

Project Progress Presentation

Shmups

Ashita , Rivva , Radha

February 19, 2026

Introduction

- Software teams continuously make architectural and design decisions.
- These decisions are scattered across commits, pull requests, and issues.
- Over time, reasoning behind changes is lost.
- This project converts GitHub activity into structured engineering knowledge.

Goal: Build a system that captures, analyzes, and enables querying of engineering decisions.

Scope of the Project

- Capture GitHub activity data
- Standardize event formats
- Extract engineering decisions using AI
- Build a decision knowledge graph
- Enable natural language "why" queries
- Provide explainable outputs with citations

Focus: GitHub-based engineering workflows and decision intelligence.

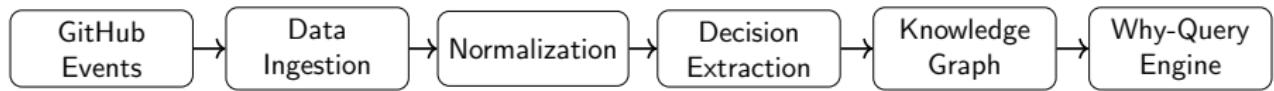
Why This Project?

- Decisions hidden in commit messages and discussions
- Lack of structured engineering memory
- Difficult onboarding for new contributors
- No system to query historical reasoning

Need:

- Persistent decision tracking
- Context-aware development insights
- Explainable engineering intelligence

System Overview



Architecture Design

- Data ingestion layer
- Event normalization
- Decision extraction engine
- Storage and knowledge graph
- Intelligence layer
- CLI / Web / API interfaces

Design Principle: Modular, scalable, and pipeline-driven.

Technology Stack

Backend

- Node.js (ES6+ modules)
- Express.js framework
- REST API architecture

Storage Architecture

- SQLite database
- Lightweight and efficient local storage
- Structured event and decision data persistence

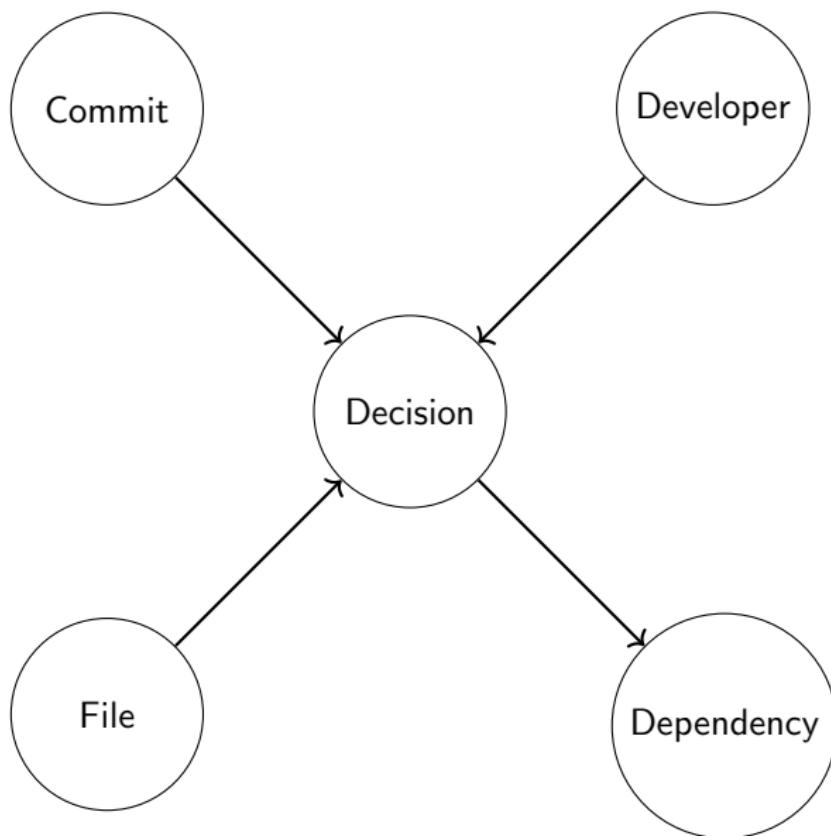
Frontend

- Vanilla JavaScript
- Canvas API for graph visualization
- CSS3 for styling and layout

Key Capability

- Interactive visualization of decisions as connected nodes
- Real-time querying and rendering of relationships

Knowledge Graph Representation



Key Technical Components

Data Layer

- Immutable GitHub event capture
- Unified event schema

AI Layer

- LLM-powered decision extraction
- Reasoning and trade-off identification

Analysis Layer

- AST and dependency analysis
- Timeline modeling

Progress So Far

Completed

- Architecture planning
- Modular project structure
- Data ingestion setup
- Normalization framework
- CLI and API scaffolding

Ongoing

- Decision extraction improvements
- Knowledge graph construction
- Freshness modeling

Challenges and Future Implementation

Challenges

- Ambiguous commit reasoning
- Extracting structured insights from text
- Handling conflicting decisions
- Designing explainable AI outputs

Future Work

- Improve extraction accuracy
- Complete knowledge graph integration
- Add visualization dashboard
- Optimize scalability and performance