GO IN PRACTICE

TEXT TO SPEECH MICROSERVICE

Tomasz Smelcerz
Paweł Sołtysek

MICROSERVICES ARCHITECTURE

OUR USE CASE

Kasa samoobsługowa
Jogurty
1 Zeskanuj produkt
2 Spakuj zakupy
3 Zapłać kartą lub gotówką
UWAGA: przy kasowaniu produktów z zabezpieczeniem poproś o pomoc asystenta kas.
Orbit

OUR USE CASE

- Let's build:

  Talking
  self-checkout machine!

OUR USE CASE

- Possible solution…

?

OUR USE CASE

Why not to use existing provider directly?

- Low latency
- Voice Message re-use
- Low cost (each message costs money)

REST API

/voiceMessages
- **POST**: Create new TTS requests (json data)
- responds with: json data

/voiceMessages/ID
- **GET**: Get TTS request data
  Response: As in POST (json data)

/media/ID
- **GET**: Get media file (audio data)

REST API, CD

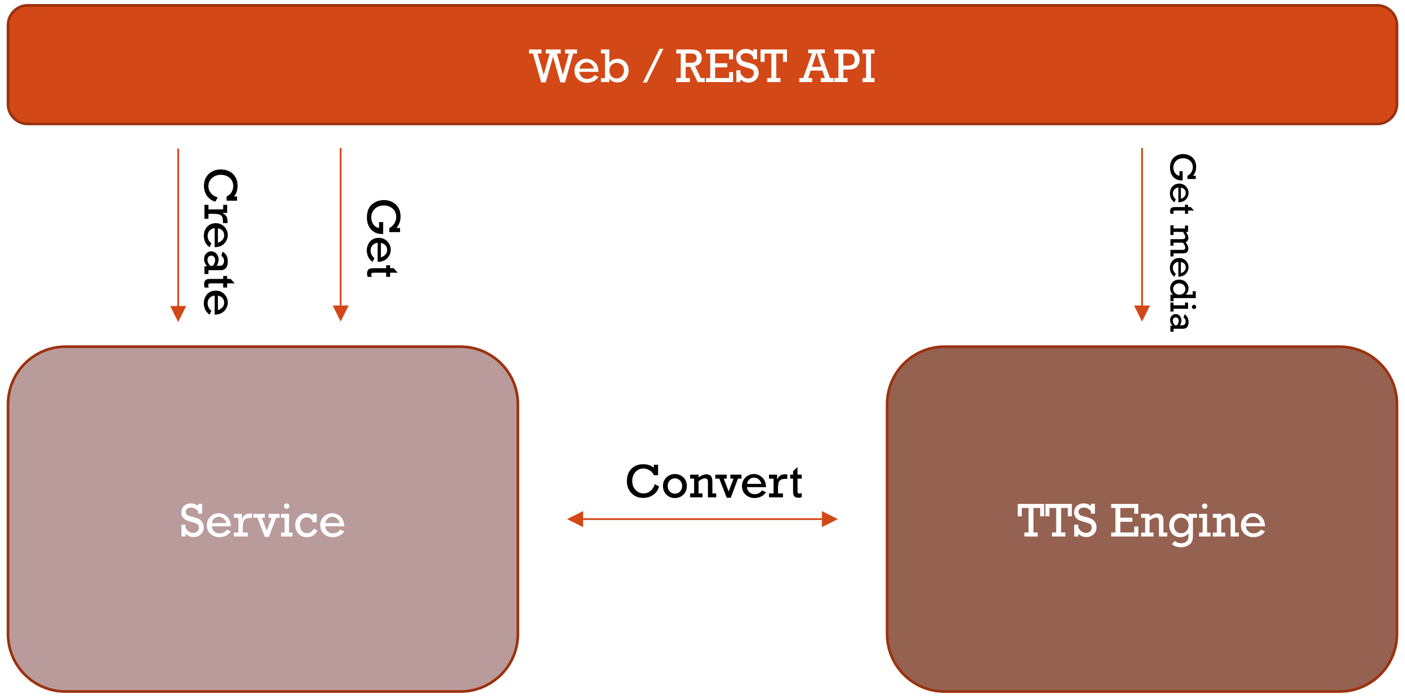
Error responses:

1. **Handled Errors:**
   Content-Type: application/json
   Status Code: 4xx/5xx
   Payload: {
      status: int,
      message: string,
      details: []string (optional)
   }

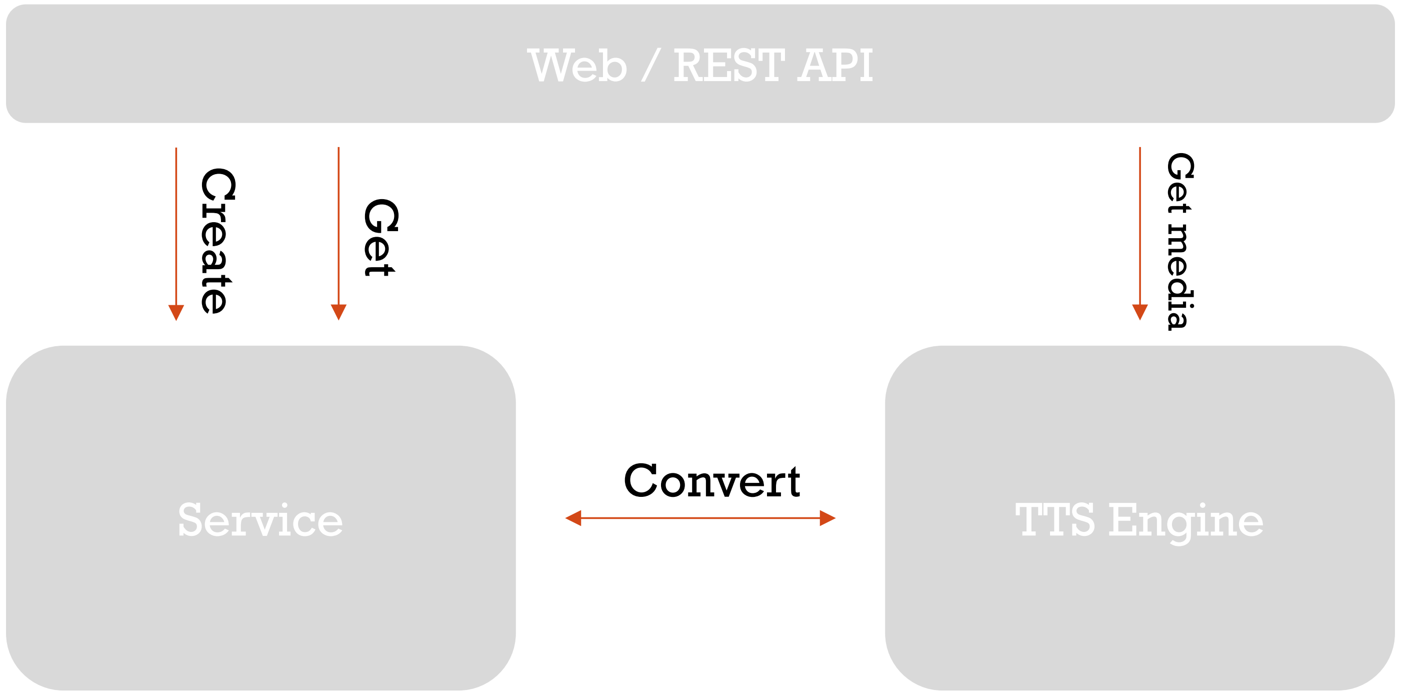
2. **Unhandled Errors** (In theory we should not have those ☺)**:**
   Status Code: 500
   Content-Type: text/plain
   Payload: A String (error message)

LIVE DEMO ☺

ARCHITECTURE OVERVIEW



ARCHITECTURE OVERVIEW



LET'S CODE ☺

SUMMARY

Go is an open source programming language that makes it easy to build **simple**, **reliable**, and **efficient** software.

Why would I use it?

- **strongly typed**
  - many errors found early
  - better IDE support
- **simple, yet powerful type system**
  - type inference
  - duck typing
  - functional programming features (first-class functions, higher order functions, function types)

SUMMARY

Continued…

- **Simple error handling**
  - Thanks to multiple returns from a function
- **Easy to use concurrency**
  - goroutines, channels
- **Built-in networking (client, server, HTTP)**
  - With testing libraries!

SUMMARY

Continued…

- **Fast**
  - Compiled to native code (no VM)
  - Low resource consumption (RAM, CPU)
  - Effective: „write/compile/test“ cycle is very short

QUIZ (WITH A PRIZE)

Extend the Service with DELETE functionality.

General rules:
- **It must work ☺**
- **Your code must be tested in every layer you change (web, service, media etc.)**
- **We wait for pull requests on GitHub**
- **You‘ve got two weeks (deadline: April 26th, 2017, 23:59:59 CET)**

Evaluation rules (in order of importance)
- **Correctness**
- **Test coverage**
- **Conformance to GO standards/conventions**
- **Overall code quality**

QUESTIONS ? / ANSWERS ☺