Hybrid Peer-to-Peer Architecture in Support of Supply Chain Visibility

Zhijie Li 1, Haoyan Wu1, Alvaro Esperanca1,

Jeffrey Tazelaar2, John Wassick2, Zina Ben Miled1

1 Electrical and Computer Engineering Department, Purdue School of Engineering and Technology, IUPUI

2The Dow Chemical Company

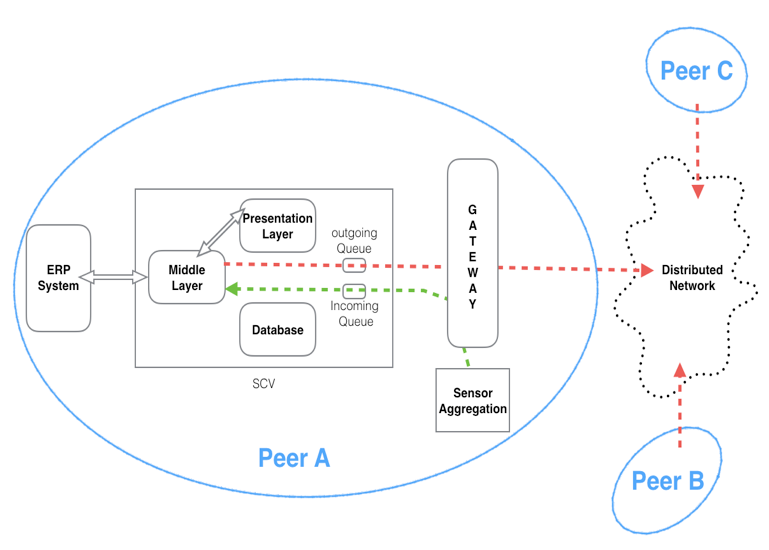
## *Abstract*

During the process of supply chain system, it is significant to distribute information among each stakeholder on time, which is laggy nowadays to some degree. The scalability is limited by the non-expandable design that cannot handle extra information when threshold exceeded. In order to address the historical problems, the new system has been implemented with such technologies to brighten the each process of supply chain, to enforce the privacy protection of each transaction, to boost the scalability of the system: 1) Real time broadcast to each stakeholder, 2) Encrypted communication, organic connection and local no-SQL database, 3) Centralized P2P network, standardized protocol. We present a prototype of the supply chain visibility system with web-page based user interface above an application layer implemented by golang and javascript.

**Keywords**: supply chain visibility, centralized P2P network, standardized protocol, organic connection, privacy protection, golang, noSQL database, javascript

## Introduction

A supply chain can be defined as a collection which is comprised of three or more organizations directly connected one or more of the flows of products, services, finances, and information from a source to a customer. [1] E-commerce spearheads the new trend of business. Following the current where it leads, supply chain management has been pushed to be more electronically. Current supply chain management system is known as Supply chain operating network (SCON). It, which is the standard used in all industries, relies on value added network (VAN)[2] which hired by companies to facilitate data’s exchanging as a private network provider and VAN’s function called electronic data interchange (EDI)[3]. It is expensive to set up VAN, a network that does not have a scalable structure. VAN was great proposal when public internet was not wildly available in the world, while the setting-up of VAN costs significant compared with public internet now. A large business company might afford the cost, but lots of small businesses are not able to set their own VAN. Thus, here comes out a feasible and solvable strategy using public internet to establish the communication between business partners. EDI is a transaction based data transfer method that conveys data from one computer system to another with a standardized format. In a supply chain management system, one party does not send updated shipment status to another until a certain transaction has been done. Furthermore, it is hard for one particular company to follow multiple EDI format since there are a lots of format standards. Nowadays, the scale of business surges up dramatically so that VAN and EDI cannot accommodate the business demand any more. Compared with old method used in document transfer in business, EDI and VAN have certain advantages, such as faster data transfer, less human intervention and more automated workflow.

This paper is intended to introduce a system called supply chain visibility (SCV) [4]. It is a system that can increase the visibility of current supply chain operating network by using a hybrid peer to peer architecture. The architecture of the system is shown in the figure 1.

Compared with traditional supply chain network, SCV delivers data faster since it is event based, and transfers data in real-time. It is also affordable to set up, because SCV uses the public network where end user does not need to spend extra cost to set up their own VAN. SCV uses a network that is called supply chain network of network which means that we can build a network over an existing network only for certain communication. The network of network is organic and shipment oriented which will be explained in the rest of the paper.

## Related work

There are some existing supply chain systems in the market such as E2Open and SAP. On one hand, these systems provide some well-developed functionalities such as real-time update shipment information, compatible with multiple enterprise resource planning (ERP). On the other hand, they are not affordable for small businesses. In addition, these systems are all cloud based (server centralized). Compared with a peer to peer based system, server centralized system is less scalable and has higher risk of single point failure. SCV is a system that uses peer to peer architecture and more flexible and affordable system. Any business/companies can use it. It is also scalable which means it can fit in any size company.

Figure 1

## Sensor aggregation and internet of things

As we moved to 21st century, an era that filled with sensors and smart devices, Internet of Things (IoT) has become the new trend of data shearing and has already been implemented in supply chain management. IoT is the network connecting every device/sensor with an on/off switch. It gives us the ability to see the real-time information of the world [5]. In order to achieve real-time data updating, the concept of IoT became the cornerstone of the SCV system. We assume there is a sensor aggregation in the operation environment which gathers all sensor data and forwards to a gateway from where the data is sent to SCV. The incoming data triggers a new event in SCV, then SCV processes a series of work corresponding to the event. As you can easily tell, that the system is an event based system using a sensor aggregation.

## Peer to peer architecture

The only centric server plays as an index directory that provides the storage and searching function about the information of the clients including their names and IP addresses to identify them in the network. Napster was a pioneer of this model [6]. The clients would access and retrieve the information they need from the server with their unique identification to establish connections with other clients.

From peer to peer perspective view point, a client can also act as a peer. Each peer plays as a client and/or a servers as needed at same time while data has been exchanged with other peers. Peer to peer model is more popular [7] than others in such aspects:

A. Ascendancies over client/server model

The obvious advantages that P2P model brought to us are boosting network traffic efficiency, subsiding the probability of occurrence of single-point-failure and saving the cost on building and maintaining server. Furthermore, P2P model overcomes client/server model on a branch of sides as,

(1). Privacy

For client/server model, data is always saved on servers where the data is easily attacked and hacked, even though data is encrypted. In contrast, data localization in P2P model provides a good way to keep high privacy in business transactions. There is no centralized storage for clients’ data. No one in the system has the global view of the whole system, and a peer is only able to communicate with others who are involved in the same transaction. Therefore, the feature guarantees that the exchanging of valuable business information will be circumscribed within the related stakeholders to protect the privacy of business activities.

(2). Scalability

The system has huge potential to scale up to meet future needs, nevertheless it is based on centralized model. The efficiency of responding to clients does not depend on how many clients have been registered in the server. Because the pairs, which include the unique IDs, the name and the IP addresses of clients, do not occupy too much space to be saved in the index server and they are easy to search with indexing in NoSQL database.

(3). Fault tolerance

Due to the localization of data, single-point-failure of index server, which provides looking up service [6], in centralized model will not have any effect on business data. The system works flawlessly, once the communication has been established between clients, even a failure happened to index server. Serval backup index servers could fix this failure efficiently by selecting one from them simply as a primary server which has the same content as the one that is out of service. If the failures come to peers, they would catch up the transaction process immediately by retrieving information from other transaction related peers when clients service return to normal.

B. Distinction with Pure P2P model

Gnutella (reference) was the first pure P2P network which was knowns as the file sharing function. It has high fault tolerance, because one failure will not have any effect on others. However, software like Gnutella, in their network, each peer keeping their own index may cause flooded queries in the network that consume significant network bandwidth when users search [6]. So comparing these two models from this aspect, centralized model which used in the system won the game. One peer connects to another one directly with the IP address of connected one whose information is retrieved from index server, then the peer queries data. The process avoids the network overlay that happens during indices being located.

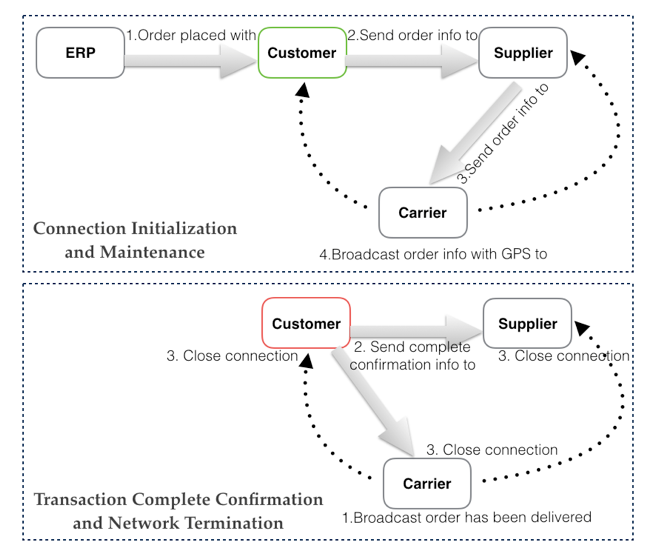
C. Difference with hierarchical model

In hierarchical model, instead of keeping by index server, the indices of nodes are managed by super nodes that connect to it. There is no doubt that the hierarchical model may more extendable than the centralized one. When speaking of complexity, however, the model needs more sophisticated algorithm to select super nodes, which are vulnerable. The hierarchical model that enables fast file transmission is unnecessary for this system where only some small-sized data are exchanged. Thus, the hierarchical model is not a good option for the SCV system.

## Hybrid P2P networks based on dynamic users

Essentially, the system itself relies on a hybrid P2P network which coordinates two kinds of network model, which are pure P2P and client/server model as discussed. With the flawless combination, the hybrid network inherits the advantages from both models. The design of centric index server avoids the flooding in pure P2P network, and the features of P2P enable the possibility to scale and inhance the ability of fault tolerrence. The network of the system is also dynamic. It is created by the customer who placed an order, and terminated when a transaction has been completed. Some researchers have developed a strategy building ephemeral social communities to save people from emergency. It establishes connections using P2P model and gathers people who link their social network account to the application where disaster happens, then ends when everything is clear. Whistle[8], as it is called, conveys the important information to thoes people. Similarly, the system has been built with same network, but the network runs dynamically upon the stakeholders in each transaction. The process of each order contains two steps which are showed in figure 2,

First, SCV builds and sustains the connection. Second the customer confirms the order and then SCV destroies the ephemeral network.



Figure

## Physical Distribution

The physical distribution phase of supply chain management is known as the transfer of good from supplier to customer. The phase starts when a supplier receives an order from customer and starts preparing the goods. It ends when a shipment is delivered. SCV addresses the visibility of this particular phase. Enabling an event base system will be a reasonable way to increase the visibility of this phase. SCV start getting sensor data when supplier begin the preparation. A sensor could be a scanner or a bar code reader. Carrier will take over when goods is picked up by a carrier’s truck. GPS signal will be generated every certain period when the goods is on the way to customer. Customer will confirm the condition of the goods once it is delivered. As you can see that the information of goods is updated in real-time. The technical detail will be discussed in the next section on how the information is shared between each stakeholder.

## Solution Overview

SCV is based on a hybrid peer-to-peer architecture that is customized dynamically for each stakeholder. The SCV rests on a collection of purpose-centric customized networks that can be configured dynamically on the fly. This is a departure from the traditional transaction based EDI systems. SCV allows stakeholders to share information related to a given shipment in real time. SCV can establish a peer to peer network based on a shipment. The network will be destroyed after the shipment is delivered. In order to facilitate dynamic IP address, an index server is brought into this system. It is responsible for storing all clients’ IP information including IP address.

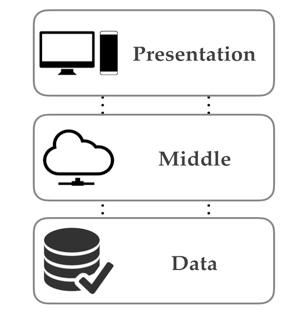
The client side software is unified. The role of a particular client can be changed depending on each shipment. The client side software uses a three tire design as shown in the figure 3 below.

Figure 3

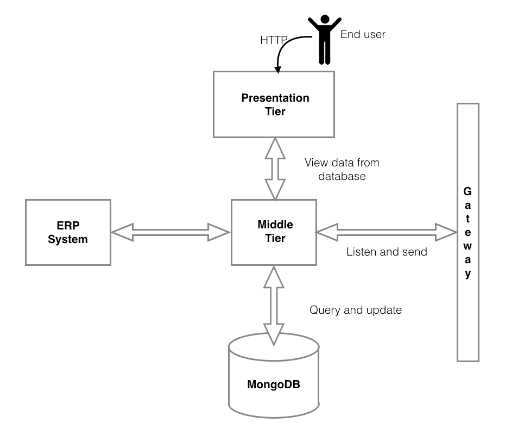
The sensor aggregation mentioned before turns the system to an event based system. SCV generates a new event according to a particular sensor signal. All clients’ actions are based on some certain events.

## Roles and Configurations

As we mentioned before clients do not have a specific role. The role of a client is assigned based on a particular shipment. Obviously, a given entity may assume more than one role over several shipments. For instance, a carrier in one shipment can also be a customer in other shipments. However, for a given shipment, these roles are in general distinct. There are four main stakeholders for a given shipment. These roles consist of customer, carrier, supplier and index server. Supplier is the stakeholder that initiates a new shipment process. A supplier will get the goods information from its own sensor aggregation. According to the sensor data, it will generate appropriate shipment message and broadcast to customer and carrier. Supplier will remain listening after the goods have been picked up by the carrier. Customer listens to other stakeholders and update its own database at most of the time. Carrier is the one that transfer the shipment from supplier to customer. Generally, there could be more than one carrier involved in one shipment. A customer only has to confirm the delivery of one shipment manually once the carrier has announced the shipment has been delivered.

## Modules

The system has two software modules, the client and the index server. The client side module uses three tire design. As shown in the following graph.



Figure

The first tire is a presentation tire. This tire represents the user interface. The presentation tire consists of a local HTTP server hosting a web application. The HTTP server is part of the middleware and is responsible for processing the user requests and calling the functions based on the requests. The presentation tire is built with some web development standards; namely HTML, CSS and JavaScript and Go programming language. The second tire is middle tire which is called middleware that is mainly responsible for four things, processing client request from presentation tire, receiving signal from sensor aggregation, generating new event and sending updated information to other stakeholders involved in one shipment. Middle tire uses a gateway and sensor aggregation to communicate with sensors as shown in figure 1. Sensor aggregation is a collector of all sensor data, it sends all sensor data to the gateway. Different sensor data triggers different events, for example a GPS data from a truck can cause a new GPS update event, the middleware updates database and send the newest geolocation information to other stakeholders. The third tire is called data tire. Database is the main software used in this tire. SCV uses mongoDB as its database.

There are certain advantages of using the three tire design. Firstly, it is more flexible, since these tires do not have to be on the same physical computer. User can split these tires onto different computer. Secondly, this design gives more security to SCV. The database can only be accessed by the middleware, an authentication mechanism will be added to the connection between middleware and database also in the future development.

## Data exchange template and management

SCV uses JSON format to share shipment information. We could choose data format between JSON and XML. XML is good for document transfer and it is easy to interoperate. In comparison with XML, JSON has more advantages. Firstly, JSON is at least easy to interoperate as XML. Secondly, XML is simpler than some other markup languages, but JSON has a much smaller grammar and maps more directly onto the data structure used in modern programing languages.

MongoDB is the database has been used in SCV. It is a no-SQL database. In comparison with SQL database, mongodb has simpler structure. Instead of using two dimensional table structures, it uses document based system. Each collection holds different document. [9] Numbers of fields and size of documents can be differed from one to the other. Every document in mongodb uses JSON format as well, which means that we can easily parse data from/to database without any extra work.

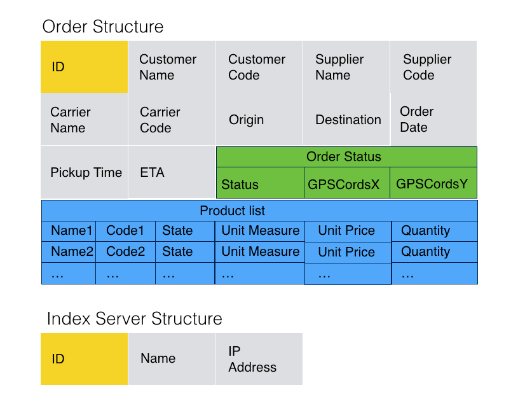
The data exchange template we used is fairly simple. There are mainly two kinds of information that we need to store in SCV, the shipment information and the clients’ information. These information templets are shown the graph below in figure 4.

Figure 5

As shown in the graph. You can see that order struct consists of few different smaller structs and fields. First there are some information about the stakeholders included in a shipment. The reason of grouping these fields is to find the IP address from index server, since we assumed that the IP address is going to be dynamically assigned to each client. Secondly, some production information is included. Thirdly, there is as small struct called "OrderStatus". This struct contains the status information of a particular order including geolocation information and the status description. The status description is only written for testing purpose. It will be re-implemented with more detail in the future development. Finally, we talk about the order code, which distinguishes one from another. For the sake of simplicity, we used the object code generated by mongodb.

The client information stored in index server is simpler than shipment information. It only contains three fields, ID, name and IP address. Clients will update their IP address with their name and ID.

## Example Scenario

In order to test SCV, we set up a simple scenario. First we assume that the order has already been committed by those stakeholders included in the shipment. SCV comes out at the beginning of physical distribution. The supplier initiates the entire process by announcing the status of the preparation of the shipment. In our test we descript it as five stages” "on the ship", "off the ship", "on the dock", and "in the storage” "getting ready", "ready for pickup" (we assumed that the goods is coming from a ship). The supplier side SCV listens to any incoming signal. A new event is triggered, once a new signal is received from sensor aggregation. The supplier will update its own database and broadcast the newest information to other stakeholders who are involved in the shipment. The carrier will arrange a truck to pick up the shipment when it receives the "ready for pickup" signal from supplier. As soon as the truck picks up the goods, a new "picked up" signal will be generated by the truck and sent to supplier and customer both. We assumed that there is a GPS sensor installed on every carrier’s truck, and the truck will keep sending GPS signals to carrier and carrier will broadcast the signal to supplier and customer. We defined this particular status as "in transit". The status will not be updated while the shipment is in transit, but the GPS information will be shared between all stakeholders using the status struct defined. The carrier will send a delivery signal when the shipment is delivered to customer. The customer has to confirm the shipment manually. The reason is that customer needs to check if the goods under the shipment is in acceptable condition. The customer will accept/reject the shipment after they have checked it. Customer will manually confirm the status through SCV. At the end the shipment, data will remain in all stakeholder’s database for future using.

## Conclusion

Overall, the current supply chain systems, which uses electronic data interchange and value added network, are not suitable for current business environment since the size of companies grow larger and the information exchange process needs to be more efficient. In another word, the supply chain visibility needs to be increased by using different techniques. In order to accommodate the needs of current supply chain visibility, we developed the system called SCV.

SCV is a hybrid peer to peer and event based system. It is low cost, more affordable setup, scalable and more flexible compared with electronic data interchange and value added network. The system has the ability of real-time event update. There are still some developments need to be added to SCV. In order to make the system more secure, we need to add some security mechanism to the communication part. The database also needs some secure login mechanism as well. Overall the system demonstrated the method of satisfying the basic need of current supply chain system.

## References

[1] J. Mentzer, W. DeWitt, J. Keebler, S. Min, N. Nix, C. Smith and Z. Zacharia, "DEFINING SUPPLY CHAIN MANAGEMENT", Journal of Business Logistics, vol. 22, no. 2, pp. 1-25, 2001.

[2] B. Jerman-Blazic, "Security in value added networks - security requirements for EDI", Computer Standards & Interfaces, vol. 12, no. 1, pp. 23-33, 1991.

[3] The EDI Handbook, edited by M.Gifkins, D.Hitchcock, Blenheim Online, London, 1988, p.4

[4] N. Tohamy, L. M. Orlov, L. Herbert “Forrester Research: Supply Chain Visibility Defined”, 2004.

[5] "IEEE Internet of Things Journal", IEEE Internet of Things Journal, vol. 3, no. 3, pp. C2-C2, 2016.

[6]Chopra D., Schulzrinne H., Marocco E. and Ivov E. (2009). Peer-Peer Overlays for Real-Time Communication: Security Issues and Solutions, IEEE Communications Surveys & Tutorials, Vol. 11, No 1, First Quarter 2009.

[7] A. Crespo, et al., “Semantic overlay networks for P2P systems,” Technical Report, Computer Science Department, Stanford University, pp. 23-32, 2002.

[8] Y. Jung, R. Figueiredo and J. A. B. Fortes, "Emergency Response using Ephemeral Social Communities across Online Social Networks", EAI Endorsed Transactions on Collaborative Computing, vol. 1, no. 5, p. 150805, 2015.

[9] S. Kanoje, V. Powar and D. Mukhopadhyay, "Using MongoDB for social networking website deciphering the pros and cons", in *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference*, Coimbatore, 2015.

http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7192924