GLOBAL DIALOGUE on Seafood Traceability

Standards and Guidelines for Interoperable Seafood Traceability Systems – Technical Implementation Guidance (Version 1.0)

February 2020

Document Summary

Document Name	***GDST Standards and Guidelines for Interoperable Seafood Traceability Systems – Technical Implementation Guidance***
Document Date	February 10, 2020
Document Version	1.0
Document Status	Released to GDST members (publication to follow on March 16, 2020)
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 1.0 materials. The packet as of February 10, 2020, includes:

Document Title	Document Date	Version	Contents
Guide to GDST 1.0 Materials	February 2020	v1.0	Overview of GDST 1.0 packet contents and "How to Use These Documents"
Executive Summary	February 2020	v1.0	Two-page description of GDST 1.0
Core Normative Standards	February 2020	v1.0	The GDST 1.0 standards themselves
Basic Universal List of KDEs (spreadsheet)	February 2020	v1.0	E-spreadsheet of appendices to *Core Normative Standards* – part of GDST 1.0 core standards
Explanatory Materials	February 2020	v1.0	Nontechnical background and introductory materials
Technical Implementation Guidance	February 2020	v1.0	Additional technical materials to facilitate implementation

A drafting history of the industry-led inputs into GDST 1.0 appears in Section 1.3 of the *Explanatory Materials* document.

For online access to the full GDST 1.0 packet, visit http://traceability-dialogue.org/core-documents/gdst-1-0-materials/.

For additional information, please contact the GDST Secretariat at info@traceability-dialogue.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 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.

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.

---

[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-aquaculture-traceability-implementation

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

Table of Contents

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
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
LGTIN	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 system requirements to plan and scope a company's implementation of the GDST standards. 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

Internal traceability, specific to a company or enterprise, is used to meet many business needs (e.g., regulatory compliance, inventory management), but for the purposes of interoperable end-to-end traceability, the system elements are straightforward:

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

1.2. 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:

- Update interfaces for traceable object capture (typically bar code scanning or manual entry via mobile or web) capable of scanning, processing, and storing the expected identifiers received from immediate trading partners.
- Update inbound and outbound machine-to-machine interfaces supporting the EPCIS format and technologies for seafood traceability interoperability.
- Test with all appropriate entities, including trading partners, regulators, certification standard bodies, brokers, importers, and exporters.

1.3. Managing Traceability Data

1.3.1. 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, locations, and assets, e.g., “Address,” and “Species Code and Name.” Data is often first captured in accounting software applications followed by logistics applications. Trading partners typically share their location and product information via paper, spreadsheets, centralized web portals, and the GS1 Global Data Sharing Network. The challenge is often the lack of a single source of truth, often leading to erroneous master data.
- **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

One of the most interesting discussions in the traceability world is the growing number of approaches to sharing data. As the GDST is focused primarily on standardizing the means of traceability data semantics and syntax, the GDST has selected an open data-sharing approach rather than explicitly requiring a particular protocol or platform. The discussion and recommendations on data sharing are in Section 2. Additionally, information on data-sharing choreographies are detailed in Appendix 1.

1.3.4. 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?

1.4. 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. Sharing Traceability Data and Communication Protocols

The GDST has taken an open approach to data sharing with the understanding that trade relationships take on various forms and technology adoption may dictate the level of coordination needed between supply chain actors. The GDST understands that the next critical step in interoperability is defining how the specified data format will be communicated. This section is meant to define how businesses and developers can communicate GDST data in a clear and concise manner.

2.1. Communication Protocol Conditions

Because many communication protocols may be used, the GDST has recommended the following conditions for selecting and using a protocol:

1. The ability to exchange the information in the specified data format (i.e., GDST EPCIS).

2. The ability to know that the communication came from the purported entity or supply chain actor.

3. The ability to know that the communication was not intercepted and changed.

4. The ability to know the communication's level of transparency and that only authorized entities may see what is being communicated.

5. The ability to receive an acknowledgment that the communication was received and to know the status on the processing of the communication, such as error messaging and success statuses.

2.2. Additional Considerations

After defining the conditions of a successful communication protocol, additional considerations for the given communication protocol include:

- The communication should be built with standards in mind, such as:
  - GS1 Seafood Traceability
  - GS1 Digital Link
  - EPCIS 1.2
  - OpenAPI
- The communication protocol should be easy for businesses and developers to implement.

2.3. Communication Scenarios

The communication protocol should support three different levels of communication. We have outlined recommended approaches to each of them in sections 2.5–2.7. All of these meet the five requirements detailed in Section 2.1.

  - **Digitized business communication**
    - This involves the communication between two businesses that have digitized systems using either internally developed software solutions or an external software solutions provider.

        - In the absence of an existing data-sharing protocol, the GDST recommends following the REST API architecture covered in Section 2.4.
    - **Nondigitized business communication**
        - This involves the communication between two businesses that do not have their information processes digitized. The expectations of these businesses are that all their data is stored in spreadsheets and they use email to communicate.
    - **Mixed business communication**
        - This involves the communication between two businesses where either the sender or the receiver is nondigitized and the other is digitized.

2.4. Digitized Business Communication Protocol Recommendation

When two digitized businesses are exchanging information but do not currently have an established protocol conveying traceability information (e.g., EDI), the GDST recommends using a REST API utilizing an OpenAPI configuration file. The businesses could be implementing this communication protocol in-house, or they could be using a software solutions provider to implement it for them.

The GDST has recommended REST APIs because of their ease of implementation and use of the most popular programming languages today, including C#, Java, JavaScript, and Python, which have built-in tools for supporting REST servers and clients. On GDST's GitHub, OpenAPI configuration files are available for the REST API so that companies may take advantage of powerful tools like Swagger Hub to jump-start developers in writing server-side and client-side code for the REST API. This REST API will rely on EPCIS 1.2 message formats to communicate visibility events optionally containing EPCIS master data.

The REST API also requires HTTPS and synchronized processing, meaning the message should be processed during the HTTP request, and the result of that processing should be responded to in the HTTP response.

2.4.1. Authentication

The REST API would support the following authentications:

1. **No authentication** – Meaning anyone can query the REST API and find out information.
2. **Basic authentication** – Meaning that the authorization header on the HTTP request contains the following format: ***Basic <username>:<password>*** with the ***<username>:<password>*** encrypted into Base64 so that it is not clear text.
3. **API key** – Meaning that there is a query parameter specified like ***?key=<insert_API_key_here>***.

On top of authentication, it's possible that, depending on whether authentication is provided, the REST API could return different levels of detailed information. For example, if authentication is not provided, it might provide a less detailed version of the information, stripping out any sensitive information and delivering just the bare minimum.

2.4.2. REST API Methods

Here, we start to go into detail about the specific methods that will be exposed on this REST API and how each method should behave. This will not include the methods defined by the

EPCIS Query Interface. More information on methods available through the EPCIS Query Interface are available here and on the GitHub.

For the purpose of this section, we will assume that the root URL for the REST API is https://example.org/GDST/.

Full examples of requests and responses, including the request URL, request HTTP headers, request HTTP body, response HTTP headers, and response HTTP body, can be found on the GitHub here.

Pull Events

This is a GET method where the sender is requesting a list of events that are associated with a specific EPC.

HTTP operation: GET

Path: /events

Example REST URL
https://example.org/GDST/events/{EPC}

Response: EPCIS document

Response content type: application/xml

Push Events

This is a POST where the sender is sending one or more EPCIS events to the receiver.

HTTP operation: POST

Path: /events

Example REST URL
https://example.org/GDST/events/

Response: Nothing; will rely on the HTTP response code to determine the response.

Request content type: application/xml

Request format: EPCIS message

2.4.3. More Help

Please visit the GDST GitHub for more information on the following:

- The OpenAPI YAML/JSON configuration file
- Example requests and responses for every method
- Example client-side implementations of the REST API
- Example server-side implementations of the REST API
- Examples on how to use Swagger Hub

2.5. Nondigitized Communication Recommendation

This communication protocol is recommended for two businesses that do not have digitized information systems. Here, we would like to recommend that these businesses use email to exchange the data proposed.

1. The sender takes the one or more XML files that they want to communicate and zips them into a WinZip file and emails it to the receiving business.
2. The receiver responds to that email with an acknowledgment as to how the XML files were received and whether they require any more information. This response will be done in free text.
3. Security protocols to prevent email spoofing should be in place to verify traceability data origin.
4. Emails should be replied to in order to ensure receipt acknowledgment.

2.6. Mixed Communication Recommendation

This communication protocol is recommended is when a nondigitized business is communicating with a digitized business and vice versa. This communication protocol is very similar to the nondigitized recommendation. The only additional work that will be required here is that the digitized business will be programmatically reading and sending the emails. This should be easily implementable – popular languages like C#, Java, and Python all have free libraries available for performing these actions.

2.6.1. Digitized Business Sending to Nondigitized Business

1. The sender programmatically takes the XML files that they wish to send and zips them into a WinZip file.
2. The sender programmatically sends the XML files to the receiver's email.
3. The receiver receives the email and verifies its contents.
4. The receiver responds to the email with a free-text acknowledgement.
5. The sender programmatically receives the acknowledgment and stores it for the user to manually read and confirm.

2.6.2. Nondigitized Business Sending to Digitized Business

1. The sender takes the one or more XML files that they want to communicate and zips them into a WinZip file and emails it to the receiving digitized business.
2. The receiver programmatically reads the email and imports the XML files into their system.
3. The receiver uses the XML schemas provided to match the XML files to their message type.
4. The receiver programmatically responds to the email with a free-text acknowledgment.
5. The sender manually receives the acknowledgment.

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 file: link

Source:
https://www.youtube.com/watch?v=B02S3GOILW4

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

Frozen Tuna Loins

500 KG

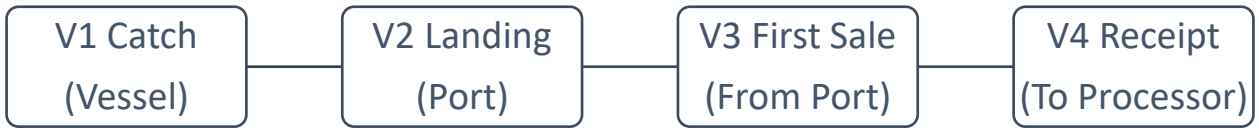
Catch Date

Product of Taiwan

July 09, 2017

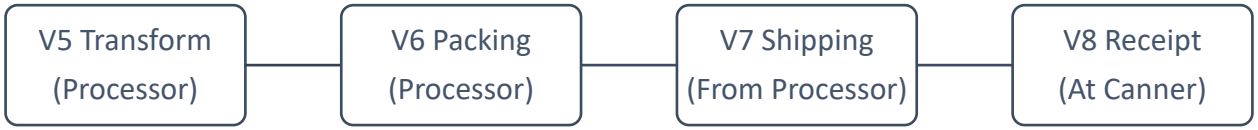
Seafood Company

Taipei, Taiwan



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
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

Dimension	V1	V2	V3	V4
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	TransformationEvent ADD	AggregationEvent ADD, Aggregation	ObjectEvent Shipping	ObjectEvent OBSERVE, Receiving
Who	Processor	Processor	Vessel Operator	Processor

Dimension	V5	V6	V7	V8
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	Lot Number 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:** Loin Plant Destination: Canner
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		

Dimension	V9	V10	V11	V12
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 file: Link

Source: food service direct

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

26–30 Raw Tail-On White Shrimp

2 LB x 5 per Case

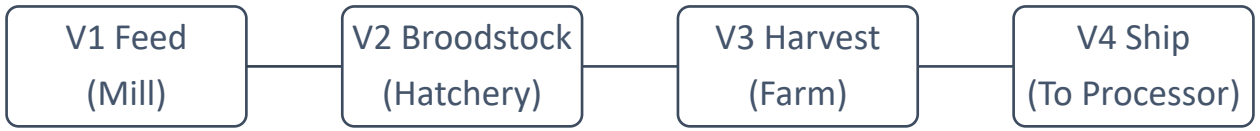
Product of Thailand

Shrimp Farming Company

Bangkok, Thailand

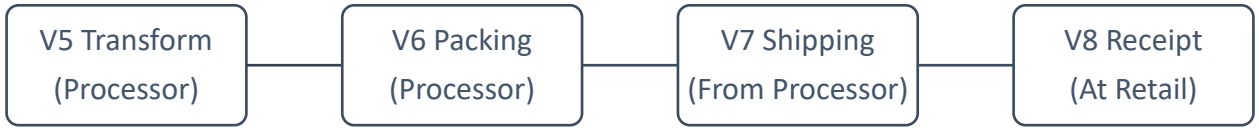
Harvest Date

July 09, 2017



Dimension	V1	V2	V3	V4
Why	TransformationEvent Add	ObjectEvent ADD, Farm Stock	TransformationEvent ADD, Farm Harvest	ObjectEvent OBSERVE, Shipping
Who	Feed Mill	Farmer	Farmer	Farmer
What	Input: Ingredients Output: Feed ID	Broodstock ID	Input: Feed ID, Broodstock 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 Broodstock Identifier	Farm Owner Farm Identifier Live Shrimp Identifier	Processor ID and Address

Dimension	V1	V2	V3	V4
ILMD	Source of Protein Certification List	Harvest Start/End Date Source of Broodstock 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

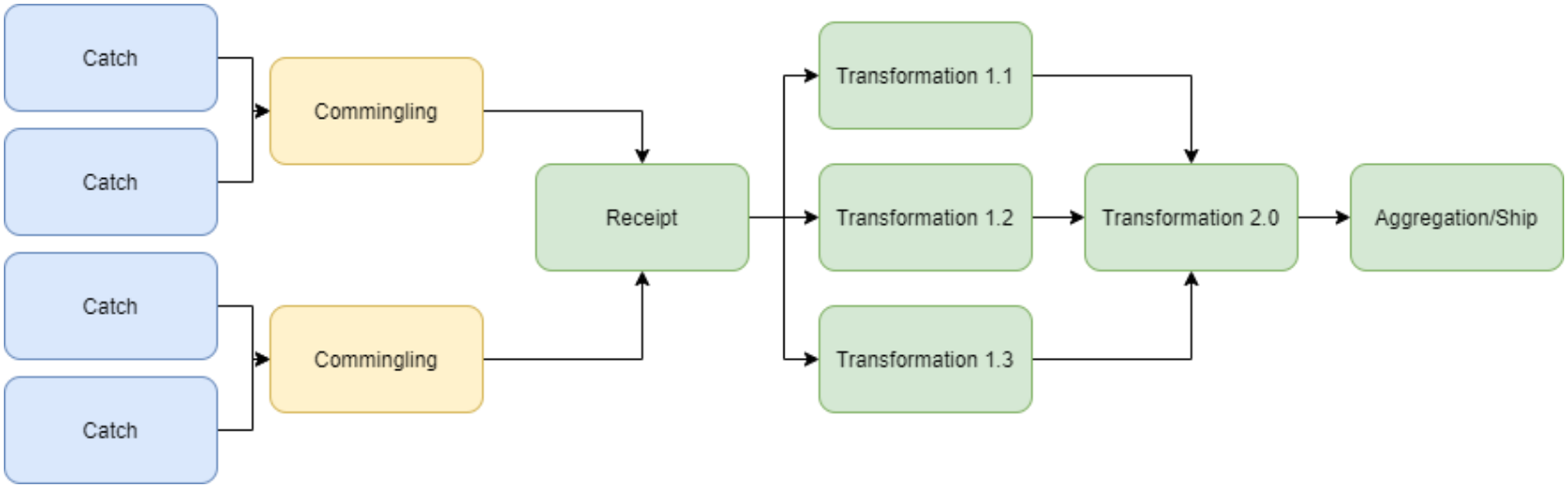


Dimension	V5	V6	V7	V8
Why	TransformationEvent	AggregationEvent ADD, Aggregation	ObjectEvent OBSERVE, Shipping	ObjectEvent OBSERVE, Receiving
Who	Processor	Processor	Processor	Retailer
What	**Input:** Live Shrimp ID **Output:** Frozen Shrimp	Parent: Pallet Children: Frozen Shrimp Cases	Pallet	Pallet
Where	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

Dimension	V5	V6	V7	V8
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.)



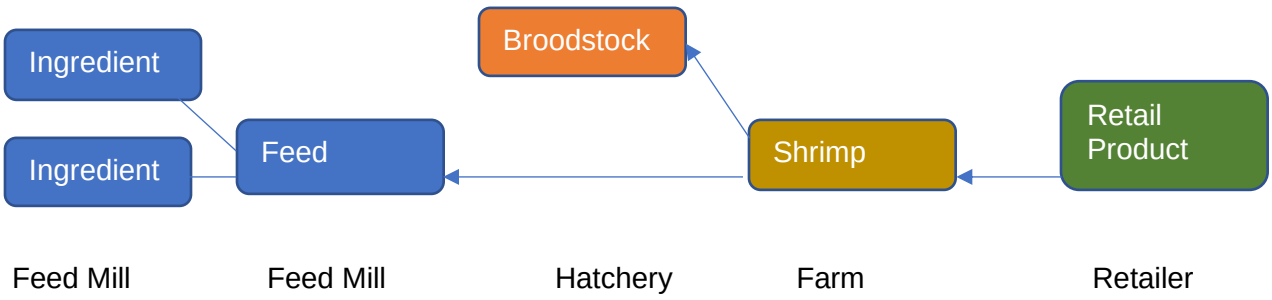
In this diagram, catches are commingled after landing and before receipt by the primary processing facility. Catch identifiers input to an output identifier associated with the commingled product (through either a GTIN + Lot, UUID, or URL), which the processor receives and feeds into their processing (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 underinclusion. 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.

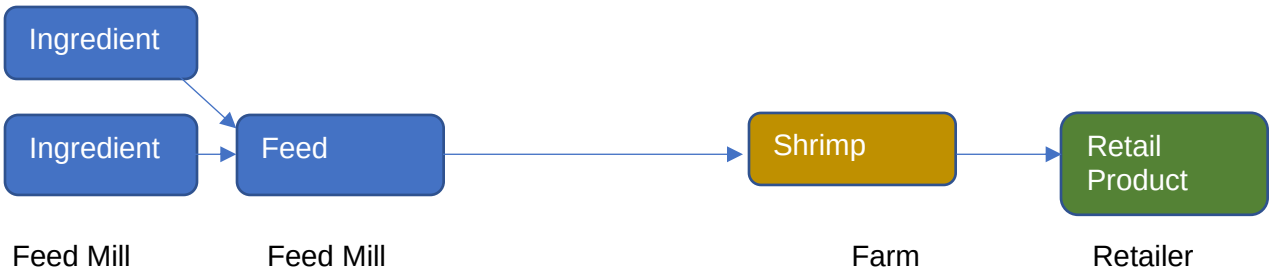
1. Query the event repository for all events related to the Object Identifier and location(s) of interest (receipts, shipments, pack, unpack, transformation). If no location is specified, all events at all locations will be examined.
2. Order the events by time stamp and group by terminal destination.
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. Traceforward

Traceforward 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 traceforward 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 traceforward 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 traceforward 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)**[5] explains how traceability systems are constructed based on the GS1 system of standards, specifically EPCIS.[6] This document provides much of the language and fundamental architecture assumed in this guide.
2. **GS1 Foundation for Fish, Seafood and Aquaculture Traceability Guideline**[7] provides a global view of seafood traceability from first sale to retail.
3. **GS1 US and NFI Seafood Traceability Implementation Guide**[8] 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 or the GDSN, 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

---

[5] https://www.gs1.org/standards/traceability/traceability/2-0

[6] https://www.gs1.org/standards/epcis

[7] https://www.gs1.org/standards/traceability/guideline/gs1-foundation-fish-seafood-and-aquaculture-traceability-implementation

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

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).

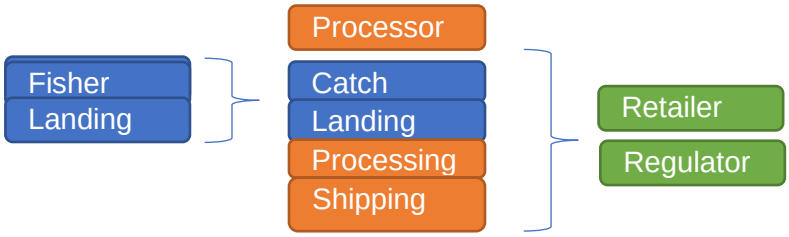
The table below details how these push and pull methods work with various choreographies.

Traceability choreography	Publishing/contributing	On-demand selective query (synchronous)	Selective standing query (asynchronous)
One step up One step down	Push data to relevant trading partner Example: bilateral EDI or EPCIS message	Pull (request, response)	Publish and subscribe (receive push notifications matching previous standing query subscription)
Centralized	Push data to centralized repository via API, file upload, mobile device, or web browser data entry 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.)	Pull via API (request, response)	N/A
Discovery of networked resources	Push referral link to a discovery service Examples: GS1 Digital Link, GS1 GDSN	Pull (request, response)	N/A
Networked	Push data to own repository Example: distributed EPCIS repository	Pull (request, response)	Publish and subscribe (receive push notifications matching previous standing query subscription)

Cumulative (recommended; see detail below)	Push cumulative data to next one-down party or between centralized databases Example: GS1 Pedigree Standard	N/A (downstream parties automatically receive all relevant upstream data; no need to support selective queries)
Decentralized and replicated	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.	

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
  - E.g., MSC/ASC and Fair Trade
- Regulatory guidance
  - E.g., SIMP or EU eCDT
- Gear-type documentation
  - E.g., trawling catch events
- Species-specific documentation
  - 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 to be included:

- Example and annotated EPCIS pedigree files corresponding with the examples above
- Open-source tools from hackathons
- Descriptions of GS1 tools for testing
  - FreEPCIS
  - Data Visualization Workbench (DataVizWorkbench)
  - Oliot

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