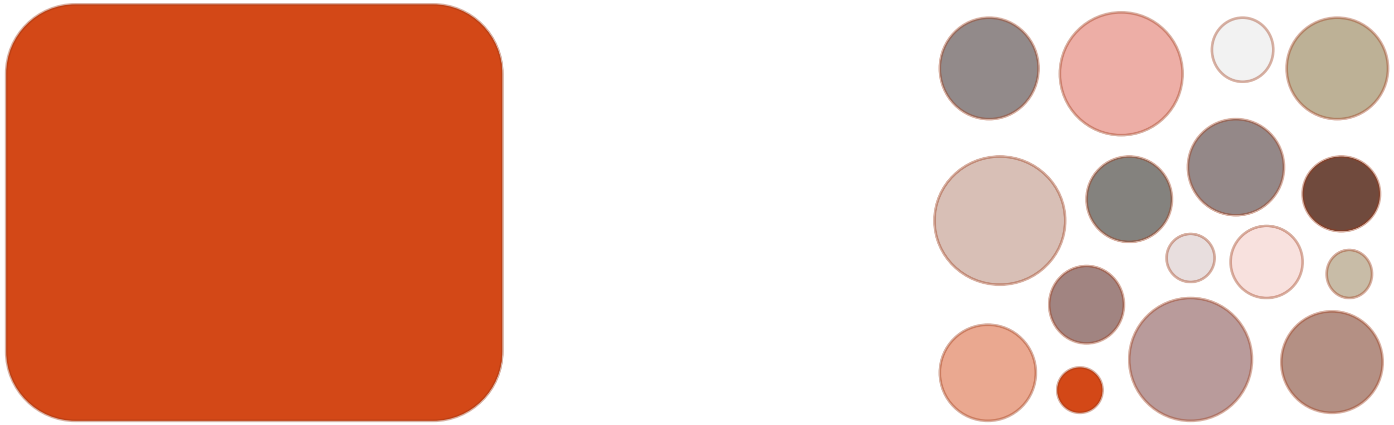
GO IN PRACTICE

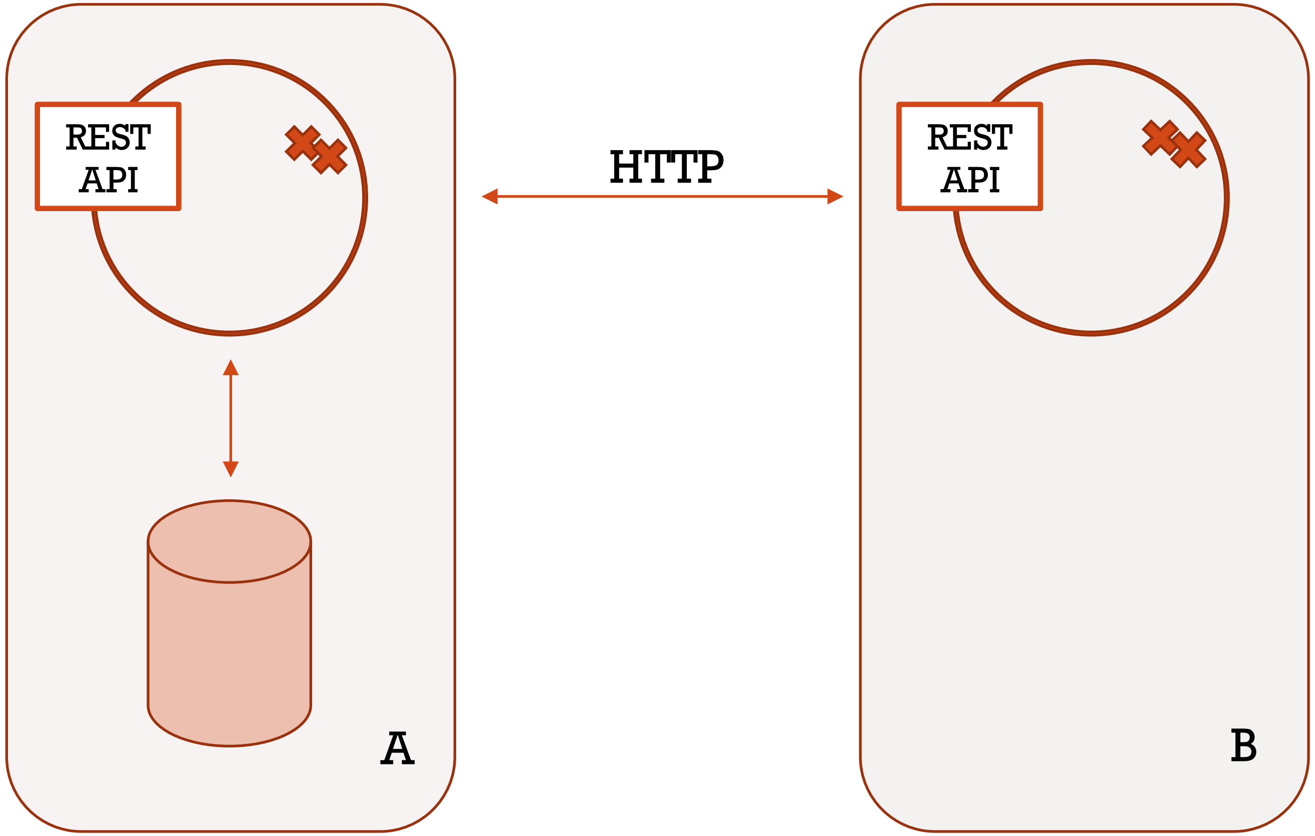
TEXT TO SPEECH MICROSERVICE

Tomasz Smelcerz
Paweł Sołtysek

MICROSERVICES ARCHITECTURE



MICROSERVICE ARCHITECTURE



OUR USE CASE

- TTS applications

- The problem we’re trying to solve

REST API

/tts

- **POST**: Create new TTS requests

Content-Type: application/json

Request: {
Text: string,
Language: "PL"/"EN"
}

Response: {
id: string,
text: string
language: "PL"/"EN"
status: "PENDING"/"READY"
mediaUrl: URL (optional, exists only if Status == READY)
}

/tts/ID

- **GET**: Get TTS request data

Response: As in POST

/media/ID

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

REST API, CD

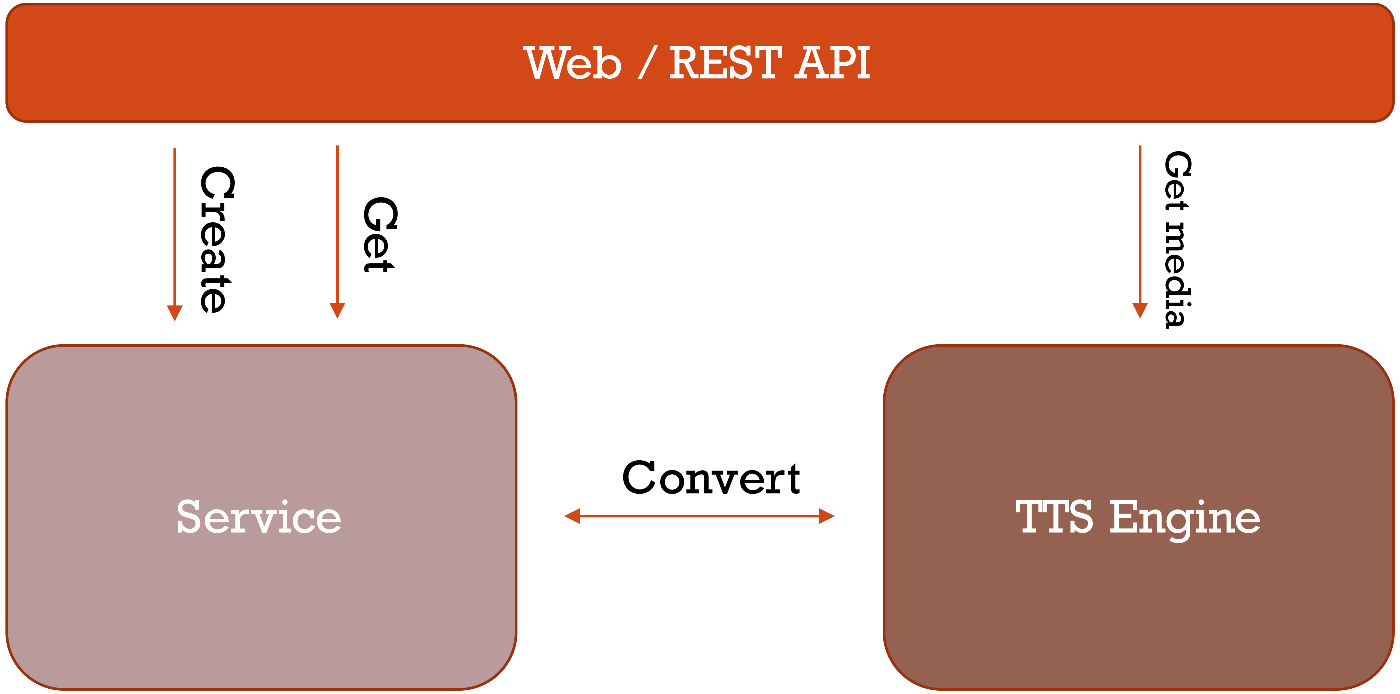
Error responses:

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

2. **Unhandled Errors:**
   Status Code: 5xx (usually 500)
   Content-Type: text/plain
   Payload: A String (error message)

LIVE DEMO ☺

ARCHITECTURE OVERVIEW



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: February 8th, 2017, 23:59:59 CET)

Evaluation rules (in order of importance)

- Correctness
- Test coverage
- Conformance to GO standards/idioms
- Overall code quality

QUESTIONS ? / ANSWERS ☺