# BUILDER-FRAMEWORK.AI

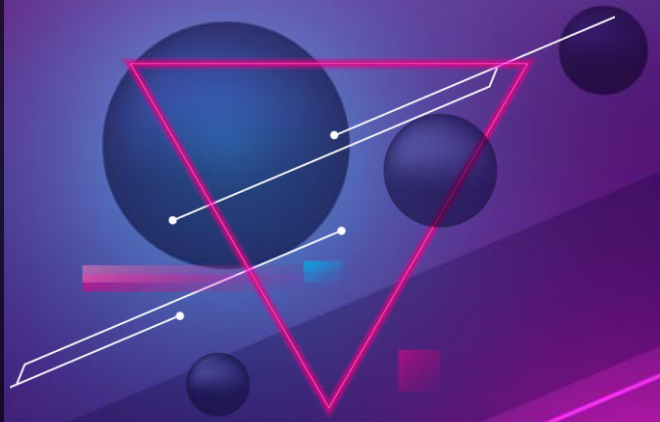
R-C Tech Nova

# PROBLEM STATEMENT

Our reusable microservice components (e.g., audit, workflow, reference data) are available only as inner-source modules in internal GitHub repositories. There is no seamless way to reference, combine, and generate multiple components as production-ready, containerized executables. This results in duplicated development effort, inconsistent implementations, and delays in delivering business solutions. Teams need a more efficient approach to leverage and assemble these assets for rapid, reliable deployment.

# PROPOSED SOLUTION

We propose a Builder Framework powered by Agentic Retrieval-Augmented Generation (RAG) using an LLM, capable of generating custom microservices (audit, attachment, workflow, reference data, etc.) based on user input and context retrieved from internal GitHub repositories/Custom Prompts.



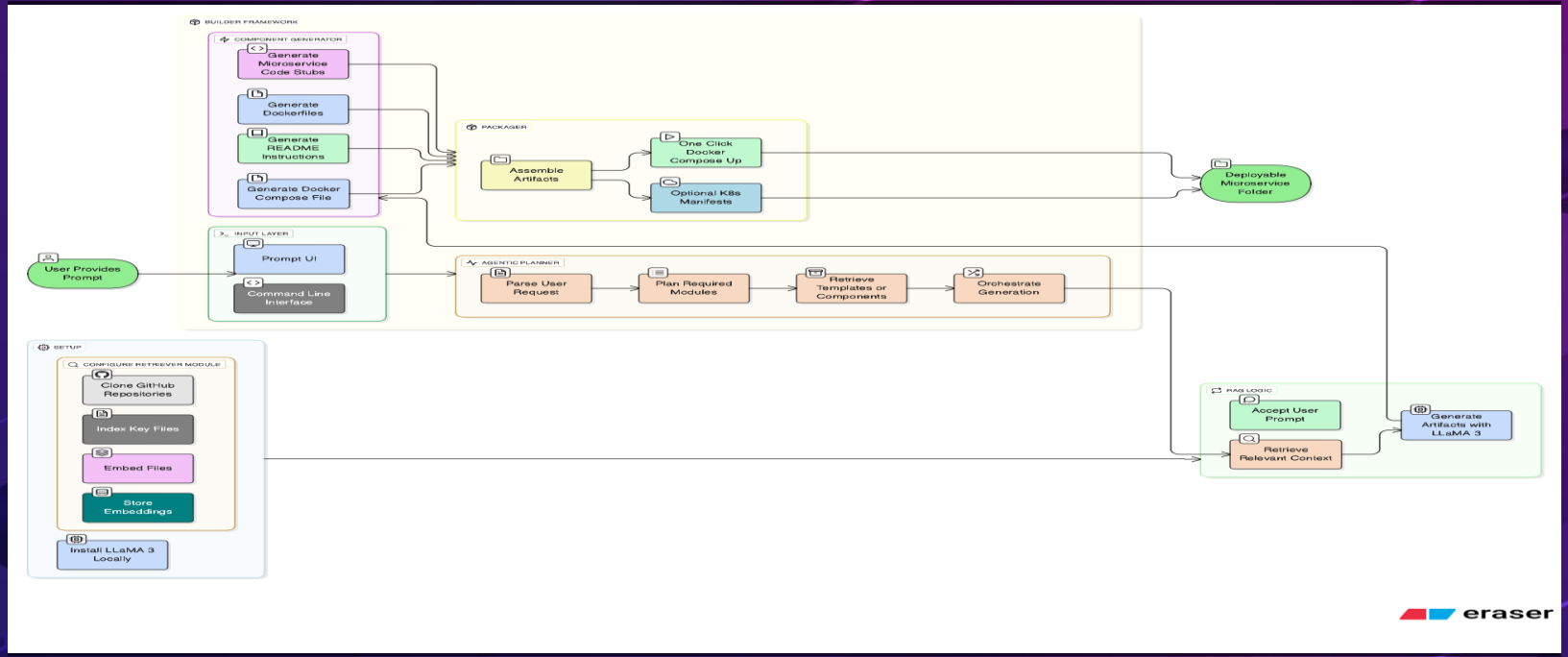
# KEY FEATURES

- Prompt-driven generation
- Hybrid Retrieval (prompts + templates)
- Generates: Entities, Controllers, Services, Repos

# TECH STACK

- Spring Boot 3
- LangChain4j
- LLaMA3
- In-memory FAISS
- Docker + Swagger/OpenAPI

# FLOW CHART


BUILDER FRAMEWORK
COMPONENT GENERATOR
Generate Microservice Code Stubs
Generate Dockerfiles
Generate README Instructions
Generate Docker Compose File
PACKAGER
Assemble Artifacts
One Click Docker Compose Up
Optional K8s Manifests
Deployable Microservice Folder
INPUT LAYER
Prompt UI
Command Line Interface
User Provides Prompt
AGENTIC PLANNER
Parse User Request
Plan Required Modules
Retrieve Templates or Components
Orchestrate Generation
SETUP
CONFIGURE RETRIEVER MODULE
Clone GitHub Repositories
Index Key Files
Embed Files
Store Embeddings
Install LLaMA 3 Locally
RAG LOGIC
Accept User Prompt
Retrieve Relevant Context
Generate Artifacts with LLaMA 3
eraser

# DEMO FLOW

- Step 1: User prompt → *“Create audit microservice”*
- Step 2: Router picks AuditRAG
- Step 3: Retriever fetches AuditPrompt + AuditTemplate
- Step 4: LLaMA generates Spring Boot code
- Step 5: Packager outputs runnable service with Docker

# THANKS!

CREDITS: This presentation template was created by **Slidesgo,** and includes icons by **Flaticon**, and infographics & images by **Freepik**