



How do we build our treasury service with Golang and Blockchain

Out 01st

Who am I?

- Tech lead at CloudWalk on Banking Team
- Organizer of GopheRio
- Passionate about open source, home automation and keyboards
- Really likes coffee and wine

@caioeverest pretty much everywhere
caio-everest@cloudwalk.io



Cloudwalk

Changing the Status-quo of payment systems

- Fintech
- Unicorns since 2021
- Have its own blockchain and stablecoin (BRLC)
- Already transactioned R\$ 50 millions in BRLC



And what is InfinitePay, after all?

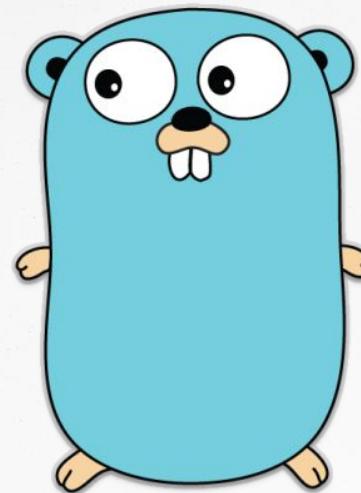
And what does blockchain have to do with it?



Challenges:

- Handling multiple databases and persistent sockets in order to merge operations with its given contexts.
- Development and environment consistent
- Clients for ethereum blockchain
- Easy to test
- Highly observable
- Documentation
- Easy to maintain

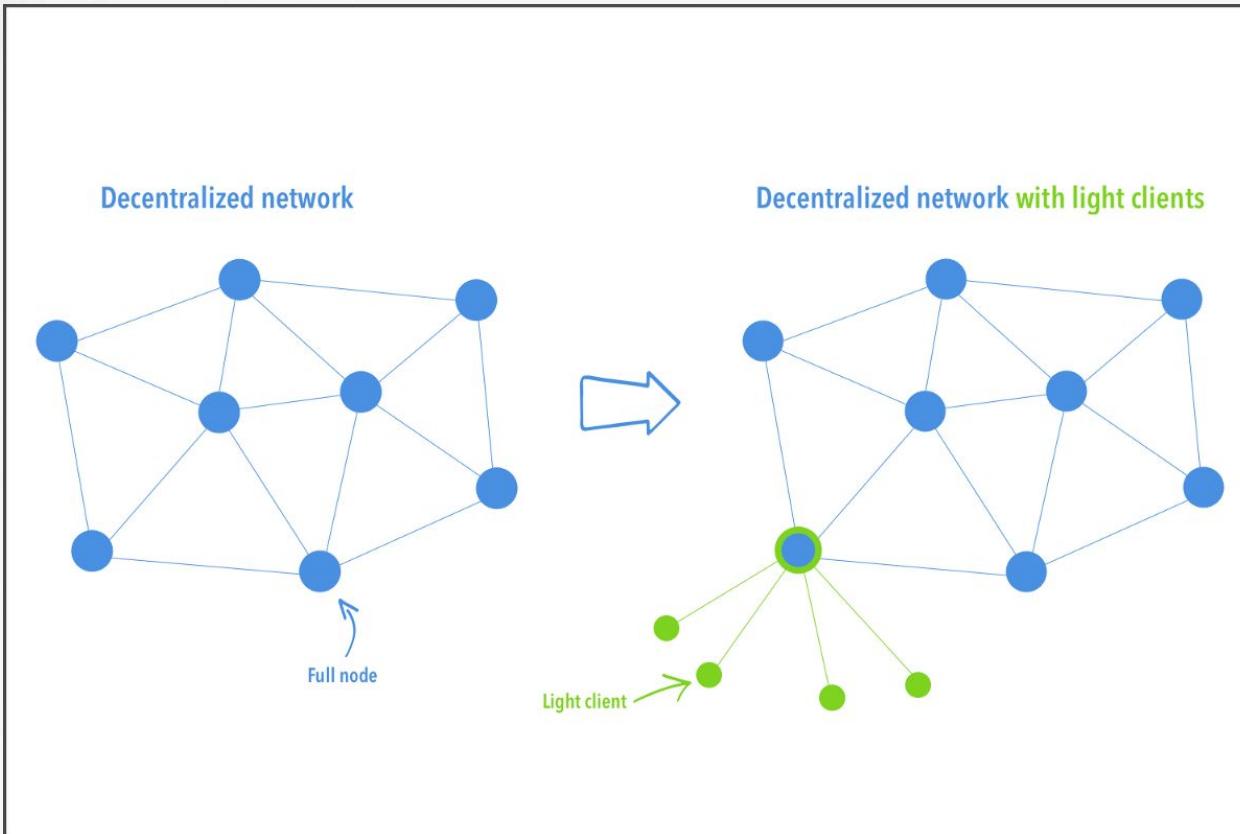
Technologies:



Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
 - Clients for ethereum blockchain
 - Easy to test
 - Highly observable
 - Documentation
 - Easy to maintain

The blockchain



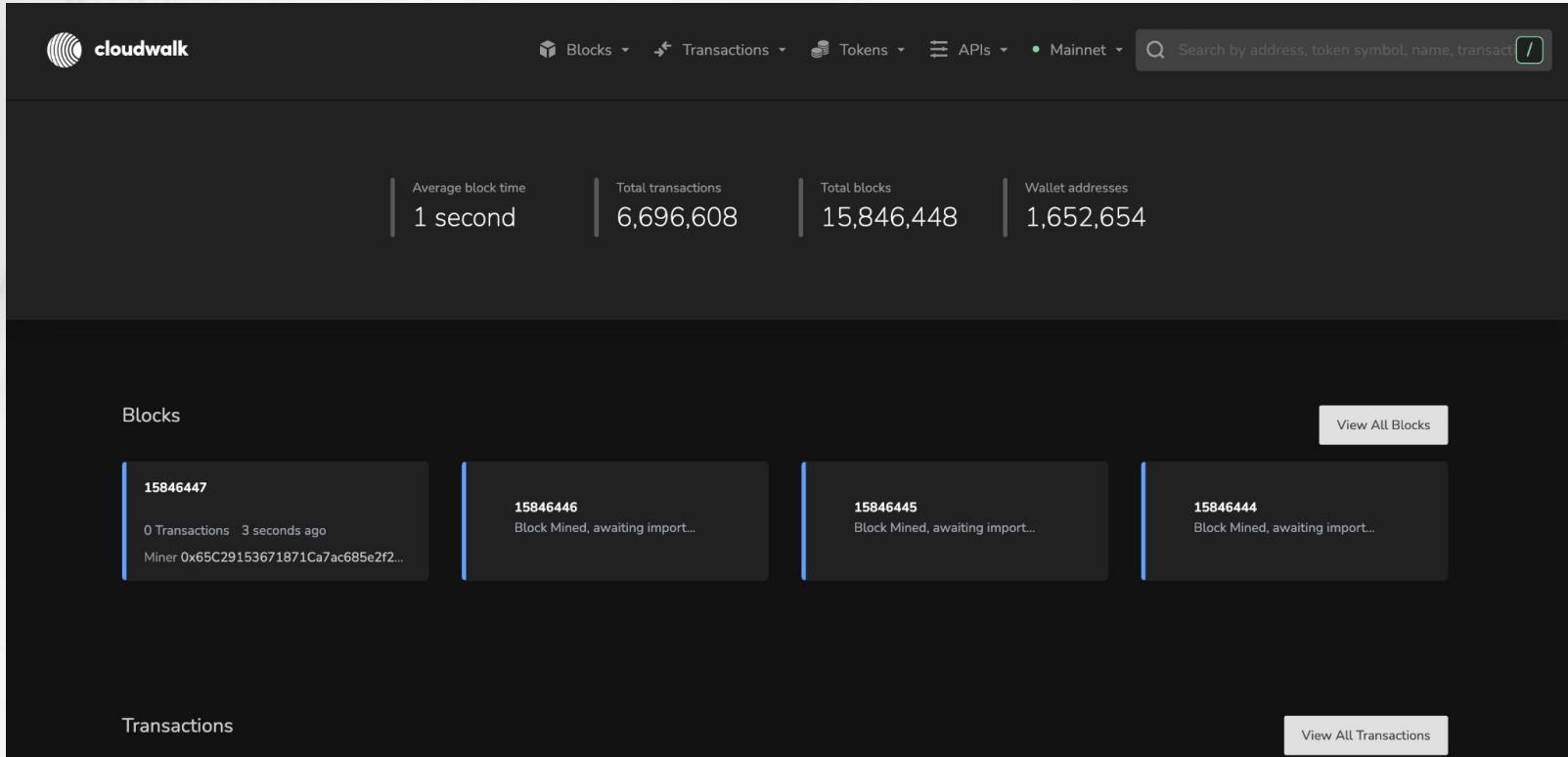
The blockchain



substrate_

The logo consists of the word "substrate" in a bold, white, sans-serif font. A large, semi-transparent circle surrounds the letters. The circle is composed of two overlapping arcs: a light blue arc on the left and a light purple arc on the right, which together form a donut-like shape. A short, horizontal teal line segment extends from the right side of the letter "e".

The blockchain



The screenshot shows the Cloudwalk blockchain explorer interface. At the top, there's a navigation bar with links for Blocks, Transactions, Tokens, APIs, Mainnet, and a search bar. Below the header, four key metrics are displayed: Average block time (1 second), Total transactions (6,696,608), Total blocks (15,846,448), and Wallet addresses (1,652,654). The main content area is divided into two sections: 'Blocks' and 'Transactions'. The 'Blocks' section shows four recent blocks with IDs 15846447, 15846446, 15846445, and 15846444. Each block card includes the number of transactions (0 for the first three, 1 for the fourth), the time since mining (3 seconds ago for the first three, awaiting import... for the fourth), and the miner's address (Miner 0x65C29153671871Ca7ac685e2f2... for the first three, awaiting import... for the fourth). A 'View All Blocks' button is located in the top right of this section. The 'Transactions' section has a 'View All Transactions' button in its top right corner.

Average block time
1 second

Total transactions
6,696,608

Total blocks
15,846,448

Wallet addresses
1,652,654

Blocks

15846447
0 Transactions 3 seconds ago
Miner 0x65C29153671871Ca7ac685e2f2...

15846446
Block Mined, awaiting import...

15846445
Block Mined, awaiting import...

15846444
Block Mined, awaiting import...

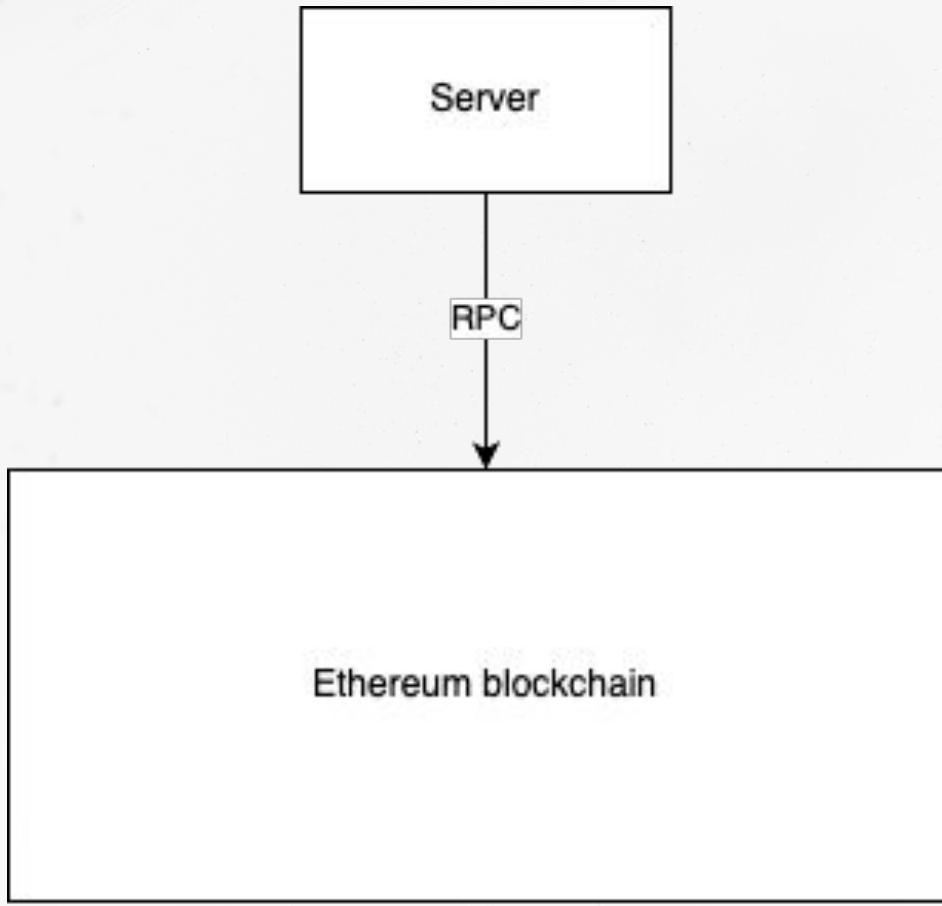
View All Blocks

Transactions

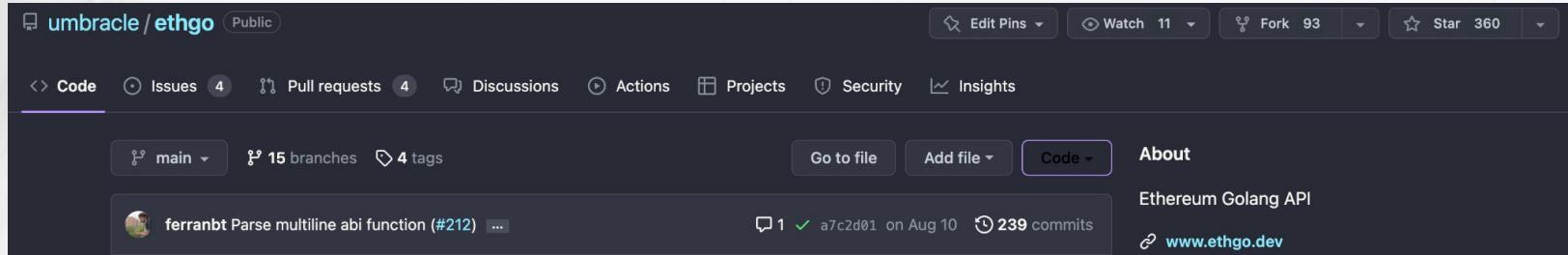
View All Transactions

<https://explorer.mainnet.cloudwalk.io/>

Communication with the blockchain



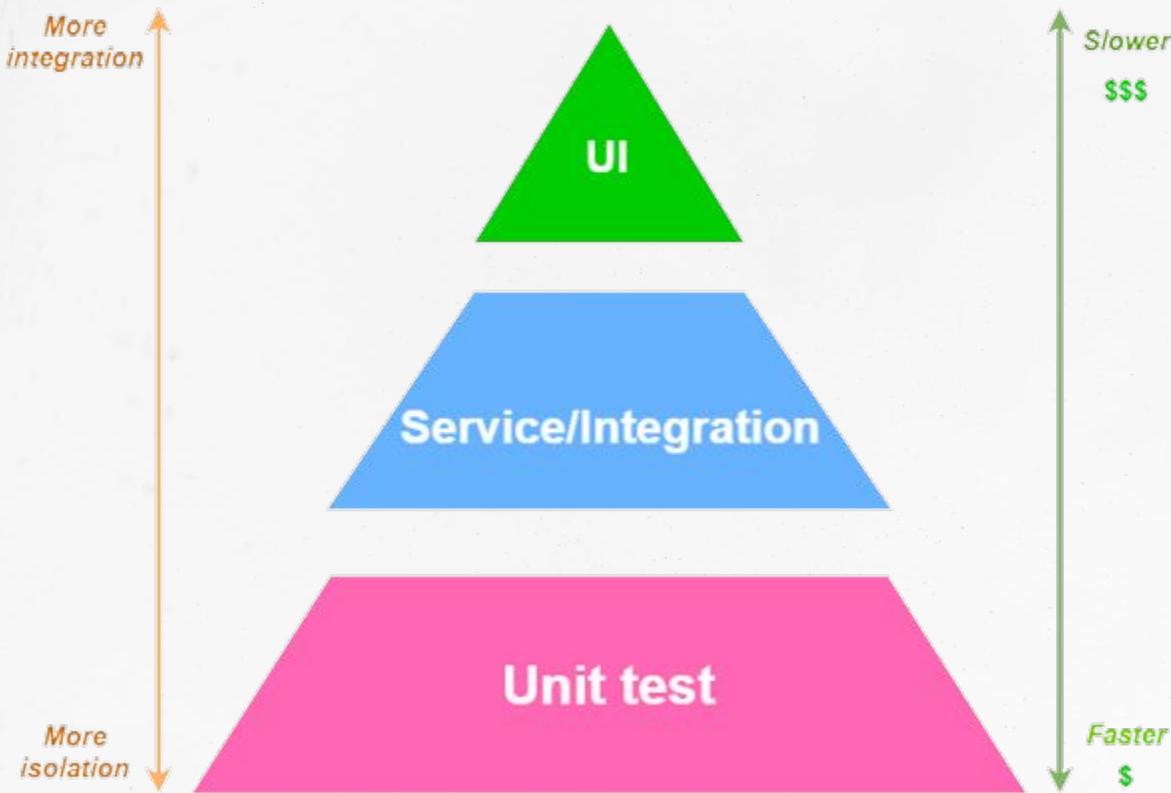
Communication with the blockchain



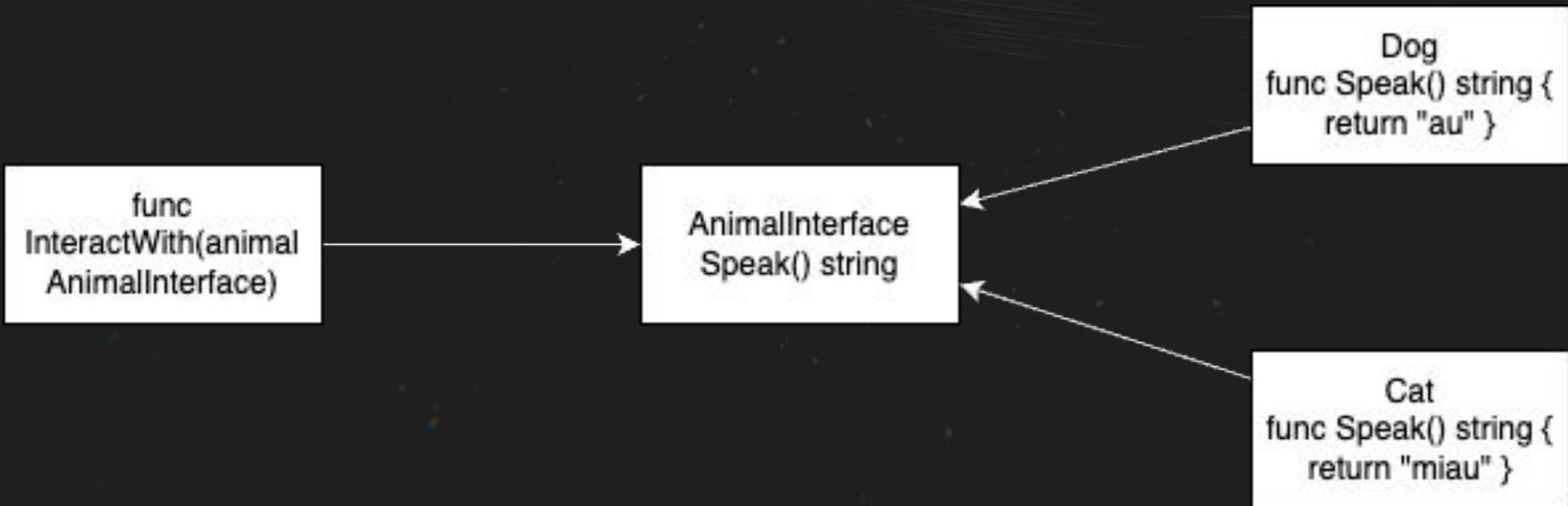
Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
- *Clients for ethereum blockchain*
 - Easy to test
 - Highly observable
 - Documentation
 - Easy to maintain

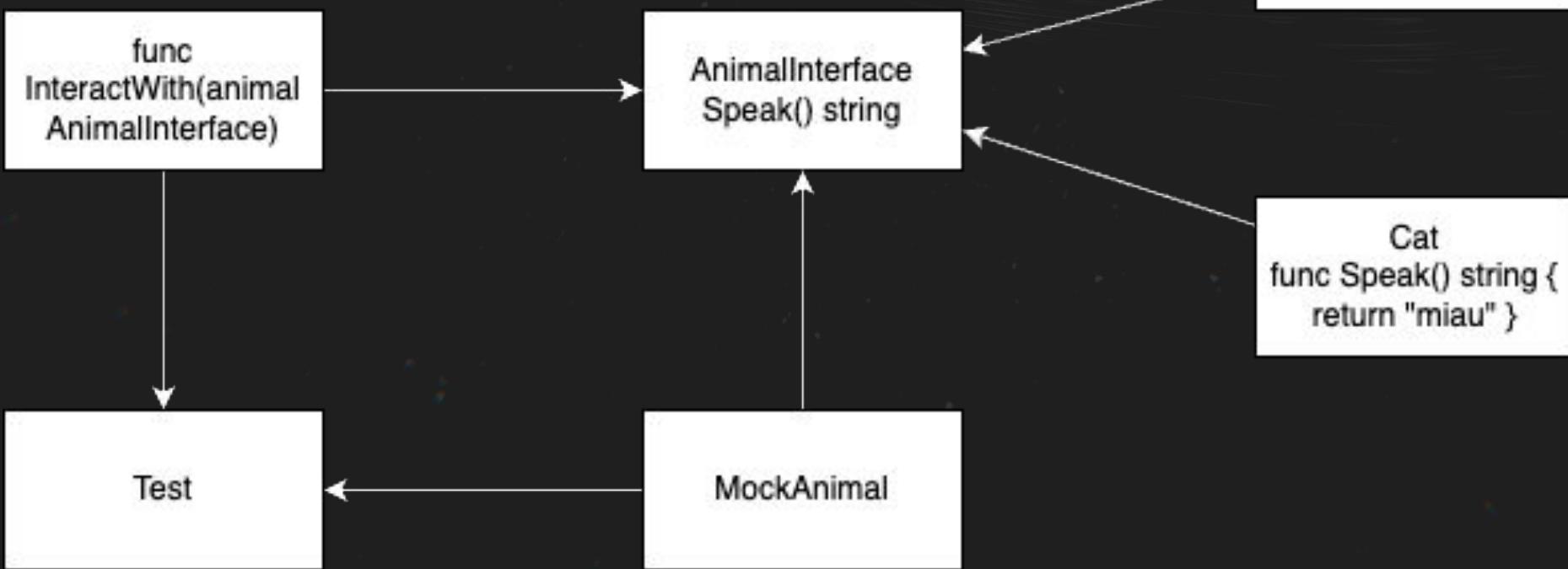
Testing



Decouple dependencies



Testing in Go



Testing in Go



```
package gopher

import (
    "testing"

    "github.com/stretchr/testify/assert"
)

func TestInteractWith(t *testing.T) {
    var (
        mock      = NewMockAnimal(t)
        expected = "test!!"
    )

    mock.On("Speak").Return(expected)
    result := InteractWith(mock)
    assert.Equal(t, expected, result)
}
```

Testing in Go

vektra / mockery Public

Code Issues 72 Pull requests 9 Discussions Actions Projects 1 Wiki Security Insights

master 2 branches 64 tags Go to file Add file Code

LandonTClipp Merge pull request #496 from ccoVeille/typos ... ✓ 546b334 11 days ago ⏲ 521 commits

.github update issue template 3 months ago

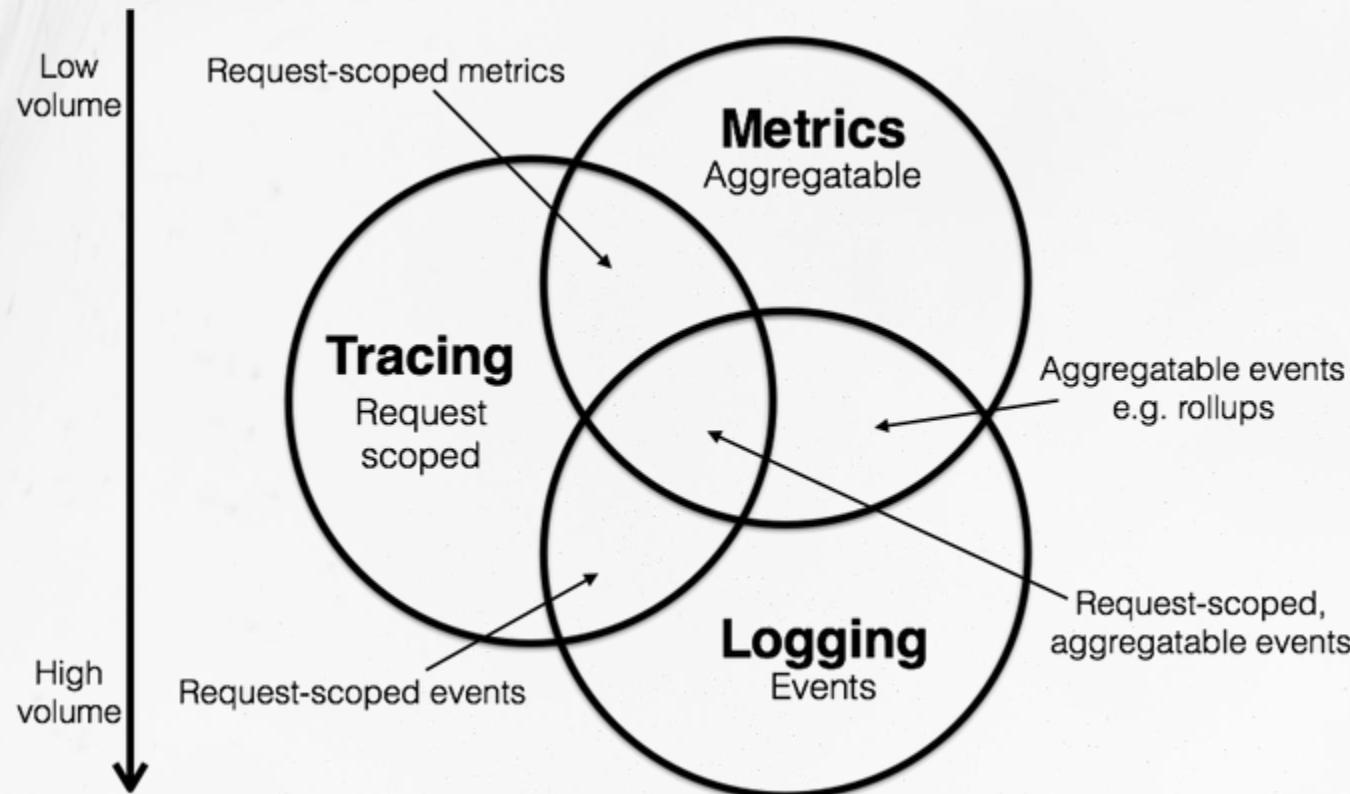
cmd fix typos and style in documentation, test and error reporting 2 months ago

About
A mock code autogenerator for Golang
testing go mock golang
generator mocking generation
mocks testify mockgen stretchr

Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
- *Clients for ethereum blockchain*
- *Easy to test*
- **Highly observable**
- **Documentation**
- **Easy to maintain**

Logs, metrics and tracing



Logs

sirupsen / logrus Public

Watch 315 Fork 2.2k Star 21.4k

Code Issues 3 Pull requests 66 Actions Projects Wiki Security Insights

master 13 branches 56 tags Go to file Add file Code About

dgsb Merge pull request #1343 from sirupsen/dbd-upd-dep ... ✓ f8bf765 on Jul 19 1,225 commits

Merge pull request #1343 from sirupsen/dbd-upd-dep

go logging logrus

uber-go / zap Public

Watch 250 Fork 1.2k Star 17.1k

Code Issues 85 Pull requests 18 Discussions Actions Security Insights

master 7 branches 34 tags Go to file Add file Code About

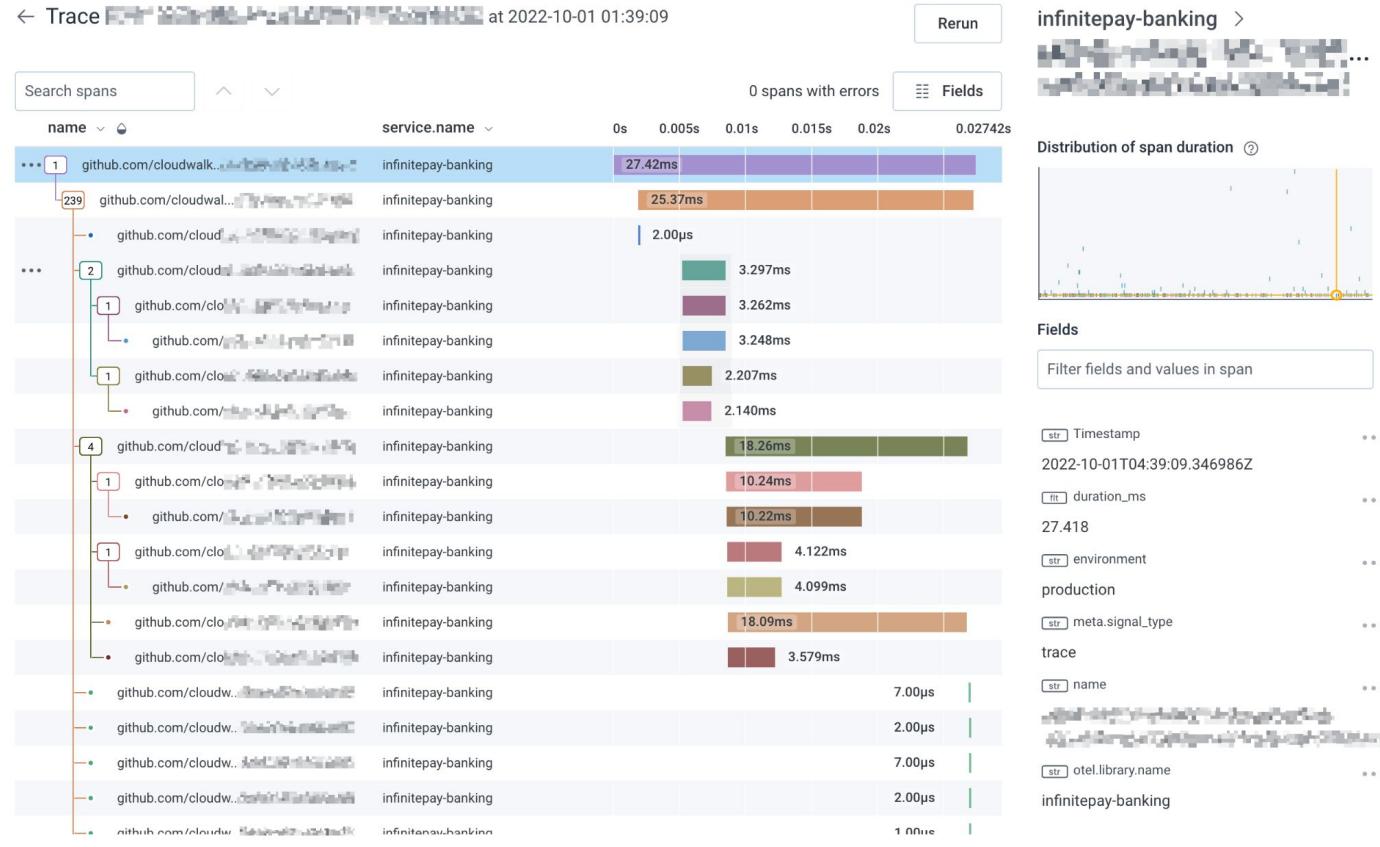
nnnkkk7 go.mod: Bump go directive to 1.19 (#1181) ✓ 9137e0e 11 days ago 577 commits

Blazing fast, structured, leveled logging in Go.

Metrics



Tracings



Tracings



```
// Trace returns the global logger pointer, a new context and a span
func Trace(ctx context.Context) (l Log, c context.Context) {
    var (
        log                  = _logger
        current, fileName, _, _ = runtime.Caller(1)
        callerName           = runtime.FuncForPC(current).Name()
        span                 Span
    )
    if ctx != nil {
        c, span = _tracer.Start(ctx, callerName)
        log = _logger.WithFields(Fields{
            "cid": span.SpanContext().TraceID(),
            "file": fileName,
        })
    }
    return &Logger{
        log: log,
        span: span,
    }, c
}
```

Tracings



```
func SomethingCool(ctx context.Context) (result string, err error)
{
    log, ctx := logger.Trace(ctx)
    defer log.End()
    log.Info("Doing something really cool!")
    ... magic ...
    return
}
```

Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
- *Clients for ethereum blockchain*
- *Easy to test*
- *Highly observable*
- Documentation
- Easy to maintain

Documentation

🔗 swag



English • 简体中文

build coverage 96% go report A+ codebeat B reference backers 8 sponsors 1
 license scan release v1.8.6

Swag converts Go annotations to Swagger Documentation 2.0. We've created a variety of plugins for popular Go web frameworks. This allows you to quickly integrate with an existing Go project (using Swagger UI).



Documentation



```
// GetAllFull godoc
// @Summary Get all statements of a given user
// @Description Return the banking statements of a given user
// @Tags statements
// @Accept json
// @Produce json
// @Param Authorization header string true "Authorization header"
// @Param limit query uint false "limit"
// @Param page_id query string false "page_id"
// @Param from_date query time.Time false "from_date"
// @Param to_date query time.Time false "to_date"
// @Param event_type query string false "event_type"
// @Security middleware.AuthJwt
// @Success 200 {object} dto.StatementsFull
// @Failure 400 {object} bankerror.BankingError
// @Failure 404 {object} bankerror.BankingError
// @Failure 500 {object} bankerror.BankingError
// @Header all {string} string "x-infinitepay-cid"
// @Router /banking/statements/full [get]
func GetAllFull(ctx *fiber.Ctx) (err error) {
    ...
}
```

Documentation

GET /banking/statements/full Get all statements of a given user ^ 

Return the banking statements of a given user

Parameters [Try it out](#)

Name	Description
Authorization * required string (header)	Authorization header <input type="text" value="token"/>
limit integer (query)	limit <input type="text" value="limit"/>
page_id string (query)	page_id <input type="text" value="page_id"/>
event_type string (query)	event_type <input type="text" value="event_type"/>

Documentation

Responses

Response content type application/json

Code	Description
200	OK

[Example Value](#) | [Model](#)

```
{  
    "balance": {  
        "brlc": 0  
    },  
    "events": [  
        {  
            "amount": 0,  
            "balance": 0,  
            "date": "2023-01-01T00:00:00Z",  
            "description": "Initial deposit",  
            "type": "deposit"  
        },  
        {  
            "amount": -100,  
            "balance": 0,  
            "date": "2023-01-01T00:00:00Z",  
            "description": "Initial withdrawal",  
            "type": "withdrawal"  
        },  
        {  
            "amount": 50,  
            "balance": 0,  
            "date": "2023-01-01T00:00:00Z",  
            "description": "Interest accrued",  
            "type": "interest"  
        }  
    ],  
    "pagination": {  
        "next": "string",  
        "previous": "string"  
    }  
}
```

Headers:

Name

Description

Type

string

x-infinitepay-cid

string

Documentation

400

Bad Request

[Example Value](#) | [Model](#)

```
{  
  "code": "string",  
  "kind": 0,  
  "message": "string"  
}
```

Headers:

Name	Description	Type
string	x-infinitepay-cid	string

404

Not Found

[Example Value](#) | [Model](#)

```
{  
  "code": "string",  
  "kind": 0,  
  "message": "string"  
}
```

Headers:

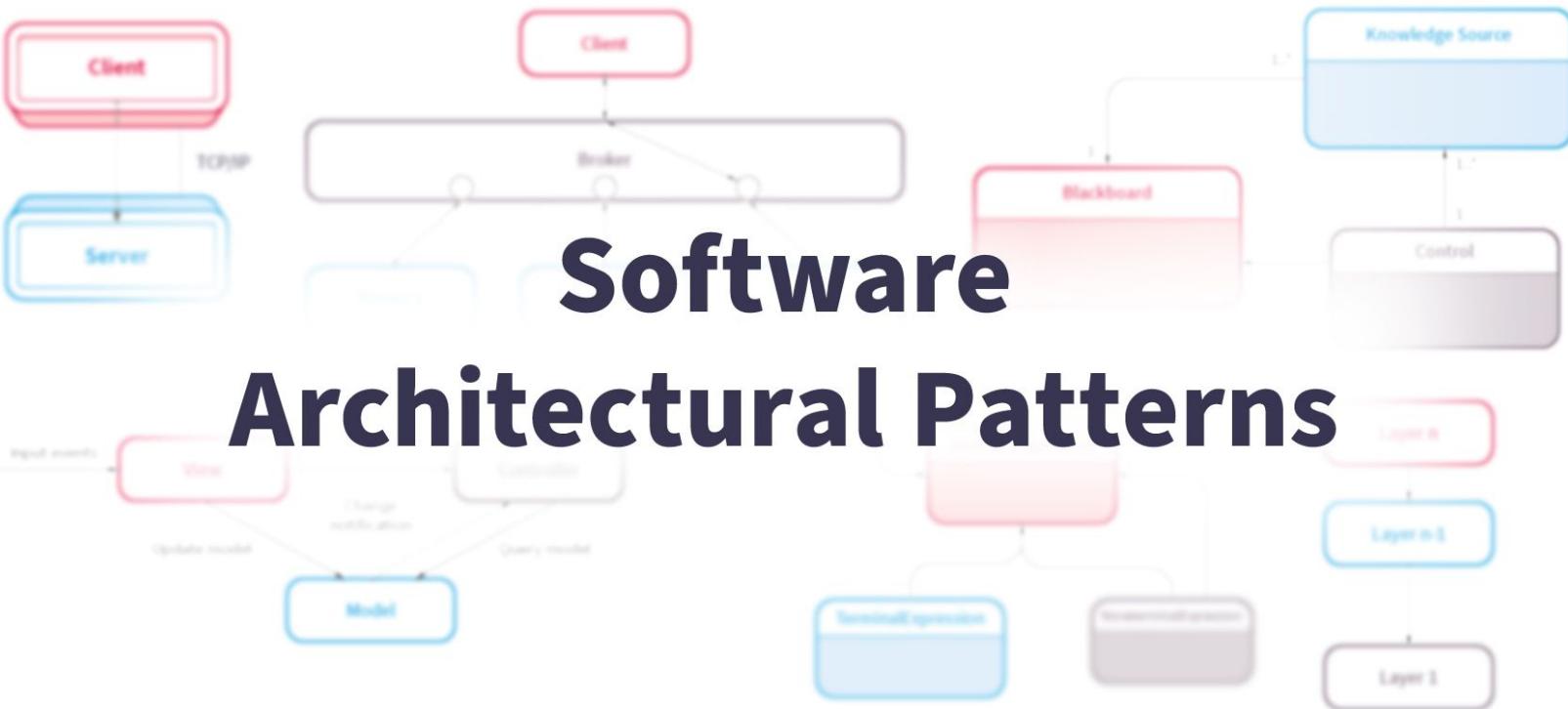
Name	Description	Type
string	x-infinitepay-cid	string

Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
- *Clients for ethereum blockchain*
- *Easy to test*
- *Highly observable*
- *Documentation*
- *Easy to maintain*

Architecture

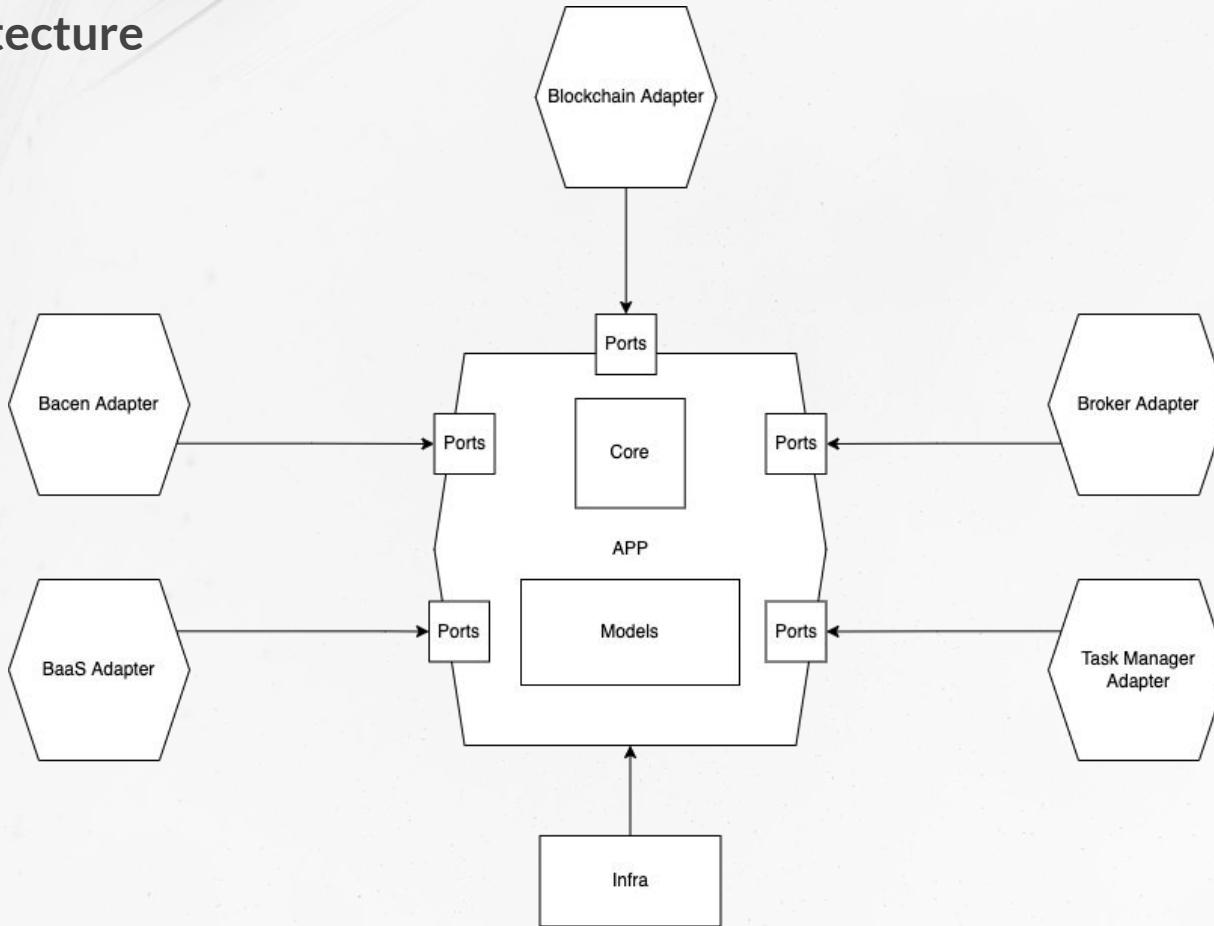
Software Architectural Patterns



Architecture



Architecture



Architecture

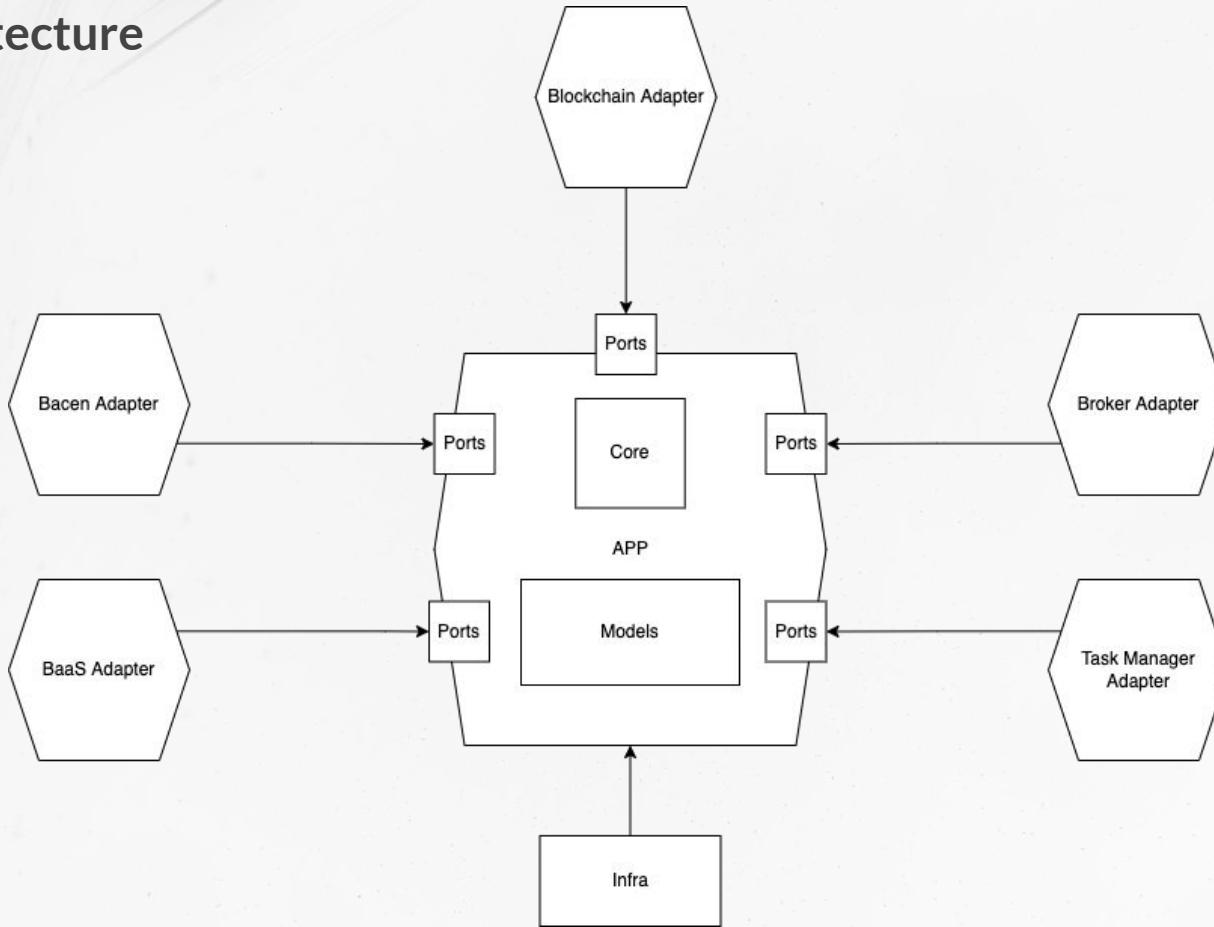


```
func Start(server *server.Server, moduleList ...modules.Module) (core *Core, err error)
{ ...
}
```



```
type Module interface {
    Name() string
    Start(core *CoreDeps) error
    GetContext(ctx context.Context, transactionHash ...string) statement.ContextMap
    GetContextFull(ctx context.Context, wallet repository.Wallet, indexRef string, limit uint, fromDate, toDate
time.Time) statement.ContextMap
}
```

Architecture



Abstractions



Abstractions



```
type RepositoryI[M Model] interface {
    CreateOne(ctx context.Context, registry M) (M, error)
    Create(ctx context.Context, registry ...M) error
    SaveOne(ctx context.Context, registry M) (M, error)
    Save(ctx context.Context, registry M) error
    FindWithoutErrorIfEmpty(ctx context.Context, query any, args ...any) (M,
        error)
    Find(ctx context.Context, query any, args ...any) (M, error)
    FindBy(ctx context.Context, column string, value any) (M, error)
    Preload(query string, args ...any) (tx RepositoryI[M])
    Select(query any, args ...any) (tx RepositoryI[M])
    Where(query any, args ...any) (tx RepositoryI[M])
    Or(query any, args ...any) (tx RepositoryI[M])
    Not(query any, args ...any) (tx RepositoryI[M])
    Limit(limit int) (tx RepositoryI[M])
    Order(value any) (tx RepositoryI[M])
    OrderBy(rule string) (tx RepositoryI[M])
    GroupBy(param string) (tx RepositoryI[M])
    Join(query string, args ...any) (tx RepositoryI[M])
    Unscoped() (tx RepositoryI[M])
    Transaction(fc Transaction[M], opts ...*sql.TxOptions) error
    First(ctx context.Context) (result M, err error)
    Last(ctx context.Context) (result M, err error)
    Scan(ctx context.Context) (result []M, err error)
    To(ctx context.Context, dest any) (err error)
    Update(ctx context.Context, newObject M) (err error)
    UpdateGroup(ctx context.Context, column string, value any, args ...any)
    Delete(ctx context.Context, query any, args ...any) error
    FieldExist(field string) bool
}
```

Abstractions



```
type Repository[M Model] struct {
    conf config.Config
    db   *gorm.DB
}

func (r Repository[M]) CreateOne(ctx context.Context, registry M) (result M, err error)
{
    var model M
    if err = r.db.WithContext(ctx).Model(model).Create(&registry).Error; err != nil {
        return result, translateError(err)
    }
    return registry, nil
}
```

Abstractions



```
type UserRepository relational.RepositoryI[User]
```

```
type User struct {
    ID          int64 `gorm:"primarykey"`
    Role        string
    PhoneNumber string
    Email       string
    Handle      string
    CreatedAt   time.Time
    UpdatedAt   time.Time
}
```

Abstractions



```
userRepository := relational.NewRepository[repository.User](dbConn)

if user, err = userRepository.FindBy(ctx.UserContext(), "id", payload.UserID); err != nil {
    return err
}
```

"It's Programming if "clever" is a compliment.
It's Software Engineering if "clever" is an accusation."

WINTERS, Titus - Software Engineering at Google: Lessons Learned from Programming Over Time

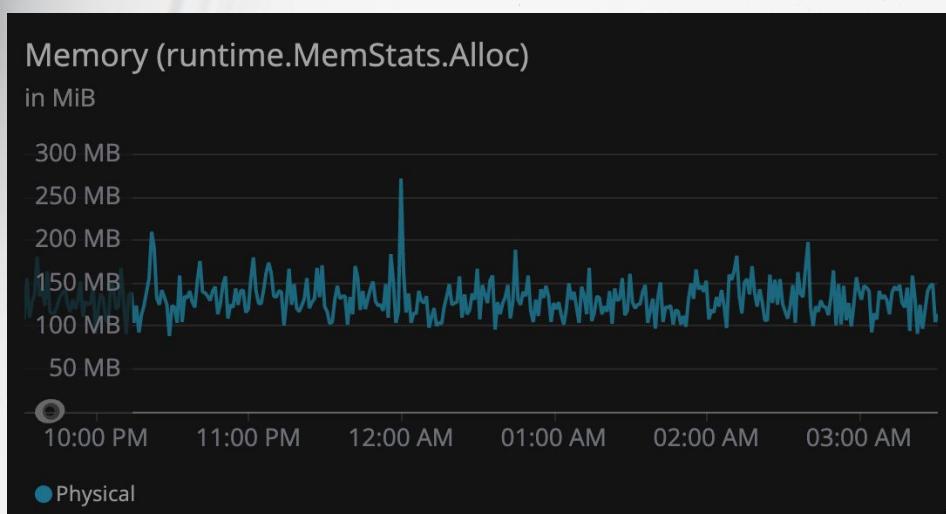
Challenges:

- *Handling multiple databases and persistent sockets in order to merge operations with its given contexts.*
- *Development and environment consistent*
- *Clients for ethereum blockchain*
- *Easy to test*
- *Highly observable*
- *Documentation*
- *Easy to maintain*

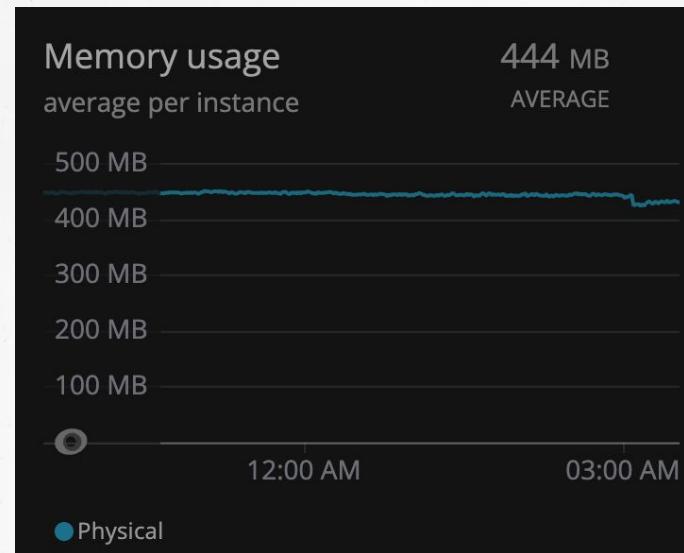
Performance

Name	Response time (ms)
★ ● infinitepay-banking	40 ms
★ ● infinitepay-banking POC	120 ms

Performance



New Golang



POC Ruby

Result

Thanks, y'all! Questions?

Sorteio da capivara



Nós vamos dar 100 de BRLC para quem twittar hoje,
aqui da GopherCon Brasil!

"Me dá meus 100 conto na tag #InfinitePay <MINHA TAG>
#GopherCon"