AI Ticket Classifier

AI Ticket Classifier

Adaptive ReAct Agent for Intelligent Ticket Classification, Enrichment & Routing — Built on Azure OpenAI · LangGraph · NestJS · Next.js

Team Bor3i
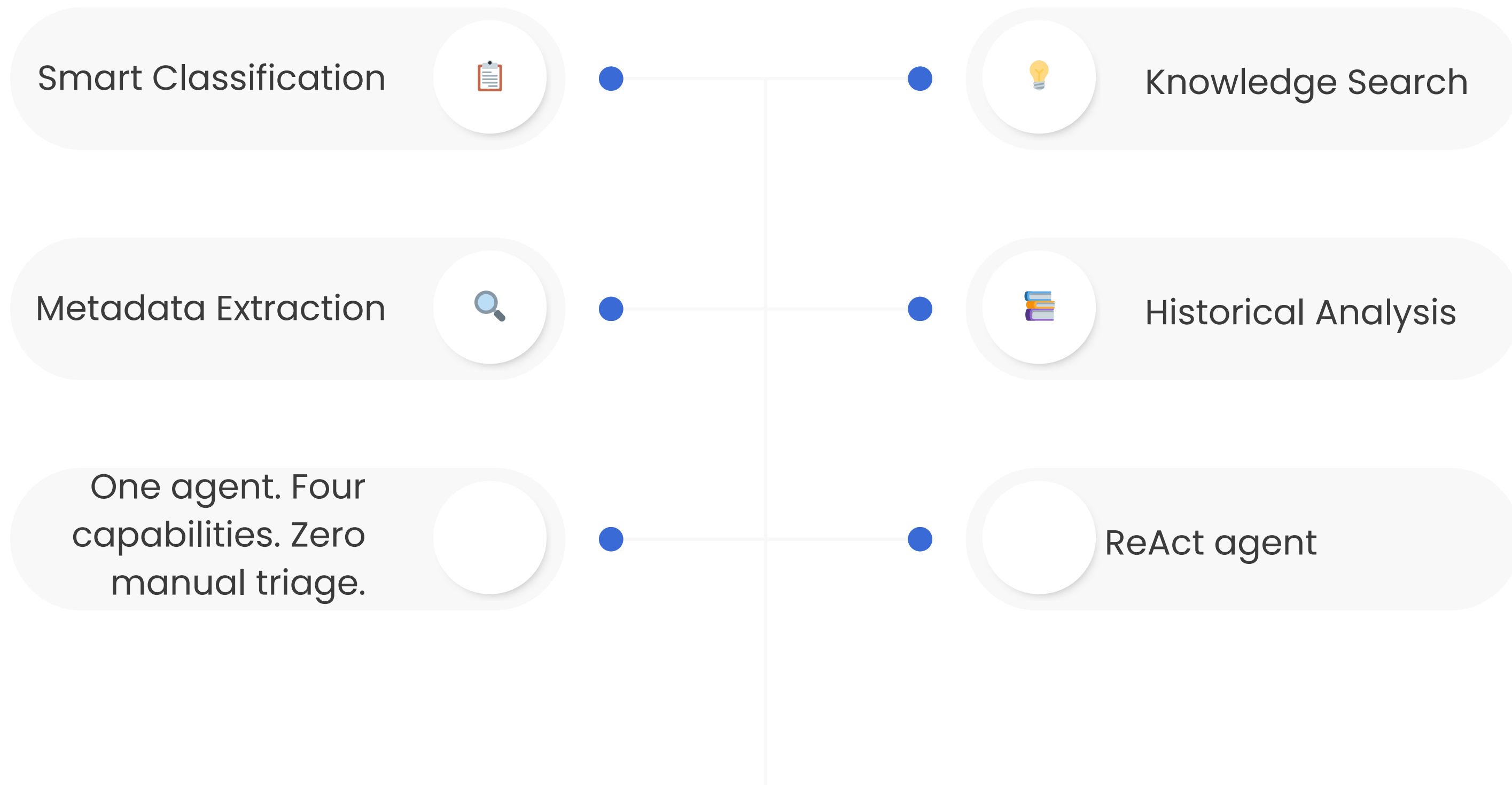
The Problem

⏱ 5–15 min manual triage per ticket

⚠ Inconsistent priority assignment across shifts

💵 Delayed P1 escalations causing business impact

Manual triage is broken at scale

The Solution

Smart Classification

Knowledge Search

Metadata Extraction

Historical Analysis

One agent. Four capabilities. Zero manual triage.
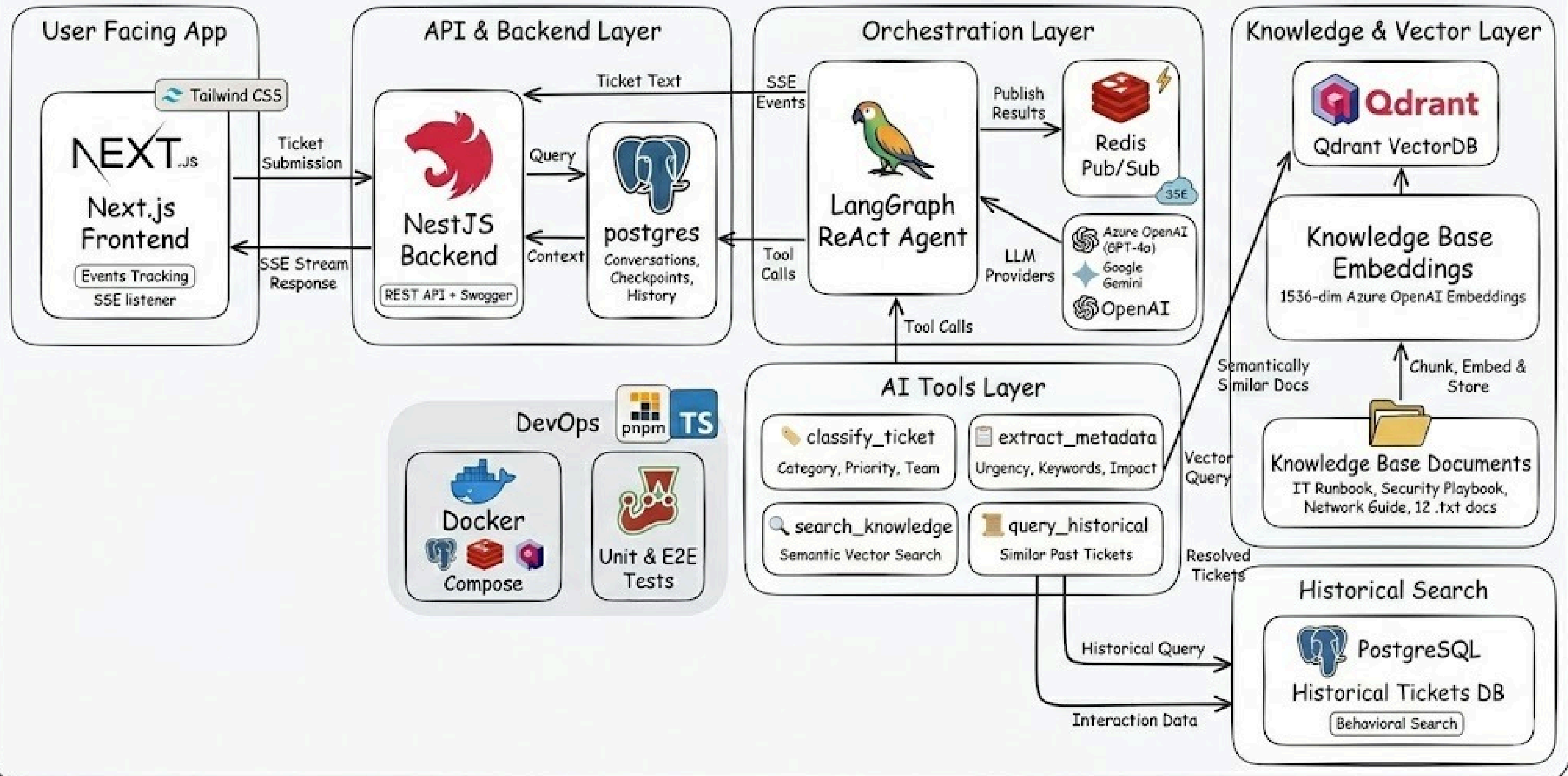
ReAct agent

System Architecture

Next.js 16 Frontend → NestJS 11 REST API

LangGraph ReAct Agent

Classification · Extraction · Knowledge · Historical

AI Ticket Classifier System

User Facing App

Next.js Frontend
Tailwind CSS
NEXT.js
Events Tracking
SSE listener

API & Backend Layer

NestJS Backend
REST API + Swagger

postgres
Conversations, Checkpoints, History

- Ticket Submission →
- ← SSE Stream Response
- Query →
- ← Context

Orchestration Layer

LangGraph ReAct Agent

- ← Ticket Text
- SSE Events →
- Publish Results →
- Tool Calls ←
- LLM Providers

Redis Pub/Sub 3SE

Azure OpenAI (GPT-4o)
Google Gemini
OpenAI

Knowledge & Vector Layer

Qdrant VectorDB

Knowledge Base Embeddings
1536-dim Azure OpenAI Embeddings

Knowledge Base Documents
IT Runbook, Security Playbook, Network Guide, 12 .txt docs

- Chunk, Embed & Store ↑
- Semantically Similar Docs
- Vector Query
- Resolved Tickets

DevOps

pnpm TS

Docker
Compose

Unit & E2E Tests

AI Tools Layer

🏷️ **classify_ticket**
Category, Priority, Team

📋 **extract_metadata**
Urgency, Keywords, Impact

🔍 **search_knowledge**
Semantic Vector Search

📜 **query_historical**
Similar Past Tickets

- Tool Calls ↑
- Historical Query →
- Interaction Data →

Historical Search

PostgreSQL
Historical Tickets DB
Behavioral Search

🧠 **THINK**

Agent reads the ticket and decides which tool to call next based on current reasoning state.

🔧 **ACT → 👁️ OBSERVE**

Invokes a specialized tool, reads the output, and updates its reasoning state. Loops until final answer is ready.

✅ **RESPOND**

Generates a fully structured JSON response: priority, category, team, urgency, affected systems, similar tickets, and recommended actions.

The ReAct Loop

05

Agent Tool Suite

🏷️ Classification Tool

Output: { category, priority P1–P4, assignedTeam, confidence, reasoning } — GPT-4o structured output

🔍 Extraction Tool

Output: { urgency, affectedSystems, userImpact, keywords, priorityScore } — GPT-4o JSON schema enforcement

Knowledge + Historical Tools

Knowledge: Ada-002 → cosine similarity over 63 Qdrant chunks. Historical: pgvector in PostgreSQL — top similar past tickets with resolutions.

Business Impact

Manual triage time per ticket (was 5–15 min) · Routing consistency, no human variance · Average response time per analysis · Thread memory, audit trail & escalation flags · Containerized microservices — scale without retraining · Junior agents get 5-year-veteran context automatically

Without AI Classifier: 5–15 min triage · Inconsistent priority · Knowledge in people's heads · No historical context · Manual escalation

With AI Classifier: <3 sec automated · Deterministic P1–P4 · Surfaced automatically · Similar tickets retrieved · Auto-flagged & routed

THANK YOU !