The Road to Zeek 9

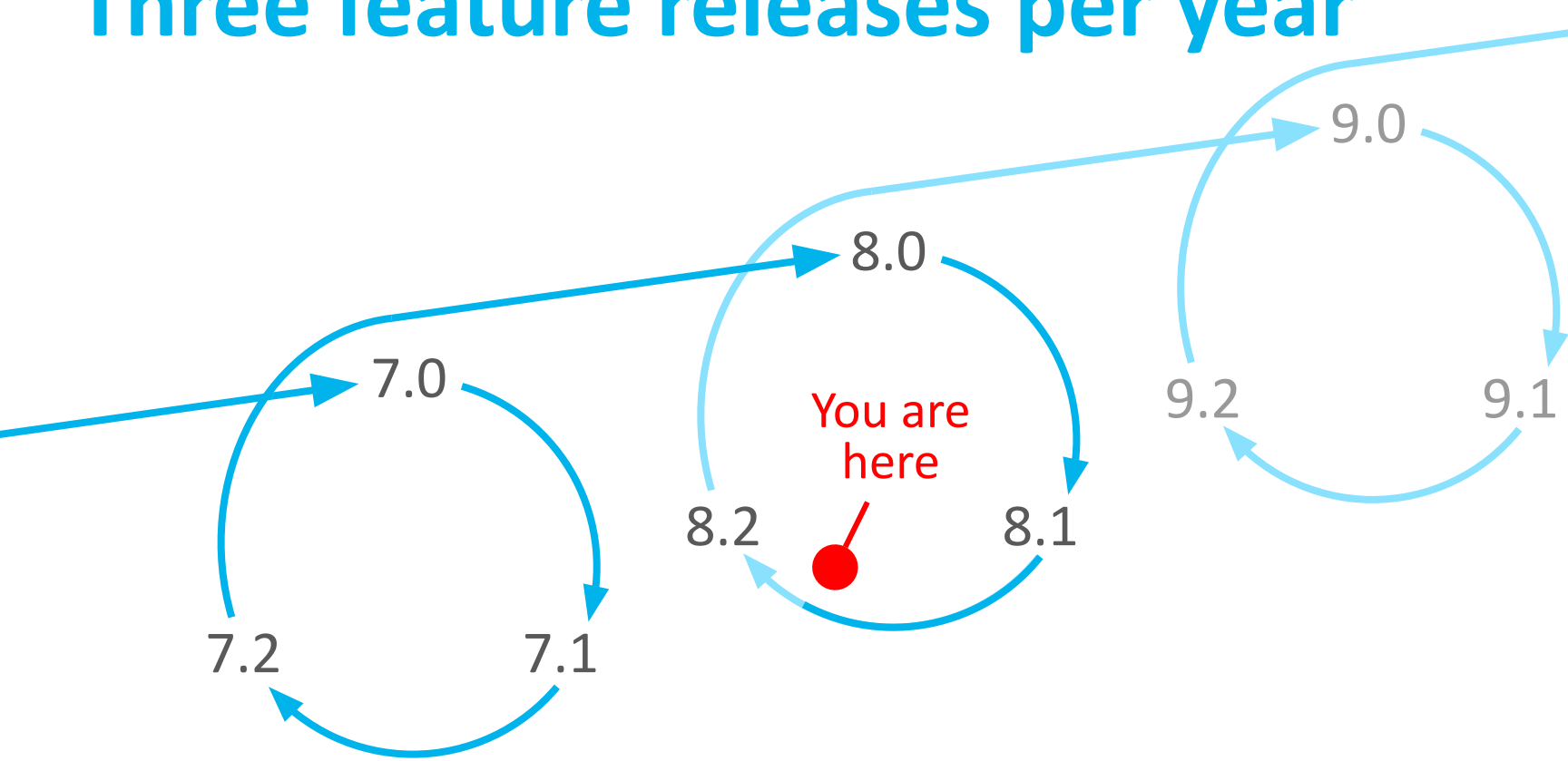
A Zeek Development Update

Christian Kreibich

christian@zeek.org

Release cadence

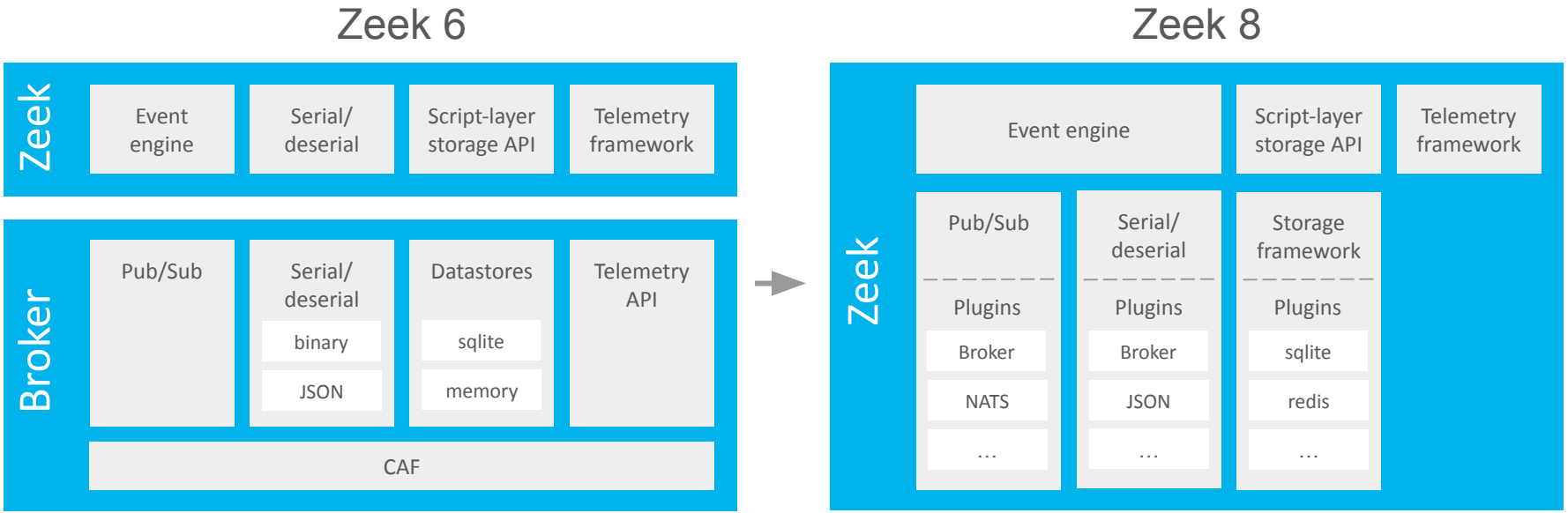
Three feature releases per year



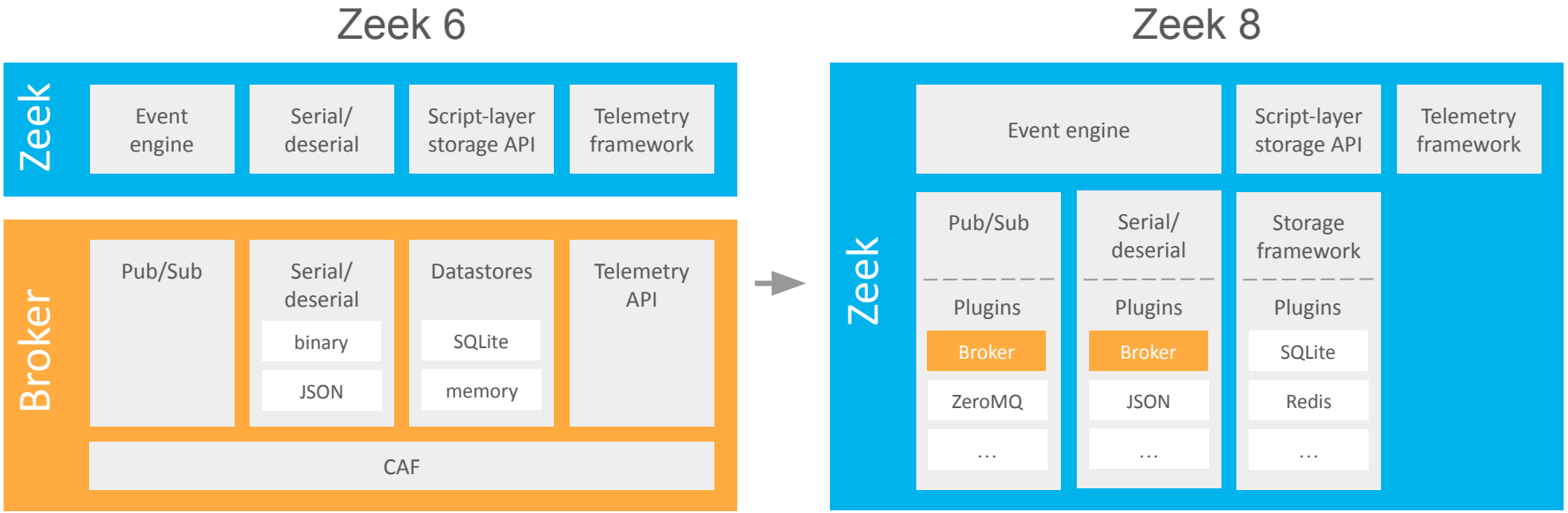
https://github.com/zeek/zeek/wiki/Release-Cadence

Zeek 8 and 8.1

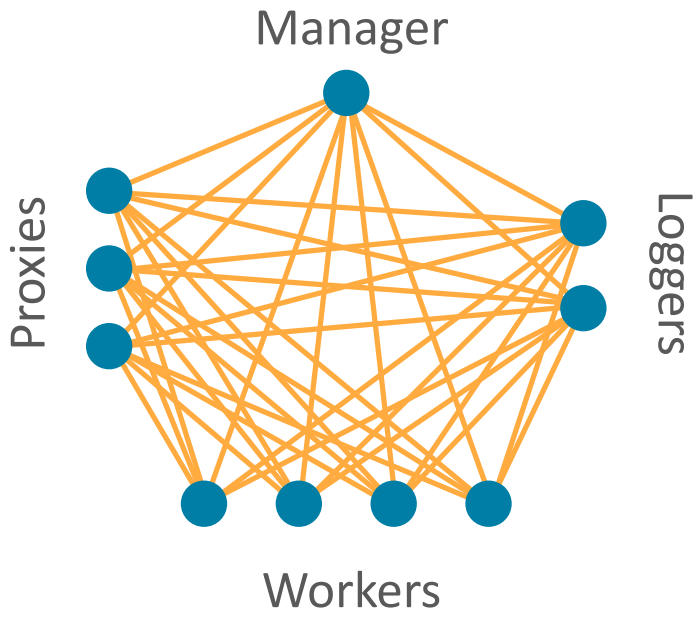
Zeek 6 -> Zeek 8 architectural revamp



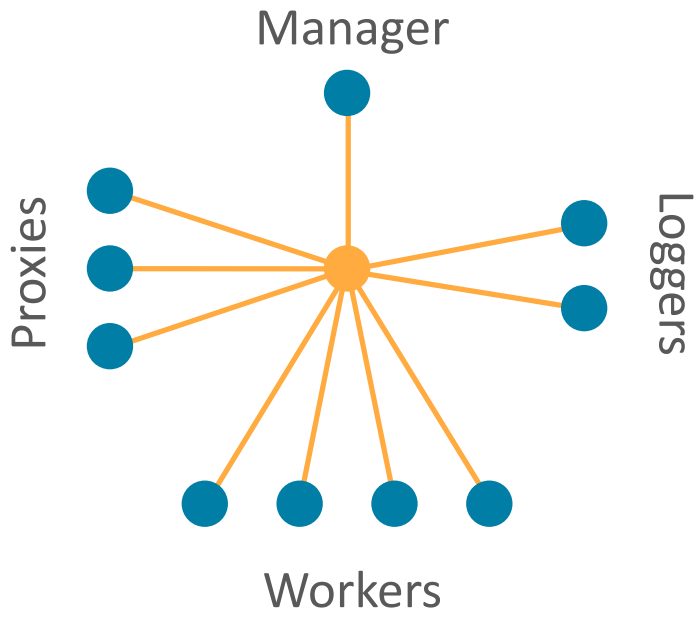
Zeek 6 -> Zeek 8 architectural revamp



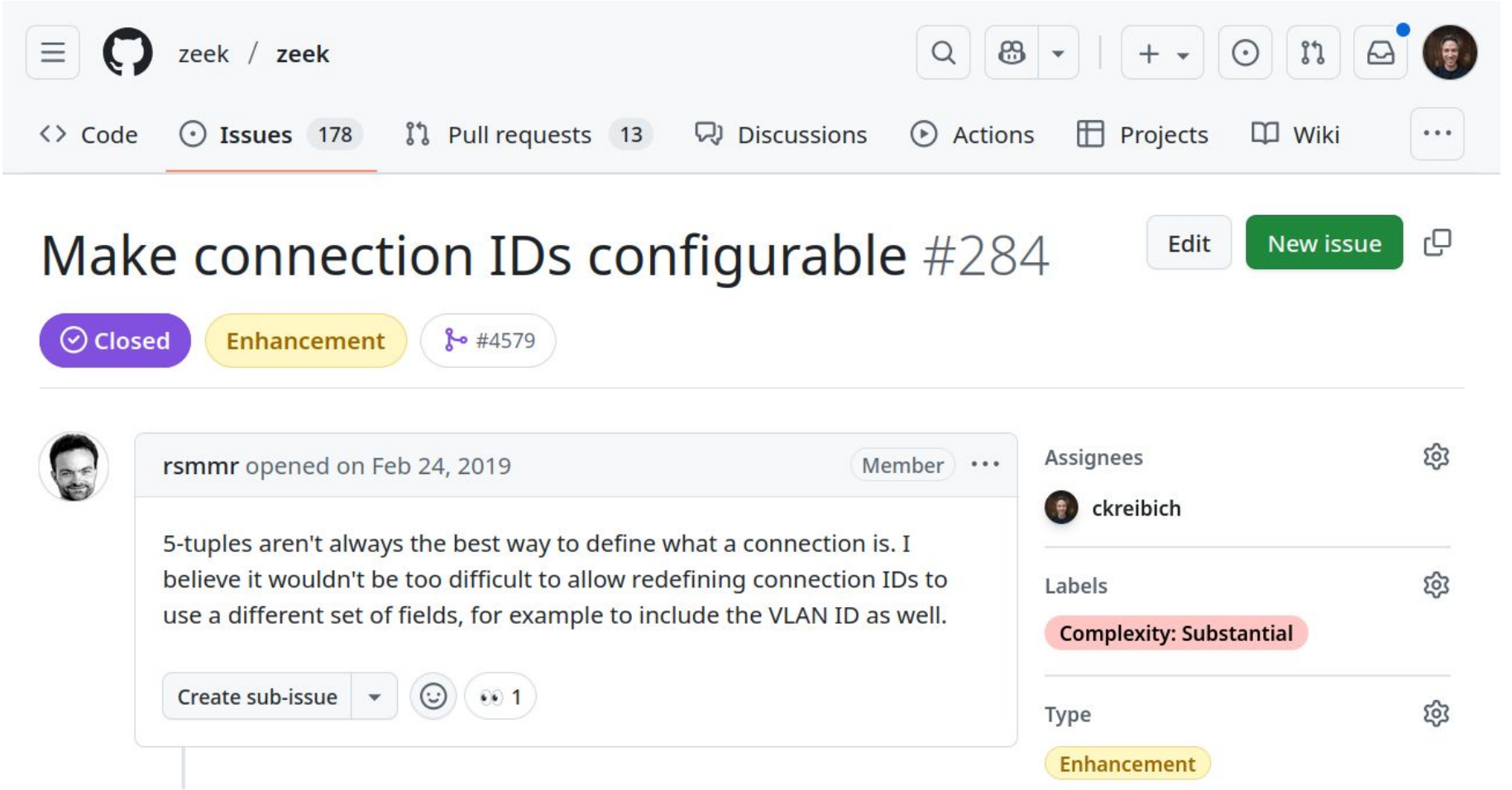
Broker



ZeroMQ



- ZeroMQ Simplifies pub/sub; central event tap now feasible
- ZeroMQ is available in Zeek 8.0, but off by default
- ZeroMQ is the default in Zeek 8.1, but easily reverted to Broker


zeek / zeek
Code
Issues 178
Pull requests 13
Discussions
Actions
Projects
Wiki
Make connection IDs configurable #284
Edit
New issue
Closed
Enhancement
#4579
rsmmr opened on Feb 24, 2019
Member
5-tuples aren't always the best way to define what a connection is. I believe it wouldn't be too difficult to allow redefining connection IDs to use a different set of fields, for example to include the VLAN ID as well.
Create sub-issue
1
Assignees
ckreibich
Labels
Complexity: Substantial
Type
Enhancement

Pluggable connection tuples

- Zeek 8 features a new pluggable component: `ConnKey` classes
- Zeek's notion of flow tuple is now configurable — you can include
  - VLANs
  - VXLAN / Geneve VNIs
  - Custom packet capture metadata
  - Anything you'd like to implement!
- Zeek 8 ships with two implementations:
  - Standard 5-tuple remains default
  - VLAN & Q-in-Q aware tuples

```
@load frameworks/conn_key/vlan_fivetuple
```

Pluggable connection tuples

- Try out VLAN awareness:

```
$ zeek -r vlan.pcap frameworks/conn_key/vlan_fivetuple tuning/json-logs
$ cat conn.log | jq
{
    "ts": 1362692526.919344,
    "uid": "CZ9ZhJgujZbSf1sa6",
    "id.orig_h": "141.142.228.5",
    "id.orig_p": 59856,
    "id.resp_h": "192.150.187.43",
    "id.resp_p": 80,
    "id.ctx.vlan": 42,
    "proto": "tcp",
    …
```

```
type conn_id: record {
        orig_h: addr &log;
        orig_p: port &log;
        resp_h: addr &log;
        resp_p: port &log;
        proto: count &default=65535;
        ctx: conn_id_ctx &log &default=conn_id_ctx();
};
```

- See Zeek's policy/frameworks/conn_key/vlan_fivetuple test for an example of colliding 5-tuples disambiguated by VLANs

Log schema support

- Goal: describe what Zeek's logs actually *look like*
- Implemented in a script-only Zeek package, `logschema`
- Supports JSON Schema, CSV, simplified JSON, and a logschema.log
- Enables change detection, data validation, log field provenance, …

```
$ zkg install logschema
$ zeek logschema/export/jsonschema
$ jq . zeek-conn-log.schema.json | head
{
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "title": "Schema for Zeek conn.log",
    "description": "JSON Schema for Zeek conn.log",
    "type": "object",
    "properties": {
      "ts": {
        "description": "This is the time of the first packet.",
        "type": "number",
    …
```

Zeek 8.2

A new state propagation mechanism

```
table state[string] of count &publish_on_change=[
    $changes=set(TABLE_ELEMENT_NEW),
    …
];
```

- A replacement for Broker-backed tables & datastores
- A more explicit, non-magic replacement of `&synchronized`
- Explicit eventing remains a viable — and now easier — option too
- Consider the Storage Framework for persistence & "shared memory"

Other improvements

- IGMP multicast group memberships
- NAT traversal protocols
- An improved & faster Spicy
- New `connection_timing_out` hook to override flow expiration
- Redefinable well-known ports
- Encrypted ZeroMQ communication
- Improved testability on Windows

A major docs improvement

Zeek 9 and beyond

(Forward-looking statements. Handle with care.)

A simpler, modern Zeek cluster

- ZeroMQ as a mature, fast, true pub/sub cluster backend
- A new WebSocket stack and client library for talking to Zeek
  - Focus on event tx/rx, not I/O, in your third-party apps
  - Already supports C, Python, Rust, and JavaScript
- JavaScript for talking from Zeek to external APIs
- New systemd- and container-oriented cluster management

A new zkg and package/plugin management

- zkg has not seen significant updates in years …
- … but there's quite a bit to improve:
  - git shines through (or masks errors!) in many places
  - It's too hard to develop packages with it
  - Bundles have strange failure modes
  - It never cleans up after itself
  - Plugin-packages can end up double-loading scripts
- We have designs for simplifying all of these: zkg itself, Zeek's plugin management, and package layouts.

Thanks!

Documentation

https://docs.zeek.org

Community links

https://zeek.org/community

Github project

https://github.com/zeek

Project wiki

https://github.com/zeek/zeek/wiki

Zeek is 30 !