

# Does DAP need two query modes?

Tim Geoghegan  
PPM - IETF 117 - San Francisco

# Recap: DAP query types

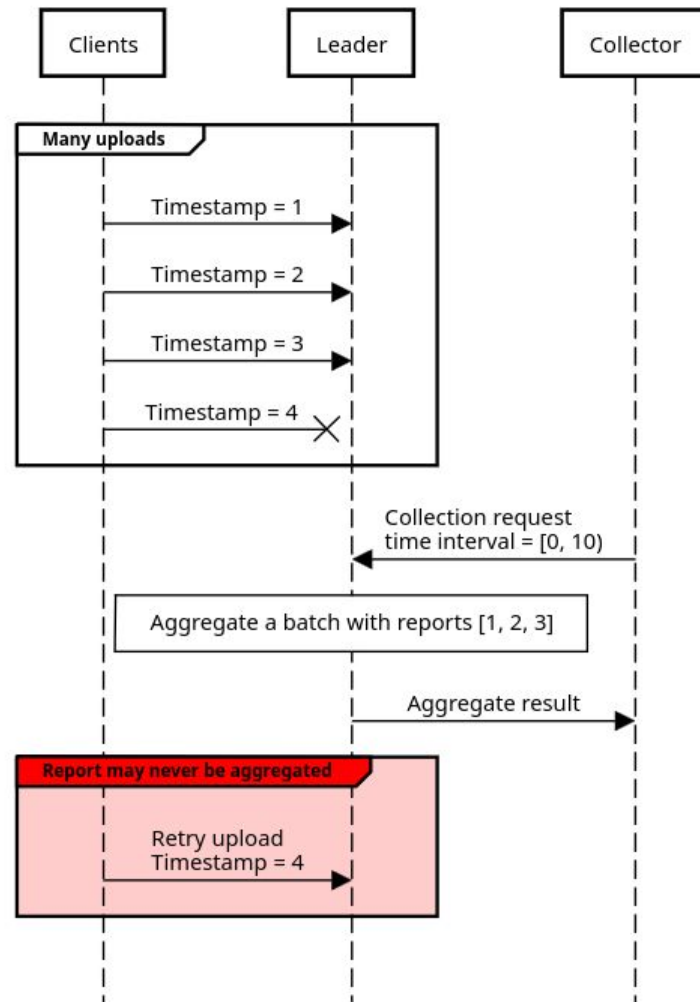
- DAP tasks must set a *query type*. Currently two types exist.
- **Time interval**
  - e.g. Aggregate over all reports whose timestamps are in the interval [09-09-2023-000000, 09-09-2023-235959)
  - Batch identifier is the interval of time
  - Either aggregator can independently verify whether some report belongs to a batch interval
- **Fixed size**
  - Task parameters include a desired batch size (e.g., aggregate over every 1,000 reports)
  - Batch identifier is a numeric ID.
    - Batch ID 1 is the first 1,000 reports, batch ID 2 is the next 1,000 regardless of their timestamps
  - Leader assigns reports to batches as it pleases
  - Batches may have overlapping time intervals
- In either case, collection result includes both the aggregate and the interval of time spanned by constituent reports

# Recap: anti-replay in the collection sub-protocol

- DAP forbids aggregating over fewer than `min_batch_size` reports
  - Aggregating Prio3Sum over 1 report would reveal that report's value
- Some VDAFs allow aggregating the same batch multiple times with different aggregation parameters
  - e.g., successively longer string prefixes in Poplar1
- DAP requires that the batch membership not change across queries
  - Otherwise collector could query over a batch  $B$  with aggregation parameter  $p_1$ , then again over  $B \cup r$  with aggregation parameter  $p_2$  and learn something about report  $r$
- The first time aggregators serve a collection request (Leader) or aggregate share request (Helper), they must commit to report membership in a batch

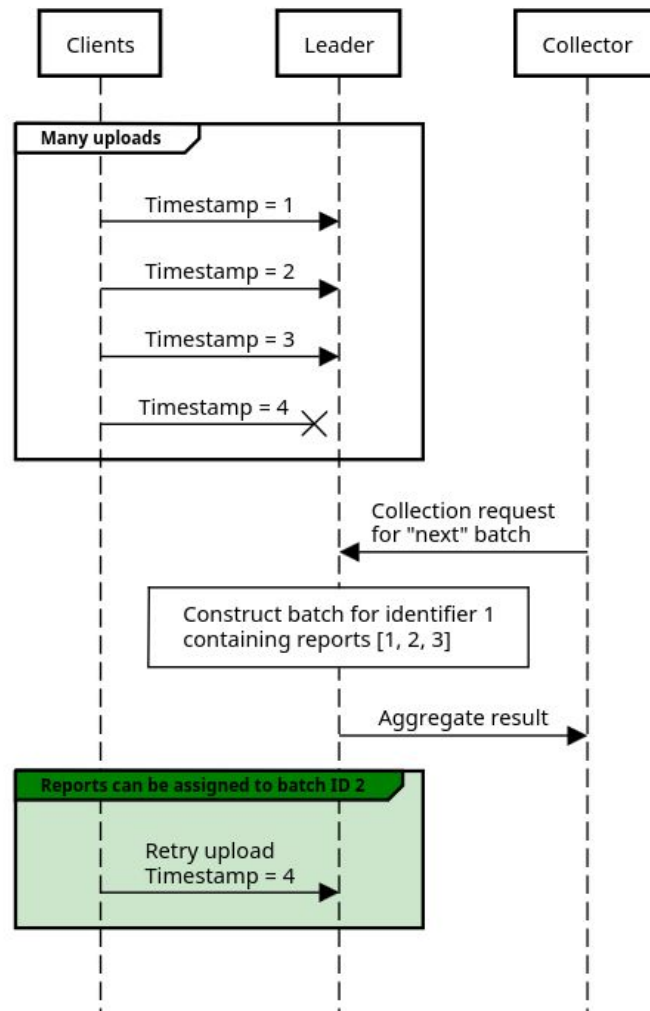
# Orphaned reports in time interval queries

- Reports may be uploaded by Clients late or out of order for myriad reasons
- Any reports that arrive after Collector makes collection request can never be aggregated
- No upper bound on number of orphaned reports
- No way for collector to know when is a "good" time to collect



## Orphaned reports in fixed-size queries

- No harm if Collector makes a collection request "too soon": late arrivals can be assigned to a subsequent batch
- Aggregate shares delivered to Collector contain the spanned time interval
- Number of orphaned reports is bounded by `min_batch_size`, which the Collector often controls



# Is fixed-size just better than time interval?

- Given the data loss risk, would anyone ever choose time interval?
- Uploads may occur at a steady rate, such that a well chosen fixed-size batch size can approximate a time interval
  - e.g., for a known number of clients, a deployment can expect some number of uploads per day, set that as their batch size and thus get batches approximately daily
- DAP does not specify how Leader assigns reports to batches
  - Leader and Collector can agree to align batches to e.g. days of the week with no protocol support

# Upsides of eliminating time interval queries

- Admits better collection API semantics
- Simplification of the protocol
  - Could allow deleting 5% of DAP-05 text
- **MASSIVE** simplification of implementations
  - Would allow deleting 10% of code in Janus
  - Removes expensive database queries across time intervals
    - Much easier to do in a key-value datastore
  - Removes error-prone time math
  - Collection requests are much easier to parallelize
  - Collector no longer needs to try bigger and bigger time intervals until `min_batch_size` is met

# Questions for the working group

- Do you have a use case where *only* time interval queries can work?
  - Please come tell us!
- Should DAP remove time interval queries?
- Should DAP remove the notion of query types?
  - Or make query types optional?