# System state clustering using eBPF data

# Contents

WHAT IS EBPF

WHAT IS CLUSTERING

EBPF + CLUSTERING

EBPF AND SRE

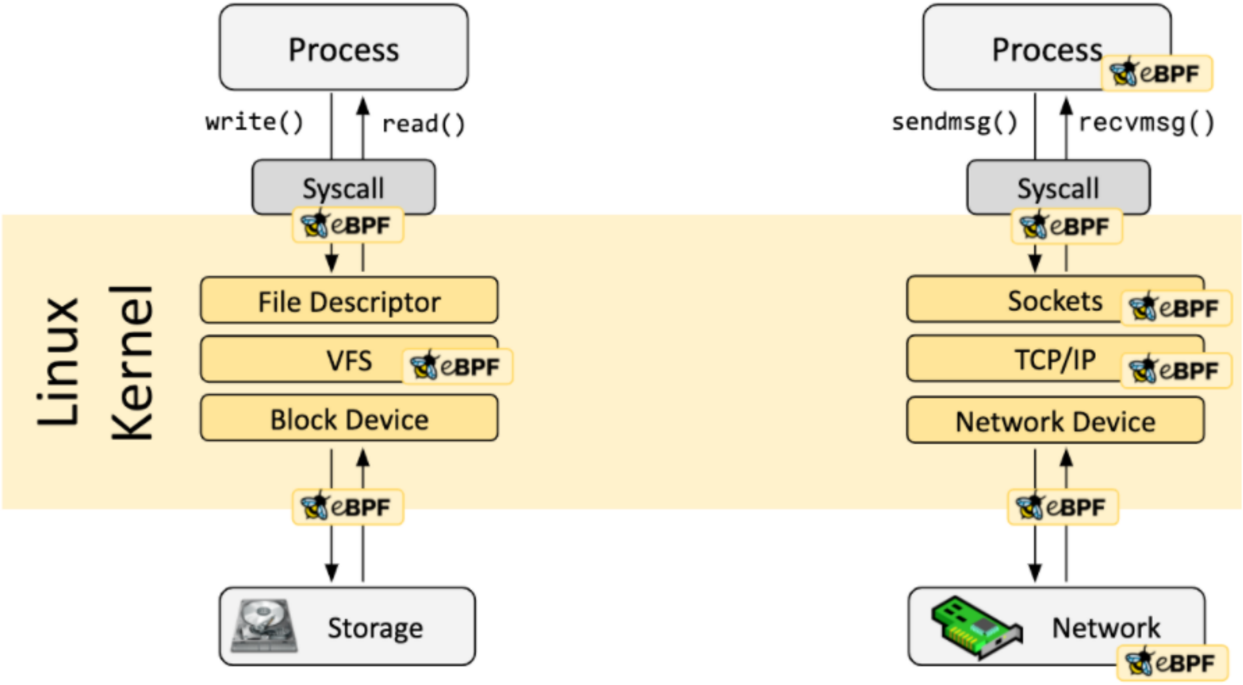
POTENTIAL USE-CASES

Q&A

# eBPF – Extended Berkeley Packer Filter

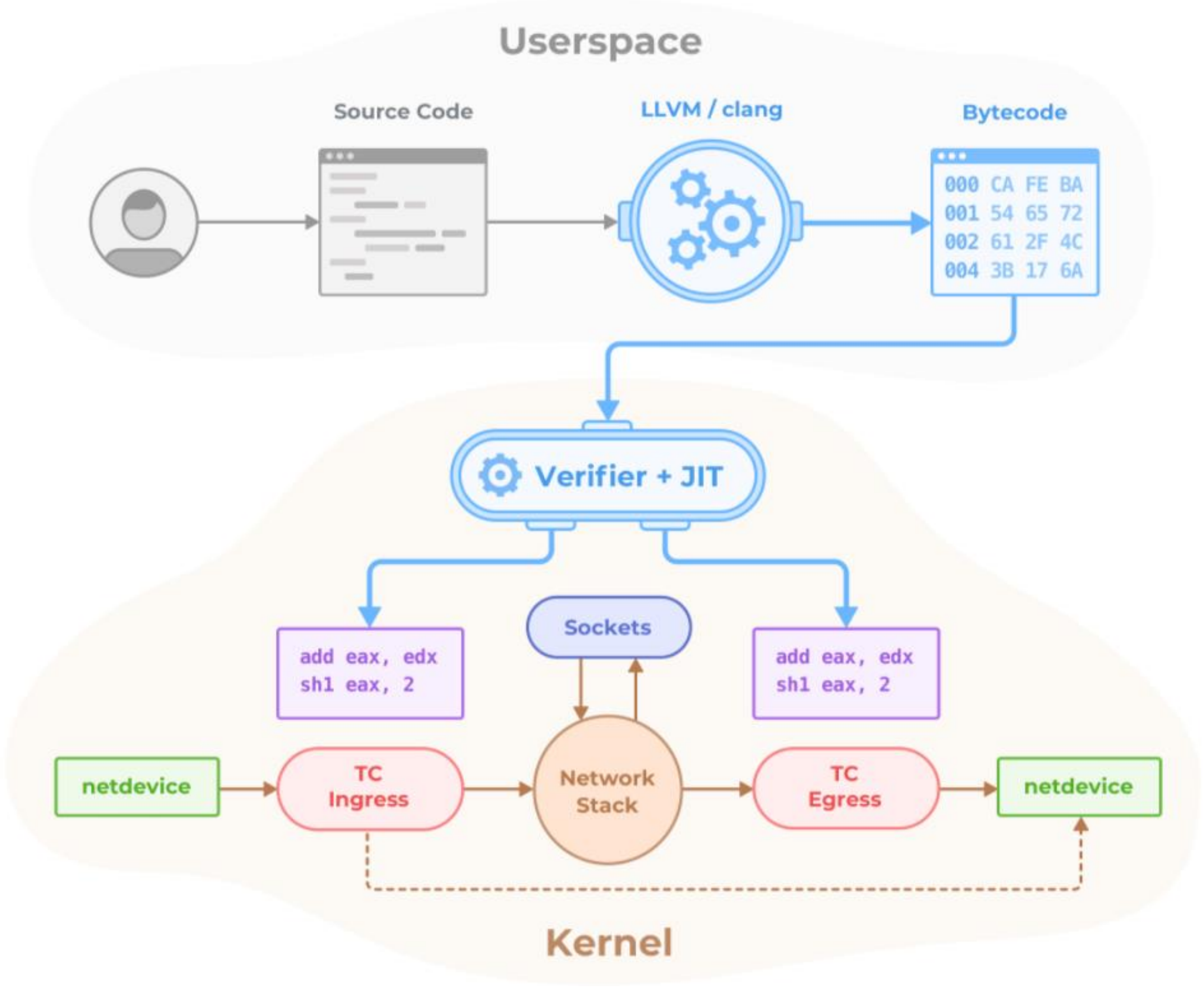
- eBPF programs are event-driven and are run when the kernel or an application passes a certain hook point
- Pre-defined hooks include system calls, function entry/exit, kernel tracepoints, network events, and several others.
- From there we can run our own programs which can pass data back to user space via BPF maps

# eBPF – Extended Berkeley Packer Filter



* Credits https://ebpf.io

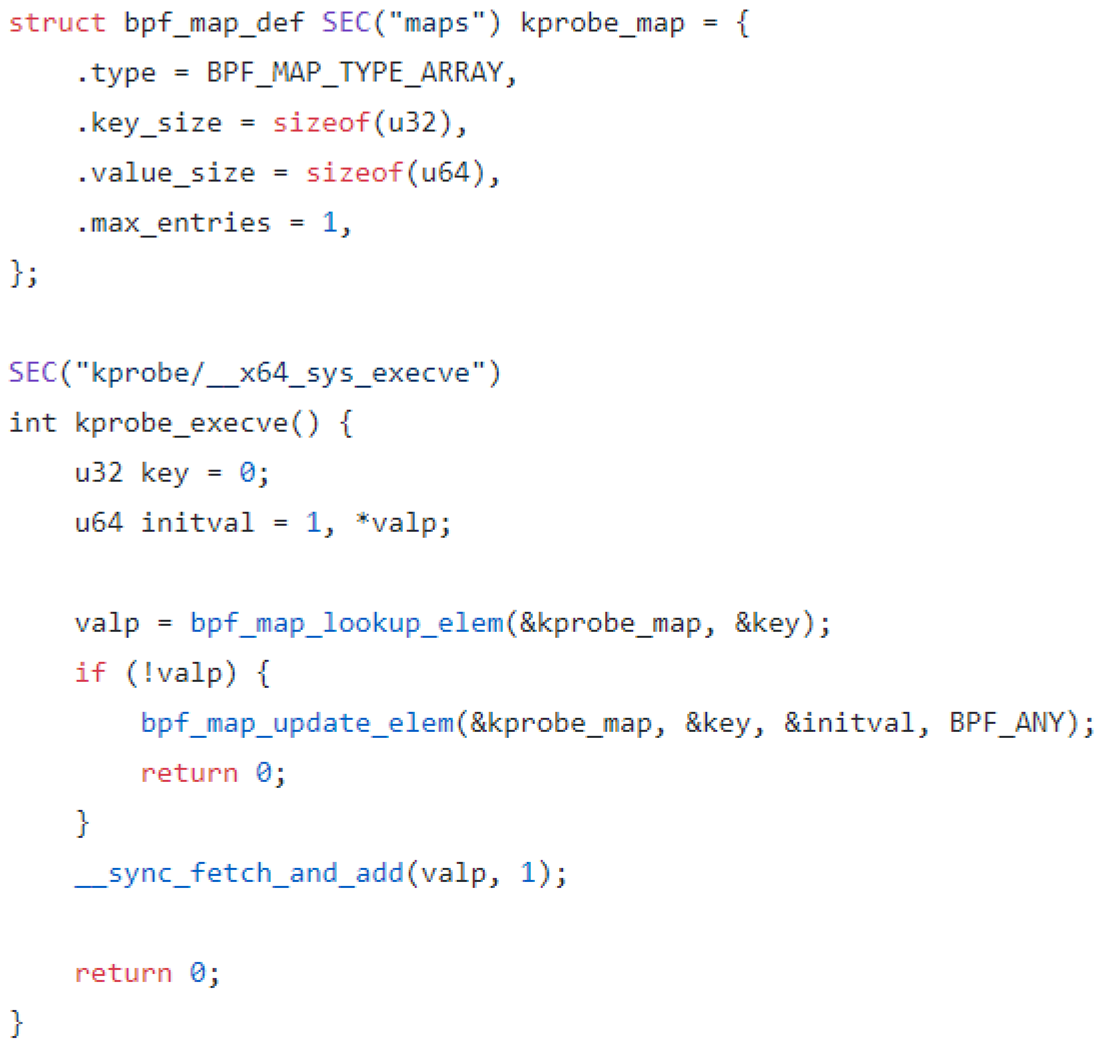
# eBPF – Extended Berkeley Packer Filter



* Credits https://github.com/cilium/cilium

# eBPF – Extended Berkeley Packer Filter

How many times is a kernel function entered

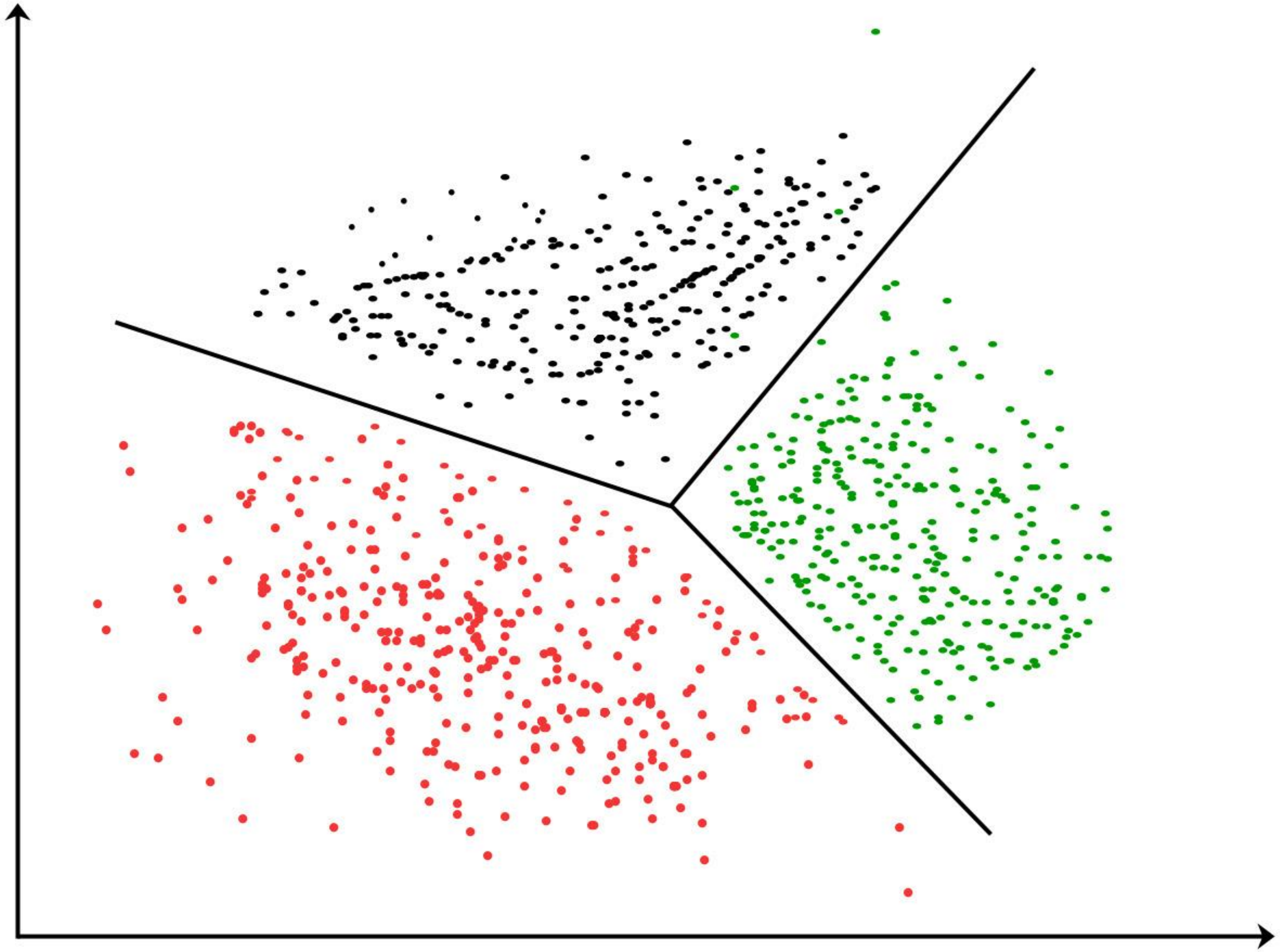


```
struct bpf_map_def SEC("maps") kprobe_map = {
    .type = BPF_MAP_TYPE_ARRAY,
    .key_size = sizeof(u32),
    .value_size = sizeof(u64),
    .max_entries = 1,
};

SEC("kprobe/__x64_sys_execve")
int kprobe_execve() {
    u32 key = 0;
    u64 initval = 1, *valp;

    valp = bpf_map_lookup_elem(&kprobe_map, &key);
    if (!valp) {
        bpf_map_update_elem(&kprobe_map, &key, &initval, BPF_ANY);
        return 0;
    }
    __sync_fetch_and_add(valp, 1);

    return 0;
}
```

* Credits https://github.com/cilium/ebpf

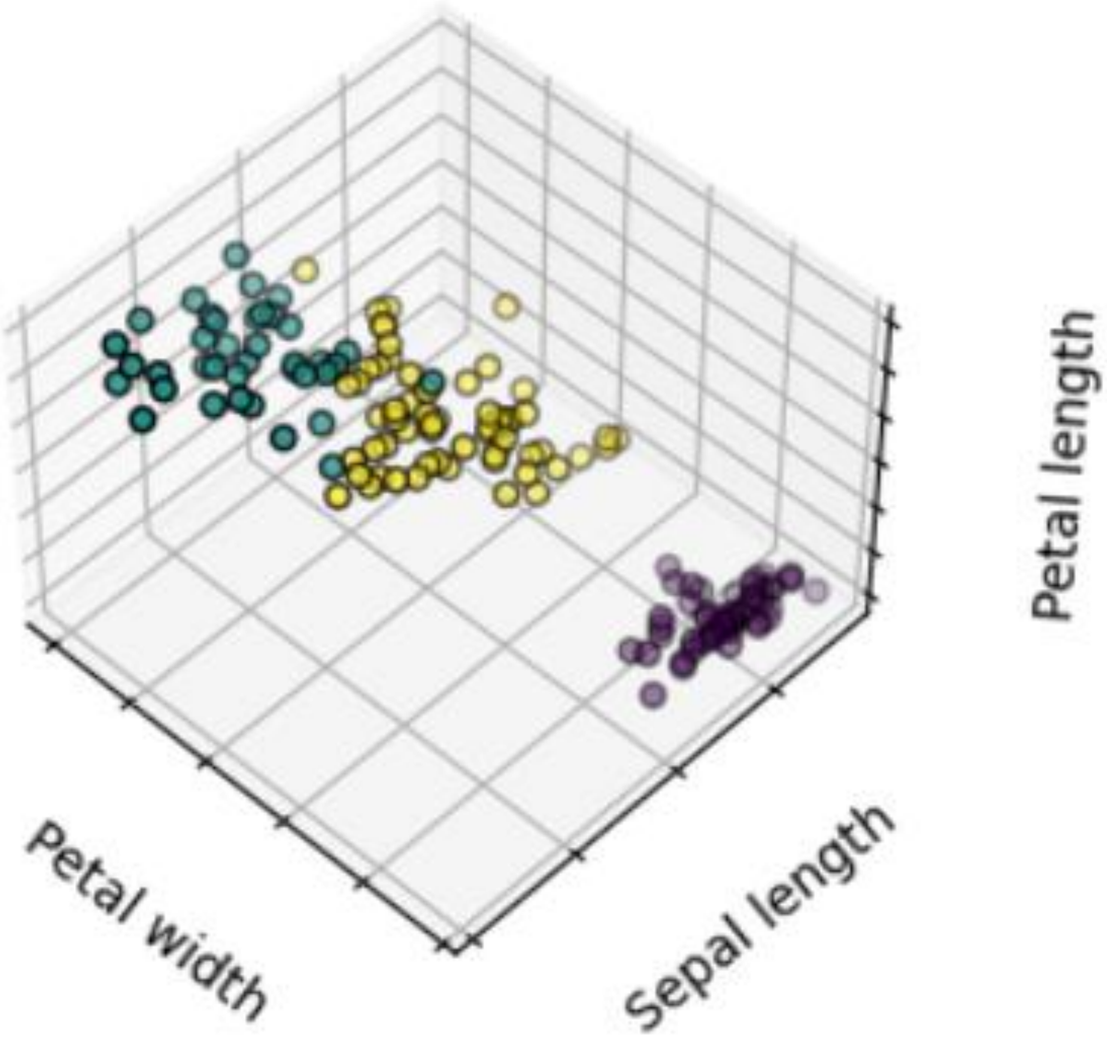
# Clustering

- Machine Learning algorithm which can help cluster different data points into classes
- Data points which are similar tend to be closer together when represented dimensionally



* Credits geekforgeeks

# Clustering – Iris Example



* Credits scikit-learn.org

# eBPF + Clustering

- eBPF generates data points
- Clustering can cluster the data points

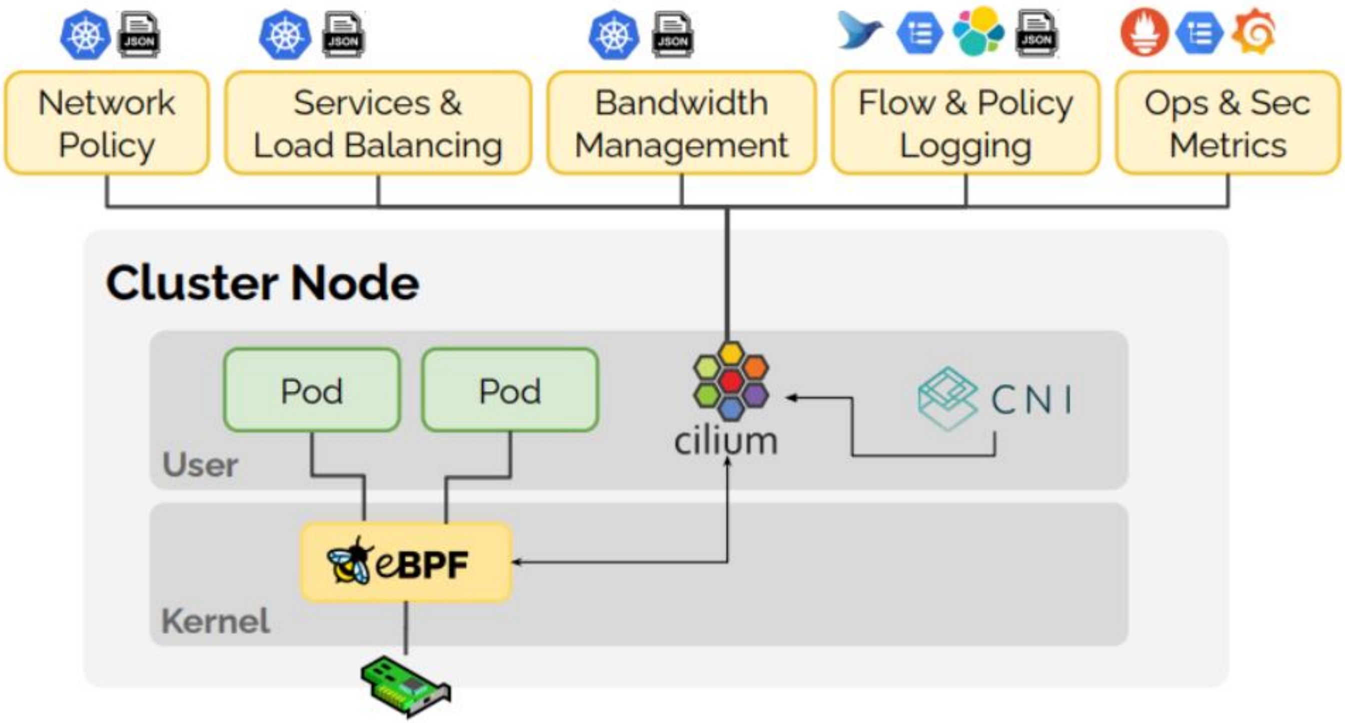
- Data labelling can initially label positive and negative classes
- Production scenario can use that model to predict the real time data point

# eBPF + Clustering

- eBPF XDP/Socket Filter programs generate data about the received packets over the network
- Sandboxed program writes the packet details like Input IP, bytes etc into a map
- User space program reads map and pushes to data store e.g Redis
- Clustering algorithm to understand if request is normal
- Timeseries algorithm to understand if this could be leading to series of not normal requests

# eBPF and SRE

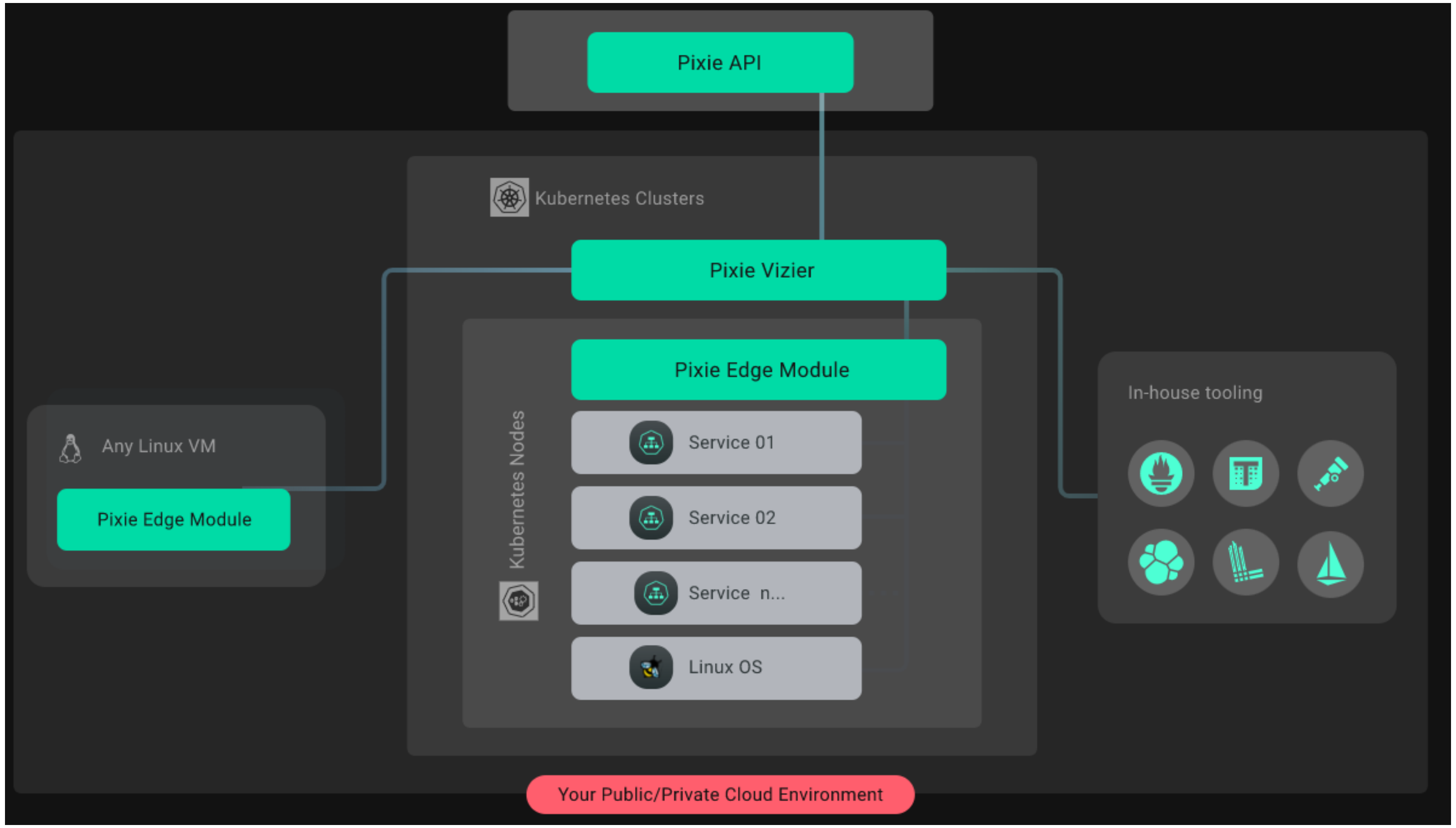
- Cilium
  - Provides the right level of information for troubleshooting application and connectivity issues
  - All of this is available via HUBBLE framework – API, CLI and GUI



* Credits https://github.com/cilium/cilium

# eBPF and SRE

- Pixie
  - Add dynamic eBPF probes to provide access to metrics, events, traces and logs
  - Pixie scripts for debugging
  - Pixie CLI and live UI



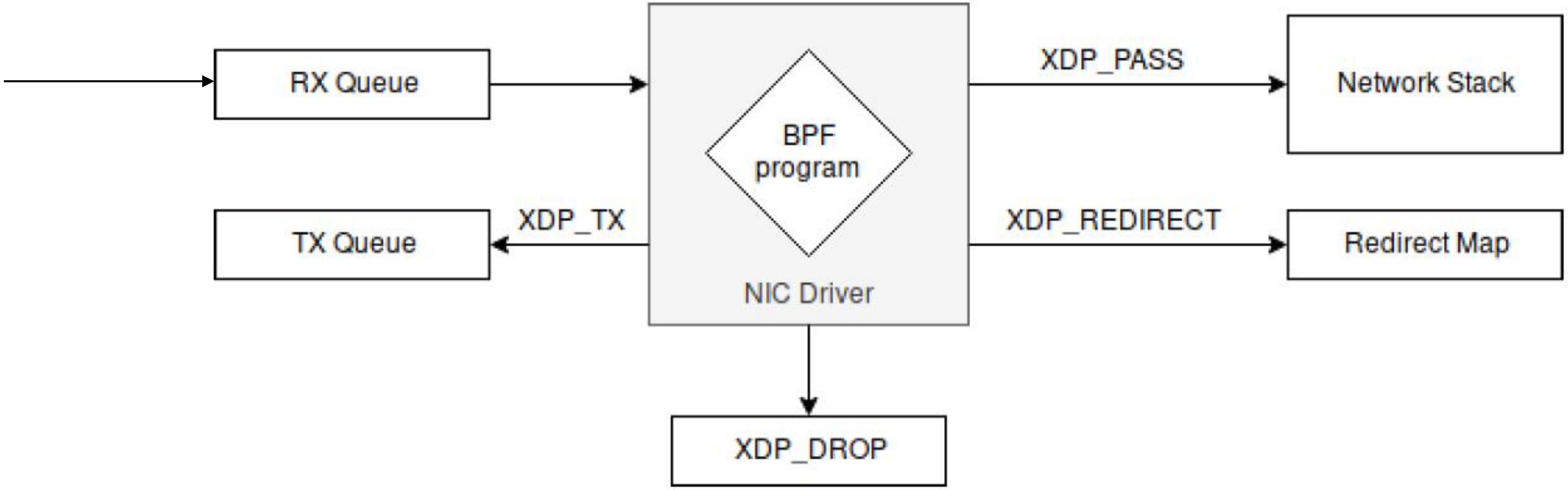
* Credits https://px.dev/

# Use-cases

- System performance degradation check
- Network traffic check
- Preventive maintenance

# Network Traffic use case

- Incoming request
- Interception by XDP BPF program
- Data written to a MAP along with details like incoming address, time, etc
- User space program reads map and sends data to clustering system to ascertain whether it belongs to a valid cluster



* Credits David Calavera_ Lorenzo Fontana - Linux Observability with BPF_ Advanced Programming for Performance Analysis and Networking-_O'Reilly Media, Inc._ (2019)

# Q & A