LTIMindtree

Let’s get to the future, faster. **Together.**

Digital and App Innovation Services

App Modernization Offerings

Advisory

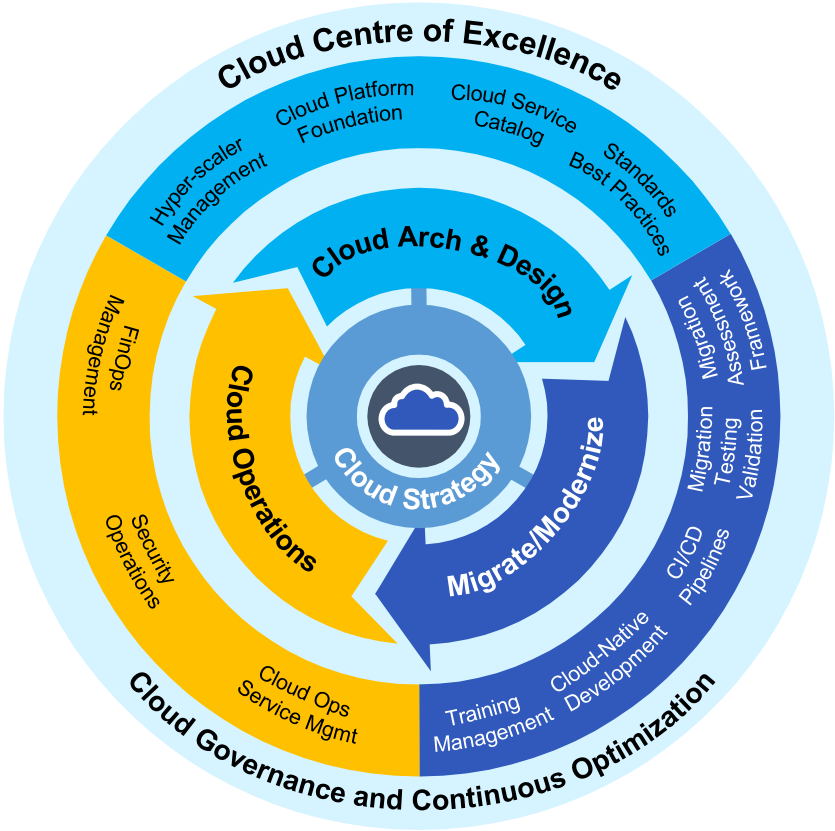
- Portfolio Analysis
- Modernization advisory & assessment
- MRAP Framework
- Cloud fitment analysis
- Business Case & TCO Analysis
- Architecture Strategy & Roadmap/Validation

Transformation

- App Migration and Modernization (Re-Factor, Re-Host, Re-Platform & Re-Architect)
- Batch offloading
- Data Modernization
- Data Migration

Enablement

- AI-Infusion
- Generative AI enablement
- API eco-system
- UI modernization
- Data Archival and Legacy decommission



Building the Knowledge Fabric at different levels

CAST Imaging

Architectural Insights

Source Code Files | Config Files | Database Scripts

Deep Analysis via "MRI for Software"

- Architecture Blueprints
- Dependencies, Call Graphs
- Health, Vulnerabilities
- End to end Transactions

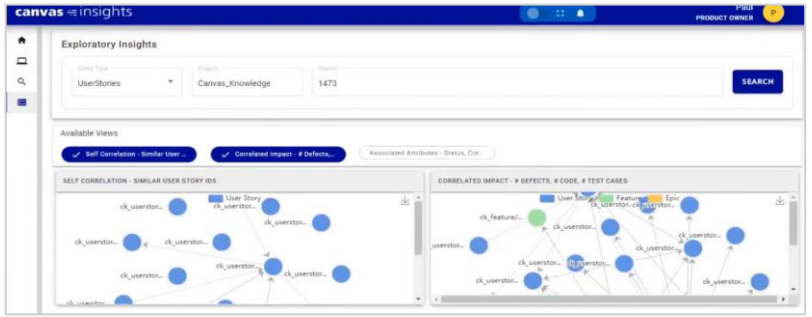
canvas insights

Traceability | Correlation | Predictive Insights

User Stories | Code & Build (Logs, Traces) | Test Artifacts

ML/NLP Algorithms to create end to end Correlation

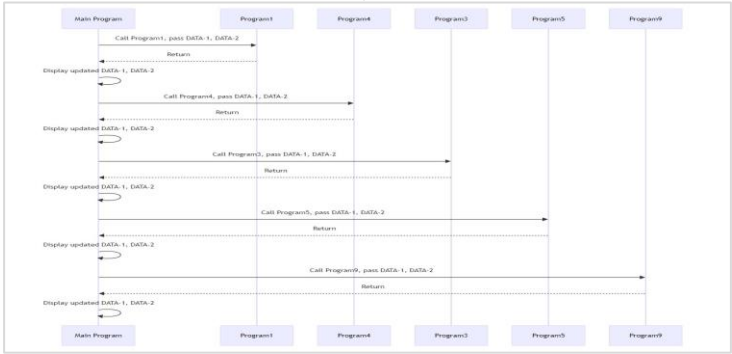
- Defects Advisor
- Code Analytics, Hotspots
- Test Coverage Insights
- Correlated Change Impact



LTIMGenie

Reverse Engineering

Reverse Engineering | Code Summary (includes Mermaid Script Generation for sequence flow diagram) | Inline Code Documentation



canvas appScribe

Agile User Stories Generation

Agile user stories generation, using INVEST Principles | Acceptance criteria generation, from Product owner perspective

Approach to crafting use cases in LTIMGenie

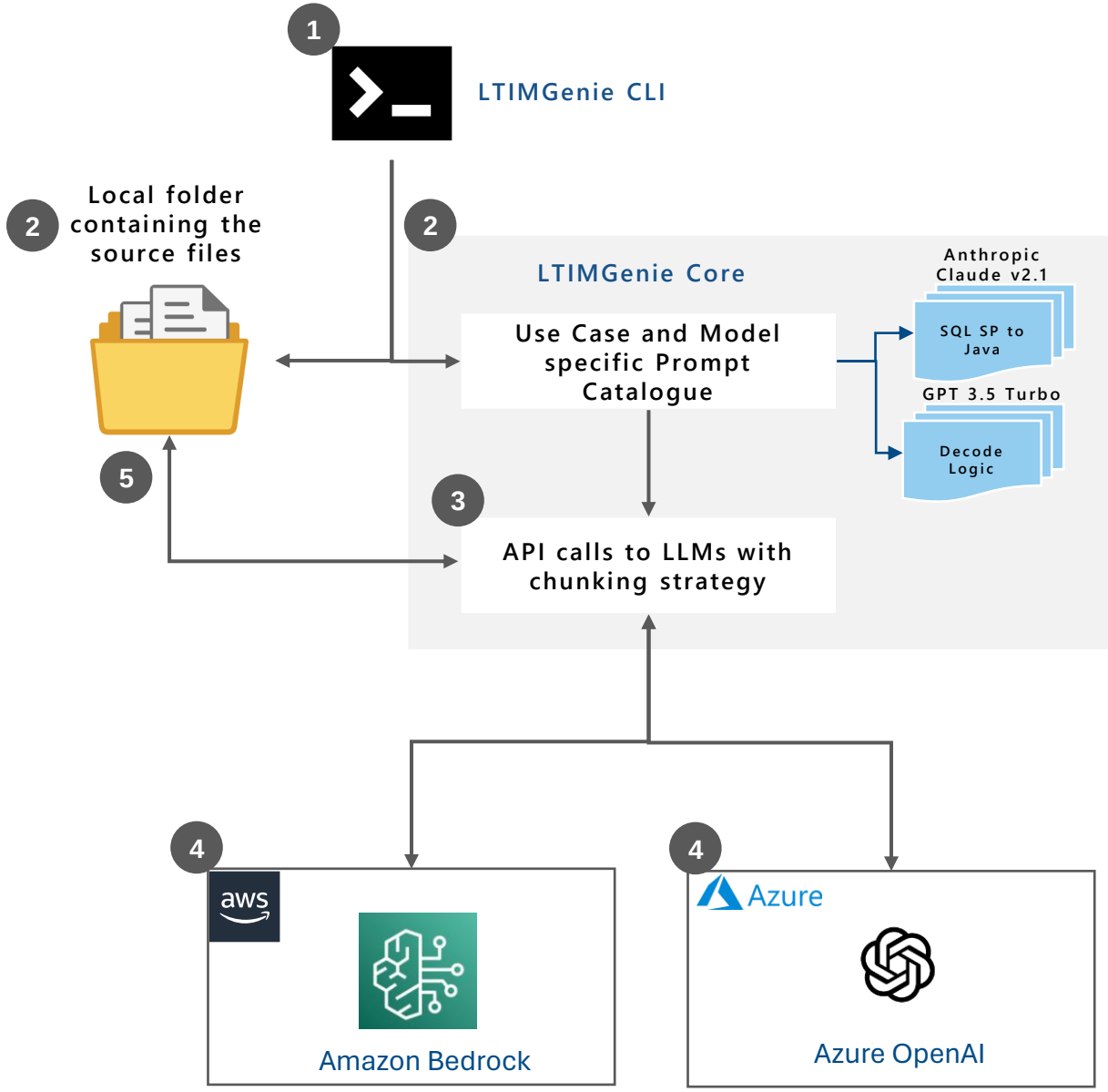
1. User invokes LTIMGenie CLI utility to select a persona, the appropriate use case, specify target large language model (LLM), provide input artifact specifics and the path for storing output.

2. Based on the chosen use case and model, LTIMGenie picks the appropriate prompt from the in-built prompt catalogue and maneuvers through the local file system to read the relevant input file

3. The prompt along with the input file contents are measured in terms of the total number of tokens consumed and appropriately chunked for aligning to the model-specific token limitations. Subsequently the appropriate LLM-specific API call is made.

4. The LLM gets the context along with the parameters to process the use case and provides the response back

5. Output from the LLM is uploaded in the specified target location.



Developer Productivity Enhancement

with GitHub Copilot

GitHub Copilot

An AI Pair Programmer that understands the context and natural language prompts made by developers to generates code suggestions

GitHub Copilot X (Beta)

A vision to AI-powered software development. Features are powered by GPT-4 model.

GitHub Copilot Chat (Beta)

Conversational AI experience to generate code, learn technology and unit test case generation

and many more as features are evolving…

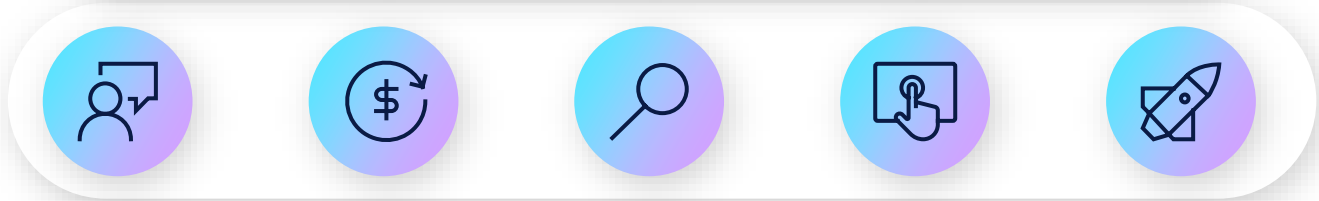
Microsoft GitHub Copilot

Designed to help developers spend less time on boilerplate and repetitive code, and more time on what really matters ! *

* Microsoft Research

55%	46%	75%
Faster Code Development	Higher Rate of Code Completion	More Developer Empowerment

Our Competency in GitHub Copilot



Code Explanation	Code Generation	Code Refactoring	Unit Test Case Generation	Recommendations on Best Practices & Standards

LTIMindtree's Enablement Program

GitHub Developers Day | Curated Learning | Certifications

Success Snippets:

- Automated bulk generation of unit test cases using GitHub Copilot
- One of the big 4 accounting firms – Start of engagement on generation of unit test cases using GitHub Copilot

Thank You