

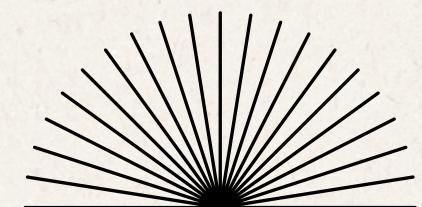
PROJECT BRIEF DECK

Self-Hosted, AI-Powered Assistant for Local Bitcoin-Paid Businesses

NAME OF PROJECT:
Bitcoin AI Hub

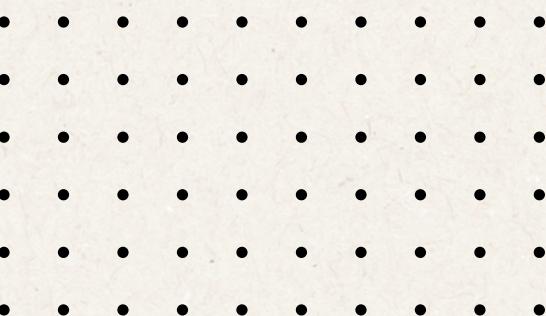
PRESENTED BY:
Alan Copa
Joel Kleibs
Diell Kryeziu
Delara Lomen

PRESENTED TO:
Bitcoin + AI
Hackathon



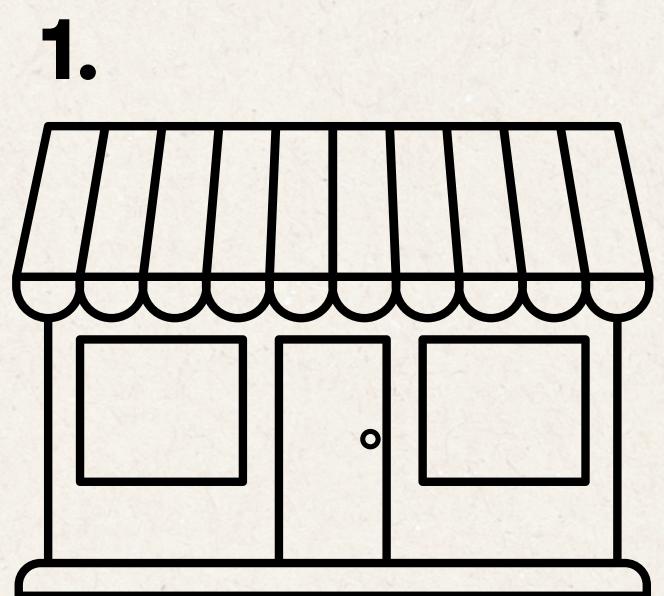
Project Idea: ClerkBit

It's your digital employee who takes orders, answers questions, collects Bitcoin, and tracks revenue.



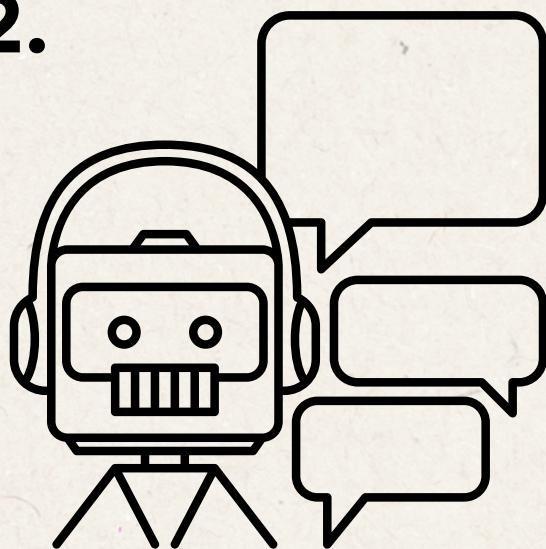
Example User Flow

A café, shop, or service provider can run their own intelligent assistant – no subscriptions, no cloud lock-in, no outages – just smart, private, Bitcoin-powered AI on-premises.



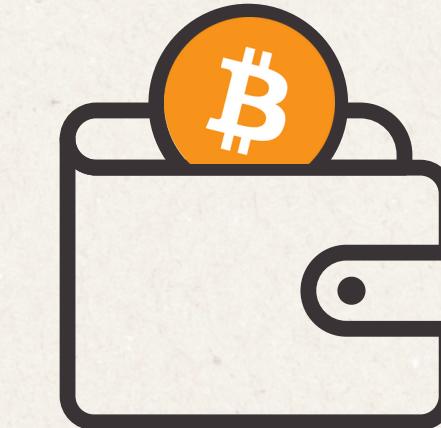
A Lugano café installs the self-hosted app.

1.



They configure an AI chatbot to greet and take orders from customers.

2.



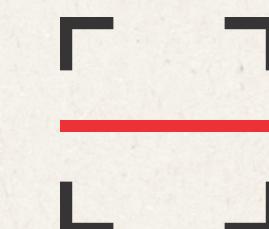
Payments go directly to the café's BTC wallet via Lightning.

3.

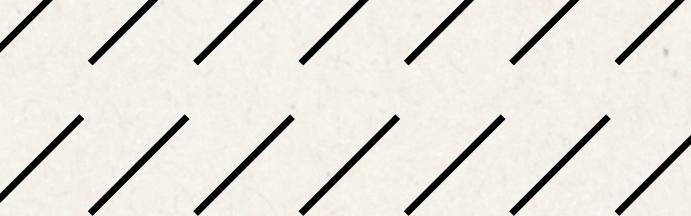


The café gets analytics: "You've earned 0.02 BTC this week" + business tips.

4.

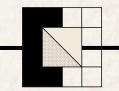
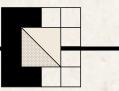


Locals or tourists interact with the AI via QR code or web app, paying in Bitcoin.



Key Features

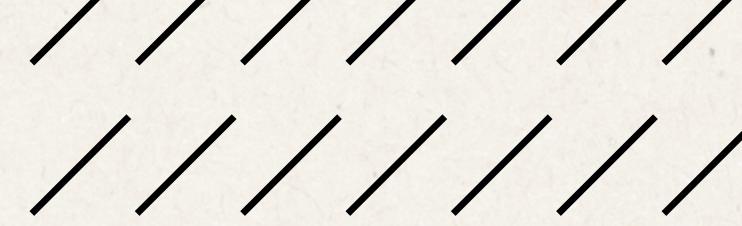
Adaptability



Security

Ownership

Traceability



Key Features - From the Merchant's Standpoint

Complete Ownership of the AI assistant and Its Functionality:

no subscriptions, no cloud lock-in, no outages – just smart, private, Bitcoin-powered AI on the premises of your business

Real-time Business Analytics:

derive valuable data that can be analyzed in real-time and aid to curate a personalized experience for your customers further down the line

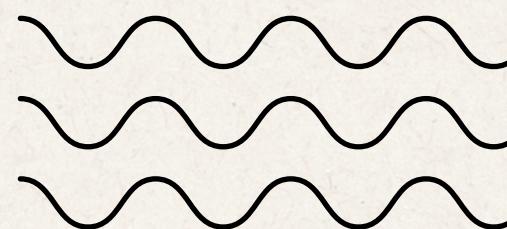
Complete Ownership of Customer Data:

no third party-access (*What happens in Vegas stays in Vegas*)

if the business is unable to self-host a bitcoin node, they could outsource the hosting in a marketplace

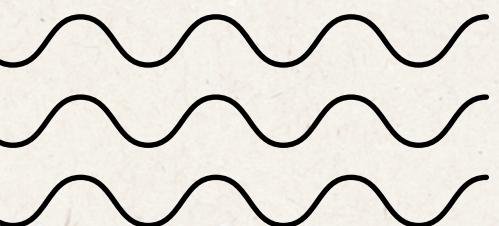
Tech Stack

- AI Models: Prem AI (provides self-hosted AI models that can be explained + external tooling for regulatory-grade traceability)
- Payments: Alby (for wallet integration, lightning payments).
- Backend: FastAPI / Node Handles AI requests, invoice logic, local storage
- Frontend: React + PWA interface for mobile use.
- Hosting: Comet (for running your own bitcoin node).
- Blockchain Analytics: Use ElectrumX (not their partner but it connects to a local Bitcoin Core full node and indexes all historical transactions) + slap an LLM on it.
- Hardware: Docker, Raspberry Pi (easy install + optional preloaded device)
- POS + Identity: Portal (for payment initiation & customer identification, essentially payment UX; customer specific data is stored locally on the merchant's device)



Hackathon MVP

- Local web app UI (React or simple Flask)
- One AI model (chat + voice) – e.g., Whisper + TinyLLM
- Lightning invoice generator (LND or BTCPay Server API)
- Basic analytics dashboard: ???
- Docker setup for local deployment



TO DO:

- figure out how to simulate payments over Lightning
- set up a local testnet node (Comet)
- connect Alby wallet with testnet node
- build basic front-end back-end
- maybe build a dashboard

