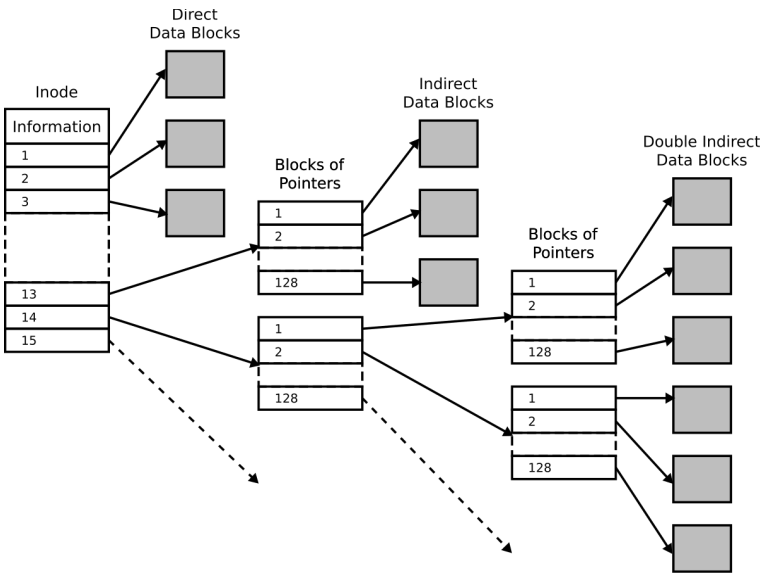
Nematode FS

1.) Introduction

At the moment there are many existing protocols to storing data on the blockchain, B://, BCAT, D://. However, despite the fact there are many different protocols to store data there is no way to organize or collate data. Modern file systems solve this problem by introducing directories, inodes, symlinks etc. Here we introduce Nematode FS a ext2-like file system built specifically to solve these problems with data storage on Bitcoin.

2.) Inodes

The most important part of the the entire file system is the inode itself. The Nematode SDK uses addresses to represent inodes. Each file that exists and each directory is described by an inode that is derived by a parent key. At the top most level there is a root key that the user holds.



Each inode contains some basic meta data as well as pointers to either data regarding an actual file or data regarding a directory which is simply a data structure that holds names associated with links to other inodes.

4.) Fuse bindings

In terms of the user experience fuse bindings as seen in the /fuse directory allows users to seamlessly integrate Nematode on their existing computers. Allowing users mount nematode to a specific directory without anything appearing to be different. On the backend however, when a user or program, creates or modifies a folder, or file, this triggers a complex series of actions that eventually ends up with virtual representation of the file hierarchy replicated on the blockchain through Nematode.

5.) Other application & Business case

Though one can expect Bitcoin to be in itself to be useful for storing permanent data, by using a file system like Nematode one can potentially decrease the overall amount of global cloud storage. This can be achieved as many different applications and products do not have to replicate the files that hold data or logic. Rather, they can be shared and reused by many different people over the course of history. I could not think of clear business case other than creating an interface (which is located in /ui) and charging small processing fees as well as processing fees for using the fuse bindings as well.