netcompany

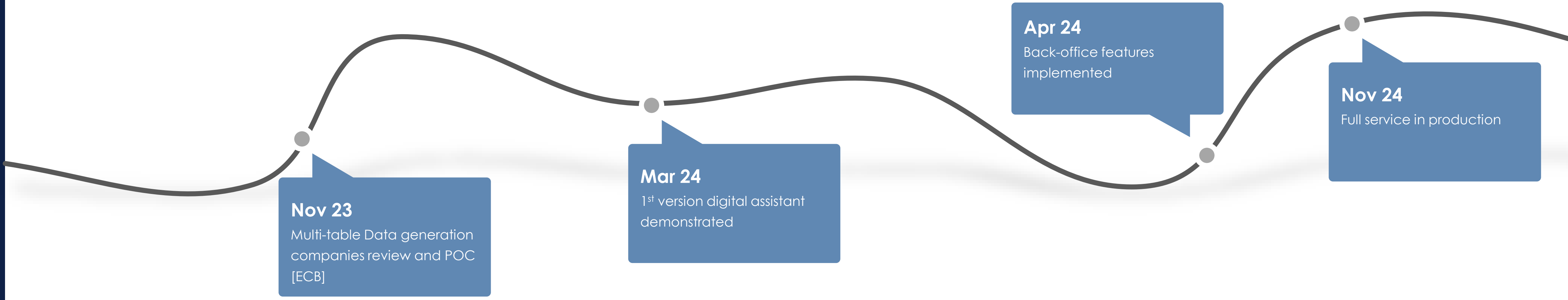
intrasoft

How to turn an on-premises LLM into a Digital Assistant for the EU

K. Chasapas, I.T. Christou, G. Lalas, P. Lapas, M. Logothetis, K. Thivaios
Research & Innovation Development
Netcompany-Intrasoft

Story, Vision and Roadmap

- **Multi-table Data generation companies review and POC: Nov-2023** we have conducted a scouting activity for **ECB** (European Central bank) to provide a benchmarking for all synthetic data providers, based on ECB's requirements. This has led to a POC with MostlyAI. It is the first time that we have identified the need for an **on-premise domain-specific LLM**
- -- research projects (there was the need for such IT solution) and for AI characteristics (ISO certification: un-BIAS, certification, traceability [evenflow])
- **Xxx digital assistant** has been our trigger point to design and develop our **on-premise LLM approach**



Nov 23
Multi-table Data generation companies review and POC [ECB]

Mar 24
1st version digital assistant demonstrated

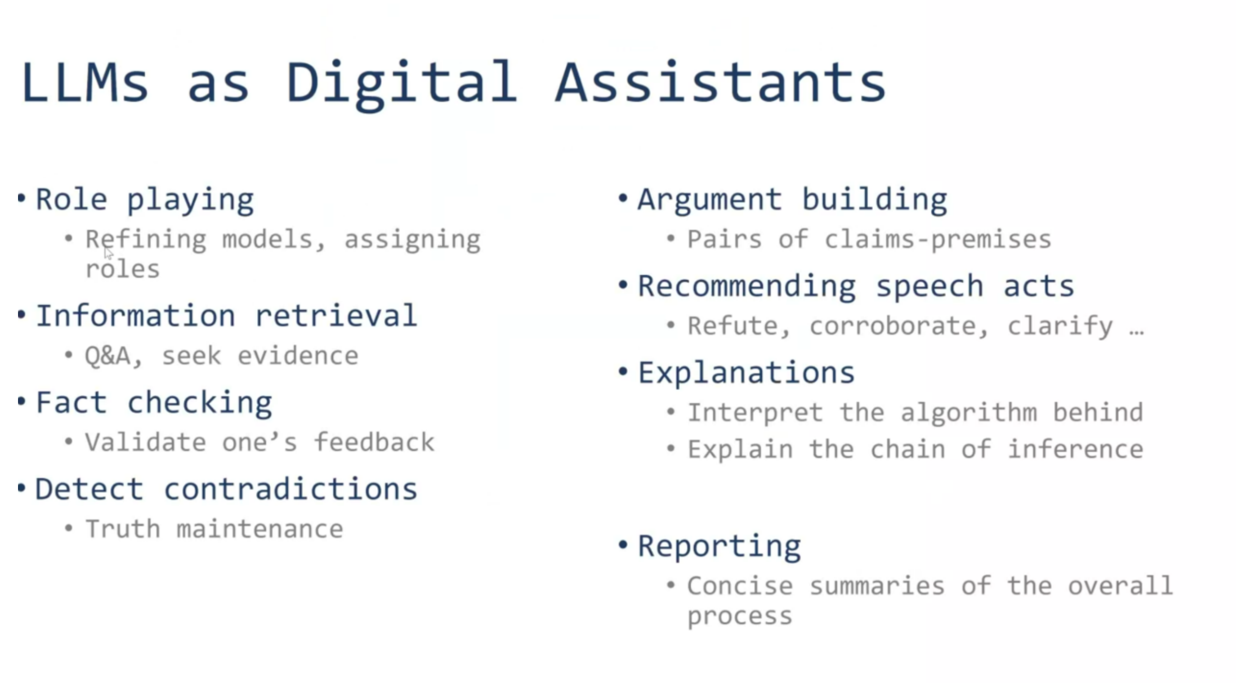
Apr 24
Back-office features implemented

Nov 24
Full service in production

Goal: Build an **on-premise domain-specific LLM,** to perform well-defined tasks dictated by organizational guidelines, taking into consideration corporate data, product data, corporate policies, and industry terminologies.

Domain-specific LLM and main use cases

- **Domain-specific Large Language Model (LLM),** designed for deployment within a secure, on-site infrastructure, equipped with security protocols to safely handle and analyze sensitive information, ensuring adherence to sector-specific data protection standards.

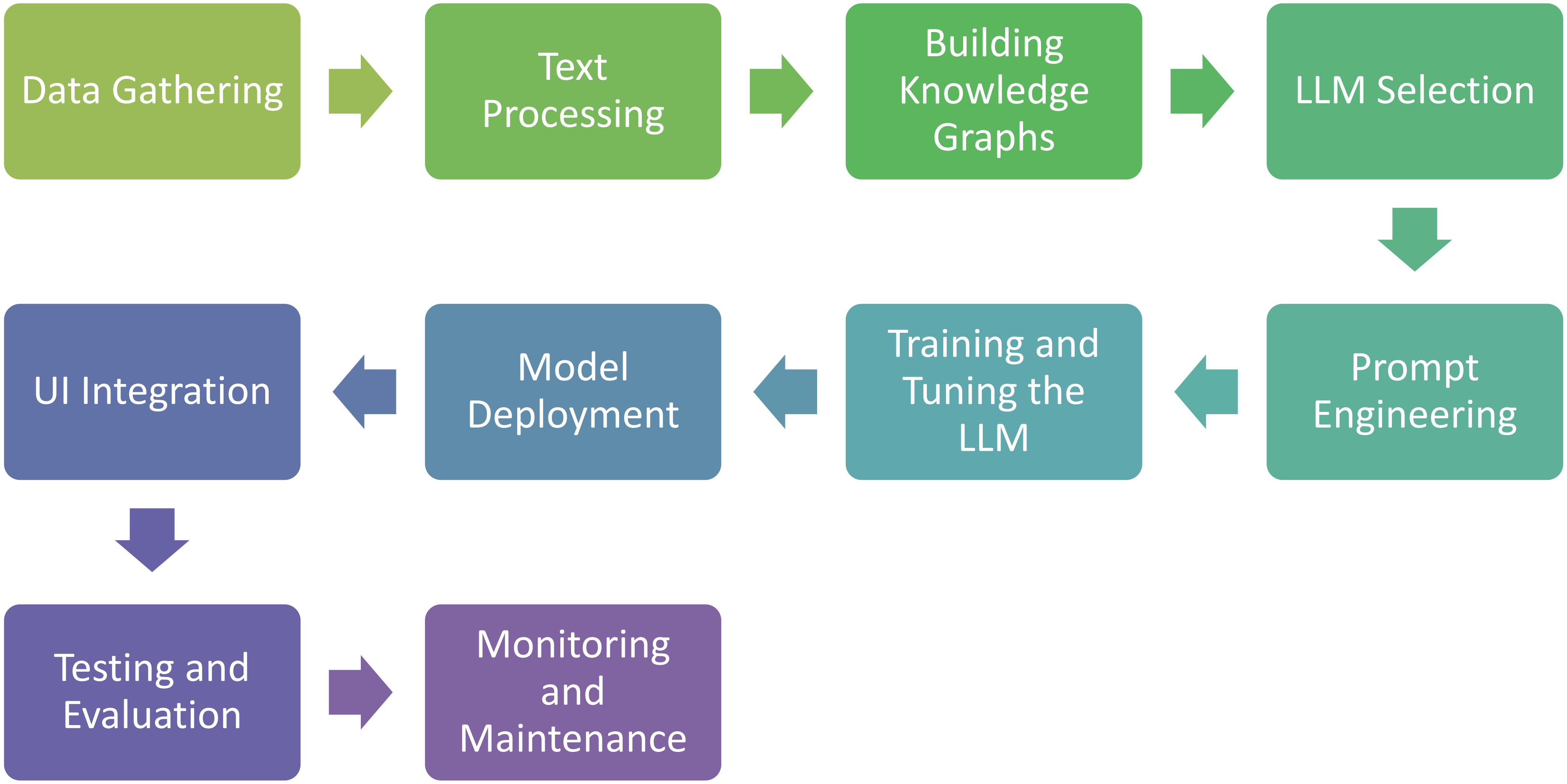


LLMs as Digital Assistants

- Role playing
  - Refining models, assigning roles
- Information retrieval
  - Q&A, seek evidence
- Fact checking
  - Validate one’s feedback
- Detect contradictions
  - Truth maintenance

- Argument building
  - Pairs of claims-premises
- Recommending speech acts
  - Refute, corroborate, clarify …
- Explanations
  - Interpret the algorithm behind
  - Explain the chain of inference
- Reporting
  - Concise summaries of the overall process

On-premise pipeline



Data Gathering

Text Processing

Building Knowledge Graphs

LLM Selection

Prompt Engineering

Training and Tuning the LLM

Model Deployment

UI Integration

Testing and Evaluation

Monitoring and Maintenance

On-premise pipeline

✓ Data Gathering

- Use python-oriented libraries for extracting text from PDFs, Excel and Word files
- Store the extracted text data in an open-source DB

✓ Text Processing

- Utilize python libraries for text preprocessing tasks such as tokenization etc.
- Implement custom text cleaning functions to remove noise, special characters, and perform normalization.
- Explore techniques like word embedding for capturing semantic relationships between words.
- Augment text preprocessing with XAI techniques to provide insights into model predictions.
- Ensure transparency and trustworthiness by documenting preprocessing steps and transformations applied to the text data.

Utilize open-source libraries like Hugging Face's **transformers** for implementing prompt-based fine-tuning and generation.

On-premise pipeline

✓ Building Knowledge Graphs

- Construct knowledge graphs using open-source libraries.
- Use techniques such as Named Entity Recognition (NER) and Dependency Parsing to extract entities and relationships from text data.
- Consider graph database technologies for storing and querying knowledge graphs efficiently.
- Enrich knowledge graph construction with explainable methods for relationship inference.

✓ Prompt Engineering

- Employ techniques for prompt engineering to fine-ture the language model for specific tasks or domains.
- Experiment with different prompt templates, prefixing strategies, and control codes to guide the model's generation process effectively.
- Utilize open-source libraries like Hugging Face’s ‘transformers’ for implementing prompt-based fine-tuning and generation.
- Incorporate explainable prompt engineering techniques by providing users with control over prompt generation parameters and displaying generated prompts alongside model predictions.
- Implement mechanisms to explain how prompts influence model behavior and predictions to enhance trustworthiness.

Utilize open-source libraries like Hugging Face's **transformers** for implementing prompt-based fine-tuning and generation.

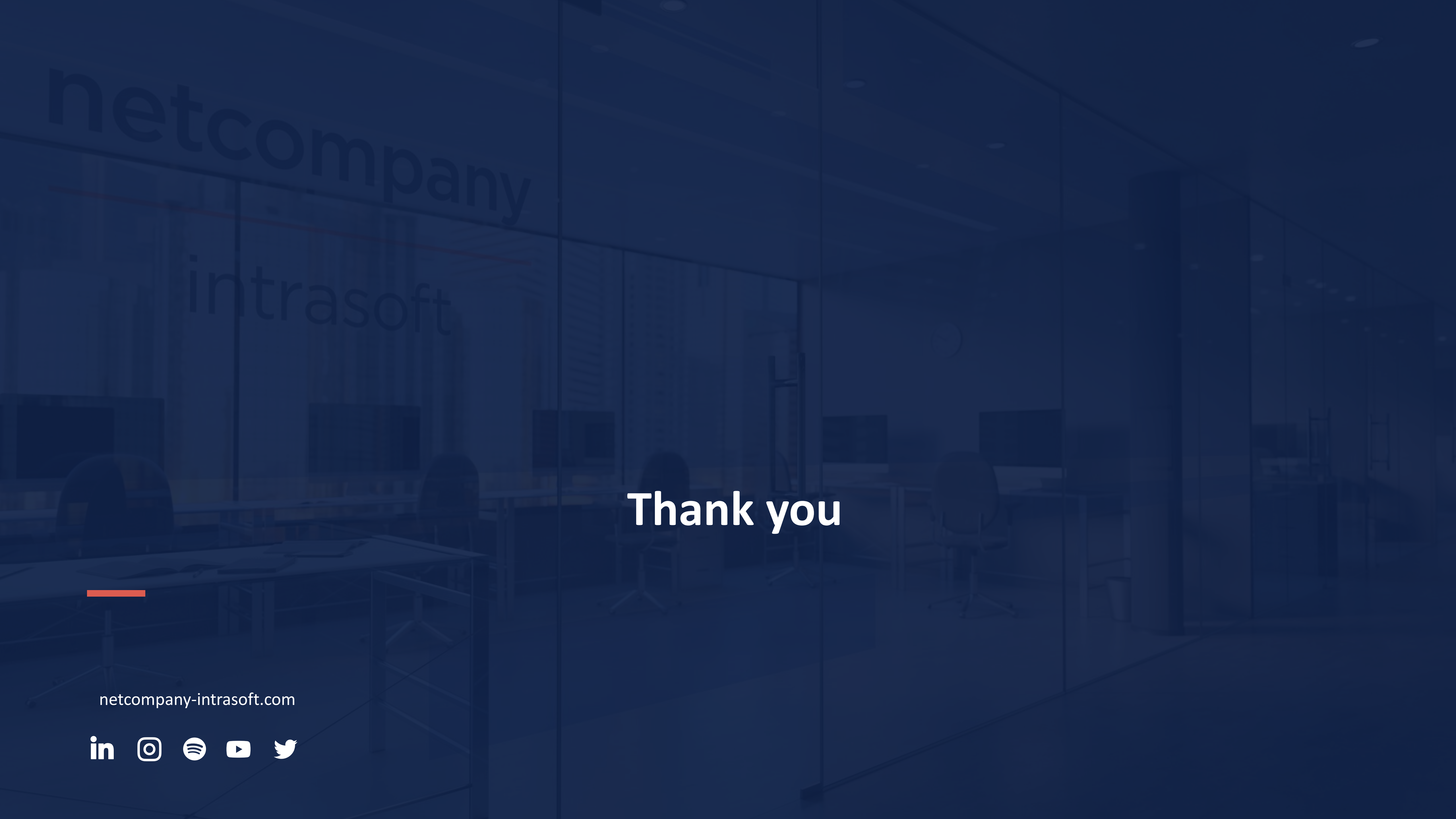
On-premise pipeline

✓ Training and Tuning the LLM

- Leverage Hugging Face’s ‘transformers’ library fir training and fine-tuning pre-trained language models such as Mistral, Falcon, etc.
- Utilize open-source frameworks like PyTorch or TensorFlow for model training, allowing for flexibility and customization.
- Explore techniques like transfer learning and domain adaptation to adapt pre-trained models to specific tasks or domains.

✓ Model Deployment

- Deploy the trained language model as a RESTful API using open-source frameworks like Flask or FastAPI for serving predictions over HTTP.
- Utilize containerization technologies like Docker for packaging the model and its dependencies into lightweight and portable containers.
- Host the deployed model on open-source platforms like Kubernetes for scalability and resource management.



Thank you

netcompany-intrasoft.com