

Cyber HPC Symposium 2025

# 研究再現性を担保する来歴管理システム

2025年3月7日

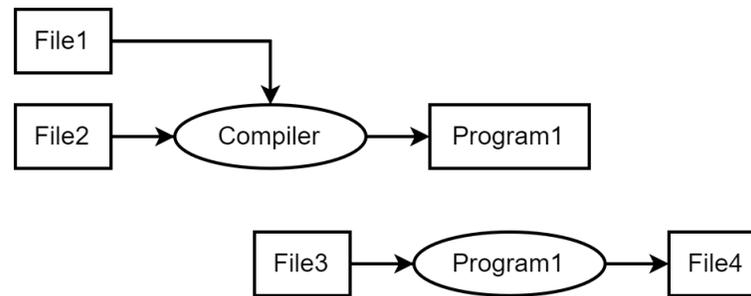
日本電気株式会社

データ基盤サービス統括部

並木悠太

# 概要

- 計算機システムにおける研究データの管理の話
- 研究活動において再現性を担保することが求められています。
- 計算機実験の再現には、結果データを出力したプログラムやその入力データなどを記述した「来歴」が重要な役割を担います。



- 本講演では計算機システムで生み出されるデータの来歴を自動的に記録する仕組みと、その再現性への貢献などのユースケースを紹介します。

# 目次

1. 背景
2. 再現性とは
3. 再現性と来歴
4. HPCシステムと来歴
5. 問題と課題
6. 提案：HPCシステムにおいてデータの来歴を自動的に記録する仕組み
7. 来歴の活用シーン
8. 今後の課題
9. まとめ

## ■研究の再現性 (reproducibility) へ要請

- 狙い：研究公正の実現、データ利活用促進による研究の迅速化
- 「オープンサイエンス」の文脈でも再現性に言及：  
研究の過程と成果を公開し、再現性の担保、活用を促進させる

## G7仙台科学技術大臣会合(概要)

令和5年5月13日(5月16日更新)  
科学技術・イノベーション推進事務局

### 1. G7仙台科学技術大臣会合の概要



令和5年5月12日(金)から14日(日)に、宮城県仙台市・秋保温泉においてG7仙台科学技術大臣会合が開催され、高市大臣が議長を務めるとともに、G7仙台科学技術大臣会合参加の要人と二国間会談を行いました。

会合では、「信頼に基づく、オープンで発展性のある研究エコシステムの実現」をメインテーマとし、今後の科学技術政策の方向性として、「科学研究の自由と包摂性の尊重とオープン・サイエンスの推進」、「研究セキュリティとインテグリティの取組による信頼ある科学研究の促進」、「地球規模課題解決に向けた科学技術国際協力」について議論が行われ、G7各国及びEUと共通認識を得ました。

本会合の成果文書として、「G7科学技術大臣の共同声明」を発出しました。

[「G7 Science and Technology Ministers' Communique」\(PDF形式: 524KB\)](#)

[「G7科学技術大臣共同声明\(仮訳\)」\(PDF形式: 615KB\)](#)

### 2. 「共同声明」のポイント

(1) G7科学技術大臣は、民主主義、法の支配、自由と人権の尊重という共通の価値観や、ジェンダー平等を含む多様性などの重要性へのコミットメントを確認

(2) 新たな知の創造に貢献できるよう、研究データや論文を含む科学的知識を公平に広めながら、オープン・サイエンスの拡大で協力

# 再現性とは

## ■研究を再現するために何が必要か：「なにからどのように作られたか？」

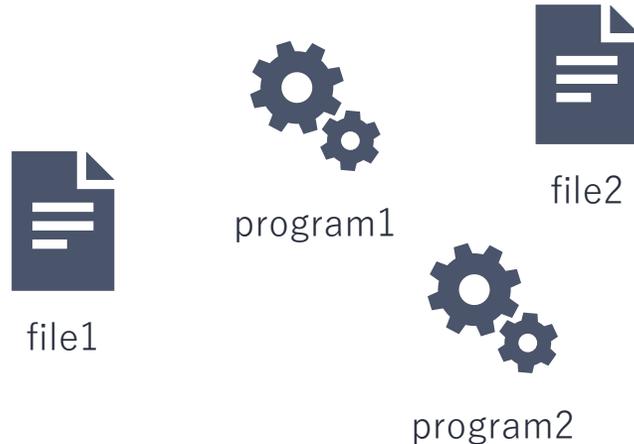
- 材料、手順

## ■ここからは研究のうち計算機実験について考える

- 計算機実験：計算機上でプログラムを実行してデータを処理する。シミュレーション、データ解析など
- 材料=入力データ、パラメータ、プログラム
- 手順=プログラムの実行方法

研究データそのものだけの管理ではたりない

材料=入力データ、パラメータ、プログラム



手順=プログラムの実行方法

```
progam1 -o file2 -param1 file1
```

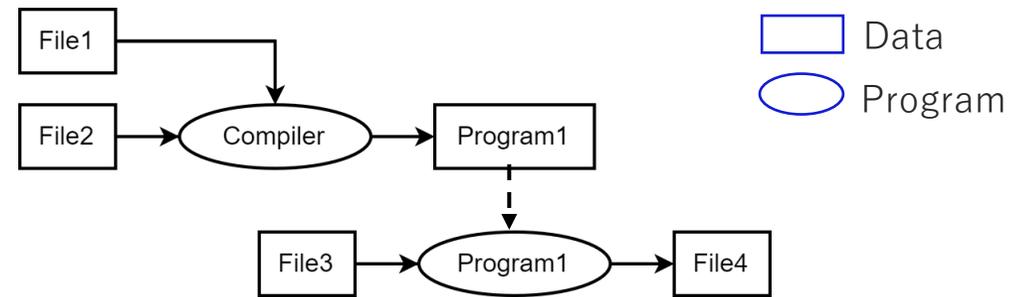
# 再現性と来歴

## ■来歴：データが生み出される過程（由来）を示す情報

- 「このデータはどのように作られたか？」

## ■役目

- 再現性の担保（研究公正の証明のひとつとして）：  
データを再生成するための情報を提供する
  - 査読者など第三者が手順を検証する
  - 第三者がデータの信頼性の判断に使用する
  - 研究者本人が意図しないデータの取り違えを防ぐ
- データ検索の手がかり（活用の促進）：  
あるデータからの関係性（入力、出力、…）でデータを検索する



Example of Data Provenance

# HPCシステムと来歴

- HPCシステムは学内外の幅広い分野の研究者に計算資源を提供している、研究の基盤である
- 計算資源だけでなく、より広い意味で研究を支えていく必要があるはず
- 計算機実験の再現性を高めるために自動的に来歴を記録、提供できないか？



# 問題と課題

## ■問題

- HPCシステムで生み出されるデータの来歴を、研究者の負担を小さくしながら記録する手法が確立していない
  - 手作業で記録するのは手間がかかりすぎる
  - 来歴を記録することができるワークフローシステムの導入は、既存のアプリケーションや実行環境に改修が必要
    - アプリケーションプログラムやその実行環境は脈々と受け継がれたもので改修は困難であること状況がありうる
    - その他にも特定の環境に依存していて改修が困難であったり、新たに依存することに抵抗がある場合がありうる

## ■課題

- 次の要件を満たす来歴記録手法の実現
  - HPCシステムでの実験を再現可能な情報を含む
  - 自動：利用者の手を煩わせない
  - 透過的：既存のアプリや実行環境の修正要求を最小にする
  - 低負荷：アプリの性能への影響を最小化する

# HPCの特性

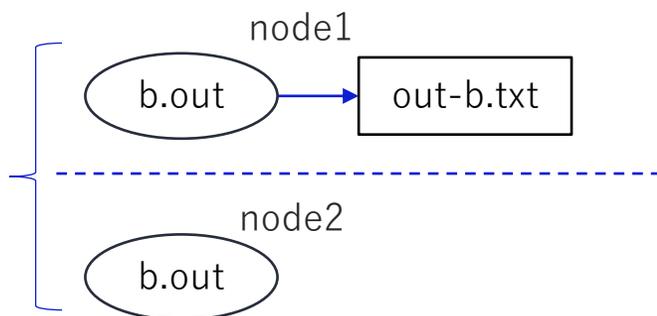
## ■普通の計算機とは使い方に違いがある：

- ①並列プログラム：複数の計算ノードで処理を分散して実行する
- ②ワークロードマネージャー（ジョブスケジューラ）の存在：バッチスクリプト（ジョブスクリプト）としてプログラムの実行手順を書き、ワークロードマネージャーに実行を依頼し、スケジュールされるのを待つ

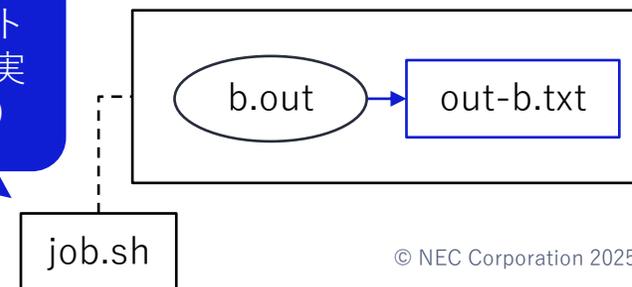
## ■これらの特性から来歴にも以下の工夫が必要

- ①複数のノードで動作する並列プログラムに対して、実行したのがひとつのプログラムであれば来歴でもひとつにまとめて表現する必要がある。
- ②プログラムの実行に対して、そのプログラムを実行したバッチスクリプトを紐づけて記録する必要がある

①複数のノードでひとつのプログラムを並列実行



②バッチスクリプト（プログラムを再実行するために必要）

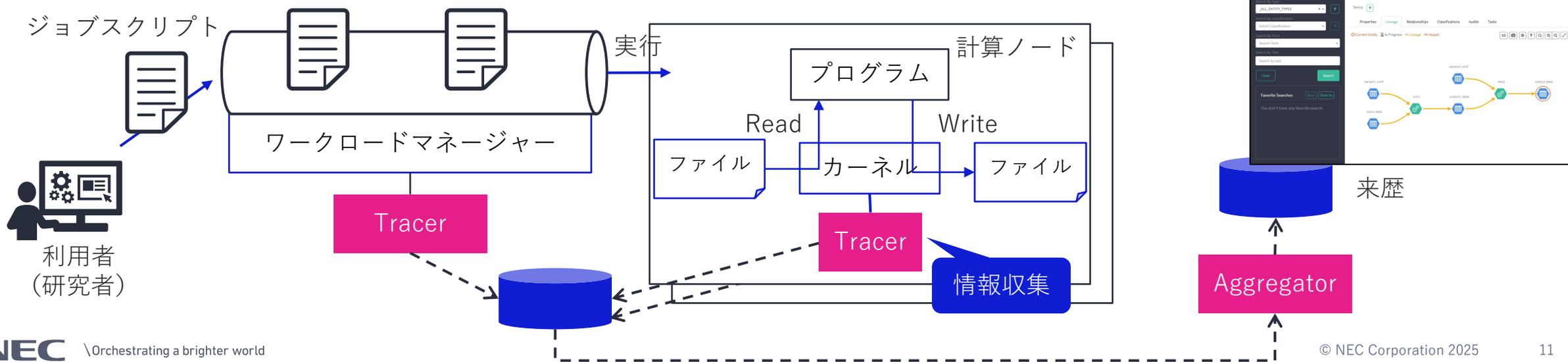


# 提案

- 実験データの再現に利用可能な来歴を自動的に、透過的に記録、構築するHPC向けフレームワーク
  - 自動 & 透過的 & 低負荷：Linuxカーネルのトレース機能（eBPF）を活用し、アプリ&システム環境の改変なく情報収集
  - 収集した情報にジョブなどHPC特有の情報なども組み合わせ、構成を工夫することで、データの再現に必要な来歴を構築

# 提案：システム構成

- クラスタを構成するノードのOSカーネルで、アプリのシステムコールの呼び出しを捕捉
  - どのアプリがどのファイルを読み書きしたか、ファイルのメタデータを採取
  - 実装にはeBPF（Linuxカーネルが提供するプラグイン機構）を使用
- ワークロードマネージャーからジョブに関する情報を採取
- これらの情報を統合、加工して来歴を構築（次ページ）



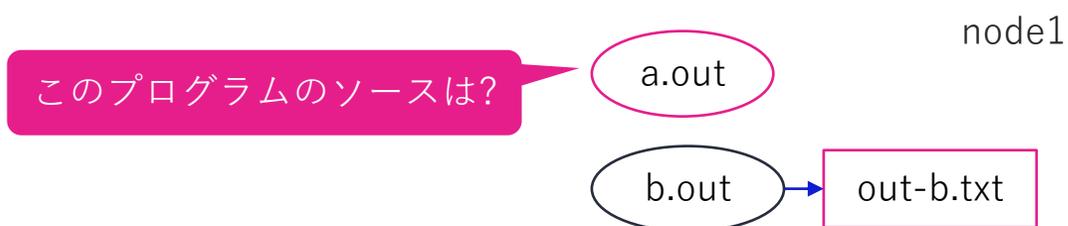
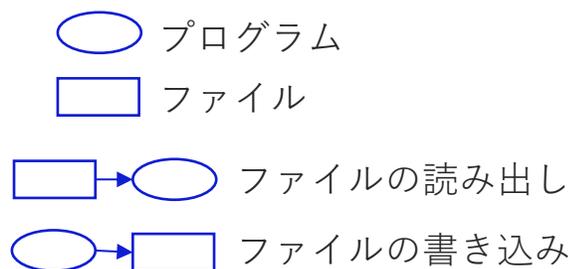
# 提案：来歴の構築

■OSレベルでの情報採取は（アプリの変更を要しないのはメリットであるが）低レベルであり、ユーザー目線では情報が不足

## ■再現に必要な情報

- ソースファイルとそこからビルドしたバイナリの実行の紐づけ
- ジョブとジョブスクリプトの紐づけ

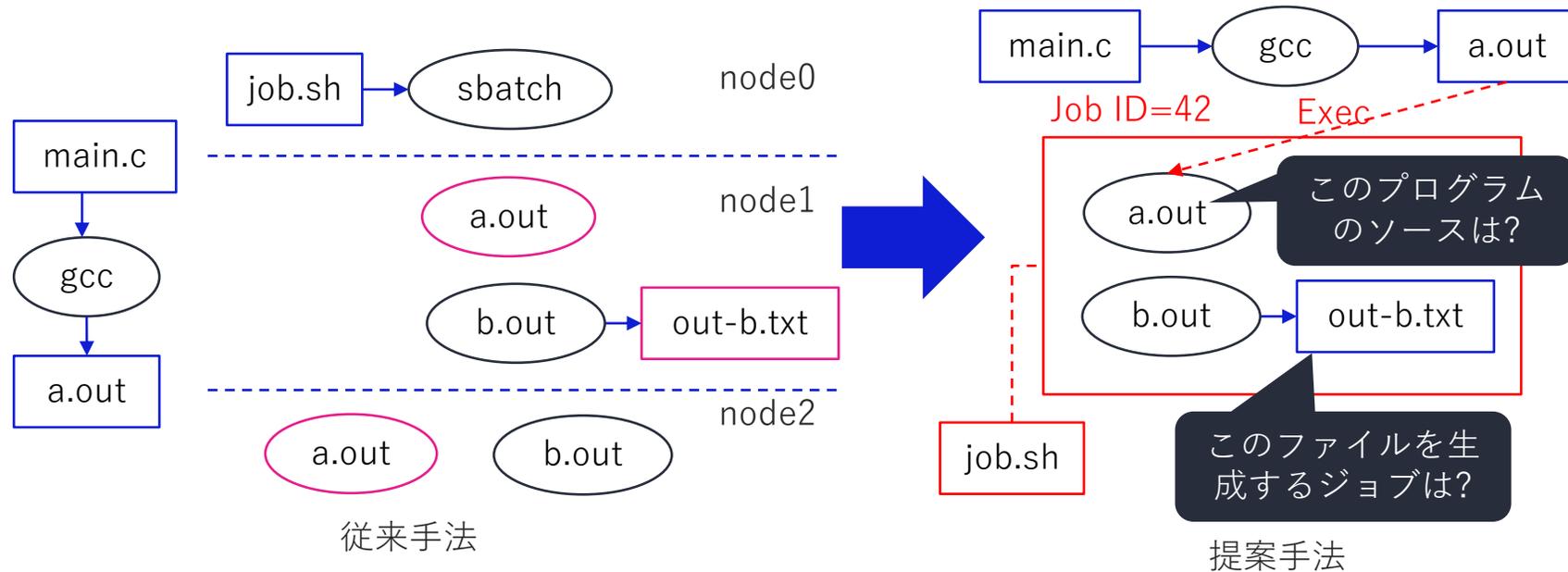
## ■情報を統合、加工するための手法を考案



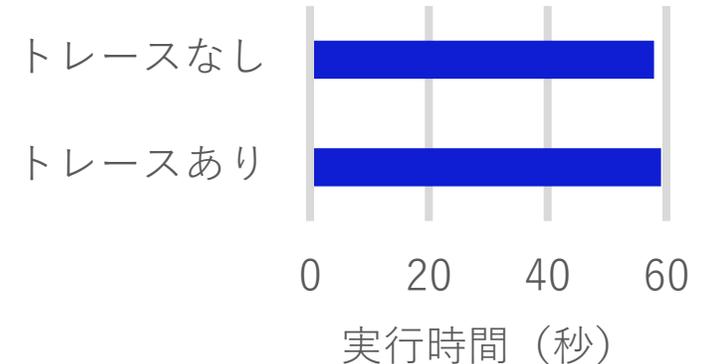
# 評価

## ■一般的なアプリ（NAS Parallel BenchmarkのBTIO）を用い、評価を実施

- 本手法で構築した来歴が、ファイルを再現するのに必要な情報を提供できていることを確認
- 情報採取をすることによるアプリケーションの性能への影響は+2.1%程度であることを確認



## アプリの実行時間への影響



# デモ (約1分)

- いつもどおりプログラムを実行すれば  
来歴が構築される



```
$ ./sim1 param1.conf ¥  
  < input.data > output1.data  
$ ./exp2 param2.conf ¥  
  < input1.data > output.data
```

The screenshot displays the Apache Atlas web interface. On the left, a search sidebar is visible with filters for 'Search By Type' (set to '\_ALL\_ENTITY\_TYPES'), 'Search By Classification', 'Search By Term', and 'Search By Text'. The main content area shows the details for the entity 'output.data (file)', including its technical properties (e.g., 'isIncomplete: false', 'name: output.data', 'qualifiedName', 'sha256', 'typeName: file') and user-defined properties (e.g., 'scale: 1000'). A lineage diagram is shown below, illustrating the flow of data from 'param1.conf' and 'input.data' through 'sim1' to 'output1.data', and from 'param2.conf' through 'exp2' to 'output.data'. A blue arrow points from the terminal command box to the search filters in the screenshot.

デモ動画：<https://onene.com/s/zqlcdt9hu7zp7jybmpwcu2zdg48rsq8g>

# 来歴の活用シーン

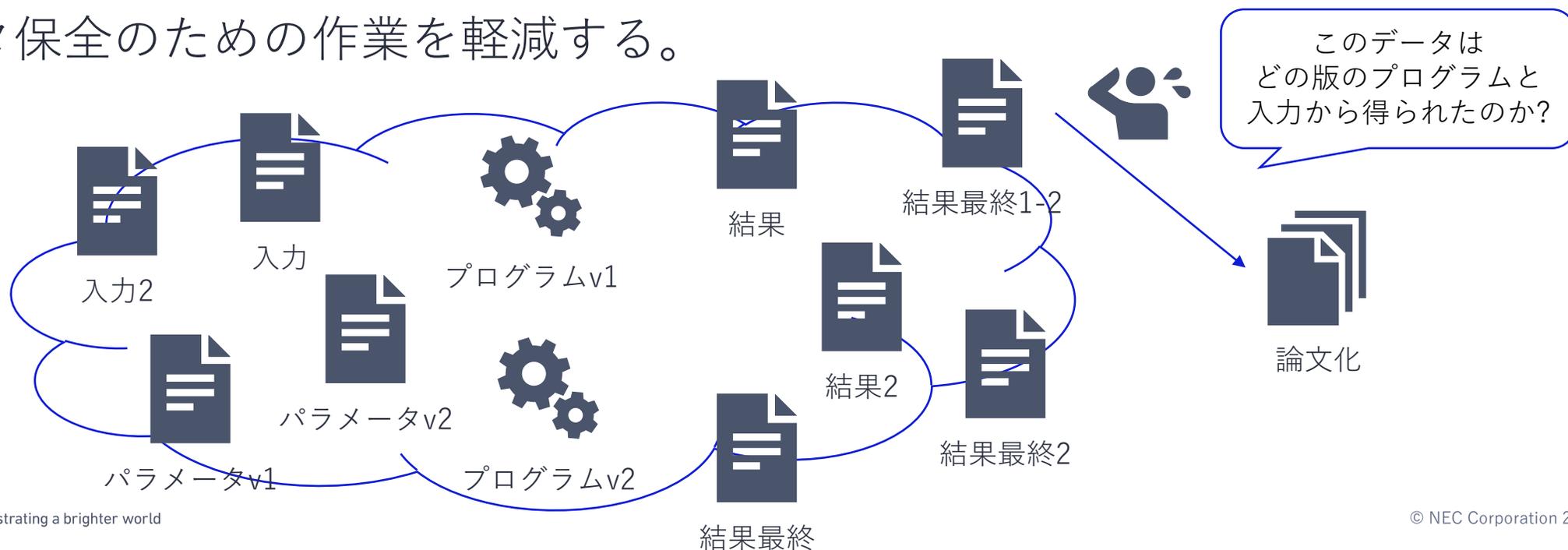
# 再現性への貢献

## ■再現性に関連するユースケース：

1. 保全すべきデータの特定
2. 実験の再実行に必要な情報の提供
3. 実験条件の確認

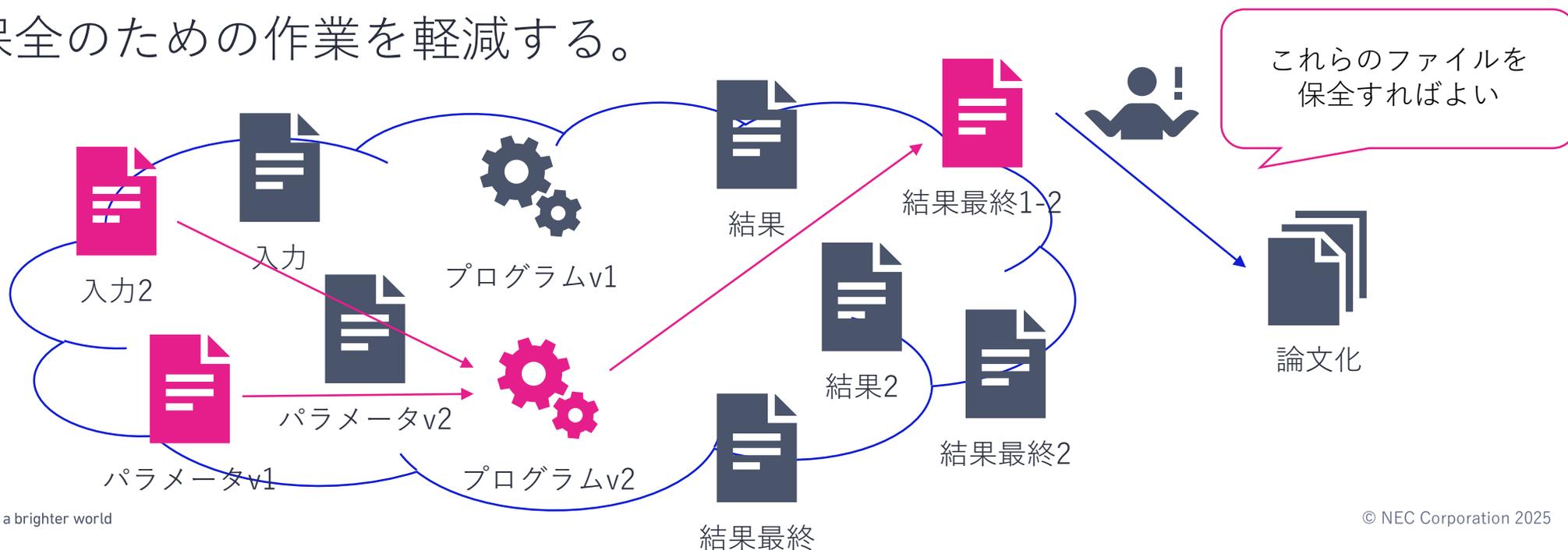
# 再現性への貢献1：保全すべきデータの特定

- 保全の対象とすべき研究データ（とプログラム）を特定する。
- 研究段階ではプログラムやパラメータを変更しながら実験の繰り返すことにより、同じようなファイルが大量に存在する状況が起こりうる。
- 来歴により、最終的に使用するファイルがどのプログラムや入力データなどに基づくものかが特定できる。
- データ保全のための作業を軽減する。



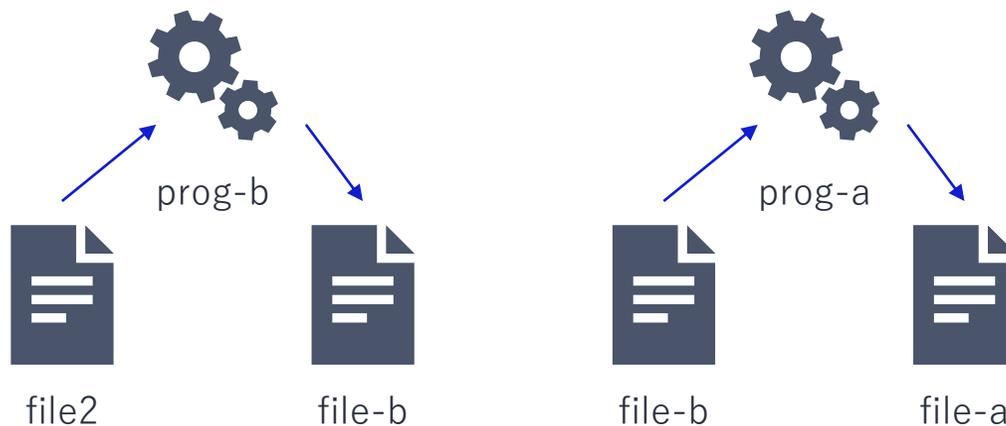
# 再現性への貢献1：保全すべきデータの特定

- 保全の対象とすべき研究データ（とプログラム）を特定する。
- 研究段階ではプログラムやパラメータを変更しながら実験の繰り返すことにより、同じようなファイルが大量に存在する状況が起こりうる。
- 来歴により、最終的に使用するファイルがどのプログラムや入力データなどに基づくものかが特定できる。
- データ保全のための作業を軽減する。



## 再現性への貢献2：実験の再実行に必要な情報の提供

- 実験を再現するための手順を明らかにする。
- 実験を再現するためには、（使用したファイルとプログラムに加え）プログラムをどのように実行するかの情報が必要。具体的には：
  - プログラムの実行順序（複数ある場合）
  - プログラムに与えたコマンドラインオプション
  - プログラムを実行するためのジョブスクリプト



どのように動かすと  
結果が得られる？

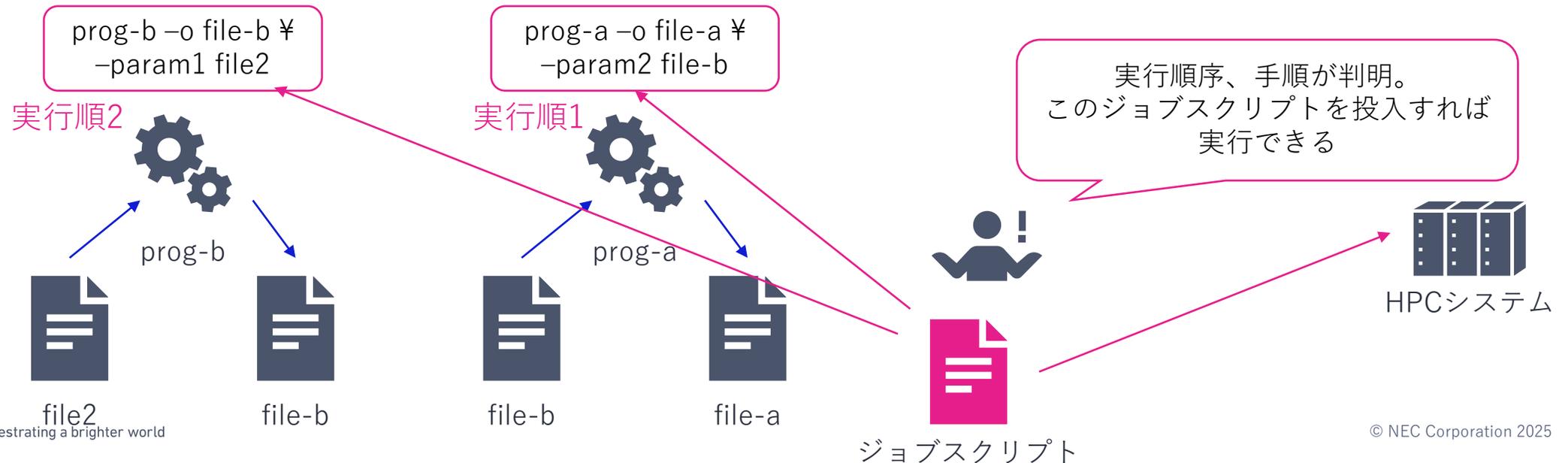
プログラムの実行には  
ジョブスクリプトが  
必要



HPCシステム

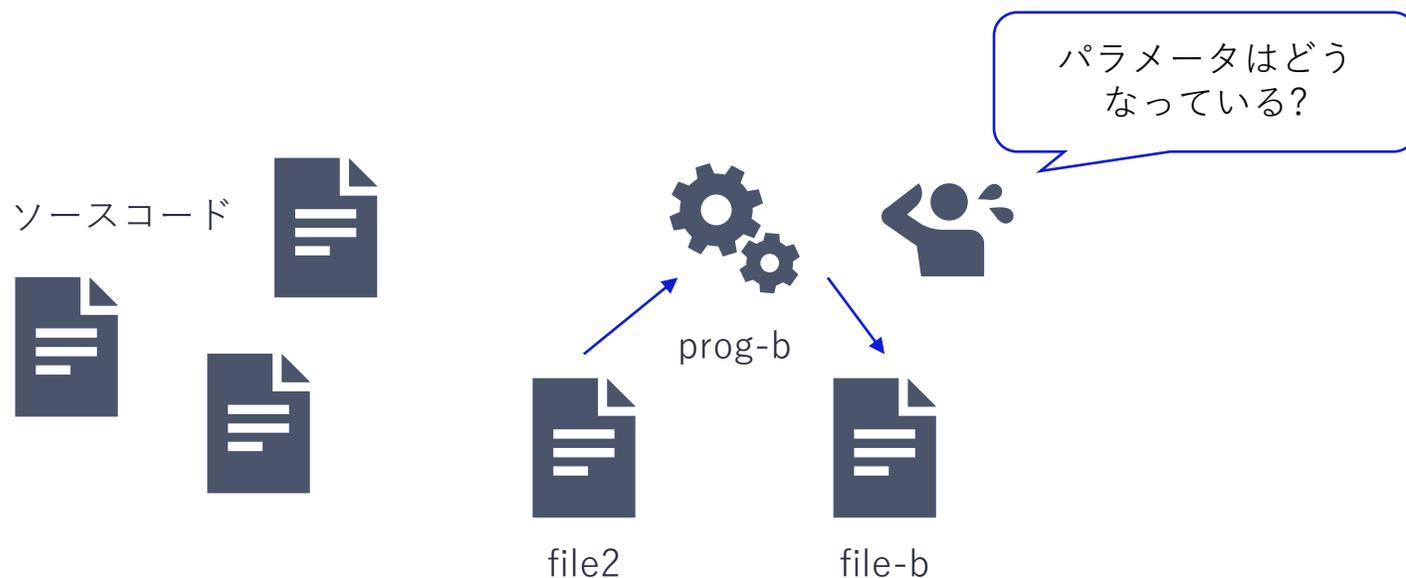
# 再現性への貢献2：実験の再実行に必要な情報の提供

- 実験を再現するための手順を明らかにする。
- 実験を再現するためには、（使用したファイルとプログラムに加え）プログラムをどのように実行するかの情報が必要。具体的には：
  - プログラムの実行順序（複数ある場合）
  - プログラムに与えたコマンドラインオプション
  - プログラムを実行するためのジョブスクリプト（HPCシステム）



## 再現性への貢献3：実験条件の確認

- 実験に使われたパラメータなどの実験条件を特定する。
- (特に自作プログラムでは) パラメータなど実験条件がソースコードにハードコードされていることがある。
- 結果を出力したプログラムのソースコードの版を特定する必要がある。



## 再現性への貢献3：実験条件の確認

- 実験に使われたパラメータなどの実験条件を特定する。
- (特に自作プログラムでは) パラメータなど実験条件がソースコードにハードコードされていることがある。
- 結果を出力したプログラムのソースコードの版を特定する必要がある。

param.h

```
float param1 = 45.6;  
...
```



prog-b



file2



file-b

パラメータを記述している  
ソースコードを特定

## その他のユースケース

■来歴は研究データの活用の促進し、研究の効率性、技術発展に貢献できる。

■データ活用の効果：

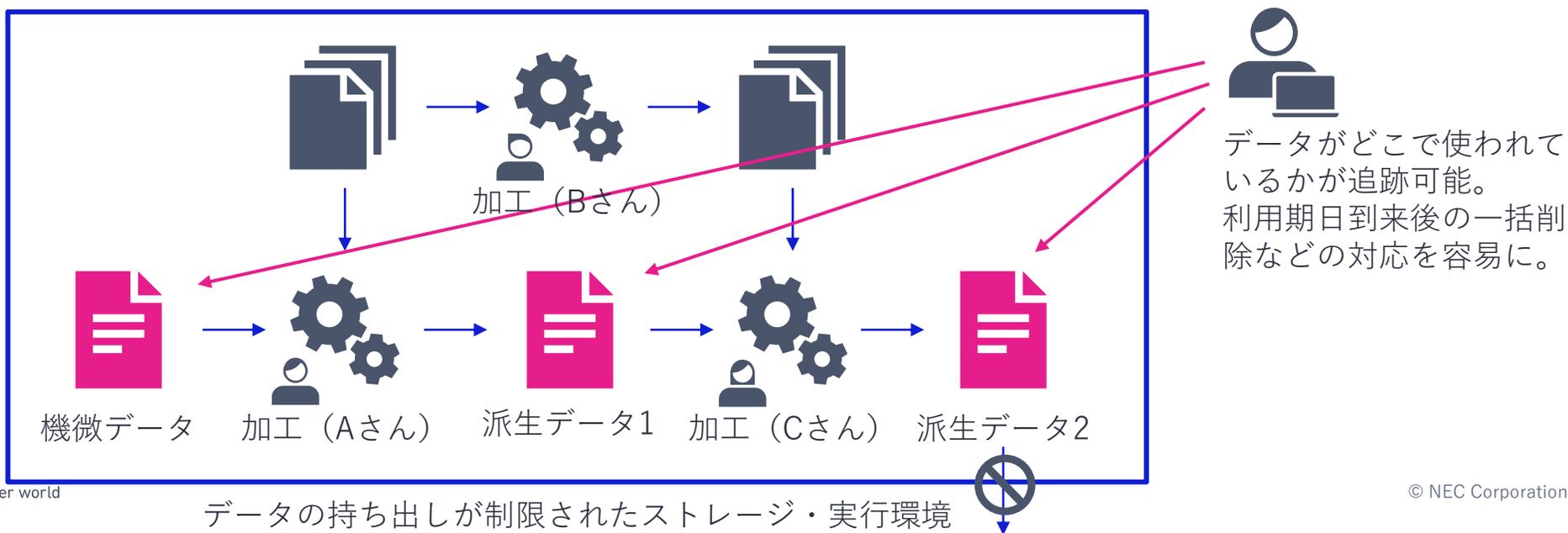
- 同じ（ような）データを研究者ごとに収集する必要がなくなる。
- あるデータを複数の研究者が多角的に分析することが可能になる。

■ユースケース：

1. データの利用状況の追跡
2. データを基点とした研究の体系化

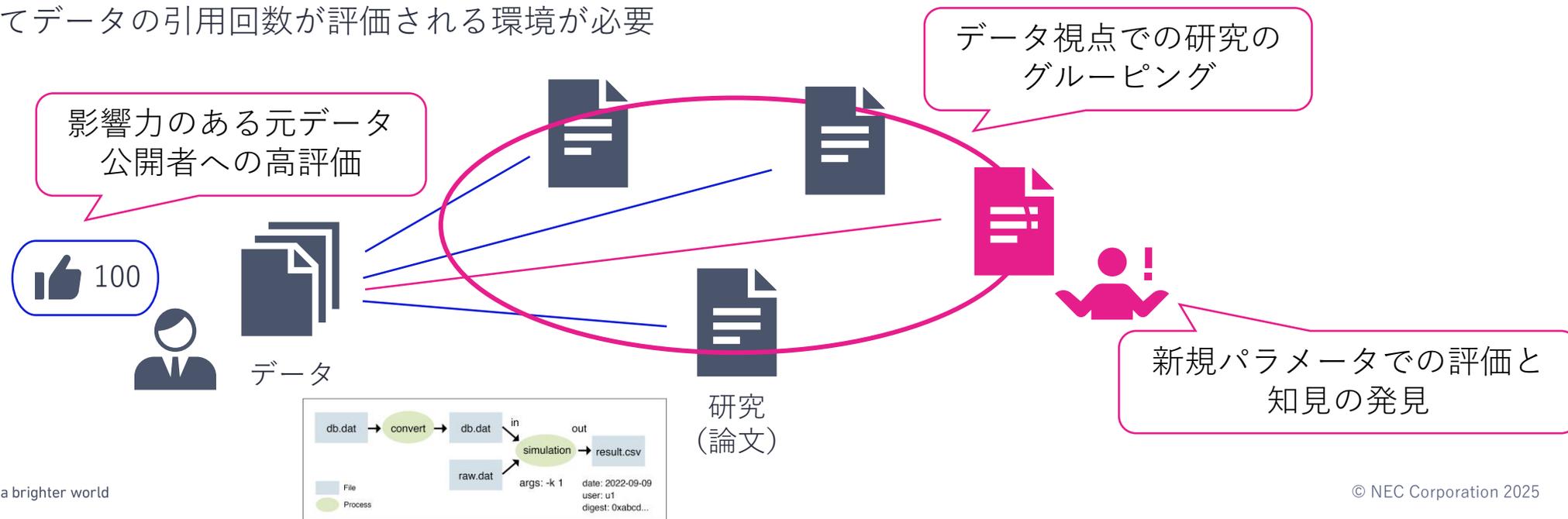
## その他のユースケース1：データの利用状況の追跡

- データとその派生データがどこで利用されているかを確認する。
- 機微なデータも安心して共有できる環境を実現し、データの活用を促進する。
- 具体例：
  - 期限到来後削除が必要など、後日派生データを含め対応が必要な個人情報など機微なデータの利用状況の追跡をする  
(ただし、追跡が行き届くストレージ、実行環境からデータが持ち出せないようになっていることが前提)。



# その他のユースケース2：データを基点とした研究の体系化

- データの利用状況をもとに先行研究を整理することを可能にする。
- 新たな視点からのデータの活用、研究を促進する。
- 具体例：
  - 同じデータを活用している研究をまとめる
  - 同じデータ、プログラムでパラメータだけ変えて実験を行う
  - 論文だけでなくデータがどれだけ使われているかを可視化し、データ公開のインセンティブにする
    - 前提としてデータの引用回数が評価される環境が必要



# 今後の課題

## ■再現環境のパッケージング、再実行

- 再現に必要な情報を記録するだけでなく、配布/保全用にワンクリックでパッケージング、さらに再実行

## ■外部機器との連携

- 研究はHPCシステムの中だけで行われているのではない
- 各種計測機器や計算機システムを跨いで、データの発生から追跡できるべき

## ■データ公開用リポジトリとの連携

- 学内、研究分野ごとに整備されたリポジトリのデータの引用、リポジトリへの公開を容易にできるように連携

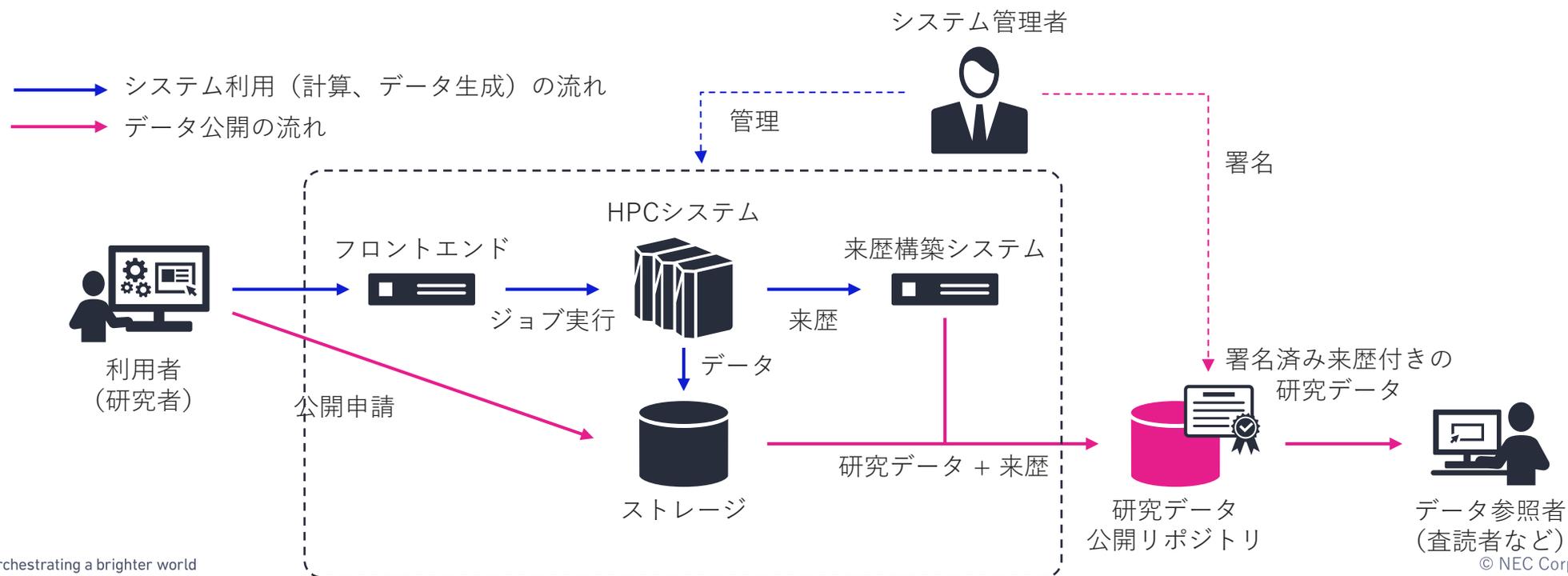
## ■来歴の内容を証明

- 電子署名などを用い、来歴の内容を計算機システム管理者が証明。
- あるデータが特定の計算機システムにより記載された通りの手順で生成されたものであることを明らかにする

## 今後の課題 (2)

■HPCシステムを普通に使うだけで、来歴（とその他のメタデータも）が自動的に記録される。ワンクリックで研究データと共に公開リポジトリに連携

- メタデータ：作成日、作成者などのデータに対するデータ。写真であれば絞り、シャッタースピード、撮影場所なども
- 来歴はシステム管理者が内容を保証→データとそれに紐づく研究の信頼性の向上
- データを探しやすく、活用されやすくする→オープンサイエンスを推進



# まとめ

# まとめ

- 研究活動における再現性への要求の高まりに対し、計算機システムによって生成されるデータを対象に、来歴を自動的に記録する仕組みを紹介
- 再現への貢献：
  - 保全すべきデータの特定
  - 実験の再実行に必要な情報の提供
  - 実験条件の確認
- 再現以外に研究の効率性、技術発展に貢献する2つのユースケースを紹介：
  - データの利用状況の追跡
  - データを基点とした研究の体系化

**NEC**

\Orchestrating a brighter world