DMPedia Lecture Notes in Computer Science & Engineering
Volume 1, Issue 2, pp. 205-216, 2026
ICICCS-2026

Cultivate: A Configurable AI Agent for Scaling Agricultural Extension Services in Ghana

Kelvin Kekeli Kobla Ahiakpor, David Ebo Adjepon-Yamoah
Department of Computer Science & Information Systems, Ashesi University, Ghana
kelvin.ahiakpor@ashesi.edu.gh, dadjepon@ashesi.edu.gh

ABSTRACT

Ghana's agricultural extension system operates at a farmer-to-extension officer ratio of 1:1,500 three times the FAO-recommended standard of 1:500. Agronomists cannot serve the farmers who need them, and no existing system lets them train and deploy their own configurable AI agents. This paper presents Cultivate, an AgroAgent-as-a-Service (AaaS) platform that enables agronomists to upload domain knowledge, configure agent behaviour, and deploy AI agents that answer farmer queries at any time, unlike Farmer. Chat and FarmerAI, which use centrally curated knowledge bases, Cultivate gives each agronomist direct control over their agent's knowledge, communication style, and escalation threshold. The platform is built on Next.js, Mastra AI, Claude Sonnet 4.5, and Supabase with pgvector for retrieval-augmented generation (RAG). A pilot with Farmitecture a Ghana-based urban farming startup with 70+ customers and two agronomists validated the system with one agronomist and one farmer. The agronomist set up a fully configured agent in under ten minutes without technical guidance and estimated that the platform could handle 50-60% of his query load. The farmer said responses felt specific and grounded. The pilot demonstrated that decentralised, agronomist-trained AI agents are technically feasible and well-received by their intended users. Limitations include a small pilot sample, heuristic confidence scoring, and no WhatsApp integration yet.

Keywords: agricultural extension services, configurable AI agents, retrieval-augmented generation, Agent-as-a-Service, urban farming, human-in-the-loop, Ghana, knowledge transfer

1. Introduction

Ghana's farmer-to-extension officer ratio is 1:1,500 three times the FAO-recommended standard of 1:500 [1][2][3]. A single officer must technically visit 4,500 farmers within three months, which is not possible. This staffing gap is one reason Ghana spends $2 billion annually on food imports despite its fertile land and favourable climate. As Ghana's Minister of Food and Agriculture stated: "Without adequate extension services, even the best agricultural policies and research breakthroughs will not reach our farmers" [4].Urban and peri-urban farming compounds the problem. Farmers growing leafy greens, tomatoes, okra, and pepper in urban settings [5] increasingly use hydroponic, vertical, and aeroponic systems that need continuous expert support. Farmitecture, a Ghana-based agritech startup, serves over 70 customers with two agronomists. The ratio is not sustainable.AI-powered agricultural advisory systems already exist Farmer. Chat [6][7], FarmerAI [8][9], Nuru AI [10], and Darli AI [11] and they show that farmers will use AI for advice. But all of them manage knowledge centrally. Individual agronomists cannot train their own agents, protect their IP, or tailor advice to the specific products and practices of their organisation. The knowledge an experienced officer builds over the years stays locked in their individual practice, unavailable to the farmers who need it at 10 pm on a Saturday. Cultivate was built to address that gap. The distinction this paper draws on is between AI-assisted advisory tools and genuinely agentic agricultural systems. Current platforms are AI-assisted: they match queries against centrally curated content. Cultivate is designed as an agentic system in the sense described by Bonifacio et al. [15] and Gavai & Heringa [16]: goal-directed, state-aware, adaptive, and operating within a human-in-the-loop feedback structure. Each deployed agent maintains a persistent conversation context, retrieves from an agronomist-curated knowledge base, and routes low-confidence responses to human review.

This paper makes the following contributions:

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering.
ISBN: 978-81-993813-1-5

(1) An AgroAgent-as-a-Service architecture enabling individual agronomists to train and deploy configurable AI agents without technical expertise.

(2) A dual-user system design covering both sides of the advisory relationship: agronomists as trainers and overseers, farmers as end users.

(3) A pilot evaluation with Farmitecture demonstrating technical feasibility and user acceptance in a real deployment context.

(4) An analysis of where Cultivate sits on the AI-assisted to agentic continuum, and what that means for trust, governance, and future development.

The central research question: *Can a configurable, agronomist-trained AI agent handle routine agricultural extension queries, and how do agronomists and farmers respond to such a system?*

2. From AI-Assisted to Agentic Agricultural Advisory

2.1 The Current Paradigm and Its Limits

Most agricultural AI systems today are AI-assisted: they observe data, produce an output, and pass it downstream. They do not maintain an explicit decision state, reason over competing objectives, or revise behaviour based on observed effects. Bonifacio et al. [15] describe these as L0-L1 systems on a continuum from isolated analytics to closed-loop agentic farming. The platforms that dominate agricultural advisory Farmer.Chat, FarmerAI, Darli AI sit in this range. They are well-built, widely used, and genuinely useful. But they share a structural property: the platform provider controls the knowledge.Gavai & Heringa [16] identify a parallel concern in AI for food systems more broadly. When datasets are proprietary or biased toward platform providers' priorities, autonomy can reinforce existing asymmetries. The critical question is not whether agents can coordinate, but who they coordinate for. In agricultural advisory, the answer should be: for the agronomist and the farmer, not the platform.

2.2 What Agentic Advisory Requires

Translating Bonifacio et al.'s framework [15] to the advisory context, an agentic agricultural advisory system should satisfy four properties: it must be goal-directed (organized around the agronomist's configured objectives), state-aware (maintaining conversation history and farmer context), action-oriented (routing queries to the right output path direct response, flagged review, or human escalation), and adaptive (learning from agronomist corrections and feedback over time).

Table 1 maps Cultivate against this continuum. The system sits at L2-L3: it integrates specialised retrieval, multi-turn memory, and bounded autonomy with human oversight, but does not yet support long-horizon planning or autonomous identification of knowledge gaps.

Table 1: Cultivate on the AI-Assisted to Agentic Continuum

Level	System Role	Cultivate Status
L0	Isolated analytics	Below scope – no static prediction only
L1	Decision support	Exceeded – agents' route, not just advise
L2	Coordinated semi-agentic	Achieved – RAG, memory, confidence routing
L3	Closed-loop agentic	Partial – human loop active; long-horizon planning not yet built

2.3 The Decentralisation Argument

The architectural choice that separates Cultivate from existing platforms is decentralised knowledge ownership. Each agronomist controls their own knowledge base, agent persona, and escalation threshold. This is not a cosmetic difference. Centralised curation means slow update cycles, no IP protection for individual practitioners, and no path for an organisation to encode product- or locally specific knowledge. Cultivate inverts this: the agronomist is the knowledge owner, trainer, and overseer. The platform provides the infrastructure; the agronomist provides the intelligence.The tradeoff is accountability. Centralised platforms can vet knowledge before it reaches farmers. Cultivate trust with the agronomist. The confidence scoring and flagged query workflow is the primary safeguard but it catches uncertain responses, not incorrect ones. This is a genuine limitation, and one that future work needs to address by conducting a structured knowledge review at upload time.

3. Requirements Analysis

3.1 Data Collection

Requirements came from a literature review of existing systems, structured surveys, and follow-up interviews. Two surveys were administered: one for agronomists (30 questions, n=3) and one for urban and peri-urban farmers (36 questions, n=12). Participants were drawn from Farmitecture's team and customer base, as well as from two other agricultural organisations in Ghana.

3.2 Survey Findings

All three agronomists reported responding immediately to roughly half their inquiries. Simultaneous incoming questions, field visits, and the need to verify information first were the main blockers. The most experienced agronomist with over 10 years in the field spent up to 10 hours per week answering repetitive questions. Two of three estimated that an AI agent could handle up to 50% of queries autonomously; the third estimated 25%.On the farmer side, 50% reported difficulty reaching an extension officer when needed, and 25% did not know who to contact. Ten of twelve farmers (83%) said they would use a 24/7 AI advisory system. The top trust factors were advice grounded in agronomist knowledge (83%), recommendations from other farmers (67%), and the option to escalate to a human. The most requested features were photo-based pest and disease diagnosis (10 of 12), instant answers (9 of 12), and personalised advice (8 of 12). Two-thirds found online searches too general for their specific crop or problem; half could not find content relevant to Ghana's conditions.

These findings echo what Gavai & Heringa [16] describe for food systems AI broadly: contextual reasoning and local grounding are exactly what task-based, centralised systems fail to provide.

3.3 Requirements Summary

Requirements were prioritised using MoSCoW. Must Have items include: agronomist knowledge base upload (FR-A1), agent configuration (FR-A3), natural language farmer query (FR-F1), context-aware responses (FR-F2), human escalation (FR-F5), RAG-based retrieval (FR-S2), confidence scoring and automatic flagging (FR-S3), multi-tenant data isolation (FR-S5), and query routing (FR-S6). Key non-functional requirements: response time under 5 seconds (NFR-U1), agronomist setup without technical training in under 30 minutes (NFR-U3), and factual accuracy in 90% of agronomist-reviewed cases (NFR-S4).

4. System Architecture

4.1 Overview

Cultivate is a layered monolith with event-driven streaming, deployed as a Progressive Web App (PWA) on Vercel. Figure 1 shows the overall system structure. It has five layers: authentication and authorisation, RESTful API, conversational AI agent with RAG, real-time communication via Server-Sent Events (SSE), and a role-based frontend. The Agronomist

Dashboard is a desktop-first web interface for knowledge upload, agent configuration, flagged query review, and analytics. The Farmer Interface is a mobile-optimised PWA for chat, photo uploads, and conversation history.

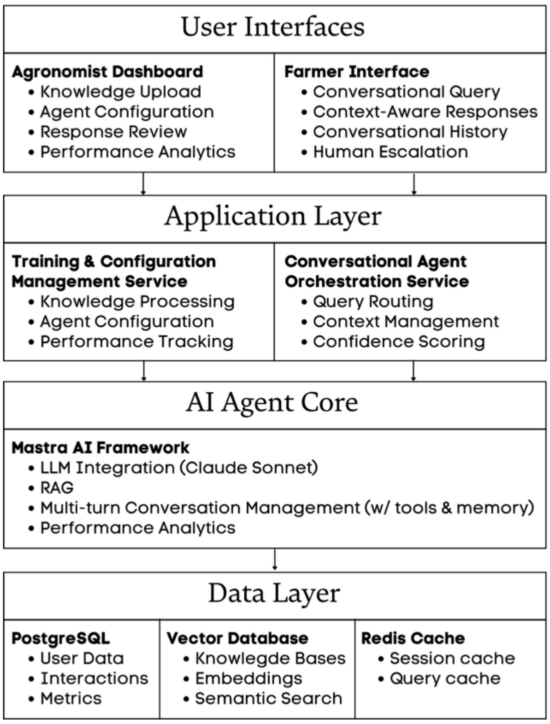

Figure 1: Cultivate's layered system architecture showing the five-layer monolith and the two user-facing interfaces.

4.2 Multi-Tenancy and Data Isolation

Every database resource agents, conversations, knowledge bases, flagged queries, and API usage records carries an organizationId foreign key. All API routes scope queries to the session's organizationId. One organisation cannot read another's data. This design supports multiple tenants in a single database instance without separate schemas per client, addressing FR-S5 and NFR-S5.

4.3 AI Agent Engine

Following Bhagwat's taxonomy [12], Cultivate's agents operate at a low-to-medium level of autonomy. They handle routine queries, retrieve from the agronomist's knowledge base, and route low-confidence responses to human review. The dynamic agent pattern drives the AaaS model: each agent's system prompt, response style, and confidence threshold are loaded from the database at runtime. Two agronomists can deploy agents with entirely different personas and escalation thresholds, without any code changes.

Agents maintain hierarchical memory with two layers per conversation, as illustrated in Figure 2. Short-term memory is a sliding window of the last 10 messages injected into the context on every query. Long-term memory is semantic retrieval via RAG: the farmer's message is embedded with Voyage 3.5-lite, cosine similarity search finds the top 10 relevant chunks from the agent's assigned knowledge bases, and those chunks are re-ranked by priority type (CORE, RELATED, or GENERAL) before injection into Claude's system prompt. This keeps responses grounded in the agronomist's uploaded content rather than in the LLM's general training.

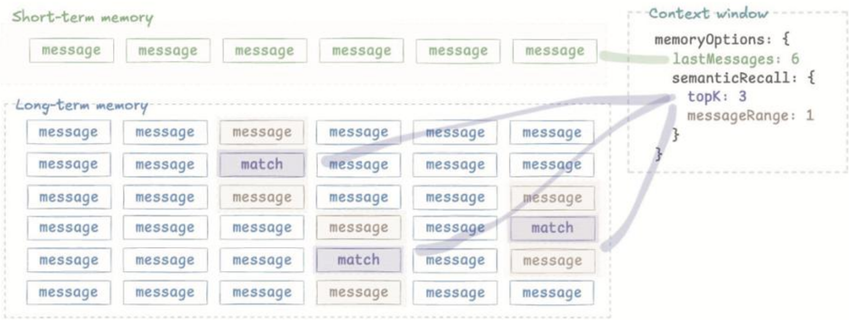

Figure 2: Hierarchical memory retrieval: short-term sliding window (last n messages) combined with long-term semantic recall via RAG, injected into the agent's context on every query.

Guardrails middleware restricts agents to agriculture-related topics. Agents decline off-topic queries and redirect farmers to farming-relevant questions. This runs at the agent perimeter, not inside Claude's reasoning loop. Access control at the API layer ensures that farmers can only access agents within their organisation.

4.4 Human-in-the-Loop Workflow

A confidence scoring module scores each AI response between 0 and 1, based on whether RAG context was retrieved, whether conversational history was present, response length, and absence of hedging phrases. Responses below the agronomist's configured threshold (default 0.7) are automatically flagged and placed in the Flagged Queries dashboard. Agronomists can verify or correct them; farmers receive a push notification when their flagged query is reviewed.This is the suspend-and-resume pattern described in [12]. Routine, high-confidence queries go straight through. Edge cases surface to the human expert. The design reflects what Gavai & Heringa [16] describe as bounded agency: user and agronomist feedback constrains algorithmic initiative, reconciling automation with human oversight. As Bonifacio et al. [15] note, human oversight must remain a structural component of agentic agricultural systems – not a post-hoc check, but a design requirement.

4.5 Use Cases

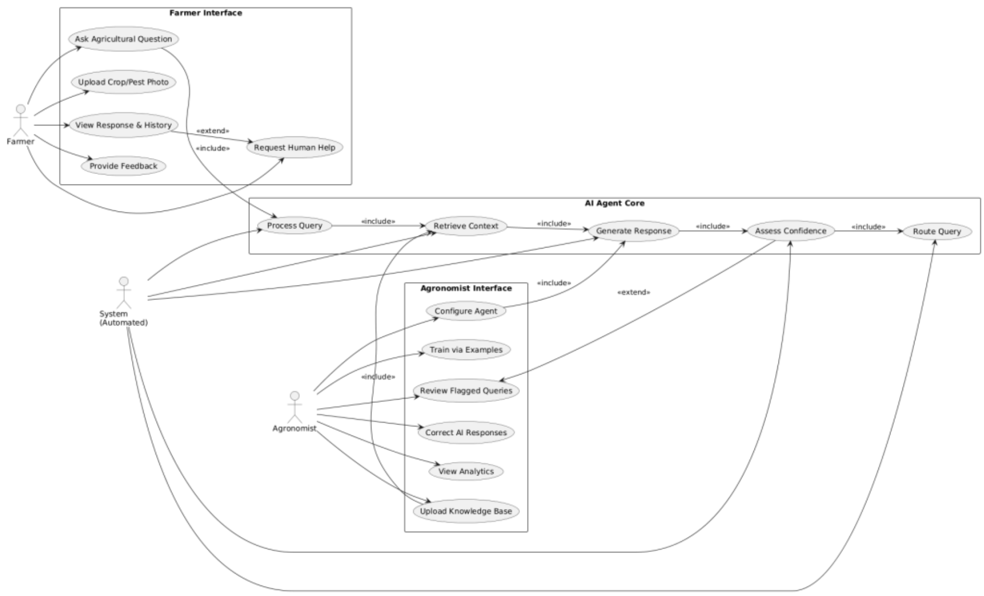

Figure 3: Main use case diagram showing the Farmer and Agronomist actors, their interactions with the AI Agent Core, and the Agronomist Interface.

Figure 3 shows the main use cases of the system. The two actors Farmer and Agronomist interact with distinct interface regions, connected through the AI Agent Core. The Farmer initiates queries, uploads photos, requests human help, and provides feedback. The Agronomist configures agents, uploads knowledge, reviews flagged queries, and corrects AI responses. The system automates the loop between these roles.

5. Implementation

5.1 Technology Stack

Table 2 lists the core technologies. pgvector was chosen over dedicated vector databases (Pinecone, Weaviate) because co-locating relational and vector data in Supabase removes cross-service latency and an additional billing dependency at this scale. SSE was chosen over WebSockets because token streaming is unidirectional the server pushes to the browser and SSE is simpler for that use case.

Table 2: Core Technology Stack

Component	**Technology**
Framework	Next.js App Router 16
Database	Supabase PostgreSQL 15 + pgvector
ORM	Prisma + PrismaPg
Authentication	NextAuth.js v5 (JWT, bcrypt)
LLM	Claude Sonnet 4.5 (Anthropic)
AI Framework	Mastra 1.12
Embedding Model	Voyage AI 3.5-lite (1024-dim)
Streaming	Server-Sent Events (SSE)
Deployment	Vercel + PWA (Serwist)

5.2 Knowledge Base Ingestion Pipeline

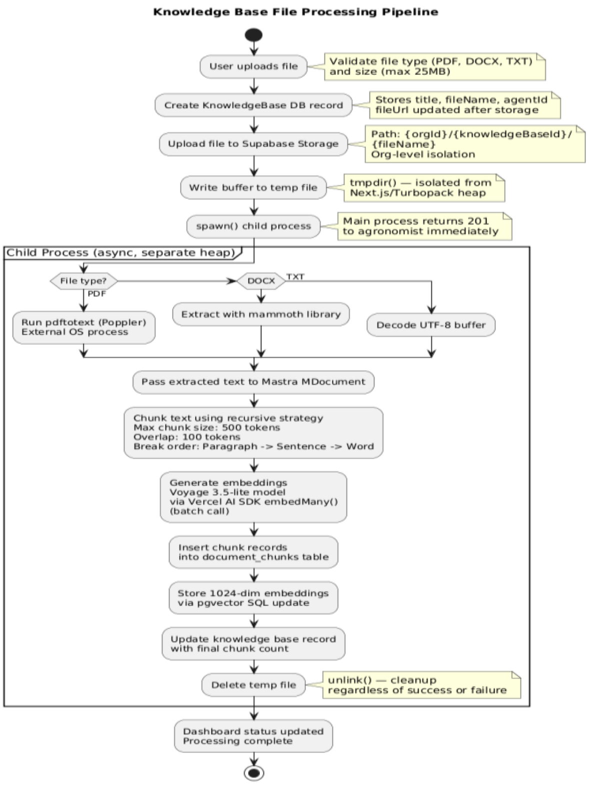

Figure 4: Knowledge base ingestion pipeline: document upload triggers an async child process that extracts text, chunks, embeds via Voyage 3.5-lite, and stores vectors in pgvector.

When an agronomist uploads a document, an asynchronous pipeline runs in a child process to protect the main server runtime. Figure 4 shows the pipeline flow. It: (1) extracts text using pdftotext for PDFs or mammoth for DOCX; (2) chunks text with Mastra's MDocument using a recursive strategy (500 tokens max, 100-token overlap); (3) embeds all chunks in a single embedMany() call to Voyage AI; (4) upserts 1024-dimensional vectors into pgvector with knowledgeBaseId metadata; and (5) writes the final chunk count back to the knowledge base record. A 164 KB, 11-page Farmitecture FAQ document produced 23 chunks, with a peak heap of 29 MB, in the child process.

5.3 Authentication and Authorisation

Three roles exist: FARMER, AGRONOMIST, and ADMIN. Role-based access control runs at three levels – route proxies, server-rendered page files, and API routes via a shared hasRole() utility. Passwords are bcrypt-hashed. User role and organisation ID are embedded in the JWT session token, so no server-side session store is required.

5.4 User Interfaces

The agronomist dashboard (Figure 5) provides live stat cards for active agents, knowledge base documents, and pending flags, plus a recent activity feed. Agents, knowledge bases, and flagged queries are managed from separate views within the same interface. The farmer chat interface (Figure 6) streams agent responses, token by token, in Markdown. Voice input transcribes speech to text before submission. Farmers can attach up to three images per message. GPS coordinates from the farmer's profile are automatically injected into the agent's runtime context, enabling location-aware advice and weather tool calls without the farmer having to state their location in each message.

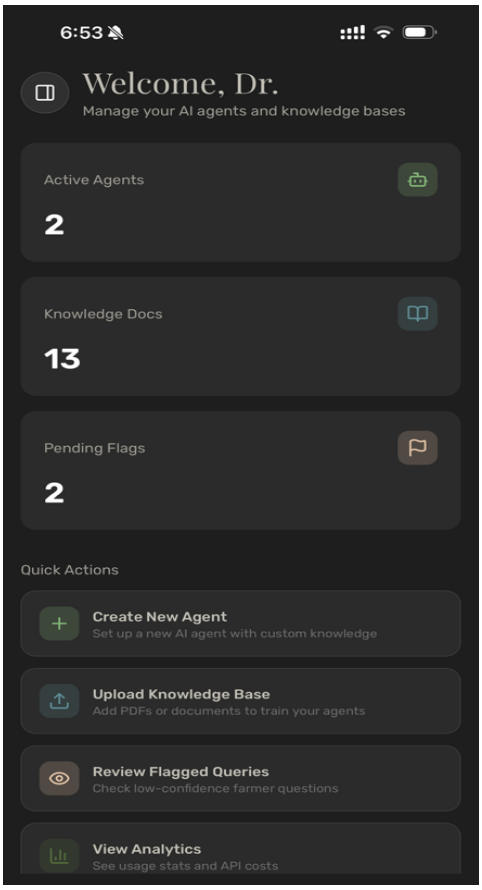

Figure 5: Agronomist dashboard showing live counts of active agents, knowledge base documents, and pending flagged queries, with a quick actions panel.

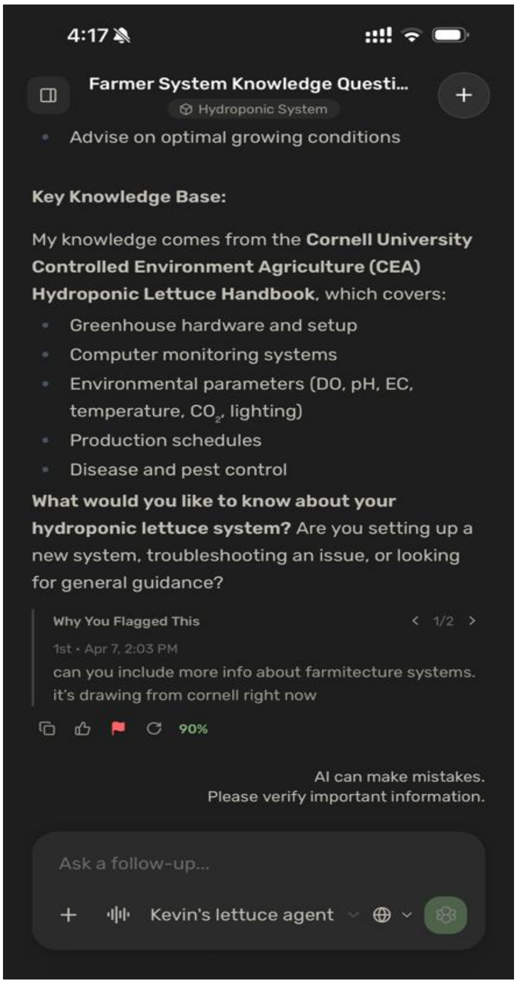

Figure 6: Farmer chat interface showing a streamed agent response with confidence score, message actions (copy, flag, retry), and the agent selector in the input bar.

6. Testing and Results

6.1 Component Testing

RAG pipeline components were tested in isolation before integration. Table 3 summarises the results. SSE streaming was verified via live chat; all five event types (user message, text, title, done, error) were confirmed in the browser's network tab. Role enforcement was tested with HTTP requests under mismatched roles: FARMER requests to agronomist endpoints returned HTTP 403, and unauthenticated dashboard access redirected to login with HTTP 302.

Table 3: RAG Pipeline Component Test Results

Component	Input	Result
PDF extraction	164 KB, 11 pages	8,840 chars extracted
Chunking	8,840 chars	23 chunks, <1 ms
Embedding	23 chunks	23 x 1024-dim vectors
pgvector storage	23 embeddings	Cosine index intact
RAG retrieval	Test query	Similarity 0.65-0.88

6.2 System-Level Functional Testing

All 15 must-have functional requirements passed manual end-to-end testing the one partial: FR-A2, the conversational Q&A pair training interface. Agronomists configure agent persona via the system prompt field as a workaround, but no dedicated Q&A ingestion interface was built within the semester. Table 4 shows the results of non-functional requirement verification.

Table 4: Non-Functional Requirement Verification

Req.	Requirement	Result
NFR-U1	Response <= 5 s	Time-to-first-token < 5 s
NFR-U2	Farmer usable after 2-min tutorial	Confirmed (Khaleel)
NFR-U3	Agronomist setup <= 30 min	Done in < 10 min (Kevin)
NFR-S5	Data isolation per organisation	Cross-org queries return 403
NFR-S6	Secure auth + RBAC	JWT + bcrypt verified

6.3 User Testing

Testing involved Khaleel (Farmitecture farmer, mobile PWA) and Kevin (Farmitecture agronomist, desktop). Six tasks were run with Khaleel; five with Kevin.

Farmer (Khaleel). Within one minute of being handed the phone, Khaleel identified the app as "an AI for farming basically a farming chatbot" without any prompting. In a multi-turn memory task, he asked, "How often should I apply them?" without naming the prior subject; the agent resolved the reference correctly. His response: "Yeah, basically continued on the conversation." On a weather-aware watering query, his saved location pulled automatically, and the agent gave time- and humidity-specific advice. He said: "It gives specific advice the timing based on location, the weather, humidity, and all the things I need to know. It felt like real advice." The off-topic guardrail redirected a barbershop business plan request back to the farming category. The only friction: the send button icon was ambiguous.

Agronomist (Kevin). Kevin set up his agent with a system prompt scoped to hydroponic lettuce, a response style descriptor, and a confidence threshold, mostly without help. He uploaded two documents; chunk counts updated on the dashboard within seconds. He confirmed the agent's Bok Choy answer matched how he would explain it to a client, and noticed it was more concise than the general advisor agent – an effect he linked to his tighter system prompt. On the DDT safety test, the agent refused, cited the Ghana ban and health risks, offered alternatives, and asked a clarifying question about the specific pest. Kevin: "It's good it actually gives an alternative." During flagged query review, he verified one response and corrected another, recommending a 15-15-15 NPK. On overnight trust: "Yes I'd just check the flagged queries every morning. I'm comfortable letting it do my job in my absence for a while." He estimated 50-60% initial query automation, rising to 70-80% with regular knowledge base updates.

6.4 Research Objectives and Results

Table 5 maps each research objective to the pilot's outputs.

Table 5: Research Objectives Mapped to Results

Objective	Result
OBJ1: Bottlenecks	Agronomists spend up to 10 hrs/week on repetitive queries; 50% of these are estimated to be automatable.
OBJ2: Agronomist acceptance	All 3 survey agronomists are comfortable with AI training; Kevin set up unaided in <10 min.
OBJ3: Farmer acceptance	83% of farmers are interested; Khaleel used the app intuitively without instruction
OBJ4: Scalability	Time-to-first-token < 5 s; Kevin estimated 50-60% automation

7. Discussion

7.1 Cultivate as an Agentic Advisory System

Placing Cultivate on the Bonifacio et al. continuum [15], it operates at L2-L3. It integrates multimodal perception (text, images, GPS, weather), specialised retrieval via RAG, multi-turn conversational memory, and bounded action selection through confidence-gated routing. It does not yet close the full loop in the sense of autonomous long-horizon planning or self-directed knowledge gap identification those remain future work. But it moves clearly beyond L1 decision support into coordinated, state-aware advisory.

The gap between Cultivate and L3 closure is primarily in adaptation over time. The current system flags low-confidence responses and surfaces them for agronomist correction, but does not yet automatically incorporate those corrections back into the knowledge base. FR-S8 (learning loop) was classified as Won't have " in the semester scope. Building this out would move the system meaningfully toward the feedback-integrated adaptation that Gavai & Heringa [16] identify as the defining property of agentic food systems AI.

7.2 Decentralised Knowledge and Institutional Trust

The pilot confirmed both sides of the decentralisation argument. Kevin's willingness to let the agent run overnight was direct evidence that agronomist-controlled knowledge produces agronomist trust. He framed it clearly: "If I upload my knowledge, the AI is representing me. So it would have to be a solid AI agent." The trust is conditional on the system's reliability and is not granted by default.

Khaleel's conditional interest in flagging he would use it only if the agronomist responded quickly shows that farmer trust in the workflow depends on agronomist responsiveness, not just technical correctness. Gavai & Heringa [16] describe this as the trust cultivation problem: trust cannot be legislated through design; it builds through iterative transparency and demonstrated reliability. The flagged query mechanism is the transparency layer. Its value depends on agronomists using it consistently.

This also surfaces a governance question the current system does not fully answer: who verifies that an agronomist's uploaded knowledge is accurate? In a centralised platform, the provider does. In Cultivate, no one does the agronomist is trusted by design. For a low-stakes deployment like Farmitecture's urban vegetable customers, this is probably acceptable. For higher-stakes contexts such as commercial crop advisory or disease outbreak response it would need structured peer review or external validation at upload time.

7.3 Open Challenges

Several challenges identified by Bonifacio et al. [15] apply directly to Cultivate's next phase. Grounding under partial observability: the current system has no mechanism for detecting when the knowledge base is insufficient for a query the confidence scorer is a proxy, not a ground-truth signal. Evaluation beyond predictive accuracy: the pilot measured usability and user acceptance, but not longitudinal decision quality. Did the advice Kevin's agent gave actually help Khaleel's plants? That question requires a longer deployment and outcome tracking. Multi-objective decision-making: the current agent optimises for answering the question asked; it does not reason over competing agronomic objectives like water use, cost, and yield simultaneously. That would require moving toward the orchestration layer described by Bonifacio et al.

7.4 Limitations

The pilot sample one agronomist, one farmer is insufficient for statistical generalisation. The confidence scoring module is heuristic-based and may produce miscalibrated flags. WhatsApp integration is not yet built, despite WhatsApp being the primary channel farmers use to contact agronomists. Load testing at the 1,000 simultaneous conversation target has not been run. The system has not been evaluated for longer-term behavioural impact on farming outcomes.

8. Conclusions and Future Work

Cultivate demonstrates that the 1:1,500 extension officer bottleneck is addressable through an agentic design that returns knowledge ownership to the agronomist. An agronomist set up a working, knowledge-grounded agent in under ten minutes. A farmer used it without instruction. The agronomist estimated it could handle more than half his query load not by replacing him, but by covering the routine so he can focus on the complex.

The system sits at L2-L3 on the agentic continuum: contextually aware, memory-equipped, and human-supervised, but not yet self-adapting or long-horizon planning. Closing those gaps is the main technical agenda for the next phase.

The immediate priorities: a 3-4 week structured pilot with 10-15 Farmitecture farmers to collect the outcome data OBJ4 needs; WhatsApp Business API integration to reach farmers on the channel they already use; a Q&A pair ingestion interface for agronomists who find system prompt writing opaque; and a learning loop that automatically incorporates agronomist corrections into the knowledge base. Medium-term, a more principled confidence scoring method – moving from heuristics to a learned signal would make the flagging workflow more reliable. Long-term, an agent marketplace where agronomists publish trained agents for other organisations would create a sustainable model for investing in high-quality knowledge bases. It could do more for extension service capacity than any hiring target alone.

The question Gavai & Heringa [16] leave open is whether agentic AI in food systems can remain answerable to the people whose lives its decisions affect. Cultivate's design agronomist as owner, farmer as user, platform as infrastructure is one answer to that question in the context of extension services. Whether it holds at scale is what the next pilot will show.

References

[1] E. Punsu, "Insufficient extension officers negatively affecting agriculture," The Business & Financial Times, Jul. 2023.

[2] Ghana Districts, "Ghana Districts: A repository of all Local Assemblies in Ghana," Ghanadistricts.com, 2017.

[3] Ministry of Food & Agriculture, "Agricultural e-Extension Services in Ghana: Strategy and Plan, 2022-2030," 2022.

[4] C. Yada Lagba, "Ghana: Agriculture Minister Decries Shortage of Extension Officer," allAfrica.com, Aug. 2025.

[5] Pro Listings, "Urban Farming in Ghana: A Guide for Expats to Grow Food and Connect with Community," Listings Pro GH, May 2025.

[6] Digital Green, "Farmer.Chat," Digitalgreen.org, 2023.

[7] N. Singh et al., "Farmer.Chat: Scaling AI-Powered Agricultural Services for Smallholder Farmers," arXiv (Cornell University), Sep. 2024. doi: 10.48550/arxiv.2409.08916.

[8] J. Dutra, "How AI is Revolutionizing Smallholder Farming in Africa with FarmerAI," Opportunity International, 2025.

[9] F. Jomo and A. Sguazzin, "The AI Revolution Comes for Farmers Growing a Third of Our Food," Bloomberg/Opportunity International, Jun. 2024.

[10] CGIAR, "PlantVillage Nuru: Pest and disease monitoring using AI," CGIAR Platform for Big Data in Agriculture, Jun. 2018.

[11] Farmerline, "Farmerline's Darli AI Recognized on TIME's List of the Best Inventions of 2024," Farmerline.co, 2024.

[12] S. Bhagwat, Principles of Building AI Agents, 2nd ed. Independently Published, Feb. 2025.

[13] Tieme Ndo, "Why Do Farmers Need the Service of an Extension Officer?," Tiemendo.com, Oct. 2020.

[14] United Nations, "Goal 2: Zero Hunger," UN Sustainable Development, 2025.

[15] A. Bonifacio, I. Iele, G. Romoli, M. Tortora, and P. Soda, "Agentic AI in Agriculture: Towards Automatic Farming Systems," SSRN, 2026. doi: 10.2139/ssrn.6716658.

[16] A. K. Gavai and J. Heringa, "Agentic artificial intelligence in food science: From automation to adaptation," Food and Humanity, vol. 6, p. 101132, 2026. doi: 10.1016/j.foohum.2026.101132.