Available online at www.centmapress.org

Proceedings in
System Dynamics and Innovation in Food Networks 2018

INTERNATIONAL JOURNAL ON FOOD SYSTEM DYNAMICS

Innovative Solution Approach for Controlling Access to Visibility Data in Open Food Supply Chains

Ralph Tröger; Sebastian Clanzett; Richard Joachim Lehmann

GS1 Germany, Maarweg 133, 50825 Cologne, Germany

troeger@gs1-germany.de; clanzett@gs1-germany.de; lehmann@gs1-germany.de

ABSTRACT

Visibility data (providing details about supply chain activities in e.g. production, logistics, and quality processes) is of highly sensitive nature – not just in the food sector, but also beyond. Amongst other things, unauthorized data access can be (mis)used to uncover supply chain relationships, volumes, and other business context information. At the same time, it becomes increasingly important to share visibility data with trading partners, e.g. to meet customer requirements and legal obligations. So far, it is not a trivial matter to access or even discover that data, which is often stored in numerous distributed databases.

A possible means to overcome this predicament is a Discovery Service (DS), which has knowledge of the parties owning information about specific objects (e.g. product batches) and can provide pointers to the actual data sources to authorized clients while leaving no opportunity to misuse accessible data. It is important to note that a DS itself does not contain actual visibility data, but only references to it. Yet, even the knowledge that party A, B and C have information about a specific product is still sensitive as the querying client would be able to reveal the flow of goods and may take advantage of that knowledge. For instance, he could identify his supplier's upstream vendor and, for the sake of saving costs, try to procure products directly from that upstream vendor rather than from his previous supplier. Hence, a DS should provide the ability to prevent such scenarios if it strives to achieve a high level of acceptance.

A key enabler for a DS is a trust-inspiring means to ascertain whether a querying party is actually part of a given chain of custody (CoC). In conjunction with a set of rules previously defined by the respective data owners, the DS is then able to decide which service addresses (if at all) can be provided in the message response to the query client. Afterwards, the querying party can gather detailed information about a specific object by querying the indicated data sources. In this context, our paper addresses the following research question: *How can a DS ascertain if and to which extent a trading partner, whether known or not, is entitled to get pointers to visibility data stored in distributed repositories while ensuring privacy of the data owners?*

In pursuing this research question, we first provide the relevant background including the current state of the art. Second, we explain and discuss the solution approach and complete our paper by outlining the solution we are going to realize in the course of the research project 'FoodAuthent' (funded by the *German Ministry of Food and Agriculture*).

Keywords: Discovery Service; Access Control; EPCIS; GS1; Blockchain; Immutable Shared Ledger; Traceability

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

1. Problem statement and state of the art

Let us assume there is a supply network consisting of several raw material suppliers, manufacturers, distributors, logistics service providers, retailers and further parties (e.g. laboratories). For traceability purposes, each of these parties keeps a record of relevant events occurring to traceable objects (e.g. products, logistics units and assets) "… during their lifecycle, such as receiving, transforming, packing, shipping, [and] transporting." (*GS1* 2017a, 16) In the food industry, further examples of such critical tracking events (CTEs) encompass harvesting, processing, and quality inspection. For illustration purposes, table 1 depicts three exemplary CTEs.

Table 1. Exemplary critical tracking events in the food sector

Key Data Element	CTE 1	CTE 2	CTE 3	…
Who	Farmer A (ID: GLN)	Distributor B (ID: GLN)	Distributor B (ID: GLN)	…
What	Product lot P-1, 950 kg (ID: GTIN + lot number)	Product lot P-1, 400 kg (ID: GTIN + lot number) and logistics unit L-2 (ID: SSCC)	Logistics unit L-2 (ID: SSCC)	…
Where	Location of farmer A (ID: GLN)	Location of Distributor B (ID: GLN)	Location of Distributor B (ID: GLN)	…
When	Date + Time	Date + Time	Date + Time	…
Why	▪ Process: Harvesting ▪ Lot-level master data (certification data, origin data, etc.)	▪ Process: Packing ▪ Business transaction: purchase order 11	▪ Process: Shipping ▪ Source: Distributor B ▪ Destination: Retailer C ▪ Business transaction: despatch advice 12	…

Legend: GTIN … Global Trade Item Number | GLN … Global Location Number | SSCC … Serial Shipping Container Code

Such visibility data is useful for all members of a supply chain and for a large number of situations and use cases, e.g. check for compliance (based on e.g. legal or customer-specific requirements), recall management, meeting consumer's information needs (e.g. by providing a subset of that data to consumer apps), or process optimization. Given the relevance and value of that data, trading partners in the food sector need an appropriate way to share it. Table 2 provides an overview of all approaches that were either designed, piloted, or implemented to date (for further information, please refer to *GS1* 2017a, p. 20 onwards).

In the food sector, two of the listed models are prevailing so far: one step up – one step down (usually by exchanging bilateral EDI messages), and central repositories (e.g. as provided by traceability platforms such as *ezTRACK*, *foodLogiQ*, *fTRACE*, and *Tracetracker*). The first one is becoming increasingly inappropriate to fulfil the information requirements of various stakeholders (see e.g. *Safe Food Alliance* 2016; *EC* 2010; *Eigenmann et al.* 2012, p. 15 onwards), for instance:

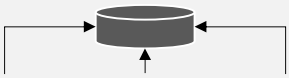
- access to visibility data beyond one tier up- and downstream (i.e. kept by supply chain parties which are sometimes – e.g. due to spot market procurement especially in the fresh food sector – a priori unknown) to enable end-to-end visibility
- global scalability to ease integration of many, inherently distributed organizations with heterogeneous IT applications
- fast data accessibility to enable prompt and effective reactions in case of supply chain issues
- on-demand data provisioning to allow for data sharing on a need-to-know basis

While central, cloud-based repositories (depending on e.g. their system architecture and service offerings) have the potential to meet the above mentioned requirements and feature additional merits such as lower overall cost, data access control management, and a higher degree of standardization (see e.g. *Troeger/Alt* 2017, 148 onwards), they usually cannot share the contained visibility data with other central repositories. There are two major reasons for that: (a) divergent interface and data standards (e.g. as to data formats, identification

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

schemes, data/vocabulary elements, etc.) and (b) the lack of an overall access control framework including a means to align their users' data access policies.

Table 2. Data sharing choreographies

Choreography model	Characterization	Illustration
One step up – one step down	▪ Visibility data is kept in local systems ▪ Data exchange via bilateral messages	
Centralized	▪ Visibility data is pushed to central repository ▪ Data access/exchange via query interface (if provided)	
Networked	▪ Visibility data is kept in local systems ▪ Any supply chain party is enabled to query for data (typically depending on access rights)	
Cumulative	▪ Visibility data is systematically enhanced and pushed forward to subsequent supply chain parties in parallel to flow of goods	
Decentralized and replicated	▪ Visibility data is systematically enhanced and all supply chain parties keep local copies of all data via decentralized replicated ledgers	

The first mentioned issue (a) can be addressed by leveraging global identification standards (e.g. the GTIN – Global Trade Item Number and the GLN – Global Location Number) and data sharing standards such as EPCIS along with its accompanying data standard, the Core Business Vocabulary (CBV). For detailed information, please refer to *GS1* 2018, *GS1* 2017c and *GS1* 2016.

So far, there is no appropriate way to address the second issue (b) though. For illustration purposes, consider the following fictitious scenario: Company 1 submits visibility data to central repository A and agrees to share a defined set of data with its successors while company 2 uses central repository B and wants to share some data with trading partner 3, 4 and 5, who also have established a connection to repository B. There currently is no means for repository providers A and B to exchange data access rights, let alone keep them in sync. Thus, there is no possibility for e.g. trading partner 5 to access company 1's visibility data in repository A even if it qualifies as one of company 1's successors.

Since a centralized approach in some cases entails a higher risk of data losses and attacks (aiming e.g. to access, delete or manipulate data in retrospect) as well as having an enhanced probability for accessibility issues (e.g. in case of high traffic), networked or decentralized solutions seem promising and may be chosen more frequently in the future. At the same time, a cumulative data sharing approach (the last available choreography model presented in table 2) does not fulfil several of the above-mentioned requirements and is thus not considered further.

Irrespective of whether companies opt to share their visibility data following a networked, centralized or decentralized approach (or mix of thereof), they are confronted with the so-called discovery problem which "… is concerned with how to directly share data between parties that are connected in a chain but do not have a direct relationship." (*GS1* 2017a, 36) The discovery problem comprises three elements: 'chaining' (determine whether a company is connected to another one), 'trust' (if chaining is confirmed, set up the conditions of trust required to share data with each other), and 'data transfer' (if trust is established, accomplish the actual data sharing).

In answering the research question formulated at the beginning, this paper contributes to the first and second element of the discovery problem.

2. Solution approach

In a nutshell, our suggestion to address the 'chaining' and 'trust' issue consists of the following principles:

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

(1) an end user company captures visibility event data while making use of standardized identifiers, data structures, and data elements
(2) an operator of an event repository sends a concise subset of these visibility data to an immutable ledger (distributed and shared with any party able and willing to provide visibility data) while obscuring any sensitive data contained in using a consistent hashing approach
(3) a provider of a DS transfers these sanitized visibility data into a graph database and applies a globally harmonized data access rights framework which, in conjunction with standardized role definitions and data elements, serves as the basis to determine if queries for visibility event data are legitimate

Figure 1 gives an overview of how the 'chaining' and 'trust' issue can be tackled. The picture also illustrates the flexibility of the suggested approach. First, it supports setups in which end user companies submit visibility data to cloud platforms which offer both an EPCIS repository as well as a means to provide sanitized CTEs to a shared ledger for access control purposes (see service providers A, B and companies 1, 2, 3). Second, it supports situations in which an organization (here: company 4) chooses to record visibility data in a local EPCIS repository while only providing corresponding sanitized datasets to a dedicated discovery service (in this case, provided by DS provider I). Third, it supports configurations in which a company not only holds its visibility data in an own EPCIS repository, but also keeps an own copy of the shared ledger (see company 5).

Taking the example of a shipping event of company 3 (similar to CTE 3 in table 1), the entire process would work as follows: First, company 3 would store an EPCIS shipping event in the cloud-based EPCIS repository of visibility service provider B. Second, B transforms the EPCIS event into a concise, sanitized transaction only containing obscured, i.e. hashed object, location and party identifiers. This transaction is then submitted to an immutable shared ledger, which is also accessible by other parties.

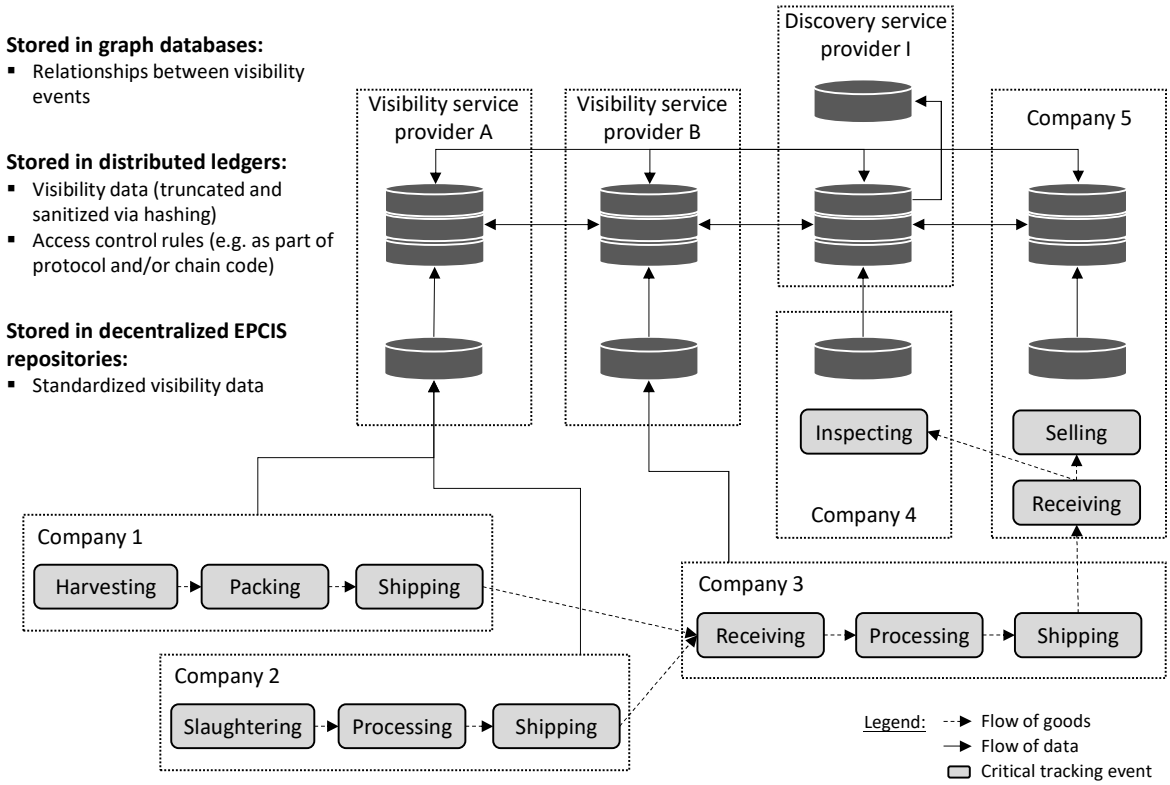


Figure 1. Solution approach illustration

Now, if company 5, which is not providing its own visibility data to the repository of B, is interested in obtaining details of that shipping event, it queries a DS that – based on the data stored in the shared ledger – can find out that 5 is actually a subsequent member of the very same chain of events as 3. Given that 3 has specified that all successors are permitted to obtain further details of its shipping events, the DS returns the service endpoint of the repository containing the actual event (ideally, accompanied by an authorization token), and 5 can access the latter. Considering the three above-mentioned principles, the following paragraphs explain how that actually can be accomplished.

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

2.1 Capturing visibility data in a standardized manner

In order to ensure that visibility applications offer a high level of cost efficiency, scalability, flexibility and investment security, the usage of globally established standards is of paramount importance. Thus, when designing a visibility system, especially when modelling the respective CTEs, organizations should make use of standardized identifiers, vocabulary elements and data structures to the greatest possible extent. For an intuitive description of how to accomplish the latter, please refer to chapter 4 of GS1's 'EPCIS and CBV Implementation Guideline' (*GS1* 2017c, 21 ff.). The guideline advocates an eight-step process starting from the collection of visibility goals/requirements and ending with the documentation of the identified visibility events in a visibility data matrix.

2.2 Sanitizing and truncating visibility data for data access control purposes

First, when providing information on visibility data to third parties for access control purposes, it is only required to focus on the essential data content. E.g., it is not necessary to include any quantities, instance-/lot-level master data attributes (e.g. best before date) or user-specific extension elements. Therefore, the dataset only needs to contain key information on the what (i.e. object identifiers), when (timestamp of the event), where/who (i.e. location and party identifiers) and why (i.e. data elements providing information on the business context). Thus, this process is not about duplicating the actual visibility events, but stripping them down to their very core. As these datasets are potentially shared with many different nodes, which entails a growing amount of computing power and storage the bigger the individual data structures are designed, data economy also comes in handy from a technical perspective. In this regard, the datasets should feature a lightweight format (e.g. JSON or JSON-LD).

Obviously, some elements of the datasets – especially object, party and locations IDs – cannot be stored in plain text. Once submitted to a shared ledger, which is accessible to many trading parties and competitors, it otherwise would be immediately transparent which company has bought or sold which products at which time. (Note that there are various ways to obtain the meaning of a given identifier. For instance, if an organization wants to know which company is associated with a specific GLN or GTIN, a query to the GEPIR web service (http://gepir.org) returns a set of corresponding master data.)

In principle, we can choose from two basic approaches to obscure identifiers: encryption and hashing. Apparently, symmetric encryption is not an option: once the key is shared with a third party, there is no way to curtail data access at a later point of time. Further, it bears the risk that the key is leaked to parties that should not have access to the data in the first place. Likewise, asymmetric encryption is not a viable option either: On the one hand, there would be only one party (i.e. the owner of the private key) able to decrypt the content of a given dataset, whereas there is often the need to share visibility data with multiple parties. On the other hand, companies would face an enormous effort for managing the respective keys.

Hashing however comes with a couple of inherent advantages. First of all, it is impossible to infer the original input string (i.e. in our case, an identifier) of a hash function. Additionally, for a given input string, the resulting hash value is always the same. For instance, when using SHA-256, 'test' always results in the following hash value: '9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08'. Even if all sensitive data is obscured through hashing, it helps to address both the 'chaining' as well as the 'trust' challenge of the discovery problem: 'Chaining' can be achieved independently of whether the identifiers are represented in plain text or whether they are hashed. What is more, it helps to build 'trust' as the identity of the data submitter cannot be disclosed and the hash of a specific product identifier can only be created by parties having knowledge of it (e.g. a specific GTIN + serial number).

A particular case is the hashing of the party and location IDs (i.e. GLNs), which in contrast to the object IDs (e.g. GTIN + serial number, GTIN + lot number, and SSCC) remain static. If companies would hash GLNs just once, querying parties could map a specific hash value to a given GLN and reveal sensitive business information. This is why we suggest at least a dual hashing for party and location IDs.

For location or party IDs as part of the 'why' dimension of a visibility event (e.g. source and destination in an EPCIS shipping or receiving event), we suggest organizations to use the respective ID of a previously agreed business transaction type (for instance, invoice, purchase, or despatch advise) in conjunction with the GLNs. As a specific business transaction ID is typically just known by a company and its immediate up- and downstream trading partners, only direct senders or recipients of goods can generate the twofold hash value indicating source or destination. For efficiency and process reliability purposes, the applied business transaction type should be indicated in the sanitized visibility dataset (of course, not the ID itself, though).

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

For location IDs as part of the 'where' dimension of visibility events (in EPCIS, read point and business location), we suggest data owners to keep a table containing an (internal) GLN masking code with an appropriate validity period (whose definitions are solely at the discretion of the data owner, see table 3). Further, if data owners wish that their identities were impossible to deduce, they should also keep a table indexing all submitted sanitized visibility datasets with a simple counter. In obscuring the actual GLN, it first hashes this counter value, the GLN as well as the GLN masking code and eventually hashes the resulting concatenated string.

Table 3. GLN masking code table for obscuring GLNs

Begin of validity period (DateTime)	End of validity period (DateTime)	GLN masking code (String)
…	…	…
2018-02-05T00:00:00.000+01:00	2018-02-11T23:59:59.999+01:00	1.FCK03l#
2018-02-12T00:00:00.000+01:00	2018-02-18T23:59:59.999+01:00	G3issb0c#
…	…	…

It is important to note that the same procedure is applicable whenever a use case requires an organization to proof that it e.g. actually was/is a legitimate owner of a specific product (i.e. is able to re-create the hash value). For that purpose, the company just has to look up the GLN masking code valid at the event time of concern as well as the respective counter value. Taking the example of GLN masking code '1.FCK03l#' and counter value '1' to obscure the GLN populating the read point as well as business transaction identifier 'urn:epc:id:gdti:4012345.12564.12' to obscure the GLNs populating source and destinations, table 5 contains exemplary computations.

To cater for flexibility for other hashing functions and to give a querying client a clear indication of the hashing function to be performed (e.g. for proofing the legitimacy of a request), the used hashing algorithm should be specified. In this regard, various solution approaches for hash URI notations have been discussed so far. For one, *Thiemann* (2003) suggests a URN-based URI, which however requires a registration with *IANA,* which has not taken place yet (*IANA* 2017). Another approach, suggested by *Leonard* (2015), is a URI syntax prefixed with the specific hashing algorithm name (e.g. 'sha256'). Similar to the first one, the respective URI schemes are not yet registered with *IANA*.

In contrast, the URI scheme as specified in RFC 6920 ('Naming things with Hashes') is already registered with *IANA* (*IANA* 2018). The authors of RFC 6920 propose "…a new URI scheme for this purpose (…) [and] a way to map these to HTTP URLs, and binary and human-speakable formats (…)" (*Farrell et al.* 2013). Thereby, the 'Named Information' (ni) URI consists of nine elements: the scheme name ('ni'), colon and two slashes ('://'), an optional authority, slash ('/'), digest algorithm (e.g. 'sha-256'), separator (';'), digest value, query parameter separator ('?', if applicable), and optional query parameters. As the ni URI scheme is officially available and based on an Internet Standard approved for publication by the Internet Engineering Steering Group, this paper follows that approach.

The only information that needs to be added to the sanitized visibility dataset is a reference to the source of the actual visibility event, so that a DS is able to provide a pointer indicating where to query for it. Again, due to privacy reasons, that information should not be included in plain text (e.g. if a data set contained the actual service endpoints such as https://query-service.epcis.henkel.de or https://epcis-query-service.eztrack.com, it obviously is quite easy inferring the parties involved or at least narrowing down the geographic region of the data submitters).

What is more, data providers should have the flexibility to change their respective service endpoints as they see fit. For this purpose we suggest, quite similar to the above-introduced dual GLN hashing, to obscure the service endpoint by hashing the service URL in conjunction with an interface masking code and to synchronize the contents of that mapping table (see table 4, without the internal interface masking code) with the respective DS provider(s). In an analogous way to the GLN masking code, the interface masking code should change in appropriate frequency, so that there only remains a negligible risk for data abuse.

Table 4. Interface masking code table for obscuring interface IDs

Hash value (String) (abridged here)	Interface masking code (String)	Start of validity period (DateTime)	End of validity period (DateTime)	Service endpoint (String)
…	…	…	…	…
ni:///sha-256; 1a2b(…)	Koe1nerD*m	2018-02-05 T00:00:00.000+01:00	2018-02-06 T23:59:59.999+01:00	https://epcis-query.ftrace.com
ni:///sha-256; 9b8c(…)	Koe11eA~af!	2018-02-07 T00:00:00.000+01:00	2018-02-08 T23:59:59.999+01:00	https://epcis-query.ftrace.com
…	…	…	…	…

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

Putting it together, table 5 illustrates how a CTE is captured as an EPCIS-based visibility event (while making use of globally standardized identification schemes and data elements) and how that visibility event in turn is converted into a sanitized visibility dataset ready to be shared via immutable ledgers set up for data discovery/access control purposes.

Although the visibility data is truncated and sanitized, it still enables a number of compelling use cases such as ensuring completeness of the chain of custody, validating product authenticity, proving legitimate possession of a specific asset, and performing plausibility checks. When stored on a shared ledger (e.g. based on blockchain technology), all these use cases still work even if the original data owner has gone out of business or has lost the actual visibility events.

Table 5. Converting CTEs into sanitized EPCIS events

Critical Tracking Event	Resulting EPCIS Event	Sanitized EPCIS Event (draft)

Who/Where
Distributor B, plant 123
(e.g. GLN '4012345000054')
 	
What
Logistics unit (e.g. SSCC '040123456666666665')
 	
When
Date + Time (e.g. 7th February 2018, 16:00 CET)
 	
Why
Process: Shipping
Source: Distributor B (e.g. GLN '4012345000009')
Destination: Retailer C (e.g. GLN '4023333000000')
Business transaction (e.g. despatch advice '12')
	…
	<EventList>
	<ObjectEvent>
	<eventTime>2018-02-07T15:00:00.000+01:00</eventTime>
	<eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
	<epcList>
	<epc>urn:epc:id:sscc:4012345.0666666666</epc>
	</epcList>
	<action>OBSERVE</action>
	<bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
	<readPoint>
	<id>urn:epc:id:sgln:4012345.00005.0</id>
	</readPoint>
	<bizTransactionList>
	<bizTransaction type="urn:epcglobal:cbv:btt:desadv">
	urn:epc:id:gdti:4012345.12564.12</bizTransaction>
	</bizTransactionList>
	<extension>
	<sourceList>
	<source
	type="urn:epcglobal:cbv:sdt:possessing_party">
	urn:epc:id:sgln:4012345.00000.0</source>
	</sourceList>
	<destinationList>
	<destination
	type="urn:epcglobal:cbv:sdt:possessing_party">
	urn:epc:id:sgln:4023333.00000.0</destination>
	</destinationList>
	</extension>
	</ObjectEvent>
	</EventList>
	…
		…
		{
		"objectEvent": {
		"eventTime": "2018-02-07T15:00:00.000+01:00",
		"epcs": {
		"id": "ni:///sha-256;7a817e8819881b9a5fb7117f94764e
		63c31160633a549ca064b69621b59461fc"
		},
		"action": "OBSERVE",
		"bizStep": "urn:epcglobal:cbv:bizstep:shipping",
		"readPoint": "ni:///sha-256;2bb4fbdeb9b8d1466b0a912b9
		318473df499e2a49921ddf068cd1cbf2853449b",
		"sources": {
		"id": "ni:///sha-256;a89cabda0db3d76c1339ec6591e077
		4857b40623ecc5f3d2dbd69cd85c9fabbe"
		},
		"destinations": {
		"id": "ni:///sha-256;1ba5958d76ebf4a6e9c1db34673386
		1377b402e2084387cde11113b26566498e"
		},
		"linkingBTT": "urn:epcglobal:cbv:btt:desadv",
		"interfaceID": "ni:///sha-256;f6d36b401b4e7458700ba25
		290947f46816603ffb6f659bdd53f01e15c9able7"
		}
		}
		…

2.3 Applying a globally harmonized data access rights framework

To date, recording data on distributed ledgers does not allow indicating any relationships between such sanitized visibility event datasets. Therefore, they need to be stored in a more appropriate manner. A suitable option is a graph database that can link logically connected events (e.g., those which share the same object identifier) and sort them into correct time order. In conjunction with previously agreed roles, an accessing application then can determine the exact relationship of various events. For scalability reasons, organizations should make use of globally standardized role definitions. As the latter are not available yet, table 6 gives a first idea how a globally harmonized role concept could look like.

Table 6. Conceptual design of globally aligned roles

Role of requesting party	Criteria for requesting party	Example (see figure 2)
Successor	Has captured an event with the same instance-/lot-level identifier (possibly (dis)aggregated and/or transformed further downstream) and a higher event time value	C, D
Predecessor	Has captured an event with the same instance-/lot-level identifier (possibly (dis)aggregated and/or transformed further upstream) and a lower event time value	A
Direct successor	Similar to successor, but requiring an event with an identical read point or destination ID as indicated in the shipping event's destination list captured by the event owner	C
Direct predecessor	Similar to successor, but requiring an event with an identical read point or source ID as indicated in the receiving event's source list captured by the event owner	A
Chain member	Any of the above	A, C, D
Third party organization	Specific ID of e.g. a brand owner, authority, etc.	(e.g.) E

Figure 2 displays a simple chain of events captured by various trading partners. Taking the example of a query request for the transformation event captured by processor B (highlighted in dark grey), it helps to understand the role concept introduced above. In this particular instance, farmer A would qualify as B's predecessor as he has made the very product ('GTIN + lot 123') consumed by processor B. At the same time, farmer A also is B's direct predecessor since B appears in the destination list of the related shipping event (and vice versa). Similarly, retailer D qualifies as successor, but not as a direct one.

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

Given B specified that a direct successor is allowed to obtain further details of its transformation events, distributor C is permitted to do so, but no one else. Likewise, if farmer A defined that all successors in the supply chain are entitled to see data of its harvesting events, distributor C and retailer D can query for them, without farmer A even need to know them in advance.

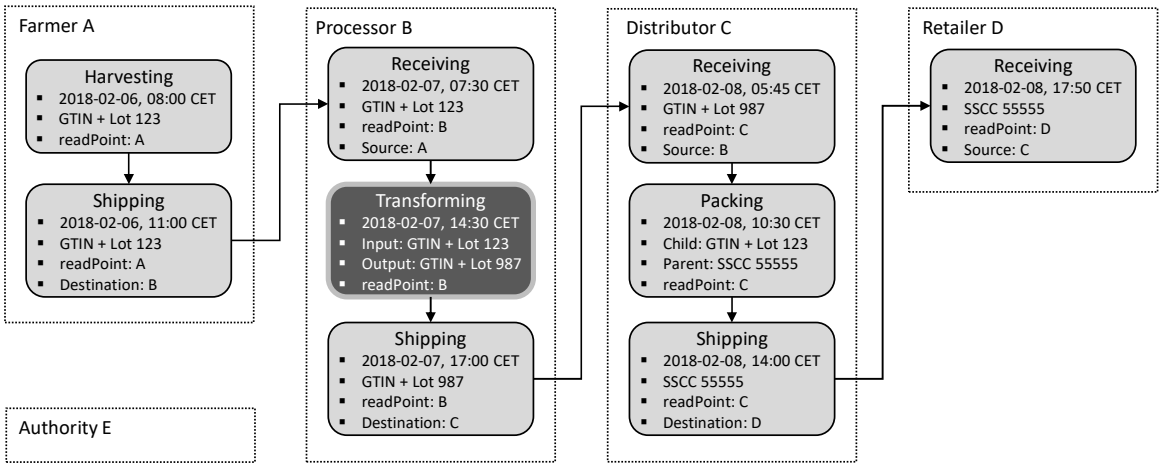


Figure 2. Exemplary chain of visibility events to illustrate the conceptual role design

Such data access rules are expressible in a vendor-independent, globally applicable manner as soon as the above-mentioned standardized role framework is available. Based on that, organizations can define who is entitled to obtain knowledge that they have data on specific products in the first place (i.e. get the query service endpoint to the actual visibility data).

In doing so, they can also confine the event profiles (e.g. harvesting, shipping, transforming, etc.) they generally are willing to share with a given supply chain role (see table 7). These data access definitions need to be shared through the immutable ledger as well, so that each DS provider can answer to information requests in a consistent and trustworthy manner.

Table 7. Data access framework

	Supply chain role			
Event profile	Successor	Predecessor	Direct successor	…
Transforming	x	x	x	
Packing			x	
Shipping	x		x	
…				

In a modified version, the same framework is also applicable for more fine-granular access control decisions when an external accessing application queries for actual visibility data in someone's EPCIS repository. Thereby, table 7 just needs to be extended with the attributes present in the respective event profiles. For instance, a company can define that a successor can access all the transformation events the successor is chained to, except for detailed quantity data and any business transaction IDs.

3. Outlook and Future Challenges

In the research project 'FoodAuthent', there are several use cases requiring a data access framework as described in the previous paragraphs. Particularly, the project aims to enable various stakeholders in the food sector to access visibility events captured by governmental or commercial food laboratories, e.g.: (a) inquire whether a specific product lot was checked for authenticity, (b) inquire details of an authenticity analysis, and (c) validate that files containing relevant information for authenticity checks are genuine.

This will entail a number of tangible benefits for organizations in the food sector. Amongst other things, (a) will enable organizations to embed that data in their respective business or consumer-facing applications (e.g. a mobile app for product information). Secondly, (b) has the potential to increase efficiency of food laboratories as it paths the way for enhanced collaboration between commercial, academic and authoritative food laboratories. Finally, (c) will enhance confidence and reliability in product authenticity operations.

In the course of the prototypical implementation, we will further refine the solution approach and investigate its feasibility, scalability, and limitations. For instance, it may be required to extend the sanitized event datasets

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

to ease data discovery. In addition to that, we expect a number of technical challenges (e.g. scalability of the graph database) we need to address. Furthermore, we will investigate mechanisms to prevent illegitimate operations (e.g. the submission of a sanitized visibility event to pretend to be a specific role for fraudulently obtaining access to data). Finally, yet importantly, we will develop a DS operator model (i.e. appropriate provider(s), organizational and technical set-up, and the actual provision of data), which is linked to the third element of the 'discovery problem' mentioned at the outset of our paper.

The access control mechanism as addressed in this paper will be an important element of the overall FoodAuthent framework, which consist of legal (e.g. terms of use), organizational (e.g. preconditions for participating organizations) and technical (e.g. data and interface standards) components.

Acknowledgement
We would like to take the opportunity to thank the *German Ministry of Food and Agriculture* for funding our project as part of its program on facilitating innovation based on a decision by the *Parliament of the Federal Republic of Germany*. Apart from *GS1 Germany*, the partners of the project 'FoodAuthent' are *Benelog*, the *German Federal Institute for Risk Assessment (BfR)*, *Eurofins Analytics*, *Lablicate*, and the Department of Computer and Information Science at the *University of Konstanz*. The three-year project is set up until September 2019. For further information, please refer to the project's website (https://www.foodauthent.de/).

References

Eigenmann, R.; Vucic, N.; Dillinger, M.; Viola, K.; Meyer, F.; Quesada Pimentel, D., Verhoosel, J.; Kaloxylos, A.; Lampropoulou, I.; István Gábor, I.; Perea Escribano, C.; Sundmaeker, H. (2012) [***Eigenmann et al.* 2012**]: Inventory of future capabilities of internet to meet future long and short term needs of the food sector, available at: http://cordis.europa.eu/docs/projects/cnect/6/285326/080/deliverables/001-SAFD70022V011Final.pdf (August 2017).

European Commission (2010) [***EC* 2010**]: Guidance on the implementation of articles 11, 12, 14, 17, 18, 19 and 20 of Regulation (EC) N° 178/2002 on general food law, available at: https://ec.europa.eu/food/safety/ general_food_law/general_requirements_en (September 2017).

Farrell, S.; Kutscher, D.; Dannewitz, C.; Ohlman, B.; Keranen, A.; Hallam-Baker, P. (2013) [***Farrell et al.* 2013**]: RFC 6920. Naming Things with Hashes, available at: https://tools.ietf.org/html/rfc6920 (February 2018).

GS1 (2018) [***GS1* 2018**]: GS1 General Specifications. The foundational GS1 standard that defines how identification keys, data attributes and barcodes must be used in business applications, release 18, available at: https://www.gs1.org/barcodes-epcrfid-id-keys/gs1-general-specifications (January 2018).

GS1 (2017) [***GS1* 2017a**]: GS1 Global Traceability Standard. GS1's framework for the design of interoperable traceability systems for supply chains, release 2.0, available at: https://www.gs1.org/traceability/traceability/latest (September 2017).

GS1 (2017) [***GS1* 2017b**]: GS1, IBM and Microsoft announce collaboration to leverage GS1 standards in enterprise blockchain applications, available at: https://www.gs1.org/articles/2256/blockchain-gs1-ibm-and-microsoft-collaborate-leverage-standards (September 2017).

GS1 (2017) [***GS1* 2017c**]: Core Business Vocabulary Standard. Specifies the structure of vocabularies and specific values for the vocabulary elements to be utilised in conjunction with the GS1 EPCIS standard, release 1.2.2, available at: https://www.gs1.org/epcis (October 2017).

GS1 (2016) [***GS1* 2016**]: EPS Information Services (EPCIS) Standard. Enables disparate applications to create and share visibility event data, both within and across enterprises, release 1.2, available at: https://www.gs1.org/epcis (October 2017).

IANA (2017) [***IANA* 2017**]: Uniform Resource Names (URN) Namespaces, available at: https://www.iana.org/assignments/urn-namespaces/urn-namespaces.xhtml (November 2017).

IANA (2018) [***IANA* 2018**]: Uniform Resource Identifier (URI) Schemes, available at: https://www.iana.org/assignments/uri-schemes/uri-schemes.xml (February 2018).

Leonard, S. (2015) [***Leonard* 2015**]: URI schemes for SHA-1 and SHA-256. Internet-Draft, available at: https://tools.ietf.org/html/draft-seantek-sha-uris-00 (November 2017).

Safe Food Alliance (2016) [***Safe Food Alliance* 2016**]: Traceability – fundamental requirements and the FSMA mandate, available at: https://safefoodalliance.com/newsletter/2016-01/traceability-fundamental-requirements-and-the-fsma-mandate (August 2017).

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817

Thiemann, P. (2003) [***Thiemann* 2003**]: A URN namespace for identifiers based on cryptographic hashes. Internet-Draft, available at: https://tools.ietf.org/html/draft-thiemann-hash-urn-01 (November 2017).

Tröger, R.; Alt, R. (2017) [***Tröger/Alt* 2017**]: Design options for supply chain visibility services: learnings from three EPCIS implementations, Electronic Markets, vol. 27, issue 2.

Tröger, R.; Reiche, R.; Schiefer, G. (2013) [***Tröger et al.* 2013**]: Benefits through utilizing EPC network components in service-oriented environments – an analysis using the example of the food industry. In: International Journal on Food Systems Dynamics, Special issue: food systems transparency and the future internet, vol. 4, no. 4.

DOI: http://dx.doi.org/10.18461/pfsd.2018.1817