OTEL Arrow SIG

Meeting times:

- Thursday 08:00 Pacific Time, 15:00 UTC, 17:00 CEST (Zoom Link)
- Tuesday 16:00 Pacific Time,

Link to recordings: 🟩 Meeting Recordings

Project charter: https://github.com/open-telemetry/community/blob/main/projects/otelarrow.md
Repository: https://github.com/open-telemetry/otel-arrow
Slack: https://cloud-native.slack.com/archives/C07S4Q67LTF

Dec 3, 2025 - Query\Transform group

Attendees:

- Joshua MacDonald (Microsoft)
- Mike Blanchard (Microsoft)
- Albert Lockett (F5)

Agenda

- Overview of record set engine data sources, traits, constants folding
- Demonstration of record set engine diagnostics
- Alignment on near-term columnar engine execution: will continue focusing on core OTAP functionality to maintain velocity (versus building something more generic)

Dec 2, 2025 - General meeting

Attendees:

- Joshua MacDonald (Microsoft)
- Cijo Thomas (Microsoft)
- Albert Lockett (F5)
- Drew Relmas (Microsoft)
- Aaron Marten (Microsoft)
- Danny Chin (CMU)

Agenda

- Triage: 2 weeks! 1449-1508
- Note: Query/transform sub-group met, see notes!
- Note: next week plan to discuss Phase 3 with more participants in the 8am slot
- [Cijo] - are we okay to update the repo's main page to point to benchmarks from phase2? Currently it points to phase1 results.
  - Yes.
  - And also include a comparison with OTel Collector

- [Jake] - Auth extensions/identity: https://github.com/open-telemetry/otel-arrow/issues/501#issuecomment-3604015669
    - About extensions:
    - https://github.com/open-telemetry/opentelemetry-collector/pull/13902
    -
    - Context-Generic Programming
- [Danny] - about OpenZL evaluation
- Open PR discussion
    - Laurent's https://github.com/open-telemetry/otel-arrow/pull/1480
    - About "exporterhelper" https://github.com/open-telemetry/otel-arrow/pull/1490
    - TLS support https://github.com/open-telemetry/otel-arrow/pull/1510
    - SDK integration https://github.com/open-telemetry/otel-arrow/pull/1511
        - Will make use of new OTel-Rust declarative configuration code (also by Andres)
        - Finding: we don't have any attributes in the multivariate metrics crate
        - Resources: will be part of the configuration
            - (today we are hard-coding the computation of metrics)
-

Nov 26, 2025 - Query\Transform group

Attendees:

- Mike Blanchard (Microsoft)
- Laurent Querel (F5)
- Albert Lockett (F5)
- Drew Relmas (Microsoft)

Agenda:

- Discussion of PR https://github.com/open-telemetry/otel-arrow/pull/1477 columnar query engine filtering
    - Uses datafusion physical expression to produce selection vectors & arrow compute functions to combine those together
    - Main motivation was b/c of difficulty of multiple tables encompassing a single batch
- View discussion
    - Attributes as virtual columns (similar to polarsignal's dynparquet). Is it possible w/ custom Table Provider?
    - would need to implement Array for the virtual columns somehow
    - Not needed for now based on impl in #1477
- OPL Followup from MS side
    - Mission is not to tie current query language to a particular language -- having support in the AST expressions is the focus, less so transpiler of a new language
    - Discussion of record set engine - processes data that isn't in the columnar format. It can process data that's not structured as OTel. Record sets can have any schema, they're more general than just OTel records. OTLP bridge links OTel logic to the record set engine.

Nov 19, 2025 - Query\Transform group

Attendees:

- Mike Blanchard (Microsoft)
- Laurent Querel (F5)
- Albert Lockett (F5)
- Andres Borja (Microsoft)
- Luke Steensen (DataDog)

Agenda:

- Introductions
- Status updates
- OPL Overview from Laurent
  - KQL inspired stream processing language, strongly typed, and guarantees valid OTel signals is the output.
- Brief description of learnings from Columnar Query Engine POC from Albert
  - Challenges with single DataFusion ExecutionPlan: Operating on multiple batches and confusing join order for filtering attributes
  - To address challenges - pipelines are multi-stage, not a single ExecutionPlan
- Discussion about OPL & pushing the pipeline stages to multiple pipeline components
  - Initial plan: start with processor, but keep in mind distributing certain pipeline stages. The motivating example is predicate pushdown (filtering in the processor).
- Alignment on our own Abstract Syntax Tree
  - We agree this is the right approach for multiple reasons: easier support for various languages, and b/c we'd like to have another layer on top of the DF logical plan (considering our query plans will be made up for multiple stages that are not necessarily a single ExecutionPlan).
- Datadog interest:
  - Improved bandwidth & filtering/processing exploration
- Discussion of OTel-Arrow vs other formats - tradeoffs

Nov 18, 2025

Attendees:

- Laurent Querel (F5)
- Albert Lockett (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Kennedy Bushnell (Microsoft)
- Aaron Marten (Microsoft)
- Drew Relmas (Microsoft)
- Cijo Thomas (Microsoft)
- Josh MacDonald (Microsoft)

- Danny Chin (CMU)

Agenda:

- Issue triage
    - [15m] About “Condense” https://github.com/open-telemetry/otel-arrow/issues/1435
        - About whether there is a performance principle here, or whether we are prematurely optimizing
        - Probably we want this transform “condence” to reduce number of database columns
        - About how this calls for “extension” interfaces.
        - About splitting CEF within syslog receiver, whether it’s a reusable library
    -
- Quiver Proposal Update: Multi-schema segment format
- Unify OTLP/OTAP gRPC Configuration and Introduce Experimental Non-Tonic Receiver by lquerel · Pull Request #1382 · open-telemetry/otel-arrow
    - A thread-per-core !Send approach to (replace) Tonic, an experiment.
    - More control over http/2 and gRPC layer for resource management / limits
    - Uses low-level h2 crate from tokio, implements OTAP / OTLP configurations w/o Tonic
    - Improvements preview in https://cloud-native.slack.com/archives/C08RRSJR7FD/p1763166844608099

Nov 13, 2025

Attendees:

- Matthias Loibl
- Joshua MacDonald
- Aaron Marten
- Danny Chin
- Utkarsh Umesan Pillai
- Pablo Baeyens
- Jake Dern
- Albert Lockett
- Andres Borja
- Laurent Querel (joined mid Quiver conversation)

Agenda:

- Issue triage
    - About crates.io and blockers for releasing (e.g., changelog, missing docs)
- Albert’s DataFusion planning exercise:
    - https://github.com/open-telemetry/otel-arrow/issues/1394
- Aaron’s Quiver proposal
    - https://github.com/open-telemetry/otel-arrow/issues/1416
    - About schema management: do we need one IPC log file per OTAP table?
    - Will we manage schema change the way OTAP-streams do?

    - How to manage references to on-disk segments, stream “state”
    - About how Parquet exporter addresses schema widening, sorting columns
    - LQ: Arrow has new encodings today; OTAP dynamic schemas were introduced because:
        - New encoding options: Run-end encoded arrows\
- Think about this until next time: What’s Phase 3 going to look like?

Nov 11, 2025

Attendees:

- .

Agenda:

- Issue triage
- About internal telemetry Rust SDK: It uses tokio. Prometheus exporter is no longer supported.

Nov 4, 2025

Attendees:

- Utkarsh (Microsoft)
- Josh M (Microsoft)
- Jake Dern (Microsoft)
- Albert Lockett (F5)
- Danny Chin (CMU)
- Aaron Marten (Microsoft)

Agenda:

- Issue triage
- Open discussions
    - About Internal telemetry metrics dispatcher · Issue #1378 · open-telemetry/otel-arrow
        - Goals? Are we replacing the OTel SDK
        - Or are we taking shortcuts
        - Choices:
            - Use the otap-dataflow engine pipeline for egress
            - Use the OTel-Rust SDK for egress, with views configuration
            - Use custom logic / configuration for egress
        - Potentially, use of otap_df_config structs to build internal telemetry graph
    -

Oct 30, 2025

Attendees:

- Utkarsh (Microsoft)
- Josh M (Microsoft)
- Jake Dern (Microsoft)
- Danny Chin (CMU)
- Matthias Loibl (Polar Signals)

Agenda:

- Issue triage
    - One new issue about logs filter API
- About Datafusion
    - About Datafusion https://www.youtube.com/watch?v=iJhRbDFJjbg
    - Talking about abstractions for observability query languages
        - PromQL: promql_parser - Rust
        - TraceQL
        - KQL
        - OTTL
        - Some more metrics:
            - Google “Monarch” : focus on “align” and “reduce”
            - Lightstep “UQL”
            - APL
            - OxQL
        - Note there is a Prometheus effort to store in Parquet files (SIG-Parquet)
- Josh’s question
    - Taking on the batch processor.
        - About optional ID columns in the logs signal
        - The code was being inconsistent about this detail
        - Josh will open a PR with more detailed questions outlined
- Question: on query
    - KQL, OTTL, and an intermediate representation
    - Likely: the intermediate representation is a logical plan tied with the OTAP model
    -

Oct 21, 2025

Attendees:

- Utkarsh (Microsoft)
- Laurent (F5)
- Josh M (Microsoft)
- Jake Dern (Microsoft)
- Mike Blanchard (Microsoft)
- Albert Lockett (F5)
- Danny Chin (CMU)

Agenda:

- Issue triage
    - Talk about OTLP metric export
        - About the diagram NUMA-node level metrics aggregator -> Rust OTel client SDK
        - This is the plan of action. However, we also can imagine a direct SDK for metrics to OTAP; **we will not prioritize**.
    - About multivariate metrics dreaming
        - For SDKs
        - For storage
        - For Analysis
    - About concurrent, backpressure, wait_for_result
        - These do have equivalent settings in the Collector, but its default is wait_for_result=false; we need a Collector issue to track
        - (we don't all always agree here!)
    -
- Benchmark plan
    - Continuous benchmarks (per commit) - "core" protocol matrix (otap, otlp)
    - Nightly - Additional protocols, configuration parameters of interest
    - Periodically (TBD) - Comparison against other solutions
- Laurent: define a list of engine-related tasks that require prioritization
- Demo: @c11y shows us logs filter processor
    - E.g., by severity, by attributes
- Parking lot:
    - About instrumentation of num_items()
    - About fanout-connector and how to handle backpressure when there are multiple exporters

Oct 16, 2025

Attendees:

- Albert Lockett (F5)
- Utkarsh (Microsoft)
- Laurent (F5)
- Josh M (Microsoft)
- Pablo Baeyens (Datadog)
- Jake Dern (Microsoft)

Agenda:

- Issue triage
    - Talk about OTLP metric export
    - About multivariate metrics dreaming
        - For SDKs
        - For storage
        - For Analysis

    -
- [Laurent] Graceful shutdown, metrics draining process

Oct 7, 2025

Attendees:

- Albert Lockett (F5)
- Utkarsh (Microsoft)
- Laurent (F5)
- Josh M (Microsoft)
- Danny C (CMU)
- Bill Z (Softprobe)

Agenda:

- Issue Triage
    - Discussion about Is the Map type being used correctly when producing BatchArrowRecords? · Issue #1223 · open-telemetry/otel-arrow
    - About facebook/openzl: A novel data compression framework
    - About Update size test metrics stream implementation for otel-arrow encoding by albertlockett · Pull Request #214 · splunk/stef
    - About UTF-8 validation and choices.
        - About simdutf8 and lossy string conversion
        - Also noted about partial succes https://github.com/open-telemetry/opentelemetry-collector/pull/13927#pullrequestreview-3311421686
        -
    - About Otel-Arrow compression level settings, including “payload” and “grpc” levels.
- About limits, especially memory.
    -
- Status (observed state), liveness, readiness endpoints

Oct 2, 2025

Attendees:

- Albert Lockett (F5)
- Utkarsh (Microsoft)
- Laurent (F5)

Agenda: Issue triage

- [Joshua] Hackathon learning experiences
    - Store logs in Parquet files
    - Use DataFusion to sample logs
    - Output sampled logs in valid Parquet files compatible with OTAP
    - **We need debug logging!**

    - Link: otel-arrow/rust/otap-dataflow/sampler-finished.md at jmacd/hack-a-thon-sampler · jmacd/otel-arrow
- [Albert] Brief discussion of Sept 18's KQL-to-DF draft
- [Laurent] Second milestone goals
- [Laurent] Add `Condition` in pipeline status to report errors from the controller, pipelines, add nodes

Sep 23, 2025

Attendees:

- Drew Relmas (Microsoft)

Agenda: Issue triage

Sep 18, 2025

Attendees

- Albert Lockett (F5)
- Josh MacDonald (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Tristan Sloughter (Groq)
- Jake Dern (Microsoft)

Agenda

- [jmacd] Talk about https://github.com/open-telemetry/otel-arrow/issues/1126
    - Somewhat stuck on Sync + Send in admin control messages
    - Action: wait for Laurent, Josh will ask LQ how to attack this next week
- [jmacd] About this hackathon
    - Josh's question is about which Arrow primitives would be useful for translating the Parquet exporter's format back into OTAP.
    - E.g., we have read a set of primary logs from the logs parquet, want to read the log_attrs (or scope_attrs or resource_attrs) table.
    - Arrow schema is written to Parquet metadata. The arrow-rs schema is encoded in parquet metadata so that we expect the proper types. Question is sort of how to be efficient about slicing a part of the record batch (e.g., of log_attrs) to place them in a new OTAP batch.
    - Think about using the Arrow kernel that subtracts a constant from an array. Arrow::compute::cast, see this https://docs.rs/arrow/latest/arrow/compute/kernels/numeric/fn.add.html
- [albert] albert hackathon
    - Demo of a KQL to datafusion logic query plan!
    - https://github.com/open-telemetry/otel-arrow/compare/main...albertlockett:otel-arrow:albert/kql-to-df-poc?expand=1

Sep 9, 2025

Attendees

- Laurent Querel (F5)
- Chris Hain (F5)
- Albert Lockett (F5)
- Josh MacDonald (Microsoft)
- Drew Relmas (Microsoft)
- Jake Dern (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda:

- [Chris] Demo of various scenarios
- [Laurent or Albert] Parquet + Datafusion
- [Josh] Brief talk about Add Context to OtapPdata, introduce OtapPayload by jmacd · Pull Request #1078 · open-telemetry/otel-arrow

Sep 4, 2025

Attendees

- Laurent Querel (F5)
- Albert Lockett (F5)
- Josh MacDonald (Microsoft)
-

Agenda:

- Discussions about ongoing PRs!
- [jmacd] Open questions around [WIP] Draft back-propagation mechanism by jmacd · Pull Request #1041 · open-telemetry/otel-arrow
- [laurent] Observed state store Endpoints to expose the current overall state of the pipelines and to stop pipelines by lquerel · Pull Request #1033 · open-telemetry/otel-arrow
- [albert] IPC stream state management https://github.com/open-telemetry/otel-arrow/issues/1061

Aug 26, 2025

Attendees

- Laurent Querel (F5)
- Albert Lockett (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Mike Heffner (streamfold/rotel)
- Ray Jenkins (rotel/streamfold)
- Josh MacDonald (Microsoft)
- Gokhan Uslu (Microsoft)
- Jake Dern (Microsoft)

Agenda

- [Group] Issue triage:
    - [OtapPdata equivalence testing · Issue #1003 · open-telemetry/otel-arrow]() discussed
    -
- [Laurent] Telemetry System PR
    - About PR 946 [Generic, high-performance, type-safe metric instrumentation framework by lquerel · Pull Request #946 · open-telemetry/otel-arrow]()
    - Roughly speaking, this is a multivariate metric SDK
        - Hot path (thread pinned to a core: no synchronization)
        - Cool path (may cross a thread boundary)
        - Concept of a MetricSet, AttributeSet
        - Live telemetry schema computed from all attributes, all metric sets
    - Static registry: knows about all Metric/Attribute sets
        - Metadata can be reported once using registration data
        - Node and pipeline control messages are used to exchange metrics across threads
        - Prometheus-compat export on HTTP servlet, capable of aggregation
    - Discussion about delta vs cumulative output mode
        - Josh refers to “lightstep-metrics SDK” which has a similar design
        - How you can choose cumulative vs delta on output, most users prefer cumulative
        - “Slot value” mechanism, question of whether to use real timestamps or …
    - Talk about dynamic attributes
- [Rotel/OTel] talk about how to organize collaboration
    - High-level ideas about next year of project priorities
    - Potential for non-Arrow data
    - Mike and Ray looking for ways to contribute
- [Josh] Engine type changes ⚠️as part of adding Nack including Option<PData> for return
    - Error<T> to Error, TypedError<T>: most errors do not need <T>, only where SendError<T> is involved, and only then a subset
    - NodeControlMsg to NodeControlMsg<PData>
    - https://github.com/open-telemetry/otel-arrow/pull/1017

Aug 21, 2025

Attendees

- Laurent Querel (F5)
- Albert Lockett (F5)
- Jake D (Microsoft)
- Tristan Sloughter
- Ray Jenkins (rotel/streamfold)
- Utkarsh Umesan Pillai (Microsoft)
- Drew Relmas (Microsoft)
- Josh MacDonald (Microsoft)

Agenda

- Issue roundup
    - Josh broke several tests, added #[ignore], will not do again
    - Albert has filed a couple of bugs, one w/ Go components
        - We may need to encode version info
- Rotel
    - Coming from librato, snowflake
    - Performance concerns in OTel Collector
    - Discussion about OTAP architecture and motivations
    - Talk about plugins and componentization, interoperability
        - [OTel-Arrow RFC]: Mixing Golang/Rust components by jmacd · Pull Request #13369 · open-telemetry/opentelemetry-collector
    - Talk about OTel-Arrow differences with the Go collector, hyper-edges, DAG
        - Josh: This is all sort of possible w/ Go collector, just not very formalized
        -
    - To be continued.
- Instrumentation/Multivariate metrics framework
    - LQ presents a draft [WIP] Internal metrics system by lquerel · Pull Request #946 · open-telemetry/otel-arrow
    - Strongly typed, multivariate, NUMA-aware.
    - Constant-value attributes
- AttributeProcessor and PData related operations

Aug 12, 2025

Attendees

- Laurent Querel (F5)
- Josh M (Microsoft)
- Jake D (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda

- Issue roundup
- [Josh] Is working on a rate-limiting proposal, studying the existing engine and effect handler
    - LQ: would like to use the rate limiter in the signal generator, to simplify the test apparatus
    - What are policies when we reach the limit:
        - Backpressure ? Drop? Redirect? Sample?
    - About rate-limiting per-core (e.g., with SO_REUSEPORT balancing) vs global ratelimits
    - Josh: Some background on the Envoy design. [WIP]
        - Rate limit requests
        - Rate limit filters
        - Rate limit local, shared, global

  - Josh: Hypothesis is the existence of a common configuration model for filters
- [Laurent] NUMA-aware and Type-safe Client SDK
  - Imagining an internal metrics library to report information about pipeline state
  - For otap-dataflow to use logs and metrics in a thread-per-core approach
    - The Rust OTel SDK itself will be used as the global implementation.
    - Logs and metrics can be inexpensively aggregated within a NUMA region (or per core)
  - We will configure the assignment between core and NUMA region
  - Metrics aggregation at NUMA region first
  - Discussion about type-safe instrumentation, connection with Weaver objectives
  - Hence: a new crate WIP, working on type-safe multivariate metrics
    - 1st case, for example the set of metrics generated by the perf exporter
    - We will use multivariate metrics, hand-code the telemetry as proof of concept
    - Eventually this will be generated from Weaver
  - [Josh] how do we answer questions like “why not use OTel-Rust SDK”?
    - It’s not designed for thread-per-core, not NUMA aware
    - Likely to want a _new_ API for type-safe metrics, including type-safety of internal feedback loops where aggregated metrics are read within the system. (Schema-driven)
    - It’s not multivariate
  - Take the perf exporter for example
    - Will maintain multiple counters: batched received{signal},
    - Point point is: {Timestamp, AttributeSet, LogsCounted, MetricsCounted, TracesCounted} for 3 metrics
    - JM: how about attribute set handling?
    - LQ: two categories of attribute,
      - Constants: numa region ID, cpu ID, pipeline ID … do not change within a node instance within a thread/core
      - Contextual, dynamic: based on PData (e.g., header, data-dependent), this is where the attribute-set lookup problem arises.
  - Example:
    - Instrument = meterProvider.Get(“scopename”).WithConstantAttributes(NUMA, etc..)
    - instrument.RecordBatch(ctx, (no attributes), Measurement(metric, value), Measurement(metric, value), ….) using empty attribute set
    - Or: instrument.RecordBatch(ctx, Attributes{...}, Measurement(metric, value), Measurement(metric, value), …)
    - OpenCensus had something like this!
    - Big idea: you have to lookup the attribute set only once
    -

Aug 7, 2025

Attendees

- Laurent Querel (F5)
- Jake Dern (Microsoft)
- Tristan Sloughter
- Chris H (F5)
- Michael S (F5)
- Albert L (F5)
- Josh MacDonald (Microsoft)
- Utkarsh P (Microsoft)

Agenda

- Laurent: Demo milestone
  - Josh: FYI: I will be focused on rate-limiting in OTAP pipelines, now in the milestones
  - Discussion about PR 803 w/ multicore results
  - Issue triage: great progress!
- Utkarsh: TLS handling through effect handler?
  - https://github.com/open-telemetry/otel-arrow/issues/889
- Go components have broken recently, working on it:
  - https://github.com/open-telemetry/opentelemetry-collector-contrib/pull/41830

Jul 29, 2025

Attendees

- Drew Relmas (Microsoft)
- Jake Dern (Microsoft)
- Albert Lockett (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Josh MacDonald (Microsoft)
-

Agenda

- Laurent: review scenarios for testing performance
  - OTLP-to-OTLP: no deserialize case (e.g., signal type router)
  - OTAP-to-OTAP: no deserialize case
  - Syslog-Attributes-Batch-OTLP: rename attribute, etc.
  - Syslog-Attributes-Batch-Parquet: to OTAP (“Barquet”), (difficult in Go)
  - Synthetic signal generator-to-OTAP or OTLP
  - Stretch goal:
    - Live reconfigure: to reconfig the individual node
    - Stretch^2: reconfigure the graph. T.B.D.
- Utkarsh/Josh: Talk about batching with Syslog receiver, UDP or TCP
  - To use views for singleton message?
  - Instead, use “embedded batching”

    - LQ: proposes a low-level batch, like Vec<Vec<u8>>, with async appender
    - Josh: looks like we have a pattern involving appending a PData to another PData
        - LQ: we have (Albert writing) a Pdata wrapper abstraction with state
        - JM: how about a simple approach, just build OTAP directly (Syslog-to-OTAP)
        - LQ: we could call this an “incremental view”.
        - Group: don’t do this first, just use a Vec<_> or binary buffer, have a multi-record view.
- Gokhan: saying hello, new member.
    - Q: must we run benchmarks (rust) on every CI/CD run? (only when rust changes, etc.)
    - LQ: no, the microbenchmarks we have are expensive and not very significant
    - LQ: we do want to have macrobenchmarks with historical performance, what Chris and Cijo are working on

Jul 24, 2025

Attendees:

- Laurent Querel (F5)
- Drew Relmas (Microsoft)
- Jake Dern (Microsoft)
- Albert Lockett (F5)
- Tristan Sloughter (MyDecisiveAI)
- David Dahl (F5)
- Utkarsh Umesan Pillai (Microsoft)

Agenda

- [Laurent] Demo first operational mini pipeline.
    - This is ready for review. First version of the engine capable of creating and executing a pipeline from a configuration by lquerel · Pull Request #532 · open-telemetry/otel-arrow
- [Drew] Release Process: [Process] Improving otel-arrow Release process · Issue #737 · open-telemetry/otel-arrow
    - [DRAFT] Release process improvements using otelbot by drewrelmas · Pull Request #739 · open-telemetry/otel-arrow
- [Jake] Missing delta dictionary support in arrow-rs
    - Support delta-encoded dictionaries in the Arrow IPC format · Issue #6783 · apache/arrow-rs
    - arrow-rs/arrow-ipc/src/reader.rs at 16794ab14fa62ecf67de0da9460cc5752a9358f4 · apache/arrow-rs
- [Josh] FYI Rust/Go proposal shared with Collector SIG [OTel-Arrow RFC]: Mixing Golang/Rust components by jmacd · Pull Request #13369 · open-telemetry/opentelemetry-collector

Jul 15, 2025

Attendees:

- Drew Relmas (Microsoft)
- Jake Dern (Microsoft)

Agenda

- [Jake] Question about otelarrowexporter - Why does it spawn 11 separate gRPC streams/connections and also time out the streams after 30 seconds by default?
    - Possible answer for the 11 separate connections is that the exporter uses half the number of reported available cores, in this case 22.
- [Jake] Question about delta dictionary support in otel-arrow-rust - Looks like support is not implemented yet. Is this something being worked on and/or open to contribution?

Jul 10, 2025

Attendees:

- Drew Relmas (Microsoft)
- Josh M (Microsoft)
- Tristan Sloughter (MyDecisiveAI)
- Jake Dern (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda:

- [Josh] Preview for Go/Rust interoperability. I will present this in the 7/23/2025 Collector SIG. https://github.com/open-telemetry/opentelemetry-collector/pull/13369
- [Drew] [otel-arrow-rust] Build Cache for CICD · Issue #380 · open-telemetry/otel-arrow
- [Drew] Schedule monthly docker digest updates with Renovate by drewrelmas · Pull Request #708 · open-telemetry/otel-arrow / Dependency Dashboard · Issue #417 · open-telemetry/otel-arrow
    - Josh suggests we archive beaubourg as historical, not for building. Maybe remove the source code, leave the README.
-

Jul 1, 2025

Attendees:

- Laurent Querel (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Jake Dern (Microsoft)
- David Dahl (F5)
- Mike Blanchard (Microsoft)
- Jacob Abraham (F5)

Agenda:

- [Laurent] on behalf of Albert: Implementation of the direct OTAP encoding for logs - "proto_bytes -> views -> OTAP". Benchmark results and key learnings. PR: https://github.com/open-telemetry/otel-arrow/pull/658
    - Conclusion: based on the benchmark results, we will only maintain OTAP pipeline engine
- [Utkarsh] [otap-dataflow] Add SyslogCEFReceiver by utpilla · Pull Request #655 · open-telemetry/otel-arrow
- [Laurent] on behalf of Joshua: Go Collector ← → Rust dataflow engine interop
- [Laurent]: First working version of the Rust dataflow engine
- [Blanch]: Query engine update
    - Pipeline expression (Rust construct)
    - Expression tree abstraction in order to run the query over multiple backends (apache arrow+datafusion, basic "OTLP query engine")
    - OTTL -> Expression tree -> Query Engine -> Output
    - KQL -> Expression tree
    - Expose this processor to the Go Collector (explored by Tom)

Jun 26, 2025

Attendees:

- Albert Lockett (F5)
- Jake Dern (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Drew Relmas (Microsoft)
- David Dahl (F5)
- Josh MacDonald (Microsoft)
- Tristan Sloughter (MyDecisiveAI)
- Mihir (Culver Max Entertainment Pvt Ltd)
- Laurent Querel (F5)

Agenda:

- [Albert] - OTAP Encoding from proto structs & bytes
    - Discuss feat: OTAP Logs Encoding from proto Structs with Views by albertlockett · Pull Request #625 · open-telemetry/otel-arrow · GitHub
    - This PR produces OTAP frames from Prost message objects directly via the trait
    - This week, working on decoding OTLP bytes as an implementation of the same trait
    - Performance? "OK". Benchmarks showing cost of black_box visitor.
        - visit bytes directly vs visit bytes after decoding Prost objects: 20% improvement
        - Perf is not as great as we would like: there is some cost associated with random-access via the View which is being deserialized

- Ownership complexity, use of a RefCell which may be hurting perf. Will continue.
- Josh thinks this is good! We’re avoiding two passes

- [Josh/Drew] Can we begin to use the KQL query engine to write simple queries for our own logs?
    - LQ has asked for simple tools to filter events from our own datafusion
    - Laurent’s PR [Internal tracing proposal by lquerel · Pull Request #633 · open-telemetry/otel-arrow · GitHub]()
    - Laurent is interested in actually using our own query language for our own instrumentation.
    - See also [opentelemetry-collector/docs/rfcs/component-universal-telemetry.md at main · open-telemetry/opentelemetry-collector · GitHub]()
    - There will be differences in how we reason about and debug the dataflow engine because of the two channels (control and data). We will not tie ourselves to the Golang collector’s instrumentation design at this time, though they are similar.
- [Group] We are close to having a bare-bones Rust pipeline!
    - We also have a test framework; this is exciting.
    - Josh would like to run a test of the Go collector’s memorylimiterprocessor on throughput/latency. (for example)
    - Stay tuned! LQ is working on synchronization-aware connectors, getting close.
    - We will connect Albert’s view-parser work with an OTLP receiver.
- [Josh’s update] Will be working on Go/Rust interoperability, looking at the [rust2go]() crate.
- [About yaml] Nevermind
- [About interop] Many ways to interoperate, at the “pipeline” or “node” level.
    - By “pipeline” level, we mean to separate pipelines into Rust and Go separately. In this model, we basically have two different runtimes.
    - By “node” level, we mean direct FFI-type integration.
    -

Jun 17, 2025

Attendees:

- Albert Lockett (F5)
- Jake Dern (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Drew Relmas (Microsoft)
- David Dahl (F5)
- Josh MacDonald (Microsoft)

Agenda:

- [Utkarsh & Albert] View pattern talk
  - Albert follows-up on the topic of Visitor patterns following last week.
  - Laurent prototyped a "View" trait alternative, giving us a simpler, lower-overhead approach (Josh was impressed, didn't realize Rust could do this!)
  - The View trait gives a more imperative interface for directly probing each visitable, making it possible to use ordinary for-loops.
  - Albert iterated, Utkarsh iterated, looking at lifetime.
  - Albert has experimented with using these Views to construct OTAP batches directly from OTLP bytes.
  - Utkarsh experimented with same, first approach was roughly equivalent to Prost, parsed objects and yielded views, and (naturally) slower than Prost; next approach was "lazy" which avoided a lot of overhead, yielding promising results
  - Laurent: many constraints, not obvious or easy how to solve this; thinks we are very close.
    - The ideal case is appealing: a set of conversion routines for each format "compatible with" OTLP, makes possible to re-use transformations
    - Josh tries to describe the "ideal" efficient view mechanism, it wants to not allocate but the order of field access is prescribed by the caller. The pros/cons of Visitors vs. Views comes out here.
    - Thinking about how to generate OTAP from a View or a Visitor; for a View the implementor has control over access, so (maybe) we memorize a set of offsets for what we skip. (This will make it hard for streaming data, but probably not an immediate problem.)
    - Josh asks about whether to memorize full random access or only what we've skipped.
    - We will anyway return to benchmarking the Prost encode/decode with translation to OTAP.
  - Utkarsh: also note that Prost uses unsafe and potentially we're not making a fair comparison?
    - LQ: we should try to avoid unsafe, but if we can't, we can't: willing to accept unsafe if we have to just test carefully.
    - I.e., does Prost assume utf8 validity?
    - Is there a problem with not scanning all bytes for validity? Flip-side: it's easy to deal with truncated protobuf objects.
    - Shall we have a recursion limit?
    - TIL in function programming the "View" traits are called Lenses
-

Jun 12, 2025

Attendees

- Laurent Querel (F5)
- Utkarsh Umesan Pillai (Microsoft)

- Josh MacDonald (Microsoft)
- Albert Lockett (F5)
- Jake Dern (Microsoft)
-

Agenda

- [Josh] Visitor pattern talk: https://github.com/open-telemetry/otel-arrow/pull/580
    - Aside: having problems with Cargo.toml and not having one in the top-two levels of directory hierarchy. The otap-dataflow directory does what we want.
    - Action: Josh will try to explain this better, research allocations; we will merge this and study it further.
- [Utkarsh] [otel-arrow-rust] Add UDP support by utpilla · Pull Request #568 · open-telemetry/otel-arrow
- [Chris] Perf Updates: moving along. OTel-Python SDK is being used to manage in-memory metrics from the experiments.
- [trask, 15 min] The GC and TC are collecting feedback from all SIGs in order to put together a roadmap (past and future) to share with the community:
    - What were the SIG's biggest achievements during the last 12 months?
        - Phase 1
    - What work is the SIG planning for the upcoming 12 months?
        - Phase 2
    - Are there any areas and/or sub projects that the GC/TC can help with? (e.g. cross-SIG blockers, prioritization, etc)
        - Promoting this groups work and goals around collector interop
        - Getting things merged into the collector contrib repository
            - If collector libraries were stable, then it would be easier to host things externally
        - Collector contrib repo -> arrow repo -> collector repo
            - Unstable dependencies causes all of these to need to be released together
        - Move retry logic into the collector instead of making it part of each exporter
        - Plugin system (that doesn't require statically compiling when building the collector)
        - Stable collector interfaces

Jun 3, 2025

Attendees

- Laurent Querel (F5)
- Albert Lockett (F5)
- Josh MacDonald (Microsoft)
- Jacob Abraham (f5)
- Ken Arora (F5)

- Jake Dern (Microsoft)
- Drew Relmas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Venkat Allam (Dell)
- Mike Blanchard (Microsoft)

Agenda

- [Laurent, Jacob, Ken] Engine configuration
    - Intro from F5 team with background on configuration model
    - Multi-tenant data plane pipeline configuration and first iteration of an internal plugin system by lquerel · Pull Request #470 · open-telemetry/otel-arrow · GitHub has merged
    - Pipelines have potential for one component to send/receive to or from many components. The Golang collector supports this in certain ways, for example processors at the beginning and end of a pipeline connect with multiple exporters/receivers. Connectors can route to multiple receivers/exporters, and @lquerel's Hyper-DAG model represents this type of structure.
    - The configuration model has also a namespace which admits potentially multiple tenants or configurations.
    - [Utkarsh] Q: in a thread-per-core environment, how should we think about say multiple receivers sending to one processor? LQ's A: We need an `await` point that interrupts the execution of a single-threaded runtime to allow other tasks to run on that cpu.
    - The configuration sub-system's output will be a constructed pipeline with the necessary adapters for converting between shared and local implementations for components in the pipeline.
    - This work, including the one above, includes use of "linkme" crate. This is an automatic registration mechanism based on the use of ELF linker sections etc.
- [Chris] Test infrastructure update
    - Working on pipeline performance orchestrator. Allows arranging of test components, Go/Rust collector/pipelines. For use with Docker, Kubernetes.
    - We will be able to write tests. Python is working well.
    - Do we have dedicated resources for running benchmarks and load tests? [Action: Josh or Cijo follow up. We think OTel does have resources such as this.]
    - [Chanly] Related topic: a "Perf" exporter monitors basic pipeline component as well as monitoring basic machine statistics (memory, CPU).
        - LQ wanted a simple implementation to start with. To use for debugging, something simple to start with makes sense.
        - This may be a short-term solution, just a basic way to monitor and debug component performance and machine stats.
        - Will avoid "self-meter", moving to "sysinfo". This just prints to the terminal.
        - We will look into using OTel-Rust, LQ has concerns about synchronization. Josh wants us to keep in mind an OTel-Arrow-direct SDK.]

        - [Utkarsh] The OTel-Rust SDK does require Send, uses a ReadWrite lock, atomic writes for counters, mutexes for histograms, etc. You can create one SDK per thread, however, but it leaves us with a problem of merging streams later.
        - More discussion in this thread: [Feature]: Lock-free Metrics Counting API · Issue #1386 · open-telemetry/opentelemetry-rust
        - We will need to experiment and benchmark to know more, and this is why we started with a simple "Perf" exporter.
- [Drew] Repository CI / PR Settings
    - Merged: Use clippy --workspace and enable warning breaks by drewrelmas · Pull Request #523 · open-telemetry/otel-arrow
    - Open Issues:
        - Pull Request Merge Queue Configuration · Issue #530 · open-telemetry/otel-arrow
        - Flaky codecov tests · Issue #529 · open-telemetry/otel-arrow
- [Utkarsh] Receiver configuration/implementation
    - While working on a Syslog/CEF receiver, … have come up with questions on how to use the new engine. Here are some assumptions:
        - Creation of sockets: responsibility of engine
        - Accepting and reading from connections: responsibility of the component
        - We have local (!Send) and shared (Send), …
        - Does each component implement both local and shared? (LQ: Not necessary.)
        - If you can, implement local; if you require Send (e.g., via tonic) then implement shared. For the local implementation, make sure to yield often enough that component tasks can interleave.
        - Does a shared processor force its traits upon a local receiver? A !Send function use or call a Send function, but the reverse is not possible.The engine transparently sets up adapters for Send/!Send conversion.
    - Role of EffectHandler. Has a method to create a TCP listener, for example, will also get UDP listener too? Yes.
        - The engine gives pre-initialized sockets, the effect handler is an interface to allow the use of system-optimized channel infrastructure. This gives us control over socket options based on the runtime and host environment.
        - The effect handler (e.g., of the receiver) abstracts these details.
        - The effect handler abstracts the channel implementation (broadcast, round-robin)
        - There are specific effect handler interfaces for each type of component, because processors are different from receivers are different from exporters.
            - Receivers have only a control channel
            - Processors have control, input and output channels
            - Exporters have control, input channels.

        - Effect handler also implements yield for local implementations to schedule tasks.
    -

May 29, 2025

Attendees

- Laurent Querel (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Albert Lockett (F5)
- Jake Dern (Microsoft)
- Drew Relmas (Microsoft)
- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- David Dahl (F5)
- Mihir Shah (Culver Max Entertainment Pvt Ltd)

Agenda

- Parquet files: what, where, how?
    - [otel-arrow-rust] Parquet Exporter · Issue #399 · open-telemetry/otel-arrow
    - Current thinking is to maintain the denormalized representation, with separate tables for each of the existing OTAP record batches (i.e., maintain the “Star” schema)
    - Maintains an order between table writes to avoid writing children before parents
    - Focus on the problem of developing unique IDs across record batches – this is addressed with a synthetic partition key. Has some concerns about idempotency, but it’s a simple approach we could get working quickly.
        - Deduplication can be addressed in compaction phase or during query.
        -
    - These partition keys enable sharing nothing between separate exporters.
    - Looking forward to object_store support.
    - The transforms applied:
        - Generate unique IDs
        - Remove parent ID delta encoding
- OTLP pipeline, relationship with OTel-Rust
    - We don’t want several levels of conversion. It’s already too many layers of translation.
    - OTel-Rust has been looking at bypassing the extra layer, too.
    - Laurent wants us to benchmark. Probably we want direct translation to/from OTAP.
    - Laurent describes an OTAP-only pipeline which serializes to/from OTLP bytes, means bypassing intermediate Prost objects.

    - Use the Rust DTO object then convert to OTAP? Or have conversions from DTO to OTAP and OTLP bytes directly?
    - Laurent: probably both OTAP and OTLP make sense, depends on scale. A smaller-scale user may prefer to generate OTLP because it's more efficient at small scale. On the other hand, we think if an OTAP pipeline makes sense, the instrumentation of that pipeline will also naturally prefer OTAP.
- Discuss how to instrument the pipeline
    - Question about Send and Sync: what constraints are there, and can we configure an OTel-Rust SDK for ?Send.
    - About Infrastructure for pipeline instrumentation · Issue #487 · open-telemetry/otel-arrow
    - Let's separate this question into Metrics and Logs
        - Metrics are a bit tricky. Lots of synchronization here.
        - Logs are not very synchronized. Tokio/tracing does not add synchronization.
        - Logs: create a struct on the stack, then export somewhere – exporter can avoid synchronization. OTLP is the worst case in this respect.
        - For OTLP logs: Send the object into a channel, channel will receive, convert to Prost message objects, then serialize.

May 20, 2025

Attendees

- Josh MacDonald (Microsoft)
- Laurent Querel (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Albert Lockett (F5)
- Jacob Abraham(F5)
- Jake Dern (Microsoft)

Agenda

- Project board: find it and other details in our community project page
- Introductions from Jake Dern, Microsoft SQL Server Azure team
    - Interest in telemetry ingest from SQL server on customer behalf, consolidation within Microsoft and an interest in improving performance
    - Laurent: What sorts of query are being used over the telemetry? "Kusto" a.k.a. KQL is used,
    - Would prefer to use Parquet representation.
- Materializing parent_id followup
    - Thanks @albertlockett https://github.com/open-telemetry/otel-arrow/pull/455
    - Next up, use this in the attributes/store.rs, Albert has opened a new PR to use the new code. Results are good! Up to 3x faster for decoding attributes from a record batch.
    - Additional context from Laurent: we designed the OTAP representation in phase 1 for transport optimization; this meant a tradeoff, it means that identifiers are

    batch-specific – because it leads to better compression. It means, for the next batch we start with an empty set and all identifier ranges reset.
  - (cont.) where do we optimize the record for “some destination”. Is it for transport? Is it for storage? If we can postpone this, we should.
  - (cont.) consider this example: an OTAP receiver receives transport-optimized batches, each batch corresponds with a set of OpenTelemetry records. There are some transformations, including parent_id and sorting which improves compression.
  - (cont.) Depending on what we want to achieve, we may or may not need to change the representation. At a minimum, we are likely to need to re-encode the parent ID
  - (cont.) We have also discussed receiving OTLP protobuf bytes, interested in directly decoding this representation into OTAP frames, leading to a set of records which are not optimized.
- Discussion about Arrow streaming (transport optimized) and de-optimized representation for in-pipeline processing.
  - (cont.) Discussion about Arrow flight. Arrow IPC is being used internally.
  - (cont.) Josh asks LQ about the “stream reset” and the management of schemas in the OTel setting. The OTAP receiver and exporter use dictionaries inside the Arrow IPC.
  - (cont.) LQ has two posts on the Arrow blog, one about dynamic schema management for OTAP. (TODO: insert links.)
  - (cont.) Back to the attribute store,talk about a “super schema” being the description of the OTAP columns that are actually present in a data set. If for example all the data uses only string and integer-value columns, then an attribute store needs only four columns (key, type, string, integer).
- Talk about what it means for a Parquet representation. What is the schema we will use?
  - LQ has two approaches in mind, one is “star schema”: a main record and a set of other records, one is a flattened record schema.
  - Problem is we are dealing with a-priori unknown schema for a stream of telemetry, in the general case, in OpenTelemetry. We need to deal with dynamic attributes (and/or separate problem to avoid this, see LQ’s weaver project).
  - Problem with flat schemas: the buggy programs that create dynamic attribute sets, could create gigantic schema.

May 15, 2025

Attendees

- Josh MacDonald (Microsoft)
- Drew Relmas (Microsoft)
- Laurent Querel (F5)
- Matthias Loibl (Polar Signals - listening only)
- David Dahl (F5)

- Albert Lockett (F5)

Agenda

- Project board: find it and other details in our community project page
- Status report / discussion topics below!
- [jmacd]: What does “materialize the parent_id column” really mean?
  - https://github.com/open-telemetry/otel-arrow/pull/447
  - Josh explained what’s going on here.
  - Laurent has feedback:
    - Key/value are dictionary encoded, they can be compared without constructing the value
    - This was done because of a tradeoff between compression of the key/value vs the parent_id.
    - Probably we should have a new column (named differently) which is the materialized parent_id. If you are working with OTAP in memory, no benefit from parent_id compression, so not needed.
    - We can postpone the sorting by key/value so that attributes are contiguous.
    - Matthias: seconded: we use the value-dictionary IDs
- [lquerel]: About https://github.com/open-telemetry/otel-arrow/pull/433
  - Problem was to add OTLP receiver based on Tonic, which requires `Send`
  - The dataflow pipeline is aiming for !Send data, to reduce memory synchronization (Arc, Mutex, etc)
  -

May 1, 2025

Attendees

- Laurent Querel (F5)
- Utkarsh Umesan Pillai (Microsoft)
- Josh MacDonald (Microsoft)
- Joseph Moore (F5)
- Lei Huang (Greptime)
- Drew Relmas (Microsoft)

Agenda

- Project board: find it and other details in our community project page
- Status report:
  - Josh: working on OTLP pdata and Golang collector tests for OTLP/OTAP
  - Drew: starting up on query support in OTAP Collector
  - Utkarsh: syslog and UDP performance testing w/ SPSC, etc.
  - Lei: has been studying incompatibility issues, looking at traces/logs support for otel-arrow-rust
  - Laurent: working on dataflow library: https://github.com/open-telemetry/otel-arrow/pull/351

- Talk about Sendable EffectHandler https://github.com/open-telemetry/otel-arrow/pull/351
  - We are looking at whether dataflow library in a thread-per-core setting requires the Send trait. For ease-of-use, this may be important. For performance, this may be a restriction.
  - We are studying what alternate runtimes e.g., Glomio/Monoio do here. We are looking at using Tokio single-threaded runtime as the most-compatible choice.
  - Problem: the Send trait interferes with competition between CPU and cache within even a single CPU. This means relying only on async/await primitives to maintain separation between competing threads.
  - Then: wanted tonic support, which requires Send.
  - Feedback:
    - Can we wrap !Send to make them Send? We can use channels and threads to bridge, which is possible because we have this distinction.
    - Lei: it's really hard to switch runtimes; nice idea but hard.
- Discussion about Substrait by Lei
  - Lei: has experience using substrate, datafusion, and telemetry data. It works.
  - Josh asks: can we build views to "modify" an OTAP data frame (a.k.a. Pdata); Lei: would call this a "plan". Yes, these can be built up, but it will likely take DataFusion to implement the projection(s).
  - Laurent: OTAP arrays have inter-relationships having to do with delta-encoding of arrays; even for internal IDs like parent_span_id, attribute refs, and so on, we have a question about materializing arrays.
  - Probably both types of approach will be used in the end; the batch processor would likely use low-level Arrow APIs, whereas a query transform processor would likely build plans.
- Discussion about protobuf generated code and where it lives
  - Not an issue for the moment; likely OTel-Rust and this project's multiple crates convene later

Apr 17, 2025

Attendees

- Trask Stalnaker (Microsoft)
- Tristan Sloughter (MyDecisiveAI)
- Laurent Querel (F5)
- Matthias Loibl (Polar Signals)
- Albert Lockett (LanceDB)
- Debabrata (Parseable)
- Josh MacDonald (Microsoft)
- Utkarsh Pillai (Microsoft)
- Jacob Abraham (F5)

Agenda

- Introductions
- Why Arrow?
- Why F5 cares?
  - With large appliances, high volumes of telemetry => large cost from OTLP because of its hierarchical organization
  - Began to look at columnar organization in OTel ~3 years ago
  - Compression rate improvements are substantial (results from Phase 1)
  - Memory pressure from dealing with OTLP objects is costly
- Why MS cares?
  - Engineering led: want a more efficient collector, want a healthy collector community
  - We like Rust
  - We like DataFusion
- Josh: talk about "pdata" in row-oriented and column-oriented world
  - https://github.com/open-telemetry/otel-arrow/issues/338
  - "Pdata" derive macros for OTLP [otel-arrow-rust] Add OTLP pipeline data object via derive macro by jmacd · Pull Request #339 · open-telemetry/otel-arrow
  - Round-trip OTLP-OTAP-OTLP validation framework by jmacd · Pull Request #346 · open-telemetry/otel-arrow
- Laurent: talk about thread-per-core "shared nothing" architecture
  - Goal is to allow single-threaded runtimes to run efficiently without sharing data
  - Interest in using shared-network ports (eg., SO_REUSEPORT, SO_REUSEADDR) to maintain isolation between single-threaded runtimes (as opposed to "work stealing")
  - New engine under development: Initial skeleton of the OTAP Dataflow library by lquerel · Pull Request #340 · open-telemetry/otel-arrow
-