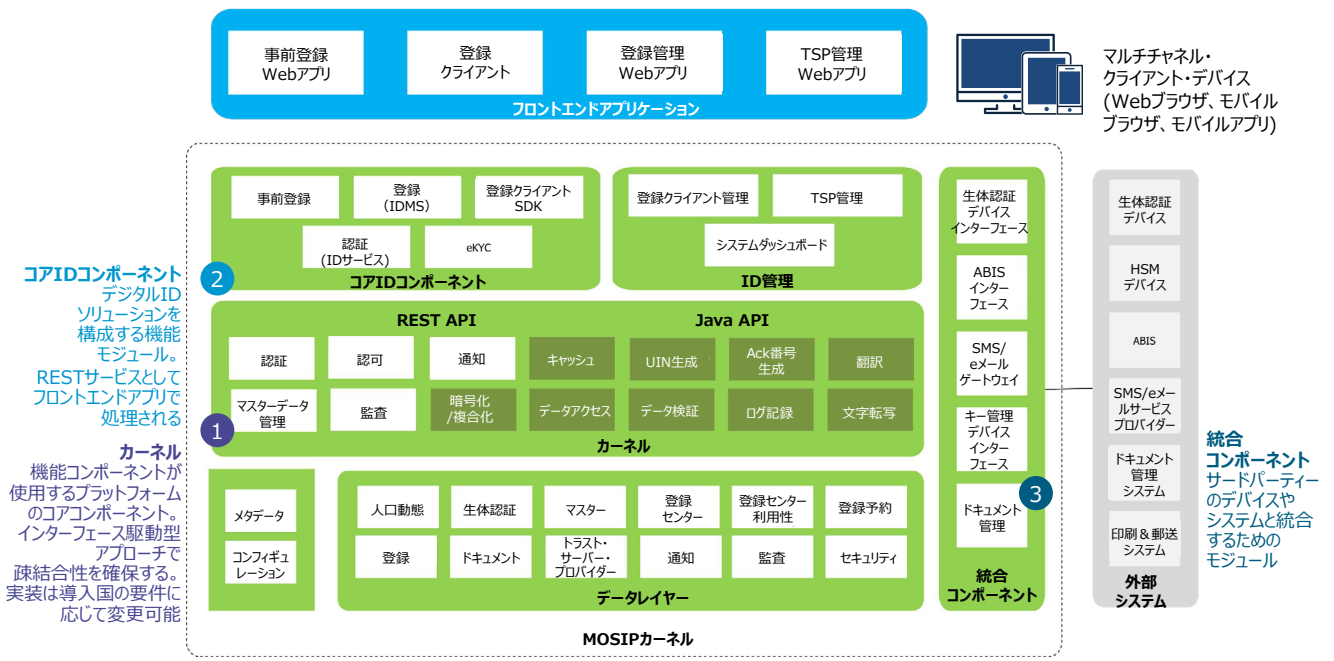
2. モジュール型アーキテクチャ

MOSIPには関連し合って機能を提供するいくつかのモジュールがある。これらには次のコアモジュールが含まれる

- 事前登録 (Pre-registration)
- 登録クライアント (Registration client)
- 登録プロセッサ (Registration processor)
- IDリポジトリ (ID repository)
- 認証 (Authentication)
- 住民サービス (Resident Services) およびサポートモジュール
- パートナー管理 (Partner Management)
- 管理 (Administration)
- レポーティング (Reporting)

下図に様々なMOSIPモジュールとそれぞれのサービス群およびその関係性を示す。





Note: すべてのユーザー・インターフェース・モジュールはリファレンス実装であり、実際のデプロイで現状のまま使用するか、リファクタリングやカスタマイズを行って、または代替実装に置き換えて使用できる

2.1.3 プライバシー & セキュリティのデザイン（1/3）

1. MOSIPにおけるプライバシー & セキュリティの目的

プライバシー

1. 個人がデータの所有者
2. 透明性は組み込み

セキュリティ

1. 個人・パートナー・政府が信頼して利用できるシステム
2. 国家やテロ攻撃からの保護
3. 本人の検証を制限できる能力を保有
4. 必要に応じてIDを取り消すことができる
5. IDの増殖を抑止する

2. プライバシーのデザインエレメント: ベーシック

個人情報の取り扱いについて

- 匿名比較のための安全な一方行ハッシュ
- 移動時および格納時のデータの暗号化

アイディーリポジトリアクセスサービス

- データベースへの問い合わせを制限
- アプリケーションによる暗号化/復号化処理
- 特定の機能を持つステートレスAPIを提供し、安全性を確保します (証明書ベースの認証 - TL1)

利用者の同意

- ユーザーのデータへのアクセスは、すべてユーザーの同意が必要

2. プライバシーのデザインエレメント: 機能性

UIN

- 重複除去後に個人に割り当てられる一意の12桁の乱数

バーチャルID

- IDの復元を有効にして、盗用を防止

トークンID

- 360度プロファイリングを抑止

プロフィール共有の限定

- 限定的なデータ共有、ユーザー中心のポリシーを提供

履歴とアラート

- 改ざん不可能なデータで透明性、通知、リアルタイムの使用状況を提供

認証ロック

- 認証およびeKYCの特定の機能をロックまたはロック解除する機能

セキュアなオフライン認証

- オフライン認証モードでもデータのプライバシーを確保

2.1.3 プライバシー & セキュリティのデザイン（2/3）

3. トークンID

ユーザーはUIN/VIDを使用して一回認証とeKYCの認証を実施

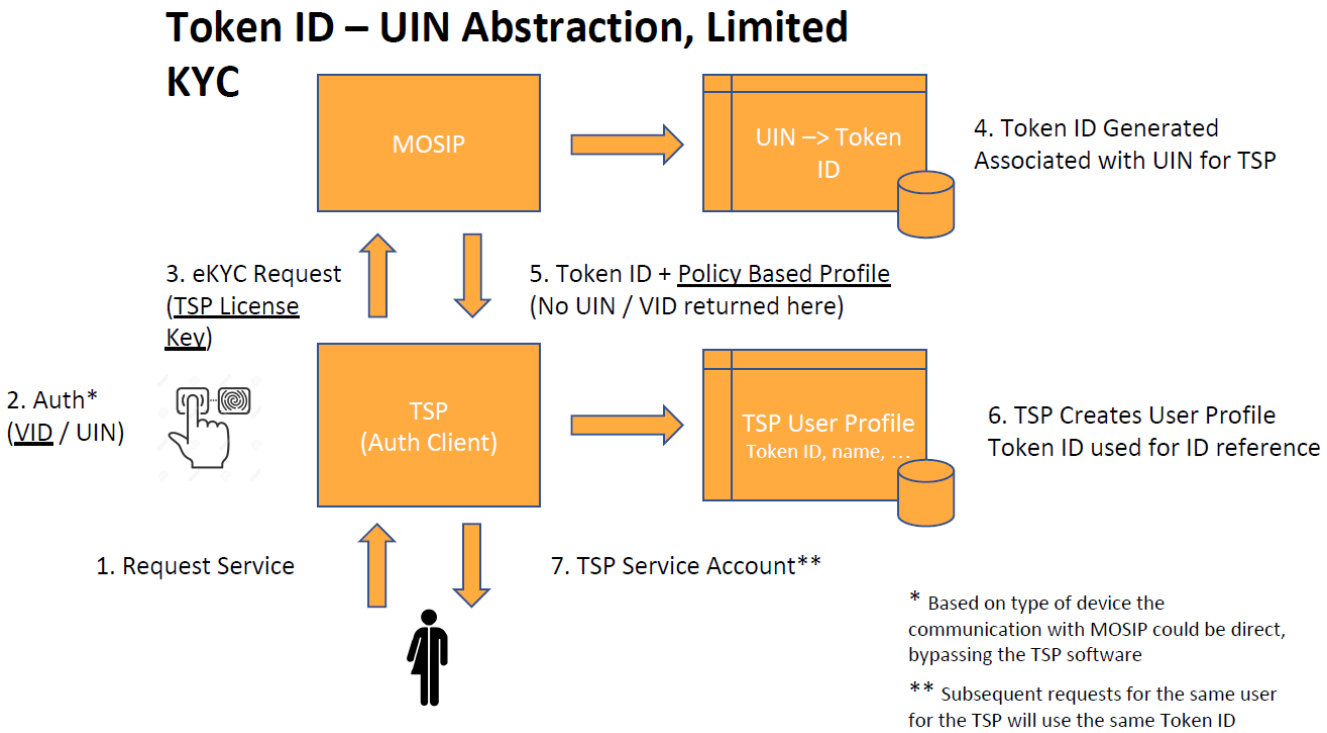
- ユーザーが提供した　第三者が情報にアクセスできない別のセキュアなチャネルで実施

第三者機関はUINを取得せず、トークンIDを取得

- 以降ののエージェンシーとの取引では、　同じトークンIDを使用

トークンIDはサードパーティ機関ごとに異なり、プロバイダ間のプロファイリングを抑止

トークンIDの取り消しやブロックが可能



2.1.3 プライバシー & セキュリティのデザイン（3/3）

4. 認証のオフラインサポート

- 最新のIDをQRコード形式でダウンロード
- 認証には暗号化されたデジタル署名付きのQRコードを使用
- OTPのようにユーザーと共有される一回限りの暗号化キーの生成
- 有効期限が明確なQRコードの発行
- 最小KYCまたはフルKYCをQRコードに格納することが可能

5. Zero Knowledge Storage

1. 提供された実際の個々のデータは、そのライフサイクルを通して暗号化された形で格納
2. 処理されたデータは、アプリケーションの暗号化されたデータベースに保存
    a. 各カラムは鍵で暗号化
    b. キー: データ
    c. 各行には、データの変更を保護する HMACを付与
    d. 暗号化に使用されるすべてのキーは、順番に単一のマスターキーで暗号化された特別なデータベースに格納
    e. 鍵データベースへのアクセス権は監視され管理

6. 認証のために保存されたデータのコピー、eKYC

a. 各レコードはデジタル署名
b. 常にVIDをメインキーとして保存
c. データは単一レコードとして暗号化
    i. Hash(VID) -> enc(Kid(hash(VID), record) ここで、Kid は与えられた VID の AES 鍵
    ii. Kid は HSM のマスター AES 鍵から派生したもの

7. 脆弱性と脅威への対応

- デフォルトの優先資産 (Dockers)
- 研究者による公式の脆弱性報告
- 脅威はDREADでモデル化され、STRIDEで評価
- DevSecOpsでは、ビルド毎に依存関係、SAST、コンテナ (OpenSCAP) の脆弱性をスキャン
- セキュリティ勧告の公式出版物
- 推奨セキュリティポリシーハンドブック

2.2.1 テクノロジースタック（1/2）

テクノロジースタック

- このページには、MOSIPの構築に使用されているすべてのテクノロジーをリストしている。可能な限り、明らかに長期サポートがあることがわかっている無料のオープンソースソフトウェアが選択されている。デプロイにあたってはその他の無料または商用オプションと置き換えることができるものもある

領域	ツール/テクノロジー	バージョン	ライセンス種別
オペレーティングシステム	CentOS	7.7	MITライセンス
インフラストラクチャ	クラウド - Azure/AWS	不明 - クラウドツール	商用
開発 - 言語 ランタイム	Java SE 11	OpenJDK 11	Oracleバイナリコード・ライセンス
開発 - 言語 ランタイム	J2EE	Java EE 8	GPL
開発 - UI アプリケーションフレームワーク	JavaFx	OpenJFX 11	GPL v2 + Classpath
開発 - アプリケーションフレームワーク	Vert.x	3.5.1	Apacheライセンス2.0
開発 - アプリケーションフレームワーク	Spring	5	Apacheライセンス2.0
開発 - ユーティリティー	Apache Commons (60以上が候補となる)	最新版	Apacheライセンス2.0
開発 - データグリッド	Apache Ignite	2.4.0	Apacheライセンス2.0
開発 - オブジェクトマッパー	Orika	1.5.2	Apacheライセンス2.0
開発 - バリデーター	Hibernate validator	5.4.2	Apacheソフトウェア・ライセンス2.0
開発 - 暗号化	BouncyCastle	1.59	MIT X11ライセンスを適用
開発 JSON Marshal/Unmarshal	Jackson	2.9.5	Apacheライセンス2.0
開発 - デバイスドライバー	RXTX	RXTX-2-2-20081207	LGPL v 2.1
開発 - 単体試験	Junit	5.x 以降	Common Public License - v 1.0
開発 - ログ	logback	1.2.3	GNU Lesser GPL Version 2.1
開発 - テンプレート作成	velocity	2	Apacheライセンス2.0
開発 - ツール	Open street view	不明 - クラウドツール	Open Database License (ODbL)
開発 - IDE	Eclipse Oxygen	4.7.3	Eclipse Public License Version 2.0
開発 - Webアプリ	Angular	4+	MITライセンス
開発 - 単体試験	Karma	2.0.x	MITライセンス
開発 - 単体試験	Jasmine	2.6.1	MITライセンス
開発 - APIドキュメンテーション	Swagger	3.13.2	Apacheライセンス2.0
開発 - アプリケーションサーバー	Tomcat server	8	Apacheライセンス2.0
開発 - オーケストレーション	Apache Camel	2.19.3	Apacheライセンス2.0

2.2.1 テクノロジースタック（2/2）

領域	ツール/テクノロジー	バージョン	ライセンス種別
開発 - WebSub	Ballerina Websub	1.2.8	Apacheライセンス2.0
開発 - データベース	H2 DB	1.4.197	
開発 - データベース	PostgreSQL	サーバー：10	Postgresライセンス
開発 - データベース	PostgreSQL	ドライバ：42.2.2	二条項BSDライセンス (簡易ライセンス)
開発 - データベース・モデリング・ツール	PG Data Modeler	0.9.2	商用
開発 - Postgres管理/スクリプト/クエリ開発ツール	pgadmin4	3	ライセンスフリー
開発 - プログラム品質	Sonar	7.2	オープンソース・ライセンス
開発 - UI設計	Pencil Project	3.0.4	GNU Public License Version 2
生体認証デバイス	未定	-	-
生体認証API	未定	-	-
試験ツール	Rest-assured	3.0.0	Apacheライセンス2.0
試験ツール	WireMock または Citrus framework	2.16.0 または各種	Apacheライセンス2.0
試験ツール	JMeter	4.x	Apacheライセンス2.0
試験ツール	Burp suite Professional +	9.0.3.7	PortSwigger - Burp suite Professional + / V1.7.33
試験ツール	TestNG	6.11	Apacheライセンス2.0
DevOpsツール	Jira	6.4以降	オープンソースなし
DevOpsツール	SonarLint	v3.5	GNU GPL
DevOpsツール	GitHub	2.7.x	商用 - Github
DevOpsツール	SonarQube	6.7.3 LTS	GNU GPL
DevOpsツール	Maven	3.53.x	Apacheライセンス2.0
DevOpsツール	Cobertura	1.12以降	GNU GPL
DevOpsツール	JFrog Artifactory (OSS)	5.9	不明
DevOpsツール	Docker	18.03.x CE	Apache 2.0
DevOpsツール	Ansible	2.2	GNU GPL v3.0
DevOpsツール	Githubアクション	不明 - クラウドツール	-
DevOpsツール	Travis	不明 - クラウドツール	MITライセンス
DevOpsツール	Glowroot	-	Apacheライセンス2.0
DevOpsツール	Prometheus	-	Apacheライセンス2.0
DevOpsツール	Grafana	-	Apacheライセンス2.0
メッセージング	ActiveMQ	-	Apacheライセンス2.0
コードの安全性スキャニング	OWASPプラグイン付きSonarQubeを使用予定	-	-
Webサーバー/HTTPプロキシサーバー	Nginx	不明 - クラウドツール	-
IAM	Keycloak	-	-

2.2.2 データアーキテクチャの基本方針（1/3）

1. MOSIP内のデータセット

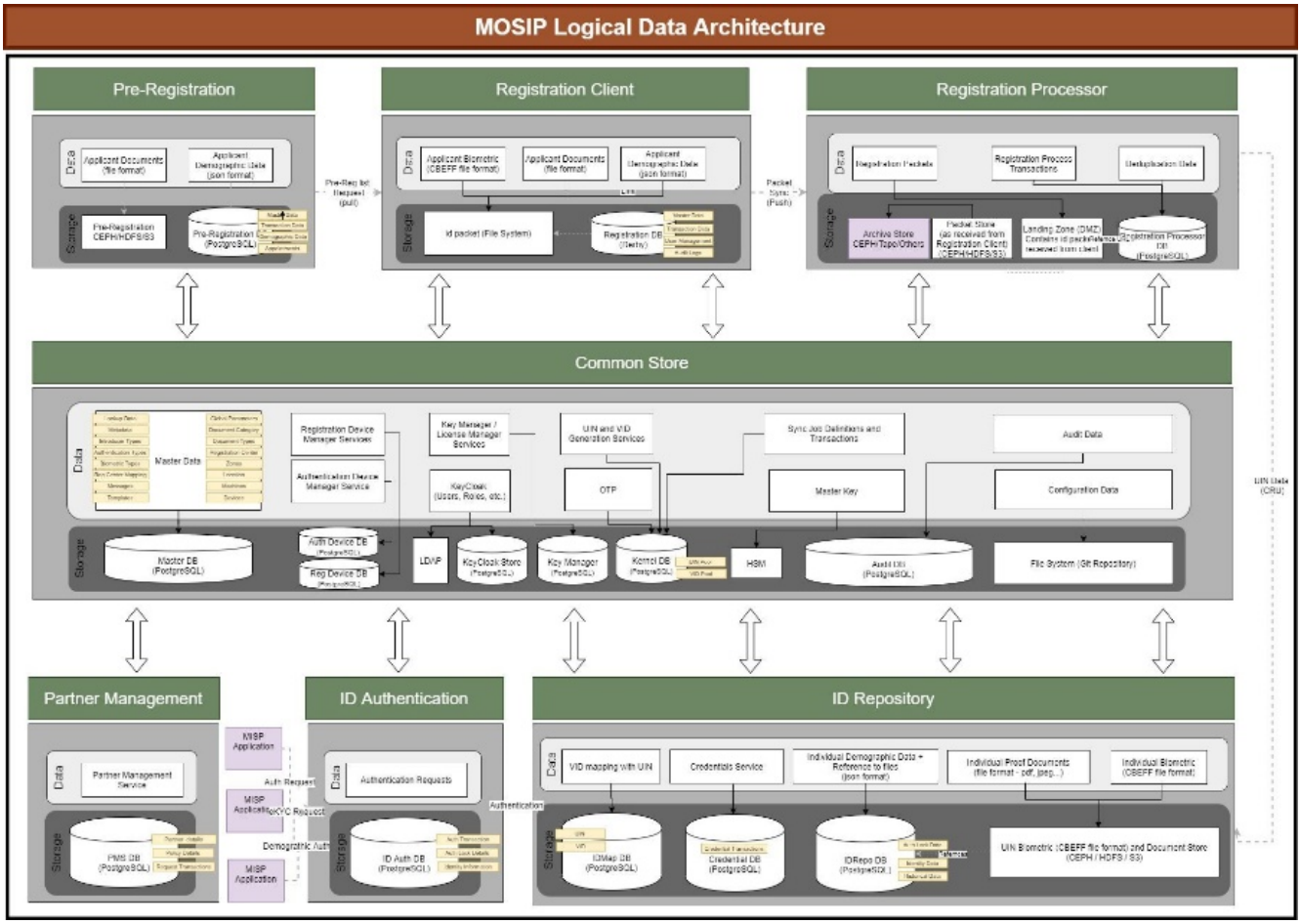
- マスターデータ - これには検索、ロケーション、センター、デバイス、ゾーンなどが含まれる。これは機密情報ではない
- 構成マネージャーデータ - これにはシステム構成情報が含まれており、保護する必要がある。個人情報は含まれていない
- 登録クライアントデータベース - これは登録クライアントのローカルデータベースである。関連するマスターデータのローカルコピー、ダウンロードされた事前登録情報、処理情報の一部が含まれる。事前登録情報は機密情報であり、暗号化されて保存される
- 登録パケット - 登録パケットは登録クライアントのメモリ内に作成され、暗号化されて以降は暗号形式のままとなる。これらのパケットは同期プロセスによってサーバーに移動し、処理される。パケットは信頼できる情報源 (source of truth) であり、デジタル署名されている
- 登録プロセッサーデータベース - 登録プロセッサーには処理するRIDの処理情報が含まれ、個人情報は含まれない
- IDリポジトリエントリ - IDリポジトリには個人識別データが含まれる。これには、経歴情報、人口動態情報、生体情報が含まれる。
  これらは実際はRDBMSとオブジェクトストレージに分散して保存されている
- 認証エントリ - 暗号化に使用したIDに片方向リンクされた、暗号化レコードが含まれる。情報はKYCレスポンスの一部として返却される際に復号できる。ここに保存されている生体情報は、非可逆的な抽出物のみである。KYC用に低解像度の写真を保存できる
- パートナー管理データ - パートナーのAPIキーや公開キーなど、保護された情報が含まれる。ここには個人情報はない
- 監査証跡 - これには個人情報は含まれない。追跡に使用するためのトランザクションIDが含まれる
- アプリケーション・ログ - これらのログには個人情報は含まれない
- 住民サービスデータ - これにはID番号付きの処理履歴が入っているが、個人情報は含まれない
- 事前登録データ - これは個人情報であり、暗号形式で保存される
- IAMデータ - MOSIPシステムユーザーリストがあり、これはシステムへのアクセス権を管理するため、保護された情報である

2. データ保存のガイドライン

- 氏名、年齢、性別、住所等の個人のPIIおよびその他の機密情報は署名され、暗号形式で保存されなければならない
- ドキュメントと画像はデータベーステーブルに保存してはならない。これらはオブジェクトストアに保存し、DBで参照する必要がある
- オブジェクトストアには、アクセス制御付きの暗号化されたデータのみが存在しなければならない
- データベースレベルでビジネスロジックは適用されず、主キー、ユニークキー、外部キー、Not-null のみが適用される。外部キーは同じデータベース内で適用される。テーブルが別のデータベースで参照される場合、外部キーは適用されない
- トリガのようなデータベース固有機能、シーケンスジェネレーターのようなDB関数等をMOSIPで使用してはならない。これにより、ベンダー囲い込みを防ぐ
- サロゲートキーを使用する場合は常に乱数でなければならず、レコードデータまたはシーケンス番号に基づいて生成してはならない
- データベースに対して手動で直接クエリを入力してはならない。データベース管理者はこのコントロールをデータベース構成設定時に必ず行うこと
- 複数言語をサポートするため、データベースはUTF-8* ファイル形式で設定する
- 次のデータタイプを使用する
  - 可変長文字 (Character Varying)
  - タイムスタンプ (Timestamp)
  - 日付 (Date)
  - 整数 (Integer)
  - 数値 (Number)
  - バイナリ (Bytea/blob)
  - ブール値 (Boolean)

2.2.2 データアーキテクチャの基本方針（2/3）

3. MOSIPデータシステムの論理ビュー



4. データのアクセス管理

MOSIPでは次のユーザーが定義され、さまざまなアクティビティを実行して、定義されたDBオブジェクトを管理する

- **sysadmin**: sysadminユーザーはスーパー管理者であり、データベース内のあらゆるタスクを実行するすべての権限を有する。現状、すべてのオブジェクトの所有者はsysadminである
- **dbadmin**: dbadminユーザーは、監視、パフォーマンス調整、バックアップ、リストア、複製の設定などすべてのデータベース管理アクティビティを扱うために作成される
- **appadmin**: appadminユーザーを使用してすべてのDDLタスクを実行する。これらのデータベースに作成されたすべてのDBオブジェクトの所有者はappadminユーザーである
- **アプリケーションユーザー**: 各モジュールにユーザーを作成してCRUDオペレーション等のDMLタスクを実行する。masteruser、prereguser、idauser、idrepouser、idmapuser、kerneluser、audituser、regprcuser、keymgruser、regdeviceuser、authdeviceuserが各モジュールのタスクを実行する

上記の役割セットで、アプリケーションユーザーのみがモジュール固有となる。その他のユーザーは共通であり、PostgreSQL DBインスタンスごとに作成する必要がある

2.2.2 データアーキテクチャの基本方針（3/3）

5. 複数言語対応

MOSIPプラットフォームは複数の国で構築され、複数の言語をサポートする必要がある。そのために、国レベルの管理者が設定する要件に従い、MOSIPは複数の言語をサポートする。複数言語をサポートする必要があるのは、次のデータセットである

- マスターデータ
- 個人のIDデータ
- トランザクションコメント
- UIで使用するラベル
- メッセージおよび通知

データベース側では、データは**UTF-8 Unicode文字セット**を使用して複数の言語で入力されたデータを保存する

データベースレベルでデータを翻訳する組み込みサポートは提供されない。翻訳または文字転写はAPIまたはUIレイヤーで処理される。
ユーザーインターフェースやコミュニケーションテンプレートで複数言語のニーズに対応するための国際化対応が可能である。マスターデータについては情報を複数の言語で保存できる。ユーザーデータについてはMOSIPでは2カ国語でのデータ保存をサポートしている

6. パフォーマンス

パフォーマンスをサポートするために、次のデータベース設計機能を検討すること

- データベースのシャーディング：　デフォルトで、シャーディングアルゴリズムはMOSIPシステムに適用されない。SIはデプロイ設定に基づいてシャーディングアルゴリズムを定義できる
- すべてのテーブルについて主キーのフィールドに主キーのインデックスが作成される。これにより、取得や結合がより高速になる
- すべての外部キーにはインデックスが定義されるので、結合が高速化される
- データベース全体でテーブルに参照の一貫性は適用されない
- パーティショニング：　パーティショニング設計はデータベース固有である。国は選択されたデータベースに応じて、データベースエンジンによって規定されたパーティショニングアプローチを採用できる
- オブジェクトストアのデータは、階層的なパス変換により簡単にアドレッシングできなければならない

7. データモデルの検討ポイント

- **意味のある命名体系**:　作成するDBオブジェクトには意味のある名前を付ける
- **柔軟モデル**:　いくつかのマッピングデータを除いてデータベースレベルでビジネスルールを設定しないこと。ビジネスロジックのほとんどはアプリケーションレイヤーで適用される
- **データベース固有の機能**:　デフォルト、DBシーケンス、IDフィールドなどDB固有の機能は使用しない
- **DBでのビジネスロジック禁止**:　主キー、ユニークキー、外部キーを除き、データベースレベルでビジネスロジックは実装しない
- **データのセキュリティ**:　個人情報とセキュリティに関する情報は暗号化される

2.3.1 Pre-Registration（1/2）

概要

このモジュールでは、住民について以下の処理を行うことができる

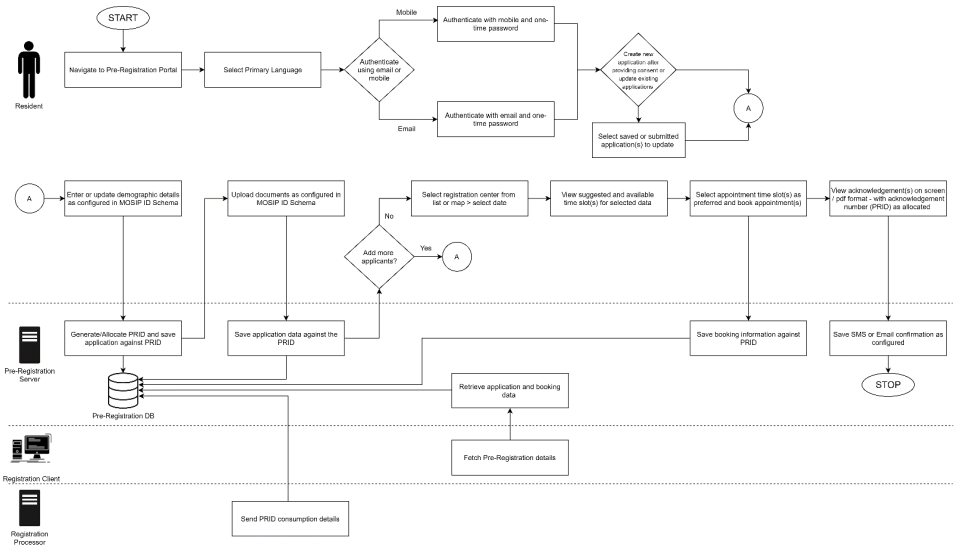
- 人口動態データの入力と添付書類のアップロード
- 1人または複数のユーザーについて適切な登録センターと時間を選んで登録のための来場予約をする
- 予約確認通知を受け取る
- 予約の日時変更やキャンセルを行う
- 予約された日の前に、登録プロセスで使用する住民データを指定された登録センターに送付する

1. 詳細機能

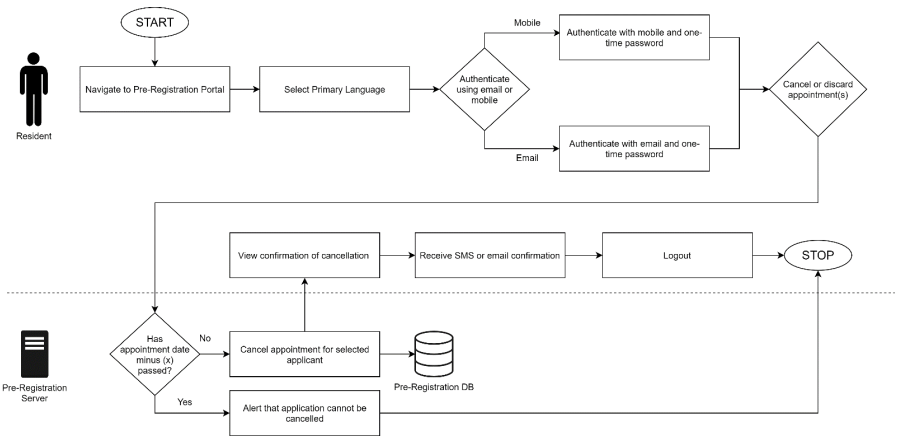
- 事前登録機能の詳細については、事前登録機能のページを参照

2. プロセスフロー

事前登録フローを作成および更新する際のプロセスフロー図。



事前登録フローをキャンセルおよび破棄する際のプロセスフロー図。



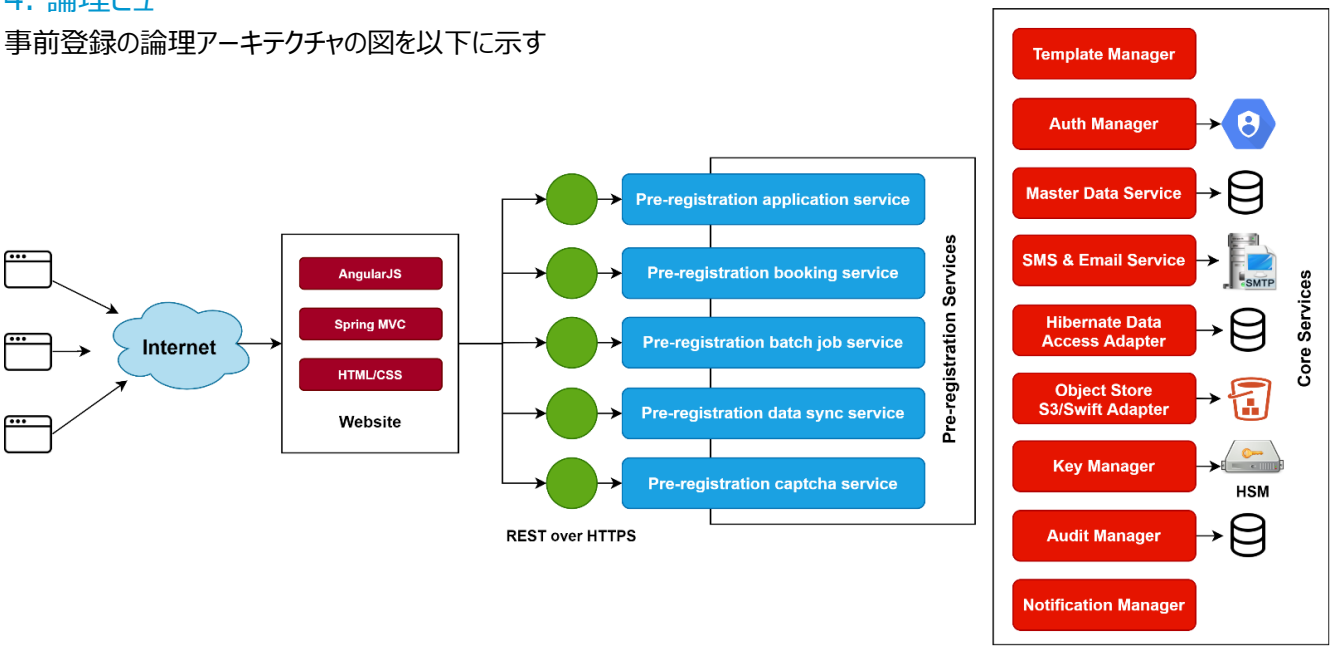
3. サービス

事前登録サービスの詳細については、事前登録リポジトリを参照
ハイレベルおよびローレベル設計については、事前登録リポジトリ/デザインを参照

2.3.1 Pre-Registration（2/2）

4. 論理ビュー

事前登録の論理アーキテクチャの図を以下に示す



5. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、事前登録リポジトリを参照

6. API

事前登録API機能の詳細については、事前登録APIのページを参照

7. UIリファレンス実装

MOSIPには国のニーズに合わせてカスタマイズできる事前登録UIのリファレンス実装が用意されている。
コードはリファレンス実装リポジトリで入手できる

2.3.2 Registration（1/5）

概要

登録クライアントはJavaベースのシッククライアントであり、住民の詳細な人口動態的情報と生体情報、添付書類をオンラインまたはオフラインモードで取得する。取得した情報は安全で改ざんできない方法でパッケージ化され、サーバーに送って処理される。

登録クライアントは次の機能を提供しなければならない

- **安全な**方法による個人の人口動態データおよび生体データの取得。取得されたデータは、改ざんできないように暗号化して安全な状態にしなければならない。これは登録パケットと呼ばれる
- 業界規格に準拠した**生体認証デバイスへのインターフェース**。これにより、規格どおりに製造されたどのデバイスも必ずMOSIPで動作するようになる
- **オンラインおよびオフライン**モードでの動作。インターネットにつながりにくい遠隔地では、登録クライアントはオフラインモードで動作しなければならない
- **遠隔ソフトウェア更新**機能。リモートでの（バグ修正/機能拡張のための）最新パッチ/アップグレードに対応した自動更新機能は必須である。数百のクライアントインスタンスがラップトップ/デスクトップPCで実行されている場合がある。これらすべての更新は中央サーバーで管理する必要がある
- **改ざん不可能なクライアントソフトウェア**。手動での改ざんで生じる可能性のある異常を検出し、キャプチャ情報を含むパケットを拒否できるよう、登録クライアントにはキャプチャされた情報の構造を検証する機能がなければならない

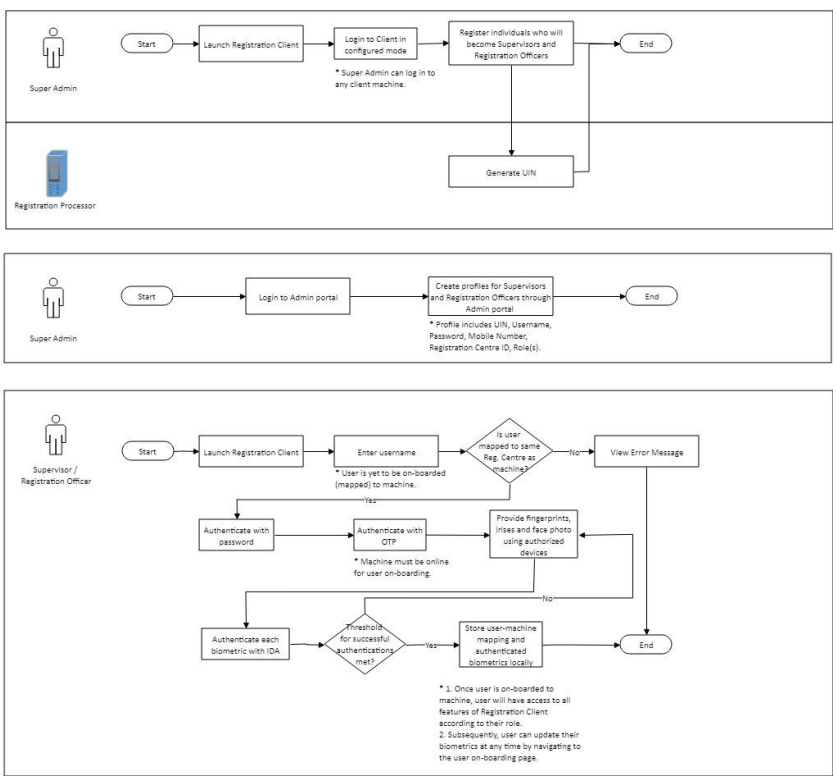
1. 詳細機能

登録機能

2. プロセスフロー

2.1 登録担当者の業務開始

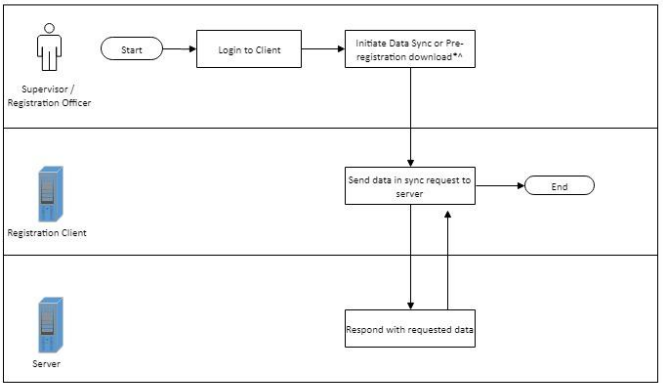
Registration Officer On-boarding



2.3.2 Registration（2/5）

2.2 登録準備

Registration Preparation



*** Data to Sync:**

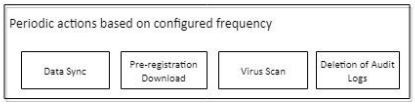
Server to Client

1. Master data
2. Config settings
3. Registration Centre details
4. User details
5. Login credentials
6. Machine details
7. Device details
8. Registration Packet status

Client to Server

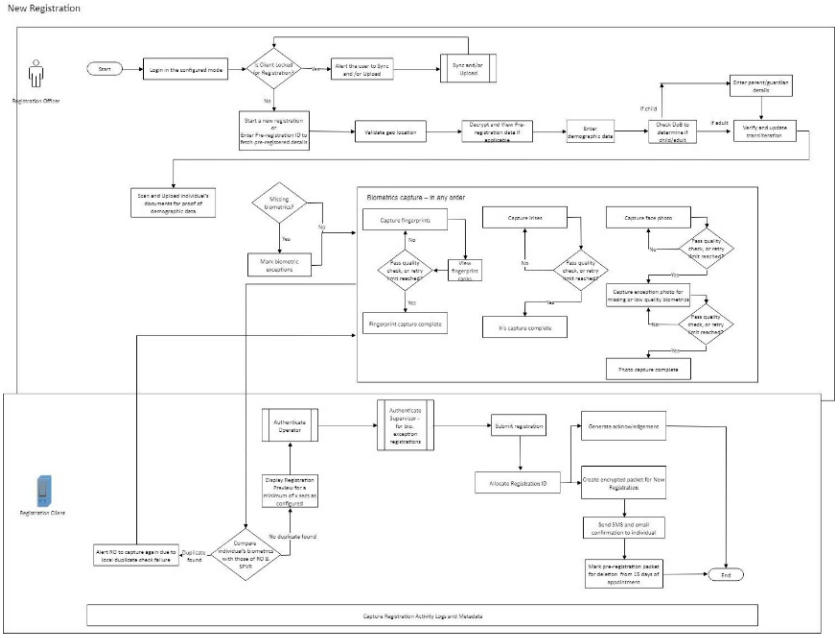
9. User-Machine mapping
10. Registration Packet IDs

^ Machine must be online for sync and download



2.3 個人の登録

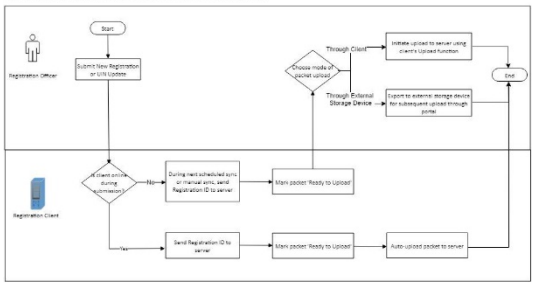
New Registration



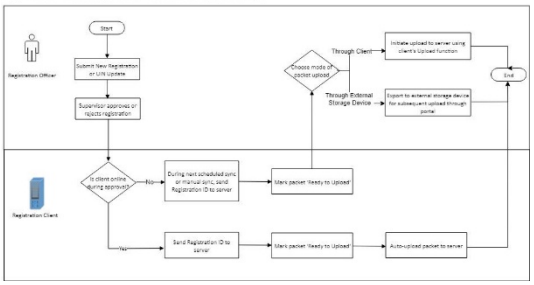
2.3.2 Registration（3/5）

2.4 パケットのアップロード

Registration - Packet Upload - when EoD Process is turned off

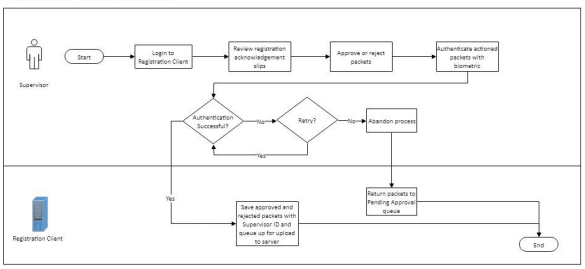


Registration - Packet Upload - when EoD Process is turned on

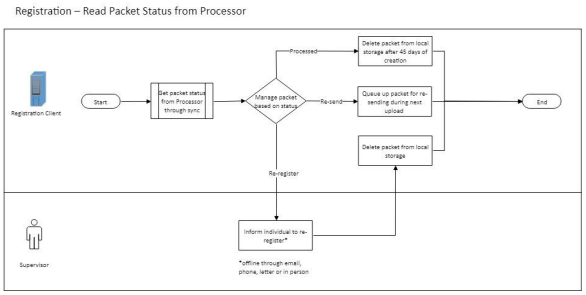


2.5 終業時 (EOD) プロセス

Registration – End of Day Process

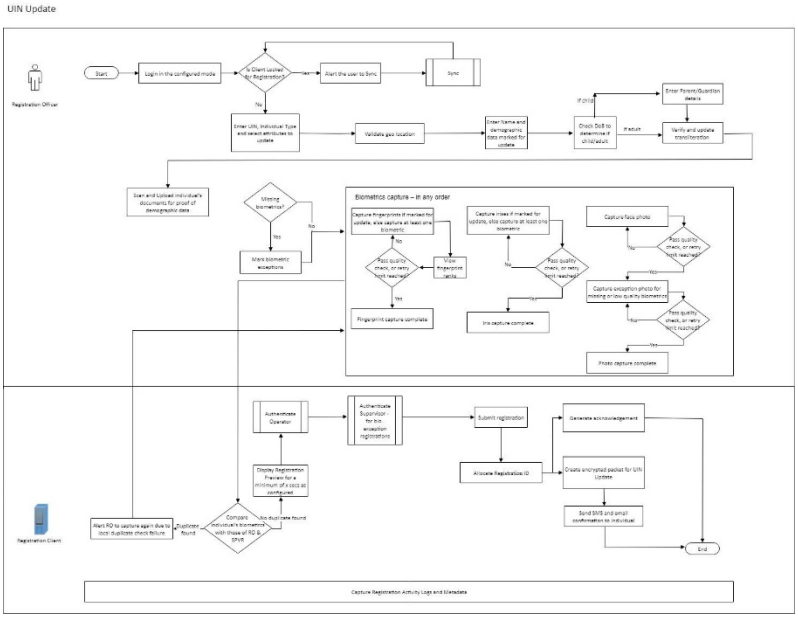


Registration – Read Packet Status from Processor



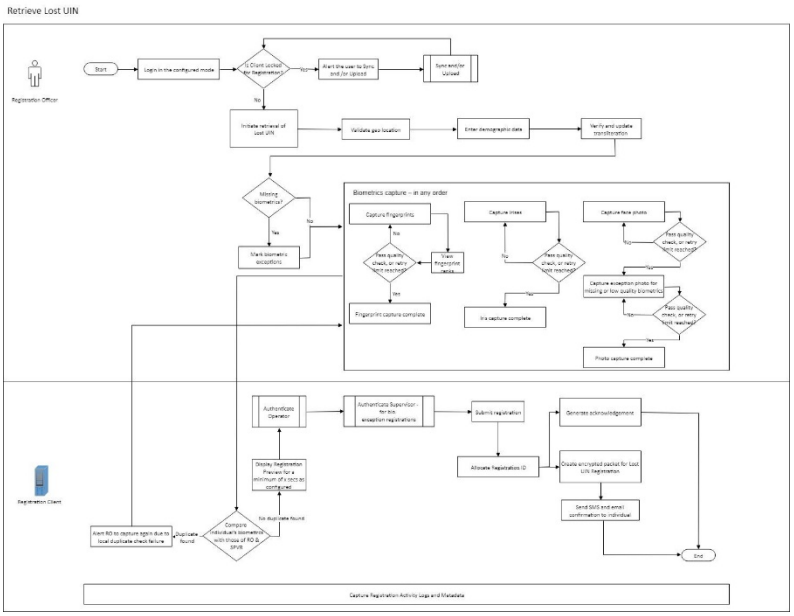
2.6 UIN更新

UIN Update



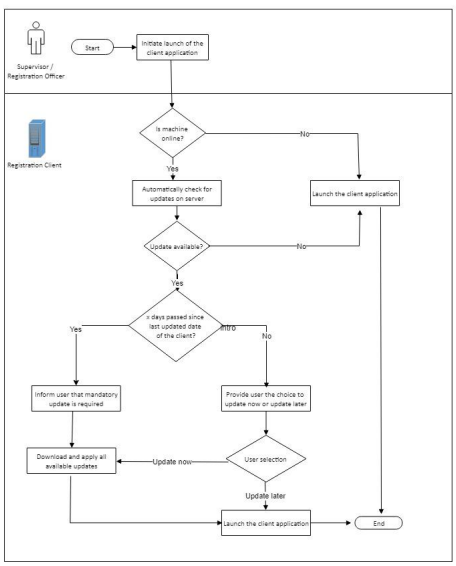
2.3.2 Registration（4/5）

2.7 UINの損失

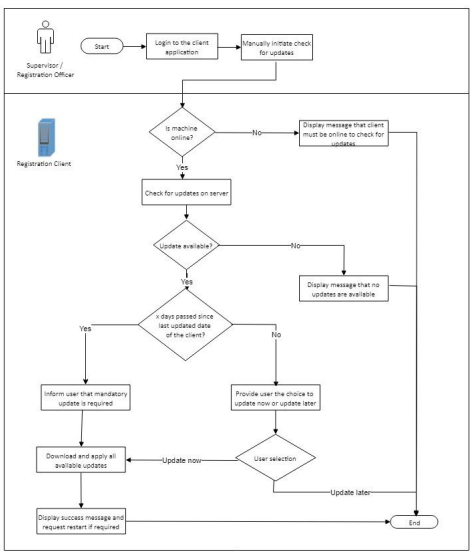


2.8 ソフトウェア送信

Registration Client Software Update - update on launch

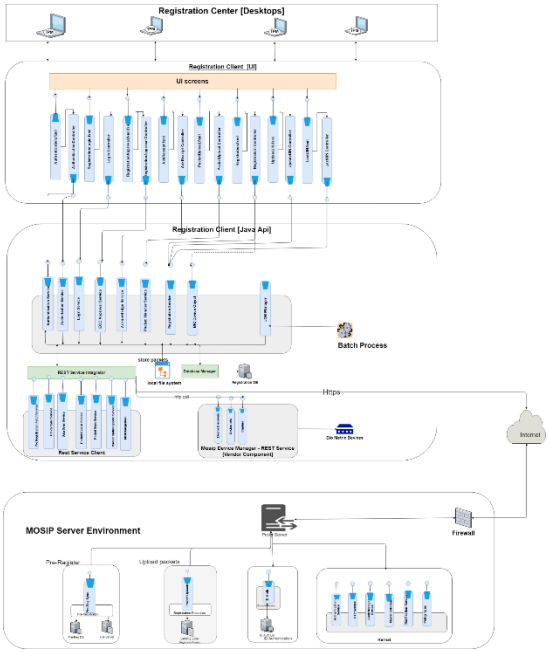


Registration Client Software Update - manually update from application menu

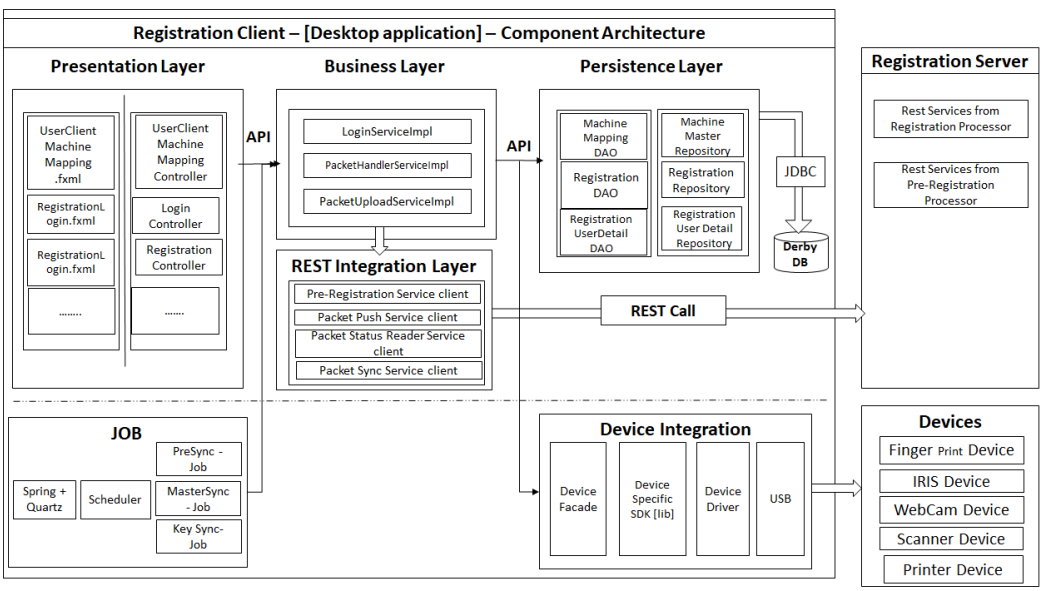


2.3.2 Registration（5/5）

2.9 論理ビュー



2.10 コンポーネントアーキテクチャ



3. 登録パケット構造

すべての登録情報はパケットでZip圧縮され、暗号化されてサーバーに送信される。パケットの構造については、こちら

4. 登録クライアントのリファレンスアプリケーション

MOSIPではWindowsベースのクライアントのリファレンス実装を提供しており、上記のプロセスフローを実行するUIとビジネスロジックが含まれている。コード、デザイン、アプリの設定、ビルドマニュアルは、登録クライアントリポジトリにある。アプリは導入国のニーズに応じて変更できる

登録クライアント設定ガイド

2.3.3 Registration Processor（1/4）

概要

登録プロセッサーは個人のデータ (人口動態データと生体情報データ) の品質と一意性について処理を行い、ユニークID番号 (UIN) を発行する。UINが損失した場合に人口動態データと生体情報データを更新して新しいUINを発行する機能も提供する。データの情報源は主に次の場所である

- MOSIP登録クライアント
- 導入国の既存のIDシステム

重要な検討事項は以下のとおりである

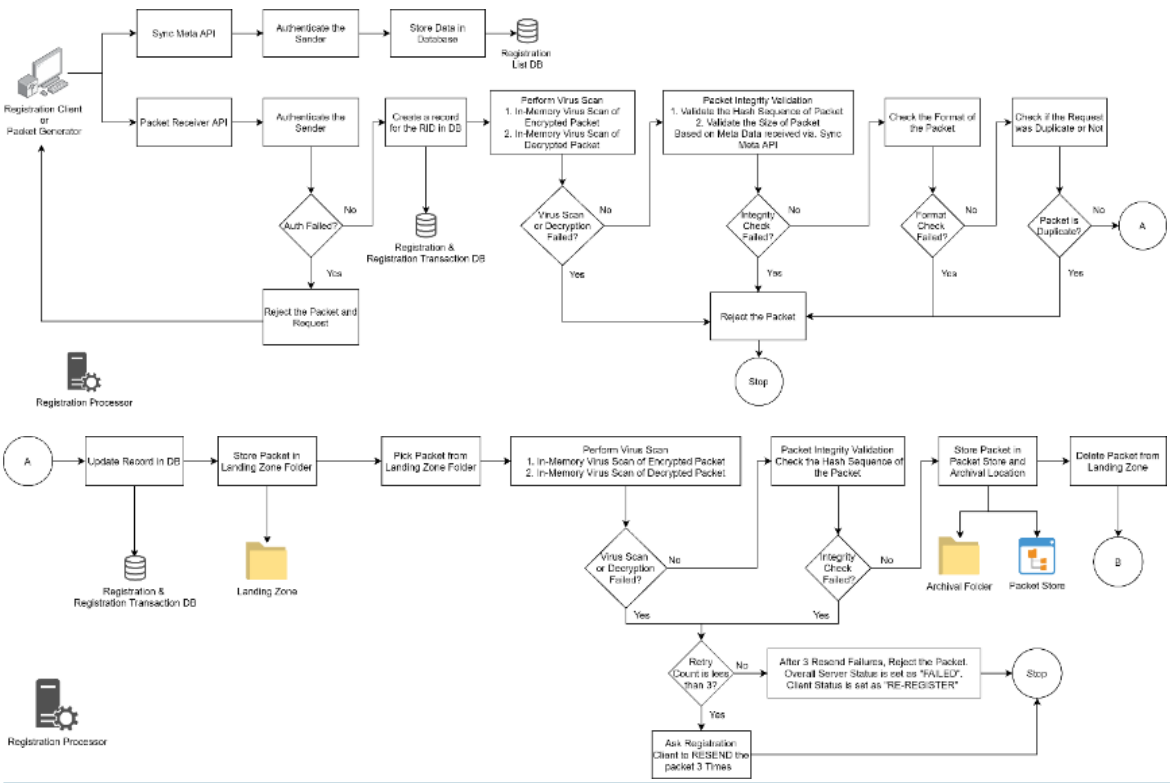
- パケットをサーバーで受信した後は、パケットの紛失がないこと
- MOSIPは基本的な登録パケット処理フローを定義し、実装する。ただし、各国には既存のIDシステムとの統合や検証用データの取込みなど、独自の処理要件がある。登録プロセッサーはそのようなステージを付加するオプションを提供する
- 登録プロセッサーは複数のABISプロバイダーを統合する機能を有する
- 負荷に応じて各処理ステージは独立にスケールできる
- プロセッサーの各ステージはその他のステージから独立しているので、全体のフローに影響を与えることなくステージの論理を変更して効率を向上させることができる

1. 詳細機能

登録プロセッサーの機能

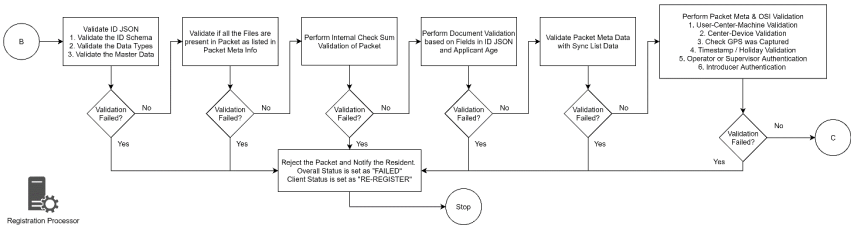
2. プロセスフロー

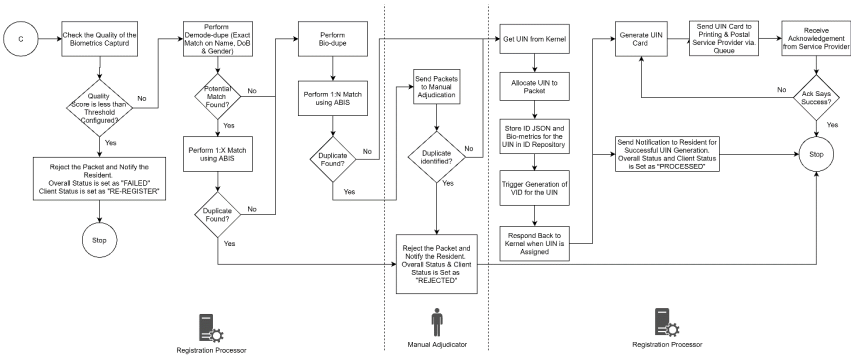
2.1 パケットの事前処理



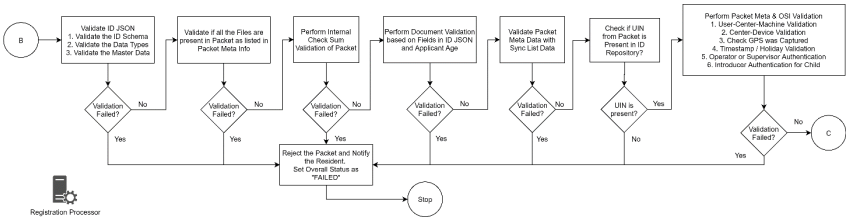
2.3.3 Registration Processor（2/4）

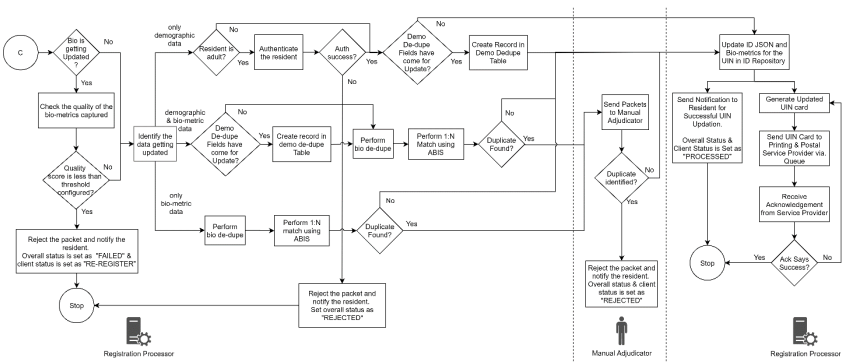
2.2 新規パケット処理





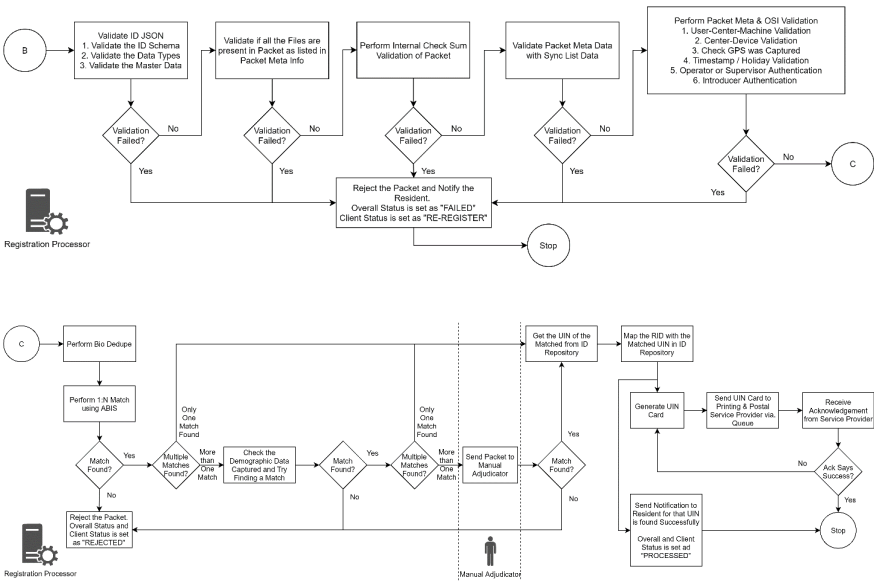
2.3 更新パケット処理



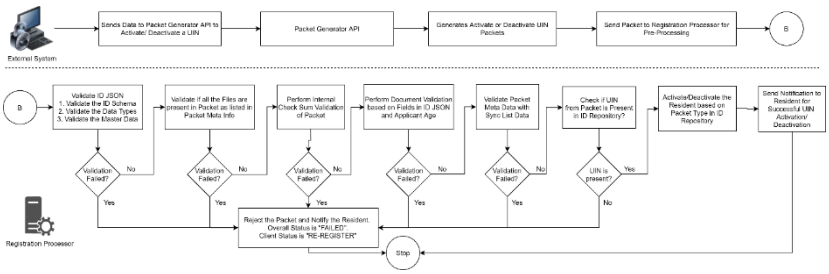


2.3.3 Registration Processor（3/4）

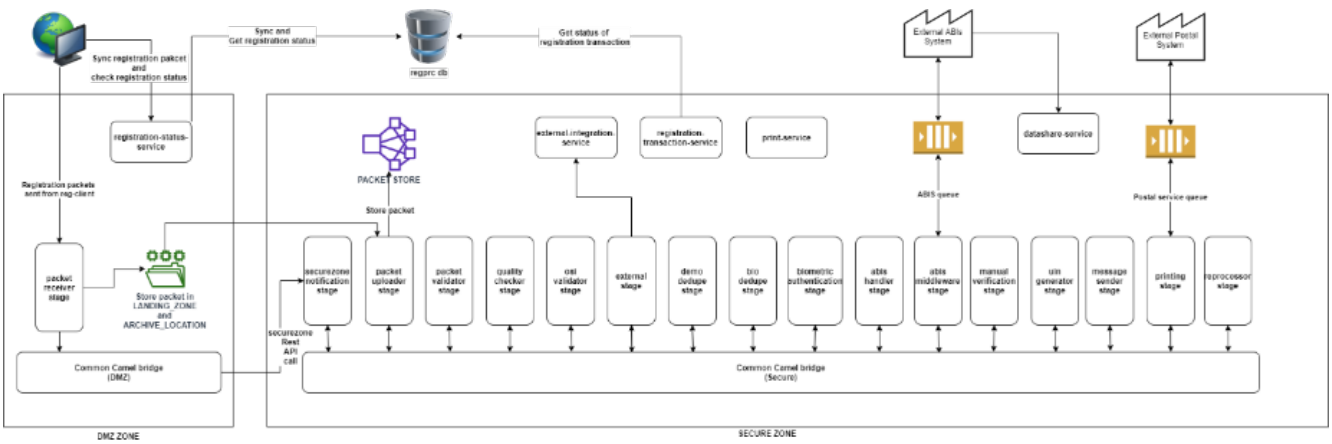
2.4 損失UINパケット処理



2.5 パケット処理の有効化/無効化



3. 論理ビュー



2.3.3 Registration Processor（4/4）

4. サービス

登録プロセッサーサービスの詳細については、事前登録リポジトリを参照
ハイレベルおよびローレベル設計については、登録プロセッサーリポジトリを参照

5. ビルドおよびデプロイ

登録プロセッサーリポジトリのビルドおよびデプロイ手順を参照

6. API

登録プロセッサーAPI

2.3.4 ID Authentication（1/5）

概要

ID認証 (ID Auth) は個人の身元を確認するためのAPIベースの認証メカニズムを提供する。ID認証は、サービスを提供する前に個人の身元を確認する最初の手続きである

以下に個人の認証を行うための前提条件を示す

- ID認証のリクエストは、MOSIPのホワイトリストに登録されている信頼されたパーティーを経由してのみMOSIPに送信される信頼されたパーティーは、MOSIPでパートナーと呼ばれる
- 認証に使用する生体認証デバイスはMOSIPに登録されていなければならない
- ID認証ではパートナーのみが認証リクエストを行うことができる。リクエストは暗号化され、保護され、検証済である。生体認証デバイスからデータを取得するパートナーは、規格に準拠し、相互運用性を確保しなければならない

個人は以下に基づいて認証される

- 人口動態データ
  - 氏名
  - 生年月日
  - 性別
  - 住所
- 生体認証情報
  - 指紋
  - 虹彩
  - 顔

安全性を高めるため、以下の第2要素がサポートされている

- OTPベース
- 静的PINベース
- チャレンジレスポンス認証

認証パターンを分析および生成するために、すべての認証リクエストが監査される。これらの監査ログは認証プロセスの中で不正があったかどうかを判断するために使用される

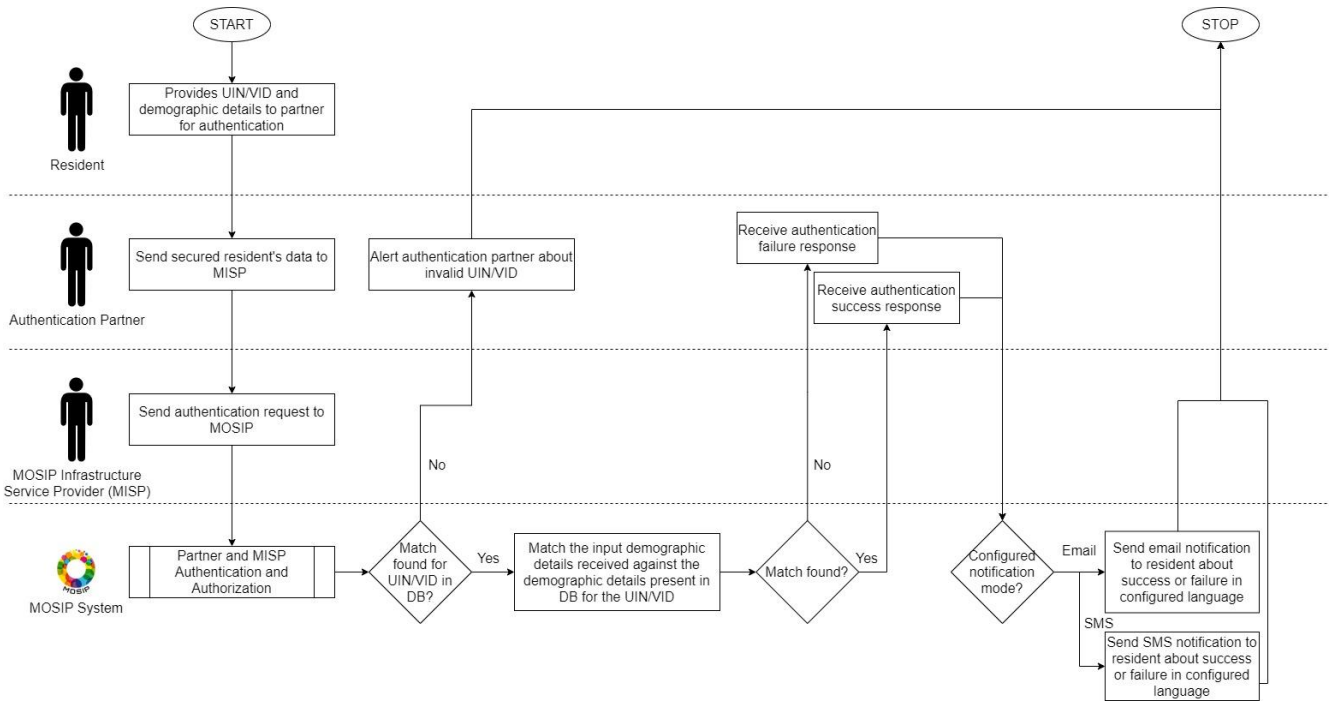
2.3.4 ID Authentication（2/5）

1. 詳細機能

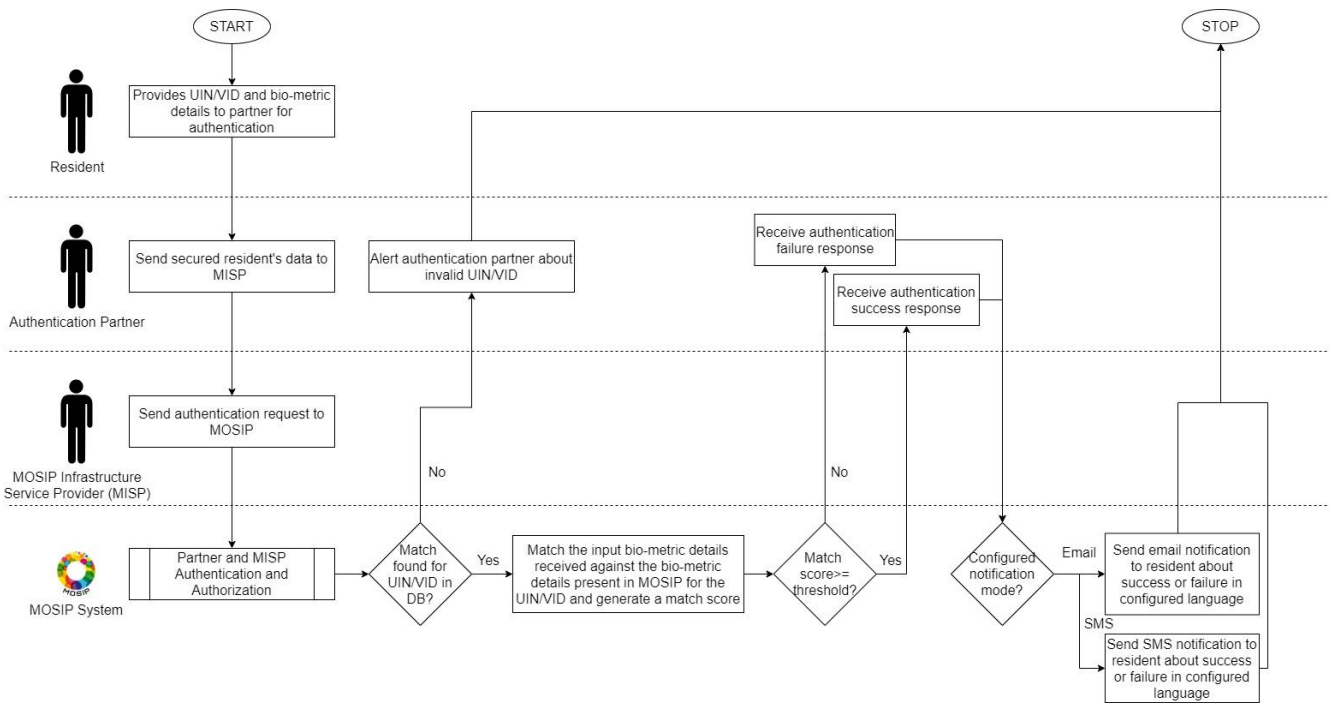
ID認証機能

2. プロセスフロー

2.1 人口動態データの認証

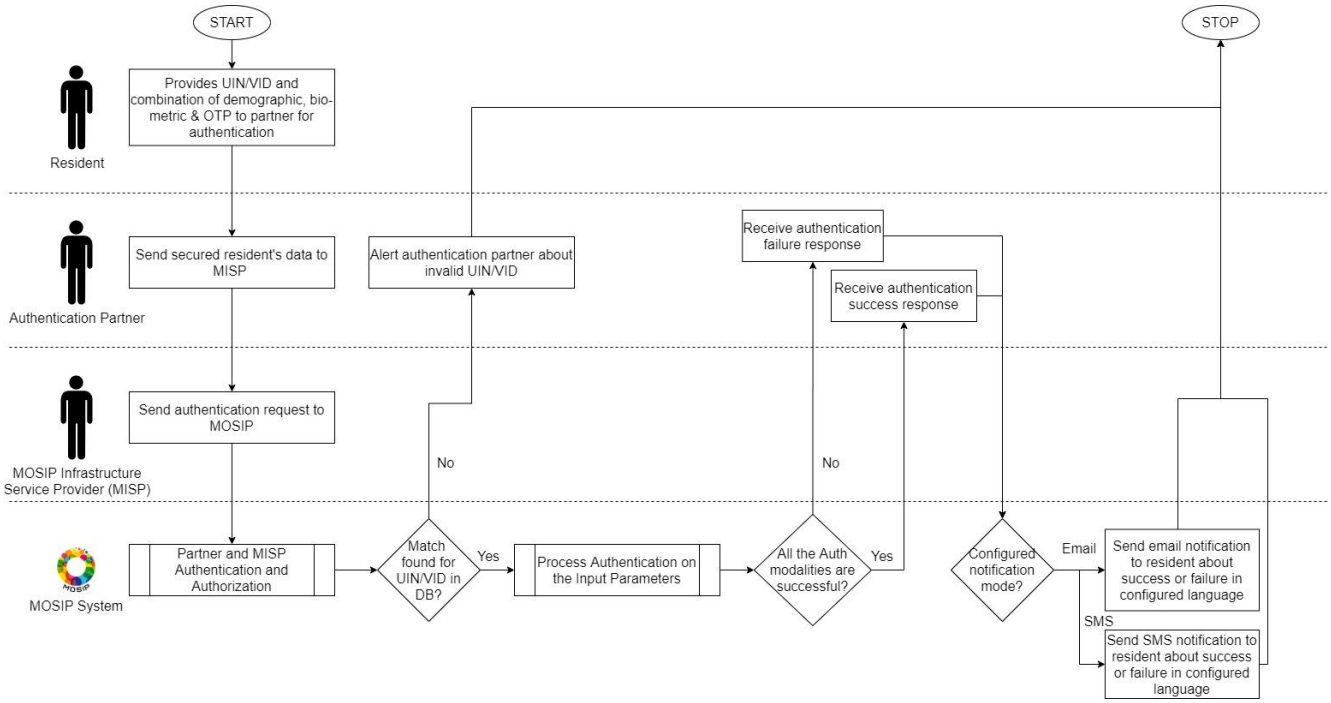


2.2 生体認証

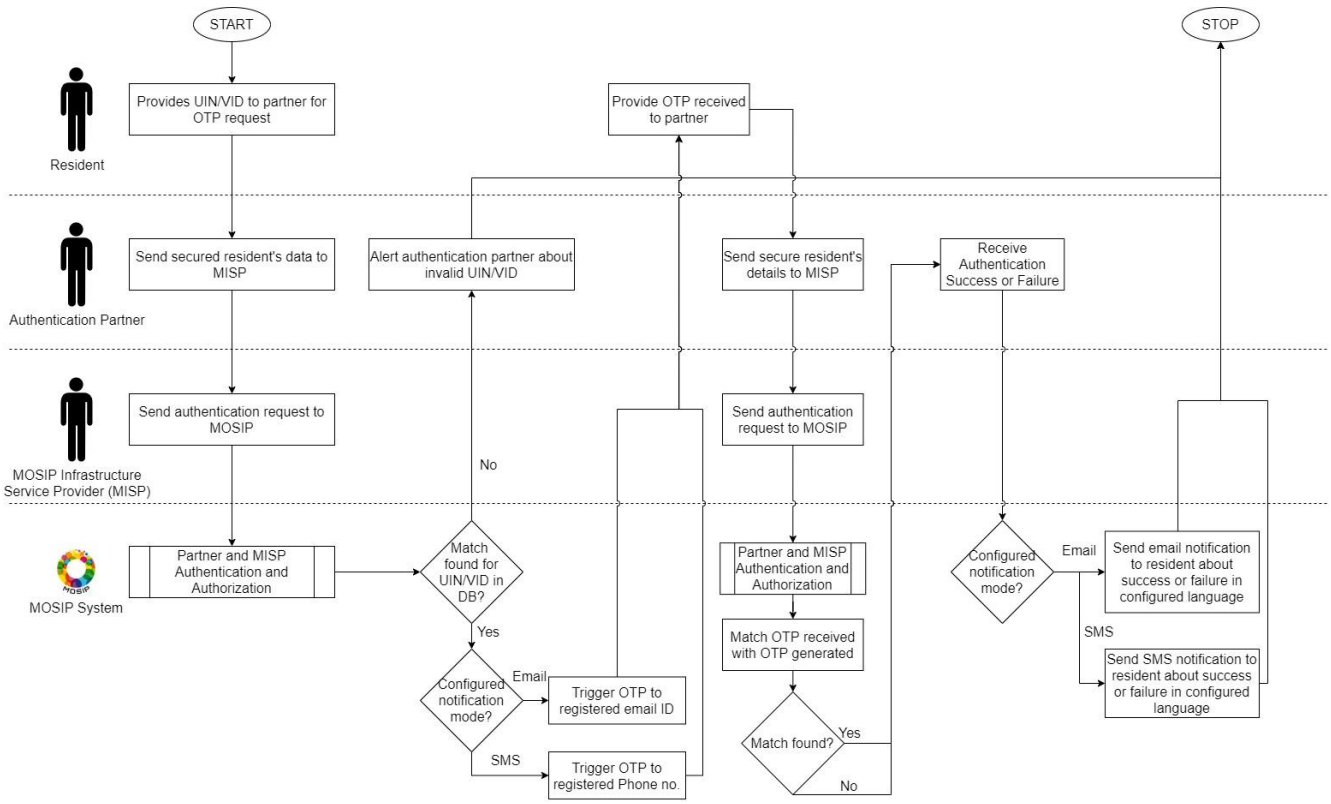


2.3.4 ID Authentication（3/5）

2.3 多要素認証

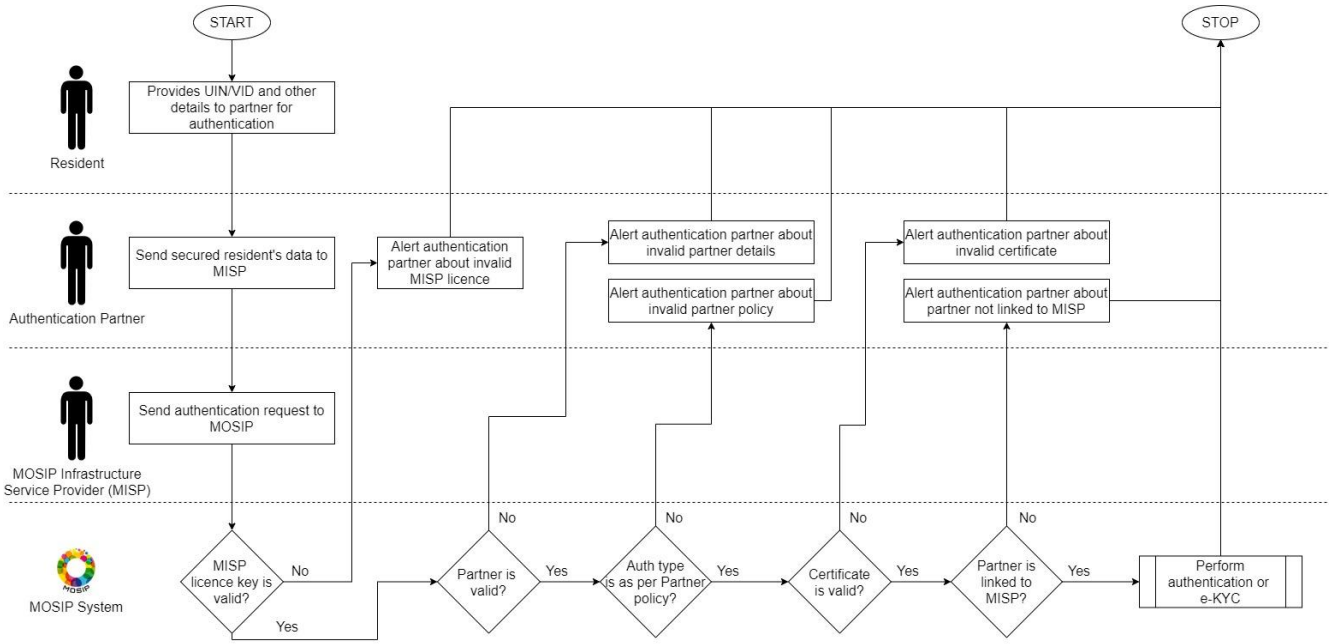


2.4 OTP認証

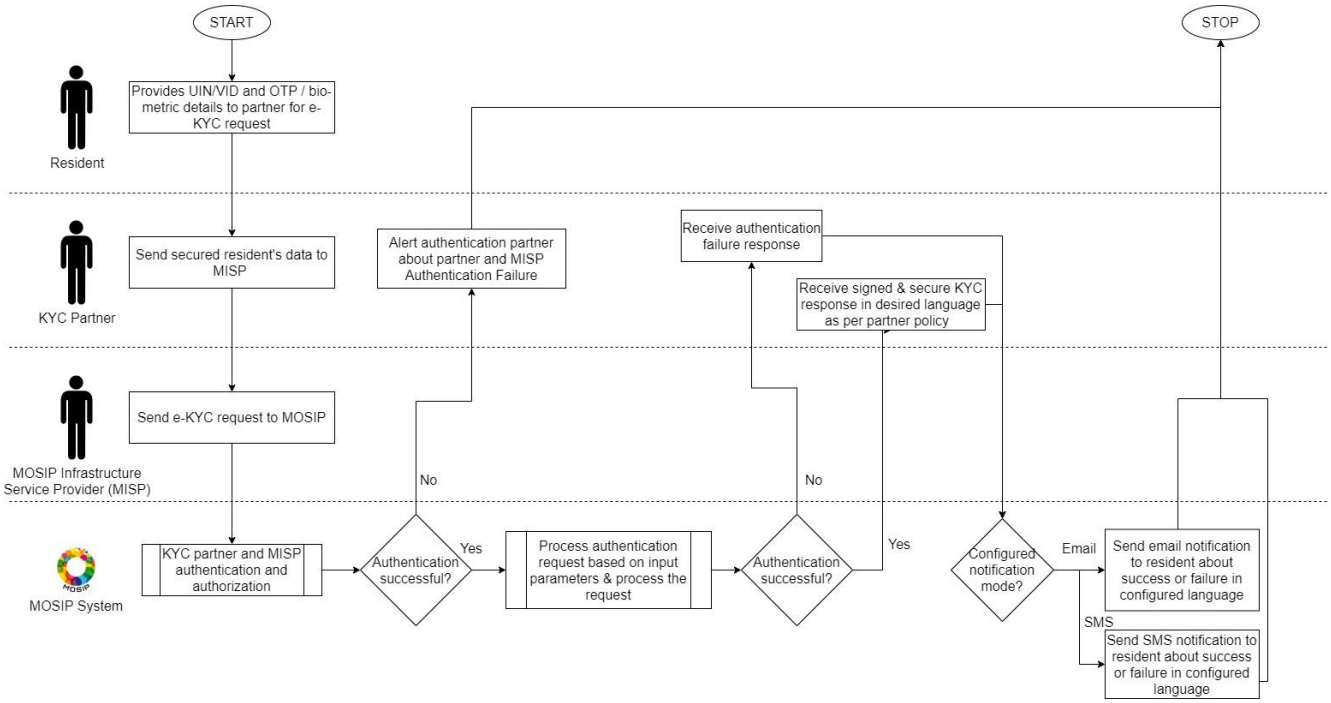


2.3.4 ID Authentication（4/5）

2.5 パートナー認証およびMISP認証

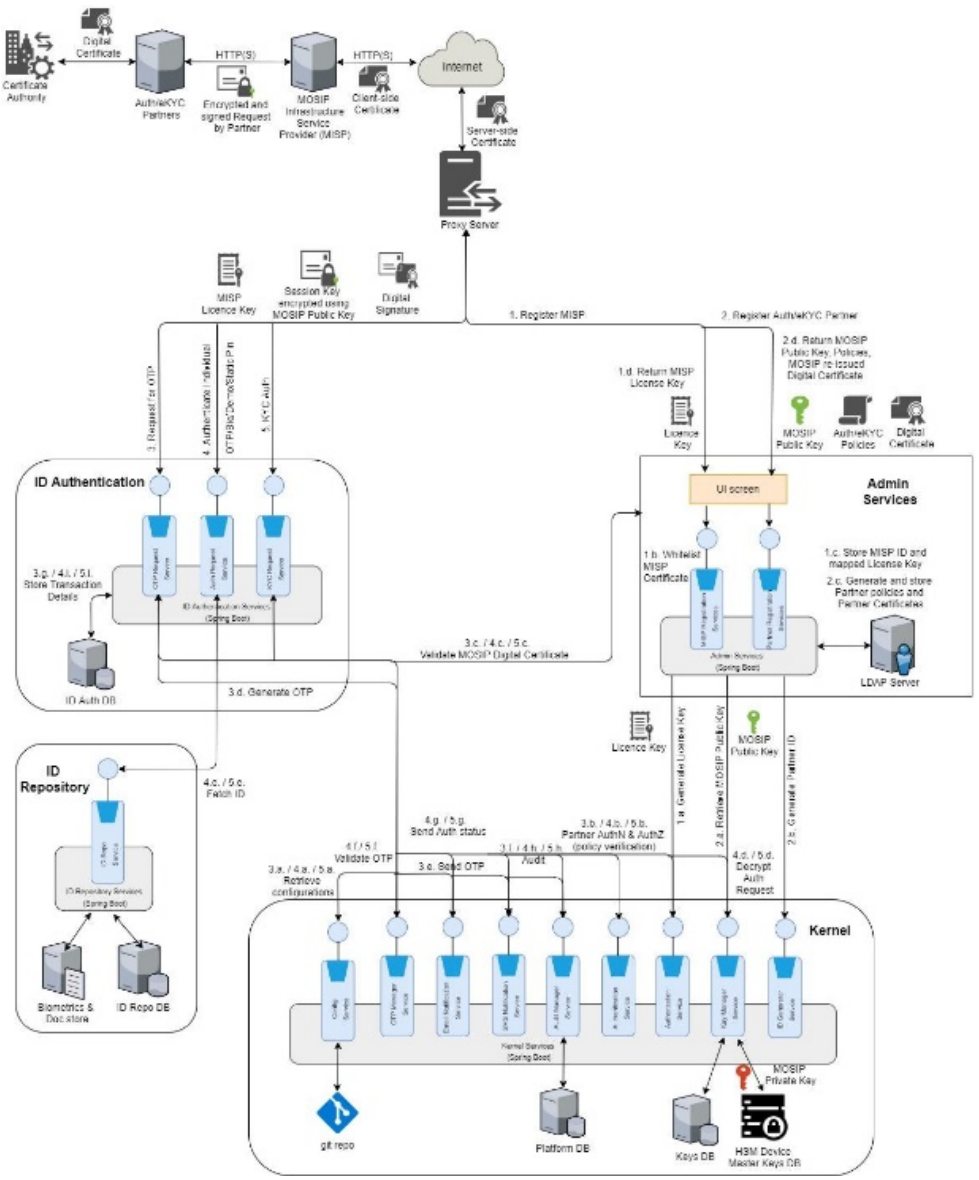


2.6 eKYC認証



2.3.4 ID Authentication（5/5）

3. 論理ビュー



4. サービス

ID認証サービス、コード、デザインの詳細については、ID認証リポジトリを参照

5. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、ID認証リポジトリを参照

6. API

ID認証API

2.3.5 Administration（1/2）

概要

MOSIPプラットフォームは管理アプリケーションを通じて構成される。このアプリケーションには、管理者の特権グループだけがアクセスできる。MOSIPプラットフォームが初期化されると、デフォルトの構成とシードデータが設定される。インストールの後、管理アプリケーションで次の操作を行うことができる

- エントリ変更の設定
- マスターデータ管理
- ユーザー管理
- マスターデータとさまざまなリソースとのマッピング

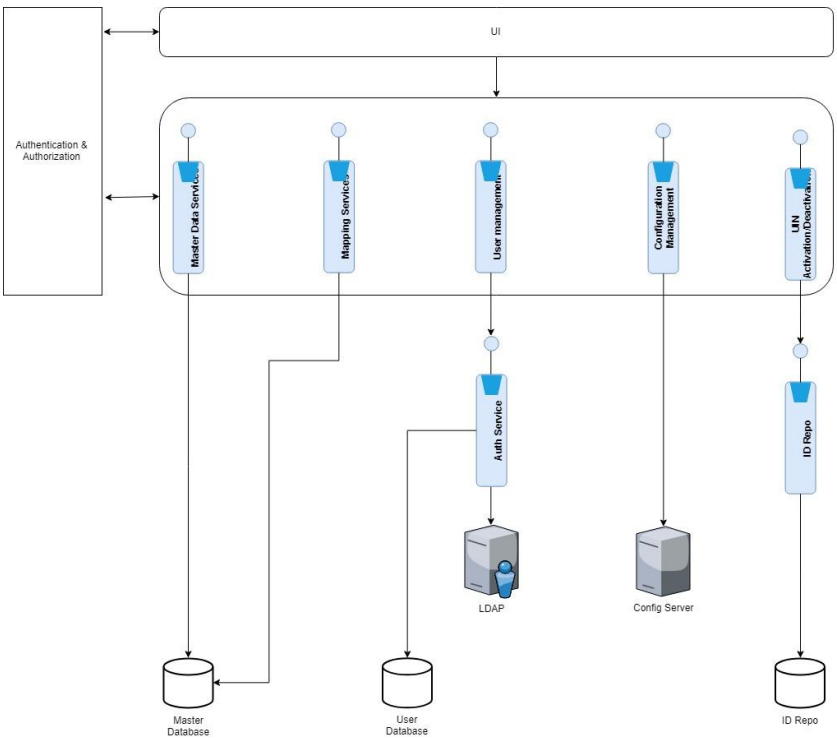
モジュールはMOSIPプラットフォームを管理するためのシングル・ユーザー・インターフェースを提供する。プラットフォームを最初にインストールする際に、データと構成をCSVファイルでアップロードできる

管理アプリケーションは、UIレイヤーとサービスレイヤーで構成される。サービスおよびUIの両方にあるすべてのコンポーネントは、安全かつ認証済である。全コンポーネントは認証モジュールのプラグインで定義する必要がある。たとえば、コンポーネントのデータが認証を受けた者しか表示できないようになっていれば、一般ユーザーは誰も表示することができない。作成、編集、削除の各機能も同様である

1. 詳細機能

管理サービス機能

2. 論理ビュー



2.3.5 Administration（2/2）

3. バックエンドサービス

管理者は、共通リポジトリにあるカーネルの一部として多くのサービスを使用する。管理リポジトリには管理者専用のサービスがある。コードと設計ドキュメントはリポジトリにある

4. フロントエンド - 管理ポータル

管理ポータルのリファレンス実装は、リファレンス実装リポジトリにある

5. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、上記リポジトリを参照

6. API

管理モジュールでは以下のような複数のサービスからのAPIが使用される

- 管理API
- ドキュメントAPI
- 登録センターAPI
- デバイスAPI
- マシンAPI
- 共通API
- ゾーンAPI
- デバイス管理API

2.3.6 Resident Services

概要

住民サービスとは、住民自身がポータルを介して利用するセルフサービスである。認証タイプのロック/ロック解除、UINの再発行、認証履歴の閲覧などの機能が利用できる。OTP方式で認証される。

バックエンドはREST APIインターフェース (MOSIPが提供) を備えたサービス群であり、フロントエンドは導入国の要件に応じて開発されるポータルである

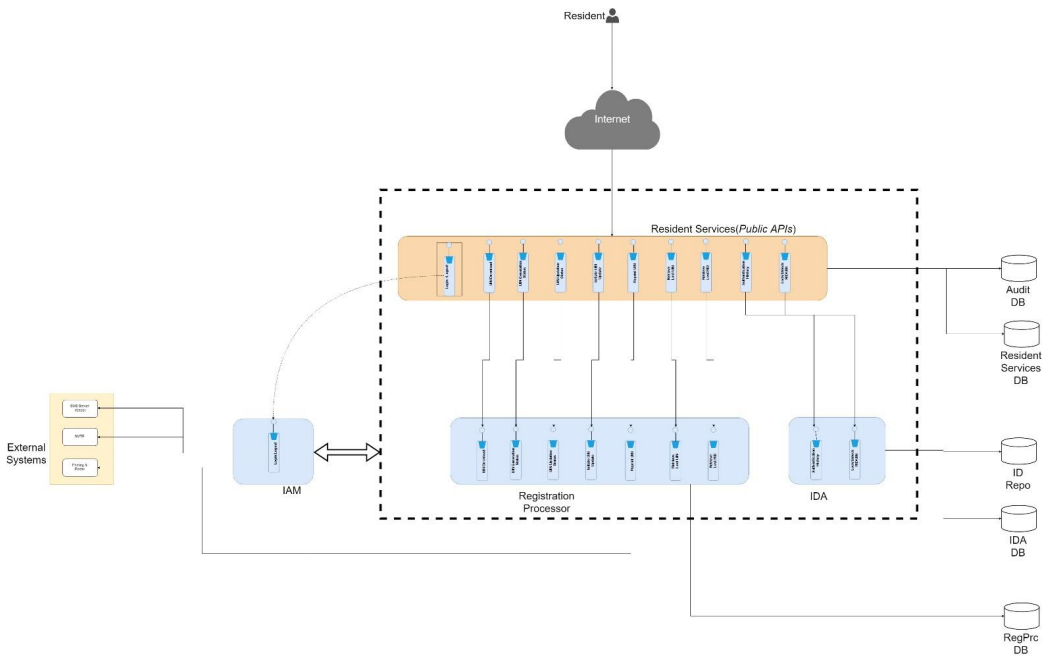
1. 詳細機能

住民サービス機能

2. プロセスフロー

- UINのロック
- UINのロック解除
- UINの更新
- UIN更新の追跡
- RIDによるUIN追跡
- 損失UINの回復
- VIDの生成
- VIDの取り消し
- e-UINのダウンロード
- e-UINの再プリント
- 損失RIDの回復

3. 論理ビュー



4. サービス

住民サービス、コード、デザインの詳細については、住民サービスリポジトリを参照

5. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、住民サービスリポジトリを参照

6. API

住民サービスAPI

2.3.7 Partner Management（1/6）

1. はじめに

パートナー管理は、MOSIPシステムに関連付けられるさまざまなタイプのパートナーにサービスを提供する。現在MOSIPではいくつかのタイプのパートナーが特定されているが、導入国はより多くのパートナーを追加する選択ができる。

1. MOSIPシステムに登録している個人に認証サービスを提供する認証パートナー
2. 安全なチャネルを通じて認証リクエストを送信するためのインフラストラクチャを提供する、MOSIPインフラストラクチャ・サービス・プロバイダー (MISP)
3. MOSIPに準拠した認証および登録用のデバイスを提供するデバイスプロバイダー。
4. SBI 2.0デバイスのチップを提供するファンデーショナル・トラスト・プロバイダー。
5. 住民用IDカードを生成する証明書/印刷パートナー。
6. 重複排除処理を実行するAutomated Biometric Integration System(ABIS)。

その他のパートナーも存在する。

登録パートナーは、MOSIPパートナー管理によって提供された役割に基づいたMOSIPサービスにのみ、アクセスが許可される。これらのパートナーは、MOSIPのパートナー管理ポータルを通じて自分で登録する必要がある。その後、パートナー管理が詳細情報を確認し、MOSIPサービスへのアクセスを提供する。
パートナー向けMOSIPサービスは、パートナーの資格情報がMOSIPに登録され、サービスによって検証された場合にのみ有効である。

パートナー管理にはパートナーに対するポリシー管理も含まれる。各パートナーは、定義されたポリシーに基づいてのみさまざまなサービスにアクセスできる。

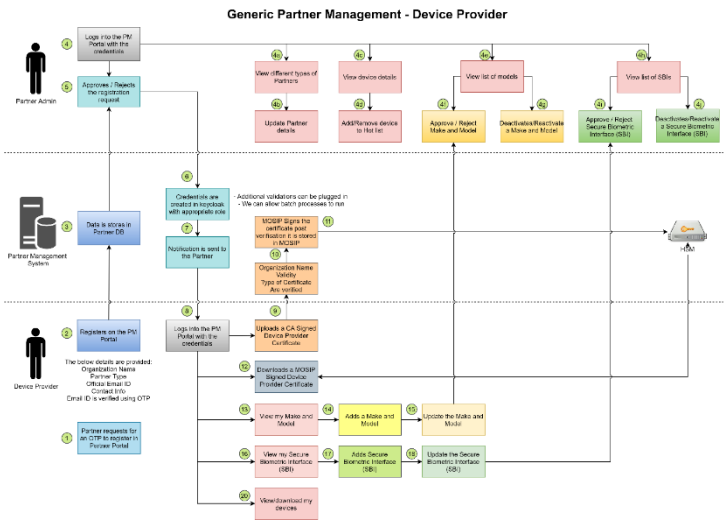
パートナーのタイプに応じて、MOSIPは各種サービスをそれぞれのパートナーに提供する。

2. 詳細機能

パートナー管理機能の詳細については、パートナー管理機能のページを参照

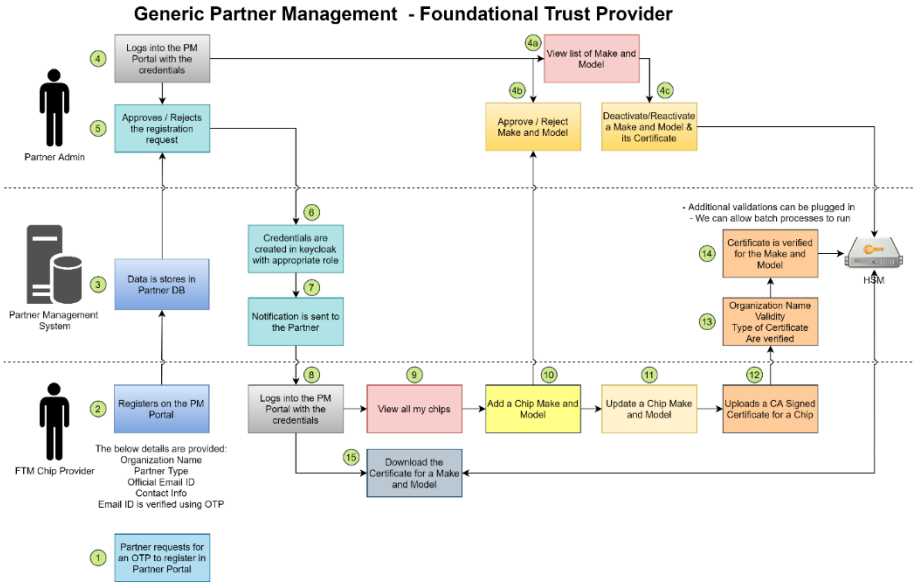
3. プロセスフロー

3.1 デバイスプロバイダー

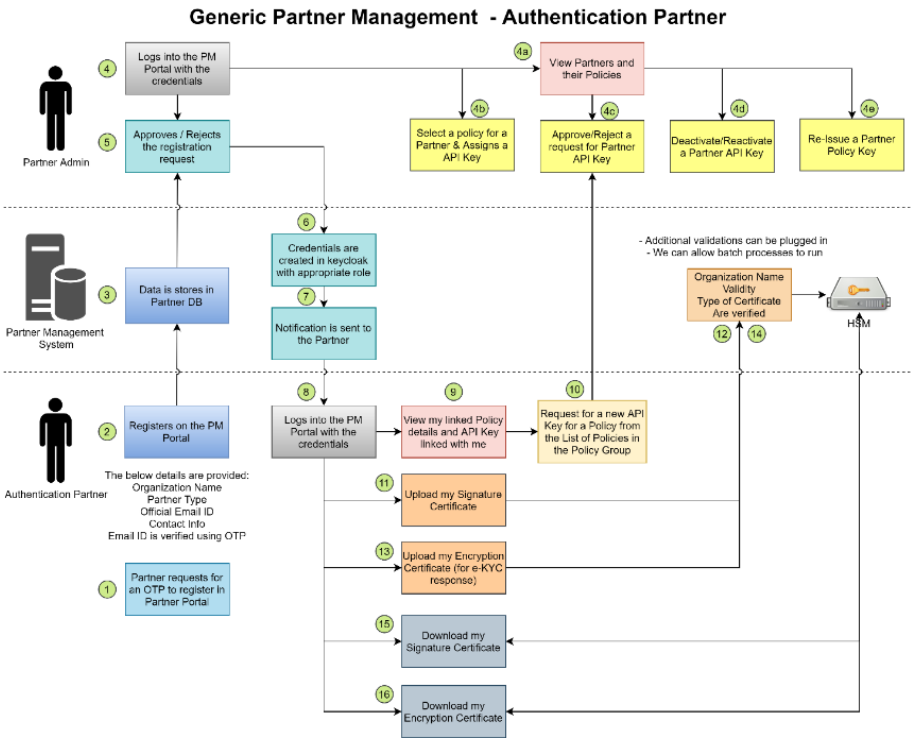


2.3.7 Partner Management（2/6）

3.2 ファンデーショナル・トラスト・プロバイダー

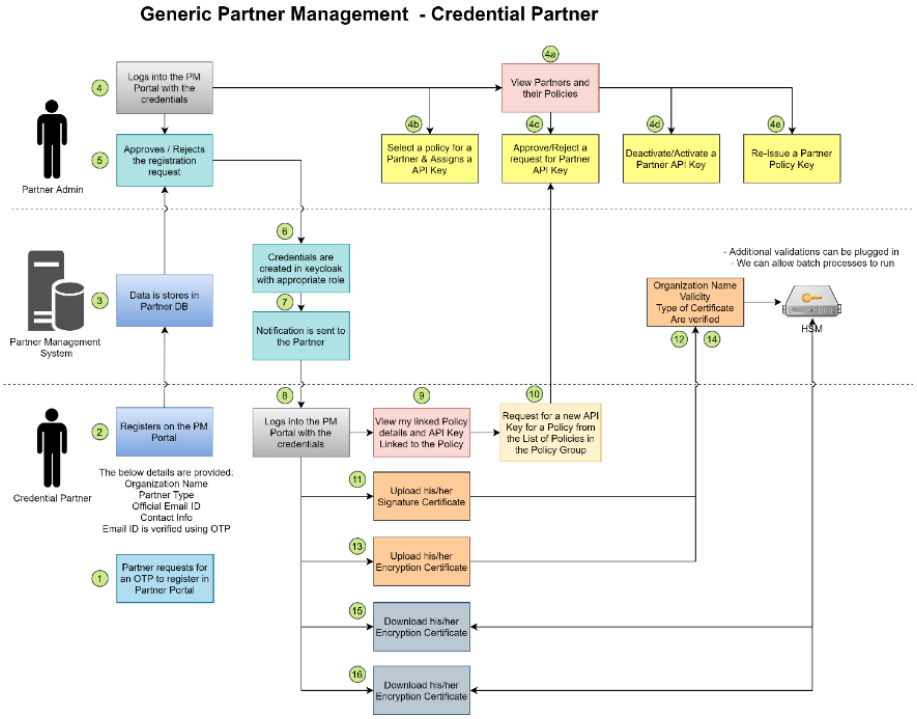


3.3 認証パートナー

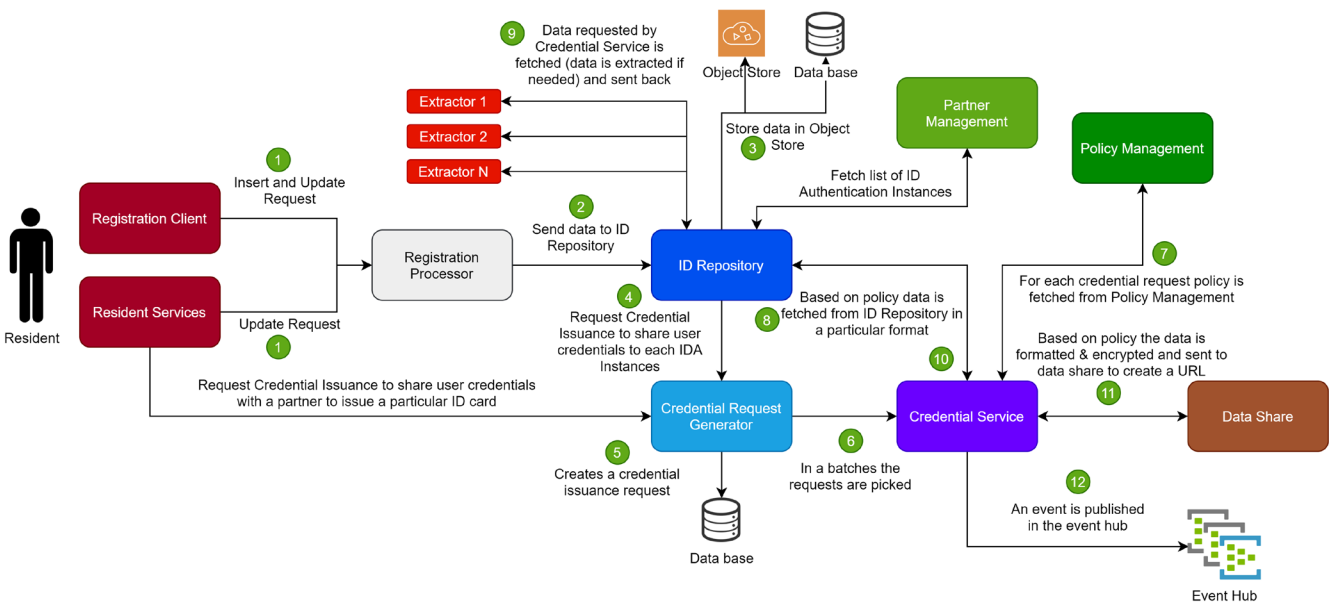


2.3.7 Partner Management（3/6）

3.4 証明書パートナー

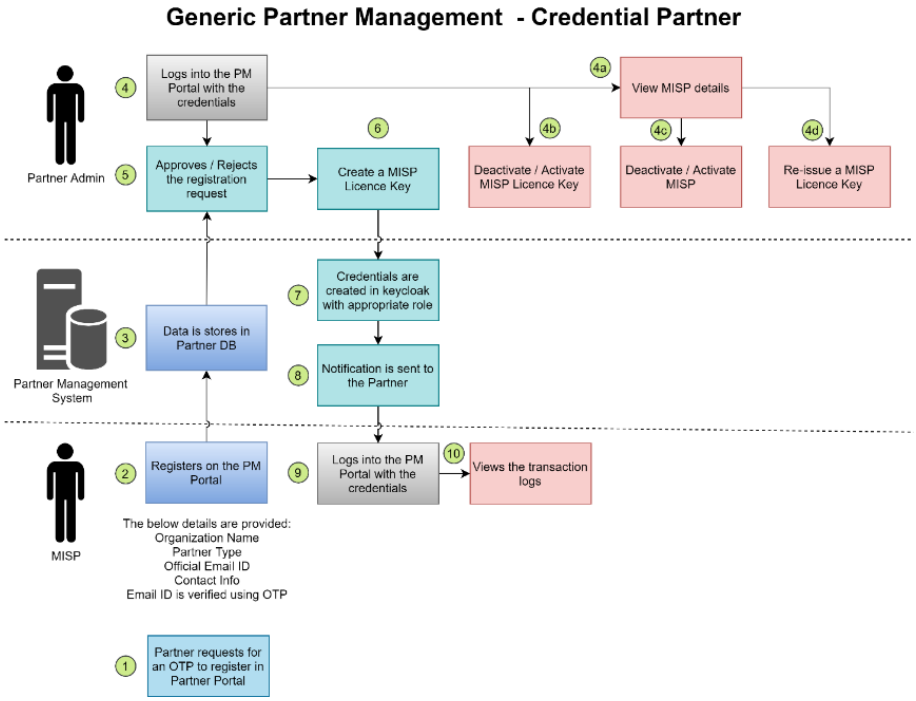


3.4 証明書パートナー：IDカードなど身分証明書発行のためのデータ取得
※Credential Serviceの詳細については[こちら]

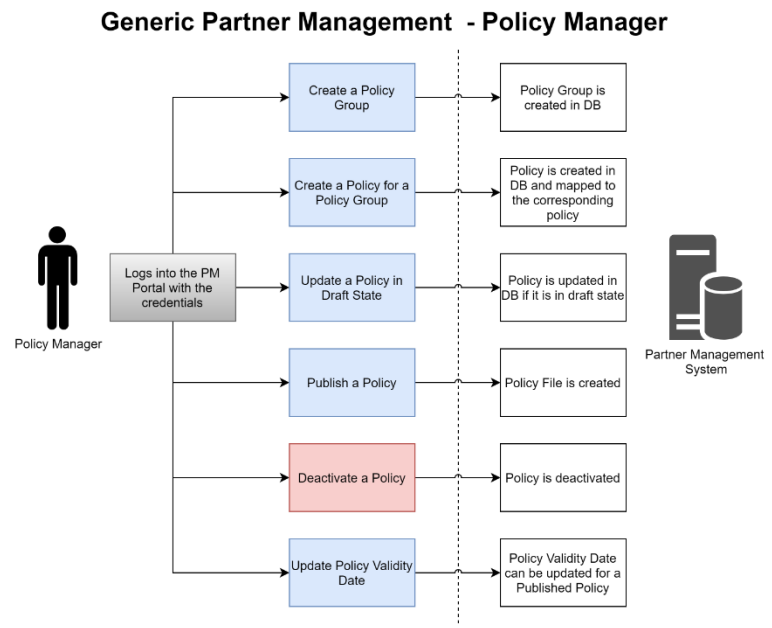


2.3.7 Partner Management（4/6）

3.5 MOSIPインフラストラクチャ・サービス・プロバイダー (MISP)



3.6 ポリシー管理



2.3.7 Partner Management（5/6）

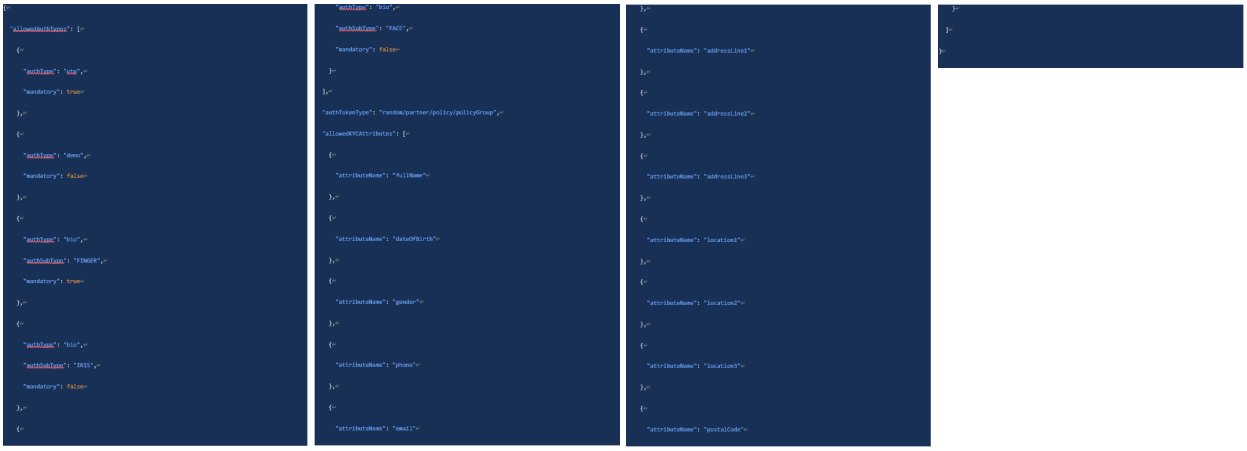
4. ポリシーおよびポリシーグループ

4.1 ポリシー

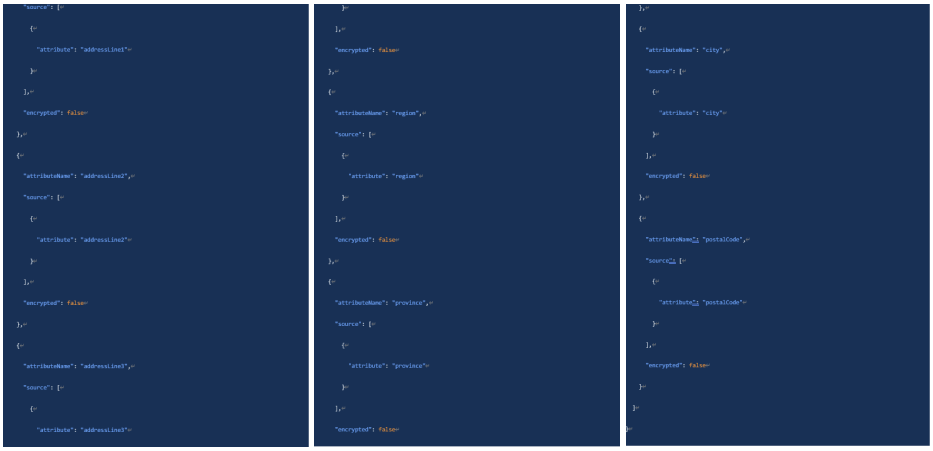
ポリシーとは、パートナーとMOSIPシステムの間のさまざまなアクションが記されたMOSIPのドキュメントである。さまざまなユースケースに応じて、いろいろなパートナーのポリシーが異なる場合がある。一般にMOSIPでは2つのタイプのポリシーがある。

1. 認証パートナー向けの認証ポリシー
2. 証明書パートナー向けの証明書発行ポリシー

4.1.1 認証ポリシーの例 (JSON)



4.1.2 証明書発行ポリシーの例 (JSON)



2.3.7 Partner Management（6/6）

4.2 ポリシーグループ

ポリシーグループとは、銀行、保険、通信などの業界または領域であり、導入国によって異なる。ポリシー管理者、パートナー管理者、およびパートナーはすべて特定のポリシーグループに所属できる。MOSIPでは、パートナー、パートナー管理者、ポリシー管理者を設定する前に、あらかじめ導入国がポリシーグループのマスターデータを作成し、定義しておくことが求められる。

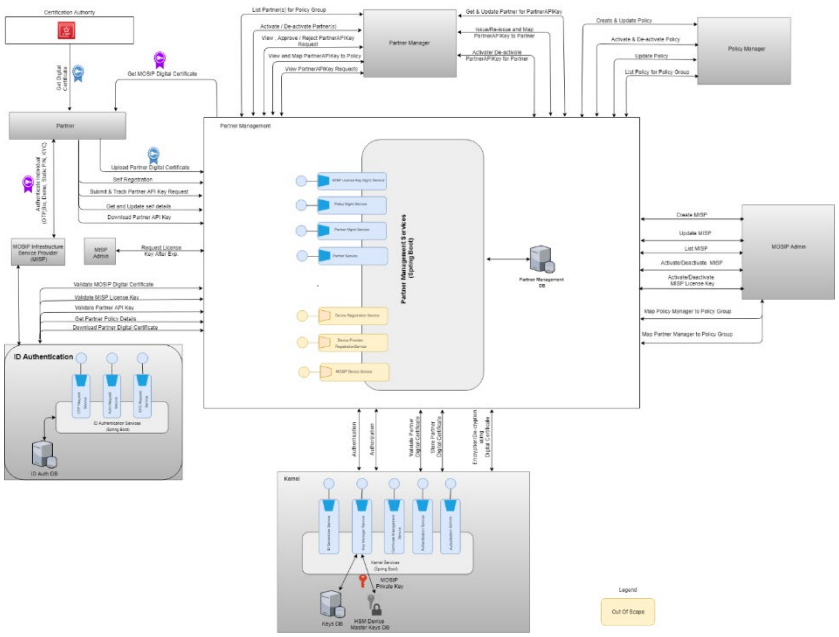
4.3 ポリシー管理者

ポリシー管理者は所属するポリシーグループのポリシーを作成および管理する。

4.4 パートナーAPIキー

認証ポリシーを受け入れたパートナーは、パートナーコード (PartnerCode)、ユースケースの説明 (UseCaseDescription)、サポート情報 (SupportingInfo)、ステータス (Status) などのパラメーターを使用して、パートナーAPIキー (PartnerAPIKey) リクエストを生成しなければならない。パートナー管理者がパートナーAPIキーリクエストを承認すると、パートナーには詳細情報が含まれるパートナーAPIキー (パートナーコードとポリシーグループおよびポリシー、発効日 (issuedOn)、有効期限 (validTill)、アクティブ状態 (isActive)などを組み合わせたもの) が提供される。

4.5 論理ビュー



5. サービス

パートナー管理サービスの詳細とハイレベルおよびローレベル設計については、パートナー管理リポジトリを参照

6. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、パートナー管理リポジトリを参照

7. API

パートナー管理

2.4.1 ID Repository（1/4）

概要

IDリポジトリには、個人のIDのレコードが含まれ、その他のMOSIPモジュールによってIDの詳細を保存、取得、更新するためのAPIベースの機能を備える。IDリポジトリは以下によって使用される

- 登録プロセッサー
- ID認証
- 住民サービス

1. 主な特徴

- 指定されたUINの識別情報を保存する
- 部分的な識別情報またはUINのステータスを更新する
- 有効なUINに関連付けられた識別情報を読み取る
- 指定されたRIDの識別情報を保存する
- UINのステータスをチェックしてUINを検証する

IDリポジトリ内に保存された識別データは暗号化される。これは最も重要なストレージリポジトリであり、次の点に留意して設定される

- スケーラビリティ
- パフォーマンス
- 高可用性

2. IDリポジトリの機能

2.1 識別データとドキュメントの保存

個人の身元情報を保存するリクエストを受信すると、リクエストに入力されているID属性を、導入国で定義されたMOSIP IDと照合して検証する

1. 登録時に生成された個人のID、JSON、生体認証ドキュメント、証跡ドキュメントが保存される
2. 個人の身元情報の保存が完了すると、個人のUINのステータスがデフォルトで 'ACTIVATED' になる

2.2 UINまたはRIDで保存された身元情報を読み出す

入力されたUINまたはRID、オプションのパラメーターであるタイプに基づいて個人の身元情報を読み出すリクエストを受信すると、以下の手順が実行される

1. 入力されたUINが 'ACTIVATED' であることを確認する
2. 個人の最新のID属性を読み出す
3. 人口動態情報または生体情報、またはその両方が読み出され、送信される

2.3 識別データとドキュメントの更新

個人の身元情報を更新するリクエストを受信すると、以下の手順が実行される

1. 入力されたUINが 'ACTIVATED' であることを確認する
2. 個人の入力ID属性を更新する
3. 人口動態情報または生体情報、またはその両方を更新する
4. リクエストに応じてUINのステータスを 'DEACTIVATED' または 'BLOCKED' に更新する
5. 更新された個人のIDの詳細情報を含む応答を送信する

2.4 UINと関連するVIDの非アクティブ化/再アクティブ化

個人のUIN/VIDは非アクティブ化または再アクティブ化できる

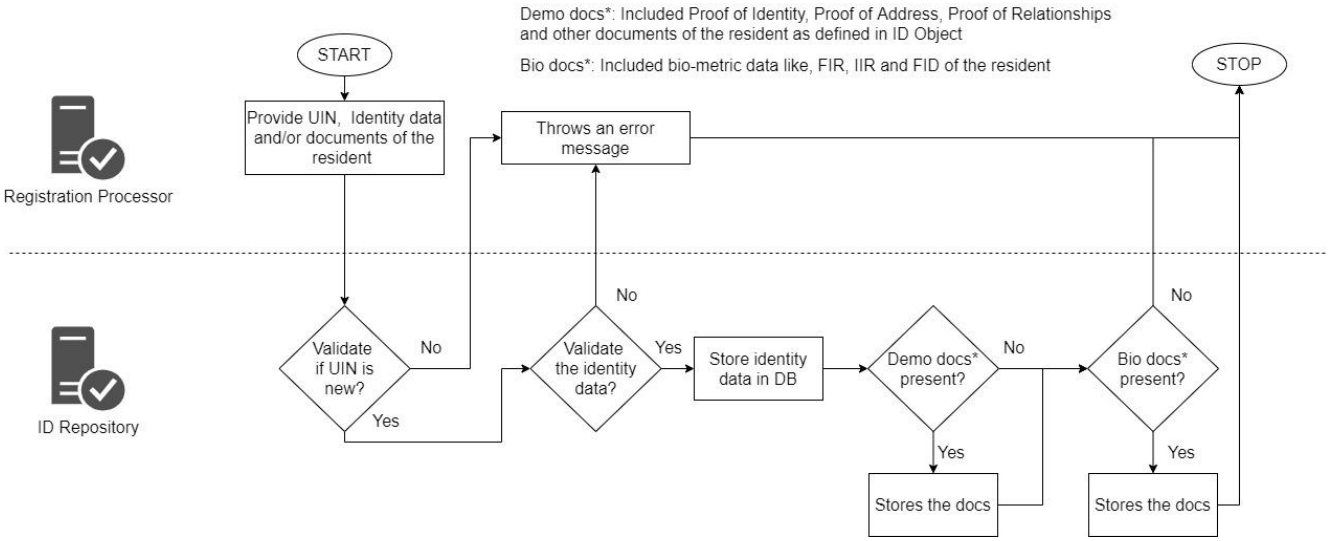
2.5 認証タイプのロック/ロック解除

個人は住民サービスを使って特定の認証タイプをロックまたはロック解除できる。たとえば、人口動態データの認証および/または生体認証をロックすると、システムで特定の検証が行われた後にリクエストされた認証タイプがロックされる。リクエストされた個人の認証タイプがロックされると、その人はロックされた認証タイプを使用して自分自身を認証できなくなる。同様に、その人はこれらの認証タイプをロック解除する選択もできる

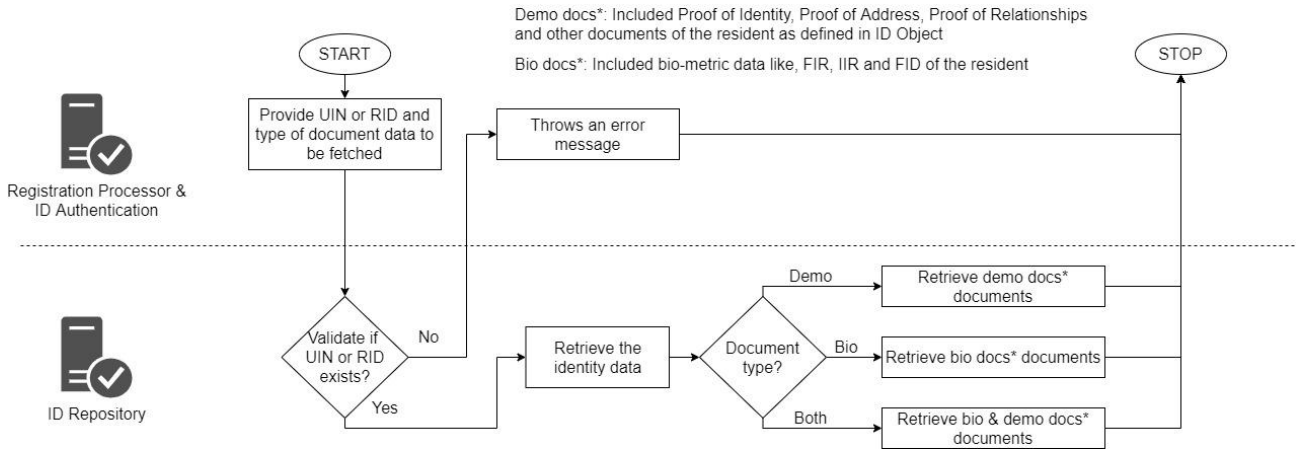
2.4.1 ID Repository（2/4）

3. プロセスフロー

3.1 識別サービス – 識別データとドキュメントの保存

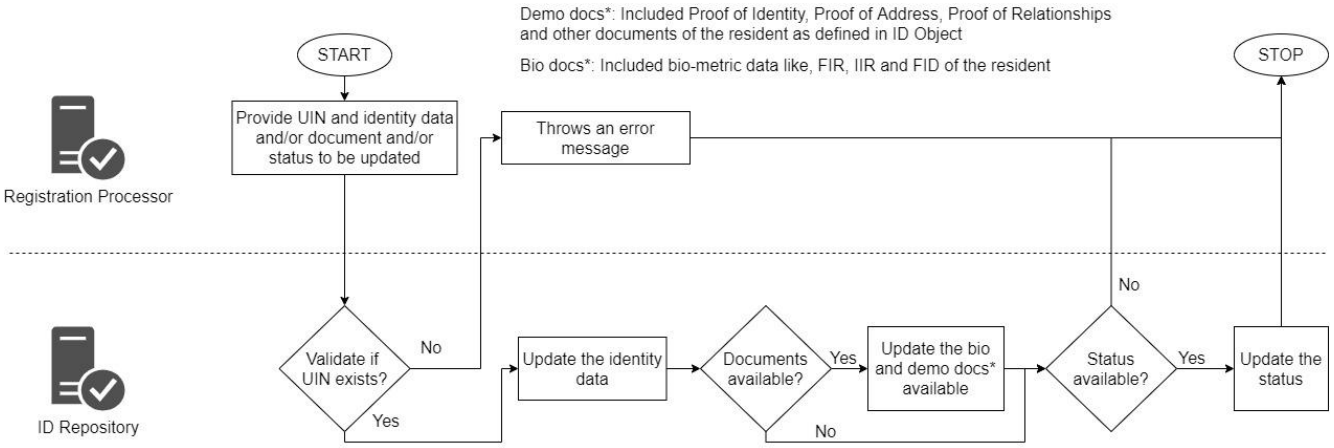


3.2 識別サービス – UINまたはRIDで保存された身元情報を読み出す

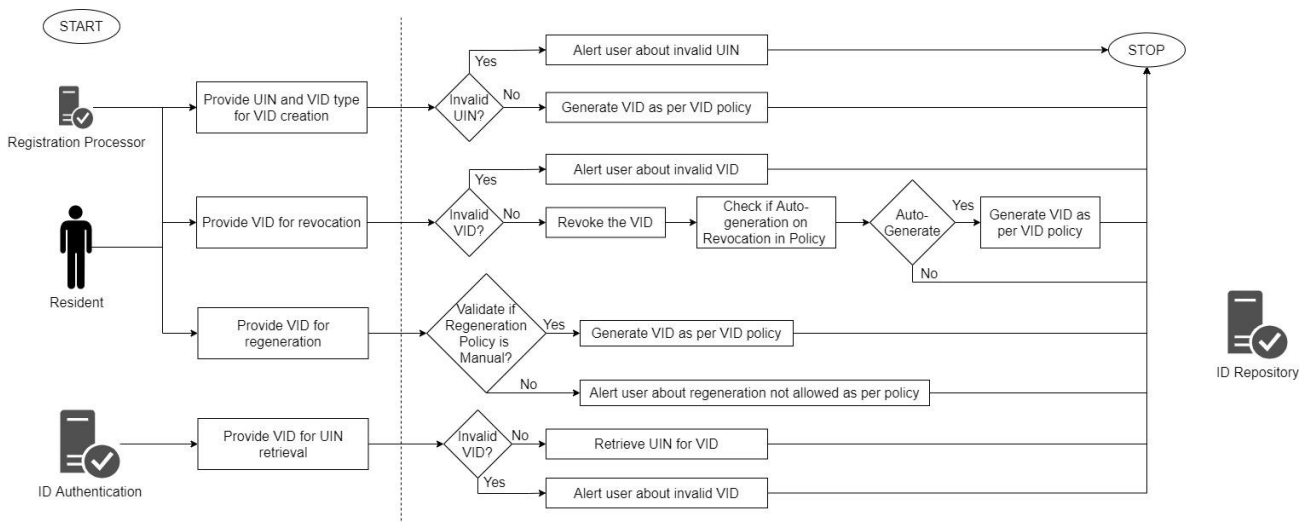


2.4.1 ID Repository（3/4）

3.3 識別サービス – 識別データとドキュメントの更新

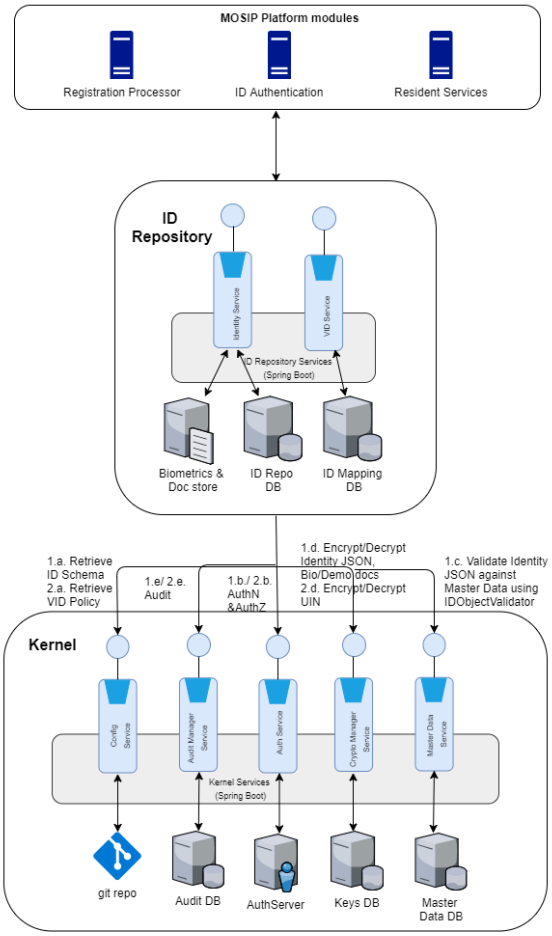


3.4 VIDサービス



2.4.1 ID Repository（4/4）

4. 論理ビュー



5. サービス

サービス、コード、デザイン、ドキュメンテーションは、共通リポジトリ/IDリポジトリにある

6. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、共通リポジトリ/IDリポジトリを参照

7. API

IDリポジトリAPI

2.4.2 Kernel

概要

MOSIPサービスはカーネル上に構築される。カーネルとは、より高レベルのサービスを構築するプラットフォームであると共に、セキュアなサンドボックスであり、高次のサービスはその中で動作する。カーネルは複数の重要不可欠な技術的機能を備えることによってサービスのビルドと実行を行う基盤を提供するので、個々のサービスはビジネス機能のみを意識すればよい。

カーネルは個別のモジュールではなく、サービスとライブラリの集合体であり、さまざまなモジュールがこれを共有する。

1. コンポーネント

カーネルの全コンポーネントについては、共通リポジトリ/カーネルを参照。

2. 詳細機能

カーネルには多数のサービスと機能がある。その一部の詳細については、以下を参照のこと。

- 共通サービス
- UIN & VID生成サービス
- データサービス
- マスターデータサービス
- キーマネージャー
- 監査マネージャー
- 認証および認可
- パケットマネージャー
- Web Sub

3. サービスとライブラリ

サービスとライブラリ、そのコードとデザインの詳細については、共通リポジトリ/カーネルを参照

4. ビルドおよびデプロイ

ビルドおよびデプロイ手順については、共通リポジトリ/カーネルを参照

5. API

カーネルAPI

2.5.1 Device Integration（1/3）

概要

MOSIPは、指紋、虹彩、顔などの生体認証を登録・認証プロセスで使用し、 生体認証の品質チェックのための生体認証データの特殊な処理や、２つの生体認証画像の照合が必要とする。バイオメトリクスSDKは、これらの機能を提供するソフトウェアライブラリで構成されている。なお、MOSIPプラットフォームにはこのようなSDKは含まれていない。

バイオメトリクスSDKは主に1:1認証と品質チェックに使用され、ABISは1:N重複排除に使用される。
MOSIPでは、1:1認証にABISシステムを使用することを推奨していない。

Biometric SDK機能

SDK Initialization

SDKに関する情報を共有し、内部変数やアルゴリズムの初期化を含む任意のワンタイムアクティビティを実行。

Quality Checker

入力されたバイオメトリクスの品質をチェックし、同じ品質スコアを返す。

ユースケース

- MOSIPが強制キャプチャを用いて登録クライアントで生体認証画像を受信した場合、この方法を用いて画像の品質を確認
- バイオメトリクス画像の品質をサーバー側で検証
- 外部からの生体認証画像を取り込んだ場合、その生体認証画像の品質を確認

Matcher

キャプチャされた生体情報レコードまたは生体情報レコードのリスト (単一一致または複合一致に基づく) が、保存された生体情報レコードのリストにマッチしているかを確認し、保存された各生体情報レコードに対するマッチングスコア、または入力された生体情報レコードのリストに対する複合マッチングスコアを返す。

ユースケース

- 認証トランザクションで受信したバイオメトリクスの1つまたは複数のモードをバイオメトリクスレコードのリストと照合
- オフラインモードでのオペレータの認証に使用
- 登録者の生体情報の代わりにオペレータの生体情報を誤って提出しないようにするために使用
- 画像と画像の比較、抽出と抽出画像の比較が可能

Extractor

入力された生体情報レコードの顕著な特徴とパターンを抽出し、高速比較、抽出された生体情報レコードを返す。

ユースケース

- 高速比較で使用するために、生体情報の顕著な特徴とパターンの抽出に使用
- 指紋の場合、これはMinutiaeと呼ばれ、minutiaeの標準的な表現はFMRのISOテンプレートを使用

Segmenter

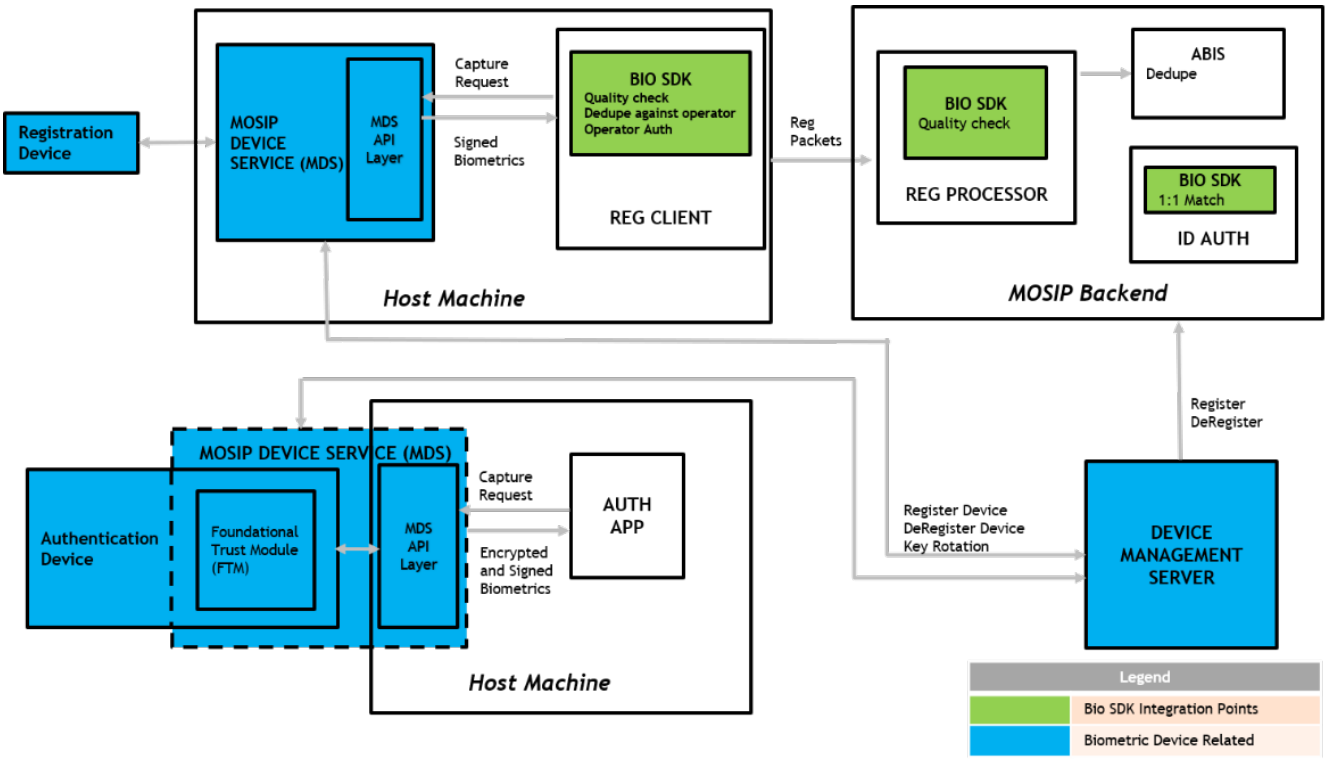
単一の生体情報レコードを複数の生体情報レコードに分割し、分割された生体情報レコードのリストを返す。

ユースケース

- 外部ソースから受信した画像を個々の生体情報セグメントに分割するために使用

2.5.1 Device Integration（2/3）

Biometric SDK integration points



Integration Point	Usage	SDK Platform
Registration Client	✓ Quality check ✓ Operator Local Authentication ✓ Prevention of erroneous submission of operator biometrics in place of registrant's biometric (1:n Match)	Windows
Registration Processor	✓ Quality Check	Linux
ID Authentication	✓ Match against a resident record ○ 1:10 (Finger) ○ 1:2 (Iris) ○ 1:1 (Face)	Linux

SDK Initialization

プラットフォームとしてのMosipは、バイオメトリクスを扱う機能を内蔵しておらず、バイオメトリクスに関連するすべての活動を実行するために、外部コンポーネントやサブシステムに依存しているが、外部コンポーネントやサブシステムのためのフォーマット、標準、インターフェイスは定義している。(リンク)

2.5.1 Device Integration（3/3）

デバイスのホワイトリスト化

MOSIPの登録機に接続して登録を行う生体認証機器は、MOSIP管理者がホワイトリストに登録し、機器提供者の管理サーバーに登録する必要がある。ホワイトリスト化は、MOSIP管理者によって実行され、それによりデバイスの詳細がMOSIP のマスターデータに保存され、MOSIP エコシステム内のさまざまなセンターにマッピングされる。

デバイスをホワイトリスト化するには、以下の手順に従う。

1. デバイスタイプが利用可能であること
2. デバイスの仕様が利用可能であること
3. デバイスを作成し、登録センターにマッピング

デバイスタイプの追加

デバイスタイプは、デバイスの種類を指定するマスターデータである。バイオメトリクスデバイスの場合、理想的には虹彩スキャナ、指紋スラップスキャナ、カメラの3種類のみとなる。 異なるタイプのデバイスに対して管理者が実施する。

管理者は、デバイスの仕様を作成する前に、デバイスタイプがデバイスに対して利用可能であることを確認し、デバイスタイプが利用できない場合は、管理者がデバイスタイプを作成することができる。

MOSIPでは、3つの方法でデバイスタイプを作成することができる。

1. デバイスタイプAPIを利用する
2. MOSIP管理者ポータルのデバイスタイプ画面を利用する (ログイン > マスターデータ > デバイスタイプ)
3. MOSIP管理者ポータルのバルクアップロード画面 (ログイン > バルクアップロード > マスターデータアップロード) を利用

2.5.2 Communication Integration

コミュニケーションインターフェース

デバイスは、以下のAPIセットのみを実装する。すべてのAPIは物理層とオペレーティング・システムから独立しており、呼び出しはオペレーティング・システムによって異なる。

本仕様書ではオペレーティングシステム名を定義しているが、特定されていないオペレーティングシステムにも同様の技術を使用することができる。デバイスサービスは、デバイスがホストにローカルに接続されていることを保証する。

様々なデバイスにおけるパラメータやリクエスト/レスポンスの詳細に関しては、"Device Service - Communication Interfaces"を参照。

MOSIPにおいて定義されているAPIは、こちらから参照のこと。

2.5.3 Offline Integration

MOSIPは、オフラインでも以下のサービスを提供。

- MOSIPオペレーター
- Design choices
- 登録
- 登録クライアントセットアップ
- Biometric SDK 機能
- オフライン検証のための準同型暗号
- Data Sync
- 分析と監査ログ
- パケットアップロード
- 認証と認可 (TBD)
- GET /policies
- POST /partners
- Configuration
- Hardware Security Module HSM Specifications

2.6.1 Automated Biometric Identification System（ABIS）（1/2）

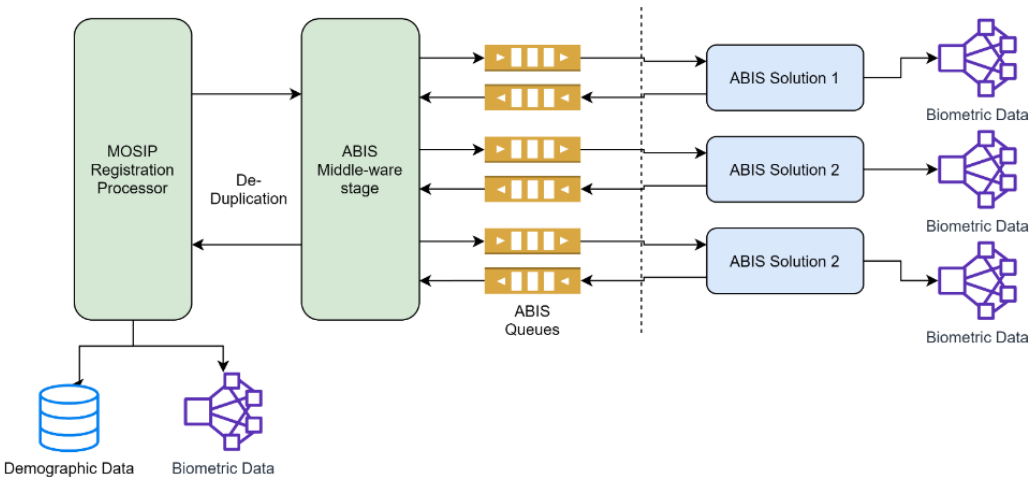
概要

住民に一意のIDを提供することは、IDプラットフォームの主要機能の1つである。このために、MOSIPは**Automated Biometric Identification System（ABIS）**と接続して住民の生体認証データの重複排除を行う。

MOSIPは複数のABISと統合してさまざまなABISプロバイダーの経験を活用できるよう設計されている。導入国は、指紋認証と虹彩認証で別のABISを使用したり、同じ生体認証データに複数のABISを使用して、重複排除の品質に基づいて最も優れたABISを評価したりできる。

ABISシステムが住民の識別情報を知ることは決してない。人口動態データ詳細や登録リクエストID（RID）といった個人識別情報（PII）はABISシステムと共有されない。MOSIPはシステム内にABIS固有の参照ID（referenceID）と住民のRIDのマッピングを保持している。

ABISは1対Nの重複排除に使用される。1対1認証には、バイオメトリクスSDKが使用される。MOSIPは、ABISを1対1認証に使用することを推奨しない。

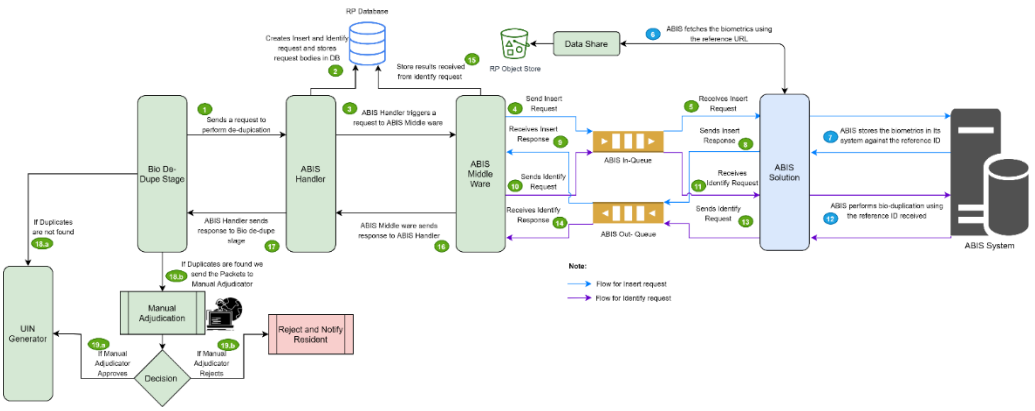


1. ABISミドルウェア

MOSIPのABISミドルウェアには以下のコンポーネントがある

- MOSIP ABISリクエストハンドラー
- リクエストルーター（ABISリクエストはルーティングポリシーに従って正しいABISシステムにルーティングされる）
- ABISレスポンスハンドラー

MOSIP ABISミドルウェアのプロセスフローを下図に示す



2.6.1 Automated Biometric Identification System (ABIS)（2/2）

2. MOSIP-ABISインターフェース

MOSIPはメッセージキューを介してのみABISと連携する。キューに入るすべての制御メッセージはJSONフォーマットである。
MOSIP ABISミドルウェアは受信キューのアドレスにリクエストを送り、送信キューのアドレスからレスポンスを受け取る。
詳細については、ABIS API仕様を参照

ABISは以下のタイプの生体情報画像をサポートしなければならない

- 個別の指紋画像（セグメント分けされたもの）
- 虹彩画像（左、右）
- 顔画像

MOSIP内の生体認証データは生体認証データ仕様に定義された形式でやり取りされる

3. ABISのデプロイ

- ABISはABIS API仕様に従わなければならない
- キューはRegistrationProcessorAbis-env.jsonファイルで構成可能
  ABISシステムは事前定義されたユーザーIDとパスワードでキューに接続される
- ABISは、登録プロセッサーをデプロイした場所と同じセキュアゾーン（武装地帯）にデプロイすることを推奨する
- ABISシステムを外部ネットワークと接続することは推奨されない

2.6.2 バイオメトリクス SDK（1/2）

はじめに

MOSIPでは、登録プロセスおよび認証プロセスに指紋、虹彩、顔などの生体情報(バイオメトリクス)を使用する。
登録および認証には、生体情報の品質チェックや2つの生体情報画像の照合といった、特殊なデータ処理を必要とする。
バイオメトリクスSDKは、これらの機能を提供するソフトウェアライブラリで構成される。MOSIPプラットフォームにはこのようなSDKが含まれないことに注意すること

バイオメトリクスSDKは、主に1対1の認証および品質チェックに使われる。一方、ABISは1対Nの重複排除に使用される。
MOSIPは、ABISシステムを1対1認証に使用することは推奨しない

1. バイオメトリクスSDKの機能

1.1 SDKの初期化(SDK Initialization)

SDKに関する情報を共有し、内部変数やアルゴリズムの初期化など1回だけのアクティビティを実行する

1.2 品質チェック(Quality Checker)

入力された生体情報のチェックを行い、同一性について品質スコアを返す

1.2.1 ユースケース

- 強制キャプチャを使用して登録クライアント内のMOSIPが生体認証画像を受信すると、このメソッドを使用して画像の品質をチェックする
- サーバー側での生体認証画像の品質の検証にこのメソッドを使用する
- レコードに追加する外部の生体情報画像を受信したとき、このメソッドを使用して受信した生体画像の品質を判断する

1.3 照合(Matcher)

取得した生体情報レコードまたは生体情報レコードのリストを(単一照合または範囲照合で)照合する。また、保存された生体情報レコードのリストに対して照合する。その後、保存された生体情報レコードのそれぞれに対して照合スコアを返すか、または入力された生体情報レコードのリストについて範囲照合スコアを返す

1.3.1 ユースケース

- 認証トランザクションで、生体情報レコードのリストと1つまたは複数のモードの生体情報の照合に使用する
- オフラインモードでオペレーターの認証に使用する
- 登録クライアントで登録者の生体情報の代わりにオペレーターの生体情報を誤って提出しないようにするために使用する
- 照合は、画像対画像、抽出データ対抽出データ、抽出データ対画像のいずれの比較もできることが期待される

1.4 抽出 (Extractor)

迅速に比較できるよう、入力された生体情報レコードから目立った形状やパターンを抽出する。抽出された生体情報レコードを返す

1.4.1 ユースケース

- 迅速に比較できるよう、生体情報から目立った形状やパターンを抽出するのに使用される
- 指紋の場合、これは特徴点と呼ばれ、標準的な表現形式はFMR用ISOテンプレートである

1.5 セグメント化 (Segmentar)

単一の生体情報レコードを複数の生体情報レコードに分割し、分割された生体情報レコードのリストを返す。
例:入力板の指紋画像を複数の指の指紋に分割したり、目の画像を左目と右目に分割したりする

1.5.1 ユースケース

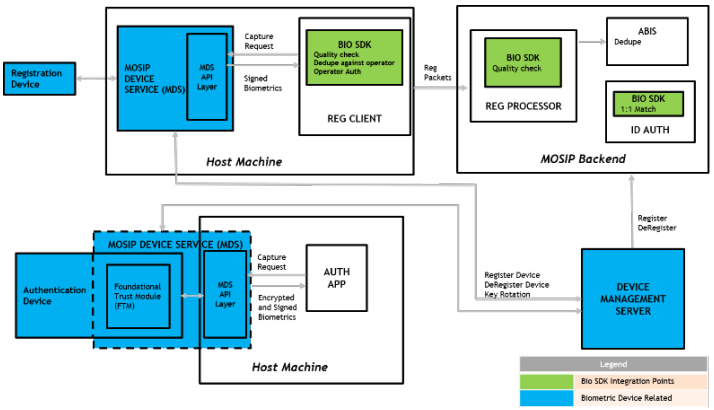
- 外部から受け取った画像を個々の生体情報セグメントに分割する

1.6 変換 (Converter)

生体認証レコード中のすべてのセグメントの画像を変換する

2.6.2 バイオメトリクス SDK（2/2）

2. バイオメトリクスSDKの統合ポイント



Integration Point	Usage	SDK Platform
Registration Client	✓ Quality check ✓ Operator Local Authentication ✓ Prevention of erroneous submission of operator biometrics in place of registrant's biometric (1:n Match)	Windows
Registration Processor	✓ Quality Check	Linux
ID Authentication	✓ Match against a resident record o 1:10 (Finger) o 1:2 (Iris) o 1:1 (Face)	Linux

3. バイオメトリクスSDK API仕様

SDKは、バイオメトリクスSDK API仕様に準拠する必要がある

2.6.3 MDS 仕様（1/36）

1. はじめに

1.1 目的

本仕様書の目的は、MOSIPソリューションで使用する生体認証デバイスに必要な技術的およびコンプライアンス的な規則および手順を確立することである

1.2 対象読者

これは生体認証デバイスの仕様書であり、生体認証デバイスの製造者、その開発者、設計者のMOSIP準拠デバイス開発を支援することを目的とする。読者はMOSIPの登録および認証サービスに精通していると仮定している

1.3 MOSIPデバイス

MOSIP用に生体認証データを収集するすべてのデバイスは、本ドキュメントの仕様の範囲で動作する

2. 改訂履歴

バージョン	状態	日付	変更箇所
0.9.2	凍結	2019年8月	
0.9.3	凍結	2020年2月	
0.9.5	草稿	2020年1月13日	
0.9.5	草稿	2020年8月10日	暗号化証明書を取得するAPIのシグネチャをGETからPOSTに変更、デバイスストリームでオプションのパラメーター timeout を新規にサポート
0.9.5	草稿	2020年12月4日	JWTの規格に従い、JWT Signatureのヘッダーにおいて型を格納するキーを "type "から "typ" へ変更。変更についてはデジタルID仕様を参照のこと

3. 用語集

- デバイスプロバイダー – デバイスを自身の名前で製造または輸入するエンティティ。このエンティティは、その国の各当局から組織レベルの電子証明書を取得する法的権利を有していなければならない
- ファンデーショナル・トラスト・プロバイダー – ファンデーショナル・トラスト・モジュールを製造するエンティティ
- デバイス – 生体情報をキャプチャ可能なハードウェア
- L1認証デバイス/ L1デバイス – デバイスの信頼ゾーンの中で仕様に従って暗号化を実行できると認定されたデバイス
- L0認証デバイス/ L0デバイス – ホストマシンのデバイスドライバーまたはMOSIPデバイスサービス上で暗号化が行われていると認定されたデバイス

2.6.3 MDS 仕様（2/36）

- ファンデーショナル・トラスト・プロバイダー証明書 – ファンデーショナル・トラスト・プロバイダーに発行されるデジタル証明書。この証明書は、プロバイダーとして必須のファンデーショナル・トラスト・プロバイダー評価に合格していることを証明する。エンティティはHSM上の安全な場所にこの証明書を保管するものとする。個別の信頼証明書はすべてこれをルート証明書として発行される。この証明書は導入国がMOSIPと連携して発行する
- デバイスプロバイダー証明書 – デバイスプロバイダーに発行されるデジタル証明書。この証明書は、プロバイダーがL0コンプライアンスまたはL1コンプライアンスに準拠していることを証明する。エンティティはHSM上の安全な場所にこの証明書を保管するものとする。個別の信頼証明書はすべてこれをルート証明書として発行される。この証明書は導入国がMOSIPと連携して発行する
- 登録 – 基本IDを適用するプロセス
- KYC – 顧客確認プロフィールの検証および更新の実行に同意を得るためのプロセス
- 認証 – 個人のIDを検証するプロセス
- FPS – フレーム/秒
- 管理サーバー – 生体認証デバイスのライフサイクルを管理するためにデバイスプロバイダーが運用するサーバー
- デバイス登録 – デバイスをMOSIPサーバーに登録するプロセス
- シグネチャ – すべてのシグネチャはRFC 7515のとおりとする
- シグネチャのヘッダー – シグネチャのヘッダーとはRS256で設定される "alg" およびbase64encoded証明書で設定されるx5cの属性を意味する
- ペイロードは実データのバイト配列であり、常にbase64urlencoded形式で表される
- シグネチャ – base64urlencodedのシグネチャバイト
- ISO形式の時間 – ISO 8601。フォーマットは yyyy-mm-dd HH:MM:ssZ である

4. デバイス仕様

MOSIPデバイス仕様は、デバイスがMOSIPで動作するためのコンプライアンスのガイドラインを提供する。コンプライアンスはデバイスの機能、信頼性、および通信プロトコルに基づく。MOSIPに準拠したデバイスは本ドキュメントで規定される規格に従う。デバイスは本仕様に準拠して試験および検証される。詳細を以降のセクションに示す

4.1 デバイスの機能

MOSIP準拠デバイスは、以下を満たす

- 1つまたは複数の生体情報を収集する機能を有すること
- キャプチャした生体情報の画像またはテンプレートに署名する機能を有すること
- 秘密鍵を保護する機能を有すること
- 生体情報を注入する仕組みを持たないこと

2.6.3 MDS 仕様（3/36）

4.2 デバイスの基本仕様

4.2.1 指紋のキャプチャ

ISO 19794-4:2011を参照

要素	登録デバイス	認証デバイス
最小解像度	ネイティブ 500 dpi 以上。これは最低限の値であり、ぎりぎり推奨されるものである。より高い値が望ましい	ネイティブ 500 dpi 以上。これは最低限の値であり、ぎりぎり推奨されるものである。より高い値が望ましい
本人拒否率（FRR）2)	各国で 2% FRR 未満	各国で 2% FRR 未満
他人受入率（FAR）2)	0.01%	0.01%
DPI	500[1]	500
画像仕様	ISO 19794-4 B.1 AFIS Normative	ISO 19794-4 B.2 Personal Verification
ESD	>= 8kv	>= 8kv
EMSコンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
動作温度2)	0 ～ 50℃	□30 ～ 50℃
生存検出3)	IEEE 2790 に従う	IEEE 2790 に従う
プレビュー	JPEG losslessフレームで3 FPS以上、NFIQ 2スコアを画面に重ねて表示	なし
画像フォーマット	JPEG 2000 lossless	JPEG 2000 lossless、WSQ (圧縮最大10:1) 2)
品質スコア	NFIQ 2	NFIQ 1
FTM	L0 – ホストベースのセキュリティを使用 L1 – FTMがサポートするセキュリティ	L1 – FTMがサポートするセキュリティ L2 – 改ざん防止付き

1. 登録には有効性を十分に確認 2. 必要な場合、MOSIP導入国で変更可能 3. この機能を使用するかどうかはMOSIP導入国で決定

2.6.3 MDS 仕様（4/36）

4.2.2 虹彩キャプチャ

ISO 19796-6:2011 Part 6 Specificationsを参照

要素	登録デバイス	認証デバイス
回転角度	圧縮前に虹彩画像に前処理を行い、回転角度を計算する必要がある。球面収差が補正された画像の回転角度計算については、ISO 19794-6 Section 6.3.1 を参照。	-
回転の不定性	ISO 19794-6 を参照	-
最小直径	ISO 19794-6:2011 のとおり、中～高品質画像のみが受付可能。したがって本規格では、受付可能な虹彩の直径の最小値は150ピクセルとする。	同じ
マージン	ISOと同じ	-
色	虹彩画像はピクセル深度8ビット/ピクセルのグレースケールでキャプチャおよび保存すること	-
照明	眼にあてる光には、高品質なグレースケール画像を生成できる赤外線またはその他の光源を使用すること	-
画像フォーマット	JPEG 2000 lossless	JPEG 2000 lossless
アスペクト比	1:1	-
画像品質	ISO/IEC 29794-6	ISO/IEC 29794-6
動作温度[1)]	□30 ～ 50℃	□30 ～ 50℃
EMS コンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
プレビュー	JPEG losslessフレームで3 FPS以上、品質スコアを画面に重ねて表示	適用なし
画像仕様	ISO 19794-6	ISO 19794-6
ISO形式	K3	K7
FTM	L0 – ホストベースのセキュリティを使用 L1 – FTMがサポートするセキュリティ	L1 – FTMがサポートするセキュリティ L2 – 改ざん防止付き

1. MOSIP導入国で決定し最終化

2.6.3 MDS 仕様（5/36）

4.2.3 顔のキャプチャ

ISO 19794-5:2011を参照

要素	登録デバイス	認証デバイス
最小解像度	2.8 mm、画角110度で 1080ピクセル	2.8 mmで1080ピクセル
肌トーン	すべて	すべて
動作温度[1)]	□30 ～ 50℃	□30 ～ 50℃
EMSコンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
画像仕様	ISO/IEC 19794-5	ISO/IEC 19794-5
例外画像仕様	FACE機能で真正面、顔の横に手のひら2つ分のスペースがある上半身の写真。6 x 4 mm	適用なし
画像品質	ICAO – 真正面の画像、回転 ±5 度、24ビットRGB、白背景、幅35 mm、高さ45 mm	-
画像フォーマット	JPEG 2000 lossless	JPEG 2000 lossless
FTM	L0 – ホストベースのセキュリティを使用 L1 – FTMがサポートするセキュリティ	L1 – FTMがサポートするセキュリティ L2 – 改ざん防止付き

登録や認証の場面で使用するデバイスは、導入国で人間工学的形状、アクセシビリティ、使いやすさ、デバイスの共用可能性を検討しつつ選択することが推奨される

1. MOSIP導入国で決定し最終化

2.6.3 MDS 仕様（6/36）

5. デバイスの信頼性

MOSIP準拠デバイスは、信頼できる環境を、登録、KYC、および認証の場面で使用されるデバイスに提供する。
信頼レベルは、動作の信頼性を確保するためにデバイスがサポートする機能に基づいて規定されている

- L1 – セキュア実行環境を備えたセキュアチップによって信頼性が提供される
- L2 – セキュア実行環境とデバイス全体の完全な改ざん防止機能および不正対抗機能を備えたセキュアチップによって信頼性が提供される
- L0 – 信頼性はソフトウェアレベルで提供される。ハードウェアで提供される信頼性は存在しない。このタイプのコンプライアンスは管理下にある環境で使用される

5.1 ファンデーショナル・トラスト・モジュール（FTM）

ファンデーショナル・トラスト・モジュール（FTM）は、必要なすべての生体認証情報処理を行うセキュアなマイクロプロセッサーと、鍵を保存するセキュアストレージで作成される。以下の要件を満たす

- モジュールは暗号鍵を安全に生成、保存、処理する能力を有すること
- 非対称鍵および対称鍵をTRNGで生成すること
- モジュールは抽出から鍵を守る機能を有すること
- モジュールは物理的な改ざん、温度、周波数、および電圧に関する攻撃から保護しなければならない
- モジュールはハードウェアのクローニングに耐えること
- モジュールはプローブ攻撃に耐えること
- モジュールは、暗号処理のためのメモリー分離とバッファオーバーフロー攻撃からの保護を提供すること
- モジュールは、電力差分解析攻撃やタイミング攻撃などの暗号のサイドチャネル攻撃に耐えること
- 暗号アルゴリズム実装はCAVP認証済であること
- モジュールは暗号的に検証可能なセキュアブートを実行できること
- モジュールは信頼されたアプリケーションを実行する能力を有すること
  このモジュールで提供される基礎的なデバイスの信頼性によって、生体認証情報をキャプチャするためのトラストベースのコンピューティングが可能になる。FTMにより、以下の機能を備える信頼された実行環境が提供される
- セキュアブート
  - 実行前にコードを暗号的に検証する
  - モジュール/デバイスの一貫性違反をチェックする
  - 故障時には停止する
  - 安全にアップグレードし、アップグレード後の処理のみを実行してダウングレード攻撃を阻止する
  - ハッシュ化が必要な場合は常にSHA256ハッシュ相当以上を使用する必要がある
  - 信頼の起点（Root of Trust）はすべて最初のブートまたはそれ以前にプロビジョニングされていること
  - すべてのアップグレードは、適切なハッシュと署名を検証して起動に成功した後にのみ成功とみなされる
  - ブート時にハッシュや署名に失敗したときはブート失敗としなければならず、中間状態では決して動作しないこと
  - 最大10回失敗した場合はアップグレードのプロセスをロックし、デバイスを動作不能とすること。ただし、チップメーカーはこれを10回よりも少なくしてもよい
- アプリケーションの安全性の確保
  - 信頼されたアプリケーションを実行できる
  - アプリケーションをダウングレードから保護する
  - 暗号化オペレーションは隔離されたメモリー上で行われる
  - すべての信頼性は最初のブート時に固定され、修正できないようになっている

2.6.3 MDS 仕様（7/36）

5.1.1 証明書

FTMは指定する要件を満たす以下の証明書を各カテゴリーで少なくとも1つ有すること

カテゴリー：暗号化アルゴリズム実装

- CAVP (RSA、AES、SHA256、TRNG (DRBGVS)、ECC)

補助アルゴリズムおよび曲線のリストはこちら

カテゴリー：FTMチップ

- FIPS 140-2 L3 以降
- PCI PTS 5 以降（認証前）
- Common Criteria（EAL4 以降）
  - TODO:FILL IN

カテゴリー：改ざん防止

- L1レベルのコンプライアンスには、改ざん証跡をサポートすること
- L2レベルのコンプライアンスには、L1をすべてサポートし、かつ不正対抗機能を導入できること

5.1.2 保護される脅威

FTMは以下の脅威から保護されなければならない

- ハードウェアクローニング攻撃 – 鍵を複製されてしまう攻撃から保護する機能を有すること
- ハードウェア改ざん攻撃
  - 物理的改ざん – 物理的に改ざんして機密を入手する手段のないこと
  - 電圧および周波数に関連する攻撃 – 電圧のリークが遮蔽され、低電圧が防御されていること。FTMは常に正常動作または動作不能のいずれかの状態をとること。FTMは入力電圧が基準を満たさない場合に決して動作してはならない
  - 暗号化ブロックへの温度攻撃 – FTMの低温（Low）または高温（High）の場合はFTMは動作するか、動作不能状態になることが期待される。中間状態は存在しない
- 電力差分解析攻撃
- プローブ攻撃 – FTMはプローブに関連する攻撃からその表面が保護されていること
- 暗号化オペレーション実行メモリーの分離（暗号ブロックはバッファオーバーフロー型攻撃から保護されていること）
- 暗号化アルゴリズム実装の脆弱性
- セキュアブートとセキュアアップグレードに対する攻撃
- TEE/セキュアプロセッサーOSへの攻撃

5.1.3 ファンデーショナル・トラスト・モジュールの識別

MOSIP導入国がFTMプロバイダーを承認すると、FTMプロバイダーは自己署名形式の公開証明書を導入国に提出する。これをFTMルートと呼ぶ。導入国はこの証明書をデバイス・トラスト・データベースのシードとして使用する。FTMルートとその鍵ペアはFIPS 140-2レベル3以降に準拠し、鍵を抽出する可能性があるメカニズムを持たないデバイスで生成し、保存する。FTMの最初のブートではランダムな非対称鍵ペアが生成され、有効な証明書を取得した鍵の公開部分が提供される。FTMプロバイダーは検証を行ってチップが一意であることを確認し、FTMルートが発行者として設定された証明書を発行する。証明書の発行の全工程は、プロビジョニングされた安全な場所で行われる。導入国またはその承認された監査人による通知により監査可能である。モジュールに発行された証明書には、MOSIP導入国によって定義されたMOSIP証明書ポリシードキュメントに従って定義された有効期限がある。FTMチップ内のこの証明書と秘密鍵は、永続的メモリー上にあることが望ましい

チップの証明書の有効期限は製造日から20年を超えてはならない

2.6.3 MDS 仕様（8/36）

5.2 デバイス

MOSIPデバイスは生体認証情報の収集に最もよく使用される。デバイスはすべてのレベルのコンプライアンスと使用方法についての仕様に準拠することが望ましい。MOSIPデバイスは、MOSIPアーキテクチャで定義される信頼レベル3（TL3）に分類される。TL3デバイスは完全な能力を持つPKIでホワイトリスト化され、鍵がハードウェアで安全に保存されるものとする

- L0 –あるデバイスがソフトウェアレベルの暗号化ライブラリを使用し、セキュアブート機能やFTMを有しない場合、デバイスはL0証明書を取得できる。これらのデバイスは異なるデバイス識別に従い、これは例外フローの一部でも言及される
- L1 – あるデバイスに安全性機能と認証済FTMのいずれかが組み込まれている場合、デバイスはL1証明書を取得できる
- L2 – あるデバイスに安全性機能と認証済みFTMのいずれかが組み込まれており、不正対抗機能を有する場合、デバイスはL2証明書を取得できる。また、デバイスはそのライフタイム全体において不正対抗が可能であること

5.2.1 デバイス識別

MOSIPに接続されたすべてのデバイスが識別できることは必要不可欠である。MOSIPでは暗号化されたIDを信頼の基礎としている

物理ID

MOSIPに準拠していることを示す識別マークと、読み取り可能な一意のデバイスシリアル番号（最低12桁）、製造情報、およびモデル。同じ情報が2次元QRコードまたはバーコードで表示されていなければならない。これは現地でのサポートおよび検証に役立つ

デジタルID

MOSIPデバイスのデジタルIDは、以下のような署名付きJSONオブジェクト（REC 7515）である

```
{
  "serialNo": "Serial number",
  "make": "Make of the device",
  "model": "Model of the device",
  "type": "Type of the biometric device",
  "deviceSubType": "Subtypes of the biometric device",
  "deviceProvider": "Device provider name",
  "deviceProviderId": "Device provider id",
  "dateTime": "Datetime in ISO format with timezone. Identity request time"
}
```

識別キー "Foundational Trust Module" のJSON Webシグネチャ（RFC 7515）で署名されたこのデータは、デバイスの基礎識別情報である。あらゆるMOSIP準拠デバイスにはファンデーショナル・トラスト・モジュールが必須である

この規則の例外は、目的に "Registration"（デバイスの登録の項で後述）が設定されたL0準拠デバイスのみである。L0デバイスは、デバイスキーでデジタルIDに署名する

署名済みのデジタルIDの例は、以下のとおりである

2.6.3 MDS 仕様（9/36）

```
"digitalId":
"base64urlencoded(header).base64urlencoded(payload).base64urlencoded(signature)"
```

デジタルIDのヘッダーには次の情報がある

```
"alg": "RS256",
"typ": "JWT",
"x5c": "<Certificate of the FTM chip, If in case the chain of certificates are sent then the same will be ignored">
```

MOSIPではx5cにある最初の証明書がFTMルート証明書によって発行されたFTMチップの公開証明書であると見なされる
署名付きでないデジタルIDの例は、以下のとおりである

```
"digitalId": "base64urlencoded(payload)"
```

payloadはデジタルID JSONオブジェクトである

未登録のL0デバイスのデジタルIDには、署名が付かない。検出呼び出しを除くすべての場面で、チップキー（L1の場合）またはデバイスキー（L0の場合）によってデジタルIDが署名される

デジタルIDで許容される値

パラメーター	説明
serialNo	デバイスのシリアル番号。 この値はデバイスに印字されているものと同じであること（[物理 ID](#physical-id)を参照）
make	ブランド名。 この値はデバイスに印字されているものと同じであること（[物理 ID](#physical-id)を参照）
model	デバイスのモデル。 この値はデバイスに印字されているものと同じであること（[物理 ID](#physical-id)を参照）
type	現在、デバイスタイプとして許可されている値は、"Finger"、"Iris"、または"Face" である。 導入国の実装に応じてより多くのタイプを追加できる
deviceSubType	デバイスサブタイプはデバイスタイプに基づく 指紋の場合 – "Slap"、"Single" または "Touchless" 虹彩の場合 – "Single" または "Double" 顔の場合 – "Full face"
deviceProvider	デバイスプロバイダーの名称。 デバイスプロバイダーは、導入国における法人であること
dateTime	このIDの発行時間。 これはタイムゾーン付きISO形式で表示されている

2.6.3 MDS 仕様（10/36）

5.3 キー

デバイスで使用されるキーとその説明のリスト

- デバイスキー

  各生体認証デバイスには、デバイス登録後に認証された秘密鍵が含まれる。このキーは、MOSIP導入国の要件に応じて頻繁にローテーションされる。デフォルトで、デバイスキーについては30日のキー・ローテーション・ポリシーが推奨される。デバイスキーは、正常登録時にデバイスプロバイダーによってFTM内に作成される。デバイスキーは生体認証情報の署名に使用される。署名と用途の詳細については、こちら。このキーはデバイスプロバイダーによって発行され、デバイスキーの証明書は、デバイス・プロバイダー・キーによって発行される。このデバイス・プロバイダー・キーは、デバイスプロバイダーの特定モデルの承認後にMOSIP導入国によって発行されるものである

- FTMキー

  FTMキーはIDのルートである。キーは製造/プロビジョニングの段階でFTMプロバイダーによって作成される。これは永続キーであり、ローテーションされることはない。このキーはデジタルIDの署名に使用される

- MOSIPキー

  MOSIPキーはMOSIP導入国が用意する公開鍵である。このキーは生体認証情報の暗号化に使用される。暗号化の詳細を以下にリストする。このキーは1年ごとにローテーションすることを推奨する

6. デバイスサービス – 通信インターフェース

本セクションでは、デバイスの構築および当該デバイスとの通信に必要な、生体認証デバイスの接続性、アクセス性、検出能、およびプロトコルの詳細情報について説明する

デバイスは次のAPIセットにのみ従って実装すること。すべてのAPIは物理レイヤーおよびオペレーティングシステムとは独立であり、呼び出しはオペレーティングシステムごとに異なる。仕様内にオペレーティングシステム名を記しているが、指定していないオペレーティングシステムについては同様のテクノロジーを使用できる。デバイスサービスはデバイスがローカルにホストに接続されていることを保証することが期待される

6.1 デバイス検出（Device Discovery）

デバイス検出は、アプリケーションがシステム内のMOSIP準拠デバイスを特定するために使用する。プロトコルは、仕様に必要な抽象化がすべて施されたシンプルなプラグ・アンド・プレイとして設計されている

6.1.1 デバイス検出リクエスト

```
{
 "type": "type of the device"
}
```

6.1.2 デバイス検出リクエストで許容される値

- タイプ – "Biometric Device"、"Finger"、"Face"、"Iris"

"Biometric Device" – 特別なタイプであり、生体認証デバイスを探索する場合に使用する

6.1.3 デバイス検出レスポンス

```
[
 {
  "deviceId": "Internal ID",
  "deviceStatus": "Device status",
  "certification": "Certification level",
  "serviceVersion": "Device service version",
  "deviceSubId": ["Array of supported device sub Ids"],
  "callbackId": "Base URL to reach to the device",
  "digitalId": "Unsigned Digital ID of the device",
  "deviceCode": "A unique code given by MOSIP after successful registration",
  "specVersion": ["Array of supported MDS specification version"],
  "purpose": "Auth  or Registration or empty if not registered",
  "error": {
   "errorCode": "101",
   "errorInfo": "Invalid JSON Value Type For Discovery.."
  }
 },
  ...
]
```

2.6.3 MDS 仕様（11/36）

6.1.4 デバイス検出レスポンスで許容される値

パラメーター	説明
deviceStatus	許容される値は、 "Ready"、"Busy"、"Not Ready"、または "Not Registered"
certification	許容される値は、認証レベルに応じて "L0"、"L1"、または "L2"
serviceVersion	サポートするMDS仕様のバージョン
deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID
deviceSubId	許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDが不明な場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
callbackId	これはOSによって異なる。 LinuxおよびWindowsオペレーティングシステムの場合、HTTP URLである。 Androidの場合、これはインテント名である。 iOSでは、URLスキームである。 コールバックURLはベースURLとして将来のリクエストよりも優先される
digitalId	デジタルID定義どおりのデジタルIDだが署名はなされない
deviceCode	正常登録後にMOSIPによって指定される一意のコード
specVersion	サポートされるMDS仕様バージョンの配列
purpose	MOSIPエコシステム内のデバイスの目的。許容される値は、"Auth" または "Registration"
error	本ドキュメントの「エラー」セクションに定義されるエラー
error.errorCode	「エラーコード」セクションに定義される標準エラーコード
error.errorInfo	エラーの説明。エンドユーザーに表示してもよい。複数言語をサポート

2.6.3 MDS 仕様（12/36）

- レスポンスは、1つのデバイスで複数の生体認証オプションを列挙させることができる配列である
- サービスは、タイプパラメーターがデバイスのタイプに一致するか、タイプパラメーターが "Biometric Device" である場合にのみ応答すること
- このレスポンスは、レスポンス内に表示されているように直接JSON形式である

6.1.5 Windows/Linux

すべてのデバイスAPIはHTTP仕様に基づく。デバイスには常に4501から4600までの範囲の使用可能ポートのいずれかが割り当てられている。割り当てに使用するIPアドレスはlocalhostではなく 127.0.0.1 でなければならない

MOSIPデバイスへのアクセスをリクエストするアプリケーションは、サポートされたポート範囲にHTTPリクエストを送信することでMOSIPデバイスを検出できる。以降では、このポートをdevice_service_portとする

HTTPリクエスト：

```
MOSIPDISC http://127.0.0.1:<device_service_port>/device
HOST: 127.0.0.1: <device_service_port>
EXT: <app name>
```

HTTPレスポンス:

```
HTTP/1.1 200 OK
CACHE-CONTROL:no-store
LOCATION:http://127.0.0.1:<device_service_port>
Content-Length: length in bytes of the body
Content-Type: application/json
Connection: Closed
```

- どちらの場合もペイロードはJSONであり、本文の一部である
- CallbackIdには http://127.0.0.1:<device_service_port> を設定する。したがって、呼び出し元はそれぞれのHTTP verb/メソッドとURLを使用してサービスを呼び出す

2.6.3 MDS 仕様（13/36）

6.1.6 Android

Androidデバイス上のすべてのデバイスはインテント "io.mosip.device" をリッスンする
このインテントで呼び出されると、デバイスはそれぞれのタイプでフィルタリングされたJSONレスポンスで応答することが期待される

Androidでは、CallbackIdにはappIdが設定される。したがって、呼び出し元はインテント "appId.Info" または "appId.Capture" を作成する

6.1.7 iOS

iOSデバイスは以下のURLスキームに応答する

```
MOSIPDISC://<call-back-app-url>?ext=<caller app name>&type=<type as defined in mosip device request>
```

MOSIP準拠デバイスのサービスアプリが存在する場合、URLでサービスが立ち上がる。相手サービスは、base64エンコードされたJSONをキーデータのURLパラメーターとし、call-back-app-urlを使用して呼び出し元に応答を返さなければならない

- iOSでは、複数のアプリを同じURLスキームに登録する場合の制約がある
- CallbackIdにはデバイスサービスアプリ名が設定される。したがって、呼び出し元はURLスキームとしてappnameInfoまたはappnameCaptureを呼び出す必要がある

6.2 デバイス情報（Device Information）

デバイス情報APIは、アプリケーションがMOSIP準拠デバイスとそのステータスを特定するために使用する

6.2.1 デバイス情報リクエスト

適用なし

6.2.2 デバイス情報リクエストで許容される値

適用なし

2.6.3 MDS 仕様（14/36）

6.2.3 デバイス情報レスポンス

```
[
 {
  "deviceInfo": {
   "deviceStatus": "Current status",
   "deviceId": "Internal ID",
   "firmware": "Firmware version",
   "certification": "Certification level",
   "serviceVersion": "Device service version",
   "deviceSubId": ["Array of supported device sub Ids"],
   "callbackId": "Baseurl to reach to the device",
   "digitalId": "Signed digital id as described in the digital id section of this document.",
   "deviceCode": "A unique code given by MOSIP after successful registration",
   "env": "Target environment",
   "purpose": "Auth  or Registration",
   "specVersion": ["Array of supported MDS specification version"],
  },
  "error": {
   "errorCode": "101",
   "errorInfo": "Invalid JSON Value "
  }
 }
 ...
]
```

結果として生成されるJSONは "Foundational Trust Module" 識別キーを使用したJSON Webシグネチャで署名され、このデータはデバイスの基礎識別情報となる。あらゆるMOSIP準拠デバイスにはファンデーショナル・トラスト・モジュールが必須である

したがって、APIは次のフォーマットで応答を返す

```
[
 {
  "deviceInfo": "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)"
  "error": {
   "errorCode": "100",
   "errorInfo": "Device not registered. In this case the device info will be only base64urlencode(payload)"
  }
 }
]
```

6.2.4 デバイス情報レスポンスで許容される値

パラメーター	説明
deviceInfo	deviceInfoオブジェクトはJSON Webトークン（JWT）として送信される。 デバイスが登録されていない場合、deviceInfoは署名されない。 デバイスが登録されている場合、deviceInfoはデバイスキーを使用して署名される
deviceInfo.deviceStatus	これはデバイスのステータスである。 許容される値は、"Ready"、"Busy"、"Not Ready"、または "Not Registered"
deviceInfo.deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID
deviceInfo.firmware	ファームウェアの厳密なバージョン。 L0の場合はserviceVersionと同じになる
deviceInfo.certification	許容される値は、認証レベルに応じて "L0"、"L1"、または "L2"
deviceInfo.serviceVersion	サポートするMDS仕様のバージョン
deviceInfo.deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID

2.6.3 MDS 仕様（15/36）

パラメーター	説明
deviceSubId	許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDが不明な場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
deviceInfo.callbackId	これはOSによって異なる。 LinuxおよびWindowsオペレーティングシステムの場合、HTTP URLである。 Androidの場合、これはインテント名である。 iOSでは、URLスキームである。 コールバックURLはベースURLとして将来のリクエストよりも優先される
deviceInfo.digitalId	デジタルIDの定義に従うデジタルID L0デバイスが登録されていない場合、デジタルIDは署名されない。 L0デバイスが登録されている場合、デバイスキーを使用してデジタルIDが署名される。 L1デバイスの場合、デジタルIDはFTMキーを使用して署名される
deviceInfo.env	ターゲット環境。 デバイスが登録されていない場合、環境は "None" である。 デバイスが登録されている場合、デバイスが登録されている環境が送信される。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"
deviceInfo.purpose	MOSIPエコシステム内のデバイスの目的。 デバイスが登録されていない場合、空である。 許容される値は、"Auth" または "Registration"
deviceInfo.specVersion	サポートされるMDS仕様バージョンの配列
error	本ドキュメントの「エラー」セクションに定義されるエラー
error.errorCode	「エラーコード」セクションに定義される標準エラーコード
error.errorInfo	エラーの説明。エンドユーザーに表示してもよい。複数言語をサポート

2.6.3 MDS 仕様（16/36）

- レスポンスは、1つのデバイスで複数の生体認証オプションを列挙させることができる配列である
- サービスは、タイプパラメーターがデバイスのタイプに一致するか、タイプパラメーターが "Biometric Device" である場合にのみ応答すること

6.2.5 Windows/Linux

MOSIPデバイスの詳細情報をリクエストするアプリケーションは、サポートされたポート範囲にHTTPリクエストを送信することで情報を取得できる。デバイスには常に4501から4600までの範囲の使用可能ポートのいずれかが割り当てられている。割り当てに使用するIPアドレスはlocalhostではなく 127.0.0.1 でなければならない

HTTPリクエスト：

```
MOSIPDINFO http://127.0.0.1:<device_service_port>/info
HOST: 127.0.0.1:<device_service_port>
EXT: <app name>
```

HTTPレスポンス:

```
HTTP/1.1 200 OK
CACHE-CONTROL:no-store
LOCATION:http://127.0.0.1:<device_service_port>
Content-Length: length in bytes of the body
Content-Type: application/json
Connection: Closed
```

どちらの場合もペイロードはJSONであり、本文の一部である

6.2.6 Android

Androidデバイス上にある場合、インテント "appId.Info" をリッスンする

このインテントを呼び出すと、デバイスはそれぞれのタイプでフィルタリングされたJSONレスポンスで応答することが期待される

6.2.7 iOS

iOSデバイス上にある場合、以下のURLスキームに応答する

```
APPIDINFO://<call-back-app-url>?ext=<caller app name>&type=<type as defined in mosip device request>
```

MOSIP準拠デバイスのサービスアプリが存在する場合、URLでサービスが立ち上がる。相手サービスは、base64エンコードされたJSONをキーデータのURLパラメーターとし、call-back-app-urlを使用して呼び出し元に応答を返さなければならない

iOSでは、複数のアプリを同じURLスキームに登録する場合の制約がある

6.3 キャプチャ（Capture）

キャプチャリクエストは、アプリケーションがMOSIP準拠デバイスから生体認証情報を取得するために使用する。キャプチャ呼び出しでは一度に1つの呼び出しのみが応答を得る。したがって、呼び出し中に別の呼び出しが行われた場合は、ステータスが "Busy" のデバイス詳細情報が送信される

6.3.1 キャプチャリクエスト

```
{
  "env": "Target environment",
  "purpose": "Auth  or Registration",
  "specVersion": "Expected version of the MDS spec",
  "timeout" : "Timeout for capture",
  "captureTime": "Time of capture request in ISO format including timezone",
  "domainUri": "URI of the auth server",
  "transactionId": "Transaction Id for the current capture",
  "bio": [
    {
```

2.6.3 MDS 仕様（17/36）

```
      "type": "Type of the biometric data",
      "count":  "Finger/Iris count, in case of face max is set to 1",
      "bioSubType": ["Array of subtypes"],
      "requestedScore": "Expected quality score that should match to complete a successful capture",
      "deviceId": "Internal Id",
      "deviceSubId": "Specific Device Sub Id",
      "previousHash": "Hash of the previous block"
    }
  ],
  customOpts: {
    //Max of 50 key value pair. This is so that vendor specific parameters can be sent if necessary. The values cannot be hard coded and have to be configured by the apps server and should be modifiable upon need by the applications. Vendors are free to include additional parameters and fine-tuning parameters. None of these values should go undocumented by the vendor. No sensitive data should be available in the customOpts.
  }
}
```

countの値は指紋および虹彩のbioSubTypeの数で決まる。顔認証の場合、bioSubTypeは存在しないがcountは "1" とすること

6.3.2 キャプチャリクエストで許容される値

パラメーター	説明
env	ターゲット環境。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"
purpose	MOSIPエコシステム内のデバイスの目的。 デバイスが登録されていない場合、空である。 許容される値は、"Auth" または "Registration"
specVersion	期待されるMDS仕様のバージョン
timeout	キャプチャまでにアプリが待機する最大時間。 APIはtimeoutまでに最大フレームで応答を返すことが期待される。 timeoutはすべてミリ秒で表される
captureTime	キャプチャした時刻をタイムゾーン付きISO形式で表示したもの。 時間はリクエストを送信するアプリケーションごとに設定される
domainUri	認証サーバーのURI。 これを使用して複数のプロバイダー、導入国、連合体を連携させることができる
transactionId	トランザクションの一意なID。 これはサービスを提供するアプリケーションへの内部IDである。 認証ごとに異なるIDを使用すること。 したがって、認証後にトランザクションが失敗した場合でも、この番号は一意である
bio.type	許容される値は、"Finger"、"Iris"、または "Face"
bio.count	指定したタイプで収集される生体認証データの数。 デバイスを検証し、この番号がデバイスがキャプチャする生体認証情報のタイプに合っていることを確認すること

2.6.3 MDS 仕様（18/36）

パラメーター	説明
bio.bioSubType	指紋の場合：["Left IndexFinger", "Left MiddleFinger", "Left RingFinger", "Left LittleFinger", "Left Thumb", "Right IndexFinger", "Right MiddleFinger", "Right RingFinger", "Right LittleFinger", "Right Thumb", "UNKNOWN"] 虹彩の場合：["Left", "Right", "UNKNOWN"] 顔の場合：bioSubTypeなし
bio.requestedScore	指定された品質スコアに達したら、生体認証デバイスで画像が自動的にキャプチャされる
bio.deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID
bio.deviceSubId	許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDがわからない場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
bio.previousHash	初回のキャプチャではpreviousHashは空のUTF-8文字列のハッシュである。 2回目のキャプチャ以降は前回の取り込みのハッシュ（16進エンコードとして）が入力として使用される。 これを使用して、モダリティをまたぐすべてのキャプチャがチェーン化され、すべての取り込みが同じトランザクションで同じ時間帯に発生するようになっている
customOpts	必要に応じてキー値ペアの送信に使用できるよう、デバイスベンダーが追加のパラメーターの送信を希望する場合がある。 値をハードコーディングしてはならない。値はアプリサーバーによって設定し、必要に応じてアプリケーションで変更する必要がある。 ベンダーはパラメーターを追加してプロセスを微調整しても構わない。 ベンダーはこれらの値を1つ残さずドキュメント化すること。 customOptsには機密データを使用しないこと

6.3.3 キャプチャレスポンス

```
{
  "biometrics": [
    {
      "specVersion": "MDS spec version",
      "data": {
        "digitalId": "digital Id as described in this document",
        "deviceCode": "A unique code given by MOSIP after successful registration",
        "deviceServiceVersion": "MDS version",
        "bioType": "Finger",
        "bioSubType": "UNKNOWN",
        "purpose": "Auth  or Registration",
        "env": "Target environment",
        "domainUri": "URI of the auth server",
        "bioValue": "Encrypted with session key and base64urlencoded biometric data",
        "transactionId": "Unique transaction id",
        "timestamp": "ISO format datetime with time zone",
        "requestedScore": "Floating point number to represent the minimum required score for the capture",
        "qualityScore": "Floating point number representing the score for the current capture"
      },
      "hash": "sha256(sha256 hash in hex format of the previous data block + sha256 hash in hex format of the current data block before encryption)",
      "sessionKey": "encrypted with MOSIP public key (dynamically selected based on the uri) and encoded session key biometric",
```

2.6.3 MDS 仕様（19/36）

```
    "thumbprint": "SHA256 representation of thumbprint of the certificate that was used for encryption of session key. All texts to be treated as uppercase without any spaces or hyphens",
    "error": {
      "errorCode": "101",
      "errorInfo": "Invalid JSON Value"
    }
  },
  {
    "specVersion" : "MDS spec version",
    "data": {
      "digitalId": "Digital Id as described in this document",
      "deviceCode": "A unique code given by MOSIP after successful registration",
      "deviceServiceVersion": "MDS version",
      "bioType": "Finger",
      "bioSubType": "Left IndexFinger",
      "purpose": "Auth  or Registration",
      "env": "target environment",
      "domainUri": "uri of the auth server",
      "bioValue": "encrypted with session key and base64urlencoded biometric data",
      "transactionId": "unique transaction id",
      "timestamp": "ISO Format date time with timezone",
      "requestedScore": "Floating point number to represent the minimum required score for the capture",
```

```
      "qualityScore": "Floating point number representing the score for the current capture"
    },
    "hash": "sha256(sha256 hash in hex format of the previous data block + sha256 hash in hex format of the current data block before encryption)",
    "sessionKey": "encrypted with MOSIP public key and encoded session key biometric",
    "thumbprint": "SHA256 representation of thumbprint of the certificate that was used for encryption of session key. All texts to be treated as uppercase without any spaces or hyphens",
    "error": {
      "errorCode": "101",
      "errorInfo": "Invalid JSON Value"
    }
  }
 ]
}
```

6.3.4 キャプチャレスポンスで許容される値

パラメーター	説明
specVersion	レスポンスの生成に使用されたMDS仕様のバージョン
data	データオブジェクトはJSON Webトークン（JWT）として送信される。 データブロックはデバイスキーを使用して署名される
data.digitalId	デジタルIDの定義に従うデジタルID（JWT形式）。 L0デバイスの場合、デジタルIDはデバイスキーを使用して署名される。 L1、L2デバイスの場合、デジタルIDはFTMキーを使用して署名される
data.deviceCode	正常登録後にMOSIPによって指定される一意のコード

2.6.3 MDS 仕様（20/36）

パラメーター	説明
data.deviceServiceVersion	MDSのバージョン
data.bioType	許容される値は、"Finger"、"Iris"、または "Face"
data.bioSubType	指紋の場合：["Left IndexFinger", "Left MiddleFinger", "Left RingFinger", "Left LittleFinger", "Left Thumb", "Right IndexFinger", "Right MiddleFinger", "Right RingFinger", "Right LittleFinger", "Right Thumb", "UNKNOWN"] 虹彩の場合：["Left", "Right", "UNKNOWN"] 顔の場合：bioSubTypeなし
data.purpose	MOSIPエコシステム内のデバイスの目的。 許容される値は、"Auth"
data.env	ターゲット環境。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"
data.domainUri	認証サーバーのURI。 これを使用して複数のプロバイダー、導入国、連合体を連携させることができる
data.bioValue	生体認証データはランダムな対称（AES GCM）鍵で暗号化され、base-64-URLエンコードされる。 bioValueの対称鍵暗号化では、(biometrics.data.timestamp XOR transactoinId)が計算され、結果の最後の16バイトがaadに、最後の12バイトがIV(salt) にセットされる。 暗号化の詳細については、認証についてのドキュメントを参照
data.transactionId	リクエストで送信された一意のトランザクションID
data.timestamp	生体認証デバイスに保持された時刻。 注記：デバイスで正しい時刻が保持されるよう、生体認証デバイスは、その時刻を管理サーバーから定期的に送信される時刻と同期することが期待される
data.requestedScore	キャプチャに必要な最小スコアを表す浮動小数点数
data.qualityScore	現在のキャプチャのスコアを表す浮動小数点数
hash	暗号化前に現在のデータブロックの16進エンコードのSHA256ハッシュと連結された、リクエストオブジェクトのpreviousHash 属性の値または前のデータブロックのハッシュ属性の値（それぞれの単一データブロックのチェーンに使用）

2.6.3 MDS 仕様（21/36）

パラメーター	説明
sessionKey	セッションキー（bioValueの暗号化に使用する）はMOSIP公開証明書を使用して RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDINGアルゴリズムで暗号化され、その後 base64-URLエンコードされる
thumbprint	セッションキーの暗号化に使用した証明書のサムプリントのSHA256表現。 すべてのテキストはスペースなし、ハイフンなしで大文字として扱われる
error	本ドキュメントの「エラー」セクションに定義されるエラー
error.errorCode	「エラーコード」セクションに定義される標準エラーコード
error.errorInfo	エラーの説明。エンドユーザーに表示してもよい。複数言語をサポート

データオブジェクト全体はJWT形式で送信される。したがって、データオブジェクトは以下のようになる

```
"data" : "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)
payload - is defined as the entire byte array of data block.
```

6.3.5 Windows/Linux

MOSIPデバイスからの生体認証情報のキャプチャをリクエストするアプリケーションは、サポートされたポート範囲にHTTP リクエストを送信することで取得できる

HTTPリクエスト：

```
CAPTURE [http://127.0.0.1:<device_service_port>/capture](http://127.0.0.1/capture)
HOST: 127.0.0.1: <apps port>
EXT: <app name>
```

HTTPレスポンス:

```
HTTP/1.1 200 OK
CACHE-CONTROL:no-store
LOCATION:[http://127.0.0.1](http://127.0.0.1):<device_service_port>
Content-Length: length in bytes of the body
Content-Type: application/json
Connection: Closed
```

どちらの場合もペイロードはJSONであり、本文の一部である

6.3.6 Android

Androidデバイス上のすべてのデバイスはインテント "appid.capture" をリッスンする。このインテントでは、 デバイスはそれぞれのタイプでフィルタリングされたJSONレスポンスで応答することが期待される

6.3.7 iOS

iOSデバイス上のすべてのデバイスは以下のURLスキームに応答する

```
APPIDCAPTURE://<call-back-app-url>?ext=<caller app name>&type=<type as defined in mosip device request>
```

MOSIP準拠デバイスのサービスアプリが存在する場合、URLでサービスが立ち上がる。相手サービスは、base64エンコード されたJSONをキーデータのURLパラメーターとしてcall-back-app-urlを使用して呼び出し元に応答を返さなければ ならない

2.6.3 MDS 仕様（22/36）

6.4 デバイスストリーム（Device Stream）

デバイスは、ストリームチャンネルを開いてライブ動画ストリームを送信する。生体認証情報を収集する補助操作がある場合に役立つ。ストリームのAPIは登録環境でのみ使用できることに注意すること

登録モジュールと互換性のあるデバイスのみが使用する。このAPIは、目的が "Registration" の登録済デバイスのみに表示される

6.4.1 デバイスストリーム・リクエスト

```
{
  "deviceId": "Internal Id",
  "deviceSubId": "Specific device sub Id",
  "timeout": "Timeout for stream"
}
```

6.4.2 デバイスストリーム・リクエストで許容される値

パラメーター	説明
deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID
deviceSubId	許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDが不明な場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
timeout	この時間の後にストリームを終了する最大の時間。 これはオプションのパラメーターであり、デフォルトで値は5分である。 timeoutはすべてミリ秒で表される

2.6.3 MDS 仕様（23/36）

6.4.3 デバイスストリーム・レスポンス

M-JPEGを使用した3フレーム/秒以上の品質のライブ動画ストリーム

プレビューは品質マーキングとセグメントマーキングを有すること。プレビューはユーザー画面へのエラーメッセージ表示にも使用される。すべてのエラーメッセージはローカライズされていること

6.4.4 デバイスストリームにおけるエラーレスポンス

```
{
  "error": {
    "errorCode": "202",
    "errorInfo": "No Device Connected."
  }
}
```

6.4.5 Windows/Linux

MOSIPデバイスの詳細情報をリクエストするアプリケーションは、サポートされたポート範囲にHTTPリクエストを送信することで取得できる

HTTPリクエスト：

```
STREAM   http://127.0.0.1:<device_service_port>/stream
HOST: 127.0.0.1: <apps port>
EXT: <app name>
```

HTTPレスポンス: フレームのHTTPチャンクが表示される。最低3フレーム/秒

6.4.6 Android

ストリーミングはサポートされない

6.4.7 iOS

ストリーミングはサポートされない

6.5 登録キャプチャ（Registration Capture）

登録クライアントアプリケーションはデバイスを検出する。デバイスが検出されると、デバイス情報APIでデバイスのステータスが取得される。登録時、登録クライアントがRECAPTURE APIを送信すると、応答として実際の生体認証データがデジタル署名付き非暗号化形式で提供される。デバイス登録キャプチャAPIが呼び出されたときは、ストリームにフレームを追加してはならない。デバイスは画像をISO形式で送信する

requestedScore は1～100の範囲である。したがって、指が4本あった場合、全部の平均がキャプチャしきい値とみなされる。デバイスはキャプチャ時、リクエストされたスコアに満たない場合であっても、常に最大限に可能なフレームレートで送信する

登録モジュールと互換性のあるデバイスがAPIを使用する。本APIは、認証と互換性のあるデバイスではサポートされないこと

6.5.1 登録キャプチャリクエスト

```
{
  "env":  "Target environment",
  "purpose": "Auth  or Registration",
  "specVersion": "Expected MDS spec version",
  "timeout": "Timeout for registration capture",
  "captureTime": "Time of capture request in ISO format including timezone",
  "transactionId": "Transaction Id for the current capture",
  "bio": [
    {
      "type": "Type of the biometric data",
```

2.6.3 MDS 仕様（24/36）

```
      "count":  "Finger/Iris count, in case of face max is set to 1",
      "bioSubType": ["Array of subtypes"], //Optional
      "exception": ["Finger or Iris to be excluded"],
      "requestedScore": "Expected quality score that should match to complete a successful capture.",
      "deviceId": "Internal Id",
      "deviceSubId": "Specific device Id",
      "previousHash": "Hash of the previous block"
    }
  ],
  customOpts: {
    //max of 50 key value pair. This is so that vendor specific parameters can be sent if necessary. The values cannot be hard coded and have to be configured by the apps server and should be modifiable upon need by the applications. Vendors are free to include additional parameters and fine-tuning parameters. None of these values should go undocumented by the vendor. No sensitive data should be available in the customOpts.
  }
}
```

6.5.2 登録キャプチャリクエストで許容される値

パラメーター	説明
env	ターゲット環境。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"
purpose	MOSIPエコシステム内のデバイスの目的。 デバイスが登録されていない場合、空である。 許容される値は、"Auth" または "Registration"
specVersion	期待されるMDS仕様のバージョン
timeout	キャプチャまでにアプリが待機する最大時間。 APIはtimeoutまでに最大フレームで応答を返すことが期待される。 timeoutはすべてミリ秒で表される
captureTime	キャプチャした時刻をタイムゾーン付きISO形式で表示したもの。 時間はリクエストを送信するアプリケーションごとに設定される
transactionId	トランザクションの一意なID。 これはサービスを提供するアプリケーションへの内部IDである。 認証ごとに異なるIDを使用すること。 したがって、認証後にトランザクションが失敗した場合でも、この番号は一意である
bio.type	許容される値は、"Finger"、"Iris"、または "Face"
bio.count	指定したタイプで収集される生体認証データの数。 デバイスを検証し、この番号がデバイスがキャプチャする生体認証情報のタイプに合っていることを確認すること
bio.bioSubType	各生体認証情報タイプのbioSubTypeの配列 指紋の場合：["Left IndexFinger", "Left MiddleFinger", "Left RingFinger", "Left LittleFinger", "Left Thumb", "Right IndexFinger", "Right MiddleFinger", "Right RingFinger", "Right LittleFinger", "Right Thumb", "UNKNOWN"] 虹彩の場合：["Left", "Right", "UNKNOWN"] 顔の場合：bioSubTypeなし これはオプションのパラメーターである
bio.exception	これは配列であり、すべての例外にマークが付いている。 顔認証時に例外が発生した場合、上記の例外写真仕様に従う。 指紋の場合：["Left IndexFinger", "Left MiddleFinger", "Left RingFinger", "Left LittleFinger", "Left Thumb", "Right IndexFinger", "Right MiddleFinger", "Right RingFinger", "Right LittleFinger", "Right Thumb"] 虹彩の場合：["Left", "Right"]

2.6.3 MDS 仕様（25/36）

パラメーター	説明
bio.requestedScore	指定された品質スコアに達したら、生体認証デバイスで画像が自動的にキャプチャされる
bio.deviceId	デバイスサービス内の実際の生体認証デバイスを特定する内部ID
bio.deviceSubId	許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDが不明な場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
bio.previousHash	初回のキャプチャではpreviousHashは空のUTF-8文字列のハッシュである。 2回目のキャプチャ以降は前回の取り込みのハッシュ（16進エンコードとして）が入力として使用される。 これを使用して、モダリティをまたぐすべてのキャプチャがチェーン化され、すべての取り込みが同じトランザクションで同じ時間帯に発生するようになっている
customOpts	必要に応じてキー値ペアの送信に使用できるよう、デバイスベンダーが追加のパラメーターの送信を希望する場合がある。 値をハードコーディングしてはならない。値はアプリサーバーによって設定し、必要に応じてアプリケーションで変更する必要がある。 ベンダーはパラメーターを追加してプロセスを微調整しても構わない。 ベンダーはこれらの値を1つ残さずドキュメント化すること。 customOptsには機密データを使用しないこと

6.5.3 登録キャプチャレスポンス

```
{
  "biometrics": [
    {
      "specVersion": "MDS Spec version",
      "data": {
        "digitalId": "Digital id of the device as per the Digital Id definition..",
```

2.6.3 MDS 仕様（26/36）

```
      "bioType": "Biometric type",
      "deviceCode": "A unique code given by MOSIP after successful registration",
      "deviceServiceVersion": "MDS version",
      "bioSubType": "Left IndexFinger",
      "purpose": "Auth  or Registration",
      "env": "Target environment",
      "bioValue": "base64urlencoded biometrics (ISO format)",
      "transactionId": "Unique transaction id sent in request",
      "timestamp": "2019-02-15T10:01:57.086+05:30",
      "requestedScore": "Floating point number to represent the minimum required score for the capture. This ranges from
0-100.",
      "qualityScore": "Floating point number representing the score for the current capture. This ranges from 0-100."
    },
    "hash": "sha256(sha256 hash in hex format of the previous data block + sha256 hash in hex format of the current data
block)",
    "error": {
      "errorCode": "101",
      "errorInfo": "Invalid JSON Value Type For Discovery.. ex: {type: 'Biometric Device' or 'Finger' or 'Face' or 'Iris' } "
    }
  },
  {
    "specVersion" : "MDS Spec version",
    "data": {
      "deviceCode": "A unique code given by MOSIP after successful registration",
```

```
      "bioType" : "Finger",
      "digitalId": "Digital id of the device as per the Digital Id definition.",
      "deviceServiceVersion": "MDS version",
      "bioSubType": "Left MiddleFinger",
      "purpose": "Auth  or Registration",
      "env":  "Target environment",
      "bioValue": "base64urlencoded extracted biometric (ISO format)",
      "transactionId": "Unique transaction id sent in request",
      "timestamp": "2019-02-15T10:01:57.086+05:30",
      "requestedScore": "Floating point number to represent the minimum required score for the capture. This ranges from
0-100",
      "qualityScore": "Floating point number representing the score for the current capture. This ranges from 0-100"
    },
    "hash": "sha256(sha256 hash in hex format of the previous data block + sha256 hash in hex format of the current data
block before encryption)",
    "error": {
      "errorCode": "101",
      "errorInfo": "Invalid JSON Value Type For Discovery.. ex: {type: 'Biometric Device' or 'Finger' or 'Face' or 'Iris' }"
    }
  }
 ]
}
```

2.6.3 MDS 仕様（27/36）

6.5.4 登録キャプチャレスポンスで許容される値

パラメーター	説明
specVersion	レスポンスの生成に使用されたMDS仕様のバージョン
data	データオブジェクトはJSON Webトークン（JWT）として送信される。 データブロックはデバイスキーを使用して署名される
data.bioType	許容される値は、"Finger"、"Iris"、または "Face"
data.digitalId	デジタルIDの定義に従うデジタルID（JWT形式）。 L0デバイスの場合、デジタルIDはデバイスキーを使用して署名される。 L1、L2デバイスの場合、デジタルIDはFTMキーを使用して署名される
data.bioSubType	指紋の場合：["Left IndexFinger", "Left MiddleFinger", "Left RingFinger", "Left LittleFinger", "Left Thumb", "Right IndexFinger", "Right MiddleFinger", "Right RingFinger", "Right LittleFinger", "Right Thumb", "UNKNOWN"] 虹彩の場合：["Left", "Right", "UNKNOWN"] 顔の場合：bioSubTypeなし
data.deviceServiceVersion	MDSのバージョン
data.env	ターゲット環境。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"
data.purpose	MOSIPエコシステム内のデバイスの目的。 許容される値は、"Auth" または "Registration"
data.bioValue	Base64-URL-encoded形式の生体情報（ISO形式）
data.transactionId	リクエストで送信された一意のトランザクションID
data.timestamp	生体認証デバイスに保持された時刻 注記：デバイスで正しい時刻が保持されるよう、生体認証デバイスは、その時刻を管理サーバーから定期的に送信される時刻と同期することが期待される

2.6.3 MDS 仕様（28/36）

パラメーター	説明
data.requestedScore	キャプチャに必要な最小スコアを表す浮動小数点数
data.qualityScore	現在のキャプチャのスコアを表す浮動小数点数
hash	暗号化前に現在のデータブロックの16進エンコードのSHA256ハッシュと連結された、リクエストオブジェクトのpreviousHash 属性の値または前のデータブロックのハッシュ属性の値（それぞれの単一データブロックのチェーンに使用）
error	本ドキュメントの「エラー」セクションに定義されるエラー
error.errorCode	「エラーコード」セクションに定義される標準エラーコード
error.errorInfo	エラーの説明。エンドユーザーに表示してもよい。複数言語をサポート

6.5.5 Windows/Linux

MOSIPデバイスの詳細情報をリクエストするアプリケーションは、サポートされたポート範囲にHTTPリクエストを送信することで取得できる

HTTPリクエスト：

```
RCAPTURE  http://127.0.0.1:<device_service_port>/capture
HOST: 127.0.0.1: <apps port>
EXT: <app name>
```

HTTPレスポンス: HTTPレスポンス

6.5.6 Android

登録キャプチャはサポートされない

6.5.7 iOS

登録キャプチャはサポートされない

7. デバイスサーバー

デバイスサーバーでは、デバイスを管理するための2つの外部デバイスAPIが開示されている。これらは、デバイスプロバイダーが作成した管理サーバーで使用される。本ドキュメントの後続のセクションを参照のこと

7.1 登録

MOSIPサーバーは以下のデバイス登録APIを提供する。このAPIはデバイスプロバイダーまたはそのパートナーの管理サーバーでホワイトリスト化されている

このAPIは、MOSIPサーバーによってデバイスプロバイダー向けに開示される

バージョン： v1

7.1.1 デバイス登録リクエストURL

POST https://{base_url}/v1/masterdata/registereddevices

7.1.2 デバイス登録リクエスト

```
{
  "id": "io.mosip.deviceregister",
  "request": {
    "deviceData": {
      "deviceId": "Unique Id to identify a biometric capture device",
      "purpose": "Auth or Registration. Can not be empty.",
      "deviceInfo": {
        "deviceSubId": "An array of sub Ids that are available",
        "certification":  "certification level",
        "digitalId": "Signed digital id of the device",
```

2.6.3 MDS 仕様（29/36）

```
      "firmware": "Firmware version",
      "deviceExpiry": "Device expiry date",
      "timestamp":  "ISO format datetime with timezone from device"
    },
    "foundationalTrustProviderId" : "Foundation trust provider Id, in case of L0 this is empty"
  }
 },
 "requesttime": "Current timestamp in ISO format from management server",
 "version": "Registration server api version as defined above"
}
```

7.1.3 登録リクエストで許容される値

パラメーター	説明
deviceData	デバイスのデータオブジェクトはJSON Webトークン（JWT）として送信される。 デバイスのデータブロックはデバイスプロバイダー証明書を使用して署名される
deviceData.deviceId	デバイスプロバイダーが使用してデバイスを特定する、一意のデバイスID。 デバイスプロバイダーがシリアル番号をすべてのデバイスで一意にしている場合、これはシリアル番号でもよい
purpose	MOSIPエコシステム内のデバイスの目的。 デバイスが登録されていない場合、空である。 許容される値は、"Auth" または "Registration"
deviceData.deviceInfo	デバイス情報オブジェクトはJSON Webトークン（JWT）として送信される。 デバイス情報ブロックはデバイスキーを使用して署名される

2.6.3 MDS 仕様（30/36）

パラメーター	説明
deviceInfo.deviceSubId	デバイスでサポートされているサブIDの配列。 許容される値は、1、2、または3。 デバイスサブIDを使用して、生体情報のキャプチャ要件に合ったスキャナー内の特定のモジュールを有効化する。 デバイスサブIDは、常に1で始まり、サブデバイスができるたびに1つずつ順に増加するシンプルなインデックスである。 指紋/虹彩の場合、左手/虹彩は1、右手/虹彩は2、親指2本/虹彩は3。 特定のデバイスサブIDが不明な場合、デバイスサブIDは0に設定すること（0は指紋入力装置には適用不可）
deviceInfo.certification	デバイスの証明書レベル。 許容される値は、L0、L1、またはL2
deviceInfo.digitalId	デジタルIDの定義に従うデジタルID。 L0デバイスの場合、デジタルIDはデバイスキーを使用して署名される。 L1デバイスの場合、デジタルIDはFTMキーを使用して署名される
deviceInfo.firmware	デバイスのファームウェアのバージョン
deviceInfo.deviceExpiry	デバイスの有効期限。 デバイスはこの有効期限を過ぎた後は動作せず、再登録はできない
deviceInfo.timestamp	リクエストが作成された時点のタイムスタンプ。 デバイス登録では、リクエストはこのタイムスタンプから5分以内にMOSIPに到達する必要がある
foundationalTrustProviderId	デバイスのチップを製造したファンデーション・トラスト・プロバイダーのID。 L0デバイスの場合、これは空である

L0デバイスの登録時は、デバイス内で生成されたキーを使用して署名し、公開鍵をx509エンコード仕様形式で送信すること。登録が完了した後、管理サーバーは登録呼び出しの応答として同じ公開鍵に対する証明書を発行すること

- デバイスデータ全体はJWT形式として送信される。したがって、以下のようになる

```
"deviceData" : base64urlencode(header).base64urlencode(payload).base64urlencode(signature)
```

- ペイロードはdeviceData内のオブジェクトである
- リクエストは管理サーバーでデバイスプロバイダーの鍵を使用して署名される

7.1.4 デバイス登録レスポンス

```
{
  "id": "io.mosip.deviceregister",
  "version": "Registration server API version as defined above",
  "responsetime": "ISO time format",
  "response": {
    "status":  "Registration status",
    "digitalId": "Digital id of the device a sent by the request",
    "deviceCode": "UUID RFC4122 Version 4 for the device issued by the mosip server",
    "timestamp": "Timestamp in ISO format",
    "env": "prod/development/stage"
  },
  "error": [
    {
      "errorCode": "Error code if registration fails. Remaining keys above are dropped in case of errors.",
      "message": "Description of the error code"
    }
  ]
}
```

2.6.3 MDS 仕様（31/36）

レスポンス全体はJWT形式として送信される。したがって、最終的なレスポンスは以下のようになる

```
"response" : base64urlencode(header).base64urlencode(payload).base64urlencode(signature)
```

7.2 デバイス登録レスポンスで許容される値

パラメーター	説明
response	レスポンスブロック全体はJWT形式で送信される。 MOSIPにより、その公開署名証明書を使用して署名される
response.status	これはデバイスのステータスである。 登録完了後、ステータスが "Registered" として送信される
response.digitalId	これはリクエストで送信されたものと同じデジタルIDである
response.deviceCode	これはMOSIPサーバーで登録後に発行されるデバイスコードである。 これはUUID RFC4122バージョン4形式となる デバイスが登録されると、デバイスにデバイスコードを設定しなければならない
response.timestamp	これはデバイスが登録された時刻のタイムスタンプである。 これはISO形式となる
response.env	デバイスが登録されたターゲット環境。 許容される値は、"Staging"、"Developer"、"Pre-Production"、または "Production"

レスポンスはデバイスに送信されること。デバイスはそのストレージ内にデバイスコードを安全に保存することが望ましい。デバイスコードはキャプチャ時に使用される

目的を指定してデバイスを登録した場合、登録完了後に変更することはできない。デバイスは特定のMOSIPプロセスのためにのみ使用できる

7.3 登録解除

MOSIPサーバーは以下のデバイス登録解除APIを提供する。このAPIはデバイスプロバイダーまたはそのパートナーの管理サーバーでホワイトリスト化されている

バージョン： v1

7.3.1 デバイス登録解除リクエストURL

POST https://{base_url}/v1/masterdata/device/deregister

7.3.2 デバイス登録解除リクエスト

```
{
  "id": "io.mosip.devicederegister",
  "version": "de-registration server api version as defined above",
  "request": {
    "device": {
      "deviceCode": "<device code>",
      "env": "<environment>"
    }
  }
  "requesttime": "current timestamp in ISO format"
}
```

リクエスト内のデバイスデータはJWT形式として送信される。したがって、最終的なリクエストは以下のようになる

```
"request": {
  "device" : "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)"
}
```

2.6.3 MDS 仕様（32/36）

7.3.3 デバイス登録解除レスポンス

```
{
  "id": "io.mosip.devicederegister",
  "version": "de-registration server api version as defined above",
  "responsetime": "iso time format",
  "response": {
    "status": "Success",
    "deviceCode": "<device code>",
    "env": "<environment>",
    "timestamp": "timestamp in ISO format"
  },
  "error": [
       {
     "errorCode" : "<error code if de-registration fails>",
     "message" : "<human readable description of the error code>"
    }
  ]
}
```

レスポンス全体はJWT形式として送信される。したがって、最終的なレスポンスは以下のようになる

```
"response" : "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)"
```

7.4 証明書

MOSIPサーバーは以下の暗号証明書取得APIを提供する。このAPIはデバイスプロバイダーまたはそのパートナーの管理サーバーでホワイトリスト化されている

7.4.1 暗号証明書取得リクエストURL

POST https://{base_url}/v1/masterdata/device/encryptioncertficates

バージョン： v1

7.4.2 暗号証明書取得リクエスト

```
{
  "id": "io.mosip.auth.country.certificate",
  "version": "certificate server api version as defined above",
  "request": {
    "data": {
      "env":  "target environment",
      "domainUri": "uri of the auth server"
    }
  },
  "requesttime": "current timestamp in ISO format"
}
```

リクエストはJWT形式として送信される。したがって、最終的なリクエストは以下のようになる

```
"request": {
  "data": "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)"
}
```

2.6.3 MDS 仕様（33/36）

7.4.3 暗号証明書取得リクエストで許容される値

```
env - Allowed values are Staging| Developer| Pre-Production | Production
domainUri - unique uri per auth providers. This can be used to federate across multiple providers or countries or unions.
```

7.4.4 暗号証明書レスポンス

```
{
  "id": "io.mosip.auth.country.certificate",
  "version": "certificate server api version as defined above",
  "responsetime": "iso time format",
  "response": [
    {
      "certificate": "base64encoded certificate as x509 V3 format"
    }
  ]
}
```

レスポンス全体はJWT形式として送信される。したがって、最終的なレスポンスは以下のようになる

```
"response" : "base64urlencode(header).base64urlencode(payload).base64urlencode(signature)"
```

8. 管理サーバーおよび管理クライアント

8.1 管理サーバーの機能と相互作用

管理サーバーには以下の目的がある

1. デバイスを検証して対応するデバイスプロバイダーが製造した純正デバイスであることを確認する。これはデバイス情報とファンデーショナル・トラスト・モジュールの証明書を使用して実施される
2. MOSIPデバイスサーバーに純正デバイスを登録する
3. エンドデバイスとサーバーで時刻を管理/同期する。同期する時刻は、デバイスで許容される信頼された時刻のみとする
4. エンドデバイス向けに以下のようなコマンドを発行できる
    1. デバイスの登録解除（デバイスキー）
    2. 遠隔でデバイスのメンテナンス、管理、サポート、アップグレードを行うためのデバイス情報収集
    3. デバイスプロバイダーの承認済デバイスの中心的リポジトリである
    4. HSM FIPS 140-2レベル3を使用して作成された鍵の安全なストレージ。これらのキーは登録時のデバイス証明書の発行に使用される。
       管理サーバーは、MOSIPソフトウェアの外部のデバイスプロバイダーによって作成され、ホストされる。MDSと管理サーバーの間の通信プロトコルはそれぞれのデバイスプロバイダーによって決定してよい。そのような通信は上記の指定された相互作用のみに制限すること。このサーバーにトランザクション情報を送信しないこと
    5. サーバーからクライアントデバイスに更新をプッシュする機能を有すること

8.2 管理クライアント

管理クライアントは、デバイスをそれぞれの管理サーバーに接続するインターフェースである。管理サーバーとそのクライアントの間の通信は、スケーラビリティ、堅牢性、パフォーマンス、セキュリティを備えるよう設計することが重要である。管理サーバーにはここに記載するよりもさらに多くの機能を追加できるが、その内容にかかわらず、基本的なセキュリティの目的は必ず常に満たしておくこと

2.6.3 MDS 仕様（34/36）

1. 多数のデバイスをよりよく効率的に取り扱うため、デバイスには管理サーバーに自動的に登録する機能があることが望ましい
2. サーバーが送受信するすべての通信は下記の特性に従うこと
    1. すべての通信は承認されたアルゴリズムでデジタル署名される
    2. サーバー宛てのすべての通信は、承認された公開鍵暗号（HTTPS – TLS1.2/1.3は承認されたアルゴリズムの1つと共に使用しなければならない）で暗号化される
    3. すべてのリクエストにはシグネチャ内にミリ秒まで表示されたISO形式のタイムスタンプが付与される
    4. やりとりするすべての通信には属性の1つとして署名付きデジタルIDを付与すること
    5. 自動登録にはデバイスを特定し、検証するための絶対的な方法があることが望ましい
    6. 管理クライアントはプラグ・アンド・プレイ・モデルでデバイスを検出できること
    7. キーローテーションはすべてサーバーからトリガすること
    8. 正しい管理サーバーと接続されているかどうかを検出する機能を有すること
    9. すべてのアップグレードは検証可能でなければならず、クライアントはソフトウェアのアップグレードを検証する機能を有すること
    10. ソフトウェアのダウングレードはすべて許可されないこと
    11. 生体認証情報をキャプチャする際にはAPIを露出しないこと。管理サーバーにはキャプチャリクエストをトリガする機能はないこと
    12. 生体認証データのログ収集は禁止される（暗号化フォーマットと非暗号化フォーマットのいずれも禁止）

9. コンプライアンス

L2認証デバイス/ L2デバイス – デバイスの信頼ゾーンの中で暗号化を実行でき、不正対抗機能を有すると認証されたデバイス。**L1認証デバイス/ L1デバイス** – デバイスの信頼ゾーンの中で仕様に従って暗号化を実行できると認定されたデバイス。**L0認証デバイス/ L0デバイス** – デバイスドライバーまたはMOSIPデバイスサービスの中で暗号化を実行できると認定されたデバイス

9.1 セキュアプロビジョニング

FTMとデバイスプロバイダーの両方に、セキュアプロビジョニングが適用できる

1. デバイスとFTMには、作成や複製の不正な試みから保護するメカニズムがあること
2. デバイスとFTMの信頼性機能は、各MOSIP導入国によって認証された安全な施設でプログラムを行うこと
3. 組織には、暗号的に有効で繰り返し可能なプロセスを使用して、FTMに組み込まれた機能とMOSIPのデバイスに組み込まれた機能を切り分けるメカニズムがあること
4. FTMまたはデバイス内にあるすべてのデバッグオプションは、永続的にオフにすること
5. プロビジョニングに必要なすべての鍵の作成は、FIPS 140-2レベル3以上のデバイスを使用して自動的に行うこと。いずれの個人、グループ、または組織もこの挙動に影響を与えるメカニズムを有しないこと
6. デバイス/FTMがセキュアプロビジョニングを行う場所を離れる前に、必要な信頼性がすべて設定され、再プログラミング不能になっていること

9.2 コンプライアンスレベル

API	互換性
デバイス検出	L0/L1/L2
デバイス情報	L0/L1/L2
キャプチャ	L1/L2
登録キャプチャ	L0/L1/L2

10. 暗号化

サポートされるアルゴリズム：

2.6.3 MDS 仕様（35/36）

用途	アルゴリズム	鍵長	ストレージ
生体認証情報の暗号化 – セッションキー	AES	最大256	ストレージなし、鍵はFTM内TRNG/DRBGで生成
FTM外部の暗号化セッションキーデータ	RSA OAEP	最大2048	FTM信頼メモリー
FTM外部の暗号化セッションキーデータ	ECC曲線25519	最大256	FTM信頼メモリー
生体認証情報の署名	RSA	最大2048	鍵はFTMで生成および破棄され、FTM外には出ない
生体認証情報の署名	ECC曲線25519	最大256	鍵はFTMで生成および破棄され、FTM外には出ない
セキュアブート	RSA	最大256	FTM信頼メモリー
セキュアブート	ECC曲線25519	最大256	FTM信頼メモリー

その他のECC曲線はサポートされない

11. 署名

上記すべてのAPIの中には、リクエスト発出元の正当性を検証するために、さまざまな鍵を使用して署名されるリクエストおよびレスポンスもある。各種APIのリクエストまたはレスポンス本文の、特定ブロックの署名に使用する鍵の詳細を示す

リクエスト/レスポンス	ブロック	シグネチャキー
デバイス検出レスポンス	デバイス情報	署名されないため適用なし
デバイス検出レスポンス	デジタルID	署名されないため適用なし
デバイス情報レスポンス	デバイス情報	登録されていないデバイスの場合は適用なし 登録されているデバイスの場合はデバイスキー
デバイス情報レスポンス	デジタルID	L0デバイスの場合はデバイスキーを使用 L1デバイスの場合はFTMチップキーを使用
キャプチャレスポンス	データ	デバイスキーを使用
キャプチャレスポンス	デジタルID	FTMチップキーを使用
登録キャプチャレスポンス	データ	デバイスキーを使用
登録キャプチャレスポンス	デジタルID	L0デバイスの場合はデバイスキーを使用 L1デバイスの場合はFTMチップキーを使用
デバイス登録リクエスト	デバイスデータ	デバイスプロバイダーの証明書を使用
デバイス登録リクエスト	デバイス情報	デバイスキーを使用
デバイス登録リクエスト	デジタルID	L0デバイスの場合はデバイスキーを使用 L1デバイスの場合はFTMチップキーを使用
デバイス登録解除リクエスト	デバイス	デバイスプロバイダーの証明書を使用
デバイス登録レスポンス	対応	MOSIP署名証明書を使用
デバイス登録レスポンス	デジタルID	リクエストと同様
デバイス登録解除レスポンス	デバイス	MOSIP署名証明書を使用

2.6.3 MDS 仕様（36/36）

12. エラーコード

コード	メッセージ
0	正常完了
100	デバイスが未登録
101	生体認証オブジェクトが検出されない
102	抽出時に技術的エラーが発生
103	デバイスの改ざんを検出
104	管理サーバーに接続できない
105	画像生成エラー
106	デバイスが見つからない
107	デバイスの公開鍵の期限が切れている
108	ドメインの公開鍵が不明
109	リクエストされた数の生体認証情報（指紋/虹彩）はサポートされていない
5xx	カスタムエラー。エラーコードとエラーメッセージは、デバイスプロバイダーが自由に選択できる

2.6.4 バイオメトリクス データ仕様（1/11）

はじめに

様々なモダリティの生体情報画像の表現およびデータ交換は、以下の仕様に従う

1. 画像フォーマット

1.1 指紋のキャプチャ

ISO 19794-4:2011 を参照

要素	登録デバイス	認証デバイス
最小解像度	ネイティブ 500 DPI 以上。これは最低限の値であり、ぎりぎり推奨されるものである。より高い値が望ましい	ネイティブ 500 DPI 以上。これは最低限の値であり、ぎりぎり推奨されるものである。より高い値が望ましい
本人拒否率（FRR）[2]	各国で 2% FRR 未満	各国で 2% FRR 未満
他人受入率（FAR）[2]	0.01%	0.01%
DPI	500[1]	500
画像仕様	ISO 19794-4 B.1 AFIS Normative	ISO 19794-4 B.2 Personal Verification
ESD	>= 8kv	>= 8kv
EMCコンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
動作温度[2]	0 ～ 50℃	□30 ～ 50℃

要素	登録デバイス	認証デバイス
生存検出[3]	IEEE 2790 に従う	IEEE 2790 に従う
プレビュー	3 FPS以上のJPEG losslessフレーム、NFIQ 2スコアを画面に重ねて表示	なし
画像フォーマット	JPEG 2000 lossless	JPEG 2000 lossless、WSQ (最大圧縮 10:1)[2]
品質スコア	NFIQ 2	NFIQ 1
FTM	SBI 1.0 – ホストベースのセキュリティを使用 SBI 2.0 – FTMがサポートするセキュリティ	SBI 2.0 – FTMがサポートするセキュリティ

1. 十分かどうか登録時に検証する。2. 必要な場合、MOSIP導入国で変更可能。 3. 本機能を使用可能にするかどうかはMOSIP導入国が決定する

2.6.4 バイオメトリクス データ仕様（2/11）

1.2 虹彩キャプチャ

ISO 19796-6:2011 Part 6 Specificationsを参照

要素	登録デバイス	認証デバイス
回転角度	圧縮前に虹彩画像に前処理を行い、回転角度を計算する 必要がある。球面収差が補正された画像の回転角度計算については、ISO 19794-6 Section 6.3.1 を参照。	-
回転の不定性	ISO 19794-6 を参照	-
最小直径	ISO 19794-6:2011 のとおり、中～高品質画像のみが受付 可能。したがって本規格では、受付可能な虹彩の直径の 最小値は150ピクセルとする。	同左
マージン	ISOと同じ	-
色	虹彩画像はピクセル深度8ビット/ピクセルのグレースケールで 取込および保存すること	-
照明	目にあてる光には、高品質なグレースケール画像を生成できる赤外線またはその他の光源を使用すること	-
画像形式	JPEG 2000 lossless	JPEG 2000 lossless
アスペクト比	1:1	-
画像品質	ISO/IEC 29794-6	ISO/IEC 29794-6

2.6.4 バイオメトリクス データ仕様（3/11）

要素	登録デバイス	認証デバイス
動作温度[1)]	□30 ～ 50℃	□30 ～ 50℃
EMCコンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
プレビュー	3 FPS以上のJPEG losslessフレーム、品質スコアを画面に重ねて表示	適用なし
画像仕様	ISO 19794-6	ISO 19794-6
ISO形式	K3	K7
FTM	SBI 1.0 – ホストベースのセキュリティを使用 SBI 2.0 – FTMがサポートするセキュリティ	SBI 2.0 – FTMがサポートするセキュリティ

1.3 顔のキャプチャ

ISO 19794-5:2011を参照

要素	登録デバイス	認証デバイス
最小解像度	2.8 mm、画角110度で 1080ピクセル	2.8 mmで1080ピクセル
肌トーン	すべて	すべて
動作温度[1)]	□30 ～ 50℃	□30 ～ 50℃
EMCコンプライアンス	FCCクラスAまたは同等	FCCクラスAまたは同等
画像仕様	ISO/IEC 19794-5	ISO/IEC 19794-5
例外画像仕様	FACE機能で真正面、顔の横に手のひら2つ分のスペースがある上半身の写真。6 x 4 mm	適用なし
画像品質	ICAO - 真正面の画像、回転 ±5 度、24ビットRGB、白背景、幅35 mm、高さ45 mm	
画像形式	JPEG 2000 lossless	JPEG 2000 lossless
FTM	SBI 1.0 – ホストベースのセキュリティを使用 SBI 2.0 – FTMがサポートするセキュリティ	SBI 2.0 – FTMがサポートするセキュリティ

登録や認証の場面で使用するデバイスは、導入国で人間工学的形状、アクセシビリティ、使いやすさ、デバイスの共用可能性を検討しつつ選択することが推奨される

2. XMLコンテナ

生体情報データはCBEFF XMLでラッピングされる

1. MOSIP導入国で決定し最終化

2.6.4 バイオメトリクス データ仕様（4/11）

CBEFF

- 規格
  - ISO 19785-3
  - OASISパトロンフォーマットISO/IEC JTC 1 SC 37 – バイオメトリクス（OASIS patron format ISO/IEC JTC 1 SC 37 - biometrics）、パトロン識別子257、パトロンフォーマット識別子11
  - フォーマットタイプ ISO/IEC JTC 1 SC 37-バイオメトリクス向けOASISバイナリデータブロック形式識別子（OASIS Binary Data Block Format Identifiers）、パトロン識別子 257、BDBパトロンフォーマット識別子7（指紋認証用）、8（顔認証用）、9（虹彩認証用）。
- スキーマ
- CBEFF XMLデータの作成、更新、検索、および検証を行うMOSIPのCBEFFユーティリティ

1. CBEFFサンプル

```
<?xml version="1.0" encoding="UTF-8"?>
<BIR xmlns="http://standards.iso.org/iso-iec/19785/-3/ed-2/">
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <!-- right index finger -->
   <BIR>
      <Version>
         <Major>1</Major>
         <Minor>1</Minor>
      </Version>
```

```
      <CBEFFVersion>
         <Major>1</Major>
         <Minor>1</Minor>
      </CBEFFVersion>
      <BIRInfo>
         <Integrity>false</Integrity>
      </BIRInfo>
      <BDBInfo>
         <Format>
            <Organization>Mosip</Organization>
            <Type>257</Type>
```

```
         </Format>
         <CreationDate>2019-06-27T13:40:06.209Z</CreationDate>
         <Type>Finger</Type>
         <Subtype>Right IndexFinger</Subtype>
         <Level>Raw</Level>
         <Purpose>Enroll</Purpose>
         <Quality>
            <Algorithm>
               <Organization>HMAC</Organization>
               <Type>SHA-256</Type>
```

```
               </Algorithm>
               <Score>100</Score>
            </Quality>
         </BDBInfo>
         <BDB>RklSAD...</BDB>
      </BIR>
      <!-- right middle finger -->
      <BIR>
         <Version>
            <Major>1</Major>
            <Minor>1</Minor>
         </Version>
```

2.6.4 バイオメトリクス データ仕様（5/11）

```
    <CBEFFVersion>
        <Major>1</Major>
        <Minor>1</Minor>
    </CBEFFVersion>
    <BIRInfo>
        <Integrity>false</Integrity>
    </BIRInfo>
    <BDBInfo>
        <Format>
```

```
            <Organization>Mosip</Organization>
            <Type>257</Type>
        </Format>
        <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
        <Type>Finger</Type>
        <Subtype>Right MiddleFinger</Subtype>
        <Level>Raw</Level>
        <Purpose>Enroll</Purpose>
        <Quality>
            <Algorithm>
                <Organization>HMAC</Organization>
```

```
                <Type>SHA-256</Type>
            </Algorithm>
            <Score>100</Score>
        </Quality>
    </BDBInfo>
    <BDB>RklSAD...</BDB>
</BIR>
<!-- right ring finger -->
<BIR>
    <Version>
```

```
        <Major>1</Major>
        <Minor>1</Minor>
    </Version>
    <CBEFFVersion>
        <Major>1</Major>
        <Minor>1</Minor>
    </CBEFFVersion>
    <BIRInfo>
        <Integrity>false</Integrity>
    </BIRInfo>
    <BDBInfo>
```

```
        <Format>
            <Organization>Mosip</Organization>
            <Type>257</Type>
        </Format>
        <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
        <Type>Finger</Type>
        <Subtype>Right RingFinger</Subtype>
        <Level>Raw</Level>
        <Purpose>Enroll</Purpose>
        <Quality>
```

```
            <Algorithm>
                <Organization>HMAC</Organization>
                <Type>SHA-256</Type>
            </Algorithm>
            <Score>100</Score>
        </Quality>
    </BDBInfo>
    <BDB>RklSAD...</BDB>
</BIR>
<!-- right little finger -->
```

2.6.4 バイオメトリクス データ仕様（6/11）

```
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Finger</Type>
      <Subtype>Right LittleFinger</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
```

```
<!-- left index finger -->
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Finger</Type>
      <Subtype>Left IndexFinger</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
```

2.6.4 バイオメトリクス データ仕様（7/11）

```
<!-- left middle finger -->
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Finger</Type>
      <Subtype>Left MiddleFinger</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
<!-- left ring finger -->
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Finger</Type>
      <Subtype>Left RingFinger</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
```

2.6.4 バイオメトリクス データ仕様（8/11）

```
    </BDBInfo>
    <BDB>RklSAD...</BDB>
</BIR>
<!-- left little finger -->
<BIR>
    <Version>
        <Major>1</Major>
        <Minor>1</Minor>
    </Version>
```

```
    <CBEFFVersion>
        <Major>1</Major>
        <Minor>1</Minor>
    </CBEFFVersion>
    <BIRInfo>
        <Integrity>false</Integrity>
    </BIRInfo>
    <BDBInfo>
        <Format>
            <Organization>Mosip</Organization>
            <Type>257</Type>
        </Format>
```

```
        <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
        <Type>Finger</Type>
        <Subtype>Left LittleFinger</Subtype>
        <Level>Raw</Level>
        <Purpose>Enroll</Purpose>
        <Quality>
            <Algorithm>
                <Organization>HMAC</Organization>
                <Type>SHA-256</Type>
```

```
            </Algorithm>
            <Score>100</Score>
        </Quality>
    </BDBInfo>
    <BDB>RklSAD...</BDB>
</BIR>
<!-- right thumb finger -->
<BIR>
    <Version>
        <Major>1</Major>
        <Minor>1</Minor>
```

```
    </Version>
    <CBEFFVersion>
        <Major>1</Major>
        <Minor>1</Minor>
    </CBEFFVersion>
    <BIRInfo>
        <Integrity>false</Integrity>
    </BIRInfo>
    <BDBInfo>
        <Format>
```

```
            <Organization>Mosip</Organization>
            <Type>257</Type>
        </Format>
        <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
        <Type>Finger</Type>
        <Subtype>Right Thumb</Subtype>
        <Level>Raw</Level>
        <Purpose>Enroll</Purpose>
        <Quality>
            <Algorithm>
                <Organization>HMAC</Organization>
```

2.6.4 バイオメトリクス データ仕様（9/11）

```
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
<!-- left thumb finger -->
<BIR>
   <Version>
```

```
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
```

```
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Finger</Type>
      <Subtype>Left Thumb</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
```

```
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
<!-- face -->
<BIR>
   <Version>
```

```
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
```

```
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Face</Type>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
```

2.6.4 バイオメトリクス データ仕様（10/11）

```
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
<!-- right iris -->
```

```
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Iris</Type>
      <Subtype>Right</Subtype>
      <Level>Raw</Level>
      <Purpose>Enroll</Purpose>
      <Quality>
         <Algorithm>
            <Organization>HMAC</Organization>
            <Type>SHA-256</Type>
         </Algorithm>
         <Score>100</Score>
      </Quality>
   </BDBInfo>
   <BDB>RklSAD...</BDB>
</BIR>
<!-- left iris -->
```

```
<BIR>
   <Version>
      <Major>1</Major>
      <Minor>1</Minor>
   </Version>
   <CBEFFVersion>
      <Major>1</Major>
      <Minor>1</Minor>
   </CBEFFVersion>
   <BIRInfo>
      <Integrity>false</Integrity>
   </BIRInfo>
   <BDBInfo>
      <Format>
         <Organization>Mosip</Organization>
         <Type>257</Type>
      </Format>
      <CreationDate>2019-06-27T13:40:06.211Z</CreationDate>
      <Type>Iris</Type>
      <Subtype>Left</Subtype>
      <Level>Raw</Level>
```

2.6.4 バイオメトリクス データ仕様（11/11）

```
    <Purpose>Enroll</Purpose>↵
    <Quality>↵
      <Algorithm>↵
        <Organization>HMAC</Organization>↵
        <Type>SHA-256</Type>↵
      </Algorithm>↵
      <Score>100</Score>↵
    </Quality>↵
  </BDBInfo>↵
```

```
    <BDB>RklSAD...</BDB>↵
  </BIR>↵
</BIR>↵
```

2.7 プライバシー & セキュリティ（1/3）

はじめに

MOSIPにおいてプライバシーとセキュリティは最優先であり、本ドキュメントでは現在プラットフォームに実装されている施策の詳細について説明する。オープンソースプロジェクトとして、MOSIPではコラボレーションとコミュニティの貢献を通じてセキュリティ機能を継続的に改善し、新規開発機能を導入し続けることを狙う。さまざまなセキュリティツールを使用してセキュリティを評価し、脆弱性を発見して対応している

1. 主要な原則

プライバシーとセキュリティについてのMOSIPのアプローチはフレームワーク原則に定められ、これに基づいて運用される

2. MOSIPセキュリティデザインの主な特徴

- データベースに保存されたデータへの直接アクセスは禁止 - データにはAPIを経由してのみアクセスする
- ゼロナレッジ管理原則が適用されるので、管理者は実際のデータを表示することなくデータを管理できる。データはAPIを経由してのみアクセスできる
- 各データベース行は一貫性が保護され、IDのスワッピングなどの悪意ある改ざんを防ぐ
- 取り消し可能な仮想IDおよびトークンを使用して、ユーザーのプロファイリングについての試みを阻止する
- データのプライバシー (誰が何を見ることができるか) を確保するためにすべてのAPIにアクセス管理が実装されている
- すべてのAPIはレート制限 (特定の時間内でアクションを繰り返すことができる頻度) をサポートしており、デジタル署名されている
- すべてのネットワークチャネルは安全ではないと仮定される。
- すべてのアーティファクト (APIで送信されるJSONデータを含む)はデジタル署名される。

3. MOSIP暗号化アルゴリズム

MOSIPでは以下のアルゴリズムが使用される。

- 1. RSA OAEP 最小2048ビット (すべての公開鍵暗号用)
- 2. AES GCM 最小256ビット (すべての対称鍵暗号用)
- 3. SHA256 (標準ハッシュアルゴリズムとして)
- 4. X509 V3 (証明書規格として)
- 5. FIPS 140-2 レベル 3 (最小限のハードウェアセキュリティモジュール(HSM)規格として)
- 6. PKCS11 (HSM通信用)

4. データベースの暗号化

原則として、MOSIPの暗号化にはデータベースに組み込まれたメカニズムを使用しない。DBに保存されるすべての機密データはDBの外部のアプリケーションレイヤーで暗号化/復号化される

- すべての機密データ（設定可能）は対称鍵アルゴリズムで暗号化される。MOSIPはデフォルトでAES 256アルゴリズムをサポートする
- 各セルはそれぞれ独自の対称鍵を用いて暗号化され、鍵はランダムに選択される
- デフォルトで、10,000個の対称鍵を生成してデータベースを暗号化する。これはソフトウェア定義された上限であり、増やすことができる
- 対称鍵はHSMのマスターキーを用いて暗号化される
- それぞれの鍵には有効期限があり、アプリケーションは有効期限に従って新しい鍵でデータを更新する

2.7 プライバシー & セキュリティ（2/3）

5. 登録データの暗号化

登録クライアントを使用してMOSIPにすべての個人情報と生体情報を収集する。クライアントはTPM互換マシンで動作するよう設計されている。クライアントは以下の原則に従う

- すべてのマシンはTPM識別鍵を使用して登録される。これらの鍵の公開部分はMOSIPデータベース内であらかじめホワイトリスト化されている
- SRK鍵からSKが作成され、これを使用してMOSIPで使用するその他すべての鍵を暗号化する
- すべてのローカル構成は同じメカニズムで暗号化される
- 登録用にRSA鍵のセット(デフォルトで最大10000)が作成される。これらの鍵はHSMで生成され、その公開部分はMOSIP登録クライアントに埋め込まれる。
  これらの鍵はユーザーデータ/住民データの暗号化に使用される
- 暗号形式の登録データは常に揮発メモリーに保存され、永続ストレージに保存されることは決してない

6. 鍵管理

MOSIPではAES鍵とRSA鍵が使用される。デフォルトで、MOSIPでは設定可能なパラメーターとして有効期限と鍵のローテーションを有するよう設計されている。デフォルト値は以下のように設定されているすべてのマシンはTPM識別鍵を使用して登録される。これらの鍵の公開部分はMOSIPデータベース内であらかじめホワイトリスト化されている

- AES 256ビット鍵 - 作成日から6カ月
- RSA 2048ビット暗号化鍵 - 作成日から1年
- RSA 2048ビット署名鍵 - 作成日から2年

6.1.1 データベースの暗号鍵

- MOSIPでは対称鍵を使用してデータベースを保護する
- すべての鍵には作成時に指定した有効期限がある (設定によって定義され、デフォルトは6カ月)。鍵管理には以下の2つの運用モードがある
- インライン
  - このモードでは、設定に従う
  - データをデータベースに書き戻すときは、新しいアクティブ鍵が使用される
  - データの読み取り時に暗号鍵の有効期限が切れている場合、有効期限の切れた鍵を使用している鍵管理に通知され、アクティブ鍵で再暗号化を行う必要がある
- バッチ
  - 本モードでは、すべてのテーブルについて有効期限の切れた鍵を持つ暗号データを検索する
  - 新しいアクティブ鍵でこれらを再暗号化する
  - このモードは必要に応じて、または隔月で実行するようスケジュールされるので、大量のデータが処理されることはなく、ほとんどのデータはインラインモードで再暗号化される

2.7 プライバシー & セキュリティ（3/3）

7. 登録データの暗号化

登録クライアントを使用してMOSIPにすべての個人情報と生体情報を収集する。クライアントはTPM互換マシンで動作するよう設計されている。クライアントは以下の原則に従う

- すべてのマシンはTPM識別鍵を使用して登録される。これらの鍵の公開部分はMOSIPデータベース内であらかじめホワイトリスト化されている
- SRK鍵からSKが作成され、これを使用してMOSIPで使用するその他すべての鍵を暗号化する
- すべてのローカル構成は同じメカニズムで暗号化される
- 登録用にRSA鍵のセット(デフォルトで最大10000)が作成される。これらの鍵はHSMで生成され、その公開部分はMOSIP登録クライアントに埋め込まれる。
  これらの鍵はユーザーデータ/住民データの暗号化に使用される
- 暗号形式の登録データは常に揮発メモリーに保存され、永続ストレージまたはファイルシステムに保存されることは決してない

7.1 鍵管理

RSA 2048ビット鍵を使用して、登録クライアントからの住民データ/ユーザーデータの暗号化を行う。有効期限ポリシーは1年間に設定される

- これらの鍵のデフォルトの有効期限は1年間に設定される
- これらの鍵はAPIを使用してローテーションされる。現在、鍵のローテーションはクライアントのアップグレード時に手作業で行われている

7.1.1 デジタル署名鍵

デジタル署名鍵はドメイン固有であり、外部で使用するためにMOSIPで生成されたアーティファクトの署名に用いる。
デジタル署名に関しては、各国はそれぞれの法制度に従うことが期待される。デフォルトの有効期限は2年間に設定される

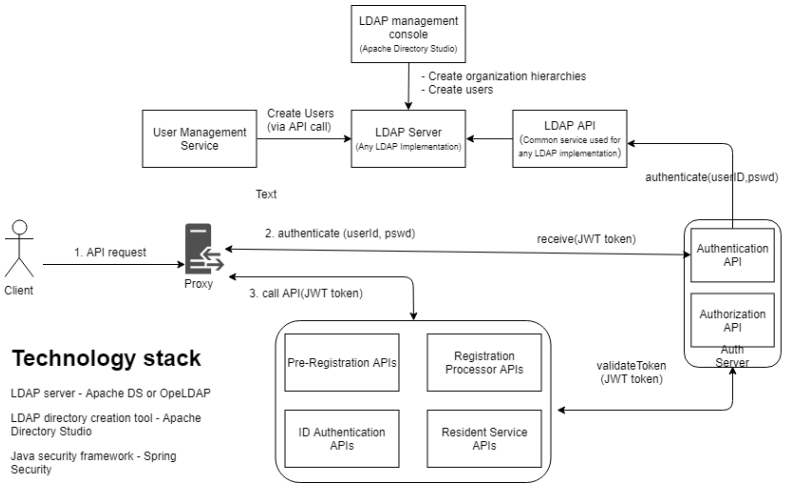
8. 認証と認可 (未定)

MOSIPの認証はおおまかに以下のカテゴリーに分類される。

- Webチャネル経由の認証 (事前登録Webアプリ、管理Webアプリ、および住民サービスポータル)
- ローカルシステム経由の認証、すなわちオフライン認証 (登録クライアント)

MOSIPの認可は以下のカテゴリーに分類される。

- Webチャネルを通じてアクセスされたAPIの認可(現時点ではKeyCloakサーバーへマイグレーション中。ドキュメントは間もなく発行予定)
- 特定のデータにアクセスするための認可 (v3で実装予定) 国には、特に登録スタッフやシステム管理スタッフといったシステム利用者の階層が存在する。そのため、MOSIPでは固定の階層を定義する代わりに、デフォルトではLDAPの実装に依存して、ユーザー、組織の階層、階層内のユーザーの役割を管理する。MOSIPではLDAPの実装としてオープンソースのLDAPサーバーを使用する。管理者はApache Directory Studioを使用して階層とユーザーを作成する



3.
システム開発

3.1.1 アーキテクチャの配置（1/2）

1. はじめに

国家のIDシステムを運用することは簡単なタスクではなく、困難な局面が多くある。中核となるソフトウェアシステムは重要なインフラであり、高可用性、信頼性、拡張性、セキュリティ、回復力、管理性を備えたものである必要がある。国の法律を遵守しつつアーキテクチャとしての目標達成を支える上で、適切なデプロイアーキテクチャを選択することが、重要な役割を果たす。そういったアーキテクチャの実装のコストも重要である

MOSIPは無数のスモールサービスと実行ユニットを編成して機能を実現するマイクロサービスアーキテクチャをとる。それぞれは個別にスケールでき、個々に置き換えやアップグレードが可能である。これにより、プラットフォームは強化され、実装を行う上で極めて柔軟かつ自由に設定できるようになる。これに伴って、構成、セキュリティ、デプロイ、依存関係管理、監視、テストの各分野でシステム内のコンポーネントが増え、扱いが複雑化することにもなる

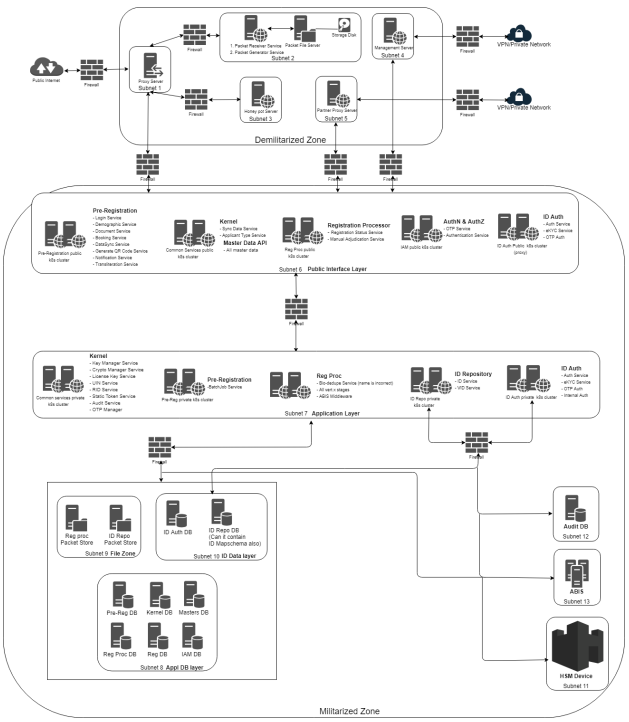
2. デプロイアーキテクチャの選択肢

MOSIPを最大限に活用しつつ高い管理性を保つには、デプロイアーキテクチャが極めて重要な役割を果たす。さまざまな観点から考えられる共通デプロイアーキテクチャのオプションには、以下のようなものがある

- パッケージの選択肢
  - オプション「Jar」 – 仮想マシン内のSpring Bootサービス
  - オプション「Docker」 – KubernetesでDockerコンテナを管理するセットアップ
- インフラの選択肢
  - オプション「オンプレミス」 – プライベートな、または自己所有のデータセンターへのデプロイ
  - オプション「クラウド」 – クラウドへのデプロイ
  - オプション「ハイブリッド」 – クラウド＋オンプレミス
- プラットフォームの選択肢
  - オプション「オープンソース」 – 実績のある、コミュニティに支持されたプラットフォーム
  - オプション「クラウドネイティブ」 – AWS、Azure、GCPなどのクラウドテクノロジー企業がサポートする最先端環境
  - オプション「商用」 – サポートの手厚い既存の有料パッケージ

3. セキュリティ: セキュアゾーンでのデプロイ

提案されるアーキテクチャはオンプレミスまたはクラウドである。ここでは、すべてのMOSIPモジュールについて武装地帯にインストールするものと非武装地帯にインストールするものとを明確に分けてインストールが行われる



3.1.1 アーキテクチャの配置（2/2）

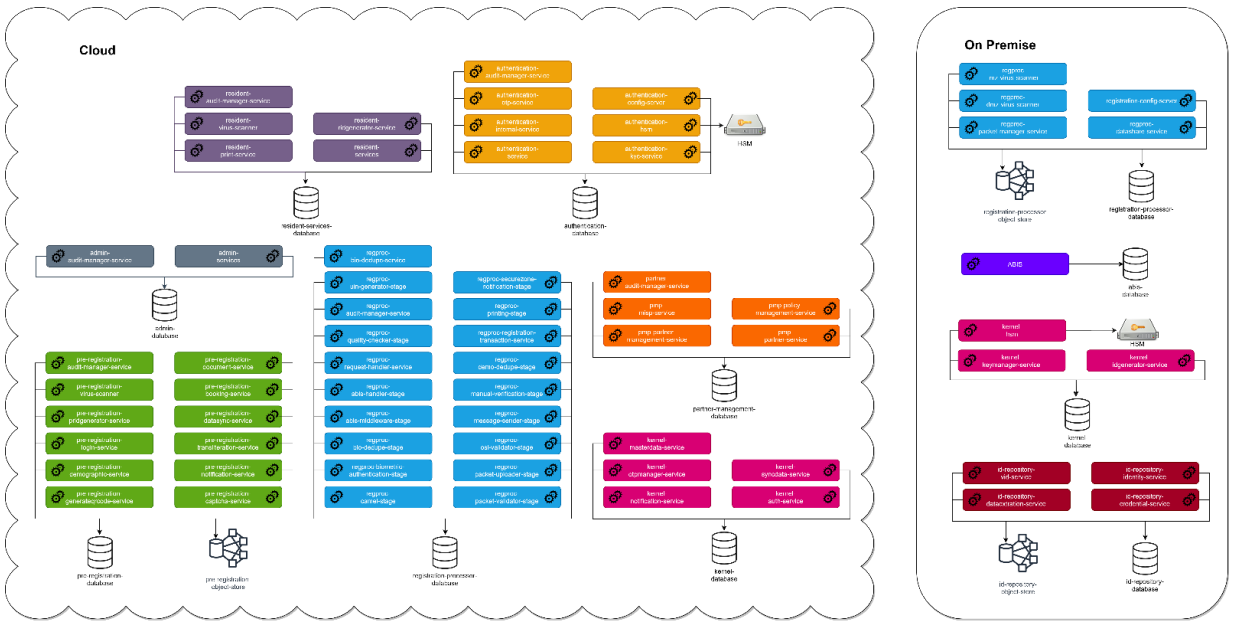
4. スケーラビリティ: セルベースアーキテクチャ

処理能力とハードウェアのプロビジョニングを線形にスケーリングするために、セルベースアーキテクチャ（セキュアゾーンも設定）が望ましい
セルベースアーキテクチャ

5. 迅速なデプロイメント: ハイブリッドアーキテクチャ

クラウドとオンプレミスの両方のメリットを活用するハイブリッドアーキテクチャを検討してもよい。クラウドでは迅速なデプロイが可能で管理も簡単な一方、オンプレミスではデータをローカル環境で扱うことができ、その他のポリシー要件もローカルに管理できる

ハイブリッドアーキテクチャの例を下図に示す



3.1.2 Cell-based アーキテクチャ

背景

複雑なシステムをスケーラブルに構築するのは簡単ではない。マイクロサービス、データベース、ストレージクラスターなどの複数のコンポーネントが複雑に連携し合って実行されている場合には特に手間がかかる。そのようなシステムでは「各部」のパフォーマンスを単純に足し合わせれば「全体」のパフォーマンスになるわけではないため、エンドツーエンドのパフォーマンスモデリングには大きな課題がある

MOSIPでは、ハードウェアとソフトウェアをセルに固定する (閉じ込める) セルアーキテクチャが推奨され、入出力の処理能力のベンチマークはセル単位で行われる。セルを複製し、ロードバランサーでトラフィックを分散させることで、本番環境の処理能力をスケールできる。理想的には、各セルは相互依存のないように分離されなければならない。しかし、実質的には特定のリソースを共有する場合がある。そのような共通リソースをスケーラブルにするためには、別々にアドレッシングする必要がある

本ドキュメントでは、主なMOSIPモジュールのすべてに関して、本番環境におけるセルアーキテクチャについて説明する

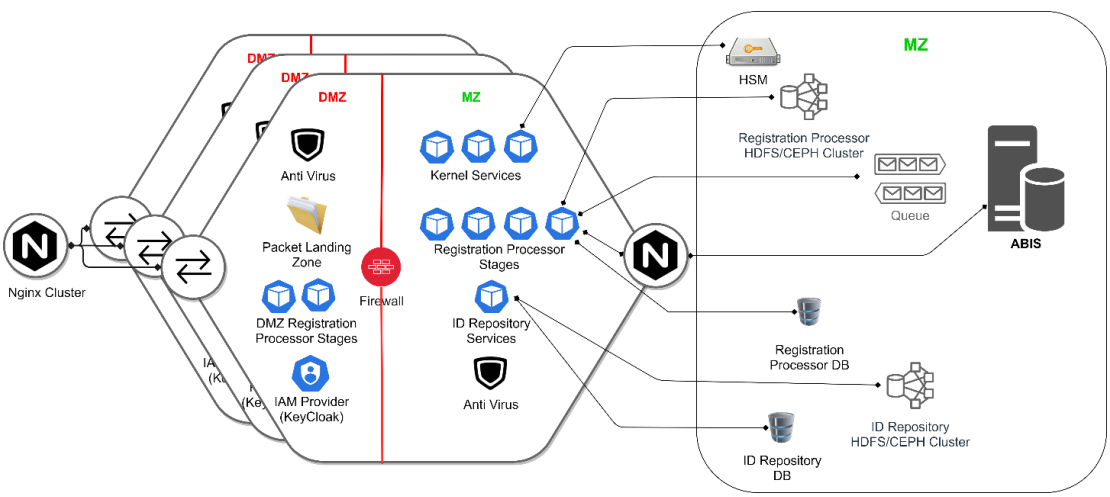
1. 登録プロセッサーセル

以下のリソースがセル間で共有される

- ABIS
- ABISキュー
- 登録プロセスDB
- IDリポジトリHDFS/ CEPHクラスター
- IDリポジトリDB

非武装地帯 (DMZ) と武装地帯 (MZ) の間の通信は、厳密にファイアウォールを介して行われる

登録クライアントが送信した暗号化済みパケットは、最初にDMZのパケット着信ゾーンに着信する。初期パケット処理のため、登録プロセッサーのステージの一部がDMZで実行される



3.1.3 ハードウェア・セキュリティ・モジュール (HSM)（1/2）

HSMとはハードウェア・セキュリティ・モジュール (Hardware Security Module) の略称であり、暗号処理と強力な認証に特化して設計された極めてセキュリティの高い物理デバイスである。デジタル鍵の暗号化、復号化、作成、保存、管理ができ、署名および認証に使用される。目的は攻撃からの鍵の防御と保護である

MOSIPは、HSMについて以下の仕様を強く推奨する

1. 暗号化のオフローディングとアクセラレーションをサポートしなければならない
2. 認証された複数の役割についてのアクセス管理を提供すること
3. 管理者とオペレーターの役割は厳格に分離されていなければならない
4. クライアント認証をサポートできること
5. セキュアなキーラッピング、バックアップ、複製、回復の機能がなければならない
6. 暗号化モジュールのFIPS 140-2レベル3認証メモリで2048個の4096 ビット RSA 秘密鍵、256ビットAES鍵をサポートしていなければならない
7. 暗号化モジュールのFIPS 140-2レベル3認証メモリで少なくとも10000個以上の2048 RSA秘密鍵をサポートしなければならない
8. クラスタリングおよび負荷分散をサポートしなければならない
9. 論理パーティションを使用したアプリケーションキーの暗号分離をサポートすること
10. N分のM多要素認証をサポートしなければならない
11. PKCS#11、OpenSSL、Java (JCE)、Microsoft CAPI、およびCNG
12. 最低2つのギガビットイーサネットポート (2つのネットワークセグメントに接続するため) と10ギガ光ポートを利用可能であること
13. 非対称公開鍵アルゴリズム: RSA、DiffieHellman、DSA、KCDSA、ECDSA、ECDH、ECIES
14. 対称アルゴリズム: AES、ARIA、CAST、HMAC、SEED、Triple DES、DUKPT、BIP32
15. ハッシュ/メッセージダイジェスト: SHA-1、SHA-2 (224、256、384、512ビット)
16. Brainpool標準曲線、カスタム曲線、安全曲線を含むフルライセンスECC付き完全Suite B実装

3.1.3 ハードウェア・セキュリティ・モジュール (HSM)（2/2）

17. 安全性および環境コンプライアンス
    1. UL、CE、FCC part 15 class Bに準拠
    2. RoHS2、WEEE準拠
18. 管理およびモニタリング
    1. リモート管理をサポート – NoCからのアプリケーション追加、ファームウェア更新、状態監視を含む
    2. Syslog診断をサポート
    3. コマンドライン・インターフェース(CLI)/グラフィカル・ユーザー・インターフェース (GUI)
    4. モニタリング用SNMPエージェントのサポート
19. 物理特性
    1. PIN ENTRYデバイス一体型1U 19インチ標準ラックマウント
20. パフォーマンス
    1. RSA 2048 署名 – 10000/秒
    2. RSA 2048 鍵生成 – 10/秒以上
    3. RSA 2048 暗号化/復号化 - 20000以上
    4. RSA 4096 署名 - 5000/秒
    5. RSA 4096 鍵生成 - 2/秒以上
    6. RSA 4096 暗号化/復号化 - 20000以上
21. DRのためオフラインの鍵付きの場所に鍵をバックアップ、複製、保存できるようにしておくこと。総容量は、所定の鍵の総数と一致させること
22. 最小で20台のHSMをクラスタリングすること
23. 鍵の複製はクラスタで30秒未満であること
24. 最小論理パーティション数は30であり、そのライセンスはコストに含まれること

3.1.4 ハードウェアサイジング（1/2）

MOSIPコアプラットフォームで使用するハードウェアの処理能力およびストレージの要件については、
以下のように見積もる

1 本番環境

1.1 処理能力

本番環境デプロイに必要なハードウェア処理能力の見積もりは、以下のとおりである

モジュール	処理能力	登録サーバー	構成
事前登録	事前登録件数　7200 件/時[1)]	10	4 VCPU[2)] 、 16 GB RAM
登録プロセッサー	登録件数　200,000件/日	80	4 VCPU、 16 GB RAM
ID認証	認証リクエスト 2,000,000件/日	20	4 VCPU、 16 GB RAM
住民サービス	住民サービス　7200件/時[1)]	10	4 VCPU、 16 GB RAM

管理、モニタリング、メンテナンスのためにプラス30% (概算値) の処理能力が必要と予測される。これは
システムインテグレーター (SI) によって最適化可能である

1. 平均スループット 2. VCPU: 仮想CPU
Note: サービスPod/Dockerあたり2という複製係数を仮定して、高可用性を考慮に入れること; 上記の見積もりには以下に必要なサーバーは含まれない
①データベース ②HDFS/CEPH ③バイオメトリクスSDK ④HSM ⑤ABIS ⑥ウィルススキャン ⑦ロードバランサー ⑧外部IAM ⑨ディザスタリカバリ (DR)

3.1.4 ハードウェアサイジング（2/2）

1.2 ストレージ

本番環境デプロイに必要なストレージの見積もりは以下のとおり

1.2.1 データベースおよびHDFS/CEPH

MOSIPストレージ要件計算用Excelシート

1.2.2 アプリケーションログおよびシステムログ

- アプリケーションログ

モジュール	単位	ログサイズ (Rawデータ)
事前登録	事前登録件数　100 件	20 MB
登録プロセッサー	登録件数　100件	200 MB

上記の見積もりは概算であり、例外トレースがあまりに多く発生したなどの場合はデータ量が急増する場合がある

ログは1週間程度の後に圧縮してアーカイブしてもよい。tar+gzユーティリティで得られる圧縮率は15～20である

- システムログ
  - デプロイ構成に応じてSIによって見積もられる

2 開発、QA、ステージング、本番前環境

以下のセットアップのための追加の処理能力およびストレージが必要である。

環境	セットアップ	サーバー数	構成	ストレージ
開発	サンドボックス	13	4 VCPU、16 GB RAM	128 GB SSD
QA	サンドボックス	13	4 VCPU、16 GB RAM	128 GB SSD
ステージング	サンドボックス	13	4 VCPU、16 GB RAM	128 GB SSD
本番前	セル	1)	4 VCPU、16 GB RAM	1)

1. 導入国またはSIによって決定される

3.1.5 導入国におけるカスタマイゼーション（1/3）

導入国はMOSIPのカスタマイズとデプロイを極めて柔軟に行うことができる。多くのコンポーネントが既成のままで使用できるが、下記のように、特殊なデプロイや特定のカスタマイズおよび追加が必要な場合もある。

1. IDオブジェクト
   - 定義
   - スキーマ
   - 登録クライアントおよび登録プロセッサーに関するスキーマとフィールドのカスタムバリデーション (リファレンスバリデーター)
2. 使用言語
   - 主言語と第二言語の定義
   - 翻訳 (文字転写) ライブラリを登録クライアントに組み込む
   - メッセージテンプレート (マスターデータおよびコンフィギュレーションファイル)
3. マスターデータ: 国固有のマスターデータ
4. 登録プロセッサーフローの追加/修正
   - 新規ステージの追加 (例: CRVSシステムからの取込データ)(Camelコンフィギュレーション)
   - ステージの削除または再アレンジ
   - 人口動態データの重複排除論理
5. コンフィギュレーション
6. 登録クライアントアプリ
   - フィールドをIDオブジェクトに合わせて設定
   - ラベルの言語を変更
   - フィールドのバリデーション
   - 画面フローの変更
   - MDSとの統合
7. 住民ポータル: UI実装
8. 管理ポータル: UI修正 (必要な場合)
9. 外部コンポーネントの統合
   - ウィルススキャン機能
   - ABIS
   - バイオメトリクスSDK (登録クライアント、登録プロセッサー、およびID認証)
   - 手作業による調整
   - IAM (OAuth 2.0準拠)
   - HSM
   - 郵便サービス
   - eメール/SMSゲートウェイ

3.1.5 導入国におけるカスタマイゼーション（2/3）

1 よくある質問

1.1 MOSIPにある機能とない機能は?

機能	MOSIP	備考
UIN生成	あり	
トークン生成	あり	
パートナー管理	あり	パートナー管理サービスAPIが使用可能。ポータルはSIが作成
デバイス管理	あり	登録されたデバイスは管理ポータルで管理できる。デバイス登録APIが使用可能。デバイスベンダーは、登録デバイスの管理とキーローテーションを行うデバイス管理サーバーを提供する
SMS通知	あり	インターフェースが使用可能。SMSゲートウェイ/サービスはSIが提供
eメール通知	あり	インターフェースが使用可能。SMSゲートウェイ/サービスはSIが提供
監査証跡	あり	
テクニカルヘルプデスク	なし	
カスタマー・リレーションシップ・マネジメント (CRM)	なし	
バックアップ/リストア管理	なし	
手作業による調整	なし	生体認証情報の重複が見つかった際のデータの読み出しとパケットの承認/拒否を行うAPIが使用可能
手動検証	なし	
解析	なし	
認証用OTP	あり	
生体認証	あり	

3.1.5 導入国におけるカスタマイゼーション（3/3）

機能	MOSIP	備考
ナレッジ・マネジメント・システム	なし	
支払いゲートウェイ	なし	
カード生成	なし	
カード管理	なし	カードをキューに送って印刷し、郵便システムに転送する実装は可能
不正リスク管理	なし	
添付書類の検索	なし	登録プロセッサーは同様にカスタマイズできる
登録センターのトークン管理	なし	
事前登録/予約の登録	あり	MOSIPの事前登録モジュールがデプロイされている場合
UINの回復 (UIN損失時)	あり	
人口動態情報の更新	あり	
生体情報の更新	あり	
苦情報告	なし	
UIN認証ロック	あり	
トランザクション履歴生成	なし	監査ログ、DBレコード、および住民サービスAPIを使用可能
登録ステータス/更新	あり	住民サービスAPI を使用可能。ポータルはSIが作成
支払いゲートウェイ	なし	
モバイル/表形式登録アプリ	なし	
ウィルススキャン	なし	統合用のフックは提供される。SIで調達および統合を行う

3.1.6 MOSIPサービス（1/3）

本ドキュメントでは、MOSIPの公開サービスおよび非公開サービスについて定義する
公開サービス: 一般の人々が使用できるMOSIPサービスであり、UIまたはユーザートークンでアクセスできる
非公開サービス: サービスからサービスを呼び出して使用できるMOSIPサービスであり、サービストークンまたは制限されたユーザーによってのみアクセスできる

MOSIPサービス	非公開サービス	公開サービス
管理/Bulk Upload	✓	
管理/Login	✓	
管理/AuditManager	✓	
管理/PacketUpdateStatus	✓	
共通/PacketReader-Writer	✓	
カーネル/AuditManager	✓	
カーネル/AuthManager	✓	
カーネル/Login	✓	
カーネル/Refresh		✓
カーネル/Jasperreport	✓	
カーネル/ClientCrypto	✓	
カーネル/CryptoManager	✓	
カーネル/KeyManager	✓	
カーネル/LicenceKey	✓	
カーネル/PartnerCertManager	✓	
カーネル/Signature	✓	
カーネル/TokenIDGenerator	✓	
カーネル/TokenIDGenerator	✓	

MOSIPサービス	非公開サービス	公開サービス
カーネル/ZKCryptoManager	✓	
カーネル/ApplicantType	✓	
カーネル/ApplicantValid Document	✓	
カーネル/Application	✓	
カーネル/BiometricAttribute	✓	
カーネル/BiometricType	✓	
カーネル/BlacklistedWords	✓	
カーネル/Device	✓	
カーネル/DeviceHistory	✓	
カーネル/DeviceProvider	✓	
カーネル/ DeviceProviderManagement	✓	
カーネル/DeviceRegister	✓	
カーネル/DeviceSpecification	✓	
カーネル/DeviceType	✓	
カーネル/DocumentCategory	✓	
カーネル/DocumentType	✓	
カーネル/DynamicField	✓	
カーネル/DynamicField	✓	

3.1.6 MOSIPサービス（2/3）

MOSIPサービス	非公開サービス	公開サービス
カーネル/ExceptionalHoliday	✓	
カーネル/FoundationalTrustProvider	✓	
カーネル/GenderType		✓
カーネル/Holiday		✓
カーネル/IdType	✓	
カーネル/IndividualType		✓
カーネル/Language		✓
カーネル/Location		✓
カーネル/LocationHierarchy	✓	
カーネル/Machine	✓	
カーネル/MachineHistory	✓	
カーネル/MachineSpecification	✓	
カーネル/MachineType	✓	
カーネル/Module	✓	
カーネル/MOSIPDeviceService	✓	
カーネル/PacketRejectionReason	✓	
カーネル/RegisteredDevice	✓	
カーネル/RegistrationCenter	✓	
カーネル/RegistrationCenterDevice	✓	
カーネル/RegistrationCenterHistory	✓	
カーネル/RegistrationCenterType	✓	
カーネル/RegistrationCenterUserMachineHistory	✓	

MOSIPサービス	非公開サービス	公開サービス
カーネル/Schema	✓	
カーネル/Template		✓
カーネル/TemplateFileFormat	✓	
カーネル/TemplateType		✓
カーネル/Title		✓
カーネル/UserDetailsHistory	✓	
カーネル/ValidDocument	✓	
カーネル/WorkingDay	✓	
カーネル/Zone	✓	
カーネル/EmailNotification	✓	
カーネル/SmsNotification	✓	
カーネル/OtpGenerator	✓	
カーネル/OtpValidator	✓	
カーネル/RidGenerator	✓	
カーネル/SyncData	✓	
ID認証/AuditTest	✓	
ID認証/Test	✓	
ID認証/CredentialIssueanceCallback	✓	
ID認証/Cryptomanager	✓	
ID認証/InternalAuth	✓	
ID認証/InternalAuthTxn	✓	
ID認証/InternalOTP	✓	

3.1.6 MOSIPサービス（3/3）

MOSIPサービス	非公開サービス	公開サービス
ID認証/InternalUpdateAuthType	✓	
ID認証/Keymanager	✓	
ID認証/Signature	✓	
ID認証/WebSub	✓	
ID認証/KycAuth		✓
ID認証/OTP		✓
ID認証/Auth		✓
ID認証/StaticPin		✓
ID認証/VID		✓
IDリポジトリ/BiometricExtractor	✓	
IDリポジトリ/CredentialRequestGenerator	✓	
IDリポジトリ/CredentialStore	✓	
IDリポジトリ/ID Repository	✓	
IDリポジトリ/Vid	✓	
パートナー管理サービス/Misp		✓
パートナー管理サービス/PartnerManagement		✓
パートナー管理サービス/DeviceDetail		✓
パートナー管理サービス/FTPChipDetail		✓
パートナー管理サービス/RegisteredDevice		✓
パートナー管理サービス/SecureBiometricInterface	✓	
パートナー管理サービス/PartnerService		✓
パートナー管理サービス/PolicyManagement		✓

MOSIPサービス	非公開サービス	公開サービス
事前登録/Demographic		✓
事前登録/Document		✓
事前登録/GenerateQRcode		✓
事前登録/Login		✓
事前登録/Notification		✓
事前登録/Transliteration		✓
事前登録/Booking		✓
事前登録/Captcha		✓
事前登録/DataSync	✓	
登録プロセッサー/BioDedupe	✓	
登録プロセッサー/RegistrationStatus	✓	
登録プロセッサー/RegistrationSync	✓	
登録プロセッサー/PrintApi		✓
登録プロセッサー/RegistrationTransaction	✓	
登録プロセッサー/External		✓
登録プロセッサー/QCUsers	✓	
登録プロセッサー/QualityChecker	✓	
住民サービス/Resident		✓
住民サービス/ResidentVid		✓

第3章
ご参考

1.
他のDigital ID プラットフォームとの比較

1.1 主要デジタルIDプラットフォームとの比較 (1/3)

比較項目	MOSIP	X-Road	CODEX
国	インド	エストニア	シンガポール
基本機能	• 各国で設定やカスタマイズが可能なコアモジュールと、デジタルIDプラットフォームを使ったデジタルサービスを提供	• 分散管理されたデータに対して，インターネットを介してセキュアにアクセスするための基盤として、デジタルIDを使ったデジタルサービスを提供	• National Digital Identity (NDI) など、共通のデジタル・プラットフォームによる、公的機関および民間企業向けのサービスを提供
システムアーキテクチャ	• India Stackをベースにした、モジュラーテクノロジーアーキテクチャ (MOSIPは、India Stackのコアテクノロジーを海外展開するためにオープンソース化したプラットフォーム)	• X-Road: Service-Oriented Architectureを採用し、データ交換プロトコルとしてSOAP、サービス記述言語としてWSDLを利用	• CODEXは、サービス・アプリケーション、SGTS(マクロサービス、ミドルウェア、ホスティングプラットフォーム)、およびデータの５つのレイヤーで構成
その他技術特性や設計思想	• Base Registry (Aadhaar, NPR, NRIC/NRC, PAN, GSTIN, CLR/SPR & ULR, Vahan, Sarathi, etc.)	• X-Road End-to-End Monitoring Tool (XRdE2E) • ‘Privacy by Design’ (PbD) プリンシパル	• AMM (アジャイル成熟度モデル)
国際的な主要基準への準拠性 (ID4Dの10原則、NISTのデジタルIDに関するガイドライン、eIDAS関連文書等)	• ID4Dをサポート • 世界銀行の「持続可能な開発のための識別に関する原則」を支持	• ID4Dをサポート	• ID4Dをサポート

Source: BCG調査

1.1 主要デジタルIDプラットフォームとの比較 (2/3)

比較項目	MOSIP	X-Road	CODEX
オープンソースソフトウェア(OSS)利用状況	• オープンソースコード • オープンスタンダード • オープンAPIによる連携	• データのリンクに暗号化ハッシュ関数を使用してるが、ブロックチェーンではない (分散型プライベートデータベース) • オープンAPIによる連携	• オープンスタンダード • オープンAPIによる連携
開発体制(官民連携)	• 機械学習(ML)や人工知能(AI)の専門知識を持つUSやUKのトップ教育機関と提携 • MOSIP上でシステムを組み込めるサービス提供者に研修を施し、MOSIPが入念に精査 • GitHub上でのフィードバック等、開発者コミュニティによる継続的なレビュー	• Cybernetica (X-Road、i-Voting、e-Customs等の開発) • SKIDソリューションズ (モバイルID/スマートID)	• 民間から技術者を雇い入れるとともに、STACK201、STACK-X(2019)、STACK2020等の開発者向けイベントを開催
機器認証プログラム	• 政府の電子認証基盤e-Pramaan • スマートフォンによる生体認証 • デジタル署名 • ユーザー同意フレームワーク • オフライン認証	• IDカード: ICチップによって非接触式の読み取りも可能、電子認証と電子署名を実行 • モバイルID: SIMベースの電子認証で、携帯電話だけで簡単に各種電子サービスを使用 • スマートID: アプリベースの電子認証で、複数の端末から電子認証が使用可能	• 公的認証システム「SingPass」 • 個人情報の登録・利用の一元化サービス「MyInfo」 • SG Verify: スマートフォンの生体認証を利用するSingPass Mobile

Source: BCG調査

1.1 主要デジタルIDプラットフォームとの比較 (3/3)

比較項目	MOSIP	X-Road	CODEX
ライセンス	• オープンライセンス	• MITライセンスのもとにオープンソースとして公開	• オープンライセンス
エコシステム	• デジタルIDシステムを導入するためのシステムインテグレータのコミュニティを構築 • IDソリューション技術者、生体認証・デバイスベンダーを招いたワークショップを実施 • IDソリューション・エンジニア、生体認証やデバイスのベンダーを招いてMOSIPとの統合を実習するワークショップを実施 • 各国が独自にベンダーを調達し、MOSIPがコアテクノロジーのトレーニングと教育、メンテナンスといったサービスの実施	• X-Roadの導入を支援してくれるテクノロジーパートナーが、拡張機能の開発や情報システムを統合 • 開発者、ユーザー、サービス提供者などのグローバルなX-Roadコミュニティを形成し、コードのテストや改善を支援	• InnoLeapプログラム: 政府機関がアイデアをクラウドソース化し、ソリューションを共創する機会を提供 • 市民参加型プログラム「Smart Nation Co-creating with our People Everywhere (SCOPE)の実施 • GovTech/National University of Singaporeハッカソンの実施

Source: BCG調査

1.2 Aadhaarとマイナンバー制度の比較 (1/2)

	Aadhaar	マイナンバー制度
Aadhaar番号 / マイナンバー	• 12桁の個人識別番号 • 全居住者が対象 • 取得は任意	• 12桁の個人識別番号 • 全居住者が対象 • すでに全員に付番済み
Aadhaarカード / マイナンバーカード	• 紙カード • Aadhaarに登録すると、住所に書類が郵送。その一部を切り取ってカードとして利用。インターネットからダウンロードも可能 • 記載事項: 氏名、性別、生年月日、顔写真 等 • カードの提示なしでも、Aadhaar番号と生体認証で本人確認が可能	• ICチップ付き・無しのプラスチックカード • 記載事項: - 表: 氏名、性別、生年月日、住所、顔写真 等 - 裏: マイナンバー • ICチップ内の構成: 券面記載事項、電子証明書、官民が活用可能な空き容量
Aadhaar認証 / 公的個人認証	• オンライン手続きにおける確実な本人確認を行える公的サービス • 認証方法: AUA[1]は以下4つの方法でCIDRに照会、「Yes」か「No」の返答を得る ①基本情報利用: Aadhaar番号保有者から提示されたAadhaar番号と基本情報で照会 ②ワンタイム・パスワード利用: ワンタイム・パスワードがAadhaar番号保有者の携帯番号かemailアドレスに送付。Aadhaar番号保有者から提示されたAadhaar番号とワンタイム・パスワードで照会 ③生体認証利用: Aadhaar番号保有者から提示されたAadhaar番号と生体情報で照会 ④上記の組み合わせ	• オンライン手続きにおける確実な本人確認を行える公的サービス • 電子証明書の形で提供。市区町村窓口で、マイナンバーカード内に記録してもらい取得 • 電子証明書には以下の2種類 ①署名用電子証明書: インターネット等で電子文書を作成・送信する際に利用。作成・送信者が本人であることを証明 ②利用者証明用電子証明書: インターネットサイトやコンビニ等のキオスク端末等にログインする際に利用。ログインした者が本人であることを証明

1. AUA (Authentication User Agency): UIDAIからAadhaar認証サービスの利用を認められた機関。CIDR (Central Identity Data Repository): 個人情報を保有・管理するデータベース
Source: 各種資料を基に日本総合研究所作成

1.2 Aadhaarとマイナンバー制度の比較 (2/2)

Aadhaar		マイナンバー制度	
Aadhaar eKYC	• KYC (本人確認義務) を果たせるサービス　※原則オフライン • AUA[1]は、Aadhaar番号保有者から提示されたAadhaar番号と、スキャナーで読み取ったAadhaar番号保有者の生体情報をCIDRに照合。一致すると、CIDRから基本情報の提供を受ける	なし	-
eSign	• Aadhaar番号保有者が電子文書への署名を可能とするサービス • Aadhaar eKYCによって、署名者の本人確認を実施	電子署名	• 署名用電子証明書によって署名者の本人確認を実施
DigiLocker	• 各種公的書類・証明書をオンライン上で発行・保存・共有するための個人用サイト • スマートフォンでの利用が基本 • 事前にアプリをスマートフォンにダウンロードし、携帯電話番号とAadhaar番号で登録手続き • アクセス方法: ①Aadhaar番号とワンタイム・パスワード、 ②ユーザーネームとパスワード、 ③FacebookID、のいずれかを入力 • 主なサービス ①自分で書類・証明書のアップロード ②アップロードした書類への電子署名 (eSign) ③行政機関による電子書類・証明書の発行 ④書類・証明書の外部との共有	マイナポータル	• 自分に関する行政サービスを確認・利用するための、オンライン上の個人用サイト • アクセス方法: パソコンにリーダライターを取り付け、マイナンバーカードを挿入し、暗証番号を入力[2)] • 主なサービス ①情報提供等記録表示: 行政機関同士による自分の個人情報のやりとりを確認 ②自己情報表示: 行政機関等が保有する自分の特定個人情報を確認 ③お知らせ: 行政機関等からのお知らせ ④民間送達サービスとの連携: 民間からのお知らせ ⑤子育てワンストップ・サービス ⑥公金決済サービス

1. AUA (Authentication User Agency): UIDAIからAadhaar認証サービスの利用を認められた機関。CIDR (Central Identity Data Repository): 個人情報を保有・管理するデータベース
2. スマートフォンの一部の機種にはマイナンバーカードのICチップの読み取り機能があり、それによってスマートフォンでも同様にマイナポータルを利用することが可能
Source: 各種資料を基に日本総合研究所作成


MOSIP


経済産業省