# BabyBTC Quant Lab

## A Baby PoW Chain + AI Coach

Scoop AI Hackathon – Autonomous Finance & Quant AI

# Problem: Knowledge Gap in On-chain Finance

- Blockchain, PoW, and on-chain risk are **hard to internalize**
- In quant / DeFi teams:
  - A few engineers really understand blocks & consensus
  - Most people only see **black-box dashboards**
- Real chains involve **real money**:
  - Not safe for “learning by breaking things”
  - Hard to experiment with attacks or failures

# Idea: A Safe Baby Chain Playground

**BabyBTC Quant Lab** =

- A **toy PoW blockchain** running locally / in LAN
- **Virtual BABY tokens only** – no real assets, no fiat
- Players join from a web UI, can:
  - Mine blocks
  - Transfer tokens
  - See the chain evolve in real time
- An **AI coach (SpoonOS Agent)** explains what just happened

# What We Built

- **Toy PoW Chain**
  - MD5-based hash, 4-digit nonce, adjustable difficulty
  - Blocks with `prev_hash`, `merkle_root`, `timestamp`, `nonce`
- **Player Balances & Transfers**
  - Simple UTXO-like / account model
  - Pending tx pool → next mined block
- **ChatGPT-style Frontend**
  - React + Tailwind, 3-column layout:
    - Mining & balance
    - Chain timeline & transfers
    - AI coach chat panel

# Architecture

```
Browser (React + Tailwind)
    ↓  HTTP /agent/ask, /api/game/...
Agent Service (FastAPI + SpoonOS)
    ↓  tools: recent_events, player_stats, chain_summary
BabyBTC Node (FastAPI)
    ↳ In-memory chain_state + event_log
```

- **Backend**: Python + FastAPI
  - Toy PoW, players, tx, event log
- **Frontend**: React + TS + Vite
  - Auto-register, mining controls, chain view
- **Agent**: SpoonOS / spoon_ai
  - ToolCallAgent with custom tools on chain_state

# How the Toy PoW Works

- Block header:

```
index : prev_hash : merkle_root : timestamp : nonce
```

- Hash:

```
h = md5(header)
hash_value = int(h[:4], "big") % 1_000_000
```

- Condition:

```
hash_value < difficulty  (e.g. 400_000)
```

- Result:
  - ~5–10 seconds to mine a block on a laptop

## Where SpoonOS Fits

- Agent Service defines tools:
  - `get_recent_events(limit)`
  - `get_player_stats(player_id)`
  - `get_chain_summary()`
- SpoonOS ToolCallAgent:
  - Chooses tools based on user questions
  - Reads real BabyBTC state
  - Responds in **educational, risk-aware language**

# Why This Is Quant / Finance AI

- Focus on **understanding**, not black-box signals
- Lets teams:
  - See how on-chain state evolves
  - Ask AI “why” in terms of:
    - Consensus
    - Risk
    - Attack surface
- Can be extended with:
  - Simple PnL / strategy simulation on top of BabyBTC
  - More advanced attack scenarios

# Demo Flow

1. Open web UI → auto-register player + mnemonic
2. Start mining → see new blocks and rewards
3. Open second client → transfer BABY between players
4. Ask AI coach:
    - “Explain the last block”
    - “What risks are present now?”
5. See AI explanation grounded in actual BabyBTC events

## Roadmap

- Add:
  - Multi-node BabyBTC network
  - Toy RSA signatures instead of HMAC
  - Attack / rollback demos
  - Simple strategy PnL analysis
- Package as:
  - A reusable **teaching lab** for blockchain & quant courses

# Thank You

**BabyBTC Quant Lab**

A safe Baby chain + AI coach

for learning on-chain finance from the bottom up.