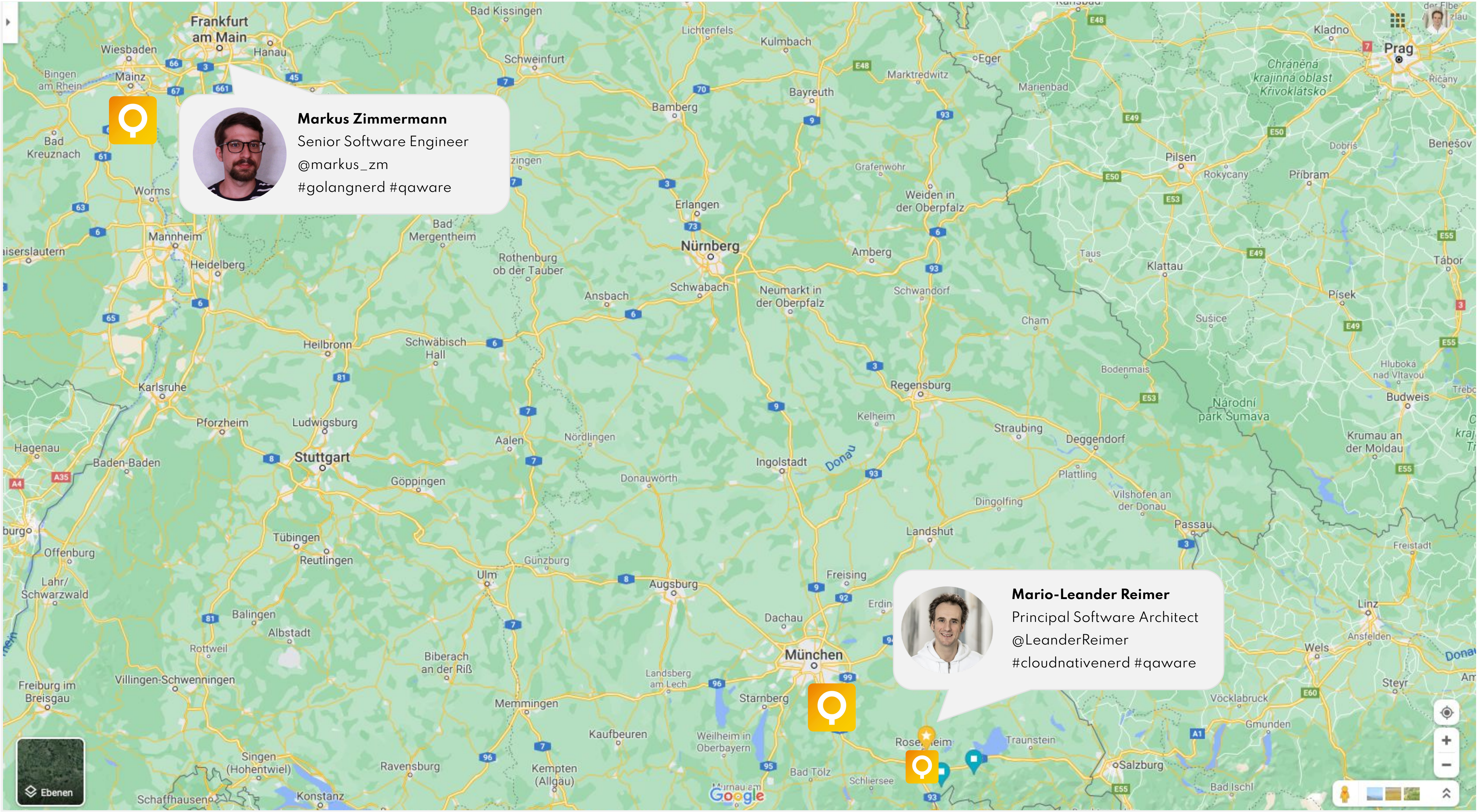
qaware.de

QA|WARE
SOFTWARE ENGINEERING

betterCode() Workshop

Effizientes DevOps-Tooling mit Go

Mario-Leander Reimer, @LeanderReimer
Markus Zimmermann, @markus_zm


Markus Zimmermann
Senior Software Engineer
@markus_zm
#golangnerd #qaware
Mario-Leander Reimer
Principal Software Architect
@LeanderReimer
#cloudnativenerd #qaware

How do you organise and enable DevOps teams for fast flow and high productivity?

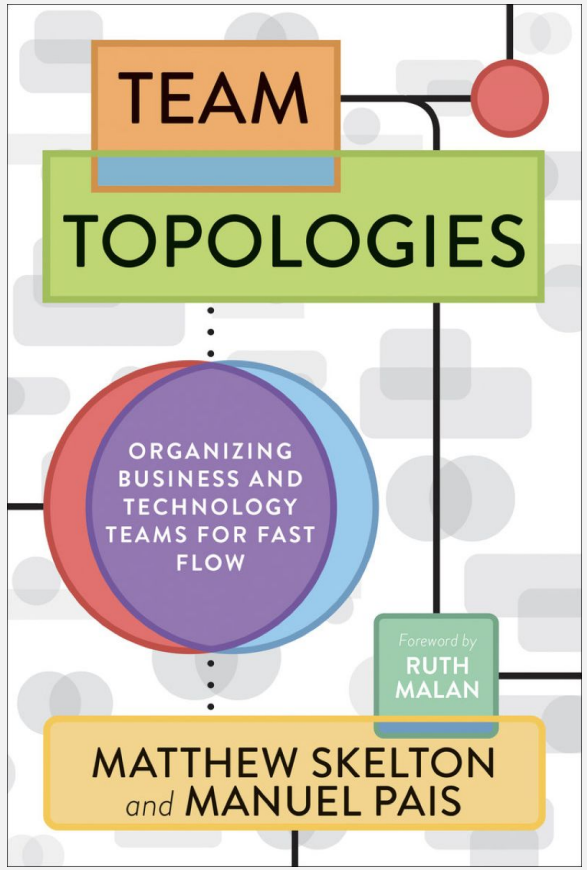
Too much cognitive load will become a bottleneck for fast flow and high productivity.

QA|WARE

- **Intrinsic Cognitive Load**
  Relates to fundamental aspects and knowledge in the problem space (e.g. used languages, APIs, frameworks)

- **Extraneous Cognitive Load**
  Relates to the environment
  (e.g. deployment, configuration, console commands)

- **Germane Cognitive Load**
  Relates to specific aspects of the business domain
  (aka. „value added“ thinking)



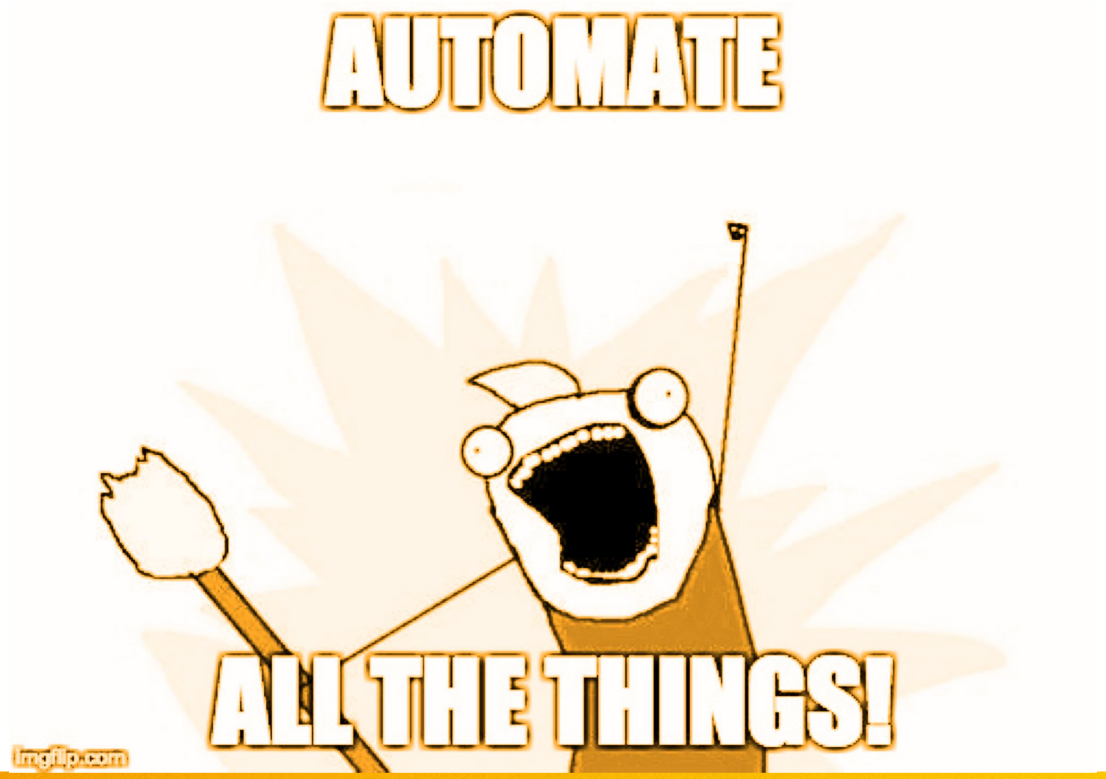
Minimize

intrinsic cognitive load

Eliminate

extraneous cognitive load

QA|WARE



Use the right tool for the job!

Ruby Python GO Ansible Bash

Why Go?

- Go is open source and maintained by Google
- Go is an efficient distributed, parallel language for systems programming at Google to solve problems of C++ code
- Single, self contained binary. Runs almost on any platform and OS.
- Vivid community. Good documentation. Good Tooling.
- Go is the major language behind the Cloud Native Stack, many important components are written in Go

Workshop Agenda

09:00 - 09:15 | Begrüßung und Einleitung

09:15 - 10:15 | Getting to Know Go: Basics and Tooling

10:30 - 12:00 | Building CLI Applications with Go and Cobra

13:00 - 14:30 | Kubernetes Sidecars in Go

14:45 - 16:00 | Building a Kubernetes operator in Go

Hello world!

https://learnxinyminutes.com/docs/go/

https://gobyexample.com

Workshop Setup

https://github.com/qaware/go-for-operations

https://golang.org/doc/install

https://code.visualstudio.com/docs/setup/setup-overview

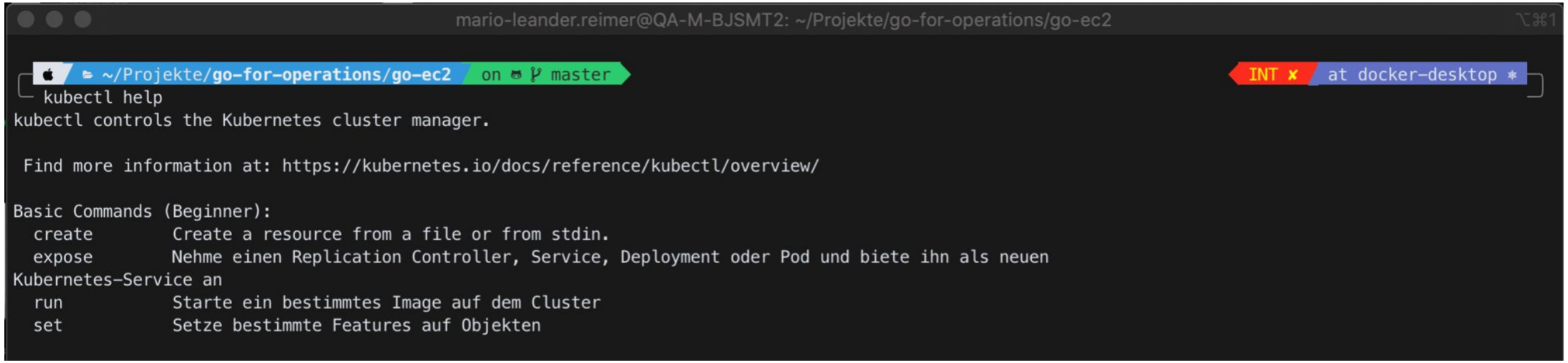
The renaissance of the plain old Makefile


```
VERSION = v1.0.0

.PHONY: build
build:
    # omit the symbol table, debug information and the DWARF table
    @go build -o go-example -ldflags="-s -w -X main.version=$(VERSION)"

clean:
    @go clean

test: build
    @go test -v

all: clean build test

release: all
    @goreleaser --snapshot --skip-publish --rm-dist
```

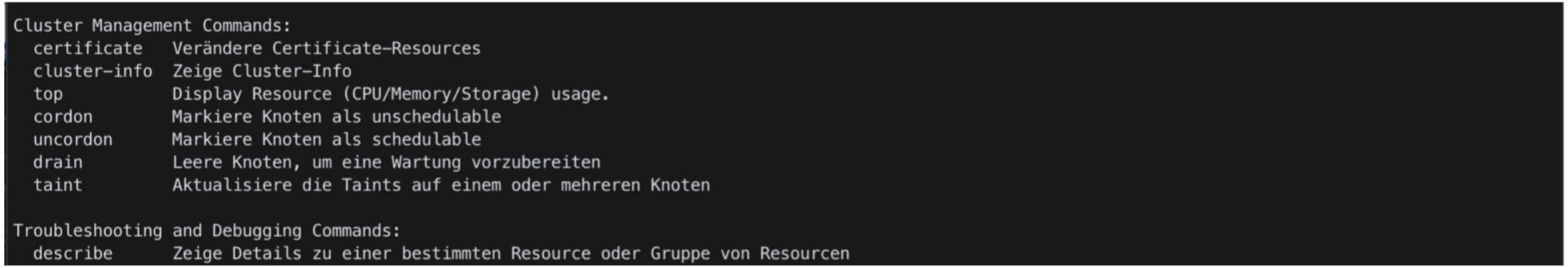
Use GoReleaser to publish multi OS binaries

- https://goreleaser.com
- Cross-compile your Go project quick and easily
- Release to GitHub, GitLab et.al
- Create Docker images and manifests
- Create Linux packages and Homebrew Taps
- Upload to Bintray, Artifactory and other Public Blob Stores
- … and much more!

```yaml
project_name: go-example
before:
  hooks:
    - go mod download
builds:
  - env:
      - CGO_ENABLED=0
    goos:
      - linux
      - windows
      - darwin
    goarch:
      - 386
      - amd64
    ldflags: -s -w -X main.version={{.Version}}
archives:
  - name_template: '{{ .ProjectName }}-{{ .Version }}-{{ .Os }}-{{ .Arch }}'
    format_overrides:
     - goos: windows
       format: zip
    replacements:
      darwin: Darwin
      linux: Linux
      windows: Windows
      386: i386
      amd64: x86_64
```



CLIs - The Swiss Army Knife of Dev and Ops



The Basics of 12-factor CLI applications

- Great help is essential. What version am I on?
- Prefer flags to positional arguments.
- Mind the streams. stdout is for output, stderr is for messaging.
- Handle things going wrong: error code, title, how to fix, URL, …
- Be fancy: use colours, have shell completion.
- Prompt if you can.
- Be speedy. CLIs need to start fast.
- Be clear about subcommands.

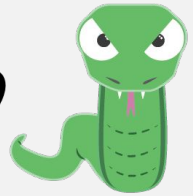
For complete list and info, read https://medium.com/@jdxcode/12-factor-cli-apps-dd3c227a0e46

cobra

QA|WARE

- https://github.com/spf13/cobra
- Cobra is a library providing a simple interface to create powerful modern CLI interfaces similar to git & go tools.
- Cobra is also an application that will generate your application scaffolding to rapidly develop a Cobra-based application.
- Cobra is used in many Go projects such as Kubernetes, Docker, Skaffold, Helm and Istio just to name a few.

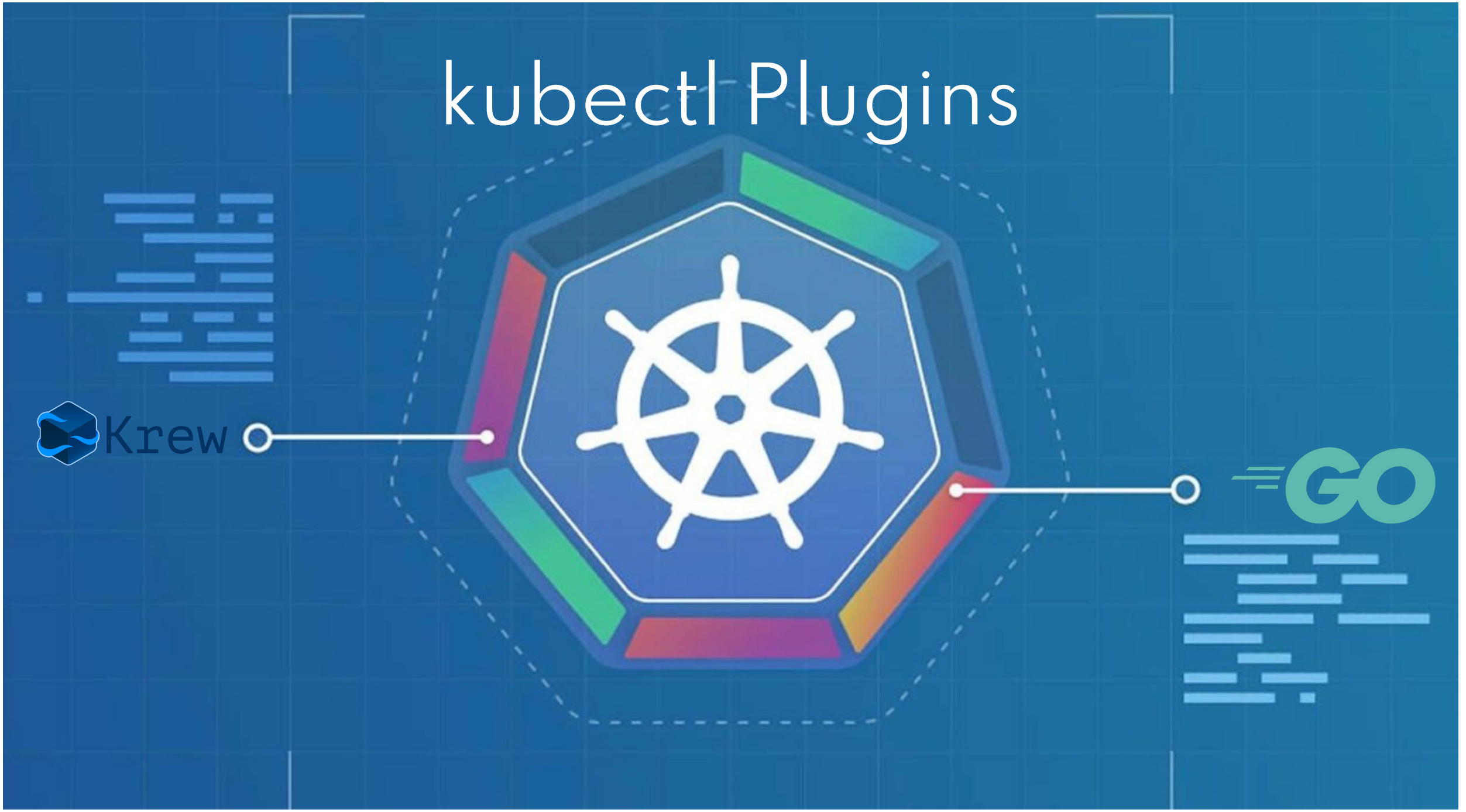
viper



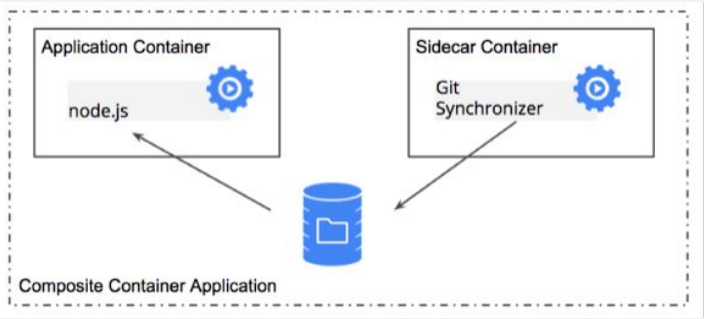
QA|WARE

- https://github.com/spf13/viper
- Viper is a library for configuration handling of Go applications.
- It supports all types of configuration whether its YAML, environment variables, flags or remote config systems
- Live watching and re-reading of config files included
- Overwriting of configuration is built-in

kubectl Plugins



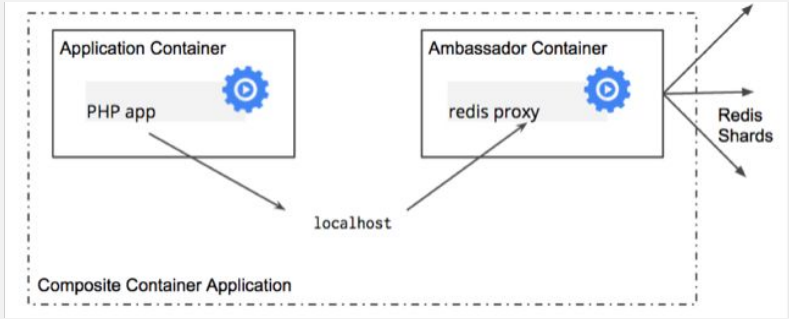
Container Orchestration Patterns



Sidecar Container

Extended Container Behaviour

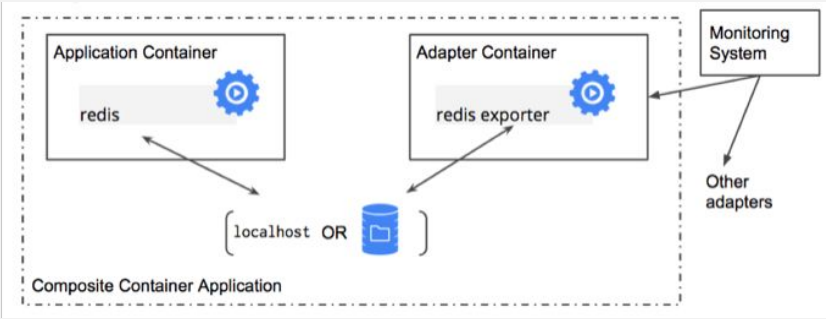
- Log Extraction / Reformatting (fluentd, file beat)
- Scheduling (cron, quartz)



Ambassador Container

Proxy Communication

- TLS Tunnel (ghostunnel, Istio)
- Circuit Breaking (linked, Istio)
- Request Monitoring (linked, Istio)



Adapter Container

Standardized Ops Interfaces

- Monitoring (Prometheus)
- Configuration (ConfigMaps, Secrets, …)

Use a multi-stage Dockerfile to build Linux binaries


```
FROM golang:1.15.2 as builder

WORKDIR /build

COPY . /build
RUN go build -o logship-sidecar -ldflags="-s -w"

FROM gcr.io/distroless/static-debian10
COPY --from=builder /build/logship-sidecar /

ENV LOG_DIRECTORY=/logs
ENV LOG_FILE_PATTERN=.+.gz
ENV LOG_SCAN_INTERVAL=10

ENTRYPOINT ["/logship-sidecar"]
CMD [""]
```

Stage 1: Building

Stage 2: Running


Operator.
- Do stuff to my Kubernetes.

What are operators?

- **Operators are codified Ops procedures!**
- Operators are the path towards Zero-Ops. They enable auto-updating, self-monitoring and self-healing infrastructure and applications.
- The concept was coined in the Kubernetes world. It's now been adopted and used widespread in the cloud native world.
- Examples: OKD, Sealed Secrets, Kube Monkey, Weave Flux

Kubernetes API Extensions with Custom Resources

- User defined extensions of the Kubernetes APIs
- Allow the abstraction of complex application constructs and concepts
- Definition solely via `CustomResourceDefinitions`
- Structure definition via OpenAPI v3.0 Validation Schema
- Default Support for several API Features: CRUD, Watch, Discovery, json-patch, merge-patch, Admission Webhooks, Metadata, RBAC, …
- Versioning und Conversion supported via Webhooks

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
        environment: integration
   spec:
      containers:
      - name: nginx
        image: nginx:1.19.4-alpine
        ports:
        - containerPort: 80

        # probe definitions
        # resource constraints
        # volumes and mounts
```

+

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: LoadBalancer
  ports:
    - port: 80
      protocol: TCP
  selector:
    app: nginx
```

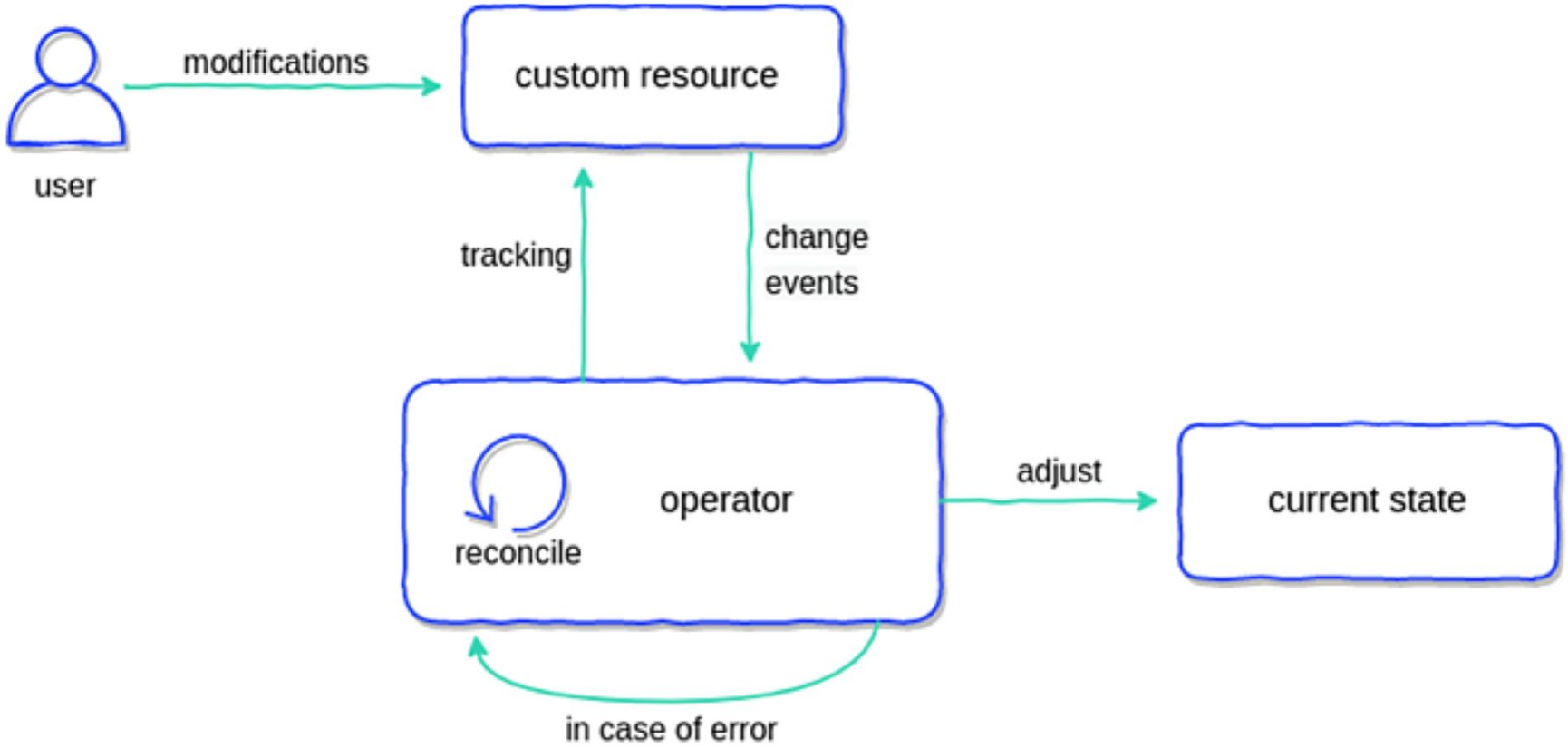
=

```yaml
apiVersion:
k8s.qaware.de/v1alpha1
kind: Microservice
metadata:
  name: microservice-example
  labels:
    app: nginx
spec:
  image: nginx:1.19.4-alpine
  replicas: 2
  serviceType: LoadBalancer
  ports:
    - 80
```

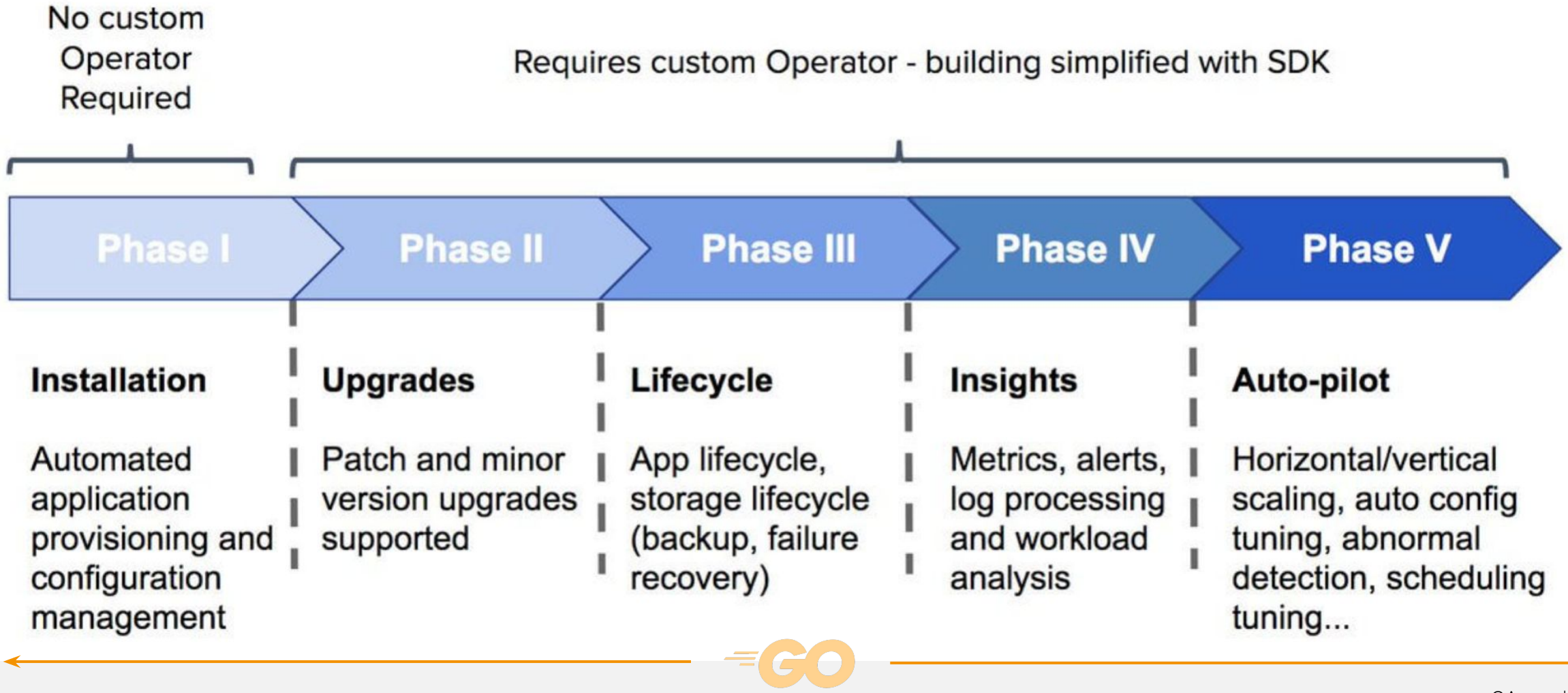
QA|WARE

Kubernetes Operators Explained

QA|WARE



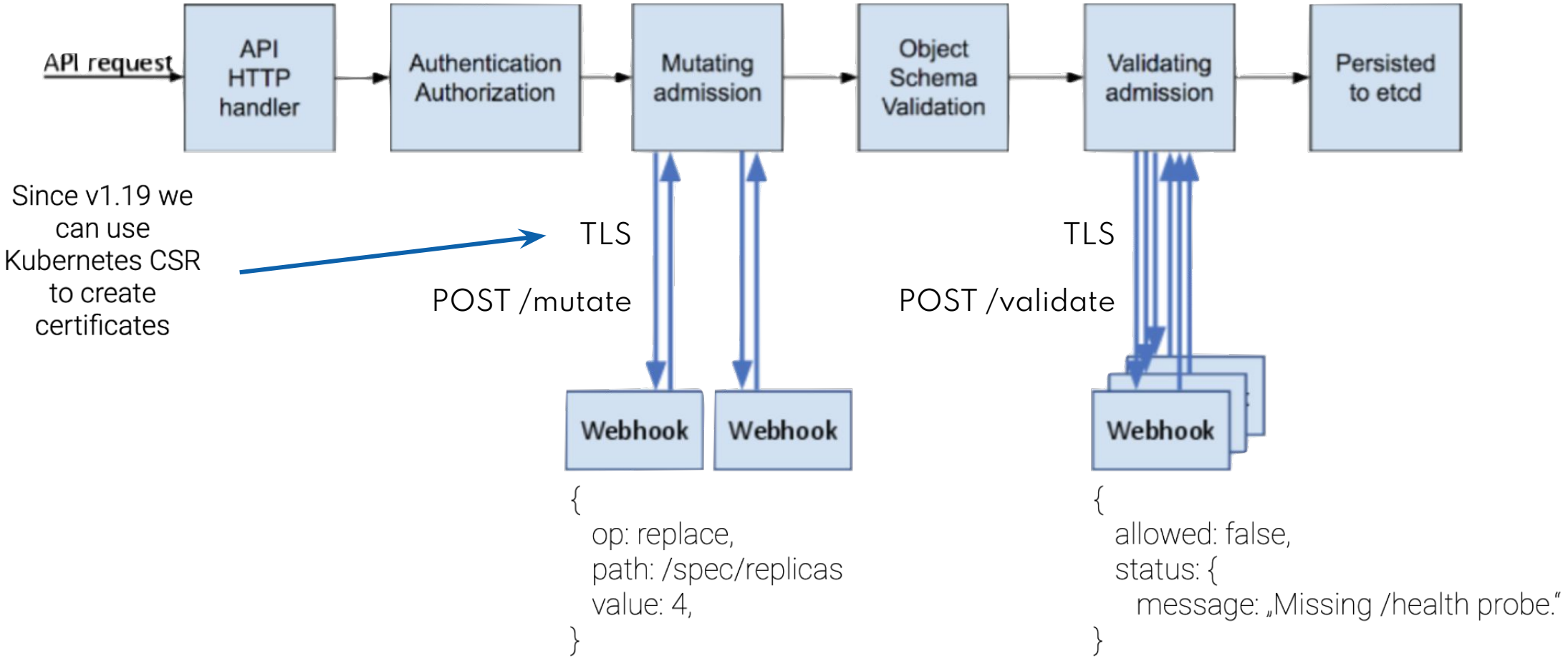
Introducing the Operator SDK



Cross-Cutting Concerns via Admission Controllers

- Admission Controllers are like (dynamic) plugins for Kubernetes
- Security: increase security by mandating a reasonable security baseline across an entire namespace or cluster, e.g. PodSecurityPolicy, OPA.
- Governance: apply and enforce the adherence to best practices, e.g. good labels, annotations, resource limits, probes.
- Configuration Management: validate configuration of objects and prevent obvious misconfigurations from hitting the cluster.
- More than 30 admission controllers shipped with Kubernetes, incl. the *MutatingAdmissionWebhook* and *ValidatingAdmissionWebhook*

Mutating and Validating Admission Explained



qaware.de

QA|WARE
SOFTWARE ENGINEERING

QAware GmbH
Aschauer Straße 32
81549 München
Tel. +49 89 232315-0
info@qaware.de


twitter.com/qaware

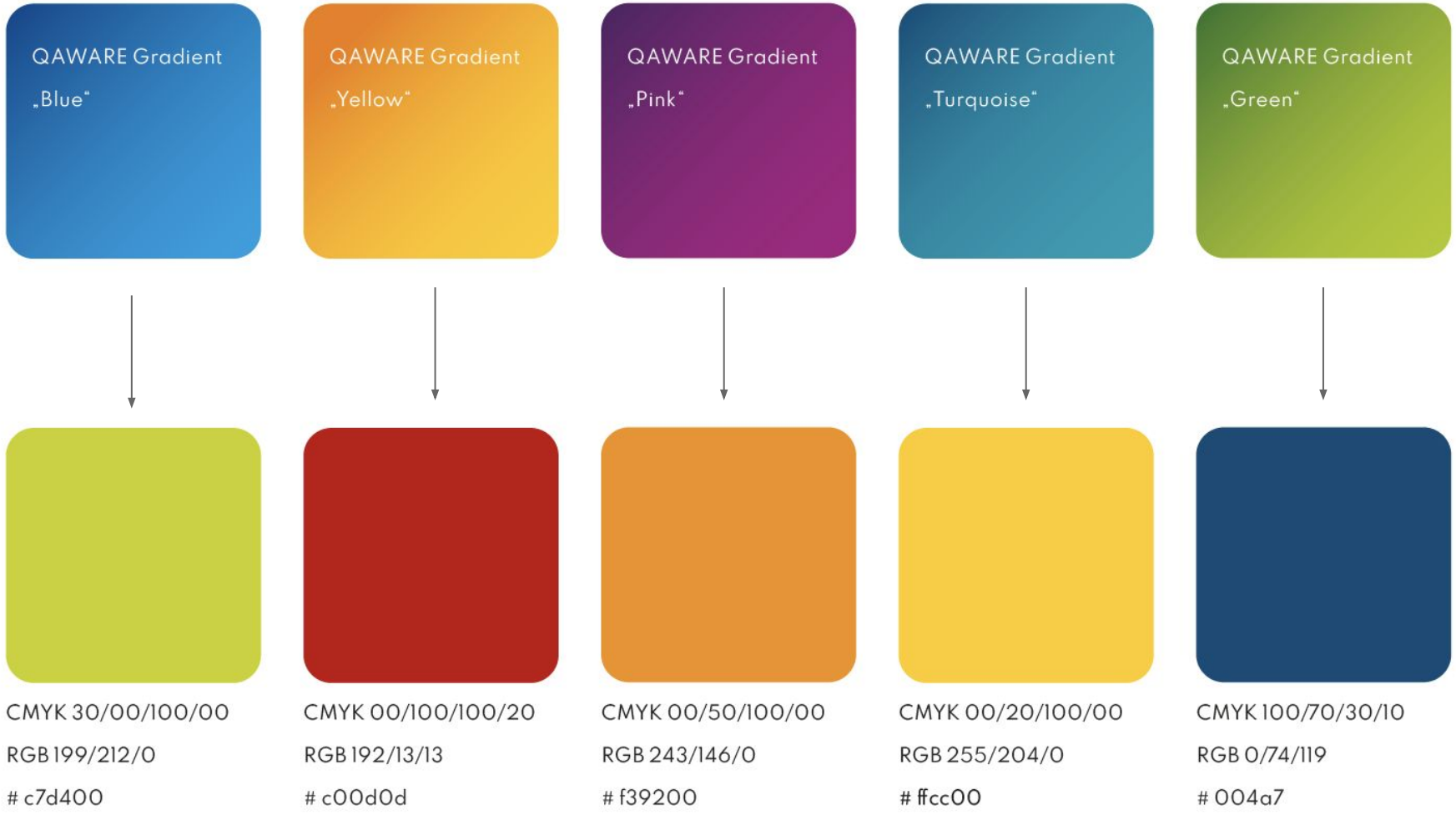

linkedin.com/company/qaware-gmbh


xing.com/companies/qawaregmbh


slideshare.net/qaware


github.com/qaware

Spot-Colors für Farbverläufe



Wer Akzente setzen und Texte hervorheben möchte, kann sich hier die Spot-Color für den verwendeten Farbverlauf raussuchen.