The Zen of Prometheus

PromCon Online 2020

Who am I?

SWE/SRE @Red Hat

Observability Platform Team

Thanos Maintainer

twitter/kkakkoyun

github/kakkoyun

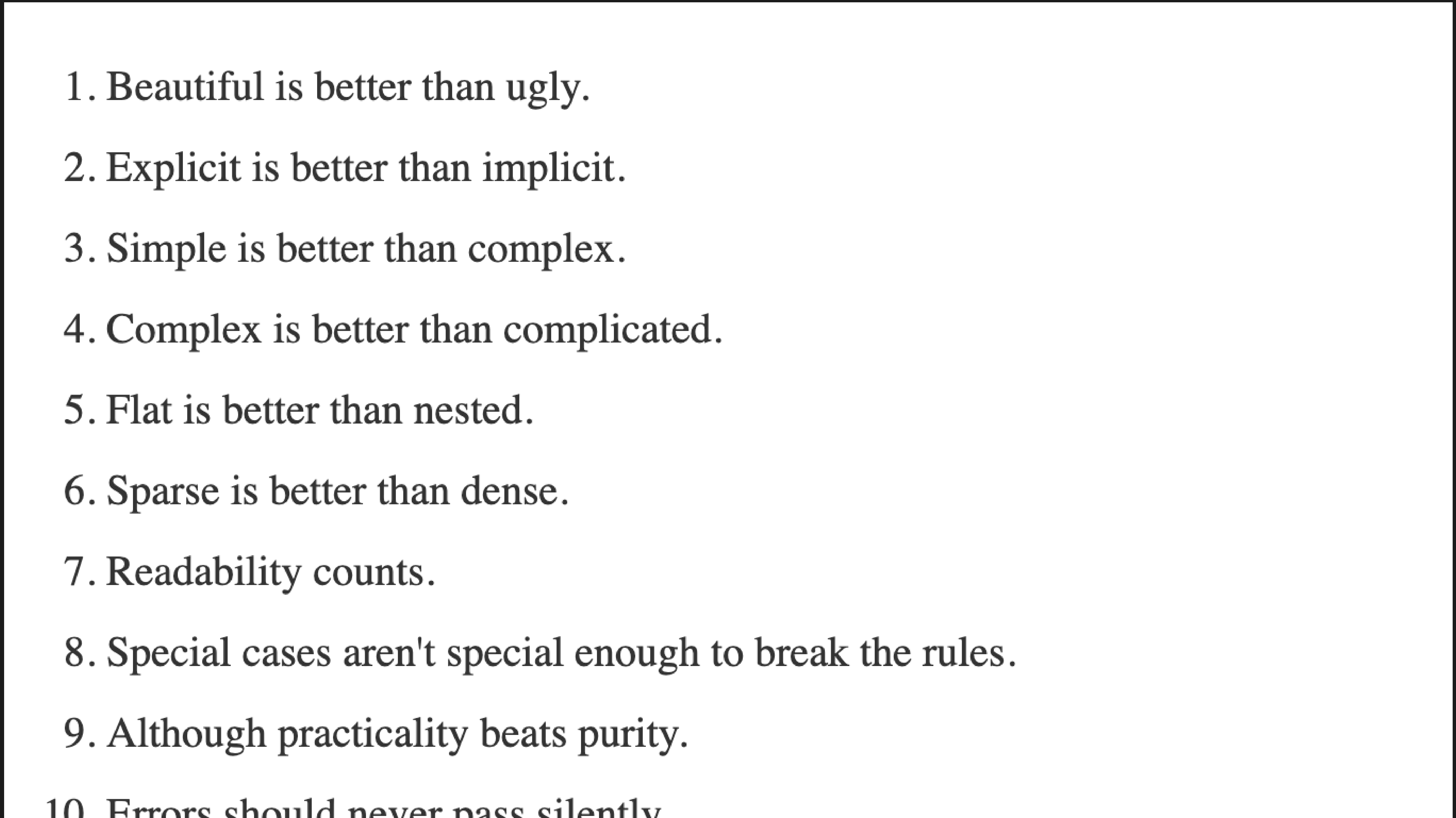
What is a *Zen*?

> *Zen emphasizes rigorous self-restraint, meditation-practice, insight into the nature of mind and nature of things, and the personal expression of this insight in daily life, especially for the benefit of others.*
>
> *- Wikipedia*

> *A collection of insights gather from the daily life for the benefit of others*

Zen of Python

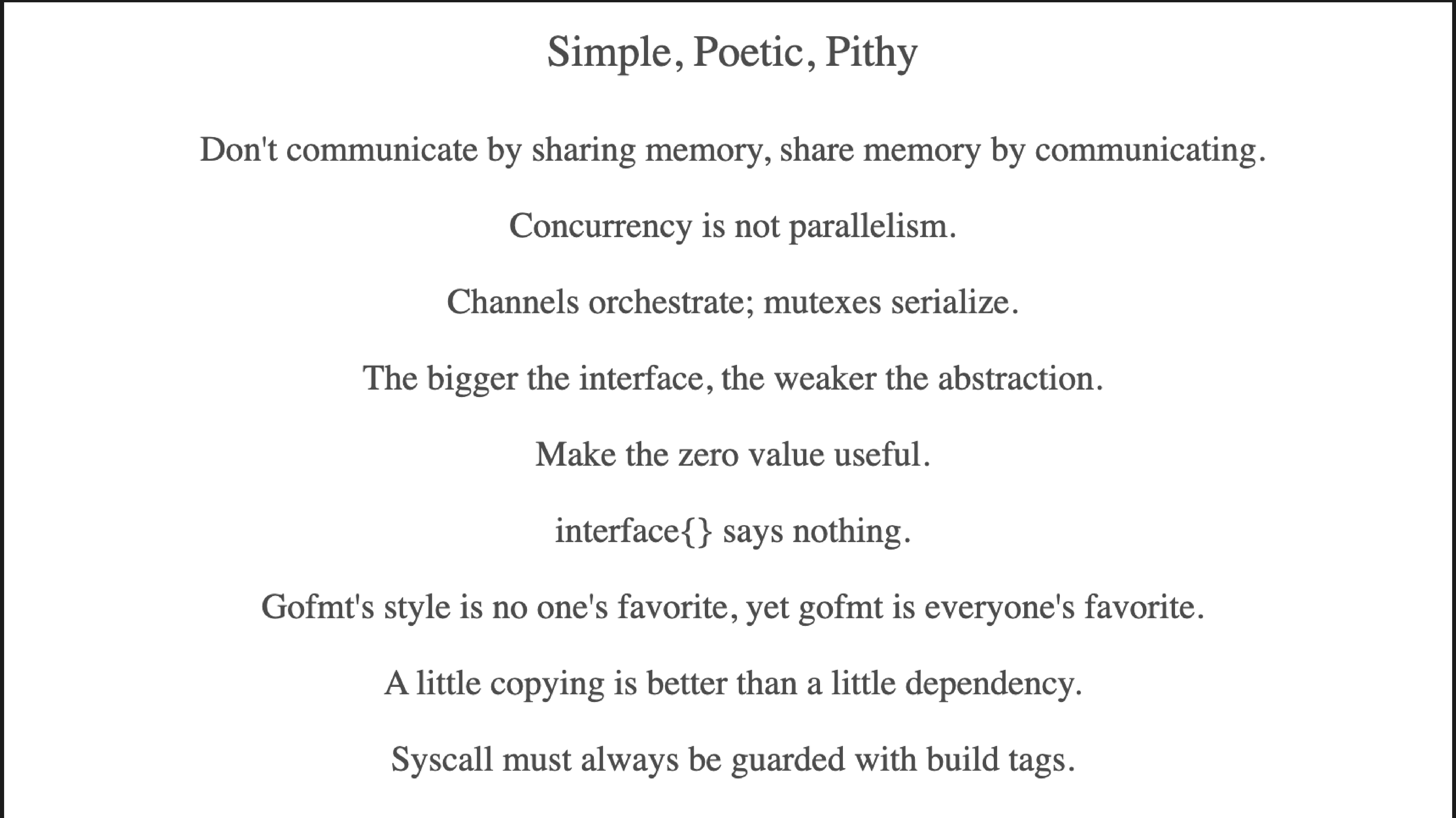
https://zen-of-python.info/



1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently

Go Proverbs

https://go-proverbs.github.io/



Simple, Poetic, Pithy

Don't communicate by sharing memory, share memory by communicating.

Concurrency is not parallelism.

Channels orchestrate; mutexes serialize.

The bigger the interface, the weaker the abstraction.

Make the zero value useful.

interface{} says nothing.

Gofmt's style is no one's favorite, yet gofmt is everyone's favorite.

A little copying is better than a little dependency.

Syscall must always be guarded with build tags.

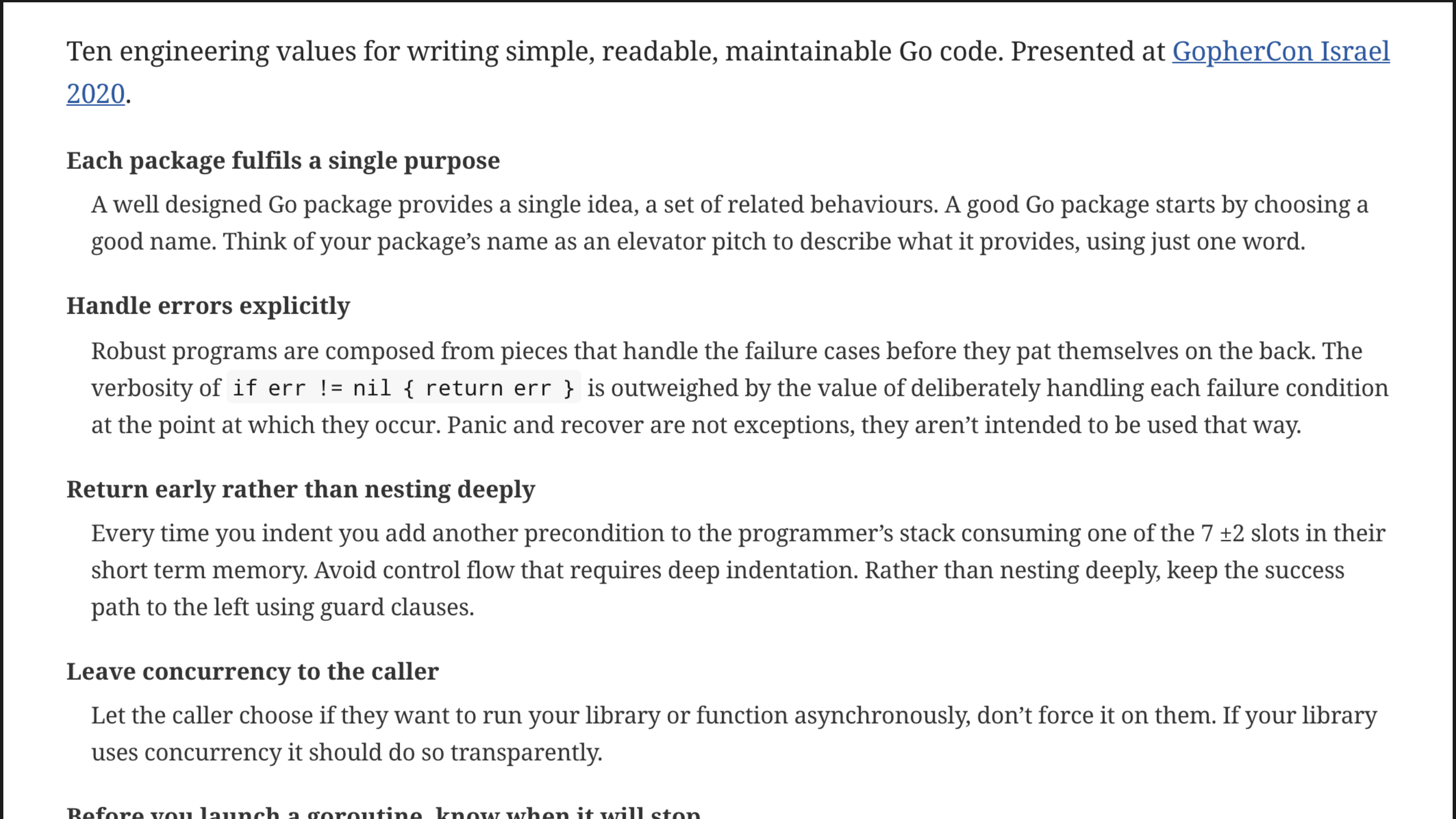
So what is a *proverb*?

What makes it different?

> *proverb: a short, well-known pithy saying, stating a general truth or piece of advice*

The Zen of Go

https://the-zen-of-go.netlify.app/



Ten engineering values for writing simple, readable, maintainable Go code. Presented at GopherCon Israel 2020.

Each package fulfils a single purpose

A well designed Go package provides a single idea, a set of related behaviours. A good Go package starts by choosing a good name. Think of your package's name as an elevator pitch to describe what it provides, using just one word.

Handle errors explicitly

Robust programs are composed from pieces that handle the failure cases before they pat themselves on the back. The verbosity of `if err != nil { return err }` is outweighed by the value of deliberately handling each failure condition at the point at which they occur. Panic and recover are not exceptions, they aren't intended to be used that way.

Return early rather than nesting deeply

Every time you indent you add another precondition to the programmer's stack consuming one of the 7 ±2 slots in their short term memory. Avoid control flow that requires deep indentation. Rather than nesting deeply, keep the success path to the left using guard clauses.

Leave concurrency to the caller

Let the caller choose if they want to run your library or function asynchronously, don't force it on them. If your library uses concurrency it should do so transparently.

Before you launch a goroutine, know when it will stop

Prometheus Proverbs

Prometheus –
Myths and Proverbs
Prometheus Meetup, Berlin – 2018-10-09
Production Engineer, SoundCloud Ltd.
SOUNDCLOUD
Prometheus

- Instrument first, ask questions later
- Counters rule and gauges suck
- First the rate, then aggregate
- Labels are the new hierarchies
- If you can graph it, you can alert on it

The Zen of Prometheus

https://the-zen-of-prometheus.netlify.app/

- Instrument first, ask questions later
- Measure what users care
- Labels are the new hierarchies
- Avoid missing metrics
- Cardinality Matters
- Naming is hard
- Counters rule and gauges suck
- First the rate, then aggregate
- If you can log it, you can have a metric for it
- One does not simply use Histograms
- If you can graph it, you can alert on it
- If you run it, then you should put an alert on it
- Alerts should be urgent, important, actionable, and real
- Symptom-based alerts for paging, caused-based for troubleshooting
- Please five more minutes
- Context is king

Instrument first, ask questions later

Measure what *users* care about

- SLOs

- SLOs
- RED

- SLOs
- RED
- USE

- SLOs
- RED
- USE
- The Four Golden Rules

Cardinality Matters

Labels are multiplicative

> Prometheus performance almost always comes down to one thing: label cardinality

One does not simply use *Histograms*

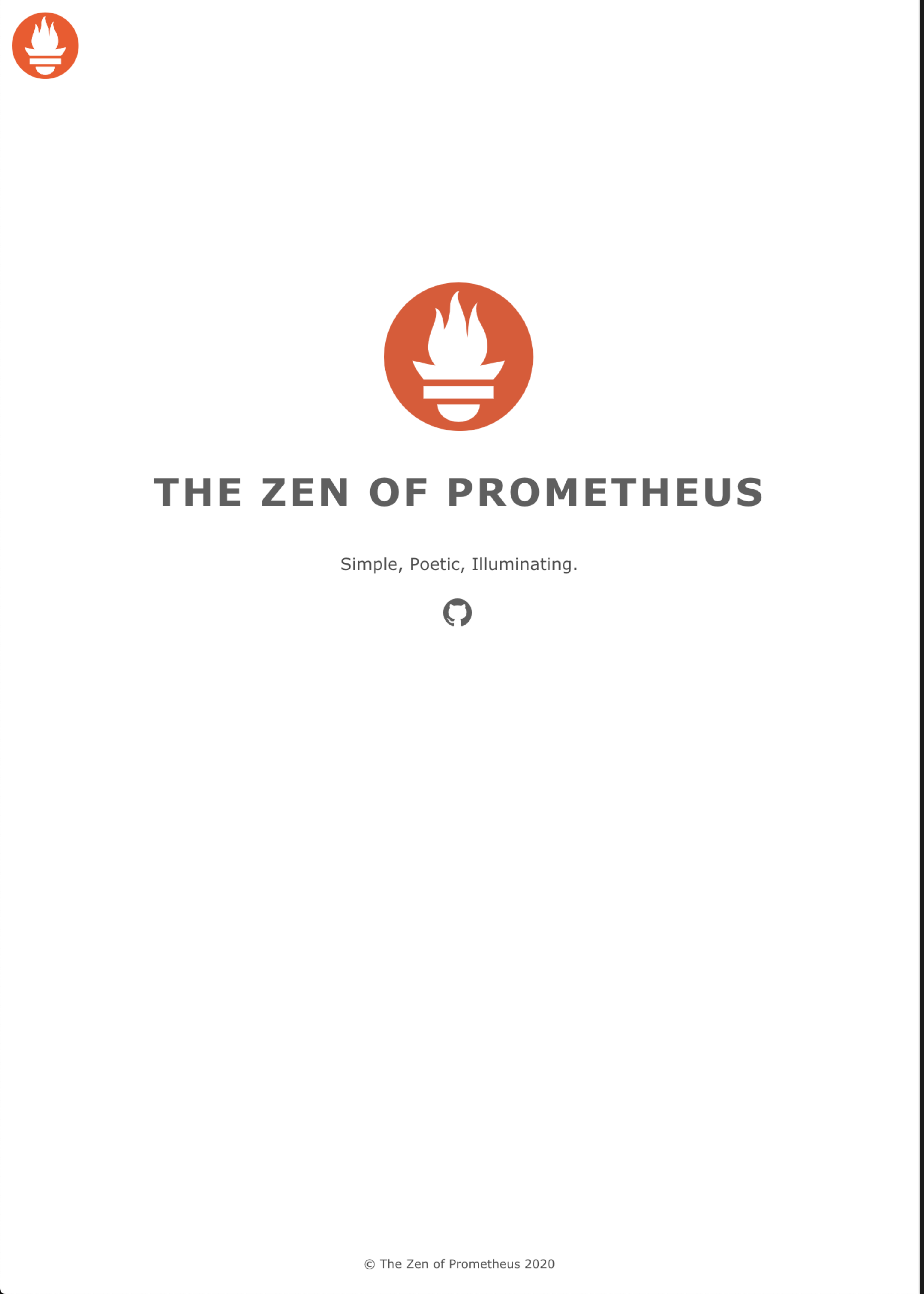
ONE DOES NOT SIMPLY USE
HISTOGRAMS
imgflip.com

With great power there must also come great responsibility

SLOS, SLOS, SLOS
SLOS EVERWHERE
imgflip.com

Cardinality is key

Symptom-based **alerts for paging,** ***caused-based*** **for troubleshooting**



THE ZEN OF PROMETHEUS

Simple, Poetic, Illuminating.

© The Zen of Prometheus 2020

Now It's *Your* Turn to contribute

Red Hat