# Resource oriented DAP API

Tim Geoghegan

PPM - IETF 115 - London

# DAP-02 HTTP API surface

- `GET [aggregator]/hpke_config[?task_id={task-id}]`
- `POST [leader]/upload`
- `POST [helper]/aggregate`
- `POST [helper]/aggregate_share`
- `POST [leader]/collect`
- `GET {leader collect URI}`
- `DELETE {leader collect URI}`

# DAP-02 HTTP API surface

- `GET [aggregator]/hpke_config[?task_id={task-id}]`
- `POST [leader]/upload`
- `POST [helper]/aggregate`
- `POST [helper]/aggregate_share`
- `POST [leader]/collect`
- `GET {leader collect URI}`
- `DELETE {leader collect URI}`

Nouns
Verbs

# DAP-02 API problems

- API paths are sometimes verbs, sometimes nouns
  - Should use HTTP methods as verbs acting on resources
- Most of the API is non-idempotent POST requests
- Unclear how aggregate sub-protocol participants are meant to recover from faults
- Servers need to partially parse request bodies to extract task ID
  - e.g. to find out what VDAF or query type is in use to then parse the rest of the message

# New HTTP API proposal: resources

| **Resource** | **Supported by…** | **Required methods** | **Relative path** |
|---|---|---|---|
| HPKE configuration | Leader, helper | GET | /hpke_config[?task_id={task-id}] |
| Report | Leader | PUT | /tasks/{task-id}/reports/{report-id} |
| Aggregation job | Helper | PUT, POST, DELETE | /tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Aggregate shares | Helper | PUT | /tasks/{task-id}/aggregate_shares |
| An aggregate share | Helper | GET, DELETE | Implementation defined |
| Collections | Leader | PUT | /tasks/{task-id}/collections |
| A collection (job?) | Leader | POST, DELETE | Implementation defined |

# New HTTP API proposal: resources

| Resource | Supported by… | Required methods | Relative path |
|---|---|---|---|
| HPKE configuration | Leader, helper | GET | /hpke_config[?task_id={task-id}] |
| Report | Leader | PUT | /tasks/{task-id}/reports/{report-id} |
| Aggregation job | Helper | PUT, POST, DELETE | /tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Aggregate shares | Helper | PUT | /tasks/{task-id}/aggregate_shares |
| An aggregate share | Helper | G | |
| Collections | Leader | P | |
| A collection | Leader | P | |

Creating an aggregate job is idempotent => PUT
Advancing it to the next round changes state => POST

# New HTTP API proposal: resources

| Resource | Supported | Required | Relative path |
|---|---|---|---|
| HPKE co | | | /hpke_config[?task_id={task-id}] |
| Report | Leader | PUT | /tasks/{task-id}/reports/{report-id} |
| Aggregation job | Helper | PUT, POST, DELETE | /tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Aggregate shares | Helper | PUT | /tasks/{task-id}/aggregate_shares |
| An aggregate share | Helper | GET, DELETE | Implementation defined |
| Collections | Leader | PUT | /tasks/{task-id}/collections |
| A collection | Leader | POST, DELETE | Implementation defined |

Aggregation job ID assigned by leader

# New HTTP API proposal: resources

| Resource | Supported | Required | Relative path |
|---|---|---|---|
| HPKE co | | | /hpke_config[?task_id={task-id}] |
| Report | Leader | PUT | /tasks/{task-id}/reports/{report-id} |
| Aggregation job | Helper | PUT, POST, DELETE | /tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Aggregate shares | Helper | PUT | /tasks/{task-id}/aggregate_shares |
| An aggregate share | Helper | GET, DELETE | Implementation defined |
| Collections | Leader | PUT | /tasks/{task-id}/collections |
| A collection | Leader | POST, DELETE | Implementation defined |

Message handler constructs resource URI

# Hopefully an easy migration

| **Action** | **DAP-02 API** | **New API** |
|---|---|---|
| Upload report | POST [leader]/upload | PUT [leader]/tasks/{task-id}/reports/{report-id} |
| Initialize aggregate job | POST [helper]/aggregate with AggregateInitReq | PUT [helper]/tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Continue aggregate job | POST [helper]/aggregate with AggregateContinueReq | POST [helper]/tasks/{task-id}/aggregation_jobs/{aggregation-job-id} |
| Initialize aggregate share | n/a | PUT [helper]/tasks/{task-id}/aggregate_shares |
| Get aggregate share | POST [helper]/aggregate_share | GET {helper aggregate share URI} |
| Initialize collect | POST [leader]/collect | PUT [leader]/tasks/{task-id}/collections |
| Get collect result | GET {leader collect URI} | GET {leader collect URI} |
| Delete collect result | DELETE {leader collect URI} | Same |

# Open questions and next steps

- Needs more analysis to show that error recovery works
- Does it makes sense to align aggregate share resource with collection resource?
- Collection resource is awkward
  - Is "collection" the right resource? Is "collection job" or "query" better?
  - We are trying to design one API to accommodate time interval and fixed batch queries. Is this a good goal?
- HTTP and API design experts: please share your critiques of the proposal
  - https://github.com/ietf-wg-ppm/draft-ietf-ppm-dap/pull/367