Scaling LiveOps with Go: Building Cloud Native Game Servers

Berkay Uckac

February 26, 2025 @ Helsinki Gophers

Today's talk:

Includes:

- Learnings, decisions, the story
- Experiences
- Problems and challenges
- Hot(?) Go topics, ...like folder structure
- Why Go?
- ... and most importantly: is Go still a good fit for our use case after all this time?

Does **not** include:

- Deep dives
- "*The way*" of accomplishing things

How do I spell your name again?

Futureplay

MERGE GARDENS 2020-2023

- Launched in 2020
- Successful launch, players are loving it, metrics look great
- A lot of competition in the market

MERGE GARDENS 2023-

- 2023, enter the revamped Merge Gardens
- New story, new marketing, new graphics, new ads, new.. everything?
- +1000% more downloads, mega influx of new or returning players

LIVE OPERATIONS(Liveops): ... refers to ongoing management, updates, and community-driven events implemented post-launch to keep players engaged, maintain game health, and drive consistent growth and revenue

FU TU
RE
PLA Y

CHALLENGE: **... how about the tech?**

FU TU
RE
PLA Y

Can we achieve it with a small team?

FU TU
RE
PLA Y

Well... can’t be a challenge if it doesn’t exist, right?

FU TU
RE
PLA Y

Not Quite... Greenfield projects come with their downsides:

- Cost and Timeline Overruns
- Uncertain ROI
- Over-Engineering
- Scope Creep
- Interfacing with Legacy Systems
- Team Experience
- Technology Obsolescence

  ...

FU TU
RE
PLA Y

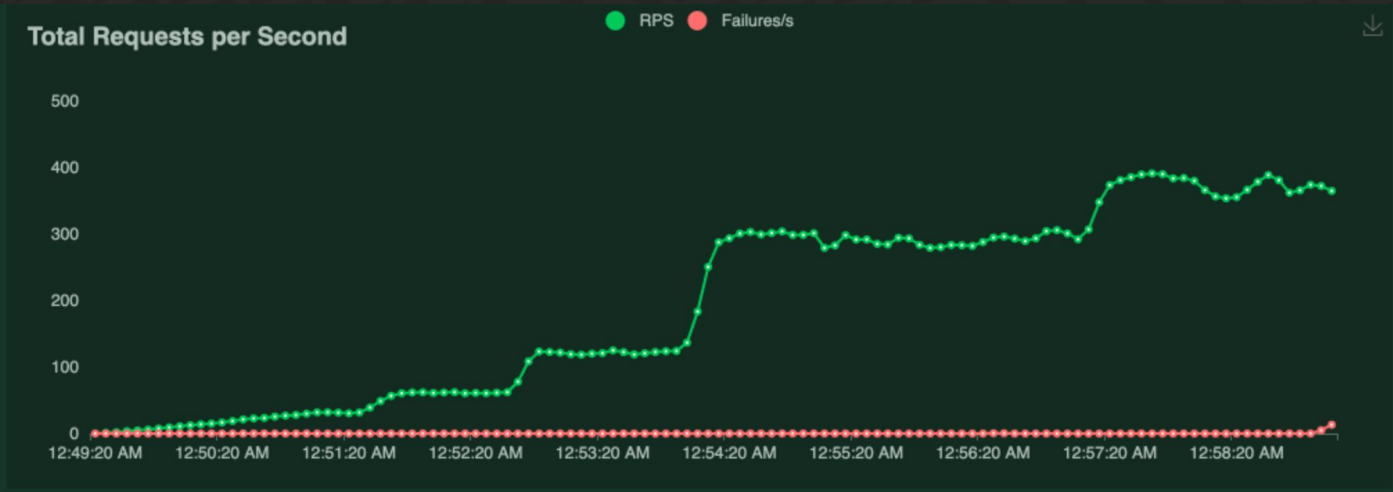
Shopping List:

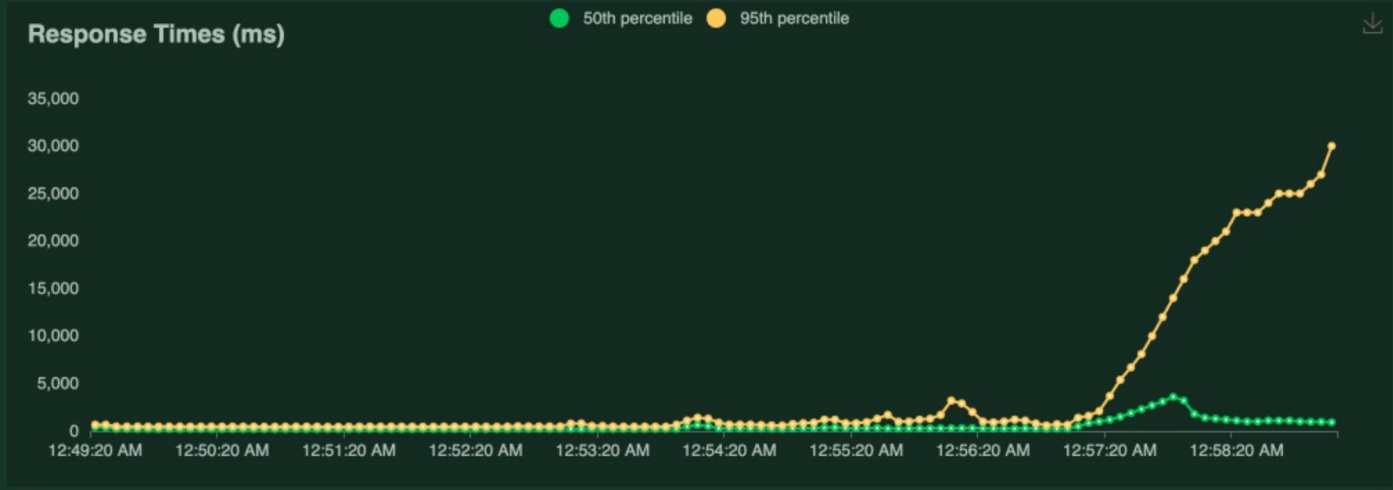
- Scalable to react to changes in traffic
- Maintainable with a small server team
- Highly available
- Using tools that do the job in ways that are easy to understand and maintain
- Mostly stateless server architecture
- Programming language
  - Performance
  - Developer happiness

Programming Language

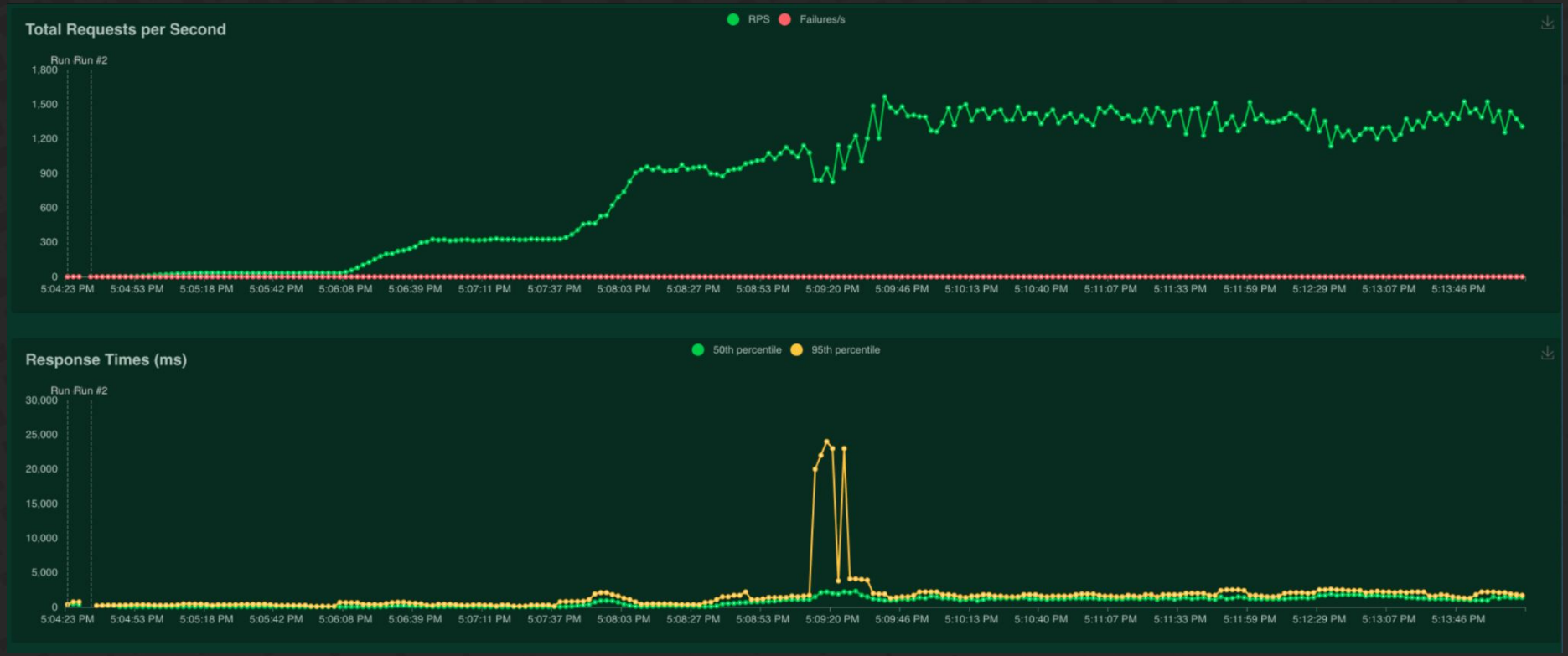
- Java (Spring Boot)
  - + Well-known, widely used in the industry, in-house experience
  - - Slow startup times and high memory consumption
- JavaScript (Node/Express)
  - + Existing in-house tools and experience
  - - Speed
  - - DevEx
- Go
  - + SPEED, modern, cloud native, 1.0 promise
  - + Isn’t magical
  - - Not a lot knowledge in-house except small prototypes

FU TU
RE
PLA Y


Total Requests per Second
RPS
Failures/s
500
400
300
200
100
0
12:49:20 AM
12:50:20 AM
12:51:20 AM
12:52:20 AM
12:53:20 AM
12:54:20 AM
12:55:20 AM
12:56:20 AM
12:57:20 AM
12:58:20 AM


Response Times (ms)
50th percentile
95th percentile
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
12:49:20 AM
12:50:20 AM
12:51:20 AM
12:52:20 AM
12:53:20 AM
12:54:20 AM
12:55:20 AM
12:56:20 AM
12:57:20 AM
12:58:20 AM

FU TU
RE
PLA Y


Total Requests per Second
RPS
Failures/s
Run Run #2
1,800
1,500
1,200
900
600
300
0
5:04:23 PM
5:04:53 PM
5:05:18 PM
5:05:42 PM
5:06:08 PM
5:06:39 PM
5:07:11 PM
5:07:37 PM
5:08:03 PM
5:08:27 PM
5:08:53 PM
5:09:20 PM
5:09:46 PM
5:10:13 PM
5:10:40 PM
5:11:07 PM
5:11:33 PM
5:11:59 PM
5:12:29 PM
5:13:07 PM
5:13:46 PM
Response Times (ms)
50th percentile
95th percentile
Run Run #2
30,000
25,000
20,000
15,000
10,000
5,000
0
5:04:23 PM
5:04:53 PM
5:05:18 PM
5:05:42 PM
5:06:08 PM
5:06:39 PM
5:07:11 PM
5:07:37 PM
5:08:03 PM
5:08:27 PM
5:08:53 PM
5:09:20 PM
5:09:46 PM
5:10:13 PM
5:10:40 PM
5:11:07 PM
5:11:33 PM
5:11:59 PM
5:12:29 PM
5:13:07 PM
5:13:46 PM

FU TU
RE
PLA Y

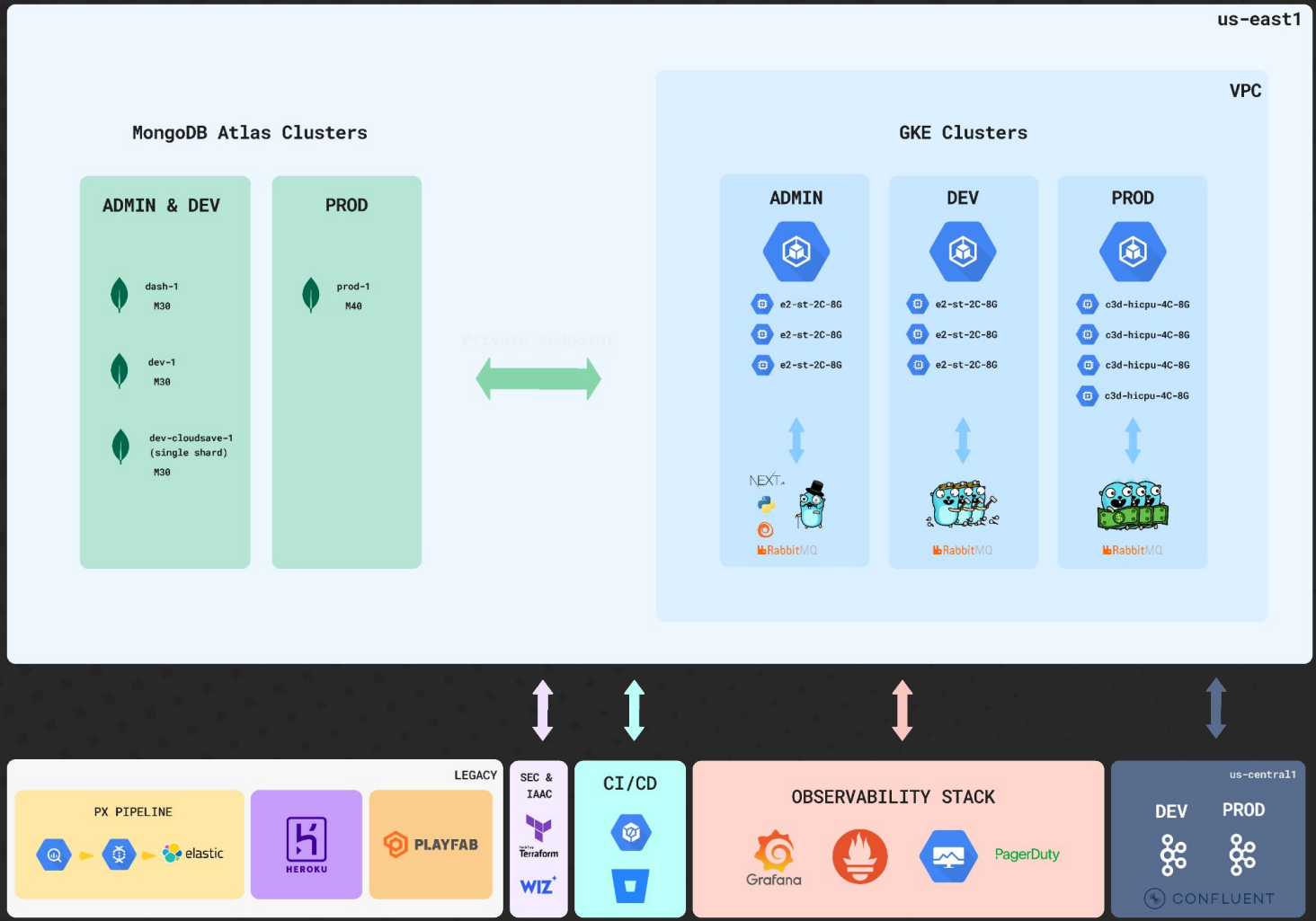
Go, Kubernetes, Terraform, MongoDB and GCP

10,000-foot view, first days

```
├── README.md
├── charts
├── k8s/
│   ├── deployment/
│   ├── helm/
│   ├── prometheus/
│   └── terraform/
├── lib/
├── services/
    ├── auth-service/
    ├── config-campaign-service/
    ├── config-session-service/
    └── player-service/
```

```
service-name/
├── Dockerfile
├── README.md
├── go.mod
├── go.sum
├── helm/
│   └── service-name/
│       ├── Chart.yaml
│       ├── managed-cert.yaml
│       ├── service-monitor.yaml
│       ├── templates/
│       │   ├── NOTES.txt
│       │   ├── _helpers.tpl
│       │   ├── deployment.yaml
│       │   ├── hpa.yaml
│       │   ├── ingress.yaml
│       │   ├── service-monitor.yaml
│       │   ├── service.yaml
│       │   ├── serviceaccount.yaml
│       │   └── tests/
│       │       └── test-connection.yaml
│       └── values.yaml
├── main.go
└── main_test.go
```

FU TU
RE
PLA Y


us-east1
VPC
MongoDB Atlas Clusters
ADMIN & DEV
dash-1
M30
dev-1
M30
dev-cloudsave-1
(single shard)
M30
PROD
prod-1
M40
Private Endpoint
GKE Clusters
ADMIN
e2-st-2C-8G
e2-st-2C-8G
e2-st-2C-8G
DEV
e2-st-2C-8G
e2-st-2C-8G
e2-st-2C-8G
PROD
c3d-hicpu-4C-8G
c3d-hicpu-4C-8G
c3d-hicpu-4C-8G
c3d-hicpu-4C-8G
RabbitMQ
RabbitMQ
RabbitMQ
LEGACY
PX PIPELINE
elastic
HEROKU
PLAYFAB
SEC & IAAC
Terraform
WIZ
CI/CD
OBSERVABILITY STACK
Grafana
PagerDuty
us-central1
DEV
PROD
CONFLUENT
FU TU RE PLA Y

10,000-foot view, present day

```
.
├── ci/                          # CI/CD scripts, Dockerfiles, ko
├── k8s/                         # K8 resources
│   ├── helm/                    # Helm charts
│   ├── helmfile/                # Helmfile configs for multiple environments
│   └── terraform/               # Terraform modules and projects
├── lib/                         # Shared libraries (auth, cache, helpers...)
├── services/                    # Microservices
│   ├── admin/
│   ├── analytics/
│   ├── auth/
│   ├── cache/
│   ├── cloudsave/
│   ├── config/
│   ├── iap/
│   ├── tournament/
│   └── ...                      # Other services (gift, websocket, time...)
├── build-to-tenant.sh           # Tenant build scripts
├── cloudbuild.yaml              # GCP Cloud Build config
├── go.mod / go.sum              # Go dependencies
└── README.md
```

FU TU
RE
PLA Y

10,000-foot view, present day

```
service-name/
├── internal/
│   ├── adapters/        # Integrations with external systems (e.g. DB, APIs)
│   ├── domain/          # Core domain/business logic
│   ├── model/           # Data models & types
│   ├── ports/           # Interfaces, adapters
│   ├── api.go           # Endpoints
│   └── service.go       # Main service orchestration (business logic)
├── inject.go            # Wires up dependencies (Google Wire)
├── main.go              # Service entry point (starts HTTP server, etc.)
└── wire_gen.go          # Generated DI code
```

Inside (service-name) Service

```go
package main

import (
    "fmt"
    "futureplay/backend/lib/setup"
    "github.com/gofiber/fiber/v2"

    "github.com/rs/zerolog/log"
)

func main() {
    err := setup.StartService("fp-backend-auth", 8000, createRoutes)
    if err != nil {
        log.Fatal().Msg(fmt.Sprintf("Error starting Fiber: %s", err.Error()))
    }
}

func createRoutes(app setup.FiberAppAdapter) {
    api := GetDefaultHandler()

    app.PublicGet("/auth/health/", func(c *fiber.Ctx) error { return c.JSON("OK") })

    app.PublicPost("/auth/register/", api.Register)
    app.PublicPost("/auth/register/:pid/", api.Register)
    app.PublicPost("/auth/login/:pid", api.Login)
    ...
    app.PrivatePostWithPlayerId("/auth/social/login/apple/", api.LoginApple)
    app.PrivatePostWithPlayerId("/auth/social/logout/facebook/", api.LogoutFacebook)
    ...
    ...
    app.InternalGet("/auth/players/apple/:appleID", api.GetPlayersWithAppleAccount)
    app.InternalPost("/auth/player/:pid/unlink/:platform", api.UnlinkSocialLogin)
    app.InternalPost("/auth/player/:pid/link/plarium", api.LinkPlariumId)
    ...
}
```

FU TU
RE
PLA Y

Inside (service-name) Service

```go
// InitializationFunction is a function signature used for callback functions that
construct
// app specific routes in StartService.
type InitializationFunction func(app FiberAppAdapter)

// PrivateRequestHandler is a request handler type for PrivateGet and PrivatePost type
// API endpoints. It receives the playerId read from JWT, so you don't need to pass the
// player ID in the request.
type PrivateRequestHandler = func(playerId string, c *fiber.Ctx) error

type FiberAppAdapter interface {
    PublicGet(path string, handlers ...func(c *fiber.Ctx) error)
    PublicPost(path string, handlers ...fiber.Handler)

    PrivateGet(path string, handlers ...fiber.Handler)
    PrivatePost(path string, handlers ...fiber.Handler)
    PrivateDelete(path string, handlers ...fiber.Handler)

    // PrivateGetWithPlayerId and PrivatePostWithPlayerId read the player ID from JWT
token,
    // that's why they call a PrivateRequestHandler, which takes the
    // player ID in addition to the Fiber Context.
    PrivateGetWithPlayerId(path string, handlers ...PrivateRequestHandler)
    PrivatePostWithPlayerId(path string, handlers ...PrivateRequestHandler)

    InternalGet(path string, handlers ...fiber.Handler)
    InternalPost(path string, handlers ...fiber.Handler)
    InternalDelete(path string, handlers ...fiber.Handler)

    // CustomGet and CustomPost and CustomDelete are for cases where you need to specify
authentication
    // handlers not covered by the above methods.
    CustomGet(path string, authHandler, handler fiber.Handler)
    CustomPost(path string, authHandler, handler fiber.Handler)
    CustomDelete(path string, authHandler, handler fiber.Handler)
}
```

FU TU
RE
PLA Y

Inside (service-name) Service

```
// StartService sets up the web service with common functionality such as telemetry and
// logging and starts listening to requests on port 8000. App specific routes should be
// configured by passing in a function that implements the InitializationFunction
signature.
func StartService(serviceName string, port int, initFunctions ...InitializationFunction)
error {
    log.Info().Msg(fmt.Sprintf("Initializing Fiber. Service: %s", serviceName))

    app := fiber.New(fiber.Config{
        AppName:                 serviceName,
    })
    ...
    app.Use("/websocket/ws", func(c *fiber.Ctx) error {
      // IsWebSocketUpgrade returns true if the client
      // requested upgrade to the WebSocket protocol.
      if websocket.IsWebSocketUpgrade(c) {
          c.Locals("allowed", true)
          return c.Next()
      }
      return fiber.ErrUpgradeRequired
    })
    ...
    app.Use(compress.New(
        compress.Config{
            Level: compress.LevelDefault,
        }))
    ...
    if rateLimiterConfig := getRateLimiterConfig(); rateLimiterConfig != nil {
        app.Use(limiter.New(limiter.Config{
            Max:        rateLimiterConfig.Max,
            Expiration: rateLimiterConfig.Expiration * time.Second,
        }))
    }
    ...
    // If one or more player facing routes were configured, we need to have the JWT keys
set up
    if appAdapter.HasRoutesWithPlayerAuth() {
        err = auth.InitJwtKeys()
        if err != nil {
            panic(fmt.Sprintf("Error reading JWT keys: %v", err))
        }
    }
    ...
}
```

FU TU
RE
PLA Y

Inside (service-name) Service

```
func GetDefaultHandler() *internal.ApiHandler {
    panic(
        wire.Build(
            wire.Bind(new(ports.DataRepository), new(adapters.MongoDataRepository)),
            wire.Bind(new(ports.GeoIpRepository), new(adapters.MaxMindGeoIpRepository)),
            wire.Bind(new(ports.FacebookRepository), new(adapters.FacebookRepository)),
            wire.Bind(new(ports.AppleRepository), new(adapters.AppleRepository)),
            wire.Bind(new(ports.PlariumPlayRepository),
new(adapters.PlariumPlayRepository)),
            wire.Bind(new(ports.MetricsRepository), new(adapters.MetricsRepository)),
            wire.Bind(new(ports.PlayerDataProvider),
new(adapters.PlayFabPlayerDataProvider)),
            wire.Bind(new(domaininterfaces.GeoLocator), new(domain.GeoLocatorImpl)),
            wire.Bind(new(domaininterfaces.SecHandler), new(domain.SecHandlerImpl)),
            wire.Bind(new(setup.TimeProvider), new(setup.RealTimeProvider)),
            wire.Bind(new(ports.GiftRepository), new(adapters.GiftRepositoryAdapter)),
            adapters.ProvideFacebookRepository,
            adapters.ProvideAppleRepository,
            adapters.ProvideGeoIpRepository,
            adapters.ProvideMongoDataRepository,
            adapters.ProvideMetricsRepository,
            adapters.ProvidePlariumPlayRepository,
            adapters.ProvidePlayFabPlayerDataProvider,
            adapters.ProvideGiftRepositoryAdapter,
            internal.ProvideService,
            domain.ProvideGeoLocator,
            domain.ProvideSecHandler,
            ProvideMongoConnectionString,
            internal.ProvideApiHandler,
            setup.ProvideRealTimeProvider,
        ))
}
```

FU TU
RE
PLA Y

Scaling Go on Kubernetes, K8 HPA

```
{{- range $serviceName, $props := .Values.services }}
{{- with $props }}
{{- if and .enabled .deployment.autoscaling .deployment.autoscaling.enabled }}
---
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: {{ include "fp-backend.servicename" . }}
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: {{ include "fp-backend.servicename" . }}
  minReplicas: {{ .deployment.autoscaling.minReplicas | default
$.Values.autoscaling.minReplicas }}
  maxReplicas: {{ .deployment.autoscaling.maxReplicas | default
$.Values.autoscaling.maxReplicas }}
  metrics:
    {{- if $.Values.autoscaling.targetCPUUtilizationPercentage }}
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: {{
$.Values.autoscaling.targetCPUUtilizationPercentage }}
    {{- end }}
    {{- if $.Values.autoscaling.targetMemoryUtilizationPercentage }}
    - type: Resource
      resource:
        name: memory
        target:
          type: Utilization
          averageUtilization:   {{
$.Values.autoscaling.targetMemoryUtilizationPercentage }}
    {{- end }}
  behavior:
    {{- if $.Values.autoscaling.scalingDownStabilizationWindowSeconds }}
    scaleDown:
      stabilizationWindowSeconds: {{
$.Values.autoscaling.scalingDownStabilizationWindowSeconds }}
    {{- end }}
    {{- if $.Values.autoscaling.scalingUpStabilizationWindowSeconds }}
    scaleUp:
      policies:
        - periodSeconds: {{
$.Values.autoscaling.scalingUpStabilizationWindowSeconds }}
      stabilizationWindowSeconds: {{
$.Values.autoscaling.scalingUpStabilizationWindowSeconds }}
    {{- end }}
{{- end }}
{{- end }}
{{- end }}
```

FU TU
RE
PLA Y

Scaling Go on Kubernetes, K8 HPA

```go
// scaleDeployment scales a deployment to the desired number of replicas
func (s *Scheduler) scaleDeployment(ctx context.Context, name, namespace string,
replicas int32) error {
   deployment, err := s.client.AppsV1().Deployments(namespace).Get(ctx, name,
metav1.GetOptions{})
   if err != nil {
       return fmt.Errorf("failed to get deployment: %w", err)
   }

   if deployment.Spec.Replicas != nil && *deployment.Spec.Replicas == replicas {
       s.logger.Printf("Deployment %s/%s already at %d replicas", namespace,
name, replicas)
       return nil
   }
...
   _, err = s.client.AppsV1().Deployments(namespace).Update(ctx, deploymentCopy,
metav1.UpdateOptions{})
   if err != nil {
       return fmt.Errorf("failed to update deployment: %w", err)
   }

   s.logger.Printf("Successfully scaled deployment %s/%s", namespace, name)
   return nil
}

// scaleStatefulSet scales a statefulset to the desired number of replicas
func (s *Scheduler) scaleStatefulSet(ctx context.Context, name, namespace
string, replicas int32) error {
...
   _, err = s.client.AppsV1().StatefulSets(namespace).Update(ctx,
statefulsetCopy, metav1.UpdateOptions{})
   if err != nil {
       return fmt.Errorf("failed to update statefulset: %w", err)
   }
...
}
```

https://github.com/berkayuckac/k8schedul8r

FU TU
RE
PLA Y

Deployments

```
  - id: 'test'
    name: 'us-east1-docker.pkg.dev/$PROJECT_ID/build/ko:v0.15.2b'
    waitFor: ['wire']
    script: |
      echo "Running tests..."
      KO_DATA_PATH=$(pwd)/services/auth/kodata go test -p 32 $(go list ./... | grep -v
/admin)
    env:
      - 'GOMAXPROCS=32'

  - id: 'build-auth-service'
    name: 'us-east1-docker.pkg.dev/$PROJECT_ID/build/ko:v0.15.2b'
    waitFor: ['test']
    script: ko build -t $SHORT_SHA -B ./services/auth
    automapSubstitutions: true
```

FU TU RE PLA Y

Deployments

```
################################################
# Run wire for the relevant services
################################################
for svc in "${BUILD_SERVICES[@]}"; do
    echo "Running wire in services/$svc..."
    (cd "services/$svc" && wire)
done

################################################
# Build the services
################################################
# We'll build each service in BUILD_SERVICES. We'll check if it's in CGO_SERVICES
or HOST_SERVICES.
for svc in "${BUILD_SERVICES[@]}"; do
    if [[ " ${HOST_SERVICES[*]} " =~ ${svc} ]]; then
        # Build host service with ko
        echo "Building host service: $svc..."
        ko build -t "$TENANT" -B "./services/$svc"

    elif [[ " ${CGO_SERVICES[*]} " =~ ${svc} ]]; then
        # Build CGO service in cgo-builder container with ko
        REMOTE_TAG="${KO_DOCKER_REPO}/${svc}:${TENANT}"
        echo "Building CGO-dependent service: $svc in cgo-builder container..."
        docker exec cgo-builder bash -c "CGO_ENABLED=1 ko publish --local -t
\"$TENANT\" -B /workspace/services/$svc"

        echo "Tagging $svc image..."
        docker tag "ko.local/${svc}:${TENANT}" "${REMOTE_TAG}"

        echo "Pushing image: ${REMOTE_TAG}..."
        docker push "${REMOTE_TAG}"
    else
        echo "Error: $svc not found in HOST_SERVICES or CGO_SERVICES? Skipping..."
    fi
done
```

FU TU
RE
PLA Y

+ **295ms** total P99, **104ms** US P99, **342ms** Germany P99, **655ms** Japan, **229ms** UK
+ **99.99896%** 2024 uptime (yearly 5m, daily 0.9s), **99.99% 1Y SLO** (yearly 52m, daily 8.6s)

FU TU
RE
PLA Y

Resources & Further Exploration

- Distributed server for social and realtime games and apps.
  https://github.com/heroiclabs/nakama

- Going Infinite, handling 1M websockets connections in Go
  https://github.com/eranyanay/1m-go-websockets

- Go Docs
  https://go.dev/doc/

FU TU
RE
PLA Y

Do we think the time we spent getting up-to-speed with Go was nice? Not even sunk-cost-fallacy?

Yes, and nope!

Is Go still a good fit for our use case after all this?

It sure is!

Do we recommend Go usage in game servers?

Yep!

Things I’d do differently at the start:

- CGO usage
- More, like a lot more interfaces
- Read more bigger repositories
- Tenant system and fast iteration from the the get-go
- Looser service coupling

WANNA MAKE GAMES WITH US?

https://www.futureplaygames.com/careers

Q&A, feedback, comments?

Thanks!

berkay.fi
hey@berkay.fi