subatomic

AI Co-Worker Agent Solution for Account Opening

Agentic-Core Software Architecture Blueprint

1) Agentic Architecture Drivers

Driver Type	Description	Source in Context
Workflow Automation	All account opening execution is an agentic workflow (no optionality); process must be automated and orchestrated by agents, not procedural code	Project Purpose, REQ-009
Compliance & Auditability	All workflow stages/actions must be explicitly traceable, readable, and replayable; system must generate operational/compliance event logs	REQ-005, Data Layer Spec
Variability & Evolvability	Rules, forms, and logic must be modifiable without full rearchitecture; continuous improvement and rule assessment cycles are required	REQ-006, Use Case 9
Human-in-the-Loop	Human reviewers must be explicitly modelled in workflow (exceptions, approval, DocuSign routing); fully auditable actions	REQ-003, Use Cases
Data-Centricity	Data handling follows raw → normalized → consumer output (Bronze/Silver/Gold), driven by agent needs and output stages	Data Layer, Use Cases
IT Security Constraint	No external data handling; deployment and tools must run in NorthRock's controlled environment	REQ-007, Constraints
Tool/System Modularity	System boundaries enforced via adapters; agents never directly access APIs or data stores outside tool contracts	Arch. Rule #5, Tooling
Observability	Full workflow/decision traceability (LangSmith/Deep Lens); replayable decision and memory for audit and compliance	REQ-005, Rule #6, #7

2) Selected Architectural Style(s)

Style	Role in Agentic Architecture	Non-Violable Rules
Hexagonal (Ports & Adapters) with Agentic Core	Agents and workflows are the system core; all I/O, storage, APIs as adapters/tools at edge	- Agents never communicate directly to the outside, always via adapters - Tool contracts enforce system boundaries, support plugability and LLM-agnosticism
Event-Driven	Orchestration of agents via event transitions in workflow graphs (LangGraph style); agents respond to state changes and events	- All transitions are explicit workflow events - Events are recorded, audited, and can trigger both agent and human tasks

Rationale: Hexagonal ensures clear separation of agent logic and side effects (integration/tooling), supporting modifiability/evolvability. Event-Driven aspects handle handoffs, human-in-the-loop, and system triggers with full observability.

3) Agent Inventory (Core)

Agent Name	Agent Type	Primary Goal	Decisions Made	Tech Stack
Workflow Orchestrator Agent	Orchestrator	Coordinate overall account opening workflow, enforce process sequence	Which agent(s) to activate or fork; state transitions	LangGraph/ LangChain FastAPI
Intake Agent	Worker/ Retriever	Ingest, normalize and associate all raw intake data (Formstack, statement uploads)	How to parse/classify, error/ exception routing	LangChain FastAPI
Validation Agent	Validator	Apply completeness, consistency, and custodian/ account rules; annotate packets	Which rules to apply; pass/fail/ exception	LangChain FastAPI
Packet Assembly Agent	Worker/ Assembler	Generate complete, compliant digital account opening packets and pre-populate forms	Select correct forms, map fields, structure packet	LangChain FastAPI
Human Review Agent	Human-in-the-loop	Route flagged packets for manual review/approval (exception, ambiguity, or required review)	Assign reviewer, track status, escalate timeout	FastAPI Workflow API, Adapter
DocuSign Envelope Agent	Worker	Prepare, send, and track DocuSign envelopes for signature	Envelope contents, recipients, resend/remind	LangChain FastAPI, DocuSign Adapter
Submission Agent	Worker	Submit finalized packet to custodian and record confirmation	Choose channel (API, SFTP); verify receipt; retry	LangChain FastAPI
Write-back Agent	Worker	Update Salesforce/Edge with results, artifacts, status, and ensure system sync	Which updates to post, handle failure/retry	LangChain FastAPI, SF Adapter
Audit/Trace Agent	Evaluator/ Auditor	Ensure all agent actions are captured for compliance/ops review (Deep Lens, LangSmith trace)	What to record, structure of logs	LangSmith Adapter, Audit Adapter
Continuous Improvement Agent	Evaluator/ Advisor	Periodically/evaluatively assess workflow quality and recommend/process changes	Flag optimizations, recommend rule/process updates	LangChain, FastAPI, Rule Adapter

All agents receive tool interfaces from the toolkit manager; all such tools are initialized per workflow run for audit/replayability.

4) Agentic Workflow Graph (Conceptual)

Workflow	Agents Involved	Control Flow	Failure Handling
Account Opening (Core)	Orchestrator → Intake → Validation → Packet Assembly → Human Review → DocuSign Envelope → Submission → Write-back → Audit/Trace	Sequential stages with branching: - To Human Review if packet invalid/ambiguous - DocuSign branch only if signature required - Retry loops on failures - Continuous audit at each stage	Retries (n x for I/O errors) Fallback agent if system down Escalation to reviewer/ manual ops if cannot auto-heal Complete workflow and agent traces in all cases
Workflow QA/ Process Improvement	Continuous Improvement, Audit/Trace, Orchestrator	Loop: extract metrics → advise optimization → human review → update rules/logic	Missed assessment flagged; recommended changes queued
Exception/Manual Handling	Human Review, Intake, Validation	Branches: Exception → Human Review → Correction → path restarts at prior stage	Timeout/inaction escalates; every action logged

All workflow state is externally persisted, and agent memory/store allows context-aware "restoration" for replay/audit.

5) Supporting Components (Non-Agent)

Component	Type (Service / Tool / Adapter)	Responsibility	Tech Stack
Formstack Adapter	Tool/Adapter	Retrieve forms/submissions	Python, FastAPI
Salesforce/Edge Adapter	Tool/Adapter	Client/household lookup, status/artifact write-back	Python, Simple Salesforce
Custodian Rulebook Adapter	Tool Adapter	Retrieve/process current custodian rules/templates	Python (custom), YML/ JSON
DocuSign Adapter	Tool/Adapter	Envelope creation, status polling, reminders	Python, docusign-esign SDK
Deep Lens Audit Adapter	Tool/Adapter	Write and expose audit/action logs	Python, Event Log API
Black Diamond Adapter	Tool/Adapter	Retrieve/verify client statements, intake artifacts	Python, REST (if needed)
ETL Ingestion Service	Service	Land raw intake and unstructured docs (Bronze)	Python ETL (Airflow/ Glue)
LangChain Tool Manager	Service	Runtime tool/app registration; injects adapters for agent execution	Python, LangChain
Workflow Persistence Store	Service	Store agent memory, context, and workflow state	Python, Postgres/KV store
Vector Store Adapter	Tool/Adapter	(optional) Context/memory storage for long-term agent learning/memory	Chroma/FAISS

6) Agent ↔ Data Layer Integration

Agent	Reads From	Writes To	Data Layer	Store Type
Intake Agent	Formstack Adapter, Black Diamond Adapter, File Upload	Bronze Layer	Bronze	Object/File Store
Validation Agent	Bronze, Custodian Rules Adapter, Salesforce Adapter	Silver Layer (Validation output/ audit log)	Silver	Structured Log Store
Packet Assembly Agent	Silver Layer	Gold Layer (Packet assembly area)	Gold	File/Object Store
Human Review Agent	Gold Layer (Packets)	Gold (Review artifacts/decisions), Silver (Audit event)	Gold/Silver	Structured Record Store
DocuSign Envelope Agent	Gold Layer (Packets)	Gold Layer (Envelope prep stanza), Silver (Audit)	Gold/Silver	Envelope Files/Metadata
Submission Agent	Gold Layer	Custodian Adapter (API/SFTP), Silver (Submission outcome log)	Gold/Silver	API/Log Store
Write-back Agent	Gold Layer	Salesforce Adapter (artifacts, status)	Gold	Salesforce API
Audit/Trace Agent	All agents (event bus)	Silver (Trace/log exposure), Deep Lens	Silver	Event Log/Structured Trace
Continuous Improvement Agent	Silver (logs, traces)	Silver (advisories/updates)	Silver	Log + Config Store

Data *ownership for each stage maps to agent-responsible entity. All schemas/versioning are enforced at contract level in adapters and record stores.*

7) Tool & Adapter Map

Tool / Adapter	Used By Agent(s)	External System	Pattern Applied
Formstack Adapter	Intake Agent	Formstack	Tool Adapter (REST, anti-corruption)
Salesforce/Edge Adapter	Intake, Validation, Write-back	Salesforce/Edge	Tool Adapter, Contract Mapper
Custodian Rulebook Adapter	Validation, Packet Assembly	Schwab, Fidelity rules	Tool Adapter (Snapshot, anti-corruption)
DocuSign Adapter	DocuSign Envelope Agent	DocuSign	Tool Adapter (SDK, event sender)
Black Diamond Adapter	Intake Agent	Black Diamond (optional)	Tool Adapter (REST)
Deep Lens Audit Adapter	Audit/Trace Agent, Orchestrator	Deep Lens/Log System	Tool Adapter (event bus, WORM)
Vector Store Adapter	Any (Continuous Improvement, optional Memory)	Vector DB (local/internal)	Tool Adapter (Plugable, Vector)
ETL Ingestion Service	Intake Agent	Internal ETL Platform	Service Adapter (File landing)

Every tool contract is versioned/tested independently, supporting modifiability and strict separation between agent logic and system-specific behavior.

8) Design Pattern Application Map

Concern	Pattern	Applied To	Reason
Agent Boundaries	Hexagonal (Ports & Adapters)	All agents and tool adapters	Isolates agent reasoning from integration complexity
Tooling Replacement	Adapter/Facade	All tool adapters	Allows adapter swap without agent change
Human Decision Flow	Saga/Process Manager	Orchestrator, Human Review	Ensures process state is tracked through async/human steps
Error Handling	Circuit Breaker/Retry/ Timeout	All agent↔tool boundaries	Ensures system resilience, reliable error surfaces
Memory & State	Memento/Event Sourcing	Workflow Persister, Audit/Trace	Enables replay, full audit, and compliance trace
Observability	Telemetry/Trace Decorator	All agent workflow steps	Guarantees trace output for external reporting (Deep Lens, LangSmith)
Rule/Process Evolution	Strategy/Configurable Pipeline	Validation, Packet Assembly, Continuous Improvement Agents	Rule/process changes handled via config, not code
Guardrails	Policy Decision Point/ Validator	Validation, Audit Agents	Safety via in-band policy/guard evaluation

9) Quality Attribute Drivers & Decisions

Quality Attribute	Architectural Decision	Tradeoff	Risk	Mitigation
Modifiability	All tool boundaries implemented as adapters; agent roles fixed	Adds upfront complexity (adapter layer)	Adapter drift, contract breakage	Version contract, comprehensive tests
Evolvability	Rules/logic/config externalized from agents	May limit ad-hoc nurse/ patching	Stale rules, drift	Scheduled rule review, agent test harness
Observability	Full workflow and agent event tracing (LangSmith, Deep Lens)	Storage overhead	High volume log/audit	Retention policy, index/ partitioning
Auditability	All agent actions and decisions recorded, stepwise	Performance impact for synclog	Log lag, lost events	Asynchronous, at-least-once delivery
Deployability	FastAPI microservices per agent team; version via container/tag	Deployment overhead	Orphaned/old agent versions	Orchestration layer, health checks
Safety	Validation agents as mandatory step, policy enforced	May slow workflow for "gray" input	Bottlenecks if rule ambiguous	Human escalation path, delayed processing
Compliance	All external integrations vetted, run in NorthRock IT	Limits rapid 3rd party tool use	Integration lag	Pre-approval process, clear change log

10) Deployment & Runtime View

Agent / Component	Runtime Form	Scaling Model	Isolation
Workflow Orchestrator Agent	FastAPI service	Singleton per workflow	Dedicated
Intake Agent	FastAPI service	Horizontal (per intake CUs)	Per workflow
Validation Agent	FastAPI service	Horizontal (batch/trigger)	Per request
Packet Assembly Agent	FastAPI service	Stateless on demand	Per workflow
Human Review Agent	FastAPI service	Pool (async/manual-ops)	Session per reviewer

Agent / Component	Runtime Form	Scaling Model	Isolation
DocuSign Envelope Agent	FastAPI service	Horizontal, per envelope	Per workflow
Submission Agent	FastAPI service	Stateless/concurrent	Per workflow
Write-back Agent	FastAPI service	Stateless/concurrent	Per workflow
Audit/Trace Agent	Async event-listener	Singleton per workflow	Dedicated
Continuous Improvement Agent	Batch/triggered	Singleton per assessment	Per-assessment
Each Tool Adapter	Shared lib/sidecar	As needed with agent	Memory/process
LangChain Tool Manager	Embedded/service	Shared runtime	Process
Workflow Persistence Store	DB/KV store	Centralized/HA	Logical (RBAC)

Deployment follows "agent-per-role" with horizontal scaling and failover, containerized for consistent releases.

11) Explicit Non-Goals

Area	Explicitly Out of Scope	Reason
BI/Analytics/Reporting	General analytics/reporting, downstream BI, data warehouse exports	Not in onboarding workflow scope
Model Training	Automated model retraining, hyperparameter search	No direct ML/AI beyond agent execution
General-purpose Integrations	Plug-and-play integration with arbitrary systems	Only specific platforms in context scope
Portfolio Management	Any process except onboarding/account open	Out of stated business workflow scope
Legacy Automation	Traditional RPA/non-agent automation	Agentic-first principle
Public/External Clouds	External, non-NorthRock environments	Compliance/security/risk limitation

Summary

This architecture ensures that **Agentic Workflows are the driving principle** at every level:
- Agents own and execute every core process state.
- All integrations, storage, and I/O pass through explicit adapters/tools.
- Data flows (Bronze/Silver/Gold) are mapped to agent responsibilities, supporting traceability, audit, and replay.
- Human-in-the-loop, continuous improvement, and audit are deeply embedded at the workflow and agent levels, not as afterthoughts.
- Everything is observable, modular, and agent-driven—with clear separation of reasoning, data, and system integration concerns.

Agents are not an afterthought—they are the primary runtime, the processes, and the business logic of this system.