SPRING IO 2026 | April 14th-15th 2026

SPRING IO 2026

OpenAPI and Spring-Boot 4 - What's New ?

Badr NASS LAHSEN

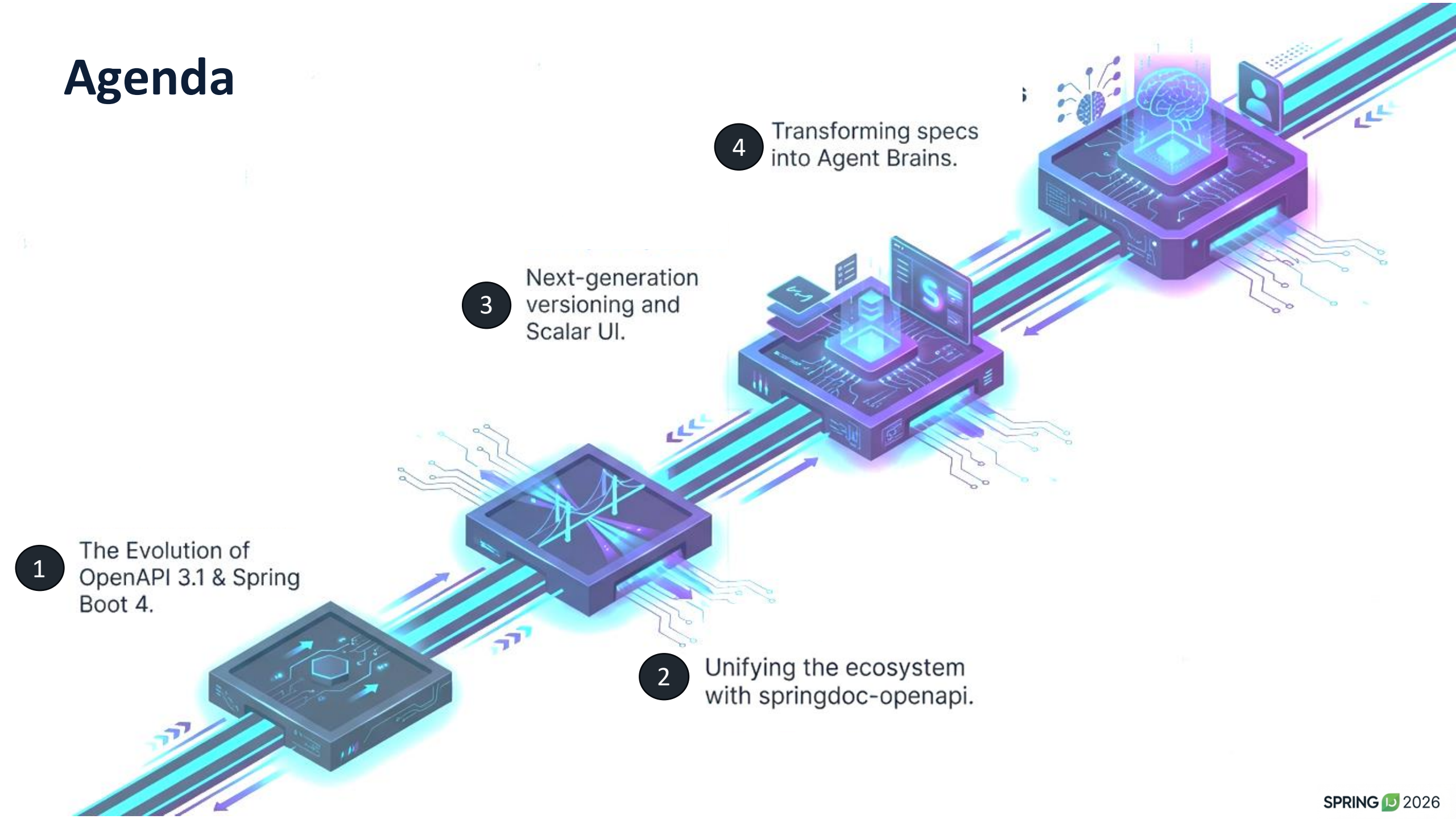
Manager Solutions Engineering
EMEA - Palo Alto Networks

@bnasslahsen

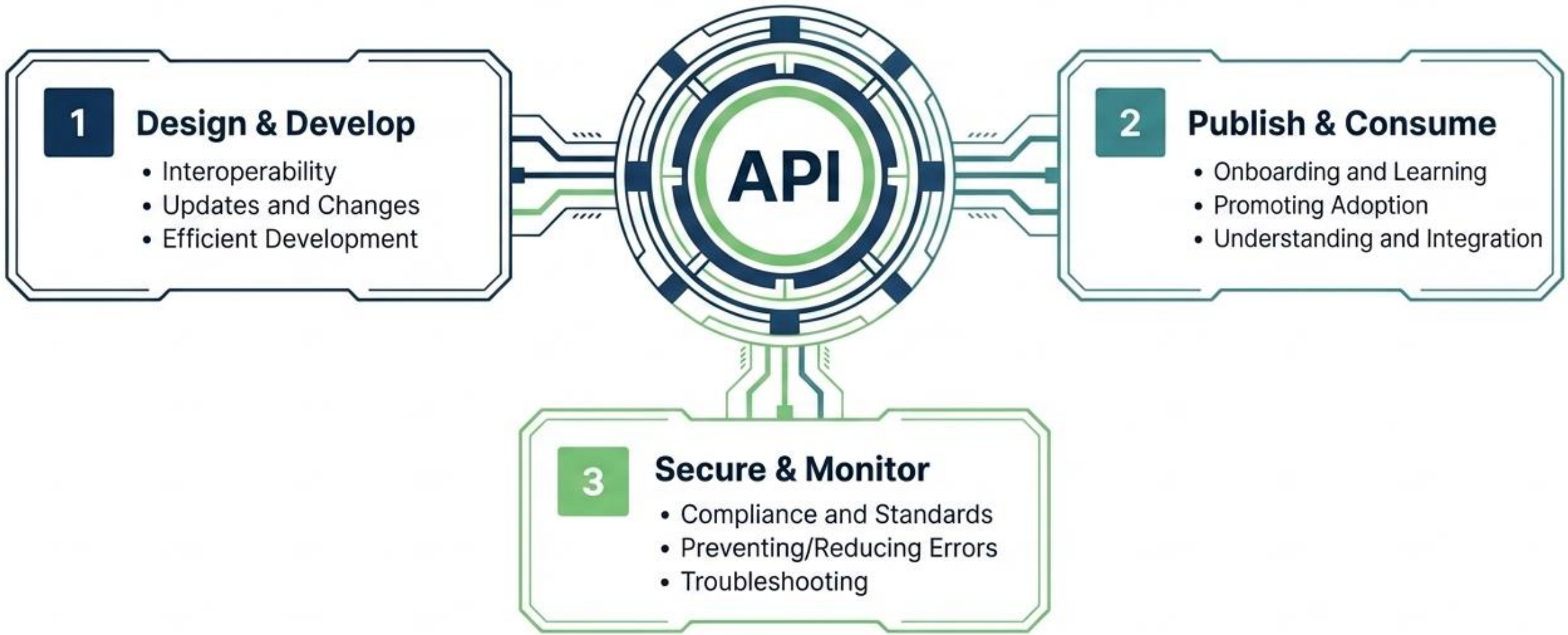
@nass_lahsen

Agenda

1. The Evolution of OpenAPI 3.1 & Spring Boot 4.
2. Unifying the ecosystem with springdoc-openapi.
3. Next-generation versioning and Scalar UI.
4. Transforming specs into Agent Brains.

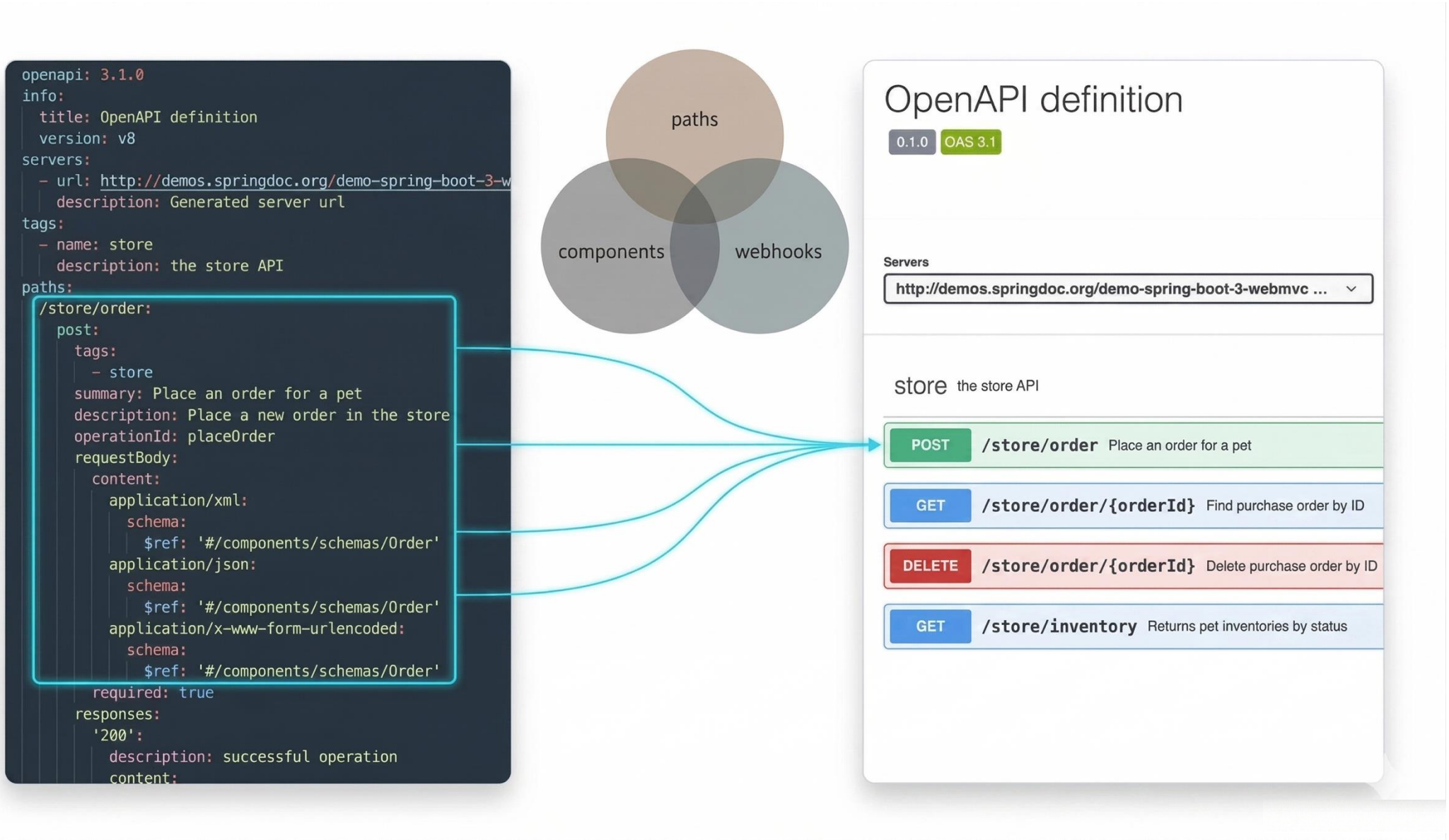


The Role of OpenAPI in Modern Architecture

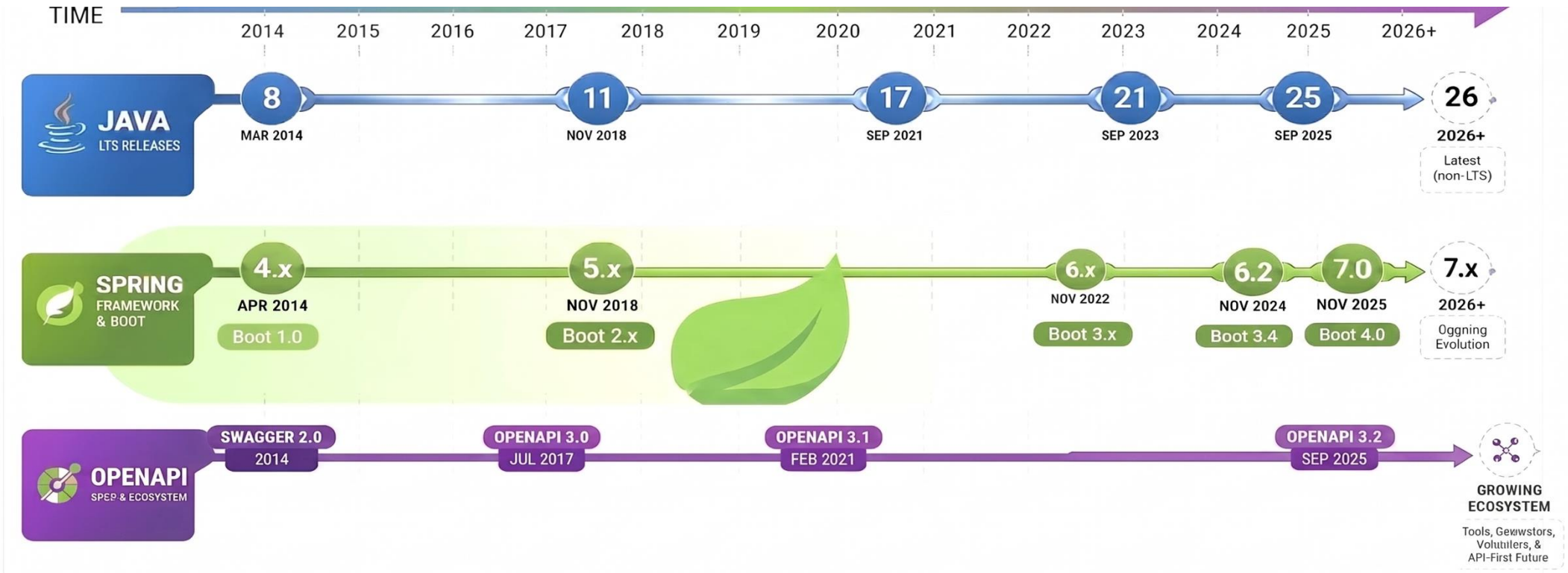


OpenAPI is no longer just documentation; it is the machine-readable **contract** driving the entire API lifecycle.

OpenAPI: The Blueprint of Modern Connectivity



Unified History, One Ecosystem, Continuous Evolution



1. JAVA LTS
STABILITY • PERFORMANCE • INNOVATION
Every 3 years, a new LTS sets the foundation for the entire ecosystem.

2. SPRING
ALIGNS • ADAPTS • ACCELEBATES
Spring evolves with Java, delivering a powerful, productive, and enterprise-ready platform.

3. OPENAPI
STANDARDIZES • CONNECTS • EMPOWERS
OpenAPI enables API-first design, collaboration, and innovation across all platforms.

SAME JOURNEY. STRONGER TOGETHER. BUILDING THE FUTURE.

OpenAPI 3.1: The Paradigm Shift

Allowed request body for all HTTP methods

Added multipart/form-data support for encoding object

Path Item parameters must be defined

Removed definition of some formats e.g. byte, binary

Responses are now optional

OpenAPI 3.0.3

type	format	Comments
integer	int32	signed 32 bits
integer	int64	signed 64 bits (a.k.a long)
number	float	
number	double	
string		
string	byte	base64 encoded characters
string	binary	any sequence of octets
boolean		
string	date	As defined by full-date - RFC3339
string	date time	As defined by date time - RFC3339
string	password	A hint to UIs to obscure input.

OpenAPI 3.1

type	format	Comments
integer	int32	signed 32 bits
integer	int64	signed 64 bits (a.k.a long)
number	float	
number	double	
string	password	A hint to UIs to obscure input.

Because of JSON schema not all format extensions are specified anymore

The Arrival of Spring Boot 4

Release & Reach

GA on November 20th, 2025. The most widely used Java framework globally.

Performance

Enhanced startup speeds and richer cloud-native capabilities.

Safety & Resilience

Null Safety implemented via Jspecify. Built-in system resilience.

API Native

Built-in API versioning to streamline lifecycle management.

* Spring Boot 3.5: End of Free Support (June 2026)

The **SpringDoc** Ecosystem by the **Numbers**

300M+
Total Events/Downloads
(Q1 2026 alone)

187K+
Dependent Repositories

3,700+
GitHub Stars

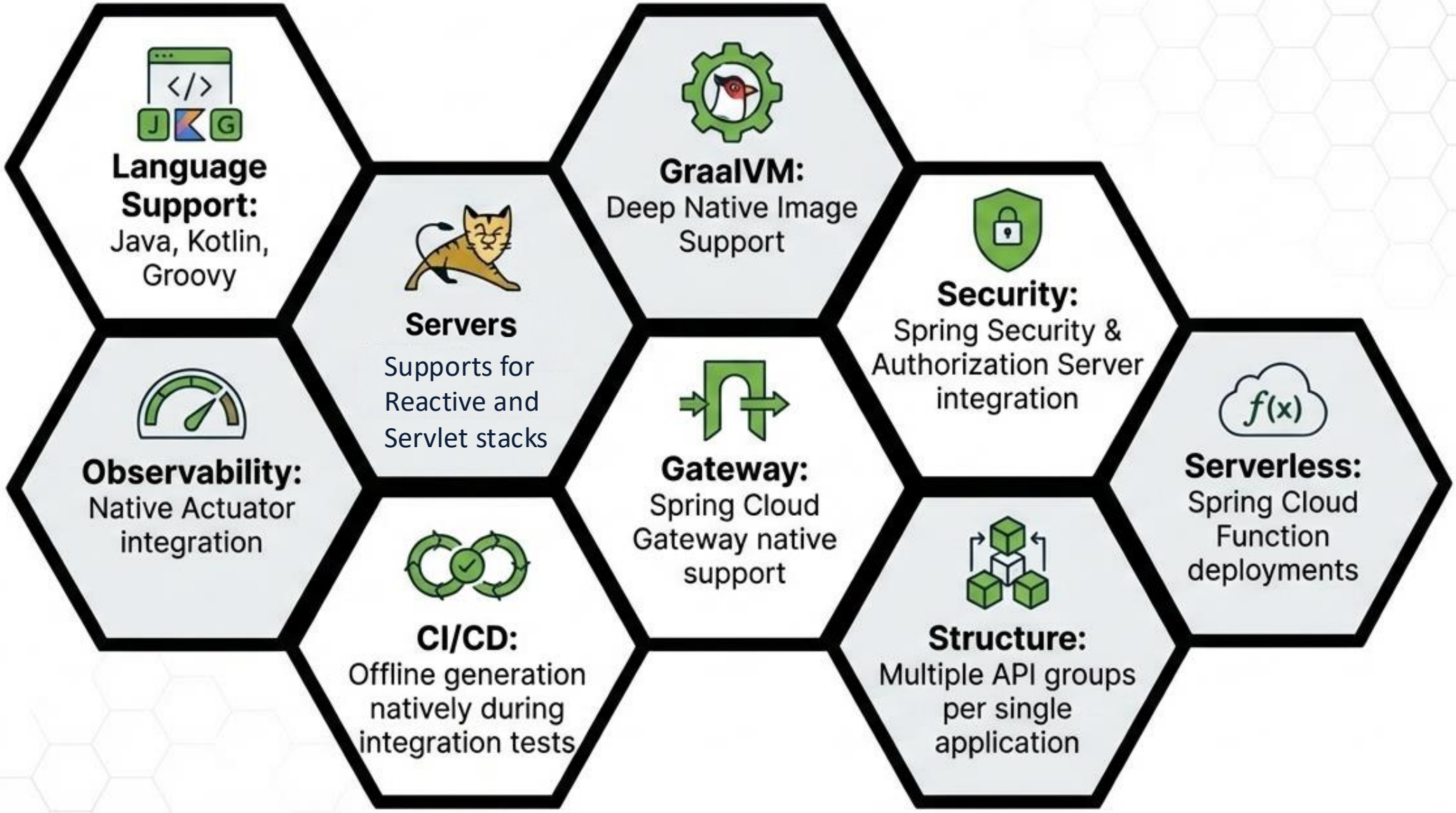
Community Drive

+ 105 contributors

Enterprise Adopters

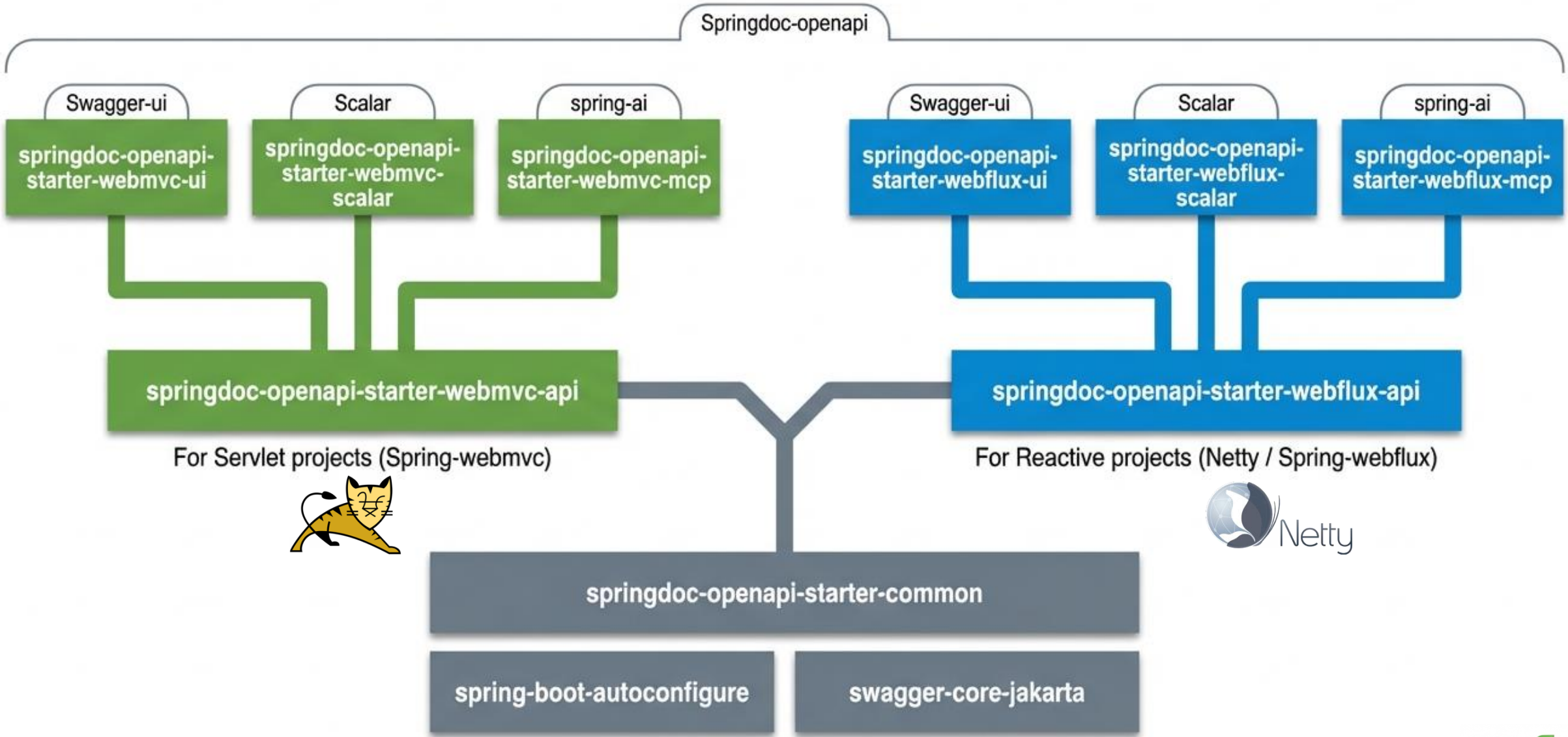
Red Hat
NETFLIX
ORACLE
AMERICAN EXPRESS
GOV.UK

Core Ecosystem Integrations



SpringDoc OpenAPI Modules Dependency Architecture

A visual reference for Spring / Java ecosystems



API Documentation Modernization with Scalar

Developer-First

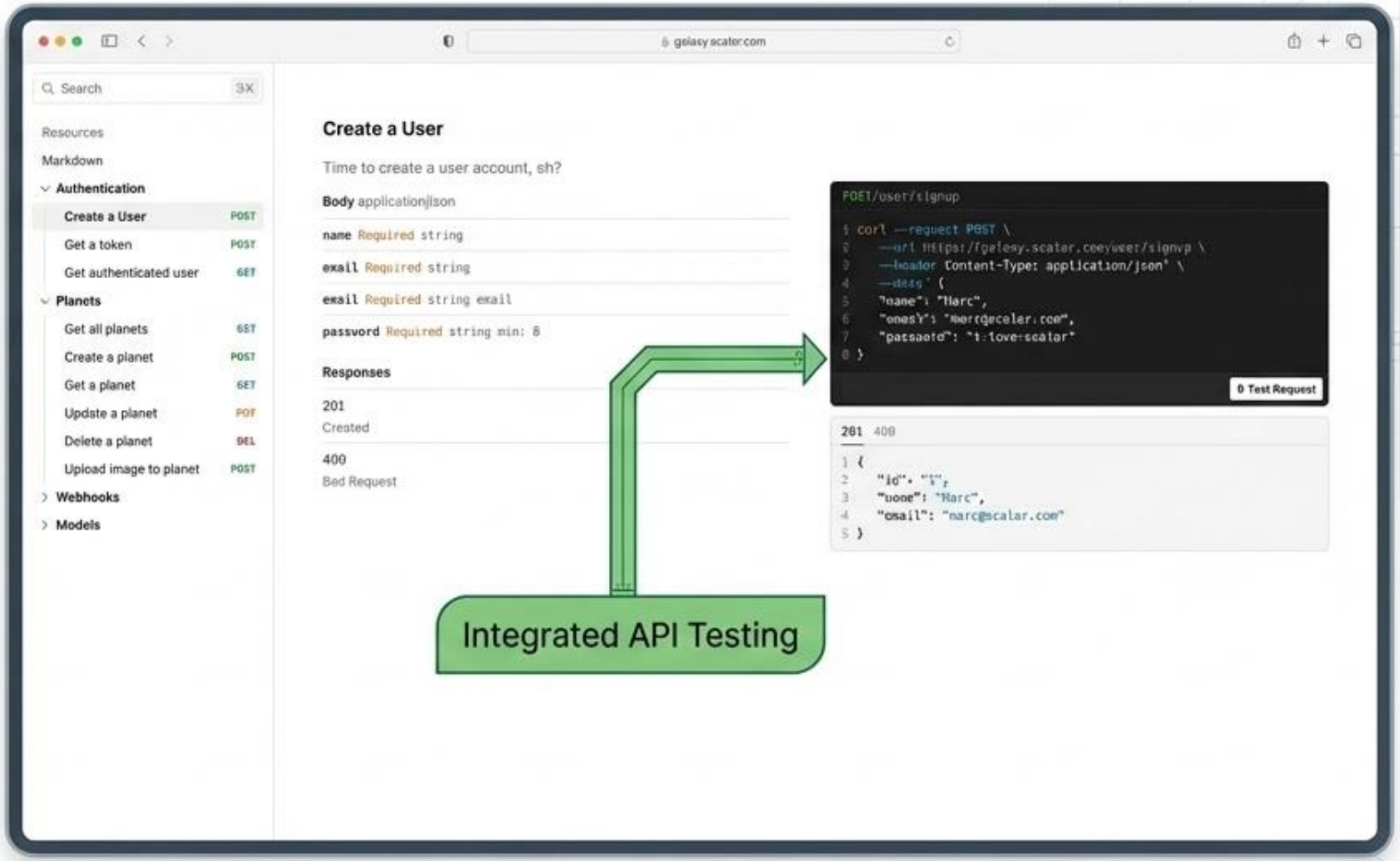
Built by engineers frustrated with traditional docs. Modern, highly readable UI.

Lightweight Integration

CDN-based integration model for effortless setup.

High-Performance

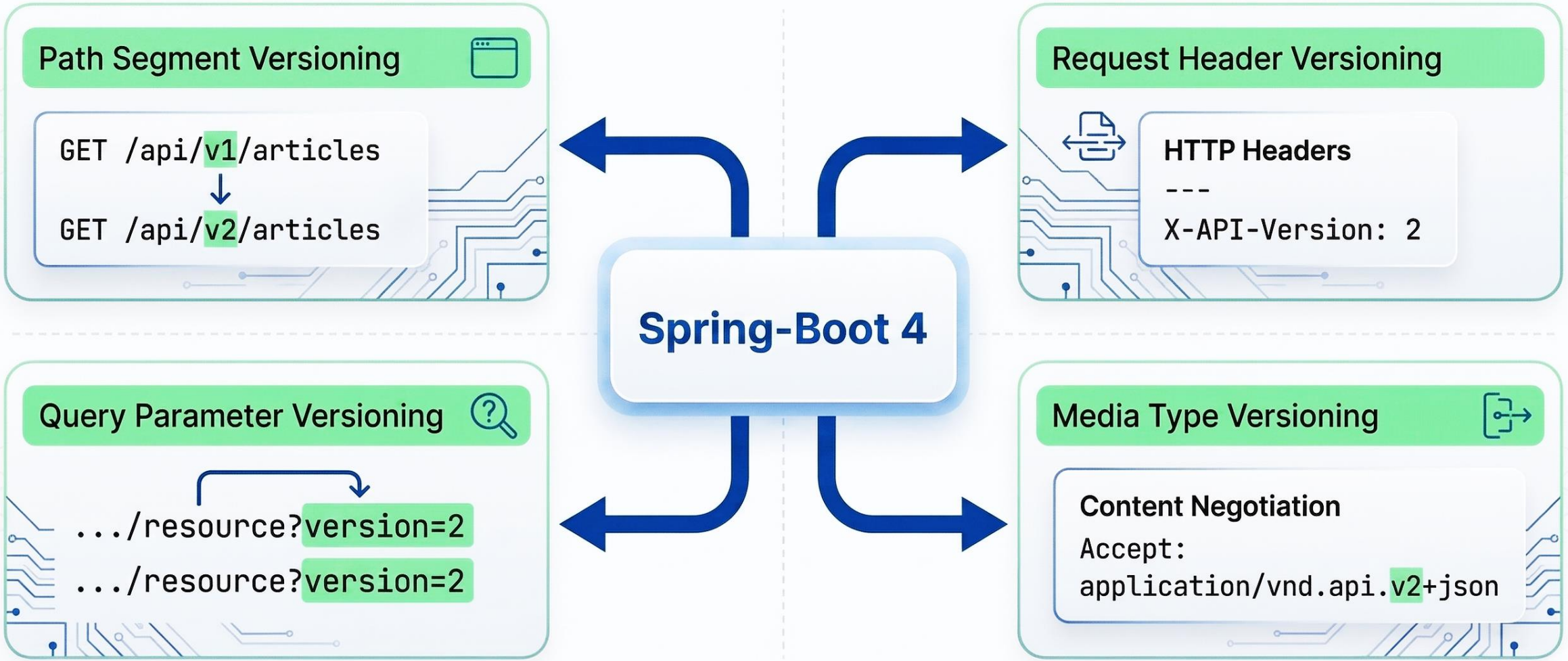
Optimized rendering that easily handles massive API specs without UI lag.



Let’s See it in Action!

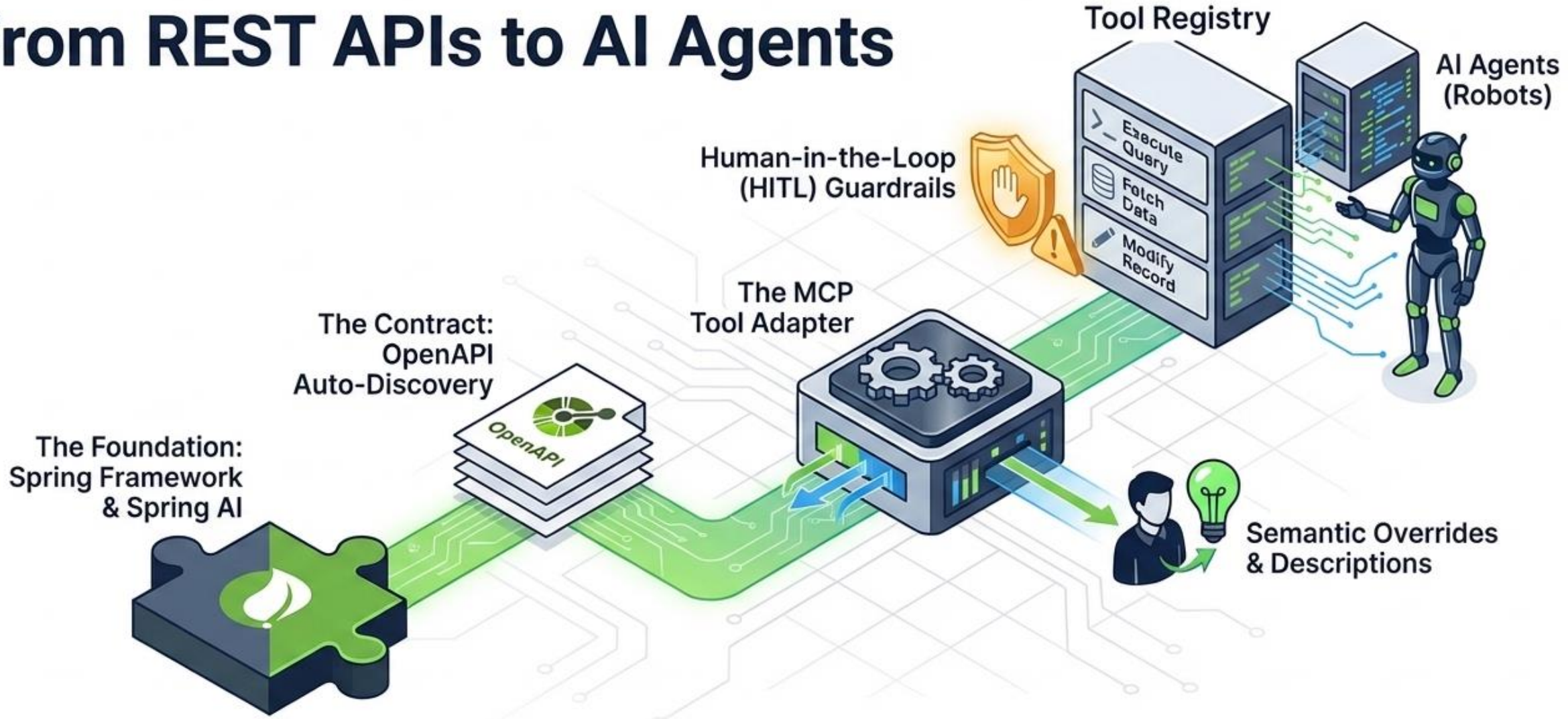
Framework-Level API Versioning Strategies

One consistent approach, zero custom routing boilerplate.



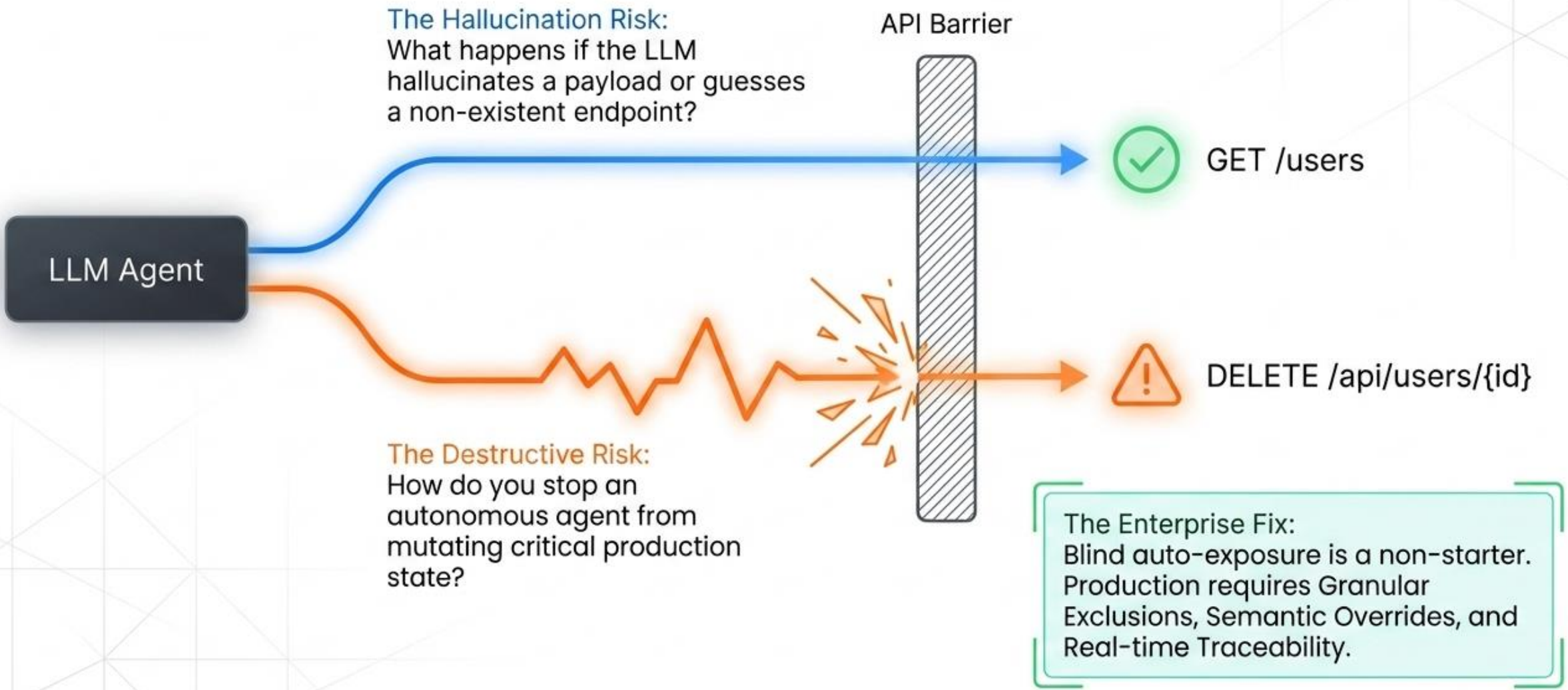
Let’s See it in Action!

From REST APIs to AI Agents



Transforming static OpenAPI documentation into a dynamic, machine-executable **Model Context Protocol** (MCP) tool registry.

The ‘Black Box’ Problem



The Paradigm Shift: From Human Docs to Machine Brains

Classic Spring Boot	AI Era Spring Boot
1 **Primary Audience:** Frontend Developers	**Primary Audience:** Autonomous LLM Agents
2 **Document Purpose:** “Try it out” in browser	**Document Purpose:** Tool Discovery & Execution
3 **Delivery Format:** Swagger UI / HTML Docs	**Delivery Format:** MCP (Model Context Protocol) / JSON Schema Tools

The OpenAPI-to-MCP Bridge

OpenAPI Field	MCP Concept
`operationId`	Tool Name
`description`	Tool Description
`requestBody`	Input Schema
`security`	Authorization Policy

Your existing OpenAPI specification is transformed dynamically into a **machine-executable tool registry.**

Zero-Manual Schemas & Semantic Overrides

```java
@RestController
@Tag(name = "Users") // Groups tools automatically for the agent
public class UserController {
    @PostMapping("/users/{id}/activate")
    @McpToolDescription(
        value = "Activate a user account. Requires valid
                 UUID. Do not use for password resets.",
        name = "activate_user"
    )
    public User activate(@PathVariable String id) { ... }
}
```

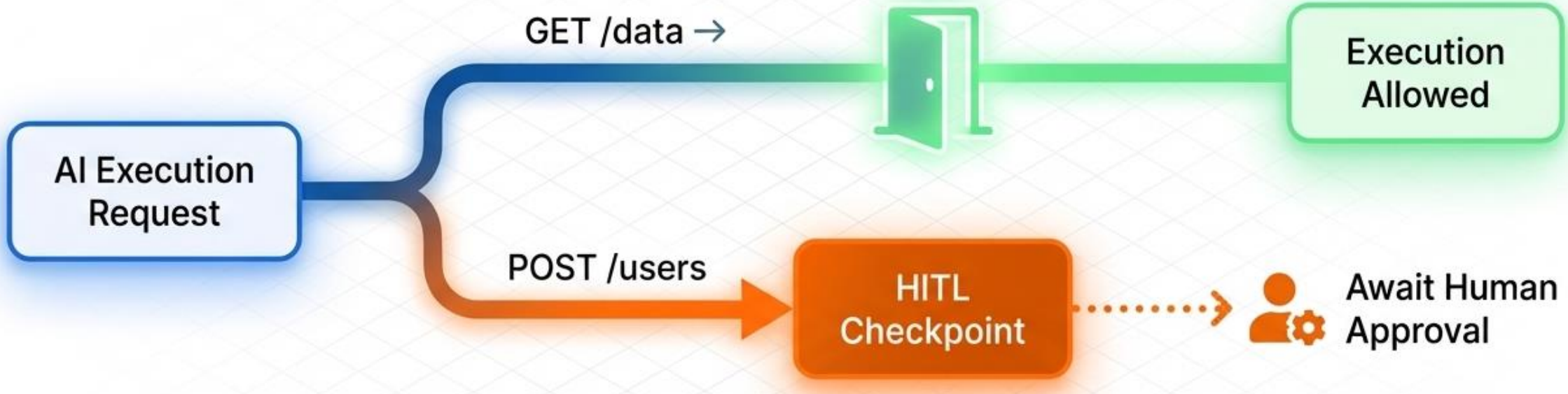
Zero Manual Schema

Springdoc scans standard endpoints and generates JSON schemas automatically.

AI-Optimized Hints

Standard OpenAPI docs are for humans; `@McpToolDescription` injects context-specific boundary instructions directly into the LLM's prompt.

Enterprise Guardrails & Execution Routing



1 Safe vs. Mutating

By default, GET/HEAD/OPTIONS are Safe. All other methods are treated as Mutating.

2 HITL Default

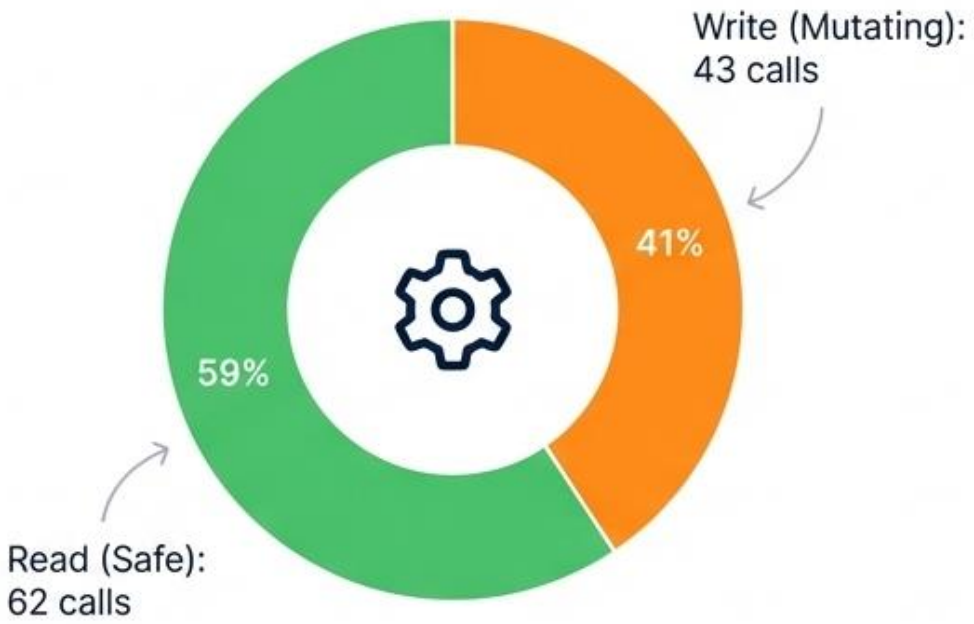
Calling a mutating tool returns a JSON payload: {'requires_human_approval': true}

3 Granular Control

Override safety dynamically via context.setSafeEndpoint(Boolean).

Developer Telemetry: The AI Health Dashboard

Read vs. Write Ratio



1
13
Tools Inventory

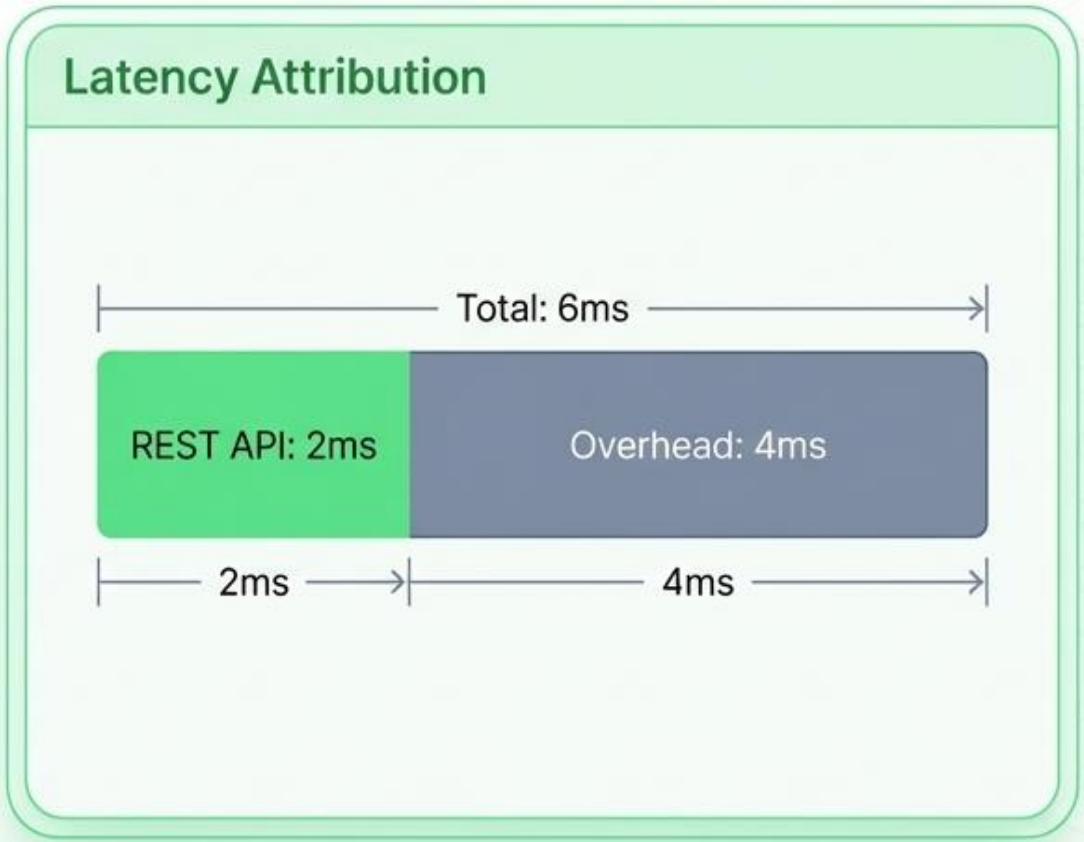
2
95%
Success Rate

3
6ms
Avg Response

Active Tools Health

	Tool Name	Status	Trend
1	list_users	Healthy	3ms
2	delete_book	Approval Required	

Context Window Protection & Latency Attribution

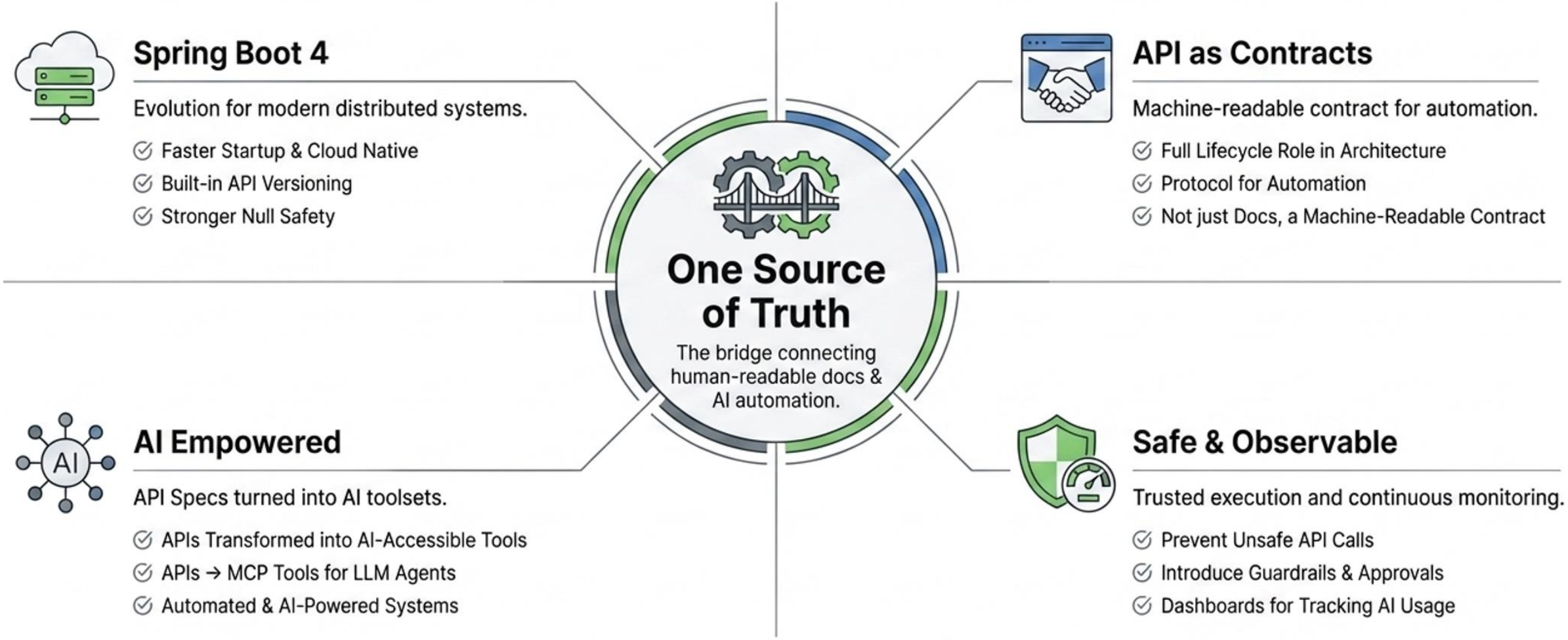


Isolate pure REST backend execution time from JSON mapping and MCP server overhead.

Context Window Protection: Auto-warns when massive payloads eat up LLM context, prompting the creation of lightweight DTOs.

Let’s See it in Action!

Architecting the AI Native Backend



SPRING IO 2026 | April 14th-15th 2026

SPRING IO 2026

Thank You

GitHub Demos

Badr NASS LAHSEN

Manager Solutions Engineering
EMEA - Palo Alto Networks

@bnasslahsen

@nass_lahsen