# Agenda F2F Meeting - Barcelona May 16th and 17th

## Location:

Dynatrace Barcelona
Avinguda Diagonal, 211
08018 Barcelona, Spain

## Participants (please add dietary restrictions)

Alois Reitbauer (Dynatrace)
Christoph Neumüller (Dynatrace)
Daniel Khan (Dynatrace)
Hamidreza Jahtalabziabari (Elastic, Vegetarian)
Matthew Wear (New Relic, gluten, grain, and egg free)
Zeke Rosenberg (New Relic, Vegetarian)
Victor Soares (New Relic)
Isobel Redelmeier (LightStep, vegetarian)
Ted Young (LightStep)
Morgan McLean (Google)
Bogdan Drutu (Google)

## Remote Participants

Tyler Benson (Datadog)
Yuri Shkuro (Uber) - afternoons only
Sergey Kanzhelev (Microsoft)

# Zoom Meeting details:

https://dynatrace.zoom.us/j/395479951

Dinner Wednesday: http://www.baliusbar.com/ 7:00 p.m.

**We will also have dinner reservations taken of care of. Please specify next to your name whether you want to join (Thur and Fri)**

## Agenda

The main discussion points for this meeting will be:

- Finishing work on Trace-Context
- Getting started on Correlation-Context
- Getting started on Trace Interchange Format
- Discuss other protocol support (Binary, MQTT, ….)

**Agenda Thursday**

| | | |
|---|---|---|
| 9:00 - 9:30 | Welcome and walkthrough of agenda (½ h) | Alois |
| 9:30 - 10:30 | Trace-Context - Impl feedback and progress (1h) | Alois and team |
| 10:30 - 12:00 | Trace-Context - What do we still need to work on (1 ½ h) | Alois and team |
| 11:00 - 12:30 | Trace-Context - Non HTTP protocols ( 1 ½ h) | Sergey |
| 12:30 - 13:30 | Lunch (1 h) | |
| 13:30 - 14:30 | Trace-Context - Non HTTP protocols ( 1 ½ h) ctd. | Sergey |
| 14:30 - 16:00 | Correlation Context - Overview and goals (1 ½ h) | Sergey and Bogdan |
| 16:00 - 17:30 | Trace InterchangeFormat - Overview and goals (1 ½ h) | Alois and Victor |
| 17:30 | End of day 1 | |

**Agenda Friday**

| | | |
|---|---|---|
| 8:45 - 9:00 | Get together | Alois |
| 09:30 - 10:30 | Trace Context in the browser (½ h)<br>- Pre-flight request requires server configuration (Only works for same-origin request out of the box)<br>- Adding `traceparent`, `tracestate` to | Hamid |

| | | |
|---|---|---|
| | CORS-safelisted request-header list<br>- Include initial html as part of the trace | |
| 10:30 -11:00 | Official Announcement (½ h)<br><br>Blog posts from vendors announcing Trace-Context<br>● Morgan has already started on one that we can all use | Morgan and Alois |
| 11:00 - 11:30 | Moving forward with typed spans (½ h) | Morgan |
| 11:30 - 12:30 | Trace Response Header (2h)<br><br>Support for delegated sampling and tail-based sampling. Would we need a response header? Load balancer that use round robin to decide on whether they store the trace based on response feedback<br>○ Batch requests that may run very long<br>○ Crossing of trust boundaries or transaction scopes (i.e. my boundary might be different than my callers) | Team |
| 12:30 - 13:30 | Lunch (1 h) | Team |
| 12:00 - 14:00 | Trace Interchange Wire Format (1 ½ h) | |
| 14:00 - 15:00 | Recorded Flag - Open Issues (1 h)<br>● Should we rename it? (https://github.com/w3c/trace-context/issues/265)<br>● Should we remove it? (because it only works within one tracing system, shouldn't it be part of tracestate?) | Team |
| 15:00 - 16:00 | Extensibility for future version (1 h) | |
| 16:00 - 16:30 | Implementation guidelines (½ h)<br>● How to put trace context into logs<br>● How to work if you are using bigger IDs<br>● How to work if you are using smaller IDs<br>● How to identify other tracing providers already involved<br>● Working with security<br>● What to use trace state for<br>● How to pick a format - text or binary - and where to put it on a protocol<br>● Landing zone | Alois and team |

| | | |
|---|---|---|
| | • | |
| 16:30 - 17:00 | Framework vendor reach out (½ h ) | Alois and Morgan |
| 19:30 | Joint dinner (open end) | |

# Meeting notes

## Attendees

- Alois - Dynatrace
- Daniel - Dynatrace
- Zeke - New Relic, Engineer on distributed tracing team
- Victor - New Relic, Product Manager
- Isobel - Lightstep, tracing and opensource
- Hamid - Elastic, working on trace context for a year
- Christoph - Dynatrace, working on trace context. Correlation context isn't a big focus. Looking forward to interchange format.
- Matt Wear - New Relic, Ruby agent engineer
- Morgan - Google, PM for OpenCensus
- Sergey - Microsoft, works on TraceContext.

# Trace-Context - Implementation feedback and progress

GitHub: https://github.com/w3c/trace-context

**Morgan:** Integrated TC in OpenCensus for most languages. Implementations in the wild still using B3. Next step to implement in cloud services (cloud load balancers, app engine, …). Not the same size issues as we see with AWS S3. Conversations around Envoy and Istio. Broadcast switch as wide as possible.

**Hamid**: Implemented for browser but still using a different header name. Standards across agents. In the browser, it requires pre-flight request and server-side change.

**Christoph**: We have trace-context for most technologies. Going to launch EAP.  Currently opt-in with double tracing. Found a concern where, if Dynatrace and the framework both create/propagate the headers. Who wins out? Recording flag.

**Matt:** Have TC on the roadmap. Intention to implement later this year.

**Isobel**: Targeting open telemetry launch.

**Sergey**: ASP.NET Core 3.0 shipped later this year. Will propagate automatically. Any ASP.NET Core app will propagate automatically. How to add to a log format. Span ID is a bit short for IIS. IIS uses GUIDs for request IDs. Agreed that it is to be expected that request IDs like that will be superseded by TC and as such, there is no need to consider that in TC. How to control automatic propagation in case of a security boundary. There might be increased misuse of trace state. Recorded flag helps to communicate tracing intent a lot.
ASP.NET Core 3.0:

- ASP.NET Core – need unique ID for logs so generating new span ID all the time
- Efficient data structure for tracestate yet to be found
- Collaboration on tracestate
- In-logs representation of Trace Context
- Trace Context span ID is too short for IIS
- Black/white list of domains to propagate

IoT:

- Binary and AMQP protocols unavailability
- Long text?
- tracestate “abuse”. How to know if a property is crucial

Backends:

- Sampling on the backend – trust issues

# Trace-Context - What do we still need to work on

**Sergey** - Two things (1) Trace State is still hard to understand. People don’t understand it and keep asking questions about it. (2) Versioning, we decided we would add and not remove fields. We should make sure v0 platforms understand that and that they support the addition of fields to be future-proof.

- Christoph - If v1 comes in, adds new fields, maybe v0 continues propagation with only the v0 (downgrade) but the trace would still continue.
- Isobel - Why is the max version 255? Spec says that it needs to be 2 hex characters. Section 3.2.5

- We need to put downgrading behavior into the processing model. “If version is higher then use what you understand to construct new headers” - this means that a version 0 system would not propagate version 1 content but downgrade to 0 using what it understands. Alois: We should assume that a V1 would add additional headers on top and not change existing headers.
  Matthew: right now the spec seems to be ambiguous on that behavior, we have to clarify that. https://github.com/w3c/trace-context/issues/296 and https://github.com/w3c/trace-context/pull/295
- Daniel: Reduce phrases/naming to a clear terminology and then put them into the glossary and refer to it. E.g ‘vendor’, ‘propagating system’, ‘participant’, …
- Alois: We should create a registry of key names. (no objections by attendees) https://github.com/w3c/trace-context/issues/58. Same repo but different doc.
  OT should be possible to use for OpenTelemetry
  Alois will talk to plh to create the right structure for the registry.
- We need to do a grammar cleanup
  Isobel could pair with a tech writer. Alois: We should lock down the spec for that time.
  Isobel will confirm tomorrow.
- Alois: Response headers should go into a separate spec. Morgan: Don’t change the existing spec. Maybe integrate back at some point.
  Create a new repository and change the charter because you are only allowed to work on things specified in the charter (Alois)
  Alois will clarify with plh
- Daniel: Sampling/recorded flag wording/semantic cleanup

## Non-HTTP Protocols

Link to binary protocol: https://github.com/w3c/trace-context-binary/

Sergey:
We have multiple protocols defined already.
AMQP is important because of IoT devices. Feedback from folks is that text is too large and needed a binary specification for efficiency. Protobuf isn’t a great way to define the spec so we started defining a custom spec for it. People would like to see trace context as a single blob.

- Isobel - to what extend do we want to extend trace context to people?
- Sergey - it’s people involved in trace context. Yuri, other folks. Vendor-side people.

Been trying to find people from MQTT to get feedback.

- Daniel - Do all these protocols have a way to define metadata? Thinking about old days where vendors are changing the message to add trace state for protocols that don’t support metadata.

- Sergey - typically there is metadata available. Some previous protocols didn't have it but we recommended fields.
- Christoph - we should define 2 pieces: Protocol and Format serialization. From implementors perspective, it would be easier to define text parsing logic and binary parsing logic and define what format is used with what protocol.
- Sergey - protocol implementors are also concerned about format.
- Daniel - shouldn't this be transparent to the protocol owner. How do they use trace state?
- Sergey - AMQP has middleware.
- Alois - we should have a blog post about the protocols and formats, how they're used, etc. These kinds of guidelines would be helpful for people implementing. We have a lot of knowledge that can help people getting started.
- Sergey - at this point, we have a lot of "what didn't work".
- Morgan - there will be more protocols that we need to build formats for
- Sergey - yes, and binary is a very important format.
- Sergey - issues we found: (1) with binary, you always need to parse trace state. (2) all the limitations we have due to HTTP bleed into other protocols.
- Isobel - we should default to lowest common and use ASCII rather than unicode.
- Christoph - agreed. I think the spec calls out ASCII.
- Isobel - need to be able to translate from TEXT -> BINARY -> TEXT. All serializers should treat state value as opaque.
- Christoph - we lose some efficiency. ASCII is not compactable. But we need to go with ASCII always for compatibility.
- Isobel - what are the use cases of non-ASCII trace state? Base64 encoding limits to ascii.
- Sergey - limits our charset.
- Alois - is there a decision on ASCII for binary format?
- Sergey - the current proposal is ASCII, fixed lengths.
- Alois - content is the same, serialization is different.
- Isobel - Another way to think about this topic: Is it always a string or a binary array of key:value pairs. If it's always a string of key=value;key=value.
- Daniel - if those protocols support metadata, and the metadata has data type, why do we care how it's serialized?
- Alois - proposing we treat trace state as one big string. Optimizing for binary effort means a lot of work and changes to the HTTP spec. Is that fine, Sergey? We can always change this in v2.
- Sergey - yes. Whenever you propose ASCII for binary, people have questions. We should have documentation that explains the reasons why we chose ASCII. This also forces people to keep trace state small.
- What protocols should we target for definitions, examples?
    - HTTP, gRPC are most important for OpenCensus. Also some messaging.

- Basically, text and binary should handle most cases. We need to have more conversations for names, etc. Hopefully, this can happen via the discussions in the GitHub issues.
- Daniel - Do we need to specify each individual protocol, or do we need a generic way to suggest to protocol owners how to implement. Guidelines for where to put trace state in attributes/metadata/headers, guidelines for naming
- Morgan - some protocols are only supported by a single vendor. We can let them implement and document those implementations. Protocols with a community and multiple client/server implementations (multiple vendors) are the ones this group should define.

Tracestate encoding:
https://github.com/w3c/trace-context-protocols-registry/issues/4#issuecomment-461916686
https://github.com/w3c/trace-context-binary/issues/4
Is human readability important for tracestate?

## Correlation Context - Overview and goals

Yuri's blog post:
https://medium.com/jaegertracing/embracing-context-propagation-7100b9b6029a

Context propagation proposal:
https://docs.google.com/document/d/1UxrEYOaQlF_E4gtiPoFmcZ4YKKe1GxohvCvQDuwvD1I/

Current spec version:
https://github.com/w3c/correlation-context

Description of current usage:
https://github.com/w3c/correlation-context/blob/master/correlation_context/HTTP_HEADER_FORMAT.md
Current published version: https://w3c.github.io/correlation-context/

- Biggest question: What exactly should it be used for. Additional tags for metrics or business-critical information.
- Morgan: Not use for business-critical information
- Yuri: Disagree. People will need some form of context propagation. Certainly see the point of tracing vendors not wanting to be responsible for business context propagation. Brown University built a propagation mechanism… the idea is that tracing and other things that require context are built on context propagation.
- Yuri: Another concern is around privacy and data leaking.

- Alois: Isn't it the case that people can put private data in headers anyway? Shouldn't security teams be managing that via egress controllers?
- Victor: How common is it that customers have egress controllers?
- Alois: Maybe not even a privacy concern because the values in correlation context are encrypted.
- Alois: Should clearly document what it's supposed to be used for.
- Yuri: there was a blog about how to use Go context
    - Context.Value should inform, not control
- Is correlation context mutable?
- Do tracing vendors provide APIs for users to interact with the correlation context?
- What's the baggage equivalent in OpenTelemetry?
    - "Tags" in OpenCensus are "baggage" in OpenTracing
    - "Attribute" is added to a span.

## Trace Interchange Format - Overview and goals

**Morgan:** We should start working on it. What scenarios do we talk about

- Use Cases:
    - Import data from other participants of a tracer
    - Getting data from cloud services or middleware
- What type of data do we define for ingest. What are the required attributes for a specific type.
- PR For Typed Spans: https://github.com/open-telemetry/opentelemetry-java/pull/187
- Goal: Spec should be the source of truth
    - Suggestion (Morgan): start with OpenTracing spec
- Prior Work for Attributes: https://github.com/opentracing/specification/blob/master/semantic_conventions.md
- For the interchange format, let's not define the span types, just define the wire format for generic spans.
- Two scenarios to support: Streaming data, batch posting.
- Do we need spans or spans and traces.
- Should also think about read APIs as well.
- Christoph presenting previous work on Span Type.
- Github Project with previous work: https://github.com/dynatrace-innovationlab/TracingApiDatamodel
- Better to use canonical type to define the type of a span, so there is no need to infer span type from attributes.
-

## Implementation Support

- Move from Gitter to Slack due to better features and wider adoption
- Use medium publication for a blog

## Action Items

- Support for delegated sampling and tail-based sampling. Would we need a response header?
    - Load balancer that use round robin to decide on whether they store the trace based on response feedback
    - Batch requests that may run very long
    - Crossing of trust boundaries or transaction scopes (i.e. my boundary might be different than my callers)
- Blog posts from vendors announcing Trace-Context
    - Morgan has already started on one that we can all use
- Update processing model to reflect downgrade
- Extensibility and handling of future versions
- Implementation guidelines
    - How to put trace context into logs
    - How to work if you are using bigger IDs
    - How to identify other tracing providers already involved
    - Working with security
    - What to use trace state for
    - How to pick a format - text or binary - and where to put it on a protocol
    - Landing zone
- Reach out to framework vendors (envoy, Node.js?, …)
- Charter update
    - Response Header
    - Timelines
- What if a framework uses trace context and an APM solution does as well?
- (Isobel) At Kubecon: ask Josh Berkus for suggestions about Postgres tracing + protocol
- Clean up Correlation Context:
    - Guidelines for when to use it, and use cases that are antipatterns
- Move typed spans forward for OT
- (Hamid) create a proposal to whitelist traceparent + tracestate headers for CORS
  Alois will take the proposal and talk to plh.

- Hamid will create a proposal that outlines the problem - including pictures
- Morgan will connect to Dave (Stackdriver / OpenCensus Web)
- For the official announcement, Morgan will make sure that we will be featured on well known media
- Morgan: At a minimum we should post that on Medium.
- Morgan will take care of getting that into well known media.
- Individual vendors will link to it.
- Target end of June.
- Alois will also reach out to other vendors that were part of the process so far.
- Alois will reach out to analysts.
- (Isobel) propose an initial RFC process
- Daniel: Rename ‘recorded’ to ‘sampled’ and create concise wording around it
- Daniel - putting versioning / downgrading into processing model and review current wording for ambiguity.\
- Provide compatibility matrix

# Trace Context in the browser

**CORS / Header standard**
Hamid:
Browser needs to do pre-flight requests.
OOTB we can only add trace-parent to same origin
If we can get the header into this core safe list there would not be the need for a pre flight request
Morgan: We talked with w3c - we should propose more seriously in Japan
Christoph, Hamid: Let’s kick it off now.
Alois: The webperf group will have privacy objections.
Christoph: We would also be interested in that.
Morgan will talk to the Chrome folks
Isobel: Mozilla might be on board as well
Daniel: We should be better prepared when we talk to them next time
Morgan: Having an implementation will be helpful, that is important to that group

**Connecting browser/serverside via trace id, including initial request**
If HTML is generated on the backend, we need a way to continue it in the browser.

Isobel: SPA could make many requests, someone might keep a tab open for a week. Those could all be separate traces
Hamid: We could store information about the initial page load
Morgan: Open census web already does some of this with server timing, creating spans in the past

Response headers from the first request are not accessible for later requests
Hamid: Whiteboarding out browser vs. server creating trace context
Isobel: This may warrant a more dedicated, in depth discussion
Morgan: I think we solved that. He will cycle back on that.
Alois: We have solved this at Dynatrace by using cookies for this, but there are drawbacks
Alois: trace context on initial request won't happen because of privacy reasons
Christoph: Wondering if there would be real security concern
Alois: The biggest challenge when working with webperf is that they look at tracing from a different angle.
In this document we should tell them how an end-to-end tracing solution works.
Alois: Having the browser aware of headers won't be possible
Hamid: What if we create an api for JS to read this header specifically
Alois: Response headers might be a good way to communicate context.
Morgan: OpenCensus uses ServerTiming API to pass back the ID
**Sampled/Recorded flag:**
We also use the recorded flag to communicate sampling decisions - let's discuss that in the dedicated slot.

**Official Announcement**
https://docs.google.com/document/d/1FfkCTYJA5yN0dUedvtWqylg-mox-akgXpvvvmy2IjFk/edit

# Trace Interchange Format / Typed Spans

- PR https://github.com/open-telemetry/opentelemetry-specification/pull/14
- Sergey: Let's split the typed span doc into separate files, with an index
    - Alois: Start with single file, then maybe split later
- Suggestion: Start by merging something to make further discussion easier
    - Mark clearly that it's extra experimental!
- Open question: how do we want to version the span types?
    1. In lockstep with the rest of the spec?
    2. Separate from the rest of the spec?
- Sergey: We should work on documenting/clarifying the content of attributes
- Christoph: Examples + descriptions (i.e., in English, not code) will be very helpful
- Christoph: This should serve 2 purposes:
    - Instrumentation implementers know what attributes to set
    - Consumers (Backend) knows what to expect from certain types of spans (which attributes to expect).
- Morgan: **Decision: we will place span type definitions (with expected metadata for each type) into the OpenTelemetry specs repository. We will merge existing pages (for gRPC and SQL) into this page.**
    - **New Relic will review and add any missing types and information**

    - This document doesn't impact any decisions around adding strongly typed spans to the APIs
- Zeke: Errors are not a 'canonical type', because a typed span can also be an error. But we do need to define attributes for errors. Are there other 'interfaces' that a typed span can implement?
- Victor: What is the relationship between the naming of a span, and the name of related metrics? Are these correlated in open telemetry?
    - Sergey: This will vary by technology, for HTTP this is all defined in a single document. It makes sense to name by url or something.

## Trace Response Headers

- Scenarios
    - Client (Browser) sends a request (without tracing). Server could return response headers, so the client can work with the traceid/spanid. Client also needs to know sampling decision.
    - Load Balancer: Records all spans at first, actual sampling decision is made based on response from the service that the LB forwarded to. LB then samples a span based on that response.
    - Restart a trace and return new trace identification information to caller
    - Send Tenant ID/identity of the service so caller knows where to query telemetry from
- Rationale Doc: https://github.com/w3c/trace-context/blob/master/spec/21-http_header_format_rationale.md#use-cases
- Open questions from rationale doc
    - If standard defines response headers - are they required or optional?
        - Optional
        - We need to define scenarios where returning a response header is recommended
    - How are they propagated to the caller of the caller? Is this done via multiple hops?
    - Some IoT devices may not expect relatively large response headers.
- What about multiple requests, should all response be "back-propagated"?
    - Combined? Choose one?
- Matt: Depending on who the caller is, we may or may not want to back-propagate response headers
- "Traceparent" does not make sense for back-propagation. Do we need a different format?
    - "Tracechild"? This is needed for the browser use case
    - Also, what about custom payload? Yes.
- This is the inverse of tracestate
- Back-propagation should not happen over multiple hops.

- If a trace is restarted by downstream app, then the child can return the new traceId (and attributes) for the parent to record
- **Proposal: return an object similar to the normal TraceContext header**
    - Attributes: TraceId, ***TraceChild***, TraceState, Sampled bit
        - TraceChild references the active span ID for the system sending the response header
    - What should the name of the response headers be?
        - **TraceParent** is a weird name for something that is not a parent.
- Sergey: we should move the response work into a new repository
    - Alois will ask Philippe to do this
- For Browsers: Rather than having these response headers exposed directly, it would be better to be able to get them via special API's

## Trace Interchange format / Wire format

- Which dataformat (protobuf, json, …)
    - Isobel: maybe both
    - Also MessagePack as an idea: https://msgpack.org/index.html
    - Pro json:
        - Better readability
        - Covered by all techs
        - Isobel: string/int handling might be easier
    - Stackdriver: grpc(protobuf) + json
    - Azure: json (internally using binary)
- Json vs. protobuf does not need to be solved now. The spec is more important right now.
- Scenarios / Use Cases
    - Backend/collector that consumes spans from agents (or any "span-producer")
    - Backend that exports spans (that can be imported by another backend, or any other analytics tool)
- Basing off of existing OpenCensus proto format, what would need to be changed?
    - Remove "Stacktrace"?
    - Add concept of trace (pre-correlated spans)?
    - Trace = Collection/Tree of spans grouped by ID
- "Typed Spans"
    - There should be a "CanonicalType" field, but we would still want a generic key/value list of attributes
        - Namespaced attributes?

# Recorded Flag

Is anyone against removing it?

- Yes; Morgan, Sergey, elastic
- When is it useful?
    - Google and Microsoft already use it for internal systems for “forward-propagate” a sampling decision.
    - Google: For outside (non-trusted) requests, it’s also interesting to know if they were sampled, it may increase the chance of sampling a request in google cloud
    - They don’t wanna take the perf-hit for parsing tracestate
    - For “simple, head-based sampling systems” this flag is useful. For more complex or tail-based sampling systems it should be ignored.

Rename to “sampled”

- Daniel: would feel better
- Original PR when this change was made: https://github.com/w3c/trace-context/pull/142
- Discussion on renaming https://github.com/w3c/trace-context/issues/265
- Daniel: We should change the name. It is a doc change. Let’s also create a concise description on what it is

Action items

- Daniel will prepare a PR for changing “recorded” to “sampled”
- Daniel will prepare a PR for making the description of the flag better
- -> no real semantic changes
- Morgan: If renaming means reset of the CR process leave it
- Alois: We will ask plh

# Extensibility for future version

Conversation about forward and backwards compatibility.

Daniel: We might just add new headers on top
Hamid: But what is with new flags? We are also ruling out that we will never change the length of a header. Maybe we decide for GUID at some point.
Christoph: We decided that we will never change the existing parts but may prepend in the end. There might be more. With that this should be solved and it’s in the spec under ‘versioning of trace parent’.
Hamid: Can we guarantee that some flags will be free forever. I would be hesitant to use trace state.
Daniel: This would take away inter vendor traceability. I would recommend to use it.

We should add it to the processing model.

## Wording and Grammar

Isobel will pair with a tech writer around the 28th to do a final wording review.
Ahead of that there should be a set of minor fixes like the recorded/sampled change.
Also, we need to review for conciseness.

## Meeting wrap-up

- **Christoph** - happy to see the recorded flag move to sample. Wondering if we're going to make progress or how fast we move to make progress on the (1) Response header spec (2) interchange format.
- **Zeke** - Typed span will be very useful, especially from a backend perspective. Not sure where we ended with the correlation context.
- **Isobel** - Have a better sense of the correlation context than I did a few days ago
- **Matt** - Liked the conversation on typed spans. Planning on reviewing what New Relic does with attributes that we capture and compare. Interesting to see Hamid's use case for the browser.
- **Victor** - We were pretty quick on going through the agenda. Not so many controversial discussions. Neutral for Correlation-Context, and still not totally clear. +1 for Span Types.
- **Hamid** - +1 for taking things offline that could be taken offline, which made the discussion more efficient. Happy we talked about the Browser-tracing use-cases.
- **Daniel** - good progress
- **Alois** - We're blending OpenTelemetric stuff with W3C work, but that's ok. Some things will require face-2-face meetings with other groups. Takeaway for next time: Everyone fill in discussion items beforehand and be better prepared for topics.
- **Morgan** - Good to meet people, getting a lot of value out of conversations. TC Spec is pretty much done and looks good. +1 for doing a final pass on editorial stuff. Excited for things that come next.
- **Isobel** - Overall, really good experience. 2 Days are more focused than video-meeting every 2 weeks. +1 for moving to Slack.