

【先進事例】製造業オープン連携フレームワーク（CIOF）によるデジタル革命
～データ取引実証実験の成果報告～

カテゴリー1

製造ノウハウを含むデータの知財管理

DMG森精機株式会社

報告者 曾我 崇明

■ 本事業の目的

- 工場間、企業間でのデータ流通ニーズが増加
一方で製造現場における知的財産の保護管理が重要

- 下記の仕組みが今後求められる
 - ・簡単に接続可能
 - ・セキュアな環境
 - ・多拠点間でデータの連携可能

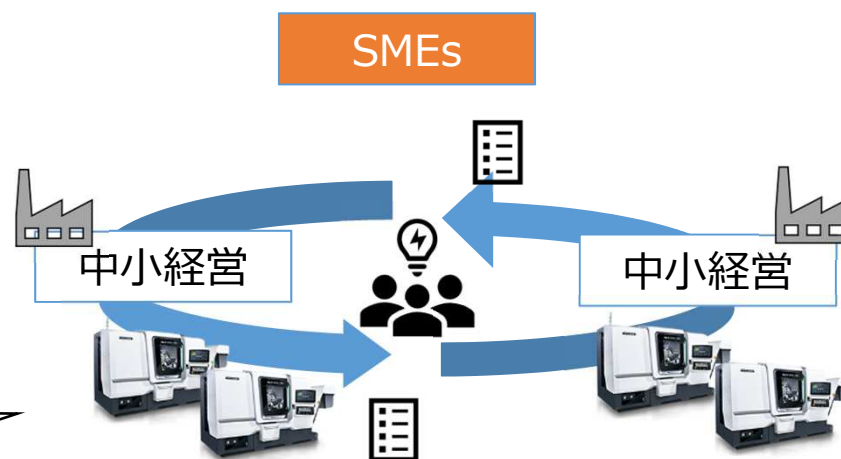
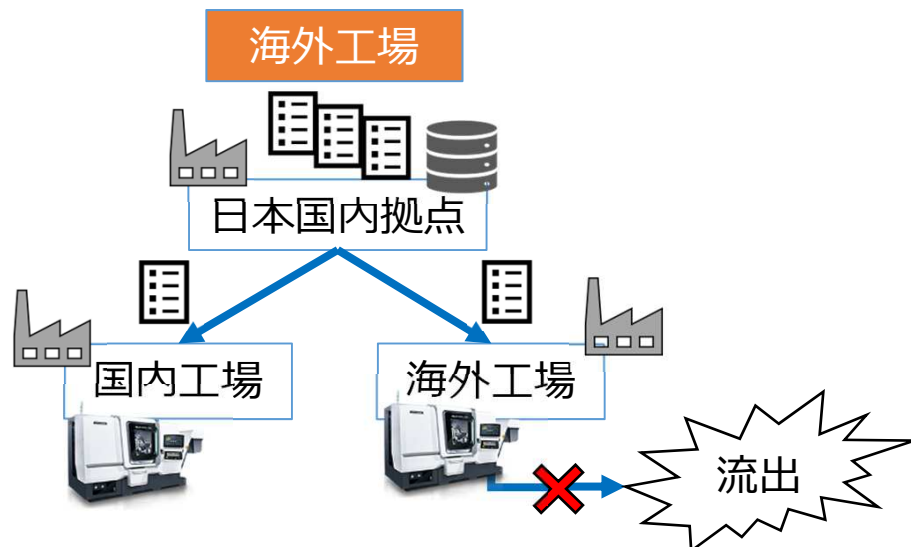


シナリオ

- 製造ノウハウを含むNCプログラム等データの知財管理は今後益々重要
→セキュアな管理が必要
- 先進国においては熟練作業者が不足
従来の生産性を維持
生産分業制
情報の一元管理

CONFIDENTIAL!

<想定されるユーザ>



シナリオ

CONFIDENTIAL!

- 生産管理と加工現場の2拠点間で流通される加工データを想定
下記2点をターゲットとし汎用的に利用できるシステムを目指す

① CIOFによって

- ・データの流通先を拠点間で締結した契約内容に限定する。
- ・データの利用を監視し、保存/修正/削除および加工実績など、不正利用を防ぐ。

② 生産管理に対して

発注時など、加工現場の機械状況等を把握できる仕組みを提供する。

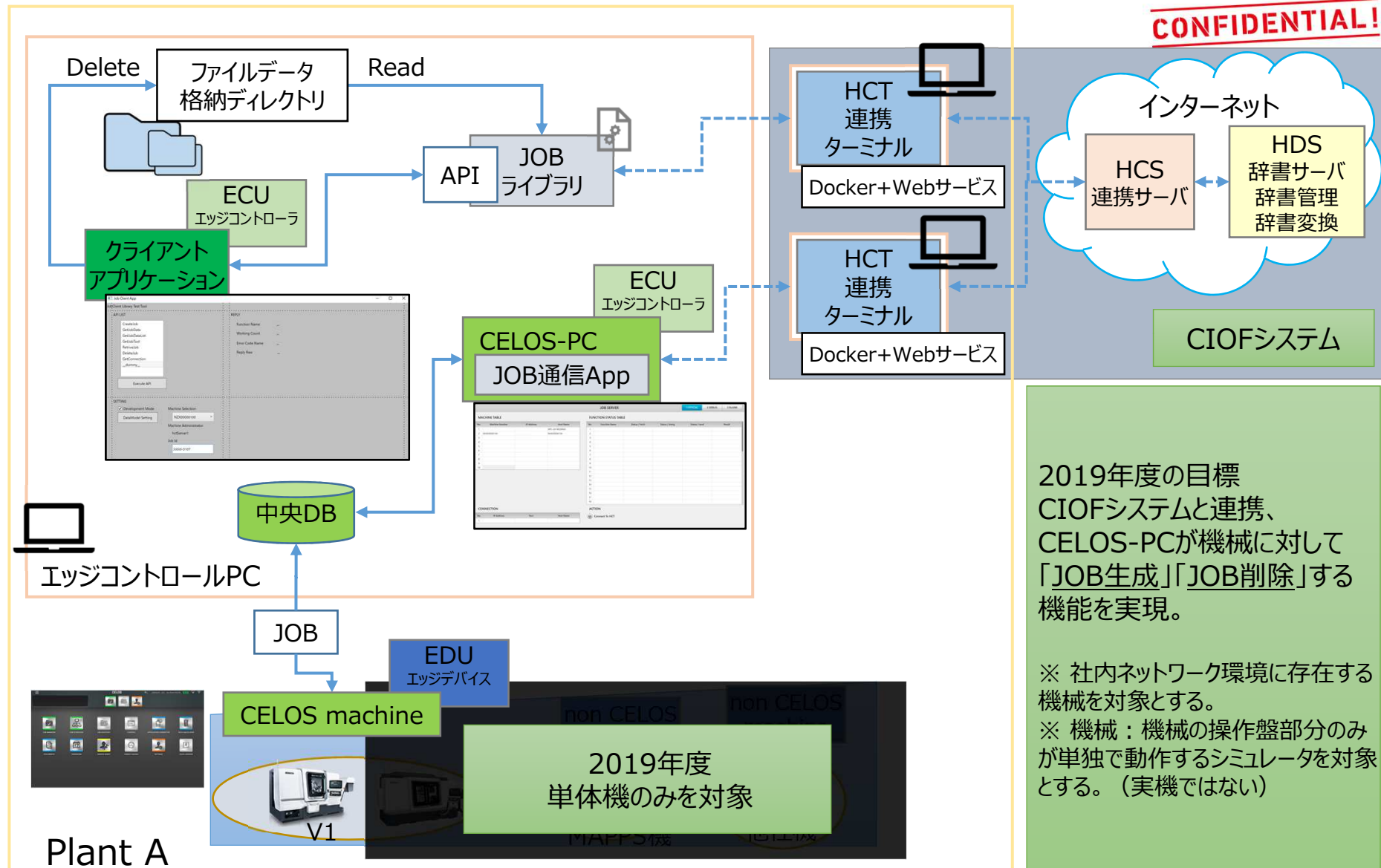
- CIOFで管理するデータ

加工に必要なデータをパッケージングし、JOBとして定義/取り扱う。

| JOB | |
|---------|----|
| NCプログラム | |
| 工具データ | |
| クランプ用装置 | |
| 補助具 | など |



システム構成



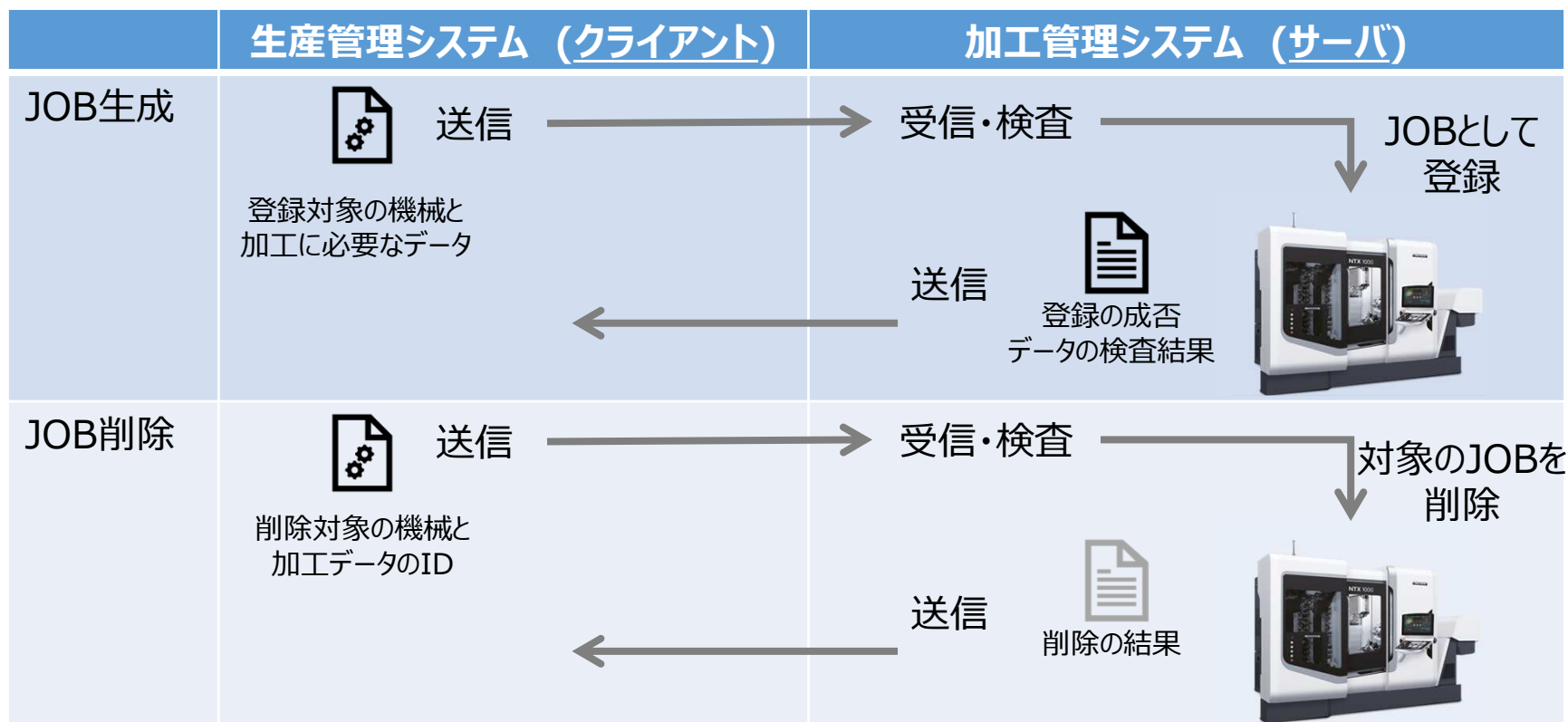
実証実験

CONFIDENTIAL!

■ 実験目標

- 前述のシステム構成にて実証実験を行う。
- 生産管理システムと加工管理システムの2拠点間でデータ流通を行う。
- 単体機に対して「JOB生成」「JOB削除」を実証する。

■ 実験内容 [各システムの振舞い]



■ 実験内容 [動画]

- 具体的なデータの流れ、ソフトウェアの動作を動画で紹介

CIOF2019
DMGMORI Demo Movie

■ 実験結果 [送受信したJSONデータ]

| | [リクエスト] クライアント → サーバ | [リプライ] サーバ → クライアント |
|-----------|---|--|
| JOB 生成 | <pre>{ "created_at": "2020-03-02T23:47:40.544Z", "dgr_id": "33856", "id": "69672", "message": [{ "values": ["JobElement", "NTX00000102", "JobId-0228", *省略*]}, { "values": ["JobInfoElement", *省略*]}, { "values": ["SetupElement", "SetupId-0228-01", *省略*]}, { "values": ["NcProgramElement", "C:/Celos/share/.../ncProgramFileList/101", *省略*]}], "message_type": "response", "request_params": [], "response_limit": "", "status": "sending", "tcb_id": "70" }</pre> <p>データの送信部</p> <p>この属性("message")に設定して送信する場合は、辞書に定義を行う必要がある。</p> | <pre>{ "created_at": "2020-03-02T23:47:44.276Z", "dgr_id": "33857", "id": "69673", "message": [{ "values": ["0", "SUCCESS"] }], "message_type": "response", "status": "sending", "tcb_id": "71" }</pre> <p>データの送信部</p> |



■ 実験結果 [送受信したJSONデータ]

| | [リクエスト] クライアント → サーバ | [リプライ] サーバ → クライアント |
|-----------|---|--|
| JOB 削除 | <pre>{ "created_at": "2020-03-02T23:51:44.534Z", "dgr_id": "33858", "id": "69674", "message": [], "message_type": "request", "request_params": [{ "name": "設備ID", "value": "NTX00000102" }, { "name": "プロダクトID", "value": "JobId-0228" }], "response_limit": "", "status": "sending", "tcb_id": "72" }</pre> <p>データの送信部</p> <p>この属性("request_params")に設定して送信する場合は、辞書への定義は不要。</p> | <pre>{ "created_at": "2020-03-02T23:51:48.167Z", "dgr_id": "33859", "id": "69675", "message": [{ "values": ["0", "SUCCESS"] }], "message_type": "response", "status": "sending", "tcb_id": "72" }</pre> <p>データの送信部</p> |

CONFIDENTIAL!

■ 実験結果 [使用した共通辞書、個別辞書のデータモデル]

- JOB生成シナリオのデータ連携に用いたモデル数。
 リクエスト用モデル：4つ
 リプライ用モデル：1つ
- JOB削除シナリオのデータ連携に用いたモデル数。
 リクエスト用モデル：無し
 リプライ用モデル：1つ
- リクエスト用モデル：
 クライアントからサーバへ伝える。
 仕事を依頼するために必要な情報。
- リプライ用モデル
 サーバからクライアントへ伝える。
 依頼された仕事の結果を表す情報。

※JOB削除では、リクエスト用のモデルを定義していない。

※要求を送る際に"request_params"を用いて「設備ID」「プロダクトID」を伝えるため。

※太字：共通辞書、各個別辞書において項目名が異なるモノ。

JOB生成／リクエスト／モデルI, II

| No. | 共通辞書 | 個別辞書 (クライアント) | 個別辞書 (サーバ) |
|-----|-------------------|------------------|-----------------|
| | プロダクトエレメント モデル | タスクエレメント モデル | ジョブエレメント モデル |
| 1 | モデル名 | モデル名 | モデル名 |
| 2 | 設備ID | 対象機ID | 対象機ID |
| 3 | プロダクトID | タスクID | ジョブID |
| 4 | プロダクト名 | タスク名 | ジョブ名 |
| 5 | 顧客ID | 顧客ID | 顧客ID |
| 6 | 顧客名 | 顧客名 | 顧客名 |
| 7 | 開始日 | 開始日 | 開始日 |
| 8 | 終了日 | 目標完了日 | 目標完了日 |
| 9 | 実行開始日時 | 実行開始日時 | 実行開始日時 |
| 10 | カスタムA | カスタムA | カスタムA |
| 11 | カスタムB | カスタムB | カスタムB |
| 12 | 優先度 | 優先度 | 優先度 |
| 13 | 素材数 | 利用可能数 | 利用可能数 |
| 14 | 必要数 | 必要数 | 必要数 |
| 15 | 素材名 | 素材名 | 素材名 |
| 16 | プロダクトデータ | タスクデータ | ジョブデータ |
| | プロダクト情報 モデル | タスク情報 モデル | ジョブ情報 モデル |
| 1 | モデル名 | モデル名 | モデル名 |
| 2 | 説明 | 説明 | 説明 |
| 3 | ファイルデータ | ファイルデータ | ファイルデータ |

モデル名 / 項目名



CONFIDENTIAL!

■ 実験結果 [使用した共通辞書、個別辞書のデータモデル]

JOB生成/リクエスト/モデルIII, IV

| No. | 共通辞書 | 個別辞書 (クライアント) | 個別辞書 (サーバ) |
|-----|---------------------|-------------------|-------------------|
| | プロダクトセットアップ モデル | タスクセットアップ モデル | ジョブセットアップ モデル |
| 1 | モデル名 | モデル名 | モデル名 |
| 2 | セットアップID | セットアップID | セットアップID |
| 3 | セットアップ名 | セットアップ名 | セットアップ名 |
| 4 | 説明 | 説明 | 説明 |
| 5 | セットアップデータ | セットアップデータ | セットアップデータ |
| 6 | セットアップ時間 | セットアップ時間 | セットアップ時間 |
| 7 | 加工時間 | 加工時間 | 加工時間 |
| 8 | 補助具データ | 補助具データ | 補助具データ |
| 9 | 工具データ | 工具データ | 工具データ |
| 10 | クランプ装置データ | クランプ装置データ | クランプ装置データ |
| 11 | 追加情報説明 | 追加情報説明 | 追加情報説明 |
| 12 | 追加情報データ | 追加情報データ | 追加情報データ |
| | プロダクトNCプログラ ムモデル | タスクNCプログラム モデル | ジョブNCプログラム モデル |
| 1 | モデル名 | モデル名 | モデル名 |
| 2 | NCプログラムデータ | NCプログラムデータ | NCプログラムデータ |
| 3 | メインプログラム選択 | メインプログラム選択 | メインプログラム選択 |
| 4 | 系統 | 系統 | 系統 |

JOB生成/リプライ/モデルI

| No. | 共通辞書 | 個別辞書 (クライアント) | 個別辞書 (サーバ) |
|-----|------------------|------------------|----------------|
| | プロダクト結果通知 モデル | タスク結果通知 モデル | ジョブ結果通知 モデル |
| 1 | エラー番号 | エラー番号 | エラー番号 |
| 2 | メッセージ | メッセージ | メッセージ |

JOB削除/リプライ/モデルI

| No. | 共通辞書 | 個別辞書 (クライアント) | 個別辞書 (サーバ) |
|-----|----------------|------------------|---------------|
| | プロダクト削除 モデル | タスク削除 モデル | ジョブ削除 モデル |
| 1 | エラー番号 | エラー番号 | エラー番号 |
| 2 | メッセージ | メッセージ | メッセージ |



■ 成果

- 取引契約に沿って、生産管理システムと加工管理システムの2拠点間でデータの流通を実現。
- 単体機かつ特定の機種をターゲットとし実証実験を行った。
生産管理システムからのオーダーに従い、加工管理システム内に加工データの登録、加工データの消去を実現。
- CIOFシステムと接続しインターネットを介したプラットフォーム間データ流通の仕組みを開発。
加工管理システム内に留まっていたデータ管理に対しインターフェースを構築。

■ 課題

- 対象機の台数を増やす
 - 本年は単体機に焦点を絞ったが、接続台数を増やしたシステム開発を目指す。
- 対象機の機種を増やす
 - 本年はターゲット機種としてDMGMORIの最新オペレーティングシステムCELOSを搭載した機械としたが、その他の当社設備機に対応したシステム開発を目指す。
- 機械の状況を把握する仕組み
 - 本年はデータの“登録”と“削除”機能の実現に留まったが、機械の状況を把握するためのデータの“照会”機能の開発を目指す。

