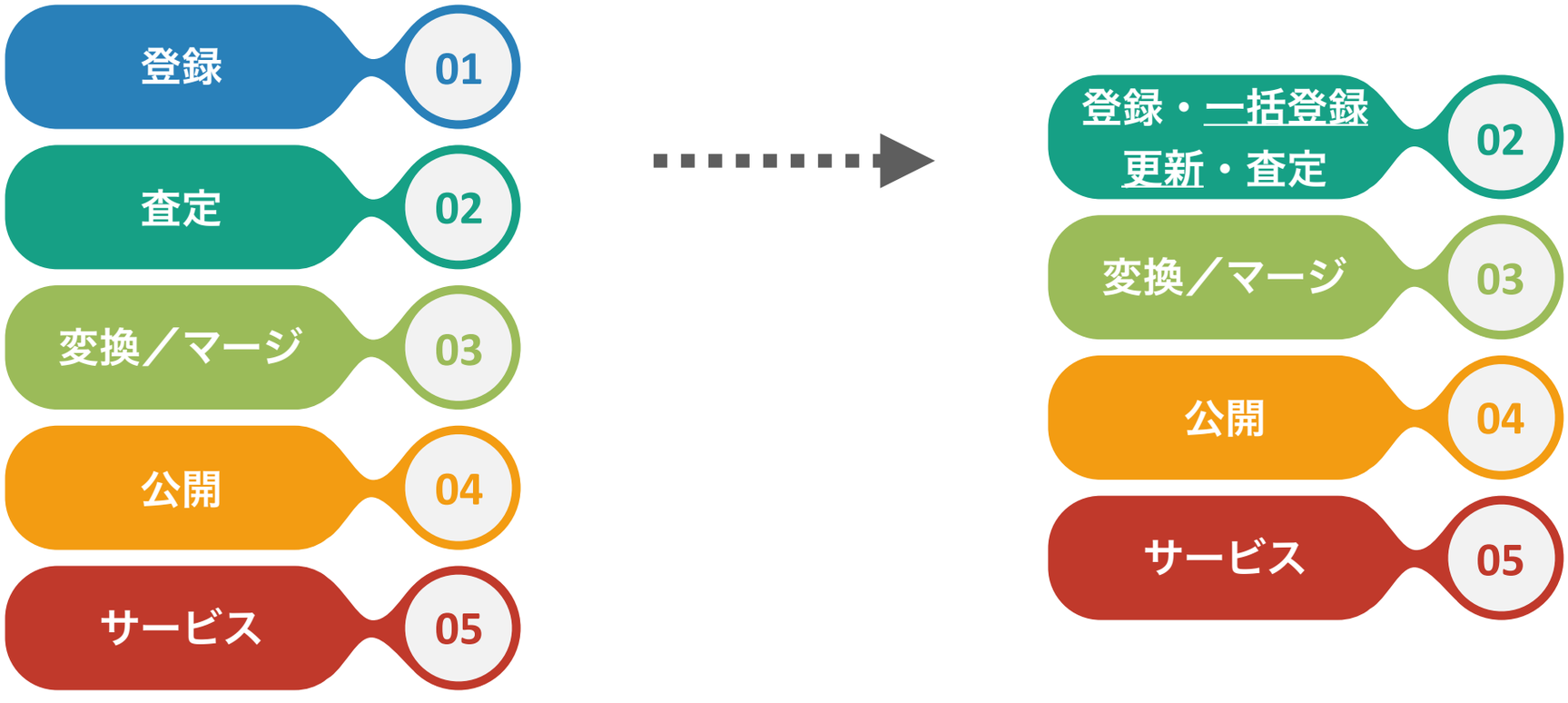
DDBJ: Submission API

2019.04.12 藤澤貴智

開発の目的

- 連携サービスとのインターフェース
  - MetaboBank入力メタデータのBioSample/BioProjectエントリー化（７月）
  - DFASTによるTrad.（ゲノム）登録
- 登録・更新・査定アプリケーションの統合化、自動化



DDBJ登録・更新・査定システム

- **ユーザ**
  - **ログイン**
  - ユーザー情報の入力・編集・管理
  - メタデータ入力（BioProject, BioSample, DRA, Tradメタデータ）
  - 配列リードデータのアップロード
  - 配列データのアップロード
  - 配列アノテーションの入力
- DDBJ
  - **バリデーション（自動、手動）＝API実装により、ほぼ統合（GEA、JGAは未統合）**
  - キュレーターによる承認 **= DDBJ管理ロールで実現可能**
  - ユーザに返却・登録プロセス **= 登録プロセスの仕様要確認**
  - Accessionの発行 **= API化**
  - 履歴管理 **= 仕様要検討**
  - 更新時のデータリストアと差分比較 **= 非公開・公開データ取得API化**

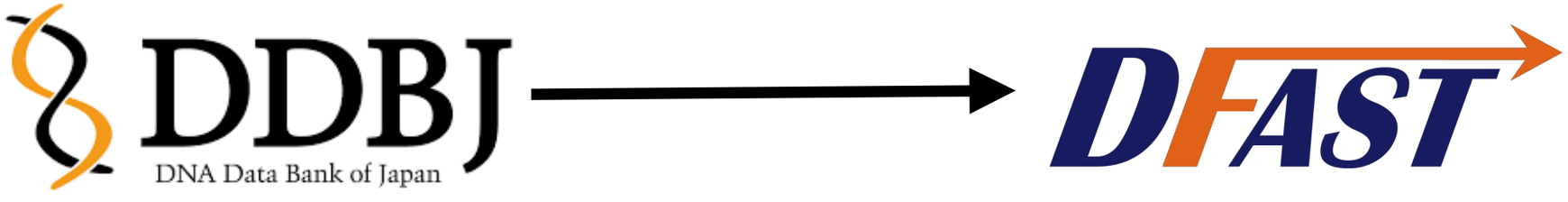
実現するために各コンポーネント化、API化

Submission API開発（案）

- DDBJバリデーター／~~キュレーター~~（自動／~~手動査定~~）→ ユーザ（論理登録）→ DDBJキュレーター（**登録**）
- DDBJバリデーター／~~キュレーター~~（自動／~~手動査定~~）→ユーザ（論理更新）→ DDBJキュレーター（**更新**）
- DDBJバリデーター／~~キュレーター~~（自動／~~手動査定~~）→ユーザ（論理削除）→ DDBJキュレーター（**削除**）
    - 更新、削除時のバリデーションは公開／非公開、DB間の依存関係の確認する必要がある
- Submission APIは、DDBJバリデーターによる自動査定を対象として**Submission ID/Accession IDをダイレクトに発行**するもの
- Validation APIで登録のインタフェースはすでにあるので、Submission APIに拡張、認証機能が足りてない
- 実装コードは現状 ruby/sinatra、開発フェーズ＋ユーザ限定なのでこのまま拡張して、別途APIサーバ/CUI部分のみJAVAによるあとでも可能
- 下流の公開までのフローを短期（〜7月）、中期的（1-2年）の視点で考える必要がある
    - バックエンドに必要なものは、**Submission DB**で、DDBJ**キュレーターによる登録、更新、削除**
    - 運用開始時はユーザ側の極力対応（当面の実ユーザはMetaboBank, DFAST）、必要に応じてキュレーターによるDBを直接編集
    - 当面、MetaboBankのBioSample/BioProjectの受け皿と公開までのフローをどうするか

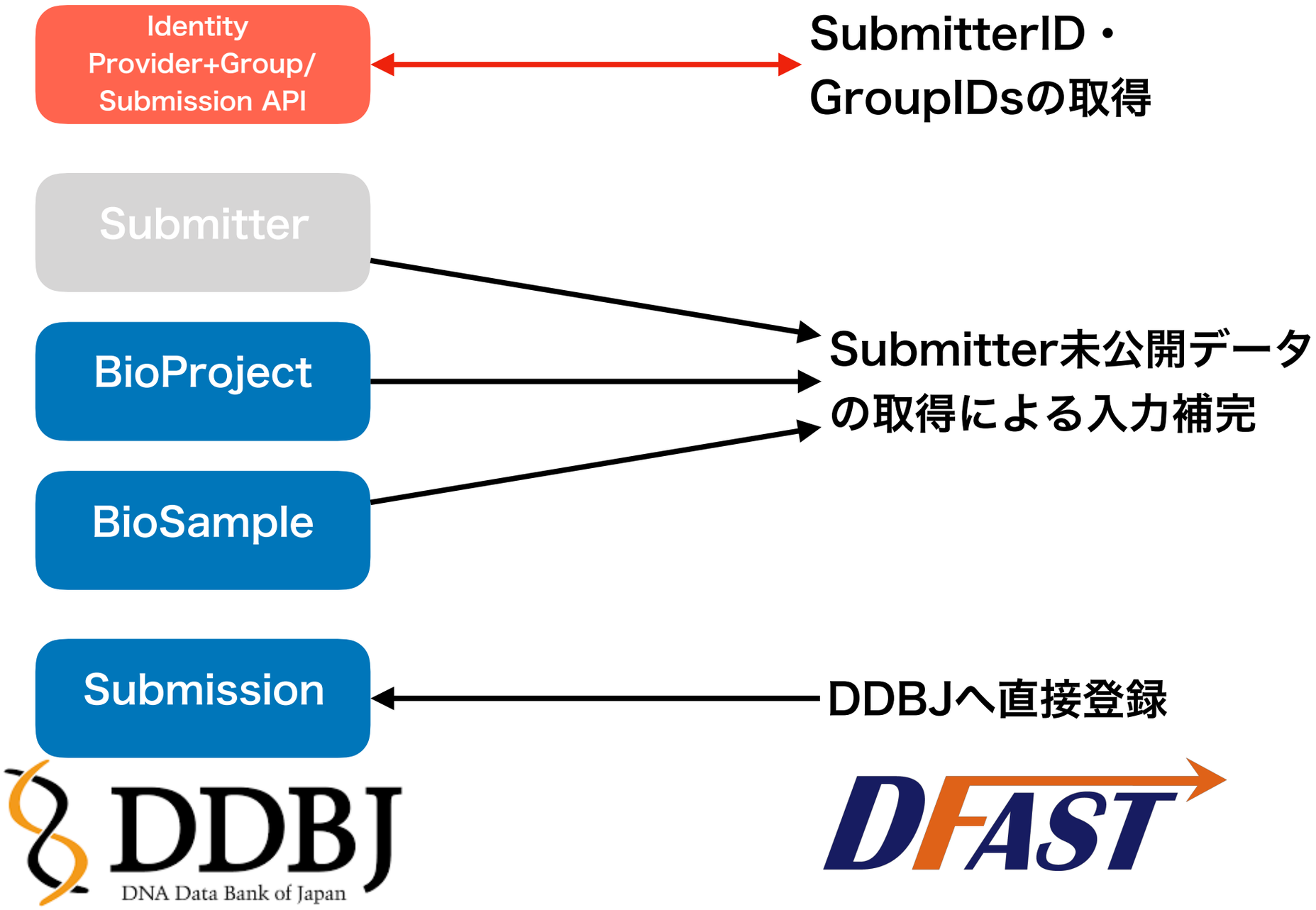
DDBJ-DFAST連携（案１）

- ウェブサービス間（SP→SP）連携
  - SSOが前提
  - D-WayウェブサービスのBioProject、BioSample登録からDFASTウェブサービスへの遷移
  - リンク+ α
  - IDトークンの引継ぎなど



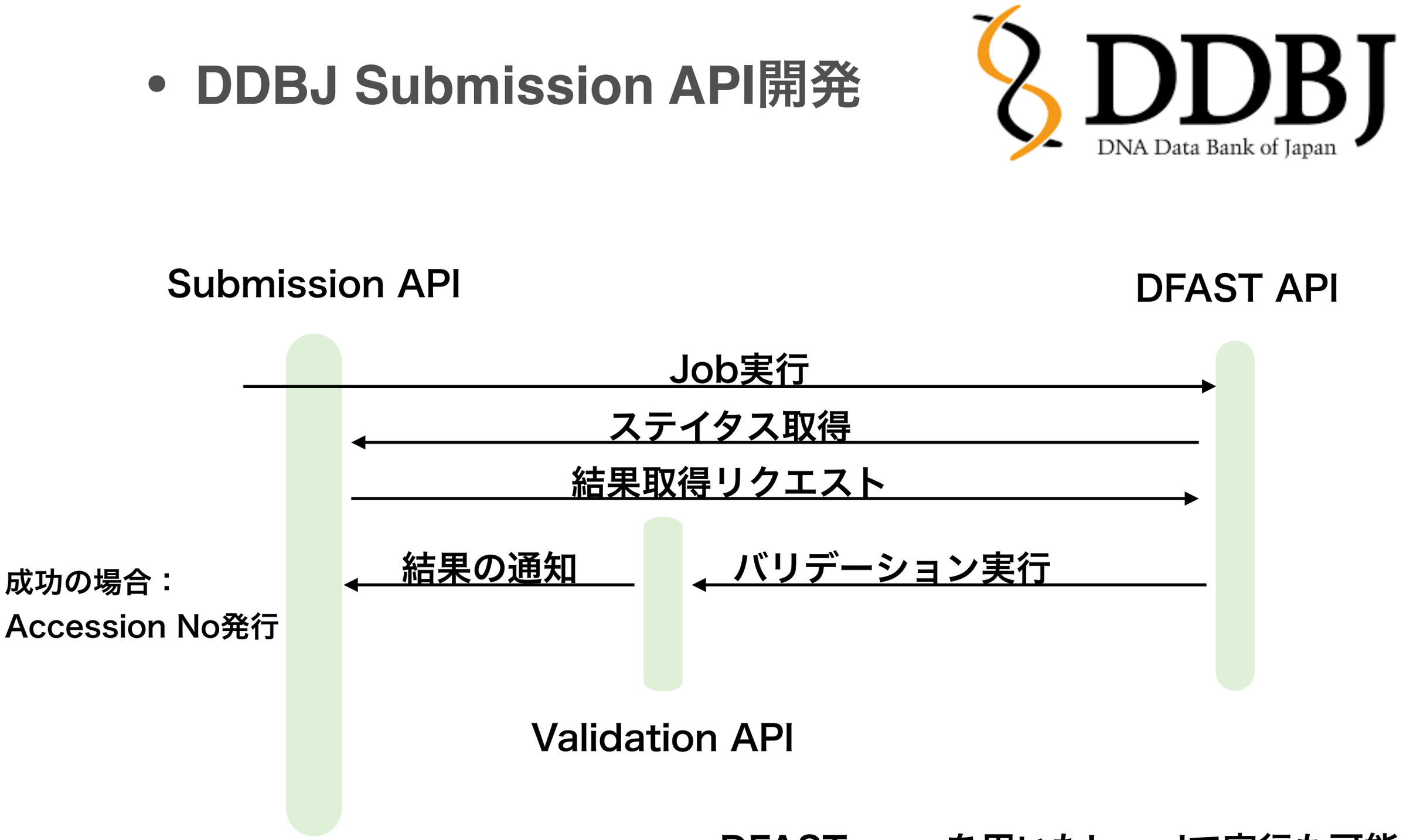
DDBJ-DFAST連携（案２）

- DFASTウェブサービスのDDBJユーザ認証クライアント化



DDBJ-DFAST連携（案３）

- DDBJ Submission API開発



DFAST coreを用いたLocalで実行も可能

DDBJ-DFAST連携（案４）

- **DFAST Submissionコンテナ開発**
  - DFAST実行
  - Trad Validation実行
  - DDBJ SubmissionへPOST
- Submission API（案２）の基盤や太田CWLシステムへの組込みに展開

DDBJシステム開発の方向性

	Trad.	BioProject	BioSample	DRA	GEA	JGA	Assembly
登録・更新・査定	DFAST・MetaboBank連携						
変換／マージ							
公開							
サービス							

中期的に登録・更新・査定をSubmission API開発を起点に統合