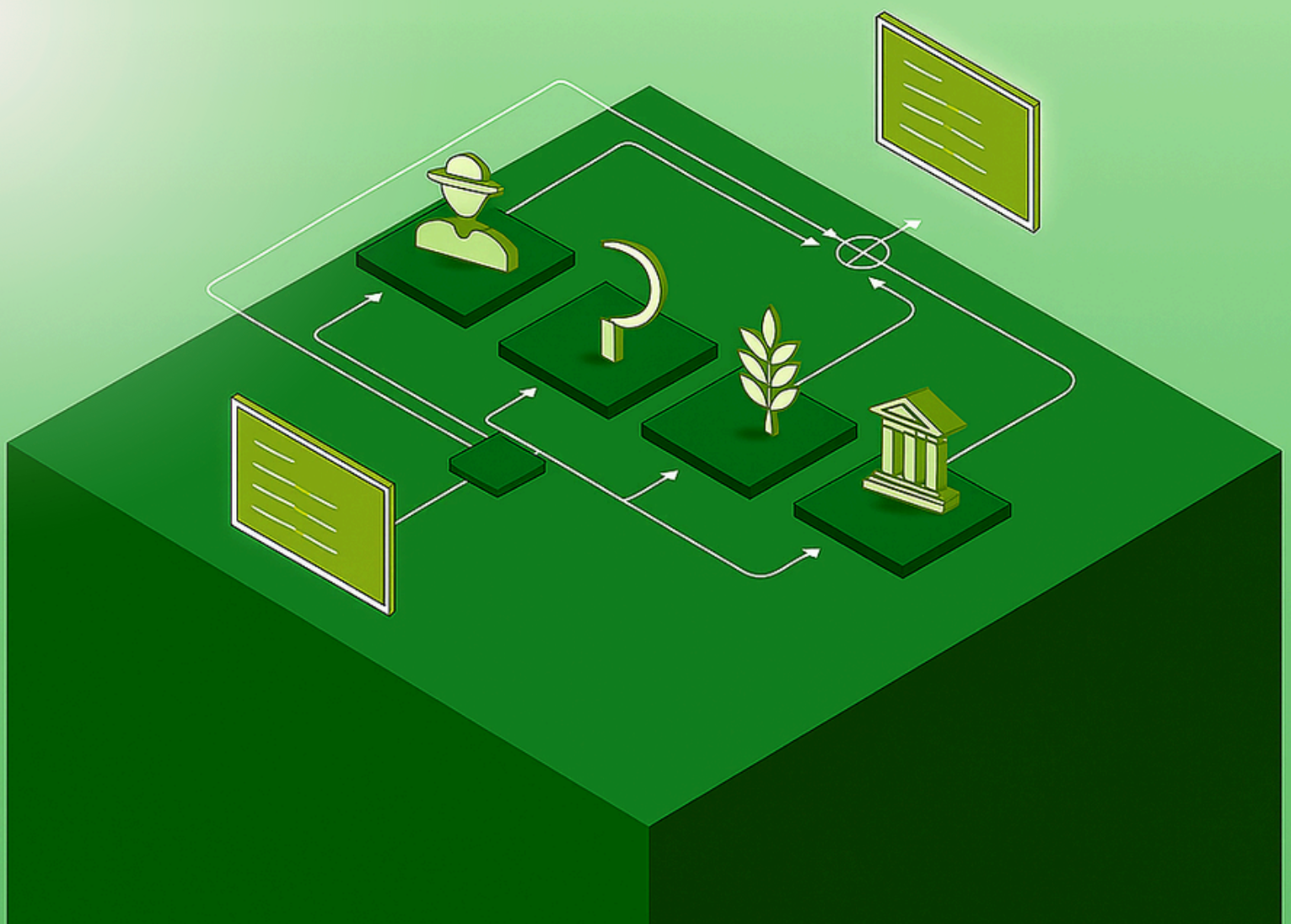
Capital One

Team DataSultants

AGENTIC AI-POWERED HYPERLOCAL AGRICULTURAL ADVISOR

KRISHAK SANGHI



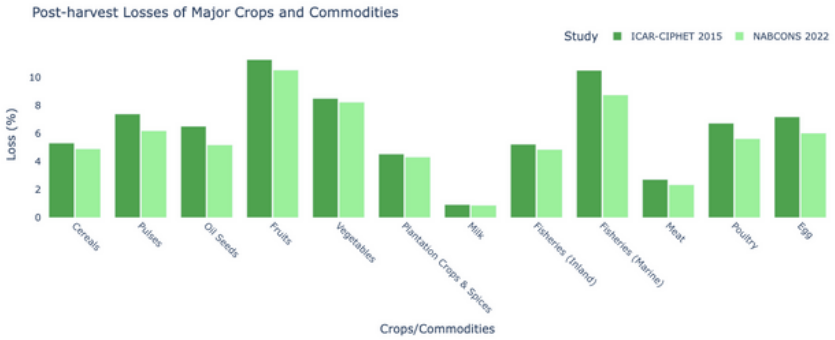
Introduction

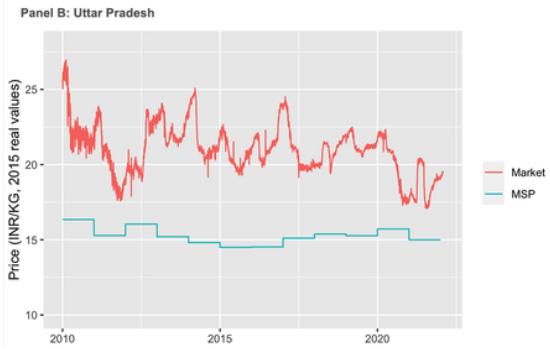
Theme Name:
Exploring and Building Agentic AI Solutions for a High-Impact Area of Society: Agriculture

Impact Analysis

India's agriculture, contributing ~**18**% to GDP and employing ~**42**% of the workforce, is dominated by smallholder farmers who face price volatility, high post-harvest losses (~15–20%, up to **35**% for perishables), and climate shocks causing 30–40% yield losses annually. Fragmented mandi, weather, and storage data, poor connectivity, and language barriers further trap farmers in low profitability and high wastage.

An **agentic AI-powered hyperlocal agricultural advisor** can address these gaps by unifying real-time mandi trends, weather forecasts, storage costs, and demand signals into farmer-friendly, explainable insights. With India's agritech market projected at US**$30–35**B by 2025 and growing rural internet penetration, offline-first, regional-language AI solutions—leveraging e-NAM, government portals, and agritech APIs—can enable smarter price timing, reduce spoilage, and enhance resilience, significantly boosting farmer profitability and efficiency across the value chain.





Solution Overview

Krishak Sanghi is an agricultural intelligence platform—an agentic AI ecosystem that provides hyperlocal, actionable advice to farmers. At its core is an Advisory Agent that interprets natural-language queries, connects them to the right tools, and delivers responses in clear, farmer-friendly language. It combines retrieval-augmented generation over a Qdrant vector store with live data from Agmarknet for mandi prices, Open-Meteo for weather, and web APIs for market and policy updates. MongoDB enables personalized recommendations, while FastAPI microservices ensure scalability and resilience.

The platform supports **use cases** like **pest detection and remedy**, **policy planner** and **Subsidy Calculation**. It also supports live weather and agricultural news along with **dynamic alerting system** for farmers through notifications, scheduling daily and long term tasks. It has a dynamic **web and mobile app User Interface**, with **multilingual support** and **voice input/output**. Krishak Sanghi also supports **offline connect**, through **SMS service**, providing instant assistance even in **low/no connectivity areas**. With its extensible design, Krishak Sanghi is built to evolve—ready for satellite imagery, financial modelling, and more—making it a trusted, continuously improving advisor for India's agricultural value chain.

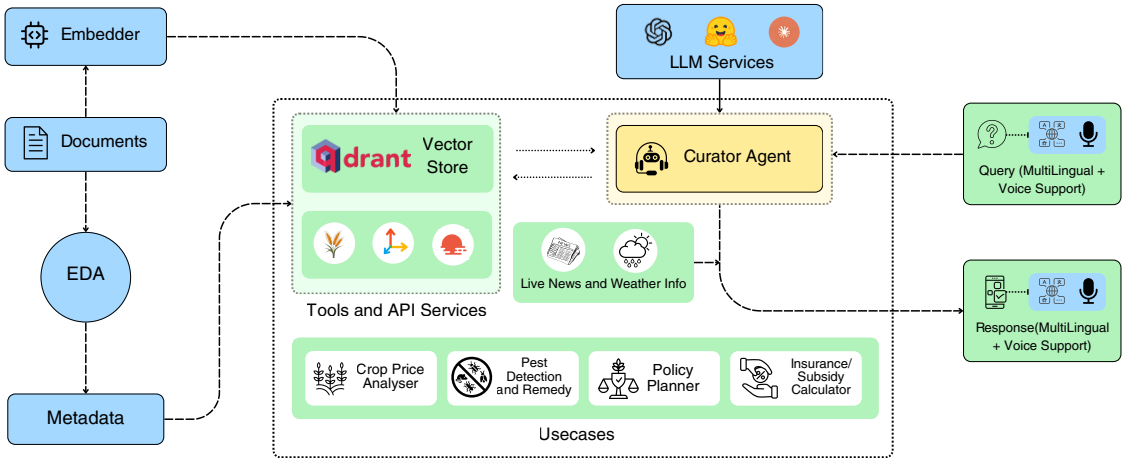


Fig 1: Architecture of Krishak Sanghi

Curator Agent

The Curator Agent functions as an agentic system, orchestrating agricultural intelligence through a combination of specialized tools and conversational reasoning. It builds its responses on a structured pipeline, where each query moves through stages of understanding, tool invocation, and synthesis. Equipped with domain-specific modules like pest detection, policy fetcher, weather tool, crop prices, and more — it transforms raw data into actionable insights. Operating as the bridge between complex systems and the farmer, the Curator doesn't just deliver static answers; it dynamically plans, retrieves, and contextualizes information, finally presenting it in the warm, human-like tone of a trusted neighbor.

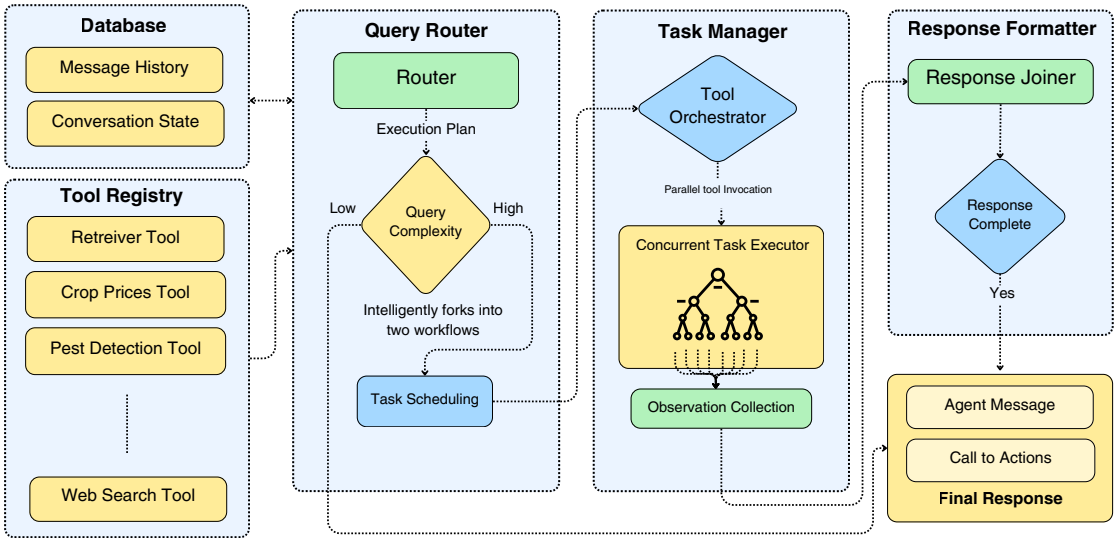


Fig 2: Curator Agent Architecture

Query Router

We employ a query routing layer that classifies farmer queries into domains like weather, prices, or pest management, and directs them to relevant tools. By reformulating inputs for tool compatibility and synthesizing outputs into clear advice, it ensures responses are precise and farmer-friendly.

Task Manager

Task management ensures tool calls are executed efficiently and reliably, bridging the gap between routing decisions and results. By parsing plans, and coordinating execution with concurrency control, it transforms high-level plans into structured outcomes reintegrated into the agent's workflow.

Response Formatter

Response Formatter transforms raw tool outputs into clear, coherent, and context-aware responses. It adapts complex technical results into simple, farmer-friendly language while preserving precision, ensuring reliability, and maintaining broad accessibility across diverse user needs.

Methods Integrated:

- **Query reformulation:** Queries are reformulated into tool-friendly inputs using farmer context like location, land, and budget to ensure intent is preserved.
- **Context chaining:** The system links current queries with past interactions and farmer profile details to provide more consistent and personalized advice.

Implementation Details:

- **Concurrent orchestration:** Multiple tools are executed in parallel with semaphore-based control, maximizing efficiency while preventing overload.
- **Result-aware integration:** Tool outputs are normalized, tracked, and reinjected into conversation history, enabling later steps to build seamlessly on prior results.

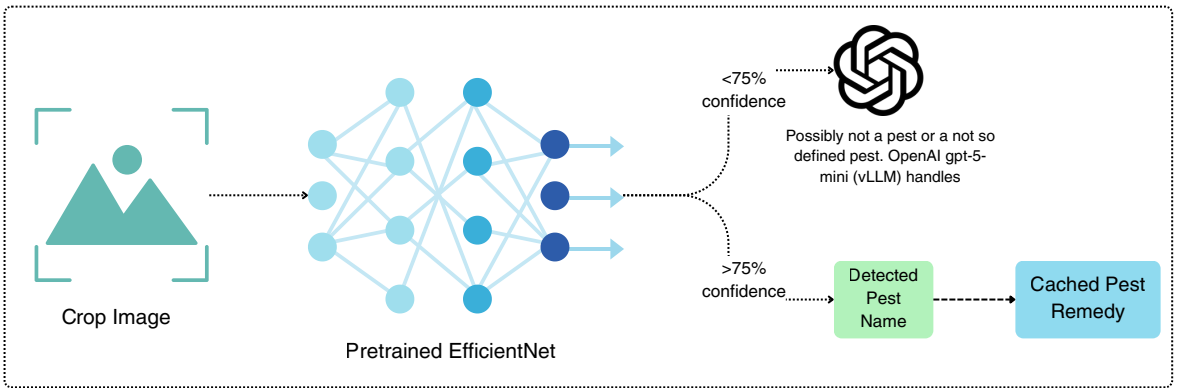
Schema:

- **Agent Message:** A short concise summary comprising of information gathered from all the tool calls and the agent's knowledge
- **CTAs:** Small actionable call to actions, for for guiding farmers toward the next best step
- **Tasks:** Agent-managed alerts, reminders, and follow-ups within the conversation.

Use-cases and Tooling

Pest Detection and Pesticide Recommendation

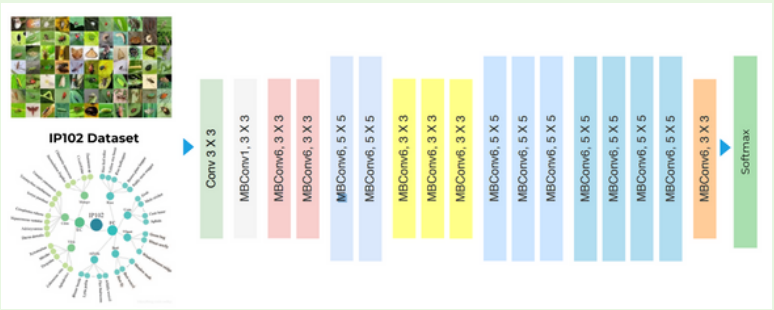
Beyond general-purpose LLMs and VLMs, the platform incorporates domain-specific fine-tuning to deliver higher accuracy. A dedicated pest detection service uses an **EfficientNet model** fine-tuned on the **IP102** dataset to identify **crop diseases** and pest infestations from field images with high precision. This specialized vision model enables the system to instantly recognize threats, retrieve targeted treatment guides, and factor in local market and weather data to recommend cost-effective intervention strategies.



Pest Detection Pipeline

The farmer uploads the image of a crop (with suspected pest). The pipeline passes it through the pre-trained **EfficientNet** model and gets the maximum probablity class (of 102 classes). If the maximum probablity is **<75%**, it is assumed that the image doesn't likely contain a pest or is likely a unknown pest. Such images are passed to OpenAI based **gpt-5-mini vLLM** for further analysis. If the model is confident about the pest (>75%), the remedy of the pest is fetched from a cache (pre-defined), for reduced **latency**.

Model and Training Details



Pest & Diseases Detection Tool is powered by our in-house fine-tuned EfficientNet-B0 model, specifically trained on the IP102 dataset to achieve **87%** accuracy across **102** pest categories. Unlike generic VLM-based inference, this model has been purpose-built and optimized for agricultural pest recognition, enabling precise and efficient identification directly from user-provided images for rapid, field-level diagnosis.

The IP102 (v1.1) provides a rigorous foundation, with over **75k images** covering a diverse, long-tailed distribution of pest species. Its challenging characteristics, subtle inter-class differences, high intra-class variability, and diverse acquisition conditions necessitate robust fine-grained classification strategies, emphasizing macro-averaged metrics, per-class recall, and hierarchical label utilization.

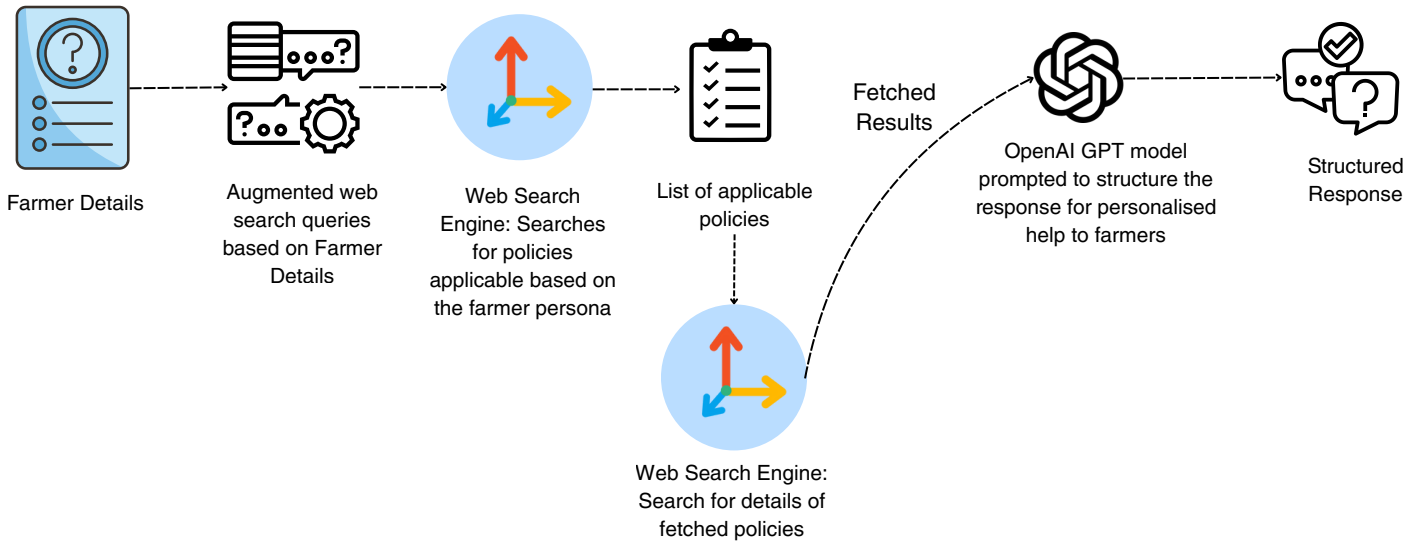
EfficientNet-B0 was selected for its balanced parameter-efficiency tradeoff, preserving crucial morphological and texture cues while remaining computationally practical. Its **MBConv** blocks with squeeze-and-excitation and Swish activations enhance feature selectivity, vital for discriminating between visually similar pests. The model was progressively fine-tuned, replacing the ImageNet head with a task-specific classifier to adapt effectively to the agricultural domain.

Training leveraged **AdamW** with cosine annealing, mixed precision, and gradient clipping, alongside imbalance-aware techniques such as class-aware sampling, focal/balanced loss, and texture-preserving augmentations (**RandAugment, colour jitter, MixUp/CutMix**). Progressive unfreezing helped retain general visual priors while specializing for pest recognition, and refinement for rare but high-impact pest species.

Use-cases and Tooling

Policy Planner

Assists the farmer in identifying current and new policies and *yojanas* by the government that can suit their needs. It also gives a detailed plan of how the farmer can avail of the scheme most efficiently. Uses a **web-search tool** to actively search for new policies and discover existing ones based on the needs and details of the farmer.



Policy Planning Pipeline

The farmer is prompted to enter details like **location**, **farm-size**, **capital** and policies availed. Custom **augmented search queries** are then created based on these details. These queries are then passed to the **Tavily** based web search engine (tool calling), which then thoroughly seraches for the details of policies all across the web, to find the most relevant policies based on the details of the farmer. It generates the list of top 3 most relevant policies (which is not availed by the farmer previously). This list then goes to another **Web Search engine** (powered by Tavily Search, tool calling), which finds the details of each of the policies from filtered set of most trusted websites. These results are then passed to **OpenAI GPT** agent, which extracts the relevant information, structures it in form of a detailed plan generating steps of policy avail, documents required, contact details and expected benefit, enabling farmer to make an informed decision and not miss on government schemes and policies.

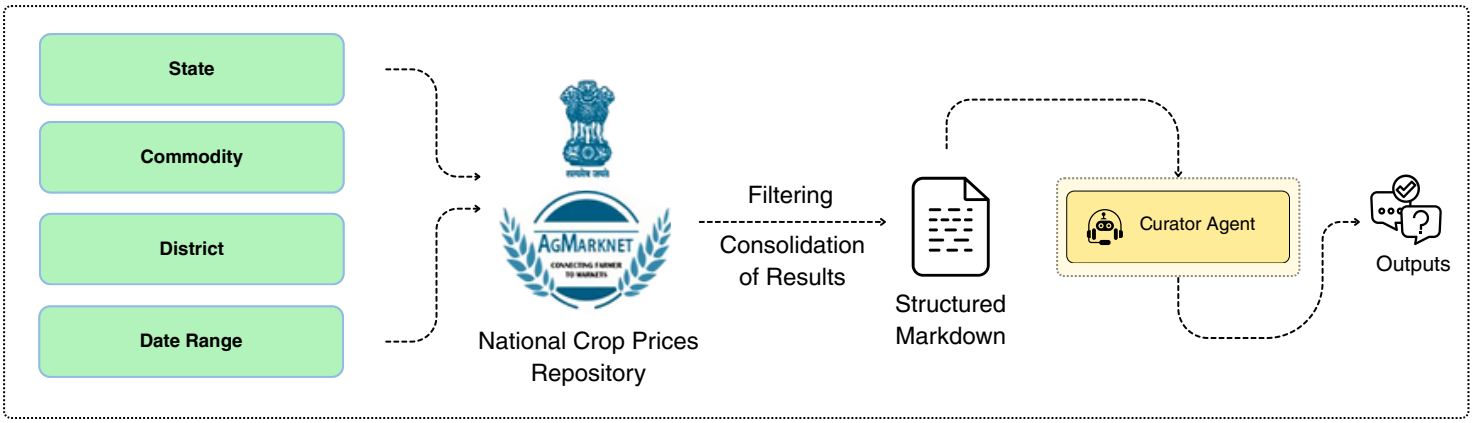
Key Features and Importance

- **Farmer Profile Personalisation:** Uses farmer details (location, farm size, crops, farming type, income, land ownership) to generate context-specific search queries and filter schemes.
- **Smart Query Generation:** Creates targeted search queries based on location, crop type, and farm size, ensuring hyperlocal and crop-specific relevance.
- **Reduced Latency and Extensible Architecture:** Supports asynchronous execution (asyncio) for handling multiple queries concurrently, with built-in rate limiting to prevent API overload. Modular design makes it easy to plug in different search tools, policy analyzers, or AI models in the future without rewriting the orchestrator.
- **Fetching from trusted sites:** Used website filtering, to show results fetched from trusted websites only (government websites, wikipedia, etc) and not from any website.
- **Reranked Results based on query:** The results of the web search are reranked based on the query generated, to get the most relevant schemes only and remove the schemes not useful to the farmer.

Use-cases and Tooling

Crop Price Analyser

Fetches and provides the latest crop prices from **Agmarknet** for farmers. It allows searching by commodity, state, and optionally district, giving detailed price data over a specified date range. Uses **fuzzy matching** to handle input variations and helps farmers track price trends across regions, enabling informed decisions on selling and crop planning efficiently.



Crop Price Analyser Pipeline

The farmer is prompted to enter details like **commodity type**, l**ocation (state/district)**, and **date range** for price analysis. Custom data retrieval queries are then created based on these details. These queries are then passed to the government agricultural market data system which constructs targeted requests for official market pricing information. The **CropPriceFetcher** then systematically retrieves comprehensive market data from across the agricultural pricing network, using **fuzzy matching algorithms** to find the most relevant commodity and location data based on the farmer's input details. It generates detailed price information for the specified crop and region. This data then goes to the data processing engine (powered by structured data extraction), which processes the raw market data from trusted **government agricultural websites, like Agmarknet**. These results are then passed to the fuzzy matching system, which extracts the relevant price information, structures it in the form of detailed market analytics including historical prices, market trends, district-wise comparisons, and pricing patterns, enabling farmers to make informed decisions about when and where to sell their crops for optimal returns.

Insurance Subsidy Calculator

Calculates and manages crop insurance premium subsidies under **PMFBY** for farmers. The agent collects farmer details such as crop type, location, and coverage, computes the total premium, and determines the farmer's contribution versus the government subsidy. It tracks subsidy distribution between Central and State Governments, monitors data uploaded to the National Crop Insurance Portal, and follows the structured release process of upfront and subsequent installments. In case of crop loss, it oversees claim assessment, processing, and final subsidy adjustment, enabling farmers to access timely and accurate insurance support efficiently.

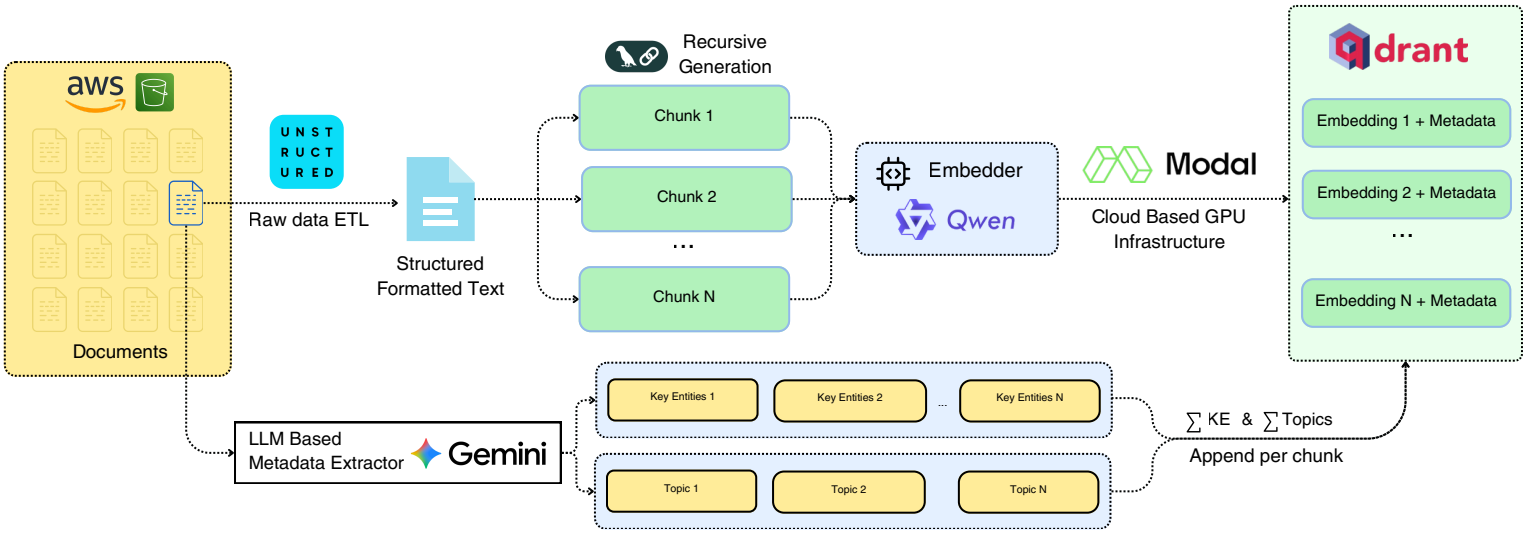
Subsidy Calculation Pipeline

Based on our research, we found that the insurance premium for farmers depends on **crop type, farm size, location**, and **cultivation season**. Using these details, our system calculates premium rates and subsidy amounts internally. The Curator Agent determines final subsidy values considering crop season (**2% for Kharif, 1.5% for Rabi**), farm size, location-specific risk factors, and sum insured. The system then generates a comprehensive subsidy breakdown, including farmer premium share, government subsidy amounts (split equally between Central and State Governments), upfront payment estimates, and claim settlement provisions, enabling farmers to understand their insurance coverage costs and government support without sending any personal data externally.

RAG Pipeline and Architecture

Architecture Details

Our **dynamic metadata-enhanced RAG** pipeline ingests raw agricultural documents into AWS, where the **Unstructured API** converts them into structured text while preserving layout. A Gemini-based metadata extractor identifies key entities and topics, stored alongside the text. Using **Recursive Character Text Splitting** (2000 chars, 400 overlap), we generate context-rich chunks, enriched for positional and semantic context. Each chunk is embedded with **Qwen-0.6B** on Modal's GPU infrastructure, combining vector embeddings with authoritative metadata. Stored in Qdrant, this fusion enables metadata-first retrieval, instantly narrowing candidates by crop, region, year, or policy before semantic scoring, delivering precise, domain-specific answers in milliseconds.



Metadata Extraction

Metadata enrichment in this RAG architecture goes beyond simple chunk storage by attaching structured, LLM-extracted signals such as key entities, topics, and contextual tags to each embedding. Using **Gemini** as the metadata extractor, the system identifies people, organizations, places, and **domain-specific concepts**, along with thematic topics that **summarize the chunk's content**. This enriched metadata is appended to the **embeddings** in Qdrant, enabling hybrid retrieval that combines semantic similarity with **symbolic filtering**. As a result, queries can be answered not only by meaning but also with precision filters—for example, retrieving chunks specifically about a certain company, policy, or event. This layer of enrichment adds transparency, improves search accuracy, and enhances explainability by showing why a particular document was retrieved.

Recursive Chunking

Recursive Chunking in this architecture ensures that large documents are broken down into manageable, **semantically coherent** pieces without losing context. Instead of splitting text arbitrarily, the recursive approach looks for **natural breakpoints** like sections, paragraphs, or sentences, and creates chunks of optimal size for downstream processing. **Overlapping windows** can be applied so that important context near chunk boundaries is preserved. This method allows the system to handle very long documents while maintaining continuity and relevance in retrieval.

Embedding Generation

Embedding Generation takes each chunk and converts it into a **high-dimensional** vector representation using the **Qwen3** embedder, running on scalable cloud GPUs provided by **Modal**. These embeddings capture the semantic meaning of the text, allowing similar chunks to be retrieved based on meaning rather than just keywords. By pairing **embeddings with metadata**, the system not only enables powerful similarity search but also ensures that retrieval is both accurate and contextually rich.

User Interface & Key Features

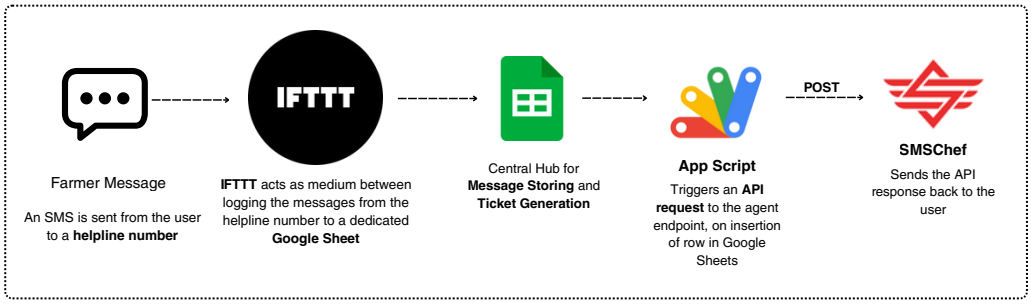
Offline SMS Access

Farmers living in **low-connectivity regions** can also access Krishak Sanghi through a simple **SMS-based service**, ensuring that digital inclusion is not limited to those with smartphones or stable internet connections. We maintain a **dedicated helpline number** that remains active round the clock for offline queries. In such cases, the farmer sends an SMS to our number, and the incoming message is automatically logged into a **Google Sheet via IFTTT**, creating a structured digital record.

Once an SMS is received, a unique support ticket is generated for that farmer. Any further queries from the same number are placed in a message queue, ensuring that no request is lost or overlooked. The queued message is then forwarded to our **AWS EC2 instance**, where it is processed by the Krishak Sanghi system. After processing, the response is routed back to the farmer in the form of an SMS using a reliable third-party service, **SMSChef**.

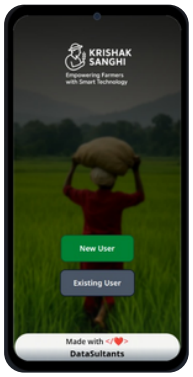
This workflow guarantees that even farmers without internet access can benefit from Krishak Sanghi's intelligence. Through SMS, they can ask questions in their local language, receive hyperlocal insights on prices, weather, and policies, and get advisory support in real time. The system is designed to be lightweight, farmer-friendly, and resilient, bridging the digital divide in rural India.

By integrating SMS with **cloud processing**, Krishak Sanghi ensures last-mile connectivity, empowering farmers in remote areas to make data-driven decisions and access the same level of support as their internet-enabled counterparts.

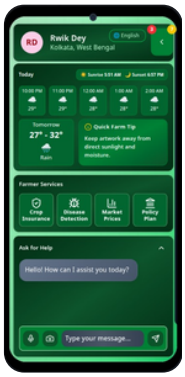


Mobile App

For easy accessibility, we have developed both a **mobile app** and a **web app**, ensuring that Krishak Sanghi can be accessed seamlessly across devices. Farmers can conveniently use the mobile app on smartphones for quick, **on-the-go access**, while traders and FPOs can leverage the web app for more detailed dashboards and analytics. The mobile app also works in low-connectivity areas, storing queries offline and syncing automatically when the internet is available, ensuring uninterrupted support. Notifications, reminders, and updates are integrated so that users never miss critical information about mandi prices, weather alerts, or government policies.



User Login Window

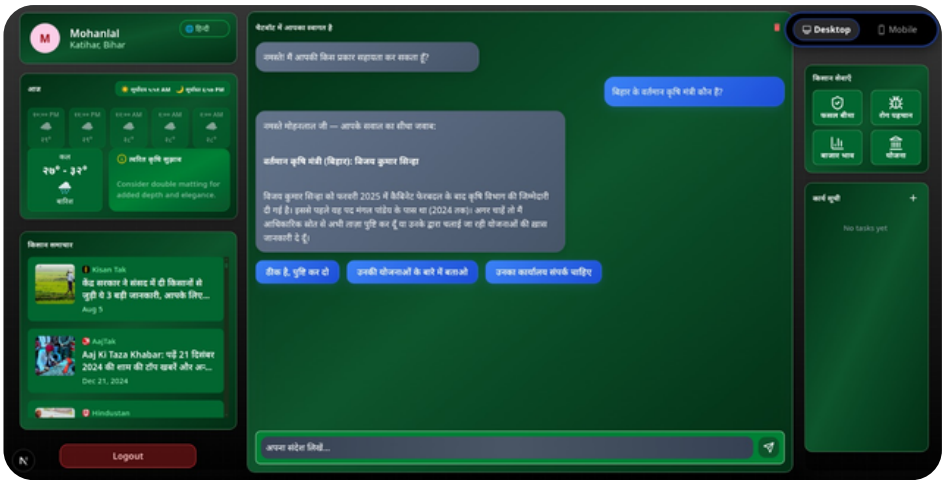


User Interface - Mobile App

User Interface & Key Features

MultiLingual Support

In India, most of the farmers are not comfortable with English as a Language and they are more fluent in their regional language. In order to cater these needs, our solution supports **10** regional languages of India, making it easy for farmers across all parts of India to communicate to and use Krishak Sanghi efficiently.



Dynamic To-Do List

The application has a **dynamically updating to-do list** based on user's conversation to chatbot. Whenever the user asks the chatbot to create a **task/ reminder** about, say a policy update or document verification, it gets added directly to the To-Do list with a checkbox. Once the task is completed by the user, they can check the box and the task gets removed from the To-Do List. The To-Do list gets dynamically updated by the **context** of user's chat.

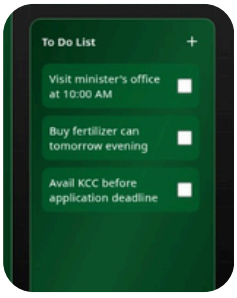
News based on farmer's Persona

Real-time news based on farmer's details and chat-context is fetched and shown on the News-Board. Information like Location, Land Size, Annual Income and crop is recorded during user-registration. Now this information is used to fetch the most relevant news across India through most trusted sites through filtering. After filtering, the outputs are reranked based on user's context. This News Board enables farmers to be updated to the latest news about their farming practices in and around their location, crop information and details on prices and subsidies.

Weather Board

Current Weather details and forecasts are displayed in the weather board based on the location of the user. Based on the weather, a useful tip is also displayed on the board based on farmer's soil, crop, land and weather conditions. OpenMeteo API is used to effectively capture the weather details and the Curator Agent further uses these details to generate the tip. This doesn't only provide the user with the weather update, but also an interpretation on what to and what not to do in his farming activies based on the current weather.

Dynamic News Board



To-Do List

News Board

Future Scope

Multimodal (Image + Voice) Support

Currently Krishak Sanghi supports textual input and image input (only for pest detection). In future work we aim to improve the capablities in multimodal front by integrating image inputs and outputs. Additionally, the roadmap includes voice-assisted conversational support where farmers can speak to the platform like a virtual advisor and receive responses in their own dialect. We also envision visual outputs such as annotated images and infographics to help explain advisory content in a more farmer-friendly way. This multimodal approach will not only reduce barriers to adoption but also ensure inclusivity across literacy levels, linguistic diversity, and varying digital familiarity.

Offline Call Access

In the future, Krishak Sanghi will expand beyond SMS by enabling farmers to **call a dedicated helpline number**. Their queries will be recorded, transcribed using speech-to-text, and processed by our system on **AWS EC2**. Responses will then be delivered back either as a **voice call in the farmer's local dialect** or as an SMS for easy reference.

This feature will be especially useful for farmers with limited literacy, allowing them to interact naturally through speech. By supporting voice-based advisory in regional languages, Krishak Sanghi will act as a virtual on-call assistant, ensuring inclusivity and last-mile connectivity for even the most digitally excluded farmers.

Smart IoT enabled Advisory

In the future, Krishak Sanghi can integrate with **IoT sensors** and smart farm devices to deliver **real-time, hyperlocal advisory**. Sensors deployed in fields can capture data on **soil moisture**, **pH, nutrient levels**, **temperature**, **humidity**, and **pest activity**, which will then be transmitted to the Krishak Sanghi system for analysis. By combining this sensor data with weather forecasts and crop models, the platform can generate automated, actionable alerts—such as when to irrigate, apply fertilizer, or prepare for pest control—directly to the farmer's phone via SMS, app notification, or voice call.

This **IoT-driven** layer will make advisory proactive instead of reactive, shifting from generic recommendations to farm-specific precision insights. Over time, farmers can also receive predictive analytics like yield estimates, disease outbreak warnings, or irrigation scheduling optimized for water savings. Such integration will not only improve productivity but also enhance sustainability by reducing input costs and resource wastage.
By connecting Krishak Sanghi with affordable IoT kits or community-shared devices, even smallholder farmers in rural areas can access the benefits of precision agriculture at scale, bridging the technology gap and unlocking data-driven farming in India.

END