



GLOBAL DIALOGUE
on Seafood Traceability

Standards and Guidelines

for Interoperable Seafood Traceability Systems

Technical Implementation Guidance (Version 1.2)

JUNE 2023

Document Summary

DOCUMENT NAME	<i>GDST Standards and Guidelines for Interoperable Seafood Traceability Systems - Technical Implementation Guidance</i>
Document Date	6/30/2023
Document Version	1.2
Document Status	Released JUNE 2023
Document Description	Additional technical guidance and recommendations to facilitate implementation of the GDST Core Normative Standards

This document is part of a packet of interconnected documents and resources that together constitute the full set of GDST materials. The packet includes:

Guide to GDST 1.0 Materials	JUNE 2023	v1.2	Overview of GDST packet contents and “How to Use These Documents”
Executive Summary	JUNE 2023	v1.2	Two-page description of GDST
Core Normative Standards	JUNE 2023	v1.2	Core Normative Standards
Basic Universal List of KDEs (spreadsheet)	JUNE 2023	v1.2	E-spreadsheet of appendices to Core Normative Standards - part of GDST core standards
Explanatory Materials	JUNE 2023	v1.2	Nontechnical background and introductory materials
Technical Implementation Guidance	JUNE 2023	v1.2	Additional technical materials to facilitate implementation

A drafting history of the industry-led inputs into GDST appears in Section 1.3 of the Explanatory Materials document.

For online access to the full GDST packet, visit traceability-dialogue.org/gdst-standards-and-materials

For additional information, please contact the GDST Secretariat at info@traceabilitydialogue.org

This document does not cover all aspects of the EPCIS, CBV, or GTSs of GS1 but utilizes and extends them for addressing IUU fishing and other sustainability challenges in seafood supply chains. We strongly recommend referencing the following documents when implementing the EPCIS extensions described in the *Core Normative Standards and Technical Implementation Guidance*:

- 1. GS1 EPCIS 2.0**
A GS1 standard for event-based traceability.
- 2. GS1 Core Business Vocabulary**
A GS1 standard that specifies the structure of vocabularies to be used in conjunction with GS1 EPCIS.
- 3. GS1 Digital Link**
A GS1 standard that covers information about master data and traceability data.

NOTE: Unlike the GDST Core Normative Standards, this document and other GDST explanatory materials are the product of the GDST Secretariat and have not been subjected to consensus decision-making by GDST members. These materials do, however, reflect extensive dialogue with GDST members and technical inputs from members and other external experts.

CONTENTS

Document Summary.....	3
Abbreviations and Acronyms.....	4
1. Traceability Data and Traceability Systems.....	5
1.1 Traceability Data within an Organization.....	5
1.2 - Traceability Data across the Seafood Supply Chain.....	6
1.3 - Managing Traceability Data.....	6
1.3.1 - Traceability Data Types and Sources.....	6
1.3.2 - Sensitivity and Data Security.....	7
1.3.3 - Data Sharing.....	7
1.3.4 - Quality and Verification.....	8
1.4 - Seafood Traceability Systems.....	8
2. Interoperability.....	9
2.1 - B2B Workflow.....	9
2.2 - Internal System to System (IS2S) Workflow.....	13
2.3 - Authentication.....	15
2.4 - Adding a Trading Party; EPCIS Data Format.....	16
2.5 - Master Data.....	17
2.5.1 - GS1 Web Vocab JSON-LD Examples	18
2.6 - GS1 Digital Link.....	18
2.6.1 - Open API Definition.....	19
2.6.2 - Example GS1 Digital Link Request #1.....	23
2.6.3 - Example GS1 Digital Link Request #2.....	23
2.7 - EPCIS Query Interface.....	26
2.7.1 - Open API Definition.....	26
2.7.2 - Example HTTP Request #1.....	28
2.7.3 - Example HTTP Request #2.....	30
3. Critical Tracking Event Examples - Wild-Caught Tuna to Canned Tuna.....	32
4. Critical Tracking Event Examples - Aquaculture.....	37
5. Internal Traceability - Commingling and Transformations.....	40
6. Traceability Data Use Cases.....	41
6.1 Traceback.....	41
6.2 Trace Forward.....	42
6.3 - Aggregation Report for CSR	43
6.4 - Mass Balance.....	43
6.5 - Chain of Custody.....	44
7 - Disposition for Product First Entering Commerce.....	44
Appendix 2.....	45
GDST GitHub.....	49

ABBREVIATIONS AND ACRONYMS

AIDC	automated identification and data capture
ALE	application-level events
API	application programming interface
B2B	business-to-business
BUL	Basic Universal List
CBV	Core Business Vocabulary (GS1)
CPG	consumer packaged goods
CSR	corporate social responsibility
CTE	critical tracking event
EAN	European Article Number
EDI	electronic data interchange
EPC	Electronic product code
EPCIS	Electronic Product Code Information Services
ERP	enterprise resource planning (software)
GDST	Global Dialogue on Seafood Traceability
GLN	Global Location Number
GTIN	Global Trade Item Number
GTS	Global Traceability Standard
GTS2	Global Traceability Standard 2.0
GUID	Globally Unique Identifier
ILMD	instance or lot master data
IoT	internet of things
ISO	International Organization for Standardization
IUU	illegal, unreported, and unregulated
KDE	key data element
LG TIN	Lot Global Trade Item Number
MES/MRP	manufacturing execution software/ manufacturing resource planning
MSC	Marine Stewardship Council
NFI	National Fisheries Institute
RFID	radio-frequency identification
SIMP	Seafood Import Monitoring Program
UPC	Universal Product Code
UUID	universally unique identifier
WG1, WG2	working group 1, working group 2



1 Traceability Data and Traceability Systems

The purpose of this section is to provide a holistic picture of traceability data and associated systems. Success of end-to-end traceability depends on several components and systems working in concert and spans unique identification, consistent and reproducible data collection, and information-sharing practices. Companies, to be compliant with regulations and accomplish business needs, must have some level of record keeping and data sharing with supply chain partners. The implementation guidance provides technical context to the **Core Normative Standards** and **Explanatory Materials** to aid IT and supply chain teams.

1.1 Traceability Data within an Organization

Specific companies or enterprises use internal traceability to meet many business needs (e.g., regulatory compliance, inventory management, recalls). To extend internal traceability to meet the business need of interoperable, end-to-end supply chain traceability, the steps for businesses across a supply chain include:

- All companies must benchmark their existing systems to ensure that all required KDEs and CTEs for their business processes are stored in their internal traceability systems.
- Vessel operators and farms must capture harvest, land, and first-sale information in a shareable format.
- Intermediate companies in the supply chain, must capture receipts, inputs, processing, outputs, shipments, and any waste associated with processing or storage.
- End point companies must capture receipts, consumption by consumers, and disposal of unsold product or other waste.
- All points must capture change of ownership and quantities at each step under their management for mass balance reporting along with inspections and certifications.

Traceability Data across the Seafood Supply Chain

Connecting an internal traceability system to up and downstream systems is somewhat more complicated. The following are the key tasks:

- Implement data-capture processes to ensure that necessary KDEs from each CTE is captured by your system.
- Implement the ability to send and receive traceability data in the GDST-required format in section 2 of this document.
- Implement the communication protocols in section 2 of this document.
- Test implementations with the GDST Capability Test.

Managing Traceability Data

Traceability Data Types and Sources

EPCIS has an events-based approach to structuring data and therefore has several data types based on their behavior and role in supply chain steps. These are described below:

- **Static master data:** Infrequently changed data that describes locations, products, parties, and assets, e.g., “Address,” and “Species Code and Name.”
- **ILMD:** Data that varies over different instances of production and is associated with either a specific serialized item (I) or lot (L), e.g., “First Freeze Date” and “Catch Area.” This information is most frequently provided in human-readable format on the product case or pallet label and must be entered upon receipt by the buyer. The same types of systems that capture this data also capture visibility event data (below).
- **Transaction data:** Data related to a business transaction, such as the completion of a transfer of ownership (purchase and sales orders, invoices) or a transfer of custody (proof of delivery, advanced shipment notice). This is commonly accomplished through EDI and AIDC.
- **Visibility event data:** Visibility events are usually captured by existing business systems – warehouse and/or accounting software, MES, ERP software, and on-vessel systems for e-logbook and vessel monitoring systems. Increasingly, organizations are investing in fit-for-purpose traceability systems and applications to support their consumer visibility efforts or to meet retailer demands.

In the ***Core Normative Standards***, KDEs are mapped to which are master, ILMD, or event-level data. The recommended approach is to share both master and event-level data governing the pedigree of the given batches/lots.

1.3.2

Sensitivity and Data Security

The GDST standards and extension to EPCIS are scoped primarily to mapping attributes and requiring the traceability information capture for addressing the IUU use case in seafood traceability. To ascertain legal catch, much of the data required may be considered sensitive by entities in the supply chain. Therefore, the GDST is not standardizing level of transparency or visibility by supply chain actors, but the recommendation is that traceability data sharing covered in this document be point to point and linked to a contract, agreement, or clear terms of usage. In other words, when data is shared, both sender and recipient have a clear and full understanding of the rules for how data can be used and shared downstream.

1.3.3

Data Sharing

Data sharing is a critical component of full supply chain traceability. The GDST standardizes data formatting and data communication protocols to enable interoperable data exchange between trading partners. The discussion and requirements on data sharing are in **Section 2**. Appendix 1 details information on data-sharing choreographies.



Quality and Verification

The quality of data is critical for supporting the goals of seafood traceability and interoperability. Therefore, a group within the GDST is developing specific guidance on how to ensure the veracity of the data collected and shared. The GDST anticipates that methods of verification and best practices for data collection, such as correlating KDEs with external data sets or remote sensing data, will grow with the implementation of the standard. However, the specifications for mitigating the risk of fraudulent data are beyond the scope of the standards.

The following are some basic criteria to consider as part of your systems design:

- **Completeness:** Are all the CTEs and KDEs captured?
- **Accuracy:** Is the recorded data accurately reflecting what happened?
- **Consistency:** Is the data aligned across systems?
- **Trusted source:** Are the shared events identified with source and digital signature?

Seafood Traceability Systems

In sum, a seafood traceability system requires the following components:

- Identification, marking, and attribution of traceable objects, parties, and locations.
- Automatic capture (through a scan or read) of the movements or events involving an object.
- Recording and sharing the traceability data, either internally or with parties in a supply chain, so that visibility into what has occurred may be realized.

The scope of the traceability system of a party will depend on the role of the party and the traceability questions that need to be addressed. Some elements that define the scope of a traceability system are:

- How many tiers up and down your supply chain will you need to share data?
- Will you need to interact with only direct supply chain partners, or will your system require a broader scope?
- Will you track main ingredients only or also packaging and indirect materials?
- Will your system need to integrate data sharing with final consumers/end customers?

2

Interoperability Data and Traceability Systems

To facilitate the exchange of traceability data we need to exchange event data and master data. This shall occur using an EPCIS Query Interface and a GS1 Digital Link Resolver.

***NOTE: For the purpose of this document, the term EPC will refer to either an SSCC, License Plate Number, GTIN + Lot Number, or a GTIN + Serial Number.*

***NOTE: This section of the document assumes that the reader is informed about the GS1 EPCIS 2.0 and GS1 Digital Link URI Syntax 1.3 standards released by GS1.*

2.1

Species Notes

Before continuing, it is important that you review each note below to assist you in understanding the remainder of this section on interoperability.

2.1.1

Electronic Product Code (EPC)

For the purpose of this document, the term EPC will refer to either:

- GTIN + Lot Number
- GTIN + Serial Number
- Logistics Identifiers
 - SSCC
 - License Plate Number

While in the context of GS1 as a whole, the EPC can identify other objects, for the purpose of this document it is important to restrict its definition.



2.1.2

GLN and PGLN

Throughout this document we will discuss two identifier types that have some ambiguous meanings throughout the GS1 standards. For the purpose of this document, we would like to clarify what we mean by them:

- **GLN** – This refers to a Global Location Number and is used to identify a physical location.
- **PGLN** – This refers to a Party Global Location Number and is used to identify a business or organization.

2.1.3

REST

All requests between two parties exchanging traceability data will be done using HTTP requests using the REST style and the expected EPCIS 2.0 format is the JSON-LD format.

2.1.4

Understanding EPCIS

Before continuing it is critical that you have foundational knowledge about EPCIS, the CBV, and the Digital Link standard. We highly recommend that you read each standard and understand them.

- **EPCIS 2.0** – While it's important to understand the standard as an entirety, we recommend focusing on sections 7, 8, 10, 12.
- **EPCIS 2.0 CBV** – Again, important to read the full document, however focusing on sections 6, 7, and 8 is a good start.
- **GS1 Digital Link 1.3** – Focus on sections 5, 6, 7 and 8.

You can also find [additional EPCIS materials here](#).



B2B Workflow

For the purpose of this workflow, we will designate the party requesting the traceability data as the Receiver and the party transferring the traceability data as the Sender. You can think of the Receiver as the party receiving goods from the Sender selling/transferring the goods to the Receiver.



This first workflow illustrates how a Receiver obtains traceability data, a set of EPCIS events and master data, through a request and response exchange with an external system of the Sender. For the purpose of this workflow, we will assume that the Receiver (Buyer) has received a list of EPC(s) through another workflow and is now attempting to resolve traceability data and master data for these EPC(s).

****NOTE: An example of this previous workflow might be the Receiver (Buyer) receiving an Advanced Shipment Notification from the Sender (Seller) through their ERP system.**

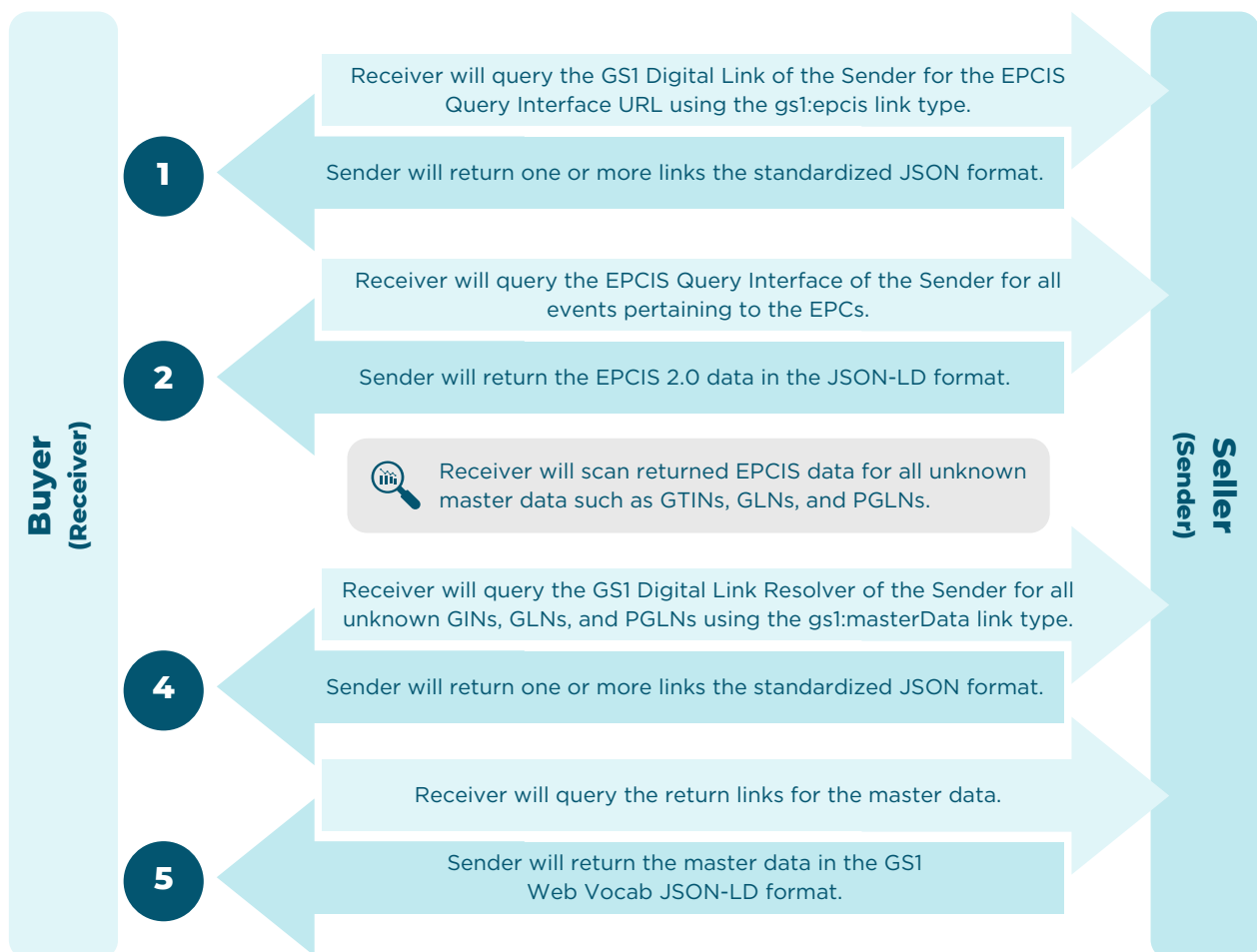


Figure 1 - B2B Workflow Diagram



Once the Receiver has a list of EPCs that they want to request traceability data for, the following workflow will take place:

1. The Receiver will format one of the EPCs as a GS1 Digital Link URI in order to query the Sender's GS1 Digital Link Resolver in order to discover the location of the EPCIS repository of the sender. This request will return a link to an EPCIS repository associated with the sender and the EPC. The GS1 Digital Link URI shall be appended with the `gs1:epcis` link type in order to express to the Sender's resolver that the Receiver wishes to locate an EPCIS repository.
 - a. They will use the link type `gs1:epcis` to signal they are looking for the EPCIS Query Interface that contains the event data for that EPC.
 - b. Alternately, the Receiver may request the location of an EPCIS repository with a GS1 Digital Link URI with a PGLN identifying the Sender as an entity through use of the GS1 Application Identifier for a PGLN identifying a party, 417.
 - c. NOTE - this step of the workflow is only necessary when the Receiver does not know the location of the EPCIS repository of the Sender. Subsequent exchanges can query the EPCIS repository directly without formatting an EPC as a GS1 Digital Link URI Syntax.
2. The Receiver will query the EPCIS Query Interface for each EPC they wish to receive traceability data for. These queries will be returned from the Sender as one or more events formatted in an EPCIS 2.0 Query Document in the JSON-LD format.
3. The *Receiver* will scan the received EPCIS event data for all distinct GTINs, GLNs, and PGLNs in the event data.
4. The Receiver will then query the Sender's GS1 Digital Link Resolver for each GTIN, GLN, and PGLN that is unknown to the Receiver. They can also choose to always query for the master data to ensure it is up to date.
 - a. The Receiver will use the link type **`gs1:masterData`** when querying the GS1 Digital Link Resolver to express to the Sender that only links or content are returned pertaining to master data.
5. The Receiver will resolve each link and will expect the links to return the master data in the GS1 Web Vocab JSON-LD.



2.2

Internal System to System (IS2S) Workflow

It may be necessary for a single company to move traceability data from one traceability system to another traceability system. This could be for the following reasons:

- The company is a large enterprise which operates separate software systems for different business units of the company.
- Data is being recorded by or on behalf of the company in a secondary traceability system.
- The company wishes to move from one traceability solution to another.



When this occurs, a different workflow is required than that of the B2B workflow.

In this workflow, the company would define a period of time that traceability data was recorded into one system and sync that data into the second system, in which they want to sync traceability data from one system that was recorded during that period of time into a second system.

For the purpose of this workflow, we will designate the Sender as the first system where the traceability data is stored. And we will designate the Receiver as the second system in which we want to move the traceability data to.

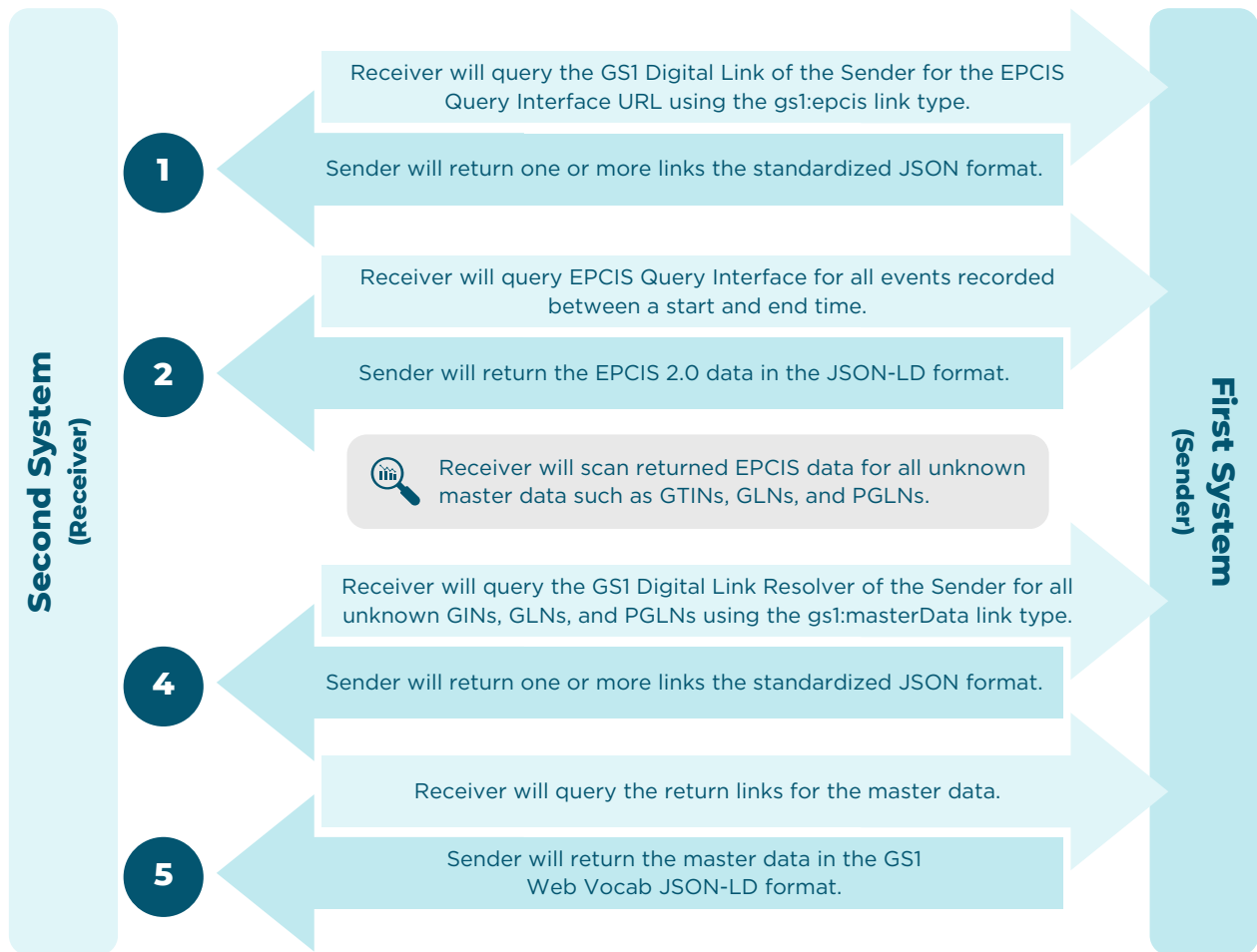


Figure 2 - IS2S Workflow Diagram

Once that period of time is defined, it is important to break the period of time into smaller slices of time such as a day, or even as small as a 1 hour window.

For each slice of time the following steps would be followed:

1. The *Receiver* will query the Digital Link URL with the link type **gs1:epcis** to obtain the EPCIS URL of the first system.
 - a. This can be done by querying the base URL of the GS1 Digital Link with the Link Type like so: `{GS1_DIGITAL_LINK_BASE_URL}/417/{PGLN}?linkType=gs1:epcis`

or

https://www.example.org/417/urn:gdst:example.org:party:123.0/digital_link?linkType=gs1:epcis

2. The Receiver will then query the EPCIS API method /queries/SimpleEventQuery for all events that were recorded between the start of the slice of time and the end of the slice of time.
3. The Receiver will scan the received EPCIS event data for all GTINs, GLNs, and Party GLNs in the event data.
4. The Receiver will then query the Sender's GS1 Digital Link Resolver for each GTIN, GLN, and Party GLN that is unknown to the Receiver.
 - a. The Receiver will use the link type **gs1:masterData** when querying the GS1 Digital Link Resolver to limit the links returned to those pertaining to master data.
5. The Receiver will resolve each link and will expect the links to return the master data in the GS1 Web Vocab JSON-LD.

2.3

Authentication

The two parties exchanging data will need to exchange API Keys that can be used to access each other's EPCIS Query Interface. Each request from the Receiver to the Sender will contain an HTTP Header X-API-Key that will contain the API Key.

X-API-Key 81C6C194-B445-4FFB-907E-CDBC7FF7F36F
<https://resolve.example.org/party/0086003130308?linkType=gs1:masterData>

Cases for not requiring Authentication can include but are not limited to:

- A retailer or brand owner may wish to expose traceability data publicly about their products to consumers. They may choose to share the complete data or a redacted form of the data.
- A brand owner may choose to expose their GS1 Digital Link Resolver to share information about their trade items such as coupons, videos, additional information, or recipes.

***NOTE: It is completely up to the Sender to manage, generate, and authenticate the API Keys. The only thing this standard specifies is where to put and look for the API Key if authentication is to be used.*

***NOTE: Authentication is optional within the EPCIS standard from GS1. It is not required within that specification but expected to occur. At a minimum, implementations following this standard are required to support the X-API-Key header for authentication. If two solution providers support other security protocols and desire to use that security protocol rather than an API Key, then they can opt to use that security protocol instead.*

Adding a Trading Party

Before companies start exchanging traceability data, they exchange company info to establish the other party in their respective systems. In addition to data about the company, the trading partners may exchange foundational info about the locations where exchanges of products will occur (e.g., processing plant, distribution center). This exchange can be accomplished simply by each company exchanging a Digital Link Resolver URI and optionally an API Key. This enables companies to request the master data describing a party or location instead of for an object like demonstrated in the previous flows.

We propose that the following information is communicated to a trading party to grant them access to request data.

- GS1 Digital Link URL+ Party GLN
- API Key

Example:

<https://resolver.trace-solution-example.com/417/086000313030881C6C194-B445-4FFB-907E-CDBC7FF7F36F>

The example GS1 Digital Link URL shown above contains two parts: the Base URL highlighted in red, and the Party GLN highlighted in blue.

The Base URL is portion until the first application identifier is found. In the example, the first application identifier is '417' for a Party Global Location Number. Then the value of '0860003130308' following the '417' is the Party GLN of the Trading Partner.

It is possible to query the GS1 Digital Link Resolver URL + PGLN with the link type `gs1:masterData` to get information about the trading party.

It is also possible to query the GS1 Digital Link Resolver URL + PGLN with the link type `gs1:epcis` to get the default EPCIS Query Interface URL of the Trading Party.

EPCIS Data Format

GDST will use the JSON-LD format from EPCIS 2.0 for communicating traceability events. The standard and additional resources can be found [here](#). On top of the additional standard we will make the following requirements:

- We will require that within the JSON-LD file, that the `@context` of the JSON-LD references the officially hosted EPCIS JSON-LD Context at <https://ref.gs1.org/standards/epcis/epcis-context.jsonld>, as typically expected by the EPCIS 2.0 standard.

“Typically, it may reference the standard JSON-LD context resource for EPCIS 2.0, which is published at <https://ref.gs1.org/standards/epcis/2.0.0/epcis-context.jsonld>.” - [EPCIS 2.0 Standard](#) - Section 10.1.2

- We will also require that the JSON-LD file is communicated in the compact format, which is recommended by the EPCIS 2.0 standard for the following reason:

*“What is meant by JSON/JSON-LD is that as far as possible, most annotations that are specific to JSON-LD are hidden from the body of the data payload by placing them within the JSON-LD context resource and through the use of aliases. This means that if the software consuming EPCIS 2.0 data only expects to treat it as JSON data, it can simply ignore the JSON-LD context resource and use existing JSON processing functions for parsing the body of the data payload. Software that specifically needs EPCIS 2.0 as a JSON-LD or Linked Data format makes use of the JSON-LD context resource (including retrieval of any referenced online context files/resources) to obtain a full JSON-LD document / data structure, which can also be easily translated into other Linked Data formats using available translation tools.” – **EPCIS 2.0 Standard** – Section 10.1.2*

- We will require that all JSON-LD context information is recorded in the root @context of the body of the JSON-LD file as recommended by EPCIS 2.0 Standard:

*“Even though each individual member object in a JSON-LD document MAY declare its local @context, EPCIS 2.0 capturing applications that use the JSONLD context resource SHOULD specify a consolidated @context for all default and user-defined namespaces at the root level of the EPCISDocument (refers to the payload of POST /capture endpoint) as well as at the root level of the EPCISEvent (refers to POST /event endpoint). This is strongly recommended, in order to simplify document parsing and validation”. – **EPCIS Standard** – Section 10.1.3*

2.4.1

Example EPCIS JSON-LD Files

Additional information and examples of GDST event data in the JSON-LD format can be found in our [developer documentation](#).

2.5

Master Data

Master data SHALL be resolved using the Digital Link Resolver. So that when an EPCIS Query Document is returned from the EPCIS Query Interface, the following steps must be run in order to ensure master data has been resolved:

1. Scan the EPCIS Query Document for all distinct master data identifiers.
 - a. GTINs, GLNs, and PGLNs.
2. The Receiver SHOULD Filter the list of distinct identifiers down to only the master data identifiers which are unknown.
 - a. The Receiver can choose to always resolve master data identifiers whether they are known to them or not.
3. Query the Digital Link Resolver of the party you are requesting traceability information from for each identifier:
 - a. GTIN
 - b. GLN
 - c. Party GLN

4. The party behind the Digital Link Resolver will return the master data associated with the identifiers in the GS1 Web Vocab JSON-LD format.

**** NOTE: EPCIS 2.0 no longer allows master data to be in an EPCIS Query Document for JSON-LD format. Therefore, we now require that all master data is resolved via the GS1 Digital Link Resolver as of GDST 1.2**

2.5.1

GS1 Web Vocab JSON-LD

For master data returned by the GS1 Digital Link Resolver, it should be returned in the GS1 Web Vocab JSON-LD format. More information about this format can be found at [Additional information and examples of GDST event data in the JSON-LD format can be found in our developer documentation.](#) Our [Developer documentation](#) contains examples of the master data KDEs in the GS1 Web Vocab JSON-LD format.



2.6

GS1 Digital Link

GS1 Digital Link as a standard is meant to serve two primary purposes for implementers of the GDST standards. The first is a way to encode the data typically encoded in a barcode with GS1 Application Identifiers as a link, creating resolvable URI identifiers for master data and EPCs. The second primary purpose is to create a “pseudo” DNS server to lookup links about information related to GTINs, GLNs, Party GLNs, EPCs, and more. For the purpose of this document, we will focus on the second primary purpose as using the GS1 Digital Link as a standard for looking up additional information around GTINs, GLNs, Party GLNs, and EPCs.

Requirements

The GS1 Digital Link contains many pieces and is quite an extensive standard to implement in its entirety. For the purpose of this standard, we will discuss which pieces from the GS1 Digital Link standard are required to be implemented.

The GS1 Digital Link resolver will support at a minimum the following link type:

- **gs1:epcis**
 - This will return a link that points to the EPCIS Query Interface for the requested EPC.
- **gs1:masterData**
 - This will return a link that can resolve the GS1 Web Vocab JSON-LD for the requested GTIN/GLN/Party GLN.

The following application identifiers for URL paths would be required:

- **01 - GTIN**
- **414 - GLN of a Physical Location**
- **417 - Party GLN**
- **21 - Serial Number**
- **10 - Lot Number**

**** Note: We have updated it so that the numeric application identifier must be used and that the alphabetic identifier is no longer allowed. This is to fall in line with the 1.3 update of the Digital Link standard from GS1. For example, you must use “01” before the GTIN in the URL, and not “gtin”.**

GDST Identifiers

GDST has introduced a type of identifier that does not require a GS1 company prefix to create. These identifiers are commonly referred to as “GDST Identifiers” and the URN registration for these identifiers can be found at:

<https://www.iana.org/assignments/urn-formal/gdst>

The GDST identifiers may populate an EPCIS event instead of identifiers built with GS1 identification keys (e.g., GTINs, GLNs, & SSCCs). They follow the pattern of Private or Industry-wide URNs for identifiers that are defined in GS1’s Core Business Vocabulary (sections 8.2.3, 8.3.3, 8.4.3, etc.). The URN format is a type of Uniform Resource Identifier which is designed to ensure an identifier is globally unique when referenced between more than one system.

When a company is requesting data through their trading partner’s Digital Link resolver, the GDST identifiers can fill the same role in the GS1 Digital Link URI syntax as a GTIN or other GS1 Identification Key:

<https://resolve.example.org/417/urn:gdst:example.org:party:ABC.12345>

https://resolver.example.org/01/urn:gdst:example.org:product:class:FARM1.white_shrimp

These identifiers are meant to only be constructed using URL compatible characters. This allows them to be easily inserted into a URL without any encoding. This is useful for inserting them into the URL when resolving with the Digital Link Resolver.

Digital Link URI Identifiers

EPCIS 2.0 has introduced digital link identifiers as additional permitted formats for representing identifiers such as a GTIN, GTIN+Lot, GTIN+Serial Number, GLN, or Party GLN. **The following are examples of the identifiers in the Digital Link Format:**

EPC Class (GTIN + LOT):

<https://id.gs1.org/01/00860003130308/10/L1234567890>

EPC Instance (GTIN + SERIAL NUMBER):

<https://id.gs1.org/01/00860003130308/21/S1234567890>

GTIN:

<https://id.gs1.org/01/00860003130308>

GLN:

<https://id.gs1.org/416/0860003130308>

Party GLN:

<https://id.gs1.org/417/0860003130308>

Per the recommendation of EPCIS 2.0, we will REQUIRE that all GS1 Digital Link identifiers use the canonical resolver domain “<https://id.gs1.org>”, meaning that the part of the URL prior to the first application identifier must be <https://id.gs1.org>. This only applies when including the identifiers within an EPCIS event of traceability data. When using the identifiers to request data from a trading partner’s Digital Link resolver it is necessary to use the domain of the trading partner.

We do recommend that Digital Link URI identifiers are used instead of the traditional URN format since the Digital Link URI does not require placing a “.” between the company prefix and the item/location reference of the GTIN/GLN/Party GLN.



Resolving Digital Link URI Identifiers through a Digital Link Resolver

When choosing to access a Digital Link Resolver of the Sender with a Digital Link URI Identifier, you simply need to replace the canonical base URL of the Digital Link URI Identifier with the Base URL of the Sender's Digital Link Resolver, so that:

If the Sender's Digital Link Resolver Base URL is <https://resolver.example.org/> and we want to resolve the master data for the PGLN <https://id.gs1.org/417/0860003130308>.

We will transform it into the URL:

<https://resolver.example.org/417/0860003130308>

GS1 Digital Link URL Structure

While the GS1 Digital Link standard discusses this topic in more detail, this section is meant to give a brief overview of the structure of a GS1 Digital Link URL.

These are constructed like so:

{Base URL}/{Application Identifier}/{Value 1}/{Application Identifier 2}/{Value 2}

<https://resolver.example.org/01/00860003130308/10/LOT07082021>

The Application Identifiers need to be placed in a specific order. For the purposes of this standard, the following Application Identifier pairs should be supported:

- **417**
 - *This is meant to return links about a specific Trading Party or Organization.*
 - **EXAMPLE:** <https://resolver.example.org/party/0860003130308>
 - You can query this Application Identifier (PGLN) with either the **gs1:masterData** link type to get a link to the master data or with the **gs1:epcis** to get the default EPCIS Query Interface for the organization.

- **414**
 - *This is meant to return links about a specific Location.*
 - **EXAMPLE:** <https://resolver.example.org/gln/urn:gdst:example.org:location:loc:FARM1.0>
 - You can query this Application Identifier (PGLN) with either the **gs1:masterData** link type to get a link to the master data or with the **gs1:epcis** to get the default EPCIS Query Interface for the organization.

- **01**
 - *This is meant to return links about a specific Trade Item. For the purpose of this standard, it is only expected to return a link for master data. Represented by link type **gs1:masterData**.*
 - **EXAMPLE:** <https://resolver.example.org/gtin/00860003130308>

- **01 / 10**
 - *This is meant to return links about a specific batch/lot of a trade item. For the purpose of this standard, the links will only be required to return a link to an EPCIS Query Interface. Represented by the link type `gs1:epcis`.*
 - **EXAMPLE:** <https://resolver.example.org/gtin/00860003130308/lot/LOT07082021>
- **01 / 21**
 - *This is meant to return links about an individual instance of a trade item. For the purpose of this standard, the links will only be required to return a link to an EPCIS Query Interface. Represented by the link type `gs1:epcis`.*
 - **EXAMPLE:** <https://resolver.example.org/gtin/00860003130308/ser/ABCD-EFGH-1234-5678>

Constructing Digital Link URL from GDST EPCs

The EPCs that are a GTIN + Lot Number or GTIN + Serial Number are constructed into a Digital Link URL differently than others. For the GS1 EPC, the GTIN and the Lot Number are split into two application identifiers, as shown above. The same goes for the GTIN + Serial Number.

In order to construct a Digital Link URL from a GDST EPC, you will follow the same pattern as constructing one from a GS1 EPC URN such as GTIN + Lot Number or GTIN + Serial Number.

So that if we have the Digital Link Resolver Base URL <https://resolver.example.org> and we would like to query for the EPCIS Query Interface of the following GDST EPC (GTIN + Lot Number):

`urn:gdst:example.org:product:lot:class:c1.i1.lot123`

We will transform it into this URL:

<https://resolver.example.org/01/urn:gdst:example.org:product:class:c1.i1/10/lot123>

Please note how the “product:lot:class” part in the URN changed to “product:class”.

Next, if we have the GDST EPC (GTIN + SERIAL NUMBER):

`urn:gdst:example.org:serial:obj:c1.i1.serial456`

We will transform it into this URL:

<https://resolver.example.org/01/urn:gdst:example.org:product:class:c1.i1/21/serial456>

Please note how the “serial:obj” changed to “product:class”.

2.6.1

Authentication

If a link returned from the GS1 Digital Link resolver requires authentication to access, then it shall carry the attribute **“authRequired”** with the value set to **“true”**. If this attribute is set to TRUE, then the receiver who wants to resolve the link will need to apply the same authentication information as they did when they requested the link from the GS1 Digital Link resolver.

2.6.2

Open API Definition

The Open API Definition for EPCIS 2.0 can be found at <https://ref.gs1.org/standards/epcis/openapi.json>

2.6.3

Example GS1 Digital Link Request #1

The following is an example response for an HTTP request made against a GS1 Digital Link Resolver with a GTIN + Lot Number and looking for an EPCIS Query Interface.

URL

{GS1_DIGITAL_LINK_BASE_URL}/01/urn:gdst:product:class:ABC.123/10/EFG-456?
linkType=gs1:epcis

HTTP Verb

GET

REQUEST HEADERS

NAME	VALUE	REQUIRED
X-API-Key	{API KEY}	
Content-Type	application/json	X
Accept	application/json	X

REQUEST BODY

No body provided because this is a GET request.

RESPONSE BODY

```
[{
  "link": "https://example.org/EPCIS",
  "title": "EPCIS Query Interface",
  "linkType": "gs1:epcis",
  "authRequired": "true"
}]
```

RESPONSE HEADERS

NAME	VALUE
Content-Type	application/json

ERROR RESPONSES

401 - Unauthorized Request

404 - Resource Not Found

500 - Unexplained error or internal error.

2.6.3

Example GS1 Digital Link Request #2

The following is an example response for an HTTP request made against a GS1 Digital Link Resolver with a GTIN and looking for master data about the product identified by the GTIN.

URL

```
{GS1_DIGITAL_LINK_BASE_URL}/01/urn:gdst:product:class:ABC.123?
linkType=gs1:masterData
```

HTTP Verb

GET

REQUEST HEADERS

NAME	VALUE
X-API-Key	{API KEY}
Content-Type	application/json
Accept	application/jso

BODY

No body provided because this is a GET request.

RESPONSE BODY

```
[{  
  "link": "https://example.org/api/products/ABC/123",  
  "title": "Product Definition",  
  "linkType": "gs1:masterData",  
  "authRequired": "true"  
}]
```

RESPONSE HEADERS

NAME	VALUE
Content-Type	application/json

ERROR RESPONSES

401 - Unauthorized Request

404 - Resource Not Found

500 - Unexplained error or internal error.



EPCIS Query Interface

The EPCIS standard defines communication protocols for querying and capturing EPCIS events within and between companies. The new EPCIS 2.0 release offers a new communication protocol for the EPCIS Query Interface that follows REST interface principles and is defined concretely with an OpenAPI definition file.

Implementations following the GDST standard are expected to implement the EPCIS Query Interface as a REST interface defined by the EPCIS 2.0 standard.

Requirements

The entire EPCIS Query Interface is extensive., To simplify adoption of interoperable seafood traceability, this standard only requires the following components be implemented from the EPCIS Query Interface:

The following REST API methods SHALL be implemented:

- **GET - /events**
 - Used to request all of the events in the repository or a set of events matching the criteria included as query parameters in the requesting URL.

The EPCIS Query Interface also needs to support the following query parameters in a named query or as parameters in the requesting URL:

- **LT_recordTime - ISO DateTime Stamp**
 - This is a date time in the UTC format, following the xsd:dateTimeStamp data type. This will return all events where the recordTime is less than the provided value.
- **GE_recordTime - ISO DateTime Stamp**
 - This is a date time in the UTC format, following the xsd:dateTimeStamp data type. This will return all events where the recordTime is either greater than or equal to the provided value.
- **LT_eventTime - ISO DateTime Stamp**
 - This is a date time in the UTC format, following the xsd:dateTimeStamp data type. This will return all events where the eventTime is less than the provided value.

- **GE_eventTime – ISO DateTime Stamp**
 - This is a date time in the UTC format, following the xsd:dateTimeStamp data type. This will return all events where the eventTime is either greater than or equal to the provided value.
 - MATCH_anyEPC – List of URIs If this parameter is specified, the result will only include events that (a) have an epcList field, a childEPCs field, a parentID field, an inputEPCList field, or an outputEPCList field (that is, ObjectEvent, AggregationEvent, TransactionEvent, TransformationEvent, AssociationEvent or extension event types that extend one of those event types); and where (b) the parentID field or one of the EPCs listed in the epcList, childEPCs, inputEPCList, or outputEPCList field (depending on event type) matches one of URIs specified in this parameter. The meaning of “matches” is as specified in section 8.2.7.1.1 of the EPCIS 2.0 standard.
- **MATCH_anyEPCClass – List of URIs**
 - If this parameter is specified, the result will only include events that (a) have a quantityList, childQuantityList, inputQuantityList, or outputQuantityList field (that is, ObjectEvent, AggregationEvent, TransactionEvent, TransformationEvent, AssociationEvent or extension event types that extend one of those event types); and where (b) one of the EPC classes listed in any of those fields matches one of the EPC patterns or URIs specified in this parameter. The meaning of “matches” is as specified in section 8.2.7.1.1 of the EPCIS 2.0 standard.
- **EQ_bizStep – List of URIs**
- **EQ_transformationID – List of URIs**
 - We added this for doing tracebacks on long running transformations.

If specified, the result will only include events that (a) have a non-null bizStep field; and where (b) the value of the bizStep field matches one of the specified values. If this parameter is omitted, events are returned regardless of the value of the bizStep field or whether the bizStep field exists at all.** NOTE: It is always possible to return a “Query Too Complex” (413) if the filters provided are too broad or will return too many results. For example, if only the eventTypes filter parameter is provided, this may return too many or be deemed too broad to execute.

The only required search patterns are:

- **MATCH_anyEPC and/or MATCH_anyEPCClass**
- **EQ_bizStep + GE_eventTime + LT_eventTime**
 - *Can return a Query Too Complex (413) error if the period of time specified is too large.*
- **EQ_bizStep + GE_recordTime + LT_recordTime**
 - *Can return a Query Too Complex (413) error if the period of time specified is too large.*

Open API Definition

A minimized version of this Open API configuration file can be found at:

<https://app.swaggerhub.com/apis/DuckScapePhilip/GDST-EPCIS/1.0.0>

Example HTTP Request #1

This example shows how to query for all receiving events that occurred between Nov. 1st 2021 and Dec. 1st 2021.

URL

`{EPCIS_BASE_URL}/events?GE_recordTime=2021-11-01T00:00:00Z&LE_recordTime=2021-12-01T00:00:00Z&EQ_bizStep=receiving`

HTTP Verb

POST

REQUEST HEADERS

NAME	VALUE
X-API-Key	{API KEY}
Content-Type	application/json
Accept	application/json
GS1-EPCIS-Version	2.0
CSI-CBV-Version	2.0
GS1-Extension	{ "gdst" : " https://traceability-dialogue.org/epcis". }

NOTE: *It is expected that the sender (server) will validate the GS1-EPCIS-Version, and GS1-CBV-Version. The EPCIS 2.0 standard enables interfaces to exchange EPCIS data following different versions of the standard, the GDST standard requires implementations to only use EPCIS 2.0 and declare the value "2.0" across those 2 headers.*

BODY

There is no body.

RESPONSE HEADERS

NAME	VALUE
Content-Type	application/json
GSI-EPCIS-Version	2.0
GSI-CBV-Version	2.0
GSI-Extensions	{ "gdst" : " https://traceability-dialogue.org/epcis" }
Location	{EPCIS_BASE_URL}/events

NOTE: *It is expected that the receiver (client) will validate the GSI-EPCIS-Version, and GSI-CBV-Version. The EPCIS 2.0 standard enables interfaces to exchange EPCIS data following different versions of the standard, the GDST standard requires implementations to only use EPCIS 2.0 and declare the value "2.0" across those 2 headers.*

RESPONSE BODY

The sender (server) will return an EPCIS Query Document containing the collection of events matching the criteria of the named query. The EPCIS Query Document needs to follow EPCIS 2.0 and be formatted in JSON-LD.

ERROR RESPONSES

- 401 - Unauthorized Request
- 403 - Client Unauthorized
- 404 - Resource Not Found
- 406 - Not Acceptable
- 413 - Query would return too many results
- 500 - Unexplained error or internal error.



2.7.3

Example HTTP Request #2

This example will demonstrate how you can query the /events end point for one or more EPCs and the business step shipping and receiving. In this example we have included the business steps in the URI format and the CURIE format. We have also included EPCs in the Digital Link URI format as well.

***NOTE: For this example, we are not URL encoding the query parameters to make it more readable. However, you should URL encode the query parameters to make sure they are communicated correctly in a URL.*

URL

```
{EPCIS_BASE_URL}/eventsMATCH_epc=https://id.gs1.org/01/00614141777778/10/987&MATCH_epcClass=urn:epc:class:lgtn:4012345.012345.998877&MATCH_anyEPC=https://id.gs1.org/01/00614141777778/10/987|https://id.gs1.org/01/00614141777778/10/987&MATCH_anyEPCClass=urn:epc:class:lgtn:4012345.012345.998877|urn:epc:class:lgtn:4012345.012345.998877&EQ_bizStep=https://ref.gs1.org/cbv/BizStep-shipping|receiving
```

In this example the receiver uses a pipe delimiter to separate each EPC value apart of this query.

HTTP Verb

GET

REQUEST HEADERS



NAME	VALUE
X-API-Key	{API KEY}
Content-Type	application/json
Accept	application/json
GSI-EPCIS-Min	2.0
GSI-EPCIS-Max	2.0
GSI-CBV-Min	2.0
GSI-CBV-Max	2.0
GSI-Extensions	{ "gdst" : " https://traceability-dialogue.org/epcis" }

NOTE: *It is expected that the sender (server) will validate the GSI-EPCIS-Min, GSI-EPCIS-Max, GSI-CBV-Min, GSI-CBV-Max, and GSI-EPC-Format. The EPCIS 2.0 standard enables interfaces to exchange EPCIS data following different versions of the standard, the GDST standard requires implementations to only use EPCIS 2.0 and declare the value "2.0" across the 4 headers expressing the version of EPCIS and CBV.*

RESPONSE HEADERS

NAME	VALUE
Content-Type	application/json
GSI-EPCIS-Version	2.0
GSI-CBV-Version	2.0
GSI-Extensions	{ "gdst" : " https://traceability-dialogue.org/epcis" }

NOTE: *It is expected that the receiver (client) will validate the GSI-EPCIS-Version, and GSI-CBV-Version. The EPCIS 2.0 standard enables interfaces to exchange EPCIS data following different versions of the standard, the GDST standard requires implementations to only use EPCIS 2.0 and declare the value "2.0" across those 2 headers.*

RESPONSE BODY

The sender (server) will return an EPCIS Query Document containing the collection of events matching the criteria of the named query. The EPCIS Query Document needs to follow EPCIS 2.0 and be formatted in JSON-LD.

ERROR RESPONSES

- 401 - Unauthorized Request
- 403 - Client Unauthorized
- 404 - Resource Not Found
- 406 - Not Acceptable
- 414 - URL is too long
- 500 - Unexplained error on the implementation



3 Critical Tracking Event Examples – Wild-Caught Tuna to Canned Tuna

The following is a walk-through of a typical wild-caught tuna supply chain beginning with a vessel through loin processing to a cannery. Rather than show generalized usage of the EPCIS event visibility standard, we have elected to use real scenarios. The EPCIS standard is very flexible, so it can be used to model many different workflows. Therefore, the downstream systems should also be designed in a flexible way.

EPCIS event files



(01)10614141123459(11)170709(10)123456

Frozen Tuna Loins

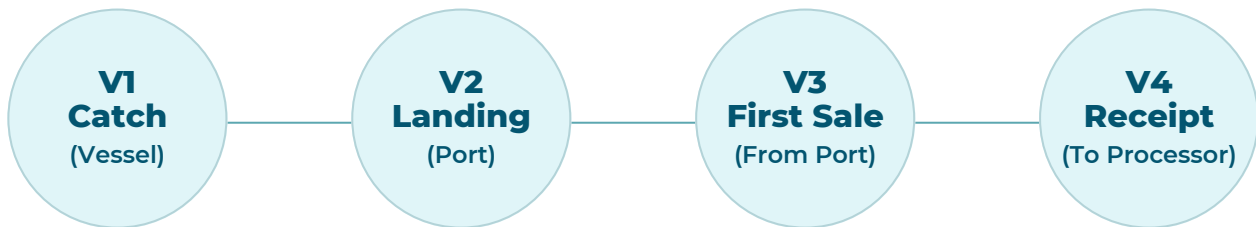
500 KG

Product of Taiwan
Seafood Company
Taipei, Taiwan

Catch Date:

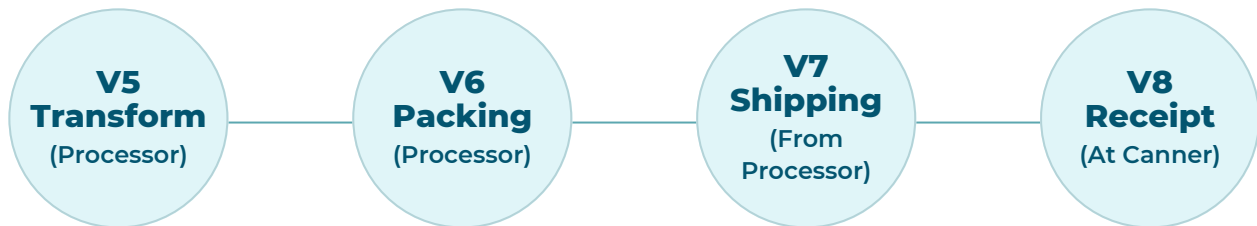
July 09, 2017

Image Source: [Products - Loins - Tri Marine Group](#)

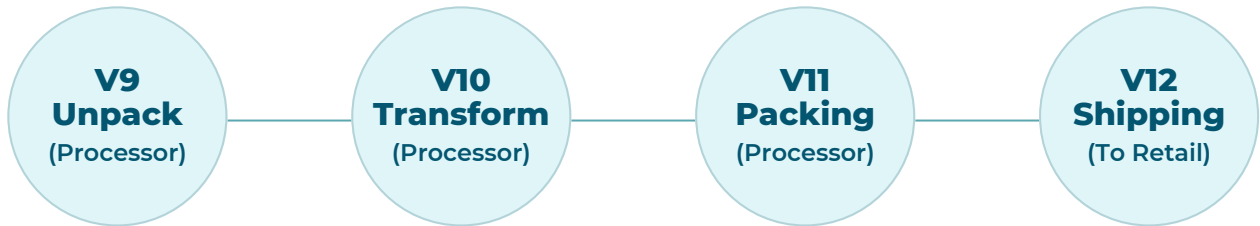


Dimension	V1	V2	V3	V4
Why	ObjectEvent ADD, Wild Harvest	ObjectEvent OBSERVE, Landing	ObjectEvent OBSERVE, Transporting	ObjectEvent OBSERVE, Receiving
Who	Vessel Operator	Vessel Operator	Vessel Operator	Processor
What	Traceable Object (Tuna) Quantity, UOM	Traceable Object (Tuna) Quantity, UOM	Traceable Object (Tuna) Quantity, UOM	Traceable Object (Tuna) Quantity, UOM
Where	Catch Area, Vessel ID	Port of Landing	Source: Port Destination: Processor	Source: Port Destination: Processor
When	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone

Dimension	V1	V2	V3	V4
Master Data Header	Vessel Operator/Owner Vessel Identifier Vessel Registration Public Vessel Registry Hyperlink Vessel Flag Catch Identifier Availability of Catch Coordinates Satellite Vessel Tracking Authority	Port Information		Customer Contact Info "Ship To" Address
ILMD	Catch Area Species Economic Zone Fishing Gear Type Production Method Harvest Start/End Certification List			
Tech Info	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event



Dimension	V5	V6	V7	V8
Why	AggregationEvent ADD	AggregationEvent ADD, Aggregation	ObjectEvent Shipping	ObjectEvent OBSERVE, Receiving
Who	Processor	Processor	Vessel Operator	Processor
What	Input: Whole Tuna Output: Frozen Loins	Parent: Container Children: Frozen Tuna Loin Cases	Container	Container
Where	Loin Processing Plant	Loin Processing Plant	Source: Loin Plant Destination: Canner	Source: Loin Plant Destination: Canner
When	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone
Master Data Header	Plant Operator/Owner Plant Identifier Frozen Tuna Loins ID		Customer Contact Info "Ship To" Address	Customer Contact Info "Ship To" Address
ILMD	Production Date Storage State First Freeze Date			
Tech Info	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event



Dimension	V9	V10	V11	V12
Why	AggregationEvent DELETE, Disaggregate	TransformationEvent ADD	AggregationEvent ADD, Aggregate	ObjectEvent OBSERVE, Shipping
Who	Processor	Processor	Processor	Processor
What	Parent: Container Children: {blank}	Input: Tuna Loins Output: Canned Tuna	Parent: Pallet SSCC Children: Cases of Canned Tuna	Pallet SSCC
Where	Canning Plant	Canning Plant	Canning Plant	Source: Canning Plant Destination: Customer?
When	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone
Master Data Header		Canner Operator/Owner Canning Plant Identifier Canned Tuna ID		Customer Contact Info "Ship To" Address
ILMD		Preservation Technique "Best Before" Date MSC Certification		
Tech Info	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event



4

Critical Tracking Event Examples – Aquaculture

The following is a walk-through of a typical aquaculture supply chain beginning with a farm through processing to a retailer. Rather than show generalized usage of the EPCIS event visibility standard, we have elected to use real scenarios. The EPCIS standard is very flexible, so it can be used to model many different workflows. Therefore, the downstream systems should also be designed in a flexible way.

EPCIS event files



Source: [food service direct](#)



(01)10614141123459(11)170709(10)123456

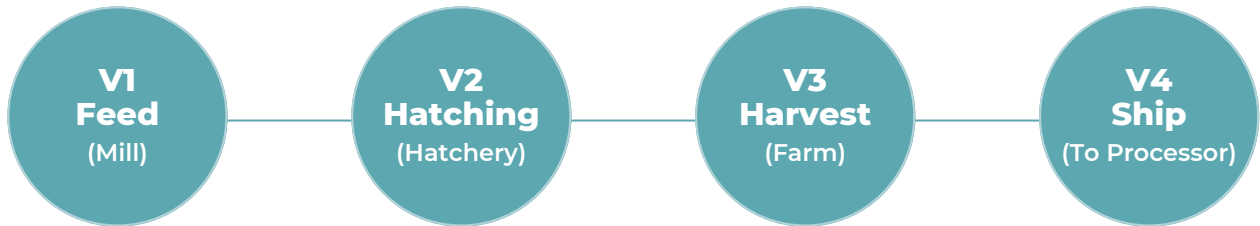
26-30 Raw Tail-On White Shrimp

2 LB x 5 per Case

Product of Thailand
Shrimp Farming Company
Bangkok, Thailand

Harvest Date:

July 09, 2017



Dimension	V1	V2	V3	V4
Why	TransformationEvent ADD	ObjectEvent ADD, Hatching	TransformationEvent ADD, Farm Harvest	ObjectEvent OBSERVE, Shipping
Who	Feed Mill	Farmer	Farmer	Farmer
What	Input: Ingredients Output: Feed ID	Fingerling ID	Input: Feed ID, Fingerling ID Output: Live Shrimp ID	Live Shrimp ID
Where	Feed Mill	Hatchery	Farm	Source: Farm Destination: Processor
When	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone
Master Data Header	Mill Owner Mill Identifier Feed Identifier	Hatchery Owner Hatchery Identifier Fingerling Identifier	Farm Owner Farm Identifier Live Shrimp Identifier	Processor ID and Address
ILMD	Source of Protein Certification List	Harvest Start/End Date Source of Fingerling Species	Farming Method Date of Harvest Species	
Tech Info	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event

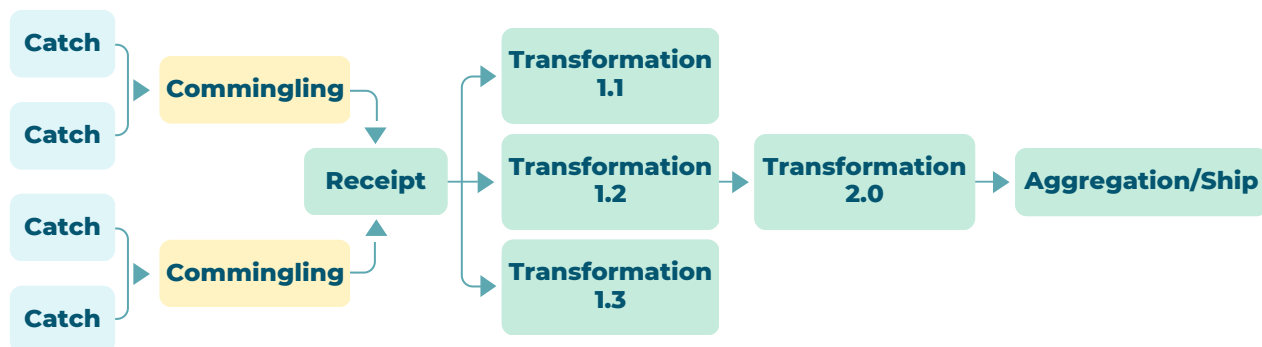


Dimension	V5	V6	V7	V8
Why	TransformationEvent	AggregationEvent ADD, Aggregation	ObjectEvent OBSERVE, Shipping	ObjectEvent OBSERVE, Receiving
Who	Processor	Processor	Processor	Retailer
What	Input: Live Shrimp ID Output: Frozen Shrimp	Parent: Pallet Children: Frozen Shrimp Cases	Pallet	Pallet
Where	Shrimp Processing Plant	Shrimp Processing Plant	Source: Shrimp Plant Destination: Retailer DC	Source: Shrimp Plant Destination: Retailer
When	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone	Date, Time, Zone
Master Data Header	Plant Operator/Owner Plant Identifier Frozen Shrimp ID			Retail Legal Entity Retail DC ID and Address
ILMD	Product Form Production Date Product Country of Origin			
Tech Info	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event	Product Owner Information Provider Geolocation of Event



5 Internal Traceability – Commingling and Transformations

To address particular concerns in IUU fishing, special attention should be given to commingling and transformation events. As noted in the Explanatory Materials, current business practices and realities in manufacturing environments may give a less than-desirable level of precision to production codes (batches/lots) back to the fishing source. Below is an illustrative, simplified diagram of a processing facility’s CTEs, along with relevant upstream events (NOTE: Other CTEs would normally be included, but for the purposes of this discussion, we’re focusing on harvest (commissioning), commingling, and transformation events, and how they are handled by supply chain actors.)



In this diagram, catches are commingled after landing and before receipt by the primary processing facility. Catch identifiers input to an output identifier associated with the commingled product (through either a GTIN + Lot, UUID, or URL), which the processor receives and feeds into their processing (transformation) steps. As shown above, there may be multiple production lines, splitting, and recombining of product in the factory. The GDST realizes that current capabilities of tracking these internal CTEs may be difficult or will take some time to implement. When combined with upstream commingling events, a production code (batch/lot) may contain many potential catches, which may be less than ideal for regulatory requirements or making sustainability claims.

As digitized traceability using the GDST standards increases, it is anticipated that techniques and reconfiguring of internal traceability practices and systems will improve this granularity. At minimum, all possible inputs should be included in commingling and transformation steps and associated with the output's identifier. If a processor cannot give this level of specificity at individual transformations within the factory, they can be treated collectively in the interim (i.e., treating transformations 1.1, 1.2, 1.3, and 2.0 as one CTE). Traceability should lean more toward overinclusion of inputs rather than under inclusion. Given a product identifier, they should be able to trace back to all possible inputs to the batch/lot (Section 6.1).

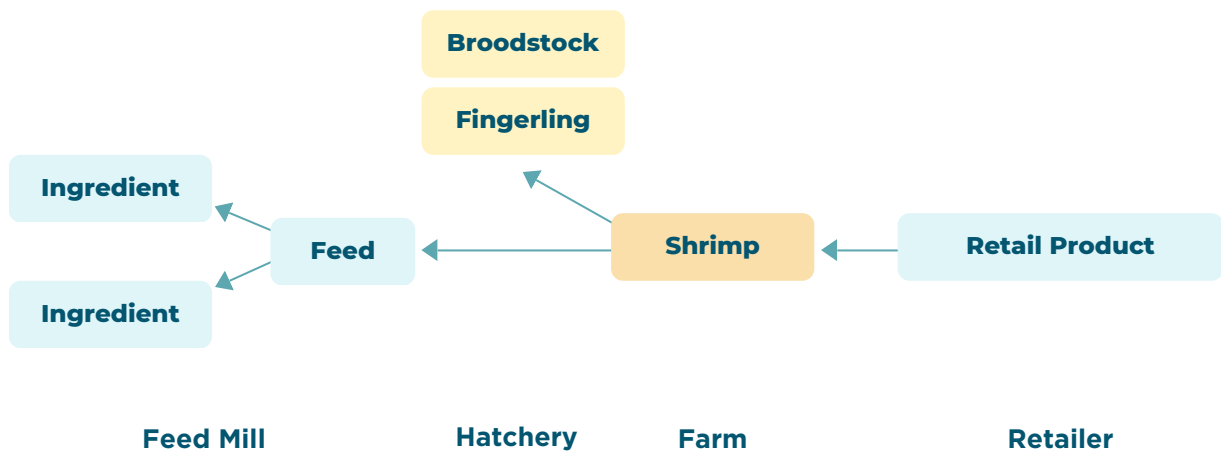


6 Traceability Data Use Cases

6.1 Traceback

Traceback is a common, straightforward use of traceability data, beginning with the traceable object of interest, e.g., the case of frozen shrimp shown in the previous examples.

1. Query the event repository for all events related to the Object Identifier and location(s) of interest (receipts, shipments, pack, unpack, transformation). If no location is specified, all events at all locations will be examined.
2. Order the events by time stamp and group by terminal destination.
3. From each terminal destination, follow the shipping and receiving (source/destination) and trail back to either the original source (catch or harvest) or the output of a transformation event.
4. From the transformation event, repeat steps 1-3 for each Input Object Identifier.
5. The resulting map looks something like this:

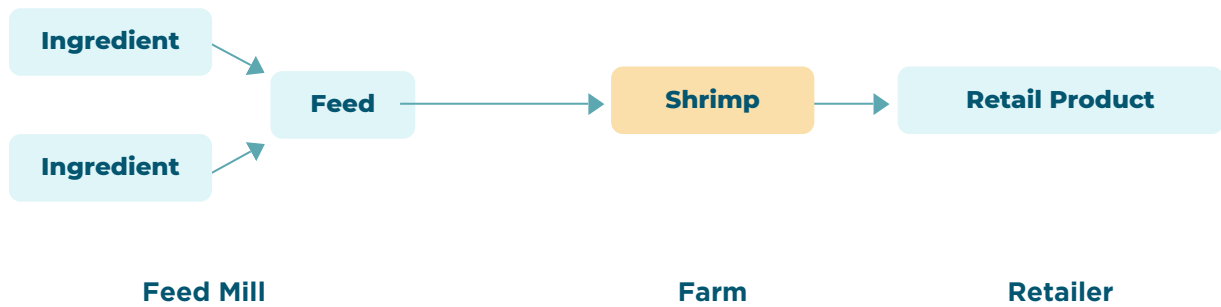


6.2

Trace Forward

Trace forward is also a common use of traceability data, beginning with the traceable object of interest, e.g., the ingredients used in a specific type of feed.

1. Query the event repository for all events related to the Object Identifier and location(s) of interest (receipts, shipments, pack, unpack, transformation).
2. Order the events by time stamp.
3. From the source, follow the shipping and receiving (source/destination) trail forward to the input of a transformation event.
4. From the transformation event, repeat the steps 1-3 for each output of the transformation event and for each new Input Object Identifier found.
5. The resulting map looks something like this:



6.3 Aggregation Report for CSR

Typical commitments are made about the amount of traceable product sold or produced by an entity. Using the same trace forward or traceback approach shown above, in conjunction with the quantity and units of measure, one can total the quantity of finished products or ingredients related to a specific commodity, product, catch location, farming region, or legal entity.

6.4 Mass Balance

Mass balance is a computation of inputs and outputs to establish usage quantities to verify legal usage. Using the traceback and trace forward methodologies described above for a particular batch/lot of products, one can compute the quantity of inputs consumed based on the transformation event. If one has access to all transformation events related to a specific catch, one can compute the total output produced and verify for reasonableness.

6.5 Chain of Custody

The shipping and receiving events include “owning party” for both source and destination. This information in conjunction with the traceback and trace forward methods described above can provide complete chains of custody.



7

Disposition for Product First Entering Commerce

For regulatory compliance, such as with SIMP, the first sale after capture may need to be documented. For this purpose, we have extended the EPCIS disposition for entering commerce.

Disposition	Description	Example
urn:gdst:disposition:entering_commerce	This indicates that a product has changed ownership for the first time since being harvested from a farm or from the wild. A product may appear only in a single event with this disposition.	A product is sold from a fishing vessel to a transshipment vessel.

APPENDIX 1

INTRODUCTORY GS1 MATERIALS

GDST Standards and Guidelines users and stakeholders wishing to have a basic introduction to GS1 and the GS1 tools integrated into the GDST approach may wish to consult the following resources:

1. **GS1 Global Traceability Standard 2.0 (GTS2)**^[1] explains how traceability systems are constructed based on the GS1 system of standards, specifically EPCIS.^[2] This document provides much of the language and fundamental architecture assumed in this guide.
2. **GS1 Foundation for Fish, Seafood and Aquaculture Traceability Guideline**^[3] provides a global view of seafood traceability from first sale to retail.
3. **GS1 US and NFI Seafood Traceability Implementation Guide**^[4] provides specific guidance for North American seafood sold at retail.

GS1 Terminology Regarding Data Sharing and Choreographies

The communication methods applied in the GS1 standards may be broadly classified in two groups:

- Push methods, where one party unilaterally transfers data to another in the absence of a prior request. Push methods may be further classified as:
 - Bilateral party-to-party push, where one party transfers data directly to another party.
 - Publish/subscribe, where one party transfers data to a data pool or repository, which in turn pushes the data to other parties that have previously expressed interest in that data by registering a subscription (“selective push”).
 - Broadcast, where a party publishes business data in a well-known or publicly accessible place, such as a webpage, where it may be retrieved by any interested party. Broadcast does not necessarily mean that the data is available to anyone; the data may be encrypted for a specific intended user, or the broadcast channel (e.g., website) may require the receiving party to authenticate and may grant access to the broadcast data only according to specific access control policies.
- Pull or query methods, where one party makes a request for specific data to another party, which in turn responds with the desired data. Note that in the classification of push methods above, the broadcast method may also involve a pull query in order to retrieve the data from a publicly accessible place (such as a website).

[1] <https://www.gs1.org/standards/traceability/traceability/2-0>

[2] <https://www.gs1.org/standards/epcis>

[3] <https://www.gs1.org/standards/traceability/guideline/gs1-foundation-fish-seafood-and-aquaculturetraceability-implementation>

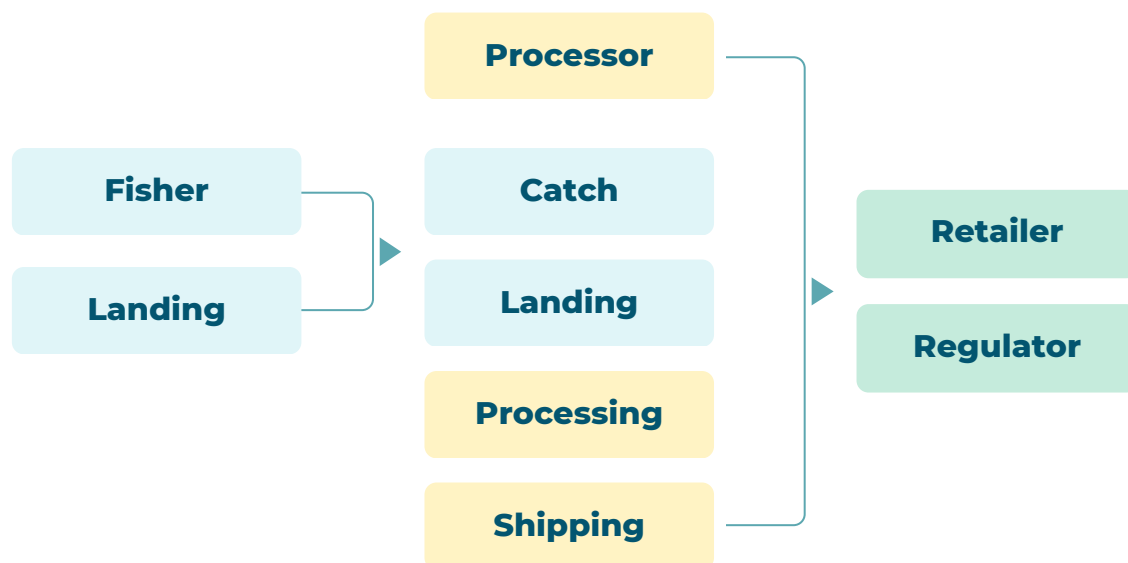
[4] <https://www.gs1us.org/industries/retail-grocery/standards-in-use/fresh-foods>

The table below details how these push and pull methods work with various choreographies.

Traceability choreography	Publishing/contributing	On-demand selective query (synchronous)	Selective standing query (asynchronous)
One step up One step down	<p>Push data to relevant trading partner</p> <p>Example: bilateral EDI or EPCIS message</p>	Pull (request, response)	Publish and subscribe (receive push notifications matching previous standing query subscription)
Centralized	<p>Push data to centralized repository via API, file upload, mobile device, or web browser data entry</p> <p>Examples: most commercial traceability applications operated by a supply chain owner (IBM Food Trust, Trace Register, FoodLogiQ, Ftrace, Origin Trail, VeChain, TE-FOOD, Fishcoin, etc.)</p>	Pull via API (request, response)	N/A
Discovery of networked resources	<p>Push referral link to a discovery service</p> <p>Examples: GS1 Digital Link, GS1 GDSN</p>	Pull (request, response)	N/A
Networked	<p>Push data to own repository</p> <p>Example: distributed EPCIS repository</p>	Pull (request, response)	Publish and subscribe (receive push notifications matching previous standing query subscription)
Cumulative (recommended; see detail below)	<p>Push cumulative data to next one-down party or between centralized databases</p> <p>Example: GS1 Pedigree Standard</p>	<p>N/A</p> <p>(downstream parties automatically receive all relevant upstream data; no need to support selective queries)</p>	
Decentralized and replicated	<p>Push and pull data to/from one node for validation then inclusion and replication in decentralized repository or ledger. This enables applications to build off each other because some or all of the data is exposed on a blockchain.</p>		

Cumulative (push): [Recommended for harvest-to-primary-processing CTEs] This approach is a push method where the traceability data is systematically enhanced and pushed forward to the next party in the chain in parallel with the product flow. It enables the sharing of upstream data with parties further downstream, but not the opposite. This approach is recommended as the first considered choreography, especially among CTEs from catch to primary processing because of the transactional nature of these events.

This approach results in asymmetric visibility across the supply chain, in which downstream parties receive a complete copy of all relevant upstream data while the upstream parties have no visibility downstream beyond their immediate one-down customer. This ensures data security for supply chain owners (retailers, restaurants) and provides explicit, fine-grained control for what is shared at each stage in the supply chain. For example, a retailer may prefer a summary of the pedigree, while a certification body might prefer a complete, unedited original event listing. This method would serve both purposes. Below is an example:



The collection of traceability data from other parties and the provision of data to other parties are essential components in distributed traceability systems.

Data-sharing choreographies are all, in principle, capable of selectively restricting access to the meaning of the exchanged data on a need-to-know basis, although they differ in the mechanisms used and in the ability to control whether a receiving party shares the data with additional parties:

- Some of the choreographies involve bilateral communication between an information requesting party (querying party) and an information-providing party, which may be the original contributor of the data or a shared repository holding the data. The privacy of such bilateral communications can be ensured via mutual authentication, the use of secure communication channels, and the potential encryption of the data payload or messages.
- Decentralized and replicated choreographies can involve a different approach to selectively restricting access to the meaning of the data. In the case of a blockchain ledger, trust in the ledger is ensured if everyone is able to independently inspect the entire ledger, including all its data, in order to be assured that no historical transaction data has been subsequently altered. Although this openness necessarily means that anyone can read all the data in the ledger, it is still possible to hide the meaning of sensitive data either by encrypting such data or by storing a hash value in the ledger. If hash values are stored in a blockchain ledger, the original data is typically stored elsewhere and exchanged by another mechanism, while the hash value recorded in the blockchain ledger effectively archives a “tamper-evident seal” that corresponds to what the data originally looked like. Both security approaches compromise the first principle of the technology – that everyone can independently inspect all the data in the ledger to ensure completeness and accuracy. Until this fundamental issue is resolved, decentralized, and replicated technologies will not provide a demonstrably better tool for traceability.

APPENDIX 2

GDST GITHUB

The GDST anticipates that the standards documented in this packet of materials will not address every supply chain contingency, certification scheme, or regulatory requirement. Because of the extensibility of EPCIS, there is the potential to have “too much” flexibility, wherein solution providers and seafood companies represent this information in divergent methods. To have a concerted approach and standard reference, the GDST has created a GitHub repository to extend documentation to new or non-normative situations, such as:

- Specific documentation to common certifications o E.g., MSC/ASC and Fair Trade
- Regulatory guidance o E.g., SIMP or EU eCDT
- Gear-type documentation o E.g., trawling catch events
- Species-specific documentation o Representing life cycle events for specific aquaculture species

The GDST GitHub utilizes the ticket system already in use by the site to flag, process, and manage new issues or documentation needs. The site uses Spectrum for users to discuss in process issues, and the site will be moderated by the GDST Secretariat. Details of the process, governance, and structure are housed on the repository site.

Other documentation:

- Example and annotated EPCIS pedigree files corresponding with the examples referenced in section 3-4.
- Capability Tool resources
- Completeness Test resources

The GDST GitHub utilizes the ticket system already in use by the site to flag, process, and manage new issues or documentation needs. The site uses Spectrum for users to discuss in process issues, and the site will be moderated by the GDST Secretariat. Details of the process, governance, and structure are housed on the repository site.

Documentation home: <https://developer.traceability-dialogue.org/>

In-process page: <https://github.com/iftgftc/gdst>



GLOBAL DIALOGUE
on Seafood Traceability

STANDARDS AND GUIDELINES FOR INTEROPERABLE SEAFOOD TRACEABILITY SYSTEMS

Technical Implementation Guidance (Version 1.2)

JUNE 2023