ZACTONICS

AI AGENT HUB

Multi-Agent Command Center for Small Business

Technical Documentation & Production Architecture Guide

Version 1.0 • February 2026

Table of Contents

1. What Are AI Agents?

AI agents are autonomous software programs that perceive their environment, make decisions, and take actions to accomplish specific goals without requiring continuous human direction. Unlike traditional chatbots that simply respond to prompts, AI agents can plan multi-step workflows, use external tools (APIs, databases, file systems), maintain context across interactions, and adapt their behavior based on outcomes.

Think of an AI agent as a digital employee that has been given a specific job description, access to the tools it needs, a set of rules to follow, and the autonomy to get the work done. Just as you would hire a specialist rather than a generalist for critical business functions, AI agents perform best when they are domain-specific — each one focused on a single area of expertise.

Key Characteristics of AI Agents

- **Autonomy:** Agents operate independently once configured, making decisions without human approval for routine tasks.
- **Perception:** They ingest data from their environment — customer messages, financial transactions, website analytics, social media feeds — and interpret it using natural language processing and machine learning.
- **Reasoning:** Using large language models (LLMs) as their "brain," agents analyze situations, weigh options, and determine the best course of action.
- **Action:** Agents execute tasks by calling APIs, updating databases, sending emails, generating documents, and triggering workflows in external systems.
- **Memory:** They maintain both short-term context (within a conversation) and long-term memory (across sessions) to provide continuity and learn from past interactions.
- **Tool Use:** Agents are equipped with specific tools — search engines, calculators, API connectors, code interpreters — that extend their capabilities beyond pure language generation.

---

2. How AI Agents Work

At their core, AI agents follow a perception-reasoning-action loop. When a trigger occurs (a customer submits a question, a new transaction appears, a deadline approaches), the agent perceives the event, reasons about the appropriate response using its LLM and domain knowledge, and then acts by executing one or more tool calls.

The Agent Execution Cycle

- **1. Trigger:** An event activates the agent — an incoming message, a scheduled timer, a webhook from an external system, or a threshold being crossed.
- **2. Context Assembly:** The agent gathers relevant context: conversation history, user profile data, relevant documents from its knowledge base, and current system state.
- **3. LLM Reasoning:** The assembled context is sent to a large language model (such as Claude, GPT-4, or an open-source model) which analyzes the situation and determines what actions to take.
- **4. Tool Execution:** The LLM's output includes structured tool calls — API requests, database queries, email sends, document generation — which the agent framework executes.
- **5. Response Synthesis:** Results from tool calls are fed back to the LLM, which synthesizes a final response or determines if additional steps are needed.
- **6. Feedback Loop:** The outcome is logged, user feedback is captured, and the agent's knowledge base is updated to improve future performance.

Key Technology Components

- **Large Language Model (LLM):** The reasoning engine. Models like Anthropic Claude, OpenAI GPT-4, Google Gemini, or open-source models (Llama, Mistral) serve as the agent's brain. The choice of model impacts cost, speed, accuracy, and context window size.
- **Agent Framework:** Orchestration software that manages the agent lifecycle. Popular frameworks include LangChain, LlamaIndex, CrewAI, AutoGen, and custom Python/Node.js implementations.
- **Vector Database:** Stores embeddings of documents, FAQs, and knowledge articles for semantic search. Options include Pinecone, Weaviate, ChromaDB, Qdrant, and pgvector (PostgreSQL extension).
- **Tool Connectors:** API integrations that give agents the ability to interact with external services — CRMs, payment processors, email systems, calendars, accounting software, and more.
- **Memory System:** Short-term (conversation buffer) and long-term (persistent database) memory that maintains context across interactions.
- **Guardrails & Safety:** Input/output filters, content moderation, rate limiting, human-in-the-loop escalation rules, and audit logging to ensure agents operate safely and within defined boundaries.

---

3. Multi-Agent Architecture

The Zactonics AI Agent Hub uses a multi-agent architecture where each agent is a specialist focused on a single business domain. This design philosophy provides several critical advantages over a single "do-everything" AI:

- **Reduced Hallucinations:** Each agent has a narrow, curated knowledge base specific to its domain. A tax agent only has tax rules; a support agent only has product documentation. This dramatically reduces the chance of the AI generating plausible-sounding but incorrect information.
- **Higher Accuracy:** Specialized system prompts, fine-tuned tool sets, and domain-specific validation rules ensure each agent operates with precision in its area of expertise.
- **Independent Scaling:** High-traffic agents (like customer support) can be scaled independently of low-traffic agents (like annual report generation) — optimizing both cost and performance.
- **Isolated Failure Domains:** If one agent encounters an error, it does not affect the others. The support agent continues resolving tickets even if the content agent is being updated.
- **Easier Maintenance:** Each agent can be updated, retrained, or replaced independently without disrupting the entire system.
- **Audit & Compliance:** Domain isolation makes it easier to implement regulatory requirements, as each agent's data access and decision patterns can be audited separately.

In production, a central orchestration layer routes incoming requests to the appropriate agent based on intent classification. Agents can also communicate with each other — for example, the Sales agent may ask the Finance agent to verify a prospect's payment history before scheduling a demo.

---

4. Agent Profiles & Production Requirements

This section details each of the six agents in the Zactonics AI Agent Hub. For each agent, we cover what it does in simulation, and exactly what would be required to make it work in a real production environment — including data sources, APIs, LLM configuration, database needs, and workflow architecture.

🎧 4.1 Customer Support Agent

The Customer Support Agent provides 24/7 automated customer service. It resolves FAQs instantly, tracks orders in real-time, and intelligently escalates complex or emotionally charged issues to human agents with full context summaries. In the simulation, it demonstrates FAQ matching, order tracking via shipping APIs, and sentiment-based ticket escalation.

Capabilities & Simulation Scenarios

- **FAQ Resolution:** Classifies customer questions by intent, searches a knowledge base, and returns pre-approved answers with 97%+ confidence matching
- **Order Tracking:** Queries shipping carrier APIs to provide real-time delivery status, estimated arrival times, and proactive delay notifications
- **Ticket Escalation:** Performs sentiment analysis on incoming messages and automatically routes high-emotion or complex issues to senior human agents with a full context brief

Production Requirements

Component	Details / Requirements
LLM / Model	Anthropic Claude Sonnet or GPT-4o-mini for fast, cost-effective classification. Fine-tuned intent classifier (BERT/DistilBERT) for high-speed routing. Embedding model (text-embedding-3-small) for semantic search.
Vector Database	Pinecone, Weaviate, or ChromaDB storing embedded FAQ articles, product documentation, return policies, and troubleshooting guides. Updated via CI/CD pipeline when docs change.
Relational Database	PostgreSQL or MySQL storing ticket history, customer profiles, order records, escalation rules, and agent performance metrics.
APIs & Integrations	Shipping carriers (FedEx, UPS, USPS APIs), helpdesk platform (Zendesk, Freshdesk, or Intercom), CRM (Salesforce or HubSpot), e-commerce platform (Shopify, WooCommerce).

Tools	Sentiment analysis model (VADER or fine-tuned classifier), intent classification pipeline, knowledge base search, order lookup API, ticket creation API, email/SMS notification sender.
Data Required	Complete FAQ database, product catalog with descriptions, shipping/return policies, order history with tracking numbers, customer contact records, escalation routing rules.
Infrastructure	WebSocket server for real-time chat, message queue (Redis/RabbitMQ) for async processing, CDN for chat widget delivery, monitoring/alerting (Datadog or similar).

Production Workflow

- **Step 1 — Trigger:** Customer submits message via chat widget, email, or social media channel
- **Step 2 — Classify:** Intent classifier categorizes the query (FAQ, order tracking, complaint, general inquiry) and assigns urgency score
- **Step 3 — Retrieve:** Agent searches vector database for relevant knowledge base articles; queries order API if tracking is detected
- **Step 4 — Reason:** LLM generates a response using retrieved context, conversation history, and customer profile
- **Step 5 — Validate:** Output is checked against approved response templates; sensitive topics (refunds >$500, legal) are flagged for review
- **Step 6 — Respond:** Approved response is delivered to customer; ticket is created/updated in helpdesk; metrics are logged
- **Step 7 — Escalate (if needed):** High-emotion or low-confidence responses trigger human handoff with full conversation summary and recommended resolution

---

📈 4.2 Sales & Lead Agent

The Sales & Lead Agent monitors website visitor behavior in real-time, scores and qualifies leads using firmographic and engagement data, drafts personalized follow-up emails, and books meetings directly on your sales team's calendar. In the simulation, it demonstrates lead scoring, email drafting from conversation context, and automated meeting scheduling.

Capabilities & Simulation Scenarios

- **Lead Qualification:** Monitors visitor behavior (pages viewed, time on site, scroll depth), enriches company data, calculates a lead score, and initiates proactive chat with high-intent visitors

- **Follow-up Email Drafting:** Retrieves conversation history, analyzes the prospect's tech stack and pain points, and generates personalized emails using the appropriate template and tone
- **Meeting Scheduling:** Checks team calendar availability, matches timezone preferences, generates video conferencing links, and sends calendar invites with pre-meeting briefing documents

Production Requirements

Component	Details / Requirements
LLM / Model	Anthropic Claude Sonnet for high-quality email generation and conversation. Lead scoring model (gradient boosted tree or logistic regression) trained on historical conversion data.
Vector Database	ChromaDB or Pinecone storing conversation transcripts, email templates, product documentation, case studies, and competitive intelligence for personalized outreach.
Relational Database	PostgreSQL storing lead profiles, engagement scores, pipeline stages, email performance metrics, meeting records, and conversion tracking data.
APIs & Integrations	CRM (Salesforce, HubSpot), website analytics (Segment, Mixpanel, or custom tracking), calendar (Google Calendar, Calendly API), email (SendGrid, Mailgun), video conferencing (Zoom API), company enrichment (Clearbit, Apollo.io).
Tools	Lead scoring model, website visitor tracking SDK, company data enrichment API, email template engine, calendar availability checker, meeting link generator, CRM record updater.
Data Required	Website visitor tracking data, CRM contact/company records, historical deal data for scoring model training, email templates by persona, product documentation, calendar credentials.
Infrastructure	Real-time event streaming (Kafka or Segment) for visitor tracking, webhook receiver for form submissions, scheduled jobs for follow-up sequences, A/B testing framework for email optimization.

Production Workflow

- **Step 1 — Trigger:** High-intent visitor detected on website (pricing page, >3 pages viewed, >5 min session) or new form submission received
- **Step 2 — Enrich:** Company and contact data pulled from enrichment APIs (Clearbit/Apollo); CRM checked for existing records; tech stack and firmographic data assembled
- **Step 3 — Score:** ML scoring model evaluates engagement signals + firmographic fit against Ideal Customer Profile; assigns score 0–100
- **Step 4 — Engage:** For hot leads (score >70): proactive chat initiated with personalized opening. For warm leads: email sequence triggered

- **Step 5 — Draft:** LLM generates personalized follow-up email using conversation context, company research, and appropriate template (technical DM, executive, end user)
- **Step 6 — Schedule:** When prospect expresses interest in a call: calendar API checks availability, timezone is matched, Zoom link generated, calendar invite sent
- **Step 7 — Update:** CRM record updated with interaction details, lead score, pipeline stage change, and next action items for sales team

---

🧮 4.3 Finance & Operations Agent

The Finance & Operations Agent automates financial bookkeeping by categorizing expenses using machine learning, monitoring accounts receivable to flag overdue invoices, and generating weekly cash flow reports with trend analysis. In the simulation, it demonstrates transaction categorization, late invoice detection with automated reminders, and executive cash flow reporting.

Capabilities & Simulation Scenarios

- **Expense Categorization:** Connects to bank APIs, fetches new transactions, and uses an ML classifier to auto-categorize each one into accounting categories (SaaS, payroll, marketing, etc.) with 95%+ accuracy
- **Late Invoice Detection:** Scans accounts receivable, cross-references payment terms and due dates, identifies overdue invoices, and generates escalating reminder emails at configured intervals
- **Cash Flow Reporting:** Aggregates all income and expense streams, calculates net cash flow, compares to prior periods, and generates executive summaries with actionable insights

Production Requirements

Component	Details / Requirements
LLM / Model	Claude Haiku or GPT-4o-mini for report narrative generation. ML transaction classifier (Random Forest or gradient boosted model) trained on historical categorized transactions. Anomaly detection model for unusual spending patterns.
Vector Database	Optional — ChromaDB for storing vendor descriptions, receipt OCR text, and contract terms for intelligent matching of ambiguous transactions.
Relational Database	PostgreSQL with full double-entry bookkeeping schema: accounts, transactions, invoices, payments, reconciliation records, chart of accounts, and budget thresholds.

APIs & Integrations	Banking (Plaid for account connectivity), accounting (QuickBooks, Xero, FreshBooks APIs), invoicing platform, payment processors (Stripe, Square), email service for automated reminders.
Tools	Transaction classifier model, bank account sync via Plaid, invoice generator, payment reminder scheduler, report builder (PDF/Excel generation), anomaly detection engine, budget comparison tool.
Data Required	Bank account credentials (via Plaid), historical categorized transactions for model training (minimum 1,000+), chart of accounts, vendor master list, invoice records with terms, budget targets.
Infrastructure	Scheduled sync jobs (every 4–6 hours for bank transactions), message queue for async categorization, secure credential vault (AWS Secrets Manager or HashiCorp Vault), encrypted database connections.

Production Workflow

- **Step 1 — Trigger:** Scheduled sync pulls new transactions from bank API (every 4–6 hours), or invoice due date threshold is crossed
- **Step 2 — Ingest:** New transactions fetched via Plaid; amounts, descriptions, vendor names, and dates extracted and normalized
- **Step 3 — Classify:** ML model categorizes each transaction against chart of accounts; confidence scores assigned; low-confidence items flagged for human review
- **Step 4 — Reconcile:** Categorized transactions matched against outstanding invoices and expected payments; discrepancies flagged
- **Step 5 — Monitor:** A/R scanner checks all outstanding invoices against terms; overdue items identified and reminder sequence triggered
- **Step 6 — Report:** Cash flow engine aggregates all data; LLM generates narrative commentary on trends, anomalies, and recommendations
- **Step 7 — Sync:** Results pushed to accounting platform (QuickBooks/Xero); dashboards updated; alerts sent for budget overruns or anomalies

---

✏️ 4.4 Content Marketing Agent

The Content Marketing Agent monitors trending topics across search engines and social media, drafts SEO-optimized blog posts targeting specific keywords, and adapts long-form content into platform-specific social media posts. In the simulation, it demonstrates trend detection with content gap analysis, blog post generation with SEO scoring, and multi-platform social media adaptation.

Capabilities & Simulation Scenarios

- **Trend Monitoring:** Scans Google Trends, social media hashtags, and competitor publications to identify rising topics in your industry with traffic potential scoring
- **Blog Post Drafting:** Researches target keywords, analyzes top SERP results for structure, generates outlined drafts with proper H2/H3 hierarchy, and runs an SEO optimization pass
- **Social Media Adaptation:** Transforms blog content into LinkedIn posts, Twitter/X threads, and Instagram carousel formats with platform-specific tone, optimal posting times, and hashtag recommendations

Production Requirements

Component	Details / Requirements
LLM / Model	Anthropic Claude Sonnet or GPT-4o for high-quality long-form writing. Smaller model (Haiku/GPT-4o-mini) for social media adaptation. SEO scoring model or API (Clearscope, SurferSEO, or custom TF-IDF analysis).
Vector Database	Pinecone or Weaviate storing your existing blog content (for internal linking and gap analysis), competitor content embeddings, and brand voice guidelines.
Relational Database	PostgreSQL storing content calendar, published post records, keyword tracking data, social post performance metrics, and editorial workflow states.
APIs & Integrations	Google Trends API (or SerpAPI), social listening (Brandwatch, Mention), SEO tools (Ahrefs, SEMrush, or Moz API), CMS (WordPress, Webflow, Ghost), social media schedulers (Buffer, Hootsuite), analytics (Google Analytics, social platform APIs).
Tools	Keyword research API, SERP analyzer, readability scorer (Flesch-Kincaid), SEO optimization engine, image placeholder generator, social media formatter, posting scheduler, content performance tracker.
Data Required	Existing blog archive, target keyword list, competitor URLs, brand voice/style guide, social media account credentials, content calendar, audience engagement data by platform.
Infrastructure	Scheduled trend scanning jobs (daily), CMS webhook integration for publish events, social media API rate limit management, image generation pipeline (DALL-E or Midjourney API for blog images).

Production Workflow

- **Step 1 — Trigger:** Scheduled daily trend scan, content calendar deadline approaching, or manual request from marketing team
- **Step 2 — Research:** Trend APIs queried for rising topics; existing content audited for gaps; competitor blogs scanned for recent publications

- **Step 3 — Plan:** Content opportunities ranked by traffic potential, competition level, and relevance; matched against content calendar
- **Step 4 — Draft:** LLM generates blog post following SEO best practices: target keyword density, proper heading structure, internal linking suggestions, meta description
- **Step 5 — Optimize:** SEO scoring engine analyzes draft; readability checked; keyword placement refined; image placeholders and alt text added
- **Step 6 — Adapt:** Blog content reformatted for each social platform with platform-appropriate length, tone, and formatting (threads, carousels, professional posts)
- **Step 7 — Publish:** Content pushed to CMS as draft for editorial review; social posts scheduled at optimal times; performance tracking initiated

---

🏛️ 4.5 Tax & Compliance Agent

The Tax & Compliance Agent manages the full spectrum of small business tax operations: calculating multi-jurisdiction sales tax, identifying and tracking deductions by IRS schedule, preparing filing summaries and estimated payment calculations, and monitoring compliance obligations across all registered states. In the simulation, it demonstrates sales tax aggregation across 12 states, Section 179 depreciation analysis, quarterly/annual filing preparation, and multi-state compliance monitoring.

Capabilities & Simulation Scenarios

- **Sales Tax Calculation:** Aggregates transactions across all sales channels, applies jurisdiction-specific tax rates (state, county, city), tracks economic nexus thresholds, monitors exemption certificate expirations, and reconciles collected amounts
- **Deduction Tracking:** Scans every business expense against IRS deduction schedules (Schedule C, Section 179, home office, vehicle, travel), identifies missed deductions, and compares actual vs. simplified methods to maximize savings
- **Filing Preparation:** Calculates quarterly estimated tax payments against safe harbor thresholds, generates document checklists for CPA handoff, compares standard vs. itemized deductions, and produces pre-filled filing worksheets
- **Compliance Monitoring:** Tracks state registration status, monitors nexus thresholds in all 50 states, manages filing frequency requirements, and alerts on regulatory changes affecting the business

Production Requirements

Component	Details / Requirements
LLM / Model	Claude Sonnet for tax narrative generation and natural language Q&A. Rule-based tax calculation engine (not LLM-dependent) for precise dollar amounts. Classification model for expense-to-deduction category mapping trained on IRS guidelines.
Vector Database	Pinecone or Weaviate storing IRS publications, state tax codes, filing instructions, exemption rules, and prior year filing data for contextual Q&A and guidance.
Relational Database	PostgreSQL with comprehensive tax schema: transactions with tax jurisdiction mapping, nexus threshold tracking per state, deduction records by IRS schedule, filing deadline calendar, exemption certificates with expiration tracking, estimated payment history.
APIs & Integrations	Tax rate engine (TaxJar, Avalara, or state API lookup tables), e-commerce platforms (Shopify, WooCommerce) for transaction data, accounting software (QuickBooks, Xero), IRS e-file integration (for estimated payments), state

	revenue department portals, document storage (Google Drive, Dropbox) for filing packets.
Tools	Multi-jurisdiction tax rate calculator, nexus threshold monitor, deduction classifier and optimizer, Section 179 depreciation calculator, quarterly estimated payment calculator, safe harbor analyzer, filing deadline tracker, exemption certificate manager, CPA document packet generator.
Data Required	All sales transactions with buyer location (state/county/city), complete expense records with vendor/category, asset purchase records for depreciation, employee/contractor payment records, prior year tax returns, current exemption certificates, state registration records.
Infrastructure	Nightly transaction sync from all sales channels, quarterly batch processing for estimated payments, regulatory change monitoring feed (daily scrape of state revenue sites or subscription service), secure document vault with encryption at rest, audit trail logging for all tax calculations.

Production Workflow

- **Step 1 — Trigger:** Monthly/quarterly filing deadline approaching, new transactions synced from sales channels, regulatory alert received, or manual filing preparation request
- **Step 2 — Aggregate:** All sales transactions pulled from connected platforms; buyer locations resolved to specific tax jurisdictions using address validation
- **Step 3 — Calculate:** Rule-based engine applies correct tax rates per jurisdiction (NOT the LLM — tax math must be deterministic); nexus thresholds updated; exemption certificates validated
- **Step 4 — Classify:** Expense transactions categorized against IRS deduction schedules using trained classifier; Section 179 eligibility assessed for asset purchases
- **Step 5 — Optimize:** Deduction strategies compared (standard vs. itemized, simplified vs. actual home office, standard mileage vs. actual vehicle costs) to identify maximum savings
- **Step 6 — Prepare:** Filing worksheets generated with all required data; document checklist created; estimated payments calculated against safe harbor requirements
- **Step 7 — Deliver:** Filing packet exported to secure document storage; CPA notification sent; compliance dashboard updated; next deadline reminders scheduled

---

📊 4.6 P&L & Financial Filing Agent

The P&L & Financial Filing Agent generates comprehensive profit and loss statements, auto-prepares GAAP-compliant financial statements for quarterly and annual filings, performs ratio analysis benchmarked against industry standards, and produces investor-ready annual reports. In the simulation, it demonstrates monthly P&L with MoM/YoY comparisons, automated Q4 financial filing packages, full ratio analysis with alerts, and board-ready annual report generation.

Capabilities & Simulation Scenarios

- **Monthly P&L Generation:** Aggregates revenue by segment, calculates COGS and gross margin, compiles operating expenses by department, and generates comparative statements with month-over-month and year-over-year variance analysis
- **Quarterly Financial Filing:** Auto-generates three core financial statements (income statement, balance sheet, cash flow) in GAAP format, runs compliance checks, adds management disclosures, and formats for CPA review
- **Ratio Analysis:** Calculates profitability ratios (gross/operating/net margin, ROE, ROA), liquidity ratios (current, quick, cash), efficiency metrics (DSO, DPO, inventory turns), and leverage ratios — all benchmarked against industry averages
- **Annual Report Preparation:** Compiles full-year financials with quarterly breakdowns, revenue segment analysis, year-over-year comparisons, forward-looking projections (conservative/base/aggressive), and key operational metrics into a board-ready presentation package

Production Requirements

Component	Details / Requirements
LLM / Model	Claude Sonnet for narrative generation (management commentary, executive summaries, investor-facing language). Deterministic calculation engine for all financial math (margins, ratios, projections). Forecasting model (time series — Prophet or ARIMA) for revenue and expense projections.
Vector Database	ChromaDB or Pinecone storing GAAP standards documentation, industry benchmark data, prior year financial narratives, and investor communication templates for contextual report generation.
Relational Database	PostgreSQL with full financial data warehouse: general ledger, chart of accounts with hierarchy, revenue by segment/channel/product, COGS detail, departmental expense budgets, balance sheet accounts, prior period comparisons, and industry benchmark reference tables.
APIs & Integrations	Accounting platform (QuickBooks, Xero, NetSuite) for ledger data, payroll system (Gusto, ADP) for compensation data, revenue platforms (Stripe, Shopify) for segment detail, industry benchmark databases (IBISWorld, Sageworks, or BizMiner), document platforms (Google Drive/Sheets, board portal software).
Tools	P&L statement generator, balance sheet builder, cash flow statement engine (indirect method), financial ratio calculator, industry benchmark comparator, variance analysis engine, forecasting model (Prophet/ARIMA), chart/visualization generator, PDF/PPTX report builder, GAAP compliance checker.

Data Required	Complete general ledger with transaction detail, chart of accounts with proper hierarchy, revenue data by segment/channel, detailed COGS breakdown, departmental expense budgets and actuals, balance sheet account balances, prior year financial statements, industry benchmark data for your SIC/NAICS code.
Infrastructure	Nightly ledger sync from accounting platform, month-end close workflow automation, secure financial data warehouse with row-level access controls, versioned report storage, board portal integration for distribution, SOC 2 compliant hosting environment.

Production Workflow

- **Step 1 — Trigger:** Month-end close completed, quarterly filing deadline approaching, board meeting scheduled, or manual report request
- **Step 2 — Extract:** General ledger data pulled from accounting platform; revenue segmented by channel/product; COGS matched to revenue; expenses allocated by department
- **Step 3 — Calculate:** Financial statements generated using deterministic calculation engine (NOT the LLM for any math); all margins, ratios, and variances computed to penny-level accuracy
- **Step 4 — Benchmark:** Current period ratios compared against industry benchmarks and internal targets; alerts generated for metrics outside acceptable ranges
- **Step 5 — Analyze:** LLM generates management commentary explaining key variances, trends, and their business implications in clear, professional language
- **Step 6 — Forecast:** Time series model generates forward projections at conservative/base/aggressive scenarios using historical trends and known upcoming changes
- **Step 7 — Package:** Complete filing package assembled: financial statements, management discussion, ratio analysis, projections, and appendices formatted for CPA review, board presentation, or investor distribution

---

5. System Architecture Overview

The Zactonics AI Agent Hub is designed as a containerized microservices application using Docker Compose. In production, this architecture would be extended with the following components:

Current Simulation Stack

- **Frontend:** Single-page HTML application with Tailwind CSS, served by Nginx. Communicates with backend via REST API.
- **Backend:** Python Flask API serving static JSON data that simulates agent responses. Each endpoint mirrors what a production agent would return.
- **Containerization:** Docker Compose orchestrates both services with networking, allowing the Nginx frontend to proxy API requests to the Flask backend.

Production Architecture Extension

- **API Gateway:** Kong, AWS API Gateway, or Nginx Plus for rate limiting, authentication, and request routing to the appropriate agent service.
- **Agent Services:** Each agent runs as an independent microservice with its own container, scaling profile, and health checks. Built with LangChain or custom Python agent framework.
- **LLM Provider:** Anthropic Claude API (primary) with OpenAI as fallback. Model selection per agent based on task complexity and cost optimization.
- **Message Queue:** Redis Streams or RabbitMQ for asynchronous task processing, inter-agent communication, and event-driven triggers.
- **Databases:** PostgreSQL for transactional data; Pinecone or Weaviate for vector search; Redis for caching and session management.
- **Monitoring:** Datadog or Grafana/Prometheus stack for agent performance metrics, error rates, response times, and LLM cost tracking.
- **Security:** OAuth 2.0 / JWT authentication, encrypted connections (TLS), secrets management (Vault), SOC 2 compliant infrastructure for financial data.

---

6. Getting Started

To run the Zactonics AI Agent Hub simulation locally:

Prerequisites

- Docker and Docker Compose installed
- Git (to clone the repository)
- A modern web browser

Quick Start

- **1. Extract:** Unzip zactonics-ai-agent-hub.zip to a directory
- **2. Build:** Run `docker-compose up --build` from the project root
- **3. Access:** Open `http://localhost:8080` in your browser
- **4. Explore:** Navigate between Dashboard, Agents, and Activity tabs
- **5. Simulate:** Click any agent, select a scenario, and watch the step-by-step simulation

Project Structure

```
zactonics-ai-agent-hub/

  docker-compose.yml        — Service orchestration

  backend/Dockerfile         — Python Flask container

  backend/app.py             — API with all agent data

  backend/requirements.txt   — Python dependencies

  frontend/Dockerfile        — Nginx container

  frontend/index.html        — Full interactive UI

  frontend/nginx.conf        — Proxy configuration
```

Built with Python, Flask, Docker, Tailwind CSS, and a passion for small business automation.