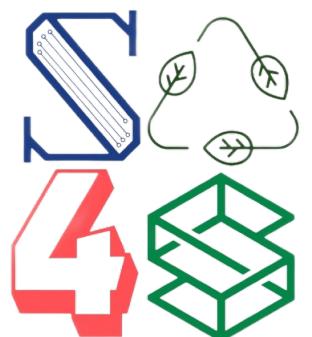


At the convergence of LLMs and Software Engineering: Perspectives from our Research

Dr. Karthik Vaidyanathan

<https://karthikvaidyanathan.com>

Feb 10, 2025



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.

February 2, 2025 Release

Introducing deep research

An agent that uses reasoning to synthesize large amounts of online information and complete multi-step research tasks for you. Available to Pro users today, Plus and Team next.

Try on ChatGPT ↗

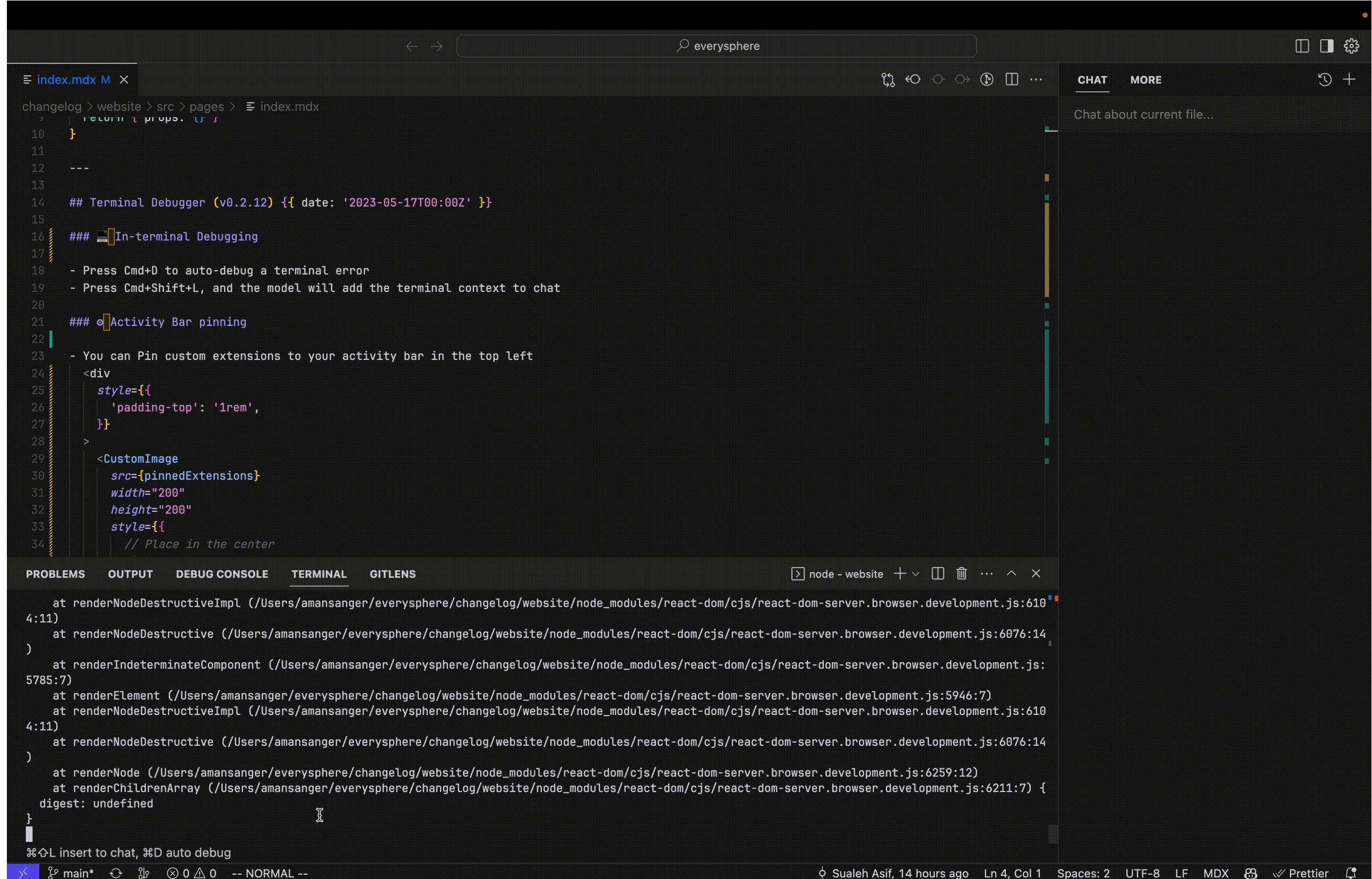


Source: openai.com, meta.com, cursor.com



Even the car that dropped me ;)

In the Context of Software Development



A screenshot of a software development environment, likely VS Code, showing a dark-themed interface. The main area displays a file named 'index.mdx' with code related to a 'changelog' and 'react-dom'. A terminal window at the bottom shows a stack trace from 'react-dom-server.browser.development.js'. The activity bar at the bottom has pinned extensions like 'node - website' and 'GITLENS'. A chat interface is visible on the right side.

FORBES > INNOVATION > CLOUD

GitHub Copilot Agent And The Rise Of AI Coding Assistants

Janakiram MSV Senior Contributor 

I cover emerging technologies with a focus on infrastructure and AI

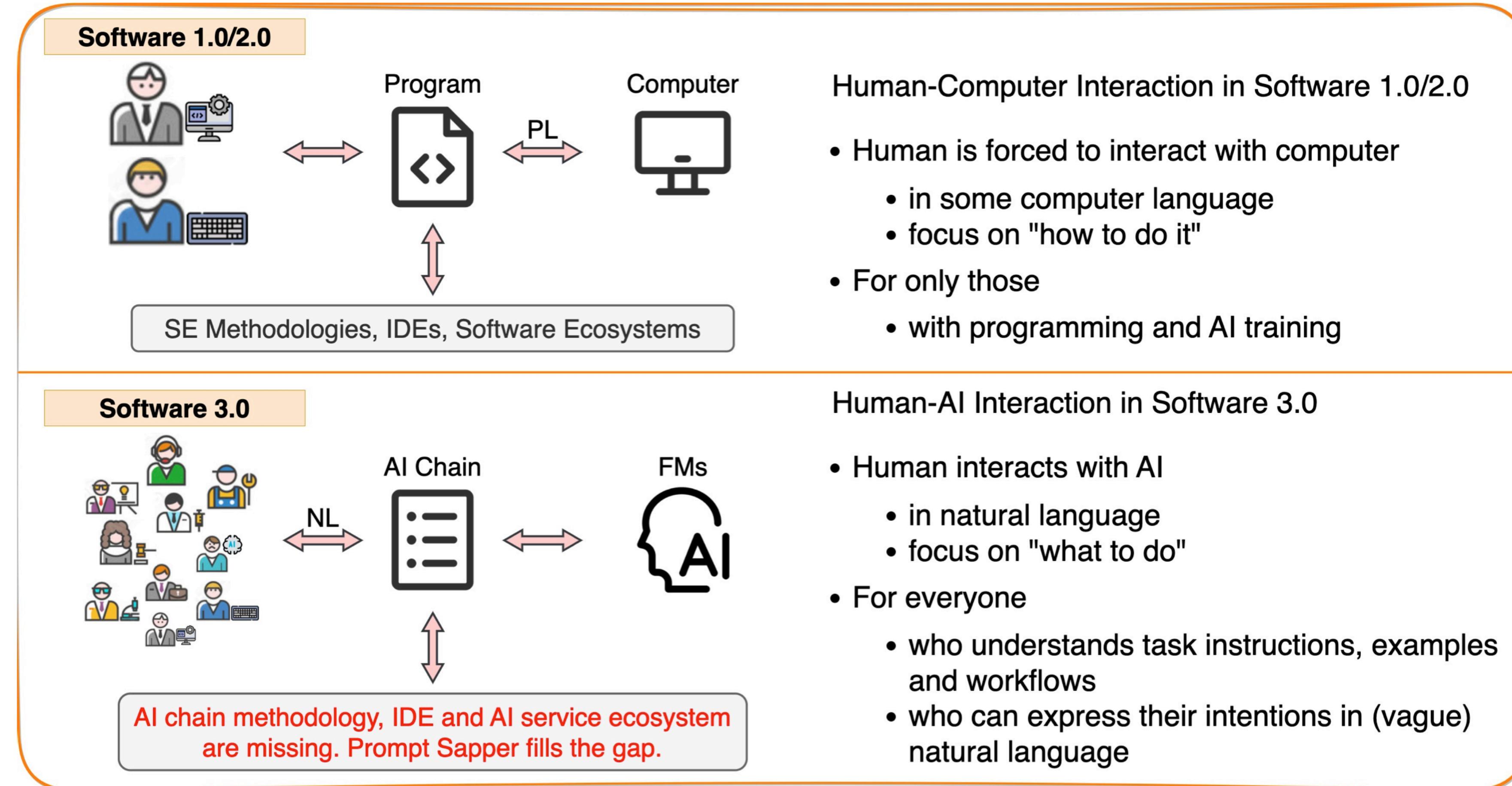
[Follow](#)



Source: cursor.com, forbes.com



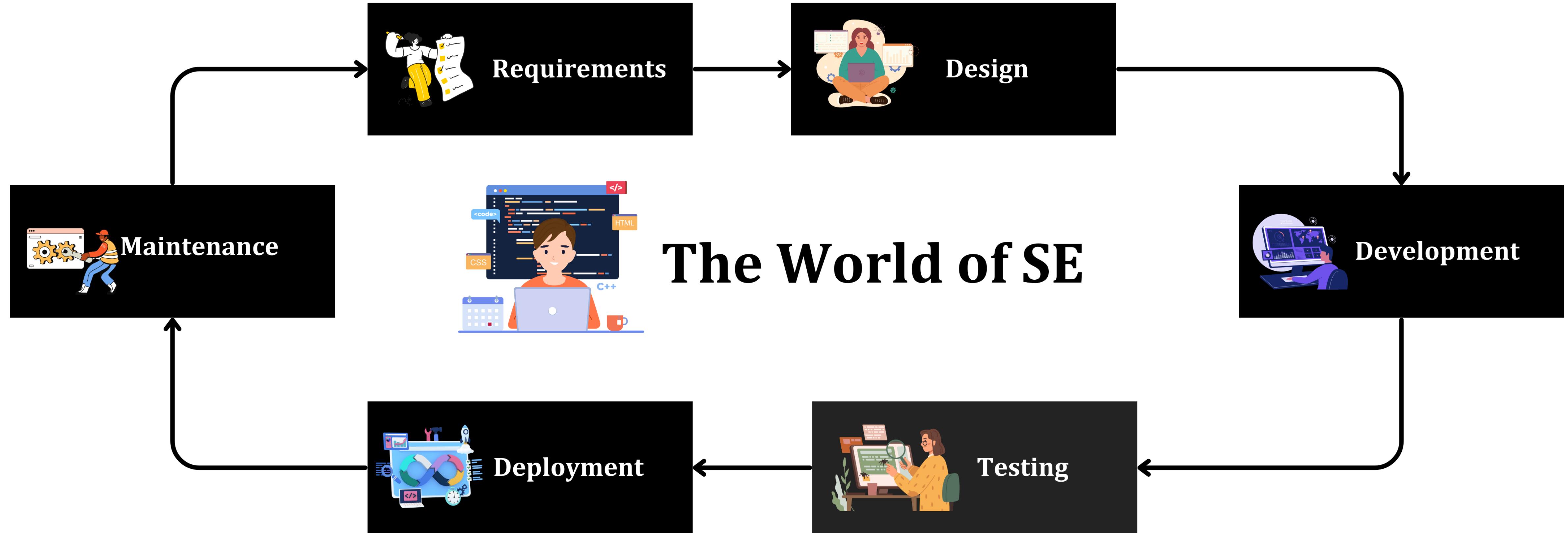
The shift in software



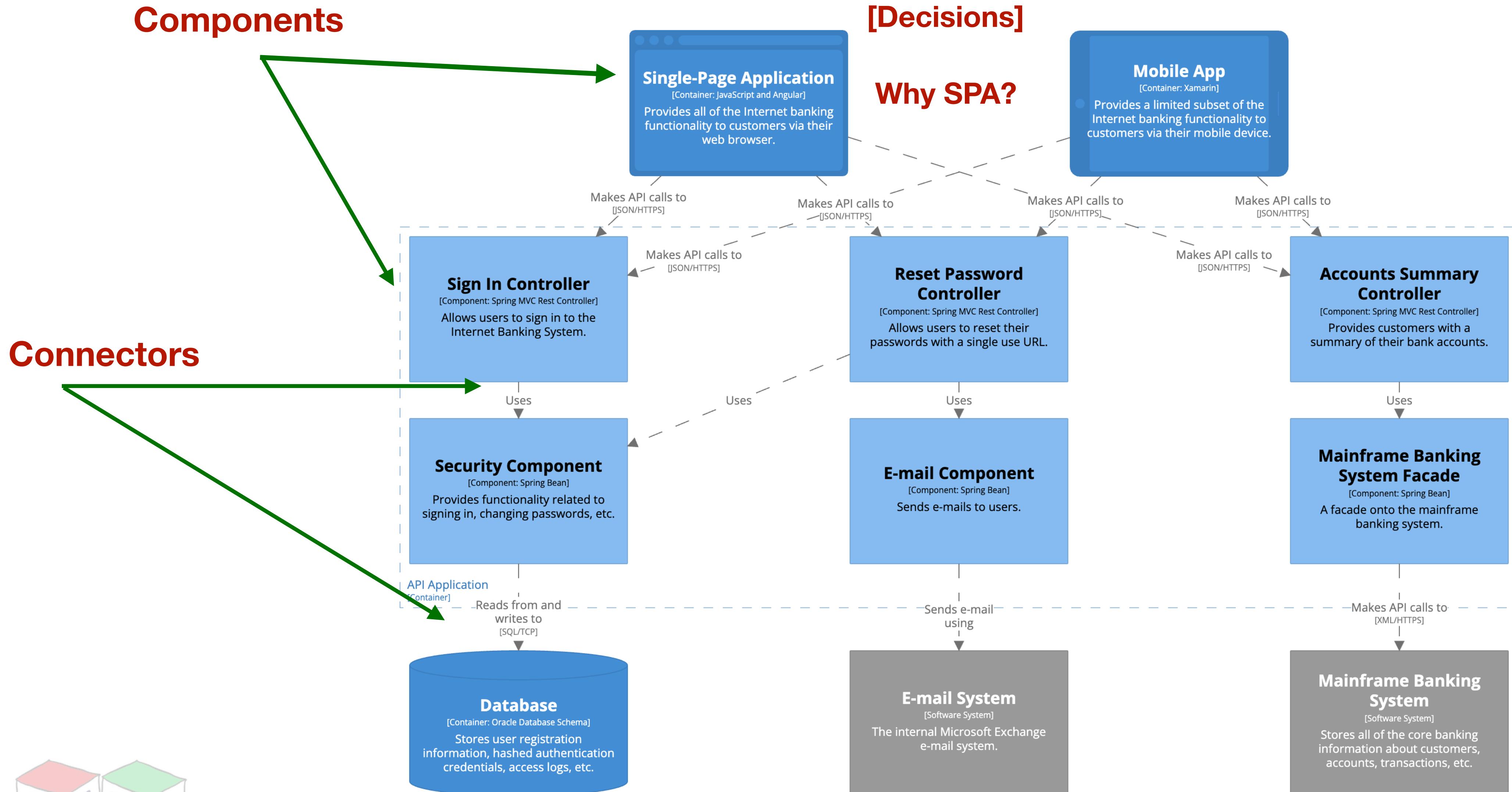
We still have a long way to go but we need to embrace!



The World of Software Engineering



Starting with Design: The Architecture Space



What essentially is Software Architecture?

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

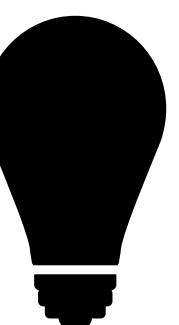
Software & Application Technologies Lab
Nokia Research Center
PO BOX 407, FI-00045, Finland
jan.bosch@nokia.com

Abstract

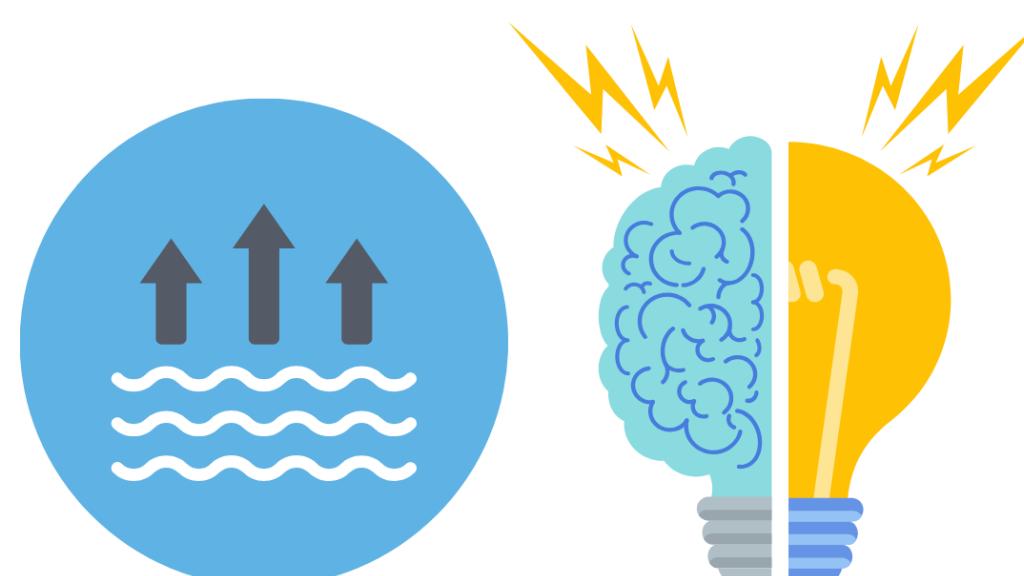
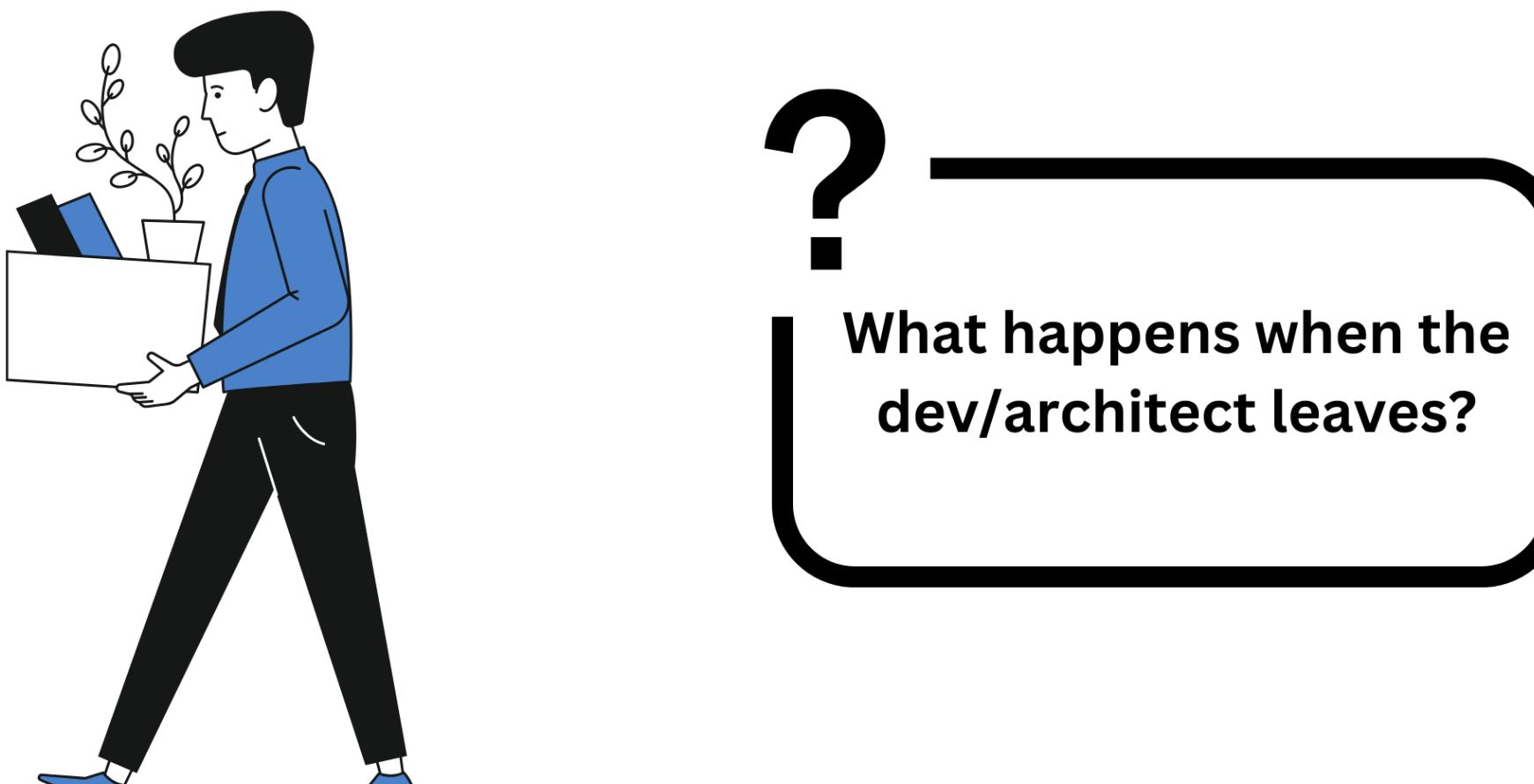
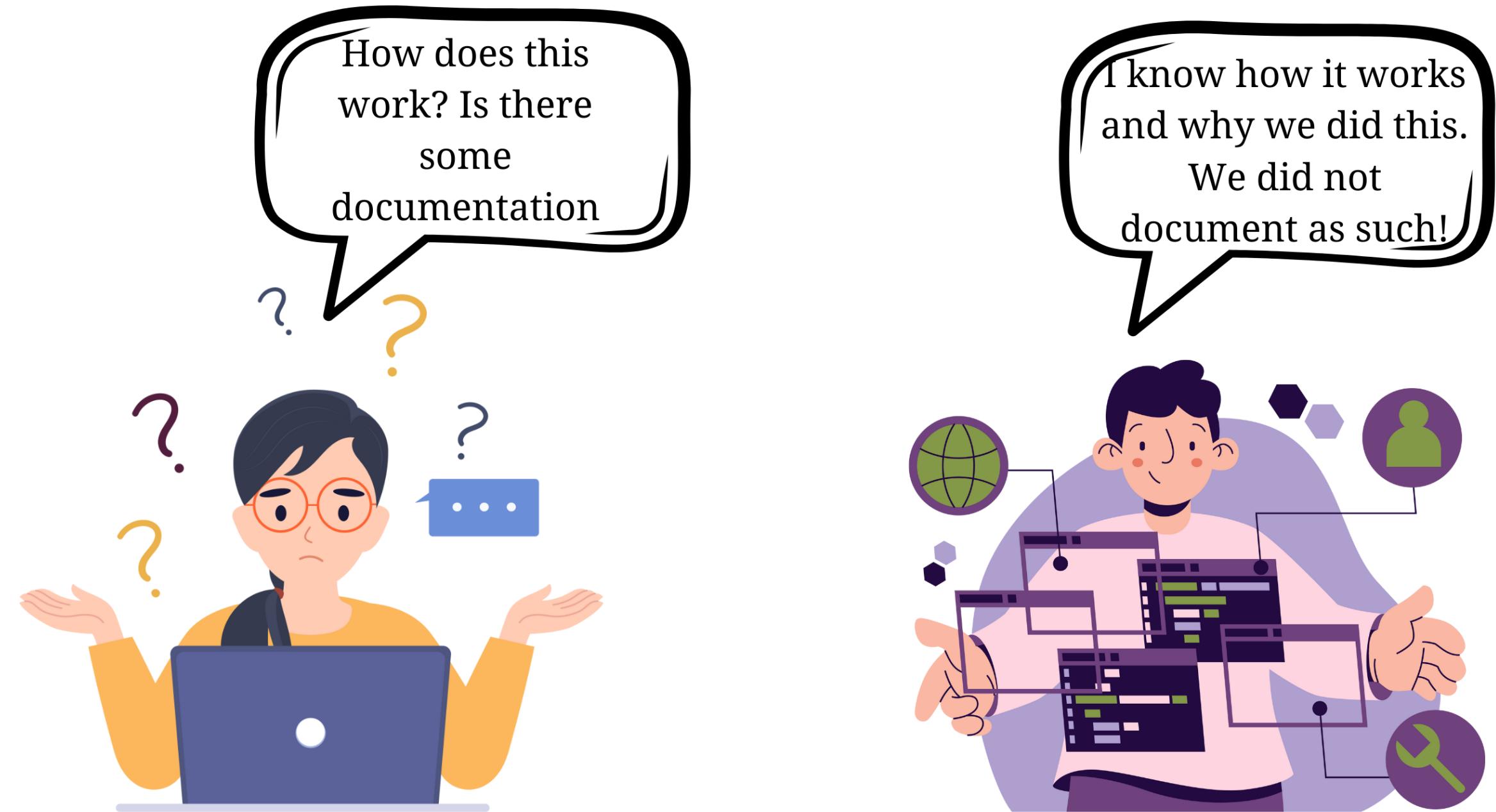
Software architectures have high costs for change, are complex, and erode during evolution. We believe these problems are partially due to knowledge vaporization. Currently, almost all the knowledge and information about the design decisions the architecture is based on are implicitly embedded in the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



The Key Issue



Takes away the knowledge!

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*



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 ▾

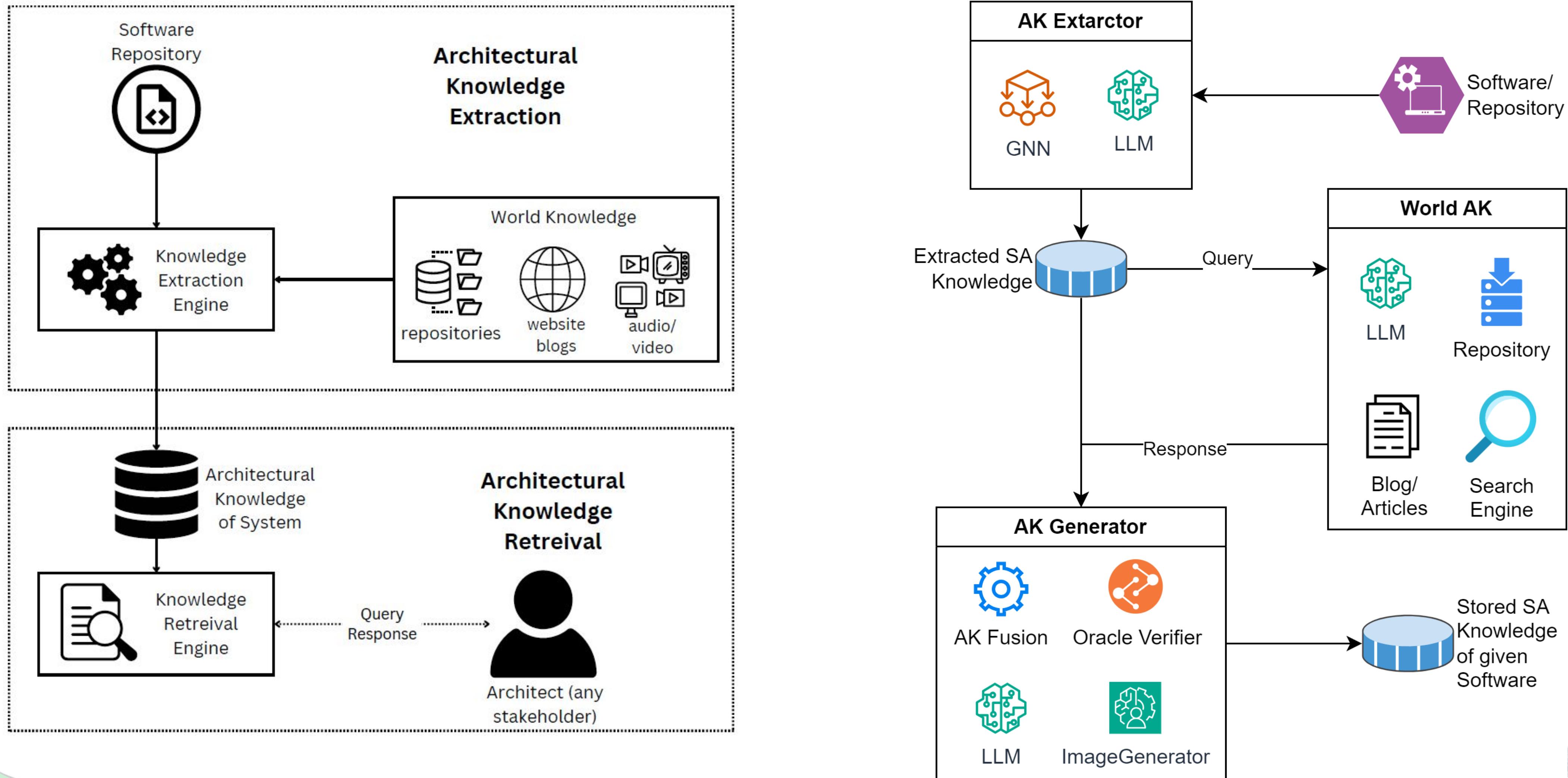
+ Add to Mendeley  Share  Cite

<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



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?**

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

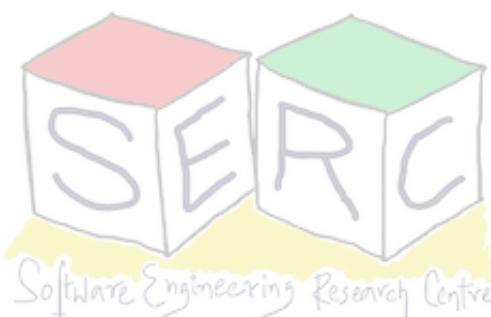
1. Python has support for various ML and data analysis
2. Team members are already familiar with Python

Status

Decided

Consequences

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



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

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

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.

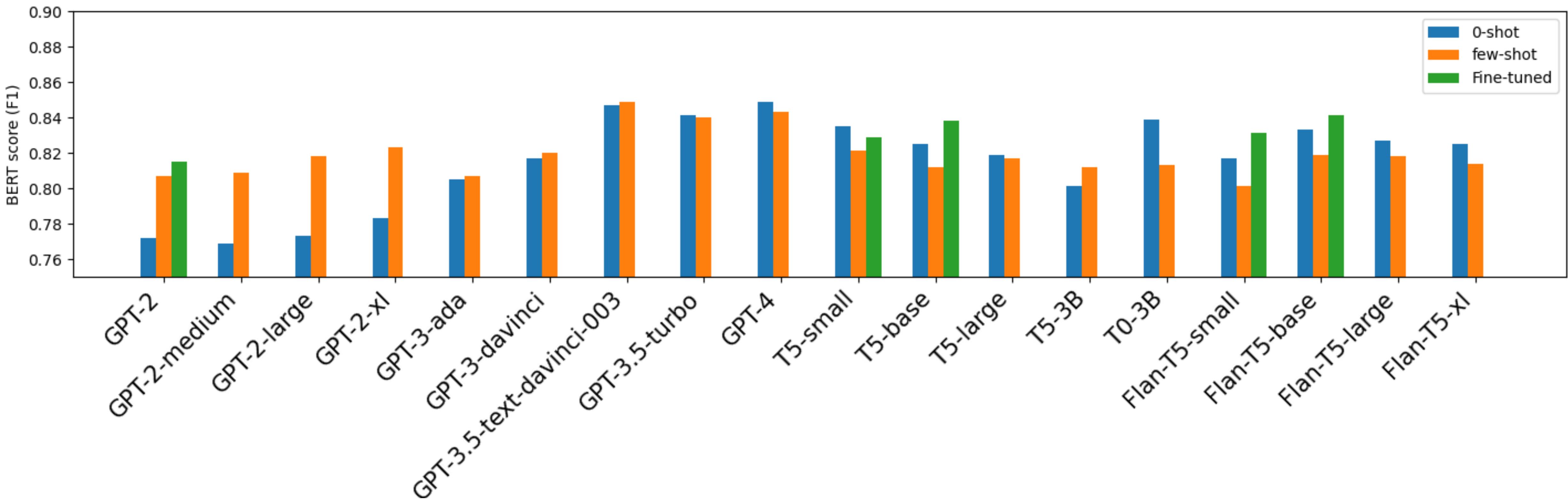
We need to choose Python as a programming language, because all of our projects will require Python to use it.

Flan-T5 fine-tuned



Study repo

The Overall Results



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



Introducing the ArchCopilot

The screenshot shows a dark-themed application window titled "Architecture Decision Records". On the left, a sidebar lists navigation options: Home, Explore, New Chat (with a pencil icon), Choosing Java for Financial Payment Gateway (highlighted in a grey box), CSS Framework Recommendations, Project Decision Recording, Managing Architecture Decision Records, Categorizing Architecture Decisions, and Data Analysis Project Decisions.

The main content area displays a user query under the heading "USER":

I want to decide between Python and Java for building my financial payment gateway module. My organization is an Oracle customer and has a lot of java developers. The primary focus of the system should be security, Privacy and Scalability

Below it, the "ASSISTANT" section provides a response:

Based on the information provided in the documents, Java seems to be a suitable choice for building your financial payment gateway module. Java is known for its strong security features, which is crucial for handling financial transactions. Additionally, Java is widely used in enterprise environments and is supported by Oracle, which is beneficial since your organization is an Oracle customer with many Java developers.

In terms of scalability, Java is known for its ability to handle large-scale applications effectively. It is used in many high-traffic and mission-critical systems, making it a reliable choice for a payment gateway module that needs to handle a high volume of transactions.

While Python is also a popular programming language with its own strengths, based on the specific requirements of security, privacy, and scalability for a financial payment gateway module, Java seems to be the more suitable choice in this case.

At the bottom, there is a search bar with the placeholder "Enter Question" and a "Search" button with a magnifying glass icon.

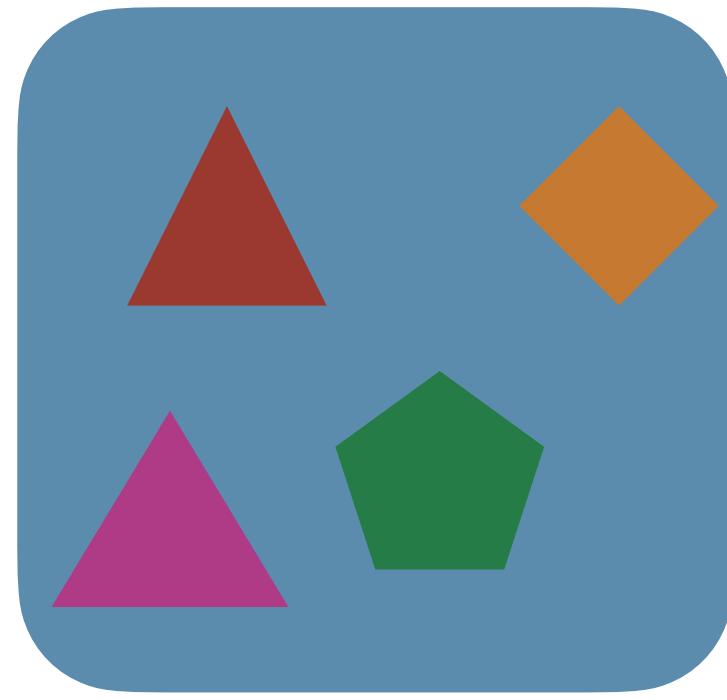
1. Add knowledge - decisions..
2. Search for similar decisions
3. Retrieve knowledge
4. Generate decision records for a given context

A companion for architect!

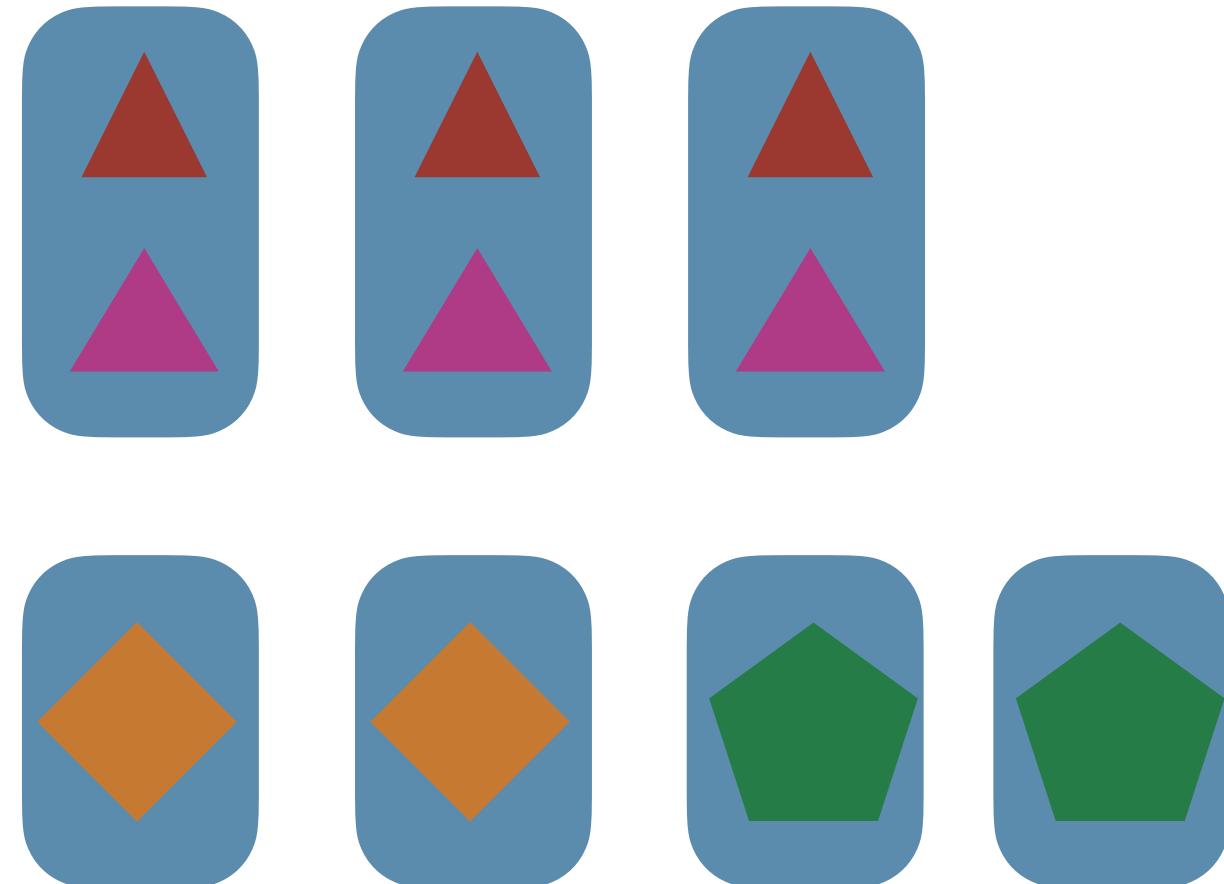
Going beyond just design - What about Components?

Can we automate generation of architectural components?

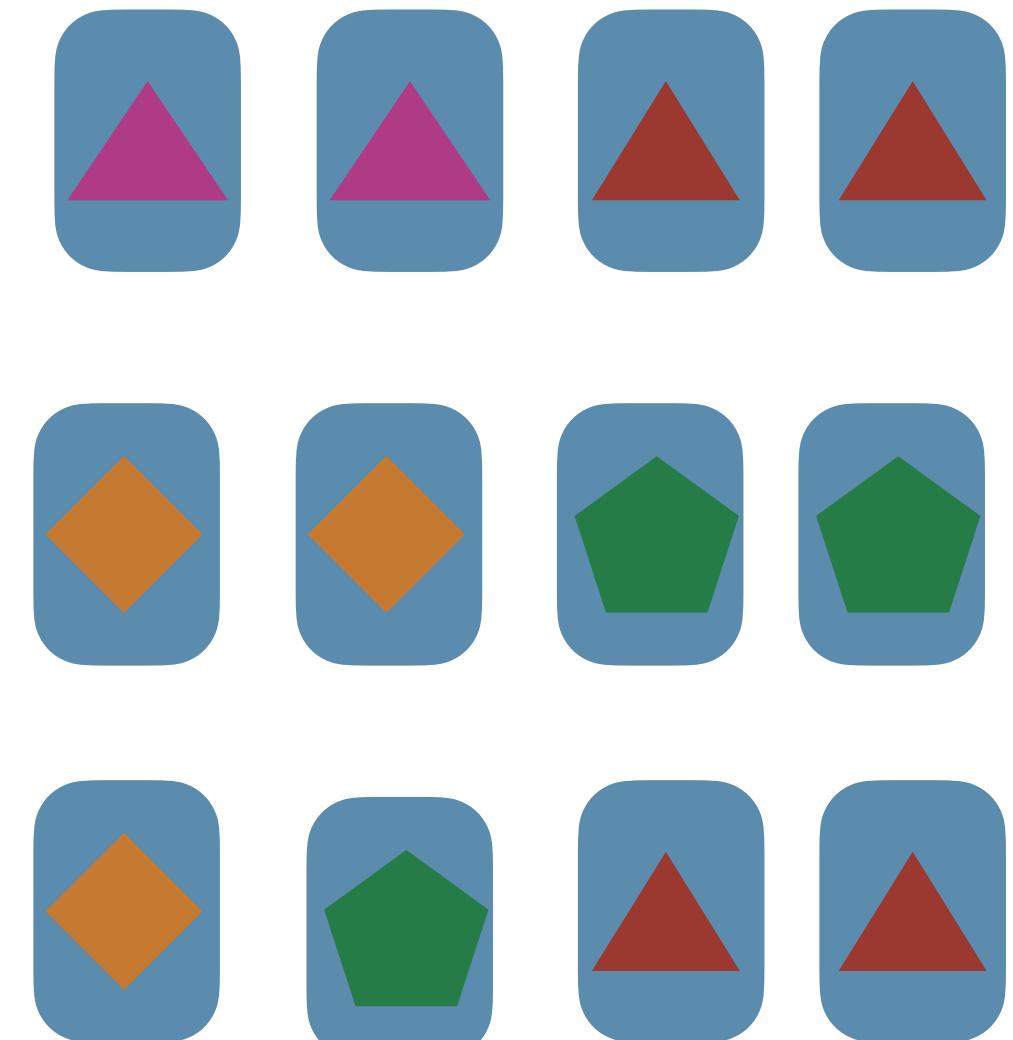
Monolith



Microservice



Serverless



Study Design

Inspired from the concept of masking in language models

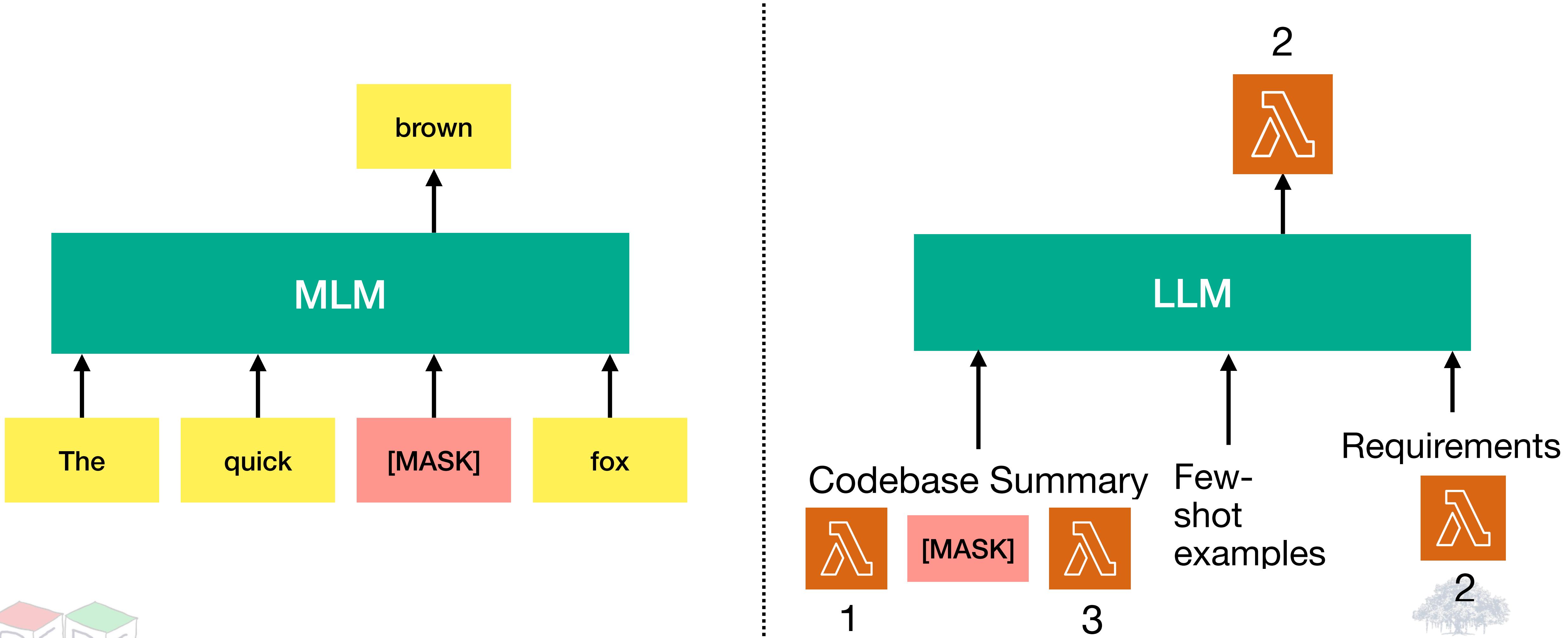
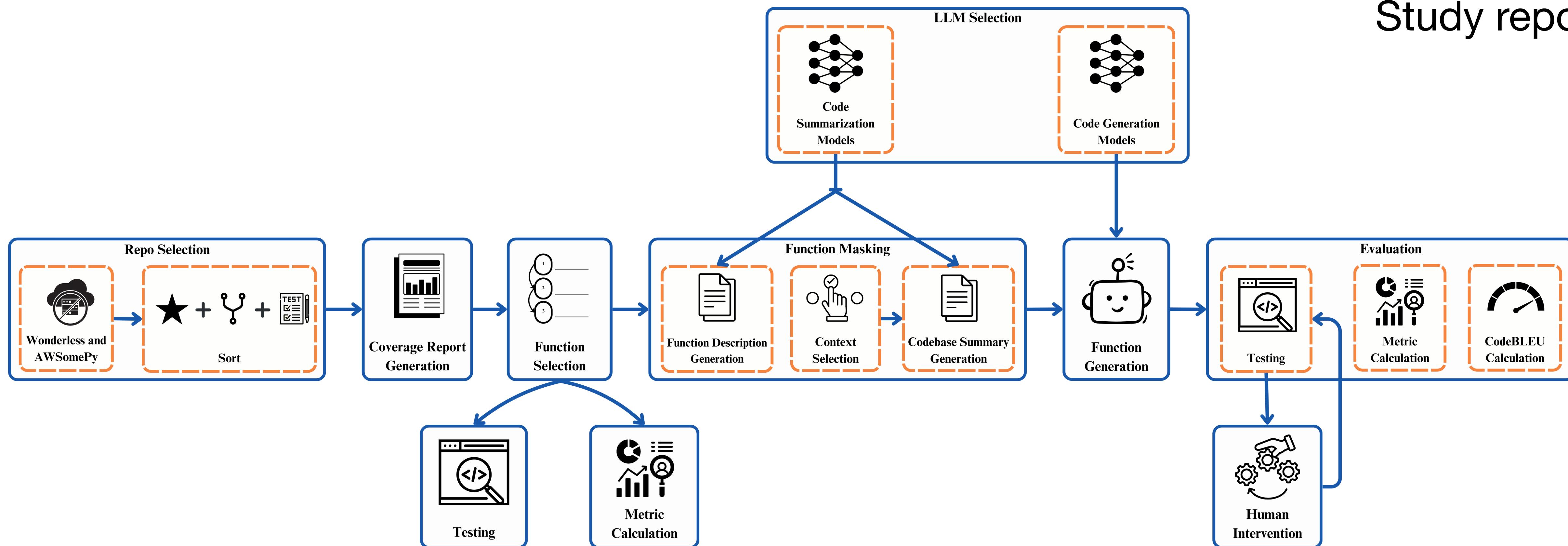


Image credits: Shrikara A



Study repo

Study Design

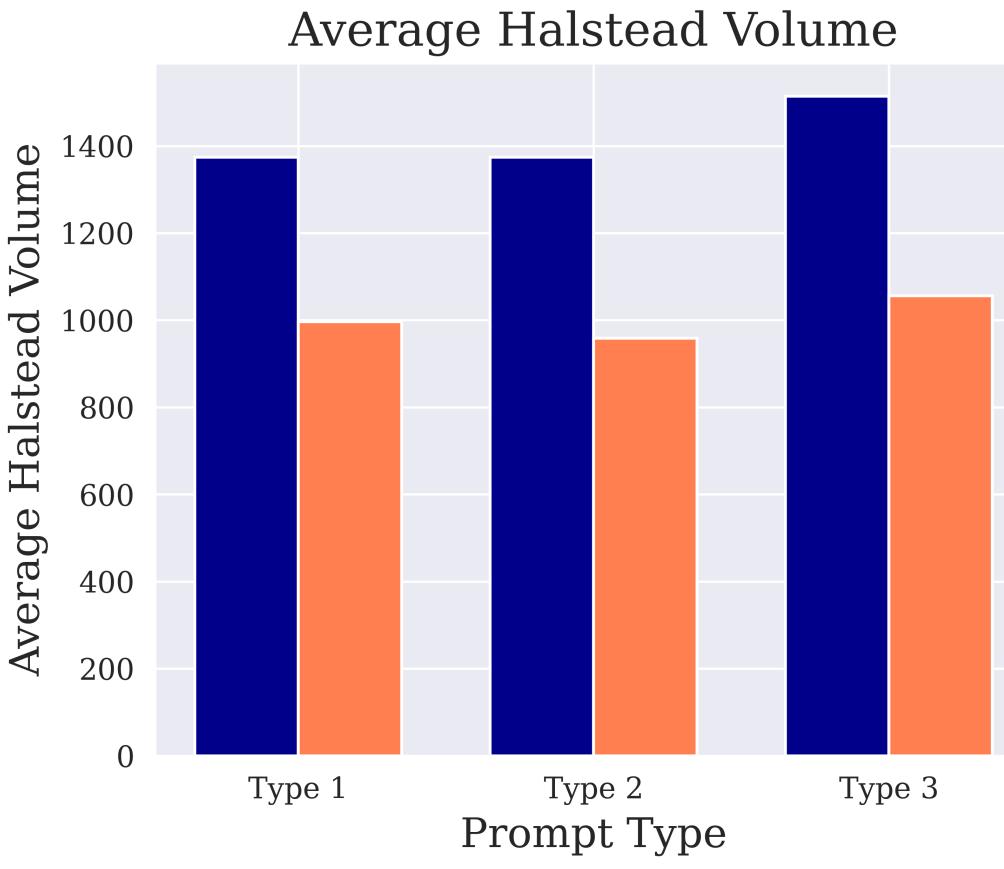
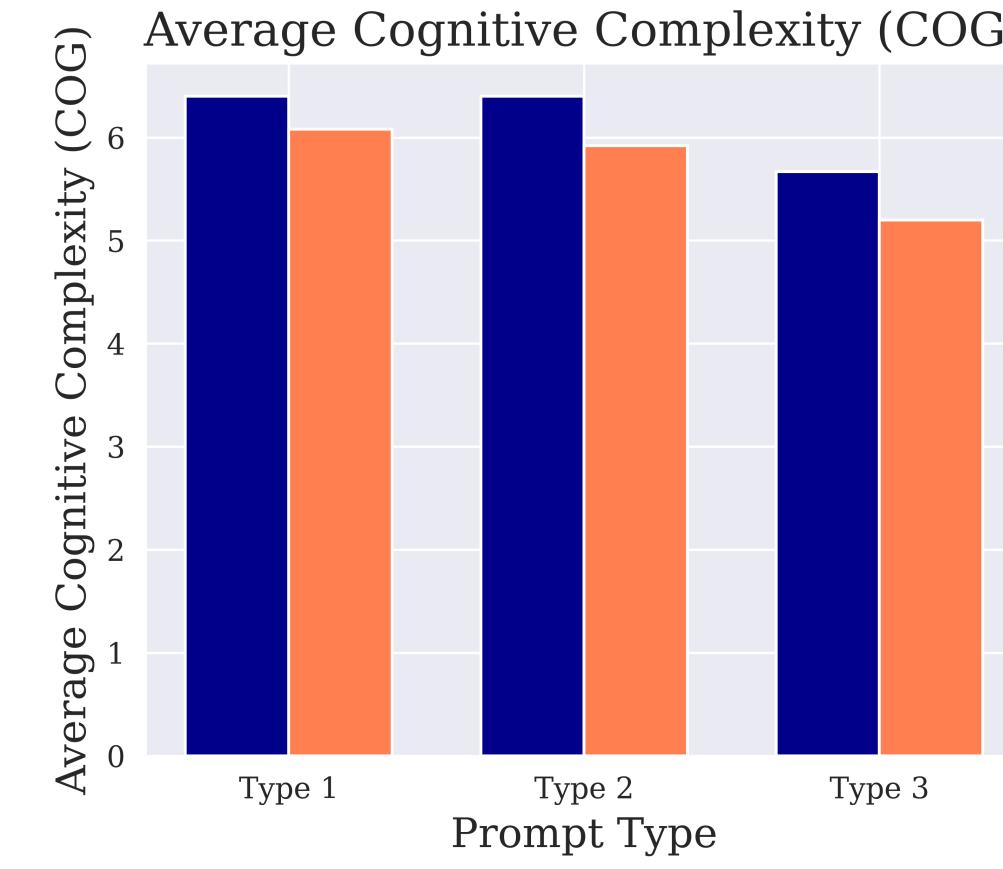
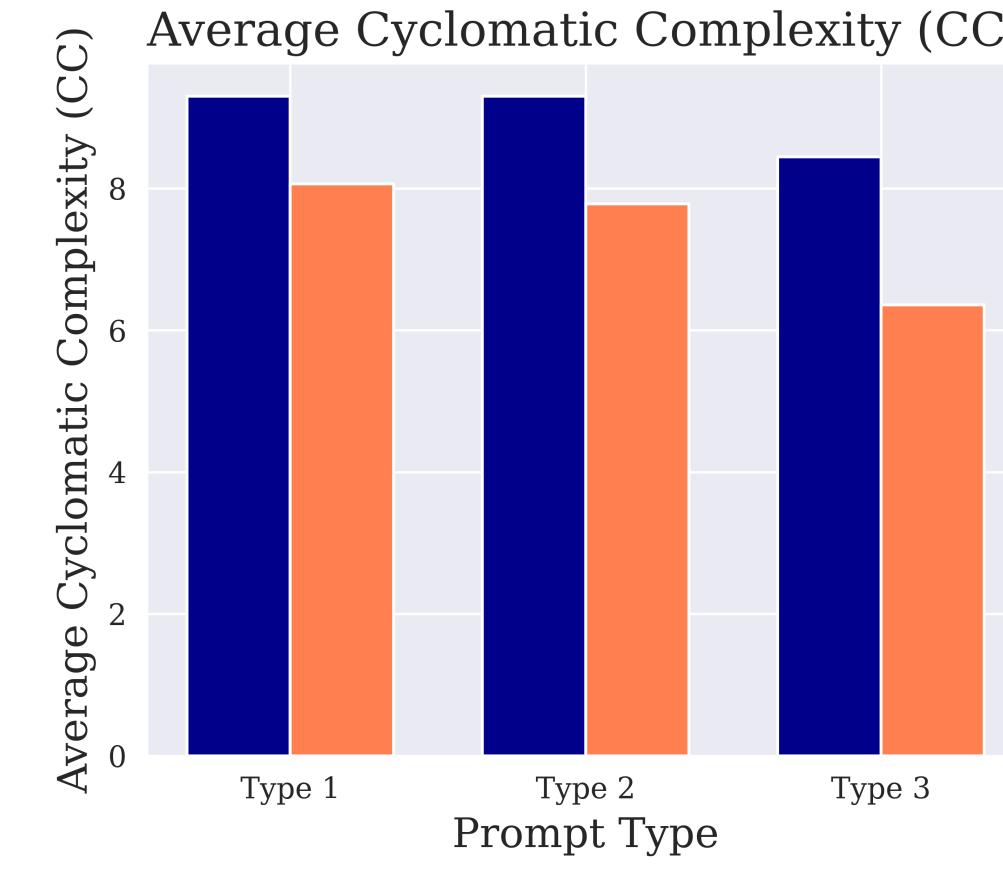
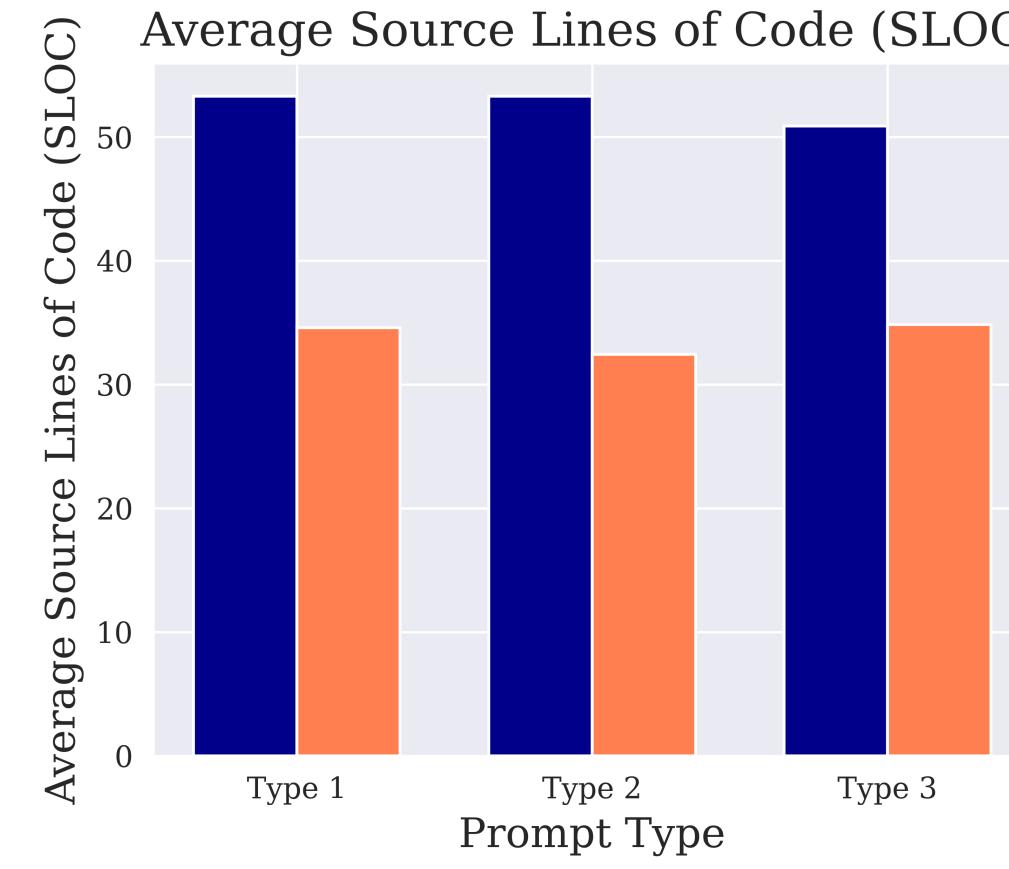
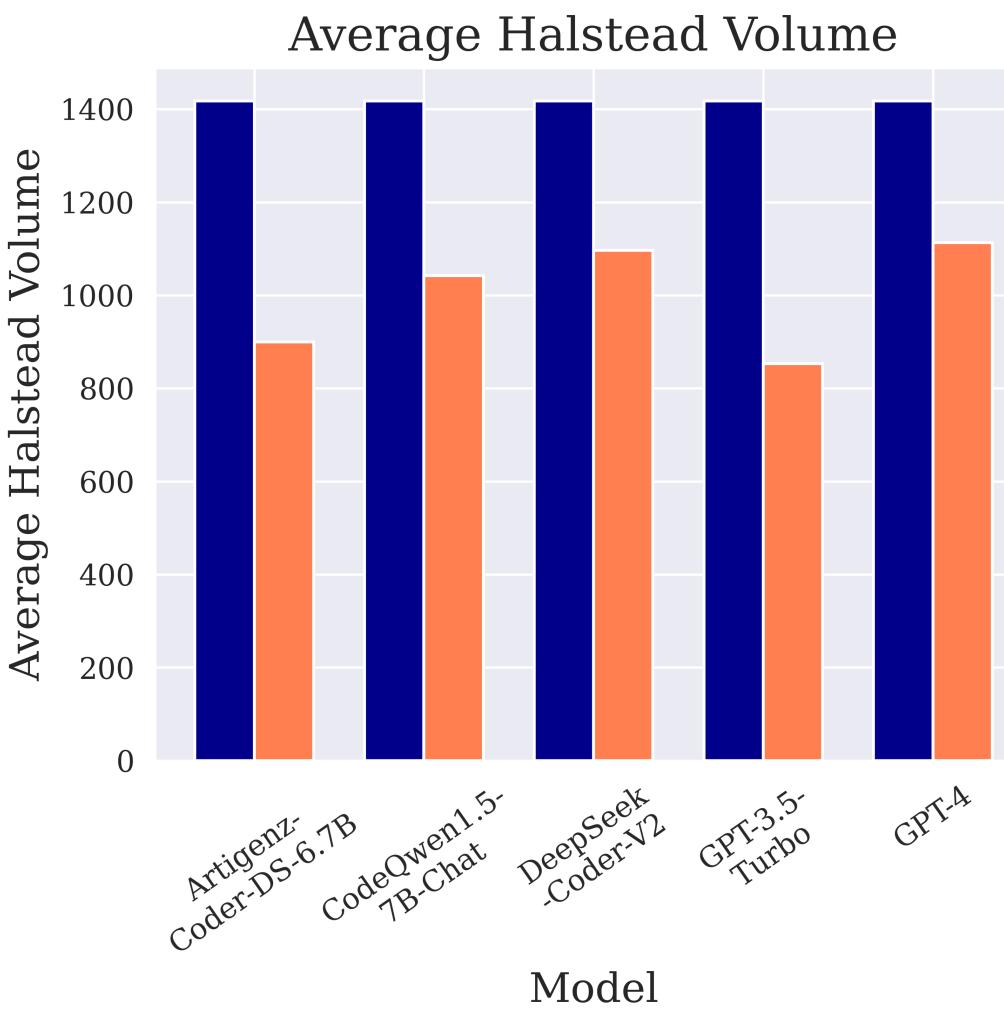
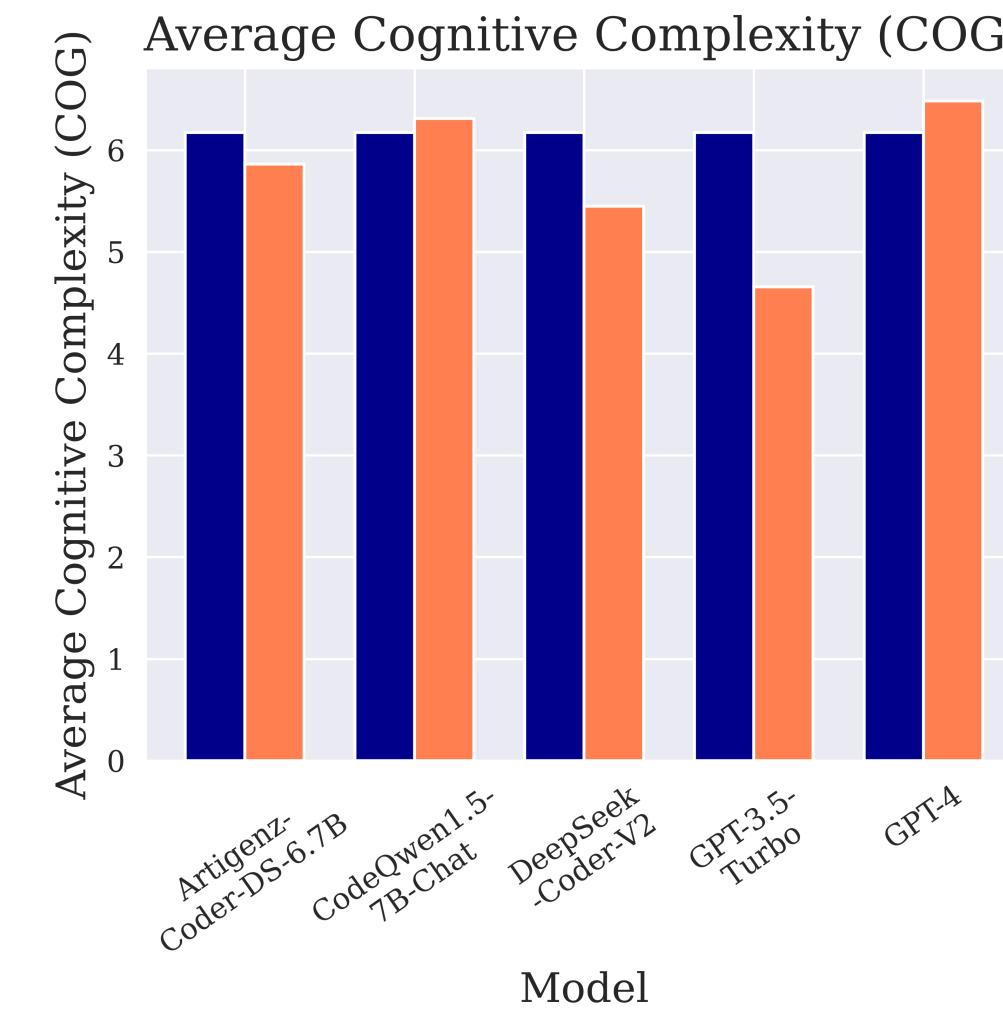
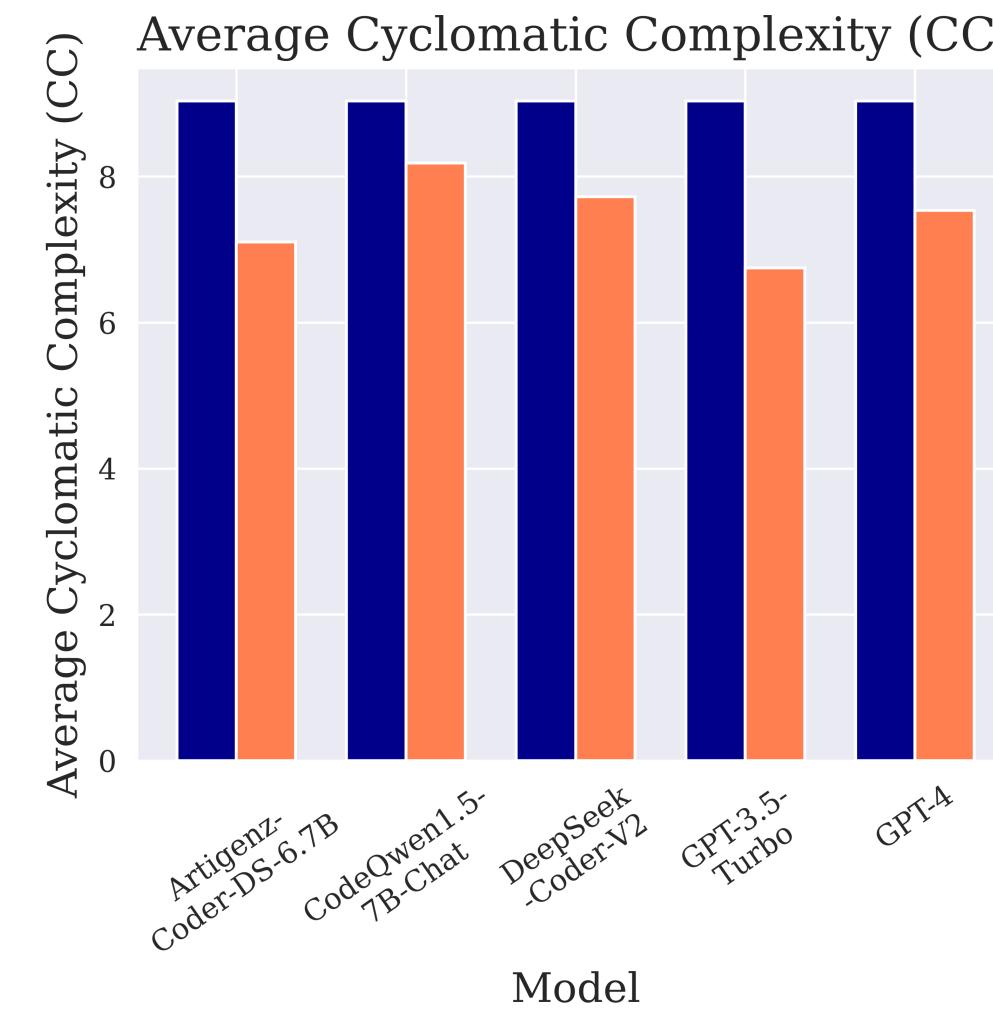
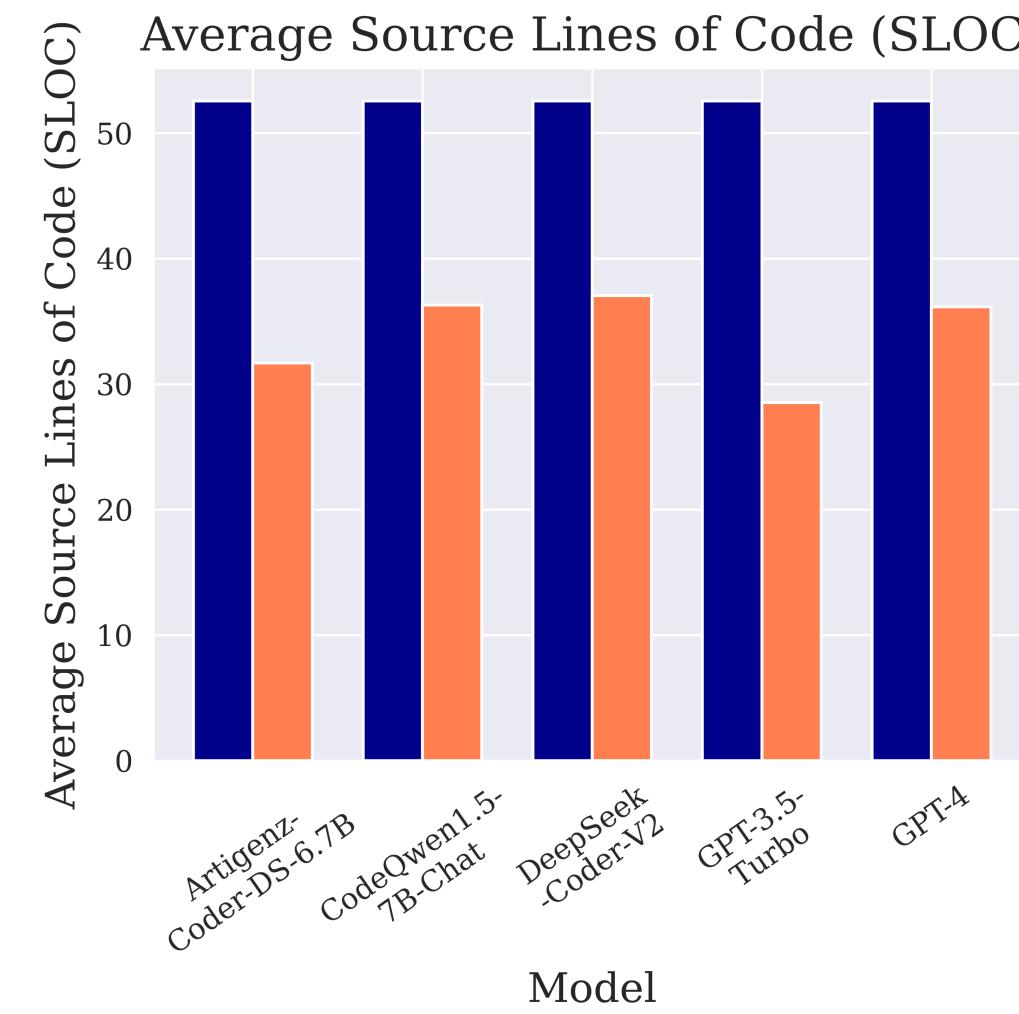


3 types of abstraction in prompt, 4 serverless repo, 5 models, 145 functions tested

Some Key Results

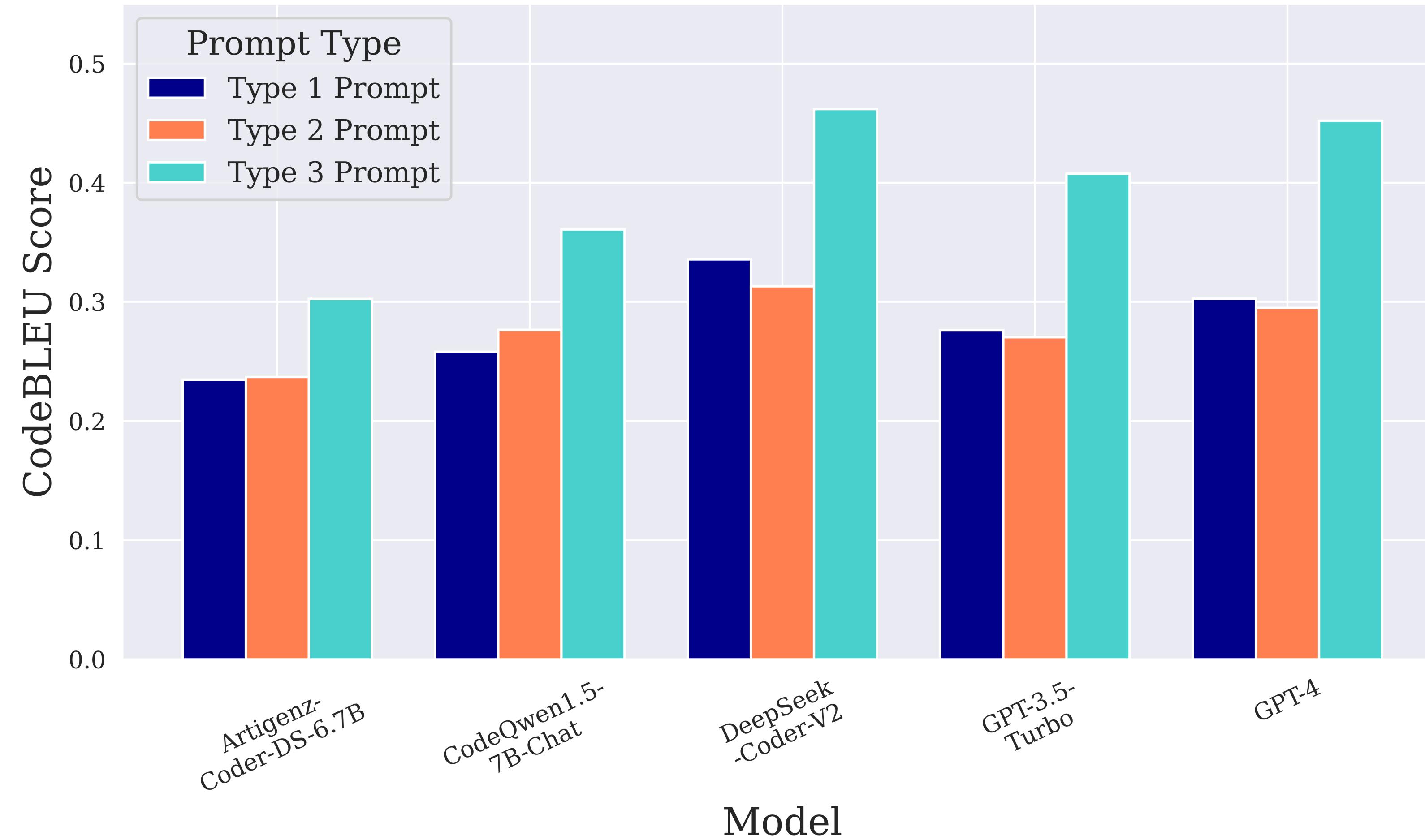
Model	Initial Test Pass Rate	Type 1 (No intervention)	Type 1 (intervention)	Type 2 (No intervention)	Type 2 (intervention)	Type 3 (No intervention)	Type 3 (intervention)
Artigenz-Coder-DS-6.7B	100	0	0	0	0	0	0
CodeQwen1.5-7 B-Chat	100	3	3	0	22	7	33
DeepSeek-Coder-V2	100	0	18	0	39	13	71
GPT-3.5-Turbo	100	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



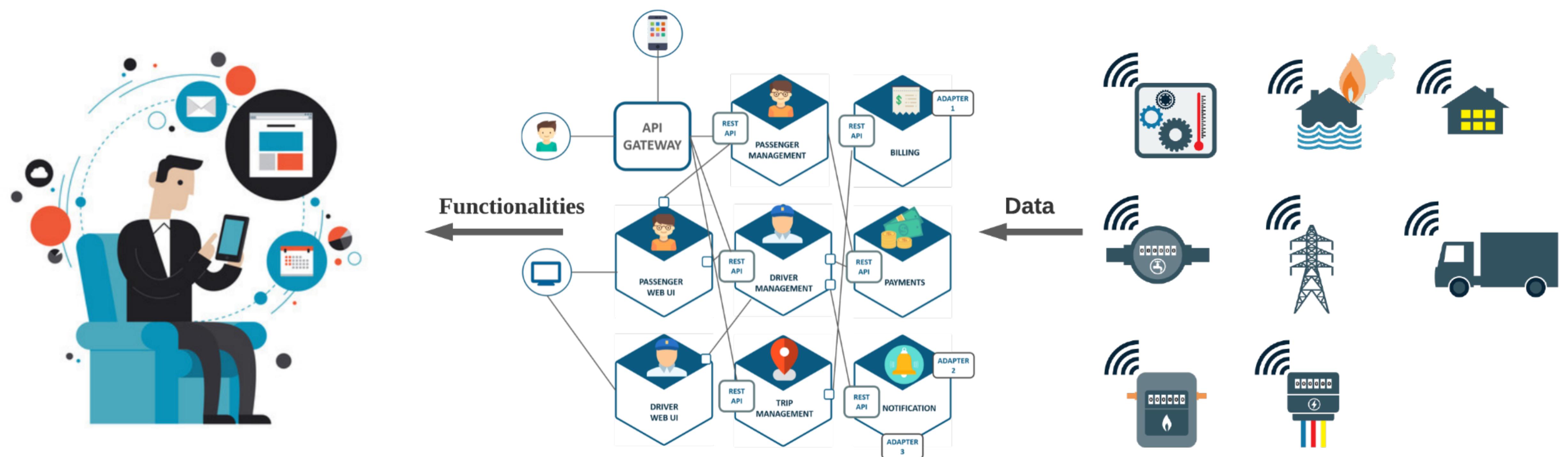
■ Original Serverless Function ■ Generated Serverless Function

Some Key Results - Code Quality



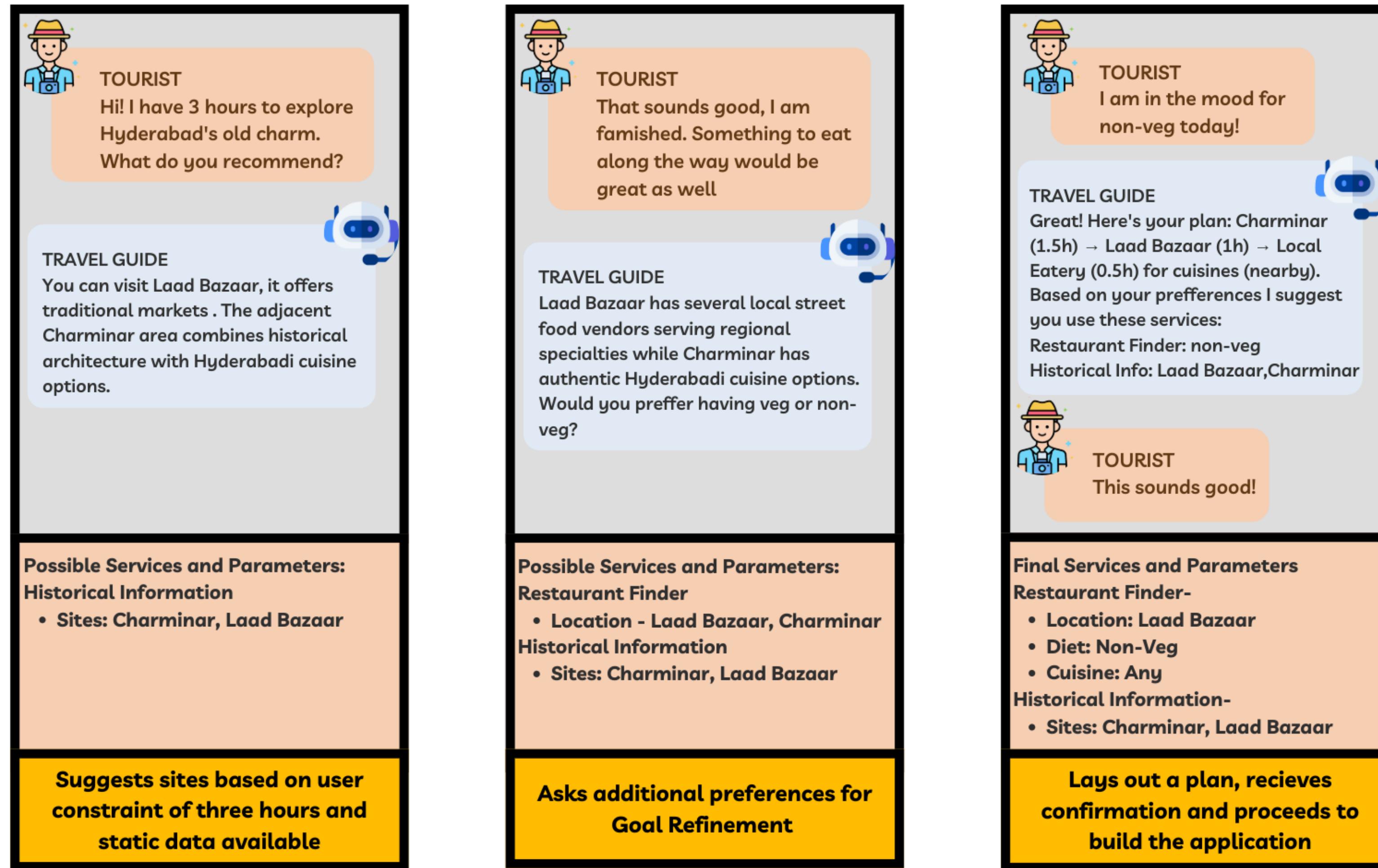
Applying to IoT Systems (Implementation and Deployment)

Dynamically generate services



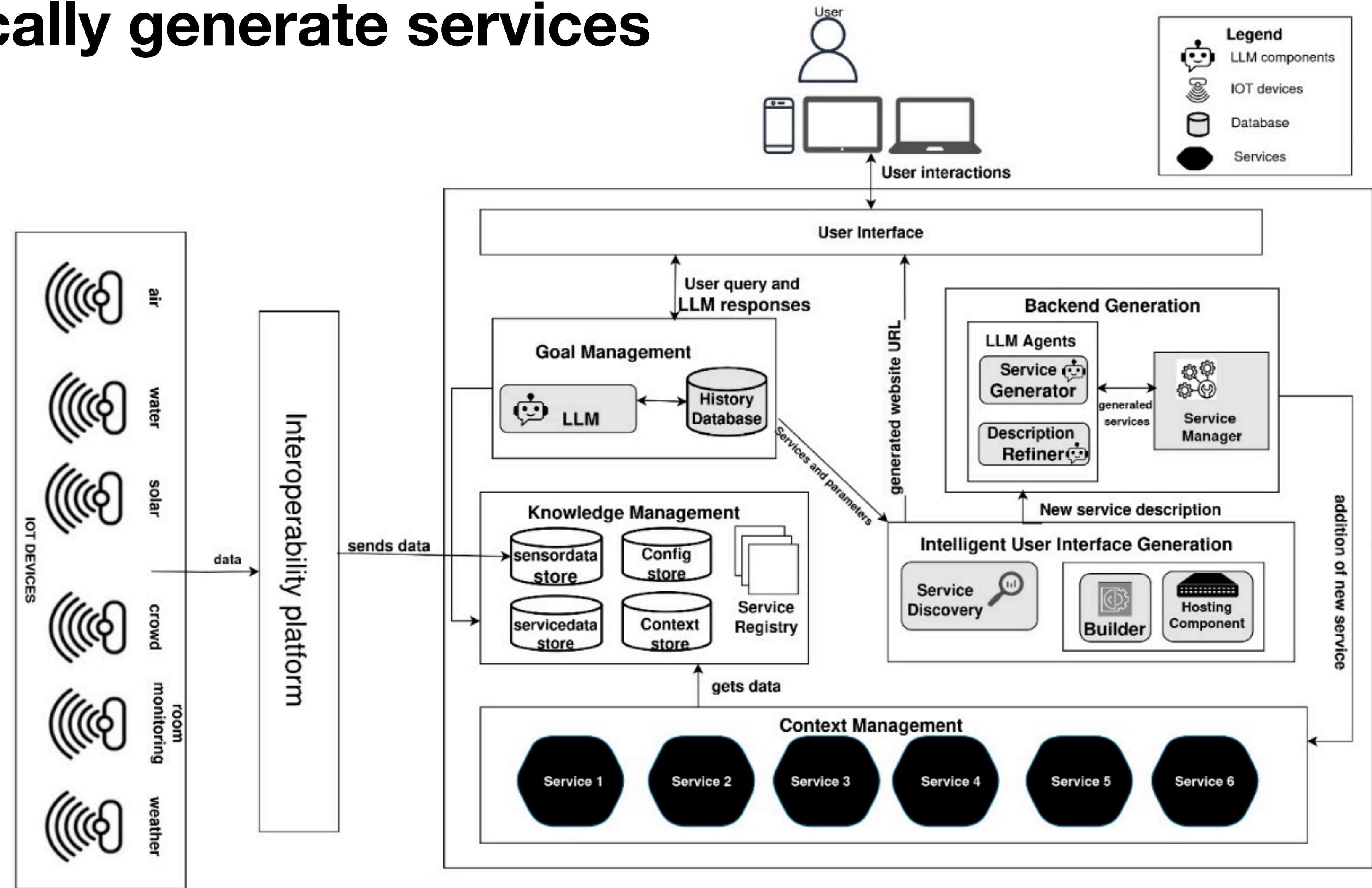
Applying to IoT Systems (Implementation and Deployment)

Dynamically generate services



IoT-Together: Mixed Initiative Interactions

Dynamically generate services





Website

Some Results

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

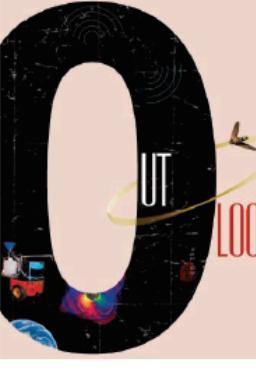
Metric	Mean ± SD	Min	Max
Total Duration (s)	23.10 ± 6.47	13.46	33.08
Total Token Usage	8164.90 ± 2718.89	5531	13991
Build Time (ms)	4.85 ± 1.98	3.50	10.49

Extending Beyond Design and Generation: 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

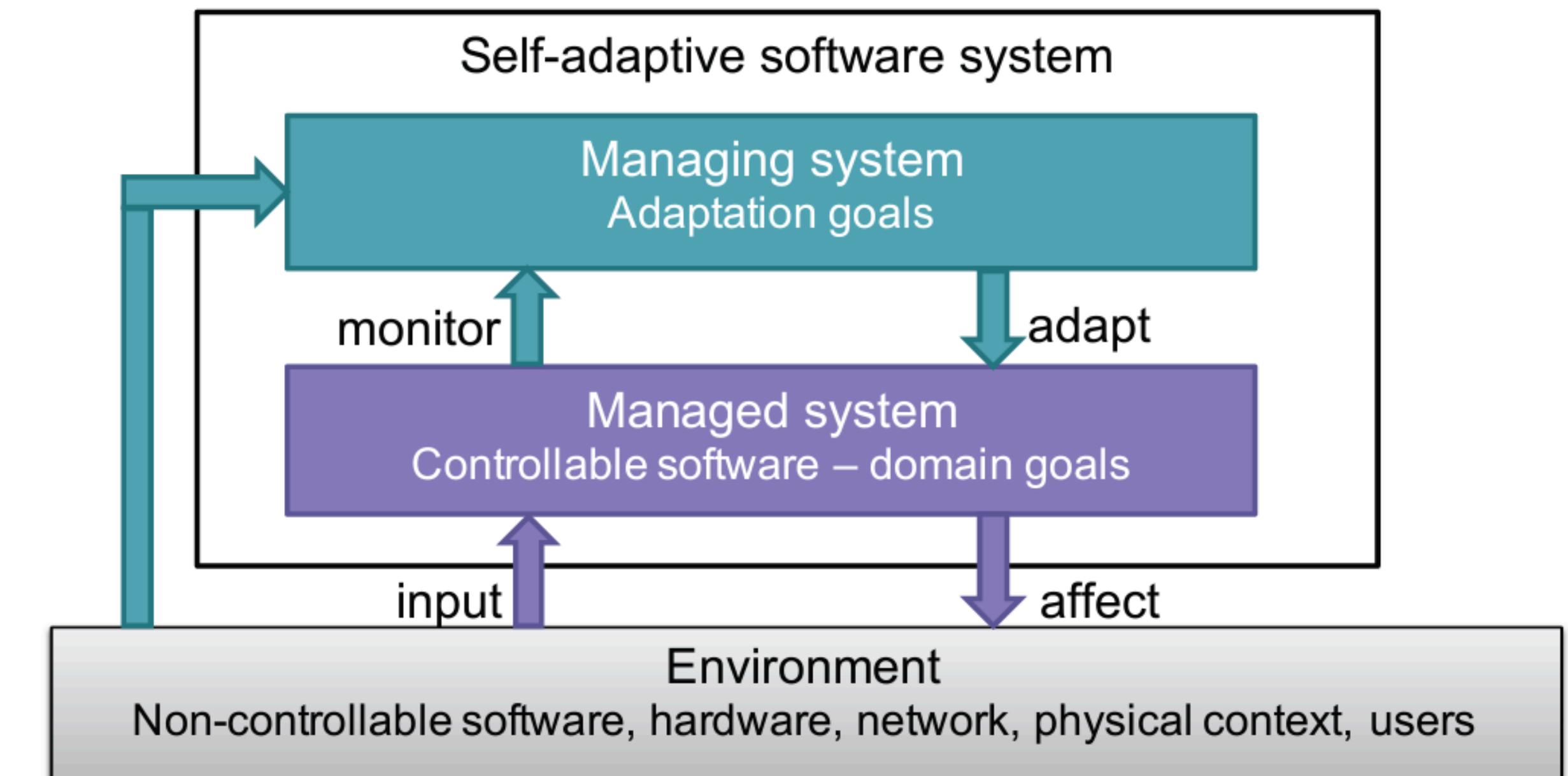
In mid-October 2001, IBM released a manifesto observing that the main obstacle to further progress in the IT industry is a looming software complexity crisis.¹ The company cited applications and environments that weigh in at tens of millions of lines of code and require skilled IT professionals to install,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

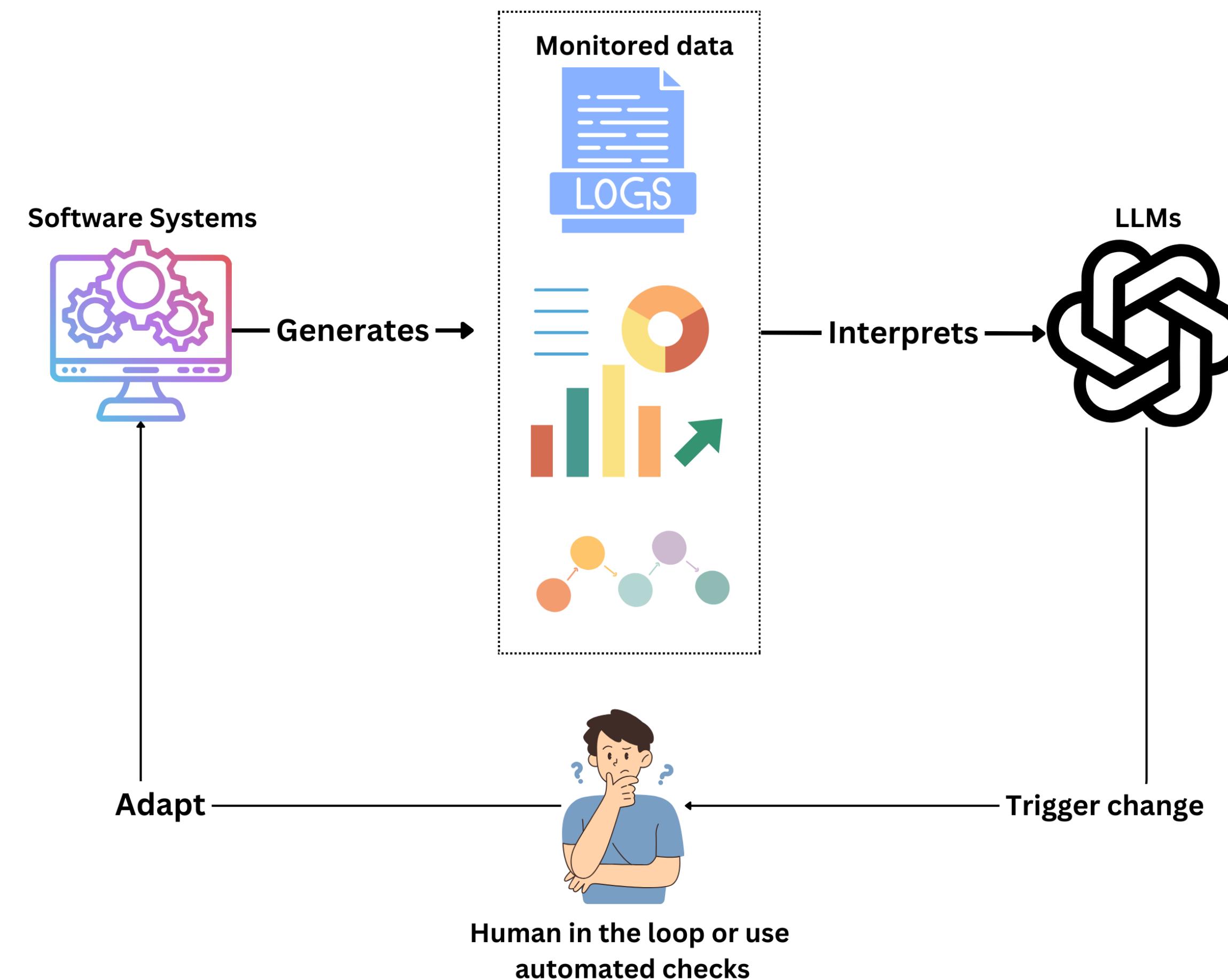
AUTONOMIC OPTION

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-

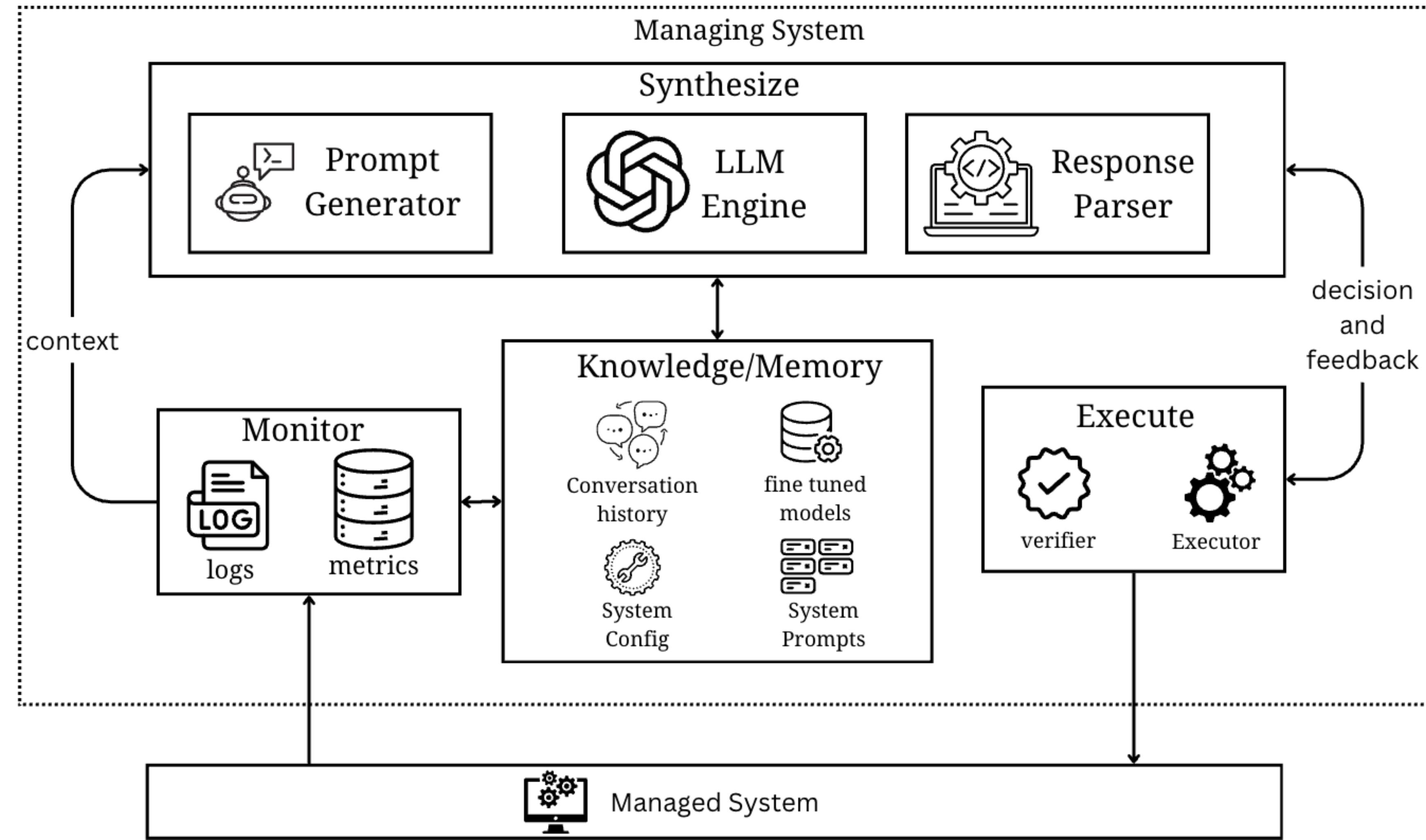


Extending beyond - Software Maintenance

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

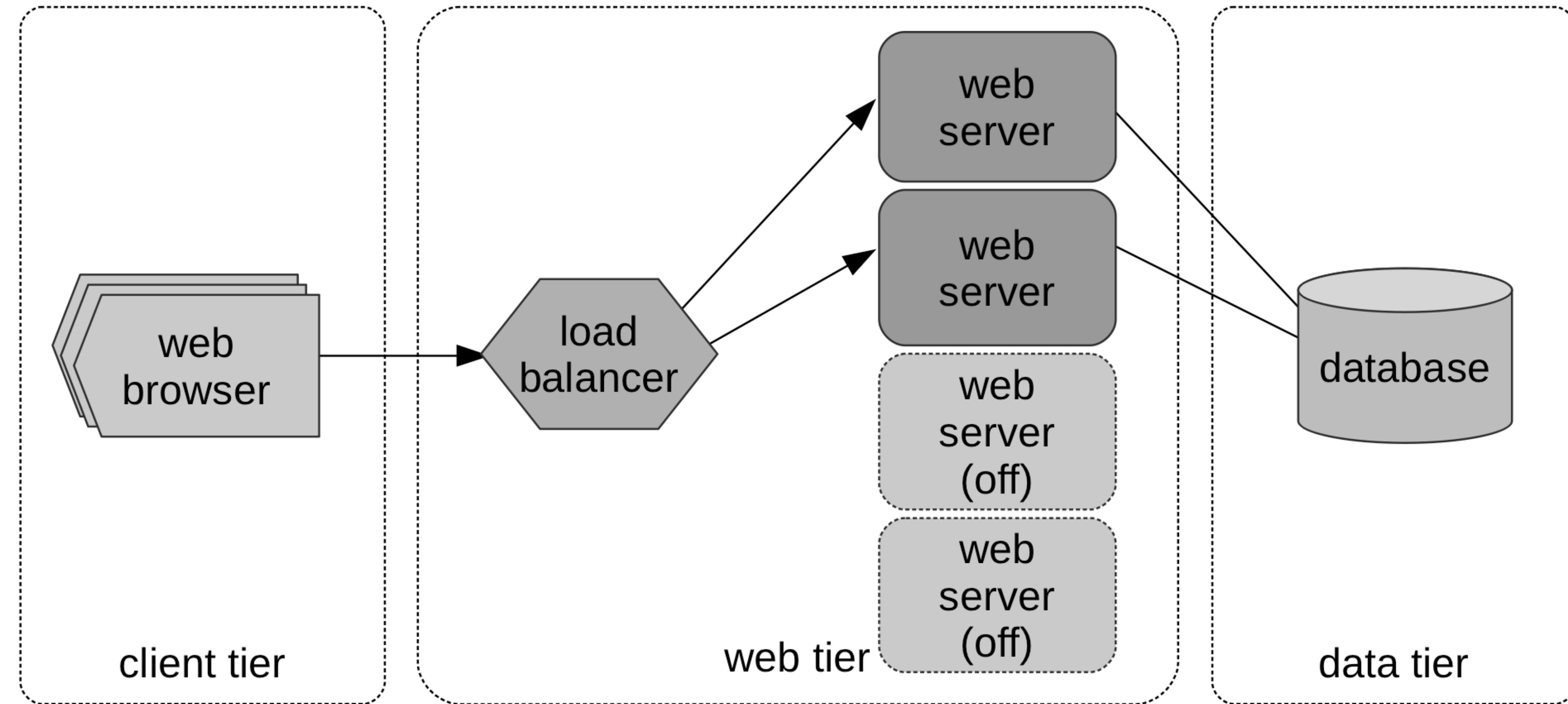


Reimagining Self-adaptation loop



SWIM case study

Web Infrastructure Simulator



Prompts used for SWIM

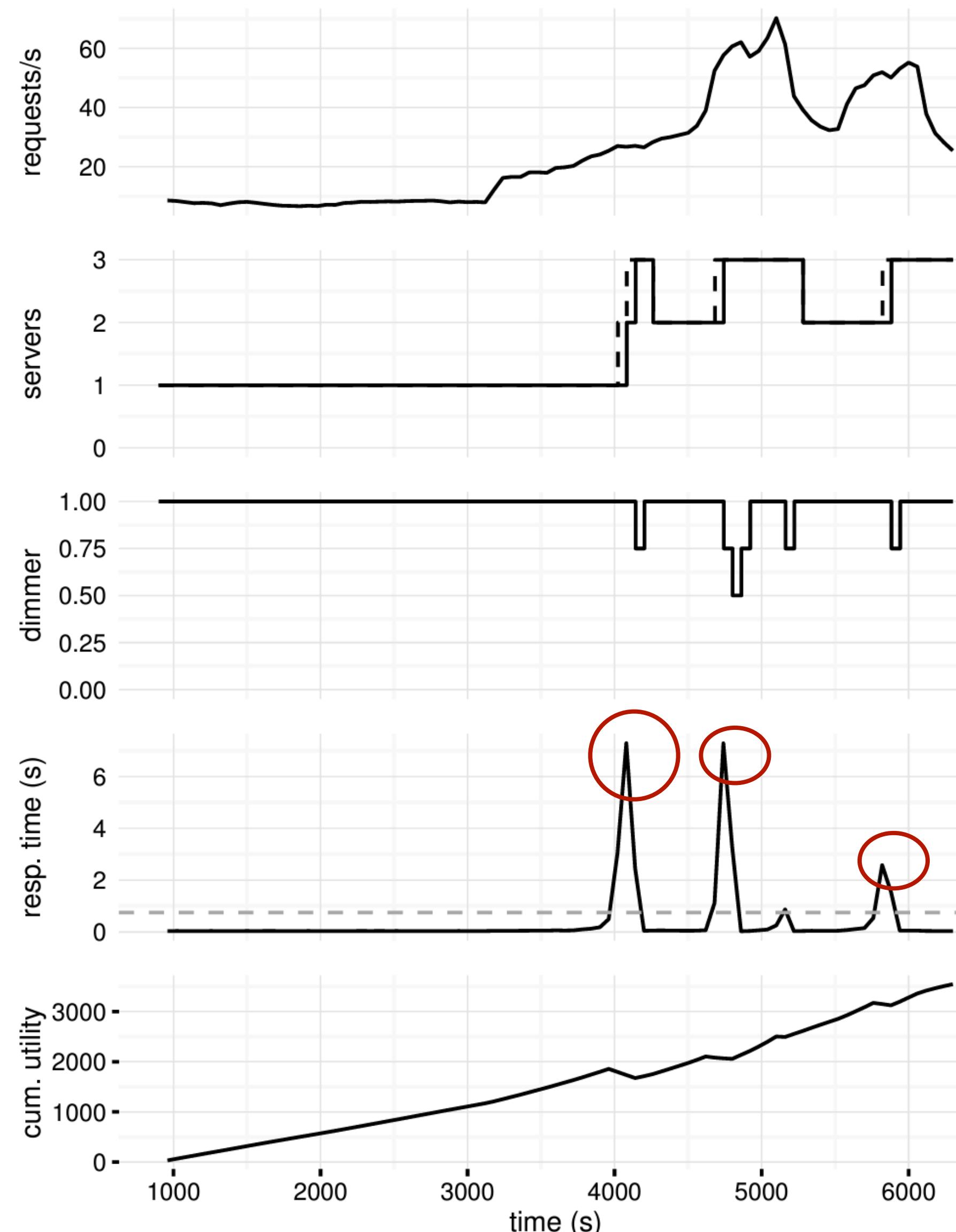
P_{SWIM}

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*

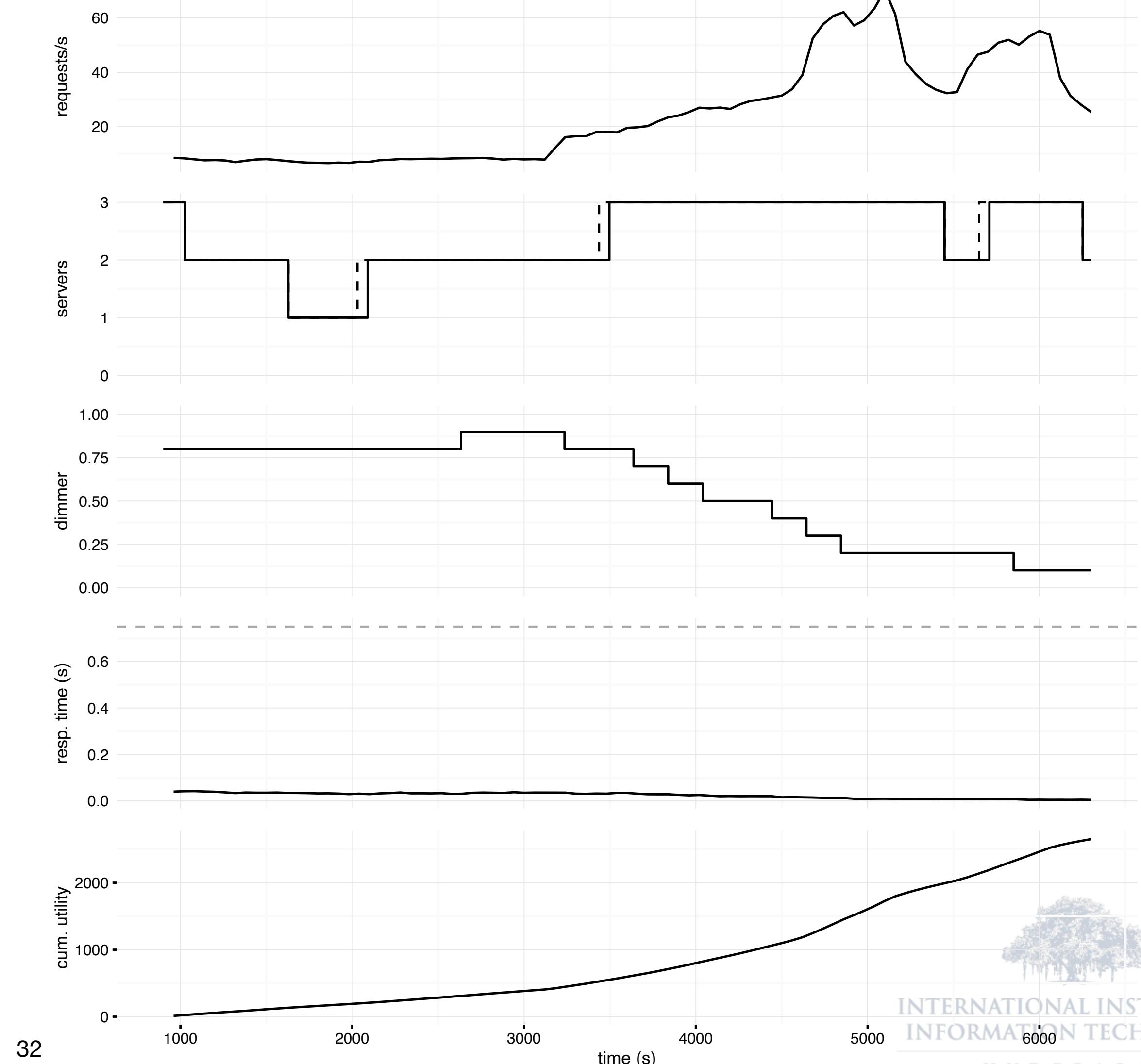
- *O*: Here the primary system objective is to keep the average response time as low as possible. Secondary objective is to keep dimmer as high as possible. Tertiary objective is to keep servers used as less as possible. These objectives are given in decreasing order of priority.
- *Terminologies*: Brief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

Some Initial Results

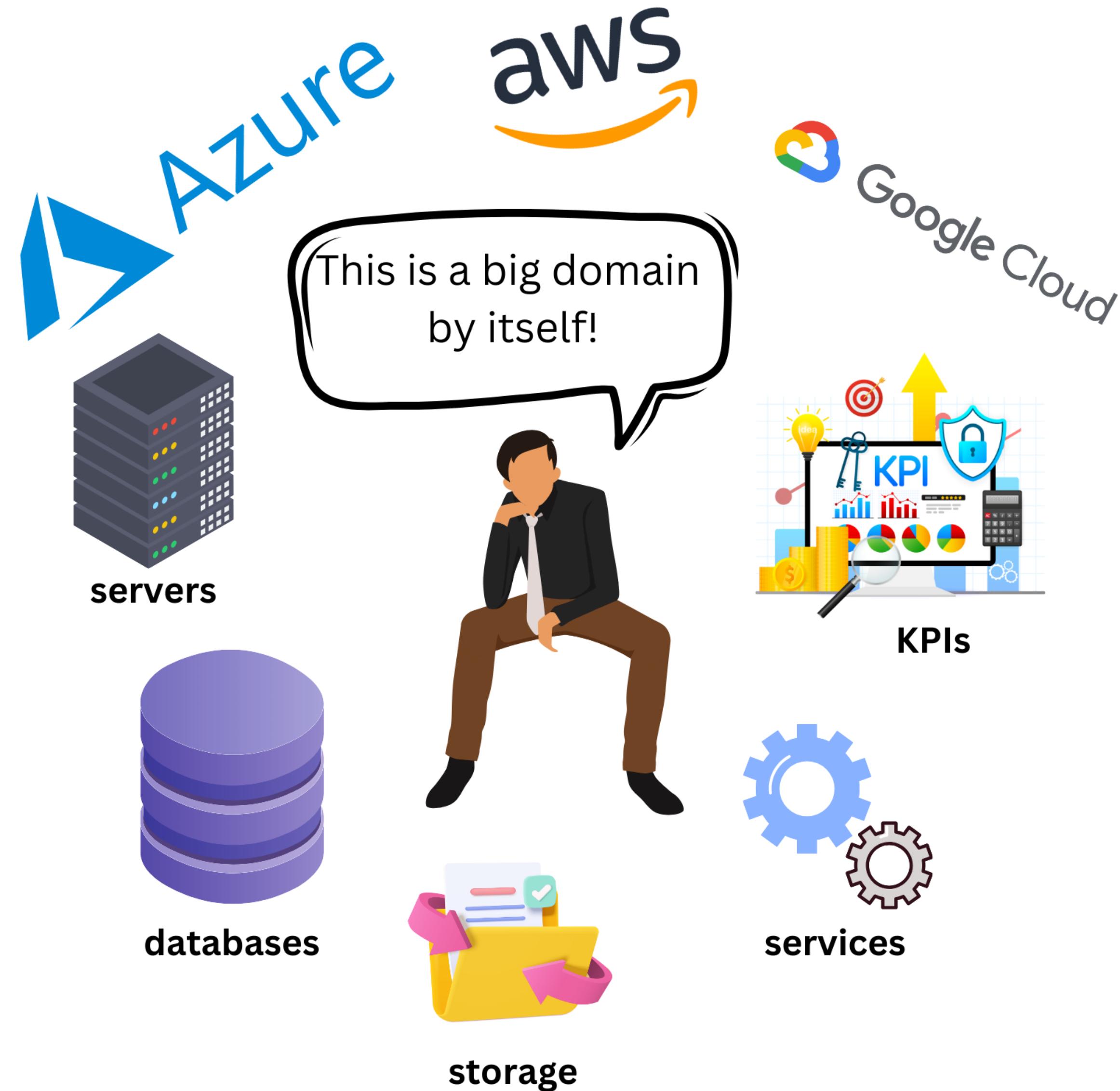
Using SWIM reactive adaptation



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

The screenshot displays the CloudOps Copilot dashboard, which integrates various cloud management tools. On the left, a sidebar lists categories like Home, Projects, Assessments, Inventory, Governance, Day2 CloudOps, Compliance, and Reports. The main dashboard features several cards:

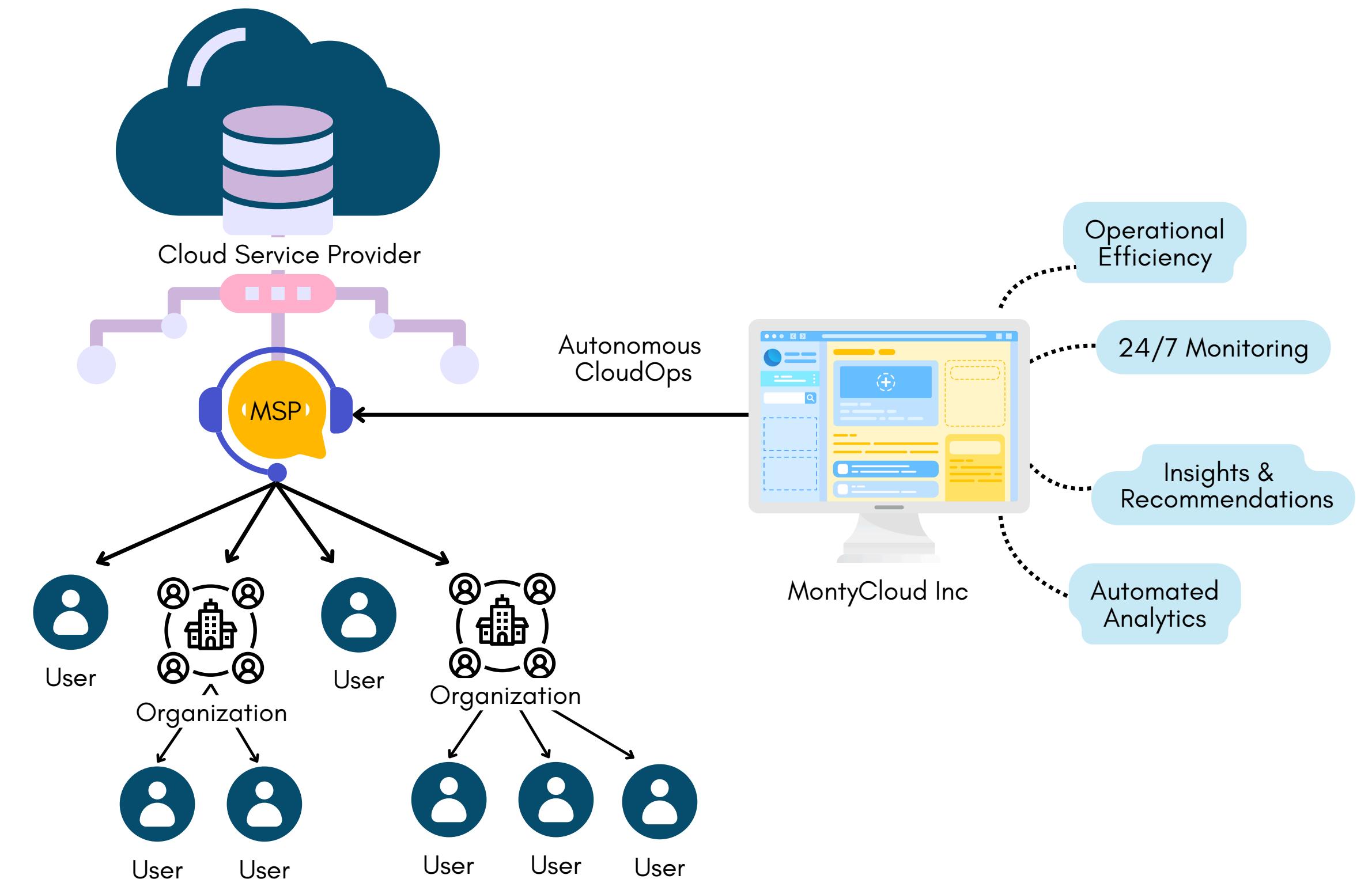
- AWS Costs:** Total Spend \$13,798.33, Cost by View (\$1456.27 for HR Department, \$400.22 for Dev Resources), Top Services (\$9,081.00 for Instance, \$4,081.00 for Volume, \$3,234.00 for Snapshots).
- Cost Optimization:** Last run 21 Nov 2023 10:13 AM, Potential Cost Savings \$1425, Over Provisioned 0, Under Provisioned 0, Abandoned Resources 64, Needs Optimization 0.
- Cloud Footprint:** A world map showing the distribution of resources across regions, with counts ranging from 61 to 147.
- Security Posture:** Last run 21-Nov-23 09:41, 190 Open Ops Issues, Remediations 80 By DAY2™, Recommendations 25 Security, 15 Compliance.
- Compliance Assessment:** Last run 21-Nov-23 09:59, 40 Open Issues, Compliance Bot ACTIVE, Industry Standards (HIPAA 83%, CIS 67%, FedRAMP 91%, NIST 83%, PCI 100%).

On the right, a sidebar titled "Top Resources" lists various resource types and their counts: Compute Instance (3005), Image (200), EBS Volume (35), VPC Endpoint (30), Virtual Private Cloud (30), EBS Snapshot (20), and SNS Topic (15).

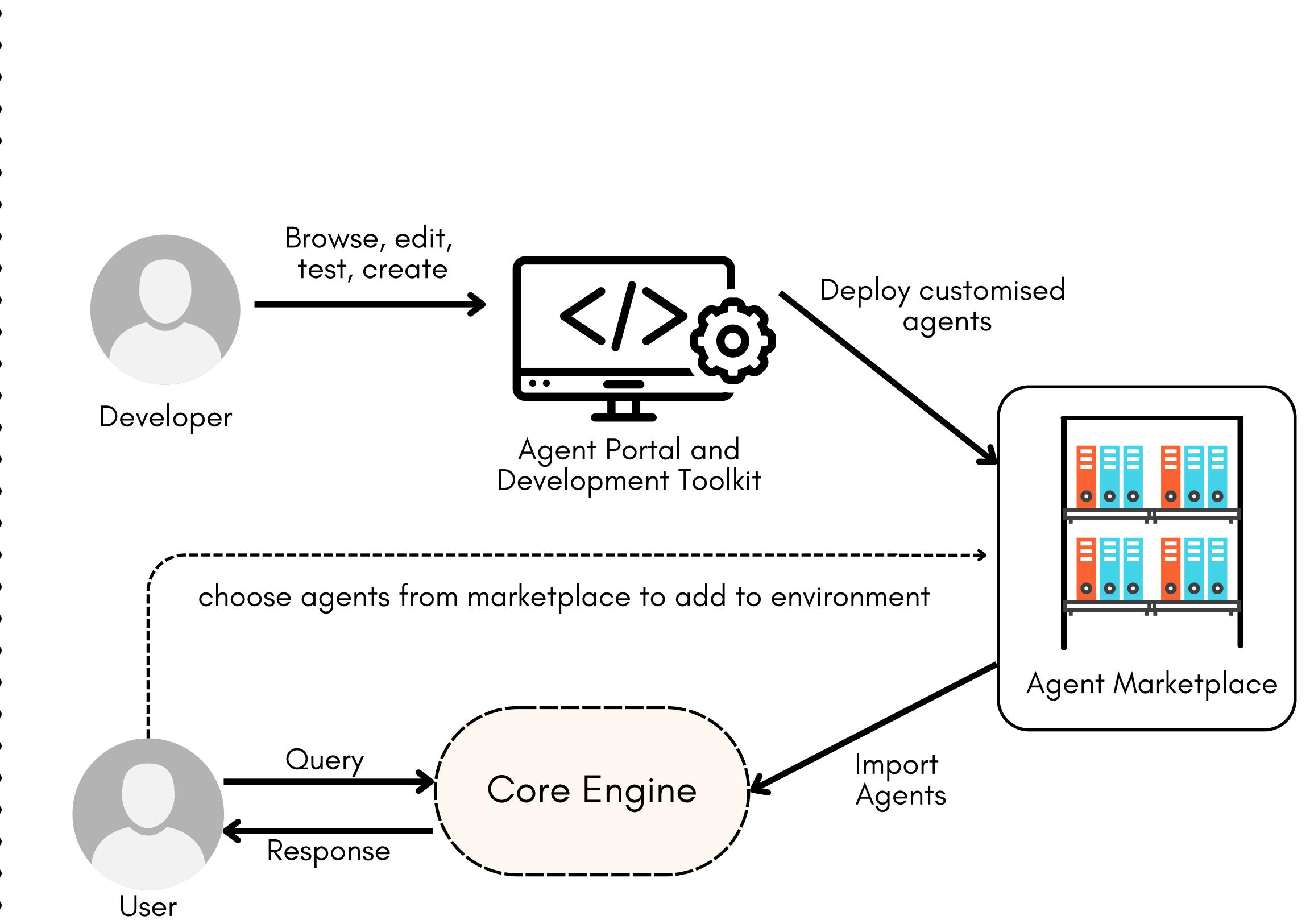
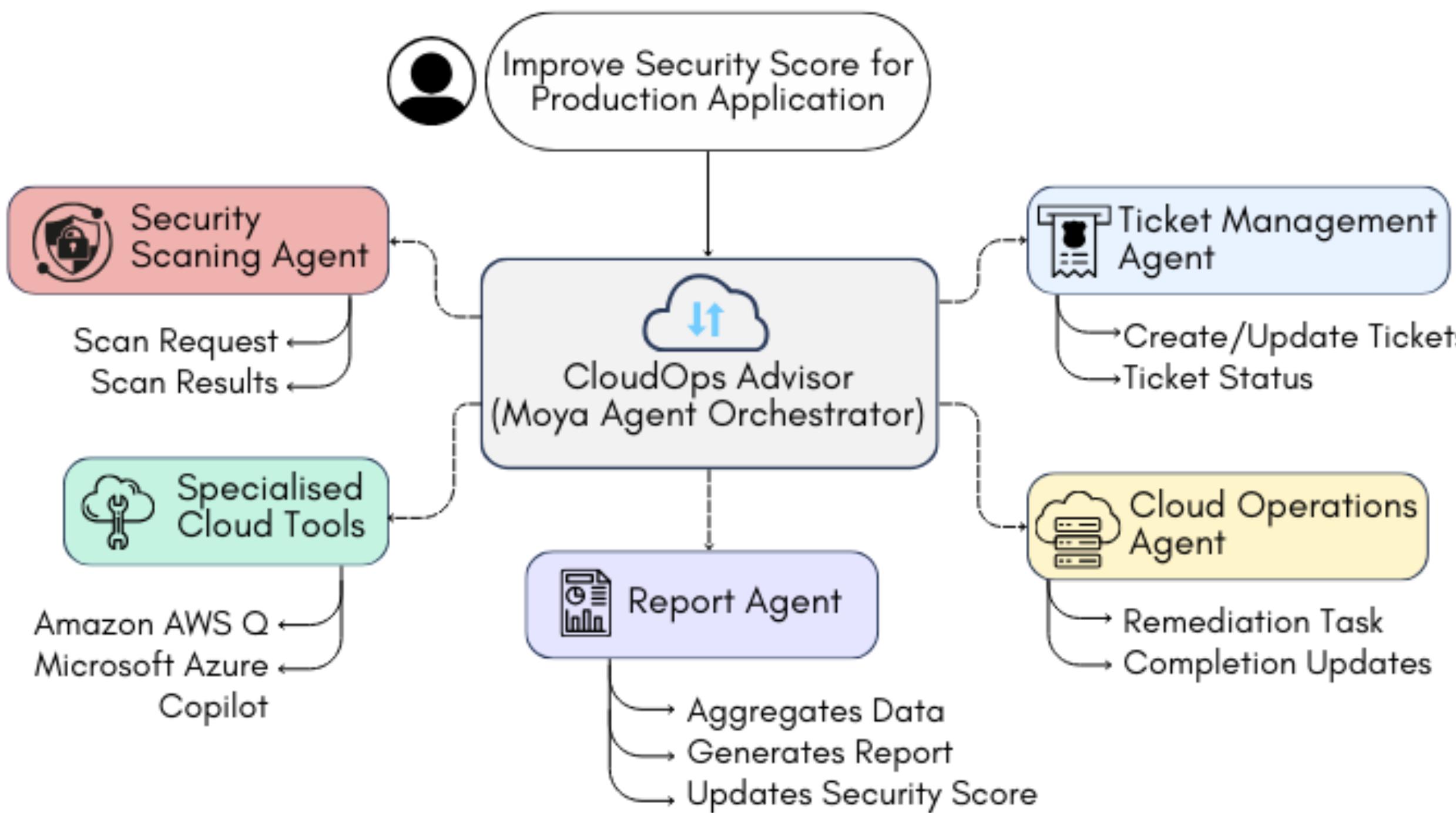
Work done in collaboration with MontyCloud Inc.

Engineering Challenges

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

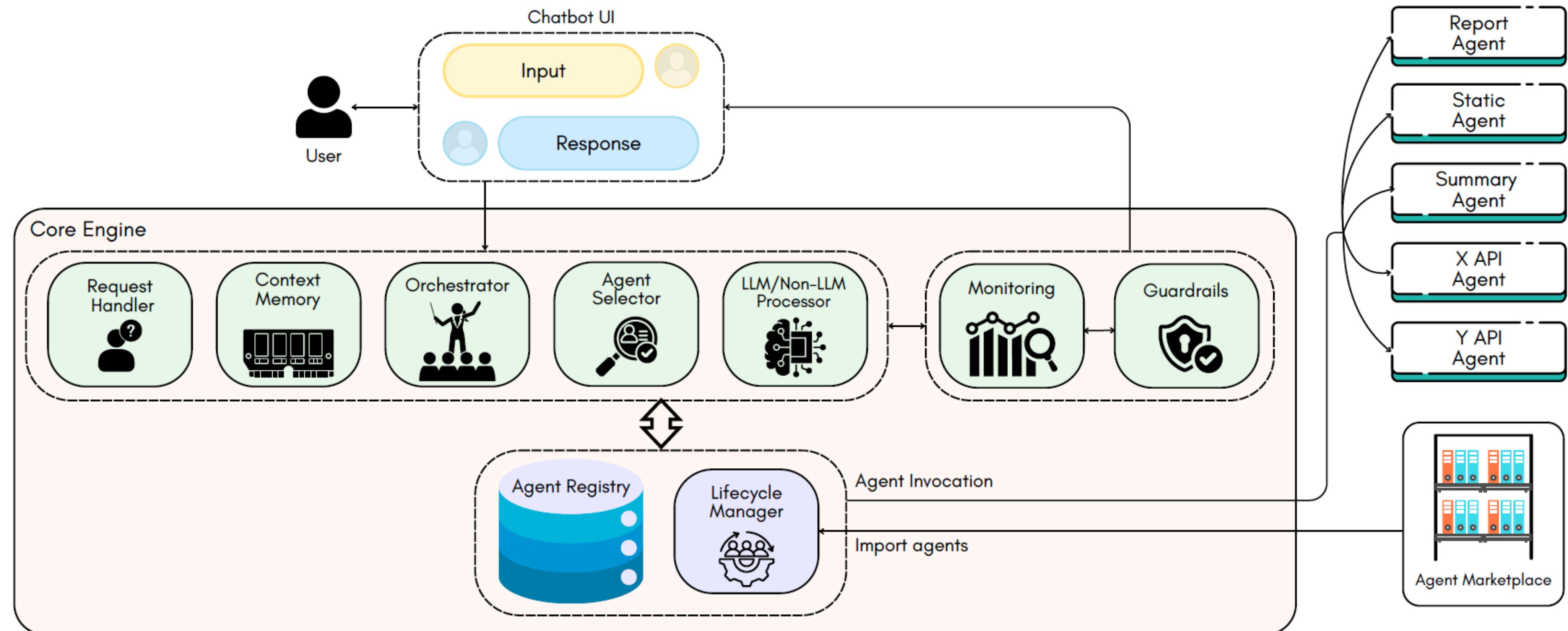


Can we go Multi-agent?



Meta orchestration Framework

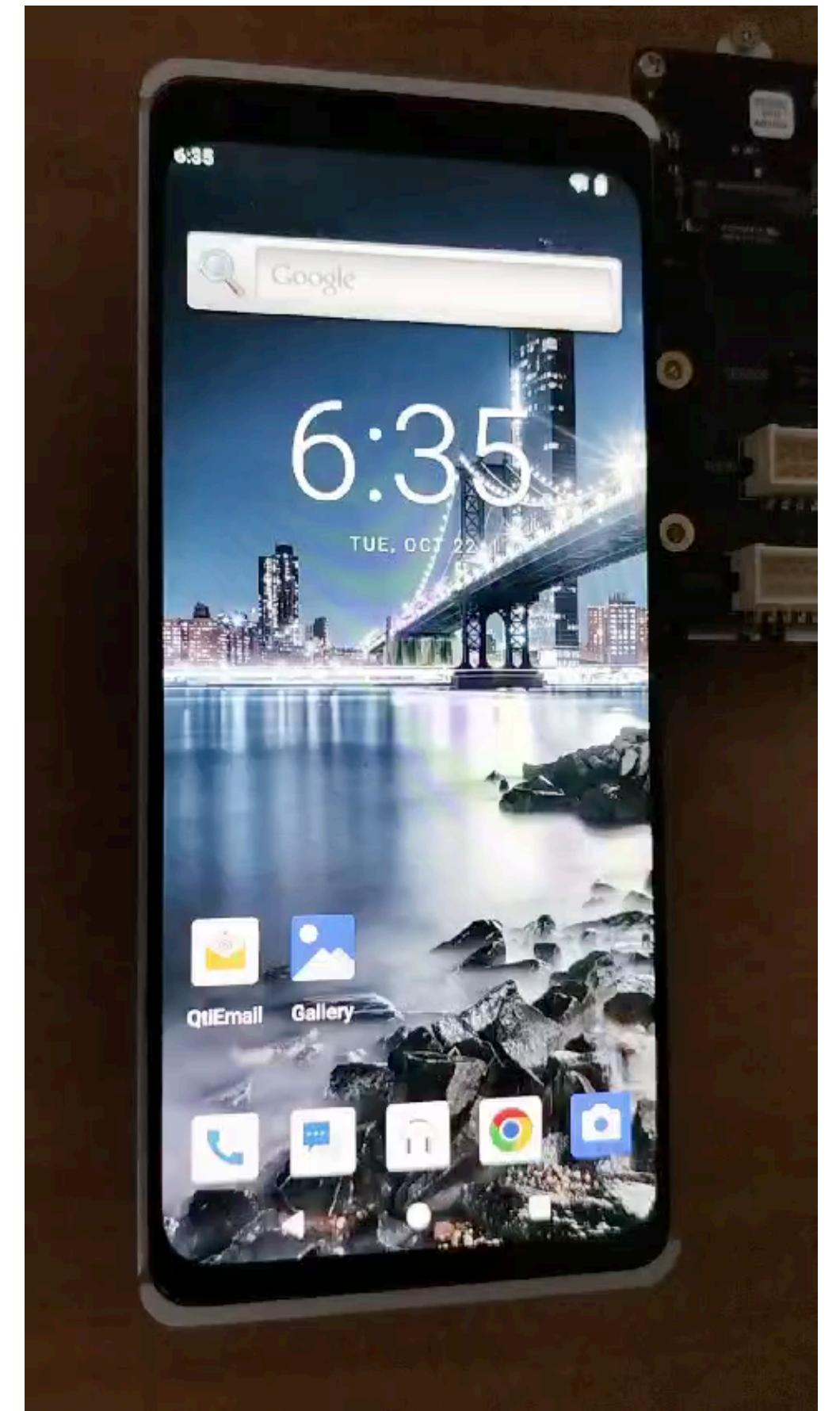
The MOYA Multi-agent Framework

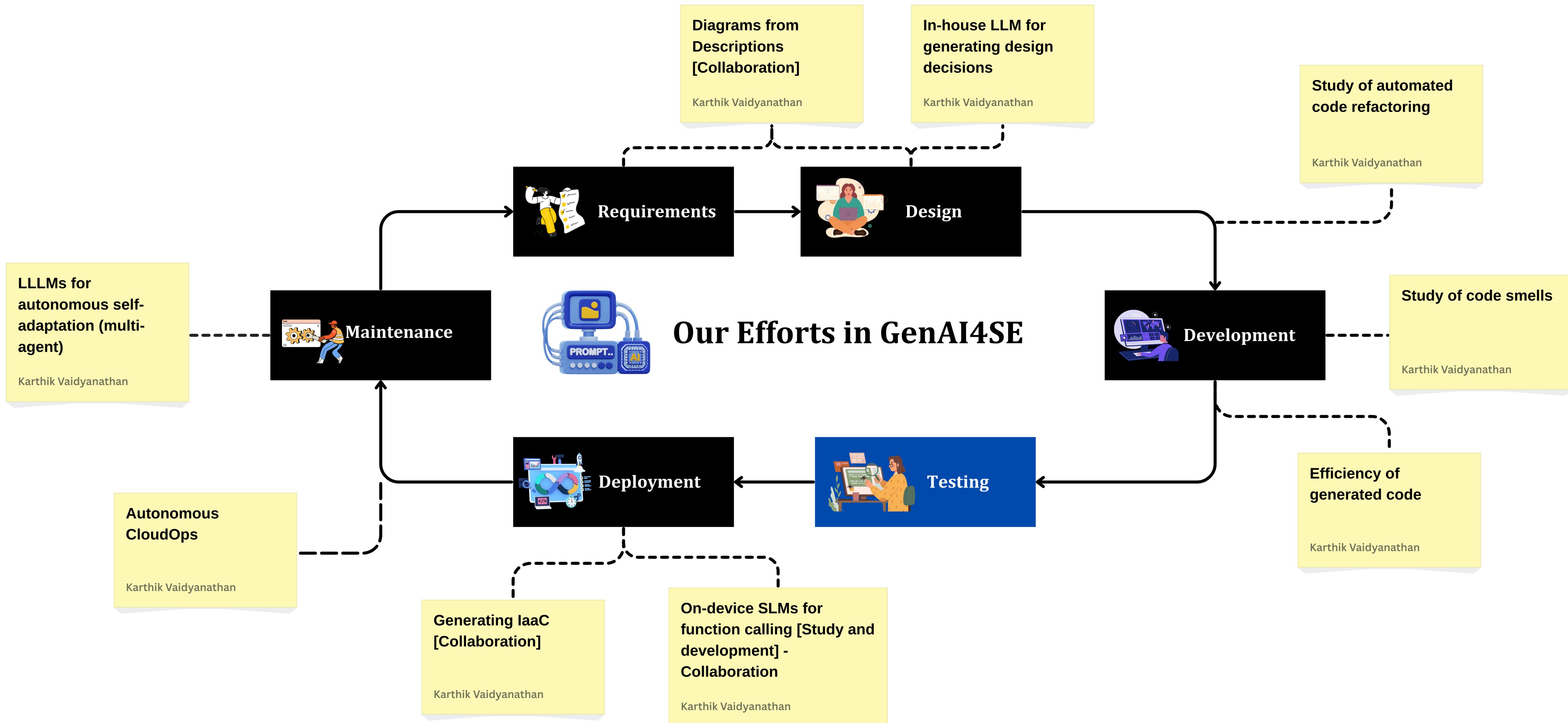


Into the world of SLMs for SE

Qualcomm

- 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)
 -



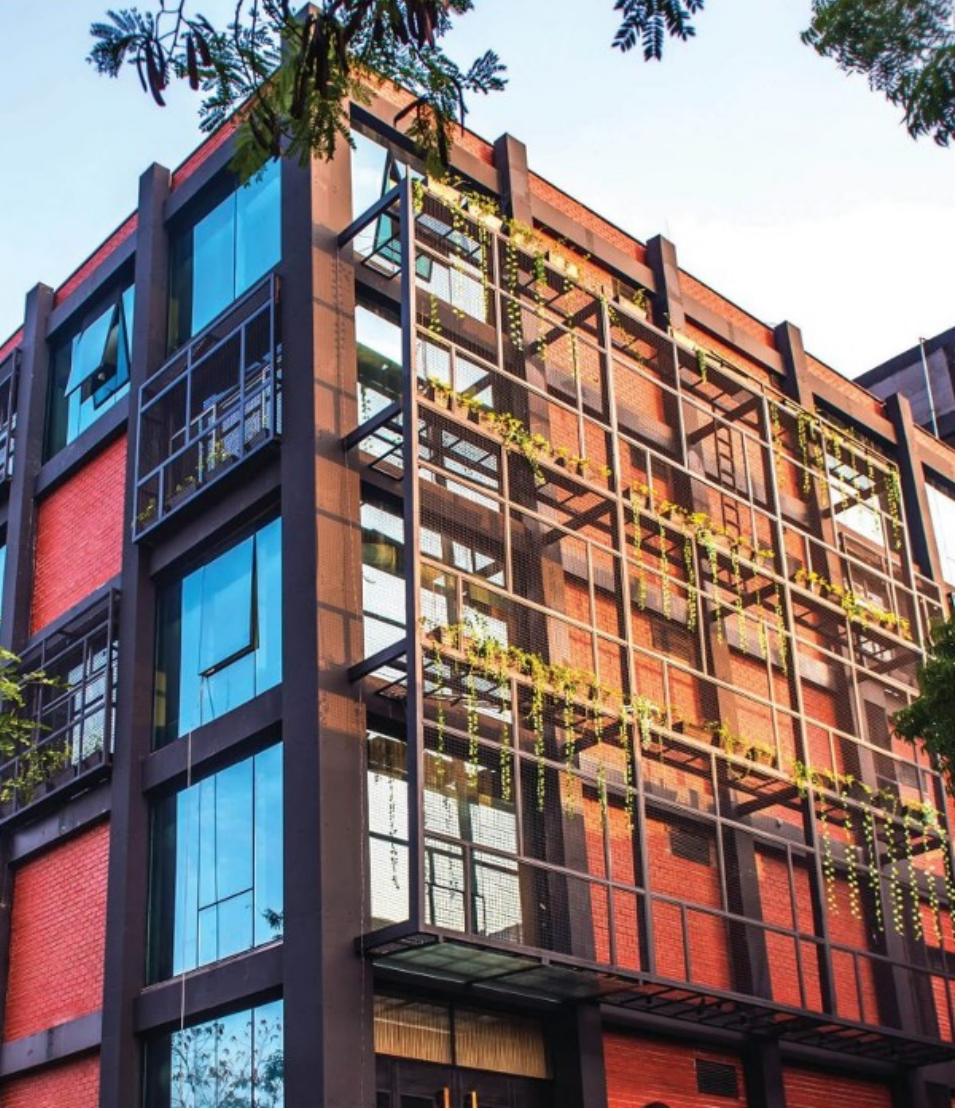
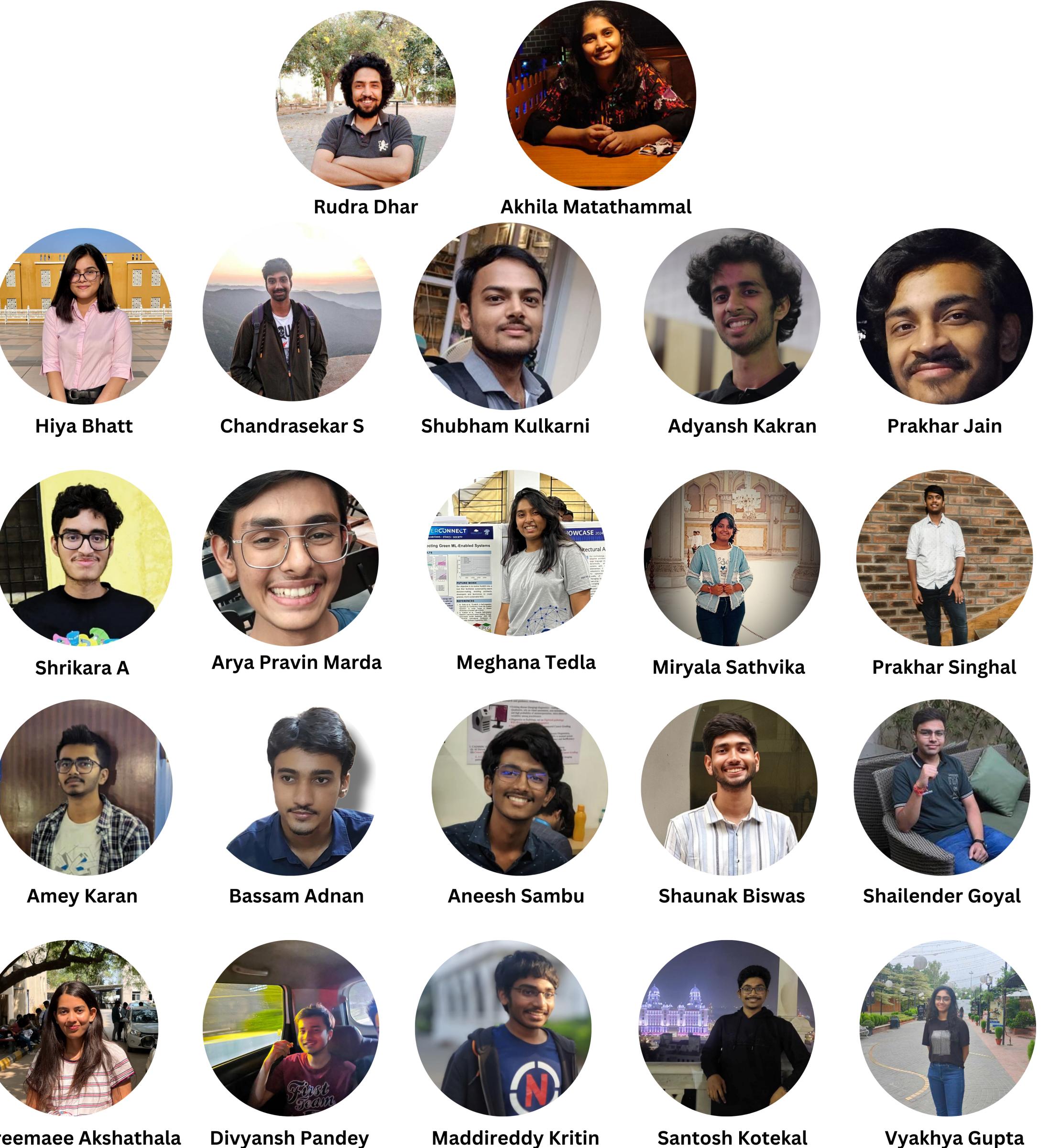


The Future is here

- **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!!**



SA4S@SERC



<https://serc.iiit.ac.in>



Team SA4S

<https://sa4s-serc.github.io>



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

@ ICSA 2025, SAGAI 2025



Thank you

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



IEEE Software Magazine



SE Radio Podcasts

