Automatic documentation of results during online architectural meetings

Oleksandra Klymenko, 17.06.2019, Garching

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

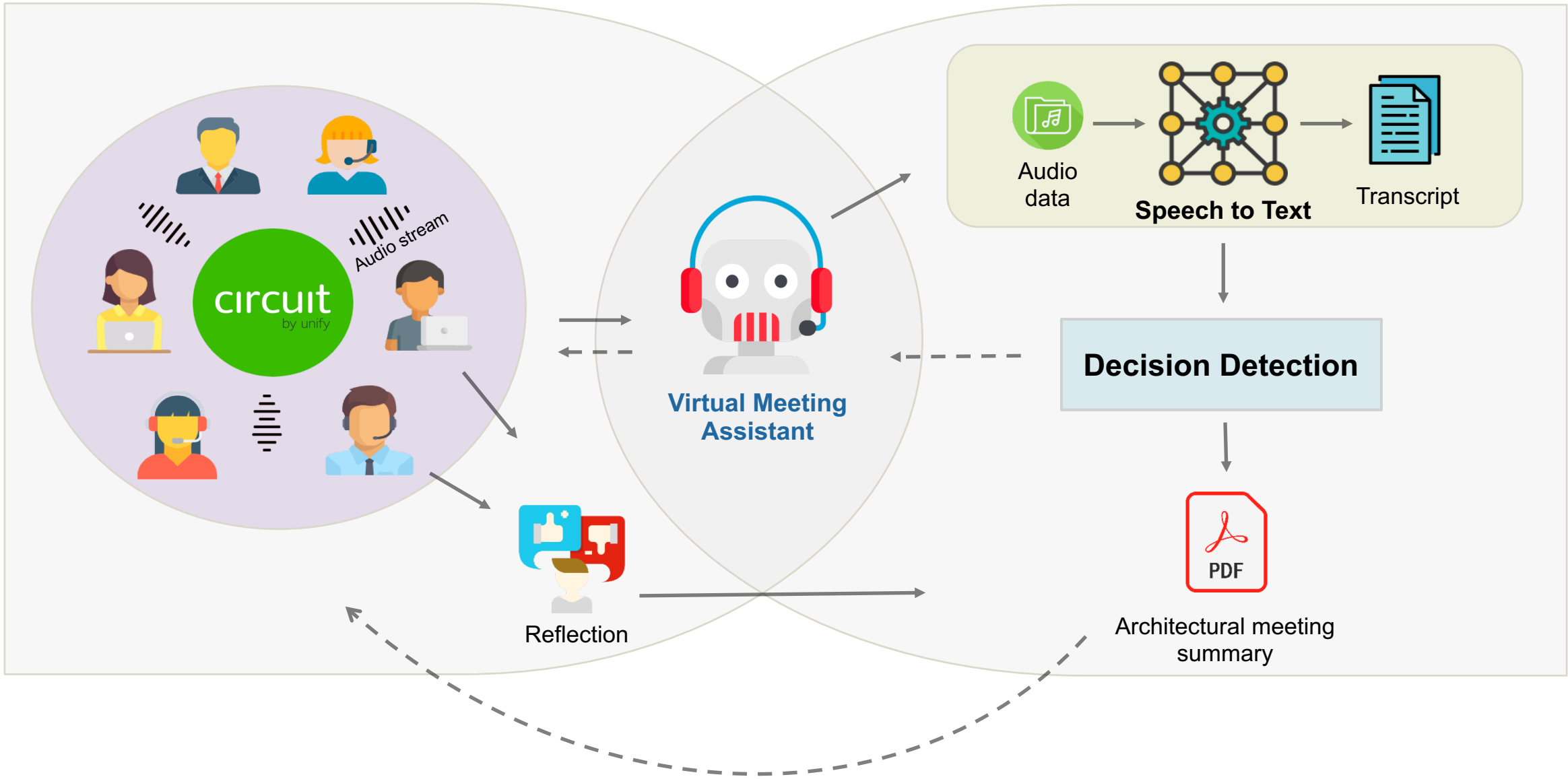
Motivation

- **Communication** between team members is crucial for efficient software development work
- In distributed teams, **most of the meetings happen online**
- Capturing and explicitly documenting decisions **enables reasoning and decision support** [1]
- Manual effort, time and cost of explicit documentation is a **concern for practitioners**
- Previous research has mostly focused on detecting decisions in **issue management systems** and **source code commits**
- However, many decisions are implicitly made in **online meetings**
- The most frequent form of documentation of architectural design decisions is **meeting minutes** [2]
- **Virtual Online Assistant** can help to document, review and refer back to made decisions

Research Questions

RQ 1	How are online meetings between software development professionals held in practice and how are they documented?
RQ 2	What is the process of decision-making in distributed software development teams?
RQ 3	What are the requirements for a system that automatically documents online architectural meetings?
RQ 4	How to identify, extract and document design decisions in online architectural meetings?

Idea

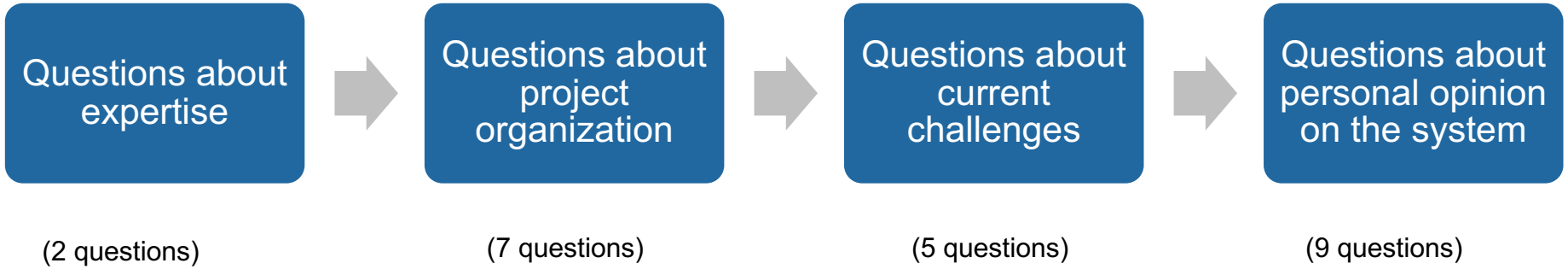

circuit
by unify
Audio stream
Virtual Meeting Assistant
Audio data
Speech to Text
Transcript
Decision Detection
PDF
Architectural meeting summary
Reflection

Research Approach

1. Literature review
2. Design and conduct interviews (RQ1, RQ2, RQ3)
   - Transcribe interview recordings
   - Analyze feedback and elicit requirements
3. Data collection and analysis
   - Record and transcribe meetings
   - Label the data
4. Technical implementation (RQ4)
   - Rasa NLU
5. Evaluation

Case Study

Interview phases



Planning

- Cooperation with UXD and RE departments
- Semi-structured interview
- 23 open questions
- 10 interviewees
- Mostly senior architects and product owners
- Ø 13 years of experience in IT industry
- Planned duration of the interview: 30 minutes, without interruptions
- Question catalog was not provided to the interviewees in advance or during the interview

Interview

Goal:

To understand participants':

- Current challenges faced with Circuit
- View of an Assistive Bot during a Virtual meeting Scenario

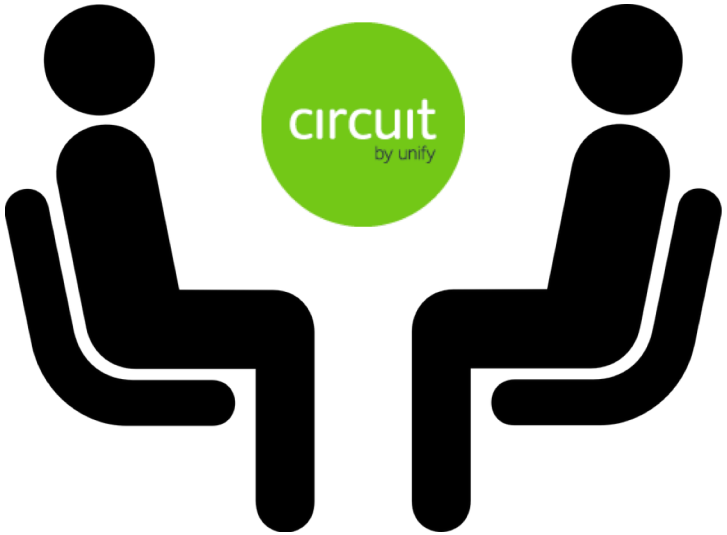
Analysis:

- Creating transcripts
- Coding transcripts
- Consolidating the list of requirements

Results:

Obtained expert feedback concerning:

- Current online meeting process
- Decision-making process
- The proposed use cases
- Requirements for the bot
- Usefulness of an automatic summary
- Other ideas



Requirements for the system

Things to include in the summary:

- ✓ Action items / TODOs
- Assigned tasks
- ✓ The person who has been assigned
- Who brought something up
- The person who made the decision
- ✓ Deadline
- Open topics (things that need follow-up)
- ✓ Catch words
- Come-in / Drop-out times
- Information / News (e.g. news from management)
- Participants' telephone numbers

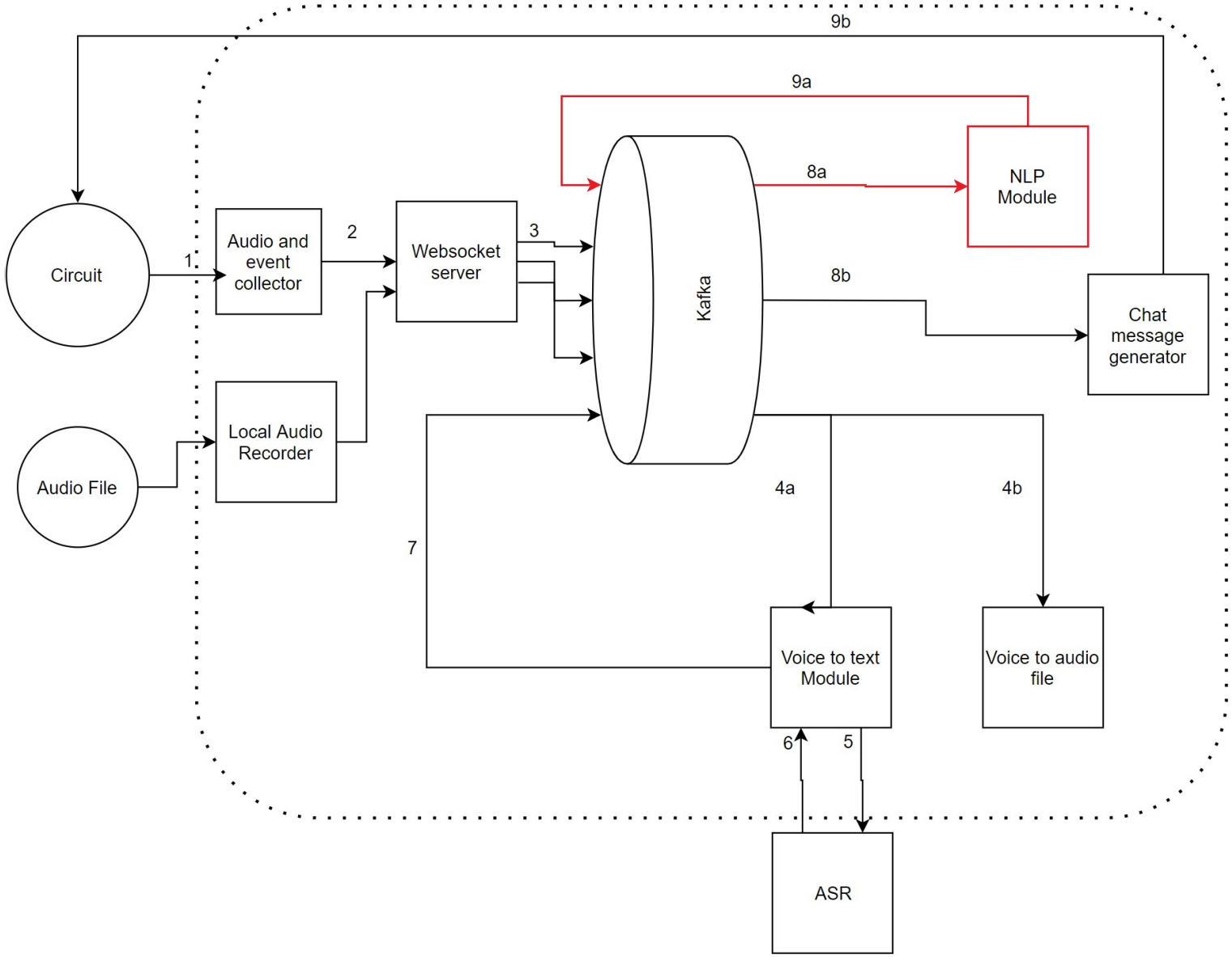
Timing of bot's questions:

- Immediate notifications vs. waiting for the end of the call
- Optimal solution: combination of both approaches

Other ideas

- Anonymous notification when discussion goes off-topic

Existing System



Implementation

Natural Language Understanding

- Rasa NLU
- spaCy pipeline
- ner_crf

Data Corpus

- Many options considered
- Only real internal meetings used
- 17 meeting recordings
- > 620 minutes of meetings
- 129 training examples in total

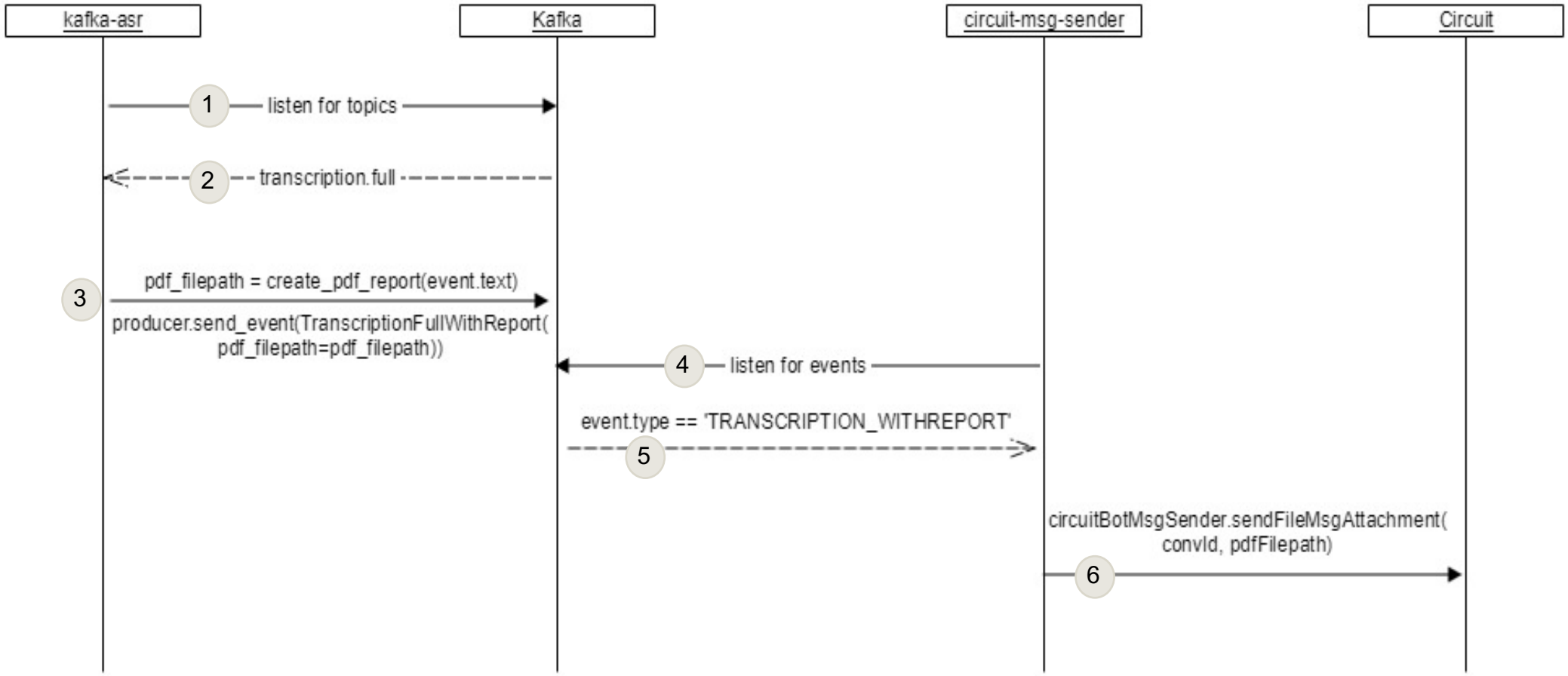
Concept Extraction

- Linked Data
- DBPedia

Model


```
{
   "intent": "decision",
   "entities": [
     {
       "start": 113,
       "end": 177,
       "value": "we can experiment with the whole of the country
       credibility part",
       "entity": "s"
     },
     {
       "start": 314,
       "end": 364,
       "value": "we have to think of a strategy of integrating that",
       "entity": "d"
     }
   ],
   "text": "And if we go to the next you did with the next big in here
   the first one I think to begin with I would still say we can experiment
   with the whole of the country credibility part as well with the existing
   dashboard as a service that is being developed in the sense that For me
   the complexity still seems to be that we have to think of a strategy
   of integrating that."
}
```

Sequence Diagram



Demo

ASR Evaluation

- Word Error Rate (WER)
- $\text{WER} = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$
  - S is the number of substitutions
  - D is the number of deletions
  - I is the number of insertions
  - C is the number of correctly transcribed words
  - N is the total number of words originally spoken (N=S+D+C)

ASR tool	**Speechmatics v.2.0.1**
Substitutions (S)	208
Deletions (D)	110
Insertions (I)	27
Correct words (C)	1014
Total number of words (N)	1333
Word Error Rate (WER)	**26%**

Table 1: Evaluation of Automatic Speech Recognition

Model Evaluation

- 15 meetings that contained decisions
- 5-fold validation
- Precision, recall, F1-score
- $R = \frac{TP}{TP+FN}$, $P = \frac{TP}{TP+FP}$, $F = 2 * \frac{P*R}{P+R}$

	Meeting 1	Meeting 2	Meeting 3
TP	2	6	8
FP	13	23	22
FN	2	11	8
TN	91	202	121
Precision	0.133	0.207	0.267
Recall	0.5	0.353	0.5
F1 score	0.207	0.261	0.35

Table 2: Evaluation of the decision detector model

Challenges and Limitations

- Data scarcity
- Quality of speech recognition
- Challenges of spoken language
- Uncertainty Expressions
  - *"**I think** we should follow up with Martin"*
  - *"**Maybe** we should have another meeting on Wednesday*
- Identifying referring expressions
  - *"We definitely have to implement **it**"*
  - *"I will ask **her** to document **it**"*
- Identifying context and distinguishing decision types
  - *"**I will** create the API"*
  - *"**I will** set up a meeting on Monday"*
  - *"**I will** share my screen"*

Future work

- **Model Enhancement**
  - Solving Data Scarcity
  - Handling uncertainties
  - Resolving referring expressions

- **Virtual Assistant Development**
  - Add communication with participants
  - Implement “Suggestions and recommendations” use case
  - Implement Anonymous notifications

References

1. Bhat, Manoj, et al. "Automatic extraction of design decisions from issue management systems: a machine learning based approach." *European Conference on Software Architecture*. Springer, Cham, 2017
2. C. Miesbauer and R. Weinreich. “Classification of design decisions–an expert survey in practice.” In: European Conference on Software Architecture. Springer. 2013, pp. 130–145.
3. Icons: https://www.flaticon.com/

M.Sc.

Oleksandra Klymenko

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

alexandra.klymenko@tum.de
wwwmatthes.in.tum.de