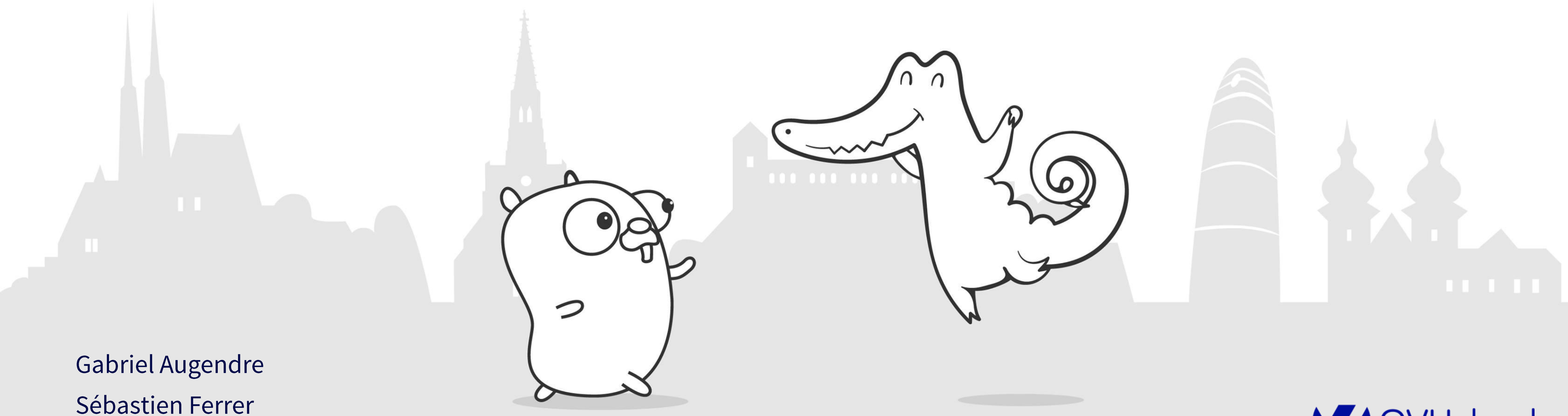
Your code deserves better

give it a linter!

Gophercamp 2025



Gabriel Augendre

Sébastien Ferrer

OVHcloud

Gabriel Augendre

Technical lead @ OVHcloud

Crocmagnon

Sébastien Ferrer

Software engineer @ OVHcloud

sebferrer

Preamble

Preamble

```
func Process() (User, error) {
    user, _ := Get_user_by_id(42)
    fmt.Println(user.Name)

    return user, nil
}
```

```
if (foo) {
  if (bar) {
    if (baz) {
      if (lorem) {
        if (ipsum) {
          fmt.Println("Hi!")
} } } } }
```

```
func LongFuction() error {

    // 4586 lines of code

}
```

```
func aVeryLongFunctionNameThatShouldntBeThatLong()
(User, error) {
user, _ := GetUserByID(42)
fmt.Println(user.Name)
notEvenUsedVariable, _ := testFunction(ctx, user)
notEvenUsedVariable2, _ := testFunction(ctx, user)
notEvenUsedVariable3, _ := testFunction(ctx, user)
fmt.Println(notEvenUsedVariable1)
fmt.Println(notEvenUsedVariable2)
fmt.Println(notEvenUsedVariable3)
if user.IsAdmin() { //  ... }
return user, nil }
```

Preamble

Everything works...

. . .but nothing is coherent

Preamble

We need **rules**

shared conventions

Linters

The compiler

All your base are beloong to us

↓

All your bases belong to us

grammarly

The linter

We own all your bases

Or

All of your bases are ours

Linters

Detects **inconsistencies**

Prevents **errors** before they are even committed

Creates a **common language** within the team

Enhances **security** and **performance**

Is your best friend

Linters, state of the art

Flake8
Ruff
Black

ESLint
StandardJS

Checkstyle
PMD

Clang-Tidy
Cppcheck

revive
go vet
staticcheck

golangci-lint: linters runner

golangci-lint

```
func Process() (User, error) {
    user, _ := GetUserByID(42)
    fmt.Println(user.Name)

    if user.IsAdmin() {
      // …
    }

    return user, nil
}
```

Error return value is not checked (**errcheck**)

```
func Process() (User, error) {
    user, err := GetUserByID(42)
    if err != nil {
        return User{}, err
    }

    if user.IsAdmin() {
      // …
    }

    return user, nil
}
```

exported: exported function Process should have comment or be unexported (**revive**)

```
// Process is an example function
// written for our Gophercamp talk
func Process() (User, error) {
    user, err := GetUserByID(42)
    if err != nil {
        return User{}, err
    }

    if user.IsAdmin() == true {
      // …
    }

    return user, nil
}
```

S1002: should omit comparison to bool constant, can be simplified to `user.IsAdmin()` (**gosimple**)

.golangci.yml

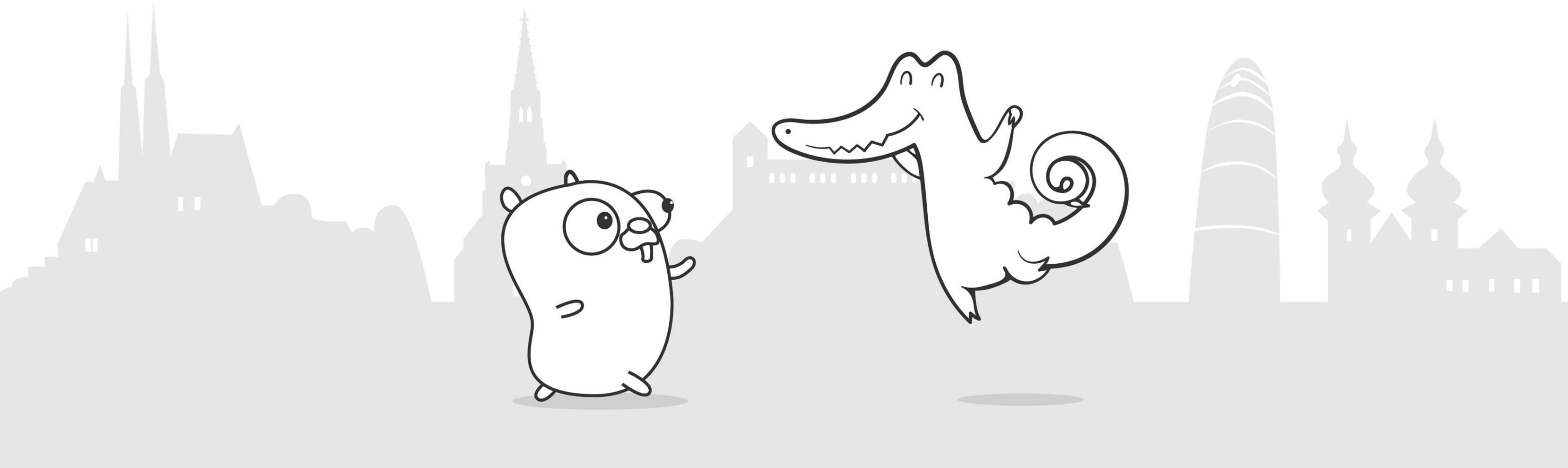
```
run:
  timeout: 2m

linters:
  enable:
    - revive
    - staticcheck
    - govet

issues:
  max-issues-per-linter: 0
  max-same-issues: 0
```

→ How the linter runs

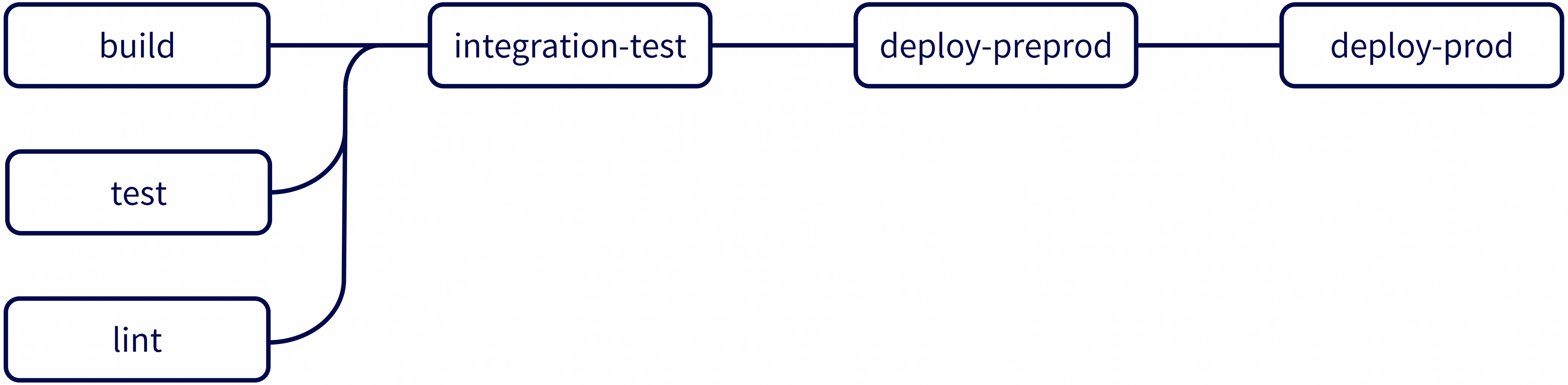
→ Which linters run

→ How results are handled

Demo time!



CI/CD


build
test
lint
integration-test
deploy-preprod
deploy-prod

Linters: limitations

They don’t replace tests or code reviews

False positive or negative

Lack of contextual understanding

Limited to defined rules

Linters: limitations

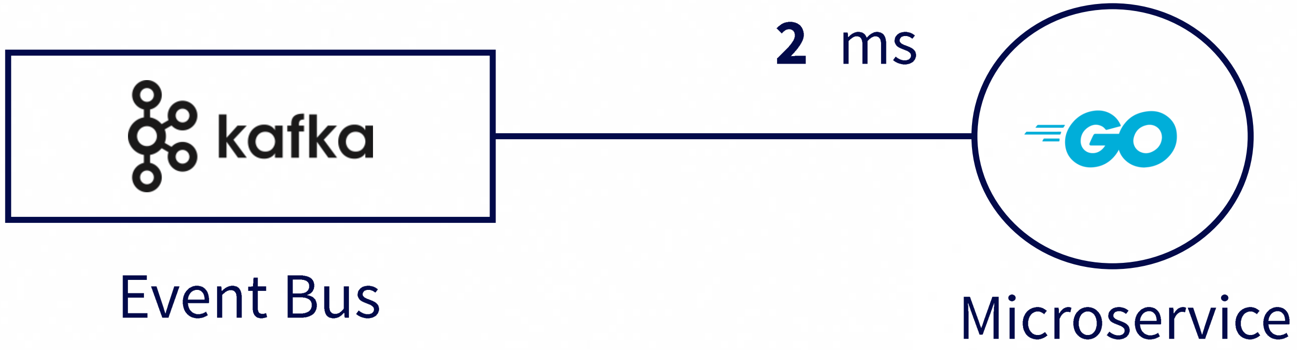
Limited to defined rules

Are we?

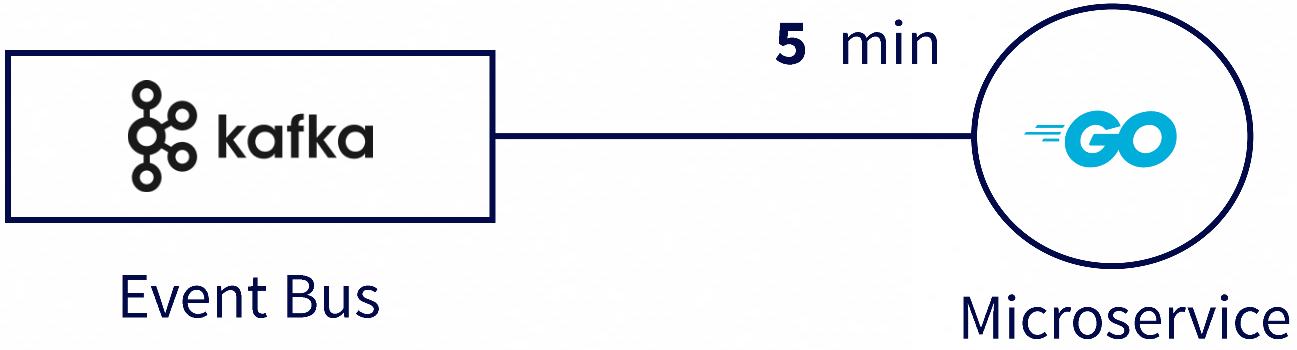
A lag history

Once upon a time…

A lag history



A lag history



A lag history

Microservice

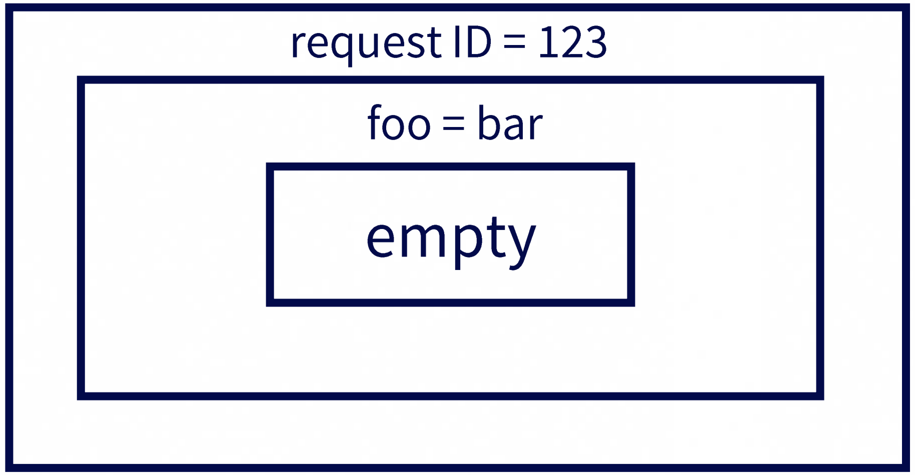
```
ctx = context.WithValue(ctx, "foo", "bar")

for msg := range kafkaConsumer() {

    ctx = context.WithValue(ctx, "requestID", msg.Key)

    fields := logrus.Fields {
        "request_id": ctx.Value("requestID"),
        "foo":        ctx.Value("foo"),
    }

    logrus.WithFields(fields).Infof("Received: %s", msg.Value)
}
```

A lag history

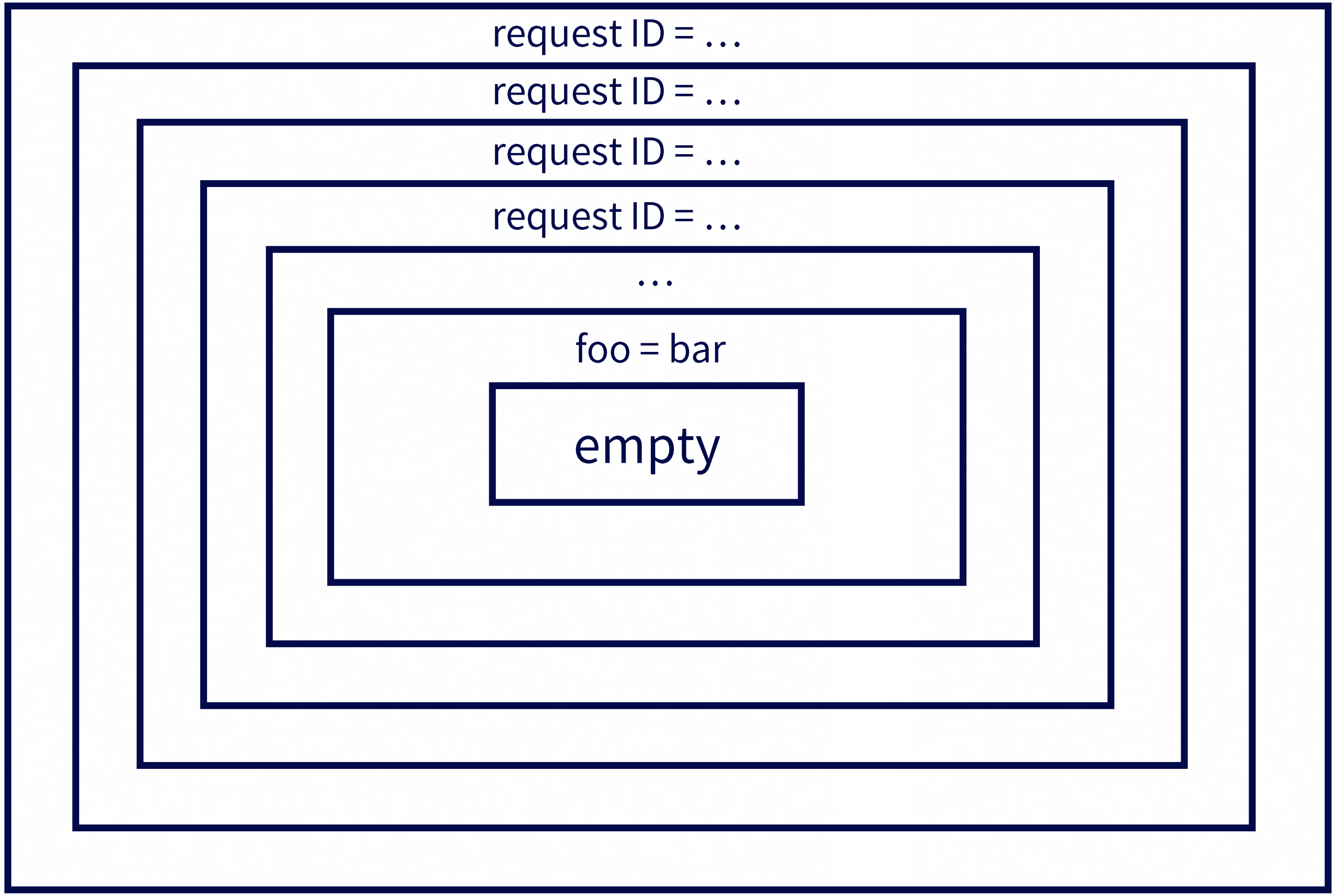
Microservice

```
ctx = context.WithValue(ctx, "foo", "bar")

for msg := range kafkaConsumer() {

    ctx = context.WithValue(ctx, "requestID", msg.Key)

    fields := logrus.Fields {
        "request_id": ctx.Value("requestID"),
        "foo":        ctx.Value("foo"),
    }

    logrus.WithFields(fields).Infof("Received: %s", msg.Value)
}
```

Context-induced performance bottleneck in Go



Context-induced performance bottleneck in Go



request ID = …

request ID = …

request ID = …

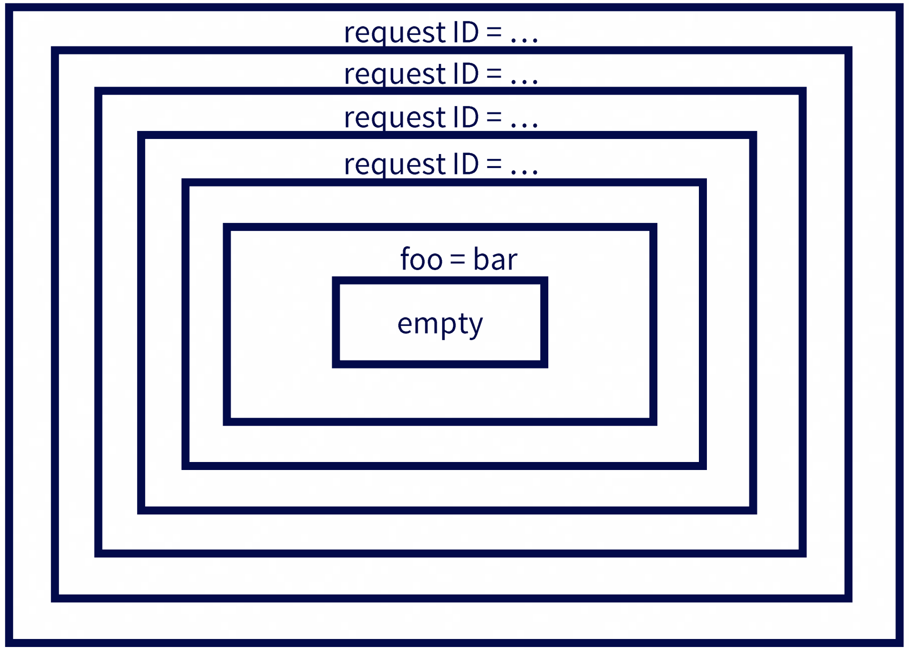
request ID = …

…

foo = bar

empty

Context-induced performance bottleneck in Go



On every iteration

a new context **wraps** the previous one

because…

A lag history

GO

Microservice

```
ctx = context.WithValue(ctx, "foo", "bar")

for msg := range kafkaConsumer() {

    ctx = context.WithValue(ctx, "requestID", msg.Key)

    fields := logrus.Fields {
        "request_id": ctx.Value("requestID"),
        "foo":        ctx.Value("foo"),
    }

    logrus.WithFields(fields).Infof("Received: %s", msg.Value)
}
```

A lag history

Microservice

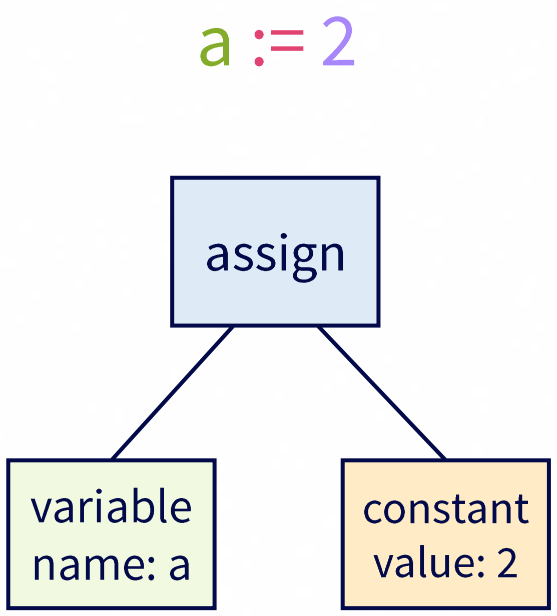
```
ctx = context.WithValue(ctx, "foo", "bar")

for msg := range kafkaConsumer() {

    ctx = context.WithValue(ctx, "requestID", msg.Key)

    fields := logrus.Fields {
        "request_id": ctx.Value("requestID"),
        "foo":        ctx.Value("foo"),
    }

    logrus.WithFields(fields).Infof("Received: %s", msg.Value)
}
```

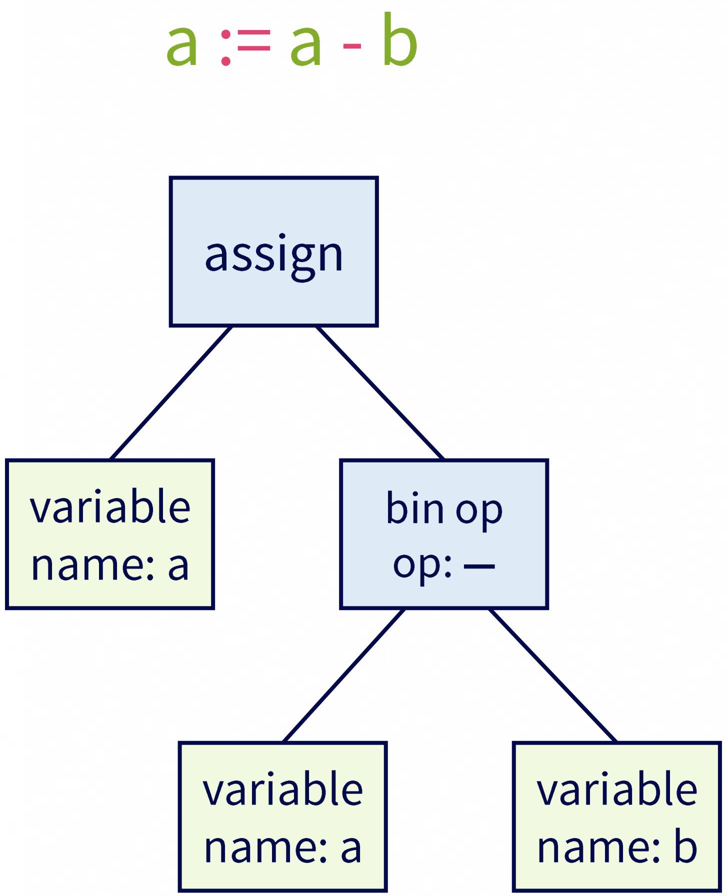
A lag history

Microservice

```
 ctx = context.WithValue(ctx, "foo", "bar")

for msg := range kafkaConsumer() {

    ctx := context.WithValue(ctx, "requestID", msg.Key)

    fields := logrus.Fields {
        "request_id": ctx.Value("requestID"),
        "foo":              ctx.Value("foo"),
    }

    logrus.WithFields(fields).Infof("Received: %s", msg.Value)
}
```

Abstract Syntax Trees (AST)

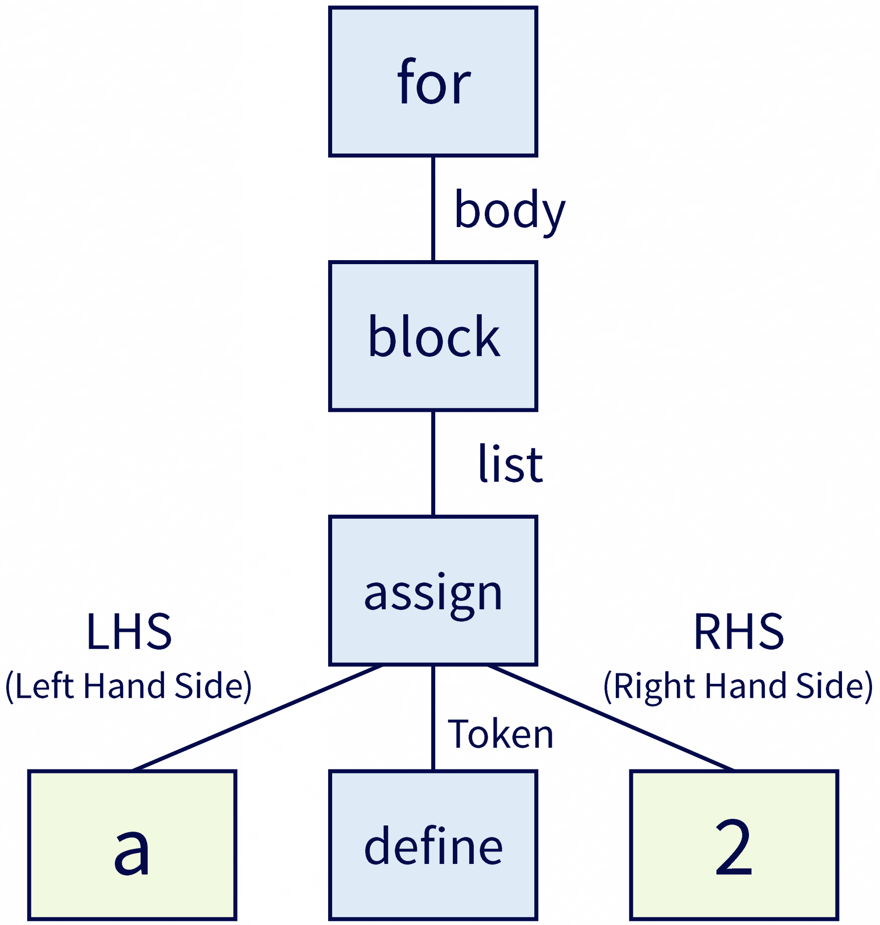




Abstract Syntax Trees (AST)

- For
- Range
- Block
- Assign

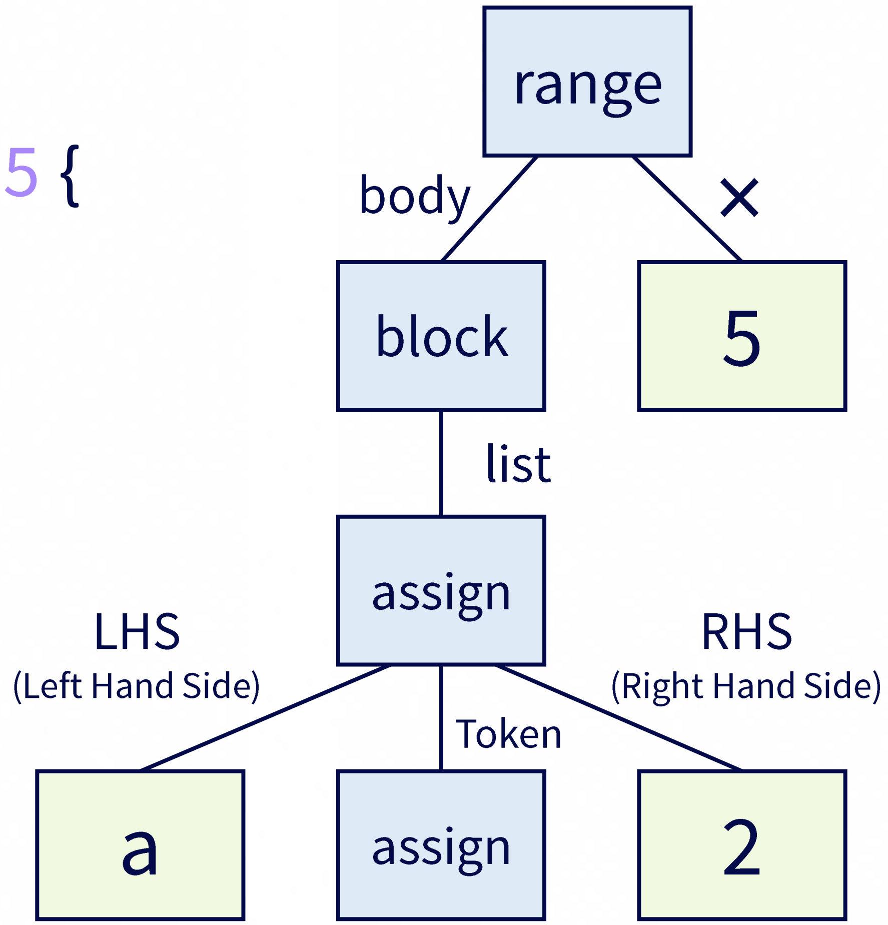
```
for {
    a := 2
}
```



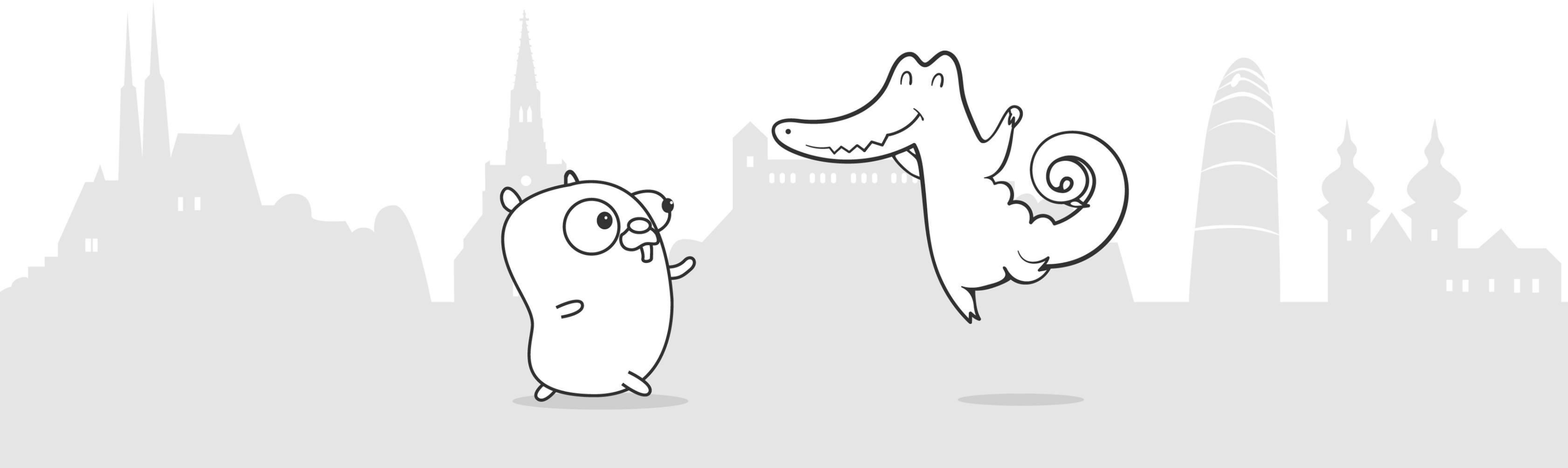
Abstract Syntax Trees (AST)

- For
- Range
- Block
- Assign

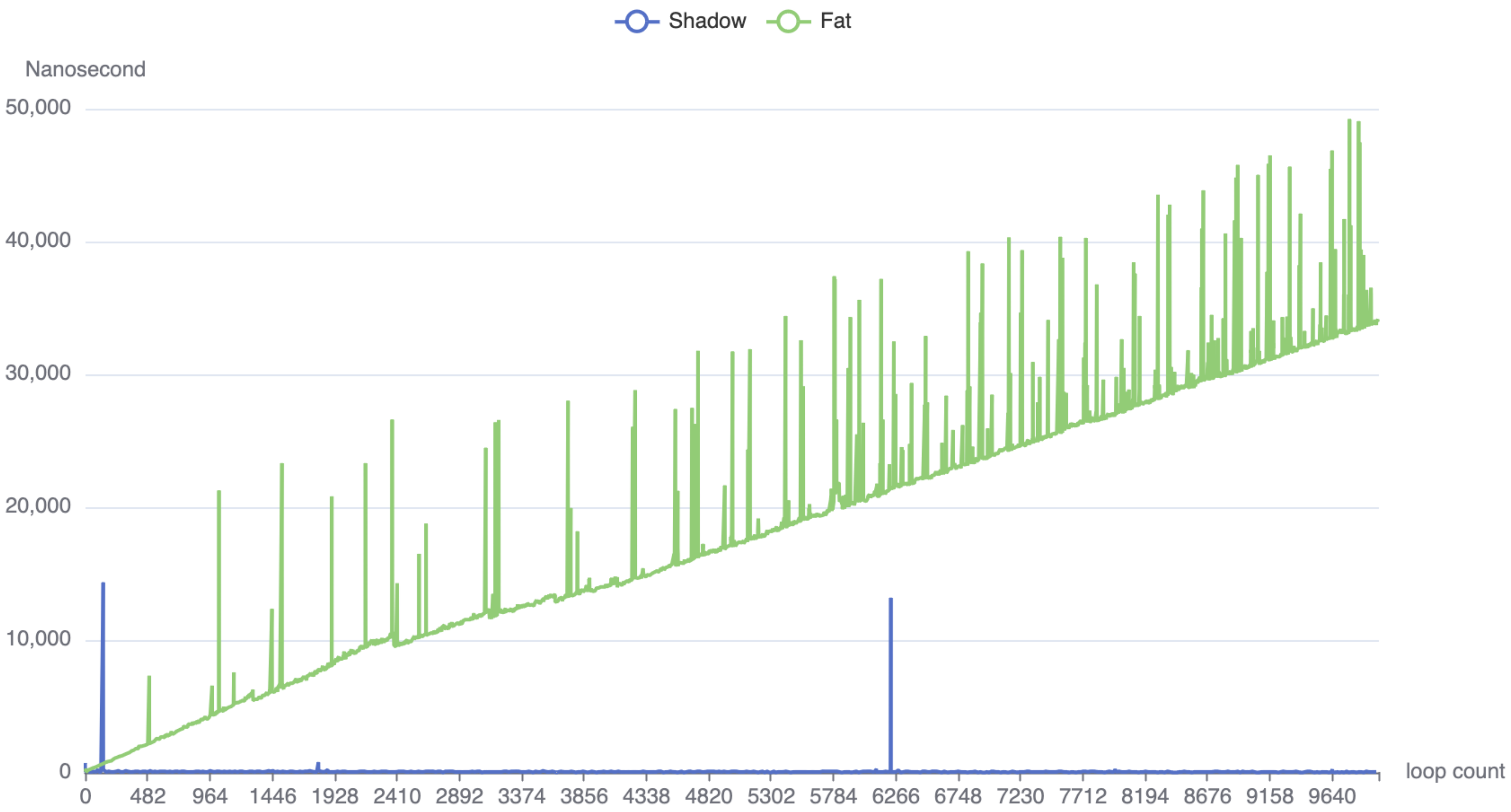
```
for range 5 {
  a = 2
}
```



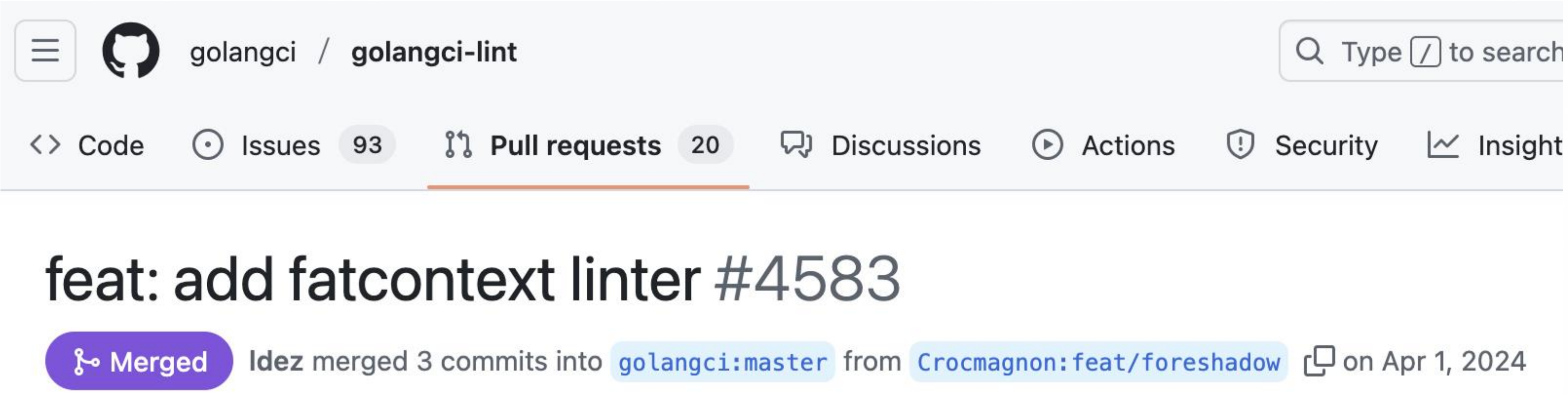
Demo time!



Benchmark


Shadow
Fat
Nanosecond
50,000
40,000
30,000
20,000
10,000
0
0
482
964
1446
1928
2410
2892
3374
3856
4338
4820
5302
5784
6266
6748
7230
7712
8194
8676
9158
9640
loop count

fatcontext linter in GitHub



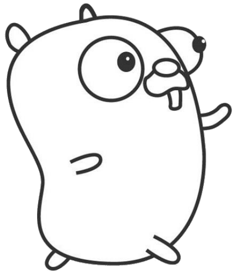
feat: add fatcontext linter #4583

Merged Idez merged 3 commits into `golangci:master` from `Crocmagnon:feat/foreshadow` on Apr 1, 2024

https://github.com/golangci/golangci-lint/pull/4583

gabnotes.org

Blog here!



Gabriel Augendre

Technical lead @ OVHcloud

Crocmagnon

Feedback ♡

Sébastien Ferrer

Software engineer @ OVHcloud

sebferrer

Blog here!

blog.kimi.ovh