W3C Distributed Tracing Working Group workshop (Nov 2019)

We will be hosting the next face-to-face W3C distributed tracing working group workshop in Seattle November 14th-15th.

We plan to concentrate on the following topics:

- Spec for response headers
- Correlation Context specification
- Advancement of trace context implementation on other protocols

Logistics

Dates: November 14th-15th, 9AM-5PM pacific time.
Location: Microsoft office at Lincoln square

Lobby is on the 15th floor of Lincoln square. Let's meet there. Please have an ID to pick up the guest badge.

There are visitor parking in the Garage or Bellevue Square mall parking just across the street.

First day: Conf Room LINCOLN SQUARE/26004 (20) Mccormick;;
Second day: Conf Room LINCOLN SQUARE/24006 (14) Aspen;

Dial-up using Microsoft Teams:

> Join Microsoft Teams Meeting
>
> +1 469-480-6275 United States, Dallas (Toll)
>
> (866) 63G9-0588 United States (Toll-free)
>
> Conference ID: 281 522 70#
>
> Local numbers | Reset PIN | Learn more about Teams | Meeting options

Participants

In-person:

- Sergey Kanzhelev, Microsoft

- Lionel Godolphin, Microsoft
- Bogdan Drutu, Google
- Victor Soares, New Relic
- Justin Foote, New Relic
- Zeke Rosenberg, New Relic
- Martin Kuba, New Relic
- Daniel Khan, Dynatrace
- Daniel Dyla, Dynatrace
- Morgan McLean, Google (1st day only)
- Christoph Neumüller, Dynatrace
- Chris Kleinknecht, Google
- Anssi Alaranta, AWS
- Matthew Wear, LightStep
- Alex Fedotyev, AppDynamics
- Eyhab Al-Masri, University of Washington (2nd day only)

Please mention your dietary restriction next to the name or send them via e-mail to Sergey.Kanzhelev_@_microsoft.com (remove underscores).

Remote:
- Yuri Shkuro, Uber Technologies
- Isobel Redelmeier, LightStep (possibly in person)
- Bob Strecansky, Mailchimp (possibly in person)
- Chris Erway, SolarWinds
- Michele Mancioppi, Instana
- Philippe Le Hegaret, W3C (sick)

Dietary: one request for vegetarian option, one paleo / keto, one vegan

Agenda

Thursday, November 14th

Time	**Topic**	**Presenter**
8:45-9	Breakfast Blazing Bagels and Bakery (delivered 8-9)	
9AM - 9:30AM	Intros, what people expect from the workshop	All
9:30 - 10AM	Current status of W3C Trace Context implementation	All

10:00AM - 10:30 AM	BREAK (OpenTelemetry public governance meeting)	None
10:30AM - 11:30AM	Response headers	Morgan McLean
11:30AM - 12:30PM	Correlation Context specification	Yuri for use cases Sergey for OTel
12:30PM - 2:00PM	Lunch. *We will have lunch in the room. And then can go downstairs for coffee. Blocking 1.5 hours as this is how much it typically takes to recover*	
2PM - 3PM	Migrating to W3C Trace Context: challenges and learnings	Group talk
3PM - 4PM	Business transactions monitoring challenges	Lionel Godolphin
4PM - 4:30PM	Built-in W3C in ASP.NET	Sergey Kanzhelev
4:30PM - 5PM	Wrapping the day: - Feedback on the first day - Review agenda to tomorrow	
6:30 PM	Dinner at Solarium Kitchen & Bar	

Friday, November 15th

We will need to wrap up a day by 4PM as many people has a 7PM flight.

Time	**Topic**	**Presenter**
8:45AM-9AM	Breakfast - will be delivered to the room. Panera Bread: Breakfast Sandwiches & Coffee (delivered 8:30-8:45)	
9AM - 10:00AM	Trace context implementation on other protocols: pure text, binary	Sergey
10:00AM - 11:00AM	Trace context implementation on other protocols: AMQP, MQTT	Sergey

11AM - 12PM	Trace Context in Browsers	Bernhard Lackner & Christoph Neumüller
12PM - 1:30PM	Lunch Order number is 231589 (delivered Between 11-12)	
1:30PM - 2:30PM	Response header: deep dive, actionable results	All
2:30 PM - 3PM	Working group priorities and what to do next. Touch on TraceData	all
3PM - 3:30PM	Free chat: how to improve the spec and make it easier to consume?	
3:30PM - 4:00PM	Wrapping up the workshop: feedback, impressions, learnings	All

Notes

Sergey, Microsoft - tracecontext current status and home everybody will start implementing. Also gather energy to the next steps

Morgan McLean - Google -

Daniel Dyla - Dynatrace - new to the group and here to contribute in any way i can

Daniel - Dynatrace - Happy we are wrapping it up and next protocols are important. Happy to see AWS in a room

Cristopher DT - browser tracing is important
Zeke, NR - tracing backends, interested in rolling out implementation + interested in browser

Chris, Google - Interested to get W3C into OpenTelemetry

Martin, NR - browser products

Justin, NR - other formats, non http

Lionel, MS - see how this all fit together into applied

Matthew, Lightstep -

Anssi, AWS - came to adopt it

Alex, AppDynamics - learn and observe, PM on .NET monitoring and catching up on this space

Bob Strecansky - Mailchimp, maintainer of opentelemetry-php, looking forward to learn from the working group the idioms we are going to use in OpenTelemetry.

Philippe - W3C rep, came to help. Big concern is ensure we are not infringing security and privacy

Michele, Instana - interested how cloud providers adopting it

Victor - arrived late. Interested in everything :D

Current status of W3C Trace Context implementation

Philippe:
https://w3c.github.io/trace-context/?specStatus=PR;crEnd=2019-09-09;prEnd=2020-01-10;publishDate=2019-11-21

https://github.com/w3c/transitions/issues/186

Candidate recommendation stage for http protocol. Stable enough that people can begin implementing. Next phase will be proposal implementation, no changes can be made at all.

Sergey: OpenTelemetry, several Microsoft / Azure services (azure functions, API gateway), ASP.Net

Philippe: Spec will be final mid January.

If we want to start a V2 specification - we can start publishing it whenever we want. Example may be introducing the response headers.

Morgan: Google: x-google-cloud-trace header will be replaced with traceparent, Work on introducing headers in Envoy and Istio. With small config change it is easy to switch to this header. Cloud services is a bigger priority.

Dynatrace, Christoph - OneAgent supports this header, demand wasn't huge, also waiting for cloud services to catch up. Customers needs to be educated.

Morgan: still a lot of confusion with customers. Explaining all the cases is hard.

Justin + Vic: NewRelic - two languages prepared to be released - starting to have conversation

with customers. Victor: needs to be careful with backwards compatibility support. About to release beta.

Christoph: either use old or use both. No option currently for using just trace context headers

Matthew: Lightstep is switching to OpenTelemetry. For the time being switching LS tracers to w3c. Lightstep is a primary, second is Legacy b3, not a single header B3.

Anssi - x-amzn header is being used. Blockers: want specification to be final. Implementation on streams and queues is a question, as there's a concern about large header sizes impacting performance on extremely small requests.

AppDynamics - not much to share. Customers are confused, especially with modern stacks. No committment yet.

Yuri: maintainer of Jaeger, tech lead of tracing @ Uber. There is a Java PR in Jaeger to support Trace Context, it's not finished… Sunsetting Jaeger client for OpenTelemetry and with the switch everything will just work.

Response headers

Today W3C spec defines headers in one direction.

Scenario:

Request gets into Google Load Balancer and then hits an application. Span from LB is interested, but not sure whether to collect it or not. Historically was implemented via response header from the app to load balancer. Same scenarios apply to web browsers and other clients where sampling decisions wants to be made on the server.

Do we define only span ID, sampled flag, or what fields we need?

Yuri: What google returns? Only sampling decisions?

Morgan: same trace header as was set in forward direction.

Yuri: why not create the trace id at LB?

Morgan: not sure

Morgan: maybe had limited influence on what that team did. Bogdan may have more context

Bob: Mailchimp - pass the context from content delivery network. Different load balancers will have different behavior.

Morgan: less trusted environment if something you don't control decides for you

Christoph: no agent may be loaded yet (browser scenario). Agent may collect trace ids from all responses on the page and correlate those with the page actions.

Justin: service gateway may have many “children”. How to resolve is later?

Anssi: capture IDs in the response from clients that don’t do tracing. Example: a storage client doesn’t have tracing support but it logs the responses. If a customer has an issue, they can use this ID when calling support. Another example - amazon lambda.

https://github.com/w3c/trace-context/blob/master/spec/21-http_header_format_rationale.md#response-headers

Use Cases

1. Restart a trace and return new trace identification information to caller.
2. Send Tenant ID/identity of the service so caller knows where to query telemetry from.
3. Notify upstream to sample trace sending a sampling flag (+ sampling score) for delegated sampling.
4. Report back data to the caller (like server timing, method name, application type and version). E.g. for HTTP call - caller only knows the url when server knows route information. Route information may be helpful to caller to group outgoing requests.
5. Clock skew / time sychronization?

What other data should be returned in the response?
Ansi: Anything that has user data/privacy concerns we should stay away from
Phillip: Anything that traces a user or identifies someone is problematic

Big concern from privacy - make sure it’s not automatic in browser.

Philippe: fetch specification from GDPR. Response headers access needs to be added here:

https://fetch.spec.whatwg.org/#cors-safelisted-response-header-name

Server timing out of scope - use special header

Christop: if we make it extensibile - we may use it. But it’s good to have. Ex: time synchronization.

What’s absolutely needs to be in it?

- **TraceID** - must

- SpanID - not needed

SpanID:
Do you need it if you have a TraceID

We didn't find scenarios where we need to know SpanID of a child. Whenever traceID different from yours will be returned - it means a new "root" span started.

- **Sampling decision** - can
- TraceState - what would be the scenarios? Didn't find any

- **Version** of the header format.

Server route (span name) can be returned via this header. Victor: route name, app id, etc. was used in NewRelic. Closer to correlation context, not the tracestate. Done so that upstream services can produce metrics with dimensions based on downstream callers.

Bogdan: this ^^^ is what we are doing in Google for server timing. LB time, server time, sampling decision.

Navigation timing spec: https://www.w3.org/TR/navigation-timing/
Navigation timing spec (Level 2): https://www.w3.org/TR/navigation-timing-2/

Matthew: Optional/required?
- Optional!
- Particularly useful for requests that cross trust boundaries and requests from clients without trace instrumentation / instrumentation loads late
- Should client libraries always look for it? Looks optional, but easy to enable by default.

Format?

AWS: (Anssi) Same header, but only return the ID and sampling.

> Morgan: how sampling may be exposing information?
> Anssi: requiring it may be bad

Google: same header in both directions. 8 bytes being added, but not required. Doesn't care about it.

Returning Tracestate also carries a lot of privacy concerns. Trust boundaries are hard to define, harder than for what you call.

```
traceparent: 00-0af7651916cd43dd8448eb211c80319c-b7ad6b7169203331-01
```

Yuri: name is confusing

> tracechild?

Minimal: Version-TraceID
With sampling: Version-TraceID-TraceFlags
Full version: Version-TraceID-SpanID-TraceFlags

Instana: server timing is used to carry TraceID back on HTML pages

Use other spec? https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Server-Timing

Philippe: server timing was never designed for it. It's in limbo as far as I can tell and I'll need to check its progress. Action item on Philippe to talk to the group and get information from them. Can we have special name/value pair for TraceID?

```
// Single metric without value

Server-Timing: missedCache

// Single metric with value

Server-Timing: cpu;dur=2.4

// Single metric with description and value

Server-Timing: cache;desc="Cache Read";dur=23.2

// Two metrics with value

Server-Timing: db;dur=53, app;dur=47.2

// Server-Timing as trailer

Trailer: Server-Timing

--- response body ---

Server-Timing: total;dur=123.4
```

Anssi: Seems unrelated, but if we can make it - it may be OK.

Instana implementation:

1. Browser loads HTML + server timing data, loads Instana script asynchronously
2. Instana script looks for server timing data, which already contains the trace ID from the server, loads instrumentation

3. Instana script generates a beacon, which acts as a span ID for the trace without any parent child relationships

Should TraceID be optional?
Bob: limit the size of the header.

Fixed size without optional headers may be faster for serializing and deserializing. Consistency may be more important.

Morgan: with the same format.

Tracechild - may be confusing
TraceResponse
Traceback
ServerTrace
TraceServer

Who is ready to implement?

OpenTelemetry JS (browser): Dave Raffsenberger @ Google, Hamid @ Elastic, Mayur Kale @ Google, Bartlomiej Obecny @ Lightstep, Daniel Dyla @ Dynatrace, Martin Kuba @ New Relic
For LBs: Bogdan.

Spec & process?

Action items:
- talk to Web Perf group/Server Timing folks
  https://github.com/w3c/server-timing/issues/65

Separate spec or new spec?

Philippe: it's better to start by iterating on a current sp Might need to talk Yoav Weiss (Goo) to get more info on browsers

Correlation context

References

- Scenarios
  https://medium.com/jaegertracing/embracing-context-propagation-7100b9b6029a
- Platforms support for correlation propagation:

https://github.com/open-telemetry/oteps/pull/42
- https://devblogs.microsoft.com/aspnet/improvements-in-net-core-3-0-for-troubleshooting-and-monitoring-distributed-apps/
- Status of a spec: https://github.com/w3c/correlation-context
- Presentation from Yuri… Link

Discussion

Bogdan: security slide. Without auth between Service 1 and Service 2 - how do you trust the token from the baggage.
Yuri: there is Service to Service auth. What is shown is external users authentication service. Services trust already established.
Bogdan: doubt it will work with the baggages, as services may impersonate people.

Other scenarios are quite useful.

Yuri: yes, there should be a trust system. Perhaps you can re-sign the token on every hop to implement security in-depth.

Daniel: is it really a part of a discussion?

Bogdan: I'm only arguing with the specific scenario, not all other scenarios.

Lionel: we creating interoperability

Anssi: scoping of scenarios for this system is important. So every actor will operate on it.

Lionel: this is why there should be some details on security and privacy and how to do it (like what encryption to use.

Anssi: without security it's very very dangerous.

Sergey: one header is better.

Anssi: this exactly what needs to be described.

Bogdan: we know we need it and missing. Let's not come with the one-of solution and create this interoperability. Do we really need it generically?

Yuri: one example where you need it for interoperability - if LB (or other SaaS) as a service is provided - request from Uber back to Uber should preserve the header.

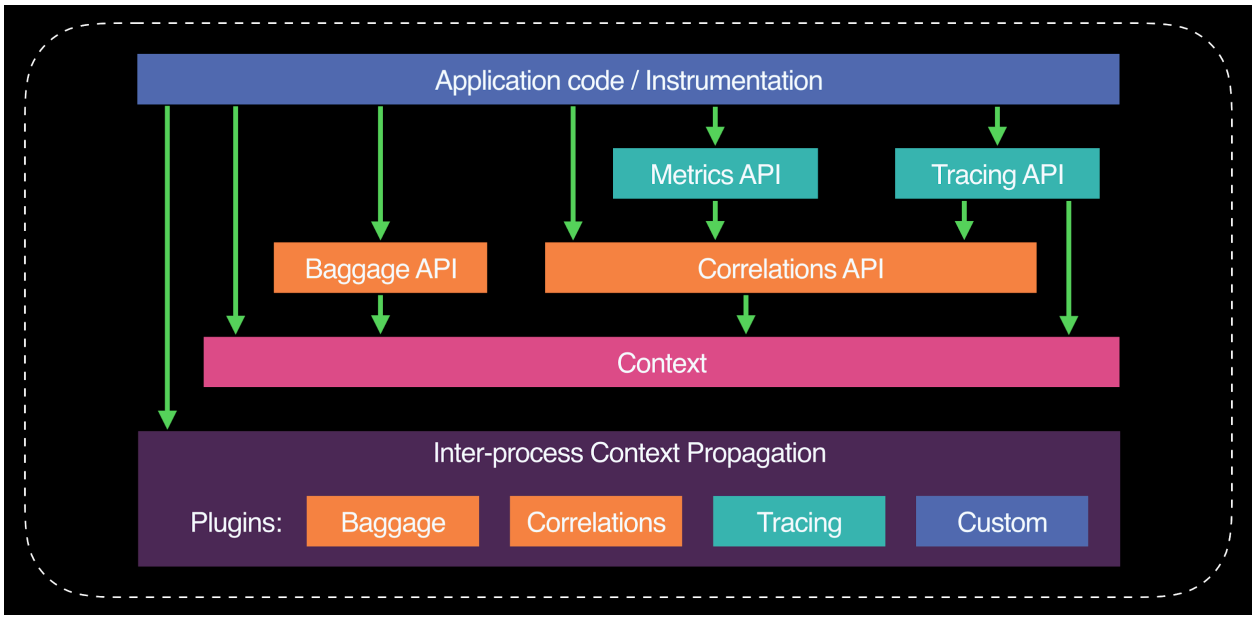
Bogdan: just having a header name may be enough.

Christopher: propagation platform is app concept.

Bogdan: definitely very important for observability. Interesting part - some of these properties are quite important for observability. Some - general purpose… How do we decide which property has what value. Separation of trace needed and other properties is very important.

Bob: as all of our companies start using SOA with interoperability over the wire (http, GRPC, et. al.) Maybe we could recognize this standard is still in flux, and attempt to define a solid standard that could spin off into a separate working group if needed. We could also lead by example here in our spec.

Context in OpenTelemetry



Platform propagation

Ideally context must be an underlying layer.

Slide from Yuri's presentation…

- Prioritization of baggage entries may be in "Context" API.
- TTL for baggage entries (only my application, only "next level").

Correlations API - write only.

Platform support provides unified rules on how to propagate the context.

Status of a spec

https://github.com/w3c/correlation-context

Bogdan, Yuri, Daniel, AlexFedotyev, Matthew, Justin

Migrating to W3C Trace Context: challenges and learnings

Morgan: google - challenges with transitioning:

x-google-trace-id and start migrating customers to OpenCensus and OpenTelemetry.

Some customers re-configure SDK to use older trace as they want to propagate trace-id with Google services. Challenge: either Envoy or Google services as envoy only supports traceparent.

Snap is the customer who went thru transition.

Anssi: problem when two headers flying and intermediate only modifies one of them. AWS: problem - migration will take years. No good motivation to transition.

Who solved the problem of two headers and only one was modified?

Morgan: traceparent takes precedence all the time.

Anssi: A->B->C will look like A->B, A->C.

Example: nginx with xray plugin - will propagate both, but only modify one.

Morgan: You still get the complete trace back. But can make wrong conclusions.

Anssi: root cause analysis analytics, human time spent on investigating the incorrect details. If linkage is shown incorrectly. Undermines trust.

Anssi: It's not a problem if intermediate service only uses one and drop another.

Anssi: can we collaborate on how the conflict resolution works?

Morgan: Timestamp on when was updated... But it's very unreliable solution

Dynatrace: Christoph. X-Dynatrace experience:

Propagate trace-parent and also propagate proprietary header. Also put legacy header inside tracestate so it can be recovered after w3c-only system. Searchability is solved using parent/traceId. Proprietary parent IDs are used everywhere for correlation. So basically sending both headers as a transition stage.

Same problem as if trace-parent would have been used as a primary one.

Morgan: there is already cases of multiple headers floating around if you need cooperation with SaaS apps.

Can it be mitigated by response header?

NewRelic: nobody needed to interop with Dynatrace or DataDog. But large enterprises have some scenarios around Jaeger used in some apps. Or some cloud-provided service in the middle.

Envoy is a trickiest example as it's proxy everywhere. Can be participating in w3c trace. And blindly forward another header.

Bogdan: inside the company migration may be easier.

Michele:

[2:31 PM] Michele Mancioppi
It's not only services in the middle, it is also "leaf" services managed by the Cloud provider

[2:31 PM] Michele Mancioppi
For example, if we can get the PoV on a call from the IaaS provider

[2:32 PM] Michele Mancioppi
And being able to query the tracing APIs of the IaaS provider and "stitch" that info to the "proprietary" trace of the APM vendor is actually a very nice thing

PLUG on response headers:

- Another scenario: is number of rows processed.

NewRelic transition from old NR header to new NR header. Had a good motivation as enabled full traces. Broke some things along the way, but got new features. **There were no interop at all.** So customers wanted that and transitioned. Switch to tarceparent has lesser carrot

Michele: in APM world lambda and xray world - inserting proprietary tracers inside the lambda. But there is no way to insert custom code in other pieces of infra. So getting this info from xRay and stitching it on backend.

Provide - understand everything, propagate w3c. Or use old format everywhere. Waiting for customers to “complain”.

Unintended correlation may spook some customers. But they realize the value.

What will help the most?

- Motivation of transition blog post/whitepaper?
- What experiences are broken explainer?
- Transition ordering? Migration plan.

Morgan: migration plan will be the most valuable. Motivation can be a simple blog post.

Also have an “warning” of what can go wrong and have an explanation why you still should trust tracing.

Bogdan: motivation includes being able to correlate customer telemetry and “support” of a cloud provider.

Daniel: IoT team also wanted end-to-end tracer to understand whether it’s customer app to blame or IoT Hub itself.

Victor: our joint customer don’t even see it as a possibility. Educating customer may be a good thing. Break thru “sounds impossible” wall with the customers.

Daniel: simple correlation between headers may be enough. But even this a “future” thing.

<more discussions on motivation to migrate>

Action item:

- Migration plan whitepaper
  Sergey, Morgan (only if Sergey is doing it), Sonja (Dynatrace, Daniel to confirm with her),

- Motivation/benefits to migrate blog post
  done.

- success stories
  - Victor, Lionel,

Business Process Monitoring challenges

Presentation from Lionel - to be shared.

Bogdan: Context needs to carry the deadline. Also there must be external system that keeps the current state.

Interoperability and stitching together elements is important.

Christoph: BPM is on top of APM.

Keeping context across the entire system may not be possible. So phone home system is important. But phone home system still requires some identifiers.

Sergey:

- Variability of context propagation technologies
- BPM requires second set of IDs that needs to be carried alongside the APM ids.

This ^^^ may give us a lot of thoughts on how propagation can be improved.

Bogdan: trace context is very much for transactions. Matches the needs to BPM and it should use the same model, but a different name?

[3:57 PM] Michele Mancioppi
BTW SAP is not necessarily a black box

[3:57 PM] Michele Mancioppi
it has a baggage,based tracing system called SAP Passport activated on demand

[3:58 PM] Michele Mancioppi
it is just really hard to lay hands on the spec :P

[3:58 PM] Michele Mancioppi
(Lionel used SAP as a back box example, just adding some info that may come in handy)

[4:02 PM] Michele Mancioppi
(Also, not to be confused with the SAP Passport that is the certificate users can use to do SSO)

Wrapping up and next steps:

- Justin: some intersection with correlation context

Built-in W3C in ASP.NET

https://devblogs.microsoft.com/aspnet/improvements-in-net-core-3-0-for-troubleshooting-and-monitoring-distributed-apps/

Christopher: How to convince to switch to other APIs rather than build new?
Sergey: “new companies” are easier to convince
Matthew: if old verisons of .NET would have supported it already - why would you build a new one?

Justin: Is it for all .NET?
Sergey: Yes

Mathew: was it done before?
Sergey: SpringBoot?

Parsing of logs and extracting the logs - should be a format.

Action items: please send us feedback!

Wrapping the day

- Feedback on the first day
- Review agenda to tomorrow

Daniel D: didn’t know what to expect. Response header is interesting, hope to get to it tomorrow. Response headers discussion was the most interesting and insightful.

Daniel: response headers topic and correlation context was the most interesting. Tomorrow interested in talking more about correlation context. Like discussions and learned a lot.

Christoph: Very good day, good discussions. CC is very interesting and may be very interesting for DynaTrace. Kudos to Sergey!

Bob: correlation context was very cool. Ideally trace context should be simple. But it is cool to see how complicated it may be and how everything converges together.

Zeke: Response headers use cases and CC use cases very very helpful. Customer focus important.

Chris: First time and great to meet everybody. Response header conversation was the best. In scope for W3C and it's great. More about response header and in-process context.

Martin: The most interesting response header. Was working on browser and some uses cases were very new. Wants to implement it in future. OpenTelemery space is also new and trying to catch up.

Justin: New and excited! My favourite part is to talk about customer part and motivation to upgrade.

Victor: Enjoyed response header conversation and CC use cases. Every meeting we are getting closer and closer to the value of it. Wants to build success stories for W3C and help customers solve real stories. For tomorrow - Trace exchange format would be a good idea to talk about. Expectation is prioritization exercise on what to do next

Lionel: learn and get a sense of what's happening and how it may intercept with experience of what is building. Wants to talk about ultimate experience of how it will look like to the customer in the end.

Matt: Always excited to meet other vendors and see how it's coming together over time. The real world situation that we are trying to solve. Correlation Context getting more concrete. Maybe there is a third header or maybe not even part of W3C. Unifying contexts. Worth talking about it tomorrow - roundtable. Got more concrete on response headers. Christoph's document for tomorrow should be quite interesting. Want to understand browser concerns more.

Anssi: Correlation context is the most interesting part. Almost too valuable to lock into OpenTelemetry. Security and privacy implications are super important. Separate deep dive on security. Defaults must be secure. Shouldn't be a tradeoff

Alex: Goal was to learn and also first timer. Was interested in discussion of a transition plan. Very interested to see how this journey is not easy and painful. Wonder how that may be done better going forward.

Michele: first timer, looking forward to learn more about correlation context and see where we going to with the response headers.

Yuri: Wish to get more concrete on some actionable steps and agreement.

Day 2

New person Eyhab Al-Masri, University of Washington. Works with IoT and interested in MQTT

Trace Context in Text and Binary Protocols - Sergey

Link to sergey's notes coming later?

Many decisions about traceparent and tracestate headers were made because of http header limits. Pure text protocols typically have fewer limitations.

Cloud Events has a distributed tracing extension. CloudEvents supports unicode but apply limitations to their tracestate that our headers apply (length, no unicode, etc). If they allowed unicode, it would need to have an interop format for propogation further in the trace.

Traceparent typically not a problem with text protocols, but tracestate may cause more issues.

HTTP seems to be one of the strictest in terms of character and length limitations

Trace context implementation on other protocols: pure text, binary

Current format

```
traceparent: 00-0af7651916cd43dd8448eb211c80319c-b7ad6b7169203331-01
tracestate: vendorname1=opaqueValue1,vendorname2=opaqueValue2
```

Text protocols

CloudEvents, AMQP, etc.

traceparent: typically doesn't cause any issues.

tracestate: typically text protocols has less limitations on charset than http. Typically support full unicode.

In OpenTelemetry we named protocol formatter: HttpTextFormat. Issue on dropping http: [open-telemetry/opentelemetry-specification#317]

Document conversion of unicode strings to tracestate entries.

JMS had weird restrictions for the key.

Do we want to define rules for unicode to ascii?
What's the use case? Something traveling from http -> http -> unicode?
Should we ignore unicode characters when translating back to ascii?

What libraries do with unicode characters? Just ignore the key/value pair.
Alternative is, you parse your part of tracestate and leave everything as-is. Depending on library, you might get errors if you leave the unicode in there.

We have a spec for HTTP and we want to extend it for all other text protocols. Should we put it in the same spec?
Phillipe - could be part of the next version of this same spec. Yes, should be part of the same spec.
So then, do we add more sections for other protocols?
In the current space, we have a link to other protocols that links to a registry for other protocols that link to a separate spec.
For text protocol, it's such a small change so seems like it should be in the same spec

Action Item: Create a Github issue to add general text format to the next version of the spec. Philippe will add MQTT to the registry. (we never wrote the MQTT spec, thus it's not listed in the registry. https://github.com/w3c/trace-context-mqtt/)

Switching gears to binary protocol…
The binary representation for traceparent is easy. Very similar to how protobuf works.
As a parser, do i need to validate that the second field is 0 and last is 1?
Yes, helps validating that this is a traceparent.
Versioning breaks because parsers won't expect differences. Should we remove it? What's the value of currently implemented things vs new binary protocol requirements.

For tracestate, same kind of process: field describes key length, then key, field describes value length and then value follows that.

Problems/Questions:

Should we use 1 field or 2 fields? (Github issue on this)
Putting in a single field would be more efficient. In previous discussions, the open source projects preferred not having to parse and deal with tracestate.
Action Item: Group members - go vote for one of the options in the github issue: Use a single field instead of two

Second issue: Compact encoding for strings
Sergey: One option is to use our own encoding to minimize the amount of data we transmit. Since we don't encode ascii for http, should we do the same for binary? This safes effort for platforms that deal with both.
Justin: Keeping it opaque is preferred. Otherwise, need to trust implementers to parse and encode without breaking other implementor's trace state.
Christoph: Could reuse the same algorithm as we have for parsing the text? Almost…

Binary protocols

Protocol proposal.

Traceparent example: https://w3c.github.io/trace-context-binary/#traceparent-example
Tracestate example: https://w3c.github.io/trace-context-binary/#tracestate-example

Active issues:

1. Use a single field instead of two
2. Compact encoding for strings
3. The need to parse tracestate. If there is no compression - why not simply write it as-is to binary in ASCII?

Yuri: at which point do you want to choose binary protocol? Kaffka has string to byte array properties. You can use http or binary - both will work. But the receiver will not know which one it is. How to distinguish binary from text?

Sergey: Magic character?

Yuri: no. Maybe not have binary encoding at all? If we can change specification, than we can benefit from binary protocol. Let's just have strings.

Christoph: for Dynatrace savings of binary protocol wouldn't make a big difference at the moment.

Christoph: similar things applies to GRPC. You choose whether it's binary or not. Should we define it based on protocol?

Dan: always use binary when you can may be a good general rule.

Sergey: different name for the field: traceparent-bin, traceparent.

Justin: 26 bytes saving may not worth it.

Eyhab: Typical size of identifiers are 12-16 bytes. For “root” devices it makes sense to omit span-id.

Christop, Victor: perhaps default parent-id to the half of trace-id.

Michele: IoT devices looks like “mobile” monitoring. POS terminals, cameras typically don’t send any telemetry. But customers really want telemetry from the devices.

Questions/actions:

- Tracestate - Christoph: let’s stay with text
- Trace-parent - Christoph: trace-parent-bin sounds desirable. Keep span-id.

Daniel: Elastic was also very interested. (thomas watson)

Trace context implementation on other protocols: AMQP, MQTT

Messaging

https://github.com/w3c/trace-context-amqp
https://github.com/w3c/trace-context-mqtt

Format - nothing too complicated. Suggestion is to use binary.

In AMQP, there are different baggage for message:
1. Transport layer properties
2. Message layer properties
Some SDKs only provide message property access. If you put it in transport layer, you can’t change spanId in every hop.
Christoph: why would you want to change spanId in the transport hub?
Sergey: some architectures have many processors for the same message.

Justin: Does this mean that the AMQP broker itself would have distributed tracing built in?
Sergey: yes.
Sergey: binary format is preferable so our previous discussion about having a separate name for binary version would fit here.

Discussions:

1. Message-id vs. TraceContext. - typically, a message system generates a messageid. TraceContext doesn't fit into messageid. Protocol owners don't like the idea of tracecontext stomping on messageid.
    a. Never used.
    b. Message ID as a correlation but use whatever messaging system provides
2. Retries and TraceContext - in the context of http, every retry generates a new span. However, for messaging systems, the tracecontext is in the message itself, can't change tracecontext on a retry. We will always have the same parentID when we retry.
    a. Justin - possibly, retries would just show up as longer latency.
    b. Eyhab - could put retry count
3. Queue operation failure vs. TraceContext. TraceContext in metadata of the message that may fail to be parsed? How to identify it than?
    a. Many reasons why failures happen. Could be message format issue. Many brokers don't parse the message so a failure wouldn't be associated with the trace.
4. Batching - people tend to put more items in a single message than just one operation. TraceContext ends up being associated with the entire message rather than each item.
    a. Zeke - similar use case to fan-in operations where you're aggregating messages.
    b. Rebatching on the server
5. Injecting trace context into the message payload (mqtt v3)
    a. For v5, it supports metadata so same mechanisms for AMQP would work
    b. For v3, try adding trace-state and trace-parent at the root of the JSON object.
    c. There are problems, for example, if the message is encrypted the broker can't get access to tracestate.

Justin: this is complex to think about because we're assuming brokers will be producers of spans. If we treat them as opaque, it's easier.

Sergey: what's the experience for kafka? Typical scenario for injecting tracecontext?
Zeke: typically not using… have a lot of batching use cases so tracing not being used right now. Experimented with tracing where the producer is adding context and consumer reads it. Skipping broker producing anything for the trace.
Christoph: On the batch scenario, we'd pick one traceId
Zeke: linking traces is one way to solve that.

Sergey: minimum specification should recommend producers stamping tracecontext on the message and consumer reading it. Would it help vendors if the protocol defined standard?

Christoph: wondering if it's worth defining a tracecontext standard for messaging. Purpose is for intermediaries to not drop context. In messaging systems, this isn't the case.

Guidance for messaging brokers may be a good solution. And tell them don't mess with trace context.

Broker engagement is not necessary for the messaging. Victor: how can broker help customer. Perhaps we can have a better explanation of the value of a need for a protocol. How will it improve interoperability and customer experience.

Victor: perhaps logging from broker may be enough with the correct attribution with the ID. Or simply instrument producer and consumer.

Actions:

- Keep working on specs.
- Bring message broker representative.

Trace Context in Browsers

Bernard (Dynatrace) - browser linking of user actions to server activity is important. Cookies are bad solution.

If resource timing may have a trace-parent - this will be very helpful.

Proposal for trace-ids in server-timings:
https://docs.google.com/document/d/1jnvS-DCtaYH6Q7o1DeNutOHdMmafVJiJfhnQNFukTOk/edit

Web performance working groups (and server-timing) may be a group that can help with this.

Philippe: it will be a tricky conversation as identifiers would carry a whole lot of other resource timing issues.

Christoph: doesn't matter what API, but browser needs to provide this API.

Bernhard: browsers generating trace-id is another option. Browser cannot know about clicks as it's user code.

Dave: DOM JS execution context can be used there. Using DOM JS all sync and async calls, including XHR can be attributed with the trace-id in this “zone”.

Also joining it to browser resource API using heuristics APIs.

Sergey: what can be improved?

Dave: Server side will generate a new trace header if not received. This parent than returned back to the client as a header. So it can be used to attribute client spans. Than page load trace-id is linked to all clicks on the page. Cookie provide mapping for the entire user session. Perhaps another attribute on all spans.

Daniel: standard response header may help to improve this experience.

Dave: Server-timing header may be used here. Also make trace-parent a standard CORS header without prefetch.

Dave: passing back trace-id, sampling decision and parent-id back to the client. Load balancer in-between would be great as it can re-use the same headers for tracing.

Bogdan: we discussed yesterday that span-id will not be returned and if returned - it will be server span ID. However in this scenario we need a “desired” parent ID.

Bernard: this scenario ^^^ makes sense.

Matt: in this case trace-parent would make sense.

Dave: returning “my span”, not the “desired parent ID” will put additional burden on visualizers as they need to understand links.

Daniel: “desired parent” header names “trace-parent” may be confusing.

Dave: may be not that confusing is you creating smaller spans

Dave: browser support may be useful to get zone.js context and attach it to the static images querying.

Alex: if you always return “desired parent” - will it break anything?

Bogdan: doesn’t see why it will break anything.

Christoph: it will break systems assuming there is a parent and waiting for it to arrive.

[11:31 AM] Bob Strecansky
CDNs also sometimes pass ID headers already. Working previously at Akamai, I don't think they'd have a problem coming up with a standard implementation for something like this if we asked.

Bogdan: does it work with AMP

Dave: probably… but will be a different implementation.

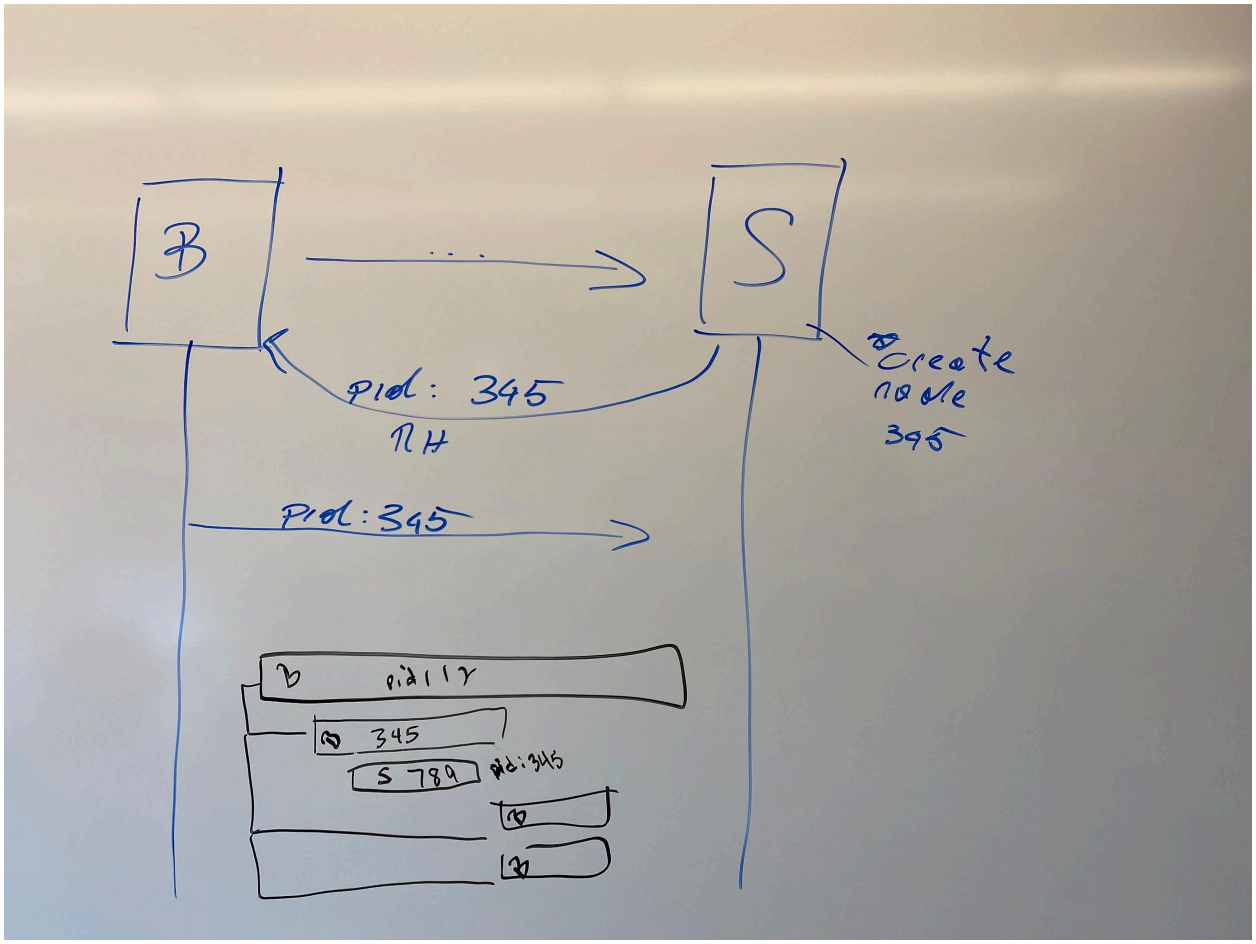
Static resources linking will still need heuristics… maybe cookies?

Configuration on where to send the trace-parent to? Dave: in OpenCensus there is a list of domains you can configure.

Dave: ask for CDN to respect trace-parent is very similar to istio.

An example polyfill service for those uninitiated: https://github.com/Financial-Times/polyfill-service

Image from whiteboard:



Action items:

- Response header - keep pushing
- CORS - start conversation. Also version of browser check will be needed.
- Context in browsers associated with the static images requests
- CDNs - ask them to respect trace-parent and/or create trace-parent ID.

Response header

Use cases

Discussed. See notes above.

Format

Version - traceid - parentID - TraceFlags

Optimizations

- Only return sampling decision (needed for scenario when incoming trace-parent present, but sampling decision is delegated).
- Browser scenarios - need all fields
- Restart trace scenario (crossing trust boundaries) - we may only need TraceID, optionally flags.

Required/optional

Whether we return it all the time when incoming request didn't have trace-parent? Or only when configured

Bogdan: Can we have a flag in incoming trace-parent header to return sampling decision. This could be another bit in the TraceFlags bitmask

Justin: We could leave the delimiters in when omitting a field:

version-traceId--traceFlags
version-traceId--
version---TraceFlags

Sergey: People may start sending trace-parent with omitted fields as a request header

Sergey: Even if we plug response header to the Server timing we still need a single string format for the response header, we wouldn't want to go with name-value pairs like traceid=XXX,traceflags=YYY.

Christoph: alternative is trace-parent is always full trace-parent. But if you want a shorter one - use tracestate.

Zeke: What happens when an instrumented service returns a header, but is the root of the trace. It returns a parentId, even though it is a root

Justin: The 'root' service can just record the parent as a separate attribute. 'Ephemeral parent id'

Daniel: We can change the body of the response header based on whether you received a traceId in the incoming header. If there is a traceId incoming, you don't need to return the parentId. If there is not (browser), then return a parentId

Delegated sampling: very nice looking scenario.

Victor: what would Lightstep do as it makes sampling decisions in satellite. From library perspective everything is sampled.

Bogdan: same possible in OpenCensus.

So we are saying Amazon LB always sample. From cloud provider it's ok.

Sergey: Should LB propagate header back?

Browser -> LB -> App scenario may break if LB asks for deferred sampling while browser is expecting for the full header.

Load Balancer will need to return the full trace-parent. AND it needs to export all spans to the backend.

For the browser use case, if there is a load balancer (or anything else) in between, you will not be able to use trace state to stitch a vendor's portion of a trace together, since the browser won't send the trace state header.

List from yesterday:
OpenTelemetry JS (browser): Dave Raffsenberger @ Google, Hamid @ Elastic, Mayur Kale @ Google, Bartlomiej Obecny @ Lightstep, Daniel Dyla @ Dynatrace, Martin Kuba @ New Relic
For LBs: Bogdan.

We will start with creating with the description of a browser scenario where server can be configured to return the **full trace-parent (with ephemeral parent).**

Optimizations may be addressed later as an optimizations on top of it.

Daniel: create an issue.

Working group priorities and what to do next. Touch on TraceData

Victor (NewRelic):
Response headers and browser scenarios are important.

Data format for TraceData + API standard to retrieve information from other vendors. Those are not a priority
Also integrating with OpenSource is important (importing from Zipkin). Now support Zipkin JSON v2.

Tracestate directory (registry of vendors).
Registry:
Christoph: Why mandatory field “documentation on what is inside”?
Sergey: the idea was to avoid name squatting.
Lionel: scenarios document?

Victor: let’s just put our prefixes there.
Michele: how long the prefix must be? One example for tracestate is to put “site” or “tenant” name.
Bogdan: one may be better off by restarting trace
Christoph: each tenant will have their own parentID.
Michele: multiple installations in a single customer.

SamplingRate is another scenario.

Priorities:

1. Response headers
2. Correlation Context
3. Other protocols
4. Tracestate registry
5. Messaging systems and trace context (batch/merging traces)
6. TraceData

Michele: new topic. In Dapper we have DAG with parent/child system. Doesn’t work well for streaming and messaging scenarios. Should we discuss those in this group.

Sergey:maybe we have enough information with links. But visualization may be hard.

Michele: wonder if tracecontext needs to be messaging aware.

Tracecontext is designed for single request. But trace-id propagation written with the assumption that it will be preserved.

Michele: in case of sync communication we likely can understand request/response with response headers. In case of async - we may or may not see propagation of a previous trace-id. So analysis of these systems are getting very hard. There is no codified way to say that trace ID was restarted and how merged traceID will look like. How managed service providers will do restart and expose information about the fact restart happened.

^^^ Topic for the next meeting

https://docs.google.com/document/d/1joFqndm6TSDt6ymJ_6kfwmaXEO9vsoYLvok159Pzo6o/edit?usp=sharing

Wrapping up

Bogdan: Good progress on response header. Not happy that message busses wasn't quite ready. More ideas on what response headers may return. Happy with the progress.

Matt: Discussion on protocols and message queue was interesting and quite deep. We need suggestions from people who actually instrumenting message brokers. We need to put feet in a door and figure more details later. Good progress on response headers was great. Like how opencensus utilize what they can. Two to three days is good enough. Maybe smaller workshops - breaking into smaller groups for the second day.

Martin: Browser, tracing use cases are interesting. Something wasn't thinking about before. Want more than 2 days. Feels like it was not enough. Smaller groups meetings would be useful.

Christoph: We have lot more to talk about response headers. What makes sense to standardize, what is just a discussion for implementation. What browser may implement. 2 or 3 days is great. Focused work is a great idea.

Daniel: Smaller groups would be a good idea. Touched on lots of topics. Not finished on any of them. Maybe better prioritization. Response headers is very hot and important. No problem going to third day. More focus is important. Let's bring experts to the room next time we discuss message brokers.

Justin: enjoyed correlation context discussion. Changed the opinion that there were no value there. Message queue was an interesting topic. Longer session would be interesting.

Alex: let's bring experts. Maybe next time can bring the right people to the room. Happy to have Anssi yesterday. Might be better to have him here today as well.

Daniel: Three days with break up day would be interesting to try. CC is interesting topic. Really enjoys industry exchange.

Zeke: One suggestion - sections with dedicated person preparing materials were much better. Need to get more things written down ahead of time. Like prioritizing excercise.

Victor: Great we are talking about whitepapers and how to guide people in these ideas. We are in 0.00001% of people who know how to do it and all the complexity. Response headers turned out to be more complicated than incoming headers. We wrapping up the first phase of work of finalizing spec. Like the free form discussion as it brings people to the level of knowledge and inderstanding. Maybe better than 1 hour presentation without discussion. 2 days is good, Three days with breakout would be ok. Kudos to Sergey.

Michele: Love the industry exchange. Looking forward for the next round. Timezone was prohibitive, maybe Europe next time.

Bob: Enjoyed last two days. Scheduling may be adjusted. Work on OT will be enhanced by what learned today.

Eyhab: wish to find out about the meeting earlier. Have ideas on what can also be explored for iOt devices.

Philippe: great meeting, amazed by what we are doing. Really want to join physically. Will publish

Sergey: Thank you!

Next W3C TPAC: Vancouver - 26-30 October 2020 - Sheraton Vancouver Wall Centre