Introduction to Watermill

let’s build event-driven application in 25 minutes

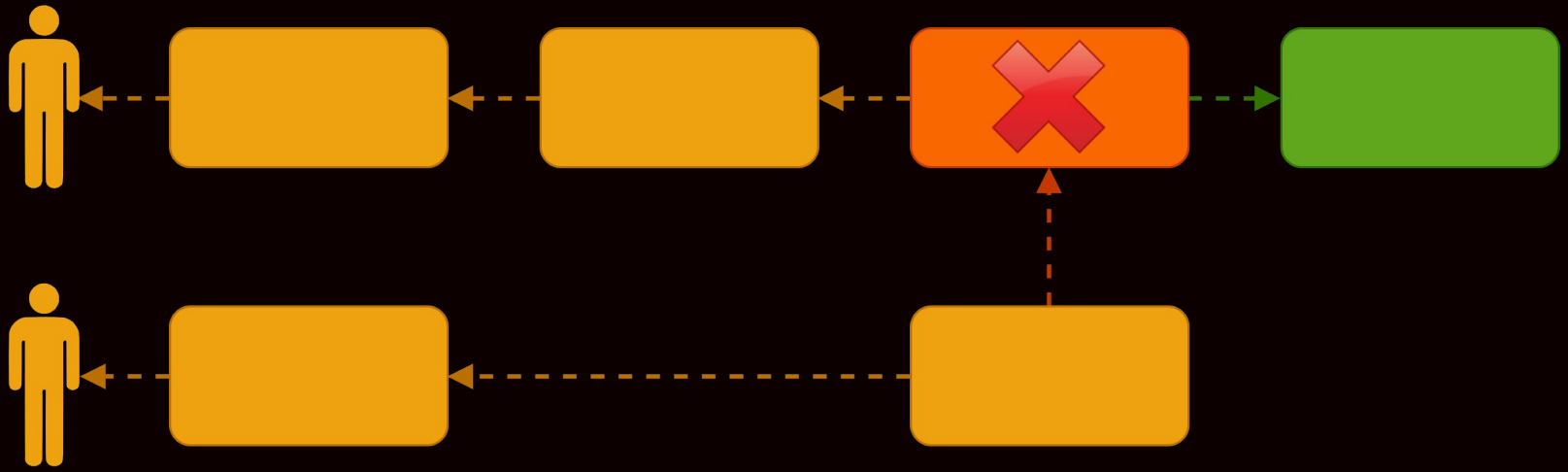
Golang Poland 25.06.2020

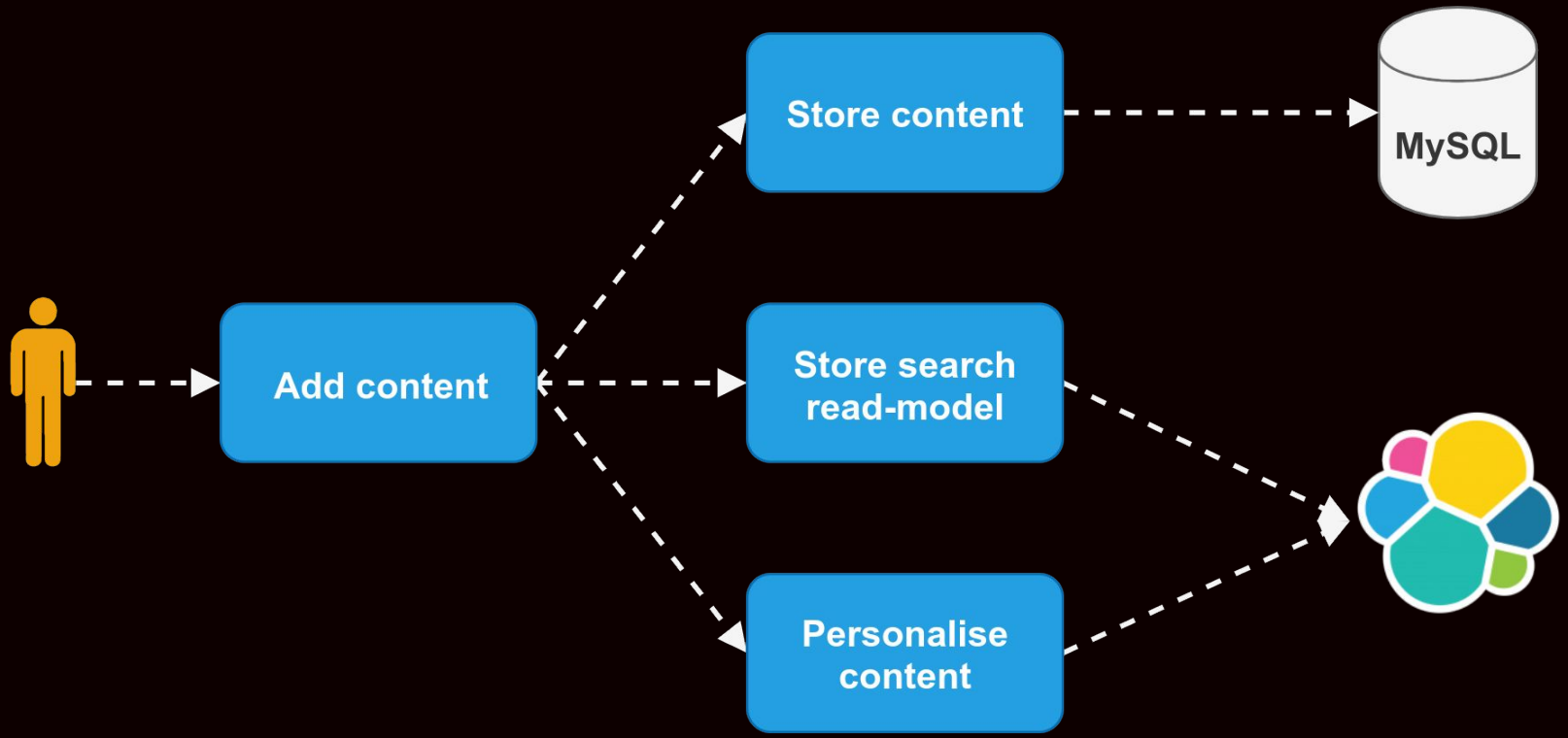
roblaszczak

About me

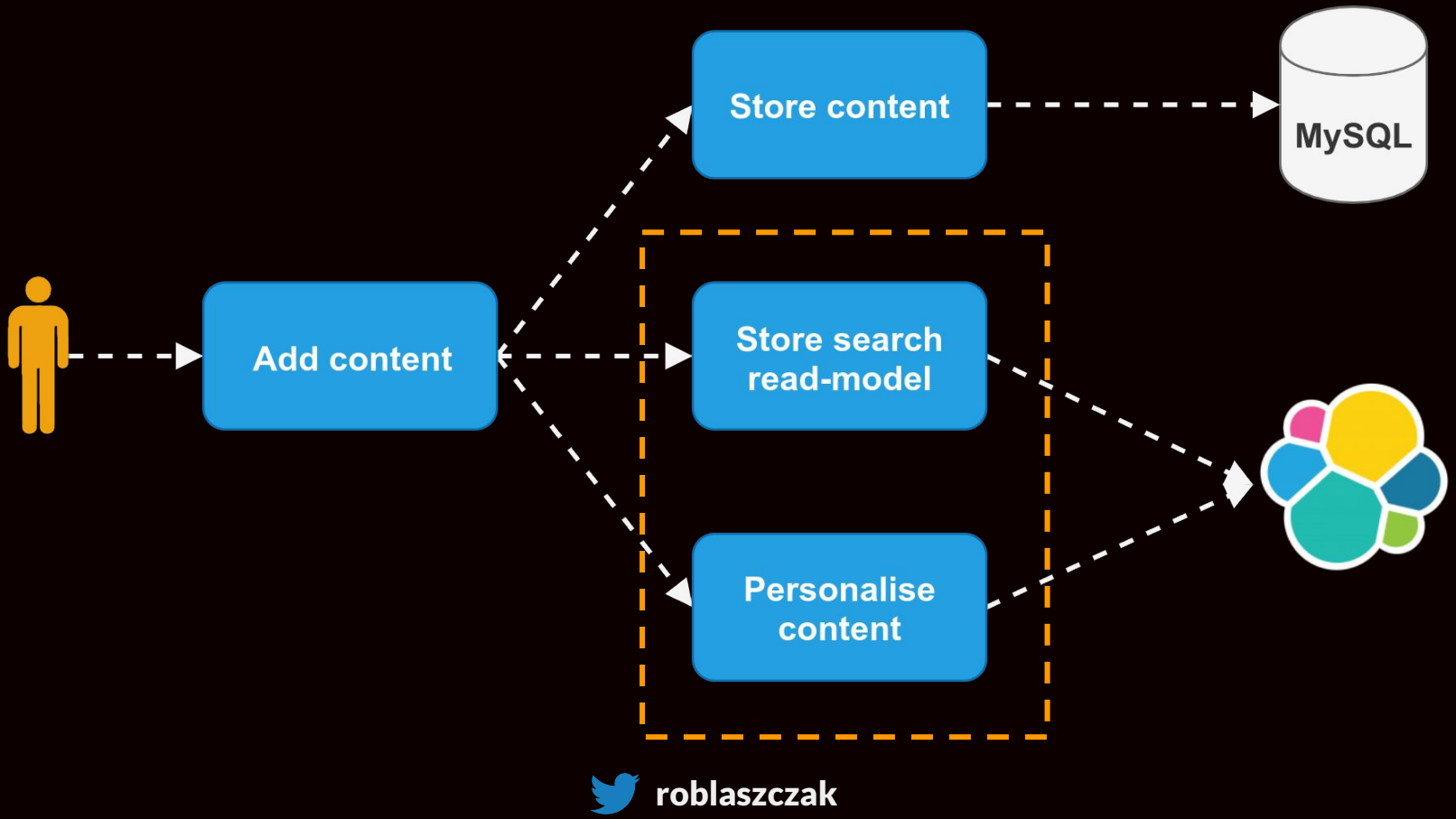
- I’m Robert ;-)
- Tech Lead @KARHOO
- Blogging at **threedots.tech**
- **@roblaszczak at**

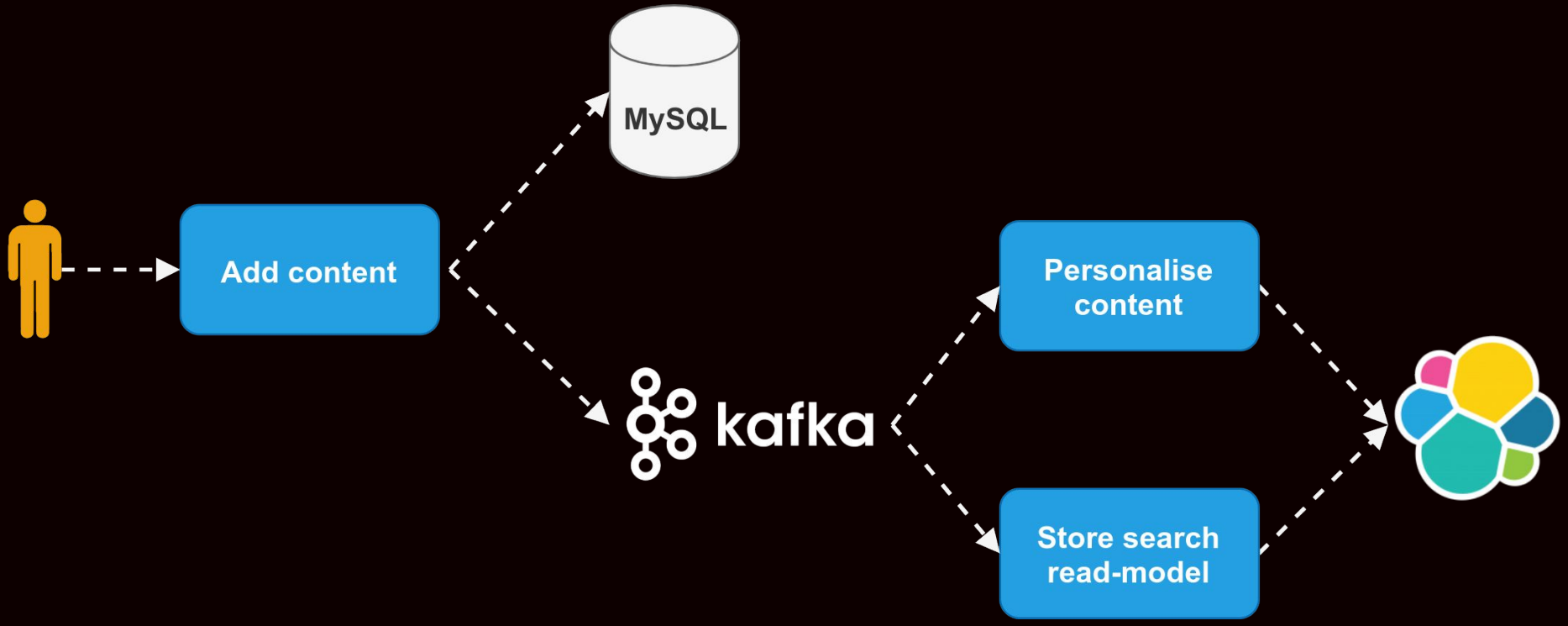
Cascading failures





roblaszczak


Store content
MySQL
Add content
Store search read-model
Personalise content
roblaszczak


MySQL
Add content
kafka
Personalise content
Store search read-model

PROBLEM?

Building applications based on Kafka is not easy

- Consumer groups
- Partitioning
- At-least-once delivery
- Message Ack and Nack support
- Not losing any message

BUILDING EVENT-DRIVEN APPLICATIONS IS NOT EASY

How to make building event-driven applications easy in Go?

Watermill

2k

GitHub Stars

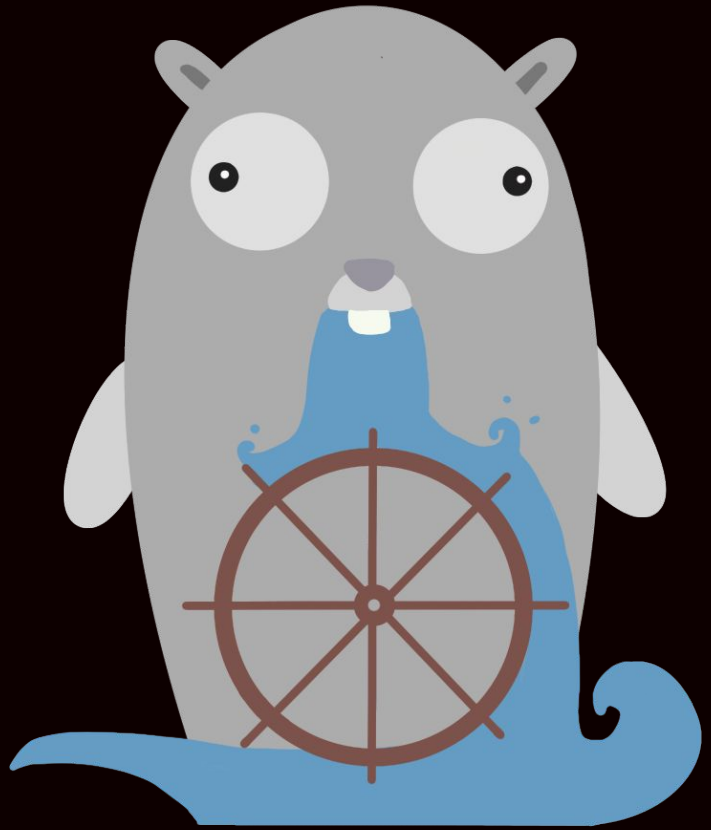
roblaszczak

20

Contributors

8

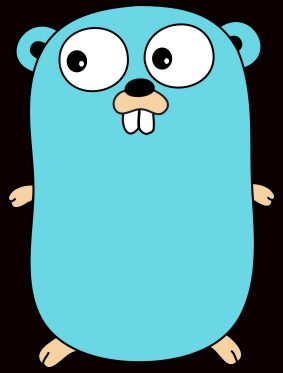
Supported Pub/Subs



roblaszczak

Watermill is not a framework

Watermill is a library



kafka

NATS

RabbitMQ

MySQL™

roblaszczak

HOW?

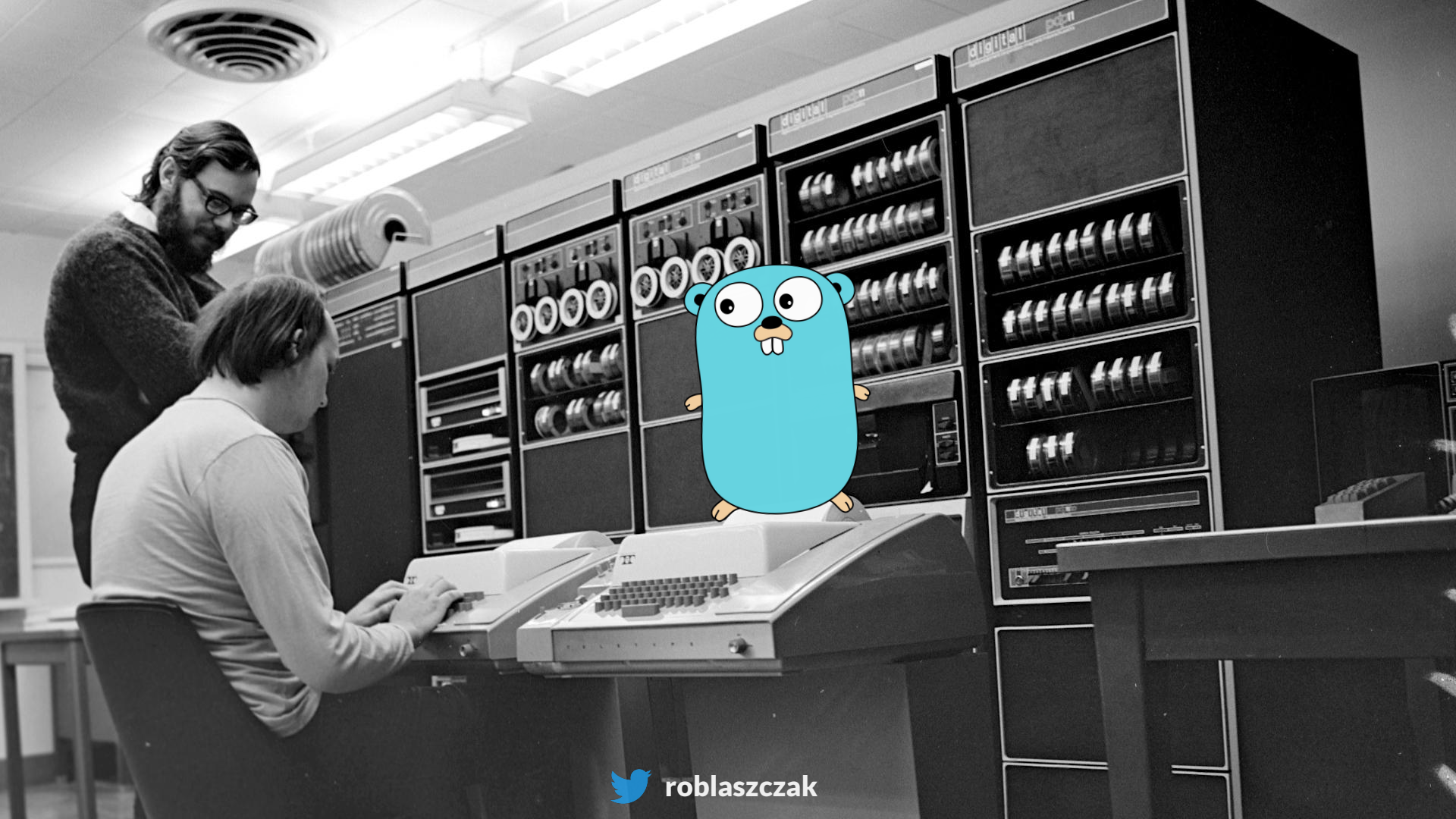
Unix philosophy (1978)

- Write programs that **do one thing and do it well**.
- Write programs to **work together**.
- Write programs to handle ~~text streams~~ **message, because that is a universal interface**.

```go
type Publisher interface {

    Publish(topic string, messages ...*Message) error

    Close() error

}

type Subscriber interface {

    Subscribe(ctx context.Context, topic string) (<-chan *Message, error)

    Close() error

}
```

```
type HandlerFunc func(msg *Message) ([]*Message, error)
```


roblaszczak


roblaszczak


roblaszczak

The Pragmatic Programmer

from journeyman
to master

Andrew Hunt
David Thomas

Foreword by Ward Cunningham

Testing

```go
func TestPublishSubscribe(t *testing.T) {
    features := tests.Features{
        ConsumerGroups:      true,
        ExactlyOnceDelivery: false,
        GuaranteedOrder:     false,
        Persistent:          true,
    }

    tests.TestPubSub(
        t,
        features,
        createPubSub,
        createPubSubWithConsumerGrup,
    )
}
```

```go
var stressTestTestsCount = 10

func TestPubSubStressTest(
    t *testing.T,
    features Features,
    pubSubConstructor PubSubConstructor,
    consumerGroupPubSubConstructor ConsumerGroupPubSubConstructor,
) {
    for i := 0; i < stressTestTestsCount; i++ {
        t.Run(fmt.Sprintf("%d", i), func(t *testing.T) {
            t.Parallel()
            TestPubSub(t, features, pubSubConstructor, consumerGroupConstructor)
        })
    }
}
```


roblaszczak

How fast is Watermill?

Pub/Sub	Publish (messages / s)*	Subscribe (messages / s)*
Kafka	70,252	117,529
NATS	76,208	38,169
SQL (MySQL)	6,989	143
Google Cloud Pub/Sub	7,416	39,591
AMQP	2,408	10,608

*1 instance, 1 process, Docker Compose

Pub/Sub	Publish (messages / s)	Subscribe (messages / s)
Kafka	70,252	117,529
NATS	76,208	38,169
SQL (MySQL)	6,989	143
Google Cloud Pub/Sub	7,416	39,591
AMQP	2,408	10,608

Pub/Sub	Publish (messages / s)	Subscribe (messages / s)
Kafka	70,252	117,529
NATS	76,208	38,169
SQL (MySQL)	6,989	143
Google Cloud Pub/Sub	7,416	39,591
AMQP	2,408	10,608

Pub/Sub	Publish (messages / s)	Subscribe (messages / s)
Kafka	70,252	117,529
NATS	76,208	38,169
SQL (MySQL)	6,989	143
Google Cloud Pub/Sub	7,416	39,591
AMQP	2,408	10,608

The first rule of live coding

Don’t do livecoding

The first rule of live coding

roblaszczak

Not covered

- It’s hard to show everything in 25 minutes :)
- Kafka and Google Cloud Pub/Sub internals
- At-least-once delivery
- CQRS component
- Middlewares

roblaszczak

Documentation

Getting started

Watermill up and running

Message

Message is one of core parts of Watermill

Pub/Sub

Publishers and Subscribers

Message Router

The Magic Glue of Watermill

CQRS Component

Command Query Responsibility Segregation (CQRS) Component

Implementing custom Pub/Sub

Bring Your Own Pub/Sub

Metrics

Monitor Watermill in realtime

Middlewares

Add generic functionalities to your handlers in an inobtrusive way

Troubleshooting

When something goes wrong

Branch: **master** ▾ **watermill** / _examples / **basic** /

Create new file | Upload files | Find file | History

 **maclav3** and **m110** Stylistic improvements (#139) ··· Latest commit `3348680` on 24 Sep

..

1-your-first-app	Bump examples to watermill-kafka v2.0.0 (#132)	last month
2-realtime-feed	Bump examples to watermill-kafka v2.0.0 (#132)	last month
3-router	Stylistic improvements (#139)	last month
4-metrics	Group examples (#137)	last month
5-cqrs-protobuf	Group examples (#137)	last month

Thanks for all contributors!

roblaszczak

Building Business applications by example

- DDD, CQRS, Clean/Hexagonal Architecture/Event Storming
- Focused on solving **real issues** instead of blindly applying patterns
- Done by **refactoring** a real Go project
- **DDD lite article - next week!**

THANKS!

https://threedots.tech/

https://watermill.io/

roblaszczak

Photo by Micheile Henderson @micheile010 // Visual Stories [nl] on Unsplash

Photo by Thomas Ehling on Unsplash