Unifying Observability: The Power of a Common Schema

FOSDEM (Monitoring & Observability devroom)
Feb 2024

Alex Wert | Engineering @Elastic
Christos Markou | Engineering @Elastic
About us...

Alex Wert
Maintainer @ OTel SemConv

Christos Markou
Approver @ OTel SemConv
OpenTelemetry Semantic Conventions
The History of Open Source Tools & Standards for Observability
O11y - The History of Open Source Tools & Standards

- **Microservices**
- **Jaeger**
- **Prometheus**
- **OpenMetrics**
- **ECS**

- **2012**: Zipkin
- **2015**: OpenTracing
- **2017**: OpenCensus
- **2019**: Jaeger, Prometheus, OpenMetrics, ECS
**O11y - The History of Open Source Tools & Standards**

**service.name**
Name of the service data is collected from.

The name of the service is normally user given. This allows for distributed services that run on multiple hosts to correlate the related instances based on the name.

**http.request.method**
HTTP request method.

The value should retain its casing from the original event. For example, GET, get, and Get are all considered valid values for this field.
O11y - The History of Open Source Tools & Standards

- Microservices
- Jaeger
- Prometheus
- OpenMetrics
- ECS
- Zipkin
- OpenTracing
- OpenCensus
There are 14 competing standards.

14?! Ridiculous! We need to develop one universal standard that covers everyone’s use cases. Yeah!

Soon:

There are 15 competing standards.
O11y - The History of Open Source Tools & Standards
OTLP - Signals - Semantic Conventions

ResourceLogs

Resource

Attributes
- host.name: my-host.xyz
- service.name: my-service

LogRecord

severity_text
body

Attributes
- http.request.method: GET
- http.route: /users/:userId
- client.address: 11.12.13.14

(simplified)
O11y - The History of Open Source Tools & Standards
There are 14 competing standards.

14?! Ridiculous! We need to develop one universal standard that covers everyone's use cases. Yeah!

Soon: There are less competing standards.

With OTEL
Announcing the Elastic Common Schema (ECS) and OpenTelemetry Semantic Convention Convergence

By Reiley Yang

Today, we’re very excited to make a joint announcement with Elastic about the future of Elastic Common Schema (ECS) and the OpenTelemetry Semantic Conventions.

The goal is to achieve convergence of ECS and OTel Semantic Conventions into a single open schema maintained by OpenTelemetry, so that OpenTelemetry Semantic Conventions truly is a successor of ECS. The Elastic Common Schema. OpenTelemetry shares the same interest of improving the convergence of observability in this space. We believe this schema merge brings huge value to the open source community because:

- ECS has years of proven success in the logs, metrics, traces, and security events schema, providing a common problem domain.
- ECS provides schema for security domain fields, which is an important aspect of telemetry.

As a result, expect to have more consistent signals across different pillars of observability and both Elastic and the OpenTelemetry community understand that converging two widely-used standards into one singular common schema, and having a smooth transition is critical for users. A dedicated OpenTelemetry Convention working group will be created with domain experts from both Elastic and OpenTelemetry to welcome domain experts who are passionate about data schemas and semantic conventions to join.

ECS SemConv
Benefits of the Merger

- Resource
- Trace
- Metrics
- Logs + Events

- Metadata
- Metrics
- Traces
- Logs + Events
- Security

OpenTelemetry

ECS
Challenges ...
ECS ↔ OTel: Challenges & differences

Breaking Changes / Merging Communities

Potential for schema conflicts & breaking changes
ECS ↔ OTel: Challenges & differences

**Structure**

- **ResourceLogs**
  - Attributes
    - host.name: my-host.xyz
    - service.name: my-service

- **ResourceSpans**

- **ResourceMetrics**

- **ScopeLogs**
  - LogRecord
    - Attributes
      - http.request.method: GET
      - http.route: /users/:userID
      - client.address: 11.12.13.14

**OTLP + Semantic Conventions**

**Plain field definition**

- **Container Fields**
  - Container fields are used for meta information about the specific container that is the source of information.
  - These fields help correlate data based containers from any runtime.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>container.cpu.usage</td>
<td>Percent CPU used which is normalized by the number of CPU cores and it ranges from 0 to 1. Scaling factor: 1000. type: scaled, float</td>
</tr>
<tr>
<td>container.disk.read.bytes</td>
<td>The total number of bytes (gauges) read successfully (aggregated from all disks) since the last metric collection. type: long</td>
</tr>
<tr>
<td>container.disk.write.bytes</td>
<td>The total number of bytes (gauges) written successfully (aggregated from all disks) since the last metric collection. type: long</td>
</tr>
<tr>
<td>container.id</td>
<td>Unique container id. type: keyword</td>
</tr>
</tbody>
</table>
ECS ←→ OTel: Challenges & differences

Semantic Conventions for HTTP Server Spans

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
<th>Req. Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>The matched route, that is, the path template in the format used by the respective server framework.</td>
<td>Cond. required</td>
</tr>
<tr>
<td>http.request.header.&lt;key&gt;</td>
<td>string []</td>
<td>HTTP request headers, &lt;key&gt; being the normalized HTTP Header name (lowercase), the value being the header values.</td>
<td>Opt-in</td>
</tr>
</tbody>
</table>

Semantic Conventions for HTTP Access Logs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
<th>Req. Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>string</td>
<td>Opt-in</td>
</tr>
</tbody>
</table>
Semantic Conventions - Attributes Registry

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>The matched route, that is, the path template in the format used by the respective server framework.</td>
</tr>
<tr>
<td>http.request.header.&lt;key&gt;</td>
<td>string []</td>
<td>HTTP request headers, &lt;key&gt; being the normalized HTTP Header name (lowercase), the value being the header values.</td>
</tr>
</tbody>
</table>

Attribute Definition

Semantic Conventions for HTTP Access Logs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Req. Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>Opt-in</td>
</tr>
</tbody>
</table>

Semantic Conventions for HTTP Server Spans

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Req. Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>Opt-in</td>
</tr>
</tbody>
</table>

Semantic Conventions for HTTP Metrics

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Req. Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.route</td>
<td>string</td>
<td>Opt-in</td>
</tr>
</tbody>
</table>
ECS ←→ OTel: Challenges & differences

Metrics format

**Metric:** `system.disk.io`

This metric is **recommended**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Instrument Type</th>
<th>Unit (UCUM)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>system.disk.io</code></td>
<td>Counter</td>
<td>By</td>
<td><a href="#">Table Row</a></td>
</tr>
</tbody>
</table>

**Attribute** | **Type** | **Description** | **Examples**
--- | --- | --- | ---
| `disk.io.direction` | string | The disk IO operation direction. | read |

**Example Metrics**

- **host.disk.read.bytes**
  - The total number of bytes (gauge) read successfully (aggregated from all disks) since the last metric collection.
  - type: long

- **host.disk.write.bytes**
  - The total number of bytes (gauge) written successfully (aggregated from all disks) since the last metric collection.
  - type: long
NOW

LET'S GET SOME WORK DONE
Examples

Add oci.manifest.digest, container.image.repo_digests and make container.image.tag array #159

~20 comments

Add semantic convention for IP addresses of a host #203

~23 comments
Evolution of the merger

- system/host metrics: moving towards stability
- process: ""
- container: 60%-> ongoing PR -> 100%
- http, network: ~50%
- databases, mobile: WiP
- cloud: WiP
- k8s: WiP
Evolution of the Sem Conv project during the past months:

The work is moving forward in a community driven way:

- joint efforts to improve the tooling
- working on the improvement of the "guideline"
- project re-structuring to group by topic
- introduction of attribute’s registry
- field reuse concept for OTel semantic attributes
How the community is organised around this and how the merger is moving forward

Working groups with domain experts focusing on the stability of the area
a) stabilizing the semantic conventions
b) tuning OTel implementations
How the community is organised around this and how the merger is moving forward

system metrics WG: board
db WG: project
security semconv WG: proposal
mobile area: approvers-group
containers/k8s: approvers-group
How the merger takes place in reality

1) Cross check of ECS and OTel SemConv
2) Check what the implementation of OTel collector and language SDKs follow
3) Proposal of merged fields
4) Open discussion in the community
   a) Measure breaking changes in both sides, if any.
   b) review cycles
5) Conclude and merge
6) Handle breaking changes
Summary

Merger is happening – contributions more than welcome :)

Community driven work

Goal: make OTel SemConv the one, unique and straightforward standard for O11y and Security
Where to find us / Questions

CNCF Slack

@AlexanderWert
@ChrisMark

Project Meetings

Monday 5:00 CET (SemConv working group)
Tuesday 5:00 CET (Specification SIG)
Thursday 5:30 CET (System metrics WG)