Verifiable Credentials and end-to-end traceability

Draft White Paper, v0.4, 1st April 2022

1 Introduction

Around the world and in multiple industry sectors (especially food and pharmaceutical products), there are increasing legislative and industry initiatives to increase the safety and security of supply chains, to reduce fraud and to ensure that the products sold or dispensed to consumers are safe, authentic and unadulterated. Some of these approaches rely on being able to check the provenance or traceability history of individual product instances from their current location or owner upstream through the supply chain to the original manufacturer, checking that there are no gaps or inconsistencies in the traceability data. Gaps or inconsistencies could indicate the introduction of counterfeit products or other suspicious behaviour.

The GS1 EPCIS standard defines an open standard data model for exchange of visibility event data, together with standardised interfaces for capturing / storage of such data and for retrieving it via a standardised query interface and query language optimised for such event data. A companion GS1 standard, CBV, defines a number of standardised code lists and code values that can be used to populate such EPCIS event data, to refer to a specific business step or the state or disposition of a product, asset or logistic unit.

Although the use of such open standards for exchange of traceability data increases the consistency and efficiency of such data exchanges for all stakeholders, there is still a reluctance to share such traceability data openly, since large volumes of such data could be mined by a competitor, to extract or infer commercially sensitive business intelligence about trading relationships, inventory levels, production volumes/rates, etc. For this reason, companies exercise great caution about sharing fine-grained traceability data about individual product instances or the logistic units (cases, pallets, vehicles) that transport them through the supply chain. They would typically only want to share information on a 'need to know' / 'right to know' basis.

A further complicating factor is the emergent nature of the supply chain path taken by a product instance of a mass-produced product. A manufacturer does not typically manufacture each specific product instance for a specific intended retail store or pharmacy. Instead, products are manufactured in bulk and distributed according to demand further downstream, as orders are placed. Therefore at the beginning of its journey through a supply chain network, the manufacturer typically has no knowledge about which organisations further downstream (wholesalers, distributors, retailers, pharmacies) will handle a specific product. This emergent nature of the supply chain path is illustrated in Figure 1.

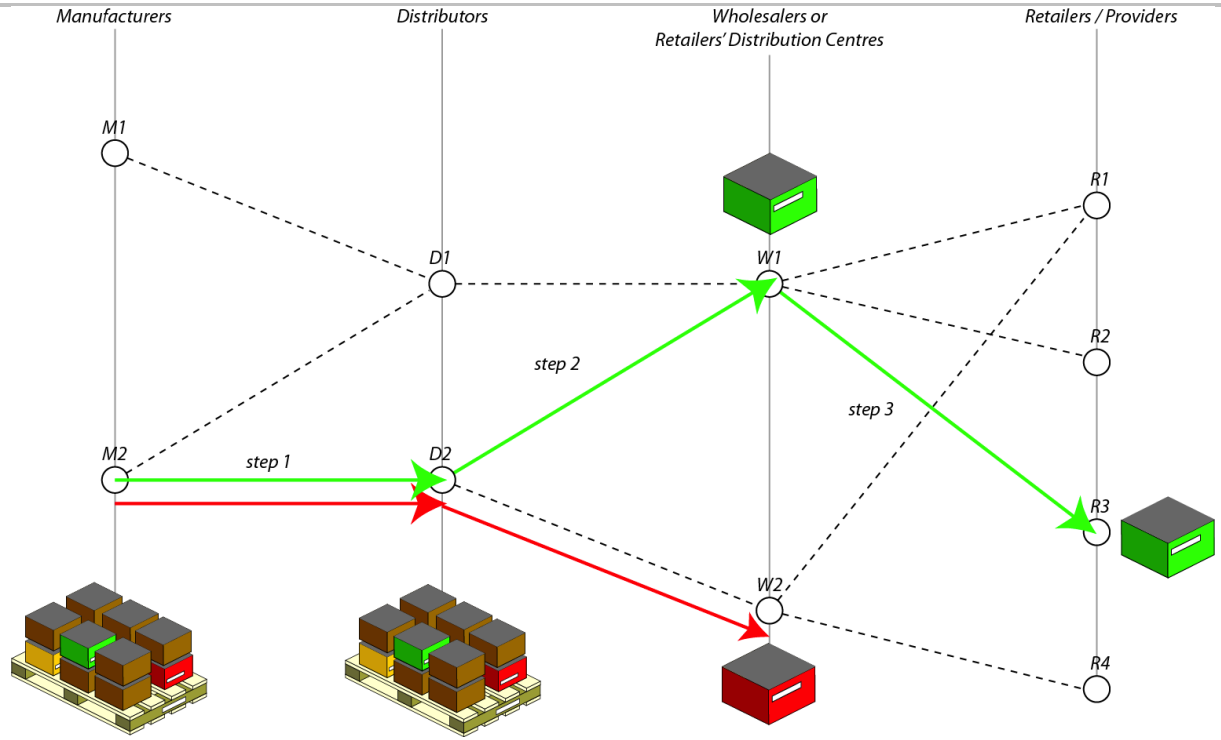

Figure 1 Due to the emergent nature of supply chain paths over time, two objects (in this example the cases shown in green and red) that were initially aggregated to the same pallet may ultimately end up at unrelated locations/organisations and the manufacturer usually does not pre-define these and typically has almost no downstream visibility about where each of its products arrives.

Nevertheless, such downstream parties may be expected to have checked upstream traceability data for the products they sell, dispense or distribute. This is clearly challenging when they do not have a direct trading relationship with the manufacturer, since it is currently difficult for such downstream parties to prove that they are connected (on the same actual chain of custody or ownership of a product), that they have possession of ownership of a specific product instance and a right or need to request such traceability data from upstream parties. Figure 2 illustrates this 'bootstrap' challenge of establishing trust or 'right to know' between organisations that are not direct trading partners.

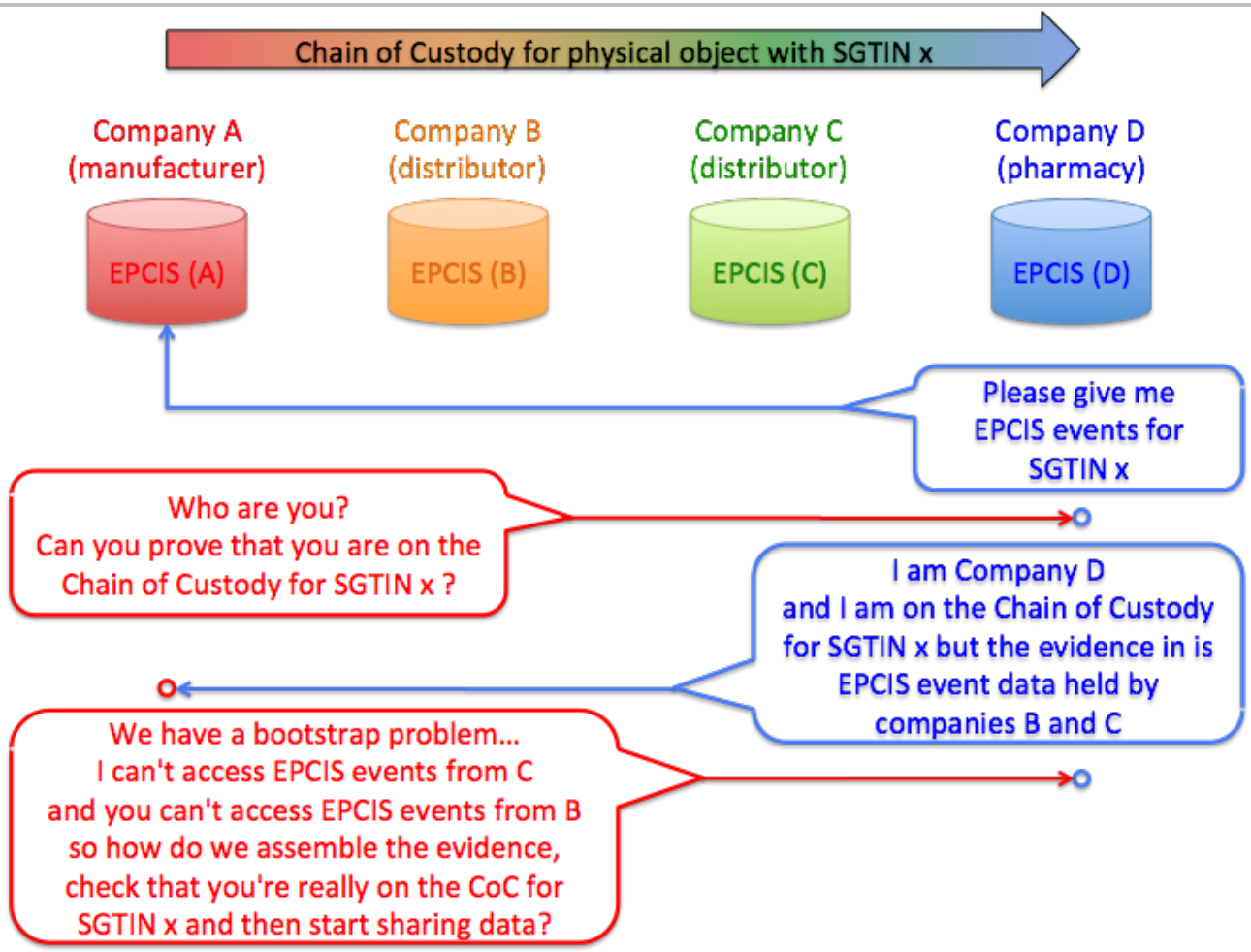

Figure 2 Graphical illustration of the 'bootstrap' challenge of establishing trust / proof of connectedness and willingness to exchange data between organisations that are not direct trading partners

This paper describes an approach in which open standard EPCIS event data can be transformed into a sanitised format that could be used within a set of one or more Verifiable Credentials as evidence that collectively prove the chain of custody or chain of ownership in a way that avoids revealing the true identity of the intermediate stakeholders or the true identity of the product instances being transported across the supply chain network. This 'evidence of connectedness for a specific product instance' could then be used to enable data sharing between organisations that do not have direct trading relationships and may be useful for enabling the sharing and checking of traceability data, by providing a technical solution to the aforementioned 'bootstrap' problem of establishing trust.

Recent GS1 standardisation activities to modernise the EPCIS and CBV standards introduce a JSON / JSON-LD data format and OpenAPI / REST Web interfaces. These are expected to be beneficial, especially as a JSON-LD data format is already envisaged within the W3C Verifiable Credentials data model.

2 Technical components

The main building blocks of this approach include:

- Open standard GS1 EPCIS event data
- Open standard Verifiable Credentials, based on the W3C Verifiable Credentials data model.
- Decentralised Identifiers (DIDs), specifically peer DIDs that are created in a pairwise ad-hoc manner between trading partners at the time of a transfer of goods between them
- An automated 'chain navigation' algorithm that can interpret event data (also in its sanitised form) to determine subsequent queries needed to collect the complete relevant set of visibility event data end-to-end, if available.

2.1 GS1 EPCIS event data

The GS1 EPCIS standard defines an open standard data model for expressing visibility event data for traceability. It was first published in 2007 and is currently undergoing a major modernisation effort, with a major version update expected for publication in 2022. The current version already defines four event types:

- `ObjectEvent` is typically used to represent observations of things at locations at a particular timestamp for a particular reason or business process step, as well as the initial commissioning (creation) of objects (product instances) and their final decommissioning, where appropriate.
- `AggregationEvent` is typically used to represent packing or unpacking processes in which 'child' objects become aggregated or disaggregated to/from a parent object, such as a container, for more convenient transportation through a supply chain network.
- `TransactionEvent` is typically used to link or unlink objects to/from a specific business transaction, although the other event types also support cross-referencing to related business transaction identifiers and types (e.g. purchase orders, invoices, bills of lading etc.)
- `TransformationEvent` is typically used to represent the irreversible transformation of one or more input objects into one or more output objects, such as the transformation of a number of food ingredients into processed food products.

EPCIS 2.0 will add support for an additional event type (`AssociationEvent`), which is similar to an `AggregationEvent` but has subtly different semantics and is more suitable for representing permanent / semi-permanent attachments, e.g. of sensor devices to returnable plastic crates used to transport fruit or vegetables, such that the sensor device remains associated even after the contents of the crate have been emptied, unpacked or disaggregated. EPCIS 2.0 will also add support for expressing business-relevant sensor data.

2.2 Verifiable Credentials

Verifiable Credentials are an open standard for secure digital credential information, intended as a more secure and more machine-interpretable equivalent to documents that are currently paper-based, such as driving licences or passports. Verifiable Credentials essentially enable an Issuer to write some factual claims about a Credential Subject (which may be any entity such as a product, organisation or even a person) using a structured data format that is not only machine-readable but ideally also machine-interpretable Linked Data and where this data is digitally signed so that the digital signature can be independently verified by a Verifier and any tampering of the data can be detected. The World Wide Web Consortium has published an open standard data model for Verifiable Credentials.

2.3 Decentralised Identifiers

Decentralised Identifiers (DIDs) enable the identification of a thing (product, person, organisation, asset etc.) in a way that does not rely on a centralised authority to generate or issue the identifiers. The Decentralised Identifiers used in the approach described in this white paper are cryptographically generated, typically using asymmetric public key cryptography in such a way that the decentralised identifier is derived from the public key using a well-defined algorithm, while the corresponding private key is kept private. The owner of a decentralised identifier is self-sovereign, in the sense that they (or rather, their public/private key pair) is the Issuer of the decentralised identifier and they can prove ownership or control of the decentralised identifier by using their secret private key to construct a digital signature for a random challenge, that signature being independently verifiable by anyone using the public key that corresponds to the decentralised identifier. Decentralised identifiers are of particular interest to privacy advocates since they need not reveal the true identity or name (or other characteristics such as gender) of the holder or Credential Subject and because each holder or Credential Subject can own and control multiple decentralised identifiers, each used for specific purposes, effectively having multiple aliases that are incognito until the holder or Credential Subject decides to prove ownership to another entity, acting as the Verifier . The use of Decentralised Identifiers together with Verifiable Credentials enables the holder or Credential Subject to control exactly how much information is shared and with whom and to minimise opportunities for organisations to collect unnecessary information that is not actually required for the decision-making process or to correlate datasets about the Credential Subject without their explicit consent or control. Their anonymous nature can also be helpful to ensure that decisions are made based objectively on factual claims and in a non-discriminatory and non-prejudicial way. Further use cases and requirements for Decentralised Identifiers are discussed in a W3C technical recommendation – see References.

2.4 Automated 'chain navigation' algorithm

As each object moves through a supply chain, it may pass through multiple stakeholder organisations, undergo multiple packing/unpacking processes as well as multiple shipping/receiving processes. In order to interpret the data and to collect the relevant data at each step in the process, simple logical rules can be defined for a chain navigation algorithm that can analyse each event and determine which events to request next – or even determine whether it needs to contact a different organisation further upstream or downstream to request further event data about the same identifier or a related identifier (such as the identifier of a parent container or the contents that have been unpacked from the container).

3 Graphical worked example

Figures 3-5 depict a set of EPCIS event data collected by a manufacturer, distributor and retailer. They also indicate graphically how each open standard EPCIS event data could be transformed into a sanitised format that is non-revealing about the identities of the physical objects (product instances, logistic units) or the stakeholder organisations involved in its supply chain path.

Figure 6 shows how a Verifiable Credential of set of Verifiable Credentials could use such sanitised event data to provide a proof of connectedness between organisations for specific physical objects such as product instances or logistic units.

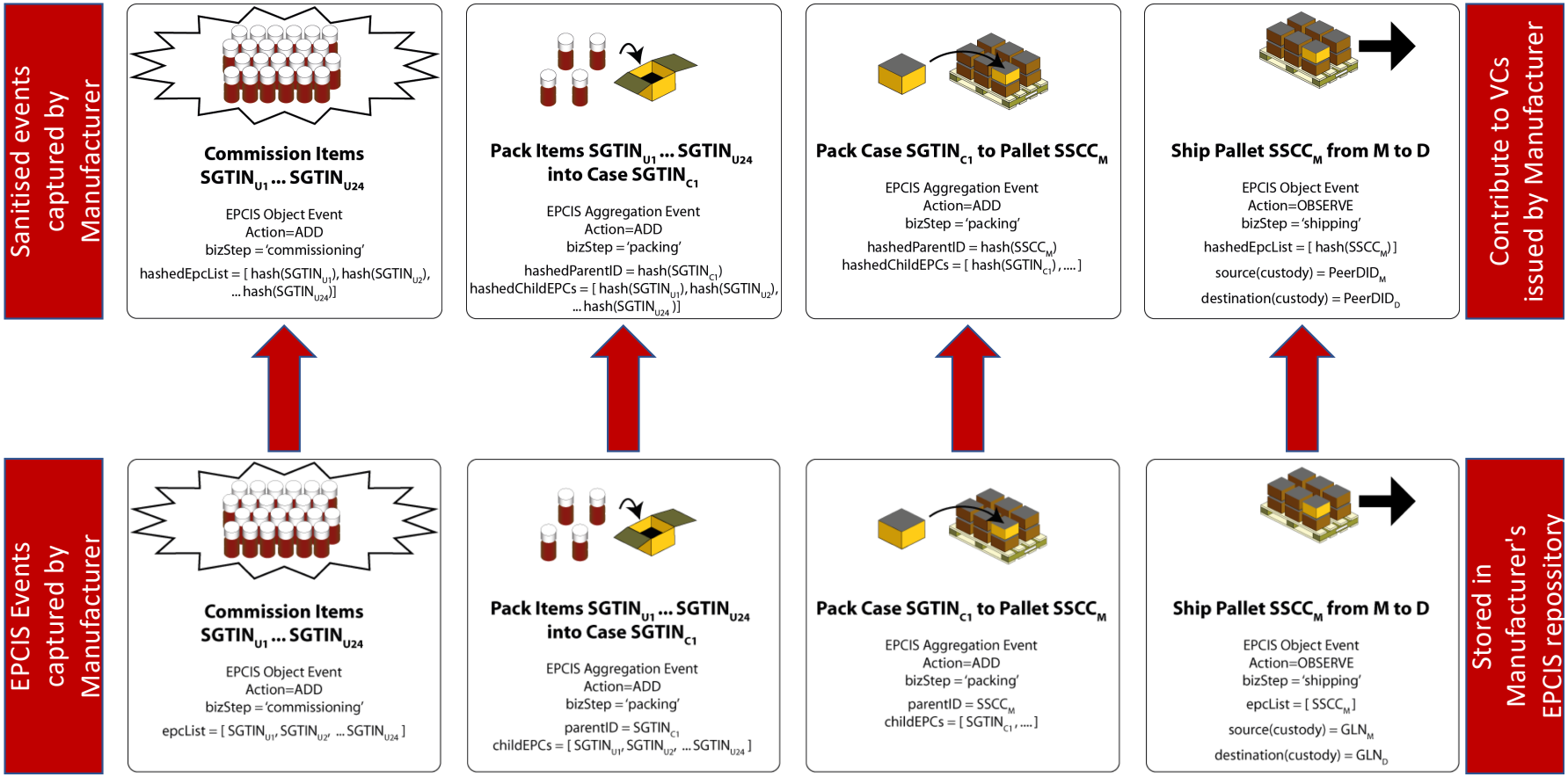

Figure 3 – an example of EPCIS events captured by a manufacturer and their transformation into a sanitised format

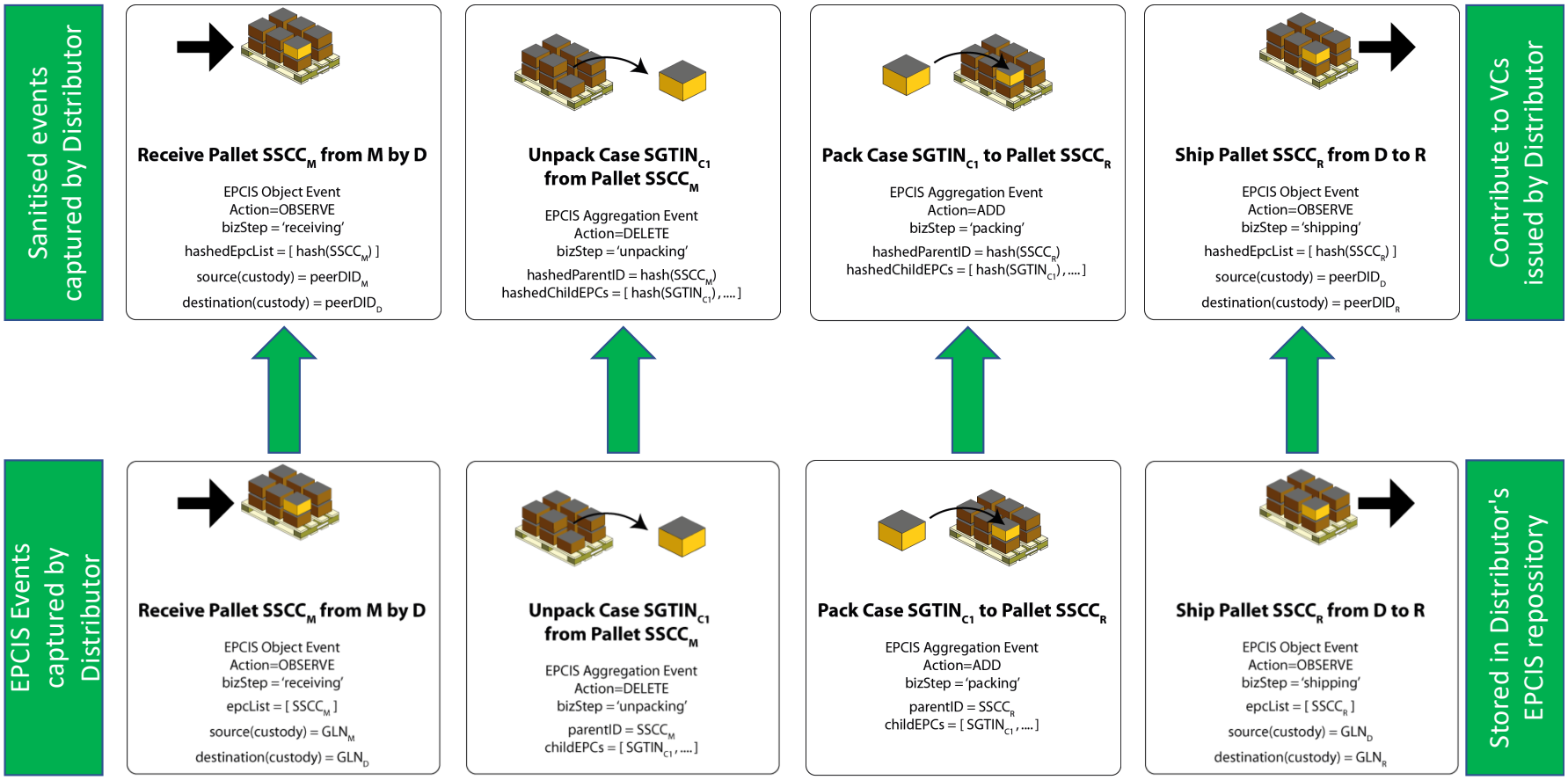

Figure 4 – an example of EPCIS events captured by a distributor and their transformation into a sanitised format

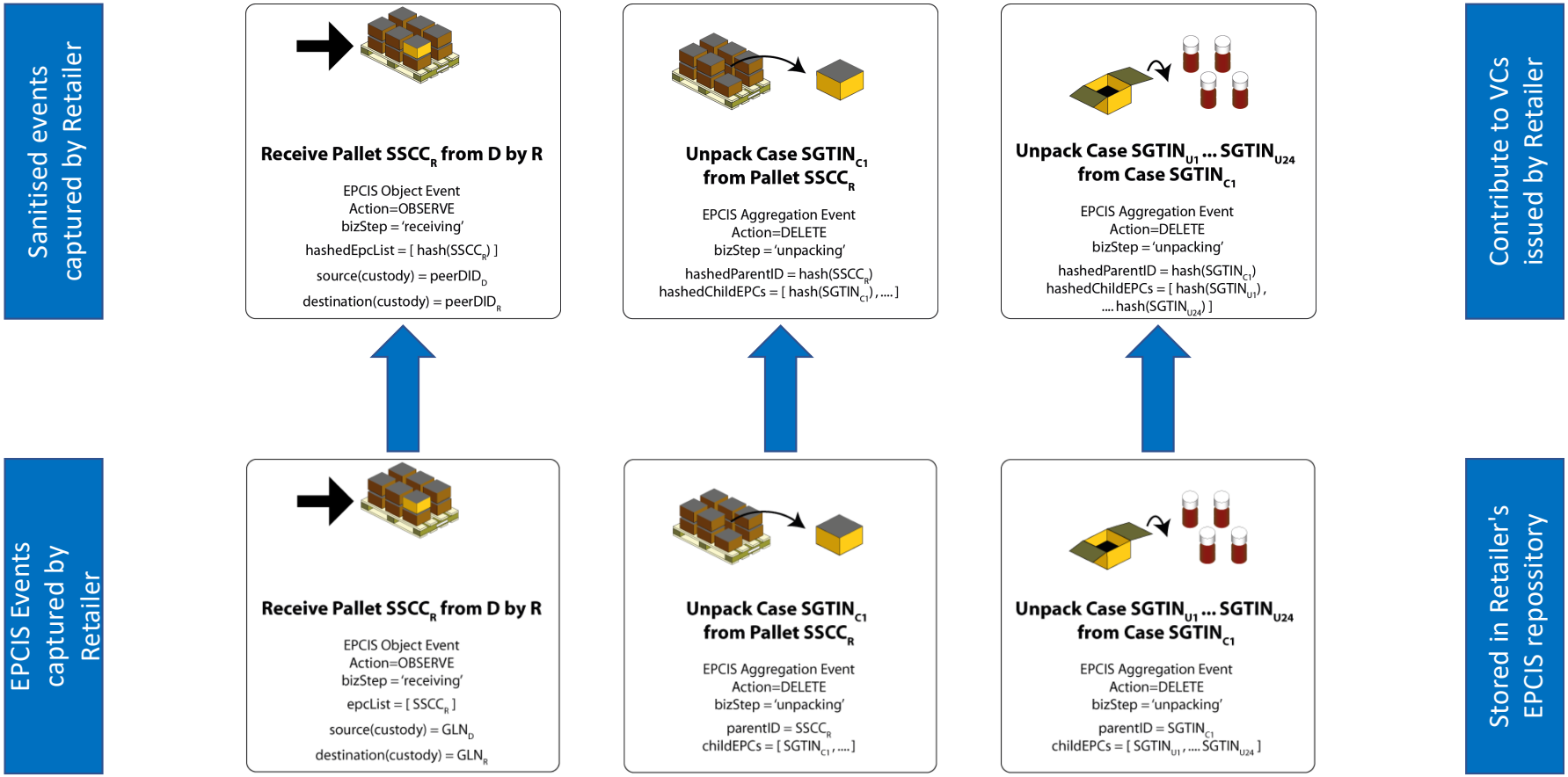

Figure 5 – an example of EPCIS events captured by a distributor and their transformation into a sanitised format

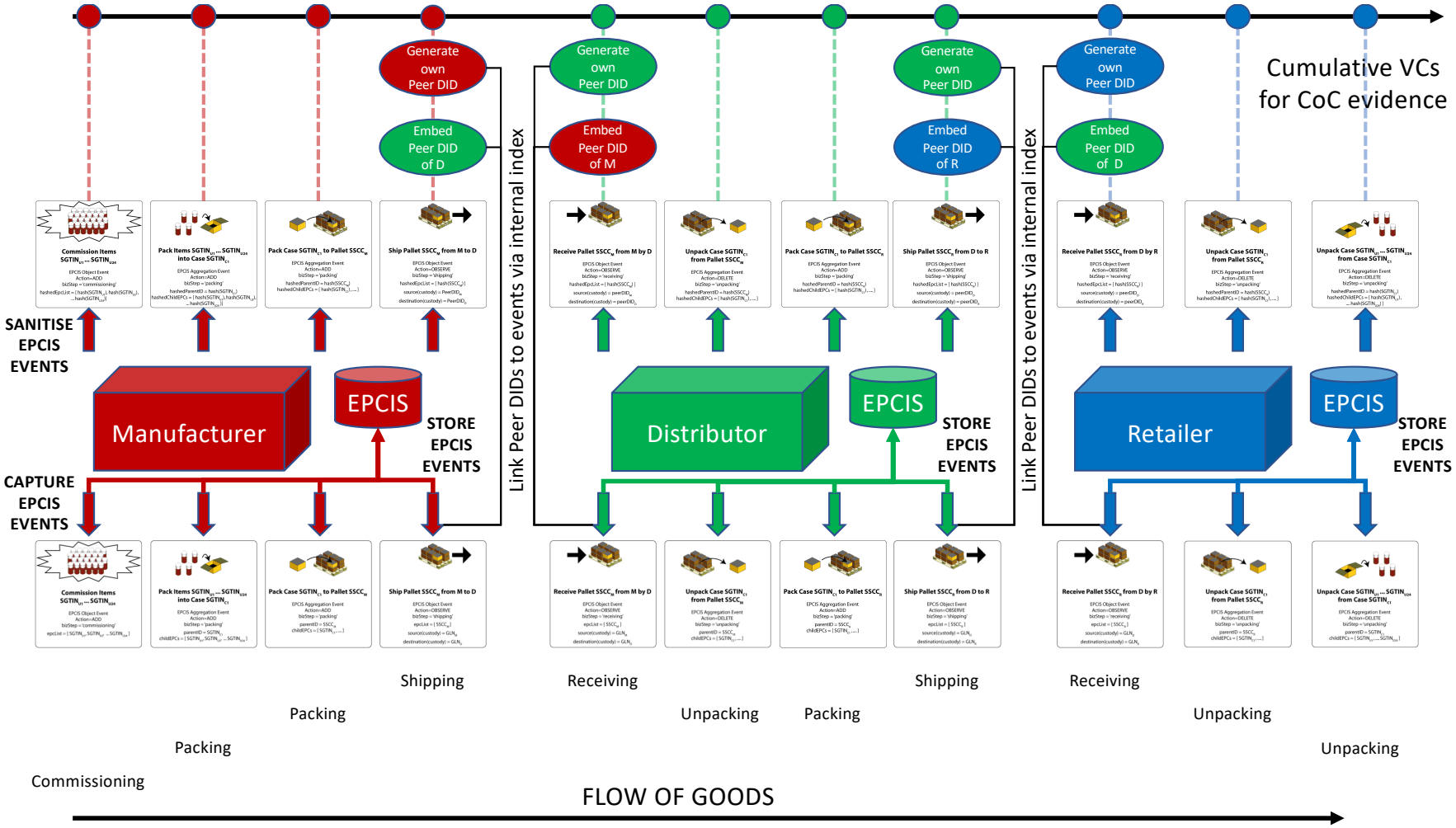

Figure 6 – an illustration of end-to-end navigation through a set of sanitised event data expressed as verifiable credentials.

4 Sanitization of EPCIS event data

A Verifiable Credential (or set of related Verifiable Credentials) to prove end-to-end connectedness for a specific object of interest (identified by a hash of its URI value) should provide:

- A list of digitally signed sanitised/transformed events in which:
  - Timestamps (`eventTime`) are unchanged, to support correlation with the original timestamp by the authoring organisation
  - `eventID` (if specified) is also unchanged, also to support correlation with the original eventID value by the authoring organisation
  - `bizStep` value is unchanged. This may be needed for the automated navigation algorithm to navigate through a chain of events end-to-end.
  - IDs of physical objects are replaced with their hash values in EPCIS event data fields such as `epcList`, `parentID`, `childEPCs`, `inputEPCList`, `outputEPCList`
  - URIs based on Global Location Number for party/organisation (417) or physical location (414)+(254) are replaced by DIDs or peer DIDs in EPCIS event data fields such as `source`, `destination`, `readPoint` and `bizLocation`
- The Peer DID of the VC bearer/holder (such that the bearer/holder can interactively prove ownership of the peer DID that appears in the event list)
- The original URI of the object of interest (whose hash value should appear in at least one of the events in the sanitised list)
- A list of all organisation DIDs (including peer DIDs) involved in the chain of custody or chain of ownership
- Potentially other summary information resulting from an automated chain navigation algorithm – although this may be redundant information

The idea is that the querying organisation holding the Verifiable Credential (VC) or set of Verifiable Credentials (VCs) should be able to present the VC(s) to an organisation elsewhere on the supply chain (usually further upstream than its immediate supplier) as a justification for making a query request for the original event data about a specific object of interest (whose traceability data is being checked) and that the set of one or more Verifiable Credentials proves connectedness between the querying organisation and the organisation being queried via a chain of custody or chain of ownership for a specified object of interest. As explained in section 1, the list of sanitised/transformed events should not leak commercially sensitive business intelligence in plaintext about trading relationships, production volumes, inventory volumes, flow rates, etc.

Only an organisation who received an actual object should know the original EPC URI whose hash value appears in the event list of the VC. Because hashing algorithms such as SHA-256 are a one-way function, this makes it difficult for outsiders to misuse any Verifiable Credential that they already have in order to ask about an unrelated object whose hash value is not mentioned in that VC.

A Decentralised Identifier (typically a Peer DID) of the querying organisation must appear in the event list and will typically also appear within the `destination` data field of the shipping event of their immediate '1-up' supplier.

The querying organisation must be able to prove ownership of the DID or PeerDID through the usual methods involving digitally signing a random challenge, using their secret private key for which the public key is used to derive the DID through a well-known algorithm.

Figure 7 shows a snippet of event data appearing within an EPCISDocument in JSON-LD format showing an example of EPCIS event data for commissioning of objects, packing and shipping. Additional events can also be captured for receiving, unpacking, transformation of input objects into output objects, using any of the standard event types defined in the EPCIS standard, as well as custom event types that extend the standardised superclass `EPCISEvent`.

The first event in Figure 7 represents the creation / commissioning of an object with Product GTIN 10614141073464 and Serial Number 2017 by Company A.

The second event represents the aggregation / packing of that object into a case identified by SSCC 106141412345678908, also performed by Company A.

The third event represents the shipping of that case identified by SSCC 106141412345678908 from Company A to Company B..

Figure 8 shows how the snippet of event data shown in Figure 7 after being sanitised to hide commercially sensitive details while still enabling proof of connectedness. Note that URI values of `epcList`, `parentID`, `childEPCs`, `inputEPCList`, `outputEPCList` are replaced by their hash values, while the identifier values of locations and organisations (typically expressed within fields such as `readPoint`, `bizLocation`, `source` or `destination` using EPC URNs or GS1 Digital Link URIs based on GS1 Global Location Number identifiers) are replaced with peer Decentralised Identifiers that may be generated per individual shipment and are typically only known bilaterally between the shipping organisation and receiving organisation. These are indicated in Figure 8 as placeholders within angle brackets, for greater readability than the actual values that mostly resemble strings of random characters . Values of `eventTime` and `eventID` should not be changed. These permit server-side correlation with the original non-sanitised event data stored within an EPCIS repository.

```
{
 "@context": ["https://gs1.github.io/EPCIS/epcis-context.jsonld"],
 "eventList": [
    {
       "eventID": "ni:///sha-256;50d11ff2139cd208d15f44eb60f842cf19a86f65a54b47cbefdcb85a1f8d5456?ver=CBV2.0",
       "type": "ObjectEvent",
       "action": "OBSERVE",
       "bizStep": "commissioning",
       "epcList": ["https://A.example.com/01/10614141073464/21/2017"],
       "eventTime": "2005-04-03T20:33:31.116000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "https://A.example.com/414/0614141073467/254/1374" }
    },
    {
       "eventID": "ni:///sha-256;0889a8f39e3aed53cb57b63ee65f865ff63904b56b5ce45b34e728c504391bd7?ver=CBV2.0",
       "type": "AggregationEvent",
       "action": "ADD",
       "bizStep": "packing",
       "parentID": ["https://A.example.com/00/106141412345678908"],
       "childEPCs": ["https://A.example.com/01/10614141073464/21/2017"],
       "eventTime": "2005-04-03T20:43:35.117000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "https://example.com/414/0614141073467/254/9875"}
    },
    {
       "eventID": "ni:///sha-256;60fa7cf55d3880cb05e13a389d3bd23972bc072acb8519f16e1362dad8f22e4e?ver=CBV2.0",
       "type": "ObjectEvent",
       "action": "OBSERVE",
       "bizStep": "shipping",
       "epcList": ["https://A.example.com/00/106141412345678908"],
       "eventTime": "2005-04-03T20:53:39.119000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "https://example.com/414/0614141073467/254/3859" },
       "sourceList": [ {"type":"possessing_party","source":"https://A.example.com/417/0614141073467"} ],
       "destinationList": [ {"type":"possessing_party","destination":"https://B.example.com/417/061414100001"} ]
    }
 ]
}
```


Figure 7 – snippet of typical EPCIS event data in JSON-LD format

```
{
 "@context": ["https://gs1.github.io/EPCIS/epcis-context.jsonld"],
 "eventList": [
    {
       "eventID": "ni:///sha-256;50d11ff2139cd208d15f44eb60f842cf19a86f65a54b47cbefdcb85a1f8d5456?ver=CBV2.0",
       "type": "ObjectEvent",
       "action": "OBSERVE",
       "bizStep": " commissioning",
       "hashedEPCList": ["ni:///sha-256;d110cb40d477be3d082d55f59eac94cfd63fcb4f4df397858007cf92e16258f2"],
       "eventTime": "2005-04-03T20:33:31.116000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "<DID_for_CompanyA>" }
    },

    {
       "eventID": "ni:///sha-256;0889a8f39e3aed53cb57b63ee65f865ff63904b56b5ce45b34e728c504391bd7?ver=CBV2.0",
       "type": "AggregationEvent",
       "action": "ADD",
       "bizStep": "packing",
       "hashedParentID": ["ni:///sha-256;87cb40642e028de540a3c275c1d629846c07001d1e683480f8723ec2149eaeca"],
       "hashedChildEPCs": ["ni:///sha-256;d110cb40d477be3d082d55f59eac94cfd63fcb4f4df397858007cf92e16258f2"],
       "eventTime": "2005-04-03T20:43:35.117000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "<DID_for_CompanyA>"}
    },

    {
       "eventID": "ni:///sha-256;60fa7cf55d3880cb05e13a389d3bd23972bc072acb8519f16e1362dad8f22e4e?ver=CBV2.0",
       "type": "ObjectEvent",
       "action": "OBSERVE",
       "bizStep": "shipping",
       "hashedEPCList": ["ni:///sha-256;87cb40642e028de540a3c275c1d629846c07001d1e683480f8723ec2149eaeca"],
       "eventTime": "2005-04-03T20:53:39.119000-06:00",  "eventTimeZoneOffset": "-06:00",
       "readPoint": {"id": "<Peer_DID_for_CompanyA_for_a_particular_shipment>" },
       "sourceList": [
          {"type":"possessing_party","source":"<Peer_DID_for_CompanyA_for_a_particular_A-B_transaction>"}
       ],
       "destinationList": [
          {"type":"possessing_party","destination":"<Peer_DID_for_CompanyB_for_a_particular_A-B_transaction>"}
       ]
    }

]}
```


Figure 8 – a sanitised version of the snippet of typical EPCIS event data shown in Figure 7.

5 Supply chain navigation algorithm

An algorithm for automated gathering/navigation of event data along the individual chain of custody or ownership of a specific product instance, asset or logistic unit has been developed and documented in more detail in work done within the former GS1 Event-Based Traceability Mission-Specific Work Group around 2014, as a functional component within 'Checking Services' that could automatically gather visibility event data end-to-end, analyse it for gaps and inconsistencies, then return actionable information about which identified objects were considered to have credible provenance and which should be quarantined for further investigation and closer physical inspection.

The essential ideas for such an algorithm are as follows:

- During tracking / tracing, the algorithm maintains a number of stacks or lists, the most important being the Object Stack; the object that is currently being tracked or traced is at the first position of the Object Stack and usually refers to an outermost container; other elements in the Object Stack refer to child objects that were previously the objects being tracked or traced before aggregation/disaggregation took place.
- For tracking downstream, when an aggregation (packing) event is encountered, the parent object ID is extracted and added to the front of the Object Stack. The algorithm continues tracking for this parent object ID until further aggregation or disaggregation events are encountered. When a disaggregation (unpacking) event is encountered, if the event indicates that a child object already in the Object Stack has been unpacked from its parent, the parent object is removed from the front of the Object Stack and tracking downstream continues for the child object (now at the front of the Object Stack). Multiple layers of parent objects (representing totes or cases, pallets, shipping containers etc.) may be added and later removed to the front of the Object Stack.
- For tracing upstream, the algorithm is similar except that the triggers are reversed; encountering a disaggregation event results in the parent object ID being added to the front of the Object Stack and the parent ID being traced further upstream; encountering an aggregation event results in the parent object ID being removed from the front of the Object Stack and the child ID (now at the front of the Object Stack) being traced further upstream.
- Similar logic applies for each transformation event encountered except that there is no need to maintain a hierarchical stack of nested object identifiers, since transformation events are irreversible (input objects into output objects).
- When encountering shipping and receiving events (EPCIS ObjectEvent with `bizStep` values indicating `shipping` / `receiving`), the algorithm for automated event gathering checks for consistency between each pair of shipping and receiving event, namely that the `source` and `destination` fields agree across the pair of events and that the same object ID (e.g. instance-level identifier of a product or logistic unit) appears in each event.

The algorithm should be able to navigate event data automatically within each organisation from the time of receiving inbound goods (or their manufacture) until the time of shipping of outbound goods to the next organisation in the supply chain network.

If the querying organisation is granted access to the corresponding plaintext EPCIS event for "`bizStep`" : "`shipping`", the value of the `destination` field may be a URI based on a GS1 Global Location Number (GLN). Since resolver infrastructure for GS1 Digital Link URIs can support redirection based on GLN values expressed within the GS1 Digital Link URIs, it may be possible for the querying organisation to discover the address of the EPCIS repository for the next organisation that received the goods, then repeat their query, presenting the evidence that they have accumulated so far, as evidence of their connectedness.

The set of Verifiable Credentials for objects could be transmitted downstream as goods flow through the supply chain, so that each organisation that receives products or logistics units has a complete set of Verifiable Credentials that correspond to the flow of goods.

In one scenario, this might be sufficient evidence to initiate an EPCIS query within the corresponding manufacturer to request the commissioning event (creation event) for a specific product instance. The manufacturer can check that the set of Verifiable Credentials represents an unbroken chain. The querying organisation then proves two things to the manufacturer or other organisation being queried:

(1) That they know the object identifier (e.g. the GTIN and the specific Serial Number for that product instance) by revealing its plaintext identifier (such as an EPC pure identity URN or the corresponding GS1 Digital Link URI) whose hash value appears in the Verifiable Credentials. If the hash value calculated for the plaintext value does not appear anywhere within the Verifiable Credential(s) then their query should be rejected.

(2) That they control one of the peer DIDs within the same sanitised EPCIS event data in which the hashed object identifier appears, where the sanitised EPCIS event data is itself digitally signed within the Verifiable Credential. They prove control of the peer DID by digitally signing a random challenge from the manufacturer, using the private key that corresponds to the public key from which the peer DID was derived. The manufacturer can then check that the digital signature can be verified by the public key from which the peer DID is derived and otherwise rejects the query if the digital signature cannot be verified.

Having passed these checks, the manufacturer may then check the original plaintext commissioning event for that specific product instance. If the values of the `eventTime` field (and `eventID` if specified) do not agree with the corresponding values in the Verifiable Credential, there is a mismatch and further investigation may be required. If they match, the manufacturer may grant the querying organisation to the original commissioning event for the specified product instance, although they are permitted to redact (exclude) details that the querying organisation is not entitled to see, such as details of any other product instances that may have been received by different organisations unrelated to the organisation making the query.

In another scenario, the querying organisation may be attempting to gather a complete set of EPCIS event data, starting at the manufacturer and working further downstream. The same considerations apply as in the previous scenario, except that now the manufacturer may return not only the initial 'commissioning' event but a complete set of events for that product instance (and its parent containers / logistic units) as far as the shipping event from the manufacturer to the next organisation, such as the first distributor. Such an automated chain navigation algorithm can be used internally by the manufacturer either on their original non-sanitised EPCIS event data or on the sanitised EPCIS event data provided via Verifiable Credentials by the querying organisation. Details of the `eventTime` and `eventID` (if specified) can also be used to quickly retrieve the corresponding original event data, while the hashed values of object identifiers can be used to determine which plaintext values of object identifiers the manufacturer should redact from the non-sanitised EPCIS event data that it provides to the downstream querying organisation, typically only retaining object identifiers whose hash value appears within the corresponding sanitised data present in the Verifiable Credentials provided by the downstream querying organisation.

The net result of this is that the manufacturer may then be willing to release a copy of the original EPCIS shipping event, albeit with suppression of unrelated object identifiers. When the querying organisation receives this, they may examine the `destination` field and may discover a plaintext URI identifier (such as an EPC URN or GS1 Digital Link URI) that identifies the next organisation on the chain who received the goods from the manufacturer. By formulating a GS1 Digital Link URI and making a Web request that specifies a `linkType` value of "`gs1:epcis`" (https://gs1.org/voc/epcis) it should be possible to find an EPCIS repository for the second organisation in the supply chain network, such as the first distributor. The downstream querying organisation then repeats the process and may receive a set of EPCIS event data collected by that organisation, potentially leading to a further shipping event and the ability to then discover the next organisation in the individual supply chain path, ultimately leading to the downstream querying organisation itself after a sufficient number of iterations.

References

GS1 CBV Standard

- https://www.gs1.org/epcis

GS1 Digital Link Standard

- https://www.gs1.org/standards/gs1-digital-link

GS1 EPCIS Standard

- https://www.gs1.org/epcis

JSON = JavaScript Object Notation

- https://tools.ietf.org/html/rfc8259

JSON-LD = JavaScript Object Notation for Linked Data

- https://www.w3.org/TR/json-ld/

W3C Use Cases and Requirements for Decentralised Identifiers

- https://www.w3.org/TR/did-use-cases/

W3C Verifiable Credentials data model

- https://www.w3.org/TR/vc-data-model/

Contributors to this White Paper

Name	Organisation
Dr Mark Harrison	Consultant to GS1 Global Office

Change Log:

0.4	1 April 2022	This initial public draft, providing additional explanation and introductory material, as well as improved alignment with current draft EPCIS 2.0 schema
0.3	November 2020	Development of Figures 3-8 within GS1 Innovation work on Verifiable Credentials and Decentralised Identifiers
0.2	July 2016	Internal GS1 paper on use of blockchain concepts to support chain of custody evidence
0.1	October 2014	Development of event gathering algorithm / chain navigation algorithm within GS1 Event-Based Traceability MSWG

Disclaimer