**来歴管理システム設計書**

第1.0版

**来歴管理システム設計書**

第1.4版



**2022年12月28日**

**富士通株式会社**

改版履歴

|  |  |  |  |
| --- | --- | --- | --- |
| **版数** | **年月日** | **担当** | **改版内容** |
| 1.0 | 2022.3.31 | 富士通 | 初版 |
| 1.1 | 2022.5.31 | 富士通 | ユーザー管理API　5.4章追記  ユーザー認証の動作について　7章追記 |
| 1.2 | 2022.08.31 | 富士通 | 認証対応に伴い6.3章　認証とAPIシーケンスを追記 |
| 1.3 | 2022.09.19 | 富士通 | 認証機能要件追記２.５章 |
| 1.4 | 2022.12.28 | 富士通 | エディトリアルな修正を実施 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

用語一覧

本資料内で使用する用語を以下に示す。

|  |  |  |
| --- | --- | --- |
| **No** | **用語** | **説明** |
| 1 | 履歴 | データの原本情報やデータ交換における送信履歴、受信履歴といったデータ流通におけるデータプロセスの履歴を指す。 |
| 2 | 来歴 | 履歴間の前後の関係性を管理し、データの原本からのデータプロセスの一連の履歴を指す。 |
| 3 | 来歴管理システム | 履歴の記録と、来歴としての管理、提供を行うシステムである。 |
| 4 | Web App | 来歴管理システムを利用するユーザー定義のアプリケーション |
| 5 | CADDE（ジャッデ） | 分野間データ連携基盤を指す。本基盤を通して、データカタログの作成やデータ発見、データ交換が行われる。  CADDEは、  Connector Architecture for Decentralized Data Exchangeの略語である。  なお、CADDEは情報・システム研究機構（国立情報学研究所）の商標である。 |
| 6 | CKAN | オープンソースのDMS （Data Management System）で、主にオープンデータのデータハブやポータルサイトのために開発、世界的に広く利用している。 |
| 7 | コネクタ | 分散型のデータ交換を実現するためにデータ利用者側とデータ提供者側にそれぞれ実装し通信を行う。 |
| 8 | データ利用者コネクタ | データ利用者システムに設置するコネクタ。データ提供者コネクタと連携しデータ提供者システムからデータを取得する。 |
| 9 | データ提供者コネクタ | データ提供者システムに設置するコネクタ。データ利用者コネクタと連携しデータ提供者システムのデータを提供する。 |

目次

[1. はじめに 6](#_Toc122336147)

[1.1. 本書の位置付け 6](#_Toc122336148)

[1.2. 来歴管理システムの前提 6](#_Toc122336149)

[1.3. CADDEについて 6](#_Toc122336150)

[1.4. 謝辞 6](#_Toc122336151)

[2. 概要要件 7](#_Toc122336152)

[2.1. 来歴管理システムの位置付け 7](#_Toc122336153)

[2.1.1. CADDEにおける機能の位置付け 7](#_Toc122336154)

[2.1.2. CADDEにおける動作としての位置付け 8](#_Toc122336155)

[2.2. 来歴管理システム概要 9](#_Toc122336156)

[2.2.1. データ来歴とは 9](#_Toc122336157)

[2.2.2. データ来歴管理の狙い 10](#_Toc122336158)

[2.2.3. データ来歴のレベルと対応 10](#_Toc122336159)

[2.3. データ来歴における履歴 11](#_Toc122336160)

[2.3.1. データの原本情報 11](#_Toc122336161)

[2.3.2. データ交換における送信履歴 11](#_Toc122336162)

[2.3.3. データ交換における受信履歴 11](#_Toc122336163)

[2.3.4. データの加工履歴 11](#_Toc122336164)

[2.4. データ来歴の動作例 13](#_Toc122336165)

[2.4.1. データ原本情報の登録 13](#_Toc122336166)

[2.4.2. データ交換における送信履歴および受信履歴の登録 14](#_Toc122336167)

[2.4.3. データ加工履歴の登録 15](#_Toc122336168)

[2.4.4. 来歴確認（前方・後方） 16](#_Toc122336169)

[2.4.5. データ来歴における分岐・合流について 17](#_Toc122336170)

[2.5. 来歴管理システムの機能要件 18](#_Toc122336171)

[2.5.1. ユーザー認証 18](#_Toc122336172)

[2.5.2. 履歴登録 18](#_Toc122336173)

[2.5.3. 履歴検索 18](#_Toc122336174)

[2.5.4. 来歴確認 18](#_Toc122336175)

[2.6. ユーザー認証について 18](#_Toc122336176)

[2.6.1. 認証の要件 19](#_Toc122336177)

[2.6.2. ユーザー認証の動作 19](#_Toc122336178)

[2.6.3. アカウントの同期について 20](#_Toc122336179)

[2.6.4. IDトークン取得・送信 20](#_Toc122336180)

[2.6.5. IDトークン受信・抽出 20](#_Toc122336181)

[2.6.6. ユーザーアカウント確認、登録 20](#_Toc122336182)

[2.7. 来歴管理システムイメージ 21](#_Toc122336183)

[3. 外部シーケンス 22](#_Toc122336184)

[3.1. 履歴登録 23](#_Toc122336185)

[3.1.1. データ原本情報登録 23](#_Toc122336186)

[3.1.2. （二次）データ原本情報登録 24](#_Toc122336187)

[3.1.3. データ取得・連携:送信履歴登録 25](#_Toc122336188)

[3.1.4. データ取得・連携:受信履歴登録 26](#_Toc122336189)

[3.1.5. データ取得・連携: データ加工履歴登録 27](#_Toc122336190)

[3.2. 取得 28](#_Toc122336191)

[3.2.1. データ発見後:来歴確認 28](#_Toc122336192)

[3.2.2. データ取得後:来歴確認 29](#_Toc122336193)

[3.2.3. 検索:履歴ID検索 30](#_Toc122336194)

[4. データモデル 31](#_Toc122336195)

[4.1. データモデル図 31](#_Toc122336196)

[4.2. 履歴項目 32](#_Toc122336197)

[4.3. 履歴の登録情報 32](#_Toc122336198)

[4.3.1. データ原本情報 32](#_Toc122336199)

[4.3.2. 送信履歴 32](#_Toc122336200)

[4.3.3. 受信履歴 33](#_Toc122336201)

[4.3.4. 4.3.4 データ加工履歴 33](#_Toc122336202)

[4.4. 履歴間の関係性設定 33](#_Toc122336203)

[4.5. データ来歴の構成 35](#_Toc122336204)

[5. 外部API 36](#_Toc122336205)

[5.1. 履歴登録API 36](#_Toc122336206)

[5.1.1. 名称 36](#_Toc122336207)

[5.1.2. API概要 36](#_Toc122336208)

[5.1.3. API詳細 36](#_Toc122336209)

[5.1.4. 対象シーン 38](#_Toc122336210)

[5.1.4.1. データ原本情報登録 38](#_Toc122336211)

[5.1.4.2. （二次）データ原本情報登録 39](#_Toc122336212)

[5.1.4.3. 送信履歴登録 40](#_Toc122336213)

[5.1.4.4. 受信履歴登録 41](#_Toc122336214)

[5.1.4.5. データ加工履歴登録 41](#_Toc122336215)

[5.2. 来歴確認API 43](#_Toc122336216)

[5.2.1. 名称 43](#_Toc122336217)

[5.2.2. API概要 43](#_Toc122336218)

[5.2.3. API詳細 43](#_Toc122336219)

[5.2.4. API実施例 44](#_Toc122336220)

[5.2.4.1. APIパラメータ 44](#_Toc122336221)

[5.3. 来歴管理I/F:履歴検索API 46](#_Toc122336222)

[5.3.1. 名称 46](#_Toc122336223)

[5.3.2. API概要 46](#_Toc122336224)

[5.3.3. API詳細 46](#_Toc122336225)

[5.4. ユーザー管理API 48](#_Toc122336226)

[5.4.1. API概要 48](#_Toc122336227)

[5.4.2. 新規ユーザー登録 48](#_Toc122336228)

[5.4.3. ユーザー情報削除 49](#_Toc122336229)

[5.4.4. 管理用セッションID取得 50](#_Toc122336230)

[6. 来歴管理システム構成 51](#_Toc122336231)

[6.1. 全体構成 51](#_Toc122336232)

[6.2. 機能内容 52](#_Toc122336233)

[6.3. 機能間シーケンス 53](#_Toc122336234)

[6.3.1. 全API共通認証 53](#_Toc122336235)

[6.3.2. 履歴登録API 54](#_Toc122336236)

[6.3.3. 履歴取得API 56](#_Toc122336237)

[6.3.4. 履歴検索API 58](#_Toc122336238)

[7. 利用OSS一覧 （主要なもの） 60](#_Toc122336239)

# はじめに

## 本書の位置付け

分野間データ連携基盤における来歴管理システムは、履歴記録や来歴取得を担う。  
本書は、来歴管理システムの概要、シーケンス、機能、データモデルおよび外部APIを記載した設計書である。

## 来歴管理システムの前提

来歴管理システムは、分野間データ連携基盤であるConnector Architecture for Decentralized Data Exchange（CADDE）におけるデータの原本情報やデータ交換、データ加工といった履歴を一連に管理し、来歴として提供する機能であり、CADDE内での利用を想定している。

## CADDEについて

CADDE（ジャッデ）は大学共同利用機関法人 情報・システム研究機構 国立情報学研究所の登録商標である。

## 謝辞

本研究（の一部）は、内閣府総合科学技術・イノベーション会議の「SIP/ビッグデータ・AIを活用したサイバー空間基盤技術」（管理法人:国立研究開発法人新エネルギー・産業技術総合開発機構）によって実施されました。

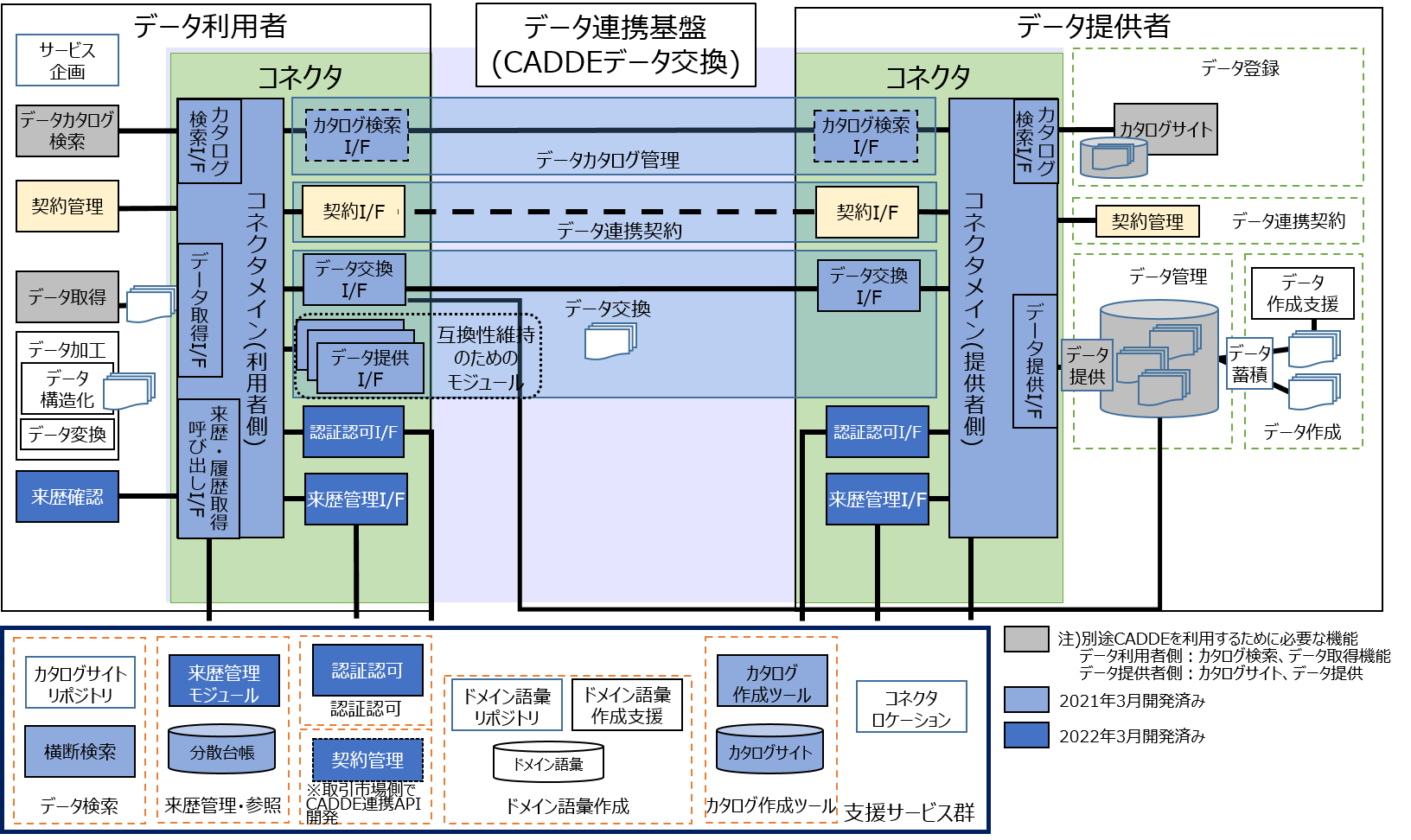
# 概要要件

本章では、来歴管理システムの概要を記載する。

## 来歴管理システムの位置付け

### **CADDEにおける機能の位置付け**

来歴管理システムは、CADDEにおけるデータの原本情報およびデータ交換、データ加工といったデータプロセスの履歴を記録し、CADDEのユーザーからの要求に応じて、関係する一連の履歴とした来歴を提供する。図表2-1-1に、CADDEの構成における来歴管理システムの機能の関係を示す。



図表2-1-1 データ連携基盤の全体像

### **CADDEにおける動作としての位置付け**

来歴管理システムは、CADDEにおけるデータへの操作に対して、履歴を記録する。履歴の記録として、データカタログの登録とともに、対象のデータの原本情報を記録や、データ提供者とデータ利用者間でデータ交換した際に、データの送信履歴および受信履歴を記録する。また、ある組織内において、データの加工を実施した場合に、そのデータ加工履歴も記録する。データ提供者あるいはデータ利用者が、自身に関係する履歴を基点として、来歴を確認した場合、来歴管理システムに対し、履歴の識別子を指定することで、一連の履歴としての来歴を取得することが可能である。

***データ交換***

**データ利用 （サービス企画、サービス提供）**

**データ提供（データカタログ/データの提供）**

CADDE利用準備（提供者IDの取得）

CADDE利用準備（利用者IDの取得）

*データ提供者アロケーション登録*

**データ提供**

提供データの準備

データ提供に係る機能の開発

データカタログの準備

カタログの登録※

**データ提供**

**データ発見**

サービス立案

カタログの検索

データカタログの提供

*カタログデータの交換*

データの発見

**利用契約**

継続的にデータ更新を伴う提供業務

**データ取得・連携**

データ提供

提供データの取得

*提供データの連携*

来歴確認

来歴登録

アプリの開発

サービス提供

図表2-1-2　来歴管理システムの動作-

## 来歴管理システム概要

来歴管理システムは、データ流通におけるデータ交換やデータ加工といったデータプロセスの履歴を、前後に関係する履歴との関係性を含めて管理することで、データ利用者ならびにデータ提供者が、自身に関係する履歴を基点に関連する履歴を一連に確認可能とする。これにより、データ利用者ならびにデータ提供者の双方にとって、データ流通における信頼性を向上させる。

### **データ来歴とは**

今後のデータ流通として、データ利用者は、データ提供者から取得したデータを自身向けに分析・加工して利用するとともに、新しく得られたデータを、今度はデータ提供者として、再流通することが想定される。この様にデータ流通が繰り返されることによって、データの二次・三次流通に発展し、多種多様なデータが流通する。

一方、このように複数の企業・組織をまたいで、二次・三次流通されて行くデータにおいて、データ利用者に対して、大元のデータからのデータ交換・加工といった一連のデータのプロセスを見せることで、データに対する信頼性を向上させることが求められる。また、データ提供者に対しても、自身が提供したデータが、その後にどのように流通されているかを確認できるようにすることで、安心してデータを提供させることが必要である。

データ来歴とは、このように複数の組織・企業をまたいで、データ交換・加工されて行くデータに対し、大元のデータから最新のデータまでの交換や加工履歴を提供する機能である。データ利用者は、取得するデータの視点で、大元の原本のデータからの一連の履歴を確認することが可能になる。一方、データ提供者は、提供するデータの視点で、その先の一連の履歴を確認することが可能になる。データ来歴は、CADDEによって二次・三次流通するデータに対して、データの原本情報と、データ交換・データ加工といったデータプロセスのトレーサビリティを実現するものである。

図2.2-1に組織をまたがるデータ来歴の全体像を示す。

データ

データ

データ

**データ加工**

**原本データ**

**データ**

**利用者**

**原本データ**



**データ**

**提供者**



**データ**

**提供者**

**データ交換**

**データ交換**



**データ  
提供者**



**データ**

**利用者**



**データ**

**利用者**

**データ流通の流れ**

**データ交換**

**自動車会社等**

**自治体等**

**データの交換・加工履歴のトレーサビリティ**

提供データ

原本データ

原本データ

**分析加工**

**データ分析会社**

データの流通先が、自治体であることが確認可能

データの流通先が、自治体であることが確認可能

図表2-2-1　データ来歴の全体像

### **データ来歴管理の狙い**

データ来歴の狙いは、データ流通のデータ利用者やデータ提供者といった参加者が、流通における交換・加工履歴を相互に共有することで、参加者および・データの信頼性を上げるとともに、信頼あるデータ流通を形成することにより、データ流通を活性化することであることを狙いとする。なお、各参加者は、正しい履歴を各々登録し相互に共有することを前提とする。

　データ来歴による効果としては、データ利用者の立場では、データ提供者が提供したデータ来歴（大元のデータの原本性から過程の加工・交換履歴）を確認することで、安心してデータ利用が可能になる。

一方、データ提供者の立場では、提供するデータの来歴を提示することで、データ利用者からの信頼を得て、データの提供を増やすことが可能になる。また、自身のデータの追跡が可能となり、安心してデータ提供が可能になる。

図表 2-2-2データ来歴の狙いに、データ来歴有無の比較を示す。

**従来（データ来歴なし）**

**データ来歴による信頼性向上**

図表2-2-2　データ来歴の狙い

データに不安を感じたときに，大元のデータ等を確認できない

利用者に安心してデータを使ってもらえない

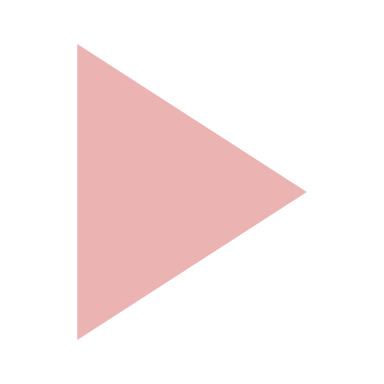
提供するデータを追跡できず不安

問題が生じたときに，大元までの履歴を確認できる

提供するデータを追跡でき，安心して提供できる

利用者に安心してデータを使ってもらえる

信頼を得る





**データ  
提供者**

**データ**

**利用者**



**データ  
提供者**

**データ**

**データ**



**データ  
提供者**

**データ**

**利用者**



**データ  
提供者**

**データ**

**データ**



**データ**

**利用者**

**来歴管理**

**確認できない　×**

**来歴で確認**

**×　確認できない**

**来歴で確認**

### **データ来歴のレベルと対応**

データ来歴のレベルは、図表 2-2-3に示すように2つの段階が想定される。現状、最初のステップとしてレベル1に向けて対応する。

図表 2‑2-3　データ来歴のレベル

|  |  |
| --- | --- |
| **レベル** | **概要** |
| 1 | 来歴（データの原本情報、データ交換履歴、データ加工履歴）を確認できること。  ・ 信頼関係のあるプレイヤ間でのデータ交換を対象とし、各プレイヤが正しい交換、正しいデータ処理を行う前提。 |
| 2 | 来歴を基づいて、原本のデータから加工を行いデータが再現できるかの検証ができること。 |

## データ来歴における履歴

組織をまたがるデータ流通におけるデータ来歴の履歴は、（a）データの原本情報、（b）データ交換における（b）送信履歴、（c）受信履歴、（d）データ加工履歴の４種とする。以降に、各々の履歴について記載する。図表2-3に、データ流通の流れにおいて記録される履歴の対応を示す。

なお、データ来歴では、履歴間の関係性を構築する上で、履歴を登録する際に、前段の履歴の識別子を指定する必要がある。

### **データの原本情報**

（a）データの原本情報は、データの大元を示す情報である。情報としては、データ提供者の情報やURLといったデータの情報、データの原本としてのデータのハッシュ値等になる。データの原本情報の記録は、データ提供者がデータカタログを作成するタイミング等、データを提供する事前に行う想定である。

図表2-3内の①に該当する。

### **データ交換における送信履歴**

（b）データ交換における送信履歴は、組織間でデータ交換する際のデータ提供者からデータを送信した履歴である。履歴としては、提供するデータの原本情報を示す履歴ID、データ提供者の情報や提供データの情報、送信した時刻等になる。

図表2-3内の②に該当する。

### **データ交換における受信履歴**

（c）データ交換における受信履歴は、組織間でデータ交換する際のデータ利用者がデータを受信した履歴である。履歴としては、データ利用者の情報や取得したデータの情報、受信した時刻等になる。

図表2-3内の②に該当する。

### **データの加工履歴**

（d）データの加工履歴は、組織内においてデータ加工した履歴である。履歴としては、加工したデータ利用者（あるいはデータ提供者）の情報や加工の情報、加工した時刻等になる。

図表2-3内の③に該当する。

**自動車会社等**

データ

データ

**原本データ**

**データ加工**



**データ  
提供者**



**データ**

**利用者**

**データ流通の流れ**

**データ交換**

**データ**

**利用者**

**データ交換**

原本データ



提供データ

**原本データ**



**分析加工**

データ



**データ**

**提供者**

**データ交換**



**データ**

**提供者**

原本データ

**自治体等**

**データ分析会社**

**①データ**

**原本情報**

**②データ**

**交換履歴**

**②データ**

**原本情報**

**③データ**

**加工履歴**

**①データ**

**原本情報**

加工データの原本情報

組織内での加工履歴

組織間での送信履歴と

受信履歴

データ来歴の起点

**データ来歴**

図表2-3 データ流通の流れにおいて記録する履歴の対応

## データ来歴の動作例

本章では、データ流通の流れにおける各種履歴の記録と、その後のデータ来歴の確認の動作例を示す。

例として、図表2-4-1に示すように、自動車会社等のデータ提供者からデータ分析会社にデータが提供され、データ分析会社は、取得したデータを分析・加工後に、新たなデータとして自治体等のデータ利用者にデータを提供する場合における履歴の記録と、データ来歴の確認の動作例を示す。

### **データ原本情報の登録**

データ提供者は、データカタログを作成するタイミング等、データを提供する事前にデータの原本情報を登録する。原本情報としては、データ提供者の情報やURLといったデータの情報、 データの原本としてのデータのハッシュ値等になる。

図表2-4-1内の①は、「自動車会社等」のデータ提供者がデータA、別のデータ提供者がデータBの原本情報を登録することを示す。

**来歴管理**

データ

データ

**原本データ**

**データ加工**



**データ  
提供者**



**データ**

**利用者**

**データ流通の流れ**

**データ交換A**

**データ**

**利用者**

原本  
データ**A**



提供  
データ**C**

**原本データ**



**分析加工**

データ

**データ交換B**



**データ**

**提供者**

**データ分析会社**



原本  
データ**Ｂ**

**自動車会社等**

**自治体等**

①原本情報登録する

②来歴データを検索確認し取得

登録データ

原本情報 **A**

来歴データ

**Ａ** 取得

来歴データ

**B** 取得

登録データ

原本情報 **B**

③データ交換履歴を各々付与する

**データ**

**提供者**

図表2-4-1 データ原本情報の登録

### **データ交換における送信履歴および受信履歴の登録**

データ提供者とデータ利用者の間で、データ交換が行われる場合、データ提供者から送信履歴、データ利用者から受信履歴を登録する。送信履歴としては、データ提供者の情報や送信データの情報、送信した時刻等、受信履歴としては、データ利用者の情報や受信データの情報、受信した時刻等になる。登録の際には、過去の履歴を指し示すポインタ情報も入力する。ここでは、送信履歴を登録する場合には、データの原本情報を指し示すポインタ情報を入力し、受信履歴を登録する場合には、送信履歴を指し示すポインタ情報を入力する。

図表2-4-2内の②③.は、「自動車会社等」のデータ提供者がデータA、別のデータ提供者がデータBの原本情報を「データ分析会社」がデータ交換履歴を各々付与することを示す。

データ

データ

**原本データ**

**データ加工**



**データ  
提供者**



**データ**

**利用者**

**データ流通の流れ**

**データ**

**利用者**

提供  
データ**C**

原本  
データ**A**

**原本データ**



**分析加工**

データ

原本  
データ**B**



**自動車会社等**

**自治体等**

**データ分析会社**



**データ**

**提供者**

**データ**

**提供者**

**来歴管理**

登録データ

原本情報 **A**

データ

加工履歴

登録データ

原本情報 **B**

データ

交換履歴

データ

交換履歴

③データ交換履歴を各々付与する

②来歴データを検索確認し取得

図表2-4-2 データ交換履歴の登録

### **データ加工履歴の登録**

データ提供者が、自身のデータあるいはデータ利用者から取得したデータを加工し、その後、加工したデータを公開する場合、データ加工の履歴を登録する。データ加工の履歴としては、加工したデータ利用者（あるいはデータ提供者）の情報や加工の情報、加工した時刻等になる。登録する際には、過去の履歴を示すポインタ情報も入力する。自身のデータに対する加工であれば、過去の履歴としては、加工対象のデータの原本情報になる。一方、データ利用者から取得したデータであれば、取得したときの受信履歴が過去の履歴になる。

図表2-4-3内の④は、「データ分析会社」がデータ加工し新たに原本情報Cとして登録した動作を示す。

データ

データ

**原本データ**

**データ加工**



**データ  
提供者**



**データ**

**利用者**

**データ流通の流れ**

**データ**

**利用者**

提供  
データ**C**

原本  
データ**A**

**原本データ**



**分析加工**

データ

原本  
データ**B**



**自動車会社等**

**自治体等**

**データ分析会社**



**データ**

**提供者**

**データ**

**提供者**

**来歴管理**

④加工データの原本情報登録

登録データ

原本情報 **A**

データ

加工履歴

登録データ

原本情報 **B**

登録データ

原本情報 **Ｃ**

データ

交換履歴

データ

交換履歴

図表2-4-3 加工データの原本情報登録

-1000

### **来歴確認（前方・後方）**

登録された履歴において、データ利用者およびデータ提供者は、自身に関係する履歴を基点に関連する一連の履歴としての来歴を確認することが可能である。データ利用者は、取得したデータにおける来歴を確認する場合、取得したデータにおける受信履歴から、それまでの後方方向の送信履歴やデータ原本情報、データ加工履歴を確認することができる。また、取得しようとするデータの場合は、そのデータの原本情報から前の履歴を確認できる。一方、データ提供者は、提供したデータにおける流通先を確認する場合、提供したデータの原本情報から先に繋がる前方方向の履歴を確認することが可能である。

図表2-4-4内の⑤では、自治体等が登録データ原本情報C新たに生成し加工した加工履歴の付与とデータの原本情報登録の動作例を示し、⑥では、自動車会社等のデータAを起点として来歴情報を追跡確認することができる。

データ

データ

**原本データ**

**データ加工**



**データ  
提供者**



**データ**

**利用者**

**データ流通の流れ**

**データ**

**利用者**

提供  
データ**C**

原本  
データ**A**

**原本データ**



**分析加工**

データ

原本  
データ**B**



**自動車会社等**

**自治体等**

**データ分析会社**



**データ**

**提供者**

**データ**

**提供者**

**来歴管理**

登録データ

原本情報 **A**

データ

加工履歴

登録データ

原本情報 **C**

登録データ

原本情報 **B**

⑤来歴取得データC　原本起点

データ

交換履歴

データ

交換履歴

⑥自動車会社が提供するデータＡ追跡

図表2-4-4 来歴情報の追跡確認

### **データ来歴における分岐・合流について**

データ来歴においては、ある履歴から複数の履歴に分岐する場合や、複数の履歴が1つの履歴に合流する場合が想定される。例えば、分岐する場合としては、あるデータに対して、複数のデータ利用者に対してデータを送信することが考えられる。データ来歴としては、データの原本情報に対して、複数の送信履歴を関係付ける必要がある。また、合流する場合としては、複数のデータを加工して1つのデータにまとめることが考えられる。この場合、データ来歴としては、複数のデータの受信履歴に対して、1つのデータ加工履歴を関係付ける必要がある。来歴管理システムでは、これらの分岐・合流のシーンも想定した履歴管理を行う。

図表2-5の①分岐のケースでは新規の原本情報が、データ交換の2件へ履歴取得するケースと、加工履歴データが更に原本情報として2件に登録情報を持つことを示す。

②の合流のケースは、「データ来歴の動作例、取得データの分析加工、登録」で説明した通り自動車会社等の各々原本情報をデータ分析会社がデータ加工した例を示す。

図表2-5 データ来歴における分岐・合流

**データ流通の流れ**

データ

交換履歴

**①分岐のケース**

データ

交換履歴

データ

原本情報

データ

原本情報

データ

原本情報

データ

加工履歴

**②合流のケース**

データ

交換履歴

データ

交換履歴

データ

原本情報

データ

原本情報

データ

加工履歴

## 来歴管理システムの機能要件

本章では、前述のデータ来歴における履歴および動作例を踏まえ、来歴管理システムとして必要な機能として（a）ユーザー認証、（b）履歴登録、（c）履歴検索、（d）来歴確認、の要件を示す。

以下に、各々の機能の要件を示す。

### **ユーザー認証**

本システムは、(a)ユーザー認証機能を有する。このユーザー認証機能は、不特定多数のユーザーによるアクセスからIDプロバイダー(IdP)によって正しく認証されたユーザーのみにアクセスを許可するものであり、これによってセキュリティの向上が見込まれる。次章の2.6.ユーザー認証について詳細を記載する。

### **履歴登録**

（b）データ来歴を構成するデータの原本情報やデータ交換の履歴、データ加工の履歴を登録する機能を要し、履歴間はN対Mの関係性を管理可能である。

前述の履歴を踏まえ、履歴登録の種類としては図表2-5-2になる。

図表2-5-2 履歴の種類

|  |  |
| --- | --- |
| **履歴種類** | **概要** |
| データ原本情報 | 提供対象のデータの原本情報（リソースURL、 ハッシュ値等）を登録する。 |
| データ送信履歴 | データ提供者がデータを送信した際に送信の履歴を登録する。 |
| データ受信履歴 | データ利用者がデータを受信した際に受信の履歴を登録する。 |
| データ加工履歴 | データ提供者がデータを加工した際に加工の履歴を登録する。 |

### **履歴検索**

（c）データ利用者およびデータ提供者が、自身に関係する履歴を検索できる機能が必要である。

履歴検索の場合は、検索キーの指定により、来歴管理システムに登録済みの履歴を検索できる。

### **来歴確認**

（d）データ利用者およびデータ提供者が、参照する履歴から過去の履歴を確認できる（例:取得したデータの加工履歴を確認したい）機能を要し、あるいはそれに続く履歴（例:提供したデータのその後の交換履歴を確認したい）を確認できる機能が必要である。

特定の履歴を指定し、それを起点とした一連の履歴（前方あるいは後方）を取得できる。

## ユーザー認証について

従来の来歴管理システムでは、認証処理を持たないため、実際の来歴の内容として書き込まれる原本の作成者を区別するための情報は、ブロックチェーンのデータとして記録していた。

実用上は、この仕様で十分であったが、これに加えてブロックチェーンのデータを書き込んだ組織を区別する情報として、従来は、事前に複数のユーザーを用意しておき、組織毎にAPIのURLを分け、APIとユーザー名の対応をとることにより、どの組織がブロックチェーンにデータを書き込んだかどうかを区別する仕組みを使用していた。図表2-6aに従来の来歴管理システムの動作を示す。

今回の版では、ブロックチェーンの操作を行うユーザーはAPI発行前に外部のIDプロバイター(外部IdP(Identity Provider))からトークンを取得し、そのトークンを使ってAPI発行を行う。来歴管理は、受け取ったトークンを外部IdPに問い合わせてトークンの正当性を確認することによって認証を行う。この変更によりブロックチェーンの操作を行うアカウントが認証に使用したトークンと連動することになり、共通のAPIのURLで複数のユーザーの管理が可能となる。図表2-6bにユーザー認証導入後の来歴管理システムの動作を示す。



ユーザーA

A



ユーザーB

A



ユーザーC

A

外部IdP

来歴管理

API1 (組織A)



ユーザーC

A



ユーザーB

A



ユーザーA

A

トークン

トークン

トークン

API2 (OrgB)

API (共通)

来歴管理

ユーザー

ユーザー OrgB

ユーザー OrgA

操作ユーザー名

従来は、認証がないため事前に複数のユーザーを用意しておき、組織毎にAPIのURLを分けることで、ブロックチェーンの操作ユーザーを切り替えていた。

API発行者がユーザーを識別可能なトークンを付与する。

ブロックチェーンの操作ユーザーはトークンの内容から決定する。

図表2-6b 今回の版の動作(認証あり)

図表2-6a従来の動作

### 認証の要件

今回の来歴管理サーバー単体では認証処理を有してないため、認証局とよばれる外部にIdPを用意し、IdPの持つ認証処理と連携することで認証機能を実現する。

認証に必要となるIdP及びトークンの要件は以下の通りとなる。

・ユーザーの正当性を示すトークンを発行する認証局(外部IdP)が存在すること。

・トークンにはユーザーの正当性及びユーザーを区別するためのユーザー名の情報が含まれていること。

・外部IdPは自分が発行したトークンの正当性を確認する手段を持つこと。

・トークンの内容から、ユーザー名を抽出できること。または、外部IdPがトークンからユーザー名を取得する手段を内部に持つこと。

・来歴管理側は、外部IdPと同一の名称でユーザー名を登録できること。

上記IdPを使うことにより本システムでは認証処理を実施する。

### ユーザー認証の動作

来歴管理では外部IdPを用いてユーザー認証を実施する。

ユーザー認証は以下の手順で実施する。この手順の中でエラーが発生した場合は、認証失敗とする。

①アプリケーションが外部IdPで発行したIDトークンを取得する。  
②アプリケーションは取得したIDトークンを付与してフロントサーバーに対してAPIを発行する。

③フロントサーバーは、付与されていたIDトークンを外部IdPで検証し、検証結果とトークン内のペイロード部に格納されているユーザーIDをデコードした結果を取得する。

④フロントサーバーは、FabricCA【認証局】にてユーザーアカウントの存在確認をし、もしユーザーアカウントが存在しなかった場合は、新規にユーザーアカウントを登録する。

⑤フロントサーバーは、APIを実行し結果をアブリケーションに通知する。

図表2-6-2に認証処理の動作を示す。詳細は後述する図表6-3-1全API共通認証を参照のこと。

フロントサーバー

**来歴管理API**

**来歴構成管理**

アプリケーション

Fabric CA

[ 認証局 ]

**外部IdP**

**ユーザー管理**



**ユーザー管理**

アカウントAさん

（組織アカウント）

**ロール管理**

ユーザーアカウント=A

図表2-6-2. 認証処理の動作

### アカウントの同期について

前述のとおり認証は外部IdPを用いて実施するが、そこで使用したユーザーアカウントを用いてFabric CA（ブロックチェーン）の操作を実施する必要がある。

そのため、外部IdPのユーザーアカウント情報とFabric CAのユーザーアカウント情報とを同期する必要がある。同期の方法は以下の通りである。

①事前にアカウント情報を手動で登録する。

②API実行時に認証で使用されたユーザーアカウントがFabric CA内に存在するかどうかを確認し、無ければその時点で新規作成する。

本仕様書では①②の両者のAPIを記載するが、実用上はどちらか片方だけでも十分である。

### IDトークン取得・送信

アプリケーションは、外部IdP（ユーザーアカウント管理）からIDトークンを取得する。取得の際、ユーザーID及びパスワードを指定し外部IdP上で認証を実施する。

フロントサーバーにAPIを発行する際は、上記で取得したIDトークンを必ず付与する必要がある。

### IDトークン受信・抽出

フロントサーバーは、受信したIDトークンを再度、IdPに対して送付する。IdPは、トークンの検証を実施し、トークン内部を解析し、ペイロード部分に格納されているユーザー名の情報を取得しフロントサーバーに送付する。

### ユーザーアカウント確認、登録

本システムではユーザーのアカウントをブロックチェーン処理システムのFabric CA【認証局】を用いて管理している。フロントサーバーは、送付されたユーザーアカウントが存在するかどうかをFabric CAに確認し、ユーザーアカウントが存在しなかった場合は、Fabric CAに新規ユーザーアカウント登録を実施する。

## 来歴管理システムイメージ

来歴管理システムは、CADDE関連モジュール/ツールに対してAPIを提供する。

図2-7に来歴管理システムのイメージおよびCADDE関連モジュール/ツールの想定を示す。来歴管理システムは、これらのモジュールおよびツールから来歴管理システムからAPI呼出しが行われ、履歴の登録および来歴の提供を行う。



図表2-7 来歴管理システムのイメージ

以下に、来歴管理システムへのデータの原本情報の登録や他の履歴の登録におけるモジュール、ツールの関係性の例を示す。

* 原本情報の登録:カタログ作成ツールにおいてカタログを公開するタイミングで、原本情報を登録する。
* 送信履歴の登録:データ提供者コネクタがデータを送信する際に送信履歴を登録する。
* 受信履歴の登録:データ利用者コネクタがデータを受信する際に受信履歴を登録する。
* 加工履歴の登録:データ加工履歴登録ツールが、加工履歴を登録する。

　また、来歴管理システムは、外部に対して、履歴検索および来歴確認を提供する。外部のツールやアプリケーションに対して、これらのAPIを提供する想定である。

# 外部シーケンス

歴管理システムに関係する外部システムシーケンスおよび関連する来歴管理の動作一覧を図表3に示す。

図表3 来歴管理の動作一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **種別** | **外部シーケンス** | | **来歴管理動作** | **動作内容** |
| 登録 | 3.1.1 | データ登録 | データ原本情報登録 | データ提供者から原本のデータを受け取ったデータのハッシュ値を計算して履歴IDを含む履歴情報を生成・登録し、カタログに付与する履歴情報の履歴IDを返す。 |
| 3.1.2 | （二次）データ登録 | 二次データ原本情報登録 | 二次データ提供者からデータと連結先の履歴IDを受け取った、データのハッシュ値を計算して履歴情報を生成・登録し、カタログに付与する履歴情報の履歴IDを返す。 |
| 3.1.3 | データ交換 | 送信履歴登録 | データ提供者側からデータ送信における履歴登録の要求を受け、送信履歴を登録する。 |
| 3.1.4 | 受信履歴登録 | データ利用者側からデータ受信における履歴登録の要求を受け、受信履歴を登録する。 |
| 3.1.5 | データ加工 | データ加工履歴登録 | データ利用者あるいはデータ提供者においてデータ加工に伴うデータ加工履歴を登録する。 |
| 取得 | 3.2.1 | データ発見後の来歴確認 | 来歴確認 | データ利用者からカタログ記載の履歴情報の履歴IDを受け取った、来歴情報を返す。 |
| 3.2.2 | データ取得後の来歴確認 | データ取得後にデータ利用者から履歴情報の履歴IDを受け取った、来歴情報を返す。 |
| 3.2.3 | 検索 | 履歴ID検索 | データ提供者から検索キーと値を受け取った、対応する履歴IDのリストを返す。 |

## 履歴登録

来歴管理システムへの履歴登録に関するシーケンスを示す。

### **データ原本情報登録**

来歴管理動作でのデータ登録における新規履歴登録を図表3-1-1のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

データ

カタログ

作成支援

ツール

詳細検索用

カタログ

サイト

(CKAN)

横断検索用

カタログ

サイト

(CKAN)

来歴管理

システム

1.情報入力

後略

**データ提供者環境**

**データ提供者**

**（ブラウザ）**

**データ提供者コネクタ**



2.新規来歴登録(データ、他）

3.OK　[履歴IDx]

4.カタログ情報登録(履歴IDx)

5.登録完了

図表3-1-1 データ原本情報登録

### **（二次）データ原本情報登録**

来歴管理動作での二次データ登録における公開履歴登録を図表3-1-2のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

データ

カタログ

作成支援

ツール

詳細検索用

カタログ

サイト

(CKAN)

横断検索用

カタログ

サイト

(CKAN)

来歴管理

システム

1.情報入力

後略

**データ提供者環境**

**データ提供者**

**（ブラウザ)**

**データ提供者コネクタ**



4.公開履歴登録(データ、履歴ＩＤｚ、他）

シーケンスは3.2.3章と同じ

（複数回行う場合あり）

2.履歴ID検索(検索キー、値）

3.OK　[履歴IDリスト]

6.カタログ情報登録(履歴IDx)

7.登録完了

5.OK　[履歴IDｗ]

図表3-1-2 データ原本情報登録

### **データ取得・連携:送信履歴登録**

来歴管理動作でのデータ取得におけるデータ提供者側の送信履歴登録を図表3-1-3のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

コネクタメイン

+データ交換

制御

カタログ

サイト

(CKAN)

データ

管理

データ提供

I/F

データ取得要求(リソースURL)

データ

データ

データ取得要求(リソースURL)

29.送信履歴登録  
(履歴IDx,他)

履歴ID取得

OK [履歴IDx]

30.OK[履歴IDy]

31.データ利用者のコネクタへ(データ、履歴IDy)

**データ提供者環境**

（前略）

（図表3-1-4へ続く）

**データ提供者コネクタ**

来歴管理

システム

図表3-1-3 送信履歴登録

### **データ取得・連携:受信履歴登録**

来歴管理動作でのデータ取得におけるデータ利用者側の受信履歴登録を図表3-1-4のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

コネクタメイン

（データ交換

制御）

来歴管理

システム

**データ利用者**

**（ブラウザ）**

**データ利用者コネクタ**



データ提供者コネクタから（データ、履歴IDy）

34.受信履歴登録  
(履歴IDy,他)

35.OK　[履歴ＩＤｚ]

利用者側

Ｗｅｂ Ａｐｐ

**データ利用者環境**

（図表3-1-3から）

（前略）

36.データ[履歴ＩＤｚ]

図表3-1-4 受信履歴登録

### **データ取得・連携: データ加工履歴登録**

来歴管理動作でのデータ取得におけるデータ提供者側のデータ加工履歴登録を図表3-1-5のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

データ

加工ツール

来歴管理

システム

1.データ加工

後略

**データ提供者環境**

**データ提供者**

**（ブラウザ）**

**データ提供者コネクタ**



2.データ加工履歴登録(加工後のデータ）

3.OK　[履歴IDx]

4.データ詳細加工履歴登録(詳細加工履歴）

5.OK　[履歴IDx]

図表3-1-5 データ加工履歴登録

## 取得

### **データ発見後:来歴確認**

来歴管理動作でのデータ利用者のデータ発見における来歴情報確認を図表3-2-1のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

コネクタメイン

来歴管理

システム

**データ利用者**

**（ブラウザ）**

**データ利用者コネクタ**



カタログ検索結果

3.来歴確認（来歴IDx）

4.OK　[来歴情報]

（JSON）

利用者側

Ｗｅｂ Ａｐｐ

**データ利用者環境**

（前略）

5.来歴情報

2.歴確認（来歴IDx）

6.来歴情報

1.来歴確認（来歴ＩＤx）

図表3-2-1 来歴確認

### **データ取得後:来歴確認**

来歴管理動作でのデータ利用者におけるデータ取得後の履歴ID指定による来歴確認を図表3-2-2のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

コネクタメイン

来歴管理

システム

**データ利用者**

**（ブラウザ）**

**データ利用者コネクタ**



データ

3.来歴確認（来歴IDx）

4.OK　[来歴情報]

（JSON）

利用者側

Ｗｅｂ Ａｐｐ

**データ利用者環境**

（前略）

5.来歴情報

2.来歴確認（来歴IDz）

6.来歴情報

1.来歴確認

データ、履歴IDz

図表3-2-2来歴確認

### **検索:履歴ID検索**

来歴管理動作でのデータ提供者からの検索により履歴IDリストの取得を図表3-2-3のシーケンスに示す。

この図では、来歴管理I/F呼出しのシーケンスおよび関連するシーケンスのみ示している。

入出力パラメータは図の朱枠内の矢印部分に記載している。

履歴ID検索はデータ利用者が行ってもよい。その場合データ利用者環境やデータ利用者コネクタも下記同ように対応すること。

コネクタメイン

（データ交換

制御）

来歴管理

システム

**データ提供者コネクタ**

3.履歴ID検索（検索キー等）

4.OK　[履歴IDリスト]

5.履歴IDリスト

2.履歴ID検索[検索キー等]

**データ提供者**

**（ブラウザ）**



利用者側

Ｗｅｂ Ａｐｐ

**データ提供者環境**

6.履歴IDリスト

1.履歴ID検索(検索キー等)

図表3-2-3 履歴ID検索

# データモデル

本章では、来歴管理システムで管理する履歴のデータモデルについて記載する。

## データモデル図

図表4-1に、データモデル図を示す。履歴は、識別子等の履歴基本情報およびデータ来歴を構成する上での追跡用情報、各履歴によって異なる登録情報から構成される。追跡用情報としては、前段および後段に対して複数の履歴識別子を設定可能である。

**履歴生成の流れ**

（未来）

（過去）

**[履歴基本情報]**

n

履歴識別子

n

履歴種別

（原本情報・送信履歴など）

n

登録日時

n

管理事業者情報

**[追跡用情報]**

n

前段の履歴識別子

n

後段の履歴識別子

**[登録情報]**

(※

履歴種別によって異なる

)

**履歴**

**前段履歴**

**後段履歴**

後段履歴へのポインタ

前段履歴へのポインタ

N

個のポインタ設定も可

N

個のポインタ設定も可

（前段履歴からのポインタ）

（後段履歴からのポインタ）

図表4-1 データモデル図

## 履歴項目

図表4-2に履歴における各項目の内容を示す。履歴の項目には、システム外から入力する項目と、システム内部で生成して設定する項目がある。

図表 4‑2　履歴項目

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 履歴基本情報 | 履歴識別子 | 外部入力、 内部生成 | 個々の履歴を識別するための識別子。  外部入力ではない場合、内部で生成も可能。 |
| 履歴種別 | 外部入力 | 履歴の種別。種別の情報は以下。   * データ原本情報 * データ送信履歴 * データ受信履歴 * データ加工履歴 |
| 登録日時 | 内部生成 | 履歴が登録された日時 |
| 管理事業者情報 | 内部生成 | 履歴を管理する管理事業者情報（URL, 事業者名）。  （※） 運用時に詳細に定義 |
| 追跡用情報 | 前段の履歴識別子 | 外部入力 | 履歴を過去方向に辿るための前段の履歴の履歴識別子。 |
| 後段の履歴識別子 | 内部生成 | 履歴を未来方向に辿るための後段の履歴の履歴識別子。 |

## 履歴の登録情報

各履歴の登録情報を示す。

### **データ原本情報**

図表4-3-1に来歴管理システムで管理するデータ原本情報を示す。

図表 ‑3-1 データ原本情報

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 登録情報 | 提供者ID （CADDE ID） | 外部入力 | 提供者を示すCADDEユーザーID。 |
| リソースURL | 外部入力 | 原本対象データのリソースURL、コネクタで交換するデータ単位。 |
| 原本情報 | 内部生成 | 外部入力したデータのハッシュを計算して管理。 |

### **送信履歴**

図表4-3-2に来歴管理システムで管理する送信履歴を示す。

図表 4‑3-2 送信履歴

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 登録情報 | 提供者ID （CADDE ID） | 外部入力 | 提供者を示すCADDEユーザーID。 |
| 利用者ID （CADDE ID） | 外部入力 | 利用者を示すCADDEユーザーID。 |
| リソースURL | 外部入力 | 送信対象データのリソースURL、コネクタで送信するデータ単位。 |

### **受信履歴**

図表4.3-3に来歴管理システムで管理する受信履歴を示す。

図表 ‑3-3 受信履歴

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 登録情報 | 提供者ID （CADDE ID） | 外部入力 | 提供者を示すCADDEユーザーID。 |
| 利用者ID （CADDE ID） | 外部入力 | 利用者を示すCADDEユーザーID |
| リソースURL | 外部入力 | 受信対象データのリソースURL、コネクタで送信するデータ単位 |

### **4.3.4 データ加工履歴**

図表4-3-4-1および表4-3-4-2来歴管理システムで管理するデータ加工履歴を示す。

図表 4-3-4-1 データ加工履歴

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 登録情報 | 提供者ID （CADDE ID） | 外部入力 | CADDEユーザーID |
| リソースURL | 外部入力 | 加工後データのリソースURL |
| 原本情報 | 内部生成 | 外部入力した加工後データのハッシュを計算して管理 |

図表 4-3-4-2 データ詳細加工履歴

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **外部入力 内部生成** | **内容・備考** |
| 登録情報 | 提供者ID （CADDE ID） | 外部入力 | CADDEユーザーID |
| リソースURL | 外部入力 | 加工履歴情報のリソースURL |
| 加工情報 | 内部生成 | 外部入力した加工履歴情報（JSON形式）のハッシュを計算して管理 |

## 履歴間の関係性設定

来歴管理システムでデータ来歴として管理するために、履歴を登録する際に、前段の履歴の履歴識別子を指定して登録することが必要である。前段の履歴が組織内で生成・登録されている場合、来歴管理システムのアプリあるいはツールは、来歴管理システムが提供する検索を活用し、履歴識別子を取得する。一方、前段の履歴が組織間で生成・登録されるデータ交換の場合、提供者コネクタから利用者コネクタへのデータ提供とともに履歴識別子を受け渡すことが必要である。

図表4-4に履歴のパターンに対する履歴間の関係性設定について示す。

図表 4‑4 履歴間の関係性設定

|  |  |  |  |
| --- | --- | --- | --- |
| **分類** | **項目** | **履歴の関係性のパターン** | **前段履歴の履歴識別子の付与** |
| 組織内での履歴登録 |  | 「原本情報」から「送信履歴」 | * 「送信履歴」を登録する提供者コネクタが来歴管理システムから履歴識別子を検索。 * 履歴種別/リソースURL（ユニーク）/提供者IDで検索し、来歴管理システムから履歴識別子を取得。 |
|  | 「受信履歴」から「加工履歴」 | * 「加工履歴」を登録する加工履歴登録ツールが来歴管理システムから履歴識別子を検索。 * 履歴種別/リソースURL（ユニーク）/利用者IDで検索し、来歴管理システムから履歴識別子を取得。 |
|  | 「加工履歴」から「原本情報」 | * 「加工履歴」を来歴管理システムで登録する際に加工データのハッシュを登録。 * 「原本情報」で登録するデータのハッシュと「加工履歴」内のハッシュ値を照合し、一致する場合に、システムが前段識別子を設定。 |
| 組織間での履歴登録 |  | 「送信履歴」から「受信履歴」 | * 提供者コネクタが「送信履歴」を登録後に取得する履歴識別子を、データともに利用者コネクタに送信。 * 利用者コネクタがデータの受信後に「受信履歴」を登録、その際に提供者コネクタから送信する「送信履歴」の履歴識別子を取得。 |

## データ来歴の構成

来歴取得では、指定される履歴識別子の履歴を起点として、履歴内の追跡用情報が指し示す履歴を順々に追跡し、全体を来歴データとして構成し提供する。

図表4-5に来歴の構成を示す。

**データ原本情報**

**[履歴基本情報]**

履歴識別子

履歴種別 （原本）

**[追跡用情報]**

後段履歴識別子

**[登録情報]**

登録者情報

データA情報

原本情報

**データ加工履歴**

**[履歴基本情報]**

履歴識別子

履歴種別 （加工）

**[追跡用情報]**

前段履歴識別子

後段履歴識別子

**[登録情報]**

登録者情報

データB情報

原本情報

**データ原本情報**

**[履歴基本情報]**

履歴識別子

履歴種別 （原本）

**[追跡用情報]**

前段履歴識別子

後段履歴識別子

**[登録情報]**

登録者情報

データB情報

原本情報

**データ送信履歴**

**[履歴基本情報]**

履歴識別子

履歴種別 （送信）

**[追跡用情報]**

前段履歴識別子

後段履歴識別子

**[登録情報]**

登録者情報

取引相手情報

データB情報

**データ受信履歴**

**[履歴基本情報]**

履歴識別子

履歴種別 （受信）

**[追跡用情報]**

前段履歴識別子

後段履歴識別子

**[登録情報]**

登録者情報

取引相手情報

データB情報

**起点**

図表4-5 来歴の構成

# 外部API

来歴管理システムは、（a）履歴登録API、（b）来歴確認API、（c）履歴検索APIの主な3つのAPIを提供する。更に、ユーザー管理のためのAPIも提供する。

履歴登録APIは、履歴を残す事象が発生した際に使用する。履歴を新たに登録する際は、過去の関連する履歴に連結する形で登録することが可能である。その場合、連結対象となる履歴のIDを指定する。

来歴確認APIは、指定した履歴のIDを起点とした一連の履歴（来歴）を確認する際に使用する。指定した履歴のIDを起点として、来歴の方向をパラメータパラメーターで指定することができる。指定した位置から過去の方向へ来歴をたどる場合はBackwardを指定し、未来の方向はForwardを指定する。

履歴検索APIは、検索条件に一致した履歴を取得する際に使用する。来歴管理システムでは、システムが自動的に生成するデータ項目に加えて、ユーザーが任意で定義できる項目を追加することが可能であり、履歴検索APIではこの両者を区別することなく検索可能である。

ユーザー管理APIは、本システムで利用するユーザーアカウント情報の登録と削除が可能である。また、ユーザー管理APIを実施するために必要なセッションIDを発行するAPIも含まれる。

以下に各APIの詳細を記載する。

## 履歴登録API

### **名称**

履歴登録API: eventwithhash

### **API概要**

（a）本APIは、入力した情報を履歴として登録し、登録したイベントの識別子を返すものである。入力情報にデータの本体の情報が記載してある場合は、データ本体は記録せず、代わりにハッシュ値を生成し履歴情報に登録する。

### **API詳細**

APIの詳細を図表5-1-3に記す。

図表5-1-3 履歴登録API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
| メソッド | | POST | | |
| パス | | /eventwithhash | | |
|  | |  | | |
| 説明 | | eventwithhash APIは、記録するイベントをJSONオブジェクトで入力する。本APIでは、データ本体のハッシュ値とイベント、タグを登録する。  本APIではmulti-part形式でデータを入力する。先頭にはContent-Type :application / json形式のJSONオブジェクトを指定し、続けて、ハッシュ値の計算に必要となるファイルを必要な数だけ入力する。  cdldatatags内に記述したタグの数と同じ数のファイルを入力する必要があり、一致しない場合はエラー応答を返す。  本APIではファイルのsha256値のみを計算するが、ファイルの中身自体は保存しないことに留意すること。  このファイルの実体保存に関しては本APIでは関与しないため、ファイルの実体の保存管理は自己責任とする。  本APIではファイルのハッシュ値を計算した後、cdldatatagsそれぞれのオブジェクトの内部にcdlsha256hash属性で、値を記録する。この時、cdlsha256hash属性が既に存在する場合は、計算したハッシュ値で上書きする。 | | |
| 入力 | URIパスパラメータ |  | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング |  | |  |
| HTTPリクエストボディ | 登録するイベントおよびデータをマルチパート形式で指定する。  マルチパートの最初のパートは記録するイベントをJSON形式で格納する。Content-Dispositionヘッダのnameを”request”とし、Content-Typeヘッダにapplication/jsonを指定する。  ２番目以降のパートはデータ本体を格納する。Content-Dispositionヘッダのnameを"upfile"とし、Content-TypeヘッダにデータのContent-Typeを指定する。  以下にリクエストボディの例を記す。  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="request"; filename="req.json"  Content-Type: application/json  {  "cdldatamodelversion":"2.0",  "cdleventtype":"Foo", // イベントタイプ  “user-defined-key1”: “user-defined-value1”, // ユーザー定義の項目  "cdldatatags": [ // タグの記述（本対応では1つのタグのみ）  {  "cdluri":"https://somewhere.else/a.txt"  },  {  "user-defined-key2”: “user-defined-valu2” // ユーザー定義の項目  }  ]  }  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="upfile"; filename="a.txt"  Content-Type: text/plain  aaa  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="upfile"; filename="b.txt"  Content-Type: text/plain  bbb  ------FormBoundaryFooBarBaz—  cdleventtypeでイベントの種類を指定する。この種類については、5つのケースで後述する。  上記例では、2つデータタグがあるので APIに2つのファイルを渡す必要がある。ほとんどのケースでは、1つのデータタグで十分である。 | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

### **対象シーン**

履歴登録には以下の5つのシーンがある。

それぞれのシーンと内容について図表5-1-4に示す。

図表 5-1-4 履歴登録のシーン

|  |  |  |
| --- | --- | --- |
| **No** | **シーン** | **内容** |
| 1 | データ原本情報登録 | データの原本情報の登録を意図したものである。過去に連結する来歴が存在しない新規の登録の際に使用する。 |
| 2 | （二次）データ原本情報登録 | データの原本情報の登録を意図したものであるが、過去に連結する来歴が存在し、それに連なる登録の際に使用する。 |
| 3 | 送信履歴登録 | データ送信における送信履歴の登録である。 |
| 4 | 受信履歴登録 | データ受信における受信履歴の登録である。 |
| 5 | データ加工履歴登録 | データ加工におけるデータ加工履歴の登録である。 |

以下にそれぞれのシーンについて述べる。

#### **データ原本情報登録**

データ原本情報登録は、原本の登録を意図したものであり、過去に連結する来歴が存在しない新規の登録の際に使用するものである。

APIのパラメータは、図表5-1-4-1の通りである。

図表5-1-4-1 データ原本情報登録におけるパラメータ設定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
| 入力 | URIパスパラメータ | - | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | |  |
| HTTPリクエストボディ | {  "cdldatamodelversion":"2.0",  "cdleventtype":"Create", // イベントタイプ（必須） : 新規来歴登録を示す"Create"を入力  "dataprovider":"AAA-BBB", // データ提供者ID: コネクタ管理のデータ提供者のIDを入力 （ユーザー定義）  "cdldatatags":[ // タグの記述（本対応では１つのタグを指定）  {  "cdluri":"https://XXXXX（略）", // データのURI: コネクタ側で管理するデータのURIを入力 （ユーザー定義）  }  ]  }  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="upfile"; filename="a.txt"  Content-Type: text/plain  Aaa | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

#### **（二次）データ原本情報登録**

（二次）データ原本情報登録は、原本の登録を意図したものであるが、過去に連結する来歴が存在し、それに連なる登録の際に使用するものである。

APIのパラメータは、図表5-1-4-2の通りである。

図表5-1-4-2 （二次）データ原本情報登録のパラメータ設定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
|  | |  | |  |
| 入力 | URIパスパラメータ | - | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | |  |
| HTTPリクエストボディ | イベントタイプには"Publish"の代わりに"Create"を使っても良い。この時の"previous event"は省略可能で、略した場合同一のキッシュ値を持つデータを探して自動的に接続する。  {  "cdldatamodelversion":"2.0",  "cdleventtype":"Publish", 　　　　// イベントタイプ（必須） : 公開履歴登録を示す"Publish"を入力  “dataprovider":"AAA-BBB", // データ提供者ID: コネクタ管理のデータ提供者のIDを入力 （ユーザー定義）  "cdlpreviousevents": [  "aaaa-bbbb-cccc-dddd", 　 // 前段のイベント識別子を入力  ],  "cdldatatags": [ 　// タグの記述（本対応では1つのタグを指定）  {  "cdluri":"https://XXXXX（略）", // データのURI: コネクタ側で管理するデータのURIを入力 （ユーザー定義）  }  ]  }  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="upfile"; filename="a.txt"  Content-Type: text/plain  Aaa | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

#### **送信履歴登録**

送信履歴登録は、データの送信を記録するためのものであり、データの送受信者の情報を指定する必要がある。

APIのパラメータは、図表5-1-4-3の通りである。

本ケースでは、ファイルの指定はないが、マルチパート形式で呼び出さなければならないことに留意すること。

図表5-1-4-3 送信履歴登録のパラメータ設定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
| 入力 | URIパスパラメータ | - | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | |  |
| HTTPリクエストボディ | {  "cdldatamodelversion":"2.0",  "cdleventtype":"Sent", 　　　　　　 // イベントタイプ（必須） : 送信履歴登録を示す"Sent"を入力  "dataprovider":"AAA-BBB", // データ送信者（提供者）ID: コネクタ管理のデータ提供者IDを入力 （ユーザー定義）  "datauser":"CCC-DDD", // データ受信者（利用者）ID: コネクタ管理のデータ利用者IDを入力 （ユーザー定義）  "cdlpreviousevents": [  "aaaa-bbbb-cccc-dddd", // 前段のイベント識別子を入力。  ]  } | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

#### **受信履歴登録**

受信履歴登録は、データの受信を記録するためのものであり、データの送受信者の情報を指定する必要がある

　APIのパラメータは、図表5-1-4-4の通りである。

本ケースでは、ファイルの指定はないが、マルチパート形式で呼び出さなければならないことに留意すること。

図表5-1-4-4 受信履歴登録のパラメータ設定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
| 入力 | URIパスパラメータ | - | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | |  |
| HTTPリクエストボディ | {  "cdldatamodelversion":"2.0",  "cdleventtype":"Received", // イベントタイプ（必須） : 受信履歴登録を示す"Received"を入力  "dataprovider":"AAA-BBB", // データ送信者（提供者）ID: コネクタ管理のデータ提供者IDを入力 （ユーザー定義）  "datauser":"CCC-DDD", // データ受信者（利用者）ID: コネクタ管理のデータ利用者IDを入力 （ユーザー定義）  "cdlpreviousevents": [  "aaaa-bbbb-cccc-dddd", // 前段のイベント識別子を入力。  ]  } | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

#### データ加工履歴登録

データ加工履歴登録は、加工したデータまたは加工情報を登録するためのものであり、データの加工者の情報を指定する必要がある

APIのパラメータは、図表5-1-4-5の通りである。

図表5-1-4-5 受信履歴登録のパラメータ設定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | | **内容** | | |
| 入力 | URIパスパラメータ | - | |  |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | |  |
| HTTPリクエストボディ | 加工情報の登録の場合、２番目のパートのContent-Typeはapplication/jsonであり、JSON形式の加工情報を格納する。  {  "cdldatamodelversion":"2.0",  "cdleventtype":"Update", // イベントタイプ（必須） : データ加工履歴登録を示す"Update"を入力  "dataprovider":"AAA-BBB", // データ加工者ID: コネクタ管理のデータ提供者のIDを入力 （ユーザー定義）  "cdldatatags":[ // タグの記述（本対応では１つのタグを指定）  {  "cdluri":"https://XXXXX（略）", // データまたは加工情報のURI  }  ]  }  ------FormBoundaryFooBarBaz  Content-Disposition: form-data; name="upfile"; filename="a2.txt"  Content-Type: text/plain  Aaa2 | | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 | |
| それ以外の値 | エラー | |
| HTTPレスポンスヘッダー |  | | |
| HTTPレスポンスボディ | レスポンスボディはJSONで以下のような値を返す。  {  "cdleventid":"string" // イベント識別子の返却  }  本APIで生成したイベントIDを文字列で返す。 | | |

## 来歴確認API

### **名称**

履歴確認API: lineage

### **API概要**

（b）本APIは、登録済みの履歴に対して、履歴の識別子を指定することで、その履歴を基点とした一連の来歴を取得する際に使用する。来歴の取得の際は検索方向および、検索深さをパラメータで指定することができる。以下に詳細を述べる

### **API詳細**

APIの詳細を図表5-2-3に記す。

図表5-2-3 来歴確認API

|  |  |  |  |
| --- | --- | --- | --- |
| **項目** | | **内容** | |
| メソッド | | GET | |
| パス | | /lineage/{event\_ID} | |
| 説明 | | 本APIは、指定したイベントIDから指定した方向へ指定した深さ分の来歴情報を取得する。 | |
| 入力 | URIパスパラメータ | event\_ID （必須） | 本APIでは、探索の開始地点となるイベントIDをパラメータとして指定する必要がある。 |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | direction | 本APIでは、探索の方向をBACKWARD、FORWARD、BOTHの文字列で指定する。省略した場合、BACKWARDがデフォルト値となる。 |
| depth | 本APIでは、探索の深さを整数値で指定する。0を指定した場合は、開始地点のオブジェクトのみが返り、1以上の正の整数を指定した場合は、その個数分の到達可能なオブジェクトが返る。また、-1を指定した場合は、全ての到達可能なオブジェクトが返る。 |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 |
| それ以外の値 | エラー |
| HTTPレスポンスヘッダー |  | |
| HTTPレスポンスボディ | 正常に動作した場合は、レスポンスコードとして200が返る。  返り値として、該当するjsonオブジェクトの配列を返す。  [  {  "cdldatamodelversion":"2.0",  "cdleventid":"194430e2-0f28-4bd4-8674-6ef6ec0198b0",  "cdleventtype":"string",  "cdlorganization":"string",  "cdltimestamp":"string",  "cdlpreviousevents": [  "string"  ],  "cdlnextevents": [],  "cdldatatags": [  {  "cdluri":"https://XXXXX（略）",  "cdlsha256hash": "b589b63bc6217ed0be9ed63b6a88984fc61e0732bbb5a7fb05a19c7b764b9c5b"  }  ]  }  ] | |

### **API実施例**

#### **APIパラメータ**

実際にAPIを実施した動作例を図表5-2-4-1に記す。

図表5-2-4-1 API動作例

|  |  |  |  |
| --- | --- | --- | --- |
| **項目** | | **内容** | |
| 入力 | URIパスパラメータ | event\_ID （必須） | パラメータとして、イベントIDにスタート地点となるイベント識別子を指定する。 |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | direction | 方向にはデフォルト値であるBACKWARDを指定する。 |
| depth | 深さにはデフォルト値である-1を指定する。 |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 |
| それ以外の値 | エラー |
| HTTPレスポンスヘッダー |  | |
| HTTPレスポンスボディ | 指定した位置からBACKWORD方向に探索を行い、3つのイベントが返ってきているのがわかる。  [  {  "cdldatamodelversion":"2.0",  "cdleventid":"aaaa-bbbb-cccc-dddd", //イベント識別子  "cdleventtype":"Received", // 受信履歴  "dataprovider":"AAA-BBB", // データ送信者  "datauser":"CCC-DDD", // データ受信者  "cdlorganization":"org01.cdl.com",  "cdltimestamp":"2022-02-10T04:17:13.802Z",  "cdlpreviousevents": [  　"bbbb-cccc-dddd-eeee" // 前段イベント識別子  ],  "cdlnextevents": []  },  {  "cdldatamodelversion":"2.0",  "cdleventid":"bbbb-cccc-dddd-eeee", //イベント識別子  "cdleventtype":"Sent", // 送信履歴  "datadrovider":"AAA-BBB", // データ送信者  "datauser":"CCC-DDD", // データ受信者  "cdlorganization":"org02.cdl.com",  "cdltimestamp":"2022-02-10T04:17:02.593Z",  "cdlpreviousevents": [  　"dddd-eeee-aaaa-bbbb" // 前段イベント識別子  ],  "cdlnextevents": [  　"aaaa-bbbb-cccc-dddd" // 後段イベント識別子  ]  },  {  "cdldatamodelversion":"2.0",  "cdleventid":"dddd-eeee-aaaa-bbbb", //イベント識別子  "cdleventtype":"Create", // 新規登録履歴  "dataprovider":"AAA-BBB", // データ提供者  "cdlorganization":"org02.cdl.com",  "cdltimestamp":"2022-02-10T04:07:03.898Z",  "cdldatatags": [ // タグの記述  {  "cdluri":"https://XXXXX（略）", // データ本体のURI  "cdlsha256hash":"xxxxxx（略）" // データのハッシュ値  }  ],  "cdlpreviousevents": [], // 前段イベント識別子  "cdlnextevents": [  　"bbbb-cccc-dddd-eeee" // 後段イベント識別子  ]  }  ] | |

## 来歴管理I/F:履歴検索API

### **名称**

履歴検索API: searchevents

### **API概要**

（c）本APIは、検索条件に一致した履歴情報を取得する際に使用する。来歴管理システムでは、システムが自動的に生成するデータ項目に加えて、ユーザーが任意で定義できる項目を追加することが可能であり、履歴検索APIではこの両者を区別することなく検索可能である。

### **API詳細**

APIの詳細を図表5-3-3に示す。

図表5-3-3履歴検索API

|  |  |  |  |
| --- | --- | --- | --- |
| **項目** | | **内容** | |
| メソッド | | POST | |
| パス | | /searchevents | |
| 説明 | | 本APIは、指定した検索条件に一致する一連のイベントを返す。検索条件の詳細については、次章を参照のこと。  本APIは、リクエストヘッダーにcontent-type:application / json行を含める必要がある。 | |
| 入力 | URIパスパラメータ | - | - |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| クエリストリング | - | - |
| HTTPリクエストボディ | 本APIは、検索条件を、CouchDB構文で指定し検索を実施する。CouchDB構文の詳細に関しては、以下のURLを参照のこと。  　https://docs.couchdb.org/en/stable/api/database/find.html#find-selectors  例えば、このJSON記述は、「dataprovider」の値として「AAA-BBB」を持つイベントを検索するものである。  {  "selector": {  "dataprovider":"AAA-BBB"  }  } | |
| 出力 | HTTPレスポンスコード | 200 | 正常に動作した場合、本APIはレスポンスコードとして200を返す。 |
| それ以外の値 | エラー |
| HTTPレスポンスヘッダー |  | |
| HTTPレスポンスボディ | 本APIでは、指定した検索条件に一致したイベントが配列で返る。  その際、リスト内の順序は意味を持たないことに留意すること。  [  {  "cdldatamodelversion":"2.0",  "cdleventid":"94430e2-0f28-4bd4-8674-6ef6ec0198b0",  "cdleventtype":"string",  "cdlorganization":"string",  "cdltimestamp":"string",  "cdlpreviousevents": [  "string"  ],  "cdlnextevents": [],  "cdldatatags": [  {  "cdluri":"https://XXXXX（略）",  "cdlsha256hash": "b589b63bc6217ed0be9ed63b6a88984fc61e0732bbb5a7fb05a19c7b764b9c5b"  }  ],  }  ] | |

## ユーザー管理API

### **API概要**

ユーザー管理APIはユーザーの新規登録及び削除する時に使用する。使用の際には管理用セッションIDを取得し、そのIDをリクエストヘッダーに付与してAPIを呼びだす。

### **新規ユーザー登録**

新規ユーザー登録の詳細を図表5-4-2に示す。

図表5-4-2新規ユーザー登録API

|  |  |  |  |
| --- | --- | --- | --- |
| 機能名 | | 新規ユーザー登録 | |
| 機能説明 | | 新たなユーザーを登録する | |
| HTTPメソッド | | POST | |
| URIパス | | /adduser | |
| 入力 | URIパスパラメータ | - | - |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| X-CDL-SessionId | 認証APIで取得した  “セッションID文字列”  新規ユーザー登録を実行する権限を持った管理者権限アカウントである。 |
| Content-Type | application/json |
| クエリストリング | - | - |
| HTTPリクエストボディ | 新規に登録するユーザー情報のJSONオブジェクト  {  “cdluserid”: “(ユーザーID)”,  “cdlorganization”: “(組織ID)”,  “cdlrole”: “(ユーザロール)”,  “cdlpassword”: “(ユーザーパスワード)”  }  cdlorganization及びcdlroleは省略可能。省略した場合は、デフォルト値が設定される。  外部IdPと連携する場合は、cdlpasswordの値は使用しない。使用しない場合であっても、cdlpasswordにはダミーの文字列(何でもよい)を指定すること | |
| 出力 | HTTPレスポンスコード | 200 | 正常終了 |
| (以降エラー未稿。API定義yaml参照) |  |
|  |  |
| HTTPレスポンスヘッダー | - | - |
| HTTPレスポンスボディ | - | |

### **ユーザー情報削除**

ユーザー情報削除APIの詳細を図表5-4-3に示す。

図表5-4-3ユーザー情報削除API

|  |  |  |  |
| --- | --- | --- | --- |
| 機能名 | | ユーザー情報削除 | |
| 機能説明 | | ユーザー情報を削除する | |
| HTTPメソッド | | POST | |
| URIパス | | /deluser | |
| 入力 | URIパスパラメータ | - | - |
| HTTPリクエストヘッダー | ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| X-CDL-SessionId | 認証APIで取得した  “セッションID文字列”  ユーザー情報削除を実行する権限を持った管理者権限アカウントである。 |
| Content-Type | application/json |
| クエリストリング | - | - |
| HTTPリクエストボディ | 削除するユーザー情報のJSONオブジェクト  {  "cdluserid": "(ユーザーID)",  } | |
| 出力 | HTTPレスポンスコード | 200 | 正常終了 |
| (以降エラー未稿。API定義yaml参照) |  |
|  |  |
| HTTPレスポンスヘッダー | - | - |
| HTTPレスポンスボディ | - | |

### **管理用セッションID取得**

管理用セッションID取得APIの詳細を図表5-4-4に示す。

図表5-4-4管理用セッションID取得API

|  |  |  |  |
| --- | --- | --- | --- |
| 機能名 | | 管理用セッションID取得 | |
| 機能説明 | | 管理用セッションIDを取得する | |
| HTTPメソッド | | POST | |
| URIパス | | /user/enroll | |
| 入力 | HTTPリクエストヘッダー | userid | 認証するユーザーID  (Hyperledger Fabric のアカウント) |
| ID\_Tokenまたは Authorization | 外部IdPから取得したトークン  ”IDトークン文字列” |
| X-CDL-Auth | 認証するユーザーのパスワード文字列 |
| Content-Type | application/json |
| クエリストリング | - | - |
| HTTPリクエストボディ | - | |
| 出力 | HTTPレスポンスコード | 200 | 正常終了 |
| (以降エラー未稿。API定義yaml参照) |  |
|  |  |
| HTTPレスポンスヘッダー | Content-Type | application/json |
| HTTPレスポンスボディ | 認証結果のJSONオブジェクト。管理用セッションIDの文字列を返す。  管理用のユーザーID及びパスワードは事前に環境変数で設定しておくものとする。  {  "cdlsessionid": "(管理用セッションID文字列)"  } | |

# 来歴管理システム構成

本章では、来歴管理システムにおけるシステム構成を示す。本来歴管理システムは、ブロックチェーンのOSSであるHyperleder Fabricをベースとしている。

## 全体構成

図表6-1 に来歴管理システムの全体構成を示す。来歴管理システムは、大きく以下から構成される。

* 組織毎の来歴管理ノード:組織毎に提供されるノードであり、組織は対象のノードに向けて、履歴の登録および来歴確認の要求を行う。
* 組織共通の来歴管理組織横断ノード:組織共通に利用するノードであり、システム全体におけるユーザー管理や来歴管理ノード間の情報の同期制御を行う。

図表6-1 来歴管理システムの全体構成

組織A

**来歴管理組織横断ノード**

**来歴管理ノード**

Fabric CA

[ 認証局 ]

来歴管理 API

来歴構成管理

Fabricアプリ

[ 履歴書込/読出 ]

Fabricノード

[履歴管理]

組織B

**来歴管理ノード**

来歴管理 API

来歴構成管理

Fabricアプリ

[ 履歴書込/読出 ]

Fabricノード

[履歴管理]

**ユーザー管理**

**ロール管理**

Orderer

[同期制御]

**履歴**

**履歴**

**同期ポリシー**

**同期ポリシー**

## 機能内容

図表6-2-1に各組織における来歴管理ノードの機能内容を示す。

図表6-2-1来歴管理ノードの機能内容

|  |  |  |
| --- | --- | --- |
| **機能** | **内容** | **備考** |
| 来歴管理API | 来歴利用に対するRest APIを提供する。 | APIの内容は5章に記載する。 |
| 来歴構成管理 | 来歴管理API を介したユーザリクエストに応じて、管理する履歴に対し、各格納先（Hyperledger Fabric）へのアクセス振り分け、ならびに履歴情報の分解・再統合処理を行う。 |  |
| Fabricアプリ | Hyperledger Fabricへの履歴登録および履歴読み込みを行う。 |  |
| Fabricノード | ユーザーから登録された履歴を管理する。 |  |

　図表6-2-2に組織共通の来歴管理組織横断ノードの機能内容を示す。

図表6-2-2来歴管理組織横断ノードの機能内容

|  |  |  |
| --- | --- | --- |
| **機能** | **内容** | **備考** |
| Fabric CA | ユーザーおよびロール管理を行う。 |  |
| Orderer | ユーザーから登録要求があった履歴に対する各Fabricノードへの書き込みの制御を行う。 |  |

## 機能間シーケンス

来歴管理システムAPIの呼出しの際に来歴管理システム内の機能間のシーケンスを示す。

### **全API共通認証**

来歴管理システムは、全てのAPIの実行に先立ちまずユーザー認証を実行するようになっている。このシーケンスはAPIにかかわらず共通となっておりシーケンスの概要について図表6-3-1に示す。

各APIの詳細については次章6.3.2以降に記載する。

フロントサーバー

**来歴管理API**

**来歴構成管理**

アプリケーション

Fabric CA

[ 認証局 ]

**外部IdP**



**ユーザー管理**

**ユーザー管理**

**ロール管理**

アカウントAさん

（組織アカウント）

ユーザーアカウント=A

管理用セッションID取得

ユーザー

登録フェーズ

このフェーズは省略可能

ユーザー登録

管理用セッションID

図表 6-3-1　全API共通認証

結果を通知

API実行結果

各API実行  
・履歴登録  
・来歴確認  
・履歴検索

ユーザーアカウント

存在確認

No

アカウント無しの場合のみ

新規アカウント登録

アカウント登録

検証結果とデコード結果を取得

①ユーザーID=A

①ユーザーアカウント=A

①ユーザーアカウント=削除

API発行

トークン検証

アカウント削除

管理用セッションID

ユーザー削除

管理用セッションID取得

ユーザー

削除フェーズ

IDトークン発行

IDトークン確認

認証フェーズ

ID/Pass

認証

IDトークン

アカウント登録

### **履歴登録API**

図表6-3-2に外部から履歴登録APIを呼出しされたときの来歴管理システム内の機能間シーケンスを示す。

**アプリケーション**

**外部IdP**

**来歴管理API  
（フロントサーバー）**

**Fabric CA**

**来歴構成管理**

**Orderer**

**Fabricアプリ**

**Fabricノード**

図表6-3-2 履歴登録API呼出しにおける機能間シーケンス

**トークン取得フェーズ**

1呼出元認証

ID/Pass

認証

IDトークン

**API発行フェーズ**

IDトークン

2呼出元APIリクエスト

**トークン検証フェーズ**

3呼出元API検証確認ID取得

IDトークン

検証結果とデコード結果を取得

①ユーザーID=A

ユーザーアカウント存在確認

No

アカウント無しの場合のみ

新規アカウント登録

アカウント登録

4ユーザーアカウント

存在確認

5履歴登録リクエスト

(前段履歴識別子など)

**API実行フェーズ**

6ハッシュ計算 イベント・タグ生成

7イベント・タグ登録

8証明書申請

9証明書発行

10イベント検証要求

11検証結果

12イベント登録

13イベント登録

14タグ検証要求

15検証結果

16タグ登録

17タグ登録

18登録完了

19イベント識別子

20イベント識別子

API実行結果

の通知

以下に、図表6-3-2におけるフローの内容を示す。

1. 「アプリケーション」は「外部IdP」に”ID/Pass”で認証し[IDトークン]発行を要求する。「外部IdP」は対応する[IDトークン]を「アプリケーション」に返す。この操作はトークンが取得できるまで複数回実施する可能性がある。
2. 「アプリケーション」は「来歴管理API」へ履歴登録APIの利用をリクエストする。「アプリケーション」はAPIのパラメータとして、”前段イベント識別子”等を入力し（詳細はAPIの入力情報を参照のこと）、1.で取得したIDトークンも同時に渡す。
3. 「来歴管理API」は「外部IdP」に [IDトークン]の検証を依頼する。「外部IdP」はトークンを検証し、デコード結果[ユーザーID]を「来歴管理API」へ返す。
4. 「来歴管理API」は「FabricCA」にユーザーアカウントの存在確認を要求する。「FabricCA」は存在確認結果で有無を判断し(無い場合は新規登録)ユーザーアカウント結果を「来歴管理API」に返す。
5. 「来歴管理API」はリクエストを受信し、「来歴構成管理」に対し履歴登録の処理を要求する。
6. 「来歴構成管理」は入力情報から”イベント”および”タグ”を生成する。その際、”タグ”のハッシュ値を算出し”イベント”に入れる。また、入力情報に”データ本体”があれば、そのハッシュ値を算出し”タグ”に入れる。
7. 「来歴構成管理」は「Fabricアプリ」に生成した”イベント”および”タグ”の登録を要求する。
8. 「Fabricアプリ」は「Fabric CA」に”証明書”を申請する。
9. 「Fabric CA」は”証明書”を発行し、「Fabricアプリ」に返す。
10. 「Fabricアプリ」は「Fabricノード」に”イベント”登録における検証を要求する。
11. 「Fabricノード」は「Fabricアプリ」に7.の検証結果を返す。
12. 「Fabricアプリ」は「Orderer」に”イベント”の登録を要求する。
13. 「Orderer」は”イベント”の情報を含むブロックを生成し、ブロックを全ての「Fabricノード」に登録する。
14. 「Fabricアプリ」は「Fabricノード」に”タグ”登録における検証を要求する。
15. 「Fabricノード」は「Fabricアプリ」に11.の検証結果を返す。
16. 「Fabricアプリ」は「Orderer」に”タグ”の登録を要求する。
17. 「Orderer」は”タグ”の情報を含むブロックを生成し、ブロックを全ての「Fabricノード」に登録する。
18. 「Fabricアプリ」は「来歴構成管理」に登録完了を通知する。
19. 「来歴構成管理」は4.で生成した”イベント識別子”を「来歴管理API」に通知する。
20. 「来歴管理API」は”イベント識別子”を「アプリケーション」に応答する。

なお、 5.～15の処理については、Hyperledger Fabricの利用バージョンに合わせて変更させる可能性がある。

### **履歴取得API**

図表6-3-3に外部から履歴取得APIを呼出しされたときの来歴管理システム内の機能間シーケンスを示す。

認証

**トークン取得フェーズ**

**トークン検証フェーズ**

**アプリケーション**

**Fabric CA**

**外部IdP**

**来歴管理API  
（フロントサーバー）**

**来歴構成管理**

**Orderer**

**Fabricアプリ**

**Fabricノード**

**API発行フェーズ**

**API実行フェーズ**

図表6-3-3 履歴取得API呼出しにおける機能間シーケンス

1呼出元認証

ID/Pass

12グローバルタグ

(イベント識別子)

(イベント識別子)

4ユーザーアカウント

存在確認

3呼出元API検証確認ID取得

2呼出元APIリクエスト

IDトークン

16来歴

15来歴

(イベントリスト)

9結果

(イベントリスト)

8結果

アカウント無しの場合のみ

新規アカウント登録

検証結果とデコード結果を取得①ユーザーID=A

(イベント識別子など)

6イベントリスト取得

(イベント識別子など)

7イベントリスト取得

IDトークン

API実行結果

の通知

11グローバルタグ取得

10グローバルタグ取得

13グローバルタグ

14来歴生成

(イベント識別子など)

5履歴取得

ユーザーアカウント存在確認

No

アカウント登録

IDトークン

以下に、図表6-3-3におけるフローの内容を示す。

1. 「アプリケーション」は「外部IdP」に”ID/Pass”で認証し[IDトークン]発行を要求する。「外部IdP」は対応する[IDトークン]を「アプリケーション」に返す。この操作はトークンが取得できるまで複数回実施する可能性がある。
2. 「アプリケーション」は「来歴管理API」へ履歴登録APIの利用をリクエストする。「アプリケーション」はAPIのパラメータとして、”イベント識別子”等を入力し（詳細はAPIの入力情報を参照のこと）、1.で取得したIDトークンも同時に渡す。
3. 「来歴管理API」は「外部IdP」に [IDトークン]の検証を依頼する。「外部IdP」はトークンを検証し、デコード結果[ユーザーID]を「来歴管理API」へ返す。
4. 「来歴管理API」は「FabricCA」にユーザーアカウントの存在確認を要求する。「FabricCA」は存在確認結果で有無を判断し(無い場合は新規登録)ユーザーアカウント結果を「来歴管理API」に返す。
5. 「来歴管理API」はリクエストを受信し、「来歴構成管理」に対し来歴確認の処理を要求する。
6. 「来歴構成管理」は「Fabricアプリ」に”イベント識別子”を入力として、それに連なるイベントのリストを要求する。
7. 「Fabricアプリ」は”イベント”を管理する「Fabricノード」にイベントのリストを要求する。
8. 「Fabricノード」は対象のイベントのリストを「Fabricアプリ」に返す。
9. 「Fabricアプリ」は「来歴構成管理」に取得したイベントのリストを返す。
10. 「来歴構成管理」は取得したイベントの”イベント識別子”を指定して「Fabricアプリ」に”タグ”を要求する。
11. 「Fabricアプリ」は「Fabricノード」に”イベント識別子”を指定して”タグ”を要求する。
12. 「Fabricノード」は「Fabricアプリ」に”タグ”を返す。
13. 「Fabricアプリ」は「来歴構成管理」に”タグ”を返す。
14. 「来歴構成管理」は”イベント”と”タグ”から”来歴”を生成する。
15. 「来歴構成管理」は”来歴”を「来歴管理API」を送る。
16. 「来歴管理API」は応答として”来歴”を「アプリケーション」に返す。

### **履歴検索API**

図表6-3-4に外部から履歴検索APIを呼出しされたときの来歴管理システム内の機能間シーケンスを示す。

認証

IDトークン

No

ユーザーアカウント存在確認

**トークン取得フェーズ**

アカウント登録

**トークン検証フェーズ**

**アプリケーション**

**Fabric CA**

**外部IdP**

**来歴管理API  
（フロントサーバー）**

**来歴構成管理**

**Orderer**

**Fabricアプリ**

**Fabricノード**

**API発行フェーズ**

**API実行フェーズ**

(イベント識別子／null)

6履歴検索

(イベント識別子／null)

11結果

(イベント識別子／null)

(検索キーなど)

9結果

(イベント識別子／null)

7履歴検索

(検索キーなど)

8結果

4ユーザーアカウント

存在確認

3呼出元API検証確認ID取得

2呼出元APIリクエスト

1呼出元認証

IDトークン

IDトークン

API実行結果

の通知

アカウント無しの場合のみ

新規アカウント登録

検証結果とデコード結果を取得

①ユーザーID=A

ID/Pass

図表6-3-4履歴検索API呼出しにおける機能間シーケンス

10結果

5履歴検索

以下に、図表6-3-4におけるフローの内容を示す。

1. 「アプリケーション」は「外部IdP」に”ID/Pass”で認証し[IDトークン]発行を要求する。「外部IdP」は対応する[IDトークン]を「アプリケーション」に返す。この操作はトークンが取得できるまで複数回実施する可能性がある。
2. 「アプリケーション」は「来歴管理API」へ履歴登録APIの利用をリクエストする。「アプリケーション」はAPIのパラメータとして、”検索キー”等を入力し（詳細はAPIの入力情報を参照のこと）、1.で取得したIDトークンも同時に渡す。
3. 「来歴管理API」は「外部IdP」に [IDトークン]の検証を依頼する。「外部IdP」はトークンを検証し、デコード結果[ユーザーID]を「来歴管理API」へ返す。
4. 「来歴管理API」は「FabricCA」にユーザーアカウントの存在確認を要求する。「FabricCA」は存在確認結果で有無を判断し(無い場合は新規登録)ユーザーアカウント結果を「来歴管理API」に返す。
5. 「来歴管理API」はリクエストを受信し、「来歴構成管理」に対し履歴検索の処理を要求する。
6. 「来歴構成管理」は「Fabricアプリ」に”検索キー”を入力として履歴検索の処理を要求する。
7. 「Fabricアプリ」は”イベント”を管理する「Fabricノード」に履歴検索を要求する。
8. 「Fabricノード」は検索結果を「Fabricアプリ」に返す。
9. 「Fabricアプリ」は「来歴構成管理」に取得した検索結果を返す。
10. 「来歴構成管理」は「来歴管理API」に検索結果を送る。
11. 「来歴管理API」は検索結果を「アプリケーション」に返す。

# 利用OSS一覧 （主要なもの）

来歴管理システムで利用しているOSSの説明とライスンスを記載する。

表 7‑1 来歴管理システムで利用するOSSおよびライセンス

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **用語** | **説明** | **ライセンス** | **備考** |
| 1 | Docker | コンテナ仮想化用プラットフォーム | Apache License 2.0 | - |
| 2 | Docker-Compose | 複数のコンテナを1つのサービスとして提供するために使用 | Apache License 2.0 | - |
| 3 | Hyperledger Fabric | ブロックチェーンの実装 | Apache License 2.0 | - |
| 4 | Minifabric | HＬＦの環境構築ツール | Apache License 2.0 | - |