SYDDANSK UNIVERSITET

Det Tekniske Fakultet

Mærsk MC-Kinney Møller Institute

ANALYSE OG DESIGN AF PLATFORM TIL REALTIDS TALEOVERSÆTTELSE

Student

Jon Friis Jakobsen
Exam Number: 475136

Indhold

1 Problem Statement

1.1 Background and Motivation

In a world characterized by increasing globalization and digitalization, the need for effective communication across languages is more urgent than ever. Real-time speech-to-speech translation has the potential to break down language barriers and create new opportunities for collaboration, knowledge sharing and access to information across borders [1,2]. This applies not only to large companies, but also to smaller actors, public institutions and civil society, where linguistic inclusion can have a major impact on participation and equality [3].

Current solutions in the real-time speech-to-speech translation market are primarily developed and owned by large technology companies such as Microsoft, Amazon and Google. These platforms are often embedded in larger, proprietary ecosystems, which can make them inaccessible, expensive or difficult to adapt to smaller actors, open source environments or specialized applications. At the same time, independent research on optimal architectural principles for such systems is lacking, especially around the balance between latency, scalability and robustness [4,5]. The market for real-time speech-to-speech translation is dominated by large technology companies such as Google (Alphabet Inc.), Microsoft Corporation and Amazon, which offer integrated solutions via Google Translate [6], Microsoft Translator [7] and AWS' combination of Transcribe, Translate and Polly [8–10], respectively.

Therefore, this project focuses on developing and evaluating an alternative architectural approach based on event-driven microservices [11–13], with a particular focus on optimizing the balance between latency, scalability and robustness in real-time speech-to-speech translation.

1.2 Problem Statement

Existing research on microservice architecture and real-time translation systems rarely addresses the complex integration of multiple AI components under stringent latency requirements. The literature focuses primarily on improving individual components (ASR models, translation quality) or monolithic end-to-end systems [14–17], but lacks systematic treatment of distributed architectures specifically designed for real-time speech-to-speech translation [4,5].

In particular, there is a need for empirical knowledge on how event-driven communication can be optimized to handle the sequential data flow from speech to speech through multiple processing steps, while allowing the system to scale horizontally and maintain high availability under varying load scenarios [18–20].

1.3 Main question

How can an event-driven microservice architecture be designed and implemented to deliver robust and scalable real-time speech-to-speech translation with measurable performance optimization?

1.4 Sub-questions

1. What architectural principles and design patterns support an efficient real-time speech translation pipeline?
2. What factors affect the system's latency and ability to scale?
3. How can these factors be measured and optimized in practice?
4. How can the system be made robust and available even if individual components fail?

5. How can the system's performance and quality be evaluated and possibly compared to existing solutions?

2 Research Contribution and Originality

The project will contribute as a reference architecture for event-driven real-time translation systems, where the central architectural principles and design patterns are analyzed and discussed based on the sub-question of precisely these choices and considerations. The empirical analysis of performance trade-offs in microservice-based AI pipelines is based on a systematic collection of data on the factors that affect the system's latency and scalability, as well as how these factors can be measured and optimized in practice. The identification of critical design patterns for low-latency integration of AI components is also included in the discussion of both architecture and performance. The development of a concrete platform and dissemination of best practices aim to achieve high robustness and enable a comparison with existing solutions, so that the system's strengths and weaknesses can be assessed in relation to these alternatives. Finally, evaluation and comparison form the framework for the work with benchmark datasets and evaluation methods, while the project's methodological contribution regarding systematic performance evaluation and measurement of robustness and scalability is integrated as a central part of the analysis.

2.1 Success criteria and evaluation parameters

The success of the project is evaluated by measuring the system's response times, capacity, availability and resource consumption during tests with increasing load. The results are compared with set requirements for response time, concurrent sessions and resource utilization. The translation quality is assessed both automatically and by human assessment in relation to commercial solutions. The system's implementation ability and documentation are tested by installation on a cloud platform, and the flexibility is tested by extension to a new language pair. All results are documented and discussed in the report.

2.2 Minimum Viable Product (MVP)

At a minimum, the project delivers an event-driven pipeline that enables live transcription and translation from one language to another in as close to real time as possible. The user speaks into the system, which transcribes the speech, translates the text and returns the translated text (possibly also as synthetic speech). The focus is on demonstrating low latency and robust event flow between the components, where the entire process from input to output occurs automatically and without manual intervention. The MVP must be demonstrable with integration of existing open source components for speech recognition, machine translation and text-to-speech.

3 Schedule

3.1 Scope

The project includes design and implementation of microservice architecture, integration of existing AI components, evaluation of performance and scalability, as well as deployment strategies for both cloud and on-premise. Development of new AI models, detailed security analysis, user interface design, legal aspects and in-depth cost optimization are outside the scope of the project.

Period	Activities	Deliverables
Sep-Oct	Literature study, requirements analysis	Literature review, Requirements specifications
Nov-Dec	Architecture design, technology evaluation	Technical architecture document
Jan-Mar	Implementation, integration testing	MVP demonstrator, working prototype, test suite
April	Performance testing, evaluation	Experiment results, benchmark data
May	Analysis, report writing	Final thesis, presentation

3.2 Expected benefits and perspectives

Academically, the project is expected to result in a master's thesis focusing on performance optimization in microservice-based AI systems, an open source reference implementation and contributions to relevant conferences. On a practical level, the project delivers a functional platform, documented best practices for deployment and a cost-effective alternative to proprietary solutions. Perspectives for future research include edge deployment, federated learning, integration of multimodal inputs and automatic scaling based on predicted demand.

3.3 Risk management

Technical risks include, among others, excessive latency, scalability issues and complex integration. These are countered with fallback solutions, focus on vertical scaling and prioritization of core functionality. Resource risks such as timeouts and lack of computing resources are handled through the use of pre-implemented components, utilization of cloud credits and local fallback options.

Litteratur

[1] (2024) Speech to speech translation market size & share analysis. Mordor Intelligence. Accessed: 2025-07-30. [Online]. Available: https://www.mordorintelligence.com/industry-reports/speech-to-speech-translation

[2] (2024) 10 stats & breakthroughs in ai speech translation in 2024. KUDO. Accessed: 2025-07-30, Blog post. [Online]. Available: https://kudo.ai/blog/the-10-most-important-statistics-breakthroughs-in-ai-speech-translation-from-2024/

[3] (2022) Inclusion in education. UNESCO. Tilgængelig: https://www.unesco.org/en/inclusion-education (besøgt juli 2025). [Online]. Available: https://www.unesco.org/en/inclusion-education

[4] P. Xu, A. Sharma, and J. Müller, "Scalability and maintainability challenges and solutions in machine learning: Systematic literature review," *arXiv preprint arXiv:2504.11079*, 2025. [Online]. Available: https://arxiv.org/abs/2504.11079

[5] S. Zhang, J. Chen, C.-H. Wang, W.-N. Hsu, K. Peng, S. Karita, Y. Qian, S. Watanabe, and W. Chan, "Direct speech to speech translation: A review," *arXiv preprint arXiv:2503.04799*, 2024. [Online]. Available: https://arxiv.org/abs/2503.04799

[6] (2024) Cloud translation. Google Cloud. Tilgængelig: https://cloud.google.com/translate (besøgt juli 2025). [Online]. Available: https://cloud.google.com/translate

[7] (2024) Microsoft translator. Microsoft. Tilgængelig: https://www.microsoft.com/da-dk/translator/ (besøgt juli 2025). [Online]. Available: https://www.microsoft.com/da-dk/translator/

[8] (2024) Amazon transcribe. Amazon Web Services. Tilgængelig: https://aws.amazon.com/transcribe/ (besøgt juli 2025). [Online]. Available: https://aws.amazon.com/transcribe/

[9] (2024) Amazon translate. Amazon Web Services. Tilgængelig: https://aws.amazon.com/translate/ (besøgt juli 2025). [Online]. Available: https://aws.amazon.com/translate/

[10] (2024) Amazon polly. Amazon Web Services. Tilgængelig: https://aws.amazon.com/polly/ (besøgt juli 2025). [Online]. Available: https://aws.amazon.com/polly/

[11] S. Newman, *Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith*. O'Reilly Media, 2019.

[12] C. Richardson, *Microservices Patterns: With Examples in Java*. Manning Publications, 2018.

[13] M. Fowler. (2014) Microservices: A definition of this new architectural term. Accessed: 2025-07-30. [Online]. Available: https://martinfowler.com/articles/microservices.html

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey *et al.*, "Whisper: Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[17] C. Wang, A. Wu, J. Pino, A. Baevski, M. Auli, and A. Conneau, "Fairseq s2t: Fast speech-to-text modeling with fairseq," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[18] M. Kleppmann, *Designing Data-Intensive Applications*. O'Reilly Media, 2017.

[19] (2023) Apache kafka documentation. Accessed: 2025-07-30. [Online]. Available: https://kafka.apache.org/documentation/

[20] (2023) Kubernetes documentation. Accessed: 2025-07-30. [Online]. Available: https://kubernetes.io/docs/