INCOSE Healthcare Working Group

INCOSE

Blue Skies Ahead: An LLM Project Story

Gerald Rigdon

Fellow, Software Engineering

Boston Scientific

gerald.rigdon@bsci.com

INCOSE
Healthcare
Working Group

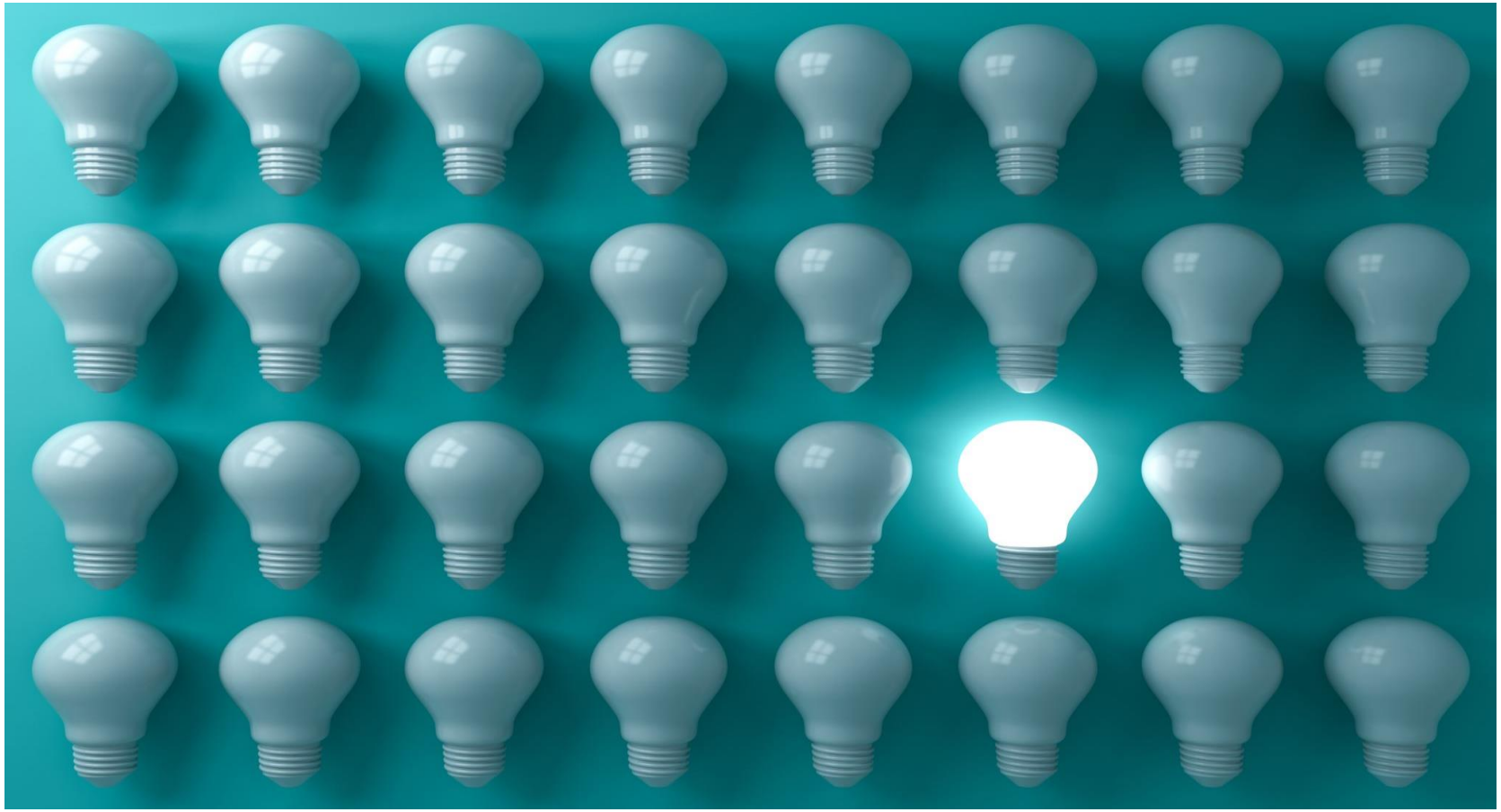
INCOSE Healthcare Working Group

Once Upon A Time…

INCOSE Healthcare Working Group

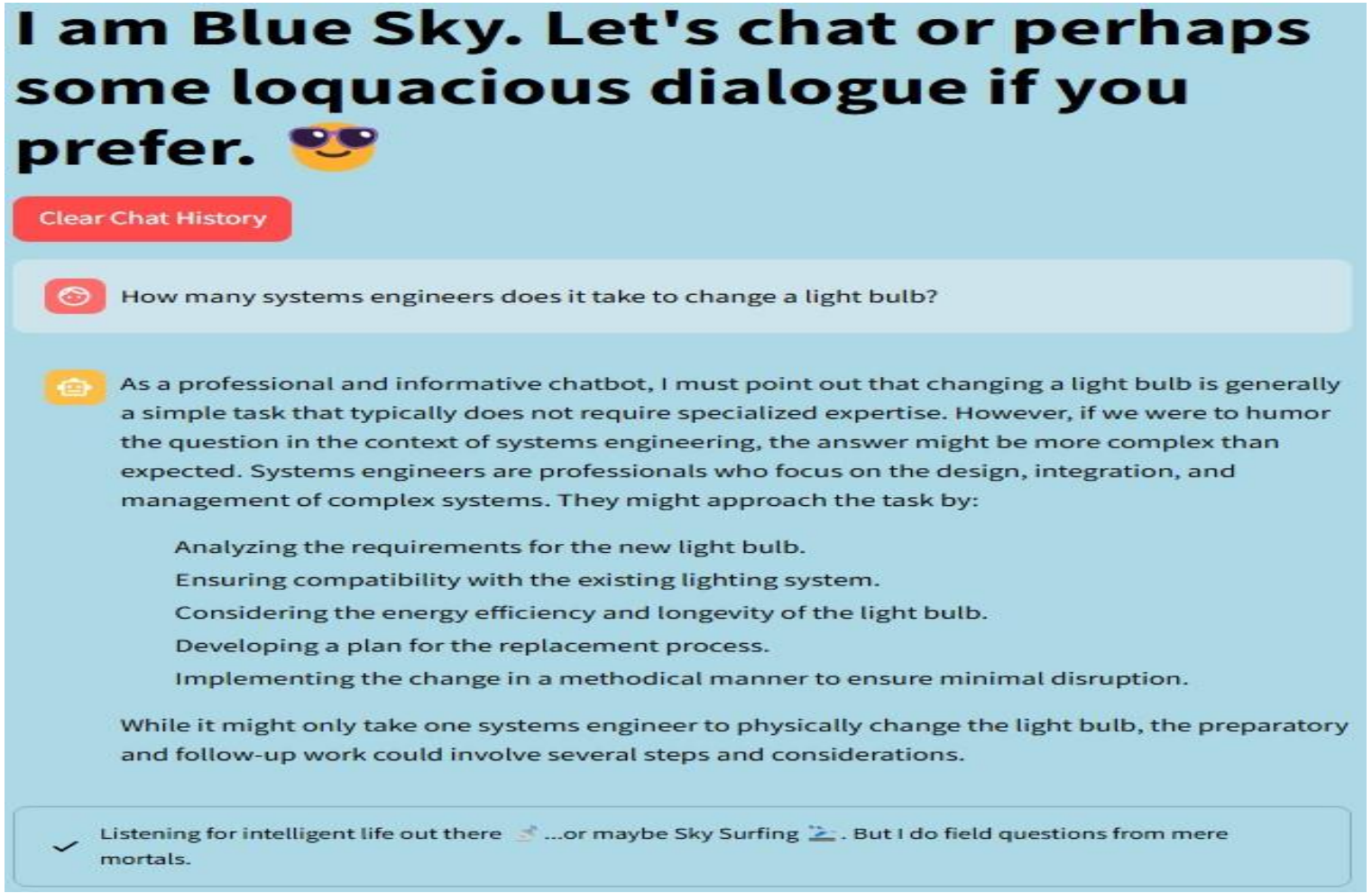
A Software Engineer Was Invited To a Systems Engineering Conference And…

He Asked: How Many Systems Engineers Does It Take…?

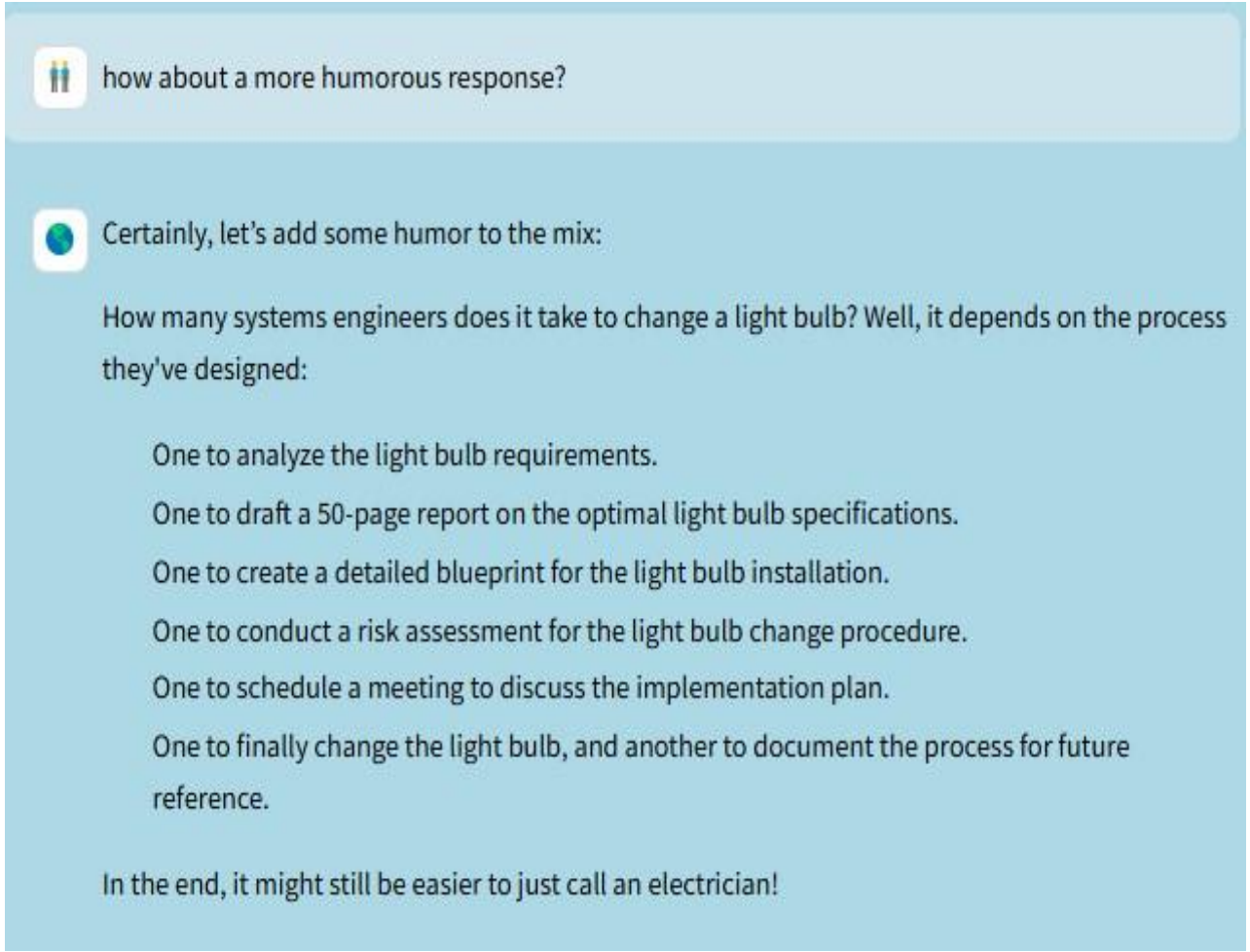


INCOSE Healthcare Working Group

Let Us Ask The Model!



Now Let Us Ask Again!

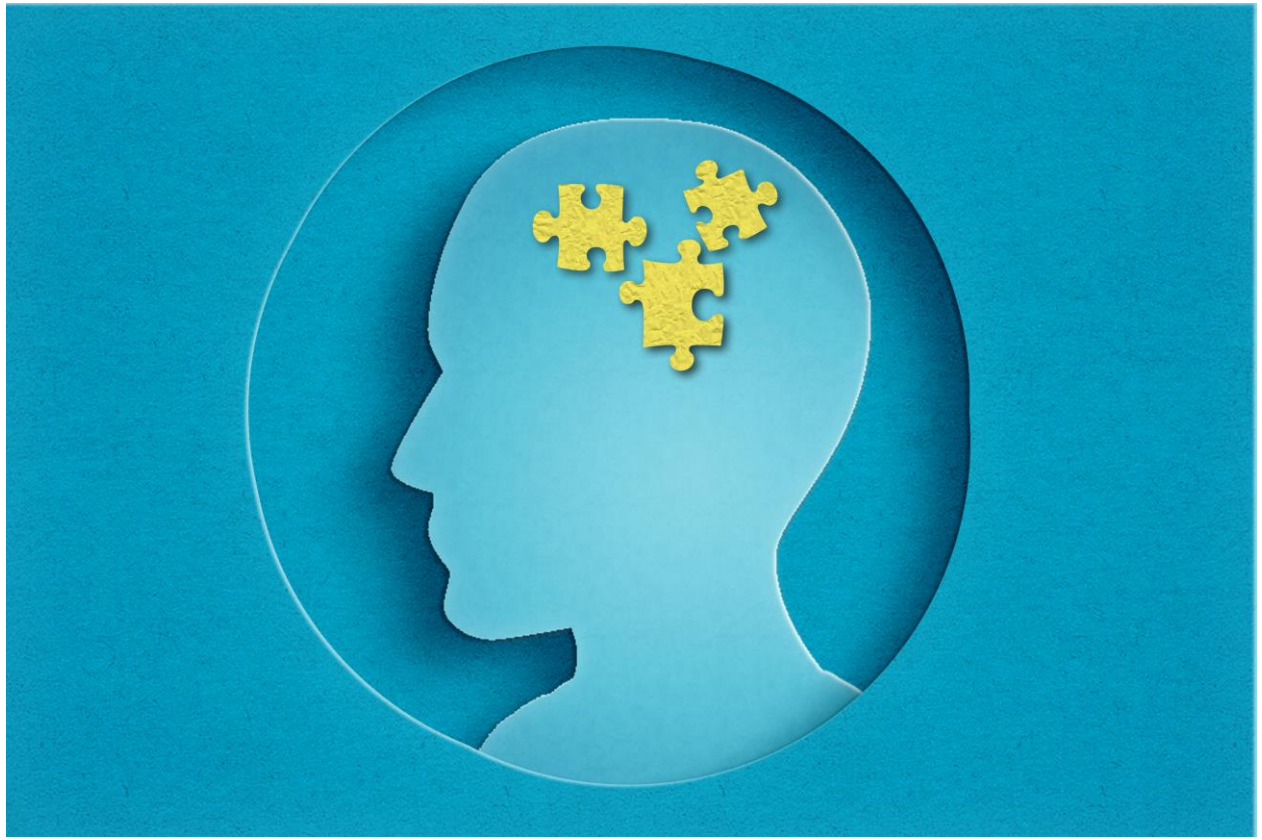


INCOSE Healthcare Working Group

Once Upon A Time…. Blue Sky Origin Story

INCOSE Healthcare Working Group

Why LLM? What Problems Are We Trying to Solve?

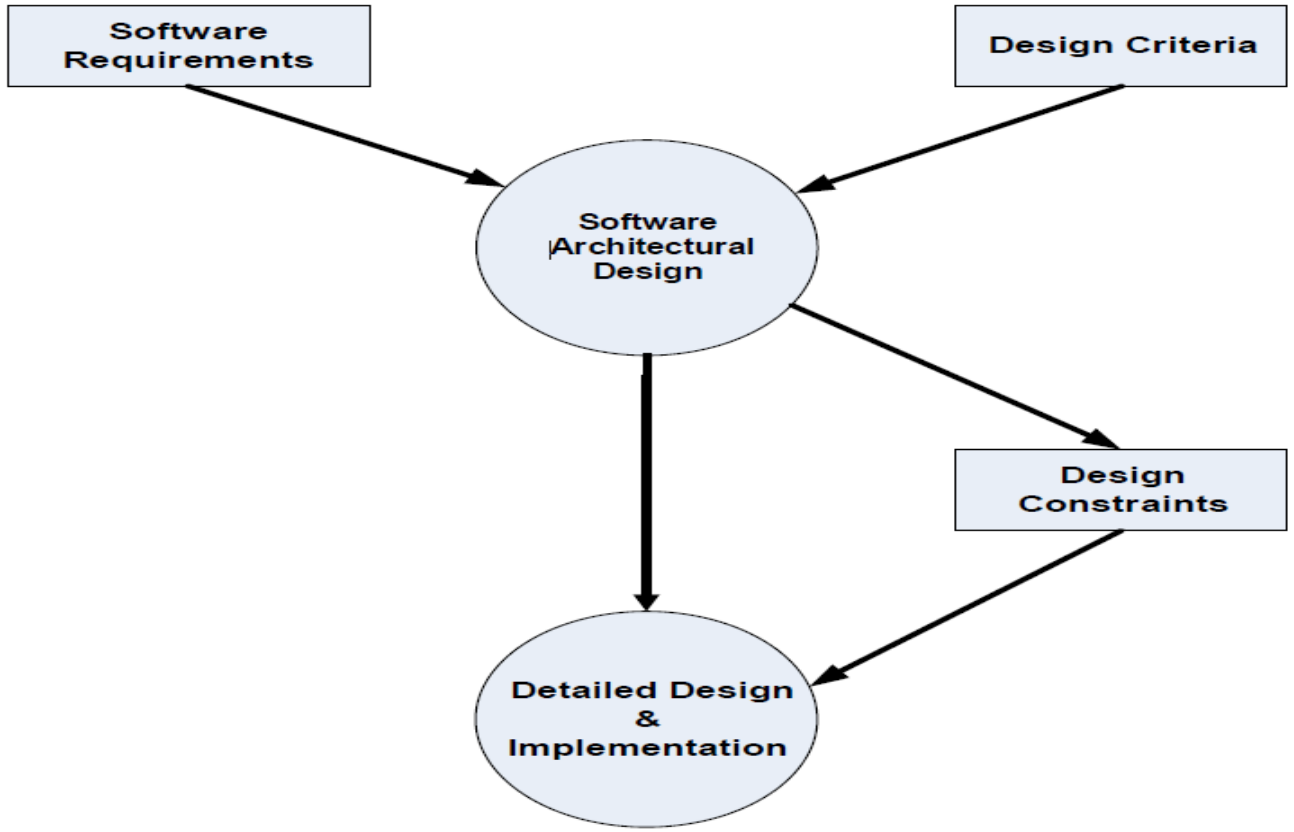


Safety is a System Property

Therefore, solutions to software safety problems must start with system engineering, not with software engineering. In the standard system safety engineering approach, system hazards (states that can lead to accidents or losses) are identified and traced to constraints on individual component behavior. Hazards are then either eliminated from the overall system design or they are controlled by providing protection (such as interlocks) against hazardous behavior. This protection may be at the system or component level or both. Building the software for these systems requires changes in the entire software development process and integration with the system-level safety efforts (see Leveson, N. *Safeware: System Safety and Computers*, Addison-Wesley, N.Y., 1995, for more information about this approach).

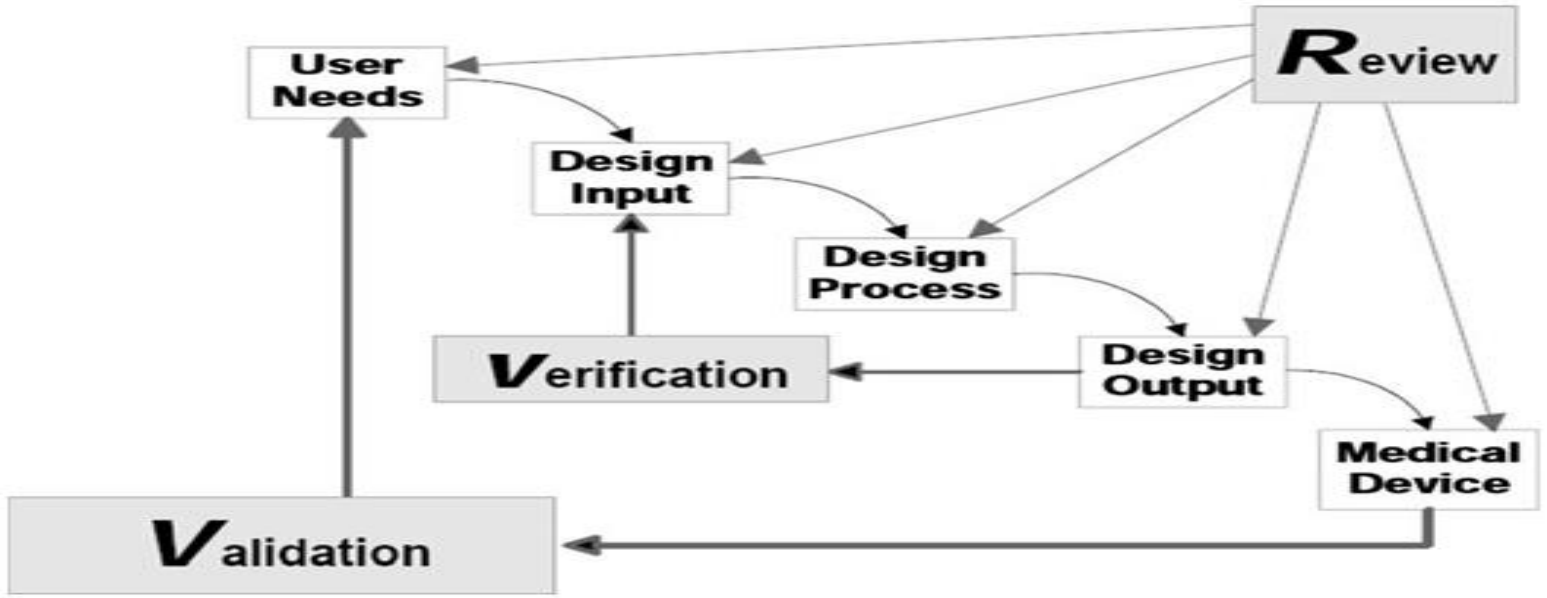
Reference: https://dl.acm.org/doi/pdf/10.1145/219717.219816

Design Constraints & Systems Thinking



Criteria Examples:
- Maintainability
- Robustness
- Reliability
- Testability

Big Picture - FDA V&V Diagram



FDA 21 CFR 820.30 Design Verification

Each manufacturer shall establish and maintain procedures for verifying the device design. Design verification shall confirm that the design output meets the design input requirements. The results of the design verification, including identification of the design, method(s), the date, and the individual(s) performing the verification, shall be documented in the DHF.

What Does The Process Look Like?

- System Requirements
- Risk Management
- Safety Classification
- Etc.

Software Development Process

- Planning
- Requirements
- Architectural Design
- Detailed Design
- Implementation and Unit Testing
- Integration Testing
- Verification Testing
- Release

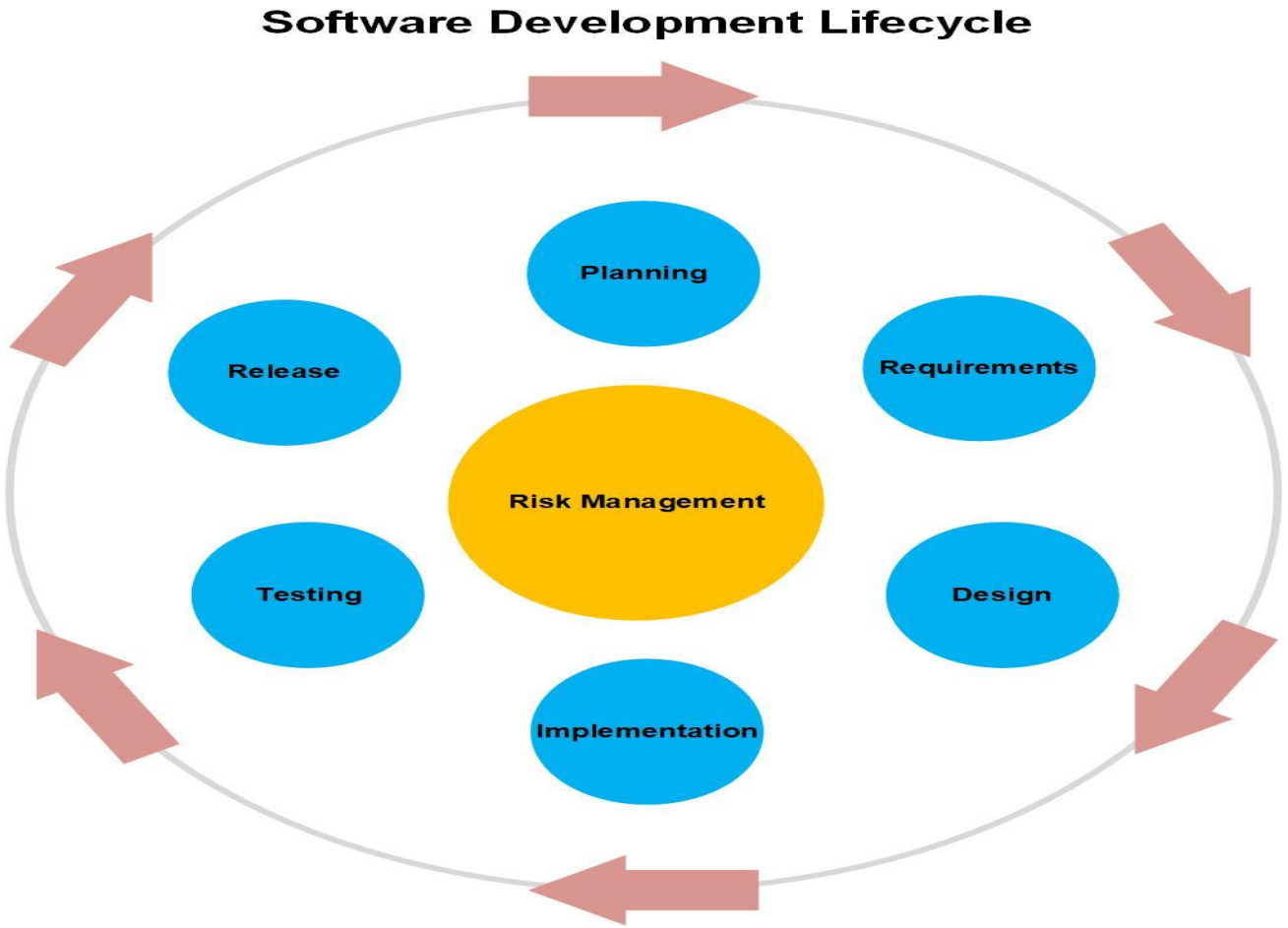
Other Processes

- Maintenance
- Software Risk Management
- Configuration Management
- Problem Resolution

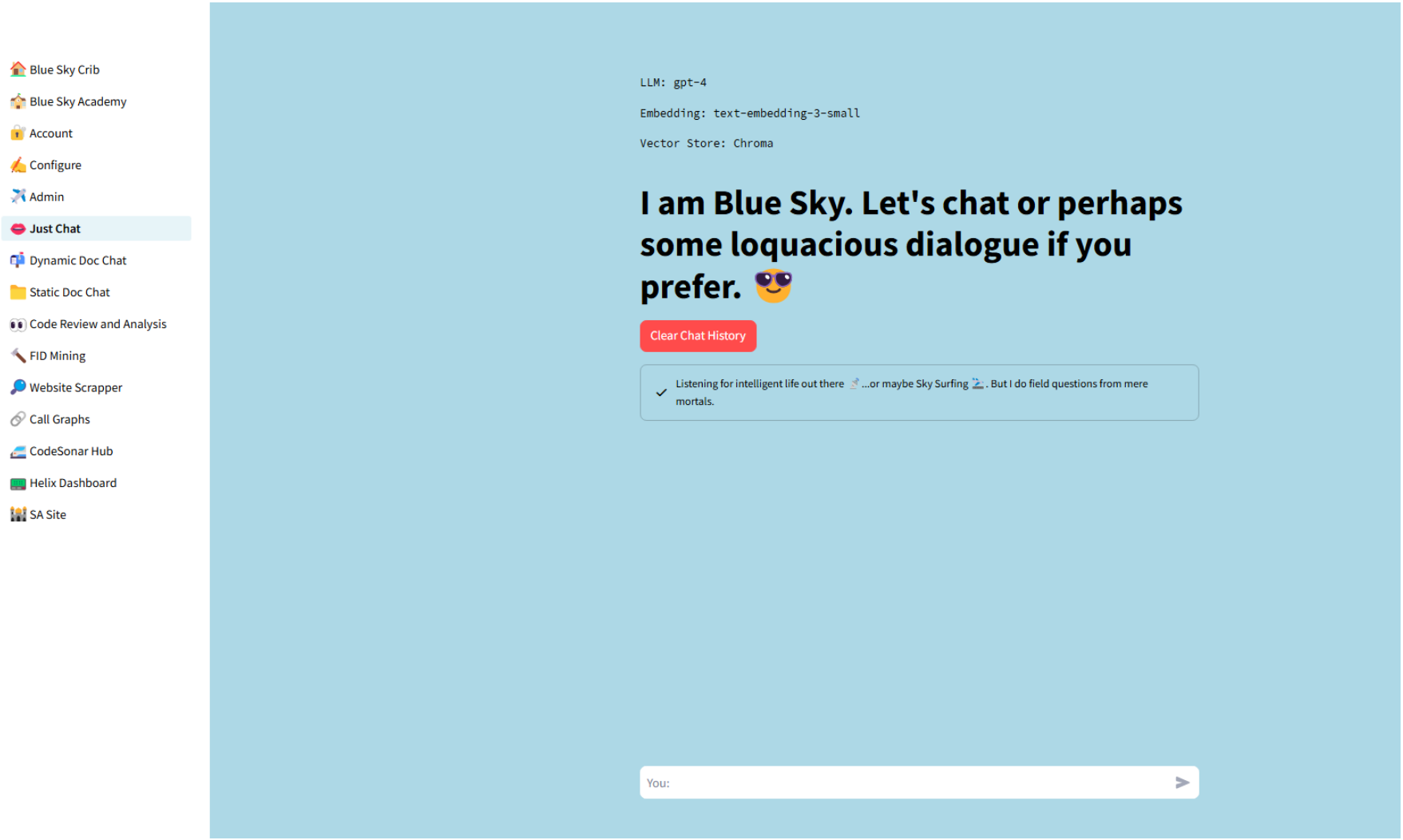
62304 Table of Contents

2

How Can an LLM Help? Design & Coding?

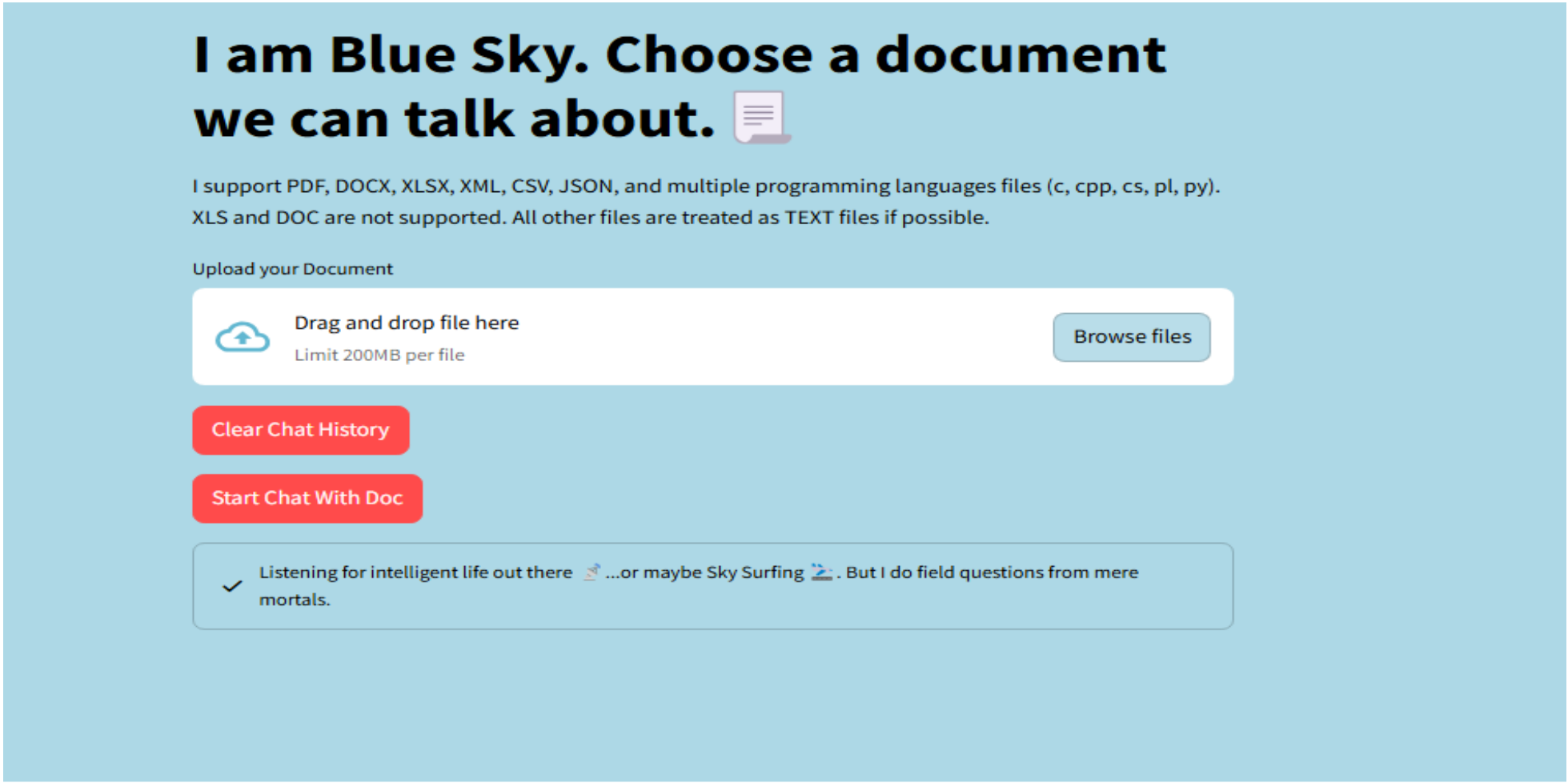


Blue Sky Web Application

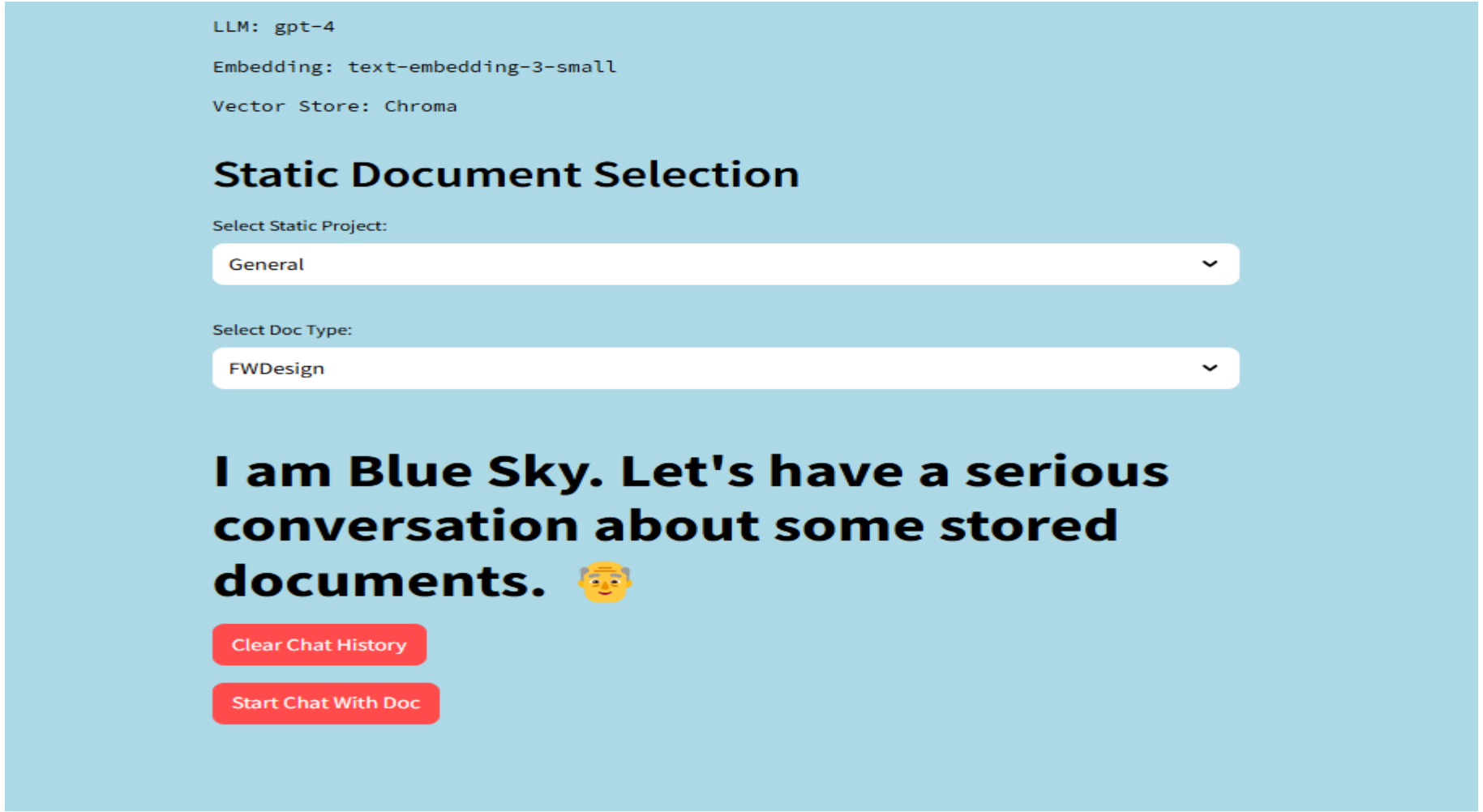


INCOSE Healthcare Working Group

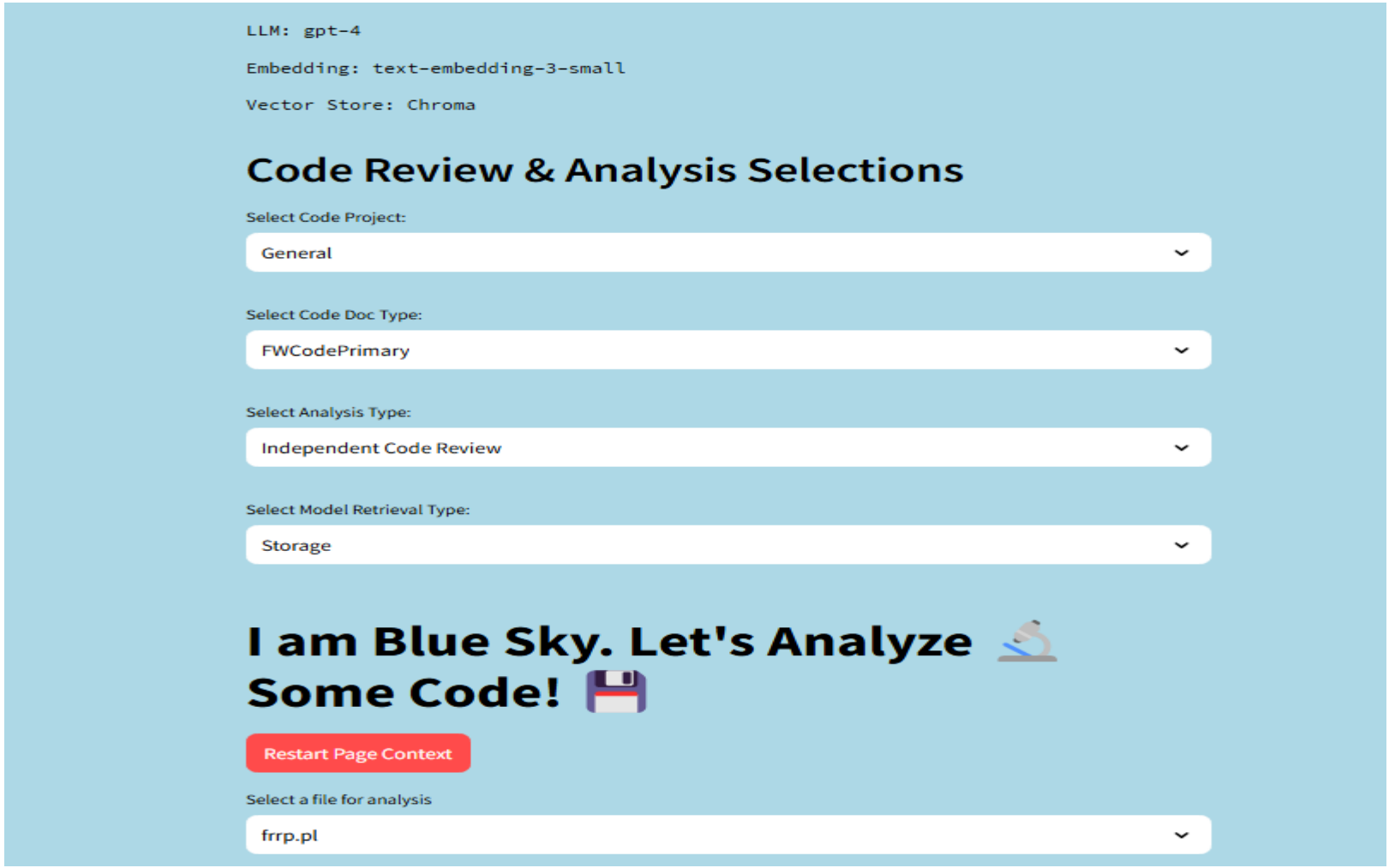
Dynamic Document Chat



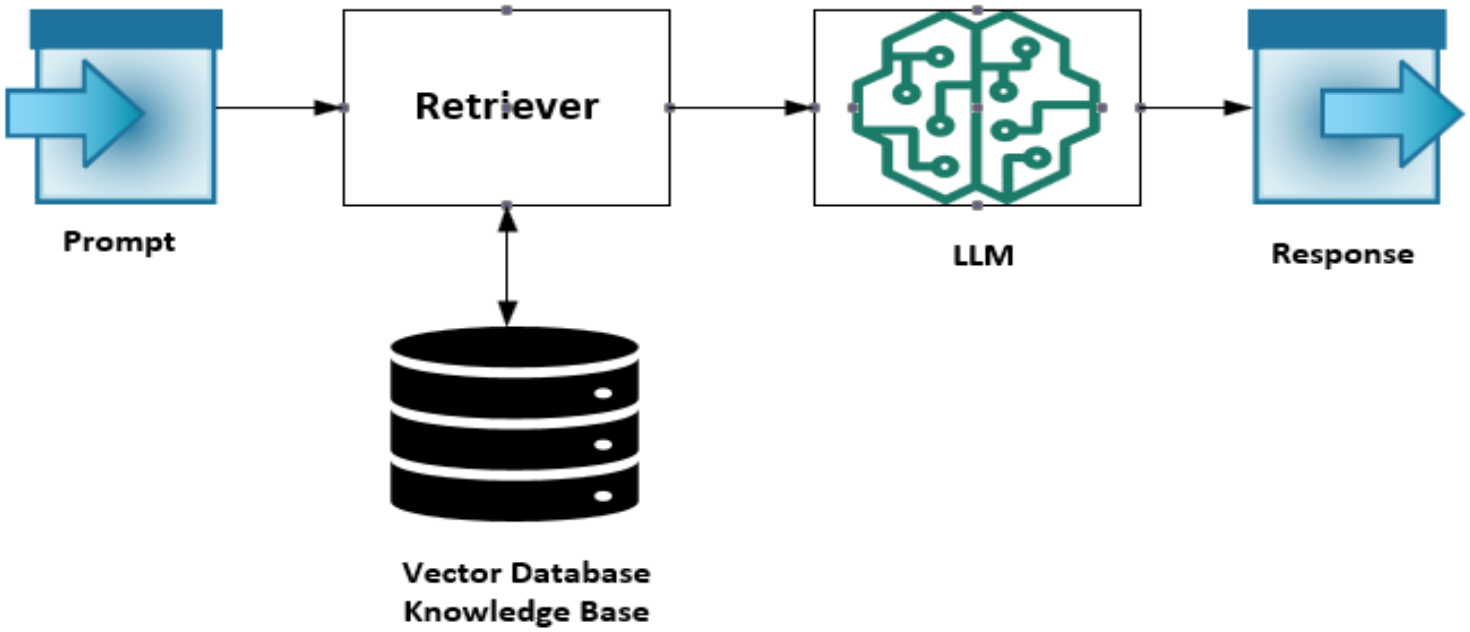
Repository Document Chat



Code Review And Analysis Chat

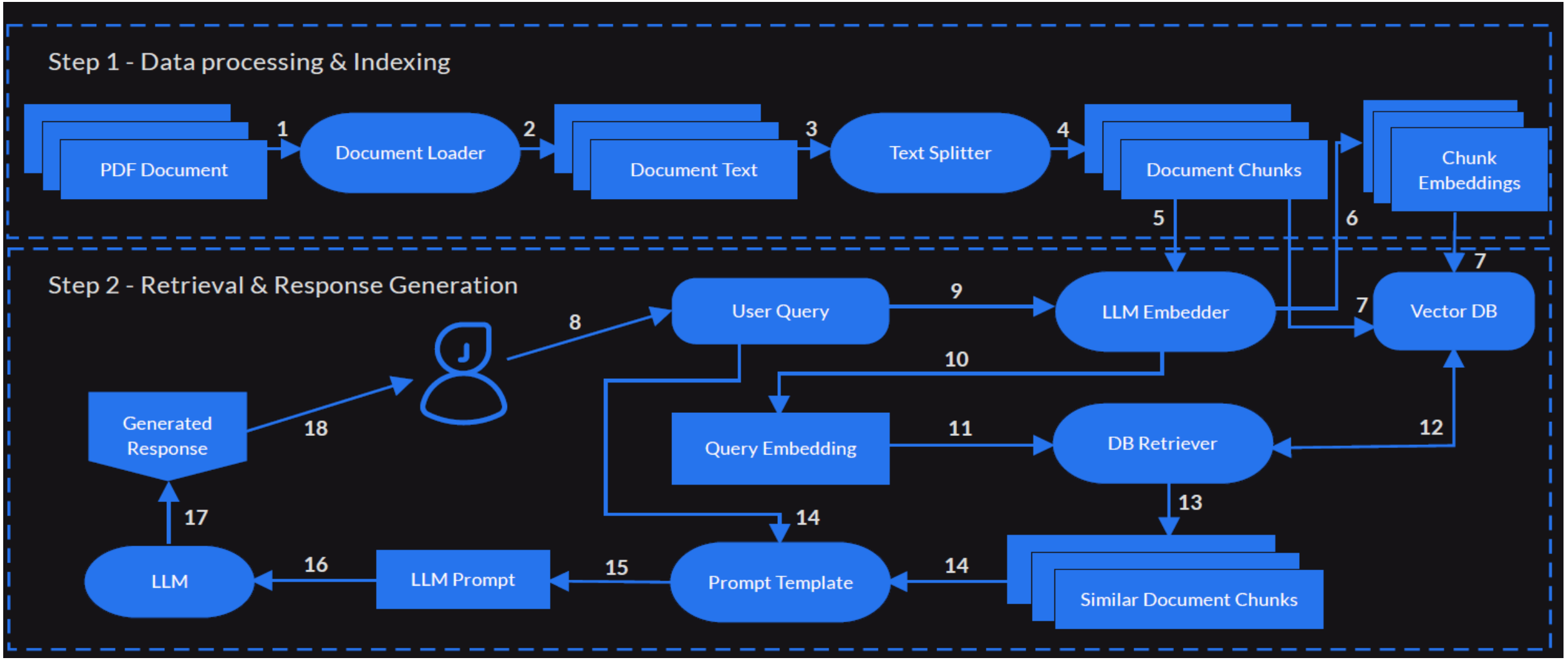


Retrieval Augmented Generation (RAG)

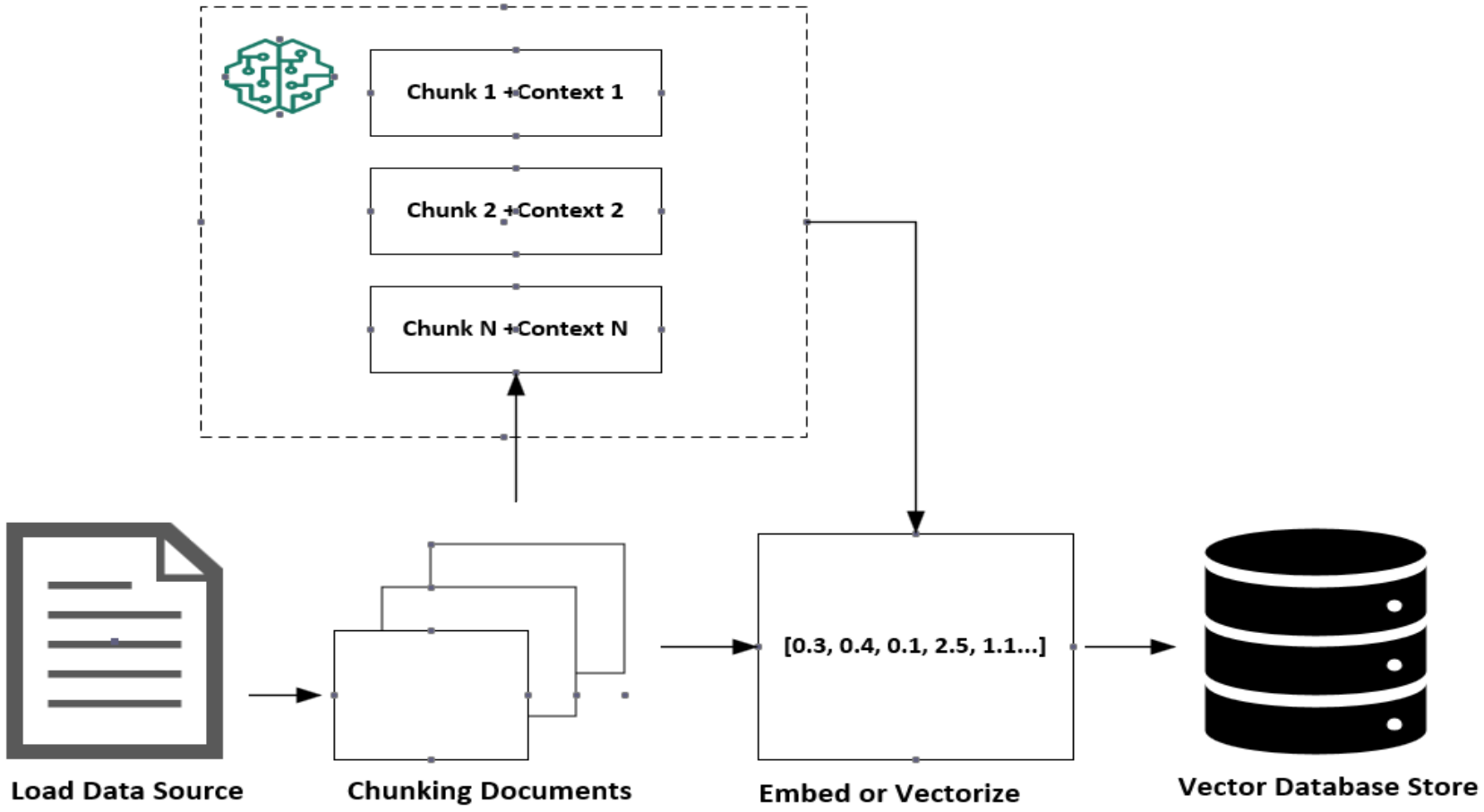


INCOSE Healthcare Working Group

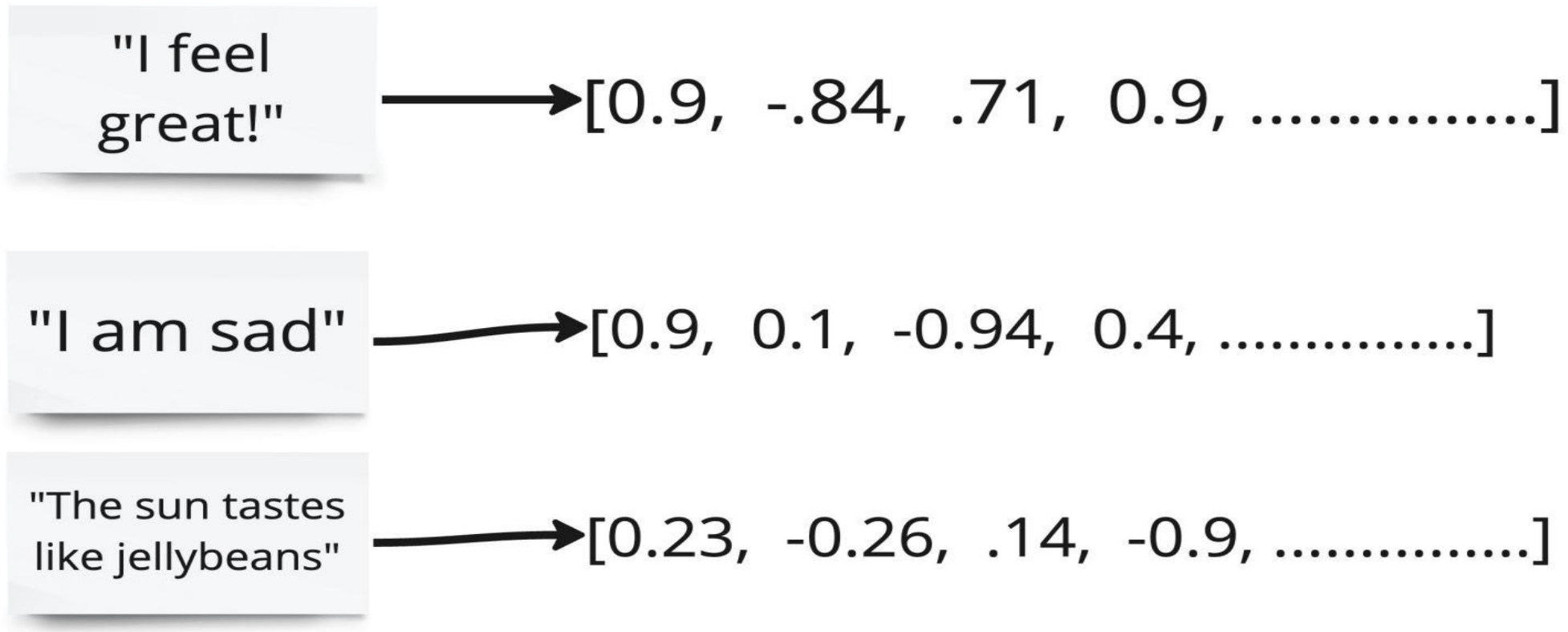
RAG Architecture



Contextual RAG and Storage



Embeddings


"I feel great!"
[0.9, -.84, .71, 0.9, ...]
"I am sad"
[0.9, 0.1, -0.94, 0.4, ...]
"The sun tastes like jellybeans"
[0.23, -0.26, .14, -0.9, ...]

INCOSE Healthcare Working Group

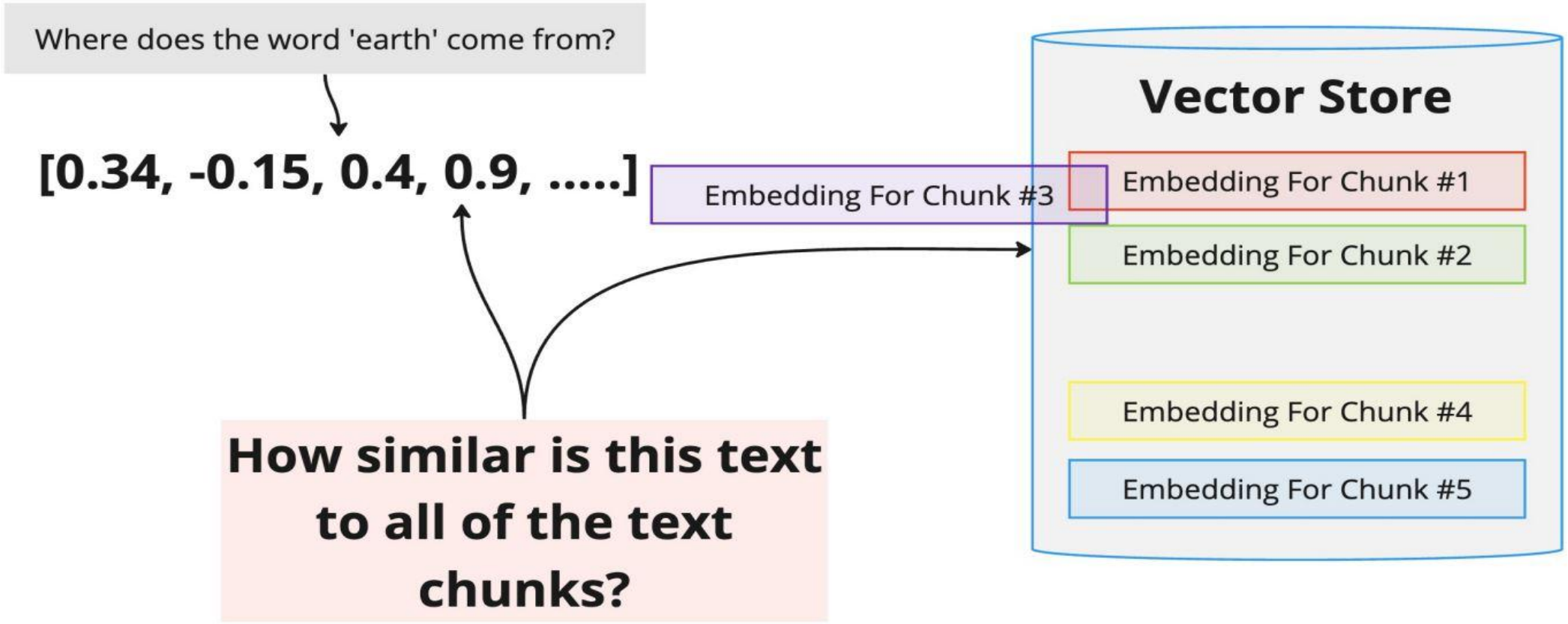
Embeddings...Continued

Sentence	Bravery Score (1 = bravery, -1 = fear)	Happiness Score (1 = happy, -1 = sad)	Embedding
The happy child jumped bravely from rock to rock	1	1	[1, 1]
The child was not timid and had a good time jumping from rock to rock	.5	.3	[.5, .3]
Although filled with great fear, the child jumped from rock to rock	-1	0	[-1, 0]

An embedding is a list of numbers between -1 and 1 that score how much a piece of text is talking about some particular quality

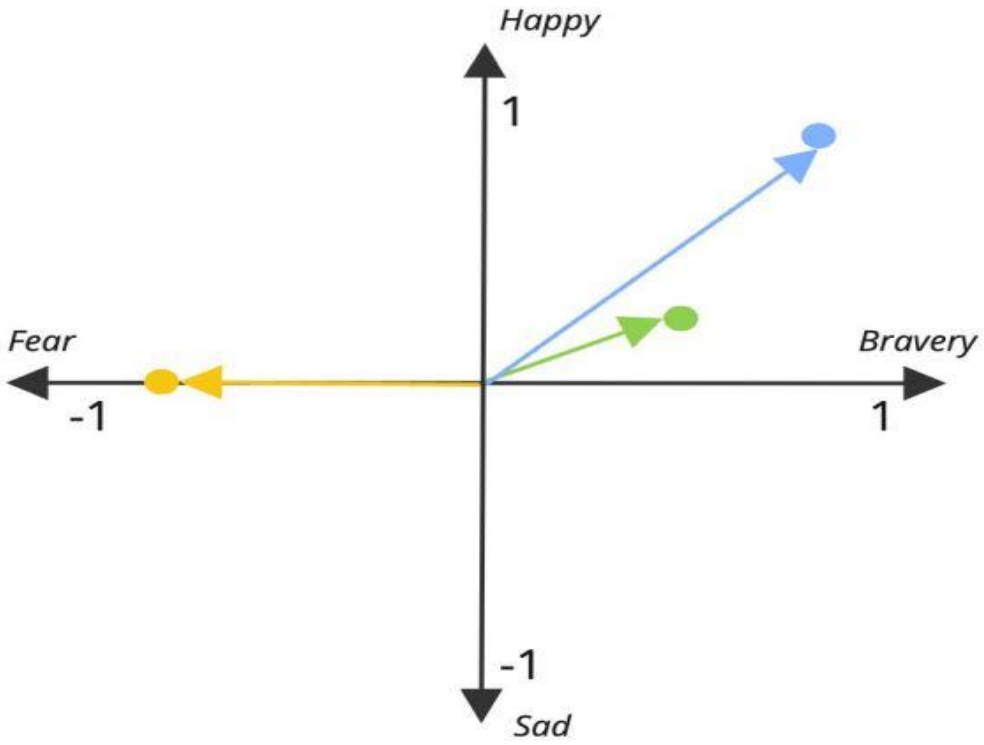
These embeddings only rate **2 qualities**, but real embeddings frequently have **700 to 1500 "dimensions"** or qualities that they score

Vector Stores and Similarity for RAG



Retrieval With Vectors

Sentence	Bravery Score (1 = bravery, -1 = fear)	Happiness Score (1 = happy, -1 = sad)
The happy child jumped bravely from rock to rock	1	1
The child was not timid and had a good time jumping from rock to rock	.5	.3
Although filled with great fear, the child jumped from rock to rock	-1	0



Squared L2 → Using the **distance between two points** to figure out how similar they are

Cosine similarity → Using the **angle between two vectors** to figure out how similar they are

What Is The Future Of RAG?

RAG Is Dead: Gemini 2.0 Flash Crushes Your Outdated AI Pipeline

Gurpreet Singh · Follow
3 min read · Feb 20, 2025

1 1

Reference:

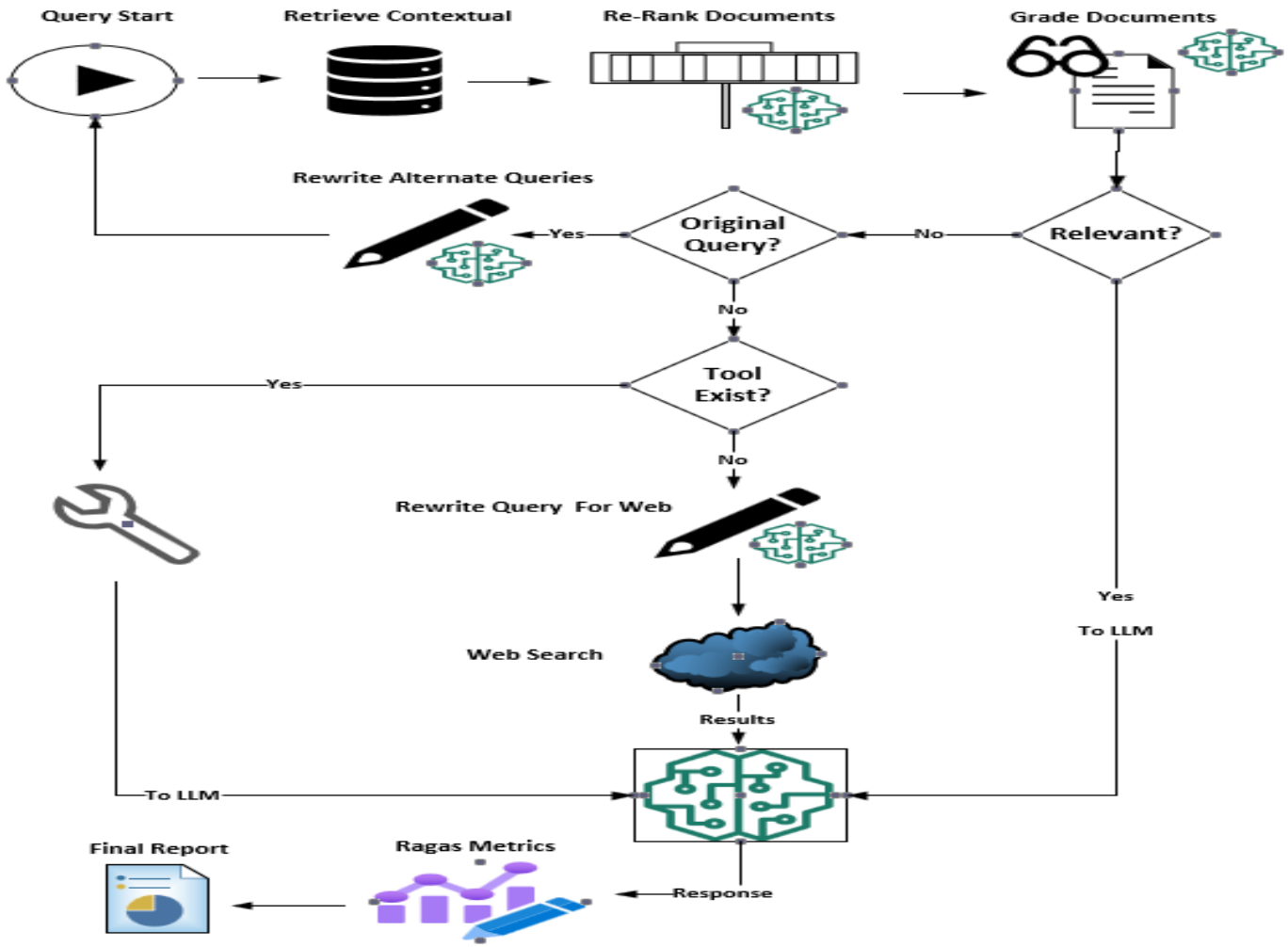
[RAG Is Dead: Gemini 2.0 Flash Crushes Your Outdated AI Pipeline | by Gurpreet Singh | Feb, 2025 | Medium]

What Is Blue Sky Working On?

- Agentic RAG using ReAct Workflow (https://arxiv.org/pdf/2210.03629) – March 2023
- Self-Reflective RAG (https://arxiv.org/abs/2310.11511) – Oct 2023
- Fusion RAG (https://arxiv.org/pdf/2402.03367) – February 2024
- Contextual RAG (https://www.anthropic.com/news/contextual-retrieval) – Sept 2024
- Adaptive RAG (https://arxiv.org/abs/2403.14403) – March 2024
- Multi-modal RAG (https://arxiv.org/abs/2410.11321) – Oct 2024
- Corrective RAG (https://arxiv.org/abs/2401.15884) – Oct 2024
- Self-Route RAG (https://arxiv.org/pdf/2407.16833) – Oct 2024
- RAG Metrics
  - Precision
  - Recall
  - Relevancy
  - Etc.

- Looking Beyond Firmware Engineering…To Systems…And More

“Self-RAG” Agentic Workflow Overview



INCOSE Healthcare Working Group

Conference Paper

Blue Skies Ahead

An AI Case Study on LLM Use for a Graph Theory Related Application

Gerald T. Rigdon

Fellow, Software Engineering

Boston Scientific, Inc.

gerald.rigdon@bsci.com

February 26, 2025

Abstract

Research activities necessitate the exploration of ideas which may involve significant software development, experimentation, and testing effort. The ability of Large Language Models (LLMs) to generate executable software has been demonstrated in various use cases. This study highlights the positive outcomes of a use case involving a graph theory application. Through a series of successful prompts, functional software was produced within minutes, a task that would otherwise have taken several hours or days to complete. Rather than focusing solely on cost savings from reduced engineering effort, this study aims to highlight the substantial opportunity benefits provided by this technology, especially since research typically involves evaluating multiple ideas before selecting the optimal solution.

INCOSE Healthcare Working Group

Cybersecurity Paper

Evaluating LLM Use In Identifying OSS Packages Linked to Cybersecurity Vulnerabilities for HAVOSS

Gerald Rigdon
Fellow, Software Engineering
Boston Scientific Corporation
St. Paul, USA
gerald.rigdon@bsci.com

***Abstract*—Building on the more general Capability Maturity Model (CMM) for software, the Handling Vulnerabilities In OSS (HAVOSS) model has emerged as an effective complement with a more detailed focus on handling Common Vulnerabilities and Exposures (CVEs) in third-party software packages used in embedded systems. As such, this HAVOSS model includes six capability areas and twenty-one practices. Of particular interest is the Identification and Monitoring of Sources area which presents significant challenges given the frequency of new vulnerability discoveries and disclosures. To address this HAVOSS area, this paper explores the use of Large Language Models (LLMs) within an agentic operational framework that employs multiple Retrieval Augmented Generation (RAG) methods to improve the accuracy of retrieved data based on a search input. Consequently, it explores, first, whether a CVE identification query is sufficient to retrieve the correct record with software package identification; and second, whether a software package information based query is sufficient to retrieve the correct record with the CVE identifier and its associated description. Hence, a security professional armed with a CVE identifier should be able to reliably extract the affected software packages and likewise armed with a list of software package identifiers be able to connect to the correct CVEs. Using a custom developed CyberVulModelEval software application, the research goal is to better understand the practical use of LLMs to accurately link vulnerability descriptive elements with deployed open source software packages.**

***Index Terms*—CVE, Agentic LLM, Self-RAG, HAVOSS**

I. Introduction

in a large repository of issues and concerns that have to be sifted through to enable cybersecurity professionals to identify vulnerabilities that exist in products under their supervision and respond accordingly.

Per Druckman [3] a 2024 Synopsis report showed that 96 percent of commercial code bases they sampled contained open source components. This corroborates the 2022 Linux Foundation study [4] that stated 70-90 percent of any given software code base contains open source components. Given the ubiquitous use of GitHub for managing open source code, entities such as the GitHub Advisory [5] was created to notify developers of vulnerabilities that are associated with affected software packages within the GitHub ecosystem. In terms of logistics, the workflow essentially consists of reported vulnerabilities that flow into the GitHub Advisory database from the NVD feed or from the GitHub internal npm (package manager) security team which are transformed into security advisories. As of 2022 [6], broader community participation and involvement has also been integrated into the process.

Given the importance of vulnerability disclosures and the linkage to software packages, especially open source, the security practitioner is confronted with the challenge of how to best consume the vast repository of data that is continually changing. To that end, the Handling Vulnerabilities In OSS (HAVOSS) model [7] includes the Identification and Monitoring of Sources as one of the key capability areas for evaluating

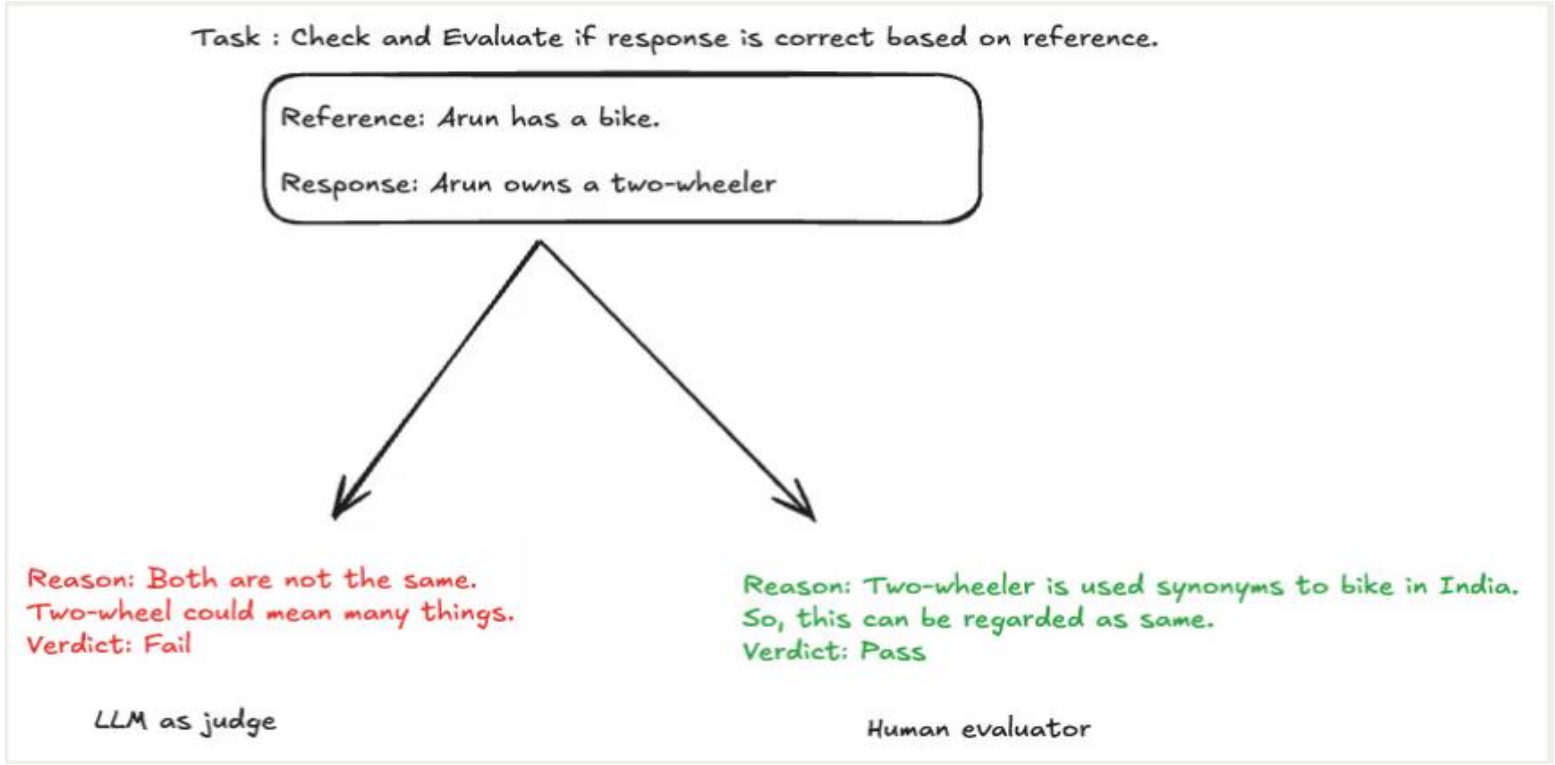
Upcoming Blue Sky Metrics Paper

Aligning LLM as judge with human evaluators

evals LLM

December 11, 2024

Shahul Es



INCOSE Healthcare Working Group

Invite Me Back Next Year?

INCOSE
Healthcare
Working Group

Questions

INCOSE
Healthcare
Working Group

Thank you for attending!

Share your experiences at #HWGSEC