GenAI (….LLMs) Everywhere…. - AI Race!

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

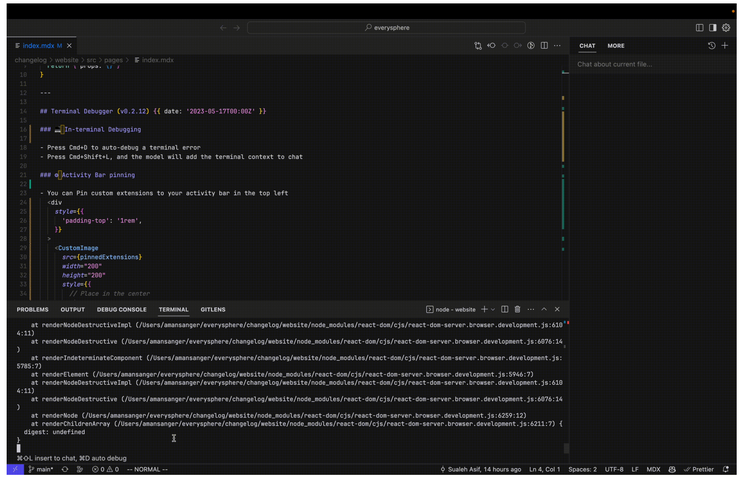
Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

Introducing the SWE-Lancer benchmark

Can frontier LLMs earn $1 million from real-world freelance software engineering?

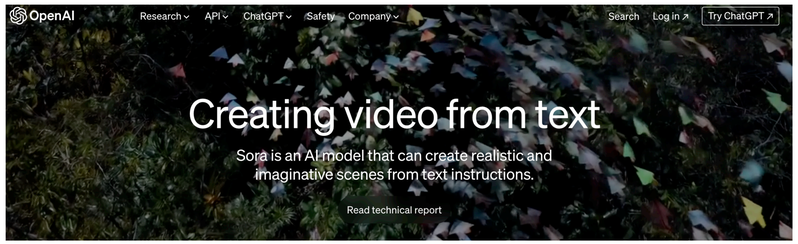
Read paper ↗ Access repository ↗



Home › Blog ›

Accelerating scientific breakthroughs with an AI co-scientist

February 19, 2025 · Juraj Gottweis, Google Fellow, and Vivek Natarajan, Research Lead



Even the car that dropped me ;)

Source: openai.com, meta.com, cursor.com

Navigating the Crossroads of GenAI and SE: Insights from our Research

Karthik Vaidhyanathan

European Data Intensive Software Systems Winter School 2025

Feb 24, 2025, L’Aquila

ABOUT ME

Logic takes you from A to B, Immagination takes you elsewhere -- Albert Einstein

Karthik Vaidhyanathan

Assistant Professor

Software Engineering Research Center and

Leadership Member, Smart City Research Center

IIIT Hyderabad, India

Research Interests

AI4SA

1. AI for Architectural Knowledge
2. AI for self-adaptation

SA4AI

1. Sustainable AI-enabled systems
2. Self-adaptive AI Systems (Edge-Cloud)

Education

Double Master Degree - Software Architecture and Machine Learning

PhD from GSSI, Italy

Postdoc, University of L'Aquila, Italy

Fun Facts!

1. Cricket fanatic!
2. Movie buff!!
3. From God's own Country!!

https://karthikvaidhyanathan.com /in/karthikv1392/ karthi_ishere karthik.vaidhyanathan@iiit.ac.in

Quest for Artificial Intelligence

Next word prediction problem

Google

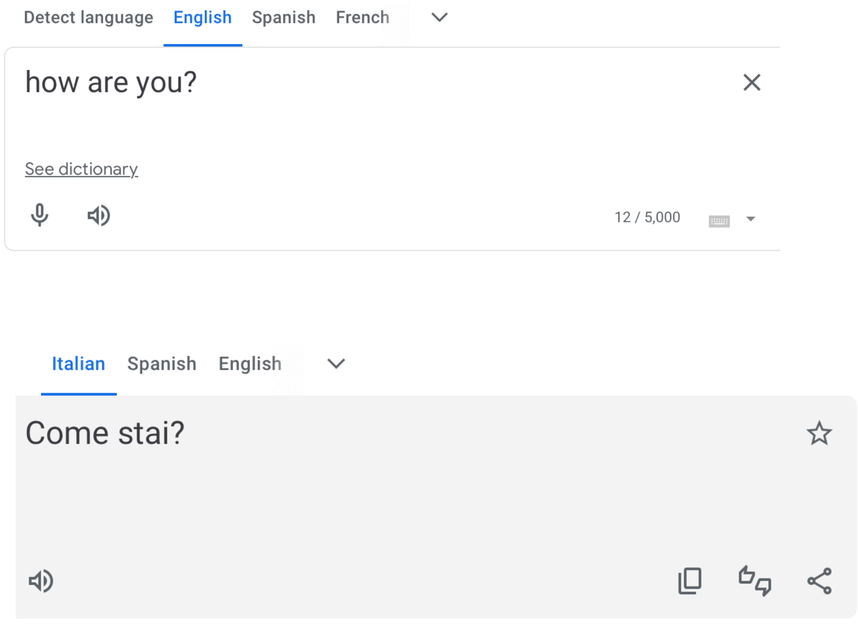
What is software

- what is software
- what is software **engineering**
- what is software **testing**
- what is software **development**
- what is software **and hardware**
- what is software **as a service**
- what is software **piracy**
- what is software **architecture**

Google

software engineering is

- software engineering is **dead**
- software engineering is **a layered technology**
- software engineering is **easy or hard**
- software engineering is **hard**
- software engineering is **what**
- software engineering is **primarily concerned with**

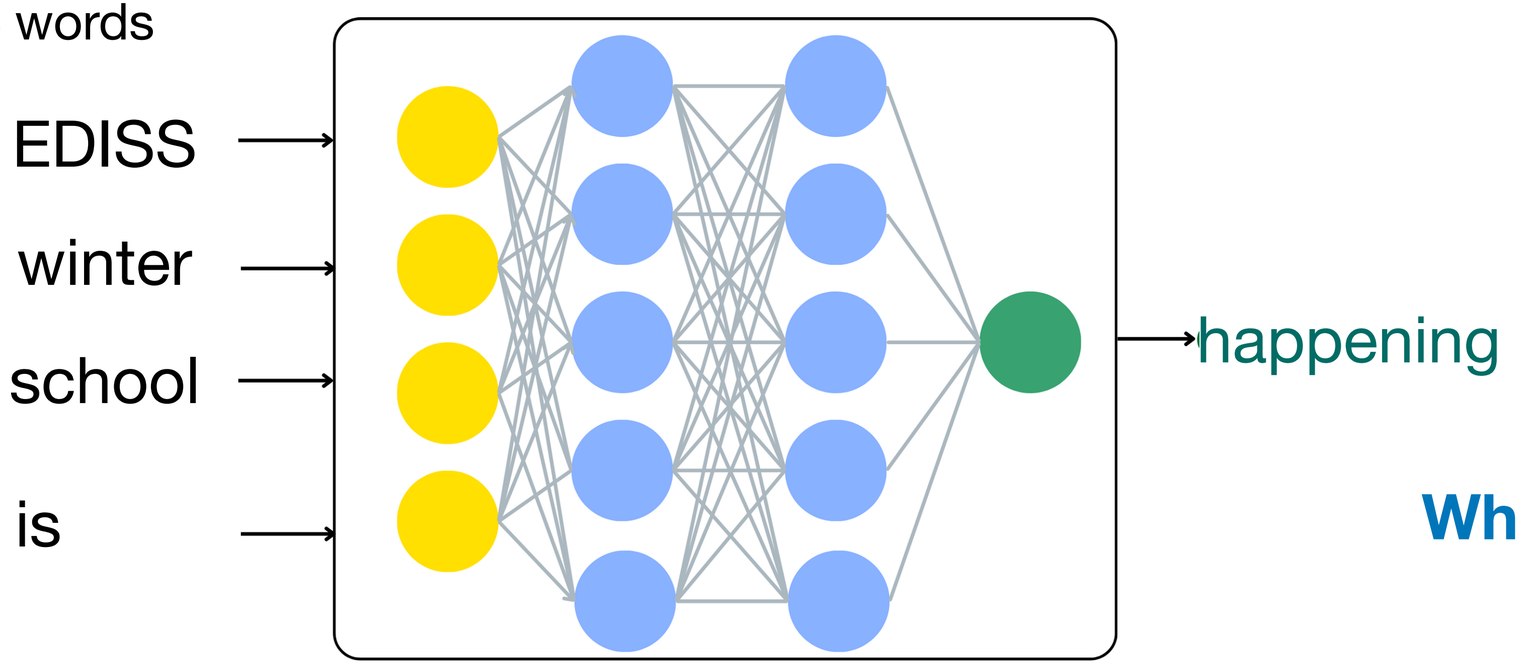


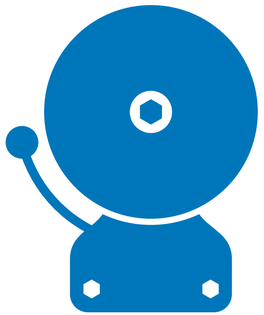
Problem since 1950's!!!

Its been a long journey MCP, ..backprop…RNN CNN,……..

Predict the next word in a sequence!

Eg: Context of 4 words





What does “it” refer to?

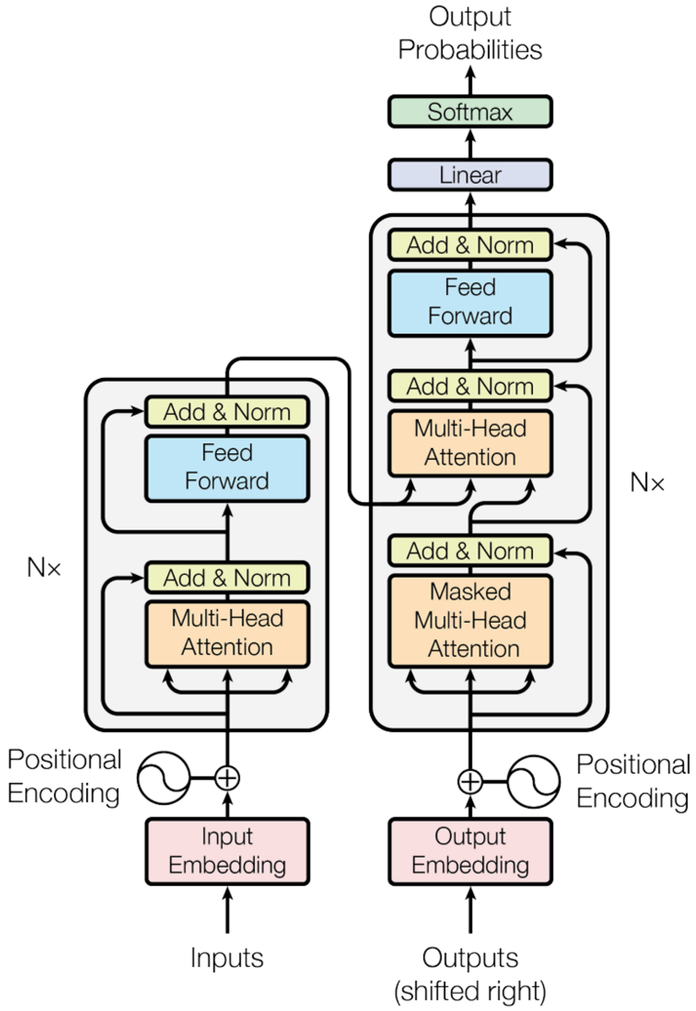
EDISS winter school is happening in

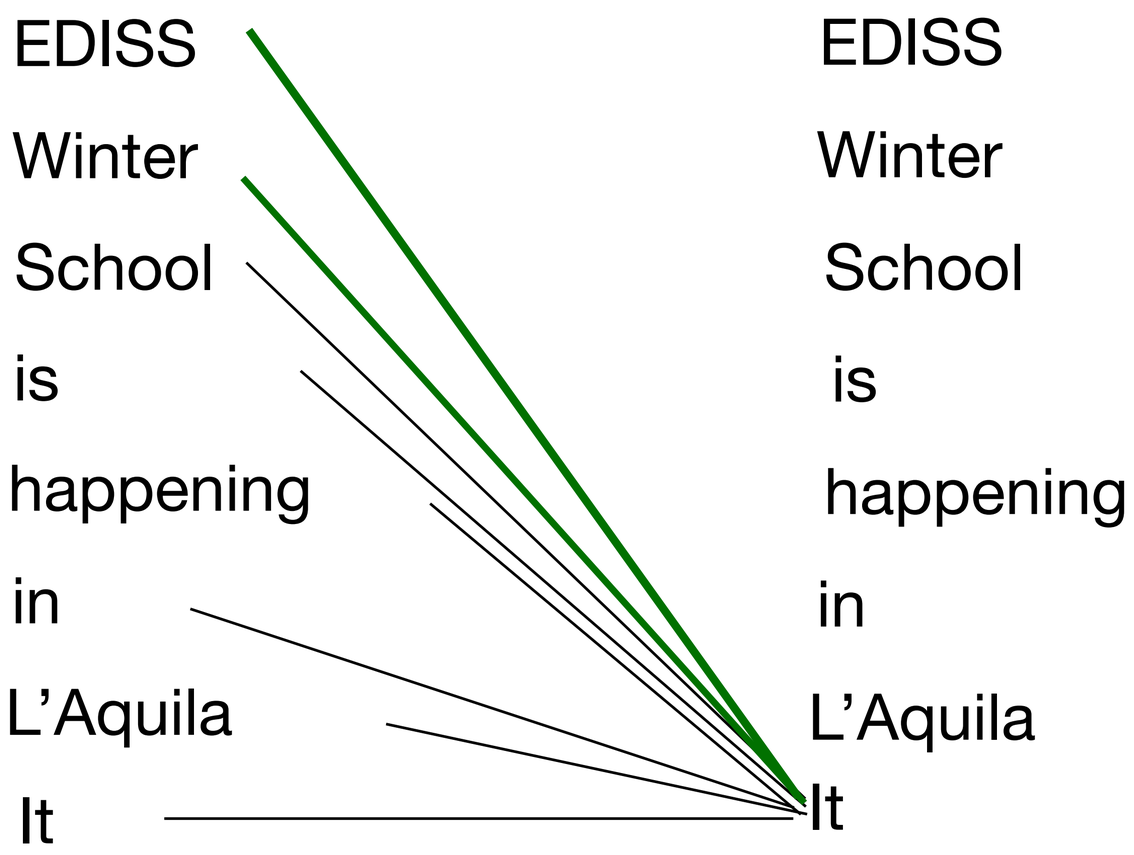
EDISS winter school is happening in L’Aquila.

EDISS winter school is happening in L’Aquila. It

Attention is all you need!

Age of Transformers



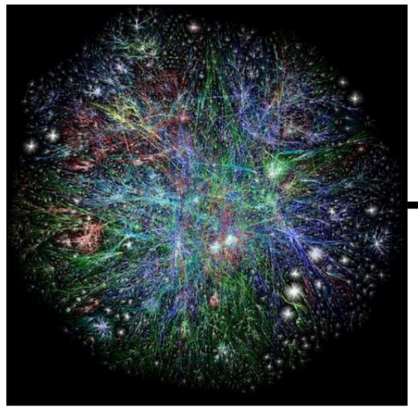


Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. **Attention is all you need.** *Advances in neural information processing systems*, *30*.

“Large” Language Models (LLM)

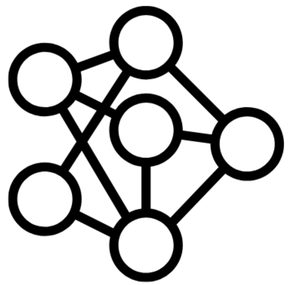
Do you have a ton of text and compute power?

Internet



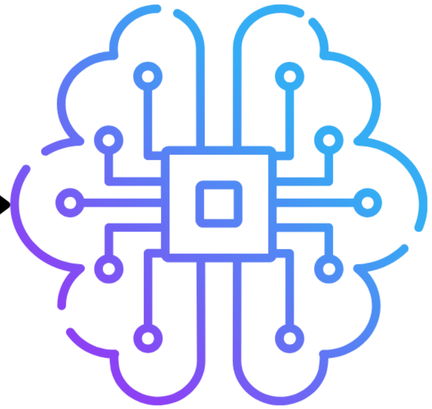
~10B of text from internet

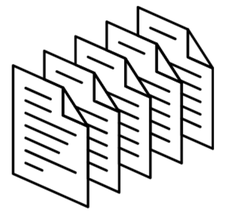
Compute



Transformer NN
trained on 1000s of GPUs
for days

Foundation model/
Base model



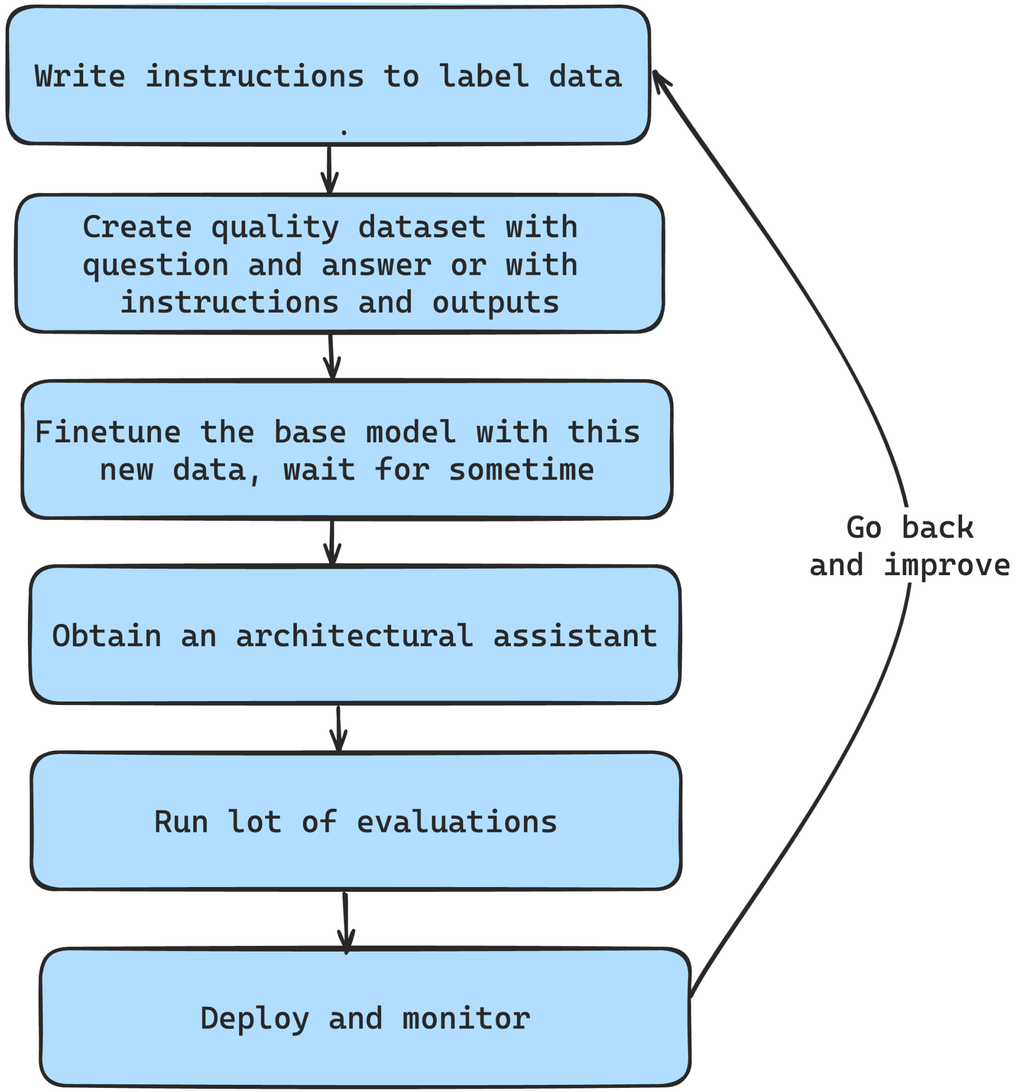


Model parameters
~some billions
100s of GB

Inspiration from slides of Andrej Karpathy, Introduction to large language models

I want something more specific to Architecture

May be you can fine-tune and create your model



<<USER>

What is Software Architecture?

<<ASSISTANT>>

Software architecture is a combination of elements, form and rationale

<<USER>

How to start thinking about architecture?

<<ASSISTANT>>

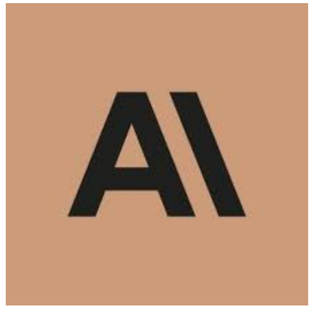
One can start with IEEE 42010 framework, understand stakeholders...

Some 1000s of examples

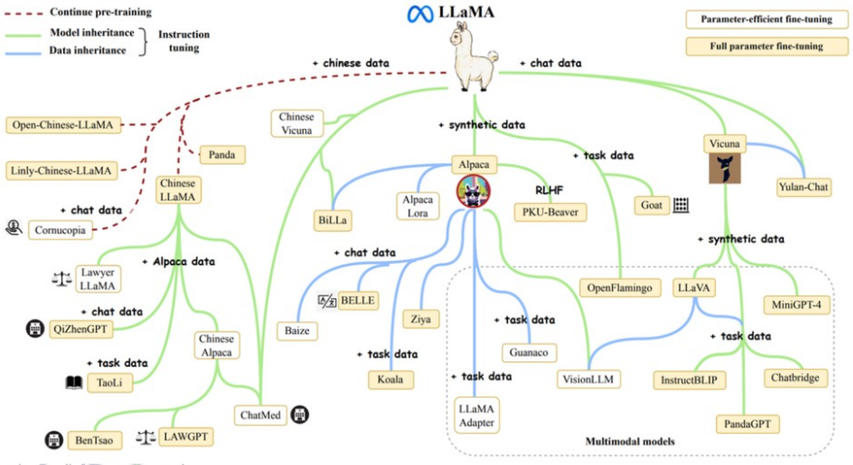
Eg: codexGPT, CodeBERT,…

Today we have different varieties of LLMs

Proprietary



Open Source



Arena (battle) | Arena (side-by-side) | Direct Chat | Vision Direct Chat | Leaderboard | About Us

LMSYS Chatbot Arena Leaderboard

| Vote | Blog | GitHub | Paper | Dataset | Twitter | Discord |

LMSYS Chatbot Arena is a crowdsourced open platform for LLM evals. We've collected over **500,000** human preference votes to rank LLMs with the Elo ranking system.

Arena Elo | Full Leaderboard

Total #models: **81**. Total #votes: **634676**. Last updated: April 9, 2024.

Contribute your vote at chat.lmsys.org! Find more analysis in the notebook.

Rank	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	Claude 3 Opus	1256	+3/-4	47589	Anthropic	Proprietary	2023/8
1	GPT-4-1106-preview	1254	+3/-4	62657	OpenAI	Proprietary	2023/4
1	GPT-4-0125-preview	1250	+3/-3	47631	OpenAI	Proprietary	2023/12
4	Bard (Gemini Pro)	1208	+5/-5	12468	Google	Proprietary	Online
4	Claude 3 Sonnet	1204	+3/-3	57740	Anthropic	Proprietary	2023/8
6	Command R+	1194	+5/-5	17404	Cohere	CC-BY-NC-4.0	2024/3
6	GPT-4-0314	1189	+4/-3	41292	OpenAI	Proprietary	2021/9
8	Claude 3 Haiku	1182	+3/-4	50689	Anthropic	Proprietary	2023/8
9	GPT-4-0613	1164	+3/-3	60213	OpenAI	Proprietary	2021/9
9	Mistral-Large-2402	1158	+3/-4	35075	Mistral	Proprietary	Unknown
10	Qwen1.5-72B-Chat	1153	+4/-5	27050	Alibaba	Qianwen LICENSE	2024/2

https://chat.lmsys.org/

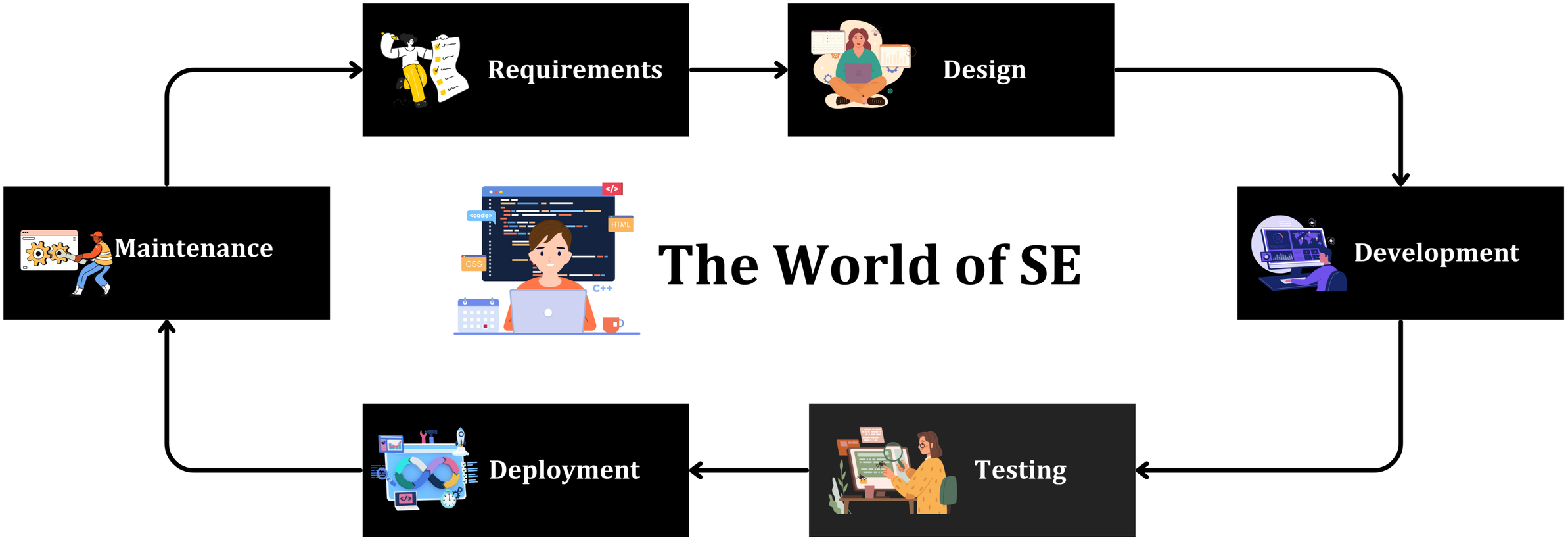
Bottomline: Text is an Abstraction of Reality!

The image shows a lively public square with a central fountain featuring a statue, surrounded by historic buildings, street lamps, and people walking, set against a bright blue sky with scattered clouds.

A sunny day in a wide, paved European square with a central fountain, surrounded by classic buildings and parked cars.

Source: Image of L'Aquila from Wikipedia

The World of Software Engineering

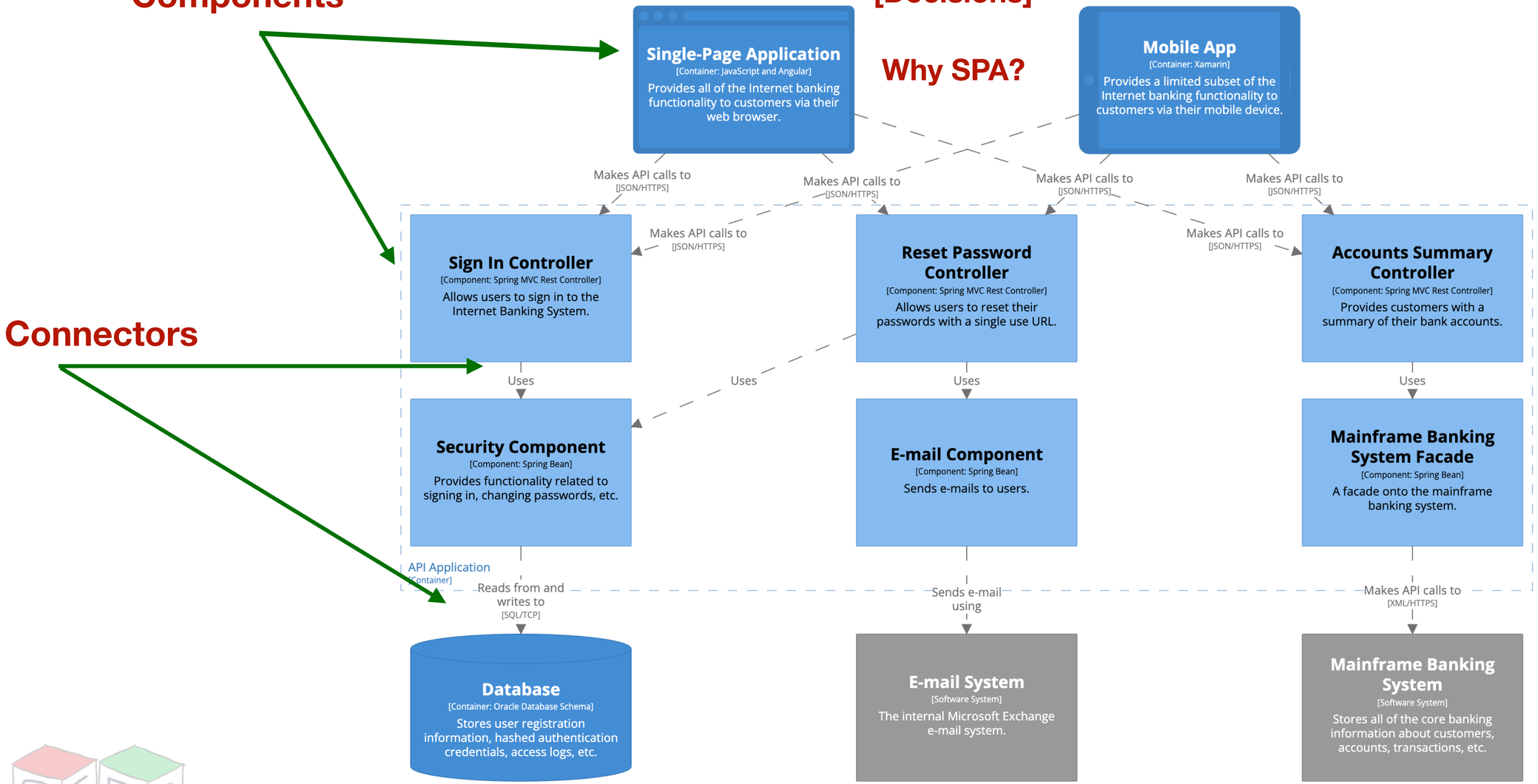


Starting with Design: Architecture an Abstraction

Components

[Decisions]

Why SPA?



[Decisions]

Why Spring?

Connectors

[Component] Internet Banking System - API Application

The component diagram for the API Application - diagram created with Structurizr.

Saturday, 11 November 2023 at 09:04 Greenwich Mean Time

Design Decisions is all you need!

Software Architecture as a Set of Architectural Design Decisions

Anton Jansen
Department of Computing Science
University of Groningen
PO BOX 800, 9700 AV, The Netherlands
anton@cs.rug.nl

Jan Bosch
Software & Application Technologies Lab
Nokia Research Center
PO BOX 407, FI-00045, Finland
jan.bosch@nokia.com

Abstract

Software architectures have high costs for change, are complex, and erode during evolution. We believe these problems are partially due to knowledge vaporization. Currently, almost all the knowledge and information about the design decisions the architecture is based on are implicitly embedded in the architecture, but lack a first-class repre-

this notion of architectural design decisions, although architectural design decisions play a crucial role in software architecture, e.g. during design, development, evolution, reuse and integration of software architectures. In design, the main concern is which design decision to make. In development, it is important to know which and why certain design decisions have been taken. Architecture evolution is about making new design decisions or removing obso-

Software Architecture is a set of key design decisions!!!

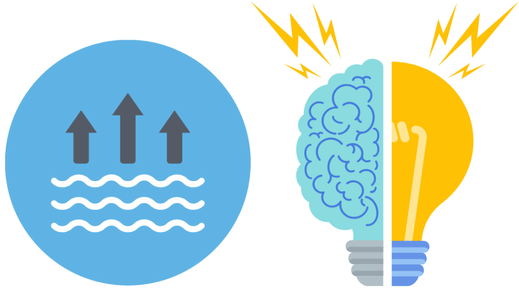
Jansen, A. and Bosch, J., 2005, November. Software architecture as a set of architectural design decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*

The Key Issue

Takes away the knowledge!

?

What happens when the dev/architect leaves?



Knowledge Vaporisation!

Architecture Knowledge Management

*Architecture knowledge management (AKM) aims to **codify and maintain** the Architectural knowledge of a software system in a form that can be **easily accessed** by different stakeholders*

ELSEVIER

Journal of Systems and Software

Volume 116, June 2016, Pages 191-205

10 years of software architecture knowledge management: Practice and future

Rafael Capilla [a], Anton Jansen [b], Antony Tang [c], Paris Avgeriou [d], Muhammad Ali Babar [e]

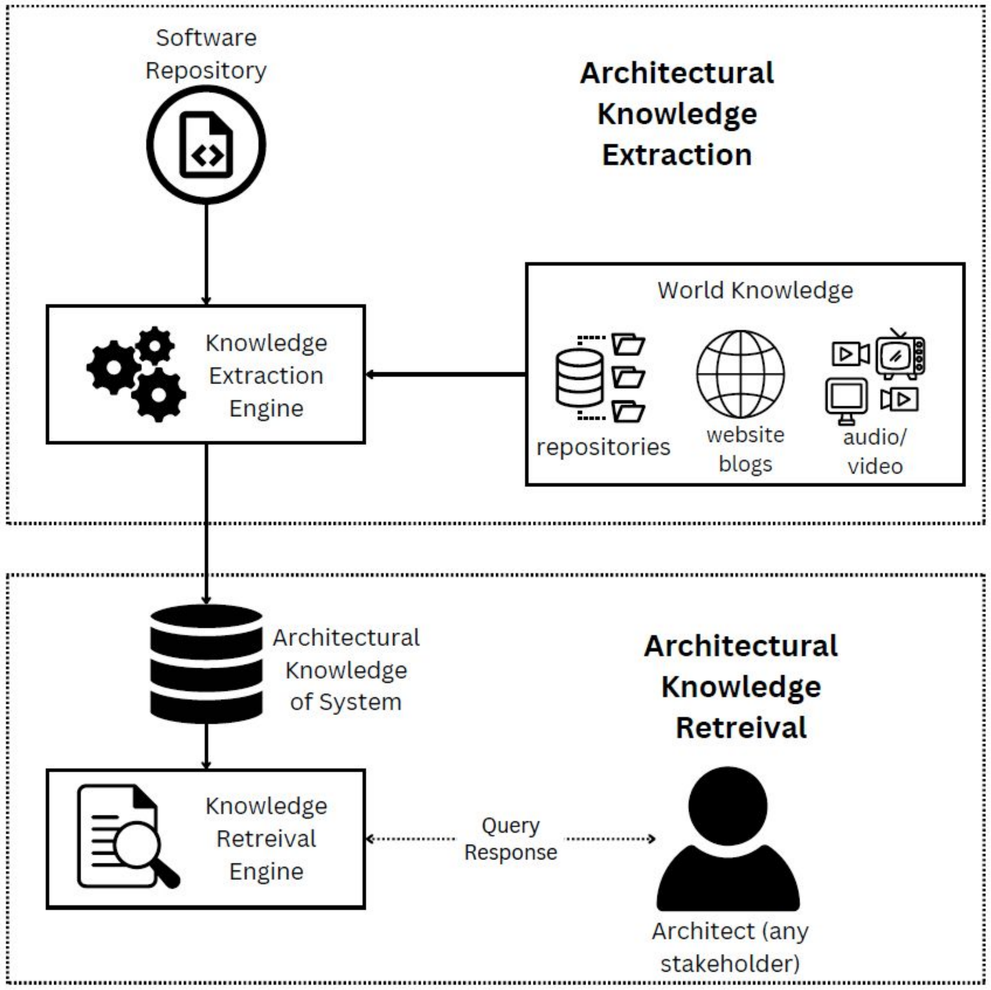
Show more

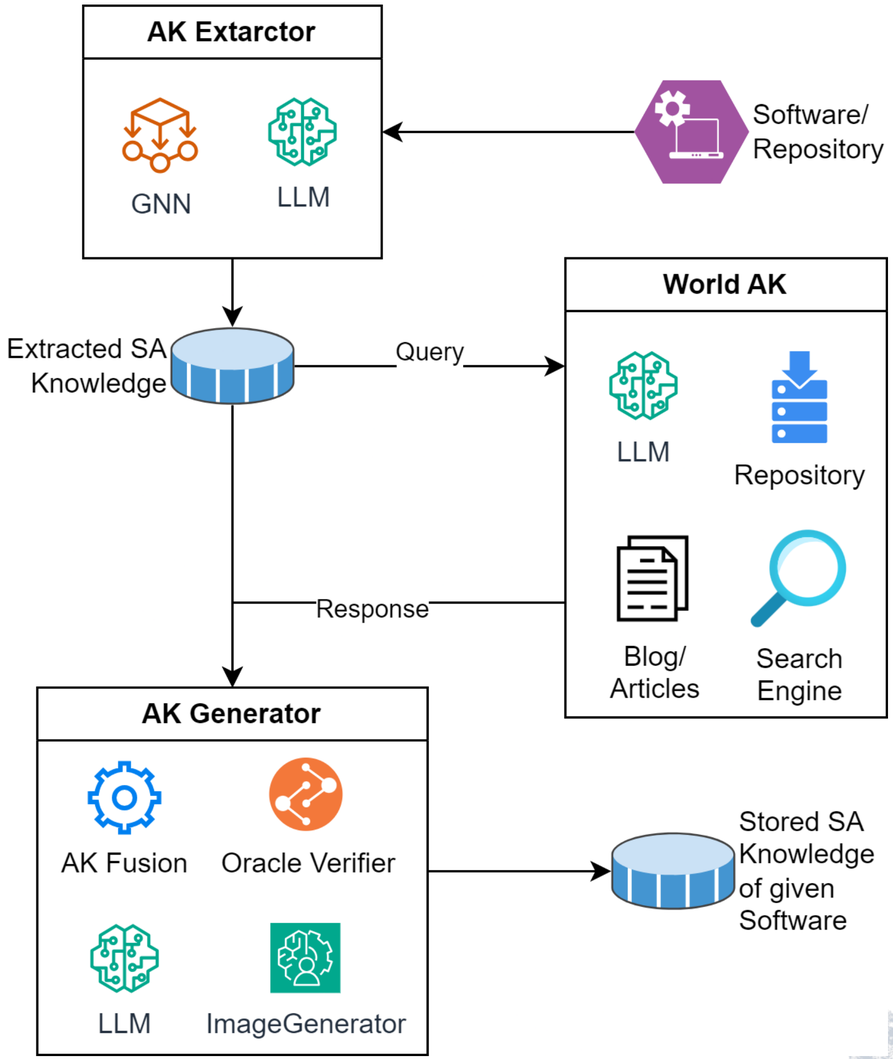
+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.jss.2015.08.054 Get rights and content

Need for better tools => Automate using **ArchBots or a co-pilot**

Generative AI for Architectural Knowledge Management





Rudra Dhar, Karthik Vaidhyanathan, Vasudeva Varma, **GenAI for Architecture Knowledge Management**, ICSA 2024, Poster

Starting with Decision Records

- **Architecture Decision Records: ADR**
- Lightweight mechanism for documenting decisions
- Design decisions require careful considerations of various parameters
  - This requires broader understanding of domain as well as expertise
  - **Can we use LLMs to generate architecture design decisions?**
  - **Can LLMs be used to extract architectural information from design decisions?**

Title: Deciding the technology for the data analysis component

Context
We need to decide whether to use Python as a programming language for our project. Our project involves data analysis, machine learning, and web development.

Decision
We have decided to use Python as our primary programming language for our project.

Rationale
1. Python has support for various ML and data analysis
2. Team members are already familiar with Python

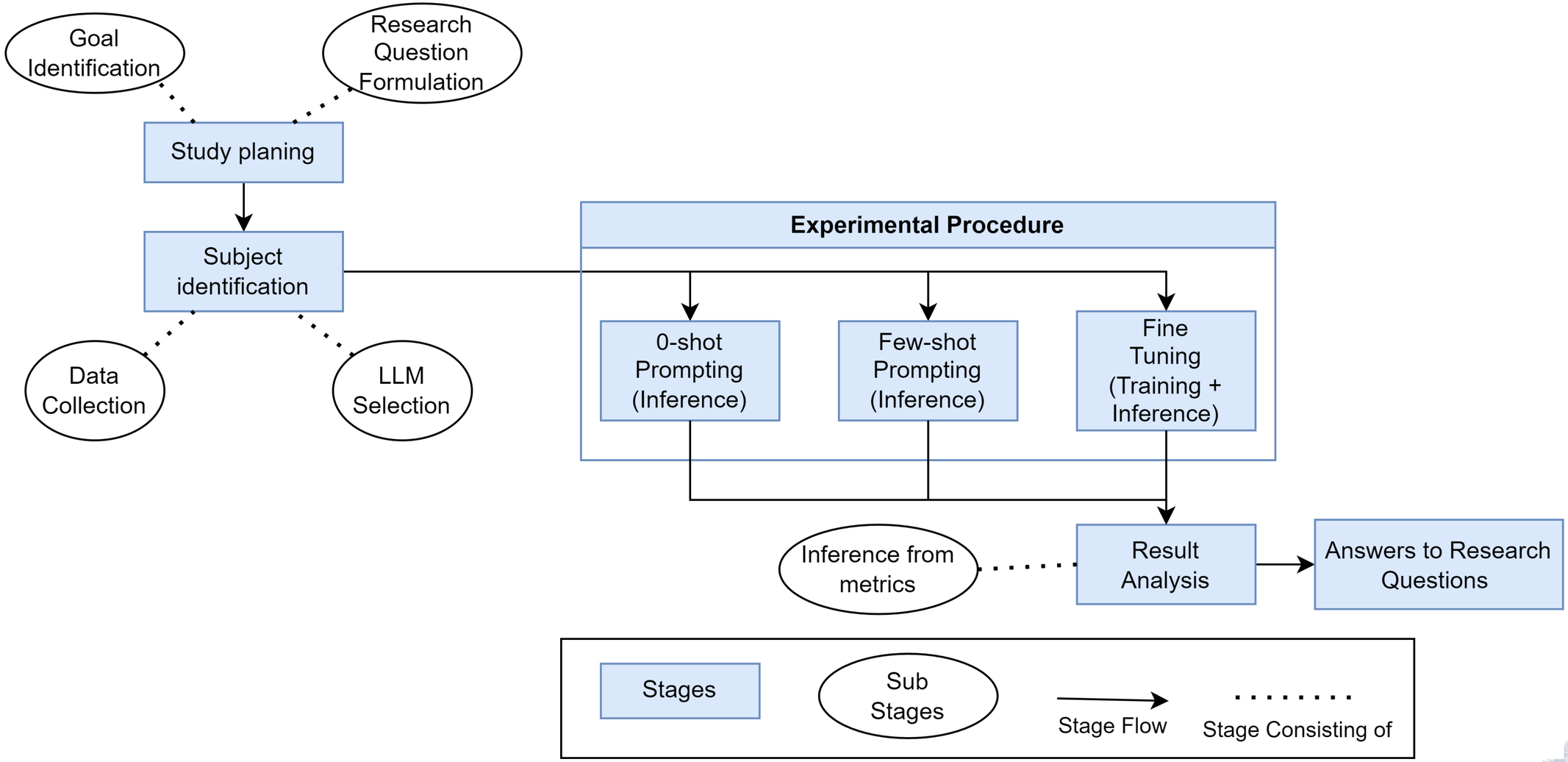
Status
Decided

Consequences
1. Functionality needs to be exposed as API for integration support

Karthik Vaidyanathan

https://github.com/joelparkerhenderson/architecture-decision-record

Performing an Exploratory Study



Generation Approaches

Zero Shot Prompting - Just ask the model and expect the answer

SYSTEM

This is an Architectural Decision Record for a software. Give a ## Decision corresponding to the ## Context provided by the User

USER

#context: " We need to decide on whether to use Python as a
programming language for our project. Our project involves data analysis, machine learning, and web development.

ASSISTANT

Decision: Use Python as the programming language for the project.

Python is well-suited for tasks involving data analysis, machine learning, and web development due to its extensive libraries and frameworks dedicated to these areas. By leveraging the rich ecosystem of Python libraries such as NumPy, pandas, scikit-learn, TensorFlow, and Django, the development team can efficiently implement the required functionalities for the project. Additionally, Python's simplicity and readability make it a popular choice for teams working on diverse aspects of a project like data analysis, machine learning, and web development.

Generation Approaches

Few Shot Prompting - Give some examples in the context

SYSTEM

These are architecture decision records. Follow the examples to get return Decision based
on Context provided by the User.

USER

Context

We need to decide on which database management system (DBMS) to use for Project X. The database will be used to store and manage large amounts of data from multiple sources. We need a DBMS that can handle transactions, offer scalability, and provide high reliability and security. Among the various options available, we are considering MySQL as a possible choice.

ASSISTANT

Decision

After evaluating the context of our project, we have decided to use Python as the programming language.

Python is a versatile language known for its simplicity and readability, making it an excellent choice for data analysis, machine learning, and web development tasks. It has a rich ecosystem of libraries and

Give some examples

USER

Context
We need to decide on whether to use Python as a programming language for our project. Our project involves data analysis, machine learning, and web development.

ASSISTANT

Decision

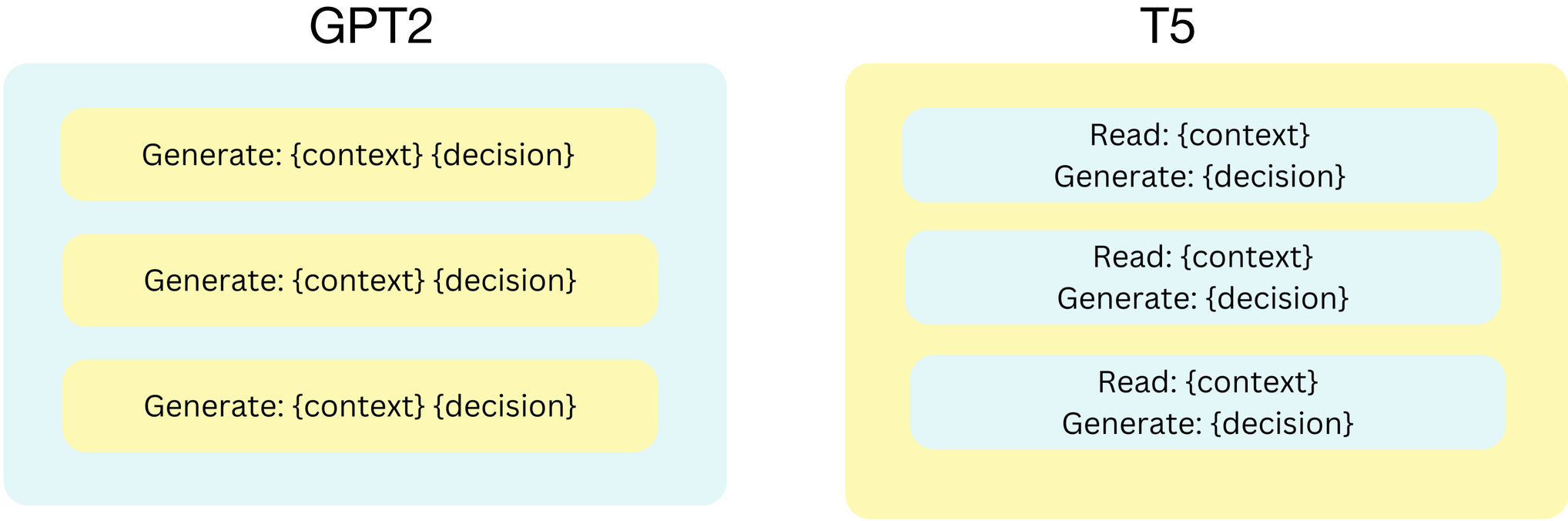
After evaluating the context of our project, we have decided to use Python as the programming language.

Python is a versatile language known for its simplicity and readability, making it an excellent choice for data analysis, machine learning, and web development tasks. It has a rich ecosystem of libraries and frameworks such as NumPy, Pandas, Scikit-learn for data analysis and machine learning, and Django, Flask for web development. Python's extensive community support and resources provide robust documentation, tutorials, and active user forums for problem-solving and knowledge sharing.

Get a response

Generation Approaches

Fine tuning - Fine tune base model with dataset



Selected LLMs

- **No of models:** 18
- **Model Series**
  - Encoder-decoder (T5 Series)
  - Decoder only (GPT series)
- **Model types**
  - Chat models
  - Text completion models
- From **small to large** models

family	model	size	context length	availability
GPT-2	GPT2 GPT2-medium GPT2-large GPT2-xl	124M 335M 774M 1.5B	1024	local
GPT-3	ada davinci	350M 175B	2048	api
GPT-3.5	text-davinci-003 GPT-3.5-turbo	175B 175B	4000	api
GPT-4	GPT-4	T+	8192	api
T5	T5-small T5-base T5-large T5-3b	60M 223M 738M 3B	infinite	local
T0	T0-3b	3B	infinite	local
Flan-T5	Flan-T5-small Flan-T5-base Flan-T5-large Flan-T5-xl	77M 248M 783M 3B	infinite	local

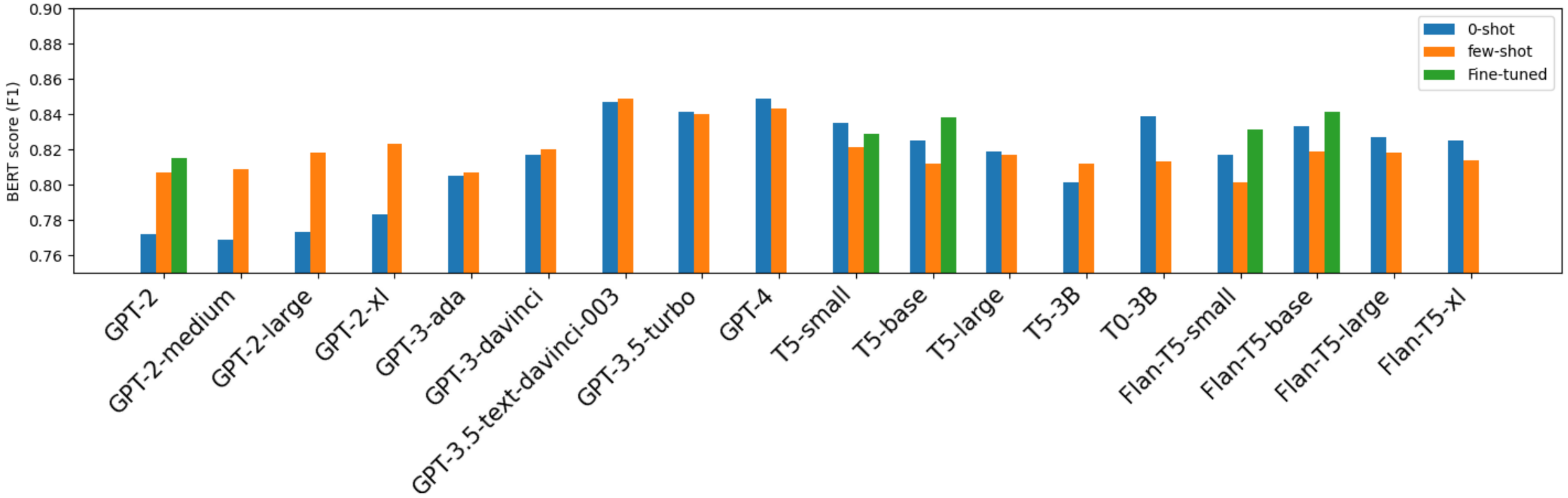
Results

Approach	Model	Rouge-1	BLEU	Meteor	Bert-Score			Support
					Precision	Recall	F1	
0-shot	GPT3.5-text-davinci-003	0.242	**0.031**	0.198	0.846	0.849	0.847	91
	GPT-4	**0.259**	0.028	**0.219**	0.847	**0.851**	**0.849**	95
	T0-3b	0.187	0.005	0.122	**0.856**	0.823	0.839	91
Few-shot	GPT2-xl	0.122	0.018	0.18	0.794	**0.857**	0.823	13
	GPT3.5-text-davinci-003	**0.245**	0.028	0.207	**0.849**	0.851	**0.849**	91
	GPT-3.5-turbo	0.226	0.027	**0.219**	0.832	0.85	0.84	92
	GPT-4	0.23	**0.03**	0.21	0.836	0.851	0.843	93
Fine-Tuning	Flan-T5-base	**0.231**	**0.028**	**0.171**	**0.842**	**0.841**	**0.841**	18

Smaller models performing well has lot of importance in software architecture domain!

The Overall Results

Study repo



Smaller models perform well!…there is a need for more data and context!

Generated Decisions

GPT4 zero shot

We will use Python as our programming language for this project. Python has emerged as a reliable, versatile language with a rich ecosystem of libraries and frameworks that cater to our specific needs including data analysis (Pandas, NumPy), machine learning (scikit-learn, TensorFlow), and web

GPT3.5 few shot

After considering the various criteria discussed, we have decided to go ahead and use Python for our project.

Python is a versatile, powerful language with a wide range of features, .. community of developers who can provide support.. Python is also open-source, free to use

Python is also a great choice in terms of **performance and scalability**…., Python is an interpreted language..easier to debug and **maintain** than compiled languages.

Overall, Python is a great choice for our

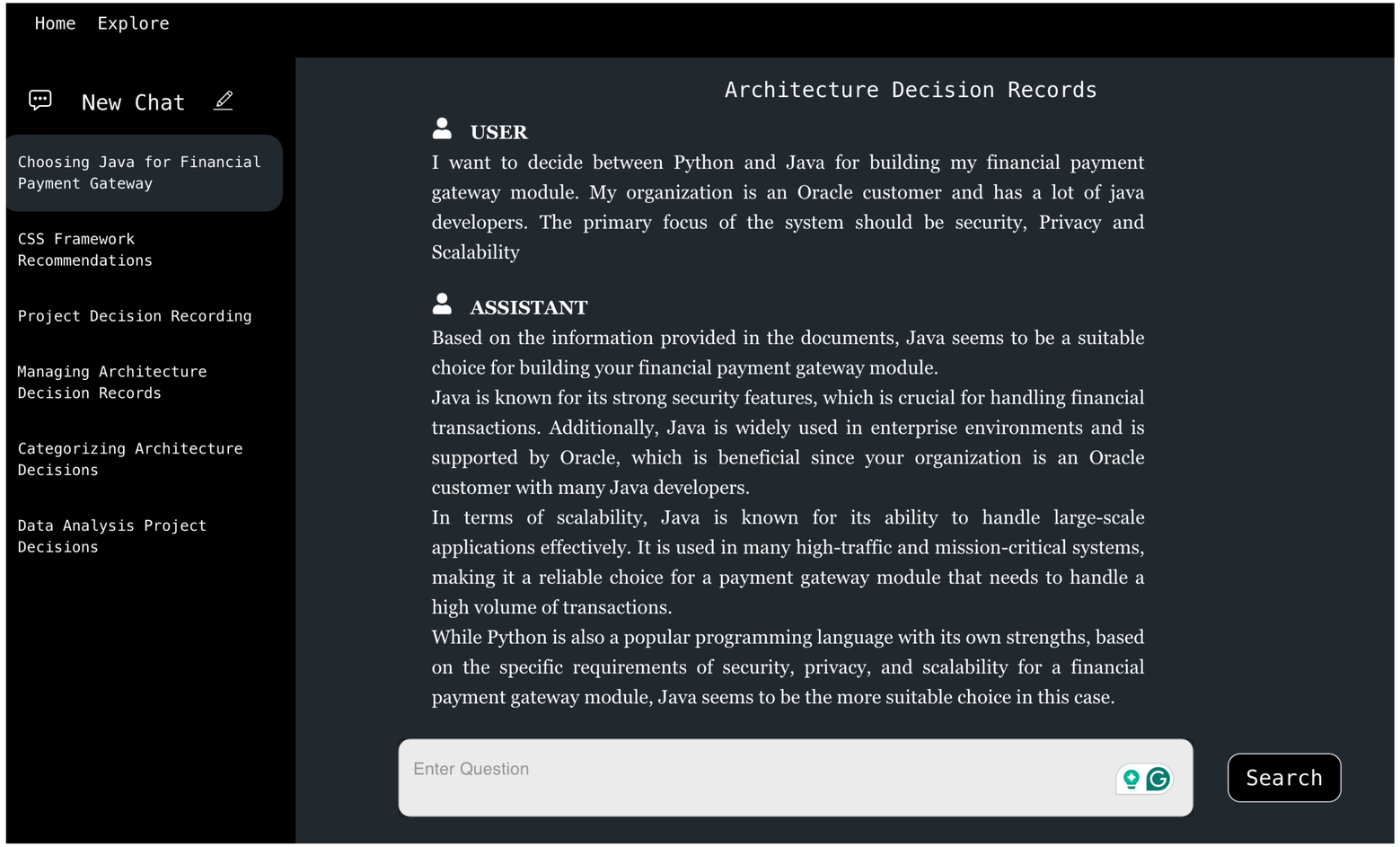
It will have a large dataset, which will display a lot of data. This can help us in preparing the application. We need to choose Python as a programming language, because all of our projects will require Python to use it.

Flan-T5 fine-tuned

Why not RAG?

- Software architecture is set of components and connectors.
- Software architecture is about forms, elements and rationale
  - [software, architecture, is, set, of, components, and, connectors]
  - Vector embedding: [1.5, 0.8, 0.6, 0.9. 0.4. 0.2, 0.5] - v1
  - [1.8, 0.2, 0.5, 0.9. 0.4. 0.6, 0.9] - v2
- Store vector embeddings in a Vector Data Base (VDB)
- **Retriever**
  - “What is software architecture”?: Matching documents V1 + V2
- **Generation:** Query + retrieved -> LLM -> “software architecture is about components and connectors. It is also about forms, elements and rationale”

Introducing the ArchCopilot (Support RAG)



1. Add knowledge - decisions..
2. Search for similar decisions
3. Retrieve knowledge
4. Generate decision records for a given context

A companion for architect!

Thanks to Nikunj Garg, Pabba Ananya, Vansh Pravin, Yatharth Gupta, Ananya Sudi, Rupasree Yeduru and Rudra Dhar

Going towards Code Generation

Can we automate generation of architectural components?

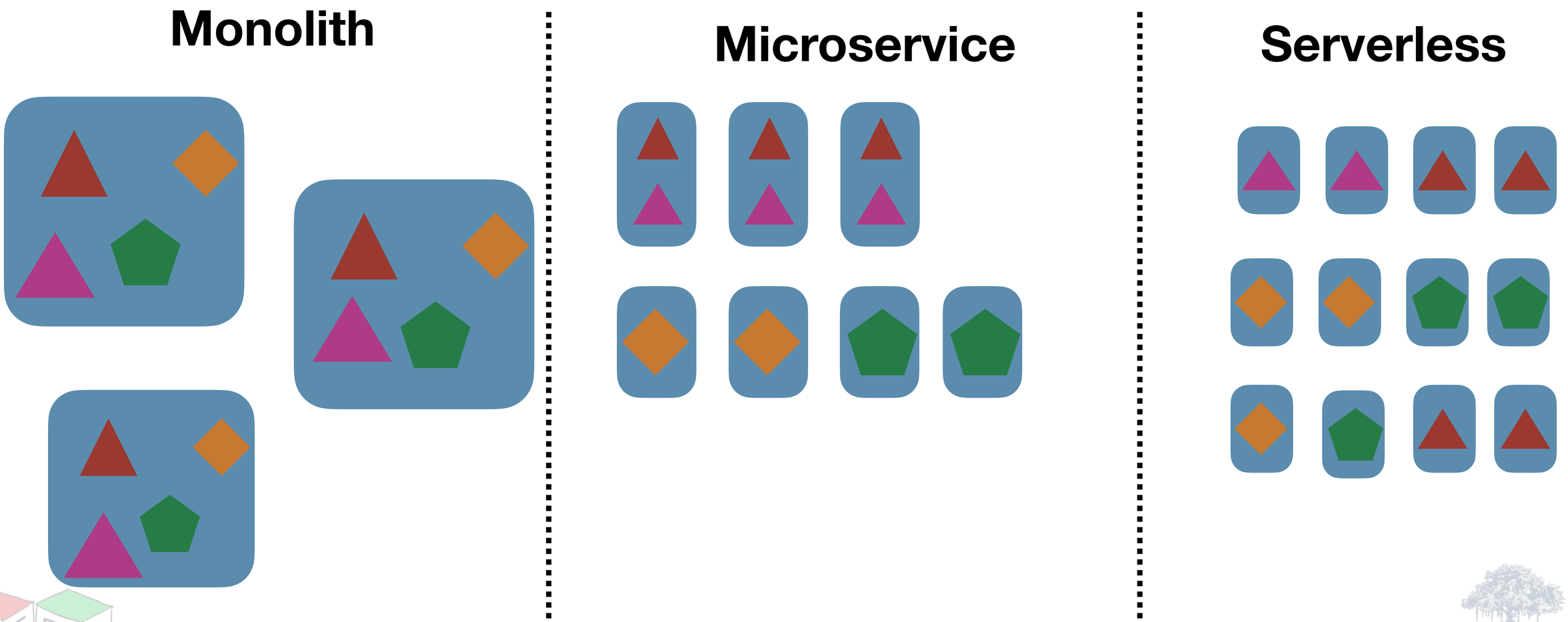
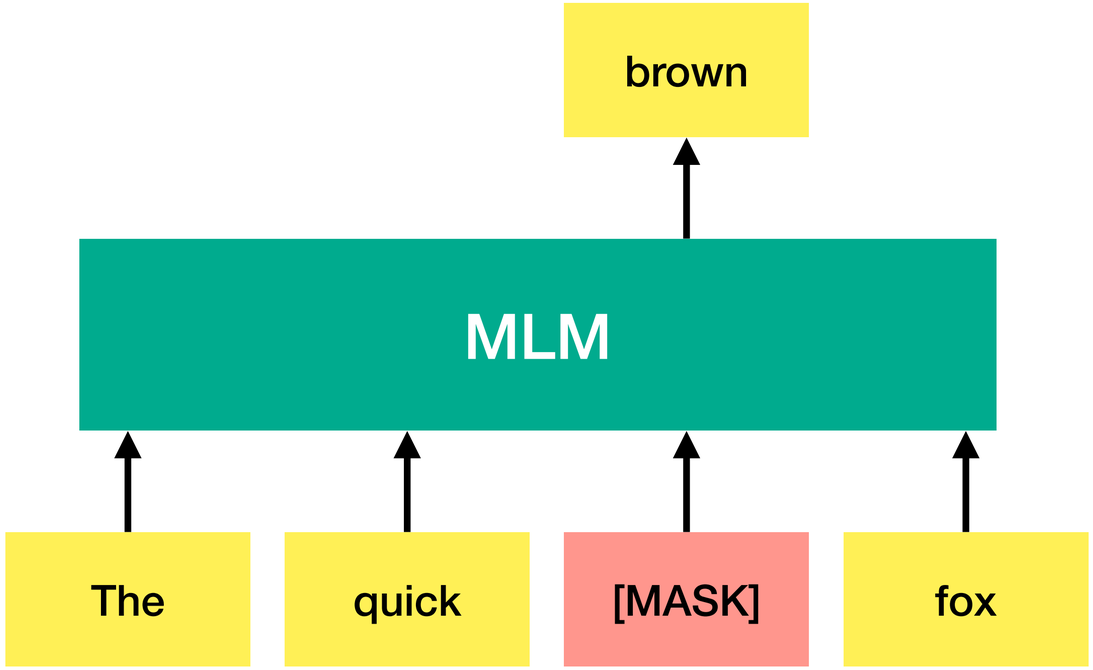


Image credits: Shrikara A

Generating Serverless Functions!

Inspired from the concept of masked language modeling



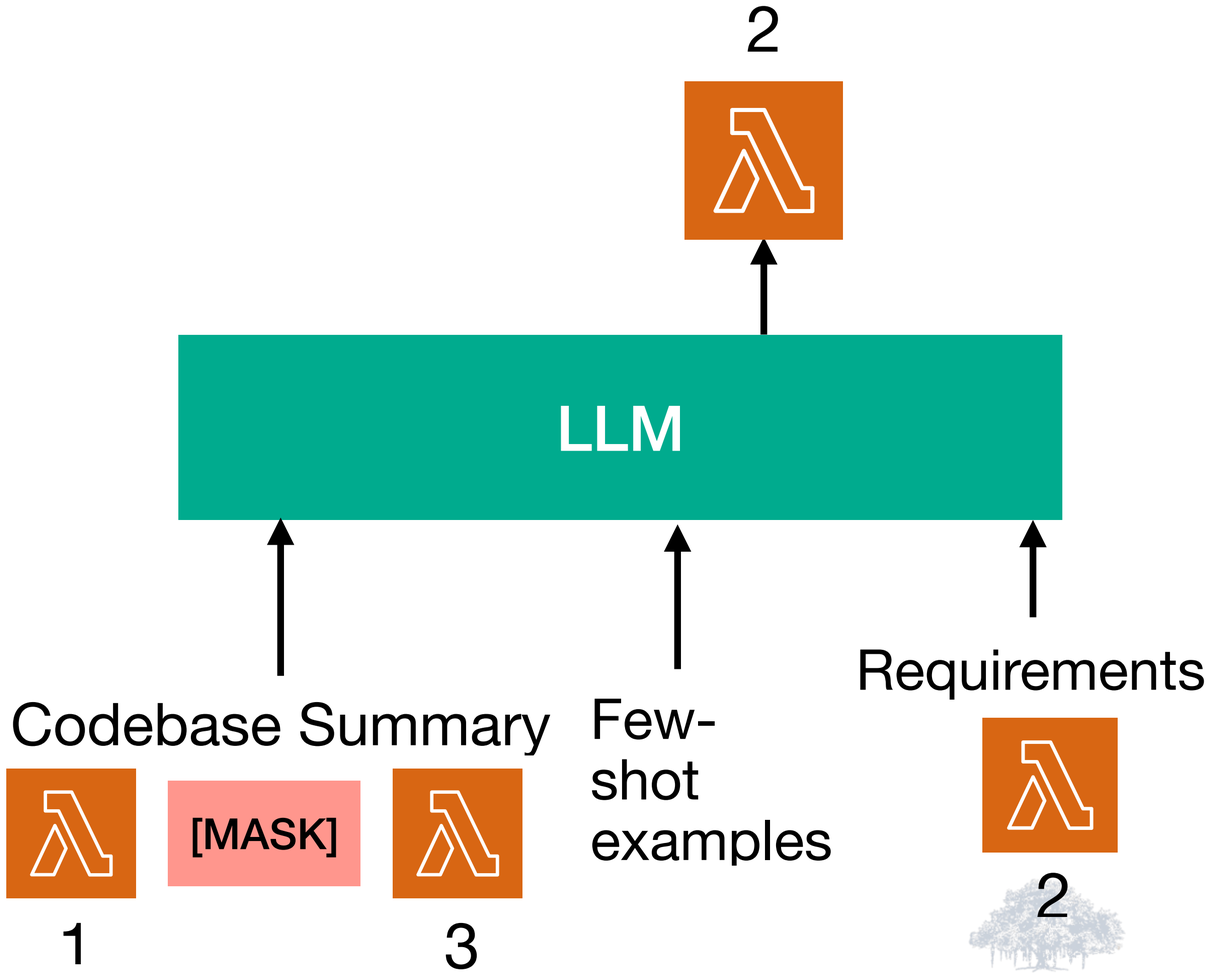
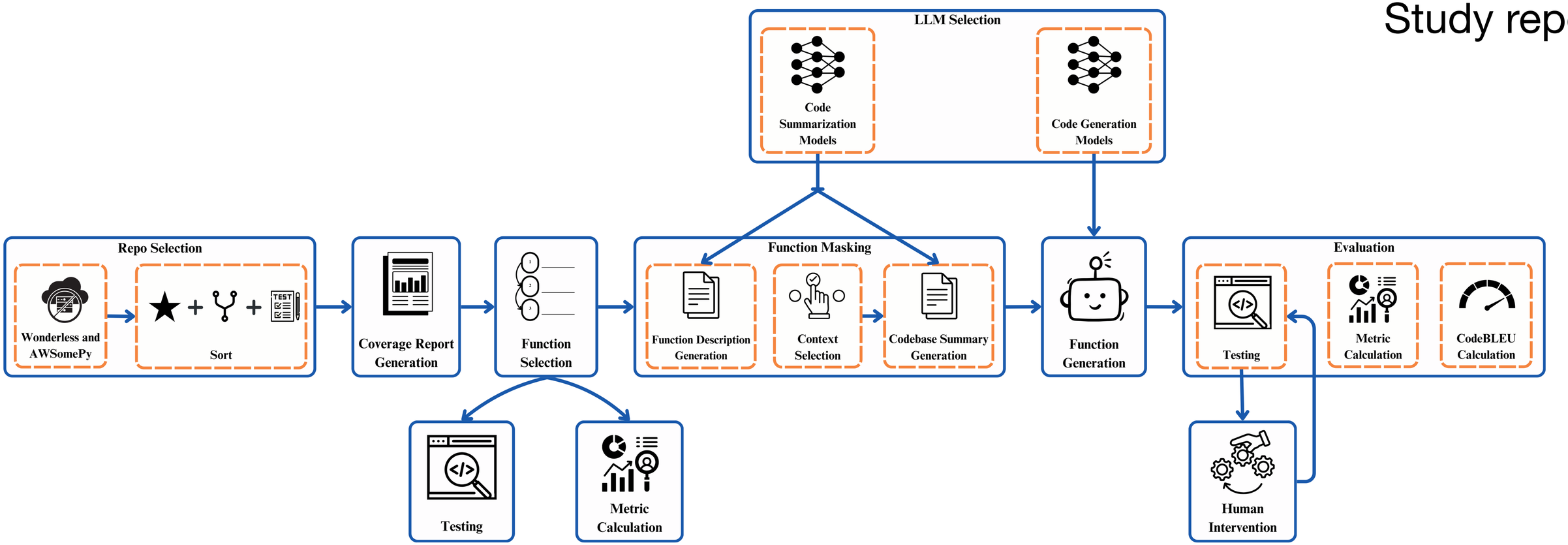


Image credits: Shrikara A

Study Design

Study repo

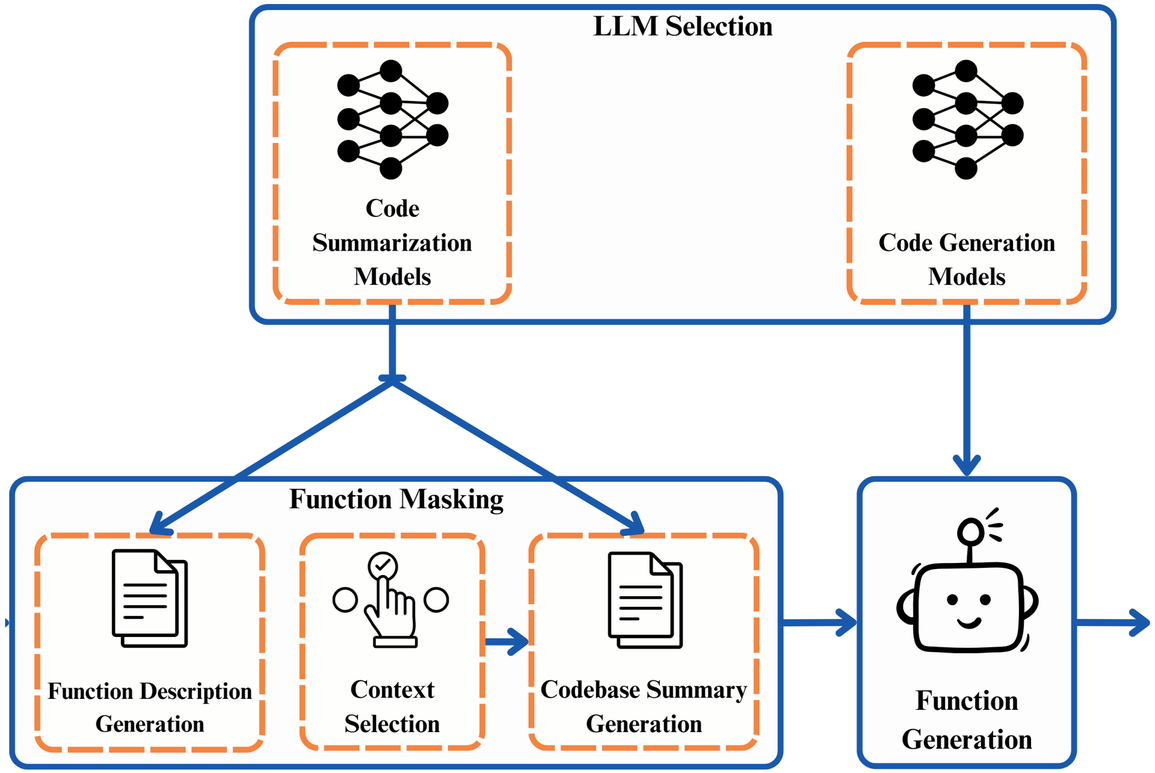


3 types of abstraction in prompt, 4 serverless repo, 5 models, 145 functions tested

Arun, S., Tedla, M. and Vaidhyanathan, K., 2025. **LLMs for Generation of Architectural Components: An Exploratory Empirical Study in the Serverless World.** ICSA 2025, To Appear. *arXiv preprint arXiv:2502.02539*.

Study Design

- Choose top LLMs from LM Arena and EvalPlus leaderboards
- Kinds of prompts:
  1. Zero-shot with Readme
  2. Zero-shot with codebase summarization
  3. Few-shot with codebase summarization



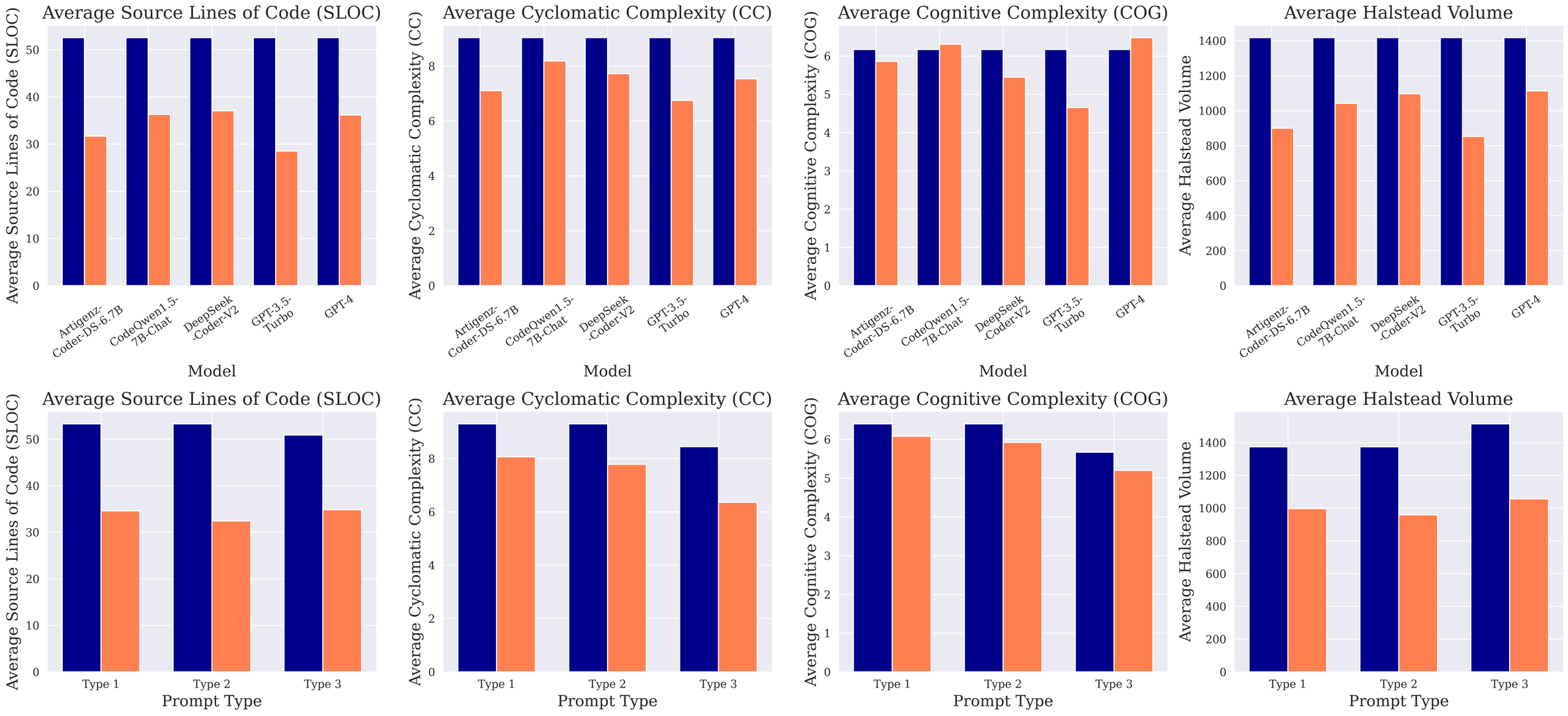
Chosen Models: Gemini 1.5 Pro for summarization, Artigenz-Coder-DS-6.7B, CodeQwen-1.5-7B, DeepSeek-V2, GPT-3.5-Turbo, GPT-4 for generation

Arun, S., Tedla, M. and Vaidhyanathan, K., 2025. **LLMs for Generation of Architectural Components: An Exploratory Empirical Study in the Serverless World.** ICSA 2025, To Appear. *arXiv preprint arXiv:2502.02539*.

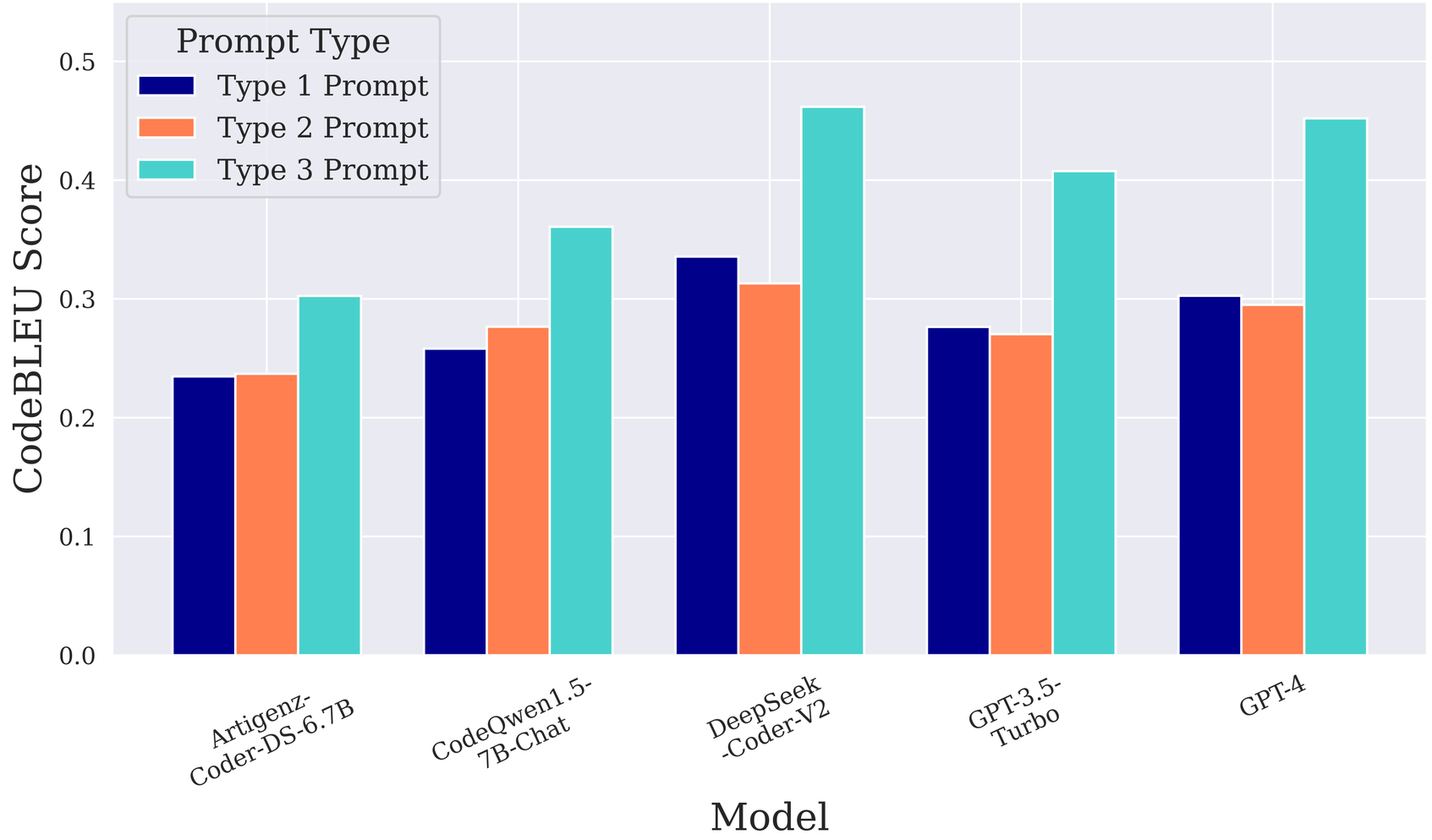
Some Key Results

Model	Initial Test Pass Rate	Type 1 (No interventio)	Type 1 (intervention)	Type 2 (No intervention)	Type 2 (intervention)	Type 3 (No intervention)	Type 3 (intervention)
Artigenz-Coder-DS-6.7B	100	0	0	0	0	0	0
CodeQwen1.5-7B-Chat	100	**3**	3	0	22	7	33
DeepSeek-Coder-V2	100	0	18	0	39	**13**	**71**
GPT-3.5-Turbo	100	0	3	0	50	4	64
GPT-4	100	0	**24**	0	**61**	10	50
Average	100	1	10	0	34	7	44

Some Key Results - Code Quality



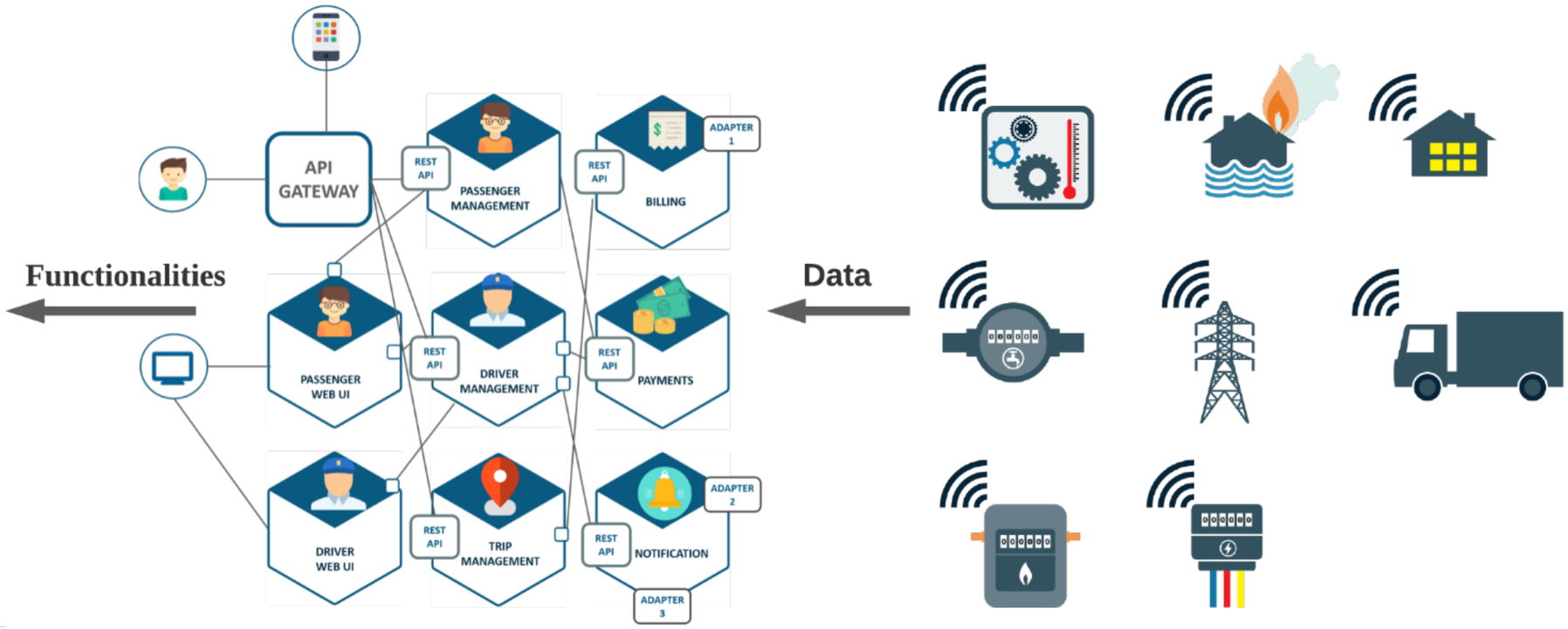
Some Key Results - Code Quality



Human Architects + Models + devs => Great combination!!

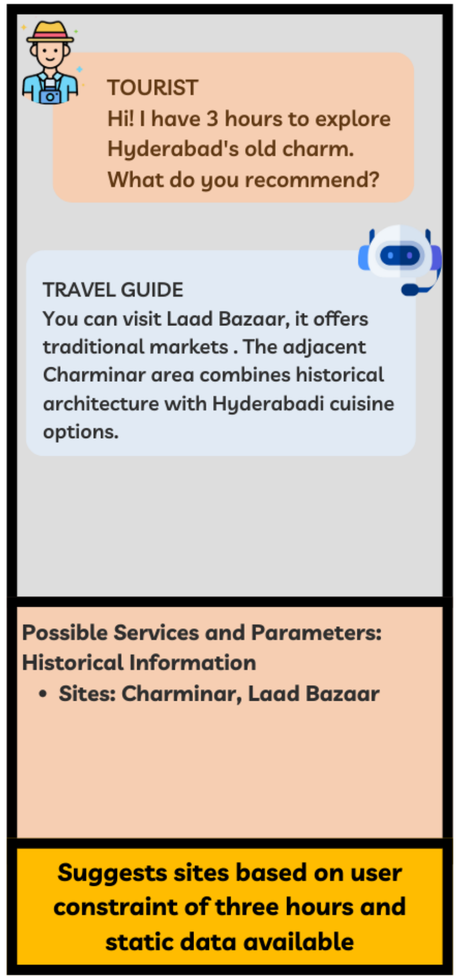
Towards deployment: Generate + Deploy

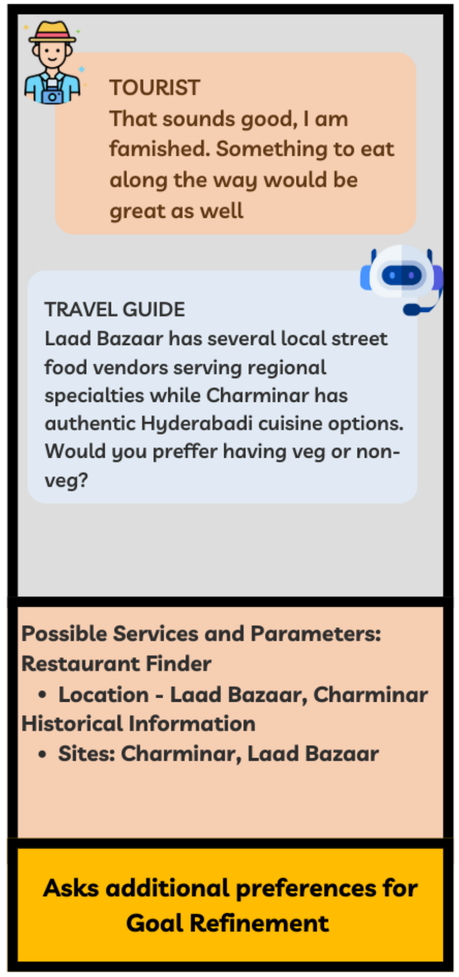
Applying to IoT Systems

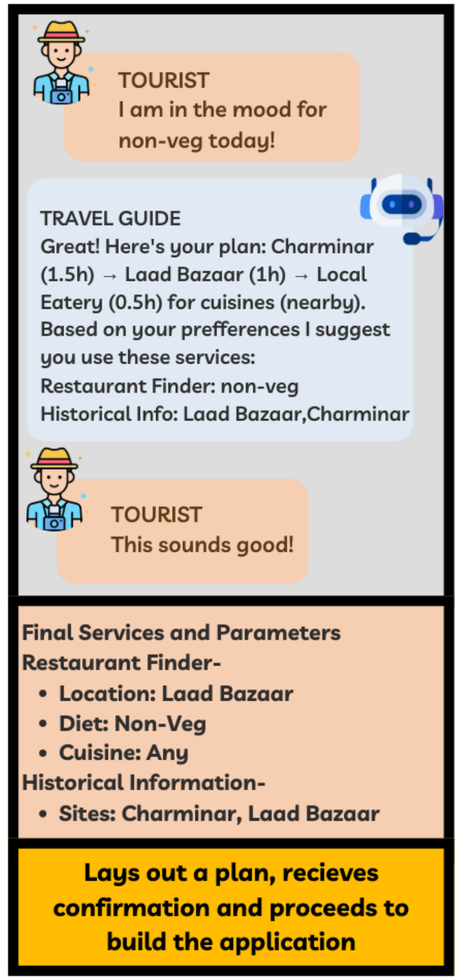


Applying to IoT Systems

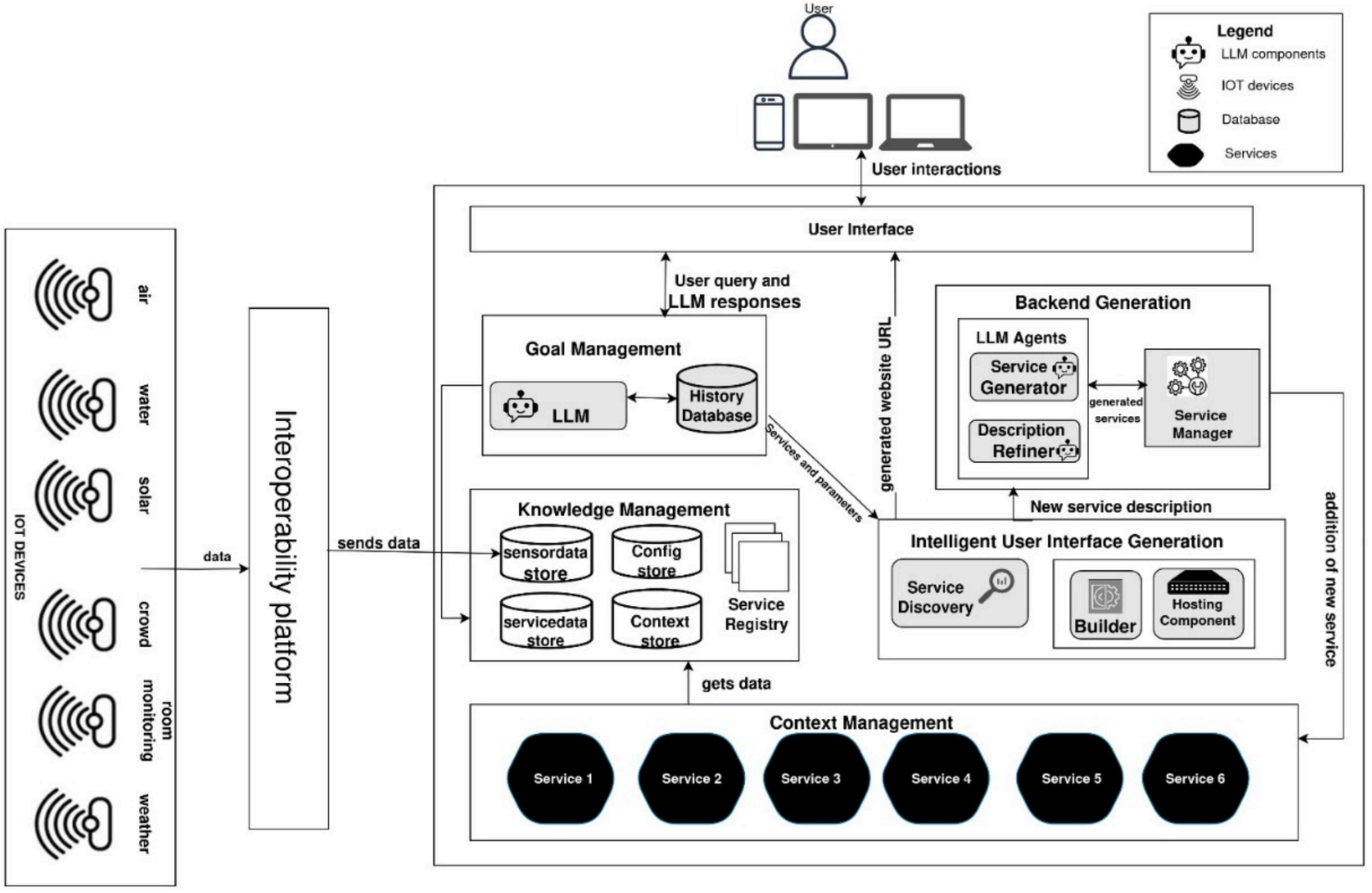
Dynamically generate services







IoT-Together: Mixed Initiative Interactions



Adnan, B., Miryala, S., Sambu, A., Vaidhyanathan, K., De Sanctis, M. and Spalazzese, R., 2025. **Leveraging LLMs for Dynamic IoT Systems Generation through Mixed-Initiative Interaction**.ICSA 2025 Companion, To appear. *arXiv preprint arXiv:2502.00689*.

Some Results

Website

GOAL PARSER PERFORMANCE BY CATEGORY

Model	Category	Precision	Recall	F1	Parameter Accuracy
CodeQwen1.5-7B	Ambiguous	0.450	0.806	0.553	0.116
	Concrete	0.206	0.609	0.288	0.051
	Overall	0.282	0.670	0.370	0.071
GPT-4o-mini	Ambiguous	0.683	0.795	0.730	0.549
	Concrete	0.467	0.773	0.559	0.739
	Overall	0.523	0.778	0.603	0.690
DeepSeek-V2.5	Ambiguous	0.681	0.788	0.725	0.585
	Concrete	0.492	0.830	0.591	0.743
	Overall	0.554	0.816	0.635	0.691

USER SATISFACTION METRICS

Metric	Average Rating (out of 5)
Application Rating	4.0
Accuracy Rating	4.1
Relevance Rating	4.2

Preliminary study with 15 users

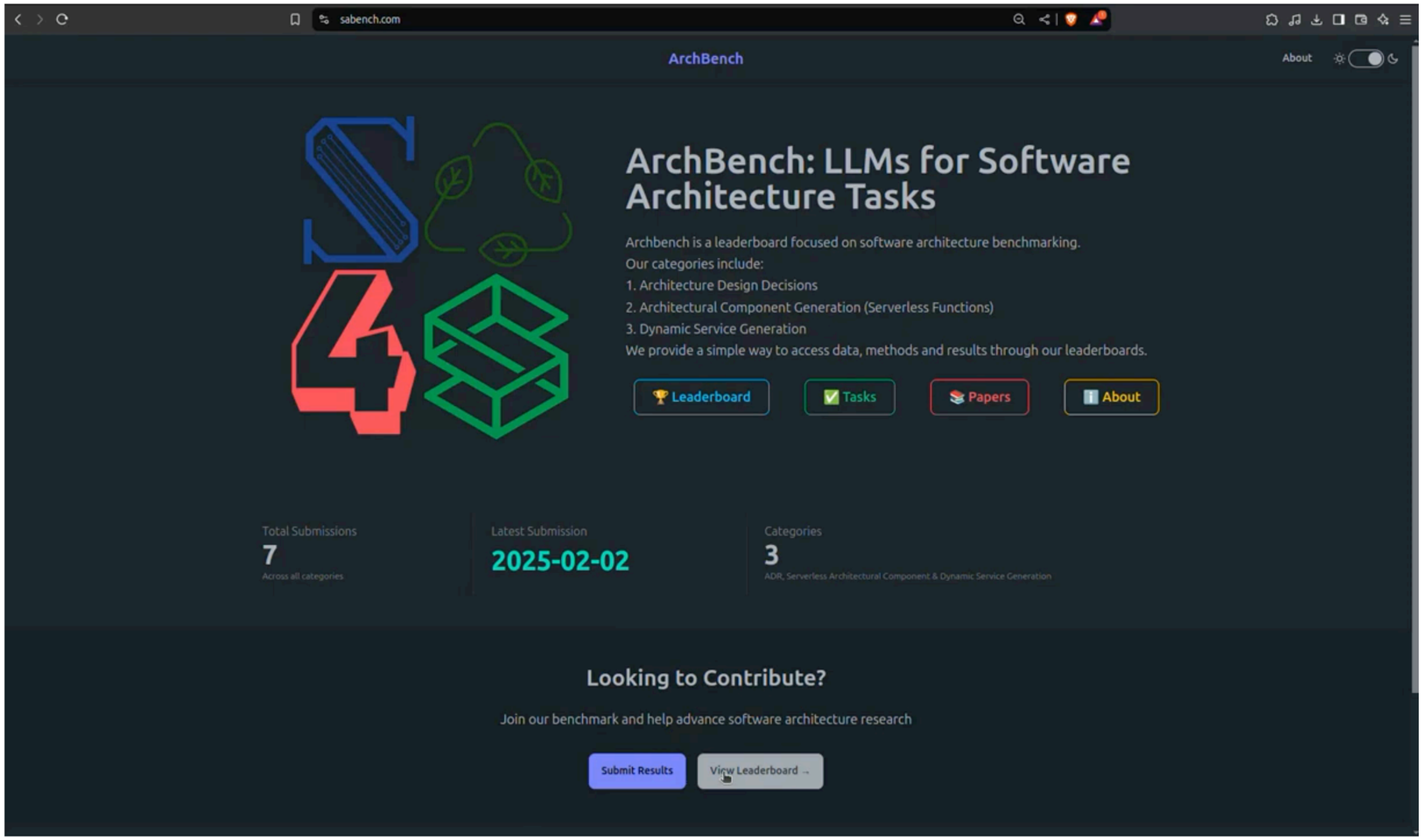
APPLICATION GENERATION PERFORMANCE METRICS

Metric	Mean ± SD	Min	Max
Total Duration (s)	23.10 ± 6.47	13.46	33.08
Total Token Usage	8164.90 ± 2718.89	5531	13991
Build Time (ms)	4.85 ± 1.98	3.50	10.49

Introducing sabench.com

LLMs for Software Architecture Tasks

- Collection of three tasks
  - ADR
  - Serverless function generation
  - Dynamic service generation
- Datasets and leaderboards
- You can contribute as well!
- **Feedbacks welcome!**



Extending Beyond Deployment: To Maintenance!

What if systems could adapt like human cells?

COVER FEATURE

The Vision of Autonomic Computing

Systems manage themselves according to an administrator's goals. New components integrate as effortlessly as a new cell establishes itself in the human body. These ideas are not science fiction, but elements of the grand challenge to create self-managing computing systems.

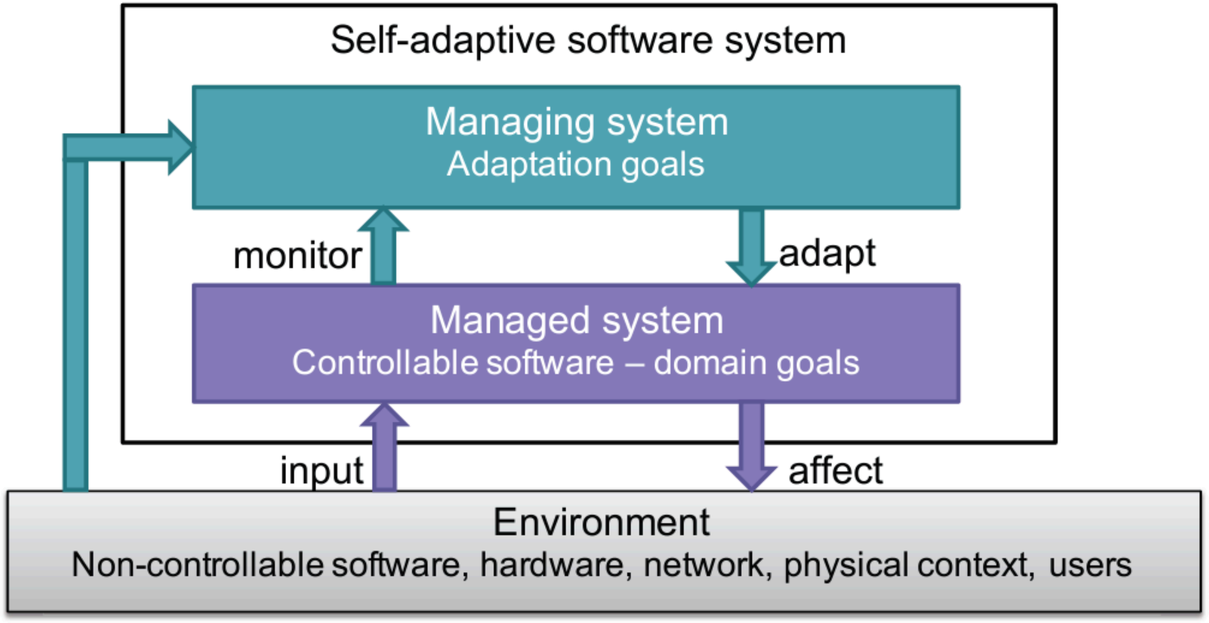
Jeffrey O. Kephart
David M. Chess
IBM Thomas J. Watson Research Center

In mid-October 2001, IBM released a manifesto observing that the main obstacle to further progress in the IT industry is a looming software complexity crisis.[1] The company cited applications and environments that weigh in at tens of millions of lines of code and require skilled IT professionals to install, configure, tune, and maintain.

The manifesto pointed out that the difficulty of managing today's computing systems goes well beyond the administration of individual software environments. The need to integrate several heterogeneous environments into corporate-wide computing systems, and to extend that beyond company figure, optimize, maintain, and merge. And there will be no way to make timely, decisive responses to the rapid stream of changing and conflicting demands.

AUTONOMIC OPTION

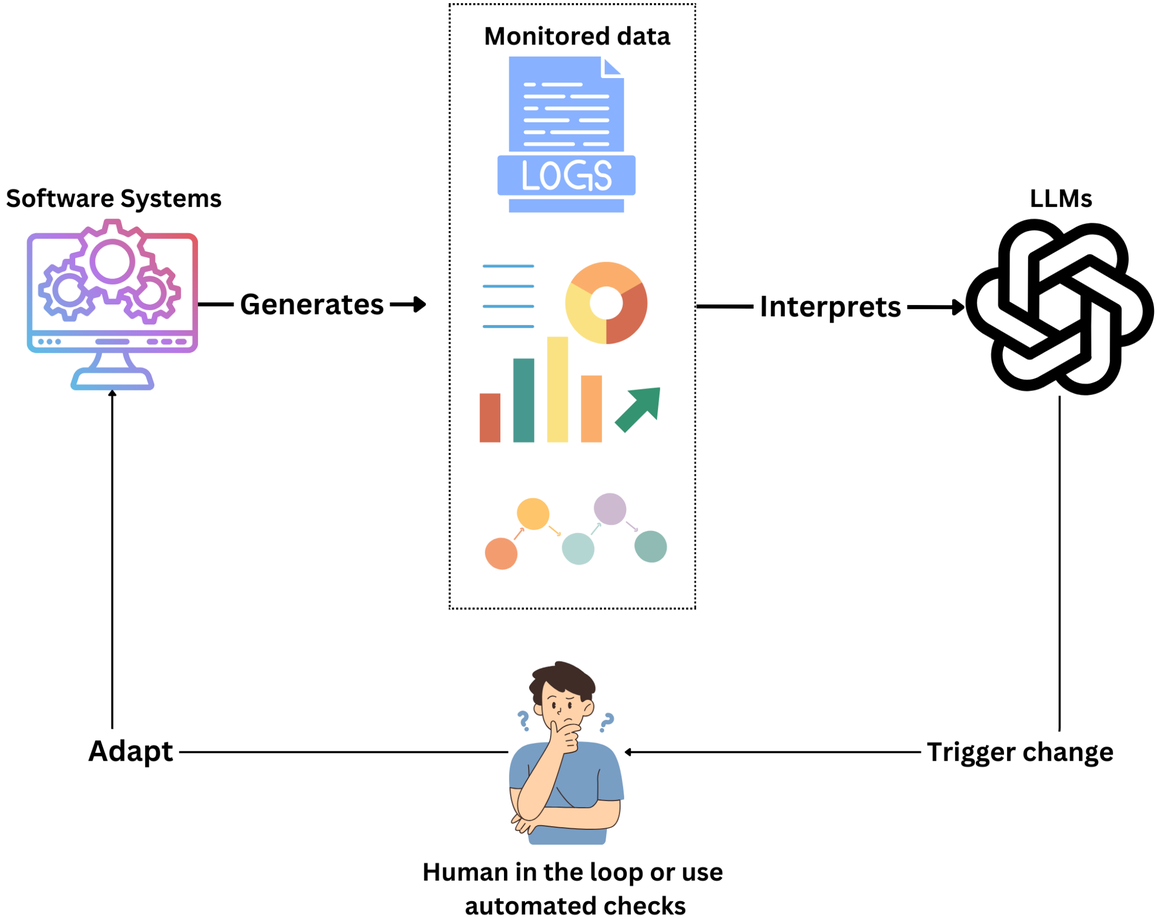
The only option remaining is *autonomic computing*—computing systems that can manage themselves given high-level objectives from administrators. When IBM's senior vice president of research, Paul Horn, introduced this idea to the National Academy of Engineers at Harvard University in a March 2001 keynote address, he deliberately chose a term with a biological conno-



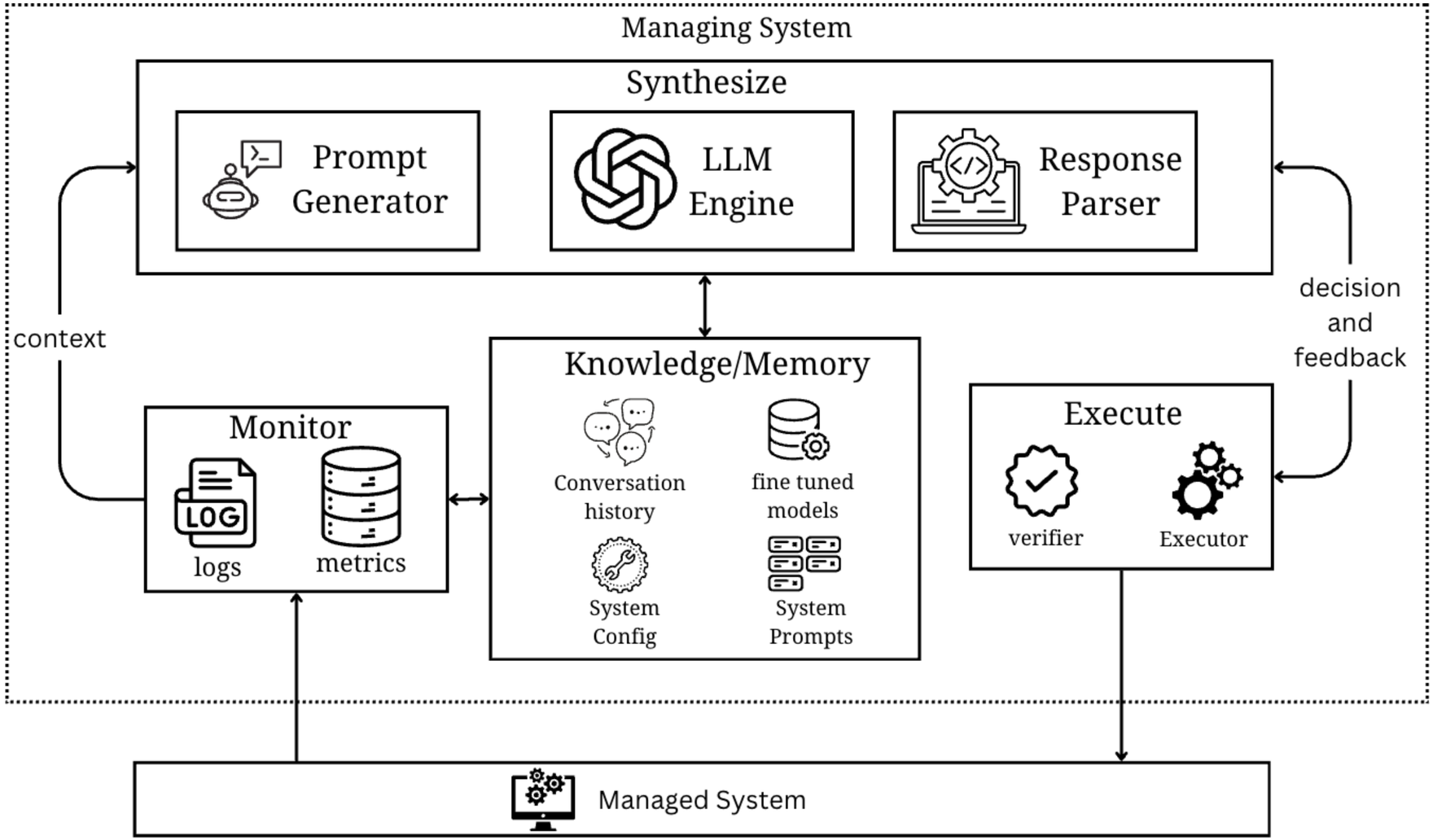
[Image: Ilias et al. 2022, ICSE NIER, Expressing the adaptation intent as a sustainability goal]

Extending beyond

Design time to run-time adaptation - Can LLMs help?



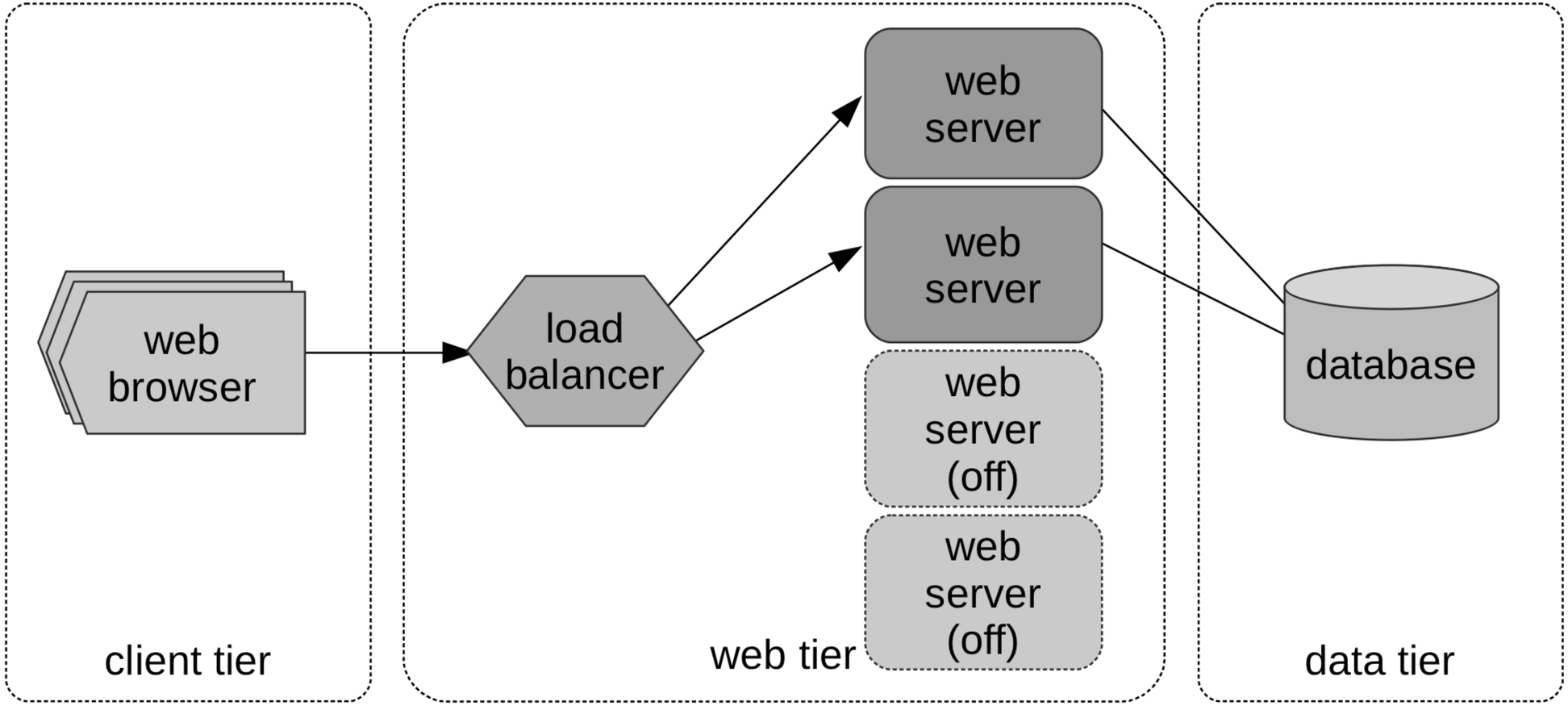
Reimagining Self-adaptation loop



R.Donakanti, P.Jain, S.Kulkarni, K.Vaidhyanathan, **Reimagining Self-adaptation in the age of Large Language Models,** ICSA 2024, Poster

SWIM case study

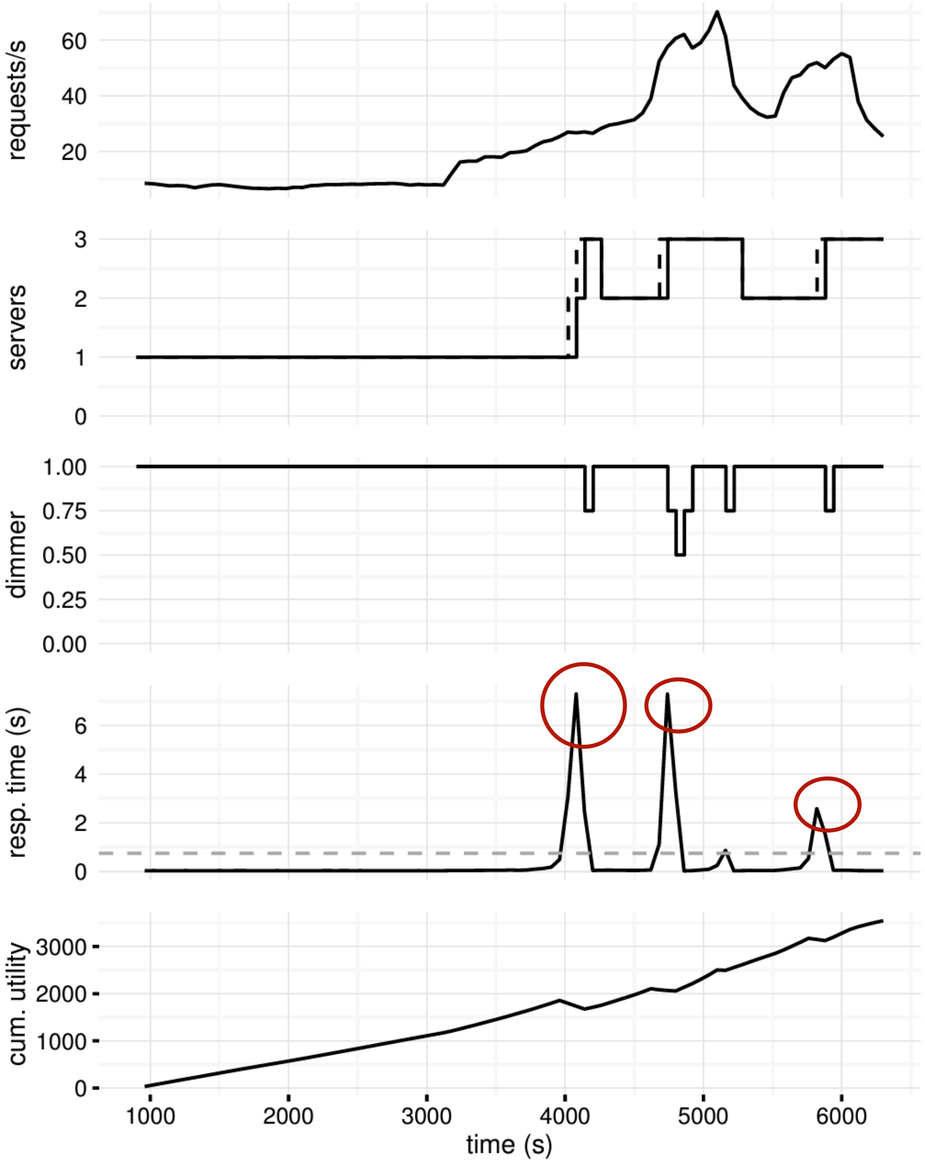
Web Infrastructure Simulator



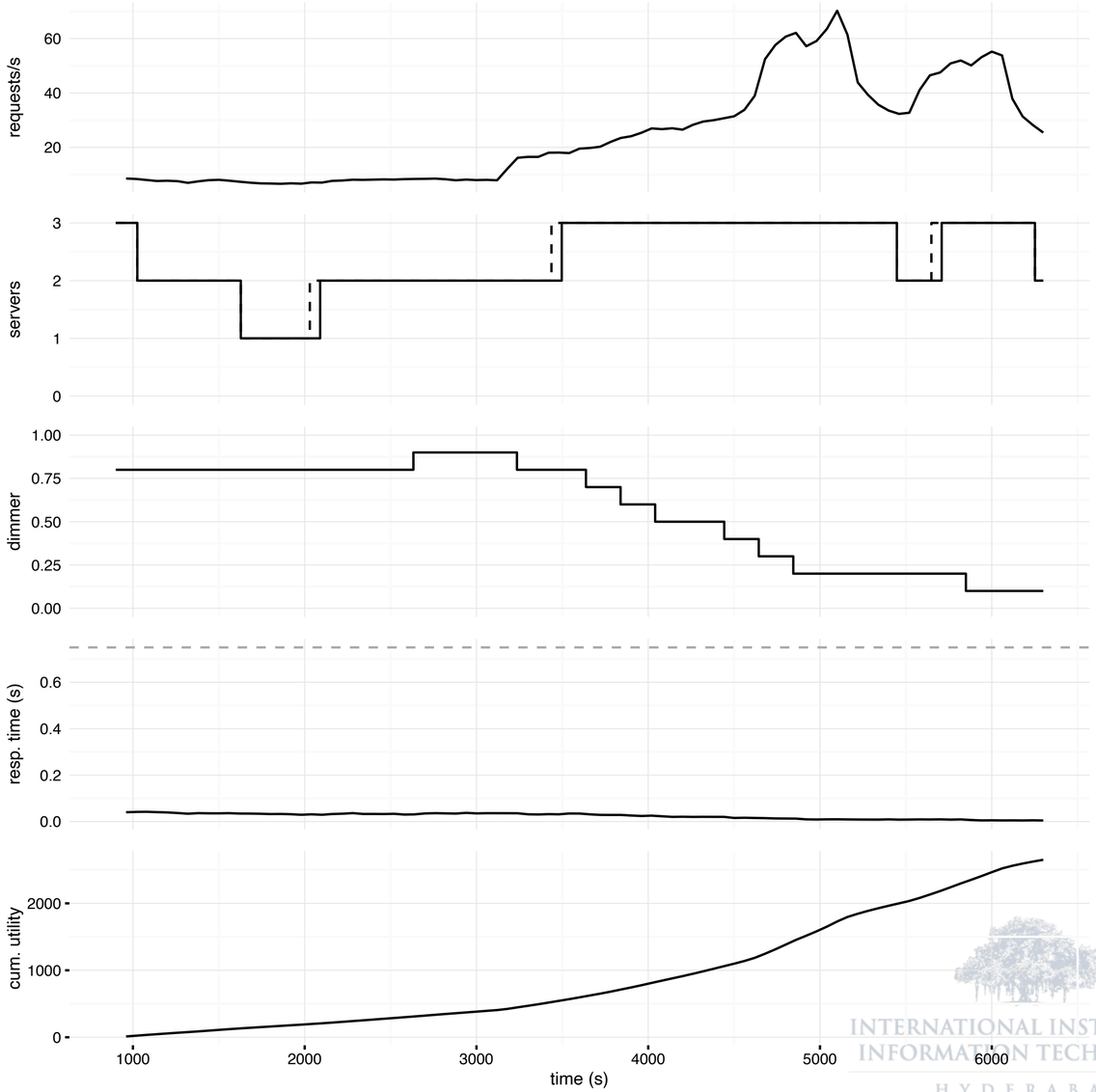
Moreno, Gabriel A., Bradley Schmerl, and David Garlan. **Swim: an exemplar for evaluation and comparison of self-adaptation approaches for web applications.** In 2018 IEEE/ACM 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), pp. 137-143. IEEE,

Some Initial Results

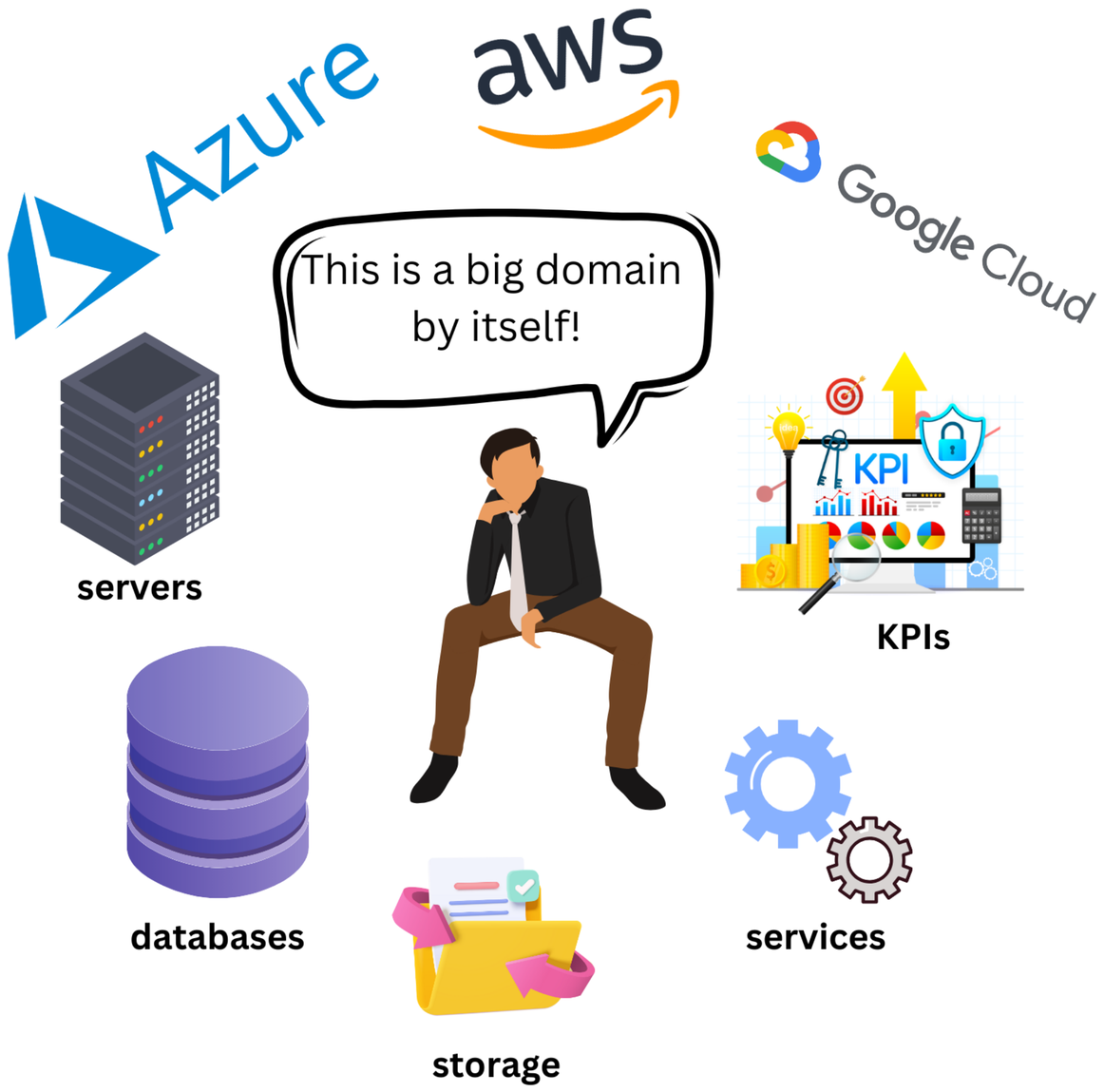
Using SWIM reactive adaptation



Using GPT-4, Promising but..



Making it more Concrete - CloudOps domain

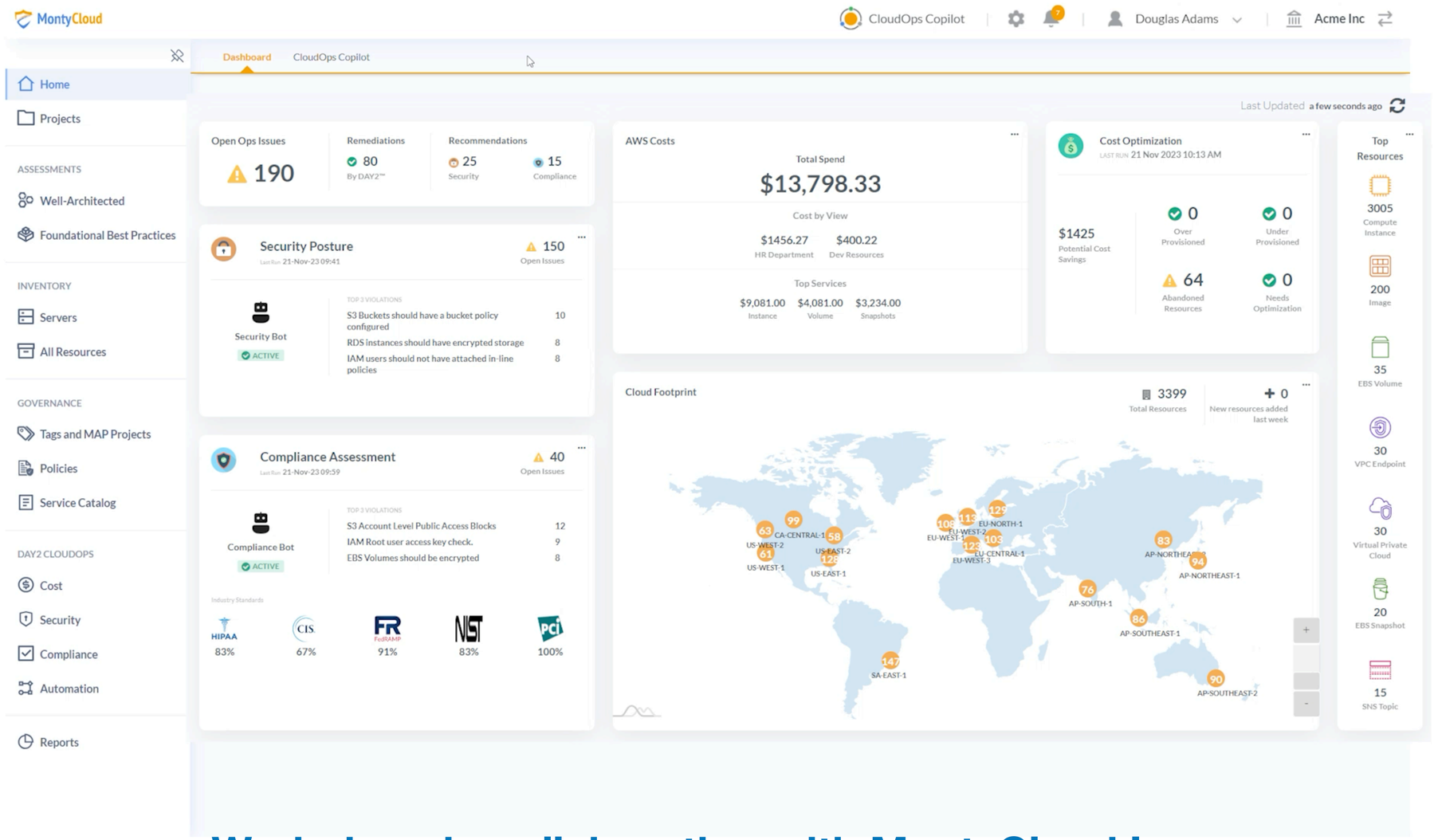


AWS Well Architected Framework

Helps cloud architects build resilient, secure and high performing infrastructure

- **Build around six pillars**
  - Operational Efficiency
  - Security
  - Reliability
  - Performance Efficiency
  - Sustainability
  - Cost

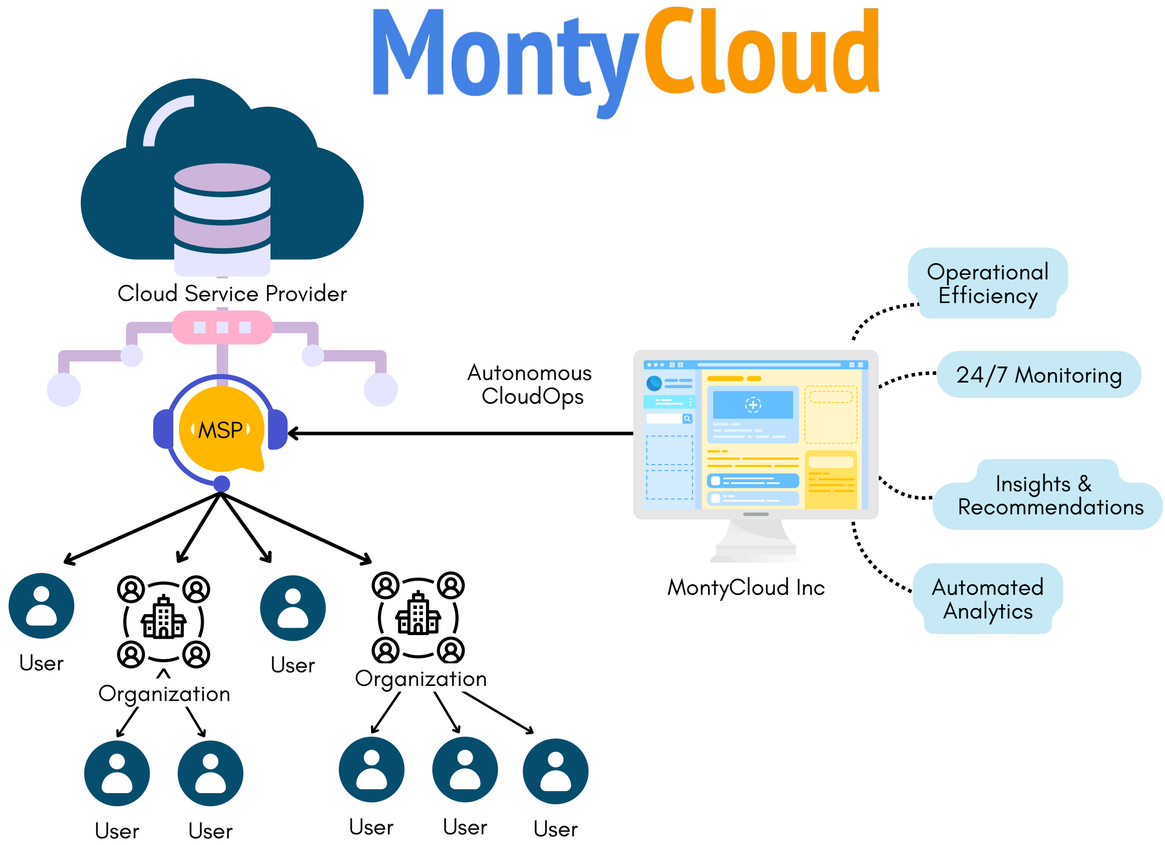
Ideas into Production: CloudOps CoPilot



Work done in collaboration with MontyCloud Inc.

Engineering Challenges (SE for GenAI)

- **Managing Distributed Data**
  - Diverse data sources
- **Maintainability**
  - Large code base, time for updates
- **Extensibility and Modularity**
  - Single vendor, ease of extensions!



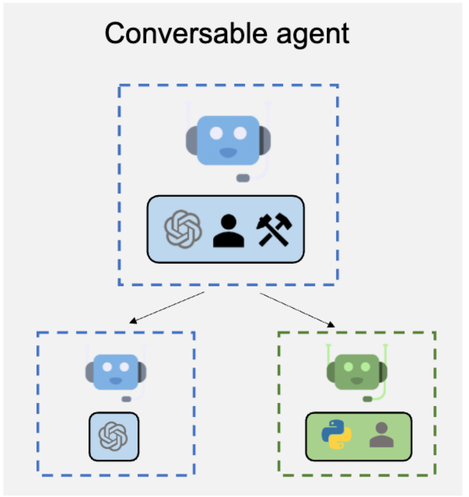
Moving to the world of Agentic AI

- **Agent:** Any system or program that can autonomously perform a task on behalf of user or another system
  - Each agent has role and responsibilities
  - Can leverage tools at their disposal to achieve their tasks
  - Have their own memory. Converse with other agents
  - Move from API to text!

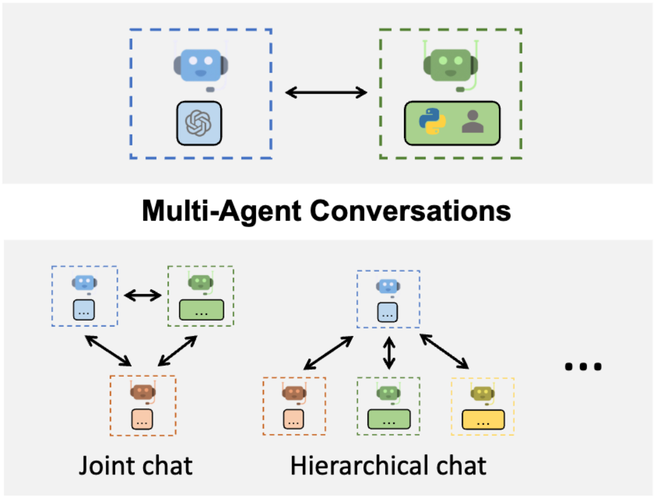
crewai

LangGraph

AG



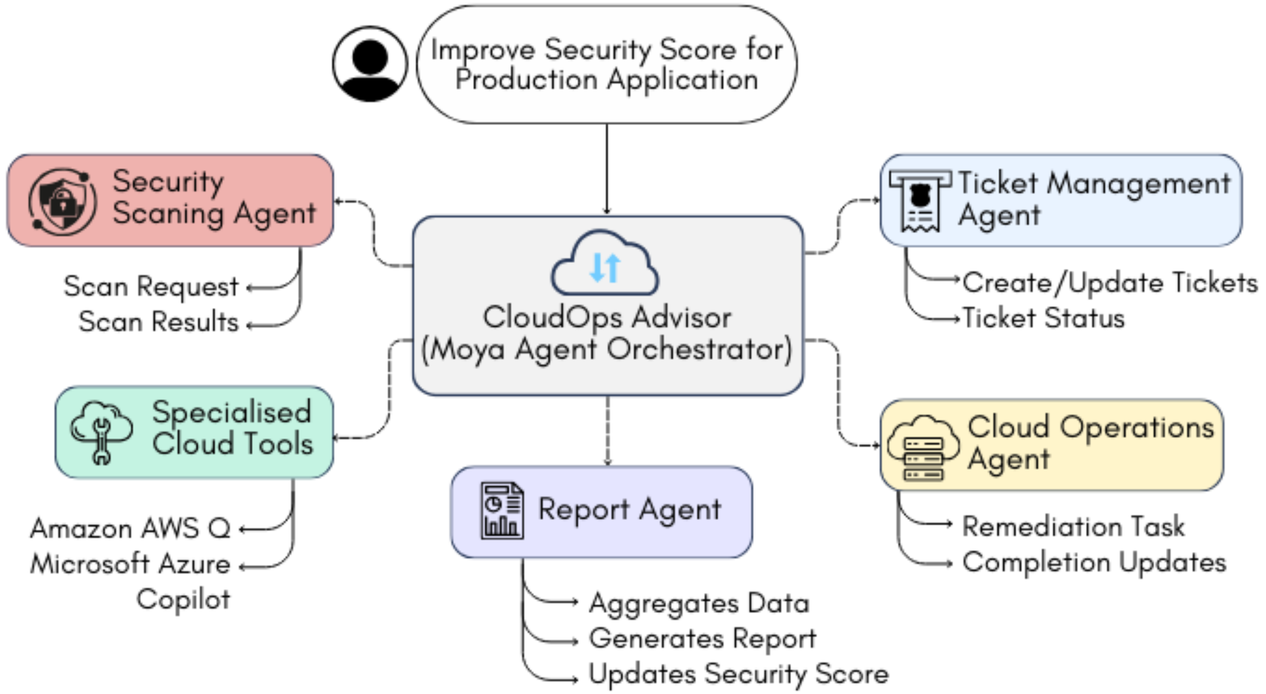
Agent Customization

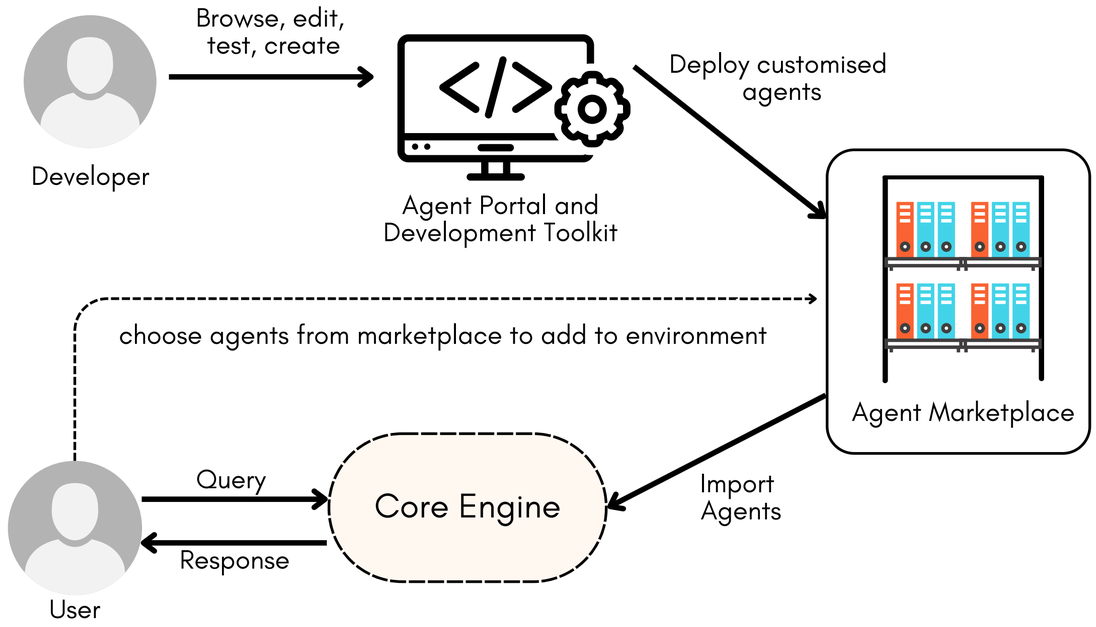


Flexible Conversation Patterns

Image source: https://microsoft.github.io/autogen

Can we go Multi-agent?

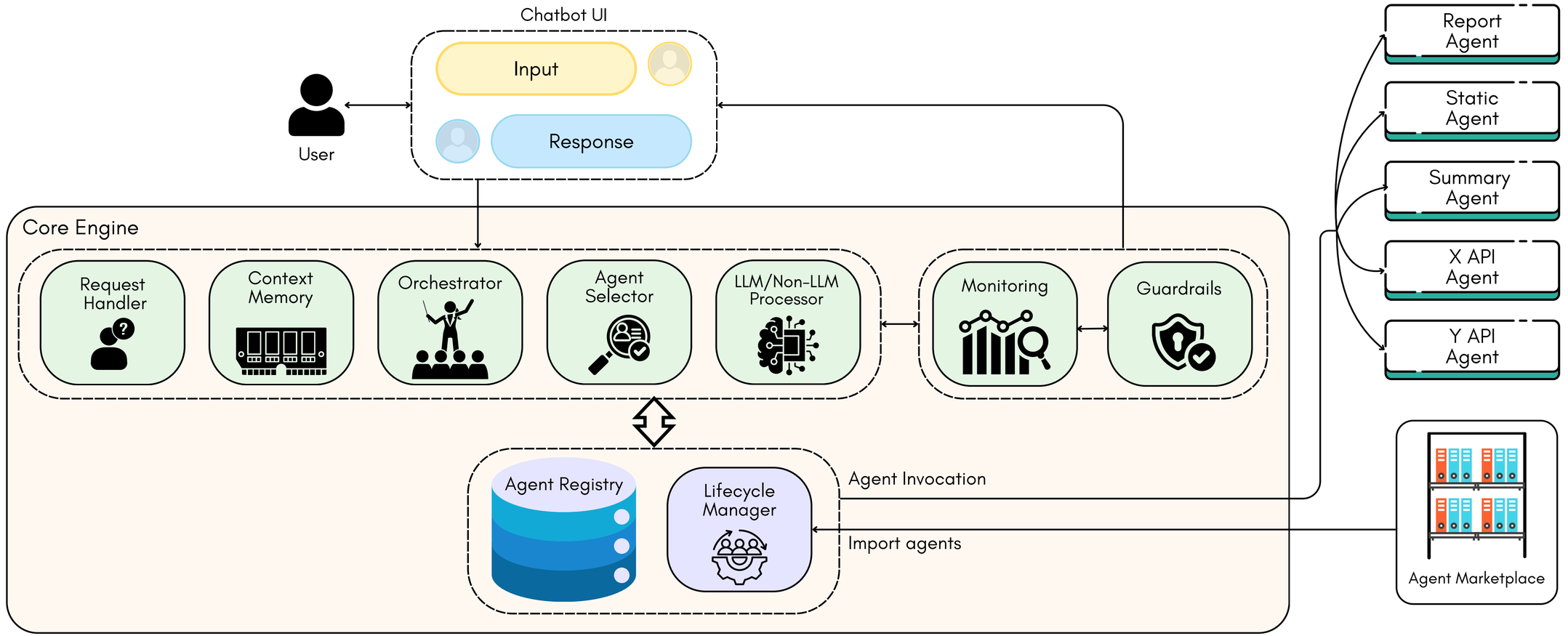




Meta orchestration Framework

The MOYA Multi-agent Framework

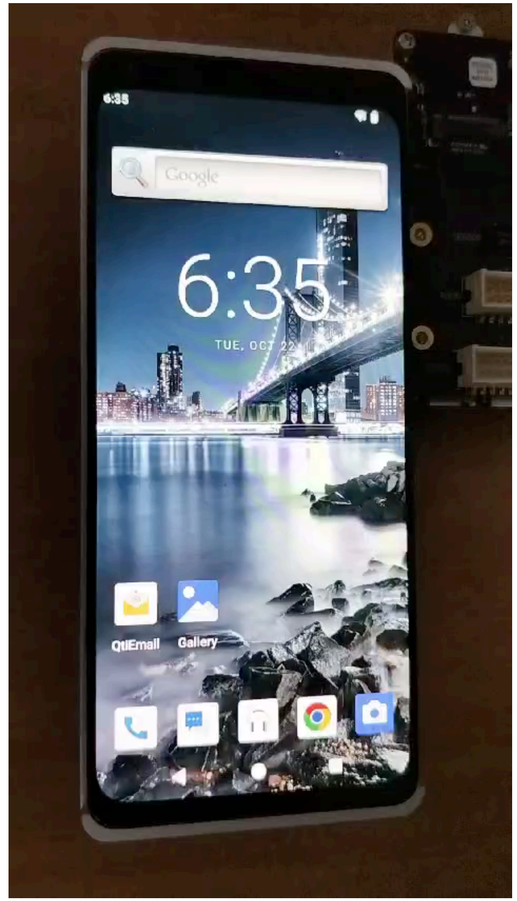
MOYA repo



Parthasarathy, K., Vaidhyanathan, K., Dhar, R., Krishnamachari, V., Muhammed, B., Kakran, A., Akshathala, S., Arun, S., Dubey, S., Veerubhotla, M. and Karan, A., 2025. **Engineering LLM Powered Multi-agent Framework for Autonomous CloudOps**. *arXiv preprint arXiv:2501.08243,* CAIN 2025. To Appear

Into the world of SLMs for SE

- LLMs are great but there are also challenges in using proprietary LLMs
- Ongoing research in:
  - SLMs for architects for design decisions
  - SLMs for edge deployment (Qualcomm EdgeAI labs @IIITH) using QIDK, Qualcomm
  - Using SLMs for function calling - Code generation (with Precog, IIITH)
  - …..



Source: https://www.qualcomm.com/videos/dynamic-ml-model-switching

Our Efforts in GenAI4SE

Requirements

Design

Development

Testing

Deployment

Maintenance

Diagrams from Descriptions [Collaboration]

Karthik Vaidyanathan

On Prem LLMs for generating Design Decisions

Karthik Vaidyanathan

Automated generation of serverless functions

Karthik Vaidyanathan

Function calling Using Small Language Models [Collaboration]

Karthik Vaidyanathan

LLLMs for autonomous self-adaptation (multi-agent)

Karthik Vaidyanathan

Autonomous CloudOps (Multi-agent)

Karthik Vaidyanathan

Generating IaaC [Collaboration]

Karthik Vaidyanathan

Automated development and deployment of services

Karthik Vaidyanathan

Key Takeaways

LLMs can be a best friend to a software architect if used wisely!

- LLM presents a great opportunity for effective SE!
- We need to have an effort to have qualitative data on architecture - **ArchBench is a step!**
- Domain specific LLMs which are smaller shall be the way forward - collection of SLMs!
- Need for better ways to architect/engineer systems around LLMs (SA4LLM-enabled Systems)
- LLMs are not here to replace but to support!
- **Systems thinking is becoming more important!**

The Future is here

- **Context is needed** - Capturing organizational aspects needs work, code can help! (Ongoing)
- **LLMs will hallucinate** - No stopping that but we can reduce it - better engineering!
- **Multiple agents collaborating** together to help software engineers
- **Large action models (LAMs)** for self-adaptation, task generation
- Lot of potential for support in **architecture migration**
- **SE Process will also need upgrades!!**

SA4S@SERC


Rudra Dhar


Akhila Matathammal


Hiya Bhatt


Chandrasekar S

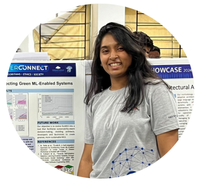

Shubham Kulkarni


Adyansh Kakran


Prakhar Jain


Shrikara A


Arya Pravin Marda


Meghana Tedla


Miryala Sathvika


Prakhar Singhal


Amey Karan


Bassam Adnan


Aneesh Sambu


Shaunak Biswas


Shailender Goyal

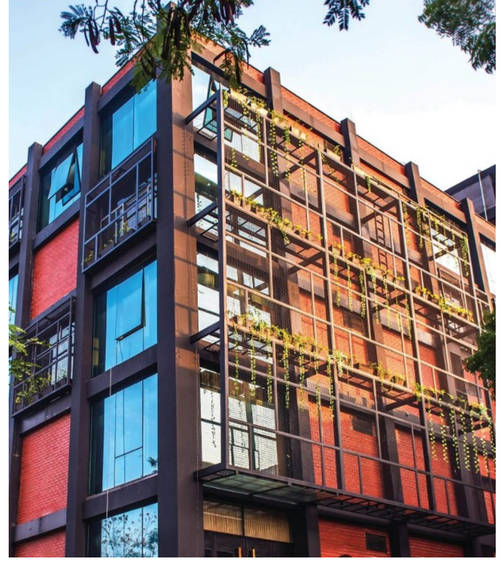

Sreemaee Akshathala


Divyansh Pandey


Maddireddy Kritin


Santosh Kotekal


Vyakhya Gupta



https://serc.iiit.ac.in

Team SA4S

https://sa4s-serc.github.io

https://sa-ml.github.io/saml2025/

@ ICSA 2025, SAGAI 2025

IEEE Software Magazine

ODENSE • DENMARK
ICSA 2025
31 MARCH - 4 APRIL

SCAN ME

Thank you

SE Radio Podcasts

Web: karthikvaidhyanathan.com
Email: karthik.vaidhyanathan@iiit.ac.in
Twitter: @karthi_ishere

SERC
Software Engineering Research Centre

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD