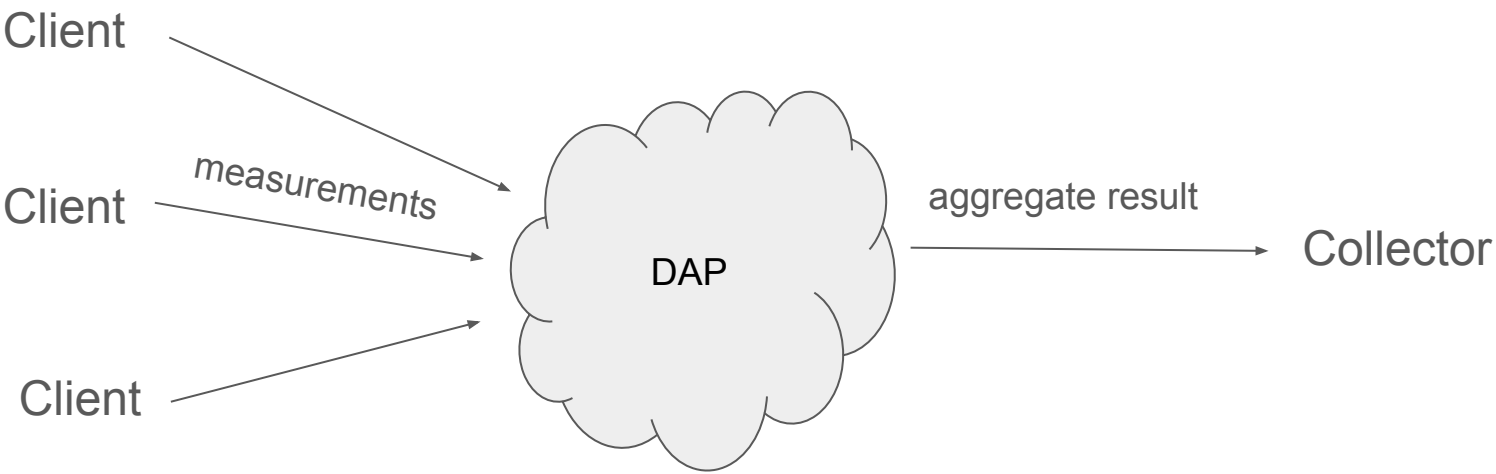
DAP feature set

interim-2025-ppm-01 – Chris P.

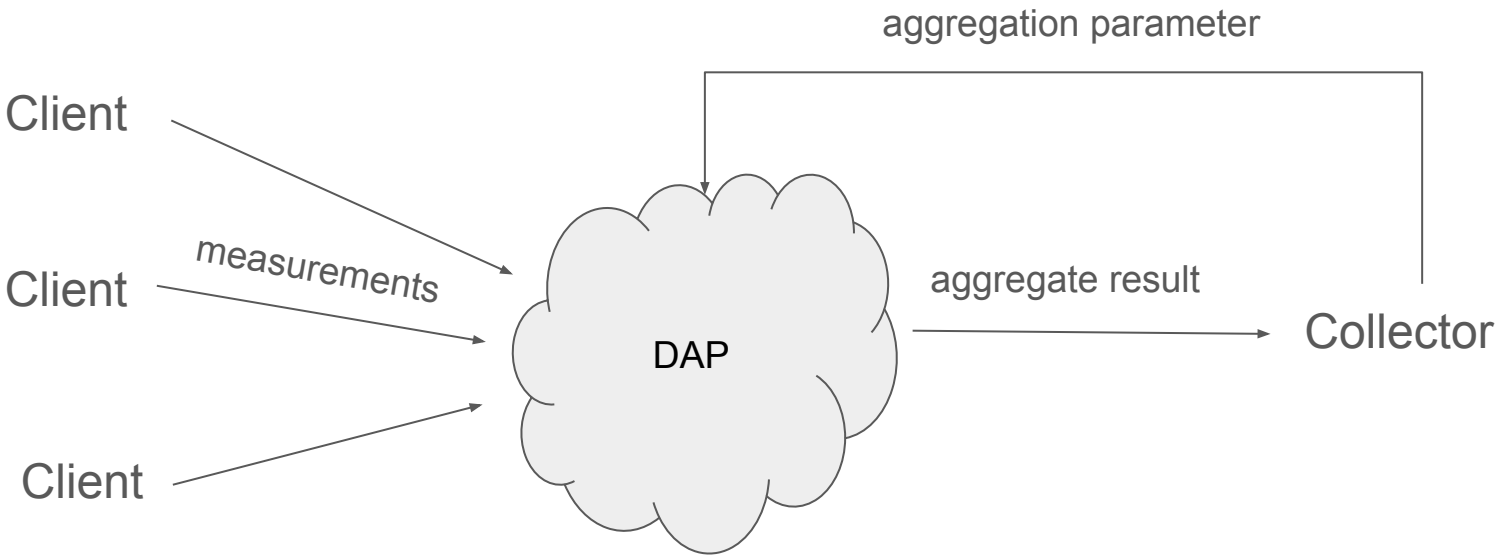
Use cases

- Pre-BoF (~2022): Client-side telemetry
  - Counts, sums
  - Mean, median, variance, histograms
  - Sketching, heavy hitters
  - Browser telemetry, web analytics, NEL, …
- Post-BoF
  - Federated machine learning
  - PPA: On-device ad-conversion measurement

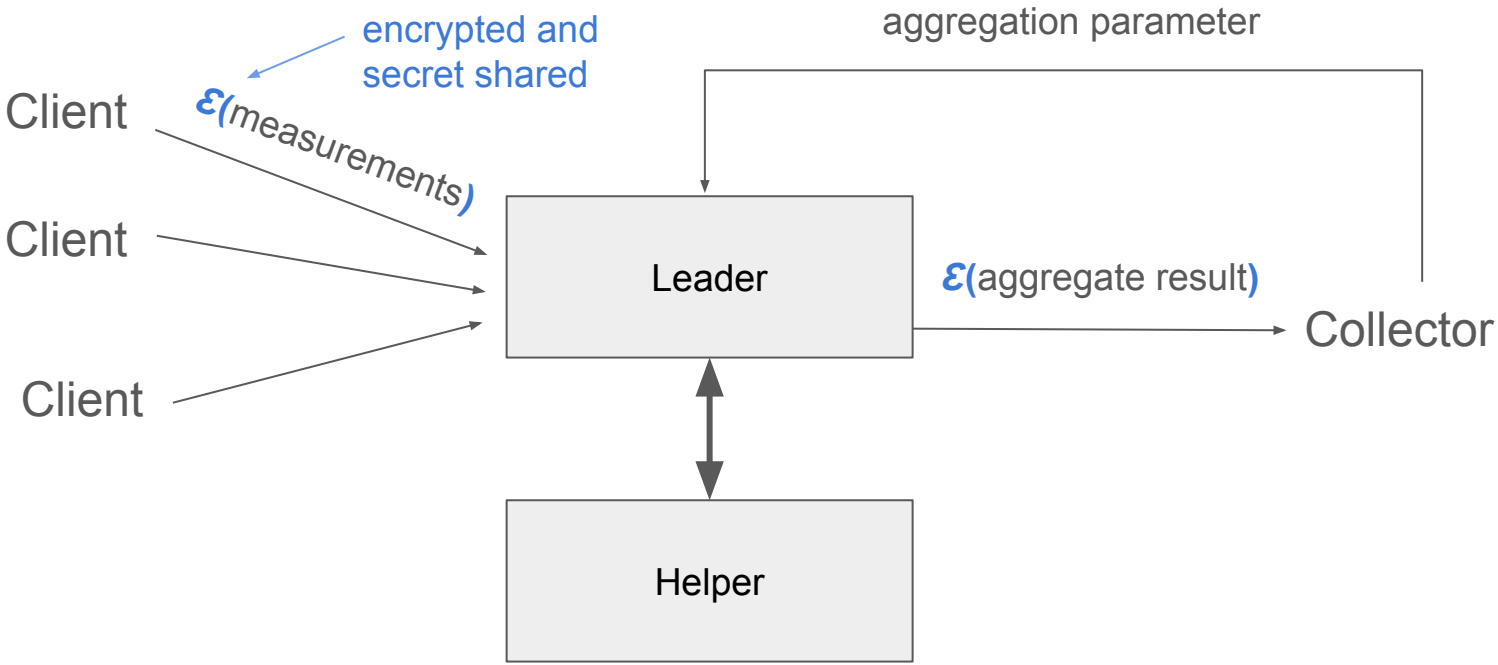
agg_result = *F*(measurements)



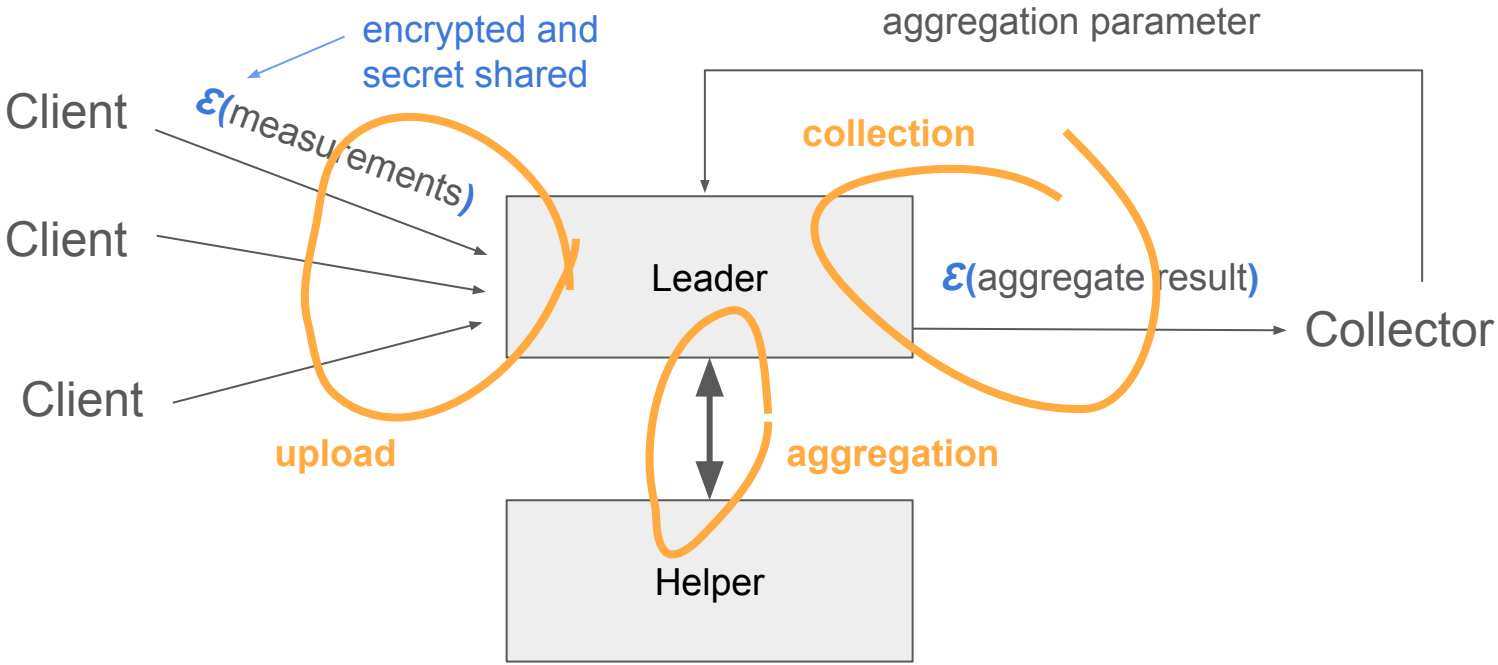
agg_result = *F*(agg_param, measurements)



Securely computing *F*



Interactions



Task — config for upload, aggregation, and collection

- Who are the Leader and Helper
  - identity of the Clients and Collector are out of scope
- VDAF: Defines the aggregation function *F* that is computed
- Batch mode: Defines how reports are partitioned into batches
- Cryptographic assets including HPKE keys and the VDAF verification key

Security goals

- Secure VDAF execution
  - Privacy if one Aggregator is honest
  - Robustness to malicious Clients when both Aggregators are honest
- Task agreement
  - Everyone agrees on the task ID
  - orchestration of tasks out of scope
- Replay protection
  - each report is aggregated at most once
- *Non* goals:
  - Differentially privacy
  - Sybil-attack resistance
  - …?

Bring your own VDAF

- Prio3: General-purpose aggregation
  - Specify your data type as an arithmetic circuit (e.g., bounded L1-norm for PPA)
- Pine: federated learning
  - Real-valued vectors with bounded L2-norm
    - Can do the same thing with Prio3, but not as efficiently
  - Gradient descent over multiple batches
- Mastic: Attribute-based aggregation
  - Prio3, but with the ability to group by client properties without losing privacy
  - Approximate heavy hitters
- Room for other cryptographic techniques
  - Data refinement via function secret sharing
  - Arithmetic sketching (Poplar1) is sometimes a better choice than an FLP
  - Implicit validation via boolean-to-arithmetic conversion (Prio+)
  - …?

Define your own batch mode

- Batching is application specific
  - `time_interval` envisions long-running telemetry systems
    - think Prometheus but with DAP!
  - `leader_selected` maximizes the flexibility for the entity orchestrating tasks
    - Task orchestrator coordinates with the Leader
- Future drafts may define new batch modes
  - ...?

Report extensions

- Modify report processing rules (affects upload and aggregation interactions)
  - draft-ietf-ppm-dap-taskprov: Enhances task agreement by cryptographically binding the task parameters to the task ID
  - draft-priebe-ppm-dap-reportauth: Helps mitigate Sybil attacks by binding a Privacy Pass token to a DAP report
  - draft-thomson-ppm-dap-dp-ext: Enforce DP privacy budget
  - …?

New APIs

- New HTTP APIs may be defined for the Leader and Helper
  - draft-dcook-ppm-dap-interop-test-design: Defines APIs for configuring tasks and driving interop tests
  - …?

Features we're not using (yet)

- Aggregation jobs with multiple steps (multi-round VDAFs)
- Aggregation parameter
  - Needed for attribute-based aggregation via Mastic
- Batch modes
  - We could maybe hard-code `leader_selected`, which leaves batching policy to Leader
  - Optimizations may be possible for more restrictive batch modes
    - `time_interval` can be made to have cheaper replay protection

Rejected features

- Methods of secure aggregation beyond VDAFs
  - general purpose MPC
  - "Asymmetric" VDAFs (validation is done by one Aggregator)
  - Shuffling
- Multiple Helpers
  - Weaker trust model (Allow more than one Aggregator to misbehave)
  - Resilience to Helper drop-outs
- Multiple collection jobs for the same batch
  - Originally envisioned for exact heavy hitters via Poplar1
    - Too much protocol complexity, uncertain security
- Batched VDAF preparation
  - Can easily be specified with new aggregation API