Release Plan

Product: Decentralized Internet

Team: Decentralized Internet

Release Date: June 6th

High Level Goal:

Provide the convenience of modern storage sharing without the need for centralized hosting.

Sprint Plans:

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
User Stories	**1.1** As a user, I want to instantiate a shared network between myself and other users **1.2** As a user, I want to save data to the network so that I can access it later, even if my device is lost **1.3** As a user, I want to be able to access data other people have written to the network	**2.2** As a creator, I want to create a webpage with links to other webpages **2.3** As a publisher, I want to distribute a web page across my network **2.4** As a user, I want to find web pages other users have created **2.5** As a user, I want to re-experience web 1.0	**3.1** As a user, I want the network to be more stable **3.2** As an admin, I want to control how much storage is used by the network **3.3** As an admin, I want to control who is able to read and write my files	**4.1** As a creator, I want to make web pages that read and write from other distributed information **4.2** As a creator, I want to publish those webapps to the network **4.3** As user, I want to access applications other people have put on the network
Infrastructure	Setup Docker to standardize deployment and developer environments Setting up a basic flask	Create and expose endpoints for servers to interact with a web frontend Integrate displaying	Expose fine grained control over fault tolerance, and network stability to the user Allow network to shard	Add support for distributed documents to run javascript that interacts with the distributed storage

	deployment on multiple containers in a shared network Create shared git repository, with automated testing	rendered HTML to the application UI Write infrastructure to allow documents to reference other documents	information so user storage sizes are smaller	Provide ways to ensure consensus amongst nodes to application developers
Spikes	Varying familiarity with distributed systems Compare the cost of using a framework (DynamoDB, Sinfonia, Ray) vs implementing our own distributed kvs Compare the different web front-end solutions for displaying data from the network Different OS issues	Build a new layer on top of the KVS for handling HTML documents Lack of familiarity with HTMX Instability of network as files get larger	No experience implementing permission systems Will need to implement either a paxos-protected directory of where information is stored, or some form of hashing The network will need to adapt to varying levels of instability	Create some apps to be run as a demonstration of it's capabilities Debug apps being developed, and the underlying system

Product Backlog:

Sprint 1	Sprint 2	Sprint 3	Sprint 4
Creating the network with multiple nodes	Displaying files on the network	Sharding	Improved Network snooping functionality
Storing files on the network	Viewing files	File Uploads	Dynamically changing networks
Joining a network	Web links	P2P connectivity	Leaving and joining active networks