

Chapter 2

Developing Pan-European e-Government Solutions: From Interoperability to Installed Base Cultivation

Ole Hanseth

Abstract In this chapter I address some of the complexities of ICT systems like Pan-European e-Government solutions, the challenges we are regularly confronted with when developing such solutions, the kinds of strategies that can help us cope with these challenges, and the extent to which an EU strategy with a strong focus on interoperability is an appropriate approach to dealing with them. I conclude that complex solutions like those discussed here can appropriately be seen as IIs. These IIs are not designed from scratch: they just evolve. A strategy for developing such solutions must therefore concentrate on how to make an II evolve in the desired direction. Standards are indeed crucial to infrastructures but the way standards are developed and their various properties have to address the need for an infrastructure to be flexible in order to grow and adapt to changing user requirements. The way the concept of interoperability is understood and used implies a static view of infrastructures that does not take into account the need for flexibility.

2.1 Introduction

The European Union's e-Justice programme aims at developing a large number of ICT solutions that should contribute to the establishment of harmonised and integrated judicial services, i.e., the circulation of legal agency across all EU member states (Lanzara, Chap. 1). This is a very ambitious task. The development of well-working ICT solutions within the judicial area has proved to be a complex and demanding undertaking at the national level (Contini and Lanzara 2009). There are good reasons to believe that the development of integrated and harmonised services and ICT solutions at the EU level will be considerably more challenging due to the

O. Hanseth (✉)

Department of Informatics, University of Oslo, Oslo, Norway

e-mail: ole.hanseth@ifi.uio.no

increase in complexity, in terms of the number of ICT solutions, actors, languages, legislations, judiciary systems and traditions, etc. This chapter will discuss some issues related to what kind of strategy and approach is required for coping with this complexity.

The EU's e-Justice programme covers one area within an ambitious program aimed at developing what the EU calls pan-European e-Government solutions. These are solutions aiming at establishing what are referred to as pan-European e-Government Services, which the EU trusts '... will enable citizens and businesses from all member states to access e-Government services in all member states. In future these services will eliminate or reduce the current limitations on the free flow of people, goods, capital and services across all member states of the European Union.'

Activities continue to aim at establishing such pan-European e-Government solutions in many sectors, like health care and customs, in addition to the justice sector. The EU has developed a set of documents outlining its strategy for developing these solutions in general and for e-Justice in particular. A major question, then, is if this strategy is appropriate or how it could/should be improved.

The key concept in the EU's strategy and approach is interoperability. This is a concept that emerged and became popular in the computer communications standardisation field during the 1980s. It usually denotes what kind of communication and integration one wants to achieve between computer systems. The way to achieve interoperability is usually considered to be by reaching agreement on a set of shared standards. The term is particularly popular among communities involved in formal standardisation activities where standards are supposed to be established by means of bureaucratic committees and based on a top-down specification-driven approach. A typical example is the failed ISO/OSI effort, which was heavily sponsored by the EU and the governments of the OECD countries.

The pan-European e-Government solutions that are being developed include a number of new solutions at the same time as they are linking together numbers of existing (and modified) solutions in each member state. This means that the overall pan-European solutions will be very complex, and so, it follows, will the organisational arrangements of the development activities. Accordingly, a successful strategy for developing these solutions needs to address this complexity in a proper way.

2.2 Pan-European e-Government Solutions and Information Infrastructures

2.2.1 From Interoperability to Complexity

Increased processing power and higher transmission and storage capacity have made it possible to build increasingly integrated and versatile Information Technology

(IT) solutions, which have dramatically increased complexity (BCS/RAE 2004; Kallinikos 2007; Hanseth and Ciborra 2007). In addition, the complexity of IT solutions has been growing continuously as existing systems, new and old, have been increasingly integrated with each other. Complexity can be defined here as the dramatic increase in the number and heterogeneity of included components, relations and their dynamic and unexpected interactions in IT solutions (Hanseth and Lyytinen 2010).

The Software Engineering Institute (SEI) at Carnegie-Mellon University describes this trend as the emergence of Ultra-Large-Scale (ULS) systems (Northrop et al. 2006). The report argues that these systems will push far beyond the size of today's systems and systems of systems by every measure:

- Number of technological components of various kinds
- Number of people and organisations employing the system for different purposes
- Number of people and organisations involved in the development, maintenance and operations of the systems
- Amount of data stored, accessed, manipulated and refined
- Number of connections and interdependencies among the elements involved

The report argues further that the sheer scale of ULS systems will change everything: ULS systems will necessarily be decentralised in a variety of ways; developed and used by a wide variety of stakeholders with conflicting needs, so that they will be evolving continuously; and constructed from heterogeneous parts. Furthermore, people will not just be users of a ULS system; rather, they will be elements of the system. The acquisition of a ULS system will be simultaneous with its operation and will require new methods for control. These characteristics are, according to the report, emerging in today's systems of systems; in the near future, they will dominate.

The SEI report notes that the scale of ULS systems presents challenges that are unlikely to be addressed adequately by incremental research within the established paradigm. Instead, they require a broad new conception of both the nature of such systems and new ideas on how to develop them. We will need to look at them differently, not just as systems or systems of systems but also as socio-technical ecosystems.

The growth in complexity has brought to researchers' attention novel mechanisms to cope with complexity, such as architectures, modularity and standards (Baldwin and Clark 2000; Parnas 1972; Schmidt and Werle 1998). Another, more recent, line of research has adopted a more holistic, socio-technical and evolutionary approach, putting the growth in the combined social and technical complexity at the centre of empirical scrutiny (see, e.g., Edwards et al. 2007). These scholars view these complex systems as new types of IT artefacts and identify them by the generic label of Information Infrastructures (IIs) (e.g., Hanseth and Lyytinen 2010; Star and Ruhleder 1996; Tilson et al. 2010).

Hanseth and Lyytinen (2010) define an II, consistent with the characterisation of ULS systems above, as a shared, open (and unbounded), heterogeneous and evolving socio-technical system (called an installed base) consisting of a set of IT

capabilities and their user, operations and design communities. Typical examples of IIs are the Internet, solutions supporting the interaction among manufacturers along a supply chain and portfolios of integrated applications in organisations (often several thousand in number).

Pan-European e-Government solutions obviously fit this definition of ULS systems and the emerging paradigm around the concept of IIs seems an appropriate one to cope with the challenges that the development of such solutions raises.

2.2.2 The Key Characteristics of Information Infrastructures

The most distinctive feature of IIs and ULS systems is their overall complexity. Developing and managing IIs, then, is about understanding and managing complexity. More specifically, IIs, like all complex systems, are evolving and not designed from scratch. Thus, managing IIs means managing their evolution.

In reality, IIs are radically different from the way information and software systems are presented in the literature. Infrastructures have no life cycle—they are ‘always already present’. This is strictly true of some infrastructures, such as our road infrastructure, which has evolved through modifications and extensions ever since animals created the first paths. IT infrastructures certainly have a much shorter history, but an II like the Internet has now been around and constantly evolving for roughly 40 years since its inception in the late 1960s. The same is true for IIs like portfolios of integrated ISs in larger organisations. Developing IIs, then, requires approaches that are different from the traditional ‘design from scratch’ ones (Hanseth and Lyytinen 2010; Edwards et al. 2007; Tilson et al. 2010). It rather requires an approach, which is the opposite of this, i.e. an approach which sees designing a new II as modifying (changing and extending) the installed base, the existing IIs, so that the installed base evolves as far as possible towards what is desired (user requirements).

The evolution of IIs regularly involves a large number of actors. All these actors cannot be strictly controlled from one single point (like, for instance, a manager at the top of a hierarchically structured project organisation). They will often act independently. In the case of the Internet, there are thousands, if not millions, of actors developing new services on top of the existing Internet or adding new features to lower-level services, such as quality-of-service mechanisms. This means that even though there are many institutions (ICANN, IETF, etc.) that are involved in the governance of the Internet, the Internet is not evolving in a strictly planned or controlled way. Its evolution is mostly the aggregated result of the various autonomous actors’ actions. Institutions having responsibility for the governance of the Internet can have an impact and shape its evolution in a way similar to how we might influence the growth of an organism or a piece of land. The same is the case for large application portfolios. The individual applications and their relations (degree of integration) will change continuously and no single actor will have a

total overview of these changes. Accordingly, we call our approach to how we can shape the evolution of an II ‘installed base cultivation’ (Hanseth and Lyytinen 2010).

If infrastructures are ‘always already present’, are new infrastructures never emerging? Wasn’t the Internet a new infrastructure at some point in time? Yes, of course it was. New IIs are indeed emerging. This happens primarily in two ways. One, a system may be growing in terms of the number of users and along that path gradually changing from being a system (or application) of limited reach and range into a large-scale II. Email, for instance, was introduced into the Internet (or rather the Arpanet) at a time when the Net consisted of only four computers. Those computers (and the email service) did not constitute an II, but a distributed system of limited complexity. As the Net grew in terms of connected computers, services, developers and users, however, it increasingly took on the character of an infrastructure. The other way an II may emerge has been seen in most large organisations. Over time, the number of applications has continuously been growing at the same time as old and new applications have been increasingly integrated. During this process, the application portfolio has grown from being an initial, stand-alone item to a few loosely integrated ones and towards an increasingly more complex II.

The next section will present a brief outline of the key challenges related to the management of the evolution of IIs and the kind of ‘tools’ available for this task.

2.3 Design Challenges and Management Tools

Successful development of IIs requires appropriate management of the tension between standardisation and flexibility. Key to the management of this tension is the management of two main, and related, challenges: the bootstrapping and adaptability challenges (Hanseth and Lyytinen 2010). I will briefly describe these challenges and then discuss how a strategy focusing on interoperability is able to cope with them.

2.3.1 Design Dilemmas

2.3.1.1 The Tension Between Standards and Flexibility

Infrastructures are made up of a huge number of components. Accordingly, standards defining the interfaces between components are essential features of IIs and the specification and implementation of standards are important activities in the establishment of IIs. Standards are closely associated with stability. This is the case partly because keeping a standard stable is required so that it is possible,

for example, to store data today and read them at a later time. Standards are also stable, however, because widely diffused standards are so hard to change when necessary (because of the lock-in problem). Nonetheless, standards need to change over time. Just like information systems, IIs need to change and adapt to changing user requirements, and some of these changes mean that implemented standards need to change, too. At the same time, the scaling and growth of an II can generate the need for changes in standards even though user requirements stay unchanged (Hanseth et al. 1996; Tilson et al. 2010). Accordingly, IIs will evolve as a dynamic driven by a tension between standards (stability, uniformity) and flexibility (change, heterogeneity). Managing this tension is a key to the management of IIs (Hanseth et al. 1996).

2.3.1.2 Bootstrapping: Adaptability

The dynamic complexity of IIs poses a chicken-egg problem for the II designer that has been largely ignored in the traditional approaches. On the one hand, IT capabilities embedded in IIs gain their value by being used by a large number of users demanding rapid growth of the user base. Consequently, II designers have to come up early on with solutions that persuade users to adopt while the user community is non-existent or small. This requires II designers to address head on the needs of the very first users before addressing the completeness of their design or scalability. This can be difficult, however, because II designers must also anticipate the completeness of their designs. This defines the bootstrap problem of II design. On the other hand, when the II starts expanding by benefitting from the network effects, it will switch to a period of rapid growth. During this growth, designers need to heed unforeseen and diverse demands and produce designs that cope technically and socially with these increasingly varying needs. This demands infrastructural flexibility so that the II adapts technically and socially. This defines the adaptability problem of II design (Edwards et al. 2007; Hanseth and Lyytinen 2010). Clearly, these two demands can contradict and generate tensions at any point of time in II design (Edwards et al. 2007).

2.3.1.3 Interoperability and Design Dilemmas

Among those who use it, the concept of interoperability has mostly positive connotations: according to the received wisdom, interoperability is the only thing one needs to focus on in order to successfully develop large-scale distributed solutions, and the more interoperability the better. Unfortunately, things are more complicated than that. If we relate the concept of interoperability to the design dilemmas mentioned above, a different picture emerges. In actuality, the concept of interoperability and the EU's Interoperability strategy and framework do not help us much in understanding or coping with any of them. The EU's strategy for developing pan-European e-Government solutions focuses on standards. Nothing

is said about the need for flexibility, the tensions between standardisation and flexibility, the deeper issues related to the complexity of the solutions and the complexities of their user and development communities. This means that a strategy for coping with the bootstrapping or adaptability challenges is totally absent.

Research on the development of pan-European e-Government solutions in the customs area, the EU's eCustoms programme, has revealed conflicting issues and demands related to interoperability (Henningsson and Hanseth 2011). The aim of the eCustoms programme has been to develop ICT solutions for electronic customs declaration in all member states, harmonise customs declaration procedures among the member states and integrate these solutions in order to reduce the costs of traders related to customs declarations (Henningsson and Hanseth 2011). In this initiative, it emerged that there was a conflict between interoperability at the national level (i.e., interoperability between the solutions involved in the overall customs declaration processes) and interoperability between the solutions at the EU level. The more interoperability at the national level, the less interoperability could be achieved at the EU level. It also turned out to be the case that the actors involved at the national level were working more closely together at the national level than at the EU level, meaning that interoperability at the national level was given the highest priority. The overall result of this was a more fragmented system for customs declaration at the EU level and increased costs for traders (Henningsson and Hanseth 2011)!

2.3.2 Management Tools

So what kinds of 'tools' are available for managing the evolution of IIs—or for installed base cultivation? The answer given here is process strategy, architecture and governance regime. The rationale behind the focus on these three aspects is, first of all, the fact that these three 'tools' are what development efforts are all about: the steps to be taken to develop some new technology (bottom-up or top-down, incremental/iterative or 'big bang', evolutionary and learning driven or specification driven, etc.); the architecture and overall design of the technology (the modularisation of a system determines how and how easy it may be maintained and modified); and how to govern, manage and organise the effort. Secondly, these 'tools' have been at the centre of extensive discussions and research on the evolution of the Internet, and they have been key factors behind its success: an experimental bottom-up development strategy (which includes a strategy and rules for bottom-up development and settlement of standards), the end-to-end architecture and distributed control and governance structures combined with Open Source software licenses (Lemley and Lessig 2000; Hanseth et al. 1996; Benkler 2006; Zittrain 2006).

Traditionally, the management of ICT has focused on the management of projects developing ICT solutions. Such projects are typically organised as a hierarchy of sub-projects, each with a sub-project manager. Each manager has the right to

make decisions within the domain of the sub-projects and give instructions to the managers at the level below. The management of such a project organisation is normally supported by various management tools, such as detailed plans and establishment of milestones. In addition, the production of detailed plans and the monitoring of the progress made in the project—if it is progressing according to plan or not—are supported by various computer-based project management tools. Together, this package of project organisation, decision rights and management tools is an example of what I call a governance regime. Governing the complexity of IIs requires new and different governance regimes. In the case of the Internet, for example, its successful evolution has been shaped by a governance regime consisting of a few central institutions, like IETF and ICANN. Another important part of this regime has been the fact that most of its technology has been distributed using Open Source licenses (like the GNU Public License). Maybe the most important feature of the governance regime, however, has been the organising of the development activities as a loosely connected network of individuals who coordinate their work through extensive use of email and by making all software and relevant information publicly available on ftp and Web servers.

Software and IS development often takes place by following specific methodologies. The central element of such methodologies is a specification of the steps to be taken, and in which sequence, to develop a specific IS solution. This is what I call a process strategy. The complexity of IIs requires process strategies different to those prescribed for traditional IS development efforts. In particular, we need process strategies that address the role of network externalities and path-dependence; that is, we need strategies that address the bootstrap and adaptability problems.

The architecture of an IS is traditionally considered important for its maintenance. In general, modularisation is an important strategy for coping with complexity and, in the case of IIs, the architecture plays a crucial role. This is illustrated by the role attributed to the Internet's architecture in explaining its successful evolution. The Internet's so-called end-to-end architecture (in which the functionality is located in the ends of the network, that is, in the computers connected to the Internet, and not in the network itself, which has been the case within traditional telecommunication networks) has made the Internet extremely flexible in the sense that anybody having a computer connected to the Internet can develop and provide new services.

The management of IIs, then, requires process strategies, architectures and governance regimes that in combination make an II evolve along the desired path. Exactly which combinations are appropriate for specific IIs is still a major research issue.

The concept of interoperability relates particularly to architecture. Accordingly, in the next section, a richer picture is offered of how architectural features shape the evolution of infrastructures and of the roles architectural features play in the cultivation of an installed base.

2.4 Architectural Shaping of Infrastructure Evolution

Traditionally, research on technological architectures in general and ICT architectures in particular has focused on how to decompose a system into modules so that system flexibility is maximised. This is assumed to be best achieved through loose coupling among components and strong internal cohesion (Henfridsson et al. 2009; Parnas 1972). Loose coupling, as opposed to tight coupling, between components means that the inner working of a component is largely irrelevant and can be hidden from other components (Baldwin and Clark 1997; Sanchez and Mahoney 1996). This is what Parnas (1972) called encapsulation. Loosely coupled components are consequently easier to modify and more available for new relationships in the reconfiguration of a modular system. Research on technological architectures has traditionally concentrated on one single software system. More recently, however, as the number of systems has been growing and their integration has increased, more attention has been directed towards architectures specifying the relations between individual solutions. This research has directed much of its focus towards Service Oriented Architectures (SOA) (Vassiliadis et al. 2006), in which the modular structure consists of services. The implementation of SOA may vary, from simple ASP solutions to Web services, to more complex SOAs based on Enterprise Software Bus middleware (Rosen et al. 2008).

The literature on ICT architectures focuses mainly on projects and solutions located within one single organisation. Pan-European e-Government solutions are different in the sense that they will be shared by a large number of independent organisations. Such large-scale solutions raise a lot of new challenges. These challenges are addressed within a growing body of research—to which the research presented here belongs—conceptualising these large-scale solutions' IIs (see for instance Ciborra et al. 2000; Edwards et al. 2007; Hanseth et al. 1996; Star and Ruhleder 1996; Tilson et al. 2010). To some extent, this literature also addresses architectural issues. It does not relate risks to specific architectures, i.e., specific ways of modularising (or decomposing) an II, but to the degree of modularisation, i.e., to what extent the modules are loosely or tightly coupled. The literature is, for instance, demonstrating how larger IIs emerge as responses to the felt need of tighter integration of applications to enable more smooth information flow and sharing in order to enable more smooth co-ordination of work tasks and more efficient ways of organising them (Hanseth and Ciborra 2007). Tighter integration leads to more complexity and new challenges for managing the IIs.

Here, I will try to move beyond this research by focusing on how specific architectures, i.e., specific ways of modularising, have an impact on challenges related to the management of IIs. I will do so by drawing upon three emerging streams (Table 2.1) of research on technological architectures that focus on, and demonstrate, how architectures relate to a broad range of issues beyond the flexibility of the technological artefact.

Table 2.1 Three emerging streams of research on ICT architectures

Stream	Key issue	References
Strategic architecting	ICT architecture and platforms win technology wars	Morris and Ferguson (1993), Tiwana et al. (2010), Woodard (2008), Rodon et al. (2012)
Mirroring and structural alignment	Technological architectures and organisational structures are mirroring each other	Henderson and Clark (1990), Baldwin and Clark (2000), Garud et al. (2002), Colfer and Baldwin (2010)
Innovations and generativity	Generativity denotes a technology's capacity to create innovation driven by large and uncoordinated networks of actors	Saltzer et al. (1984), Abbate (1994, 1999), Benkler (2006), Zittrain (2006)

2.4.1 Strategic Architecting

Morris and Ferguson (1993) argued that 'architecture wins technology wars' in complex high-tech markets. Companies being successful over time in such markets achieve this not primarily because of the superior qualities of their products or their production processes but because they control architectures that have become de facto standards in a product domain. Important examples supporting this hypothesis are IBM in the mainframe area and Intel and Microsoft in the PC (desktop) area. Morris and Ferguson argue that Borland and Lotus were losers in the competition with Microsoft for exactly this reason (lack of control of architecture), although they at a certain point in time had superior product families in terms of functionality. They conclude that technological architectures are crucial for the long-term competitiveness and commercial success of high-tech firms.

Research on architectural strategies is limited, but growing. In particular, there is a rapidly growing number of what Tiwana et al. (2010) call platform-centric ecosystems and a correspondingly growing research interest related to such platforms covering the importance of platform-centric architectures, how specific platforms emerge as dominant within an ecology and strategies that platform owners can pursue in order to control the evolution of the platform as well as the whole ecology (Cusumano and Gawer 2002; Gawer 2009). Tiwana et al. (2010) mention a number of areas where such platforms are emerging: browsers (e.g., Firefox, Chrome and Opera), smartphone operating systems (iPhone, Android), Web services (Google Payments, Amazon Elastic Cloud), social media (Facebook, Apple's Ping), marketplaces (SABRE, eBay) and gaming consoles (Xbox, Apple's iPod Touch, Sony PlayStation). Platform-centric architectures are examples of what Jason Woodward (2008) calls architectural control points, i.e., architectures that contain certain components of strategic importance in the sense that if an actor is controlling the evolution of this component (i.e., the platform), she can control the evolution of the whole ecology.

Rodon et al. (2012) report on a case in which the issue of strategic architecting and architectural control points were central. In mid-2004, the Catalan Health

Service (CHS) set the foundations for the development of an electronic prescription system in Catalonia. The CHS started the project in a spirit of cooperation and invited all the stakeholders, among them the Catalan College of Pharmacists (CCP), the natural spokespersons of pharmacists, to participate in the project. The CCP described its role in the project, and particularly in the IT architecting phase, as a way to protect the interests of community pharmacies and minimise any potential negative impact on the collective of pharmacists. Early in the project's history, CHS designed an architecture for the II with a central database where all prescriptions were stored. This architecture did not assign any role to the CCP in the operations of the II. The CCP strongly objected to this, and through series of strategic and political manoeuvres the college was able to make modifications to the architecture which meant that the pharmacies were accessing prescriptions in a database operated by the CCP that was mirroring the database operated by CHS and which General Practitioners accessed. Through this strategy, the CCP was able to modify the architecture so that it included one component that the CCP controlled, and through the control of this component the CCP also obtained substantial control over the whole infrastructure.

2.4.2 Mirroring and Structural Alignment

Recently, the scope of research related to technological architectures has expanded. This includes research related to the relations between a product's architecture and the structure of the organisation developing or producing it, and how these relations shape each other's evolutionary dynamics. Within the field of Organisation Studies, substantial empirical material has been collected supporting the hypothesis that technological architectures and organisational structures are mirroring each other. In a critical test of this hypothesis, Colfer and Baldwin (2010) found that it was strongly supported by empirical evidence, both at the firm and at the industry level. Henderson and Clark (1990) found that as a consequence of this mirroring, established firms are systematically unable to come up with architectural innovations—only component innovations. This mirroring is also an important part of Clayton Christensen's (1997) explanation for established firms' inability to come up with disruptive innovations.

At the industry level, the mirroring of technological architecture and organisational structure also has important implications. For instance, Utterback (1994) argues that virtually all industries are evolving according to a life-cycle process. In the early phases, the diversity of products and producers proliferates until the industry reaches a certain level of maturity and a de facto standard architecture (often called the 'dominant design') emerges. At this stage, the degree of product variety and number of producers decrease dramatically at the same time as the products are assembled out of standardised components produced by a growing number of component suppliers. This transformation occurred in the car industry in the 1920s and in the computer industry in the 1980s. Baldwin and Clark (1997)

argue that the evolution of the concept of modularity has had a significant impact on the evolution of high-tech industries in general and the computer and software industry in particular. For this reason, increasingly complex but modular architectures have emerged and, because of their high degree of modularity, computer and software architectures have changed considerably over time at the same time as the individual modules have evolved. Overall, they argue, the computer and software industries have changed over time in terms of a co-evolution of modular technological architectures and what they call modular clusters (of companies). Further, 'managing in the modular age' (Baldwin and Clark 1997) requires new strategies: managers need to focus on, and understand, how the overall ecology (the technological architecture and the structure of the modular cluster) is evolving and how their companies' products and overall strategy can continuously adapt to this.

The research presented above focuses on the relations between a product and its producer (both at the company and industry level). This relationship is also found to be important within the II domain. For example, significant investments have been made by the powerful actors in the European mobile communications industry to implement the very successful Japanese mobile Internet service *i-mode* in many European countries. According to Tee and Gawer (2009), however, they all failed because of the differences in organisational structures in the mobile communications industry in Japan on the one hand and in Western European countries on the other; *i-mode*'s architecture was congruent with the former but not the latter. Sæbøe et al. (2011) have been doing research on the implementation of a national health II in Malawi. The actors involved in this effort tried to build the II based on a set of standards initially developed in South Africa. They found that in order to succeed, they had to redesign the II's overall architecture and the individual standards so that they reflected relations between the vendors of the various Health Information Systems being integrated through the national II. In the case of IIs, however, it is not only the mirroring of the structures of the producer and the product that matters. Forster and King (1995) have, for instance, found the mirroring of the structure of an II and the structure of the user community to be a critical success factor—or the cause of failure. They found that efforts aiming at adapting the very successful booking systems for air passenger transport to air cargo transport failed even though the functionality required to support both processes is more or less exactly the same. Forster and King argue that these efforts failed because of the differences in organisational structures in the air cargo and passenger transport industries.

Some research is also emerging on interactions between an II's architecture and the unfolding of an II's development and implementation processes. In a comparative study of two Danish projects aimed at developing nationwide Electronic Patient Records infrastructures, Aanestad and Jensen (2011) found that the one project that developed an II based on a modular architecture that allowed for incremental implementation emerged as the national II. The architecture of this solution was mirroring the structure of the collaborative arrangements with health care. The other project failed in spite of extensive funding and political support, according to Aanestad and Jensen, because it was based on an architecture that required all modules to be implemented before the solutions could be used for meaningful purposes.

Table 2.2 Comparing the two approaches

	INA approach	SPA approach
ICT architecture	Many applications, sending messages to each other	Single application, distributed to clients
Project organisation	Co-ordinated teams in many organisations	Single team, within one organisation
Overall risk	High	Medium to low

2.4.3 *Architecture, Organisational Complexity and Project Risks*

Hanseth and Bygstad (2012) have studied the evolution of 10 IIs within the health care sector of Norway for a period of more than 20 years. They found that the chosen architectures had strong implications for the organisation of the development activities and that different architectures generated development organisations of hugely different complexities, which again had a huge impact on development costs and outcomes. Hanseth and Bygstad also found that the infrastructures examined were developed by basing them on two different architectures. The first architecture was an integrated part of a broader paradigm, which they call the EDI paradigm, a paradigm which says that an II should be built by extending existing applications so that they can send and receive (standardised) messages. They call this architecture ‘Institutional Interface/Application Centric Architecture’, or INA, to reflect the fact that the overall infrastructure is built by focusing on existing applications and that the main interfaces between the main modules of the infrastructure reflects the ‘interfaces’ between the organisations. The second architecture they call ‘Service Provider/Communication System Centric Architecture’, or SPA, to point out that the infrastructures based on it are focused on service providers offering service consumers electronic access to their services and that the infrastructure is built by developing a separate module taking care of as much as possible of the communication (and leaving the applications untouched).

The architectures and the roles they play in shaping the respective infrastructures will be described in more detail below. The main findings are summarised in Table 2.2.

2.4.3.1 **The INA Approach**

All the INA projects were problematic. They all suffered from various problems associated with complexity: the large number of involved actors, the heterogeneity of technical solutions and the many dependencies that created postponements and frictions when schedules were not kept. The key to these problems are illustrated in Fig. 2.1. The chosen ICT architecture was (as usual in EDI solutions) based on the data flow between the involved organisations. This led to a relatively complex ICT architecture, with a large number of messages flowing between a large number

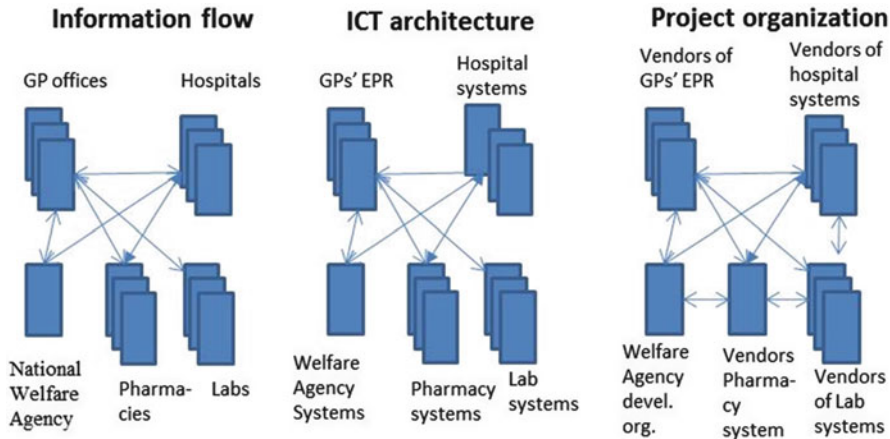


Fig. 2.1 The INA approach

of systems, meaning that many local applications had to be modified in order to produce and receive messages. In practice, each vendor had to develop their own client modules of the solution. For example, in the ePrescription project, of the six vendors of EPR systems present in the Norwegian market (three GP and three hospital systems), each had to make their own version of a quite complex piece of software with exactly the same functionality.

Further, the INA solution implies a project organisation with participants from all involved actors, usually organised as a number of sub-projects, with a central co-ordinating actor. As observed by van der Aalst (1999), this increases the challenge of co-ordination. The co-ordinating actor cannot usually instruct the other participants (since they represent independent organisations), but will have to manoeuvre by means of compromises and politics. This combination of technical and organisational complexity increases significantly the risk of postponements and even failure, as revealed by the cases presented.

2.4.3.2 The SPA Approach

The SPA projects, although different in type and scope, were all successful. The analysis shows that the overall reason was the chosen ICT architecture. This architecture did not reflect the information flow between the numerous organisations but was based on a solution from one application service provider. As illustrated in Fig. 2.2, this greatly simplified the design solution.

In the SPA-based solutions, the important interfaces within the overall solution are the interfaces between the communication solution and the applications, not the interfaces between the different modules of the communication system that are running within different organisations. In the SPA architecture, there is a tight coupling between the different components of the communication system and weak

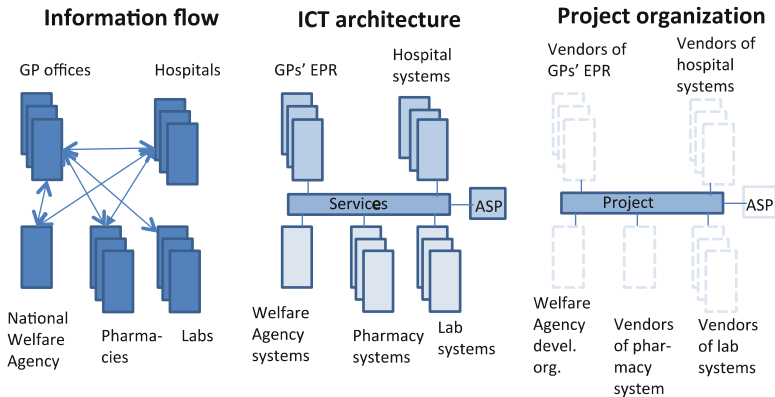


Fig. 2.2 The SPA approach

coupling between the communication system and the applications, whereas the INA-based solutions are based on tight coupling between the applications and the communication system (i.e., the module running within the same institution) and loose coupling between the modules within the communication system.

The most crucial aspect of the SPA-based solutions, in the context of this chapter, is the fact that the architecture of the communication solution allows the complete solution to be developed by one single project team within one single formal organisation. Only minor development work needs to be done by other organisations, such as application vendors. In more general terms, the important aspect of the SPA architecture is the fact that the complexity of the development organisation becomes dramatically reduced compared to that of the INA-based solutions.

To sum up, the increased risks of the INA approach compared to the SPA approach are:

- A more complex technical solution, with a higher technical risk;
- A more complex project organisation, with very challenging co-ordination;
- Higher costs, because the vendors will all have to develop their own client solutions; and
- Higher implementation risk, because the INA solution requires that all actors start at the same time. Such a 'big bang' strategy is more risky than an incremental approach (Bygstad et al. 2010) that the SPA allows for.

2.4.4 Innovations and Generativity

The role of technological architecture on the evolution of an II has been extensively discussed in relation to the Internet. Its end-to-end architecture is widely held

to be a prime and distinguishing feature of the Internet compared to traditional telecommunication (Saltzer et al. 1984; Abbate 1994, 1999). This means that the functionality ('intelligence') of the overall network is located in the ends—its terminals (i.e., computers in the Internet case)—rather than in the network, as in the case of traditional telecom architecture. This end-to-end architecture made the Internet extremely flexible: anybody who had a computer could develop and provide a new service. Abbate (1999) also illustrates how the successful development of Internet services has been based on an approach in which each layer of services acted as a platform for experimental development of the next layer. The importance of the end-to-end architecture has also been forcefully argued by Lawrence Lessig (2001) in debates about issues like network neutrality.

Yochai Benkler (2006) developed this 'end-to-end' argument one step further by underscoring the mutual dependence of the end-to-end architecture of the network and (easily) programmable terminals, in the form of general purpose computers. Benkler makes a conceptual contrast between programmable computers and appliances. An appliance is a device with a limited and well-defined set of functions that (normally) cannot be modified after the users have bought it. Typical examples include washing machines, radios and phones (traditional ones, at least). Most such devices have computers inside but their software cannot usually be modified by their users. Benkler also makes a strong link between the Internet's architecture and its model for organising and governing development activities. Central here is the Open Source model, which implies a loosely organised system with distributed control, and the Open Source software license.

Jonathan Zittrain takes this argument one step further by means of the concept of generative technology. Generativity 'denotes a technology's overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences' (Zittrain 2006). Zittrain argues that the grid of PCs connected by the Internet has developed in such a way that it is consummately generative. Zittrain defines generativity in more detail as a function of a technology's capacity for leverage across a range of tasks, adaptability to a range of different tasks, ease of mastery and accessibility. Leverage describes the extent to which these objects enable valuable accomplishments that would otherwise be either impossible or not worth the effort to achieve. Adaptability refers to the breadth of a technology's use without change and the readiness with which it may be modified to broaden its range of uses. A technology's ease of mastery reflects how easy it is for broad audiences to adopt and adapt it: how much skill is necessary to make use of its leverage for tasks they care about, regardless of whether the technology was designed with those tasks in mind. Accessibility refers to the ease with which people can come to use and control a technology (along with what information might be required to master it).

Even though the Internet's end-to-end architecture has contributed significantly to the Net's successful evolution, its future is uncertain. The Internet's growth has generated new demands. For instance, issues such as security and the illegal distribution of spam, music and child pornography have become major concerns. Many actors are arguing that these issues demand technological mechanisms (filters

and security technologies, such as trusted computing) to be put into the Net. Network providers also argue that they have to implement quality of service mechanisms to guarantee better services for those willing to pay for them, in order to afford further expansion of their bandwidth capacities. Scholars such as Benkler (2006), David (2005), Lemley and Lessig (2000), Wu (2003) and Zittrain (2006) are worried that the proposals for addressing these issues will destroy the end-to-end architecture and turn the Internet into an appliance, as well as dramatically reducing the rate of innovations related to the Internet in the future. Other researchers argue that the Internet's architecture has to change to allow further growth (Clark et al. 2005). This relates to 'tussles in cyberspace' emerging out of the growth in number and variety of Internet Service Providers. This makes their relationships complex and the conditions for sustainable and co-ordinated growth of the Internet are eroding. A new architecture is also considered necessary to maintain, or preferably enhance, the Internet's resilience (Trimintzios 2011).

2.5 Conclusion

In this chapter, I have addressed some of the complexities of ICT systems like pan-European e-Government solutions, the challenges we are regularly confronted with when developing such solutions, the kinds of strategies that can help us cope with these challenges, and the extent to which an EU strategy with a strong focus on interoperability is an appropriate approach to dealing with them.

I have concluded that complex solutions like those discussed here can appropriately be seen as IIs. These IIs are not designed from scratch: they just evolve. A strategy for developing such solutions must therefore concentrate on how to make an II evolve in the desired direction. Standards are indeed crucial to infrastructures but the way standards are developed and their various properties have to address the need for an infrastructure to be flexible in order to grow and adapt to changing user requirements. The way the concept of interoperability is understood and used implies a static view of infrastructures that does not take into account the need for flexibility.

I have argued that there are three 'tools' that are of particular importance when discussing how to control the evolution of infrastructures: process strategies, architectures and governance regimes. Of these three 'tools', the concept of interoperability is most directly linked to architecture. For this reason, I have presented existing research on how an infrastructure's architecture is linked up to a broad range of organisational and strategic issues. This research demonstrates the breadth of ways in which an architecture shapes the evolution of an infrastructure by linking up and interacting with a huge range of organisational features. This implies that an appropriate strategy for developing pan-European e-Government solutions need to be based on a much more nuanced and richer understanding of architectural issues.

References

- Aanestad, M., and T. Jensen. 2011. Building nation-wide information infrastructures in health care through modular implementation strategies. *The Journal of Strategic Information Systems* 20(2): 161–176.
- Abbate, J. 1994. The Internet challenge: Conflict and compromise in computer networking. In *Changing large technical systems*, ed. J. Summerton, 193–210. Boulder: Westview Press.
- Abbate, J. 1999. *Inventing the Internet*. Cambridge, MA: MIT Press.
- Baldwin, C.Y., and K.B. Clark. 1997. Managing in an age of modularity. *Harvard Business Review* 75(5): 84–94.
- Baldwin, C., and K. Clark. 2000. *Design rules*. Cambridge, MA: MIT Press.
- BCS/RAE. 2004. The challenges of complex IT projects. The report of a working group from the Royal Academy and the British Computer Society. <http://www.bcs.org/upload/pdf/complexity.pdf>. Accessed Aug 2009.
- Benkler, Y. 2006. *The wealth of networks: How social production transforms markets and freedom*. New Haven: Yale University Press.
- Bygstad, B., P.A. Nielsen, and B.E. Munkvold. 2010. Four integration patterns: A socio-technical approach to integration in IS development. *Information Systems Journal* 20(1): 53–80.
- Christensen, C.M. 1997. *The innovator's dilemma: When new technologies cause great firms to fail*. Boston: Harvard Business School Press.
- Ciborra, C.U., K. Braa, A. Cordella, B. Dahlbom, A. Failla, O. Hanseth, V. Hepsø, J. Ljungberg, E. Monteiro, and K.A. Simon. 2000. *From control to drift: The dynamics of corporate information infrastructures*. Oxford: Oxford University Press.
- Clark, D.D., J. Wroclawski, K.R. Sollins, and R. Braden. 2005. Tussle in cyberspace: Defining tomorrow's Internet. *IEEE/ACM Transactions on Networking* 13(3): 464–475.
- Colfer, L., and C. Baldwin. 2010. The mirroring hypothesis: Theory, evidence and exceptions. Harvard Business School working papers no. 10-058.
- Contini, F., and G.F. Lanzara (eds.). 2009. *ICT and innovation in the public sector: European studies in the making of e-Government*. Basingstoke: Palgrave Macmillan.
- Cusumano, M.A., and A. Gawer. 2002. The elements of platform leadership. *MIT Sloan Management Review* 43: 51–58.
- David, P.A. 2005. The beginnings and prospective ending of “end-to-end”: An evolutionary perspective on the Internet's architecture. SIEPR Discussion Paper No. 01-04. <http://www.stanford.edu/group/siepr/cgi-bin/siepr/?q=system/files/shared/pubs/papers/pdf/01-04.pdf>. Accessed 18 Jan 2013.
- Edwards, P., S. Jackson, G. Bowker, and C. Knobel. 2007. Report of a workshop on ‘History and theory of infrastructures: Lessons for new scientific infrastructures’. University of Michigan, School of Information. <http://www.si.umich.edu/InfrastructureWorkshop/documents/UnderstandingInfrastructure2007.pdf>. Retrieved 15 March 2007.
- Forster, P.W., and J.L. King. 1995. Information infrastructure standards in heterogeneous sectors: Lessons from the worldwide air cargo community. In *Standards policy for information infrastructure*, ed. B. Kahin and J. Abbate, 148–177. Cambridge, MA: MIT Press.
- Garud, R., S. Jain, and A. Kumaraswamy. 2002. Institutional entrepreneurship in the sponsorship of common technological standards: The case of Sun Microsystems and Java. *Academy of Management Journal* 45(1): 196–214.
- Gawer, A. 2009. Platforms, markets and innovation: An introduction. In *Platforms, markets and innovation*, ed. A. Gawer, 1–16. Cheltenham: Edward Elgar Publishing.
- Hanseth, O., and B. Bygstad. 2012. ICT architecture and project risk in inter-organizational settings. ECIS 2012 Proceedings. Paper 130. <http://aisel.aisnet.org/ecis2012/130mt1312>. Accessed 18 Jan 2013.
- Hanseth, O., and C. Ciborra (eds.). 2007. *Risk, complexity and ICT*. Cheltenham: Edward Elgar Publishing.

- Hanseth, O., and K. Lyytinen. 2010. Design theory for dynamic complexity in information infrastructures: The case of building Internet. *Journal of Information Technology* 25(1): 1–19.
- Hanseth, O., E. Monteiro, and M. Hatling. 1996. Developing information infrastructure: The tension between standardization and flexibility. *Science, Technology and Human Values* 21(4): 407–426.
- Henderson, R.M., and K.B. Clark. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly* 35: 9–30.
- Henfridsson, O., L. Mathiassen, and F. Svahn. 2009. Reconfiguring modularity: Closing capability gaps in digital innovation. Sprouts: Working papers on information systems. <http://sprouts.aisnet.org/9-22>. Accessed 18 Jan 2013.
- Henningsson, S., and O. Hanseth. 2011. The essential dynamics of information infrastructures. ICIS 2011 Proceedings. Paper 14. <http://aisel.aisnet.org/icis2011/proceedings/projmanagement/14>. Accessed 18 Jan 2013.
- Kallinikos, J. 2007. Technology, contingency and risk: The vagaries of large-scale information systems. In *Risk, complexity and ICT*, ed. O. Hanseth and C. Ciborra, 46–74. Cheltenham: Edward Elgar Publishing.
- Lemley, M.A., and L. Lessig. 2000. The end of end-to-end: Preserving the architecture of the Internet in the broadband era. UC Berkeley Law research paper no. 37. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=247737. Accessed 18 Jan 2013.
- Lessig, L. 2001. *The future of ideas*. New York: Random House.
- Morris, C.R., and C.H. Ferguson. 1993. How technology wins technology wars. *Harvard Business Review* 71(2): 86–95.
- Northrop, L., P. Feiler, R.P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivanand, and K. Wallnau. 2006. *Ultra-large-scale systems: The software challenge of the future*. Carnegie Mellon Software Engineering Institute.
- Parnas, D.L. 1972. A technique for software module specification with examples. *Communications of the ACM* 15(5): 330–336.
- Rodon, J., J. Trullen, L. Silva, and F. Sese. 2012. Counter-designing power in healthcare interorganizational information systems: The design of an electronic prescription system. Unpublished manuscript.
- Rosen, M., B. Lublinsky, K.T. Smith, and M.J. Balcer. 2008. *Applied SOA: Service-oriented architecture and design strategies*. Indianapolis: Wiley.
- Sæbø, J.I., E.K. Kossia, O.H. Titlestada, R.R. Tohouria, and J. Braaa. 2011. Comparing strategies to integrate health information systems following a data warehouse approach in four countries. *Information Technology for Development* 17(1): 42–60. doi:10.1080/02681102.2010.511702.
- Saltzer, J.H., D.P. Reed, and D.D. Clark. 1984. End-to-end arguments in system design. *ACM Transactions on Computer Systems* 2(4): 277–288.
- Sanchez, R., and J.T. Mahoney. 1996. Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal* 17: 63–76.
- Schmidt, S.K., and R. Werle. 1998. *Coordinating technology: Studies in the international standardization of telecommunications*. Cambridge, MA: MIT Press.
- Star, S.L., and K. Ruhleder. 1996. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research* 7(1): 111–134.
- Tee, R., and A. Gawer. 2009. Industry architecture as a determinant of successful platform strategies: A case study of the I-mode mobile Internet service. *European Management Review* 6: 217–232.
- Tilson, D., K. Lyytinen, and C. Sørensen. 2010. Research commentary—digital infrastructures: The missing IS research agenda. *Information Systems Research* 21(4): 748–759.
- Tiwana, A., B. Konsynski, and A.A. Bush. 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research* 21(4): 675–687.

- Trimintizos, P., ed. 2011. Inter-X: Resilience of the Internet interconnection ecosystem. Summary Report Enisa. <http://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-infrastructure-and-services/inter-x/interx/report/interx-report>. Accessed 18 Jan 2013.
- Utterback, J. 1994. *Mastering the dynamics of innovation: How companies can seize opportunities in the face of technological change*. Boston: Harvard Business School Press.
- Van der Aalst, W.M.P. 1999. Process-oriented architectures for electronic commerce and interorganizational workflow. *Information Systems* 24(9): 639–671.
- Vassiliadis, B., A. Stefani, J. Tsaknakis, and A. Tsakalidis. 2006. From application service provision to service-oriented computing: A study of the IT outsourcing evolution. *Telematics and Informatics* 23(4): 271–293.
- Woodard, C.J. 2008. Architectural control points. Research Collection School of information systems. <http://works.bepress.com/cjwoodard/14>. Accessed 18 Jan 2013.
- Wu, T. 2003. Network neutrality, broadband discrimination. *Journal of Telecommunications and High Technology Law* 2: 141. doi:10.2139/ssrn.388863. <http://ssrn.com/abstract=388863>. Last access 18 January 2013.
- Zittrain, J. 2006. The generative Internet. *Harvard Law Review* 119: 1974–2040.