instinctools

AI Agents for Partner Onboarding in Insurance

How a conversational, UI-driven multi-agent system helped a global insurance aggregator cut partner onboarding from **3-6 months to 2 weeks**, enable multilingual self-service for carriers and brokers, and keep quality high at optimal cost through governed model selection.

Industry:
Fintech

MULTI-AGENT AI
ENTERPRISE AUTOMATION

Business challenge

Our client, a global insurance aggregator, scales by adding new partners (carriers, MGAs, regional brokers) across dozens of countries. Each partner comes with different **APIs**, **schemas**, **languages** (including non-Latin scripts and right-to-left layouts), **and regulatory constraints**.

Historically, onboarding a single partner took **3-6 months** of cross-functional effort: clarifying requirements, interpreting sparse and heterogeneous documents, writing adapter code, preparing test data, iterating through compliance checks. Multiplied by hundreds of partners, the cost ballooned and timelines stretched.

The client needed to:

01 **Compress time-to-integration** from months to weeks without compromising compliance or auditability;

02 **Accommodate variability** in partner maturity: from bringing different API protocols (REST, SOAP, etc.) to a common denominator to handling multiple document types (PDFs, spreadsheets, or even email samples);

03 **Let non-technical partner representatives self-serve** in their native language, with a guided, transparent flow;

04 **Stay model-agnostic and cost-efficient** as the LLM landscape evolves.

Instinctools' dedicated AI team took on creating a faster and smoother partner onboarding workflow.

Solution

We delivered a **production-ready**, **UI-first**, **multi-agent system** that turns partner inputs (documents, answers, samples) into working adapters and automated tests, finishing with a GitHub pull request. The solution combines a structured AI adoption process, an orchestration pipeline that thinks before it codes, and model governance from our AI Center of Excellence.

01 Running an AI adoption workshop

To de-risk the initiative and align on outcomes, we started with the AI adoption workshop. Within it, our team:

- **Conducted business discovery**, mapping the core flows (Quote → Bind → Endorsement → Cancellation → Claims/FNOL), the country-specific nuances, and the compliance checkpoints;
- **Identified process bottlenecks and time sinks**, including document triage, field mapping, eligibility rules, error semantics, and under-specified specifications;
- **Defined clear success criteria and guardrails** with a target of ~2 weeks per partner, compile-clean builds, minimum test-coverage thresholds, and audit-ready artifacts;
- **Produced solution blueprint** around a three-phase agentic pipeline (Analyze → Plan → Generate) with human-in-the-loop validation.

02 Selecting the right model

Before designing the agentic pipeline, our AI engineers tested multiple LLMs using reverse-engineered samples from existing API adapters to see which fits the client's needs best. Among GPT, Gemini, Grok, and Anthropic (Opus and Sonnet), **Claude Opus 4.1** delivered the most stable and production-ready output, especially when guided by structured prompts and incremental checkpoints.

Model governance by our AI Center of Excellence

Choosing and re-choosing models is integral to value. Instinctools' very own AI CoE runs a repeatable framework that evaluates models by code-gen accuracy, context window, speed, modality coverage, hosting options, API availability, cost per token, language coverage, and deployment constraints. Our experts continuously benchmark new releases and propose controlled switches (with cost deltas and risk notes), so the client benefits as the market shifts.

03 Building an agentic pipeline

Agents don't jump to code. They **analyze** inputs, **plan** the work with acceptance checks, then **generate** code and tests, looping until the build is clean.

Each adapter runs through a 13-step playbook where every output becomes the next input, while the agent distills patterns from the prior implementations into a consolidated guide for what to build. A short intake questionnaire captures the business rules the docs miss, and we seed the workspace with curated reference repos so the agent can "look up" proven approaches.

With an agentic approach, a large endpoint is typically completed in 2-3 hours for roughly $50-100 of model spend.

04 Ensuring quality and auditability

Every step is pinned to hard checks, such as unit, contract (schema), and integration tests, a compile-fix loop, and smoke/HTTP probes that prove the service actually runs. Humans step in only at high-leverage moments, dedicating, on average, up to 20 minutes to polishing the output.

Under the hood, we **trace prompts**, **tool calls**, **inputs/outputs**, **latencies**, **and token spend with Langfuse**. That observability makes audits boring in the best way: each decision and artifact is tied to a PR with change logs and test evidence, ready for reviewers and regulators.

Following our AI in SDLC practices, we've also baked drift controls into prompts ("out of scope" rules), keeping models focused on what's required and nothing more. And when something slips, those traces collapse time-to-root-cause from hours to minutes, so the next run captures the fix by design.

05 Putting a premium on AI security controls

For AI-powered onboarding to be safe and sound, we had to ensure compliance with a plethora of security policies by design:

- **Core security and privacy baselines**, such as NIST and OWASP
- **Insurance and financial-sector cybersecurity regulations** by region, such as the NYDFS 23 NYCRR 500 and the NAIC Insurance Data Security Model Law for the US, the DORA for the EU, the FCA/PRA Operational Resilience for the UK, and the APRA CPS 234 for Australia, among others
- **Data privacy laws**, such as CCPA, PIPEDA, GDPR, PDPA, APPI, etc.
- **AI governance regulations**, including NIST AI RMF and EU AI Act

To check all the boxes, our team zeroed in on:

- Regional hosting options to store EU data within the EU, US data within the US, etc.
- Role-based data access
- PII protection measures like field-level masking to ensure no sensitive data is fed to the model
- Prompt "guardrails" to prevent the model from going off-scope or fabricating logic
- Human-in-the-loop reviews of AI artifacts, with all of them linked to a GitHub PR for audit trails

06 Designing self-service UI

Once the agentic approach proved consistent across 10 API adapters, our AI team wrapped the workflow in a simple web UI.

- **Partner-facing portal**. Secure onboarding wizard and chat. Partners **drag-and-drop** PDFs/Swagger/Postman/XLSX or describe their process in free text.
- **Multilingual by default**. The agent converses in the partner's language and normalizes vocabulary into the platform's canonical model.
- **Progress and transparency**. Step-by-step status, logs, downloadable reports, and validation results.

Human expertise still guides quality and regulatory soundness. The agent automates the mechanical steps, **delivering a compile-clean adapter with unit tests and CI checks**, which used to consume months of execution time.

Agent-powered partner onboarding with a human in the loop

At the end of each step of the workflow, a developer reviews the output before the model proceeds. Such application of human judgment where it matters most makes the agentic onboarding approach **predictable and trustworthy**, **not a "black box"**.

Stage	Automated by agents	Human touch
Partner intake and registration	Self-service sign-up, chat onboarding in any language, guided file upload	Provide access keys if required
Documentation ingestion	Parses PDFs/Postman collections, extracts relevant sections per step/endpoint	Upload docs, remove obviously irrelevant pages if flagged
Contract and reference analysis	Builds reports on aggregator API contract; mines patterns from prior adapters	Launch a quick sanity check for hallucinations on big reports
Implementation plan	Scans repo, identifies existing services/DTOs, proposes what to add/change	Approve/adjust plan if edge cases
Code generation (per endpoint)	Generates services/DTOs/mappers/tests aligned to contract	—
Build and self-fix loop	Compiles project, iterates on compile errors until green, starts app	—
Smoke checks and PR	Optional HTTP smoke tests, opens a GitHub PR with artifacts	Review PR, merge
Orchestration and audit	One-click run from UI, step-by-step logs/metrics (LangFuse)	Monitor runs, rerun if a step stalls
Compliance and escalation	Routes ambiguity/questions via chat, escalates to legal/SME when needed	Answer a short checklist (e.g., coverage limits, exclusions, etc.)

Before

- 3-6 months to onboard a new partner
- Fragmented documentation in multiple languages and formats
- Slow feedback loops and unclear ownership of quality gates
- Client's software engineers do all the groundwork manually
- Each adapter is treated as a one-off build

After

- 2 weeks of onboarding time per partner, end-to-end, including review and PR
- Governed model policy that keeps quality up and token costs down
- Client's engineers take on the human-in-the-loop role while agentic AI carries on the integration process step-by-step
- Knowledge now compounds: the more adapters are built, the easier it is to build the next ones

Business value

- **Up to 12× faster** partner onboarding
- **10× decrease** in operational costs
- **80–90% less** repetitive development work
- **Near-zero rework** on recurring issues
- **Full auditability** of AI actions for compliance

Multiplier effect

With the hard parts of partner onboarding automated and a **model strategy that evolves with the market**, the platform can expand faster into new countries and niches. As the library of reference adapters grows, each subsequent integration benefits from **pattern reuse**, shrinking effort even further.

Do you have a similar project idea?

Contact us — and we will estimate your projects costs for free!

CONTACT US