



KubeCon



CloudNativeCon

North America 2024

Contribfest: Kickstart Your eBPF Journey with Tetragon

Joe Stringer, Mahé Tardy, Kornilios Kourtis, John Fastabend

1. Architecture overview
2. Core concepts
 - a. Events
 - b. TracingPolicy
3. Internals
 - a. Kernel/User split
 - b. Policy handling
 - c. Sensors
 - d. Life of a TracingPolicy
4. How to contribute





KubeCon

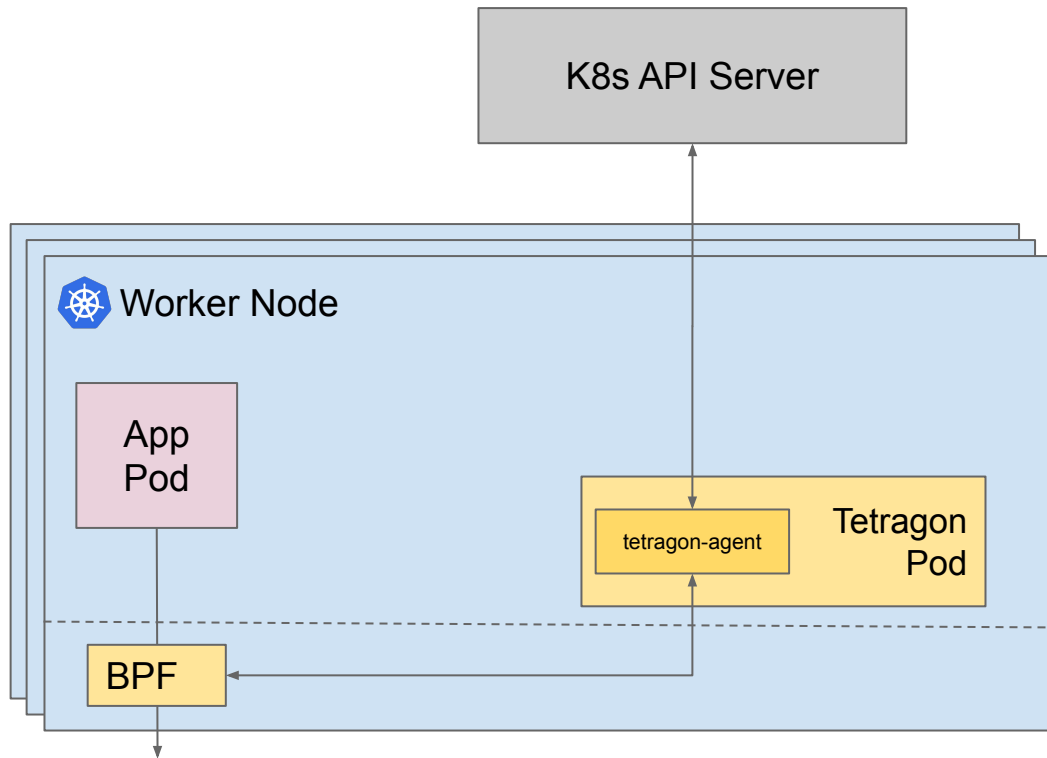


CloudNativeCon

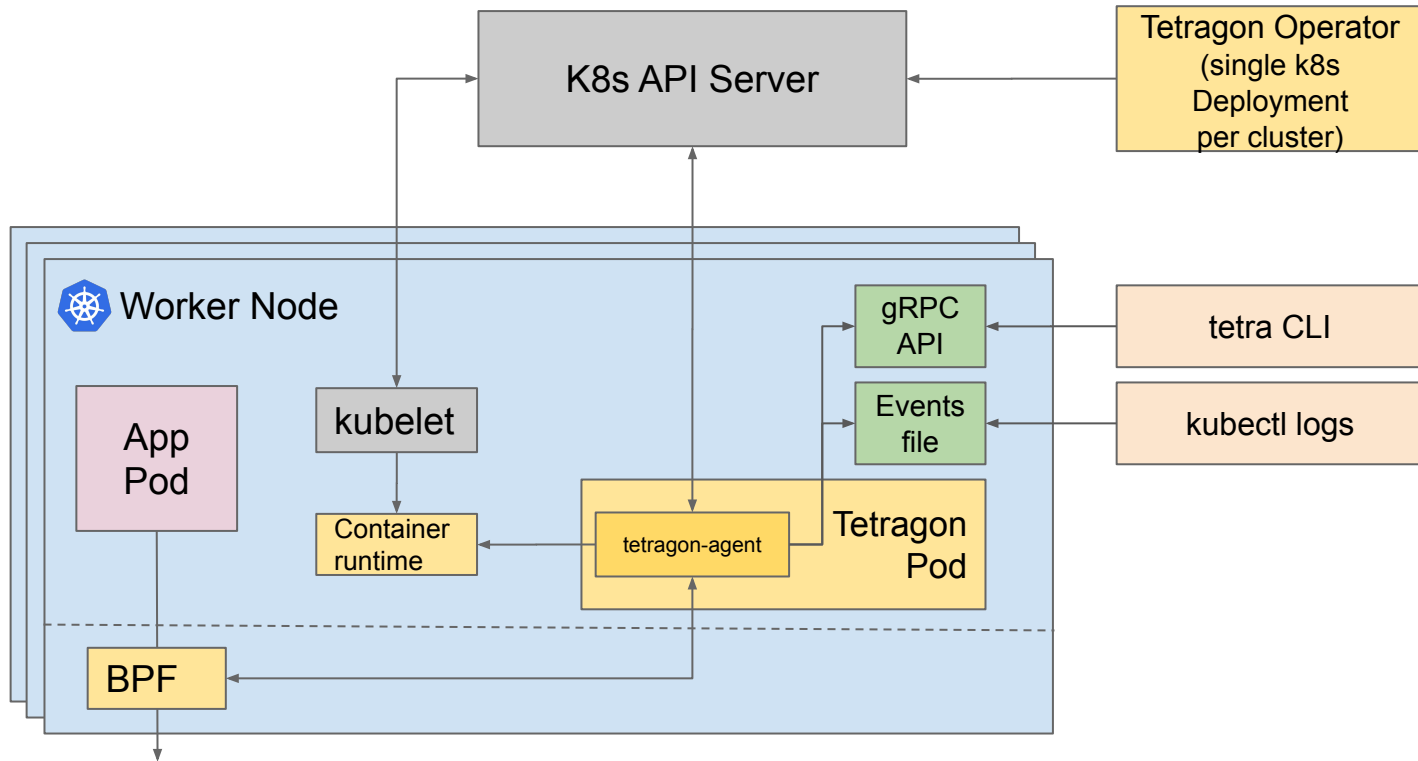
North America 2024

Architecture Overview

Architecture - High-level View



Architecture - More Details





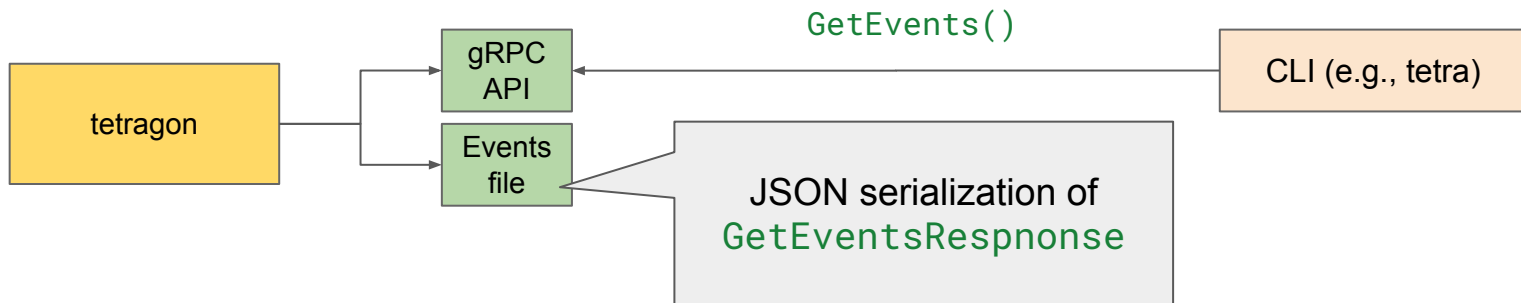
KubeCon



CloudNativeCon

North America 2024

Core Concepts



`rpc GetEvents(GetEventsRequest) returns (stream GetEventsResponse) {}`

Configure event stream,
e.g.,. Filters (allow, deny,
fields)

Event:

- event
- node_name
- cluster_name
- time



```
oneof event {
```

```
    ProcessExec process_exec = 1;  
    ProcessExit process_exit = 5;
```

Base events

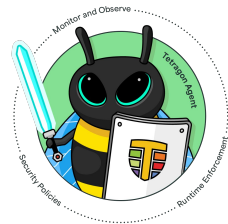
```
    ProcessKprobe process_kprobe = 9;  
    ProcessTracepoint process_tracepoint = 10;  
    ProcessLoader process_loader = 11;  
    ProcessUprobe process_uprobe = 12;  
    ProcessThrottle process_throttle = 27;  
    ProcessLsm process_lsm = 28;
```

Configurable
events

```
}
```



```
message ProcessExec {  
    Process process = 1;  
    Process parent = 2;  
}  
  
Message Process {  
    string exec_id = 1;  
    ...  
    string binary = 5;  
    string arguments = 6;  
    ...  
    Pod pod = 10;  
    ...  
}
```



Exec event



KubeCon

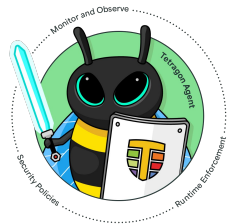


CloudNativeCon

North America 2024

```
{
  "process_exec": {
    "process": {
      "binary": "/usr/bin/find",
      "arguments": "-name pizza",
      "pod": {
        "namespace": "default",
        "name": "test",
        "container": {
          "id": "docker://acddc67175a09036565f34b796b191f68dc770cad50197fc43b23fc354254502",
          "name": "test",
          "image": {
            "id": "docker-pullable://debian@sha256:27586f4609433f2f49a9157405b473c62c3cb28a581c413393975b4e8496d0ab",
            "name": "debian:latest"
          },
          "start_time": "2024-10-02T10:41:14Z",
        },
        "pod_labels": {
          "run": "test"
        },
        "workload": "test",
        "workload_kind": "Pod"
      }
    },
    "node_name": "minikube",
    "time": "2024-10-02T10:42:19.860322547Z"
  }
}
```





```
message ProcessExit {  
    Process process = 1;  
    Process parent  = 2;  
    string  signal  = 3;  
    uint32  status  = 4;  
}
```

All events have **.process**
and **.parent**

.process.exec_id can be
used to match events of the
same process

```
oneof event {  
    ProcessExec process_exec = 1;  
    ProcessExit process_exit = 5;
```

```
    ProcessKprobe process_kprobe = 9;  
    ProcessTracepoint process_tracepoint = 10;  
    ProcessLoader process_loader = 11;  
    ProcessUprobe process_uprobe = 12;  
    ProcessThrottle process_throttle = 27;  
    ProcessLsm process_lsm = 28;
```

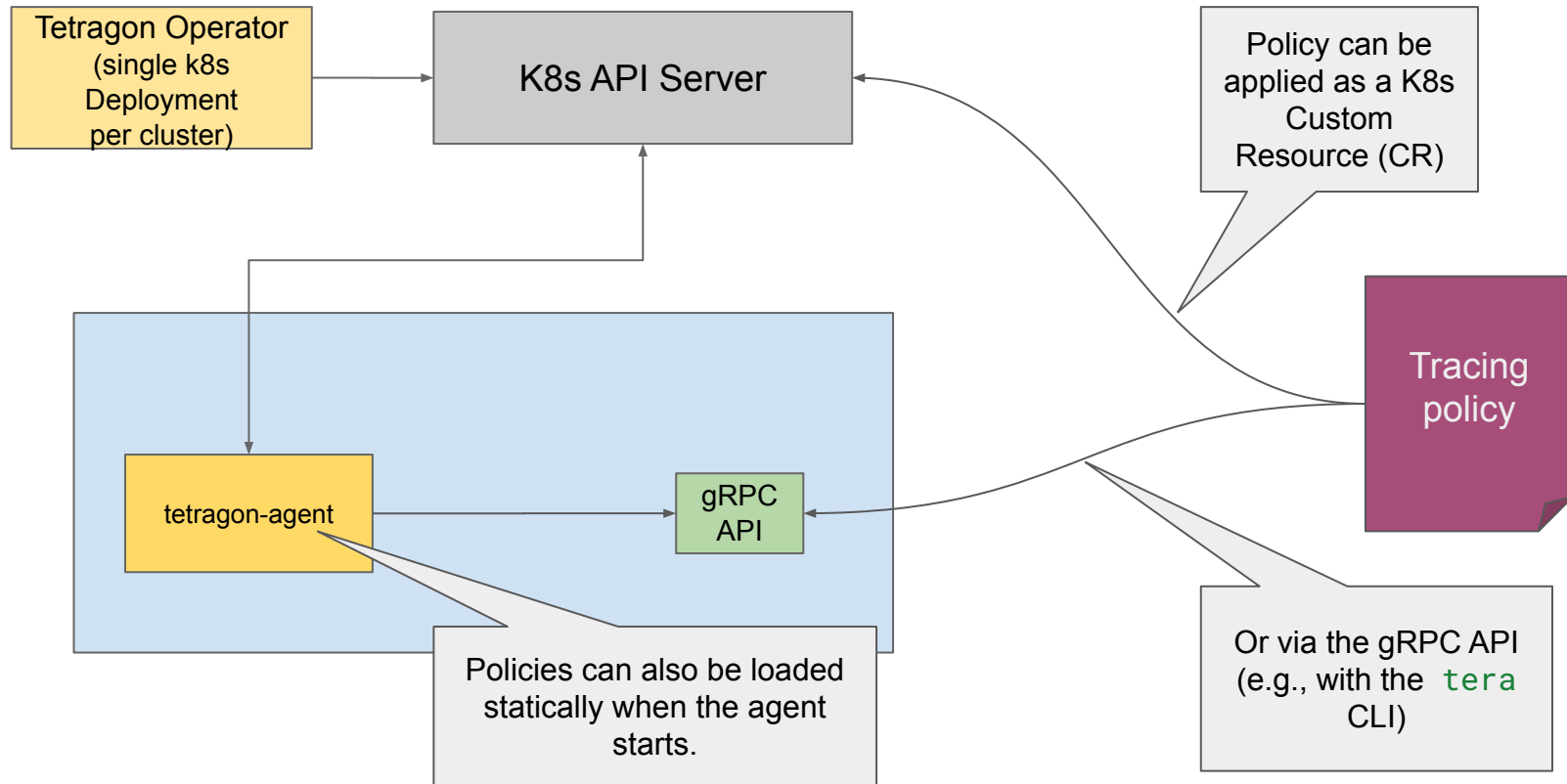
```
}
```

Tetragon can be configured
via **TracingPolicies** to
generate events beyond
exec/exit

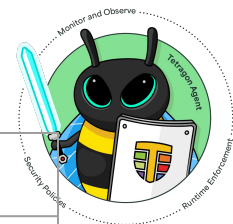
Configurable
events



Tracing Policies



Policies and events



Policy section	Event type	
<code>spec.kprobes</code>	<code>ProcessKprobe</code>	generic kprobe hooks
<code>spec.tracepoints</code>	<code>ProcessTracepoint</code>	generic tracepoint hooks
<code>spec.uprobes</code>	<code>ProcessUprobe</code>	generic uprobe hook
<code>spec.lsmhooks</code>	<code>ProcessLsm</code>	generic LSM hook
<code>spec.loader</code>	<code>ProcessLoader</code>	Shared library / binary loading



KubeCon

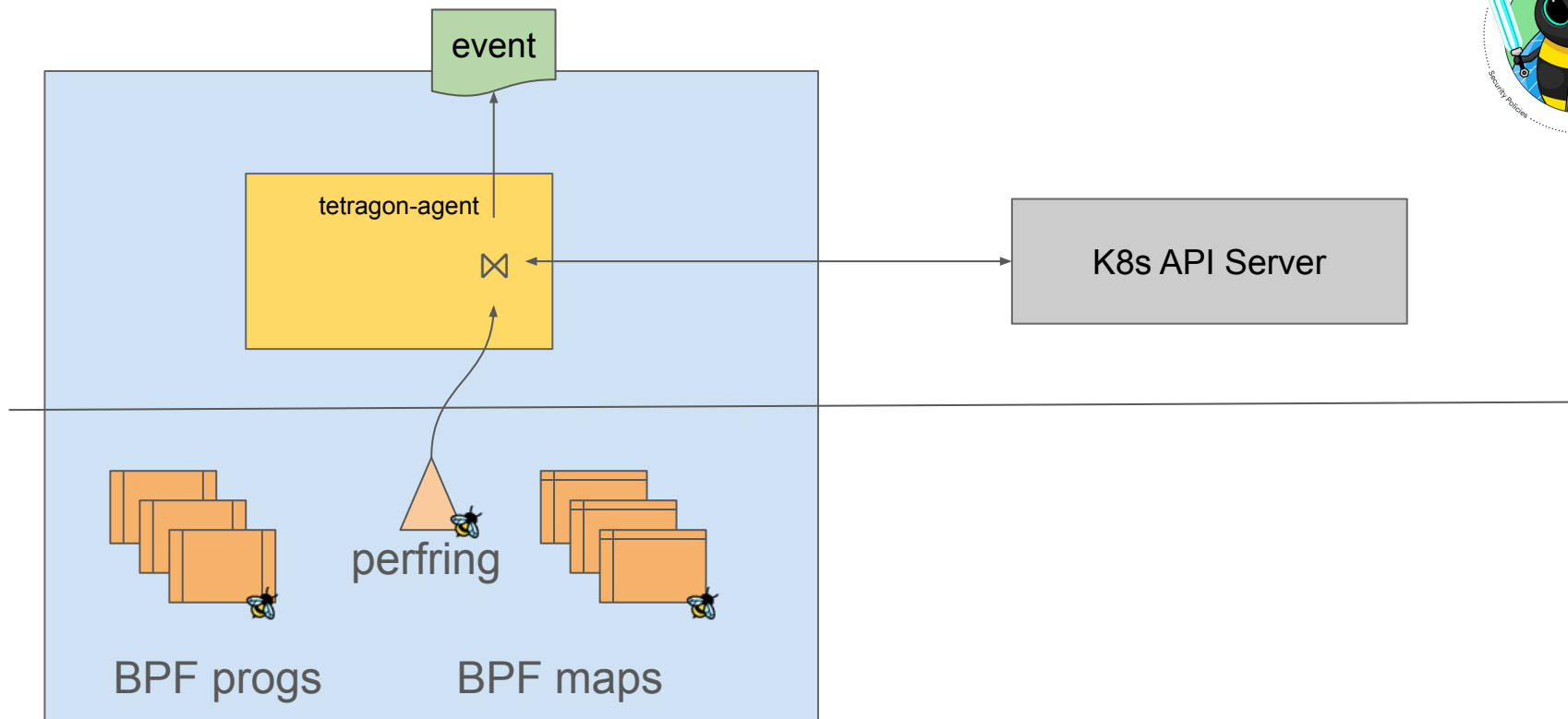


CloudNativeCon

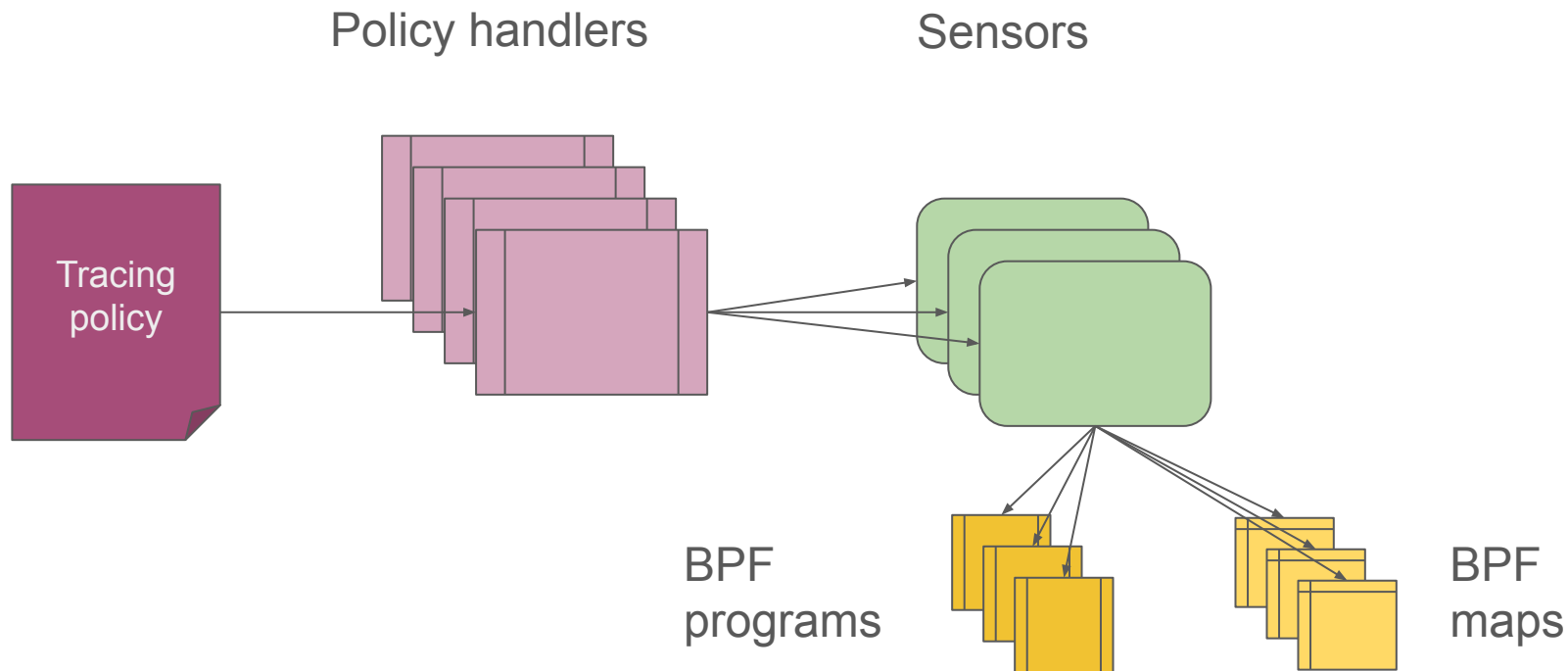
North America 2024

Internals

Kernel/user split



Policy Handling

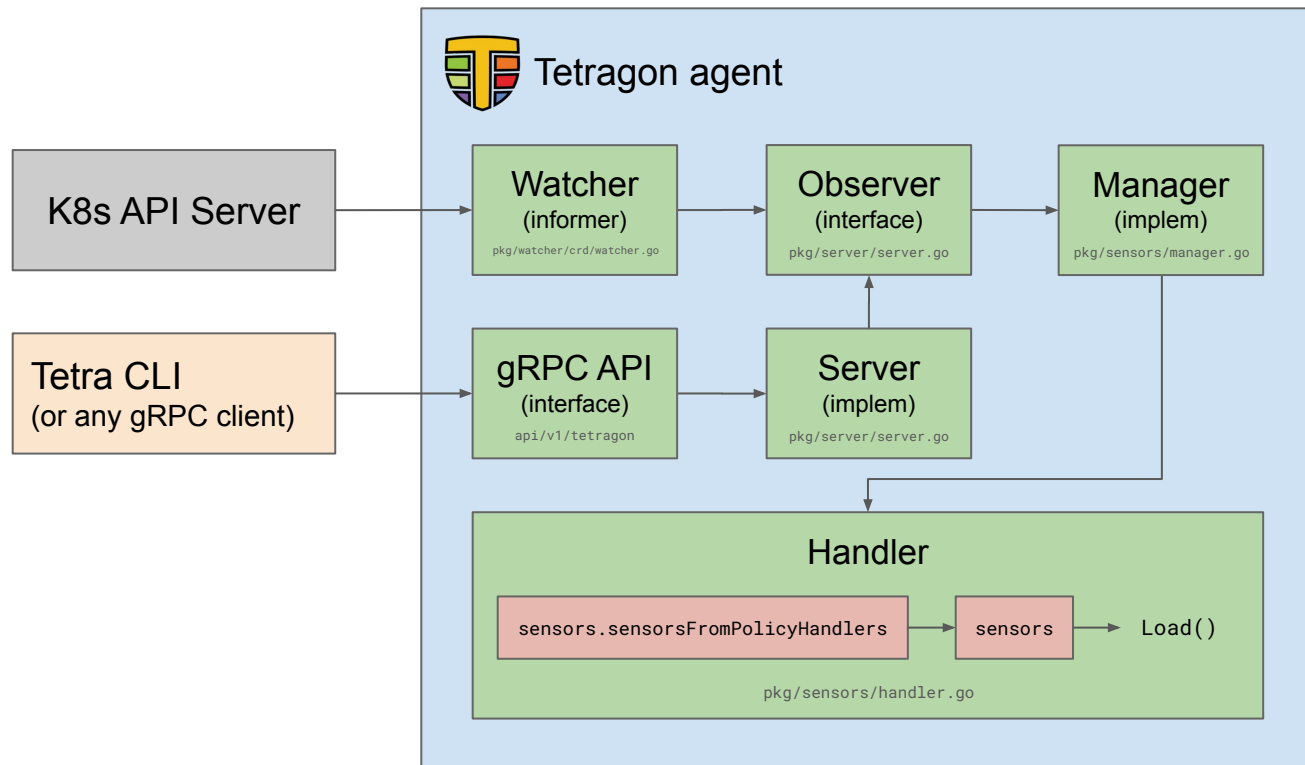




A sensor is a collection of BPF programs and map that is managed as a single unit

- The “exec” (or base) sensor is always loaded
- Other sensors are loaded dynamically by tracing policies

Life of a TracingPolicy





KubeCon



CloudNativeCon

North America 2024





KubeCon

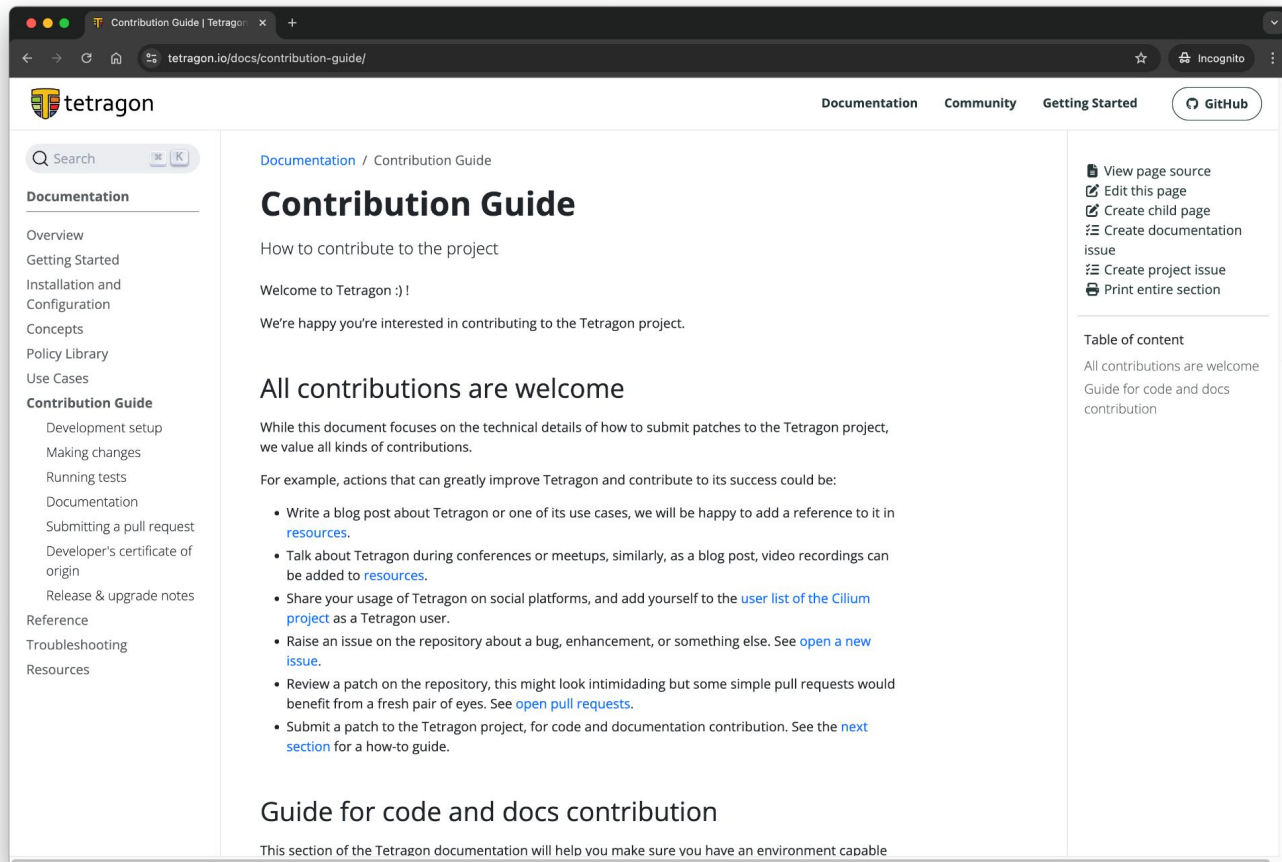


CloudNativeCon

North America 2024

Let's contribute!

Contribution Guide



The screenshot shows a web browser displaying the Tetragon Contribution Guide. The browser's address bar shows the URL `tetragon.io/docs/contribution-guide/`. The page has a dark header with the Tetragon logo and navigation links for Documentation, Community, Getting Started, and a GitHub button. A left sidebar contains a search bar and a table of contents with links to Overview, Getting Started, Installation and Configuration, Concepts, Policy Library, Use Cases, Contribution Guide, Reference, Troubleshooting, and Resources. The main content area is titled "Contribution Guide" and includes a sub-header "How to contribute to the project". It welcomes contributors and lists ways to contribute, such as writing blog posts, talking at conferences, sharing on social media, raising issues, reviewing patches, and submitting pull requests. A right sidebar offers links to view page source, edit the page, create child pages, create documentation issues, create project issues, and print the section. A "Table of content" section also lists links for code and docs contributions. The page footer states that the section will help users set up a capable environment.

Contribution Guide | Tetragon

tetragon.io/docs/contribution-guide/

tetragon

Documentation Community Getting Started GitHub

Search

Documentation

- Overview
- Getting Started
- Installation and Configuration
- Concepts
- Policy Library
- Use Cases
- Contribution Guide
 - Development setup
 - Making changes
 - Running tests
 - Documentation
 - Submitting a pull request
 - Developer's certificate of origin
 - Release & upgrade notes
- Reference
- Troubleshooting
- Resources

Documentation / Contribution Guide

Contribution Guide

How to contribute to the project

Welcome to Tetragon :)!

We're happy you're interested in contributing to the Tetragon project.

All contributions are welcome

While this document focuses on the technical details of how to submit patches to the Tetragon project, we value all kinds of contributions.

For example, actions that can greatly improve Tetragon and contribute to its success could be:

- Write a blog post about Tetragon or one of its use cases, we will be happy to add a reference to it in [resources](#).
- Talk about Tetragon during conferences or meetups, similarly, as a blog post, video recordings can be added to [resources](#).
- Share your usage of Tetragon on social platforms, and add yourself to the [user list of the Cilium project](#) as a Tetragon user.
- Raise an issue on the repository about a bug, enhancement, or something else. See [open a new issue](#).
- Review a patch on the repository, this might look intimidating but some simple pull requests would benefit from a fresh pair of eyes. See [open pull requests](#).
- Submit a patch to the Tetragon project, for code and documentation contribution. See the [next section](#) for a how-to guide.

Guide for code and docs contribution

This section of the Tetragon documentation will help you make sure you have an environment capable

View page source
Edit this page
Create child page
Create documentation issue
Create project issue
Print entire section

Table of content

All contributions are welcome
Guide for code and docs contribution



The contribution documentation provides many guides:

- Development setup
- Making changes and running the codegen
- Running the tests
- Modify the documentation
- Submit your pull request containing your signed patch
- Write release and upgrade notes



Good first issues

