Contents

Provenance briefing – September 2024

This document sets out the current thinking on Provenance for a non-technical audience. It is subject to change as IB1 engages with stakeholders.

In the context of a Trust Framework, Provenance is a record of where data came from, the members who collected and processed it, and what processing was performed.

The Provenance record may contain additional information relevant to each step of origination and processing. A TF or Scheme would typically add Assurance information, such as data quality and trust indicators.

Decentralised

Provenance must be decentralised, with records passed directly between data providers and data consumers. Each participant involved in the data origination and processing adds *steps* to the record which describes their activity.

Records are signed using certificates issued by the Directory. After steps have been added by a participant, they sign the entire record, including the signatures of previous participants. This forms an unbreakable chain of signatures, where attempting to alter previous steps would break the final signature.

The choice of a decentralised architecture has been made because it is consistent with the overall architecture of Trust Frameworks, avoids the overhead of additional API calls to transfer data, and avoids relying on another party to provide a service with 100% uptime.

However, it does require compromises to be made with how the record is signed, and prevents an audit from being able to prove it has seen all transactions between the members.

Example

This conceptual example of a Provenance record shows the steps in receiving data from an external organisation to bring it into the Trust Framework, processing by an Application, and transfer to another member who relies on this data for decision making.

Step 1: Receipt
Member a82f23 confirmed receipt of *consumption data* from external organisation *Smart DCC*. *External signature* embedded.
Assurance: *Certified data from meter*.

Step 2: Transfer
Member a82f23 transferred *consumption data* from *Step 1* to *Member b11e21*.

Signature: by *Member a82f23*
Signs *Steps 1 & 2*.

Step 3: Receipt
Member b11e21 confirmed receipt of *consumption data* from *Step 2* from *Member a82f23*.

Step 4: Processing
Member b11e21 processed data from *Step 2* using *Application fa234b*.

Step 5: Transfer
Member b11e21 transferred *emissions report data* from Step 4 to *Member e123df*.

Signature: by *Member b11e21*
Signs *Steps 3, 4 & 5*, and inclusion of *Step 1 & 2 + signatures*.

Step 6: Receipt
Member e123df confirmed receipt of *emissions report data* from *Step 5* from *Member b11e21*.

Signature: by *Member e123df*
Signs *Step 6*, and inclusion of *Step 1, 2, 3, 4, & 5 + signatures.*

No suitable standards

There are standards for provenance data (see annex), but none that are suitable for TF use. Either they are too general, and cannot capture TF data without being extended so much they would appear as a custom format, or they are too specific and include complexity and concepts that have no application.

There does not seem to be widespread adoption of provenance metadata, leading to low usage and software support, or complete abandonment of standards. IB1 would not benefit from existing software support, community, or implementation experience.

Non-repudiation: Transfer and Receipts

After a participant creates or adds information to a Provenance record, they must not be able to deny that they did it, nor can all or part of a record be faked by another party.

When data is transferred there is a two step process. Firstly, a *transfer* action is added to the Provenance record and sent with the data. Then, the recipient must then add a *receipt* to confirm they received it and it meets the description in the transfer action. Once the two are in place, neither party can deny the transaction took place.

However, records must be visible to other parties to be non-repudiable. If the receipt is only held by the consumer, it can be "lost" and the transaction denied. If a Scheme requires that data providers must have confirmation the consumer received the data, then the receipts must be sent back, incurring the overhead of another API call.

If both sides have the receipt, it would require collusion by both to deny a specific transfer took place.

Support for data outside the Scheme

Not all data will be received from TF participants. A Provenance record will allow receipts of data from outside the TF, signed by the participant that brings the data into the TF to assert the data's origin. These receipts will include any external signatures, for example, an Energy Data Provider which brought smart meter data into the TF would include the DCC signatures for smart meter data.

Audits and logging

Because of the choice to decentralise:

- A data provider **cannot** trace what happens to data after it is transferred to another participant.

- A data processor in the middle of the chain **can** use the Provenance record to see where the data came from and all steps along the way to them, but they **cannot** see what happens next in the chain.

All participants are trusted to follow the licences and pre-executed agreements, so enabling onward tracing of data is not worth the significant costs of architectural change or communication overheads.

Provenance records will be logged by participants so that audits can trace the data through the participants. Logs will need to include sufficient data to be able to trace data over multiple transactions throughout the entire value chain.

All audits require cooperation from all parties, because there is no central database. If multiple parties are willing to collude, they can hide data and actions from the audits.

No proof of processing

Provenance records only provide verifiable proof that a participant created their part of the Provenance record. They do not provide verifiable proof that the processing asserted actually took place. There is an implicit reliance on auditing and contractual controls being sufficient.

Schemes may layer their own solutions for proof-of-processing within the assurance payload, but there is no sufficiently lightweight general solution that can be used across all Trust Frameworks.

Assurance

There is value in having assurance data readily available for all transactions, as it is an important input into data-driven decision making. Assurance data naturally piggybacks onto the Provenance record, taking advantage of transport alongside data, and signatures to verify authenticity.

However, the Provenance specification will only define where the Assurance data is included in a record. The definition and encoding of assurance information will be defined by individual Schemes or TFs.

Minimum contents of record

While a record may contain additional data, the minimum information is:

- Participants and Applications (as Directory URLs)
- Dataset / Data Services used (as Directory URLs)
- Calculations performed (as Registry URLs)

- URL of endpoints
- Timestamps
- Licence used (implied by Data Service)
- FAPI Transaction ID
- Permission identifier (if transferred with end user permission)

Since multiple parties must collude to fake data, it is not necessary to use an external timestamp authority, where an impartial third party provides signed timestamps within the signatures.

Compromises for signatures to be practical

For a record to be completely self-contained, it would need to include all signing certificates, which will be around 10 times larger than the data. Instead, the Directory is likely to be required to provide an API to fetch certificates for signature validation.

The record structure will be more complex than most TF data structures, won't be directly human readable, and requires library code to decode and verify. If IB1 were to provide this library, versions would need to be written for most common platforms and there would be an ongoing maintenance burden.

Decentralisation means that there cannot be ongoing oversight from a central service trusted to ensure the validity of Provenance records. If participants collude, they can apply valid signatures to fake records that rewrite the past or claim that some activity took place. Detection will be extremely difficult.

Flexibility for Schemes

Schemes are free to use Provenance in any way they need, including:

- Not using provenance at all
- Using a completely different provenance record
- Receipt handling
- Adding a central service to record all transactions for auditing purposes
- Completely centralised provenance services
- Additional metadata in Provenance records

Out of scope for this document

- When Provenance records are required
- What Assurance information is collected and how it is represented
- Logging requirements

Annex

Standards

PROV *(release and activity in 2012 and 2013)*
An open standard for general purpose usage, describing a provenance data model that can be used in a decentralised system. It would need extending to cover TF use cases, and would end up being so extended, we might as well use something custom which fits our use case exactly. Everyone who uses/used PROV extends it. There's a mapping from Dublin Core (DC) to PROV, but although we use DC for some of the information to describe data sources, only a small number of fields would map through.

Open Provenance *(work started 2006, latest release 2010, last commit on library in 2011)*
Similar to PROV, but more comprehensive, although still not aligned to TF use case. Expresses a graph of derivation.

CoreTrustSeal *(work started 2008, last new certified repository in Jan 2023)*
A set of requirements and a review process for academic repositories to show they can be trusted to have acceptable content and preserve it adequately. Only relevant from a policy / governance point of view.

C2PA *(work started 2019, specification releases in 2021 and 2023)*
Relevant as it describes a series of modifications by independent organisations, but it is very specific to media editing, and is encoded with a standard that is not typically used in the ecosystems targeted by Trust Frameworks.

OpenLineage *(2021, active development since)*
An API specification for a centralised service to track dataset origination and processing jobs. Simple data model, with expectation of extensions for specific use cases.