sovanta
simplicity first

Bringing NLP to Production

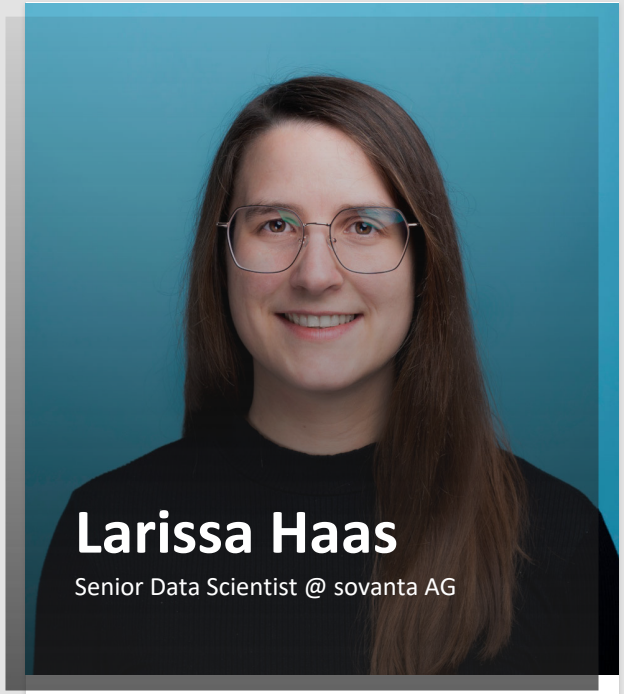
(an end to end story about some multi-language NLP services)

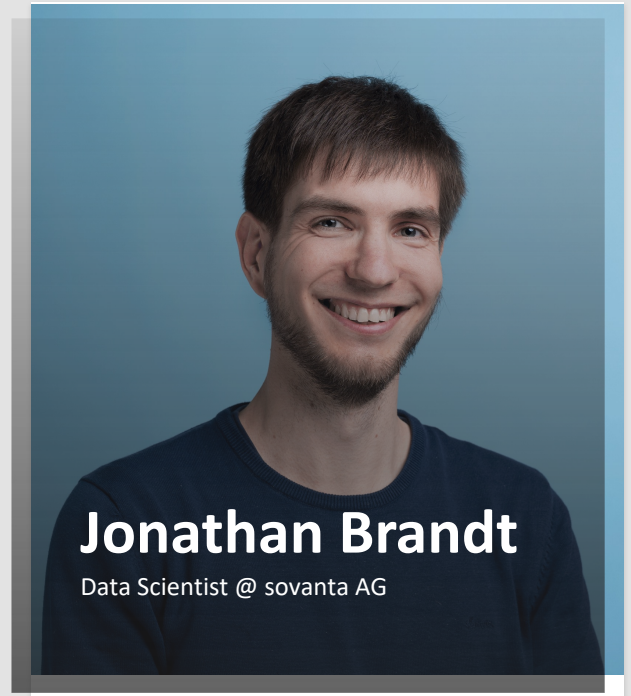
sovanta AG, 19.04.23

Larissa Haas, Jonathan Brandt

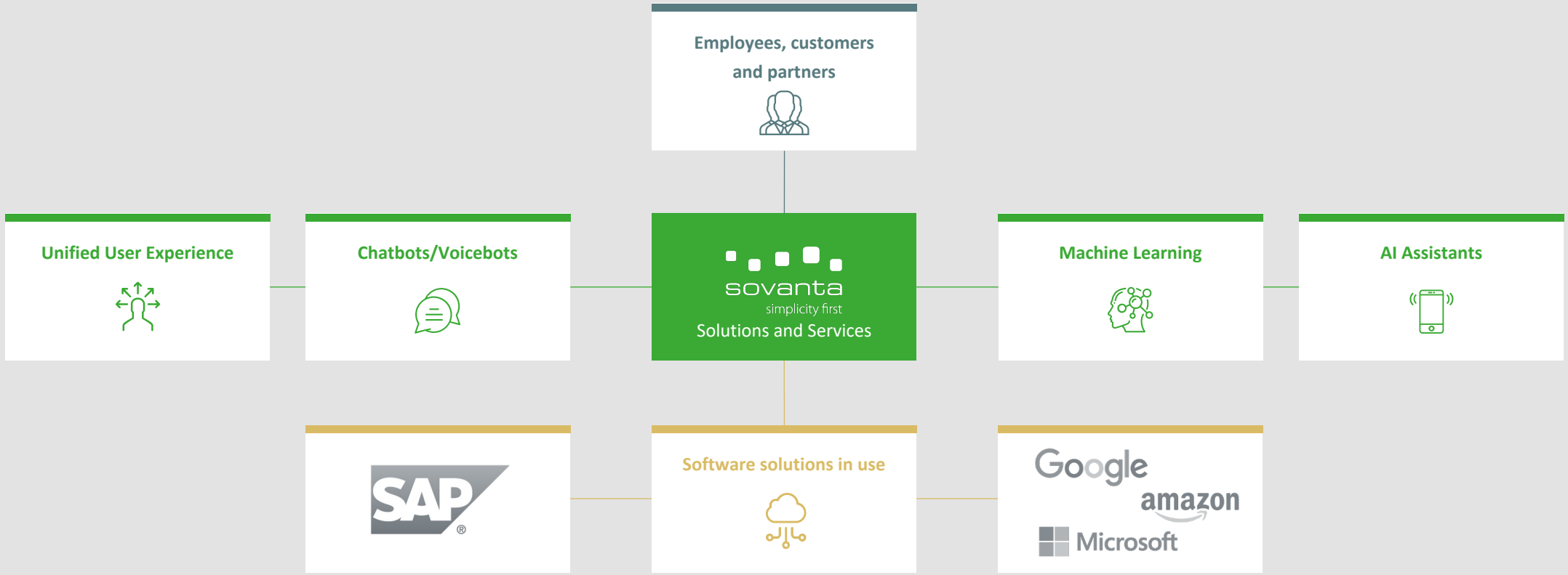
Agenda

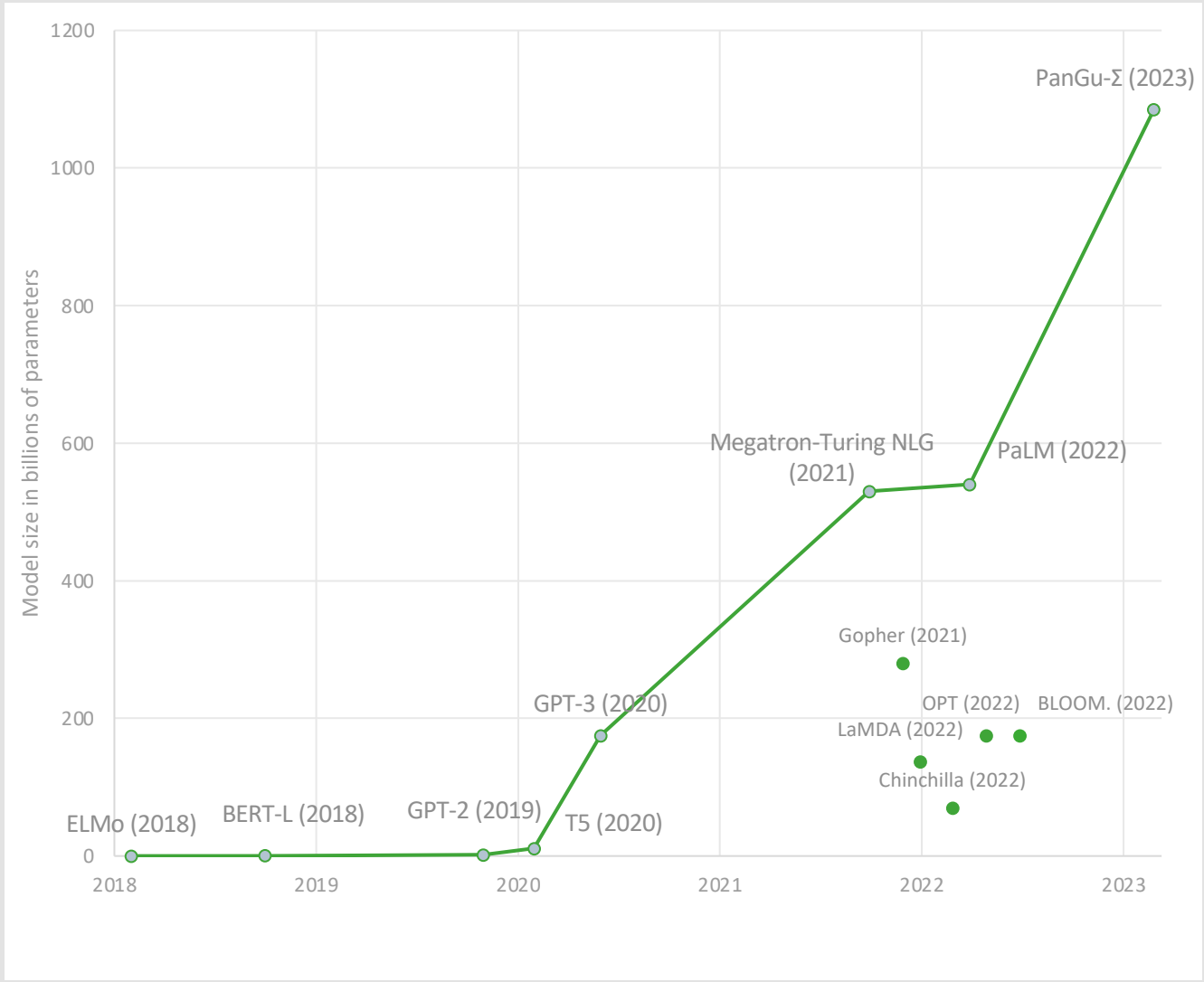
1. Journey to NLPOps
2. Use Case 1 - Productsearch
3. Use Case 2 - Text classification
4. Checklist to take away





User Experience closes the Business Gap



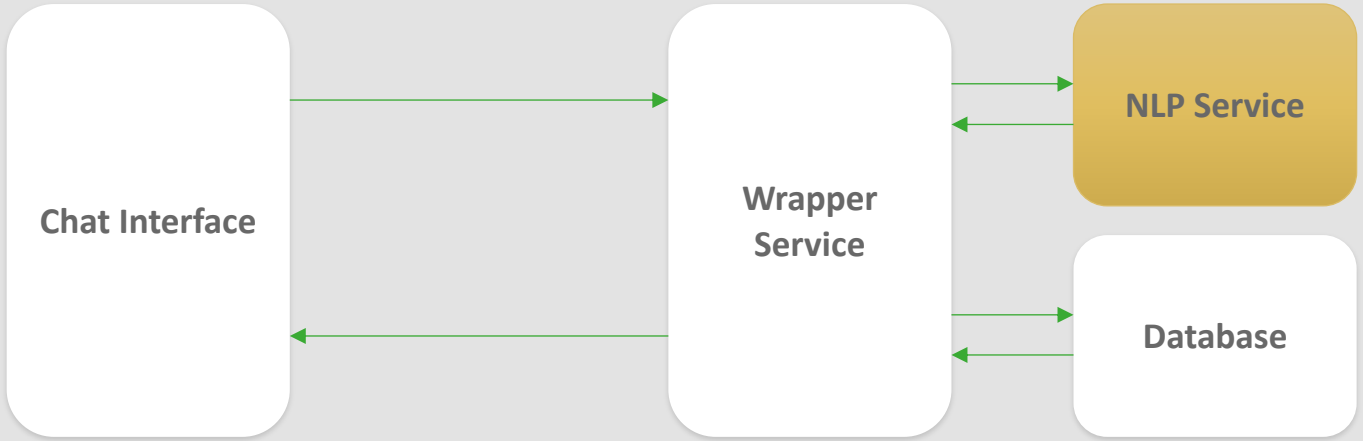

1200
1000
800
600
400
200
0
Model size in billions of parameters
2018
2019
2020
2021
2022
2023
ELMo (2018)
BERT-L (2018)
GPT-2 (2019)
T5 (2020)
GPT-3 (2020)
Megatron-Turing NLG (2021)
PaLM (2022)
PanGu-Σ (2023)
Gopher (2021)
OPT (2022)
BLOOM. (2022)
LaMDA (2022)
Chinchilla (2022)

Journey to NLPOps

Deployment

NLPOps Checklist
☑ Model Type
☑ Languages
☑ Size of Dependencies
☑ Retraining Frequency and Location
☑ Request Frequency and Availability
☑ Response Times
☑ Running Perspective
☑ Monitoring / Security
☑ Running Context / Existing Infrastructure

Use Case 1 – Productsearch



I want to buy vegan pizza

```
[
  {“text”: “vegan”, “ner”: “property”},
  {“text”: ”pizza”, “ner”: “product”}
]
```

- Pizza XYZ
- Pizza ABC

Here are your vegan pizza options: …

Use Case 1 – Productsearch

Feature	Characteristic
Model Type	SpaCy Model
Languages	One language
Size of Dependencies	ca. 300 MB
Retraining Frequency and Location	none
Request Frequency and Availability	
Response Times	
Running Perspective	PoC

Cloud Foundry (self-hosted)

Advantages:

- No need to care about infrastructure
- Complete service is pushed
- Quick (re)starts

Drawbacks:

- Dependent on buildpack: only certain python versions available
- 1 GB upload limitation (with all dependencies)
- 4 GB disk space & limited memory allocation

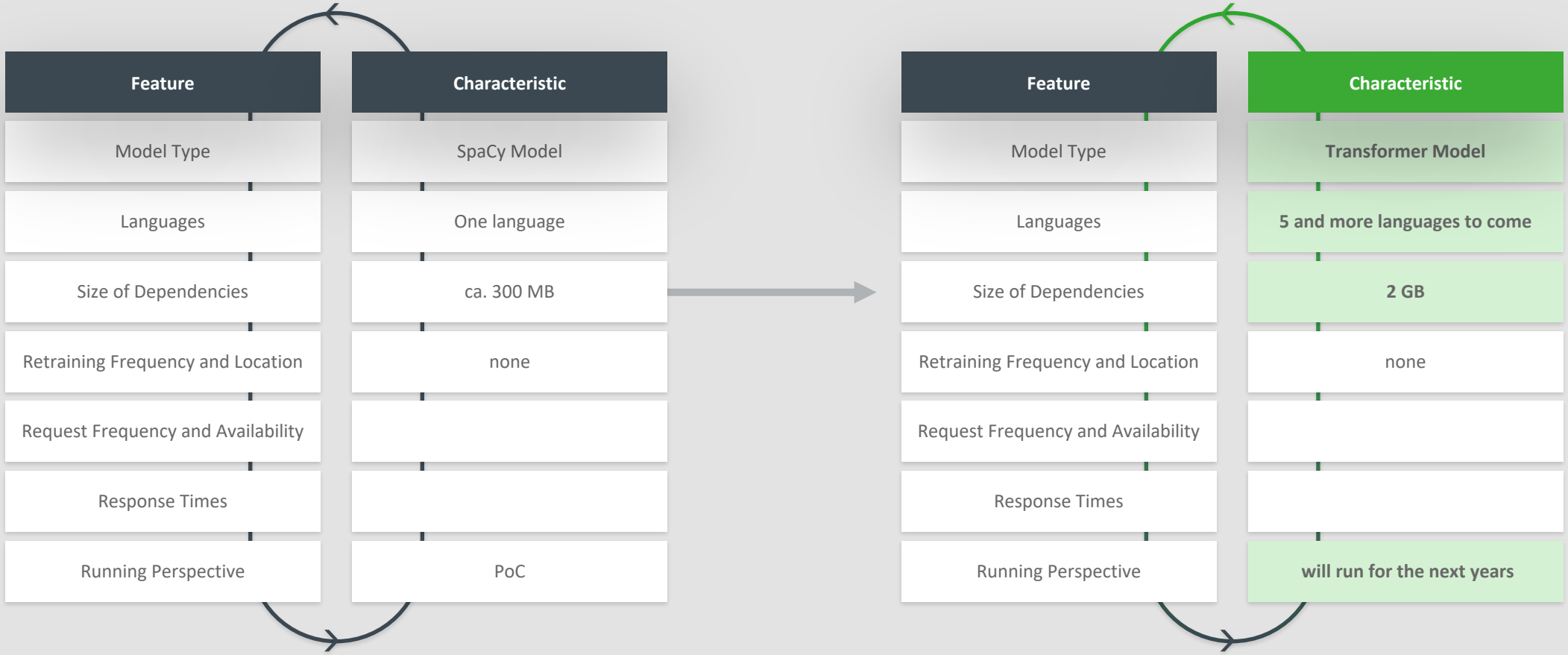
manifest-qx.dev.yaml

```yaml
applications:
  - name:          -connector-dev
    type: python
    path:          -connector
    parameters:
      memory: 250
      disk_quota: 250
      instance: 1
      health-check-type: process
      buildpacks:
        - https://github.com/cloudfoundry/python-buildpack
      buildpacks: [python_buildpack]
    provides:
      - name:          -connector-binding
        properties:
          url: ${default-url}
```

CLOUD FOUNDRY

Use Case 1 – Productsearch

Updating Requirements



Use Case 1 – Productsearch

Feature	Characteristic
Model Type	Transformer Model
Languages	10 languages
Size of Dependencies	ca. 2 GB
Retraining Frequency and Location	none
Request Frequency and Availability	
Response Times	
Running Perspective	will run for the next years

Azure Stack (Kubernetes / Container Instances)

Advantages:

- Quick, easy starting of containers without configuring clusters
- Known environment with Docker and deployment yamls

Drawbacks = Requirements:

- Docker or CI/CD pipeline
- Registry

Use Case 2 – Text classification

no error was detected with the device

… “text”: “no error was detected with the device”, “lang”: “EN” …

“pred”: 0 – not relevant

das Sensorkabel ist kaputt

… “text”: “das Sensorkabel ist kaputt”, “lang”: “DE” …

“pred”: 1 – relevant

Use Case 2 – Text classification

Feature	Characteristic
Model Type	"plain" NLP
Languages	English + German
Size of Dependencies	ca. 300 MB
Retraining Frequency and Location	none
Request Frequency and Availability	scheduled scans + rare manual requests
Response Times	
Running Perspective	PoC

Cloud Foundry (SAP Business Technology Platform)

Advantages:

- Hosting existing application
- Easy communication between app and model
- BTP offers additional services

Drawbacks:

- 1.5 GB upload limitation (with all dependencies)
- 4 GB disk space & 8 GB memory allocation
- CPU only

CLOUD FOUNDRY

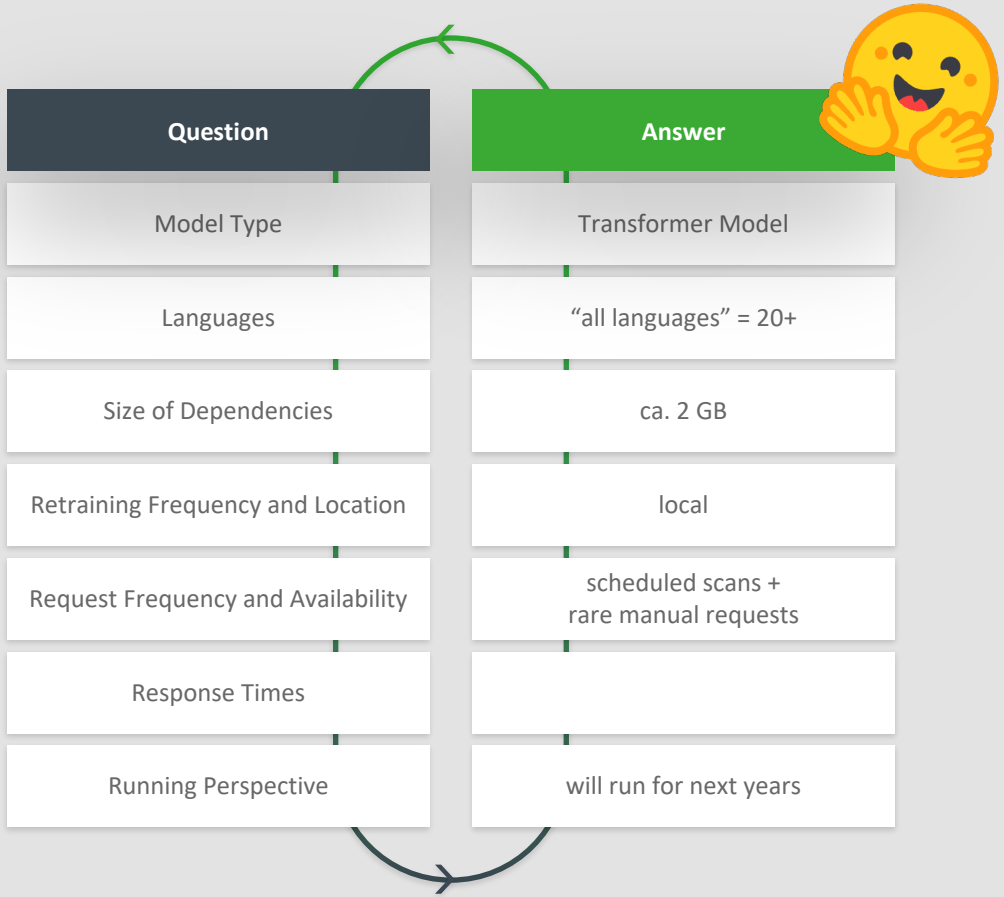
Use Case 2 – Text classification

Updating Requirements

Feature	Characteristic
Model Type	"plain" NLP
Languages	English + German
Size of Dependencies	ca. 300 MB
Retraining Frequency and Location	none
Request Frequency and Availability	scheduled scans + rare manual requests
Response Times	
Running Perspective	PoC

Question	Answer
Model Type	**Transformer Model**
Languages	**"all languages" = 20+**
Size of Dependencies	**ca. 2 GB**
Retraining Frequency and Location	none
Request Frequency and Availability	scheduled scans + rare manual requests
Response Times	
Running Perspective	**will run for the next years**

Use Case 2 – Text classification



Question	Answer
Model Type	Transformer Model
Languages	“all languages” = 20+
Size of Dependencies	ca. 2 GB
Retraining Frequency and Location	local
Request Frequency and Availability	scheduled scans + rare manual requests
Response Times	
Running Perspective	will run for next years

Cloud Foundry (SAP Business Technology Platform)

Preferred:

- Dedicated AI runtime not available

  SAP AI Core (Managed Kubernetes Cluster)

Workarounds

App Size

- Save model in external BLOB storage
- Convert model to ONNX with smaller runtime

ONNX

Memory

- Limit batch classifications

NLPOps Checklist
☑ Model Type
☑ Languages
☑ Size of Dependencies
☑ Retraining Frequency and Location
☑ Request Frequency and Availability
☑ Response Times
☑ Running Perspective
☑ Monitoring / Security
☑ Running Context / Existing Infrastructure

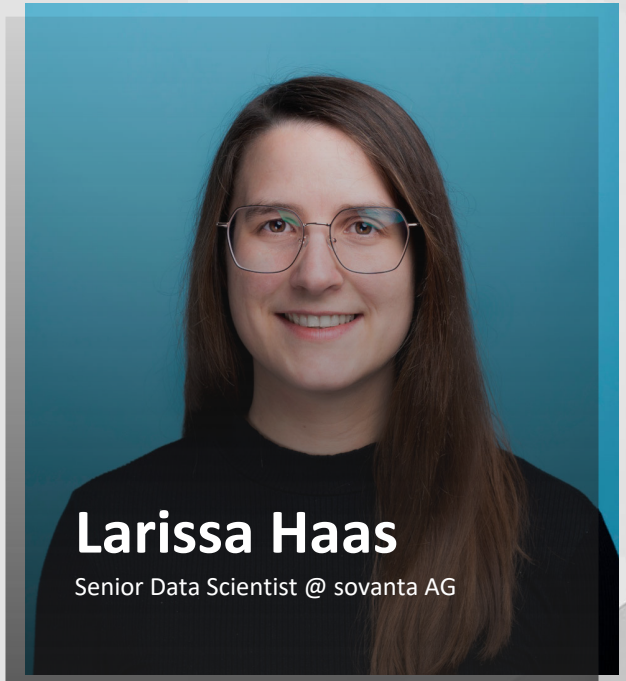
Key Takeaways and Further Reading

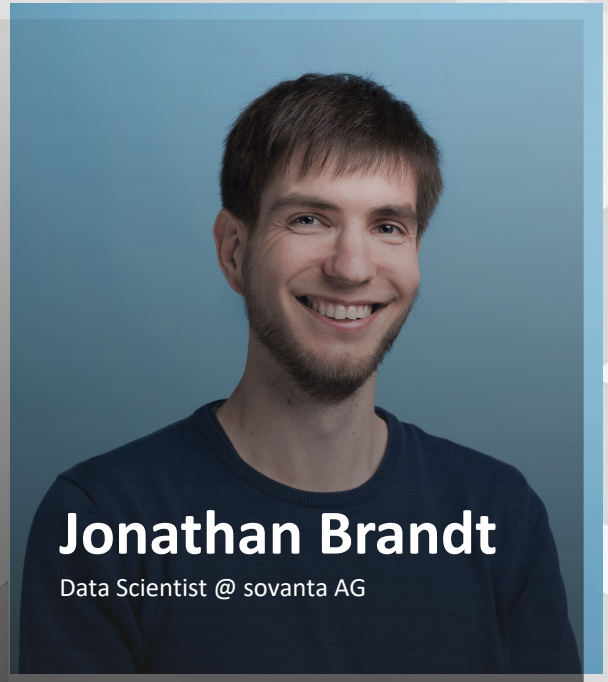
- Think about your goal with the model beforehand
- Clarify all options/restrictions you have
- Make yourself familiar with condensing mechanisms that work for your model
- Take this as guideline, there is no black and white

Blog post about ONNX and CF

Or Read more here

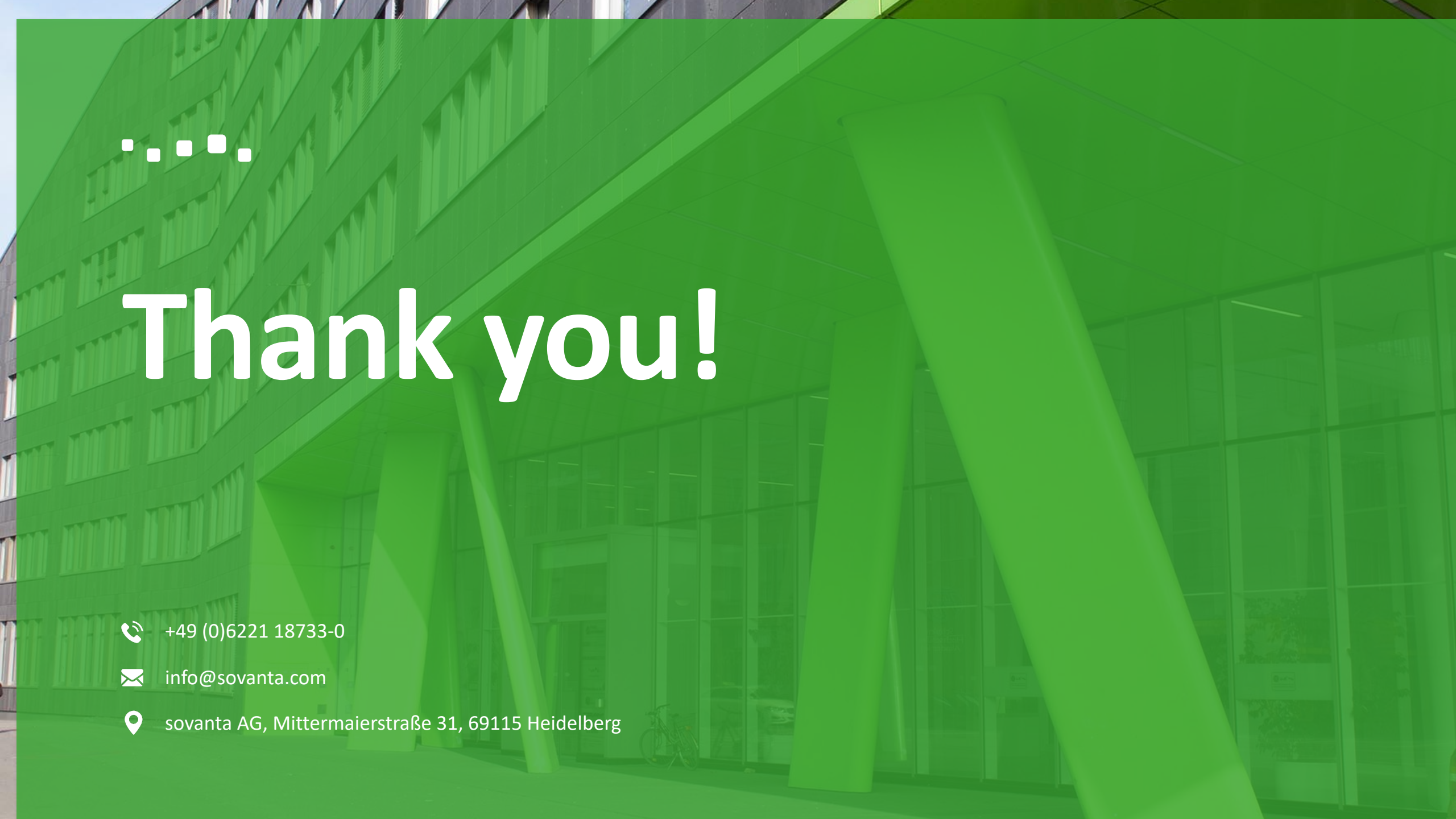
Questions?





Find the talk slides here:

sovanta / We are hiring!



Thank you!

+49 (0)6221 18733-0

info@sovanta.com

sovanta AG, Mittermaierstraße 31, 69115 Heidelberg

NLPOps Checklist	Explanation
☑ Model Type	“Plain” NLP / SpaCy / Huggingface / Transformers
☑ Languages	One or two languages / More languages / Multilanguage Model / Translation Service available?
☑ Size of Dependencies	Pytorch / Tensorflow / SpaCy LMs / …
☑ Retraining Frequency and Location	Weekly / sporadically / locally
☑ Request Frequency and Availability	Scale automatically for frequent requests / high request times
☑ Response Times	Certain requirements for response times / not important because of scheduled runs
☑ Running Perspective	PoC / Further development planned
☑ Monitoring / Security	Requirement for extensive monitoring / already existing for specific infrastructure
☑ Running Context / Existing Infrastructure	Other services running on specific infrastructure? Dependencies to certain services / applications?