ARCHITECTING A DISTRIBUTED SYSTEM FOR DATA SHARING

Léa Stiefel* - Alain Sandoz
University of Lausanne, Switzerland - University of Neuchâtel, Switzerland
lea.stiefel@unil.ch - *alain.sandoz@unine.ch*

Abstract

Keywords: architecture; architect's work; design; constraints; strategy; distributed system; organizations; data sharing;

INTRODUCTION

For many researchers, IT architecture appears to be an ill-defined concept (Vassilakopoulou and Grisot 2013). Generally used at a high level of abstraction, where everyone can agree on its usefulness (Bidan et al. 2012), the notion lacks precision.

To bring the notion out of this "elusive" state, some researchers have recommended to start from the study of architects at work. Scheil (2008), for example, defends this perspective as follows: being a praxiographic endeavor, architecture requires to leave the field of abstract definitions and to enter the field of concrete practices in order to be understood.

There have been repeated calls for empirical studies of how systems architects perform their daily work (Grinter 1999; Smolander 2002; Scheil *ibid*.; Figueiredo et al. 2012). So far, they have received few replies (more on this later). This article intends to contribute to filling this gap.

Over a year and a half, we followed an architect's daily work alongside his team members, engaged together in the design and development of a distributed system for the sharing of data in Swiss agriculture. Organizations in the sector, both public and private, were invited to exchange data by transmission, with the authorization of the data-owners, *i.e.*, the farmers. At the time we entered the field in January 2018, the system was in its early conceptualizations. In July 2019 a first operational prototype had entered production. This marked the end of our fieldwork.

This report is ethnographic in nature but also technical in some respects, since it sometimes requires going into the details of the functioning of the target system. To present its results, the first author of this article, who is a sociologist, joined forces with the architect, who is also a trained computer scientist. We will return to this partnership when we present our methodology.

We first summarize research on the notion of IT architecture to highlight some qualifications previously made to the work of architects and what we see as their limitations. The next section presents our case study, methods, and research materials. It is followed by our results. The final section relates these results to current research to highlight our contributions and outline some directions for future research.

STATE OF THE ART

In their exploratory state of the art, Vassilakopoulou and Grisot (*ibid.*) identify three ways of qualifying architecture, each supporting distinct research objectives. The first, called *technical/structural*, links architecture to the structure of a system, its components, their operating principles, and their interconnections. This definition would dominate the engineering literature whose research objective would be to develop frameworks and models to support the design process. The second, called *enterprise architecture,* views architecture as a socio-technical arrangement of software, hardware, organizational structures, and work practices. It would dominate the management literature whose research objective would be to define good management practices to support the design process, taking into account the needs of organizations that use systems. The last one, called *strategic*, designates architecture as the result of a process that aligns heterogeneous business and technical elements. It would dominate the innovation literature, which would focus on the benefits and challenges of technical principles such as modularity, layering, or the end-to-end principle, to support strategic interests such as innovation or generativity.

This exploratory review of the literature shows that research on the notion of architecture tends to focus on an outcome rather than on the processes leading to that outcome. It also shows that in its efforts to qualify architecture, the literature generally adopts a prescriptive rather than descriptive perspective. As a result, what system design might mean in practice, in terms of the architect's work, remains a blind spot.

A few empirical studies, of a more descriptive orientation, exist nevertheless. Smolander et al. (2008) propose a study based on interviews with employees of software companies. They identify four metaphors of architecture, representing their interviewees' views of the notion: architecture as blueprint, as language, as decision and as literature.

Architecture as blueprint is similar to the technical/structural qualification identified by Vassilakopoulou and Grisot (*ibid*): a high-level description of the system, its components and modes of communication, that guides the implementation of the system. *Architecture as language* views architecture as a boundary object (Star and Griesemer 1989) that progressively leads to an understanding of the system common to stakeholders, customers, development teams, managers, marketing experts, etc. *Architecture as decision* refers to the product of a decision process, based on strategic, functional, or quality objectives and design constraints, defined by the socio-technical environment, such as existing systems, available technology or expertise, time resources, and/or budget. Finally, *architecture as literature* is similar to the blueprint metaphor, a document of technical structures, concerned however with the maintenance of a system rather than its implementation.

These metaphors are interesting in that they provide a sense of what the work of architects might be like. However, the contours of this work remain vague, given the nature of the authors 'investigation.

We understand that the work of architects is oriented towards the delivery of a system. We understand that this work involves making plans or descriptions of the system to be built, talking with various stakeholders, developing strategies, and making decisions based on goals and constraints determined by the environment. But what else?

Overall, the literature highlights a set of variables at work. For Hoorn et al (2011), the architect's work is to balance a set of objectives and constraints. Objectives are driven by the needs, interests, and concerns of different stakeholders, which may be functional, qualitative, or strategic (*e.g.*, in business terms). They involve trade-offs (Grinter, *ibid.*) that may be difficult to achieve because stakeholders, anticipating that design decisions will affect the distribution of power, will actively ensure that developments are compatible with their interests (Robey and Markus 1984). Constraints are referred to as technical or financial (van Heesch and Avgeriou 2011). They depend on project resources (budget, schedule, team size, available expertise), industry standards and best practices, legislation (Van den Berg et al., 2009) or legacy systems and organizational norms (Figueiredo et al., *ibid.*).

These variables say important things about the architect's work. But how these variables play out, how they are exploited, and how they translate into a system *of a particular shape* remain open questions. They seem all the more important since many of these works present not only an objective of knowledge, but also of prescription, on how to best align computer systems with their socio-technical environments.

Behind-the-scenes investigations of systems factory are undoubtedly difficult to conduct. As Rechting and Meier (2010) put it, "many of the most interesting stories are buried behind walls of proprietary information." There are ways to come around this difficulty, such as to move away from closed systems towards more open, collaborative, inter-organizational systems.

This strategy has been adopted by a body of work, rooted in the field of information infrastructure, that has studied cases of digital health systems deployed in developing countries (Sahay et al. 2009; Poppe et al., 2014; Nielsen and Saebo 2016) and from a sector perspective. Systems studied serve a plurality of heterogeneous and autonomous entities: organizations structured around health programs such as HIV/AIDS or malaria (ministries of health, global donors, and aid organizations), which operate their own systems but would benefit from data flows between them. The research question guiding this body of work is how to strategically integrate these systems, while ensuring flexibility for change.

As Robey and Markus (*ibid.*) had already pointed out, this work shows that the issue is highly political: integration risks challenging the position of system owners, and they do not hesitate to actively work to ensure that developments correspond to their interests. In this context, architecture appears to be a delicate negotiation activity. The authors report on the strategies used for integration, between unilateral developments (offering new functional modules) or negotiated developments with the surrounding systems (or API connections).

Our paper is in line with this work, which focuses on describing architecture in the making (as a process, an activity). The system we studied was at a sectoral level and aimed at enabling plural, heterogeneous and autonomous organizations to share data. The strategies pursued and decisions made during the system design process, as well as the shape of the resulting system, were nevertheless of a different nature than what the previously mentioned corpus has documented. The purpose of the following section is to introduce this system and the methods and materials we used to study it.

CASE STUDY AND RESEARCH METHOD

The system emerged in a particular context which strongly influenced its design. We propose to report here the broad outlines, before presenting our methods and materials.

During 2017, the Swiss agricultural sector was faced with a private attempt to centralize all farmers' data in a single database. The stakeholders were important actors in the field: a national extension center and an IT company active in the animal sector, in majority state-owned. Starting in 2018, other stakeholders joined the effort, including Switzerland's largest cooperative, the farmers' main supplier and a major buyer of their products, as well as a European software company linked to the cooperative through a German machinery manufacturer.

The centralization of data was presented as a way to simplify the administrative work of farmers. Farmers would no longer have to feed the same data to the different applications of the public administrations and private organizations that managed the sector. A platform anchored to the centralized database would offer smart farming modules and increase competitiveness.

Farmers and leaders of agricultural organizations were not convinced. For farmers, the centralization of their data would above all lead to increased control over their production activities. This was all the more worrying since the largest agricultural cooperative in

Switzerland, already known for its aggressive commercial methods towards producers, was part of the project. Farmers would continue to assume the risks of on-farm production and would now bear the development costs, given the platform's business model based on license payments, while the platform's profits would go to its shareholders.

For public organizations such as cantonal administrations and private organizations such as certification bodies, centralization meant the introduction of new dependencies, threatening the autonomous management of their activities. That farmers were to enter their data on a single platform meant that organizations would have to connect to the platform to get the data they needed for their tasks and services. There was no guarantee that they would get the data they needed, when they needed it, nor that access would be free of charge.

So many farmers and organizations were not ready to see "all data in one database" become a reality. Among them, an association of producers in the field of integrated production (that we will call *APIP*) decided to find a way to stop the project. By the end of 2017, it called in an expert in strategy, engineering, and information systems architecture, with whom it had worked in the past.

To prevent the concentration of all data in one new central system, he proposed to design a distributed system for the sharing of data between existing systems. The proposal was accepted by *APIP*, soon joined by a large control coordination body owned by crop producer associations (that we will note *CCO*). Together they represented more than 50% of Swiss farmers. They appointed the expert as architect of the distributed solution and leader of the alternate project to centralization.

In the rest of the paper, we will examine this counter-project to data centralization from the perspective of the architect's work.

We followed the project from its beginning, for the first author in early 2018 as an ethnographer engaged in the study of digitization of Swiss agriculture since late 2017 and alerted to the alternative project by field actors she had interviewed; and for the second author, himself as the architect and leader of the project since its inception. Subsequently, our partnership was formally defined during meetings in May and June 2018, after the project had been publicly established as a serious contender against centralization.

The ethnographer would have the opportunity to go behind the scenes of the project, to follow and document all its developments. In return for this full access to the project, she would provide the architect with regular feedback on her observations, based on her knowledge of her disciplinary field, the social studies of science and technology; and on her progressive understanding of the dynamics of digitization in the sector, via anonymized accounts of the interviews she planned to conduct and actually did conduct between 2018 and 2019 with some 40 farmers and leaders of public and private organizations.

In addition to his professional experience and practice with distributed computer systems, the architect would benefit from this informed "socio-technical" perspective in conducting the project.

A set of materials was produced during the project, from which we draw our conclusions. They tell the story of the system's journey from its first conceptualizations in late 2017, to its entry into production as an operational prototype in July 2019.

They highlight the architect's exchanges and meetings with: *i)* numerous actors in the agricultural sector, to interest them and make them adhere to the project; *ii)* members of the boards of the organizations carrying the project, to keep them informed of its developments; *iii)* members of the project teams, *iv)* IT staff of partner organizations, and supporting farmers, respectively to develop and specify, integrate and use the future system.

In detail, these materials include: *i)* all documents produced during the project, *ii)* all emails exchanged during the project, *iii)* a corpus of press clippings published about the project from the agricultural press, *iv)* verbatim transcripts of sessions held internally by the project teams, as well as interviews conducted by the ethnographer with the project leader and members of his teams, vi) notes from a 160'000-word field diary kept by the ethnographer throughout her investigation. Table 1 provides an overview of these materials.

Table 1: Fieldwork materials (D: documents, E: emails, C: clippings, T: transcripts, J: field journal)

No.	Contents	Type	Nb
1.	Powerpoints produced by the project leader and architect in support of his meetings and public presentations of the project, intermediate reports documenting the project	D (J)	34
2.	Contributions in the form of blog posts or scientific publications by the architect about the system under construction.	D (J)	4
3.	Internal team exchanges (reports on developments, problems encountered, questions on the why and how of the system under construction, or more programmatic).	E (J)	Approx. 50
4.	Exchanges between the architect and actors in the agricultural sector (reports on project developments, invitations to public presentations, or expressions of interest to participate or join the project).	E (J)	Approx. 30
5.	Exchanges between the ethnographer and the architect (feedback or strategic reflections on past or upcoming meetings or events of importance to the project, questions or suggestions on the documents produced).	E (J)	Approx. 40
6.	Press clippings published on the project from the local agricultural press (in the form of interviews with the project leader or its sponsoring organizations)	C (J)	6
7.	Working sessions held internally by the project teams (Hangout, sessions, workshops)	T (J)	20
8.	Interviews conducted by the ethnographer with the project leader and architect	T (J)	3
9.	Interviews conducted by the ethnographer with members of the project team	T (J)	4

RESULTS

This section reports on the tasks executed by the architect to reach the stage of a first operational prototype of the distributed system. The work required was broken down into different activities, which involved different stakeholders, each with its own goals and challenges.

These activities consisted of: *i)* designing a solution to the problem within a set of identified constraints; *ii)* building a team and bringing it to understand the system's design, while opening it up to their own contributions; *iii)* climbing on stage, debunking the competitor, and gaining the public (*i.e.*, the future users) support; *iv)* defining a project strategy and preventing the project's sponsors from drifting away; *v)* preparing the solution's future operators for integration.

i) Designing a solution within a set of identified constraints

The architect's mandate was to stop the development of the centralized platform, and to do so before it enrolled additional shareholders and came to be seen as an unavoidable necessity. The architect translated this objective into the need to offer the agricultural sector an alternative solution to the problem of the farmers' digital administrative burden, radically different from the centralized platform, but also faster and less expensive to develop.

The centralized platform's "solution" was to provide farmers with a single place to enter their data. This de facto meant that other organizations would stop collecting data directly from farmers to fulfill their public missions or provide their private services and would have to connect their systems to the central platform to get the data. In IT terms, this meant that they would have to give up their status as "masters" of collecting data from farmers and take on the role of "slaves" to the centralized platform. Not only did this threaten the proper functioning of the organizations that depended on farmers' data, but it also concentrated the potential added value of all that data in the hands of the platform's shareholders. On the farmers' side, there was no indication of how they could, or could not, control the flow of their data.

For the architect, these logics of interference, capture and opacity were not to be found in the design of an alternative. His solution to the problem of farmers' administrative burden had to preserve the sovereignty (Hummel *et al.*, 2018) of organizations over the data they were responsible for and their information systems. Moreover, it had to enhance farmers' control over their data and ensure the distribution of any added value that could be created from this data.

In December 2017, the architect laid the groundwork for his solution: it would be an infrastructure with only two "simple" functionalities: for organizations to transmit data, and for farmers to authorize the transmission of their data. Data transmission would be executed point-to-point directly between the systems, and under the control, of the two organizations involved in a transfer. Authorization would be under the control of the data owners, the farmers. There would be no central component in this infrastructure, fully distributed among the agricultural organizations. It would not imply any intrusion into their information systems, nor involve any third party in any aspect of its operations.

The transmission of data from one system to another would reduce the administrative burden on farmers, who might no longer have to enter some of their data repeatedly. Authorization would ensure farmers' control over the flow of their data. Data exchange would help organizations offer new value-added services, useful for farm competitiveness, determined by service providers and by the farmers controlling the data flow, and without interference from the infrastructure.

Specifically, the infrastructure would consist of a network of nodes, inter-connected via the Internet. Each node would be under the legal and operational responsibility of a single organization and connected to its information system via an application programming interface, *independent of the data structures used in agriculture*. The same software would run in each node, making the nodes functionally identical and making the network a true peer-to-peer network. The software would be developed by the project under the GPL license to ensure an open and transparent infrastructure. It would include a minimal set of "services" designed to enable the transmission of authorized data under conditions that enhance trust between partners, farmers, and organizations. A certification process would ensure that these conditions would be technically meet at each operating node.

At a minimum, the set of services locally implemented in each node would include:

- a **transmission service**, connecting an organization's information system to its own node and enabling it to send and/or receive data;
- an **identity provisioning service** and an **authorization service**, accessible via the farmer’s mobile app, which would provide farmers with credentials at each node involved in a data transfer (without the nodes or operators having to share a common identity for the farmer), and enable farmers to define and manage their authorizations;
- a **communication service**, for the transmission of data between pairs of nodes, with all the encryption functions necessary to secure the transfers;
- a **logger service** to ensure the traceability of data flows between pairs of nodes;
- and finally, a **leger service** to enforce a distributed consensus on the data structures available for transmission, *i.e.*, to enable each organization, peer of the network, to publicly register/subscribe, in an unalterable and ordered manner, the datatypes it wished to share and/or receive, informing the other peers of their structure and usage.

In keeping with best practices in system design, the infrastructure would also be layered like the Internet. Logically, service-to-service transactional operations would be implemented pairwise between nodes' transmission, authorization and communication services respectively. Each node's identity provisioning service would be autonomous (allowing its own identification and authentication of its farmer-clients). Each node's logger would be local and totally controlled by its peer. Finally, the ledger would be a common decentralized application whose sole functionality would be to provide consensus on datatypes and the order in which they would be published by peers.

The layered approach would enable the functionality of the system to be adapted or extended over time as needed, without having to disassemble and rebuild from scratch. In particular, it would enable existing organizations, farmers, and future members of the open-source community, to steer the evolution of the software according to needs, with minimal technical intervention on the platform.

Significantly less complex than the centralized platform, the solution promised faster and less expensive development. The architect estimated that it would cost 6 to 8 times less, *i.e.,* 800'000 CHF (Swiss francs), or 750'000€, for a first productive version by summer 2019, compared to the 5 to 7 million CHF planned for the centralized platform over the 2018-2024 period.

Over time and through interactions between the architect and team members, this initial design was refined, until it stabilized in February 2019, when the architect put a first complete description of the system architecture on paper. Part of his job, however, would be to ensure that none of these design evolutions would conflict with the core principles of the solution he had determined in December 2017, as we will see in the next subsection.

ii) Assembling a team and bringing it build to the system

In 2017, when the possibility of a system-building project was just emerging, the architect contacted an acquaintance with whom he had worked on e-government projects for the State of Geneva years earlier. Working as a freelancer since he had left a position of division-CTO at Hewlett-Packard on the eve of the merger with Compaq, he provided 5-6 days of consulting to the project and brought his technological vision of what would be needed to support the distributed system that was in the architect's mind. His proposal was to go with a "massively distributed" and open source microservices infrastructure, similar in principle, but a lower level of the architecture of the envisioned system and called Kubernetes (K8s). To implement the distributed consensus on the data structures (the ledger service), he proposed to rely on another open-source project called Hyperledger Fabric that was also running on K8s. When the project started, he was appointed its chief technologist.

The chief technologist was also connected to a strong near-shore team of developers in Bulgaria, that he had built up since 2003 through family connections, and closely followed in their startup project. It took a few months to gather enough developers around the project. Another freelancer, former technical manager of the team before they started their own company, had to be called in as a high-level distributed systems engineering backup. By mid-2018, the developer team was complete.

Alongside the chief technologist and developers, an analyst was brought on board in early 2018, to work on the issue of what type of data structures would be used for data-exchange. Like the chief technologist, this person had worked for 20 years at Hewlett-Packard and had also met the architect at the time of the e-government project in Geneva.

Another person approached by the architect before the project was launched was the scientific director of a Swiss university of applied sciences. He and the architect were long-time colleagues and friends. The architect wanted to explore the possibility of having thinkers who could provide a specification of the system to be built. He thought the university was the right place for this, as it would also give credence to the project, whose technical baseline could be publicly presented as scientifically supported. The science director favored hiring a PhD student, but the idea had to be abandoned due to a lack of candidates. He finally managed to assemble a specification team for the project from different units of his department.

Another person approached by the architect in early 2018 was the director of a company specialized in IT hosting. He and the architect had known each other for 25 years. They had worked together for the federal government before they both left in late 2000, each to start his own company: the architect for "more strategic and conceptual" services, his colleague for "more operational" services, as the latter reported to the ethnographer. In May 2018, he reserved the system project Internet domain name. In the summer of 2018, he was appointed to equip the project with a robust production environment.

By summer 2018, the entire project team was assembled, based on the architect's personal relationships, including, in addition to the specification (5 people), development (7 people), and infrastructure (2 people) teams, a small peasant advisory group (3 people), a web designer, and a graphic designer. The sociologist (first author) would join the team in September 2018 as part of her fieldwork.

The architect's mission was to counter the centralized platform project, and quickly. To do this, he parallelized the three tracks of 1) design/strategy, 2) development, and 3) system specification, whereas usually these tracks are placed sequentially one after the other in the order 1-3-2). In other words, the teams were invited to progress at eye's sight on their own and in discussion with the architect who would synchronize the tracks.

A series of "sprints" of work sessions and workshops were organized for this purpose, accompanied by weekly online hangouts and numerous email exchanges. This very agile and unconventional approach, like any exploratory effort, and under time pressure, revealed its moments of high tension but also opened the field to invaluable contributions from the team. The challenge for the architect would be to communicate his uncommon conception of the system (being an innovative project), while remaining open to input, but unwavering in the fundamentals of his design set in winter 2017.

One element of conceptualization that proved difficult to get the teams to understand was related to the notion of state values. This notion came from his academic training (the architect had obtained his PhD in distributed systems in 1992 at the École Polytechnique Fédérale de Lausanne (EPFL)) and seemed unknown to the team. For the architect, every system had a state, qualified by values, but values of three types for a system of distributed nature: local, shared, and global.

Local state values, the architect explained, qualified a state specific to a node in the network and were unique to that node. Shared state values qualified a state shared by a fixed number of nodes in the network and were unique to those nodes (in the case of data transmission, only two, *i.e.*, the nodes involved in one transmission). Global state values qualified a state that was shared by all nodes present in the network. Local state values involved local data persistence, shared state values involved bilateral transactions. Globally shared state values involved consensus.

It was important for the architect to distinguish between these different types of state values. According to a principle called "need-to-know", it was imperative that the values be distributed only to the actors concerned by them. If a state concerned only one peer in the network, its value had to remain local to the node, and moreover not to be in control of remote nodes; if a state concerned only two peers, its value had to be shared consistently only by their two nodes; if some state concerned all the peers, then its value had to be shared globally, and only in that case. This vision set a design constraint that the development or specification of the system should not violate.

While teams seemed to understand this vision in theory, it sometimes seemed to be quickly forgotten, in favor of a vision guided by the turnkey solutions offered by available technologies.

A member of the specification team suggested in December 2018 that blockchain be used to store data transfers and farmers' authorizations on those transfers. The architect nearly fell off his chair. They had talked about it many times. Here were state values that were only to be shared between the two nodes in the network directly involved in an authorization and subsequent transfers. For what compelling reason should one manage that value with a blockchain?

Data transfers and authorizations were sensitive information. The organizations, peers of the network, had to have the means to control, each for itself, these operations and, if necessary, to be able to prove that things had been executed according to the will of each party: organization B had requested to receive this data from organization A, the farmer had authorized this transfer, A had sent the data to B that had confirmed its receipt.

A transaction with public-private key exchange guaranteed, by its relatively simple implementation, a secure execution of these processes and a control over the results by each participant in the exchange. The blockchain, in addition to being expensive, unnecessarily complex to solve such a problem, was also likely to compromise the process, its locality and timeliness. The team was familiar with problems like the Byzantine Generals: network failure, shady qualified majority, etc. There was plenty to compromise the consensus and thus the process and its outcomes, not to mention the political uprising that would follow a statement like "all farmer data is managed in a blockchain".

iii) Going on stage and demonstrating an alternative

In 2016, *APIP* had first been approached by the centralized platform project (its origins go back 2015). A delegation had come to politely ask that the data management of the 20'000 farmers under *APIP*'s integrated production label be handed over. Concerned by this request, *APIP* had consulted its legal advisor to find out if it could be legally enforced, given that the centralized platform's promoters were bound to the state by public mandates.

The lawyer concluded that the request contained no legal basis and *APIP* decided to ignore it. Disturbed by the project's approach, which bordered on legality, the association nevertheless undertook to speak with the Federal Office for Agriculture (FOAG).

The architect, who was then working for *APIP* to help determine how the association's IT resources could be used strategically on the long term, attended the meeting. He presented his vision of the centralized platform: not only was the initiative unfeasible, without legal basis, excessively costly (costs that would be passed on to farmers, given its license-based business model), but it threatened the activities, and thus the very existence of the sector's organizations, and also of the farmers, at least the smallest, who would be subjected to an uncontrolled increasing pressure for verticalization from the large distributors (such as the cooperative, shareholder of the platform).

The FOAG was said to publicly defend an open approach to data exchange in the agricultural sector, based on open source and free software (FOSS). The architect argued that *APIP* was willing to develop and publish an open concept on its behalf. The representatives of the FOAG politely acknowledged, letting it be understood that they were not interested.

In 2017, *APIP* had the unpleasant surprise to learn that the centralized platform would be presented at an event hosted by the Minister of the Economy. More than 200 people, representing the main players in the agricultural sector, were expected to attend. Now that it was publicly supported by the state, *APIP* feared that the centralized platform would not give up so easily.

After having met with the federal administration, the architect took the initiative to meet with the Swiss Farmers' Union, an important player in the defense of professional interests, which had expressed interest in joining the central platform's shareholders. He used the same strategy: to try to detach them by argumentation.

Meeting with key players and arguing against the platform was one thing. But it was unlikely to be enough to bring down the centralized platform project. In December 2017, the plan to form an alternative was launched, despite the lack of interest shown by the federal administration, which was deemed entangled in its biases.

The producer association (*APIP*) and the control coordination organization (*CCO*) were going it alone with their financial mean. This could last for some time, but other organizations would have to be found to invest and make the alternative sustainable. And for that, the attention of the agricultural sector had to be mobilized. The architect defined his strategy: they would have to get on stage and publicly unfold the alternative approach.

In February 2018, the project team managed to get invited to present their alternative, face-to-face with the centralized smart farming platform, in Bern, the capital of Switzerland, and in front of about 100 representatives of public and private organizations from the Swiss agricultural sector. The architect justified the move as follows:

"*When you want to strategically occupy a position, you have to take it. There's no point in politely knocking on the door and asking, "Can I come on stage? Because I'd like to …". Okay, you go. Finally, you walk into the middle of a play. You see the reactions that this can cause.*"

The event was organized by the association of milk producers who wanted to learn more about the alternative before committing to one of the projects. Participants included representatives from agricultural research, federal (FOAG) and cantonal (cantonal agricultural services) administrations, control bodies, extension services, cooperatives, trade associations, public and private organizations from the dairy, animal and plant sectors, label holders, and service providers, including digital service providers.

The design of the distributed system had only been deposited for two months. The event was an opportunity to present the project guidelines, but of course nothing had been developed yet. It was necessary to show the alternative, quickly and with the financial means at disposal, to hope for the attachment of new actors to the distributed system, or even the detachment of actors from the centralized platform in favor of the distributed system. On the other hand, the centralized platform had important actors in the agricultural sector, substantial investments, and even the authority of actors such as the government and the Swiss Farmers' Union, who publicly supported it. Moreover, it represented something that was well known to agricultural actors: centralization, unlike the alternative.

The management of farmers' data was organized locally by each actor in a centralized manner. Each organization had information systems and databases that farmers had to fill in directly. Data exchange between organizations was practiced in the sector, but according to a master-slave scheme, with APIs dictating, by the master, what data to transmit, in what format and at what time. In other words, the peer-to-peer model, without data standardization thanks to the principle of distributed consensus on data structures (published in the ledger by the senders), as proposed by the alternative, was totally unknown in the sector.

It was necessary to show the alternative, to show that it could work, but also to give to understand the logic of its "invisible piping" (or its functioning in back-end). This is what the architect undertook to do. The idea was to continue the bilateral meetings on the one hand, and on the other hand to organize public presentations of the project as in February, but this time supported by "releases” of the solution in progress.

The February event had attracted a number of new parties interested in the alternative solution, including cantonal administrations. Among these, one canton had expressed interest in engaging in a prototype to test the distributed solution. This would provide significant political support to the project. In November 2018, the architect took the stage a second time, this time with the alternative as the only candidate presented, in front of the same audience of about 100 representatives of public and private organizations in the sector. This was an opportunity to present the proof of concept (PoC) for the distributed system.

The PoC consisted of four downloadable Android apps: an app for the farmer to define and manage authorizations, and three apps containing dummy data, designed solely to test the authorization- and data transmission-functions of the system. The demonstration focused on the user-visible part, the front-end for farmers. The back-end of the solution, organized as a network of nodes, with its set of identical services for each node, deployed on a productive (K8s) infrastructure, had of course not yet been developed in its final version. Nevertheless, the event had the desired effect.

New parties showed interest in the solution, including suppliers of software and hardware for the agricultural sector, additional cantonal administrations, and organizations in the plant sector. The Swiss Farmers' Union expressed interest in continuing the exchange with the alternative, in parallel with the centralized platform. The federal government announced its interest in participating in a pilot to test the system, if the smart-farming platform could be involved (*e.g.*, as a test recipient of data). Another positive outcome was that the architect was invited to present his solution and its future possibilities, especially in the field of traceability, at an event organized by the FOAG in March 2019. This event attracted new interested parties, including a major Swiss retailer and a multinational food company.

These events and meetings were beneficial for the visibility of the project, which was covered during the same period by the agricultural press on three occasions. However, they did not translate into financial contributions to the project, which nevertheless needed resources to sustain its development. *APIP* and *CCO* remained the only sponsors of the distributed system. Development continued until June 2019 with a new public presentation of the project.

The architect announced that the system was ready to go into production. The infrastructure was now ready to integrate real information systems with real data. The first demonstrators were going to be *APIP*, *CCO*, and the canton of Bern (which had expressed its interest in a pilot to test the system). Interested parties were invited to contact the project management if they wanted to launch their own demonstrator. Two more press clippings appeared about the project, and one additional canton asked to schedule a meeting to launch its own demonstrator.

During those months of work to develop the solution and get the sector interested and enrolled, the centralized platform had not disappeared from the landscape. By the end of 2018, it had managed to enlist new shareholders from the animal and dairy sector (6 organizations). Two modules had been introduced to the market. However, this was far from the promise of a single entry through the interconnection of sector databases. As for the number of farmers using the solution, 2'400 users out of 53'000 farmers in Switzerland were the figures retained by its promoters.

iv) Defining a strategy and staying on track

The project's strategy had been defined by the architect in the winter of 2017: the threat of the centralized platform would be eliminated by presenting the industry with a credible alternative that would not play into the hands of its powerful opponents. Maintaining this strategic line, endorsed by the peer-to-peer platform's sponsoring organizations, proved tricky at times, involving confrontations between the architect and his own project sponsors. One such difficult moment occurred in early 2019.

By the end of 2018, the distributed system project had gained visibility equal to that of the centralized platform project. The government had been prompted to propose a pilot project to test authorized data exchange between organizations, via the peer-to-peer platform. The pilot would test transmission between 1) one of the government's legacy applications, 2) the centralized platform, and 3) a third system to be determined. A precondition was that the project should involve the centralized platform (2). A joint working session was organized in January 2019 to discuss the concept. Anticipating a financial contribution from the state to set up the pilot, representatives of *APIP* and *CCO* attended the meeting, as well as the architect.

After an initial round of chilly small-talk, the centralized platform's side required that the canton of Bern be the third party (3) involved in the pilot. Bern was a key partner in the distributed system project. The promoters of the centralized platform had tried repeatedly to forcefully pull the cantonal administrations in their own project, without success. For the architect, the proposal was of very poor taste.

There were no representatives from Bern in the room. It was inconceivable to enroll a project partner on a sensitive track in their absence. More fundamentally, to accept such a proposal was to publicly befriending what the distributed platform project was fighting. It was adding risky tasks to the project, potentially alienating an existing partner, and benefiting the centralized platform which would now boast of working with the cantons. And it was not even for a benefit, on the contrary, since the host of the meeting had let it be understood the that government had no money.

"This is crap", the architect said to the project's sponsors, who looked at him with wide eyes, not seeing what the problem was. They had been seeking government support for months, hoping to finally get financial backing for a project they had been carrying (for too long) alone. Even if the government didn't pay immediately, their presence on board would certainly help.

The architect made it clear that if the pilot project were to be set up, he wouldn't get involved; moreover, if it were to be done with Bern, he would step down. In the meantime, he communicated, he would bring the distributed system project into a clean state, so that it could be restarted seamlessly, regardless of delays, interventions, or replacement of the project leader and architect. In February 2019, the board of directors (project sponsors) would finally express their support for the architect's view and ask him to continue development.

v) Preparing the organizations for integration

The fact that the distributed platform would be cheaper for organizations and farmers and much faster to develop than the centralized platform was a key argument for the architect to get stakeholders to commit to this alternative. It also meant that the system had to be designed in such a way that the effort of integrating organizations on the distributed platform would be as low as possible.

All the functional software components of the distributed system developed in-house by the project teams were designed with this in mind. Once a node had been set up, all that remained for an organization, and more precisely for its IT operator, was to use the transmission service, *i.e.*, to trigger the data transmission function through the application programming interface between its information system and its node; and to do this, to parameterize this interface on the objects (data) that the organization wished to transmit, by defining a segmentation (data structures) of its choice.

If the distributed system did not imply any intrusion in the organizations' information systems, a functional extension of the latter had to be necessarily realized in order to implement the authorized transmission of data.

This integration effort was estimated, based on the experience of the project teams, at two man-months, *i.e.*, the worktime of one full-time operator during two months. With the integration support of the project teams, this figure could be reduced to one man-month.

It was also recognized that some organizations might not have the infrastructure in place to run their nodes (based on Kubernetes and Hyperledger Fabric technologies). The architect and chief technologist came up with the idea of offering these nodes as commodities, which could be provided under contract, for example, by the hosting company associated with the project or by other organizations in the network of peers.

In financial terms, and at the demonstrator stage, this amounted to 6'000 CHF for integration assistance, the effort of one month's full-time manpower as discussed, and 1'000 to 1'500 CHF monthly rent for a node until the organization felt ready to build it itself, locally (and if it was more cost-efficient for it to do so). These costs were low compared to what is common in open IT services market.

To the canton of Bern, the first to commit to the project, the project offered to build and operate its entire demonstrator free of charge. The canton brought political credibility of considerable value to the distributed solution. In addition, it would enable the team to steer the entire process according to plan and better support future demonstrators.

Part of the architect's work, in addition to getting the organizations in the sector interested and enrolled, was to meet and talk with the systems operators of these organizations, to show them what kind of effort the integration was and, more generally, to solicit their curiosity about the project. The sociologist observed the enthusiasm of these actors as they were introduced to the project. In some cases, they proved to be key intermediaries in motivating organizations, especially cantonal ones, to join the project.

With first partners, the second development phase of the project, the deployment of the solution as an open standard, could be envisaged. From the beginning, the project was announced as a non-profit project. However, the sponsors expected their investment to be covered.

Node rental and integration support would be a source of revenue. In the longer term, these could be supplemented by a fee for use of the open standard once deployed in Phase II, estimated at 40'000 francs per organization, as well as by a node certification process for 10'000 francs per small organization and 50'000 francs per large organization. In addition to these elements, the architect also conceived of services, particularly in the area of traceability, which the project sponsors could develop on the basis of the infrastructure and the data exchanged, like any other partner.

The architect submitted this business model to the sponsors in March 2019. In April and June, the sponsors committed to funding the second phase of the project: publishing the platform as an open standard (FOSS). But a change in leadership within the control coordination body caused these prospects to be put on hold in July 2019. The continuation of the project would now be in the hands of its sponsors.

DISCUSSION

The literature argues that the architect's work consists of a process leading to the delivery of a system, whose form translates the socio-technical environment for which it is intended. It emphasizes that the realization of this translation, conducted in a processual manner, is delicate. The architect must grasp and assemble the visions (needs, interests, concerns) of the various stakeholders, which are often conflicting, even contradictory. In addition, a set of constraints also weighs in the definition of what can be done. The articulation of these visions and constraints often involves trade-offs. Balancing them is at the heart of the architect's work.

Our results do not call these elements into question. But their ethnographic nature (the fact of having gone behind the scenes, of having followed and documented on a daily basis a system construction project, in an architect-ethnographer dialogue) allows us to go a little further in the qualification of the architect's work, of the dimensions it implies.

First, as the literature has shown, systems-building never occurs in a vacuum. The architect works with a set of "constraints" that will guide his design strategies and decisions: the legal context, established industry standards, best practices, legacy systems and their associated organizational environment. This is also true in our case. But our case also shows that what motivates the construction of a system may be another system: one that is about to become dominant and that has to be destroyed, or at least maimed, before that happens. The construction of a system can take place in environments where several competing solutions oppose each other. This competition/opposition, source of tension, was a determining factor in the strategy adopted by the architect and his design of the system.

Proposition 1: The "competitive" environment in which system building projects can emerge can act (as a determining factor) not only on the strategy adopted by the architect, but also on the system's design. This competitive environment is then to be elevated to the rank of "constraint" playing on the architecture [as decision].

Second, as the literature has argued, the architect's work uses negotiation and compromise, aimed in particular at reconciling the plural and sometimes contradictory perspectives of various stakeholders. What this doesn't say, and is revealed by the case study, is that negotiation can also be marked by conflict. All the more so when the context of the project is highly political and takes place under the pressure of time and money. Compromise may not always be an option. We saw clashes between the architect and his team, when the latter tended to deviate from the fundamental design principles. We also saw the tensions between the architect and his sponsors, when the latter tended to deviate from the agreed-upon strategy. These deviations were unacceptable to the architect, as he perceived them as a threat to the project's survival and the system's deployment.

Proposition 2: Struggle with the stakeholders and rejection of compromise can characterize the work of the architect committed to a project's survival. Architecture is a boundary object, a language, but a [language] that can at times also be impetuous.

Third, as the literature has affirmed, the architect's task is to deliver a system. To this end, the architect discusses with stakeholders [architecture as language], strategizes and makes decisions [architecture as decision], *and draws plans* [architecture as blueprint]. The literature focuses broadly on the outcome rather than on the process and sees the architecture as a description of the system that leads to its implementation. Our case challenges this view. Confronted to a strong centralized competitor, speed of development of the distributed solution was an issue for the architect. This led him to parallelize design, specification and development. Initial partial development of the solution was completed by mid-2018, while its specification did not emerge until February 2019.

Proposition 3: The architect's work is subject to contextual constraints that must be embraced to move forward. Politics, pressure on resources, and strategy may lead the architect to start development before any specification is available, to postpone, to resume, to continue development on some basis, or to abandon it… Architecture [as a blueprint] does not always (if ever) follow a linear process. It is iterative.

The literature adopts an intra-organizational orientation in the definitions of the architect's work and of architecture. This is probably because most of the research on the subject is in the field of management. Vassilakopoulou and Grisot (*ibid*) speak of enterprise architecture. Smolander and al. (*ibid*), list customers, managers and marketing experts as stakeholders. That was not the perspective of the system we studied, nor of the work of its architect: it was oriented towards an entire sector, composed of farmers, and public and private organizations with managers and IT operators. The system was built to be inter-organizational. This difference in orientation might explain the nature of our contributions.

In terms of research perspectives on the work of the architect and the notion of architecture itself, it would be interesting, in addition to repeating the investigative effort, to compare our results with those arising from other systems, intra- or inter-organizational, with functionalities different from data sharing, and/or in other sectors of activity.

REFERENCES

Bidan, M., Rowe, F., & Truex, D. (2012). An empirical study of IS architectures in French SMEs: integration approaches. *European Journal of Information Systems*, *21*(3), 287-302.

Figueiredo, M., de Souza, C., Pereira, M., Nicolas Audy, J., & Prikladnicki, R. (2012). On the role of information technology systems architects.

Grinter, R. E. (1999, March). Systems architecture: product designing and social engineering. In *Proceedings of the international joint conference on Work activities coordination and collaboration* (pp. 11-18).

Hoorn, J. F., Farenhorst, R., Lago, P., & Van Vliet, H. (2011). The lonesome architect. *Journal of Systems and Software*, *84*(9), 1424-1435.

Hummel, P., Braun, M., Augsberg, S., & Dabrock, P. (2018). Sovereignty and data sharing. ITU Journal: *ICT Discoveries*, Special Issue No. 2, 23 Nov. 2018

Nielsen, P., & Sæbø, J. I. (2016). Three strategies for functional architecting: cases from the health systems of developing countries. *Information Technology for Development*, *22*(1), 134-151.

Poppe, O., Sæbø, J., & Nielsen, P. (2014, August). Architecting in Large and Complex Information Infrastructures. In *Scandinavian Conference on Information Systems* (pp. 90-104). Springer, Cham.

Rechtin, E., & Maier, M. W. (2010). *The art of systems architecting*. CRC press.

Robey, D., & Markus, M. L. (1984). Rituals in information system design. *MIS quarterly*, 5-15.

Sahay, S., Monteiro, E., & Aanestad, M. (2009). Configurable politics and asymmetric integration: Health e-infrastructures in India. *Journal of the Association for Information Systems*, *10*(5), 4.

Scheil, M. (2008). IT Architecturing: Reconceptualizing Current Notions of Architecture in IS Research.

Smolander, K. (2002, October). Four metaphors of architecture in software organizations: finding out the meaning of architecture in practice. In *Proceedings International Symposium on Empirical Software Engineering* (pp. 211-221). IEEE.

Smolander, K., Rossi, M., & Purao, S. (2008). Software architectures: blueprint, literature, language or decision?. *European Journal of Information Systems*, *17*(6), 575-588.

Star, S. L., & Griesemer, J. R. (1989). Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social studies of science*, *19*(3), 387-420.

Van den Berg, M., Tang, A., & Farenhorst, R. (2009, August). A constraint-oriented approach to software architecture design. In *2009 Ninth International Conference on Quality Software* (pp. 396-405). IEEE.

Van Heesch, U., & Avgeriou, P. (2011, June). Mature architecting-a survey about the reasoning process of professional architects. In *2011 Ninth Working IEEE/IFIP Conference on Software Architecture* (pp. 260-269). IEEE.

Vassilakopoulou, P., & Grisot, M. (2013). Exploring the concept of architecture in Technology and Organization studies. In *36th Information Systems Research Seminar in Scandinavia (IRIS 2013)*.