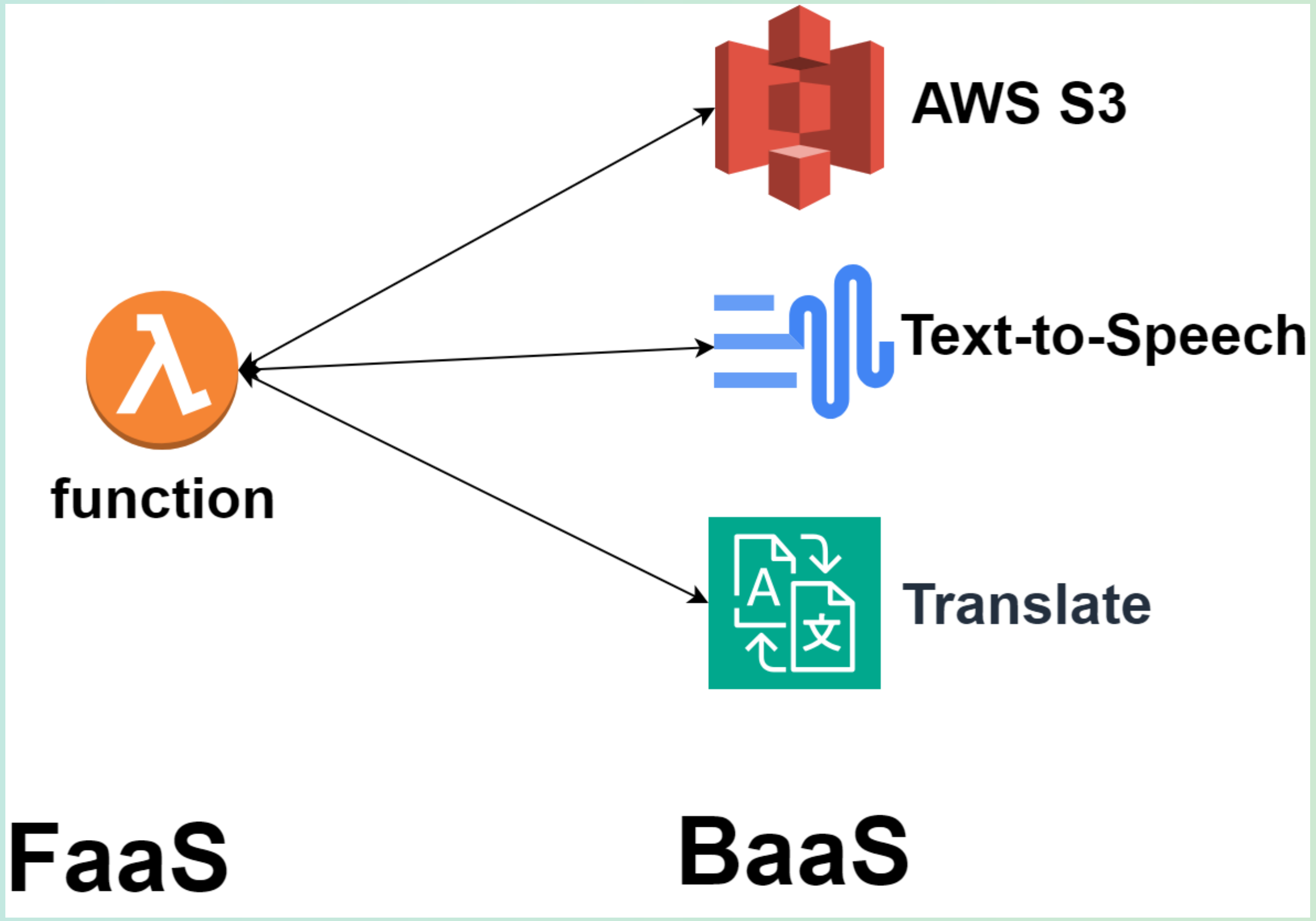
Scale Composite BaaS Services With AFCL Workflows

Thomas Larcher, **Sashko Ristov**

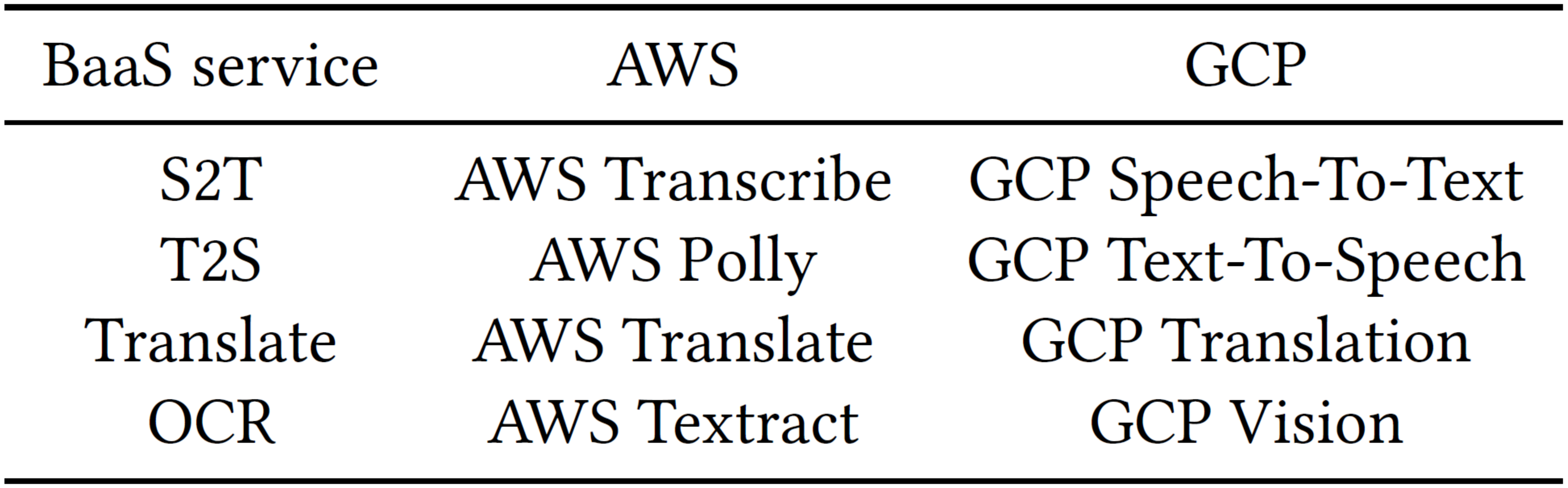
University of Innsbruck, Austria

Serverless = FaaS + BaaS

- **Function-as-a-Service + Backend-as-a-Service**



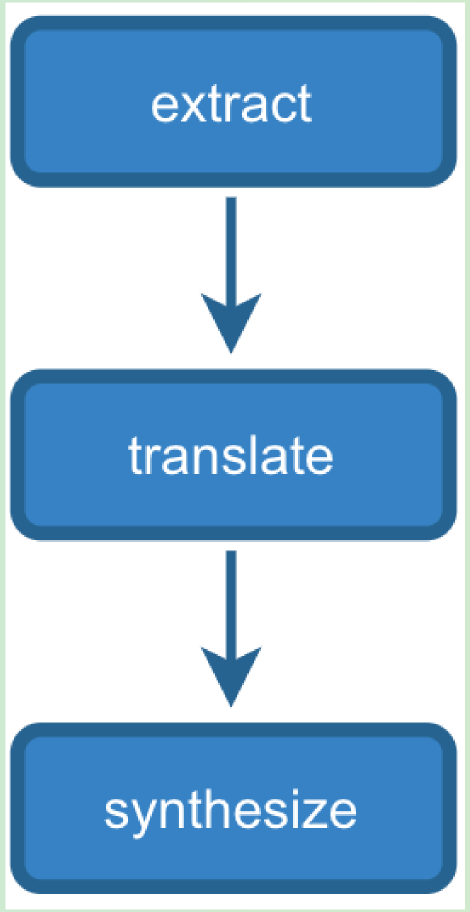
Cloud providers offer various **managed BaaS services**



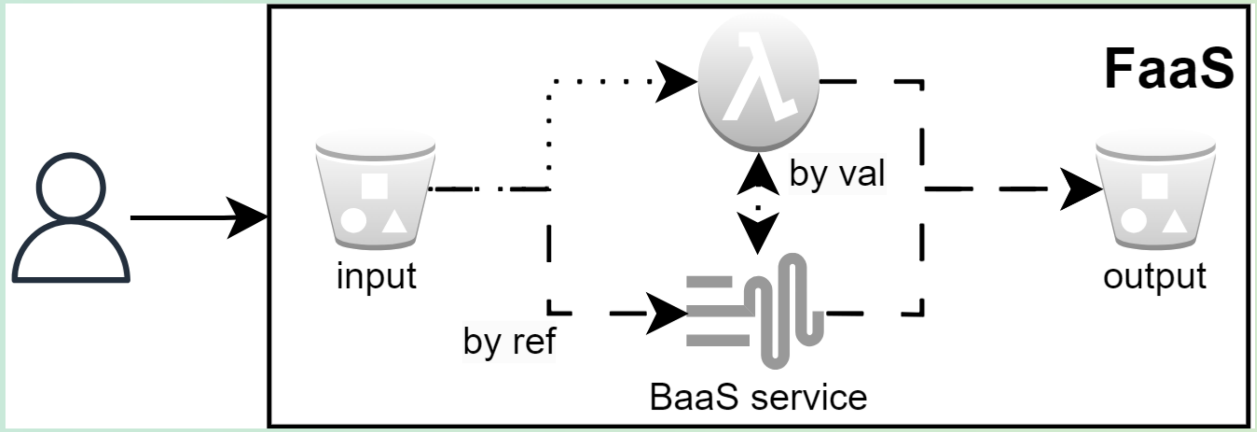
BaaS service	AWS	GCP
S2T	AWS Transcribe	GCP Speech-To-Text
T2S	AWS Polly	GCP Text-To-Speech
Translate	AWS Translate	GCP Translation
OCR	AWS Textract	GCP Vision

Research **Challenges**

- **Lack of complex BaaS services**
  - Create a composite BaaS service by pipelining several BaaS services
- **Limited BaaS composition**
  - limited due to the limited package size
- **Performance bottleneck**
  - OCR is restricted to a single pdf page only
  - restricted HDD space within the function.



BaaS Service **Characteristics**

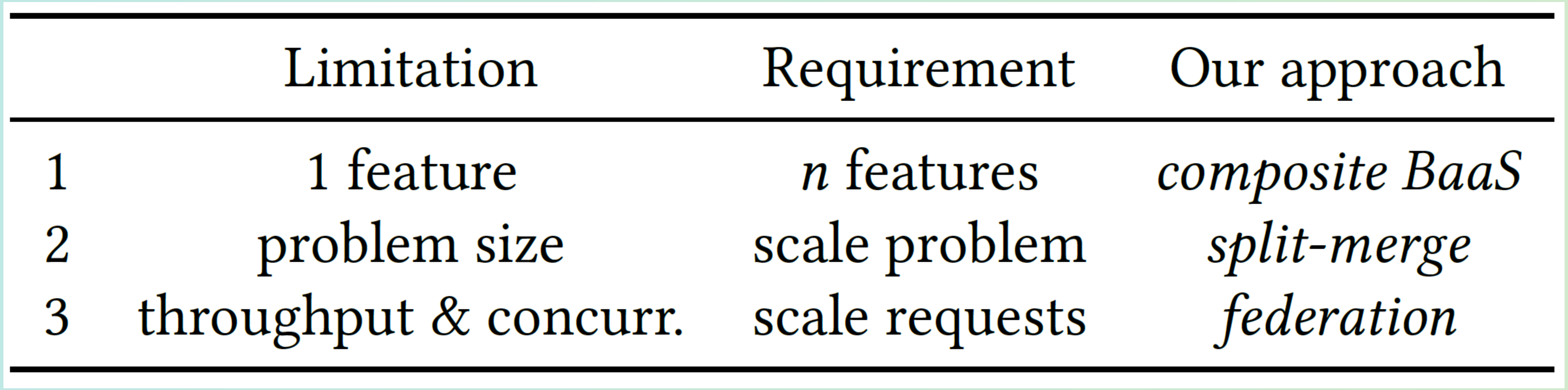


- **Data access type**
  - *by value*
    - the input or output data is submitted as payload
  - *by reference*
    - the BaaS service receives or returns a reference to the object storage
- **Features**
  - *data format*
    - audio, text, pdf, or image
  - *natural language*
    - English, German, …

BaaS service **limitations**

- **Convert only one feature**
  - either *data format*, or
  - the *natural language*,
  - but **not both**
- **Problem size**
  - **Requests** whose problem size is **above the threshold are rejected**
- **Number of requests**
  - Throughput
    - restricts the **number of requests FOR a given time period** (e.g., in a minute)
  - Concurrency
    - restricts the **number of active requests at the moment**

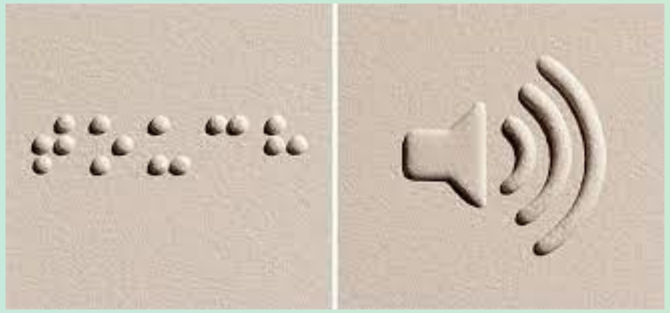
Requirements and Our Approaches

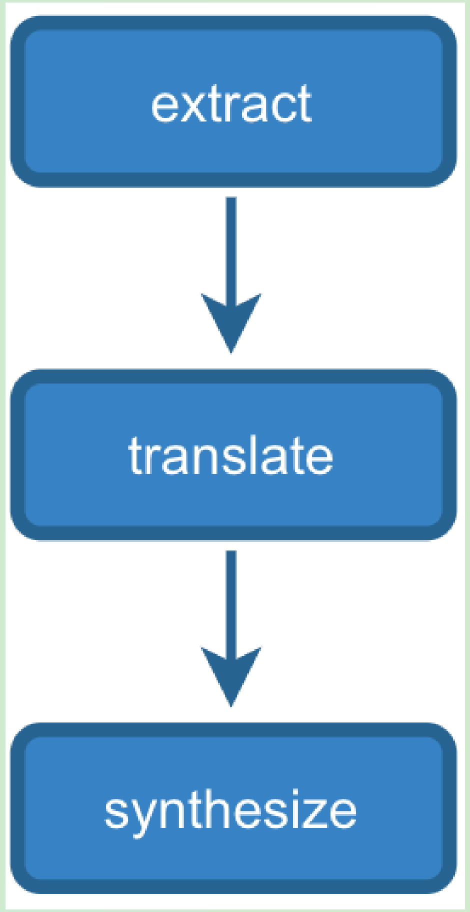


	Limitation	Requirement	Our approach
1	1 feature	*n* features	*composite BaaS*
2	problem size	scale problem	*split-merge*
3	throughput & concurr.	scale requests	*federation*

Real Life Use Cases Require **Composition**

- **Blind people**
  - **Listen** instead of reading
  - OCR → T2S
- In another language
  - **OCR → translate → T2S**

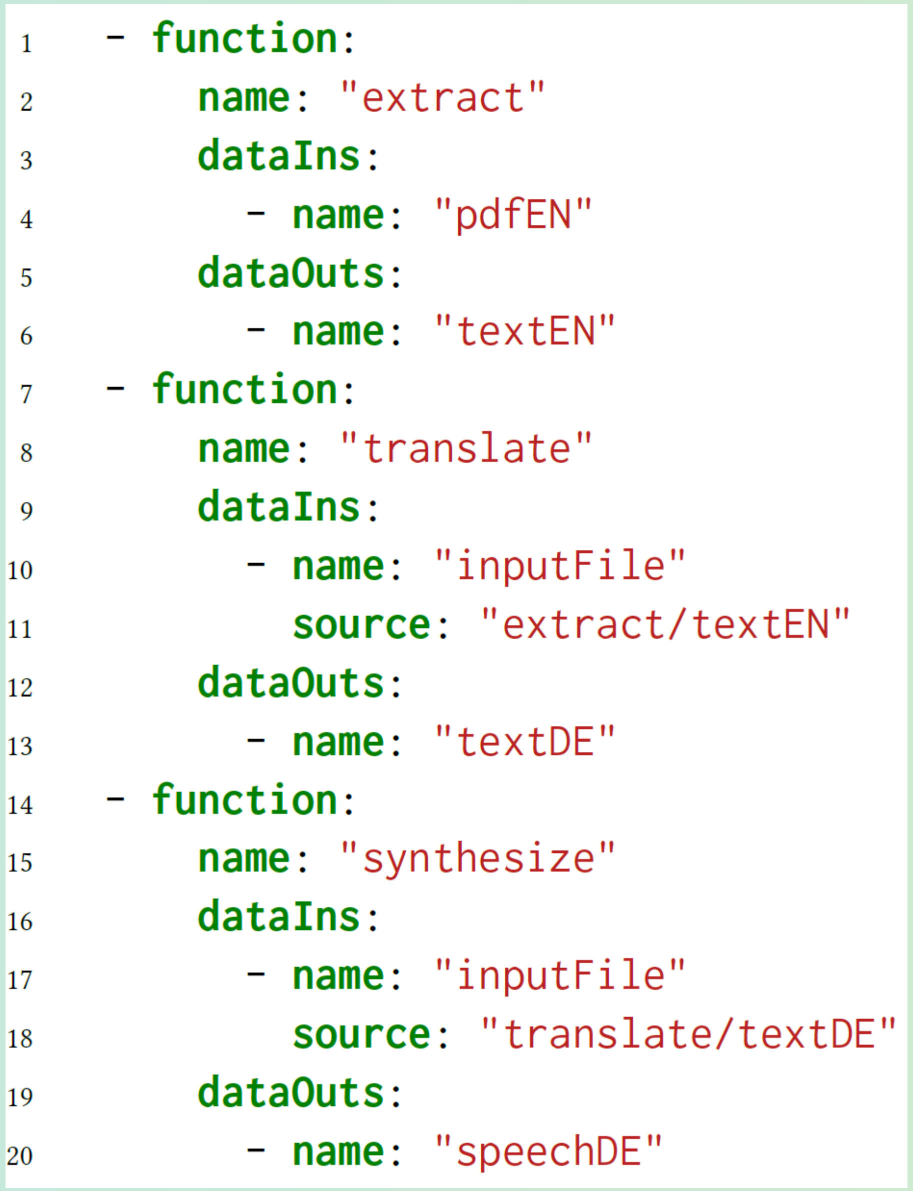




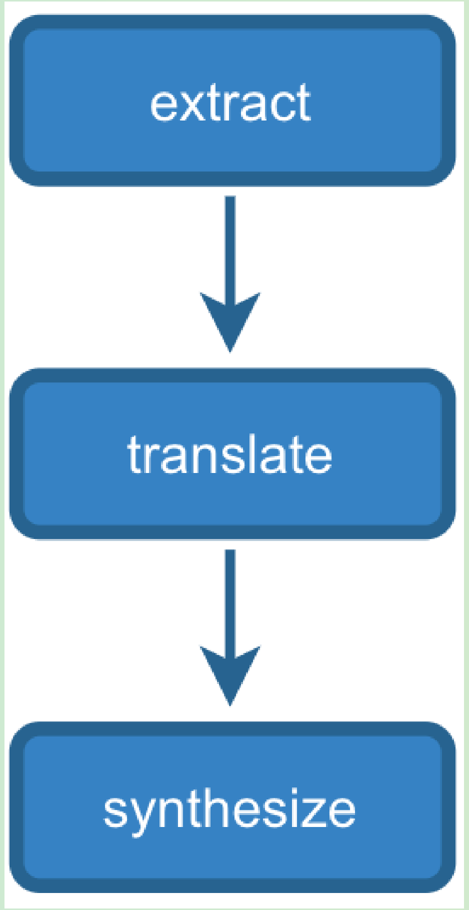
Compose With AFCL

- **OCR → translate → T2S**

- Wrap BaaS into FaaS
- A **sequence** of three functions
    - Extract
    - Translate
    - Synthesize
- Abstract Function Choreograpy Language
    - Ristov *et al*., “AFCL: An Abstract Function Choreography Language for serverless workflow specification”, FGCS, 2021, https://doi.org/10.1016/j.future.2020.08.012
- Run with **xAFCL**
    - S. Ristov *et al*., "xAFCL: Run Scalable Function Choreographies Across Multiple FaaS Systems," *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 711-723, 1 Jan.-Feb. 2023, doi: 10.1109/TSC.2021.3128137.



```
- function:
    name: "extract"
    dataIns:
      - name: "pdfEN"
    dataOuts:
      - name: "textEN"
- function:
    name: "translate"
    dataIns:
      - name: "inputFile"
        source: "extract/textEN"
    dataOuts:
      - name: "textDE"
- function:
    name: "synthesize"
    dataIns:
      - name: "inputFile"
        source: "translate/textDE"
    dataOuts:
      - name: "speechDE"
```

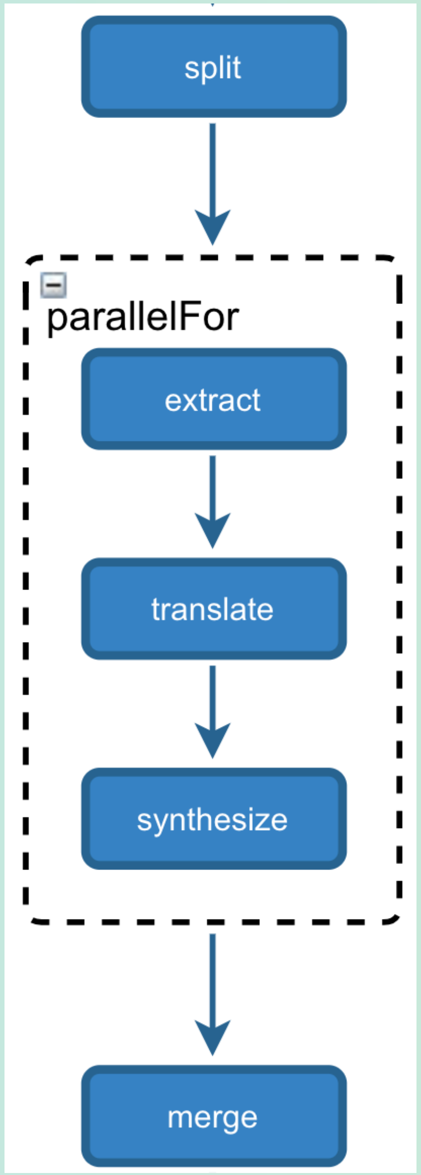


Two Composite BaaS Services

OCR → translate → T2S

AFCL-pdf2SpeechDE

- **Split** pdf into pages
- **Merge** the audio files

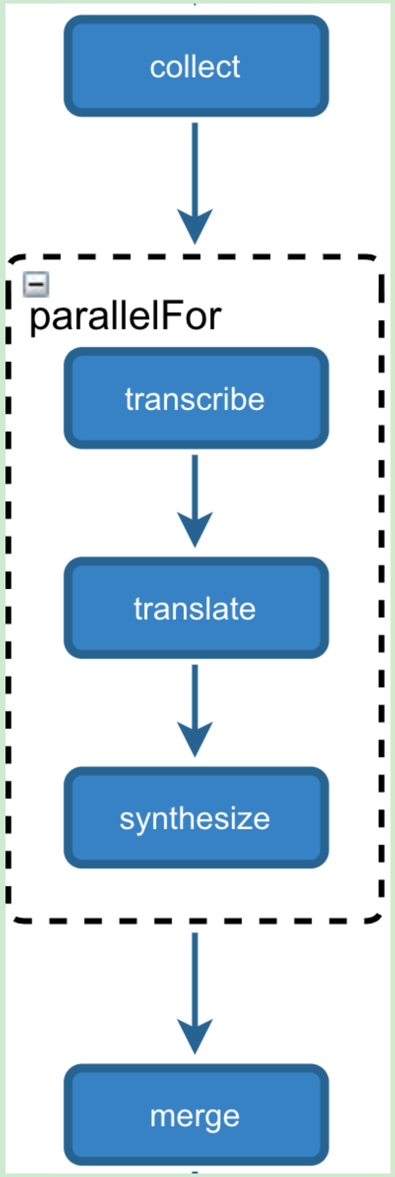


S2T → translate → T2S

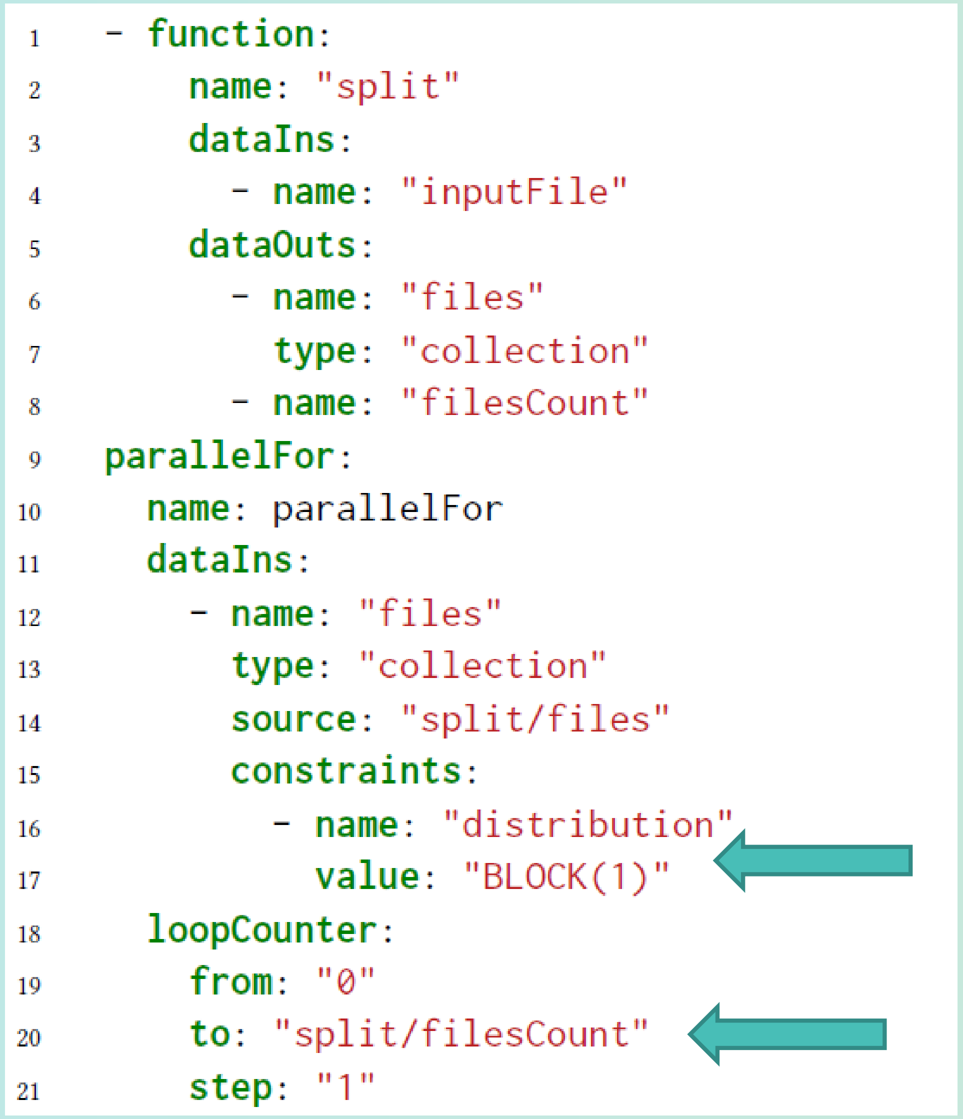
AFCL-Speech2SpeechDE

Collect file URLs

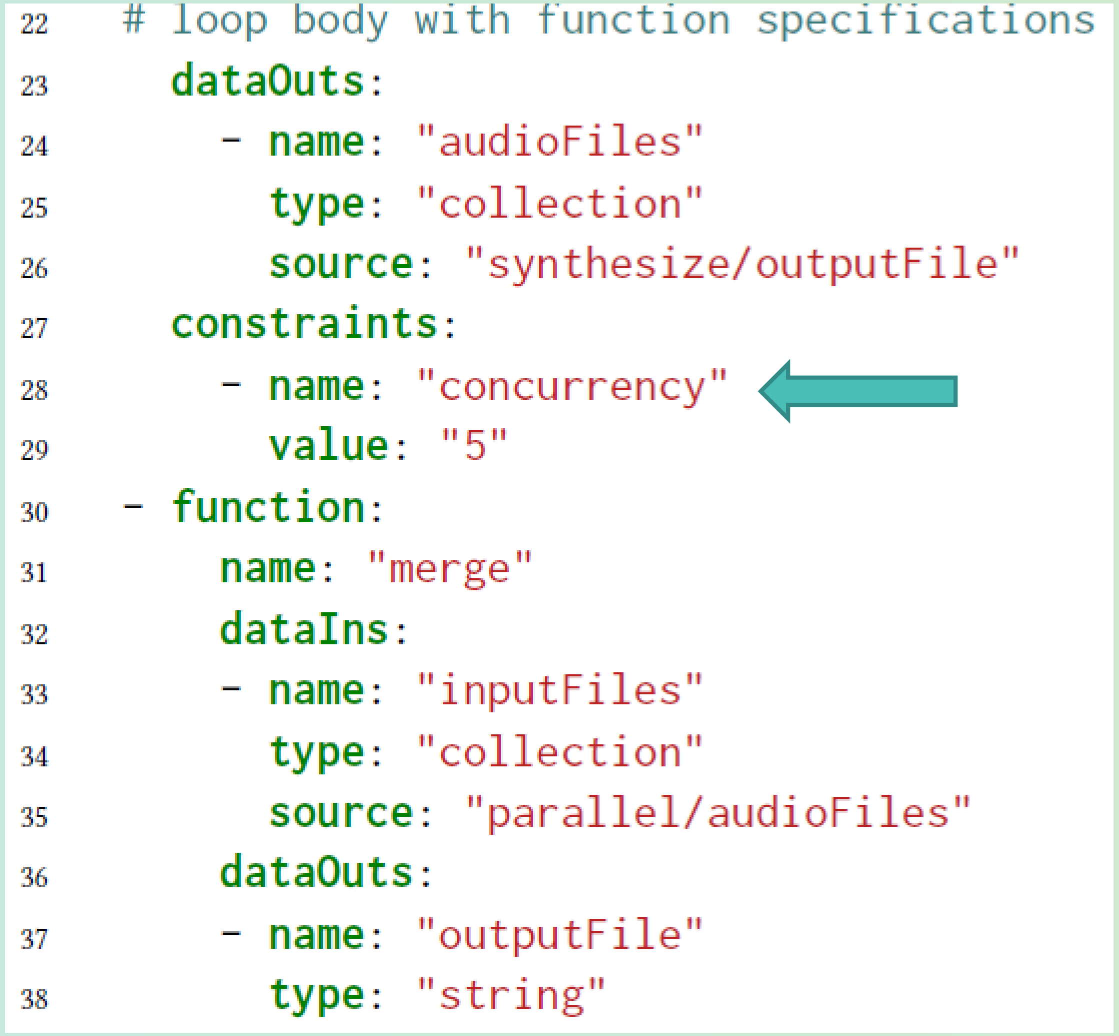
Merge the audio files



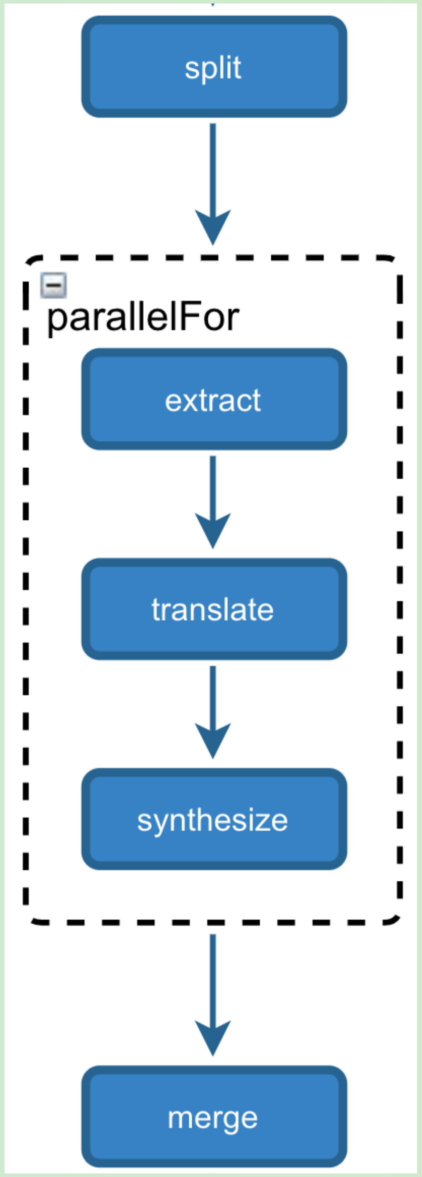
Two Composite BaaS Services



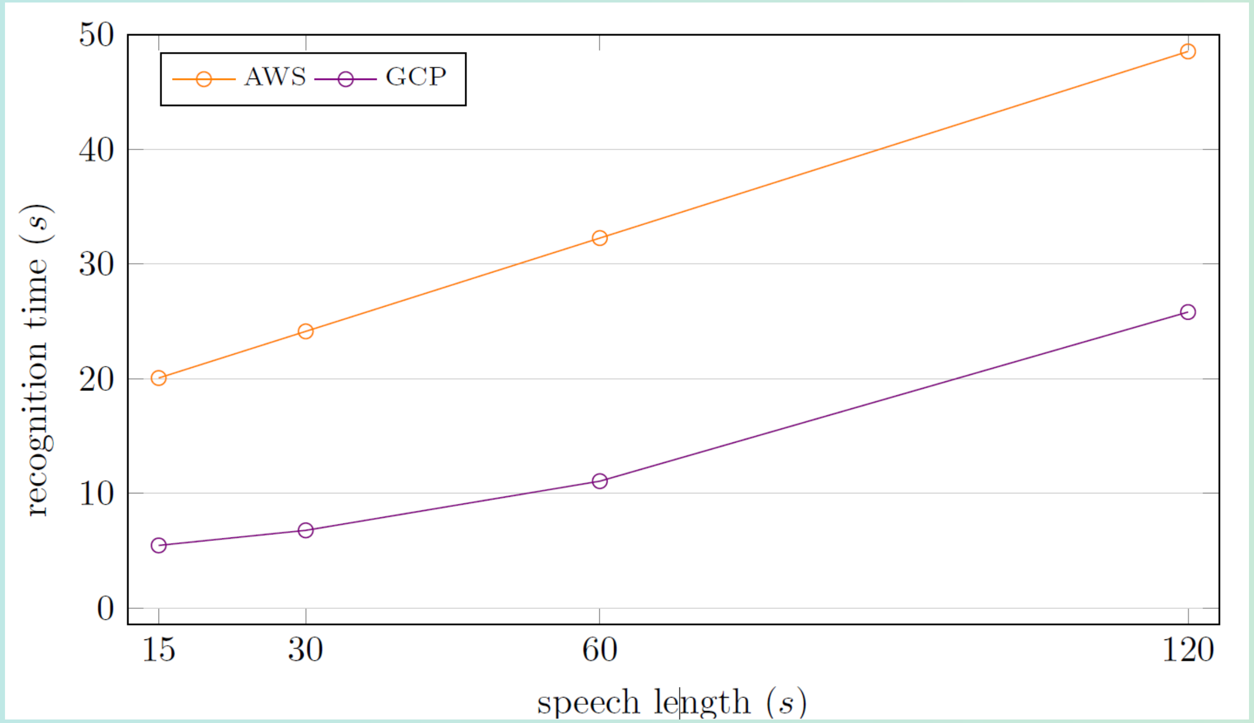
```
- function:
  name: "split"
  dataIns:
   - name: "inputFile"
  dataOuts:
   - name: "files"
    type: "collection"
   - name: "filesCount"
parallelFor:
 name: parallelFor
 dataIns:
  - name: "files"
   type: "collection"
   source: "split/files"
   constraints:
    - name: "distribution"
     value: "BLOCK(1)"
 loopCounter:
  from: "0"
  to: "split/filesCount"
  step: "1"
```

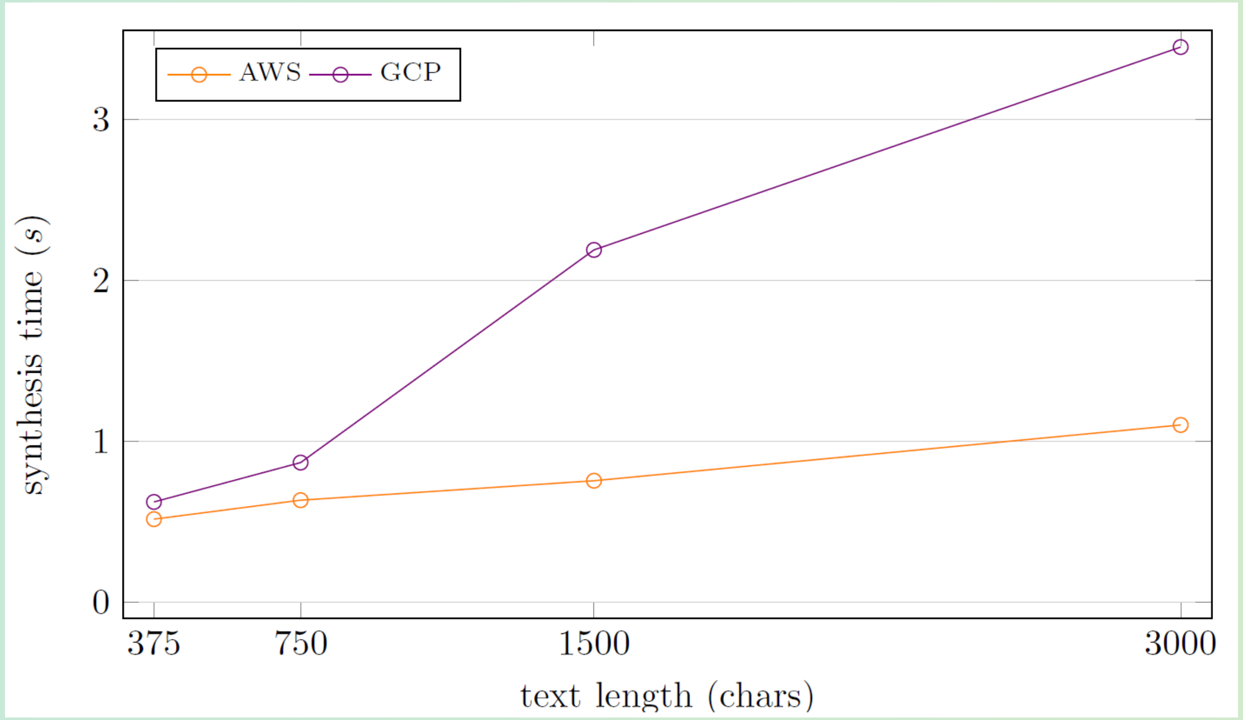


```
# loop body with function specifications
 dataOuts:
  - name: "audioFiles"
   type: "collection"
    source: "synthesize/outputFile"
 constraints:
  - name: "concurrency"
   value: "5"
- function:
  name: "merge"
  dataIns:
  - name: "inputFiles"
   type: "collection"
    source: "parallel/audioFiles"
  dataOuts:
  - name: "outputFile"
   type: "string"
```



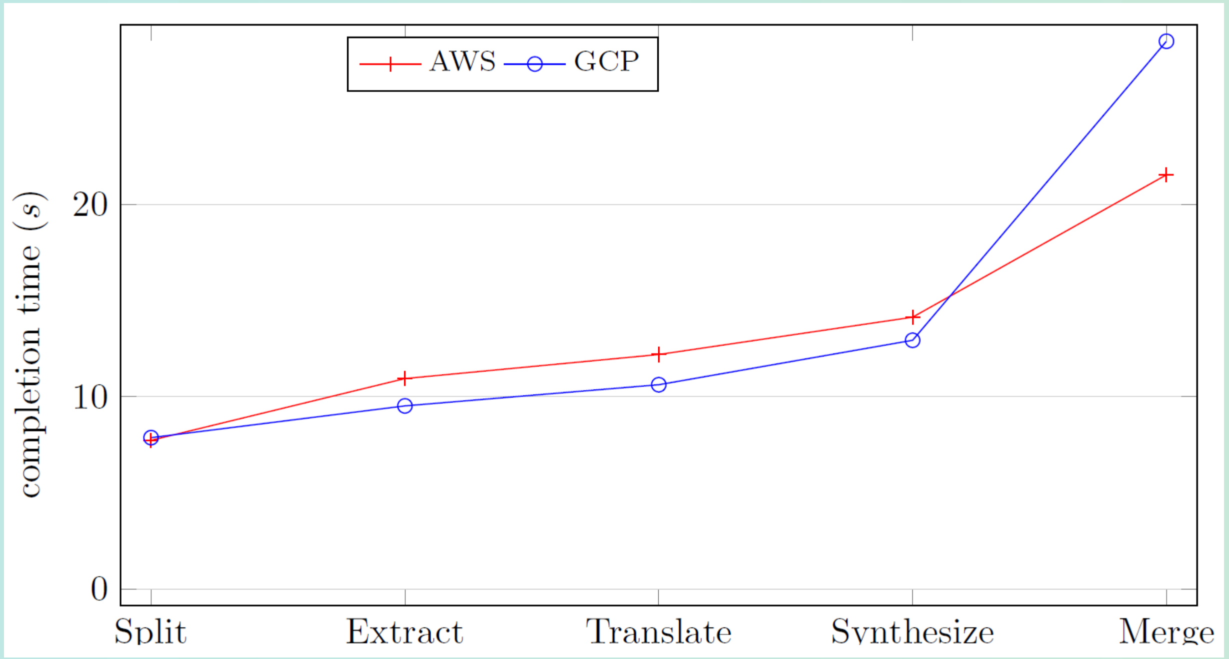
Individual BaaS Service Execution Time

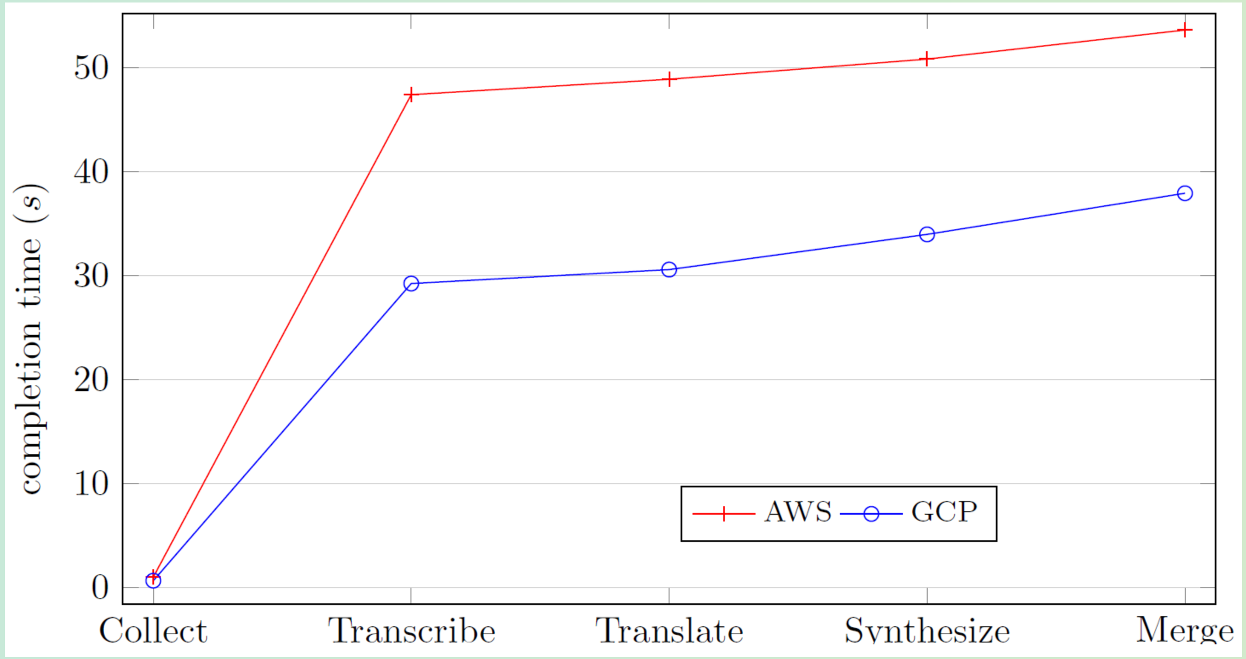




- Speech to text Text to speech

AFCL Workflows - Characteristics





pdf2SpeechDE

Function	*Instances*	*Download*		*Upload*		*BaaS*
		files	*data* [MB]	*files*	*data* [MB]	
Split	1	1	0.645	25	0.65	
Extract	25	1	0.026	1	∼ 0	✓
Translate	25	1	∼ 0	1	∼ 0	✓
Synthesize	25	1	∼ 0	1	2.5	✓
Merge	1	25	62.5	1	62.3	
Total	77	29	63.17	29	65.45	

speech2SpeechDE

Function	*Instances*	*Download*		*Upload*		*BaaS*
		files	*data* [MB]	*files*	*data* [MB]	
Collect	1	0	0	0	0	
transcribe	5	1	3.8	1	∼ 0	✓
Translate	5	1	∼ 0	1	∼ 0	✓
Synthesize	5	1	∼ 0	1	2.8	✓
Merge	1	5	14	1	14	
Total	17	8	6.6	4	16.8	

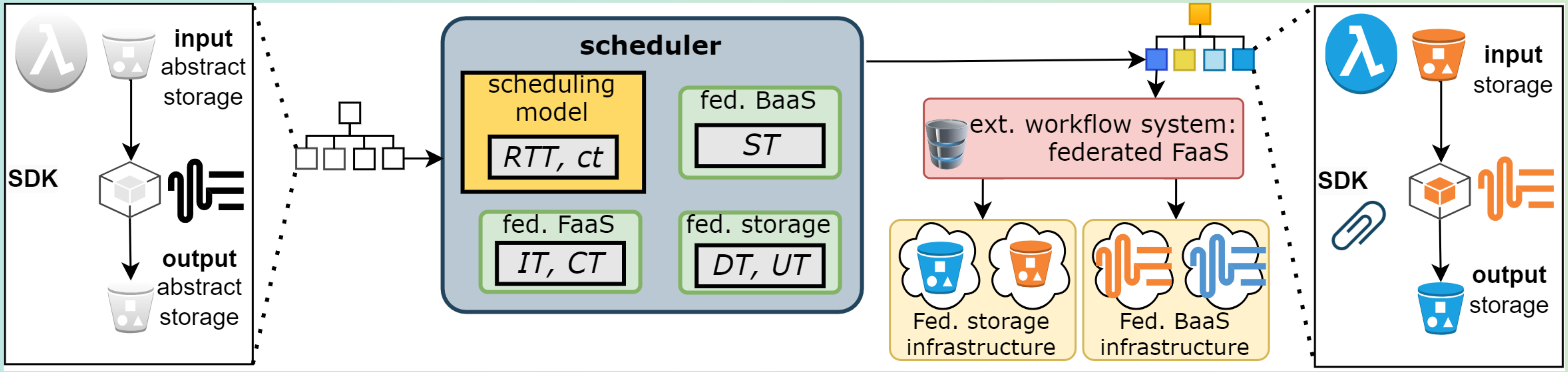
Conclusion

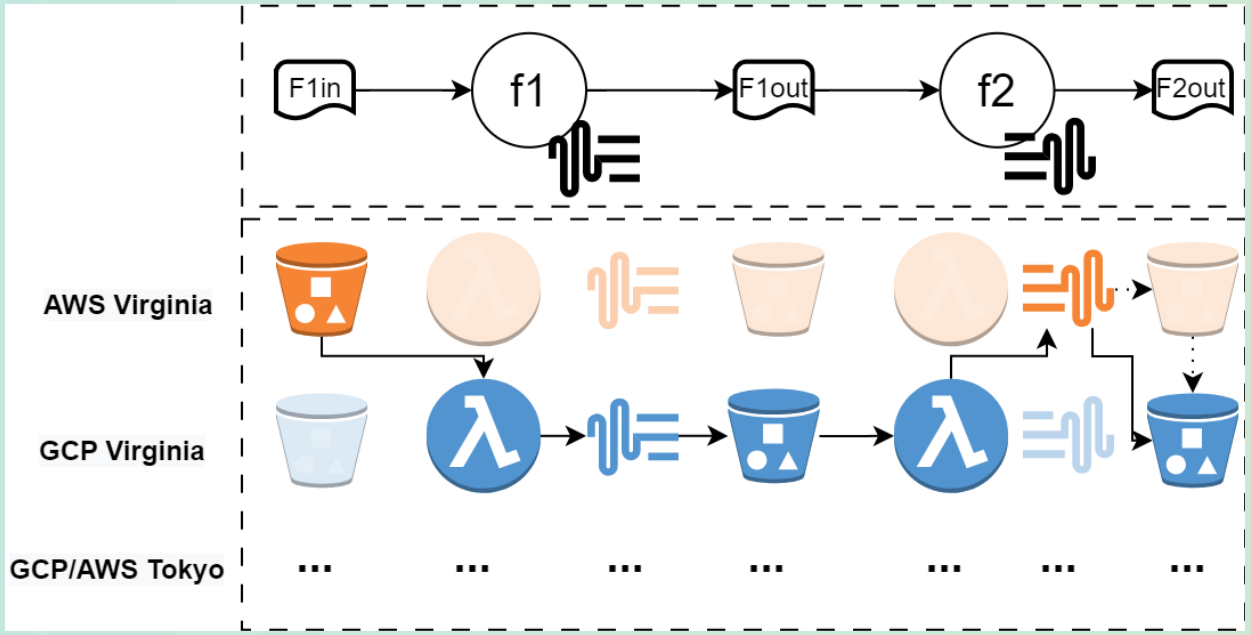
- Two scalable and composite BaaS services (no single BaaS service exists)
  - **AFCL-pdf2SpeechDE**
    - allows the blind people that understand German to "read by listening" the pdf files written in English
  - **AFCL-speech2SpeechDE**
    - Translates audio files from one natural language to another
- **Overcome three limitations**
  - Change *multiple features* of BaaS services (language and data format)
  - (un)limited *problem size* with parallelFor
  - (un)limited number of *concurrent* requests and *partially* the *throughput*

AFCL Workflows

- Workflows are publicly available (https://github.com/AFCLWorkflows)
  - Terraform deployment scripts including functions for AWS and GCP
  - AFCL files and input files
- Users can ***dynamically*** select the providers (AWS or GCP)
  - Function
  - Storage (https://github.com/FaaSTools/)
    - Go, Python, and Java.
  - BaaS services (https://github.com/FaaSTools/)
    - S2T, T2S, Translate, OCR, Face recognition
- Also other AFCL workflows
  - Montage, BWA, Genome, Monte Carlo, Celebrity Detection, Stock Prediction …

Future (Current) Work


input abstract storage
SDK
output abstract storage
scheduler
scheduling model
RTT, ct
fed. BaaS
ST
fed. FaaS
IT, CT
fed. storage
DT, UT
ext. workflow system: federated FaaS
Fed. storage infrastructure
Fed. BaaS infrastructure
input storage
SDK
output storage


F1in
f1
F1out
f2
F2out
AWS Virginia
GCP Virginia
GCP/AWS Tokyo
...