Ansvarig
Claes-Olof Olsson, Sambruk

Upprättat av
Sven-Håkan Olsson, DocAccount
Anders Lindgren, Know IT

Verva
Karl-Mårten Karlsson
Box 214
101 24 STOCKHOLM

Synpunkter på det europeiska ramverket för interoperabilitet, EIF 2.0

Sammanfattning

Sambruk[1] meddelades under juli 2008 av Verva om möjligheten att lämna synpunkter på det reviderade europeiska ramverket för interoperabilitet, EIF 2.0 (nedan benämnt EIF2).

Sambruk inkommer därför med nedanstående synpunkter. Eftersom ursprungs-dokumentet är på engelska, och eftersom Verva kommer att sammanställa synpunkter till IDABC inom EU-kommissionen har vi för enkelhets skull valt att skriva resten av synpunktsdokumentet på engelska.

Sambruk[2] has the view that a lot of the content of the EIF2 document is of great value.

However, we would like to point out the following:

- There is a large risk that the impact of EIF2 on the MS[3] plans/implementations will not be powerful enough. Thus, there is a risk that interoperability will not appear in any substantial degree.

  Because of basic principles of the EU, IDABC will not have the power to decide about interoperability standards etc for the MS – IDABC can only

[1] Sambruk är en medlemsförening med fokus på gemensam verksamhetsutveckling och sambruk av kommunala e-tjänster. I dagsläget är c:a 80 kommuner medlemmar. SKL och Verva ingår i styrelsen. Se även www.sambruk.se .
[2] Sambruk is a co-operation organization for municipalities in Sweden. It focuses on business development and on e-services. Approximately 80 municipalities are members. Also, refer to www.sambruk.se.
[3] MS – Member State

publish guidelines and recommendations.

Good interoperability only happens if several (preferably all) MS chooses the same standards and specifications to a certain degree. We foresee that the only way to help make these interoperability guidelines and recommendations effective and practically implemented, is that IDABC (or similar EU organization) takes an extremely active part in promoting them. It is not enough just to collect MS interoperability specifications or just to publish MS case studies etc.

Instead the organization must enthusiastically evangelize about good interoperability patterns and specifications. Thus, inspiration, good examples, best practices and positive arguments to use certain specifications in common, has a real chance to make an impact in the MS. Among other things, this absolutely requires frequent evangelization tours to all MS, central seminars and workshops as well as publications and continually updated web sites.

- In combination with the above, a strongly recommended way to make interoperability happen, is to very actively support or organize prioritization and funding for early interoperability projects to be carried out by one or several MS.

Further elaborated viewpoints and comments from Sambruk about the EIF2

General – Evangelization for interoperability

Practically, and in accordance with the subsidiarity and proportionality principles, the IDABC or other EU organizations cannot dictate interoperable specifications that are crucial to be in accordance between information exchange parties, yielding interoperability. (This also relates to chapter 5.2 of EIF2.)

Thus, there is a strong need for an agile, enthusiastic, evangelist EU organization in this area. Especially, following the more abstract framework definitions, concrete standards and specifications have to get used across MS. The word “campaign” is mentioned in chapter 5.1, but we would like to emphasize that even more is needed, and also that it cannot be a one-time campaign, it must be a continuous effort.

General – Prioritization and funding for early interoperability projects

One of the best ways to get things done is to supply funding. There seem to exist funds in the EU in this sector, e.g. for Large Scale Pilots. Should that not suffice, IDABC should strive to increase these means. IDABC (or similar organization)

should take active and concrete steps to propose specific early projects to MS (or possibly to other organizations).

Since there are so many activities that are recommended in EIF2 (see our next comment below), it is very important to make a priority for early projects. Since focus is crucial, we recommend that five to ten early activities are turned into concrete project plans by IDABC (or similar organization), or that enthusiastic MS are found (one or preferably several) that get good support in creating these project plans. Active support should also take place to organize the funding.

Also, refer to our comment to EIF2 chapter 4.2 below.

General – 95 action points

In the EIF2 document, text that describe actions that should be taken to progress towards the goal of interoperability are printed with a grey background.

There are approximately 95 such imperative "action points" marked with grey in the EIF2. Some of these action points lack a specification of who it is that should carry out the action. In other cases, e.g. it is specified that the MS should carry out the action. But because of the subsidiarity and proportionality principles mentioned above, this would be voluntary actions. In a few cases it might be IDABC itself that could carry out the action. In other action points it is noted that the Commission should do it, but for the document reader it is unclear how easy it would be to get the Commission to actually perform this, or which organ in the Commission it would be.

The problem here is of course that the probability to get this big amount of 95 action points to be executed , may be quite low – if powerful measures are not taken.

- One such measure is the aforementioned energetic and active evangelization, together with supported and funded early projects.
- Another measure is to be more direct in the action point text as to exactly which organ it is that is proposed to carry out an action point, and to advocate decisions by the Comission to accept to carry out the proposed action points.
- The action points should be turned into a proper project plan where the action points' expected finish time is shown, resource needs, who is responsible, priority etc.

Chapter 3.3.1.1 – Agreement

In the definition of interoperability in this chapter it says "mutually beneficial and agreed common goals". We would like to include a comment about two meanings of the word agree. There might be a risk that "agree" here can get used too strictly, yielding too static a view. We have learned at successful enterprises recent years that a learning organization that continually develops its processes often need information that could not beforehand be foreseen. That leads to a more "open" approach, that

information (not considered sensitive or secret) should be available without previous agreement. Otherwise the lead time to organize an agreement and develop IT-solutions to get to the information will be much too long. Also compare with the wave of "Web2.0" solutions that are used also in enterprises, and where the information usage is not agreed upon beforehand. In the SOA[4] community many people suggest that some services are to be published based on an "educated guess" that they will be useful. This does not mean that intense work should not be carried out when it comes to finding out information requirements and common syntax and semantics, but it means that we should not restrict ourselves just to beforehand agreed usages.

This aspect also is consistent with transparency efforts in the EU.

The other meaning of the word "agree" is that once a new usage of an interoperable information exchange has started to execute, an information agreement is needed (in the SOA context usually called a contract, containing descriptions of semantics, syntax, SLA, trust principles etc).

Chapter 3.3.1.1 – Integration and compatibility

The text about what interoperability is not, may be too strong. For example most would argue that "integration" could be designed as either loosely or tightly coupled. Typical "integration" methods and mechanisms are useful (but not enough on their own) when striving towards interoperability.

Also, "compatilibity" is usually seen as a subset of Interoperability, thus compatibility efforts must not be forgotten, when striving towards interoperability.

Chapter 3.3.4 – NIF Observatory

The proposed NIF Observatory is a very good initiative but if not complemented with the above mentioned evangelization, it would be too passive a measure.

Also, the document mentioned cannot be found (http://ec.europa.eu/idabc/en/document/6227).

Chapter 4.2 – Pilot project support

The text in this chapter mentions that EIF2 should support the development and deployment of PEGS at the conceptual level. We would like to add that in parallel with the conceptual work, there is a strong need for IDABC or similar organization to support actual projects and pilots as well, thus to also become concrete and practical.

Also, in the figures, "BO" and "FO" is mentioned but never explained.

[4] SOA – Service Oriented Architecture

Chapter 5.2 – Crosscheck

The action point reads: “Member States and the European Commission should establish and then crosscheck their PEGS development roadmaps in order to ensure interoperability at the EU level and validate that the challenges are addressed effectively.” This is an example of the general problem with the action points in EIF2 (see the general comments above). It is hard to understand how this action would get realized.

Chapter 6.6 – Database

The relational database should not be emphasized since it has more to do with an implementation detail inside a black-box and should not be used as an interoperable interface. The relational model is usually not considered suitable for interoperable transmission of information.

Even though a relational model may be of use to understand information structures, a hierarchical model is often much more suitable for interoperable information exchange – e.g. as used in the mentioned XML standards.

Chapter 7.2.1.1 – Abstraction layer

It is not quite clear what “abstraction layer” means here. We guess that the meaning should be the Aggregate Public Services layer in the GPSCM.

Chapter 7.2.2 – Optimal security

It is very important that for each type of information exchange, the optimal level of security is chosen, not necessary the highest security.

The security requirements are not at all the same in all information exchanges. Obviously, it is not good if too low security is chosen for a certain type of information. But it is also a bad thing if overly secure measures are taken for information that does not require it, specifically since this will add huge costs, implementation complexity, give big implementation risks and increase lead time substantially.

One interesting idea would be to create “security profiles” for a few typical usages that gives a pattern of signing, encryption, logging etc that is relevant to that specific information usage.

As an example of profiles, Denmark has begun work found in http://www.itst.dk/arkitektur-og-standarder/infrastruktur-og-felles-losninger/implementeringsmodeller/implementeringsmodel-for-forretningsservices/Implementeringsmodel_for_forretningsservices_onlineversion.pdf. Sambruk has also defined simple communication profiles in its Open Technincal Platform (ÖTP).

Chapter 7.2.3 – The Aggregate Services Layer

The title should be Aggregate Public Services Layer, in accordance with the contents.

Chapter 7.3.1 – Cross-border Trust

The trust principle is very important. At least some simple, early patterns for trust should be established very soon, otherwise all PEGS work will grind to a halt when people get unsure about how to move on regarding security issues. The trust principle (a.k.a. delegated security) – together with proper agreements/contracts – is in itself relatively straightforward and has been used successfully in enterprises for many years.

The action point included here is very important: “Efforts by the MS guided by the Commission should be undertaken to produce a clear, detailed and systematic definition of the roles, rights and responsibilities of data "owners", data "custodians", and data "users", including the cross-border dimension in these definitions. It also requires technical, organisational, and legal support.”

However, to us it is unclear how this actually will come about. But at least, evangelization from IDABC or similar organization around the trust principle is needed.

Chapter 9 – Open standards, software and methods

A general comment about chapter 9 is that while some of the content is very relevant to interoperability in a direct fashion, some of it is only relevant in quite an indirect way. One viewpoint would be to shorten chapter 9. Nevertheless, below we choose to comment on the different parts of the whole chapter.

Using open standards and specifications

We agree indeed that open standards and specifications should be prioritized for getting good interoperability without lock-in problems. We also agree to the distinction being made in chapter 9.2 between Open Standards and Open Source Software.

Using Open Source Software (e.g. chapter 9.5)

It is not self-evident that Open Source Software (OSS) always should be used. There is a discussion in chapter 9.5 about criteria for choosing OSS and we agree to that. But, the factor that is not mentioned in the EIF2 text is cost. Public sector ICT must be cost-efficient at the same time as avoiding cumbersome lock-in etc. While it is very probable that it is a good thing to use existing OSS implementations of infrastructure components and middleware, if such implementations should be missing or not successful, then commercial/proprietary implementations cannot be ruled out. Besides cost, another factor is implementation risk – will the software be bug-free enough, stabile and get future support?

Creating new OSS public sector projects (e.g. chapter 9.4)

Should there be no good alternatives in the OSS arena, it may be tempting to create a new OSS development project. However, that would give big risks since the success of new OSS development projects in the public sector is not at all guaranteed. There are several success factors that can be studied for existing, well functioning OSS projects like Linux, jBoss and Apache that probably are not valid for public sector OSS projects, such as: The lack of a big community of programmers (working for free or not), the risk that the needed steering oligarchy[5] may not be easy to form, the need for initial funding etc. These aspects, we would like to add to chapter 9.4.

Using OSS business solutions behind interoperable interfaces (e.g. chapter 9.5)

A yet more complex thing than OSS infrastructure solutions is OSS business application software. (Business applications must be the foundation of the Basic Public Functions in the GPSCM – i.e. the underlying application code and databases that in turn get exposed as services through Interoperability Services.) There are very few examples of successful OSS software in this arena, maybe the only one is Compiere (and in turn, Compiere is an ERP and CRM application which inherently has the potential to be much more reusable than for instance, a national fiscal taxation application). Thus, in-house development, or more seldomly, commercial/proprietary software will often have to be chosen.

Sharing ICT solutions and experiences (e.g. chapter 9.6)

The above said, it is at the same time a very good opportunity in that the public sector creates forums for sharing experiences and also actual software, making it available to each other. See chapter 9.6. But that would be a much more “light-weight” cooperation than a real OSS project, avoiding the need for a very strong inner structure that could be difficult to achieve among many public organizations participating in one project.

ICT Reuse (e.g. chapter 9.6)

Something may be mentioned about reuse (mentioned several times in thEIF2 and especially in chapter 9.6). The reuse of ICT resources on the source code level can sometimes work (especially inside one single in-house project or inside one single OSS project), but on a broader scale, that type of reuse has proven very difficult. Instead, a lot of focus (and with a lot of successful cases) is instead put on the reuse of “services” according to the Service Oriented Architecture (SOA). In SOA, the black-box principle is very important, where the myriads of source code details and dependencies are isolated inside the black-box.

The exception to problematic new public OSS projects may be if it from the start can be formed an agile, effective and consistent steering group, and that funding can be supplied etc. In those cases, OSS may be successful in this context. It is also quite

[5] Successful OSS projects so far, have each had a very executive and powerful decision oligarchy, usually a meritocracy. In the case of some OSS projects, like Linux, the oligarchy consisted of one person alone (at least the first years). While it is functioning to create open standards with e.g. requests-for-comments and subsequent voting, that process would be far too slow for all the daily architectural and code quality decisions that take place in an OSS project.

probable that such projects would not rely on a volunteer developer community, but rather that developers are supplied in-house, or procured from a software consultant company. Sambruk has investigated a number of the factors and possibilities around OSS projects, including legal matters and funding models.

I detta ärende har Föreningens verkställande tjänsteman, Claes-Olof Olsson, beslutat. Yttrandet har tagits fram av Sven-Håkan Olsson, DocAccount, Anders Lindgren, Know IT

För föreningen Sambruk

Claes-Olof Olsson
Verkställande tjänsteman