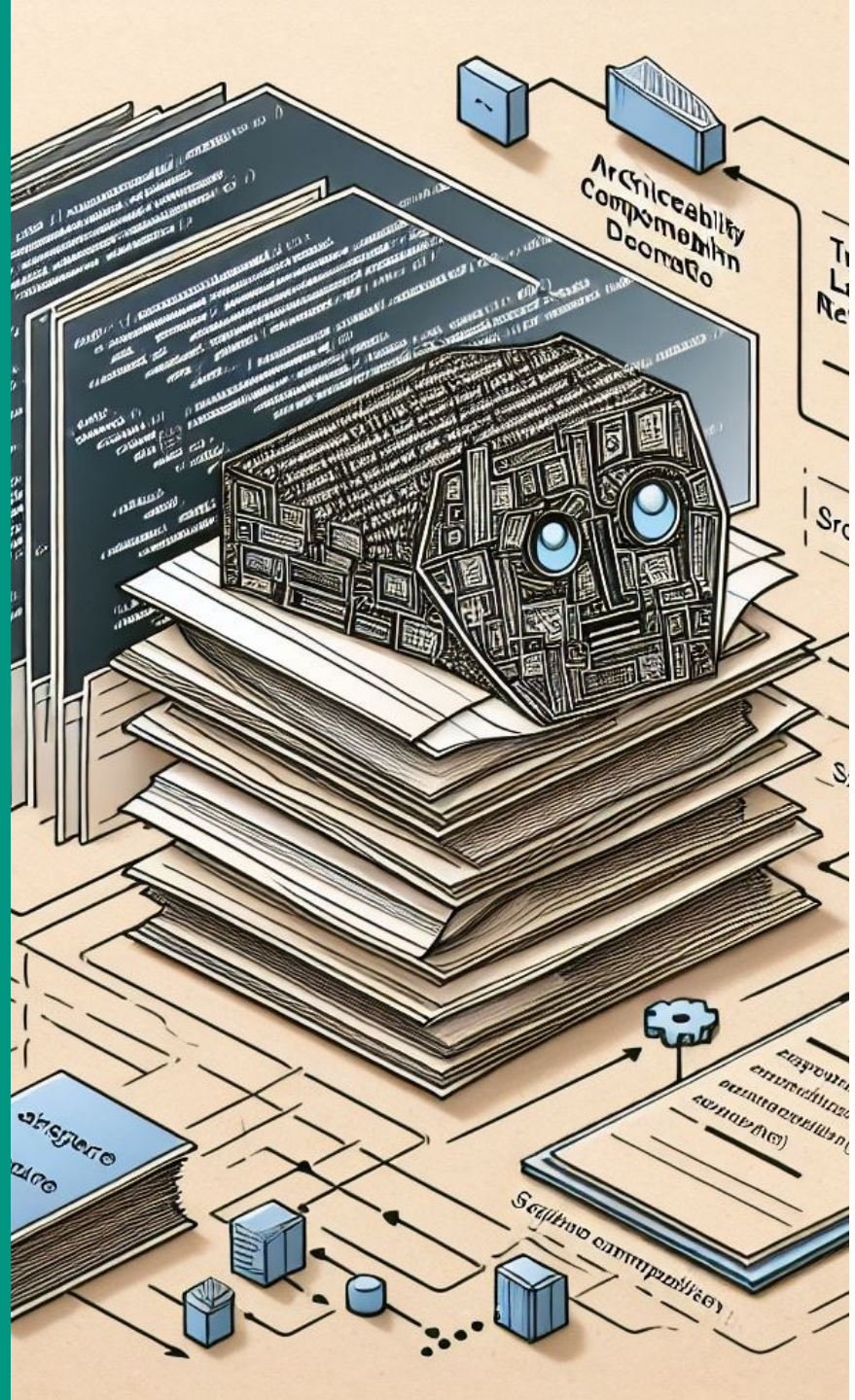
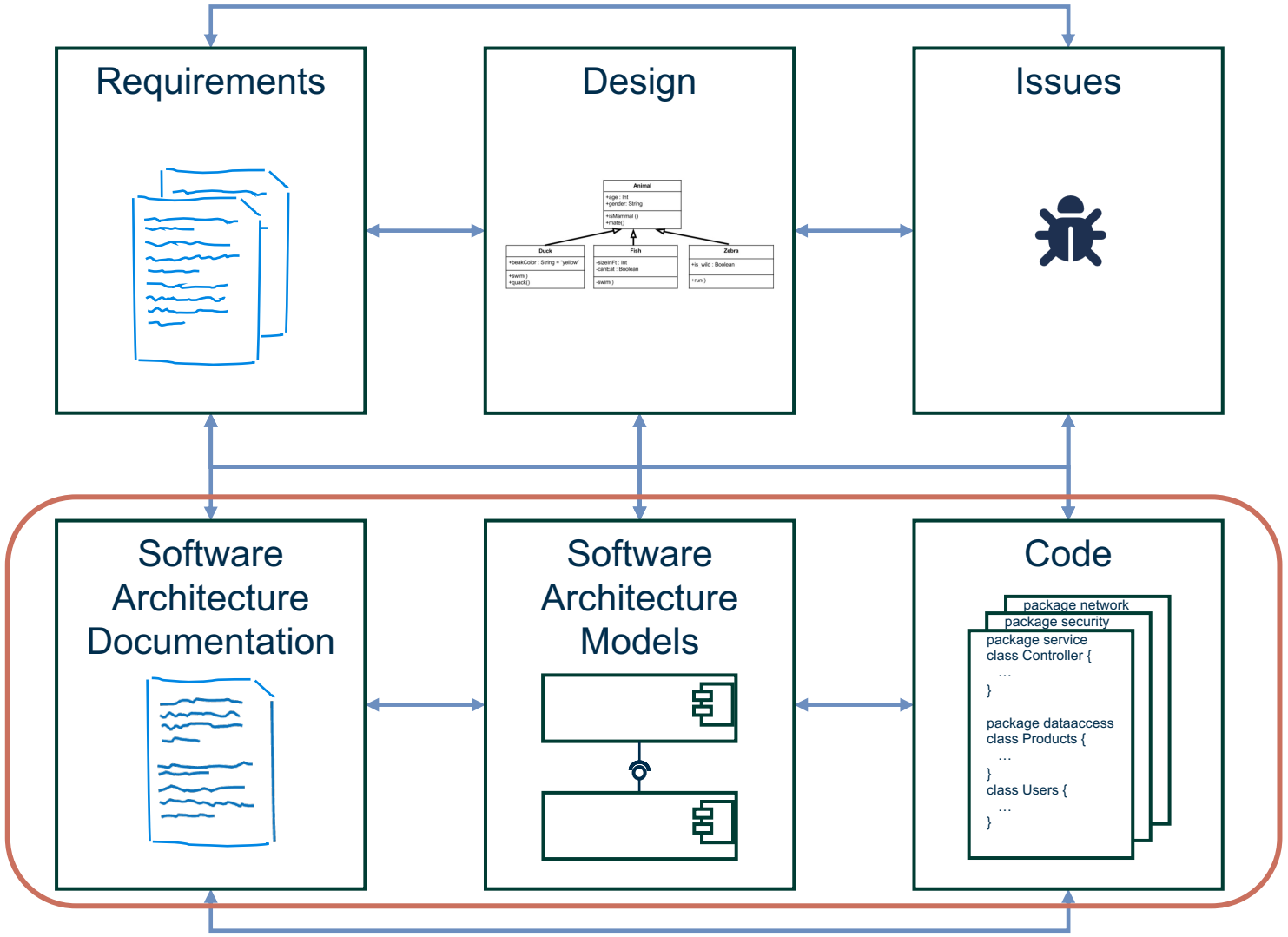
Enabling Architecture Traceability by LLM-based Architecture Component Name Extraction



Dominik Fuchß, Haoyu Liu, Tobias Hey, Jan Keim, Anne Koziolek, KASTEL – Institute of Information Security and Dependability

What makes Trace Links important?

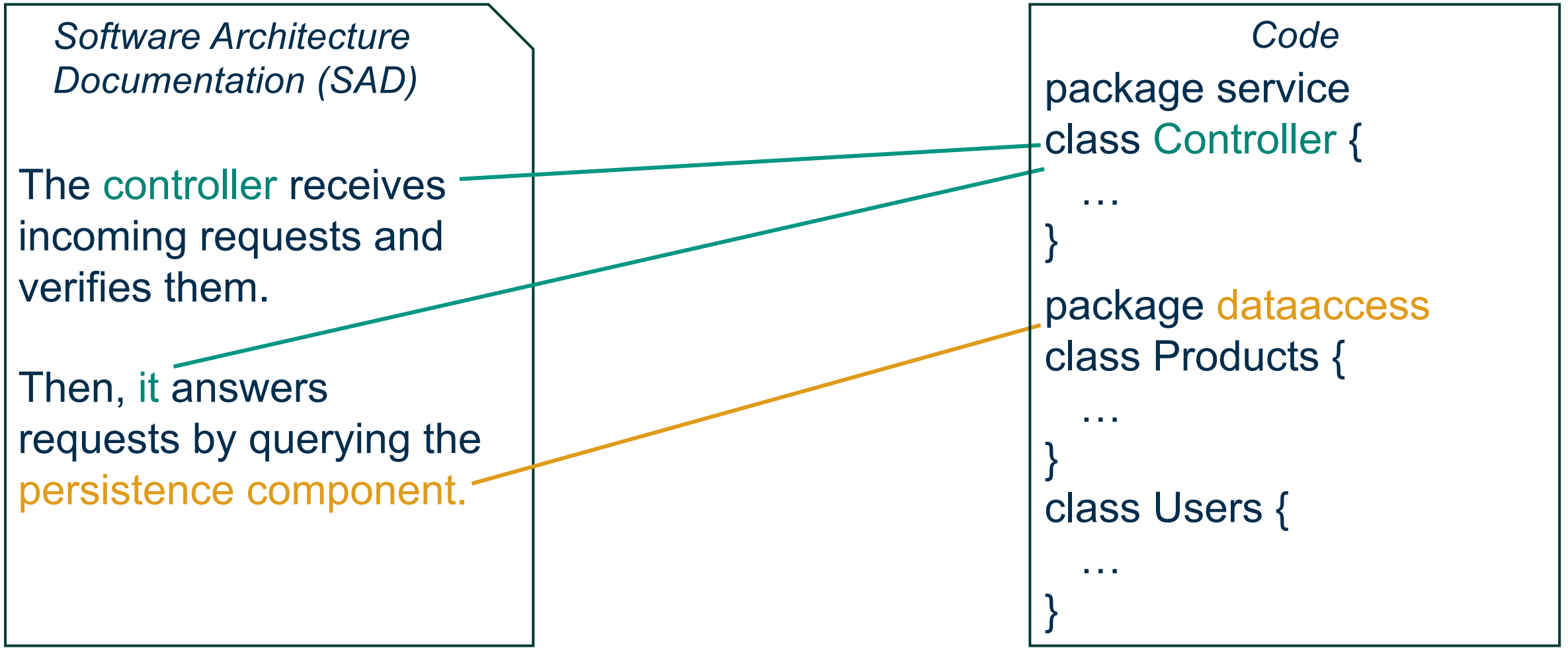


Trace links are evidently useful for

- Software Maintenance
- Bug Localization
- Change Impact An.
- System Security
- …

Artifact
Relation

Traceability Link Recovery between Documentation & Code

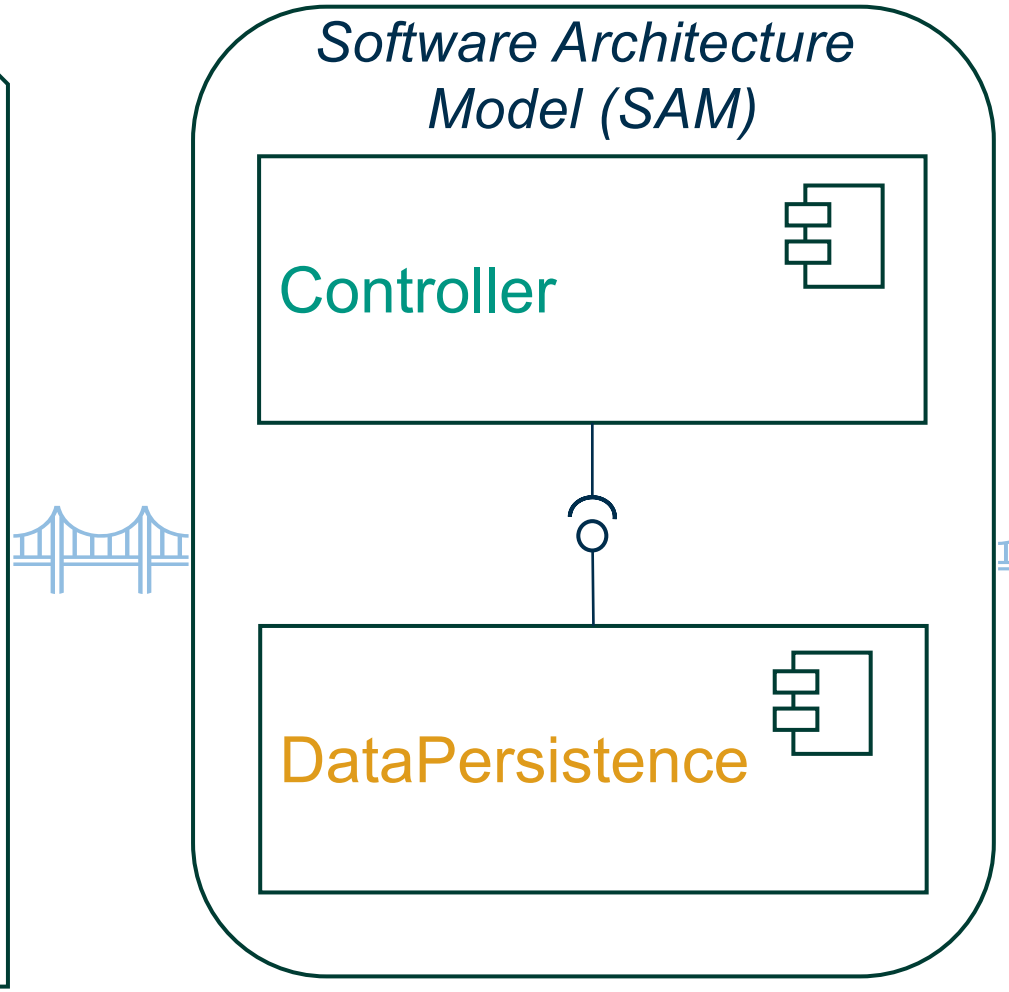


Traceability Link Recovery between Documentation & Code

Software Architecture Documentation (SAD)

The controller receives incoming requests and verifies them.

Then, it answers requests by querying the persistence component.



Code

```
package service
class Controller {
    …
}
package dataaccess
class Products {
    …
}
class Users {
    …
}
```

Traceability Link Recovery between Documentation & Code

Software Architecture Documentation (SAD)

The controller re
incoming reque
verifies them.

Then, it answers requests by querying the persistence component.

Software Architecture Model (SAM)

Code

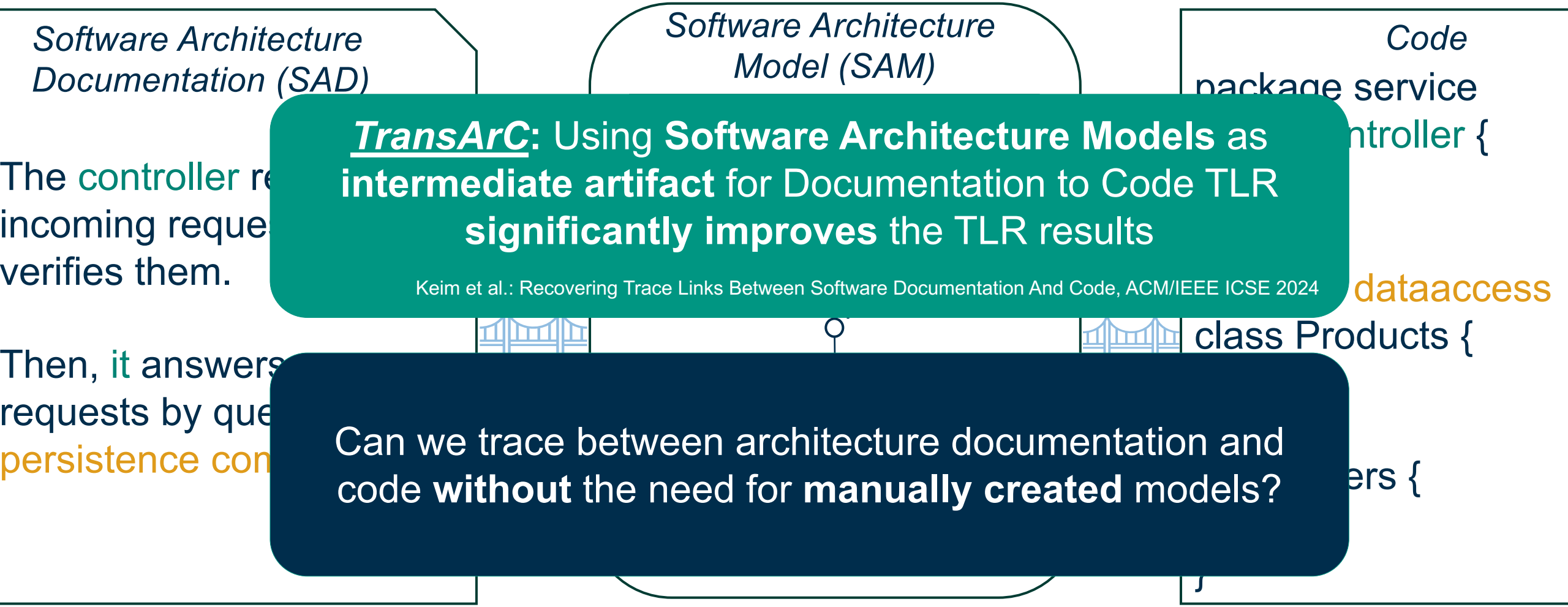
package service
ntroller {

dataaccess
class Products {
…
}
class Users {
…
}

***TransArC*: Using **Software Architecture Models** as **intermediate artifact** for Documentation to Code TLR **significantly improves** the TLR results**

Keim et al.: Recovering Trace Links Between Software Documentation And Code, ACM/IEEE ICSE 2024

Traceability Link Recovery between Documentation & Code

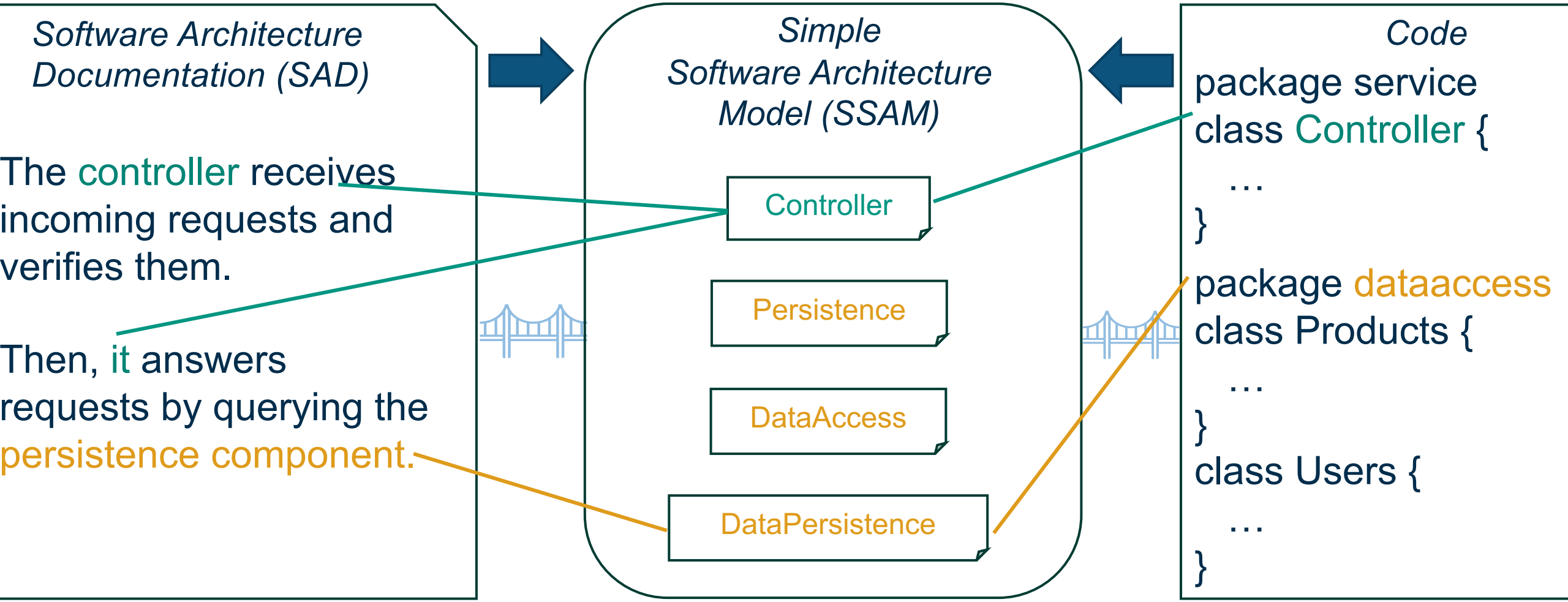


***TransArC*: Using **Software Architecture Models** as **intermediate artifact** for Documentation to Code TLR **significantly improves** the TLR results**

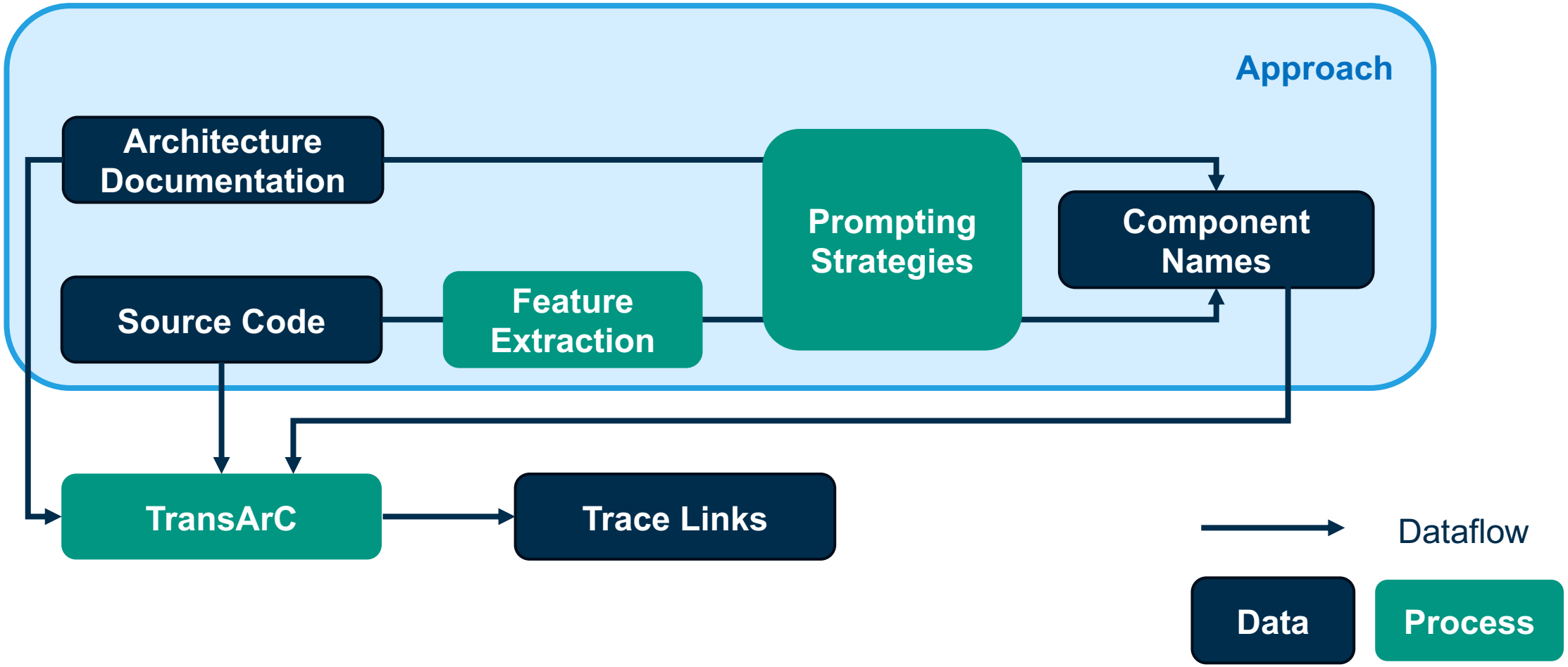
Keim et al.: Recovering Trace Links Between Software Documentation And Code, ACM/IEEE ICSE 2024

Can we trace between architecture documentation and code **without** the need for **manually created** models?

Traceability Link Recovery between Documentation & Code



Approach to get a Simple Software Architecture Model



Approach to get a Simple Software Architecture Model

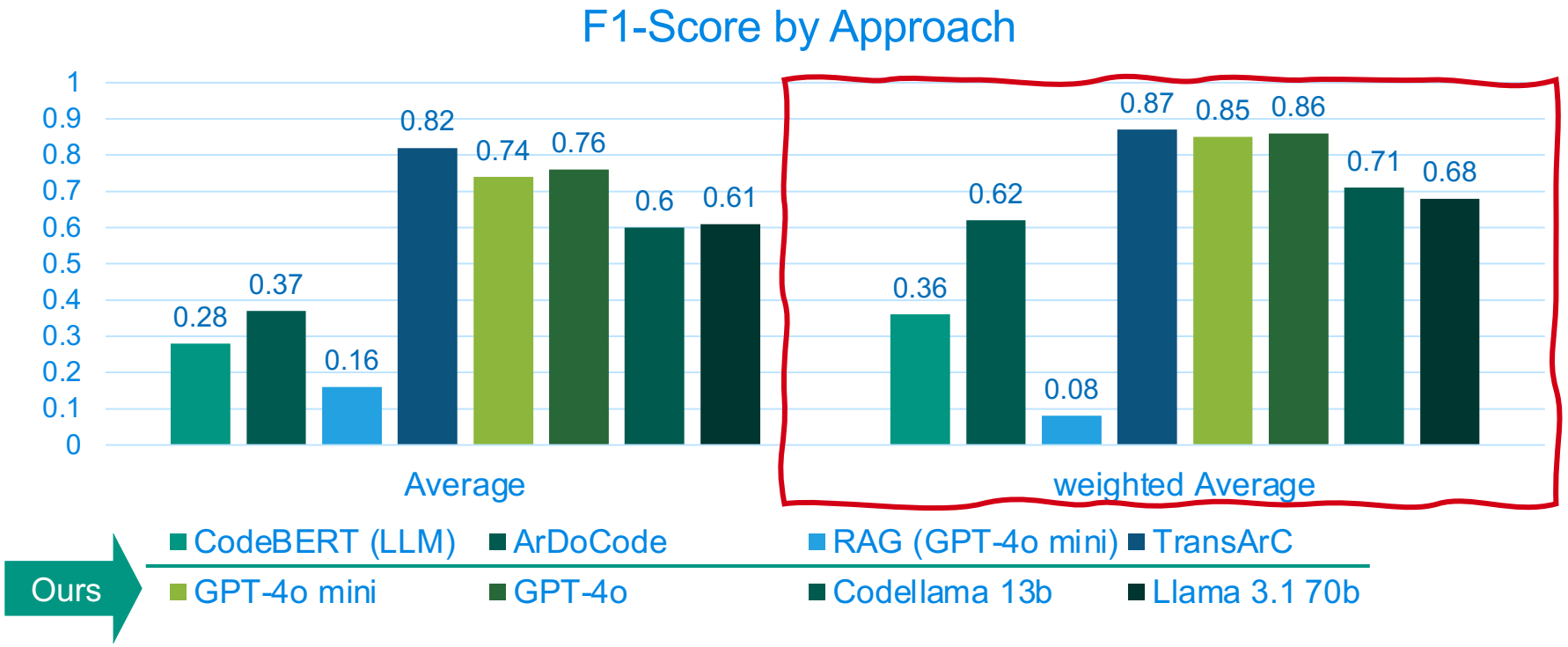
Prompt: Documentation to Architecture

Your task is to identify the **high-level components** based on the software architecture documentation.

In a first step, you shall elaborate on the following documentation:

{*Software Architecture Documentation*}

Eval: Component Names derived from Documentation



RQ 1: Comparable to TransArC ?

RQ 2: Better than SotA w/o SAMs?

RQ 3: Open-Source vs. Closed-Source LLMs?

F1-Score: Harmonic Mean of Precision and Recall

Eval: Component Names derived from ...

SSAM derived from …	Average F1	Weighted Average F1
Documentation	**0.76**	**0.86**
Code	0.58	0.81
Both (Similarity Aggregation)	0.73	0.86
Both (Prompt Aggregation)	0.72	0.85

F1-Score: Harmonic Mean of Precision and Recall

RQ 1: Comparable to TransArC ?

RQ 2: Better than SotA w/o SAMs?

RQ 3: Open-Source vs. Closed-Source LLMs?

RQ 4: Influence of different artifacts (i.e., Code/SAD)

Conclusion

- We use LLMs to derive Simple Software Architecture Models (Component Names) to support the TLR between Architecture Documentation and Code

- In the evaluation,
  - Our approach performs comparable to TransArC (best avg. F1: 0.76)
  - Extraction based on documentation often performs better than only code (best avg. F1: 0.76 vs. 0.58)
  - Fusion (Doc + Code) can reach similar, but still less good results

- Outlook:
  - Prompt Optimization might be needed (e.g., dealing with “sub-components” in the results)
  - Analysis of the different meanings of “Trace Link”

ardoco.de/c/icsa25

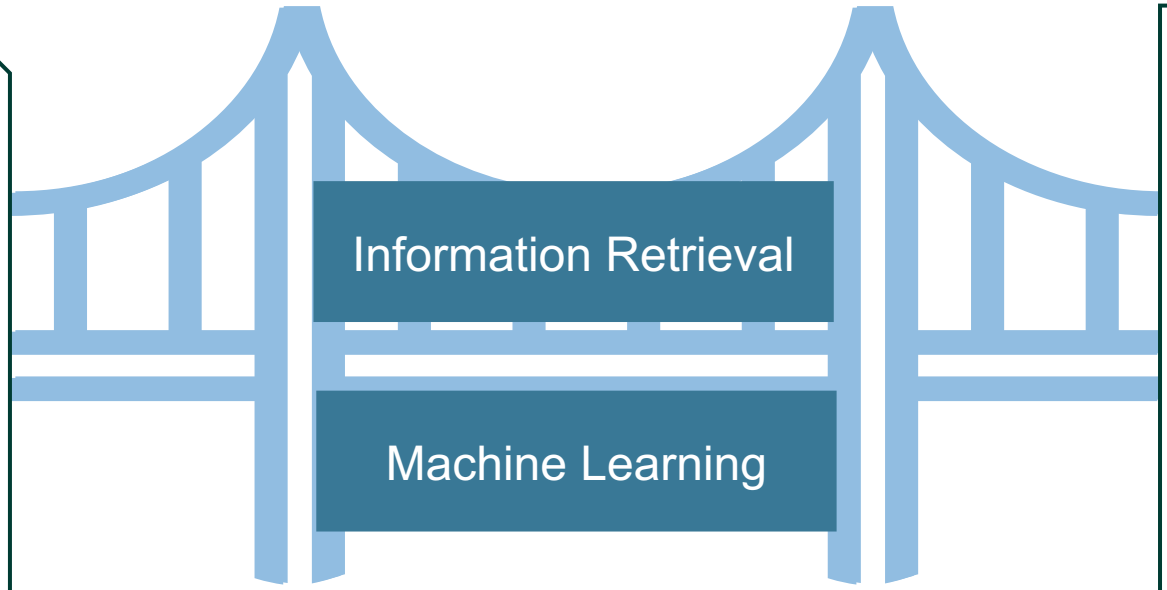
Backup

Traceability Link Recovery between Documentation & Code

Software Architecture Documentation (SAD)

The controller receives incoming requests and verifies them.

Then, it answers requests by querying the persistence component.



Problem: The results of direct TLR approaches are not good enough.

Code

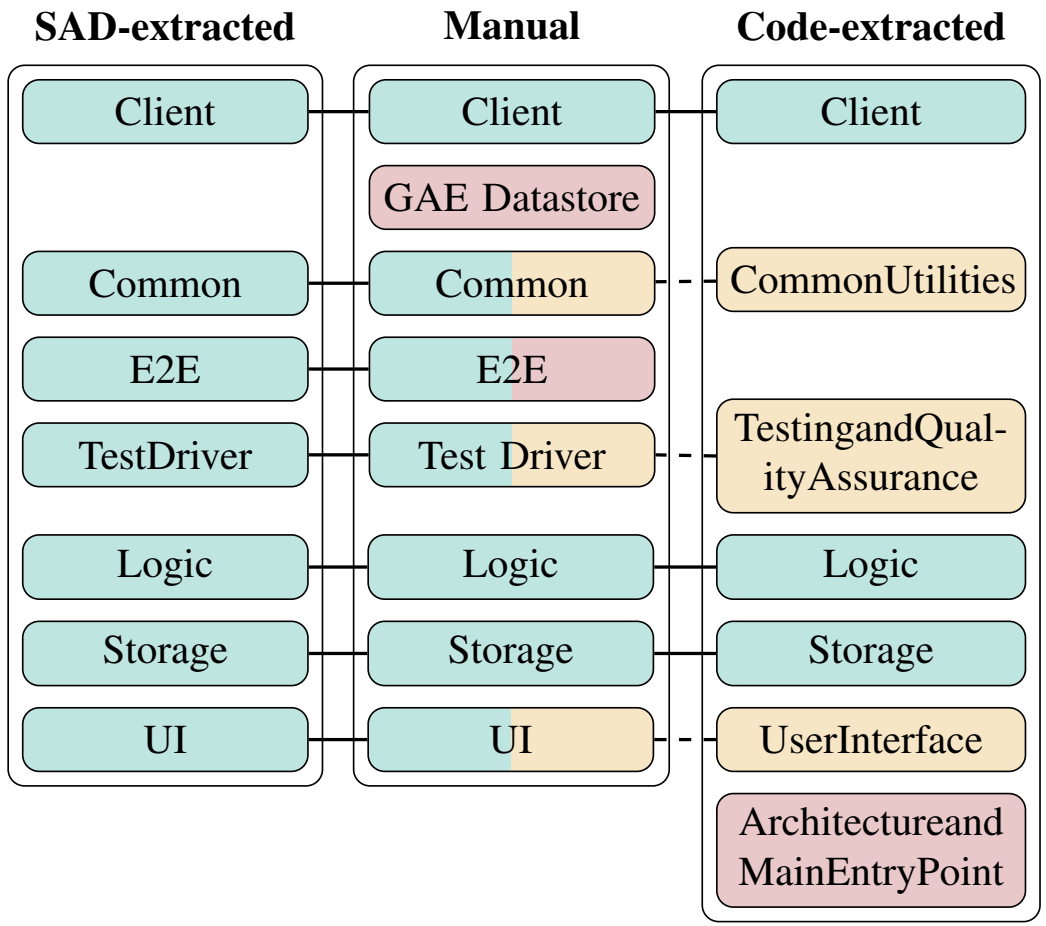
```
package service
class Controller {
   …
}
package dataaccess
class Products {
   …
}
class Users {
   …
}
```

Eval: Problems with LLM-extracted Simple Models

Example:

- GPT-4 Turbo
- TEAMMATES

- F1-Score
  - Documentation: 0.80
  - Code: 0.34



RQ 1: Comparable to TransArC ?

RQ 2: Better than SotA w/o SAMs?

RQ 3: Open-Source vs. Closed-Source LLMs?

RQ 4: Influence of different artifacts (i.e., Code/SAD)

Eval: Setup

Artifact Type		MediaStore	TeaStore	TEAMMATES	BigBlueButton (BBB)	JabRef
SAD	#Sentences	37	43	198	85	13
SAM	#Model Elements	23	19	16	24	6
Code	#Files	97	205	832	547	1,979
SAD-Code	#Trace Links	50	707	7,610	1,295	8,240

Eval: Component Names derived from Documentation (F1)

	Approach	MediaStore	TeaStore	TEAMMATES	BBB	Jabref	Avg.	w. Avg.
	CodeBERT (LLM)	.17	.36	.12	.12	.61	.28	.36
	ArDoCode	.09	.31	.53	.13	.80	.37	.62
	RAG (GPT-4o mini)	.08	.38	.06	.24	.05	.16	.08
	TransArC	**.68**	**.83**	**.80**	**.84**	**.94**	**.82**	**.87**
Ours	GPT-4o mini	.50	.78	.80	.68	.94	.74	.85
Ours	**GPT-4o**	**.50**	**.79**	**.80**	**.75**	**.94**	**.76**	**.86**
Ours	Codellama 13b	.63	.79	.56	.08	.94	.60	.71
Ours	Llama 3.1 70b	.49	.70	.41	.51	.94	.61	.68

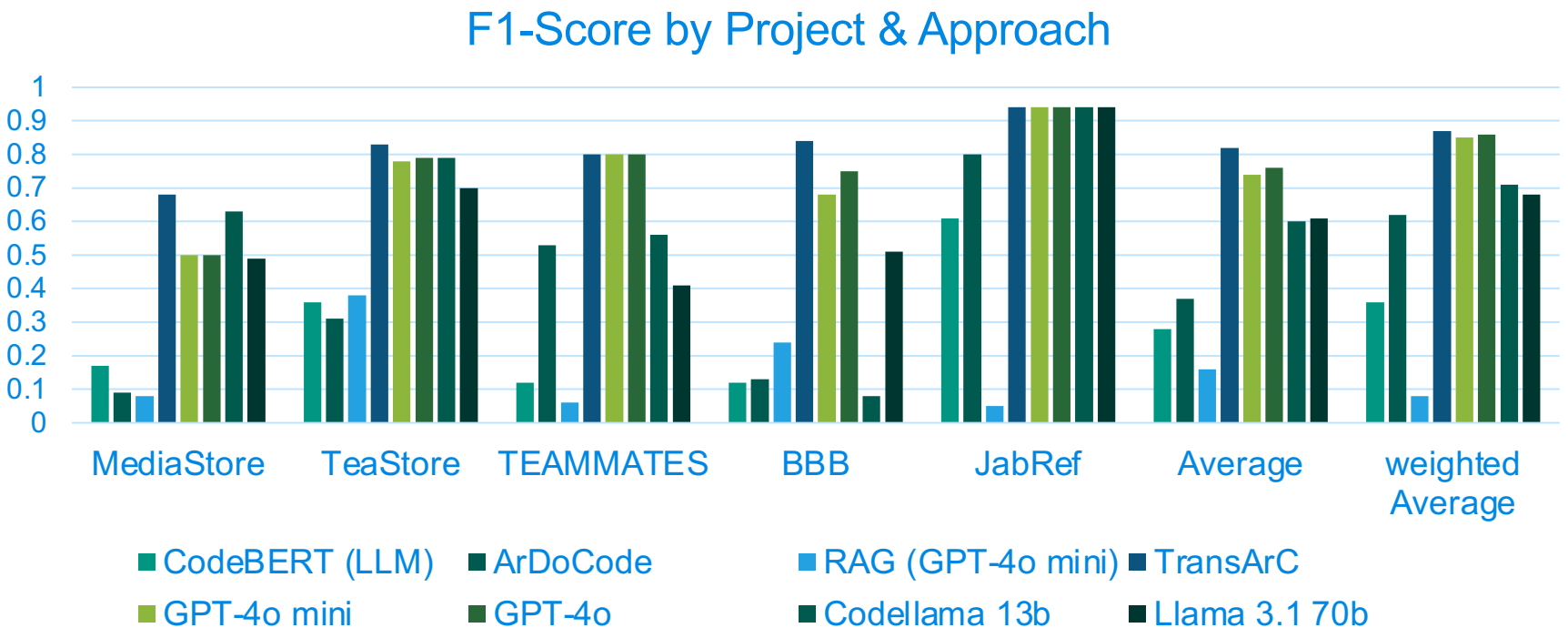
RQ 1: Comparable to TransArC ?

RQ 2: Better than SotA w/o SAMs?

RQ 3: Open-Source vs. Closed-Source LLMs?

F1-Score: Harmonic Mean of Precision and Recall

Eval: Component Names derived from Documentation (F1)



RQ 1: Comparable to TransArC ?

RQ 2: Better than SotA w/o SAMs?

RQ 3: Open-Source vs. Closed-Source LLMs?

F1-Score: Harmonic Mean of Precision and Recall

Eval: Problems with LLM-extracted Simple Models

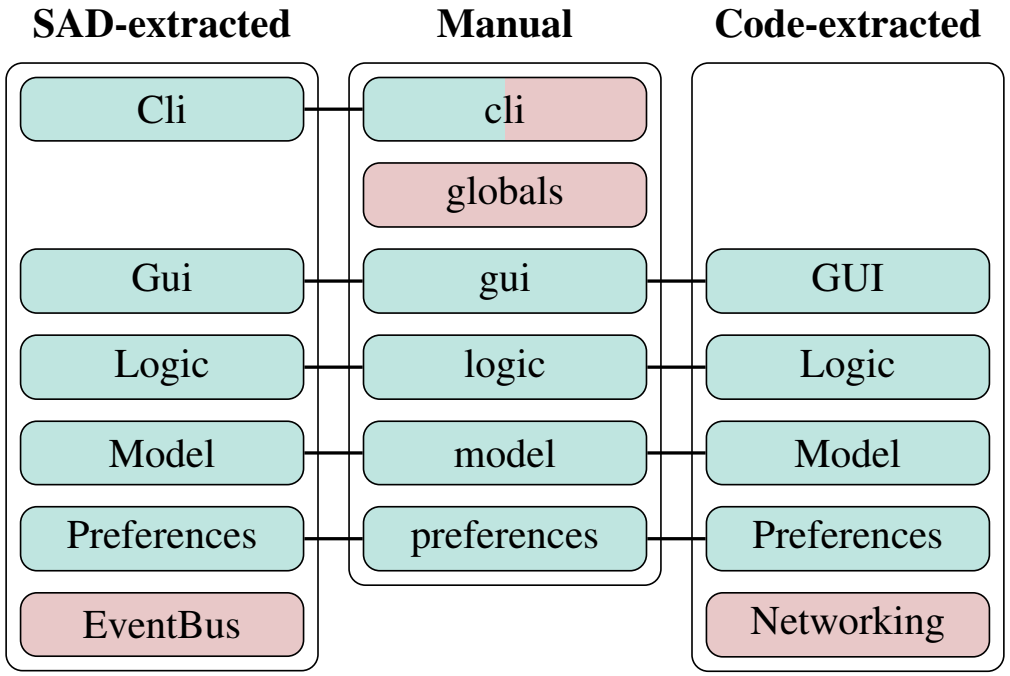
RQ 4: Influence of different artifacts (i.e., Code/SAD)

Example:

- Llama 3.1 70b
- JabRef

- Code: only “components” shown; “sub-components” didn’t affected the TLR task

- F1 score
  - Documentation: 0.94
  - Code: 0.94



TransArC via LLM-extracted Component Names

Prompt: Documentation to Architecture

Your task is to identify the **high-level components** based on the software architecture documentation.

In a first step, you shall elaborate on the following documentation:
{Software Architecture Documentation}

Prompt: Code to Architecture

You get the **{Features}** of a software project.

Your task is to summarize the {Features} w.r.t. the **high-level architecture** of the system.

Try to identify **possible components.** {Features}: {Content}