Ideas on reference models, formal languages and communication design patterns in a physical internet

Jean-Yves Colin, Moustafa Nakechbandi, Hervé Mathieu

Ideas on reference models, formal languages and communication design patterns in a physical internet

Jean-Yves Colin,
LITIS, Le Havre University,
5 rue Ph. Lebon, Le Havre, France
jean-yves.colin@univ-lehavre.fr

Moustafa Nakechbandi
LITIS, Le Havre University,
5 rue Ph. Lebon, Le Havre, France
moustafa.nakechbandi@univ-lehavre.fr

Hervé Mathieu
ISEL, Le Havre University,
Quai Frissard, Le Havre, France
herve.mathieu@univ-lehavre.fr

Abstract— **This work is an attempt to start to define more formal communication languages and interfaces between layers of a reference model of a physical internet. We first present the electronic Internet and OSI communication models, and then our NOLI model and the original OLI model of Montreuil 2009, and we compare them. Next, we present common communication "Design Patterns". Focusing on the interactions inside a specific node, we identify several types of exchanges. We discuss the usefulness of the presented communication patterns in the context of layers communication in a Physical Internet, and the types of exchanges identified.**

Keywords— Physical Internet, Digital Internet, Layered Reference Model, Pattern Languages, Communication Protocols

I. INTRODUCTION

The idea of a "Physical Internet" is based on the hugely successful "electronic" Internet. First proposed by (ref Montreuil), it supposes that it is possible to import some of the concepts and solutions used in the internet into the world of moving physical objects, to build an efficient, robust and sustainable global logistic network.

The internet itself is based on a Reference Model, named TCP/IP. The TCP/IP Reference Model is divided into a "stack" of 4 or 5 cooperating "layers". This layered approach is used to keep the complexity of the whole problem to a manageable level. Another Reference Model, the more theoretical OSI model, is divided instead into a stack of 7 layers, that cooperate in the same way.

Reference Models for a physical Internet follow the same concept, and they all propose a layered approach [7, 3, 5]. They also divide the problem into 7 or 5 layers. In all cases, the layers communicate data and transfer physical objects, or both.

Following the ideas of using "patterns" to tackle classical situations or problems, such a patterns in architectural or urban development [1], or design patterns in software development [10], a set of useful patterns for data communications in electronic networks is presented by Lascano and Clyde in [6].

We propose that some patterns also apply to the data and physical objects movements in a Physical Internet.

In this paper, we study the data and physical transfers between the layers.

We first present several Reference Models for a physical internet, and we compare them. Next we focus on the interactions between the layers, identifying four types of typical interactions. Then we present a set of communication patterns that are found useful in the case of data communications in electronic networks

We finally argue that they apply to most cases of interactions between layers of a Reference Model for a Physical Internet.

II. THE REFERENCE MODELS

Here, we briefly present our NOLI model [3, 4], the original OLI model of Montreuil 2009 [7, 8, 9], and the Reference Model of Hofman, Spek and Brewster 2017 [5], and we compare them and to the electronic Internet and OSI reference models.

All Reference Models are layered models. In a layered reference model, each layer is in charge of some functionalities. A layer at level N uses the services furnished by the layer N-1 below, and furnishes services to the layer N+1 above. Two layers communicate thanks to interfaces and follow well defined protocols. In a node "a", a layer N can only exchange data (and physical objects in logistic networks) with its contiguous layers N-1 or N+1 located on the same node (exchange of information inside the same node). Also, layer N of a node "a" can also potentially exchange information with the layers N located on other nodes. As an example, Figure 1 presents the structure of the NOLI Reference Model, with its 7 layers and some of the functionalities of each of its layers.

TABLE I. THE REFERENCE MODELS FOR ELECTRONIC AND LOGISTIC NETWORKS

TCP/IP Layer Name (Internet)	**OSI reference Model Layer Name (electronic network)**	**OLI Layer Name [7] (logistic network)**	**NOLI Layer Name [3,4] (logistic network)**	**Hofman, Spek and Brewster Layer Name [5] (logistic network)**
Application	7. Application	7. Logistics Web	7. Product	Customer
	6. Presentation	6. Encapsulation	6. Container	
	5. Session	5. Shipping	5. Order	
Transport	4. Transport		4. Transport	Shipment
Network	3. Network	4. Routing	3. Network	Logistic Network
		3. Network		
Network Access	2. Data Link	2. Link	2. Link	Corridor/Transport
Physical	1. Physical	1. Physical	1.Physical Handling	Physical Infrastructure

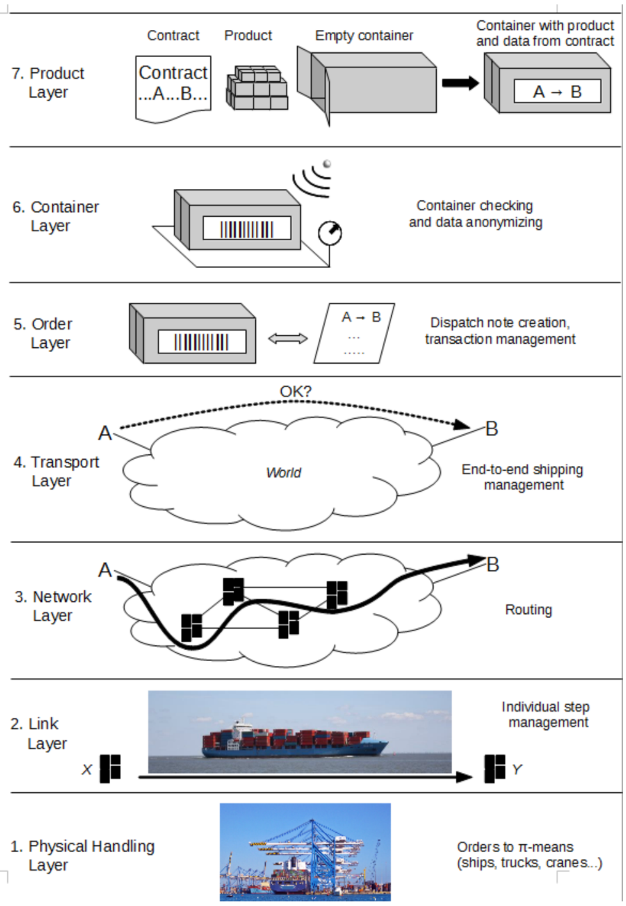

Fig. 1. Example the structure of the NOLI Reference Model.

As indicated above, the NOLI model mirrors the OSI model with its 7 layers, and so do the OLI model. The Hofman, Spek and Brewster Reference Model [5] mirrors the TCP/IP Reference Model and the functionalities of its 5 layers, so its top layer covers the functionalities of the 3 highest layers of the OSI and NOLI. The NOLI model also differs from the TCP/IP, the OSI, the OLI and the Hofman, Spek and Brewster models in that it does not locate all physical objects definitions in the lowest layer, but instead dispatches them into several layers according to the needs.

Table I presents a synthetic view of the layers of the above Reference Models and the global structure of each one when compared to the others. One can note that the reference models presented below follow either the OSI Reference Model that has 7 layers (OLI and NOLI), or the TCP/IP Reference Model that has 5 layers (Hofman, Spek and Brewster). However, for example, the OLI and NOLI Reference Models differ in the functionalities that are covered by their respective 3, 4 and 5 Layers models.

III. Common Communication Design Patterns

Next, we present the common communication "Design Patterns" for electronic networks proposed in Lascano and Clyde [6]. The following presentation in this part is based on [6].

1. Request-Reply: This pattern is for simple exchanges either between two layers in the same node, or between two layers at the same level in different nodes. The exchange is fast but there is no strict control or guaranty, thus losses of messages may be harder or slow to detect.

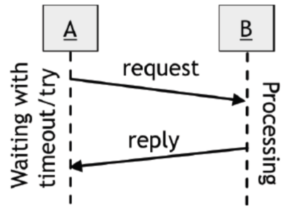

Fig. 2. Request-Reply Message Sequence..

2. Request-Reply-Acknowledge: This pattern builds on the Request-Reply pattern to add some guaranties in the exchanges. Losses of messages are easier to detect with this pattern, but the whole exchange is slower than a simple Request-Reply.

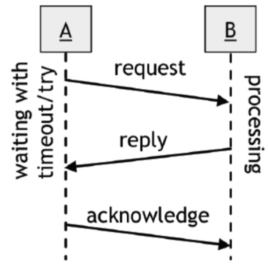

Fig. 3. Request-Reply Message Sequence..

3. Idempotent Retry: Like Request-Reply, this pattern is used when one side requested data only with no "border effects", and this side must try the request again because a problem occurred with the first request. This supposes that no data was changed by the request or the retry. Thus this kind of retry can be used several times without troubles.

4. Intermediate State Messages: To solve the problem of some actions taking a long time to complete, thus implying long running exchanges, this pattern allows a side to send intermediate messages that inform the other side of the current state of the side sending them (see Figure 4).

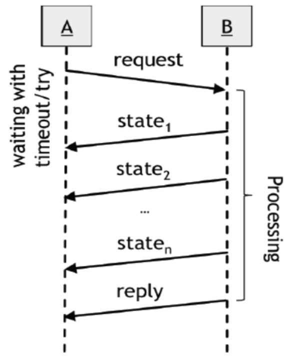

Fig. 4. Intermediate State Message Sequence.

5. The Second Channel pattern: This pattern is for situations involving significant amounts of data transfers. Because the large data transfers can delay intermediate state messages, this pattern's solution suggests opening a second communication channel between A and B that is dedicated to data transfer, leaving the original communication channel available for intermediate state or control messages, as it is shown in Figure 5.

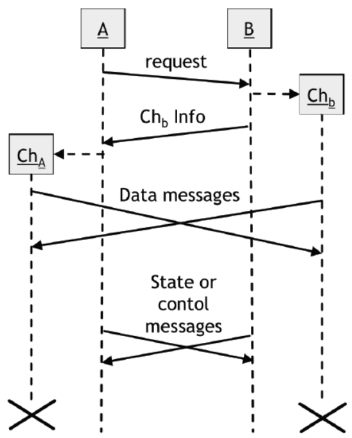

Fig. 5. Second Channel Message Sequence.

6. The Front-End pattern: This pattern addresses the problems of making the location of shared resource transparent to the client, allowing the number of resources to change dynamically. It has a resource client send requests to a front-end process that automatically redistributes them to appropriate resource managers, B processes. After processing the request, a resource manager replies back to the client directly (see Figure 6).

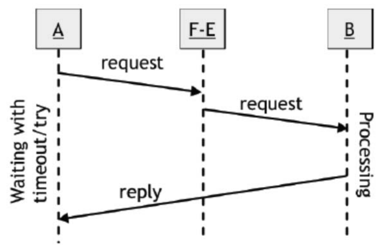

Fig. 6. Front-End Message Sequence.

7. Proxy pattern: This pattern presented in Figure 7, introduces a process between a resource client and a resource manager. However, the intermediate process, called a proxy, serves other functional purposes besides re-distribution of the requests, for example it may provide authentication, access control, audit logging, and data transformation functionality. Also, the resource manager returns replies through the proxy to client, completely isolating the client from the resource manager.

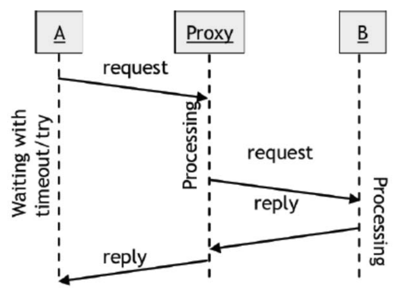

Fig. 7. Proxy pattern Sequence.

8. The Reliable Multicast pattern: This pattern builds on the multicast idiom to provide reliability and synchronization among a group of processes. Its solution is a protocol that starts with a process A sending a request message to a group of process, B= $\{b_1, .., b_n\}$. Each process b_i sends a reply back to A when it receives the request and is ready to process it. After A receives reply from all B processes, then A will multicast a go-ahead message back out to all B message indicating that they can proceed with the processing of the request, shown in Figure 8. In this way, the execution of the request is synchronized among all of the B processes. If A fails to receive a reply from every B process, it can resend the request to some or all of them until it gets a reply from all of them or terminates the conversation as failed. This pattern focuses on providing strong reliability and synchronization, but can also help with scalable distribution of resources.

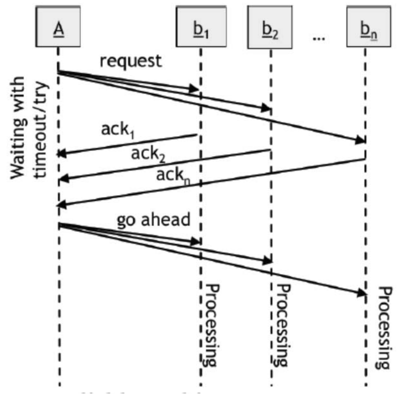

Fig. 8. Reliable Multicast Sequence.

9. The Publish-Subscribe pattern: Finally, this pattern tries to decouple message senders (publisher) from message receivers (subscribers). An intermediate process acts as a store-and-forward buffer for message transmission with the capabilities for managing subscribers and delivering individual message to multiple subscribers.

Additionally, Lascano and Clyde in [6] identify two additional operations to allow patterns combinations:

1. Sequential Composition

2. Nested Composition

IV. Application to Logistics Systems

Of course, all the above patterns apply to the electronic part of all logistics systems. Here, we review these patterns in the context of moving physical objects in a logistics system such as a Physical Internet.

1. Request-Reply: the pattern can be applied directly to the physical part of a logistics system. For example, an electronic request for an empty container of a specific kind may be send to a lower layer. The Reply will then be an empty container, or a reply message stating that no container is currently available.

2. Request-Reply-Acknowledge: this pattern is a more secured version of the above pattern, and can be applied directly too.

3. Idempotent Retry: This pattern can be applied to the cases where the location, or the physical state of a container or of a mover is needed. For example, requests to get the internal temperature of a filled refrigerated container may be regularly sent. If an answer is lost, there is no problem with retrying by sending a new request. Another example is the physical printing of a connaissement of a container during its trip.

4. Intermediate State Messages: This pattern can be applied to the sending of a containers, with the added request that a message must be sent each time the container goes

through a formality. It can also be applied to the sending of a large shipment that must be divided into several containers. In this case, the sender may for example request that he must be informed when each container reaches each intermediate destination, to allow him to better plan the reception at the final destination.

5. The Second Channel pattern: some containers may have an higher priority than others and may be delayed to much by the presence of many low priority containers. In these cases, this may lead to having to use tools and means specifically dedicated to their handling. For example, a fast large x-ray container scanner may be dedicated to high priority containers see figure 9. As for the other patterns, this pattern may be applied to exchanges between layers at the same location, or to exchanges between identical layers at separate locations (for example, the renting of an additional external truck because of an unexpected urgent request).

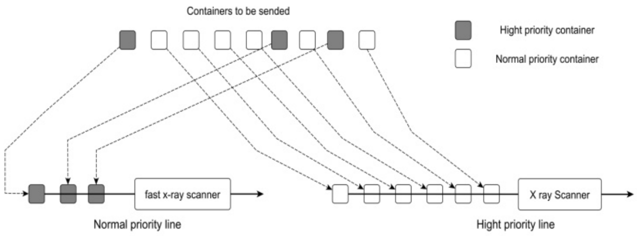

Fig. 9. Second Channel logistic pattern.

Although there are two obvious identified channels, namely the "physical" channel for the physical objects, and the "data" channel for electronic informations, one can note that it is very common in logistics systems to consider financial matters as belonging to a third specific "financial" channel, although this channel is in fact supported by the "data" channel.

6. The Front-End pattern: this pattern may be applied to the use of a broker for a simple request to get a resource. For example, to get some space, for a total of 4 containers, on a ship able to do a direct shipment between two specific ports.

7. Proxy pattern: one can note that the layers used in all reference models are all based on this pattern. That is, a layer at level N exchanges data (and physical objects here) with other lower layers through the intermediate N-1 layer that may thus be considered as a proxy according to this pattern. But this pattern also includes cases of using brokers for complex requests, such as the handling of customs formalities, etc.

8. The Reliable Multicast pattern: One example of possible use-case for this pattern is a situation in which it is necessary to have a group of containers traveling more or less together. For example, several special containers must travel with one specific "integrated logistic support" container that contains test gears, protections or spare parts. Although these containers may travel separately, it is logical to have regular synchronization points during the whole points so that they never get too separated. (cf. Fig. 10).

One can note that the multicast term used by Lascano and Clyde [6] does not match with this action, the term used in computer science for this action is rather fork–join.

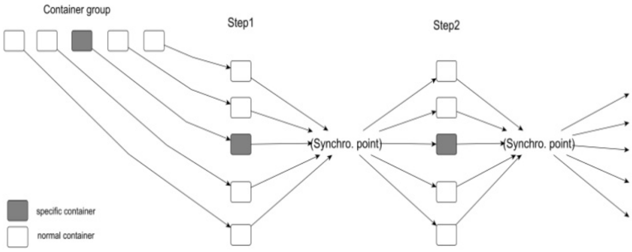

Fig. 10. Reliable Multicast logistic pattern.

9. The Publish-Subscribe pattern: Applied to a Logistics System such as a Physical Internet, this pattern is used when a separate location is useful to serve as a temporary storage buffer between a sender and some or all of the delivery points. This pattern is actually the one used by Amazon and other online stores when they deliver physical goods to an intermediate place, such as a general store, instead of delivering them directly to the customer see Figure 11. This intermediate place then stores the parcels until the appropriate buyer comes to get it, check the identity of this person and manages other formalities, and finally gives the parcel to the customer. This is also the same pattern used by Post Offices in the case of "Poste Restante" services.

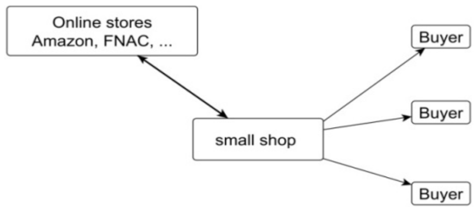

Fig. 11. Internet sales schema.

V. Using the Patterns to Express Interactions between Layers

To illustrate the use of the above patterns in Logistics Systems, we introduce a simple language to express common simple operations in this context

We identify two kinds of "objects": the physical objects ("P") and the data objects ("D").

We also identify three kinds of exchanges between layers: Request ("R"), Answer to request ("A"), and additional Control exchanges ("C").

An "exchange" between layers will then be expressed by the following form:

<OE, sending layer, receiving layer, #identifier : "infos">

in which:

- O is either P or D
- E is either R, or A or C,
- sending and receiving layers values identify the exchanging layers.,
- the #identifier is an unique identifier for this "exchange",
- "infos" includes all necessary informations, such as the identifier of the Request if this is an answer, the identifier of the physical objects involved, the kind of request if this is a request, etc.

Thus a <PR, 3, 2, #11 : "infos"> exchange is actually a physical request, from Layer 3 to layer 2, uniquely identified as #11. The "infos" filed includes what kind of request it is, the physical objects involved, etc.

We now present a first example of exchange between Layer 6 and Layer 5 about the handling of filled, closed, and duly validated containers, using the simple Request-Response pattern (cf. Figure 12):

- <PR, 6 ,5 ,#22 : "infos"> // Request with id=#22 from Layer 6 to Layer 5: the "infos" data field contains the description of the request, here the taking over of identified containers.
- <DA, 5, 6, #77, Re: PR, 6, 5, #22: "infos"> // Answer #77 to Request #22, from Layer 5 to Layer 6: "OK to take over" the containers.

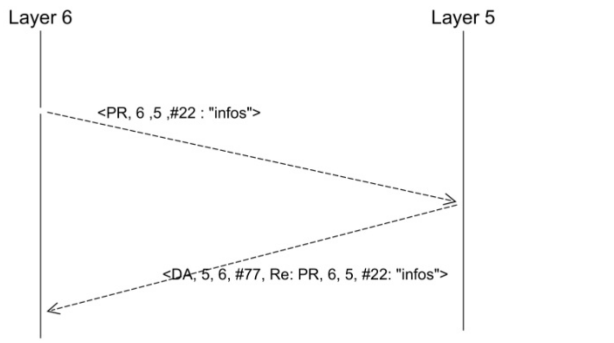

Fig. 12. Exchange between Layer 6 and 5. about the handling of some filled containers

In the second example, we use the more complex Intermediary State Messages pattern (cf. Figure 13) to start the shipment of a container and then keep track of the advance of a container toward its destination in Layer 4

- <PR, 4, 3, #344, "infos"> // Request with id=#344 from Layer 4 to Layer 3: the "infos" data field contains the description of the request, here the shipment of the identified container, with the added request that the location of the container must be regularly send.
- <DA, 3, 4, #557, Re: PR, 4, 3, #344 : "infos"> // Answer #557 to Request #344, from Layer 3 to Layer 4: "OK to ship" the container.

The container then starts to advance.

- <DC, 3, 4, #789, Re: PR, 4, 3, #344 : "infos"> // Control message related to Request #344, to signal the physical departure of container from it starting location

Once the first intermediate location is reached, a state message is send:

- <DC, 3, 4, #1003, Re: PR, 4, 3, #344 : "infos"> Control message related to Request #344, to signal the physical arrival of container to the first intermediate location

Then when the container leaves the first intermediate location :

- <DC, 3, 4, #1120, Re: PR, 4, 3, #344 : "infos"> // Control message related to Request #344, to signal the physical departure of container from it first intermediate location etc.

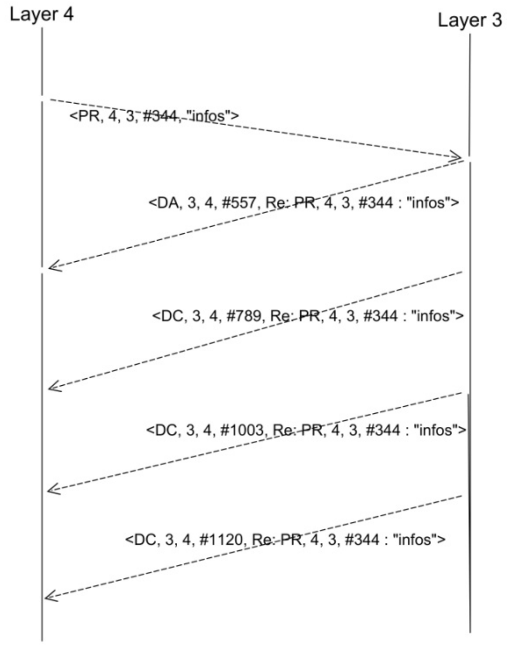

Fig. 13. Exchange between Layer 3 and 4 to start the shipment of a container and then keep track of the advance of this container toward its final destination in Layer 4.

The other patterns can be illustrated in the same way.

VI. Conclusion and Remarks

We presented patterns common in electronic networks, and we then studied them and their usefulness in the context of logistics networks, particularly when physical objects are to be moved or stored.

The above patterns may be useful in whatever the reference model will be used in the future, be it OLI, NOLI, or another one.

An interesting question then is to determine if there are new specific patterns that may be useful in logistics networks.

From a theoretical point of view, it seems probable that no new pattern is needed because there seems to be no specific situation that may happen in logistic networks that could not happen in an electronic network. However, patterns are not theoretical structures, but are more like commonly acknowledged "generic" solutions developed to solve problems that appear again and again. It is then perfectly possible that a common problem in logistic systems does not happen in electronic network often enough to trigger the development of its own solution pattern, so it seems to us that additional patterns may indeed appear in the future in logistics networks.

Additional points are still open, among others:

Is it necessary to enforce a Strict Layering approach or not? In a non strict layering approach, non-contiguous layers may directly communicate for efficiency reasons. However, this muddles the simplicity of the Layered approach;

Also, is the stack notion still usable when the various layers can be geographically separated? in electronic networks, this never happens, but it is not clear if this may occur in logistics networks.

ACKNOWLEDGMENT

This work is co-financed by the European union with the European regional development fund and by the Normandy regional council: Project CLASSE2 "Corridors Logistiques : Applications à la vallée de la Seine et Son Environnement", France.

REFERENCES

[1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, A Pattern Language: Towns, Buildings, Construction. New York: Oxford University Press, 1977.

[2] Chuanwen Dong and Rod Franklin, "From the Digital Internet to the Physical Internet: A conceptual framework with a simple network model" (IPIC 2018), 2018.

[3] J.-Y. Colin, M. Nakechbandi and H. Mathieu,"A Proposal for an Open Logistics Interconnection Reference Model for a Physical Internet", GOL 2016, The 3rd International IEEE Conference on Logistics Operations Management, 23-25 May 2016, FEZ-Marocco, 2016.

[4] J.-Y. Colin, H. Mathieu and M. Nakechbandi, "NOLI A Proposal for an Open Logistics Interconnection Reference Model for a Physical Interne", IPIC 2017, 4th International Physical Internet Conference, 2017. https://www.pi.events/IPIC2017/sites/default/files/IPIC2017-Workshop-2.2_presentations.pdf

[5] W. Hofman, J. Spek and C. Brewster. Applying blockchain technology for hyperconnected logistics, in: Fourth International Physical Internet Conference(IPIC 2017), 2017.

[6] J. E. Lascano and S. Clyde, “A Pattern Language for Application-level Communication Protocols,” presented at the ICSEA 2016, The Eleventh International Conference on Software Engineering Advances, pp. 22–30, 2016.

[7] B. Montreuil, "Physical Internet manifesto: globally transforming the way physical objects are handled, moved, stored, realized, supplied and used" Technical Report, 2009.

[8] B. Montreuil, E. Ballot, F. Fontane,’An Open Logistics Interconnection model for the Physical Internet’, Proceedings of INCOM 2012 Symposium, Bucharest, Romania, 2012.

[9] R. Sarraj, E. Ballot, S. Pan, Montreuil, B.’Analogies Between Internet Networksand Logistics Service Networks: Challenges Involved in the Interconnection’, The Journal of Intelligent Manufacturing 25, 6, 1207-1219, 2014.

[10] D. Schmidt, M. Stal, H. Rohnert, and F. Buschmann, Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects, Volume 2 edition. Chichester England ; New York: Wiley, 2000.