KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon
Europe 2022

Why, How to, and Issues:

Tail-Based Sampling in the OpenTelemetry Collector

Reese Lee, New Relic

AGENDA

- Introduction to OpenTelemetry, the Collector, and Distributed Tracing
- Sampling Overview
- Sampling in Action!
- Concerns and Limitations
- Closing
- Q&A

WHO AM I?

Reese Lee

Developer Relations Engineer,
OpenTelemetry Community Team,
New Relic

Also enjoys eating, watching horror movies, and paddleboarding.

INTRODUCTION

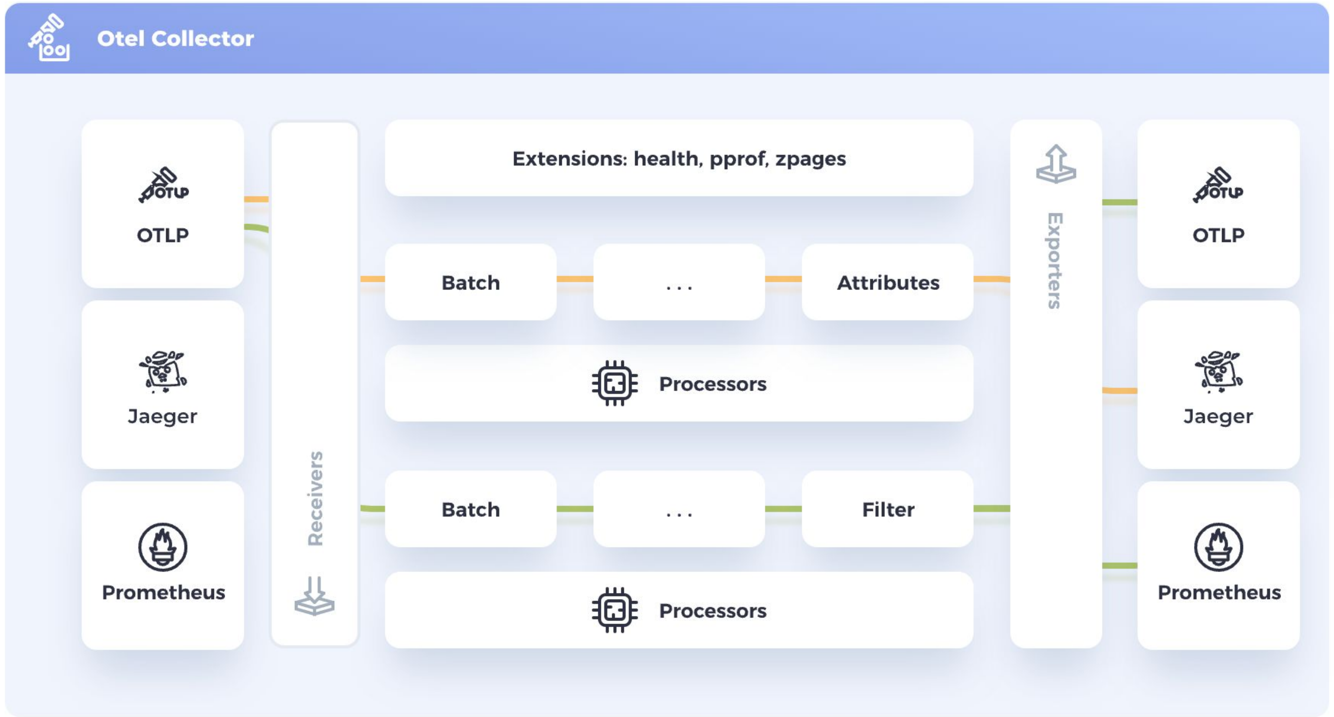
- What is OpenTelemetry?
- What is the Collector?
- What is distributed tracing?

What is OpenTelemetry?

- 2nd most active CNCF project
- Unified standard for generating and collecting telemetry
- Set of APIs, SDKs, and tools

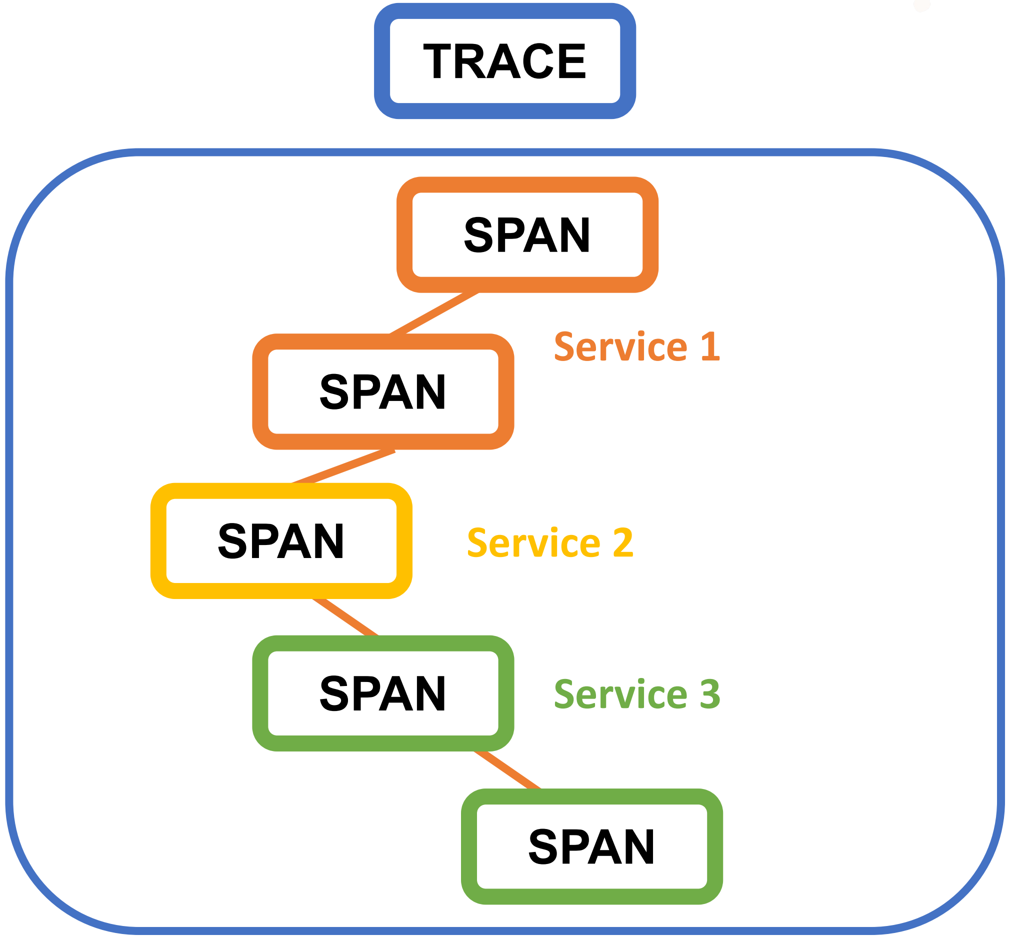
What is the Collector?

- Extremely configurable system for processing telemetry data
- Receivers, processors, and exporters
- Sampling, host metrics collection, data scrubbing, data normalization

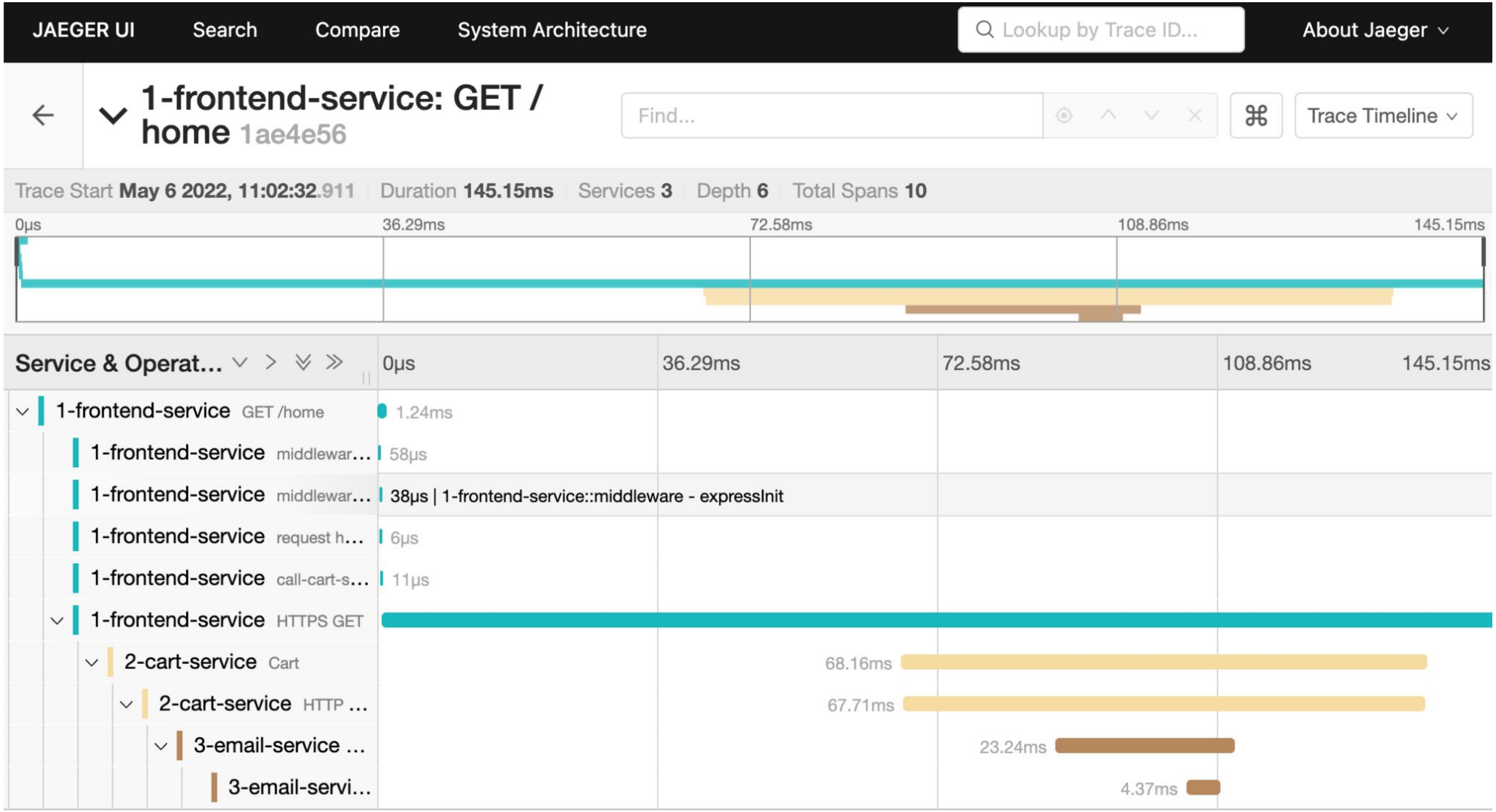


What is distributed tracing?

- Method of observing requests from one service to another
- Helps you understand your systems
- Traces consist of spans



What is distributed tracing?



1-frontend-service: GET / home 1ae4e56

Trace Start May 6 2022, 11:02:32.911 | Duration 145.15ms | Services 3 | Depth 6 | Total Spans 10

0μs 36.29ms 72.58ms 108.86ms 145.15ms

Service & Operat…	0μs 36.29ms 72.58ms 108.86ms 145.15ms
1-frontend-service GET /home	1.24ms
1-frontend-service middlewar…	58μs
1-frontend-service middlewar…	38μs | 1-frontend-service::middleware - expressInit
1-frontend-service request h…	6μs
1-frontend-service call-cart-s…	11μs
1-frontend-service HTTPS GET	
2-cart-service Cart	68.16ms
2-cart-service HTTP …	67.71ms
3-email-service …	23.24ms
3-email-servi…	4.37ms

What is distributed tracing?

We Can
SAMPLE!!
POST FEB. 15 TO FEB. 28
WAR PRODUCTION CO-ORDINATING COMMITTEE

SAMPLING OVERVIEW

- What is sampling?
- Why do I want to sample?
- Head and tail-based sampling

What is sampling?

- Reduce number of created spans
- Can be implemented at different stages of span processing

SPAN

To keep, or not to keep
(a span or a trace)?
That is the question!

SPAN

SPAN

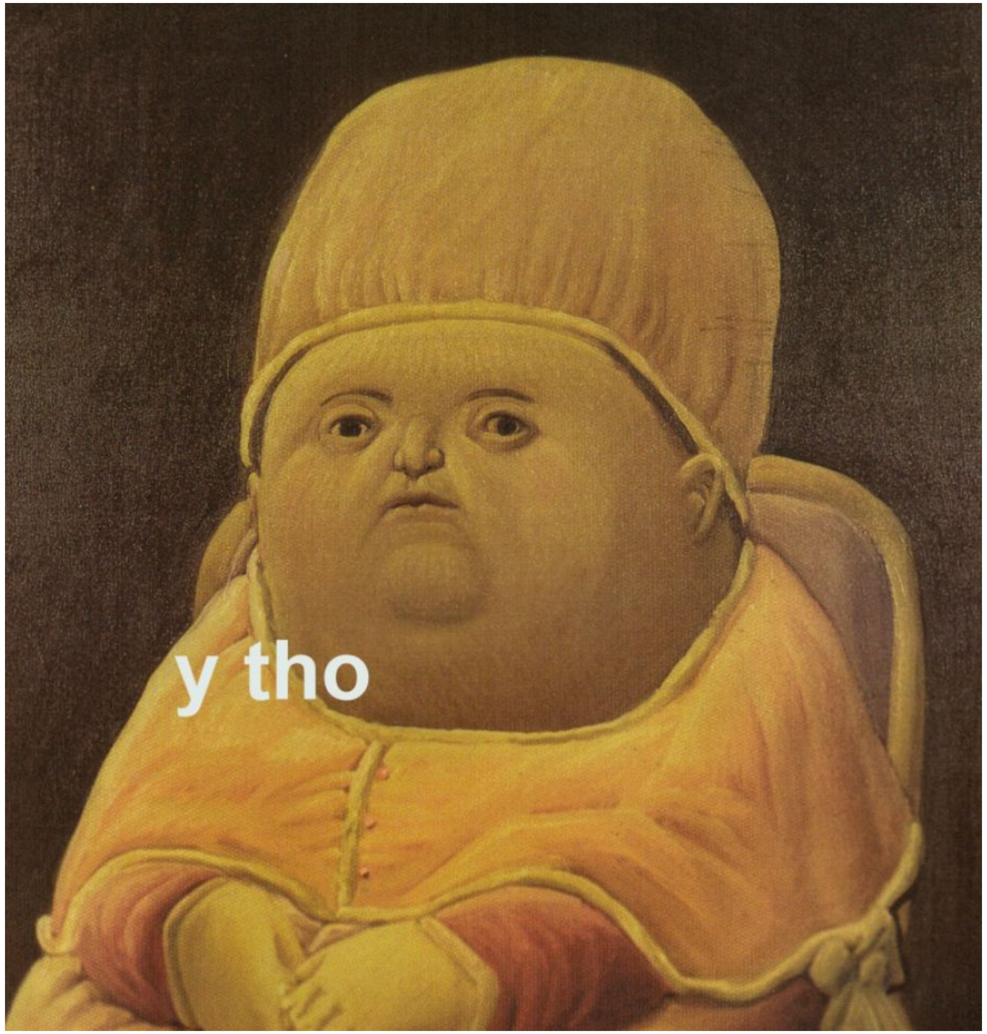
SPAN

SPAN

SPAN

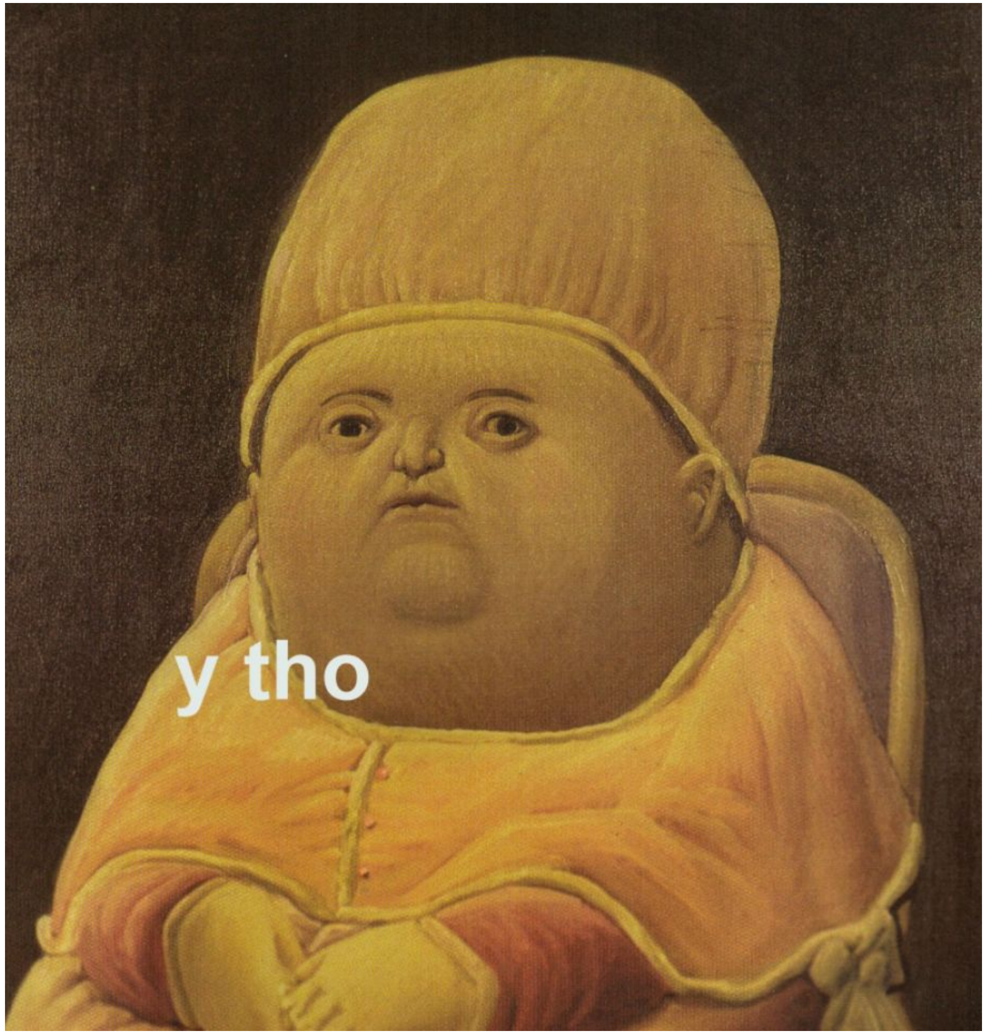
Why do I want to sample?

- To reduce cost of data egress, data storage, data ingest
- See only interesting traces, filter out noise
- Reduce performance impact of tracing every request*



Why do I want to sample?

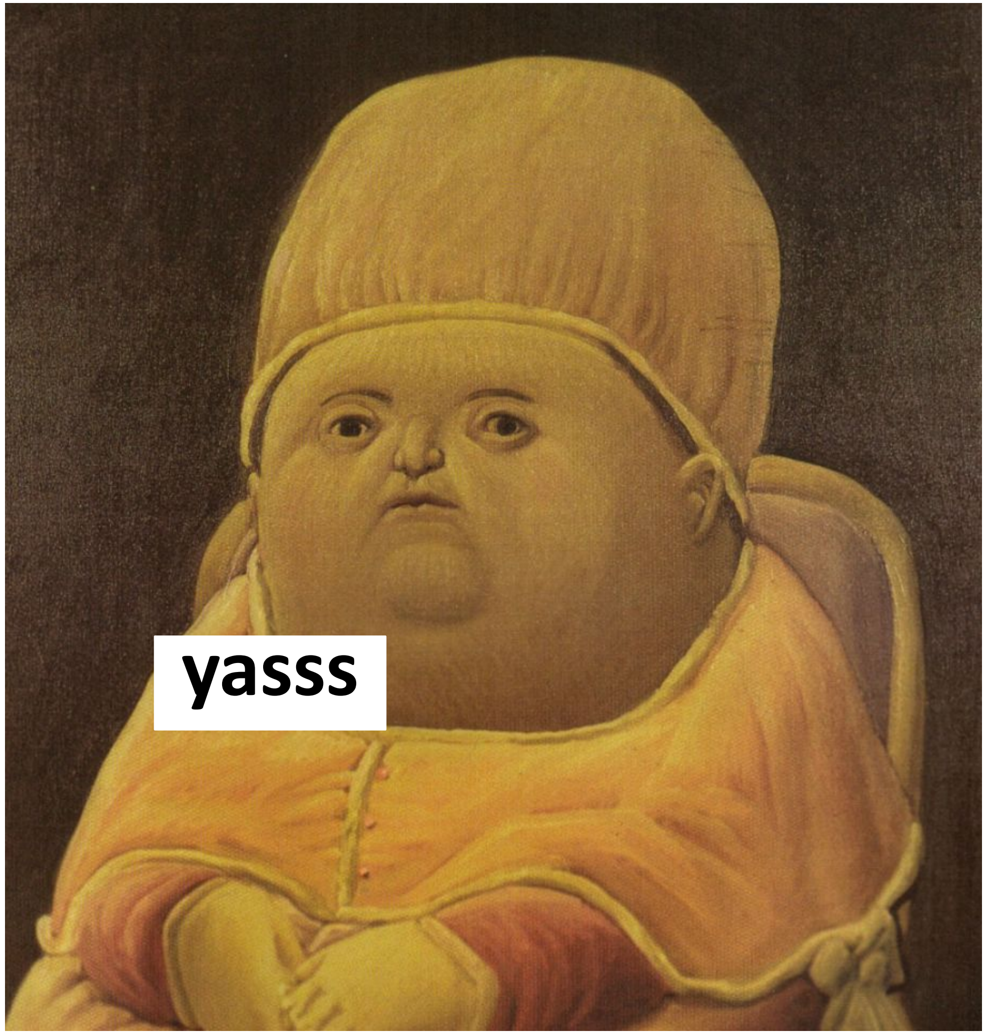
- To reduce cost of data egress, data storage, data ingest
- See only interesting traces, filter out noise
- Reduce performance impact of tracing every request*



*More applicable to head-based sampling

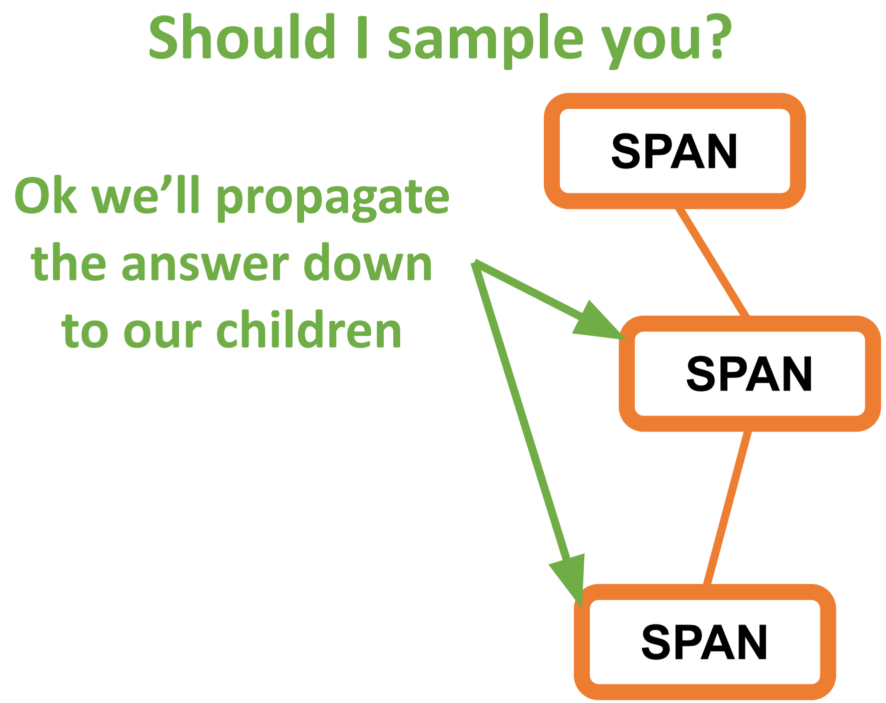
Why do I want to sample?

- You don’t need a ton of data to find the right insights - you need the right sampling of data



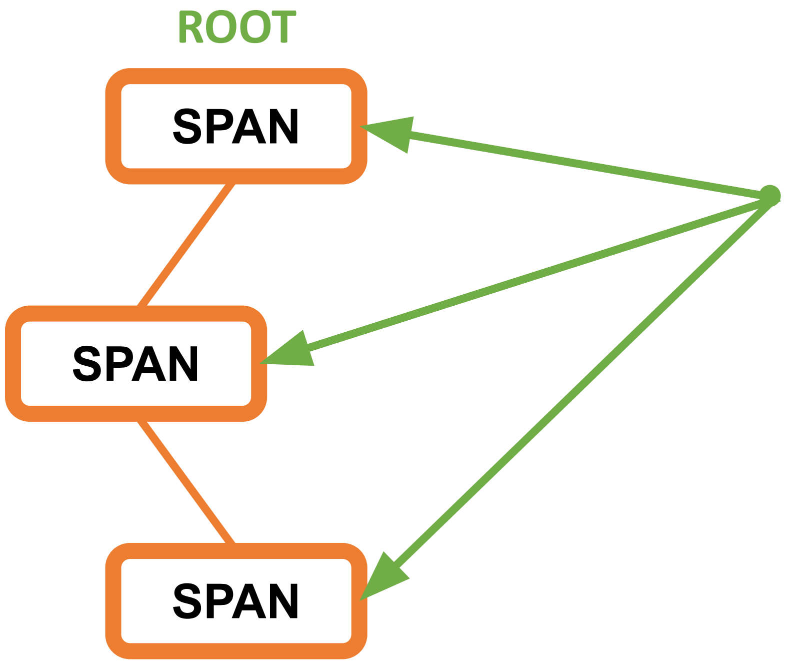
Head-based sampling

- Sampling decision is made before the span is created
- Simple, efficient, unbiased
- Built-in samplers include:
  - ParentBased
  - AlwaysOn
  - TraceIdRatioBased



ParentBased(root=AlwaysOn)

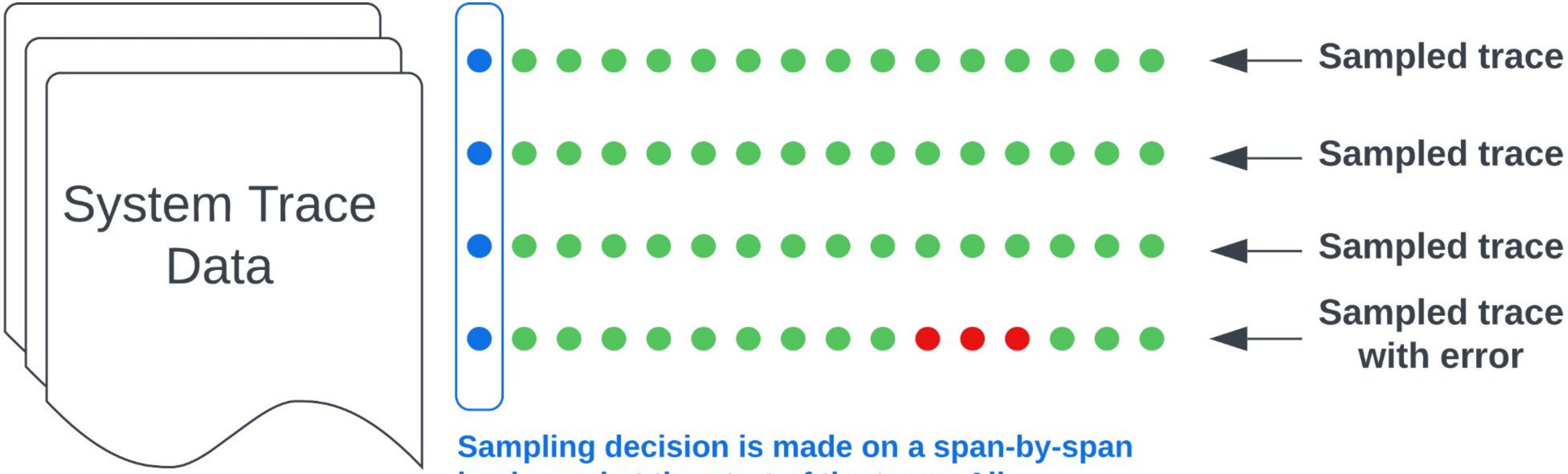
OpenTelemetry default



Are you a root span?

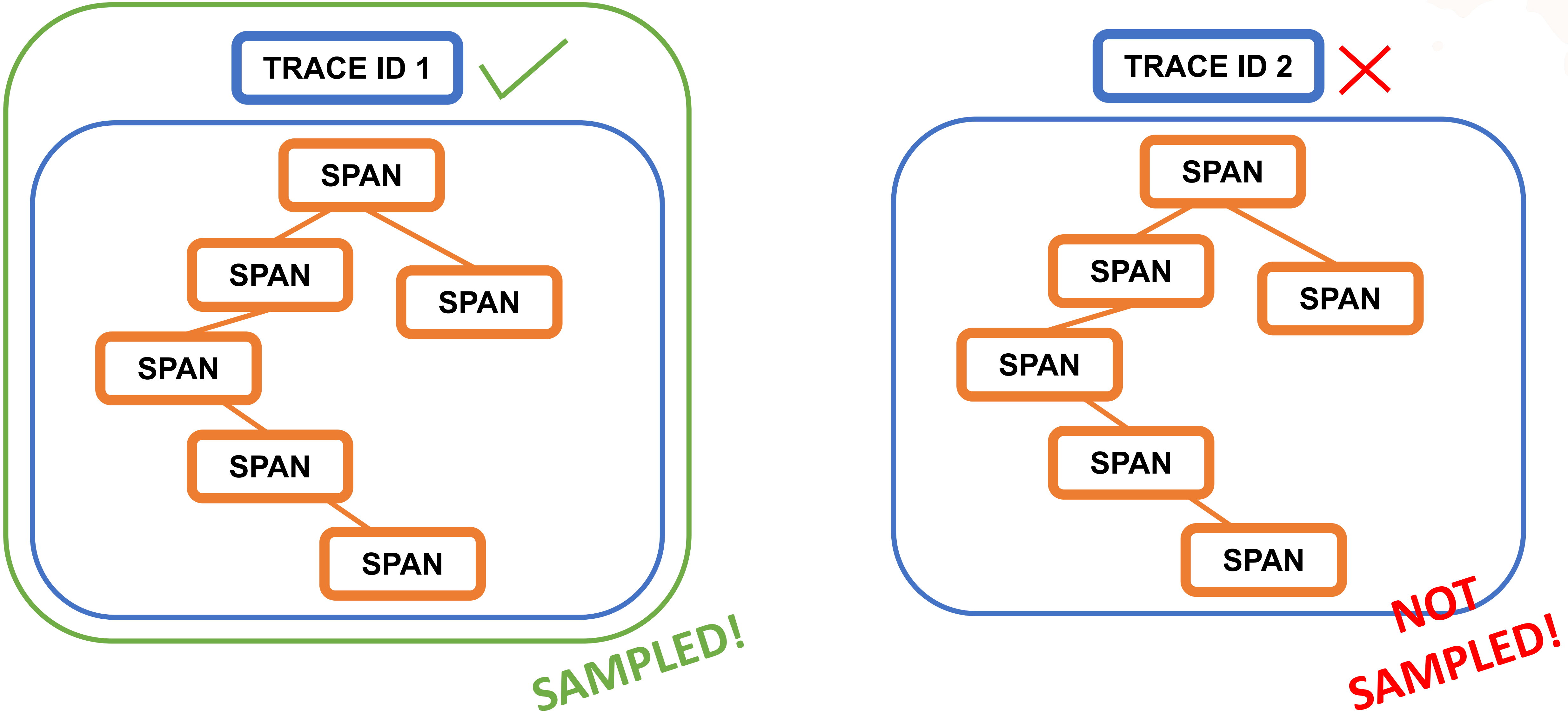
- YES: I will sample you!
- NO: Was your parent sampled?
  - YES: I will sample you!
  - NO: I will not sample you!

ParentBased(root=AlwaysOn)

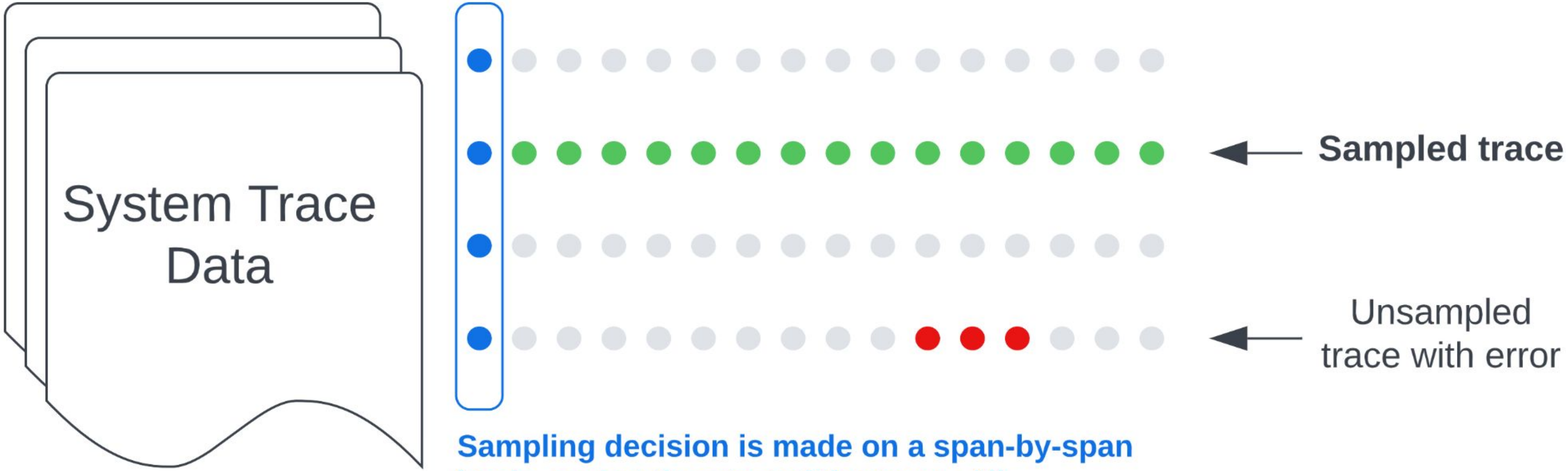


Sampling decision is made on a span-by-span basis, and at the start of the trace. All subsequent spans inherit its parent's sampling decision.

TraceIdRatioBased



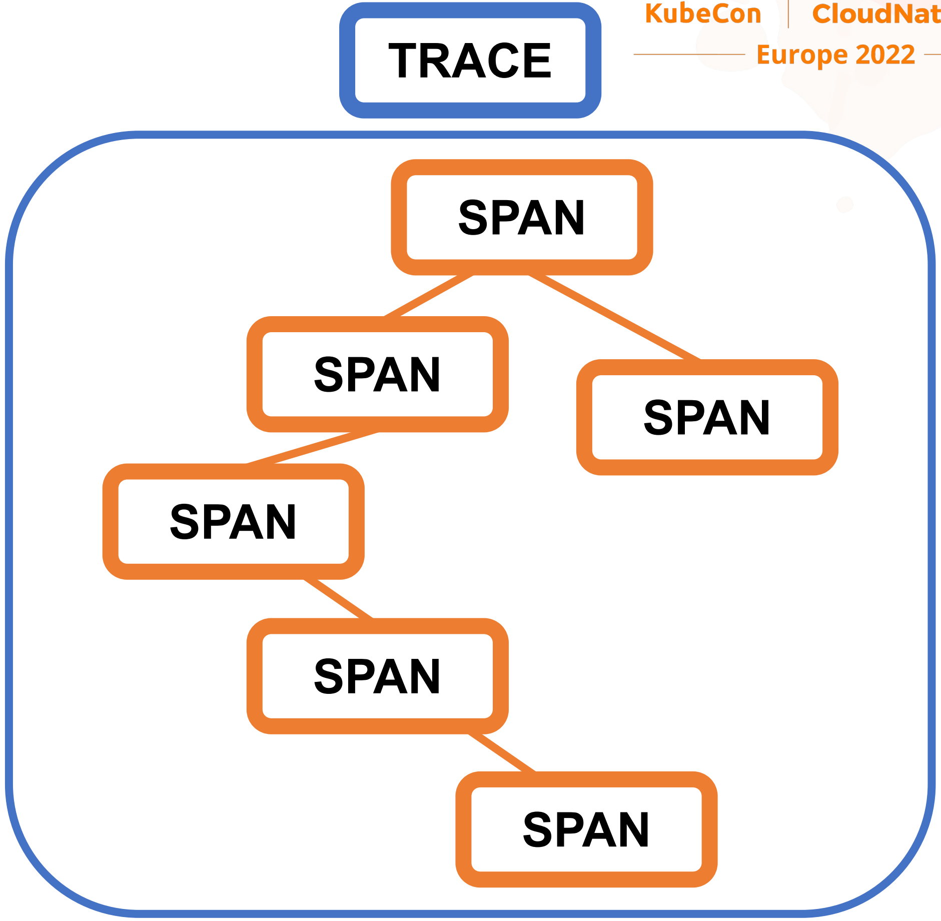
TraceIdRatioBased



Sampling decision is made on a span-by-span basis, and at the start of the trace. All subsequent spans using parent-based inherit its parent's sampling decision.

Tail-based sampling

- Sampling decision is made at the end of the trace
- Able to filter traces based on specific criteria
- Useful for efficiently seeing interesting traces
- Optimal

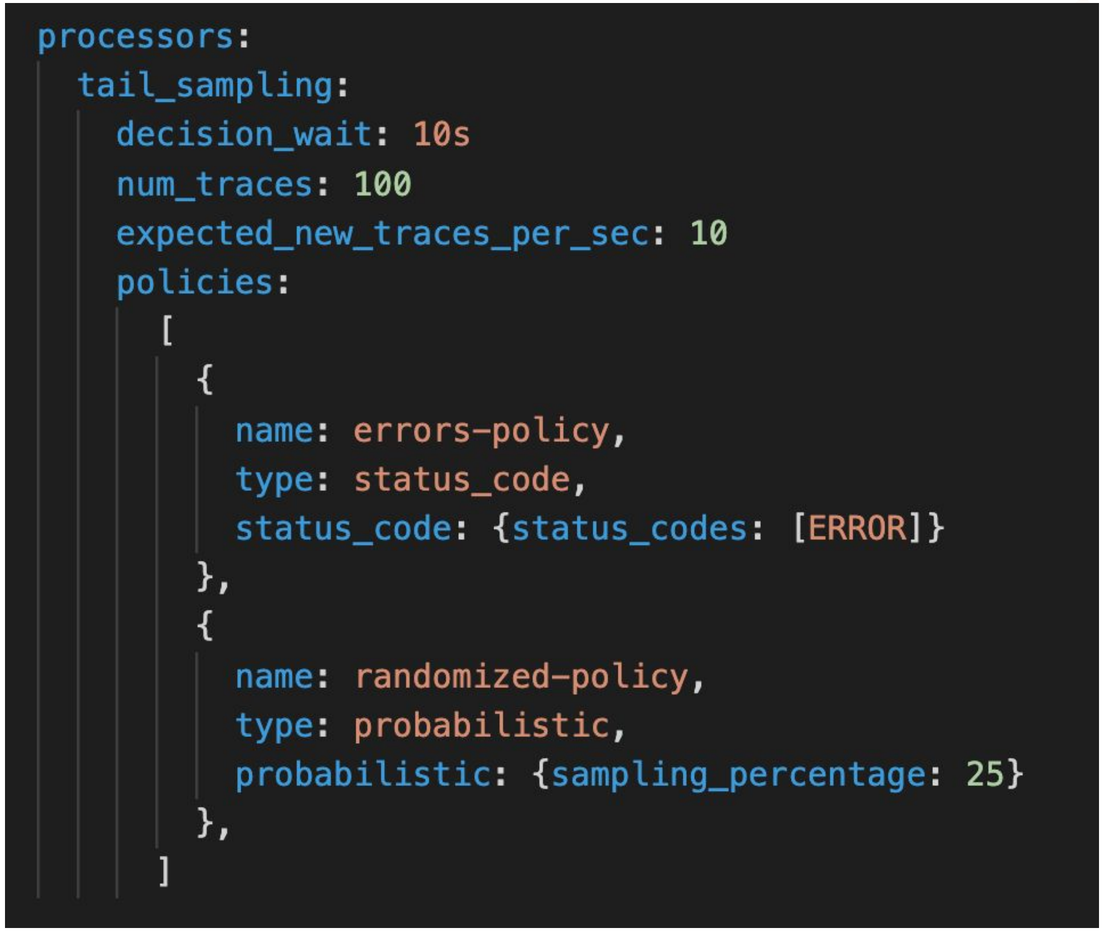


All the spans are done? Ok NOW I'm ready to make my sampling decision

Tail sampling processor

Policies include:

- Latency
- Status code
- Attributes
- “And”
- Composite

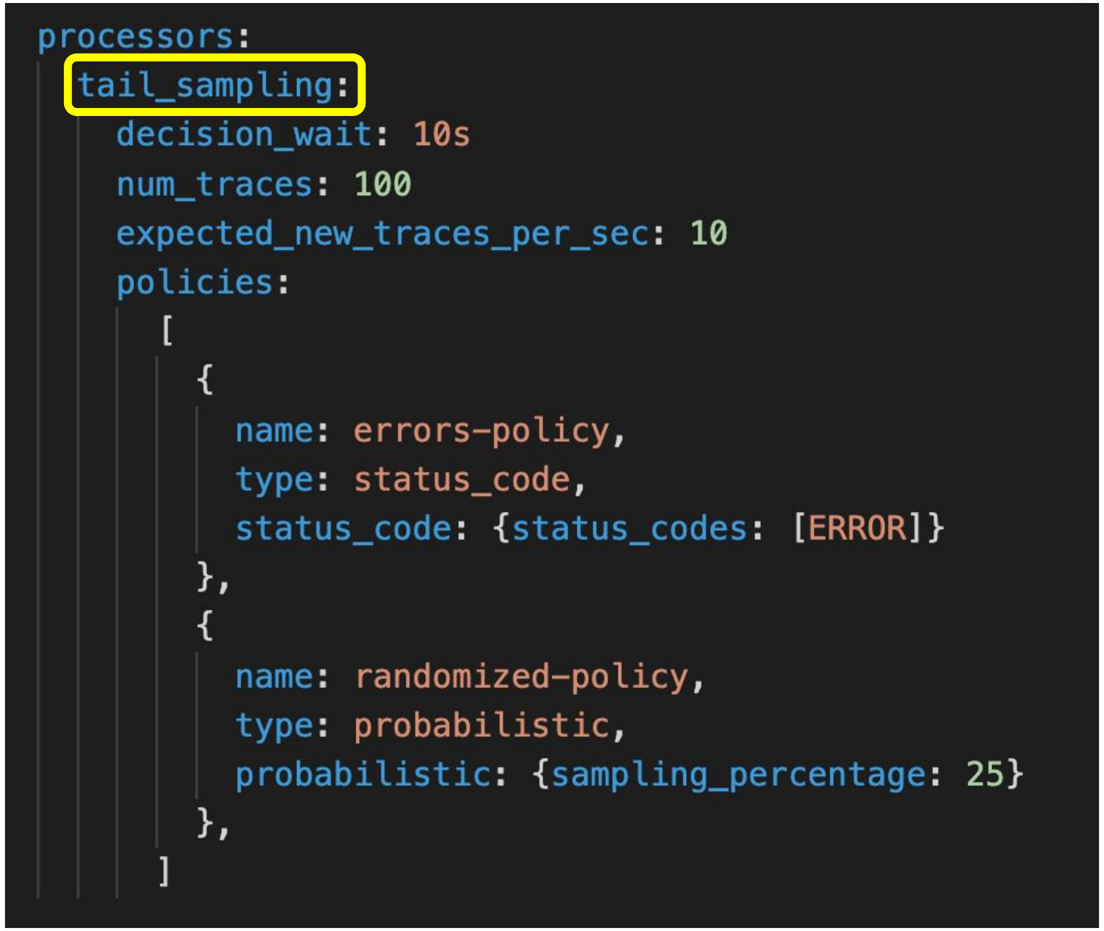


```
processors:
 tail_sampling:
  decision_wait: 10s
  num_traces: 100
  expected_new_traces_per_sec: 10
  policies:
   [
    {
     name: errors-policy,
     type: status_code,
     status_code: {status_codes: [ERROR]}
    },
    {
     name: randomized-policy,
     type: probabilistic,
     probabilistic: {sampling_percentage: 25}
    },
   ]
```

Tail sampling processor

Policies include:

- Latency
- Status code
- Attributes
- “And”
- Composite

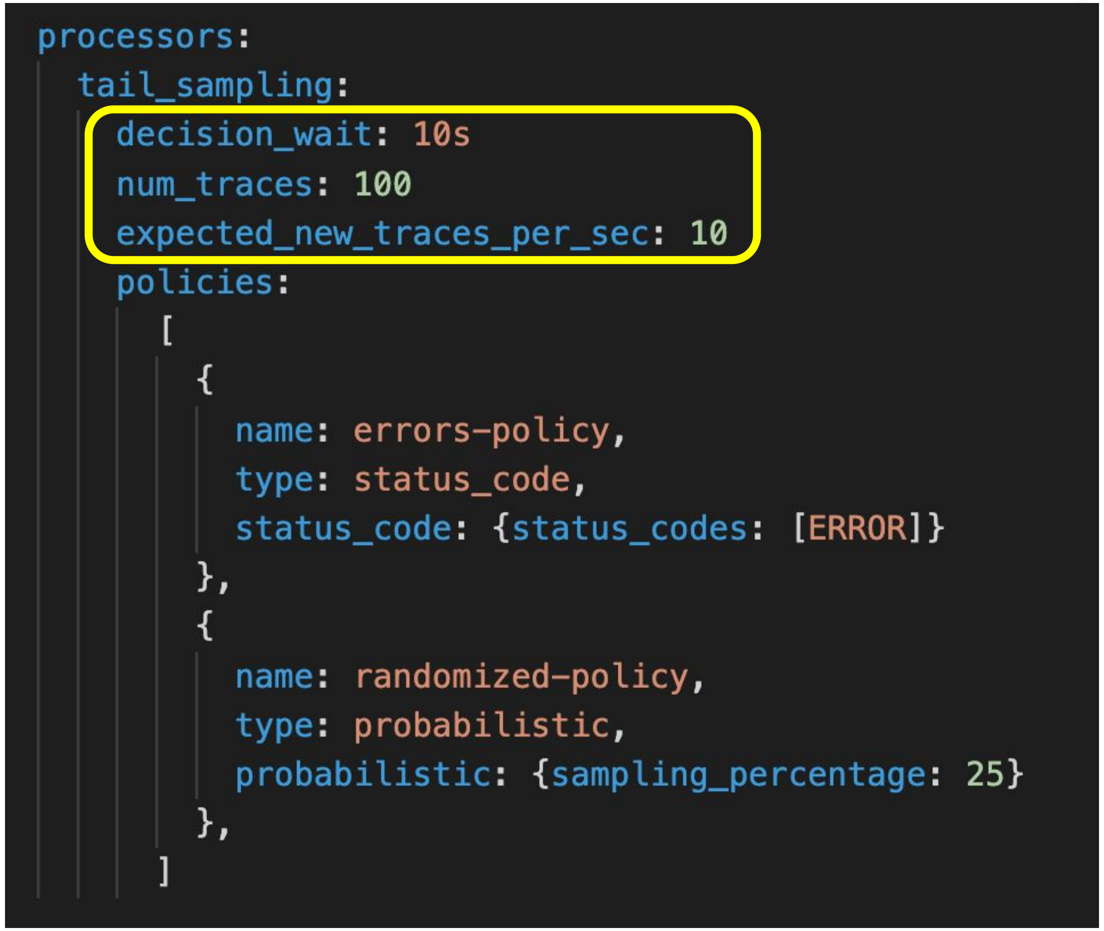


```
processors:
  tail_sampling:
    decision_wait: 10s
    num_traces: 100
    expected_new_traces_per_sec: 10
    policies:
      [
        {
          name: errors-policy,
          type: status_code,
          status_code: {status_codes: [ERROR]}
        },
        {
          name: randomized-policy,
          type: probabilistic,
          probabilistic: {sampling_percentage: 25}
        },
      ]
```

Tail sampling processor

Policies include:

- Latency
- Status code
- Attributes
- “And”
- Composite

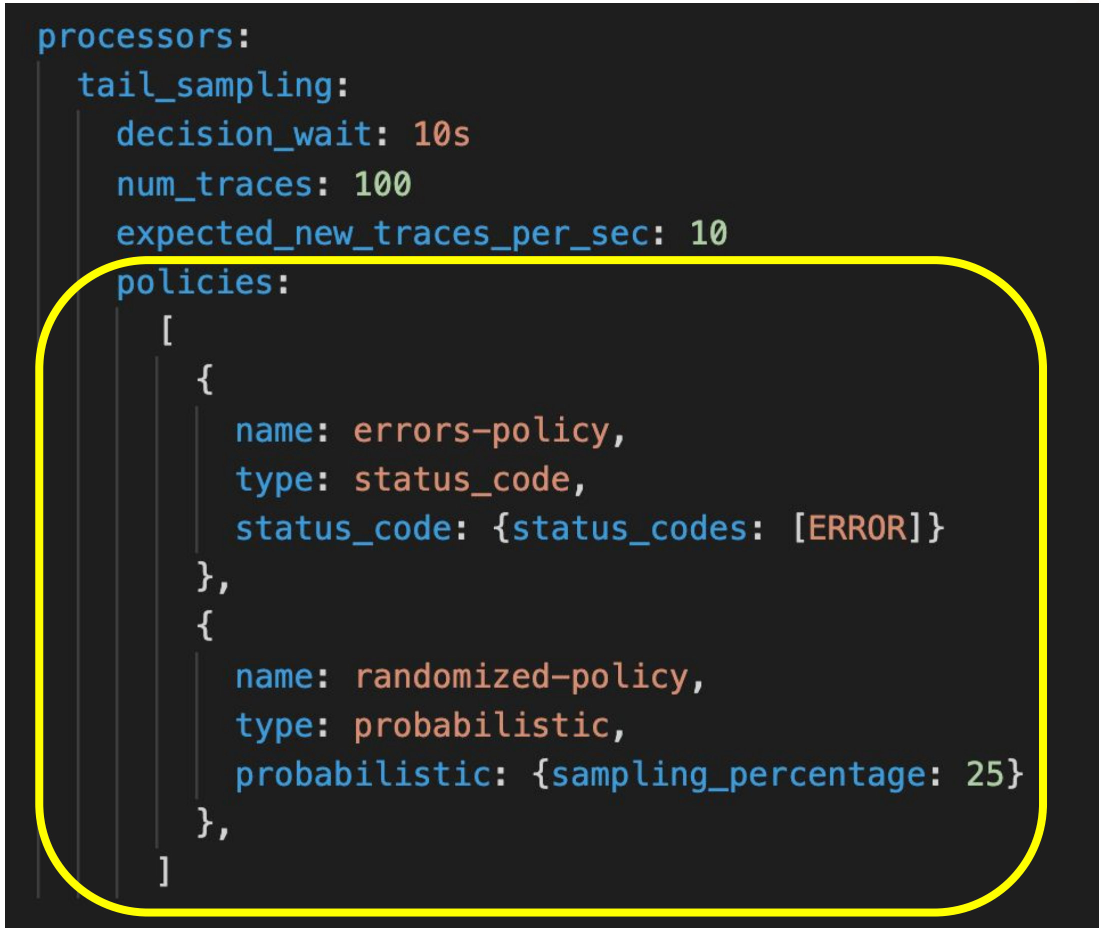


```
processors:
  tail_sampling:
    decision_wait: 10s
    num_traces: 100
    expected_new_traces_per_sec: 10
    policies:
      [
        {
          name: errors-policy,
          type: status_code,
          status_code: {status_codes: [ERROR]}
        },
        {
          name: randomized-policy,
          type: probabilistic,
          probabilistic: {sampling_percentage: 25}
        },
      ]
```

Tail sampling processor

Policies include:

- Latency
- Status code
- Attributes
- “And”
- Composite

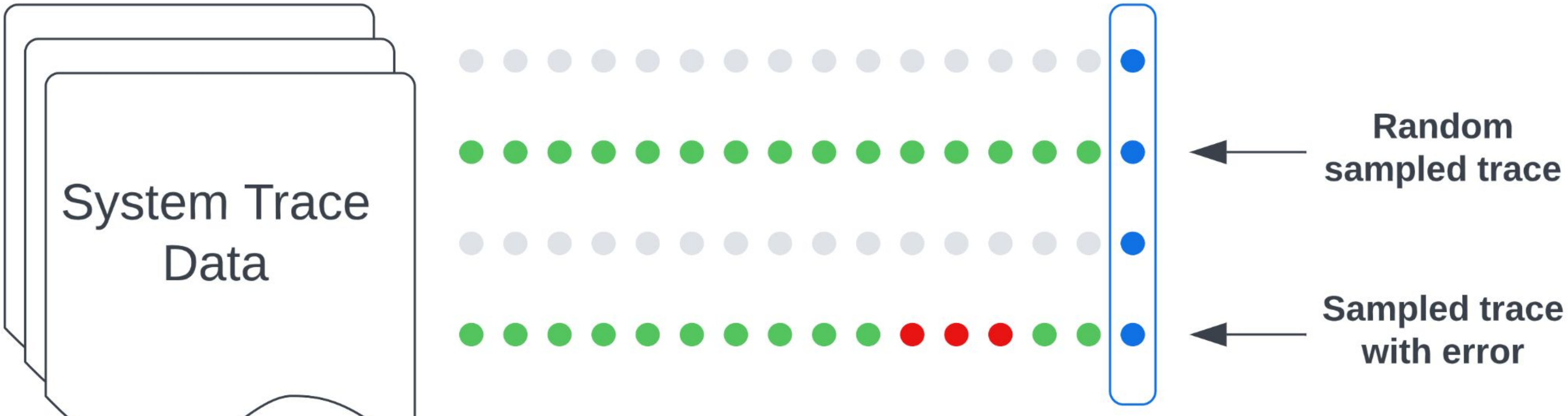


```
processors:
  tail_sampling:
    decision_wait: 10s
    num_traces: 100
    expected_new_traces_per_sec: 10
    policies:
      [
        {
          name: errors-policy,
          type: status_code,
          status_code: {status_codes: [ERROR]}
        },
        {
          name: randomized-policy,
          type: probabilistic,
          probabilistic: {sampling_percentage: 25}
        },
      ]
```

Tail-based sampler

KubeCon CloudNativeCon Europe 2022

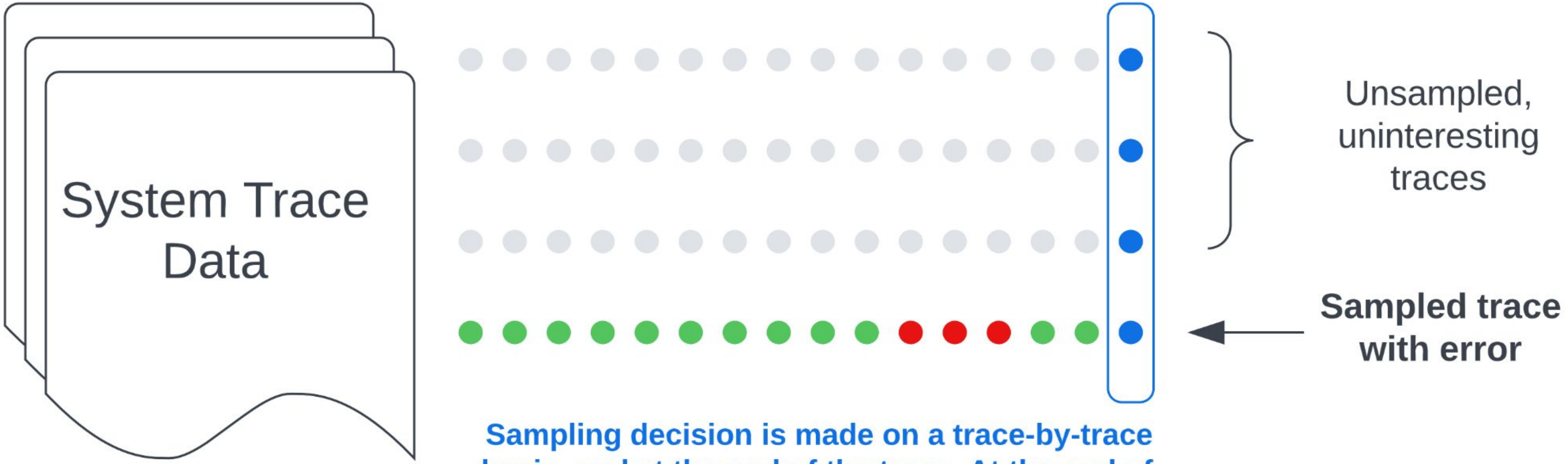
Errors + a random sampling of all other traces



Sampling decision is made on a trace-by-trace basis, and at the end of the trace. At the end of a trace, sampler makes a sampling decision based off created policies.

Tail-based sampler

Errors only



Sampling decision is made on a trace-by-trace basis, and at the end of the trace. At the end of a trace, sampler makes a sampling decision based off created policies.

SAMPLING IN ACTION!

- Demo context
- Sampling demos
- Demo reflection

Demo context

- Three services that always call each other in order
- Load generator - 20 calls - 1 error each time
- Traces of interest - traces with errors
- Jaeger
- **Which sampling strategy is optimal for getting me what I want?**

Sampling scenarios

Head-based sampling scenarios

1. ParentBased(root=AlwaysOn)
2. ParentBased(root=TraceIdRatioBased)

Tail-based sampling scenarios

1. `status_code` policy
2. `status_code` + `probabilistic` policies

Demo reflection

Demo reflection

Head-based sampling scenarios

1. **ParentBased(root=AlwaysOn)**
    - Always saw our traces of interest
    - But also saw everything

2. **ParentBased(root=TraceIdRatioBased)**
    - A random sampling
    - Didn’t always see our traces of interest

Demo reflection

Tail-based sampling scenarios

1. `status_code` policy
   - Only errors!

2. `status_code` + `probabilistic` policies
   - Errors and a random sampling of other traces

CONCERNS & LIMITATIONS

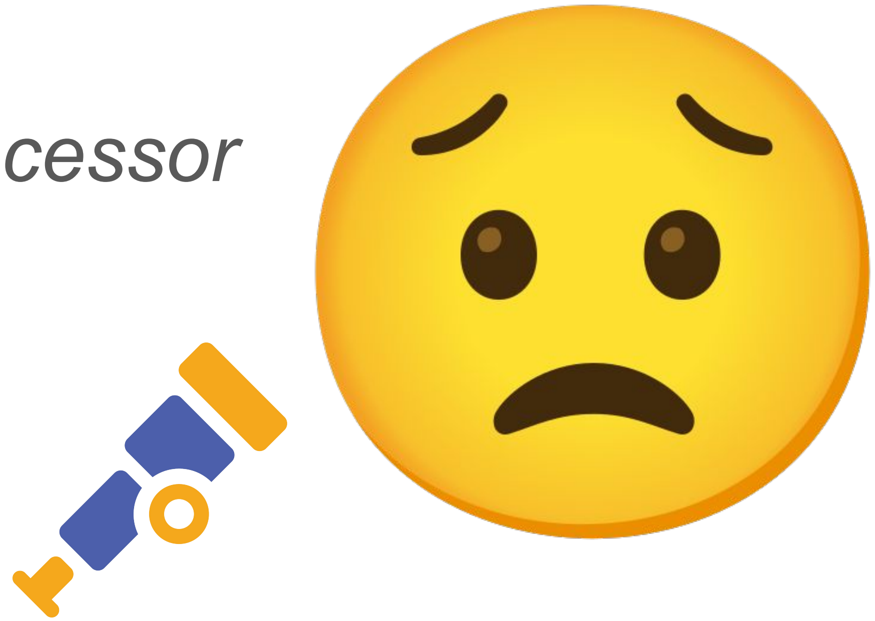
- General concerns
- OpenTelemetry limitations

General concerns

- Performance consideration
- Determining the interesting traces
- Data egress and storage costs

OpenTelemetry limitations

- Have to stand up collector
- Scalability
  - Collector deployment pattern
  - Load balancing
- The future of the *tailsamplingprocessor*



CLOSING

- Summary
- Future
  - Log sampling
  - Contribute!

Credits

Thank you!

- Sharr Creeden, New Relic
- Alan West, New Relic
- Chris Ventura, New Relic
- Tyler Helmuth, New Relic
- Martin Kuba, New Relic
- Svetlana Brennan, New Relic
- Jack Berg, New Relic
- Juraci Paixão Khröhling, Grafana
- Richard Bannin, The Confident Speaker
- Everyone who took the time to watch me practice and share feedback!

Q&A

What can I answer for you?

Resources

- CNCF Slack
- OpenTelemetry public Google calendar
- OpenTelemetry docs
- OpenTelemetry booth #23

Scan here to share feedback with the OpenTelemetry Community about your experience with OpenTelemetry!

Connect with me!

- Twitter: @reesesbytes
- LinkedIn: Reese Lee

GRACIAS!