Abhishek Nandakumar

AI Software Engineer

Originally from Chennai, India, now based in the US building AI systems that solve real problems. Over four years, I've designed multi-agent orchestration platforms with LangGraph and Bedrock, architected serverless enterprise applications on AWS, and led the modernization of legacy mainframe codebases using Claude. This portfolio covers nine projects and 350,000+ lines of production code. Open to relocation anywhere in the world where AI takes me.

abhismail998@gmail.com | +1 (602) 820-2950 (USA) | +91 9444970728 (India)

Tempe, AZ, USA | Chennai, Tamil Nadu, India | linkedin.com/in/abhin1998

github.com/Abhishekn1947 | aboutabhi.com

9	350K+	12+	155+	2,983
Projects	Lines of Code	AI Agents	API Endpoints	Test Cases

EDUCATION

M.S. Information Technology, Arizona State University (GPA: 4.0/4.0)

B.E. Computer Science, Sri Venkateswara College of Engineering, Anna University

CERTIFICATIONS

AWS Certified Cloud Practitioner | DASM | CSPO | Salesforce Administrator

Table of Contents

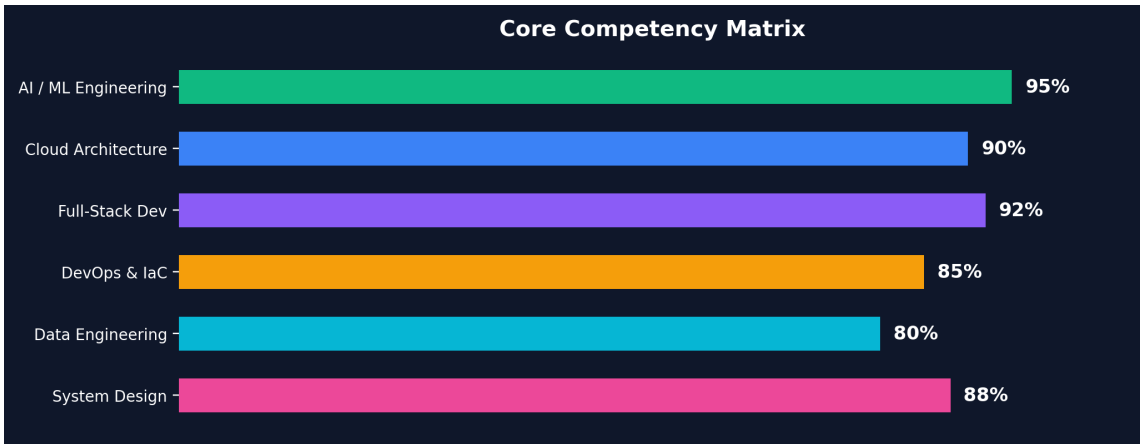
1 Executive Summary

What if AI could do more than chat? Over four years, that question turned into 350,000+ lines of production code across enterprise platforms, prediction engines, mainframe modernization tools, and multi-agent systems.

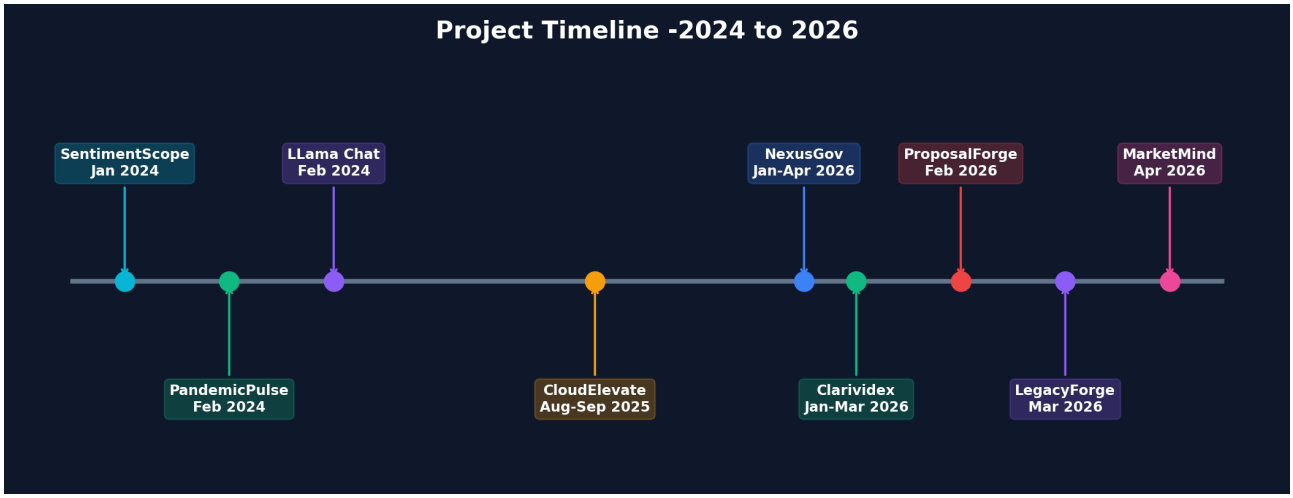
This portfolio walks through that journey. Nine projects, each one harder than the last, from early LLM experiments to AI agents that convert decades-old COBOL with 95.6% confidence.

> **Note on Demos**
> Several projects were built as internal enterprise applications. Live demos are available upon request, but public URLs cannot be shared.

Core Competency Matrix



Project Timeline



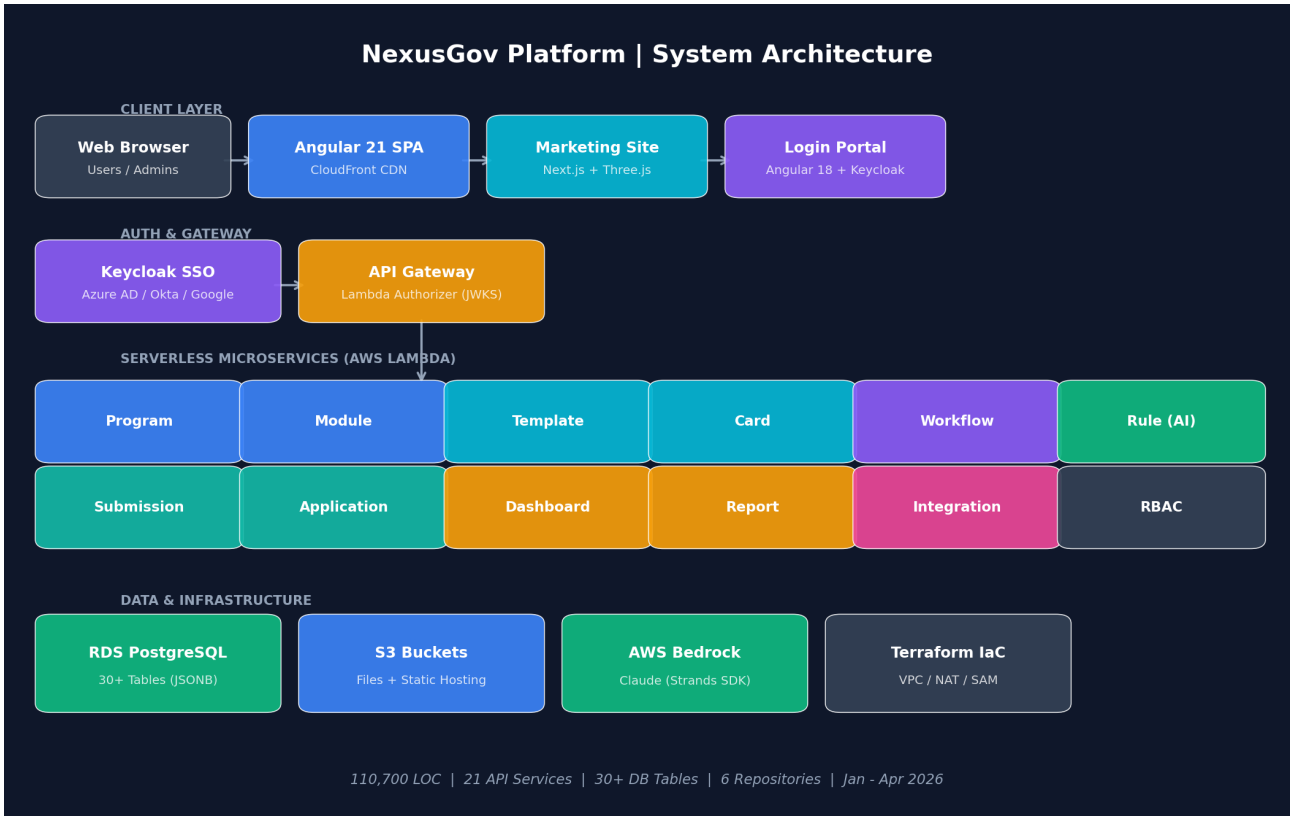
2 NexusGov | Enterprise Government Platform

Government agencies needed a way to build custom business applications without writing code. I designed NexusGov -a no-code platform where administrators drag-and-drop programs, workflows, and dashboards into existence. The crown jewel is the AI Rule Engine: business rules written in plain English, evaluated by Claude on AWS Bedrock, with deterministic tool functions to guarantee accuracy. Today it powers a Nevada State engagement across 6 microservices and 110K lines of code.

PaaS (No-Code Platform) | Client: Nevada State | Aditi LLC | Dec 2025 - Mar 2026 | Internal (demo available)

110,700	21	30+	4 months
Lines of Code	API Services	DB Tables	Duration

System Architecture



Key Features

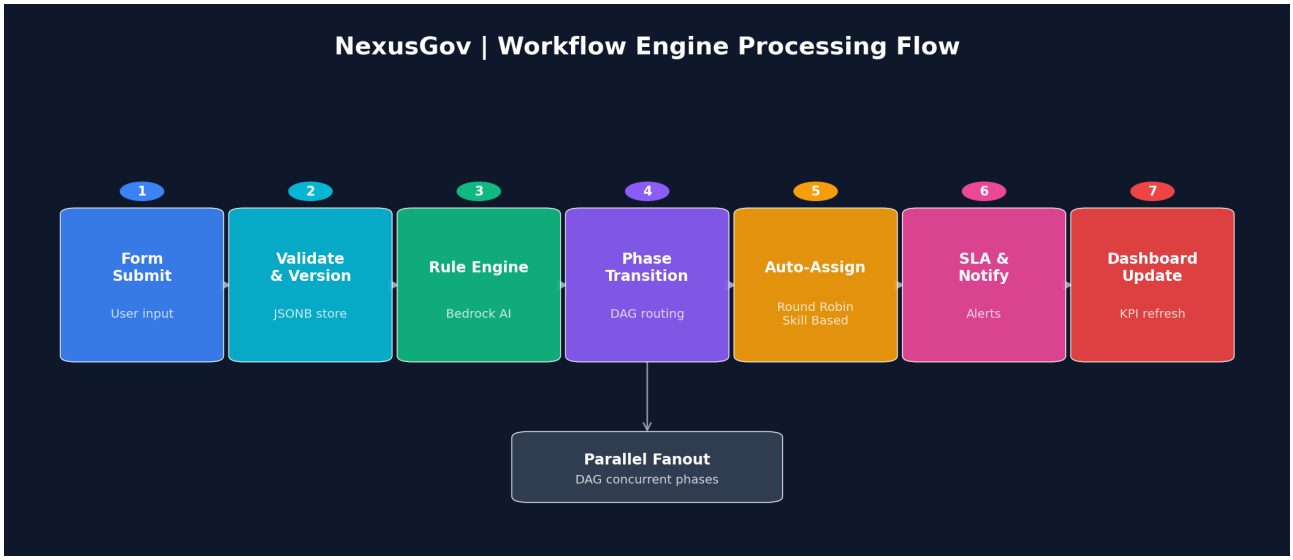
- No-Code Program/Module/Template/Card Builder with 15+ widget types
- DAG Workflow Engine with phases, transitions, fanout, SLA tracking, auto-assignment
- AI Rule Engine | NL rules evaluated by AWS Bedrock Claude (Strands SDK)
- Integration Hub | REST/SOAP/Webhook with OpenAPI/WSDL import

- Dashboard Builder -14 KPI types, 7 chart types, role-based overrides
- Report Engine | Parameterized queries, scheduled CSV/PDF/Excel export
- Keycloak SSO with Azure AD, Okta, Google federation and PKCE
- RBAC | Resource-level + builder permissions, superadmin bypass

Tech Stack

Layer	Technology	Purpose
Backend	Python 3.11, FastAPI, Mangum	Serverless per-Lambda microservices
Frontend	Angular 21, Material, Mermaid	SPA with workflow visualization
Database	PostgreSQL (RDS), Liquibase	JSONB storage with migrations
Auth	Keycloak (RS256 JWKS)	Multi-realm SSO federation
AI	AWS Bedrock Claude, Strands SDK	NL rule evaluation
IaC	Terraform, AWS SAM	VPC, RDS, Lambda, CloudFront

Process Flow



Entity Relationship Diagram

Multi-Agent Orchestration

The Rule Engine employs AWS Bedrock Claude (Haiku) via the Strands Agents SDK. Rules are written in natural language and evaluated by the LLM agent with deterministic tool functions (compare_values, calculate_age) to prevent hallucination.

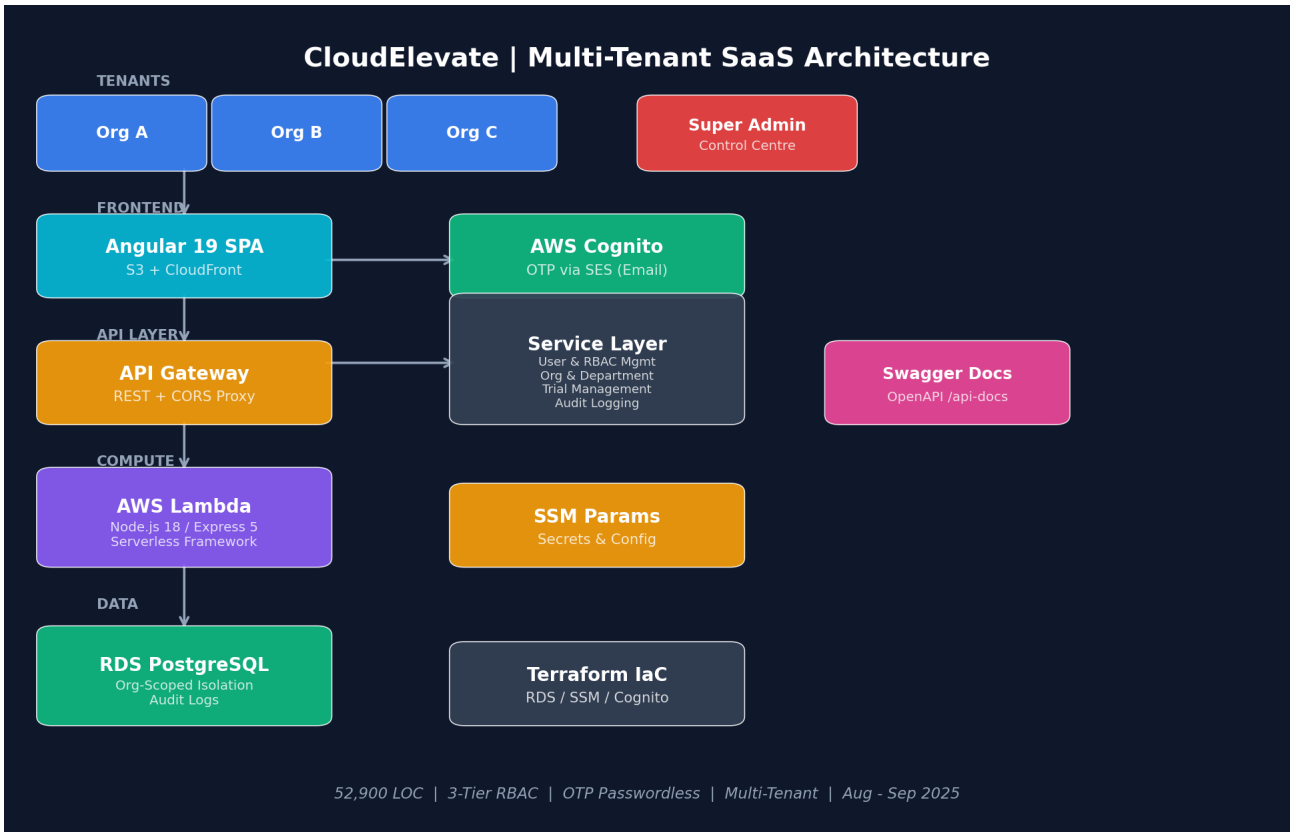
3 CloudElevate | Multi-Tenant SaaS Platform

Enterprise clients needed isolated cloud environments without sharing data. CloudElevate solved this with true multi-tenant isolation -every organization gets its own data scope, and admins log in with just an email (OTP via Cognito, no passwords). I built the entire platform in 5 weeks: a 3-tier RBAC system, trial management, and a super admin control centre for cross-org visibility.

SaaS (Multi-Tenant Cloud Platform) | Aug - Sep 2025 | Internal (demo available)

52,900	3	5 weeks
Lines of Code	RBAC Tiers	Duration

System Architecture



Key Features

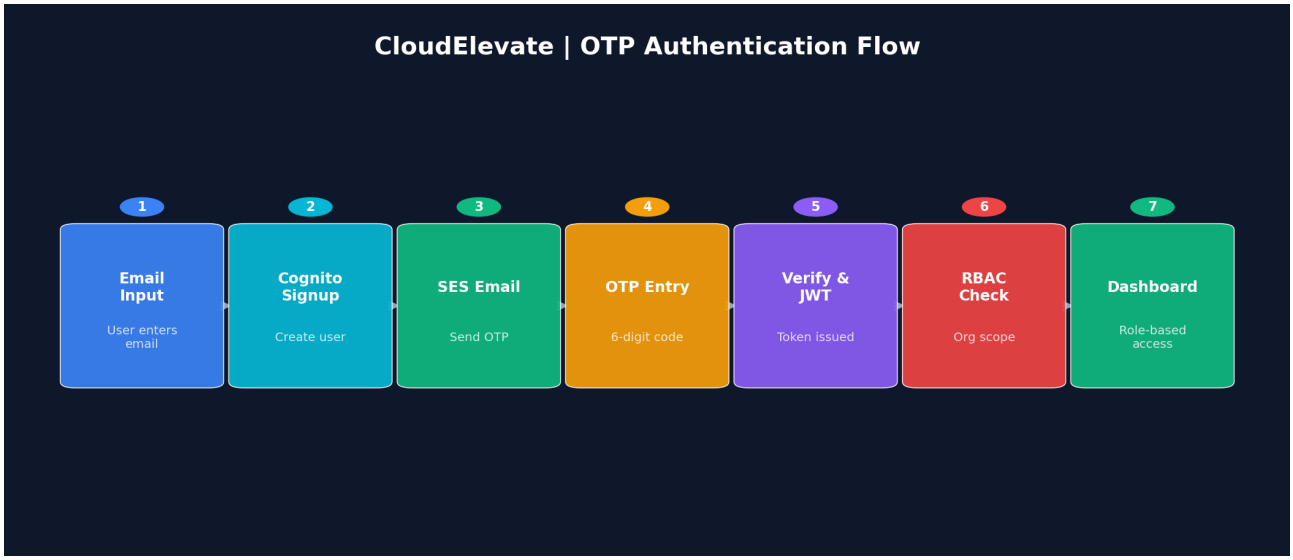
- Multi-Tenant Organization System with domain-based data isolation
- OTP Passwordless Auth | Cognito + SES email-only login flow
- 3-Tier RBAC (Super Admin > Admin > User) with middleware enforcement
- Trial Management with expiration tracking and extension workflows
- Comprehensive Audit Logging with IP, user agent, old/new values

- Swagger OpenAPI documentation at /api-docs

Tech Stack

Layer	Technology	Purpose
Backend	Node.js 18, Express 5, TypeScript	REST API server
Frontend	Angular 17, Angular Material	Super Admin control panel
Database	PostgreSQL (RDS)	Org-scoped relational store
Auth	AWS Cognito + SES	OTP passwordless
Deploy	Serverless Framework, Lambda	API Gateway proxy
IaC	Terraform	RDS, SSM, Cognito

Process Flow



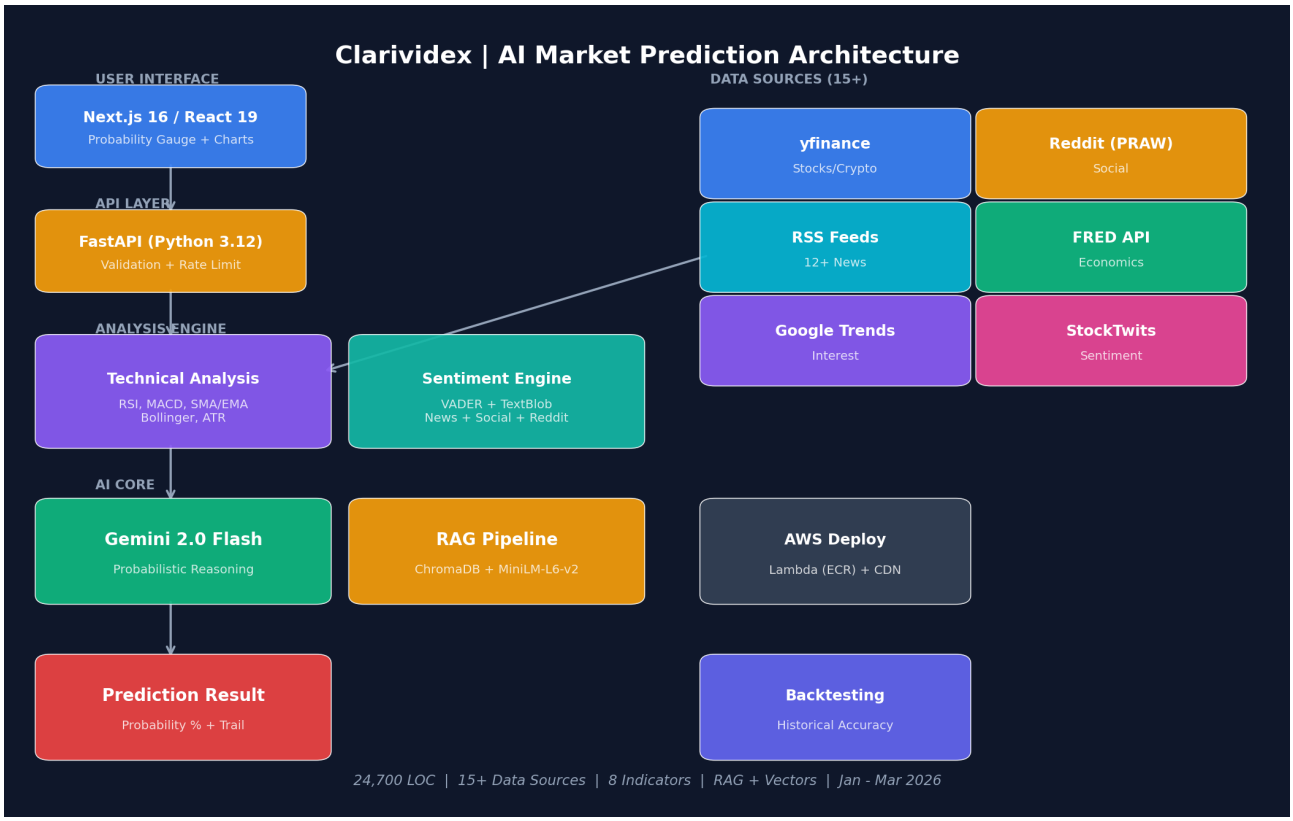
4 Clarividex | AI Market Prediction Engine

Can AI predict stock movements? I built Clarividex to find out. It pulls from 15+ real-time sources -yfinance, Reddit, RSS news feeds, FRED economic data -runs technical analysis (RSI, MACD, Bollinger Bands) and NLP sentiment scoring, then feeds everything into Gemini 2.0 Flash for probabilistic reasoning. Every prediction comes with a full decision trail so you can see exactly why the AI thinks what it thinks.

AI Web Application (Serverless) | Jan - Mar 2026 | Personal Project

24,700	15+	8	5 weeks
Lines of Code	Data Sources	Indicators	Duration

System Architecture



Key Features

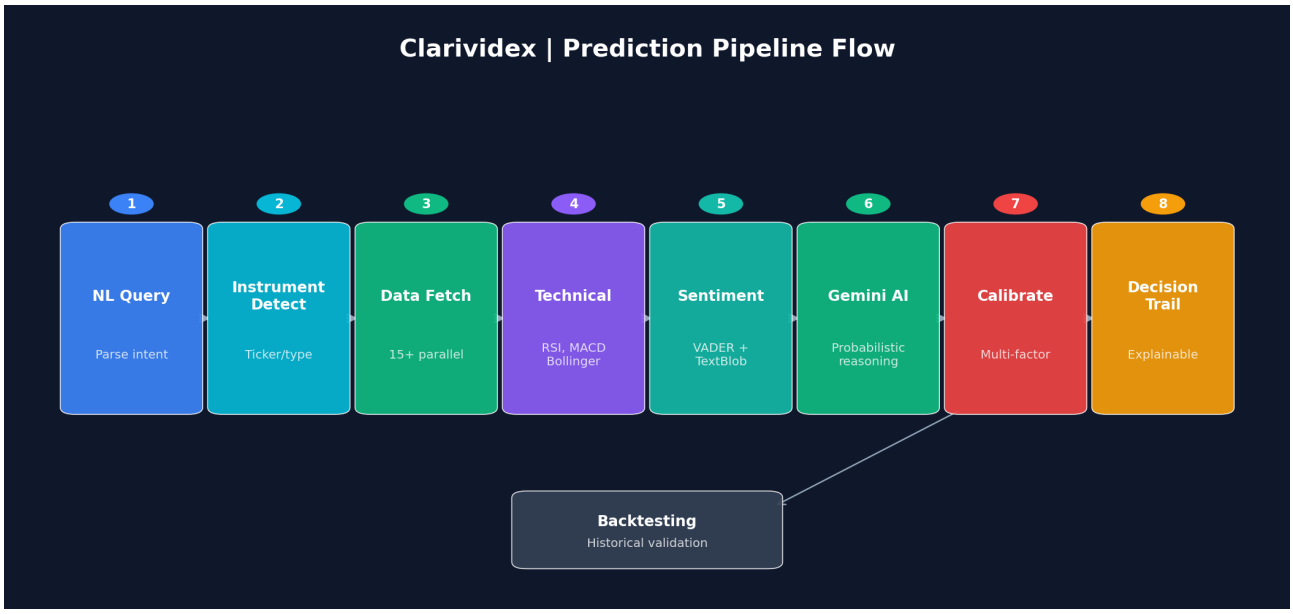
- Multi-instrument | Stocks (US/India), Crypto, Forex, Commodities, ETFs
- Technical Analysis | RSI, MACD, SMA/EMA, Bollinger, ATR
- NLP Sentiment | VADER + TextBlob from 12+ RSS feeds, Reddit, StockTwits
- Decision Trail | Full explainability with category scores and weights
- RAG Chatbot | ChromaDB + MiniLM-L6-v2 embeddings

- Backtesting Engine | Historical accuracy evaluation
- Containerized Lambda via ECR with CloudWatch monitoring

Tech Stack

Layer	Technology	Purpose
Backend	Python 3.12, FastAPI, Mangum	Serverless API on Lambda
Frontend	Next.js 16, React 19, Recharts	Probability gauges
AI	Google Gemini 2.0 Flash	Prediction reasoning
NLP	VADER, TextBlob	Sentiment scoring
Vector DB	ChromaDB, MiniLM-L6-v2	RAG retrieval
IaC	Terraform (6 modules)	ECR, Lambda, CDN

Process Flow



Multi-Agent Orchestration

Uses multi-agent data aggregation: 15+ collectors operate in parallel, each for a specific domain. Gemini orchestrates reasoning across all signals with technical weights and sentiment calibration.

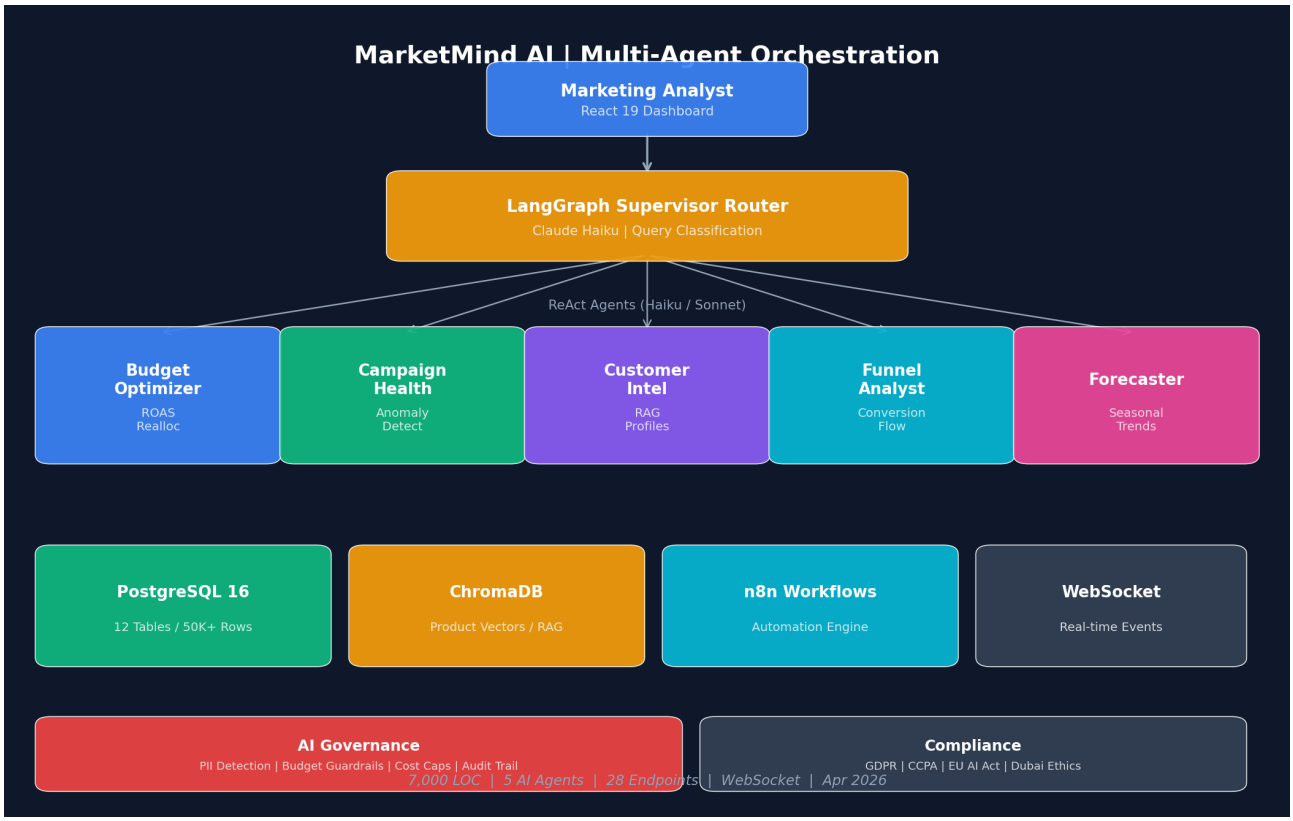
5 MarketMind AI | Multi-Agent Marketing

Marketing teams drown in data but starve for insight. MarketMind fixes this with 5 AI agents -each a specialist. Ask about budget allocation and the Budget Optimizer agent kicks in; ask about customer segments and the Customer Intelligence agent searches your product vectors via RAG. A LangGraph supervisor routes every query to the right agent, while a governance layer ensures PII never touches the LLM.

AI SaaS (Multi-Agent Analytics Platform) | Apr 2026 | Personal Project

7,000	5	28	12
Lines of Code	AI Agents	Endpoints	DB Tables

System Architecture



Key Features

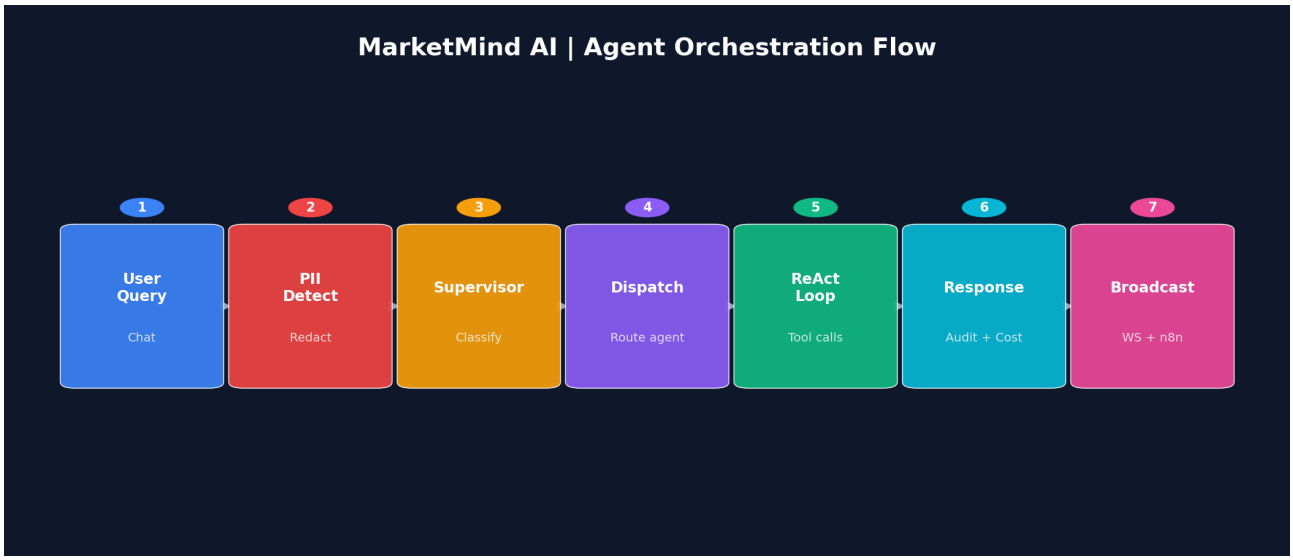
- 5 Agents | Budget Optimizer, Campaign Health, Customer Intel, Funnel, Forecaster
- LangGraph Supervisor | Haiku router classifies and dispatches queries
- ReAct Pattern | Each agent has SQL/vector tools for autonomous access
- RAG Customer Intelligence | ChromaDB vector search (Sonnet 4.5)
- AI Governance | PII redaction, budget guardrails, daily cost caps

- Compliance | GDPR, CCPA, EU AI Act, Dubai AI Ethics
- Real-time WebSocket event bus + n8n workflow automation

Tech Stack

Layer	Technology	Purpose
AI	LangGraph, LangChain, Bedrock	Multi-agent orchestration
Models	Claude Haiku 4.5, Sonnet 4.5	Cost-optimized tiers
Backend	Python 3.12, FastAPI	Async API server
Frontend	React 19, Vite, Recharts	Real-time dashboard
Database	PostgreSQL 16, ChromaDB	Relational + vector
Automation	n8n, WebSocket	Workflow + real-time

Process Flow



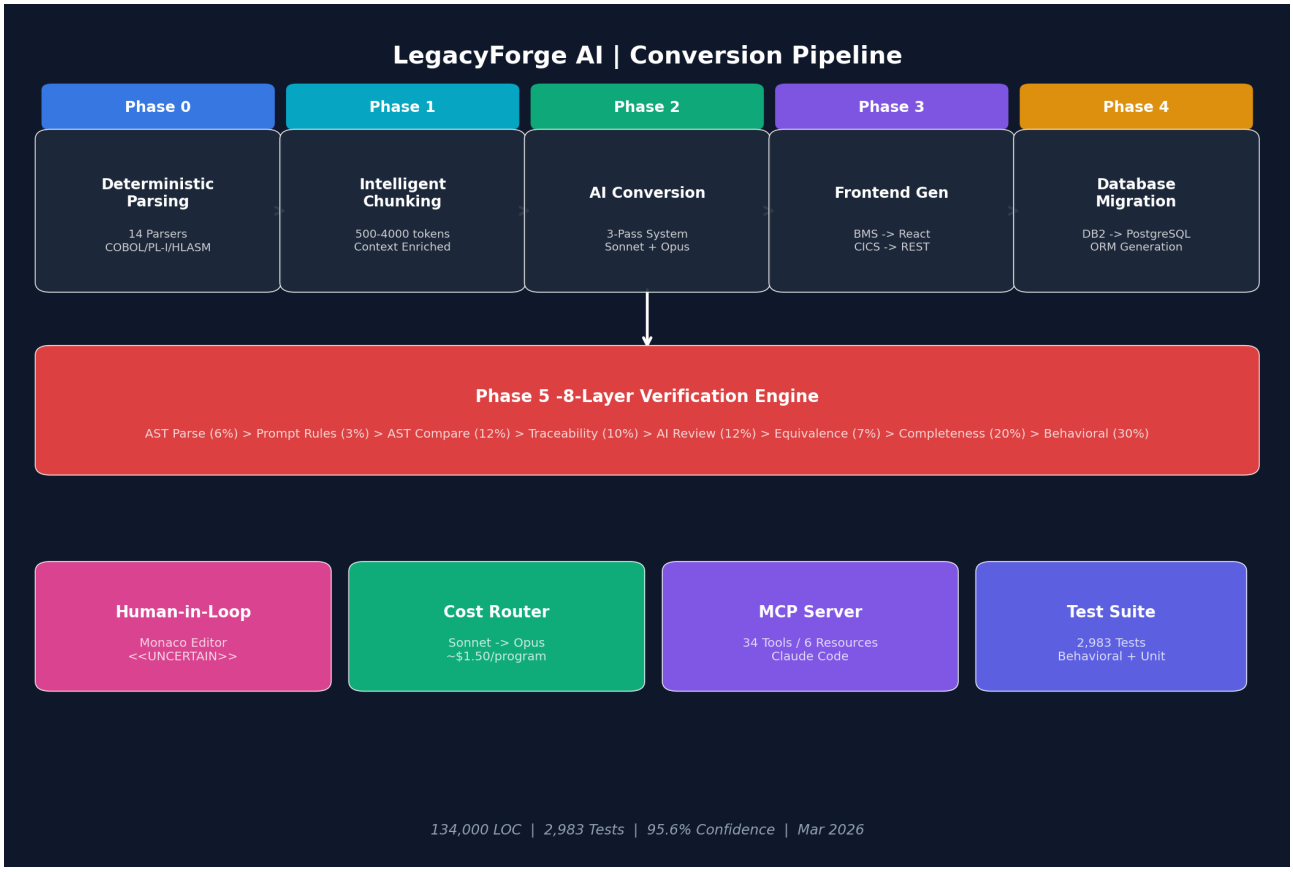
6 LegacyForge AI | Mainframe Modernization

The biggest project in this portfolio -and the hardest problem I have tackled. Legacy mainframes run the world but nobody wants to maintain COBOL. LegacyForge takes COBOL, PL/I, and Assembler programs through a 6-phase AI pipeline: parse, chunk, convert (3-pass with Claude), generate frontend, migrate database, then verify with 8 layers of automated testing. The result? 95.6% average confidence, with human-in-the-loop review for anything uncertain. 134,000 lines of code, 2,983 tests.

Enterprise Tool (AI Conversion Platform) | Mar 2026 | Internal (demo available)

134,000	2,983	95.6%	34
Lines of Code	Tests	Confidence	MCP Tools

System Architecture



Key Features

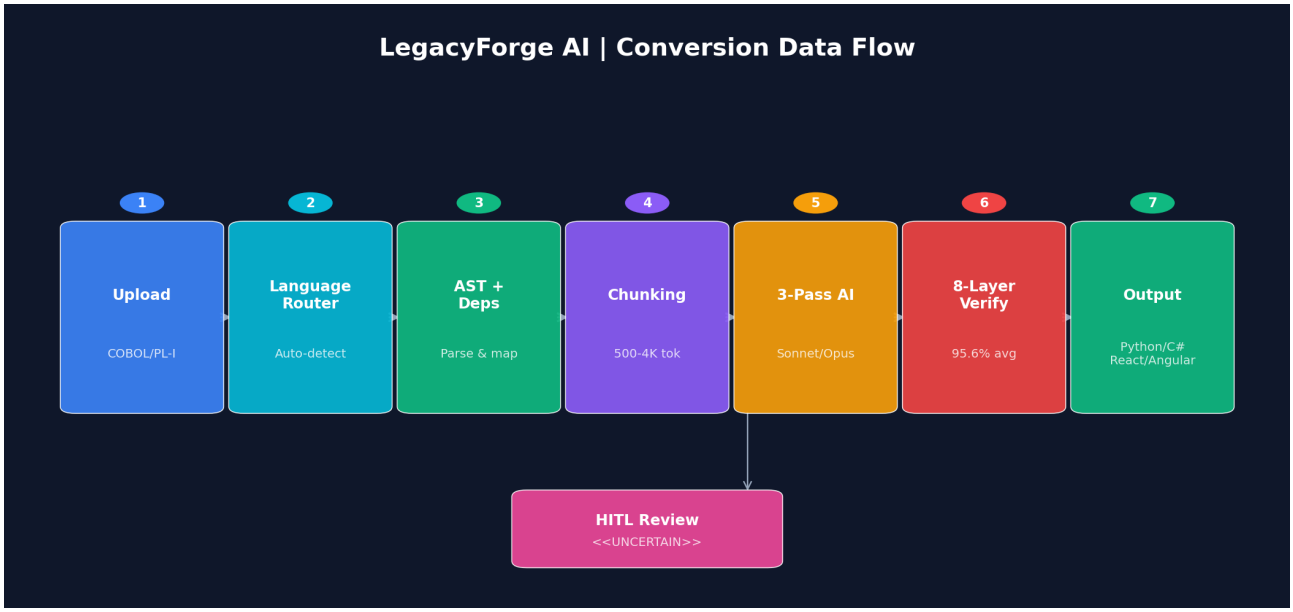
- 6-Phase Pipeline | Parse, Chunk, Convert, Frontend, Database, Verify
- 14 Deterministic Parsers | COBOL, PL/I, HLASM, Natural, CLIST, IDMS, JCL
- 3-Pass AI Conversion | Structure Map, Node-by-Node, Stitch & Reconcile

- 8-Layer Verification | AST, traceability, AI review, behavioral (GnuCOBOL)
- Cost-Aware Router | Sonnet first, Opus escalation (~$1.50/program)
- HITL | Monaco editor with side-by-side diff for <<UNCERTAIN>> markers
- MCP Server -34 tools, 6 resources for Claude Code integration
- 2,983 automated tests across 40 test files

Tech Stack

Layer	Technology	Purpose
AI	Claude Sonnet 4.5, Opus 4.6	Cost-aware conversion
Framework	LangGraph, LlamaIndex	Orchestration + RAG
Backend	Python 3.11, FastAPI, Celery	Async pipeline + queue
Frontend	React, Monaco Editor	HITL review dashboard
Database	PostgreSQL 16 (pgvector)	Embeddings + data
Protocol	MCP (34 tools)	Claude Code integration

Process Flow



Multi-Agent Orchestration

Multi-agent pipeline: each phase is an autonomous agent. CostAwareRouter supervises Sonnet/Opus routing. Verification meta-agent coordinates 8 layers. MCP server exposes 34 tools for IDE orchestration.

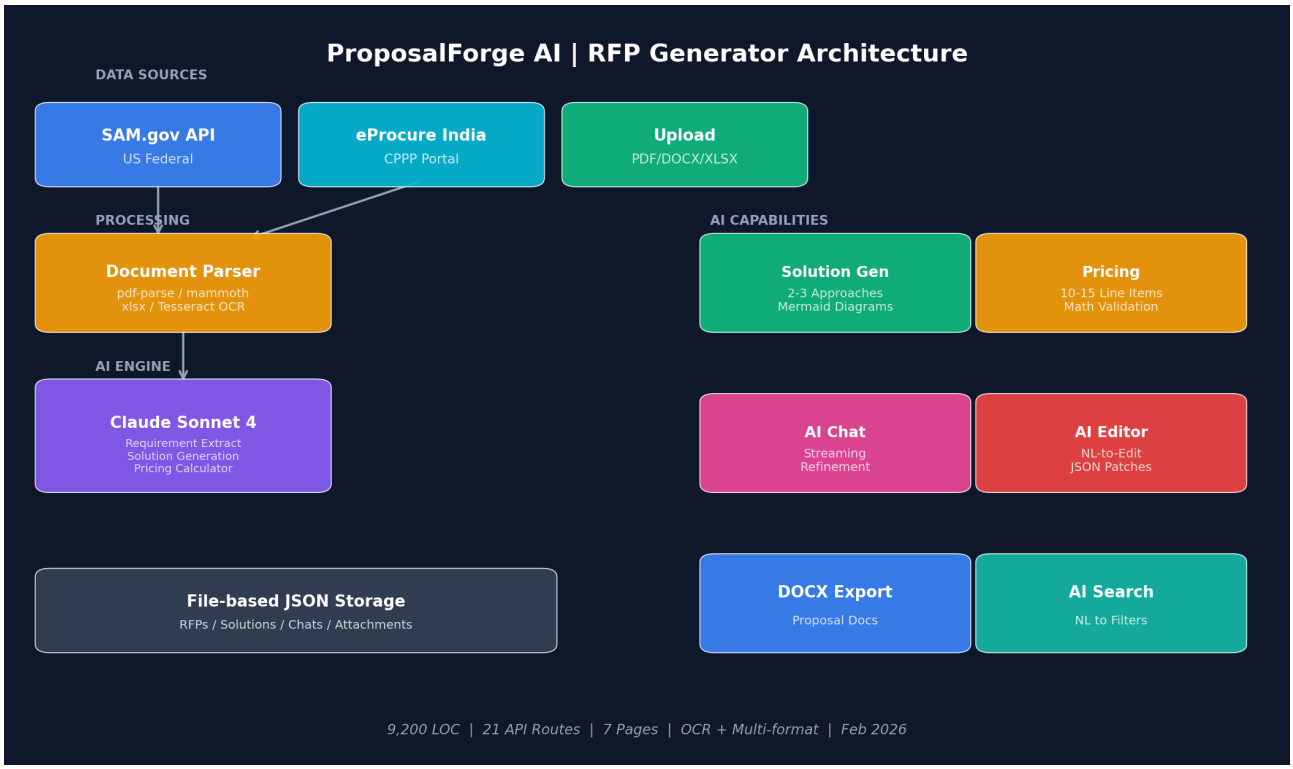
7 ProposalForge AI | RFP Generator

Writing government proposals is tedious -reading 50-page RFPs, extracting requirements, pricing solutions. ProposalForge automates the whole cycle: fetch RFPs from SAM.gov and India's eProcure, parse any format (PDF, DOCX, even images via OCR), then have Claude Sonnet 4 extract requirements and generate 2-3 solution approaches complete with Mermaid flow diagrams and validated pricing estimates.

Internal Web App (AI-Powered RFP Tool) | Feb 2026 | Internal (demo available)

9,200	21	7
Lines of Code	API Routes	Pages

System Architecture



Key Features

- Dual-Country Fetch | SAM.gov (US) + eProcure/CPPP (India)
- Multi-Format Parse | PDF, DOCX, XLSX, OCR (Tesseract.js)
- AI Solutions -2-3 approaches with Mermaid diagrams, team sizing
- Pricing Calculator -10-15 line items with math validation
- AI Chat | Streaming refinement of solutions
- DOCX Export | Professional formatted proposals

Tech Stack

Layer	Technology	Purpose
Framework	Next.js 14, React 18, TypeScript	Full-stack + API routes
AI	Claude Sonnet 4 (Bedrock)	Analysis + generation
Parsing	pdf-parse, mammoth, tesseract.js	Multi-format + OCR
UI	Tailwind, shadcn/ui	Component library

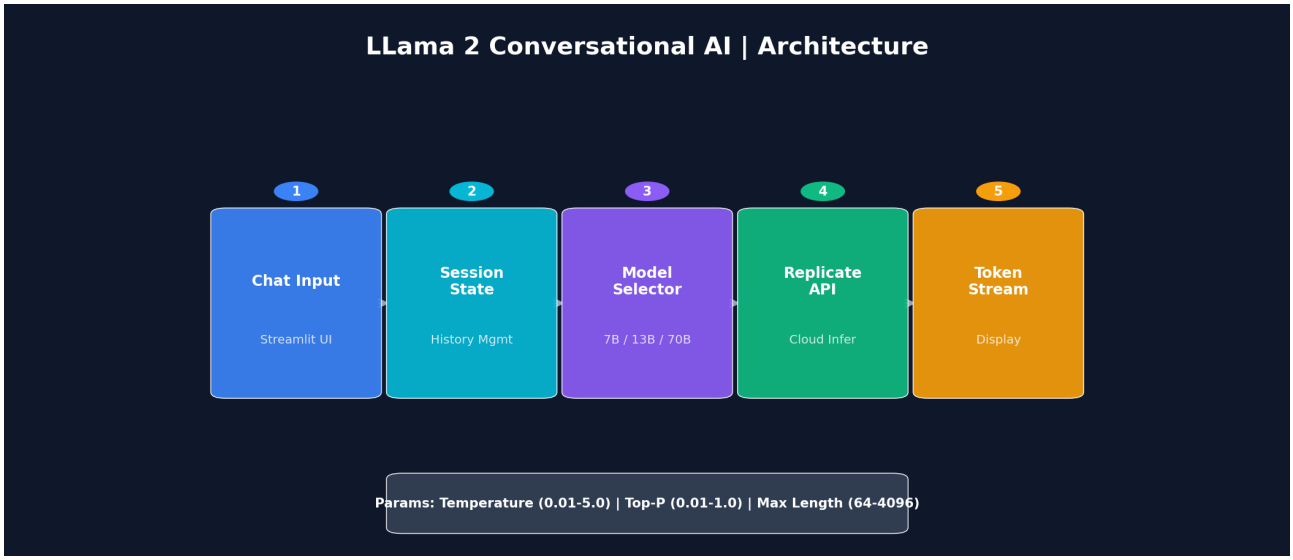
8 LLama 2 Conversational AI

My first deep dive into LLMs. I built a chatbot that lets you switch between three sizes of Meta's LLama 2 (7B, 13B, 70B) in real-time -see how model size affects response quality firsthand. Streaming tokens, configurable temperature and top-p, full conversation context. Simple, but it taught me the fundamentals that made everything else possible.

Web Application (AI Chatbot) | Feb 2024 - Apr 2025 | Personal Project

3	70B	14 months
Models	Max Params	Duration

System Architecture



Key Features

- Multi-model | LLama2 7B, 13B, 70B with runtime switching
- Configurable | Temperature, Top-P, Max Length
- Streaming | Incremental token display
- Context-aware | Full dialogue history per session

Tech Stack

Layer	Technology	Purpose
Language	Python	Single-file app
UI	Streamlit	Chat interface
LLM	Meta LLama 2 (7B/13B/70B)	Multi-model
Inference	Replicate API	Cloud serving

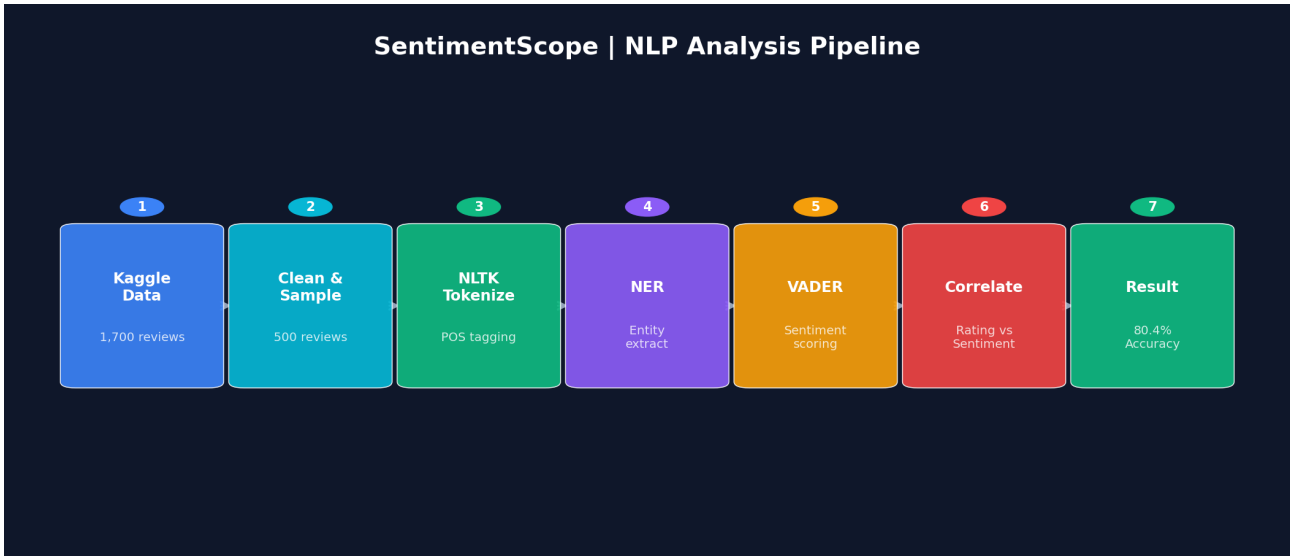
9 SentimentScope | NLP Analysis

Do star ratings actually match what people write in their reviews? I analyzed 1,700 Amazon MacBook Air reviews with VADER sentiment analysis and NLTK NLP to find out. The answer: 80.4% of the time, yes. But the 19.6% where they diverge told a more interesting story -slang, sarcasm, and seasonal trends all confuse rule-based sentiment models.

Data Science Notebook (NLP Pipeline) | Jan 2024 | Academic Project (ASU)

1,700	80.4%	5
Reviews	Accuracy	Techniques

System Architecture



Key Features

- VADER Sentiment | Compound, positive, negative, neutral scoring
- NLTK Pipeline | Tokenization, POS tagging, NER
- Correlation Analysis | Sentiment vs star ratings
- Confusion Matrix -80.4% accuracy evaluation

Tech Stack

Layer	Technology	Purpose
Language	Python 3 (Jupyter)	Interactive notebook
NLP	NLTK, VADER	Sentiment + NER
Data	pandas, numpy	Processing
Viz	matplotlib, seaborn	Charts

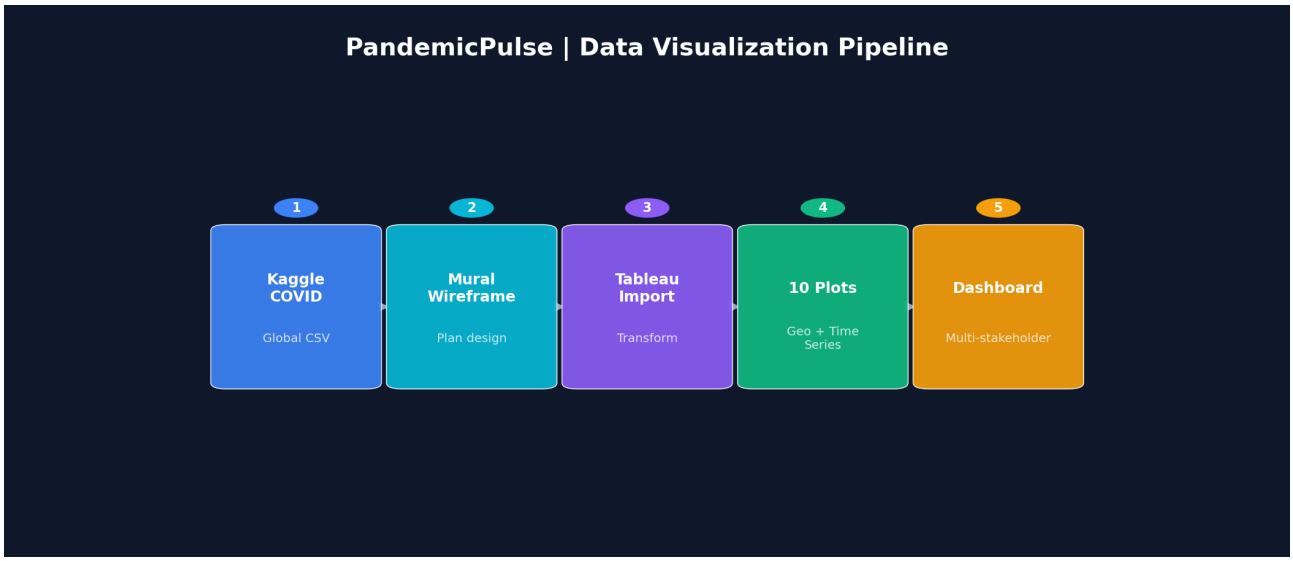
10 PandemicPulse | COVID-19 Dashboard

During the pandemic, I saw how hard it was for decision-makers to get a clear picture from raw data. PandemicPulse turned the Kaggle COVID-19 dataset into 10 interactive Tableau visualizations designed for four different audiences: public health officials tracking hotspots, policy makers evaluating restrictions, businesses assessing risk, and communities seeking transparency.

BI Dashboard (Data Visualization) | Feb 2024 | Academic Project (ASU)

10	4
Visualizations	Stakeholders

System Architecture



Key Features

- Geographic spread | Regional and national tracking
- Time-series | Cases, deaths, recoveries over time
- Multi-stakeholder | Health, policy, business, community views
- Mural wireframing for collaborative planning

Tech Stack

Layer	Technology	Purpose
Tool	Tableau	Interactive dashboards
Planning	Mural	Wireframing
Data	Kaggle COVID-19	Global records

The Engineer Behind the Code

From a Computer Science degree in Chennai to a Master's at Arizona State (4.0 GPA), and from writing my first Python script to orchestrating multi-agent AI systems that convert decades-old COBOL -this portfolio represents a journey of relentless curiosity. Every project here was built to solve a real problem, not to check a box. The best engineers don't just write code; they understand why the code matters.

- **Multi-Agent AI -**Supervisor patterns, ReAct agents, cost-aware routing across Claude, Gemini, LLama
- **Enterprise Scale -**Serverless platforms with multi-tenant isolation, Keycloak SSO, RBAC (110K+ LOC)
- **Legacy Modernization -**LegacyForge: COBOL-to-Python with 8-layer verification at 95.6% confidence
- **Full-Stack -**Angular, React, Next.js, FastAPI, NestJS, Express, PostgreSQL, Terraform
- **Quality -**2,983 automated tests, CI/CD, Docker, MCP integration, OpenAPI documentation

abhismail998@gmail.com | linkedin.com/in/abhin1998 | github.com/Abhishekn1947 |