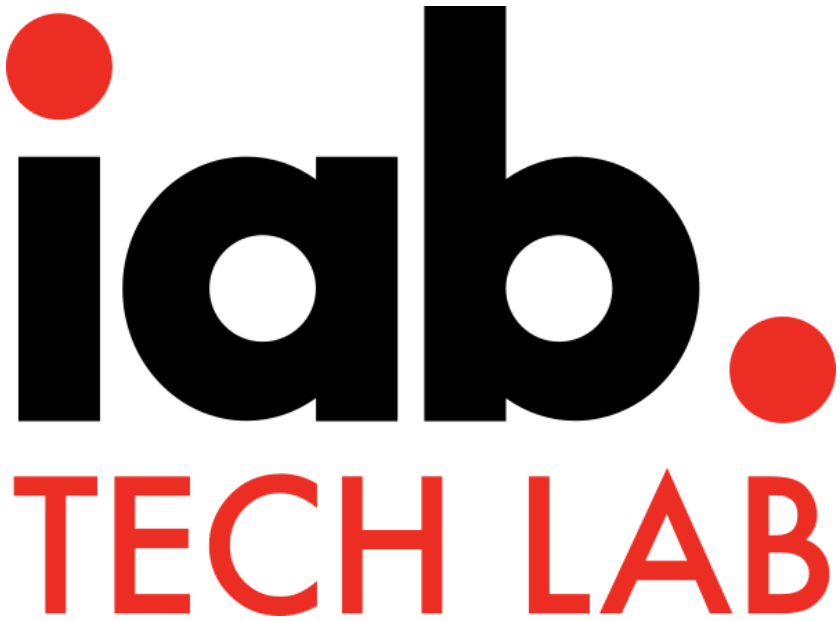


Accountability Platform

Version 1.0

Released November 5, 2024

Please email support@iabtechlab.com for questions. This document is available online at https://iabtechlab.com/accountability-platform/.

About this document
The Accountability Platform is a specification for open, auditable data structures and standard practices to detect miscommunications and demonstrate, via standard data structures and reporting, accurate communications across the digital advertising supply chain of data use preference and restriction signals set by users and the digital properties they visit.

This document has been developed by the Rearc Accountability Working Group.

About IAB Tech Lab
The IAB Technology Laboratory is a nonprofit research and development consortium charged with producing and helping companies implement global industry technical standards and solutions. The goal of the Tech Lab is to reduce friction associated with the digital advertising and marketing supply chain while contributing to the safe growth of an industry. The IAB Tech Lab spearheads the development of technical standards, creates and maintains a code library to assist in rapid, cost-effective implementation of IAB standards, and establishes a test platform for companies to evaluate the compatibility of their technology solutions with IAB standards, which for 18 years have been the foundation for interoperability and profitable growth in the digital advertising supply chain. Further details about the IAB Technology Lab can be found at https://iabtechlab.com.

IAB Tech Lab Contact
Rowena Lam, Sr Director, Privacy & Data

License

Disclaimer

THE STANDARDS, THE SPECIFICATIONS, THE MEASUREMENT GUIDELINES, AND ANY OTHER MATERIALS OR SERVICES PROVIDED TO OR USED BY YOU HEREUNDER (THE "PRODUCTS AND SERVICES") ARE PROVIDED "AS IS" AND "AS AVAILABLE," AND IAB TECHNOLOGY LABORATORY, INC. ("TECH LAB") MAKES NO WARRANTY WITH RESPECT TO THE SAME AND HEREBY DISCLAIMS ANY AND ALL EXPRESS, IMPLIED, OR STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AVAILABILITY, ERROR-FREE OR UNINTERRUPTED OPERATION, AND ANY WARRANTIES ARISING FROM A COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. TO THE EXTENT THAT TECH LAB MAY NOT AS A MATTER OF APPLICABLE LAW DISCLAIM ANY IMPLIED WARRANTY, THE SCOPE AND DURATION OF SUCH WARRANTY WILL BE THE MINIMUM PERMITTED UNDER SUCH LAW. THE PRODUCTS AND SERVICES DO NOT CONSTITUTE BUSINESS OR LEGAL ADVICE. TECH LAB DOES NOT WARRANT THAT THE PRODUCTS AND SERVICES PROVIDED TO OR USED BY YOU HEREUNDER SHALL CAUSE YOU AND/OR YOUR PRODUCTS OR SERVICES TO BE IN COMPLIANCE WITH ANY APPLICABLE LAWS, REGULATIONS, OR SELF-REGULATORY FRAMEWORKS, AND YOU ARE SOLELY RESPONSIBLE FOR COMPLIANCE WITH THE SAME, INCLUDING, BUT NOT LIMITED TO, DATA PROTECTION LAWS, SUCH AS THE PERSONAL INFORMATION PROTECTION AND ELECTRONIC DOCUMENTS ACT (CANADA), THE DATA PROTECTION DIRECTIVE (EU), THE E-PRIVACY DIRECTIVE (EU), THE GENERAL DATA PROTECTION REGULATION (EU), AND THE E-PRIVACY REGULATION (EU) AS AND WHEN THEY BECOME EFFECTIVE.

Glossary

Ad-related Sharing	Transactions with partners or providers related to, or resulting from, the delivery of advertising.
Accountability Platform	A platform designed to provide a consistent means by which digital ad industry participants, self-regulatory regimes, auditors, and other interested parties can evaluate the correctness and completeness of communication of user preference signals within the digital ad supply chain.
Chain Identifier	An Accountability Platform-specific, globally-unique transaction identifier generated by the initial Sender in a chain of Sender/Receiver transactions and logged by subsequent participants.
Common Operator	An entity or role that coordinates the gathering of data from participants, processes the data, and makes outputs available. The Common Operator is expected to consist of a group of geographically distributed nodes coordinated through a common process and configuration data.
Ecosystem Participants	Entities in the AdTech value chain that transact data for the purpose of delivering advertising and which could have associated preference signals, but may not.
Global Privacy Platform (GPP) String	A specific type of user preference signal defined by the IAB Tech Lab Global Privacy Platform specification.
Identifiers	Unique values assigned to entities such as devices, user-agents, users or households for the purpose of associating them with datasets.
Match Value	The value used to ensure that Senders and Receivers provide samples from the same subsets of records, enabling their matching for comparison.
Receiver	Entity receiving data in an ad-related transaction.
Sender	Entity sending data in an ad-related transaction.

Transaction Identifier	A UUID (Universally Unique Identifier) for a discrete or longitudinal transaction.
Transparency & Consent Framework (TC) String	A specific type of user preference signal defined by IAB Tech Lab's Transparency & Consent Framework specification.
User Preference Signals	Signals that communicate user preferences related to digital advertising. Examples include the Global Privacy Platform (GPP) String and the Transparency & Consent Framework (TC) String.
Value-chain Transactions	Transactions within the AdTech ecosystem.

Table of Contents

Overview

The objective of the Accountability Platform is to provide a consistent means by which digital ad industry participants, self-regulatory regimes, auditors and other interested parties can evaluate the correctness and completeness of communication of user preference signals within the digital ad supply chain and does so at AdTech scale.

The primary intent is to support parties communicating user preference signals (e.g., the IAB Tech Lab Global Privacy Platform GPP string). The first goal is to ensure the integrity of the signal by providing a means of validating that signals are being received as sent by all participants in a chain or, in cases where there are issues, providing information that can aid in diagnosing and remediating problems. A second and equally important goal is to provide data that can be used to monitor the completeness of communication throughout the ads ecosystem, from original source to final destination, and enable diagnosis and remediation of propagation failures.

More simply these can be thought of as ensuring correctness of communication, assuring completeness of communication and understanding pervasiveness of communication of preference signals.

The general intent of the platform is to encourage responsible use of identifiers in ad delivery and measurement use-cases by providing support for employing them in accordance with user preferences. It is expected that the platform will encourage participants to be more mindful of their use of identifiers and careful in the handling and application of them. It is anticipated that as a consequence users will be more willing to support use of identifiers, knowing their use is monitored, validated and easily audited.

Although support for monitoring signals that accompany user identifiers specifically is a critical focus of the platform, increasing requirements for respecting more general user preferences to opt-out of any targeted advertising, such as the Global Privacy Control, suggested the focus of the platform should be extended to include all ad-related user-preference signaling.

Scope

The platform is intended to support monitoring of any ad-related sharing of user data through periodic collection of samples. These samples are explicitly designed to not contain any user data other than the preference signals. User data includes any data which may inform decisions affecting the user and over which the user could be provided an opportunity to exercise control. Ad-related sharing includes any transactions with partners or providers related to, or resulting from, the delivery of advertising. Samples are intended to provide a representative view of how preference signals are employed within the ecosystem while minimizing resource demands. Any set of data generated from ad-related events which includes the fields required by this platform is considered to be covered by the platform.

Out-of-scope
The focus of the current version is specifically not concerned with how participants apply such signals or with the monitoring of what data is shared.

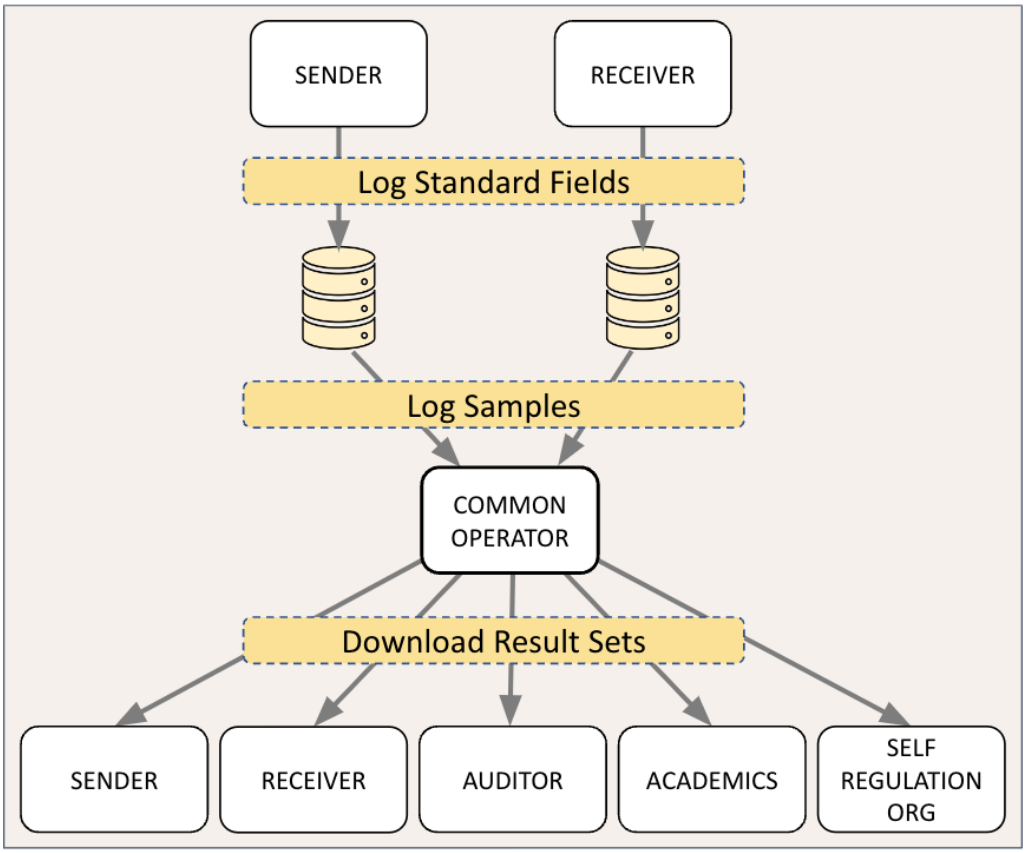
Architecture

Roles

There are two principal roles in the Accountability Platform: ecosystem participants and a Common Operator.

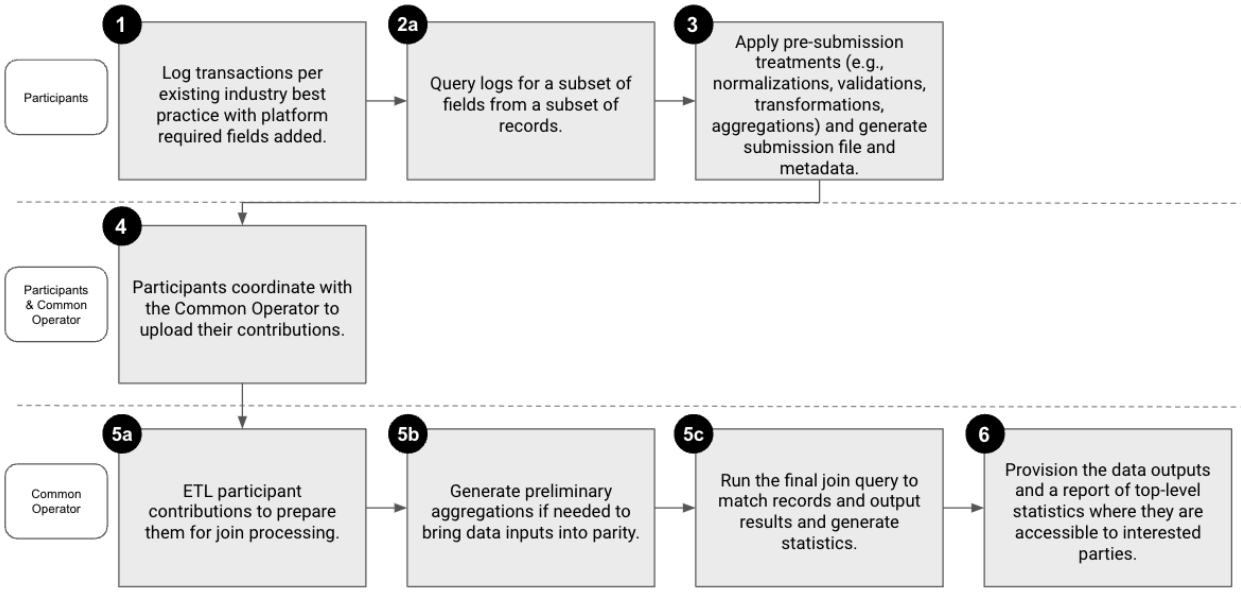
Ecosystem participants include all entities in an AdTech value chain that transact data derived from a relationship with a user and which could have associated preference signals. Ecosystem participants are further divided into two additional roles played in data-related transactions: they are either a Sender who is providing data or a Receiver to whom data is being provided. A given participant can play both roles and, in the case of intermediaries, generally will.

The Common Operator coordinates the gathering of data from participants, the processing of the data and making outputs available.



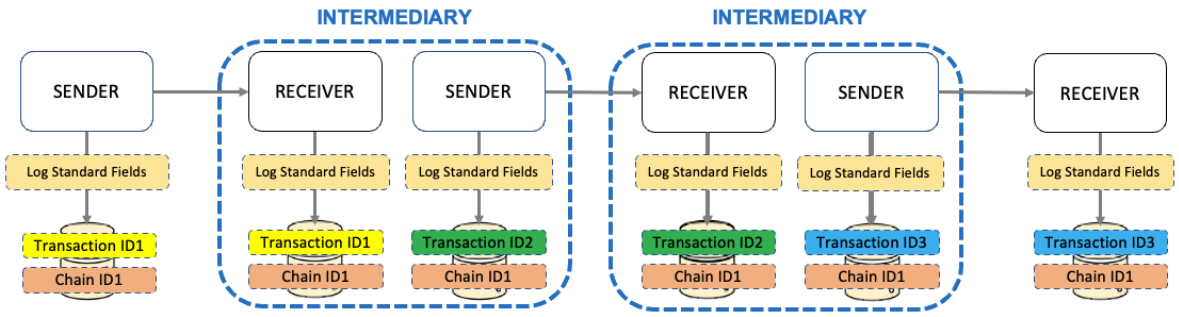
The accountability platform defines a cycle consisting of six phases:

1. Logging - Responsibility of ecosystem participants
2. Data collection - Responsibility of ecosystem participants and Common Operator
3. Post-collection preparation - Responsibility of ecosystem participants and Common Operator
4. Submission - Responsibility of ecosystem participants and Common Operator
5. Processing - Responsibility of Common Operator
6. Provisioning - Responsibility of Common Operator



Logging

During the logging phase, ecosystem participants record information about value-chain transactions. The data, described in the sections Sender Log Data Requirements and Receiver Log Data Requirements below, consists primarily of values already being logged for other use-cases. There are two exceptions: (1) a platform-specific globally-unique chain identifier which is generated by the initial Sender in a chain and logged by subsequent participants (2) a transaction identifier in the form of a UUID generated by a Sender in a Sender / Receiver pair and logged by the Receiver.



Sender Log Data Requirements

For all covered transactions, Senders will log the following:

- Transaction time in a form which can be converted with second accuracy to unix epoch time.
- Receiver in a form which can be converted to the non-repudiable eTLD+1 domain of the entity receiving data.
- Chain ID which is a UUID. If a chain ID was received from an upstream Sender, it should be logged and provided to the Receiver. If no chain ID is available, one is generated and provided to the Receiver.
- Transaction ID which is a UUID that uniquely identifies the single transaction.
- Any user preference signal provided to the Receiver.

Receiver Log Data Requirements

For all covered transactions, Receivers will log the following:

- Transaction time in a form which can be converted with second accuracy to unix epoch time.
- Sender in a form which can be converted to the non-repudiable eTLD+1 domain of the entity sending data.
- Transaction ID which is a UUID included in the transaction by the Sender.
- Any user preference signal string provided by the Sender.

Data Collection, Preparation and Submission

The Common Operator will periodically initiate a data collection cycle by posting a request for submissions in a well-known location. Participants will be expected to poll the well-known location on a periodic basis to determine if a new request has been made.

Posted requests will include parameters describing the data to be submitted and other aspects of the submission process as described in Data Collection Request. When a request is received, participants will be expected to gather the data requested, prepare it so it is

normalized, mutated and formatted properly and deliver it in an appropriate form to a designated location by a provided submission deadline.

Data Collection

In the first part of the process, a sample of log data is gathered by participants based on parameters supplied by the Common Operator.

General Configuration Information

The Common Operator defines the normalization, transformation, and/or aggregation rules that should be applied to field values before submission. The Common Operator will also define a lookback window. Normalization, validation, transformation, and aggregation rules are independently versioned, with the version being incremented when material changes are made to any rules. The Common Operator will maintain all historical ruleset versions.

Field	Description
Lookback Window	The maximum number of seconds in the past that log data may be requested for and therefore should be maintained for. The Common Operator uses this parameter to indicate to participants the minimum amount of time they need to retain log data that may be included in a request for and it allows the Common Operator to balance ensuring participants can support requests with limiting the resources participants must expend for retaining logs.
Normalization Rules Version	Identifies the version of the data normalization rules participants should apply to field values before submission.
Validation Rules Version	Identifies the version of the data validation rules participants should apply to field values before submission.
Transformation Rules Version	Identifies the version of the data transformation rules participants should apply to field values before submission.
Aggregation Rules Identifier and Version	Identifies the aggregation rules, if any, the participant should perform on the data prior to submission in a given jurisdiction, along with the aggregation ruleset version.

Data Normalization

In order to minimize processing errors resulting from data incompatibilities, all participants will be expected to normalize field values according to standard rules prior to submission. The rulesets will provide per-field requirements for each field in the submission record.

Data Validation

In order to minimize processing errors resulting from data errors, all participants will be asked to perform validation checks of gathered data using standard rules and make corrections as appropriate prior to submission. The checks will validate fields and records meet the platform submission record.

Data Transformation Process

Concerns have been raised about the potential for privacy signal strings being used to gain information about users (a review of the platform data requirements found that no other fields had this potential). To address these concerns, it was suggested that privacy signal strings could be transformed to remove uniqueness or replace them with non-transferable versions.

Two categories of transformations are envisioned:

1. **Redaction** - rules for redacting privacy strings would be provided, for replacing them with boolean flags describing their attributes and a combination of both.
2. **Hashing** - a hash value is generated from the original privacy string and chain ID concatenated together. The chain ID is included to make the resulting hash comparable across the interactions in a single chain, while also making it unique to that set of interactions and unusable as a means of identifying the privacy string in other contexts. The hash function chosen would be used by all participants so that they produced the same outputs given the same inputs. The hashed result would be usable as a means of determining whether Senders and Receivers were providing the same inputs to the function and therefore working with the same signal, but they wouldn't be directly usable to look up the user's privacy signal string in other data sets. If it was determined that additional privacy safeguards were required, the number of bits output by the hash function could also be limited enough so its collision rate guaranteed brute force efforts would only indicate a probability of a match between a given privacy signal string and hash value.

In order to guard against potentially revealing data being submitted, all participants will be expected to transform data according to standard rules prior to submission. The rulesets will provide per-field requirements for fields in the submission record. Only transformations to the privacy signal strings are expected to be needed.

Data Aggregation Process

There is an ongoing effort to enhance the submission of event-level data to the Common Operator while preserving privacy. As a solution, various methods of data aggregation, in conjunction with data transformations applied to privacy strings, provide participants with valuable statistical insights into the effectiveness of privacy signal propagation without risking the exposure of event-level information.

In order to ensure that inputs to aggregations are consistent, participants will be expected to follow the previously defined process for gathering an event-level data sample and then

aggregate those results as specified in the aggregation ruleset identified by the aggregation identifier and version in the data collection request. The aggregation will generate statistics based on subsets of records in the sample and may include rules for removing subsets with too few members. The aggregation results will then be what is submitted to the Common Operator.

Since the aggregations are based on event-level samples generated using the same process as the samples submitted by participants who provide event-level data, the Common Operator will be able to perform aggregations necessary to enable appropriate comparison in cases where one party to transactions provides event-level data and the other provides aggregated data.

The aggregation ruleset will define an aggregation key and associated counts or other statistics. The key likely being a combination of senderId + receiverId + time period (e.g. minute, 10 minutes or hour) + transformed privacy signal. Alternatives to providing statistics have also been proposed, such as the use of bloom filters generated from subsets of privacy signal strings.

It is assumed that different jurisdictions may have different aggregation requirements, so each aggregation rulesets will have a unique identifier. In addition, each will be versioned, with the version being incremented when material changes are made to them. The Common Operator will maintain all historical aggregation ruleset versions for each identified aggregation.

Data Collection Request

The Common Operator will initiate data gathering by posting a read-only JSON file containing parameters for the submission to a well-known location. The JSON file will include the following information:

Field	Description
Version	Identifies the version of the job request JSON file.
Common Operator Identifier	A value that uniquely identifies the Common Operator issuing the data collection request. It is assumed there may need to be region-specific Common Operators so each is assigned a unique identifier.
Request Identifier	A value that uniquely identifies the request and is used by the Common Operator to manage collection cycles.
Submission Deadline	Deadline for submissions in unix epoch time. The Common Operator will only accept submissions until the deadline is reached. At the deadline the final processing of the cycle will be initiated and no new data will be accepted.
Period Start	The start of the period for which data is being requested in unix epoch time.

Period Window	The number of seconds in the period covered by the request. A participant will be expected to include records where the transaction time converted to unix epoch time is greater than or equal to Period Start and less than or equal to Period Start + Period Window.
Match Value for Sample	A randomly selected integer value that is used to determine the subset of records to be included in the submission. See the section: Match Value for Samples Application.
Maximum Records	The maximum number of records a participant should include in a submission. Note that Maximum File Size (below) takes precedence over this value.
Maximum File Size	The maximum file size in bytes a participant should submit. The submission should be bounded by the lower of this field or Maximum Records, above. If maximum file size is exceeded, records should be removed until the size is below this limit.
Submission Requirements Identifier/Version pairs	A comma separated list of identifier and version pairs that identify the different file/record formats the Common Operator accepts. Participants will choose one for the submission they make based on the request.
Submission Endpoint	URL for a common API endpoint called by each participant to coordinate the submission of their data.

Participants will use the data collection request parameters to query their logs and create an output record for each entry for a covered transaction that matches the submission criteria. The format of the record is described below under Output Record Format.

Match Value for Samples Application

In the AdTech industry, where transactions occur on a massive scale, achieving comprehensive monitoring of privacy signal propagation for completeness and correctness can be challenging. Instead we must gather a representative sample with which we can make meaningful inferences of a general nature and through which we can identify indications of potentially significant issues which can be further investigated.

The strategy employed within the Accountability Platform is based on the following requirements:

- The sampling method must support the capture of complete, end-to-end transactions so that signals at the head of a transaction chain can be compared to signals at its tail.
- The sampling method should not allow participants to anticipate what subset of transactions will be included in the sample request and potentially treat them differently.

- The sampling method should allow for capping of the number of records each participant is asked to contribute so that data levels can be managed.

In order to support these requirements, the Common Operator will include a Match Value for Sample in requests which is a randomly generated integer value that participants will use to identify a consistent subset of previously collected records to be submitted.

To use the Match Value for Sample, participants pass the chain IDs in logged records into a common hashing function which outputs an integer value. They then compare the low-order bits of the function output with the Match Value for Sample and where they match, the record is included in the sample. Using this means of identifying the sample ensures that Senders and Receivers are including the same subsets of records in their submissions without knowing ahead of time what records will be included. It also provides the Common Operator a rough means of controlling the sample size by adjusting the Match Value for Sample: a Match Value for Sample of “1” returns 10x more records than match value of “10”, which returns 10x more records than a match value of “100”, etc.

Controlling Sample Size

There are three means by which the sample size can be adjusted to keep the overall data set size manageable. The first two are the size of the Period Window in seconds and the magnitude of the Match Value for Sample. Increasing the former will increase the number of records in the sample for a given Match Value for Sample and increasing the latter will decrease the number of records for a given Period Window.

A third limit is provided by the Maximum Records request parameter which indicates the maximum records any participant should submit. In cases where participants end up with more than Maximum Record records in their sample, they will sort the sample by chain ID to maintain alignment with submissions provided by other participants and submit the first Maximum Records records.

Data Collection Process

When participants retrieve a new data collection request, they execute a query which selects all log entries where the transaction time is greater than or equal to Period Start and less than or equal to Period Start + Period Window and where the low order bits of the hash of the chain ID match the Match Value for Sample, orders them by chain ID and outputs the top Maximum Records.

Output Record Format

The following will be output for each record in the query result. For event-level submissions, this will be the format of records provided to the common operator, for aggregate submissions it will be the input record format.

Field Name	Description	Type	Values
`version`	Record version. Indicates the version of the record format; it starts at 1 and is incremented when there are material changes to the record or fields which impact processing.	CHAR	1
`timestamp`	Transaction time. Time of the transaction as unix epoch time in seconds.	VARCHAR	10 digits
`senderId`	Sender identifier. Non-repudiable, eTLD+1 domain of the entity sending or otherwise providing data.	STRING	Plain string eTLD+1 of the Sender
`receiverId`	Receiver identifier. Non-repudiable, eTLD+1 domain of the entity receiving or otherwise acquiring data.	STRING	Plain string eTLD+1 of the Receiver
`transactionRole`	Transaction role. Flag indicating whether the entity providing the record was acting as Sender or Receiver.	CHAR	`0` for Sender `1` for Receiver
`transactionId`	Transaction identifier. A unique, per-transaction identifier generated by Senders and logged by Senders and Receivers.	VARCHAR	A 9 character alphanumeric value provides over 100 trillion combinations.
`chainId`	Chain identifier which is used to identify all of the transactions related to a given covered event. A UUID generated by the original Sender in a transaction chain. If a chainId is received, it should be included in subsequent transactions. If a chainId was not received, one should be generated.	VARCHAR	32 hexadecimal characters with four hyphens: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX
`privacySignal`	Privacy signal. Tech Lab supported privacy signals provided by the Sender, including TC string or GPP string. If no Tech Lab supported signal is present, any other signal provided by the Sender should be included.	STRING	Privacy string or “apnone” when no privacy signal is available.
`transformed`	Transformation flag. This flag indicates whether the party submitting the data applied any transformations to the privacy signal.	CHAR	0 for No 1 for Yes

Data Submission

Once the data to be submitted has been gathered and prepared, the following submission process will commence.

1. The participant will package the data based on the requirements defined in the submission requirement ruleset identified in the data collection request. The submission requirements will include things related to packaging data for submission, like the file and record formats the Common Operator accepts, compression supported, maximum file size, whether multi-part submissions are allowed and if so how they are identified, etc.

   It is assumed that Common Operators may support more than one set of submission requirements, so submission requirements rulesets will have a unique identifier. In addition, each will be versioned, with the version being incremented when material changes are made to them.
2. The participant will call the submission endpoint provided in the data collection request to indicate it has data to submit. The participant will include the data collection request information originally received from the Common Operator as well as information about what it intends to submit.

Field Name	Description
`version`	The version of the job request JSON file
`operatorId`	Common operator identifier from the data collection request provided by the Common Operator.
`requestId`	Request identifier from the data collection request provided by the Common Operator.
`subDeadline`	Deadline for submissions from the data collection request provided by the Common Operator.
`matchValue`	Match value for sample from the data collection request provided by the Common Operator.
`numRecords`	The number of records in the submission. This should not exceed the maximum number of records from the data collection request provided by the Common Operator.
`fileSize`	The file size for the submission in bytes. This should not exceed the maximum file size from the data collection request provided by the Common Operator.
`subFormat`	Submission format and version pair for this submission which should be one of the formats identified in the data collection request "Submission Requirements Identifier/Version pairs" field

3. The submission endpoint will validate that the data collection request information is correct.
    a. If not correct, the submitter will be provided with information about why the submission cannot be accepted and the interaction ended.
4. The submission endpoint will analyze the information the submitter provided about what they are submitting and return to the submitter information to be used in uploading the data, including things like an upload endpoint to be used and parameters to be provided to the endpoint. It will also provide a submission ID that can be used by participants in queries sent to the common operator about a submission.
5. The participant will contact the upload endpoint, providing it with the parameters received from the submission endpoint.
6. The upload endpoint will validate the information and then instruct the submitter to upload the data about the submission that was provided to the submission endpoint. In addition, the submitter will provide an endpoint to which data ingestion results can be posted.
7. The upload endpoint will instruct the submitter to upload the data for the submission.

Common Operator Processing, Provisioning and Reporting

After participant data is uploaded successfully, the Common Operator will prepare and stage it for final processing. Once the submission deadline for the data collection request has passed, the Common Operator will prepare and stage any remaining submissions and then perform final processing on the data that was successfully staged. The results of the final processing will then be packaged and made available for download along with a summary report providing information about the collection cycle, including statistics related to what was submitted, what was processed and what was output.

In cases where participants miss submission deadlines, no attempt will be made to recover and their failure to successfully submit the data will result in all records sent to them or received from them in the covered period being reported as orphaned. The intent of the platform is to provide feedback about chronic issues that show up across a number of submissions or participants in the value chain and not to focus on discrete or transient issues.

Common Operator Processing
The Common Operator will process submitted data in two cycles: an initial ETL cycle in which data is validated and staged as it is received and a final join cycle which will be initiated after the submission deadline is past and any remaining submissions have been staged.

ETL Cycle

Once a submission has been successfully uploaded, it will be queued for ETL processing. The ETL process reads each submitted data record, performs validation checks and outputs the result. If all steps are completed successfully, the record is added to a staging repository in a format appropriate for final processing. If errors are encountered, information sufficient to

identify the record that had the problem and indicate what the problem was is output to an exceptions file.

When the ETL processing is complete, results are posted and provided to the data ingestion results endpoint provided by the submitter when the submission was uploaded.

Final Join Cycle

Once the submission deadline has passed and the submissions ETL queue is emptied, the Common Operator will process the final join. During this process, records provided by Senders will be matched with records provided by Receivers. The matching criteria will vary depending on the record format submitted data and whether it is event-level or has been aggregated.

Event Level Record Join

Event level records will be joined by matching senderId, receiverId and transactionId in records where the transactionRole is Sender with the same fields in records where the transactionRole is Receiver.

For each pair of matched records, the following will be output:

Field Name	Description	Type
`requestId`	Request identifier from the data collection request provided by the Common Operator. Included so records can be tied back to the original request for which they were submitted.	STRING
`senderOperatorId`	Common operator identifier from the data collection request provided by the Common Operator to the Sender. Included so records can be tied to the operator the Sender submitted data to.	STRING
`receiverOperatorId`	Common Operator identifier from the data collection request provided by the Common Operator to the Receiver. Included so records can be tied to the operator the Receiver submitted data to.	STRING
`senderId`	Matched Sender identifier.	STRING
`receiverId`	Matched Receiver identifier.	STRING
`transactionId`	Matched transaction identifier.	VARCHAR
`isMatch`	Is a match. A flag indicating whether the chainId, privacySignal and transformed values provided by the Sender and the Receiver match.	CHAR
`snd_version`	Sender: record version.	CHAR

Field Name	Description	Type
snd_timestamp	Sender: transaction time.	VARCHAR
snd_chainId	Sender: chain identifier.	VARCHAR
snd_privacySignal	Sender: privacy signal.	STRING
snd_transformed	Sender: transformation flag.	CHAR
rcv_version	Receiver: record version.	CHAR
rcv_timestamp	Receiver: transaction time.	VARCHAR
rcv_chainId	Receiver: chain identifier.	VARCHAR
rcv_privacySignal	Receiver: privacy signal.	STRING
rcv_transformed	Receiver: transformation flag.	CHAR

For each unmatched Sender or Receiver record, records with the original fields will be output:

Field Name	Description	Type
requestId	Request identifier from the data collection request provided by the Common Operator. Included so records can be tied back to the original request for which they were submitted.	STRING
senderOperatorId	Common operator identifier from the data collection request provided by the Common Operator to the submitter. Included so records can be tied to the operator the data was submitted to.	STRING
version	Record version.	CHAR
timestamp	Transaction time.	VARCHAR
senderId	Sender identifier.	STRING
receiverId	Receiver identifier.	STRING
transactionRole	Transaction role.	CHAR
transactionId	Transaction identifier.	VARCHAR
chainId	Chain identifier.	VARCHAR
privacySignal	Privacy signal.	STRING

Field Name	Description	Type
`transformed`	Transformation flag.	CHAR

In addition to records output, statistics reflecting the results of the join operation will be generated and will include values like: records processed, successful joins, successful joins that match, successful joins that do not match, unmatched Sender submitted records, unmatched Receiver submitted records.

Aggregate Level Record Join

The details of the join process for aggregated data submissions will vary depending on how the data has been aggregated prior to submission, however, in all cases the join key will be the senderId and receiverId. In some cases it may also incorporate the timestamp or a minute or hour of day, etc., again depending on how the data is aggregated.

Event To Aggregate Conversion

There may be cases in which the Common Operator receives submissions with event-level records that overlap with submissions containing aggregated records. In cases like this, the Common Operator would perform aggregations of event-level data using the same rules as were applied to the aggregated submission and then perform an aggregate level record join.

Results Provisioning

Once the final join cycle is complete, the Common Operator will provision the results in a publicly accessible location. It will also post a report of the results to a standard location along with information about from where and for how long the results data will be available for download. The report of results will also be added to a publicly accessible historical record maintained in a standard location by the Common Operator.

Results Reporting

As indicated under Results Provisioning, a report of the results of the data collection process will be posted to a standard location. The report will include operational statistics like: number of submissions, number of records, number of errors, processing time, etc.

The platform could also provide statistics about the data, like: number of Senders and Receivers, number of transactions that did and did not include a privacy signal, number of transactions where Sender data didn’t match Receiver data, number of transactions where privacy signals for otherwise matching records were not the same, etc. However, this information could also be generated using the data available for download, so decisions on whether the Common Operator will provide them directly will be deferred.