**Agritech Decentri-net**

**A decentralized network of agricultural producers, suppliers, and consumers to facilitate greater sharing of goods, make the agricultural market more efficient, and empower smaller farmers to compete with larger industry.**

**Stage 1:**

1. Agritech decentri-net is a decentralized network of agricultural producers, consumers, and suppliers. Each node in the network contains a list of all their yearly production/consumption, as well as a list of all other nodes that they supply to or receive from. Each user stores a list of their own needs and output as well as their neighbors. Whenever they are pinged by the network, they will send the source of the ping the requested information. Each node locally stores a list of all its neighbors, and so the entire decentralized graph can be constructed from flooding through the network and recursively asking each neighbor for their neighbors. This way, each user can create an entire graph of every graph on the network and how they are connected, and explore that graph and its information.
2. Each node is uniquely identified by its IP address (as a string). The Graph class is a local representation of the total network. It has a set of nodes and a set of edges between nodes. The ActorData class is a sparse class that defines several fields such as yearly production, IP address, neighbors (list of IPs), type of actor (producer/supplier/consumer), and the name of that node.

**Stage 2:**

Our system will use three classes to model a decentralized network of agricultural producers.

1. The first class will be called ActorData and will be our producers, suppliers, and consumers. We will use an Enum called kind to distinguish between the three and during instantiation the user will have to specify which of the three he is modeling. The class has a String which stores the IP address of the actor. The class will also have a hashset of strings to store the neighbors of this specific actor. Additionally there will be two hashmaps to store data for consumption and production. The hashmap will map types of produce(strings) to quantity (ints).
2. The second class will be called graph and this is where we are going to store our nodes(strings of actor types) and our edges(connections within the network, ie. connections between producers, suppliers, and consumers.
3. The third and last class for our system will be called local user and this is where a producer, supplier, or consumer can store their information and all their connections. They will be able to look through all their neighbors to get the information on the overall economy and who has what. Every local user will have a graph variable which stores a reference to a graph object to represent the network of connections. Every local user will also store a hashmap that will map strings to actor data so that the user can get the data of that actor.