CertiK Backend Engineer Interview Project: A Naive Distributed Transaction System

Write a naive distributed transaction system that supports two or more nodes. Each node can be thought of as a RESTful server.

- Node can broadcast and respond messages from other nodes.
- A new node will broadcast a message when initiated and choose $k(k \leq 5)$ responded nodes as its peers. The address/id of the new node will be broadcast and synced by every other node in the network.
- Each node is capable of sending and receiving an encrypted transaction to/from another designated node (which may not be its peer). Transactions should be synced by all nodes in the network.
- Node supports RESTful query of its transactions.

Specifications:

1. Design your own transaction format and API. Use any main-stream encryption method you like.
2. Choose your own favorite language and framework (except PHP). Node should be able to work in a Linux environment.
3. A database must be used to save the transactions and the query events. You can choose from either SQL(e.g. PostgreSQL), or NoSQL(e.g. MongoDB), or Key-Value (e.g. Redis) databases.
4. Be free to apply any dark magic that you believe would demonstrate your skill.

You should return:

1. A GitHub/GitLab repository containing documentation of the API.
2. A brief demo (document, video, etc) of your system.