Chord will have to provide few basic operations. In particular:

-add(data), will include the data provided into the network. It will return the index of the added item

-get(n), will get the data at index n

-delete(n), will remove the data at index n from the network.

-create(), will create a new Chord network

-join(node), will enter the network whose node is part

-leave(), will leave (softly) the network

Actors:

**API**: it will handle the communication with the application.

**Block Manager**: it will handle the data stored within the node.

**Router**: it will handle the lookups operations and the finger table management.

**Communication Manager**: it will handle all the message received from the network.

**Stabilizer**: verifies immediate successor, and tells the successor about itself. Periodically.

**Fixer**: refreshes finger table entries. Periodically.

**Checker**: checks if the predecessor failed. Periodically.

Initialization

chordInit will become the Api and the Communication Manager will be created from it. After a join/create, Communication Manager will create Router and block manager, notifying API about it. Router will then proceed creating the other 3 services.

Closure

Leave command will proceed to pass all the blocks stored to the successor. Then will kill all the services except Api and Communication Manager, returning in a post chordInit state. Exit will then close the package completely.