Innovation partner H2S HACK2SKILL

Media partner YOURSTORY

AI for Bharat Hackathon

Powered by aws

Team Name : **CIPHER AGENTS**

Team Leader Name : **Sourav Kumar**

Problem Statement : **Design an AI-driven solution that helps create, manage, personalize, or distribute digital content more effectively.**

Brief about the Idea:

The Concept: A Self-Learning, Closed-Loop Content Ecosystem:

- Current Generative AI tools operate in silos—they generate text but don't know if it's true, and they definitely don't know if it actually performed well after posting. Our solution, the **Noetic.AI - Intelligent Content Creator That Learns**, bridges these gaps by unifying **Creation**, **Verification**, and **Analytics** into a single, serverless workflow on AWS.

- **What our project does?**

  ➢When a post succeeds, the system remembers *why*.
  ➢When a draft is flagged for bias, the system remembers *what to avoid*.
  ➢When a user creates new content, the AI instantly recalls their specific tone and style preferences.
  ➢The platform doesn't just generate content; it gets smarter and more personalised with every single interaction.

Brief about the Idea:

How It Works (The 3 Pillars of Noetic.AI):

- **Agent-Driven Creation (Not just a Chatbot):**
  Instead of a simple prompt-response box, we use **AWS Step Functions** to orchestrate a team of specialized AI agents. One agent handles "Ideation" (brainstorming angles), another handles "Structuring" (formatting for LinkedIn/Twitter), and another handles "Refinement." This ensures high-quality, platform-native content every time.

- **The Trust Engine (Safety First):**
  In the age of misinformation, generation without verification is dangerous. Our platform includes a dedicated **Trust Analysis Engine**. Before you publish, the content is passed through:

  1. Fact-Check Agents (using Claude 3.5 Sonnet to verify claims).
  2. Bias Detectors (using Amazon Comprehend).
  3. Forensic Scanners (using Amazon Rekognition to detect fake media).

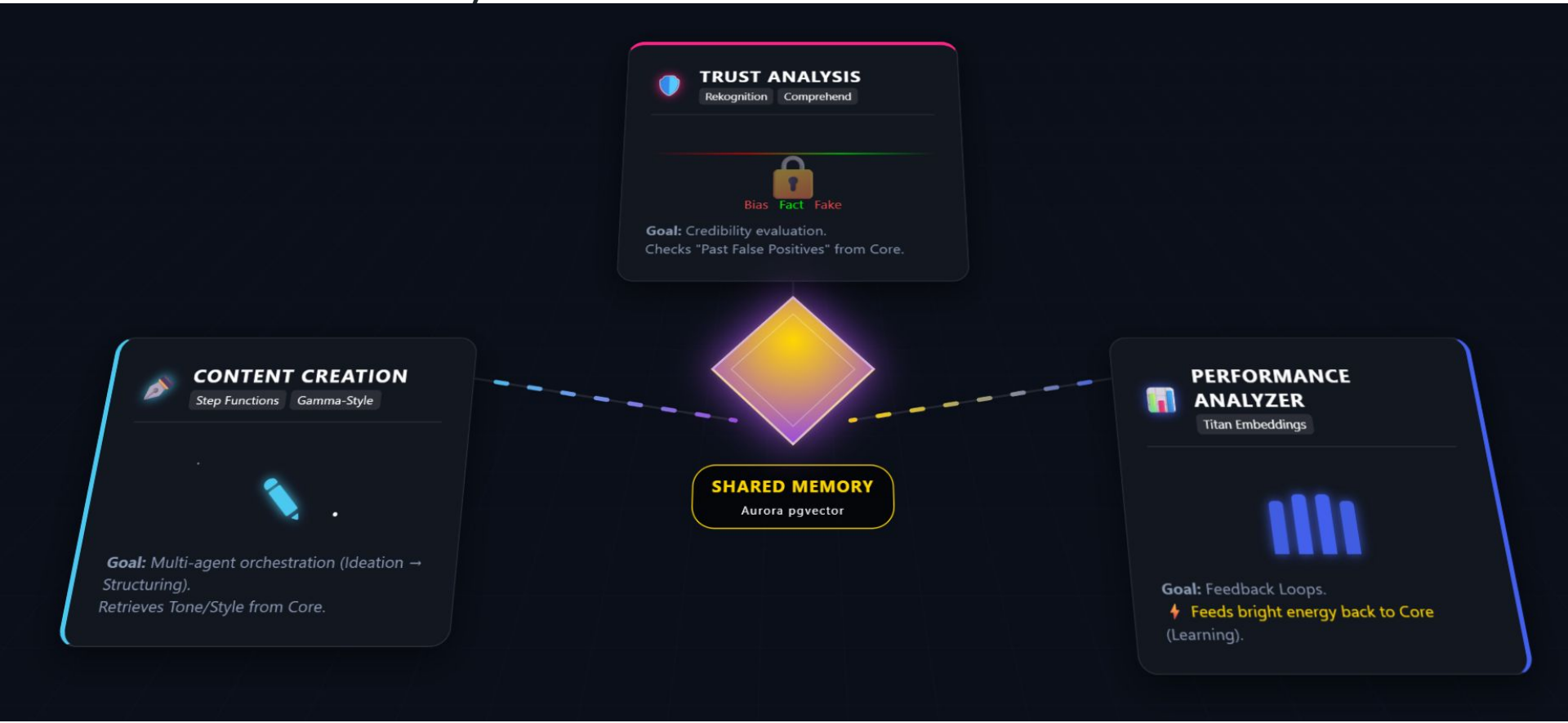
- **Shared Memory (The "Brain"):**
  This is our core innovation. We use **Amazon Aurora PostgreSQL with pgvector** to build a "Shared Memory" for the AI.

  - ➢ *The Cycle:* When you analyze the performance of a past post, the system extracts **insights** (e.g., "Short, question-based hooks got 20% more engagement")
  - ➢ *The Learning:* These insights are vectorized and stored.
  - ➢ *The Application:* The next time you generate content, the Creation Agent queries this memory. It automatically applies verified success patterns to your new drafts.

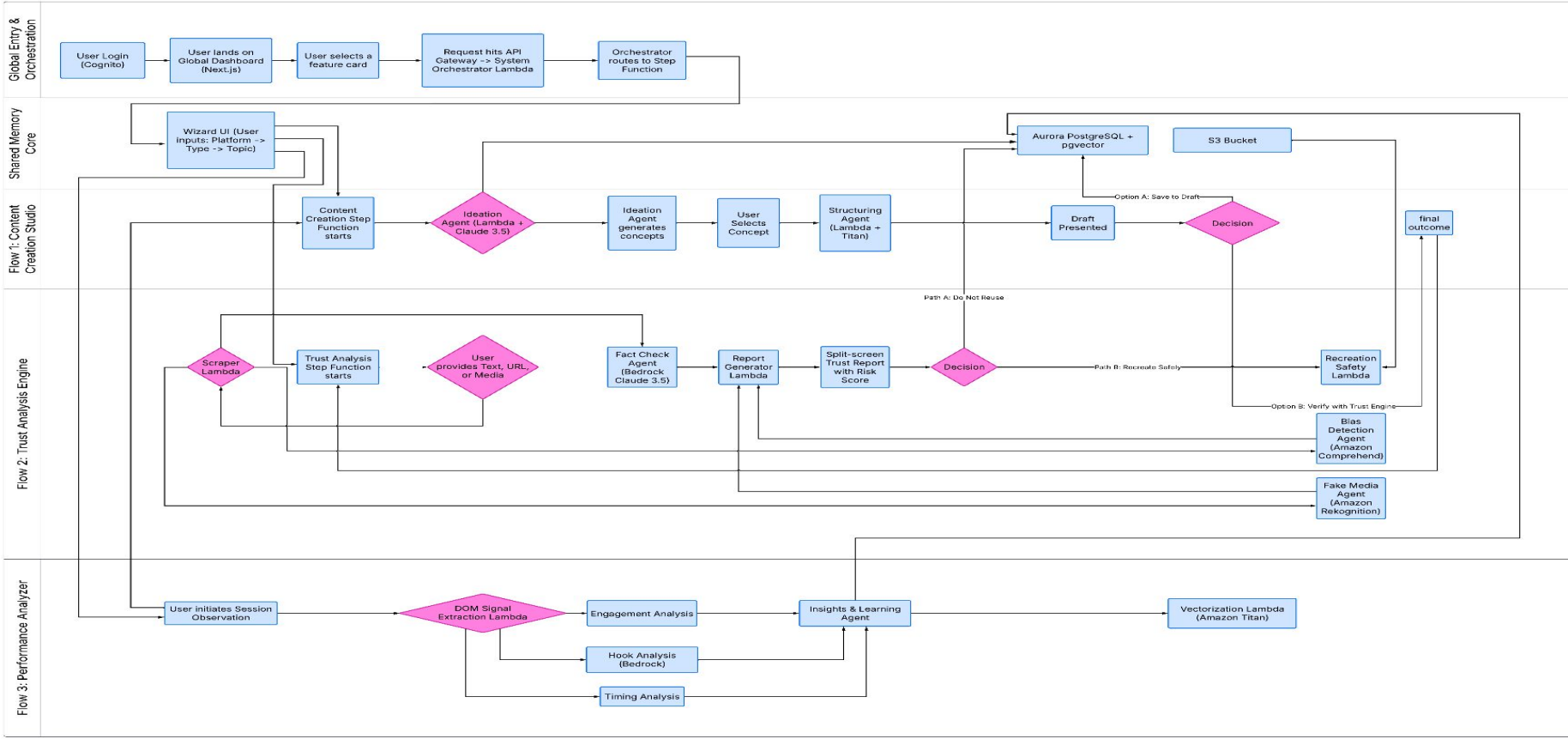
Your solution should be able to explain the following:

- **How different is it from any of the other existing ideas?**.
  - ➢ Existing tools are either "Generators" (ChatGPT wrappers) or "Schedulers" (Buffer/Hootsuite). We are a **Closed-Loop System**. We don't just output content; we verify it with specialized Trust Agents and refine it with Performance Agents. Our system remembers your specific tone and success patterns across different sessions, whereas standard chatbots "forget" context once a chat closes.
- **How will it be able to solve the problem?**
  - ➢ It solves the **efficiency** problem by using AWS Step Functions to orchestrate specialized agents (Ideation, Structuring, Fact-Checking) in parallel.
  - ➢ It solves the **quality** problem by enforcing a Trust Engine that scans for bias and fake media before publishing.
  - ➢ It solves the **optimization** problem by vectorizing high-performing metrics into Amazon Aurora, so the AI knows exactly what works for *your* audience next time.
- **USP of the proposed solution:**
  - ➢ **"Shared Memory" Architecture:** Using Amazon Aurora PostgreSQL with pg vector, the platform learns. If a "question-based hook" gets 20% more engagement, the Ideation Agent automatically prioritizes that structure for future posts.
  - ➢ **Seamless Trust Handoff:** If the Trust Engine detects misinformation, it doesn't just block it; it offers a "Recreate Safely" workflow that guides the Generative AI to rewrite the content accurately.

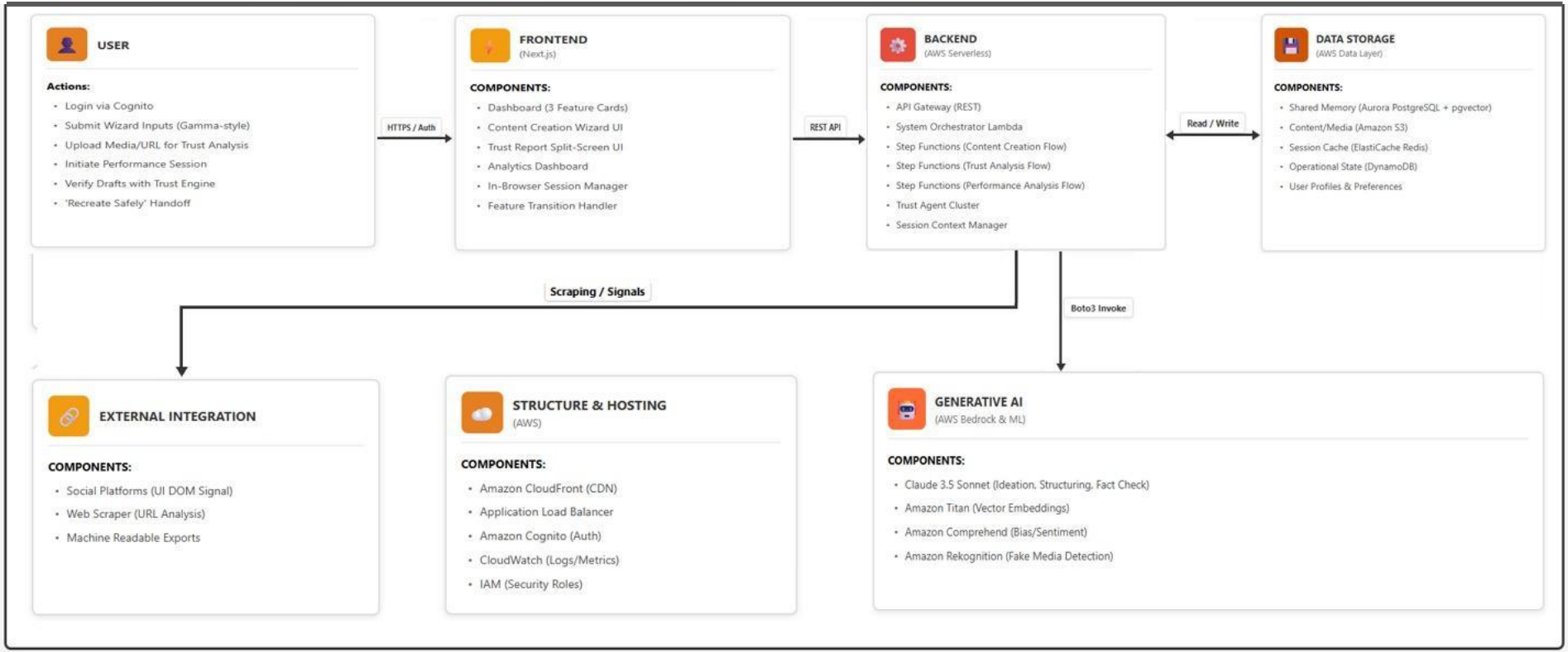
List of features offered by the solution



Process flow diagram or Use-case diagram



Architecture diagram of the proposed solution:



Technologies to be used in the solution:

- **Core Cloud Infrastructure:** AWS Serverless (Lambda, API Gateway, EventBridge).
- **Orchestration:** AWS Step Functions (for managing complex agent workflows).
- **Generative AI & LLMs:** AWS Bedrock (Claude 3.5 Sonnet, Amazon Titan).
- **Machine Learning Services:** Amazon Comprehend (NLP/Sentiment), Amazon Rekognition (Computer Vision).
- **Database & Storage:** Amazon Aurora PostgreSQL with pgvector extension (Vector Database), Amazon DynamoDB (State management), Amazon S3.
- **Frontend:** Next.js, React, Tailwind CSS.

Estimated implementation cost (optional):

- **Architecture Type:** Fully Serverless (Pay-per-use).
- **Estimated MVP Monthly Cost:** ~$120 - $180 USD.
- *AWS Lambda & Step Functions:* Free tier eligible, then ~$10/month for initial scale.
- *Amazon Aurora Serverless v2:* ~$60 – $90/month (scales down when not in use).
- *AWS Bedrock:* ~$50/month (based on token usage for generation).
- *Other (S3, Logs, Gateway):* ~$10/month.
- **Cost Efficiency:** No idle servers (EC2) means zero cost when users aren't active, making it highly cost-effective for a hackathon/startup launch.

Innovation partner H2S HACK2SKILL | Media partner YOURSTORY

AI for Bharat Hackathon

Powered by aws

Thank You