

Proposal: AI-Powered Support Agent for Akash Network

Executive Summary

This proposal presents a solution to address onboarding and support challenges within the Akash ecosystem using a Retrieval-Augmented Generation (RAG)-based AI Agent. The agent will be integrated into both Discord and a lightweight WebApp, enabling real-time assistance for new users, developers, and community members.

The agent aims to:

- Reduce response delays
- Handle frequently asked questions
- Escalate complex issues to human experts

We are seeking community engagement and support for a small community pool funding request to develop a minimum viable product (MVP) in two clear milestones that serve the ecosystem in a practical, sustainable way.

About the Contributor

Name: Fenil Modi

Background: Developer focused on decentralized infrastructure and AI-native tooling for Web3 ecosystems.

Relevant Experience

- Building support agents using ElizaOS, a modular AI agent framework
- Integrating RAG systems with:
 - Akash Documentation
 - Awesome Akash repositories
 - AkashJS GitHub
 - Other community-maintained resources
- Using Akash ChatAPI, a hosted, zero-cost inference layer on Akash

- Designing cross-platform UX for Discord and WebApps
-

Problem Statement

As Akash grows, new users rely heavily on Discord for onboarding help. While the Vanguard program plays a critical support role, several issues persist:

- Delays during off-hours
- Repetitive FAQs burdening contributors
- User drop-offs due to lack of immediate support

A lightweight, AI-powered support agent can significantly improve onboarding efficiency and reduce these friction points.

Proposed Solution

Platforms

- **Discord Integration:** Bot responds in support channels or DMs
- **WebApp Interface:** Lightweight, browser-accessible UI for help without requiring Discord

Key Features

- Real-time responses to frequently asked questions
- Live information retrieval from documentation and codebases
- RAG-based knowledge base trained on Akash Docs, GitHub, and community content
- Integration with Akash ChatAPI for zero-cost inference
- Web search integration for enhanced contextual accuracy
- Smart escalation:

- Redirect the user to Vanguard contributors
 - Mention or ping relevant community members
 - Surface the question into active discussion threads
- Personalized tooling or documentation suggestions
- 24/7 availability
- Modular and expandable with community input

Technical Architecture

Component	Technology
Framework	ElizaOS (modular AI Agent Framework)
Inference Engine	Akash ChatAPI (runs fully on Akash infrastructure)
LLMs	Meta LLaMA-3, DeepSeek R1, or similar OSS models
Retrieval Layer	Akash Docs, GitHub, Awesome Akash (indexed search)
Web Search	Integrated using privacy-respecting APIs
Interfaces	Discord Bot, Web UI (Next.js)

Project Milestones & Budget

Milestone 1: AI Agent Core + Discord Bot + Web Search Integration

Includes full core development, Discord bot functionality, RAG-based retrieval system, Akash ChatAPI integration, and intelligent web search.

Task	Estimated Time	Cost
AI Agent Core (RAG + API)	10 hours	\$350
Discord Bot Integration	5 hours	\$175
Web Search Integration	3 hours	\$105
QA & Testing	2 hours	\$70
Subtotal	20 hours	\$700

Milestone 2: WebApp Interface with Fine-Tuned Capabilities

Task	Estimated Time	Cost
WebApp (UI + integration)	6 hours	\$210
Finetuning Responses + UX	2 hours	Included

Maintenance (3 months)	Flat rate	\$100
Subtotal		\$310

Total Estimated Budget: \$1,010 USD

- Hosting: Minimal (via Akash infra)
- Inference: No AI LLMs cost (via Akash ChatAPI)
- One-time Setup: Covers Discord and WebApp
- Maintenance: 3-month MVP phase included

Deployment Options

Option A: Community-Owned

After delivery, the Akash community (or a Vanguard member) can maintain and scale the agent.

Option B: Maintained by Contributor

The contributor will manage deployment and updates during the initial 3-month MVP period, with the option to transition to community stewardship.

Risks and Mitigation

Risk	Mitigation
Inaccurate answers	Use confidence thresholds, source citations, and smart escalation

Lower human interaction	The AI acts as a triage layer, not a replacement
Scope creep	Strictly limit implementation to core MVP features
Low engagement	Launch in a test Discord channel and actively promote through the WebApp

Community Call-to-Action
Feedback and support are welcomed from Akash community members, Vanguard contributors, and governance participants. Your input will help refine the roadmap and strengthen the formal funding request.

Development Timeline

Phase	Duration
Milestone 1: Bot & AI Agent Core	Week 1
Milestone 2: WebApp + Fine-Tuning	Week 2
Feedback & Iteration	Weeks 3–4
Optional Handoff Phase	Post Month 3