

Jev for Software Developers

Separating bounded decisions from generation

FA-0816 · 1 day

Agentic & Generative AI · Software Development

Integrate Jev decision primitives into software and build a guided Jev-plus-LLM workflow with routing, evaluation and human escalation.

Introduction

Jev is TypeSafe AI's System One model for bounded decisions that application code can consume directly. Software supplies state and typed questions; Choice selects an option, Score evaluates a rubric, and Noul evaluates a statement on a 0–1 scale. Choice and Score return probability distributions and confidence information. These outputs support application decisions, but still require task-specific evaluation and controls.

For developers, the useful question is not simply how to add another model. It is which component should own a decision. Conventional code handles deterministic rules, Jev can support bounded judgement, generative models handle tasks such as explanation and content generation, and people review high-risk or unresolved cases. The course explores classification, routing, prioritisation and deciding when an LLM is worth invoking.

This intensive one-day course is for experienced software developers who already work with LLM APIs and agents. Participants use a prepared environment and one guided workflow to connect the decision layer to application code, compare routing choices, and test thresholds and fallbacks. The broader architecture and production topics are focused discussions and demonstrations, not a promise to deliver a production-ready system in a day.

Learning outcomes

By the end of the course, participants will be able to:

- Explain the purpose of Jev and TypeSafe AI's System One model architecture.
- Distinguish decision models from generative LLMs and conventional deterministic code.
- Identify software-development problems that are appropriate and inappropriate for Jev.

- Use Jev's Choice, Score, and Noul decision primitives.
- Integrate Jev into an application using its API or SDK.
- Design bounded decision spaces with application-side validation and permission controls.
- Work with probabilities, confidence, thresholds, fallbacks, and escalation.
- Use Jev for classification, routing, gating, prioritisation, and tool selection.
- Design Jev-driven LLM routing that determines when an LLM should be invoked.
- Combine Jev with generative LLMs in practical agentic software architectures.
- Decide when conventional code, Jev, an LLM, or a human should own a decision.
- Evaluate decision quality and failure behaviour before considering production use.

Prerequisites

Participants should have:

- Software-development experience.
- Ability to read and modify Python or JavaScript/TypeScript code.
- Working experience consuming HTTP/REST APIs.
- Deep understanding of LLMs and API-based AI applications.
- Prior Agent building experience.

Access to an approved TypeSafe API account and a prepared Python or JavaScript/TypeScript development environment. Any LLM integration requires its own approved access; API usage and availability depend on the providers and account settings.

Training outline

1. 1. Understanding Jev and System One Models

1. The problem Jev is designed to solve
2. System One versus generative AI
3. Decisions rather than generated strings
4. Jev within conventional software architecture
5. Current Jev ecosystem and positioning

2. 2. Understanding the Jev Decision Model

1. State and typed questions
2. Choice decisions

3. Score decisions
4. Noul: evaluating a statement on a 0–1 scale
5. Multiple questions against shared state
6. Probability distributions and confidence for Choice and Score; interpreting Noul separately
7. Structured machine-consumable responses

3. 3. Where Jev Fits in Software Development

1. Classification
2. Routing
3. Scoring
4. Prioritisation
5. Model-assisted policy triage versus deterministic policy enforcement
6. Workflow selection
7. Human-review decisions
8. Agent and tool selection

4. 4. Where Jev Does Not Fit

1. Free-form generation
2. Software code generation
3. Long-form reasoning
4. Conversation
5. Summarisation and explanation
6. Exact deterministic operations
7. Conventional business rules

5. 5. Developing with Jev

1. Jev API architecture
2. Authentication and API access
3. Request and response structure
4. Working with application state
5. Defining typed questions
6. Processing Jev responses in code
7. Error handling and fallbacks
8. Python and JavaScript/TypeScript SDK considerations

6. 6. Designing Reliable Decisions

1. Defining bounded answer spaces
2. Writing focused decision questions
3. Separating judgement from application logic
4. Confidence thresholds and their limits
5. Uncertain decisions
6. Safe fallbacks
7. Human escalation
8. Testing decision behaviour

7. 7. Jev as an Application Decision Layer

1. Request classification
2. Workflow routing
3. Feature and action gating with deterministic permission enforcement
4. Queue prioritisation
5. Risk and review decisions
6. Decision-driven application control flow

8. 8. Jev and Large Language Models

1. Complementary System One and generative architectures
2. Determining when generation is actually required
3. Jev before the LLM
4. Jev after the LLM
5. Jev around agent workflows
6. Separating judgement from generation

9. 9. Intelligent LLM Invocation

1. Identifying requests that do not require an LLM
2. Complexity classification
3. Estimating task complexity and reasoning needs, then testing routing quality
4. Model-tier selection
5. Fast versus reasoning-model routing
6. Escalation to frontier models

7. LLM bypass paths
8. Cost and latency-aware routing

10. 10. Jev in Agentic Software Systems

1. Agent routing
2. Tool selection
3. MCP tool routing
4. Action approval gates enforced by application policy and human authority
5. Human-in-the-loop decisions
6. Guardrails around consequential actions
7. Context relevance decisions
8. Agent escalation paths

11. 11. Designing Hybrid AI Architectures

1. Conventional code for deterministic behaviour
2. Jev for bounded judgement
3. LLMs for generation and deeper reasoning
4. Humans for high-risk or unresolved decisions
5. Decision orchestration between components
6. Building maintainable hybrid pipelines

12. 12. Production Considerations

1. Decision thresholds and calibration
2. Evaluation datasets
3. False-positive and false-negative trade-offs
4. Observability and decision logging
5. Failure and fallback strategies
6. Security boundaries
7. Cost and latency considerations
8. Gradual production rollout

13. 13. Building a Practical Jev-LLM Workflow

1. Incoming application state

2. Jev decision layer
3. LLM invocation decision
4. Model and workflow routing
5. Application action
6. Human escalation
7. Feedback and evaluation loop

Disclaimer

Topics and exercise depth may be adjusted to participant experience, available time and product changes while retaining the course's core learning objectives.

© 2026 Faysal Aziz · Kuala Lumpur

<https://faysalaziz.com/courses/jev-for-software-developers/>