CAPRA: Scaling Feedback on Software Architecture Deliverables with a Multi-Agent LLM System

Marco Becattini
University of Florence

Niccolò Caselli
University of Florence

Matteo Minin
University of Florence

Roberto Verdecchia
University of Florence

Enrico Vicario
University of Florence

Paper:

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

{ST LAB} Software Technologies Laboratory

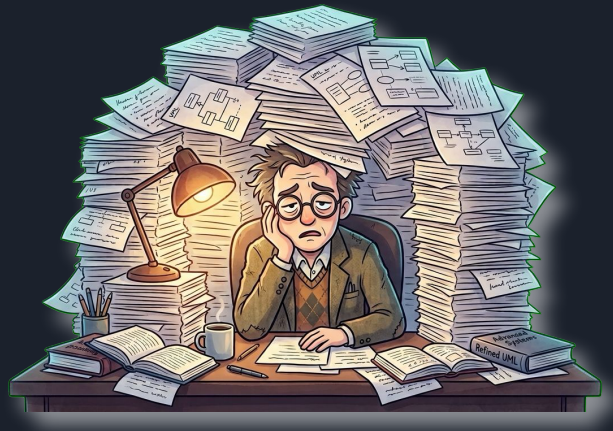
CSEE&T 2026

CAPRA: Configurable Architecture Proficiency Report Assessment

Paper:

Why we created CAPRA?

- As SE educators, we believe ***hands-on project-based learning is a quintessential pedagogical tool***
- **Reviewing software architectural artifacts**, where heterogeneous quantitative and qualitative aspects mix, can be **highly time-consuming**
- As **class sizes grow**, thorough manual review becomes a **bottleneck limiting the frequency & quality** of **formative feedback**
- ***CAPRA** aims to **support educators** **to** **efficiently and effectively** provide **tailored feedback on software architecture deliverables***



How does CAPRA work?

Stage 1

Document Parsing

Stage 2

Document Verification

Stage 3

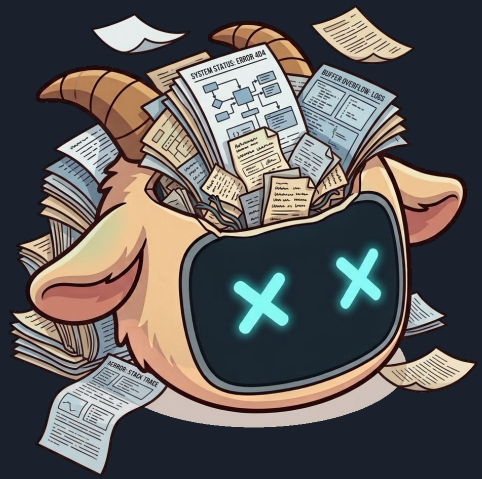
Evidence Anchoring

Stage 4

Feedback Generation

CAPRA's Configuration

Stage 0

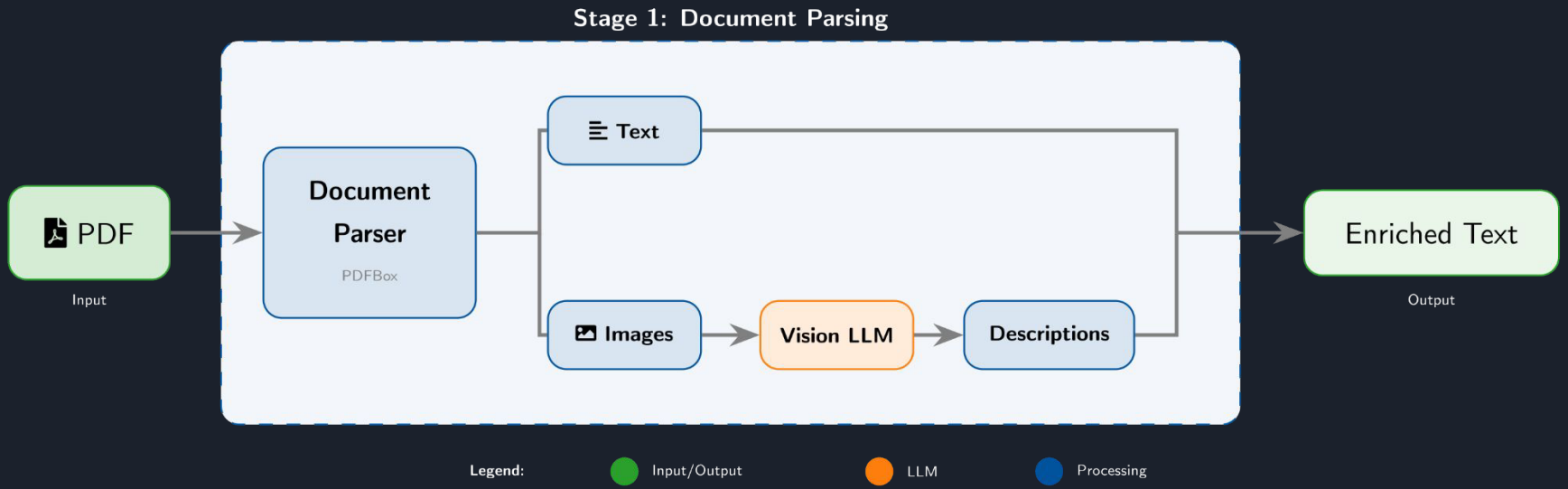


CAPRA is an **empty box**; it doesn't know what to teach his students

Filled with the best reports from the course and **tailored** by teacher's rules, CAPRA knows how to help every student.

Document Parsing

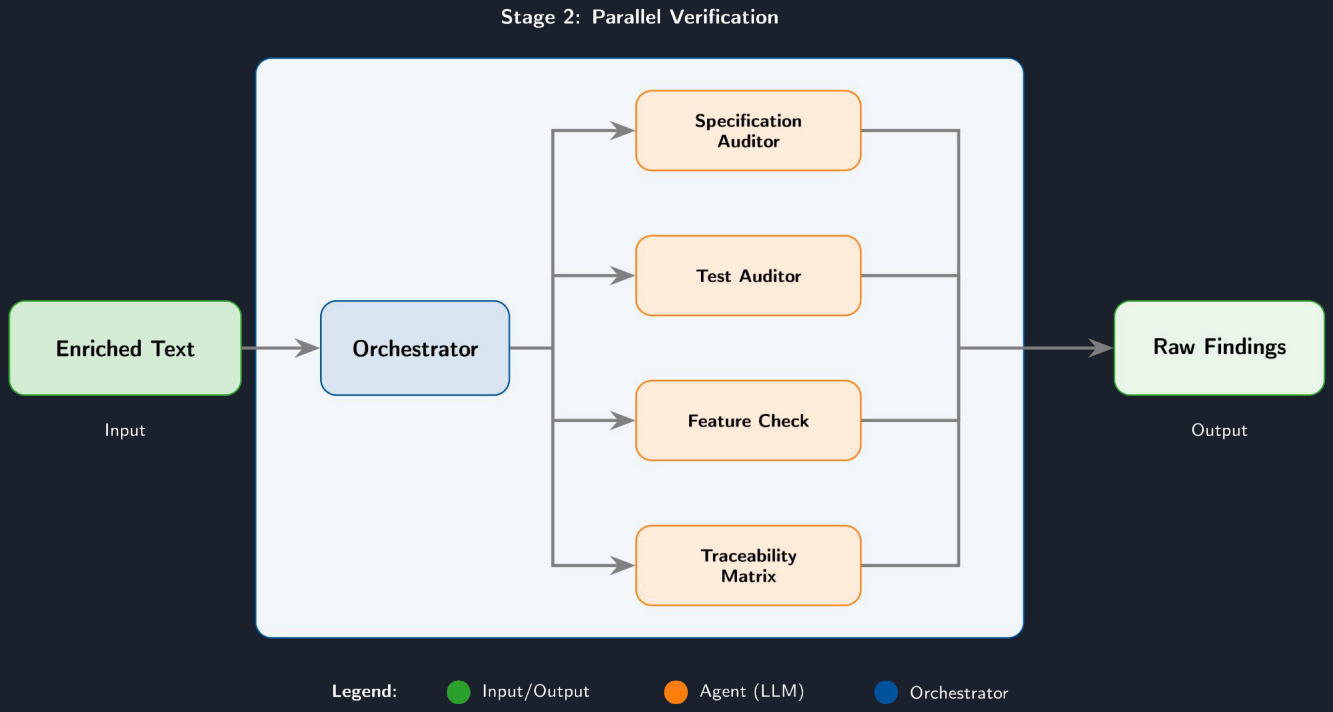
Stage 1



Document parsing pipeline: takes a PDF document and enriches it with precise description of every image

Document Verification

Stage 2



Multi-Agent stage with 4 specialized Agents

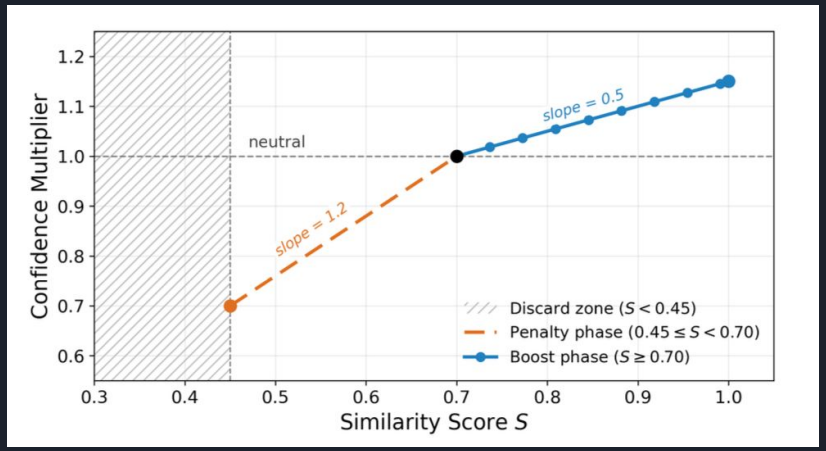
Evidence Anchoring

Stage 3

Every CAPRA's finding is anchored to a document's quote and has an associated confidence

Is the cited quote anchored in the document?

Long quote (> 15 characters)



Short quote

Findings with less than 15 characters are penalized

*(confidence * 0.7)*

No quote

Findings without quotes are ignored

(confidence = 0)

Feedback Generation

Stage 4

LLM generated narrative summaries

+

Deterministically inserted findings

=

Empirical Evaluation: Research Questions

RQ1	How valid, reliable, and constructive is the feedback generated by CAPRA?
RQ2	How does CAPRA's processing time compare to manual software architecture review?
RQ3	How accurately does CAPRA identify Knowledge Base features in student deliverables?

CAPRA Empirical Evaluation: How We Answered Our Research Questions

- **2 researchers** independently analyzed a sample of **10 deliverables**, scoresing them on a custom made evaluation rubric
- Sampled deliverables from Bachelor's Software Engineering course held by E. Vicario at UniFI
- Agreement measured with Cohen's Kappa alongside raw agreement

Dimension	Criteria	Focus
A - Extraction Completeness	4	Requirements, use cases, patterns, tests
B - Feature Validation	1	Knowledge Base feature presence / absence
C - Issues & Recommendations	2	Grounded issues, actionable recommendations
D - Template & Tone	1	Expected sections presence and formal tone

CAPRA Empirical Evaluation: Results

88.8%
evaluated criteria passed under the strict aggregation rule

0.582
agreement score between the two reviewers (moderate)

~4 min
per report. Manual review takes 30 to 45 minutes

7 to 11x
faster than manual review, at $0.44 per report

Key takeaway:

- CAPRA performs reliably on tasks with an **objective answer** (e.g., extracting requirements or verifying required features).
- It performs less reliably on judging whether a flagged issue is significant, and clearly explained. Human reviewers themselves showed only moderate agreement on this judgment, so **it remains a task for human oversight!**

CAPRA Conclusion and Future Work

CAPRA is a first-pass teaching assistant, *not* an autonomous grader.

What CAPRA gives teachers

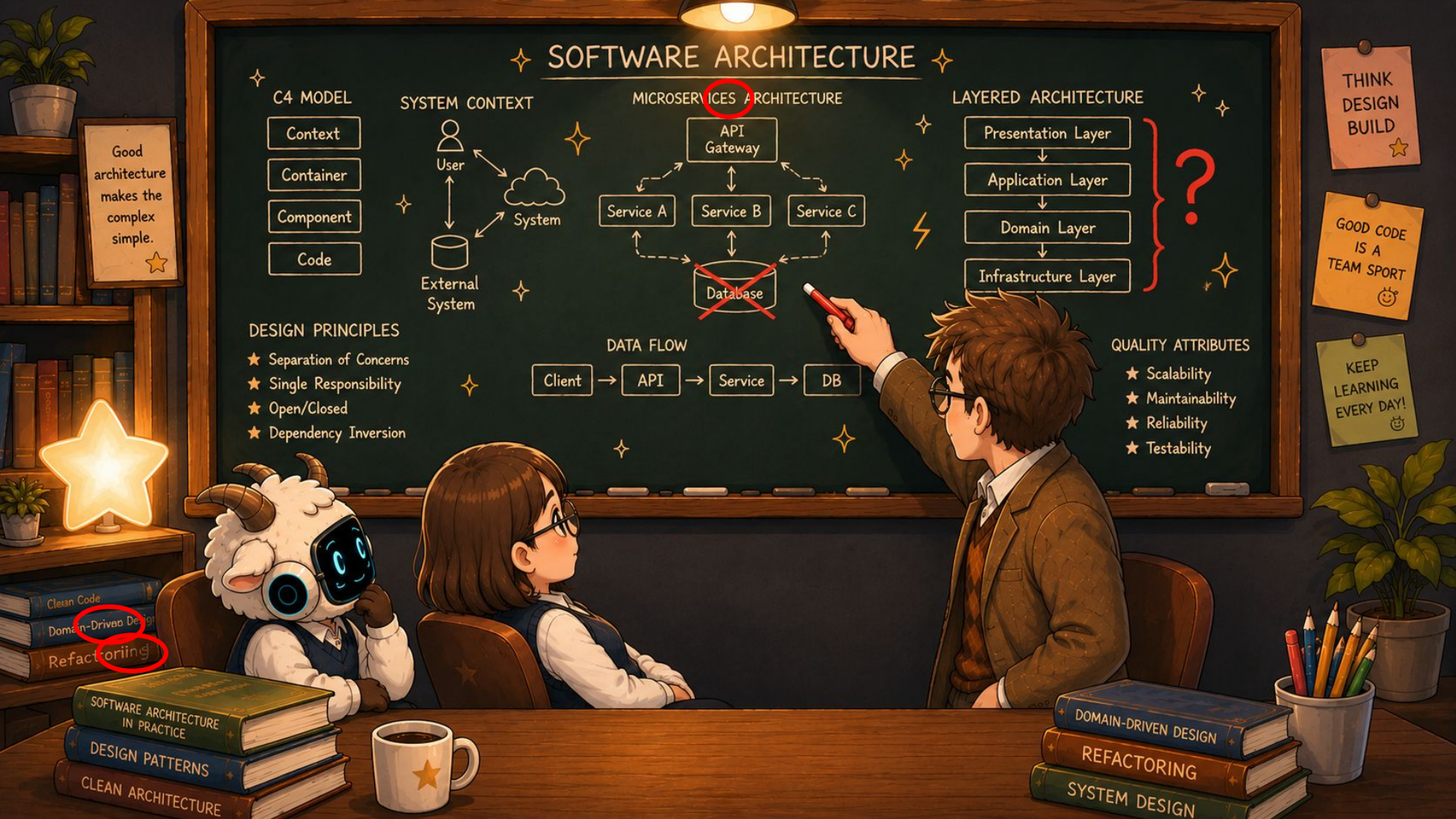
- Rapid formative feedback cycles
- Earlier feedback student can act on
- More time for nuanced architectural judgment

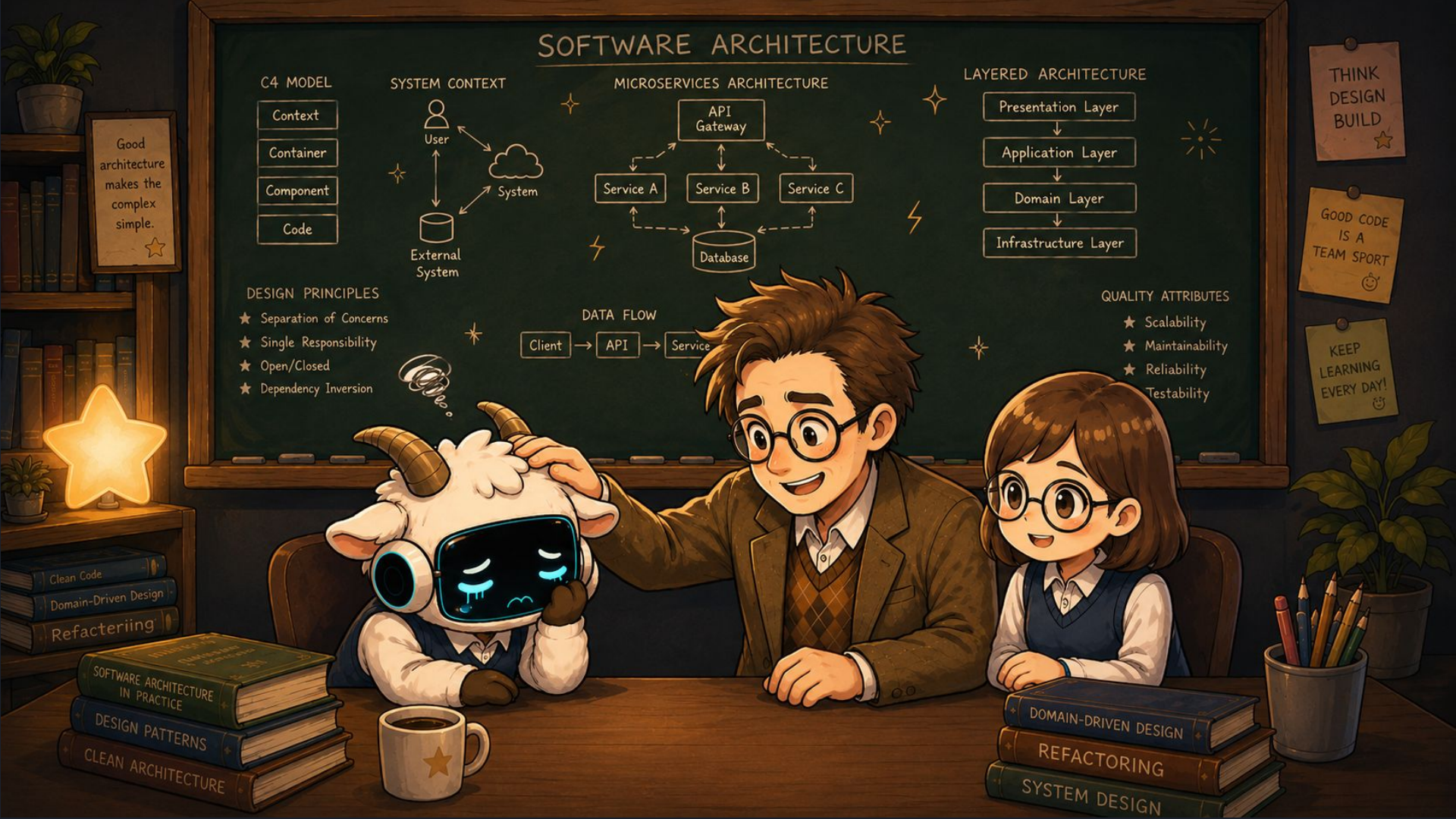
Future work

- Evaluate across more courses and institutions
- Include weaker submissions
- Test open-source LLMs and anchoring thresholds
- Evaluate student impact

Responsible deployment means keeping the teacher in the loop!


Final word: AI is just a tool.
Its misuse in education context is a dangerous practice
SOFTWARE ARCHITECTURE
Context
Container
Component
Code
External System
API Gateway
Service A
Service B
Service C
Database
LAYERED ARCHITECTURE
Application Layer
Domain Layer
Infrastructure Layer
DESIGN PRINCIPLES
Separation of Concerns
Single Responsibility
Open/Closed
Dependency Inversion
DATA FLOW
Client
API
Service
DB
QUALITY ATTRIBUTES
Scalability
Maintainability
Reliability
Testability
Good architecture makes the complex simple.
THINK DESIGN BUILD
GOOD CODE IS A TEAM SPORT
KEEP LEARNING EVERY DAY!
Clean Code
Domain-Driven Design
Refactoring
SOFTWARE ARCHITECTURE IN PRACTICE
DESIGN PATTERNS
CLEAN ARCHITECTURE
DOMAIN-DRIVEN DESIGN
REFACTORING
SYSTEM DESIGN

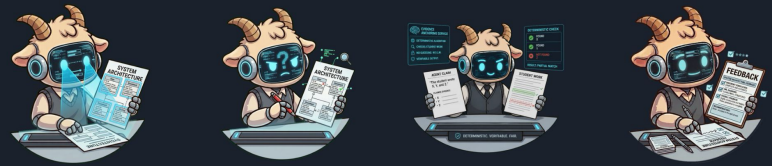

SOFTWARE ARCHITECTURE
C4 MODEL
Context
Container
Component
Code
SYSTEM CONTEXT
User
System
External System
MICROSERVICES ARCHITECTURE
API Gateway
Service A
Service B
Service C
Database
LAYERED ARCHITECTURE
Presentation Layer
Application Layer
Domain Layer
Infrastructure Layer
?
DESIGN PRINCIPLES
Separation of Concerns
Single Responsibility
Open/Closed
Dependency Inversion
DATA FLOW
Client
API
Service
DB
QUALITY ATTRIBUTES
Scalability
Maintainability
Reliability
Testability
Good architecture makes the complex simple.
THINK DESIGN BUILD
GOOD CODE IS A TEAM SPORT
KEEP LEARNING EVERY DAY!
Clean Code
SOFTWARE ARCHITECTURE IN PRACTICE
DESIGN PATTERNS
CLEAN ARCHITECTURE
DOMAIN-DRIVEN DESIGN
REFACTORING
SYSTEM DESIGN


SOFTWARE ARCHITECTURE
C4 MODEL
Context
Container
Component
Code
SYSTEM CONTEXT
User
System
External System
MICROSERVICES ARCHITECTURE
API Gateway
Service A
Service B
Service C
Database
LAYERED ARCHITECTURE
Presentation Layer
Application Layer
Domain Layer
Infrastructure Layer
DESIGN PRINCIPLES
Separation of Concerns
Single Responsibility
Open/Closed
Dependency Inversion
DATA FLOW
Client
API
Service
QUALITY ATTRIBUTES
Scalability
Maintainability
Reliability
Testability
Good architecture makes the complex simple.
THINK DESIGN BUILD
GOOD CODE IS A TEAM SPORT
KEEP LEARNING EVERY DAY!
Clean Code
Domain-Driven Design
Refactering
SOFTWARE ARCHITECTURE IN PRACTICE
DESIGN PATTERNS
CLEAN ARCHITECTURE
DOMAIN-DRIVEN DESIGN
REFACTORING
SYSTEM DESIGN

Paper:

CAPRA: Configurable Architecture Proficiency Report Assessment

How does CAPRA work?



CAPRA Empirical Evaluation: Results

Evaluation on 10 student reports, 2 independent reviewers

88.8%	0.582	~4 min	7 to 11x
of feedback items judged correct by human reviewers	agreement score between the two reviewers (moderate)	per report. Manual review takes 30 to 45 minutes	faster than manual review, at $0.44 per report

Key takeaway:

- CAPRA performs reliably on tasks with an **objective answer** (e.g., extracting requirements or verifying required features).
- It performs less reliably on judging whether a flagged issue is significant, and clearly explained. Human reviewers themselves showed only moderate agreement on this judgment, so it remains a task for human oversight.

Final word: AI is just a tool. Its misuse in education context is a dangerous practice