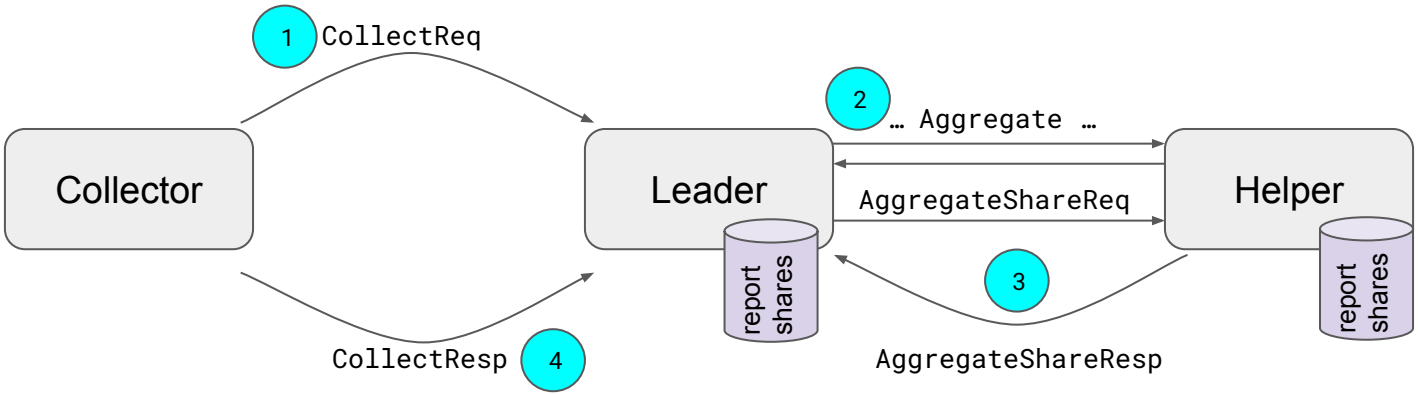
Collect Constraint Requirements

PPM - IETF 113 - Vienna

Collect Flow

1. Collect flow is initiated by Collector with `CollectReq` message indicating an interval (time window start, and time window length)
2. Leader and Helper perform aggregation (if not done already)
3. Leader and Helper produce aggregate shares, encrypted to the Collector
4. Collector polls for the aggregate output from the Leader



Collect Goals (informally)

Correctness: Aggregation shares must include shares of the same reports, otherwise the output will be garbage

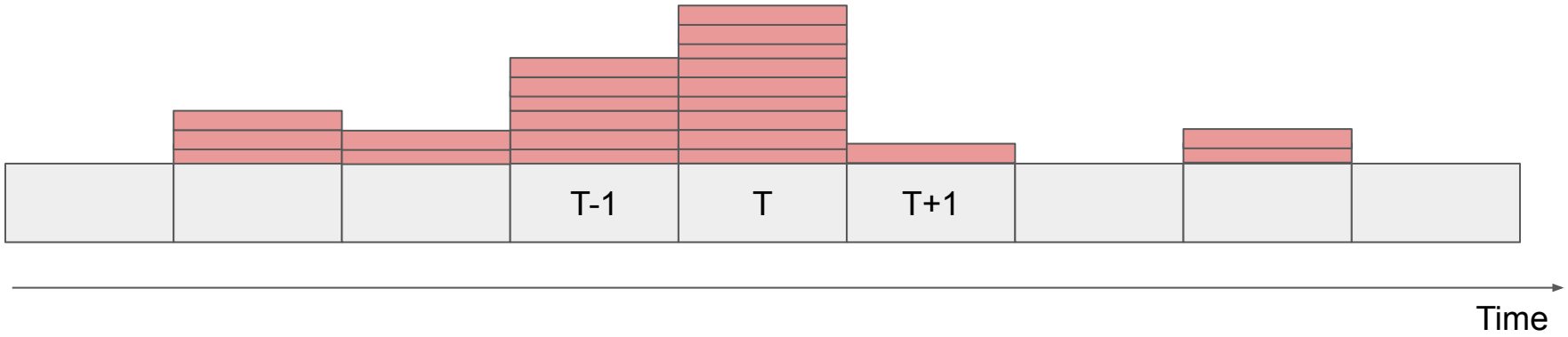
Privacy: Aggregation must ensure that at least **min_batch_size** reports in them

Report Batches and Time Windows

Batch windows are divided into epochs of size **min_batch_duration**

Collect requests are parameterized by a start and end time window, each aligning on a **min_batch_duration boundary**

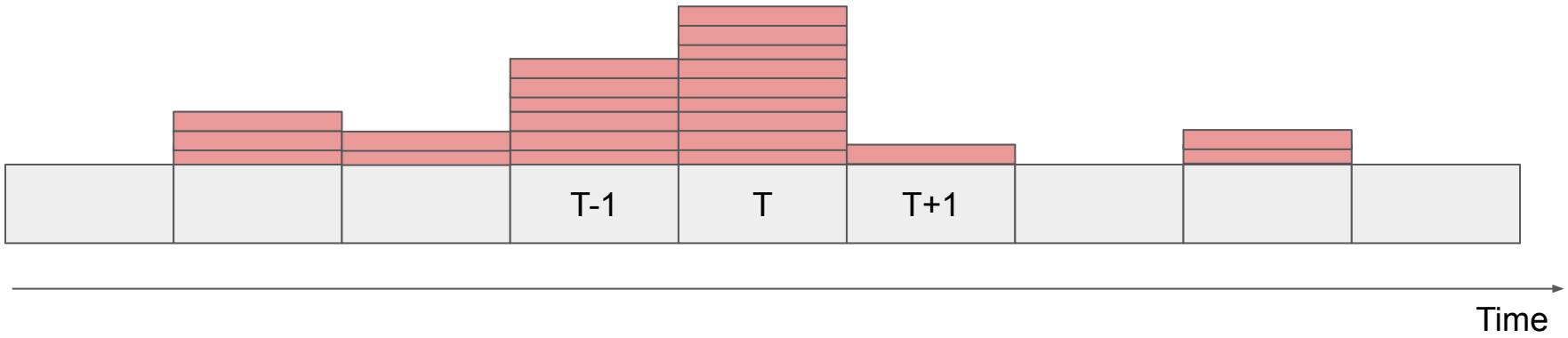
Each time window corresponds to a set of **reports**



Current Requirements for Collect Requests

Collect request validation:

1. Check that the interval is valid, i.e., aligned with **min_batch_duration** bounds
2. Check that the number of reports in the window is at least **min_batch_size**

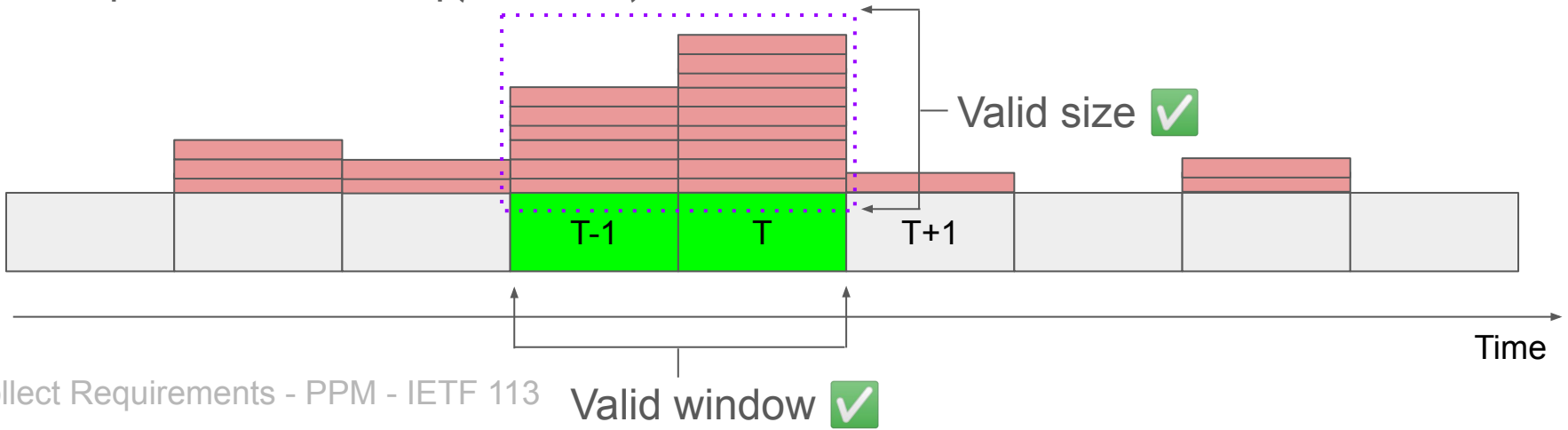


Current Requirements for Collect Requests

Collect request validation:

1. Check that the interval is valid, i.e., aligned with **min_batch_duration** bounds
2. Check that the number of reports in the window is at least **min_batch_size**

Example: `CollectReq(T-1, T)`

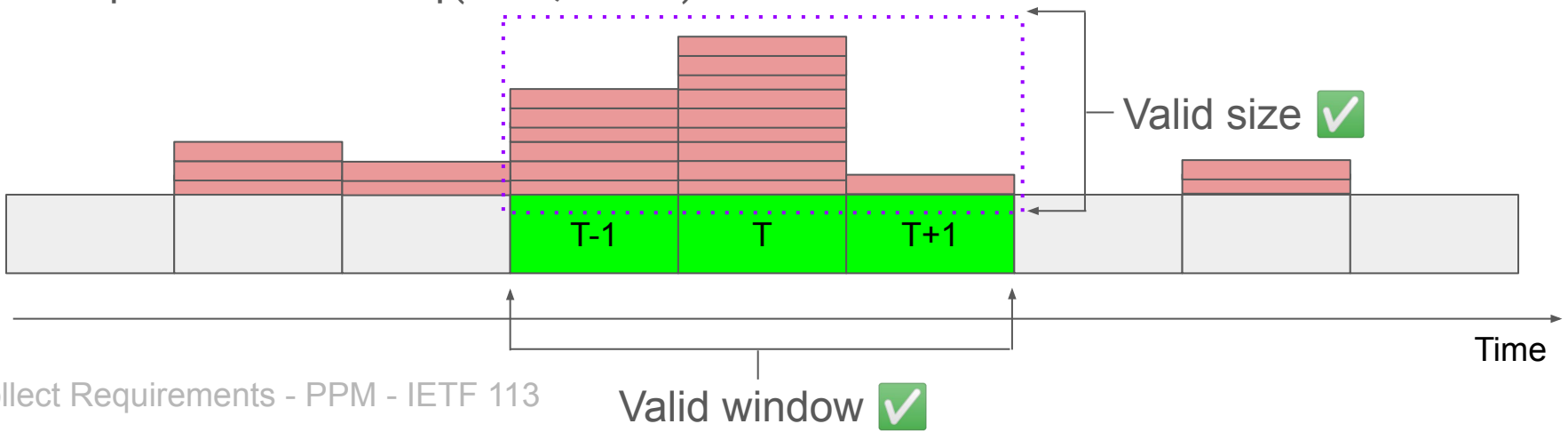


Current Requirements for Collect Requests

Collect request validation:

1. Check that the interval is valid, i.e., aligned with **min_batch_duration** bounds
2. Check that the number of reports in the window is at least **min_batch_size**

Example: `CollectReq(T-1, T+1)`

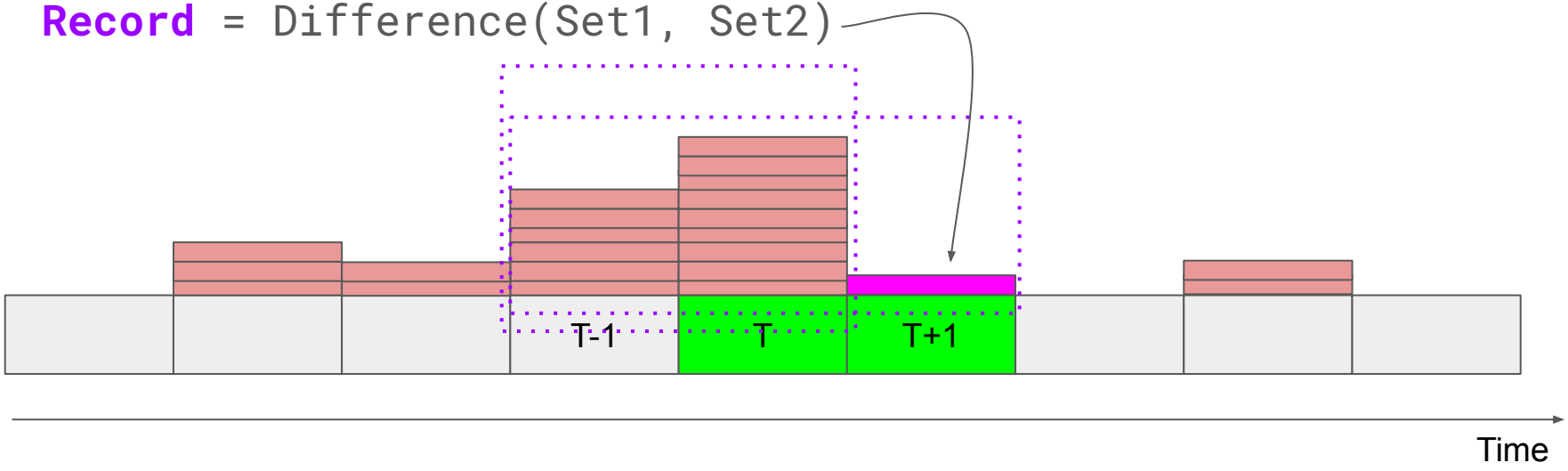


Privacy Violation

```
Set1 = CollectReq(T-1, T)

Set2 = CollectReq(T-1, T+1)

Record = Difference(Set1, Set2)
```



Query Dimensions

Beyond restricting queries in time, some may want to restrict queries in space

- “Give me the aggregate of all queries in this time window that came from clients with this specific User-Agent string”
- “Give me the aggregate of all queries in this time window that came from clients in this geographic region”

Currently, the Collect flow is only parameterized in *time*, not in *space*

Privacy Requirement and the Collect Constraint

Informally: given any sequence of collect requests with corresponding *time* and *space* constraint parameters, it MUST NOT be possible for the Collector to compute an aggregate output based on some subset of reports of size less than **min_batch_size**

Enforcement intuition: Given a CollectReq and sequence of preceding CollectReq messages, each Aggregator ensures that the size of all possible subsets that can be computed based on the union of CollectReq messages is at least **min_batch_size**

This is fairly easy to implement if space is bounded and contiguous, but would require Aggregators to know about the space dimension

Questions

1. Is the collect validation problem clear?
2. Is the proposed requirement clear?
3. Does the collect flow need to be parameterized by time and space?
    a. If both, how are reports marked with space? (They currently only carry timestamps.)
4. Should the collect validation algorithm be specified, or should aggregators rely on collect *correctness* to ensure that **min_batch_size** is honored?