1. We need: Peer base that contains all the peers we have
2. Do we know something about data type?
3. When another module wants receive a message it sends notify message, gossip stores it in it's cage, then if it receives a message of this type, it asks
4. Gossip Announce
    a. Sent from other modules to notify other peers about some data with **data type**.
5. Gossip Notify
    a. Module asks gossip to notify when it receives data of **data type**.
6. Gossip Notification
    a. Sends data to a subscriber to ask about validation.
7. Gossip Validation
    a. Validation results from a module. If the message is valid, we send Gossip Announce to every peer.
8. Dev:
    a. Peer-base module (Lukas + later Kyryll)
        i. Create ID for our own peer
            1. Can be hash of ip address + Nonce
            2. Hash starts with x leading 0s
        ii. Peers initialization
            1. Can be defined statically as a first approach.
        iii. Peer choosing module
            1. Peers list
            2. ID Validator
                a. Check amount of leading 0s
                b. Check that hash corresponds to ip address
    b. Announce module
        i. Communication module
            1. Class where we can send a message to another peer
    c. Notify module (Kyryll)
        i. Communication module
        ii. Receive "Gossip Notify", manage them and
        iii. Send "Gossip Notification"
        iv. Cache of messages
    d. Messages (Kyryll)

9. Security
    a. Important only with choosing peers
    b. When we listen to "Gossip Announce", do we listen to the announcement of all our peers? Yes.
    c. When the cage is full we have to decide what we want to remove.
        i. We have to have defensive mechanism vs spamming messages