OpenTelemetry: Rust SIG

THESE MEETINGS ARE RECORDED AND STORED IN ZOOM CLOUD
Meeting Recordings (PUBLIC)

For latest meeting Info Check: https://github.com/open-telemetry/community#calendar
Meeting Room:
https://zoom-lfx.platform.linuxfoundation.org/meeting/99328757633?password=4549d918-1e04-4295-8a37-f93d6fdb3135

Template - 2026/MM/DD (Tuesday 9:00 AM PT)

-
-

Agenda/minutes:

2026/08/05 (Wednesday 5:00 PM CET)

- scott@datadoghq.com (Datadog)
- Ishan

Agenda/minutes:

Talked about autoinstrumentation and where it belongs (contrib? Another repo?)
Some PRs for Scott to review tomorrow (#3761, #3759, #3758)
Also we need to fix the LF calendar invite

2026/08/25 (Tuesday 9 AM PT)

- Cijo Thomas (Microsoft)

No-one around.

2026/08/05 (Wednesday 5:00 PM CET)

- Scott Gerring (Datadog)

No-one around.

Agenda/Minutes

-

2026/06/24 (Wednesday 5:00 PM CET)

-

Agenda/minutes:

- [Scott Gerring] To review:
  - #3413 - Stabilise runtime trait - adds dynamic runtime detection & a joinable abstraction
  - #3512 otel-proto / SDK decouple (which will make #3460 cleaner)
  - #3460 - adding process context - we're almost there with the thread context OTEP-4947, which combined with this will give us runtime profiler support for rust apps.
  - Bryant's PR #3380 - relevant comment linked - fixing deadlocks on constrained tokio runtimes - I think we should get it in for OTLP stabilisation

2026/06/10 (Wednesday 5:00 PM CET)

- Scott Gerring (Datadog)
- Björn Antonsson

2026/05/27 (Wednesday 8:00 AM PT)

- Scott Gerring (Datadog)
- Cijo Thomas (Microsoft)

Notes:
Wed - move 1 hour earlier
Tuesday - move 2 hour later
Re-emphasize the priority for upcoming release:

Tracing API stable
OTLP stable - main blocker is async runtime stabilization

2026/05/19 (Tuesday 9:00 AM PT)

- Cijo, Bjorn

Agenda/minutes:

- Discussion on Tracing stability, open PRs, release plans

2026/05/13 (Wednesday 8:00 AM PT)

- Scott Gerring (Datadog)

Agenda/minutes:

- No one around! Clocking off.

2026/04/01 (Wednesday 8:00 AM PT)

- Scott Gerring (Datadog)

Agenda/minutes:

No-one around! Clocking off.

2026/03/04 (Wednesday 8:00 AM PT)

- Scott Gerring (Datadog)
- Björn Antonsson (Datadog)
- Warren Snipes
- Parth Sharma

Agenda/minutes:

- Warren had an issue setting up mixed logs+traces export where logs go out stdout via tracing, and correlation is difficult. This seems like a common issue and perhaps we can

either point out a better solution, or at least document best-practices in our examples repo
- Warren is also trying to do context propagation from Rust into a Python app via an FFI (for data science-y stuff on the inside) - interested in either 1/ returning nested telem back to parent rust, or 2/ exporting spans from python. We have difficulty thinking of a clean way of doing this; if anyone reading this has a good idea, plus write ;)

2026/02/18 (Wednesday 8:00 AM PT)

- Franco Posa (Grafana)
- Alex Bird (Elastiflow)
- @ Brett Mitchell (Elastiflow)
- Yadi Abdalhalim (Mux)
- Björn Antonsson (Datadog)
-

Agenda/minutes:

- Elastiflow is exploring the possibility of a Rust OTel collector - looking to understand the current relationship between the OTel-Arrow / Dataflow project and Rust OTel maintainers.

2026/02/10 (Tue (AM PT)

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Paul Le Grand des Cloizeaux (Datadog)

Agenda/Minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/3262
- Reviving this PR https://github.com/open-telemetry/opentelemetry-rust/pull/2962
- Discussion on the Tracing performance issues
    - We need to fix performance before calling API stable, as fixing perf most likely involve change to API itself.
    - Eg: https://github.com/open-telemetry/opentelemetry-rust/issues/1109

2026/02/04 (Wed 8 AM PT)

- Cijo Thomas (Microsoft)
- Franco Posa (Grafana)
- Björn Antonsson (Datadog)

Agenda/Minutes:

Not much progress on pending PRs, Cijo to take a look.

Lalit is helping with OTLP related open PRs.

Bjorn will ping Paul to get latest status of SpanProcessor refactor

Franco continuing with tower instrumentation stabilization

[cijo] - lets add benchmarks and weaver check to the same before declaring stable.

2026/01/13 (Tuesday 9 AM PT)

- Cijo Thomas (Microsoft)
- Franco Posa (Grafana)
- Björn Antonsson (Datadog)
- Christian Leghadjeu
- Utkarsh Umesan Pillai (Microsoft)

Agenda/Minutes:

- [Cijo] Milestones updates
  - OTLP stable, Tracing API stable - moved to Mar 31 2026
  - Tracing SDK stable - moved to June 2026
- [Björn] Is the doc link in the calendar meeting broken?
  - Cijo will fix it.

2026/01/07 (Wednesday 5pm CET)

- Scott Gerring (Datadog)
- Franco

Agenda/Minutes:

- Scott's just got back on deck and hasn't had much of a look at anything yet
- Franco's keen to do some more work on the Tower integration in contrib
    - Created tracking issue for stabilization: https://github.com/open-telemetry/opentelemetry-rust-contrib/issues/526
-

time taken,2025/12/02 (Tuesday 9:00 AM PT)

- Björn Antonsson (Datadog)
- Cijo Thomas (Microsoft)
- Christian Leghadjeu

Agenda/minutes:

2025/11/25 (Tuesday 9:00 AM PT)

- Björn Antonsson (Datadog)
- Cijo Thomas (Microsoft)

Agenda/minutes:

- Fix for current_context in span processor
    - https://github.com/open-telemetry/opentelemetry-rust/pull/3262
- Tracer convenience methods
    - https://github.com/open-telemetry/opentelemetry-rust/pull/3251
- Span processor API refactor
    - https://github.com/open-telemetry/opentelemetry-rust/pull/2962

2025/11/04 (Tuesday 9:00 AM PT)

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Utkarsh Umesan Pillai (Microsoft)
- Prasad (Jio)

Agenda/minutes:

- Remove circular dependencies with tracing-opentelemetry https://github.com/open-telemetry/opentelemetry-rust/issues/3190
    - Action item: remove the experimental feature in the tracing appender Björn Antonsson
- Builds are failing in `cargo-deny` https://github.com/open-telemetry/opentelemetry-rust/issues/3216
-

2025/10/21 (Tuesday 9:00 AM PT)

- Björn Antonsson (Datadog)
- Lalit Kumar Bhasin (Microsoft)
- Nikhil Bhatia
- Christian Leghadjeu
- Zhongyang Wu

Agenda/minutes:

- Alternating meeting between timezone

2025/9/23 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- Release proto crate

2025/9/16 (Tuesday 9:00 AM PT)

Attendees:

- Björn Antonsson (Datadog)
- Scott Gerring datadog)

- Christian Leghadjeu

Agenda/minutes:

- The circular dependencies between `tracing-opentelemetry` and `opentelemetry-appender-tracing` makes it impossible to clean up things. → https://github.com/open-telemetry/opentelemetry-rust/issues/2886
- Issue & PR triage
    - -> probably need to talk to @Cijo and the gang to work out what to do with this.
- backoff/retry OTLP PR
- Open PRs from Paul from May could use eyes / merge-queing

2025/9/09 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Nikhil Bhatia
- Björn Antonsson (Datadog)
- Sven Cowart (ElastiFlow)
- Christian Leghadjeu

Agenda/minutes:

- backoff/retry OTLP PR 👼 - thanks for the review Björn! I think it could use more eyes with an eye to merging it soon

2025/8/26 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/3131
- Tuesdays 9 AM PT and Wednesdays 8 AM PT - alternating.
-

2025/8/19 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)
- Christian Leghadjeu

Agenda/minutes:

- We had a new member Christian Leghadjeu join the meeting. He is looking to contribute to the repo.
- [Not discussed since due to lack of attendance] OTLP - backoff/retry handling - is part of the OTLP stabilisation efforts and could use some eyes. Needs `experimental_async_runtime`
- [Not discussed since due to lack of attendance] Potential alternating slots for SIG to spread TZ impact around ?

2025/8/12 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

-

2025/8/5 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Nikhil Bhatia
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Copilot fully enabled, and EasyCLA issues resolved.
- OTLP Exporter updates to gear it towards stable release.
- Distributed Tracing - no progress, tracing-opentelemetry has not yet accepted the PR from Bjorn
    - https://github.com/open-telemetry/opentelemetry-rust/issues/3121
    - Writeup about tracing guidance (draft).

Template - 2025/07/29 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)
- Nikhil Bhatia
- Zhongyang Wu

Agenda/minutes:

- We had Nikhil Bhatia who wants to add resource detectors for containers, process, host in C++ repo but wanted to ask us about our opinion

Template - 2025/07/22 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Scott Gerring (Datadog)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes

OTLP release and blockers.
Discussion about next release
    Looks like we have some breaking changes, so better to do 0.31
    And delay until we get majority of Tracing API breaking changes in
Discussion on the OTel Rust's multiple APIs to create/activate Span
    Tracer.in_span
    Tracer.SpanBuilder()...builder()
    Both has usability issue and likely performance issues.
    [cijo] to add benchmarks to see if the perf issues are measurable/impactful

2025/07/15 (Tuesday 9:00 AM PT)

Attendees:

- Scott Gerring (Datadog)
- Zhongyang Wu
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/2962
    - Let's check with Paul to get last lint in then potentially merge?
    - Update: we're actually trying to hold this open and do the breaking changes after the next PR
- https://github.com/open-telemetry/opentelemetry-rust/pull/3028

2025/07/08 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)
- Björn Antonsson (Datadog)
- Cijo Thomas (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

-

2025/07/01 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
-

Agenda/minutes:

- Brief discussion on the SPAN processor refactor API
    - Should we do all breaking changes in one release, as opposed to doing them in every release?
    - A large number of breaking changes are pending, once the tracing-bridge refactor is done, so might be prudent to wait till that and merge all breaking changes in open release.
    - Paul's PR is about processor, so breaking change is restricted to the sdk crate only..
- Low attendance and ended early.
-

2025/06/24 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

-

2025/06/17 (Tuesday 9:00 AM PT)

Attendees:

- Anton Grübel (Baz)
- Björn Antonsson (Datadog)
- Cijo Thomas (Microsoft)

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/2962

2025/06/03 (Tuesday 9:00 AM PT)

Attendees:

- Paul Le Grand des Cloizeaux
- Zhongyang Wu

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/2962
- No one else joined so we ended early

2025/05/27 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Scott Gerring (Datadog)
- Björn Antonsson (Datadog)
- Paul Le Grand des Cloizeaux (Datadog)
- Harold Dost
- Zhongyang Wu

Agenda/minutes:

- 0.30 release is done for the main repo.
    - Metric SDK is now marked stable 🎉.
    - Includes Views, Cardinality capping
- 0.31 is the next release
  -https://github.com/open-telemetry/opentelemetry-rust/milestone/22
    - No new items added, except those carried over from 0.30 plan.
    - Logs, Metrics API/SDK are stable now, can focus more in Distributed Tracing.

    - OTLP Exporters in general need some attention to stabilize them too.
        - Integration tests are flaky, need to fix them.
        - Expand coverage for the same.
- tracing-opentelemetry discussion
    - We should try to get to some conclusion about ownership/maintainers of `tracing-opentelemetry` so we can move this PR forward https://github.com/tokio-rs/tracing-opentelemetry/pull/202
    - One proposal is for OTel to adopt the tracing-otel integration.
    - Given Tracing's being the defacto standard, we'd be helping majority users by adopting this and making sure it works e2e.
    - We already do a similar thing with async-runtimes - there is official support for the popular runtime already.
    - Issues:
        - Currently is uses a lot of internals
        - `tracing` typically acts as a firehose for spans, the bridge currently does a lot to reduce the firehose so that
        - There's a desire to ensure correlations between spans on either side of the bridge
- Outcome:
    - Lets give ~ 1 week for tracing's owners to comment on the overall plan for tracing-opentelemetry
    - Zhongyang/Cijo to ping the tracing maintainers again.
    - Then we can make a proposal to host a crate in the otel-rust-contrib repo, which would cover the same functionality as covered by tracing-opentelemetry.
    -
- https://github.com/open-telemetry/opentelemetry-rust/pull/2962 ← Paul's PR

2025/05/20 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Paul Le Grand des Cloizeaux

Agenda/minutes:

- Tracing-opentelemetry PR
- 0.30 release

2025/05/13 (Tuesday 9:00 AM PT)

Attendees:

- Björn Antonsson (Datadog)
- Scott Gerring (Datadog)
- Paul Le Grand des Cloizeaux (Datadog)

Agenda/minutes:

- There is a refactor PR for the Span Processor API to overcome a few issues https://github.com/open-telemetry/opentelemetry-rust/pull/2962
- We should try to get to some conclusion about ownership/maintainers of `tracing-opentelemetry` so we can move this PR forward https://github.com/tokio-rs/tracing-opentelemetry/pull/202
- Björn: When looking at the OpenTelemetry Specification it says that in a language that supports concurrent execution, the `Span` API should be thread safe. The current implementation has a `Span` trait that is not thread safe and then forces all implementations into the same mutex protected `SynchronizedSpan` to be able to share it in the `Context`. I'm thinking that the `Span` trait should be `Send + Sync` and it's up to the implementations to ensure this.

Any issues with
https://github.com/open-telemetry/opentelemetry-rust/pull/2758
https://github.com/open-telemetry/opentelemetry-rust/pull/2765

2025/05/06 (Tuesday 9:00 AM PT)

Attendees:

- Björn Antonsson (Datadog)
- Scott Gerring (Datadog)
- Anton Grübel (Baz)
- Lalit Kumar Bhasin (Microsoft)
- Paul Le Grand des Cloizeaux (Datadog)
- Cijo Thomas (Microsoft)

Agenda/minutes:

-
- There is now an open PR for the OpenTelemetry - Tokio Tracing changes https://github.com/tokio-rs/tracing-opentelemetry/pull/202
- Any issues with https://github.com/open-telemetry/opentelemetry-rust/pull/2758 https://github.com/open-telemetry/opentelemetry-rust/pull/2765

2025/04/29 (Tuesday 9:00 AM PT)

Attendees:

- Paul Le Grand des Cloizeaux (Datadog)
- Björn Antonsson (Datadog)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Discussed open PRs and recent issues.

2025/04/22 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Paul Le Grand des Cloizeaux (Datadog)
- Björn Antonsson (Datadog)
- Anton Grübel (Baz)
- Utkarsh Umesan Pillai (Microsoft)

Agenda

- https://github.com/open-telemetry/opentelemetry-rust/issues/2939
    - https://github.com/open-telemetry/opentelemetry-rust/pull/2895

    - Need to solve the SpanProcessor interface holistically, including adding the experimental OnEnding addition from spec, and also show an example of authoring SpanProcessors.
- Deprecate DatadogExporter?
    - Datadog is currently working on an official supported exporter, outside of the contrib repo, which should come within the next few months.
      Let's wait a bit until then

2025/04/15 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Caleb Schoepp (Fermyon)
- Utkarsh Umesan Pillai (Microsoft)

Agenda

- https://github.com/open-telemetry/opentelemetry-rust/milestone/21 Walked through
- Security issues resolved.
-

2025/04/08 (Tuesday 9:00 AM PT)

Attendees:

- Scott Gerring (Datadog)
- Björn Antonsson (Datadog)
- Utkarsh Umesan Pillai (Microsoft)
- Anton Grübel (Baz)
- Paul Le Grand des Cloizeaux (Datadog)
- Zhongyang Wu

Agenda/minutes:

- Tokio-tracing update
    - => would be good to have a chat with the tokio-tracing folk soon

- Close this PR: feat: allow retrieval of Context from ContextGuard by brettmc · Pull Request #2894 · open-telemetry/opentelemetry-rust (Need to keep the public API surface minimal. Moreover, there is already a low cost workaround for their issue)
- Need to test this [Bug]: PeriodicReader with with_interval not respected · Issue #2888 · open-telemetry/opentelemetry-rust (Another user has commented that they are facing the same issue)
-

2025/04/01 (Tuesday 9:00 AM PT)

Attendees:

- Scott Gerring (Datadog)
- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Utkarsh Umesan Pillai (Microsoft)
- David Hewitt (Pydantic)
- Anton Grübel (Baz)

Agenda/minutes:

- Issue management
    - Hierarchy - e.g. #2886
- Tokio-tracing next steps linky
- Hello David Hewitt!

2025/03/25 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Shivanshu Raj Shrivastava (SigNoz)
- Björn Antonsson (Datadog)
- Zhongyang
- Anton Gruebel (Baz)
- Utkarsh Umesan Pillai (Microsoft)

Agenda:

- 0.29 release done for main repo
- 0.30 milestone created
    - Focusing on fixing telemetry-induced-telemetry
    - tokio-rs/tracing interop
    - Stabilize Context/Baggage
    - Metric cardinality limit and stabilize SDK's public API
    - Example app showing all capabilities/signals
- Would like to get help with -contrib repo management
    - Auto assign PRs, instructions for making a new component, layout requirements etc.
    - Copy from other SIGs.
    - CI stability, policies etc.
- Question: Are we testing the combination of features? Björn
-

2025/03/18 (Tuesday 9:00 AM PT)

Attendees:

- Björn Antonsson (Datadog)
- Cijo Thomas (Microsoft)
- Scott Gerring (Datadog)
- Anton Grübel (Baz)
- Shivanshu Raj Shrivastava (SigNoz)
- Mindaugas Vinkelis
- Lalit Kumar Bhasin (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda:

- 0.29 release
    - https://github.com/open-telemetry/opentelemetry-rust/milestone/19
    - Prometheus deprecation?
- Contribution for Actix instrumentation: https://github.com/open-telemetry/opentelemetry-rust-contrib/pull/202
- Need some understanding on how to implement shutdown for gRPC (crate tonic)
  I see this change
  https://github.com/open-telemetry/opentelemetry-rust/pull/2764/files#diff-7e030c09778110d005a4f1b3030b8318bffa2ca09a7d897877ec71a495a971f4R16
  Q: Why is mutex required here, and how to implement shutdown now?
  another example where this needs to be implemented
  https://github.com/open-telemetry/opentelemetry-rust/pull/2812/files#diff-d03e458f06d40

7ca06ccd831165daaba1b88a777af800bd695d4cc37b46073f5R92

ToDo:
1) use tokio mutex in metrics - 2802
2) implement shutdown for tonic for logs and spans, once 1 is fixed, implement it for metrics too - 2779 - reason for spawning a thread to acquire a lock is, the shutdown is not async.

-

2025/03/11 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Scott Gerring (Datadog)
- Björn Antonsson (Datadog)
- Paul Le Grand des Cloizeaux (Datadog)
- Drew Relmas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Zhongyang Wu

Agenda:

- Hello from Lucas Jahier!
- Release plans https://github.com/open-telemetry/opentelemetry-rust/milestone/19
- Open PRs.
- Docs PR:
- https://github.com/open-telemetry/opentelemetry.io/pull/6515
- More
- https://github.com/open-telemetry/opentelemetry-rust/pull/2664#discussion_r1975582373
    -

2025/03/04 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Paul Le Grand des Cloizeaux (datadog)
- Anton Grübel (Baz)

- Björn Antonsson (Datadog)
- Drew Relmas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Follow up from last week:
    - Start a milestone to track the catchup
    - [cijo] - Tracing API and SDK has a lot more cleanups required. Catching up with Logs/metrics is just one thing, but much more.
    -
- Context out-of-drop support https://github.com/open-telemetry/opentelemetry-rust/pull/2378
    - Merging today.
    - Context lacks sufficient test coverage overall, but that is separate (below)
- Plan to get Context, Baggage to be stable
- Logs Sdk/Appender-Tracing to be moved to stable soon.
- Misc
    - Need to add Microsoft specific exporter in contrib-repo.
    - Auto Instrumentation repo for Rust.
        - Community issue to request
    - Potential in optimizing the tracing API

2025/02/25 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Anton Grübel (Baz)
- Björn Antonsson (Datadog)
- Zhongyang Wu
- Mason Gup (Medidata)
- Lalit Kumar Bhasin

Agenda/minutes:

- Welcome Mason!
- Welcome Scott Gerring as new Approver.

    - Need to check if all existing maintainers/approvers want to keep the position or move to Emeritus.
- Env variable vs code https://github.com/open-telemetry/opentelemetry-rust/issues/1225#issuecomment-2682568493
- Tokio-tracing
    - https://github.com/open-telemetry/opentelemetry-rust/issues/2420
    - A write up of end goal.
    - Send PR with unit-tests that shows expected behavior (okay it'll be broken now)
- Tracing needs a lot of catching up with logs/metrics.
    - Need a list of issues,
        - Zhongyang will help with a list.Target to open the issue by EOW
    - Performance checks.
- Rust 2024 edition
    - Anton will start with this.

2025/02/18 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Nahum Shalman
- Scott Gerring
- Anton Gruebel
- Drew Relmas (Microsoft)
- Björn Antonsson (Datadog)

Agenda/minutes:

- Calling for more Approvers.
    - OTel Rust and OTel Rust Contrib repo needs more hands
    -
- Env vs Code preference: https://github.com/open-telemetry/opentelemetry-rust/issues/1225
-

2025/02/11 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Drew Relmas (Microsoft)
- Nahum Shalman
- Björn Antonsson (Datadog)
- Anton Grübel (Baz)
- Paul Le Grand des Cloizeaux (Datadog)
- Zhongyang

Agenda/minutes:

- Fix Google doc link in Calendar invite/desc - Cijo to fix this.
- 0.28 release done
- Next release?
    - 0.28.1 is better than 0.29 as we expect to handle bug fixes mostly
    - Appender-Tracing needs a 0.28.1 once `tracing-opentelemetry` updates.
- Any updates on Otel Tracing API vs Tokio-Tracing? · open-telemetry opentelemetry-rust · Discussion #1032 · GitHub -> OpenTelemetry Tracing API vs Tokio-Tracing API for Distributed Tracing · Issue #1571 · open-telemetry/opentelemetry-rust · GitHub
    - An opt-in feature to translate tokio-tracing into open telemetry
    - This will make OTel be aware of current-span, even when span is created using tokio-tracing
    - At some point in future, OTel will also have easy-to-use macros like #instrument wrapping the underlying API.
    - There is a weaver tool which can auto-generate type-safe APIs.
    - eBPF - early stage exploration

2025/02/4 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- 0.28 release

- Error handling PRs

2025/01/28 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Zhongyang Wu
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- 0.28 delayed
    - Error types
- [Stable Release] Cardinality Cap removed, need to bring it back, which has pre-req of stabilizing Views.
- 1.0 Stable - ~~ June 2025
    - The plan is to let Tracing catch up so as to do 1.0 for all 3 signals.
    - Step 1 is to get a list of Tracing 1.0 blockers.
-

2025/01/21 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Kumar Bhasin

Agenda/minutes:

- -
- Release plan review.
- Issues to discuss:
    - chore: modify LogExporter and TraceExporter interfaces to support returning failure by scottgerring · Pull Request #2381 · open-telemetry/opentelemetry-rust

    - Will do thorough review as this has public API changes that needs to be replicated throughout the repo.
    - Add force_flush to LogExporter by ThomsonTan · Pull Request #2276 · open-telemetry/opentelemetry-rust
    - https://github.com/open-telemetry/opentelemetry-rust/pull/2462
    - [Cijo to send a PR to remove is_shutdown checks in hotpath.
- Misc-
    - Update the doc link in the community page.

2025/01/14 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Shivanshu Raj Shrivastava (SigNoz)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- 0.28 release to be expected next week

2025/01/07 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)
- Dimitar Valkov
- Lalit Kumar Bhasin (Microsoft)
- Shivanshu Raj Shrivastava (SigNoz)
- Zhongyang Wu

Agenda/minutes:

- 0.28 release to be expected in two weeks
- Metrics Shutdown issue https://github.com/open-telemetry/opentelemetry-rust/issues/2442
    - Doing cleanup using `Weak` instead of `Arc` does not seem worth the perf cost

    - It might be okay to not worry about the cleanup, as for most use cases `MeterProvider` has the same lifecycle as the application
- https://github.com/open-telemetry/opentelemetry-rust/pull/2485 Need to review if using `bytes::Bytes` instead of `Vec<u8>` helps with performance
- https://github.com/open-telemetry/opentelemetry-rust/pull/2479
    - General consensus among the attendees was that it's better to NOT provide default implementation for the trait methods
    - Not having a default implementation, pushes the component author to understand these public APIs and make a conscious decision for their components
    - Also, it's SemVer compatible to add default implementations later (if needed). However, removing default implementations will be a breaking change.

2024/12/17 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Shivanshu Raj Shrivastava (SigNoz)
- Björn Antonsson (Datadog)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Covered 0.28 milestones
- https://github.com/open-telemetry/opentelemetry-rust/issues/1678 can be revisited to add a milestone and GA date for Tracing, since we have more people working on Tracing signal.
- No meeting next 2 weeks due to OTel wide policy!
- Enjoy vacations.

2024/12/10 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Scott Gerring (Datadog)
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang Wu
- Nahum Shalman

Agenda/minutes:

- Release -
    - 0.28.0 is the next, and marking Log, Metric SDK as RC
    - Milestone created: https://github.com/open-telemetry/opentelemetry-rust/milestone/18
    - Would remove trace from default features in api/sdk/otlp/stdout crates
    - Must fix https://github.com/open-telemetry/opentelemetry-rust/issues/2386 , long delayed!
- A
- B

2024/12/03 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Björn Antonsson (Datadog)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- Release - 0.27.1 is done
- 0.28.0 is the next, and marking Log,Metric SDK as RC
    - Blocking issues
- Discuss overlapping context scopes https://github.com/open-telemetry/opentelemetry-rust/pull/2378
- Tokio-Tracing
    - https://github.com/cijothomas/opentelemetry-tracing/blob/main/src/opentelemetry_sdk.rs#L115 Old prototype for using tracing to take care of Context
    - Fixing 2378 is a pre-req to explore storing Context purely in OTel itself.
- Sync vs Async exporter
    - We can offer separate traits for exporters that do no need async vs that need async

2024/11/26 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Björn Antonsson (Datadog)
- Lalit Kumar Bhasin (Microsoft)
- Scott Gerring (Datadog)

Agenda/minutes:

- Scott Gerring (Datadog) - demo on integrating rust with Otel.
- 0.27.1 Patch Release
    - Nov 27
- 0.28.0 Release - End of year
    - Cijo - still looks good, breaking changes in SDK
    - RC for logs and metrics SDK.
- 1.0.0 Release - Jan 2025
    - On track for logs, metrics
- Blockers for stable release - Logs
- Blockers for stable release - Metrics
- Open PRs
- Open Issues

2024/11/19 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Björn Antonsson (Datadog)
- Lalit (Microsoft)

Agenda/minutes:

- Metrics performance improvements
    - Contention is the key remaining issue.
        - Sharding (manually or via external crates) prototypes are shown to fix this issue.
    - Also need to improve the collect() path.

- 0.27.1 Release
    - Nov 29
- 0.28.0 Release - End of year
- 1.0.0 Release - Jan 2025
    - On track for logs, metrics
- Blockers for stable release - Logs
- Blockers for stable release - Metrics
-

2024/11/12 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Bjorn Antonsson (Datadog)
- Harold Dost
- Nahum Shalman (Equinix)
- Zhongyang

Agenda/minutes:

- Release 0.27
    - Metrics API and Logs API to RC
    - Metrics SDK / Metrics OTLP exporter to Beta
        - View configuration under feature flag.
        - With Periodic Reader without runtime under feature flag
    - Nahum:
        - Update to tonic - configuration for TLS not working.(0.23 -> 0.24)
            - https://github.com/davidB/tracing-opentelemetry-instrumentation-sdk
    - Issue: configuration issue with custom headers.
- Harold: Meetup in local area (Paris/december), to talk about the current otel-rust status.
    - https://www.meetup.com/rust-paris/events/ < December Edition of the meetup
- Open PRs/Issues
- Bjorn Antonsoon has an interest in working on Tracing.
    -

2024/11/05 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Fabien Savy
- Mindaugas Vinkelis

Agenda/minutes:

- Release 0.27:
    - [Feature]: STREAM_CARDINALITY_LIMIT should be config · Issue #1951 · open-telemetry/opentelemetry-rust · GitHub
    - Instrumentation Scope and cloning · Issue #2246 · open-telemetry/opentelemetry-rust · GitHub
        - Need to discuss further whether we need to remove Cow.

  * https://github.com/open-telemetry/opentelemetry-rust/pull/2142
- Open PRs/Issues
    - Moving Results from opentelemetry to opentelemetry-sdk
- Internal logging review.
- Crazy idea, on how to improve performance of metric measurement by a lot.
    - Contention is still an issue, need to revisit sharding of HashMap to reduce this.
    - counter.Add(); counter.Add() - all goes to the same thing underneath, what if each clone works independently so they don't interfere, and merge then in collect/background-thread phase.
    - Similar ideas as Shaun suggested here : https://github.com/open-telemetry/opentelemetry-rust/issues/1386
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1564/files#diff-efe82800c0ca89e5cdf4c14f0674f753b0148351f0bfad79f89b20d295c4e6e4R58
    - One suggestion is to try the idea from above: Offer feature-flag and take dependency on external crates.
    - https://github.com/open-telemetry/opentelemetry-specification/issues/3904#issuecomment-1970349802 Spec's flexibility in enforcing cardinality limit
    -

2024/10/29 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang
- Harold Dost
- Tom Tan

Agenda/minutes:

- Release 0.27:
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1951
      OpenTelemetry Maintainers Meeting Notes
    - https://github.com/open-telemetry/opentelemetry-rust/issues/2246 \
        - Need to discuss further whether we need to remove Cow.

    * https://github.com/open-telemetry/opentelemetry-rust/pull/2142
      Create a separate reader (say) PeriodicReaderDedicatedThread.
      @cijo to confirm if this is for 0.27
    - Are we good for 1st Nov ?

    * Issues

    * PR
      * ValueMap interface change : The perf is not consistent in different tests.
      https://github.com/open-telemetry/opentelemetry-rust/pull/2117
        - Lalit to check the perf with flamegraph.
- Inconsistency in shutdown interface:
    - Metrics async shutdown - return status
    - Logs shutdown without return status
    - Traces async shutdown without return status
        - Lalit to create an issue to discuss.
        - Also to check force_flush.

2024/10/22 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)

- Lalit (Microsoft)
- Tom Tan (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

-

2024/10/15 (Tuesday 9:00 AM PT)

Attendees:

- Utkarsh (Microsoft)
- Lalit (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- Meter builder pattern: https://github.com/open-telemetry/opentelemetry-rust/issues/2164
    - Fabien is working on it.
- Metrics API Cleanups
    - Utkarsh continuing on this.
- Use dedicated thread for Metrics PeriodicReader
    - OTLP Exporter - hyper, tonic, reqwest need tokio runtime
        - reqwest-blocking should be usable without tokio runtime
- BatchExporter to dedicated thread
    - Custom runtime - using tokio current thread
    - We don't support any remote n/w clients supporting async.
        - grpc-io doesn't need runtime -
    - OTLP/Zipkin + HTTP (reqwest-blocking)
- ValueMap interface change by fraillt · Pull Request #2117 · open-telemetry/opentelemetry-rust
    - PR looks good, but we see some perf drop in histogram throughput.
    - To be tested manually, without release flag.

2024/10/08 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Tom Tan (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Fabien Savy

Agenda/minutes:

- 0.26.1 patch for `opentelemetry-proto`
  - https://github.com/open-telemetry/opentelemetry-rust/pull/2168
- 0.27 milestone review
  - https://github.com/open-telemetry/opentelemetry-rust/milestone/16
- Meter builder pattern: https://github.com/open-telemetry/opentelemetry-rust/issues/2164
  - Issue with Builder is that it will make `MeterProvider` non-object safe
  - Alternatives:
    1. For 1.0 timeline, only allow Meter name. And remove the ability to set version/schema…
  - 2. Keep `InstrumentationLibrary` and its builder public
    - MeterProvider.library_meter(instrumentation_library);
    - MeterProvider.meter(name);
      Keep it consistent for all 3 signals
      Simple “name” Tracer,Meter,Logger
      Optional one accepting InstrumentationLibrary

    Concerns about the `Arc` wrapping for the instrumentation library

    Document that the user should store the structures and not create them repeatedly?

    Utkarsh/Fabian/Lalit will discuss in slack and propose changes. This is important for both Logs and Metric API RC release.
- Metrics API Cleanups
  - Want to make sure that the public `opentelemetry::metrics` API is correct
  - Multiple instrument callbacks
    - Fabien: this is important.
      - https://github.com/open-telemetry/opentelemetry-rust/issues/2151
- Metrics SDK Cleanups
  - Tests mostly cover the happy path
  - Decided to drop views from the initial release
    - Histogram bucket customisation, possible at the API level

        - Cardinality control : the spec has ways to handle that in the API, unimplemented yet
- Background thread for `PeriodicReader`
    - https://github.com/open-telemetry/opentelemetry-rust/pull/2142

2024/10/01 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)
- Tom Tan (Microsoft)

Agenda/minutes:

- Release 0.26 today
    - Zhyongyang to review before merge.
- 0.27 planned in 3 weeks
- Internal Logging Improvements
    - Keep expanding `tracing` usage, and remove current global error handler
    - For error!, should it be under feature flag ?
    - Lalit to make current wrappers mirroring the tracing API.
- Background processing
    - Draft for Metrics.
    - No easy way to enforce timeouts, without another background thread.
- Metric API validation
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1699
    - “init”/”try_init” looks odd.
        - Cijo:
            - keep init() after renaming to build(), which returns the NoOp version of the instrument.
            - Remove try_init().
    - build() that always returns instrument

2024/09/24 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)

- Laurent Querel (F5)
- Harold Dost
- Mindaugas Vinkelis

Agenda/minutes:

- MSRV
    - https://github.com/open-telemetry/opentelemetry-rust/pull/2140
    - Why do we have a separate MSRV for what is supposed to be the core exporter?
    - Two reviews ?
        - We'll continue with current process where maintainers use their judgement to decide if a PR needs to wait for 2 reviews.
        - For anything breaking CI - merging asap is recommended.
        - Releases are when things get shipped, so that requires multiple approvals to proceed.
- Release (0.26)
    - Metrics API is ready to be marked Beta (opentelemetry crate)
    - Metrics SDK - likely in the next release, not the imminent one.
- Metrics SDK discussion : https://github.com/open-telemetry/opentelemetry-rust/pull/2117
    - Suggested to open sub-issues for each of the proposed improvements (All are valid ones!)
- Metrics Background thread design
    - Cijo - briefly covered background thread design. For metrics Exporter.Export() is still async and no issues with that. Logs may need revisit.
- [Laurent] Is there anything blocking the approval of https://github.com/open-telemetry/opentelemetry-rust/pull/2098
    - Zhongyang/Cijo will review today!

2024/09/17 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Tom Tan (Microsoft)

Agenda:

- Issue: https://github.com/open-telemetry/opentelemetry-rust/issues/1146
- Patch release - https://github.com/open-telemetry/opentelemetry-rust/issues/2094
- Next release priorities:

        - 1. Metric API move to beta - Utkarsh
            - Prove that API additions are doable in additive way (eg: Histogram Bucket Hints)
        - 2. Metric SDK move to beta - Cijo
            - Remove async runtime dependency
        - 3. Logs remain RC/Beta - Lalit
            - Remove async runtime dependency for BLRProcessor. - Tom
        - 4. Internal Logging - Zhongyang/Lalit
            - https://github.com/open-telemetry/opentelemetry-rust/issues/1146
            - Approach 1 - Use your own abstraction and let user figure out how to pipe through tokio-tracing OR logs or something else.
            - Approach 2 - Offer a feature-flag 'enable-internal-logging-via-tracing' which will leverage `tracing` for internal logs (of all severity). Document to the user that they should apply a filter to prevent OTel internal logs from being piped back into OTel appender tracing. Lalit will prototype this on `opentelemetry-sdk` to start with.
            - Cijo- Voting for Approach 2 for the imminent release.
            - Lalit - Voting for Approach 1 for the imminent release
            - Action: Lalit will prototype Approach 2, and come back with findings.
                - Cijo - ask how other major libraries like tracing itself, and Microsoft internal solutions are solving this problem.
- Tracing vs log - TBD. Log does have one advantage that it can be bridged easily to tracing anyway.
    - The above summarizes the priority for 0.26 release. Everything else is lower priority until 0.26 is done.

2024/09/10 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda:

- Release done, 0.25
- Next release - tentatively targeted for Oct 1st week,
    - Focus on bug fixes and feature gaps
    - Move metrics to Beta
    - Move logs to RC

2024/09/03 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Harold Dost
- Juliano Costa (Datadog)
- Lalit Bhasin (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- [Juliano]: https://github.com/open-telemetry/sig-developer-experience/issues/3#issuecomment-2289546617
- Release
    - Cijo will do the release tomorrow,
    - No blockers from logs, metrics , traces for the release
- Issue - https://github.com/open-telemetry/opentelemetry-rust/issues/2060
- Brief chat about background processing.
    - How about spinning OTel's own background thread and doing aggregation/exporting inside it. (We still keep the existing burying-you-own-runtime option)
    - This may be okay, but we need to fully measure the ramifications.
    - We need to support multiple, parallel exporting. That should not be ruled out.
    - The background threads - they are fixed/capped - 1 for metrics, 1 for logs, 1 for traces, so it is different from repeatedly creating tons of thread.
    - Example of 1 physical core machine - if the background thread is blocked, would the OS take it out and let the other thread run?
      Zhongyang - we need to support parallel exporting, which is already supported in Spans, but not yet in Logs. This was an issue reported by users before, and was fixed for Spans.
      // TODO: Please get the link to the issue, to gain more understanding of the issue.

// What is the need of sync exporting? ETW/user-events
span.end() or error!(....); // by the time this line returns control, the exporting is also done, i.e serialized telemetry is handed over to Operating System Memory.
Fn ()
{

```
        ///code1
        // async thing in here
async().await

}

fn _async()
{
//code2
}


// This is fastest
Fn ()
{
        ///code1
        //code2

}
```

Should we support these both, Or just prefer to support one

OTLP Exporters typically use grpc/http, and typically don't care about few nanosecs here and there.
ETW/User-events exporters typically are done in few nanoseconds, so every ns improvement matters here.

Alternate:
Model etw/user-events exporter, as simply Processors, they don't care if export() is async or not.

2024/08/27 (Tuesday 9:00 AM PT)

Attendees:

- Harold Dost
- Zhongyang Wu

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/issues/1966
    - Harold:

        - I am thinking that as has been discussed we can talk with the tokio folks and see about maybe deprecating `tracing-opentelemetry` and/or having a different sort of bridge `opentelemetry-bridge-tracing-vXX`
            - The idea behind the naming would allow use to move forward with changes in our side while maintaining compatibility with potentially multiple versions of `tracing` crate.
            - This would allow us to not need to maintain complex feature properties within the same crate and allow for simpler flow for compatibility. If you're using XX version of tracing then you use the particular bridge associated, and we can worry about deprecating them without having to worry about some weird hybrid of versioning for this allowing the bridges to live and die on their own. Thinking of Python versions where 2.7 was the zombie that never died and 3.x< 3.6 which passed by without anyone noticing.
- https://github.com/open-telemetry/opentelemetry-rust/issues/2031
    - Zhongyang: I am not sure the making everything async as suggested in this PR is good approach
        - I think the processor should be the boundary between async and sync, which will use channels to mitigate those two. The async part is exporting, the sync part is sending spans/logs/metrics.

2024/08/20 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Utkarsh Umesan Pillai (Microsoft)
- Harold Dost

Agenda/minutes:

- Metrics release
    - Ready for release on Aug 30, changelog update needed to call out improvements
    - All instruments get the same perf now.
    - There are more refactorings required which need breaking API change, that is not planned for Aug 30 release
- Logs release
    - Pending 2 open PRs, we are good for a release.
    - There are other improvements in the pipeline that may come after the Aug 30 release.

        - String allocations
        - Unnecessary Vec
    - Tracing - bridge?
        - Feature flag gated support for tracing:
            - https://github.com/open-telemetry/opentelemetry-rust/issues/1571#issuecomment-2258910019
            - Not feasible for Aug 30 release.
            - [Cijo] to update the issue about the updated timeline.
    - Open PRs/issues
        - Logs

2024/08/13 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin
- Tom Tan
- Utkarsh

Agenda/minutes:

- Metrics SDK updates are progressing.
    - Should be done before the planned Aug 30th release.
    - Cutting scope PR: https://github.com/open-telemetry/opentelemetry-rust/pull/2015
    - Cijo - rearrange the unit-tests based on instruments, threaded. Coverage for up-down counter missing.
- Preallocated attributes:
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1965
        - If we can optimize synchronous export using stack allocated data
        - Benchmark for batch processing, no-op and simple processing
- Next Release:
    - Aug 30th
        - Targeting beta for Metrics

- Open issues?

2024/08/06 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
-

Agenda/minutes:

. Metrics SDK refactor is progressing with good perf gains, expect PR for Counter today, to be expanded to others shortly after.
Discussion about Async Runtime impln for TokioCurrentThread:
It is simply spawning a thread anyway, so why bother with runtime abstraction instead of spinning a background threads

2024/07/30 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas
- Utkarsh

Agenda/minutes:

- Tokio-tracing vs OTel API
    - We recognize it'll be a while before https://github.com/open-telemetry/opentelemetry-rust/issues/1571 can be fully sorted out
    - We continually see issues - upgrades are particularly hard, otel-demo is broken, and users are unsure which versions are compatible
    - Short/medium - have otel-sdk offer tracing integration (via feature flag), and absorb most of the tracing-opentelemetry functionality into OTel SDK itself.
        - This will make tracing integration "our" problem and will be caught in CI checks itself.
        - This will not deprecate `tracing-opentelemetry` (We don't own it to deprecate)
- Next release is tentatively on Aug 30.

Zhongyang will work on the above (native tracing recognition in Otel sdk) for this release.

2024/07/23 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin
- Utkarsh
- Ted
- Tom Tan.

Agenda/minutes:

- Issues:
    - opentelemetry-sdk leaves telemetry.sdk.* resources empty #1940
        - [Utkarsh] Can user use the existing resource detectors to create the set of resource
        - [Zhongyang] Current approach provides flexibility to user to remove the default set of SDK resource.
        - [Lalit] - To look into the specs if resource configuration should be additive.
- Support BatchLogProcessor without async runtime
    -
- Zipkin release:
    - Need to push the release, Version update is done.
- Prometheus release:
    - Prometheus excluded from CI build - https://github.com/open-telemetry/opentelemetry-rust/pull/1792
- [Ted] GC check in
    - More focus on metrics and logs as of now.

2024/07/16 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- .24 released
    - Lalit will push contrib crates except datadog,stackdriver
    - Use same version for all crates
        - Example: sdk did a bump from 0.24.0 to 0.24.1
        - Otlp crate 0.15.0 dependency was 0.24.0
            - OTLP should do 0.16.1 if it bumps sdk to 0.24.1

- tracing-opentelemetry issue
    - Too many spans issue in OTel Demo
        - Cijo will reply with the current known issue and see if tracing-opentelemetry folks can help give guidance
- Metrics plans
    - Cijo/Utkarsh will send PRs with cleanups/refactoring SDK.

2024/07/09 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- [Lalit] To review: Upgrade to hyper/http v1.0 by aumetra · Pull Request #1674 · open-telemetry/opentelemetry-rust (github.com)
- Release plan:
    - Can we move to release and mark Logs as beta?
        - AnyValue improvements - no need of holding off release.
        - Proceed with the release, and mark Logs (API/SDK/APpener/OTLP) as Beta
            - Hold until http v1.0 upgrade PR is also done.
        -

2024/07/02 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Tom Tan (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Release next week.
    - Will do the release July 9th, and move Logs to beta
        - Traces and Metrics status remains the same, but still part of the release next week.
        - Metrics beta will likely be ~ 1 month out, so mid August.
        - Traces - still beta, even with tokio-tracing integration.
    -
- Revisit/revert https://github.com/open-telemetry/opentelemetry-rust/pull/1846/files
        - Remove the part where we use hashset to prevent message being logged more than once - Lalit
    -
-

2024/06/25 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

-
-
- Beta release for logs and metrics -
    - June end ?

    - Open issues.
        -

- PR: Journald exporter for Otel Logs (https://github.com/open-telemetry/opentelemetry-rust-contrib/pull/84)
    - Uses libsystemd (tokio tracing journald subscriber uses direct UDS to write to journald listening socket)
    - Optionally support JSON serialization
    - Different exporter for traces (?)
        - Zhongyang: IIRC there is a journald exporter from tracing(https://docs.rs/tracing-journald/latest/tracing_journald). Would be interesting to see a comparison.
-

2024/06/18 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Cijo Thomas (Microsoft)
- Tom Tan
- Fabien Savy

Agenda/minutes:

- Beta release for logs and metrics - Sticking to timeline of June end.

2024/06/11 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- The discussion about tracing vs otel
    - Support remote parent while creating span
    - Extracting Span ID from tracing spans
    - Need a storage for OTEL metadata
    - Existing working example:
        - https://github.com/TommyCpp/opentelemetry-tracing/tree/main
    - Next step: list of asks
-

Template - 2024/06/04 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Tom Tan (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- The discussion about tracing vs otel will be done next week.
    -
- Discussed open issues/PRs.
-

2024/05/28 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Fabien Savy
- Utkarsh Umesan Pillai (Microsoft)
- Tom Tan (Microsoft)
- Harold Dost

Agenda/minutes:

- Context propagation with `tracing` - OTel SDK
    - Macros: https://github.com/TommyCpp/opentelemetry-tracing/blob/main/src/lib.rs#L13
    - Fix broken trace example:
        - `tracing-opentelemetry` already has the tool ones needed.
        - See https://github.com/TommyCpp/opentelemetry-rust/commit/ed3d5ec71264786f83b5c0285327eb3ef117fe15 for working example
- Builder pattern for `Provider`s vs. `InstrumentationLibrary`
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1527#issuecomment-2123213978
- MRSV version - opentelemetry-otlp dependencies need higher msrv · Issue #1827 · open-telemetry/opentelemetry-rust (github.com)

2024/05/21 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Fabien Savy
- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Handling self-instrumentation (“infinite trace generator”)
    - Options listed by Lalit #761-comment
        1. Use a `tracing` filter (current workaround, not suitable to move forward as it is too restrictive)
        2. “Thread-local” suppression flag (was tried in a PR, does not work for nested futures?)
        3. Ditto, but use a single export thread to work around issues
        4. Tokio-task storage flag
        5. `tracing` `Context` or span `Extension`
    - ACTION(Lalit): look for the solution implemented in other tokio `Subscriber` implementations
    - The problem with option 2 is that uninstrumented `async` blocks will not propagate the `Context` so we’ll emit the data anyways
- Context propagation with tracing - OTel SDK
    - TLDR: It works as it is but it’s awkward

    - ACTION: see if we can write a helper/macro
    - ACTION: try to fix broken trace example
    - ACTION: document progress in dedicated Slack channel
    - https://github.com/TommyCpp/opentelemetry-tracing
- Builder pattern for `Provider`s vs. `InstrumentationLibrary`
    - Instrumentation scope - move to builder pattern? · Issue #1527 · open-telemetry/opentelemetry-rust · GitHub
        - We can just pass the instrumentation library as a parameter to one of the builder functions.

2024/05/14 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
-

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-specification/pull/4049/files - FYI
- Release https://github.com/open-telemetry/opentelemetry-rust/pull/1738
  - Need manual validation of OTLP and Jaeger too.
  - Prometheus text output - visually check.
  - We'll proceed with this release, once manual validation above is done.
  - Cijo - Send a PR calling out http/json is not stable/working, and point to an issue : https://github.com/open-telemetry/opentelemetry-rust/blob/main/opentelemetry-otlp/src/lib.rs
  - Cijo - Ping the original author to see if they can fix.
- Upcoming releases:
  - Metrics : https://github.com/open-telemetry/opentelemetry-rust/issues/1719
  - Logs: https://github.com/open-telemetry/opentelemetry-rust/issues/1733
  - Traces - If no progress in resolving the API part, could we consider adopting tracing-opentelemetry bridge into OTel SDK, to unblock a relatively reasonable path to a stable release for tracing as well?
    - This is a scoped-down version of option3 - https://github.com/open-telemetry/opentelemetry-rust/issues/1571. The missing thing is the interoperability part.
- Tokio-tracing vs OTel tracing - No update. Zhongyang will continue exploring out-of-process propagation.
- Metrics restructuring
- Seattle area Meetup : https://www.meetup.com/seattle-rust-user-group/events/299509369/
  - Cijo, Zhongyang is planning to attend.

2024/05/07 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)

Agenda/minutes:

- Release plan updates
  - Metrics
  - Logs
  - Traces
- Metrics beta/rc/stable release timelines
  - https://github.com/open-telemetry/opentelemetry-rust/issues/1719
  - Planning to remove `Views`
  - Zhongyang: aren't we trying to go too fast and putting unneeded pressure on the project?
  - Cijo: it's time we define formal dates to drive the project forward and show the community progress is being made
- Breaking changes on observable instruments: removed `register_callback`
  - https://github.com/open-telemetry/opentelemetry-rust/pull/1715
  - The API was awkward with the `Any` slices
- Tokio-tracing prototype updates

2024/04/30 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Fabien Savy
- Lalit Kumar Bhasin
- Laurent Querel (F5)
-

Agenda/minutes:

- Fix observable counter cumulative aggregation #1644
  - Asking for Reviews, CLA is cleared now.
- Proposal to Adopt Tokio Tracing as the OTel Tracing API #1689
  - Cijo: showed a proof-of-concept
    - https://github.com/cijothomas/opentelemetry-tracing
    - Relies on a `Layer` implementation, largely inspired `tracing-opentelemetry`
    - Possibilities when a user calls `error!()` inside a `Span`
      - 1. Convert the error event as `SpanEvent` for the current `Span`
      - 2. Convert the error event as `LogRecord` correlated to the current `Span`
      - Both (because the trace backend might not support span events, or just to export to different backends)
  - Please give feedback on the issue
  - ACTION (Cijo): reach out to `tracing` maintainers
- [Logs API] Remove global provider for Logs #1691
- Related discussion:
  - One possibility to further reduce the scope of Logging Signal
    - Remove Log-Bridge API from the opentelemetry crate
    - Modify opentelemety-sdk, to specially treat/handle tracing events.
    - Downside : None will be able to bridge from, say, `slog` to OTel via us.
      - In fact, `log` already has a bridge to `tracing`, so we are not going to disappoint `log` users.
    - Positives: Some perf gains,
      - Free us devs to work on SDK

      (For reference, OTel .NET did exactly this - it treat ILogger specially in its SDK)
- Welcome Laurent!

    - Opentelemetry-weaver maintainer.
    - Tool written in Rust!
- [Laurent] SemConv code generation with Weaver
    - [WIP] Using Weaver to Generate Semantic Convention Attributes Code #136
        - Meant to replace the existing tooling written in Python
    - https://github.com/open-telemetry/oteps/blob/main/text/0243-app-telemetry-schema-vision-roadmap.md

2024/04/23 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Utkarsh Umesan Pillai (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Tom Tan (Microsoft)
- Fabien Savy
- Zhongyang

Agenda/minutes:

- Welcome Utkarsh! (Emeritus maintainer - OTel .NET)
- Maintainers got access to enforce CI checks. Will update and then document changes for OTel compliance.
- [Cijo] Review Release Plans
    - TODO: add link to issue here
- [Cijo] - Updates of tokio-tracing vs OTel tracing
    - Metrics-rs
    - Cannot deprecate our own APIs without consulting OpenTelemetry Technical Committee first
- Open PRs
    - feat(logs): make logger shutdown &self by TommyCpp · Pull Request #1643 · open-telemetry/opentelemetry-rust (github.com)
    - [Logs SDK] Send resource once to processor and exporter, and not for every event. by lalitb · Pull Request #1636 · open-telemetry/opentelemetry-rust (github.com)

2024/04/16 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin
- Toby Lawrence (Datadog)
- Fabien Savy

Agenda/minutes:

- Welcome Toby!
    - Maintainer of `metrics` (personal project) https://github.com/metrics-rs/metrics
    - Cijo reached out to Toby to discuss similarities/collaboration
    - Related to the `tokio/otel` discussion https://github.com/open-telemetry/opentelemetry-rust/issues/1571 with pretty similar possible outcomes
- `metrics-rs` discussion
    - Toby: would prefer keeping both crates, `metrics` is meant to be quite generic
    - Toby: a bridge looks possible
    - Zhongyang: people tend to prefer macro APIs
    - Cijo: macros also came up in the issue related to Key/Value cost
    - ACTION(Cijo): Experiment with the other crates to get a feel of their APIs
    - Toby: macros from `metrics-rs` are less “magic” than `tokio`’s
- Open PRs, shutdown
- Resource - optimize perf
- Metrics-rs
- Release plans/timeline and owners.
- Suppression of nested logs from dependencies
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1330

2024/04/09 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Harold Dost
- Zhongyang Wu

Agenda/minutes:

- Tracing + OTel Tracing interop - Harold is checking
- Cijo+Zhongyang discussed potentially announcing tentative dates for beta/rc/stable release
    - ~3 months for move things to beta (logs,metrics)
    - ~6 months for moving to RC.
    - Unknown for GA date, we can get a good estimate for that once we reach beta state.
    - Cijo - Metrics (specifically the sdk side) is much more complex than other signals, and would want to cut down features to accelerate beta/rc/stable.
    - Cijo - will open a PR with possible timelines, and bring to discussion / review at the next tuesday meeting.

2024/04/02 (Tuesday 9:00 AM PT)

Attendees:

- Lalit Kumar Bhasin
- Zhongyang Wu
- Fabien Savy
- Bhargav Voleti

Agenda/minutes:

- Shutdown updates
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1643
    - Currently, shutdown takes a `&mut self` for loggers
        - Would like to take a `&self` instead
    - Future work? Make the shutdown asynchronous
    - Bhargav: we need to keep cancellation in mind (what happens if the future is dropped?)
    - ACTION: Bhargav to take a look and review the PR
    - ACTION: Zhongyang to start an issue discussing how to handle global providers
- Resource Propagation optimization:
    - [Logs SDK] Send resource once to processor and exporter, and not for every event. by lalitb · Pull Request #1636 · open-telemetry/opentelemetry-rust (github.com)

2024/03/26 (Tuesday 9:00 AM PT)

Attendees:

- Harold Dost
- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Fabien Savy

Agenda/minutes:

- OTel Demo app https://github.com/open-telemetry/opentelemetry-demo/pull/1433#discussion_r1534870389
    - Missing `spanid` and `traceid` in logs
- Contrib CI broken : https://github.com/open-telemetry/opentelemetry-rust-contrib/pull/57#issuecomment-2015740499
    - Cijo will raise a request to get this fixed.
- Discuss https://github.com/open-telemetry/opentelemetry-rust/issues/1642
    - Concerns about the performance impact of `KeyValue`
- https://github.com/open-telemetry/opentelemetry-rust/pull/1643
- https://github.com/open-telemetry/opentelemetry-rust/pull/1644

2024/03/19 (Tuesday 9:00 AM PT)

Attendees:

- Zhongyang Wu
- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)

Agenda/minutes:

Shutdown issue

- Shutdown discussion : https://github.com/open-telemetry/opentelemetry-rust/pull/1623#issuecomment-2002014018
- Another shutdown issue
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1625#issuecomment-2002680753
- https://github.com/open-telemetry/opentelemetry-rust/blob/main/opentelemetry-sdk/src/logs/log_emitter.rs#L249
    - As shown in this test, shutdown has no effect, unless every Logger is shutdown. This is possible only if app owner controls every Logger, But not feasible in practice, as libraries maybe holding Loggers, preventing shutdown.
    - (Same issue for TracerProvider too)
- Zhongyang:
    - Tracer can hold owned TracerProvider/Inner, and inner can hold weak ref to Tracers. Shutdown requires upgrade of weak->Arc, causing perf issues, but okay, as shutdown is a rare event.

Global_signal_Shutdown/Flush issue

App owners typically have the need to set the provider to Global, so that library/other-components can do Global:Meter/Tracer/Logger.

This forces app owners to give up ownership of the provider they created.

This means, app owners have no way to call force_flush() or shutdown().

Global_provider_shutdown (on opentelemetry crate), “indirectly” achieves shutdown() effects, by causing the original provider to be dropped.

- This “indirect” shutdown, still does not allow force_flush(), so the issue is not resolved.
- Its bit un-spec to add a shutdown API in the opentelemetry crate, given it is a pure sdk aspect.

Some suggestions:

What if we provide shutdown_signal() in the SDK crate, and app owners can call it....

How does this get a hold of the provider that was set in the API...

Metrics specific shutdown

Shutdown, does not cleanup/re-claim memory used for aggregators. No metrics are exported, but the memory is unusable.

Discussion about Key,Value

Tracing has high perf, when no listener, the key/value are not even evaluated. On the contrary, OTel always evaluates, leading to non-trivial perf hit.

Normally not an issue for app developers, but likely a blocker for library authors, who expect 0/near-0 overhead when nothing is enabled.

If OTel does not solve this, there will be no way libraries bet on OTel API, and they'll stick with or onboard to tracing, forcing us to do #2 from https://github.com/open-telemetry/opentelemetry-rust/issues/1571, for not only tracing but all 3 signals.

This is already shown in https://github.com/open-telemetry/opentelemetry-rust/blob/main/opentelemetry/benches/attributes.rs

2024/03/12 (Tuesday 9:00 AM PT)

Attendees:

- Zhongyang Wu
- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)
- Fabien Savy
- Roger Coll (NewRelic)

Agenda/minutes:

- Welcome Roger! (NewRelic)
- `async` and `shutdown` situation
    - What does the specification say about `shutdown`?
    - Should we work towards a unified runtime abstraction and fix current (incorrect) blocking behavior?
        - [Zhongyang] Not sure I'd call it incorrect. Error prone as users usually don't realize the blocking nature of the call
    - Current architecture based on background jobs based on the `Runtime` abstraction
    - Two possible directions
        1. Roll our own runtime
        2. Make `shutdown` async (should be less error-prone)
            - Problem: some components do not support `async`
    - Cijo: Why do we need a runtime?
    - ACTION(Zhongyang): Experiment with async `shutdown`
        - Conflict with the `Drop` behavior, need to document the necessity to call `shutdown` but it's less error-prone than silently blocking
        - Note: .NET `dispose` flushes the traces, C++ also
- OTLP implementation (specification)
    - Acknowledgements
    - Retries
- Stalled issues and PRs

    - https://github.com/open-telemetry/opentelemetry-rust/pull/1056 (to close)
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1089 (to discuss)
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1361
        - Superseded by https://github.com/open-telemetry/opentelemetry-rust/pull/1567
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1515
        - To reconsider, `add_link` is now stable in the specification
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1330
- PRs/Issues needing discussion
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1610
        - See comments in the PR
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1564
- Call for review
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1608

2024/03/05 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)
- Zhongyang Wu
- Fabien Savy
- Bhargav Voleti
- Harold Dost

Agenda/minutes:

- OTel vs `tracing` API discussion status?
    - No recommendation at this stage, not enough data to make a decision
    - Zhongyang: might use a 5th option, maintaining both APIs but with no overlap?
        - Cijo: Kinda similar to option 2?
        - ACTION(Cijo): will edit the issue to reflect this
    - Need to clarify the path forward for missing APIs in both crates
        - This is a discussion that need to happen with `tracing` maintainers
    - The OTel API could take the form of wrapper macros
    - Harold : One thing to note is that tracing is a synchronous API and OTel is an async API or at least relies on async runtimes.
- Pending pull requests
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1564
- Questions
    - Why do we have our own benchmark utility instead of using `criterion`?
        - ACTION(Bhargav): investigate `divan` which has throughput testing

        - https://crates.io/crates/divan
- New issues
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1595
        - Inter-process traces
        - ACTION(Fabien): respond to the issue, talk about propagation
- https://github.com/open-telemetry/community/issues/1857 << Roadmap 2024
- March 10 - Daylight saving 🕒

2024/02/27 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Tom Tan (Microsoft)
- Fabien Savy
- Harold Dost
- Lalit Bhasin (Microsoft)

Agenda/minutes:

- Welcome Fabien!
- Welcome Tom!
    - Maintainer for the C++ SIG
- Update on releases
    - Maintainers have been given access to publish releases
    - Contrib repos PR: https://github.com/open-telemetry/opentelemetry-rust-contrib/pull/41
- Stable release update
    - Overview issue of blockers for the stable release: https://github.com/open-telemetry/opentelemetry-rust/issues/1572
        - Please provide feedback :)
    - `tracing` vs OTel issue is a blocker: https://github.com/open-telemetry/opentelemetry-rust/issues/1571
        - Requires some expertise working with both crates, help appreciated
- Open PRs
    - Release process: https://github.com/open-telemetry/opentelemetry-rust/pull/1555
        - Stuck on version semantics
        - Harold: we should probably not go with unified versions
        - Thomas: unified versions makes it easier to perform a release (no need to specify the versions of each crate)
        - Zhongyang: we could use the API/SDK version as an anchor
        - Thomas: unified versions will force us to make “empty” releases
        - Lalit: What about hotfixes? Thomas: we’ll use the patch number
        - *Proposal*: unified version numbers but skipping “empty” releases?
        - Lalit: Did we look at the ecosystem?
        - No consensus reached
- Tracing vs OTel Tracing discussions
    - Take a look at (and participate in) the thread in the dedicated Slack channel

- `async` handling in rust-otel crates
    - Zhongyang: working on some POC on the idea to unify the async tasks in Rust
        - The problem here is that we need some sort of runtime for background tasks in Otel. One example is batch and exporting in BatchSpanProcessor
        - The goal is build a runtime that can
            - Drive async tasks in either runtime
            - Can manage its own lifetime
- CNCF Meet and greet
    - Meeting up at Paris on Wednesday

2024/02/20 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Matthew Shapiro
- Zhongyang Wu
- Bhargav Voleti
- Lalit Bhasin

Agenda:

- https://github.com/open-telemetry/opentelemetry-rust/issues/1395
    - The current implementation is problematic because it relies on blocking, which is incorrect in an `async` context.
    - Proposal for OTel-rust to have its own runtime
        - Concerns of conflicts if using `tokio` because we cannot register a global runtime
        - The difficulty lies in complying with the specification because flushing must be a blocking operation?
        - Even if we rely on the current `Runtime` abstraction, `spawn_blocking` cannot be achieved with a single thread.
        - So we would probably end up having a few threads for OTel operations
    - Q: Why is it synchronous today?
        - Compatibility with synchronous world (the API was first designed without `async` in mind)
        - Also, we want to make sure the flush completed
    - Q: Do we really want to call `shutdown` when `drop`ping the structure?
        - The opposite would be surprising if it was not the default because of lost data
- Open PRs
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1563
    - Q: Why are we enforcing a limit on the number of data points?

        - The specification requires memory usage to be bounded
    - Q: Are there benchmarks for memory usage?
        - No, for now we focused on CPU usage
    - Cijo proposed to reserve some time to go over the current metrics design
    - Cannot use a channel because the aggregation has to happen directly (the spec will probably be updated this month to mark it explicitly non-compliant)
    - The current implementation stores the aggregations in a hashmap behind a lock
    - Concurrent data structures - dashmap, flurry.

- Release
    - Q: Did we do a release last week?
        - No, but everyone should have access now
    - There's a version bump PR https://github.com/open-telemetry/opentelemetry-rust/pull/1539

2024/02/13 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Matthew Shapiro
- Lalit Bhasin

Agenda:

- Performance work (https://github.com/open-telemetry/opentelemetry-rust/pull/1519)

2024/02/06 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang Wu
- Matthew Shapiro

Agenda:

- Release from contrib
    - The issue with 404 can be patched by revising the repo attributes in the Cargo.toml files. Zhongyang will propose a PR for it.

        - How to add teams to Cargo https://doc.rust-lang.org/cargo/reference/publishing.html#cargo-owner.
    - Metrics API - attributeset PR

2024/01/24 (Tuesday 9:00 AM PT)

Attendees:

-
- Cijo Thomas (Microsoft)
- Lalit Bhasin

Agenda/minutes:

- Release from contrib, all readme points to 404 from old repo
    Zhongyang
- Project plan/milestones?
    - Cijo to create milestones (some exist already), for specific components like API/SDK/Exporter separately, instead of overall milestones.
- Use of sync::Mutex in async contexts in codebase.

2024/01/23 (Tuesday 9:00 AM PT)

Attendees:

-
- Cijo Thomas (Microsoft)
- Matthew Boddewyn (Microsoft)
- Bhargav Voleti
- Zhongyang Wu

Agenda/minutes:

- Welcome Matthew Boddewyn!
    - Starting with help on deprecating jaeger exporter https://github.com/open-telemetry/opentelemetry-rust/pull/1487
- Lalit: Permission for publishing of the crates in the contrib repo.
    - Action item: Reach out to add Cijo or Lalit to event* crates
    - Cijo - create a issue to discuss how to handle crate releases
- event_name propagation to attributes

        - We can add it to LogRecord as a top level field.
        - Exporters can choose to decide how to export it.
            - OTLP Exports as attributes, but the keyname is to be decided.
            - Java exports it as instrumentation scope in OTLP Exporter.
- InMemory*Exporter modules are doc hidden, giving the impression that they are not for public use, and only for internal testing. Refer - https://github.com/open-telemetry/opentelemetry-rust/issues/1466
    - Lalit will fix the doc and close the issue.
- Http upgrade discussion - https://github.com/open-telemetry/opentelemetry-rust/issues/1427

2024/01/16 (Tuesday 9:00 AM PT)

Attendees:

-
-
-

Agenda/minutes:

- http 0.2 vs http1.0 https://github.com/open-telemetry/opentelemetry-rust/issues/1427
    - Potentially we can drop opentelemetry-http crate entirely
    - We probably can't upgrade without tonic being upgraded.
    - Potentially could rely on Tower?
        - @KallDrexx mentioned he had some success with this solution so we can probably take some learning from it and do it in opentelemetry

2024/01/09 (Tuesday 9:00 AM PT)

Attendees:

- Harold Dost
- Shaun Cox (Microsoft)
- Bhargav Voleti (Microsoft)

Agenda/minutes:

- Upgrade weak https://github.com/open-telemetry/opentelemetry-rust/issues/1209
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1455

        - We'll go don't do that, but we need to verify the shutdown procedure works.
        - We still need to write a test, so that we understand how shutdown should function.
            - We may want to keep track of the shutdown state (as RwLock or Atomic Bool) Prefer Atomic Bool since it's not used until the end "one way door"
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1446
    - Bhargav will take a look into doing a similar patch for the TracerProvider.
        - Previously there was a circular dependency which caused issues because Span held a reference to the TraceProvider.
- We need a review on:
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1444
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1421
- Future: How do we want to manage bug fix release.

2023/12/19 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Matthew Shapiro (Microsoft)
- Harold Dost
- Bhargav Voleti (Microsoft)
- Lalit (Microsoft)

Agenda/minutes:

- SIG meeting cancellation - Dec 26, Jan 2.
    - Will cancel.
- Pre Creation of AttributeSets
  https://github.com/open-telemetry/opentelemetry-rust/pull/1421
    - We're going to look at moving towards `impl Cow<&str>`

2023/12/11 (Tuesday 9:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)
- Shaun Cox (Microsoft)
- Matthew Shapiro (Microsoft)
- Bhargav Voleti (Microsoft)
- David Barsky (Facebook)
- Kevin Minutti
- Zhongyang Wu

Agenda/minutes:

- Welcome Bhargav!
- Welcome David (again)!
- Welcome Kevin (again)!
- Cijo - Dec 26, Jan 2 SIG meetings - check if we should cancel. (OTel Maintainer/Community calls are canceled for these dates)
    - Lot of OOF on Dec 26th
- PR 1410: Add experimental synchronous gauge (On experimental feature flag)
  Issue 1411 : Proposal: Naming for features
- Issue #1427: How to handle http 1.0 upgrade
- Cijo : Tokio Tracing API vs OTel Tracing API.
    - Continuing from the previous week.
    - https://www.cncf.io/blog/2019/05/21/a-brief-history-of-opentelemetry-so-far/
        - OTel itself was formed after merging Census and Tracing in 2019.
    - .NET's Activity API was similar to tokio/tracing (in 2019)
        - Later enhanced to match OTel Tracing API spec ~90%, and OTel .NET repo killed its own API in favor of .NET's own API for Tracing, and later continued the same idea for Metrics.
    - Two competing APIs are bad for the ecosystem, new users have to pick between them.

    - Tokio-Tracing is extremely popular, most people are going to start with that to achieve in-proc tracing, and eventually learn about OTel Rust.
    - Tracing is more mature, it'd be good to make it work nicely in Distributed Tracing and make it easy/efficient to propagate Context.
        - The current way of bridging them is affecting perf.
    - Zhongyang, Bhargav can help with initial steps, Shaun,David can offer support as well, especially with maintaining perf.
        - The https://github.com/open-telemetry/opentelemetry-rust/discussions/1032#discussioncomment-6223345 perf benchmarks need to re-done, given the large set of changes in OTel Tracing API/SDK since.
- https://github.com/tokio-rs/tracing-opentelemetry/tree/v0.1.x - the repo bridging tracing -> OTel
    - https://cloud-native.slack.com/archives/C069U408RNW Slack channel dedicated to the discussion of OTel Tracing vs Tokio-Tracing
- https://github.com/open-telemetry/opentelemetry-rust/issues/1189 - Discussion about
- Main branch `cargo test` compile errors https://github.com/open-telemetry/opentelemetry-rust/issues/1439
-

2023/12/05 (Tuesday 8:00 AM PT)

Attendees:

- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Shaun Cox (Microsoft)
- Matthew Shapiro (Microsoft)
-

Agenda/minutes:

- Harold: Logging and Tracing Attributes: https://github.com/open-telemetry/opentelemetry-rust/issues/1300#issuecomment-1831580839
    - There were changes which removed the deduplication, and again while we've stated performance is definitely a goal, I am wondering if we should have some level of consistency across the APIs such that once we have our determination on the Metrics we can act similarly on Logs/Tracing.

- Cijo : Tokio Tracing API vs OTel Tracing API.

    - All agree this is an issue and needs to be addressed
    - Lets start with documenting the fundamental differences:
        - Sampling (OTel is more flexible allowing attributes to be part of sampling decision)
        - Tokio-Tracing - more geared towards in-proc tracing, whereas OTel is more towards out-of-proc, dealing with propagation/restoring context etc.
    - Zhongyang can look at checking how much Tracing Spec compliance would tokio-tracing have today.
    - Would need to get more folks involved especially from Tokio Tracing group!
- More topics

2023/11/28 (Tuesday 4:00 PM UTC)

Attendees:

- Harold Dost
- Shaun Cox
- Cijo Thomas
- Kevin Minutti
- Julian Tescher
- Zhongyang Wu
- Leighton Chen

Agenda/minutes:

- SIG TIme https://cloud-native.slack.com/archives/C03GDP0H023/p1699374697857159
    - Harold is working on updating the entry, we can assume starting next week we move to the 9AM PT time.
    - We're currently in conflict with the Spec SIG.
- Perf improvements.
    - Cases we don't cover are with dynamic attributes.
        - The main issue is in hot loop
    - Potentially the metrics
    - Julian Propose we use AttributeSet as input for user
        - Consider creating a guide for high performance use cases.
    - Today we take a slice, and we do a lot of alloc and dealloc as a result of the Slice API.
    - Questions:
        - Should we remove the slice increment API? Yes
            - This moves the decision point earlier and allows the user to manage how they want to deal with the performance.

            - Should we take a reference to the AttributeSet or the AttributeSet value?
            - Users would just need to add `into()`
        - Bounded could be approached after the pre-created AttributesSets are approached.
        - **Consensus: Matt will move forward with his change on exposing add using AttributeSet instead. After that review we'll continue.**
- Metric View - question (Cijo)
    - We can potentially refactor how we generate using views, exposing Aggregation be traits.
    - **Consensus: We'll remove the existing example, and we can propose alternatives moving forward. Cijo will update his PR and create an issue to follow up on the alternatives.**
- Integration Tests. - tracing-opentelemetry
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1340
    - This is actually due to a mismatch in version
    - Potentially we could provide a compatibility matrix?
    - Basically getting to 1.0 will help with some of these things.
    - Consider we should change the drop behavior to 1.0?
    - Harold will follow up.

2023/11/21 (Tuesday 4:00 PM UTC)

Attendees:

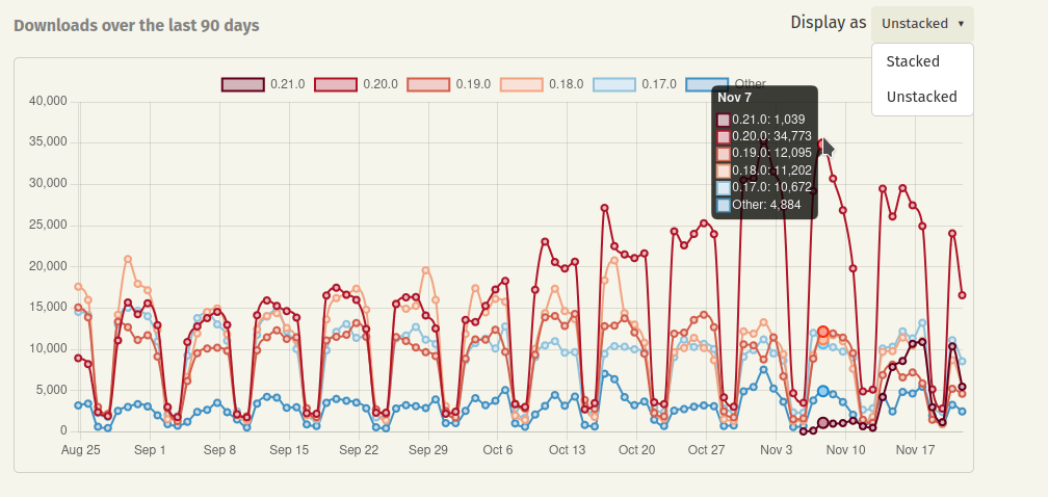
- Harold Dost
- Shaun Cox
- Cijo Thomas
- Mathew Shapiro
- Julian Tescher
- Zhongyang Wu

Agenda/minutes:

- Deduplication of attributes on metrics
  https://github.com/open-telemetry/opentelemetry-rust/pull/1379
- What is the direction that we want to take for the project?
    - Should we be blazingly fastest project?
    - Should we be a minimal spec version?
    - Harold: My opinion is that we don't have to best 100% spec compliant if it's detrimental to performance. We can hide things that are necessary
    - Shaun: We should be as fast as possible to avoid splitting the community.

    - ZhongYang: We should follow the MUSTs but SHOULDs and MAYs don’t need to be there.
    - Julian: We shouldn’t focus on the casual user.
        - A large amount of time is spent on the start and end. We could reconsider things like how time is handled in tracing.
            - Example: https://github.com/tikv/minitrace-rust for tracing performance comparisons
    - Consensus: We want to focus on performance. Short Term: We may want to consider moving AttributeSet into `opentelemetry` this would allow users to cache AttributeSets and/or we could look into bounded instruments.
    - Could we look at a “compat” flag which allows for better performance? Not necessarily spec compatible.
    - How can we handle interop? A number of issues exist.
        - tracing Crate
        - logs crate
    - Areas we should continue to consider:
        - Performance
        - Spec Compliance
        - Ecosystem
        - Interop (within Rust)
        - Interop (with other langs?)
- Can we look at improving the actual counting by removing locking?
    - Shaun to create an issue. #1386 (Just a stub for now. Needs more context/details.)
    - Take a look at the `minitracing` mentioned above.
- Crates.io Access - Harold
    - Currently we have a blocker between Crates.io and Github. There's a pending request.
    - PRs setting
- Release Cadence

- We need to be conscious of releasing too much as we can end up with breaking changes that happen as a result of differences.



- What would be a good cadence?

2023/11/14 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- Propagator API([issue 1013](#))
  - Finished investigation. I'd recommend to move the composite propagator to API but keep others in the SDK
- Is opentelemetry-rust-contrib released? (Having difficulty getting it to build alongside opentelemetry consumed as a fork because it depends on a released opentelemetry, not my fork)
  - Harold: i'm working on a patch to fixup the build

2023/10/31 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas
- Shaun Cox
- Zhongyang Wu
- Lalit Bhasin

Agenda/minutes:

- Triage new tickets
    - Does it need to be in one of the milestones?
    - Does it need more information?
- PR:
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1315

2023/10/24 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Lalit Bhasin
- Shaun Cox
- Zhongyang Wu
- Harold Dost

Agenda/minutes:

- [Morgan] Remember to vote in the governance committee elections
    - https://vote.heliosvoting.org/helios/elections/1ee70ee4-11ce-11ee-aaf8-0a8c9aac83f9/view
    - Check voting eligibility here https://github.com/open-telemetry/community/issues/1561
- [Morgan] Kubecon: we have a meeting / discussion space called the OpenTelemetry Observatory that anyone in the community can sign up for: https://docs.google.com/forms/d/1MaUy6i5hASesz7QxAvrdBMQICdJMxGOXKvsi6bWmNPQ/viewform?edit_requested=true
    - For example, someone could host a feedback session or walkthrough of the updated semantic conventions

- Backlog Refinement for API Stability
    - We went through the existing API Milestone and pruned it, next week we'll look to go through un-triaged issues and ensuring they don't threaten stability/need to be completed.
- For deviation please mention in https://github.com/open-telemetry/opentelemetry-rust/issues/1297, so we can draft

2023/10/17 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Harold Dost
- Zhongyang Wu
- Lalit Bhasin

Agenda/minutes:

GA planning (Cijo)

Proposing to track API, SDK, OTLP Exporter separately.

There is no need to do tracing first, as all signals are stable from spec. (unlike other languages who had to following tracing followed by metrics as spec was evolving)

Possibly prioritize Tracing API, Metrics API, Logging Bridge API so as to give more end users start taking a bet on us. Also helps in conveying the right message to all - that OTel Rust is going to get GA in a TBD date. Followed by SDKs, and OTLP Exporters. (And deprioritize everything else to keep focus)

Zhongyang: Still prioritize Tracing API *and* SDK. before other signals.
Harold +1.

Cijo - Tracing API > Metrics API > Tracing SDK -> Metrics SDK/Log SDK -> Tracing OTLP -> Metrics OTLP

Harold: Time between is too long, so people may not feel contributing , as they are not seeing any immediate benefit. So a fixed cadence is important.

Need to make sure more than 1 person has permission to publish.

Is the process manual/labor intensive?

Not good test coverage so a simply release won't work..

Things leftover from PR are not easy to track.

https://crates.io/crates/release-plz << Could be used?

PRs : https://github.com/open-telemetry/opentelemetry-rust/pull/1293

Action Items

- Improve Test Coverage - Cijo to send a PR showcasing In-MemoryExporter.
    - Zhongyang might explore spinning collector instances for doing integration tests. Done in other languages - .NET - simply check if export is successful or not.
    - Cpp did similar.
- Provide Maintainers access to publish - Zhongyang
- Explore ways to make publishing crates easier like https://crates.io/crates/release-plz - Zhongyang
- Create separate milestones for API/SDK/Exporter for each signal. - Cijo
    - Every SIG meeting should do the triaging and assign to milestones/owners.

2023/10/10 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Shaun Cox (Microsoft)
- Leighton Chen (Microsoft)

Agenda/minutes:

1. Discussed the implementation of enforcing spanattribute limits etc in the SDK.
    a. The earliest when SDK can enforce the limit is at SDK, but the memory is already allocated, so we won't really protect the user from OOM.
    b. The current implementaion's Truncate() on Vec! Is not going to free any memory, so that won't really solve any potential OOMs.
    c. Its best to move this to Exporters. (So that exporters can enforce limits so as to not get throttled/dropped by backends)
2. Potential for a more ergonomic API to provide attributes at span creation.

Instrumentation has a way to provide attributes at the span_creation, in 2 separate buckets - one available for sampler, and the other made part of span_attributes post sampling.

```
  let span_builder = SpanBuilder::from_name("test_span").with_sampling_attributes(vec![
        KeyValue::new("attribute_at_span_start1", "value1"),
        KeyValue::new("attribute_at_span_start2", "value2"),
    ])
with_attributes(vec![
        KeyValue::new("attribute_at_span_start1", "value1"),
        KeyValue::new("attribute_at_span_start2", "value2"),
```

....
....
....
]);

2023/10/03 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- (Cijo) Question about EvictedQueue/IndexMaps/EvictedHashMaps..
    - Discussed about refactorings needed to make the perf much better.
    - Also discussed the idea of removing de-dup logic from SDK or make it opt-in.
    - Zhongyang suggests to make sure we dont make user's life complicated by making config too complex!.
    - We need to balance the perf, adhering with the Rust spirit of zero cost abstractions, and ease of use at the same time
    - Cijo will open specific issues with perf numbers demonstrating the impact.
    - Briefly discussed the idea of splitting stabilizing API/SDK/OTLPExporter into separate milestones, with the idea of API getting stable earlier, allowing a wider audience to start using OTel Rust, and share feedback.

2023/09/26 (Tuesday 4:00 PM UTC)

Attendees:

- Zhongyang Wu
- Shaun Cox (Microsoft)
- Leighton Chen (Microsoft)
-

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/1267
- Reviewed the open issues and labeled them

-

Template - 2023/09/19 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Leighton Chen (Microsoft)
- Zhongyang Wu
-
-

Agenda/minutes:

- A few PRs are ready to be merged
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1268
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1270
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1274
- Discussion on whether we should prioritize env vars or compiling time value
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1225
- Please take a look at this PR
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1272
- Require >2 approval

2023/09/12 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Leighton Chen (Microsoft)
- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Lalit Kumar Bhasin(Microsoft)

Agenda/minutes:

- Feasibility of achieving performance goals with current api abstractions?
    - I have before/after performance improvement data to show/share
    - Related to https://github.com/open-telemetry/opentelemetry-rust/pull/1258

    - We may be hitting the limit of perf optimization without some change in spec itself. The current perf (especially for the ones when no sdk or no-from-sampler) is unacceptably low for high-perf libraries. The general agreement is we'll confirm that we cannot fix these in Rust without violating spec - at that stage we can reachout to spec owners and seek guidance (be it choosing to violate spec OR alter spec itself or relax spec wording..)
    -
- https://github.com/open-telemetry/community/issues/1674

2023/09/05 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Leighton Chen (Microsoft)
- Cijo Thomas (Microsoft)
-
-

Agenda/minutes:

- https://github.com/open-telemetry/opentelemetry-rust/pull/1232
    -
- https://github.com/open-telemetry/opentelemetry-rust/pull/1246
    - We need benchmark to see if there is perf improvements
    - What's the difference between return a mutable reference and return a object in builder pattern

2023/08/29 (Tuesday 4:00 PM UTC)

Attendees:

- Harold Dost
- Shaun Cox (Microsoft)
- Cijo Thomas (Microsoft)

Agenda/minutes:

- Shaun: Curious on others thoughts on [api proposal by shaun-cox · Pull Request #1232 · open-telemetry/opentelemetry-rust (github.com)]
    - How is sampling expected to work?
        - Library:
            - let span = tracer.StartSpan("name of span", OptionalAttributes) // Shaun - the cost of even invoking sampler is not acceptable when writing high perf libraries.
            - If (span.IsRecording == true)
            - {
                - span.AddAttribute/AddEvent...etc.
            - }
    - Goal is to allow library developers to create spans only when needed or desired by the application.
    - Current issues is that there's a significant overhead on generating a span, and there's a fair amount of boilerplate to reduce the performance impact.
    - Shaun: When working with:
        - Tokio tracing + tracing-opentelemetry + opentelemetry sdk seems more expensive opentelemetry + opentelemetry sdk
    - Cijo: We may want to look into how we could improve sampling performance.
    -
- Wonder if there would be benefit to changing opentelemetry dependency from futures-util to futures-core + futures-sink. futures-util, like futures, is a wrapper crate. E.g. see [Futures — async Rust library // Lib.rs]
-

2023/08/22 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Shaun Cox (Microsoft)
- Aaron Clawson (Lightstep)
- Leighton Chen (Microsoft)
- Adam Kaczmarski
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- Welcome Adam!
    - https://github.com/open-telemetry/opentelemetry-dotnet/blob/main/CONTRIBUTING.md#find-a-buddy-and-get-started-quickly
    - https://communityinviter.com/apps/cloud-native/cncf
- https://github.com/open-telemetry/opentelemetry-rust/issues/1209
- https://github.com/open-telemetry/opentelemetry-rust/pull/1199
    - Cijo to ping Zhongyang to see the order of PR
    - waiting for fix(zpages): use tonic based generated files. by TommyCpp · Pull Request #1214 · open-telemetry/opentelemetry-rust (github.com) -- This should go first, before 1199.
- https://github.com/open-telemetry/opentelemetry-rust/pull/1210
    - Need to create a separate issue w.r.t priority of Env vs Code
    - Code is preferred - Zhongyang, Cijo agrees with this.
        - Shaun is not objecting if we advise users to configure anything in code and leave it to ENV.
    - **- What is more idiomatic in Rust.**
    - - .NET, Go, Python, follow this.
    - Env var overrides code - Shaun agrees with this.
    - - C++ follows this.
    -
-

2023/08/15 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Cijo Thomas (Microsoft)
- Harold Dost
- Aaron Clawson (Lightstep)
- Zhongyang Wu

Agenda/minutes:

- Merging PR https://github.com/open-telemetry/opentelemetry-rust/pull/1199 after https://github.com/open-telemetry/opentelemetry-rust/pull/1202

    - Blocked on zpages, we'll disable the zpage on build, and we can followup on whether it should be removed.
        - https://github.com/open-telemetry/opentelemetry-dotnet/pull/4604
        - https://github.com/open-telemetry/opentelemetry-specification/pull/3577
        - https://github.com/open-telemetry/opentelemetry-cpp/issues/2217
- Anyone explored the use of (e.g. https://docs.rs/bumpalo/latest/bumpalo/) for allocating logs/tracing/metrics records as they pass from instrumented app to the exporter?
- https://github.com/open-telemetry/opentelemetry-specification/blob/main/specification/protocol/exporter.md#specify-protocol

2023/08/08 (Tuesday 4:00 PM UTC)

Attendees:

- Shaun Cox (Microsoft)
- Harold Dost
- Zhongyang Wu
- Aaron Clawson (Lightstep)
- Cijo Thomas (Microsoft)
- Lalit Bhasin (Microsoft)

Agenda/minutes:

- Harold: Labeling - I added a few different labels
    - Triaging: triage:accepted, triage:needmoreinfo
        - Close as not planned can be used instead of `triage:rejected`
    - Versioning: version:breaking, version:minor, version:patch
    - Platform: platform:linux, platform:mac, platform:windows
    - Priority: priority:p0, …, priority:p3
- https://github.com/open-telemetry/opentelemetry-rust/pull/1193
    - Congrats!
- "Api" vs. "sdk" and crate granularity https://github.com/open-telemetry/opentelemetry-rust/issues/1186
    - E.g. should we have just one opentelemetry crate, with api vs sdk isolated to modules, or should we consider opentelemetry crate represent the api, and have an opentelemetry_sdk crate for sdk concerns? The latter is what seems to be preferred by, as example, the (tokio) tracing ecosystem.
    - We should look at deprecating the `opentelemetry` crate
    - **Action:** Write up the plan about deprecating the opentelemetry
- Stdout should we split things out from a Stdout and FileExporter?

2023/08/01 (Tuesday 4:00 PM UTC)

Attendees:

- Harold Dost
- Cijo Thomas (Microsoft)
- Shaun Cox (Microsoft)
- Lalit Kumar Bhasin (Microsoft)
- Zhongyang Wu
- Aaron Clawson (Lightstep)

Agenda/minutes:

- Release Cadence
    - What's a good cadence? Potentially quarterly
    - We should start to look at adding labels to mark changes as breaking on PRs to avoid breaking the stability guarantees.
        - Look for some prior art.
- Tickets to implement https://github.com/open-telemetry/opentelemetry-rust/issues/977
- Stability
    - We initial talked about this **2023/04/07**
        - We should also ensure we validate the calls and how they affect performance as this can affect the API and we don't want to be stuck with
        - Finish what we think should be complete (See notes from the above mentioned meeting
        - Validate the performance
            - What are the important areas we need to focus?
        - Reach out to TC Members.
    - We can update https://github.com/open-telemetry/opentelemetry-rust/issues/977 to reflect this.
- Adding new functionality even if it's not part of the spec.
    - Feel free to create an Issue or PR

2023/07/25 (Tuesday 4:00 PM UTC)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Cijo Thomas (Microsoft)
- Shaun Cox (Microsoft)
- Zhongyang Wu

- Leighton Chen (Microsoft)
- Aaron Clawson (Lightstep)

Agenda/minutes:

- Release: [Prepare for v0.20.0 release by jtescher · Pull Request #1156 · open-telemetry/opentelemetry-rust (github.com)]
    - Its been a while since the last release.
    - Metrics have complete rewrite, so best to get it out.
    - Logs are totally new, and okay to have breaking changes in next release.
    - Tracing already has had a bunch of breaking, there *could* be more, but its not worth waiting for all such changes as that'll be an indefinite wait..:D
        - Shaun has improvements to fix KeyValue as Tuple, Public API breaking change.
          Give this PR 1-2 days to see if can reach an agreement, else we should proceed with release.
          https://github.com/open-telemetry/opentelemetry-rust/pull/1089
- For review : [Design Proposal to export logs only if the level/logger/target is enabled. by lalitb · Pull Request #1147 · open-telemetry/opentelemetry-rust (github.com)]
    - Okay to make it part of the coming release.
    - Few design nits discussed, nothing blocking.
        - Potentially cache the Is_Enabled to avoid syscall in the hotpath.
        - Zhongyang can help review from Approvers/Maintainers group.
        - Make sure all changes are under the feature flag.

2023/07/18 (Tuesday 4:00 PM UTC)

Attendees:

- Lalit Kumar Bhasin (Microsoft)
- Cijo Thomas (Microsoft)
- Leighton Chen (Microsoft)
- Aaron Clawson (Lightstep)

Agenda/minutes:

- Release: [Prepare for v0.20.0 release by jtescher · Pull Request #1156 · open-telemetry/opentelemetry-rust (github.com)]

- For review : Design Proposal to export logs only if the level/logger/target is enabled. by lalitb · Pull Request #1147 · open-telemetry/opentelemetry-rust (github.com)
- Stress test for metric modified to use random attributes by cijothomas · Pull Request #1126 · open-telemetry/opentelemetry-rust (github.com)

2023/07/11 (Tuesday 4:00 PM UTC)

Attendees:

- Lalit Bhasin (Microsoft)
- Zhongyang Wu
- Leighton Chen (Microsoft)
- Shaun Cox (Microsoft)

Agenda/minutes:

- Get your free slurpee's in US ;)
- Do exporters have to have clone/copy trait? What are the implications for not being having those?
    - We don't need those for now
- Internal logging proposal PTAL https://github.com/open-telemetry/opentelemetry-rust/issues/1146
- Design proposal to export logs only if leve/target is enabled - Design Proposal to export logs only if the level/logger/target is enabled. by lalitb · Pull Request #1147 · open-telemetry/opentelemetry-rust (github.com)

2023/07/04 (Tuesday 4:00 PM UTC)

Canceled because of holiday in NA

2023/06/27 (Tuesday 4:00 PM UTC)

Attendees:

- Leighton Chen (Microsoft)
- Aaron Clawson (Lightstep)
- Shaun Cox (Microsoft)
- Lalit Kumar Bhasin (Microsoft)

- Zhongyang Wu

Agenda/minutes:

- PeriodicReader.shutdown() does not propagate to exporter.shutdown() · Issue #1118 · open-telemetry/opentelemetry-rust (github.com)
    - Sounds like a bug
    - We don't have a good mechanism to handle timeout
- How do we handle errors/warnings?
    - How to handle the case where we just want to log a warning and proceed with logic? Is println sufficient?
    - There needs to be something similar to how the error handling is done
    - Not appropriate to use the logging api
    - Not a good idea to use println directly
    - Possibly use the error handling mechanism with WARNING level
- Is shutdown() called on application end?
    - User responsibility to call shutdown.
    - Not an automagic feature.
- Otel user_events Metrics exporter. by lzchen · Pull Request #1117 · open-telemetry/opentelemetry-rust (github.com)
- Canceled the July 4 meeting due to US holiday.

2023/06/20 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Aaron Clawson (Lightstep)
- Zhongyang Wu
- Lalit Kumar Bhasin (Microsoft)
- Shaun Cox (Microsoft)

Agenda/minutes:

- Harold - Releasing
    - Can we have a Cadence?
    - Setup a rolling cadence at least until 1.0
    - Testing and tagging for breaking changes after 1.0
- PeriodicReader.shutdown() does not propagate to exporter.shutdown() · Issue #1118 · open-telemetry/opentelemetry-rust (github.com)

- How do we handle errors/warnings?
    - How to handle the case where we just want to log a warning and proceed with logic? Is println sufficient?
- Is shutdown() called on application end?
    - User responsibility to call shutdown.
    - Not an automagic feature.
- user_events Exporter PR : https://github.com/open-telemetry/opentelemetry-rust/pull/1110
    - A lot of follow up PRs expected (span, etw, metrics, etc.)

2023/06/13(Tuesday 4:00 PM UTC)

Attendees:

- Aaron Clawson (Lightstep)
- Lalit Bhasin (Microsoft)
- Leighton Chen (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- Metrics api performance improvements
    - Remove de-dup: Remove de-duping of attribute keys · Issue #1098 · open-telemetry/opentelemetry-rust (github.com)
        - Setting an attribute with the same key as an existing attribute SHOULD overwrite the existing attribute's value.
        - opentelemetry-specification/specification/common/README.md at main · open-telemetry/opentelemetry-specification · GitHub
    - Why do we reverse the order of the array?
        - opentelemetry-rust/set.rs at main · open-telemetry/opentelemetry-rust · GitHub
        - [Aaron] To have last added attribute win.
    - Question: Do we need to sort or de-dup the attributes
    - Go: Recommend users to create AttributeSet beforehand and then caching it
        - Can we calculate the hash/fingerprint BEFORE we de-dup and sort?
        - C++, uses smart hashing, irrespective of the order of the attributes, they get the same hash
    - Can we use the assumed length of the attributeset to do some perf improvements? (e.g. 2000 max time series)

2023/06/06 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Aaron Clawson (Lightstep)
- Lalit (Microsoft)

Agenda/minutes:

- Logging recommendations.Can we make a recommendation to use tracing crate + the appender/bridge shipped from xyz (TBD)?
    - Do we intend to make a new end user callable Logging API?
        - No. There is already good enough.
        - {Aaron} Link to the otel.io registry. Tracing is the currently most popular, but that may not always be the case always.
        - We can mention the plan to NOT make a new one but not necessarily recommend any particular one.
    - [Cijo]- to send a PR making a note.
    - Cijo - to open an issue requesting new release, Zhongyang can do the actual release.
        - Once this is done, tracing-opentelemetry need to pick it up, before we can host the tracing-appender in that repo.
    - Remove Ctx from Metric
        - Blocked on examples being broken. There is a active PR adjusting examples, Zhongyang to help fix the broken examples, then the Remove Ctx from Metric PR can progress.

2023/05/30 (Tuesday 4:00 PM UTC)

Attendees:

- Leighton Chen
- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Aaron Clawson (Lightstep)
- Lalit (Microsoft)

Agenda/minutes:

1. Discuss https://github.com/open-telemetry/opentelemetry-rust/pull/1085
2. Improve stress tests by lzchen · Pull Request #1082 · open-telemetry/opentelemetry-rust (github.com)

2023/05/23 (Tuesday 4:00 PM UTC)

Attendees:

- Aaron Clawson (Lightstep)
- Leighton Chen
- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Lalit Kumar Bhasin (Microsoft)

Agenda/minutes:

- Implement cardinality limits for metric streams by lzchen · Pull Request #1066 · open-telemetry/open telemetry-rust (github.com)
- https://github.com/open-telemetry/opentelemetry-rust/pull/1071
- https://github.com/open-telemetry/opentelemetry-rust/issues/1070
    - Log crate appender is in PR now.
    - Next is appender for tracing crate
    - There is already https://github.com/tokio-rs/tracing-opentelemetry having the similar for Traces and Metrics...Should we continue/build there ? leveraging the same tracing subscriber
    - VS Create a separate tracing subscriber specifically for Logs (events)..
        - Won't have the ability to put Context.. Could be solved with right examples/docs.
    - User experience:
        - Instrument using tokio/tracing (SPAN, etc.)
        - Setup OpenTelemetry TracerProvider + https://github.com/tokio-rs/tracing-opentelemetry
        - Instrument using tokio/tracing for logs (INFO,ERROR etc.)
            - Dual event issue i.e tracing-opentelemetry converts this as SpanEvents + OpenTelemetry-Appender-Tracing will convert this to Logs.
        - Setup OpenTelemetry LoggerProvider + OpenTelemetry-Appender-Tracing
    - Proceed with this for now, and re-asses this once we have it working E2E.

2023/05/16 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas
- Lalit Bhasin

Agenda/minutes:

- None, meeting was over in ~5 mins.
- Cijo/Lalit will ask maintainers to merge the Logging PR, so we can iterate on top of it.

2023/05/09 (Tuesday 4:00 PM UTC)

Attendees:

- Aaron Clawson (Lightstep)
- Gary White (Verizon)
- Harold Dost
- Zhongyang Wu
- Lalit Bhasin (Microsoft)

Agenda/minutes:

- Spec Matrix PR for metrics:
    - Please review: Update spec matrix for OT Rust metrics by lzchen · Pull Request #3475 · open-telemetry/opentelemetry-specification (github.com)
        - Zhongyang Wu will take a look
        - Harold Dost started looking before meeting 👍
    - Left out functionality related to concurrency (need help on this)
- [Cijo] - https://github.com/open-telemetry/opentelemetry-rust/pull/788/files
    - Zhongyang can help fix Lint and unblock PR for merge.
    - Harold can help review.
    - Cijo/Lalit will iterate on top of it (and add an appender for tracing crate).
- Perhaps we can make lint a “required” CI for PRs?
- [Lalit] - In-process tracing between C++/Rust:\
    - FFI interface to call C++ libraries from Rust.
    - Context propagation from Rust to C++

2023/05/02 (Tuesday 4:00 PM UTC)

Attendees:

- Harold Dost
- Cijo Thomas (Microsoft)
- Lalit Kumar (Microsoft)
- David Barsky (works on tokio-rs/tracing, me@davidbarsky.com)
- Zhongyang Wu
- Leighton Chen (Microsoft)

Agenda/minutes:

- Stress Testing?
    - Harold: I'm wondering if we're testing the stress testing framework currently?
    - Cijo: For others we started with latency, but transited to the polling method
    - Leighton: For now we can just focus on the number purely relatively, and use this as a baseline.
- Version mapping?
    - Zhongyang: We have multiple crates and some users are a bit confused due to implementation issues.
    - Leighton: For pre-stable used specific pinned in Python
    - Example:
    - OTLP Exporter Create v 0.19.0
        - Depends on >= OTEL_sdk v 0.19.0

      Otel_sdk v 0.20.0 (breaking changed compared to 0.19.0) shipped

      Start : App -> OTLP Create v 0.19.0, OTEL_sdk v 0.19.0
      After bump: App -> OTLP Create v 0.19.0, OTEL_sdk v 0.20.0
      Users should bump OTLP Crate as well, else it can break

- Tracing API
    - Cijo: Should we have our own public API?
        - In C# the big draw was that there's a strong backwards compatibility
            - Large similarity between the two and Activity was already there.
            - Was already existing
            - Was part of the .NET Runtime
    - We can continue the conversation in https://github.com/open-telemetry/opentelemetry-rust/discussions/1032

2023/04/25 (Tuesday 4:00 PM UTC)

Attendees:

- Leighton Chen (Microsoft)
- Cijo Thomas (Microsoft)
- Zhongyang Wu
- Harold Dost
- Lalit (Microsoft)

Agenda/minutes:

- Harold: Updates from last meeting
    - Doc and Meeting are updated.
- Leighton: Are there any popular stress testing tools to measure throughput of a function?
    - Cargo bench possibly a solution
    - If not, custom solution can be created
    - Context: https://docs.kernel.org/trace/user_events.html and https://learn.microsoft.com/en-us/windows-hardware/drivers/devtest/event-tracing-for-windows--etw- exporters to come, and they need to support high throughput
- Cijo: https://github.com/open-telemetry/opentelemetry-rust/pull/788/files has been updated by the original author, kindly requesting a review.
    - How to mark one signal as “experimental”, while others can be stable.
        - Possible with “features”.
- Lalit: OTLP Metrics Unit tests not there. There are only smoke tests for OTLP traces:
    - https://github.com/open-telemetry/opentelemetry-rust/pull/1020: To add unit-tests for OTLP HTTP metrics exporter.
    - Doc Tests don’t count for coverage.

2023/04/18 (Tuesday 4:00 PM UTC)

Attendees:

- Cijo Thomas (Microsoft)
- Leighton Chen (Microsoft)
- Zhongyang Wu

Agenda/minutes:

- Harold: Updates from last meeting
    - I reached out to doc owner as Trask has asked us to move ownership, and the notes are now in the calendar

        - I have reached out to doc owner.
    - PR to update details https://github.com/open-telemetry/community/pull/1418
- Leighton: Metrics questions
    - New metrics SDK by jtescher · Pull Request #1000 · open-telemetry/opentelemetry-rust (github.com)
    - Need for basic metrics example
        - Difficulty creating simple metrics export workflow
            - Which reader/exporter combo to use?
        - Will review new stdout PR
        - Create opentelemetry-stdout by jtescher · Pull Request #1027 · open-telemetry/opentelemetry-rust (github.com)
    - In_memory_exporter `clone_data()` error
        - Leighton create an issue
    - Benchmarking - cargo bench for micro
        - Begin with benchmark for Gauge type metrics
        - Leighton to post some current benchmark results with system information
- Leighton: Additional changes to merged PR
    - feat: use `Cow<'static, str>` instead of `&'static str` by tqwewe · Pull Request #1018 · open-telemetry/opentelemetry-rust (github.com)
    - Leighton to create a new PR to address outstanding comments
- (Cijo): An overview of OTel Exporters to OS native tracing mechanism - ETW, user_events.
    - Microsoft will be contributing such exporters to OTel Rust.
    - Already exists for .NET, C++.
        - It was agreed to accept these Exporters as separate crates into the main repo now, and will be moved out to -contrib repo when they are made.
- Zhongyang: start the tracing API gap discussion in https://github.com/open-telemetry/opentelemetry-rust/discussions/1032
- ...

2023/04/11 (Tuesday 4:00 PM UTC)

Attendees:

- Harold Dost
- Cijo Thomas
- Zhongyang Wu
- Aaron Clawson
- Leighton Chen

Agenda/minutes:

- (Cijo) OTel Tracing API vs Tokio-Tracing API.

        - How does the user pick one over the other?
            - Julien has the most context on this
            - Tracing “tokio” definitely has the most usage.
            - tokio tracing is not a stable API
            - In C# using the built in API “Activity” for tracing.
            - Can/ should we approach tokio?
            - Tokio tracing using markos
        - We should do a gap analysis
- (Cijo) Logging API
    - There seems to be an abandoned PR: https://github.com/open-telemetry/opentelemetry-rust/pull/788
    - Rust log facade
        - https://docs.rs/log/latest/log/
    - We’ve posed on the PR
    - Not a priority currently, but people can feel free to work on it.
- Harold Dost : I’m not sure I’ll be able to make this one
    - I have updated https://github.com/open-telemetry/opentelemetry-rust/issues/977
- Jaeger Exporter:
    - We mentioned deprecating this
    - There’s concern about users not having a good upgrade path.
    - Cijo: Will put some examples in a PR which can help provide it.
- Meeting time
    - **Action:** Harold: I’ll put PR out to update in community repo.

2023/04/07

Attendees:

- Harold Dost
- Cijo Thomas
- Lalit Bhasin
- Zhongyang Wu
- Leighton Chen

Agenda/minutes:

- Intros
- Harold Dost Set regular meeting times

    - Current time slots https://calendar.google.com/calendar/u/0/embed?src=google.com_b79e3e90j7bbsa2n2p5an5lf60@group.calendar.google.com
        - Shown as alternating each week at 2PM UTC and 11PM UTC
    - **4PM-4:30PM UTC Tuesday from now**
- Harold Dost Clean up users groups
    - https://github.com/open-telemetry/opentelemetry-rust/pull/973/files We've merged this, but the user groups in Github still need to be cleaned up.
        - May need to follow up with Julian
- Zhongyang Wu Contrib repo
    - https://github.com/open-telemetry/opentelemetry-rust/issues/841
    - https://github.com/open-telemetry/opentelemetry-rust/issues/1014
    - Leighton: There be dragons in separate contrib repo
        - Python there's a cross build
        - The contrib repo policy is result
    - Cijo: Post 1.0 it shouldn't be as much of a problem.
        - In Dotnet all things which (except popular ones) were moved out into contrib repo.
    - We can accept outside "contrib" crates into the main opentelemetry-rust repo for now.
    - We may
- Harold Dost Path to GA Tracing
    - Milestone https://github.com/open-telemetry/opentelemetry-rust/milestone/1
    - Notes from previous meeting seem still relevant
        - We should likely start to triage the issues related to this. I did an initial pass, but there are some I'm not so sure. And potentially we should have some more tickets to at least follow up on:
            - Rust API Checklist
            - Validate Compliance
            - Maybe some other tasks beyond what's there.
    - Zhongyang Wu may need more integration test coverage
    - May want to run Rust API checklist https://rust-lang.github.io/api-guidelines/checklist.html
    - Leighton: Mentioned that it's possible to have a TC member review the Tracing Version
        - Also validate based on the compliance Matrix
            - opentelemetry-specification/spec-compliance-matrix.md at main · open-telemetry/opentelemetry-specification (github.com)
    - https://github.com/open-telemetry/opentelemetry-rust/issues/977 (Stability)
    - For what is defined as stable for tracing:
        - API,SDK, and Jaeger + OTLP Exporters would be best for our users.
            - Jaeger is the most used exporter.

            - Zipkin can be stable after GA for tracing.
            -
    - Action:
        - Review the Matrix
        - Triage Stable list
        - **Contact Communit team: 4PM-4:30PM UTC Tuesday from now**
        -

2021/11/23

Attendees:

-

Agenda/minutes:

- How to understand where we are towards GA?
    - Need more tests.
    - May want to run Rust API checklist https://rust-lang.github.io/api-guidelines/checklist.html
    - May also try “Dog food survey” from otel side https://docs.google.com/document/d/1rXT2eEAniVM7r8KiikzgJJ-r36shOg_twZP6GPzNvao/edit#
- Do we have a conclusion on https://github.com/open-telemetry/opentelemetry-rust/issues/364

2020/11/17

Attendees:

- Filip (@aheadmode)
- Julian Tescher
- Zhongyang Wu

Agenda/minutes:

- opentelemetry-otlp with hyperium/tonic (https://github.com/open-telemetry/opentelemetry-rust/pull/352)
    - Split out impls into feature flags
    - Remaining work to do:
        - Async optional
        - Expose configuration for tonic under flags

        - Add tests and benchmarks
    - Can be merged once the async runtime questions have been answered
    - Slides ttps://docs.google.com/presentation/d/1NQJcPxzZllb0GcPVXQS8V-q2XTCh9uzr6lf4KsUqqJk
- How should errors be handled in tracing?
    - Trace sdk should handle error cases by either returning a result where possible, or delegating to the global error handler
    - Some form of `Other` error case should exist that is open for extension by crates like exporters

2020/10/20

Attendees:

- Julian Tescher
- Zhongyang Wu
- Jan Kuehle

Agenda/minutes:

- Consider best implementation of https://github.com/open-telemetry/opentelemetry-specification/pull/1103 should we move propagators back into api module to support this?
    - Julian will open an issue in the spec repo to clarify.
- Go moved their API exports under a top level module, does that make sense for rust as well?
    - Remove api prefix for usability
- Should we have a Minimum Supported Rust Version (MSRV)? If so, how far back?
    - Julian will open an issue for 1.42 support
- Jan will open an issue for metrics docs
- Global tracer generation (may?) be slow? Should have more benchmarks to see if we need to change something here.

2020/10/06

Attendees:

- Julian Tescher
- Andreas Wiede
- Zhongyang Wu

Agenda/minutes:

- Next release
    - Can we 1.0 at GA?
    - We will keep releasing breaking changes until 1.0
    - Remove unstable `metrics` feature
    - Julian will prepare a release today
- 3rd party propagators
    - Release what is there currently, update if there are changes.
- Test coverage
    - GRPC is difficult to cover
    - Some initial coverage would be good to get a baseline
    - Are there ways of tracking perf test runs?
    - Common test utils (mock span processor, etc)
    - Makefile or script to run _all_ cargo tasks (fmt, clippy, test, etc)
- GRPC library decision (tonic vs grpcio)
    - Tonic likely only supports Tokio executor (should check)
    - Tonic's export structure leads to a messy mod structure
    - Important to still support other executors, even if it complicates the code
- Remove release schedule from README
- Are the top level re-exports a good idea?
    - Should api be top level module? Likely no as it can be confusing
    - Should trace module be flattened into the sdk and api modules (as it is currently) Don't flatten.
    - This should be done before release
- Add Andreas to otel

2020/03/10

Attendees:

- Mike Goldsmith (LightStep)
- Julian Tescher (Out There Labs)

Agenda/minutes:

- Review PRs
    - https://github.com/open-telemetry/opentelemetry-rust/pull/72 - I think this can be merged, we can add an alternative at a future point if we want to
    - https://github.com/open-telemetry/opentelemetry-rust/pull/51 - no changes in a few weeks, added comment to ask for final piece of feedback to be looked at

- Mike created beta tracker: https://github.com/open-telemetry/opentelemetry-rust/issues/74

2020/03/03

Attendees:

- Julian Tescher (Out There Labs)

Agenda/minutes:

- Canceled: No others available

2020/02/25

Attendees: Mike Goldsmith (LightStep)

Agenda/Minutes:

- Julian unable to attend today
    - Pushed PR for adding SpanBuilder
- Mike to create to issue to track beta readiness using template from other languages
    - Rust has not been committed to for March, but it's good to have visibility

2020/02/11

No meeting today

2020/02/04

Attendees:

- Julian Tescher (Out There Labs)
- Isobel Redelmeier (LightStep)
- Amim Knabben

Agenda/minutes:

- Review PRs
- Discuss 0.2.0 release
    - Add PR with changelog before release
- Review missing components vs spec
- Futures?

- FFI?

2020/01/28

Attendees:

- Julian Tescher (Out There Labs), Mike Goldsmith (LightStep)

Agenda/minutes:

- Review PRs
    - Futures looks good, final reviews
    - SpanData serializer look good, but concern around pinning to set Serde version - could look into making a feature
- Review missing components vs spec
    - How has OTEP 66 affected what the project looks like, probably fair amount of some work
    - Functionally work but requires work to separate prometheus implementation and introducing global refereces
- Milestone dates in README
    - Some discussion around some updating milestones (related #37)
- Possible release next alpha after pending reviews - release both OTel-rust and jeager exporter creates
    - Maintainers to be added to crates.io

2020/01/21

Attendees:

- Julian Tescher (Out There Labs)
- Amim Knabben (Loadsmart)

Agenda/minutes:

- PR review
    - Issue #50/#51 is ready to be merged, but ci is blocking. Nightly needs to be marked as optional as it is flakey.

2020/01/14

Attendees:

- Isobel Redelmeier (LightStep)

Agenda/minutes:

- Welcome back… again :)
- Issue review
- PR review
- Futures?
- FFI?

2020/01/07

Attendees:

- Isobel Redelmeier (LightStep)
- Alex Boten (LightStep)
- Amim Knabben

Agenda/minutes:

- Welcome back!
- Issue review
- PR review
- FFI

12/24/2019 & 12/31/2019

No meeting today ☃

12/17/2019

Attendees:

- Mike Goldsmith (LightStep)
- Julian Tescher (Out There Labs)
- Amim Knabben (Loadsmart)
- Isobel Redelmeier (LightStep)

Agenda/minutes:

- More reviewers?
    - Mike added to reviewers, can look to add others next year
- Meeting rotation + holidays
    - The calendar now has the rotation 🎉
    - No meeting next week!
- PR review
- Issue review

12/10/19

Attendees:

- Julian Tescher (Out There Labs)
- Isobel Redelmeier (LightStep)

Agenda/minutes:

- Administration
    - More reviewers?
        - Mike? → already an org member 🎉
        - How to fast track Sabree and Eliza as org members? (already SIG maintainers)
        - Kim? Alberto? Greg? (not yet org members)
    - Holidays - things might be slow
    - Meta
- Spec-level:
    - https://github.com/open-telemetry/oteps/issues/68
    - https://github.com/open-telemetry/oteps/pull/66
- Context propagation
    - Likely starting point: (tokio-) tracing
    - What would it entail for contexts to work with eg Futures + crossbeam?
    - Future combinator
- PR review
- Issue review

12/10/19 (7AM PT / 3PM GMT)

Attendees:

- Mike Goldsmith (LightStep)

Agenda/minutes:

- No other attendees, likely due to expected meet time of 3pm PST.

12/03/19

Attendees:

- Mike Goldsmith (LightStep)
- Julian Tescher (Out There Labs)
- Isobel Redelmeier (LightStep)
- Vineeth Pothulapati

Agenda/minutes:

- SIG approvers / maintainers
    - Isobel & Julian are now approvers, will look to add maintainers soon
- Issue review
    - Julian to create new issue to work on architecture of component communication to facilitate better abstraction (Jaeger & Promethus) #4 & #5
    - Query #6 to ask for further details & idea on what it's intended for
    - #13 & #14 someone to create implementation to get a better idea of what looks good in Rust (eg options don't work like in other languages)
    - #15 needs further review and agreement
    - #16 & #17 blocked due to #3 & #4
    - #21 merge into #15
    - #22 wait for metrics spec to stop being as volatile
    - #23 wait for exporter abstraction
    - #24 do we want to expose named meters globally using trait objects or local context - do other crates do it differently?
    - Create issue for benchmarks
- PR review
    - Both PRs ready to be merged
- Add issue for adding basic / more advanced async example to repo - related to #15/21
- Add issue to track meta info for repo - current state, targets & milestones

11/26/19

Attendees:

- Isobel Redelmeier (LightStep)

- Sabree Blackmon
- Alberto Leal (Sentry.io)
- Julian Tescher (Out There Labs)

Agenda/minutes:

- Intros
- How do we want to run the SIG (e.g., meeting cadence)?
    - Let's try rotating times to start to accommodate different time zones
    - Isobel will send a Doodle (or something) to the Gitter room
- What starting point should we use for the code?
    - Copy and push Julian's code into the official repo
    - Isobel will attempt to make sure we all have at least approver access
    - Isobel can ask about how to manage an official OpenTelemtry crates.io account
    - For potentially unstable APIs: let's treat things as unstable for now and look for feedback