Scaling using Go

Luno’s journey

Adam Hicks
Staff Engineer

Who are Luno?

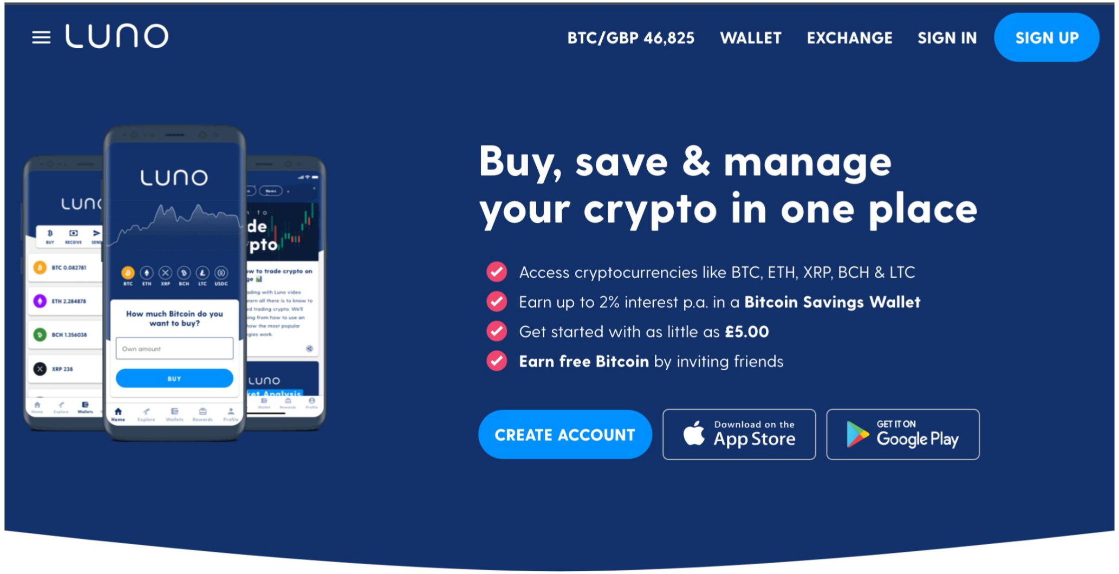
Cryptocurrency Wallet

9 Million Customers

Worldwide

500+ Employees

70 Go Engineers



Buy, save & manage your crypto in one place

- Access cryptocurrencies like BTC, ETH, XRP, BCH & LTC
- Earn up to 2% interest p.a. in a **Bitcoin Savings Wallet**
- Get started with as little as **£5.00**
- **Earn free Bitcoin** by inviting friends

CREATE ACCOUNT
Download on the App Store
GET IT ON Google Play
How much Bitcoin do you want to buy?
BUY

LUNO

Our Challenges

Banks

Legacy “API”
Robustness?

Huh, what does
breaking change mean?

Our Challenges

Blockchains

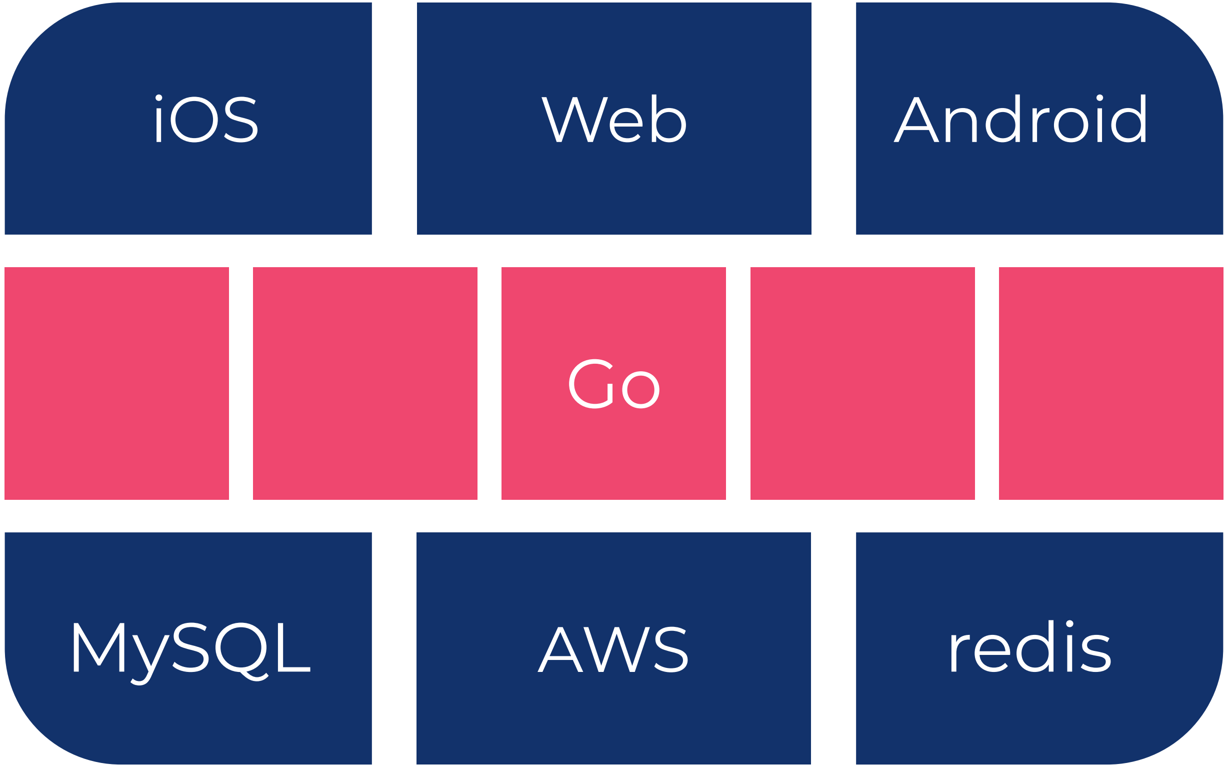
KYC

Regulation

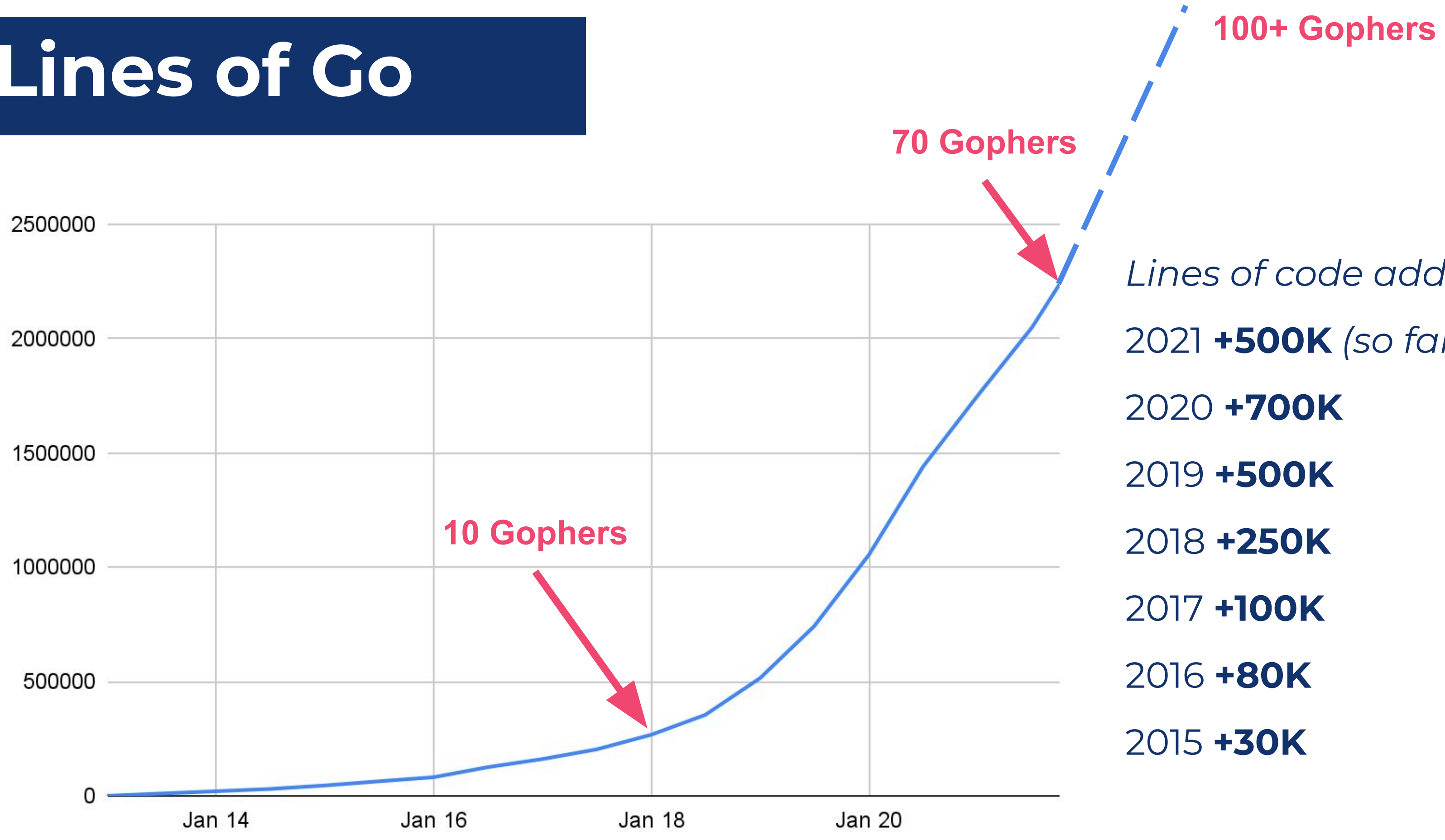
Fraud

LUNO

Architecture

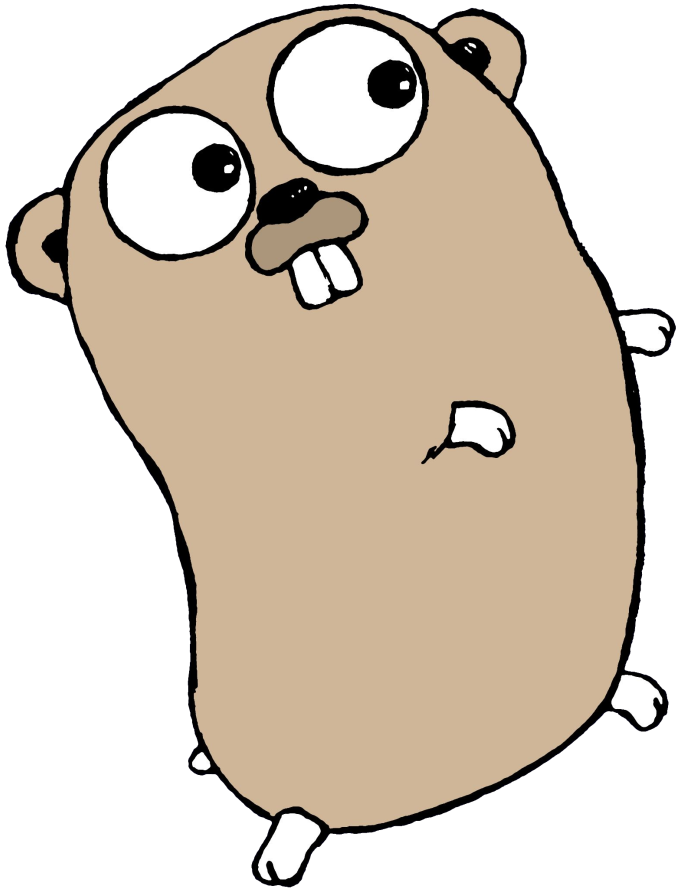


Lines of Go



Lines of code added

2021	**+500K** *(so far)*
2020	**+700K**
2019	**+500K**
2018	**+250K**
2017	**+100K**
2016	**+80K**
2015	**+30K**



Go Solves These Things

slow builds

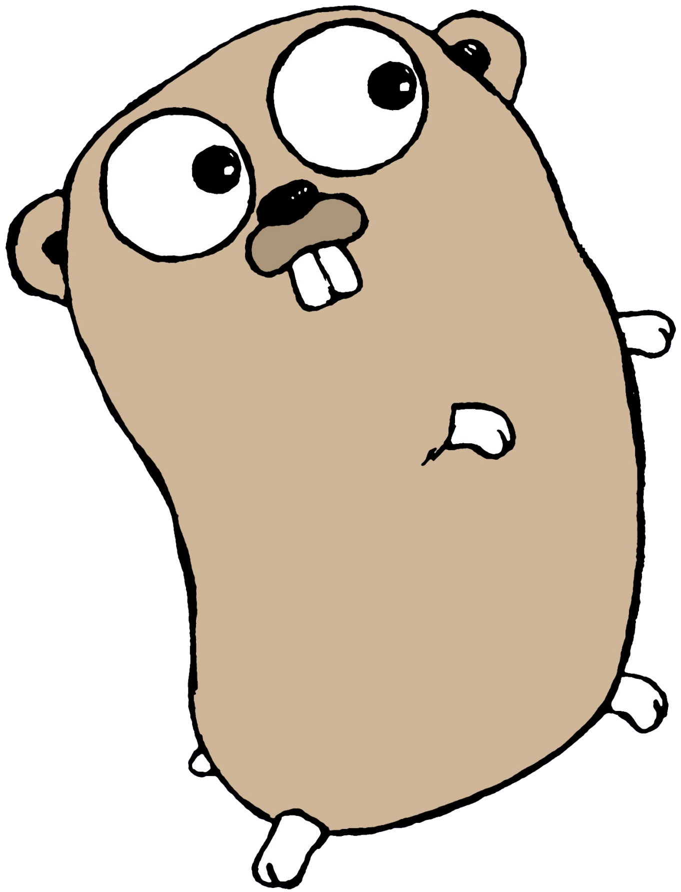
uncontrolled dependencies

each programmer using a different subset of the language

poor program understanding (code hard to read, poorly documented, and so on)

duplication of effort

difficulty of writing automatic tools



Simplicity = Scalability

System Complexity ⬆️

Codebase Complexity ➡️

Idiomatic Go

How can we **hire** more developers?

Easy to learn

Onboard new devs **quickly**

Learn Go

Learn Luno patterns

Onboard	1 week
Training Exercise	2 - 4 weeks
Smaller Tasks	0 - 2 weeks
Performing	

Idiomatic Go

How can we **hire** more developers?

Go is **easy** to read

All code looks **familiar**

Reviews are **quicker**

Devs leaving has **less** impact

Devs can **switch** teams

Don't make me think

LUNO

APIs

We need **microservices** but how can they **communicate**?

gRPC

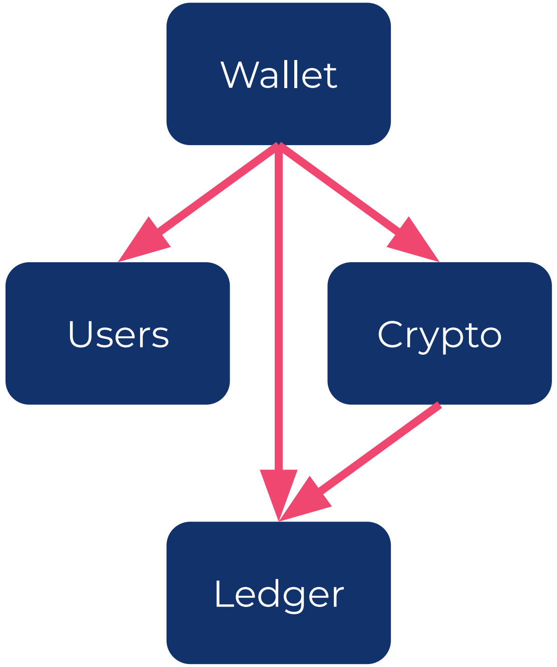
Low latency

Powerful **protobuf** definitions

Upgrade with zero downtime

Teams **own** the interface

Platform wide **monitoring**



LUNO

APIs

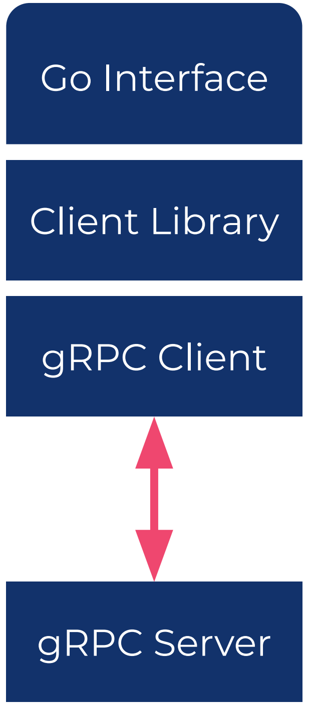
We need **microservices** but how can they **communicate**?

Client packages

Popular Netflix architecture

Easy for developers to read

Common access patterns



LUNO

APIs

We need **microservices** but how can they **communicate**?

Event Driven

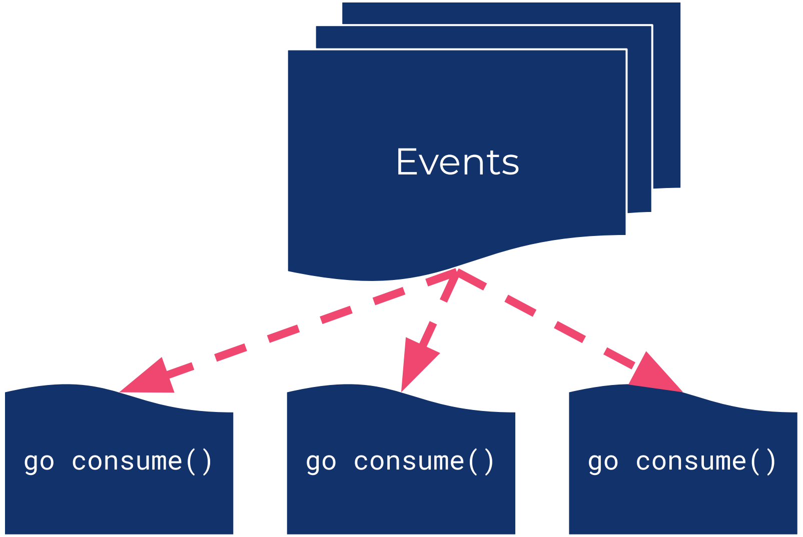
Asynchronous processing

Scaleable consumers

Goroutines for **performance**

Open sourced library

github.com/luno/reflex



//go:generate

“Let the machine do the work” - Rob Pike

Go Generate

SQL Row <> Go Struct

Simple for devs to write

Standard access patterns

Safe from common mistakes

100+ tables

Millions of queries

Everyone is writing the **same** database code!

```
//go:generate glean -table=users

type User struct {
     ID        int64     `glean:"user_id"`
     Name      string    `glean:"name"`
     Type      UserType  `glean:"type"`
     CreatedAt time.Time `glean:"created_at"`
}


Type glean struct {
     User
     Name NullString
}
```

Go Generate

We’re writing the same **serialisation** code over and over!

Protobuf Message <> Go Struct

Faster than reflection

Backwards **compatible**

Less code 👍

100s of message types

```
//go:generate protocp Wallet

type Wallet struct {
     ID         int64      `protocp:"1"`
     Currency   string     `protocp:"2"`
     Balance    Decimal    `protocp:"3"`
     Available Decimal     `protocp:"4,7"`
}


func WalletToProto(w Wallet) proto.Wallet {...
```

LUNO

Go Generate

And...

Finite State Machine + SQL Tables

Database Migrations

Templated Code Structure

Mock Interfaces

Dependency Injection

Linting

We're wasting **time** in code **reviews** on the same issues!

Commit Hooks

Runs on git pre-commit

Checks changed packages

Errors if rules are broken

Rules are defined using Go Code

Fewer comments on reviews

Code standards maintained

e.g.

Return errors

Log message formats

Standardise package usage

- `.Assert()` from multiple packages
- `import "errors"` vs `"jettison/errors"`

Standard tools

gofmt

Luno is scaling...

to the Moon!

LUNO

Future

Starting a team for **developer** tooling

Go Generics

We're hiring!

LUNO

github.com/**luno**

github.com/**adamhicks**

Thank you