

Bachelorverteidigung

LSML: Generating Spoken Lectures From Scripts

Christian Dunkel

Gutachter: Prof. Dr. Benno Stein, Jun.-Prof. Dr. Jan Ehlers

Betreuer: Jun.-Prof. Martin Potthast, M.Sc. Lars Meyer

1. Motivation und Anforderungen

- 2. LSML
- 3. Implementierung
- 4. Evaluierung
- 5. Zukünftige Arbeit
- 6. Zusammenfassung

Motivation

Die Digitalisierung der Bildung schreitet voran und Vorträge werden zunehmend online gehalten. Das Pandemiejahr 2020 wird hierfür als Katalysator dienen. Durch die schnelle Umsetzung ergeben sich aber eine Reihe neuer Probleme:

- Wenige haben Übung darin digitale Vorträge zu erstellen
- Die Unkosten, besonders der Zeitaufwand, sind um ein Vielfaches höher als bisher
- Nicht jeder fühlt sich wohl dabei seine Stimme oder sein Bild aufzuzeichnen
- Die Produktionsqualität der Ergebnisse lässt meist zu wünschen übrig

Motivation

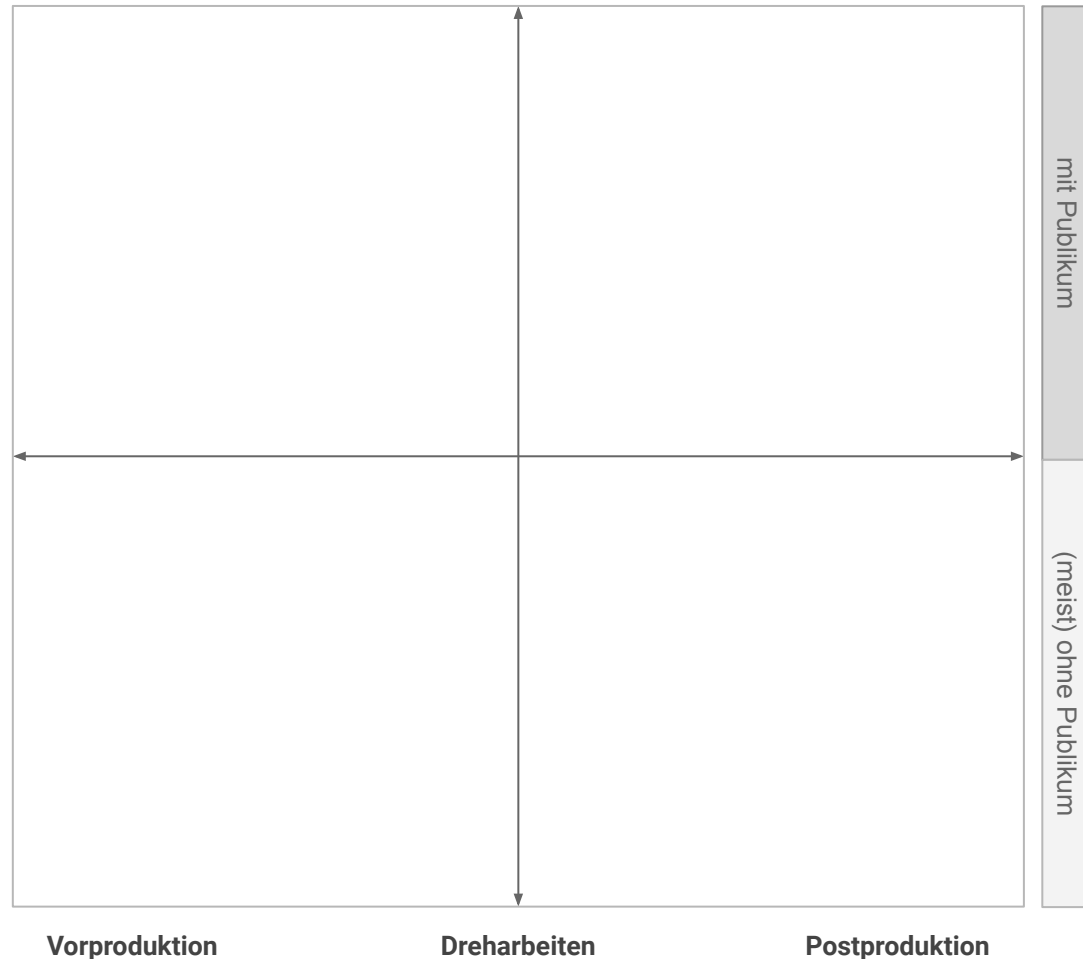
Medien lassen sich nach Phasen
der Produktion (x) und Arten der
Aufzeichnung (y) organisieren.

Aufführung
(eingeschränkt, kurzlebig)

Liveübertragung
(öffentlich, archivierbar)

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)



Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

- Klassische Medien

Aufführung
(eingeschränkt, kurzlebig)

Theater, Konzert, Rede

Liveübertragung
(öffentlich, archivierbar)

Live TV, Radio

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)

Film, Fernsehen

Vorproduktion

Dreharbeiten

Postproduktion

mit Publikum

(meist) ohne Publikum

Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

- Klassische Medien
- Ausgewählte Spezialfälle

Aufführung
(eingeschränkt, kurzlebig)

Theater, Konzert, Rede

Mitschnitt, Bootleg

Liveübertragung
(öffentlich, archivierbar)

Live TV, Radio

Verzögerte
Ausstrahlung

Voraufzeichnung
(ein Take)

Nachrichten, Interviews

Voraufzeichnung
(mehrere Takes)

Film, Fernsehen

Vorproduktion

Dreharbeiten

Postproduktion

mit Publikum

(meist) ohne Publikum

Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

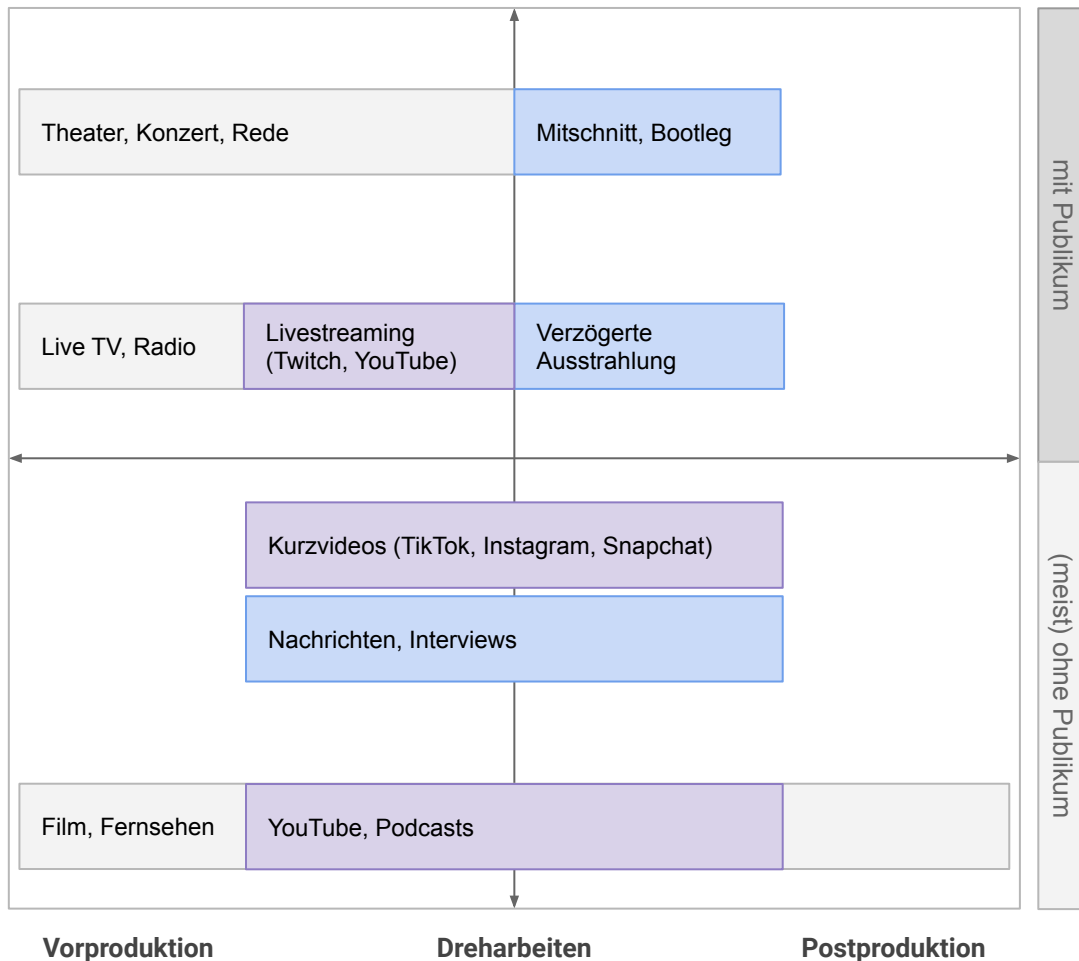
- Klassische Medien
- **Ausgewählte Spezialfälle**
- **Neue Medien**

Aufführung
(eingeschränkt, kurzlebig)

Liveübertragung
(öffentlich, archivierbar)

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)



Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

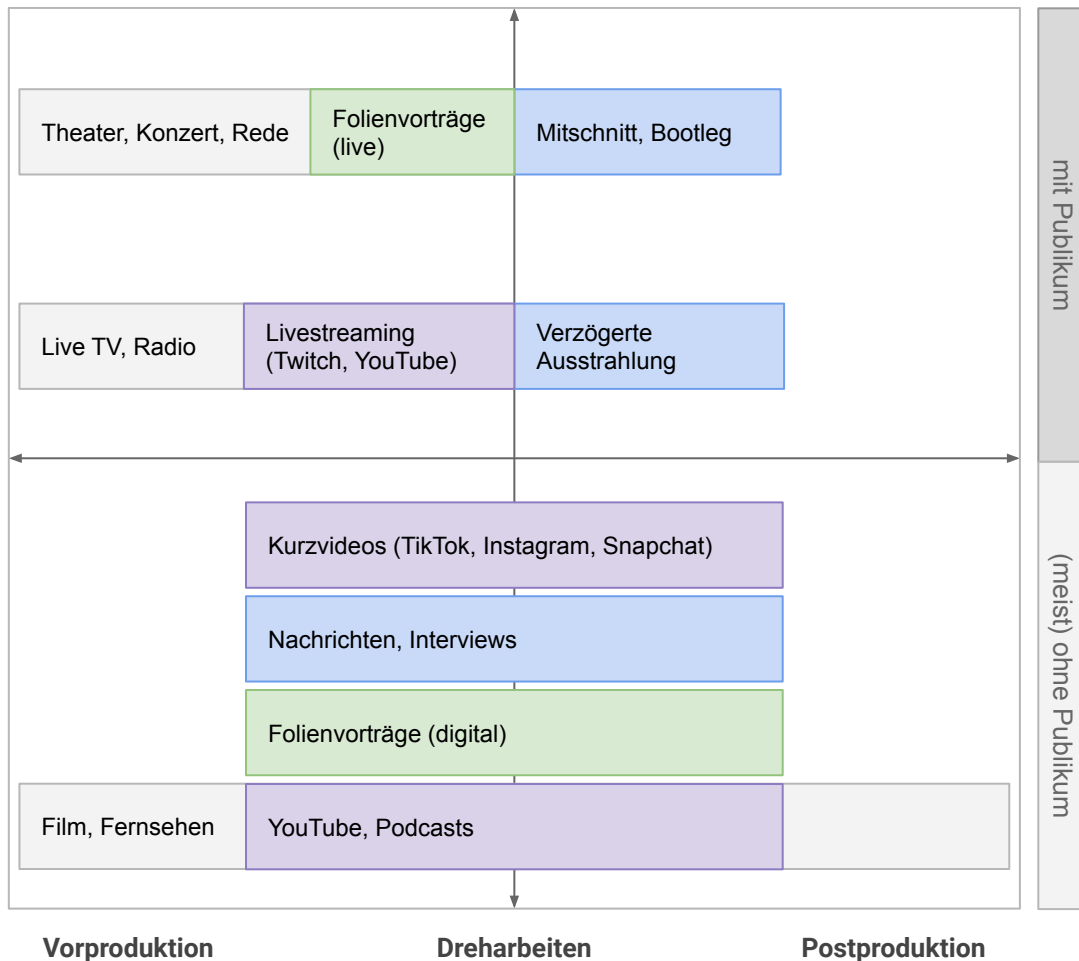
- Klassische Medien
- Ausgewählte Spezialfälle
- Neue Medien
- Wissenschaft und Bildung

Aufführung
(eingeschränkt, kurzlebig)

Liveübertragung
(öffentlich, archivierbar)

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)



Fragen

1. Kann der Produktions-Overhead digitaler Folienvorträge computergestützt reduziert werden?
2. Sind Sprachsynthesemodelle weit genug fortgeschritten, um einen Teil der Arbeit zu übernehmen?

Fragen

1. Kann der Produktions-Overhead digitaler Folienvorträge computergestützt reduziert werden?
2. Sind Sprachsynthesemodelle weit genug fortgeschritten, um einen Teil der Arbeit zu übernehmen?

↓ in Zukunft

3. Können Vorlesungen (teil-)automatisch generiert werden?
4. Können Folienvorträge vollständig aus Büchern (vor-)generiert werden?

Beiträge

1. Entwicklung einer grundlegenden Markup Sprache für Folienvorträge
 - LSML (Lecture Synthesis Markup Language)
2. Referenzimplementierung der LSML als Lecture.js
 - Open Source Toolset als Fundament
3. Evaluierung der Qualität und des praktischen Einsatzes
 - Manuelle Analyse der Aussprache
 - Nutzerstudie

Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

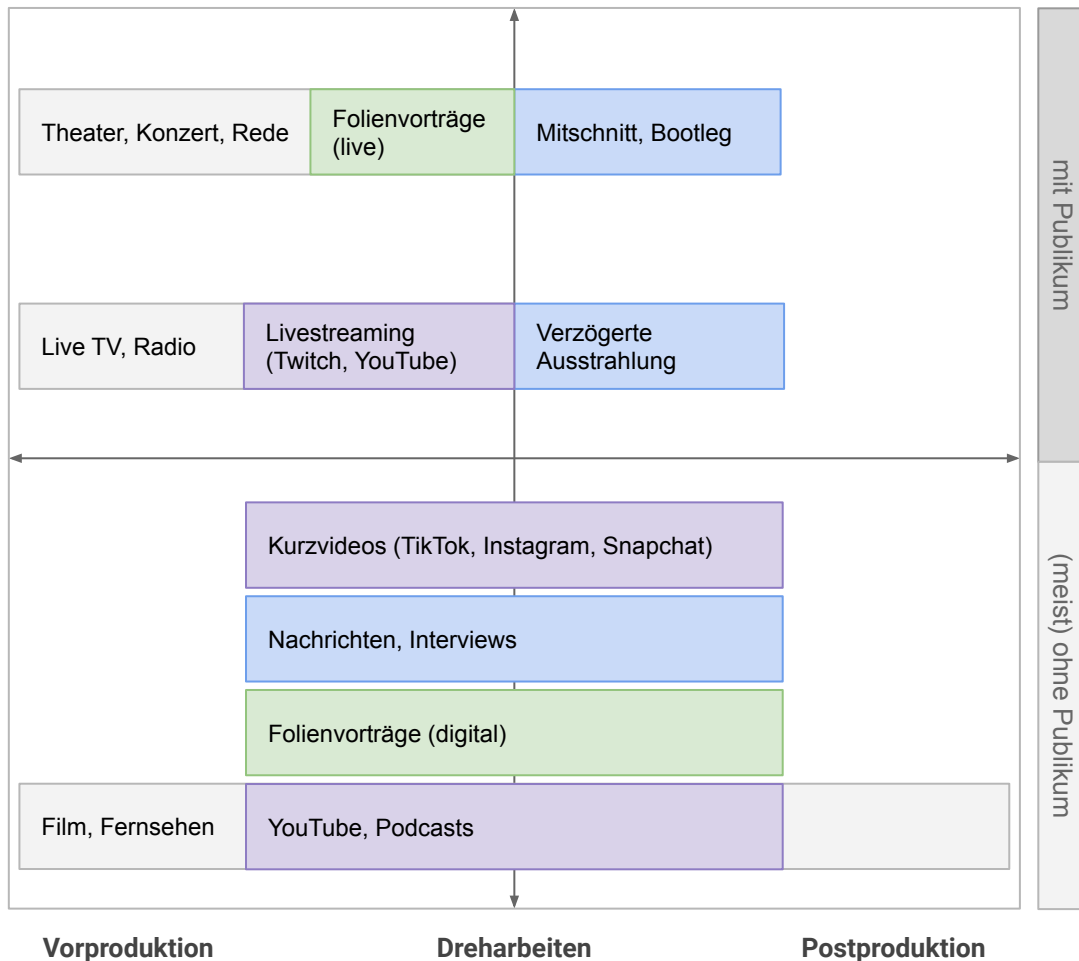
- Klassische Medien
- Ausgewählte Spezialfälle
- Neue Medien
- Wissenschaft und Bildung

Aufführung
(eingeschränkt, kurzlebig)

Liveübertragung
(öffentlich, archivierbar)

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)



Motivation

Medien lassen sich nach Phasen der Produktion (x) und Arten der Aufzeichnung (y) organisieren.

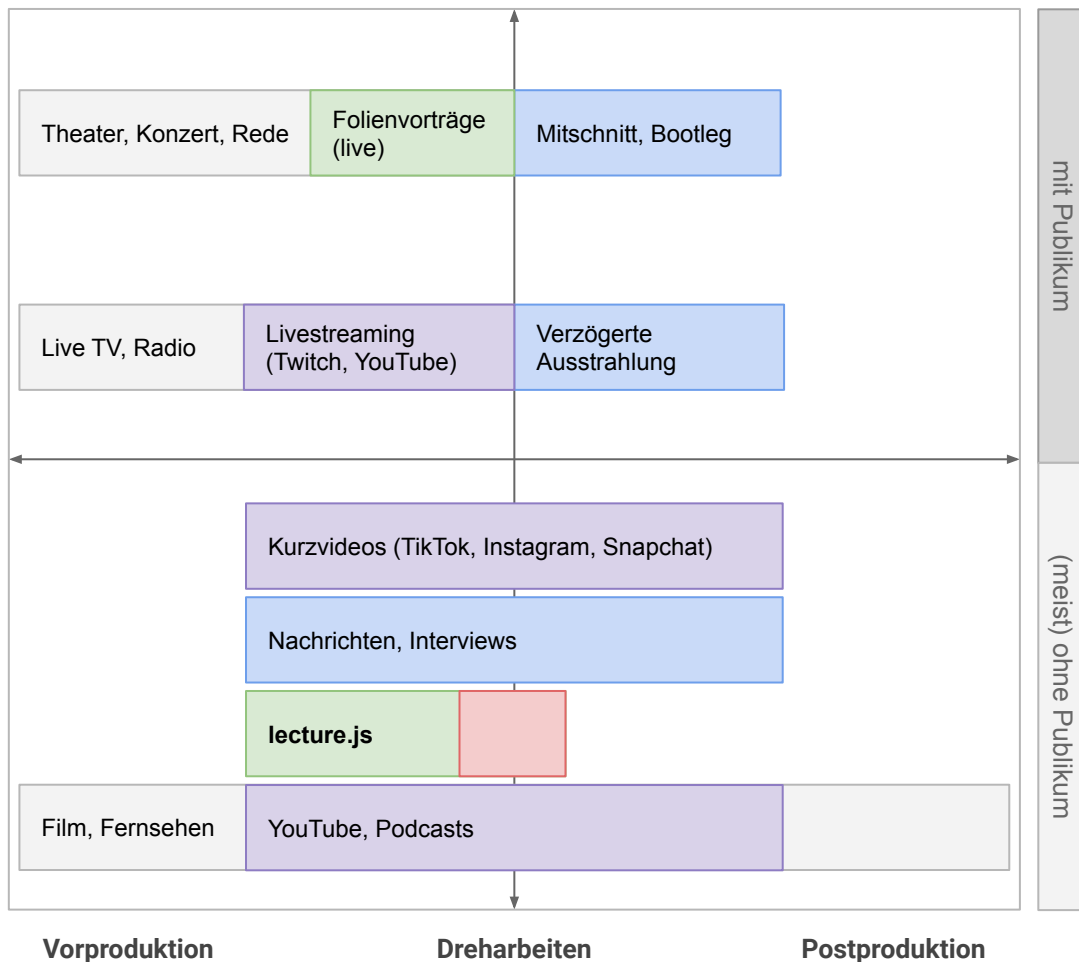
- Klassische Medien
- Ausgewählte Spezialfälle
- Neue Medien
- Wissenschaft und Bildung
- Automatische Generierung

Aufführung
(eingeschränkt, kurzlebig)

Liveübertragung
(öffentlich, archivierbar)

Voraufzeichnung
(ein Take)

Voraufzeichnung
(mehrere Takes)





1. Motivation und Anforderungen

2. LSML

3. Implementierung

4. Evaluierung

5. Zukünftige Arbeit

6. Zusammenfassung

Demonstration

```
1 ▼ <lecture>
2
3   <info
4     title="Eine Einfuehrung in LSML"
5     description="Dies ist eine kurze Einfuehrung in die LSML für die Verteidigung meiner
6       Bachelorarbeit."
7     authors="Christian Dunkel"
8   />
9   <settings
10     voice="amazon-de-de-vicki"
11     resolution="1920x1080"
12     fps="30"
13     breakAfterSlide="1300"
14     breakAfterParagraph="550"
15   />
16   <deck id="slides" src="defense.pdf" active="true" />
17
18   <slide page="14" />
19
20   Willkommen zu einer Demonstration der Funktionen der Markup Sprache <say-as interpret-
21     as="characters">LSML</say-as>.
22
23   <slide page="15" />
24
25   Hier seht ihr den Programmcode der gerade verlesen wird. Aber bevor wir uns mit Code
26   auseinandersetzen, kommen wir erst einmal zum Wesentlichen.
```

LSML

Lecture Synthesis Markup Language

LSML ist eine Sprache um Vorlesungen visuell und audial, sowie menschen- und maschinenlesbar, in Textform zu definieren.

LSML

Lecture Synthesis Markup Language

LSML ist eine Sprache um Vorlesungen visuell und audial, sowie menschen- und maschinenlesbar, in Textform zu definieren.

- Basiert auf der hierarchisch strukturierten Markup Sprache XML
- SSML (Speech Synthesis Markup Language) ist eine Auszeichnungssprache für Sprachsynthese
- SSML findet Verwendung bei vielen *Text-to-Speech* Diensten, darunter *Amazon Polly* und *Google Cloud Text-to-Speech*, welche in *Lecture.js* integriert werden sollen

SSML

Dokument Struktur	Text Struktur	Sprachen und Stimmen	Manipulation der Aussprache	Externe Ressourcen
<code>< speak ></code> <code>< meta /></code> <code>< metadata ></code> <code>< mark ></code>	<code>< p ></code> <code>< s ></code> <code>< token ></code> <code>< w ></code>	<code>< voice ></code> <code>< lang ></code>	<code>< lexicon ></code> <code>< lookup ></code> <code>< say-as ></code> <code>< sub ></code> <code>< phoneme ></code> <code>< emphasis ></code> <code>< break /></code> <code>< prosody ></code>	<code>< audio ></code> <code>< desc ></code>

LSML

Dokument Struktur	Text Struktur	Sprachen und Stimmen	Manipulation der Aussprache	Externe Ressourcen
<code>< speak ></code> <code>< meta /></code> <code>< metadata ></code> <code>< mark ></code>	<code>< p ></code> <code>< s ></code> <code>< token ></code> <code>< w ></code>	<code>< voice ></code> <code>< lang ></code>	<code>< lexicon ></code> <code>< lookup ></code> <code>< say-as ></code> <code>< sub ></code> <code>< phoneme ></code> <code>< emphasis ></code> <code>< break /></code> <code>< prosody ></code>	<code>< audio ></code> <code>< desc ></code>

LSML

SSML (Original)

LSML (Geändert)

LSML (Entfernt)

Dokument Struktur	Text Struktur	Sprachen und Stimmen	Manipulation der Aussprache	Externe Ressourcen
<peak> <meta /> <metadata> <mark>	<p> <s> <token> <w>	<voice> <lang>	<lexicon> <lookup> <say-as> <sub> <phoneme> <emphasis> <break /> <prosody>	<audio> <desc>

LSML

SSML (Original)

LSML (Geändert)

LSML (Entfernt)

LSML (Neu)

Dokument Struktur	Text Struktur	Sprachen und Stimmen	Manipulation der Aussprache	Externe Ressourcen
<speech> <meta /> <metadata> <mark> <lecture> <info /> <settings /> <deck /> <slide />	<p> <s> <token> <w>	<voice> <lang> <de> <de-DE> <en> <en-AU> <en-GB> <en-GB-WLS> ...	<lexicon> <lookup> <say-as> <sub> <phoneme> <emphasis> <break /> <prosody>	<audio> <desc> <video /> <image />

```
1 <lecture>
```

```
2 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3   Das wird gesprochen.
4 </lecture>
```


Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3   Das wird gesprochen.
4 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3   <p>
4     <s>Das wird gesprochen.</s>
5   </p>
6 </lecture>
```

Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3   Das wird gesprochen.
4 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   Das wird gesprochen.
5   <slide page="2" />
6   Das wird auf Seite 2 gesprochen.
7 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3   <deck id="slides2" src="slides2.pdf" />
4
5   Das wird gesprochen.
6   <slide page="2" />
7   Das wird auf Seite 2 gesprochen.
8   <slide deck="slides2" page="1" />
9   Das wird auf Seite 1 des zweiten Foliensatzes gesprochen.
10 </lecture>
```

Slide 1
Page 1

Slide 1
Page 2

Slide 2
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <mark name="intro" chapter="Einleitung" />
5   Das ist die Einleitung.
6
7   <mark name="kapitel-2" chapter="Kapitel 2: Der Anfang" />
8   Das ist Kapitel 2.
9 </lecture>
```

Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <voice name="google-es-es-wavenet-b">
5     Hola como estas?
6   </voice>
7
8   <voice name="amazon-en-us-kimberly">
9     I'm fine! Thank you for asking!
10  </voice>
11 </lecture>
```

Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   Hier wird eine <break time="1500ms" /> Pause eingefügt.
5   Und dieser Satz wird <emphasis level="strong">stark betont.</emphasis>
6 </lecture>
```

Slide 1
Page 1


```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <sub alias="Alpha Beta">AB</sub>
5 </lecture>
```

Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <say-as interpret-as="characters">123</say-as>
5 </lecture>
```

Slide 1
Page 1

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <video src="video.mp4" clipEnd="3s" />
5 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <image src="image.png" />
5   Das ist ein Beispielbild.
6 </lecture>
```

```
1 <lecture>
2   <deck id="slides1" src="slides1.pdf" active="true" />
3
4   <audio src="melody.m4a" />
5 </lecture>
```

1. Motivation und Anforderungen
2. LSML

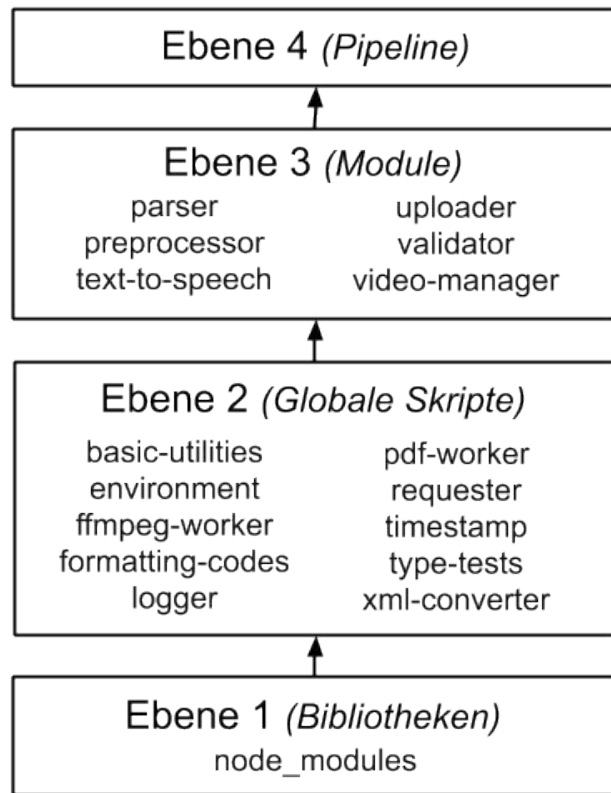
3. Implementierung

4. Evaluierung
5. Zukünftige Arbeit
6. Zusammenfassung

Implementierung

Struktur

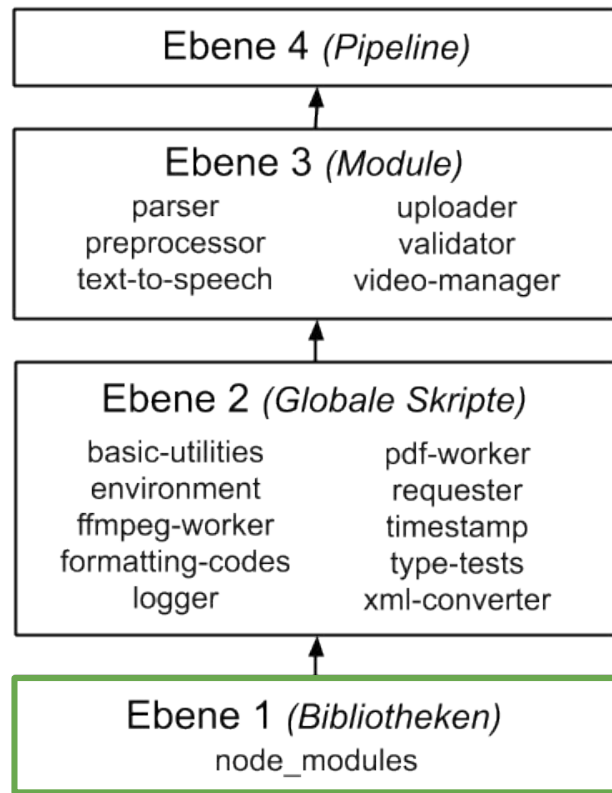
- **Herausforderung:** Erweiterbarkeit, Selbstständigkeit
→ modulare Programmstruktur
- **4 Ebenen**
 - Abhängigkeiten von niedrigeren Ebenen
 - Abhängigkeiten untereinander begrenzen



Implementierung

Struktur

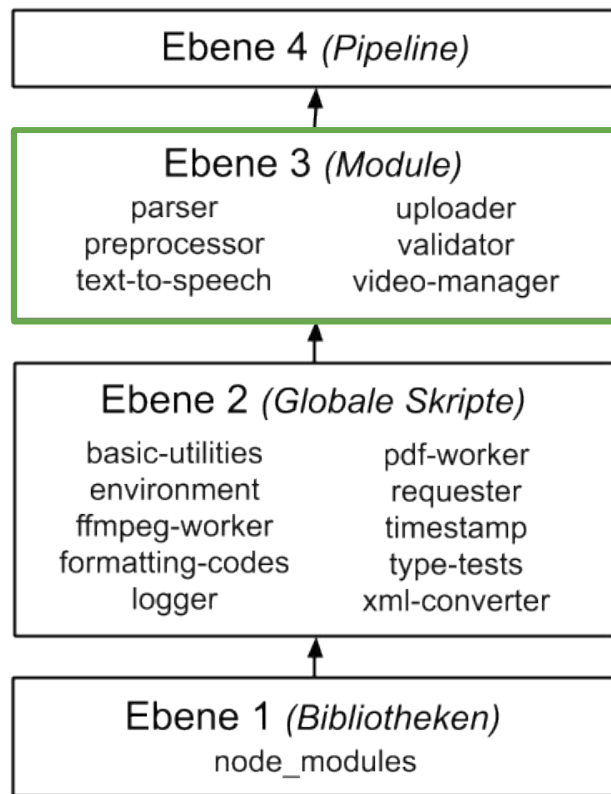
- **Herausforderung:** Erweiterbarkeit, Selbstständigkeit
→ modulare Programmstruktur
- **4 Ebenen**
 - Abhängigkeiten von niedrigeren Ebenen
 - Abhängigkeiten untereinander begrenzen



Implementierung

Struktur

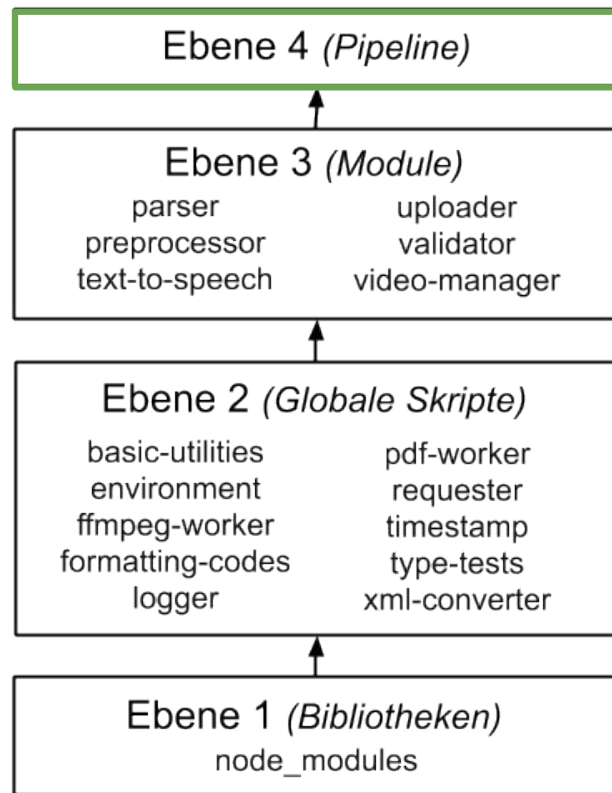
- **Herausforderung:** Erweiterbarkeit, Selbstständigkeit
→ modulare Programmstruktur
- **4 Ebenen**
 - Abhängigkeiten von niedrigeren Ebenen
 - Abhängigkeiten untereinander begrenzen



Implementierung

Struktur

- **Herausforderung:** Erweiterbarkeit, Selbstständigkeit
→ modulare Programmstruktur
- **4 Ebenen**
 - Abhängigkeiten von niedrigeren Ebenen
 - Abhängigkeiten untereinander begrenzen
- **Pipeline**
 - Programm-Konfiguration
 - Anwendung aller Module
 - Großteil des Logging
 - Prozess beenden



Implementierung

Benutzerschnittstelle

- *Information Mode*
 - Ausgabe von Informationen
 - Version
 - Hilfemenü
 - Stimmen
 - Sprachen
- *Sample Mode*
 - Generieren einer Text-zu-Sprache Probe
 - SSML
 - Plaintext
- *Lecture Mode*

Implementierung

Benutzerschnittstelle

- *Information Mode*
 - Ausgabe von Informationen
 - Version
 - Hilfemenü
 - Stimmen
 - Sprachen
- *Sample Mode*
 - Generieren einer Text-zu-Sprache Probe
 - SSML
 - Plaintext
- *Lecture Mode*

```
node lecture.js --voices
```

Implementierung

Benutzerschnittstelle

- *Information Mode*
 - Ausgabe von Informationen
 - Version
 - Hilfemenü
 - Stimmen
 - Sprachen
- *Sample Mode*
 - Generieren einer Text-zu-Sprache Probe
 - SSML
 - Plaintext
- *Lecture Mode*

```
node lecture.js --voices
```

```
node lecture.js -s --voice="amazon-en-us-matthew"  
--text="This is a sentence."
```

Implementierung

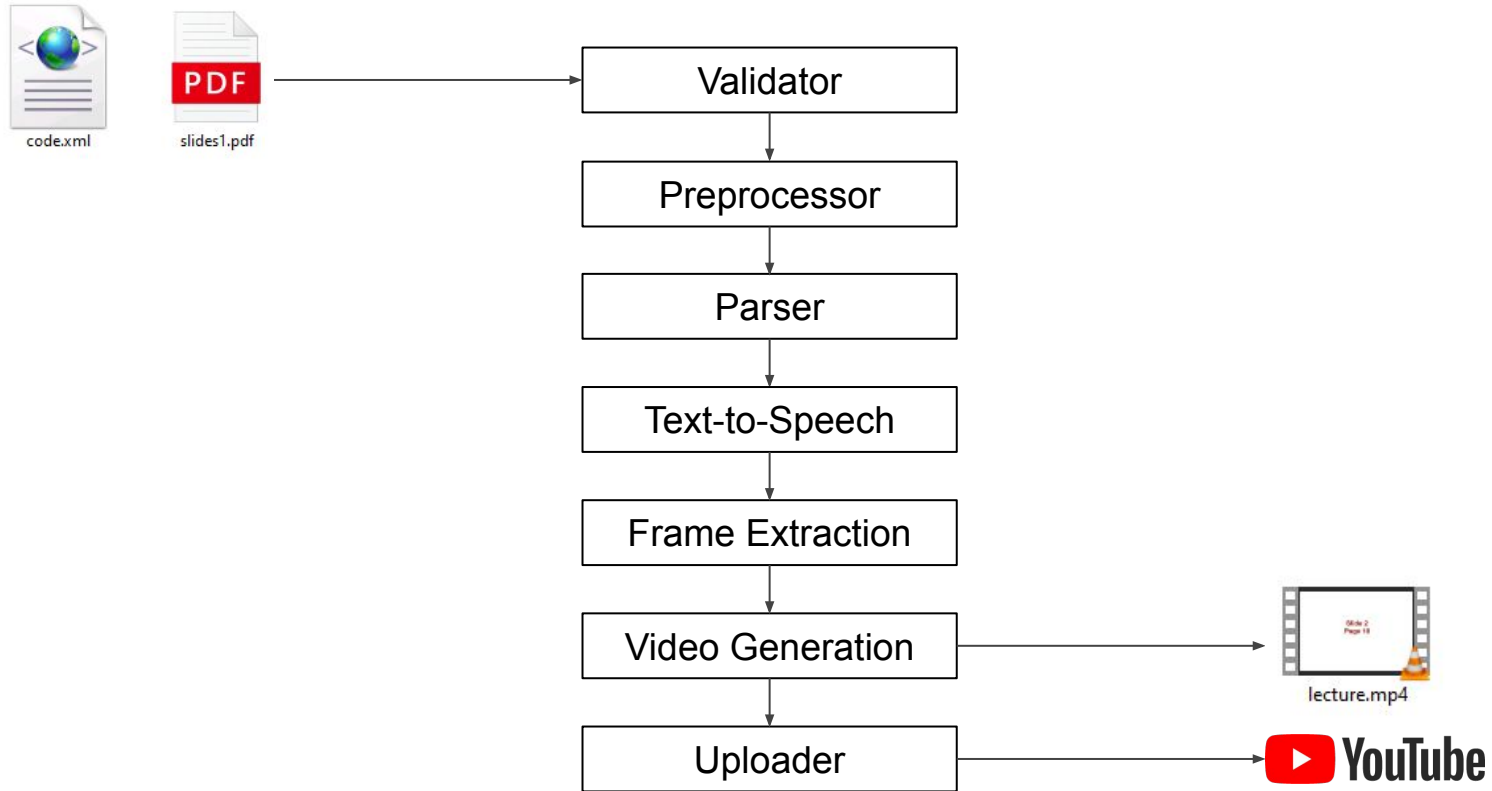
Benutzerschnittstelle

- *Information Mode*
 - Ausgabe von Informationen
 - Version
 - Hilfemenü
 - Stimmen
 - Sprachen
- *Sample Mode*
 - Generieren einer Text-zu-Sprache Probe
 - SSML
 - Plaintext
- *Lecture Mode*

```
node lecture.js --voices
```

```
node lecture.js -s --voice="amazon-en-us-matthew"  
--text="This is a sentence."
```

```
node lecture.js -i="input/example.xml" -o="output/"
```

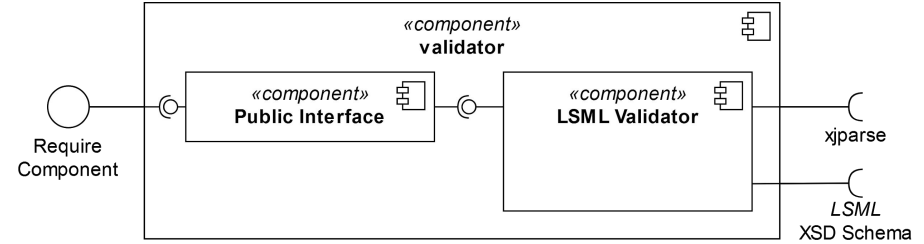





Implementierung

Modul: Validator

- Validation über XSD Schema
- *xjparse* als XSD Parser



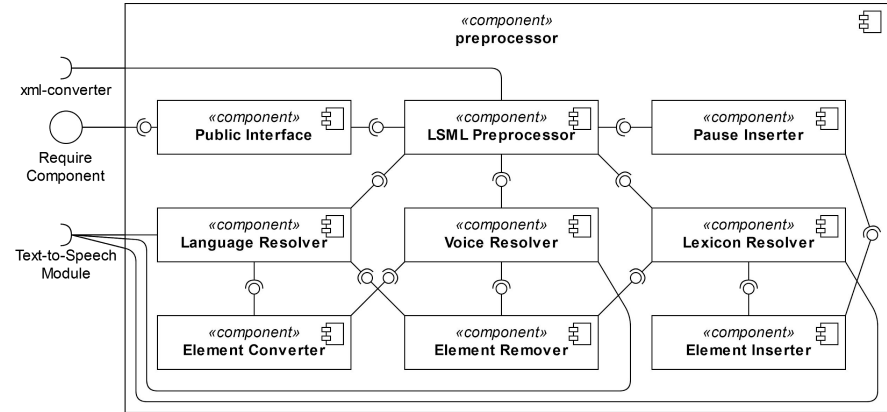
```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3             xmlns:ssml="http://www.w3.org/2001/10/synthesis"
4             xmlns="https://example.com/lecture.js"
5             targetNamespace="https://example.com/lecture.js"
6             elementFormDefault="qualified">
7
8   <xs:annotation>
9     <xs:documentation>
10       LSML Schema
11     </xs:documentation>
12   </xs:annotation>
13
14   <!-- import SSML schema https://www.w3.org/TR/speech-synthesis11/#AppD -->
15   <xs:import namespace="http://www.w3.org/2001/10/synthesis" schemaLocation="synthesis.xsd" />
16
17   <!-- import dependent namespaces, e.g., xml:id -->
18   <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd" />
19
20   <!-- LSML types -->
21
22   <xs:simpleType name="positiveInteger">
23     <xs:restriction base="xs:integer">
24       <xs:minInclusive value="1" />
25     </xs:restriction>
26   </xs:simpleType>
```



Implementierung

Modul: Preprocessor

- Transformation des LSML-Markup
 - Vereinfachte Nutzung
 - Anpassung an Sprachsynthesis-Dienste
- **3 Grundfunktionen**
 - Konvertieren
 - Entfernen
 - Einfügen
- Aufgaben
 - Fügt Pausen ein
 - Entfernt Elemente mit falschen Attributen
 - Konvertiert Sprache-Elemente
 - Wendet Lexikas an

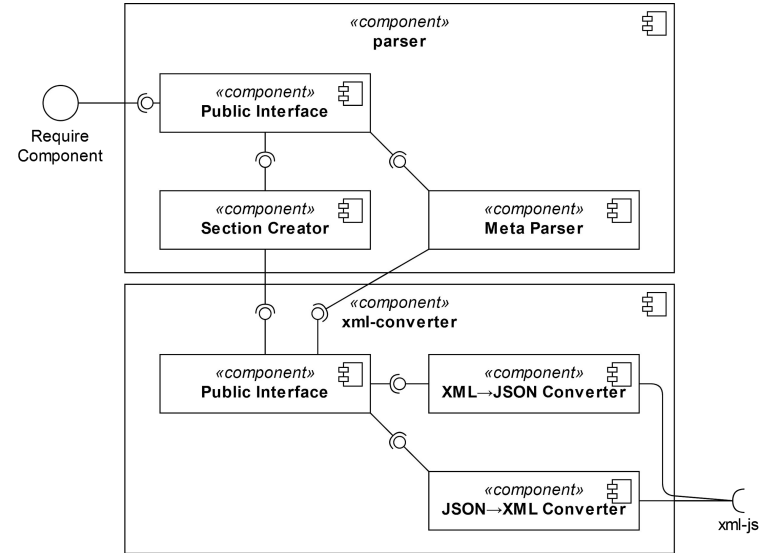




Implementierung

Modul: Parser

- *xml-js* als *Apache Xerces* Wrapper
- Analyse des LSML-Markup
- erstellt internes Datenformat
 - generelle Informationen
 - Einstellungen
 - Foliensätze
 - Lexikas
 - Externe Ressourcen
 - Marker
 - Kapitel
 - **Sektionen**





Implementierung

Generierung von Sektionen

```
<lecture>  
  <deck id="slides" src="slides.pdf" active="true" />  
  
  Hallo, ich werde von der Standard-Stimme gesprochen.  
  
  <voice name="amazon-de-de-marlene">  
    Ich werde von einer anderen Stimme gesprochen.  
  </voice>  
</lecture>
```



Implementierung

Generierung von Sektionen

```
<lecture>
```

```
<deck id="slides" src="slides.pdf" active="true" />
```

Hallo, ich werde von der Standard-Stimme gesprochen.

```
<voice name="amazon-de-de-marlene">
```

Ich werde von einer anderen Stimme gesprochen.

```
</voice>
```

```
</lecture>
```

google-de-de-wavenet-a

Hallo, ich werde von der Standard-Stimme gesprochen.

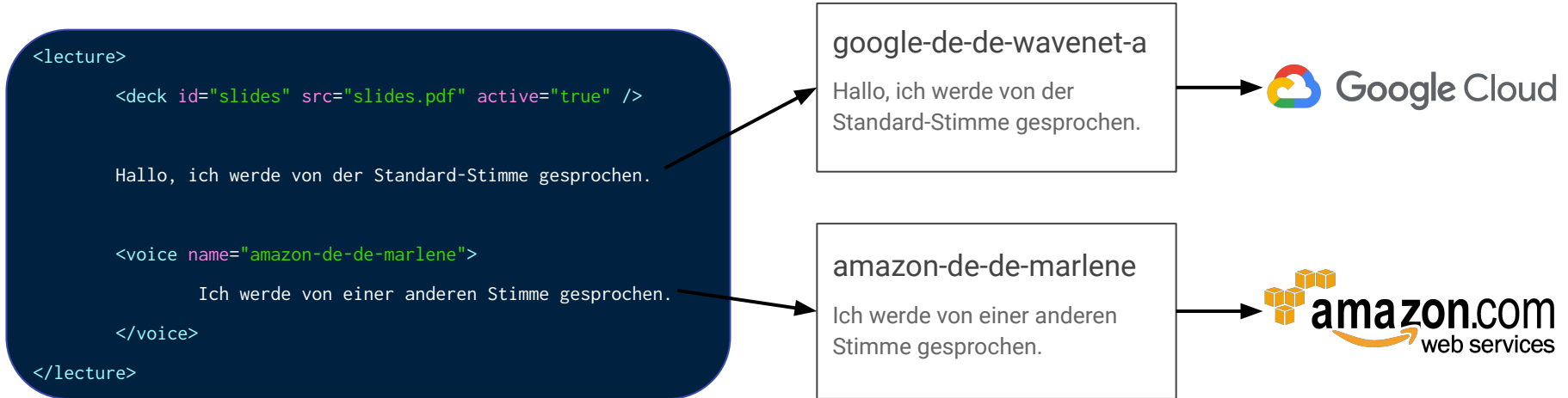
amazon-de-de-marlene

Ich werde von einer anderen Stimme gesprochen.



Implementierung

Generierung von Sektionen

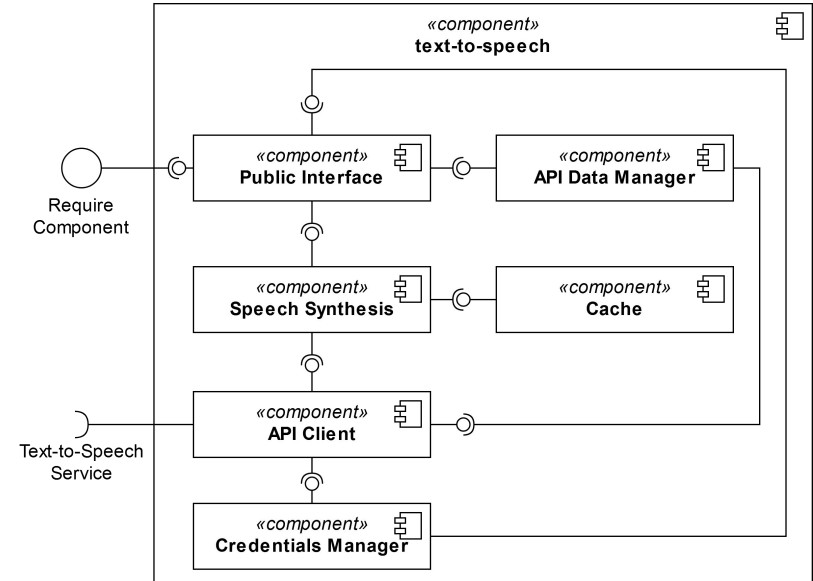




Implementierung

Modul: Text-to-Speech

- Kommunikation mit Amazon Polly und Google Cloud
- löst interne Namen für Stimmen auf
- **Caching**
 - Zeit und Kosten sparen
 - \$4-16 für 1 Millionen Zeichen
 - indiziert und speichert generierte Audio Sektionen





Implementierung

Aufgabe: Frame Extraktion

1. Bild `<image />`



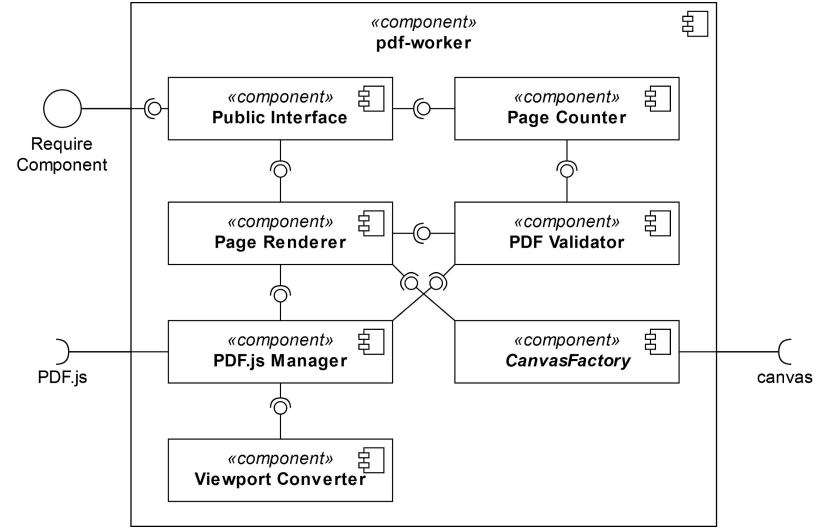
Implementierung

Aufgabe: Frame Extraktion

1. Bild `<image />`

2. Folie `<deck />` `<slide />`

- PDF.js





Implementierung

Aufgabe: Frame Extraktion

1. Bild `<image />`

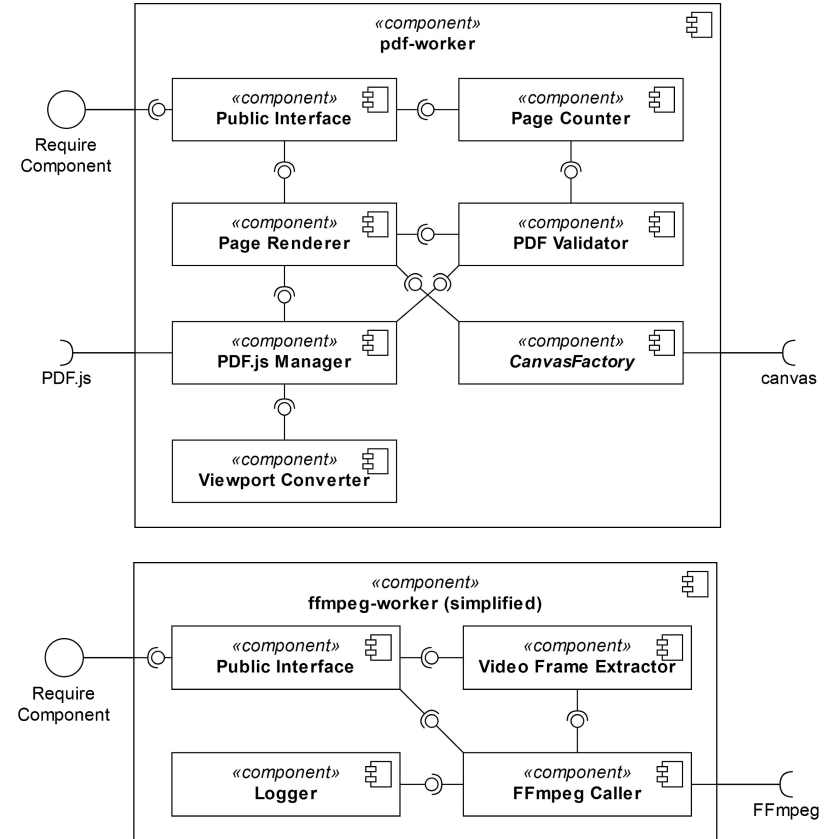
2. Folie `<deck />` `<slide />`

- PDF.js

3. Bleibender Video Frame

`<video keepFrame="true" />`

- FFmpeg

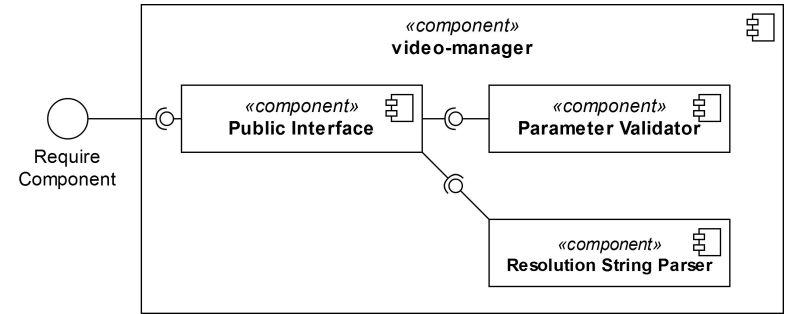


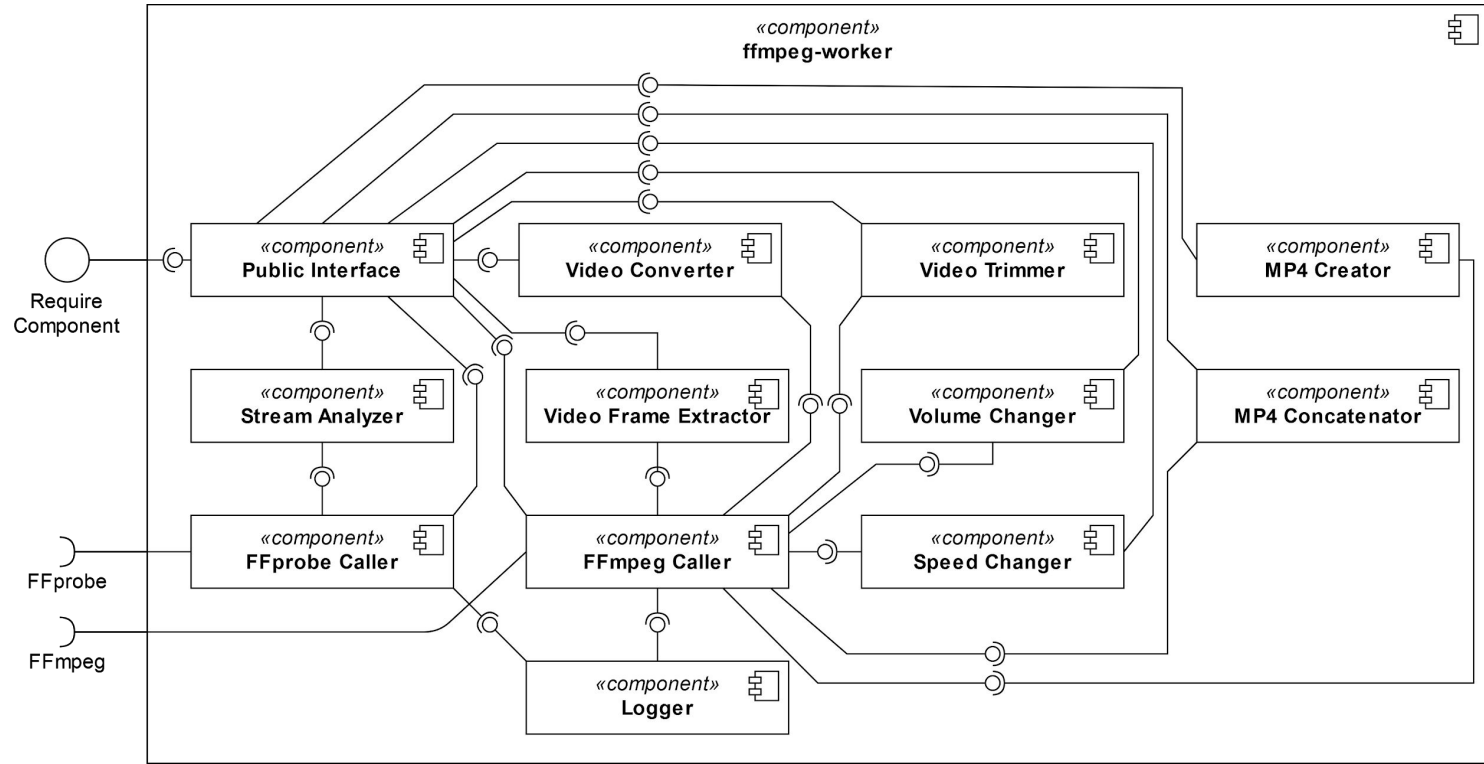


Implementierung

Aufgabe: Video Generation

- **Video Manager:**
 - verwaltet Video Parameter Limits
- **FFmpeg Worker:**
 - erstellt, konvertiert und konkateniert Videos
 - analysiert Video- und Audio-Ströme
 - manipuliert Media Dateien
 - Geschwindigkeit
 - Lautstärke
 - Start-/Endzeit



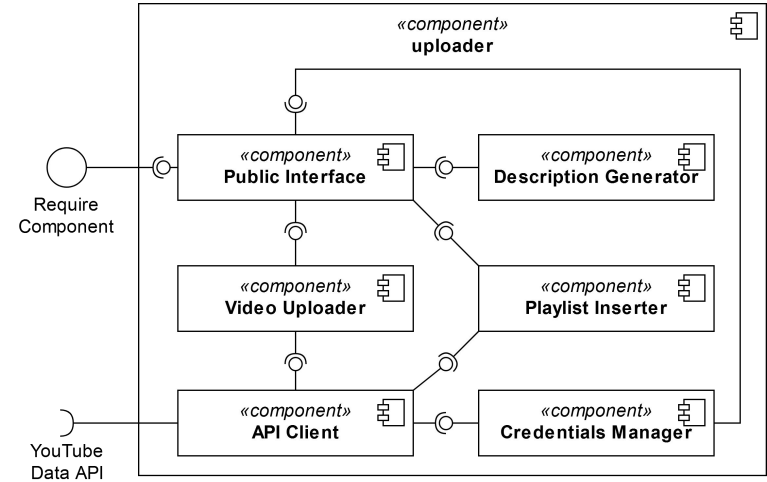




Implementierung

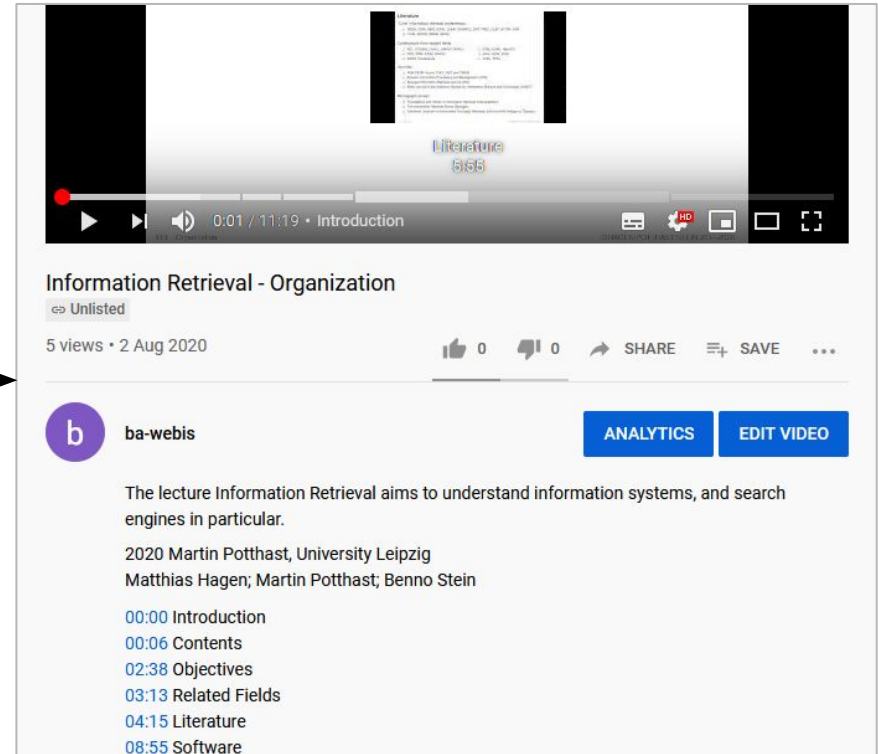
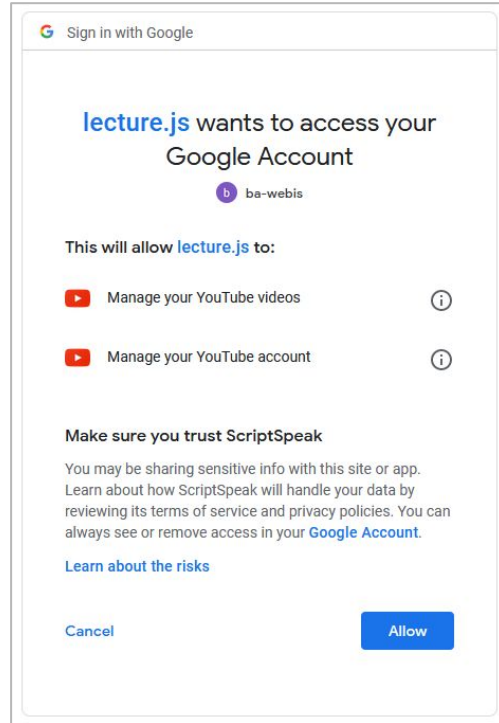
Modul: Uploader

- Kommunikation mit *YouTube Data API*
- Video Beschreibung mit Inhaltsverzeichnis





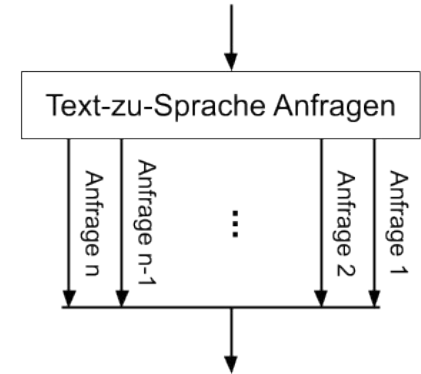
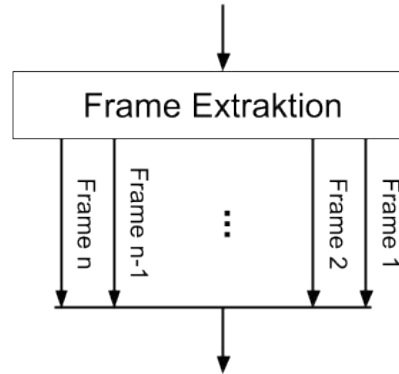
Implementierung



Implementierung

Parallelisierung

- Parallelisierungs Modul
 - per Sekunde
 - in Total
- parallelisierte Aufgaben:
 - Manipulation externer Video und Audio Ressourcen
 - Anfragen zu Text-zu-Sprache APIs
 - Frame Extraktion
 - Generation von Video Sektionen



Implementierung

Style Guide

- JavaScript *ES6* Syntax
- Dokumentstruktur
- Funktionsstruktur
 - asynchrones Verhalten
- Variablendefinition
- Namensgebung

Dokumentation

- LSML Markup
 - Vergleich von Text-to-Speech APIs
- Module API:
 - JSDoc-Kommentare
 - automatisch generiert

Element	Attribute	Description	LSML (lecture.js)	SSML	Amazon Polly	Google Cloud TTS	OpenMARY
<voice>		requests a change in speaking voice	✓ - implements custom voice names to ensure cross-compatibility between APIs	✓	✓ - only available in a basic form that allows for specifying language names	✗ - some of its attributes are available in the options part of the request to the API	✓
	[name]	indicates one or multiple processor-specific voice names to speak the contained text - can be a space-separated list of preferred voices, with the top choice upfront	✓ - only supports specifying a single voice name	✓	✓ - defines the name of an Amazon Polly voice - cannot be a space-separated list of voices - to speak in a different language, combine it with <lang> - supports these voices	✗	✓
	[gender]	indicates the preferred gender of the speaking voice	✗	✓	✗	✗	✓
	[variant]	indicates a preferred variant of voice characteristics as a positive integer, e.g., second male child voice	✗	✓	✗	✗	✓
	[age]	defines the age of the voice	✗	✓	✗	✗	✓
	[languages]	defines a list of languages the voice is desired to speak	✗	✓	✗	✗	✗
	[required]	defines a list of features that should be used by the voice selection algorithm - can be a space-separated list - the default value is "languages"	✗	✓	✗	✗	✗
	[xml:lang]	defines the language	✗	✗	✗	✗	✓

1. Motivation und Anforderungen
2. LSML
3. Implementierung

4. Evaluierung

5. Zukünftige Arbeit
6. Zusammenfassung

Evaluierung

Empirische Analyse der korrekten und inkorrekten Aussprache gesprochener Wörter in **4** verschiedenen Stimmen

- **2 Stimmen (Englisch):**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Neural)
- **2 Stimmen (Deutsch)**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Standard)

3 Texte in **2** Sprachen wurden gerendert und analysiert

Evaluierung

Empirische Analyse der korrekten und inkorrekten Aussprache gesprochener Wörter in **4** verschiedenen Stimmen

- **2 Stimmen (Englisch):**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Neural)
- **2 Stimmen (Deutsch)**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Standard)

3 Texte in **2** Sprachen wurden gerendert und analysiert

<lecture>

The contents of the lecture are divided into 10 chapters. Following the introduction, the second chapter gives an overview of the architecture and components of a search engine using the example of web search.

Text 1 - Beispiel Vorlesung

Evaluierung

Empirische Analyse der korrekten und inkorrekten Aussprache gesprochener Wörter in **4** verschiedenen Stimmen

- **2 Stimmen (Englisch):**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Neural)
- **2 Stimmen (Deutsch)**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Standard)

3 Texte in **2** Sprachen wurden gerendert und analysiert

<lecture>

The contents of the lecture are divided into 10 chapters. Following the introduction, the second chapter gives an overview of the architecture and components of a search engine using the example of web search.

Text 1 - Beispiel Vorlesung

<lecture>

In today's digital world, it is becoming increasingly important to integrate digitalization into the classroom to reap the advantages of greater interactivity and distance learning.

Text 2 - Thesis Kapitel 1/2

Evaluierung

Empirische Analyse der korrekten und inkorrekten Aussprache gesprochener Wörter in **4** verschiedenen Stimmen

- **2 Stimmen (Englisch):**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Neural)
- **2 Stimmen (Deutsch)**
 - **1** (Google Cloud, WaveNet)
 - **1** (Amazon Polly, Standard)

3 Texte in **2** Sprachen wurden gerendert und analysiert

<lecture>

The contents of the lecture are divided into 10 chapters. Following the introduction, the second chapter gives an overview of the architecture and components of a search engine using the example of web search.

Text 1 - Beispiel Vorlesung

<lecture>

In today's digital world, it is becoming increasingly important to integrate digitalization into the classroom to reap the advantages of greater interactivity and distance learning.

Text 2 - Thesis Kapitel 1/2

<lecture>

From can't night beast. Us sixth which rule blessed you're female one thing greater gathering. Face from heaven subdue itself herb forth second upon kind moveth forth air.

Text 3 - Zufällige Wörter

Evaluierung

Versuchsaufbau

- Texte wurden als Plaintext eingespeist ohne LSML-Elemente die die Aussprache beeinflussen
- Ein Programm wurde benutzt um schnell Zeitstempel im Video zu erfassen ohne große Ablenkungen

Limitierungen

- Die Analyse ist subjektiv (nur 1 Testsubjekt)
- Das Testsubjekt ist kein englischer Muttersprachler, was den Sprachschatz limitiert, womit es zu *false positives* und *false negatives* kommen kann

Evaluierung

	Beispiel Vorlesung 1228 (en) 1239 (de)	Thesis Kapitel 1/2 1669 (en) 1657 (de)	Zufällige Wörter 1300 (en/de)	Total 4197 (en) 4196 (de)
amazon-en-gb-amy <i>Neural, weiblich</i>	1210 Korrekt 12 Auffällig, 6 Inkorrekt	1655 Korrekt 8 Auffällig, 6 Inkorrekt	1295 Korrekt 3 Auffällig, 2 Inkorrekt	4160 Korrekt 23 Auffällig, 14 Inkorrekt
google-en-gb-wavenet-f <i>WaveNet, weiblich</i>	1223 Korrekt 2 Auffällig, 3 Inkorrekt	1664 Korrekt 3 Auffällig, 2 Inkorrekt	1299 Korrekt 1 Auffällig, 0 Inkorrekt	4186 Korrekt 6 Auffällig, 5 Inkorrekt
amazon-de-de-vicki <i>Standard, weiblich</i>	1208 Korrekt 29 Auffällig, 2 Inkorrekt	1642 Korrekt 11 Auffällig, 4 Inkorrekt	1300 Korrekt 0 Auffällig, 0 Inkorrekt	4150 Korrekt 40 Auffällig, 6 Inkorrekt
google-de-de-wavenet-a <i>WaveNet, weiblich</i>	1217 Korrekt 22 Auffällig, 0 Inkorrekt	1652 Korrekt 5 Auffällig, 0 Inkorrekt	1300 Korrekt 0 Auffällig, 0 Inkorrekt	4169 Korrekt 27 Auffällig, 0 Inkorrekt

Evaluierung

$$\text{Wertung} = \frac{T_{\text{Korrekt}} \cdot 1 + T_{\text{Auffällig}} \cdot 0.5 + T_{\text{Inkorrekt}} \cdot 0}{T_{\text{Alle}}}$$

$$\text{Wertung} = \frac{1200 \cdot 1 + 70 \cdot 0.5 + 30 \cdot 0}{1300} = 0.95$$

Evaluierung

	Beispiel Vorlesung 1228 (en) 1239 (de)	Thesis Kapitel 1/2 1669 (en) 1657 (de)	Zufällige Wörter 1300 (en/de)	Total 4197 (en) 4196 (de)
amazon-en-gb-amy <i>Neural, weiblich</i>	0.9902	0.9940	0.9973	0.9939
google-en-gb-wavenet-f <i>WaveNet, weiblich</i>	0.9967	0.9979	0.9996	0.9980
amazon-de-de-vicki <i>Standard, weiblich</i>	0.9866	0.9942	1	0.9938
google-de-de-wavenet-a <i>WaveNet, weiblich</i>	0.9911	0.9984	1	0.9967

Evaluierung

Nutzerstudie

- **27** Gruppen wurden angeschrieben
- Teilnehmer des PAN Workshops (CLEF 2020)
 - Lecture.js mit allen nötigen Funktionen
 - Credentials für Amazon Polly / Google Cloud
 - Nutzungsanleitung und Dokumentation
- **Auswertung**
 - Fragebogen mit **14** Fragen
 - Wenig Antworten, dafür aber immerhin **4** Teilnehmer in der finalen Studie
 - Das Werkzeug wurde auch in einem Hauptvortrag für die Konferenz getestet

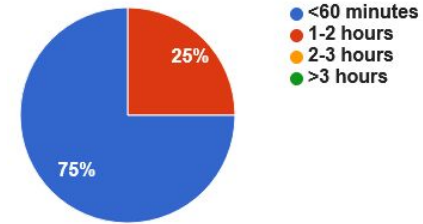
https://www.youtube.com/watch?v=699e_l6XDI8&list=PLqD1TOdHQCl_R9aX5MvoM8kMqz_YECTke

Evaluierung

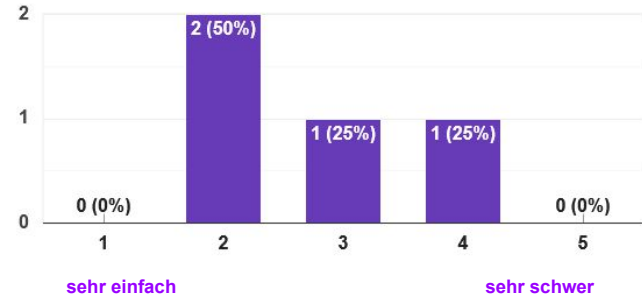
Nutzerstudie

- **27** Gruppen wurden angeschrieben
- Teilnehmer des PAN Workshops (CLEF 2020)
 - Lecture.js mit allen nötigen Funktionen
 - Credentials für Amazon Polly / Google Cloud
 - Nutzungsanleitung und Dokumentation
- **Auswertung**
 - Fragebogen mit **14** Fragen
 - Wenig Antworten, dafür aber immerhin **4** Teilnehmer in der finalen Studie
 - Das Werkzeug wurde auch in einem Hauptvortrag für die Konferenz getestet

https://www.youtube.com/watch?v=699e_l6XDI8&list=PLqD1TOdHQCl_R9aX5MvoM8kMqz_YECTke



Wie lange hast du gebraucht, um dich mit der Software vertraut zu machen?



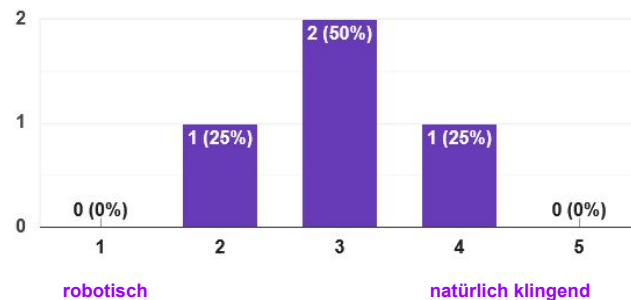
Wie schwierig war die Anwendung für dich?

Evaluierung

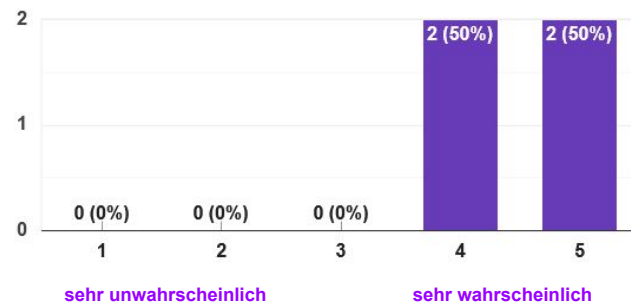
Nutzerstudie

- **27** Gruppen wurden angeschrieben
- Teilnehmer des PAN Workshops (CLEF 2020)
 - Lecture.js mit allen nötigen Funktionen
 - Credentials für Amazon Polly / Google Cloud
 - Nutzungsanleitung und Dokumentation
- **Auswertung**
 - Fragebogen mit **14** Fragen
 - Wenig Antworten, dafür aber immerhin **4** Teilnehmer in der finalen Studie
 - Das Werkzeug wurde auch in einem Hauptvortrag für die Konferenz getestet

https://www.youtube.com/watch?v=699e_l6XDI8&list=PLqD1TQdHQCl_R9aX5MvoM8kMqz_YECTke



Wie empfandest du die Qualität der Stimmen?



Würdest du die Software wieder verwenden?

Evaluierung

Feedback

- (1) Vereinfachung von LSML
- (2) Integration von Bild-Ressourcen
- (3) Überlagerung von Audio- und Video-Ressourcen
- (4) Vorschau einzelner Sektionen
- (5) Integration von PDF *Named Destinations*

1. Motivation und Anforderungen
2. LSML
3. Implementierung
4. Evaluierung

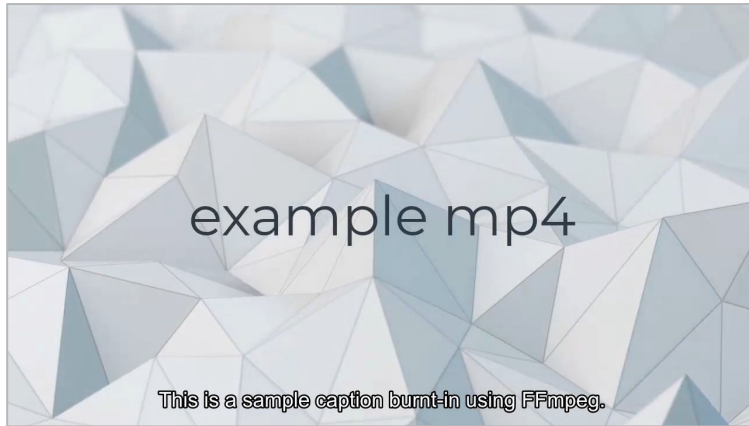
5. Zukünftige Arbeit

6. Zusammenfassung

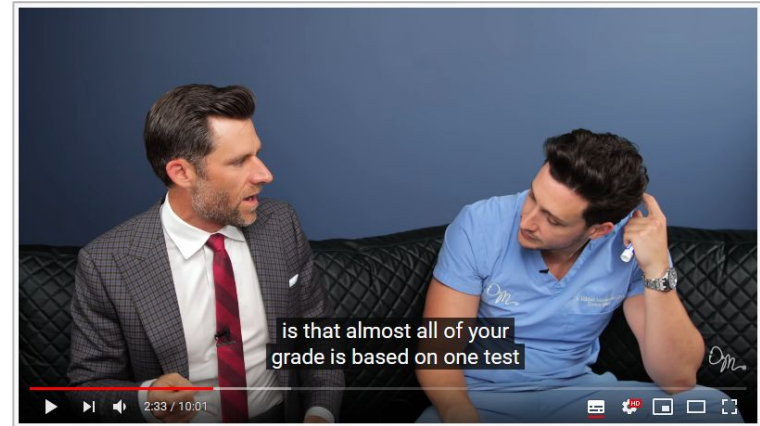
Zukünftige Arbeit

Untertitel

Untertitel bieten eine Unterstützung für gehörlose oder schwerhörige Studenten



Untertitel eingebrannt in ein Video durch FFmpeg



Schwebende Untertitel auf YouTube

youtube.com/watch?v=z9e8GcQvNHo

Workflow

LSML

Der derzeitige Arbeitsablauf wenn man lecture.js verwendet ist kompliziert für Nicht-Programmierer und unnötig zeitaufwendig in bestimmten Aspekten

1. Vorlesungsfolien erstellen
2. Skript fehlerfrei ausformulieren
3. LSML-Elemente einfügen
 - externer Text-Editor
4. Generierte Vorlesung auf Fehler kontrollieren

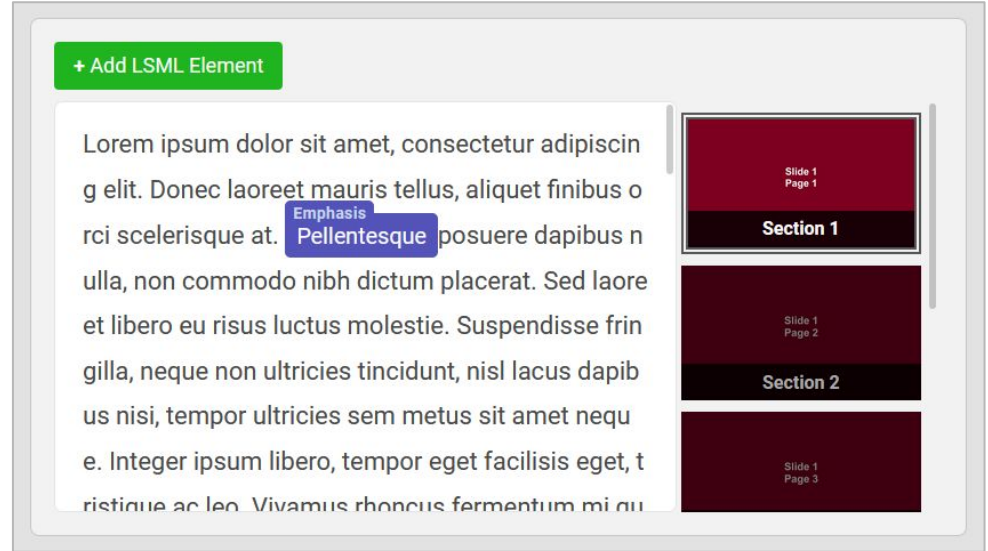
```
19
20 <deck id="organization" src="unit-en-ir-organization.pdf" active="true" />
21
22 MOP. UNIT EN IR ORGANIZATION NOTES.
23 DATE. May 24th, 2020.
24
25 <mark name="intro" chapter="Introduction" />
26
27 <en-GB>Information Retrieval</en-GB>
28 Eine Vorlesung von Matthias Hagen, Martin Potthast und Benno Stein.
29
30 <mark name="contents" chapter="Contents" />
31
32 <slide page="2" />
33
34 Die Inhalte der Vorlesung sind in 10 Kapitel unterteilt.
35
36 Im Anschluss an die Einleitung gibt das zweite Kapitel einen Überblick über die
37 Architektur und Komponenten einer Suchmaschine am Beispiel der Websuche.
38 Die drei folgenden Kapitel befassen sich mit der Akquise von Daten aus dem Web, der
39 Analyse und Aufbereitung von so gesammelten Textdaten, sowie der effizienten
  Datenverwaltung in Form von darauf spezialisierten Algorithmen und Datenstrukturen.
```

LSML-Code in der *Brackets* Entwicklungsumgebung

Zukünftige Arbeit

Grafische Benutzeroberfläche

- Versteckt LSML-Syntax in graphischen Elementen
- Ermöglicht die Vorschau einzelner Sektionen der Vorlesung



Modell einer einfachen GUI

Zukünftige Arbeit

Grafische Benutzeroberfläche

- Versteckt LSML-Syntax in graphischen Elementen
- Ermöglicht die Vorschau einzelner Sektionen der Vorlesung



Hervorgehobene Sektion

Follow-up Studien

- Ist es möglich Signale des Lernenden bei Lernproblemen zu erkennen?
 - Signale wie Pupillen-Erweiterungen und Cursor-Bewegungen könnten dazu verwendet werden Literatur-Vorschläge und Tipps in bestimmten Abschnitten zu geben
 - Die Daten könnten auch als Feedback für den Professor dienen
- Vergleich der Reaktion von Lernenden auf Vorlesungen aufgezeichnet von einem Professor verglichen mit automatisch generierten Vorlesungen
- Vergleich von natürlichen Stimmen und maschinengenerierter Sprache
 - Sind sie in 5 Jahren noch zu unterscheiden?

1. Motivation und Anforderungen
2. LSML
3. Implementierung
4. Evaluierung
5. Zukünftige Arbeit

6. Zusammenfassung

Zusammenfassung

- Entwicklung der grundlegenden Markup Sprache LSML für Vorlesungen
 - Ermöglicht semi-automatisiertes Generieren von Vorlesungen
 - Erweiterung von SSML, basierend auf XML
- Implementierung der LSML als Lecture.js
- Evaluierung über Analyse der Aussprache und eine Nutzerstudie
 - Nutzer empfanden die Software als einfach erlern- und nutzbar und bewerteten die Qualität der Stimmen als zufriedenstellend

Zusammenfassung

- Entwicklung der grundlegenden Markup Sprache LSML für Vorlesungen
 - Ermöglicht semi-automatisiertes Generieren von Vorlesungen
 - Erweiterung von SSML, basierend auf XML
- Implementierung der LSML als Lecture.js
- Evaluierung über Analyse der Aussprache und eine Nutzerstudie
 - Nutzer empfanden die Software als einfach erlern- und nutzbar und bewerteten die Qualität der Stimmen als zufriedenstellend

Zeit für Fragen!