Analysis and design of a platform for real-time speech translation

Jon Friis Jakobsen 123456789

Masters Project in Software Engineering

June 1887

The Maersk Mc-Kinney Moeller Institute
University of Southern Denmark

Abstract

Many companies are struggling to combine Java and Machine Learning...

Contents

1 Introduction

Background and Motivation

Beskriv behovet for sproglig inklusion og problemet med de nuværende proprietære løsninger fra Google og Microsoft.

Problem Statement

Den manglende forskning i arkitektoniske principper for realtids-systemer med lav latens.

Research Questions

Indsæt din hovedproblemstilling og de fem underspørgsmål (SQ1-SQ5).

Success Criteria

Beskriv de målbare parametre for svartid, kapacitet og kvalitet.

2 Background

State of the Art

Gennemgang af eksisterende løsninger (Google Translate, Microsoft Translator, AWS).

Microservices vs. Monoliths

Teori om hændelsesdrevet arkitektur (Newman, Fowler).

AI Components

Teoretisk baggrund for ASR (f.eks. Whisper), MT og TTS.

Event-Driven Principles

Hvorfor Kafka og asynkron kommunikation er valgt til realtidsdata.

3 Analysis and Design

System Architecture

Beskrivelse af Kafka-brokeren, Zookeeper og Confluent Schema Registry. Hvordan mikroservices kommunikerer asynkront via Kafka-emner.

Data Modeling

Dokumentation af dine Avro-schemas (BaseEvent.avsc, VoiceDetectedEvent.avsc) for at sikre type-sikkerhed.

End-to-End Pipeline

Indsæt dine Mermaid-sekvensdiagrammer, der viser flowet fra API Gateway gennem VAD, ASR, Translation og TTS.

Orchestration Logic

Hvordan din Pipeline Orchestrator styrer tilstanden og bruger Correlation IDs til tracking.

4 Implementation

Container Orchestration

Opsætning af Kubernetes (GKE/EKS) med GPU-optimerede node-pools til ML-modeller.

Scaling Strategies

Implementering af Horizontal Pod Autoscaler (HPA) og din prædiktive skaleringsmodel.

CI/CD & GitOps

Brug af ArgoCD til Blue-Green og Canary deployments.

Metrics Collection Setup

Implementering af Prometheus, Grafana og din "Metrics Collector Service" til opsamling af forskningsdata.

5 Evaluation

Performance Results

Præsentation af latens-målinger (mål: < 3 sekunder).

Scalability Testing

Grafer der viser systemets adfærd under stigende belastning (op til 1000+ events/sek).

Cross-Linguistic Analysis

Evaluering af hvordan forskellige sprogfamilier påvirker systemets præcision og hastighed.

Robustness Evaluation

Test af fejlscenarier (f.eks. service-nedbrud) og systemets evne til auto-recovery.

6 Conclusion

Summary

Sammenfatning af hvordan din event-drevne arkitektur løser problemet med skalering og latens.

Future Work

Forslag til forbedringer, såsom integration af flere sprog, forbedret fejlhåndtering og avancerede skaleringsalgoritmer.

A Appendix

We include here the API documentation for our library.

Bibliography

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, second edition, 2001.

[2] Tim Lindholm and Frank Yellin. *The Java^{TM} Virtual Machine Specification*. Addison-Wesley, 2nd edition, 1999.