Tessera

Trillian TNG

Martin Hutchinson & Al Cutter
9th October 2024

Google

Trillian: A brief history

- github.com/google/trillian [2016]
- Application agnostic
- Microservices
- Multi-tenancy
- Tiles internally, but not via API
- Actively maintained...

  ... But new features not planned

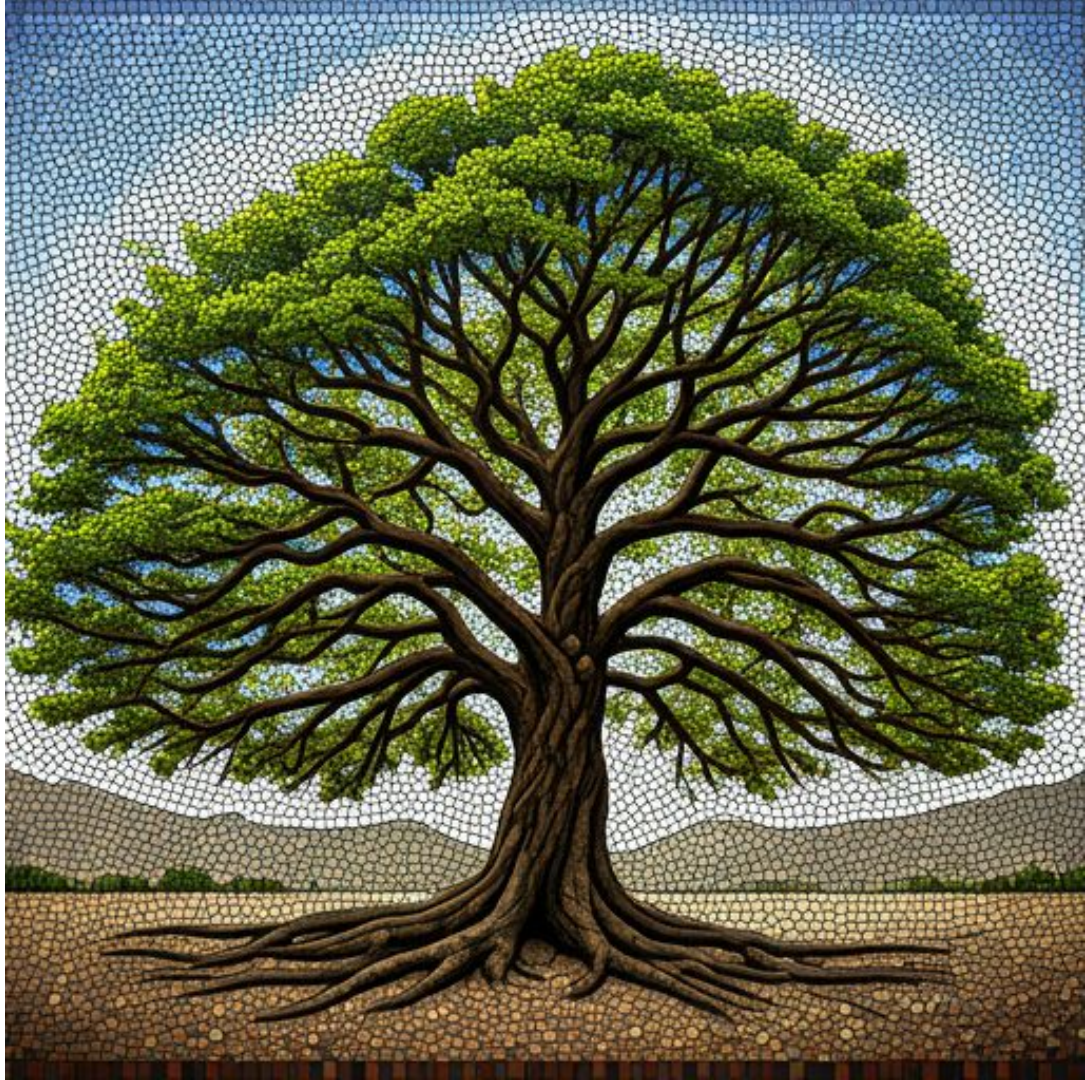
A different approach

Goals:

- Simplicity as a 1st class goal
- Pay only for what you *need*
- Storage infrastructure native
- Synchronous sequencing...
- ...but asynchronous integration
- Opinionated about logs

Tessera: Overview

A *lightweight* Go *library* for building *tiled logs*



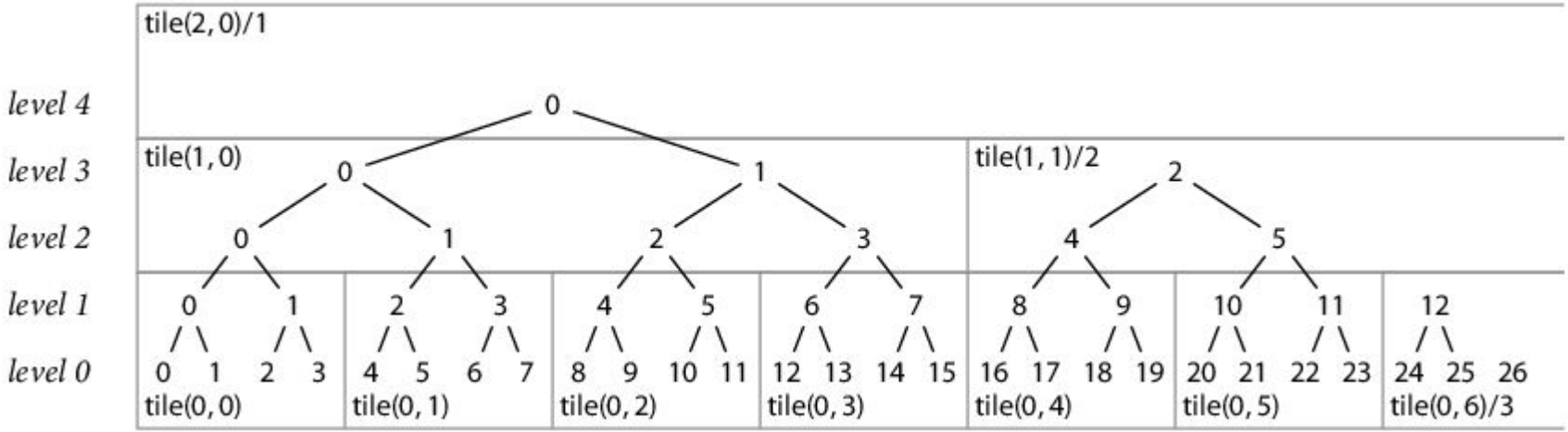
What are Tiles?

- Address regions of tree
- Coordinate address:
  - Level
  - Offset

example.com/mylog/tile/2/1

API: c2sp.org/tlog-tiles

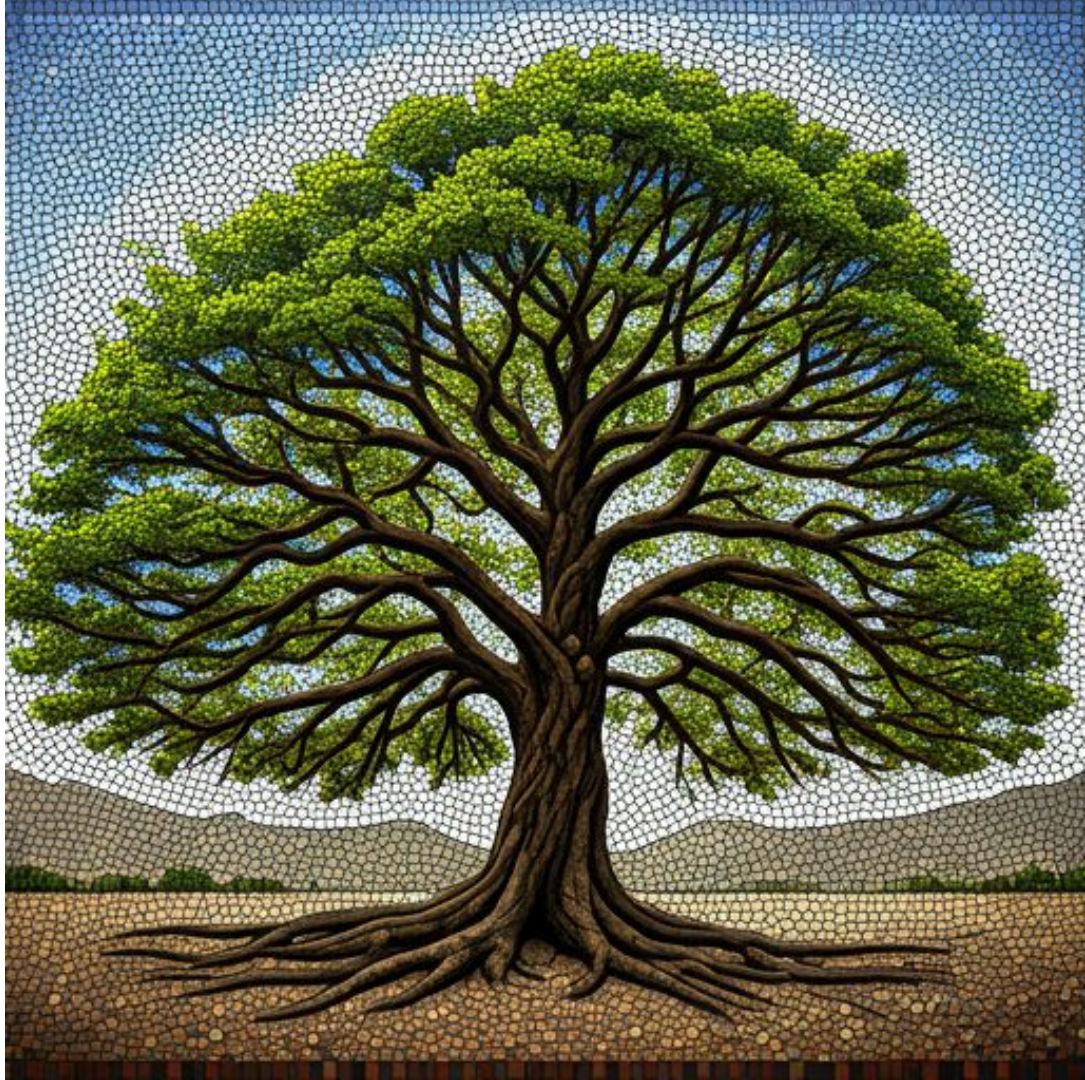
Tiles



API formalized as c2sp.org/tlog-tiles - hurrah for intercompatibility

Tessera: Overview

A *lightweight* Go *library* for building *tiled logs*



Tessera: Lightweight

- Native APIs for:
  - GCP
  - AWS (coming soon)
  - MySQL
  - POSIX
- Reads are *cheap*

Simpler:

- No multi-tenancy

Tessera: Library

You're in control:

- Deployment architecture
- Scale
  - multiple writers, if needed!
- Instrumentation

Tessera: Tiled Logs

Tiles native:

- Tiles in storage
- Tiles via API

Supports:

- tlog-tiles API
- CT Static API

Who can use Tessera?

Anyone!

Deploying *log* with:

- tlog-tiles API; or
- CT Static API

Prepared to be early adopter

Tessera: Library Usage

```
// Initialise the Tessera MySQL storage
storage, err := mysql.New(ctx, db,
  tessera.WithCheckpointSignerVerifier(s, v))


// Add an entry (e.g. in a POST request handler)
idx, err := storage.Add(ctx, tessera.NewEntry(bs))()
```

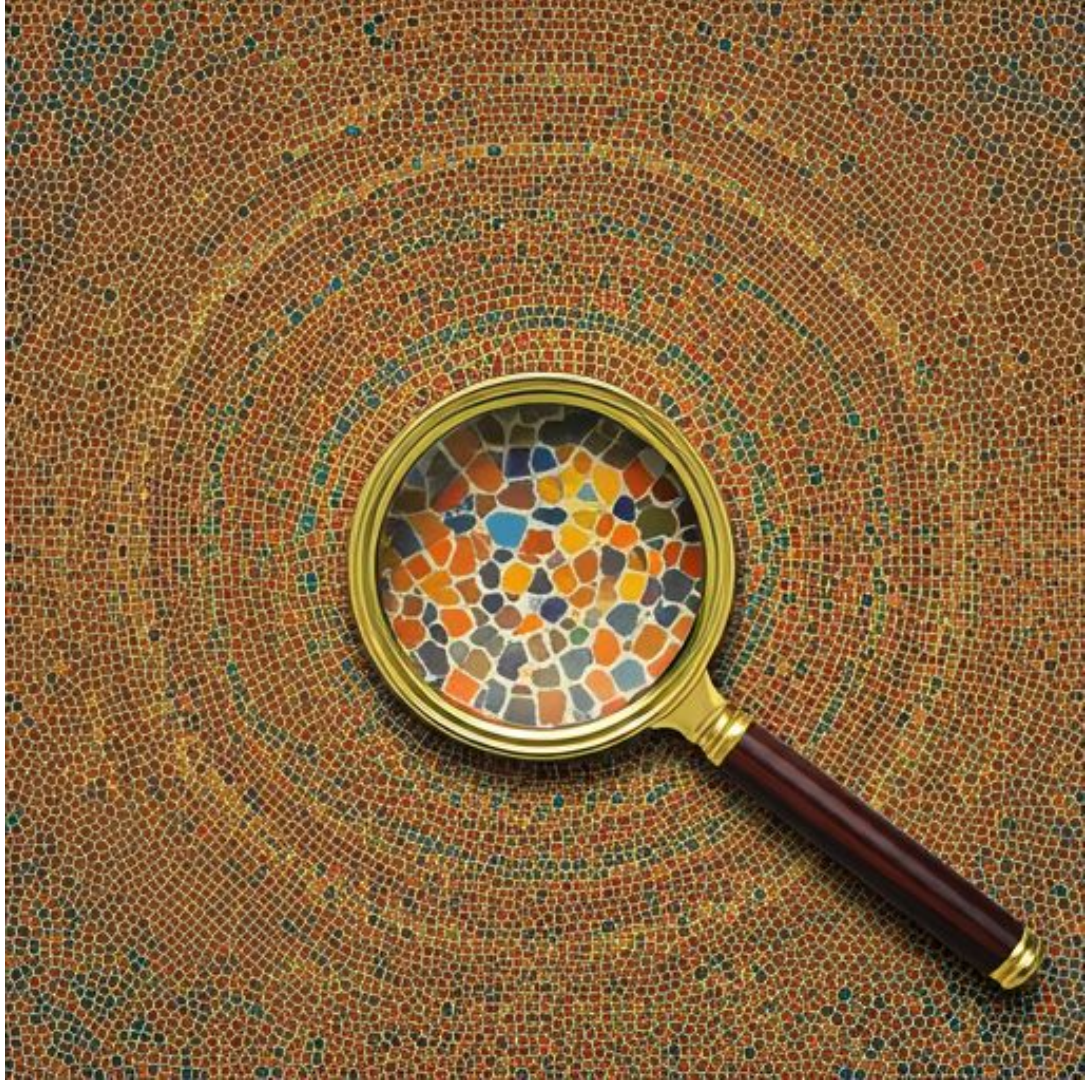
~/git/trillian-tessera/cmd/conformance/mysql/docker ——— mhutchinson@thinkcentre 10:03:59

❯

Tessera: Examples

`/cmd/`:

- `conformance/`
  - `gcp/`
  - `mysql/`
  - `posix/`
- `examples/`
  - `posix-oneshot/`



Implementations

- GCP
  - Spanner for temporary sequencing
  - GCS for constructed tiles & serving
- AWS
  - Implementation TBD
- MySQL
  - Simple tables
  - Point lookups for serving
- POSIX
  - Files on disk

Status and Timeline

- Current Status
  - GCP, MySQL, POSIX are done; AWS implementation underway
  - Pre-alpha, but things are in good shape and CI tested
  - ~1kqps writes*
- Next steps
  - Aiming for alpha within a month or so
- Getting involved
  - Repo is public, and we actively welcome people to try deploying what we have
  - Example code can be deployed via
    i. `go run`; OR
    ii. `docker compose`; OR
    iii. `terragrunt deploy`

Questions?