Storage Framework: A Primer

Tim Wojtulewicz, Corelight

Why Add a New Framework?

- Part of a greater push to move on from Broker
  - Telemetry, cluster, storage
- Issues with the existing API
- Difficulty in adding new backends

The Old API

- `Broker::create_master`/`create_clone`
  - Create the master once, then create clones
  - What happens if the master crashes? Or the workers?
- `Broker::put`/`put_unique`/`get`/etc
  - Split APIs for overwriting are annoying
  - Custom Broker types for everything
- `&backend`
  - Table-based interface is nice, but complicated underneath
- Async only

The New API (added in 7.2)

- Generic API for interacting with storage
- Plugin-based, like so many things in Zeek
  - Storage backends and serialization
- Good for long-term storage
  - And maybe low-traffic per-node synchronization? Depends on the backend.
- Sync and async modes
- No table-based model (yet?)

Example

(known-services.zeek)

What's Missing?

- A table-based interface
  - Storage-backed tables
  - `&publish_on_change`
- A memory-backed backend
- Serialization other than JSON
- Mutating operations (`add`/`subtract`, `push`/`pop`)