

Schlussbericht

SPEAKER: Aufbau einer führenden Sprachassistenten-
plattform „Made in Germany“



Zuwendungsempfänger: SPEAKER Konsortium

Förderkennzeichen: 01MK20011A

Laufzeit des Vorhabens: 01.04.2020 bis 30.09.2023

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Inhaltsverzeichnis

<i>Abbildungsverzeichnis</i>	4
<i>Tabellenverzeichnis</i>	6
<i>Literaturverzeichnis</i>	6
<i>1 Aufgabenstellung</i>	11
<i>2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde</i>	11
<i>3 Planung und Ablauf des Vorhabens</i>	17
<i>4 Anknüpfung an wissenschaftlichen und technischen Stand</i>	20
4.1 Akustische Vorverarbeitung (Acoustic Frontend, AFE)	20
4.2 Automatische Spracherkennung (Automatic Speech Recognition, ASR)	20
4.3 Keyword / Wakeword Detektor (KWS)	21
4.4 Sprechererkennung (Speaker Recognition, SR)	22
4.5 Natural Language Understanding (NLU) / Dialog Management (DM)	23
4.6 Machine Reading Comprehension (MRC).....	23
4.7 Question Answering über Knowledge Graphen (KGQA).....	24
4.8 Sprachsynthese (Text-to-Speech, TTS).....	24
4.9 Emotionserkennung	25
4.10 Natural Language Generation	26
4.11 Audio Visuelle Spracherkennung	27
4.12 Question Answering über Knowledge Graphen.....	27
4.13 Storytelling	28
4.14 Offline Question Answering	28
4.15 Cloud-Skalierungsfähigkeiten mit zusätzlichen Metriken	29
<i>5 Zusammenarbeit mit anderen Stellen</i>	31
<i>6 Ergebnisse des SPEAKER Projekts</i>	32
6.1 Anforderungsanalyse, Architektur, Benchmarks	32

6.2	Geschäftliche Plattformentwicklung & Community Management	32
6.3	Technische Modul- sowie Plattformentwicklung	35
6.3.1	Entwicklung der SPEAKER Plattform	35
6.3.2	Entwicklung der SPEAKER Web Frontends	40
6.3.3	Audiovisuelle Spracherkennung	50
6.3.4	Akustische Vorverarbeitung	54
6.3.5	Keyword Erkennung	55
6.3.6	Sprechererkennung	57
6.3.7	Automatische Spracherkennung (ASR)	60
6.3.8	Natural Language Understanding (NLU)	61
6.3.9	Dialog Management (DM)	64
6.3.10	Text to Speech Sprachsynthese (TTS)	66
6.3.11	Machine Reading Comprehension	73
6.3.12	Knowledge Graph Question Answering	75
6.4	Tools und Daten	76
6.5	Evaluation in Anwendungsfällen und Modulentwicklung	77
6.5.1	Comma Soft AG – Sprachbasiertes Gesundheitslogbuch	78
6.5.2	DATEV eG – Sprachassistent im Kundenservice	79
6.5.3	Deutsche Bahn AG – Sprachassistenten in der Instandhaltung	81
6.5.4	SAP SE – Transkription von Live Events	82
6.5.5	SAP SE – Spracherkennung für Botbau Plattform	83
6.5.6	IAV – Sprachassistent für die Fahrzeugentwicklung	84
6.5.7	ONSEI – Smart Speaker in Hotels	85
6.5.8	Retresco – Natürlichsprachliche Textgenerierung	88
6.5.9	Siemens – Verschiedene Szenarien im industriellen Umfeld	90
6.6	Projektmanagement und Öffentlichkeitsarbeit	98
6.7	Notwendigkeit und Angemessenheit der geleisteten Arbeit	100
6.8	Voraussichtlicher Nutzen und Verwertbarkeit	101
6.9	Fortschritte von Sprachtechnologien im Förderzeitraum	103
6.10	Geplante und erfolgte Veröffentlichungen	103
<i>Anhang</i>		<i>107</i>

Abbildungsverzeichnis

Abb. 1: Darstellung der Konsortialstruktur des SPEAKER Projekts.....	11
Abb. 2: Krake Beispielgrafik für verteilten Workload basierend auf Energieeffizienz-bezogene Kriterien	30
Abb. 3: SPEAKER Key-Visual	34
Abb. 4: Übersicht über die SPEAKER Plattform mit deren unterschiedlichen Komponenten	36
Abb. 5: CI/CD Pipelines zum Erstellen der Docker Container und dem Deployment im Kubernetes Cluster	37
Abb. 6: Unterschiedliche Umgebungen um kontinuierliches Entwickeln sowie Deployment sicherzustellen.....	38
Abb. 7: Screenshot der Landing Page des Evaluations Web Frontends der SPEAKER Services: SPEAKER Stage.....	40
Abb. 8: SPEAKER Stage – Sprachsynthese	41
Abb. 9: SPEAKER Stage – Automatische Spracherkennung	42
Abb. 10: SPEAKER Stage – Dialogmanagers.....	43
Abb. 11: SPEAKER Stage – Sprechererkennung – Sprecherregistrierung	44
Abb. 12: SPEAKER Stage – Sprechererkennung – Sprecherklassifizierung	44
Abb. 13: SPEAKER Stage – Keyword Spotter – Keyword Konfiguration	45
Abb. 14: SPEAKER Stage – Keyword Spotter.....	46
Abb. 16: Konfigurationsmöglichkeiten im Assistant Tab	47
Abb. 15: Screenshot der WebApp zum Testen eines Variabel zusammenstellbaren Gesamtsystems	47
Abb. 17: Ausschnitt eines Dialoggraphs des Dialogeditors	48
Abb. 18: Build Pipeline zum Erzeugen von Dialog Managern aus dem Dialog Editor heraus	49
Abb. 19: Integriertes Interface zum Testen von Dialogmanagern	50
Abb. 20: Die echtzeitfähige Bildvorverarbeitung im Demonstrator	50
Abb. 21: WER (%) Vergleich von audiovisuellen Modellen ohne und mit selbstüberwachtem Vortraining (SSL) unter Babble Noise (Diffuser Stimmteppich, Beispiel: Cocktailparty)	52
Abb. 22: Mit dem interaktiven Demonstrator kann audiovisuelle Spracherkennung individuell ausprobiert werden. Gezeigt werden jeweils vier Hypothesen (Sätze) nach Erkennung auf Audio-Basis (oben), Video-Basis (Mitte), und mit audiovisueller Spracherkennung (unten)	53
Abb. 23: Auswirkungen der AFE auf die Wortfehlerrate der SPEAKER ASR bei in Abhängigkeit von verschiedenen Stärken der Störgeräusche und Dämpfungsggrade.....	55
Abb. 24: Schematische Darstellung der Verarbeitungsschritte des KWS-Moduls.....	56
Abb. 25: Evaluationsergebnisse verschiedener neuronaler SR Architekturen	59

Abb. 26: Güte der Trennung zwischen registrierten und unbekannten Speaker Embeddings in Abhängigkeit von der Dauer des Sprachsignals.....	60
Abb. 27: Überblick über die Modelle, die für die Umwandlung von geschriebenem Text in Sprache notwendig sind.	67
Abb. 28: Minimierung der Modellgröße und Erhöhung der Effizienz der SPEAKER TTS während des Projektzeitraums.	69
Abb. 29: Überblick über die multi-lingualen Erweiterungen des Akustischen Modells zum	72
Abb. 30: Anwendungsfall Sprachassistentz im Service	80
Abb. 31: Prototypische Anwendung zur Transkription von virtuellen Events.....	83
Abb. 32: SPEAKER Technologien als mögliches Sprachinterface für Chatbots.....	83
Abb. 33: Tutorial zur Anbindung eines Sprachinterfaces an Chatbots	84
Abb. 34: Das Tool SCD (speech control demonstrator), welches als Stand-Alone Software die Messtools INCA und CANape mit einer Sprachsteuerung nachrüstet	85
Abb. 35: Illustration des Smart Speakers Melody für die Hotellerie.....	86
Abb. 36: Flyer zum Smart Speaker Melody	87
Abb. 37: Schematische Darstellung der Anbindung von textengine.io mit Hilfe des Dialog Editors	90
Abb. 38: Web Frontend zur Konfiguration der textengine.io.....	90
Abb. 39: Ergebnisse des Benchmarkings mit Utterances aus der Domäne „Digital Factory“	92
Abb. 40: Demo-Szenario Servicetechniker	93
Abb. 41: Demo- Szenario wikiGPT.....	94

Tabellenverzeichnis

Tabelle 1: Überblick über die geförderten Partner im SPEAKER Konsortium, sowie deren Aufgaben	12
Tabelle 2: Meilensteine des SPEAKER Projektvorhabens	17
Tabelle 3: Veranstaltungen zur Bekanntmachung der SPEAKER Plattform	34
Tabelle 4: Stattgefundene Workshops und Vorträge	35
Tabelle 5: Pilotanwendungen ausgewählter Projektpartner	94
Tabelle 6: Stattgefundene Treffen des Konsortiums	98
Tabelle 7: Übersicht der assoziierten Partner	108

Literaturverzeichnis

- [1] Y. Hu u. a., „DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement“, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, und T. F. Zheng, Hrsg., ISCA, 2020, S. 2472–2476. doi: 10.21437/INTER-SPEECH.2020-2537.
- [2] J.-M. Valin, U. Isik, N. Phansalkar, R. Giri, K. Helwani, und A. Krishnaswamy, „A Perceptually-Motivated Approach for Low-Complexity, Real-Time Enhancement of Fullband Speech“, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, und T. F. Zheng, Hrsg., ISCA, 2020, S. 2482–2486. doi: 10.21437/INTER-SPEECH.2020-2730.
- [3] M. Strake, B. Defraene, K. Fluyt, W. Tirry, und T. Fingscheidt, „INTERSPEECH 2020 Deep Noise Suppression Challenge: A Fully Convolutional Recurrent Network (FCRN) for Joint Dereverberation and Denoising“, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, und T. F. Zheng, Hrsg., ISCA, 2020, S. 2467–2471. doi: 10.21437/INTER-SPEECH.2020-2439.
- [4] S. Alharbi u. a., „Automatic Speech Recognition: systematic literature review“, *IEEE Access*, Bd. 9, S. 131858–131876, 2021, doi: 10.1109/ACCESS.2021.3112535.
- [5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, und I. Sutskever, „Robust Speech Recognition via Large-Scale Weak Supervision“, in *International Conference on Machine Learning (ICML)*, in *Proceedings of Machine Learning Research*, vol. 202. 2023, S. 28492–28518.
- [6] M. Burchi und V. Vielzeuf, „Efficient Conformer: Progressive Downsampling and Grouped Attention for Automatic Speech Recognition“, in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, S. 8–15. doi: 10.1109/ASRU51503.2021.9687874.
- [7] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, und J. Jensen, „Deep Spoken Keyword Spotting: An Overview“, *IEEE Access*, Bd. 10, S. 4169–4199, 2022, doi: 10.1109/ACCESS.2021.3139508.
- [8] G. Chen, C. Parada, und G. Heigold, „Small-footprint keyword spotting using deep neural networks“, in *IEEE International Conference on Acoustics, Speech and Signal Processing*,

- ICASSP 2014, Florence, Italy, May 4-9, 2014, IEEE, 2014, S. 4087–4091. doi: 10.1109/ICASSP.2014.6854370.
- [9] T. N. Sainath und C. Parada, „Convolutional neural networks for small-footprint keyword spotting“, in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, ISCA, 2015, S. 1478–1482. doi: 10.21437/INTERSPEECH.2015-352.
- [10] R. Tang und J. Lin, „Deep Residual Learning for Small-Footprint Keyword Spotting“, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, IEEE, 2018, S. 5484–5488. doi: 10.1109/ICASSP.2018.8462688.
- [11] Y. Zhuang, X. Chang, Y. Qian, und K. Yu, „Unrestricted Vocabulary Keyword Spotting Using LSTM-CTC“, in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, N. Morgan, Hrsg., ISCA, 2016, S. 938–942. doi: 10.21437/INTERSPEECH.2016-753.
- [12] C. T. Lengerich und A. Y. Hannun, „An End-to-End Architecture for Keyword Spotting and Voice Activity Detection“, *CoRR*, Bd. abs/1611.09405, 2016.
- [13] T. Bluche, M. Primet, und T. Gisselbrecht, „Small-Footprint Open-Vocabulary Keyword Spotting with Quantized LSTM Networks“, *CoRR*, Bd. abs/2002.10851, 2020.
- [14] B. Kim, M. Lee, J. Lee, Y. Kim, und K. Hwang, „Query-by-Example On-Device Keyword Spotting“, in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, IEEE, 2019, S. 532–538. doi: 10.1109/ASRU46091.2019.9004014.
- [15] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, und S. Khudanpur, „X-Vectors: Robust DNN Embeddings for Speaker Recognition“, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, S. 5329–5333. doi: 10.1109/ICASSP.2018.8461375.
- [16] M. Ravanelli und Y. Bengio, „Speaker Recognition from Raw Waveform with SincNet“, in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, IEEE, 2018, S. 1021–1028. doi: 10.1109/SLT.2018.8639585.
- [17] L. Wan, Q. Wang, A. Papir, und I. López-Moreno, „Generalized End-to-End Loss for Speaker Verification“, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, IEEE, 2018, S. 4879–4883. doi: 10.1109/ICASSP.2018.8462665.
- [18] T. Bian, F. Chen, und L. Xu, „Self-Attention Networks for Text-Independent Speaker Verification“, in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, S. 3955–3960. doi: 10.1109/CCDC.2019.8833466.
- [19] J. Jung, H.-S. Heo, J. Kim, H. Shim, und H.-J. Yu, „RawNet: Advanced End-to-End Deep Neural Network Using Raw Waveforms for Text-Independent Speaker Verification“, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin und Z. Kacic, Hrsg., ISCA, 2019, S. 1268–1272. doi: 10.21437/INTERSPEECH.2019-1982.
- [20] J. Jung, S. Kim, H. Shim, J. Kim, und H.-J. Yu, „Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms“, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, und T. F. Zheng, Hrsg., ISCA, 2020, S. 1496–1500. doi: 10.21437/INTERSPEECH.2020-1011.
- [21] B. Desplanques, J. Thienpondt, und K. Demuyndck, „ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification“, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, und T. F. Zheng, Hrsg., ISCA, 2020, S. 3830–3834. doi: 10.21437/INTERSPEECH.2020-2650.

- [22]Z.-H. Ling u. a., „Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends“, *IEEE Signal Process Mag*, Bd. 32, Nr. 3, S. 35–52, 2015, doi: 10.1109/MSP.2014.2359987.
- [23]A. van den Oord u. a., „Parallel WaveNet: Fast High-Fidelity Speech Synthesis“, in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, J. G. Dy und A. Krause, Hrsg., in *Proceedings of Machine Learning Research*, vol. 80. PMLR, 2018, S. 3915–3923. [Online]. Verfügbar unter: <http://proceedings.mlr.press/v80/oord18a.html>
- [24]J.-M. Valin und J. Skoglund, „LPCNet: Improving Neural Speech Synthesis Through Linear Prediction“, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, Mai 2019, S. 5891–5895.
- [25]R. Prenger, R. Valle, und B. Catanzaro, „WaveGlow: A Flow-based Generative Network for Speech Synthesis“, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, Mai 2019, S. 3617–3621.
- [26]Y. Ren u. a., „FastSpeech: Fast, Robust and Controllable Text to Speech“, in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, und R. Garnett, Hrsg., 2019, S. 3165–3174.
- [27]I. Elias u. a., „Parallel Tacotron: Non-Autoregressive and Controllable TTS“, *CoRR*, Bd. abs/2010.11439, 2020, [Online]. Verfügbar unter: <https://arxiv.org/abs/2010.11439>
- [28]Y. Zhang u. a., „Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning“, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin und Z. Kacic, Hrsg., ISCA, 2019, S. 2080–2084. doi: 10.21437/INTERSPEECH.2019-2668.
- [29]J. Wagner u. a., „Dawn of the Transformer Era in Speech Emotion Recognition: Closing the Valence Gap“, *IEEE Trans Pattern Anal Mach Intell*, Bd. 45, Nr. 9, S. 10745–10759, 2023, doi: 10.1109/TPAMI.2023.3263585.
- [30]E. Reiter und R. Dale, „Building Natural Language Generation Systems“, *Nat Lang Eng*, Bd. 7, Nr. 3, S. 271–274, 2001, doi: 10.1017/S1351324901212704.
- [31]A. Gatt und E. Krahmer, „Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation“, *J Artif Intell Res*, Bd. 61, S. 65–170, 2018, doi: 10.1613/JAIR.5477.
- [32]A. Vaswani u. a., „Attention is All you Need“, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, und R. Garnett, Hrsg., 2017, S. 5998–6008. [Online]. Verfügbar unter: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- [33]T. Afouras, J. S. Chung, A. W. Senior, O. Vinyals, und A. Zisserman, „Deep Audio-Visual Speech Recognition“, *CoRR*, Bd. abs/1809.02108, 2018, [Online]. Verfügbar unter: <http://arxiv.org/abs/1809.02108>
- [34]B. Shi, W.-N. Hsu, K. Lakhotia, und A. Mohamed, „Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction“, in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Verfügbar unter: <https://openreview.net/forum?id=Z1Qlm11uOM>
- [35]T. Afouras, J. S. Chung, und A. Zisserman, „LRS3-TED: a large-scale dataset for visual speech recognition“, *CoRR*, Bd. abs/1809.00496, 2018, [Online]. Verfügbar unter: <http://arxiv.org/abs/1809.00496>

- [36] B. Shi, W.-N. Hsu, und A. Mohamed, „Robust Self-Supervised Audio-Visual Speech Recognition“, in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko und J. H. L. Hansen, Hrsg., ISCA, 2022, S. 2118–2122. doi: 10.21437/INTERSPEECH.2022-99.
- [37] C. Raffel u. a., „Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer“, *J. Mach. Learn. Res.*, Bd. 21, Nr. 140, S. 1–67, 2020.
- [38] P. Bhargava, A. Drozd, und A. Rogers, „Generalization in NLI: Ways (Not) To Go Beyond Simple Heuristics“. 2021.
- [39] I. Turc, M.-W. Chang, K. Lee, und K. Toutanova, „Well-Read Students Learn Better: The Impact of Student Initialization on Knowledge Distillation“, *CoRR*, Bd. abs/1908.08962, 2019, [Online]. Verfügbar unter: <http://arxiv.org/abs/1908.08962>
- [40] A. Perevalov, D. Diefenbach, R. Usbeck, und A. Both, „QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers“, in *16th IEEE International Conference on Semantic Computing, ICSC 2022, Laguna Hills, CA, USA, January 26-28, 2022*, IEEE, 2022, S. 229–234. doi: 10.1109/ICSC52841.2022.00045.
- [41] D. Diefenbach, P. H. Migliatti, O. Qawasmeh, V. Lully, K. Singh, und P. Maret, „QAnswer: A Question Answering prototype bridging the gap between a considerable part of the LOD cloud and end-users“, in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, und L. Zia, Hrsg., ACM, 2019, S. 3507–3510. doi: 10.1145/3308558.3314124.
- [42] „KRAKE“. [Online]. Verfügbar unter: <https://krake.cloud/>
- [43] M. Anwar, B. Shi, V. Goswami, W.-N. Hsu, J. Pino, und C. Wang, „MuAViC: A Multilingual Audio-Visual Corpus for Robust Speech Recognition and Robust Speech-to-Text Translation“, *CoRR*, Bd. abs/2303.00628, 2023, doi: 10.48550/ARXIV.2303.00628.
- [44] Z. Li, T. Graave, J. Liu, T. Lohrenz, S. Kunzmann, und T. Fingscheidt, „Parameter-Efficient Cross-Language Transfer Learning for a Language-Modular Audiovisual Speech Recognition“, in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023, Taipei, Taiwan, December 16-20, 2023*, IEEE, 2023, S. 1–8. doi: 10.1109/ASRU57964.2023.10389795.
- [45] Z. Li, „Real-Time Interactive Demonstrator for Audiovisual Speech Recognition and Lip Reading“, in *Proc. of ICASSP Show&Tell*, Rhodes, Greece, Juni 2023.
- [46] T. Lohrenz, Z. Li, und T. Fingscheidt, „Multi-Encoder Learning and Stream Fusion for Transformer-Based End-to-End Automatic Speech Recognition“, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, und P. Motlíček, Hrsg., ISCA, 2021, S. 2846–2850. doi: 10.21437/INTERSPEECH.2021-555.
- [47] T. Lohrenz, P. Schwarz, Z. Li, und T. Fingscheidt, „Multi-Head Fusion Attention for Transformer-Based End-to-End Automatic Speech Recognition“, in *14th ITG Conference on Speech Communication, online, September 29 - October 1, 2021*, VDE, 2021, S. 1–5. [Online]. Verfügbar unter: <https://ieeexplore.ieee.org/document/9657498>
- [48] Z. Li, T. Lohrenz, M. Dunkelberg, und T. Fingscheidt, „Transformer-Based Lip-Reading with Regularized Dropout and Relaxed Attention“, in *IEEE Spoken Language Technology Workshop, SLT 2022, Doha, Qatar, January 9-12, 2023*, IEEE, 2022, S. 723–730. doi: 10.1109/SLT54892.2023.10023442.
- [49] T. Lohrenz, B. Möller, Z. Li, und T. Fingscheidt, „Relaxed Attention for Transformer Models“, in *International Joint Conference on Neural Networks, IJCNN 2023, Gold Coast, Australia, June 18-23, 2023*, IEEE, 2023, S. 1–10. doi: 10.1109/IJCNN54540.2023.10191643.
- [50] Z. Li, C. Liang, T. Lohrenz, M. Sach, B. Möller, und T. Fingscheidt, „An Efficient and Noise-Robust Audiovisual Encoder for Audiovisual Speech Recognition“, in *Proc. of Interspeech*, 2023, S. 1583–1587.

- [51]S. S. Shetu, S. Chakrabarty, O. Thiergart, und E. Mabande, „Ultra Low Complexity Deep Learning Based Noise Suppression“. arXiv, 13. Dezember 2023. Zugriffen: 26. Februar 2024. [Online]. Verfügbar unter: <http://arxiv.org/abs/2312.08132>
- [52]S. Asaadi, Z. Kolagar, A. Liebel, und A. Zarcone, „GiCCS: A German in-Context Conversational Similarity Benchmark“, in *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, 2022, S. 351–362.
- [53]A. Sauer, S. Asaadi, und F. Küch, „Knowledge Distillation Meets Few-Shot Learning: An Approach for Few-Shot Intent Classification Within and Across Domains“, in *Proceedings of the 4th Workshop on NLP for Conversational AI, ConvAI@ACL 2022, Dublin, Ireland, May 27, 2022*, B. Liu, A. Papangelis, S. Ultes, A. Rastogi, Y.-N. Chen, G. Spithourakis, E. Nouri, und W. Shi, Hrsg., Association for Computational Linguistics, 2022, S. 108–119. doi: 10.18653/V1/2022.NLP4CONVAI-1.10.
- [54]„RASA“. [Online]. Verfügbar unter: www.rasa.com
- [55]P. Govalkar u. a., „A Lightweight Neural TTS System for High-quality German Speech Synthesis“, in *Proceedings of the ITG Conference on Speech Communication*, virtual, 2021, S. 39–43. [Online]. Verfügbar unter: <https://ieeexplore.ieee.org/document/9657503>

1 Aufgabenstellung

Im Rahmen der SPEAKER Umsetzungsphase bestand die Aufgabenstellung im Aufbau einer Sprachassistentenplattform "Made in Germany". Hierbei wurde eine Plattform entwickelt, welche insbesondere im professionellen Umfeld Anwendung findet. Daher wurde die Plattform zielgerichtet den Anforderungen der Industrie und Business-to-Business (B2B) Anwendungen zugeschnitten und deren speziellen Bedürfnisse berücksichtigt. Hierzu zählen Datensouveränität, ein transparenter und sicherer Umgang mit Nutzerdaten, domänenspezifische Anpassbarkeit, Flexibilität beim Produktivbetrieb, einheitliche Schnittstellen, sowie attraktive Lizenzmodelle. Schon während der Projektlaufzeit wurde die Plattform und die Technologiemodule in verschiedenen Anwendungsfällen der Projektpartner erprobt.

2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Die Konzeptphase wurde vom 15. April 2019 bis zum 15. Oktober 2019 durchgeführt. Die Umsetzungsphase, welche hier beschrieben wird, folgte vom 01. April 2020 bis 30. September 2023.

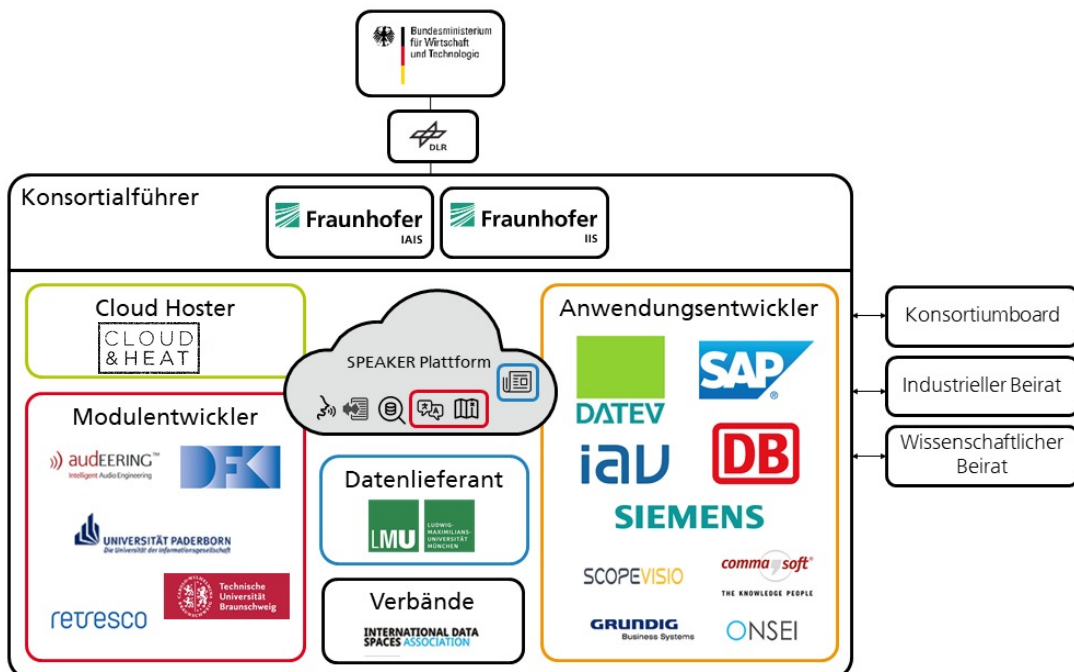


Abb. 1: Darstellung der Konsortialstruktur des SPEAKER Projekts

Verschiedene Partner haben an dem Projekt zum Aufbau einer Sprachassistentenplattform „Made in Germany“ mitgewirkt. Die Konsortialleitung übernahmen die Fraunhofer-Institute IAIS und IIS. Zudem bestand das Konsortium aus 16 Verbundpartnern und einer während der Projektlaufzeit stetig anwachsenden Anzahl assoziierter Partner. Eine tabellarische Auflistung der assoziierten Partner ist in Tabelle 7 im Anhang zu finden.

Die nachfolgende Tabelle 1 gibt einen Überblick über die Verbundpartner und der Aufgabenverteilungen während der Umsetzungsphase des Projekts.

Tabelle 1: Überblick über die geförderten Partner im SPEAKER Konsortium, sowie deren Aufgaben

Partner	Rolle	Aufgabe
Konsortialführung		
Fraunhofer IAIS	Im Fraunhofer IAIS sind führende Algorithmen im Bereich der Spracherkennung und der automatisierten Beantwortung von Fragen entwickelt worden. Die Arbeiten von Prof. Lehmann (Leiter des Bereiches Forschung in SPEAKER) und seiner Gruppe im Bereich Fragebeantwortung und Wissensmanagement werden jährlich mehrere tausend Male zitiert und haben zu zahlreichen internationalen Auszeichnungen geführt. Einige der für das Projekt notwendigen Basistechnologien befinden sich bereits im kommerziellen Einsatz bei führenden Partnern aus Industrie und Rundfunkanstalten. Darüber hinaus verfügt das Fraunhofer IAIS im Rahmen des EU Projekts AI4EU (Aufbau einer Europäischen KI-On-Demand-Plattform) über die notwendige Erfahrung, komplexe Plattformstrukturen zu etablieren.	Siehe Abschnitt 2.1
Fraunhofer IIS	Die Forschung und Entwicklung am Fraunhofer IIS konzentriert sich auf zwei Leitthemen: Audio- und Medientechnologien und Kognitive Sensorik. In den Bereichen Audio- und Sprachsignalverarbeitung gehört das Fraunhofer IIS zu den weltweit führenden Forschungseinrichtungen, die bereits mehrfach Ergebnisse der Spitzenforschung in kommerzielle äußerst erfolgreiche Produkte, wie z.B. die Audiokompressions-Formate MP3 und AAC überführt hat. Auf dem Gebiet der Audiovorverarbeitung und Sprachsignalverbesserung konnte das Fraunhofer IIS bereits eine weltmarktführende Position etablieren und ist in mehreren kommerziell erhältlichen Smart-Speakern integriert. Diese	Siehe Abschnitt 2.1

Partner	Rolle	Aufgabe
	Technologie ist die Grundlage für hohe Zuverlässigkeit und Robustheit von Sprachverarbeitung.	
Industriepartner		
AudeERING GmbH	Die AudeERING GmbH wurde 2012 als Spin-Off der TU München gegründet und ist Träger des Innovationspreises Bayern 2018. AudeERING ist als einziges europäisches Unternehmen Innovationstreiber im Bereich emotionaler künstlicher Intelligenz mit Fokus auf intelligente Audioanalyse. AudeERING hat eine Technologiekomponente zur Emotionsdetektion aus Sprachsignalen in die SPEAKER-Plattform eingebracht und somit ein einzigartiges Alleinstellungsmerkmal im Vergleich zu vorhandenen Plattformen geschaffen.	AP1: Architekturplanung AP5: Anpassen der Datenkonnectoren AP6: Entwicklung der Module und entsprechenden Anwendungsfälle Cloudbasierte und Hybride Emotionserkennung AP7: Reporting
Comma Soft AG	Die Comma Soft AG ist eine 1989 gegründete Bonner Unternehmerberatung, welche auf die digitale Transformation in Unternehmen spezialisiert ist. Sie brachte Erfahrung in Datenwissenschaften, insbesondere Smart Data und Predictive Analytics, in das SPEAKER-Projekt ein. In ihrem Anwendungsfall wurde ein sprachbasiertes Gesundheitslogbuch realisiert, welches datenschutzkonform personenbezogene Gesundheitsdaten verarbeitet.	AP1: Erarbeitung des Konzepts AP2: Marketing und Öffentlichkeitsarbeit AP6: Generierung anwendungsbezogener Testdaten, implementieren von Algorithmen zur Erkennung von gesundheitsrelevanten Auffälligkeiten im Gesundheitstagebuch, sowie das Umsetzen einer prototypischen Anwendung sowie die Durchführung von Feldstudien. AP7: Projektmanagement / Reporting
DATEV e.G.	Die DATEV e.G. zählt zu den größten IT-Dienstleistern und Softwarehäusern in Deutschland und erreicht mit ihren Lösungen mehr als 2,5 Mio. deutsche Unternehmen, Kommunen sowie Vereine. Im SPEAKER-Projekt ist das Ziel, mit Steuerberatern und Rechtsanwälten den Einsatz von interaktiven, multimodalen Sprachassistenten zu gestalten. Hierbei ist DSGVO-Konformität gefordert, sowie die Fähigkeit domänenspezifisches Wissen zu verarbeiten.	AP1: Anforderungsanalyse, Architekturspezifikation und Kuratierungsprozess AP2: Teilnahme an Öffentlichkeits- und Netzwerkaktivitäten AP5: Erstellung und Validierung eines Demonstrators AP6: Umsetzung eines Sprachassistenten für Anwaltskanzleien AP7: Reporting
Deutsche Bahn AG	Die DB AG ist ein führender Anbieter im Bereich Mobilität und Logistik. Sie wird einen Anwendungsfall im Bereich Logistik einbringen, bei dem die SPEAKER-Plattform einen Sprachassistenten zur Unterstützung bei Wartungsaufgaben umsetzen wird. Das Anwendungsspektrum in der Instandhaltung der Deutschen Bahn erstreckt sich hierbei von der Unterstützung bei einfachen Tätigkeiten zur Datenerfassung im Betrieb über aufwendige Wartungs- und Inspektionstätigkeiten in der Instandhaltung bis hin zu komplexen Ursachenanalysen.	AP1: Anforderungsanalyse und Spezifikation definieren AP2: Begleiten der geschäftlichen Plattformentwicklung AP3: Begleitung der technischen Plattformentwicklung AP4: Begleitung der technischen Entwicklung in den Sprachmodulen AP5: Begleitung der Bereitstellung von Daten und Durchführen von Voice User Experience Tests AP6: Implementieren eines Sprachassistenten zur Vereinfachung von Inspektions-/Befundungsprozessen AP7: Reporting

Partner	Rolle	Aufgabe
Grundig Business Systems GmbH	Die Grundig Business Systems GmbH, ein mittelständisches Unternehmen mit Sitz in Bayreuth und Nürnberg, ist seit 2001 unabhängig von seiner Muttergesellschaft Grundig. Der Komplettlösungsanbieter im Bereich des Diktierens und der Sprachverarbeitung bietet außer analogen und digitalen Diktiersystemen auch softwarebasierte Lösungen an. Dank dieser langjährigen Erfahrung liefert GBS einen wertvollen Beitrag zum SPEAKER-Projekt. In Ihrem Anwendungsfall wird ein softwarebasiertes Diktiersystem entwickelt, welches auch die hohen Datenschutzanforderungen des Gesundheitswesens und öffentlicher Behörden erfüllt.	AP1: Anforderungsanalyse, Architektur- und Schnittstellendefinition AP5: Datenbereitstellung für domänenspezifische Anwendungsfälle AP6: Spracherkennung und Interpretation in geführten Gesprächssituationen AP7: Reporting
IAV (Ingenieurgesellschaft Auto und Verkehr) GmbH	IAV ist mit mehr als 7.600 Mitarbeitern einer der weltweit führenden Engineering-Partner der Automobilindustrie. IAV entwickelt seit 40 Jahren innovative Konzepte und Technologien für zukünftige Fahrzeuge. Im Projekt SPEAKER sollte sowohl ein Sprachassistent für Produktion und Entwicklung als auch ein endgeräteübergreifender Sprachassistent für Endkunden geschaffen werden.	AP1: Erfassen der Anforderungen und definieren der Schnittstellen AP6: Entwickeln eines Sprachassistenten für die Fahrzeugentwicklung sowie eines endgeräteübergreifenden Sprachassistenten AP7: Projektkoordination, Fehleranalyse und Dokumentation
ONSEI GmbH	ONSEI bietet KI-basierte Softwarelösungen für einen digitalen und effizienten Kundendialog. ONSEI's Expertise umfasst Sprachassistenten und Chatbots, Textautomationen sowie die Entwicklung von Datenstrukturen für KI-Anbindungen. Dadurch werden Mitarbeiter entlastet und Service vertieft. Mit Hilfe der SPEAKER-Plattform soll ein domänenspezifischer Sprachassistent für die Hotellerie umgesetzt werden.	AP1: Anforderungsanalyse, Architektur, Benchmarks AP5: Voice User Experience Testing AP6: Umsetzung eines Sprachassistenten in der Hotellerie AP7: Projektplanung und -koordination sowie Dokumentation
Retresco GmbH	Die Retresco GmbH ist ein weltweit führender Anbieter von KI-Lösungen zur automatisierten Verwertung von Sprach- und Textdaten. Sie besitzen langjährige Erfahrung auf den Gebieten NLU und NLG. Entsprechend beteiligt sich Retresco als technischer Umsetzungspartner hinsichtlich der NLG sowie als allgemeiner Berater der Plattform.	AP1: Spezifikation Architektur und Anforderung AP3: Datenhandling und Identity Management AP6: NLG-Komponente für die SPEAKER-Plattform AP7: Projektmanagement, Koordination und Controlling
SAP SE	Die SAP SE wird zwei Anwendungsfälle mit SPEAKER entwickeln. Zum einen wird die SAP Conversational AI Plattform an die SPEAKER-Plattform angebunden, zum anderen wird der Einsatz von SPEAKER-Technologie bei der Live-Transkription während Events erprobt, um eine höhere Barrierefreiheit zu ermöglichen. Zudem erfolgt eine Unterstützung beim Aufbau des SPEAKER-Ökosystems und Qualitätsstandards.	AP1: Anforderungsanalyse, Architektur, Benchmarks AP5: Tools und Daten - Voice User Experience AP6: Evaluation in Anwendungsfällen und Modulentwicklung AP7: Projektmanagement und Öffentlichkeitsarbeit
Scopevisio AG	Die im Jahr 2007 in Bonn gegründete Scopevisio AG beschäftigt in Deutschland und Österreich mehr als 100 Mitarbeiter. Heute zählt	AP1: Anforderungsanalyse, Architektur, Benchmarks

Partner	Rolle	Aufgabe
	len über 3.000 mittelständische Unternehmen, über 500 Steuerberater und mehr als 2.500 Hotels zu ihren Kunden. Ein erster Sprachassistentenprototyp namens Scoper zur Unterstützung der Nutzer ihrer Unternehmenssoftware wurde 2017 auf der CeBIT positiv aufgenommen. Dieser Sprachassistent wird nun unter Beachtung von Datenschutzauflagen mittels der SPEAKER-Plattform umgesetzt und weiterentwickelt.	AP5: Erfassen von domänenspezifischer Sprache und Kuration der Daten AP6: Sprachassistent zur Einbindung in domänenspezifische Unternehmenssoftware AP7: Geschäftsmodell, Öffentlichkeitsarbeit und Reporting
Siemens AG	Die Siemens AG wird die SPEAKER-Plattform für die Steuerung von Maschinen in der industriellen Produktion einsetzen. Insbesondere werden Aspekte der robusten Spracherkennung im industriellen Umfeld und die Integration von Domänenwissen untersucht und in konkreten Anwendungen erprobt.	AP1: Anwendungsszenarien und Anforderungsanalyse AP5: Voice User Experience, Benchmarking AP6: Architektur und Entwicklung Machine App AP7: Reporting
Hostingpartner		
Cloud & Heat Technologies GmbH	Aufgrund des unerwarteten Wegfalls unseres für das Cloud Hosting zuständigen Verbundpartners iNNOVO Cloud GmbH, mussten Teile der Arbeitspakete des Partners sowie die eingeplanten Ressourcen für das Hosting der SPEAKER Plattform anderweitig bereitgestellt werden. Nach Absprache mit dem Projektträger, konnten wir die Cloud & Heat Technologies GmbH als engagierten sowie erfahrenen deutschen Hoster für das Verbundvorhaben gewinnen. Die Arbeitspakete des nun weggefallenen Partners iNNOVO Cloud werden Cloud & Heat im entsprechenden Maße übernehmen. Hinsichtlich der fehlenden beizustellenden Rechenressourcen zum Hosting der Plattform werden diese in Form eines Unterauftrags bei Fraunhofer vorgesehen.	AP1: Erarbeiten des Conceptual Technical Model (CTM) sowie der Logical Technical Model (LTM) und das Physical Technical Model (PTM) der SPEAKER-Plattform AP2: Erarbeiten des Governance Modells der Schnittstelle zwischen Betreibergesellschaft und des Cloud Hosters. AP3: Aufbau/Betrieb der SPEAKER Modul-, Daten- und Web-Plattform sowie des Billing/Identity Managements. Sicherstellen der Skalierbarkeit der Plattform und Erarbeiten eines Konzeptes zur Datensicherheit. AP7: Reporting
Wissenschaftliche Partner		
Deutsche Forschungszentrum für Künstliche Intelligenz	Das Deutsche Forschungszentrum für Künstliche Intelligenz ist auf dem Gebiet der künstlichen Intelligenz eine führende deutsche wirtschaftsnahe Forschungseinrichtung und konnte mit ihrer Beteiligung an META-NET und dem European Language Grid (ELG) Expertise in dem Bereich Multilingualität in Verbindung mit KI aufbauen. Da Multilingualität ein wichtiges Kriterium der Marktakzeptanz darstellt, liefert das DFKI mit dieser Expertise einen wichtigen Beitrag zur SPEAKER-Plattform.	AP1: Anforderungsanalyse, Architekturspezifikation und Kuratierungsprozesse AP3: Aufbau der SPEAKER-Datenplattform (multilinguale Workflows, Schnittstellen zu europäischen Plattformen) AP4: Einbindung von Knowledge Graphs, Question-Answering mono- und multilingual, Entwicklung eines Storytelling Moduls für komplexe Dialogsequenzen AP5: Multilingualität AP7: Reporting
International Data Space	Die International Data Space Association (IDSA) ist eine Vereinigung für den sicheren	AP1: Anforderungsanalyse, Architektur, Benchmarks

Partner	Rolle	Aufgabe
Association (IDSA)	Datenaustausch zwischen Unternehmen. Die IDSA definiert dafür internationale Standards und Governance-Strukturen und stellt technische Schnittstellen bereit. Die Aufgaben von IDSA in SPEAKER liegen in der Beteiligung am Aufbau eines Ökosystems, der Governance-Struktur für SPEAKER, in der technischen Integration an die IDS-Referenzarchitektur sowie in der Bekanntmachung des Projekts durch geeignete Marketingaktivitäten im Umfeld des bestehenden International-Data-Space-Ökosystems.	AP2: Geschäftliche Plattformentwicklung & Community Management AP3: Technische Plattformentwicklung (Schnittstellenbildung zur IDS-Referenzarchitektur) AP7: Reporting
Institut für Phonetik und Sprachverarbeitung der LMU München	Das Institut für Phonetik und Sprachverarbeitung der LMU München bringt Synergien hinsichtlich deutschsprachiger Sprachkorpora und Kuratierungswerkzeuge in das SPEAKER-Projekt ein. Schon im Projekt "Bavarian Archive for Speech Signals" zeigt sich die Kompetenz in der Kuration von großen Sprachkorpora.	AP1: Anforderungsanalyse für APIs / Consulting AP3: Realisierung des Datenkonn timers deutsche Sprachdaten sowie Realisierung des Webservice Phonetische Transkription für verschiedene Sprachen. AP5: Beratung hinsichtlich der Sprachdatenerhebung und Bereitstellung eines Webservices zur automatischen Kuration von Sprachdatensammlungen.
Institut für Nachrichtentechnik der TU Braunschweig	Das Institut für Nachrichtentechnik der TU Braunschweig bringt mit der Beteiligung von Prof. Dr.-Ing. Tim Fingscheidt einen Wissenschaftler mit vieljähriger Erfahrung auf dem Bereich der Sprachsignalverarbeitung in das SPEAKER-Projekt ein. Aus dieser Kooperation soll ein Technologiemodul zur audiovisuellen Spracherkennung erforscht und einer entsprechenden Anwendungsfall im autonomen Kleinbus entwickelt werden.	AP1: Anforderungen und Schnittstellen für die audiovisuelle Spracherkennung AP4: Entwicklung eines Technikmoduls zur multimodalen Spracherkennung AP5: Auswahl multimodalen Datenbanken zum Trainieren von Spracherkennungsmodulen sowie die Entwicklung eines Testframeworks AP6: Umsetzung eines Echtzeitdemonstrators zur audiovisuellen Spracherkennung AP7: Reporting, Beteiligung an Gremien
Data-Science-Lehrstuhl der Universität Paderborn	Der Data-Science-Lehrstuhl der Universität Paderborn hat langjährige Erfahrungen in internationalen und nationalen Projekten im Bereich Question Answering Systeme vorzuweisen. In dem SPEAKER-Projekt wird sein im Eurostars-Projekt QAMEL aufgebautes Offline Question Answering-System weiterentwickelt, um Unternehmen wie bspw. der Deutschen Bahn oder ONSEI in Situationen ohne Internetverbindung einen antwortenden Sprachassistenten bereitstellen zu können.	AP1: Anforderungen und Schnittstellen für Offline QA AP6: Unterstützung der Deutsche Bahn AG bei der Umsetzung ihres Anwendungsfalls Sprachassistent zur Schadensprotokollierung, sowie der ONSEI GmbH bei der Umsetzung ihres Anwendungsfalls Sprachassistent in der Hotellerie. Entwicklung von Modulen für Offline Question Answering. AP7: Öffentlichkeitsarbeit und Reporting

Die Konsortialleitung, mit Standorten Sankt-Augustin, Erlangen und Dresden, stimmten sich hierbei regelmäßig ab. Zum Austausch mit den Verbundpartnern fanden monatliche Treffen aller

Verbundpartner während der Projektlaufzeit statt. Die Zusammenarbeit erfolgte über mehrere verschiedenen kollaborative Plattformen sowie Online-Tools (Confluence, JIRA, Office 365, Git-Lab, OwnCloud), welche von der Konsortialleitung bereitgestellt wurde. Aufgrund der Covid-19 Pandemie mussten Treffen überwiegend in Form von Videokonferenzen stattfinden. Nichtsdestotrotz konnte während der gesamten Projektlaufzeit ein regelmäßiger Austausch sowie eine enge Zusammenarbeit zwischen allen Partnern gewährleistet werden.

Der Markt für Sprachassistenzlösungen wurde zu Beginn des Projektvorhabens von den US-amerikanischen GAFA Unternehmen (Google, Apple, Facebook und Amazon) und zunehmend auch von asiatischen Marktteilnehmern (Baidu, Samsung) dominiert. Es bestand konkreter Bedarf Deutscher Unternehmen nach Sprachassistenzlösungen, welche uneingeschränkte Datensicherheit sowie -Souveränität gewährleisten. Zudem bestand im Business-to-Business und Industriebereich der Bedarf nach Lösungen unabhängig von den existierenden, konsumentenorientierten Sprachtechnologieplattformen.

Die Leistungsfähigkeit der konsumentenorientierten Sprachassistenzlösungen waren insbesondere hinsichtlich deutscher Dialekte, schwieriger akustischer Umgebungen, Sprecherkennung und komplexen wirtschaftlichen Domänen noch nicht ausgereift. Diese Aspekte sind für die Industrie von großer Relevanz.

3 Planung und Ablauf des Vorhabens

Zunächst wurde ein Umsetzungskonzept erarbeitet, welches dem Projektträger am 16.08.2019 präsentiert wurde. Dieses diente zur detaillierten Planung der Umsetzungsphase, in welcher der Aufbau der Sprachassistenzplattform erfolgte. Die Umsetzungsphase wurde anhand feingranularer Arbeitspakete strukturiert.

Das SPEAKER Projekt orientierte sich im Wesentlichen an sechs Meilensteinen (vgl. *Tabelle 2*). Diese wurde kontinuierlich durch die Projektleitung und das Leitungsgremium überprüft. Auch fanden entsprechend Meilensteintreffen statt, bei welchen die Projektergebnisse den Stakeholdern präsentiert wurden.

Tabelle 2: Meilensteine des SPEAKER Projektvorhabens

Meilenstein	M	Beschreibung
M1	6	Workshop Anforderungsanalyse, Festlegung der Referenz-Architektur, Festlegung der Benchmarks und Qualitätskontrolle, Webseite online

Meilen- stein	M	Beschreibung
		• KPI 1.1: Webseite mit Projektinhalten, Social Media und Feedbackkanälen verfügbar • KPI 1.2: Referenzarchitektur finalisiert • KPI 1.3: Agiler Entwicklungsprozess etabliert (Fraunhofer intern) • KPI 1.4: Kommunikations-Infrastruktur für Projektpartner und Community • KPI 1.5: 2 Community Workshops
M2	12	Initialversion, MVP (V1) der SPEAKER-Plattform (Projektinterner Zugang zu den Modulen) • KPI 2.1: MVP der Plattform auf Cloudinfrastruktur verfügbar • KPI 2.2: Notwendige APIs für ausgewählte Anwendungsfälle spezifiziert • KPI 2.3: Initiale Module für Sprachassistentenfunktionalitäten auf der Plattform verfügbar • KPI 2.4: 10% mehr assoziierte Partner zur Ausweitung des Ökosystems • KPI 2.5: Community-Event im industriellen Umfeld
M3	18	Initiale Version der Module und erste Anwendungsfälle • KPI 3.1: Alle für die Anwendungsfälle notwendigen Module auf der Plattform verfügbar • KPI 3.2: Basisversion der Anwendungsszenarien verfügbar • KPI 3.3: Trainingsinfrastruktur und Dateninfrastruktur vorhanden • KPI 3.4: Community-Event im akademischen Umfeld
M4	24	Stabile und erprobte Version (V2) der SPEAKER-Plattform intern verfügbar (Verbundpartner, assoziierte Partner) • KPI 4.1: Version 2 der SPEAKER-Plattform verfügbar (getestete Produktionsumgebung) • KPI 4.2: Billing und Identity Management und KPI-Tracking etabliert • KPI 4.3: Governance-Konzept für Plattform verfügbar • KPI 4.4: 10% mehr assoziierte Partner zur Ausweitung des Ökosystems • KPI 4.5: Community-Event im industriellen Umfeld
M5	30	Öffnung/Launch der Plattform (V3) in die Öffentlichkeit • KPI 5.1: Marketing-Kampagne gestartet • KPI 5.2: Öffentlicher Launch der Plattform (Beta Testing) • KPI 5.3: Prototyp der Anwendungsfälle der Anwendungspartner getestet und stabil • KPI 5.4: Community-Event zum Launch der Plattform • KPI 5.5: Rechtliche Grundlagen für Betreibergesellschaft geschaffen
M6	36	Übergabe an Betreiberorganisation und Betrieb der Plattform • KPI 6.1: Geschäftsmodell (Zahlungsworkflows) ausgearbeitet

Meilen- stein	M	Beschreibung
		• KPI 6.2: 10% mehr assoziierte Partner zum Anstoß des selbsttragenden Öko-systems • KPI 6.3: Übergabe an Betreibergesellschaft • KPI 6.4: Abschlussveranstaltung

4 Anknüpfung an wissenschaftlichen und technischen Stand

4.1 Akustische Vorverarbeitung (Acoustic Frontend, AFE)

Die Vorverarbeitung von Mikrofonsignalen in Sprachassistenten- und Kommunikations-Anwendungen wird oft als Acoustic Frontend (AFE) bezeichnet, alternative Fachbegriffe sind auch Speech Enhancement (SE) oder Voice Quality Enhancement (VQE). In allen Fällen wird darauf abgezielt, die Stimmufnahmen der Nutzer so zu prozessieren, dass diese optimal weiterverarbeitet werden können und möglichst wenig durch Störsignale überlagert sind. Dies ist gerade in Verbindung mit automatischer Spracherkennung von entscheidender Wichtigkeit, da ein klar verständliches Sprachsignal in der Regel auch zu einer besseren Erkennung des gesprochenen Worts beiträgt. Der wissenschaftlich-technische Stand zu Beginn des SPEAKER Projekts lässt sich am besten anhand der Ergebnisse der Deep Noise Suppression Challenge darstellen, welche in etwa zeitgleich mit dem Start unseres Projekts im Rahmen der Konferenz Interspeech 2020 statt fand. Der Wettbewerb wurde von Microsoft organisiert und hatte das Ziel, die Forschung und Entwicklung im Bereich des einkanaligen Speech Enhancement unter Echtzeitbedingungen zu fördern. Sprachverbesserung in Echtzeit mit dem Ziel, die subjektive (wahrnehmbare) Qualität der verbesserten Sprache zu maximieren. Die Bewertung der verschiedenen Verfahren erfolgte auf Basis von online durchgeführten ITU-T P.808 Test. Drei der Systeme konnten sich auf die ersten Ränge behaupten. Das System von Hu et al. [1] verwendet rekurrente und faltungsbasierte neuronale Netze und bezieht Phaseninformationen mit ein. Der Ansatz von Valin et al. [2] ist besonders für Echtzeit-Anwendung mit niedriger Komplexität ausgelegt. Die Arbeit von Strake et al. [3] ermöglicht sowohl Störgeräuschreduktion als auch Enthüllung. Wie in Abschnitt 6.3.4 dargestellt, vereint die im SPEAKER Projekt eingesetzte Lösung viele dieser Eigenschaften.

4.2 Automatische Spracherkennung (Automatic Speech Recognition, ASR)

Automatische Spracherkennung (Automatic Speech Recognition, ASR) wandelt gesprochene Sprache in geschriebene Sprache um. ASR wird im SPEAKER Projekt zur Eingabe von Sprachbefehlen und Echtzeit-Transkription der Nutzereingaben verwendet. Dieses Projekt knüpft an die klassischen Ansätze [4]: Hybridmodelle aus einem neuronalen Netzwerk und konventionellen statistischen Modellen sowie einsprachigen, rein neuronalen Modell (End-to-End-Modellen an). Diese Modelle boten in der Vergangenheit meist keine Anpassungsmöglichkeiten, wenn sie einmal trainiert waren. Außerdem lieferten End-to-End-Modell noch über eine geringere Erkennungsqualität verglichen zu Hybridmodellen. An diesen Stand wurde angeknüpft, um eine anpassbare auf der SPEAKER Plattform einsetzbare Lösung zu entwickeln.

Durch eine umfangreiche Recherche in der Fachliteratur und Online-Quellen wurden potentielle Vergleichsprojekte identifiziert und diese mit der entwickelten ASR von Fraunhofer verglichen. Für die ASR kamen namentlich der Fraunhofer Docker-Container zum Einsatz und die Projekte Whisper (in den Varianten tiny, medium, large) von OpenAI [5], Vosk von Alpha Cephei und Conformer-1 [6] von AssemblyAI. Die Ergebnisse der Evaluation zeigen, dass die Fraunhofer ASR die besten Resultate bei den gesamt getesteten Modellen liefert.

4.3 Keyword / Wakeword Detektor (KWS)

Keyword/Wakeword Detektoren suchen nach bestimmten Schlüsselwörtern innerhalb eines gegebenen Audiodatenstroms. Während sie anfänglich noch häufig als Tool zum Audio Mining verwendet wurden, richtete sich mit dem Aufkommen erster sprachbasierter Assistenzsysteme der Fokus auf die Detektion eines Aktivierungswortes, welches das System in einen aktiven Modus versetzt.

Anfängliche Keyword Detektoren basierten lange Zeit auf Hidden Markov Modellen (HMMs), allerdings verloren sie mit den Entwicklungen im Deep Learning Bereich zunehmend an Bedeutung. Ein umfassender Überblick über den Stand der Technik findet sich in der Veröffentlichung von López-Espejo et al. [7]. Einer der ersten neuronalen Ansätze war dabei [8], bei dem ein Audiodatenblock in ein neuronales Modell gefüttert wird, welches dann eine Wahrscheinlichkeitsverteilung über mehrere Klassen von Schlüsselwörtern ausgibt. Ergänzt wurde die Signalverarbeitungskette um eine entsprechenden Vor- bzw. Nachverarbeitung, die die Erkennungsrate verbessern und einen Dauerbetrieb mit einem beständig zugeführten Audiodatenstrom erst ermöglichen sollte. Dieser Ansatz war und ist tatsächlich bis heute noch eine sehr beliebte Methode. Verbesserungen wurden dabei meist durch das Integrieren neuerer Deep Learning basierter Methoden erreicht, z.B. durch den Einsatz von Convolutional Neural Networks [9] oder Residual Networks [10].

Ein anderer Ansatz sind Systeme basierend auf dem Prinzip der Connectionist Temporal Classification (CTC) [11], [12]. Hierbei erzeugt das Modell ähnlich wie bei der Spracherkennung eine Wahrscheinlichkeitsverteilung über einen Zeichensatz, der aus Wörtern, Phonemen oder Graphemen bestehen kann. Aus diesen Verteilungen kann dann errechnet werden, ob ein bestimmtes Schlüsselwort in dem jeweiligen Audiodatenblock auftritt. Ein Vorteil dieser Ansätze im Gegensatz zu der vorher beschriebenen Methode ist, dass man in der Regel das Schlüsselwort für ein bereits trainiertes Modell beliebig wählen kann. Dieser Umstand hat uns dann auch dazu motiviert, ein System auf Basis des CTCs zu entwickeln. Ferner ließen wir Elemente aus weiteren Methoden [13], [14] einfließen, um die Nachverarbeitung und die Konfigurierung zu verbessern.

4.4 Sprechererkennung (Speaker Recognition, SR)

Zum Start des SPEAKER Projekts war der Stand der Technik bei Sprechererkennungssystemen gerade im Umbruch hin zu textunabhängigen, End-to-End Systemen, die aussagekräftige Informationen direkt aus den Sprachsignalen der Sprecher extrahierten. Diese Informationen wurden üblicherweise in Form eines Speaker Embeddings kodiert. X-Vektor-Architekturen [15] erzeugen Speaker Embeddings, die bei der Sprecherverifizierung verwendet werden. X-Vektoren verwenden neuronale Netze mit Zeitverzögerung (TDNN), um sowohl die spektralen als auch die zeitlichen Merkmale von Sprache zu erfassen. Das Ergebnis ist ein Embedding mit fester Länge für eine Eingabe mit variabler Länge.

Das von Ravanelli et al. entwickelte SincNet [16] ist eine CNN-Architektur, die einen so genannten "Sinc-Layer" einführt, welcher die zufällig initialisierten Faltungskerne der Convolutional Layer durch Sinc-Filter ersetzt, deren trainierbarer Parameter die Grenzfrequenz ist. Die Impulsantworten der beim Training gelernten Filter behalten somit die Sinc-Funktion bei. Der verallgemeinerte End-To-End-Loss [17] wurde entwickelt, um das Training der Sprecherverifikation durch die Konzentration auf schwierige Beispiele aus den Trainingsdaten zu optimieren.

Angesichts des Erfolgs von Architekturen, die auf Attention-Mechanismen basieren, wurden von Bian et. al. [18] Kombinationen aus ResNet-Modellen und Self-Attention eingeführt, um langfristige Abhängigkeiten zwischen Elementen der Eingabesequenz zu berücksichtigen und die Anzahl der Trainingsparameter zusätzlich zu reduzieren. Jung et al. entwickeln RawNet [19], welches Embeddings auf Frame-Ebene unter Verwendung von Residualblöcken mit vortrainierten CNN generiert und diese mit Hilfe des LSTM-Modells zu einem Embedding auf Äußerungsebene aggregiert. Für das Training wird die Loss-Funktion angepasst, um die LDA-Analyse zu berücksichtigen. In einer späteren Arbeit fügten Jung et. al. dem RawNet-Modell skalierende Merkmalskarten hinzu und führten Rawnet2 [20] ein, was zu einer Verbesserung der Leistung von 50% gegenüber RawNet führte.

ECAPA-TDNN [21] von Desplanques et. al. baut auf der X-Vektor-Architektur auf und verbessert sie durch mehrere Änderungen erheblich. Erstens wurden 1-dimensionale Squeeze-and-Excitation Res2Blocks in die x-Vektor-Architekturen eingeführt, um explizit gegenseitige Abhängigkeiten zwischen den Kanälen im Netzwerk zu modellieren und den zeitlichen Kontext der Frame-Layer zu erweitern. Zweitens wurde eine mehrschichtige Merkmalsaggregation eingesetzt. Schließlich wurde das Statistik-Pooling-Modul durch eine kanalabhängige Frame-Attention verbessert. Mit dieser Arbeit wurde eine erhebliche Verbesserung gegenüber den bisherigen X-Vektor-Architekturen erreicht.

4.5 Natural Language Understanding (NLU) / Dialog Management (DM)

Das Natürliche Sprachverstehen (Natural Language Understanding) umfasst Technologien wie Intent-Klassifikation und Slot-Filling mittels Entitätenerkennung und wird ergänzt durch das sogenannte Dialogmanagement zur Kontrolle der Interaktion des Nutzers mit Sprachassistenzsystemen. Die u.a. aus diesen Einzelkomponenten bestehenden Dialogsysteme, werden in zwei Kategorien unterteilt: Zum einen existieren zielgetriebene Systeme, welche in der Industrie Anwendung finden. Zum anderen gibt es nicht-zielorientierte Systeme, wie sie beim Smalltalk verwendet werden. Bisherige industrielle Systeme nutzen oftmals domänenspezifische, hart einprogrammierte Regeln, welche zumeist von Domänenexperten erstellt werden, ein eingeschränktes Vokabular sowie eine begrenzte Anzahl an Aufgaben haben. Ein Transfer auf andere Domänen und Anwendung ist nicht direkt möglich. Die Intent-Klassifikation und das Slot-Filling umfassen die Erfassung und Einordnung der Absicht eines Nutzers, bei dessen Verarbeitung zusätzliche, relevante Informationen erkannt und an ausführende Routinen weitergegeben werden. Neben einfachen Ansätzen basierend auf z.B. Schlüsselworterkennung finden auch Methoden des maschinellen Lernens mit neuronalen Netzen Anwendung in der Intent-Klassifikation und Entitätenerkennung. Allerdings benötigten die, zum Startpunkt des SPEAKER Projekts etablierten Verfahren eine große Anzahl von annotierten Trainingsbeispielen, um Praxis-relevante Erkennungsraten zu erreichen. Dies stellt insbesondere bei der Einführung von völlig neuen Anwendungen in verschiedensten Domänen eine große Herausforderung dar, da das Erstellen von umfangreichen, manuell annotierten Trainingsdatensätzen zeitaufwändig und damit kostenintensiv ist. Daher ist die Entwicklung von leistungsfähigen neuronalen Ansätzen zur Intent-Klassifikation und Entitätenerkennung, die nur mit sehr wenigen Trainingsdaten auskommen, von großer praktischer Bedeutung.

4.6 Machine Reading Comprehension (MRC)

Machine Reading Comprehension beschreibt die Aufgabe, aus einer Sammlung natürlichsprachlicher Texte eine korrekte Antwort auf eine Nutzeranfrage zu generieren. Dabei sollen die Antworten, je nach Anforderung an Nachvollziehbarkeit und Korrektheit, direkt aus Dokumenten extrahiert, oder auf Basis dieser generiert werden. Gerade die Generierung bietet Vorteile in der Einbindung von MRC in den natürlichen Gesprächsfluss eines Gesamtdialogsystems.

Typischerweise ist eine MRC modular aufgebaut. Textdokumente werden in einem ersten Schritt in eine semantische Repräsentation überführt. Das können extrahierte Features wie Part-of-Speech (POS) Tags, Named Entities, aber auch Embeddings sein. Diese Repräsentationen werden indiziert und für die Suche verfügbar gemacht. Auf Nutzeranfragen werden die entsprechenden

Textpassagen über die Suche bereitgestellt. Für die konkrete Beantwortung werden Reader Modelle trainiert. Diese identifizieren den Start- und Endpunkt eines Abschnitts innerhalb der gefundenen Dokumente, der die Antwort auf die Frage enthält. Andere, unüberwachte Verfahren nutzen In-Context Learning um Antworten aus dem Kontext direkt verbalisieren zu lassen. Aktuelle Ansätze beziehen ebenfalls die Gesprächshistorie mit zur Beantwortung der Fragen hinzu.

4.7 Question Answering über Knowledge Graphen (KGQA)

Im Question Answering über Knowledge Graphen (KG) werden natürlichsprachliche Nutzeranfragen anhand von Daten innerhalb eines Knowledge Graphen beantwortet. Als Antwort kann direkt ein Teilgraph aus dem Graphen identifiziert und zurückgeliefert werden. Als weitere Möglichkeit kann zuerst eine Graphquery erzeugt werden, die auf dem KG ausgeführt wird. Vorteile von KGQA im Gegensatz zu MRC ist die Möglichkeit, sehr komplexe Zusammenhänge, wie sie in einem KG dargestellt sind, beantworten zu können. Logische Schlussfolgerungen sind ebenso möglich. Darüber werden Antworten nicht nur extrahiert, sondern können weiterverarbeitet werden. Beispiele sind klassische Datenbankoperationen wie Aggregationen, Sortieren, etc. welche bei der Beantwortung von Fragen ausgeführt werden können.

Als Datenformate und Abfragesprachen haben sich RDF, SPARQL und, alternativ, Cypher etabliert. Öffentlich verfügbare, kuratierte KG lassen sich direkt mit eigenen Daten verbinden, um so noch komplexere Fragen zu beantworten.

4.8 Sprachsynthese (Text-to-Speech, TTS)

Die Sprachsynthese (Text-to-Speech, TTS) ist eine essenzielle Komponente für sprach-basierte Benutzerschnittstellen. Sie wandelt textuelle Informationen in gesprochenes Wort um, welches die Nutzer über Lautsprecher hören können. Obwohl künstliche Stimmen auf Basis von konkatenativer Synthese bereits seit den mehr als 20 Jahren Stand der Technik sind (z.B. in Navigationsgeräten) war zum Start des SPEAKER Projekts klar, dass eine viel höhere Natürlichkeit und Sprachqualität durch Deep Learning-basierte, neuronale TTS Systeme erreicht wird [22]. Zum damaligen Zeitpunkt war es jedoch noch problematisch, in Echtzeit Sprachsignale mit überzeugender Sprachqualität zu generieren. Als einziger Anbieter ermöglichte Google über seine Webservices, hochqualitative, neuronale TTS-Stimmen zu generieren. Ein wichtiger Baustein war dabei die Parallel-Wavenet Architektur [23], welche Inverse Autoregressive Flows verwendet. In der Folge kamen weitere Ansätze wie z.B. LPCNet [24] und WaveGlow [25] hinzu. Die leistungsfähigsten akustischen Modelle waren gerade im Umbruch hin zu nicht-autoregressiven Architekturen [26],

[27] und vielversprechende Ergebnisse hinsichtlich multilingualer Modelle [28] waren bekannt geworden. Für die Entwicklung einer TTS Komponente im SPEAKER Projekt waren also gute Ansätze in der Forschung vorhanden, die Umsetzung in ein leichtgewichtiges, produktionsreifes System war jedoch noch offen.

4.9 Emotionserkennung

Emotionaler Ausdruck in der Sprache ist ein grundlegendes Mittel menschlicher Kommunikation, und kann im Falle von Ironie die semantische Aussage von sogar ins Gegenteil verkehren. Die maschinelle Erkennung von Emotion im textuellen und akustischen Sprachausdruck ist das Kerngeschäft von audEERING. Ein international viel beachteter Fachartikel [29] sowie ein öffentlich zugängliches Modell dokumentiert den Stand der Forschung. Im Rahmen des SPEAKER Projekts wurde die kommerzielle Schnittstelle (devAlce) in die Plattform integriert. Im Rahmen der Förderung wurden darüber hinaus eine Verbesserung der bestehenden Modelle erreicht, durch Hinzunahme neuer Trainingsdaten und der Entwicklung eines Testframeworks zur Auswahl des besten Modells durch Kriterien wie Genauigkeit, Fairness, Robustheit und Ressourcenverbrauchs.

4.10 Natural Language Generation

Natural Language Generation (NLG)

Im Bereich der Künstlichen Intelligenz nimmt NLG eine Schlüsselrolle bei der Überbrückung der Kommunikationslücke zwischen Menschen und Maschine ein. Menschen nutzen eine natürliche Sprache, die durch ihre Vielfältigkeit und Mehrdeutigkeit geprägt ist, während Maschinen für die Verarbeitung strukturierte Daten erfordern. NLG zielt darauf ab, diesen Austausch zu erleichtern, indem es die Erstellung natürlichsprachlicher Texte durch Software mittels Umwandlung strukturierter Daten in lesbare und verständliche Text ermöglicht.

SaaS Plattform textengine.io

Die für das Projekt SPEAKER zum Einsatz kommende Komponente für NLG ist die textengine.io von Retresco. Sie wird in einer Vielzahl unterschiedlichster NLG-Anwendungsszenarien in der Praxis eingesetzt. Beispielsweise nutzt die Schweizer Radiostation Viasuisse sie zur automatischen Generierung von Verkehrsnachrichten oder die Immobilienplattform Immobilienscout24 bietet hierüber ihren Nutzenden die Möglichkeit, Objektbeschreibungen automatisiert auf Basis von Eckdaten zu erstellen.

Dabei handelte es sich zum Zeitpunkt des Projektbeginns um eine ausschließlich regelbasierte Verfahrensweise für die Generierung von Texten. Hierfür müssen auf Grundlage der bekannten Datenstrukturen vorab Vorlagen und Bedingungen definiert sein. Die Vorlagen sind im Wesentlichen vorformulierte Templates, die mithilfe von Daten- und Lexikalisierungsalgorithmen individualisiert und instanziiert werden. Bedingungen sind die Umstände, die erfüllt sein müssen, damit die Vorlage zum Einsatz kommt. NLG kann zusätzlich "Big Data" in die Texterstellung einbeziehen. Abhängig von den Input-Daten und der Komplexität der definierten Bedingungen ist Natural Language Generation somit in der Lage, Sachverhalte darzustellen oder Zahlen detailliert zu interpretieren.

Zu Projektbeginn bestand noch nicht die Möglichkeit NLG unter den Bedingungen eines dialogbasierten Systems einzusetzen, da die Textgenerierung ausschließlich auf die Erstellung für sich allein stehender Beschreibungs- oder Berichtstexte aus Daten ausgerichtet war. Die Anforderungen auf spezifische User Intents aus einem sich aufbauenden, interaktiven Dialog heraus mit NLG zu antworten und diese Antworten mittels fortgeschrittener sprachlicher Methoden (z.B. Referenzketten wie in: iPhone X -> das Handy -> das Gerät -> es) einwandfrei zu erstellen, wurden zu Projektbeginn noch nicht erfüllt.

Unterscheidung zwischen regelbasierter NLG und Machine Learning

Es lassen sich zwei Hauptansätze in der NLG-Technologie unterscheiden: regelbasierte Systeme und solche, die auf maschinellem Lernen (ML) basieren [30], [31]. Regelbasierte Systeme zeichnen sich durch ihre Kontrollierbarkeit und Zuverlässigkeit aus, wohingegen Systeme, die auf maschinellem Lernen basieren, das Potenzial bieten, aus großen Datenmengen zu lernen und sich weiterzuentwickeln. Allerdings sind mit ihnen auch Herausforderungen wie die Gewährleistung der inhaltlichen Vollständigkeit und das Vermeiden von statistischen Fehlinterpretationen (sog. Halluzinationen) verbunden.

4.11 Audio Visuelle Spracherkennung

Audiovisuelle Spracherkennung (AV-ASR) bietet im Vergleich zur rein akustischen Spracherkennung einen wichtigen Mehrwert in automotiven Anwendungen und Smart-Home-Geräten, da sie durch ihre erhöhte Robustheit gegenüber Störgeräuschen eine wichtige Verbesserung darstellt. Dies führt zu einer natürlicheren und präziseren Interaktion zwischen Mensch und Maschine, insbesondere wenn Kameradaten verfügbar sind. Dadurch werden wichtige Anwendungsbereiche wie Mobilität und Industrie 4.0 unterstützt.

Als eine der End-to-End-Methoden wurden Transformer-Modelle nach ihrer Vorstellung im Jahr 2017 [32] auch zum ersten Mal im Jahr 2018 in audiovisuellen Spracherkennungsaufgaben erfolgreich eingesetzt [33]. Transformer-Modelle kann man sich – wie der Name es andeutet – als Transformationen oder Konversionen z.B. von gesprochener Sprache in geschriebenem Text vorstellen. Transformer nutzen einen sog. Attention-Mechanismus, bei dem das Netz lernt, seine „Aufmerksamkeit“ auf bestimmte Zeitpunkte des mit dem Mikrophon erfassten Sprachsignals zu fokussieren. In der rein akustischen Spracherkennung sind Transformer-Modelle bereits weit verbreitet. Das kontinuierliche Literatur-Monitoring hat gezeigt, dass sich die Forschung im Bereich der audiovisuellen Spracherkennung im Projektzeitraum immer mehr auf Transformer-Modelle fokussiert [33], [34], da diese aktuell die besten Resultate auf öffentlichen Datensätzen [35], [36] erzielen – und das teilweise mit deutlichem Abstand zu hybriden Netzen. Auch prominente Modelle wie ChatGPT, Whisper und BERT nutzen alle Transformer-Architekturen. Im Berichtszeitraum haben wir daher Transformer-Modelle auf ihre Anwendbarkeit im SPEAKER-Projekt geprüft.

4.12 Question Answering über Knowledge Graphen

Question Answering over Knowledge Graphs gliedert sich in ein monolinguales Basis-QA-Modul, das auf Text-Suche basiert. Dieses Modul basiert auf dem Open-Source-Tool Haystack, an dem

wir verschiedene Erweiterungen vorgenommen haben, damit es Texte in deutscher und englischer Sprache richtig verarbeiten kann (es wurden neue Modelle angewendet). Das Modul benutzen wir als Basis für eine zusätzliche Erweiterung, in der wir die Wissensgraphen integriert haben. Dazu gibt es auch ein multi- und cross-linguales Basis-QA-Modul, welches auf Text-Suche basiert. Dieses Modul basiert ebenfalls auf dem Open-Source-Tool Haystack. Um cross- und multilinguale Verarbeitung zu ermöglichen, kombinieren wir verschiedene Haystack-Implementierungen (jeweils eine pro Sprache). Auch die Einbindung mehrsprachiger Wissensgraphen (Wikidata) erfolgt über Haystack.

4.13 Storytelling

Storytelling ist ein Modul, das es im Rahmen komplexerer Dialoge mit der SPEAKER-Plattform erlaubt, weitere Informationen zu bereits extrahierten Antworten zu präsentieren, z.B. weitere Beispiele, Gegensätze, Hintergründe, zusätzliche Details. Die Ergebnisse gliedern sich in ein Modul für Semantic Storytelling, das für die QA-Module eingesetzt wurde, um komplexere und flexiblere Dialoge zu ermöglichen. Diese Dialoge bestehen aus einer Benutzeranfrage und einer Systemantwort sowie dem Kontext der Antwort, also dem Text, in dem die konkret angebotene Antwort zu finden ist. Dieser Kontext wurde verwendet, um den Dialog zu bereichern, indem zusätzliche Informationen erhalten wurden. Für diesen Zweck wurden Ergebnisse aus dem laufenden DFKI-Projekt QURATOR zum Themenkomplex Semantic Storytelling genutzt, um semantische Relationen zwischen Textfragmenten zu ermitteln. Wir verwenden das Tool für die Analyse der semantischen Beziehungen zwischen dem Kontext der Antwort und dem Wissen des Systems, welches aus Sammlungen von Dokumenten oder Wissensdatenbanken gewonnen wird. Das monolinguale QA-System wurde mit dem semantischen Relations-Erkennungssystem im Text integriert. Zusätzlich haben wir eine grafische Oberfläche (in Form eines Chats) implementiert, die es uns ermöglicht, nicht nur den Ansatz darzustellen, sondern auch die Ergebnisse visuell auszuwerten.

4.14 Offline Question Answering

Im Bereich Question Answering Systeme auf Wissensgraphen, gibt es sowohl Retrieval basierte Verfahren, welche Antwortentitäten direkt aus dem Graphen extrahieren als auch Semantic Parsing Systeme, welche zunächst eine Query in einer Anfragesprache, wie SPARQL generieren, die anschließend auf einer Datenbank oder Triplestore ausgeführt wird, um Antworten zu generieren. Der Vorteil letztgenannter Semantic Parsing Systeme ist, dass durch die Query eine gewisse Erklärbarkeit der Antworten vorhanden ist. Aus diesem Grund haben wir uns im Rahmen des

Projekts vorwiegend mit Semantic Parsing Systemen beschäftigt. In der Praxis werden häufig zunächst relevante Entitäten aus der Frage identifiziert, auf deren Basis dann die Query erzeugt wird.

Für den Bereich Question Answering auf Wissensgraphen wird zunehmend auf Transformer-basierte Sprachmodelle [32] gesetzt. So können vortrainierte Modelle wie T5 zur Generierung von Queries verwendet werden. Zusätzlich werden oft Query Templates verwendet, was die Generierung von Queries enorm vereinfacht. Für den offline Einsatz stehen kleinere Varianten der Sprachmodelle wie zum Beispiel T5-small [37] und BERT-small [38], [39] zur Verfügung, welche die Verarbeitung kurzer Texte wie z.B. Fragen auch mit geringen Rechenressourcen ermöglichen. Die in der Forschung verwendeten Benchmarks basieren häufig auf großen Wissensgraphen wie Freebase, Wikidata oder DBpedia [40] [41]. Für die schnelle Beantwortung von Queries werden deshalb schnelle und präzise Triplestores benötigt. Im Bereich offline Question Answering wurde seitens der Universität Paderborn zwischen 2015 und 2018 bereits das Projekt QAMEL durchgeführt, welches Queries mit einer Relation beantworten konnte. Mit den Entwicklungen im Bereich Natural Language Processing und der effizienteren Hardware konnten im Projekt SPEAKER die Generierung von komplexeren Queries erreicht werden.

4.15 Cloud-Skalierungsfähigkeiten mit zusätzlichen Metriken

Kubernetes bietet verschiedene Mechanismen zur Skalierung von Workloads, um eine effiziente Ressourcennutzung und eine optimale Leistung zu gewährleisten. In diesem Abschnitt werden die verschiedenen Kubernetes-Skalierungsfähigkeiten sowie die Open-Source-Software KRAKE vorgestellt, die unter anderem auf Umgebungsdaten und Datenschutzerfordernungen basierende Skalierung ermöglicht. Zudem wurde ein erweitertes Skalierungskonzept erarbeitet, welches die Möglichkeiten mit zugrundeliegender Open-Source-Software evaluiert.

Kubernetes-Skalierungsfähigkeiten:

Horizontal Pod Autoscaler (HPA): Das HPA-System von Kubernetes ermöglicht die automatische Skalierung von Pods basierend auf CPU-Auslastung oder anderen benutzerdefinierten Metriken. Wenn die definierten Schwellenwerte überschritten werden, werden zusätzliche Pods erstellt, um die Last zu bewältigen. Bei geringerer Auslastung werden Pods entfernt, um Ressourcen zu sparen.

Vertical Pod Autoscaler (VPA): Im Gegensatz zum HPA skaliert der VPA die Ressourcen (CPU und RAM) der Pods vertikal, indem er ihre Ressourcenanforderungen basierend auf der tatsächlichen

Nutzung anpasst. Dadurch wird eine optimale Ressourcennutzung erreicht und Engpässe vermieden.

Cluster Autoscaler (CAS): Der CAS von Kubernetes ermöglicht die automatische Skalierung des gesamten Clusters, indem zusätzliche Knoten hinzugefügt oder entfernt werden, um die aktuelle Arbeitslast zu bewältigen. Dies gewährleistet eine optimale Nutzung der verfügbaren Ressourcen und verhindert Überlastungen des Clusters.

KRAKE-Software: KRAKE ['kra:kə] [42] ist eine Open-Source-Software, die speziell für Kubernetes entwickelt wurde und erweiterte Skalierungsmöglichkeiten bietet, die über traditionelle Metriken hinausgehen. Sie ist eine Orchestrator-Engine für containerisierte und virtualisierte Workloads auf verteilten und heterogenen Cloud-Plattformen. Zur Laufzeit des Projekts wurde unter anderem die Software angepasst und erweitert. KRAKE ermöglicht die Skalierung und Verwaltung von Kubernetes-Workloads basierend auf verschiedenen Metriken. Zum Beispiel basierend auf Umgebungsdaten wie der Verfügbarkeit von erneuerbaren Energien wie Wind, Wasser und Sonnenenergie.

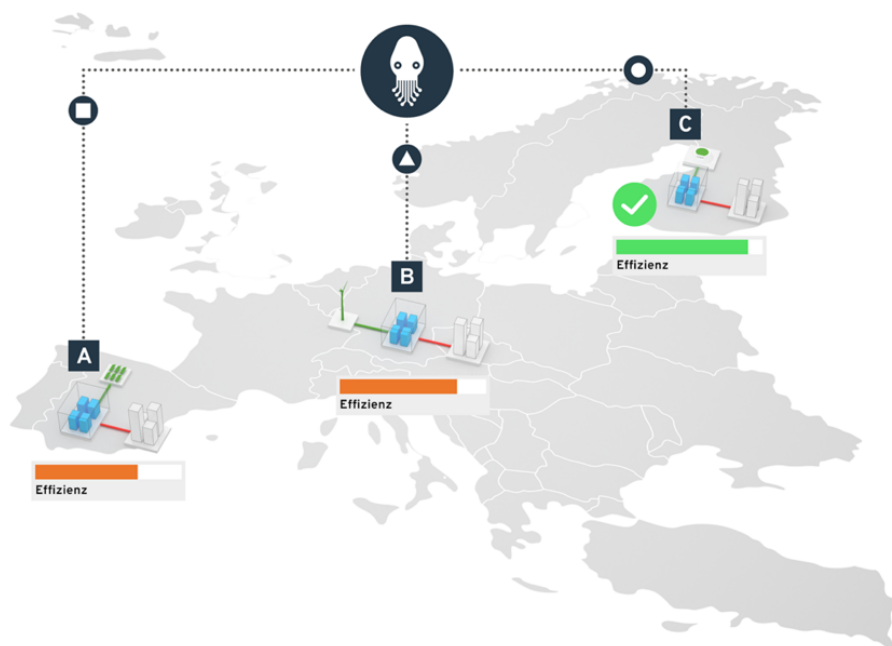


Abb. 2: KRAKE Beispielgrafik für verteilten Workload basierend auf Energieeffizienz-bezogene Kriterien

Darüber hinaus kann KRAKE Workloads basierend auf Datenschutzanforderungen verschiedener Länder planen und skalieren. Dies ermöglicht es Organisationen, ihre Workloads entsprechend den lokalen Datenschutzgesetzen zu verwalten und zu optimieren, indem sie beispielsweise Daten in Rechenzentren speichern, die den rechtlichen Anforderungen entsprechen.

Die verschiedenen Skalierungsfähigkeiten von Kubernetes, einschließlich HPA, VPA und CAS, bieten vielfältige Möglichkeiten zur Anpassung der Ressourcennutzung an die aktuellen Anforderungen von Anwendungen und Workloads. Die KRAKE-Software erweitert diese Möglichkeiten, indem sie auf Umgebungsdaten wie erneuerbare Energien und Datenschutzanforderungen basiert, um eine noch präzisere und nachhaltigere Skalierung zu ermöglichen.

5 Zusammenarbeit mit anderen Stellen

Bei der Umsetzung der SPEAKER Plattform waren eine Reihe an Partnern beteiligt bestehend aus den Verbundpartnern Tabelle 1, welche aus Anwendungspartnern, Hosting-Partnern und Technologiepartnern zusammengesetzt waren. Neben den geförderten Verbundpartnern wurde das Projekt begleitet durch eine Vielzahl assoziierter Partner, welche in Tabelle 7 gelistet sind.

6 Ergebnisse des SPEAKER Projekts

Die Verwendung der Zuwendung für die Erzielung der Projektergebnisse gehen aus den insgesamt sieben Arbeitspaketen (vgl. Anhang B) hervor. In diesem Abschnitt werden zunächst die wesentlichen Ziele der einzelnen Arbeitspakete genannt, anschließend werden die erzielten Ergebnisse den vorgegebenen Zielen gegenübergestellt.

6.1 Anforderungsanalyse, Architektur, Benchmarks

Ziel des **ersten Arbeitspaketes** war es, anhand von strukturierten Workshops eine detaillierte Anforderungsanalyse mit allen Verbundpartnern und interessierten assoziierten Partnern zu erarbeiten. Hieraus sollte eine feingranulare Architekturspezifikation abgeleitet werden, welche den Entwicklungsprozess steuert und mittels KPIs nachvollziehbar macht. Daneben sollte ein Kuratierungsprozess abgeleitet werden, welcher Benchmarks (Daten, Experimenttypen, Metriken), Tests sowie eine Umsetzungsstrategie von Standards und (KI-)Zertifizierungen beinhaltet.

Die Ziele des ersten Arbeitspaketes wurden wie folgt umgesetzt. Zunächst hat das Fraunhofer einen detaillierten Fragebogen zur Anforderungsanalyse erarbeitet. Mittels dieses Fragebogens wurden wichtige Eigenschaften der Pilotanwendungen erfasst. Die anvisierte Anwendungen wurden von den Partnern ausdefiniert und entsprechende Antworten auf die Fragen erarbeitet. Darauf basierend wurden von Fraunhofer bilaterale Anforderungsworkshops mit den Anwendungspartnern organisiert. Aus den Ergebnissen dieser Workshops konnten die Anforderungen an die Architekturspezifikation abgeleitet werden. Es wurde ein initialer Entwurf einer Architektur erarbeitet und im Konsortium zur Diskussion bereitgestellt. Dokumentiert wurden die Ergebnisse im gemeinsam mit den Partnern genutzten Wiki-System Confluence, welches vom Fraunhofer bereitgestellt wurde.

Auf dieser Basis begann anschließend die systematische Sammlung vorhandener und frei verfügbarer Datensätze zum Benchmarking der Komponenten entlang der Pipeline (ASR, VQE, SR, TTS, ER, IC, QA). Aufgrund der agilen Natur des Projektes wurden die Anforderungen bezüglich einzelner Sprachassistentenkomponenten stetig weiterverfolgt, weiter ausdefiniert und angepasst.

6.2 Geschäftliche Plattformentwicklung & Community Management

Das Ziel des **zweiten Arbeitspaketes** war es SPEAKER zu einer erfolgreichen und weit ausstrahlenden KI-Ökosystems werden zu lassen. Hierzu sollten von Beginn an mit den Partnern aus Marketing-, Öffentlichkeits- und Netzwerkaktivitäten zusammengearbeitet werden. Zudem sollten

industrielle Messen, wissenschaftliche Workshops, Vorträge und Tutorials ausgerichtet werden. Zur Nachsteuerung der Plattform, sollte eine Markt- und Wettbewerbsbeobachtungsstrategie ausgearbeitet werden. Zusätzlich sollten in diesem Arbeitspaket Governance-Modelle und –Strukturen etabliert werden. Diese sollten dazu dienen notwendige Schritte für die Konzeption und Etablierung der SPEAKER-Betreibergesellschaft abzuleiten und umzusetzen.

Die Ziele des zweiten Arbeitspaketes wurden wie folgt erlangt. Es wurden Governance-Modelle und –Strukturen erarbeitet. Der Konsortialvertrag wurde zwischen allen Verbundpartnern ausverhandelt und entsprechend gezeichnet. Darüber hinaus wurden Muster eines Auftragsdatenverarbeitungsvertrags beigefügt, um zukünftig den Datenaustausch und die Nutzung der Plattform DSGVO gerecht gestalten zu können. Neben dem Konsortialvertrag wurde eine Geheimhaltungsvereinbarung mit den Konsortialpartnern ausverhandelt und den assoziierten Partnern zur Unterschrift ausgehändigt. Diese Geheimhaltungsvereinbarung wurde ergänzt durch einen Evaluationsvertrag, welcher es unseren assoziierten Partnern ermöglichte, vor dem öffentlichen Release der Plattform verschiedene Technologiekomponenten zu evaluieren.

Projektbegleitend wurde eine kontinuierliche Wettbewerbsbeobachtung und Marktanalyse durchgeführt und entsprechend die Entwicklungen von Mitbewerbern verfolgt. Hierdurch wurden Trends frühzeitig am Markt erkannt und aufstrebende Mitbewerber identifiziert. Durch Vergleiche mit den identifizierten Marktteilnehmern wurde die Wettbewerbsfähigkeit der im SPEAKER Projekt angestrebten Lösung beurteilt.

Es wurde eine Reihe an projektspezifischen Marketingmaßnahmen durchgeführt. Um über den aktuellen Projektstand zu informieren, wurde seit Projektbeginn ein Newsletter an assoziierte und Verbundpartner herausgegeben. Es wurde ein neuer Webauftritt für SPEAKER erarbeitet, welcher unter www.speaker.fraunhofer.de erreichbar ist, dafür wurde die eigständige Internet-Domain für das Projekt reserviert: www.speaker-projekt.de. Auf der Projekthomepage sind auch Anwendungsfälle der Partner zu finden.

Ein neues Key Visual ist entstanden, welches aufgrund des Seitenverhältnisses besser auf moderne Anwendungen angepasst ist. Darüber hinaus ist dieses an die Corporate Identity (CI) Vorgaben unserer Partner anpassbar. Dies ermöglicht es, dass sich das Logo in die jeweiligen CIs der Unternehmen einfügt und repräsentiert die große Diversität des Konsortiums.



Abb. 3: SPEAKER Key-Visual

Durch eine Social Media Kampagne wurde Aufmerksamkeit für das Projekt geschaffen. Über die Twitter Profile der Fraunhofer Institute IAIS und IIS wurden abwechselnd Twitter Cards gepostet, welche über das SPEAKER Projekt informierten. Insgesamt wurden 19 Twitter Cards veröffentlicht, welche zusammengenommen über 50000 Impressionen und nahe 1400 Interaktionen erzielen konnten. Dabei fanden folgende Hashtags Verwendung: #SPEAKERProjekt, #ConversationalAI, #KIWettbewerb. Die Twitter Cards wurden ebenfalls auf der Plattform LinkedIn veröffentlicht, wo diese nahe 40000 Impressionen und über 1300 Interaktionen erzielten. Ausschnitte aus den Social-Media-Aktivitäten zum SPEAKER Projekt sind in Anhang C beigefügt.

Eine Reihe an industrielle Messen sowie wissenschaftliche Workshops, Vorträge und Tutorials, welche vorgesehen waren, mussten aufgrund der besonderen Umstände der COVID-19 Pandemie Remote stattfinden oder ausfallen. Tabelle 3 und Tabelle 4 geben einen Überblick über Veranstaltungen, welche genutzt werden konnten, um auf das SPEAKER Projekt aufmerksam zu machen:

Tabelle 3: Veranstaltungen zur Bekanntmachung der SPEAKER Plattform

Datum	Veranstalter	Titel	Teilnehmer
23.09.2020	ECO Akademie Servicemeister	Webinar: Akzeptanz von KI – Ablehnung oder Vertrauen?	Fraunhofer
07.12.2020	SAP	InspirationSessions Conversational AI and Speech Recognition	Fraunhofer
22.09.2020	Maschinenbau Lehrstuhl, FAU Erlangen	Fachseminar „Smart Living - intelligent, vernetzt, energieeffizient“	Fraunhofer
09.12.2020	Center for Digital Technology and Management in München	NLP Meetup Series: In Search of Best Practices for NLP Projects	Fraunhofer

Tabelle 4: Stattgefundene Workshops und Vorträge

Datum	Veranstaltung	Teilnehmer
30.07.2020	Voice UX Workshop	Verbundpartner, assoziierte Partner
13.11.2020	Data Annotation Workshop	Verbundpartner, assoziierte Partner
26.11.2020	Plattform Workshop	Verbundpartner
09.12.2020	Modell-Workshop Spracherkennung	Verbundpartner
23.02.2021	Wikispeech-Workshop	Verbundpartner
04.03.2021	Workshop Dialogmanager, Dialogeditor und NLU	Verbundpartner
16.03.2021	Multimodality Workshop	Verbundpartner, assoziierte Partner
18.03.2021	Workshop Text-to-Speech	Verbundpartner
20.05.2021	Monthly	Verbundpartner
24.06.2021	Data Annotation Workshop	
24.02.2022	Question Answering over Knowledge Graphs	Verbundpartner, assoziierte Partner

6.3 Technische Modul- sowie Plattformentwicklung

6.3.1 Entwicklung der SPEAKER Plattform

Das Ziel dieses Arbeitspaketes ist es eine skalierbare Plattform zu entwickeln, welche es ermöglicht die Services, welche im Rahmen des SPEAKER Projektes entwickelt werden, zur Evaluation and Dritte bereitzustellen. Die Ausgestaltung dieser Plattform bildet ferner auch die Grundlage für einen möglichen Produktivbetrieb, welcher die zukünftige Verwertung der Technologiemodule vereinfacht.

Hierbei stehen Modularität, Sicherheit sowie die Möglichkeit der kontinuierlichen Bereitstellung verbesserter Technologiemodule im Vordergrund.

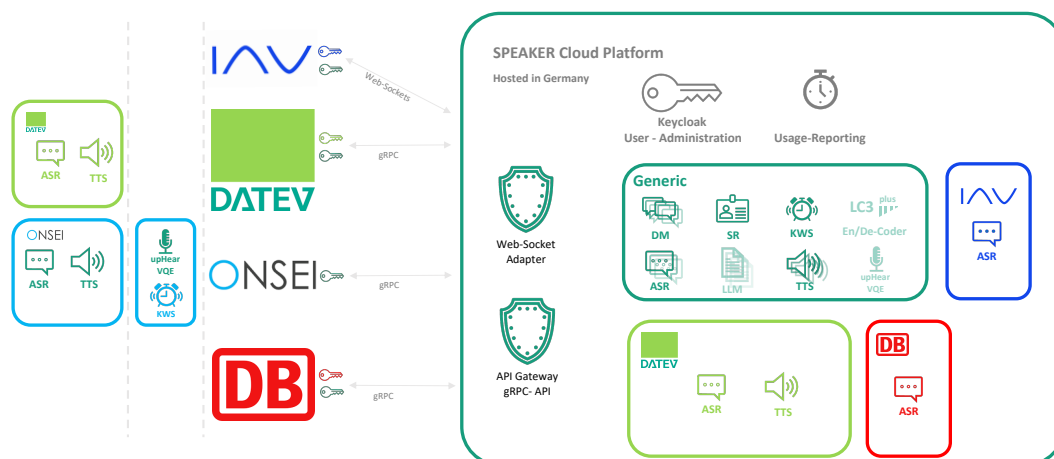


Abb. 4: Übersicht über die SPEAKER Plattform mit ihren unterschiedlichen Komponenten

SPEAKER SDK: Um den Partnern des SPEAKER Projekts Zugriff auf die Technologiemodule der Plattform zu gewährleisten, wurden die Schnittstellenbeschreibungen entsprechend dokumentiert und Beispielimplementierungen bereitgestellt. Die Schnittstellen selbst, sowie deren Dokumentation wurden während dem Projekt kontinuierlich weiterentwickelt, es wurde jedoch stets auf eine Abwärtskompatibilität zu vorherigen Versionen geachtet.

Die Bereitstellung erfolgte versioniert über das **SPEAKER SDK**, welches über das von Fraunhofer gehostet GitLab zur Verfügung gestellt wurde. Hier hatten alle Partner Zugriff und konnten auch die Möglichkeit wahrnehmen beizutragen oder Probleme / Anregungen zu melden.

Als Schnittstellenprotokoll wurde auf das moderne und effiziente gRPC Protokoll gesetzt. Da die Unterstützung der Programmiersprache JavaScript / TypeScript bei diesem nur eingeschränkt gewährleistet ist, wurde neben dem gRPC Protokoll auch die Schnittstellenbeschreibung für das Websocket Protokoll bereit gestellt. Daher konnten nun auch WebApps vollumfänglich mit Streaming-fähigen Schnittstellen umgesetzt werden.

Der Beispielcode, um Anwendungen umzusetzen wurde in verschiedenen Programmiersprachen bereitgestellt. Als primäre Programmiersprache wurde Python gewählt, da viele der Konsortialpartner ihre Beispielanwendungen in Python umgesetzt haben. Neben Python wurden Beispielschnittstellen in JavaScript bereitgestellt, der gängigsten Programmiersprache für Webanwendungen.

Um native Anwendungen auf den Betriebssystemen Android und iOS für Mobiltelefone umzusetzen wurden auch hier Beispielanwendungen in Kotlin für Android, sowie Swift für iOS implementiert. Neben den Beispielclients haben wir auch fertige Anwendungen für Smartphones erstellt um diese entsprechend zur Dissemination auf Messen oder Events präsentieren zu können.

Service Desk: Um einen effizienten Feedbackprozess bzgl. Fehler in Programmteilen oder Web Frontends zu ermöglichen, haben wir den Partnern Zugriff auf ein Ticketing System ermöglicht, welches direkt in den Agilen Management Prozess eingebunden werden konnte. Um den Prozess weiter zu vereinfachen für die Partner wurden hierbei Links zum Ticketing System in den jeweiligen Web Frontends platziert.

KeyCloak: Zur Nutzerverwaltung verwenden wir KeyCloak, welches sich als Lösung im Cloud Umfeld etabliert hat. Die Implementierung sieht vor, dass KeyCloak, basierend auf dem Nutzer Account JWTs (Json Web Tokens) erzeugt, welche signiert sind. Das API-Gateway prüft entsprechend die Signatur, wodurch sichergestellt ist, dass die jeweiligen JWTs valide sind und von unserer Keycloak Instanz ausgestellt wurden. Mit Hilfe der JWTs kann man dann eine sehr feingranulare Nutzerverwaltung mit beschränktem Zugriff realisieren.

Release Prozess: Um einerseits die kontinuierliche Weiterentwicklung, sowie Auslieferung zu ermöglichen, andererseits aber stabil lauffähige Services für die Partner bereitzustellen wurde ein Test und Release Prozess mit verschiedenen Umgebungen etabliert. Der Zugriff auf die unterschiedlichen Umgebungen wird durch das User Management Keycloak geregelt. So haben Partner nur Zugriff auf die Umgebungen Production und Collab. Die Beschreibung der jeweiligen Umgebungen ist in Abb. 6 zu finden.

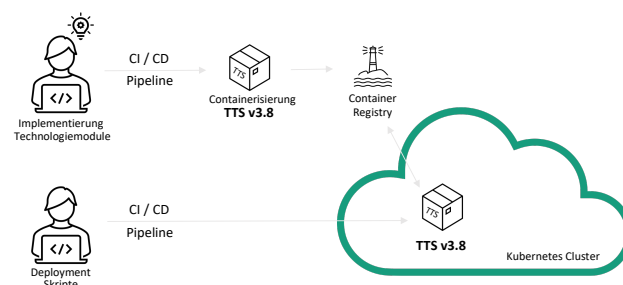


Abb. 5: CI/CD Pipelines zum Erstellen der Docker Container und dem Deployment im Kubernetes Cluster

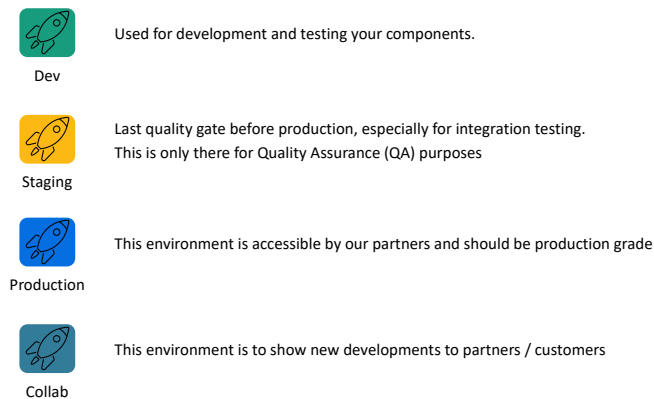


Abb. 6: Unterschiedliche Umgebungen um kontinuierliches Entwickeln sowie Deployment sicherzustellen.

Usage Reporting Server: Während der Anforderungsanalyse der USE-Cases der Partner wurde unter anderem auch die Relevanz verschiedener Abrechnungsmodalitäten identifiziert. Um eine, wie auch bei den Mitbewerbern übliche nutzungsbasierte Abrechnung zu ermöglichen wurde hier prototypisch ein Usage-Reporting-Server implementiert. Hierzu ist das Zusammenspiel verschiedener Komponenten notwendig. In der Architektur wurde berücksichtigt, dass eine Billing-ID zu den jeweiligen Technologiemodulen in die Metadaten der Requests injiziert wird. Zusammen mit dieser Billing-ID können die individuellen Technologiemodule den jeweiligen Verbrauch zum Usage-Reporting-Server übermitteln und somit protokollieren. Über eine dokumentierte API kann dann wiederum der kumulierte Verbrauch, sowie eine Art Einzelnachweis vom Usage-Reporting-Server abgerufen werden.

Kubernetes: Das gesamte SPEAKER Cluster basiert auf Kubernetes, was aktuell der DeFacto std. für die Orchestrierung von Container Workloads im Cloud Native Umfeld ist. Kubernetes übernimmt damit verschiedene Aufgaben. Unter Anderem realisiert es die Skalierung der Workloads, dies kann statisch, anhand der Vorgabe der Anzahl an Replikas erfolgen oder dynamisch mit Hilfe des Horizontal Pod Autoscalers (HPA). So würde der HPA im einfachsten Fall basierend auf der Ressourcennutzung der jeweiligen Container diese replizieren. Die eingehenden Anfragen werden dann über die jeweiligen Services aufgeteilt auf die replizierten Container.

Neben der Skalierung übernimmt Kubernetes auch die Überwachung der Zustände der Container. Über Health- und Readiness Probes werden diese entsprechend abgefragt; Falls ein Container nicht in einem vorgegebenen Zeitintervall antwortet, wird dieser neugestartet.

Ferner übernimmt Kubernetes auch die Aufteilung der Workloads über die verschiedenen Server des Clusters und würde diese bei Bedarf auch umverteilen.

Die Definition des Clusterzustandes mit Deployments, Namespaces, Network Policies, Pod Security Admission Policies etc. wird mit Hilfe von Terraform realisiert. So verwenden wir hier auch sogenannte ConfigMaps um Komponenten wie z.B. Grafana aus dem Observability Stack entsprechend unseren Bedürfnissen zu konfigurieren. Dies hat den Vorteil, dass wir diese Konfigurationsdateien in unserem eigenen GitLab revisionieren können und diese mit Hilfe von Pipelines entsprechend deployed werden. Somit werden Roll-Backs vereinfacht, sowie die Wiederherstellung des Clusters.

Observability Stack: Zur Überwachung des Clusters sowie des Gesundheitszustandes der einzelnen Services wurden verschiedene Toolings entwickelt. Hierbei setzen wir auf eine Kombination aus Grafana, Promtail, Prometheus, OpenTelemetry, Loki sowie Tempo.

Mit diesem Technology Stack ist es möglich die Logs aller Container an einer zentralen Stelle zu durchsuchen, auch von Containern, welche zum aktuellen Zeitpunkt nicht mehr deployed sind. Dies ist gerade im Umfeld von Kubernetes wichtig, da fehlerhafte Container automatisch neugestartet werden und somit die Logs verloren gehen würden, welche jedoch für die Fehlersuche sowie Behebung essentiell sind.

Speicherverwaltung: Die jeweiligen Speicher der einzelnen Nodes wurden mit Hilfe von Longhorn verwaltet. Longhorn ist eine Speicherlösung, welche den Speicher Redundant auf alle zur Verfügung stehenden Server aufteilt. Dies hat den Vorteil, dass bei Ausfall eines Servers die persistierten Daten direkt auf einem anderen zur Verfügung stehen.

Verschlüsselte Backups: Verschiedene unserer Services benötigen Datenbanken, so zum Beispiel zum Speichern von Dialog Graphen des Dialog Editors, oder der Nutzerdatenbank von Keycloak. In regelmäßigen Abständen wird von den Inhalten dieser Datenbanken ein Backup erstellt, welches verschlüsselt ist und in einem S3 Storage abgelegt. Das regelmäßige Ausführen dieser Backups wurde über einen CronJob im Cluster realisiert.

Sicherheitsbericht Fraunhofer SIT: Während des Projektes fand ein Security Audit Seitens des Fraunhofers SIT statt um offensichtliche Fehler und Sicherheitslücken in der Architektur zu identifizieren. Dieser konnte keine Grundlegenden Probleme identifizieren.

6.3.2 Entwicklung der SPEAKER Web Frontends

SPEAKER Stage: Zur komfortablen Evaluierung und zu Demonstrationszwecken wurde für die verschiedenen Technologiemodule ein Web Frontend entwickelt. Hier fand sich der Name SPEAKER Stage, da wir auf diesem sich die Technologiemodule vorführen können. Die Landing Page der SPEAKER Stage ist dargestellt in Abb. 7. Diese Landingpage wird dynamisch erstellt und listet nur Technologiemodule zu deren Nutzung der jeweilige Partner autorisiert ist. Die Evaluations Web Frontends der einzelnen Technologiemodule sind hier als Webapps integriert. Auf die einzelnen Webapps wird im Folgenden kurz eingegangen.

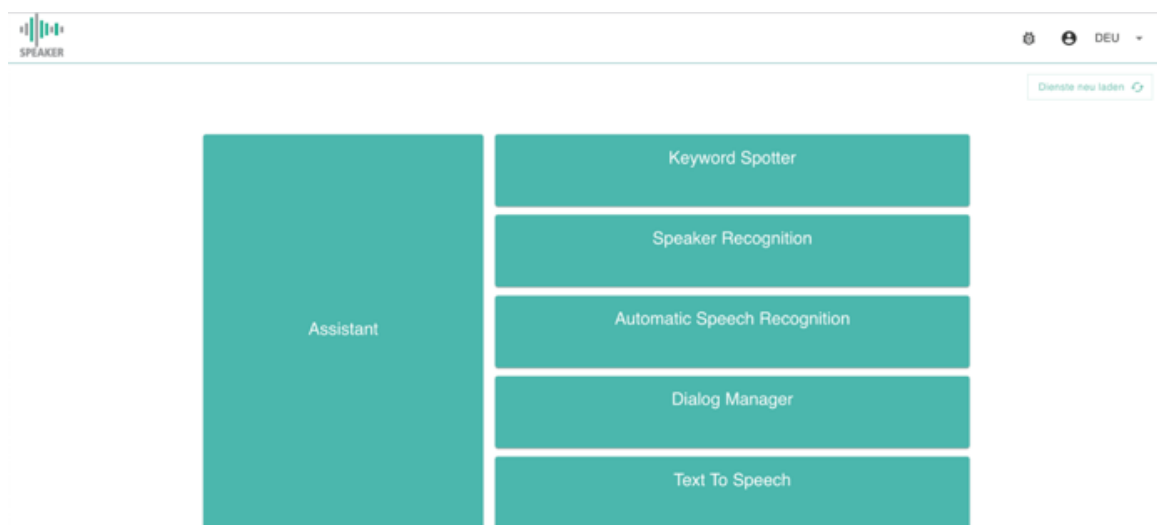


Abb. 7: Screenshot der Landing Page des Evaluations Web Frontends der SPEAKER Services: SPEAKER Stage

Sprachsynthese – Webapp: In Abb. 8 ist das Web Frontend zur Interaktion mit der Sprachsynthese abgebildet. Wie man aus den Drop Down Menüs sieht, stehen verschiedene Services zur Auswahl. Die Liste an Services wird dynamisch, abhängig von den Zugriffsrechten erstellt; So gibt es spezielle, für einzelne Partner nachtrainierte Sprachsynthesemodule auf die nur der jeweilige Partner Zugriff hat. In der Auswahl Umgebung kann man die Umgebung wählen, auf welche im Abschnitt *Release Prozess* näher eingegangen wurde. Auch steht hier die gewünschte Landessprache, sowie die Stimme zur Auswahl. Ferner ist die Sprechgeschwindigkeit einstellbar.

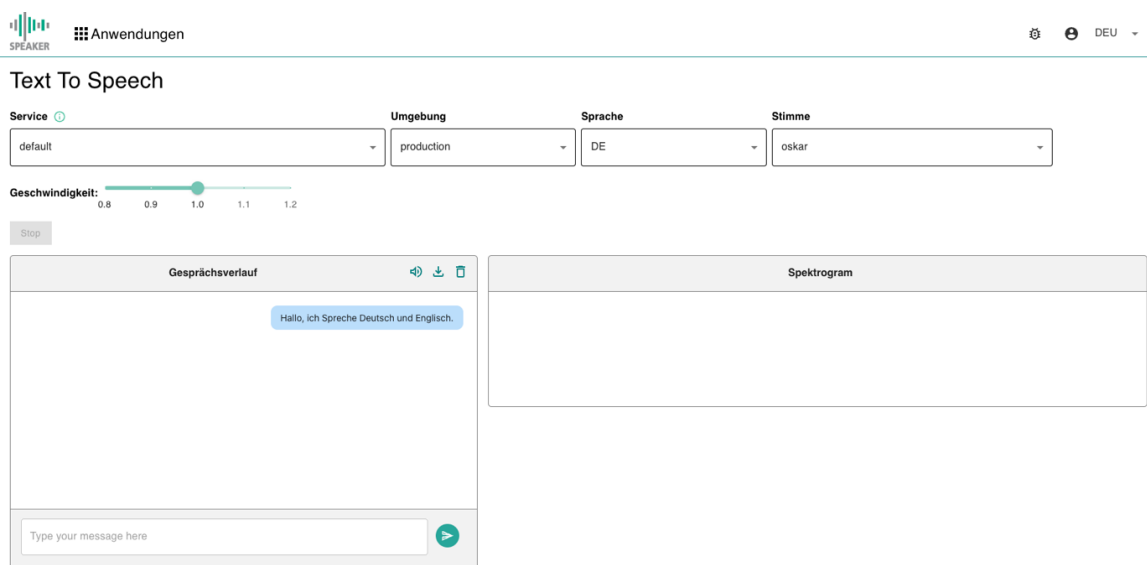


Abb. 8: SPEAKER Stage – Sprachsynthese

Automatische Spracherkennung – Webapp: Analog zur Sprachsynthese bietet das Web Frontend für die automatische Spracherkennung ähnliche Auswahlmöglichkeiten. Wie in Abb. 9 dargestellt kann auch hier zwischen verschiedenen Services und Umgebungen gewählt werden. Diese Auswahl ist analog zum Web Frontend der Sprachsynthese abhängig von den Zugriffsrechten. Da es auch hier speziell nachtrainierte Technologiemodule für ausgewählte Partner gibt, können

diese hier exklusiv dem jeweiligen Partner bereitgestellt werden. Auch die ASR steht in unterschiedlichen Landessprachen zur Verfügung, zwischen denen im Web Frontend umgeschaltet werden kann.

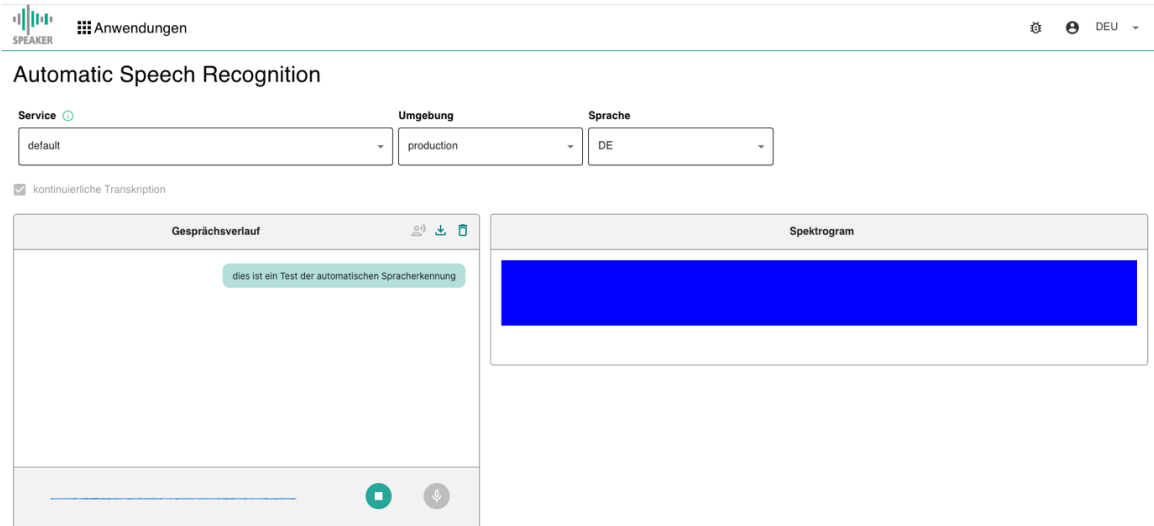


Abb. 9: SPEAKER Stage – Automatische Spracherkennung

Dialog Manager – Webapp: Dargestellt in Abb. 10 ist das Web Frontend zur Evaluierung des Dialog Managers. Dieses basiert zunächst auf einem Fenster ‚Gesprächsverlauf‘, welches eine ähnliche Interaktion wie ein Chatbot zulässt. In diesem Fall wurde der Service ‚car-function-commands-v2‘ gewählt, welches der Fraunhofer-interne Beispiel Use-Case ist, welcher zur Integration der unterschiedlichen Module und zum Durchführen von Tests dient. Bei einer entsprechenden Eingabe, hier der Wunsch das ‚Abblendlicht‘ einzuschalten, wird das vom DialogManager gesendete Event ‚set low-beam 1‘ vom Web Frontend interpretiert und entsprechend optisch dargestellt.

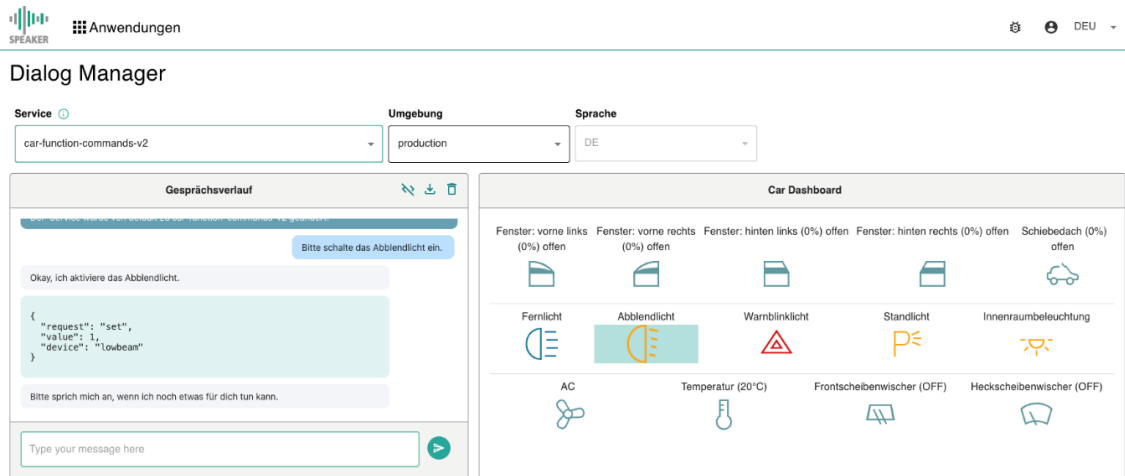


Abb. 10: SPEAKER Stage – Dialogmanager

Sprechererkennung – Webapp: Die Sprechererkennung dient dazu den Sprecher einer Sprachaufnahme zu klassifizieren. Dieser wird klassifiziert als einer der Sprecher aus einer vorgegebenen Sprecherdatenbank oder als unbekannter Sprecher. Um Sprecher in die Datenbank aufzunehmen werden müssen diese registriert werden. Dieser Prozess kann über das Web Frontend entweder mit Hilfe von Audiodateien erfolgen oder man kann direkt im Web Frontend Audio aufnehmen. Alternativ kann man eine Sprecherdatenbank auch in Form einer JSON Datei hochladen, wie in Abb. 11 dargestellt.

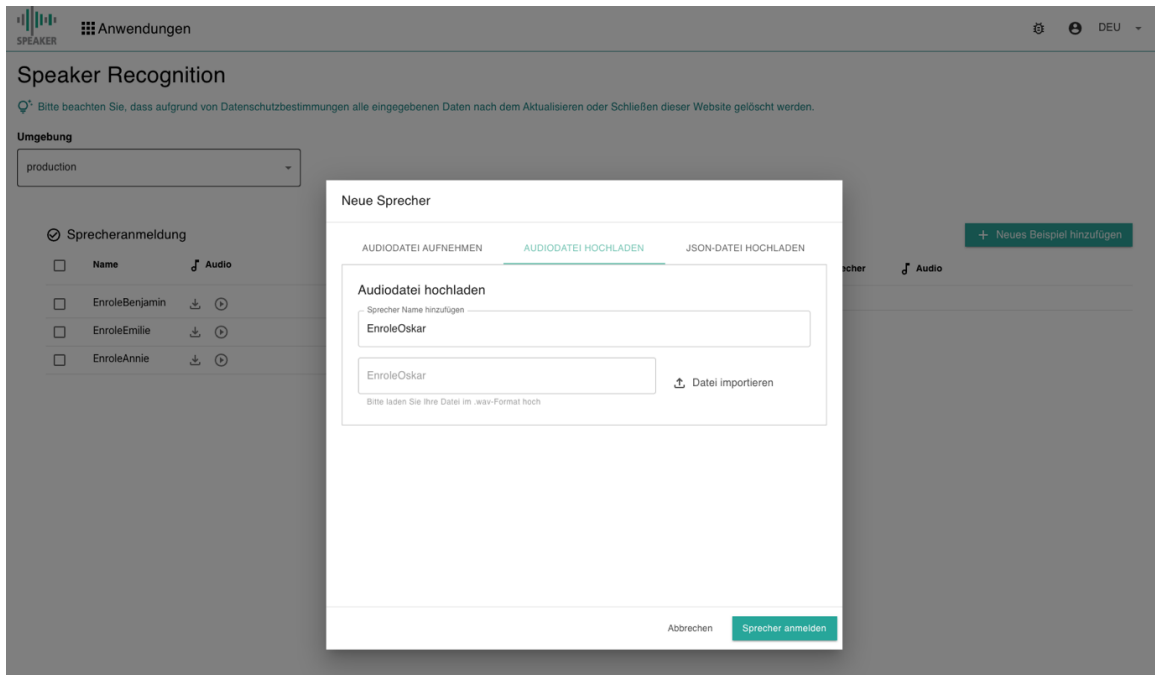


Abb. 11: SPEAKER Stage – Sprechererkennung – Sprecherregistrierung

Die Klassifizierung sowie die Darstellung der Ergebnisse der Klassifizierung sind in Abb. 12 dargestellt. Hier wurden testweise die verfügbaren Stimmen unserer Sprachsynthese mit jeweils unterschiedlichen Texten zum Registrieren sowie Klassifizieren verwendet.

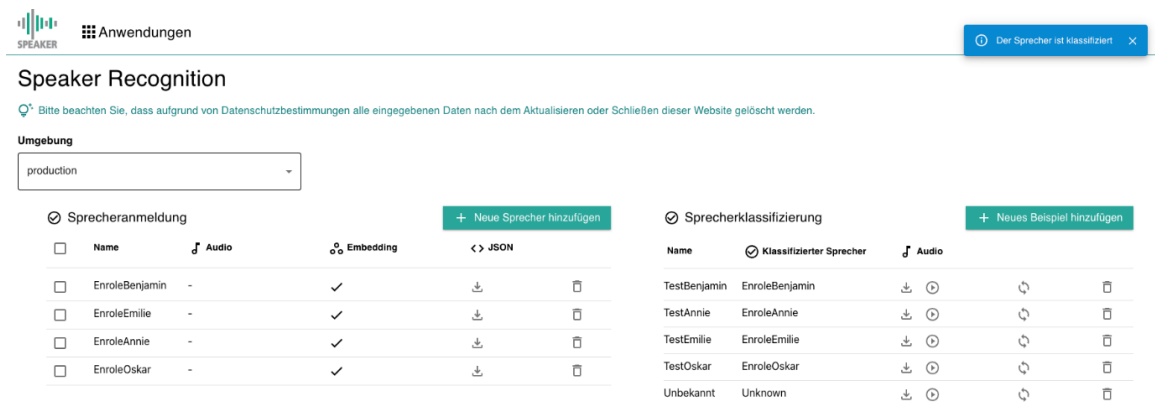


Abb. 12: SPEAKER Stage – Sprechererkennung – Sprecherklassifizierung

Keyword Spotter – Webapp: Im Web Frontend des Keyword Spotters kann man entweder das vorkonfigurierte Keyword ‘Sophia’ nutzen oder ein eigenes Konfigurieren. Wie in Abb. 13 dargestellt kann man hier über eine Audiodatei, oder über eine Audioaufnahme direkt im Web Frontend das Keyword konfigurieren. Alternativ kann man auch die Phonetische Transkription in ARPAbet zur Konfiguration verwenden.

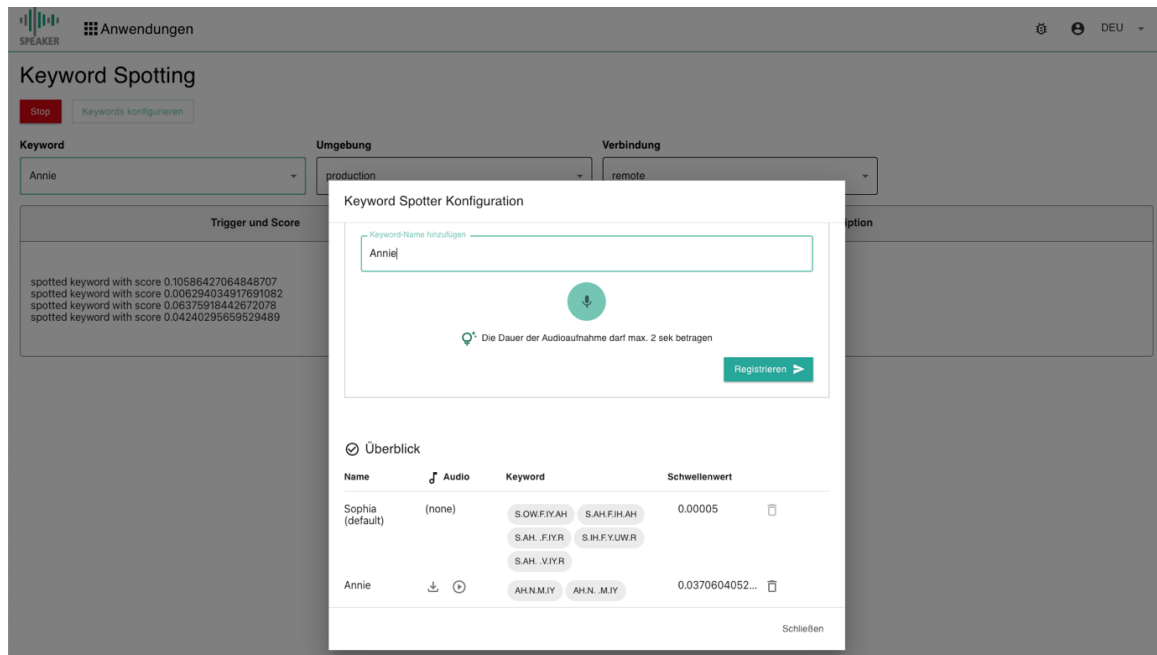


Abb. 13: SPEAKER Stage – Keyword Spotter – Keyword Konfiguration

Der Keyword Spotter kann dann entsprechend im Web Frontend getestet werden, welches in Abb. 14 abgebildet ist. Analog zu den Web Frontends der anderen Komponenten, hat man auch hier die Auswahl an Umgebungen. Ferner kann zwischen den konfigurierten Key Words gewählt werden. Eine Besonderheit des Keyword Spotter Web Frontends ist es, dass mit dem Drop Down zwischen der Instanz auf dem Cluster und einer lokal installierten Instanz gewählt werden kann.

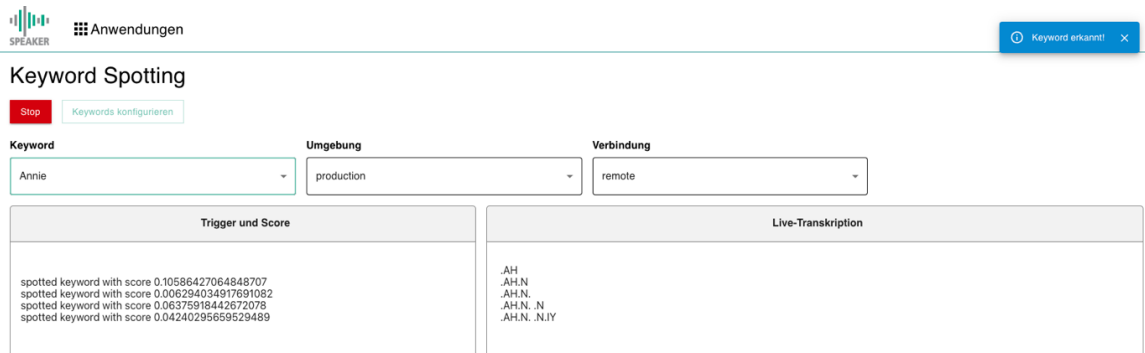


Abb. 14: SPEAKER Stage – Keyword Spotter

Assistant – Webapp: Dieses Web Frontend bietet die Möglichkeit verschiedene Kombinationen an Technologiekomponenten zu konfigurieren und zu testen. Die Reihenfolge der Verarbeitungskette ist jedoch vorgegeben, bestehend aus Keyword Spotter, Automatische Spracherkennung, Dialog Manager, sowie Sprachsynthese. Allerdings können Pipelines gewählt werden, welche nur einen Teil dieser Verarbeitungskette abbilden, falls z.B. keine Sprachausgabe gewünscht wird. In Abb. 15 sind die Konfigurationsmöglichkeiten dargestellt. Bei jeder Komponente kann man entsprechend der Konfiguration, analog der Web Frontends der jeweiligen Komponente ändern. Dies umfasst nicht nur den Service, welcher ein speziell nachtrainiertes Modul für den jeweiligen Partner sein kann, sondern auch die Umgeben, oder die Technologie-spezifischen Konfigurationen, wie zum Beispiel die Geschwindigkeit der Sprachsynthese.

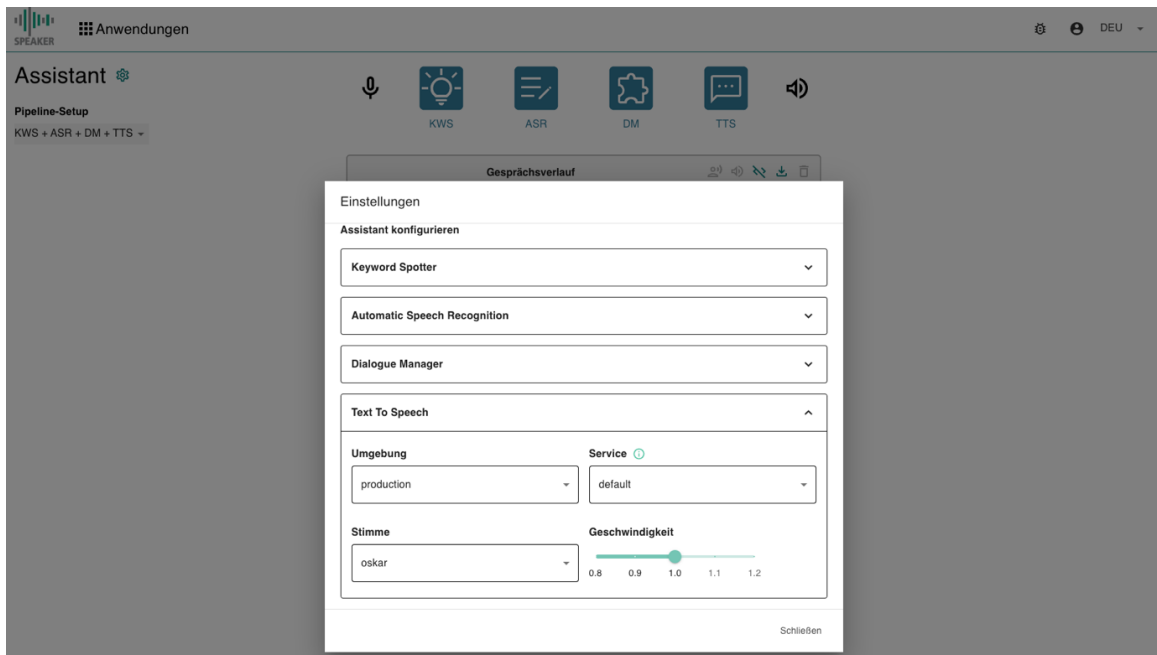


Abb. 15: Konfigurationsmöglichkeiten im Assistant Tab

In Abb. 16 ist die Webapp dargestellt in welcher der Sprachassistent getestet werden kann. Hier kann man natürlichsprachlich das Sprachassistentensystem testen und evaluieren.

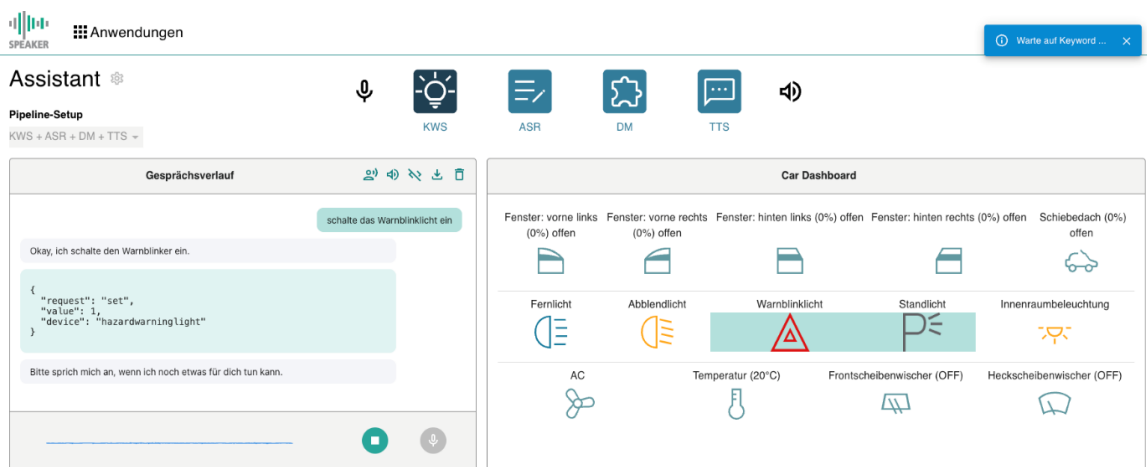


Abb. 16: Screenshot der WebApp zum Testen eines variabel zusammenstellbaren Gesamtsystems

Dialog Editor: Der Dialogeditor ist ein umfangreiches Web Frontend zur Ausgestaltung von deterministischen Dialogen. Wie in Abb. 17 zu sehen können in diesem Tool Dialoge in Form von Graphen erstellt werden. Einen detaillierten Überblick über die Möglichkeiten, welche dieses Tool bietet, kann man sich mit Hilfe der Dokumentation verschaffen, welche im Anhang dieses Berichts zu finden ist.

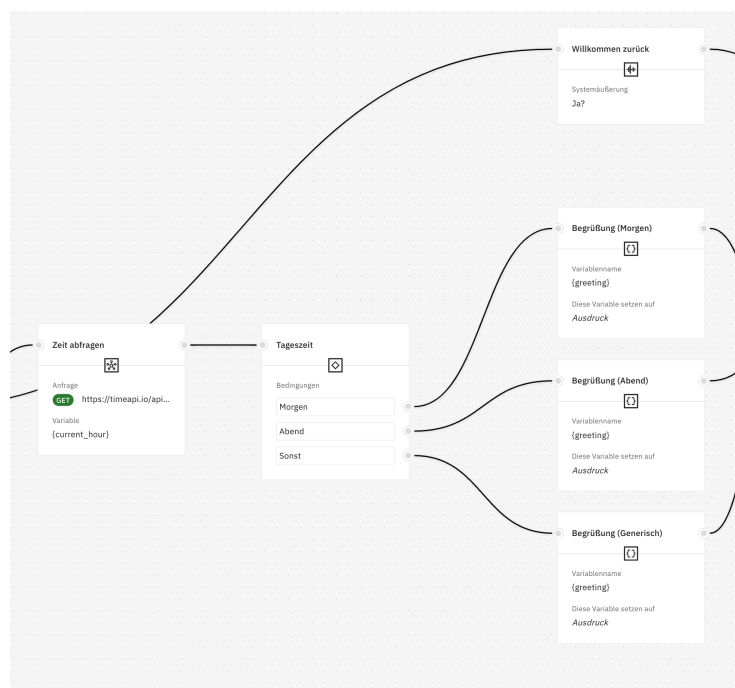


Abb. 17: Ausschnitt eines Dialoggraphs des Dialogeditors

Mit Hilfe einer umfangreichen Tool Chain können dann mit einem Click aus den Dialoggraphen Dialogmanager gebaut werden, welche dann auch in der SPEAKER Stage evaluiert werden können. Das Interface zum Erstellen von Dialogmanagern aus dem Dialog Editor heraus ist in Abb. 18 dargestellt.

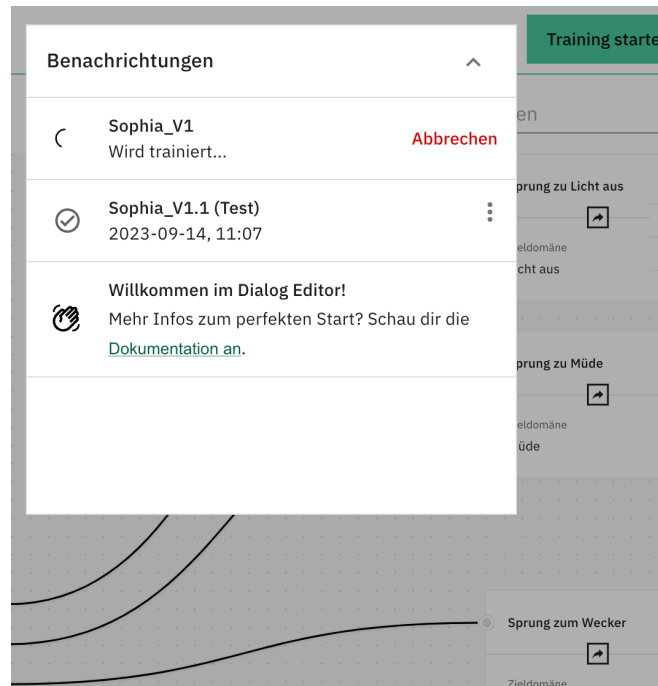


Abb. 18: Build Pipeline zum Erzeugen von Dialog Managern aus dem Dialog Editor heraus

Der daraus entstandene Dialog Manager kann dann direkt im Web Frontend des Dialog Editors getestet werden wie in Abb. 19 dargestellt.

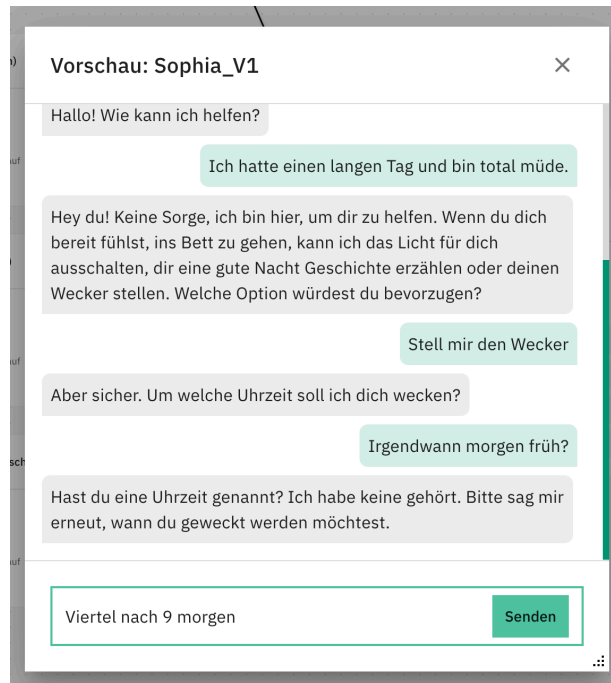


Abb. 19: Integriertes Interface zum Testen von Dialogmanagern

6.3.3 Audiovisuelle Spracherkennung

Aufgabenstellung: Der Beitrag der Technischen Universität Braunschweig (TUBS) im SPEAKER-Projekt beinhaltet die Entwicklung einer audiovisuellen Spracherkennungskomponente. Diese ist in der Lage, durch die Integration bzw. Fusion zusätzlicher visueller Informationen – konkret Kameraaufnahmen des Gesichts des Sprechenden – die Erkennungsgenauigkeit der automatischen Spracherkennung (ASR) besonders in störgeräuschbehafteten Umgebungen zu steigern.

Im Fokus des Teilvorhabens stand die Entwicklung und Optimierung eines deutsch- und englischsprachigen audiovisuellen Spracherkennungssystems. Darüber hinaus wurde ein Echtzeit-Demonstrator für audiovisuelle Spracherkennung implementiert.

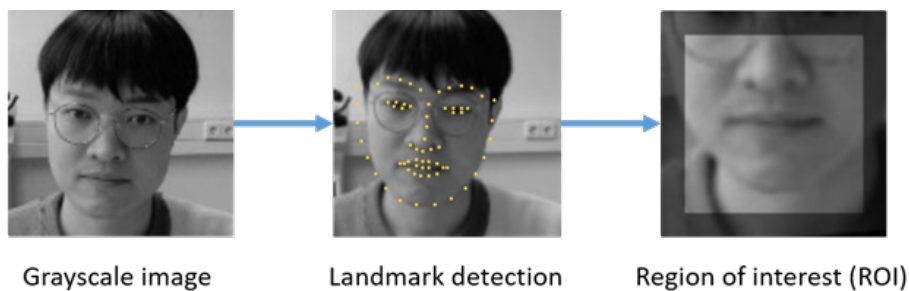


Abb. 20: Die echtzeitfähige Bildvorverarbeitung im Demonstrator

Benchmarks: Da wir beabsichtigten, ein audiovisuelles Spracherkennungssystem für englische und deutsche Sprache zu entwickeln – und darüber hinaus möglicherweise auch mehrsprachige AV-ASR-Aufgaben lösen wollen – haben wir drei Benchmarks ausgewählt, um unser AV-ASR-System zu trainieren und zu evaluieren. Die ersten beiden Benchmarks basieren auf zwei englischsprachigen audiovisuellen Datensätzen: Lip Reading Sentences 2 (LRS2) [33] und Lip Reading Sentences 3 (LRS3) [35]. Der dritte Benchmark ist der erst im Jahr 2023 veröffentlichte MuAViC-Datensatz [43], ein multilingualer audiovisueller Korpus für robuste Spracherkennung. Der MuAViC-Datensatz umfasst 8 Sprachen, darunter Deutsch, Spanisch und Französisch. Nach unserem Wissensstand ist der MuAViC-Datensatz der einzige öffentlich verfügbare, mehrsprachige audiovisuelle Datensatz mit Transkriptionen.

Visuelles Frontend: Für die robusten visuellen Merkmale wurde eine Gesichtsmerkmalserkennung sowie eine Sprachpausenerkennung für den Live-Demonstrator im Projekt entwickelt, mit dem Ziel, eine möglichst benutzerfreundliche Interaktion zu ermöglichen. Ein Beispiel der echtzeitfähigen Bildvorverarbeitung wird in Abbildung 1 gezeigt. Die einzelnen Frames eines Videos werden in Graustufen-Bilder umgewandelt und auf einen 96×96 Pixel großen Bereich (Region of Interest, ROI) basierend auf der Gesichtsausrichtung beschnitten. Dabei wird der Mund in den Videoframes zentriert, um robuste Merkmale für das neuronale Netzwerk zu generieren, auch wenn sich der Sprecher vor der Kamera bewegt. Die im Demonstrator integrierte Sprachpausenerkennung ermöglicht eine permanente Erkennung von Sprache, indem sie Pausen zwischen Sätzen erkennt und den Spracherkennungsvorgang in diesen automatisch startet. Dies erleichtert es dem Spracherkennungssystem, Sätze korrekt zu identifizieren und zu verarbeiten.

Englische audiovisuelle Spracherkennung mit einem Transformer-basierten Backend:

Das Backend wurde durch das moderne selbstüberwachte Lernen (Self-supervised Learning, SSL) [34] des audiovisuellen Encoders optimiert. Hierbei wurde der Transformer-Encoder auf großen Datenmengen an untranskribierten Sprechervideodaten vortrainiert, insbesondere auf 1759 Stunden Daten aus VoxCeleb2 und LRS3. Abbildung 2 zeigt die Wortfehlerrate (Word Error Rate, WER, kleiner = besser) auf Testdaten des LRS2 Datensatzes bei unterschiedlichen Signal-zu-Störgeräusch-Abständen (Signal-to-Noise Ratio, SNR) von -10dB bis 10dB. Die rote Kurve repräsentiert die WER des SSL-vortrainierten Modells, während die graue Kurve die WER des Modells ohne SSL zeigt. Deutlich ist zu erkennen, dass das selbstüberwachte Lernen bei allen SNRs eine erhebliche Verbesserung bewirkt. Im Vergleich zum Modell ohne SSL ergibt sich eine *absolute* Verbesserung von 6,9% bei 10dB Babble Noise und sogar 20,41% bei anspruchsvollen -10dB Babble Noise. Diese Ergebnisse belegen, dass SSL durch die Nutzung großer Mengen untranskribierter Daten die Robustheit des audiovisuellen Modells gegenüber Störgeräuschen erheblich steigern kann.

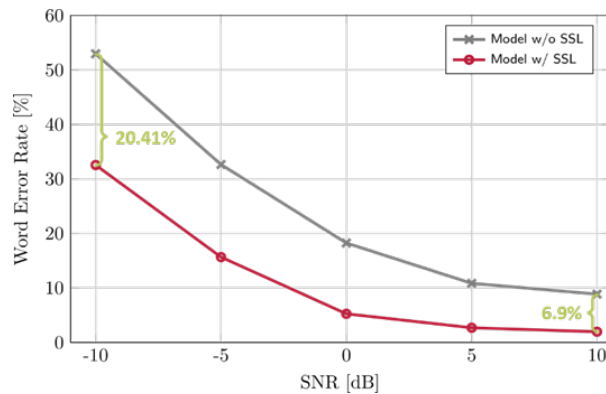


Abb. 21: WER (%) Vergleich von audiovisuellen Modellen ohne und mit selbstüberwachtem Vortraining (SSL) unter Babble Noise (Diffuser Stimmteppich, Beispiel: Cocktailparty)

Deutsche audiovisuelle Spracherkennung durch das Sprache-zu-Sprache-Transferlernen:

Wir konzentrierten uns auf die Erweiterung der existierenden Module hinsichtlich Multilingualität. Unsere englischen audiovisuellen Modelle setzten bereits einen neuen Benchmark (Stand der Technik), allerdings konnten diese auch nur durch die Verfügbarkeit großer englischsprachiger Datensätze trainiert werden. Für die deutsche Sprache sind insbesondere audiovisuelle Datensätze kaum verfügbar und es ist nicht möglich, das deutschsprachige Modell genau so aufwendig vorzutrainieren wie das englische Modell. Um trotzdem von der guten Erkennungsleistung des englischen Modells zu profitieren, verwendeten wir für das deutschsprachige Modell sogenanntes Transfer-Lernen. Diese Methode ist aufgrund der akustischen Ähnlichkeiten zwischen Sprachen für die automatische Spracherkennung mit geringen Datenressourcen weit verbreitet. Diese Ergebnisse wurden in einer Veröffentlichung zusammenfasst und wurden in Form eines Posters auf dem IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop 2023 in Taipei, Taiwan, präsentiert [44].

Echtzeitfähiger Demonstrator: Wir haben einen echtzeitfähigen Demonstrator implementiert. Insbesondere in störgeräuschbehafteten und Multi-Sprecher-Umgebungen übertrifft die audiovisuelle Spracherkennung (AV-ASR) die klassische automatische Spracherkennung (ASR), die ausschließlich auf akustischer Eingabe basiert. Der Demonstrator wurde in einer Show & Tell Session auf der ICASSP 2023 vorgestellt [45].

In diesem Demonstrator haben wir zunächst drei Transformer-Modelle, die vollständig auf dem Prinzip der Attention basieren, auf dem öffentlichen LRS3 Datensatz trainiert: AV-ASR, ein Lippenlesemodell und das konventionelle akustische ASR-Modell. AV-ASR und das Lippenlesemodell verwenden das vortrainierte (state-of-the-art) Audiovisual Hidden Unit BERT (AV-HuBERT), während die akustische ASR neu trainiert wurde (training from scratch).

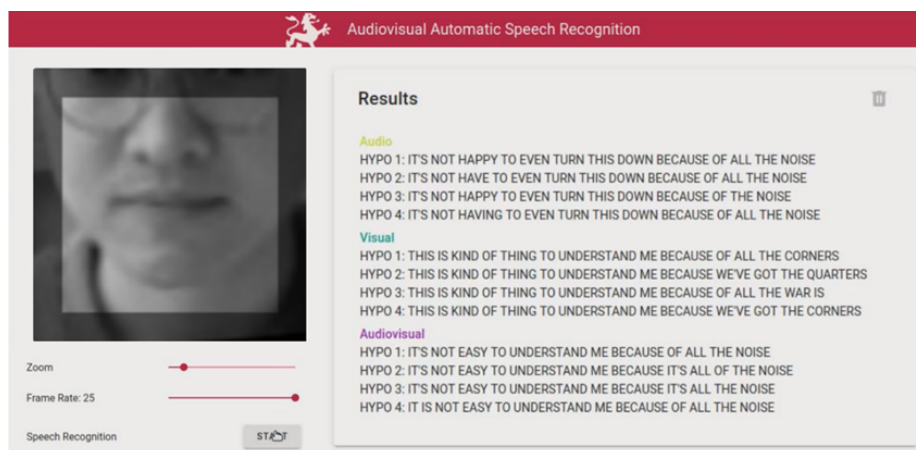


Abb. 22: Mit dem interaktiven Demonstrator kann audiovisuelle Spracherkennung individuell ausprobiert werden. Gezeigt werden jeweils vier Hypothesen (Sätze) nach Erkennung auf Audio-Basis (oben), Video-Basis (Mitte), und mit audiovisueller Spracherkennung (unten)

Weiterhin haben wir einen echtzeitfähigen webbasierten Demonstrator mit einer interaktiven Benutzeroberfläche für diese drei Modelle entwickelt. Benutzer können mit dem Demonstrator sprechen und der transkribierte Text dieser drei Modelle wird im Anschluss dem Nutzer angezeigt. Der Demonstrator erfasst die Stimme über ein Mikrofon und erkennt die Lippenbewegung des Sprechers über eine Webkamera. Die Erkennung von Gesichtsankerpunkten und die Ausrichtung des Gesichts sind in diesen Demonstrator integriert, um die Lippenregion im Eingabevideo zu zentrieren. Der Demonstrator bietet den Benutzern eine ansprechende und informative Möglichkeit, die Fähigkeiten dieser drei Modelle in verschiedenen Lärmpegeln zu erkunden und ihr Potenzial zu verstehen.

In Abbildung 3 ist die Nutzerschnittstelle des Demonstrators dargestellt. Mit einer Webcam zur Echtzeit-Eingabe der visuellen Information kann man die audiovisuelle Spracherkennung interaktiv ausprobieren. In diesem Beispiel konnte das gesprochene Wort „NOISE“ nur über die **Audio**-Modalität erkannt werden. Das Wort „UNDERSTAND“ wurde hingegen ausschließlich über die **Video**-Modalität erkannt. Nur die **audiovisuelle** Spracherkennung konnte durch die Kombination beider Modalitäten den Satz korrekt erkennen („HYPO 1“).

Publikationen: Die Veröffentlichung der Ergebnisse aus dem SPEAKER-Projekt erfolgte im Laufe des Projekts. Insgesamt haben wir sieben Beiträge auf sieben internationalen Tagungen im Bereich der Informationsfusion [46] [47], der Verbesserung vom Lippenlesen [48], [49], der effizienten audiovisuellen Spracherkennung [50], der deutschen audiovisuellen Spracherkennung [44] und des echtzeitfähigen Demonstrators vorgestellt [45].

6.3.4 Akustische Vorverarbeitung

Die Akustische Vorverarbeitung (Acoustic Frontend, AFE) zielt darauf ab, Störgeräusche bei der Aufnahme von gesprochenem Wort über Mikrofon zu unterdrücken. Um die Interaktion mit Sprachassistenten zu erleichtern, werden moderne AFE Methoden benötigt, da Ansätze mit konventioneller Signalverarbeitung in dynamischen akustischen Umgebungsbedingungen nicht robust genug funktionieren. Obwohl die akustische Vorverarbeitung von Mikrofonsignalen schon intensiv erforscht wurde haben sich in den letzten Jahren signifikante Leistungssteigerungen durch den Einsatz von Deep-Learning ergeben. Deswegen setzen wir auch im SPEAKER Projekt auf Deep-Learning basierte Methoden der Quellentrennung, welche das Nutzsignal verbessern und die Störgeräusche adaptiv herausfiltern. Dabei ist es wichtig, dass auch bei starker Bedämpfung von Störgeräuschen die Sprachverständlichkeit so gut wie möglich erhalten bleibt und nur wenige Audio-Artefakte entstehen. Üblicherweise werden Encoder-Decoder Architekturen, wie z.B. U-Net verwenden, die eine Art Maske schätzen anhand derer Störgeräuschbehaftete Teile in einer Zeit-Frequenz Darstellung des Mikrofonsignals bedämpft werden. Die meisten dieser Ansätze erfordern aufgrund der hohen Anzahl von Frequenzbins große Rechenkapazitäten, was dem Einsatz in Echtzeitsystemen entgegensteht. Für das SPEAKER Projekt wurde eine parallel am IIS entwickelte, sehr effiziente Lösung (Arbeitstitel UROC) für die einkanalige Störgeräuschbefreiung integriert und als Dienst auf der SPEAKER Plattform ausgerollt.

Die Anwendbarkeit der AFE als Vorverarbeitungsschritt vor der SPEAKER Spracherkennung wurde mittels Wortfehlerraten (Word Error Rate, WER) erprobt. Dabei wurde der BITS-US Datensatz zugrundegelegt, welcher vom Projektpartner BAS bereitgestellt wurde. Es handelt sich um einen Korpus mit Deutschen Sprachaufnahmen professioneller Sprecher (2x männlich, 2x weiblich), welcher unter Studiobedingungen aufgezeichnet wurde. Dieser Datensatz ist herausfordernd für Automatische Spracherkennung, da viele außergewöhnliche Wörter, Lehnwörter und Eigennamen vorkommen. Die enthaltenen Sätze sind in vielen Fällen semantisch nicht vorhersagbar. Aus diesem Grund liegt die mittlere WER bereits im ungestörten Fall, d.h. bei Verwendung der qualitativ hochwertigen Originalaufnahmen bei 14.7% für weibliche Sprecherinnen und 13.9% für männliche Sprecher (siehe Abbildung x). Werden nun Störgeräusche in verschiedenen Abstufungen hinzugefügt verschlechtert sich erwartungsgemäß die WER, im schlechtesten Fall bis zu 80%. Unsere Experimente zeigen, dass ein moderater Einsatz der SPEAKER AFE, also eine Dämpfung der Störgeräusche zwischen 7.5dB und 10.0 dB hier eine deutliche Verbesserung der WER erzielen kann. Im ungestörten Fall wirkt sich der Einsatz der AFE nicht negativ aus. Das bedeutet, wenn der geringe zusätzliche Rechenaufwand und eine kleine zusätzliche Latenz für den jeweiligen Anwendungsfall akzeptabel ist, so kann die SPEAKER AFE immer aktiv mitlaufen.

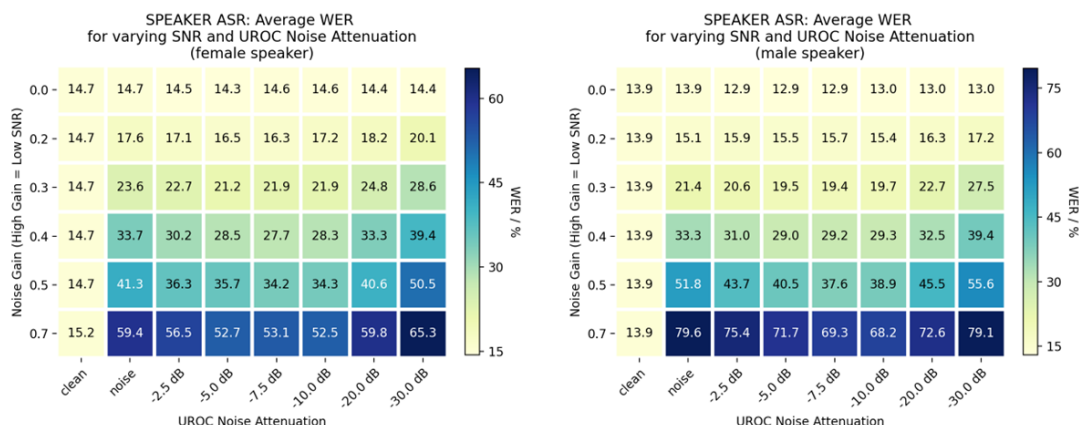


Abb. 23: Auswirkungen der AFE auf die Wortfehlerrate der SPEAKER ASR bei in Abhängigkeit von verschiedenen Stärken der Störgeräusche und Dämpfungsgrade.

Die Entwicklungsziele bzgl. AFE konnten in allen Punkten umgesetzt werden. Dies geschah unter Berücksichtigung der durch den Einsatz in der SPEAKER Plattform resultierenden Rahmenbedingungen hinsichtlich Echtzeitfähigkeit. Die Publikation [51] wurde auf ArXiv veröffentlicht und beschreibt im Detail, welche neuartigen Optimierungsmethoden es ermöglichten, auf 25% der Rechenlast des Stand der Technik zu kommen.

6.3.5 Keyword Erkennung

Die Keyword Erkennung fungiert im Rahmen eines Sprachassistenten als audiobasierter Anschaltknopf. In der Regel läuft sie auf einem normalerweise leistungsschwachen und sparsamen Endgerät und analysiert dabei permanent den aufgenommenen Audiodatenstrom. Falls das gewünschte Schlüsselwort im Audiosignal erkannt wird, wird dieser dann zur weiteren Verarbeitung an die Plattform in der Cloud weitergeleitet.

Das implementierte Keyword Spotting (KWS) Modul basiert dabei im Kern auf einer sehr kleinen und leistungsarmen Spracherkennung (ASR), wodurch es dem Nutzer ermöglicht wird, beliebige neue Schlüsselwörter zu konfigurieren. In der folgenden Grafik kann man schematisch die groben Verarbeitungsschritte des KWS-Moduls entnehmen.

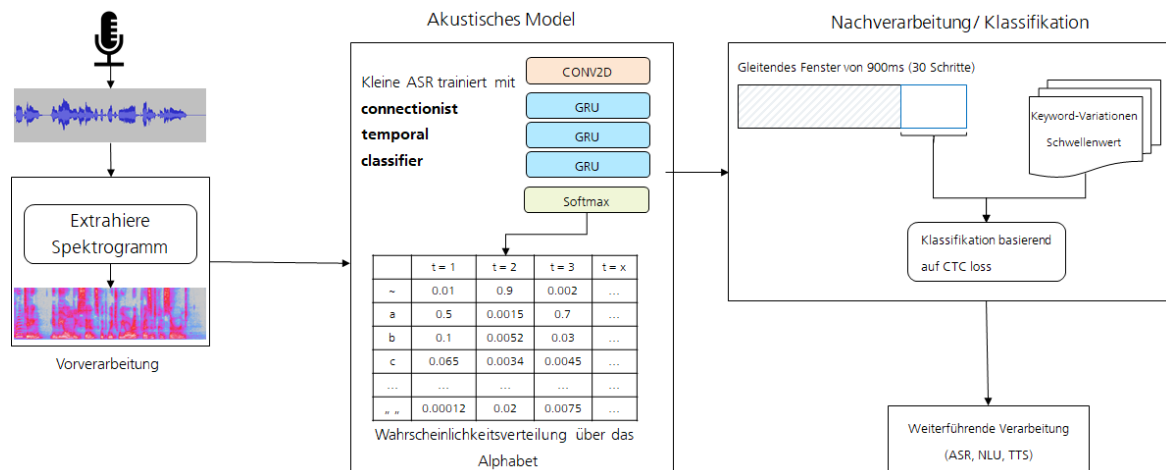


Abb. 24: Schematische Darstellung der Verarbeitungsschritte des KWS-Moduls

Im ersten Schritt wird der Audiostrom in ein Melspektrogramm umgewandelt. Dabei kann der Audiodatenstrom im Prinzip direkt vom Mikrofon abgegriffen werden, besser wäre es jedoch, vorher noch die akustische Vorverarbeitung (AFE) durchzuführen, um die Signalqualität zu verbessern und etwaige Störquellen, welche sich durch das simultane Abspielen von Medien wie z.B. Musik ergeben, herauszufiltern.

Das Melspektrogramm wird daraufhin an das akustische Modell weitergereicht. Das akustische Modell ist ein kleines neuronales Netz, dass – ähnlich zu einer herkömmlichen ASR – mittels des Connectionist Temporal Classifiers (CTC) dafür trainiert wurde, das Spektrogramm in eine Sequenz von Buchstaben umzusetzen. In jeder Zeiteinheit wird dabei vom Modell eine Wahrscheinlichkeitsverteilung über den vorliegenden Zeichensatz berechnet. Dabei fokussierte man sich auf den Phonemsatz von "CMUdict", einem Subset von "ARPAbet" was einen guten Kompromiss in Hinblick auf Qualität und Konfigurierbarkeit bieten sollte. Schlussendlich werden im letzten Schritt die resultierenden Wahrscheinlichkeitsverteilungen mittels eines gleitenden Fensters analysiert. Basierend auf der CTC Lossfunktion lässt sich dann die Wahrscheinlichkeit, dass eine bestimmte Zeichenfolge – das gewünschte Schlüsselwort - auftritt, berechnen. Übersteigt diese einen zuvor definierten Schwellenwert, gilt dies als positiver Ausschlag und führt zu einem detektierten Schlüsselwort.

In der ersten Iteration wurde ein neuronales Modell mit dem Tensorflow-Framework trainiert und in Python in ein auf WebSocket basierendes Server-Client-Gespann integriert sowie mittels Docker containerisiert. Der resultierende Backend-Container konnte dann auf der SPEAKER Plattform gehostet werden. Eine entsprechende Webapp wurde implementiert, was das Erproben des KWS-Moduls über den Browser ermöglichte.

Diese initiale Basisimplementierung wurde dann um eine Funktionalität zur Schätzung einer Schlüsselwortkonfiguration auf Basis von Beispielaudiodateien erweitert sowie mehrere Performanceoptimierungen unterzogen. Zu letzteren gehört der Umstieg auf die TensorFlow Lite Inferenzbibliothek, welche die Handhabbarkeit vereinfacht und speziell für Edge Devices optimiert ist. Der Backend-Container wurde mit ARM-basierten CPUs kompatibel gemacht und auf einem Raspberry Pi 4, Pi 3 und einem Smartspeaker von ONSEI (mit der selben CPU wie im Pi 3) erprobt. Das Modul konnte auf allen Plattformen in Echtzeit laufen. Auf dem Pi 4 belegte das Backend dabei ca. ~20% eines Kernes unter einem 32Bit System (+ zzgl. ca. 9% für die zugehörige Clientsoftware), auf dem Pi 3 ergaben sich 62% CPU-Last für das Backend (+ca. 12% Client) auf einem 32Bit System bzw. ca. 50% CPU-Last (+ca. 11% Client) unter einem 64Bit System.

In einer darauffolgenden Iteration wurde das KWS-Modul in die AFE integriert. Dies ermöglicht einen einfacheren Betrieb im Verbund mit der Audiovorverarbeitung sowie eine weitere Senkung der benötigten CPU-Last, da die AFE in C implementiert ist. Da viele Partner allerdings Python nutzen, wurde ein Python-Wrapper implementiert, der von der AFE Bibliothek Gebrauch macht.

Schlussendlich wurde noch ein reines Python Pip Package mit Fokus auf einfache Integrierbarkeit implementiert. Es nutzt analog zur Client-Server-Variante TensorFlow Lite und kommt mit einem minimalen Satz an Abhängigkeiten aus. Das Package bietet direkt ausführbare Beispielimplementierungen sowie hilfreiche Tools zum Debuggen und Konfigurieren des KWS-Moduls.

6.3.6 Sprechererkennung

Bei der Sprechererkennung (Speaker Recognition, SR) besteht das Ziel darin, einen Sprecher oder eine Sprecherin anhand charakteristischer Eigenschaften ihrer Stimmen zu identifizieren. Die Identität ist dabei nicht mit Klarnamen oder personenbezogenen Daten verknüpft, sondern bezieht sich lediglich auf einen eindeutigen Identifizierer (z.B. ein Pseudonym). Diese Komponente kann und sollte nicht für sicherheitskritische Anwendungen, wie z.B. Authentifizierungsaufgaben verwendet werden. Viel eher ist sie als unterstützende Funktionalität bei Protokollierungsaufgaben gedacht. Zum Beispiel, um bei der automatischen Spracherkennung eines Gesprächs zwischen zwei Personen das gesprochene Wort den jeweiligen Sprechern zuzuordnen.

Im SPEAKER Projekt war das Ziel, eine textunabhängige SR zu entwickeln, in diesem Fall ist der Inhalt des gesprochenen Worts irrelevant und hat keinen Einfluss auf die Erkennung. Die Erkennung der Sprecheridentität erfolgt durch den Vergleich von so genannten Speaker Embeddings (vergleichbar einem akustischem "Fingerabdruck") eines unbekannten Sprechers mit einer begrenzten Menge von Embeddings bekannter Sprecher, die zuvor in einer Datenbank registriert

wurden. Die SR kennt folglich nur die Identitäten dieser zuvor registrierten Sprecher. Jeder Sprecher, der dieser Gruppe nicht zugehörig ist sollte von der SR als unbekannter Sprecher erkannt werden.

Die SPEAKER SR verwendet eine Sprachaufnahme des zu identifizierenden Sprechers als Eingabe und extrahiert daraus das zuvor erwähnte Speaker Embedding. Diese Merkmalsextraktion ist so trainiert, dass unabhängig von der Dauer der Sprachaufnahme ein Merkmalsvektor fester Länge für die gesamte Aufnahme berechnet wird. Wie bereits erwähnt wird das Embedding aus den charakteristischen Merkmalen der Sprecherstimme berechnet, wobei die Grundannahme ist, dass die Stimmen verschiedener Sprecher zu unterschiedlichen Embeddings führen, während die Verwendung der Stimme desselben Sprechers, selbst unter verschiedenen akustischen Aufnahmebedingungen, idealerweise zu einem sehr ähnlichen Embedding führen sollte. Um die Identität eines unbekannten Sprechers zu ermitteln wird folglich dessen Embedding mit allen Embeddings der registrierten Sprecher verglichen. Wenn die Kosinus-Distanz zur nächsten Übereinstimmung kleiner als ein bestimmter Schwellenwert ist, wird der Sprecher als identifiziert betrachtet.

Ungefähr zur Hälfte der SPEAKER Projektlaufzeit startete die Entwicklung einer eigenen SR mittels eines Benchmarks mehrerer Back-End-Architekturen mit dem Ziel, die geeignetste Architektur für den Einsatz im Projekt zu bestimmen. Für die vorgesehenen Anwendungs-Szenarien stellte sich die Titanet Architektur als am vielversprechendsten heraus. Wie auch bei den anderen Sprachtechnologiekomponenten wurde dieses Modell als Docker Container mit gRPC Schnittstellen implementiert. In der darauffolgenden Phase wurde ein Python-basiertes Client-Skript für das SPEAKER SDK entwickelt. Dieses Skript ermöglicht die Kommunikation mit dem auf der SPEAKER Plattform deployten Sprechererkennungssystem.

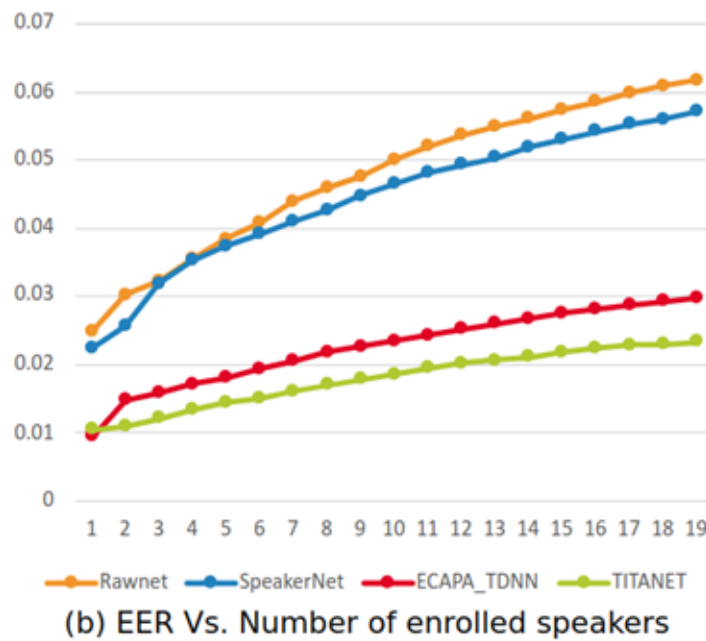


Abb. 25: Evaluationsergebnisse verschiedener neuronaler SR Architekturen

Anschließend erfolgte der erste Einsatz des SR-Servers auf dem Cluster der SPEAKER-Plattform. In den Folgemonaten wurde die Backend-Implementierung weiter verfeinert und auf ausgewählten Architekturen basierend auf den Benchmarking-Ergebnissen optimiert. Eine enge Zusammenarbeit mit dem UX-Team führte zur Entwicklung eines neuen Frontend-Designs für die Sprechererkennung. Diese Webapp ermöglicht die Registrierung von Sprechern über verschiedene Quellen wie Audiodateien, Mikrofonaufnahmen oder bestehende Datenbankdateien. Zudem ermöglicht es die Erkennung über verschiedene Eingabequellen wie Audiodateien oder Mikrofonaufnahmen.

Im letzten Jahr des SPEAKER Projekts wurde eine zusätzliche Phase des Benchmarkings und der Feinabstimmung des SR für verschiedene Anwendungsszenarien durchgeführt. Daraufhin erfolgte die endgültige Integration des SR-Servers in die SPEAKER-Umgebung. Diese Integration ermöglicht die reibungslose Kommunikation mit dem Frontend sowie den im Cluster eingesetzten SR-Modulen.

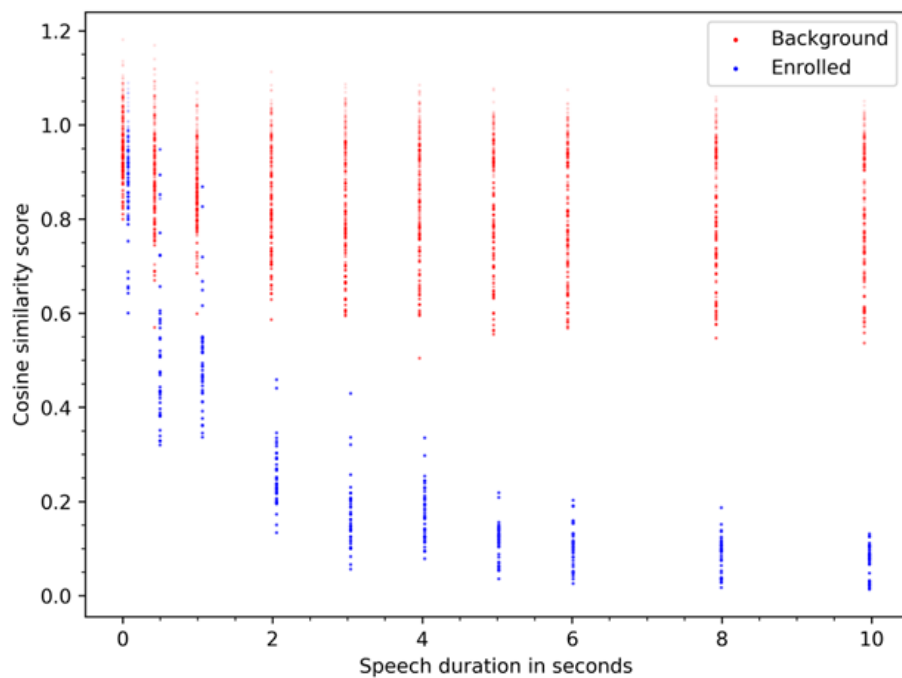


Abb. 26: Güte der Trennung zwischen registrierten und unbekannten Speaker Embeddings in Abhängigkeit von der Dauer des Sprachsignals

Die Leistung des SPEAKER SR Systems hängt von der Fähigkeit des Modells ab, ähnliche Embeddings für denselben Sprecher und unterscheidbare Embeddings für andere Sprecher zu erzeugen. Das System ist leistungsfähiger bei längeren Audiodateien. Bei der Registrierung gibt es keine Beschränkung hinsichtlich der Länge der eingehenden Audiodateien, da diese nur einmal aufgenommen und offline verarbeitet werden. Bei unserem Benchmarking haben wir festgestellt, dass eine Audiodauer von 5 Sekunden zu einer zuverlässigen Sprechererkennung und eine Audiodauer von 3 Sekunden zu einem zufriedenstellenden Ergebnis führt. In diesem Zusammenhang empfehlen wir mindestens 10 Sekunden Audio für die Registrierung und mindestens 3 Sekunden Audio für die Erkennung, um eine ähnliche Leistung wie bei den Benchmark-Ergebnissen zu erzielen.

6.3.7 Automatische Spracherkennung (ASR)

Es wurde ein Modul zur Spracherkennung für die SPEAKER Plattform implementiert. Hierzu wurde das bestehende ASR-Modul vom Fraunhofer IAIS um die entsprechenden gRPC Schnittstellen für die Plattform erweitert und containerisiert. Basierend Anhand von Marktbeobachtungen und Literaturrecherche wurden Entwicklungen hinsichtlich der ASR beobachtet und das Modul für die SPEAKER Plattform überarbeitet. Hierzu wurde unter anderem Möglichkeiten zur Vokabular- und

Grammatikanpassung untersucht und realisiert. Insbesondere durch Datenaugmentation der Trainingsdaten für den Sprachkenner mit Typischen Störszenarien (Hall und Hintergrundgeräusche) konnte die Robustheit der Spracherkennung bei gestörten Aufzeichnungen verbessert werden. Auch neue End-to-end Ansätze wurden untersucht, aufgrund ihrer Ressourcenanforderungen und längeren Verarbeitungszeit noch nicht auf der Plattform realisiert. Um eine Anpassung der Modelle für die *automatische Spracherkennung (ASR)* zu realisieren, wurde zunächst eine Pipeline zum lokalen vollautomatischen Trainieren des Spracherkennungsmoduls implementiert. Des Weiteren wurde ein Benchmark System zur Auswertung und Visualisierung der Performance des ASR-Modul hinsichtlich der Erkennungsqualität erstellt. Basierend auf Feedback der Partner und entsprechenden Trainings- und Evaluationsdaten wurden die Modelle stetig angepasst, beispielsweise durch das Hinzufügen von Vokabular-Listen oder auch Aussprachelexika, und deren Performance kontinuierlich evaluiert. Abhängig vom Anwendungsfall konnte die Erkennungsleistung deutlich gesteigert werden. Durch die Integration in der Plattform wurde die Verwendbarkeit des ASR-Moduls deutlich vereinfacht.

6.3.8 Natural Language Understanding (NLU)

Datenkuration und Datenmanagement: Es wurde ein Datenmanagement-System implementiert, um annotierte Trainingsdaten effizient zu verwalten und zu versionieren. Das System ermöglicht die zentrale Speicherung von Daten, die Versionierung von Datensätzen und die Nachverfolgung von Änderungen von z.B. Annotationen.

Ein wichtiger Aspekt beim Training von neuronalen Modellen zur Intent-Klassifizierung ist die Verfügbarkeit von möglichst vielen annotierten Trainingsdaten, die in der Praxis aber typischerweise nicht vorhanden sind. Es wurde daher ein regelbasierter Prozess konzeptioniert und implementiert, um Trainingsdaten für die Intent-Klassifizierung zu augmentieren. Der Prozess nutzt Regel-basierte Vorschriften, um neue Datensätze aus bestehenden Datensätzen zu generieren bzw. diese zu erweitern. Verschiedene Ansätze zur automatischen Augmentierung von Trainingsdatensätzen, basierend auf generativen Sprachmodellen wie ChatGPT, wurden untersucht. Diese Modelle können dann dazu verwendet werden, synthetische Erweiterungen von Datensätze, z.B. durch Umformulierungen einer Nutzereingabe, zu erzeugen.

Des Weiteren wurden Trainingsdatensätze zur Intent-Klassifizierung in Deutsch durch maschinelle Übersetzung großer englischer Datensätze erstellt. Etwaige Übersetzungsfehler wurden manuell korrigiert. Ein Benchmarking-Datensatz in deutscher Sprache wurde zur Evaluierung von Sprachmodellen zur Erzeugung von sogenannten Satz-Embeddings erstellt. Der Datensatz basiert auf

dem Maß der Semantic Text Similarity zur Beurteilung der semantischen Ähnlichkeit von Sätzen. Die Datensätze wurden via Crowdsourcing gewonnen und sind auf die Anwendung im Bereich Sprachassistenzsysteme zugeschnitten, da sie mehrstufige Gesprächsabläufe berücksichtigen. Die Ergebnisse wurden in der Publikation [52] veröffentlicht. Die Verbundpartner aus dem Projekt wurden durch Workshops und Handlungshilfen im Bereich Datensammlung und -annotation beraten und unterstützt.

Natural Language Understanding Module: Im Rahmen der Projektarbeit wurden verschiedene Ansätze und Workflows bzgl. Fragestellungen des natürlichen Sprachverstehens (Natural Language Understanding, NLU) untersucht und entwickelt. In zielorientierten Sprachassistenzsystemen sind die Aufgaben der Intent-Klassifikation, also die Erkennung der Nutzerabsicht, sowie die Entitätenerkennung (entity recognition) von vorrangiger Bedeutung und daher wurde der Schwerpunkt der Entwicklungsarbeiten auf diese beiden Problemstellungen gelegt.

Intent-Klassifikation: Durch den Austausch mit den Anwendungspartnern aus dem SPEAKER Projekt wurden wichtige Anforderungen abgeleitet, die bei den Entwicklungen besonders berücksichtigt werden sollten. Zum einen liegen in der Regel nur sehr wenige Beispieldaten zum Training von neuronalen Modellen zur Intent-Klassifikation vor. Daher wurde der Fokus auf Ansätze des sogenannten Few-Shot Learnings gelegt, um die bestmögliche Erkennungsraten für den Fall zu erreichen, dass nur sehr wenige Anwendungs-spezifische Trainingsdaten vorliegen. Ein weiterer Aspekt ist, dass in typischen Anwendungsszenarien eine große Anzahl an Intents an unterschiedlichen Stellen des Dialogverlaufs erkannt werden müssen, wobei nur ein Bruchteil der gesamten Intents an einem bestimmten Zeitpunkt im Dialog relevant sind. Es ist daher von Vorteil, viele dedizierte Intent-Klassifikatoren zu entwickeln, anstatt eines großen Modells, dass alle Intents gleichzeitig abdeckt. Um den Speicherbedarf vieler dedizierten Modelle zu verringern, wurden Ansätze verfolgt, die eine Wiederverwendung der speicheraufwändigen Komponenten innerhalb eines Dialogverlaufs ermöglicht. Dies wurde dadurch erreicht, dass der Vorgang der Intent-Klassifikation in zwei Schritte aufgeteilt wurde, nämlich die Erzeugung von Featurevektoren, die die semantische Bedeutung einer Nutzeingabe repräsentieren (die sogenannten Satz-Embeddings) und der eigentlichen Klassifikation basierend auf den Embeddings. Die Satz-Embeddings werden dabei für alle Intent-Klassifizierer aus dem gleichen Sprachmodell erzeugt, wohingegen die Klassifizierung der Intents Kontext-abhängig in separaten, speziell optimierten Modulen erfolgt. Dazu wurden Methoden von sogenannten Prototypical Networks geeignet angepasst. Da der Speicherbedarf für die Sprachmodelle zur Erzeugung der Embeddings um ein Vielfaches größer ist als der der Klassifizierer, kann so eine signifikante Reduzierung des Speicherbedarfs erzielt werden.

Im Rahmen der Projektarbeit wurden frei verfügbare Sprachmodellen zur Erzeugung von Satz-Embeddings mittels Fine-Tuning weiter angepasst, um deren Leistungsfähigkeit in Bereich Intent-Klassifikation zu optimieren. Dieses Task-spezifische Pre-Training wurde sowohl für Deutsche als auch für Englische Modelle durchgeführt. Da für das Training des Deutschen Sprachmodells keine geeigneten Trainingsdaten vorlagen, wurden entsprechende englische Daten maschinell übersetzt und anschließend manuell korrigiert. Experimente mit Benchmark-Datensätzen haben gezeigt, dass die entsprechenden Intent-Klassifizierer nicht nur effizient implementiert werden können, sondern auch bessere Klassifizierungsergebnisse erzielen als kommerzielle Konkurrenzansätze.

Bei der Umsetzung von Modulen zur Intent-Klassifikation in Sprachassistenzsystemen ist es für die Verbesserung der User Experience wichtig, nicht nur eine Entscheidung zu treffen, welche Nutzerabsicht vorliegt, sondern auch eine Aussage zu treffen, wie sicher sich der Klassifikationsalgorithmus bzgl. der Entscheidung ist. Dies ermöglicht erweiterte Strategien beim Fehlermanagement umzusetzen. Wenn z.B. die Konfidenz bei der Intent-Klassifizierung gering ist, kann man die entsprechende Information verwenden, um beim Nutzer nachzufragen, ob er das Klassifizierungsergebnis bestätigen kann oder ob das System eine Fehlermeldung zurückgeben muss. Dazu wurden Methoden aus dem Bereich der Clusteranalyse angepasst, um aussagekräftige Konfidenz-Information zusätzlich zu dem Klassifizierungsergebnis zu erzeugen.

Ein weiterer Aspekt beim Fehlermanagement in Sprachassistenzsystemen ist es, Anfragen der Nutzer zu detektieren, die nicht vom System bearbeitet werden können. Dazu wurde eine generische Out-of-Scope Detektion entwickelt, die nicht zusätzlich trainiert werden muss, sondern ausschließlich Informationen der Intent-Klassifizierer verwendet.

Um den Anwendungsentwicklern eine niederschwellige Möglichkeit zur Verwendung von neuronalen Modellen zur Intent-Klassifikation an die Hand zu geben, wurde eine automatisierte Trainingspipeline implementiert, die Anwendungs-spezifische Trainingsdaten zusammen mit vortrainierten Embeddingmodellen kombiniert, um leistungsfähige Klassifizierer zu erzeugen.

Es wurden zusätzliche, weiterführende Untersuchungen von Ansätzen zur Reduzierung des Speicher- und Rechenbedarfs bei der Few-Shot Intentklassifizierung mit Prototypical Network durchgeführt. Die Ergebnisse wurden erfolgreich auf einer Fachkonferenz publiziert: [53] Anna Sauer, Shima Asaadi, and Fabian Küch. „Knowledge Distillation Meets Few-Shot Learning: An Approach for Few-Shot Intent Classification Within and Across Domains” . In Proceedings of the 4th Workshop on NLP for Conversational AI, Association for Computational Linguistics.

In vielen praktischen Anwendungsfällen werden nicht nur neuronale Modelle zur Intent-Klassifizierung verwendet, sondern auch einfache Ansätze, die Schlüsselwörter erkennen oder Nutzereingaben mittels Fuzzy Matching klassifizieren. Entsprechende Module zur Intent-Klassifizierung wurden implementiert und können durch Anwendungsentwickler auf einfache Weise konfiguriert werden.

Erkennung von Entitäten: Neben der Erkennung der Nutzerabsicht ist es in Sprachassistenzsystemen wichtig weitere Informationen aus der Spracheingabe zu extrahieren, die notwendig sind, um eine vorliegende Aufgabe zu bearbeiten. Zum Beispiel benötigt durch Erfüllung einer Nutzeranfrage ein Taxi zu bestellen zusätzlich Informationen über Entitäten wie Zeitpunkt, Abfahrtsort und das Ziel.

Im Rahmen der Projektarbeit wurden Module zur Entitätenerkennung entwickelt, die auf Schlüsselwörtern und logischen Ausdrücken basieren. Diese sind können als sogenannte Systementitäten vordefiniert und direkt durch den Anwendungsentwickler genutzt werden. Alternativ können die entsprechenden Definitionen Anwendungs-abhängig vom Anwendungsentwickler definiert werden.

In vielen Anwendungsfällen ist es nicht praktikabel die Entitätenerkennung ausschließlich auf Schlüsselwörtern oder logischen Ausdrücken basieren zu lassen. Daher wurden auch Methoden zur Verwendung von neuronalen Modellen untersucht bzw. neu entwickelt. Ein wichtiger praxisrelevanter Aspekt bei der Entwicklung neuer Ansätze ist die Leistungsfähigkeit in Few-Shot Learning Szenarien, d.h. für den Fall, dass nur sehr wenig Trainingsdaten vorhanden sind. Die im Rahmen der Projektarbeit entwickelte Methode basiert auf der Verwendung von natürlich-sprachlichen Annotation zur Definition der Entitäten, wodurch die Leistungsfähigkeit von allgemeinen Sprachmodellen als Ausgangsbasis effektiv ausgenutzt werden kann. Im Gegensatz zu anderen publizierten Ansätzen in denen zwei Modelle nötig sind, um einerseits den Bereich der Wörter zu identifizieren, die einer Entität zugeordnet sind und andererseits das Label der Entität zu bestimmen, benötigt der entwickelte Ansatz nur ein einziges Modell, welches beide Aufgabenschritte gemeinsam adressiert. Dadurch kann bei gleicher Erkennungsrate der benötigte Speicher- und Rechenbedarf signifikant reduziert werden.

6.3.9 Dialog Management (DM)

Der Dialog Manager stellt ein zentrales Modul eines Sprachassistenten dar. Er kontrolliert die Interaktion des Endnutzers mit dem System unter Berücksichtigung von vordefinierten Dialogflüssen, die für eine Anwendung vorgesehen sind. Der Eingang des Dialog Managers besteht aus der transkribierten Nutzereingabe, die von der automatischen Spracherkennung geliefert wird, sowie

optionale, zusätzliche Kontext-abhängige Nebeninformationen. Der Ausgang des Dialog Managers besteht aus Systemantworten in Form von Text oder Event-IDs, die von der Sprachassistenten-Anwendung interpretiert oder weiterverarbeitet werden können. Zum Beispiel werden typischerweise Text-Antworten des Systems mit Hilfe des Text-to-Speech (TTS) Modules in eine Sprachausgabe gewandelt.

Bei der Realisierung des SPEAKER Dialog Managers wurde ein Hybrid-Ansatz gewählt, in dem die Dialogflüsse in Form von deterministischen Graphen umgesetzt werden, Aufgaben im Bereich des natürlichen Sprachverstehens (siehe oben Intent-Klassifikation und Entitätenerkennung) unter Verwendung moderner Verfahren des maschinellen Lernens mit neuronalen Netzen gelöst werden können. Dies ermöglicht eine Kombination aus nachvollziehbarer und deterministischer Abbildung der Nutzerführung innerhalb einer Anwendung und der Verwendung moderner Verfahren aus dem Bereich künstliche Intelligenz.

Ausgangspunkt bei der Entwicklung des Dialog Managers war das Open-Source Framework Rasa [54], welches in geeigneter Weise erweitert und angepasst wurde, um die Anforderungen an das SPEAKER Sprachassistentensystem zu erfüllen. Besonderer Fokus bei den Erweiterungen wurde auf die NLU-Pipeline zum Trainieren und Orchestrieren der jeweiligen neuronalen Module zur Domänen-, und Intent-Klassifikation sowie der Entitätenerkennung gelegt. Diese Erweiterungen erlauben eine wesentlich flexiblere und bzgl. der Rechen- und Speicherressourcen effizienteren Umsetzung der NLU-Aufgabenstellungen innerhalb eines Sprachassistentensystems ohne Kompromisse bei der Leistungsfähigkeit der neuronalen Modelle zur Folge zu haben.

Die Spezifikation und Implementierung der NLU-Pipeline Architektur innerhalb des Dialogmanagers basieren auf der Anforderungsanalyse der SPEAKER Anwendungsfälle. Die gewählte Architektur erlaubt es effizient eine Vielzahl von spezifischen Intent- bzw. Domänen-Klassifizierern auf ein gemeinsam genutztes Sprachmodell aufzusetzen. Der NLU Controller zur Orchestrierung des Ensembles aus NLU Modulen erlaubt es den Bausteinen anwendungsspezifischer Dialoggraphen jeweils optimierte Module zuzuordnen, anstatt z.B. ein einziges weitgefasstes NLU Modell einzubinden. Der Controller ermöglicht auch eine flexible Kombination von sowohl Schlüsselwort bzw. Regel-basierten als auch neuronalen NLU Modulen.

Die Konfiguration des Dialogmanagers zur Umsetzung Anwendungs-spezifischer Dialogflüsse in Sprachassistentensystemen erfolgt über den Dialogeditor. Der Dialogeditor ist ein graphisches Werkzeug zur Erstellung der Dialoggraphen und zur Spezifikation von Trainingsdaten für die relevanten neuronalen Intent-Klassifizierer. Der Dialogmanager stellt eine Vielzahl grundlegender, funktionaler Blöcke zur Umsetzung von umfangreichen Dialogflüssen zur Verfügung, die über

den Dialogeditor in Form von Knoten konfiguriert und verknüpft werden können. Mehr Informationen über den Dialogeditor sind in dem entsprechenden Abschnitt 6.3.2 dieses Berichts näher ausgeführt. Beispiele für die erwähnten Knoten sind Verzweigungen basierend auf Ergebnissen der Intent-Klassifikation oder allgemeineren logischen Ausdrücken, dynamische Erzeugung von Systemantworten, Sprungknoten, API-Knoten zur Anbindung externer Services, Aktionsknoten zur Ausführung von Codefragmenten oder andere vordefinierten Aktionen, Knoten zum Befüllen von Variablen mit Werten aus der Entitätenerkennung. Über den Dialogeditor können auch das Verhalten von erweitertem Fehlermanagement (z.B. wenn eine Nutzeranfrage nicht erkannt wurde) definiert werden.

Die eigentliche Umsetzung der Dialogkonfiguration aus dem Dialogeditor in eine entsprechende Dialogmanager-Instanz, die über die SPEAKER-Plattform ansprechbar ist, erfolgt über die sogenannte Engine-Factory, die im Rahmen des Projekts entwickelt wurde. Die Engine-Factory ist ein eigenständiger Service, der über eine Web-API angesprochen werden kann. Der Engine-Factory werden die Konfigurationsinformationen aus dem Dialogeditor inklusive den Trainingsdaten der neuronalen NLU-Modelle übergeben und somit der Build- und Deployment-Prozess des Dialogmanagers in Form eines Docker-Containers angestoßen. Dieser Prozess beinhaltet auch das Training der neuronalen NLU-Modelle mit den spezifizierten Trainingsdaten. Nach Abschluss des Deployment-Prozesses auf die SPEAKER-Plattform ist der Dialogmanager unmittelbar erreichbar und kann durch eine Sprachassistentz-Applikation oder den Testmodus im Dialogeditor angesprochen werden. Dieser vollautomatisiert Build- und Deployment-Prozess ermöglicht es den Dialogdesignern die erzeugten Dialogflüsse direkt und interaktiv zu testen, ohne dass sie technisches Wissen bzgl. Build- und Deployment-Prozessen oder fachliches Wissen zum Training von neuronalen Netzen benötigen und beschleunigt dadurch Entwicklungszyklen von Sprachassistentensystemen signifikant.

Im Rahmen des Projekts wurden in Zusammenarbeit mit den Anwendungspartnern mehrere Anwendungsspezifische Dialogmanager entwickelt und zu Test- bzw. Evaluationszwecken in der SPEAKER-Plattform zur Verfügung gestellt. Des Weiteren wurde ein Demonstrationssystem zur Steuerung von Fahrzeugkomponenten (Licht, Fensterheber, Klimaanlage, etc.) durch den Fahrer erfolgreich umgesetzt.

6.3.10 Text to Speech Sprachsynthese (TTS)

Die Sprachsynthese ist eine unerlässliche Ausgabeschicht, um bei einer Sprachschnittstelle textuelle Informationen als gesprochenes Wort an die Nutzer weiterzugeben. Nach aktuellem Stand der Technik wird die höchste Natürlichkeit und Sprachqualität durch Neuronale TTS erzielt. Hierbei

kommen typischerweise spezialisierte Modelle zum Einsatz, die jeweils darauf trainiert sind, verschiedene Repräsentationen von Sprache in einander zu überführen. Die folgende Abbildung verdeutlicht dies an einem dreistufigen System, welches in dieser Form auch in der SPEAKER TTS implementiert wurde.

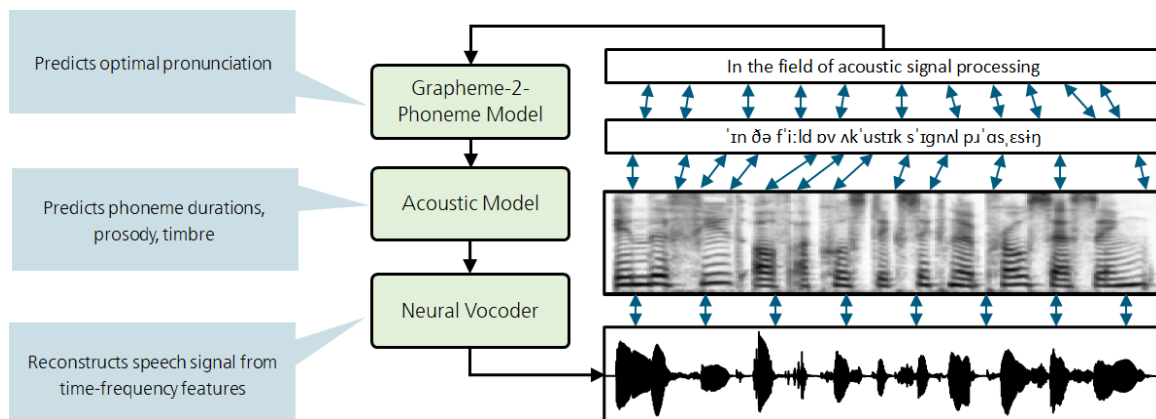


Abb. 27: Überblick über die Modelle, die für die Umwandlung von geschriebenem Text in gesprochene Sprache notwendig sind.

Am Anfang der Kette steht die textuelle Information, also eine Folge von Buchstaben und Satzzeichen. Diese wird zuerst mittels eines spezialisierten Modells für Graphem-zu-Phonem Konvertierung (G2P) in eine Folge von Phonem-Symbolen überführt. Diese geben explizit die optimale Aussprache der geschriebenen Worte wieder. Bei der SPEAKER TTS wird zur Kodierung der Phoneme das Internationale Phonetische Alphabet (IPA) verwendet, welches aber äquivalent auch mittels X-SAMPA Kodierung abgebildet werden kann. Die G2P Konvertierung kann als Übersetzungsaufgabe zwischen verschiedenen symbolischen Sprachrepräsentationen interpretiert werden, wobei regelbasierte Ansätze hier schon vergleichsweise gute Ergebnisse erzielen. Allerdings ist das Programmieren und Erweitern entsprechender Regelwerke sehr aufwendig und setzt umfangreiches Expertenwissen in Phonetik voraus. Somit hat es sich etabliert, datengetriebene Modelle wie z.B. Finite State Transducer oder Neuronale Netze für diese Aufgabe zu trainieren. Da es in vielen Sprachen eine ganze Reihe von Ausnahmefällen in der phonetischen Aussprache gibt (z.B. Lehnwörter, Abkürzungen, etc.) werden zusätzlich Aussprachelexika verwendet und die automatische G2P greift immer dann, wenn Wörter nicht im Lexikon enthalten sind.

Als zweiter Verarbeitungsbaustein folgt das so genannte Akustische Modell, welches anhand der Phonemsequenz eine Sequenz von Sprachmerkmalen prädiziert. Üblicherweise dienen Spektro-

gramme mit Mel-skaliertem Frequenzverlauf als Merkmalsdarstellung. Diese kodieren implizit wichtige Spracheigenschaften wie Phonemdauer, Grundfrequenzverlauf, Energie, Stimmhaftigkeit und Formanten. Da die Melspektrogramme eine viel höhere zeitliche Dimensionalität (Anzahl von Elementen) aufweisen als die zugrundeliegenden Phonemsequenzen ist eine der Hauptaufgaben des Akustischen Modells, diese unterschiedlichen Zeitachsen nicht äquidistant aufeinander abzubilden.

Der letzte Verarbeitungsschritt wird durch den neuronalen Vokoder realisiert, welcher anhand der Mel Spektrogramme ein plausibles Sprachsignal prädiziert. Hierbei muss wiederum eine deutliche Erhöhung der zeitlichen Dimension im Vergleich zum Mel Spektrogramm realisiert werden, damit ein hochqualitatives Zeitsignal mit einer ausreichend hohen Abtastrate (mindestens 16 kHz) synthetisiert werden kann. Im Unterschied zum Akustischen Modell kann aber eine äquidistante Verarbeitung erfolgen, da die Zuordnung der Zeitachsen zu einander linear ist.

Die Bewertung der Leistungsfähigkeit von TTS Systemen erfolgt anhand verschiedener Kriterien. Aus technischer Sicht ist der Echtzeitfaktor (Real Time Factor, RTF) eine wichtige Maßzahl, anhand derer abgeschätzt werden kann, wie effizient die Sprachsynthese erfolgt. Dieser Faktor muss größer als 1.0 sein, um Sprachsynthese in Echtzeit zu ermöglichen. Hier ein anschauliches Beispiel zur Interpretation des Echtzeitfaktors: Bei einem RTF von 60.0 wird ein Sprachsignal mit einer Gesamtdauer von einer Minute innerhalb einer Sekunde Rechenzeit synthetisiert. Im Mittel entspricht eine Minute Sprachdauer einem Text mit einer Länge von ungefähr 1000 Zeichen. Insbesondere für Sprachassistentenanwendungen ist ein hoher RTF wichtig, da er auch mit einer geringen Latenz korreliert, was für die Nutzer zu einer besseren User Experience führt.

Natürlich sind auch die Sprachverständlichkeit und die Natürlichkeit wichtige Indikatoren für die Gesamtqualität einer TTS. Zur objektiven Bewertung existieren eine Vielzahl von Methoden, wie z.B. Berechnung der Word Error Rate (WER) zwischen dem Text der TTS Anfrage und einer Automatischen Spracherkennung auf dem synthetisierten Sprachsignal. Allerdings sind subjektive Hörtests noch immer der zu bevorzugende Goldstandard, weswegen diese Methode im Laufe des Projekts immer wieder eingesetzt wurde.

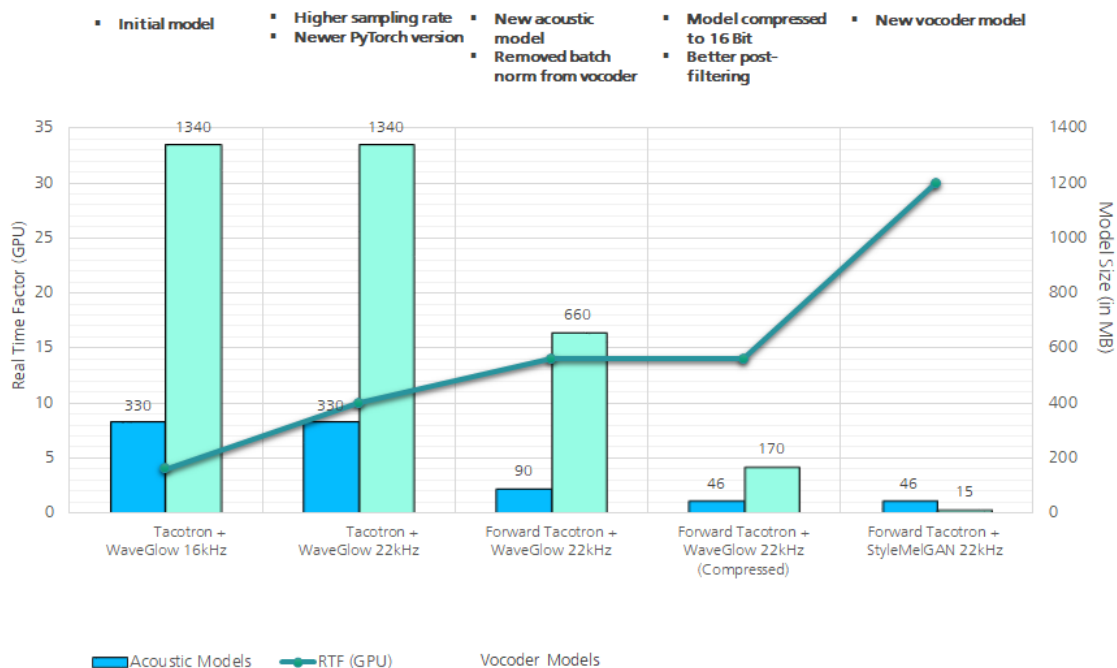


Abb. 28: Minimierung der Modellgröße und Erhöhung der Effizienz der SPEAKER TTS während des Projektzeitraums.

Zu Beginn des SPEAKER Projekts startete die Entwicklung der Komponente zur Sprachsynthese nach dem damaligen Stand der Technik mit der Implementierung einer Serveranwendung, welche gRPC Schnittstellen zur standardisierten Kommunikation auf der SPEAKER Plattform bietet. Als Technologiekerne kamen das Akustische Modell Tacotron2 und der Neuronale Vokoder WaveGlow zum Einsatz. Die Kombination dieser beiden Modelle erreichte einen RTF 5 auf GPU Hardware, war also unter bestimmten Voraussetzungen echtzeitfähig.

Bereits in Voruntersuchungen vor dem Start des SPEAKER Projekts zeichnete sich ab, dass der Umfang bestehender Sprachdatensätze in deutscher Sprache (z.B. beim Projektpartner BAS) nicht ausreichend sein würde, um eine neuronale TTS mit guter Qualität zu trainieren. Aus diesem Grund wurde ein neuer Korpus mit Skripten erstellt, welcher dann nach einem Casting geeigneter Sprecher/innen in einem professionellen Tonstudio aufgenommen wurde. Dieser proprietäre Datensatz wurde im Laufe des Projekt erweitert und die textuellen Annotationen mit Hilfe von semi-automatischen Methoden immer weiter verbessert. Unter dem internen Arbeitstitel OMONDI bildet der Korpus heute die Grundlage für das Produktivsystem der SPEAKER TTS.

Die erste wichtige Weiterentwicklung der TTS wurde durch die Umstellung des Akustischen Modells auf ForwardTacotron ermöglicht. Im Gegensatz zu Tacotron2 entfällt hier die autoregressive Prädiktion und der Attention Mechanismus. Dies führt zum einen zu einem höheren RTF, hat aber auch Vorteile bezüglich der Robustheit der Spektrogrammprediktion. Der Attention Mechanismus

wird durch einen Duration Prädiktor in Kombination mit einem Length Regulator ersetzt. Vereinfacht gesprochen schätzt der Duration Prediktor für jedes Phonem in der Eingabesequenz eine plausible Dauer, welche dann durch den Length Regulator mittels nicht-äquidistanter Überabtastung umgesetzt wird.

Die zweite wichtige Weiterentwicklung wurde durch die Entwicklung und Anpassung von StyleMelGAN ermöglicht, welcher den zuvor verwendeten WaveGlow Vocoder ersetzt. WaveGlow basiert auf Normalizing Flows und ist deswegen mit relativ hohem Speicherbedarf und Rechenaufwand verbunden. StyleMelGAN basiert auf Generative Adversarial Networks und wurde ursprünglich am Fraunhofer IIS für die Sprachkodierung entwickelt. Durch gezielte Weiterentwicklung konnte das Modell erfolgreich auf die Verwendung im TTS Kontext angepasst werden.

Die Kombination aus proprietären Trainingsdaten, ForwardTacotron und StyleMelGAN legte die Grundlage für das finale Produktivsystem der SPEAKER TTS. Durch mehrmalige Minimierung des Speicher- und Rechenbedarfs wurde der RTF auf über 30 erhöht, während die Modellgröße auf knapp über 60 MB herunterskaliert werden konnte. Das Gesamtsystem wurde mittels Hörtests auf seine Eignung hinsichtlich Deutscher Sprachsynthese untersucht und die Ergebnisse auf der ITG 2021 veröffentlicht [55].

Im weiteren Verlauf wurden umfangreiche Stresstests auf Cloud&Heat Hardware durchgeführt. Dies diente zum einen für die Ermittlung von Leistungsparametern der TTS, zum anderen für die Abschätzung der Skalierfähigkeit der SPEAKER Plattform. Da erste SPEAKER Partner bereits Interesse an einem on-premise Deployment der SPEAKER TTS angemeldet hatten wurde weiterer Entwicklungsaufwand in die Containerisierung (mittels Docker) und Sicherheitshärtung der Modelle gesteckt. Die Modelle wurden portiert um sie als dynamische C++ Bibliotheken mit verschlüsselten Modellparametern ausliefern zu können.

Ein weiterer Hörtest unter Einbeziehung mehrerer Partner aus dem SPEAKER Konsortium wurde durchgeführt, um einen Vergleich der Qualität zu zwei namhaften US-Anbietern für TTS ziehen zu können. Insbesondere die SAP AG konnte eine überdurchschnittliche hohe Anzahl von Testhörern (>120) aktivieren. Die Qualität unserer deutschen Stimmen wurde klar gegenüber der Konkurrenz bevorzugt. Allerdings gab es auch kritische Einschätzungen an den englischen Stimmen. Mit der Beschaffung weiterer englischer Trainingsdaten konnte dies in der Folge behoben werden.

In der Folge kam mit der Implementierung der Unterstützung für die Speech Synthesis Markup Language (SSML) eine wichtige Funktionalität für den Einsatz in der Praxis und die Interoperabilität mit anderen Anbietern hinzu. SSML kann beispielsweise genutzt werden, um die genaue Aus-

sprache bestimmter Begriffe zu spezifizieren, Sprechpausen zu setzen oder zwischen verschiedenen Sprechern umzuschalten. Eine Anforderung von SSML ist die Möglichkeit, Prosodie (Sprachmelodie) und Emphasis (Betonung) detailliert steuern zu können. Dies konnte durch umfangreiche Erweiterungen am Modell und entsprechende Erweiterung der Trainingsdaten erzielt werden. Insbesondere zusätzliche Prädiktoren für Grundfrequenz, Energie und Stimmhaftigkeit spielen hier eine entscheidende Rolle.

Durch intensive Zusammenarbeit mit den Projektpartnern DATEV und SAP wurde deutlich, wie wichtig die G2P Qualität insbesondere im Hinblick auf die korrekte Aussprache von Fachbegriffen ist. So wurde z.B. eine spezialisierte Erweiterung des Aussprachelexikons eigens für mehr als 6000 Begriffe der DATEV entwickelt und ausgeliefert. Mit der SAP wurde ein weiterer Hörtest hinsichtlich der Eignung verschiedener G2P Modelle für das Produktivsystem konzipiert und durchgeführt. Erfreulicherweise wurde das effizienteste Modell (Phonetisaurus) auch qualitativ am besten bewertet und wurde in der Folge in das Produktivsystem integriert.

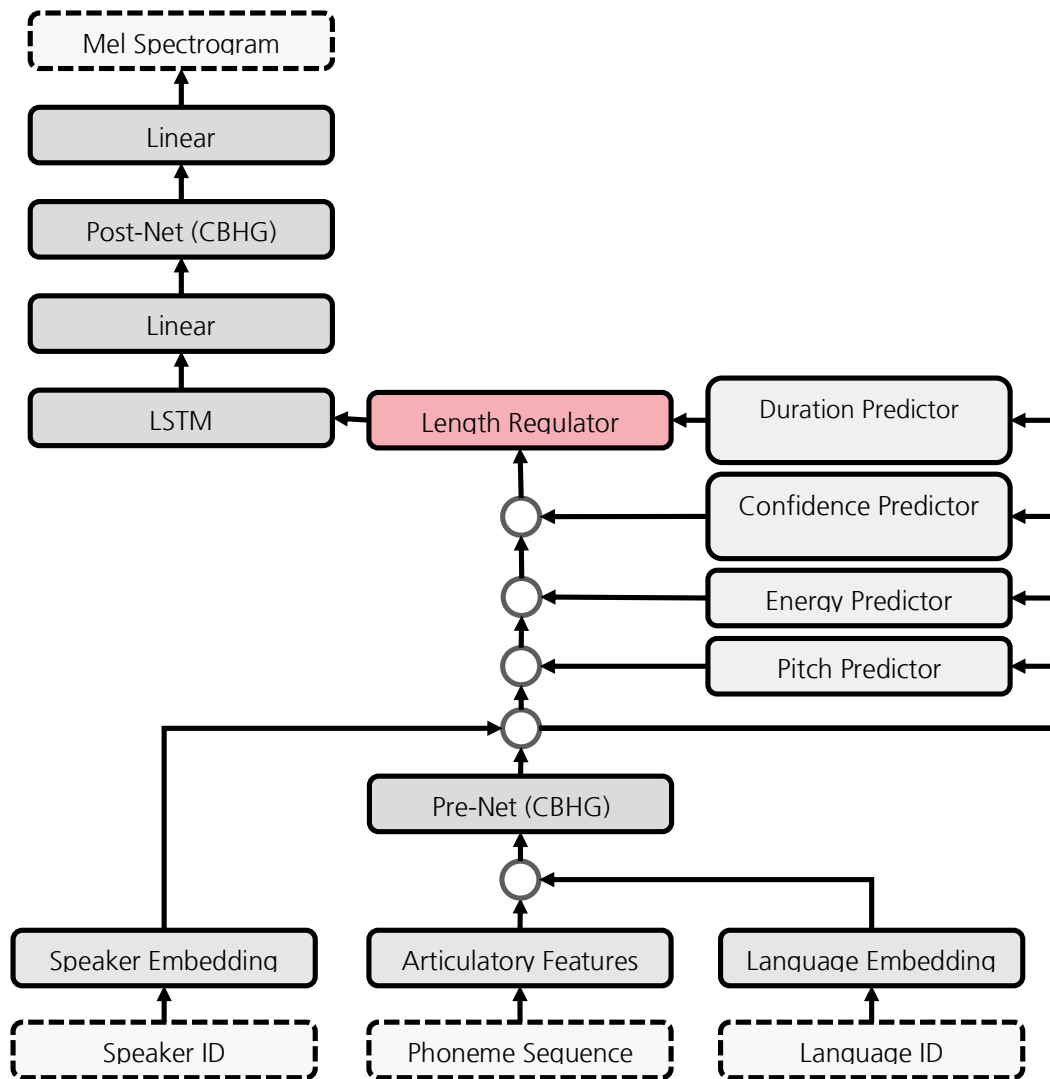


Abb. 29: Überblick über die multi-lingualen Erweiterungen des Akustischen Modells zum Abschluss des SPEAKER Projekts.

Eine letzte Anforderung, die mehrfach im Projekt von Partnern geäußert wurde war die Multilingualität der SPEAKER TTS. Für viele Anwendungen ist es wichtig, dass innerhalb eines Satzes nahtlos in andere Landessprachen umgeschaltet werden kann (Code-Switching), um zum Beispiel Englische Eigennamen korrekt auszusprechen. Eine erneute Erweiterung der Architektur des Akustischen Modells war notwendig, um diese Funktionalität zu ermöglichen. Zum einen wurden Artikulatorische Merkmale verwendet, um die interne Repräsentation von Phonemen besser zu trainieren, zum anderen kommen werden interne Repräsentationen (Embeddings) der Landessprache gelernt, um hier die Sprecheridentität von der gesprochenen Sprache getrennt zu halten. Somit können nun alle Sprecher, mit denen das System trainiert wird, alle dem System bekannten Sprachen sprechen auch wenn keine Trainingsdaten in einer spezifischen Sprache vorliegen.

6.3.11 Machine Reading Comprehension

Machine Reading Comprehension ist ein wichtiger Baustein von Dialogsystemen um Kundendokumente direkt in den Konversationsfluss einzubinden und schnell und einfach zugänglich zu machen. Im Rahmen von SPEAKER haben wir verschiedene Architekturen und Frameworks evaluiert und weiterentwickelt. Das Ergebnis ist ein eigenständiges MRC Modul, welches die Hauptaufgaben der MRC, Dokument-Retrieval und Beantwortung der Frage anhand der Dokumente, anbietet. Darüber hinaus entstand eine Reranker Komponente, um Retrieval-Ergebnisse zu verbessern.

Zu Beginn von SPEAKER identifizierten wir vier vielversprechende Tools und Frameworks für MRC, DrQA, T5-basierte MRC, Jack the Reader und Haystack. Für unsere Analyse und Bewertung der Frameworks sind verschiedene Faktoren wichtig. Neben state-of-the-art Performance sollten alle Teil-Module einer MRC abgedeckt sein, kein Vendor-Lock-In entstehen, Modelle einfach und schnell angepasst werden können sowie eigene Architekturen implementiert werden können. Zudem lag unser Fokus auf einer breiten Anwendbarkeit für vielfältige Use Cases der SPEAKER Partner.

DrQA ist ein auf sehr große Textmengen ausgelegtes Framework. Der Retriever verwendet einen einfachen TF-IDF Ansatz. Der Reader verwendet ein vortrainiertes RNN. Evaluationen auf Benchmark Datensätzen zeigten, dass gerade der einfache Retriever-Ansatz bei thematisch komplexen Dokumenten nicht ausreichend gut funktioniert. Die Verwendung von RNNs ist aufgrund ihrer langsamen Laufzeit proportional zur Dokumentenlänge bei großen Dokumenten ungeeignet.

Als Frameworks für Text Comprehension kamen Jack the Reader und Encoder-Decoder Modelle der T5-Familie in Betracht. Jack the Reader ermöglicht eine einfache Anbindung eigener, annotierter Datensätze zum Trainieren eigener Reader Modelle. Als Referenzimplementierung wird FastQA verwendet. Eigene Architekturen können implementiert werden, jedoch sind Modelle von FastQA nicht mit anderen bei huggingface vorhandenen Modellen kompatibel. Damit ist ein Wechsel zu anderen Reader Frameworks schwierig.

Modelle der T5-Familie sind Encoder-Decoder Transformer, welche auf einer Vielzahl von Tasks vortrainiert wurden. Speziell angepasste T5 Varianten spezialisieren sich auf Reading Comprehension und auf Question-Answer Pair Generierung, wobei gerade der zweite Punkt im Unternehmenskontext relevant ist, da selten annotierte Daten vorliegen und der Aufwand, Daten labeln zu lassen, zeit- und kostenintensiv ist. Als letztes komplettes MRC Framework haben wir Haystack von Deepset evaluiert. Haystack umfasst Retriever und Reader Komponenten, arbeitet mit Huggingface Modellen und erlaubt das Trainieren eigener Modelle.

Als Ergebnis haben wir uns für einen kombinierten Ansatz entschieden. Unsere MRC folgt dem Schema Reader - Retriever, erweitert es aber um einen Reranker, um Suchqualität zu verbessern. Haystack-basierte Retriever und Reader können in unserer Implementierung benutzt werden, aber eigene Architekturen sind ebenfalls möglich. Außerdem verwenden wir T5 Modelle um Trainingsdaten zu synthetisieren.

Da zu Beginn des SPEAKER Projekts noch keine Unternehmensdaten zum Testen der MRC bereitstanden, haben wir uns auf den Use Case Climate Change konzentriert. Dabei entstand ein Datensatz zum Thema Klimawandel aus öffentlich zugänglichen Quellen, wie verschiedener Nachrichtenportale und Berichten des UN Climate Change Panel. Der Datensatz umfasst 7400 Paragraphen und 21.081 Frage-Antwort-Paare, die teilweise Fraunhofer-intern, als auch per Crowd Sourcing annotiert wurden. Als Retriever verwenden wir einen Dense-Passage-Retriever, welcher die gegebenen Paragraphen und die Nutzeranfrage mit Sentence-Bert Embeddings indiziert. Ein von uns adaptiertes ALBERT Modell kommt als Reader zum Einsatz. Die Evaluation des Reader lieferte Scores wie F1 (0.81) und BLEU (0.678) und sind damit für einen Nicht-Benchmark-Datensatz auf hohem Niveau.

Im Rahmen des Projektes wendeten wir unser MRC Modul auf einen Teil der Steuerelemente aus LEXInform der DATEV an. Ohne angepasste Reader Modelle gelingt nur die Beantwortung einfacher Faktenfragen. Gerade Formulierungen im Steuerrecht sind äußerst kompliziert, weshalb ein fine-tuning nötig ist. Verschiedene Basismodelle wurden auf diese Art angepasst, wobei die QA-Pairs synthetisch erstellt wurden. Auch hier zeigt sich aber, dass generierte Fragen häufig nur einfache Faktenfragen sind, die zwar die Domänenanpassung ermöglichen, aber Reader-Ergebnisse für schwierige Fragen kaum verbessern.

Als Ausweg bieten sich Prompt-basierte LLM-Ansätze für den Reader an, welche zum Ende des SPEAKER Projektes begonnen wurden, aber nicht abschließend evaluiert werden konnten.

6.3.12 Knowledge Graph Question Answering

Neben MRC spielt die Beantwortung von Fragen anhand von Knowledge Graphen in Dialogsystemen eine wichtige Rolle. Knowledge Graphen können in Unternehmen sehr spezifisches Wissen und Prozesse modellieren. Außerdem können schnell, zielgerichtet aktualisiert werden. Wissen aus Knowledge Graphen zur Verfügung zu stellen ist allerdings meist nur für Experten möglich, weshalb eine natürlichsprachliche Schnittstelle essentiell ist.

Ziel im SPEAKER Projekt war es, ein KGQA Modul zu implementieren, welches alle Teilaufgaben in einer Pipeline bearbeitet. Dies beinhalten Named Entity Extraction, Entity und Relation Linking, SPARQL Query Generation, Time Extraction und Answer Verbalization.

In einer ersten Version wurde eine Template KGQA entwickelt, welche alle diese Teilmodule umfasst. Für Named Entity Extraction kommen Open Source Tools wie Spacy und Stanza zum Einsatz. Eine erste Version selbst-entwickelte Version des Entity und Relation Linking nutzt in Elasticsearch indizierte Entitäten und Relationen und verlinkt diese mittels BM25 Suche. Zur Generierung von SPARQL Queries stehen vordefinierte SPARQL Templates zur Verfügung. Anhand gelinkter Entitäten und Relationen werden SPARQL Query Templates ausgewählt und ausgefüllt. Zur Verbalisierung werden mit SPARQL Templates assoziierte Verbalization Templates benutzt.

Ein erster Test erfolgt anhand von Covid19 Daten, welche öffentlich von der John Hopkins University und vom Robert Koch Institut als JSON bzw. CSV bereitgestellt sind. Diese werden über den Smart Data Connector in einen eigens modellierten Knowledge Graph überführt. Insgesamt 14 Templates stehen zur Verfügung, mit denen Fragen zu Fallzahlen und Impfungen beantwortet werden können. Die Zahl der beantwortbaren Fragen ist um ein Vielfaches höher, da alle Templates konfigurierbar sind. Die Ergebnisse der Komponente sind gut, Fragen auch in abgewandelter Form können beantwortet werden. Außerdem sind Aggregationen bspw. über die Anzahl der Fallzahlen verschiedener Länder möglich, was gerade für Datenanalysten relevant ist. Gleichzeitig zeigt sich aber der Template Ansatz als schwer erweiterbar und im Allgemeinen als zu starr, um dynamisch auf neue Daten und unerwartete Fragen zu reagieren.

Eine zweite Version der KGQA setzte bei Verbesserungen an zwei Stellen an. Einerseits sollte Entity und Relation Linking robuster gegen semantisch ähnliche, aber topographisch verschiedene Variationen gemacht werden. Andererseits sollte in dynamisches System zur Generierung von SPARQL Queries entwickelt werden.

Für das Entity und Relation Linking wurde ein Ansatz entwickelt, der Entitäten mit bestehenden Embedding Modellen in einen Vector-Space projiziert und per k-NN Suche auffindbar macht. Für das Relation Linking werden alle einer identifizierten Entität zugeordneten Relationen sowie die natürlichsprachliche Nutzeranfrage mit einem Vectorizer embedded und deren Cosine Similarity

berechnet. Die ähnlichsten Relationen werden zurückgeliefert, wobei ein Threshold ein Mindestmaß an Ähnlichkeit erzwingt. Damit ist besonders das Relation Linking weniger anfällig für Probleme mit Synonymen und Formulierungen, die nicht den Relationen im KG entsprechen.

Als Alternative zu SPARQL Templates wurde ein auf Graphisomorphismus basierender Ansatz weiterentwickelt mit dem Ziel, einzelne Elemente der SPARQL Query mittels kleinerer Klassifikatoren vorherzusagen. Die Idee der Nutzung des Graphisomorphismus ist, dass beliebige SPARQL Queries auf grundlegende Schemata gemappt werden können, die Triple Patterns sind gleich. Es wurden bestehenden Benchmark-Datensätze wie qald-9 prozessiert und die vorhandenen, annotierten SPARQL Queries auf ihre Triple Patterns abgebildet. Insgesamt war die Anzahl an Patterns, je nach Datensatz, zwischen 30 und 50. Verschiedene leichtgewichtige Klassifikatoren zur Erkennung von Triple Patterns anhand von natürlichsprachlichen Nutzeranfragen wurden evaluiert, wobei RandomForest Klassifikatoren in verschiedenen Konfigurationen überlegen waren.

Andere Elemente der SPARQL Queries, wie bspw. Aggregationen, Sortierungen, etc. wurden mittels Klassifikatoren auf Basis von Frage-Embeddings klassifiziert. Die einzelnen Bestandteile der Query wurden dann gegen bereits identifizierte Entitäten und Relationen gematcht und entsprechend zusammengeführt.

Die Ergebnisse dieses Ansatz zeigen sich als deutlich flexibler als der Template-basierte Ansatz. Auch Variationen von Fragen aus dem Covid-Datensatz waren deutlich häufiger zu beantworten. Gleichzeitig zeigte sich eine erhöhte Anzahl an inkorrekten Antworten, wobei diese mit Thresholds der SPARQL Query Klassifikatoren reduziert werden konnten.

Abschließend wurden ebenfalls generative Ansätze zur Erzeugung von SPARQL Queries implementiert, speziell S-GPT. Dabei wird ein Transformer-Decoder Language Model mit KGQA Trainingsdaten durch fine-tuning adaptiert, wobei spezifische linguistische Informationen, wie Dependency Trees, POS Tags, Entitäten und Relationen, ebenfalls kodiert werden. Erste Ergebnisse anhand von Benchmark Datensätzen sind vielversprechend. Eine Anwendung des Algorithmus auf Knowledge Graphen von SPEAKER Partnern war zum Ende der Projektlaufzeit nicht mehr möglich.

6.4 Tools und Daten

Das **fünfte Arbeitspaket** hatte zum Ziel, die SPEAKER Plattform im B2B-Bereich domänenübergreifend anwendbar zu machen. Hierzu sollten Datenkonnektoren für die Module bereitgestellt werden, welche ein Andocken an etablierte Datenquellen ermöglichen. Ebenfalls wurde in diesem

Arbeitspaket das Ziel definiert die existierenden Module auf multilinguale Erweiterungen zu prüfen und wo möglich diese vorzunehmen. Es sollten ebenfalls Benchmark-Frameworks für einzelne Module und für die Voice User Experience implementiert sowie dafür notwendige Daten von Verbundpartnern erhoben werden. Hierdurch sollte die hohe Qualität sichergestellt und Module verschiedener Anbieter vergleichbar gemacht werden.

Auch die Ziele dieses Arbeitspaktes wurden vollumfänglich erreicht. Im Zuge der Anforderungsanalyse wurde eine Übersicht an Sprachen erarbeitet, welche umgesetzt werden sollten. Zunächst wurde sich darauf verständigt, dass die Module in Deutsch und Englisch verfügbar sind, der Bedarf an weiteren Sprachen wurde bei den Projektpartnern abgefragt und dokumentiert. Anhand der Anforderung der Partner konnten notwendigen Benchmarking Strategien anhand gebräuchlicher Metriken erarbeitet werden.

Als Benchmarking Tool wurde für die Sprachsynthese WebMUSHRA genutzt, um Hörtest im Webbrowser durchführen zu können. Das Benchmarking wurde laufend durchgeführt, um sicherzustellen, dass sich die Leistungsparameter während der Entwicklung verbessert haben.

Bezüglich der Voice User Experience wurde eine Fehlerkorrekturstrategie ausgearbeitet. Diese dienen als Grundlage für die Implementierung im Dialog Manager und Dialog Editor. Die SPEAKER-Partner wurden zudem hinsichtlich der Dialoggestaltung für die individuellen Use Cases beraten.

Vom Technologie-Partner LMU München wurden für die Dauer des Projektes umfangreiche Sprachkorpora des Bayerischen Archivs für Sprachsignale zur kostenfreien Nutzung zur Verfügung gestellt. Für die Realisierung von individuellen Sprachaufnahmen wurde von Technologie-Partner LMU München die Online-Plattform WikiSpeech zur kostenfreien Nutzung für alle SPEAKER Partner freigeschaltet. Ein Workshop zur Nutzung von WikiSpeech wurde durchgeführt. Mehrere Partner haben WikiSpeech während der Projektlaufzeit für Sprachaufnahmen und -annotationen genutzt, und wurden dabei laufend beraten.

6.5 Evaluation in Anwendungsfällen und Modulentwicklung

Das Ziel des **sechsten Arbeitspaketes** war die Umsetzung erster Pilotanwendungen auf der SPEAKER-Plattform. Folgende Pilotanwendungen sollten hierbei umgesetzt werden: CommaSoft AG: Sprachbasiertes Gesundheitslogbuch, Deutsche Bahn AG: Sprachassistent zur Schadensprotokollierung, Grundig Business Systems GmbH: Intelligentes Diktiersysteme, ONSEI GmbH: Smart Speaker für Hotellerie, SAP SE: Live Transkription von Events, Scopevisio AG: Sprachassistent Unternehmenssoftware, Siemens AG: Machine Assistant App, TU Braunschweig: Audiovisuelle Spracherkennung, Universität Paderborn: Offline Question Answering.

Die Ziele des Arbeitspaketes konnten alle erreicht werden. Zur Umsetzung fanden zahlreiche bilaterale Gespräche statt, es wurden aber auch Workshops angeboten, in welchen wir mit den Partnern das weitere Vorgehen der Technologieentwicklung abstimmten und die Anbindungen in der SPEAKER Plattform definierten.

6.5.1 Comma Soft AG – Sprachbasiertes Gesundheitslogbuch

Die Comma Soft AG hat den Use-Case zur Implementierung eines sprachbasierten Gesundheitslogbuchs erfolgreich abgeschlossen. Die Entwicklung umfasst eine Web- und Smartphone-App mit dialogbasierter Steuerung, ein Backend zur Extraktion der im Gesundheitskontext relevanten Informationen, sowie verschiedenen visuellen Aufbereitungen der extrahierten Daten im zeitlichen Verlauf.

Im Rahmen der Entwicklung wurde laufend das Feedback verschiedener Ärzte eingeholt, um sicherzustellen, dass die App sowohl unter dem Gesichtspunkt der Bedienbarkeit durch die angestrebte Zielgruppe als auch unter dem Gesichtspunkt der Nützlichkeit für die ärztliche Behandlung allen Anforderungen genügt. Insbesondere wurde in diesem Rahmen die visuelle Darstellung der Daten mehrfach überarbeitet und sichergestellt, dass die App über die Möglichkeit verfügt, über verschiedene Konfigurationen an die individuelle gesundheitliche Situation sowie die persönlichen Präferenzen eines Patienten angepasst werden zu können.

Der Workflow des Gesundheitslogbuchs nutzt eine Reihe verschiedener, im Rahmen des Gesamtvorhabens entwickelter Technologien:

1. Die Eingabe des Nutzers erfolgt über natürliche Sprache und wird zur weiteren Verarbeitung mittels der zur Verfügung gestellten ASR transkribiert.
2. Verschiedene NLP-Algorithmen im Backend kommen zum Einsatz, um relevante Informationen wie beispielsweise Gewicht, allgemeines Befinden, Medikamente, Schmerzintensität und Blutdruck zu extrahieren.
3. Zu noch fehlenden Informationen werden dynamisch gezielte Rückfragen in einem Dialogformat gestellt.

In mehreren internen Evaluationszyklen wurde sichergestellt, dass alle Komponenten Ergebnisse in ausreichender Qualität liefern, um den angestrebten Workflow zu ermöglichen.

Neben der eigentlichen Entwicklung haben sich die Mitarbeiter:innen der Comma Soft AG in den verschiedenen Meetingformaten des Verbundvorhabens engagiert und regelmäßig innerhalb der eigenen Firma über das SPEAKER-Vorhaben als Ganzes sowie das Gesundheitslogbuch im Speziellen informiert.

6.5.2 DATEV eG – Sprachassistent im Kundenservice

DATEV hat im Projekt ein breites Spektrum von technischen Sprachassistenten- und KI-Modulen aus dem SPEAKER Projekt begleitet und evaluiert. Im Kontext der DATEV Szenarien wurden die technischen Lösungen erforscht und diese dann geprüft, getestet und prototypisch umgesetzt. Die Zusammenarbeit mit der Fraunhofer Projektleitung und Ansprechpartnern wurde in Form von Statustreffen, Workshops und bilateralen Arbeitsterminen etabliert.

Projektergebnisse:

Über die Projektlaufzeit wurden Evaluationen durchgeführt. So wurden die ASR und TTS Module von Fraunhofer ausgiebig unter anderem mit SSML getestet und erfolgreich mit domänenspezifischen Inhalten trainiert. Mit den Komponenten Dialogmanager, Dialogeditor und NLU wurden zwei fachliche Prototypen erstellt

Die Sentimentanalyse des Verbundpartners Audeering wurde verprobt, und diesen Ansatz für die nächste Entwicklungsstufe im Auge behalten.

Das Open Source Rasa Framework (NLU, ML, QA, Dialog- / Actionmanager) Adaption wurde validiert und auf Integrationsmöglichkeiten geprüft. Ab Q2/2022 wurden alternative marktgängige Anbieter für Sprachassistentenmodule evaluiert dazu wurden alternative kommerzielle Plattformen und Services (Microsoft, Parloa, IBM, Cognigy) analysiert, gebenchmarkt und bewertet. Als Ergebnis davon haben wir uns für die weitere Umsetzung der Prototypen für Cognigy einschließlich der Microsoft Speech-Services und dem MS Open AI Sprachmodell entschieden.

Demonstratoren

Bei der Erstellung der Prototypen wurden -neben dem eigentlichen Demonstrator- wichtige Zwischenergebnisse erzielt. So konnten bei der Konzeptionierung der Anwendungsfälle die Sprachassistenten-Szenarien für DATEV geschärft werden, die zu einer erfolgreichen Bewertung der Einsatzfelder/ Use Cases führten.

Bei der Vorbereitung der Demonstratoren wurden die Anwendungsfälle analysiert und fachlich definiert und es wurden die vorhandenen Daten für die prototypischen Umsetzungen modelliert und sowohl technisch als auch konform zu den Anforderungen der DSGVO nutzbar gemacht.

Während der Projektlaufzeit wurden drei Prototypen zu Sprachassistenten im Service und für den Kunden implementiert, in denen entsprechende Dialoge und Workflows konzipiert und orches-

triert wurden. Diese wurden durch Durchführung von Kundeneinbezugsmaßnahmen DATEV-intern und auch -extern validiert. Dazu wurden quantitative und qualitative Studien auf Basis der Demonstratoren durch Fokusgruppen und Befragungen durchgeführt.



Abb. 30: Anwendungsfall Sprachassistent im Service

Die strategische Ausrichtung der DATEV eG wurde gestärkt. Dies erfolgte im Zusammenhang der Sprachassistent, die Wirkung ging jedoch erfreulicherweise auch über diesen Kontext hinaus.

So fand eine feste Verankerung Sprachassistent in interner Unternehmenskommunikation statt, unter anderem in Blogs, unserem internen Social-Dienst Viva Engage und in Kunden- und Mitarbeiterpräsentationen u.a. in Form von Projekt-Videos. Wichtige Erkenntnisse im Bereich KI und Sprachassistent wurden gewonnen und in die DATEV transportiert. Ebenso wurden zahlreiche Fragen im Bereich Datenschutz, Recht und IT-Security zur Nutzung von Sprachassistent mit sensiblen Daten erfolgreich geklärt.

Es wurde ein dauerhaftes Sprachassistent-Team zum Ausbau der technologischen Fähigkeiten etabliert, damit verbunden ist die Verankerung von Sprachassistent-Systemen in den Unternehmensprozessen. Den über Fördermittel finanzierten Projektressourcen konnten dauerhaft an das Unternehmen gebunden werden. Durch die frühzeitige Evaluierung von LLMs im Umfeld dynamischer QA-Dialoge und Conversational AI konnte DATEV zeitnah den Trend aufnehmen und aus technologischer und geschäftspolitischer Sicht zeitnah bewerten und Maßnahmen ableiten.

Das Ziel der prototypischen Errichtung einer DSGVO-kompatiblen Public Cloud Plattform zum Projektende konnte aus DATEV Sicht nicht erreicht werden. Im Projektverlauf stellte sich heraus, dass die Herausforderungen beispielsweise zu Datenschutz/-sicherheit und Betriebskonzept absehbar nicht gelöst werden können, da sie nicht die Anforderungen unserer Kunden abbilden können. Daher haben wir uns für einen domänenspezifischen Ansatz entschieden.

6.5.3 Deutsche Bahn AG – Sprachassistenten in der Instandhaltung

Die Deutsche Bahn AG hat mit Unterstützung des internen Technologiepartners DB Systel sowie dem Anwendungspartner DB Fahrzeuginstandhaltung einen MVP in Form einer App für mobile Endgeräte erstellt. Diese war die technische Umsetzung des Use Case "Dokumentation von technischen Fahrzeuguntersuchungen".

Hierzu wurden aus Perspektive der User und der Prozessrichtlinien funktionale und nicht-funktionale Anforderungen in Form von User Stories formuliert. Hierbei war es erforderlich, einerseits die nutzerspezifischen, andererseits die prozess- bzw. qualitätsrelevanten Anforderungen abzudecken. Das umfasste folgende Punkte:

- die regelbasierte, fachlich richtige Dokumentation von technischen Zuständen am Fahrzeug.
- die freie Navigation für erfahrenes Fachpersonal sowie Reaktionsmöglichkeiten in Störfällen
- prozessnahe Benutzerführung zur Standardisierung
- Balance zwischen vorgegebenem Fachvokabular und Synonymen
- Antwortverhalten des Sprachassistenten (Spannweite von Wiederholung des Gesagten bis hin zu kurzen Earcons)

Die Entwicklung umfasste das technische Grundgerüst einer Android-kompatiblen App, sowohl die mündliche, dialogbasierte Bedienung (VUI) als auch die notwendigen visuellen Komponenten zur nicht-sprachbasierten Benutzerführung. Insbesondere zu Qualitäts- und Kontrollzwecken ist das GUI wichtig und schafft eine höhere Nutzerakzeptanz. Eine weitere Größe war die Aufbereitung der notwendigen Stammdaten zum Training der Sprachmodelle und Plausibilisierung der Nutzereingaben für die Sicherstellung der Prozessqualität.

Neben rein dialogbasierten Herausforderungen wurden vor allem auch Einflüsse aus dem industriellen Umfeld des Nutzers studiert

- Umgebung mit hoher Lärmintensität
- Störquellen bei der Konnektivität

- Zusammenarbeit mit Sprachassistenten und menschlichen Kollegen

Die technischen Komponenten beinhalteten den gesamten Tech-Stack eines Sprachassistenten (AFE, ASR, DM, KG, KWS, TTS). Hierbei haben wir unter Berücksichtigung des Entwicklungsfortschritts vonseiten Fraunhofer teilweise den Fraunhofer Tech-Stack oder dritte Technologie (u.a. Microsoft, IBM, Open Source) verwendet. Dies ermöglichte uns auf der einen Seite unabhängig von Fraunhofer eine hohe Umsetzungsgeschwindigkeit zum schnellen Test des Use Case unabhängig von spezifischen Lieferanten. Auf der anderen Seite konnten wir so Benchmarks zwischen den Komponenten von Fraunhofer und Dritten durchführen.

Im Ergebnis wurde ein technisches Grundgerüst entwickelt, das eine hinreichende Robustheit und Anwendungsfähigkeit zur Bewertung des Use Case darstellte. Die technische und betriebliche Einsetzbarkeit wurde in variierenden Nutzertests und Befragungen unter Beweis gestellt. Die Komponenten des Fraunhofer Tech-Stack wurden teilweise isoliert, teilweise in der App integriert getestet, wobei der Reifegrad für einen fortlaufenden Produktiveinsatz zum Abschluss des Projekts noch nicht gegeben war. Seitens der DB AG wird daher ein modularer Aufbau und gemischter Betrieb von Fraunhofer und dritten Komponenten empfohlen. Hinsichtlich der finalen Entwicklung des MVP zu einem Produktivsystem war der Aufwand höher als der wirtschaftliche Nutzen, sodass eine unmittelbare Fortführung der Entwicklung dieses spezifischen Use Case nach Abschluss des SPEAKER Projekts Stand heute nicht angestrebt wird.

Durch den nachgewiesenen Nutzen in der Domäne der Fahrzeuginstandhaltung wird angesichts der schnellen technischen Weiterentwicklung von LLM die Kombination von Sprachassistenten und Augmented Reality als zukunftssträftig gesehen. Hier sehen wir eine einfache Aufbereitung, Abrufbarkeit und Durchsuchbarkeit von Informationen perspektivisch als nächsten Schritt, um den Mehrwert und das Potenzial von Sprachassistenten zu steigern.

6.5.4 SAP SE – Transkription von Live Events

Simultane, automatische Transkription und Textanzeige während Veranstaltungen: SAPs in-house Lösung für virtuelle Events wurde prototypisch an die SPEAKER ASR angebunden, um die Live-Verschriftlichung zu ermöglichen. Dazu wurde die Lösung um generische Schnittstellen für Spracherkennung ergänzt. Der Live-Betrieb wurde mit mehreren SprecherInnen sowohl für Deutsch als auch für Englisch zum einen mit Spontan- und zum anderen mit Schriftsprache getestet. Die dabei aufgezeichneten Daten wurden genutzt, um die Erkennungsgenauigkeit der Spracherkennung zu bewerten.

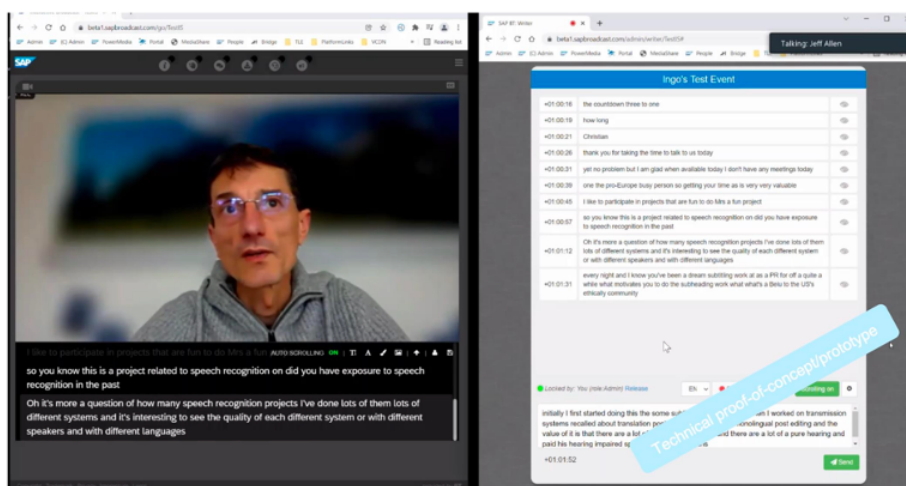


Abb. 31: Prototypische Anwendung zur Transkription von virtuellen Events

1.

6.5.5 SAP SE – Spracherkennung für Botbau Plattform

Komplementäre Spracherkennung im Kontext einer Plattform für den Bau von Bots: Das Produkt SAP Conversational AI wurde um eine Schnittstelle für Spracherkennung ergänzt. Mit dem Produkt wurde eine Reihe von prototypischen FAQ chatbots (jeweils Deutsch und Englisch) mit SAP-Bezug erstellt (unter anderem basierend auf der SAP Terminologiedatenbank SAPterm).

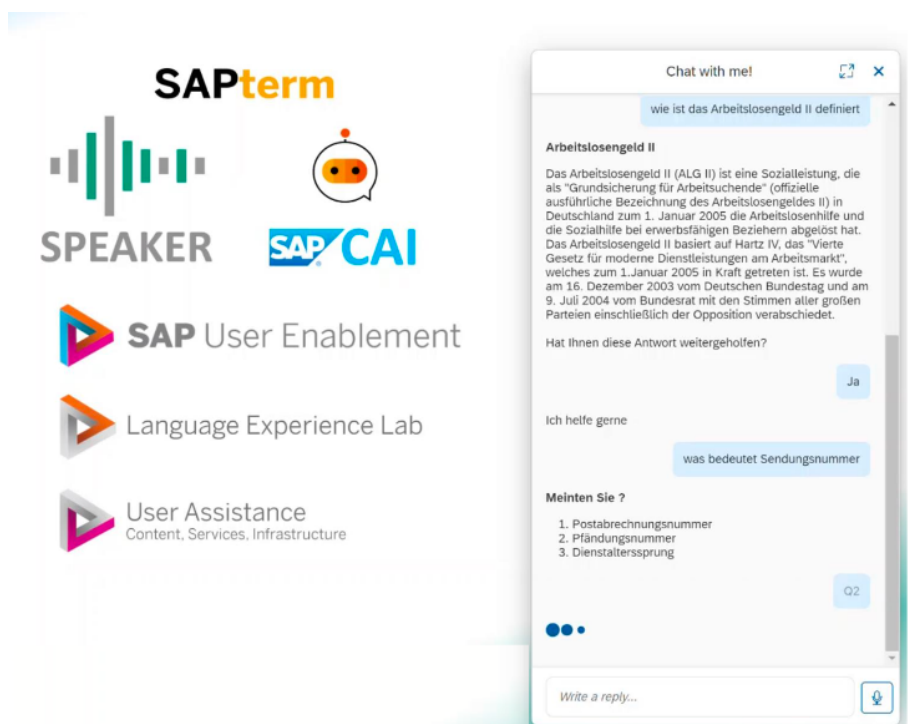


Abb. 32: SPEAKER Technologien als mögliches Sprachinterface für Chatbots

Zur Umsetzung der Anwendungsfälle hat die SAP – unter Nutzung unter anderem von Technologien des Konsortialpartners BAS der LMU – Daten für die Evaluierung von Spracherkennung und Sprachsynthese erstellt. Unter anderem wurden mit einer Vielzahl von SprecherInnen mehrere Hundert Sprachaufnahmen erstellt und qualitätsgesichert. Dabei wurden Erkenntnisse rund um den Datenschutz vertieft, sowie im gegebenen Rahmen Evaluierungen (automatisch) und Beurteilungen (durch Menschen) durchgeführt.

Während des Projekts hat sich die SAP stark im bilateralen und multilateralen Austausch mit den Konsortialpartnern engagiert. Außerdem haben die SAP-Beteiligten durch intensive interne (u.a. rund 20 interne blog posts, mehrere screencasts und aufgezeichnete Informationssession) als auch ausgewählte externe Kommunikation auf das SPEAKER-Projekt hingewiesen. Außerdem hat sich die SAP sowohl im wissenschaftlichen als auch im industriellen Beirat von SPEAKER engagiert.

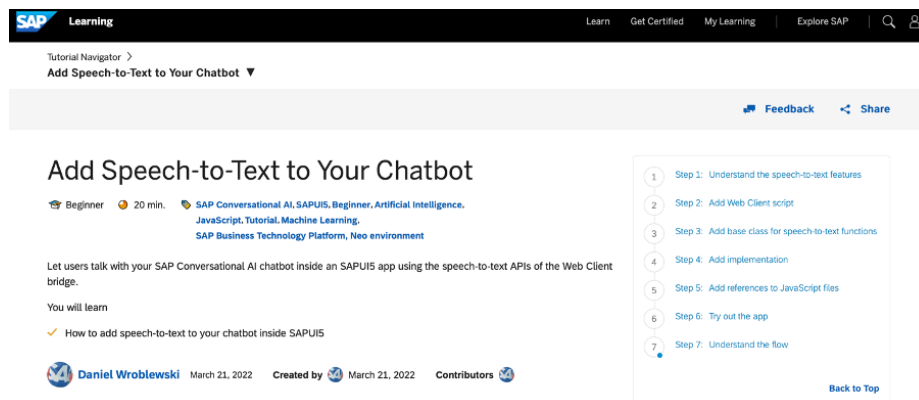


Abb. 33: Tutorial zur Anbindung eines Sprachinterfaces an Chatbots

6.5.6 IAV – Sprachassistent für die Fahrzeugentwicklung

Auf IAV-Seite konnte die Entwicklung einer Unterstützung von Testfahrer:innen durch einen applikationsnahen Sprachassistenten erfolgreich umgesetzt werden. Hierfür wurde das Tool SCD (speech control demonstrator) entwickelt, welches als Stand-Alone Software die Messtools INCA und CANape mit einer Sprachsteuerung nachrüstet. Das gesamte Tool funktioniert ohne Internetanbindung und läuft lokal auf dem Messrechner (siehe Abbildung 34).

SCD besitzt zahlreiche Einstellungsmöglichkeiten und kann sowohl für das Trigger Word als auch für die Automatische Spracherkennung (Automatic Speech Recognition, ASR) verschiedene Softwarelösungen einbinden.

Für das Trigger Word wurden der Fraunhofer Keyword Spotter implementiert und zusätzlich das Open-Source Projekt OpenWakeWord und das Porcupine Wake Word von PicoVoice als Vergleichsoption integriert.

Für die ASR kam der Fraunhofer Docker-Container zum Einsatz und optional die Projekte Whisper (in den Varianten tiny, medium, large) von OpenAI, Vosk von Alpha Cephei und Conformer-1 von AssemblyAI.

Ein typischer Use-Case ist die Testfahrt in einem Auto, die mit der Nennung des Trigger Words und dem Text „Starte die Messung“ begonnen wird. Es funktionieren weitere Befehle wie „Stoppe die Messung“ oder „Wechsel auf Referenzseite“. Texte wie „Der Motor ruckelt“ werden als Notiz erkannt und mit Zeitstempel in der Messsoftware vermerkt. Dadurch können während der gesamten Testfahrt beide Hände am Lenkrad verbleiben.

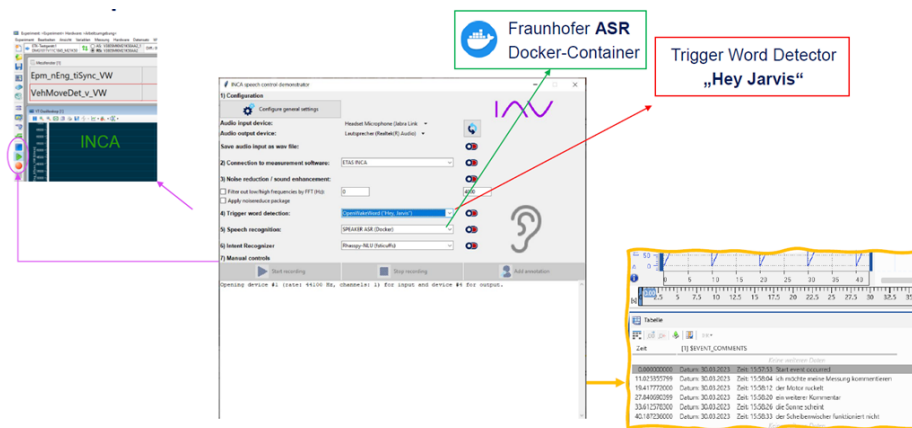


Abb. 34: Das Tool SCD (speech control demonstrator), welches als Stand-Alone Software die Messtools INCA und CANape mit einer Sprachsteuerung nachrüstet

6.5.7 ONSEI – Smart Speaker in Hotels

Zielstellung: Ziel des Projekts war es, einen funktionsfähigen Smart Speaker für den Einsatz in Hotelzimmern zu entwickeln, der mit Infotainment Funktionen das Gasterlebnis bereichert und gleichzeitig das Hotelpersonal entlastet.

Umsetzung: Als Hotelpartner konnte ONSEI die Deutsche Hospitality gewinnen, der Hotelmarken wie Steigenberger, Intercity Hotel, MAXX oder Jaz in the City angehören. In enger Abstimmung mit dem Hotelpartner wurden die Funktionalitäten des Smart Speakers definiert, die UX ausgearbeitet und die technischen Gegebenheiten des Testhotels analysiert, um den Use-Case optimal auf die Anforderungen abzustimmen. Nach intensiver Recherche und Kontaktaufnahme

mit verschiedenen Anbietern konnte ONSEI einen passenden Smart Speaker eines chinesischen Herstellers identifizieren, der sämtlichen Projektanforderungen entsprach. Zur Umsetzung wurden unter anderen die im Rahmen des SPEAKER Projekts entwickelten Module ASR, TTS und KWS sowie eine VQE/AFE Lösung von Fraunhofer genutzt. Um das Risiko von Fehlaktivierungen durch Hintergrundgeräusche, wie beispielsweise die Wiedergabe von Audio aus dem Fernseher zu minimieren, wurden gezielte Anpassungen des Keyword-Spotters vorgenommen. Außerdem wurden zusätzliche Filterfunktionen implementiert, die diese Fehlaktivierungen erfolgreich minimierten.

Testphase: Die Erprobung des Hotel Smart Speakers erfolgte im Jaz in the City Hotel Vienna vom 13.01.2023 bis zum 25.09.2023. Im Rahmen des Rollouts vor Ort wurden 38 Hotelzimmer unterschiedlicher Kategorien mit jeweils einem Smart Speaker und begleitenden Informationsflyern ausgestattet. Parallel dazu fand ein Workshop für das Hotelpersonal statt, in dem die Nutzungsoptionen der Geräte sowie das Handling vermittelt wurden. Zusätzlich erfolgte die Übergabe von Ersatzgeräten, einem Trouble-Shooting-Laptop und einem Tablet zur Nutzung der Admin-App.



Abb. 35: Fotografie des Smart Speakers Melody bei der Pilotierung in einem Hotelzimmer



Abb. 36: Flyer zum Smart Speaker Melody

Während der Testphase wurde eine umfassende Betreuung durchgeführt, einschließlich regelmäßiger Abstimmungen mit den Verantwortlichen auf Hotelseite.

Analyse und Evaluierung

Eine detaillierte Analyse zur Wake-Word-Erkennung und Nutzung der Funktionalitäten wurde durchgeführt. Auf Basis der Erkenntnisse wurden Logging und Serverstabilität weiter optimiert. Die Auswertung der Nutzung der Smart Speaker basierte auf den Belegungszahlen der Testzimmer sowie den Logdaten. Die Nutzeraktivität zeigte erfreulicherweise, dass der Smart Speaker in bis zu 89% der belegten Testzimmer, die mindestens einmal täglich genutzt wurden.

Das Feedback der Hotelgäste, erhoben durch eine Online-Umfrage, wurde intensiv ausgewertet und floss in die abschließende Beurteilung ein.

Fazit: Das Projektteam nahm aktiv an verschiedenen Workshops, monatlichen Meetings, Fokus- und Meilensteintreffen des SPEAKER Konsortiums teil, um eine kontinuierliche Optimierung sicherzustellen. Die Teilnahme und Vorstellung des Projekts bei der SPEAKER Abschlussveranstaltung rundete das Projekt erfolgreich ab.

Insgesamt konnte das Projektziel, die erfolgreiche Implementierung von Sprachassistenten im Hotelumfeld, erreicht werden. Die gewonnenen Erkenntnisse und Erfahrungen bilden eine solide Grundlage für zukünftige Entwicklungen im Bereich der Sprachtechnologie in der Hotellerie.

6.5.8 Retresco – Natürlichsprachliche Textgenerierung

Ziele des Moduls Natürlichsprachliche Textgenerierung (NLG): Durch Dialoge und Suchanfragen auf der SPEAKER Plattform können eine Vielzahl von Datenpunkte gewonnen oder generiert werden. Um daraus einen dialogbasierten Austausch mit den Nutzenden zu ermöglichen, benötigt es eine Komponente zur natürlichsprachlichen Textgenerierung, welche Use Case-spezifisch und Intent-basiert Antworten verbalisiert. Damit wird das Spektrum, der auf der SPEAKER Plattform abbildbarer Use Case-Szenarien, erheblich erweitert.

Ergebnisse: Erweiterungen der SaaS NLG Plattform textengine.io: Neben der Anbindung auf Schnittstellenebene (Authentifizierung, Implementierung Datenmodell, API-Requests und Error-Handling) wurde die bestehende SaaS-Anwendung textengine.io um notwendige Features erweitert, um die universelle Einsetzbarkeit einer Textgenerierung im Voice-Kontext zur Verbalisierung komplexer Aussagen aus Daten zu ermöglichen:

- Unterstützung von verschachtelten json-Datenstrukturen
- Definition von Bedingungen für Message-Elemente zur intent-bezogenen Textgenerierung im Dialog Management
- Wiederholung von Message-Elementen zum Ausdrücken unterschiedlicher Textvarianten für einzelne Listenwerte in verschachtelten Datensätzen (z.B. bei Auflistungen)
- Gruppieren von Message-Elementen und wiederholen ganzer Gruppierungen (z.B. zur Abbildung komplexer Vorgänge und entsprechender Datenkonstellationen)
- Optimierung Sprachfluss im Voice-Kontext: Generierung von Pronomen statt Wiederholung von Nomen (Referenzketten)

Hybrid NLG: Kombination regelbasierter Textgenerierung mit generativer KI (GPT): Im Gegensatz zum ausschließlichen und unmittelbaren Einsatz generativer KI zur natürlichsprachlichen Textgenerierung (GPT), verfolgt Retresco dabei einen hybriden Ansatz, der die Vorteile einer regelbasierten Ausspielung von Texten mit dem kreativen Potential und dem Weltwissen großer Sprachmodelle (Large Language Models) verbindet.

Gerade mit Blick auf die oft sensiblen Daten in B2B Anwendungen und der Bedingung generierte Antworten mit fachlich und sachlich unkorrekten Aussagen (sog. Halluzinationen) auszuschließen, setzt Retresco auf einen Ansatz, der auf einmalig einzurichtende Textmodellen basiert. Dieses Prinzip hat den großen Vorteil, dass User nach der initialen Einrichtung eine hundertprozentige Automatisierung erzielen können. Alle Texte, die entstehen, sind grammatikalisch geprüft und nachweislich fehlerfrei. Diese Anforderungen konnten durch den Austausch mit Anwendungspartnern aus dem SPEAKER Konsortium bestätigt werden. Durch den gewählten Ansatz lassen sich rechtlich überprüfbare und verifizierbare Aussagen generieren, was gerade bei einer Einbindung in einen Sprachassistenten mit einer potenziell enormen Anzahl möglicher Dialogverläufe und Textkonstellationen, die nicht in ihrer Gänze durch eine manuelle Qualitätsanalyse aller generierten Texte abgedeckt werden können, für viele Anwendungsfälle unerlässlich ist.

Die traditionell aufwendige Erstellung der hierfür notwendigen regelbasierten Templates konnte allerdings während der Projektlaufzeit durch die Integration großer Sprachenmodelle (GPT) erheblich beschleunigt werden. Textengine.io ist damit zu einer hybriden Textgenerierungssoftware geworden, die alle relevanten Anwendungsfälle und Text-Typen unterstützt und die Vorteile regelbasierter und generativer KI verbindet. Anwender profitieren dabei von einer gestiegenen Flexibilität und einem schnellen Start ihrer Projekte auf der SPEAKER Plattform. Ganz im Sinne des Datenschutzaspekts der SPEAKER Plattform, reichen hierfür beispielhafte Datenpunkte aus, so dass die eigentlichen Datenpunkte für die Textgenerierung nicht an GPTs und somit auch nicht an Services von Dritten gesendet werden müssen.

Die so entstandenen Textgenerierungsmodelle bleiben geistiges Eigentum des Nutzers, und die bereitgestellten Trainings- und Generierungsdaten werden strikt je Nutzer bzw. Mandant getrennt, um eine große Bandbreite (auch sensibler) Use Cases zu ermöglichen.

Use Cases und Demonstration: Im Rahmen der Modulentwicklung wurden unterschiedliche Use Cases konzipiert (z.B. für Hotellerie, Service und Sportberichte) und die Anbindung der textengine.io mit der SPEAKER Plattform exemplarisch aufgezeigt und demonstriert. Dabei konnte erfolgreich gezeigt werden, wie eine einfach und nutzerfreundliche Integration natürlichsprachlichen Textgenerierung in das Dialog-Design der SPEAKER Plattform in der Praxis umgesetzt werden kann. Relevante Punkte waren dabei:

- Datenbankabfrage (Rückgabe Datenobjekt)
- Anfrage an textengine.io (mit Datenobjekt)
- Verarbeitung der Antwort (generierter Text)
- Text-to-Speech Ausgabe

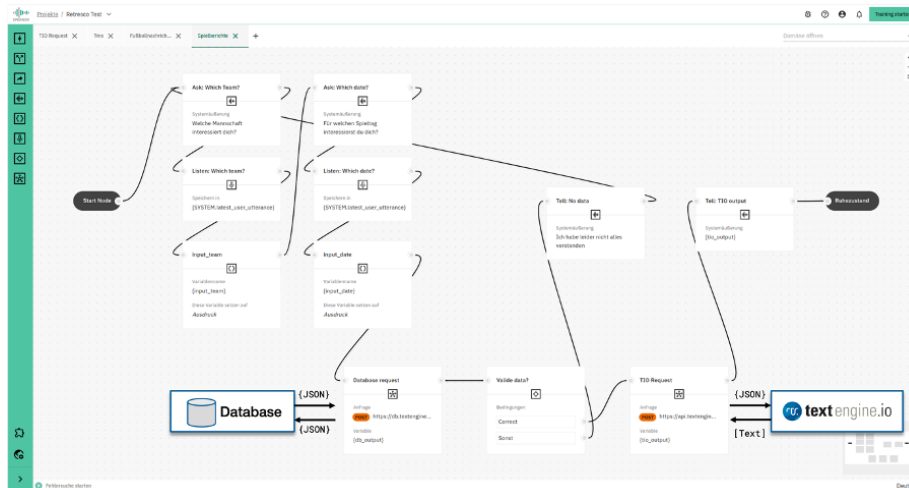


Abb. 37: Schematische Darstellung der Anbindung von textengine.io mit Hilfe des Dialog Editors

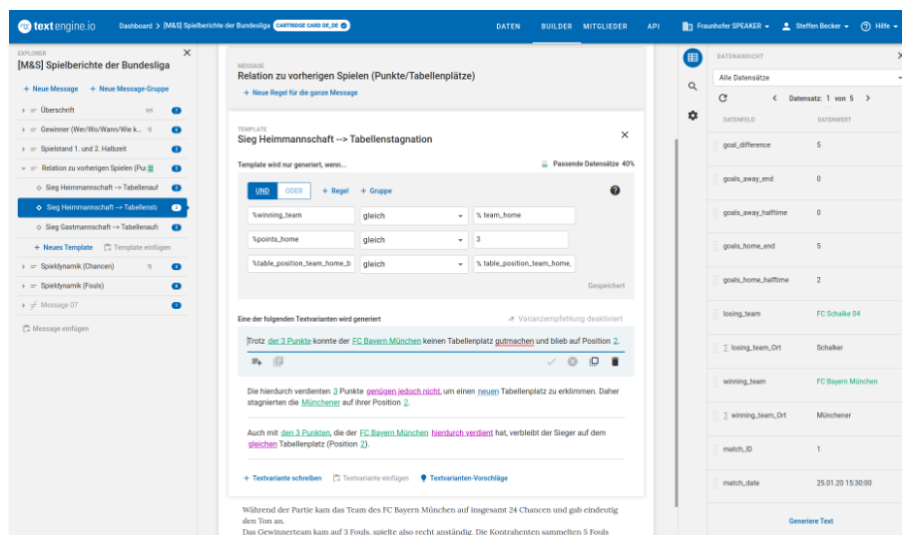


Abb. 38: Web Frontend zur Konfiguration der textengine.io

6.5.9 Siemens – Verschiedene Szenarien im industriellen Umfeld

Die Siemens AG hat verschiedene Szenarien für den Einsatz der SPEAKER-Plattform im industriellen Umfeld erstellt und daraus Anforderungen abgeleitet. Dabei wurden insbesondere Aspekte

der robusten Spracherkennung im industriellen Umfeld und die Integration von Domänenwissen betrachtet und in konkreten Anwendungen erprobt.

Es wurden drei use cases detailliert untersucht:

- Industrial Edge (Simatic)
- SCADA
- Servicetechniker Berichterstellung

Dazu gehörte auch die differenzierte Betrachtung der Vor- und Nachteile von Cloudlösungen gegenüber On-Premises Lösungen. Desweiteren ist der Themenbereich User Experience ausführlich behandelt worden, da nur die nutzergerechte Gestaltung von Sprachassistenten zu Akzeptanz und Effizienzsteigerungen führen kann. Hierfür wurden entsprechende Vorgehensmodelle zur Gestaltung multimodaler Interaktion vorgestellt.

Eine Einschätzung und Bewertung der Use Cases hinsichtlich der Erwartung eines geschäftlichen Erfolgs durch Sprachassistenten wurde gemeinsam mit den betroffenen Siemens Geschäftsbereichen durchgeführt. Als Ergebnis wurden insbesondere drei Requirements herausgestellt:

1. Benchmark muss im Wesentlichen gleichauf oder besser sein als vergleichbare Angebote von Microsoft, Google, Amazon, etc.
2. Multi-Lingualität mit mindestens 5 Kernsprachen und der Option auf weitere
3. Kostentransparenz zu Cloud-Hosting und einer on-premise Installation

Da die Erwartung nicht erfüllt werden konnte, dass ein Benchmarking von ASR und NLU im Laufe des ersten Projektjahres von anderen Konsortialpartnern übernommen werden kann, wurde mehr Arbeit als geplant auf dem Thema Benchmarking geleistet, da die involvierten Geschäftsbereiche darin einen besonderen Bedarf signalisiert hatten.

Daher wurde die Durchführung und Auswertung eines Benchmarkings der ASR-Komponente mit Hilfe von 8 Probanden vorgenommen, wobei die Aufnahme typischer Mensch-Maschine Dialogfragestellungen aus der Simatic Edge Umgebung als Testsatz definiert wurde. Die Ergebnisse zu diesem Thema mündeten in einer Veröffentlichung mit Konferenzvortrag:

Speech Assistance: Use Cases in Industry. 28th IEEE ICE/ITMC & 31st IAMOT Conference on Technology, Engineering, and Innovation Management Communities as Enablers for Social-Ecological Transitions, Nancy 2022

Im Ergebnis zeigte sich, dass die erste Anforderung der Geschäftsbereiche hinsichtlich des Benchmarkings für die Sprachen Deutsch und Englisch erfüllt werden können. Jedoch waren die beiden Anforderungen zu Multi-Lingualität und der Verfügbarkeit eines Cloud-Services nicht erfüllt.

Utterances related to SAM – German and English

German					English				
	FhG		other ASR			FhG		other ASR	
	LSDString	LSDSender	LSDString	LSDSender		LSDString	LSDSender	LSDString	LSDSender
	SAM-DE	SAM-DE	SAM-DE	SAM-DE		SAM-EN	SAM-EN	SAM-EN	SAM-EN
Speak 1	0,954	0,833	0,974	0,832	Speak 1	0,897	0,788	0,928	0,813
Speak 2	0,962	0,836	0,976	0,826	Speak 2	0,826	0,633	0,851	0,733
Speak 3	0,928	0,792	0,763	0,604	Speak 3	0,835	0,713	0,697	0,554
Speak 4	0,963	0,806	0,872	0,702	Speak 4	0,844	0,720	0,794	0,664
Speak 5	0,964	0,843	0,863	0,701	Speak 5	0,832	0,704	0,790	0,658
Speak 6	0,962	0,835	0,958	0,808	Speak 6	0,860	0,728	0,893	0,791
Speak 7	0,921	0,769	0,670	0,491	Speak 7	0,850	0,722	0,675	0,569
Speak 8	0,946	0,817	0,851	0,685	Speak 8	0,800	0,666	0,745	0,621
	0,950	0,816	0,866	0,706		0,843	0,709	0,797	0,675
Wieviele Fehlerteile wurden gestern hergestellt?					What is the speed of conveyer belt DOX074?				
Wie hoch ist die mechanische Belastung in Achse eins?					Please open the dispatching machine terminal screen!				

Abb. 39: Ergebnisse des Benchmarkings mit Utterances aus der Domäne „Digital Factory“

Zu den ersten beiden Use Cases aus „Digital Industries“ gehört auch eine Audio-Demo mit einem Simatic Dialog unter Verwendung von Fraunhofer ASR und TTS-Komponenten, die bei einem Meilensteintreffen präsentiert wurde.

Im weiteren Projektverlauf lag der Fokus auf dem dritten Use Case „Servicetechniker“. Zum Meilensteintreffen wurde eine Demo mit einem Service Maintenance Dialog unter Verwendung von Fraunhofer ASR und TTS-Komponenten präsentiert. Ein GUI unterstützt multimedial den Sprachdialog und überträgt nach Abschluss des Service-Reports die semi-strukturierten Daten auf eine wiki-Seite der Siemens Wikisphere. Die folgende Abbildung zeigt das Konzept mit den dazugehörigen Komponenten und Datenflüssen.

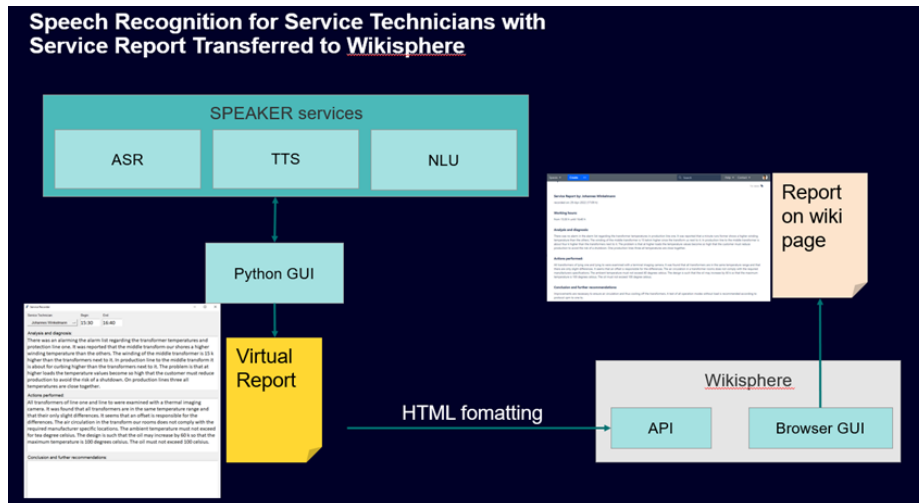


Abb. 40: Demo-Szenario Servicetechniker

Ein weiterer Demonstrator wurde zum Thema Question & Answering (Q&A) erstellt. Der Import von Inhalten auf FAQ Seiten des Enterprise Wikis für Q&A Dialoge wurde so ausgebaut, dass die Inhalte der FAQ Seiten für einen Q&A Sprachdialog mit SPEAKER Komponenten verwendet werden konnten. Hierbei wurden zwei Varianten implementiert. Die erste Variante verwendet einen Ähnlichkeitsalgorithmus (Levenshtein Distanz), um die von der ASR erkannte Frage mit einer Fragestellung der FAQ in Verbindung zu bringen. Die zweite Variante verwendet die NLU-Komponente der Speaker Tools. Es zeigte sich, dass die Verbesserungen, die sich aus der NLU-Komponente ergaben, zu einem relativen hohen Trainingsaufwand erkauft werden mussten.

Während der Kosten-neutralen Verlängerung des Projekts wurde ein weiteres Szenario implementiert, in dem die NLU-Komponente durch „retrieval augmented generation“ (RAG) ersetzt wurde. Hierbei wird zunächst mit einer Suche (Keyword-based oder Vector-embedding-based) eine kleine Menge an Dokumenten gefunden, deren Inhalt dann als „few shot injection“ gemeinsam mit einem prompt an ChatGPT übergeben wird. Aus diesem Input generiert ChatGPT die Antwort auf die gestellte Frage. Dieses Szenario soll für ein generelles wikiGPT mit einem out-of-the-box Confluence Macro weiterentwickelt und verwertet werden.

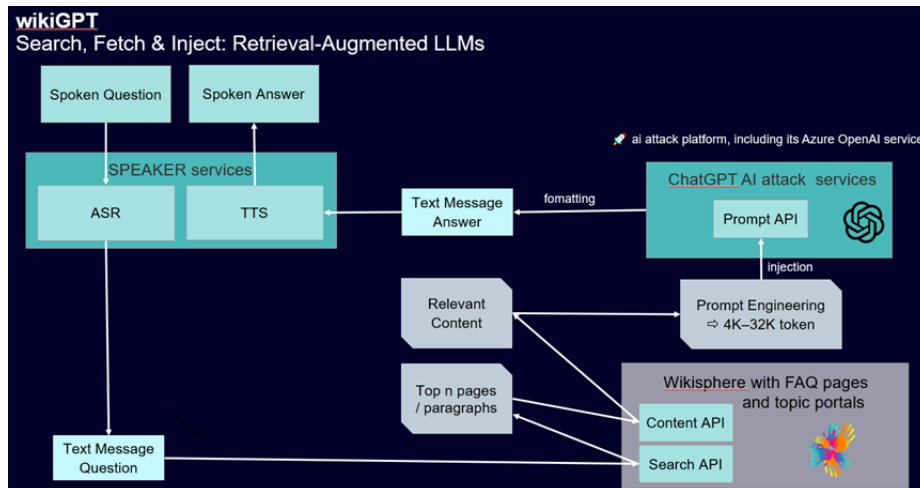


Abb. 41: Demo- Szenario wikiGPT

Die dokumentierten und im Fraunhofer Projekt-Wiki abgelegten Dokumente umfassen:

- M2 Bericht zu den Szenarien und Anforderungen der industriellen Anwendungen
- Konsortial-interner Benchmarking Bericht
- IEEE conference paper on ASR benchmarking
- Audio-Demo SCADA dialog
- Video-Demo Service Report als wiki Dokument
- Video-Demo Question & Answering mit der wiki knowledge base (FAQ)
- Bonus-Demo Q&A unter Verwendung von GPT als NLU für ein RAG Szenario.

Parallel dazu wurden die Ergebnisse des Förderprojekts innerhalb der Siemens AG über das Unternehmenswiki „wikisphere“ publiziert. Es ist eine wichtige Komponente zur Kommunikation und dient als Grundlage zur Erreichung der im Verwertungsplan formulierten Ziele.

In der nachstehenden *Tabelle 5* gibt eine Übersicht über die Beiträge einzelner Anwendungspartner.

Tabelle 5: Pilotanwendungen ausgewählter Projektpartner

Anwendungspartner	Umgesetzte Anwendung
Comma Soft AG	Entwicklung und Feinschliff des grundlegenden App-Konzepts

Anwendungspartner (Sprachbasiertes Gesundheitslogbuch)	Umgesetzte Anwendung
	• Implementierung, Verbesserung und Verfeinerung verschiedener benötigter Algorithmen für Datenanalyse, Nutzerführung und weitere zentrale App-Bestandteile • Implementierung einer allgemeinen Erkennung/Identifikation von Befindlichkeiten (bspw. Weh-tun, jucken, schmerzen), Körperteilen (Bein, Kopf, etc.) und von Medikamenten (Ibuprofen, Aspirin, etc.) • Entwicklung und dynamische Einbindung von Algorithmen zur Datenerkennung und Extraktion. Insbesondere können numerische Ausprägungen zu den o.g. Merkmalen erhoben werden (Beispiel: Schmerzen werden auf einer Skala zwischen 0 und 10 eingeordnet). • Entwicklung eines NLP-Komponente zur Erkennung von Qualifikationen von Scherzen. D.h. Ausdrücke wie z.B. „Keine Schmerzen [mehr]“ können korrekt erkannt und interpretiert werden. • Fortlaufende Weiterentwicklung und iterative Verbesserung der App • Weiterentwicklung und Skalierung der Testumgebung. • Vorbereitung der Datenhaltung/Speicherung für Nutzerstudien. • Ausarbeitung einer (zunächst für Nutzerstudien) geeigneten Datenschutzerklärung. • Prototypische Einbettung/Aufnahme in den Android App-Store (Google Play). Über diese Plattform kann die App bezogen werden, werden Nutzer verwaltet und können neue Versionen der App ausgerollt werden. • Die Demo der App sowie das App-Konzept wurden verschiedenen interessierten Stakeholdern präsentiert. Dazu gehören u.a. Ärzte, Krankenkassen und Pflegeeinrichtungen. Dieses Feedback ist, über die rein inhaltliche Weiterentwicklung, auch für eine spätere wirtschaftliche Weiterverwendung relevant.
DATEV eG	DSGVO-konforme Konzeptionierung und Aufbereitung der zugrundeliegenden Datenquellen finalisiert Domänenspezifisches Vokabular und Terminologie sowie Entities und Intents definiert Für die Use Cases erforderliche Strukturen konzipiert und im Dialogeditor abgebildet. Dynamisiertes Dialog Management in Arbeit und multimodales Interaktionskonzept vorhanden Konzeption von User Experience und Ergonomie abgeschlossen. Es wurde mit Kunden evaluiert, wie dieses es einschätzen, welche fachlichen Fragen von einem Sprachassistenten beantwortet werden können. Es wurden evaluiert, wie die Verständlichkeit von Äußerungen eines Sprachassistenten optimiert werden können. Eine Fokusgruppe zur Beurteilung eines ersten Prototypen (von Fraunhofer) im Umfeld des telefonischen Service wurde durchgeführt mit zahlreichen wertvollen Hinweisen zur Optimierung und Erweiterung des Prototypen. Abschließend wurden ein quantitativer und ein qualitativer Usabilitytest mit Kunden durchgeführt, um auf der Basis dieser Ergebnisse das Dialogdesign einem Feinschliff zu unterziehen. Die Erstellung der Dialoge und die Implementierung der ersten Prototypen ist erfolgt auf Basis der von den Technologiepartnern bereitgestellten Module (z.B. ASR, TTS, Dialogeditor). Training der ASR und TTS mit domänenspezifischen Begriffen durchgeführt und erfolgreich evaluiert.

Anwendungspartner	Umgesetzte Anwendung
Deutsche Bahn AG	• Definition der funktionalen und nicht-funktionalen Anforderungen in Bezug auf den identifizierten Use-Case (sprachbasierte Dokumentation der Fahrzeugbegutachtung in der Instandhaltung), u.a. durch Experteninterviews • Analyse von Fraunhofer-Tools und wettbewerbsrelevanten Anbietern zum Benchmarking des Tech-Stacks (AFE, ASR, TTS, KG, NLU, DM) • Design einer Software-Architektur und Integration des notwendigen Tech-Stacks zum Reifegrad eines MVP (Basis ist eine App auf einem Android-Endgerät) • Erprobung des MVP im Rahmen des Use Cases durch Nutzertests und technische Validierung des Funktionsumfangs und des Tech-Stacks im realen Nutzungskontext • Aufbereitung von Feedback zur Weiterentwicklung des Tech-Stacks basierend auf technischer und nutzerzentrierter Validierung Bewertung der Wirtschaftlichkeit für die Nutzung und Skalierung im Scope des MVP-Use Case • Vorbereitung und Teilnahme an öffentlichkeitswirksamen Kampagnen • Vorbereitung und Teilnahme an Regelterminen, Workshops und individuellen Fachdiskussionen mit Fraunhofer und div. Kooperationspartnern
Grundig Business System GmbH	Durchführung von mehreren Interviews mit der neuen Zielgruppe (Polizeikräfte auf Einsätzen) Auswertung der Interviews zur Anforderungsanalyse Erstellung von Mockups basierend auf den Anforderungen und folgende Evaluation der Mockups durch Nutzer aus der Zielgruppe Implementierung des Use Case Demonstrators basierend auf den Mockups ◦ Integration der SPEAKER Komponenten Spracherkennung und Dialogmanagement ◦ Integration von Konkurrenzprodukten zur vergleichweisen Nutzung ◦ Design von Dialogen mit dem Dialogeditor • Bereitstellung von Trainingsdaten zur Verbesserung der Erkennung von Emails ◦ Erstellung einer Vielzahl von Beispiemailen ◦ Aufnahme der Emails durch mehrere Sprecher mit Software von der LMU • Feedback zur TTS in Form von Hörtests und zum Dialogeditor in gemeinsamen Treffen • Teilnahme an allen Workshops, monatlichen, Fokus- & Meilenstein-treffen • Vorbereitung der Veröffentlichung des Demonstrators für Nutzertests
SAP SE	Iterative Proof-of-concept Implementierung des use case Prototypen "Komplementäre Spracherkennung im Kontext einer Plattform für den Bau von chatbots" FAQ-bot Implementierung (Deutsch/Englisch) mit Bezug zu User Assistance Iterative Proof-of-concept Implementierung des use case Prototypen "Simultane, automatische Transkription und Textanzeige während Veranstaltungen" (Iteration 1: ohne Streaming, Iteration 2: mit Streaming)
ONSEI GmbH	Erfolgreiche Umsetzung des Use-Cases "Sprachassistenten im Bereich Hotellerie"

Anwendungspartner	Umgesetzte Anwendung
	Testphase zur Erprobung des Use-Cases im Jaz in the City Hotel Vienna von 13.01.2023-25.09.2023 • Rollout vor Ort mit Ausstattung von 30-38 Hotelzimmern unterschiedlicher Kategorien mit jeweils 1 Smart Speaker & Informationsflyer • Workshop für Hotelpersonal zu Nutzungsoptionen und Handling der Geräte sowie Übergabe von Ersatzgeräten, Trouble Shooting Laptop und Tablet zur Nutzung der Admin App • Betreuung, Analyse und Reporting der Testphase mit regelmäßigen Abstimmungen mit den Verantwortlichen auf Hotelseite • VQE (Voice quality enhancement) Update im Hotel vor Ort (25.-26.5.2023) • Evaluation von alternativen Wake-Word Erkennungslösungen mit Whisper • Optimierung von Logging und Serverstabilität • Analyse und Auswertung der Speaker Nutzung unter Verwendung der Belegungszahlen der Testzimmer und der Logs ◦ Nutzeraktivität: bis zu 89% der belegten Testzimmer mind. 1x/Tag Nutzung • Auswertung des Feedbacks der Online-Umfrage durch Hotelgäste Vorbereitung und Teilnahme an Workshops, monatlichen, Fokus- & Meilensteintreffen sowie SPEAKER Abschlussveranstaltung
Scopevisio AG	
Siemens AG	
TU Braunschweig	
Universität Paderborn	Entwickelt wurde ein offline Question Answering System Dieses besteht aus folgenden Komponenten: • Query Generierung • Graph Zusammenfassung • Triplestore • Verbalisierung(optional) Für die Graphzusammenfassung wurden verschiedene Algorithmen zum Ranking von Entitäten in Wissensgraphen evaluiert. Zusätzlich wurde eine Zusammenfassung von Entitäten durchgeführt. Das Ziel war es den Graphen zu verkleinern, um somit die Datenmenge zu verringern. Im Ergebnis konnten die Antwortzeiten des QA Systems verbessert werden. Für die Generierung der Queries wurde das System TeBaQA entwickelt, welches auf Query Templates basiert. Später wurden auch Sprachmodelle zur Generierung der Queries verwendet, welche die Qualität der Queries stark verbessern konnten und auch für die weitere Forschung relevant sind. Im Bereich Triplestores wurde der Triplestore Tentris weiterentwickelt. Unter anderem wurde eine Persistente Version entwickelt und Update und Delete Queries wurden ermöglicht. Zudem wurden wird nun der SPARQL 1.1 Standard unterstützt. Für die Interaktion mit Nutzern ist zudem die Verbalisierung von Antworten relevant. Zu dem Zweck wurde das System Nabu zur Verbalisierung von Graph Daten entwickelt. Ziel der Weiteren Forschung ist zudem die Integration von Queries in das Verbalisierungsmodell.

6.6 Projektmanagement und Öffentlichkeitsarbeit

Das **siebte Arbeitspaket** umfasste Ziele des Projektmanagements und Öffentlichkeitsarbeit. Hierzu sollte wissenschaftliche Forschung zu Sprachassistententechnologien einem regelmäßigen Monitoring unterzogen werden, sodass auf aktuelle wissenschaftliche Entwicklungen schnell reagiert werden konnte. Auch die industriellen Entwicklungen sollten über das Monitoring von Presse, Fachtagungen und Veranstaltungen verfolgt werden. Wichtige Informationen sollten den Projektpartnern umgehend bereitgestellt werden. Zur Erarbeitung und Umsetzung von rechtlichen Rahmenbedingungen sollten Kontakte und Austausch zu Rechtsexperten und Datenschutzbeauftragten etabliert werden. Darüber hinaus sollte das Projektmanagement unter anderem folgende Aufgaben durchführen: Berichtswesen, Durchführung von regelmäßigen Meetings, Einberufung der Beiräte, Konfliktmanagement, Qualitätsmanagement, Einhaltung von Terminen und Fristen.

Zur Erreichung der Ziele wurde die folgende Projektinfrastruktur vom Fraunhofer bereitgestellt:

- 1 *OwnCloud*: Zum Tausch von sensiblen Daten im Konsortium nutzen wir die von Fraunhofer gehostete OwnCloud Instanz. Diese ermöglicht feingranulare Zugriffbeschränkungen. Entsprechend wurden Ordner eingerichtet auf welche alle Verbundpartner des Konsortiums Zugriff haben.
- 2 *GitLab*: Zum Austausch von Source Code sowie zur Erstellung von einfachen Tickets verwenden wir eine von Fraunhofer gehostete Instanz von GitLab
- 3 *Confluence*: Wird als Wiki System verwendet um verschiedenste Informationen strukturiert ablegen zu können. Hier werden Technologiebeschreibungen ebenso wie Besprechungsnotizen abgelegt und dokumentiert.
- 4 *Jira Service Desk*: Hier konnten evaluierende Partner Bugs in unseren Technologiekomponenten melden, welche als Tickets direkt in den agilen Softwareentwicklungs
- 5 *Mailinglisten*: Mehrere Mailinglisten wurden eingerichtet um auf einfache Art und Weise verschiedene Teilgruppen des Konsortiums erreichen zu können.

Projektbegleitend fanden zahlreiche Treffen mit allen Beteiligten des Projekts statt. Die nachstehende *Tabelle 6* zeigt eine Auswahl der Treffen, welche im Rahmen des SPEAKER Projekts stattgefunden haben.

Tabelle 6: Stattgefundene Treffen des Konsortiums

Datum	Veranstaltung	Teilnehmer
07.04.2020	Projekt Kick-Off	Begleitforschung, Fördergeber, Projektträger, Verbundpartner
09.07.2020	Monthly	Verbundpartner
30.07.2020	Voice UX Workshop	Verbundpartner, assoziierte Partner

Datum	Veranstaltung	Teilnehmer
13.08.2020	Monthly	Verbundpartner
10.09.2020	Monthly	Verbundpartner
30.09.2020	Vorstellung Cloud & Heat	Verbundpartner
08.10.2020	Milestone Meeting	Begleitforschung, Fördergeber, Projektträger, Verbundpartner
12.11.2020	Monthly	Verbundpartner
13.11.2020	Data Annotation Workshop	Verbundpartner, assoziierte Partner
26.11.2020	Platform Workshop	Verbundpartner
09.12.2020	Modell-Workshop Spracherkennung	Verbundpartner
10.12.2020	Monthly	Verbundpartner
14.01.2021	Monthly	Verbundpartner
11.02.2021	Monthly	Verbundpartner
23.02.2021	Wikispeech-Workshop	Verbundpartner
04.03.2021	Workshop Dialogmanager, Dialogeditor und NLU	Verbundpartner
11.03.2021	Monthly	Verbundpartner
16.03.2021	Multimodality Workshop	Verbundpartner, assoziierte Partner
18.03.2021	Workshop Text-to-Speech	Verbundpartner
08.04.2021	Monthly	Verbundpartner
20.05.2021	Monthly	Verbundpartner
10.06.2021	Monthly	Verbundpartner
24.06.2021	Data Annotation Workshop	
08.07.2021	Monthly	Verbundpartner
12.08.2021	Monthly	Verbundpartner
09.09.2021	Monthly	Verbundpartner
14.10.2021	Monthly	Verbundpartner
11.11.2021	Monthly	Verbundpartner
06.12.2021	Wissenschaftlicher Beirat	Wissenschaftlicher Beirat
08.12.2021	Industrieller Beirat	Industrieller Beirat
09.12.2021	Monthly	Verbundpartner
20.12.2021	Milestone Meeting	Begleitforschung, Fördergeber, Projektträger, Verbundpartner
13.01.2022	Monthly	Verbundpartner
10.02.2022	Monthly	Verbundpartner
24.02.2022	Question Answering over Knowledge Graphs	Verbundpartner, assoziierte Partner
10.03.2022	Monthly	Verbundpartner
14.04.2022	Monthly	Verbundpartner
12.05.2022	Monthly	Verbundpartner
01.06.2022	Wissenschaftlicher Beirat	Wissenschaftlicher Beirat
08.06.2022	Industrieller Beirat	Industrieller Beirat
23.06.2022	Monthly	Verbundpartner
14.07.2022	Milestone Meeting	Begleitforschung, Fördergeber, Projektträger, Verbundpartner
11.08.2022	Monthly	Verbundpartner
08.09.2022	Monthly	Verbundpartner
13.10.2022	Monthly	Verbundpartner
08.12.2022	Monthly	Verbundpartner

Datum	Veranstaltung	Teilnehmer
13.12.2022	Industrieller Beirat	Industrieller Beirat
15.12.2022	Wissenschaftlicher Beirat	Wissenschaftlicher Beirat
12.01.2023	Monthly	Verbundpartner
09.02.2023	Monthly	Verbundpartner
09.03.2023	Monthly	Verbundpartner
11.05.2023	Monthly	Verbundpartner
29.06.2023	Monthly	Verbundpartner
13.07.2023	Monthly	Verbundpartner
10.08.2023	Monthly	Verbundpartner
14.09.2023	Monthly	Verbundpartner
28.09.2023	Abschlussveranstaltung	Begleitforschung, Fördergeber, Projektträger, Verbundpartner, assoziierte Partner

Wissenschaftliche Forschung zu Sprachassistententechnologien wurden fortlaufend beobachtet. Ebenfalls wurden die industriellen Entwicklungen einem regelmäßigen Monitoring unterzogen. Die Entwicklungen konnten den Partnern beispielsweise durch das monatlich stattfindende Monthly mitgeteilt werden. In die Agenda dieses regelmäßig stattfindenden Treffens wurde die Diskussion neuer Veröffentlichungen und relevanter Veranstaltungen stets in die Agenda mitaufgenommen.

6.7 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die geleistete Arbeit diene dem Ziel eine Sprachassistentenplattform „Made in Germany“ zu errichten. Hierdurch sollte eine Alternative zu amerikanischen und asiatischen Anbietern geschaffen werden, mit besonderem Fokus auf die Bedürfnisse der Industrie (Bspw. Datensicherheit, Datensouveränität, ...) siehe auch Abschnitt 2.3.

Die Erarbeitung einer detaillierten Anforderungsanalyse, Architektur und Benchmarks war notwendig zur Entwicklung der Sprachassistententechnologien. Hierdurch konnten Prioritäten in der Entwicklung gesetzt werden, um nach Abschluss der Umsetzungsphase relevante Anwendungsfälle mit der Sprachassistentenplattform bedienen zu können.

Die Notwendigkeit der geschäftlichen Plattformentwicklung und des Community Management stellen sich wie folgt dar. Zum einen konnte hierdurch das SPEAKER-Ökosystem nach außen bekannt gemacht werden und die Reichweite und Sichtbarkeit des Projektes erhöht werden. Durch eine Markt- und Wettbewerbsbeobachtungen konnten Trends in der Branche erkannt und die Plattform nach neuen Gegebenheiten des Marktes angepasst und ausgerichtet werden. Zudem

konnten SPEAKER-interne Governance-Strukturen und ein Konzept für eine mögliche Betreiber-gesellschaft erstellt werden.

Die technische Plattformentwicklung ist grundlegenden für den Aufbau der SPEAKER Plattform gewesen. Hierdurch wurde eine skalierbare, vertrauenswürdige und getestete SPEAKER Plattform geschaffen. Gleiches zeichnet die Notwendigkeit der Entwicklung der Sprachassistenzmodule. Auf diese Weise entstanden getestete, dokumentierte und evaluierte Module für die SPEAKER-Plattform.

Die Bereitstellung von Datenkonnektoren für die Module ermöglichten das Andocken an etablierte Datenquellen. So müssen weder individuelle Konnektoren geschaffen noch separate Module mit ähnlichen Fähigkeiten pro Anwendungsfall implementiert werden. Die Erhebung von Daten beispielsweise zur Voice User Experience stellten eine hohe Qualität sicher und dienten der Vergleichbarkeit der Module mit denen anderer Anbieter. So entstand eine Sammlung an getesteten Datenkonnektoren, Benchmarkframeworks und Daten sowie multilinguale Erweiterungen geeigneter Module.

Die Evaluation der SPEAKER Plattform in Anwendungsfällen diente dazu, erste Pilotanwendungen zu testen und somit sicherzustellen, dass die Plattform produktiv nutzbar und auch in der Anwendung erprobt ist. Hierzu wurden mehrere Pilotanwendungen aus verschiedenen Feldern getestet, wodurch spezialisierte Technologiemodule entstanden, welche beispielsweise auch mit Fachvokabular umgehen können.

Das Projektmanagement diente dazu einen reibungslosen Ablauf des Projekts zu gewährleisten.

6.8 Voraussichtlicher Nutzen und Verwertbarkeit

Die errichtete SPEAKER Plattform bietet zahlreiche Verwertungsperspektiven. Die „Made in Germany“ Sprachassistenzplattform ist eine branchenübergreifende Lösung und als diese in diversen Markt- und Anwendungsbereichen nutzbar. Viele Unternehmen nutzen Sprachschnittstellen bereits für zahlreiche Anwendungen.

In erster Linie richtet sich die Plattform an Unternehmen aus der Industrie und an B2B-Anwendungen. Hierbei ist die Plattform speziell für Unternehmen interessant, welche auf ein hohes Maß an Datensouveränität und Datenunabhängigkeit setzten. Da die Plattform beispielsweise auch lokal, sprich on-premise nutzbar ist.

Im Rahmen des Projekts wurden zudem Cloud-Technologien weiterentwickelt und implementiert, um die digitale Souveränität in Deutschland und Europa zu stärken. Nachfolgend eine Zusammenfassung der verwendeten Technologien sowie deren Beitrag zur Verbesserung der Datensouveränität.

Verwendete Cloud-Technologien:

Die verwendeten Cloud-Technologien basieren auf Open Source und sind frei verfügbar. Als Basistechnologie wurde OpenStack (<https://www.openstack.org/>) eingesetzt. OpenStack ist eine bewährte Open-Source-Plattform für die Bereitstellung von Cloud-Infrastrukturen und bietet eine Vielzahl von Diensten für die Verwaltung von virtuellen Maschinen, Speicher und Netzwerken.

Zusätzlich wurde Kubernetes eingesetzt, das mit der Open-Source-Software Yaook (<https://yaook.cloud/>) provisioniert wurde. Yaook ist in den ALASCA-Verband (Verband für betriebsfähige, offene Cloud-Infrastrukturen e.V.; <https://alasca.cloud/>) eingebettet. ALASCA fördert betriebsfähige Open-Source-Technologien, bündelt Kompetenzen und ermöglicht den Wissenstransfer.

Beitrag zur Datensouveränität:

Die Verwendung von OpenStack und Yaook, die beide auf Open-Source-Prinzipien beruhen, trägt wesentlich zur Datensouveränität bei. Durch den Einsatz von Open-Source-Technologien wird die Abhängigkeit von proprietären Lösungen reduziert, was Unternehmen und Organisationen mehr Kontrolle über ihre Daten gibt.

Der ALASCA-Verband spielt ebenfalls eine wichtige Rolle bei der Förderung der Datensouveränität. Durch die Bündelung von Kompetenzen und den Wissenstransfer trägt ALASCA dazu bei, dass Organisationen in Deutschland und Europa ihre Cloud-Infrastrukturen eigenständig betreiben und kontrollieren können.

Die verwendeten Cloud-Technologien, basierend auf OpenStack und Yaook, sowie die Unterstützung durch den ALASCA-Verband, leisten einen bedeutenden Beitrag zur digitalen Souveränität in Deutschland und Europa. Durch die Förderung von Open-Source-Technologien und den Aufbau lokaler Kompetenzen wird die Abhängigkeit von ausländischen Anbietern verringert und die Kontrolle über Daten gestärkt.

Auf Basis der Pilotanwendungen ist die Plattform nach der Umsetzungsphase bereits auf einige Branchen/Anwendungsfälle spezialisiert. Hierzu gehört beispielsweise die Hotellerie, Medizinbranche sowie die öffentlich/digitale Verwaltung.

6.9 Fortschritte von Sprachtechnologien im Förderzeitraum

Der Sprachassistentenmarkt und Wettbewerb wurde im Förderzeitraum kontinuierlich beobachtet und analysiert. Die meisten Anbieter von Sprachtechnologien haben während des Projektzeitraums die Anzahl verfügbarer Sprachen und Stimmen ausgebaut. Zu den Entwicklungen und Trends der von Text-to-Speech Anwendungen gehören eine umfassende Stimmauswahl und Dialekte, einzigartige Markenstimmen, die Anpassbarkeit von Stimmen in Bezug auf den Text, die Geschwindigkeit, Tonhöhe und Stil sowie eine genderneutrale Stimme. Im Bereich der Automatic-Speech-Recognition (ASR) wurde die Multikanal-Erkennung, Sprechererkennung, Sentimentanalyse, Identifikation sowie das Filtern von Inhalten und Nebengeräuschen vorangetrieben.

In Bezug auf die Datenschutzstandards, erhöhten einige Anbieter die Datensicherheit und Souveränität ihrer Sprachlösung. Beispielweise stieg das Angebot an Sprachtechnologien, welche lokal bzw. on premise verfügbar sind.

Die Marktstrukturen haben sich nur in einem geringen Maße verändert. Die Big Player auf dem Markt sind nach wie vor Microsoft, Amazon, Google und IBM. Microsoft hat mit dem Kauf von Nuance im April 2021 einen beachtlichen Zugewinn and Knowhow verbuchen können. Durch den Erwerb konnte Microsoft unter anderem in der Medizinbranche Marktanteile gewinnen. Die Aneignung von KnowHow und Technologien durch die Akquise von Wettbewerbern ist gängige Praxis.

6.10 Geplante und erfolgte Veröffentlichungen

A. Leschanowsky, B. Brüggemeier, N. Peters, (2021) Design Implications for Human-Machine Interactions from a Qualitative Pilot Study on Privacy. Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication, 76-79, doi: 10.21437/SPSC.2021-16 https://www.isca-speech.org/archive/spsc_2021/leschanowsky21_spsc.html

A. Sadeghi, D. Collarana, D. Graux, J. Lehmann, (2021) Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities. ECML/PKDD (2) 2021: 548-564. Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II. Lecture Notes in Computer Science 12976, Springer 2021, ISBN 978-3-030-86519-1

- A. Zarcone, J. Lehmann, E. Habets (2021) Small Data in NLU: Proposals towards a Data-Centric Approach. Proceedings of the NeurIPS Data Centric AI Workshop
- D. Graux, D. Collarana, F. Orlandi, (2021) Formal Concept Analysis for Semantic Compression of Knowledge Graph Versions. FCA4AI@IJCAI 2021: 131-138. Proce. 9th International Workshop "What can FCA do for Artificial Intelligence?" co-located with the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), Montréal, Québec, Canada, August 21, 2021. CEUR Workshop Proceedings 2972
- L. Hrycyk, A. Zarcone, L. Hahn, (2021) Not So Fast, Classifier – Accuracy and Entropy Reduction in Incremental Intent Classification. Proc. 3rd Workshop on Natural Language Processing for Conversational AI, Punta Cana, 2021 <https://aclanthology.org/2021.nlp4convai-1.6/>
- M. Kurz, B. Brüggemeier, M. Breiter, (2021) Success is not Final; Failure is not Fatal - Task Success and User Experience in Interactions with Alexa, Google Assistant and Siri. HCI. <https://www.springerprofessional.de/en/success-is-not-final-failure-is-not-fatal-task-success-and-user-/19326834>
- N. Wilhelm, D. Collarana, J. Lehmann, (2021) A Virtual Knowledge Graph for Enabling Defect Traceability and Customer Service Analytics. ESWC (Satellite Events) 2021: 245-248. The Semantic Web: ESWC 2021 Satellite Events - Virtual Event, June 6-10, 2021. Lecture Notes in Computer Science 12739, Springer 2021, ISBN 978-3-030-80417-6
- P. Govalkar, A. Mustafa, N. Pia, J. Bauer, M. Yurt, Y. Özer, C. Dittmar, (2021) A Lightweight Neural TTS System for High-quality German Speech Synthesis. Proceedings of the ITG Speech Communication, Kiel, 2021
- T. Alam, A. Zarcone, S. Padó, (2021) New Domain, Major Effort? How Much Data is Necessary to Adapt a Temporal Tagger to the Voice Assistant Domain. Proc. 14th International Conference on Computational Semantics (IWCS), Groningen, 2021 <https://aclanthology.org/2021.iwcs-1.14/>
- Y. Frommherz and A. Zarcone. (2021) Crowdsourcing Ecologically-Valid Dialogue Data for German. Frontiers in Computer Science, 3, 2021 <https://www.frontiersin.org/articles/10.3389/fcomp.2021.686050/full>
- F. Zalkow et al. "Evaluating speech–phoneme alignment and its impact on neural text-to-speech synthesis". Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP). 2023, pp. 1–5.
- K. Kayyar et al. "Multi-Speaker text-to-speech using ForwardTacotron with improved duration prediction". Proceedings of the ITG Conference on Speech Communication. 2023.
- P. Sani et al. "Improving the naturalness of synthesized spectrograms for TTS using GAN-based post-processing". Proceedings of the ITG Conference on Speech Communication. 2023.
- K. Kayyar et al. "Low-Resource text-to-speech using specific data and noise augmentation". Proceedings of the European Signal Processing Conference (EUSIPCO). 2023.
- K. Kayyar et al. "Subjective evaluation of text-to-speech models: Comparing absolute category rating and ranking by elimination tests". Proceedings of the Workshop on Speech Synthesis (SSW). 2023, pp. 191–196.

F. Zalkow et al. "The AudioLabs System for the Blizzard Challenge 2023". Proceedings of the Blizzard Challenge Workshop, 2023, pp. 63-68.

Moussallem, Diego and Gnaneshwar, Dwaraknath and Castro Ferreira, Thiago and Ngonga Ngomo, Axel-Cyrille: NABU - Multilingual Graph-based Neural RDF Verbalizer, The Semantic Web -- ISWC 2020

Bigerl, Alexander and Conrads, Felix and Behning, Charlotte and Sherif, Mohamed and Saleem, Muhammad and Ngonga Ngomo, Axel-Cyrille: Tentriss – A Tensor-Based Triple Store, The Semantic Web -- ISWC 2020

Firmansyah, Asep Fajar and Moussallem, Diego and Ngomo, Axel-Cyrille Ngonga: GATES: Using Graph Attention Networks for Entity Summarization -- K-CAP 2021: Knowledge Capture Conference, 2021

Vollmers, Daniel and Jalota, Richa and Moussallem, Diego and Topiwala, Hardik and Ngomo, Axel-Cyrille Ngonga and Usbeck, Ricardo: Knowledge Graph Question Answering using Graph-Pattern Isomorphism, Further with Knowledge Graphs - Proceedings of the 17th International Conference on Semantic System, 2021

Jalota, Richa and Vollmers, Daniel and Moussallem, Diego and Ngomo, Axel-Cyrille Ngonga: LAUREN - Knowledge Graph Summarization for Question Answering, Proceedings of the 15th IEEE International Conference on Semantic Computing ICSC, 2021

Röder, Michael and Frerk, Philip and Conrads, Felix and Ngomo, Axel-Cyrille Ngonga Applying Grammar-based Compression to RDF, Eighteenth Extended Semantic Web Conference - Research Track, 2021

Manzoor Ali, Muhammad Saleem and Axel-Cyrille Ngonga Ngomo: REBench: Microbenchmarking Framework for Relation Extraction Systems -- ISWC 2022

Hamada M. Zahera, Daniel Vollmers, Mohamed Ahmed Sherif and Axel-Cyrille Ngonga Ngomo: MultPAX: Keyphrase Extraction using Language Models and Knowledge Graphs -- ISWC 2022

Alexander Bigerl, Lixi Conrads, Charlotte Behning, Muhammad Saleem and Axel-Cyrille Ngonga Ngomo: Hashing the Hypertrie: Space- and Time-Efficient Indexing for SPARQL in Tensors-- ISWC 2022

Anhang

<i>Anhang A</i>	<i>Übersicht - Assoziierte Partner</i>	<i>108</i>
<i>Anhang B</i>	<i>Arbeitspakete</i>	<i>109</i>
<i>Anhang C</i>	<i>Social-Media-Aktivitäten</i>	<i>118</i>
<i>Anhang D</i>	<i>Nutzerdokumentation Dialog Editor v3</i>	<i>124</i>

Anhang A Übersicht - Assoziierte Partner

Tabelle 7: Übersicht der assoziierten Partner

Unternehmen	Branche
4FriendsOnly Internet Technologies AG	IT Dienstleister
Aristech GmbH	Softwareentwickler
axxessio GmbH	Unternehmensberatung
BitVox	Softwareentwickler
C&S Computer und Software GmbH	Softwareanbieter
Comevis GmbH und Co. KG	Voice Branding
davero dialog GmbH	Call Center
divoice UG	Handel
E&L	Softwareentwicklung Medical (Befundung)
eco Verband der Internetwirtschaft e.V.	Internetbranche
Elektrobit Automotive GmbH	Automotive
Fauser AG	Softwareentwicklung
Future of Voice GmbH	Voice Startups
InfAI Institut für Angewandte Informatik	Verein zur Förderung von Wissenschaft und Forschung
INIT GmbH	Informatikanwendungen in Transport-, Verkehrs- und Leitsystemen
LAN1 Hotspots GmbH	Tourismus
Linguwerk GmbH	Softwareentwickler
LZE GmbH	Forschung und Produktentwicklung
msg systems ag	IT-Dienstleistung
NEC Laboratories Europe GmbH	Forschungsunternehmen, Öffentlichkeitssektor
Nuuk GmbH	Voice Startups
Omnibot GmbH	IT Dienstleister
Sensape GmbH	Retail/Handel
Sikom Software GmbH	Softwareentwickler
SimplyNote	Apps, Software
Tenios	Telekommunikationsanbieter
Uni Magdeburg	Forschung
viind GmbH	Softwareentwickler
VITAS GmbH	Voice Startups
VIVAI Software AG	Softwareentwickler
Wake Word GmbH	Medien
Westdeutscher Rundfunk Köln	Medien
Wikifriend UG	Sprachassistenten auf Wikitech
Wolters Kluwer Deutschland GmbH	Rechtswesen
WorkHeld (Tablet Solutions GmbH)	Assistenzapps für Industrie
Yellow Map AG	Mobilität, Geomarketing

Anhang B Arbeitspakete

AP-Nr.:1			Anforderungsanalyse, Architektur, Benchmarks							
Startdatum:			04/2020							
Enddatum:			03/2023							
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	1,5	1,7	4,0	5,2	6,0	15,0	8,0	4,0	5,0	3,0
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	9,0	2,0	1,0	9,0	6,0	6,0	9,0	3,0	6,0	104,4
Ziel des Arbeitspaketes: Zu Beginn von SPEAKER führen wir anhand von strukturierten Workshops eine weitere detaillierte Anforderungsanalyse zusammen mit allen Verbundpartnern und interessierten assoziierten Partnern durch. Daraus wird eine feingranulare Architekturspezifikation abgeleitet, welche den Entwicklungsprozess steuern und mittels KPIs nachvollziehbar machen wird. Daneben leiten wir einen Kuratierungsprozess ein, welcher Benchmarks (Daten, Experimenttypen, Metriken), Tests sowie eine Umsetzungsstrategie von Standards und (KI-)Zertifizierungen beinhaltet.										
1. Voraussetzung (Input) Die Verbundpartner, die spezielle Anforderungen an die SPEAKER-Plattform haben, sollen diese nach Möglichkeit vorab formulieren und priorisieren.										
2. Lösungsweg Über von Fraunhofer bereitgestellte Online-Tools zum Wissensmanagement werden bis zum ersten Meilenstein die spezifischen Anforderungen aller Verbundpartner gesammelt, geclustert und priorisiert. Daraus werden in von Fraunhofer organisierten Workshops eine konkrete Spezifikation der Architektur abgeleitet. Die Architektur bildet das Fundament für alle Implementierungs-Arbeitspakete, somit ist eine sorgfältige Umsetzung von enormer Wichtigkeit.										
Arbeitsschritt 1: Anforderungsanalyse Die Anforderungsanalyse erfolgt unter Beteiligung der Verbundpartner in mehreren Iterationen. Als Resultat entsteht eine nach Priorität sortierte Liste von Anforderungen an die einzelnen Technologiekomponenten.									64,8 PM	
Arbeitsschritt 2: Architekturspezifikation Die Architekturspezifikation beschreibt die Funktionalitäten der einzelnen Technologiekomponenten sowie deren Schnittstellen zur Ein- und Ausgabe von Datenströmen. Final bilden alle Komponenten zusammen die Gesamtarchitektur der SPEAKER-Plattform.									25,1 PM	
Arbeitsschritt 3: Kuratierungsprozess Ein wesentlicher Bestandteil sind die Benchmarks für jede einzelne Technologiekomponente. Diese werden von den entsprechenden Experten vorgeschlagen und priorisiert. Wichtig ist hierbei sowohl der Einsatz öffentlich verfügbarer Testdatensätze, um eine Vergleichbarkeit mit anderen System zu gewährleisten, als auch das Aufsetzen und Pflegen von SPEAKER-spezifischen Datensätzen.									14,5 PM	
Personalaufwand gesamt (in PM)									104,4 PM	
Ergebnis (Output): Eine detaillierte Beschreibung der Anwendungsfälle für den Einsatz von Sprachassistententechnologie, eine Zielspezifikation der umzusetzenden Funktionen und eine Ressourcensammlung für den Kuratierungsprozess von Datenressourcen.										
3. Sonstige Ressourcen Cloud-basierter persistenter Speicherplatz für umfängliche Testdatensätze										

AP-Nr.:2			Geschäftliche Plattformentwicklung & Community Management							
Startdatum:			04/2020							
Enddatum:			03/2023							
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	0,0	1,0	3,0	1,7	0,0	11,0	20,0	0,0	0,0	5,0
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	3,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	44,7
Ziel des Arbeitspaketes: Um SPEAKER zu einem erfolgreichen und weit ausstrahlenden KI-Ökosystem werden zu lassen, werden wir von Beginn an mit unseren Partnern Marketing-, Öffentlichkeits- und Netzwerkaktivitäten starten. Dazu werden wir industrielle Messen sowie wissenschaftliche Workshops, Vorträge und Tutorials ausrichten. Weiterhin werden wir eine Markt- und Wettbewerbsbeobachtungsstrategie ausarbeiten, welche zur Nachsteuerung der Plattform dient. In diesem AP werden wir auch Governance-Modelle und -Strukturen etablieren. Daraus werden wir die nötigen Schritte für die Konzeption und Etablierung der SPEAKER-Betreibergesellschaft ableiten und umsetzen.										
1. Voraussetzung (Input) Idealer Weise sind die Konsortialverträge bereits ausgehandelt, ausformuliert und von allen Partnern im Konsortium unterzeichnet.										
2. Lösungsweg Neben der Durchführung von Workshops mit interessierten am SPEAKER Projekt, werden wir in internen Workshops das Geschäftsmodell weiter detaillieren und ausarbeiten sowie die Ausgründung der Betreibergesellschaft am Modell der LZE GmbH vorbereiten und durchführen. Um den Erfolg der SPEAKER-Plattform sicherzustellen, werden neben Marketingaktivitäten, welche die Sichtbarkeit des Projekts erhöhen, eine kontinuierliche Markt- und Wettbewerbsbeobachtung durchgeführt. Diese ermöglicht es, neue Trends oder Konkurrenzen auszumachen und entsprechend den Funktionsumfang der SPEAKER-Plattform anzupassen.										
Arbeitsschritt 1: Marketing-, Öffentlichkeits- und Netzwerkaktivitäten Koordiniert durch Fraunhofer IAIS und Fraunhofer IIS werden alle Verbundpartnern unterschiedliche Maßnahmen nutzen (z.B. Messe-Exponate, Workshops), um die Sichtbarkeit des SPEAKER-Ökosystems zu stärken und somit die Reichweite des SPEAKER Netzwerks zu erhöhen.									20 PM	
Arbeitsschritt 2: Markt- und Wettbewerbsbeobachtung Durch eine kontinuierliche Recherche nach Marktstudien sowie durch eine intensive Beobachtung des Wettbewerbs wird sichergestellt, dass Trends rechtzeitig erfasst und die Ausrichtung der Plattform an neue Gegebenheiten des Marktes angepasst werden können.									5 PM	
Arbeitsschritt 3: Governancemodelle und -strukturen In diesem Arbeitsschritt werden wir zusammen mit unseren Partnern entsprechende Governancemodelle und -strukturen der Plattform erarbeiten und dokumentieren, um somit den funktionierenden Betrieb der Plattform und eine reibungslose Übergabe der Plattform an die Betreibergesellschaft sicherzustellen.									5 PM	
Arbeitsschritt 4: Gründen der Betreibergesellschaft Unter Führung von Fraunhofer IAIS und Fraunhofer IIS werden gemeinsam mit den Verbundpartnern die Details des Geschäftsmodells erarbeitet und die Betreibergesellschaft konzipiert. Hierzu müssen, nach dem Vorbild der LZE-GmbH Gesellschafter Verträge ausgestaltet werden und die Gesellschaft gegründet werden.									14,7 PM	
Personalaufwand gesamt (in PM)									44,7 PM	
Ergebnis (Output): Events zur Bekanntmachung und Verstetigung des SPEAKER-Ökosystems nach außen sowie die Etablierung der SPEAKER internen Governance-Struktur und Betreibergesellschaft mit Fokus auf die Zeit nach der Umsetzungsphase.										
3. Sonstige Ressourcen:										

- Materialien für Öffentlichkeitsarbeit (Roll-ups, Flyer, Broschüren)
- Aufträge an Messebauer und Eventagenturen zur Organisation von Veranstaltungen für das SPEAKER Ökosystem
- Unterbeauftragung von Agenturen zum Erstellen von Videos, Grafiken etc.

AP-Nr.:3		Technische Plattformentwicklung								
Startdatum:		04/2020								
Enddatum:		03/2023								
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	0,0	0,0	0,0	2,3	33,0	147,0	73,0	0,0	0,0	6,0
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	20,5	14,0	0,0	11,0	0,0	0,0	0,0	0,0	0,0	306,8
Ziel des Arbeitspaketes: Das Herz von SPEAKER ist die Modulplattform , welche den Austausch, Betrieb und die Nutzung von KI-getriebener Sprachassistententechnologie erst ermöglicht. Daneben bauen wir eine Datenplattform auf, welche das Training der Module in speziellen Domänen ermöglicht und neue Geschäftsmodelle (bspw. Kuratierung und Bereitstellung von Datensätzen) fördert. Darüber hinaus erstellen wir eine Web-Plattform sowie Schnittstellen zu deutschen und europäischen Plattformen. Ebenso werden wir in diesem AP am 1) Aufbau des Billing und Identity-Mechanismus, 2) der Skalierbarkeit der Plattform und 3) Safety-, Security- und Privacy- sowie Standardisierungs- und Zertifizierungsaspekten arbeiten.										
1. Voraussetzung (Input) Klare Definition der Anforderungen und Schnittstellen aus AP1. Bereitstellung von Cloud-Computing-Ressourcen durch den Cloud-Hosting-Partner inNOVO Cloud.										
2. Lösungsweg Die SPEAKER-Plattform wird als System von Microservices, d.h. voneinander unabhängigen Diensten, implementiert. Alle auf Fraunhofer-Technologie basierenden Module (AFE, ASR, NLU, QA, DM, TTS, etc.) werden als skalierbare Docker-Container gekapselt. Erweiterungsmodule der Verbundpartner werden ebenfalls auf diesem Weg in der Cloud instanziiert und über eine einheitliche Schnittstelle zur Verfügung gestellt. Die SPEAKER-Datenplattform und SPEAKER Web-Plattform komplettieren den Funktionsumfang der Plattform.										
Arbeitsschritt 1: Aufbau der SPEAKER Modulplattform In diesem Arbeitsschritt wird die SPEAKER-Plattform als System von Microservices, also voneinander unabhängigen Diensten implementiert. Alle Module (AFE, ASR, NLU, TTS, etc.) werden als skalierbare Docker-Container gekapselt. Die in der Cloud laufenden Dienste werden über eine einheitliche Schnittstelle zur Verfügung gestellt.									61,3 PM	
Arbeitsschritt 2: Aufbau der SPEAKER Datenplattform Es wird eine Datenplattform aufgebaut, welche für den Austausch, die (Weiter-) Nutzung und das Benchmarking von Sprachassistentenmodulen genutzt werden kann. Dies wird basierend auf der Weiterentwicklung der Technologien aus den Vorprojekten (HOBbit, GeoKnow, QROWD, European Language Grid, LOD2) durchgeführt.									42,5 PM	
Arbeitsschritt 3: Aufbau einer Web-Plattform Durch die Implementierung einer grafischen Web-Oberfläche wird der Zugriff und die Verwaltung der SPEAKER-Plattform für Anwender ermöglicht.									60,5 PM	
Arbeitsschritt 4: Schnittstellen zu deutschen und europäischen Plattformen Um die Reichweite des SPEAKER Ökosystems sicherzustellen, wird eine Anbindung an bereits bestehende Plattformen angestrebt. Hierbei sollen bereits von Beginn an neben Deutschen auch Europäische Plattformen mit angebunden werden.									28,0 PM	
Arbeitsschritt 5: Aufbau des Billing-Mechanismus									3,5 PM	

In diesem Arbeitsschritt wird eine KI-Plattformarchitektur entwickelt, welche das Billing hinsichtlich der Nutzung von Sprachassistenten-Modulen ermöglicht.	
Arbeitsschritt 6: Skalierbarkeit In diesem Schritt wird sichergestellt, dass die Dienste skalierbar sind, d.h. dass jeder Dienst auf beliebigen Servern ausgeführt werden kann, so dass entweder alle Module in der Cloud oder lokal auf der Hardware der Verbundpartner durch die Nutzung der Docker-Technologie ausgeführt werden können.	40,5 PM
Arbeitsschritt 7: Safety, Security, Privacy, Zertifizierung Um das Vertrauen der Nutzer in die Plattform zu stärken, wird eine Zertifizierung einzelner KI-Module angestrebt. Anpassungen, die für eine erfolgreiche Zertifizierung notwendig sind, werden entsprechend vorgenommen. Diese Komponente muss parallel zur Entwicklung der SPEAKER-Web-Plattform bereitgestellt werden.	30,5 PM
Arbeitsschritt 8: Aufbau des Identity-Managements Entwicklung einer Authentifizierung-Schnittstelle inkl. eines Registrierungs-Endpunktes für die Integration von Identities aus anderen Diensten.	8,5 PM
Arbeitsschritt 9: Standardisierung Implementierung neuer und Optimierung bereits bestehender Komponenten zur Wahrung von einheitlichen Standards.	12,5 PM
Arbeitsschritt 10: Multilingualität, Europäisierung, Internationalisierung Die Plattform soll über viele Sprachen hinweg nutzbar sein. Dafür sollen bereits frühzeitig Möglichkeiten geschaffen werden, so dass die Erweiterungen keinen bedeutenden Implementierungsaufwand erfordern. Die Umsetzung soll anhand der beiden Sprachen Deutsch und Englisch getestet werden.	19,0 PM
Personalaufwand gesamt (in PM)	306,8 PM
Ergebnis (Output): Skalierbare, vertrauenswürdige und getestete SPEAKER-Plattform zur Bereitstellung und Nutzung von Sprachassistententechnologien und -daten.	
3. Sonstige Ressourcen Keine	

AP-Nr.:4		Sprachassistentenmodule								
Startdatum:		04/2020								
Enddatum:		12/2022								
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	0,0	0,0	0,0	2,3	35,0	63,0	95,0	0,0	0,0	0,0
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	0,0	0,0	0,0	0,0	0,0	0,0	0,0	24,0	0,0	219,3
Ziel des Arbeitspaketes: In diesem Arbeitspaket werden folgende Funktionalitäten und Module für die SPEAKER-Plattform weiterentwickelt und als Docker-Container bereitgestellt, um diese dann in der Plattform zu integrieren und bereitzustellen: • Acoustic Frontend / Quellentrennung (Unterdrückung von Echo, Hall und Störgeräuschen) • Spracherkennung (Robustheit, optimierte Erkennungsraten, Dialekte, Spontansprache) • Natural Language Understanding (Domänenunabhängig, präzise, skalierbar, Offline-fähigkeit, semantisch) • Dialogmanager und -interfaces (Einfach und intuitiv sowie erweiterbar, aus Daten antrainierbar, on-premise) • Sprachsynthese und Generierung (weitere Stimmen, Echtzeitfähigkeit, Prosodie) Dazu werden bestehende Technologien und Module erweitert, angepasst, neu trainiert und einem ausführlichen Benchmarking unterzogen.										
1. Voraussetzung Basierend auf schon bereits durch Fraunhofer IAIS und Fraunhofer IIS entwickelte Technologiekomponenten (AFE, ASR, NLU, DM, TTS) werden erweiterte und verbesserte Module dieser										

Technologiebausteine auf die SPEAKER-Plattform portiert und können mithilfe von klar definierten Schnittstellen als Docker Container bereitgestellt werden.	
2. Lösungsweg Mit Hilfe der Definitionen der speziellen Anforderungen der Verbundpartner und deren Anwendungsszenarien werden die beiden Fraunhofer Institute bereits vorhandene Technologiemodule entsprechend der Anforderungen weiterentwickeln und anpassen. Andere Verbundpartner erweitern die Plattform um Module, so liefert z.B.: AudeERING ein Emotionserkennungsmodul, Retresco ein Modul zur automatischen Sprachgenerierung und die TUB ein Modul zur multimodalen Spracherkennung.	
Arbeitsschritt 1: Acoustic Frontend / Quellentrennung Basierend auf bereits vorhandenen Technologiekomponenten des Fraunhofer IIS werden diese in diesem Zusammenhang auf die Anwendung im Kontext des SPEAKER Projektes optimiert, um eine zuverlässige Spracherkennung auch in akustisch schwierigen Szenarien zu gewährleisten.	41 PM
Arbeitsschritt 2: Spracherkennung Basierend auf dem Spracherkennungsmodul des Fraunhofer IAIS wird dieses verbessert und auf die speziellen Anforderungen unserer Partner angepasst. Das Fraunhofer IIS wird in diesem Zusammenhang weitere Module zum Keyword-Spotting sowie zur Sprechererkennung entwickeln und auf der Plattform bereitstellen.	54,6 PM
Arbeitsschritt 3: Natural Language Understanding Basierend auf den Vorarbeiten seitens Fraunhofer IIS und IAIS werden die Technologiekomponenten zum Sprachverstehen verbessert und an die Anforderungen der Verbundpartner angepasst sowie auf der SPEAKER-Plattform bereitgestellt.	89,9 PM
Arbeitsschritt 4: Dialogmanager und -interface Für die SPEAKER-Plattform wird das Fraunhofer IAIS vorhandene Technologiekomponenten hinsichtlich des Dialogmanagements verbessern, auf die Anforderungen unserer Verbundpartner anpassen und auf der Plattform bereitstellen.	9,6 PM
Arbeitsschritt 5: Sprachsynthese und Verbalisierung Basierend auf Vorarbeiten des Fraunhofer IIS wird ein universell einsetzbares Sprachsynthesemodul entwickelt und bereitgestellt. Hier wird besonders auf die Bedürfnisse der Verbundpartner hinsichtlich des Fachvokabulars eingegangen.	24,5 PM
Personalaufwand gesamt (in PM)	219,6 PM
Ergebnis (Output): Getestete, dokumentierte und evaluierte Module für die SPEAKER-Plattform.	
3. Sonstige Keine	Ressourcen

AP-Nr.:5		Tools und Daten									
Startdatum:		04/2020									
Enddatum:		03/2023									
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA	
PM	1,5	0,0	6,0	3,3	3,0	57,0	73,0	2,0	0,0	0,0	
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM	
PM	0,0	20,0	1,0	0,0	5,0	4,0	0,0	3,0	0,0	178,8	
Ziel des Arbeitspaketes: Um im B2B-Bereich erfolgreich und großflächig domänenübergreifend anwendbar zu werden, wird SPEAKER Datenkonnektoren für die Module bereitstellen, welche ein Andocken an etablierte Datenquellen ermöglichen. So müssen weder individuelle Konnektoren geschaffen noch separate Module mit ähnlichen Fähigkeiten pro Anwendungsfall implementiert werden. Daneben werden wir die existenten Module auf multilinguale Erweiterungen prüfen und, wenn möglich, diese vornehmen. In diesem AP werden ebenfalls Benchmark-Frameworks für einzelne Module und für die Voice User Experience implementiert											

sowie dafür notwendige Daten von Verbundpartner erhoben , um eine hohe Qualität zu sichern und Module verschiedener Anbieter vergleichen zu können.	
1. Voraussetzung Die skizzierten Pilotanwendungen, welche im SPEAKER Konzeptpapier dargelegt werden. Idealerweise werden diese weiter ausdefiniert und somit ist eine grobe Einschätzung des Bedarfs an Auswertungswerkzeugen sowie Daten schon im Voraus vorhanden.	
2. Lösungsweg Gemeinsam mit den Verbundpartnern werden Strukturen geschaffen, um eigene, domänen-spezifische Datensätze zu erstellen sowie bestehende Datenplattformen an die SPEAKER-Plattform anzubinden. Hierzu werden Datenkonnektoren entwickelt und Benchmarking Tools implementiert, die allen Anwendungsfällen zu Gute kommen.	
Arbeitsschritt 1: Datenkonnektoren Hier werden Konnektoren zu bestehenden sowie eigenen Datensätzen definiert und implementiert, um diese zu Trainings- sowie Evaluationszwecken zu nutzen.	56,2 PM
Arbeitsschritt 2: Multilingualität Es werden Konzepte zur Umsetzung von Mehrsprachlichkeit erarbeitet, welche eine separate Entwicklung für die unterschiedlichen Sprachen umgehen. Es wird die Eignung von sprachunabhängigen, semantischen Repräsentationen untersucht, welche dies ermöglichen.	12,2 PM
Arbeitsschritt 3: Benchmarking Tools Hier werden Werkzeuge entwickelt, welche es ermöglichen, die Leistungsfähigkeit der Module anhand von Objektiven und Subjektiven Maßen zu erheben. Dies dient der kontinuierlichen Erfassung der Leistungsfähigkeit, sowie dem Vergleich mit Technologie der Mitbewerber.	43,8 PM
Arbeitsschritt 4: Voice User Experience Es werden Methoden zur kontinuierlichen Erhebung der Nutzerzufriedenheit entwickelt. Die vorgeschlagene NLoop Methodik wird hierzu unter Berücksichtigung strengster Datenschutzrichtlinien umgesetzt. Des Weiteren werden Strategien zur Dialogführung hinsichtlich der Nutzerzufriedenheit optimiert.	66,6 PM
Personalaufwand gesamt (in PM)	178,8 PM
Ergebnis (Output): Sammlung an getesteten Datenkonnektoren, Benchmarkframeworks und Daten sowie multilinguale Erweiterungen geeigneter Module.	
3. Sonstige Keine	Ressourcen

AP-Nr.:6		Evaluation in Anwendungsfällen und Modulentwicklung								
Startdatum:		04/2020								
Enddatum:		03/2023								
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	10,0	24,4	105,0	16,8	0,0	9,0	9,0	14,0	44,5	0,0
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	0,0	0,0	13,0	17,0	10,0	22,0	27,0	4,5	29,0	355,2
Ziel ist die Umsetzung erster Pilotanwendungen auf der SPEAKER-Plattform, welche in das Anwendungsfeld Protokollierungsaufgaben fällt. Wie im Konzeptpapier dargelegt werden in Kooperation mit folgenden Verbundpartnern Pilotanwendungen umgesetzt. CS: Sprachbasiertes Gesundheitslogbuch, DB: Sprachassistent zur Schadensprotokollierung, GBS: Intelligentes Diktiersysteme, SAP: Live Transkription von Events, SV: Sprachassistent Unternehmenssoftware, SE: Machine Assistant App, TUB: Audiovisuelle Spracherkennung, UPB: Offline Question Answering.										
1. Voraussetzung (Input) Verfügbarkeit der SPEAKER Minimum Viable Plattform (MVP) mit allen für die Sprachassistentanwendungen benötigten Modulen. Für Verbundpartner, deren Anwendungsentwicklung										

schon vor dem MVP starten soll, sind vorab ausgewählte Module auch ohne Cloud-Zugriff bereitzustellen.		
2. Lösungsweg Die Verbundpartner werden durch Fraunhofer IAIS und Fraunhofer IIS bei der Implementierung ihrer Pilotanwendungen mit Hilfe der SPEAKER-Plattform unterstützt. Weitere Verbundpartner, die Module beisteuern (z.B. die Uni Paderborn), werden hier für spezifische Anwendungsfälle Support anbieten.		
Arbeitsschritt 1:	Cloud basierte Emotionserkennung Hier entwickelt AudeERING eine Technologie-Komponente zur Emotionserkennung von Sprechern als Online-Modul auf der SPEAKER-Plattform.	6 PM
Arbeitsschritt 2:	Hybride Emotionserkennung AudeERING wird die Technologie-Komponente zur Online Emotionserkennung um eine Offline-Funktionalität erweitern.	4 PM
Arbeitsschritt 3:	Sprachbasiertes Gesundheitslogbuch CommaSoft erstellt eine Pilotanwendung, mit welcher Patienten ein Gesundheitslogbuch mit Hilfe eines Sprachassistenten führen können.	24,4 PM
Arbeitsschritt 4: Kommunikation mit Steuerberatern und Rechtsanwälten Evaluierung vorhandener Technologien für die jeweiligen Pilotanwendungen durch die DATEV e.G.		105 PM
Arbeitsschritt 5:	Sprachassistent zur Schadensprotokollierung Die Deutsche Bahn wird einen Sprachassistenten zur Vereinfachung von Inspektions-/Befundungsprozessen in der Instandhaltung und dem Betrieb implementieren.	17,8 PM
Arbeitsschritt 6:	Diktiersystem Grundig Business Systems entwickelt eine Anwendung, mit der in geführten Gesprächssituationen mit mehr als einem Sprecher eine Dokumentation der Sprecher, der Inhalte und der sprachlichen Ableitungen (teil-)automatisch erstellt werden kann.	14 PM
Arbeitsschritt 7:	Geräteübergreifender Sprachassistent Der iAV entwickelt mit unserer Unterstützung einen geräteübergreifenden Sprachassistenten. Ziel ist es einen messereifen Prototyp zu Demonstrationszwecken zu entwickeln.	20,5 PM
Arbeitsschritt 8:	Sprachassistent Produktentwicklung Automotive Der iAV entwickelt eine Initialversion eines Assistenten basierend auf der SPEAKER-Plattform, welcher es ermöglicht, u.A. bei Probefahrten Kommentare zu erfassen und Messdaten zu kommentieren.	24 PM
Arbeitsschritt 9:	Sprachassistent Hotellerie ONSEI entwirft einen Sprachassistenten zum Einsatz in Hotelzimmern, über welchem u.A. Informationen abgerufen werden können oder die Smart-Home-Dienste gesteuert werden können.	16 PM
Arbeitsschritt 10:	NLG Komponente Retresco wird ein Modul und einen entsprechenden Anwendungsfall zur automatischen Textgenerierung auf der SPEAKER-Plattform entwickeln, welches den NLG-Core um die kontextabhängige Generierung von Referenzketten, sowie Intent-bezogener Features erweitert.	17 PM
Arbeitsschritt 11:	Live Transkription von Events Die SAP SE entwickelt mit Hilfe der SPEAKER-Plattform eine simultane, automatische Transkription und Textanzeige während Veranstaltungen als Beitrag zu mehr Barrierefreiheit.	5 PM
Arbeitsschritt 12:	Voice Input für Bot Building Plattform Die SAP SE entwickelt mit Hilfe der SPEAKER-Plattform eine komplementäre Spracherkennung im Kontext einer Plattform für den Bau von Bots.	5 PM

Arbeitsschritt 13: Sprachassistent Unternehmenssoftware Die ScopeVisio AG erarbeitet einen Sprachassistenten, welcher es ermöglicht ihre Softwareprodukte per Spracheingabe zu steuern.	22 PM
Arbeitsschritt 14: Machine Assistant App Die SIEMENS AG entwickelt ein Sprachassistenzsystem, das den Nutzer einer Maschine beim operativen Betrieb bzw. den Servicetechniker bei der Wartung einer Maschine unterstützen soll.	27 PM
Arbeitsschritt 15: Sprachdatenbankkuration Dieser Arbeitsschritt ist hinfällig, da die korrespondierenden Arbeitsschritte unter AP5 und AP3 veranlagt wurden.	0 PM
Arbeitsschritt 16: Multimodale Spracherkennung Erstellung und Aufbau eines Echtzeitdemonstrators der audiovisuellen Spracherkennung für die Anwendung in einem autonomen Kleinbus, wobei die Auskunftsfunktion des Busfahrers durch einen SPEAKER-basierten Sprachassistenten zur Verfügung gestellt wird.	4,5 PM
Arbeitsschritt 17: Offline Question Answering Iterative Verbesserung der Anwendung	25 PM
Arbeitsschritt 18: Unterstützung der Unternehmen bei der Implementierung der Pilotanwendungen Fraunhofer IAIS und IIS werden als Hauptmodulentwickler die jeweiligen Verbundpartner bedarfsgerecht bei der Implementierung der Pilotanwendungen unterstützen.	18 PM
Personalaufwand gesamt (in PM)	355,2 PM
Ergebnis (Output): Technologiekomponenten auf der SPEAKER-Plattform, welche produktiv nutzbar und in Anwendungen erprobt sind. Spezialisierte Technologiemodule, welche auch mit Fachvokabular umgehen können. Erprobte Pilotanwendungen aus den verschiedenen Anwendungsfeldern.	
3. Sonstige Ressourcen Pro Verbundpartner individuelle Anschaffungen von Hardware für Demonstrationszwecke.	

AP-Nr.:7	Projektmanagement und Öffentlichkeitsarbeit									
Startdatum:	04/2020									
Enddatum:	03/2023									
Partner	AE	CS	DV	DB	DFKI	IAIS	IIS	GBS	IAV	IDSA
PM	1,0	3,0	1,0	8,4	1,0	12,0	6,0	1,0	17,6	1,7
Partner	iO	LMU	OE	RC	SAP	SV	SE	TUB	UPB	Σ PM
PM	6,0	0,0	1,0	7,0	3,0	4,0	0,0	1,5	1,0	76,2
Ziel des Arbeitspaketes: Das Projektmanagement umfasst folgende Tätigkeiten: Zunächst werden wissenschaftliche Forschungen auf dem Gebiet der Sprachassistenten (Publikationen, Open-Source, Konferenzen, usw.) regelmäßig einem Monitoring unterzogen, so dass auf aktuelle wissenschaftliche Entwicklungen sofort reagiert werden kann. Selbiges gilt für industrielle Entwicklungen, die über das Monitoring von Presse, Fachtagungen und Veranstaltungen verfolgt werden. Wichtige Informationen werden den Projektpartnern umgehend bereitgestellt. Die Erarbeitung und Umsetzung von rechtlichen Rahmenbedingungen stellen eine weitere Tätigkeit des APs dar. Dazu werden Kontakte und Austausch zu Rechtsexperten und Datenschutzbeauftragten etabliert. Darüber hinaus führt das Projektmanagement folgende Aufgaben durch: Berichtswesen, Durchführung von regelmäßigen Meetings, Einberufung der Beiräte, Konfliktmanagement, Qualitätsmanagement, Einhaltung von Terminen und Fristen, usw., um einen reibungslosen Projektablauf zu gewährleisten.										
1. Voraussetzung (Input)										

Keine	
2. Lösungsweg Durch Einsatz bewährter Projektmanagement-Tools und -Methoden erfolgt die Organisation von Berichtswesen, Durchführung von regelmäßigen Treffen, Konfliktmanagement, Qualitätsmanagement und Terminmanagement. Außerdem wird eine kontinuierliche Beobachtung wissenschaftlicher und industrieller Entwicklungen und Fortschritte auf dem Gebiet der Sprachassistenten etabliert. Alle Verbundpartner werden vom Konsortialführer bei der Umsetzung von rechtlichen Rahmenbedingungen unterstützt.	
Arbeitsschritt 1: Etablierung und Durchführung von Monitoring-Prozessen Überwachung des Fortschritts im SPEAKER-Projekt bzgl. Meilensteinen, Deliverables und Zuarbeiten aller Beteiligten. Daneben erfolgt ein wissenschaftliches Monitoring auf dem Gebiet der Sprachassistenten (Publikationen, Open-Source, Konferenzbesuche).	7,5 PM
Arbeitsschritt 2: Rechtliche Rahmenbedingungen Erarbeiten der rechtlichen Richtlinien und Bedingungen für Lizenz-Fragen und Datenschutz.	7,1 PM
Arbeitsschritt 3: Reporting Verfassen von Dokumentationen sowie Zwischenberichten und dem Abschlussbericht. Begleitend zu den 6 Meilensteinen des Teilvorhabens werden Zwischenberichte zum Projektfortschritt verfasst, das Schreiben der Zwischenberichte der Partner koordiniert und die Gesamtberichte mit Beiträgen aller Verbundpartner abgeglichen. Zum Ende der Umsetzungsphase wird ein Abschlussbericht erstellt. Des Weiteren dient dieses AP dem Reporting gegenüber dem Projektträger und Ministerium sowie der Koordination des Projektes nach innen, inklusive Problemvermeidungen und Konfliktlösungen. Durch die Pflege eines online Projekttagbuches, bspw. Confluence, werden aktuelle Arbeiten nachgehalten.	61,6 PM
Personalaufwand gesamt (in PM)	76,2 PM
Ergebnis Technologie- und Trendradar (Industrie und Wissenschaft); Rechtliche Richtlinien und Rahmenbedingungen; Projektreports	
3. Sonstige Unterbeauftragung zur Patentrecherche und Patentprüfung	
Ressourcen	

Anhang C Social-Media-Aktivitäten

Twitter Kampagne 27.01.2021 – 14.12.2021

Unternehmen	Zitatgeber	Veröffentli- chung
Fraunhofer IAIS	Prof. Dr. Jens Lehmann	27.01.2021
Fraunhofer IIS	Dr. Alessandra Zarcone	10.02.2021
Grundig Business Systems GmbH	Roland Hollstein	24.02.2021
Fraunhofer IIS	Dr. Birgit Brüggemeier	10.03.2021
DATEV eG	Prof. Dr. Christian Bär	17.03.2021
Fraunhofer IAIS	Maria del Mar (Guzman) Zumaya	24.03.2021
Deutsche Bahn AG	Dr. Miroslav Obrenovic	07.04.2021
Fraunhofer IAIS	Dr. Diego Collarana	23.04.2021
Fraunhofer IIS	Dr. Johannes Fischer	05.05.2021
DB Systel GmbH	Dr. Julia Hölterhoff	19.05.2021
Grundig Business Systems GmbH	Simon Verdenhalven	02.06.2021
ONSEI GmbH	Jochen Emig	16.06.2021
Retresco GmbH	Alexander Siebert	30.06.2021
Fraunhofer IAIS	Ulrike Stiefelhagen	14.07.2021
Deutsche Bahn AG	Philipp Niemann	28.07.2021
DB Fahrzeuginstandhaltung GmbH	André Obiora	11.08.2021
IAV GmbH	Matthias Kratzsch	25.08.2021
audEERING GmbH	Felix Burkhardt	08.09.2021
Cloud&Heat Technologies GmbH	Dr. Ronny Reinhardt	14.12.2021

Grundig Business Systems
GmbH, Roland Hollstein,
24.02.2021



Fraunhofer IAIS @FraunhoferIAIS · 24. Feb. 2021

...

Austausch & Vernetzung rund um **#KI** und Sprachtechnologie im Unternehmen – Roland Hollstein, **#Grundig Business Systems**, setzt im **#SPEAKERprojekt** auf die strategische Zusammenarbeit zwischen Partnern.

Infos für potenzielle Projektpartner: speaker.fraunhofer.de
#ConversationalAI



DATEV eG, Prof. Dr.
Christian Bär, 17.03.2021



Fraunhofer IIS @FraunhoferIIS · 17. März 2021

...

Prof. Christian Bär, Chief Digital Officer bei **#DATEV**, sieht im **#SPEAKERprojekt** großes Potential, »DIE Sprachassistentenplattform für Unternehmen zu entwickeln, die mit Sicherheit Mehrwert schafft.«

Mehr Infos: speaker.fraunhofer.de **#ConversationalAI**



Deutsche Bahn AG, Dr.
Miroslav Obrenovic,
07.04.2021



Fraunhofer IIS @FraunhoferIIS · 7. Apr. 2021

...

Sprachassistenten für digitale Instandhaltung – die **#DeutscheBahn AG** treibt im **#SPEAKERprojekt** die **#Digitalisierung** ihrer Prozesse voran.

Mehr erfahren: speaker-projekt.de
#ConversationalAI #KIWettbewerb @DB_Presse



DB Systel GmbH, Dr. Julia
Hölterhoff, 19.05.2021



Fraunhofer IAIS @FraunhoferIAIS · 19. Mai 2021

...

Sprachassistenten erleichtern Arbeitsabläufe – Dr. Julia Hölterhoff von @dbsystel über die Partnerschaft im #SPEAKERprojekt.

SPEAKER bietet Unternehmen die Möglichkeit, spezifische Anforderungen an ein natürlichsprachliches Assistenzsystem zu realisieren speaker-projekt.de



Grundig Business Systems
GmbH, Simon Verdenhal-
ven, 02.06.2021



Fraunhofer IIS @FraunhoferIIS · 2. Juni 2021

...

Wie können Unternehmen von der Sprachassistentenplattform #SPEAKERprojekt profitieren?

Simon Verdenhalven, #Grundig Business Systems, setzt den Fokus auf #Datenschutz und effiziente Prozesse für Unternehmen sowie Kundinnen und Kunden.

#ConversationalAI #KIWettbewerb



ONSEI GmbH, Jochen
Emig, 16.06.2021



Fraunhofer IAIS @FraunhoferIAIS · 16. Juni 2021

...

Natürlich, effektiv, intuitiv – im **#SPEAKERprojekt** entstehen neue **#DSGVO**-konforme Voice-Technologien. @ONSEI_Berlin-Geschäftsführer Jochen Emig @jsemig freut sich, sie gemeinsam mit Partnern aus Forschung & Wirtschaft voranzubringen. speaker.fraunhofer.de

#ConversationalAI



Retresco GmbH, Alexan-
der Siebert, 30.06.2021



Fraunhofer IIS @FraunhoferIIS · 30. Juni 2021

...

Wie gelingt im **#SPEAKERprojekt** der Transfer von der Forschung in die Praxis? retresco-Geschäftsführer Alexander Siebert arbeitet Hand in Hand mit @FraunhoferIAIS, #FraunhoferIIS & namhaften Partnern an einer branchenübergreifenden Sprachassistentenplattform. **#ConversationalAI**



Deutsche Bahn AG, Philipp
Niemann,
28.07.2021

Fraunhofer IIS @FraunhoferIIS · 28. Juli 2021
Digitale Instandhaltung bei @DB_Bahn – mit Hilfe des **#SPEAKERProjekt**s werden Informationen schneller geteilt, branchenübergreifend entwickelt, made in **#Germany**. Das steht für Projektleiter Philipp Niemann im Fokus. Mehr erfahren 🗣️ speaker-projekt.de **#ConversationalAI**



DB Fahrzeuginstandhaltung GmbH, André Obiora,
11.08.2021

Fraunhofer IAIS @FraunhoferIAIS · 11. Aug. 2021
Ein neues Level der Effizienz zu erreichen, ist Ziel von André Obiora & der DB Fahrzeuginstandhaltung GmbH @DB_Bahn beim **#SPEAKERprojekt**.
Interesse an einer assoziierten Partnerschaft mit SPEAKER? Infos und Kontaktmöglichkeiten: speaker.fraunhofer.de **#ConversationalAI**



IAV GmbH, Matthias
Kratzsch, 25.08.2021

FraunhoferIIS_en @FraunhoferIIS_e · 25. Aug. 2021
Robustheit, Performance, Datensicherheit – in der Fahrzeugentwicklung sind **#Sprachassistenten** gefragt & die Anforderungen hoch. Im **#SPEAKERprojekt** arbeiten Expert*innen aus Forschung & Wirtschaft wie IAV_de-CEO Matthias Kratzsch an der Umsetzung.
Infos: speaker-projekt.de



audEERING GmbH, Felix
Burkhardt, 08.09.2021



Fraunhofer IAIS @FraunhoferIAIS · 8. Sep. 2021

...

Im **#SPEAKERprojekt** entstehen intelligente **#Sprachassistenten** »made in Germany« mit **#Datenschutz** an erster Stelle. Felix Burkhardt @audEERING bringt im Projekt sein Fachwissen im Bereich Emotionserkennung mit Audio-**#KI** ein.

Infos: speaker-projekt.de

#ConversationalAI



Cloud&Heat Technologies GmbH, Dr. Ronny
Reinhardt, 14.12.2021



Fraunhofer IAIS @FraunhoferIAIS · 14. Dez. 2021

...

Das **#SPEAKERprojekt** soll die hohen B2B-Anforderungen an **#Sprachassistenten** erfüllen und die entsprechende Infrastruktur, Technologien & Standards der Industrie & öffentlichen Einrichtungen zur Verfügung stellen. Dr. Ronny Reinhardt erklärt, welchen Beitrag @CLOUDandHEAT leistet.



🗨 1

🔄 2

❤ 3



Anhang D Nutzerdokumentation Dialog Editor v3



Dialog Editor v3

Inhaltsverzeichnis

1	Preface	4
2	What is the Dialog Editor?	4
3	How can I access the Dialog Editor V3?	6
3.1	<i>Where can I find the Dialog Editor?</i>	6
3.2	<i>How can I get my login data for the Dialog Editor?</i>	6
3.3	<i>How to use the Dialog Editor?</i>	7
3.4	<i>I found a bug, what should I do?</i>	7
3.5	<i>I have further questions, comments, or suggestions. What should I do?</i>	8
4	1. Create Projects	9
4.1	<i>Project dashboard</i>	9
4.2	<i>Create new project</i>	10
4.3	<i>Edit project</i>	11
4.4	<i>Save a local copy</i>	11
4.5	<i>Import project</i>	11
5	2. Design Dialogs	12
5.1	<i>Canvas Basics</i>	12
5.1.1	<i>Introduction to canvas</i>	12
5.1.2	<i>Adding nodes</i>	14
5.1.3	<i>Node states</i>	15
5.1.4	<i>Configuring nodes</i>	17
5.1.5	<i>Displaying complete prompt text in canvas view</i>	17
5.1.6	<i>Adding and deleting connectors</i>	17
5.1.7	<i>Reordering intents</i>	18
5.1.8	<i>Resizing configuration dialogs</i>	19
5.2	<i>Domains</i>	20
5.2.1	<i>What is a Domain?</i>	20
5.2.2	<i>Create Domain</i>	21
5.2.3	<i>Open Domain</i>	24
5.3	<i>Global Settings</i>	25
5.3.1	<i>Global prompts</i>	26
5.3.2	<i>Global error handling</i>	27
5.4	<i>Nodes</i>	27
5.4.4	<i>Decision Node</i>	32
5.4.5	<i>Action Node</i>	36
5.4.6	<i>System Utterance Node</i>	38
5.4.7	<i>Variable Node</i>	39
5.4.10	<i>API Request Node</i>	45
5.4.11	<i>Go To Idle State Node</i>	46
5.5	<i>Management</i>	46
5.5.1	<i>Domain Management</i>	47

5.5.2	Action Management	48
5.5.3	Entity Management	49
5.5.4	Variable Management	53
6	3. Track Issues.....	57
6.1	<i>Start issue tracking</i>	57
6.2	<i>Issue tracking mode</i>	58
6.3	<i>Issue panel</i>	58
6.3.1	Issue types	59
6.4	<i>Stop issue tracking</i>	59
7	4. Build and Test Assistants	60
7.1	<i>Set up build</i>	60
7.3	<i>Build job</i>	63
7.3.1	Successful build	64
7.3.2	Failed build	64
8	Explore Dialog Editor Templates	65

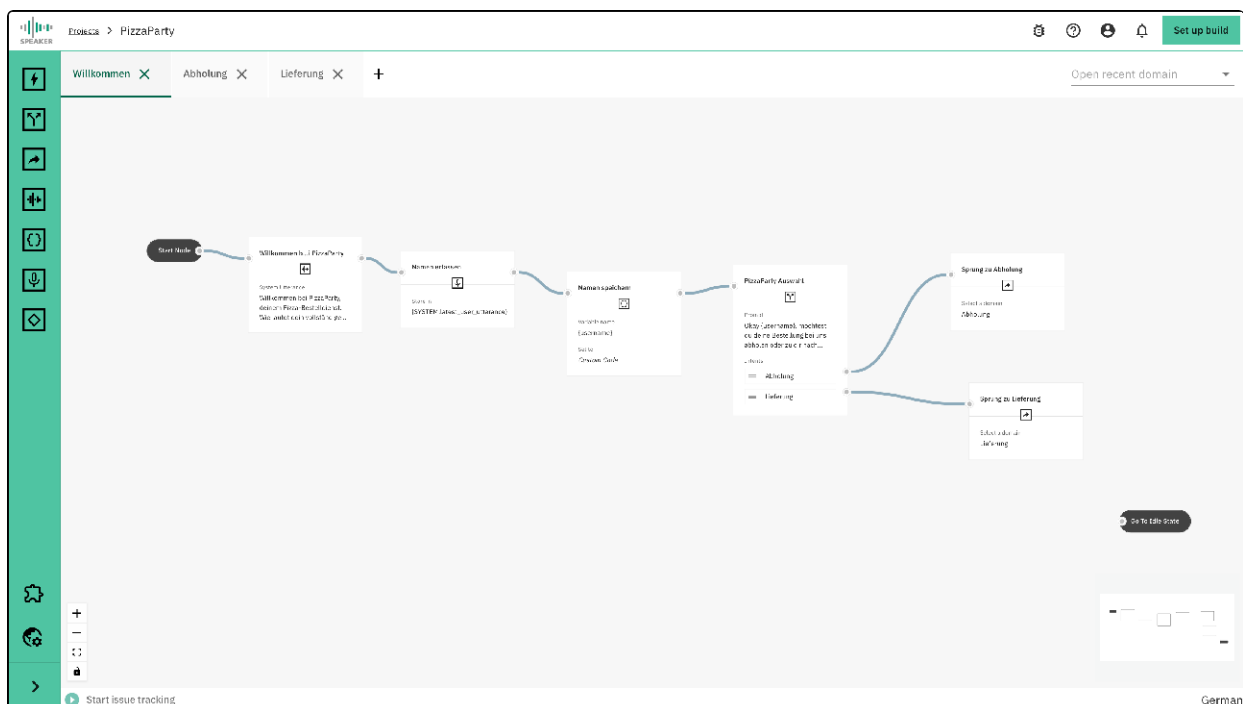
1 Preface

The following document was provided to the users of the SPEAKER Dialog Editor as an online documentation and was optimized for that form of representation.

It was exported and placed in the attachment of the SPEAKER final report only to give an overview over the structure and the scope of the Dialog Editor. Thus, the structure, look and feel of this document as a printed version is not on a par with its online equivalent. As this excerpt is only in the attachments of the SPEAKER final reports for documentation purposes, the efforts for improving the layout for a printed format were limited to a reasonable extend.

2 What is the Dialog Editor?

The Dialog Editor is a low-code tool that enables conversational designers to prototype and implement voice experiences. The current version allows you to explore the features of the Dialog Editor and to get familiar with the operation of the system. You have access to all the functionalities that are required to design, build, and test dialogs. The further development of the Dialog Editor is a work in progress, as new features and system improvements are added with every new release. Our goal is to meet all the needs inspired by your use cases. So, we'll stay in touch with you!



This documentation is put together to guide you through every step of the process, from getting started, to understanding the basics, to mastering the advanced features, so that you can create dialogs that are tailored to your specific needs.

But first, for newbies and those needing a reminder, [How can I access the Dialog Editor?](#) (see page 7)

Learn step by step how to: [create projects](#) (see page 10), [design dialogs](#) (see page 13), [track issues](#) (see page 58) and finally how to [build and test assistants](#) (see page 61). You can also select your topic of interest directly on the left sidebar.

- The Dialog Editor is continuously being improved, therefore this documentation is a living document. If you have questions, comments, or found a mistake, we would be grateful for your feedback. Please use our [Jira Service Desk](#)¹ for this. You can login with your regular Jira credentials.

¹ <https://jira.iais.fraunhofer.de/jira/servicedesk/customer/portal/9/user/login?destination=portal%2F9>

3 How can I access the Dialog Editor V3?

- [Where can I find the Dialog Editor? \(see page 7\)](#)
 - [How can I get the login credentials for the Dialog Editor? \(see page 0\)](#)
 - [How to use the Dialog Editor? \(see page 8\)](#)
 - [I found a bug, what should I do? \(see page 8\)](#)
 - [I have further questions, comments, or suggestions. What should I do? \(see page 9\)](#)
-

3.1 Where can I find the Dialog Editor?

The Dialog Editor is a web application which can be accessed via [this link](#)². Please also note the information given in the box below.

- Please note that there are some limitations associated with this version of the Dialog Editor:

 - One account shouldn't be used simultaneously by multiple users under the same account. If more than one user logs in using the same account, it may lead to compromised functionality.
 - The Dialog Editor allows importing the created dialogs as JSON files and saving them locally on your device. We aim to make the file formats compatible with future versions of Dialog Editor, but cannot guarantee this.
 - Please also note that the Dialog Manager already supports more features than are configurable in this version of the Dialog Editor. We strive to make these features configurable via the Dialog Editor as soon as possible.

3.2 How can I get my login data for the Dialog Editor?

If you are already a SPEAKER partner you can find your login data on your [sub-page in Confluence](#)³. If you can no longer access them or have problems with the credentials, please [contact us](#)⁴.

² <https://dialog-editor.speaker-plattform.de/>

³ <https://jira.iais.fraunhofer.de/wiki/display/SPKVP/Anwendungspartner>

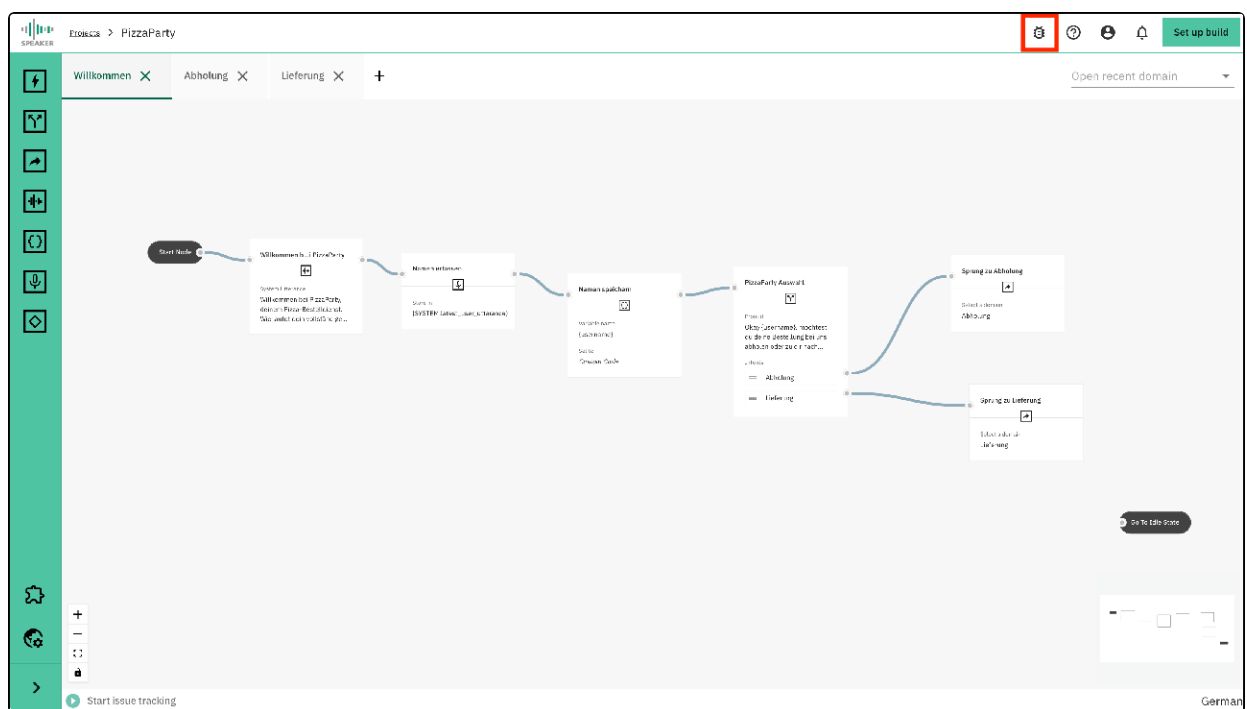
⁴ <mailto:platform-speech-assistant-request@iais.fraunhofer.de>

3.3 How to use the Dialog Editor?

With the Dialog Editor, you can design a voice experience by creating dialogue graphs and configuring the corresponding nodes accordingly. An overview of the interaction with the Dialog Editor, as well as more detailed information on its modules and functions can be found in further chapters of the [documentation](#) (see [page 5](#)).

3.4 I found a bug, what should I do?

Although we do our best to prevent bugs by extensive testing, we cannot exclude the possibility that a bug has crept in. Please inform us in case you encountered a problem. For that purpose, you can use our [Jira Service Desk](#)⁵. The service desk can also be reached from within the Dialog Editor. The feedback button is in the top-right corner (see screenshot). For logging into the Jira Service Desk, you can use your regular Jira credentials.



⁵ <https://jira.iais.fraunhofer.de/jira/servicedesk/customer/portal/9/user/login?destination=portal%2F9>

3.5 I have further questions, comments, or suggestions. What should I do?

We would also be thankful if you get in touch with us in case you have further questions, comments, or suggestions about the Dialog Editor. Please use the [Jira Service Desk](#)⁶ in these cases as well.

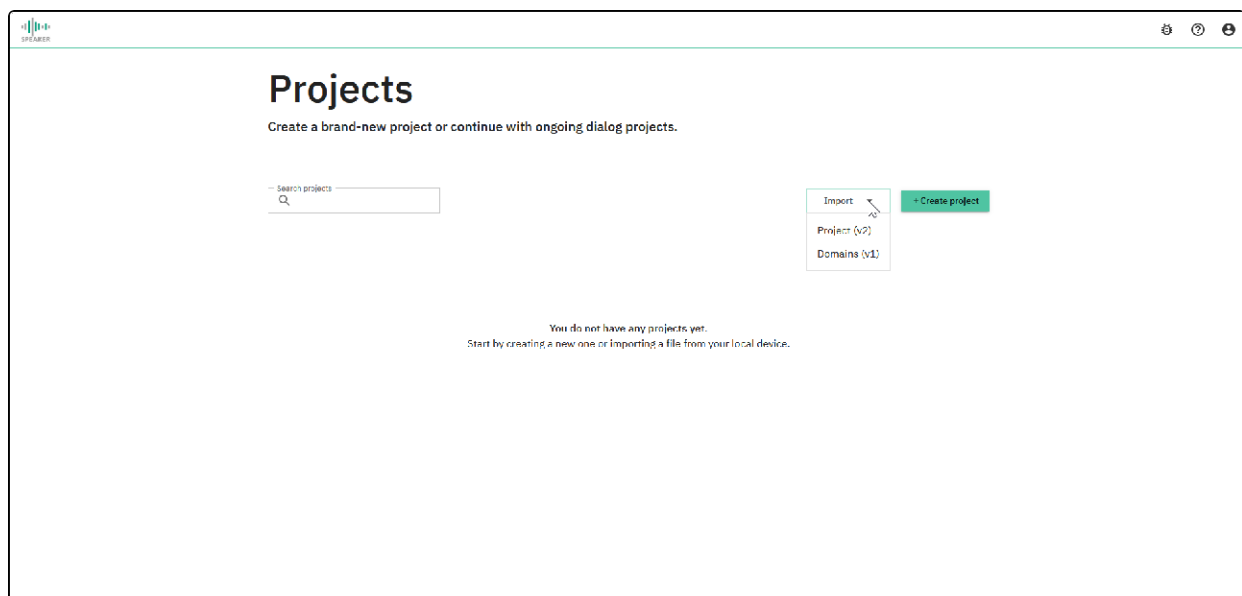
⁶ <https://jira.iais.fraunhofer.de/jira/servicedesk/customer/portal/9/user/login?destination=portal%2F9>

4 1. Create Projects

4.1 Project dashboard

After logging in, you will land on the project dashboard. Here, you can access projects that have been created by members of your user group. You also have the option to [create a new project](#) (see page 11) from scratch, or [import](#) (see page 12) one from your local device.

In the header, you also find links to your user profile settings and some helpful resources, i.e. the user documentation, release notes and our service desk.



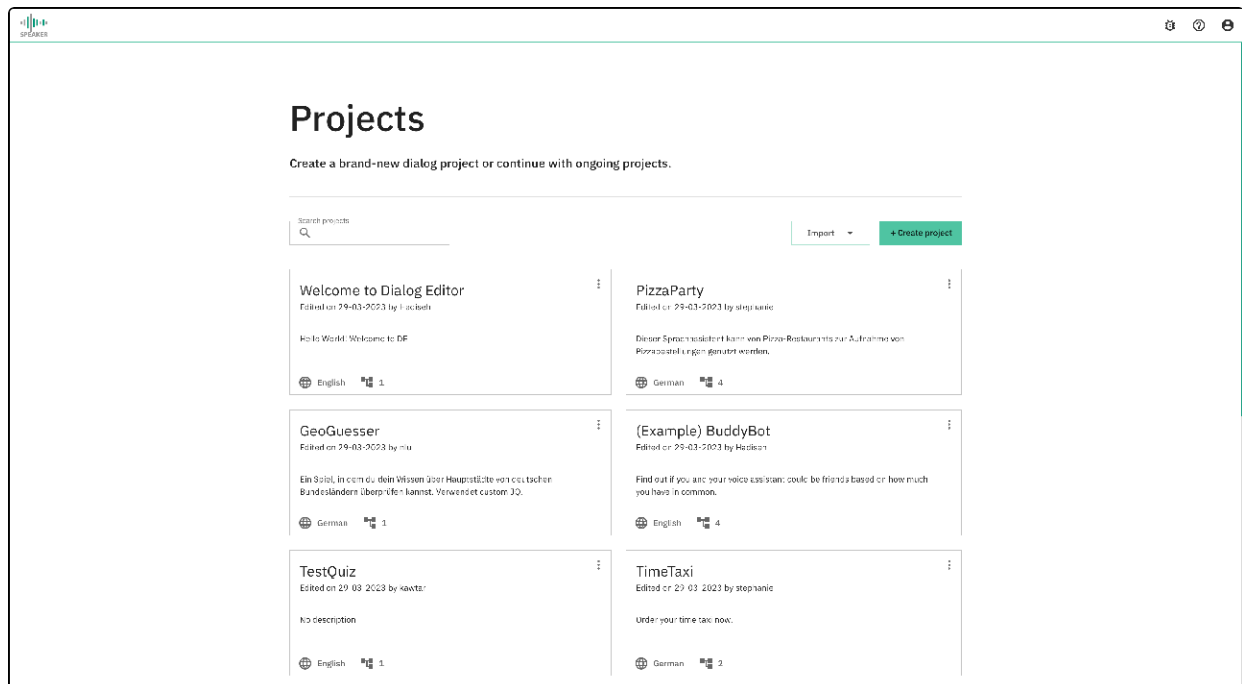
Project cards

In the project dashboard, projects are displayed as cards. They contain:

- the project title
- a timestamp for when the project was last edited and the user who did this last edit
- the project description
- the [assistant language](#) (see page 11)
- the number of [domains](#) (see page 21) within that project

- a menu with options to
 - [edit](#) (see page 12) the project settings
 - [save a local copy](#) (see page 12) of that project
 - delete a project

Clicking on the project card will open the project in the [canvas](#) (see page 13).



4.2 Create new project

You can create a new project by simply choosing a project title (required), a project description (optional, but recommended) and the assistant language. After confirming, a new [project card](#) (see page 10) will be added to the [project dashboard](#) (see page 10).

Assistant language

This is the language your voice assistant is going to understand. You decide here, on which language model you want to base your dialog on. This setting would also affect the [built-in entities](#) (see page 52). Especially before setting up a new build, you should make sure that the assistant language setting matches the language you used when defining your dialog, e.g. for prompts, intents, etc.

You can always see the current assistant language setting in the status bar as well.

4.3 Edit project

Via the menu on every [project card](#) (see page 10), you can edit the project title, the project description and the assistant language of your project.

4.4 Save a local copy

Via the menu on every [project card](#) (see page 10), you can save a local copy of your project. You have the option to deselect [domains](#) (see page 21) you want to exclude from the local copy, though it is recommended to include all the existing domains within a project to make sure that all configurations and links are preserved.

You can [import](#) (see page 12) this local copy of your project into the Dialog Editor again.

4.5 Import project

On the [project dashboard](#) (see page 10), you will find two options for importing a project:

Import domains from Dialog Editor V1

You can import JSON files containing domain-specific information for dialogs that have been created in Dialog Editor V1. For that, you upload a JSON file or multiple files at once, if you want to have them within one project afterwards. All the domains that you upload within that step will be combined into one project in the next step.

It is neither possible to add single domain files to an existing project, nor to merge projects.

After importing, you will be asked for a project title, description and the [assistant language](#) (see page 11) to [create a new project](#) (see page 11) out of your imported files.

This feature is going to be deprecated.

By the end of September 2023, this feature is going to be deprecated. Make sure that you have imported all your domains from Dialog Editor V1 by that time.

Import projects from current versions

You can import JSON files containing projects that have been created in current versions of the Dialog Editor. For that, you simply upload a JSON file containing your project data. After that, you will find your project added to the dashboard.

5 2. Design Dialogs

For learning the basics or gaining a deeper understanding of the Dialog Editor design process, you can check out:

[Canvas Basics](#) (see page 13)

[Domains](#) (see page 21)

[Global Settings](#) (see page 26)

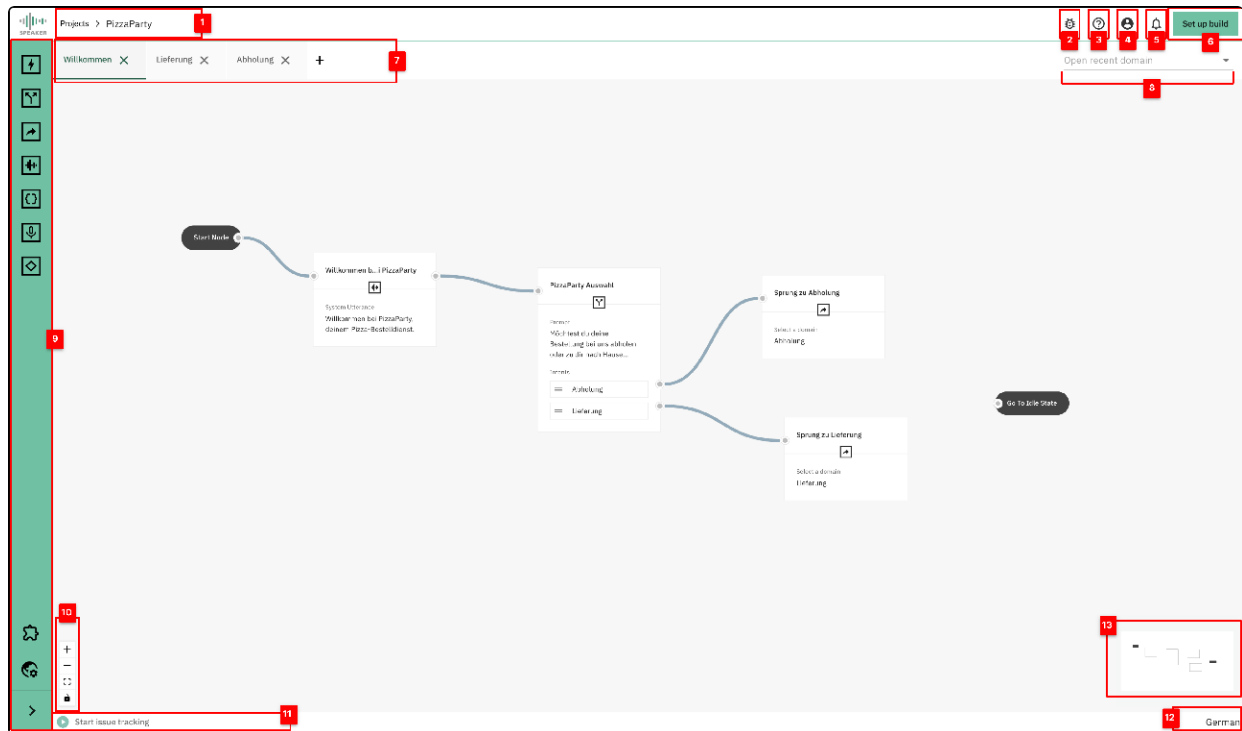
[Nodes](#) (see page 28)

[Management](#) (see page 47)

5.1 Canvas Basics

5.1.1 Introduction to canvas

You are able to create one dialog flow per domain by adding and connecting nodes from the node bar. In order to add a node drag & drop the node from the side bar on the left.

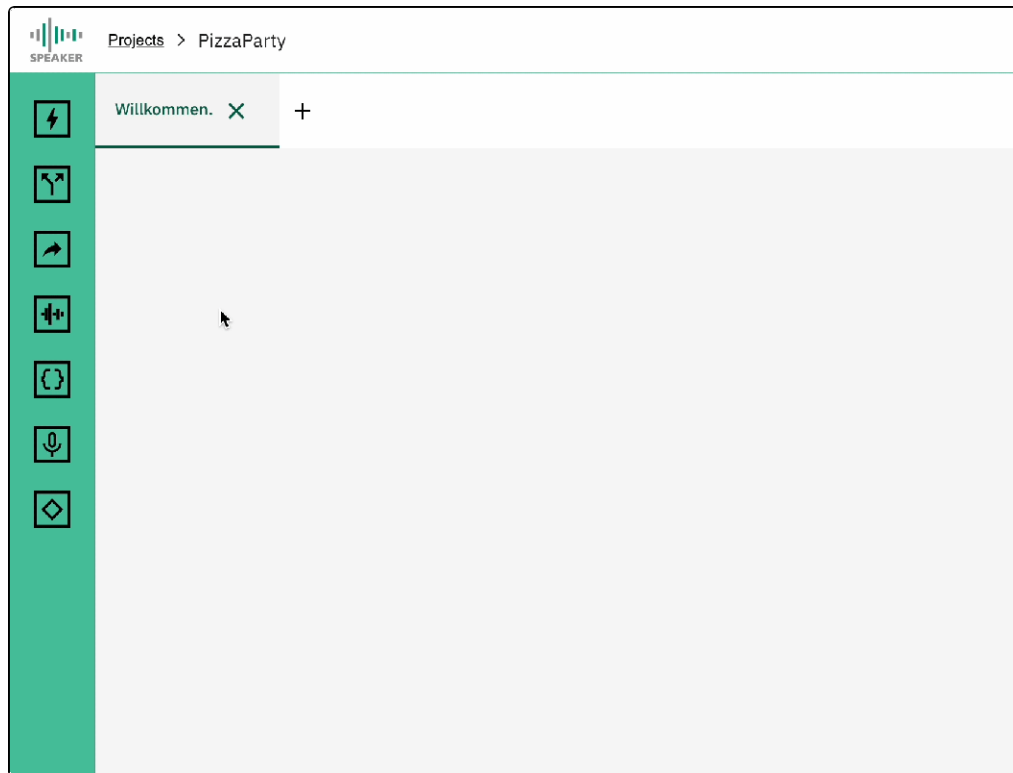


1. **Breadcrumb** → Path that shows the location of the current [project](#) (see page 10) you are in.
2. **Bug report** → Link to our bug report and support.
3. **Help** → Click on the "Help" icon to find the documentation of the Dialog Editor and release notes of the current version, imprint and legal terms.
4. **Profile** → By clicking on the "Profile" icon, you see a menu containing options to open the user profile settings and to logout. By clicking on "Profile settings", you will get the option to select or change your user avatar and to change the language of the Dialog Editor interface text (either to English or German). You can also logout of the session by clicking on the "Logout" button.
5. **Notification** → Click on the "Notification" icon to access your build history. If you see a green dot on this icon, it means that either a build job was completed or it failed.
(see page 61)
6. **Set up build button** → Clicking on this button will start creating a build of your assistant. [Read about building assistants](#) (see page 61).
7. **Domain tabs** → The currently open domains are displayed in the tabs. You can switch between the domains by clicking on a tab. Click on the "hide" button, to close any domain from the open tabs. By hiding a domain you are not deleting it. You can also reopen it later or change the order of the tabs per drag & drop. You can create a new domain directly from the domain tab by clicking on the "create" icon (plus). An overlay called [Create domain](#) (see page 0) will be displayed over the current domain you are navigating.

8. **Search domains** → If you would like to open a recently used domain click on the "Open recent domain" drop-down. You can choose the domain from a drop-down list. You can type the name into the search field and the matching result will appear first in the list. If you cannot remember the name of the domain and it is not in the list, go to [Domain Management \(see page 48\)](#) [\(see page 0\)](#) to look for it in the list.
9. **Sidebar** → The side bar offers an overview of the available [nodes \(see page 28\)](#) and other dialogue configurations, namely [Global Settings \(see page 26\)](#) and [Management \(see page 47\)](#). The side bar is a dynamic element that can be expanded and collapsed in order to show detailed information or to increase the space in the canvas. By clicking on the arrow placed in the bottom, you can expand or collapse the bar.
10. **Canvas control** → You can use the two zoom buttons in the toolbar to zoom in and out your canvas. The "fit to screen" button adjusts the display size, so that the whole dialog flow (of the current domain) fits on the screen. Also, the screen can be locked, which means that nodes and paths on the canvas can't be moved anymore across all open domains.
11. **Issue tracking** → Here you can find a button to start [issue tracking \(see page 58\)](#).
12. **Assistant language** → It shows you the current setting for the assistant language, which can be changed if you go over the breadcrumbs back to the [project cards \(see page 10\)](#).
13. **Mini map** → This shows the user's position and zoom level of the current view.

5.1.2 Adding nodes

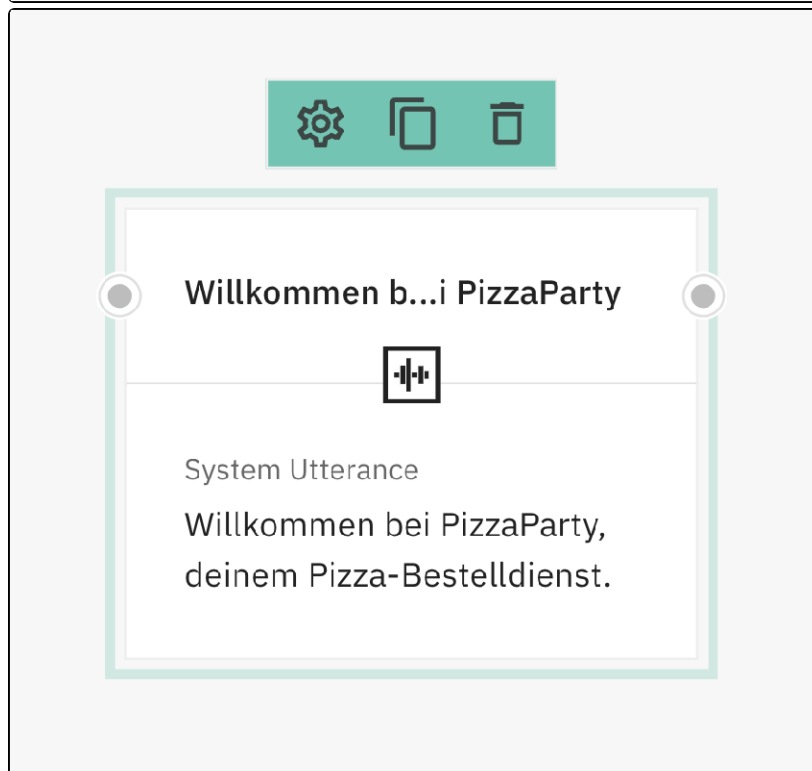
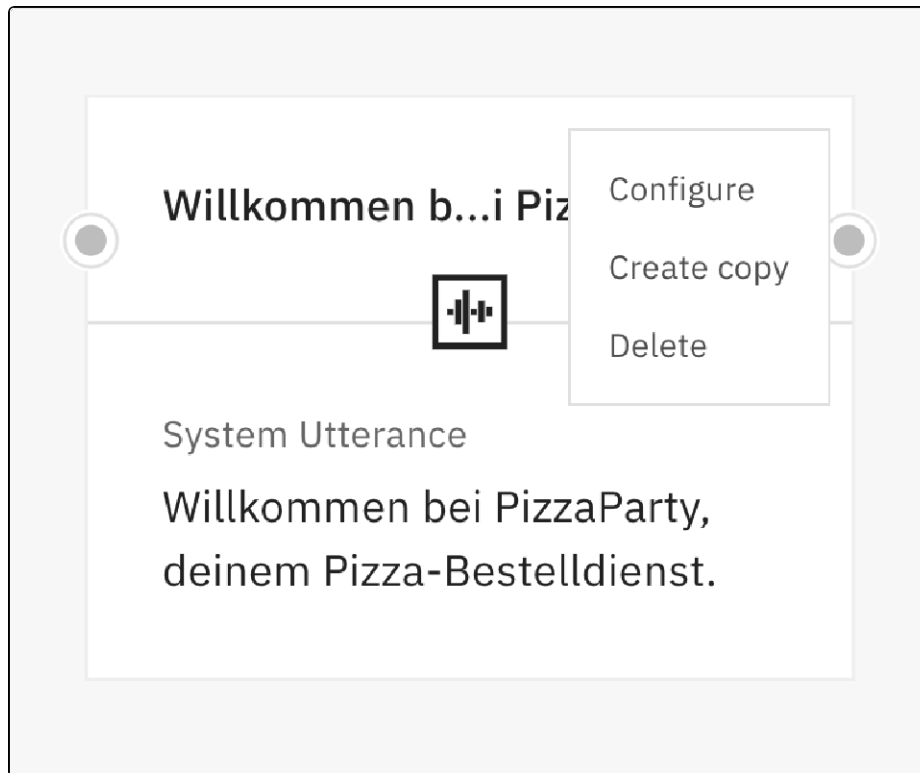
A node can be added to the dialog flow canvas by dragging & dropping the node into the canvas.



5.1.3 Node states

By (left) clicking on a node, a floating context menu appears and on top of the node there are three options to configure, copy or delete a node. By right clicking on a node surface, you'll see another context menu with same three options to configure, copy or delete a node.

The floating context menu can be dismissed when a node is being moved around on the canvas or when you click somewhere outside.

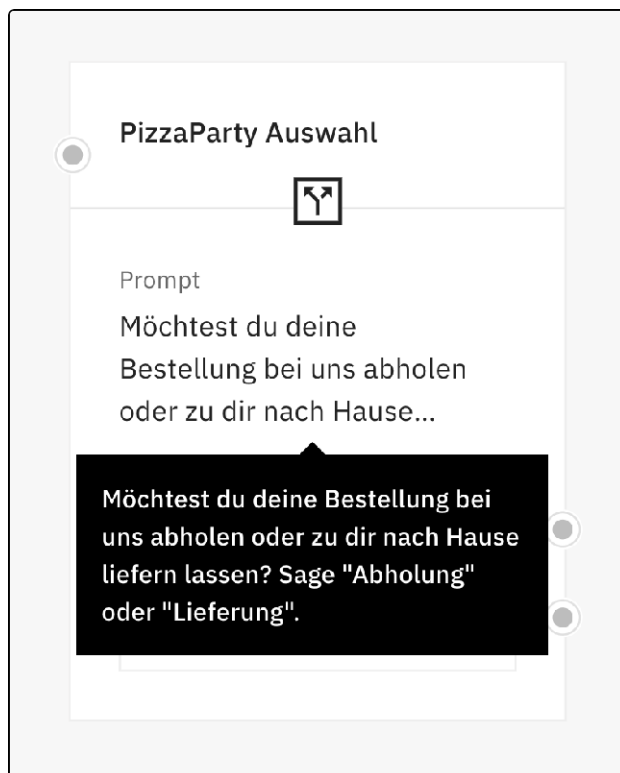


5.1.4 Configuring nodes

Each node needs to be configured in order for the system to work. To open the configuration menu of a node, double click on the surface of the node or use the floating context menu (see screenshot above). More detailed information for configuring each node type can be found in the [nodes section](#) (see page 28).

5.1.5 Displaying complete prompt text in canvas view

When navigating through the dialogue in the canvas view, it is helpful to read the prompt text and proof the consistency of the interaction. To view the complete prompt text hover over the prompt area. A tooltip will be displayed, this is specially useful for very long prompt text or for navigating the complete dialogue and not having to zoom in and out.

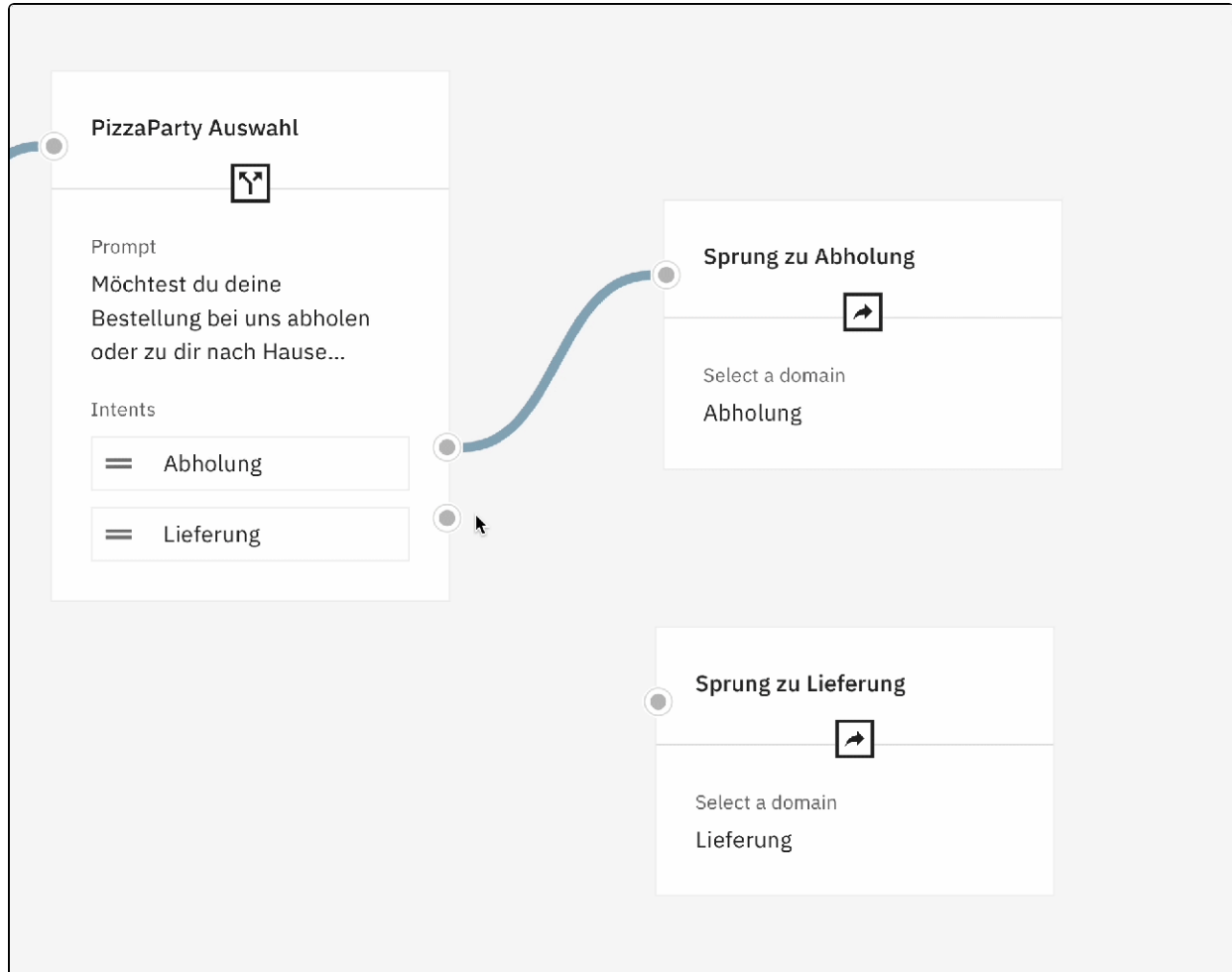


5.1.6 Adding and deleting connectors

Each node can be connected to at least one follow up node. This way it is defined how the user can navigate through the dialog. To achieve this, (left) click on one of the connector dots on the right edge of the first node and drag the connector to the desired follow-up node. Some nodes can also be linked to multiple follow-up

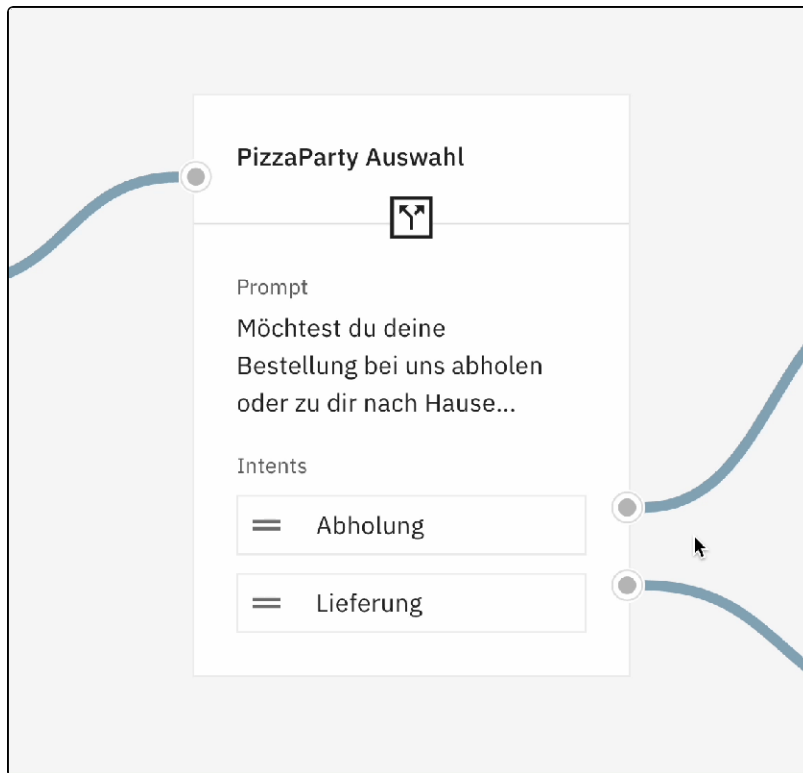
nodes (e.g. the decision node). If no follow-up node is specified, the dialog ends and the system returns to the initial state.

To **delete** a connector, click on the connector. When it turns green, press **fn + Backspace** on your keyboard.



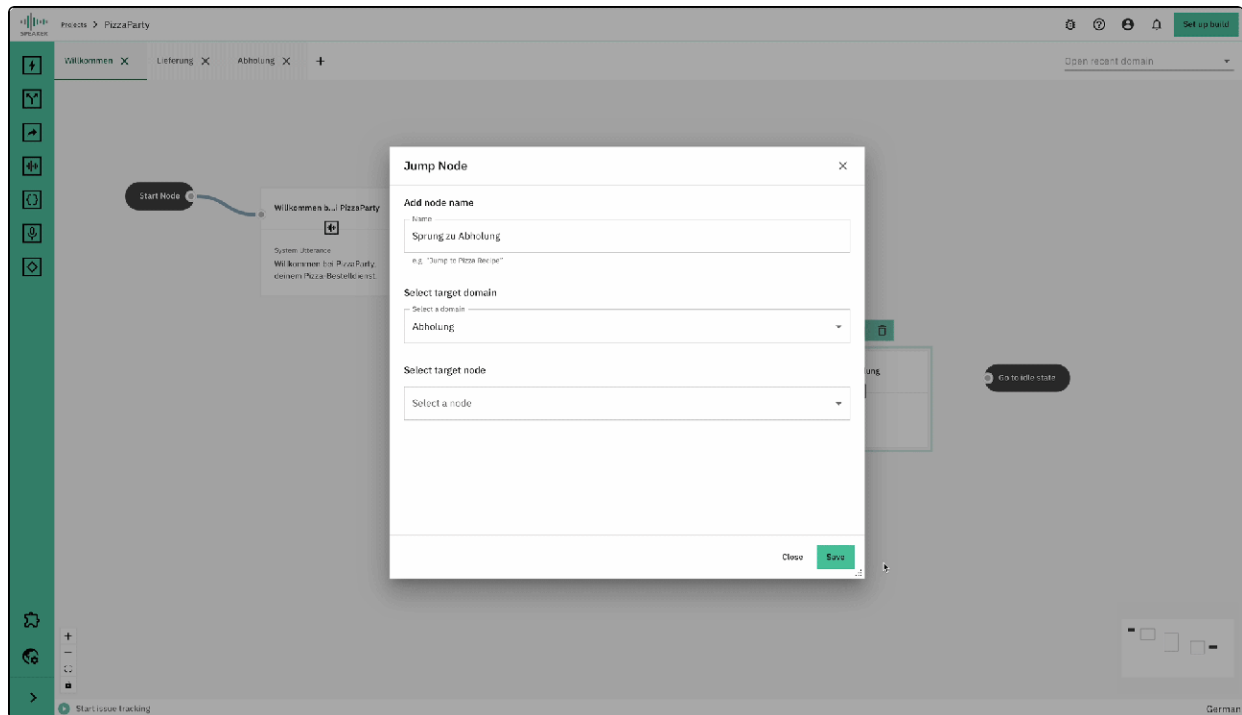
5.1.7 Reordering intents

It is possible to *reorder* the intent in the Decision Node. To start reordering the intents, click an intent and simply drag it up or down.



5.1.8 Resizing configuration dialogs

You can resize a configuration dialog by having the mouse pointer over the resize handles on the bottom right corner of the dialog and then click and drag the mouse.



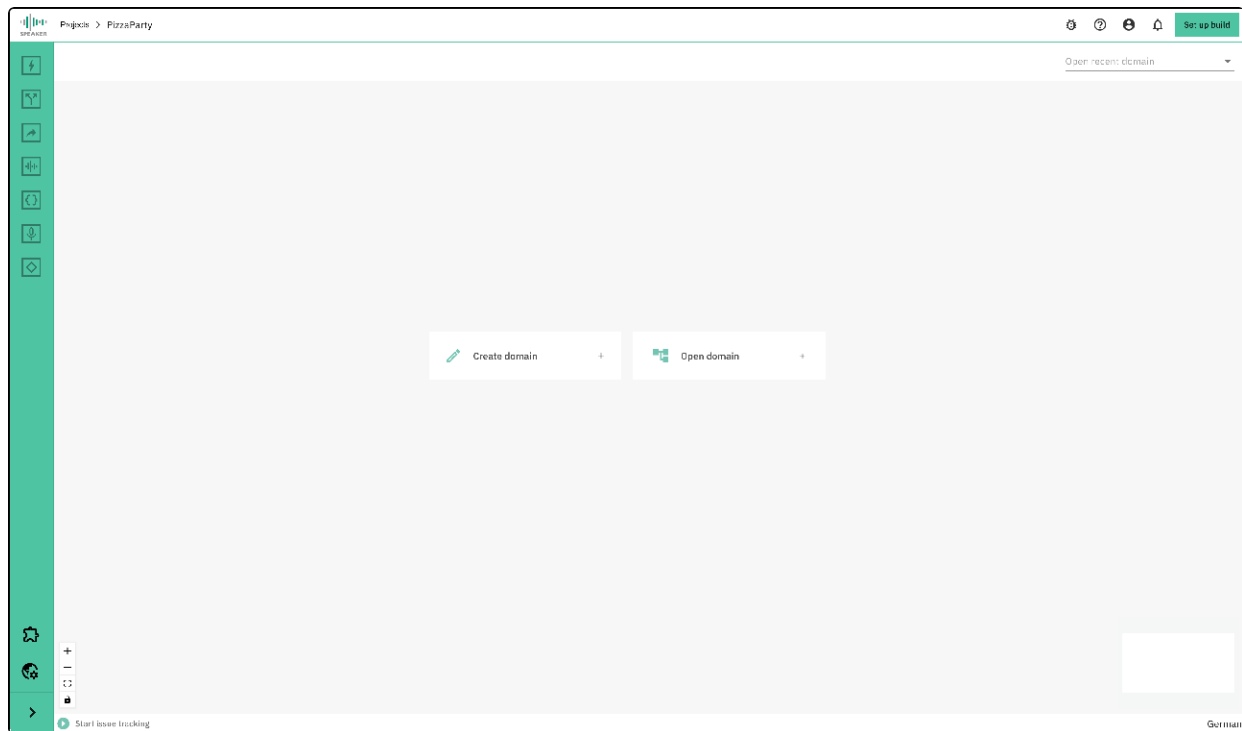
5.2 Domains

5.2.1 What is a Domain?

The dialog tree is divided into different domains. The domains are sub-dialogs, typically used for grouping related elements, for example of a specific topic or function (e.g. calendar). As an example, let us assume that the speech assistant can be used to order pizza. We could then create separate domains for "ordering pizza" and "pizza menu". The separation into different domains leads to a better overview of the dialog and also offers several technical advantages. Usually it makes sense to group thematically related elements under a domain. But from a technical perspective, it is also possible to combine arbitrary elements in one domain. It is important, however, that the domains are clearly separated from each other (e.g. no overlapping branches/nodes). With the so called [Domain Trigger](#) (see page 23) and [Jump Node](#) (see page 32) it is possible to switch to a specific domain. The dialog follows the paths contained within a domain as soon as this domain is activated. Only one domain can be active at a time. After the last node of a domain path has been traversed, the domain is exited. If the last node is a Jump Node, the system switches to the target that has been determined. Otherwise the system reaches the [Go To Idle State Node](#) (see page 0) and rests in the idle state until it can react to a new user utterance.

- The more nodes are included in a domain, the longer it will take to load the dialog. Therefore we recommend not to make the domains larger than needed.

After creating a project you'll be redirected to the canvas to create a new domain or open a domain.



5.2.2 Create Domain

You can create a domain by clicking on the Create domain card. If you have already created domains, you can add new ones by clicking on the plus symbol next to the [domain tabs](#) (see [page 13](#)). A Create domain configuration window will pop up.

Create domain

Add domain name

Name

e.g. "PizzaParty"

Select classifier of domain trigger

Intent classifier

Keywords

Add keywords for *

Press Enter to add variants, e.g. "Open PizzaParty"

Protect domain

Request confirmation before leaving the domain

Close

Submit

Configuration

- **Name:** Each domain needs a name that serves as a unique identifier (e.g. "Calendar").

i A **domain trigger** is used to activate the switch of domains. It allows you to enter a domain or switch from one domain to another, when the dialog reaches any decision node during the flow. You define trigger utterances as intents that are located in a different domain than the current one. For example if the user says open Calendar, while in a different domain, the system will capture the user utterance as an intent. If you want to take full control of how the system handles domain jumps, you can use * or other special characters (e.g. \$turn_tv_on\$ and \$call_taxi\$) as domain triggers, when setting up the domain. This will avoid the recognition of trigger utterances and make sure these domains are never directly triggered and can only be accessed by a jump node.

- **Classifiers of a domain trigger**
 - **Neural:** A neural domain trigger are suitable if an intent can be expressed in very different ways. With the right quantity and quality of training data (utterances), the algorithm can even

predict intents of user utterances that were not among the examples in the training data. It needs a lot of data to be trained. Once trained, they ideally generalize to unseen data.

- **Training utterances:** You can add training data for a neural classifier either manually or by uploading a `.txt` file containing the utterance examples. You should upload different files for the trigger of each domain and use enter, semicolon, | or a tab character to separate each training utterance. For example for a Calendar domain, you can add multiple training utterances such as: "Open my calendar", "Show me my calendar", "What's in my calendar?", "Check my appointments", "What does my schedule look like?" etc.
- **Confirmation sentence:** In case of choosing a neural classifier for domain triggers, you need to specify the "Confirmation sentence" as well. If the system has only medium confidence that it recognized the trigger correctly, it prompts the user to confirm their utterance. Example: for the category "order pizza", if the neural classifier has not clearly recognized a trigger, the system therefore responds with the confirmation sentence, e.g. "I understood you would like to order a pizza. Is that correct?".
- **RegEx:** This classifies utterances according to defined and certain regular expressions, e.g. e-mail address or serial numbers. However, it only "knows" what is specified in the RegEx and cannot generalize beyond it.
- **Keywords:** You can use keywords when there are very specific ways for users to refer to the same thing. It is recommended to provide multiple keywords in order to make it more likely that the user's utterance matches the domain trigger (for example for "Calendar" domain we might choose "Calendar" and "Appointment").

□ Global Recognition of the Domain Trigger

The trigger utterances entered for each domain are recognized globally at any decision node. Decision nodes actively listen for domain triggers and intents. This means that triggers can be extracted at different decision nodes within the dialog, even in other domains of the same project. Therefore, it is important to ensure that the trigger utterances do not overlap with any trigger utterance of other domains or intents. For example, if the trigger utterance for the "address book" domain is "add number", while the "calendar" domain also has a command/intent named "add number", The system first tries to map the user utterance "add number" to a branch within the current domain, which is "calendar". Only if this does not succeed, it is tested whether a domain change was requested. Since "add number" can already be mapped locally, the domain trigger most likely will not be recognized.

On the other hand, if the user is in the "e-mail" domain and says "I'd like to add an appointment", the system will first check whether the utterance can be mapped to the current domain and if not possible, it recognizes "add appointment" as a trigger utterance for the "calendar" domain and switches to it from the current domain.

□ Configuration of single-domain projects

A domain trigger for single-domain projects has to be defined, but it will not actually be evaluated. In the single domain case, the system will always enter the domain with the first user utterance, no matter whether the domain trigger is matched or not.

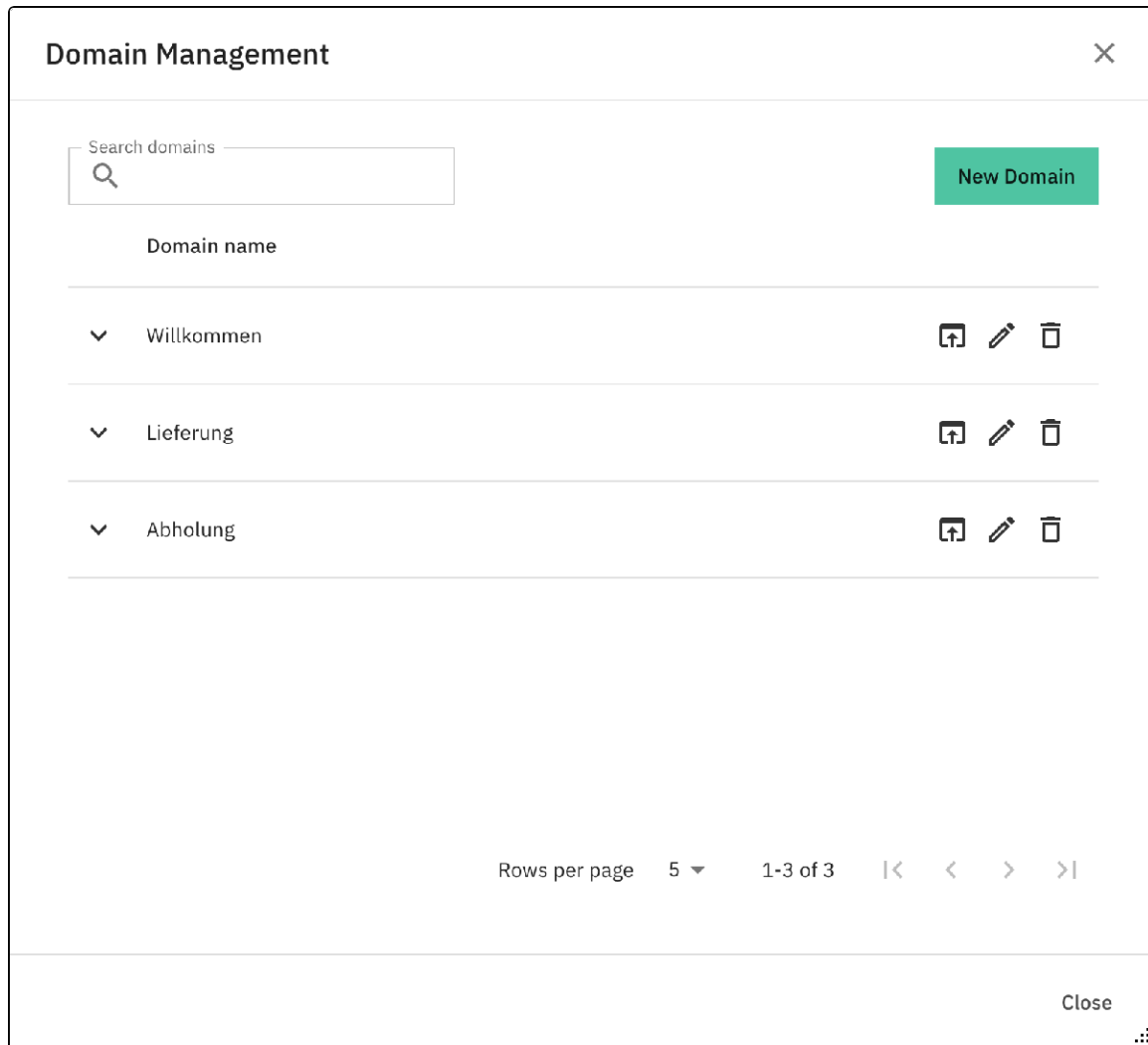
- **Protect domain:** It can be specified whether a domain change needs to be explicitly confirmed by the user. This feature is useful when losing progress in the first domain likely has undesirable effects. A domain can either be marked globally as "protected". In that case all domain changes originating from nodes within the protected domain need to be explicitly confirmed by the user. It is furthermore possible to customize the system's confirmation message.

Example: The user is in the calendar domain and is creating a new appointment. The current flow is in the decision node and the user utterance "I would like to write a new mail" is not matched as an intent of this node, instead the utterance is recognized as a domain trigger for a switch to the e-mail domain. With explicit confirmation enabled, the system would ask "Switching to e-mail, okay?" before the domain change occurs. Without explicit confirmation, the system would immediately activate the mail domain and progress for making the appointment could not be restored.

- When switching to a new domain, the progress within the first domain is not saved, except for **variables** and their content.

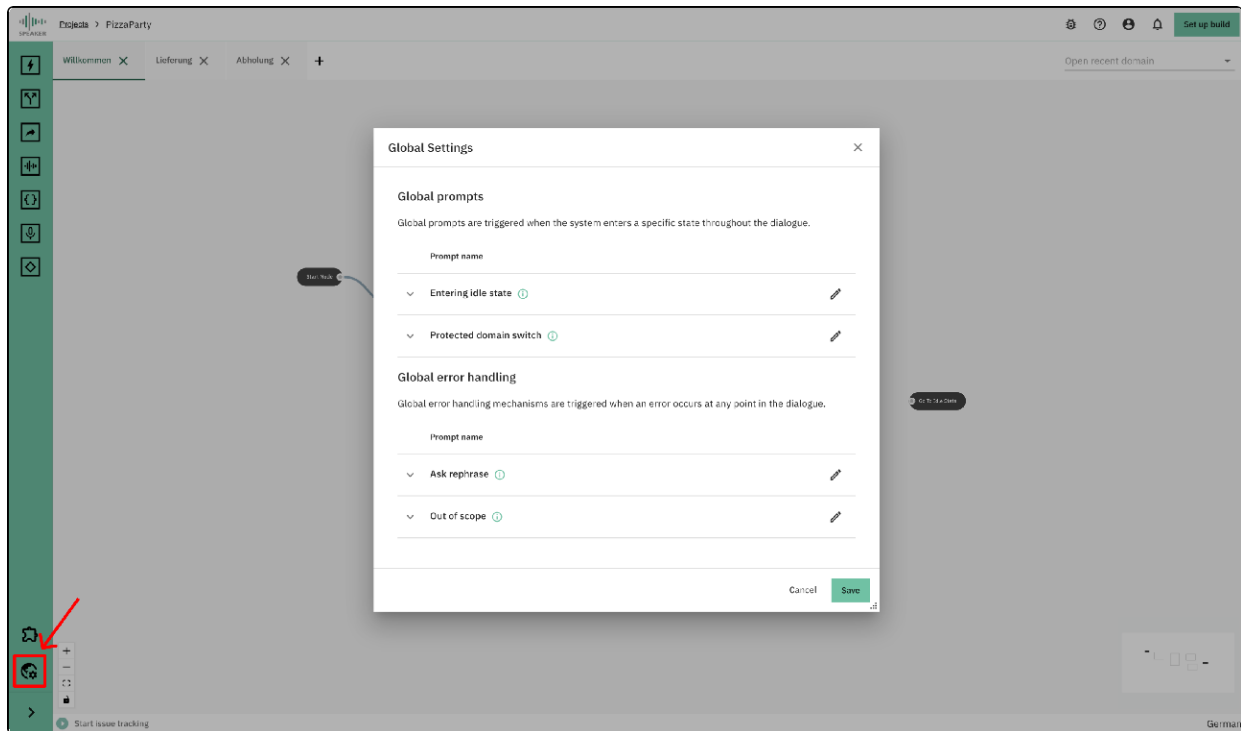
5.2.3 Open Domain

By clicking on the Open Domain card on the canvas, the Domain Management window will pop up. There you can see list of previously created domains to choose from. You can also create a new domain here.



5.3 Global Settings

Global Settings are located at the bottom of the navigation sidebar and consist of **Global prompts** and **Global error handling**.



5.3.1 Global prompts

A set of default global prompts is available in the Dialog Editor. Global prompts are triggered when the system enters a specific state. As the name suggests, global prompts are not specific to an individual node, but the same prompt is used throughout the dialog.

Currently there are two types of global prompt for two different states available in the Dialog Editor:

5.3.1.1 Protected domain switch

If no confirmation message is provided in a protected domain or node (see [Domain protection \(see page 25\)](#)), the system will use this generic message to notify the user before leaving a protected domain/node and switching to another domain.

5.3.1.2 Idle state

The idle state refers to two situations in a dialog system. The first is at the beginning of the conversation when the system receives the first user input, and the second is when the "Go to Idle State" node is reached in the dialog. It's important to note that the system is not necessarily concluding the conversation in the latter case, but only transitioning to the idle state until further input is received.

5.3.2 Global error handling

A set of default error handling mechanisms is available in the Dialog Editor. These error handling options are not specific to an individual node, but are available globally throughout the dialog.

Currently there are two types of error handling mechanisms available in the Dialog Editor:

5.3.2.1 Out of scope

The system uses the Out of Scope prompt, when the user's request can't be mapped to a known intent or domain.

5.3.2.2 Ask rephrase

In case a user intent is classified with a low confidence score, the system will prompt users with a request to rephrase their utterance.

5.4 Nodes

The Dialog Editor provides several building blocks for creating a dialog flow, each with its own unique function. These building blocks are called "nodes". The [Action Node \(see page 37\)](#) for example, enables you to perform specific actions such as playing an audio or video file. At each node, the dialog branches out based on the user's response. Nodes can be linked together to create a complete dialog flow. In the following section, we'll provide a more detailed explanation of each type of node.

☐ Deprecation of the Yes-No Node and the Control Node

For usability reasons the Yes-No Node and the Control Node will no longer be supported in the DE V2 & V3.

We found out that the usage of these two nodes was rather rare due to their "complexity" and "low added value" when creating dialogs. Therefore we will not support dialogs containing these two nodes. If you have been using these nodes in the DE V1, please contact us for us to translate your dialogs into a compatible format.

□ Custom jq code in the Dialog Editor

Condition Node and Variable Node allows you to write **custom jq code** within your assistant. Writing custom jq code provides you with a lot of flexibility and customization options to create more advanced and specific conversational AI experiences that might not be possible with pre-built features alone.

To learn more about jq, [see the documentation here](#)⁷.

We recommend you to try out [jq play](#)⁸ as a testing ground to input your jq code and see the results immediately.

Example of a serialized tracker object available to the jq-processor at runtime.

1. Go to [jq play](#)⁹
2. Copy the JSON code below in the "JSON" Input field.

```
{
  "slots": {
    "CTRL.await_domain_confirmation": false,
    "CTRL.confirmation_state": false,
    "CTRL.direct_forward": null,
    "CTRL.domain": "smalltalk",
    "CTRL.leave_ff_node": null,
    "CTRL.node": "entry_counter",
    "CTRL.oos_domain_trigger": false,
    "CTRL.original_intent": null,
    "CTRL.original_message": null,
    "CTRL.paused_forms_content": null,
    "CTRL.recognized_domain": "smalltalk",
    "CTRL.registered_form": null,
    "CTRL.requested_slot": null,
    "fruit_choice": "pineapples",
    "number_of_smalltalk_entries": 1.0,
    "SYSTEM.form_history": {},
    "SYSTEM.context": null,
    "SYSTEM.random_number": 0.2104714074942884,
    "SYSTEM.latest_user_utterance": "Hey friend!",
    "SYSTEM.latest_user_intent": "pineapple",
    "SYSTEM.latest_user_intent_confidence": 0.76,
    "SYSTEM.latest_user_domain": "smalltalk",
    "SYSTEM.latest_user_domain_confidence": 1.0
  },
  "latest_message": {
```

⁷ <https://stedolan.github.io/jq/>

⁸ <https://jqplay.org/>

⁹ <https://jqplay.org/>

```

    "intent": {
      "name": "pineapple",
      "confidence": 0.76,
      "extractor": "PrototypicalIntentClassifier"
    },
    "domain": {
      "name": "smalltalk",
      "confidence": 1.0,
      "extractor": "StaticDomainClassifier"
    },
    "entities": [
      {
        "entity": "find_me",
        "start": 6,
        "end": 30,
        "value": "some_span"
      }
    ],
    "text": "Hey friend!",
    "message_id": null,
    "metadata": {},
    "context": null
  }
}

```

3. In the "Filter" field, paste the following filters and hit "Enter" to execute the filters.

Example expression to query an intent name and confidence:

```
.latest_message.intent | {(name): (confidence)}
```

Expected output:

```

{
  "pineapple": 0.76
}

```

Example expression to query the entities extracted from the last message:

```
.latest_message.entities[].entity
```

Expected output:

```
"find_me"
```

Example expression to search for specific entity value by its type in the entities extracted from the last message. Using this in a Variable node retrieves only the first match:

```
.latest_message.entities[] | select(.entity == "find_me") | .value
```

Expected output:

```
"some_span"
```

Example expression to query a System-Slot

```
.slots["SYSTEM.context"] | if .context then .context else "nothing"  
end
```

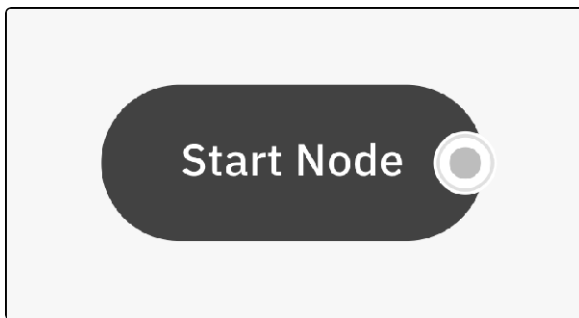
Expected output:

```
1
```

5.4.1

Start Node

The Start Node marks the beginning of the flow in your voice system experience and defines where your domain starts from. If a Start Node is not connected to any other nodes, it may indicate that the dialog flow designed on the canvas does not have any subsequent steps or actions to continue with.



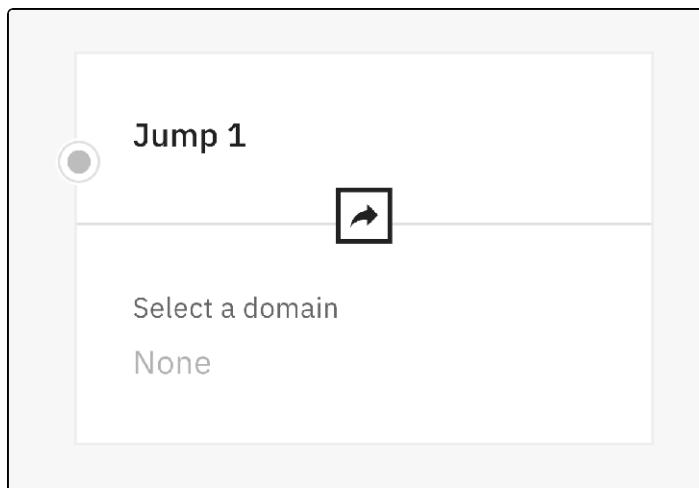
5.4.2

1 StartNode

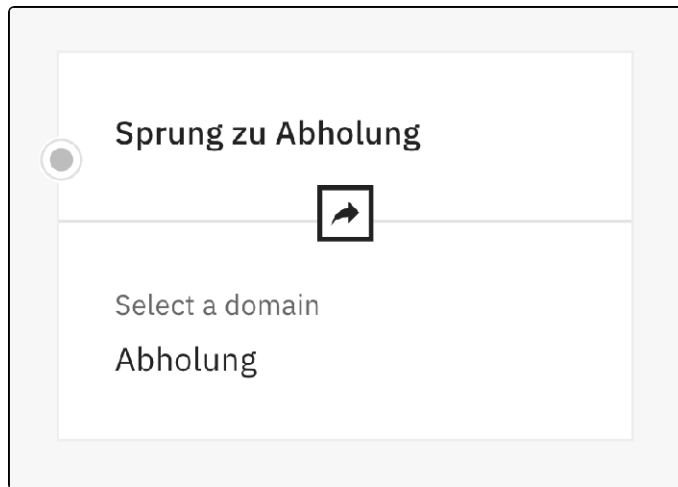
5.4.3

Jump Node

Jump Nodes enable the system to switch to a specific target node in a domain in response to user input.



2 JumpNode_ unconfigured



3 JumpNode_Configured

With Jump Nodes it is possible to trigger a switch to a **specific target node** in another domain. For example, if the user reaches a Jump Node with the target domain "calendar" in the dialog, then the system activates the corresponding target domain and the dialog follows the paths contained within the calendar domain.

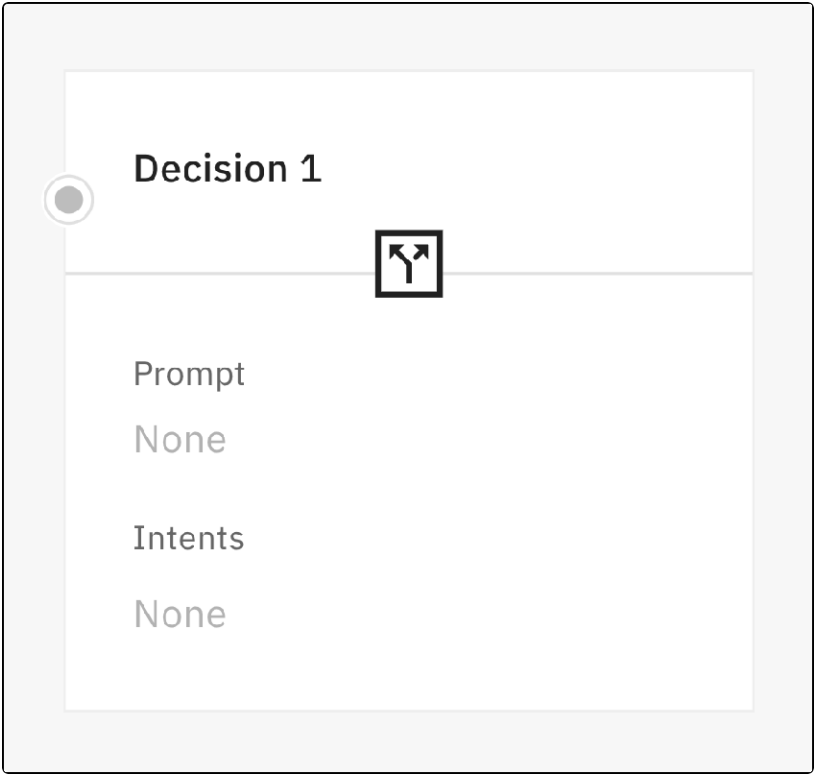
Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.
- **Target domain:** Specifies to what domain should be switched when the node is reached.
- **Target node:** Specifies at which node in the target domain the dialog should continue.

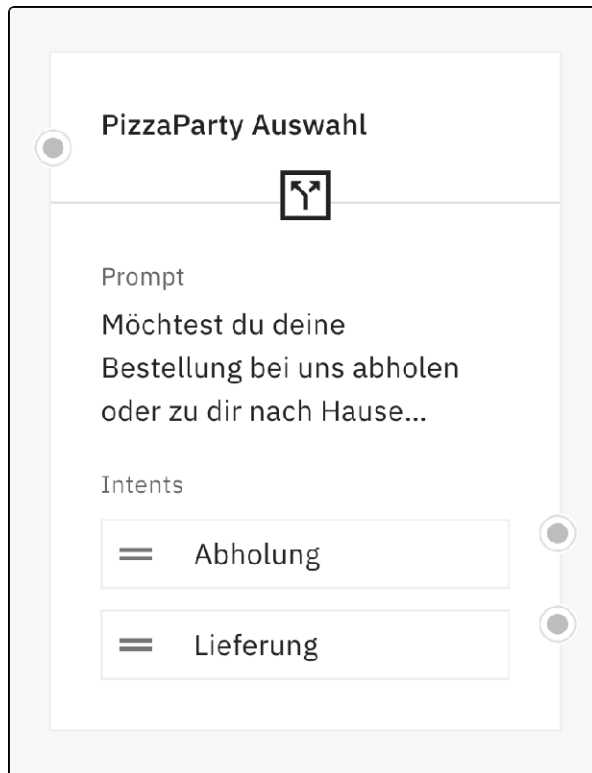
□ When switching to a new domain, any progress made in the first domain will not be saved, except for the [variable values](#) (see page 54).

5.4.4 Decision Node

Decision Nodes are used to present the user with multiple options and branch the dialog based on their input and selection.



4 DecisionNode_Unconfigured



5 DecisionNode_Configured

This node is used for letting the user decide between two or more options. From a technical perspective, there is no limitation on the number of options. From a dialog design perspective however, in most cases it is best to let the user choose between two to three options, in order not to overburden the user's memory capacity. The user can then choose one of the options and the dialog branches off. To be more precise, the system tries to map the user's utterance to one of the specified options. For each of the options, it can be configured what happens if one is selected.

Example: The user selected the pizza ordering domain and should now specify which task should be performed next. There should be three options: order pizza, cancel pizza, and request menu. To achieve this, three branches are added and configured for a Decision Node. To specify what should happen if a branch is selected, a follow-up node has to be added to each of the three branches and configured accordingly.

Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.
- **System prompt:** System prompts are utterances the system reads out before the microphone is opened for the user to speak. Multiple prompts can be provided for more variation in the dialog; the system then randomly chooses one of them. You can also include variables in the prompt text by surrounding the variable name with curly braces { } within the input field.

Example: If you have a variable named {username} in the variable management, you can use it in the input field as follows: Hello, {username}!

- **Intents:** An intent represents the goal or intention of the user's input. Each intent needs a unique name and at least one word or utterance by which it can be selected (see intent classifiers below).

Example: The user should be able to order or cancel a pizza or request the menu. Hence, three intents are specified. Before the intent is uttered by the user, we might add a system prompt like "Would you like me to read the menu to you, order a pizza or cancel an standing order?". For each of the three branches, a follow-up node has to be added and configured.

- **Intent classifiers**

- **Neural:** Neural intent classifiers are suitable if an intent can be expressed in very different ways. With the right quantity and quality of training utterances, the algorithm can even predict intents of user utterances that were not among the examples in the training data. For this though, you need a lot of training data. Once trained, the system ideally generalizes to unseen data. Neural classifiers output confidence levels (see below).
 - **Training utterances:** You can add training data for a neural classifier either manually or by uploading a .txt file containing the utterance examples. You should upload different files for the trigger of each intent and use enter, semicolon, | or a tab character to separate each training utterance. For example for a Calendar domain, you can add multiple training utterances such as: "Open my calendar", "Show me my calendar", "What's in my calendar?", "Check my appointments", "What does my schedule look like?" etc.
 - **Intent confirmation sentence:** In case of choosing neural intent classifiers, you need to specify the "Intent confirmation sentence" as well. If the system has only medium confidence that it recognized the intent correctly, it prompts the user to confirm their utterance. Example: for the category "order pizza", if the neural classifier outputs medium confidence, the system therefore responds with the intent confirmation prompt, e.g. "I understood you would like to order a pizza. Is that correct?".
- **RegEx:** This classifies utterances according to defined and certain regular expressions, e.g. e-mail address or serial numbers. However, it only "knows" what is specified in the RegEx and cannot generalize beyond it. It does not output confidence levels, since it either detects a match or not. It is suitable if an intent is always expressed in a similar way or if training data for a neural intent classifier is missing.

 Please note that the expected RegEx input should match the Python standard

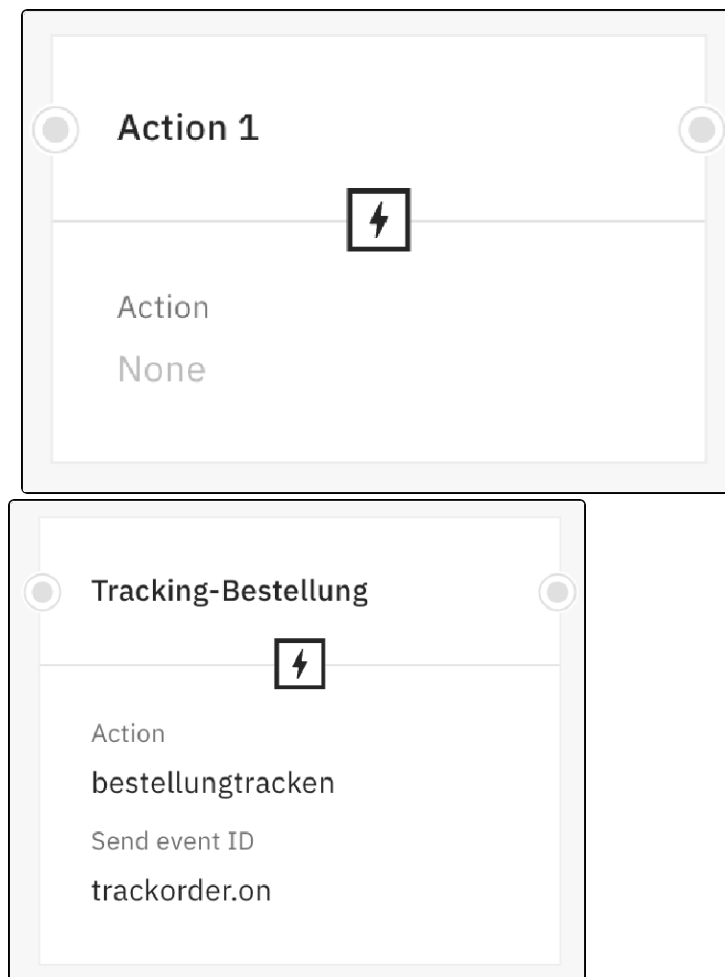
- **Keywords:** For each category, at least one keyword has to be specified by which the corresponding category can be selected by the user (for adding an ordering a pizza we might choose "order pizza" and "deliver pizza". It is recommended to provide multiple keywords in order to make it more likely that the user's utterance matches one of the keywords.

- Fuzzy matching:** When selecting keywords as intent classifiers, you can enable fuzzy matching, which is a technique used to identify and match text or data that may contain errors, misspellings, or variations from the expected format. In this method, the data input is labeled as a whole sequence and matched against the classifier, allowing for partial matches instead of exact ones. This flexibility increases the accuracy of text or data matching. For instance, if a user requests "mushroom" or "shrooms" as a pizza topping, but the system only has the entity value "mushrooms" in its database, fuzzy matching can identify "mushroom" or "shrooms" as sufficiently similar to "mushrooms" and extract it as the intended keyword. However, large differences in vocabulary items or utterance length can lower the matching accuracy
 -  Currently, the system will not automatically create plural forms of the keywords, nor forms for multiple genders (where applicable, e.g. in German). These should therefore be added manually, e.g. "pizzas" should also be added as a keyword in addition to "pizza"
- Node protection:** Similar to [domain protection](#) (see page 25), individual nodes can also be marked as "protected". In this case the confirmation prompt only appears if a domain change is triggered from this specific protected node. It is also possible to customize the system's confirmation prompt.

 If the keywords overlap with the [domain trigger](#) (see page 23), it is likely that switching domains will not work reliably. The system first tries to map the user's utterance to one of the branches of the Decision Node. Only if this does not succeed, it is evaluated if the user wanted to switch to another domain. If a keyword of a Decision Node is identical to a trigger utterance for a domain (e.g. for both, the word "order" is specified), the dialog will only follow the corresponding branch of the Decision Node and ignore the request for a domain change.

5.4.5 Action Node

Action Nodes are used within a dialog flow to initiate the execution of commands or actions when they are reached.



Actions are commands, which will be executed when an Action Node in a dialog flow is reached. Also, actions are defined as global elements within a project. Therefore, they are stored and managed within [Action Management](#) (see page 49) and they can be used within a dialog flow by using an action node to call them.

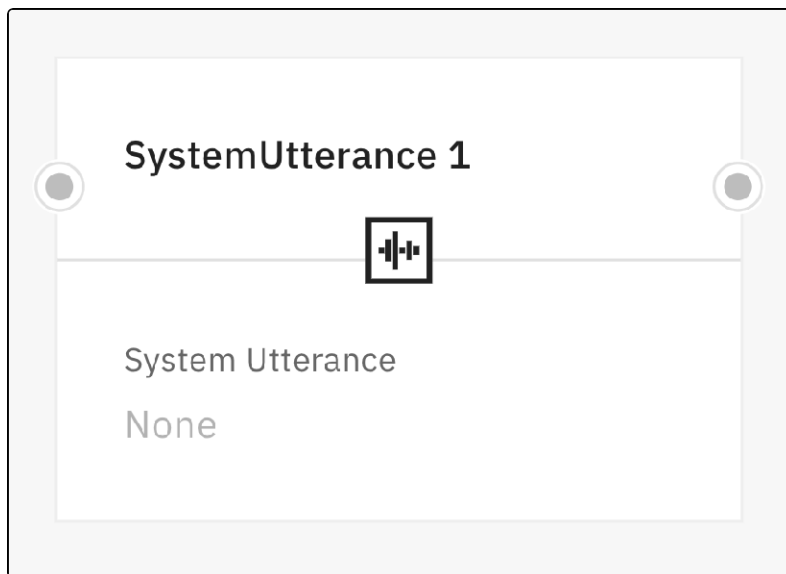
Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Action_1"), however you can change the initial node name and add a custom one.
- **Add action:** For each node, an action needs to be specified. Select from the list of existing actions or start typing to search for an existing action.
 ⚠ Please note that action names are only allowed to contain letters, numbers, hyphen, underscore and German umlauts (ä, ö, ü, ß).
- **Create an action:** In case none of the actions created in the current project fits your current step, you can create an action by clicking on the "Create action" menu item. On click you will be able to define the action type and property value.
- **Action type:** You can select one of the three [action types](#) (see page 50) available: "Play audio/video file" and "Send event ID"

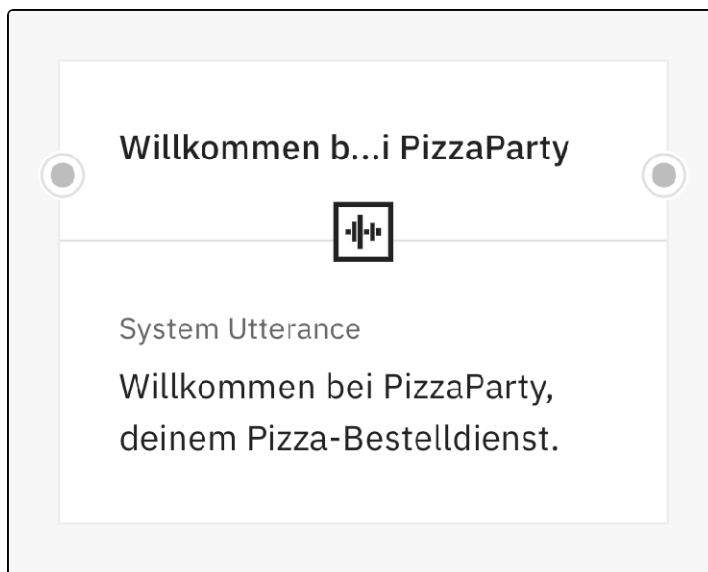
- If you want to edit existing actions, please do this in [Action Management](#) (see page 49). There you have full access to all existing actions within the project.

5.4.6 System Utterance Node

System Utterance Nodes are used for letting the system say something and then proceeding directly to the next node.



6 SysUtteranceNode_Unconfigured



7 SysUtteranceNode_Configured

Once the node is reached, the prompt text is uttered and the system automatically proceeds to the next node afterwards without any user input.

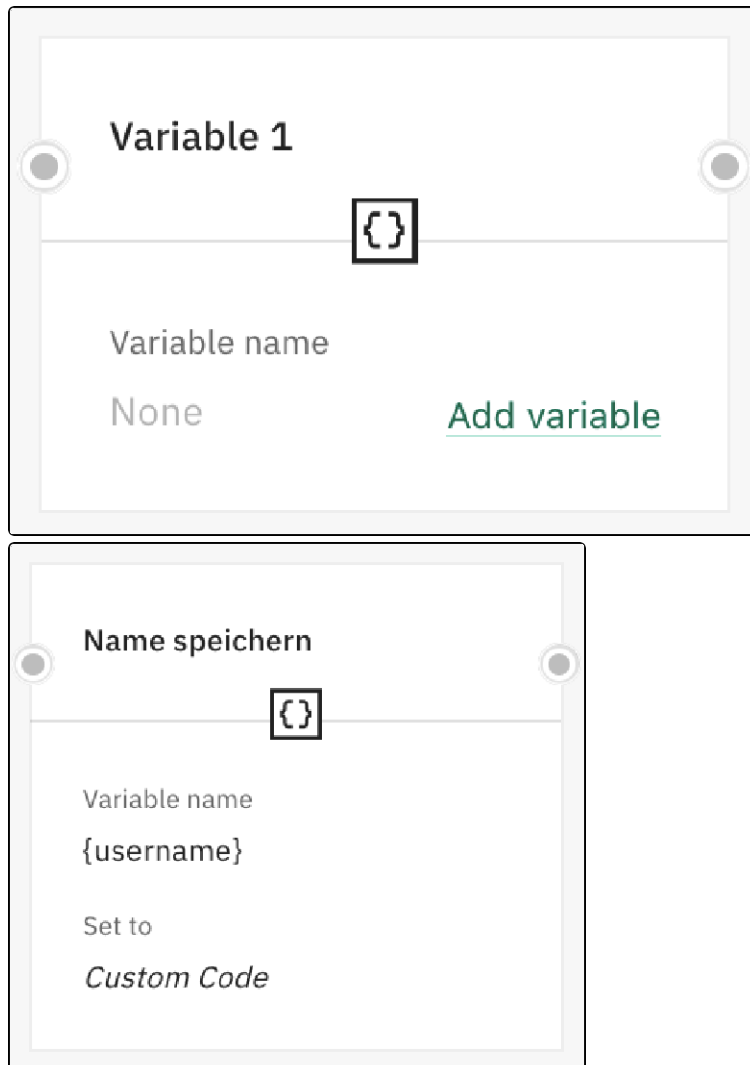
Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.
- **System utterance:** System prompts are utterances the system reads out before the microphone is opened for the user to speak. Multiple prompts can be provided for more variation in the dialog; the system then randomly chooses one of them. You can also include variables in the prompt text by surrounding the variable name with curly braces { } within the input field.

Example: If you have a variable named {username} in the variable management, you can use it in the input field as follows: Hello, {username}!

5.4.7 Variable Node

Variable Nodes are used to specify a [variable](#) (see page 54) that should be set and the value that should be assigned to it. You can view and edit them in the [Variable Management](#) (see page 0).



Setting a variable and performing an operation is executed as soon as the node is reached.

Configuration

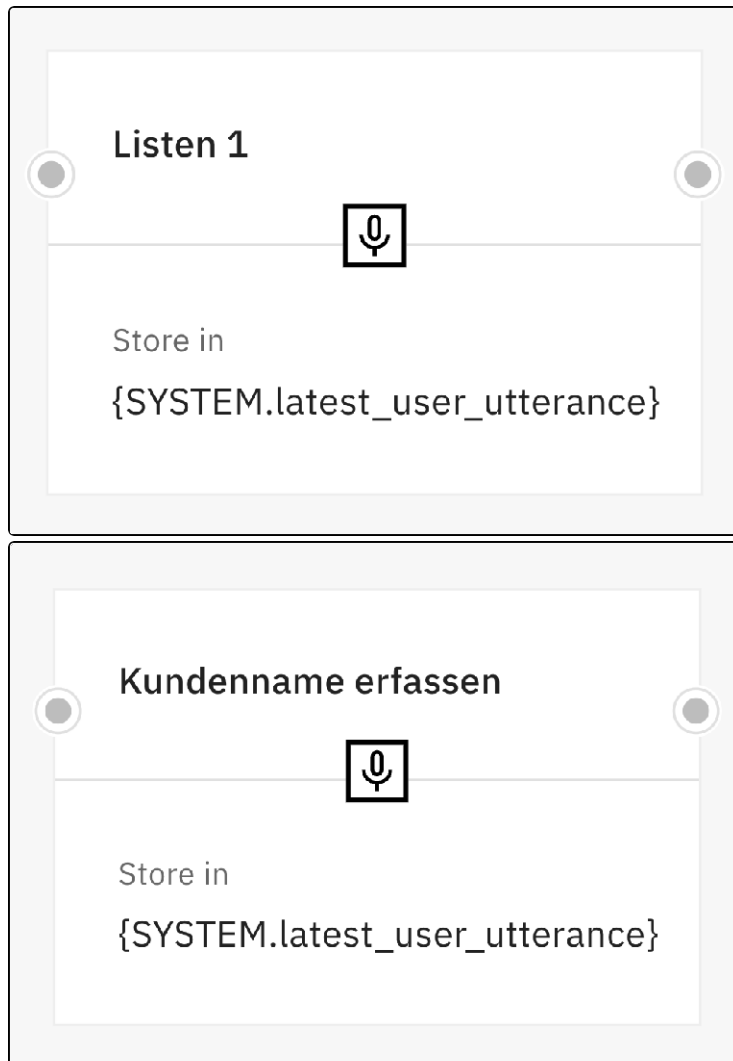
- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.
- **Find or create variable:** You can search for an existing variable to access or create a new variable.
- **Set variable:** You set a variable to an entity or an expression.
 - **Set variable to entity:** You can set your variables to an [entity](#) (see page 50), meaning filling a variable with an entity value. This usually means extracting a specific piece of information from the user's input and storing it in a variable for later use. Using this option provides you with a list of available entities, both built-in and custom to choose from.
 - **Set variable to an expression:** You can set your variables to an expression. The expression can be as simple as setting a variable to a value, another variable, or writing more complex operations. This option will provide you with the following list of operators to choose from.

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Modulo (%)
- Equal (==)
- Unequal (!=)
- Less (<)
- Greater (>)
- Less than or equal (<=)
- Greater than or equal (>=)
- **Developer mode:** To set a variable you can also write your own jq custom code by switching on the *Advanced mode* toggle. Writing custom code allows for greater flexibility and control over the system being built.

5.4.8

Listen Node

Listen Nodes are used to listen to user utterances and capture them for further processing.



By using this node, you can tell the system should listen for user input at a particular point in the conversation. When the system receives the user inputs it will capture them within the built-in variable `{SYSTEM.latest_user_utterance}` (see page 55) and then move to the next node in the conversation. The user input stored in these variables can be used later in the dialog or to track the state of the conversation.

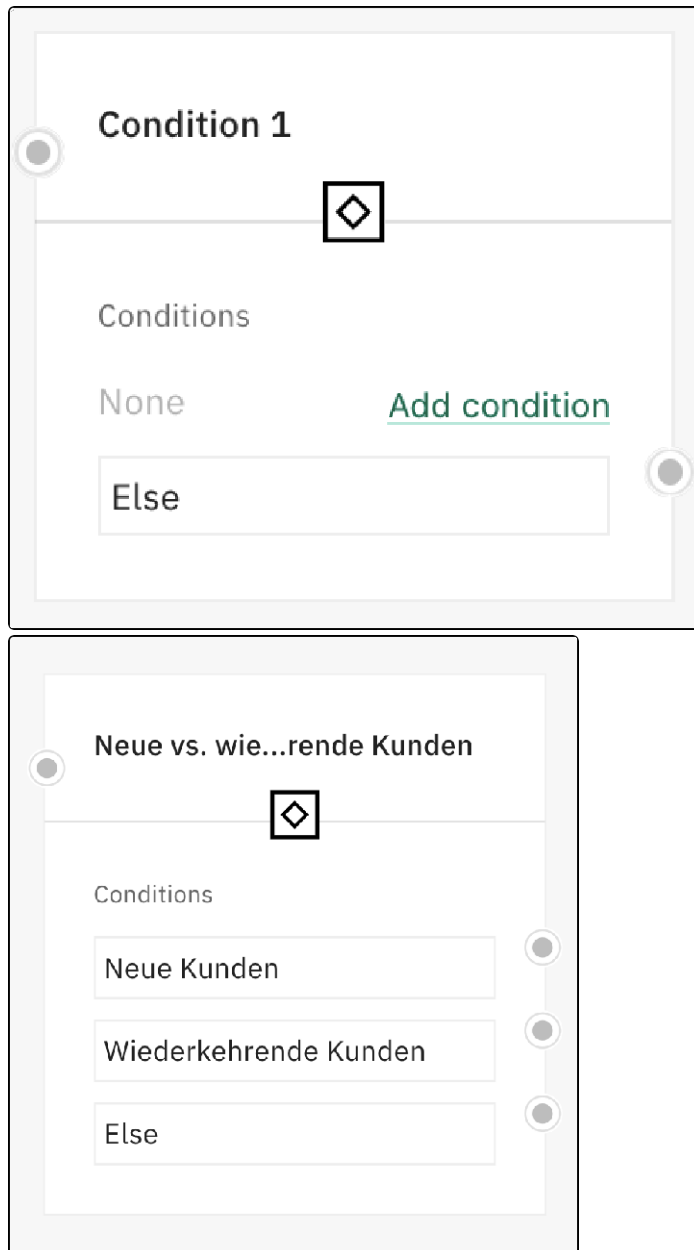
Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.

5.4.9

Condition Node

Condition Nodes are used to represent a decision point in the conversation flow, depending on specific conditions.



When the system reaches a Condition Node, it evaluates a specific condition to determine which path to follow next.

Example: A Condition Node might contain the condition "If the user's age is greater than 18". In case the condition is true, the system will follow one path through the conversation flow. But when the condition is not met, i.e. the user's age is less than 18, the system will follow a different path.

Configuration

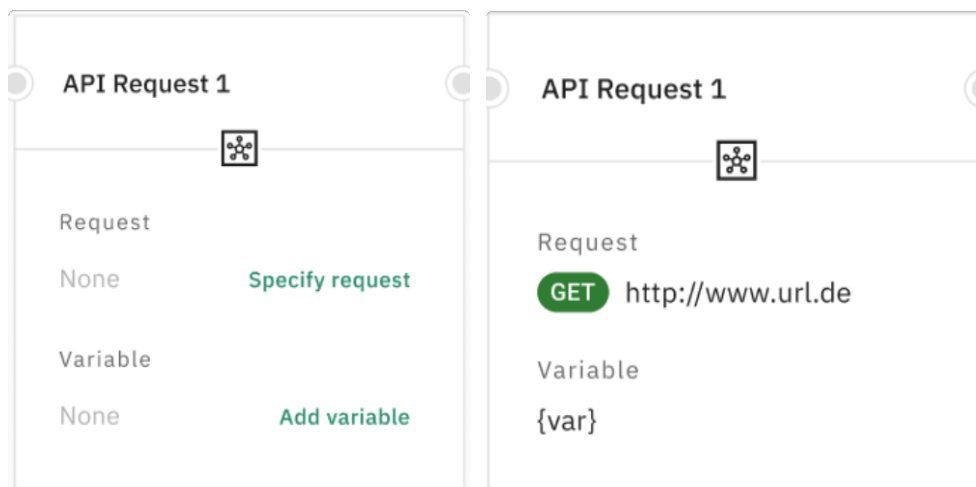
- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "Calendar_Jump_1"), however you can change the initial node name and add a custom one.
- **Condition label:** You can add a condition label to a group or set of conditions that are related to each other and evaluated together.
- **Set if statements:** To create conditions, you can use the UI functionality to add your "if" statement and multiple sub-conditions using "and" or "or" statements if needed.
 - To set a condition, first select a variable or entity from the dropdown menu, and then choose an **operator** to apply.
 - Is
 - Is greater than
 - Is greater than or equal
 - Is less than
 - Is less than or equal
 - Contains
 - not contain
 - Starts with
 - Ends with
 - Is empty
 - Is not empty
 - Is not
 - After selecting an operator for your condition, you can enter the value or another variable or entity to reference in the final field.
- **Developer mode:** To set conditions you can also write your own jq custom code by switching on the *Advanced mode* toggle. Writing custom code allows for greater flexibility and control over the system being built.
- **Add new condition:** You can set multiple conditions, each condition can consist of one "if" statement and multiple "and" or "or" statements. To create a condition, you can access both variables and entities available in your system as well.
- **Else:** The "Else" path is a predefined path in every Condition Node. To ensure proper flow in a node-based system, it's important to connect the path on the canvas, in case none of the conditions are met. Not having an "Else" path in conditions, can result in unexpected results or errors.

□ You can use the entities in a Condition Node to create specific requirements for your dialog system to follow. Please remember that in order for this to work correctly, you need to make sure that the dialog system is designed to collect the necessary information from the user in the previous turn, and extract the relevant entities accordingly. This might involve using a [Listen](#)

[Node](#) (see page 42) or a [Decision Node](#) (see page 33) to gather input from the user and extract the desired information. If you don't design the dialog system in this way, the entities in the Condition Node won't be filled with the necessary information, and the system won't be able to properly evaluate the condition.

5.4.10 API Request Node

The API Request Node is used to send a message to a server asking an API to provide a service or information.



When the system reaches an API Request Node, it uses the specified method and URL to perform an API call, whose response is then stored in a predefined [variable](#) (see page 54).

Configuration

- **Node name:** Each node needs a name that serves as a unique identifier. Every node comes with an auto-generated node name (e.g. "API Request 1"), however you can change the initial node name and add a custom one.
- **Specify request:** You have to decide which method you want to apply within this node; you can choose between GET and POST.
 - The **GET** method is used to request data from a specified resource. It's important to note that the query string (name/value pairs) is transmitted within URL of a GET request.

Example: The following GET request gives you back the order that has been delivered to a customer with the ID "1".

```
/api/orders/filter?delivery_status=delivered&userId=1
```

- The **POST** method is used to send data to a server to create or update a resource. The data sent to the server with POST is stored in the request body.

Example: The following POST request sends the first and last name of a customer, previously stored in respective variables, to the server.

```
{
  "first_name": {first_name},
  "last_name": {last_name},
}
```

- **Select storage variable:** Here you decide for a variable where you want the system to store the API response in. You can select from existing variables, or create one on the fly.
- (optional) **Fill in request body:** The request body is a representation of the resource to be created or edited. You will need it primarily when using the POST method for your API request.
- (optional) **Set timeout:** Setting up a timeout for API requests can prevent the connection from hanging forever, waiting for the response. The default is set to 3 seconds.
- (optional) **Set expression:** You can use [jq code](#) (see page 29) to select a part of the whole response that you want to be stored in the storage variable. The default is set to ".", meaning that the entire response will be stored unchanged. For instance, if you add a key like ".size", which in our case can be the size of the ordered pizza, only the value for this key will be stored.

5.4.11 Go To Idle State Node

The [idle state](#) (see page 27) refers to two situations in a dialog system. The first is at the beginning of the conversation when the system receives the first user input, and the second is when the "Go To Idle State" node is reached in the dialog. It's important to note that the system is not necessarily concluding the conversation in the latter case, but only transitioning to the idle state until further input is received.

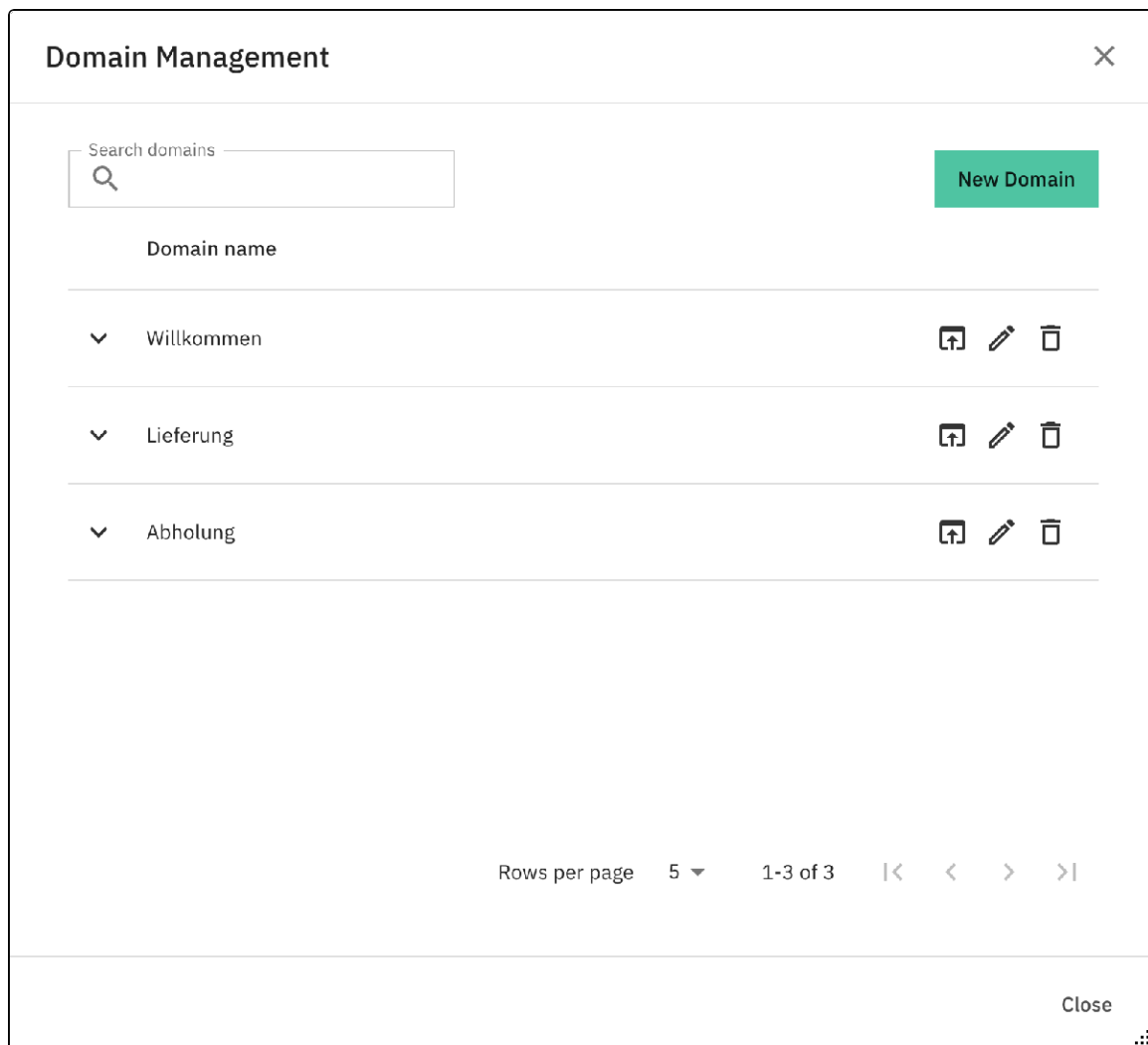


5.5 Management

5.5.1 Domain Management

The [domains](#) (see [page 21](#)) within a project are collected in a table, providing you several options to manage them. If you are looking for a specific domain, you can use the search bar to filter the table. The "New domain" button will open a new window, letting you [create a new domain](#) (see [page 22](#)).

5.5.1.1 Domain Management table



In the domain table, you see either all of your domains or the ones matching your search input. You also have the option to open a domain on the canvas, edit or delete it by clicking one of the three buttons on the right.

When extending a row in the domain table, you will additionally see:

- the domain classifier

- the assigned domain triggers, i.e. the intents that are used to activate a domain switch

5.5.2 Action Management

Actions are commands, which will be executed when an [Action Node](#) (see page 37) in a dialog flow is reached. The concept of action types can be also understood as "templates" of actions.

Action management involves tasks such as:

- viewing a list of all actions
- creating new actions
- editing and deleting actions

5.5.2.1

Action Management table

Action Management

×

Actions are commands which will be executed when an Action Node in a dialog flow is reached.

[More about actions](#)

Search actions

🔍

+ Create action

	Action name	Type ▼	Value		
▼	bestellung_tracken	Send event ID	bestellungtracken.ein	ⓘ	⋮

Rows per page

5 ▼

1-1 of 1

|<

<

>

>|

Cancel

⋮

In the actions table you will see a list of all actions created for your project or the ones matching your search input. It's also possible to filter based on action types. This table also shows you:

- the action name
- the action type

- the action property value
- a list of all domains this specific action is being used in
-  Use the expandable table option to access this information. To read more about domains in which a particular action is being used throughout your project, click on table row expand button.

Using the three dots lets you see the following:

- edit an action
- delete an action
-  Any changes to this action configuration will affect all action nodes in which this action is used project-wide.

5.5.2.2 Create new action

An action is composed by *name*, *properties* and *values*. The properties of the actions are data types that indicates the system what is required to execute an action. The value of the property is the specific information that points to how to execute the action.

Configuration

- **Action name:** Each action needs a name that serves as a unique identifier. Every action comes with an auto-generated node name (e.g. "Action 1"), however you can change the initial action name and add a custom one.
- **Action types:** You can choose out of these three action types:
 - **Play audio/video file:** This action has the *property* URL (data type), which stores the *value* from where to call the audio from. For example, one may create the action with the *name* "Play Intro Music" and select the action type "Play audio/video file".
 - **Send event ID:** This action type has the property event ID, which stores a descriptive signal trigger related to the output event (e.g. "lights.on"). The outcome action can be either a spoken response, like the system saying "the lights are now on", or a non-spoken action, i.e. actually turning on the lights.
 - **Request custom action:** "Request custom action" is the current interface to insert Python code snippets or describing what your desired action should execute. Please get in touch with tget in touch with your contact person at Fraunhofer to clarify the integration of your custom action into your system.
-  After defining the action type, add a value to your property. Property values change depending on the action type as described above. Once the action is saved, it will be listed on the actions table in Action Management.

5.5.3 Entity Management

Entities refer to specific pieces of information that can be identified and extracted from users' input. Entities can represent real-world objects, such as people, places, organizations and products, or abstract concepts, such as dates, times, and numerical values.

 Entities are signified by a hashtag # in the DE.

Entity Management involves tasks such as:

- view a list of all the entities, both built-in entities and custom entities
- creating new entities using two different extraction methods, namely RegEx and list.
- testing fuzzy matching for entities created using list extraction method

5.5.3.1 Entity Management table

In the Entity Management table you will see a list of all built-in and custom entities in your project. You can use the search bar to filter the table, if you are looking for a specific entity. Also, the "Create custom entity" button will open a new window, letting you [create a new entity](#) (see page 53).

Entity Management

You can use entities to extract information about real-world subjects from user input. There are built-in entities available or you can create custom entities. [More about entities](#)



+ Create entity

☐ Entity name	Type	Extraction method	
☐ volume	Built-in	default	
☐ url	Built-in	default	
☐ time	Built-in	default	
☐ temperature	Built-in	default	
☐ quantity	Built-in	default	

Rows per page 5

1-5 of 13






Cancel

This table also shows you:

- the entity name
- the entity type, [built-in](#) (see page 0) or [custom](#) (see page 0)
- the [extraction method](#) (see page 53)

Using the three dots lets you see the following:

- edit an entity
- delete an entity

5.5.3.2 Built-in entities

Built-in entities can be used to extract various types of information from users' input, such as dates, times, durations, and other structured data. It's designed to work with input in **both English and German**. The following built-in entities are available in the DE V3.

Name of entity	Description
AmountOfMoney	Extracts data of monetary value and unit. For example, "42 €" can be identified.
CreditCardNumber	Extracts the value of the credit card and its issuer. For example, it would identify "4111-1111-1111-1111" and "Visa" as the issuer.
Distance	Extracts the length and unit of distance from a user statement. For example: "6 miles".
Duration	Extracts the time duration and its unit. For example, users can say: "3 min".
Email	Extracts the email address. An example of an input can be: "question@pizzaparty.com".
Numeral	Extracts numerical data. A user input can be any number such as "eighty eight".
Ordinal	Extracts ordinal numbers of a user statement. For example: "33rd".
PhoneNumber	Extracts the sequence of digits of a phone number. For example: "+1 (650) 123-4567" can be identified.
Quantity	Extracts quantities according to the amount, unit and product. For example users can say: "3 cups of sugar".

Name of entity	Description
Temperature	Extracts the degree and unit of a temperature. For example, users can say: "80 C".
Time	Extracts the day and time of a user statement. For example: "Today at 9 am".
Url	Extracts the name and domain of a URL. For example: "www.pizzaparty-specials.com".
Volume	Extracts the number and unit on volume. For example: "4 liters" can be identified.

5.5.3.3 Create Custom entities

Configuration

- **Entity name:** In this field, you can enter an entity name. Please ensure that the entity name is unique.
- **Extraction method:** You can create custom entities using one of the following two extraction methods:
 - **Regex:** Regex, short for Regular Expression, is a powerful pattern-matching technique that uses a specific sequence of characters to define a search pattern. Regex can be used to identify and extract entities such as phone numbers, email addresses and more. For example, the following Regex can be used to extract a phone number from users' input: `^[2-9]\d{2}-\d{3}-\d{4}$`. When using Regex as an entity extraction method in the DE, it's important to follow the Python syntax standard for pattern matching.
 - **Regex group:** It is also possible to create Regex groups to extract entities. It is useful when you need to define a set of patterns that match a specific entity or concept. For instance, if you want to extract the payment method from users' input, you can create a Regex pattern for each of the payment methods you are interested in and label them accordingly, such as "Paypal", "credit card", and "debit card". This way, you have defined the patterns for each payment method with relevant labels and grouped them together under the common entity name "payment method".
 - **List:** Using list extraction method, you can define a set of any items that can be grouped together. This works well for short lists with easy item labels. For example, if you're building a voice bot for a pizza restaurant, and you want to be able to extract information about the toppings that customers want on their pizza. Instead of creating a separate entity for each topping (e.g. pepperoni, mushrooms, olives, etc.), you can use the list extraction method to

define a set of topping options that customers can choose from. Here's how you might define the list entity for pizza toppings:

Entity name: Toppings

Entity items: Mushrooms, Olives, Onions.

Synonyms: It is possible to add synonyms to each entity value. Synonyms are alternate names or spellings that refer to the same entity value. Adding synonyms can improve the accuracy of entity recognition and make it more user-friendly. For example you may consider adding "Bell Pepper" and "Paprika" as synonyms to Peppers or "Funghi" to Mushrooms. With these synonyms, your system will be able to recognize and extract the correct pizza topping even if the user inputs variations of the entity items.

- **Fuzzy matching:** When creating entity name using list extraction method, you can enable fuzzy matching. This is a technique used to identify and match text or data that may contain errors, misspellings or variations from the expected format. Instead of exact matches, fuzzy matching allows for partial matches, which increases the accuracy and flexibility of text or data matching. For example, when a user asks for "mushroom" or "shrooms" as a topping, but the system only has the entity value "mushrooms" in its database. Fuzzy matching can identify "mushroom" or "shrooms" as similar enough to "mushroom" and extract it as the intended entity item.

The fuzzy matching panel lets you test if a string is similar enough to any entity items and their synonyms to be considered a synonym and be extracted. By default, a 85 % match threshold is used, meaning that if the test string matches any synonym with a score of 85 % or higher, it will be extracted as a valid entity item. If the match score is below 85 %, it will not be extracted and you may need to add it manually to your list.

5.5.4 Variable Management

Variables are used to store data that can be manipulated or used by the system as it executes. For example, in DE, variables might be used to store information such as the user's name, the last action performed by the user, or the current step in the conversation.

 Variables are signified by curly brackets { } in the DE.

Variable management involves tasks such as:

- view a list of all the variables, both built-in variables and custom variables
- defining new variables
- making a custom default promptable

5.5.4.1 Variable Management table

In the Variable Management table, you will see a list of all built-in and custom variables in your project. If you are looking for a specific variable, you can use the search bar to filter the table. Also, the "Create new variable" button will open a new window, letting you creating a new custom variable.

This table also shows you:

- the variable name
- the variable type, [built-in](#) (see page 0) or [custom](#) (see page 56)

Using the three dots lets you see the following:

- edit a variable
- delete a variable

Variable Management

You can use variables to store data that is dynamic and can be applied by the system as it executes. A variable is enclosed in a curly bracket. [More about variables](#)

Create variable

☐	Variable name	Type	
☐	{pizza_list}	custom	⋮
☐	{pizza_count}	custom	⋮
☐	{pizza_amount}	custom	⋮
☐	{user_address}	custom	⋮
☐	{username}	custom	⋮

Rows per page

5

1-5 of 10

<<

<

>

>>

Cancel

5.5.4.2 Built-in variables

Built-in variables are variables that are already defined by the system and are available for use without the need to first create them. In the Variable Management in DE, the following built-in variables are accessible:

Name of variable	Description
{SYSTEM.latest_user_utterance}	It is used to store the most recent message made by the user in the recent flow in the dialog. This variable is typically updated at the end of each turn in the conversation, when the system receives a new message from the user. The value of the {SYSTEM.latest_user_utterance} variable is then used by the system to determine how to respond and what to do next in the conversation.
{SYSTEM.latest_user_intent}	It is used to store the most recent intent in a user's intent or message in a dialog. The {SYSTEM.latest_user_intent} variable is an important part of the dialog management system, as it allows the system to understand the user's goals and provide relevant responses or actions based on the user's intent. It is often used in conjunction with the {SYSTEM.latest_user_utterance} variable, which stores the actual words or text of the user's message, to provide a more complete picture of the context of the conversation.
{SYSTEM.latest_user_intent_confidence}	It is a variable that is used to store a numerical value representing the system's confidence in its identification of the user's intent. This variable can be used to allow the system to adjust its responses or actions based on the level of confidence it has in its understanding of the user's intent.
{SYSTEM.latest_user_domain_confidence}	It is a variable that is used to store a numerical value representing the system's confidence in its identification of a trigger within a particular domain.
{SYSTEM.latest_user_domain}	It stores information about the potential domain that the user's current utterance may be referring to.
{SYSTEM.random_number}	Stores and generates a random number between 0 and 1 that changes on each turn.

5.5.4.3 Create custom variable

Configuration

- **Variable name:** In this field, you can enter a variable name. Please ensure that the variable name is unique.

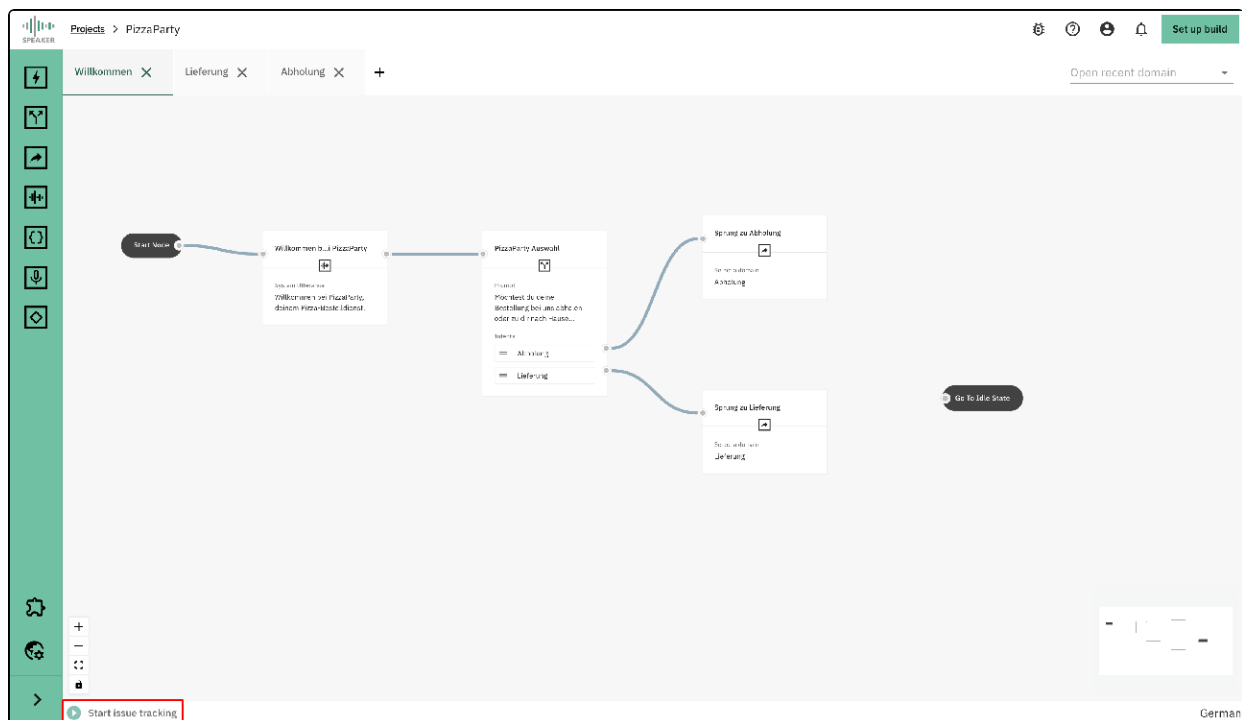
- **Promptable or non-promptable switch:** All newly created variables are promptable by default, it means that any promptable variable can be called in the prompt field of other nodes. DE uses curly braces { } to denote (variable) placeholders.

For example, if your system request from the user their names during a conversation, the user's response can be stored in a variable called {user_name}. In the next turn, to call the promptable variable {user_name} in a prompt field you might need to write your message like this: "Hello {user_name}, what can I do for you today?". This way DE will substitute the value of the {user_name} variable with the users' name when the prompt is read out to the user. In case you don't want to see a variable to be suggested in the prompt field, you can toggle the promptable switch off.

6 3. Track Issues

With issue tracking, you can run a validation for your whole project. Typically, you will do this when you're finished with your assistant creation and want to check, if there are any issues in your domains. The validation is not checking for logical coherence of the individual dialog flow, it rather picks up issues concerned with the proper and complete configuration of nodes and domains. Issue tracking allows you to make sure that you have not forgotten about any configurations before you start to [build and test your assistant](#) (see page 61).

6.1 Start issue tracking



In the status bar, you find a button to start the issue tracking. Using this button will

- initiate the issue tracking mode
- start the validation process for all your domains
- add red dots to all domains and nodes that contain issues

- open the issue panel

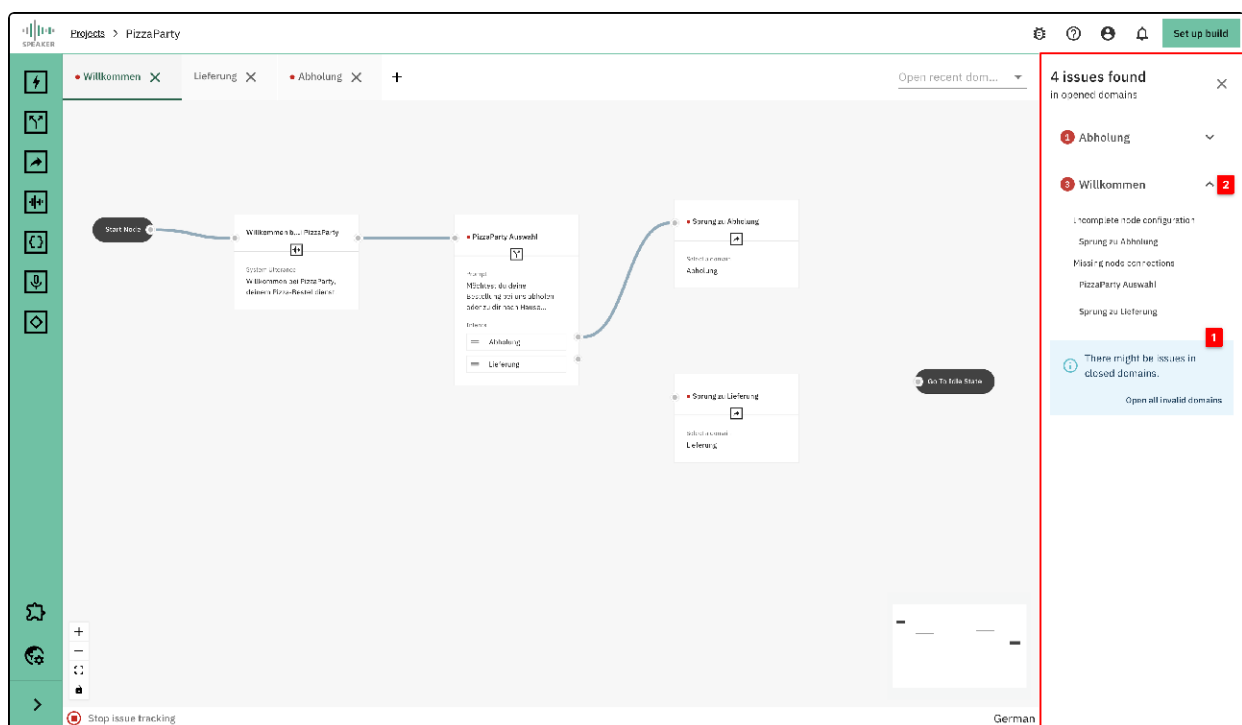
You can start or stop issue tracking at any time. It has no effect on the progress of your assistant creation.

6.2 Issue tracking mode

By starting issue tracking, you are entering the issue tracking mode. In this mode, a validation process is running in the background. Validation will be triggered by:

- saving a node configuration
- adding, changing or removing connections between nodes

6.3 Issue panel



The issue panel provides you with a list of domains that are open and contain issues. Any issue-free domain is not reflected in the issue panel. However, also domains that are currently not open are validated. If the system finds issues in closed domains, you see a hint with the option to open those domains as well .

The list of issues for the currently active domain is always expanded . The issues are categorized after the issue type.

After an issue was fixed, the respective item automatically disappears from the issue panel.

6.3.1 Issue types

6.3.1.1 Invalid node configuration

This issue type indicates that you missed to add some required information in the configuration of a node. For example, if a system utterance node has an undefined prompt, your assistant does not know what to say. This would cause an error in the dialog.

You can click on the node in the issue panel to center it on the canvas, go to the configuration and add the required information there.

 Keep in mind, that most of the fields in the configuration are required to create an assistant that works as expected. If any information is optional, it will be highlighted in the respective form field.

6.3.1.2 Missing node connection

This issue type indicates that a connection between two nodes is missing. We can't allow your dialog running into a dead end, this is why we ask you to add this connection before build.

 There is an exception for the [Go To Idle State Node \(see page 47\)](#). This node does not need to be connected.

6.4 Stop issue tracking

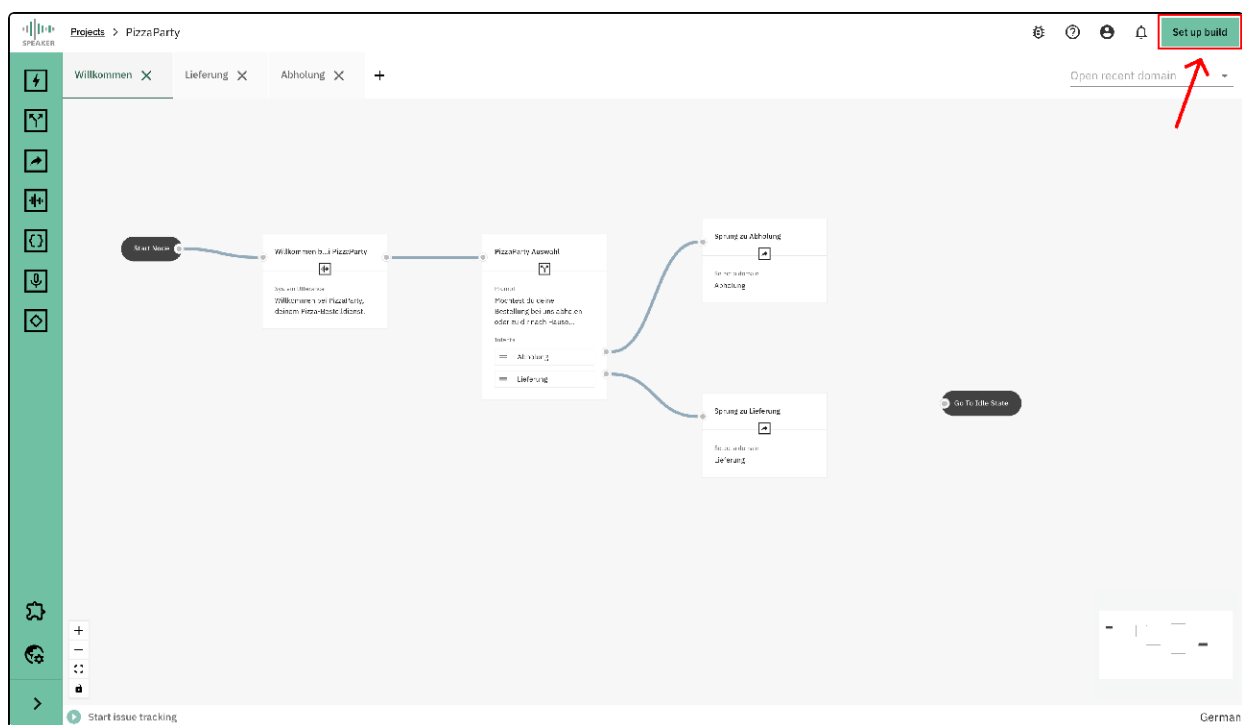
While you are in issue tracking mode, you find a button in the status bar to stop the issue tracking. Using this button will

- stop the issue tracking mode
- stop validation process for all your domains
- remove the red dots that indicate issues in domains or nodes
- close the issue panel

Now you can continue working on your dialog without being notified about issues.

7 4. Build and Test Assistants

In order to bring your dialog design to life, the included data needs to be converted into **executable code** that is ready for use (= **build**). Creating a build of your dialog design and deploying it to [Allinga Stage](https://stage.speaker-plattform.de)¹⁰ are the final steps before you are able to interact with the assistant and test out its functionalities.



7.1 Set up build

In the upper right corner of the screen, you can configure your build via the "Set up build" button. It will open another window, giving you the option to select domains for your build. In the first step, it runs a validation for your whole project. There are two possible results of this validation:

1. Your project only contains successfully validated domains. In this case, you can continue to build.

¹⁰ <https://stage.speaker-plattform.de>

Set up build

Select domains for your build

☒	Domains
☒	Willkommen
☒	Lieferung
☒	Abholung

Rows per page: 5 ▾ 1-3 of 3 |< < > >|

3/3 domains selected

Close

Continue to build

2. During validation, issues have been detected in your domains.

Set up build

Select domains for your build

 Your selection still contains invalid domains. They need to be repaired before starting the build process.

Open invalid domains

☒	Domains
☒	Willkommen
☒	Lieferung
☒	Abholung
☒	1 Reservierung

Rows per page: 5 ▾ 1-4 of 4 |< < > >|

4/4 domains selected

Close Continue to build

In that case, you have the following options:

- Fix all the invalid domains (using [issue tracking \(see page 58\)](#)) and try to create a build again after you're done. This is the recommended solution.
- Deselect invalid domains from your build to continue.

□ Excluding a domain from your build might break links between domains if you were using jump nodes to connect them. In this case, you will receive an error message and you will not be able to create the build.

7.2

Create new build or overwrite

In the next step, you will decide on either creating a new build or overwriting an existing build.

Every user group has a limited number of deployments.

When no build has been deployed yet, you only have the option to create a new one. Once all slots are used up by your group, you will have to start overwriting existing builds. You will also have the option to delete existing builds, to make those slots available again for new deployments.

After entering a name and a description (optional) for your build, you can initiate the build job.

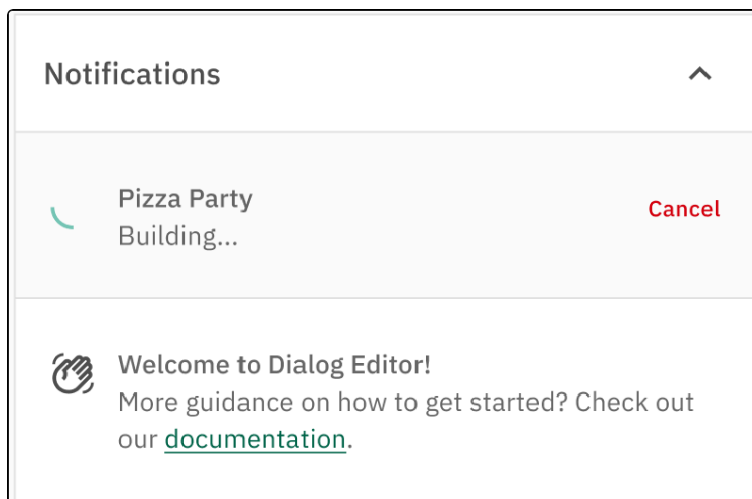
7.3 Build job

After initiating the build job, you see your running build job as the top item in your build history. You can access it via the notification bell in the header.



While your build job is running, you can continue working on your project or even go to another project.

In the build history, you find the status for every build job that was initiated (running / successful / failed) as well as details for every closed build job.



7.3.1 Successful build

When a build job is done, you will see a confirmation message at the bottom of the screen and a green dot on the notification bell. You can now go to the build history, where you get an overview of all the build jobs initiated within your group.

- By using the "Preview" button, you can test your assistant in a chat window (without ASR and TTS components).
- You can also access your build in [Allinga Stage](#)¹¹. Here, you can select your build (in the "Service" dropdown) and test it, including the other components as needed.

7.3.2 Failed build

Also, a build job might fail. In this case, you will see an error message at the bottom of the screen and also a green dot on the notification bell. For example, a build can fail due to:

- connection breakage
- infrastructure maintenance
- deployment conflicts
- problems within the dialog structure that have not been detected with [issue tracking](#) (see page 58)

In any case, please give it another try. If the problem persists, please contact us via the [Service Desk](#)¹² for further support.

¹¹ <https://stage.speaker-plattform.de>

¹² <https://jira.iais.fraunhofer.de/jira/servicedesk/customer/portal/9>

8 Explore Dialog Editor Templates

PizzaParty

PizzaParty is a simplified version of a German-speaking assistant used for taking pizza delivery orders.

You will get to [decide](#) (see page 33) for either pick-up or delivery. To process your order, the assistant will store your {user_name} and (in case of delivery) your {user_address}. After that, you will be asked for the amount of pizzas you'd like to order. As the shop is still very new, you only get to choose between the 3 most popular pizzas in Germany. A list of your order is generated in a loop (using [variables](#) (see page 43) and [conditions](#) (see page 43)) until you made a choice for your desired amount of pizzas.).

PizzaParty is aligned with example instructions used within Dialog Editor. So if you want to learn step-by-step how to use Dialog Editor, joining the PizzaParty will get you there.

BuddyBot

Introducing **BuddyBot** - the friendly chatbot that wants to be your friend! BuddyBot is a simple chatbot that loves to ask you questions about your favorite colors, fruits, and seasons, all in the hopes of finding out if you two could be friends.

Every time you tell BuddyBot one of your favorite things and it matches with a pre-defined entity list of Good_colors, fruits, or seasons, your {friends_score} goes up by one. But if your {friends_score} is lower than 1, BuddyBot will apologize and say that it's not meant to be friends.

BuddyBot is a fun and easy way to learn more many Dialog Editor features including [Listen Node](#) (see page 42), [Jump Node](#) (see page 0), [variables](#) (see page 0) and [entities](#) (see page 50).

Give it a try and see if BuddyBot and you can hit it off!

GeoGuesser

Want to check your knowledge about capitals of German federal states? Look no further than **GeoGuesser** - it asks you a series of questions about the capital cities of German states, and every time you answer correctly, your score goes up by 1.

Behind the scenes, GeoGuesser uses a pre-defined list of German states and their capitals, stored in a variable called {database} using [custom JQ code](#)¹⁶. And to keep things interesting, the chatbot uses `{SYSTEM.random_number}` (see [page 55](#)), a built-in random number generator, to choose a random question from the database each time you play. Look into the configuration of the [Variable Nodes](#) (see [page 40](#)) to see how it's done.

⚠ **SPOILER ALERT:** The answers for the quiz are inside the code as well. So better take the quiz first before you take a look into the dialog depth 😏

¹³ <https://stedolan.github.io/jq/>

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel SPEAKER: Aufbau einer führenden Sprachassistentenplattform „Made in Germany“	
4. Autor(en) [Name(n), Vorname(n)] Jeffrey Ahmad, Markus Bäuerle, Steffen Becker, Dr. Felix Burkhardt, Dr. Christian Dittmar, Jochen Emig, Prof. Dr.-Ing Tim Fingscheidt, Dr. Johannes Fischer, Jan Frömberg, Thomas Geiger, Viktor Haag, Mohammad Joshaghani, Dr. Christian Kruschel, Dr. Fabian Küch, Dr. Manfred Langen, Zhengyang Li, Christian Lieske, Dr. Julian Moreno Schneider, Philipp Niemann, Florian Sattler, Dr. Florian Schiel, Julia Schneider, Claudia Schönfelder, Dr. Sebastian Schönnenbeck, Edgar Suarez, Roman Teucher, Simon Verdenhalven, Daniel Vollmers, Kim Wagener, Dr. Oliver Walter, Dr. Thomas Wiesner	5. Abschlussdatum des Vorhabens 30.09.2023
	6. Veröffentlichungsdatum
	7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) SPEAKER Konsortium	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen <i>01MK20011A</i>
	11. Seitenzahl 190
12. Fördernde Institution (Name, Adresse) Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn	13. Literaturangaben 55
	14. Tabellen 7
	15. Abbildungen 41
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum)	
18. Kurzfassung Im Rahmen des SPEAKER Projekts bestand die Aufgabenstellung im Aufbau einer Sprachassistentenplattform "Made in Germany". Hierbei wurde eine Plattform entwickelt, welche insbesondere im professionellen Umfeld Anwendung findet. Daher wurde die Plattform zielgerichtet den Anforderungen der Industrie und Business-to-Business (B2B) Anwendungen zugeschnitten und deren speziellen Bedürfnisse berücksichtigt. Hierzu zählen Datensouveränität, ein transparenter und sicherer Umgang mit Nutzerdaten, domänenspezifische Anpassbarkeit, Flexibilität beim Produktivbetrieb, einheitliche Schnittstellen, sowie attraktive Lizenzmodelle. Schon während der Projektlaufzeit wurde die Plattform und die Technologiemodule in verschiedenen Anwendungsfällen der Projektpartner erprobt.	

19. Schlagwörter Sprachassistentenplattform, Datensouveränität	
20. Verlag	21. Preis

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report
3. title SPEAKER: Building a leading voice assistant platform "Made in Germany"	
4. author(s) (family name, first name(s)) Jeffrey Ahmad, Markus Bäuerle, Steffen Becker, Dr. Felix Burkhardt, Dr. Christian Dittmar, Jochen Emig, Prof. Dr.-Ing Tim Fingscheidt, Dr. Johannes Fischer, Jan Frömberg, Thomas Geiger, Viktor Haag, Mohammad Joshaghani, Dr. Christian Kruschel, Dr. Fabian Küch, Dr. Manfred Langen, Zhengyang Li, Christian Lieske, Dr. Julian Moreno Schneider, Philipp Niemann, Florian Sattler, Dr. Florian Schiel, Julia Schneider, Claudia Schönfelder, Dr. Sebastian Schönnenbeck, Edgar Suarez, Roman Teucher, Simon Verdenhalven, Daniel Vollmers, Kim Wagener, Dr. Oliver Walter, Dr. Thomas Wiesner	5. end of project 30.09.2023
	6. publication date
	7. form of publication
8. performing organization(s) (name, address) SPEAKER Consortium	9. originator's report no.
	10. reference no. 01MK20011A
	11. no. of pages 190
12. sponsoring agency (name, address) Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn	13. no. of references 55
	14. no. of tables 7
	15. no. of figures 41
16. supplementary notes	
17. presented at (title, place, date)	
18. abstract As part of the SPEAKER project, the task was to create a voice assistant platform "Made in Germany". A platform was developed specifically for professional use, tailored to the requirements of the industry and business-to-business (B2B) applications, taking into account their specific needs. This includes data sovereignty, transparent and secure handling of user data, domain-specific adaptability, flexibility in productive operation, standardized interfaces, and attractive licensing models. During the project duration, the platform and technology modules were tested in various application cases with project partners.	

19. keywords	
20. publisher	21. price