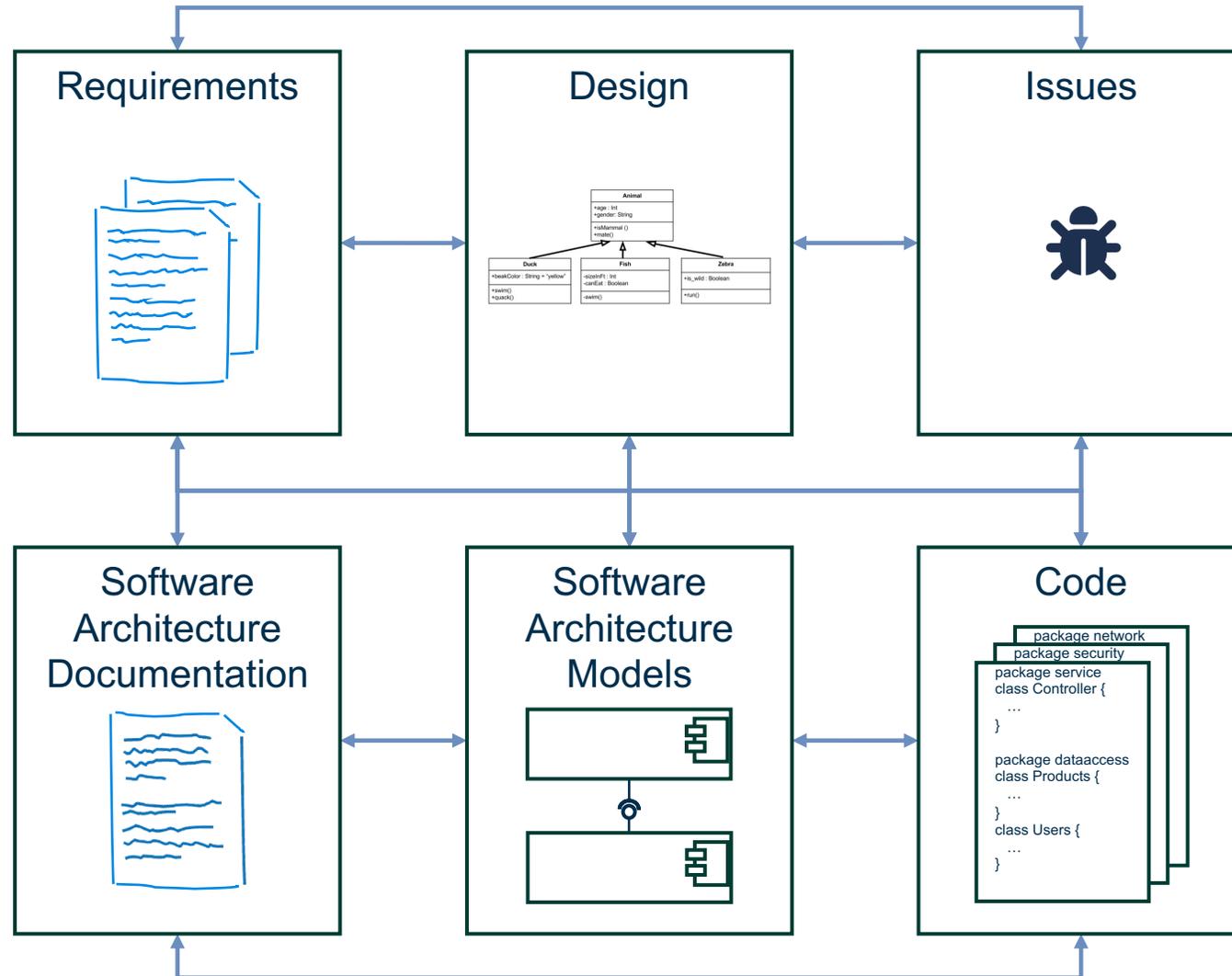




# LiSSA: Toward Generic Traceability Link Recovery through Retrieval- Augmented Generation

Dominik Fuchß, Tobias Hey, Jan Keim, Haoyu Liu, Niklas Ewald, Tobias Thirolf, Anne Kozirolek  
*SE 2026, Bern (Originally presented at ICSE 2025 in Ottawa, Canada)*

# What makes Trace Links important?



Trace links are evidently useful for

Software Maintenance

Bug Localization

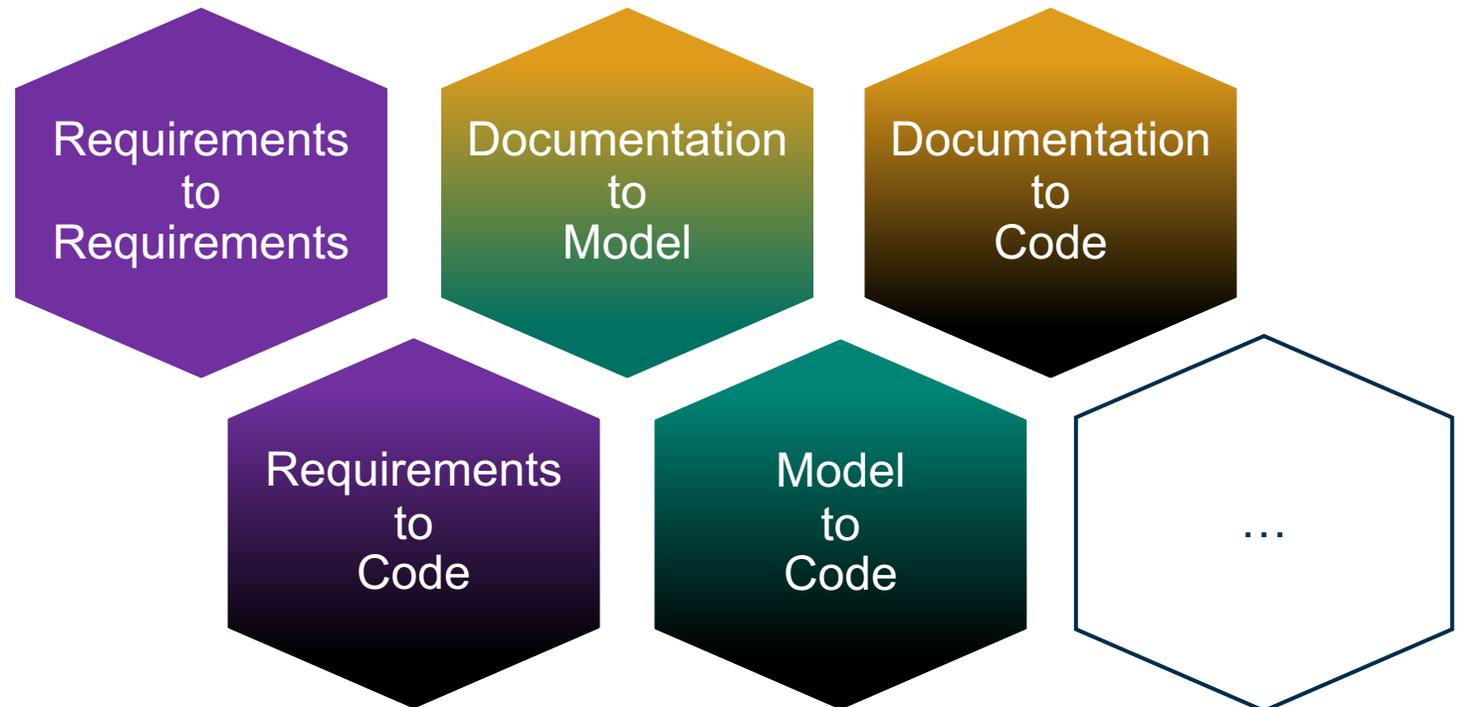
Change Impact An.

System Security

...

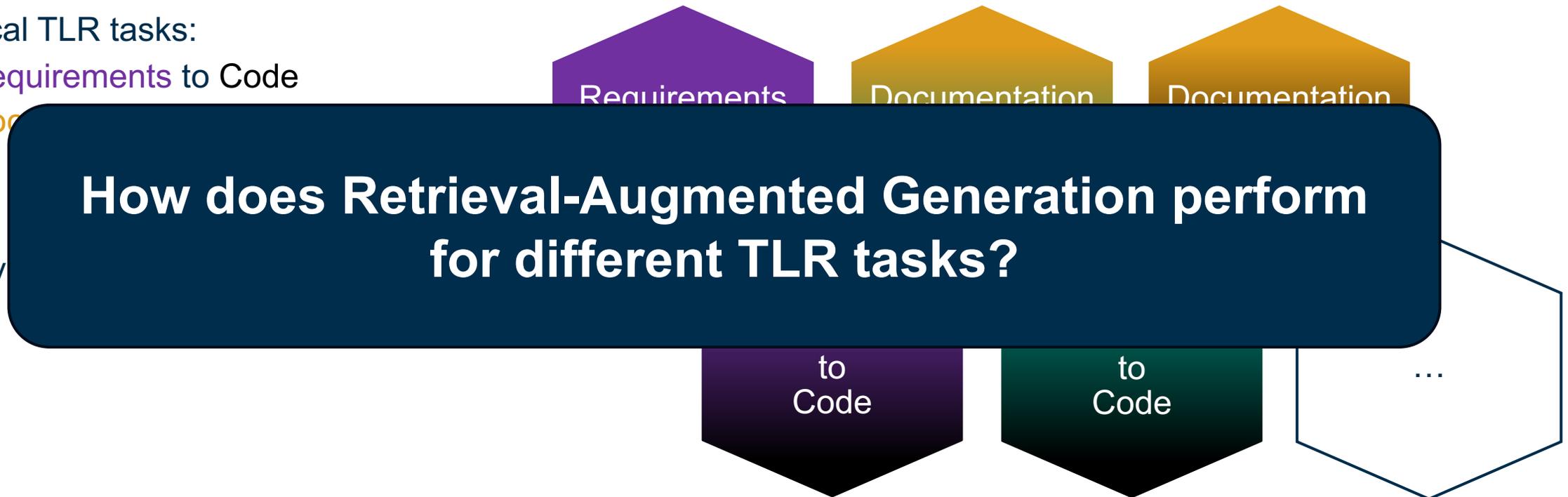
# What makes Trace Links important?

- Multitude of different artifacts
- Typical TLR tasks:
  - Requirements to Code
  - Documentation to Code
  - ...
- Many specialized approaches



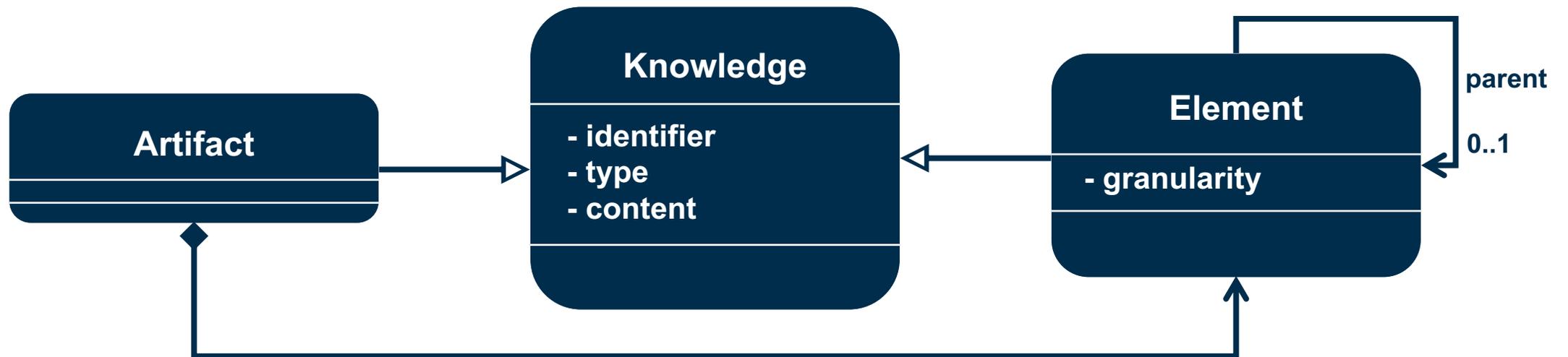
# What makes Trace Links important?

- Multitude of different artifacts
- Typical TLR tasks:
  - Requirements to Code
  - Doc
  - ...
- Many

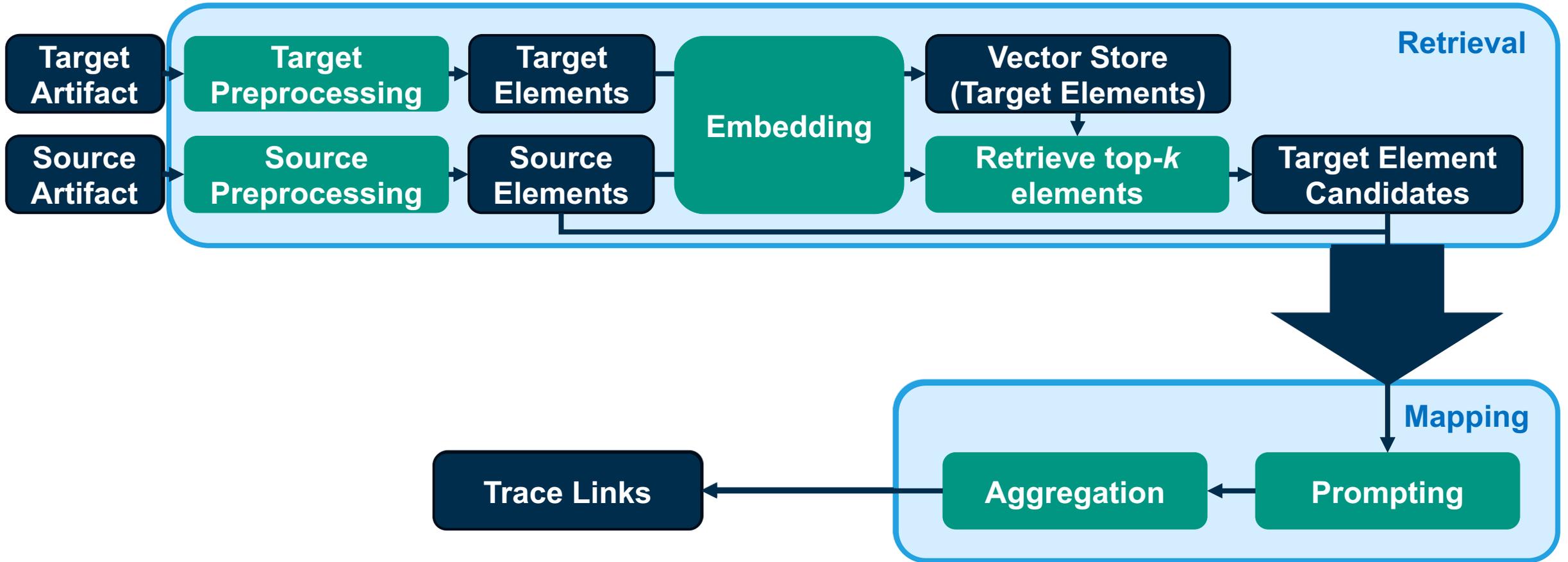


# LiSSA: Concepts

- Similar treatment of similar artifacts
  - Code-like artifacts (e.g., source code, test code)
  - Natural language artifacts (e.g., requirements, documentation, issues)
  - Structural model artifacts (e.g., UML component models)



# LiSSA: Process



# LiSSA: Process



**Preprocessing:** Extract *elements* from *artifacts*

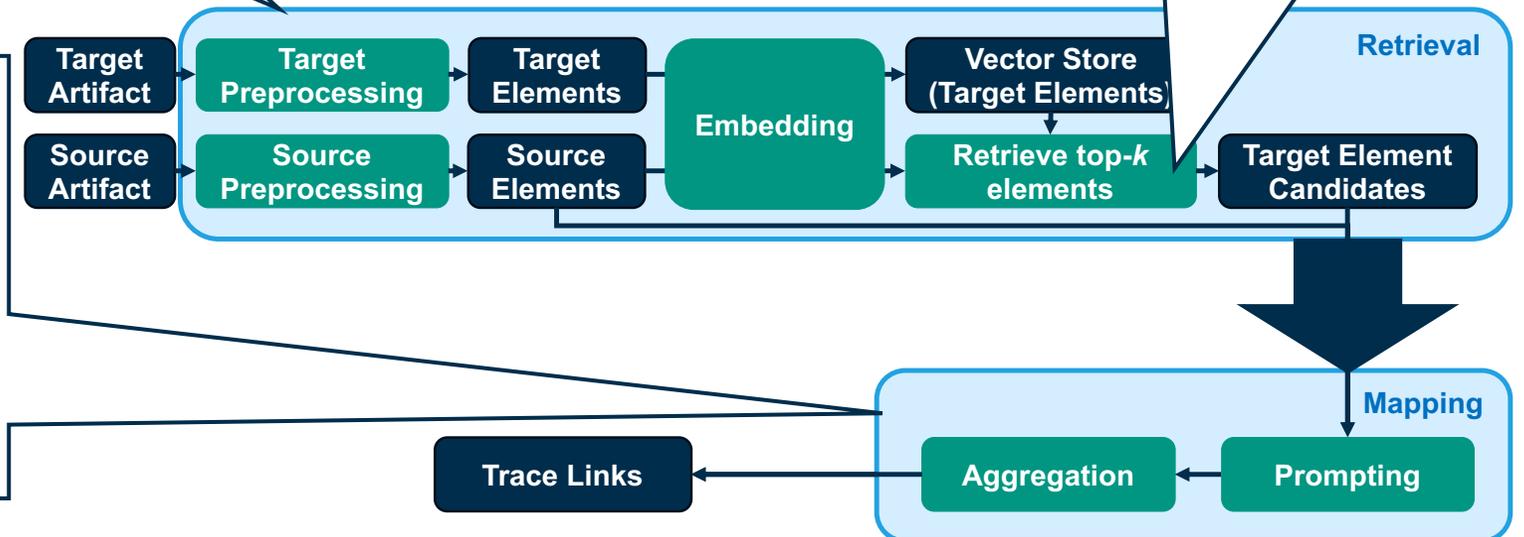
- No preprocessing
- Code chunking (fixed size)
- Code method splitting
- Model element extraction
- Sentence splitting

**Retrieval:** Identify related elements

- Embedding-based
- Ensemble of LLMs (AIRE25)

**Prompting:** Classify whether elements belong to each other

- No prompt: IR baseline
- Zero-Shot Yes/No Classification
- Chain-of-thought



# Evaluation

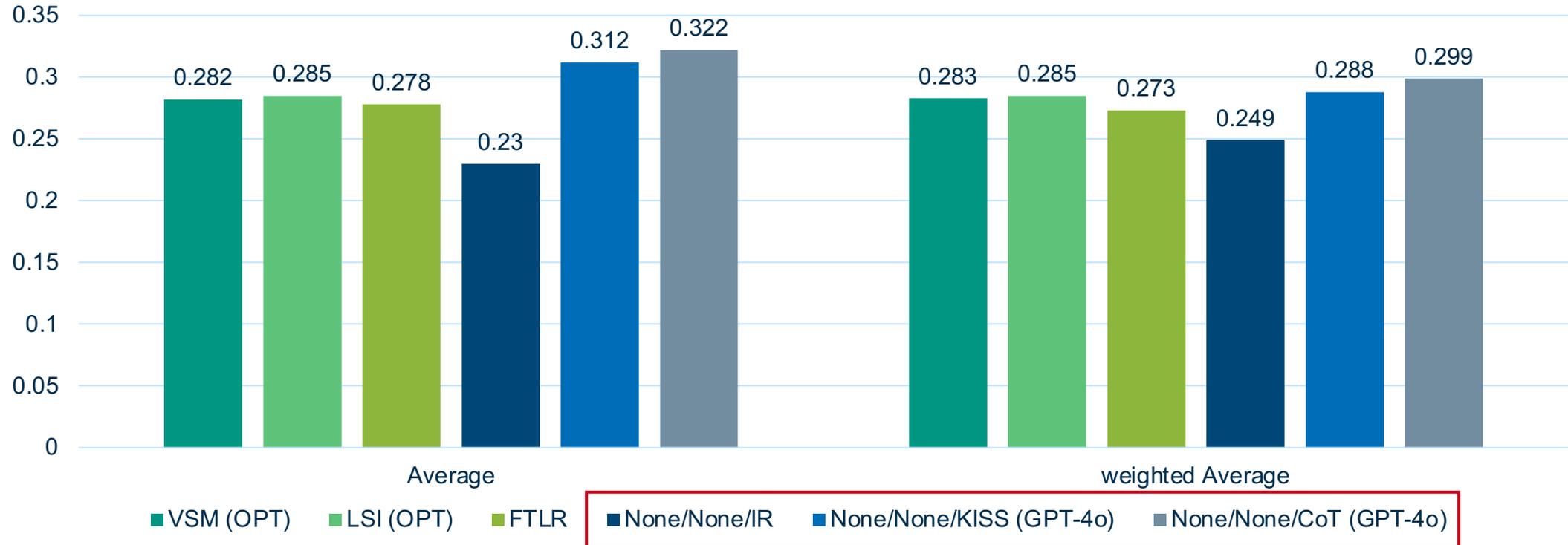
Requirements  
to Code TLR

Documentation  
to Code TLR

Documentation  
to Model TLR

# Evaluation: Requirements to Code

## F1-score for Requirements to Code TLR



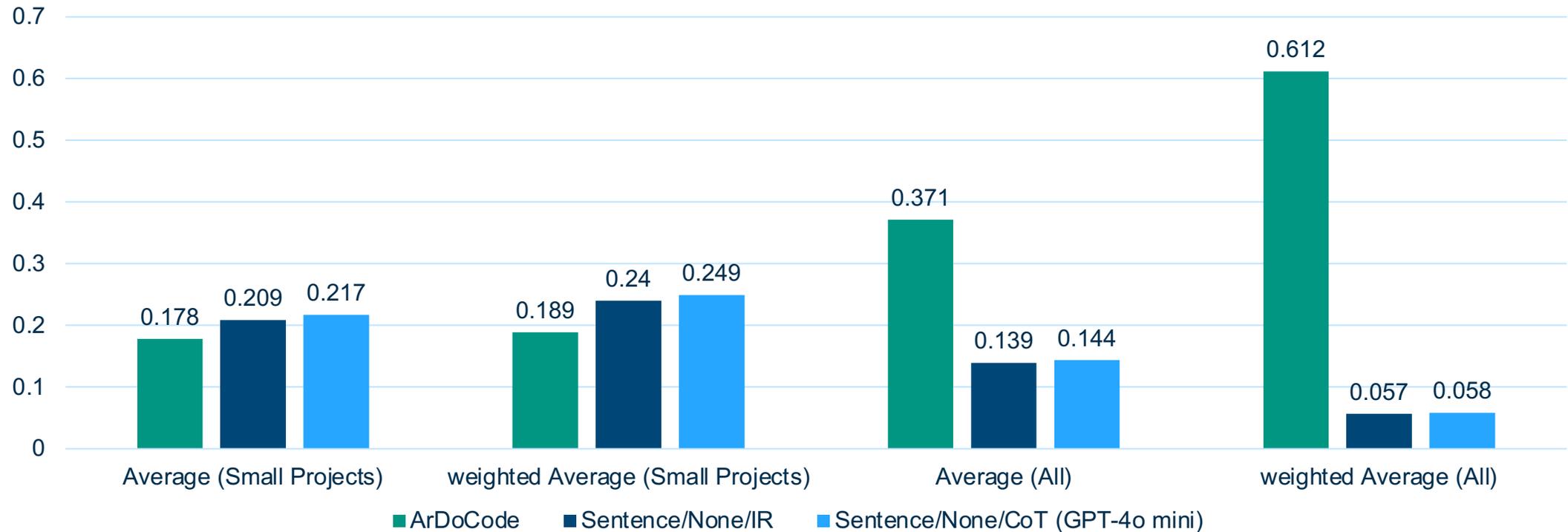
Requirements to Code TLR

Documentation to Code TLR

Documentation to Model TLR

# Evaluation: Documentation to Code

F1-score for Documentation to Code TLR



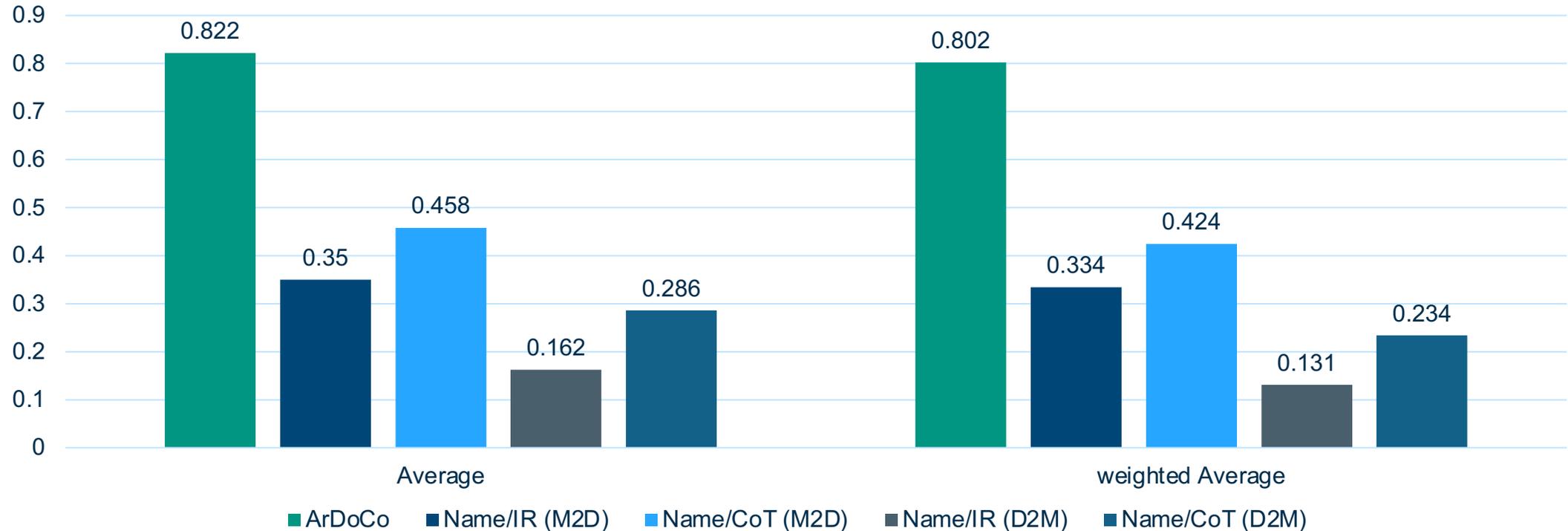
Requirements  
to Code TLR

Documentation  
to Code TLR

Documentation  
to Model TLR

# Evaluation: Documentation to Model

F1-score for Documentation to Model TLR



Requirements  
to Code TLR

Documentation  
to Code TLR

Documentation  
to Model TLR

# Conclusion

- We presented LiSSA, a generic TLR framework that uses RAG
- In the evaluation,
  - Our approach can significantly outperform state-of-the-art for requirement to code TLR – avg. F1: 0.278 (FTLR) vs. 0.322 (GPT-4o + CoT)
  - Chain-of-thought (CoT) prompting was on average more effective than simple classification prompting
  - Artifact-to-artifact TLR was (on average) better than fine-grained mappings
- Outlook:
  - Replace IR-based retrieval with small LLMs (see <https://ardoco.de/c/aire25>)
  - Inter-requirements TLR with LiSSA (see <https://ardoco.de/c/refsq25>)
  - Documentation to code TLR (see <https://ardoco.de/c/icsa25>)
  - Automatic prompt engineering (currently WIP)

