Inter-operating Co-opeting Entities

A Peer-to-Peer Approach to Cooperation between Competitors

Alain Sandoz
Informatics Institute,
University of Neuchâtel
Neuchâtel, Switzerland
e-mail: alain.sandoz@unine.ch

***Abstract*—The paper identifies conditions that enable competing business entities to interoperate through their mutual cooperation function while maintaining a strict separation of their competing functions, and in particular the secure operations of their core IT-business infrastructure. Implications on the architecture of the cooperation function and an implementation realized using the Kubernetes micro-service infrastructure and Hyperledger Fabric are presented.**

Keywords - horizontal cooperation; co-opetition; peer-to-peer network; micro-service architecture; Kubernetes; Hyperledger Fabric.

I. INTRODUCTION

This short paper describes work in progress started in 2017 that has led our team to design and implement generic mechanisms to enhance interoperability among distrustful actors. The resulting *peer-to-peer* configuration enables private entities (i.e., competitors) and/or public entities (i.e., regulators) to interoperate under good conditions where they are willing or compelled to collaborate. On the other hand, functions that are too sensitive to be exposed to information leakage or tampering from an entity's environment are kept safe (up to the level of safety provided by each entity for its own resources).

The initial project was meant to speed up and improve information flow between the many actors of the milk production and processing sector in Switzerland: farmers, transporters, label organizations, laboratories, the dairy industry (buyers, transformers, retailers, exporters), regulators, and of course the end-consumer, all require information in a maze of formats and temporalities. Often, the complexity of data-management and the lack of synchrony between data-flows and the actual logistics of production along the value chain prevent improvements or paralyze processes. Even competing entities were willing to work together to overcome difficulties, i.e., cooperate.

Our work on behalf of the milk sector delivered mechanisms that apply to other sectors of the economy or of society, including, e.g., banking, insurance, and healthcare.

The results we describe pertain to specific business conditions called co-opetition, together with specific technical conditions that are found in distributed systems, in particular, but not restricted to peer-to-peer (P2P) networks.

The paper is structured as follows. Section II briefly describes the state of the art from where we start. In section III we define the *cooperation function* and the *coordination function* in the context of interoperability. In section IV we state a small set of conditions, or *principles*, for the digitalization, integration, and interoperation of the cooperation function by and among co-opeting entities and define the architecture of a P2P network that operates the cooperation function. In section V we describe a productive implementation of the concept. We conclude in section VI with implications on co-opetition among software-providers and a possible transformation of some regulatory tasks currently implemented *de facto* in the form of centralized coordination functions among networks of competing economic actors.

II. STATE OF THE ART

This section briefly describes the notions of co-opetition, peer-to-peer collaboration, and cooperation function.

A. Horizontal cooperation, co-opetition

In business, many situations arise where competitors must cooperate to sustain their access to the market, reduce costs, or collectively realize positive conditions that would be impossible on an individual basis. This was first described in [1]. It happens e.g., in logistics and transport [2], in industry [3], in banking [4], and is generally called "horizontal cooperation" or co-opetition [5], as opposed to the master-slave-type of dependency between customers and suppliers in a vertical setting, or the possible cooperation of business entities that are not competitors.

Co-opetition is a sensitive endeavor, where cooperation between competitors on some specific function is *beneficial*, whereas the core business goal of each party in the cooperation remains *domination* of the other(s).

Since cooperation implies the sharing of resources, e.g., at least of information, to manage which resources can be shared for mutual benefit without compromising individual survival is delicate.

B. Peer-to-peer networks

On the other hand, peer-to-peer collaboration in computer networks is a well-established practice for groups of otherwise autonomous entities to share commonly valuable resources [6], [7]. Shared resources may be files,

 ISBN: 978-1-61208-785-6

computing power, voice over internet protocol (VoIP) [8], partial solutions to broken-down problems, storage, etc.

An interesting aspect of P2P networks is that parties to the P2P activity usually have no additional interaction. Their common interest and benefit can be to swap multimedia resources, solve together a genome or signal-processing problem, take part in an elaborate mesh-based resource sharing configuration [9], participate in multiplayer games [10], or even provide "services" as in [11], but in general, stakeholders in P2P activities do not otherwise interact.

The absence of "external relationship" is not necessary in P2P networking, nor does the latter *a priori* exclude competition (in the business sense) between peers.

Though competition between stakeholders in a peer-to-peer network is not excluded, it does require some caution: P2P activities imply that peers execute foreign operations on their computer infrastructure, or, to be more precise, some computer infrastructure that they control. Therefore, certain conditions should be guaranteed to stakeholders before they open-up their strategic resources, information, processes, or core-functions like customer relationship management systems (CRM) to "peers" that are, in essence their enemies.

C. *Cooperation and coordination functions*

The functions shared between co-opetitors are called *cooperation* functions and *coordination* functions. They are usually implemented on the IT infrastructure of a central actor, e.g., SWIFT [4]. For many reasons including trust and fault-tolerance, centralization of any function should however be avoided in P2P networks, the more so if peers are inherently distrustful of each other.

The approach we follow is *fully distributed*: there is no central component and every peer operates and executes every function that it requires on its own. Execution of functions is traced and logged, and *correct behavior* can *always be proven* unless the conditions for consensus (among misbehaving peers) are met and used to disqualify an honest peer. This is about the best that one can reach under the general conditions of distributed computing systems.

III. COMPETITION, COOPERATION AND COORDINATION

Two important features of co-opeting entities in the traditional business environment are [12]: *1)* the separation of the cooperation function from the core business functions within each of the co-opeting entities, and *2)* the presence in some form of a coordinating actor.

The first feature is easily understandable for the security of the core function of the business, but is also related to the different social and relational skills of "competitor-" and "cooperator-" types of workers in any given company.

Figure 1 illustrates the situation of a set of competing entities CE-***1***, CE-***2***, … CE-***N***, that compete to access their share of the market, whereas they each operate a cooperation function clearly separated from the core functions of the business with defined access conditions. A *coordination function* is necessary to establish consistency among entities with regards to cooperation. Of course, cooperation among entities in complex business environments like global transport or banking requires a computer-supported coordination function. As in [4], this function is often complex itself and is centralized, which enables strong semantics of transactions (e.g., non-repudiation of bank transfers in SWIFT). In this case, the coordination function might be owned and operated, or at least controlled by the community of co-opetitors.

A central exogenous component of this type implies a (very heavy) client-server model of coordination, as opposed to the fully distributed model we are looking after.

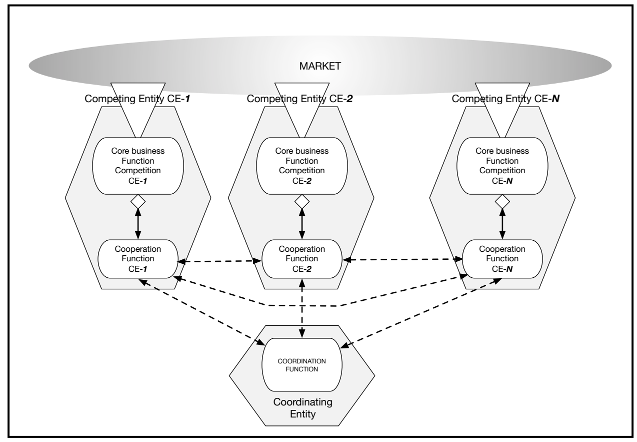

Figure 1. Business entities that compete *and* cooperate.

Huge sets of co-opeting entities like SWIFT or large stock-exchange platforms might need to manage billions of transactions monthly (that incidentally generate millions of euros of costs for the stakeholders) with a central coordination function, or they might be too big to get rid of a system that was once a solution to their reliability problem.

However, co-opetition on a smaller scale doesn't require and usually cannot economically support, a centralized, dedicated and humanly operated coordination function. In this case, the coordination function might consist only in managing reliable communication and consensus on a small set of global state values necessary for all actors to make mutually consistent and locally secure decisions.

Figure 2 illustrates this situation: the cooperation functions of co-opeting entities interact within a P2P network. Access by the coordination function to the core IT infrastructure of each competing entity is strictly controlled. There is no more active autonomous coordination function.

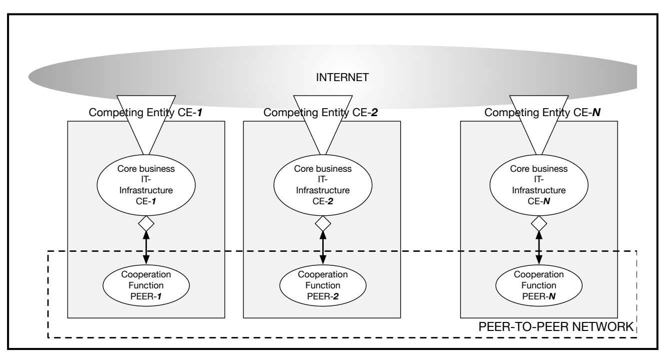

Figure 2. To implement cooperation in a peer-to-peer network.

In this representation, each competing entity is a *node* of the peer-to-peer network. At first glance, it should be possible to implement this configuration with traditional web-services (e.g., SOAP/XML or REST). However, this is not the case, specifically because a distributed coordination function cannot consistently be implemented using web-services that operate between IT-infrastructures of independent competing entities. Also, note that using a distributed database system in this situation is technically equivalent to using a centralized coordination function.

IV. Architecture of the Peer-to-Peer Network

In order for the cooperation- and the distributed coordination- functions to be implemented within a set of co-opeting entities the following conditions are necessary:

1) each competing entity completely, autonomously, and separately operates its own core functions, **and** each competing entity operates an instance of the cooperation function within a node of the P2P network;

2) each instance of the cooperation function maintains state values of the global coordination function. State values can be local values of the cooperation function instance executed on some node; or distributed state values that are consistent among the cooperation function instances of a subset of competing nodes; or consensus state values that must be kept consistent on all running cooperation function instances of the set of competing nodes;

3) the set of all coordination function instances of the group of competing entities, together with all state values of all types of these instances, defines a consistent distributed information system under conditions 1) and 2) above.

Since the cooperation function is specific to the type of business activity considered, the architecture of the P2P network and of each node have to provide some domain-specific services (in particular, the services required by the coordination function, if any), whereas the conditions that pertain to the operation of a distributed system rely on general-purpose services. This is illustrated in Figure 3.

Since the IT-infrastructures operated by different competing entities are by nature different, the implementation of the cooperation function, i.e., the implementation of the individual nodes in the P2P network, are bound to be different. The manner by which competing entities are brought to trust each other with regards to the correctness of their competitors' cooperation function (and possibly the underlying coordination function) can be left to each group of entities. However, *certification of nodes*, *fully traced communication*, and *non-repudiation* (in the sense that correct behavior of a peer can always be proven) are properties that can help foster trust. These features were implemented in the project described in the next section.

V. Implementation

To illustrate the development above, we briefly present the implementation of the peer-to-peer network that was implemented in view of [13]. In this case, the cooperation function was relatively complex (managing the transmission, authorized by their owners, of information between operators of public or private databases of farm-related data) with a coordination function that enabled data-owners (i.e., farmers) to enforce in real time together with each competing entity concerned, who was entitled to receive their data.

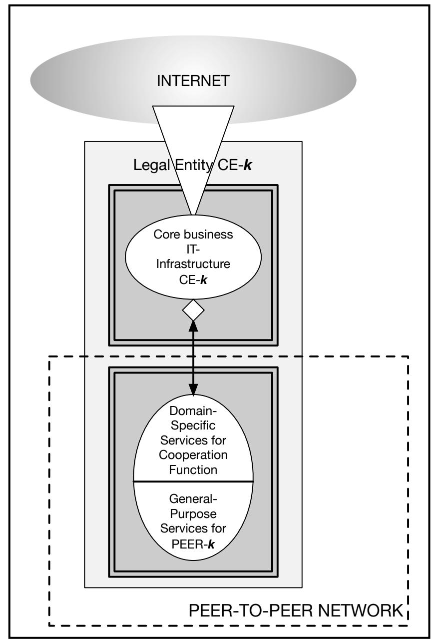

Figure 3. Architecture of a peer (node).

Because of the nature of the competing entities (often small, autonomous, and distrustful organizations with weak or external IT-resources), it was decided to provide the cooperation function, with each competing entity's node, in a separately operable Kubernetes (K8s) [14] cluster (see Figure 4). The goal is to facilitate integration and long-term maintenance by using standard infrastructure components.

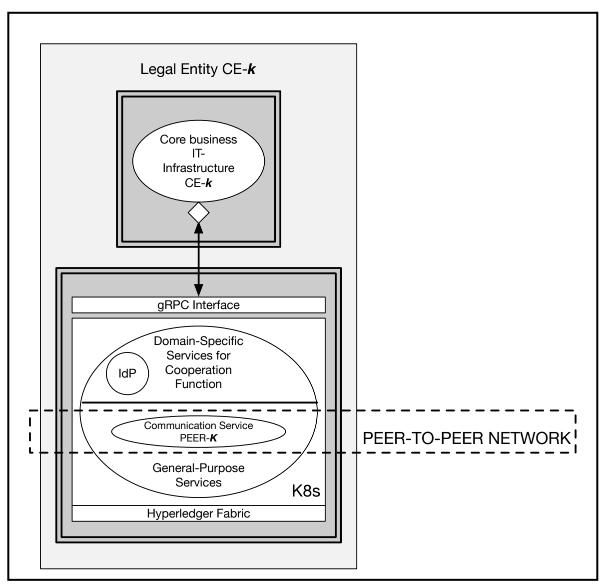

Figure 4. Implementation of a node with gRPC / K8s / HLF.

This feature also enables competing entities to outsource the construction and operation of their nodes, and at the same time withhold legal control and responsibility on their operations with contracts and service level agreements.

The interface (API) between the core function and the cooperation function is realized using the gRPC [15] framework. Identity and access management is realized with OpenID Connect/OAuth 2.0 [16]. The coordination function will rely on Hyperledger Fabric [17] (which has a K8s implementation) and is currently only partially realized with the ledger (integration, along with the implementation of traceability for sensitive products or objects in the value-chain using the ledger are planned in future work).

Hyperledger Fabric is a permissioned ledger well adapted to the situation of a set of co-opeting entities (by nature reconfigurable at any time, but not dynamic in real time nor open to unidentified / unqualified peers).

The implementation is meant for private and for public entities alike, in the agro-food sector. The heterogeneity of actors tolerates a model infrastructure like Kubernetes (that peers can bypass using another implementation at their own risk), and the usage of a permissioned ledger for storing and accessing global state values; it does not however allow the general usage of a blockchain for the storage of local or transactional state values, because of the IT policy of each peer that could possibly prevent it, e.g., for public agencies.

VI. CONCLUSION

We have shown how the local cooperation- and the distributed coordination- functions of a group of co-opeting entities can be implemented in a peer-to-peer network.

The successful implementation of the approach leads to two remarks. First, *co-opetition in business requires some sort of co-opetition among the software-providers of the business entities concerned.* Lines of business in economic sectors have their established sets of IT-tools and SW-providers (e.g., SAP among others for the enterprise resource planning core function of a business). If a group of business entities is lead to co-opete in its sector, then the group of associated software-providers should do so also: in order to supply their customers with the necessary cooperation and coordination functions (i.e., cooperate) and thus remain competitive on that market.

Second, if some economic activity is subject to central regulatory coordination (control) as in banking or in animal production, then *each business entity that is registered for the activity must implement and operate the coordination / control function in its processes and its IT*.

The approach proposed in this paper shows how the coordination / control function of the regulator could be distributed among these co-opeting entities in a P2P framework under good conditions (i.e., using the group of entities concerned to overlook the correct implementation of the coordination / control function). This could lead in some sectors to replace costly and rigid control structures embedded in public administrations by distributed resources that already operate the same functions, possibly reducing by half the cost of some regulatory controls.

These remarks, as well as the implementation of the ledger as a means for traceability in the cooperation function are the subject of ongoing and future work.

ACKNOWLEDGMENT

This work was conducted with the help and precious contribution in Switzerland of Léa Stiefel, STS-Lab, University of Lausanne (lea.stiefel@unil.ch), of Alain Buehler, Yves Gabry, and Jordan Latinov, and finally of Sunny Soft Ltd. in Sofia, Bulgaria.

REFERENCES

[1] B. Nalebuff and A. Brandenburger, Co-opetition, paperback ed., London: Profile Books Ltd, 2002 (first published 1996)

[2] F. Cruijssen, W. Dullaert, and H. Fleurens, "Horizontal Cooperation in Transport and Logistics: A Literature Review", Transportation Journal, Vol. 46, No. 3, pp. 22-39, Summer 2007, doi: 10.2307/20713677

[3] A. Kosansky and T. Schaefer ,"Should you swap commodities with your competitors?", CSCMP Supply Chain Quarterly, 2010, https://www.supplychainquarterly.com/topics/Logistics/scq201002swap/

[4] Society for Worldwide Interbank Financial Telecommunication (SWIFT) [retrieved: Feb. 2020] from https://en.wikipedia.org

[5] M. Bengtsson and S. Kock, " "Coopetition" in Business Networks—to Cooperate and Compete Simultaneously", Industrial Marketing Management, vol. 29, pp. 411–426, Sept. 2000, doi: 10.1016/S0019-8501(99)00067-X

[6] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies", ACM Computing Surveys, Vol. 36, No. 4, Dec. 2004, pp. 335–371, doi: 10.1145/1041680.1041681

[7] R. Sarkar, "Distributed Systems - Peer-to-Peer", University of Edinburgh, Lecture Notes, Fall 2014, [retrieved: Feb. 2020] from https://www.inf.ed.ac.uk/teaching/courses/ds/slides1415/p2p.pdf

[8] F. Victora Hecht and B. Stiller, "Enabling Next Generation Peer-to-Peer Services", Proceedings of the 2nd international conference on Autonomous Infrastructure, Management and Security: Resilient Networks and Services, Jul. 2008, pp. 211–215

[9] C. Canali, M.E. Renda, P. Santi, and S. Burresi, "Enabling Efficient Peer-to-Peer Resource Sharing in Wireless Mesh Networks", IEEE Transactions on Mobile Computing, March 2010, doi: 10.1109/TMC.2009.134

[10] M. Boron,·J. Brzezinski, and A. Kobusinska, "P2P matchmaking solution for online games", Peer-to-Peer Networking and Applications, Jan. 2019, doi: 10.1007/s12083-019-00725-3

[11] J. Gerke, D. Hausheer, J. Mischke, and B. Stiller, "An Architecture for a Service Oriented Peer-to-Peer System (SOPPS)", Praxis der Informationsverarbeitung und Kommunikation, June 2003, doi: 10.1515/PIKO.2003.90

[12] R. Leitner, F. Meizer, M. Prochazka, and W. Sihn, "Structural concepts for horizontal cooperation to increase efficiency in logistics", CIRP Journal of Manufacturing Science and Technology, vol. 4, pp. 332–337, 2011, doi: 10.1016/j.cirpj.2011.01.009

[13] L. Stiefel and A. Sandoz: "Reshaping Swiss Agriculture Through A Peer-To-Peer Approach", Society for Social Studies of Science, Annual Meeting, New Orleans, Sept. 2019

[14] Kubernetes, [retrieved: Feb. 2020] from: https://en.wikipedia.org/wiki/Kubernetes

[15] gRPC, [retrieved: Feb. 2020] from: https://grpc.io

[16] OpenID, [retrieved: Feb. 2020] from: https://openid.net/connect/

[17] Hyperledger Fabric, [retrieved: Feb. 2020] from: https://en.wikipedia.org/wiki/Hyperledger