← Back to Current Research

TR-2025-02 · PUBLISHED AUGUST 10, 2025 · VIRTUAL VAKIL AI LABS

Virtual Vakil: A Multi-Agent Legal Intelligence System for WhatsApp-Native Delivery in India

Virtual Vakil introduces a multi-agent legal intelligence system that serves Indian users through WhatsApp. Eight role-specialised agents — research, drafting, judgment, rapid help, case tracking, bare-act lookup, current affairs, and argument simulation — share a common Indian-legal foundation and cooperate behind a single conversational surface. The system is built around the constraints of the messenger it runs in, so that legal guidance reaches users where they already are.

Multi-Agent | Legal AI | Indian Judiciary | 8 Agents | WhatsApp

ABSTRACT

Abstract

Indian users arrive at a legal-AI system with heterogeneous needs. In a single conversation a user may ask what a section of the criminal code says, ask the system to draft a complaint under it, ask whether a pending FIR has any court listing, and ask for the most recent judgment on a similar fact pattern. Each of these is a different sub-task with a different notion of a correct answer, and a single general-purpose prompt does not serve all of them well.

We describe **Virtual Vakil** as deployed in August 2025: a multi-agent legal intelligence system in which eight role-specialised agents — Chanakya, Vidhi-Vetta, Nyaydhish, Sahaayak, Munshi, Pustakalya, Gidh, and Vad-Vivad — each carry a distinct persona, prompt discipline, and trigger surface, while sharing a common Indian-legal foundation. A lightweight keyword-aware classifier routes each inbound WhatsApp message to the right specialist before any generation is performed, so that routing itself remains inexpensive and the specialists can be iterated on independently.

This paper documents the agent hierarchy, the routing protocol, the WhatsApp-native delivery constraints — the 24-hour customer-care window, the approved-template regime for outbound broadcasts, and the plain-text formatting that renders correctly inside a chat bubble — and the production feedback loop from practising advocates. The system is in active production use today, with Indian litigants, junior advocates, and law students as its primary users.

AUTHORS & AFFILIATIONS

Who wrote this paper

Mahir Gupta

PRINCIPAL INVESTIGATOR

Virtual Vakil AI Labs · Advocate, Delhi High Court

Piyush Gupta

CO-INVESTIGATOR

Virtual Vakil AI Labs

Virtual Vakil AI Labs

ENGINEERING AND REVIEWERS

New Delhi, India

1. INTRODUCTION

Why Indian legal AI needs specialisation

The Indian judicial system carries a pendency of more than fifty million cases across its trial courts, High Courts, and the Supreme Court. The population it serves spans dozens of languages and hundreds of dialects, with Hindi and code-mixed Hinglish dominating everyday conversation and English dominating the statutory and courtroom register. A consumer-grade legal assistant for India cannot confine itself to a single language, a single register, or a single type of query, and it cannot assume that users will come to a website. It must meet them where they already transact, which in India means WhatsApp.

Early attempts at Indian legal AI treated the problem as a single-model task: one large language model answers every legal query. In practice, what users arrive with is rarely a single legal question. A citizen handed a traffic challan wants to know whether the challan is valid, the statutory section under which it was issued, the settlement route, a draft reply if the challan is contested, and to be notified when any related hearing is listed. Each of these is a different sub-task with a different notion of correctness.

Trying to handle this breadth with one general prompt produces a well-known failure mode: the model performs acceptably in the middle of the distribution but degrades at the tails, and because each sub-task has its own tail, the aggregate reliability is poor. A multi-agent decomposition, in which each sub-task is handled by an agent whose persona and prompt discipline have been tuned for that specific output style, lets each surface be iterated on independently while keeping the conversational front-end coherent for the user.

The final design constraint is delivery. WhatsApp is the only channel that reliably reaches Indian users across devices, languages, and bandwidth regimes. This imposes its own rules — the 24-hour customer-care window, the approved-template regime for outbound broadcasts, and the plain-text formatting that renders correctly inside a

chat bubble — and the system described here is built around those rules as primary constraints rather than as downstream packaging.

2. SYSTEM ARCHITECTURE

The Eight Agents

Each agent is a distinct persona with its own prompt discipline and trigger surface. The agents are role-specialised on a shared Indian-legal foundation, and are surfaced to the user as named specialists rather than as a single anonymous assistant.

Chanakya

RESEARCH & PRECEDENT ANALYSIS

The research specialist. Handles legal research, precedent retrieval, and strategic case planning. Takes queries that concern case law, citation trails, and courtroom strategy and returns structured, advocate-readable analysis.

Vidhi-Vetta

DOCUMENT DRAFTING

The drafting specialist. Produces first-draft legal documents — notices, petitions, affidavits, agreements — from structured fact inputs supplied by the user, using templates that match Indian procedural conventions.

Nyaydhish

JUDGMENT & CASE MERIT ANALYSIS

The judgment specialist. Analyses case merit, interprets judgments, and explains judicial reasoning so that a litigant or junior advocate can see how a matter is likely to be

received by a court.

Sahaayak

QUICK LEGAL HELP

The rapid-help specialist. Delivers fast, plain-language answers for urgent queries — fundamental rights, procedural routes, arrest and bail guidance, emergency contact points.

Munshi

CASE TRACKING & HEARINGS

The case-tracking specialist. Handles CNR lookup, hearing schedules, listing information, and filing deadlines, so that users and their counsel never miss a court date.

Pustakalya

BARE-ACT LIBRARY

The bare-act specialist. Retrieves exact section text from the Indian statutory corpus together with historical amendments and cross-references, so that users work from authoritative statutory language rather than paraphrase.

Gidh

LEGAL NEWS & AMENDMENTS

The current-affairs specialist. Surfaces recent judgments and gazette notifications from Indian legal newsrooms and statutory gazettes so that users see the law as it stands, not

as it was.

Vad-Vivad

ARGUMENT SIMULATION

The argumentation specialist. Generates prosecution-versus-defence argument pairs, cross-examination prompts, and rebuttals for a given fact pattern, useful for moot preparation and courtroom rehearsal.

3. SMART INTENT ROUTING

How a message reaches the right specialist

An inbound WhatsApp message is first labelled with an intent — greeting, legal question, traffic challan, case-tracking query, document help, emergency legal matter, legal news, cybercrime, consultation booking, or conversational exchange. Labelling is performed by a lightweight keyword-aware classifier that draws on a curated vocabulary of legal terms, Hindi and Hinglish phrasings, and example queries gathered from production traffic. Most messages are labelled decisively by this stage, and only genuinely ambiguous ones are escalated to a deeper classification step.

Once the intent is known, the labelled message is delivered to the appropriate specialist. Messages concerning drafting — notices, petitions, affidavits — go to Vidhi-Vetta. Messages concerning statutory text — sections, acts, articles — go to Pustakalya. Messages concerning current judgments and gazette notifications go to Gidh. Where no specialist-specific trigger matches, the message is handled by Chanakya as the default research agent. Structured flows (traffic challan settlement, case lookup, consultation booking) bypass the specialist layer and are served by dedicated interactive flows.

The effect of this two-stage design is that the full generation stack is invoked only when the message genuinely requires it. Routing itself is cheap, specialists are chosen deterministically where the evidence is strong, and the user experiences a single conversational front-end whose specialist behind the scenes is always the one best suited to the task.

4. WHATSAPP-NATIVE DELIVERY

Delivery is part of the system, not after it

WhatsApp imposes three constraints that shape how agents must generate output. First, the 24-hour customer-care window: free-form text from the business is only permitted within 24 hours of the user's last inbound message. Any notification outside that window — hearing alerts, payment receipts, challan reminders — must use a pre-approved Template with typed placeholders. Second, messages are rendered inside a chat bubble; Markdown headers, fenced code blocks, and triple-backtick formatting render as raw characters and are unreadable. Third, interactive flows for challan payment, document upload, and case lookup are served through a validated flow schema that must itself be maintained inside an approval pipeline.

Each agent's prompt discipline carries explicit formatting rules for WhatsApp: asterisks for bold, underscores for italics, bullet or numbered lists for steps, no Markdown headers, no code fences, and short paragraphs. The agent is also instructed to match the user's language — English, Hindi, or code-mixed Hinglish — and to address the user by name when known and with formal Sir or Ma'am otherwise. This conservative addressing rule was adopted after early casual-register experiments produced complaints from professional users.

Outbound notifications — challan-found alerts, slot-ready confirmations, payment invoices, receipt nudges — are delivered through a fixed catalogue of approved Templates maintained on the WhatsApp Business Platform. Agent-generated text is never used for out-of-window broadcasts. This separation keeps the specialist layer free to iterate on tone and content without entangling template-approval cycles.

5. KEY CONTRIBUTIONS

Research Highlights

1. Eight role-specialised agents, each with a distinct persona, prompt discipline, and trigger surface, together covering the sub-tasks that Indian users actually arrive with.

2. A lightweight keyword-aware intent classifier that routes every inbound WhatsApp message to the right specialist without incurring the cost of a full generation call on routing alone.

3. A WhatsApp-native delivery layer designed around the platform's 24-hour customer-care window, approved-template regime for outbound broadcasts, and the plain-text formatting that renders correctly inside a chat bubble.

4. Production-deployment constraints treated as primary design inputs rather than packaging details: the system is shaped by the messenger it runs in, not retrofitted to it.

5. Evaluation by practising advocates on sampled production traffic, rather than by crowdsourced labels or synthetic benchmarks, which tends to over-report fluency as correctness on statutory material.

6. EVALUATION

Advocate-in-the-loop review on production traffic

Evaluation of a legal-AI system on crowdsourced labels is known to over-report accuracy — non-expert annotators mistake fluency for correctness, particularly on statutory citation. We evaluate agent outputs by sampling from production WhatsApp traffic and routing sampled threads to practising advocates on the team for review. Each sample is rated on four axes: statutory correctness (is the cited section the right section for the fact pattern), procedural correctness (is the suggested next step consistent with current Indian procedure), linguistic fidelity (does the response match the user's language, including Hinglish code-mix), and user-readiness (is the response directly usable, or does it require further lawyer interpretation).

The evaluation is deliberately qualitative at this stage. The domain does not admit the kind of clean held-out split that makes benchmark numbers meaningful: the statutory corpus itself is undergoing transition, with the Bharatiya Nyaya Sanhita and Bharatiya Nagarik Suraksha Sanhita progressively replacing the Indian Penal Code and the Code of Criminal Procedure. Qualitative advocate review is, under these conditions, the most informative signal we can collect, and it is the signal that drives agent-prompt iteration.

7. LIMITATIONS

Scope and ongoing directions

⚠ Language coverage is expanding.

The specialist prompts are tuned for English, Hindi, and Hinglish. Broader coverage for other Indian languages, particularly in southern and eastern regions, is an active direction of work; the specialist architecture does not itself limit language expansion.

⚠ Case-type coverage is broadening.

The specialists are strongest on criminal procedure, consumer disputes, family matters, and traffic enforcement, which reflect the volume of queries we see in production. Specialised commercial and constitutional matters benefit from the same architecture but continue to be refined with advocate feedback.

⚠ Agent boundaries are refined by hand.

The trigger surface that selects a specialist is curated from production traffic and refined as new phrasings emerge. Occasional mis-routes — a news request that really wants a legal explanation, a drafting request that is really a precedent search — are surfaced by the advocate-review loop and corrected at the trigger level.

⚠ Human advocate oversight remains essential.

On contested matters the specialists are intended to accelerate drafting and triage, not to replace the professional judgment of a practising advocate. Every user-facing response is framed as a starting point for human review rather than as final legal advice.

8. CONCLUSION

What this paper establishes

The system described here establishes three design commitments. First, routing precedes generation: a lightweight classifier chooses the specialist persona before any generation is performed. Second, delivery is a first-class constraint: specialist prompts, interactive flows, and outbound templates are designed against WhatsApp's policies as primary inputs, not as a packaging step after the model is trained. Third, evaluation is by practising advocates on sampled production traffic, not by crowdsourced labels or synthetic benchmarks.

The multi-agent decomposition presented here is the foundation that Virtual Vakil's subsequent research — on persistent case comprehension, conversation continuity, and India-hosted inference — is built upon. The April 2026 successor paper describes the advances that sit on top of this foundation.

HOW TO CITE

Citation

```
@techreport{gupta2025marl,
  title        = {Virtual Vakil: A Multi-Agent Legal Intelligence System
                  for WhatsApp-Native Delivery in India},
  author       = {Gupta, Mahir and Gupta, Piyush and {Virtual Vakil AI Labs}},
  institution  = {Virtual Vakil AI Labs},
  year         = {2025},
  month        = {August},
  number       = {TR-2025-02},
  url          = {https://virtualvakil.com/research-2025},
}
```

Tap the **Cite** button at the top of the page to copy this BibTeX to your clipboard.

REFERENCES

References

[1] Ouyang, L. et al. (2022). Training language models to follow instructions with human feedback. NeurIPS 2022.

[2] Park, J. S. et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023.

[3] Wu, Q. et al. (2023). AutoGen: Enabling Next-Gen Applications via Multi-Agent Conversation. arXiv:2308.08155.

[4] Supreme Court of India. (2024). National Judicial Data Grid: Pendency statistics, accessed August 2025.

[5] Ministry of Law and Justice, Government of India. (2023). The Bharatiya Nyaya Sanhita, 2023; The Bharatiya Nagarik Suraksha Sanhita, 2023.

[6] Ministry of Electronics and Information Technology, Government of India. (2023). The Digital Personal Data Protection Act, 2023.

Read the latest research

The April 2026 successor paper introduces VIM-1 — Virtual Vakil's proprietary, India-hosted legal AI — along with persistent case comprehension, conversation continuity across days, and locally-solved court-record ingestion.

Read VIM-1 Paper (2026)