

目次

1. はじめに
2. 現行情報システムの概要
3. 実証アルファ版の構成概要
4. 本調達でのシステム方針
5. 非機能要件

はじめに

概要

本調達におけるシステム面での現行システムの説明と要求事項に関してを記載する。

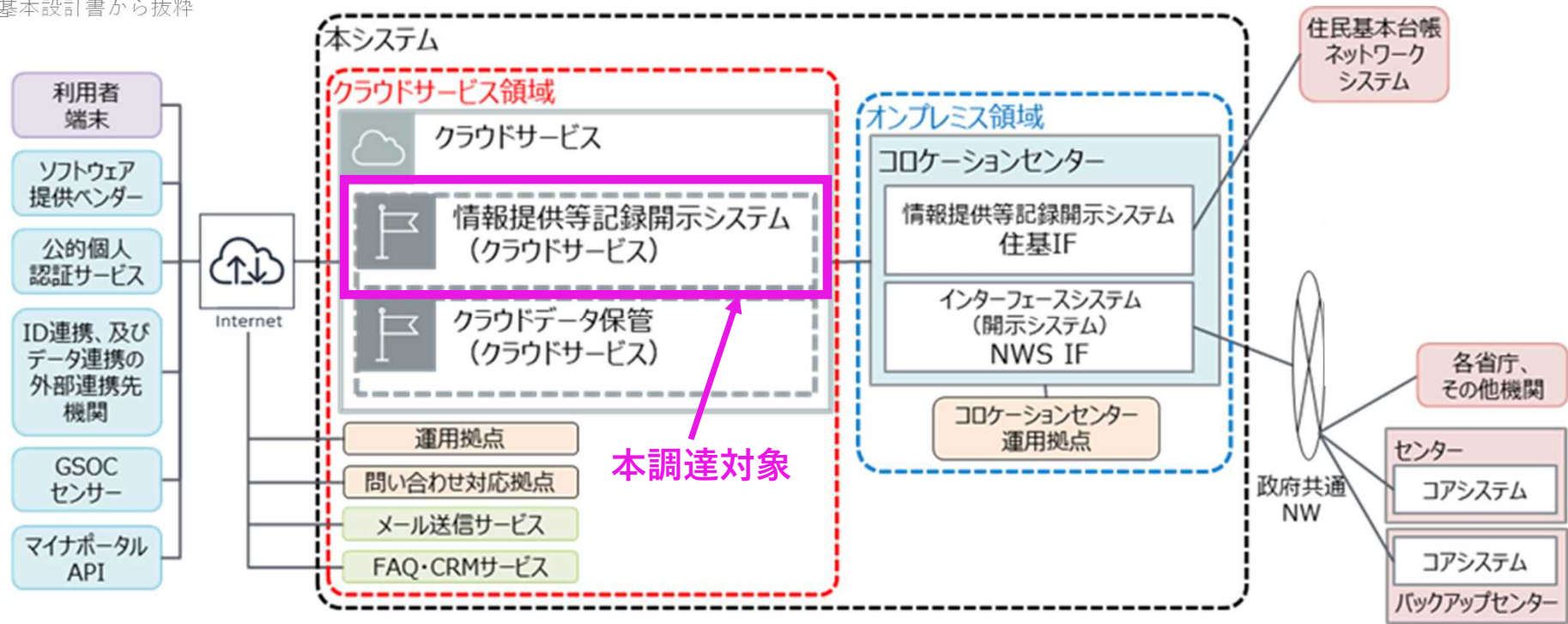
- 想定対象
 - プロジェクトマネージャー
 - システムアーキテクト
 - 開発エンジニア
- 前提事項
 - 既存システムの詳細資料（API仕様、非機能要件定義書等）は資料閲覧対象とする。
 - 方針も含め対話の上で詳細を決定するものとする。

現行の情報システム稼働環境の概要

全体構成

現行の情報提供等記録開示システム（以下、本システム）の全体構成は以下となる。

※基本設計書から抜粋

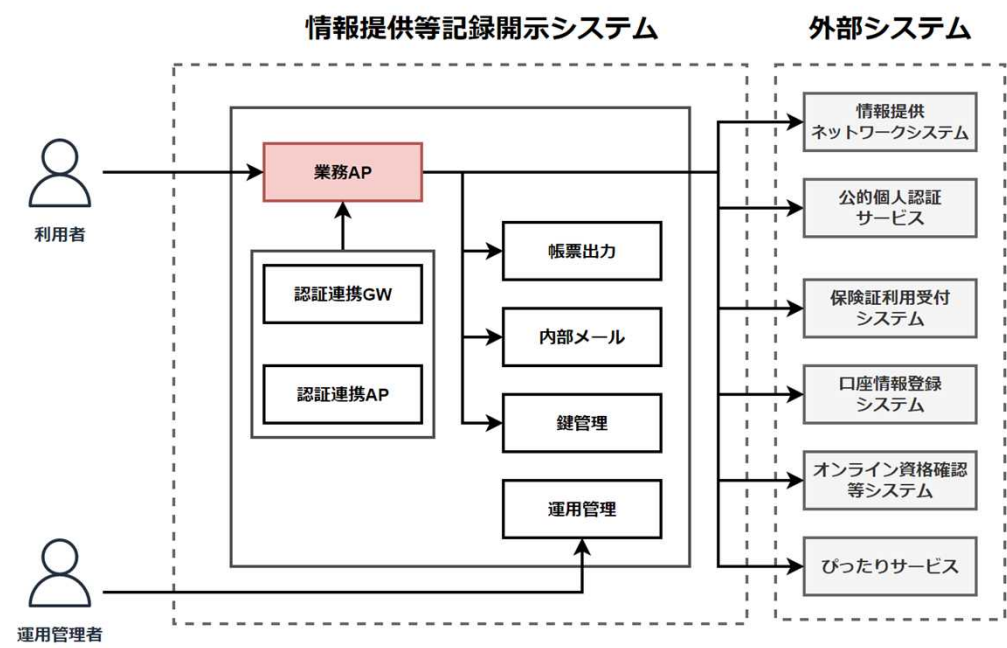


本事業における調達範囲は、枠線にて明示した情報提供等記録開示システム（クラウドサービス）の一部が対象となる。

現行の情報システム稼働環境の概要

情報提供等記録開示システム（クラウドサービス）構成

本システム内の情報提供等開示システム（クラウドサービス）内のおおまかな論理構成は以下となる。



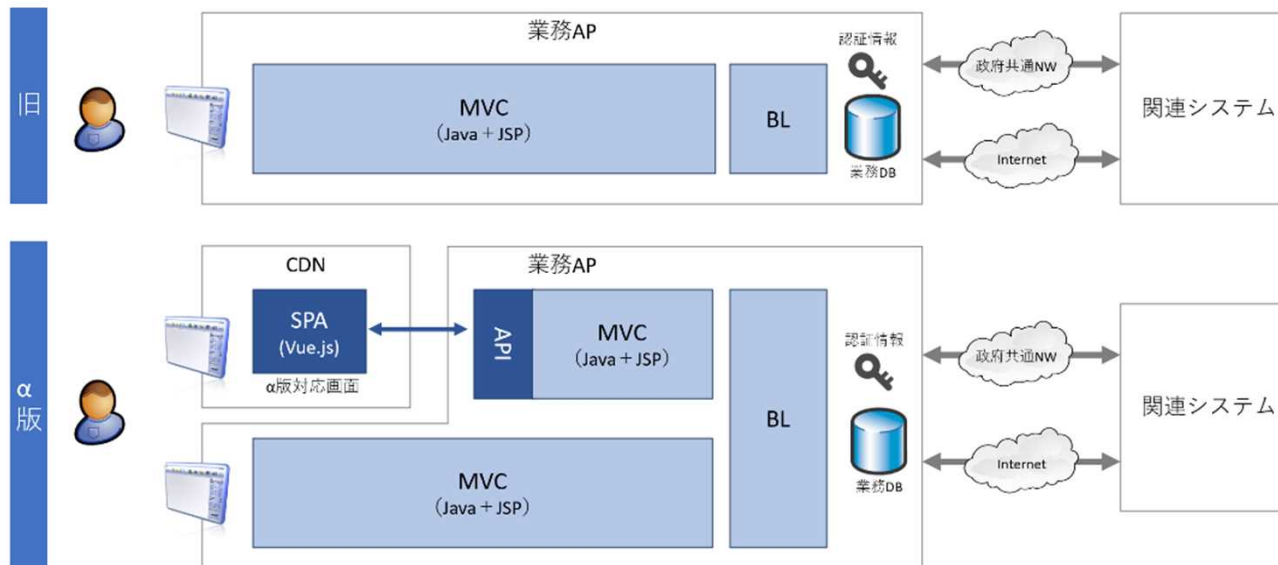
情報提供等開示システム（クラウドサービス）は内部にさまざまな機能を持ち、それらが外部システムと連携している。本事業ではその中の、業務APと呼ばれる機能を対象とする。

業務APは利用者端末からの処理を受け、インターフェースシステム（開示システム）、公的個人認証サービスなどと連携して各々の処理要求結果を返却するものである。

実証アルファ版の構成概要

新旧構成

本事業に先立ち、実証アルファ版として一部の機能をフロントエンド・バックエンドアプリケーションの分離を行った。その際の構成は以下となる。

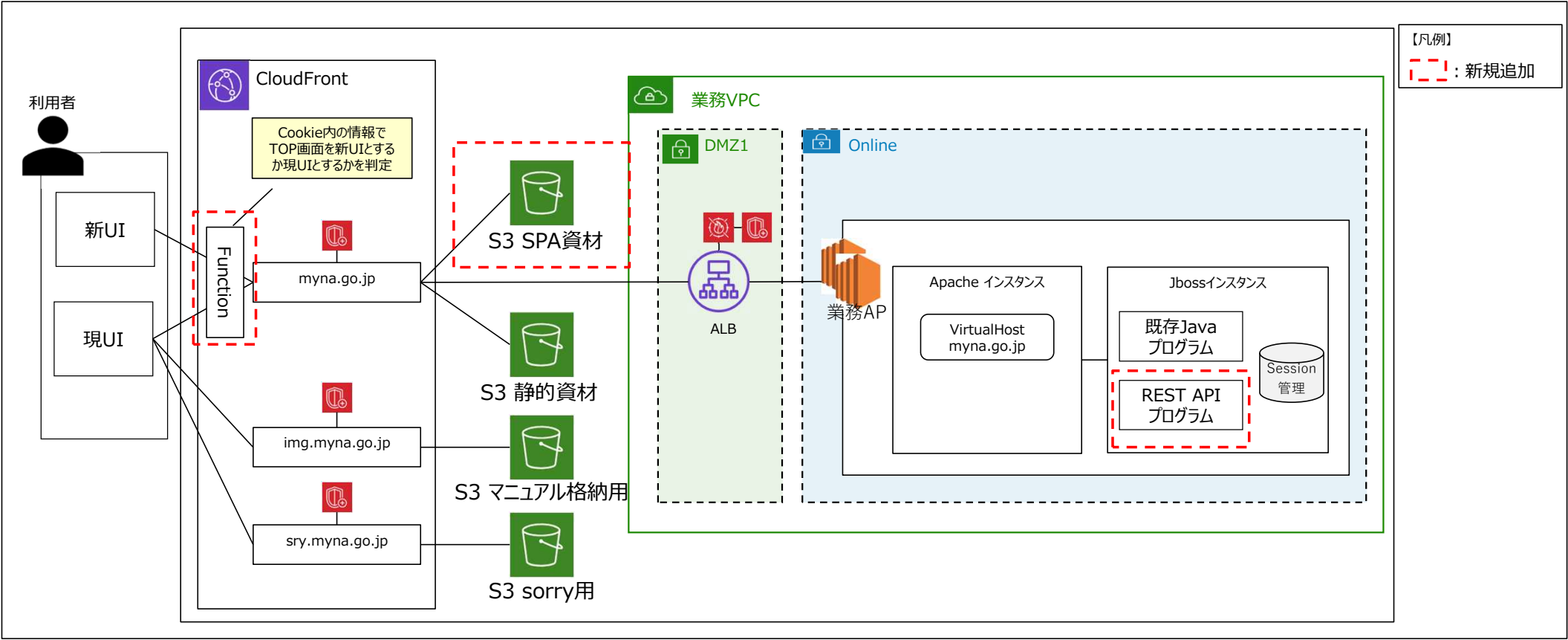


- 既存の業務APを改修しフロントエンドアプリケーション向けAPIを作成
- SPAを新設しAPIと連携し画面を描写する形式を採用
- 一部の機能以外は、従来のバックエンドから直接画面を返却する構成が残存

実証アルファ版の構成概要

インフラ構成

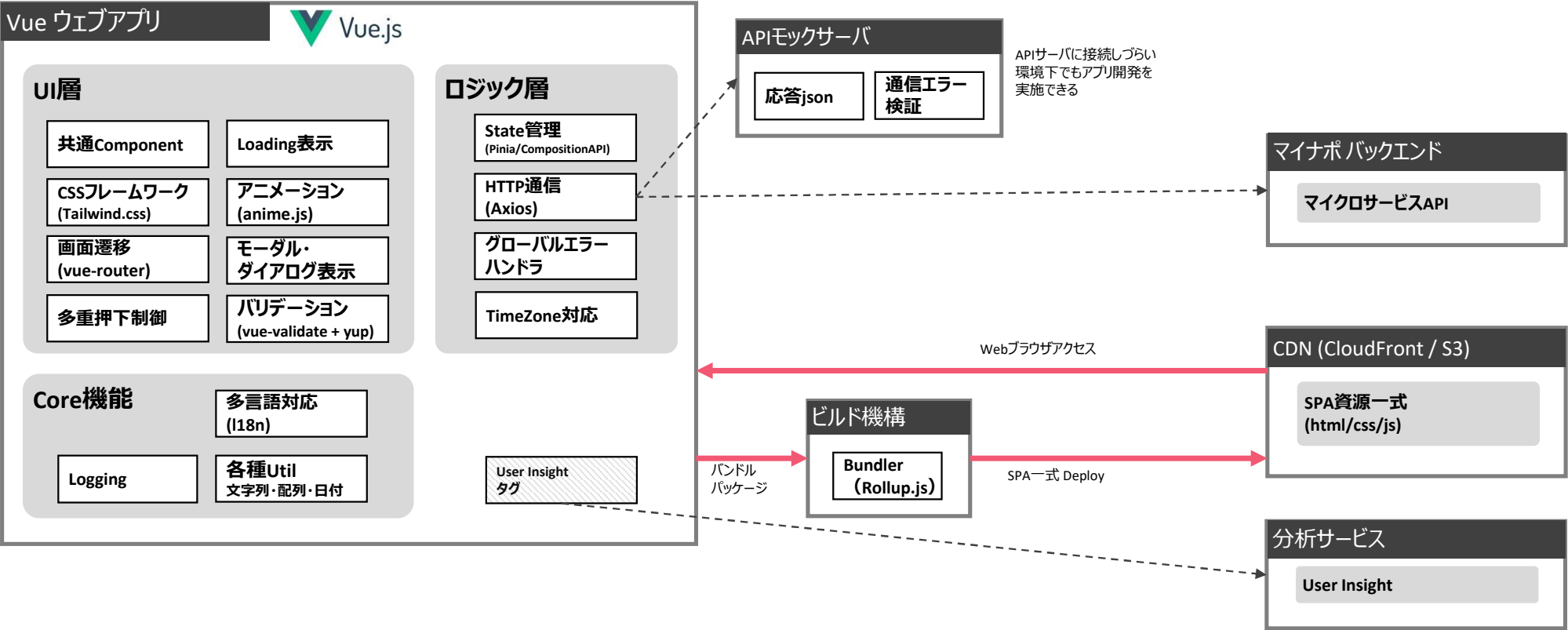
本事業に先立ち、実証アルファ版として一部の機能をフロントエンド・バックエンドアプリケーションの分離を行った。その際の構成は以下となる。



実証アルファ版の構成概要

SPA 技術構成

実証アルファ版におけるフロントエンドアプリケーションの技術構成は以下となる。



本調達でのシステム方針

アーキテクチャの方針

本事業では実証アルファ版の結果を踏まえ、その延長線上でフロントエンド・バックエンドアプリケーションの分離をより広範囲に適応する方針とする。

- バックエンド API は既存のマイナポータル運用管理事業者にて開発を行う
 - 調達事業者・既存運用管理事業者・当庁と協議の上で、外部インターフェースを策定し、外部インターフェースを責任分解点とした上で開発を行うこと
 - この前提はユーザエージェントからバックエンドAPIの間に、BFFなどの中間レイヤを調達事業者が設けることは妨げない
- 認証方式は既存のマイナポータルの発行するセッションクッキーを用いるものとする
 - 将来的に関連調達である個人認証アプリケーションを用いた認証機能を移行するため、認証機能は本事業ではスコープ外とする
 - この前提に伴い、ドメインの移行は本フェーズではスコープ外とする
 - 将来的な認証機能の移行を前提とした提案であることが望ましい
- 先行して実施した実証アルファ版の成果物を考慮し設計すること
 - 同等機能を再実装する場合に限り、提案内容に沿った形での再構築は妨げない

アプリケーション設計時の方針

アプリケーション設計時には以下の方針を考慮し設計すること。

- デジタル庁デザインシステム（注1）をベースとした、宣言的UIとコンポーネント指向によるUI構築
- バックエンドアプリケーションとの境界を分離し、API定義を基にしたモックベースの開発環境の採用
- 技術環境の変化を前提とした継続的なライブラリ更新、およびリリース時の起点となる自動テストの構築
- UX観点でのパフォーマンスの最適化を継続的に行うための指標の定義、およびデータの収集
- SEO に配慮した各種メタデータの提供

注1: 参考URL <https://www.digital.go.jp/policies/servicedesign/designsystem/>

本調達でのシステム方針

対話における技術的論点

本調達範囲のアーキテクチャについてを以下の観点で対話にて協議する。

1. BFF の有無

1. 実証アルファ版では CSR のみとしたが、SSR/SSG/ISRなどを視野に入れた最適な構成を協議して決定する。

2. マイナポータル既存運用事業者との連携と責任分界点

1. アプリケーションレベルでは HTTP API を責任分界点とする。
2. インフラにおいて既存の AWS アカウントと連携する必要がある適切な連携と責任分界点を協議して決定する。

3. アプリケーションの配信方式とリリース工程

1. 既存の AWS と連携し、どのようにアプリケーションを配信・リリースするかを協議して決定する。

4. 外部システムの提供する複雑化したデータの扱い

1. 歴史的経緯によりそのままでは表示上扱いにくい元データが一部存在するため、UI要件に沿った構造へと変換する処理を今回構築するシステム内に設ける必要がある。
2. 詳細に関しては、元データの定義書を共有し対話により方針を協議して決定する。

非機能要件

非機能要件

原則マイナポータルで定義している非機能要件を踏襲するが（関連資料は資料閲覧可能）、本事業の影響を考慮し当庁と見直しと取り決めを行う。決定した内容を考慮した設計・構築を行うこと。

ここでは本事業にて特記すべきものを記載する。

非機能要件

性能要件

- 当庁と協議の上で全体の性能上の指標、および計測に用いる端末を決定し順守すること。
 - 特定の機能（依存するAPI呼び出し数）などを考慮し実現が困難な場合は協議の上で指標を緩和する。
 - 計測時には Web Vitals 等の定量的な指標を設けるものとする。
- 計測は検証時のみでなく利用者の情報も収集する仕組みを設けること。
 - リリース後に劣化が確認された場合は、当庁と協議の上でバックログとして管理すること。
 - バックエンドアプリケーションとの責任分解を明確化するために、呼び出しに要した時間も含め情報を収集すること。

非機能要件

テスト要件

- 単体テスト・アプリケーション内の結合テストを作成し、CI上で自動化すること。
- 初回リリースおよびセキュリティ上重要な変更を加える際には、当庁のセキュリティチームによる脆弱性診断も併せて実施するものとする。
- 性能テストでは当庁と協議の上で決定する端末・OSバージョンにて実施するものとする。
- E2E/UI/リグレッションテストは自動化することが望ましい。
- 自動化にあたりバックエンドアプリケーション側での対応が必要な場合は、本調達事業者・既存運用事業者・当庁の三社で協議の上で実現方針を決定する。

非機能要件

技術的要件 1/2

- **Git** を用いたソース管理を行うこと
- デジタル庁の指定するソースコードコラボレーションツールを使用すること
- コラボレーションツール上ではプルリクエストを介した開発を行うこと
- プルリクエストに対しては、変更理由を追跡可能とするため基本的にチケットを紐付けること
 - プルリクエストは実装者以外のレビュー・承認をへてマージするよう設定すること
- 依存パッケージ・ライブラリは使用言語が提供する標準のツールを用いてバージョン管理すること
 - 更新時はロックファイルなどを用いて依存バージョンを固定すること
 - 依存パッケージ・ライブラリはリリースサイクルに併せて定期的に更新すること
- 開発開始前にコーディング規約を定義し、静的解析ツールを用いて機械的に検証・修正する仕組みを設けること
 - 静的解析ツールは開発者の作業環境下と、**CI** ツール上の両方で実行すること
 - 静的解析ツールの定義ファイルは同一リポジトリ内でバージョン管理すること
- 開発開始前に開発者の **IDE** / エディタにスペルチェックプラグインを導入すること

非機能要件

技術的要件 2/2

- 社内の独自フレームワークは使用しないこと
- 社内の独自ライブラリに関しては問わないが、それらも含めてソースコードを納品すること。
(納品物の著作権は仕様書の規定に基づき扱うものとする)
- 可能な限り **OSS** のライブラリを使用し、必要外の実装を避けること
 - 使用する主要な **OSS** ライブラリの候補が決まった段階で、当庁職員のレビューを受けること
- 依存物のライセンスを管理すること
 - アプリケーションの配布形式に応じて、許可リスト以外のライセンスが混入しないよう管理すること
- 文言は **I18N** を前提として管理すること
- **API** やプロトコルは極力標準化されているものを採用すること
- **API** のスキーマ定義 (**OpenAPI**等) を定義し、スキーマをインターフェースとした開発・運用を行うこと
- セキュリティバイデザイン・プライバシーバイデザインを考慮した設計・実装とすること