Research & Engineering Insight Document

AgentCheck: AI Agent for Automated Qualification Verification

1. Research Insight

AI Agent Patterns Studied

Pattern	Considered	Trade-offs
ReAct	Interpretable reasoning	Verbose, slower, expensive
Function Calling	☑ Selected	Structured tool invocation, less reasoning visibility
Plan-and-Execute	Considered	Plan may become stale mid-workflow
Multi-Agent	☑ Selected	Modularity vs. coordination overhead

Architecture Decision: Multi-Agent + Function Calling

I selected this hybrid approach for three reasons:

1. **Separation of Concerns**: Each agent has clear responsibility—ExtractionAgent (documents), EmailAgent (communication), DecisionAgent (compliance reasoning)
2. **Auditability**: Distinct agents make it easy to trace who did what—critical for RegTech compliance
3. **Right-sized Flexibility**: Function calling is applied only where ambiguity exists (interpreting university replies), not where determinism is needed (PDF parsing, workflow order)

Why Not LangChain?

Consideration	Function Calling	LangChain
Dependencies	Zero new packages	50+ packages
Debugging	Direct stack traces	Framework abstractions
Scope fit	Matches prototype scale	Over-engineered

Key insight: **Start simple, add complexity when earned.** I can add LangChain later if scale demands it.

2. Engineering Solution Thinking

Balancing Hard-Coded Logic vs. Generative Reasoning

Step	Approach	Reason
Workflow order	Hard-coded	Must be consistent for audit

Step	Approach	Reason
Field extraction	LLM	Handles certificate format variations
University lookup	Database + fuzzy match	Deterministic, fast
Email drafting	LLM	Natural language generation
Reply analysis	LLM + Function Calling	Handles varied response formats
Compliance mapping	Hard-coded	Regulatory requirement

Key Architectural Decisions

1. **Tools as First-Class Citizens**: Each tool (`parse_pdf`, `extract_fields`, `draft_email`, etc.) is a discrete, testable unit with clear interface contracts
2. **Externalized Prompts (Jinja2)**: `config/prompts/*.j2` enables non-engineers to modify prompts without code changes
3. **Config-Driven University Mappings**: JSON config allows business users to add universities without deployment

Failure Handling

Failure Case	Mitigation
PDF unreadable	LLM Vision API for scanned documents
University not found	Mark as INCONCLUSIVE, flag for review
Ambiguous reply	Lower confidence → human review queue
LLM hallucination	Schema validation + structured outputs

Audit Trail Design

Every action creates an `AuditLogEntry` with: timestamp, step number, agent, tool, sanitized inputs/outputs, and success status. This ensures complete traceability for regulatory compliance.

3. Real-World Applicability

Production Extensions

Current	Production Path
Simulated outbox/inbox	SendGrid/AWS SES + IMAP/Gmail API
Sample PDFs	LLM Vision API (handles scanned certificates)
JSON university config	Database + University verification APIs (e.g., National Clearinghouse, HEDD UK)

Scaling to 1,000+ Checks/Day

Architecture: **FastAPI → Redis Queue → Celery Workers (horizontal scaling)**

Key components: Rate limiter per university, exponential backoff retries, dead letter queue for failed tasks.

4. Security & Compliance Thoughts

Data Privacy

- Audit logs sanitize sensitive data automatically
- Production: Encrypt at rest (AES-256) + in transit (TLS 1.3), GDPR/CCPA compliance

Model Hallucination Risks

- Schema validation on all LLM outputs
- Confidence thresholds: `< 0.7` triggers human review
- Deterministic keyword fallback for critical decisions

Human-in-the-Loop

Scenario	Action
Low confidence (`< 0.7`)	Human review queue
INCONCLUSIVE result	Human review queue
NOT_COMPLIANT affecting employment	Human confirmation required
Anomaly detected	Flag for review

Only high-confidence COMPLIANT results auto-approve. All others require human verification.

Conclusion

This multi-agent architecture balances **reliability** (structured tool execution), **flexibility** (LLM-powered analysis where needed), **auditability** (comprehensive logging), and **extensibility** (modular design). The system is designed with RegTech compliance requirements at its core—every decision can be traced, explained, and audited.

Document Version: 2.0.0
Last Updated: December 2024