Purpose

In order to increase visibility of current supply chain system we want to build a peer to peer supply chain system that can share shipment information between all parties that are involved in the shipment. The system uses peer to peer model mixed with a small part of server centralized model.

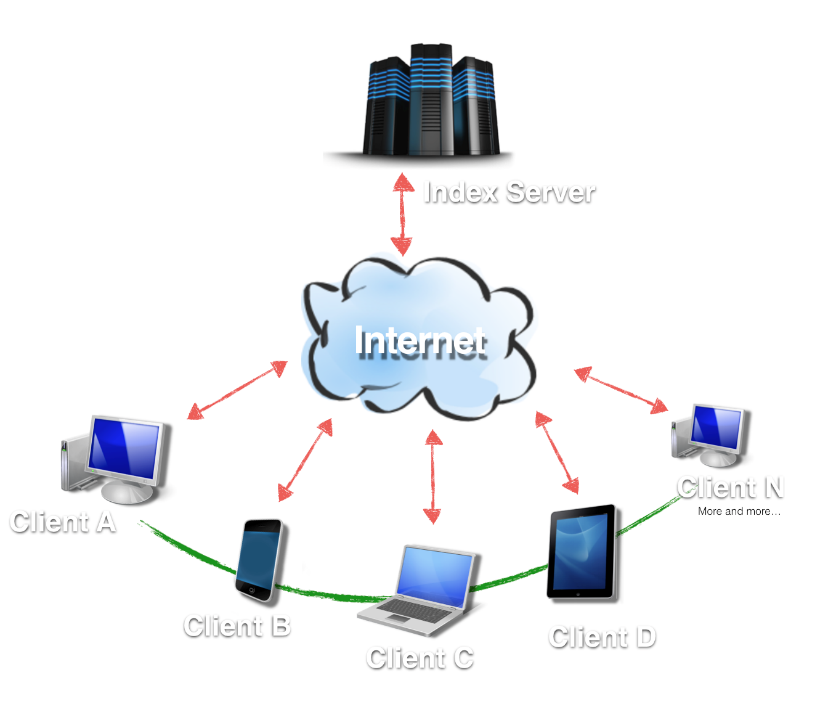
Architecture

This system semi- peer to peer model. This model is based on a particular transaction. There are mainly four entities involved in this model (a particular transaction).

1. Customer
2. Carrier
3. Supplier
4. Index server

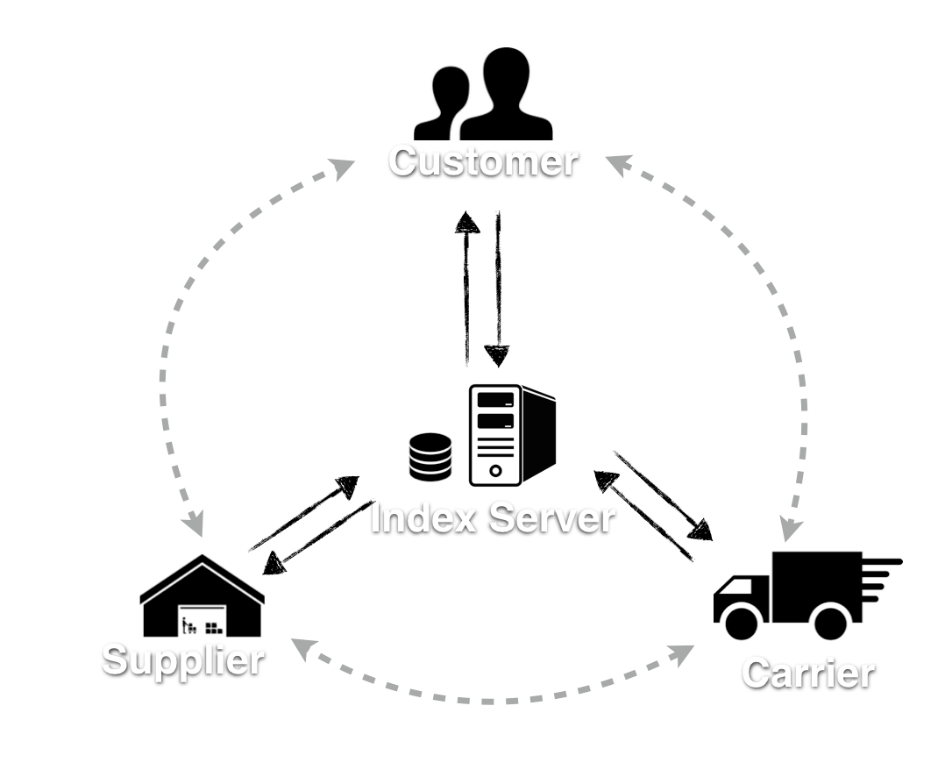
The rolls in the model do not represent the kind of client software. Customer, carrier and supplier is the roll that a client plays in one particular transaction. Index server holds all clients’ information. It play a roll of yellow page. Every time before a communication takes place client has to query index server for others’ IP address.

The overview of the model is shown in the following picture.



Index server acts as a yellow page. It holds all clients’ information such as IP address, clients’ names. When a client wants to communication with others, it will first query index server for the information of others’. Client will update their IP address regularly.

Each client will act as customer or carrier or supplier. The communication procedure is shown on the following picture.



This model is based on the assumption that the order information has already been initialized by each company’s ERP (enterprise resource planning) and the order information is already in our MongoDB. The order information is shown in appendix. The communication is initialized by the supplier. The supplier will establish a TCP connection with customer and carrier. It will update the shipment information to carrier and customer when a new signal arrives. The signal is sent by some integrated sensor network such as bar code scanner. Our software will broadcast to all parties that involved in the shipment, once it receives the signal. It will keep update the status until carrier picks up. The carrier will receive GPS signal from the track and it will use the same mechanism to broadcast to other parties. The customer will be required to verify the status of the shipment and manually submit the final acceptance of the shipment once the order is delivered.

There are few problem needs to be addressed in the communication between clients. First we will need the IP address of all parties that involved in one particular shipment. Because of changing IP address (we cannot assume the IP address of every client does not change, it has to be update), we cannot store all the IP address in each client’s database, so we added a light called index server in to the system. Index server acts like a yellow page. It stores all clients’ information such as IP address client’s name. Client queries index server about others’ IP address according to the information contained in Order struct which will be explained later in this paper. There are few advantage of adding one index server in to the system. Firstly it can reduce the amount of network traffic. one client has to send IP address update message to all clients’ in the network if every client stores each other’s IP address in its own database, instead of doing this, we have index server to handle all the client information related communication, so client only have to send one message to index server to update its’ information. Secondly this structure can help protect the privacy of each company’s information. Company’s information will only be accessible by index server and the parties that are involved in one transaction. Other parties can only access the information while a shipment is taking place. The information will not be stored in any place other than index server.

Peer to peer

This system uses peer to peer architecture for the communication between clients. Peer to peer architecture can reduce network traffic , since communication are between clients, there is no server interfere in between the communication. Using peer to peer architecture can avoid single point failure, even one client goes down the communication will continue. Once the client resumes online it will receive the newest information from other clients. In addition it can reduce data storage cost

//reduce data storage

//congestion control

Client design