Fujitsu Data e-TRUST

function manual

FUJITSU-DATA-E-TRUST

March, 2024

Preface

Purpose of This Manual

This document describes the features of Fujitsu Data e-TRUST.
Everyone involved with Fujitsu Data e-TRUST should read this manual first.

Audience for This Manual

This document is intended for anyone involved with the Service.
Chapter 1 provides an overview of this service, and Chapter 2-5 provides the main functions and information you need to understand to use this service.
Chapter 2-5 is intended for those who plan or develop applications and services that utilize this service.
To do so, you need to have the following knowledge.

- Basic knowledge of the Internet
- Basic knowledge of the Web API
- Basic knowledge of the database (DB)

For details of each function and API not described in this manual, refer to the API Reference Manual and the API Reference Manual: separate volumes.

Organization of Manuals

Please read the following manuals according to your purpose and purpose.

manual name	Purpose and purpose
Function Manual (this document)	This document provides an overview of the Service, main functions, and information you need to understand to use the Service.
API Reference Manual	Provides a detailed reference for using the Web API. It is written in HTML format.
API Reference Manual: separate volumes	Supplements the API Reference Manual. Please check it with the API reference.
Message Collection	This document describes the message contents and the action to be
Notes and Limitations	This document describes precautions and restrictions for using Fujitsu Data e-TRUST.
Release Information	Provides release information for Fujitsu Data e-TRUST.
Licensing Information	This document describes the license of the software used in this service.

Organization of This Document

This manual is organized as follows.

Chapter/Appendix	Contents
Chapter I Overview of Fujitsu Data e-TRUST	Provides an overview of Fujitsu Data e-TRUST.
Chapter II Prerequisites for using Fujitsu Data e-TRUST	This section describes the knowledge required to use Fujitsu Data e-TRUST. Please check it when developing applications using Fujitsu Data e-TRUST. Yes.

Chapter III Data distribution of Fujitsu Data e-TRUST	This section describes data management using the distributed data linkage function and consent management function of Fujitsu Data e-TRUST. Please confirm when using the data

Chapter/Appendix	Contents
Chapter IV Fujitsu Data e-TRUST trails and auditing	Describes the trail and audit features of Fujitsu Data e-TRUST. Please check when using the trail
Chapter V Fujitsu Data e-TRUST Trust Seal Function	Describes the trust seal feature of Fujitsu Data e-TRUST. Check this when using the trust seal
Appendix A JSON Format for the Trail and Audit Facility	Describes the JSON format required to take advantage of the trail and audit features.

Terms of Use of Opensource Software or Software Provided by Third Parties

For the terms and conditions of use of the opensource software used by this service or software provided by a third party, refer to the license information.

Export Control Regulation

If this document is to be exported or provided to a third party, please review the regulations of your country of residence and the U.S. export control laws and regulations before proceeding.

Change History

Edition	Date	Changes
1.0	2023/3/24	First edition published
1.1	2024/3/22	Revise the technology name from “Data e-TRUST” to “Fujitsu Data e-TRUST”

Copyright

Index

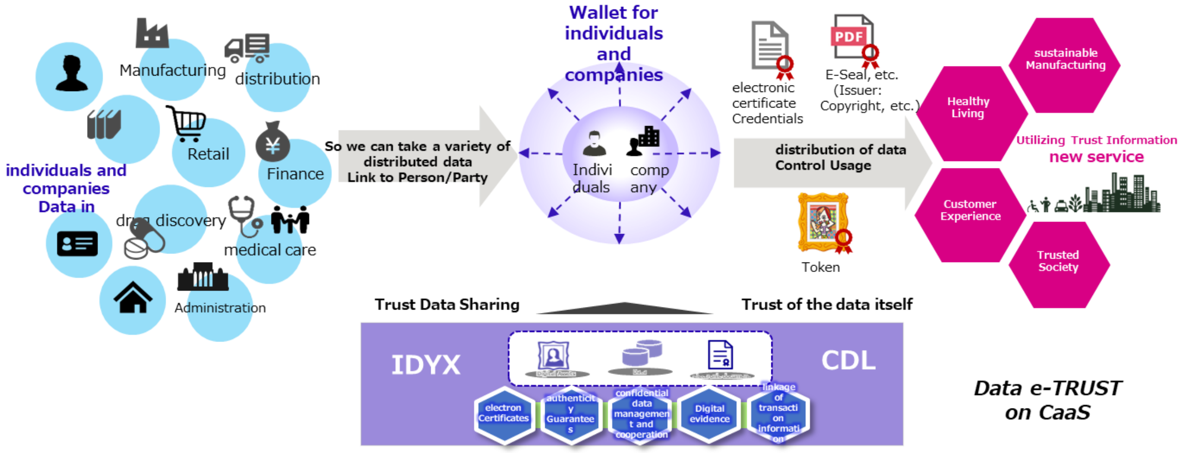
Chapter I Overview of Fujitsu Data e-TRUST

1.1 What is Fujitsu Data e-TRUST?

Fujitsu Trust Service Fujitsu Data e-TRUST provides a set of APIs for the safe and secure distribution and use of data between different services and between individuals and companies.
By using Fujitsu Data e-TRUST, Fujitsu's unique technologies IDentity eXchange (IDYX) and Chain Data Lineage (CDL), exchanged data can be managed in tamper-proof form along with certification of origin, ownership, and authenticity of data.
By granting trust to all online transactions involving data such as digital documents and digital contents, we support customers solve their business and social issues and contribute to the realization of a sustainable society.

Fujitsu Trust Service Data e-TRUST

Collaborate freely, safely and securely with distributed personal and corporate information

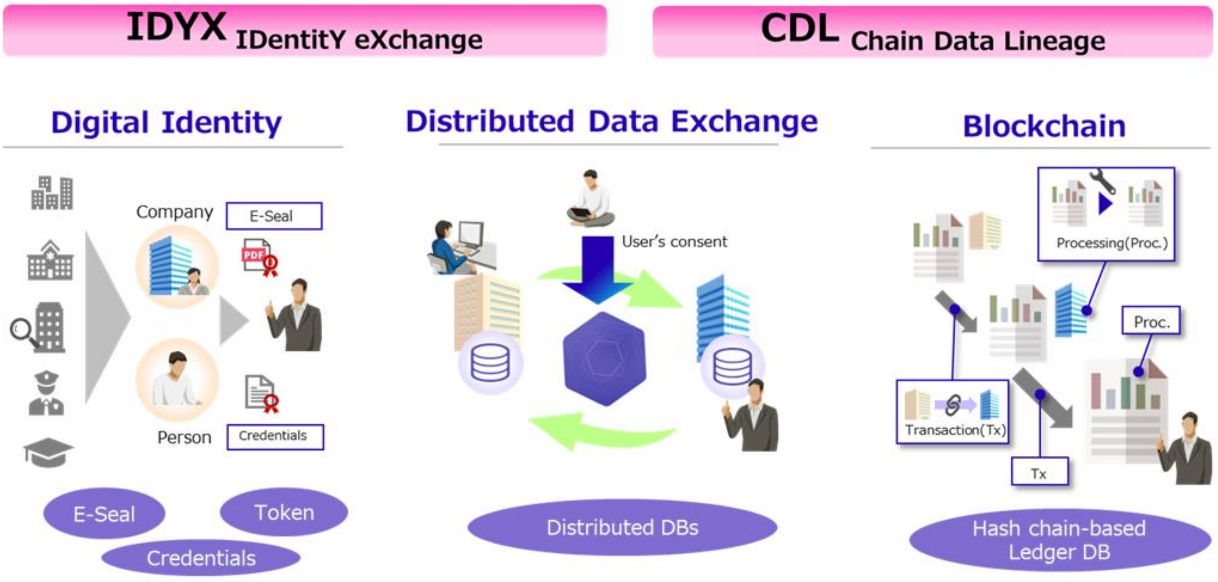

Manufacturing
distribution
Retail
Finance
individuals and companies Data in
drug discovery
medical care
Administration
So we can take a variety of distributed data Link to Person/Party
Wallet for individuals and companies
Indivi duals
comp any
electronic certificate Credentials
E-Seal, etc. (Issuer: Copyright, etc.)
distribution of data Control Usage
Token
sustainable Manufacturing
Healthy Living
Utilizing Trust Information
new service
Customer Experience
Trusted Society
Trust Data Sharing
Trust of the data itself
IDYX
CDL
Digital Assets
Data
electronic documents
electron Certificates
authenticity Guarantees
confidential data management and cooperation
Digital evidence
linkage of transaction information
Data e-TRUST on CaaS

[About IDYX Technology]

IDentityeXchange (IDYX) technology is Fujitsu’s technology that can ensure that the data being used is unalterable and has not been tampered with.
IDYX enables the issuance and use of various electronic certificates for digital information and ensures the authenticity of information exchanged in digital transactions.

[About CDL Technology]

Chain Data Lineage (CDL) technology is Fujitsu’s technology that can centrally manage transaction and activity history between individuals and companies using hash chain ledger technology. CDL enables you to store transaction trail in an unfalsifiable manner, as well as link and manage a series of activities between individuals and companies during transactions.



1.2 Fujitsu Data e-TRUST Features

Fujitsu Data e-TRUST has three features that enable the authentication of all information related to individuals and companies in digital transactions and the safe and free distribution of data.

- Secure, distributed data federation
- Ensuring data authenticity
- Tamper-proof evidence management

[Cooperation of Secure Distributed Data]

By providing consent and fine-grained access control for the provision of data by individuals and organizations, we provide data ownership and information disclosure management functions at the time of the provision of data, enabling data collaboration across people, organizations, and companies.

[Ensuring the Authenticity of Data]

It provides a variety of digital certificates for authenticating people, organizations, and companies and can be used in the authentication scene for a variety of services.

[Tamper-proof Evidence Management]

It connects and manages trails of transactions and activities across people, organizations, and companies, helping to provide advanced visibility into value chains and customer journeys.

1.3 Core Fujitsu Data e-TRUST Features

Fujitsu Data e-TRUST has three core features. The three core functions support the creation of ecosystems, business process reengineering, and new businesses and will vigorously promote DX across industries and solve issues in various business categories such as finance, manufacturing, distribution, and healthcare.

Three core features are:

- Trusted Data Hub
- Digital Proof
- Digital Footprint

[Trusted Data Hub]

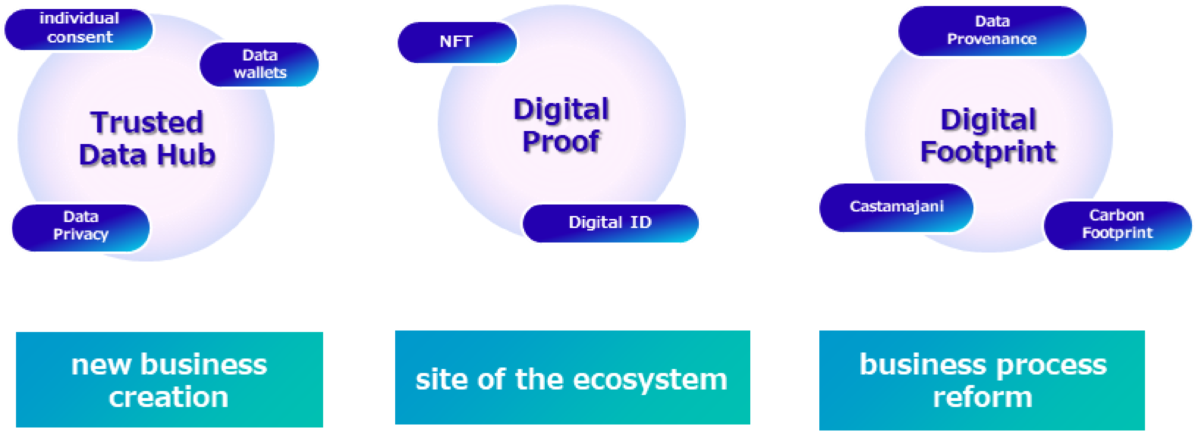
Data items to be linked are carefully controlled among distributed databases that are kept secret by individuals and companies and exchanged with users’ consents.
This enables secure, on-demand data collaboration across individuals and companies.
Fujitsu Data e-TRUST enhances data ownership and governance of information disclosure through fine-grained control over where data is distributed and privacy. As a result, individuals and companies can safely provide their own diverse data to multiple companies and services under their own control.

[Digital Proof]

IDYX technology, which ensures that the data to be used is correct and has not been tampered with, enables the issuance, and use of various electronic certificates for digital information and ensures the authenticity of information exchanged in digital transactions.
IDYX technology supports a variety of authentication scenarios to ensure the authenticity of digital information, such as strengthening the authentication process by checking individuals’ skills and careers, corporate authentication, one-stop contract procedures by mutually linking customer information, and management of copyright and ownership of digital documents and content.

[Digital Footprint]

CDL technology, extended blockchain technology, enables flexible, scalable, centralized management of transaction histories across individuals and companies. Digital transactions and activity trails are tied to individual and corporate interactions and managed in a tamper-proof manner. CDL technology makes it possible to use various transaction histories as evidence of the health of each business activity and social contribution.
For example, CDL can visualize and manage supply chains and values chains of carbon footprint and consumer behavior data related to CO2 emissions.



1.4 Functions of Fujitsu Data e-TRUST

Fujitsu Data e-TRUST provides five functions for safe and secure data distribution and utilization.
For details on each function, refer to the API Reference Manual and the API Reference Manual: separate volumes.

[Functions of Fujitsu Data e-TRUST]

- distributed data linkage function

  You can send and synchronize registered data between agents.
  The main flow of using the distributed data linkage function is described in Chapter 3.

- consent management function

  When sending or synchronizing data between agents, the data owner can process the agreement.
  The main flow of using the consent management function is described in Chapter 3.

- trail and audit function

  You can record and validate data transactions between agents.
  The main flow of using the trail and audit function is described in Chapter 4.

- trust seal function

  Verify that the data publisher and the data body have not been tampered with.
  The main flow of using the trust seal function is described in Chapter 5.

- management function

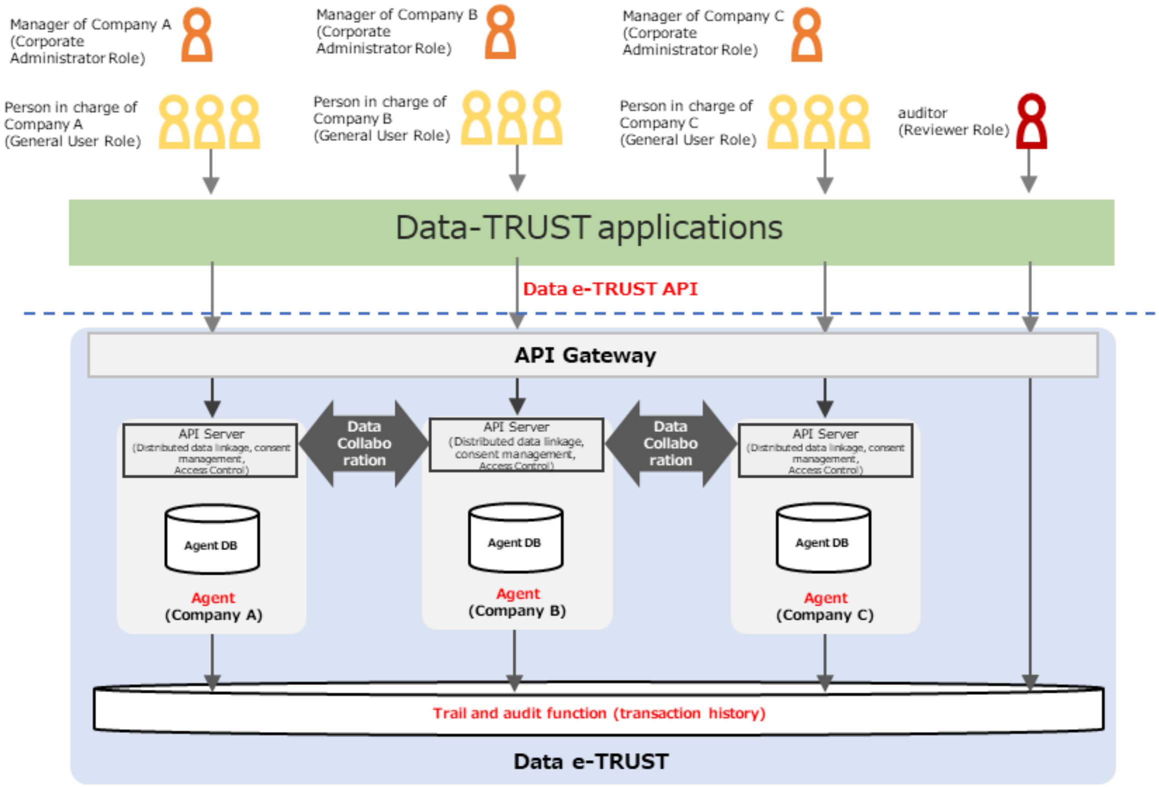
  Provides the functions required to use each Fujitsu Data e-TRUST function.
  For more information about the administration functions, see the API Reference Manual and the API Reference Manual: separate volumes.

Chapter II Prerequisites for Using Fujitsu Data e-TRUST

This chapter illustrates the terminology and concepts of this service to develop applications using Fujitsu Data e-TRUST.

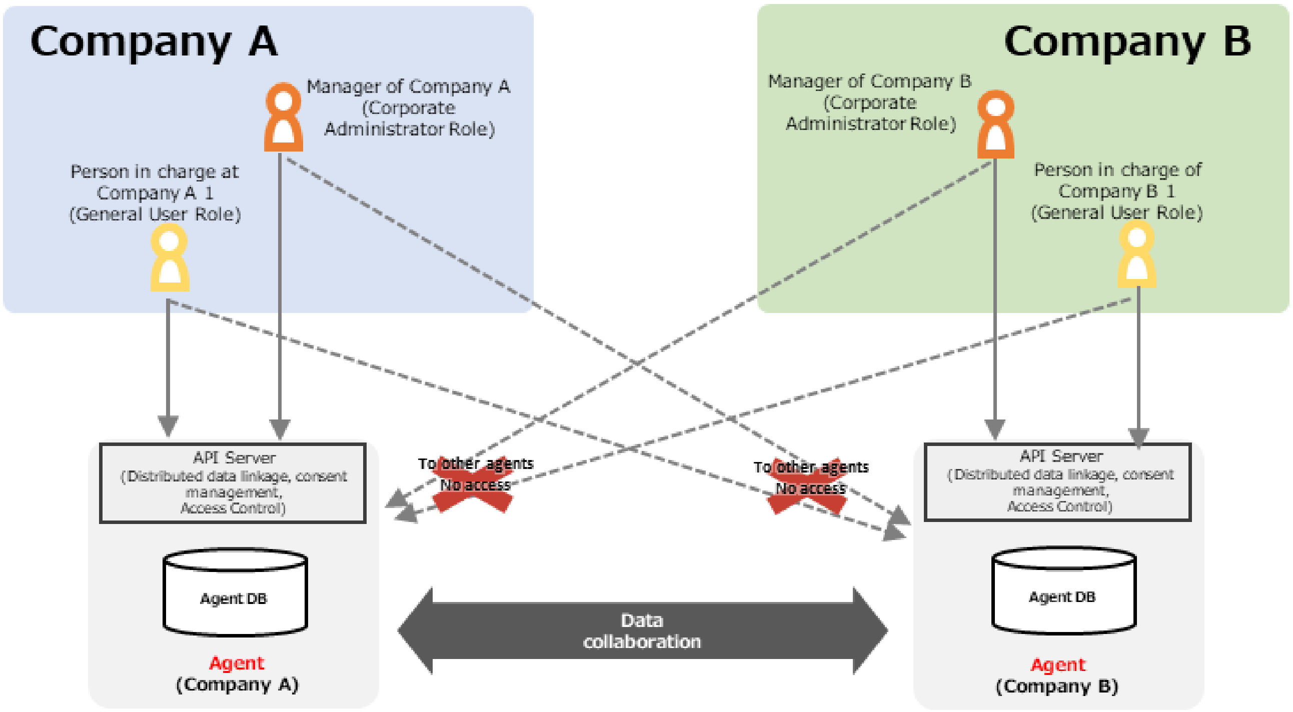
[Terms and concepts to be understood]

- Agents in Fujitsu Data e-TRUST
- Users and Roles in Fujitsu Data e-TRUST


Manager of Company A (Corporate Administrator Role)
Manager of Company B (Corporate Administrator Role)
Manager of Company C (Corporate Administrator Role)
Person in charge of Company A (General User Role)
Person in charge of Company B (General User Role)
Person in charge of Company C (General User Role)
auditor (Reviewer Role)
Data-TRUST applications
Data e-TRUST API
API Gateway
API Server (Distributed data linkage, consent management, Access Control)
Data Collabo ration
Agent DB
Agent (Company A)
Agent (Company B)
Agent (Company C)
Trail and audit function (transaction history)
Data e-TRUST

2.1 Agents in Fujitsu Data e-TRUST

In Fujitsu Data e-TRUST, an agent is a database management unit created for each company or organization. It acts as an intermediary between data manipulation and the transfer of data between corporate organizations with API requests. Database access permissions are controlled on a per-agent basis.


Company A
Manager of Company A (Corporate Administrator Role)
Person in charge at Company A 1 (General User Role)
Company B
Manager of Company B (Corporate Administrator Role)
Person in charge of Company B 1 (General User Role)
API Server (Distributed data linkage, consent management, Access Control)
Agent DB
Agent (Company A)
To other agents No access
Data collaboration
To other agents No access
API Server (Distributed data linkage, consent management, Access Control)
Agent DB
Agent (Company B)

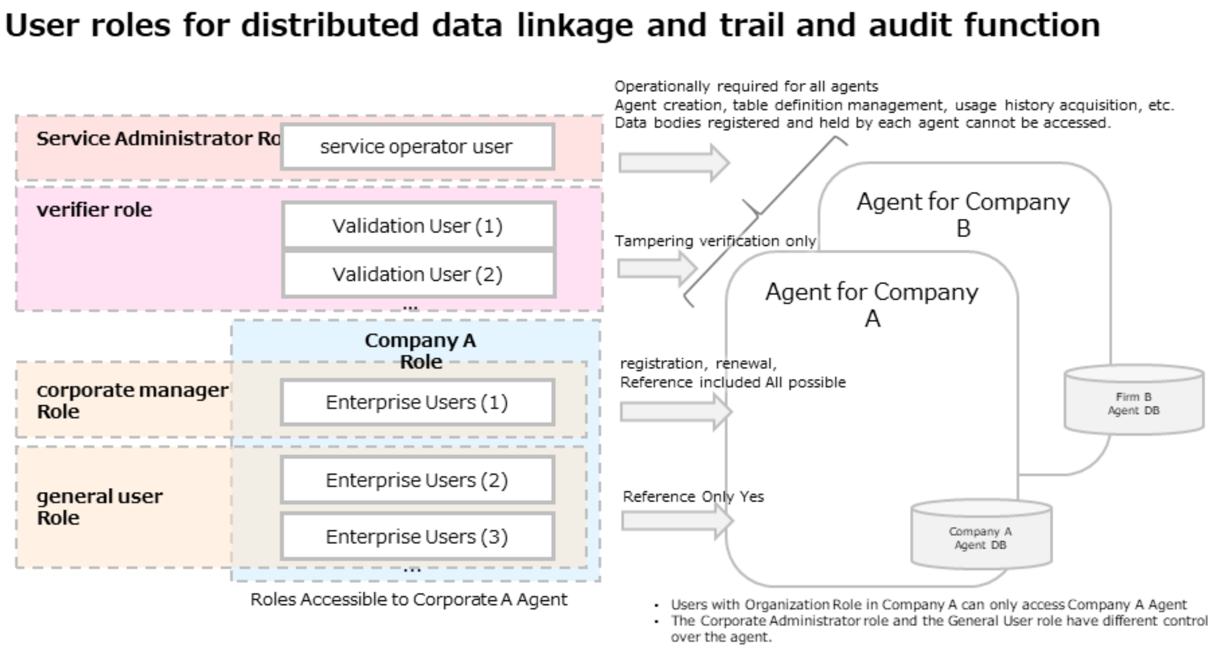
2.2 Roles in Fujitsu Data e-TRUST

Roles in Fujitsu Data e-TRUST are privileges granted according to the user's role.
The granted role determines the API that can be executed, the options that can be specified when the API is executed, and the response content. You can also grant multiple roles to a user.

There are four roles for distributed data linkage, trail and audit function, and three roles for trust seal function.
The roles for the distributed data linkage function, trail and audit function, and the role for the trust seal function are independent of each other.

- Role for distributed data linkage function, trail and audit function
  - Service Administrator role
  - Corporate Administrator role
  - General User Role
  - Verifier Role
- Role for trust seal function
  - Trust Seal Administrator Role
  - Trust Seal User Role for Agents
  - Trust Seal User Role for Users

Role for distributed data linkage function, trail and audit function



User roles for distributed data linkage and trail and audit function

- Users with Organization Role in Company A can only access Company A Agent
- The Corporate Administrator role and the General User role have different control over the agent.

[Service Operator Role]

This role is granted to administrators who operate services using Fujitsu Data e-TRUST.

You can create agents, manage table definitions, and collect usage logs required for service operation.
You cannot access the data bodies that each agent registers and holds.

[Corporate Administrator Role]

This role is granted to administrators of companies and organizations that use services utilizing Fujitsu Data e-TRUST.

You can access data registered and held by agents you manage.

[General User Role]

Roles granted to general users belonging to each enterprise agent.

You can only reference access to data held by the agent to which you belong.

[Verifier Role]

Roles granted to users who use the trail and audit function to perform audit tasks.

Only falsification verification using the trail and audit function is possible.

Table 2.1 Roles for Distributed Data Collaboration and Trail and Audit Functions

Role Type	Specifying parameters during API requests		Notes
	user_role	agent[1-10]_role *	
Service Operator Role	operator	-	
Corporate Administrator Role	user	administrator	specified by a combination of *user_role* and *agent[1-10]_role*
General User Role	user	user	specified by a combination of *user_role* and *agent[1-10]_role*
Verifier Role	verifier	-	

* You can specify 10 types of agent1_role, agent2_role, and ...... agent10_role.

Trust seal role

[Trust Seal Administrator Role]

This role can only view the certificate and trust seal of the agent to which it belongs.

[Trust Seal User Role for Agents]

Roles that can use and create certificates and trust seals on a per-agent basis.

[Trust Seal User Roles for Users]

Roles that can use and create certificates and trust seals on a per-user basis belonging to the agent.

Table 2.2 Roles for Trust Seal Function

Role Type	Specifying parameters during API requests		Notes
	user_role	agent[1-10]_role *	
Trust Seal Administrator Role	user	tseal_administrator	specified by a combination of *user_role* and *agent[1-10]_role*
Trust Seal User Role for Agents	user	tseal_agent	specified by a combination of *user_role* and *agent[1-10]_role*
Trust Seal User Role for Users	user	tseal_user	specified by a combination of *user_role* and *agent[1-10]_role*

* You can specify 10 types of agent1_role, agent2_role, and ...... agent10_role.

When you request an API, the agent to be operated on is specified in the request header, and the role to be operated on is specified in the token to be granted to the request. A user can belong to multiple agents and have roles for each agent.

Therefore, specify the role that corresponds to each agent that belongs to it, with the value *agent1_id* being the agent ID of the Corporate A agent and the value *agent1_role* being the role granted by the Corporate A agent.

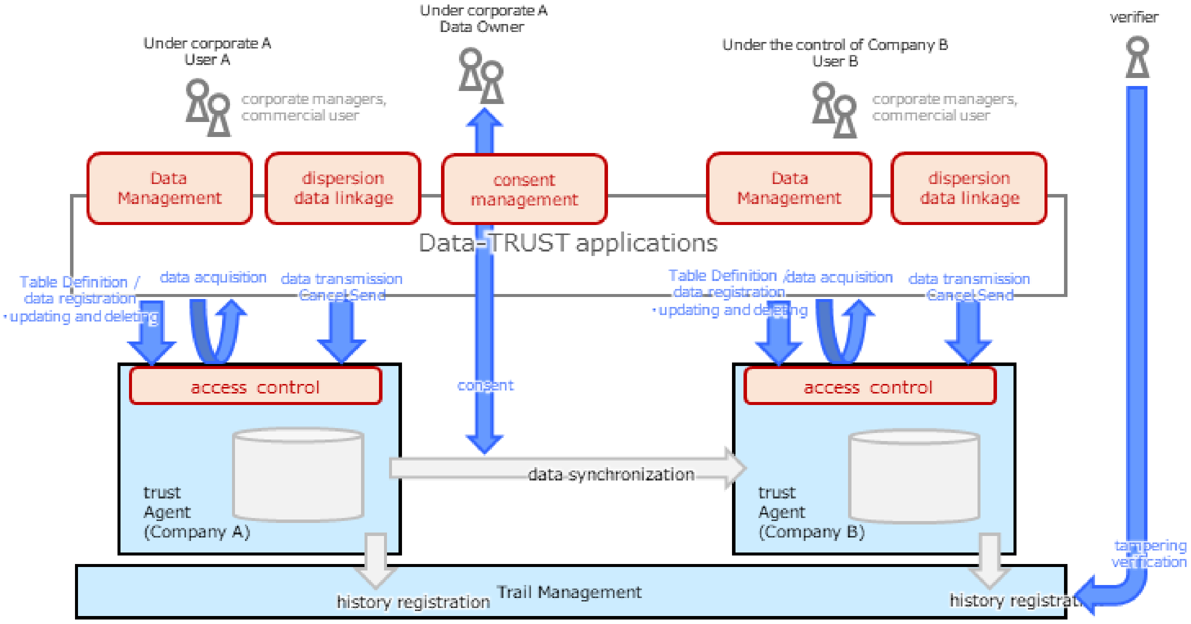
Chapter III Data Distribution Function

Fujitsu Data e-TRUST's distributed data linkage function enables companies and organizations to share and coordinate data among their agents. In addition, by combining the distributed data linkage function with the consent management function, data distribution based on the agreement of the data owner can be realized.

3.1 Prerequisite Knowledge for Using Data Distribution

The relationship between each API required to use Fujitsu Data e-TRUST data distribution is shown.

Data distribution is achieved through APIs for distributed data linkage and APIs for consent management.
The relationship between features related to data distribution and key APIs is shown in the diagram below.
For more information about each API, see the API Reference Manual and the API Reference Manual: separate volumes.



3.2 Data Flow in Fujitsu Data e-TRUST

This section describes the basic flow of data distribution using the distributed data linkage function of Fujitsu Data e-TRUST. In this section, the basic methods of data transmission to other organizations are described in "Preparation Procedures for Data Distribution", "Starting Procedures for Data Distribution", and "Stopping Procedures for Data Distribution".
Procedures for starting and stopping data distribution are described separately.

	No.	Operation	Description
Preparation Procedures	1	Creating and Registering Agents	Create DBs for each agent ID which are issued for each company or organization. This is described in Section 3.3.1. in detail.
	2	Registering table Definitions	Register the table in the created DBs on the agent. This is described in Section 3.3.2. in detail.
Starting Procedures	3	Data Registration	Register data in defined tables. This is described in Section 3.3.3. in detail.

	No.	Operation	Description
Starting Procedures	4	Sending Data between Agents	Send and synchronize data between agents. This is described in Section 3.3.4. in detail.
	5	Data Acquisition	Acquire registered data and data sent to the local agent. This is described in Section 3.3.5. in detail.
Stopping Procedures	6	Stopping Data Synchronization	Stops synchronization of data being sent/synchronized to another agent. This is described in Section 3.3.6. in detail.
	7	Delete Data	Deletes data registered. This is described in Section 3.3.7.

3.3 Basic Data Distribution Operations in Fujitsu Data e-TRUST

3.3.1 How to Create and Register an Agent in Fujitsu Data e-TRUST

To use Fujitsu Data e-TRUST, create and register agents using the agent creation API of the management function.

Create agent API

Agent creation is the first operation to use Fujitsu Data e-TRUST.
Create an agent and an agent-specific database associated with the specified agent ID.
This allows you to manage data for each agent.
The agent creation API can only be run by the Service Operator and Corporate Administrator roles.
Note that single database is created per agent.

3.3.2 Table Definition Registration in Fujitsu Data e-TRUST

In preparation for data registration to be handled by Fujitsu Data e-TRUST, define the table with table definition API of the distributed data linkage function.

Table definition

Define a table definition to manipulate data in the database for each agent,.

Points

About Data Owner Types

In addition to common data types such as string types, Date-TRUST can define its own data owner types for the types of columns that can be specified for a table. This allows you to specify the data owner that owns the record.
When records with columns of data owner type are sent to data transmission and synchronization, you can require consent from the owner.
For more information, see Data Submission API, Agreement API.

The table definition has three API endpoints:

Manage table definition (create) API

Creates a new table with the specified table configuration.
Use to create a new table.

Manage table definition (update) API

Adds and removes columns, indexes, and references to the specified table.
Use this to update table definitions that have already been created.

Manage table definition (delete) API

Deletes the specified table.

When you delete a table, the data stored in the table is also deleted.

3.3.3 How to Register Data Handled by Fujitsu Data e-TRUST

Data handled by Fujitsu Data e-TRUST is registered in the agent's database.

Data Registration

Register data in the defined table.
The data is registered in the column configuration set in the table definition.

There are two types of APIs used for data registration. The individual data registration and update API registers data in JSON format, and the batch data registration and update API registers data registered in a file.

Register data (single record) API

Registers the data specified in JSON format to the table.

Upload data (multiple records) API

Register and update data in CSV format or in ZIP format, which is a compressed CSV file.
Use this function to register a large amount of data at one time, such as when initially registering data.

3.3.4 How to send data handled by Fujitsu Data e-TRUST

You can send (synchronize) your agent's data registered in Fujitsu Data e-TRUST to other agents.
Data transmission is processed with combination of the three APIs of the distributed data linkage function and one API of the consent management function.

APIs related to data submission operations

Send data API

Send and synchronize the data to be linked to the specified agent.

Request data transmission API

Requests other agents to send the specified data.

Respond to data transmission request API

Reply to the requesting agent whether the data requested by the data submission request API can be sent.

Respond to consent request API

The data owner receives the client notification of the consent request notification and replies to the notifying agent whether the data can be sent.

Reference

About Client Notifications

For information about client notifications, see the API Reference Manual and the API Reference Manual: separate volumes.

Data transmission pattern between agents

There are three main patterns of data transmission processing between agents.

- Sending data that does not require consent

- Sending data requiring consent
- Send data based on requests from other agents

The following shows the data transmission pattern between agents when Company A sends data to Company B.

Characters

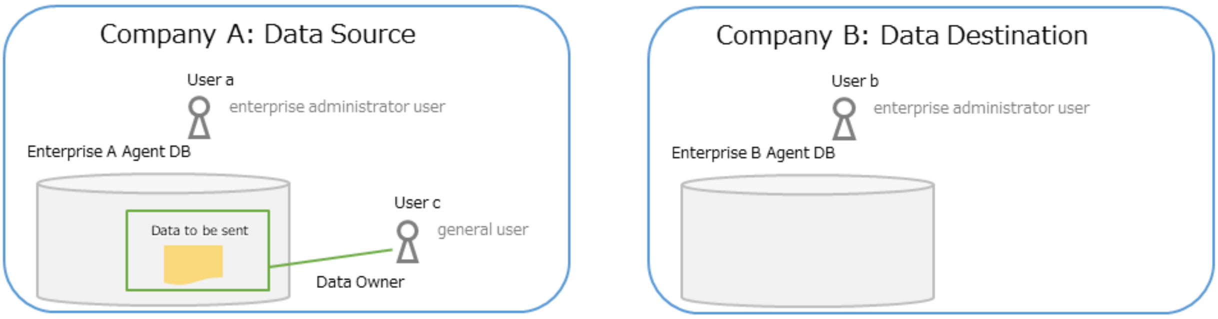
- Source Company: Company_A

  - User_a: Corporate administrator user under Company_A

  - User_c: General user in Company_A, data owner of data to be sent

- Destination Company: Company_B

  - User_b: Corporate administrator user under Company_B



Caution

About Client Notification Recipients

Notifications are posted to the payload URL of the client notification destination.
Therefore, when client notification is used, the application must receive the notification and process the notification to each user.

Although "User ● ● is notified of △ △ by ○ ○'s client notification." is described in the following descriptions for convenience, the Fujitsu Data e-TRUST function does not directly notify the user. Users should be notified by the application.

For more information about client notifications, see the Client Notification Configuration API section of the API Reference Manual and the API Reference Manual: separate volumes Chapter 6, Section 3.

Sending data that does not require consent

When sending data to another agent, if the consent of the data owner user c is not required, use the data sending API.
The flow of this process is as follows.

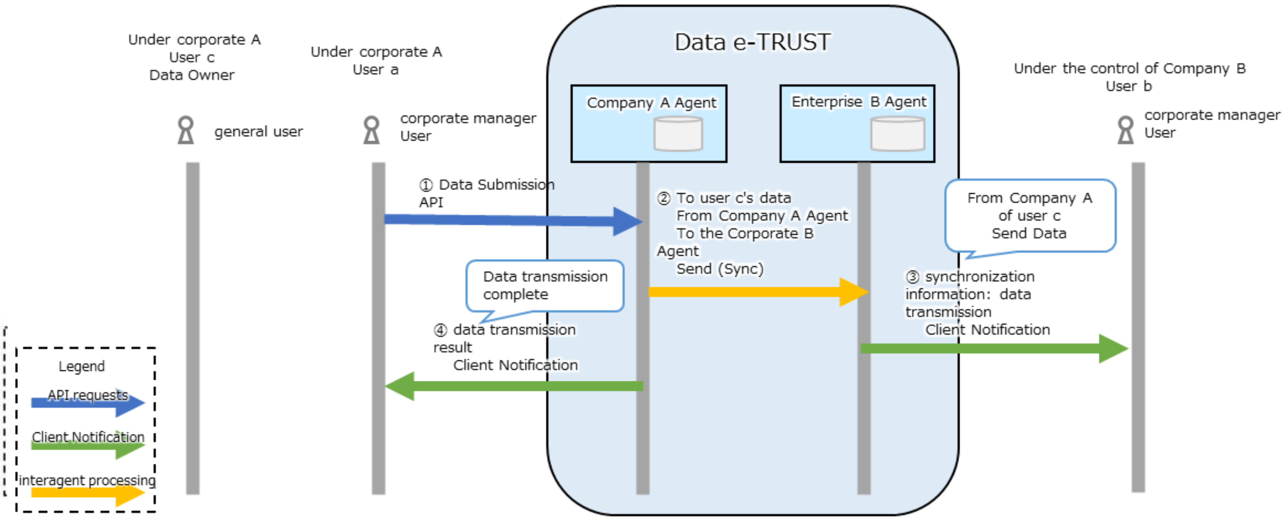
1.User_a in Company_A executes the data transmission API.

2.Company_A agent sends data to Company_B agent.

User_b in the Company_B can use the data transmitted by the User_a.

3.By a client notification of "synchronization information: data transmission", the result of data transmission processing is notified to User_b of a transmission destination.

4.The result of the data transmission processing is notified to the sending User_a by the client notification of "synchronization information: data transmission result".



Sending data requiring consent

When sending data to another agent, if the consent of the data owner User_c is required, two APIs, the data sending API and the consent response API, are used.

The flow of this process is as follows.

1.User_a in Company_A executes the data transmission API.

2.By a client notification of "consent request", a consent request for data transmission is notified to User_c of a data owner under Company_A.

3.User_c executes the consent response API and sends an agreement to send to Company_A agent.

4.The “Agree Response” client notification informs User_a that User_c has consented to the data transmission.

5.Company_A agent sends data to Company_B agent according to the consent response.

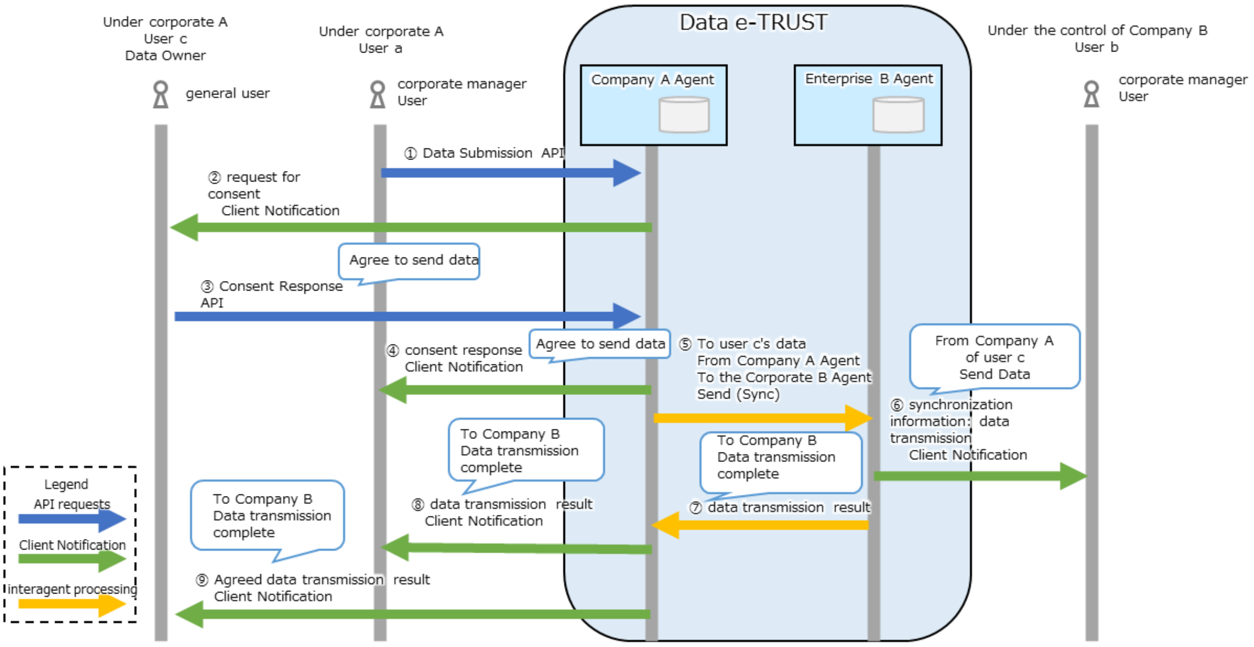
User_b in Company_B can use the data transmitted by User_a.

6.By a client notification of "synchronization information: data transmission", the result of data transmission processing is notified to User_b of a transmission destination.

7.Interagent processing notifies Company_A agent of the completion of data transmission from Company_B agent.

8.The result of the data transmission processing is notified to the sender User_a by the client notification of the "data transmission result".

9.The result of the data transmission processing is notified to the User_c of the data owner by the client notification of the "agreed data transmission result".



Send data based on requests from other agents

When sending data at the request of User_b in Company_B of another agent, three APIs, data submission request API, response data submission request API, and data submission API, are used.
The flow of this process is as follows.

1.User_b in Company_B executes the data submission request API.

2.In inter-agent processing, Company_B agent notifies Company_A agent of data transmission request.

3.A data transmission request is notified to User_a under Company_A by a client notification of "data transmission request".

4.User_a executes the data transmission request response API and notifies Company_A agent that he/she agrees to the data transmission.

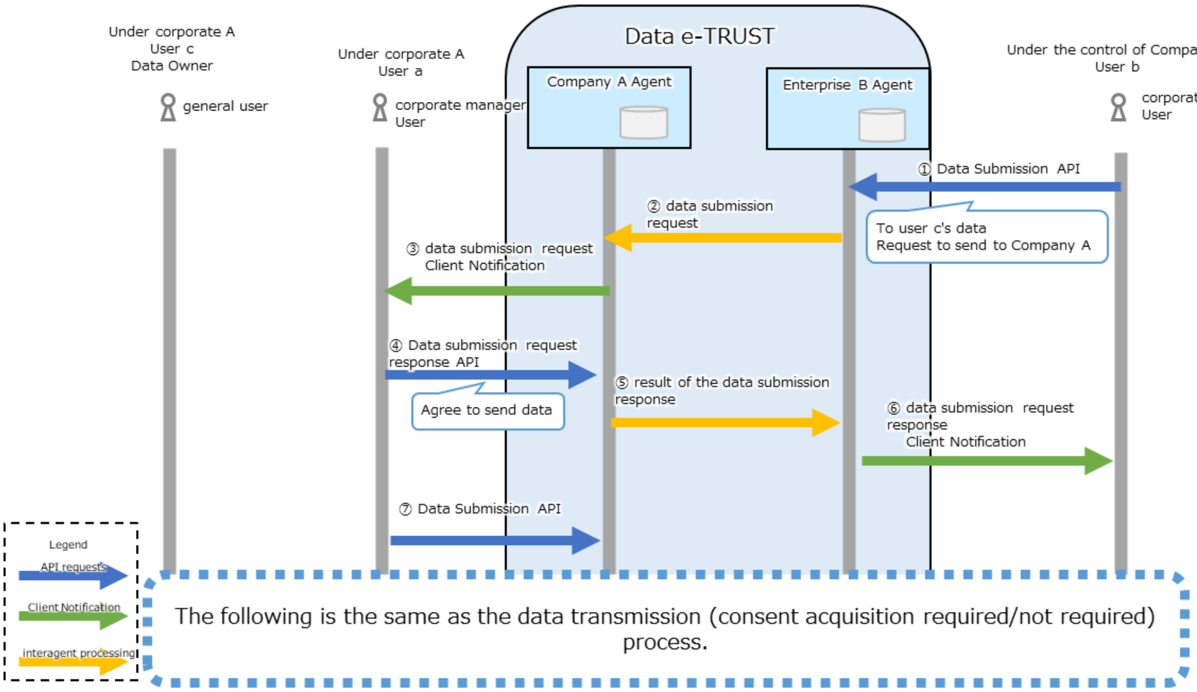
The data is not transmitted to the Company_B only by executing the data transmission request response API and answering consent to the data transmission.

5.In inter-agent processing, Company_A agent notifies Company_B agent of the result of data transmission request response.

6.The client notification of the "data transmission request response" informs the user b that the consent of data transmission has been obtained.

7.User_a runs data submission API

Thereafter, in order to send the data to the Company_B agent, it is necessary to perform either of the processes of “sending data for which consent acquisition is not required” and “sending data for which consent acquisition is required ”.



3.3.5 Acquiring Data Handled by Fujitsu Data e-TRUST

You can use the data acquisition API to acquire data registered on the local agent.

Get data API

You can retrieve data from a table in the local agent or an agent to which you have access by using specified conditions.

Use this function to obtain data registered on the local agent or data sent or synchronized to the local agent.

3.3.6 Stopping Data Synchronization Handled by Fujitsu Data e-TRUST

Data submission cancel API allows you to stop the synchronization of data sent and synchronized to other agents.

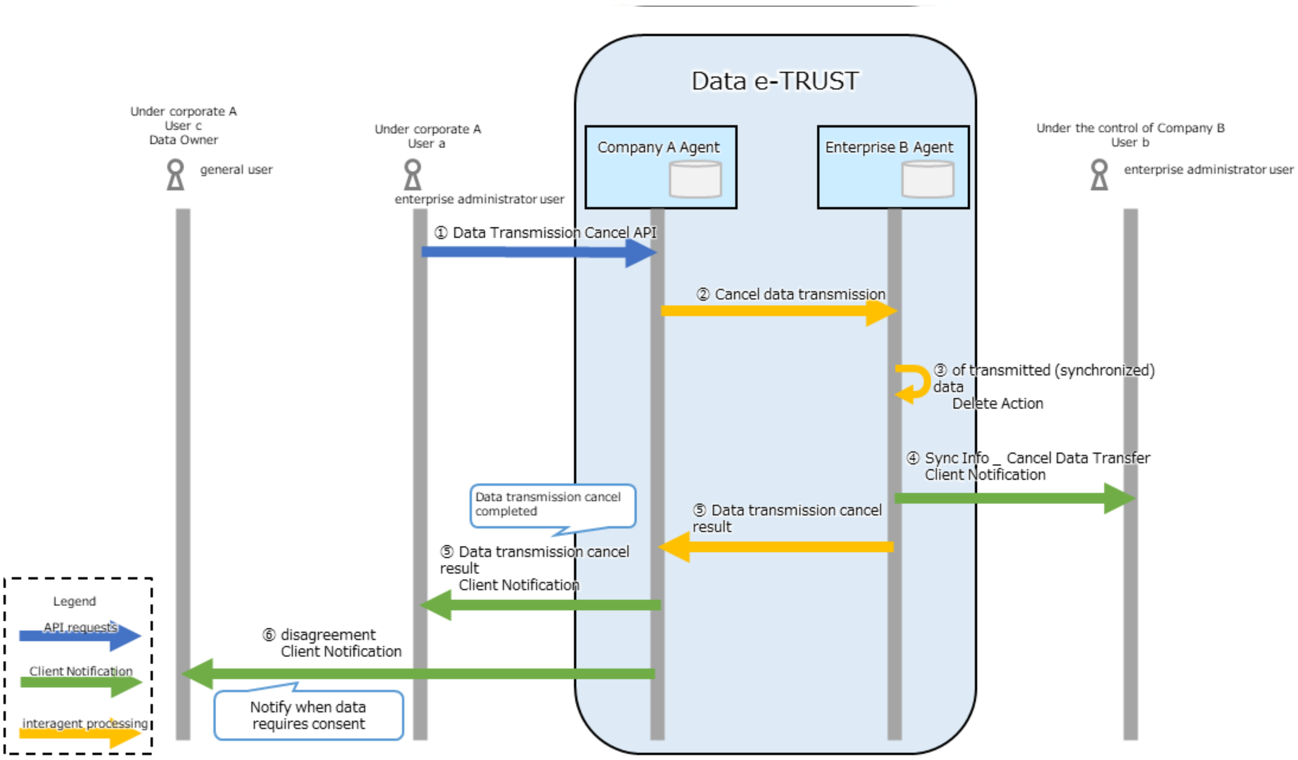
Cancel data transmission API

The synchronization of the specified data is stopped and the data on the data destination agent is deleted.

Stopping Data Synchronization

User_a in the source Company_A executes the data submission cancel API to delete data at the destination agent Company_B.

The result of stopping data transmission synchronization is notified by client notification to the data source Company_A agent, the data destination Company_B agent, and the data owner User_c.


Data e-TRUST
Under corporate A User c Data Owner
general user
Under corporate A User a
enterprise administrator user
Company A Agent
Enterprise B Agent
Under the control of Company B User b
enterprise administrator user
① Data Transmission Cancel API
② Cancel data transmission
③ of transmitted (synchronized) data Delete Action
④ Sync Info _ Cancel Data Transfer Client Notification
⑤ Data transmission cancel result
Data transmission cancel completed
⑤ Data transmission cancel result Client Notification
⑥ disagreement Client Notification
Notify when data requires consent
Legend
API requests
Client Notification
interagent processing

3.3.7 Deleting Data Handled by Fujitsu Data e-TRUST

To delete data registered in Fujitsu Data e-TRUST, use the data deletion API.

Delete data API

The data deletion API allows you to delete any record registered with the local agent under specified conditions.

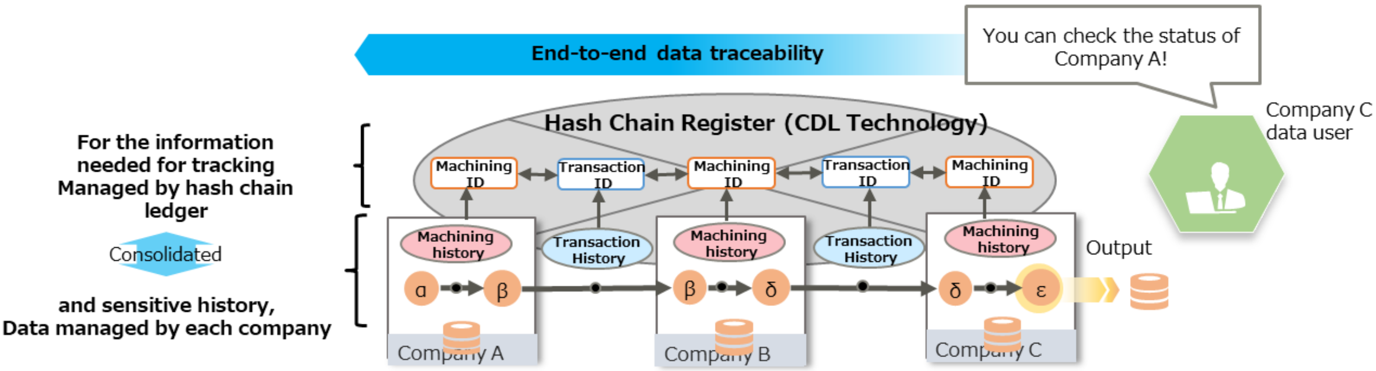
If the data to be deleted has already been sent and synchronized to another agent using the data transmission API, all data registered in the destination agent will also be deleted.

Use this function to delete unnecessary data in units of records.

Chapter IV Trail and Audit Function

The trail and audit function of Fujitsu Data e-TRUST manages a series of histories that occur during data transactions and distribution in a format that is unalterable, mutually verifiable, and publicly and privately controllable.

This chapter illustrates the knowledge required to use the trail and audit function and the basic flow of use.



4.1 Prerequisite knowledge for using trail and audit functions

This section describes the terminology, data model, and data structure you need to know to use the trail and audit features of Fujitsu Data e-TRUST.

4.1.1 Terms required for trail and audit functions

Definitions of the terminology required to use the trail and audit capabilities of the CDL.

Trail/Trail Information

Information that is managed by the trail and audit function to represent individual occurrences, matters, processes, and events.
An example of trail is shown below.

- Transaction information and historical information representing individual events such as "sent" and "received" between companies and organizations
- Individual events and processing information in the supply chain and traceability of goods
- Data processing and transmission information in data utilization

Lineage

It is a group of histories that are connected in sequence.

A set of data that is represented by series of individual histories indicating when and what happened.

Global Data

Information that constitutes a trail and is unconditionally published and shared with all organizations.

Local Data

Information that constitutes a trail is not unconditionally disclosed to all organizations but is disclosed only to specific organizations and users with access control.

4.1.2 Data model and lineage structure for the trail and audit function

The trail and audit function has its own data model to map and record various real-world supply chains and traceability.

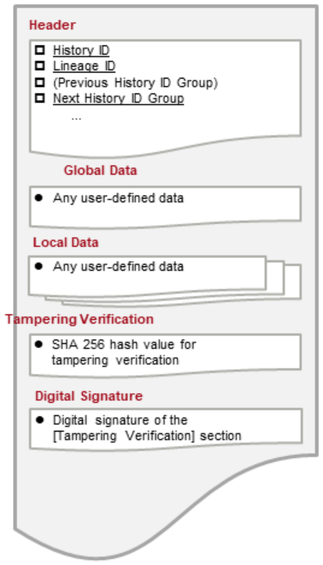
The data model consists of the smallest unit of data managed by the CDL, Trail Information, and Lineage, which links the trail information back and forth.

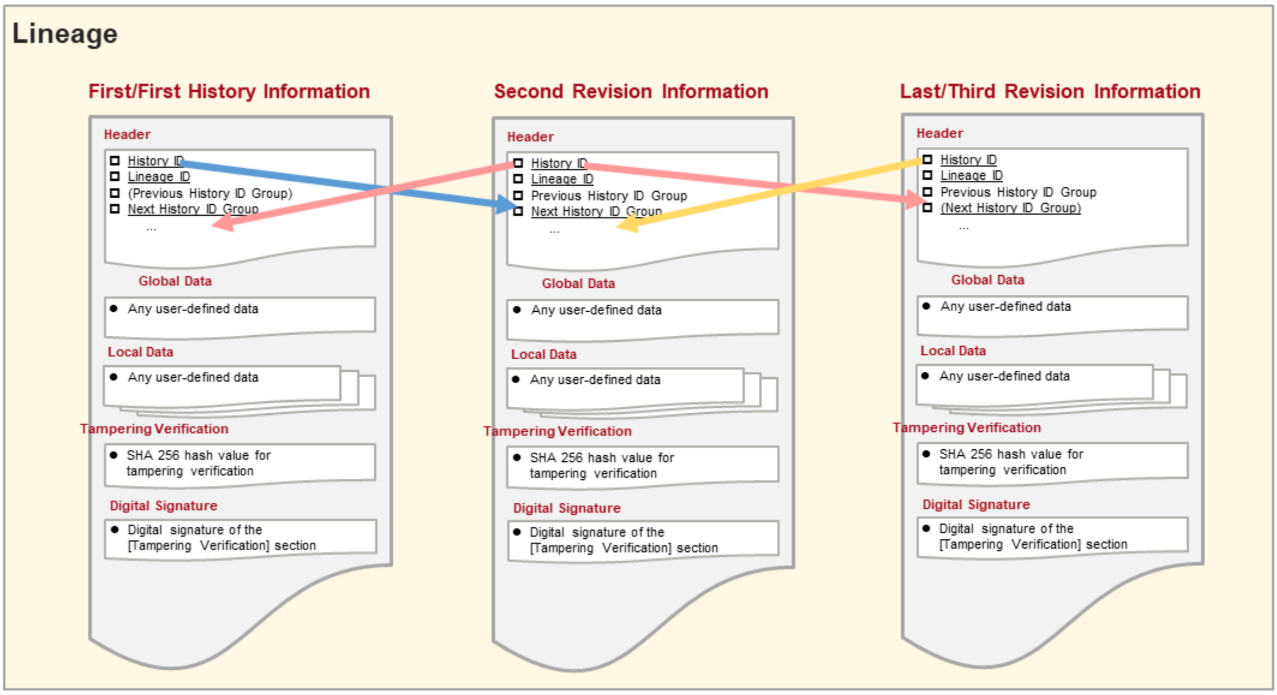
Lineage consists of trail data that is a concatenation of the "next trail ID group" and "previous trail ID group" in the [Header] section.

Lineage data structures allow you to verify that data has not been tampered with after it has been retrieved from the CDL. If there is a branch in the Lineage trail, there are multiple histories at the Lineage end, but each trail at that end is digitally signed.

When the trail and audit function's tampering verification API is executed, the hash value of each item is calculated and collated so that data tampering verification can be performed.

Historical information



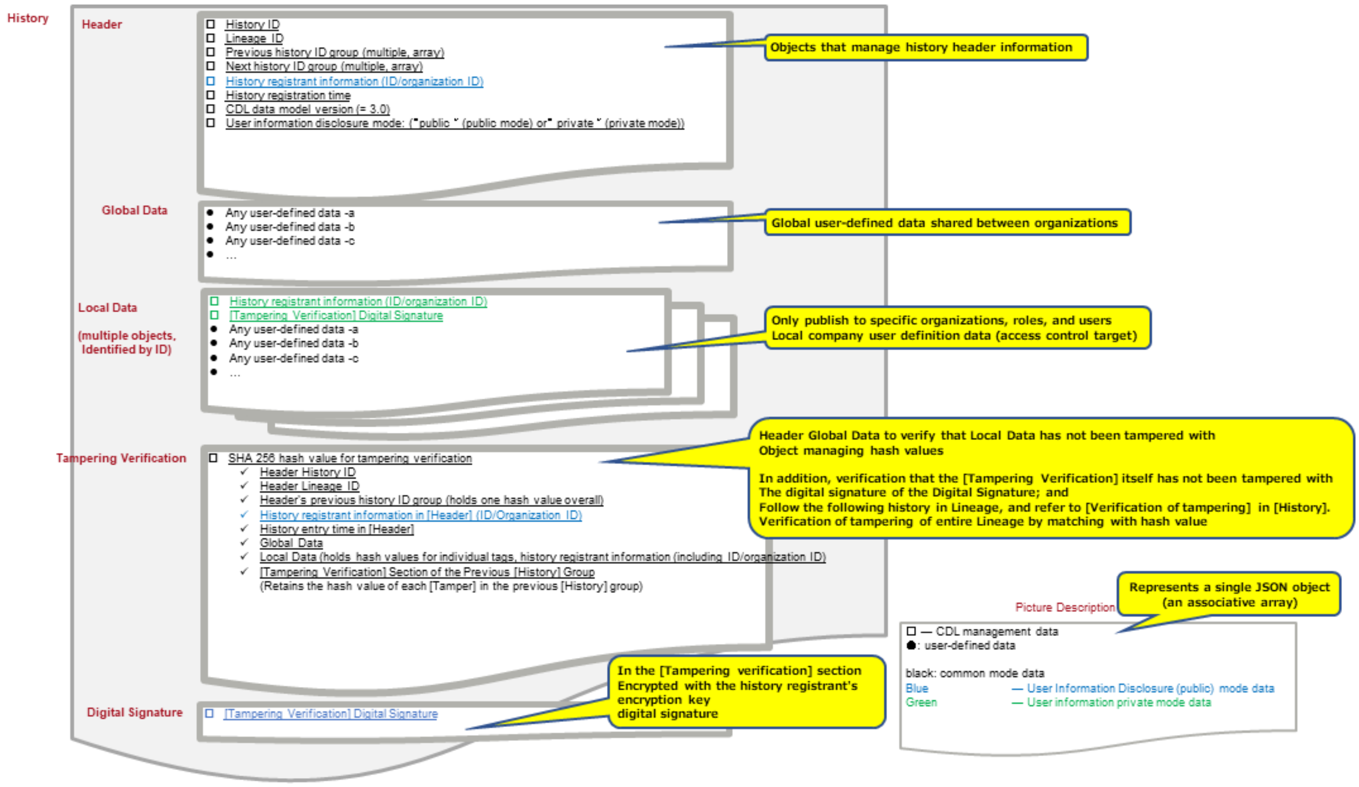

Lineage
First/First History Information
Header
History ID
Lineage ID
(Previous History ID Group)
Next History ID Group
...
Global Data
Any user-defined data
Local Data
Any user-defined data
Tampering Verification
SHA 256 hash value for tampering verification
Digital Signature
Digital signature of the [Tampering Verification] section
Second Revision Information
Header
History ID
Lineage ID
Previous History ID Group
Next History ID Group
...
Global Data
Any user-defined data
Local Data
Any user-defined data
Tampering Verification
SHA 256 hash value for tampering verification
Digital Signature
Digital signature of the [Tampering Verification] section
Last/Third Revision Information
Header
History ID
Lineage ID
Previous History ID Group
(Next History ID Group)
...
Global Data
Any user-defined data
Local Data
Any user-defined data
Tampering Verification
SHA 256 hash value for tampering verification
Digital Signature
Digital signature of the [Tampering Verification] section

4.1.3 Data Structure of Trail Information Configuring Lineage of CDL

The individual trail information that makes up the CDL lineage consists of five sections:

- header
- global data
- local data
- falsification verification
- digital signature

Figure 4.1 Data Structure of Trail Information


History
Header
□ History ID
□ Lineage ID
□ Previous history ID group (multiple, array)
□ Next history ID group (multiple, array)
□ History registrant information (ID/organization ID)
□ History registration time
□ CDL data model version (= 3.0)
□ User information disclosure mode: ("public " (public mode) or" private " (private mode))
Objects that manage history header information
Global Data
● Any user-defined data -a
● Any user-defined data -b
● Any user-defined data -c
● ...
Global user-defined data shared between organizations
Local Data
(multiple objects, Identified by ID)
□ History registrant information (ID/organization ID)
□ [Tampering Verification] Digital Signature
● Any user-defined data -a
● Any user-defined data -b
● Any user-defined data -c
● ...
Only publish to specific organizations, roles, and users
Local company user definition data (access control target)
Tampering Verification
□ SHA 256 hash value for tampering verification
✓ Header History ID
✓ Header Lineage ID
✓ Header's previous history ID group (holds one hash value overall)
✓ History registrant information in [Header] (ID/Organization ID)
✓ History entry time in [Header]
✓ Global Data
✓ Local Data (holds hash values for individual tags, history registrant information (including ID/organization ID)
✓ [Tampering Verification] Section of the Previous [History] Group
(Retains the hash value of each [Tamper] in the previous [History] group)
Header Global Data to verify that Local Data has not been tampered with
Object managing hash values
In addition, verification that the [Tampering Verification] itself has not been tampered with
The digital signature of the Digital Signature; and
Follow the following history in Lineage, and refer to [Verification of tampering] in [History].
Verification of tampering of entire Lineage by matching with hash value
Digital Signature
□ [Tampering Verification] Digital Signature
In the [Tampering verification] section
Encrypted with the history registrant's
encryption key
digital signature
Picture Description
Represents a single JSON object
(an associative array)
□ — CDL management data
●: user-defined data
black: common mode data
Blue — User Information Disclosure (public) mode data
Green — User information private mode data

[Header] Section

The [Header] section consists of two parts: trail index and lineage information.

Trail index

The trail index consists of four parts:

- Trail ID
- Registrant ID
- Registrant Organization ID
- Registration time

Lineage information

Manage trail context.
Lineage information consists of two parts:

- Previous Trail ID Group
- Next trail ID group

[Global Data] Section

Represents part of trail information that is to be disclosed to other organizations.

[Local Data] Section

It represents information that is only published to authorized organizations and access controlled to other organizations. You can register multiple data items. Individual data is identified by an ID (local data ID).

[Falsification Verification] Section

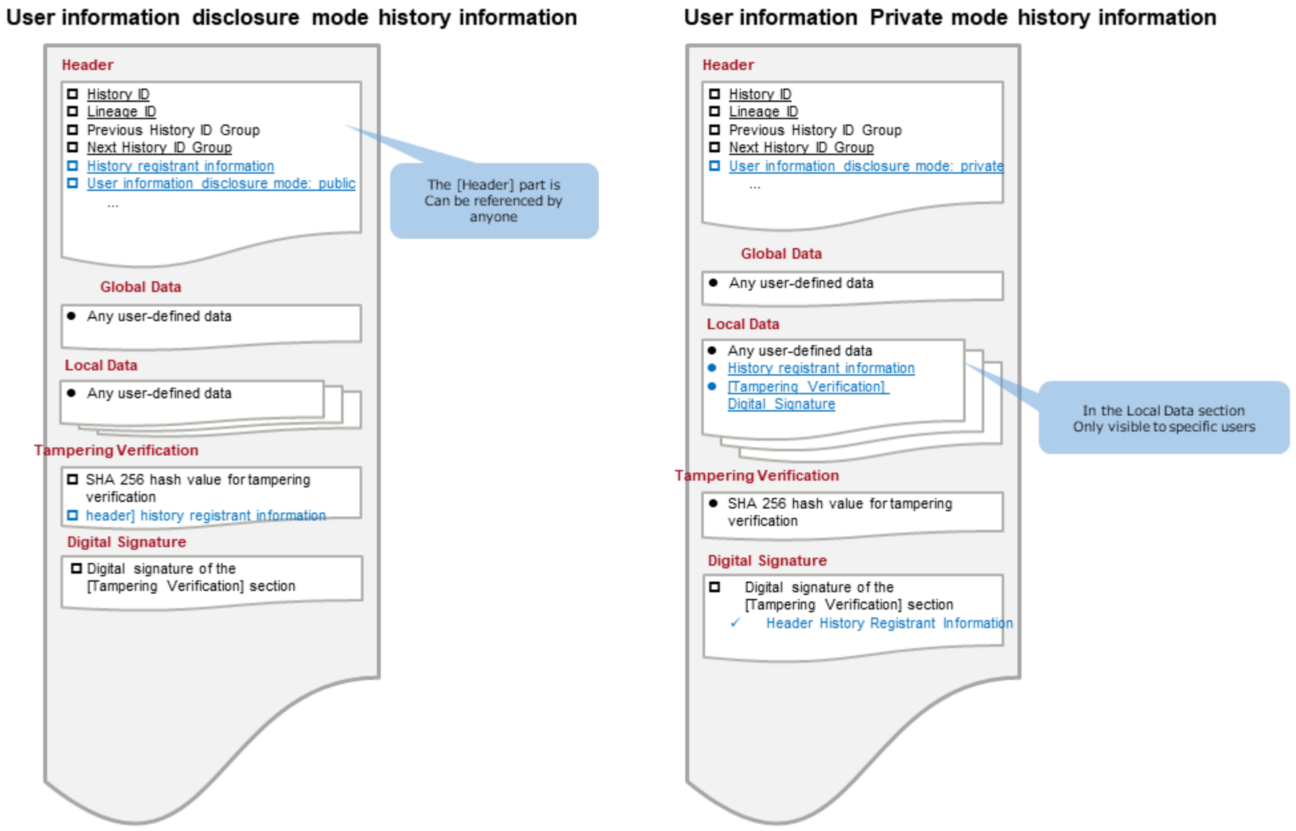
This information is used to verify that trail information has not been tampered with even after the trail information group has been extracted from the CDL as a lineage. Stores the SHA 256 hash value of the [Header], [Global Data], and [Local Data] sections.

[Digital Signature] Section

Stores a digital signature that encrypts the SHA 256 hash value of the [Falsification Verification] section with the private key of the trail registrant to enable the [Falsification Verification] section to verify that it has not been tampered with (from someone other than the trail registrant).

4.1.4 "User Information Disclosure Mode" and "User Information Non-Disclosure Mode"

There are two modes for the trail and audit function: user information disclosure mode and user information non-disclosure mode.
Each mode can be selected when building the environment.


User information disclosure mode history information
Header
History ID
Lineage ID
Previous History ID Group
Next History ID Group
History registrant information
User information disclosure mode: public
...
The [Header] part is Can be referenced by anyone
Global Data
Any user-defined data
Local Data
Any user-defined data
Tampering Verification
SHA 256 hash value for tampering verification
header] history registrant information
Digital Signature
Digital signature of the [Tampering Verification] section
User information Private mode history information
Header
History ID
Lineage ID
Previous History ID Group
Next History ID Group
User information disclosure mode: private
...
Global Data
Any user-defined data
Local Data
Any user-defined data
History registrant information
[Tampering Verification]
Digital Signature
In the Local Data section Only visible to specific users
Tampering Verification
SHA 256 hash value for tampering verification
Digital Signature
Digital signature of the [Tampering Verification] section
Header History Registrant Information

User information disclosure mode

Select for operations that emphasize information openness and transparency and share basic information among organizations (agents).

User information (Registrant ID, Registrant Organization ID) and the trail registrant's digital signature are stored in the Header section and are published to all organizations.

User information non-disclosure mode

Select this option for operations in which user information is confidential except between organizations (agents) that directly deal with each other, and data transaction information between organizations is not shown.
In user information non-disclosure mode, you can still publish user information to organizations that do not directly transact by setting a reference policy after you register.

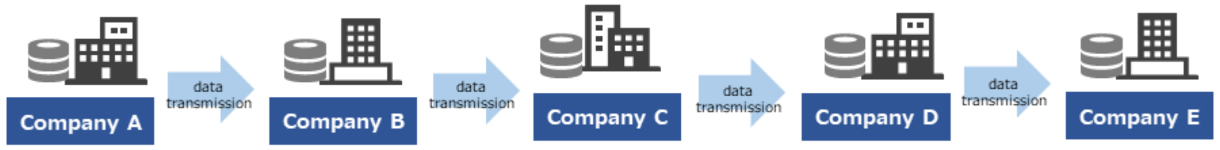
In the following example, data is sent from Company_A to Company_E.

Company_C only deals directly with Company_B, the data source, and Company_D, the data destination.

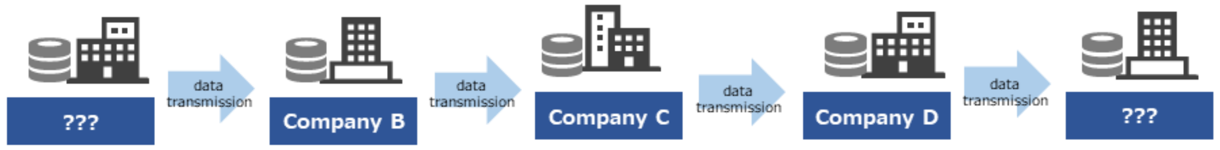
In the user information disclosure mode, Company_C can check all related business partners from Company_A to Company_E from the trail information.

In the user information non-disclosure mode, Company_C can check Company_B and Company_D, but cannot check Company_A and Company_E.

In the user information disclosure mode: You can check the person whom Company C does not deal with directly from the history information.



User information non-disclosure mode: I don't know who company C doesn't deal with directly.



4.1.5 JSON format of the data model for the trail and audit function

Use the JSON format for trail data when using the trail and audit feature. There are two types of JSON formats:

- Historical Data JSON Format
- JSON Format for Trail Registration

For more information about each JSON format, see the appendix.

4.2 Overview of each operation of the trail and audit function

The following is an overview of each of the operations available for the Fujitsu Data e-TRUST trail and audit function.

Operation	Description
Trail registration	Registers the specified trail information and manages it as a lineage.
Lineage acquisition	You can obtain the lineages registered by "Trail registration".
Trail search	You can search the trail information registered by "Trail registration" by the specified conditions.
Local Data deletion	Of the trail information registered by "Trail registration", you can delete the information contained in the local data part of the specified trail information.
Reference Policy Settings	For each trail information, you can manage the permissions required to view the local data portion.
Falsification verification	You can detect tampering and verify the registrant for the entire specified lineage or individual trail information.

To acquire Lineage, search trail, delete local data, set reference policy, or verify tampering, the Lineage to be subject to each operation must be registered by trail registration in advance.

4.3 How to use the trail and audit functions

4.3.1 Trail Registration

To use trail and audit functions with Fujitsu Data e-TRUST, trail information is registered with the trail registration API and managed as a lineage.

Register trail API

Registers and manages the specified log information as a lineage.

4.3.2 Lineage Acquisition

The Lineage Acquisition API is used to acquire the lineage registered by the trail registration API.

Get Lineage API

Acquires the lineage to which the specified trail ID belongs.

4.3.3 Trail Search

API for searching trail information registered by the trail registration API.

There are five API endpoints, depending on what you are searching for.

Search trail (by header) API

You can specify a search method for the header part of the trail and perform a trail search.

Search trail (by global data) API

You can search the global data part of the trail by specifying the search method.

Search trail (by local data across organizations) API

You can perform a trail search across agents (organizations) by specifying a search method for the local data portion of the trail.

Search trail (by local data and organization) API

You can specify a search method for the local data portion of the trail and search the trail only within the specified agent (organization).

Search trail (by verification) API

You can search the trail by specifying the search method for the falsification verification part of the trail.

4.3.4 Local Data Deletion

The local data deletion API is an API for deleting the information contained in the local data part of the specified trail information among the trail information registered by the trail registration API.

Delete local data API

Deletes the local data contained in the trail information for the specified trail ID or local data ID.

At this time, the hash value of the local data added to the falsification verification unit at the time of trail registration is not deleted.

4.3.5 Reference Policy Settings

The reference policy configuration API sets the permissions required to reference the local data part for each trail information.

The reference policy configuration API has three endpoints:

Manage reference policy (create) API

You can set a reference policy for a specified local data ID by specifying the organization (agent) name, role, or user.

If more than one is specified, an error occurs.

In addition, by using the reference policy setting for the trail registrant information stored in the local data part in the user information non-disclosure mode, the trail registrant information can be disclosed even in organizations (agents) that do not have direct data transactions.

Manage reference policy (delete) API

You can delete a reference policy by organization (agent) name, role, or user for a specified local data ID.

Manage reference policy (get list) API

Gets the list of reference policies set for the specified local data ID.

4.3.6 Falsification Verification

By using the falsification verification API, you can detect tampering with the lineage registered by the trail audit function and verify the registrant.

Verify trail API

By using the falsification verification section for the entire specified lineage or individual trail information, tampering can be detected, and the registrant can be verified.

Chapter V Trust Seal Function

The Fujitsu Data e-TRUST trust seal feature utilizes a trust seal that can prove that the data issuer or the data itself has not been tampered with.

Trust seals are created from certificates created by certification authorities and data to be certified.

By trading data with this trust seal, you ensure not only the authenticity of the data body, but also the correct existence of the organization or user who issued the data.

5.1 Prerequisite Knowledge for Using Trust Seal Function

The following roles appear when using the Fujitsu Data e-TRUST trust seal function.

Role of Trust Seal Function

There are four roles when using the trust function with Fujitsu Data e-TRUST: issuer, holder, creator, and verifier.

issuer

the issuer is the creator of the certificate.
As a user (person) or agent (organization), create a certificate to prove the authenticity of the holder.

holder

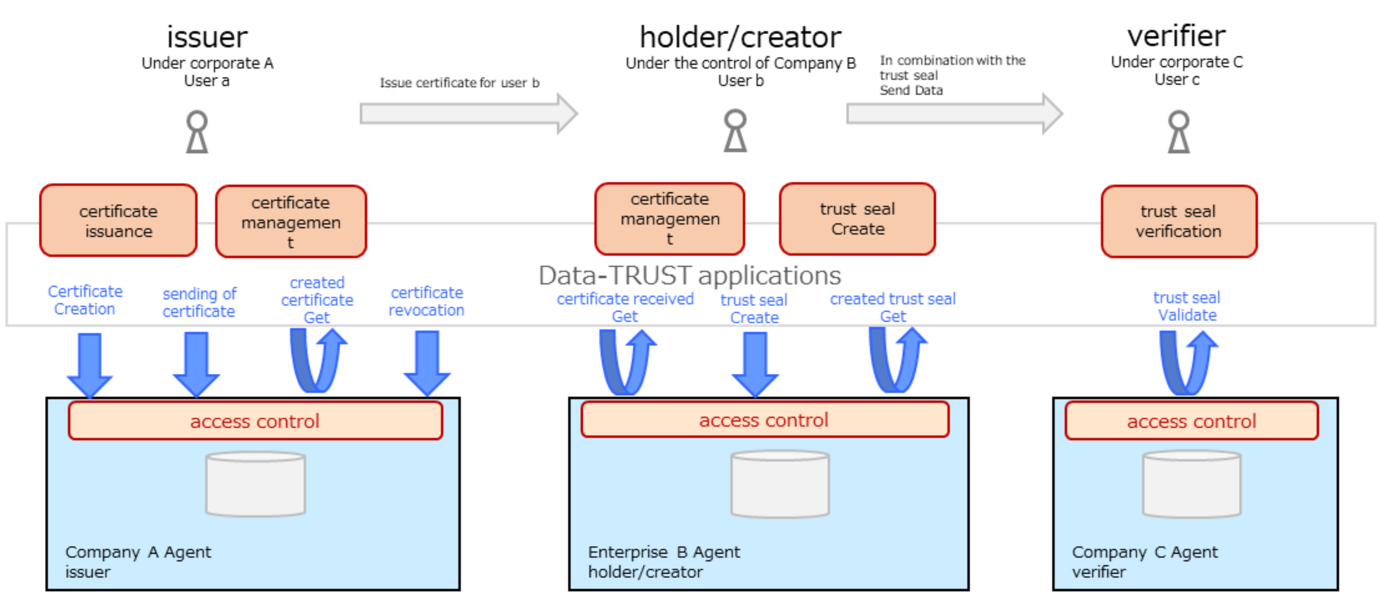
The holder is the recipient of the certificate created by the issuer.
Receive the certificate as a user (person) or as an agent (organization). The certificate is used to create a trust seal by holder itself as a creator.

creator

The creator is the creator of the trust seal.
As a user (individual) or agent (organization), create a trust seal using a certificate. The created trust seal is transmitted with the data.

verifier

The verifier is the verifier of the trust seal.
As a user (individual) or an agent (organization), verifier verifies target data whether the creator of the trust seal is correct and also verifies that the data body has not been tampered with.



The relationship between the role for the trust seal function, the role when using the trust seal, and the operations that can be performed is as follows.

Role	Operation		Trust Seal Administrator Role	Trust Seal User Role for Users	Trust Seal User Role for Agents
issuer	Certificate issuance	Publish as Individual • Example: Personally guarantee specific individual’s capability	×	○	×
		Publish as Organization • Example: School transcripts, etc.	×	×	○
holder	Browse certificates	Browse personal certificate	○	○ only the certificate of your own	×
		Browse certificate of the organization	○	○	○
creator	Trust seals creation using certificates	Using a personal certificate	×	○ only the certificate of your own	×
		Using the certificate of the organization	×	×	○
verifier	Trust seal verification	Seals with individual as verifier	○	○ only the seal of your own	×
		Seals with organization as verifier	○	×	○

5.1.1 Creating Certificates

The certificate creation API creates the certificate needed to create the trust seal.

Create credential API

The issuer executes the certificate creation API to create a certificate for the specified holder.

Certificates created by the certificate creation API do not have browse permissions to holder at the time of creation, so you must grant browse permissions separately.

The certificate creation certificate API is executed by the issuer.

5.1.2 Granting Permission to Reference Certificates

The Certificate submission API allows holders to have browse rights to certificates created with the Certificate Creation API.

Send credential API

The holder of the certified person does not have permission to view the certificate created by the certificate creation API. Therefore, you grant reference privileges through the Certificate Submission API.

The certificate submission API is executed by the issuer.

5.1.3 Managing Certificates

There are three APIs for managing certificates you create:

Revoke credential API

Revoke a certificate when it is no longer needed.

Get credential API

You can obtain a list of created certificates under specified conditions.

Get received credential API

You can obtain a list of certificates received from the issuer through the Certificate submission API under specified conditions.

Holder executes the received certificate acquisition API.

5.1.4 Creating Trust Seals

The trust seal creation API creates a trust seal to be sent with the data that you want to prove authenticity.

Create trust seal API

The creator creates a trust seal using the certificate received by the Certificate Delivery API.

The created trust seal is sent to the verifier with data that you want to prove authenticity.

The trust seal creation API is executed by the creator.

5.1.5 Verifying Trust Seals

The trust seal verification API performs trust seal verification.

Verify trust seal API

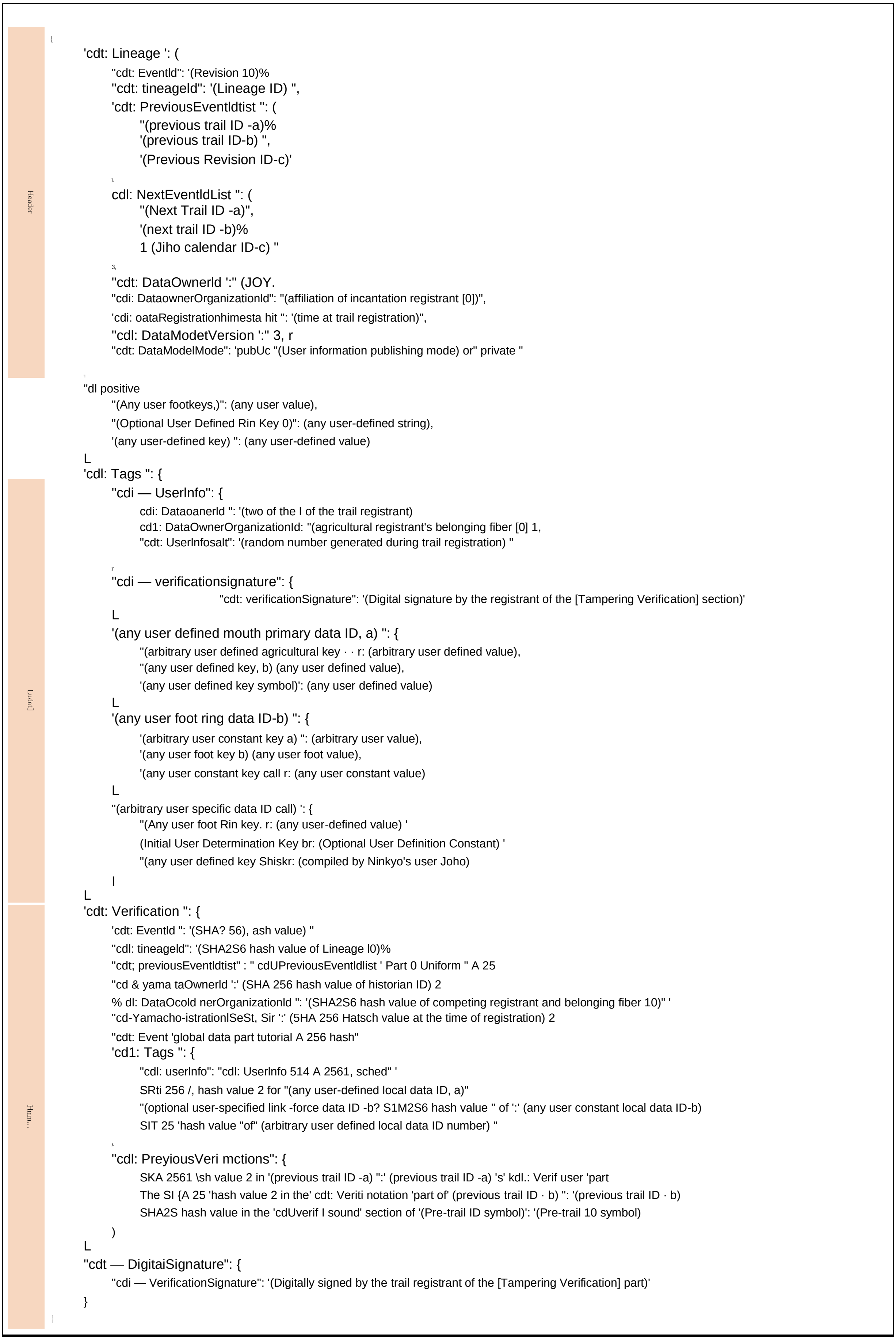
The received data and the trust seal are used to verify that the certificate used to create the trust seal is the creator itself or that the trust seal itself has not been tampered with.

The trust seal verification API is executed by verifier.

Appendix A JSON Format for the Trail and Audit Function

Trail Data JSON Format

JSON format for historical data that is returned when a lineage is acquired through the API using the trail and audit function.


Header
Ludat]
Hmm...

```
{
    'cdt: Lineage ': (
        "cdt: Eventld": '(Revision 10)%
        "cdt: tineageld": '(Lineage ID) ",
        'cdt: PreviousEventldtist ": (
            "(previous trail ID -a)%
            '(previous trail ID-b) ",
            '(Previous Revision ID-c)'
        ],
        cdl: NextEventldList ": (
            "(Next Trail ID -a)",
            '(next trail ID -b)%
            1 (Jiho calendar ID-c) "
        3,
        "cdt: DataOwnerld ':" (JOY.
        "cdi: DataownerOrganizationld": "(affiliation of incantation registrant [0])",
        'cdi: oataRegistrationhimesta hit ": '(time at trail registration)",
        "cdl: DataModetVersion ':" 3, r
        "cdt: DataModelMode": 'pubUc "(User information publishing mode) or" private "
    '|
    "dl positive
        "(Any user footkeys,)": (any user value),
        "(Optional User Defined Rin Key 0)": (any user-defined string),
        '(any user-defined key) ": (any user-defined value)
    L
    'cdl: Tags ": {
        "cdi — UserInfo": {
            cdi: Dataoanerld ": '(two of the I of the trail registrant)
            cd1: DataOwnerOrganizationId: "(agricultural registrant's belonging fiber [0] 1,
            "cdt: UserInfosalt": '(random number generated during trail registration) "
        }'
        "cdi — verificationsignature": {
                                "cdt: verificationSignature": '(Digital signature by the registrant of the [Tampering Verification] section)'
        L
        '(any user defined mouth primary data ID, a) ": {
            "(arbitrary user defined agricultural key · · r: (arbitrary user defined value),
            "(any user defined key, b) (any user defined value),
            '(any user defined key symbol)': (any user defined value)
        L
        '(any user foot ring data ID-b) ": {
            '(arbitrary user constant key a) ": (arbitrary user value),
            '(any user foot key b) (any user foot value),
            '(any user constant key call r: (any user constant value)
        L
        "(arbitrary user specific data ID call) ': {
            "(Any user foot Rin key. r: (any user-defined value) '
            (Initial User Determination Key br: (Optional User Definition Constant) '
            "(any user defined key Shiskr: (compiled by Ninkyo's user Joho)
        I
    L
    'cdt: Verification ": {
        'cdt: Eventld ": '(SHA? 56), ash value) "
        "cdl: tineageld": '(SHA2S6 hash value of Lineage l0)%
        "cdt; previousEventldtist" : " cdUPreviousEventldlist ' Part 0 Uniform " A 25
        "cd & yama taOwnerld ':' (SHA 256 hash value of historian ID) 2
        % dl: DataOcold nerOrganizationld ": '(SHA2S6 hash value of competing registrant and belonging fiber 10)" '
        "cd-Yamacho-istrationlSeSt, Sir ':' (5HA 256 Hatsch value at the time of registration) 2
        "cdt: Event 'global data part tutorial A 256 hash"
        'cd1: Tags ": {
            "cdl: userlnfo": "cdl: UserInfo 514 A 2561, sched" '
            SRti 256 /, hash value 2 for "(any user-defined local data ID, a)"
            "(optional user-specified link -force data ID -b? S1M2S6 hash value " of ':' (any user constant local data ID-b)
            SIT 25 'hash value "of" (arbitrary user defined local data ID number) "
        },
        "cdl: PreyiousVeri mctions": {
            SKA 2561 \sh value 2 in '(previous trail ID -a) ":' (previous trail ID -a) 's' kdl.: Verif user 'part
            The SI {A 25 'hash value 2 in the' cdt: Veriti notation 'part of' (previous trail ID · b) ": '(previous trail ID · b)
            SHA2S hash value in the 'cdUverif I sound' section of '(Pre-trail ID symbol)': '(Pre-trail 10 symbol)
        )
    L
    "cdt — DigitaiSignature": {
        "cdi — VerificationSignature": '(Digitally signed by the trail registrant of the [Tampering Verification] part)'
    }
}
```

List of components of trail data

historical data component	Key name	Type	Required	Contents and remarks
Header section	cdl: Lineage	Object	○	
Trail ID	cdl: EventId	String	○	Identifies the individual trail. No duplication across histories Generate UUIDs by default
Lineage ID	cdl: LineageId	String	○	A unique ID that identifies an individual lineage. Default If a previous trail ID is specified → Set the Lineage ID of the previous trail (I mean, I take over the Lineage of my previous trail.) If no previous trail ID is specified (= Lineage start) → Set the same ID string as the trail ID
Previous Trail ID Group	cdl: PreviousEventIdList	array	○	List of previous trail IDs (string). The trail at the beginning of the lineage is an empty array. When there are a plurality of previous trail IDs, it indicates the merging of lineages. When an empty array is specified at the time of trail registration, the previous trail ID is extracted and set by an automatic connection function of the Lineage by the Lineage ID.
Next Trail ID Group	cdl: NextEventIdList	array	○	List of next trail IDs (strings). The trail of the Lineage end is empty. When there are a plurality of next trail IDs, they represent branches of Lineage. The next trail ID is added when the next trail of this trail is added.
Trail registrant ID	cdl: DataOwnerId	String	-	ID of the trail registrant Required only in Public User Information
Trail Registrant Organization ID	cdl: DataOwnerOrganizationId	String	-	Organization ID of the trail registrant Required only in Public User Information mode
Trail registration time	cdl: DataRegistrationTimeStamp	String	○	Time when trail data is stored
CDL Data Model Version	cdl: DataModelVersion	String	○	3.0 (fixed)
CDL Data Mode	cdl: DataModelMode	String	○	public: (user info public mode) or private: (user info private mode)
Global Data section	cdl: Event	Object	-	If there is no user-defined data in the global data part, the key itself does not exist.

(user-defined data)	Any user-defined key (but not starting with "cdl:")	(any user defined value)	-	Any user-definable key-value data
[Local Data] section	cdl: Tags	Object	-	If there is no user-defined data in the local data part, the key itself does not
User information	cdl: UserInfo	Object	-	User information Required only in user information private
ID of the trail registrant	cdl: DataOwnerId	String	-	ID of the trail registrant

historical data component	Key name	Type	Required	Contents and remarks
				Required only in user information private mode
Organization ID of the trail registrant	cdl: DataOwnerOrganizationId	String	-	Organization ID of the trail registrant Required only in user information private mode
Random number generated during trail registration	cdl: UserInfoSalt	String	-	A random number for the purpose of user information identification prevention from a hash value. Required only in user information private mode
[Tampering Verification] Digital	cdl: VerificationSignature	Object	-	
Digital signature by the trail registrant of the [Tampering Verification] section	cdl: VerificationSignature	String	-	Digital signature by the trail registrant of the SHA 256 hash value of the [Tampering Verification] section Required only in user information private mode
local data ID group	Identify any user-defined key as a raw data ID (but not starting with "cdl:")	Object	-	User-definable "any local data ID-object" pair May overlap with local data ID of other trail, but treat as another local data (unique by trail ID + local data ID)
[Tampering	cdl: Verification	Object	○	
[Tampering Verification] Digital Signature	cdl: VerificationSignature	String	-	Digital signature by the trail registrant of the SHA 256 hash value of the [Tampering Verification] section
Lineage ID Tampering Verification	cdl: LineageId	String	○	SHA 256 hash value of [Header] section "Lineage ID"

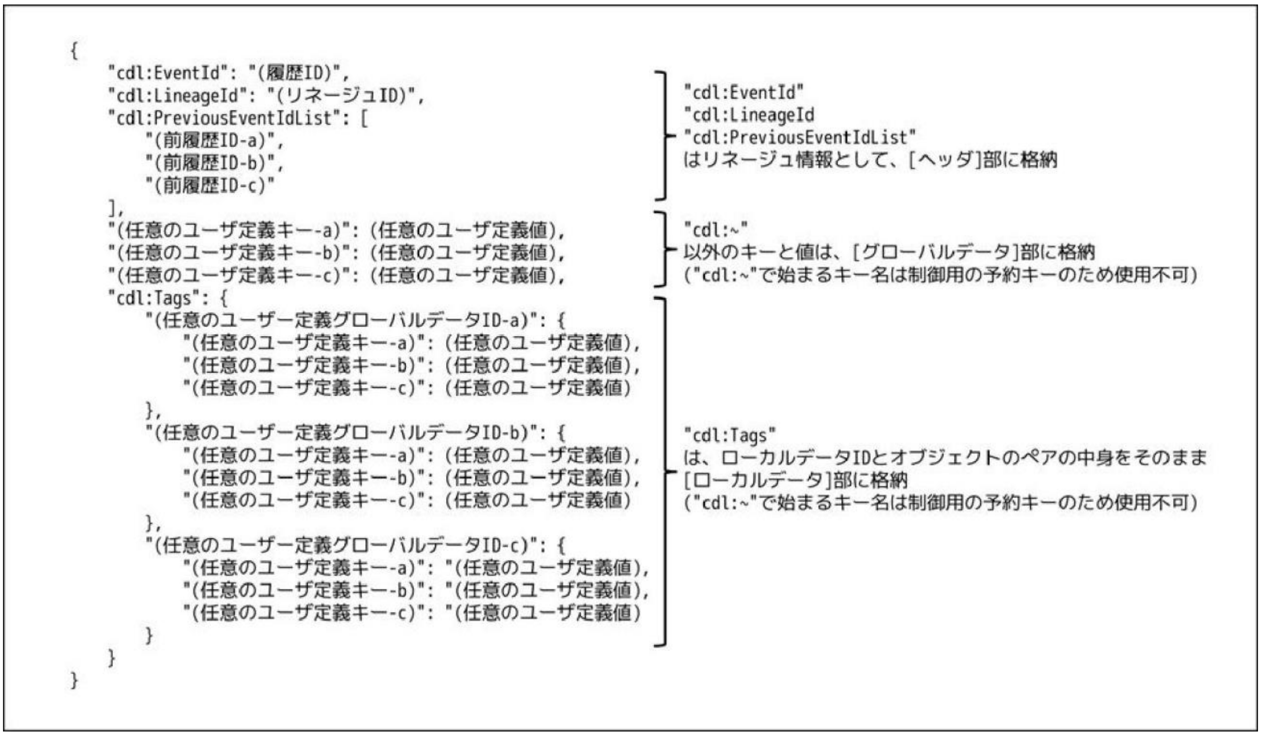
	Prior trail ID group tampering verification	cdl: PreviousEventIdList	String	○	SHA 256 hash value of [Header] section "Previous trail ID group"
	Verification of trail registrant ID	cdl: DataOwnerId	String	○	SHA 256 hash value of [Header] section [Trail registrant ID]
	Verification of alteration of organization ID	cdl: DataOwnerOrganizationId	String	○	SHA 256 hash value of "Organization ID belonging to trail registrant" in the [Header] section
	Tampering verification of trail	cdl: DataRegistrationTimeStamp	String	○	SHA 256 hash value of [Header] section [Trail registration time]
	[Global Data] part falsification	cdl: Event	String	-	SHA 256 hash value in the Global Data section
	[Local data] part falsification verification	cdl: Tags	Object	-	Local data in the [Local Data] part of the key ID " The string SHA 256 ha of the local data Chu value
	Previous trail [falsification verification] part falsification	cdl: PreviousVerifiactions	Object	○	Key Previous Trail ID The character string "In the [Tampering Verification] part of the previous trail, SHA 256 hash value "
[Digital Signature] Section		cdl: DigitalSignature	Object	-	Required only in Public User Information mode
	[Tampering Verification] Digital Signature	cdl: VerificationSignature	String	-	Digital signature by the trail registrant of the SHA 256 hash value of the [Tampering Verification] section
	Lineage Termination	cdl: LineageTerminationDigitalS	String	-	Present only in Lineage end trail. From the SHA 256 hash value and CDL of the

historical data component		Key name	Type	Required	Contents and remarks
					A string digitally signed with JWS (RFC 7515) the time the nege was retrieved.

JSON Format for Trail Registration

This is the JSON format that is used for trail registration.

Because the JSON format for trail information is cumbersome, such as having identical content in multiple places, a simplified version of the format is used for trail registration API requests.



```
{
    "cdl:EventId": "(履歴ID)",
    "cdl:LineageId": "(リネージュID)",
    "cdl:PreviousEventIdList": [
        "(前履歴ID-a)",
        "(前履歴ID-b)",
        "(前履歴ID-c)"
    ],
    "(任意のユーザ定義キー-a)": (任意のユーザ定義値),
    "(任意のユーザ定義キー-b)": (任意のユーザ定義値),
    "(任意のユーザ定義キー-c)": (任意のユーザ定義値),
    "cdl:Tags": {
        "(任意のユーザー定義グローバルデータID-a)": {
            "(任意のユーザ定義キー-a)": (任意のユーザ定義値),
            "(任意のユーザ定義キー-b)": (任意のユーザ定義値),
            "(任意のユーザ定義キー-c)": (任意のユーザ定義値)
        },
        "(任意のユーザー定義グローバルデータID-b)": {
            "(任意のユーザ定義キー-a)": (任意のユーザ定義値),
            "(任意のユーザ定義キー-b)": (任意のユーザ定義値),
            "(任意のユーザ定義キー-c)": (任意のユーザ定義値)
        },
        "(任意のユーザー定義グローバルデータID-c)": {
            "(任意のユーザ定義キー-a)": "(任意のユーザ定義値),
            "(任意のユーザ定義キー-b)": "(任意のユーザ定義値),
            "(任意のユーザ定義キー-c)": "(任意のユーザ定義値)
        }
    }
}
```

"cdl:EventId"
"cdl:LineageId
"cdl:PreviousEventIdList"
はリネージュ情報として、[ヘッダ]部に格納

"cdl:~"
以外のキーと値は、[グローバルデータ]部に格納
("cdl:~"で始まるキー名は制御用の予約キーのため使用不可)

"cdl:Tags"
は、ローカルデータIDとオブジェクトのペアの中身をそのまま
[ローカルデータ]部に格納
("cdl:~"で始まるキー名は制御用の予約キーのため使用不可)

JSON format component list at trail registration

JSON format when registering trail To	Key name	Type	Contents	Default behavior
Trail ID	cdl: EventId	String	A trail ID that identifies an individual trail. No overlap between histories	Automatically generate and set UUIDs

Lineage ID	cdl: LineageId	String	Lineage ID identifying the individual lineages. ID used by the Lineage auto-connect function	If a previous trail ID is specified for the previous trail ID group: ->Set the Lineage ID of the previous log (That is, it takes over the Lineage ID of the previous trail.) When the previous trail ID is not specified in the previous trail ID group Yes (= Lineage start): ->Set the same string as the trail ID
Previous Trail ID Group	cdl: PreviousEventIdList	array	List of previous trail IDs (string) representing previous trail of this trail If this record is at the beginning of Lineage, it is empty.	When the key itself is omitted, the previous trail ID is automatically extracted and set by an automatic connection function of the lineage by the lineage ID.

JSON format when registering trail To	Key name	Type	Contents	Default behavior
			When a plurality of previous trail IDs are specified, it indicates that lineages are	
Local Data	Any user-defined key (must not begin with "cdl:")	Any user defined value (any type)	Any user-definable key-value data Stored in the Local Data part of historical data and shared synchronized with all participating organizations (no access control)	If there is no user-defined data, the key “cdl: Event ” that represents the [Global Data] part of the trail data does not exist.
Local Data	cdl: Tags (must not begin with "cdl:")	Object	The contents of the object are given local data in "(any local data ID) - (JSON object)" pairs, which the user is free to define. It is stored in the [Local Data] part of historical data and is protected and hidden by access control when referenced. The local data ID may overlap with the local data ID of another trail, but it is treated as another local data (unique by trail ID + local data ID).	If no local data is specified, the key “cdl: Tags ” itself does not exist to indicate the [Local Data] part of the historical data.

Appendix B List of Service Delivery Types

The service delivery types for Fujitsu Data e-TRUST are:

Table B.1 Specifications of Standard Models by Type

Type	Number of company IDs	Storage Capacity (per company ID)	Number of users (per company ID)
Type SS	1000	0.05 GB	-'10
Type S	100	0.5 GB	-'100
Type M	10	5GB	-'1000
Type L	1	50 GB	-'10000

Table B.2 Optional Specifications (Add Company ID to Standard Model)

Option Type *	Number of additional company IDs	Storage capacity of the company ID to be	Number of enterprise ID users to add
Type SS	1	0.05 GB	-'10
Type S		0.5 GB	-'100
Type M		5GB	-'1000
Type L		50 GB	-'10000

*You can choose a different option type from selected standard model type.

Table B.3 Specifications of Development and Demonstration Models

Number of company IDs	Storage Capacity (per company ID)	Number of users (per company ID)
100	0.5 GB	-'100