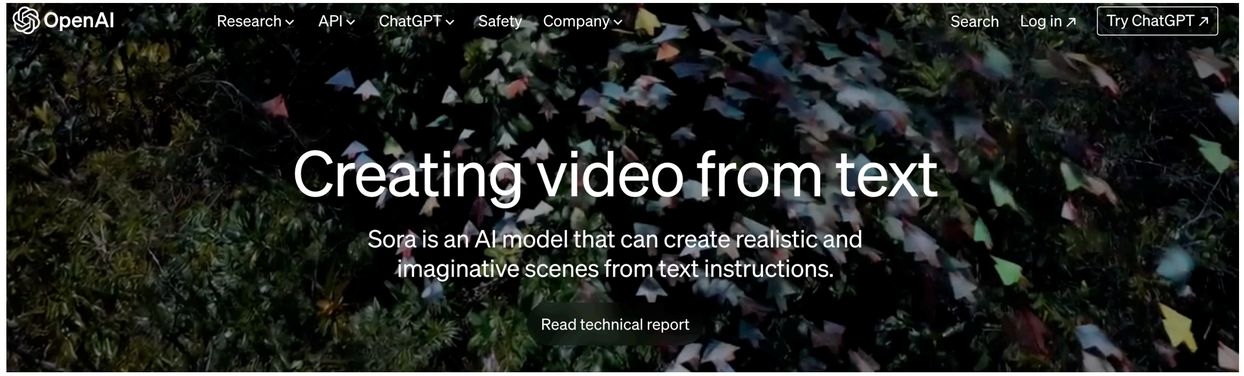
GenAI Everywhere!



Cognition

About us Blog

March 12th, 2024 | Written by Scott Wu

Introducing Devin, the first AI software engineer

And setting a new state of the art on the SWE-bench coding benchmark

Meet Devin, the world's first fully autonomous AI software engineer.

Devin is a tireless, skilled teammate, equally ready to build alongside you or independently complete tasks for you to review.

With Devin, engineers can focus on more interesting problems and engineering teams can strive for more ambitious goals.

FORBES > INNOVATION > ENTERPRISE TECH

6 Ways Generative AI Will Transform Healthcare

Bernard Marr Contributor ⓘ

Follow

0

Feb 21, 2024, 02:05am EST

Even the car that dropped me ;)

GenAI for Software Engineering: Perspectives from Research to Practice

Dr. Karthik Vaidhyanathan

NLP and LLM Course, IIIT Kottayam

Oct 26, 2024

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

ML4SA

1. ML for continuous self-adaptation
2. Gen AI for Architectural Knowlege

SA4ML

1. Sustainable ML-enabled systems
2. Autonomous ML-ops

Education

Double Master Degree - Software Architecture and Machine Learning

PhD from GSSI, Italy

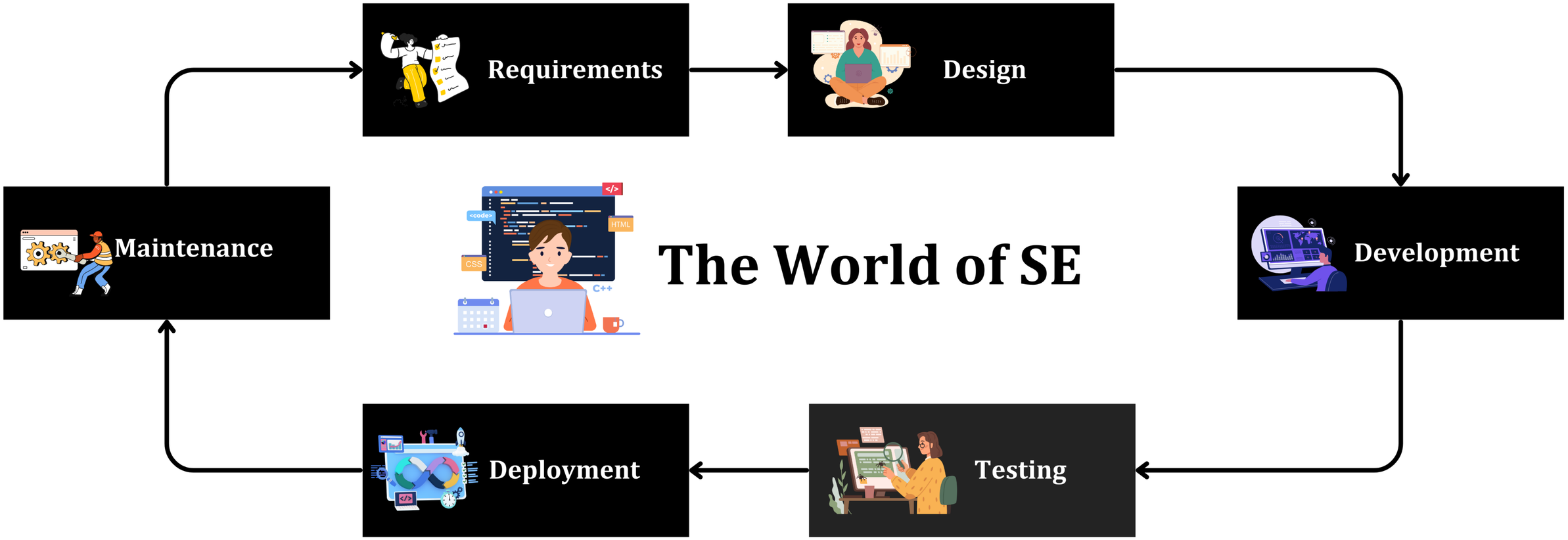
Postdoc, University of L'Aquila, Italy

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

The World of Software Engineering



Starting with Software Design!

“Abstraction is the art of painting with thoughts, not brushes”

You

How does abstraction work? Can you help me create an abstract representation of an internet banking system?

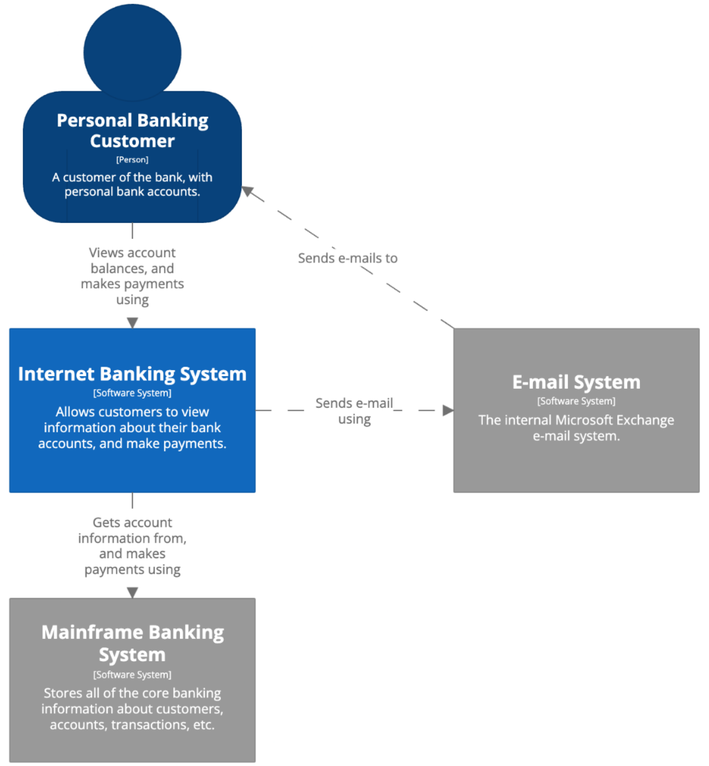
ChatGPT

Abstraction is a fundamental concept in software engineering and design, which involves hiding the complex reality while exposing only the necessary parts. It allows developers to focus on interactions at a higher level without being bogged down by the details of lower-level implementations. In the context of an internet banking system, abstraction helps in managing the system's complexity by breaking it down into more manageable, discrete components.

Creating an Abstract Representation of an Internet Banking System

To abstractly represent an internet banking system, we can divide it into several key components or layers. Each layer focuses on a specific aspect of the system, providing services to the layer above it and abstracting the details of the layer below it.

1. **Presentation Layer (User Interface)**

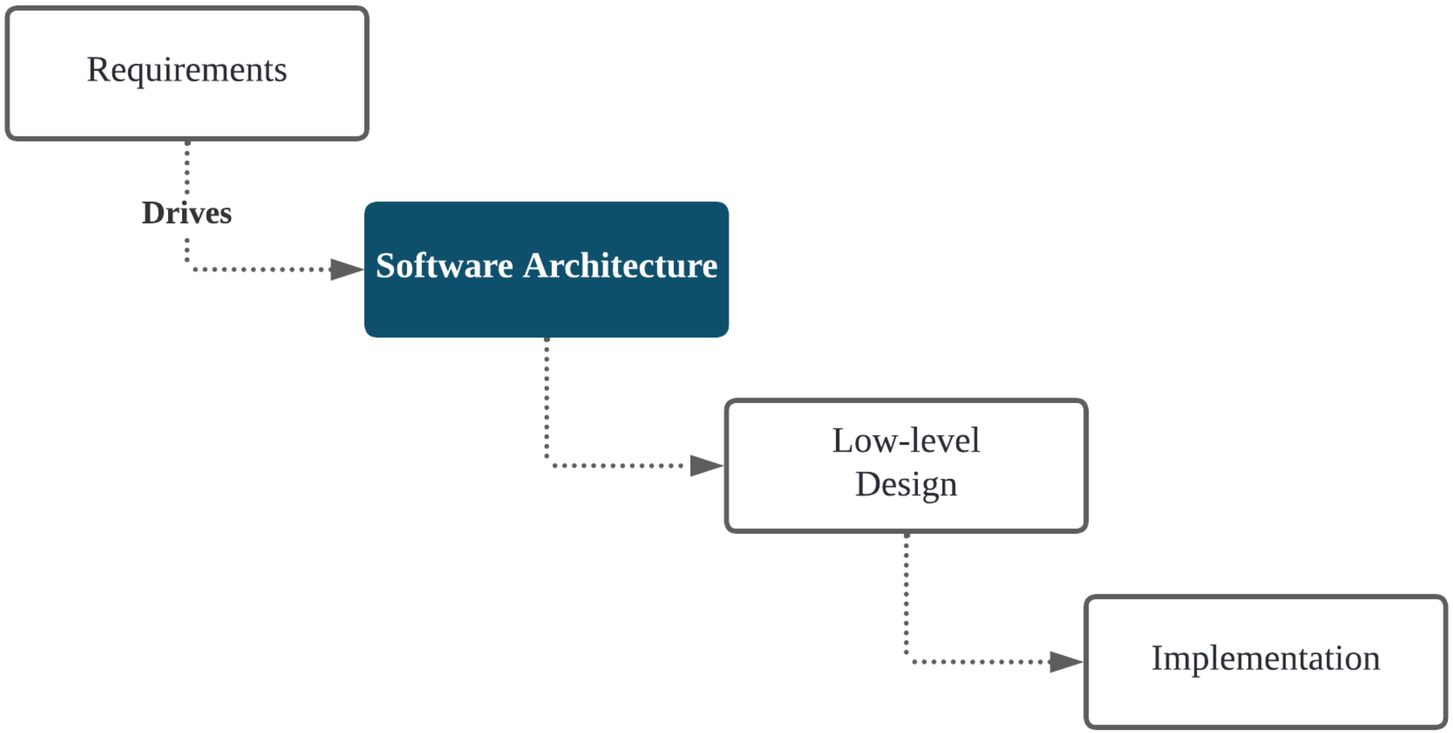


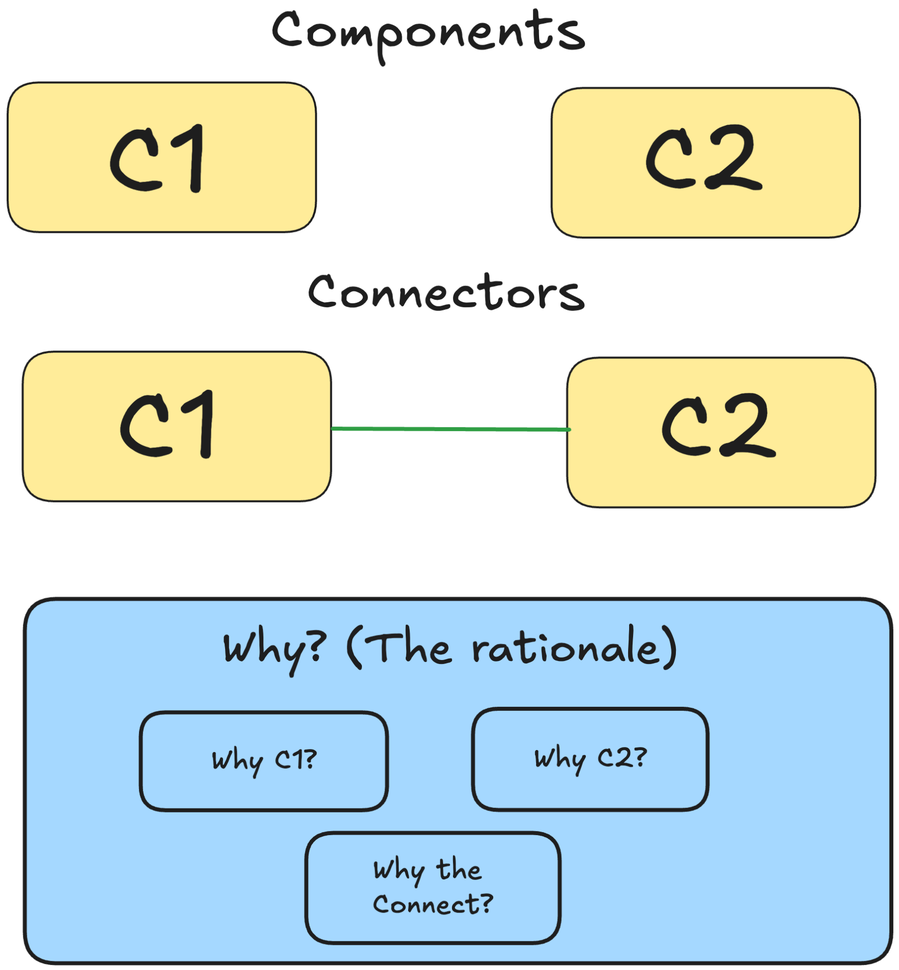
DALL·E

This image is a system context diagram for an Internet Banking System. It outlines the interactions between a personal banking customer and the system components. The customer uses the internet banking system to view account balances, make payments, and also sends emails. The internet banking system, in turn, interacts with a mainframe banking system that stores core banking data, and it uses an email system (specifically mentioned as Microsoft Exchange) for sending emails. The arrows indicate the direction of data flow or interaction between the different entities.

Diagram source: c4model.com

What is Software Architecture?





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

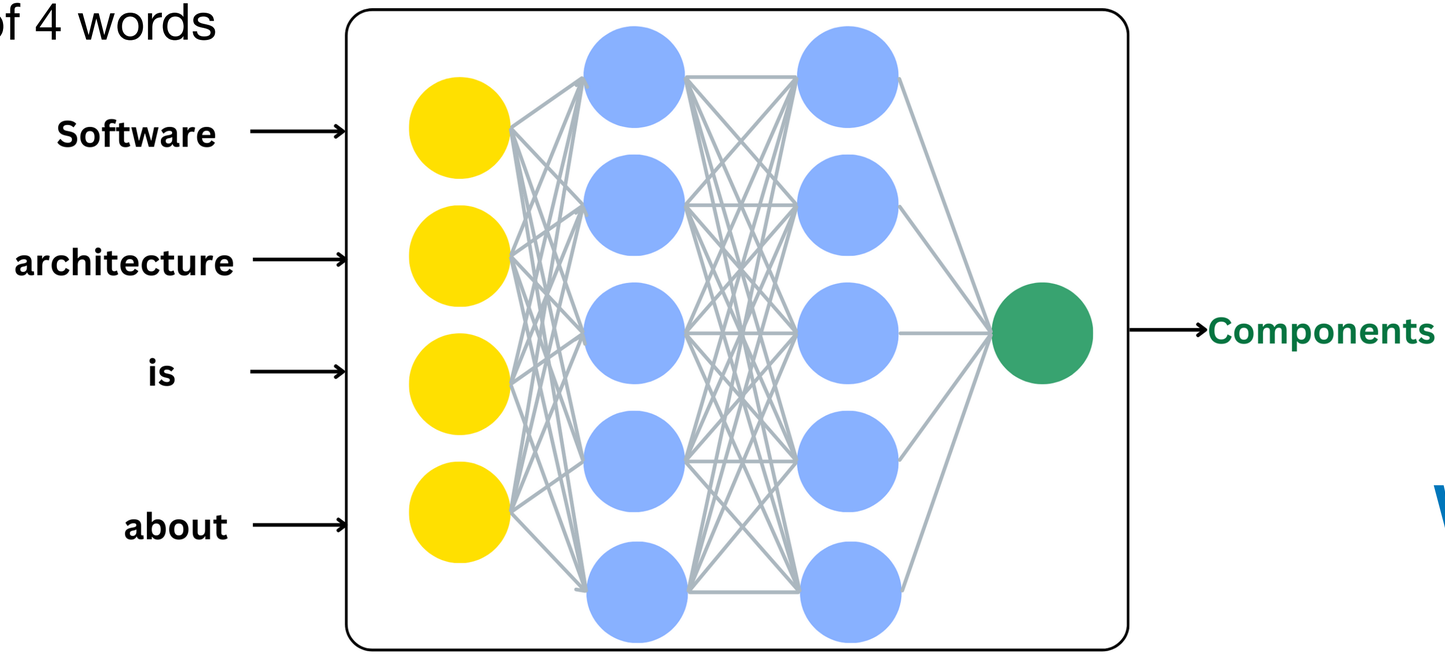
https://doi.org/10.1016/j.jss.2015.08.054 Get rights and content

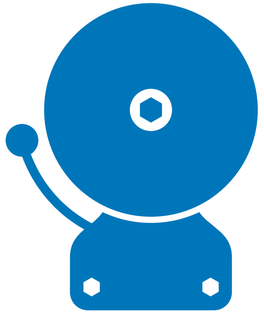
Need for better tools => Automate using **ArchBots or a co-pilot**

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?

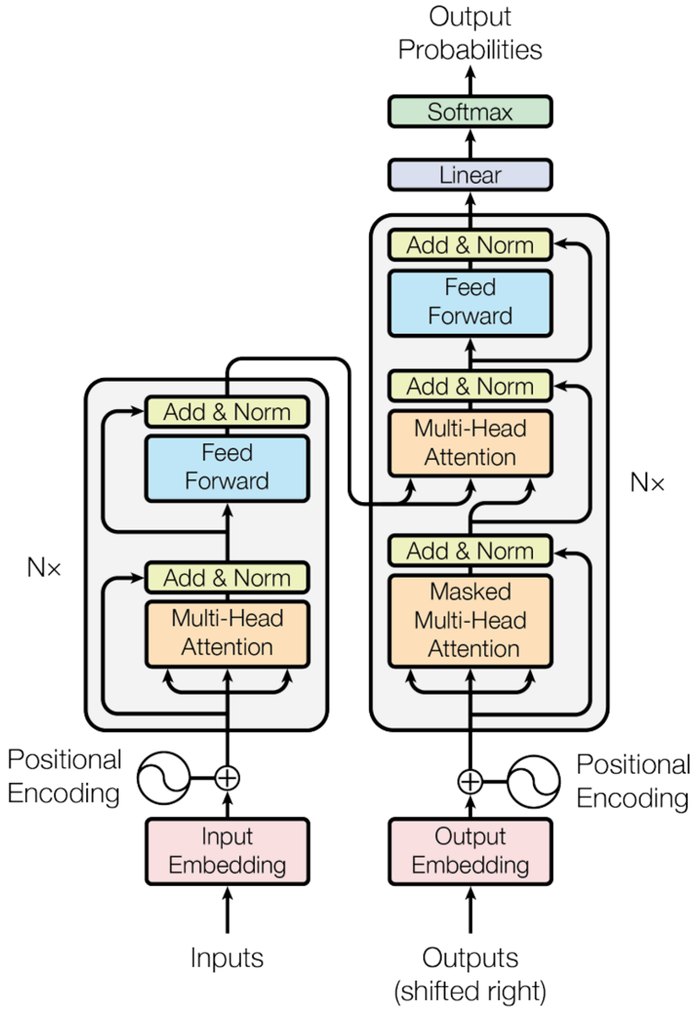
Software architecture is about components and

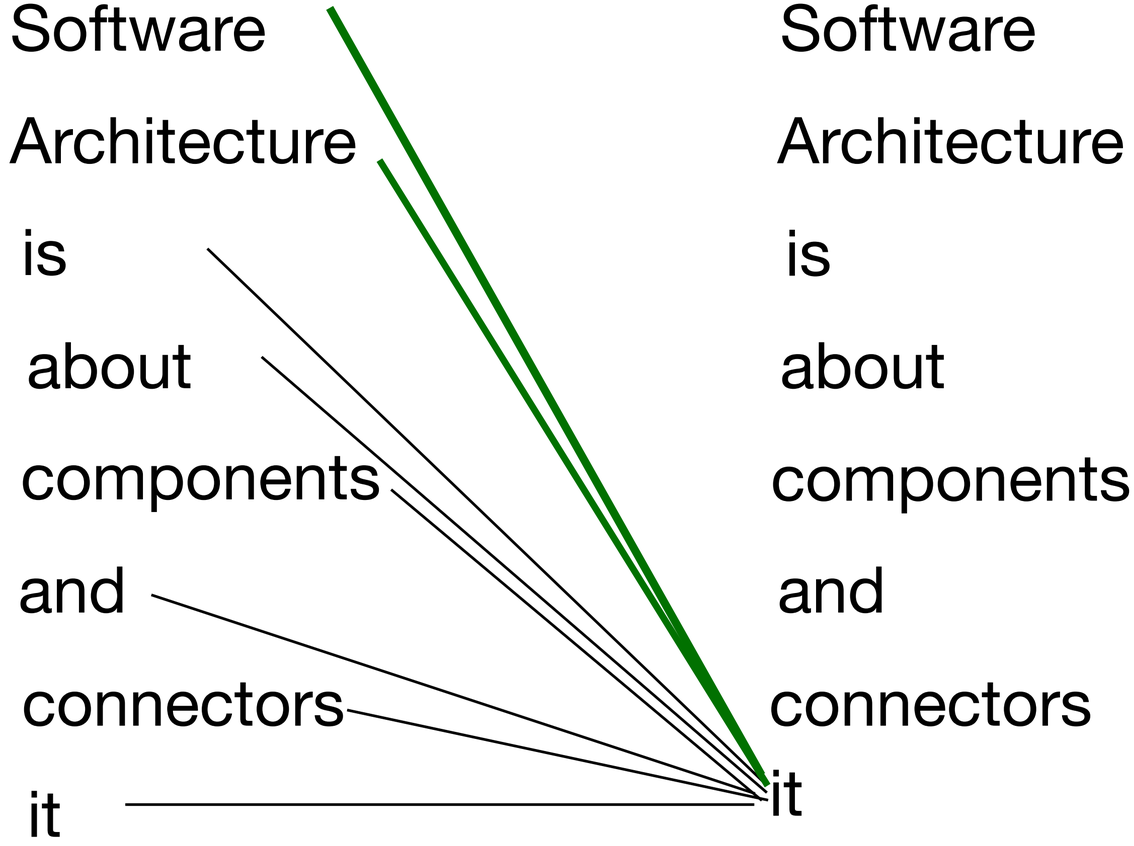
Software architecture is about components and connectors.

Software architecture is about components and connectors. It

Attention is all you need!

Age of Transformers



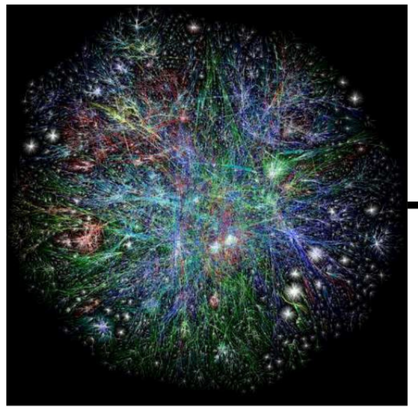


Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. **Attention is all you need.** *Advances in neural information processing systems*, *30*.

“Large” Language Models (LLM)

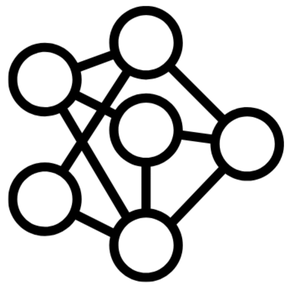
Do you have a ton of text and compute power?

Internet



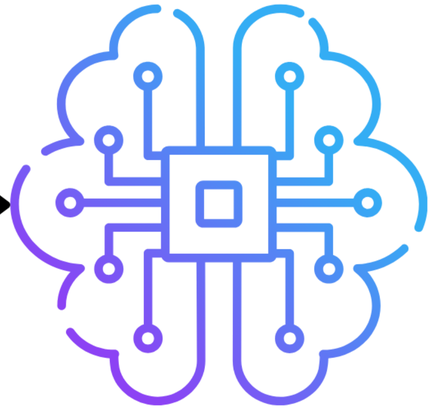
~10B of text from internet

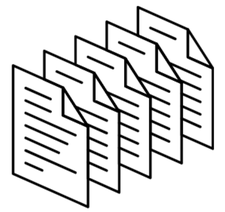
Compute



Transformer NN
trained on 1000s of GPUs
for days

Foundation model/
Base model



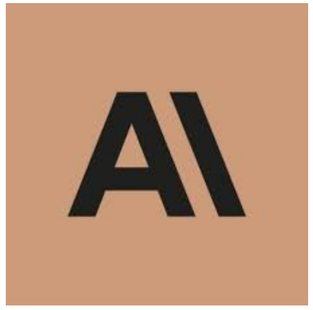


Model parameters
~some billions
100s of GB

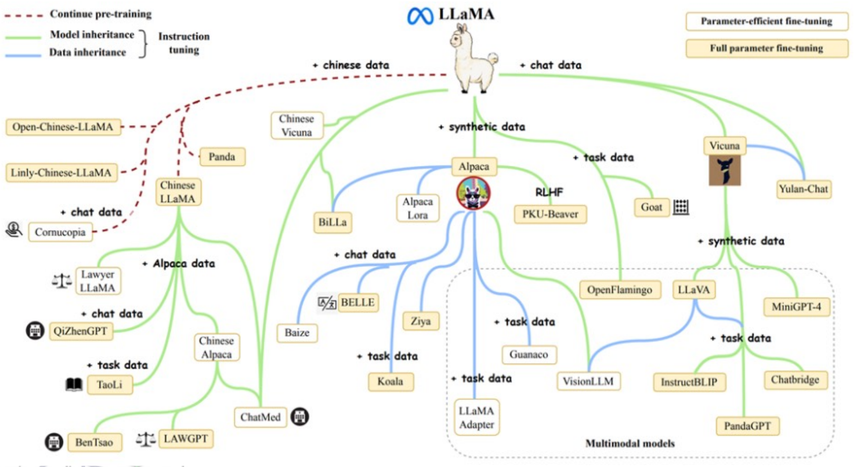
Inspiration from slides of Andrej Karpathy, Introduction to large language models

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.

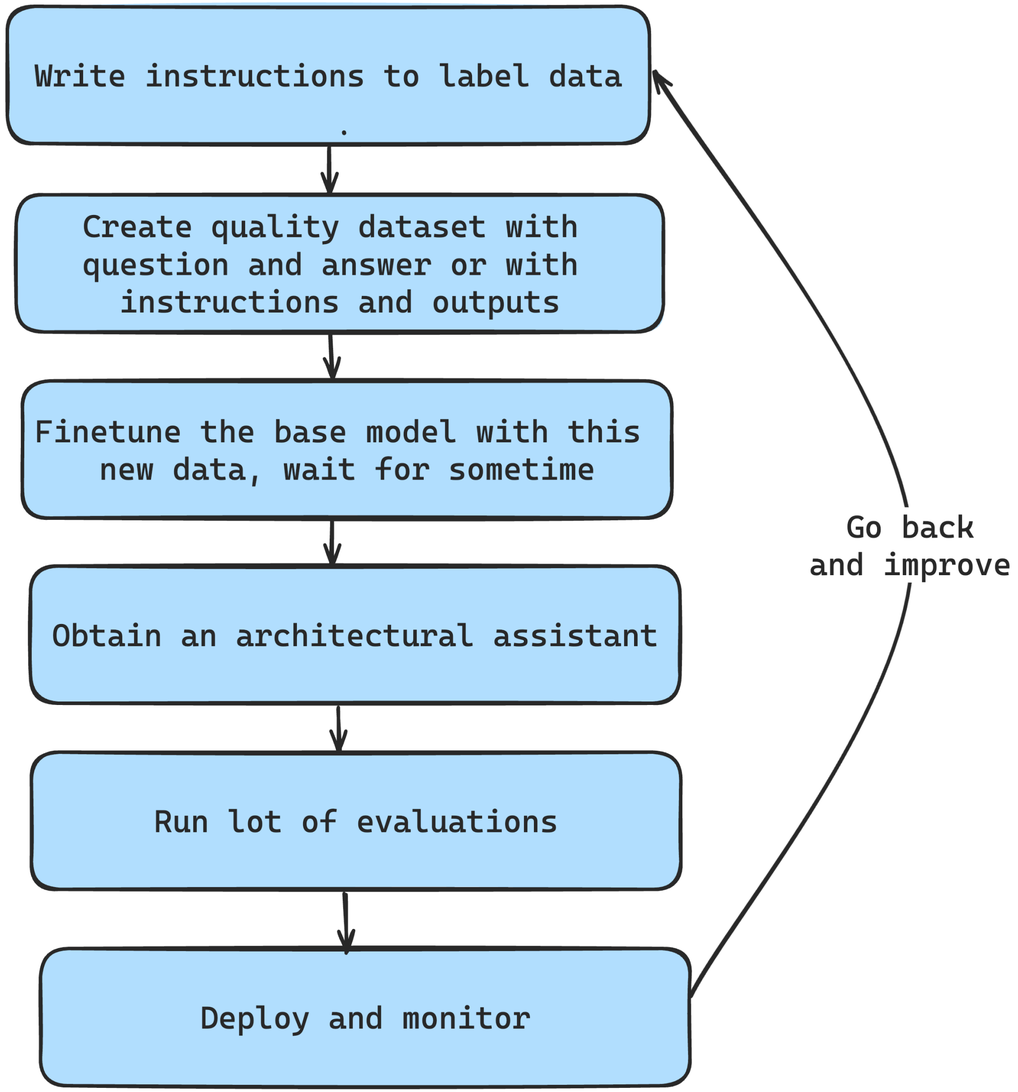
Contribute your vote at chat.lmsys.org! Find more analysis in the notebook.

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/

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,…

How can all these help Software Architect?

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

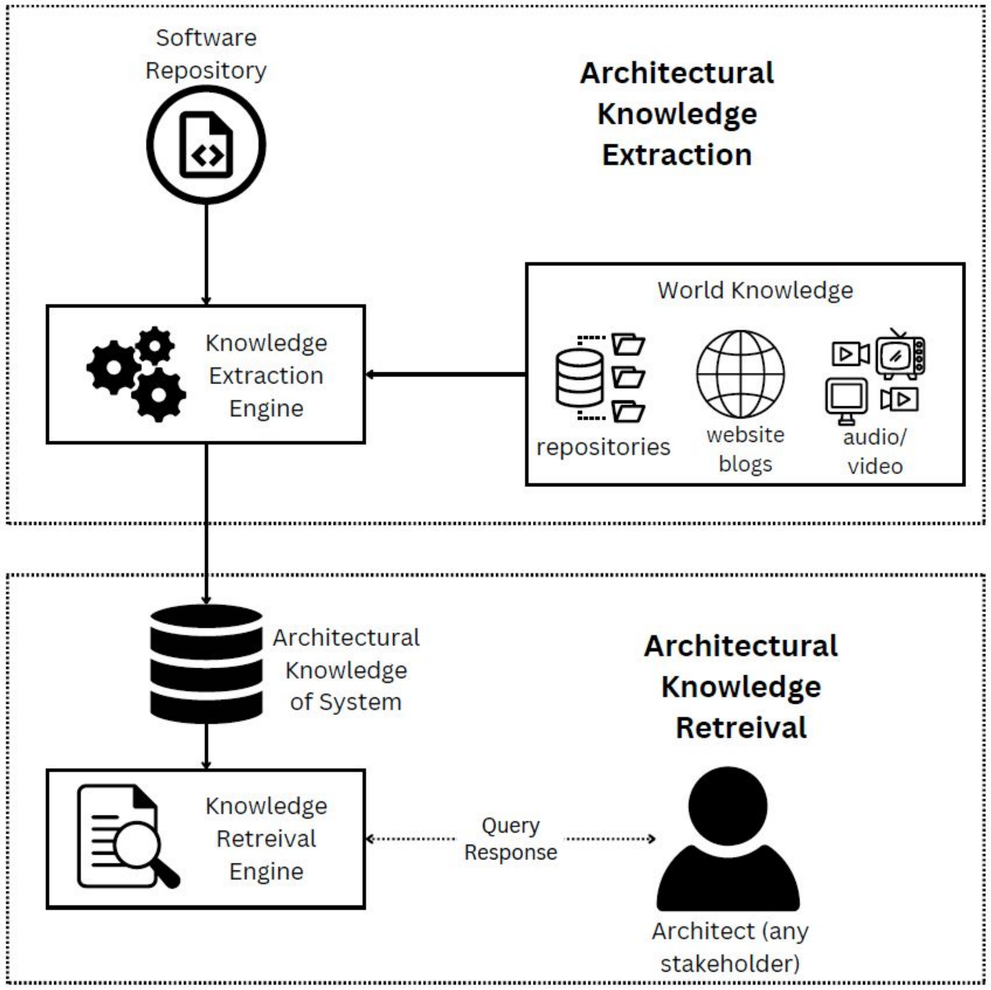
Software architectures have high costs for change, are complex, and erode during evolution. We believe these problems are partially due to knowledge vaporization. Currently, almost all the knowledge and information about the design decisions the architecture is based on are implicitly embedded in the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

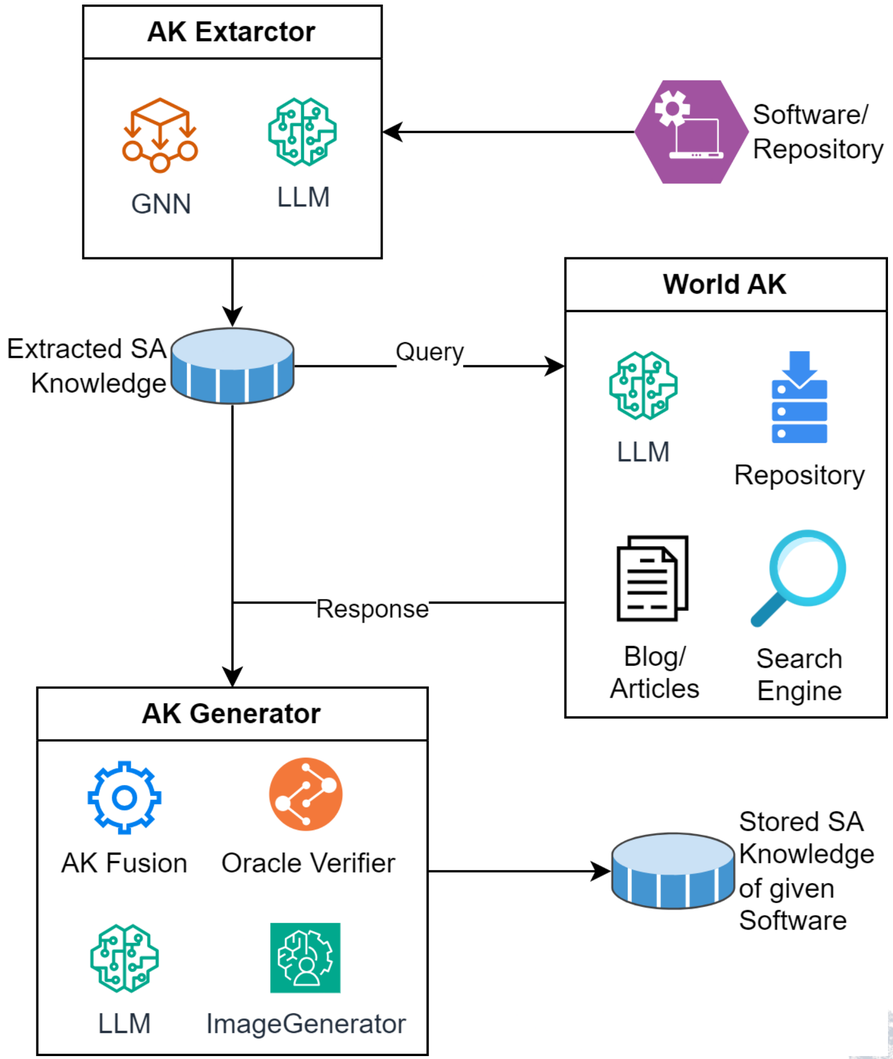
this notion of architectural design decisions, although architectural design decisions play a crucial role in software architecture, e.g. during design, development, evolution, reuse and integration of software architectures. In design, the main concern is which design decision to make. In development, 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)*

Generative AI for Architectural Knowledge Management





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

Starting with Design Decisions

- **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?**

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

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

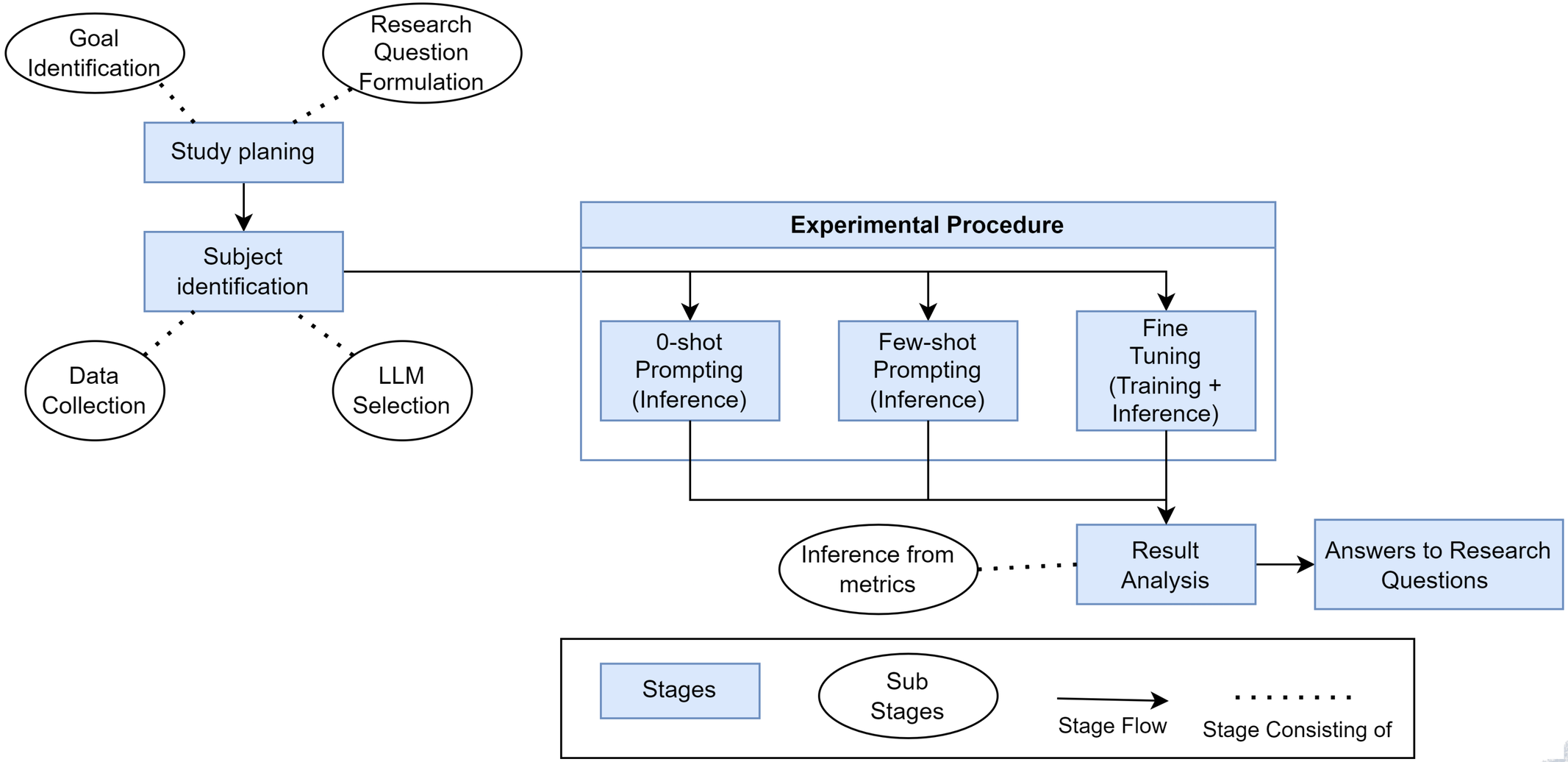
Consequences
1. Functionality needs to be exposed as API for integration support

Karthik Vaidyanathan

Three Key Research Questions

- **RQ1:** Can LLMs be successfully employed to generate architecture design decisions from a given context in a zeroshot setting?
- **RQ2:** Does few-shot approach affect or improve a LLM’s ability to generate Design Decisions?
- **RQ3:** Does Fine-tunning LLM enhance its capability of generating architectural Design Decisions based on a provided context?

Performing an Exploratory Study



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

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

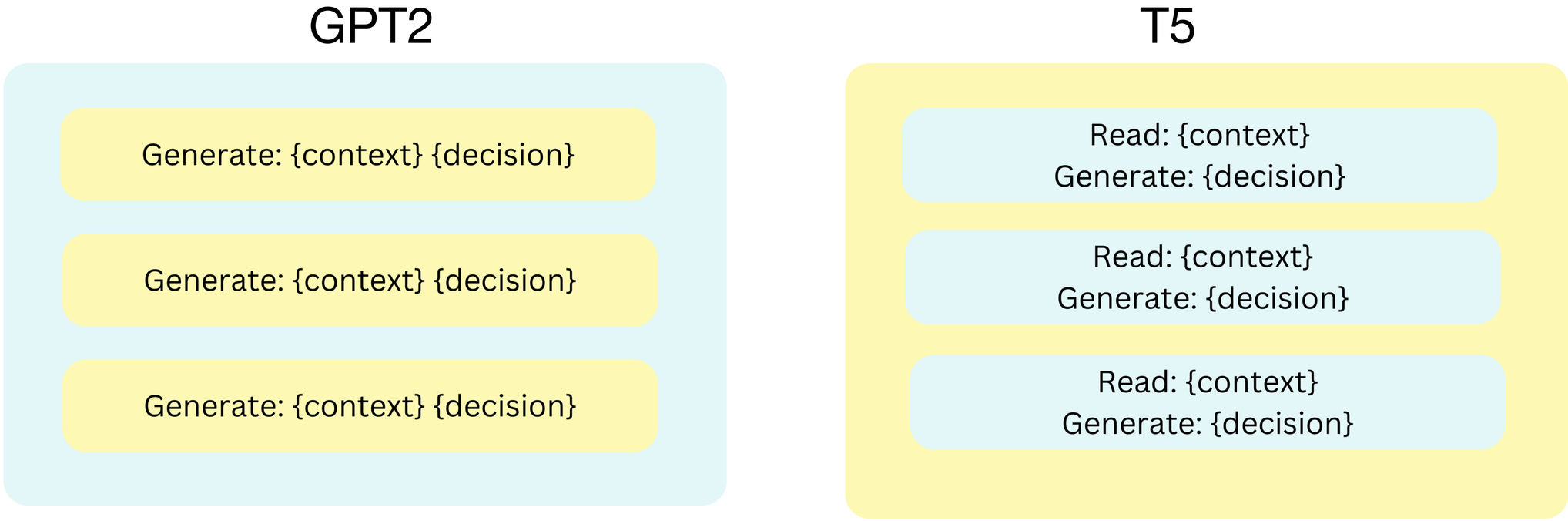
After evaluating the context of our project, we have decided to use Python as the programming language.

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



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!

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

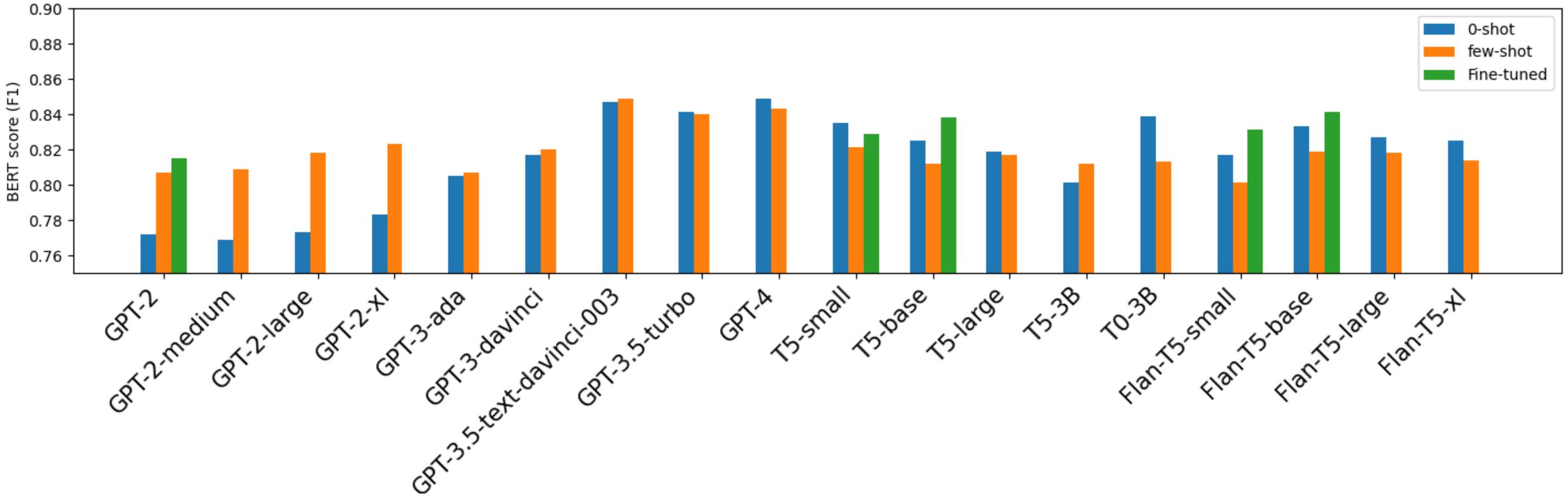
Python is also a great choice in terms of **performance and scalability**…., Python is an interpreted language..easier to debug and **maintain** than compiled languages.

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

The Overall Results

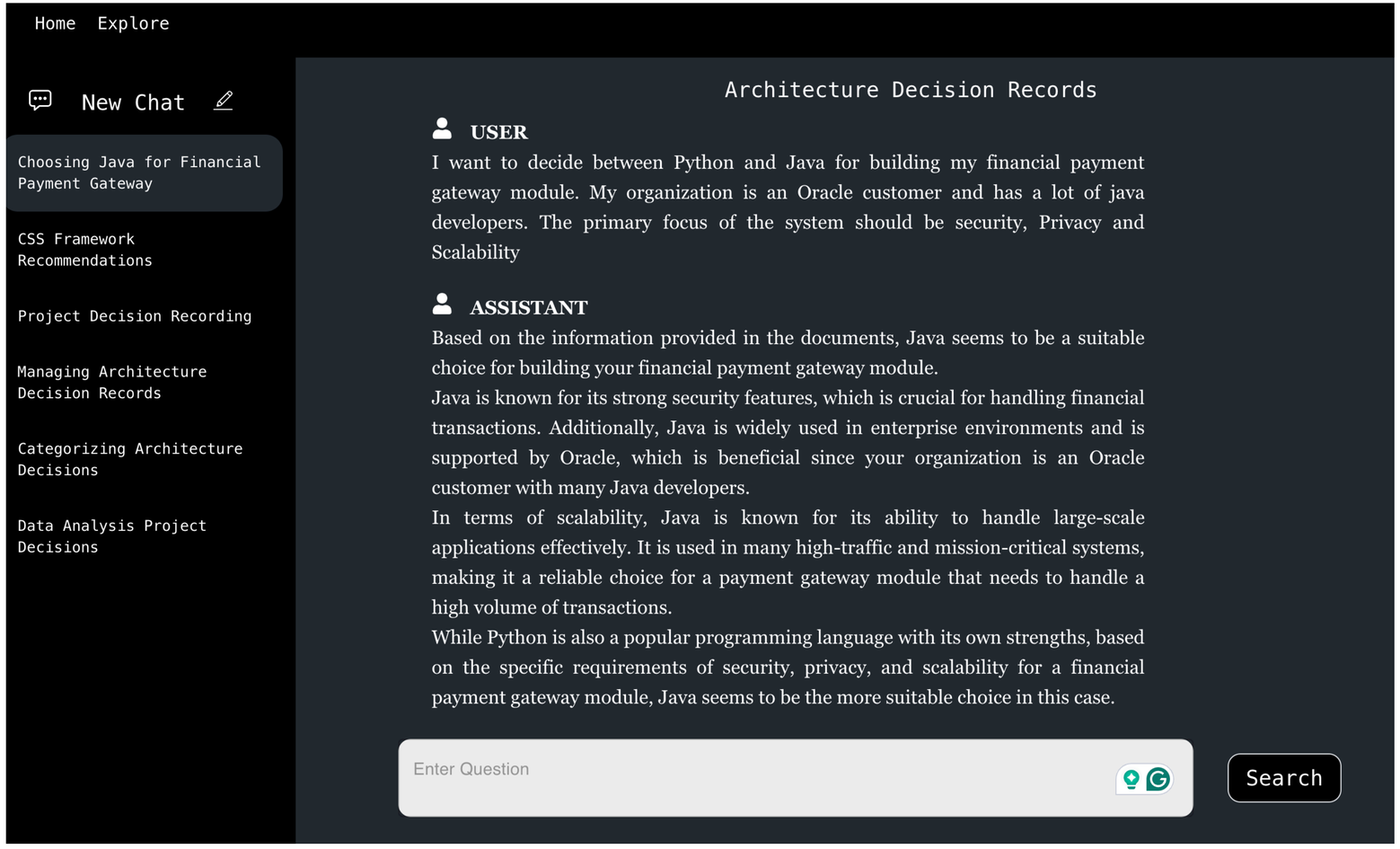


There is a need for more data and definitely more context!

Key Observations

- **RQ1: Generating design decisions in 0-shot approach**
  - While LLMs can generate design decisions using 0-shot approach, the quality does not reach human level
- **RQ2: Impact of few-shot approach in generating design decisions**
  - The overall performance of LLMs in few shot remains inconclusive and lacks generalisation
- **RQ3: Impact of fine-tuning in Generating Design Decisions**
  - Fine tuned LLMs exhibit improved capability
  - Compact fine-tuned models demonstrate results comparable with proprietary LLMs

Introducing the ArchCopilot



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

Extending Beyond Design: Introducing Self-adaptation

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.

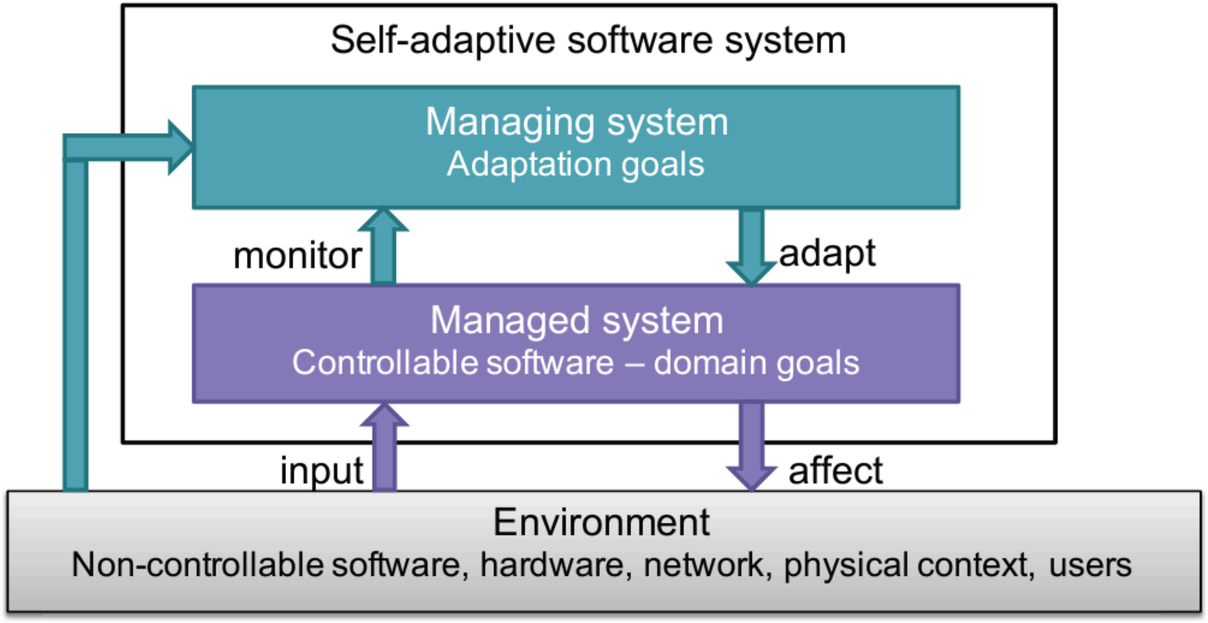
Jeffrey O. Kephart
David M. Chess
IBM Thomas J. Watson Research Center

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

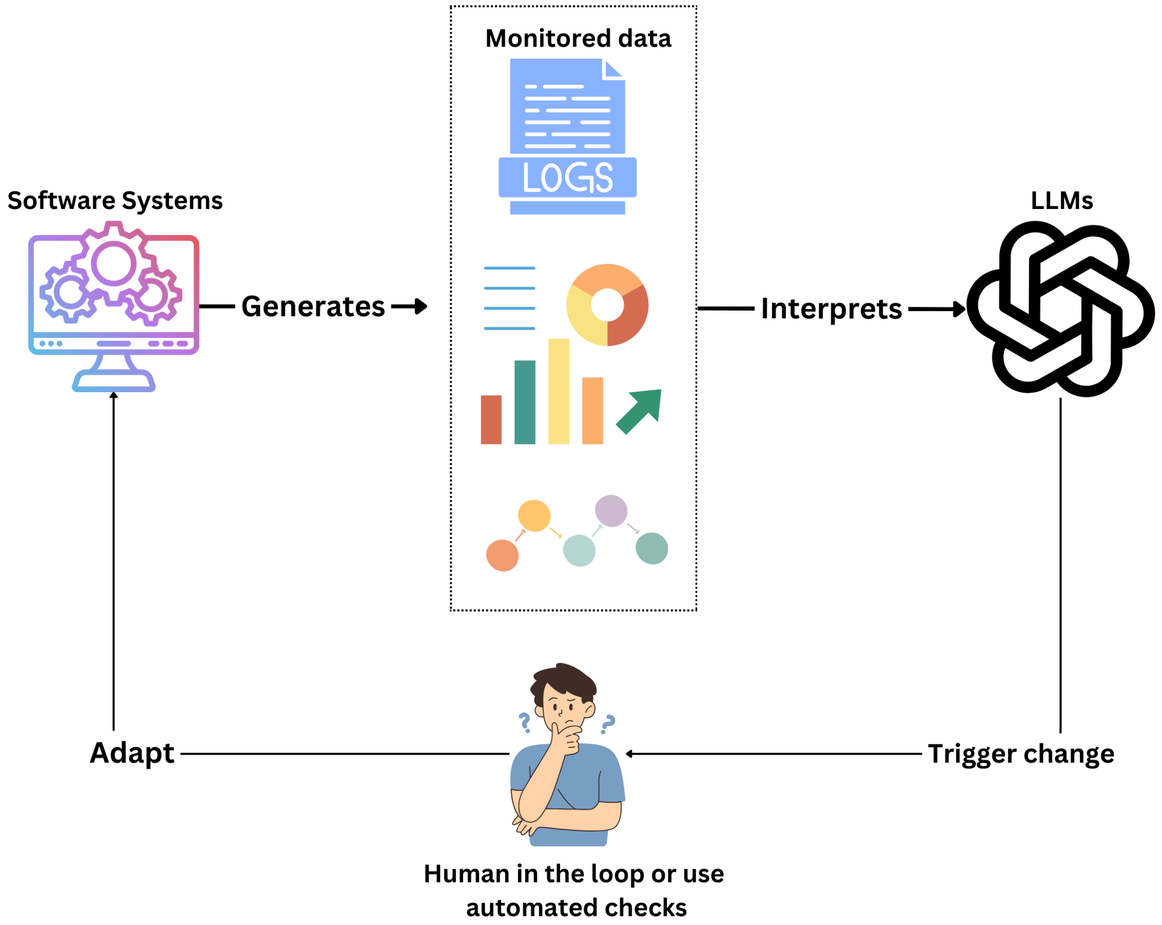
The only option remaining is *autonomic computing*—computing systems that can manage themselves given high-level objectives from administrators. When IBM's senior vice president of research, Paul Horn, introduced this idea to the National Academy of Engineers at Harvard University in a March 2001 keynote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

On to Software Maintenance

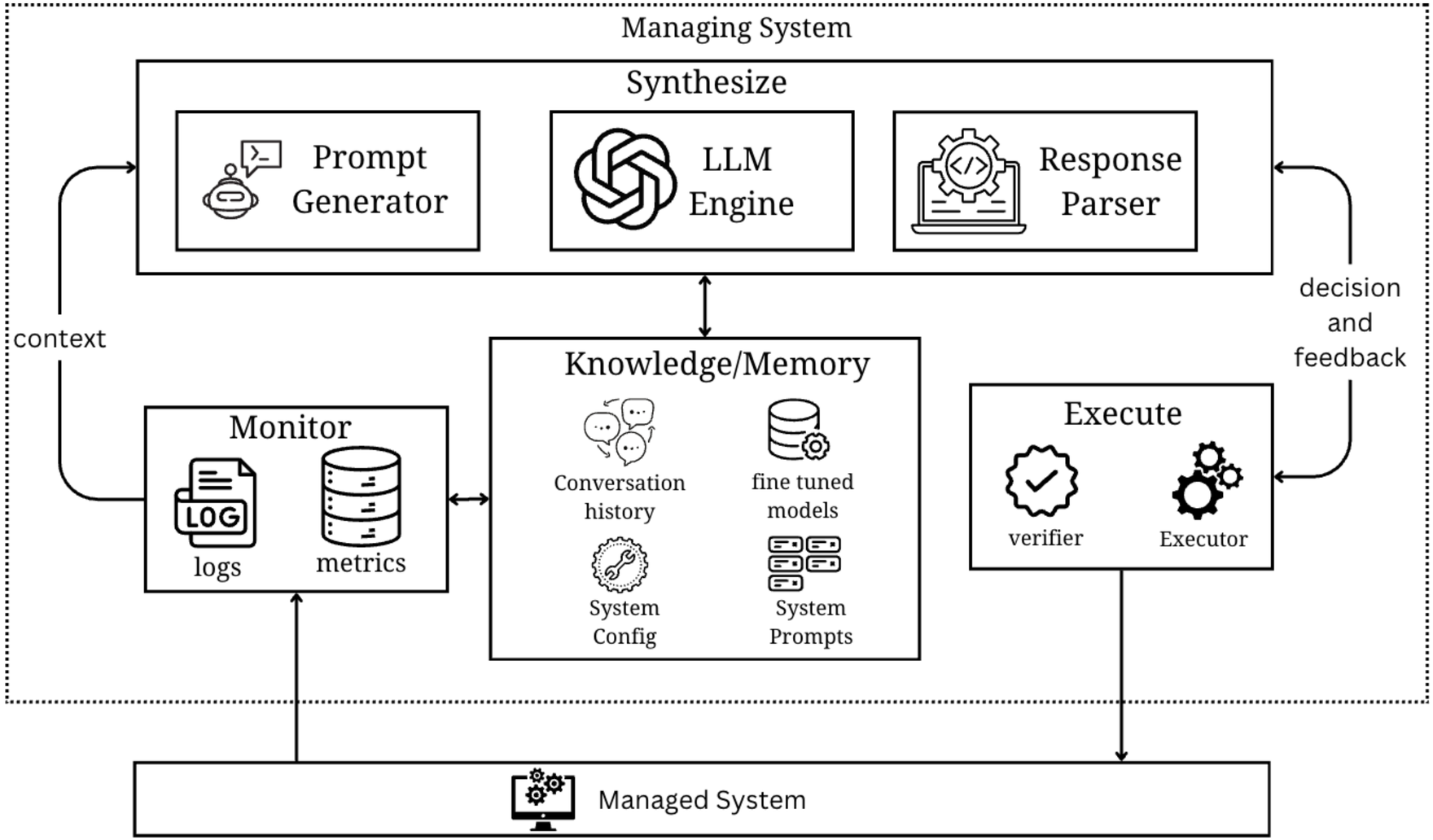
Design time to run-time adaptation - Can LLMs help?



>70% of the cost goes for maintenance

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

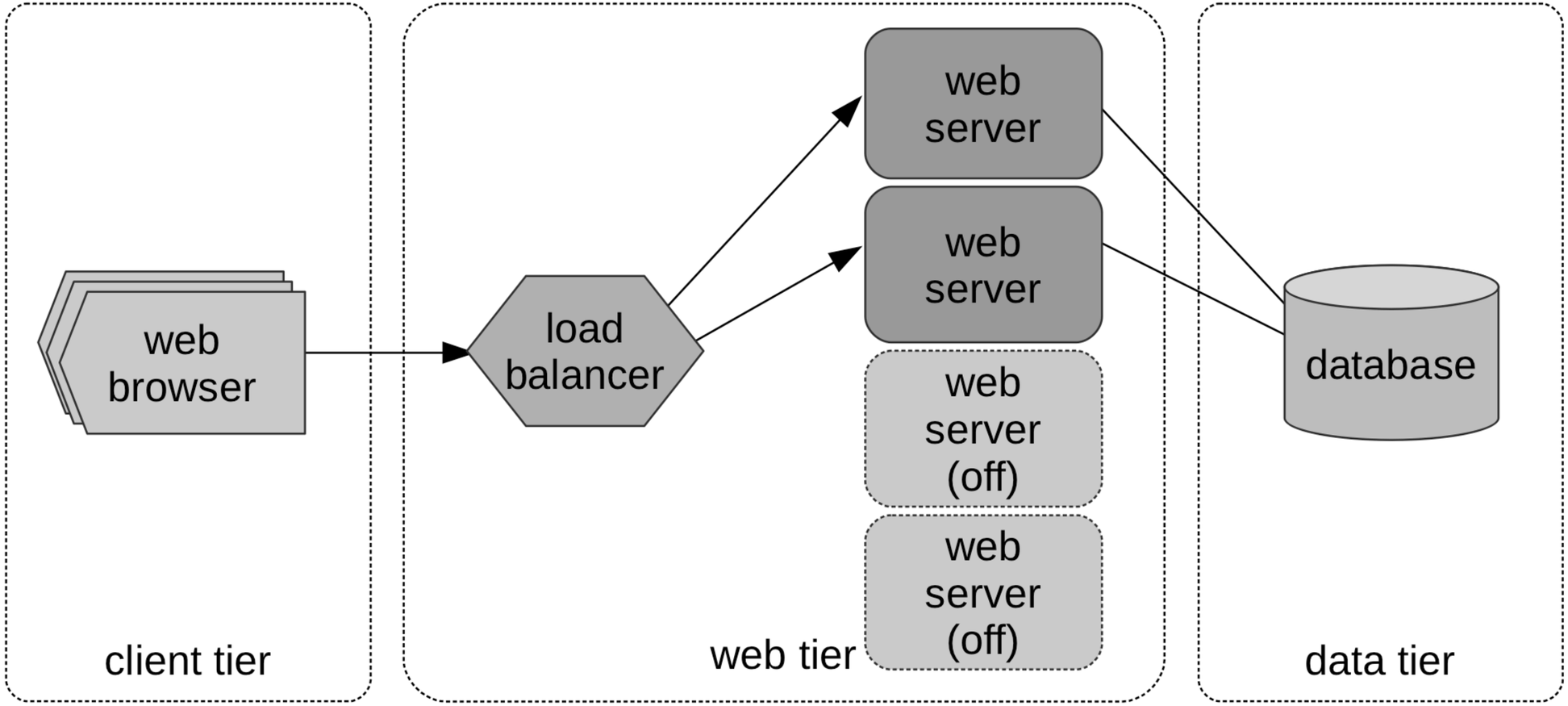
Reimagining Self-adaptation loop



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

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,

Prompts used for SWIM

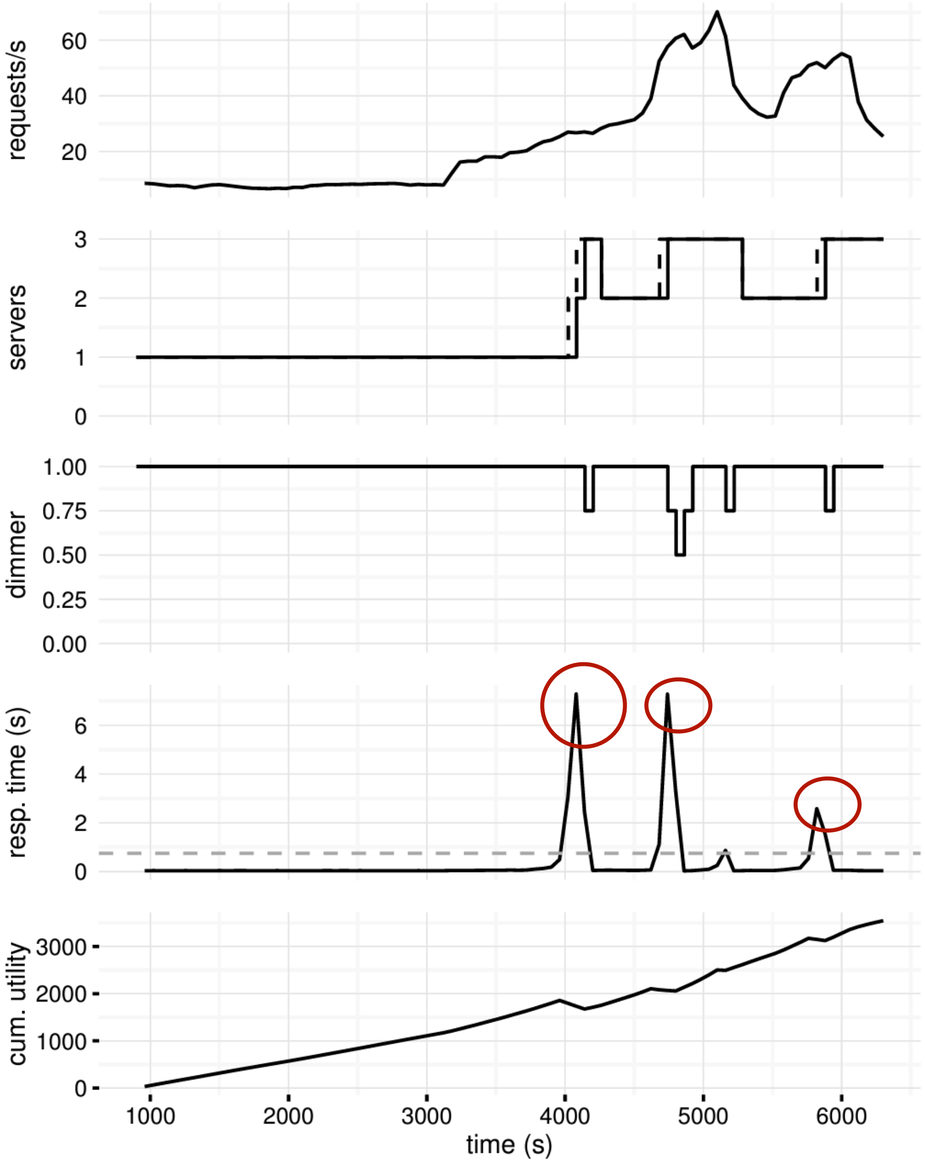
P_{SWIM}

You are an adaptation manager for a server system handling user requests. Self-adaptive systems are...O...You being the adaptation manager are responsible for modifying the runtime behavior of the system...decide if adaption is required and what type of adaptation is needed. *Terminologies* Here are a few examples on how you are to interact with the system...*Few Shot*

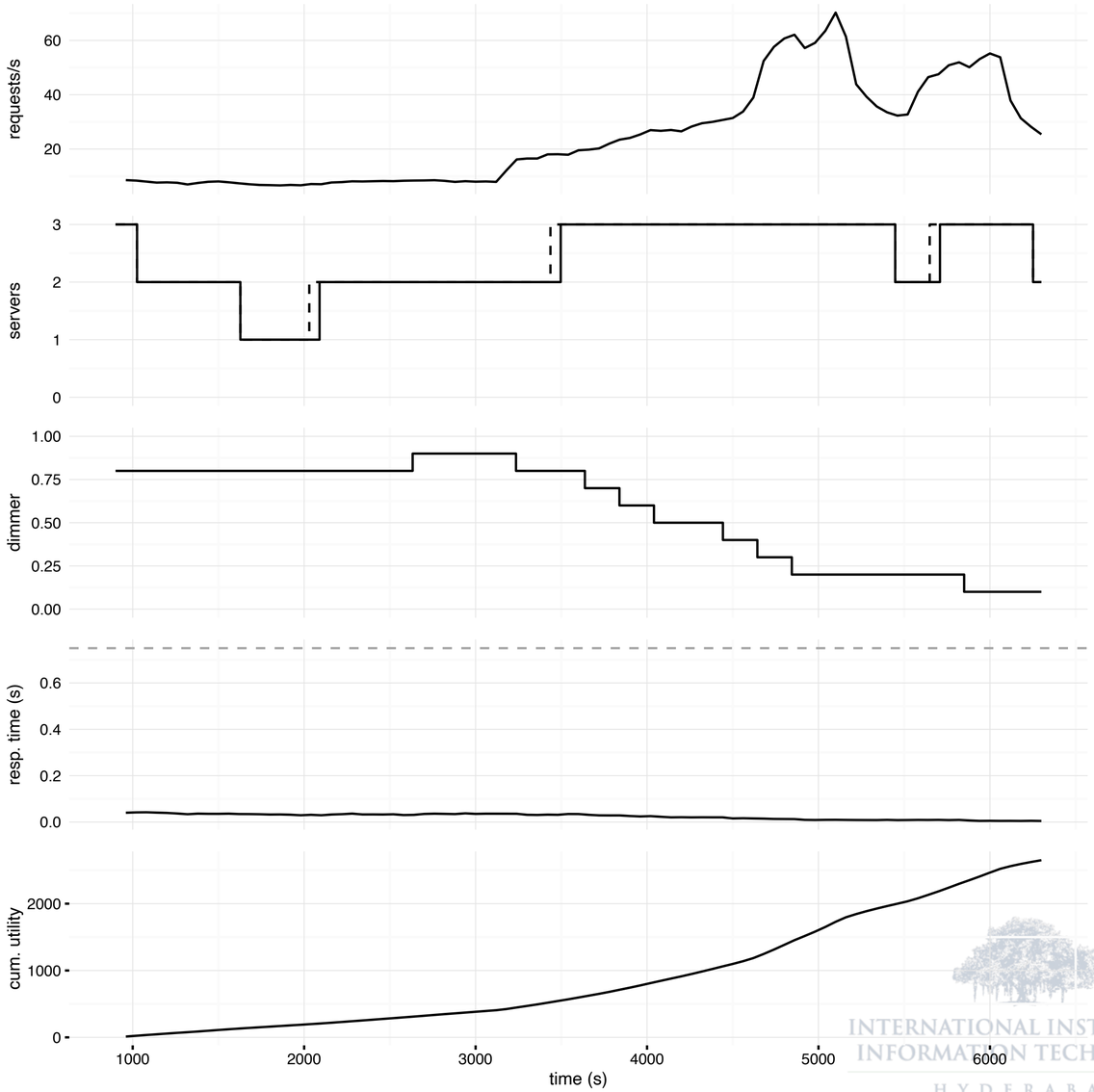
- O: Here the primary system objective is to keep the average response time as low as possible. Secondary objective is to keep dimmer as high as possible. Tertiary objective is to keep servers used as less as possible. These objectives are given in decreasing order of priority.
- *Terminologies*: Brief description of metrics in C. Metrics are dimmer,active servers, max servers, utilization, average response time, arrival rate, and time elapsed.
- *Few Shot*: $\{(C_{s1}, AD_{s1})...\}$

Some Initial Results

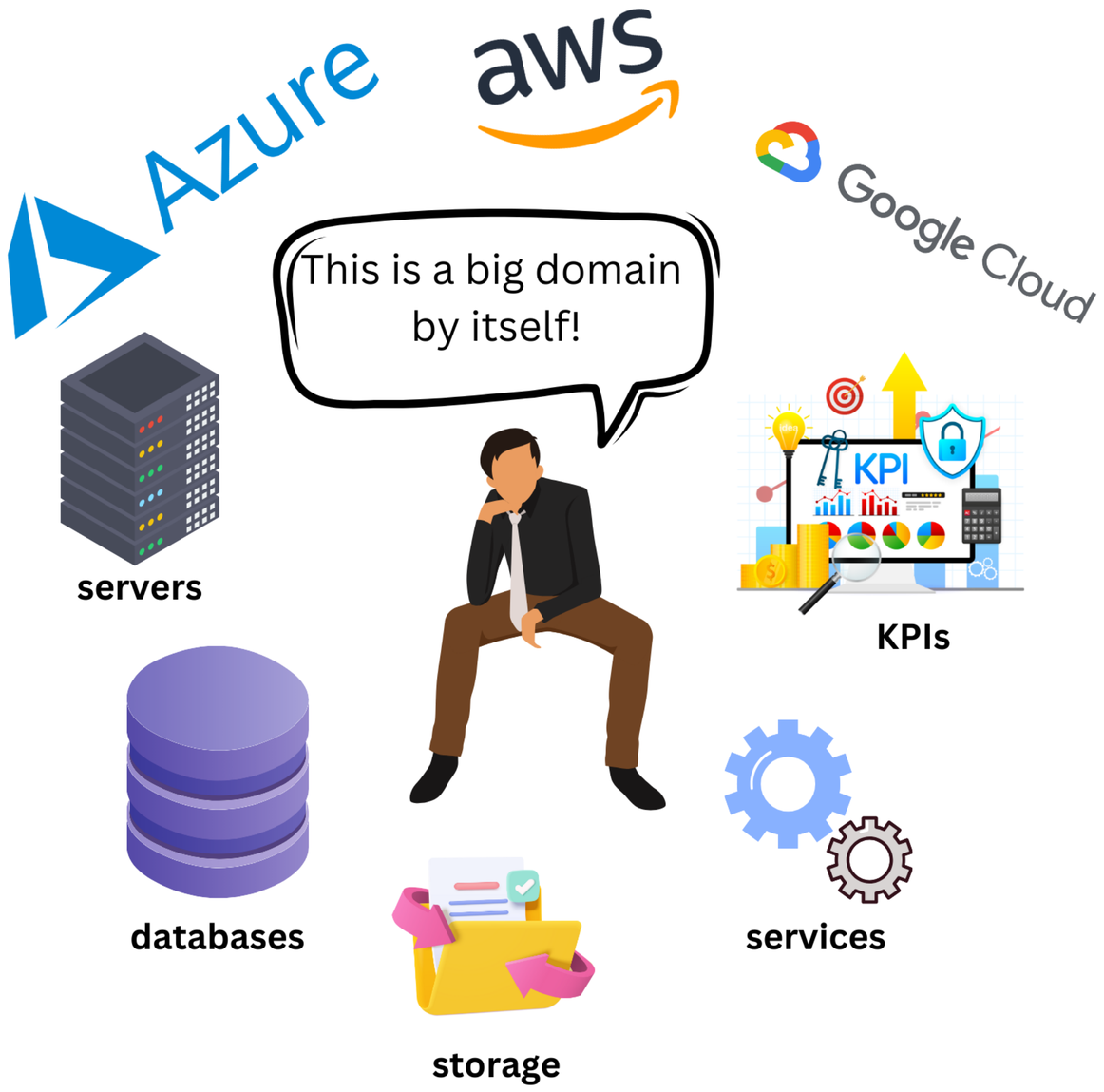
Using SWIM reactive adaptation



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

CloudOps Copilot

- Conversational support for cloud SMEs and architects
- Autonomously manage cloud footprint
  - Provision or decommission resources (human-in the loop)
  - Generate tickets or raise alerts
  - Provide insights on the large data captured
  - Perform regular mundane tasks

No-Code Cloudops Company founded in 2018

https://montycloud.com/cloudops-copilot

MontyCloud

CloudOps Copilot | Douglas Adams | Acme Inc

Dashboard | CloudOps Copilot

- Home
- Projects

ASSESSMENTS
- Well-Architected
- Foundational Best Practices

INVENTORY
- Servers
- All Resources

GOVERNANCE
- Tags and MAP Projects
- Policies
- Service Catalog

DAY2 CLOUDOPS
- Cost
- Security
- Compliance
- Automation

- Reports

Last Updated a few seconds ago

Open Ops Issues
190

Remediations
80
By DAY2™

Recommendations
25 Security
15 Compliance

Security Posture
Last Run 21-Nov-23 09:41
150 Open Issues

Security Bot
ACTIVE

TOP 3 VIOLATIONS

S3 Buckets should have a bucket policy configured	10
RDS instances should have encrypted storage	8
IAM users should not have attached in-line policies	8

Compliance Assessment
Last Run 21-Nov-23 09:59
40 Open Issues

Compliance Bot
ACTIVE

TOP 3 VIOLATIONS

S3 Account Level Public Access Blocks	12
IAM Root user access key check.	9
EBS Volumes should be encrypted	8

Industry Standards

HIPAA	CIS	FedRAMP	NIST	PCI
83%	67%	91%	83%	100%

AWS Costs
Total Spend
$13,798.33

Cost by View

$1456.27	$400.22
HR Department	Dev Resources

Top Services

$9,081.00	$4,081.00	$3,234.00
Instance	Volume	Snapshots

Cost Optimization
LAST RUN 21 Nov 2023 10:13 AM

$1425
Potential Cost Savings

0 Over Provisioned	0 Under Provisioned
64 Abandoned Resources	0 Needs Optimization

Cloud Footprint
3399 Total Resources
+ 0 New resources added last week

Top Resources
3005 Compute Instance
200 Image
35 EBS Volume
30 VPC Endpoint
30 Virtual Private Cloud
20 EBS Snapshot
15 SNS Topic

Moving towards Development: A study on energy efficiency and LLMs

A Controlled Experiment on the Energy Efficiency of the Source Code Generated by Code Llama

Vlad-Andrei Cursaru
Vrije Universiteit Amsterdam
The Netherlands
v.cursaru@vu.nl

Laura Duits
Vrije Universiteit Amsterdam
The Netherlands
l.b.m.duits@student.vu.nl

Joel Milligan
Vrije Universiteit Amsterdam
The Netherlands
j.a.milligan@student.vu.nl

Damla Ural
Vrije Universiteit Amsterdam
The Netherlands
d.ural@vu.nl

Berta Rodriguez Sanchez
Vrije Universiteit Amsterdam
The Netherlands
b.rodriguezsanchez@student.vu.nl

Vincenzo Stoico
Vrije Universiteit Amsterdam
The Netherlands
v.stoico@vu.nl

Ivano Malavolta
Vrije Universiteit Amsterdam
The Netherlands
i.malavolta@vu.nl

***Abstract*—*Context*. Nowadays, 83% of software developers use Large Language Models (LLMs) to generate code. LLMs recently became essential to increase the productivity of software developers and decrease the time and cost of software development. Developers ranging from novices to experts use LLM tools not only to detect and patch bugs, but also to integrate generated code into their software. However, as of today there is no objective assessment of the energy efficiency of the source code generated by LLM tools. Released in August 2023, Code Llama is one of the most recent LLM tools.**

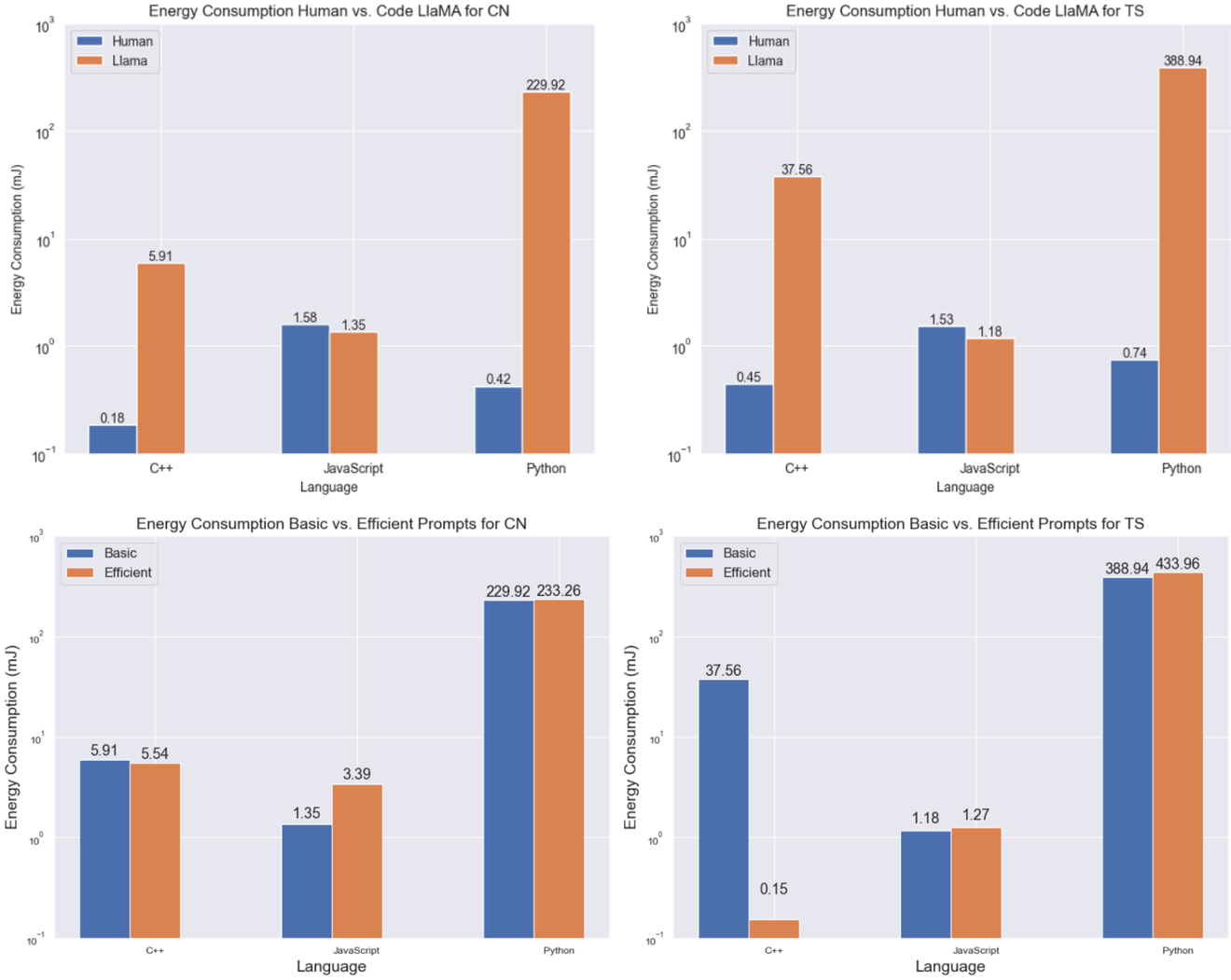
***Goal*. In this paper, we present an empirical study that assesses the energy efficiency of Code Llama with respect to human-written source code.**

***Method*. We design an experiment involving three human-written benchmarks implemented in C++, JavaScript, and Python. We ask Code Llama to generate the code of the benchmarks using different prompts and temperatures. Therefore, we execute both implementations and profile their energy efficiency.**

***Results*. Our study shows that the energy efficiency of code generated by Code Llama is heavily-dependent on the chosen programming language and the specific code problem at hand.**

Program Repair (APR) techniques are, for instance, widely adopted to detect and provide patches for bugs [2]. The majority of developers using AI tools (*i.e.,* 83%) indicate to use these tools to generate code [1]. *Large Language Models (LLMs)* are a specific category of generative AI tool that can assist developers write code. GitHub Copilot[1] is an LLM model that generates code snippets based on the context provided by the user (*e.g.,* surrounding code and comments). Similarly, OpenAI's ChatGPT[2] is able to translate natural language to code. On August 24, 2023, Meta AI released its own LLM model: Code Llama. Code Llama is a variant of Llama 2[3], a general-purpose LLM model, obtained by training Llama 2 with code-specific datasets.

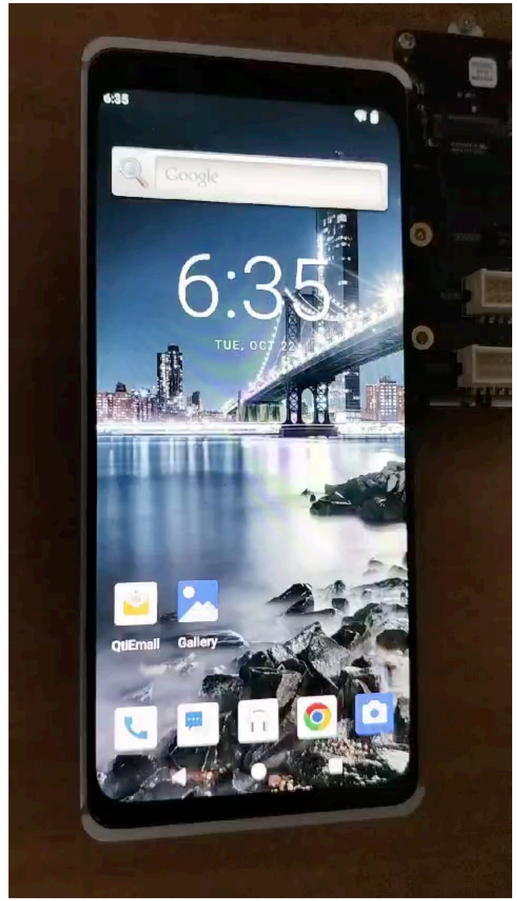
The training phase of LLaMA and Code LLaMA already produced about 1,015 tons of carbon emission (tCO2eq) [3] and 63.5 tCO2eq [4], respectively. The developers of LLaMA hope that releasing the models will reduce greenhouse gas emissions in the future because, due to their small size, they



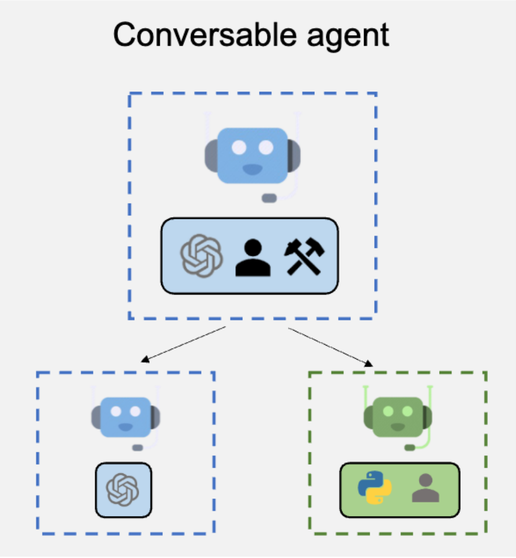
Code generated may not be energy efficient!

Into the world of SLMs for SE

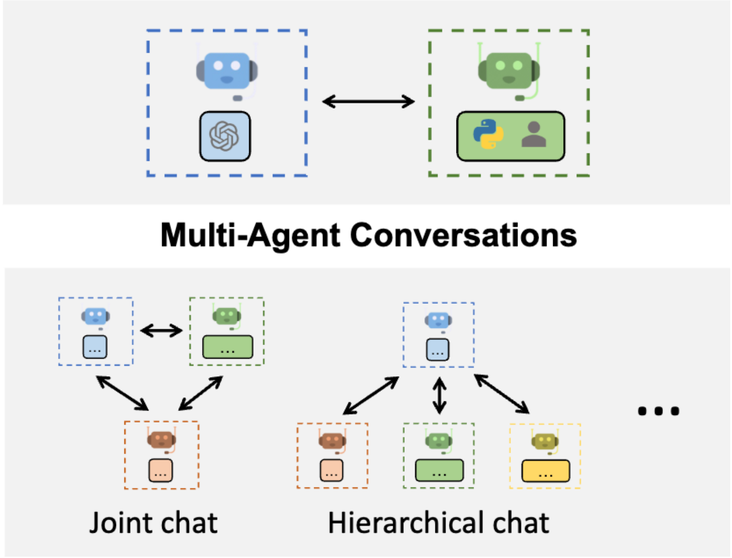
- LLMs are great but there are also challenges in using proprietary LLMs
- Ongoing research in:
  - SLMs for architects for design decisions
  - SLMs for edge deployment (Qualcomm EdgeAI labs @IIITH) using QIDK, Qualcomm
  - Using SLMs for function calling - Code generation (with Precog, IIITH)
  - .....



The age of Multi-agent Frameworks: AutoGen



Agent Customization



Flexible Conversation Patterns

AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation

Qingyun Wu[†], Gagan Bansal[*], Jieyu Zhang[±], Yiran Wu[†], Beibin Li[*]

Erkang Zhu[*], Li Jiang[*], Xiaoyun Zhang[*], Shaokun Zhang[†], Jiale Liu[∓]

Ahmed Awadallah[*], Ryen W. White[*], Doug Burger[*], Chi Wang[*1]

[*]Microsoft Research, [†]Pennsylvania State University

[±]University of Washington, [∓]Xidian University

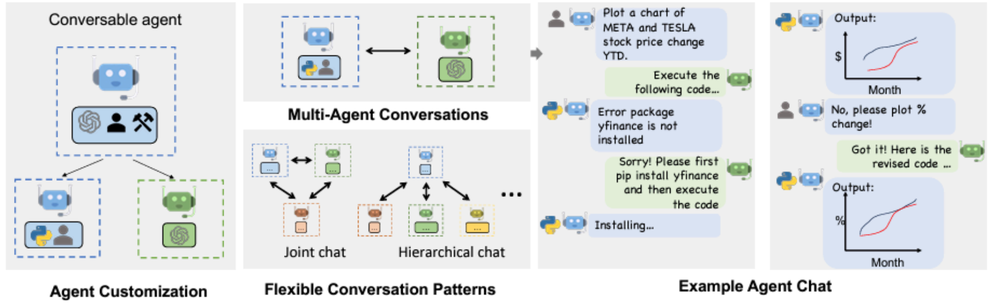
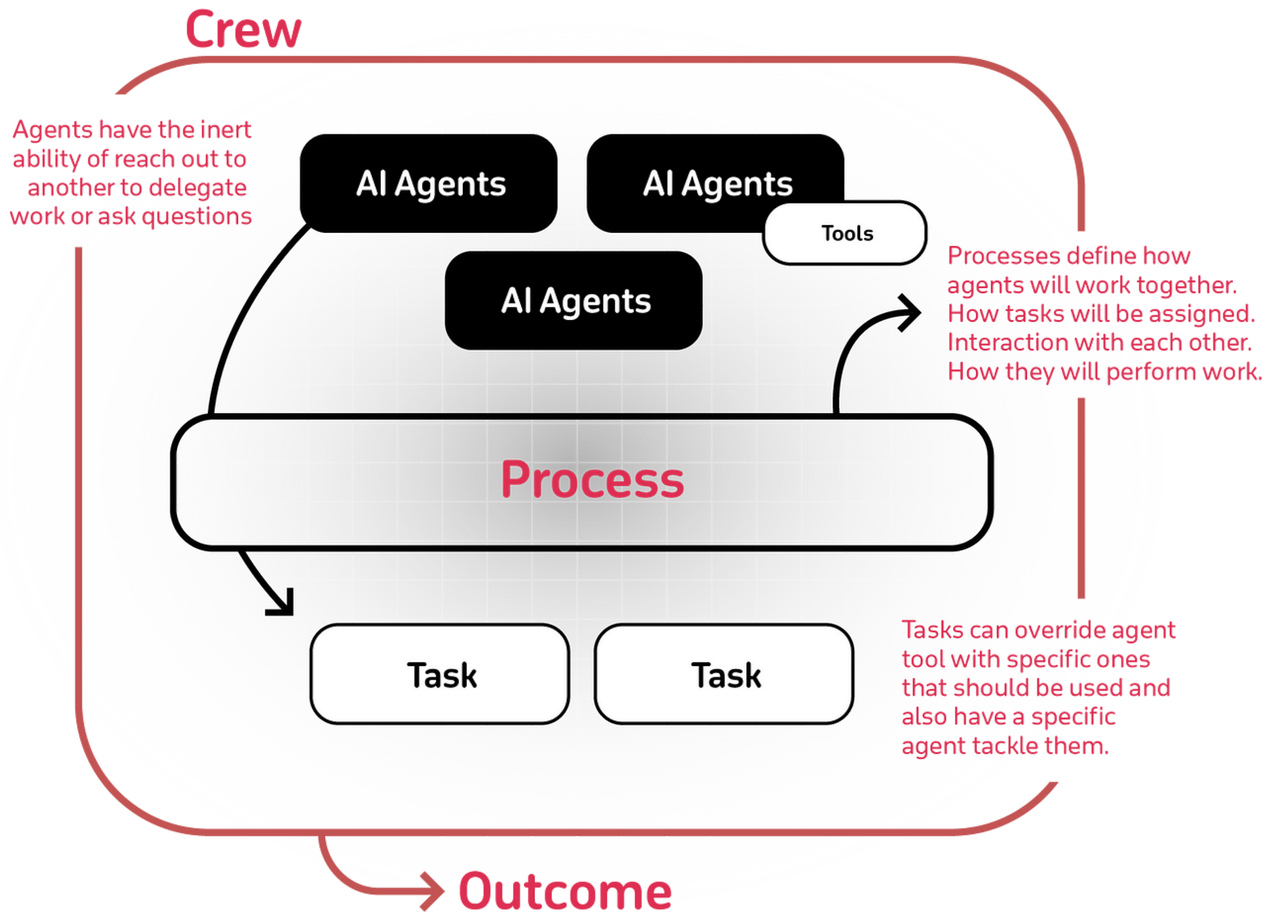


Figure 1: AutoGen enables diverse LLM-based applications using multi-agent conversations. (Left) AutoGen agents are conversable, customizable, and can be based on LLMs, tools, humans, or even a combination of them. (Top-middle) Agents can converse to solve tasks. (Right) They can form a chat, potentially with humans in the loop. (Bottom-middle) The framework supports flexible conversation patterns.

https://microsoft.github.io/autogen

The Crew Framework



- Agents can be composed to perform a broader task
- Assemble a crew of agents
- Each agent can have role and responsibility
- Agents interact with each other to achieve a functionality
- Other frameworks: langraph, autodev, etc.

https://crewai.com

If you are interested do give a read!

- LLMs can be used in different phases of the SDLC
- There has already been some works done in this space
- More works to be done in requirements, design, testing and maintenance
- There is also lot of scope for various empirical studies
- GenAI in SE must be human-centred [The Copenhagen Manifesto]

Large Language Models for Software Engineering: A Systematic Literature Review

XINYI HOU*, Huazhong University of Science and Technology, China
YANJIE ZHAO*, Huazhong University of Science and Technology, China
YUE LIU, Monash University, Australia
ZHOU YANG, Singapore Management University, Singapore
KAILONG WANG, Huazhong University of Science and Technology, China
LI LI, Beihang University, China
XIAPU LUO, The Hong Kong Polytechnic University, China
DAVID LO, Singapore Management University, Singapore
JOHN GRUNDY, Monash University, Australia
HAOYU WANG†, Huazhong University of Science and Technology, China

Large Language Models (LLMs) have significantly impacted numerous domains, including Software Engineering (SE). Many recent publications have explored LLMs applied to various SE tasks. Nevertheless, a comprehensive understanding of the application, effects, and possible limitations of LLMs on SE is still in its early stages. To bridge this gap, we conducted a systematic literature review (SLR) on LLM4SE, with a particular focus on understanding how LLMs can be exploited to optimize processes and outcomes. We select and analyze 395 research papers from January 2017 to January 2024 to answer four key research questions (RQs). In RQ1, we categorize different LLMs that have been employed in SE tasks, characterizing their distinctive features and uses. In RQ2, we analyze the methods used in data collection, preprocessing, and application, highlighting the role of well-curated datasets for successful LLM for SE implementation. RQ3 investigates the strategies employed to optimize and evaluate the performance of LLMs in SE. Finally, RQ4 examines the specific SE tasks where LLMs have shown success to date, illustrating their practical contributions to the field. From the answers to these RQs, we discuss the current state-of-the-art and trends, identifying gaps in existing research, and flagging promising areas for future study. Our artifacts are publicly available at https://github.com/xinyi-hou/LLM4SE_SLR.

The Journal of Systems and Software 216 (2024) 112115

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

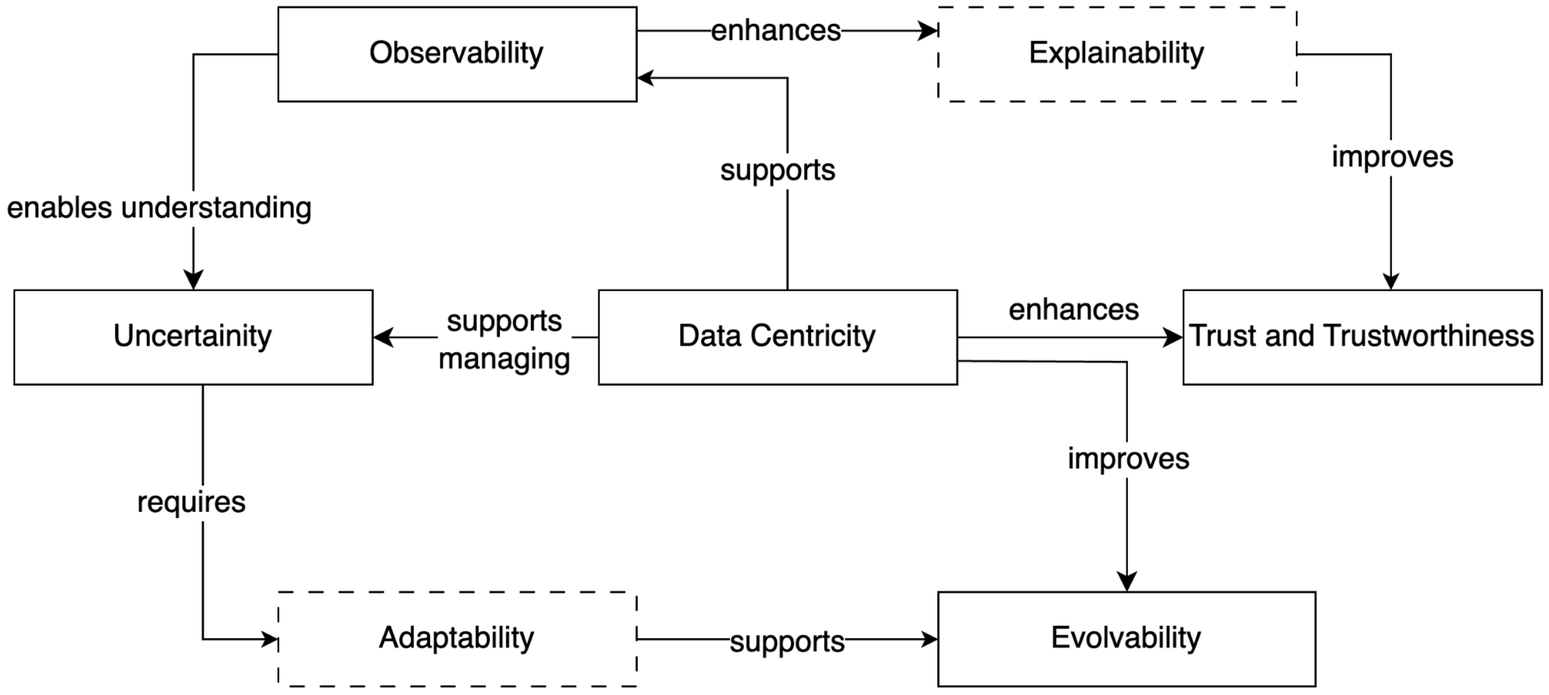
Generative AI in Software Engineering Must Be Human-Centered: The Copenhagen Manifesto

Key Takeaways

LLMs can be a best friend to the architect/engineer if used wisely!

- LLM presents a great opportunity for effective AKM with potential for run-time adaptation
- Domain specific LLMs which are smaller shall be the way forward - SLMs!
- There are many areas in SE that needs a lot of exploration
- Need for more tools - like cursor, GitHub co-pilot,..
- Need for better ways to architect/engineer systems around LLMs
- LLMs are not here to replace but to support!

SA for ML-enabled Systems: Quality Centric View and Challenges



Report from Dagstuhl Seminar 23302

Software Architecture and Machine Learning

Grace A. Lewis[*1], **Henry Muccini**[*2], **Ipek Ozkaya**[3], **Karthik Vaidhyanathan**[†4], **Roland Weiss**[*5], and **Liming Zhu**[*6]

1 Carnegie Mellon Software Engineering Institute – Pittsburgh, US. glewis@sei.cmu.edu
2 University of L'Aquila, IT. henry.muccini@univaq.it
3 Carnegie Mellon Software Engineering Institute – Pittsburgh, US. ozkaya@sei.cmu.edu
4 IIIT Hyderabad, IN. karthik.vaidhyanathan@iiit.ac.in
5 ABB – Mannheim, DE. roland.weiss@gmail.com
6 Data61, CSIRO – Sydney, AU. liming.zhu@data61.csiro.au

Abstract

This report documents the program and outcomes of Dagstuhl Seminar 23302, "Software Architecture and Machine Learning". We summarize the goals and format of the seminar, results from the breakout groups, key definitions relevant to machine learning-enabled systems that were discussed, and the research roadmap that emerged from the discussions during the seminar. The report also includes the abstracts of the talks presented at the seminar and summaries of open discussions.

Seminar July 23–28, 2023 – https://www.dagstuhl.de/23302
2012 ACM Subject Classification Software and its engineering → Software architectures; Computing methodologies → Machine learning; Software and its engineering → Extra-functional properties; Computing methodologies → Artificial intelligence; Software and its engineering
Keywords and phrases Architecting ML-enabled Systems, ML for Software Architecture, Software Architecture for ML, Machine Learning, Software Architecture, Software Engineering
Digital Object Identifier 10.4230/DagRep.13.7.166

1 Executive Summary

Grace A. Lewis (Carnegie Mellon Software Engineering Institute – Pittsburgh, US)
Henry Muccini (University of L'Aquila, IT)
Ipek Ozkaya (Carnegie Mellon Software Engineering Institute – Pittsburgh, US)
Karthik Vaidhyanathan (IIIT – Hyderabad, IN)
Roland Weiss (ABB – Mannheim, DE)
Liming Zhu (Data61, CSIRO – Sydney, AU)

The pervasive and distributed nature of many of today's software systems requires making complex design decisions to guarantee important system qualities such as performance, reliability, safety and security. The practices within the field of software architecture guide the design and development of software systems from its high-level blueprint down to their implementation and operations. While the fundamentals of software architecture practices

* Editor / Organizer
† Editorial Assistant / Collector

Software Architecture and Machine Learning, *Dagstuhl Reports*, Vol. 13, Issue 7, pp. 166–188
Editors: Grace A. Lewis, Henry Muccini, Ipek Ozkaya, Karthik Vaidhyanathan, Roland Weiss, and Liming Zhu
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The Future is here

- **Context is needed** - Capturing organizational aspects needs work, code can help!
- **Text to requirements to Design** - Fasten the cycle
- **LLMs will hallucinate** - No stopping that but we can reduce it - better engineering!
- **Multiple agents collaborating** together to help architects/developers
- **Large action models (LAMs)** for self-adaptation, task generation, resolution
- Lot of potential for support in **architecture migration and assessment (Technical Debt reduction!)**
- **SE Process will also need upgrades!!**

Thanks to my team - SA4S@SERC


Rudra Dhar


Akhila Matathammal


Hiya Bhatt


Chandrasekar S

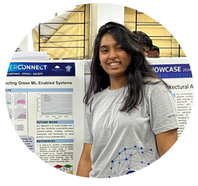

Shubham Kulkarni


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

SERC
Software Engineering Research Centre

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

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

@ ICSA 2025, SAGAI 2025

IEEE Software Magazine

SCAN ME

Thank you

SE Radio Podcasts

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