Writing an API to expose your tools / services

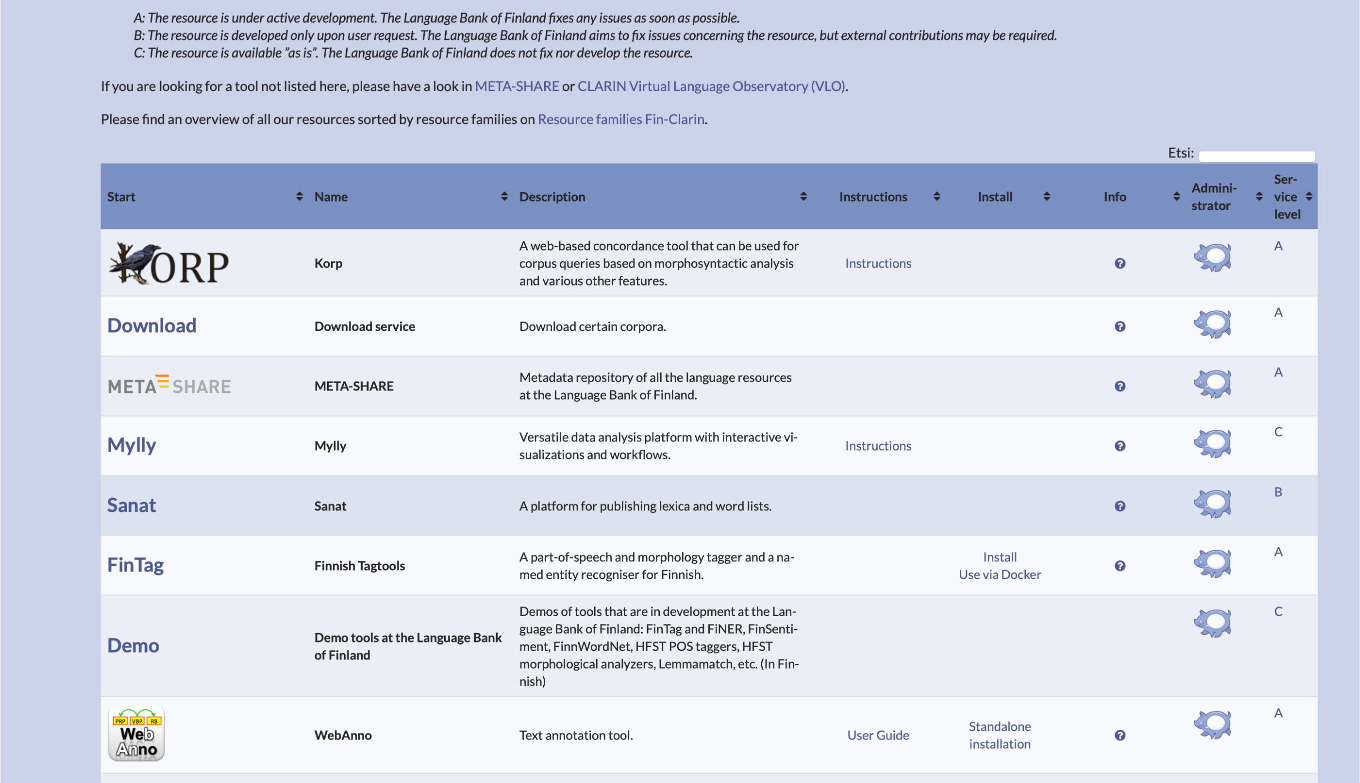
Case: Kielipankki / The Language Bank of Finland @ CSC

User Support Coffee

Sam Hardwick 30.11.2022

History of tools at Kielipankki

Our most obvious tools are those used to browse our corpora (eg. korp.csc.fi), but we make, host and curate a bunch of other tools as well (kielipankki.fi/tools)



A: The resource is under active development. The Language Bank of Finland fixes any issues as soon as possible.
B: The resource is developed only upon user request. The Language Bank of Finland aims to fix issues concerning the resource, but external contributions may be required.
C: The resource is available "as is". The Language Bank of Finland does not fix nor develop the resource.

If you are looking for a tool not listed here, please have a look in META-SHARE or CLARIN Virtual Language Observatory (VLO).

Please find an overview of all our resources sorted by resource families on Resource families Fin-Clarin.

Etsi:

Start	Name	Description	Instructions	Install	Info	Administrator	Service level
KORP	Korp	A web-based concordance tool that can be used for corpus queries based on morphosyntactic analysis and various other features.	Instructions		?		A
Download	Download service	Download certain corpora.			?		A
META SHARE	META-SHARE	Metadata repository of all the language resources at the Language Bank of Finland.			?		A
Mylly	Mylly	Versatile data analysis platform with interactive visualizations and workflows.	Instructions		?		C
Sanat	Sanat	A platform for publishing lexica and word lists.			?		B
FinTag	Finnish Tagtools	A part-of-speech and morphology tagger and a named entity recogniser for Finnish.		Install Use via Docker	?		A
Demo	Demo tools at the Language Bank of Finland	Demos of tools that are in development at the Language Bank of Finland: FinTag and FiNER, FinSentiment, FinnWordNet, HFST POS taggers, HFST morphological analyzers, Lemmamatch, etc. (In Finnish)					C
WebAnno	WebAnno	Text annotation tool.	User Guide	Standalone installation	?		A

History of tools at Kielipankki

Kielipankki ingests a lot of data:

- Newspaper & book collections, with existing metadata
- Internet data
- Speech

This needs a lot of processing, annotating and enriching for which we have internal tools

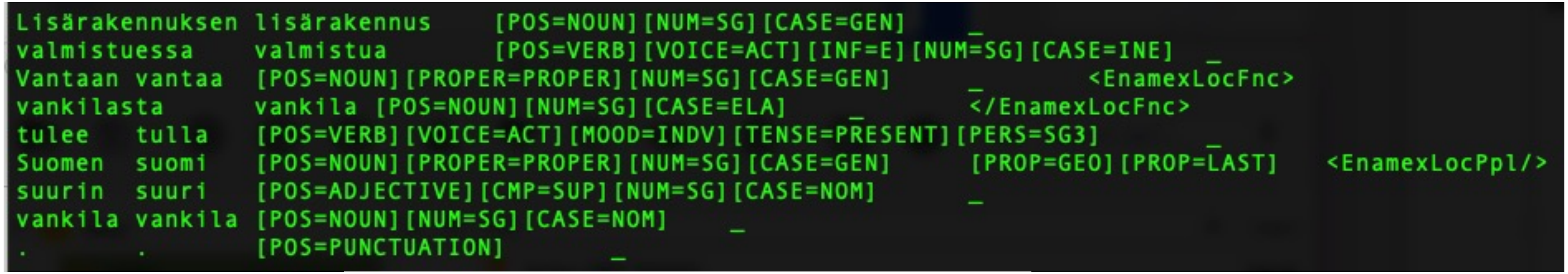
History of tools at Kielipankki

What kind of processing?

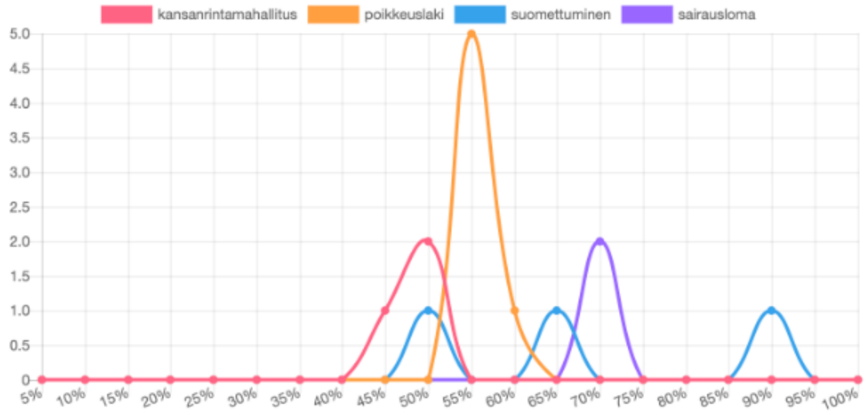
- linguistic analysis: lemmatization, morphology, syntax, ...
- named entities: persons, places, organizations, events, ...
- sentiment: positive, neutral, negative
- classification: topic, genre, ...
- automatic speech recognition

Getting to the users

Many of our tools produce intermediate results, which are not interesting in themselves, but may be used to make other tools.



```
Lisärakennuksen lisärakennus    [POS=NOUN][NUM=SG][CASE=GEN]      _
valmistuessa    valmistua       [POS=VERB][VOICE=ACT][INF=E][NUM=SG][CASE=INE]  _
Vantaan vantaa  [POS=NOUN][PROPER=PROPER][NUM=SG][CASE=GEN]       _         <EnamexLocFnc>
vankilasta      vankila [POS=NOUN][NUM=SG][CASE=ELA]    _         </EnamexLocFnc>
tulee   tulla   [POS=VERB][VOICE=ACT][MOOD=INDV][TENSE=PRESENT][PERS=SG3]       _
Suomen  suomi   [POS=NOUN][PROPER=PROPER][NUM=SG][CASE=GEN]       [PROP=GEO][PROP=LAST]   <EnamexLocPpl/>
suurin  suuri   [POS=ADJECTIVE][CMP=SUP][NUM=SG][CASE=NOM]        _
vankila vankila [POS=NOUN][NUM=SG][CASE=NOM]    _
.       .       [POS=PUNCTUATION]       _
```



Getting to the users

Some tasks (ASR) are highly in demand but our service was hard to use (log in to Puhti). How do we encourage integration (or even use)?

Endpoints

Idea: we could have API endpoints for different outputs:

kielipankki.rahtiapp.fi/text/fi/{postag, nertag, sentiment}

kielipankki.rahtiapp.fi/audio/asr/fi/submit_file

…

No end-user installation, updates and scalability are up to the service.

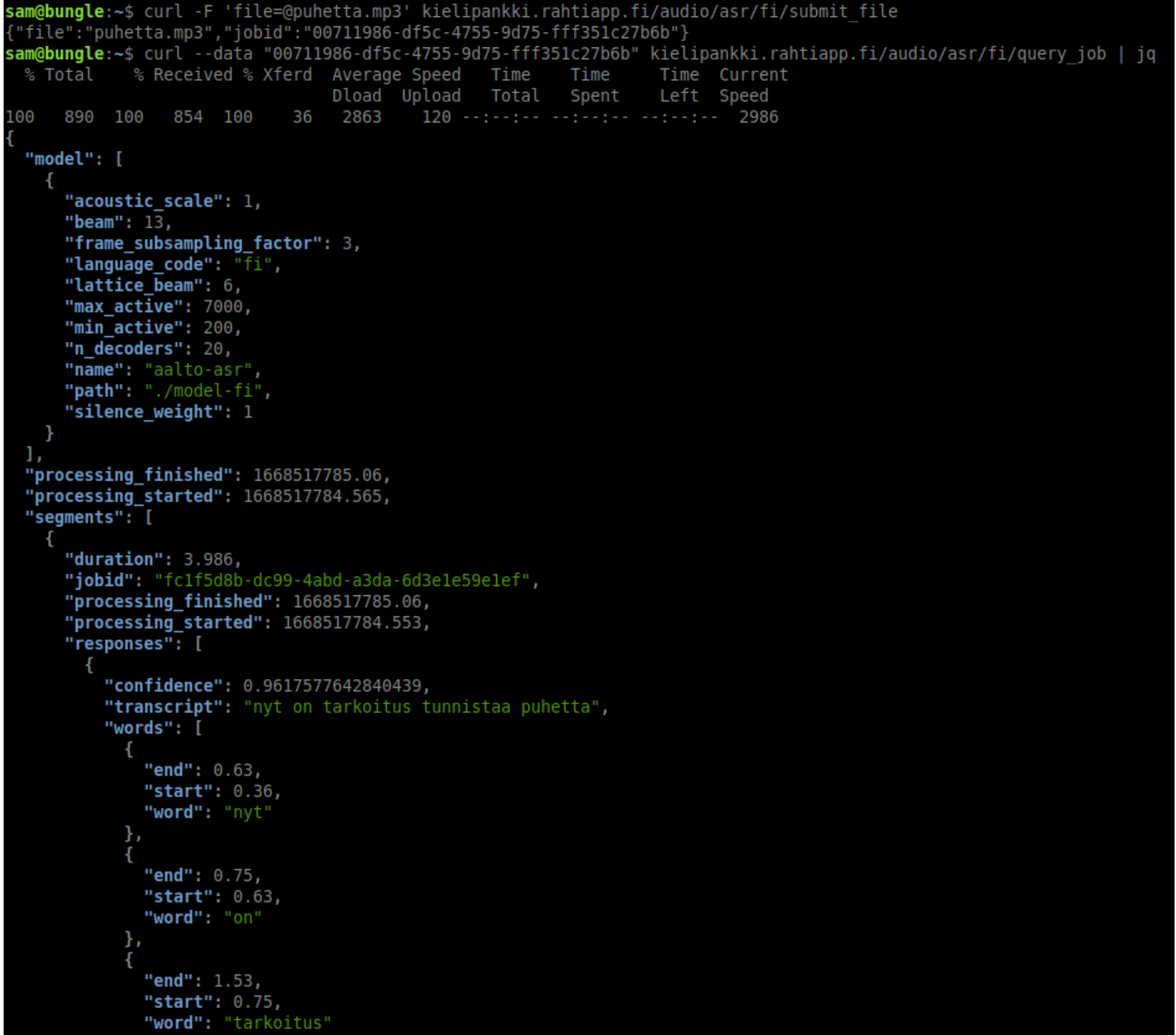
A file is submitted

You get a UUID and poll for results

We can verbosely include model data in each response to support data versioning end references

How long did it take? We also have a load / queue endpoint

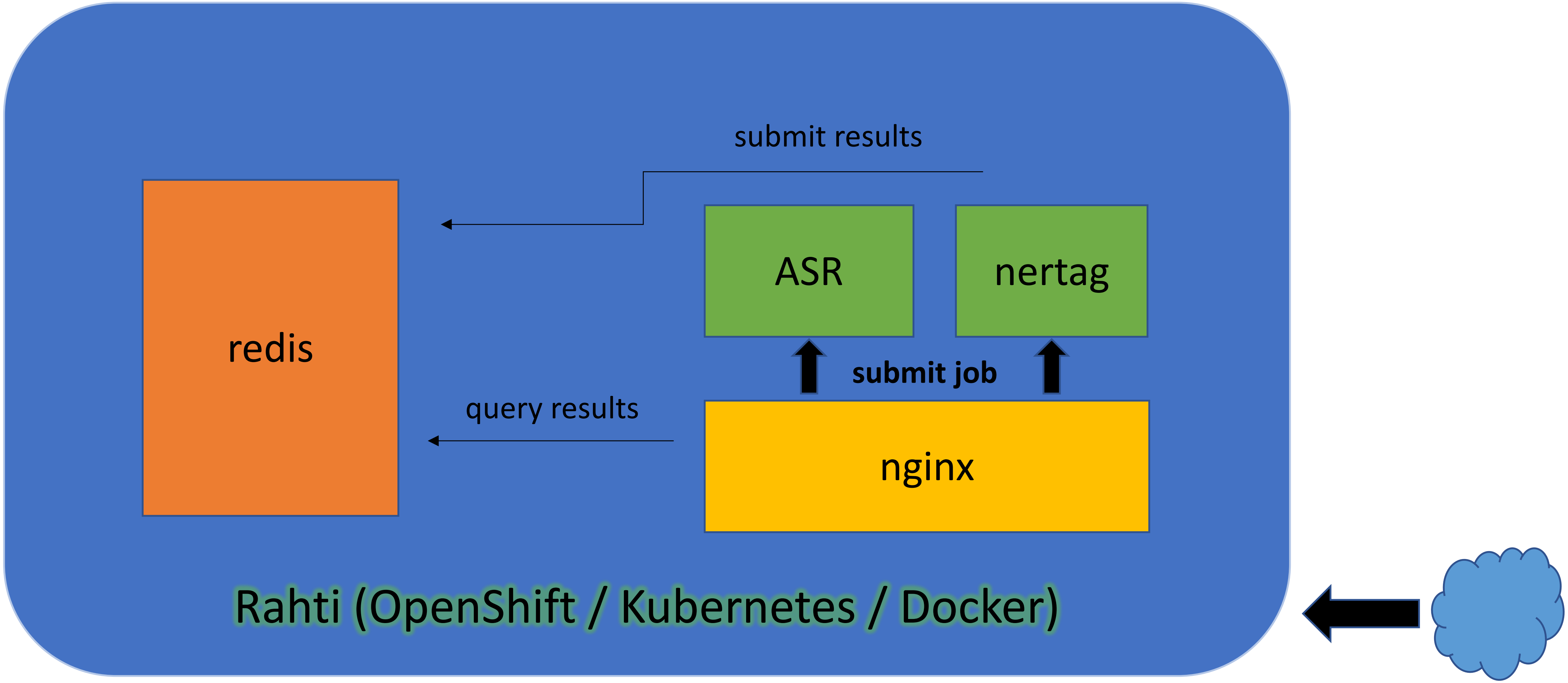
Confidence score, possibly multiple responses, word alignment, diarization & punctuation forthcoming



```
sam@bungle:~$ curl -F 'file=@puhetta.mp3' kielipankki.rahtiapp.fi/audio/asr/fi/submit_file
{"file":"puhetta.mp3","jobid":"00711986-df5c-4755-9d75-fff351c27b6b"}
sam@bungle:~$ curl --data "00711986-df5c-4755-9d75-fff351c27b6b" kielipankki.rahtiapp.fi/audio/asr/fi/query_job | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   890  100   854  100    36   2863    120 --:--:-- --:--:-- --:--:--  2986
{
  "model": [
    {
      "acoustic_scale": 1,
      "beam": 13,
      "frame_subsampling_factor": 3,
      "language_code": "fi",
      "lattice_beam": 6,
      "max_active": 7000,
      "min_active": 200,
      "n_decoders": 20,
      "name": "aalto-asr",
      "path": "./model-fi",
      "silence_weight": 1
    }
  ],
  "processing_finished": 1668517785.06,
  "processing_started": 1668517784.565,
  "segments": [
    {
      "duration": 3.986,
      "jobid": "fc1f5d8b-dc99-4abd-a3da-6d3e1e59e1ef",
      "processing_finished": 1668517785.06,
      "processing_started": 1668517784.553,
      "responses": [
        {
          "confidence": 0.9617577642840439,
          "transcript": "nyt on tarkoitus tunnistaa puhetta",
          "words": [
            {
              "end": 0.63,
              "start": 0.36,
              "word": "nyt"
            },
            {
              "end": 0.75,
              "start": 0.63,
              "word": "on"
            },
            {
              "end": 1.53,
              "start": 0.75,
              "word": "tarkoitus"
```

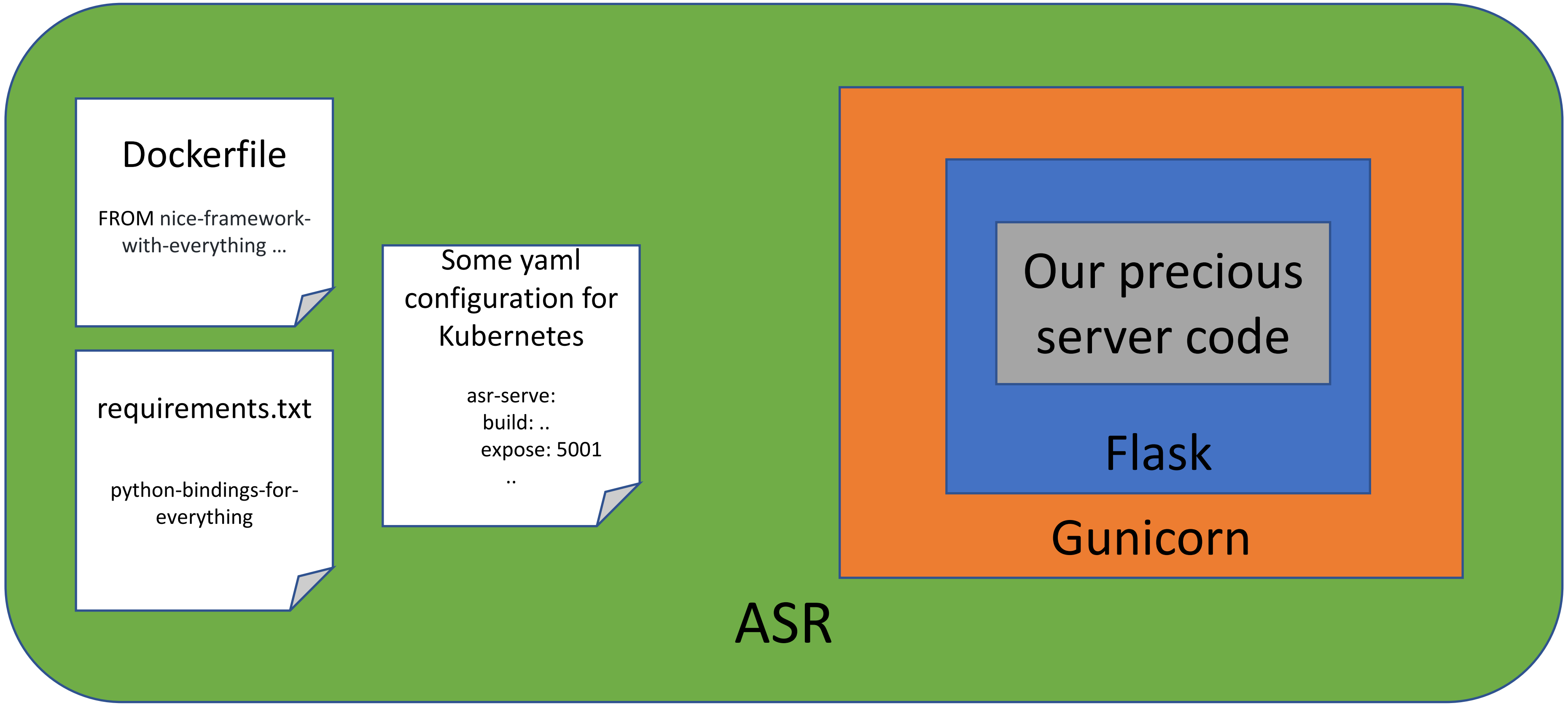
Endpoints

“That’s nice, but sounds hard”



Endpoints – you don’t need a lot of code

Inside the container:



Dockerfile

FROM nice-framework-with-everything …

requirements.txt

python-bindings-for-everything

Some yaml configuration for Kubernetes

```
asr-serve:
  build: ..
    expose: 5001
  ..
```

Our precious server code

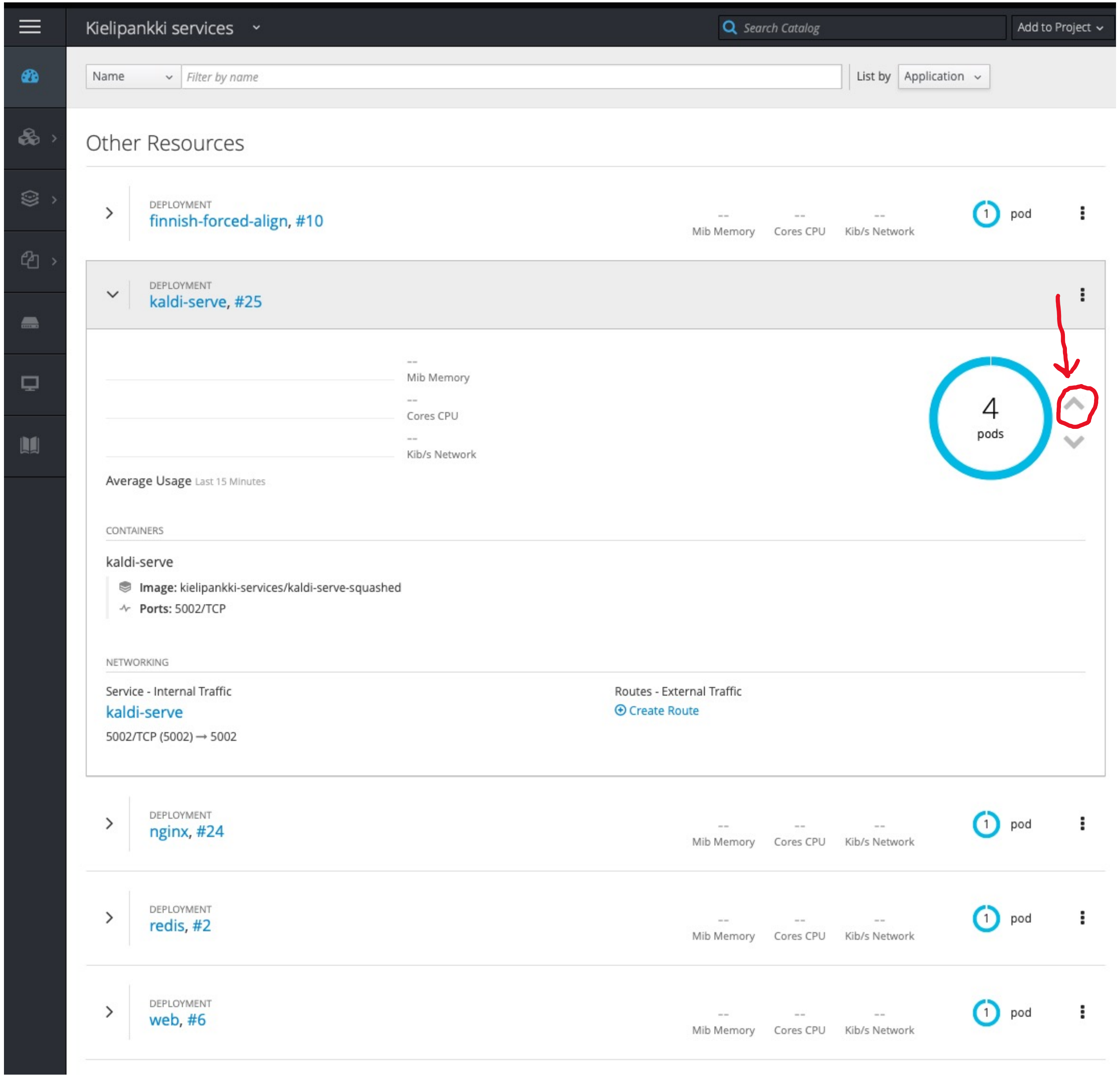
Flask

Gunicorn

ASR

Endpoints

Now scaling is easy, in theory:



Other Resources

DEPLOYMENT
finnish-forced-align, #10
-- Mib Memory -- Cores CPU -- Kib/s Network
1 pod

DEPLOYMENT
kaldi-serve, #25

-- Mib Memory
-- Cores CPU
-- Kib/s Network

Average Usage Last 15 Minutes

4 pods

CONTAINERS

kaldi-serve

Image: kielipankki-services/kaldi-serve-squashed
Ports: 5002/TCP

NETWORKING

Service - Internal Traffic
kaldi-serve
5002/TCP (5002) → 5002

Routes - External Traffic
Create Route

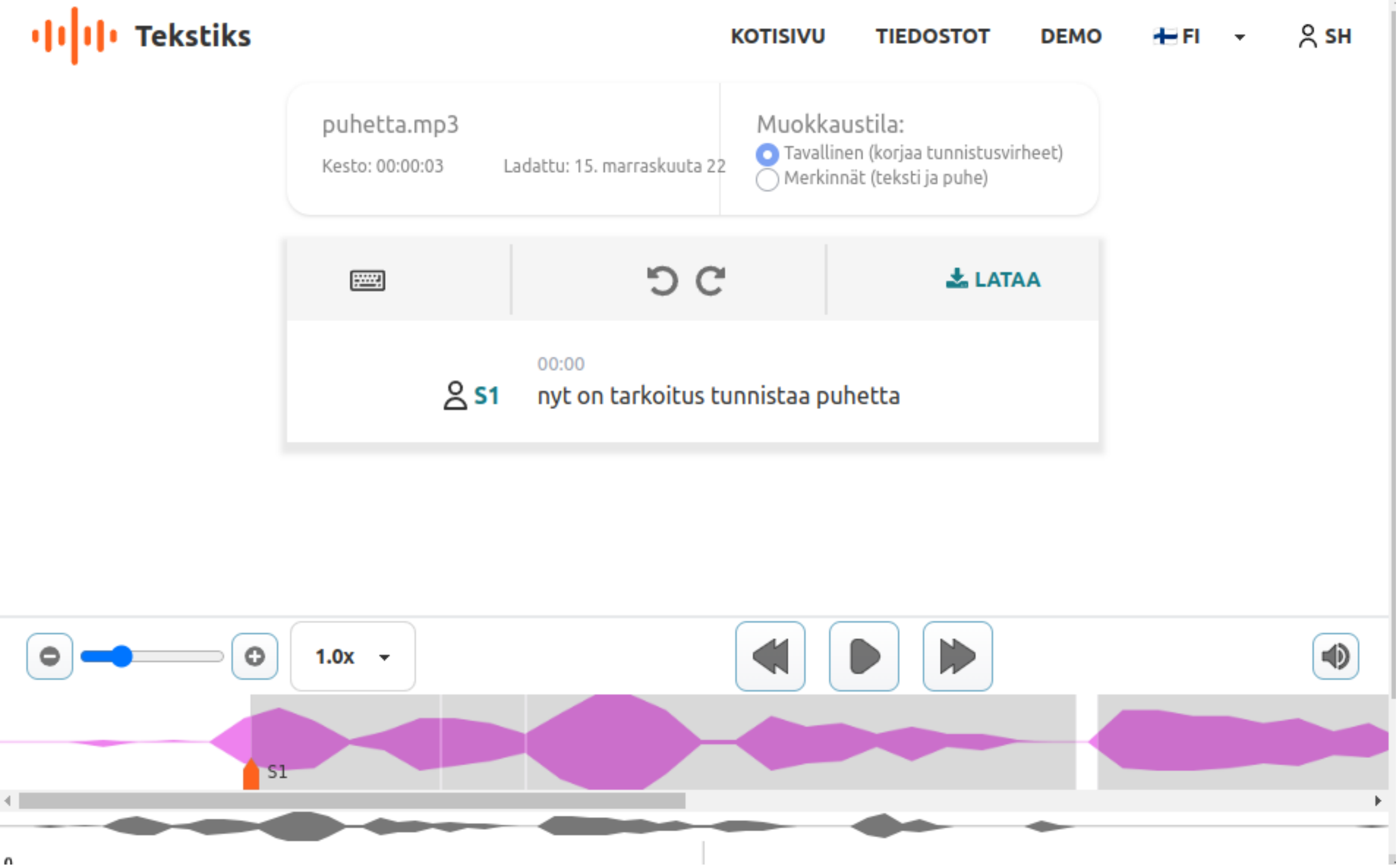
DEPLOYMENT
nginx, #24
-- Mib Memory -- Cores CPU -- Kib/s Network
1 pod

DEPLOYMENT
redis, #2
-- Mib Memory -- Cores CPU -- Kib/s Network
1 pod

DEPLOYMENT
web, #6
-- Mib Memory -- Cores CPU -- Kib/s Network
1 pod

Integration

Our demo site uses the ASR endpoint to do ASR, but we also got a very nice third-party integration with it



puhetta.mp3

Kesto: 00:00:03

Ladattu: 15. marraskuuta 22

Muokkaustila:

● Tavallinen (korjaa tunnistusvirheet)

○ Merkinnät (teksti ja puhe)

00:00

S1 nyt on tarkoitus tunnistaa puhetta