

OpenFRUCtoSサーバのご紹介

2021年3月24日

株式会社ケーアイエス 小西 由貴範

アジェンダ

1. OpenFRUCtoSの概要

1. 開発の経緯
2. 提供体制
3. 提供物
4. ゲートウェイの特徴と提供形態
5. サーバの特徴と提供形態（他製品との比較）
6. サポートするFHIRバージョン
7. 想定するユースケース

2. OpenFRUCtoSの機能

1. ゲートウェイ
2. サーバ
 1. システム全体構成
 2. インターフェースレイヤの機能
 3. サービスレイヤの機能
 4. リポジトリレイヤの機能

3. 今後の予定

1. ロードマップ
2. リリース内容

1.1 開発の経緯

- **OpenFRUCtoSとは**

- 日本固有のプロファイルをサポートするFHIRサーバ

- **開発の目的**

- HL7 FHIR規格の普及を推進するため、無償で導入できるプロダクトとしてHL7 FHIRを展開し国内の医療機関等のシステムへ実装を進めていくことで、誰でも医療情報が容易に活用できる世界の実現を目指す。

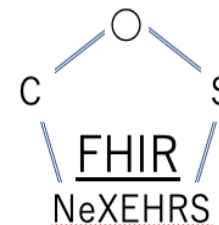
- **経緯**

- OpenFRUCtoSサーバ開発は、東京大学とソフトバンク株式会社による共同研究「健康医療情報プラットフォーム構築に関する研究」の研究成果物として位置付け、HL7 FHIRの日本における実装を最小限のコンポーネントとして有するEditionと、加えて有用な関連機能、システム、ツール、ドキュメントおよびサポートを有するEditionの構成で進めている。

1.2 提供体制

• 提供体制

- FRUCtoS Japan (フルクトース・ジャパン)
(任意団体)



- ソフトウェアサイト URL : <https://open.fructos.jp>
 - 運営者 : FRUCtoS Japan (フルクトース・ジャパン)
 - 代表者 : 東京大学・ソフトバンク (株) 共同研究ユニット
 - サポーター :
 - 株式会社ケーアイエス
 - 日本総合システム株式会社
 - NeXEHRsコンソーシアム
 - 事務局 : NeXEHRsコンソーシアム内
-
- Bリリース
 - ダウンロード開始日 : 2021年1月20日
 - ダウンロード数 : 30 件 (2021年3月4日時点)

1.3 提供物

・提供物と2つのエディション

・ゲートウェイ

- ・ FHIRサーバへのアクセスに対するクライアント認証機能を追加するためのアドオンシステム

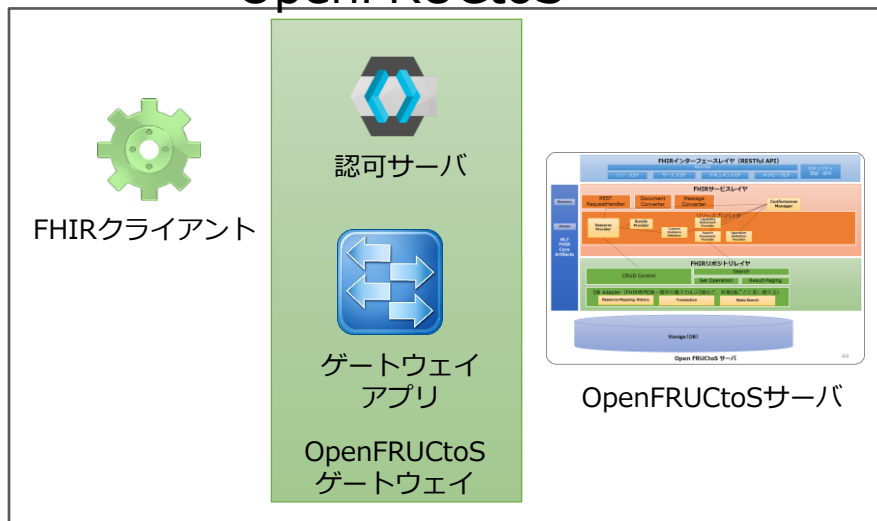
・サーバ

- ・ 日本固有のプロファイルに対応したFHIRサーバ
- ・ Webアプリケーションとして動作

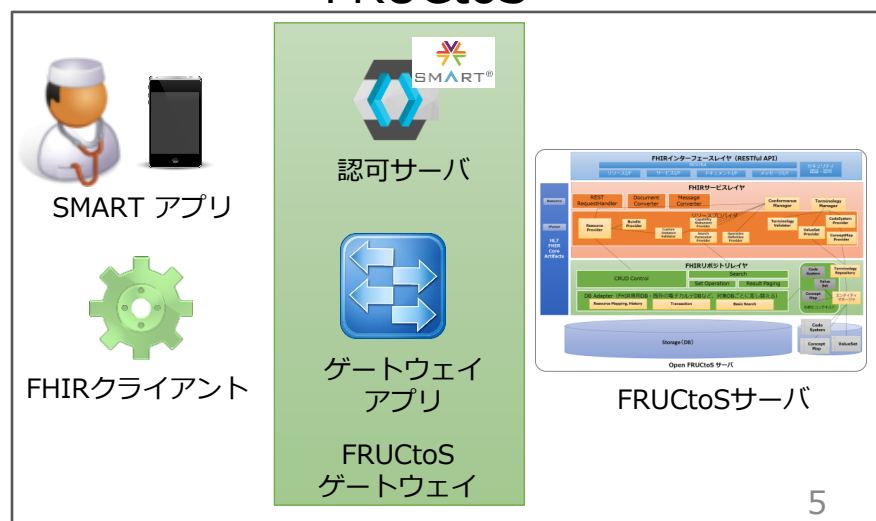
・エディションによる構成の違い

- ・ 詳細は、P.6、P.7に記載

OpenFRUCtoS



FRUCtoS



1.4 ゲートウェイの特徴と提供形態

- 認可サーバとしてのKeyCloakとAPIゲートウェイ（ゲートウェイアプリ）から構成される
- OpenFRUCtoS :
 - OAuth2.0による基本的な認証・認可機能をサポート
- FRUCtoS :
 - SMART on FHIRをサポート

OpenFRUCtoSとFRUCtoSの主な違い

	OpenFRUCtoS	FRUCtoS
使用許諾料	商用利用：有り 商用利用以外：無し	有り
利用サポート	有償サポート有り	有償サポート有り
OAuth2.0対応	OAuth2.0 Client Credentials のみ	OAuth2.0 Authorization Code Grant のサポート
SMART on FHIR対応	スコープ非対応 Smart App Launchフレームワーク一部対応	スコープ制御の対応 Smart App Launchフレームワーク1.0.0 のサポート

1.5 サーバの特徴と提供形態

- OpenFRUCtoS
 - HL7コアとJPプロファイル（HL7 FHIRの日本における実装）をサポート
 - 非営利利用は無償で提供予定
- FRUCtoS
 - JPプロファイルに限らず任意のプロファイルをサポート
 - 有用な関連機能、システム、ツール、およびサポートを有する
 - 有償提供・有償サポート

OpenFRUCtoSとFRUCtoSの機能比較

	OpenFRUCtoS	FRUCtoS
使用許諾料	商用利用：有り 非営利利用：無し	有り
利用サポート	有償サポート有り	有償サポート有り
サポートするFHIRバージョン	最新のPublicationバージョンにのみ対応	複数バージョンに対応可能
サポートプロファイル	JPプロファイルのみ	任意のプロファイルをサポート
パラダイム	RESTful APIのみ	全て

1.5 サーバの特徴と提供形態

OpenFRUCtoSとFRUCtoSの機能比較（続き）

		OpenFRUCtoS	FRUCtoS
基本機能	データ型	基底仕様（サポート対象リソース） + JPプロファイルに含まれるデータ型のみ	任意のデータ型をサポート
	エンコーディング	JSON	JSON、XML
	拡張	基底仕様（サポート対象リソース） + JPプロファイルに含まれる拡張のみ	任意の拡張をサポート
	検索パラメータ	基底仕様（サポート対象リソース） + JPプロファイルに含まれる検索パラメータのみ	任意の検索パラメータをサポート
	永続化	基底仕様（サポート対象リソース） + JPプロファイルに含まれるリソースの永続化のみ	任意のリソースの永続化をサポート
	Terminologyサービス	外部Terminologyサービスの利用機能のみ	外部Terminologyサービスの利用と、Terminology関連サービスの提供をサポート
セキュリティ	TLS(https)	サポート有り	サポート有り
	認証	OAuth2.0による基本的な機能のサポート (P.16参照)	SMART on FHIRをサポート
	認可	OAuth2.0による基本的な機能のサポート (P.16参照)	SMART on FHIRをサポート
	監査ログ	サポート無し	サポート有り
管理機能	サーバ設定	設定ファイルベース	設定画面あり
	利用者管理	サポート無し	設定画面あり
	ログ管理	ログ出力のみ	ログ出力 + ログ管理機能 トランザクションログ機能
非機能	可用性	サポート無し	サポート有り

1.5 サーバの特徴と提供形態

・他のFHIRサーバ製品との機能比較

マッピング定義を作ることで独自テーブル構造に対応可能

主なFHIRサーバ製品の比較

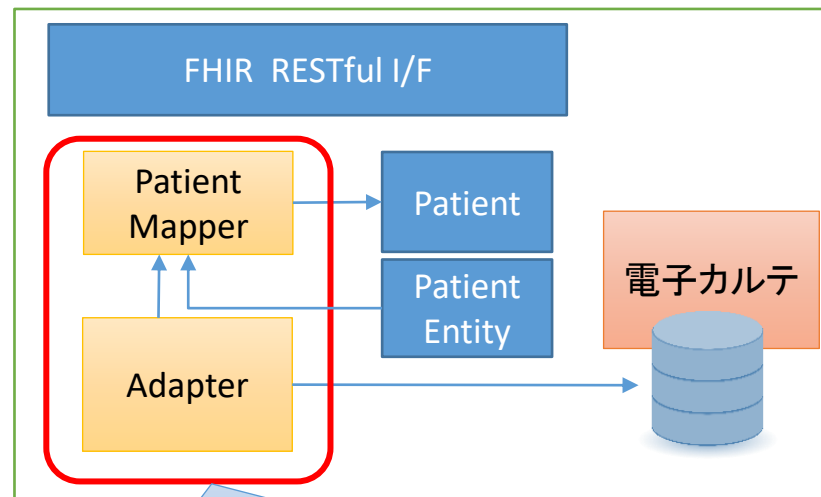
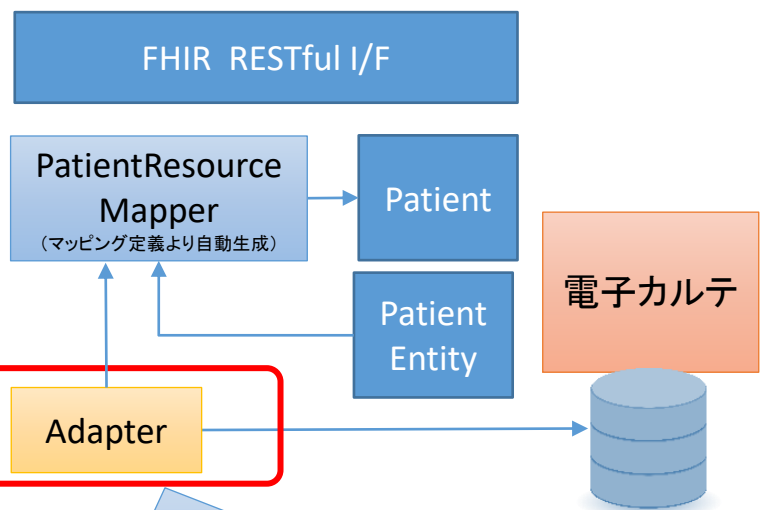
	OpenFRUCtoS	HAPI/Smile CDR
実装言語	Java (Open JDK11)	Java (JDK8以降)
ライセンス	検討中	Apache Ver.2.0
利用サポート	有り	有り : Smile CDR
対応するFHIRバージョン	R4 (4.0.1) 以降	DSTU2 (1.0.2) 以降
FHIRリソースの永続化	FHIR専用データベースアダプタと既存の電子カルテシステムに対応するアダプタを提供	リソースを1つの共通のテーブルに格納し、検索に使用するプロパティのみを別途構造化して格納
独自テーブル構造への対応	個別のデータベースのマッピング機能を提供	独自テーブル構造とのデータマッピング処理を個別に実装する必要あり
パラダイム	OpenFRUCtoS : RESTful APIのみ FRUCtoS : 全てのパラダイムを対象	RESTful API のみ
用語サービス	サポート	サポート

1.5 サーバの特徴

- 典型的なFHIRファサードと呼ばれるアプリケーションを例にすると、既存の電子カルテをFHIR API対応とする場合、以下のような違いがある

OpenFRUCtoS/FRUCtoS

他製品 (HAPI)



- 電子カルテのデータモデルとFHIRのモデルのマッピングを支援するためのツール群が用意されており、少ない実装量で実現できる

- 電子カルテのデータモデルとFHIRのモデルをマッピングするためのアダプター及びマッパーの実装を、電子カルテのデータモデルやサービスI/Fにあわせて個別に実装しなければならない

1.6 サポートするFHIRバージョン

• FHIRバージョンの考え方

• FHIRバージョンの構成： Publication.Major.Minor

Date	Version	Description
Current Versions		
2019-10-30	4.0.1	FHIR Release #4: First Normative Content
(current)	(last commit)	Current Development build (about 30min behind version control, may be incoherent and change rapidly)
R5 Sequence (Work in Progress)		
2020-05-04	4.4.0	FHIR Release #5: Preview #2
2019-12-31	4.2.0	FHIR Release #5: Preview #1
R4 Sequence (Current)		
2019-10-30	4.0.1	FHIR Release #4 First Normative Content with 1 technical errata (Permanent Home) <ul style="list-style-type: none"> • <i>Technical Errata Archive (zip): v4.0.0</i> (Permanent Home)
2018-11-09	3.5a.0	Special R4 Ballot #3 : Normative Packages for Terminology / Conformance + Observation
STU 3 Sequence (Historical)		
2019-10-24	3.0.2	FHIR Release 3 (STU) with 2 technical errata (Permanent Home) <ul style="list-style-type: none"> • <i>Technical Errata Archive (zip): v3.0.1</i> • <i>Technical Errata Archive (zip): v3.0.0</i>

4.0.1: **Current Publication**
最新のPublicationバージョン

4.4.0: R5系のプレビュー版#2

4.0.1: R4系の最新
v4.0.0の技術的修正版

FHIRバージョンの構成要素

構成要素	説明
Publication (パブリケーション)	✓ 更新された仕様 (FHIRのTrial Use版、または、Normative版) としてHL7がFHIRを公開するときにインクリメント
Major (メジャー)	✓ 画期的な変更 (Breaking change) がある度にインクリメント
Minor (マイナー)	✓ 1つ以上の実質的な変更 (Substantive change) を含む、更新のスナップショットリリースが作成されるたびにインクリメント

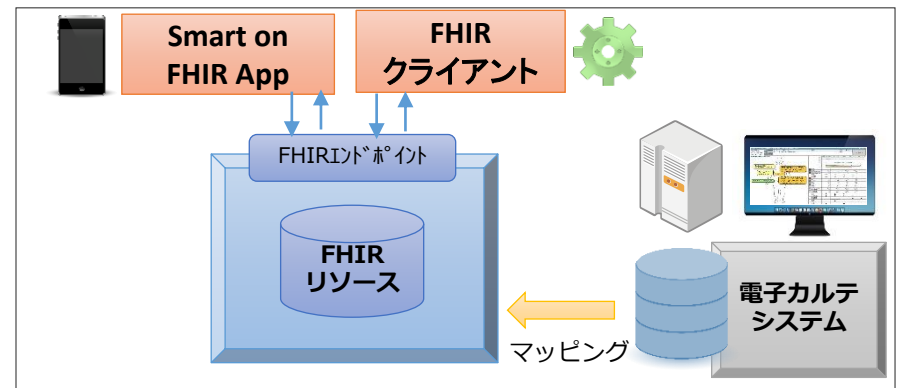
1.6 サポートするFHIRバージョン

・ FHIRバージョンサポートルール

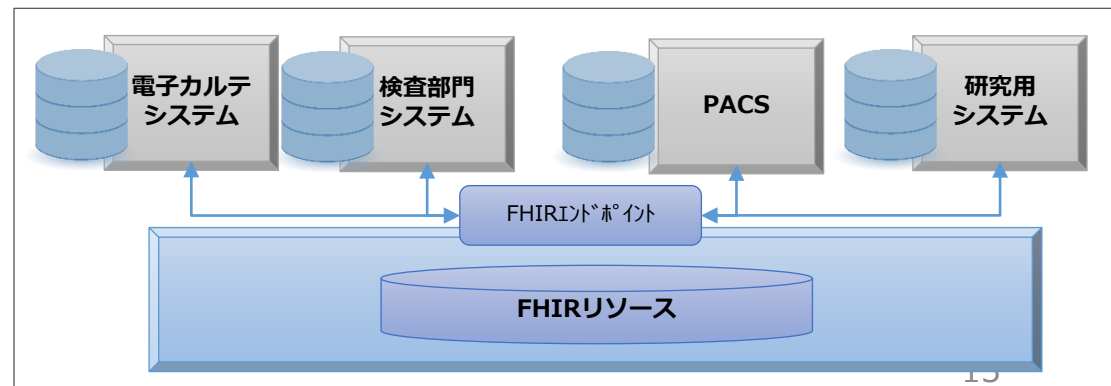
	OpenFRUCtoS	FRUCtoS
Publication バージョンのサ ポートルール	Current Publicationバ ージョンのみ をサポート 例：R5がCurrent Publication バージョンとしてリリースされ た場合、R5のみをサポート	全てのPublication バ ージョン (R4以降)をサポート 例：R5がCurrent Publication バージョンとしてリリースされ た場合、R4とR5の両方をサポ ート
Current Publication 内 のメジャーバ ージョンアップ/ マイナーバ ージョンアップ の考え方	Current Publication内で、メ ジャーバージョンアップ、が あった場合には、 その最新 バージョンのみ をサポート。 マイナーバージョンアップは サポートしない。 例：4.0.0 に対して4.1.0 が リリースされた場合には、R4 系として、最新の 4.1.0 のみ をサポート	Current Publication内で、メ ジャーバージョンアップ、また は、マイナーバージョンアップ があった場合には、 その最新 バージョンのみ をサポート 例：4.0.0 に対して4.0.1 がリ リースされた場合には、R4系と して、最新の 4.0.1 のみをサ ポート

1.7 想定するユースケース

- 1. 相互運用性インターフェース
 - 電子カルテシステムに対して相互運用性を実現するためのFHIRインターフェースとしての利用

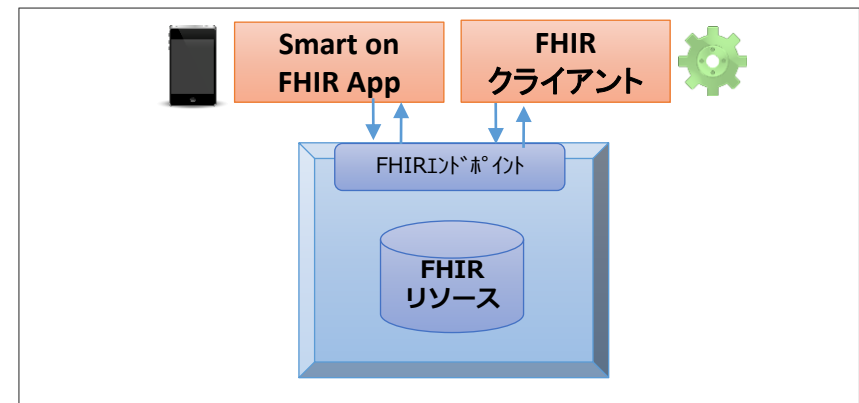
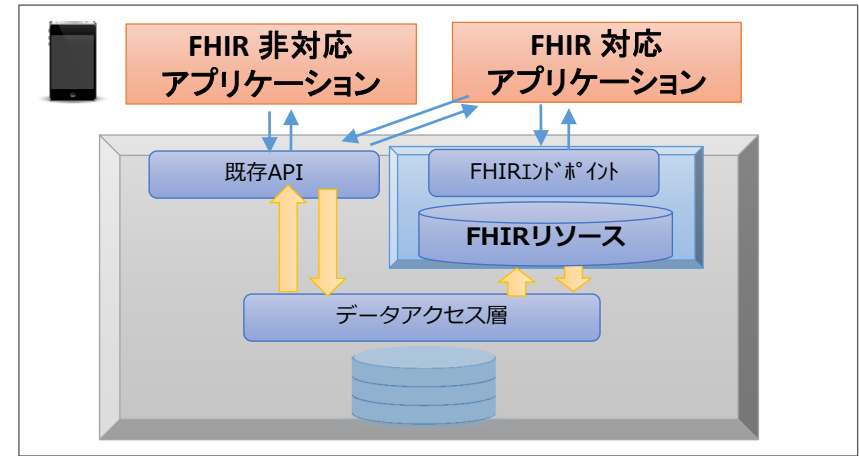


- 2. FHIR ベンダー中立リポジトリ
 - マルチベンダーシステムに対する中立なリポジトリへの一貫したAPIとしてFHIRを利用



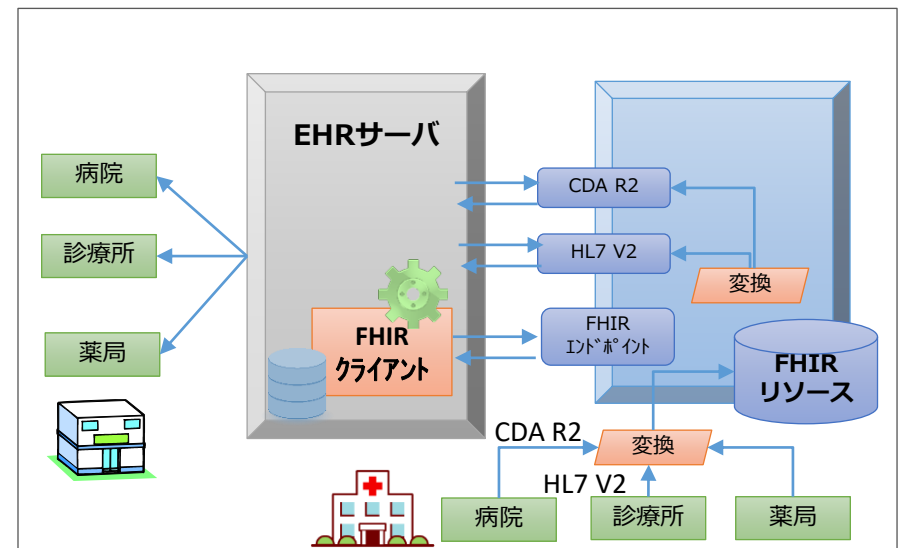
1.7 想定するユースケース

- 3. FHIR/レガシーAPIの混在
 - FHIRがまだサポートしていないデータモデルを扱う必要がある場合や、FHIR非対応のクライアントのサポートが必要な場合にFHIR APIとレガシーAPIの両方をサポートするために、既存システムにFHIRインターフェースをアドオンする
- 4. FHIR 臨床データリポジトリ
 - PHRやお薬手帳、電子処方箋など主に患者向けに診療情報を公開するシステムのFHIRネイティブなデータリポジトリとしての利用



1.7 想定するユースケース

- 5. FHIR ベース統合ハブ
 - 地域医療連携ネットワーク（EHR）サーバなどデータ交換・共有のためのデータ統合ハブとしての利用



2.1 ゲートウェイ

- 認可サーバとの連携

- OpenFRUCtoS

- クライアント・クレデンシャルズフローに基づく認可をサポート
 - OAuth 2.0の認可フローの1つ
 - 「クライアントID」と「クライアントシークレット」をクライアント側に保持し、認可サーバのトークンエンドポイントにリクエストすることでアクセストークンを取得

- FRUCtoS

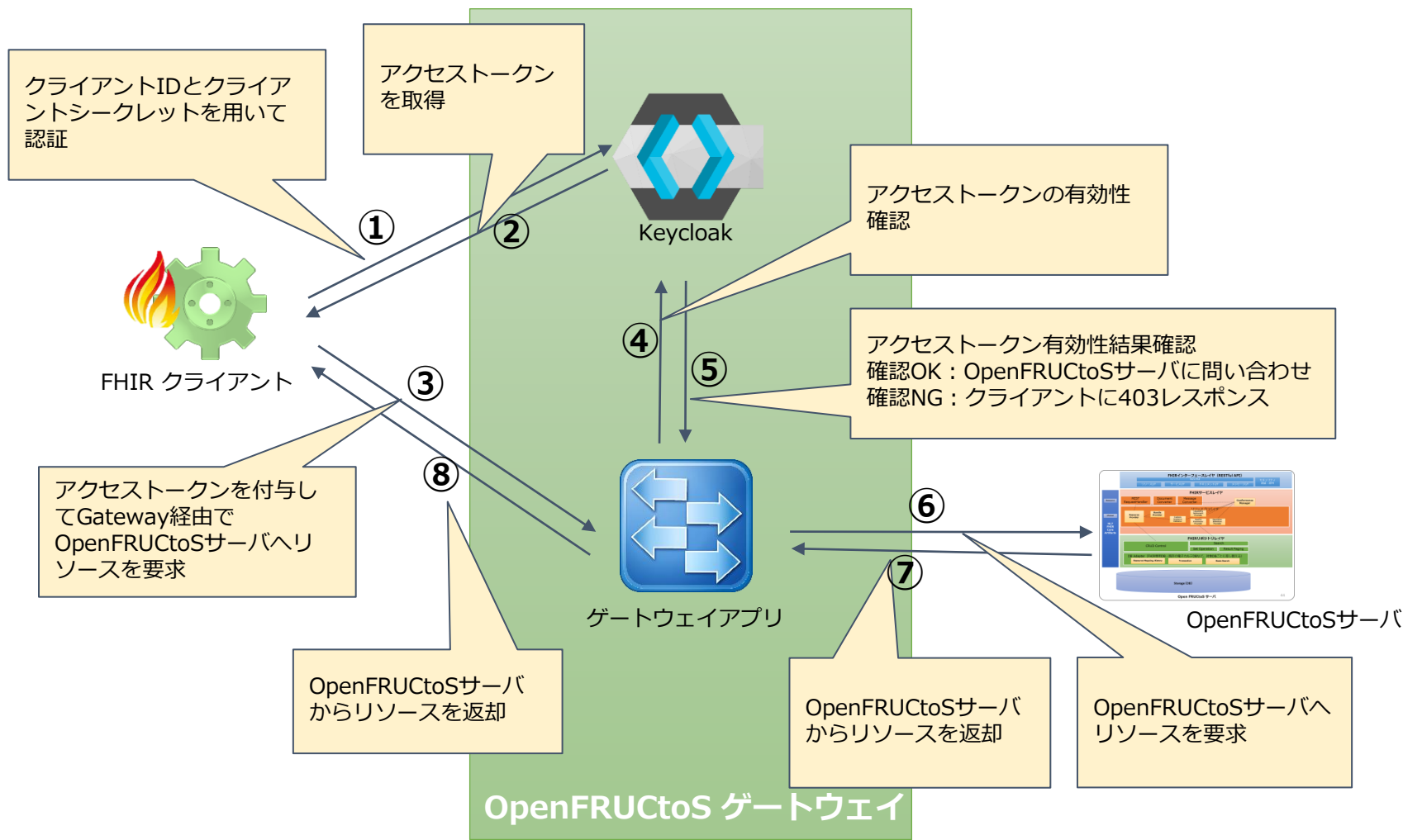
- 認可コードフローに基づく認可をサポート
 - OAuth 2.0の認可フローの1つで、SMART on FHIRのプロファイル（SMART App Launch Framework）で使用が規定されている
 - エンドユーザとの対話の中でクライアントに権限を委譲できる
 - OAuth 2.0 スコープによるアクセス制御をサポート

- FHIRサーバとの連携

- FHIRサーバのCapability Statement（またはWell-known URIs JSON ファイル）より、認可サーバのエンドポイントURLを取得（FRUCtoS）
 - FHIRサーバは、ゲートウェイを通過したリクエストのみを、認可済リクエストとして処理

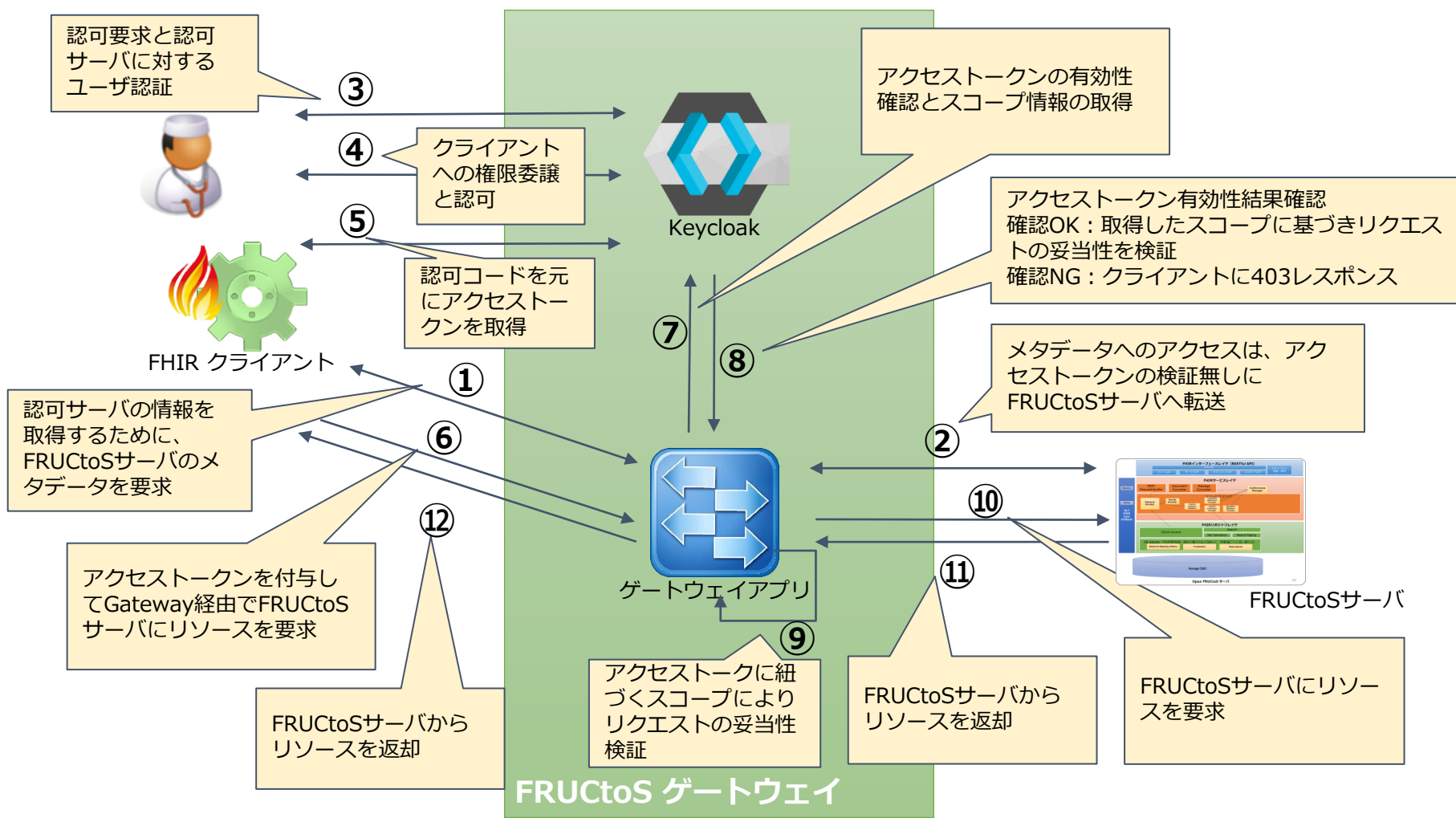
2.1 ゲートウェイ

- OpenFRUCtoSゲートウェイを使用したアクセスフロー

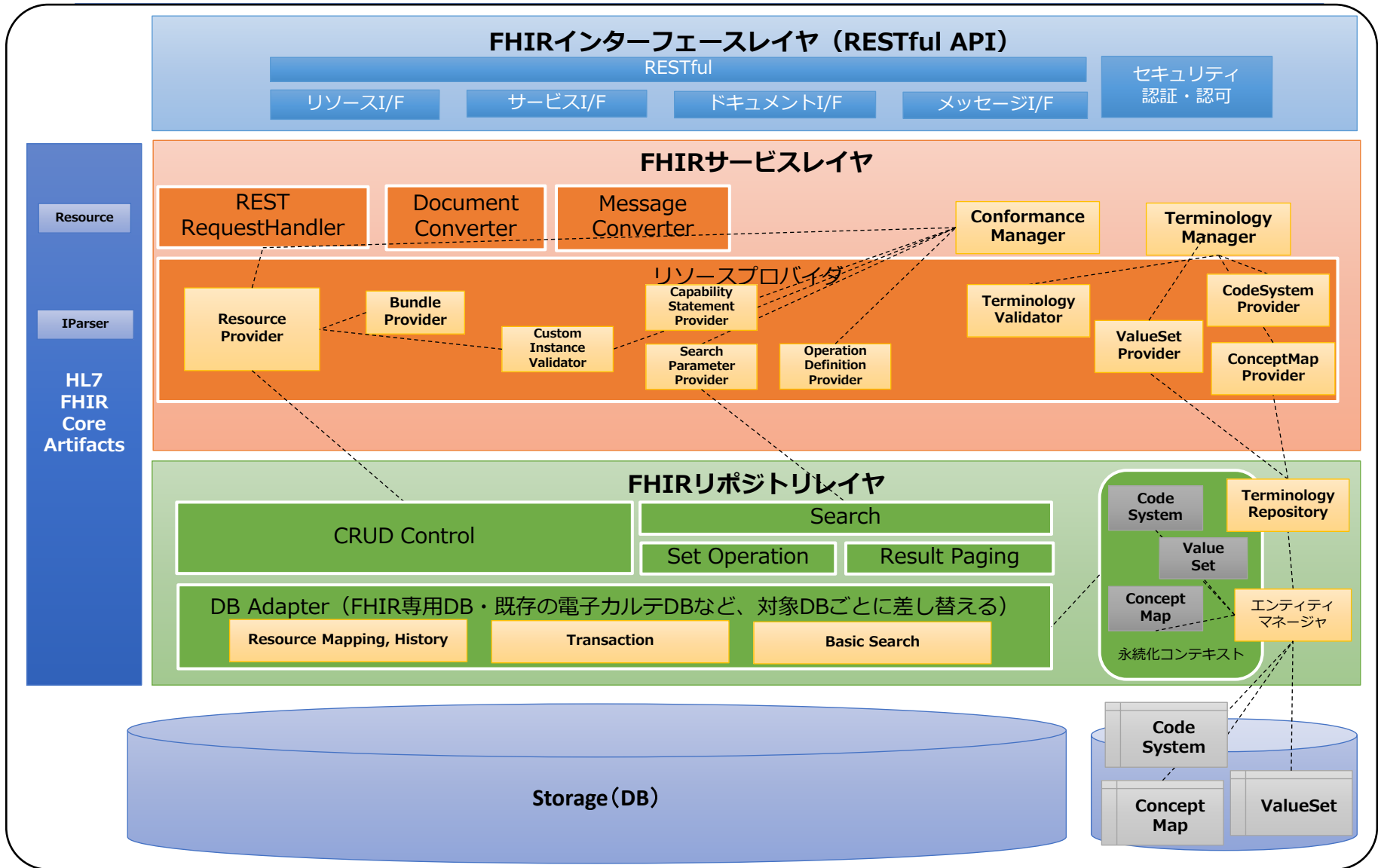


2.1 ゲートウェイ

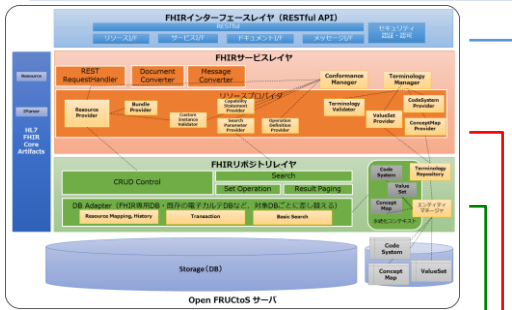
・FRUCtoSゲートウェイを使用したアクセスフロー



2.2 サーバ



2.2.1 システム全体構成



FHIRサーバーのレイヤ構成

	レイヤ	概要
1	FHIRインターフェースレイヤ	FHIRクライアントから受信したHTTP RESTful API を処理し、サービスレイヤからの処理結果を応答するレイヤ。
2	FHIRサービスレイヤ	FHIRロジックを実装。インターフェースレイヤから渡されたリクエスト情報を元に、FHIR仕様に従いFHIRリソースに変換したり、バリデーション処理を行うレイヤ。
3	FHIRリポジトリレイヤ	サービスレイヤから渡されたFHIRリソースを永続化したり、リソースに対する検索要求をSQLへ変換するレイヤ。

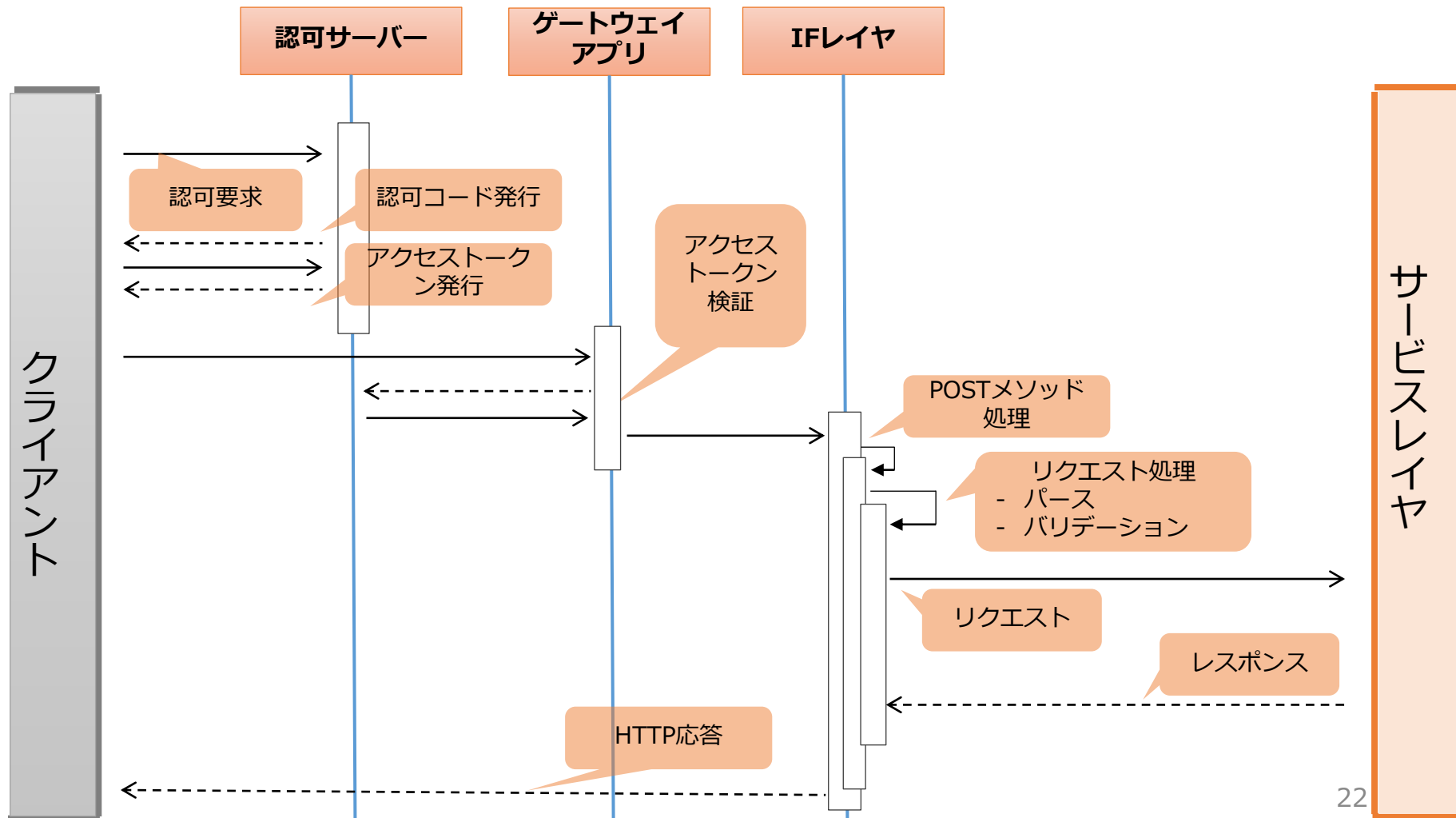
2.2.2 インターフェースレイヤの機能

・インターフェースレイヤの機能

	機能	主な機能
1	エンドポイントの管理	<ul style="list-style-type: none">• FHIRサーバーに登録されているエンドポイントの公開
2	RESTful APIのパー ス・パラメータ取得	<ul style="list-style-type: none">• URL をパースしリクエストの詳細情報やパラメータを取得• HTTPリクエストの情報をカプセル化し、 FHIRサービスレイヤに渡す
3	APIレベルのバリデー ション	<ul style="list-style-type: none">• URLの構文チェック
4	例外処理とエラー通知	<ul style="list-style-type: none">• 例外発生時のエラー処理とエラーコードの返却処理
5	Terminologyサービ ス	<ul style="list-style-type: none">• ボキャブラリサービスのためのRESTfulインターフェースを提供
6	認証・認可	<ul style="list-style-type: none">• SMART on FHIR認証フレームワークに対応するための拡張を含むCapabilityStatmentの返却処理
7	監査証跡	<ul style="list-style-type: none">• AuditEventリソースを使用

2.2.2 インターフェースレイヤの機能

- ・インターフェースレイヤ処理フロー（POST（create）の例）



2.2.3 サービスレイヤの機能

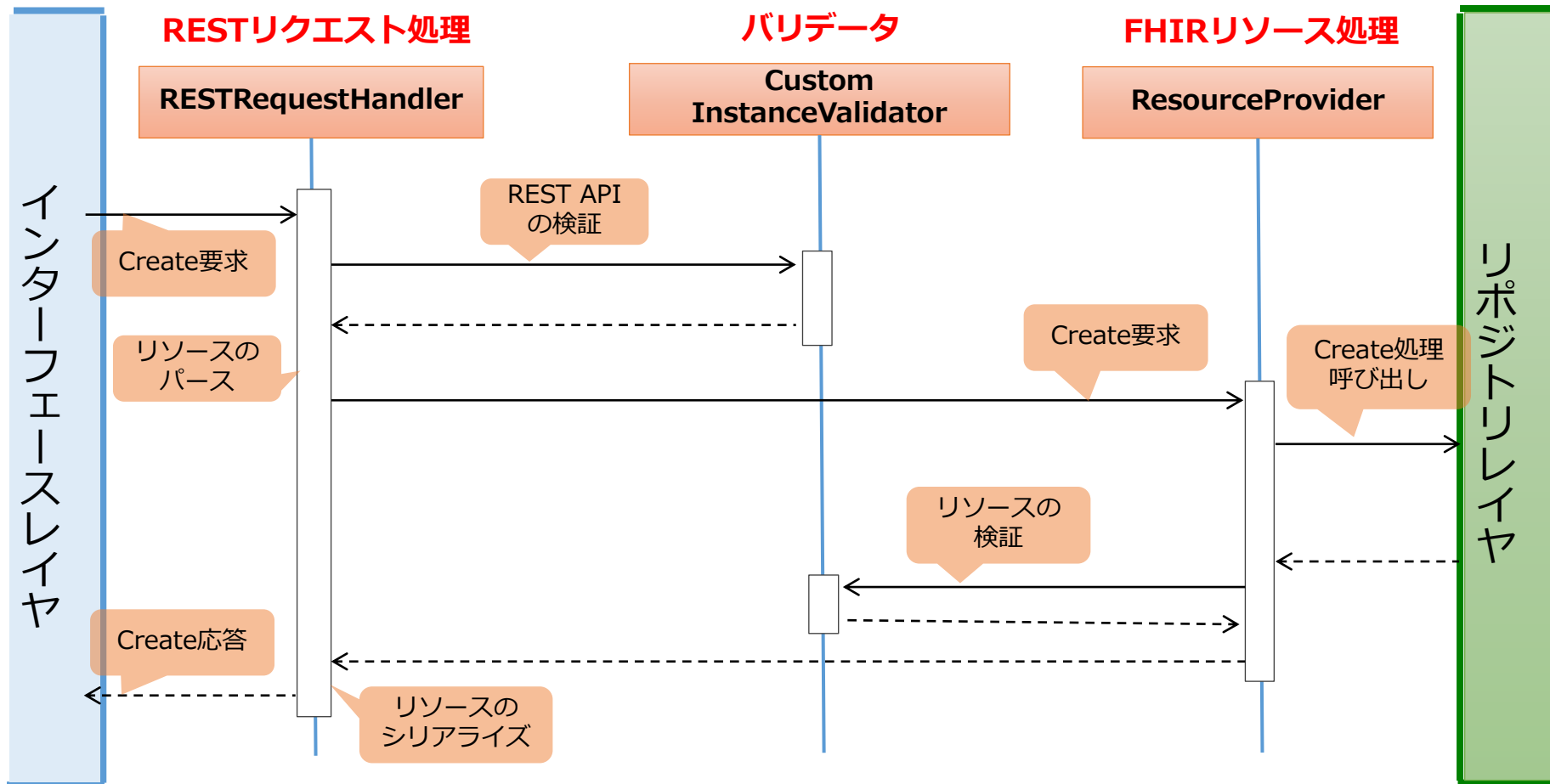
・サービスレイヤの機能

サービスレイヤの主な機能

	機能	主な機能
1	RESTリクエスト処理	<ul style="list-style-type: none">・ インターフェースレイヤから渡されたRESTful APIのリクエスト情報をFHIR仕様に従って解釈・ URLやリクエストボディのパーズ・ レスポンスへのFHIRリソースのシリアライズ
2	FHIRリソース処理	<ul style="list-style-type: none">・ FHIR仕様に準じたFHIRリソースの処理・ リポジトリレイヤの処理に必要なパラメータの受け渡し・ Bundleリソース処理・ Transaction、Batch処理に必要な情報の受け渡し
3	パフォーマンス管理	<ul style="list-style-type: none">・ サーバのCapability Statement や プロファイル (StructureDefinition) の読み込み
4	バリデータ	<ul style="list-style-type: none">・ 指定された Capability Statementに対する RESTful APIの検証・ FHIR基底仕様、または、指定されたプロファイルに対するFHIRリソースの検証
5	Terminologyサービス	<ul style="list-style-type: none">・ バリデーション処理において外部Terminologyサービスの利用・ 標準病名マスタやJLAC10/JLAC11など国内の標準ボキャブラリやローカルコード体系をリソースとして管理・ ローカルコード体系のCSVファイルによる取込み・ コードマッピングのサポート

2.2.3 サービスレイヤの機能

- サービスレイヤ処理フロー
 - 例：Createインタラクション



2.2.4 リポジトリレイヤの機能

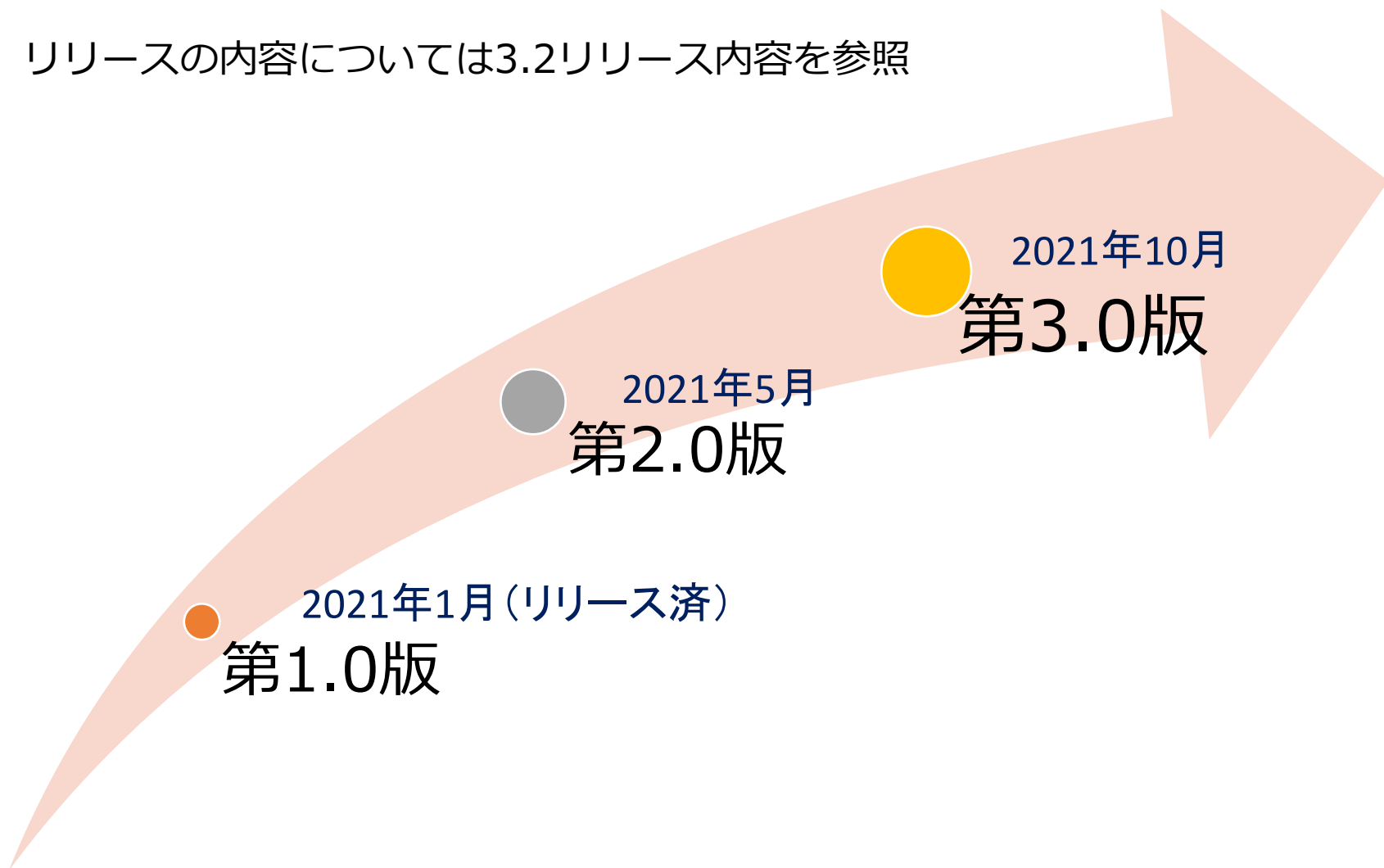
- FHIRリソースの永続化の仕組み
(FHIR専用データベースのDB Adapter)
 - FHIRの全機能の実行に対応する専用設計
 - リソース本体とhistoryをJSON形式でBLOBとして格納
 - 検索インデクスを検索対象プロパティの型ごとにRDBテーブルとして格納
 - リソースグラフ構造の整合性を保証するTransaction操作
- 検索コマンドは「単純なフィルタリング」演算へ分解して実行
例：Observation?subject.name="山田 花子"
 - Searchモジュールが、Observation.subjectが指すリソースのうち「name」という検索が存在するリソースが「Patient」であることをSearchParameterの定義から導出し、DB Adapterにフィルタ操作を依頼。
 - DB Adapterが、Patient全体から「name="山田 花子"」である要素を抽出。
 - DB Adapterが、Observation全体から「subject=前段で得られたリソース集合の要素」である要素を抽出。

2.2.4 リポジトリレイヤの機能

- データベースマッピングの概要
(既存の電子カルテシステムのDB Adapter)
 - DB Adapterの機能
 - 電子カルテDBレコードキーとリソースIDとの双方向のマッピング
 - 電子カルテDBレコードフィールドからリソースプロパティへのマッピング
(現在はFHIRインタフェースからの既存電子カルテデータの書き込みはサポートしていない)
 - 検索用の「単純なフィルタ」
 - DB Adapter開発ツール：MapperGenerator
 - 入力：
 - 電子カルテDBレコードに対応するJavaエンティティクラス
(各フィールドにマッピング方法のアノテーションを付加する)
 - 値の単位やデータ型を変換するメソッドを実装したコンバータクラス
 - 出力：
 - 読み出したエンティティのフィールドからFHIRリソースのプロパティへのマッピング
を実行するJava Classのソースコード
 - Search名と対象カラムの対応表 (制作すべきフィルタの一覧)

3.1 ロードマップ

リリースの内容については3.2リリース内容を参照



・JPプロファイルのリリースやFHIR R5に向けた進捗により、計画を変更する可能性がある

3.2 リリース内容

対象リソース

括弧内の数字は成熟度レベル

		第1.0版	第2.0版	第3.0版以降
Foundation	Conformance	<ul style="list-style-type: none">• CapabilityStatement [N]• StructureDefinition [N]• ImplementationGuide [1]• SearchParameter [3]• MessageDefinition [1]• OperationDefinition [N]• CompartmentDefinition [1]	<ul style="list-style-type: none">• GraphDefinition [1]• StructureMap [2]	<ul style="list-style-type: none">• ExampleScenario [0]
	Terminology	<ul style="list-style-type: none">• CodeSystem [N]• ValueSet [N]• NamingSystem [1]	<ul style="list-style-type: none">• ConceptMap [3]	<ul style="list-style-type: none">• TerminologyCapabilities [0]
	Security	<ul style="list-style-type: none">• Provenance [3]• AuditEvent [3]	<ul style="list-style-type: none">• Consent [2]	
	Documents		<ul style="list-style-type: none">• Composition [2]• DocumentManifest [2]• DocumentReference [3]	<ul style="list-style-type: none">• CatalogEntry [0]
	Other	<ul style="list-style-type: none">• Basic [1]• Binary [N]• Bundle [N]• MessageHeader [4]• OperationOutcome [N]• Parameters [N]	<ul style="list-style-type: none">• Subscription [3]	<ul style="list-style-type: none">• Linkage [0]

3.2 リリース内容

対象リソース

括弧内の数字は成熟度レベル

		第1.0版	第2.0版	第3.0版以降
Base	Individuals	<ul style="list-style-type: none">• Patient [N]• Practitioner [3]• RelatedPerson [2]• Person[2]	<ul style="list-style-type: none">• Group [1]• PractitionerRole [2]	
	Entities #1	<ul style="list-style-type: none">• Organization [3]• Location [3]	<ul style="list-style-type: none">• HealthcareService [2]• Endpoint [2]	<ul style="list-style-type: none">• OrganizationAffiliation [0]
	Entities #2		<ul style="list-style-type: none">• Device [2]• Substance [2]• DeviceMetric [1]	<ul style="list-style-type: none">• BiologicallyDerivedProduct [0]
	Workflow		<ul style="list-style-type: none">• Task [2]• Appointment [3]• AppointmentResponse [3]• Schedule [3]• Slot [3]	<ul style="list-style-type: none">• VerificationResult [0]
	Management	<ul style="list-style-type: none">• Encounter [2]• List [1]	<ul style="list-style-type: none">• Flag [1]• Library [2]• EpisodeOfCare [2]	

3.2 リリース内容

対象リソース

括弧内の数字は成熟度レベル

		第1.0版	第2.0版	第3.0版以降
Clinical	Summary	<ul style="list-style-type: none"> AllergyIntolerance [3] Condition [3] Procedure [3] 	<ul style="list-style-type: none"> FamilyMemberHistory [2] DetectedIssue [1] 	<ul style="list-style-type: none"> AdverseEvent [0] ClinicalImpression [0]
	Diagnostics	<ul style="list-style-type: none"> Observation [N] Specimen [2] QuestionnaireResponse [3] 	<ul style="list-style-type: none"> Media [1] BodyStructure [1] MolecularSequence [1] DiagnosticReport [3] ImagingStudy [3] 	
	Medications	<ul style="list-style-type: none"> MedicationRequest [3] MedicationAdministration [2] Medication [3] 	<ul style="list-style-type: none"> MedicationDispense [2] ImmunizationRecommendation [1] MedicationStatement [3] Immunization [3] 	<ul style="list-style-type: none"> MedicationKnowledge [0] ImmunizationEvaluation [0]
	Care Provision		<ul style="list-style-type: none"> CarePlan [2] CareTeam [2] Goal [2] ServiceRequest [2] VisionPrescription [2] RiskAssessment [1] RequestGroup [2] NutritionOrder [2] 	
	Request & Response		<ul style="list-style-type: none"> Communication [2] CommunicationRequest [2] DeviceRequest [1] GuidanceResponse [2] SupplyRequest [1] SupplyDelivery [1] 	<ul style="list-style-type: none"> DeviceUseStatement [0]

3.2 リリース内容

対象リソース

括弧内の数字は成熟度レベル

		第1.0版	第2.0版	第3.0版以降
Financial	Support		<ul style="list-style-type: none">Coverage [2]CoverageEligibilityRequest [2]CoverageEligibilityRespons [2]	<ul style="list-style-type: none">EnrollmentRequest [0]EnrollmentResponse [0]
	Billing		<ul style="list-style-type: none">Claim [2]ClaimResponse [2]	<ul style="list-style-type: none">Invoice [0]
	Payment		<ul style="list-style-type: none">PaymentNotice [2]PaymentReconciliation [2]	
	General		<ul style="list-style-type: none">Account [2]Contract [1]ExplanationOfBenefit [1]	<ul style="list-style-type: none">ChargeItem [0]ChargeItemDefinition [0]InsurancePlan [0]

3.2 リリース内容

対象リソース

		第1.0版	第2.0版	第3.0版以降	括弧内の数字は成熟度レベル
Specialized	Public Health & Research		<ul style="list-style-type: none"> • ResearchStudy [1] • ResearchSubject [1] 		
	Definitional Artifacts	<ul style="list-style-type: none"> • Questionnaire [3] 	<ul style="list-style-type: none"> • ActivityDefinition [2] • PlanDefinition [2] 	<ul style="list-style-type: none"> • DeviceDefinition [0] • EventDefinition [0] • ObservationDefinition [0] • SpecimenDefinition [0] 	
	Evidence-Based Medicine			<ul style="list-style-type: none"> • ResearchDefinition [0] • ResearchElementDefinition [0] • Evidence [0] • EvidenceVariable [0] • EffectEvidenceSynthesis [0] • RiskEvidenceSynthesis [0] 	
	Quality Reporting & Testing		<ul style="list-style-type: none"> • Measure [2] • MeasureReport [2] • TestScript [2] 	<ul style="list-style-type: none"> • TestReport [0] 	
	Medication Definition			<ul style="list-style-type: none"> • MedicinalProduct [0] • MedicinalProductAuthorization [0] • MedicinalProductContraindication [0] • MedicinalProductIndication [0] • MedicinalProductIngredient [0] • MedicinalProductInteraction [0] • MedicinalProductManufactured [0] • MedicinalProductPackaged [0] • MedicinalProductPharmaceutical [0] • MedicinalProductUndesirableEffect [0] • SubstanceNucleicAcid [0] • SubstancePolymer [0] • SubstanceProtein [0] • SubstanceReferenceInformation [0] • SubstanceSpecification [0] • SubstanceSourceMaterial [0] 	

3.2 リリース内容

対象RESTfulインタラクション

第1.0版	第2.0版
<ul style="list-style-type: none">• read• search (タイプレベル)• capabilities• create• update• delete• transaction• batch	<ul style="list-style-type: none">• vread• history (インスタンスレベル)• history (タイプレベル)• search (全システムレベル)• patch• history (全システムレベル)

3.2 リリース内容

対象検索パラメータ

第1.0版	説明
_format	・ 代替応答形式の指定 ※第1フェーズではJSONのみをサポートする
_pretty	・ プリティプリント形式（改行やインデントによって、人が応答内容を確認できるようにレイアウトされた形式）による応答
_elements	・ 返却されるリソースに含まれる要素の指定
_id	・ Resource id による検索
_lastUpdated	・ 最終更新日による検索
_has	・ 検索対象のリソースを参照するリソースのプロパティに基づく検索 (例) GET [base]/Patient?_has:Observation:patient:code=1234-5 検索対象のリソース (Patient) を参照するリソース (Observation) のプロパティ (Observation.subjectがPatientで Observation.codeが1234-5) で検索
_type	・ リソース型による検索 (例) GET [base]/?_type=Observation,Condition&other params... リソース型がObservationとConditionで、他の検索条件が…による検索
_profile	・ 対象プロフィールにタグ付けされた全リソースの検索
第2.0版	説明
_list	・ 指定されたリストに含まれる全リソースの取得
_tag	・ リソースが持つタグによる検索
_security	・ セキュリティラベルによる検索
_content	・ 全リソース内容に対するテキスト検索
_text	・ リソースのナラティブに対するテキスト検索
_query	・ 名前付きクエリ
_filter	・ 複数の検索条件の組み合わせ検索

3.2 リリース内容

対象検索パラメータ

第1.0版	説明
<code>_include</code>	<ul style="list-style-type: none">検索対象のリソースから参照されるリソースを返却 (例) GET [base]/MedicationRequest?_include=MedicationRequest:patient 検索対象のリソース (MedicationRequest) が参照しているリソース (Patient) を返却

第2.0版	説明
<code>_count</code>	<ul style="list-style-type: none">1ページ当たりの検索結果数を指定
<code>_sort</code>	<ul style="list-style-type: none">検索結果のソート順を指定
<code>_reinclude</code>	<ul style="list-style-type: none">検索対象のリソースを参照するリソースを返却 (例) GET [base]/MedicationRequest?_reinclude=Provenance:target 検索対象のリソース (MedicationRequest) を参照しているリソース (Provenance) を返却
<code>_contained</code>	<ul style="list-style-type: none">他のリソースに包含されるリソースを返却するかどうかを制御<ul style="list-style-type: none">– false (default) : 他のリソースに包含されるリソースは返却しない– true : 他のリソースに包含されるリソースのみを返却– both : 他のリソースに包含されるリソースと包含されない (通常の) リソースの両方を返却
<code>_containedType</code>	<ul style="list-style-type: none">他のリソースに包含されるリソースを返却する場合、包含されるリソースのみを返却するのか、コンテナリソースを含み返却するかを制御<ul style="list-style-type: none">– container (default) : コンテナリソースを返却– contained : 他のリソースに包含されるリソースのみを返却
<code>_summary</code>	<ul style="list-style-type: none">サマリ要素のみを返却<ul style="list-style-type: none">– true : リソースの基本定義で「summary」としてマークされている要素のみを返す– text : 「text」要素、「id」要素、「meta」要素、および最上位の必須要素のみを返す– data : テキスト要素を削除して返す– count : 検索のみ。一致したリソースの数のみを返す– false : リソースの全てを返す