Enterprise Customer Onboarding Agent

Solution Design Document | February 2025 | StackAdapt Case Study

1. Architecture Overview

The solution implements an **AI-powered automation agent** using **LangGraph** for state machine orchestration and **FastAPI** for REST interface, automating customer onboarding from deal closure through SaaS provisioning.

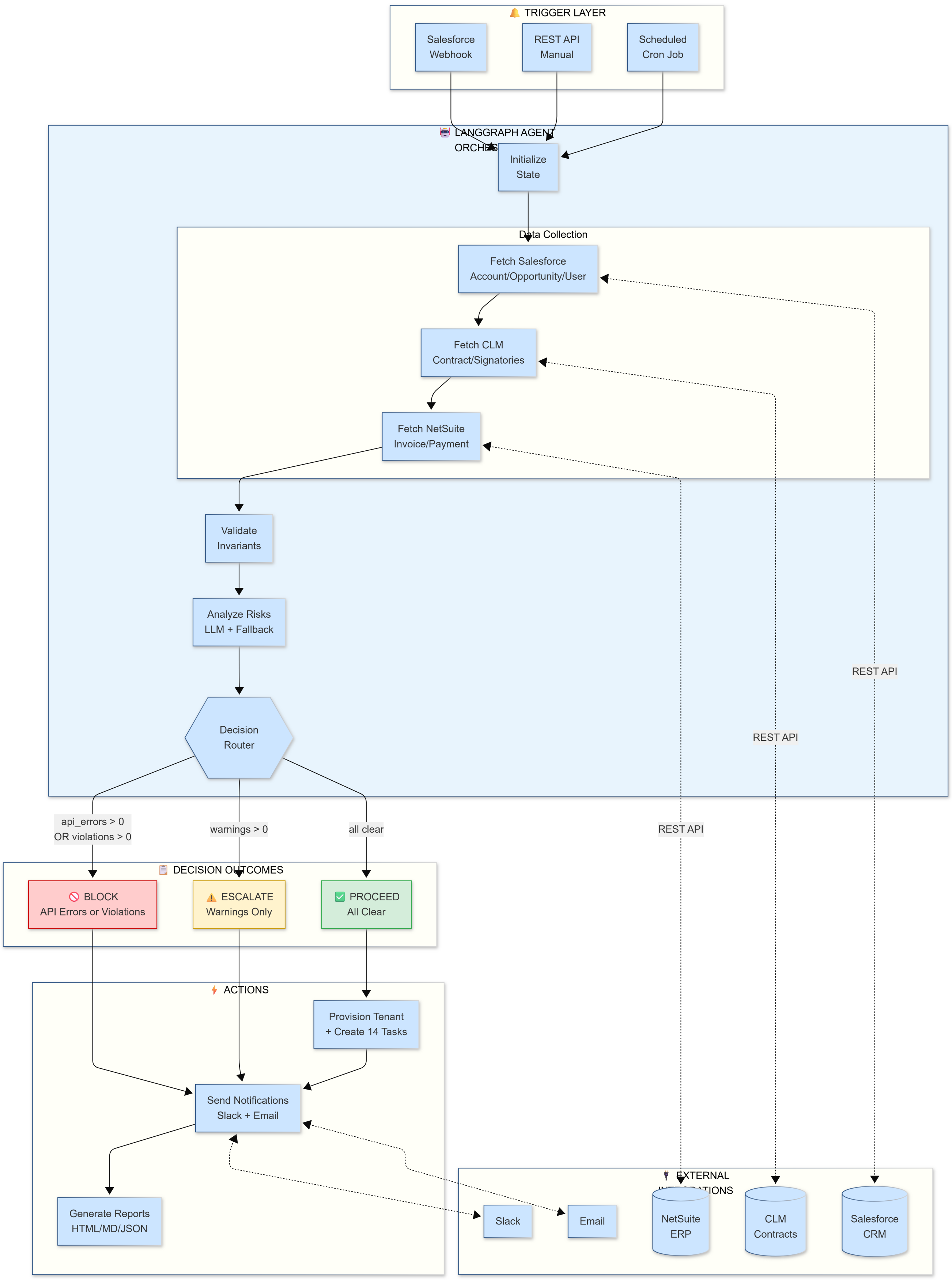


Figure 1: Architecture: Triggers → LangGraph Agent (Init→Fetch→Validate→Analyze→Decide) → Actions

1.1 System Integrations & Field Selection

System	Objects & Key Fields	Reference
Salesforce	`Account` (IsDeleted, Status), `Opportunity` (StageName, Amount), `User` (IsActive)	SF Objects
NetSuite	`Invoice` (status, dueDate, amountRemaining, paymentStatus)	NS REST API
CLM	Contract status, signatories, effective dates (Mock API)	Custom
Provisioning	Tenant creation + 14-task onboarding checklist with dependencies	Internal

Why these fields? Only business-critical fields selected: `Account.IsDeleted` for validity, `Opportunity.StageName="Closed Won"` for deal verification, `Invoice.status` for payment issues. Minimizes API payload and focuses on decision-relevant data.

2. AI Agent Application

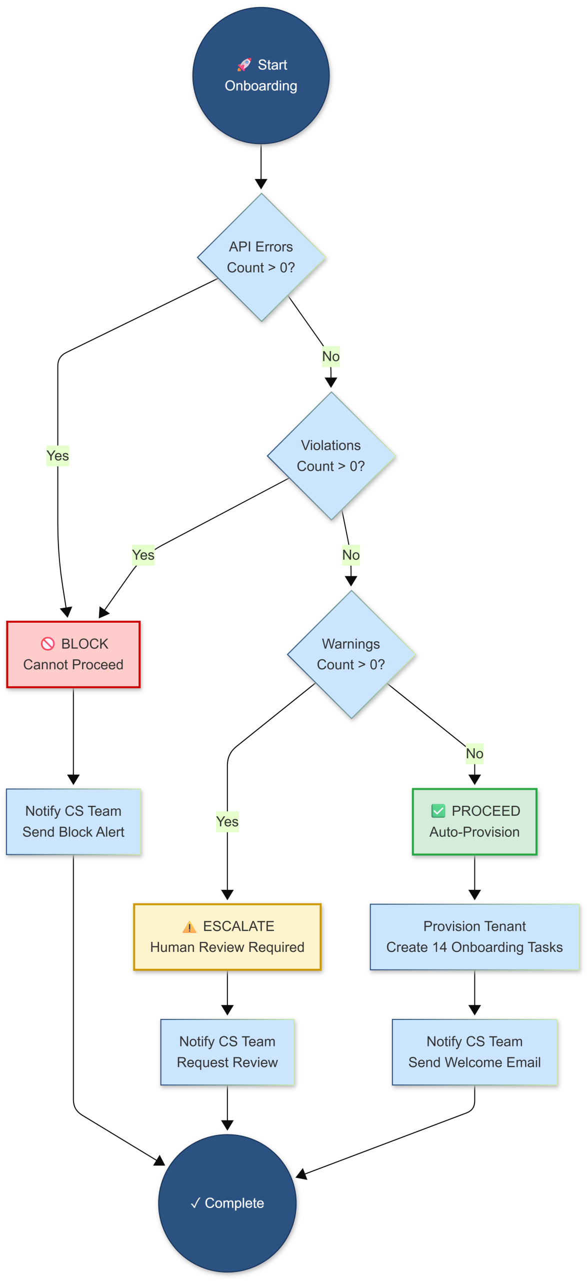
The agent uses **OpenAI GPT-4** with a deterministic rule-based fallback ensuring availability without LLM connectivity.

2.1 LLM-Powered Intelligence

- **Risk Analysis**: Evaluates API errors, violations, warnings → generates risk level (low/medium/high/critical) with business impact
- **Summary Generation**: Human-readable reports for CS teams
- **Action Recommendations**: Prioritized steps with owner assignment (CS, Finance, IT) and resolution timelines
- **Error Interpretation**: Translates `INVALID_SESSION_ID` → plain English fix instructions

2.2 Autonomous Actions

Auto-provisions tenant (tier-appropriate config) • Creates 14-task onboarding checklist • Sends Slack alerts + welcome emails • Records API failures → influences BLOCK decisions



🚀 Start Onboarding
API Errors Count > 0?
Yes
No
Violations Count > 0?
Yes
No
🚫 BLOCK Cannot Proceed
Warnings Count > 0?
Notify CS Team Send Block Alert
Yes
No
✅ PROCEED Auto-Provision
⚠️ ESCALATE Human Review Required
Provision Tenant Create 14 Onboarding Tasks
Notify CS Team Request Review
Notify CS Team Send Welcome Email
✓ Complete

Figure 2: Decision Logic: API Errors → Violations → Warnings → Proceed

3. Orchestration & Event-Driven Flows

3.1 Triggers & State Machine

Trigger	Source	Use Case	Endpoint
Webhook	Salesforce	Opportunity → "Closed Won"	`POST /webhook/onboarding`
Manual	CS Team	On-demand for specific account	`POST /demo/run/{id}`
Batch	Scheduler	Overnight processing	`POST /demo/run-all`
Task	CS Action	Mark task complete	`PUT /demo/tasks/{id}/{task}`

State Machine: LangGraph manages transitions: Init → Fetch(SF/CLM/NS) → Validate → Analyze → Decide → [Provision] → Notify → Complete. Each node handles errors gracefully, returning updated state. Conditional routing based on `api_errors`, `violations`, `warnings`.

Error Simulation: `/demo/enable-random-errors` injects configurable failures (401 auth, 400 validation, 429 rate limit, 500 server) for resilience testing.

4. Trade-offs, Assumptions & Considerations

4.1 Design Decisions

Decision	Rationale
API Errors → BLOCK	Data integrity over speed
Sync Processing	Simple for demo; prod uses queues
Rule-based Fallback	Works without LLM connectivity
14 Fixed Tasks	Predictable; prod uses templates

4.2 Scalability & Security

Scalability: Sync → SQS/RabbitMQ, Memory → Redis, Single → K8s, + LLM caching

Security: OAuth 2.0 with token refresh, audit trails with correlation IDs, PII masking in logs, RBAC for API operations

5. Multi-Agent Collaboration via MCP

The **Model Context Protocol (MCP)** enables specialized agents to collaborate through standardized tool interfaces:

Agent	Responsibility & Tools
Coordinator	Orchestrates workflow (`salesforce.*`, `provision.*`)
Contract	Monitors signatures (`clm.get_contract`)
Finance	Tracks payments (`netsuite.get_invoice`)
Task Monitor	Overdue detection (`tasks.get_overdue`)

Benefits: Domain expert agents • Shared MCP servers reduce duplication • Add agents without modifying existing • Cross-agent context via tool responses

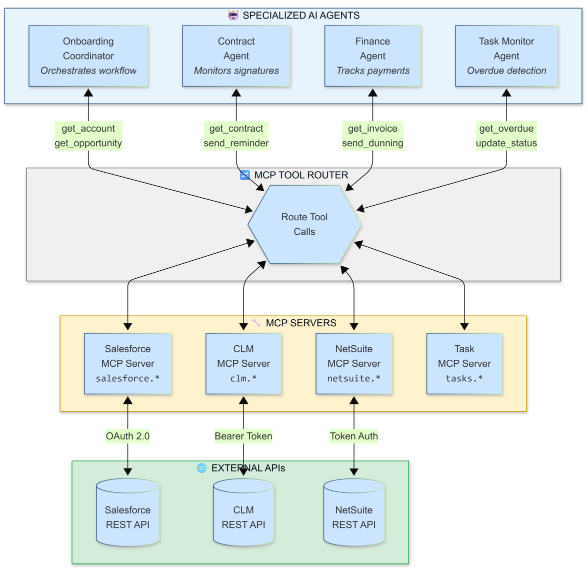

SPECIALIZED AI AGENTS
Onboarding Coordinator Orchestrates workflow
Contract Agent Monitors signatures
Finance Agent Tracks payments
Task Monitor Agent Overdue detection
get_account get_opportunity
get_contract send_reminder
get_invoice send_dunning
get_overdue update_status
MCP TOOL ROUTER
Route Tool Calls
MCP SERVERS
Salesforce MCP Server salesforce.*
CLM MCP Server clm.*
NetSuite MCP Server netsuite.*
Task MCP Server tasks.*
OAuth 2.0
Bearer Token
Token Auth
EXTERNAL APIs
Salesforce REST API
CLM REST API
NetSuite REST API

Figure 3: MCP: Agents → Router → Servers → APIs