広島大学大学院先進理工系科学研究科情報科学プログラム修士論文

教育データの組織横断的な利用を支援する処理配信システムの提案

広島大学大学院先進理工系科学研究科
情報科学プログラム

M221924

森田 崇大

指導教員 西村 浩二

広島大学

Master's Thesis
Informatics and Data Science Program, Graduate School of Advanced Science and Engineering,
Hiroshima University

Proposal for a Processing Distribution System to Support Cross-Organizational Use of Educational Data

Informatics and Data Science Program
Graduate School of Advanced Science and Engineering
Hiroshima University

M221924

Takahiro Morita

Supervisor: Kouji Nishimura

Hiroshima University

要旨

学校や企業で DX を推進するために，安全かつ効率的なデータ利活用が求められている．特に近年では，組織内での利活用だけでなく，複数組織が連携してデータを利活用する動きが増えている．組織を横断したデータ利活用を推進するためのプラットフォームがいくつか設計・開発されているが，それらのプラットフォームではデータを組織の外に持ち出す必要があったり，処理の秘匿性を重視して処理内容が公開されず，データに対してどのような処理を行っているのか，データ管理者が把握できないという問題がある．そのような既存のプラットフォームは，特に教育データの利用には適していないため，異なる性質や条件を持ったシステムを開発することで，教育データの利用を促す必要がある．

本研究では，教育データの組織横断的な利用を支援するシステム，処理配信システムを提案する．本システムでは，処理プログラムとその実行環境を含んだ処理環境を，データを保有する組織の，データが保存されているサーバに配信することで，データそのものを外部に持ち出すことなく，処理を行うことを可能にする．処理内容については事前にテストして説明と共に公開・共有することで処理の透明性を担保し，データ管理者がデータ利用に同意するかしないかを決められるようにする．

本システムを用いてデータの組織内利用，組織横断的利用の動作検証を行い，教育データの組織横断的な利用を支援するシステムとして適していることを示した．また，システムの応用例として，アカウント年度更新のためのフォローアップ講習や情報セキュリティインシデント対応訓練を複数の大学で行うことができ，確認テストなどの結果データを安全に収集し，組織横断的な集計・分析が可能となる．

Abstract

In order to promote DX in schools and companies, secure and efficient data utilization is required. In particular, there has been an increase in the use of data not only within organizations, but also through collaboration among multiple organizations. Several platforms have been designed and developed to promote cross-organizational data utilization, but these platforms require data to be taken outside of the organization, and do not disclose the contents of the data processing due to the emphasis on confidentiality, making it difficult for data managers to understand what kind of processing is being performed on the data. Such existing platforms are not particularly suited for the use of educational data, so it is necessary to promote the use of educational data by developing systems with different characteristics and conditions.

In this study, we propose a processing distribution system that supports the cross-organizational use of educational data. This system distributes a processing environment that includes a processing program and its execution environment to the server where the data is stored in the organization that owns the data, enabling processing without taking the data itself outside the organization. The contents of the processing are tested in advance and disclosed and shared with the public along with explanations to ensure transparency of the processing and to allow the data manager to decide whether or not to agree to the use of the data.

Using this system, we verified the operation of intra- and cross-organizational use of data, and showed that it is suitable as a system to support cross-organizational use of educational data. As an example of application of the system, follow-up training for annual account updates and information security incident response training can be conducted at multiple universities, and data on the results of verification tests can be securely collected, aggregated, and analyzed across organizations.

目次

第 1 章　はじめに

1.1　研究背景

学校や企業で教育やビジネスに変革を起こすために DX (Digital Transformation) が推進されており，DX を成功に導くためにデータ利活用が重要視されている．組織が保有するデータを利活用することで，その組織が抱えている課題を発見し改善につなげることができる．また，複数組織が連携してデータを利活用する動きが増えており，そのような組織横断的な利活用は，単一組織のデータだけでは見えなかったものが可視化されて新たなサービスを作るきっかけになるなど，組織内で閉じたデータ利活用を行う場合に比べて得られるものが多い．そこで，組織が保有しているデータをセキュアに処理・分析して利活用できるプラットフォームを開発する研究が行われている．望月ら [1] の研究では，サービスごとに独立したサンドボックス環境を構築し，パーソナルデータストアからサービスに必要なデータのみをサンドボックスに取り入れ，その中で処理させることで，パーソナルデータストアから大量のデータが流出することを防いでいる．坂本ら [2]，大村ら [3] は，セキュアな実行環境に複数組織がデータとプログラムを持ち込むことで，処理した結果を得ることができるプラットフォームを開発している．また，大規模な教育データを扱った研究の再現性を担保するためのプラットフォームを開発する研究も行われている [4].

教育データの利活用に着目したとき，各組織が保有する生の教育データや機密性の高いデータを収集して分析することは，プライバシー確保の観点から難しい．したがって，各組織の内部で共通の前処理を行い，匿名化等の後処理を行った後に，それらのデータを収集して分析する仕組みが必要となる．また，データの処理内容についてデータ管理者が同意できるように処理の透明性を担保する必要がある．既存のプラットフォームでは処理するためにデータを外部へ持ち出す必要がある．また，処理の秘匿性を重視しているために処理の透明性は担保されない．したがって，教育データ利活用のためのプラットフォームには異なる性質や条件が必要だと考えられる．

1.2　研究目的

本研究では，教育データの組織横断的な利用を支援するシステム，処理配信システムを提案する．本システムでは，データを処理するプログラムを含んだ処理環境を，データを保有している組織の，データが保存されているサーバに配信する．これによって，組織が保有しているデータを外部機関に渡すことなく，組織内での処理を可能とする．その時，保有しているデータの形式や各組織に配信する処理内容を標準化することで，組織横断的にデータを利用する際に，形式の揃ったデータを収集し，スムーズな分析を可能とする．また，処理内容について，データ管理者がデータ利用の可否を決められるように，プログラムをテストして，その説明と共に公開・共有することで処理の透明性を担保する．これらの機能を有した本システムを用いることで，新規に複数組織のデータを収集し， 組織横断的なデータの利用が可能となることを示す.

1.3　本論文の構成

本論文では，第 2 章で文部科学省や広島大学が提示している教育データの取り扱いや活用方法について述べた後，データ利活用のための処理プラットフォームについての関連研究を示し，それらの情報を基に教育データ利用の要件を示す．第 3 章では要件を満たすための提案システムの構成と機能，システムを用いた際のデータ利活用の流れを説明し，第 4 章でシステムの動作検証および応用例を示す．最後に第 5 章でまとめと今後の課題を述べる.

第 2 章　教育データ利用の要件

本章では，文部科学省や広島大学が提示している教育データの取り扱い・利活用ポリシー及びその活用方法を述べる．その後，データ利活用のための処理プラットフォームについての関連研究を示し，教育データの取り扱い，活用方法と関連研究の課題から教育データ利用の要件を示す．

2.1　教育データの取り扱い・利活用ポリシー

2.1.1　小学校，中学校，高等学校における教育データの取り扱い

文部科学省が提示している教育情報セキュリティポリシーに関するガイドライン [5] では，主に小学校，中学校，高等学校における情報資産について，セキュリティを考慮して分類し，分類に応じた管理方法・取り扱い例を規定している（図 1）．図 1 のそれぞれの重要性分類に該当する情報資産の例を表 1 に示す．図 1 と表 1 を見ると，多くの教育データは組織外部への持ち出し・送信を行う場合，適切なアクセス制御や匿名化を行う，情報セキュリティ管理者の承認を得るなどの手順を踏む必要がある．また，外部に持ち出したデータの管理・処理を行わせる場合は安全管理措置の規定を必要とするなど，教育データの取り扱いは制限が厳しいことがわかる．

情報資産の分類					情報資産の取扱例								
重要性分類	定義	機密性	完全性	可用性	複製・配布	組織外部への持ち出し制限*	端末制限	情報の組織外部への送信**	情報資産の運搬***	組織外部での情報処理****	使用する電磁記録媒体	情報資産の保管	情報資産の廃棄
Ⅰ	セキュリティ侵害が教職員又は児童生徒の生命、財産、プライバシー等へ重大な影響を及ぼす。	3	2B	2B	必要以上の複製及び配布禁止	本ガイドラインに準拠していることを確認した上で業務遂行上必要な場合には、情報セキュリティ管理者の判断で持ち出しを可	支給以外の端末での作業の原則禁止	限定されたアクセスの措置がとられていること*****	鍵付きケースへの格納	禁止	施錠可能な場所への保管	・耐火、耐熱、耐水、耐湿を講じた施錠可能な場所に保管（電子データの場合もこれらの対策に準じたサーバに保管） ・情報資産を格納するサーバのバックアップ ・6か月以上のログ保管 ・サーバの冗長化（推奨事項） ・オンラインで情報資産を利用する場合は通信経路の暗号化を実施 ・保管場所への必要以上の電磁記録媒体の持ち込み禁止	電子記録媒体の初期化、復元できないようにして廃棄
Ⅱ	セキュリティ侵害が、学校事務及び教育活動の実施に重大な影響を及ぼす。	2B	2B	2B	同上	同上		同上	同上	安全管理措置の規定が必要	同上	同上	同上
Ⅲ	セキュリティ侵害が、学校事務及び教育活動の実施に軽微な影響を及ぼす。	2A	2A	2A	同上	情報セキュリティ管理者の包括的承認で可		同上	同上	同上	同上	・耐火、耐熱、耐水、耐湿を講じた施錠可能な場所に保管（電子データの場合もこれらの対策に準じたサーバに保管） ・情報資産を格納するサーバのバックアップ（推奨事項） ・一定期間以上のログ保管 ・サーバーハードディスクの冗長化（推奨事項） ・オンラインで情報資産を利用する場合は通信経路の暗号化を実施 ・保管場所への必要以上の電磁記録媒体の持ち込み禁止	同上
Ⅳ	影響をほとんど及ぼさない。	1	1	1									

図 1　情報資産の取扱例（[5] の p.39 より引用）

2.1.2　大学における教育データの利活用ポリシー

AXIES (大学 ICT 推進協議会)[6] は，大学の教育・学習データの利活用の推進を図る目的で，教育・学習データ利活用ポリシーのひな形を提供している．[6] のひな形を参考に，各大学機関が利活用ポリシーを定めることで，教育データの安全かつ円滑な利活用を促し，大学改革や教育改善を行っていくことが求められている．

表 1　各重要性分類に該当する情報資産の例（[5] の p.37 を基に筆者一部抜粋）

重要性分類	情報資産の例
Ⅰ	指導要録原本，教職員の人事情報，教育情報システム仕様書など
Ⅱ	児童生徒・教職員の個人情報（住所，生年月日など），通知表，健康診断表など
Ⅲ	授業用教材，児童生徒の氏名，児童生徒の学習記録など

広島大学では，[6] のひな形を基に，広島大学教育・学習データ利活用ポリシー [7] を定めている．[7] が示している教育・学習データ取扱 8 原則は以下の通りになっている．

1. 利用目的を明示し，目的外には使用しません．
2. 利用手法とその結果を明示します．
3. 権利者がいつでもデータ利活用に関する同意を取り下げることができます．
4. 個人情報保護法などの関連する法令を遵守します．
5. 権利者がいつでも自分のデータにアクセスできるようにします．
6. データの分析結果の公表については個人が決して特定されないようにします．
7. データに適切な安全管理措置を施します．
8. 研究成果やデータの共有によって，人類の福利に貢献します．

利用目的はデータの分析や可視化によって教育改善や学習を支援することであり，それ以外の目的では使用しないことを原則としている．また，データ主体（学生や教職員）には利用するデータと目的を伝え，その上でデータ主体がデータ取得について同意するかしないかを決めることができ，いつでも同意を取り下げることができるとしている．つまり，教育データを利活用する際は，データ主体や社会にとって有益な目的である必要があり，データの利用に関して同意の可否を決めることができるように，データの処理内容の透明性を担保する必要があると考えられる．データ利用後は結果をデータ主体や社会に公表することで，利活用の成果を還元することも重要である．教育データは個人情報として扱い，法律や規則に従って管理することでプライバシーを保護する．データに適切な安全管理措置，つまり適切なアクセス制御や匿名化等を施すことで，データ主体やデータ管理者が不利益を被らないようにデータを扱い，その上でデータ主体や社会に貢献する必要がある．

2.2　教育データの活用方法

文部科学省や AXIES が述べているように，教育データの取り扱いには厳しい制限がある一方で，生徒や学生の学習の改善や教員の指導改善など，教育をより良いものにするために，教育データの利活用が求められている．教育データの利活用に係る論点整理（中間まとめ）[8] では，教育データの利活用の視点をまとめている．教育データの利用方法としては主に一次利用と二次利用に分けられる．一次利用と二次利用のイメージを図 2 に示す．一次利用では児童生徒や教員に関わる具体的なデータを対象とし，それらのデータを直接，組織内で利用することで児童生徒の学びを改善するなど，学校現場での実践を目的としている．二次利用では文部科学省が行う調査のデータなどを対象とし，学校全体の状況や傾向の把握・比較を行うなど，学校や社会全体のための利用を目的としている．また，図 2 にもあるように，二次利用で得た結果から，一次利用で取得するべきデータやその利用方法を明らかにすることで，二次利用から一次利用へ成果の還元を行うことができる．つまり，一次利用のように組織内でデータを利活用するのみならず，二次利用のように組織を横断してデータを利活用することが求められている．これは主に小学校や中学校を対象として議論することで提示された活用方法だが，大学の教育データにおいてもこの活用方法が適用できると考えられる．

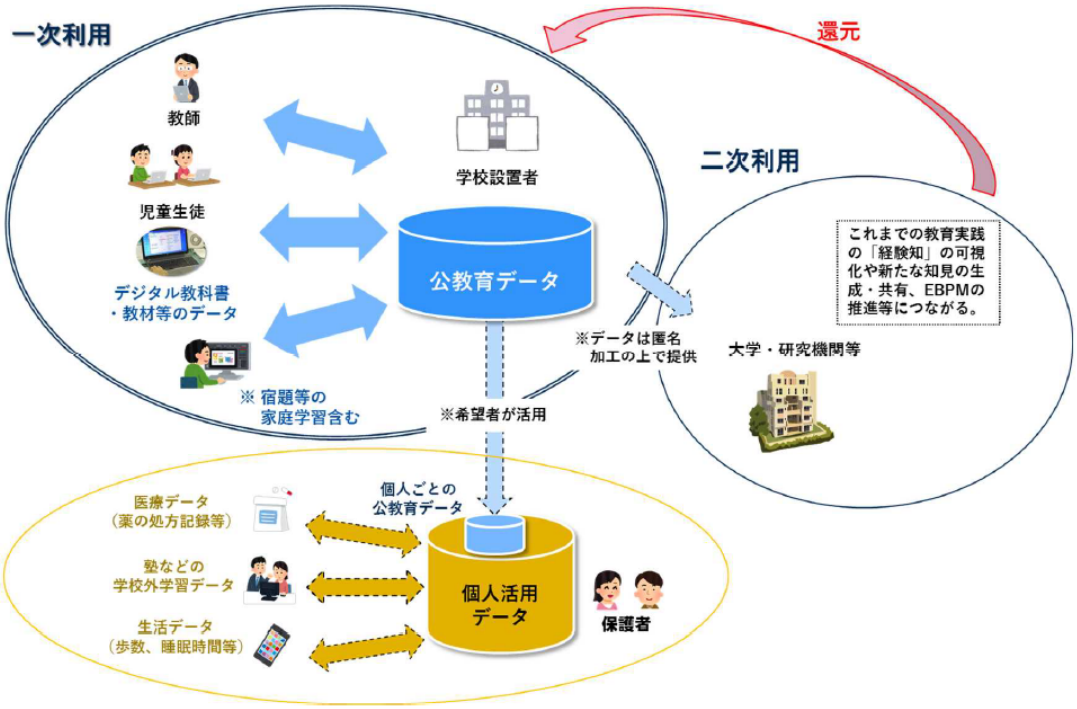

図 2　一次利用と二次利用，公教育データと個人活用データのイメージ（[8] の p.7 より引用）

2.3　既存のデータ処理プラットフォーム

2.1 節，2.2 節で述べたように，教育を改善するために，教育データの組織内での利用や組織横断的な利用を行うことが求められている．現在，多様な目的でデータを処理して利活用するためのプラットフォームが開発されているため，本節では，それらの既存のデータ処理プラットフォームについて説明する．

2.3.1　パーソナルデータストアサンドボックス

望月ら [1] は，データ保有者が自身のパーソナルデータを管理し，データ提供の可否を決めることができる機能を持つパーソナルデータストア（以下，PDS）[9] に着目し，PDS のデータを利用したアプリケーションの脆弱性によって PDS にある大量のデータが流出することを防ぐためのサンドボックス環境を設計・提案している．方法としては，図 3 のようにパーソナルデータを利用する事業者のサービスごとに必要なデータのみを取得し，他のサーバから独立したサンドボックスにデータを取り込み，その中で処理している．サンドボックスの実現にはコンテナ型仮想化技術を利用しており，一部のサンドボックスが攻撃を受けたとしても，その他のサンドボックスやデータの取得元となった PDS には攻撃の影響が及ばないような設計となっている．

2.3.2　データ価値共創プラットフォーム

坂本ら [2] は，複数組織が連携してデータを利活用することで，価値を共創することを目的としたデータ価値共創プラットフォームを提案している．図 4 にプラットフォームの構成を示す．図 4 のデータ保護部では，DRM (Digital Rights Management) 技術を用いてデータに所有者権限や利用権限など適切な権限を付与した後に，公開鍵暗号方式で暗号化を行う．そのようにして権限付与と暗号化が施されたデータはセキュアストレージに保存され，データのカタログ情報が生成される．これによってデータ利用者はどのような種類のデータがどこにあるか検索できるようになる．データ利用者は目的のデータをセキュアコンテナ内に取り込み，復

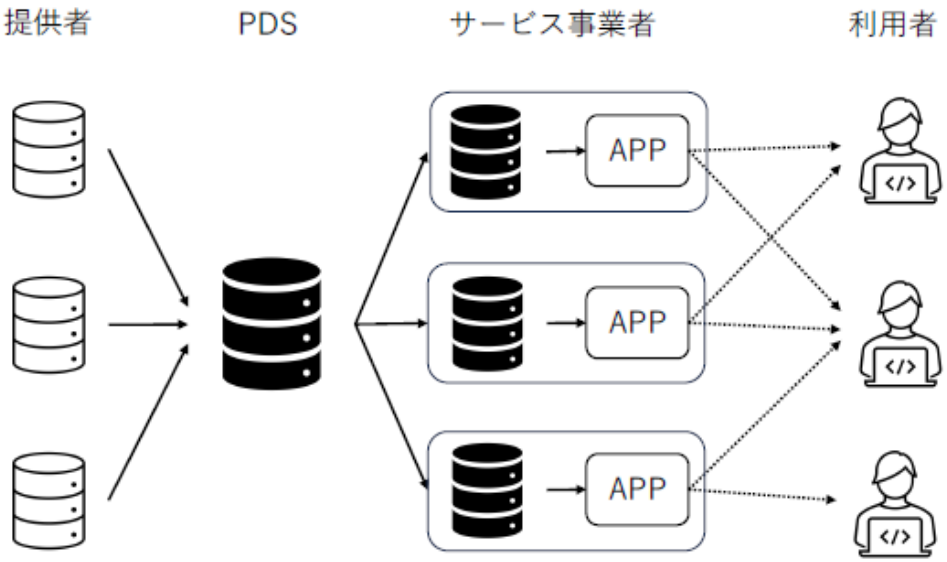

図 3　[1] の提案システムにおけるデータ提供（[1] より引用）

号した後，プログラムによる処理を行う．[2] におけるセキュアコンテナは，他者からはコンテナ内の処理やデータは見えないような設計となっており，処理されたデータは適切な権限付与と暗号化が施され，セキュアストレージに保存される．このプラットフォームを利用することで，データとプログラムは保護され，漏洩防止を実現できる．また，利用した結果，生成されるデータについても元のデータ提供者が意図していた権限が反映されるため，元のデータ提供者の意図に沿わない組織のデータ利用は制限され，データ提供者が自身のデータの利用履歴を把握することを容易にしている．

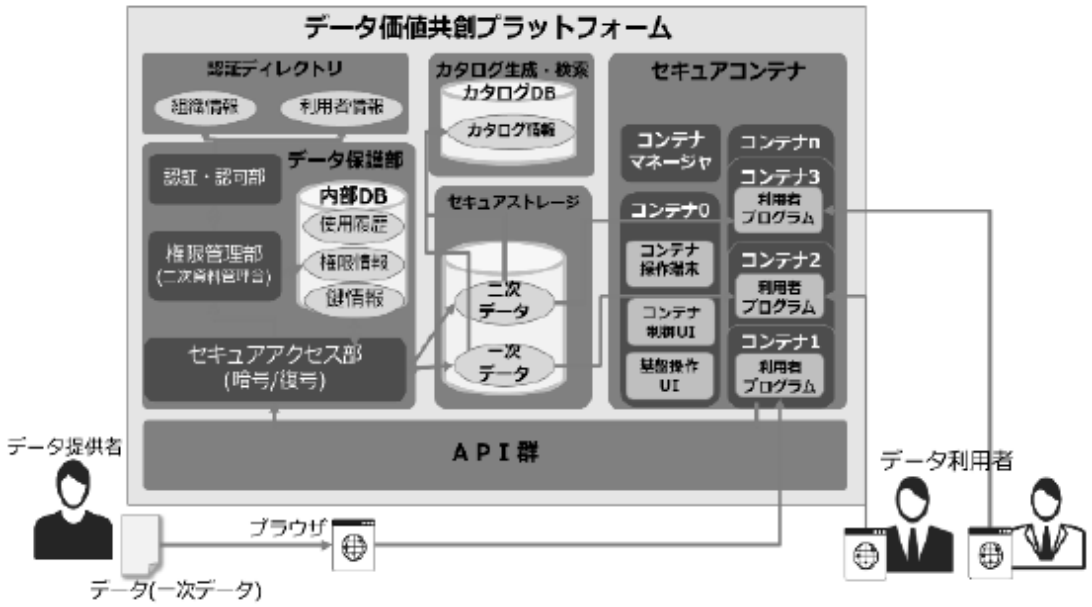

図 4　データ価値共創プラットフォームの構成（[2] より引用）

2.3.3　次世代データハブ

大村ら [3] は，[2] と同様の目的で，企業や組織を超えたさまざまなデータの利活用をセキュアに行うためのプラットフォーム，次世代データハブを提案している．図 5 に次世代データハブの全体像を示す．図 5 の仮想データレイクでは，データ本体を一箇所に集めるのではなく，データの所在や形式などを示すメタデータを収集している．データ利用者はメタデータの情報を見て必要なデータを探す．そしてデータ利用者の要求があって初めて，データ提供者はデータを提供する．これにより，データ提供者が保有しているデータを無制限にコピーされることがなくなり，データの流通・利用実績の管理が容易になっている．提供されたデータは図 5 のデータサンドボックスに送信される．データサンドボックスの中では TEE (Trusted Execution Environment) を利用したセキュアな環境が作られている．TEE とは，「OS の管理権限を持つユーザによるメモリ参照を防止するため，CPU がメモリ領域を暗号化して利用するように構成された隔離実行環境のこと」である [3]．その環境内にデータとプログラムが送信され，処理を行った結果が必要な人の元へ送信される．処理完了後は実行環境が消去され，データやプログラムの漏洩を防止する．データとプログラムがそれぞれ異なる組織から提供された場合，それらのデータとプログラムは互いに秘匿される．さらに TEE を使っているため，プラットフォーム事業者にもそれらの内容は秘匿されることになる．

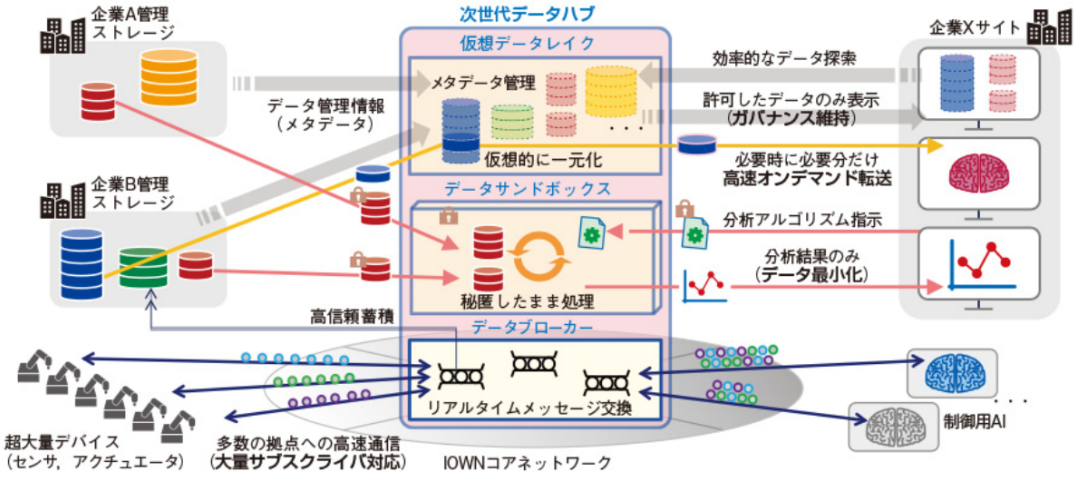

図 5　次世代データハブの全体像（[3] より引用）

2.3.4　A Framework for Predictive Modeling and Replication At Scale With Privacy-Restricted MOOC Data

ビッグデータを扱った研究の再現性の欠如を指摘し，再現性を担保するための研究 [4] も行われている．[4] では，実験結果の再現性を確保するためにコンテナを使用しており，ユーザは実験を行うための処理プログラムを含んだコンテナイメージと，実験のワークフローを示すコントローラースクリプト，ジョブ・メタデータを含むコンフィグファイルをプラットフォームに送信することで，データに対して指定した処理が実行される．実験がエラーなく完了した場合，コンテナイメージがパブリックリポジトリで公開・共有され，他の研究者はそのイメージを用いて実験の追試などができるような仕組みとなっている．

2.3.5　関連研究の比較表

表 2 に関連研究の機能についての比較表を示す．

組織横断的な利用についてはデータ利用者の視点で評価した．[1] はデータを横断的に利用することは目的とされていない．[2] や [3] はプラットフォーム上に複数組織がデータやプログラムを持ち込むことで，組織横

表 2　関連研究の比較表

機能	関連研究 [1]	関連研究 [2]	関連研究 [3]	関連研究 [4]
組織横断的な利用	×	○	○	○
組織内でのデータ処理	×	×	×	×
処理の透明性	×	×	×	△
データの秘匿性	○	○	○	○
アクセス制御	○	○	○	○
データ検索	○*	○	○	○*

断的なデータ利用が可能になっている．[4] は研究の実験の追試を行うために，複数組織がデータおよびその処理プログラムなどを利用できる．

組織内でのデータ処理についてはデータ管理者の視点で評価した．つまりデータ管理者の組織内でのデータ処理が可能かどうかである．[1] は PDS からデータを取得してサービス事業者の元で処理を行っている．[2][3][4] はデータを外部のプラットフォーム上で処理している．

処理の透明性についてはデータ管理者の視点で評価した．[1][2][3] では処理を秘匿しており，データ管理者に対して公開されない．[4] はコンテナイメージを公開しているものの，その内容はテストされておらず，適切な動作を行うかどうかが保証されない．

データの秘匿性についてはデータ管理者の視点で評価した．データは暗号化などが施され，[1][2][4] ではデータ管理者とデータ利用者，プラットフォーム事業者のみが閲覧可能となっており，[3] ではデータ利用者やプラットフォーム事業者に対しても TEE によって秘匿されている．

アクセス制御についてはデータ管理者の視点で評価した．関連研究のプラットフォームはすべて認証機能が導入されており，[2][3] は認証に加えて，ネットワークによる外部へのアクセスを禁止している．

データ検索についてはデータ利用者の視点で評価した．[1][4] はデータ管理者が提供したデータをプログラムから利用することができるが，[2][3] はそれに加えて，どのような種別のデータがどこにあるのか，クエリを発行して送ることで柔軟なデータ検索が可能となっている．

2.4　教育データ利用の観点から見た関連研究の課題

2.3 節で述べたように，多様な目的でデータ処理プラットフォームが設計・開発されているが，教育データの利用という視点から関連研究の課題を以下に示す．

1. データを組織の外部に持ち出して処理している．
2. データの処理内容が秘匿されており，処理の透明性が担保されない．

1 つ目の課題について，関連研究では図 6 のように，データはインターネットを介して外部のプラットフォームに提供される．2.1 節で述べた通り，教育データを外部に持ち出すには情報セキュリティ管理者の承認を必要とするなど，データの持ち出しに対する制限が厳しい．また，教育データに限った話ではないが，業務委託先での個人情報漏洩 [10][11] などのインシデントも度々起きており，その点からもデータの持ち出しは望ましくない．

2 つ目の課題について，特に [1][2][3] ではデータの処理内容が公開されておらず，データ管理者やプラットフォーム事業者に対して処理内容を秘匿している．企業や研究機関などが独自の分析アルゴリズムを使っており，それを公開したくないというケースではこれらのプラットフォームが適しているが，教育データの利活用という観点で見ると，2.1 節で述べた通り，データ主体やデータ管理者はデータの利用に関して同意の可否を

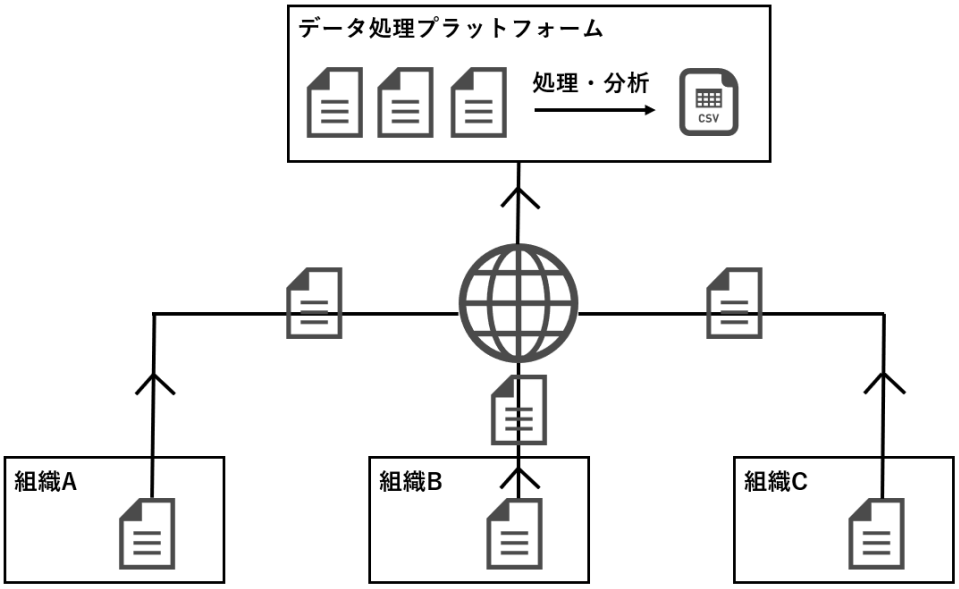

図 6　データの外部への持ち出し

決める権利があるため，処理内容について同意するかしないかを決めることができるように，プログラムなどを公開することが必要である．[4] ではプログラムなどは公開されているが，その処理内容がデータ管理者にとって不都合なものになっていないかどうかをテストして，そのテストを通過できればプログラムの説明と共に公開・共有するという手順を踏んだほうがセキュリティの面で良いと考えられる．

2.5　教育データ利用の要件

教育データの取り扱いや利活用ポリシーが定められている中，教育データの実際の利用，特に組織横断的な利用に関してはプライバシー保護などの問題があり，積極的に行われていない．本節では，これまで述べてきた教育データの取り扱い及び活用方法と関連研究の課題を踏まえて，組織横断的な利用も可能となるような，教育データ利用の要件を以下に示す．また，教育データ利活用の流れを図 7 に示す．なお，一次利用は組織内で直接利用する形態であり，二次利用は複数の組織を横断して利用する形態であるため，本論文では一次利用，二次利用をそれぞれ組織内利用，組織横断的利用と呼ぶことにする．以下でそれぞれの要件について説明する．

1. 生データを外部に持ち出さずローカルで処理を行うこと
2. データ形式・処理内容の標準化
3. 処理の透明性を担保すること
4. ID フェデレーション

2.5.1　生データを外部に持ち出さないローカル処理

関連研究ではデータを外部に持ち出さなければ利用できないことと，教育データを安易に外部に持ち出してはならないという要望から，この要件が必要だと考えられる．この要件を満たせば組織内利用を実現できる．図 7 では黄色の枠線で囲んだ部分が組織内利用にあたる．図 7 の組織 A では組織内利用だけで処理が完結しており，組織 A は処理を行った結果を組織内で活用することができる．組織 B, C は組織横断的利用の前段階の処理として組織内利用を行っている．図 7 では緑の枠線で囲んだ部分が組織横断的利用にあたる．このように，組織内利用と組織横断的利用をそれぞれ別の利用形態にするのではなく，組織内利用から組織横断的利用

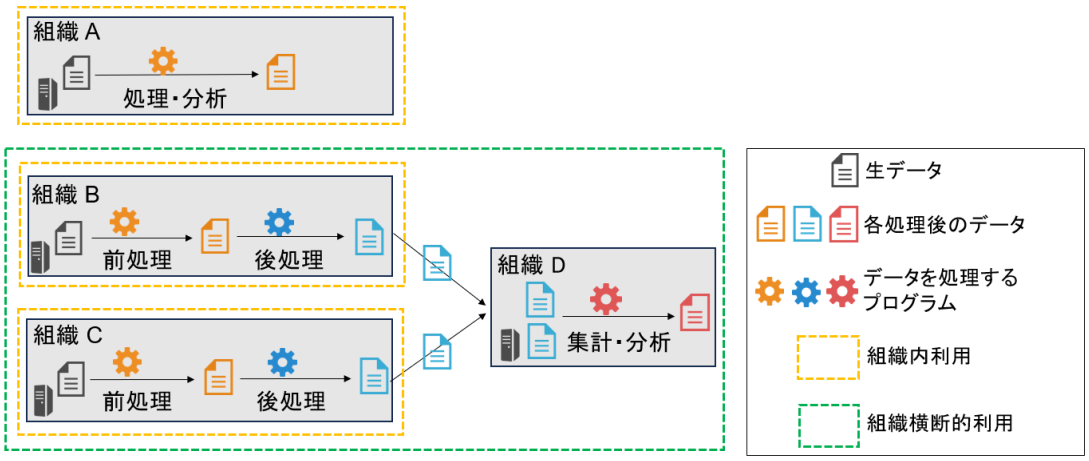

図 7　教育データ利活用の流れ

へとスムーズにつなげられるような設計にする必要がある．保有しているデータを外部に持ち出すには匿名化等の処理を施す必要があるが，匿名化したデータでは分析精度が下がるという問題がある．この問題に対し，生データを外部に持ち出さないローカル処理を行うことで，生データを直接分析して組織内で活用でき，外部に持ち出す場合は，生データを分析した後の結果データを匿名化して持ち出すことができる．匿名化等の処理は図 7 では組織 B, C の中の後処理の部分にあたる．

2.5.2　データ形式・処理内容の標準化

組織横断的利用を行う場合は，組織内のデータに対して匿名化等の処理を施した後に，その結果データが収集されることになるが，収集したデータの形式にばらつきがあると，その後の集計・分析が困難になる．したがって，扱うデータの形式及びその処理内容について標準化し，データを収集する前に，各組織で共通の形式のデータに対して共通の処理を行う必要がある．

データ形式を標準化するには，事前にデータ形式についての仕様やルールを定めておき，必要に応じて，組織内でデータのマッピングやクレンジングを行う．マッピング，クレンジングは図 7 における組織 B, C の前処理の部分にあたる．この前処理には，データ形式を整えた後に集計処理や分析を行うことも含まれる．[8] においても教育データの標準化について触れられており，例としては，学習指導要領のコード化の取り組みがあり，学習指導要領のテキストが学校種，教科，学年等の分類で 16 桁の数字で表されるようになっている．ただし，このような取り組みがあってもなお，組織がすでに保有しているデータに関して，複数組織で形式を揃えることは難しいと考えられる．そこで，新規にデータを取得するところから始めて，各組織に対して共通の形式のデータが取得できるようなコンテンツを組織に配布することで，データ形式の標準化を行うという手段もあり，本研究では特にこのような利用方法を可能にする．

処理内容を標準化するには，処理内容を含んだプログラムとその実行環境を外部から各組織に配信して，各組織で共通の処理を実行する．これによって，多忙な学校現場の教員が処理を書く必要がなくなり，教員の負担を減らすこともできる．ただし，処理を外部から持ち込むため，後述する処理の透明性の担保が必要となる．

2.5.3　処理の透明性の担保

関連研究 [1][2][3] では，企業や研究機関の独自の分析アルゴリズムの機密性を重視して処理内容を秘匿しており，[4] では十分なテストを行わないまま処理内容を公開している．[5] によれば，教育データの組織外部での処理はデータの種類によっては禁止され，処理可能な場合でも安全管理措置の規定を必要としている．した

がって，組織横断的利用のように，教育データを組織外部で処理させる場合は処理の透明性を担保する，つまり処理を書いた人が処理内容をテストして，説明と共に公開・共有することで，データ管理者は自組織のデータの内容を取得したり改ざんしたりするプログラムになっていないかどうかを確認してデータの利用に同意をするかしないかを決める必要がある．また，前述の通り，組織内利用では，データを処理するプログラムは外部から受け取る．外部の人が書いたプログラムを使用するのであれば，組織内利用においても処理の透明性を担保する必要がある．

2.5.4 ID フェデレーション

組織内利用から組織横断的利用へとスムーズに繋ぐには，ID フェデレーションが必要だと考える．ID フェデレーションを利用することで，ユーザは 1 つの IdP (ID プロバイダ) で複数組織のサービスに SSO (Single Sign On) でログインできるようになる．これにより，組織内利用を行うために一度認証して処理の指示を行った後は，他の複数組織のサービスに SSO でログインでき，同じように処理の指示を行い，各組織で処理を行った結果のデータを収集することで，収集したデータの集計・分析へとスムーズに繋げることができる．

第 3 章　処理配信システムの構成と機能

第 2 章で述べた要件を基に，教育データの組織横断的な利用を支援するシステム，処理配信システムを提案する．本章では，提案システムの構成と機能について述べる．流れとしては，まず提案システムの全体像と登場人物の役割を示し，必要な前提条件について述べた後，システム中の個々の機能について説明する．その後，システムを用いた際のデータ利活用の流れを示す．

3.1　システムの全体像

図 8 に提案システムの全体像を示す [12]．図 7 と同じく，黄色の枠線は組織内利用，緑の枠線は組織横断的利用を表す．また，青の枠線は GakuNin (学術認証) フェデレーション [13] を表しており，本システムを利用する組織はフェデレーションに参加することで，組織間で横断的な処理の実行を可能にする．本システムではコンテナ型仮想化技術を利用しており，データを処理するプログラムとその実行環境をまとめてパッケージ化したコンテナイメージを各組織に配信することで，データを外部に持ち出さず，組織内でのデータ処理を行う．図 8-(a)(b) では，目的のコンテナイメージを選択してダウンロードすることを表しており，各組織はローカルに持ってきたコンテナを実行することで，組織内利用を行う．コンテナイメージを組織内にダウンロードさせるのは組織内の人と組織外の人のどちらが行っても良い．ただし，組織外の人が行う場合は対象組織に対して認可を得る必要がある．特に組織横断的利用を行う場合に，組織外の人が各組織のシステムに対して処理の指示を行う．図 8-(c) では組織横断的利用に必要な処理の指示を行っており，組織 B, C のシステムはその指示に基づいて必要なコンテナイメージをダウンロードした後，各組織内でコンテナを実行し，データを処理する．その処理の中で匿名化等を行って外部に持ち出せる形にした後，その結果データを外部組織に集約し（図 8-(d)），横断的な処理の実行を行う．

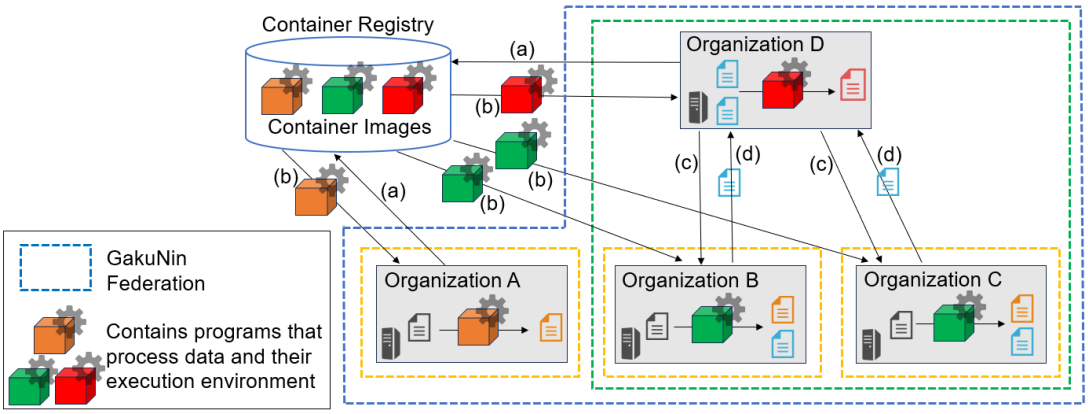

図 8　処理配信システムの全体像

3.2　登場人物

システムを利用する上での登場人物を以下に示し，それぞれの役割を述べる．

- コンテナイメージ作成者
- システム管理者
- データ管理者

- 処理実行者
- 最終的なデータ利用者

3.2.1 コンテナイメージ作成者

コンテナイメージ作成者はデータを処理するプログラムを書く．プログラムの動作をテストした後，プログラムとその実行環境をまとめたコンテナイメージ（図 8-Container Images）を作成する．それと同時にコンテナイメージを使うためのマニュアルを作成する．マニュアルには処理内容についての説明や，利用するためのコマンドのほかに，利用するデータの名前や形式，保存場所などを詳細に記述する．その後，コンテナイメージとマニュアルを Docker Hub[14] などのパブリックレジストリ（図 8-Container Registry）に公開・共有することで，システム管理者やデータ管理者がそれらを確認できるようにする．文部科学省や教育委員会が指定した IT 事業者，IT を理解している教員などがコンテナイメージ作成者に当たる．

3.2.2 システム管理者

システム管理者は組織内の処理サーバでコンテナを自動で実行するためのシステムを作成・管理する．本システムではコンテナを用いるため，処理サーバには Docker[15] をインストールしておく．また，コンテナイメージ作成者が作成したマニュアルを基に，コンテナを処理サーバ上に自動で送り込んで実行する機能を有した Web サイトを作成する．他組織の人がそのサイトを使って組織横断的にシステムを利用できるようにするために，フェデレーションに参加し，Web サイトを SP (Service Provider) として登録する．

3.2.3 データ管理者

データ管理者は組織や部署内のデータを管理する．コンテナイメージ作成者が作成したマニュアルを基に，データ処理の可否を決める．その際，組織内利用のみ許可するか，匿名化した結果データを収集させ，組織横断的利用を許可するかも決めることができる．データ処理に同意する場合は，マニュアルを基に，処理サーバの指定のディレクトリに指定の形式のデータを配置しておく．

3.2.4 処理実行者

処理実行者はシステム管理者の作成した Web サイト（SP）に自組織のアカウントで認証を通してアクセスし，コンテナ実行のボタンを押すことで，対象組織内のサーバにコンテナイメージを送り込み，コンテナ内で処理を実行させる．組織内のデータ管理者，他組織だがフェデレーションに参加している組織の人などが処理実行者に当たる．

3.2.5 最終的なデータ利用者

最終的なデータ利用者は処理実行後の結果データを利用する．組織内利用のみで処理が完結する場合は，最終的なデータ利用者はデータ管理者となる．組織横断的利用の場合は，処理実行者がそのまま結果データを収集して利用するケースや処理実行者とはまた別の機関に結果データを集約して利用するケースなどが考えられる．

3.3 システムの前提条件

3.3.1 処理環境

本システムでは，コンテナを利用するためのプラットフォームとして Docker を用いる．また，コンテナによる処理を行うためにシェルでコマンドを実行する．そのため，前提として各組織の処理サーバの OS は

Linux で，Docker Engine がインストールされているものとする．

3.3.2 データに関する情報の共有

コンテナイメージ作成者が作成したコンテナを想定通りに実行できるように，コンテナイメージ作成者とデータ管理者の間でデータに関する情報を事前に共有しておく必要がある．以下にデータに関して共有する必要がある情報を示す．

- ファイルの名前
- データの保存場所
- データ形式

ファイルの名前に関して，プログラムに読み込ませる入力用のファイルと，実行した結果生成されるファイルの名前を共有する必要がある．前者はコンテナイメージ作成者がプログラムを書けるようにするため，後者はデータ管理者が出力されたファイルを確認するため，また，その後に組織横断的利用に用いるためである．

データの保存場所に関して，これも入力用のファイルと出力されるファイルの両方について保存場所を決めて共有する必要がある．理由としては，ファイルの名前を共有する理由に追加して，コンテナを実行する際にデータの保存場所を指定する必要があるからである．

データ形式に関して，事前に形式を定めておき，その形式に沿ったデータを用意しなければ，プログラムが正しく動作しない．また，組織横断的にデータを収集した際にデータ形式が揃っていない場合，その後の集計・分析が困難になる．これから組織内でデータを測定・取得するという場合には，そのデータを取得するためのコンテンツをコンテナ化して，そのコンテナを各組織に配信することで，各組織で一律の形式のデータを用意することができる．すでに組織が保有しているデータに関しては，定めた形式に則ってデータのマッピングを行うなどしてデータ形式を揃える必要がある．

3.3.3 コンテナ内プログラムのテスト

コンテナイメージ作成者は，プログラムとその実行環境をまとめたコンテナイメージをパブリックのコンテナレジストリにアップロードするが，処理の透明性を担保するために，プログラムが正しく動作するかどうかをテストした後，説明と共に公開・共有する必要がある．筆者は，仕様に対して正しい動作を行うかどうかをテストするプログラムを自動生成する研究を行ってきた [16]．[16] では，テスト対象のプログラムに対してテストコードを自動生成し，形式仕様を事前に書いてテストすることで，与えられた入力に対して適切な出力が得られるかどうかを確認できる．ただし，[16] では Java プログラムにのみ対応しているといった点や，参照型の変数を含んだプログラムには対応していないといった課題があるため，[16] が適用できない場合は，別途テストコードを用意し，テストを行う必要がある．データ管理者はプログラムの動作を理解して，データ利用について同意の可否を決める必要があるが，データ管理者はマニュアル内の処理内容についての記述を読むことで，プログラムの動作について一定の理解が可能であるとする．また，ここで行っているのはコンテナ内プログラムのテストであり，コンテナそのものの正しさを確認しているわけではないため，コンテナのセキュリティについては別で考慮する必要がある．

3.3.4 ID フェデレーション

学校アカウントなどで認証を行い，一度認証した後は複数組織のシステムを SSO (Single Sign On) で利用可能にするために，ID フェデレーションが必要になる．例えば大学が本システムを利用する場合は，GakuNin フェデレーションのような既存のフェデレーションに参加して適切な設定を行うことで組織横断的にシステムが利用可能となる．

3.4　コンテナ内へのデータの取り込みと結果データの保存

コンテナイメージ作成者が作成したコンテナイメージは各組織内の処理サーバに配信され，コンテナが実行されることになるが，送られてきたコンテナにはプログラムとその実行環境しか含まれていないため，利用するデータをホストである処理サーバからコンテナ内に取り込む必要がある．また，コンテナ実行後にコンテナを削除すると，コンテナ内にあるデータはすべて削除される．したがって，コンテナ内のデータが削除されても問題がないように，必要なデータはホスト側で取り出せる必要がある．このような，コンテナ内へのデータの取り込みと結果データの保存を実現するために，Docker コンテナのマウント機能を使う．Docker が提供しているマウントの方法としては，主にボリュームマウントとバインドマウントの 2 種類に分類される．以下でそれぞれのマウント方法の概要を説明し，どちらが本システムの利用に適しているかを述べる．

3.4.1　ボリュームマウント

ボリュームマウントは，Docker が管理している領域にボリュームと呼ばれる記憶領域を作成し，その領域をコンテナにマウントする方法である．図 9 にボリュームマウントのブロック図を示す．ユーザがボリュームを作るには，ボリュームの名前を付けるだけでよく，作成したボリュームの保存場所は Docker が管理するため，ユーザが保存場所を意識する必要はない．ボリュームマウントは，データベースコンテナのデータを保存するときなど，ホスト側から触る必要はないが，データを永続化したいという場合に使われる．

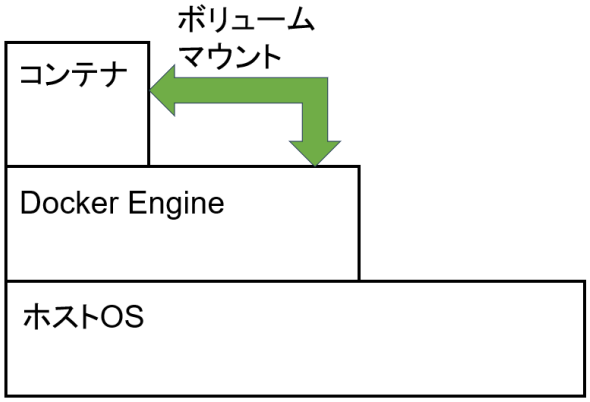

図 9　ボリュームマウントのブロック図

3.4.2　バインドマウント

バインドマウントは，ホスト側のディレクトリをコンテナにマウントする方法である．図 10 にバインドマウントのブロック図を示す．ホスト側のディレクトリをマウントするため，ボリュームマウントとは違って，ホストマシンのユーザがマウント領域を管理できる．バインドマウントは，設定ファイルやソースコードをホスト側で用意しておき，それらをコンテナと共有したいという場合などに使われる．

3.4.3　バインドマウントの利用

本システムでは，バインドマウントを行うことで，ホスト側のデータのコンテナ内への取り込みと，結果データのホストへの保存を可能にする．本研究ではホスト側の生データを外部に持ち出さないことが要件となっているため，コンテナイメージを作成する時点ではイメージ作成者側にデータを用意できない．データを用意できなければコンテナイメージの中にデータを含めることはできず，ボリュームマウントを行ったとして

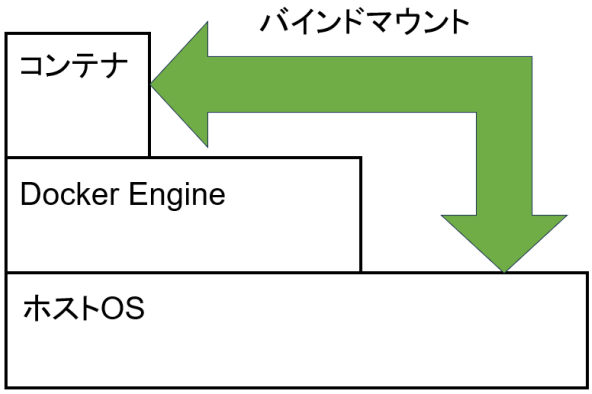

図 10　バインドマウントのブロック図

も，ホスト側にあるデータをコンテナ内に取り込めない．バインドマウントであれば，ホスト側のデータの保存場所についての情報を事前にコンテナイメージ作成者と共有しておくことで，ホスト側のデータをコンテナ内に取り込むことができ，コンテナを実行した結果生成される結果データもホスト側で保存が可能になる．ただし，バインドマウント先のコンテナ側ディレクトリの変更はすべてホスト側のマウント元ディレクトリにも影響するため，ディレクトリの削除を行うプログラムになっていないかどうかなど，プログラムの内容をしっかり確認して，処理の透明性を担保することが重要である．

3.5　結果データの共有

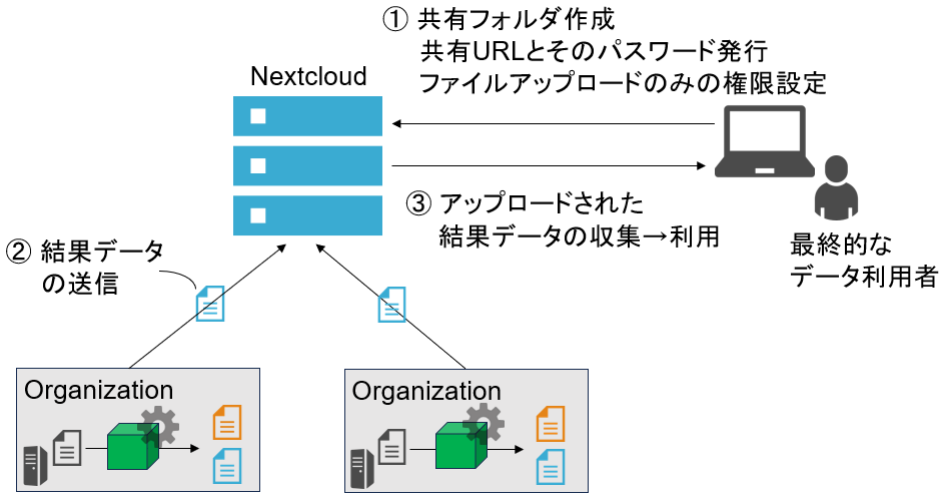

図 11　結果データの共有

バインドマウントによって，コンテナを実行した結果はホスト側に保存されることになるが，特に組織横断的利用を行う場合は，元データのデータ管理者と最終的なデータ利用者との間で，匿名化等必要な処理を行った結果データを共有する必要がある．本システムでは，Nextcloud[17] を用いて結果データを安全に共有する．Nextcloud とは，オンプレミスのサーバに導入できるオンラインストレージであり，組織のセキュリティポリシーに合わせた運用が可能で，安全にファイルを共有することができる [18].

各組織の結果データを収集して利用する役割を持っている最終的なデータ利用者は，自身の Nextcloud アカ

ウントで共有フォルダを作成し，それに外部からアクセスするための共有 URL と保護パスワードを発行する．外部の人はファイルアップロードのみ可能となるように権限を設定する（図 11-①）．各組織で処理が実行され，匿名化まで行われた結果データは，共有 URL のフォルダにアップロードされる（図 11-②）．その後，最終的なデータ利用者は各組織からアップロードされた結果データを収集して，組織横断的な利活用を行うことができる（図 11-③）．

3.6 システムを用いた際のデータ利活用の流れ

本節では，これまで述べてきた登場人物の役割やシステムの機能を踏まえて，システムを用いた際のデータ利活用の流れを説明する．事前準備，処理の実行，結果データの収集と利活用の 3 つのステップに分けて説明する．

3.6.1 事前準備

- コンテナイメージ作成者が行うこと
  1. 処理プログラムを書き，その処理内容をテストする．
  2. プログラムとその実行環境をまとめたコンテナイメージを作成する．
  3. 処理の説明や利用するデータの形式，保存場所など必要な情報をマニュアルとして書く．
  4. コンテナイメージとマニュアルを Docker Hub のパブリックレジストリに登録して公開・共有する．
- 最終的なデータ利用者が行うこと
  1. 自身の Nextcloud アカウントで結果データ収集用の共有フォルダを作成する．
  2. 共有フォルダに外部からアクセスするための共有 URL とその保護パスワードを発行し，外部からはファイルアップロードのみできるような権限を設定する．
  3. 共有 URL とその保護パスワードをシステム管理者に通知する．
- システム管理者が行うこと
  1. 処理サーバに Docker をインストールしておく．
  2. マニュアル，共有 URL と保護パスワードを基に，処理サーバに自動でコンテナを送り込んで実行させる機能と結果データの自動収集機能を有した Web サイトを作成する．
  3. Web サイトをフェデレーションの SP として登録しておく．
- データ管理者が行うこと
  1. データ利用の可否を決める．
  2. データ利用に同意する場合，組織内利用のみ許可するか，匿名化した結果データを収集させ，組織横断的利用を許可するかを決める
  3. データ利用に同意する場合，マニュアルを基に処理サーバの指定のディレクトリに指定の名前・形式のデータを配置しておく．

3.6.2 処理の実行

- 処理実行者が行うこと
  1. 自組織のアカウントで認証し，データ利用対象の組織の SP にアクセスする．
  2. 各組織の SP で実行ボタンをクリックし，各組織の処理サーバ上でコンテナを実行させる．
  3. 組織横断的利用を行う場合は，匿名化ボタンをクリックすると，実行結果のデータがコンテナ処理によって匿名化され，収集ボタンを押すと最終的なデータ利用者が指定した共有 URL に匿名化された結果データがアップロードされる．

3.6.3 結果データの収集と利活用

- データ管理者が行うこと
  - 組織内で結果データを活用する場合は，データ管理者が処理サーバの指定のディレクトリから結果データを取り出して活用する．
- 最終的なデータ利用者が行うこと
  - 組織横断的に収集したデータを活用する場合は，最終的なデータ利用者が指定した共有フォルダに集約された結果データを取り出して活用する．

第 4 章　評価

4.1　組織内利用の動作検証

本システムの一つ目の動作検証として，単一組織の組織内利用を行った．一つの仮想マシンを組織内の処理サーバと見立てて動作検証を行う．表 3 に動作検証に用いた仮想マシンのスペックを示す．この動作検証では各登場人物が行うことを筆者一人が行っているが，誰が何を行うのかを述べるために各登場人物を使って説明する．

表 3　組織内利用の動作検証環境

名称	IP アドレス	スペック
tm-node1 (VM)	10.20.22.96	OS: Ubuntu Server 22.04 vCPU: 2 メモリ: 8GB Docker: version 25.0.2

4.1.1　動作検証の概要

簡易的なデータ利活用アプリケーションを作成し，そのアプリをコンテナイメージにして Docker Hub に登録して公開・共有する．処理実行者はコンテナイメージをデータのある VM に配信し，コンテナを実行させる．実行後は結果データがホスト側（この場合は VM）に保存されていることを確認する．

4.1.2　データ利活用アプリケーションの内容

この検証で用いるデータ利活用アプリケーションの内容は，「国語，数学，英語の 3 教科の得点が生徒ごとに記録されている CSV ファイルを読み込み，生徒ごとの合計点及び平均点を算出し，その結果を CSV ファイルに出力する.」というものである．

4.1.3　前提

3 章で述べた前提条件を基に，この検証での前提を述べる．処理環境は表 3 に示した通り，条件を満たしている．扱うデータに関して，データ管理者とコンテナイメージ作成者の間で共有すべき内容を表 4 に示す．コンテナ内プログラムのテストは行っているものとし，ID フェデレーションはできているものとする．

表 4　共有すべきデータの情報

	入力用データ	結果データ
ファイルの名前	test_score.csv	calc_output.csv
データの保存場所	～/data	～/data
データ形式	1 行目の内容 データの中身の内容（文字列か数値かなど）	

4.1.4　事前準備

コンテナイメージ作成者がアプリの内容とデータに関する前提を基に，データ利活用アプリケーションを作成する．その後，Dockerfile を作成し，docker image build コマンドでコンテナイメージを作成する．コンテナイメージの名前は “calc-sum-average” とした．作成したコンテナイメージは docker image push コマンドで Docker Hub に登録して，公開・共有する．

システム管理者は処理サーバ上で自動でコンテナを実行させるための Web サイトを用意する．

データ管理者は前提条件として指定されたディレクトリに指定のデータを配置しておく．前提条件で指定したディレクトリと入力用データの中身を，それぞれ図 12 と図 13 に示す．氏名や点数は架空のものである．

```
m221924@tm-node1:~/data$ ls
test_score.csv
```

図 12　処理実行前の指定のディレクトリ

```
m221924@tm-node1:~/data$ cat test_score.csv
ID,氏名,国語,数学,英語
1,愛媛 健一,63,77,94
2,高知 さくら,85,75,90
3,香川 健二,91,62,88
```

図 13　入力用データ

4.1.5　処理の実行

処理実行者はコンテナイメージを Docker Hub からダウンロードさせ，コンテナを実行させる．この時，データ管理者が実行するのであれば，図 14 のように，自身がコマンドを打って実行することもできるが，フェデレーションに参加している処理実行者は組織の SP (Service Provider)（図 15）にアクセスして実行させることもできる．この方法であれば，データ管理者が実行する場合でも適用でき，組織横断的利用を行う際にも，各組織の SP に SSO (Single Sign On) でアクセスし，各組織に対して処理の実行を自動で行うことができる．

```
m221924@tm-node1:~$ docker container run --rm --name calc-sum-average-app --mount
 type=bind,src=$(pwd)/data,dst=/data tmsky18/calc-sum-average
結果データ "calc_output.csv" が指定の場所に出力されました。
m221924@tm-node1:~$ ls ./data
calc_output.csv  test_score.csv
```

図 14　データ管理者が処理のコマンドを実行した場合

図 15　処理実行のための SP

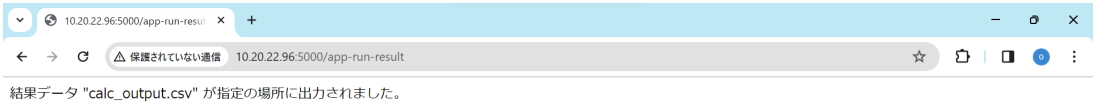

図 16　実行後のブラウザ画面

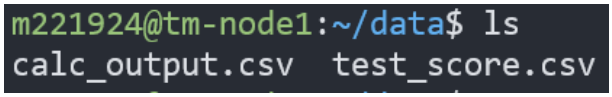

図 17　処理実行後の指定のディレクトリ

4.1.6　実行結果

処理実行のための SP (図 15) の実行ボタンを押した結果を述べる．図 16 に実行後のブラウザの画面，図 17 に実行後の指定ディレクトリ，図 18 に結果データの中身を示す．実行ボタンを押すことで，対象組織の処理サーバに指定のコンテナイメージがダウンロードされ，コンテナ内にデータを取り込んで処理が実行される．ブラウザには，正常に結果データが出力されたという旨のメッセージが表示され，実際に指定のディレクトリを見ると結果データ（calc_output.csv）が生成されていることが分かる．結果データを見ると，元の入力用データに対して合計と平均の列が追加されており，データ利活用アプリケーションの内容が正常に反映されていることが分かる．

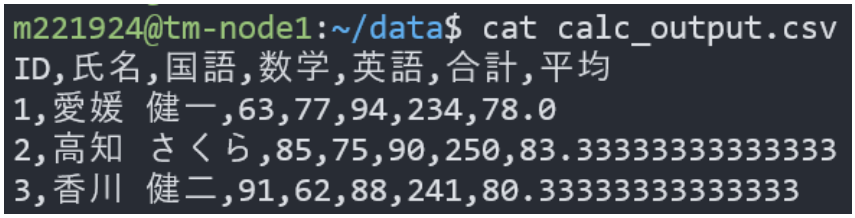

図 18　結果データ

4.2　組織横断的利用の動作検証

本システムの二つ目の動作検証として，組織横断的利用を行った．組織内利用の動作検証と同じく，仮想マシンを組織の処理サーバと見立てて動作検証を行う．表 5 に動作検証に用いた仮想マシンのスペックを示す．ここではそれぞれの VM を区別するため，tm-node2 を組織 B の処理サーバ，tm-node3 を組織 C の処理サーバと呼ぶ（図 8 の Organization B, Organization C に対応）．

表 5　組織横断的利用の動作検証環境

名称	IP アドレス	スペック
tm-node2 (VM) tm-node3 (VM)	10.20.22.97 10.20.22.98	OS: Ubuntu Server 22.04 vCPU: 2 メモリ: 8GB Docker: version 25.0.2

4.2.1　動作検証の概要

まずそれぞれの VM で組織内利用を行う．データ利活用アプリケーションや用いるデータの形式などの前提は 4.1 節で述べたものと同じとする．その後，各組織の指定ディレクトリに出力された結果データに対してコンテナによる匿名化を行う．匿名化された結果データを Nextcloud の指定の共有フォルダにアップロードさせ，最終的なデータ利用者と結果データを共有できることを確認する．

4.2.2　事前準備

4.1.4 節で述べた事前準備に加えて，コンテナイメージ作成者は自身が作ったデータ利活用アプリケーションのコンテナを実行した結果得られる結果データに対して匿名化を行うコンテナイメージを作成する．ここでの匿名化は, 氏名をランダムな文字列に置き換える処理を行う．コンテナイメージの名前は “anonymize-sample” とした．作成したコンテナイメージを Docker Hub に登録して公開・共有する．

最終的なデータ利用者は自身の Nextcloud アカウントで結果データ収集用の共有フォルダを作成する．フォルダ名は “cross_org_data_use”とした．共有フォルダに外部からアクセスするための共有 URL と保護パスワードを発行し，外部の人の権限はファイルドロップ（アップロードのみ）とする．そして共有 URL と保護パスワードをシステム管理者に通知する．

システム管理者はマニュアル，共有 URL と保護パスワードを基に，匿名化機能と結果データを自動で収集できるような機能を SP に搭載する．結果データを自動で収集させるために [19] のシェルスクリプト（cloudsend.sh）を用いた．このシェルスクリプトを用いることで，ローカルにあるファイルを外部のパスワード付き共有 URL に自動でアップロードでき，その際にローカルにあるファイルとは別の名前を付けることも

できるため，ファイル名の最初に orgB, orgC などを付記しておくことで，最終的なデータ利用者はどの組織のデータなのか判別できるようになる．シェルスクリプトは結果データと同じディレクトリに置いておく．処理実行前の組織 B の指定のディレクトリと入力用データの中身を，それぞれ図 19 と図 20 に示す．組織 C については，入力用データの氏名や点数以外は組織 B と同じである．

```
m221924@tm-node2:~/data$ ls
cloudsend.sh  test_score.csv
```

図 19　処理実行前の組織 B の指定ディレクトリ

```
m221924@tm-node2:~/data$ cat test_score.csv
ID,氏名,国語,数学,英語
1,広大 太郎,90,55,82
2,東広島 次郎,64,92,80
3,西条 花子,75,73,71
```

図 20　組織 B の入力用データ

4.2.3　処理の実行

処理実行者は各組織の SP に認証を通してアクセスする．フェデレーションに参加していれば，一度認証をした後は SSO で各組織の SP にアクセスすることができる．組織 B の SP を図 21 に示す．認証後は，実行，匿名化，収集ボタンを順に押す．

図 21　組織 B の SP

4.2.4　実行結果

図 22 に利活用アプリ実行後のブラウザ画面, 図 23 に匿名化後のブラウザ画面, 図 24 に結果データ収集後のブラウザ画面を示す．正常に処理が実行されたという旨のメッセージが表示されていることが分かる．図 25，図 26, 図 27 を見ると，指定のディレクトリに結果データ，匿名化後の結果データ（anonymized_calc_output.csv）

が出力されていることが分かる．同じことを組織 C に対して行い，指定の Nextcloud フォルダ（図 28）を確認すると，組織 B と組織 C の匿名化後の結果データがアップロードされていることが分かる．

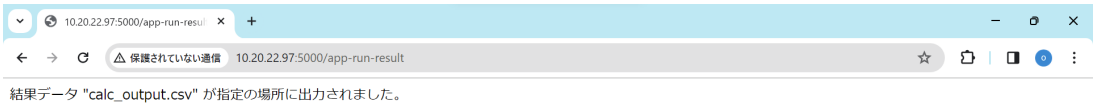

図 22　利活用アプリ実行後のブラウザ画面

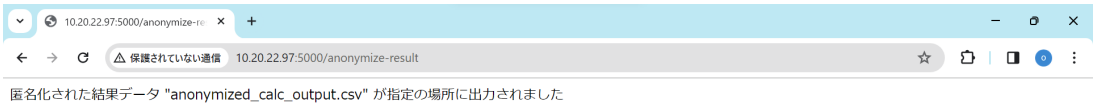

図 23　匿名化後のブラウザ画面

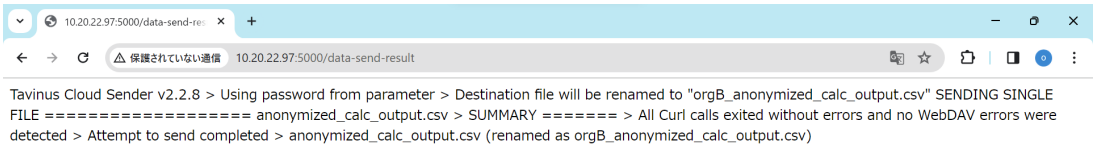

図 24　結果データ収集後のブラウザ画面

```
m221924@tm-node2:~/data$ ls
anonymized_calc_output.csv  calc_output.csv  cloudsend.sh  test_score.csv
```

図 25　各処理実行後の組織 B の指定ディレクトリ

```
m221924@tm-node2:~/data$ cat calc_output.csv
ID,氏名,国語,数学,英語,合計,平均
1,広大 太郎,90,55,82,227,75.66666666666667
2,東広島 次郎,64,92,80,236,78.66666666666667
3,西条 花子,75,73,71,219,73.0
```

図 26　組織 B の結果データ

```
m221924@tm-node2:~/data$ cat anonymized_calc_output.csv
ID,氏名,国語,数学,英語,合計,平均
1,DvN8GBozlX,90,55,82,227,75.66666666666667
2,UDbPFRoERV,64,92,80,236,78.66666666666667
3,RaejDpR1IW,75,73,71,219,73.0
```

図 27　組織 B の匿名化後の結果データ

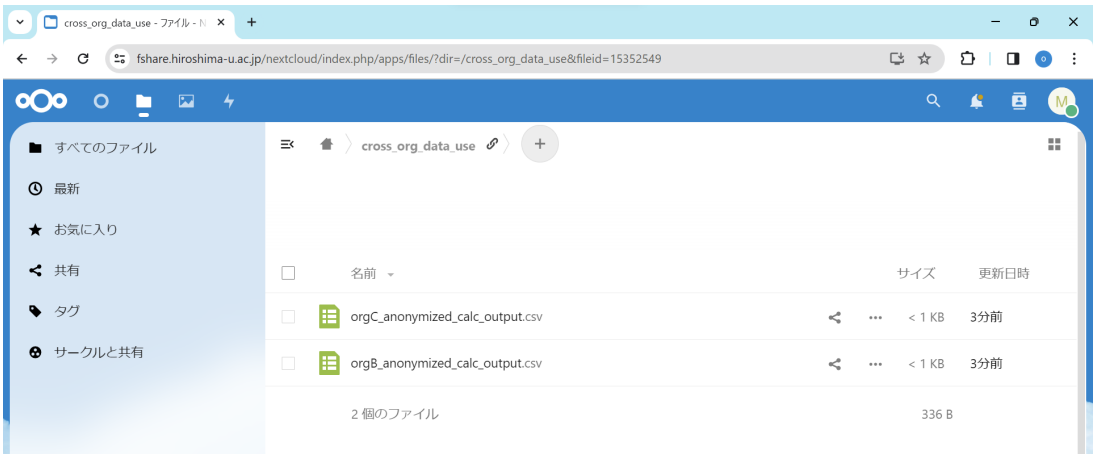

図 28　結果データ収集用の共有フォルダ

4.3　関連研究との比較

表 6 に本研究と関連研究との比較表を示す．関連研究の部分は 2 章に掲載したものと同様である．

組織横断的な利用について，本研究ではコンテナを自動で実行させるための Web サイトをフェデレーションの SP として登録すれば，フェデレーションに参加している組織の人が各組織の SP に実行の指示を出すことで，各組織内で処理を実行でき，実行した結果は匿名化後に収集され，組織横断的な利用が可能となる．

組織内でのデータ処理について，本研究ではデータの形式や保存場所などのルールを決めて，処理環境はコンテナとして組織に配信することで，組織が保有するデータを外部に持ち出すことなく，組織内でのデータ処理が可能である．

処理の透明性について，本研究ではコンテナ内プログラムのテストをした上で説明と共に公開・共有することで処理の透明性を担保する．しかし，ここで確かめているのはプログラムの処理内容だけなので，コンテナの真正性を確認するなど，コンテナセキュリティを担保することで，よりセキュアなデータ利用が可能になると考えられる．

データの秘匿性ついて，本研究では組織内でのデータ処理によって，外部からデータを秘匿している．結果データを収集する場合は，匿名化を施し，Nextcloud のフォルダ（権限はファイルアップロードのみ）に送信することで結果データを外部から秘匿する．

アクセス制御について，本システムを利用するにはフェデレーションに参加している必要があり，学校アカウントなどでの認証に基づいたアクセス制御が可能である．

データ検索について，本研究では関連研究 [2][3] のように，どのような種類のデータがどこにあるのか，クエリを発行して検索する手段はないが，データ管理者が指定の場所に指定の形式でデータを配置しておくことで，データを利用することができる．

表 6　本研究と関連研究との比較

機能	関連研究 [1]	関連研究 [2]	関連研究 [3]	関連研究 [4]	本研究
組織横断的な利用	×	○	○	○	○
組織内でのデータ処理	×	×	×	×	○
処理の透明性	×	×	×	△	○
データの秘匿性	○	○	○	○	○
アクセス制御	○	○	○	○	○
データ検索	○*	○	○	○*	○*

4.4　考察

事前にデータ形式や保存場所などの必要な情報をデータ管理者とコンテナイメージ作成者との間で共有しておくことで，データそのものを外部に持ち出すことなく，ローカルでのデータ処理が可能であることを示した．これにより，データ管理者は結果データを取り出して組織内で活用することができ，組織横断的利用を行う場合は，各 SP の匿名化ボタンと収集ボタンを押すことで匿名化後の結果データを収集できることを示した．収集先の共有フォルダはファイルアップロードのみの権限となっているため，第三者が共有 URL や保護パスワードを知ったとしてもデータの閲覧や編集はできない．処理の透明性の担保に関して，この動作検証では前提としたが，プログラムをテストして公開・共有するというステップを踏むことで安易に悪意のあるプログラムが配信されることはないと考えられる．ID フェデレーションに関して，これも前提としたが，フェデレーションの仕組みがあれば，各組織の SP に SSO でアクセスし，組織横断的に処理を実行できると考えられる．

4.5　応用例

本システムは，複数組織に対して新規にデータを測定・取得するような利用に適している．これはデータを取得するためのコンテンツをコンテナ化して各組織に配信することで，各組織で同じ形式のデータを揃えることができるからである．本システムを使った具体的な応用例としては，広島大学が行っているフォローアップ講習や情報セキュリティインシデント対応訓練がある．これらは広島大学以外の他大学においても重要であり，これらの学習コンテンツや確認テストなどをコンテナ化することで，他大学にも同様のコンテンツを展開できる．確認テストの結果は各大学で同じ形式のデータとなっているため，結果を分析したり匿名化したりするコンテナを配信して実行することで，組織内での活用や組織横断的に収集して更なる集計・分析ができるようになると考えられる．

第 5 章　まとめ

本研究では，教育データの組織横断的な利用を支援するシステムである処理配信システムを提案した．教育データの利用においては，データを外部に持ち出さないことや処理の透明性を担保してデータ主体やデータ管理者の同意を得ることが重要である．本研究では，データの形式や保存場所などのルールを事前に決めておき，コンテナの処理環境を組織内に配信することで，データを外部に持ち出さず，組織内でのデータ処理が可能となる．その際，コンテナ内プログラムをテストして妥当性を検証し，説明と共に公開・共有することで処理の透明性を担保する．現状では，各組織がすでに保有している既存のデータは，組織によって形式や粒度が異なるため，それらのデータを組織横断的に利用することは難しい．本研究では，新規にデータを測定・取得するケースに対して，データを取得するためのコンテンツをコンテナ化して，各組織に配信することで，各組織で共通の形式のデータを用意でき，それらのデータに対して分析や匿名化を行うコンテナを配信して実行することで，組織内でのデータ処理や匿名化後の結果データを収集して組織横断的に利用することが可能となる．

今後の課題としては，実際にシステムをフェデレーションの SP として登録することで，他組織の人でもフェデレーションに参加していれば，一度認証した後，SSO で複数組織の処理サーバに対してコンテナ処理を実行できるかどうか確かめることが必要となる．また，コンテナそのもののセキュリティを担保して，よりセキュアなシステムにしていくことが今後の課題となる．

謝辞

指導教員である広島大学情報メディア教育研究センター西村浩二教授には，多大なご指導・ご協力を賜りました．心より感謝申し上げます．

広島大学情報メディア教育研究センター渡邉英伸准教授，村上祐子助教におかれましては，個別ミーティングや論文執筆の際に多くの助言を頂戴いたしました．心より感謝申し上げます．

広島大学情報メディア教育研究センター近堂徹教授，岸塲清悟助教，田島浩一助教，下地寛武特任助教，相原玲二上席特任学術研究員におかれましては，研究ミーティングにおいて多くの助言を頂戴いたしました．心より感謝申し上げます．

副指導教員である広島大学岡村寛之教授には，中間発表時にご助言していただきました．心より感謝申し上げます．

轟木皓平さん，中野敦斗さん，高橋朋也さんをはじめ研究室の皆さんには，研究活動を通して良い刺激を頂きました．厚くお礼申し上げます．

最後に，ここまで研究を続けるにあたり，様々な面で支えて頂いた友人，そしで，常に陰ながら支えて頂いた両親・家族の皆様に心から感謝いたします．

参考文献

[1] 望月尋斗, 藤本まなと, 阿多信吾. データ利活用サービス構築のためのパーソナルデータストアサンドボックスの設計. 電子情報通信学会技術研究報告: 信学技報, Vol. 123, No. 198, pp. 31–36, 2023.

[2] 坂本久, 石田和生, 加藤孝浩, 稲垣嘉信. 組織が保有する情報を他組織に安全に共有し, 複数組織でデータの価値を向上させる 「データ価値共創プラットフォーム」. 研究報告ドキュメントコミュニケーション(DC), Vol. 2021, No. 10, pp. 1–8, 2021.

[3] NTT R&D. 組織を越えたデータ利活用を安全・便利にする次世代データハブ. https://www.rd.ntt/research/JN202202_17186.html, 2022. (参照 2024/01/09).

[4] Josh Gardner, Christopher Brooks, Juan Miguel Andres, and Ryan S. Baker. MORF: A Framework for Predictive Modeling and Replication At Scale With Privacy-Restricted MOOC Data. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3235–3244, 2018.

[5] 文部科学省. 教育情報セキュリティポリシーに関するガイドライン (令和 4 年 3 月). https://www.mext.go.jp/content/20220304-mxt_shuukyo01-100003157_1.pdf, 2022. (参照 2024/01/15).

[6] AXIES.「教育・学習データ利活用ポリシー」のひな型の策定について. https://axies.jp/report/publications/formulation/, 2023. (参照 2024/01/24).

[7] 広島大学. 学びのサポート. https://momiji.hiroshima-u.ac.jp/momiji-top/learning/post_38.html, 2023. (参照 2024/01/24).

[8] 文部科学省. 教育データの利活用に係る論点整理（中間まとめ）. https://www.mext.go.jp/content/20210331-mxt_syoto01-000013887_1.pdf, 2021. (参照 2024/01/15).

[9] 総務省. 広がるデータ流通・利活用. https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/n2100000.pdf, 2017. (参照 2024/01/13).

[10] アフラック生命保険株式会社. 個人情報流出に関するお詫びとお知らせ. https://www.aflac.co.jp/news_pdf/2023011001.pdf, 2023. (参照 2024/01/14).

[11] 福岡県庁. 委託業務による個人情報の漏えい等事案について（第 1 報）. https://www.pref.fukuoka.lg.jp/press-release/zei-20231017.html, 2023. (参照 2024/01/14).

[12] Takahiro Morita, Yuko Murakami, Hidenobu Watanabe, and Kouji Nishimura. Proposing a Processing Distribution System for Cross-Organizational Use of Educational Data. In *Proceedings of the 31st International Conference on Computers in Education. Asia-Pacific Society for Computers in Education*, p. 929–931, 2023.

[13] 国立情報学研究所. トップページ — 学術認証フェデレーション 学認 gakunin. https://www.gakunin.jp/, 2009. (参照 2024/01/22).

[14] Docker Inc. Docker hub container image library — app containerization. https://hub.docker.com/. (参照 2024/02/03).

[15] Docker Inc. Docker: Accelerated container application development. https://www.docker.com/. (参照 2024/01/22).

[16] 森田崇大. SOFL 形式仕様に基づくテストコード自動生成と テスト関連情報の自動記録. 卒業論文, 広島大学情報科学部, 2022.

[17] Nextcloud GmbH. Nextcloud - open source content collaboration platform. https://nextcloud.com/. (参照 2024/01/24).

[18] 株式会社スタイルズ. Nextcloud 公式サイト — 株式会社スタイルズ. https://nextcloud.stylez.co.jp/. (参照 2024/02/03).

[19] tavinus. Github - tavinus/cloudsend.sh: Bash script that uses curl to send files to a nextcloud/owncloud shared folder. https://github.com/tavinus/cloudsend.sh. (参照 2024/02/04).

業績

Takahiro Morita, Yuko Murakami, Hidenobu Watanabe, and Kouji Nishimura. Proposing a Processing Distribution System for Cross-Organizational Use of Educational Data. In Proceedings of the 31st International Conference on Computers in Education. Asia-Pacific Society for Computers in Education, p. 929–931, 2023.

付録

処理サーバの作り方

Linux サーバに Docker をインストールし，マニュアルで指定したディレクトリに指定のデータを置いておくことで，システムの処理サーバとして利用できる．コンテナを実行するための SP (Web サイト) は処理サーバとは別のサーバで動かすのが一般的だと思われるが，現段階ではプロトタイプとなっているため，同一の処理サーバ上で SP を動かしている．また，現段階では各組織の SP にアクセスして処理を実行する形になっているが，処理の指示を行う組織が，自組織の SP から実行対象の組織を選択して処理実行の指示を行うという形にすることが今後の課題である．以下，Docker のインストール方法，処理サーバ（SP 含む）のディレクトリ構成，本研究で用いた SP のソースコードを示す．SP の構築には Python, Flask を用いた．

Docker のインストール

```
$ sudo apt update
$ sudo apt install ca-certificates curl gnupg lsb-release

$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg

$ echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" |
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

$ sudo apt update
$ sudo apt install docker-ce docker-ce-cli containerd.io

$ sudo groupadd docker
$ sudo usermod -aG docker $USER
```

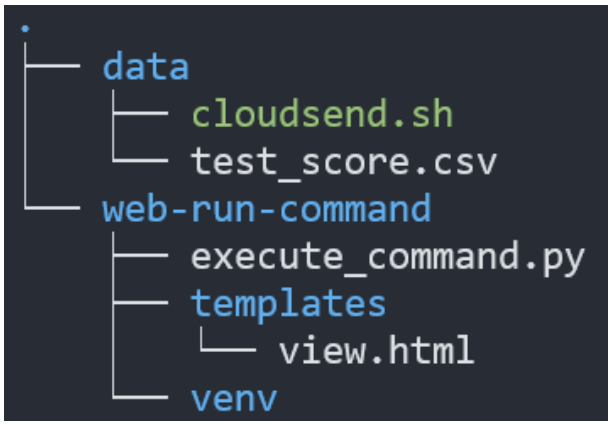

図 29　処理サーバのディレクトリ構成

ソースコード 1　execute_command.py (SP)

```
from flask import Flask, render_template, redirect, url_for
import subprocess

app = Flask(__name__)

@app.route('/')
def index():
    return redirect(url_for('show_command'))

@app.route('/view')
def show_command():
    return render_template('view.html')

@app.route('/app-run-result', methods=['POST'])
def run_app():
    command = """cd && docker container run --rm --log-driver syslog --name calc-sum-average-app \
    --mount type=bind,src=$(pwd)/data,dst=/data tmsky18/calc-sum-average
    """
    try:
        result = subprocess.run(command, shell=True, capture_output=True,
                                text=True, check=True)
        return result.stdout
    except subprocess.CalledProcessError as e:
        return e.stderr

@app.route('/anonymize-result', methods=['POST'])
def anonymize():
    command = """cd && docker container run --rm --log-driver syslog --name anonymize-sample-app \
    --mount type=bind,src=$(pwd)/data,dst=/data tmsky18/anonymize-sample
    """
    try:
        result = subprocess.run(command, shell=True, capture_output=True,
                                text=True, check=True)
        return result.stdout
    except subprocess.CalledProcessError as e:
        return e.stderr

@app.route('/data-send-result', methods=['POST'])
def send_data():
    command = """cd && cd data && ./cloudsend.sh -p 'EeWR62G7' -r 'orgB_anonymized_calc_output.csv' \
```

```
    './anonymized_calc_output.csv' \
    'https://fshare.hiroshima-u.ac.jp/nextcloud/index.php/s/ZgDAQPZCSNpD5ni'
    """
    try:
        result = subprocess.run(command, shell=True, capture_output=True,
                                text=True, check=True)
        return result.stdout
    except subprocess.CalledProcessError as e:
        return e.stderr

def main():
        app.run(host="0.0.0.0")

if __name__ == '__main__':
        main()
```

ソースコード 2　view.html (SP)

```
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="UTF-8">
    <title>データ利活用サイト</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            display: block;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f0f0f0;
        }

        h1 {
            text-align: center;
        }

        .container {
            max-width: 650px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        .description {
            margin-bottom: 10px;
        }

        .button-container {
            display: flex;
            justify-content: center;
            gap: 10px;
        }

        .button {
            padding: 10px 20px;
            font-size: 16px;
```

```
            background-color: #007bff;
            color: #fff;
            border: none;
            cursor: pointer;
        }
    </style>
</head>

<body>
    <h1>処理内容の説明と実行</h1>
    <div class="container">
        <div class="description">
            <p>
                国語、数学、英語の 3教科の得点が生徒ごとに記録されているCSV ファイルを読み込み、
                生徒ごとの合計点及び平均点を算出し、その結果をCSV ファイルに出力する。<br>
                下の実行ボタンをクリックすると実行される。
            </p>
            <p>
                結果データの匿名化を行う場合は、下の匿名化ボタンをクリックする。
            </p>
            <p>
                結果データの収集を行う場合は、匿名化を行った後、下の収集ボタンをクリックする。
            </p>
            <p>
                入力用データ & 結果データの保存場所: ~/data
            </p>
        </div>
        <div class="button-container">
            <form action="/app-run-result" method="post">
                <button class="button" type="submit">実行</button>
            </form>
            <form action="/anonymize-result" method="post">
                <button class="button" type="submit">匿名化</button>
            </form>
            <form action="/data-send-result" method="post">
                <button class="button" type="submit">収集</button>
            </form>
        </div>
    </div>
</body>
</html>
```

データ利活用アプリケーション

組織内利用及び組織横断的利用の動作検証で使ったデータ利活用アプリケーションのプログラムをソースコード 3 に，その時使ったパッケージのリストをソースコード 4 に，これらをまとめてコンテナイメージにするために作成した Dockerfile をソースコード 5 に示す.

ソースコード 3　calc_sum_average.py

```
import pandas as pd

# ファイルのパスはコンテナ内(マウント先)のパス
df = pd.read_csv('/data/test_score.csv')

df_sum = df.assign(合計 = df.iloc[:, 2:].sum(axis=1))
df_sum_ave = df_sum.assign(平均 = df.iloc[:, 2:].mean(axis=1))

df_sum_ave.to_csv('/data/calc_output.csv', index=False)
print('結果データ␣"calc_output.csv"␣が指定の場所に出力されました。')
```

ソースコード 4　requirements.txt

```
numpy==1.26.3
pandas==2.1.4
python-dateutil==2.8.2
pytz==2023.3.post1
six==1.16.0
tzdata==2023.4
```

ソースコード 5　Dockerfile

```
FROM python:3.10

WORKDIR /usr/src/app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["python", "calc_sum_average.py"]
```

匿名化アプリケーション

組織横断的利用で使った匿名化プログラムをソースコード 6 に，これをコンテナイメージにするために作成した Dockerfile をソースコード 7 に示す.

ソースコード 6　anonymize_data.py

```python
import csv
import random
import string

def anonymize_name(n):
    # n 文字のランダムな文字列を生成して返す
    return ''.join(random.choices(string.ascii_letters + string.digits, k=n))

def anonymize_data(input_file, output_file):
    with open(input_file, 'r', encoding='utf-8') as infile:
        reader = csv.DictReader(infile)
        fieldnames = reader.fieldnames

        # 氏名が匿名化されたデータを格納する
        result_data = []

        for row in reader:
            # 氏名を匿名化
            row['氏名'] = anonymize_name(10)
            result_data.append(row)

    with open(output_file, 'w', encoding='utf-8') as outfile:
        writer = csv.DictWriter(outfile, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(result_data)

def main():
    # ファイルのパスはコンテナ内(マウント先)のパス
    input_file = '/data/calc_output.csv'
    output_file = '/data/anonymized_calc_output.csv'
    anonymize_data(input_file, output_file)
    print('匿名化された結果データ␣"anonymized_calc_output.csv"␣が指定の場所に出力されました')

if __name__ == "__main__":
    main()
```

ソースコード 7　Dockerfile

```
FROM python:3.10

WORKDIR /usr/src/app

COPY . .

CMD ["python", "anonymize_data.py"]
```