

CIOFアーキテクチャ概要とシステム実装手順

2020年10月1日

西岡靖之

インダストリアル・バリューチェーン・イニシアティブ 理事長

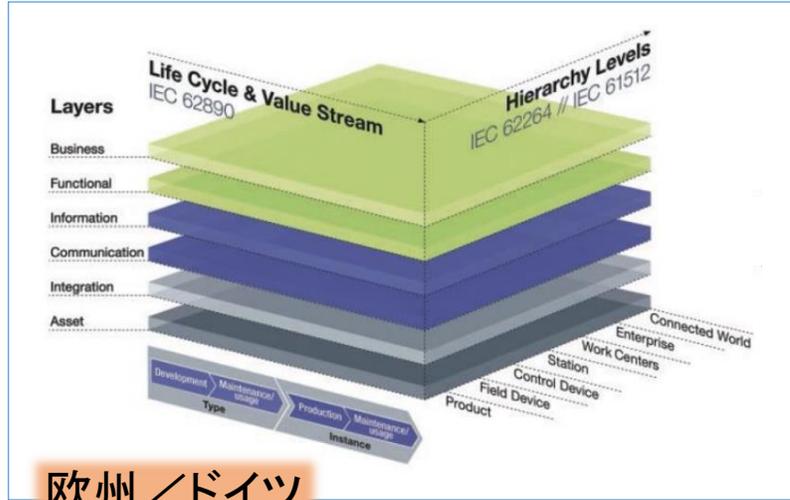
1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



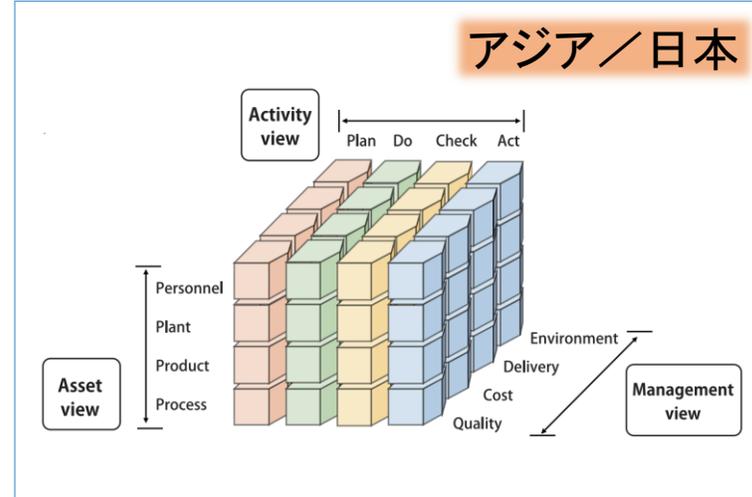
リファレンスアーキテクチャー(ものづくり分野)



Reference Architecture
Model Industrie 4.0

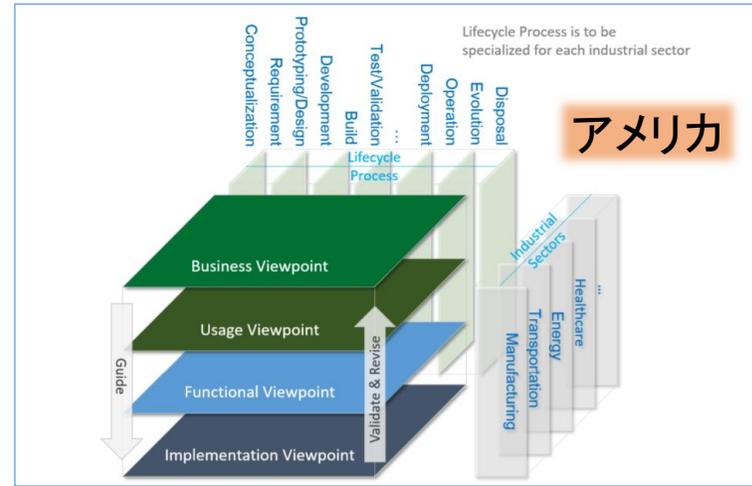
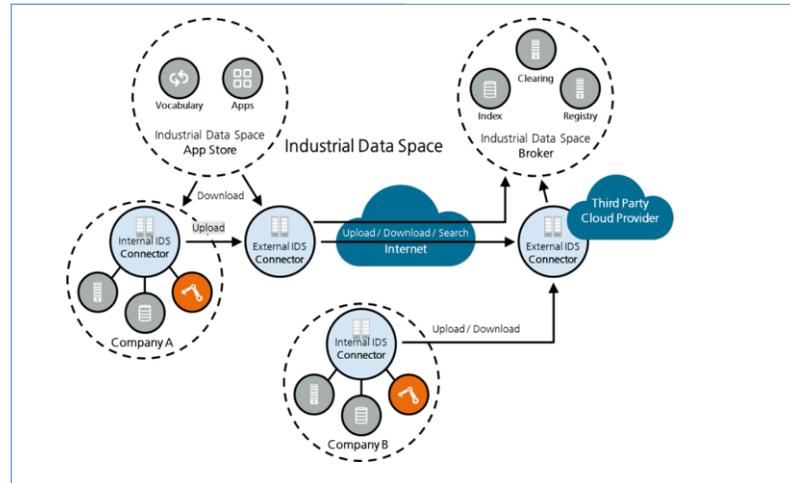


欧州/ドイツ



Industrial Value Chain
Reference Architecture

Industrial Data Space
Reference Architecture



Industrial Internet
Reference Architecture



日本のものづくりの特徴(良さ)は何か？



スマート
ものづくり単位
(SMU)

活動の視点

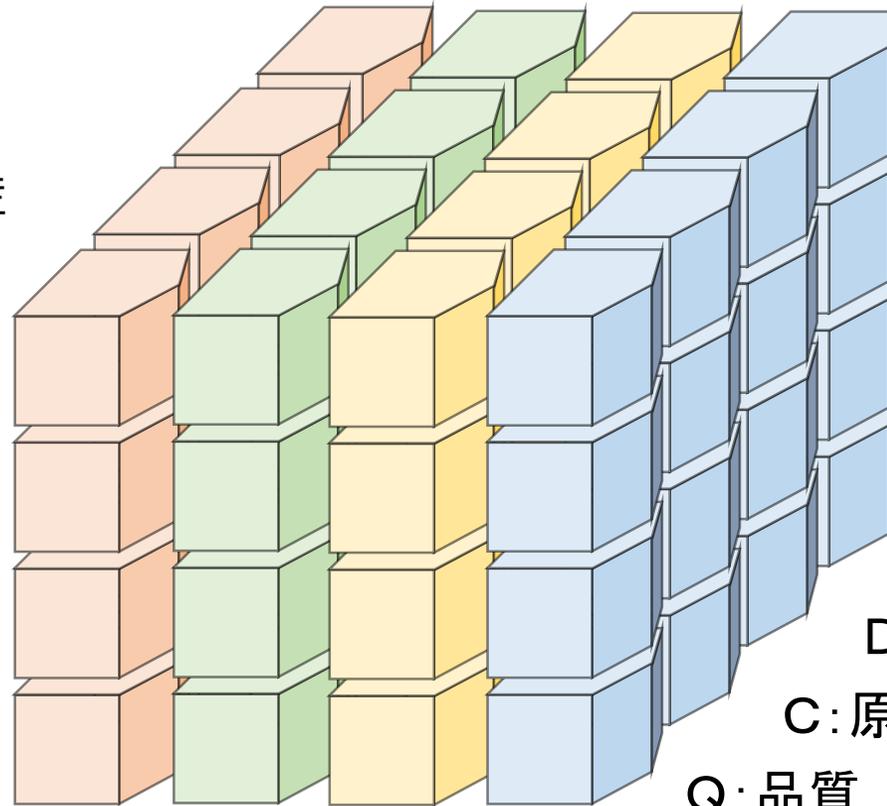
P: 計画
D: 実施
C: 解析
A: 改善

計画(P)、実行(D)、確認(S)ではなく、PDCAサイクルによって、改善をもとにうサイクルとしたことで、常に内部から進化し続けるしくみとなった。

現場の品質管理における要因は、ひと、機械、材料、方法のいずれかに集約される。ものづくりの資産はこの4つから構成される。

資産の視点

M:ひと
M:機械
M:材料
M:方法



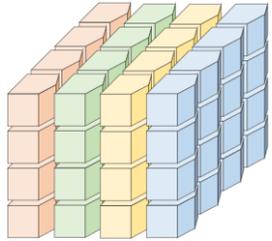
品質と合わせて原価低減、ジャストインタイム、環境への配慮などを徹底し、バランスのとれた管理指標を意思決定の基本とする。

E:環境
D:納期
C:原価
Q:品質

管理の視点



スマートものづくり単位 (SMU)



活動の視点 (PDCA)

資産の視点 (4M)

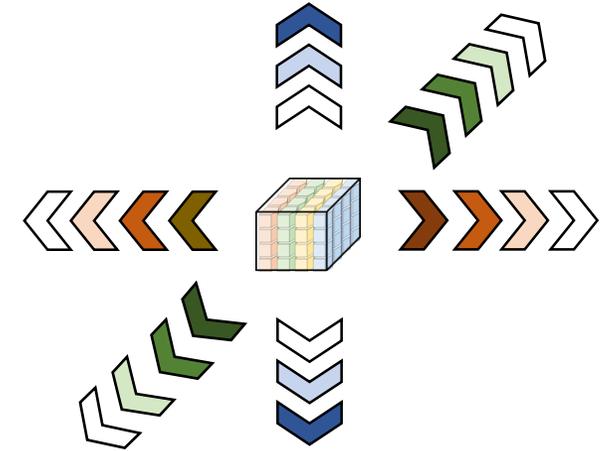
管理の視点 (QCDE)

3つの軸のバリューチェーン

製品軸

サービス軸

知識軸

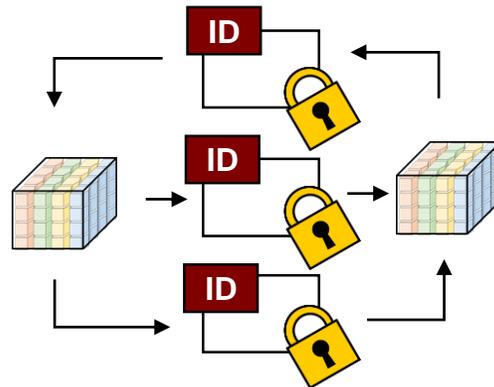


スマート移動単位 (PLU)

バリューコンテナ

サイバーコンテナ

フィジカルコンテナ



IVIオントロジ



役者



情報



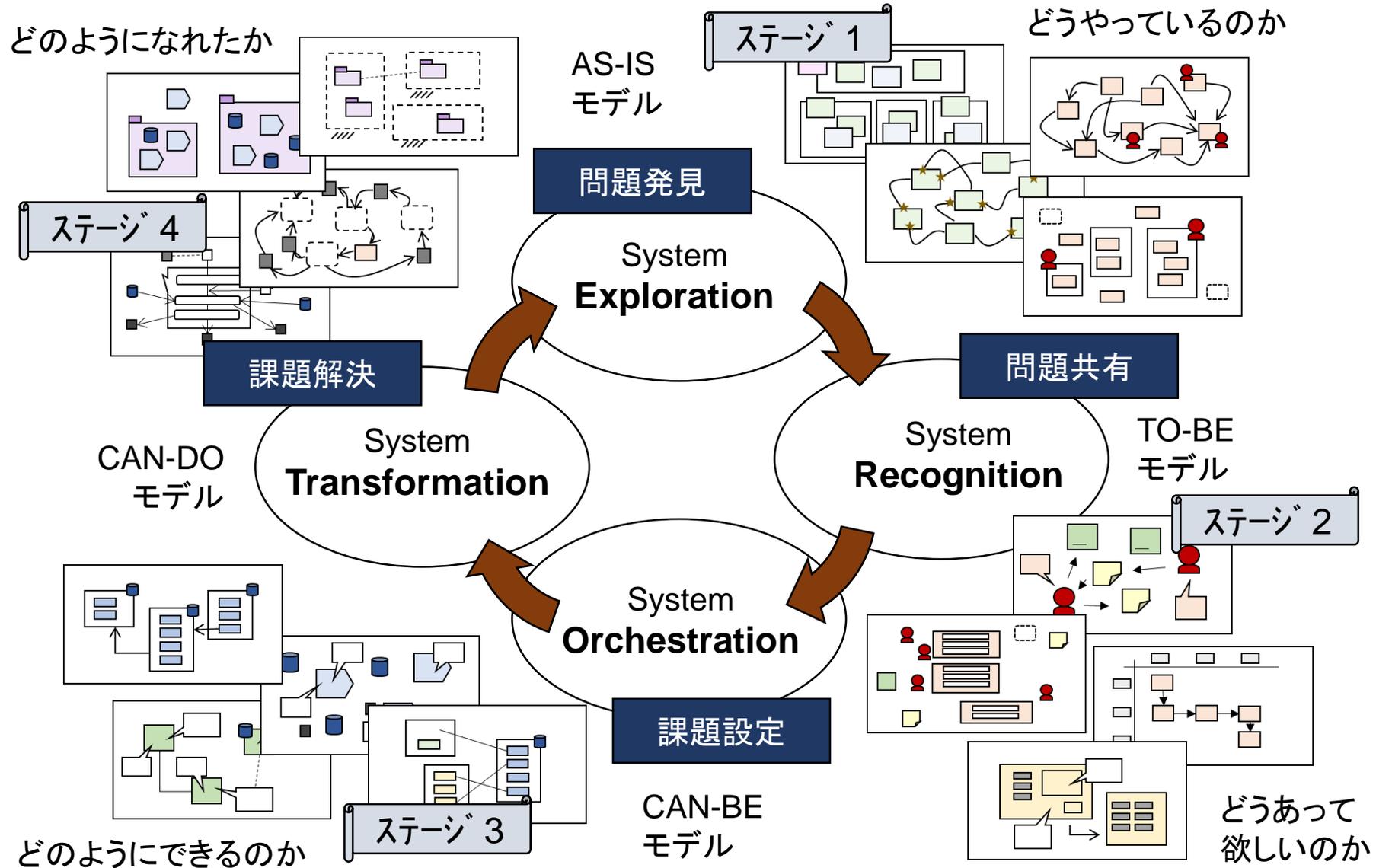
モノ



データ



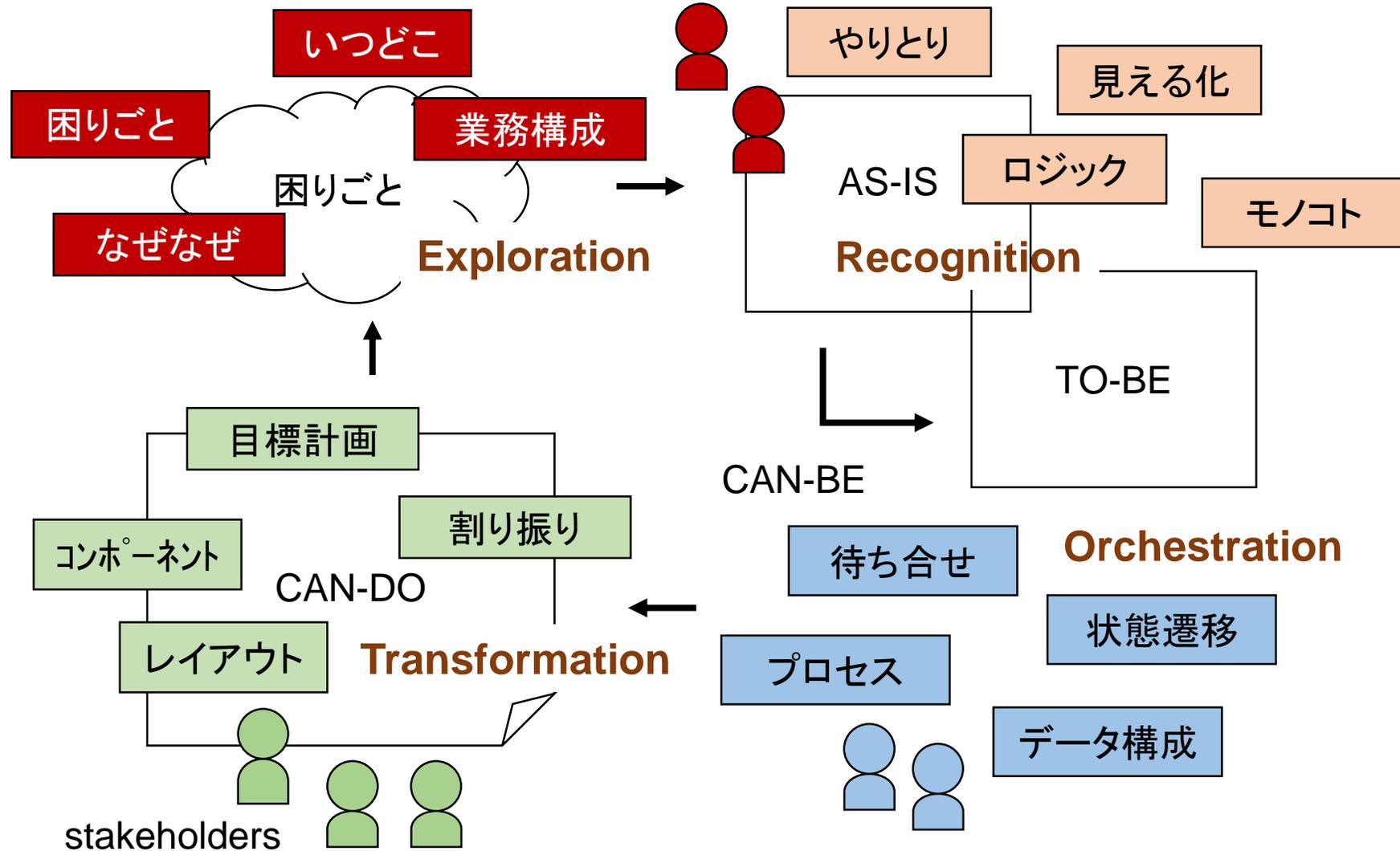
スマートシンキングのサイクル



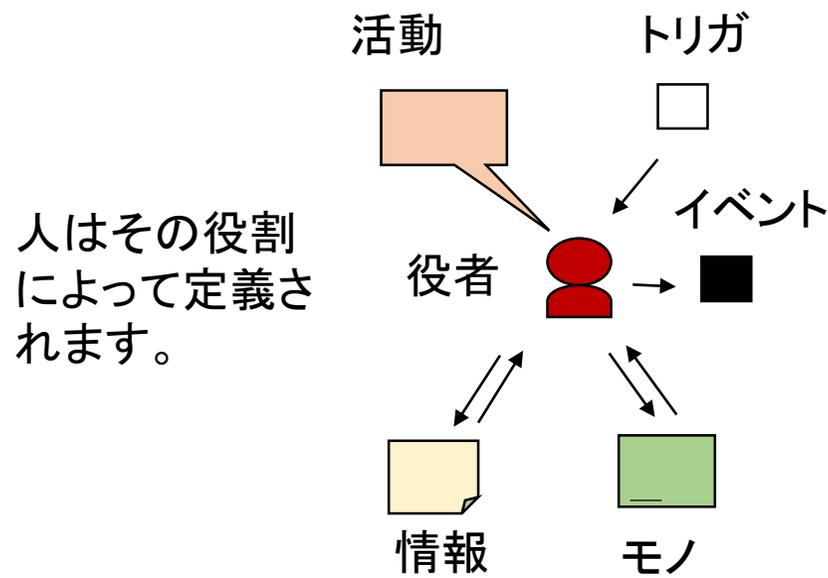
- AS-IS
 - 現状の仕事のやり方、業務の進め方をそのまま記述し、何が問題であるのか、そしてその原因となっている問題は何かを相互に理解する。
- TO-BE
 - あるべき姿、ありたい姿として、理想ではなく、実際にこれから実現させる内容を、関係者が理解し共通の目標となるよう具体的に示す。
- CAN-BE
 - あるべき姿を実現するためのソリューション(解決策)について具体的な手順を明らかにし、そのための道具とリソースを準備する。
- CAN-DO
 - 実際に実現するために、人の問題(イナーシャ、抵抗勢力など)、お金の問題、時間の問題、不確定要素の問題などに対応する。



スマートシンキングの流れ



オペレーションレベルのモデリング

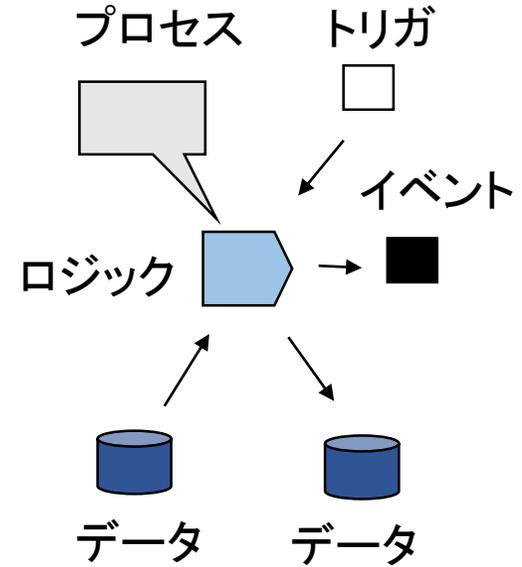
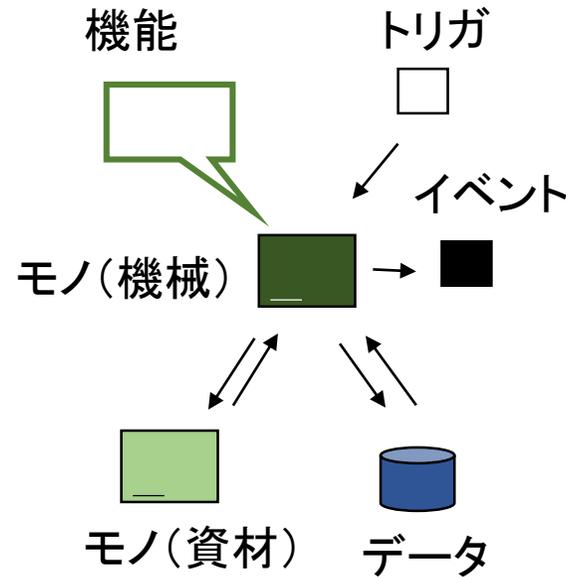


人はその役割によって定義されます。

データと紐づいたモノ、情報はデータアイコンを付与します。



役者は直接データを扱わず、情報として認識します。



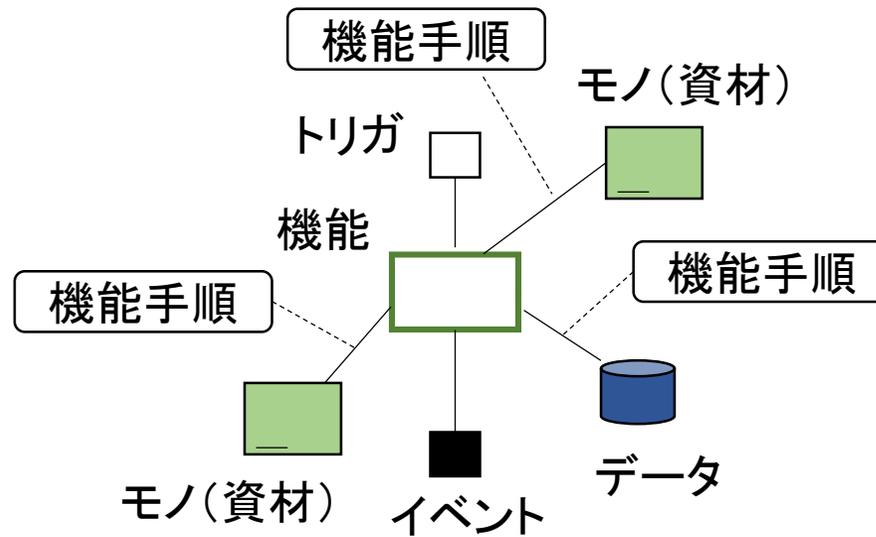
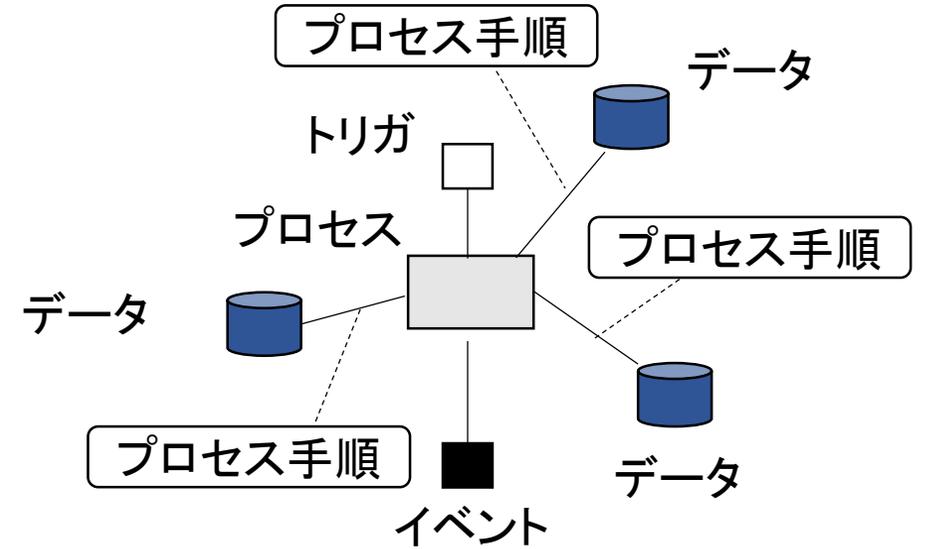
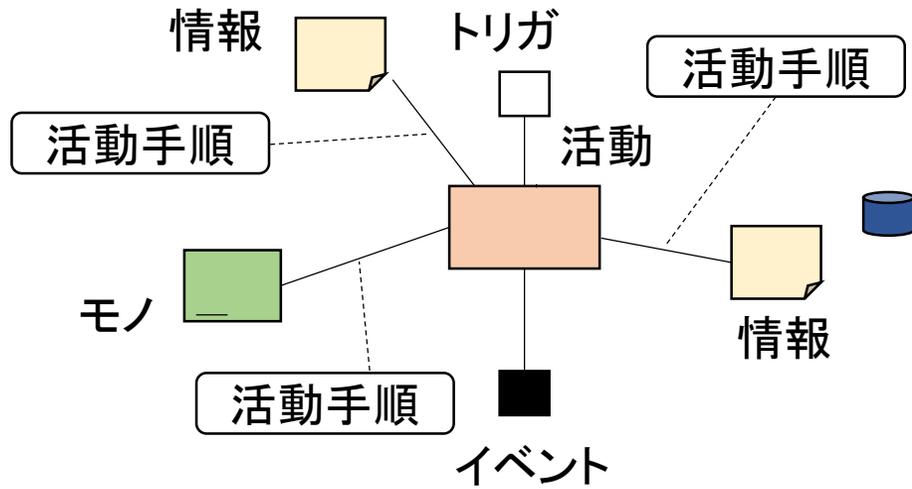
注)ロジックは、CIOFでは“サービス”と呼ばれます。

イベントはトリガとして他の動作に伝播します。

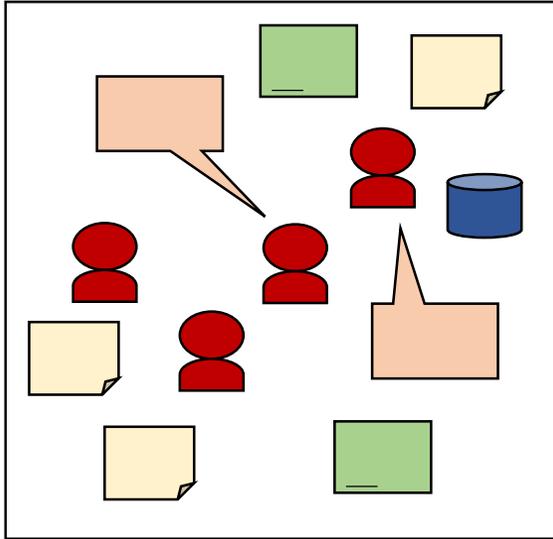
モノは設備(主体)と資材(客体)の2つの側面があります。



オペレーションレベルのモデリング(詳細)



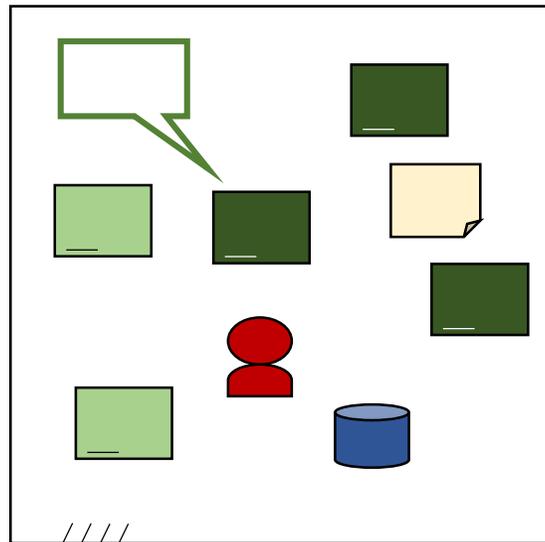
組織



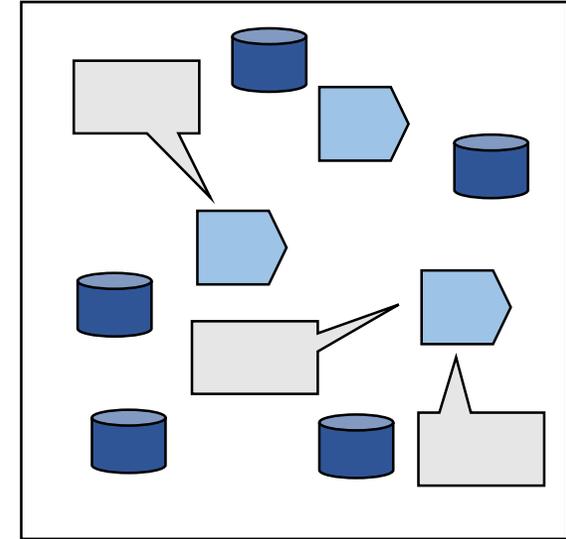
ヒューマン視点でまとまった単位を“組織”として定義します。

フィジカル視点でまとまった単位を“エリア”として定義します。

エリア



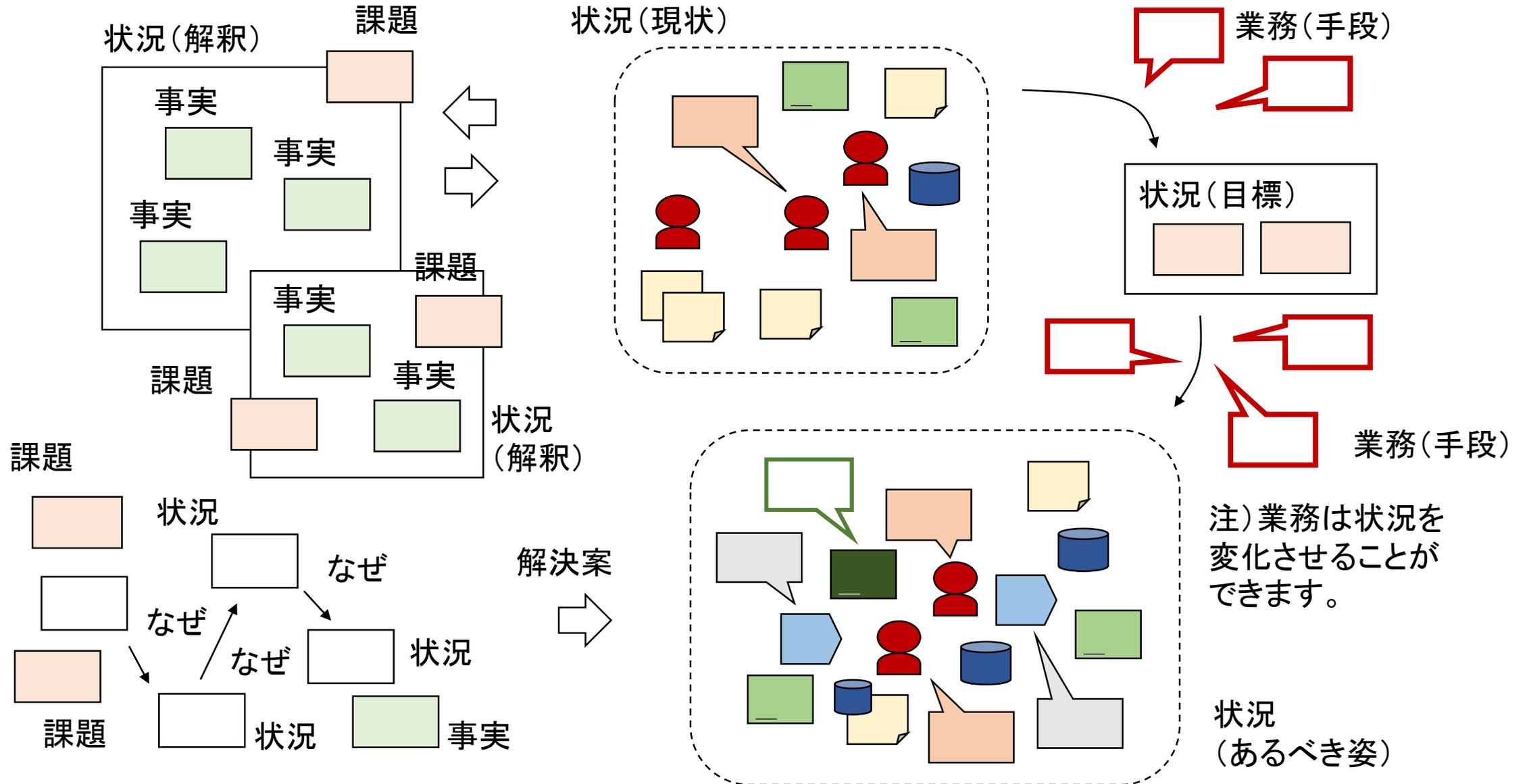
コンポーネント



サイバー視点でまとまった単位を“コンポーネント”として定義します。

注) エリアには組織やコンポーネントを含むことができます。

オントロジーとEROTの関係



オントロジー(対象世界を記述する基本語彙)



マネジメント レベルの単位

事実

状況

課題

業務

価値

オペレーションレベルの単位

活動



機能



プロセス



役者



モノ(機械)



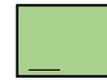
ロジック



情報



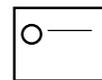
モノ



データ



カレンダー



トリガ



イベント



インプリメンテーション レベルの単位

作業

手順

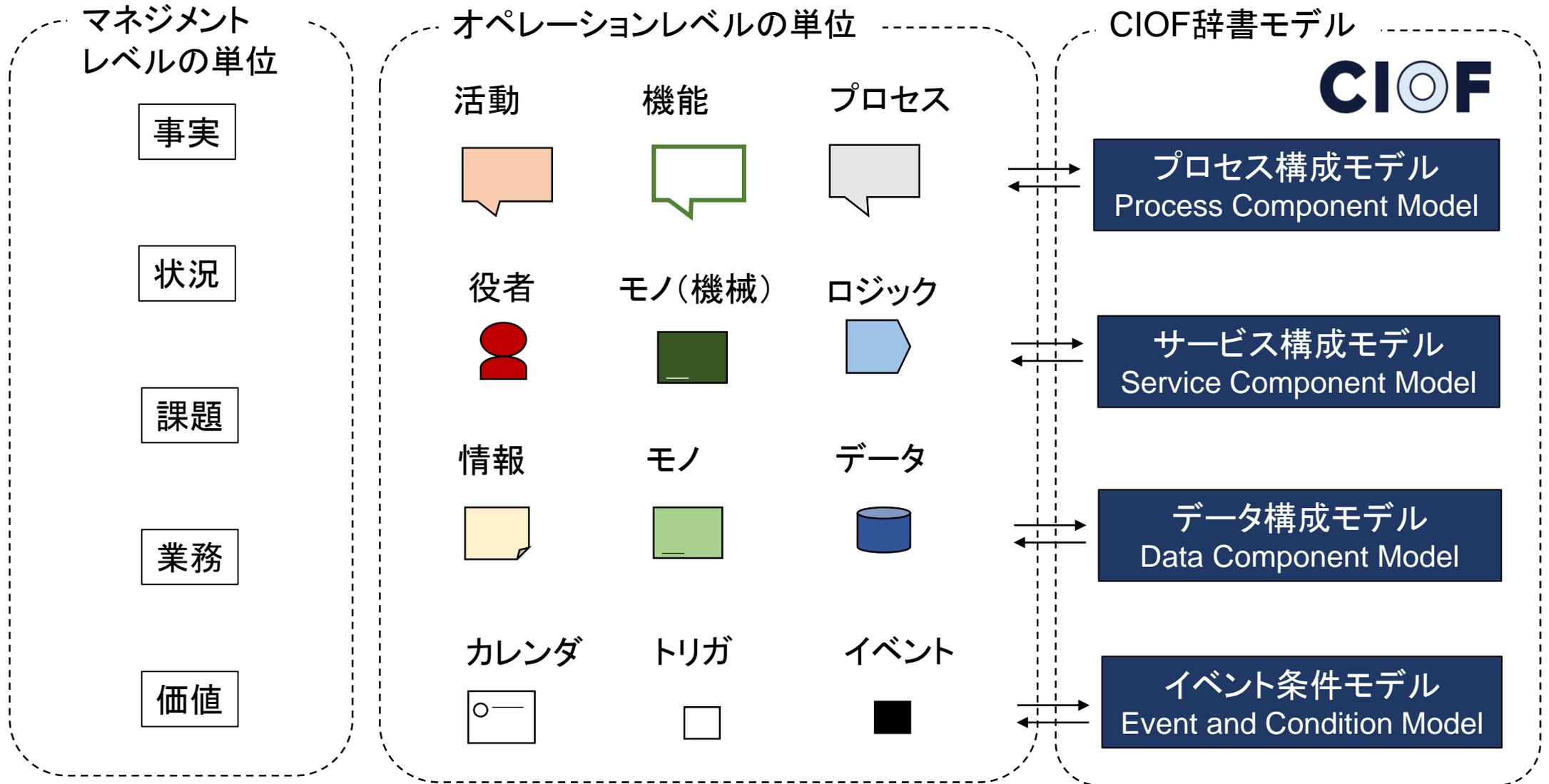
項目

状態

条件

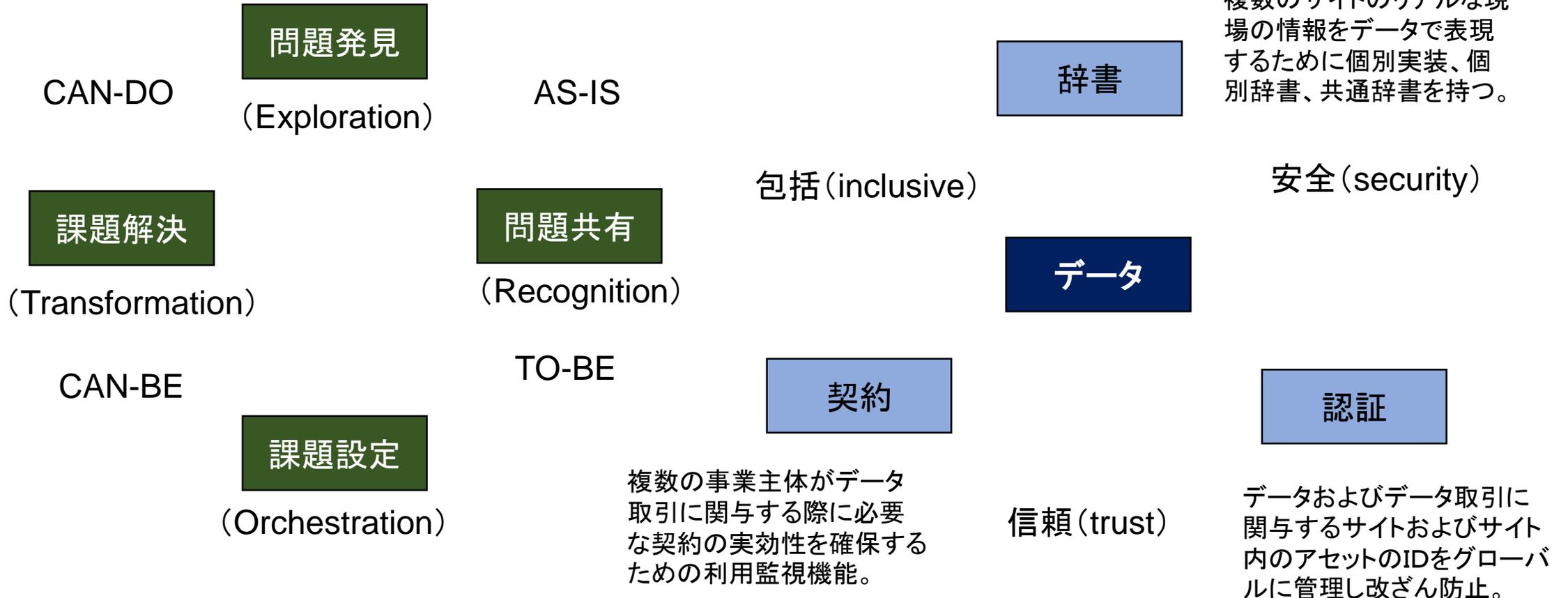


オントロジー（対象世界を記述する基本語彙）



スマートシンキング(EROT)

企業間オープン連携(CIOF)



1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



- 基本サービス
 - 機器のコンフィグ設定、アセットの設定(実装)、ユーザ設定、権限付与、IDの発行など
- 契約サービス
 - 個別契約書、共通契約書の作成、相手との通信、契約情報の照会、過去履歴の管理など
- 伝送サービス
 - PUSH/PULL、パブサブ、コレクト等のデータ伝送、履歴の記録、送信先の認証、暗号化、経路設定、リカバリなど
- 辞書サービス
 - 共通辞書、外部辞書の登録、検索機能、リコメンド機能、パブスク、履歴(バージョン)管理など



- 変換サービス
 - 伝送データの変換、レコード変換、パラメータ変換、API設定、ルール設定、テスト環境など
- 検索サービス
 - 取引先検索、プロフィール検索、ビジネスマッチング、取引履歴照会、企業グループ設定など
- 証明書サービス
 - データ伝送履歴の照会、データ利用実績の照会、証明書の発行、正当性保証機能、改ざん履歴の管理、など
- 認証サービス
 - グローバルID機能、ソフトウェア認証、ハードウェア認証、認証局の管理、など(実装対象外)



- データ取引 (Data Trading)
 - データ取引とは、データを異なるターミナル間で伝送するに際して、あらかじめ、データの提供者とデータの利用者の双方で、データに関する権利および義務を定め、それを履行すること。1つの取引は、その契約の締結から実際のデータの伝送、そしてそれに伴う権利と義務の履行、そして契約の失効までの一連の行為を含む。ここで対象となるデータを、取引データと呼ぶ。
- 取引データ (Trading Data)
 - データ取引の対象となる個々のデータのこと。連携サーバが固有のIDと付与する。連携サーバIDと取引データIDでグローバルにユニークとなる。取引データは、ハッシュが生成され、それを取引データの実体の代わりに保存することで、取引データ間の突合せを可能とする。また、取引データのトレーサビリティは、このハッシュを用いる。



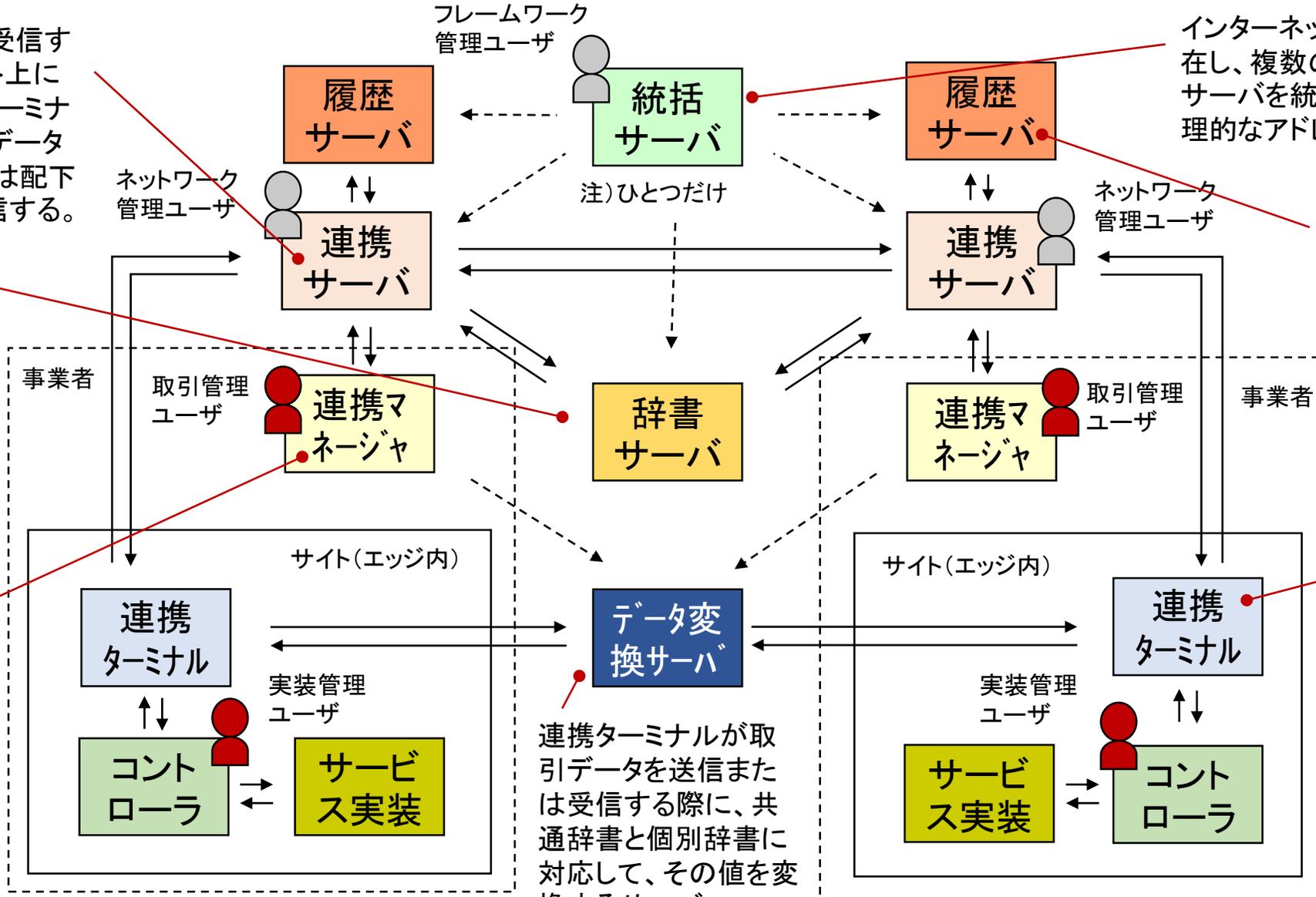
アーキテクチャ全体構成



実際取引データを送受信するために、インターネット上に1つ以上存在し、連携ターミナルから受け取った取引データを他の連携サーバまたは配下の連携ターミナルに送信する。

共通辞書、個別辞書、外部辞書を管理し、その照会および登録、更新、削除を行う。また、名寄せなどにより、個別辞書から共通辞書の半自動生成やリコメンドのための機能をもつ。

辞書の登録管理および取引契約の登録管理等を行う。また、履歴の確認および証明のためのユーザインタフェースを持つ。



インターネット上にただ1つだけ存在し、複数の連携サーバ、履歴サーバを統括し、それらのIDや物理的なアドレスなどを管理する。

データ取引契約や、連携サーバが行った実際の取引データの送受信記録、さらには連携ターミナルから送られるデータ利用実績などを、分散台帳として管理する。

ファイアウォールの内側において、エッジに近い特定のエリアに含まれるデバイスに対して、データの送受信の際の外部との接点となりセキュリティやトレーサビリティを一括管理する。



論理的階層

説明

つながる世界 (Connected World)

世の中のサーバー・フィジカル世界全体を示す。ここでは、事前の合意なく、さまざまなネットワークが自己増殖的にエコシステムを形成されている。そうした複数のエコシステムからなる世界である。

CIOF世界 (COIF World)

CIOFのサービスが適用できる全体の範囲。CIOFが関与したつながる世界として定義できる。これは、統括サーバ(FCS)が管理する範囲に相当する。

ドメイン (Domain)

さまざまな事業者が、ひとつのプラットフォーム企業あるいは団体によってまとまった単位。これは、連携サーバが管理する範囲に相当する。事業者は必ずひとつのドメインに属する。

事業者 (Enterprise)

データ取引を主体的に行う単位で、企業または事業部に相当する。事業者が管理する範囲は複数の工場におよぶ場合もある。したがって、事業者は複数のサイトをもつことができる。

サイト (Site)

機器が相互に連携する範囲であり、1つの連携ターミナルが管理する。モノの移動、およびデータの移動は、この単位で管理する。サイト内は、ファイアウォールによって外部から守られる。

エッジ (Edge)

リアルタイム性が要求されるエッジ内部の世界。ハードウェアに相当するエッジデバイス、ソフトウェアに相当するサービス実装およびデータ実装が存在し、独自のIDによって管理される。

現物 (Physical Asset)

個別のアセットとして単体で機能をもつ単位の集まり。ハードウェアとソフトウェアが一体となって、目に見えるモノとして認識可能。IoTにおけるつながる最終的な対象(Thing)に相当する。



CIOFの階層構造



論理的階層	CIOFアセット	権限の階層	IDの階層
つながる世界 (Connected World)			標準グローバルコード
CIOF世界 (CIOF World)	統括サーバ (Framework Control Server)	 フレームワーク管理者	CIOF世界内のユニークID
ドメイン (Domain)	連携サーバ (Hyper Connection Server)	 ドメイン管理者	
事業者 (Enterprise)	連携マネージャ (Hyper Connection Manager)	 事業者管理者	ドメイン内のユニークID
サイト (Site)	連携ターミナル (Hyper Connection Terminal)	 取引管理者	
エッジ (Edge)	エッジコントローラ (Edge Control Unit)	 実装管理者	コントローラ内部の管理ID
現物 (Physical Asset)			サービス固有ID



標準グローバルコード

あらゆる世界でユニークなIDは、理論的には存在しませんが、特定の分野や標準などを指定することで、一意に定めることは可能です。CIOFでは、CIOF世界でのユニークIDと、外部のグローバルIDとを対応づけることで、あらゆる外部のシステムとつなげることができます。

CIOF世界内のユニークID

CIOFは異なる事業者が管理するプラットフォームをつなげる分散システムであるため、それぞれのプラットフォームが自律的に機能するため共通的なID管理は必要最小限にしています。例としては、ドメインID、事業者ID、サイトIDがあります。

ドメイン内のユニークID

CIOFで利用する多くのデータは、ドメイン内、つまり連携サーバごとに定義されるプラットフォーム内で共通のIDによって管理されます。事業者またはサイトが、他のドメインに移管される場合は、別途IDを発行する必要があります。

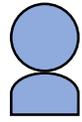
コントローラ内部の管理ID

エッジ内部でローカルなIDは、CIOFが管理するIDとは別に、独自に発行し管理することができます。エッジコントローラは、CIOFが管理するIDと、この独自の内部IDとの間で、コントローラの責任の中で対応づけを行う必要があります。

サービス固有のID

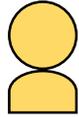
現場において実際に利用されるさまざまなアセットは、それぞれのメーカーやサービス提供者が設定した独自のIDが設定されています。こうしたIDは、コントローラ内部の管理IDとして、そのまま利用することも可能です。





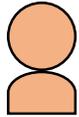
**フレームワーク
管理者**

CIOF世界全体を管理する。ドメインの新規追加、事業者およびサイトの登録および管理などを行う。また、取引履歴に関する正当性の管理や証明書の発行、およびセキュリティの管理を行う。



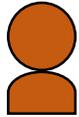
ドメイン管理者

各プラットフォームとして、共通辞書の管理、事業者の登録受付、連携サーバの管理とメンテナンス、課金およびサポートなどを行う。事業者管理者に対するIDやパスワードなどの権限設定を行う。



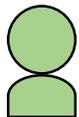
事業者管理者

事業者全体の管理者として配下にあるサイト(連携ターミナル)の設定、連携マネージャへのログイン可能なユーザを設定し、取引管理者、実装管理者、および辞書管理者の権限設定を行う。



取引管理者

データ取引に関する契約を作成することができる権限を持つ。取引管理者は、対象サイトに帰属するデータについて、自分の事業者を含むすべての他のサイトとの間のデータ流通を管理する。



実装管理者

個々のサイトにおけるデータ実装、サービス実装の管理する。また、サイト内のエッジコントローラの追加や削除、およびコントローラ内で扱うデータ実装、サービス実装と、それぞれのコントローラ内で有効な内部IDとの対応を管理する。

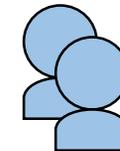


データ取引の5つのフェーズ



モデリング
フェーズ

対象業務に対応した辞書や業務アプリの構造などを登録する。



プラットフォーム、コンポーネント企業、標準化団体

インテグレーション
フェーズ

対象現場のデバイスや機器を登録し、利用する辞書に設定する。



システムインテグレーター、対象業務のエキスパート

データ取引契約
フェーズ

提供パターンに対応したデータの取り扱いに関する契約を交わす。



マーケティング担当者、取引先営業、契約担当者

データ取引実施
フェーズ

契約内容にしたがって、実際にデータ提供またはデータ利用を行う。



調達担当者、出荷担当者、現場の業務担当者

データ取引確認
フェーズ

すでに実施したデータ提供や利用履歴の照会や事実認証を行う。



経理担当者、品質担当者、経営戦略室スタッフ



共通辞書

プラットフォーム企業

- ✓異なるターミナル間でデータやサービスに関する用語を対応づけるために利用する。プラットフォーム企業または標準化機関によって登録されます。

外部辞書

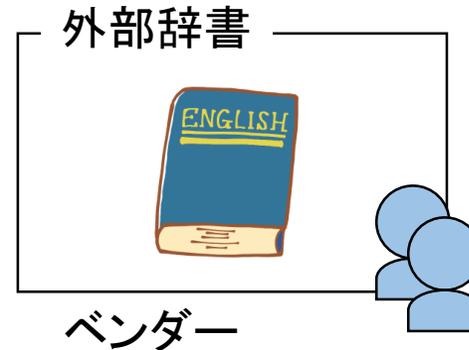
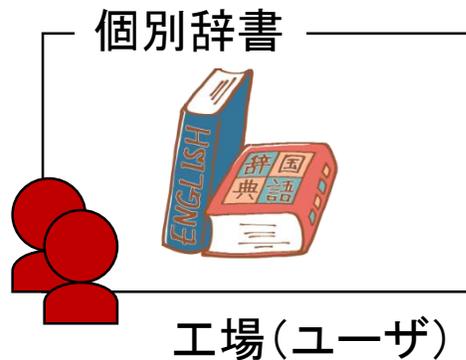
コンポーネント企業

- ✓アプリやデバイスの提供者が作成し、その内容をデータ取引を行う事業者が個別辞書インポートして利用する。

個別辞書

データでつながる企業

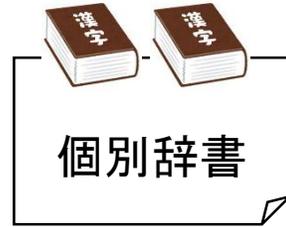
- ✓それぞれのターミナルで、独自に実装されているデータやサービスの内容を定義する。データ取引はこの内容に対して行われる。



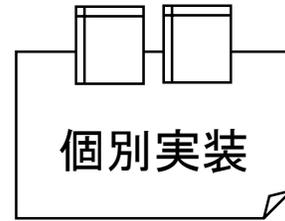
共通辞書と取引データの関係



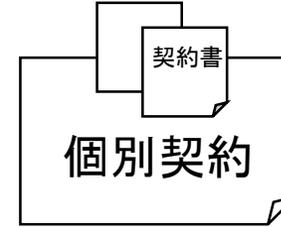
事業者間で意味を伝えるための共通となる用語を定義します。



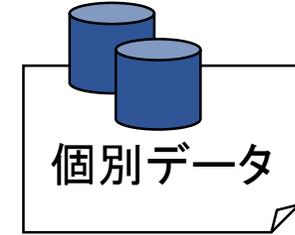
事業者内で利用する用語を定義または外部から取得します。



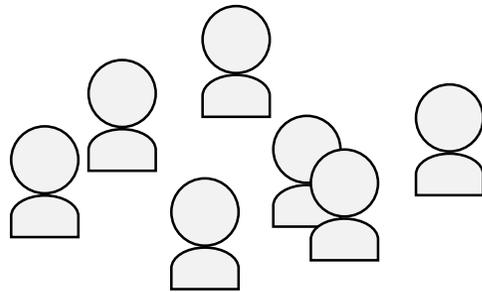
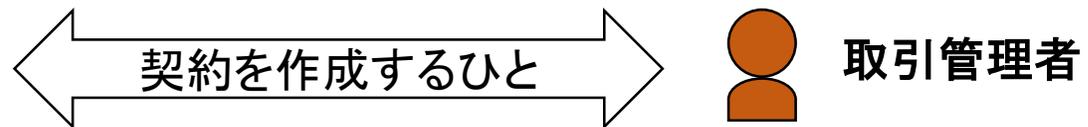
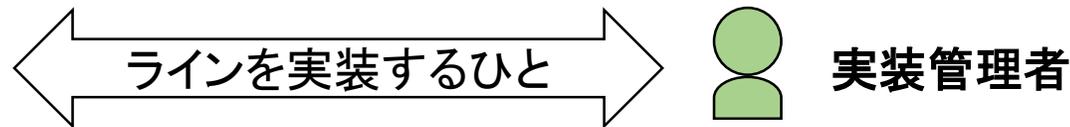
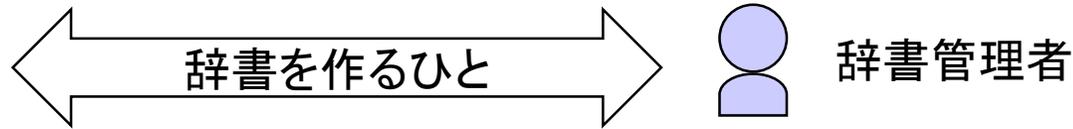
サイトに個別のシステムとしてサービスとデータを実装します。



サイト間で個別に契約を作成するときを設定します。



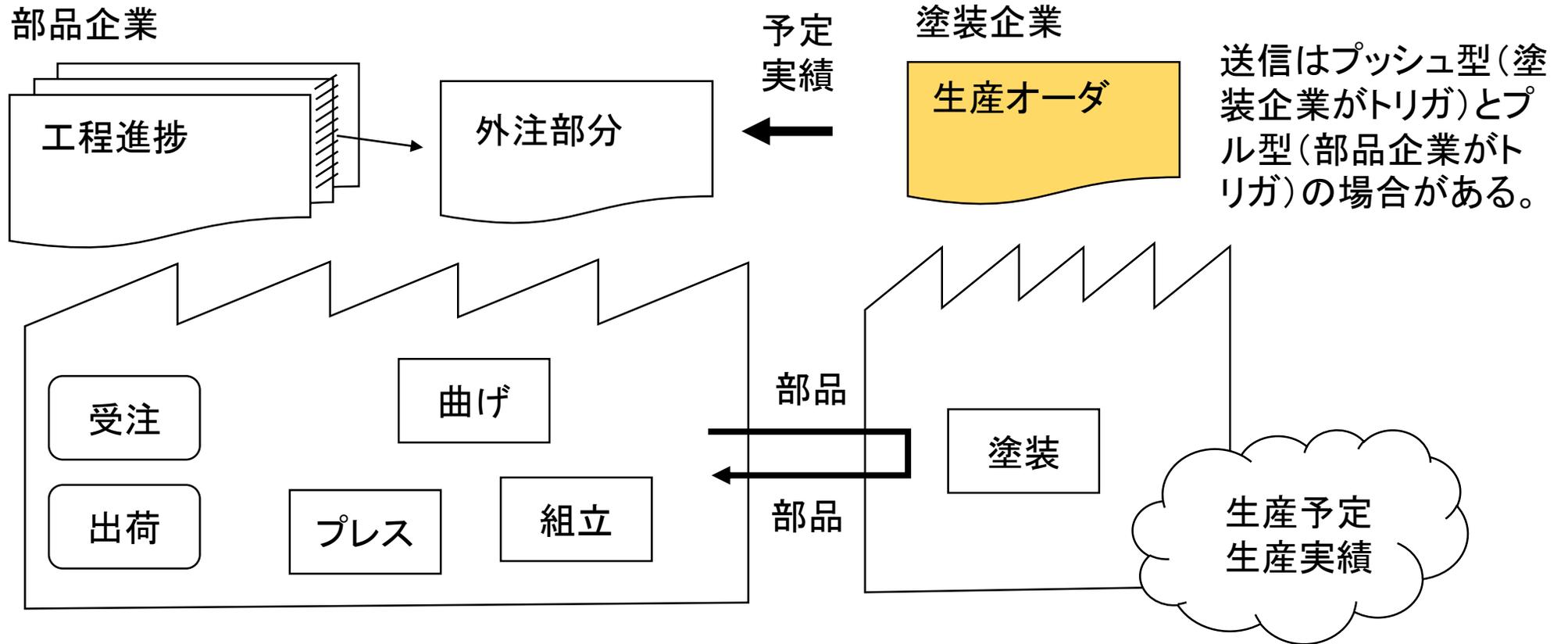
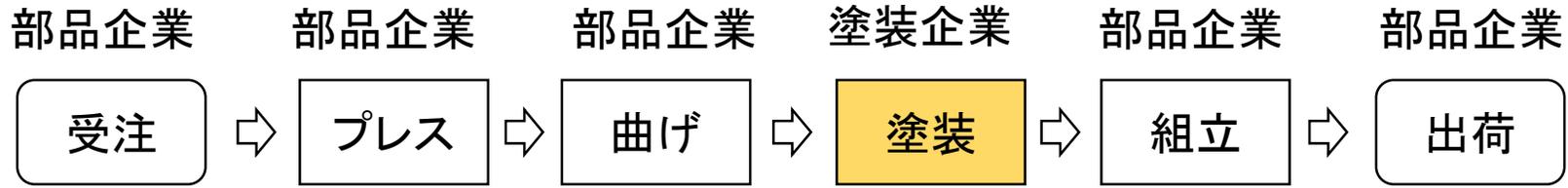
契約にしたがい一回でデータを送受信する単位です。



1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



ユースケース(1)



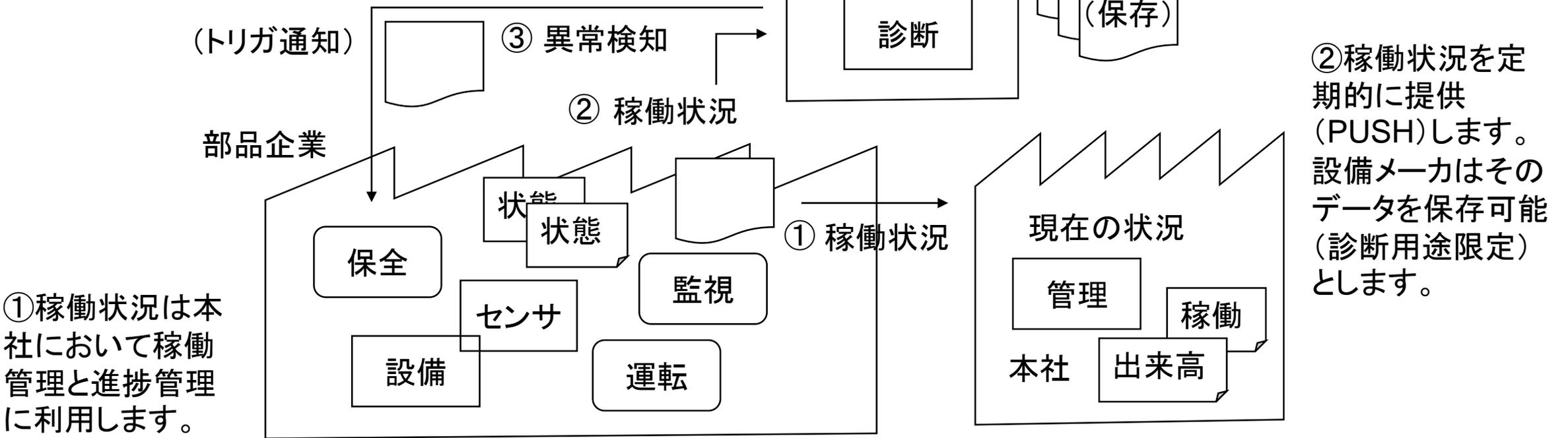
協力企業に外注した工程の着手、完了予定、および実績を、定期的に発注元企業に送信する。



ユースケース(3)



③異常が検知された場合には、トリガ実装により異常箇所と状況が通知されます。

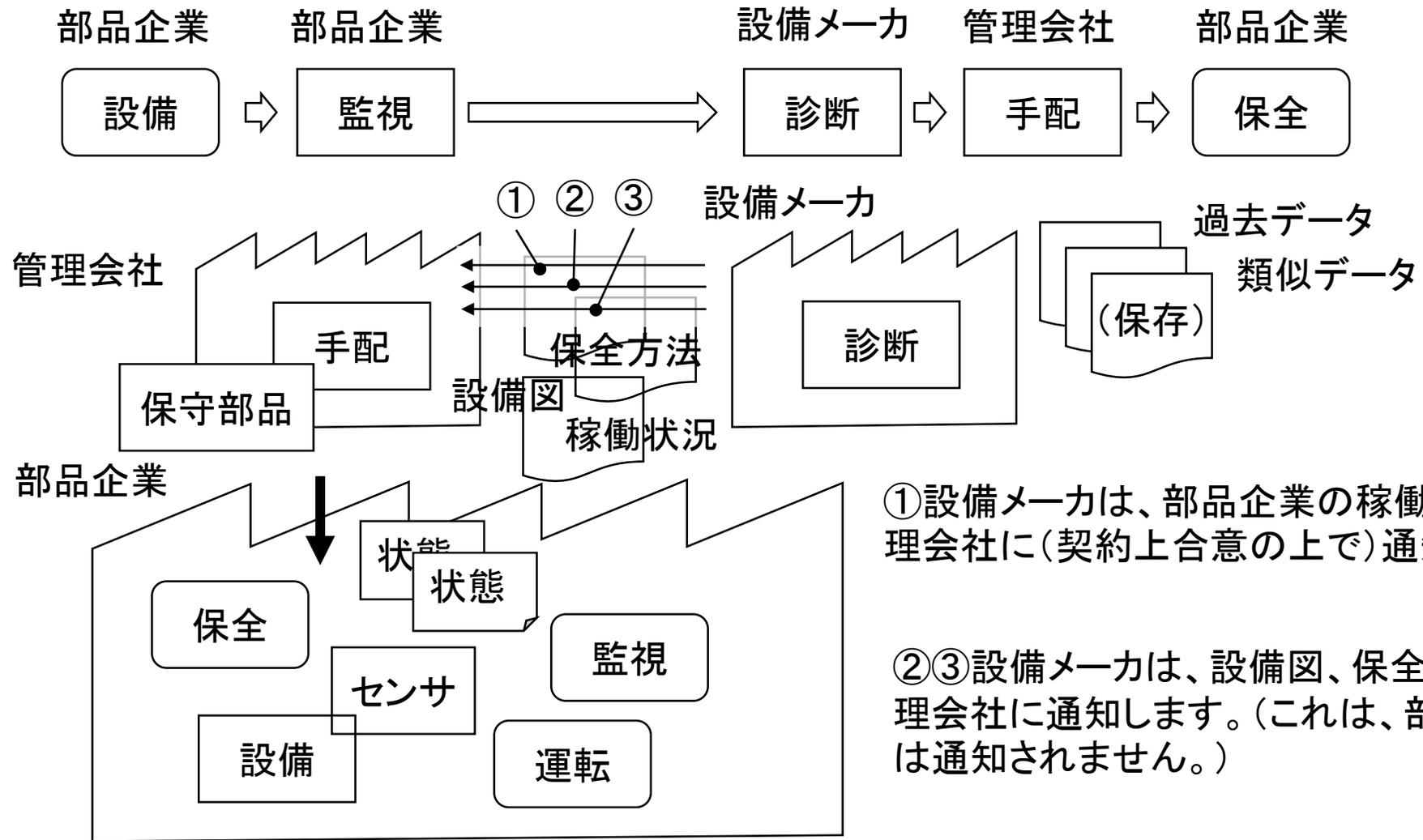


①稼働状況は本社において稼働管理と進捗管理に利用します。

②稼働状況を定期的に提供 (PUSH) します。設備メーカーはそのデータを保存可能 (診断用途限定) とします。



ユースケース(4)

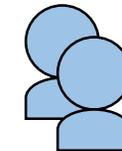


データ取引の5つのフェーズ



モデリング
フェーズ

対象業務に対応した辞書や業務アプリの構造などを登録する。



プラットフォーム、コンポーネント企業、標準化団体

インテグレーション
フェーズ

対象現場のデバイスや機器を登録し、利用する辞書に設定する。



システムインテグレーター、対象業務のエキスパート

データ取引契約
フェーズ

提供パターンに対応したデータの取り扱いに関する契約を交わす。



マーケティング担当者、取引先営業、契約担当者

データ取引実施
フェーズ

契約内容にしたがって、実際にデータ提供またはデータ利用を行う。



調達担当者、出荷担当者、現場の業務担当者

データ取引確認
フェーズ

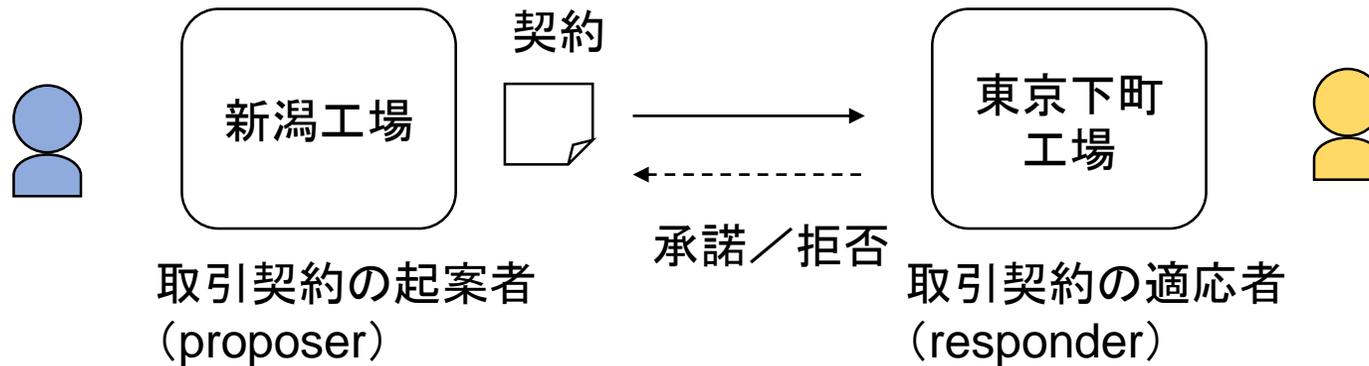
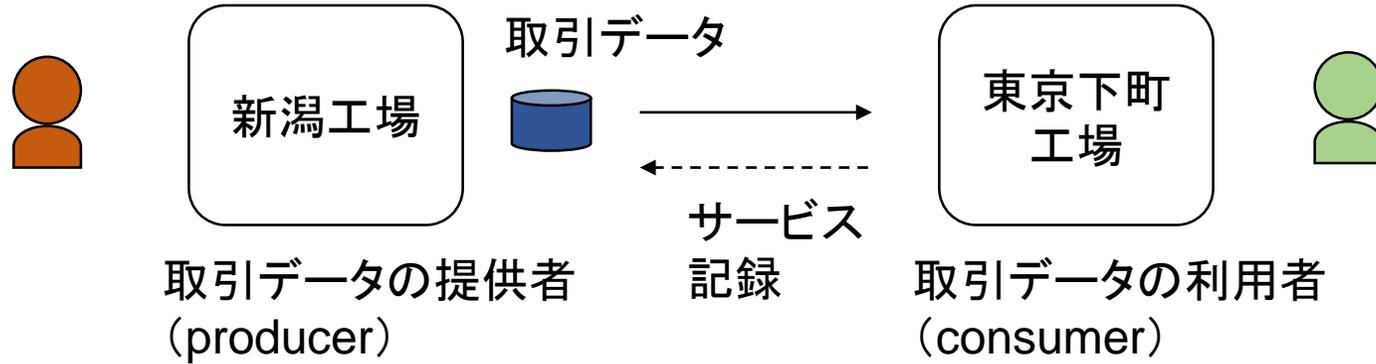
すでに実施したデータ提供や利用履歴の照会や事実認証を行う。



経理担当者、品質担当者、経営戦略室スタッフ

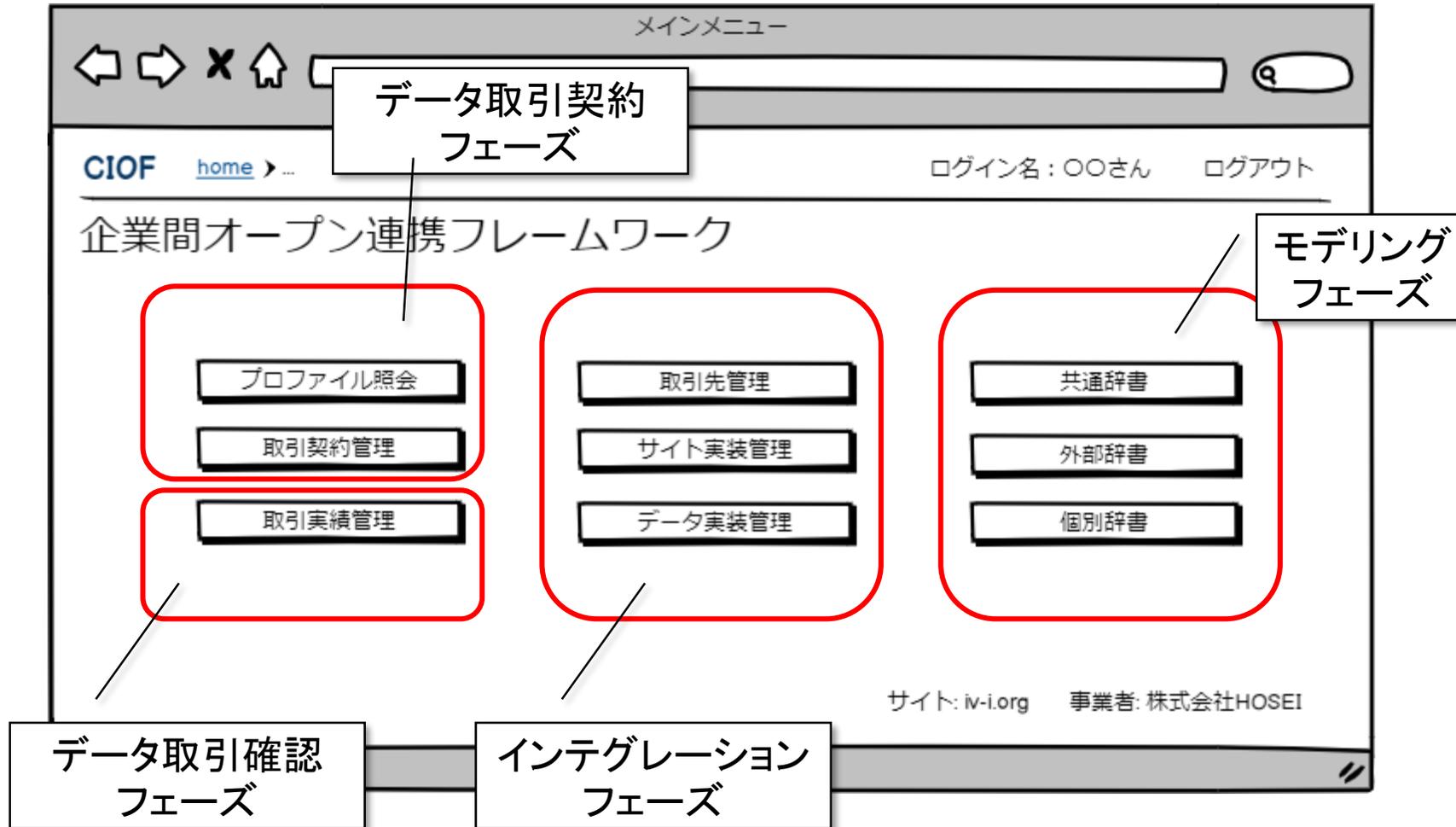


データ取引のロールモデル



取引契約の起案者は、取引データの提供者の場合と利用者の場合があります。





事業者プロフィール照会



登録された事業者を検索します

事業者の内容が表示されます

事業者がもつデータが表示されます

事業者がもつサービスが表示されます

The image shows a sequence of four browser screenshots illustrating the user interface for searching and viewing business profiles. The first screenshot shows a search form with fields for ID and name, and a search button. The second screenshot shows the search results table. The third screenshot shows the 'Data Profile' view for a selected business, including a table of data items. The fourth screenshot shows the 'Service Profile' view, including a table of services. Callout boxes with arrows point to specific elements in each screenshot, explaining their function.

ID	名称	所在地	説明
1234567890	カルフォルニア大学	California	米国西海岸にあるITがとて...
2345678901	東京大学	東京都	東京にあるとても偏差値が...
2345678901			
4567890123			
5678901234			
6789012345			

事業者	ID
法政大学	1234567890
英語名	Hosei University
所在地	東京都
サイト名	東京第一工場

連番	名称	説明	データ型	主キー	必須
01	記録ID	データ識別用	文字列	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
02	センサーID	データ識別用	文字列	<input type="checkbox"/>	<input checked="" type="checkbox"/>
03	測定値	測定された値	数値	<input type="checkbox"/>	<input checked="" type="checkbox"/>

名称	説明	生成	照会	変更	削除
プロセス1-1	プロセスの説明	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
プロセス1-1	プロセスの説明	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

名称	説明	プロセス	記録内容
正常終了	プロセス1の終了	プロセス1	稼働時間(s)
サービス1	サービスの説明	BBHG23	あああいうろう
サービス1	サービスの説明	2341112	あああいうろう

05-01

契約作成中の一覧が表示されます

取引契約管理

契約作成中 取引募集中 データ取引中 通信履歴

起案 申込

ステータス	区分	契約名	データ名	事業者	作成日
起案中	提供	取引契001	電流センサー異常	法政鉄鋼	2020/7/2
起案中	提供	取引契002	環境データ1日分	法政鉄鋼	2020/7/2

募集 05-02

募集 申込

契約に関する通信履歴のリストです

取引契約管理

取引募集中 データ取引中 通信履歴

から まで search

サイト	内容	取引名	日付
第一工場	有償ならいいますよ。	最初の取引	2020/7/11
研究室	よろしくお願います。	異常値送付の件	2020/7/10

取引契約管理

取引募集中 データ取引中 通信履歴

提供 利用 取引終了

ステータス	区分	契約名	データ名	事業者	作成日
取引中	提供	取引契001	電流センサー異常	法政鉄鋼	2020/7/2
取引中	提供	取引契002	環境データ1日分	法政鉄鋼	2020/7/2
取引中	提供	取引契001	電流センサー異常	法政鉄鋼	2020/7/2
取引中	提供	取引契002	環境データ1日分	法政鉄鋼	2020/7/2
取引中	提供	取引契002	環境データ1日分	A社	2020/7/3
取引中	提供	取引契002	環境データ1日分	B社	2020/7/4

取引実施中の一覧が表示されます

募集した契約とその申込みリストです

データ名	募集日	ステータス	事業者	作成日
001 電流センサー異	2020/7/2	申込み	法政鉄鋼	2020/7/2
002 環境データ1日	2020/7/2	申込み	A社	2020/7/3
001 電流センサー異	2020/7/2	申込み	B社	2020/7/4
		申込み	C社	2020/7/4



05-04-03 契約起業フロー

CIOF home > 取引契約管理 > 取引契約閲覧

取引契約編集

契約名 最初の取引 ID A9LXA3RL6ELQ4
説明 CIOFの最初の記念すべき取引です。作成日 2020/9/3
取引先 南カルフォルニア大学 サイト North bdg.
ステータス 申請中 共通辞書 IVI共通辞書v12

提供者 利用者 起案者 応答者 全て表示 全て隠す

▼ データプロフィール

データ 加工温度 説明 加工時の温度データ
データ センサー 説明 センサーデータすべて

追番	名称	対応 実装名	説明	データ型	必須
01	記録ID	01 ID	ユニークキー	文字列	<input checked="" type="checkbox"/>
02	センサーID		センサー識別用	文字列	<input checked="" type="checkbox"/>
03	測定値	03 センサー値	測定された値		

▼ サービスプロフィール (提供)

サービス センサー値の検証
説明 センサー値が正常かどうかを判定する

区分	説明	プロセス
正常終了	検査の異常	検査プロセス
日で繰り返し	毎朝 8 時	カレンダー

▼ サービスプロフィール (利用)

サービ 温度監視
説明 異常であれば警告するしくみです。

区分	説明	プロセス	記録内容	記録
正常終了	プロセスAの終了	プロセスA	稼働時間(e)	<input checked="" type="checkbox"/>
異常終了	プロセスAの異常			<input type="checkbox"/>

契約の概要

取引対象データ

提供する側のサービス

利用する側のサービス

契約の個別の条項

▼ 契約プロフィール

契約文書

契約項目	契約内容	変更
基本契約書	基本設計書123-456.pdf	
個別契約書	株式会社ABC まるまる取引個別契約書.pdf	修正
機密情報規定	機密情報規定-20200202.pdf	削除

契約条項

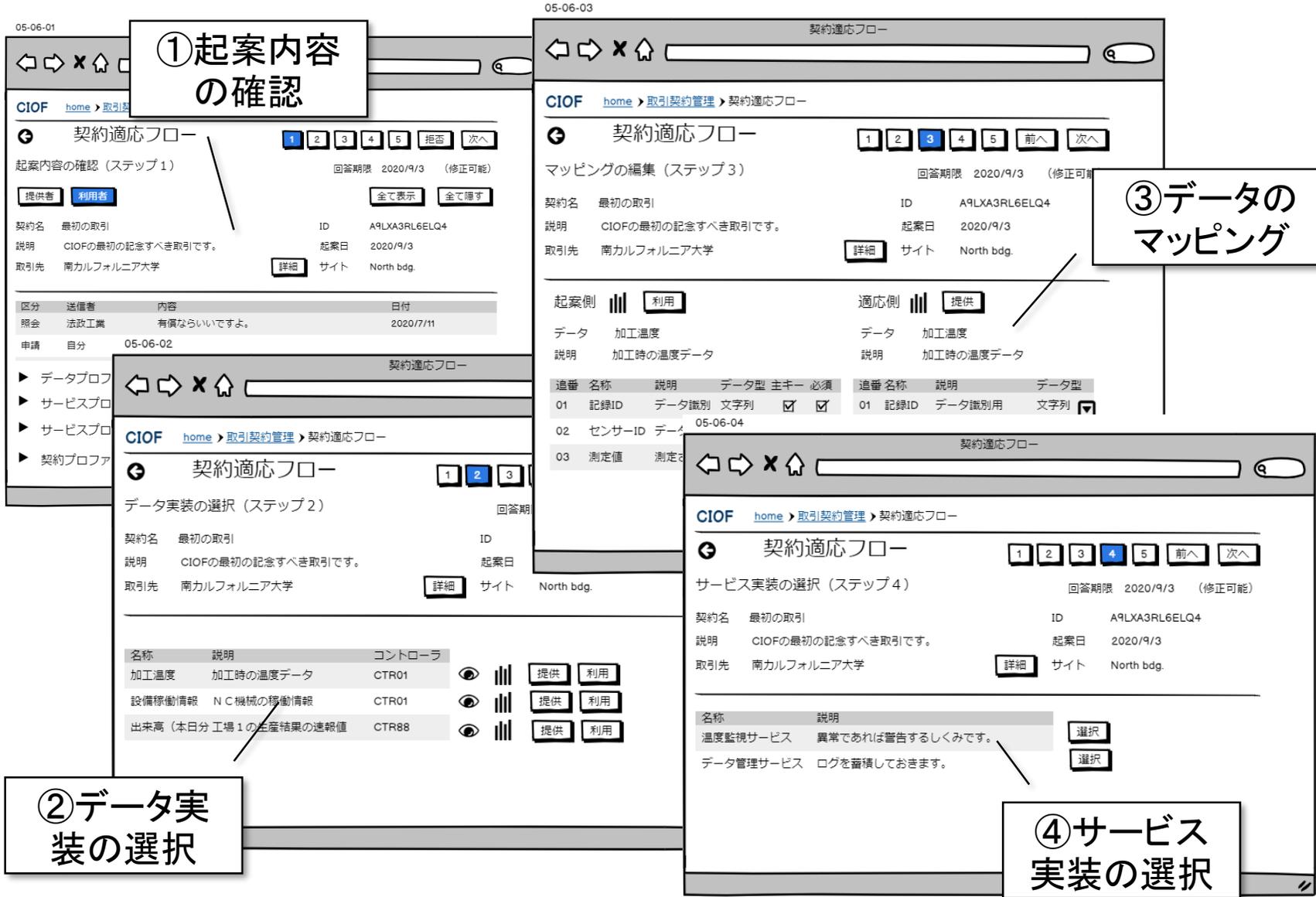
契約項目	契約内容	変更
利用目的	本取引の対象データは、乙が実施する生産管理に利用する。それ以外の目的で利用してはならない。	
利用者	本取引の対象データおよび本取引の対象データを加工することで得られたデータは、該当製品事業部の生産管理部門の人のみが利用し、それ以外の者が利用してはならない。	修正

個別事項

契約項目	契約内容	変更
利用期限	完成品メーカーの納期 (+余裕) まで	
保存場所	生産管理部門が保有するサーバに保存可能	修正
二次利用	NG	修正
第三者提供	NG	修正
削除義務	有 (削除時に報告要)	
利用期限	10年間	追加

メッセージ

取消 再送



1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



レベル0: 必須

- インターネット接続機能
- サービス構成管理機能
- データ構成管理機能
- 取引契約管理機能
- 取引データ送信機能
- 取引データ受信機能
- リクエスト送受信機能

レベル1

- 取引データ削除機能
- 実行ログ管理機能
- 実装ステータス管理機能
- サービス記録管理機能

- イベント監視機能
- 起動リクエスト機能
- カレンダー管理機能
- データベース照会機能
- 取引データトレース機能

レベル2



■ エッジコントローラ機能(チェックリスト)



エッジコントローラ名: _____

開発会社名: _____

インターネット接続機能

実装ステータス管理機能

サービス構成管理機能

データ構成管理機能

取引契約管理機能

サービス記録管理機能

取引データ送信機能

取引データ受信機能

リクエスト送受信機能

取引データ削除機能

データベース照会機能

取引データトレース機能

トリガイベント監視機能

カレンダー管理機能

起動リクエスト機能

実行ログ管理機能



- インターネット接続機能

エッジコントローラは、連携ターミナルとの間で通信プロトコルはHTTPとし、連携ターミナルが提供するREST/APIを利用可能とする。通信におけるセキュリティ設定として、あらかじめ連携マネージャが発行したIDおよびAPIキーによるベーシック認証に対応する。

- 実行ログ管理機能

エッジコントローラは、自分自身の起動および停止のログをローカルに記録として保存する。また、配下のサービス実装について、その起動および停止をログとしてローカルに記録する。また、配下のイベント実装の発動や連携ターミナルとの通信を記録する。



- 取引契約管理機能

5.取引契約の取得

エッジコントローラは、起動時または必要なタイミングで、連携ターミナルから、その時点で登録されている取引契約の内容の中で、配下のサービス実装が関与するものをすべて取得する。それぞれの取引契約のIDと、対応するサービス実装、データ実装、プロセス実装、そしてイベント実装のIDを対応づける。



• サービス構成管理機能

1. サービス実装の取得

エッジコントローラは、連携ターミナルから受け取ったサービス実装の内容に基づき、配下にある実装されたすべてのソフトウェアモジュールを、サービス実装として識別する。識別のために、エッジコントローラが独自に管理する内部IDと、CIOFが設定したサービス実装IDを1対1に対応付ける。

• データ構成管理機能

3. データ実装の取得

エッジコントローラは、連携ターミナルから受け取ったデータ実装の内容に基づき、配下にあるサービス実装が提供または利用するすべてのデータ構成モデルの実装を、データ実装として識別する。エッジコントローラの内部にデータベースをもつ場合は、データベースにおけるテーブル名に対応する内部IDと、CIOFが設定したデータ実装IDを1対1に対応付ける。



• 取引データ受信機能

7.取引データの取得

エッジコントローラは、連携ターミナルから受け取った取引データの内容を、取引契約で指定されたサービス実装に、ただちに送信する。サービス実装からの応答がない場合は、一定の期間、リトライする。サービス実装から正常応答があった場合は、サービス記録(送信)として連携ターミナルへ報告する。

• 取引データ送信機能

8.取引データの送信

エッジコントローラは、サービス実装からCIOF向けのデータを受信した場合、契約情報を手掛かりに取引データを構成し、連携ターミナルに対して、ただちに送信する。送信した取引データがもつ内部のIDに対して、連携ターミナルからの返信にある取引データIDを対応づけて管理する。



• リクエスト送受信機能

9.リクエストパラメータの取得

10.リクエストパラメータの送信

エッジコントローラは、連携ターミナルから受け取ったリクエストパラメータの内容を、取引契約で指定されたサービス実装に、ただちに送信する。サービス実装からCIOF向けのリクエストパラメータを受信した場合、契約情報を手掛かりに取引パラメータを構成し、連携ターミナルに対して、ただちに送信する。



- 取引データ削除機能

エッジコントローラは、連携ターミナルから取引データに対するデータ削除リクエストをリクエストパラメータとして受け取った場合に、関係するサービス実装に対してその依頼を通知し、すべての削除が完了したらその結果を報告する。

• 実装ステータス管理機能

2. サービス実装の状態通知

4. データ実装の状態通知

エッジコントローラは、配下のサービス実装、プロセス実装、イベント実装、およびデータ実装のステータスを連携ターミナルに通知する。連携マネージャで設定した内容が、具体的な実装の実体と異なる場合などは、その事実を連携ターミナルに通知する。

• サービス記録管理機能

11. サービス記録の通知

エッジコントローラは、配下のサービス実装から受信したイベント情報を、取引契約と対応づけて管理する。受信したイベント情報が、取引中の取引契約に対応するものであり、かつ、記録の報告義務がある場合は、連携ターミナルに対して、サービス記録(利用)として、ただちに送信する。



- イベント監視機能

エッジコントローラは、取引契約の中で定義されたイベント実装の実行を監視する。イベント実装が他のプロセス実装でトリガ実装として指定されている場合は、そのプロセス実装に対する起動リクエストを発行する。

- 起動リクエスト機能

エッジコントローラは、プロセス実装に対する起動リクエストが発行された場合に、該当するサービス実装に対して、対象となるプロセス実装の起動を依頼し、実際に実行されたかを監視する。



• カレンダー管理機能

6. カレンダーの取得

エッジコントローラは、連携ターミナルから、カレンダー情報を取得し、カレンダー機能をもつサービス実装がもつプロセス実装とイベント実装を生成する。カレンダー機能をもつサービス実装は、カレンダー情報に対応してイベント実装を発動する。

- データベース照会機能

エッジコントローラは、データベース機能を持つサービス実装に対して、データ実装に対応する取引データの提供を依頼する。リクエストパラメータの指定がある場合は、対象レコードの限定を行う。データベースでは、連携ターミナルから受け取った取引データIDと、データベース内部のレコードIDとを関係づけて管理する。

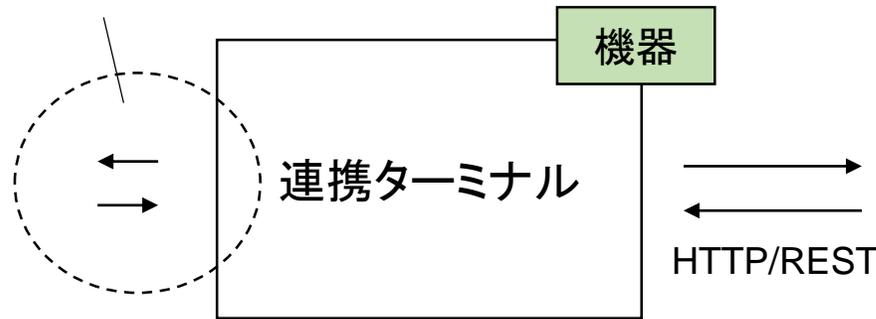
- 取引データトレース機能

エッジコントローラは、データベース機能をもつサービス実装が取引データを保存した場合に、サービス記録(保存)として、連携ターミナルに通知する。また、該当する取引データが、他の取引契約で提供された場合や、外部のサービスで利用された場合に、サービス記録(読出)として、連携ターミナルに通知する。

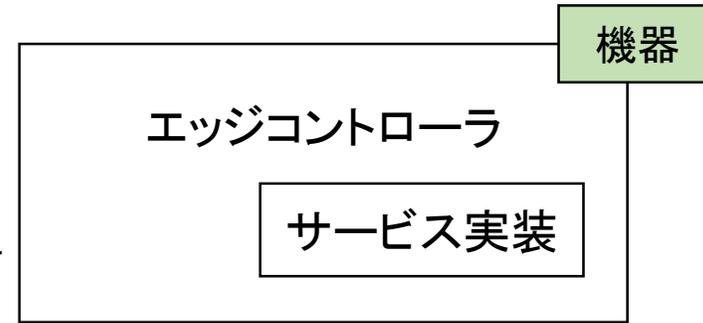


エッジ側のネットワーク構成

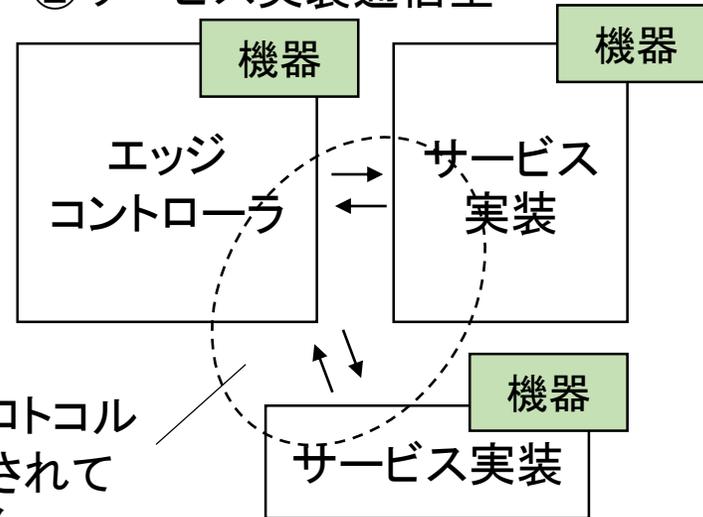
ファイアウォール経由で
外部インターネット接続



① サービス実装内包型



② サービス実装通信型

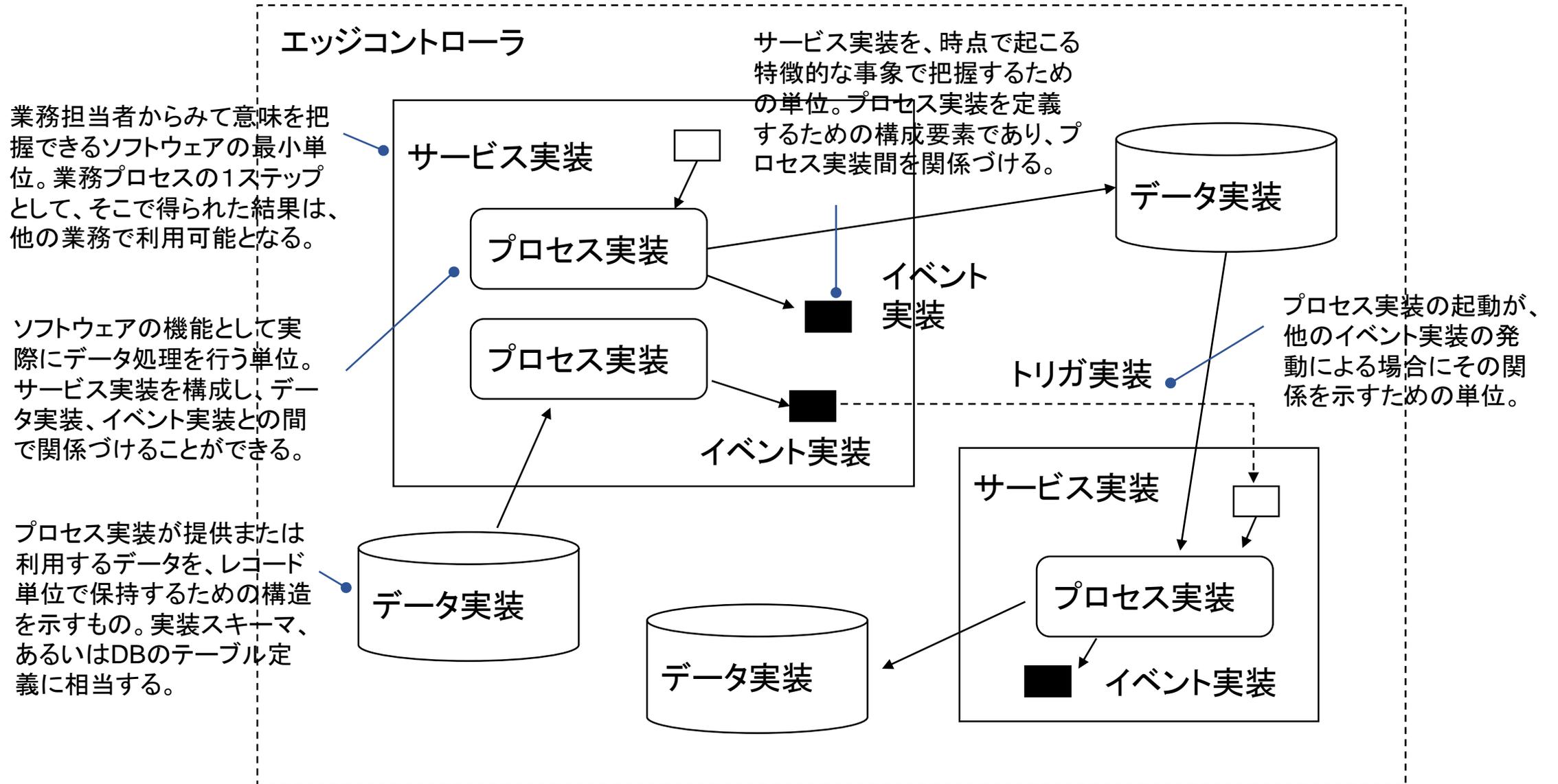


HTTP/REST

注) 連携ターミナルと同じハードウェアにエッジコントローラが実装される場合もあります。ただし、通信方式は同じです。

通信プロトコルは規定されていません。





1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



連携ターミナルが提供するAPI



No	説明	HTTP	API name
1	サービス実装の取得	GET	service_implementations
2	サービス実装の状態通知	PUT	service_implementations
3	データ実装の取得	GET	data_implementations
4	データ実装の状態通知	PUT	data_implementations
5	取引契約の取得	GET	trade_contracts
6	カレンダーの取得	GET	calendars
7	取引データの取得	GET	messages
8	取引データの送信	POST	messages
9	リクエストパラメータの取得	GET	requests
10	リクエストパラメータの送信	POST	requests
11	サービス記録の通知	POST	service_record



Overview

API一覧

サービス実装

- 1. サービス実装の取得
- 2. サービス実装の状態通知

データ実装

- 3. データ実装の取得
- 4. データ実装の状態通知
- 5. 取引契約の取得
- 6. カレンダの取得

取引データ

API Reference for Edge Controller Unit(Ver 2.10)

本ドキュメントでは連携ターミナルで使用可能なエッジコントローラ向けのAPI仕様を定義します。

変更履歴

Ver.	変更日	変更内容
1.00	2020/01/19	初版リリース
1.01	2020/01/30	履歴管理API仕様変更
1.02	2020/02/17	履歴管理API仕様変更
2.00	2020/08/26	CIOF仕様一新による大幅修正
2.10	2020/09/17	リクエストパラメータに関する仕様変更

GET /hct/api/v2/service_implementations

Requests example 1

Headers

```
Content-Type: application/json
Authorization: Bearer xxxxxxxxxxxxxx
```

Responses 200

Headers

```
Content-Type: application/json
```

Body

```
[
  {
    "id": "10501",
    "local_id": "x5VrQsPfiqrzc2J",
    "name": "環境情報取得サービス",
    "description": "環境データを取得する",
    "device_id": [
      "device001"
    ],
    "process_implementations": [
      {

```

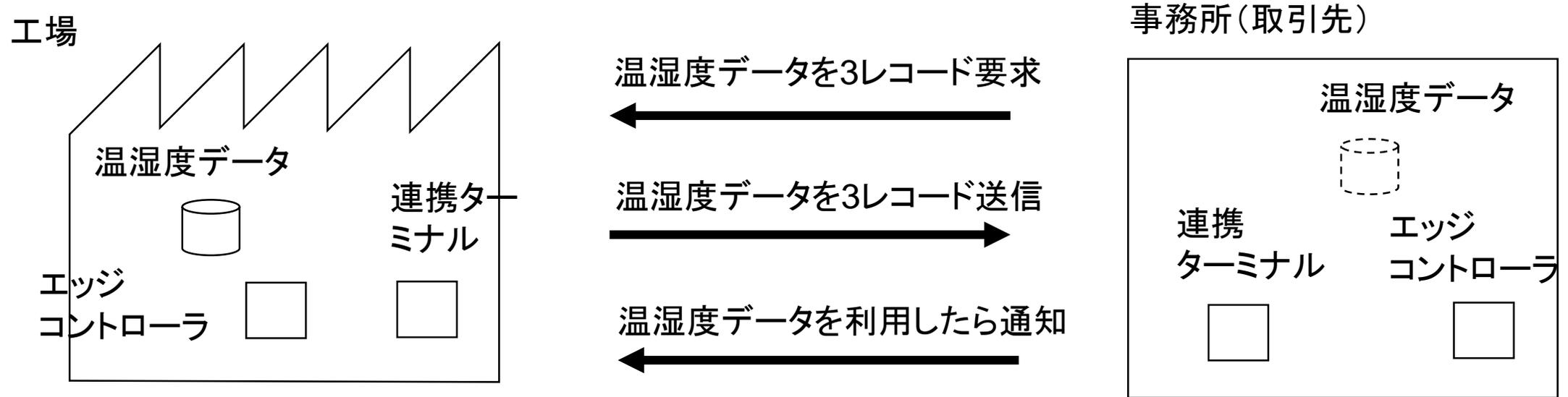
GET /hct/api/v2/service_implementations 1. サービス実装の取得



本仕様書のパラメータが想定するストーリー



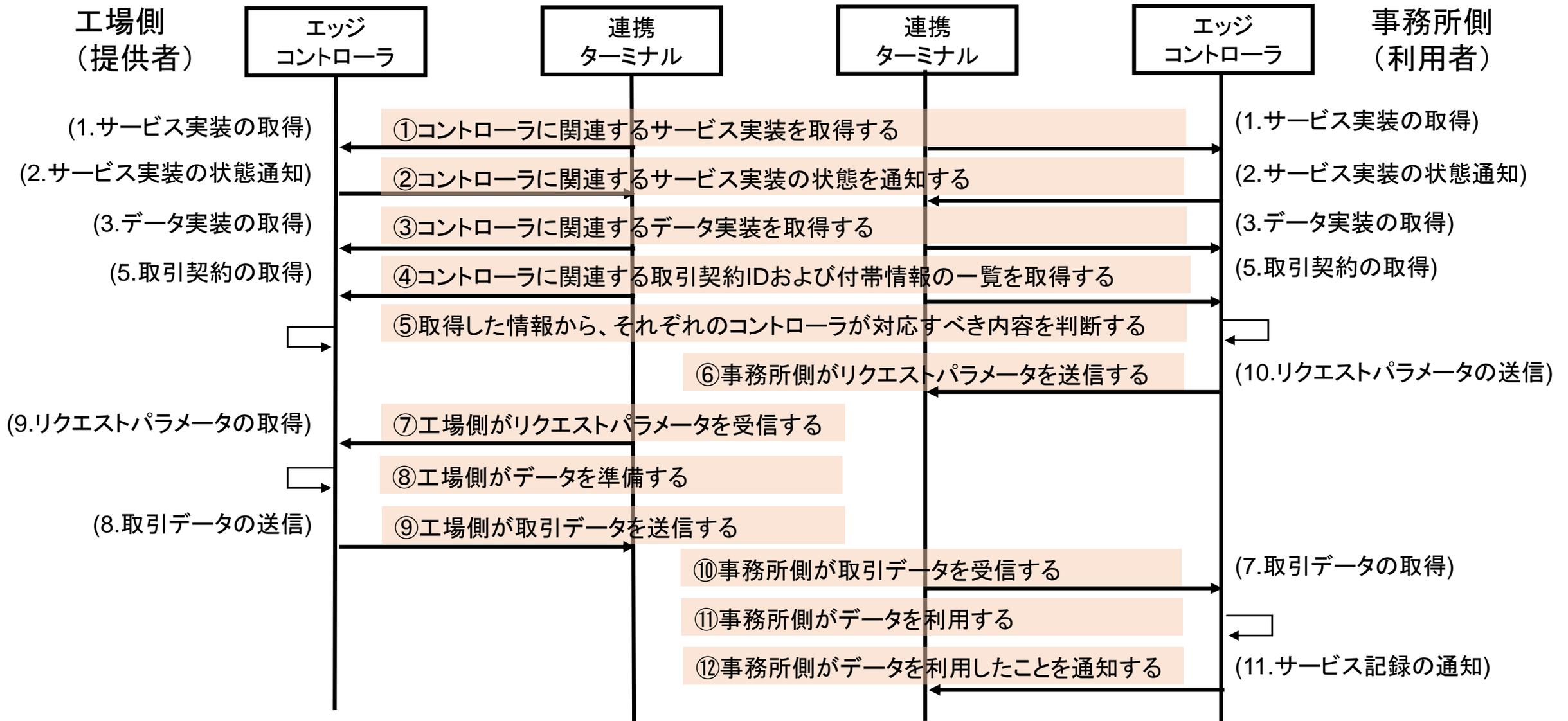
1. 事務所側から工場側(取引先のため別事業者を想定)へ、現在の温湿度データを3レコード要求する。
2. 工場側から事務所側へ、温湿度データを3レコード送信する。
3. 事務所側は、受け取った温湿度データを利用したら、それを知らせる。



※サンプルストーリーがPullタイプの取引のため、6.カレンダーの取得APIを使用していません。



ストーリーに沿ったAPIの利用手順

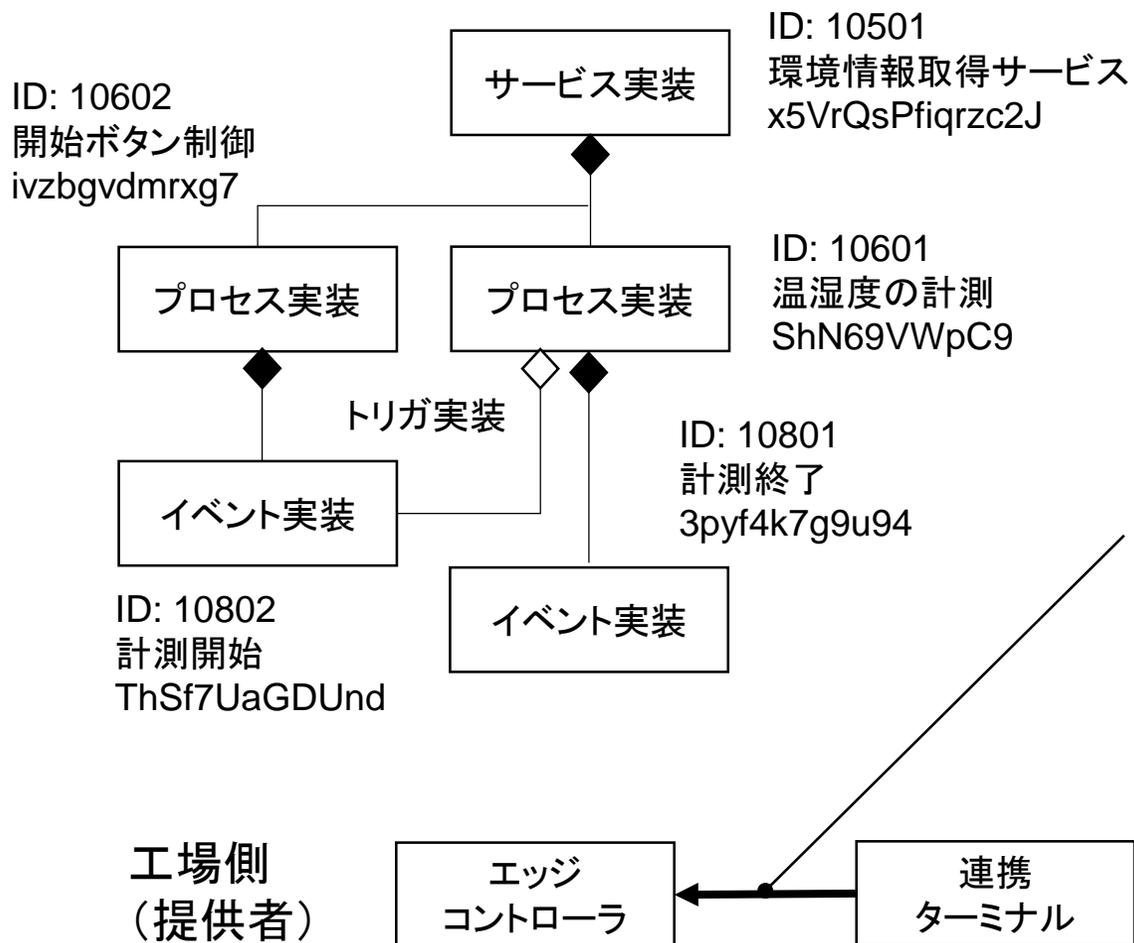


ステップ1 (1.サービス実装の取得)



コントローラに関連するサービス実装を取得する

GET /hct/api/v2/service_implementations



```
[
  {
    "id": "10501",
    "local_id": "x5VrQsPfiqrzc2J",
    "name": "環境情報取得サービス",
    "description": "環境データを取得する",
    "device_id": [
      "device001"
    ],
    "process_implementations": [
      {
        "id": "10601",
        "local_id": "ShN69VWpC9",
        "name": "温湿度の計測",
        "description": "温度を計測して、値を応答する",
        "process_operation_implementations": [
          {
            "id": "10701",
            "index": "1",
            "description": "計測値を生成する",
            "data_implementation_id": "10901",
            "operation_type": "create"
          }
        ]
      }
    ]
  }
]
```

```
"event_implementations": [
  {
    "id": "10801",
    "local_id": "3pyf4k7g9u94",
    "event_type": "monitor",
    "name": "計測終了",
    "description": "計測を終了するイベント"
  },
  {
    "id": "10802",
    "local_id": "ThSf7UaGDUnd",
    "event_type": "trigger",
    "name": "計測開始",
    "description": "計測を開始するトリガイベント"
  }
]
}
```

トリガ実装は、event_type="trigger"として設定される。

このプロセスによってデータ(計測値)が生成されることを示す。

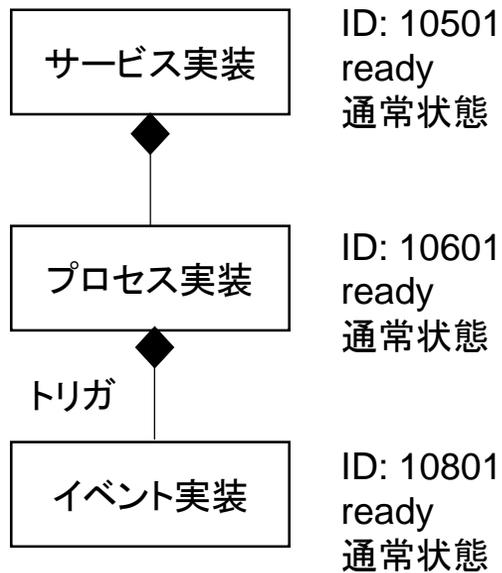


ステップ2(2.サービス実装の状態通知)



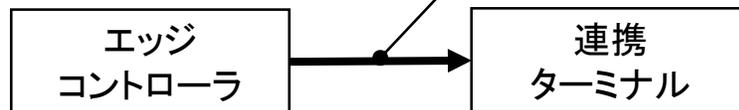
コントローラに関連するサービス実装の状態を通知する

PUT /hct/api/v2/service_implementations



```
[
  {
    "id": "10501",
    "local_id": "x5VrQsPfiqrzc2J",
    "status": "ready",
    "remarks": "通常状態",
    "process_implementations": [
      {
        "id": "10601",
        "local_id": "ShN69VWpC9",
        "status": "ready",
        "remarks": "通常状態",
        "event_implementations": [
          {
            "id": "10801",
            "local_id": "ThSf7UaGDUnd",
            "status": "ready",
            "remarks": "通常状態"
          }
        ]
      }
    ]
  }
]
```

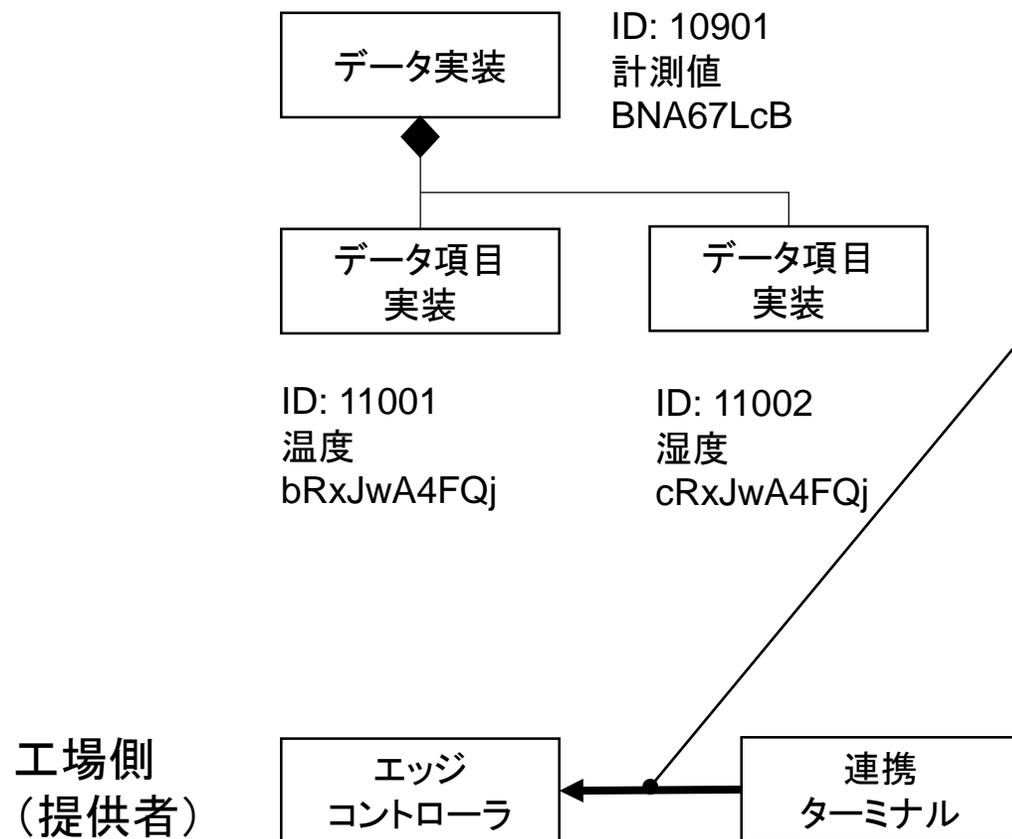
工場側
(提供者)



ステップ3(3.データ実装の取得)



コントローラに関連するデータ実装を取得する



GET /hct/api/v2/data_implementations

```
[
  {
    "id": "10901",
    "local_id": "BNA67LcB",
    "name": "計測値",
    "description": "計測された値を示す",
    "service_implementation_id": "10501",
    "data_property_implementations": [
      {
        "id": "11001",
        "index": "1",
        "local_id": "bRxJwA4FQj",
        "name": "温度",
        "description": "温度の値を摂氏で示す",
        "data_type": "float",
        "is_primary_key": false,
        "is_required": true,
        "default_value": "null"
      },
      {
        "id": "11002",
        "index": "2",
        "local_id": "cRxJwA4FQj",
        "name": "湿度",
        "description": "湿度の値を相対湿度で示す",
        "data_type": "float",
        "is_primary_key": false,
        "is_required": true,
        "default_value": "null"
      }
    ]
  }
]
```



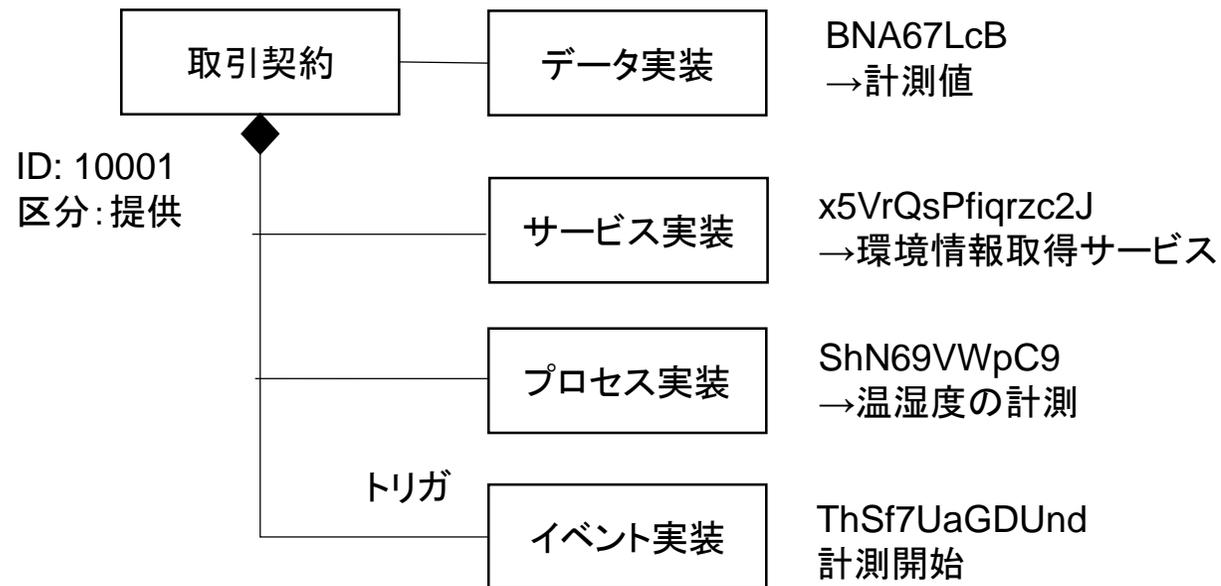
ステップ4(5.取引契約の取得)



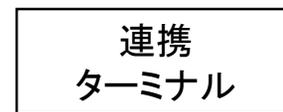
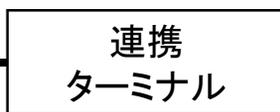
コントローラに関連する取引契約IDおよび付帯情報の一覧を取得する

GET /hct/api/v2/trade_contracts

```
[  
  {  
    "id": "10001",  
    "contract_type": "produce",  
    "data_implementation_local_id": "BNA67LcB",  
    "service_implementation_local_id": "x5VrQsPfiqrzc2J",  
    "process_implementation_local_id": "ShN69VWpC9",  
    "event_implementation_local_id": [  
      "ThSf7UaGDUnd"  
    ],  
    "start_datetime": "2019-06-12T09:10:06.922Z",  
    "end_datetime": "2021-06-12T09:10:06.922Z"  
  }  
]
```



工場側
(提供者)



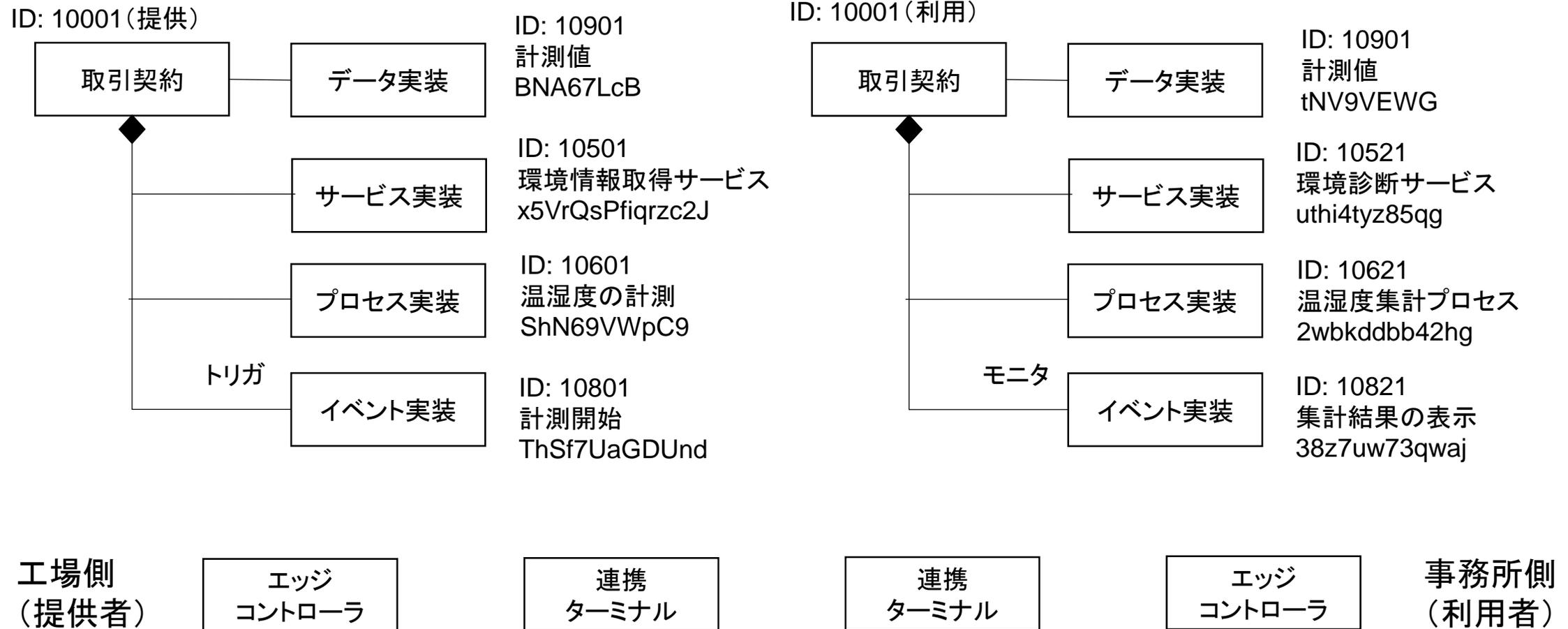
事務所側
(利用者)



ステップ5



取得した情報から、それぞれのコントローラが対応すべき内容を判断する



ステップ6 (10.リクエストパラメータの送信)



事務所側がリクエストパラメータを送信する

“最新の3レコードをください”という部分は、サービス実装が理解可能な文字列として設定します。たとえば、SQLのwhere句に相当する文字列を設定します。

POST /hct/api/v2/requests

取引契約

契約ID: 10001
パラメータID: 20001
リクエスト区分: 生成

```
{  
  "contract_id": "10001",  
  "request_type": "create",  
  "created_at": "2016-06-12T09:10:06.922Z",  
  "response_limit": "2019-06-12T09:10:06.922Z",  
  "condition": "最新の3レコードをください"  
}
```

```
{  
  "contract_id": "10001",  
  "request_type": "create",  
  "created_at": "2016-06-12T09:10:06.922Z",  
  "response_limit": "2019-06-12T09:10:06.922Z",  
  "condition": "最新の3レコードをください",  
  "request_parameter_id": "20001"  
}
```

工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

エッジ
コントローラ

事務所側
(利用者)



ステップ7 (9.リクエストパラメータの取得)



工場側がリクエストパラメータを受信する

GET /hct/api/v2/requests

```
[
  {
    "contract_id": "10001",
    "request_type": "create",
    "created_at": "2016-06-12T09:10:06.922Z",
    "response_limit": "2019-06-12T09:10:06.922Z",
    "condition": "最新の3レコードをください",
    "request_parameter_id": "20001"
  }
]
```

取引契約

契約ID: 10001
パラメータID: 20001
リクエスト区分: 生成

工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

エッジ
コントローラ

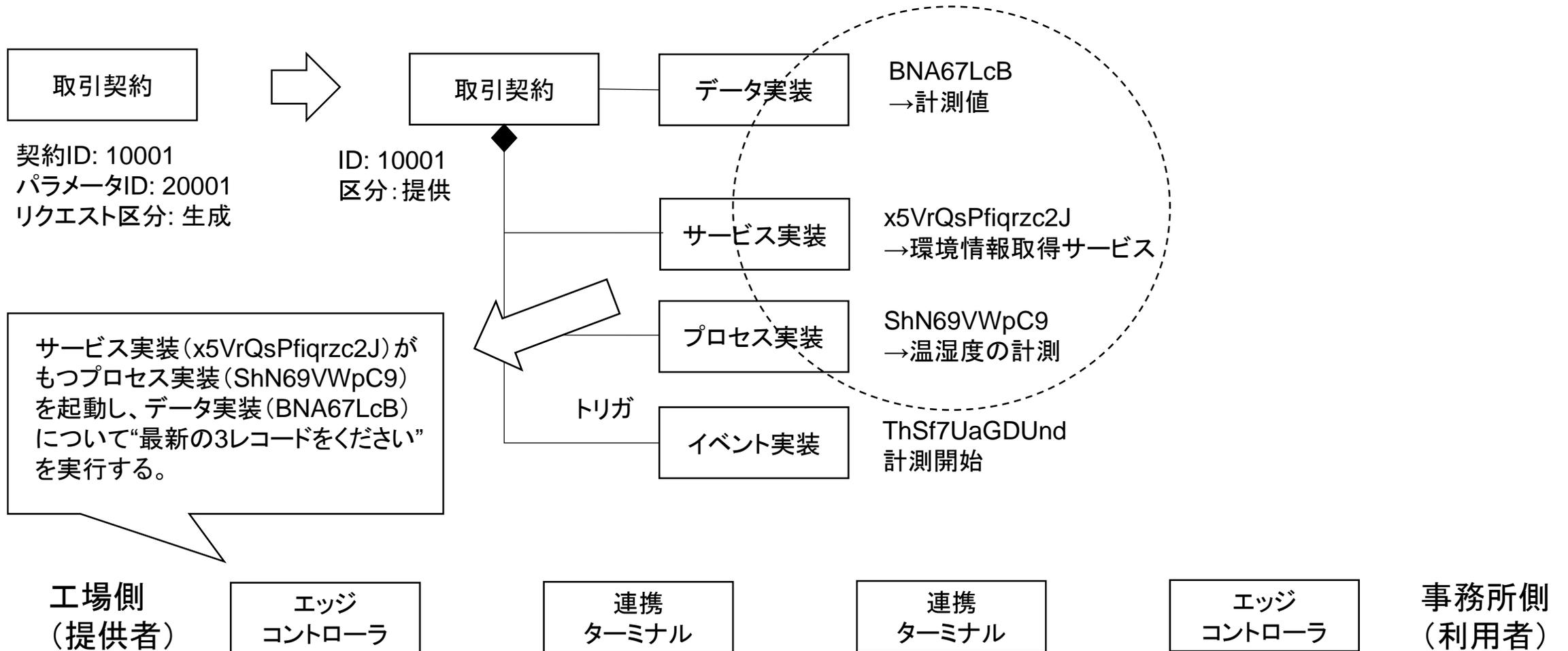
事務所側
(利用者)



ステップ8



工場側がデータを準備する



ステップ9 (8.取引データの送信)



工場側が取引データを送信する

POST /hct/api/v2/messages

```
{
  "request_parameter_id": "20001",
  "trade_contract_id": "10001",
  "contents": [
    [
      "25",
      "85"
    ],
    [
      "27",
      "80"
    ],
    [
      "21",
      "76"
    ]
  ]
}
```

```
{
  "id": "12345678"
}
```

取引データID: 12345678
リクエストパラメータID: 20001
取引契約ID: 10001

ID	属性1	属性2
1	25	85
2	27	80
3	21	76

工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

エッジ
コントローラ

事務所側
(利用者)



ステップ10 (7.取引データの取得)



事務所側が取引データを受信する

辞書サーバよりヘッダを追加することができます。

取引データID: 12345678
リクエストパラメータID: 20001
取引契約ID: 10001

ID	温度	湿度
1	25	85
2	27	80
3	21	76

GET /hct/api/v2/messages

```
[  
{  
  "id": "12345678",  
  "domain": "81",  
  "request_parameter_id": "20001",  
  "trade_contract_id": "10001",  
  "headers": [  
    "温度",  
    "湿度"  
  ],  
}
```

```
"contents": [  
  [  
    "25",  
    "85"  
  ],  
  [  
    "27",  
    "80"  
  ],  
  [  
    "21",  
    "76"  
  ]  
]  
}
```

工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

エッジ
コントローラ

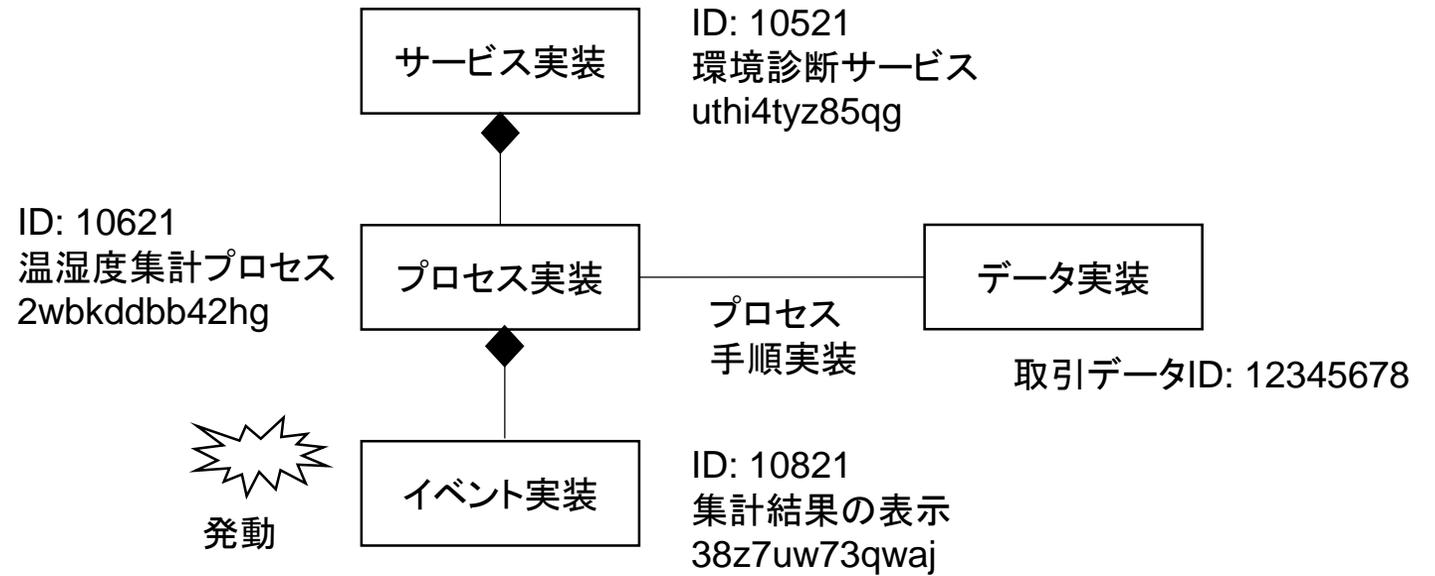
事務所側
(利用者)



ステップ11



事務所側がデータを利用する



工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

エッジ
コントローラ

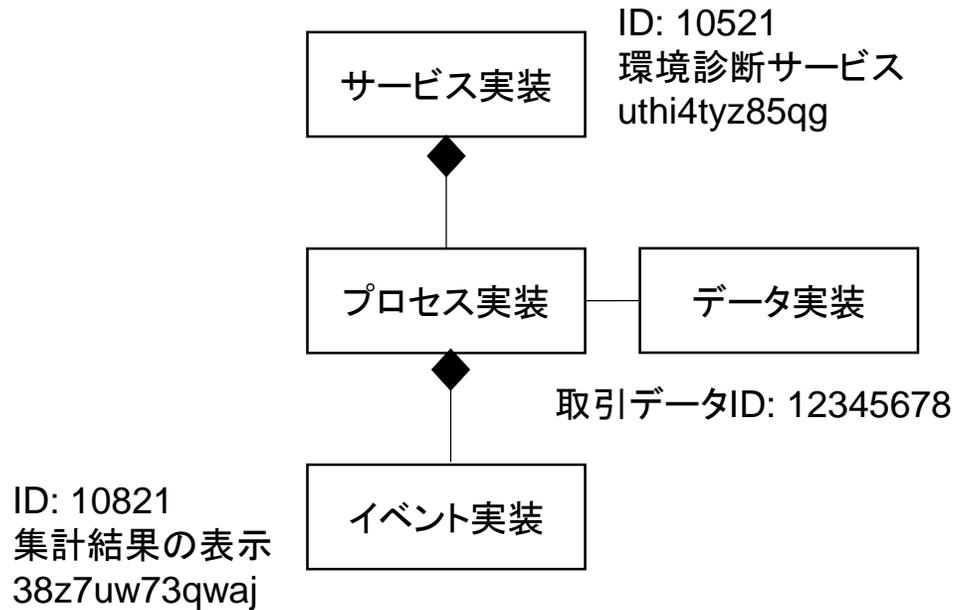
事務所側
(利用者)



ステップ12 (11.サービス記録の通知)



事務所側がデータを利用したことを通知する



POST /hct/api/v2/service_record

```
{  
  "service_implementation_id": "10521",  
  "event_implementation_id": "10821",  
  "data_id": "12345678",  
  "event_type": "use",  
  "timestamp": "2020-09-17T09:10:06.922Z",  
  "result": "0",  
  "note": "値を利用し、端末に表示しました。"  
}
```

```
{  
  "id": "980001",  
  "service_implementation_id": "10521",  
  "event_implementation_id": "10821",  
  "data_id": "12345678",  
  "event_type": "create",  
  "timestamp": "2020-09-17T09:10:06.922Z",  
  "result": "0",  
  "note": "値を利用し、端末に表示しました。"  
}
```

工場側
(提供者)

エッジ
コントローラ

連携
ターミナル

連携
ターミナル

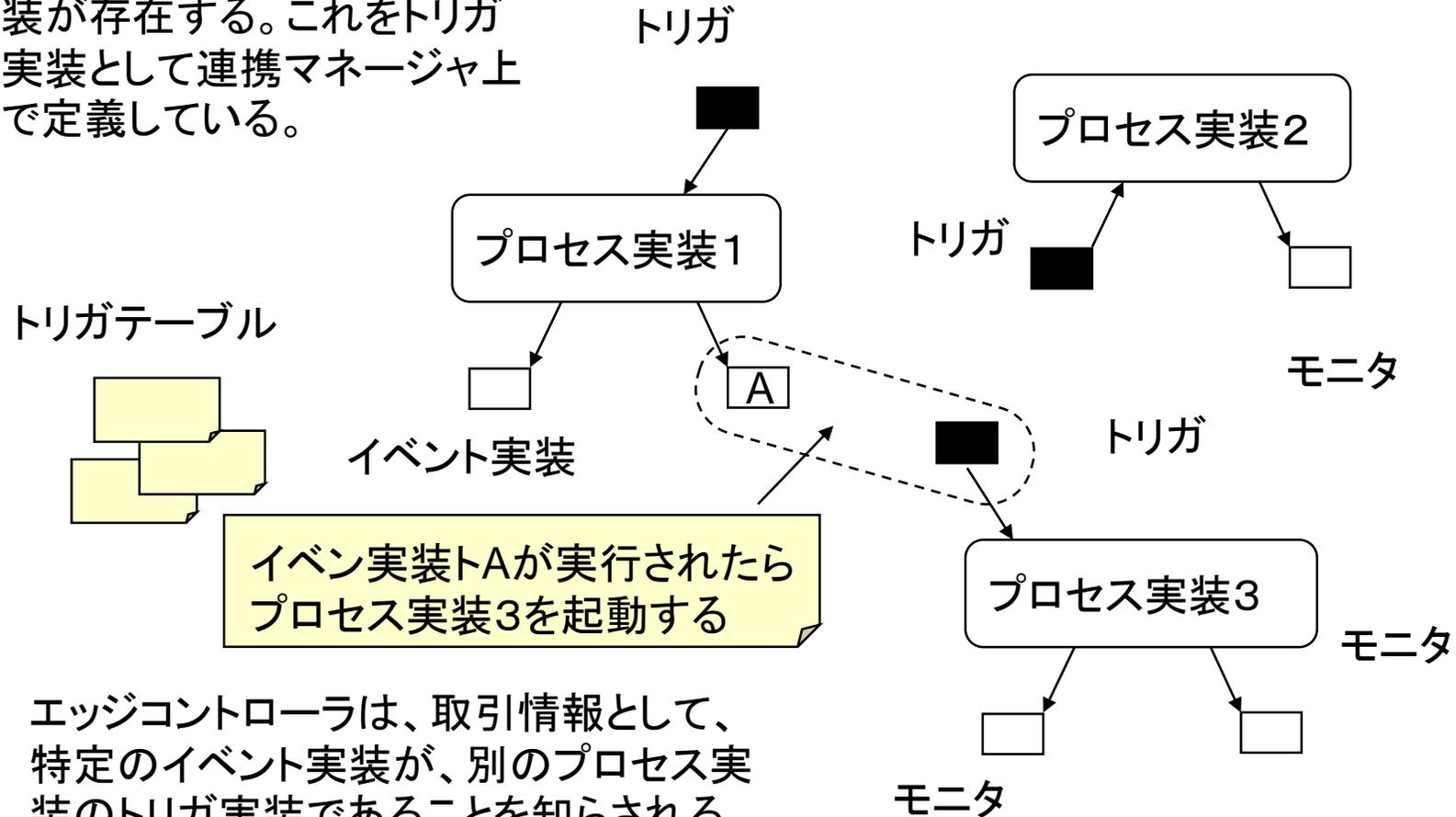
エッジ
コントローラ

事務所側
(利用者)



イベント実装によるプロセスの起動

プロセス実装には起動するためのトリガとなるイベント実装が存在する。これをトリガ実装として連携マネージャ上で定義している。



エッジコントローラは、取引情報として、特定のイベント実装が、別のプロセス実装のトリガ実装であることを知らされる。



カレンダーの構造



項目名	英語名	説明
カレンダーID	id	定義したカレンダーを識別するID
名称	name	カレンダーの実装名称
基準日	start_date	カレンダーイベントを実行するにあたっての基準となる日時。
曜日	days_of_week	週の曜日を要素とするリスト
間隔	interval	間隔区分に対応する数値
間隔区分	interval_type	間隔の単位
終了日	end_date	イベントを監視を終了する日時
回数	number_of_occurrences	イベントを実行する回数。この回数おこなったら終了する。
タイムゾーン	recurrence_time_zone	該当するタイムゾーン

月(Monday)
火(Tuesday)
水(Wednesday)
木(Thursday)
金(Friday)
土(Saturday)
日(Sunday)

サービス実装
・・・"Calendar"

プロセス実装
・・・カレンダーID

プロセス実装
・・・"CalendarExecution"

分ごと(minute)
時間ごと(hour)
日ごと(day)
週間ごと(week)
カ月ごと(month)
年ごと(year)

× カレンダー表示ダイアログ 完了

名称

基準日 時 分

月 火 水 木 金 土 日

間隔 日ごと 週間ごと
 時間ごと カ月ごと
 分ごと 年ごと

終了 なし
 終了日
 回数 回

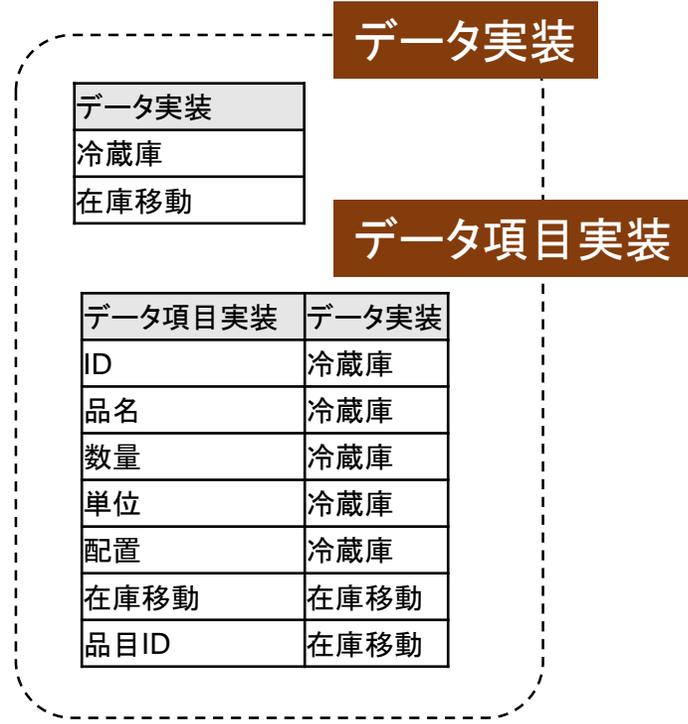


データベースとの対応



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶		2L	冷蔵棚
10002	BLACK COFFE		1L	冷蔵棚
10003	1日分の野菜		1L	冷蔵棚
10004	キムチ		1パック	冷蔵棚
10005	OICOS		4個	冷蔵棚
10006	絹豆腐		2パック	冷蔵棚
10007	オクラ昆布		3パック	冷蔵棚
10008	ウィダーゼリー		4個	冷蔵棚
10009	缶ビール		4本	冷蔵棚
10010	ロースハム		2パック	冷蔵棚
10011	生ハム		2パック	チルド
10012	ポークウィンナー		2袋	チルド
10013	豚肩ロース		744g	チルド
10014	鶏モモ肉		817g	チルド



テーブル名: 在庫移動

	在庫移動	品目ID	品名	単位	区域	数量	移動日時	
10015	豚切り落とし							
10016	卵	10002	10039	カレー粉	パック	冷蔵棚	-2	2020/7/13
10017	バター	10003	10038	ジャガイモ	個	冷凍庫	-2	2020/7/13
10018	マヨネーズ	10010	10027	玉ねぎ	個	野菜室	-1	2020/7/13
10019	ケチャップ	10011	10013	豚肩ロース	g	チルド	-300	2020/7/13
		10012	10003	1日分の野菜	L	冷蔵棚	1	2020/7/13



取引データとレコードの関係



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶	2	L	冷蔵棚
10002	BLACK COFFE	1	L	冷蔵棚
10003	1日分の野菜	1	L	冷蔵棚
10004	キムチ	1	パック	冷蔵棚
10005	OICOS	4	個	冷蔵棚
10006	絹豆腐	2	パック	冷蔵棚
10007	オクラ昆布	3	パック	冷蔵棚
10008	ウィダーゼリー	4	個	冷蔵棚
10009	缶ビール	4	本	冷蔵棚
10010	ロースハム	2	パック	冷蔵棚
10011	生ハム	2	パック	チルド
10012	ポークウィンナー	2	袋	チルド
10013	豚肩ロース	744	g	チルド
10014	鶏モモ肉	817	g	チルド
10015	豚切り落とし	748	g	チルド
10016	卵	14	個	冷蔵トビラ
10017	バター	1	パック	冷蔵トビラ
10018	マヨネーズ	1	本	冷蔵トビラ
10019	ケチャップ	1	本	冷蔵トビラ

- D0001
- D0003
- D0001
- D0002
- D0002

取引データ管理テーブル

取引データID	データ実装	ID	サービス実装	プロセス実装	事実区分
D0001	冷蔵庫	10001	DBMS	データ保存	保管
D0001	冷蔵庫	10002	DBMS	データ保存	保管
D0001	冷蔵庫	10003	DBMS	データ保存	保管
D0002	冷蔵庫	10004	DBMS	データ保存	保管
D0002	冷蔵庫	10005	DBMS	データ保存	保管
D0003	冷蔵庫	10002	DBMS	データ保存	変更
D0002	冷蔵庫	10003	DBMS	データ読出	読出
D0002	冷蔵庫	10003	DBMS	データ読出	読出



取引データとレコードの関係



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶		2L	冷蔵庫
10002	BLACK COFFE		1L	冷蔵庫
10003	1日分の野菜		1L	冷蔵庫
10004	キムチ		1パック	冷蔵庫
10005	DICOS		4個	冷蔵庫
10006	絹豆腐		2パック	冷蔵庫
10007	オクラ昆布		3パック	冷蔵庫
10008	ウィダーゼリー		4個	冷蔵庫
10009	缶ビール		4本	冷蔵庫
10010	ロースハム		2パック	冷蔵庫
10011	生ハム		2パック	チルド
10012	ポークウィンナー		2袋	チルド
10013	豚肩ロース		744g	チルド
10014	鶏モモ肉		817g	チルド
10015	豚切り落とし		748g	チルド
10016	卵		14個	冷蔵トビラ
10017	バター		1パック	冷蔵トビラ
10018	マヨネーズ		1本	冷蔵トビラ
10019	ケチャップ		1本	冷蔵トビラ

取引1

D0001
D0001
D0001

取引データD0001として、10001, 10002, 10003を取得し保管

取引データ管理テーブル

取引データID	データ実装	ID	サービス実装	プロセス実装	事実区分
D0001	冷蔵庫	10001	DBMS	データ保存	保管
D0001	冷蔵庫	10002	DBMS	データ保存	保管
D0001	冷蔵庫	10003	DBMS	データ保存	保管



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶		2L	冷蔵棚
10002	BLACK COFFE		1L	冷蔵棚
10003	1日分の野菜		1L	冷蔵棚
10004	キムチ		1パック	冷蔵棚
10005	OICOS		4個	冷蔵棚
10006	絹豆腐		2パック	冷蔵棚
10007	オクラ昆布		3パック	冷蔵棚
10008	ウィダーゼリー		4個	冷蔵棚
10009	缶ビール		4本	冷蔵棚
10010	ロースハム		2パック	冷蔵棚
10011	生ハム		2パック	チルド
10012	ポークウィンナー		2袋	チルド
10013	豚肩ロース		744g	チルド
10014	鶏モモ肉		817g	チルド
10015	豚切り落とし		748g	チルド
10016	卵		14個	冷蔵トビラ
10017	バター		1パック	冷蔵トビラ
10018	マヨネーズ		1本	冷蔵トビラ
10019	ケチャップ		1本	冷蔵トビラ

取引2

- D0001
- D0001
- D0001
- D0002
- D0002

取引データD0002として、10004, 10005を取得し保管

取引データ管理テーブル

取引データID	データ実装	ID	サービス実装	プロセス実装	事実区分
D0002	冷蔵庫	10004	DBMS	データ保存	保管
D0002	冷蔵庫	10005	DBMS	データ保存	保管



取引データとレコードの関係



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶		2L	冷蔵棚
10002	BLACK COFFE		-9L	冷蔵棚
10003	1日分の野菜		1L	冷蔵棚
10004	キムチ		1パック	冷蔵棚
10005	OICOS		4個	冷蔵棚
10006	絹豆腐		2パック	冷蔵棚
10007	オクラ昆布		3パック	冷蔵棚
10008	ウィダーゼリー		4個	冷蔵棚
10009	缶ビール		4本	冷蔵棚
10010	ロースハム		2パック	冷蔵棚
10011	生ハム		2パック	チルド
10012	ポークウィンナー		2袋	チルド
10013	豚肩ロース		744g	チルド
10014	鶏モモ肉		817g	チルド
10015	豚切り落とし		748g	チルド
10016	卵		14個	冷蔵トビラ
10017	バター		1パック	冷蔵トビラ
10018	マヨネーズ		1本	冷蔵トビラ
10019	ケチャップ		1本	冷蔵トビラ

取引3

取引データD0003として、10002を取得。すでにIDがあるのでDBを更新

- D0001
- D0003
- D0001
- D0002
- D0002

取引データ管理テーブル

取引データID	データ実装	ID	サービス実装	プロセス実装	事実区分
D0003	冷蔵庫	10002	DBMS	データ保存	改変



取引データとレコードの関係



テーブル名: 冷蔵庫

ID	品名	数量	単位	配置
10001	麦茶		2L	冷蔵棚
10002	BLACK COFFE		1L	冷蔵棚
10003	1日分の野菜		1L	冷蔵棚
10004	キムチ		1パック	冷蔵棚
10005	OICOS		4個	冷蔵棚
10006	絹豆腐		2パック	冷蔵棚
10007	オクラ昆布		3パック	冷蔵棚
10008	ウィダーゼリー		4個	冷蔵棚
10009	缶ビール		4本	冷蔵棚
10010	コースハム		2パック	冷蔵棚
10011	生ハム		2パック	チルド
10012	ポークウィンナー		2袋	チルド
10013	豚肩ロース		744g	チルド
10014	鶏モモ肉		817g	チルド
10015	豚切り落とし		748g	チルド
10016	卵		14個	冷蔵トビラ
10017	バター		1パック	冷蔵トビラ
10018	マヨネーズ		1本	冷蔵トビラ
10019	ケチャップ		1本	冷蔵トビラ

取引4

取引データD0001である10003、D0002である10004レコードをリクエストにより提供

- D0001
- D0003
- D0001
- D0002
- D0002

取引データ管理テーブル

取引データID	データ実装	ID	サービス実装	プロセス実装	事実区分
D0001	冷蔵庫	10003	DBMS	データ読出	読出
D0002	冷蔵庫	10004	DBMS	データ読出	読出



1. IVIの技術体系とCIOF
2. CIOFアーキテクチャの解説
3. ユースケースと操作手順
4. エッジコントローラの機能要件
5. サンプルシナリオによる説明
6. プロジェクト参加方法



IVI会員	NEDO –CIOFプロジェクト メンバ企業	現在第二期開発中(2022 年3月プロジェクト完了)
	IVI正会員／サポート会員 ／実装会員	コンポーネント申請が必要で す(10月9日より受付開始)
	IVI正会員(工場または事業所 をもつ企業)	IVI業務シナリオ2020にて、 実証実験(2021年3月完了)
非会員	IT企業(コンポーネント開発 企業またはインテ部レータ)	API情報、技術情報は公開、 サンプルやSDKは非公開
	工場または事業所をもつ企業 およびその取引先	IVIコンポーネントの利用は、 会員／非会員を問わず可能



<https://pfcp.iv-i.org/wp/>

IVI Platform Component 2020

概要 プラットフォーム コンポーネント 参加

概要

IVIプラットフォーム・コンポー

IVI Platform Component 2020

概要 プラットフォーム コンポーネント 参加

IVIプラットフォーム・コンポーネントの3つ特徴

- ものづくりが主役-**

ものづくりを実践する企業にとっての価値を最大化することを第一の目的とします。

ものづくり企業のものづくり企業によりものづくり企業のためのプラットフォームとして、コストを抑え、効果を最大化することで、結果としてIT企業にとっても大きなメリットがあるしくみとします。
- オープンなエコシステム-**

構成するコンポーネントについて、オープンな仕様にもとづくエコシステムとします。

個別の機能を提供するコンポーネントとして、顧客の要望によって、その都度異なるプラットフォーム上で稼働できる環境を作ります。これにより、プラットフォームの側は、より多くのすぐれたコンポーネントを集めるためにさらにつなげる機能が向上していくことを狙っています。
- 知財としてのデータ管理-**

企業データの所有者は企業自身とし、ボトムアップなくみづくりを可能とします。

プラットフォームで扱うデータは、一義的にはものづくり企業のもので、IVIプラットフォームでは、原則として製造業のエッジ側で得られたデータに関する権利は、その製造業が保持するものとし、ものづくりの技術やノウハウの健全な取引を支援します。





参加方法

参加方法は下記の手順となります。なお、参加希望の際は
問い合わせ先 (Mail) : office@iv-i.org
までご連絡ください。

1. 申し込み書を記入し事務局へ送付
2. 詳細な記述フォームに内容を記入
3. モデラーと辞書ツールにて内容を登録
4. グレードの登録審査申請を事務局へ送付
5. 審査WGにて内容を精査
6. プラットフォーム委員会にて審査
7. 認証の場合は事務局から請求書を送付
8. グレード登録料の支払い
9. グレード認定

(審査期間は申請から1年間となります。)



グレード認定 (コンポーネント企業)

IVIのプラットフォーム委員会において、各コンポーネントのグレード認定を行います。

グレード1 :

- プロフィールとともにIVIモデラーにモデル登録した。
- CIOF辞書ツールで連携に必要なモデルを定義した。

グレード2 :

- CIOFコンポーネントとしてエッジコントローラの仕様に従い実装した。
- 独自に接続テストを行い、他のCIOFコンポーネントと通信を確認した。

グレード3 :

- IIVIが定める認証試験でコンポーネント間の相互接続が確認された。
- CIOF上で外部辞書を提供し、事業者間のデータ流通を可能とした。

IVIコンポーネントの本年度の申請受付は10月9日からとなります。
なお、IVIプラットフォームは本年度は募集していません。



活動内容

• 総合企画委員会／規格ビジネスWG

- IVRA普及啓発と国際展開
- スマートシンキング標準化
- CIOFコンポーネント認証

アーキテクチャ
／国際連携

データ取引
ビジネスモ
デル規約

つながる化
方法論リ
サーチ

• ビジネス連携委員会／プラットフォームWG

- IVIモデラーとスマートシンキング開発
- 共通辞書／テンプレート開発
- CIOFビジネスモデル構築支援

業務シナリ
オ事例テン
プレート

辞書開発、
共通スキーマ設計

CIOFの実装
(エッジコント
ローラ、ツール)



- 月一回のWeb会議（第3木曜15時30より）
- 年内は合同WGとして実施（西岡も参加）
- メンバー受付は10月のシンポジウムにて開始
- 第1回は10月15日（木）より、ガイダンス含む
- 12月よりSWGごとの活動を開始、ゴール設定
- 最終ゴールは3月報告書→ハノーバーメッセ発表

メンバーページにて各自で登録可能です。10月10日以降は、事務局までご連絡ください。

メンバーページ→グループ→
2020年度規格ビジネス／プラットフォーム合同WG

参加方法



10月15日（木）
11月19日（木）
12月17日（木）
1月21日（木）
2月18日（木）
3月18日（木）

IVRA-Next/CIOF解説&議論
CIOF実装／IVIモデラー解説&議論
2020年度業務シナリオWG分析
個別のWGに分かれての活動
個別のWGに分かれての活動
報告書まとめ



IVI公開シンポジウム2020-Autumn- ～ コロナ禍で、ものづくり革命進行中！ ～

主催：一般社団法人インダストリアル・バリューチェーン・イニシアティブ

日時：2020年10月08日（木）12:30～18:40

場所：第一部（Youtube Liveによる配信）

第二部（MicroSoft Teamsによる相互交流）

定員：特に制限を予定しておりません

参加費：無料

【プログラム】 ※予告なく演題・内容・講演者が変わる場合がありますので、あらかじめご了承ください。

12:15	【第1部】受付開始 総司会：西村 栄昭（IVI代表幹事／ブラザー工業）
12:30	【開会挨拶】 渡部 裕二（IVI事務局長）
12:40	【招待講演】 「データで進める『At Your Side経営』」 小池 利和 様（ブラザー工業株式会社 代表取締役会長）
13:20	【IVIオピニオン】 「デジタル化とデータ化と価値経済の行方」 西岡 靖之（IVI理事長／法政大学 教授）
14:00	【2019年度業務シナリオWG 優秀事例紹介】 市本 秀則（IVIビジネス連携副委員長／マツダ） ※優秀事例は、下記「2019年度業務シナリオWG 優秀事例紹介」を参照

15:00	休憩
15:10	【講演】 「スマートシンキングで乗り越える 新常態のものづくり」 西村 栄昭（IVI代表幹事／ブラザー工業）
15:30	【2020年度業務シナリオWG 進捗報告】 水野 博之（IVIビジネス連携委員長／CKD） 各シナリオWG発表者 ※発表者は、下記「2020年度業務シナリオWG 発表詳細」を参照

16:30	休憩
16:40	【先進研究分科会 ハイライト】 活動紹介：AI・深層学習応用研究分科会 富田 浩治（IVI総合企画委員長／安川電機） 平田 俊明（IVI・AI・深層学習応用研究分科会 主査／コンピュータロン）
17:00	【IVIパネルディスカッション】 「コロナ禍を生き抜くものづくりの智慧・知恵・知慧！！」 堀水 修（IVIフェロー／日立製作所） 関 行秀（IVIフェロー／NEC） 渡邊 嘉彦（IVI副代表幹事／伊豆技研工業） 古賀 康隆（IVI技術統括）

皆さんの参加をお待ちしています



業務シナリオWGの内容を一挙公開！



【2020年度業務シナリオWG 発表詳細】

※予告なく演題・内容・講演者が変わる場合がありますので、あらかじめご了承ください。

セッション1：ひと手間加えてみんなが幸せ、知恵と工夫で前に進め！		
WG番号	業務シナリオWGタイトル	発表者
6E02	マスカスタマイゼーションに効くつなげ方	山本 博士/IHI
6C05	工程能力の可視化による業務効率化	藤田 亮介/神戸製鋼所
6B01	生産設備の消耗部品の予知保全	森下 篤史/栗田産業
6A04	製品管理のための低コストな情報取得の実現	遠塚 弘/レイマック
セッション2：実装天国・分析パラダイス、IoTのツートップ！		
6C02	AIによる製造ラインの生産性向上 第4弾	市本 秀則/マツダ
6A01	検査の自動化プラットフォーム 活用天国	本田 祥/CKD
6A03	ダイカストシリンダーブロック素材品質向上	野口 智史/三菱電機

17:40よりオンライン懇親会があります。

各WGの発表者への直接の質問の会場や、CIOFの詳細な議論(西岡担当)会場などに参加して、理解を深めてください。

セッション3：自粛できない運搬は、可視化・自立化・自動化の3拍子で！		
6C01	搬送機器の遠隔操作による部品庫物流自動化	奥屋 太志/マツダ
6E01	価値を生まない“モノの搬送”革新-分析編	大島 啓輔/セレンディップ・ホールディングス
6C03	人・モノの実績可視化－Ⅲ（次世代IE追究）	吉岡 新/マツダ
セッション4：遠くで現場に行きながら・・・、つながるFormation!		
6C04	製造工程（外観検査）のリモート化	瀬戸 大樹/ニコン
6E03	エッジAIとデータ流通でIVI型製造進化	松岡 康男/東芝
6A02	エッジと遠隔による現場支援	吉本 廉浩/三菱電機





Connected Industries Open Framework
Powered by Industrial Value Chain Initiative