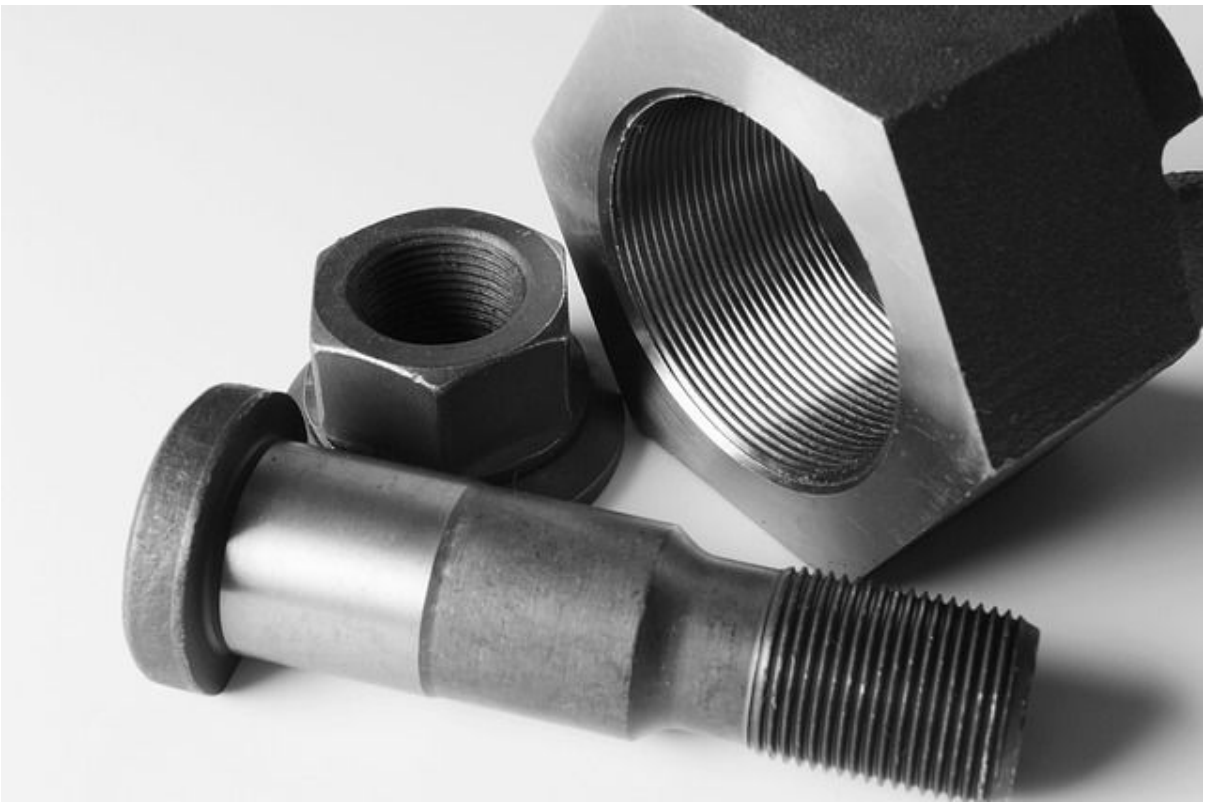
Solving Complex Business IT Problems

The Open Architecture Playbook



Version: August 2016

Preface

Why this playbook

This playbook has only one simple goal: Offer you as much as practical open tools as possible to speed up your process for solving your business IT problem.

Developing business IT systems using architecture methods and tools should give benefits. If done well minimal:

1. Simplifies IT investment decisions
2. Accelerates system implementation
3. Risk reduction
4. Cost reduction

However getting these real benefits from IT architecture can be hard.

Many many books and websites exist where you can find exauthic information on what problem solving is, methods and research papers. When it comes to IT architecture is appears you have entered another domain were decades of process in the field soft and hard problem solving have never entered. This book is not about describing methods or philosophical considerations regarding the field of IT architecture. This books aims to deliver value by offering a collection of practical tools that are directly usable for your business IT problem situation.

We live in a complex world. So spend your time well in the time consuming process of solving business IT problems. Re-use knowledge and tools where possible. And remember a tool is no solution. But without good tools solutions are much harder to accomplish.

Who should use this book?

This playbook is created for everyone who wants to solve business IT problems with less friction. So this book is created for:
Business owners, consultants, IT architects, business architects, information analist, engineers and developers and other people who are heavily involved with the practice of solving business IT problems for real.

What is architecture?

Many definitions exist on what architecture is. This makes is easy just chose one without thinking over the concept of architecture in your specific problem context.

This book is not about definitions and semantics. However solid working definition is:
A good architecture language should be level-of-detail independent.

Architecture is not by definition high-level and sometimes relevant details are of the utmost importance.

Why to use architecture to solve problems

Architecture helps with solving your business IT problems. Architecture for solving IT problems can be regarded as a problem solving method (PSM). The science behind problem solving methods is derived from system science approaches. Solving business IT problems without a clear business context is hard, if not impossible. Solving business IT problems without knowledge of the IT domain is almost impossible. However using generic problem solving methods can speed up the process, since not all challenges within business IT problems demand knowledge of specific IT knowledge fields, like software, infrastructure and internet technologies.

This publication is not an introduction on managing and performing steps within the architecture process. Many general books exist already that describe what IT architecture or enterprise architecture is and how activities can be performed.

But be aware: Many architecture methodologies have no or a very limited scientific foundation. Most methods are commercial or proprietary methods that put limit on usage.

Advantages of using architecture for problem solving

The main advantage for using an architecture approach for solving business IT problems is that the architecture process steps can be applied from different viewpoints. Also all good architecture approaches are system approaches since problems and solutions are related, can have causal dependencies and problems are approached pluriform.

Advantages of using architecture for solving business IT problems:

- Cost reduction and technology standardization
- Process improvement
- Reducing and solving risks
- Reducing complexity

Solve your business IT problem: Creating your architecture

Creating an overall architecture which covers all aspects, elements, relationships and technology building blocks involved is hard. Therefore it is common practice in problem solving to incorporate only factors that are relevant in your model.

There is no best way. Period.

Open without limitations

This playbook is build from an open approach. This means that only references are made to tools,models and reference models that are free to use without limitations.
This means that all references to tools made in this publications are:

- Open Source and
- Common Creative approved license

Many tools you can use for solving business IT problems are very expensive to acquire, require additional training and certification and put limitations on usage.

E.g. you need a commercial license for commercial exploitation of ArchiMate(™). Which makes adaption low and puts extra barriers on improvements of this architecture modelling language.

Of course there is nothing wrong with commercial architecture tools and commercial architecture problem solving methods. Let that be clear. However if you:

- Do not like limitations your tool in any way;
- Do not want to pay large amounts of license fees for IT problem solving tools;
- Want full control on the value created using these tools (e.g. share created documents under your constraints);
- Think reuse of work of others will benefit you;
- Like to contribute and improve tools based on ideas and work on others, so everyone can benefit;

Then you should consider using open tools and use material that is published under an free Creative Commons License.

Stay in touch

Let's keep the conversation on collecting real open tools that help in developing IT architectures and designs going! I am on Twitter: @maikelmardjan

To comment or ask technical questions about this book, send email to info[at]nocomplexity.com

For more information about our books, courses, services, news, and solutions for you see our website at https://nocomplexity.com

Share this book!

The best way to help is share this eBook!

Please share the link where this eBook is published (https://nocomplexity.com) or the ePub/PDF version with all your collegues.

See the section CONTRIBUTION or my github site if you want to make this book better and better.

The website for this book is https://nocomplexity.com The Creative Commons license that this book is released under gives you the right to copy, duplicate and improve this book as much as you want! So go for it.

Table of Contents

Introduction

Smart people have been thinking on how to create IT architectures as long as there has been computers. Ideas come and go, however creating a good architectures can still be complex and time consuming. Especially when you try to invent the wheel for yourself. With this interactive playbook you can create your IT architecture better and faster. The focus of this architecture playbook is in on:

1. Knowledge reuse. Why reinvent the wheel again? It is far and more fun to create a better wheel for your organisation or IT project instead! Focus on the hard complex context specific issues. Use good open tools and knowledge for the easy 80%!
2. Easier creation of architecture documents and deliverables. This playbook has an extensive list of *all(*)* open tools available for creating your IT architecture or design. Using these open tools will speed up the process of creating your architecture deliverables and reduce your risks.
3. Quality improvement. By making use of content parts provided for various architecture deliverables you will lower your business risks. Complex business IT projects will fail. Now and in future. But if you make use of proven methods and tools developed from decades of IT architecture scientific research you will lower the risk for your project. Architecture will help to make your projects more successful in terms of costs, speed and profitability.

(*) If you miss a good open tool that is also usable for IT architecture creation, do not hesitate to contribute to this open publication!

This architecture playbook is divided in the commonly used architecture sections:

- Business
- Data
- Applications and of course
- Technology Infrastructure (TI)

This playbook is primarily created for on-line usage. But also ePUB and PDF versions are available.

Why this architecture playbook?

Creating a business IT architecture has many benefits. However creating a complete architecture is known to be:

- Difficult
- Time consuming

Of course many tools have been developed the last 30 years to make creating architecture (documents) easier. However many tools force you into a very tight harness without the needed flexibility. Complex problems need flexible solutions and tools should not slow you down in creating sketches, documents and typical architecture deliverables. A one size fits tool for solving complex business IT problems does not exist. And since creating good architectures is knowledge intensive work that delivers real value for business you should be aware of vendor lock-ins and using methods and tools that do not share knowledge sharing.

Open Source tools and open access documentation offer a default head start for reuse and improvement of knowledge work in IT architecture.
You can buy and read many many books on how to do architecture well. All books claim a magic method and promise ultimate success when followed. We love books and reading methods. E.g. TOGAF. However this playbook is different. This architecture playbook is not about describing how you should create your architecture. You are free to use every method you want. This architecture playbook is all about giving you direct usable content and tools that you can use and reuse for your architecture challenge.
So this playbook brings you ultimate freedom and flexibility. You can export the results from certain tools in any desired format. In this way you can still use your architecture enterprise tool(s) that you already have invested deeply in. So use the outcome of tools within this playbook in combination with the enterprise architecture tool that are mandatory within your organisation.

What is in this architecture playbook?

Many good books and courses already exist that cover the why and how of (Enterprise) IT architecture. So this book will not explain aspects and concepts of the very broad field of IT architecture. This playbook is targeted on providing practical tools for creating your IT architecture or design faster and better. So the focus is 100% on the real thing by using the following leading principles:

1. Tools that speed up creating your business IT Architecture.
2. Templates ready to use for creating business IT Architecture documents.
3. Collections of principles, requirements, standards and reference architecture that speed up the creation of your architecture deliverable.
4. Collection of various Architecture OSS Tools that speed up creating your business IT architecture.
5. Collection of architecture QA tools to control and manage your IT projects.
6. Collection of reusable reference architectures, foundation architecture, architecture journals and more. Knowing where you can find what in an easy way will reduce your time. So you will have more time to spend on solving your problem in your organization.

Caution: This architecture playbook contains OPEN material only!

This architecture playbook is created to be open. This means that:

1. This architecture playbook is licensed using a liberal license (CC-BY-SA). This means that you can reuse remix, transform, and build upon the material in this architecture playbook for any purpose.
2. All tools mentioned within this architecture playbook are licensed under a OSS license and all material referenced too is licensed under an CC license whenever possible.

This gives you the opportunity to reuse and mix content from this architecture playbook the way you want. This also means that all tools mentioned to in this architecture playbook are open and free to use without costs. We believe that sharing tools and knowledge, especially knowledge for making better architectures SHOULD be free. If you want to host this Architecture Playbook on your own company site, Intranet or just want to have more information on the OSS tools: Contact Us, or visit the github repository.

Using the templates

The various templates can be downloaded in html or depending on your browser preferences directly opened in another browser tab. The templates can be easily used within various office suites (e.g. MSWord, OpenOffice, GoogleDocs). When you press *[CTRL+A]* and *[CTRL+C]* and paste the complete content into your favourite text editor with *[CTRL+V]* you can directly work on the various sections as outlined in the template. In case you need the template in another format of have other questions regarding using the templates: Just contact us!

Notational Conventions

This architecture playbook uses the key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL”. This key words are to be interpreted as described in RFC 2119. Using these key words gives clarity and avoids confusion.

Business Architecture

A Business Architecture defines the structure of your organization in terms of its governance structure, business processes, business services and products, business information and stakeholders. A business architecture outlines the relation to strategic goals towards a working system. So in short a business architecture can be regarded as a blueprint of the working of an organization. By using a business architecture your organisation can handle changes and problems with less cost and in less time since you know all the important dependencies, relation and information flows. Every successful organisation, small or large, SHOULD have a business architecture. A business architecture does not have to be fully descriptive, complete or created among a certain standard. Designing organisations and describing the working of an organisation is know to be complex and hard. However simple tools exist to create a meaningful business architecture for your organisation.
This section is created to speed up the process for creating a real business architecture.
You can use one of the following tools:

- Business architecture template(s)
- (Re)use business principles
- Modelling your business processes
- Defining your business products

Tools for creating a business architecture

To speed up the process of creating your business architecture you can make use of one of the following tools:

- (Generic) Business principles
- Archi. Archi™ GUI tool for creating a business architecture using the ArchiMate modelling™ language. The Archi tool is targeted toward all levels of Architects. The tool is MIT Licensed, so it provides a low cost solution to users who are looking for a free, open ArchiMate modelling tool.
- Camunda Modeler. Camunda Modeler is an OSS desktop application for editing BPMN process diagrams(2.0) and DMN decision tables. It is very easy to use, which means that business analysts can use it as well as developers, working on the same diagrams. Camunda Modeler is part of an open source platform for workflow and business process management. So when you use the Camunda suite you can also use the execution engine for your processes you have modelled.
- Protégé. Protégé is an OSS web or desktop application that can be used for building business ontologies.

Business architecture templates

Creating a business architecture means doing real business research. However for a quality business architecture it make sense to make use of a draft template. From a template you can easily add, remove or change subjects that need special attention within your context. Also since architecture documents always should be created for a clear business goal and to

be used by different stakeholders, all quality documents SHOULD contain some default reusable text blocks.

- Basic business architecture template
- Stakeholder template
- Business Architecture. EU template/document that describes the product, service strategy, organizational, functional, process, information, and geographic aspects of the business architecture.
- Business Architecture Document Template. Typical Togaf(tm) like business architecture template.

Using business viewpoints

Viewpoints can provide benefit to address specific concerns for certain stakeholders. Below a list of most used viewpoint within a business architecture:

- Motivation view: describes what the business achieves for itself and its stakeholders (direct and indirect value).
- Capability view: describes how the business delivers direct and indirect value in response to the challenges of the environment.
- Activity view: describes the day-to-day behaviour of the business.
- Responsibility view: captures the relationships between individuals and organizations in terms of responsibilities and commitments. These relationships and organizational interfaces may be represented as business services.

Help for creating a business architecture

- Help Guide for creating a business architecture when dealing with SOA/Integration. (CC License)

Business Principles

Having solid business principles is key for a successful company. Small or large. Having good and solid business principles is key for developing a good architecture within solid time and cost constraints. Business principles SHOULD be defined and agreed upon within a solid process with all key business stakeholders involved. The list with principles below can give you a head start, since these principles are collected from various successful businesses.

Be Collaborative	
Statement	Be Collaborative

Rationale	The saying: “If you want to go fast, go alone. If you want to go far, go together.” is attributed to an African proverb, but could easily be a mantra for technology-enabled development projects. The principle: Be Collaborative suggests strategies for leveraging and contributing to a broader commons of resource, action, and knowledge to extend the impact of development interventions.
Implications	● Engage diverse expertise across disciplines and industries at all stages. ● Work across sector silos to create coordinated and more holistic approaches. ● Document work, results, processes and best practices and share them widely. ● Publish materials under a Creative Commons license by default, with strong rationale if another licensing approach is taken.
Tag(s)	business, design

Business continuity	
Statement	Business continuity of Corporate activities must be maintained, despite IT interruptions.
Rationale	Hardware failures, natural disasters, and lack of data integrity must not interrupt business activities.
Implications	- Recoverability, redundancy, and maintenance must be approached at inception. - Applications must be assessed regarding criticality and impact on the company's mission to determine which continuity level is required and which corresponding recovery plan must be implemented. - A business continuity plan must be present/developed.
Tag(s)	availability, business, continuity

Business Principle	
Statement	These architectural principles will apply to all organisational units within the enterprise.
Rationale	The only way the University will be able to provide a consistent and measurable level of appropriately robust, reliable, sustainable services and quality information to decision-makers, is if all stakeholders abide by the University’s overarching principles for its technology, information and business architectures.
Implications	This fundamental principle will ensure inclusion, consistency, fairness

	and continual alignment to the business. Without this the management of our technologies, information and business processes would be quickly undermined. Business Partners engaging with the business will work to find accommodation between interested parties around any conflicts with a principle relevant to the proposal.
Tag(s)	business

Cohesiveness	
Statement	Our organization and business unites (and departments) shall present a consistent and unified face through a common and integrated approach to service delivery.
Rationale	The intent of this principle is to ensure that services are presented to consumers in a consistent and cohesive manner. Similarly, consumers will be presented with a common look, feel and experience regardless of what business unit service or channel they are accessing at the time. Adoption and continued use of a service by consumers will depend, to an extent, on ease of use enhanced by consistency of service delivery.
Implications	Departments and Business unites will need to collaborate in the development of services. This will require open sharing of information, cross-agency planning, and understanding of whole-of-government service channels, segments and lines of business.
Tag(s)	business

Compliance with Statutory Obligations	
Statement	Enterprise data and information management processes must comply with all relevant internal and external laws, policies, and regulations.
Rationale	Enterprise policy is to abide by laws, policies, and regulations. This will not preclude business process improvements that lead to changes in policies and regulations.
Implications	The enterprise must be mindful to comply with all laws, regulations, and external policies regarding the collection, retention, and management of data. Continual education, access and awareness to the rules must be maintained. Efficiency, need, and common sense are not the only drivers. Changes in the law and changes in regulations may drive changes in our processes or applications.
Tag(s)	business

Create a MVP fast	
Statement	If your MVP takes a year to build…it's not an MVP.
Rationale	Creating a MVP should take not more than 1 month.
Implications	A created MVP is not ready for sale, but ready to learn from for later stages.
Tag(s)	business

Ease-of-Use	
Statement	Applications are easy to use for end-users and administrators.
Rationale	The more a user has to understand the underlying technology, the less productive that user is. Avoid mistakes due to difficult comprehension interaction with a system. Most of the knowledge required to operate one system will be similar to others. Using an application should be as intuitive as driving a different car.
Implications	• The underlying technology is transparent to users. • Training is kept to a minimum, and the risk of using a system improperly is low. • Default (de-facto) GUI's are used for interacting with the system. • No large user manual is needed.
Tag(s)	business

First, make it easy. Then make it fast.	
Statement	First, make it easy. Then make it fast. Then make it pretty.
Rationale	When launching a new product cost are high. Making a good performing product is complex and expensive. So when working on a MVP, try to make a product easy to use and easy to create. Focus on UX.
Implications	Difficult back-end performance and scaling issues are handled later. This can increase development cost if performance is never account for in a design.
Tag(s)	business

Give before receiving

Statement	Give before receiving
Rationale	Giving is the only way to establish a real relationship and a lasting connection. Focus solely on what you can get out of the connection and you will never make meaningful, mutually beneficial connections and a sustainable business
Implications	Invest time and money in all stakeholder relations.
Tag(s)	business

Information Management is Everybody's Business	
Statement	All organisations in the enterprise participate in information management decisions needed to accomplish business objectives.
Rationale	Information users are the key stakeholders, or customers, in the application of technology to address a business need. In order to ensure information management is aligned with the business, all organisations in the enterprise must be involved in all aspects of the information environment. The business experts from across the enterprise and the technical staff responsible for developing and sustaining the information environment need to come together as a team to jointly define the goals and objectives of IT.
Implications	To operate as a team, every stakeholder, or customer, will need to accept responsibility for developing the information environment. Commitment of resources will be required to implement this principle.
Tag(s)	business

IT Responsibility	
Statement	The IT organisation is responsible and accountable for owning and implementing all IT processes and infrastructure that enable solutions to meet business-defined requirements for functionality, service levels, cost, and delivery timing. Decisions should always align back to the requirement of the Business.
Rationale	Effectively align expectations with business requirements and our overall capabilities so that all projects are cost-effective and can be completed in a timely manner. Efficient and effective solutions should have reasonable costs and clear benefits relative to the business proposition.
Implications	The IT function must define processes to manage business expectations and priorities. Projects must follow an established

	process to reduce costs and to ensure the project has a timely completion. Data, information, and technology should be integrated to provide quality solutions and to maximise results.
Tag(s)	business

Make things open: it makes things better	
Statement	Make things open: it makes things better
Rationale	We should share what we're doing whenever we can. With colleagues, with users, with the world. Share code, share designs, share ideas, share intentions, share failures. The more eyes there are on a service the better it gets — howlers are spotted, better alternatives are pointed out, the bar is raised. Much of what we're doing is only possible because of open source code and the generosity of the web design community. We should pay that back.
Implications	
Tag(s)	business

Maximise Benefit to the Enterprise	
Statement	Information management decisions are made to provide maximum benefit to the enterprise as a whole.
Rationale	This principle embodies ‘‘service above self ’’. Decisions made from an enterprise-wide perspective have greater long-term value than decisions made from any particular organisational perspective. Maximum return on investment requires information management decisions to adhere to enterprise-wide drivers and priorities. No Organisation Unit will detract from the benefit of the whole. However, this principle will not preclude any Organisation Unit from getting its job done.
Implications	Achieving maximum enterprise-wide benefit will require changes in the way we plan and manage information. Technology alone will not bring about this change. Some organisations may have to concede their own preferences for the greater benefit of the entire enterprise. Application development priorities must be established by the entire enterprise for the entire enterprise. Applications components should be shared across organisational boundaries. Information management initiatives should be conducted in accordance with the enterprise plan. Individual organisations should pursue information management initiatives which conform to the blueprints and priorities established by

	the enterprise. We will change the plan as we need to.
Tag(s)	business

Maximize Benefit to the complete enterprise	
Statement	Information management decisions are made to provide maximum benefit to the enterprise as a whole.
Rationale	Decisions made from an enterprise-wide perspective have greater long-term value than decisions made from any particular organizational perspective. Maximum return on investment requires information management decisions to adhere to enterprise-wide drivers and priorities. No minority group will detract from the benefit of the whole. However, this principle will not preclude any minority group from getting its job done.
Implications	Application development priorities must be established by the entire enterprise for the entire enterprise. Strong enterprise governance is required. Information services should be shared across organizational boundaries. Information management initiatives should be conducted in accordance with the enterprise plan. Individual organizations should pursue information management initiatives which conform to the blueprints and priorities established by the enterprise. We will change the plan as we need to.
Tag(s)	business

Reliability	
Statement	Information and information systems are reliable, accurate, relevant and timely
Rationale	The take-up and use of lower cost channels will depend on users of services trusting the ability of the organization to provide reliable, accurate, relevant and timely information to consumers.
Implications	Good processes create good data. Processes will need to be the focus of ongoing continuous improvement (which in turn will improve reliability, accuracy, relevancy and timeliness). Our organization needs to deliver information which customers can rely upon.
Tag(s)	business

Reuse and Improve

Statement	Reuse and Improve
Rationale	As the use of information and communications technologies in international development has matured, so too has a base of methods, standards, software, platforms, and other technology tools. Yet too often we see scarce resources being invested to develop new tools when instead existing tools could be adapted and improved. This principle: Reuse and Improve highlights ways that adaptation and improvement can lead to higher quality resources available to the wider community of international development practitioners.
Implications	• Use, modify and extend existing tools, platforms, and frameworks when possible. • Develop in modular ways favoring approaches that are interoperable over those that are monolithic by design.
Tag(s)	business, design

Reuse before Buy, Buy before Build

Statement	Prior to acquiring new assets, the company will reuse applicable existing information and technology assets. If no existing internal asset is available for reuse, the company prefers to acquire, by purchasing or licensing, applicable externally available assets. The comapny least preferred option is to custom build a new asset.
Rationale	• Reusing IT assets (for example, IT systems or data) that are already available is often the simplest, quickest, and least expensive solution, assuming that the IT assets in question sufficiently fit the intended purpose. • It is less expensive to buy standard IT solutions than to custom build them, as long as they are not adapted and maintenance is left to the product supplier. • Many authoritative data sources make their data products available (or offer data acquisition / generation services), reducing the company's need to generate such data itself. • Custom development of IT assets is often very expensive to sustain.
Implications	• When functionality is required, existing IT assets in the organization must be evaluated and used first, unless they do not exist and/or are a significant mismatch to the required functionality. • To ensure that IT assets are being reused as much as possible, business areas must be prepared to adapt to existing solutions that provide adequate functionality, particularly in

	situations where the accountable governance body does not deem that business area's practices to be required to be different from industry standard practices. ● The company will prefer COTS products and particularly those that are configurable. Some products are so configurable that there is little difference between extensive configuration and custom development. The company y must clearly understand when configuration equates to custom development (that is, the level of configuration is so high that the COTS solution is essentially the same as custom development). In these cases, the scenario will change from buy to build. ● Agreements or licenses to use data may have legal implications and legal consultation should be part of the process of deciding to use a new data source.
Tag(s)	business

Routine Tasks are Automated Where Appropriate	
Statement	Routine tasks that can be automated are automated, where the benefit justifies the cost.
Rationale	● Routine tasks require relatively little specific knowledge and can be automated fairly easily. ● Automated tasks are more cost efficient and timeefficient, and less errorprone, than manual tasks. ● Employee capacity requirements can be optimized, freeing them up to focus on more complex activities.
Implications	● The knowledge required to perform certain tasks is analyzed and embedded in an IT system when it can be easily formalized. ● Nonroutine tasks may not be automated. ● Individual performers will need to be able to automate their own tasks. Business areas should integrate automated work flows, where one business unit receives another business unit's automated output as its input.
Tag(s)	business, functionality

Solution space	
Statement	Never try to use technical solutions to a non-technical problem.
Rationale	Of course some IT and technology can help to solve problems, but technology and IT is never enough. A non technical activity, process,

	behaviour change, etc is always needed.
Implications	Do not focus on technology and IT solution only when solving problems.
Tag(s)	business, design

Start simple	
Statement	Start simple.
Rationale	Start simple. Build a high quality solution to a real problem for a cohesive group of people. If you solve one problem really well, then you can move on to the next problem (one simple approach at a time) instead of trying to tackle several things at once and, as a result, not really solving anything. Product development is about earning the right to build the next thing.
Implications	Complexity will come later when having invest lots of time and money and simple adjustments have more impact. Fixes will be more complex when products are mature anyway due to backward compatibility requirements.
Tag(s)	business, design

Strategic focus	
Statement	Investment decisions are driven by business requirements
Rationale	Core government needs and priorities should be the primary drivers for investment. Investment decisions should be defined by the agency's vision and strategic plans as well as the requirements of the business. These should also take into account whole-of-government strategic guidance. A business-led and business outcome-oriented architecture is more successful in meeting strategic goals, responding to changing needs and serving consumer expectations. Government service requirements will define any required technological support.
Implications	Agencies need to align with whole-of-government strategic direction. • Agency's strategic plans need to align with whole-of-government strategic direction. • Investment decisions should be made in accordance with the agency's vision and strategic plan. • Changes to processes, applications and technology should be made in response to an approved business initiative. • Design of business solutions will need to be aligned with, and traceable to strategic goals and outcomes. • Services, processes and applications will need to be designed from the perspective of the service user. • Building or

	redevelopment of applications and solutions will be undertaken only after business processes have been analysed, simplified or otherwise redesigned as appropriate. • Applications are delivered in a collaborative partnership with the business owners to enable solutions to meet user-defined requirements for functionality, service levels, cost and delivery timing.
Tag(s)	business

Use Open Data, Open Standards, Open Source, and Open Innovation	
Statement	Use Open Data, Open Standards, Open Source, and Open Innovation
Rationale	Too often in international development, scarce, public resources are spent investing in code, tools, and innovations that are either locked away behind proprietary, fee-based firewalls, or created in a bespoke way for use in sector-specific silos. This principle: Use Open Data, Open Standards, Open Source, and Open Innovation provides a framework to consider an “open” approach to technology-enabled international development.
Implications	● Adopt and expand existing open standards. ● Open data and functionalities and expose them in documented APIs (Application Programming Interfaces) where use by a larger community is possible. ● Invest in software as a public good. ● Develop software to be open source by default with the code made available in public repositories and supported through developer communities.
Tag(s)	business, design, Integration, Software development

Data Architecture

Every business, small or large SHOULD have a data architecture. In the core a data architecture gives the overview and insights into the only one real value of your IT: Information. A data architecture gives overviews, visuals and describes e.g.:

- What data is used where and how.
- Who owns what data.
- How is information created from data sources.
- Internal and external data sources used.
- Privacy & Security aspects of data (so be sure to have an data owner)
- What structures, objects and relations are the core of your information model?

Tools for creating a data architecture

To speed up the process for creating a data architecture you SHOULD use tools. Below is a selection of tools that will speed up the process of creating your data architecture. Be aware that data modelling and database design are two very different activities. The emphase for a data architecture SHOULD be on the conceptual and logical data models. In general physical data models are related to the sql or nosql storage engine used, in combination with scalability(performance) and security requirements.

- Data Principles. Using data principles saves you time and cost. Especially in the long term. Selecting data principles you need in your project of a solid collection gives you a head start.
- Protégé. Protégé is a free, open source ontology editor and knowledge-base framework. This open-source ontology editor and framework can be used for for building intelligent systems. The tool is developed by the Stanford Center for Biomedical Informatics Research and has a large and active community of users and developers.
- Archi. Archi™ GUI tool for creating an architecture, using the ArchiMate modelling™ language. Since Archi is targeted to all architecture aspects, this tool is usuable for creating conceptual, logical and physical data models too.
- WWW SQL Designer. This tool allows you to draw and create database schemas (E-R diagrams) directly in browser. A physical data model (sql) can also be imported and adjusted visually.
- MySQL Workbench. MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

Data Architecture Templates

To speed up the process of create your data architecture you SHOULD make use of standardized open templates. The following templates are provided:

- Data Architecture View(s)
- Data Model Architecture
- Data Architecture template
- Data Store template

Data Principles

Principles are used within successful business IT projects. A principle is a qualitative statement of intent that should be met by the architecture.
Do not hesitate to add your data principles to this list so everyone can benefit. Click here to add your contribution.

Accessible	
Statement	Data is available to the widest range of users for the widest range of purposes.
Rationale	
Implications	
Tag(s)	data, Integration

Machine processable	
Statement	Data is reasonably structured to allow automated processing.
Rationale	
Implications	
Tag(s)	data, design, Integration

Primary data	
Statement	Data is as collected at the source, with the highest possible level of granularity, not in aggregate or modified forms.
Rationale	
Implications	
Tag(s)	data, Integration

Timely

Statement	Data is made available as quickly as necessary to preserve the value of the data.
Rationale	
Implications	
Tag(s)	data, Integration

Application Architecture

The application architecture describes the application components of a system. Part of the application architecture is providing high level insights in the software building blocks, services and micro services that are part of the application landscape.

Tools for creating an application architecture

Many OSS tools for creating an application are targeted on creating software code. Although an IT architect should be able to code, most of the time architects are more concerned with designing building blocks, interfaces and implementation constrains.
Using the following tools for creating an application architecture saves time and increases the quality of your application architecture:

- UMLet. UMLet is an open-source UML tool with a simple user interface: draw UML diagrams fast, export diagrams to eps, pdf, jpg, svg, and clipboard, share diagrams using Eclipse, and create new, custom UML elements.
-
- Papyrus Modeling environment. Papyrus is an industrial-grade open source Model-Based Engineering tool. Papyrus is the base platform for several industrial modelling tools. Papyrus can be used as graphical modelling tool, but aims to support MDA (Model Driven Architecture) completely.
- Archi. Since Archi(tm) is a real TOGAF based architecture/desing tool, Archi is a good solution for creation of an application architecture.
- Open ModelSphere. Open ModelSphere is a powerful data, process and UML modeling tool.
- Modelio. Modelio is a modelling environment, supporting a wide range of models and diagrams, and providing model assistance and consistency checking features. Modelio can also generate Java code.

Creating an application architecture will lead to creating internal and external interfaces. APIs form the connecting glue between modern applications. Creating good interfaces is a MUST for every good architecture. Below some open tools that can help to speed up this step:

- Integration Principles. With this tool you can easily (re)use good integration principles for your project.
- API Blueprint. API Blueprint is all about the design-first philosophy. API Blueprint itself is OSS and has a growing base of OSS tools based on the spec. API Blueprint supports the complete chain for interface development (design, test, create etc).
- Swagger based tools. Swagger is a simple yet powerful representation of your RESTful API. Swagger has a large ecosystem of OSS tools that assist in creating, testing and documenting APIs.
- RAML based tools. RESTful API Modeling Language (RAML) makes it easy to manage the whole API lifecycle from design to sharing. The RAML specification is designed for design APIs better and faster.

Application Architecture Templates

To speed up the process of creating an application architecture you SHOULD make use of one of the templates below.

- Architecture description template for use with ISO/IEC/IEEE 42010:2011. (MSWord template or PDF)
- SAD (Software Architecture Document)
- Software Engineering Institute's(SEI) Architecture template (MSWord)
- Solutions Architecture Template (U.S. Government, HUD) (MSWord)

Integration Principles

Integration principles are used within successful business IT projects. (Re)use the integration principles from the collection below.
Do not hesitate to add your data principles to this list so everyone can benefit. Click here to add your contribution.

Accessible	
Statement	Data is available to the widest range of users for the widest range of purposes.
Rationale	
Implications	
Tag(s)	data, Integration

Documentation	
Statement	Documentation
Rationale	It is well documented that documentation is an important characteristic for making software components reusable. Documentation for software is essential for any future use or modification and critical for maintainability. Programmers are unlikely to reuse software that is not well-documented or commented since it makes it harder to understand and maintain. Documentation should be self-contained, adaptable and extensible. Specific documentation for reuse of the component enhances the chances for usage of the component in future.
Implications	
Tag(s)	design, Integration, Software, Software development

Interoperability	
Statement	Software and hardware should conform to defined standards that promote interoperability for data, applications, and technology.
Rationale	(Open) Standards help ensure consistency, thus improving the ability to manage systems and improve user satisfaction, and protect existing IT investments, thus maximizing return on investment and reducing costs. (Open) Standards for interoperability additionally help ensure support from multiple vendors for their products, and facilitate supply chain integration.
Implications	(Open) Interoperability standards and industry standards will be followed unless there is a compelling business reason to implement a non-standard solution. A process for setting standards, reviewing and revising them periodically, and granting exceptions must be established. The existing used standards used within IT platforms and applications must be identified and documented.
Tag(s)	Integration

Machine processable	
Statement	Data is reasonably structured to allow automated processing.
Rationale	
Implications	
Tag(s)	data, design, Integration

Prefer Real-time Data Exchange	
Statement	Data exchange speed and latency must be based on business need, with a preference for real-time exchange to improve the delivery of services.
Rationale	• Decisions made based on old data have lower accuracy and may lead to errors and/or inconsistencies. • Users/Services expect the most recent data to be available in their work processes.
Implications	• All changes to data are processed immediately and are distributed to all other IT systems that use the data. • Where data capture systems and the authoritative source for data are different, data must be shared with the authoritative

	source at the speed required by the most timesensitive usage of the data. This includes all data that is encompassed in a given transaction. • Data exchange speeds will often be driven by the needs of the most time sensitive usage of the data.
Tag(s)	Integration

Primary data	
Statement	Data is as collected at the source, with the highest possible level of granularity, not in aggregate or modified forms.
Rationale	
Implications	
Tag(s)	data, Integration

Restrictiveness	
Statement	Restrictiveness
Rationale	Restrictiveness is one of the important general properties of good reusable component designs. “State everything about the behaviour that is expected of a correct implementation—and nothing more (“restrictiveness”).” For example, consider a component that has a functionality of performing certain operations on only the string data type. The component could be restricted to accept only the string data type, not other types such as integers or floating point numbers. This is contrary to genericity but the component requires only one data type and a trade-off is made with genericity (type independence).
Implications	
Tag(s)	design, Integration

Timely	
Statement	Data is made available as quickly as necessary to preserve the value of the data.
Rationale	
Implications	
Tag(s)	data, Integration

Transparent information flow and usage	
Statement	Use and share information in a way that is transparent and benefits the user.
Rationale	User generated data belongs to the user.
Implications	No hidden analytics performed on user data without explicit approval from users. Open transpartent design (OSS)
Tag(s)	Integration

Use Open Data, Open Standards, Open Source, and Open Innovation	
Statement	Use Open Data, Open Standards, Open Source, and Open Innovation
Rationale	Too often in international development, scarce, public resources are spent investing in code, tools, and innovations that are either locked away behind proprietary, fee-based firewalls, or created in a bespoke way for use in sector-specific silos. This principle: Use Open Data, Open Standards, Open Source, and Open Innovation provides a framework to consider an “open” approach to technology-enabled international development.
Implications	● Adopt and expand existing open standards. ● Open data and functionalities and expose them in documented APIs (Application Programming Interfaces) where use by a larger community is possible. ● Invest in software as a public good. ● Develop software to be open source by default with the code made available in public repositories and supported through developer communities.
Tag(s)	business, design, Integration, Software development

Well-defined Interface	
Statement	Well-defined Interface
Rationale	A well-defined interface aids the reusability of software components. An interface determines how a component can be reused and interconnected with other components. If the component’s interface is simpler, it should be easier to reuse. There are three types of interfaces: application programming interface (API), user interface and data interface. APIs may be the most important type of interface for reuse. In reuse, a well-defined API can be used to integrate the

	application’s functionality into the new software system. APIs may be language dependent or independent.
Implications	
Tag(s)	design, Integration

Technology Infrastructure (TI) Architecture

A Technology Infrastructure (TI) Architecture is all about the hard IT. Hosting, capacity, connectivity, monitoring and operations. Having a flexible TI architecture serves all business processes.
Tools for creating a Technology Infrastructure Architecture:

- IP Address range calculator
- Simple Network Bandwidth Calculator
- Simple Capacity Calculator for web servers

For systems that are not using PaaS Cloud Services a lot of attention, time and effort is needed for the TI part of the system. Below some TI templates that will make with creating the technical solution.
TI Templates:

- Technical Architecture (MSWORD). Template used at US Florida Department of Transportation.
- High-Level Technical Design (MSWord). HLD Template used at US Centers for Medicare & Medicaid Services.
- System Design Document (MSWord). Source US CMS.
- Service Design & Transition (MSWord). Source EU FitSM program.
- IT Service Capacity Plan (MSWord). Source EU FitSM program.

Quality Management & Risk Reduction

Quality is and will always the number one aspects. This section provides various tools and templates for managing your business IT quality aspects.

Tools for Architecture QA processes

- Architecture checklists.
- Architecture Documentation Checklist
- Make re(Use) of defacto standards for requirements
- Non Functional Requirements catalogue (NFR list).
- Short overview of ISO 25010

Templates for better IT Architecture QA

Reuse of templates means you have more time to spend on the real quality aspects. So use the templates below.

- Risk Assessment Toolkit (source State of California)
- Risk assessment Toolkit (source State of Oregon)
- Risk Management Toolkit (mitre)
- Constraints template

Architecture checklists

An architecture checklist helps in the governance process. Architecture checklist can become long, complex and time consuming. However our aim with this architecture checklist is that this checklist helps you and all stakeholders involved in improving the various architecture activities needed to reach the defined goals.
This architecture checklist is composed out of critical questions that all relate back to the main goal of doing architecture in the first place.
Checklist questions:

- Does the architecture address operability?
- Does the architecture address the following quality attributes:
    - performance
    - availability
    - maintainability
    - modifiability
    - security
    - privacy
    - testability
    - operability

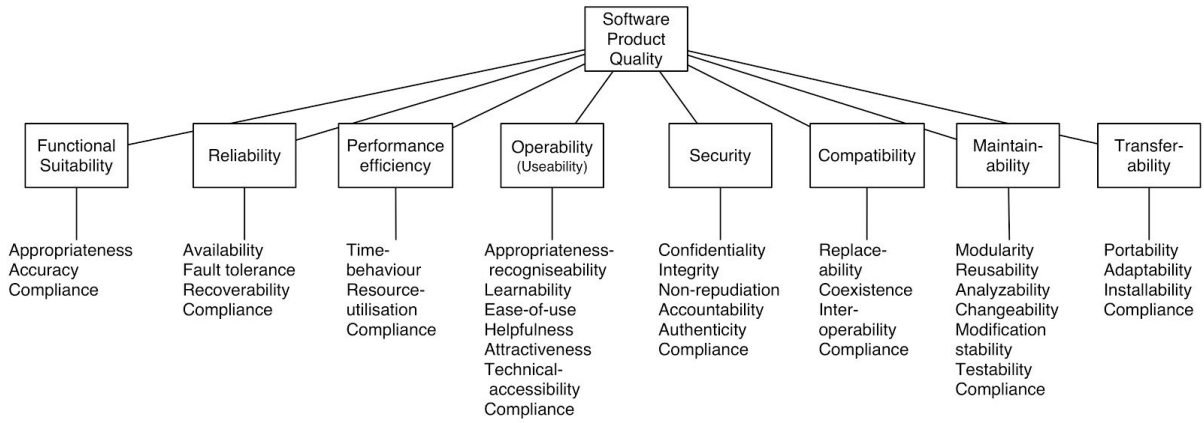
    - flexibility
- Does the architecture address principles of design?
- Can risks easily be derived from the architecture deliverables?
- Are architecture reviews done in a structured way? Architecture reviews SHOULD be performed to increase quality, control costs and reduce risks.
- Is it clear what assumptions are used for creating the architecture? (Take explicit and implicit assumptions into account!)

Architecture Documentation Checklist

When creating business IT documentation following these rules will help for producing sound documentation:

1. Documentation should be written from the point of view of the reader, not the writer.
2. IT Documentation should be organized for ease of reference, not ease of reading. A document may be read from cover to cover at most once, and probably never. But a document is likely to be referenced hundreds or thousands of times.
3. Avoid repetition. Each kind of information should be recorded in exactly one place. This makes documentation
4. easier to use and much easier to change as it evolves. It also avoids confusion.
5. Avoid unintentional ambiguity. In some sense, the point of architecture is to be ambiguous. However unplanned
6. ambiguity is when documentation can be interpreted in more than one way, and at least one of those ways is incorrect. A well-defined notation with precise semantics goes a long way toward eliminating whole classes of linguistic ambiguity from a document. This is one area where architecture description languages help a great deal, but using a formal language isn't
7. always necessary.
8. Standardize your documentation. Use always the same templates for the same documents within your organization. Even better: Make use of de-facto standardizations that also make sense for people outside your organization, since you will be working with many people outside your organization during the life cycle of your product. A standard organization offers many benefits. It helps the reader navigate the document and find specific information quickly.
9. Record rationale. So document decisions made. Recording rationale will save you enormous time in the long run, although it requires discipline to record in the heat of the moment. It will prevent the same discussions over and over again and everyone knows why the chosen path is taken.
10. Keep it current. Documentation that is out of date, does not reflect truth, and does not obey its own rules for form and internal consistency will not be used. Documentation that is kept current and accurate will be used.
11. Review documentation for fitness of purpose. Only the intended users of a document will be able to tell if it contains the right information presented in right way.

Overview of ISO 25010



(Figure:source ISO25010 document)
The material of ISO is unfortunate not open. But since quality matters and ISO 25010 is used heavily for managing quality aspects within business IT systems a short overview.

Functionality

Functionality: A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.

- Functional suitability: Degree to which a product or system provides functions that meet stated and implied needs when used underspecified conditions.
- Functional completeness : Degree to which the set of functions covers all the specified tasks and user objectives.
- Functional correctness : Degree to which a product or system provides the correct results with the needed degree of precision.
- Functional appropriateness : Degree to which the functions facilitate the accomplishment of specified tasks and objectives.

Reliability

Reliability: A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. Attributes:

- Reliability:Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.
- Maturity : Degree to which a system, product or component meets needs for reliability under normal operation.
- Availability : Degree to which a system, product or component is operational and accessible when required for use.
- Fault tolerance : Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.

- Recoverability : Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.

Performance Efficiency

A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

- Performance efficiency: Performance relative to the amount of resources used under stated conditions.
- Time behaviour : Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.
- Resource utilization : Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.
- Capacity : Degree to which the maximum limits of a product or system parameter meet requirements.

Compatibility

Compatibility:Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

- Co-existence : Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product.
- Interoperability : Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.

Usability

Usability:Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

- Appropriateness recognizability : Degree to which users can recognize whether a product or system is appropriate for their needs.
- Learnability : Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.
- Operability : Degree to which a product or system has attributes that make it easy to operate and control.
- User error protection : Degree to which a system protects users against making errors.
- User interface aesthetics : Degree to which a user interface enables pleasing and satisfying interaction for the user.
- Accessibility : Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

Security

Security:Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.

- Confidentiality : Degree to which a product or system ensures that data are accessible only to those authorized to have access.
- Integrity : Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.
- Non-repudiation : Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.
- Accountability : Degree to which the actions of an entity can be traced uniquely to the entity.
- Confidentiality : Degree to which a product or system ensures that data are accessible only to those authorized to have access.
- Authenticity : Degree to which the identity of a subject or resource can be proved to be the one claimed.

Maintainability

- Maintainability:Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.
- Modularity : Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.
- Reusability :Degree to which an asset can be used in more than one system, or in building other assets.
- Analysability : Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.
- Modifiability : Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.
- Testability : Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.

Transferability

- Portability:Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another
- Adaptability : Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.
- Installability : Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or un-installed in a specified environment.

- Replaceability : Degree to which a product can replace another specified software product for the same purpose in the same environment.

Note that when you do a critical review on ISO 20510 you will find that missing in ISO 25010 is:

- Functional requirements
- Compliance (e.g. with laws, standards) requirements
- Documentation, Support and Training requirements and of course:
- Project Timing requirements
- Project Budget requirements

Requirements:Defacto standards

Every serious project SHOULD have principles and/or requirements defined. Using de-facto standards as add-on for your functional requirements can speed up your project. Below a list of sources of requirements that can speed up business IT projects:

Business Process Models must be based on BPMN 2.0	
Name of REQ	Business Process Models must be based on BPMN 2.0
Type	Business Requirement
despcription	BPMN 2.0 must be used A standard Business Process Model and Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations. See: http://www.bpmn.org/
Priority	Could
Tag(s)	standard

Digital Signature Standard (DSS) (FIPS PUB 186 – 4)	
Name of REQ	Digital Signature Standard (DSS) (FIPS PUB 186 – 4)
Type	QoS(Quality-of-Service)
despcription	Digital Signature Standard (DSS) - FIPS PUB 186-4 This Standard defines methods for digital signature generation that can be used for the protection of binary data (commonly called a message), and for the

	verification and validation of those digital signatures. Three techniques are approved. (1) The Digital Signature Algorithm (DSA) is specified in this Standard. The specification includes criteria for the generation of domain parameters, for the generation of public and private key pairs, and for the generation and verification of digital signatures. (2) The RSA digital signature algorithm is specified in American National Standard (ANS) X9.31 and Public Key Cryptography Standard (PKCS) #1. FIPS 186-4 approves the use of implementations of either or both of these standards and specifies additional requirements. (3) The Elliptic Curve Digital Signature Algorithm (ECDSA) is specified in ANS X9.62. FIPS 186-4 approves the use of ECDSA and specifies additional requirements. Recommended elliptic curves for Federal Government use are provided herein. See: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
Priority	Could
Tag(s)	Security, standard

FIPS PUB 198-1:The Keyed-Hash Message Authentication Code (HMAC)	
Name of REQ	FIPS PUB 198-1:The Keyed-Hash Message Authentication Code (HMAC)
Type	Business
despcription	FIPS PUB 198-1:The Keyed-Hash Message Authentication Code (HMAC) Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity checks based on a secret key are usually called message authentication codes (MACs). Typically, message authentication codes are used between two parties that share a secret key in order to authenticate information transmitted between these parties. This Standard defines a MAC that uses a cryptographic hash function in conjunction with a secret key. This mechanism is called HMAC [HMAC]. HMAC shall use an Approved cryptographic hash function [FIPS180-3]. HMAC uses the secret key for the calculation and verification of the MACs. See:http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
Priority	Could
Tag(s)	Security, standard

Implementation must comply to the”Do Not Track Compliance Policy”	
Name of REQ	Implementation must comply to the”Do Not Track Compliance Policy”
Type	Implementation
despcription	The website (DNS) is compliant with the privacy-friendly Do Not Track (DNT) Policy of the EFF.org organization. Reference: https://www.eff.org/dnt-policy w3c reference: http://www.w3.org/TR/tracking-dnt/
Priority	Could
Tag(s)	Privacy, standard

JSON-LD	
Name of REQ	JSON-LD
Type	Implementation
despcription	JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as CouchDB and MongoDB. See: http://json-ld.org/
Priority	Must
Tag(s)	standard

Payment Card Industry (PCI) DSS v3.1	
Name of REQ	Payment Card Industry (PCI) DSS v3.1
Type	Business
despcription	The design/implementation must be compliant with the Payment Card Industry (PCI) Data Security Standard version 3.1 PCI DSS provides a baseline of technical and operational requirements designed to protect account data. PCI DSS applies to all entities involved in payment card processing — including merchants, processors, acquirers, issuers, and service providers. PCI DSS also applies to all other entities that store, process or transmit cardholder data (CHD) and/or sensitive authentication data (SAD). Detailed spec on: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1.pdf

Priority	Could
Tag(s)	Security, standard

Payment Card Industry (PCI) Point – to – Point Encryption	
Name of REQ	Payment Card Industry (PCI) Point – to – Point Encryption
Type	QoS(Quality-of-Service)
despcription	Payment Card Industry (PCI) Point - to- Point Encryption Solution Requirements and Testing Procedures Version 2.0 (Revision 1.1) The objective of this standard is to facilitate the development, approval, and deployment of PCI-approved P2PE solutions that will increase the protection of account data by encrypting that data from the point of interaction within the encryption environment where account data is captured through to the point of decrypting that data inside the decryption environment, effectively removing clear-text account data between these two points. The requirements contained within this standard are intended for P2PE solution providers and other entities that provide P2PE components or P2PE applications for use in P2PE solutions, as well as P2PE assessors evaluating these entities. Additionally, merchants benefit from using P2PE solutions due to increased protection of account data and subsequent reduction in the presence of clear -text account data within their environments. Details specs can be found at: https://www.pcisecuritystandards.org/documents/P2PE_v2_r1-1.pdf
Priority	Could
Tag(s)	Security, standard

Secure Hash Standard (SHS) – FIPS PUB 180 – 4	
Name of REQ	Secure Hash Standard (SHS) – FIPS PUB 180 – 4
Type	Implementation
despcription	This Standard specifies secure hash algorithms, SHA -1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA- 512/256. All of the algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest. These algorithms enable the determination of a message's integrity: any change to the message will, with a very high probability, result in a different message digest. The digests are used to detect whether messages have been changed since the digests were generated. See:

	http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf
Priority	Could
Tag(s)	Security, standard

Semantic Versioning	
Name of REQ	Semantic Versioning
Type	QoS(Quality-of-Service)
despcription	All version numbering must match Semver 2.0 Given a version number MAJOR.MINOR.PATCH, increment the: 1. MAJOR version when you make incompatible API changes, 2. MINOR version when you add functionality in a backwards-compatible manner, and 3. PATCH version when you make backwards-compatible bug fixes. Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format. See: http://semver.org/
Priority	Must
Tag(s)	standard

Systems must comply with BIOS Integrity Measurement Guidelines	
Name of REQ	Systems must comply with BIOS Integrity Measurement Guidelines
Type	QoS(Quality-of-Service)
despcription	The systems must implement measurements as described in the NIST Special Publication 800-155 (BIOS Integrity Measurement Guidelines). Client computers such as desktops and laptops rely on the Basic Input/Output System (BIOS) to initialize their hardware during boot. The BIOS is firmware, and it can be configured. If the BIOS code or configuration is altered from the intended state, either maliciously or accidentally, the desktop or laptop may experience losses of confidentiality, integrity, and availability, including system instability, system failure, and information leakage. See: http://csrc.nist.gov/publications/drafts/800-155/draft-SP800-155_Dec2011.pdf for more information
Priority	Could
Tag(s)	Security, standard

Web Hypertext Application Technology Working Group (WHATWG)	
Name of REQ	Web Hypertext Application Technology Working Group (WHATWG)
Type	Implementation
despcription	The Web Hypertext Application Technology Working Group (WHATWG) is a growing community of people interested in evolving the Web. It focuses primarily on the development of HTML and APIs needed for Web applications. The WHATWG was founded by individuals of Apple, the Mozilla Foundation, and Opera Software in 2004, after a W3C workshop. Apple, Mozilla and Opera were becoming increasingly concerned about the W3C's direction with XHTML, lack of interest in HTML and apparent disregard for the needs of real-world authors. So, in response, these organisations set out with a mission to address these concerns and the Web Hypertext Application Technology Working Group was born. See: https://whatwg.org/
Priority	Must
Tag(s)	Browser, standard

NFR Requirements list

Having solid NFR (Non Functional Requirements) makes the difference between successful business IT projects and IT disasters. Below a list of NFR requirements that are ready to be (re)used within your project:

A test specification for the system must be available	
Name of REQ	A test specification for the system must be available
Type	Business Requirement
despcription	A test specification for the system must be available in order to perform test of the created system.
Priority	Could
Tag(s)	Documentation, NFR

Data logging:Sensitive data is not logged in clear text by the application.

Name of REQ	Data logging:Sensitive data is not logged in clear text by the application.
Type	Implementation
despcription	Sensitive data is not logged in clear text by the application.
Priority	Could
Tag(s)	NFR, Security

Database connections, passwords, keys, or other secrets are not stored in plain text.	
Name of REQ	Database connections, passwords, keys, or other secrets are not stored in plain text.
Type	Business
despcription	Database connections, passwords, keys, or other secrets are not stored in plain text.
Priority	Could
Tag(s)	NFR, Security

Disaster Recovery	
Name of REQ	Disaster Recovery
Type	Business Requirement
despcription	The solution will be configured to be split across the two data centers where possible with failover from data center to data center in the event of a disaster, In addition, each data center needs to be able to run in a self sufficient manner should it become isolated from the other.
Priority	Could
Tag(s)	NFR

Documentation must be available in an open document format	
Name of REQ	Documentation must be available in an open document format
Type	System Requirement

despcription	All system documentation must be made available in open document format. System documentation is (not exhausted) operational manuals, code documentation, test specs and test reports, installation manuals.
Priority	Could
Tag(s)	Documentation, NFR

High Availability	
Name of REQ	High Availability
Type	Business Requirement
despcription	All components should be configured in a high availability configuration to eliminate single points of failure, and minimize solution outages.
Priority	Could
Tag(s)	NFR

Maintainability	
Name of REQ	Maintainability
Type	Business Requirement
despcription	The system should allow for easy software upgrades with minimal outage. The outage should be restricted to no longer than one day, and allow for the use of a back up system for service continuity while the upgrade the taking place.
Priority	Could
Tag(s)	NFR

Maintainability	
Name of REQ	Maintainability
Type	System Requirement
despcription	Any solution must be maintainable by the affected maintenance team, both initially and throughout its lifecycle. Unnecessary complexity in maintenance, such as by requiring additional / unusual skills or tools or having a complex solution design, adds risk to the solution’s

	supportability and must be justified.
Priority	Could
Tag(s)	maintainability, NFR

Manageability	
Name of REQ	Manageability
Type	System Requirement
despcription	All solutions must be managed throughout their lifecycle, including startup, shutdown, backup, updates, security / permission changes, etc. Administrators and support personnel must be able to conduct such routine activities effectively in order to ensure that the solution does not incur excessive cost or experience unnecessary outages.
Priority	Could
Tag(s)	manageability, NFR

Minimize Footprint	
Name of REQ	Minimize Footprint
Type	Business Requirement
despcription	Stack multiple components within single operating system instances where possible to minimize both the number of physical and virtual servers required to run the solution.
Priority	Could
Tag(s)	NFR

Service levels	
Name of REQ	Service levels
Type	System Requirement
despcription	The required service level(s) for any solution affect its design, cost, maintenance and support. As such, these must be known.
Priority	Could
Tag(s)	NFR, service levels

Supportability	
Name of REQ	Supportability
Type	IT Requirement
despcription	Any solution must be supportable by the affected operations and maintenance teams, both initially and throughout its lifecycle. Unnecessary complexity in support, such as by requiring additional / unusual skills or tools, adds risk to the solution's supportability and must be justified. The maintenance support team will be responsible for the success of the solution during its production life. As such, they require training, mentoring and appropriate transition measures to ensure they are able to successfully support the new component in the production environment.
Priority	Could
Tag(s)	NFR, supportability

The certificate must be an X.509v3 certificate	
Name of REQ	The certificate must be an X.509v3 certificate
Type	Business
despcription	The certificate must be an X.509v3 certificate. The certificate must be within the valid period. The certificate must be verified and validated through authentication. The system will not issue digital certificates. Users will present trusted third party-issued certificates that are valid and verifiable by the system.
Priority	Could
Tag(s)	NFR, Security

User ID must be unique and passwords must be stored in irreversible encrypted form	
Name of REQ	User ID must be unique and passwords must be stored in irreversible encrypted form
Type	Business
despcription	User ID must be unique. Passwords must be stored in irreversible encrypted form, and the password file cannot be viewed in unencrypted form. A password must not be displayed on the data

	entry/display device. Passwords must be at least eight characters long. Passwords must be composed of at least three of the following: English uppercase letters, English lowercase letters, numeric characters, and special characters. Password lifetime will not exceed 60 days Users cannot use the previous six passwords. The system will give the user a choice of alternative passwords from which to choose. Passwords must be changed by the user after initial logon.
Priority	Could
Tag(s)	NFR, Security

Architectures References

The field of business IT Architecture is covered with many many methods, foundations, industry organizations and reference architectures. So whatever the specific domain you are active in use and re-use information already there as described a solid reference architecture. And do not forget: When you publish your architecture document on the web, make sure to use a common creative license, so others can improve, reuse and build upon your work.

Reference architectures

Link Name	Link Description
ATHENA Interoperability Framework (AIF)	Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application. Find here the public architecture deliverables of the ATHENA project.
Australian Government Architecture Reference Models	The AGA Reference Models provide a common language for Australian Government agencies so that their architectures can be described in a common and consistent manner. cc-by licensed material See also: http://www.finance.gov.au/policy-guides-procurement/australian-government-architecture-aga/aga-rm/
Cloud Computing Portability and Interoperability (Open Group)	Open Group document. This guide analyzes cloud computing portability and interoperability.
Cybersecurity Framework (NIST)	The Framework Core offers a way to take a high-level Security Framework.
DoDAF (Department of Defense Architecture Framework)	Reference architecture of the Department of Defense (US) The DoDAF Architecture Framework Version 2.02
EAM Pattern Catalog	The objective of the EAM Pattern Catalog is to complement existing Enterprise Architecture (EA) management frameworks
Eucalyptus Cloud Reference	A catalog of private cloud reference architecture and architectural layers that can be used as the foundation for production

Architectures	deployments.
EURIDICE	EURIDICE is an EU funded project which deals with the development and implementation of new concepts in the area of intelligent Cargo.
GEMMA (GEMeentelijk Model Architectuur)	(Dutch site) GEMMA 2.0 is een doorontwikkeling naar een architectuur die de gehele gemeentelijke informatievoorziening beschrijft, helpt bij het reduceren van de complexiteit van de informatievoorziening, bij het organiseren van samenwerkingsverbanden en het positioneren van functies in de cloud.
GERAM (Generalised Enterprise Reference Architecture and Methodology)	The scope of GERAM encompasses all knowledge needed for enterprise engineering / integration. Thus GERAM is defined through a pragmatic approach providing a generalised framework for describing the components needed in all types of enterprise engineering/enterprise integration processes.
IBM Cloud Computing Reference Architecture 2.0	Currently adopted by the Open Group
Interoperability Solutions for EU public administrations	Reference architecture documents for use and reuse developed within EU program. ISA's collaborative platform to find, reuse and share a wealth of ready-to-use interoperability solutions for eGovernment and best practices and discuss with your peers!
ITAG – Information Technology Architecture Group	The MIT Enterprise Architecture Guide (EAG) documents MIT's architectural principles and goals, the current state of MIT's enterprise architecture, and a future state architectural vision. The EAG also includes information regarding the ITAG architecture review process. Since this document serves to inform developers about available enterprise tools and services, we expect the EAG will be useful to enterprise system developers across the institute.
ITSM Reference Architecture Framework	FitSM is a free and lightweight standards family aimed at facilitating service management in IT service provision, including federated scenarios.
Microsoft Industry Reference Architecture for Banking (MIRA-B)	MIRA-B This 2012 Microsoft Industry Reference Architecture for Banking gives financial institutions a framework to ensure IT meets their strategic goals across channels and various customer needs.
Mobile Security	The Mobile Security Reference Architecture (MSRA) is a deliverable

Reference Architecture	of the Digital Government Strategy (DGS). A key objective of the DGS is to procure and manage mobile devices, applications, and data in smart, secure, and affordable ways. The MSRA has been released by the Federal CIO Council and the Department of Homeland Security (DHS) to assist Federal Departments and Agencies (D/As) in the secure implementation of mobile solutions through their enterprise architectures
NEXOF-RA	The overall ambition of NEXOF-RA is to deliver a Reference Architecture for the NESSI Open Service Framework (ranging from the infrastructure up to the interfaces with the end users) leveraging research in the area of service-based systems to consolidate and trigger innovation in service-oriented economies
NIH Enterprise Architecture Framework (National Institute of Health)	As a comprehensive framework the NIH Enterprise Architecture identifies how IT assets directly enable NIH
NIST Cloud Computing Reference Architecture (Version 2)	NIST Cloud Computing Reference Architecture.
NORA 3.0	Dutch Government Reference Architecture (version 3.0))
Open Security Architecture (OSA)	OSA distills the know-how of the security architecture community and provides readily usable patterns for your application. OSA shall be a free framework that is developed and owned by the community.
Oracle Enterprise Architecture Framework : Information Architecture Domain	The OEAF:Information Architecture. Oracle EA Information reference architecture. OEAF Domain consists of the following components: Data Realms, Capability Model
Oracle Reference Architecture Security Release 3.1	This document (2010) provides a reference architecture for designing an enterprise security framework. This framework supports the security needs of business solutions and helps to unify the disparate security resources commonly found in IT today. It offers security services that are critical to the integrity of modern distributed and service-oriented solutions, and beneficial to legacy systems as well.
Reference	Description of scalability and deployment options when using AWS

Architecture for Scalable Word Press Websites (on AWS)	for hosting of WorPress sites.
Reference Architecture Foundation for Service Oriented Architecture	This document specifies the OASIS Reference Architecture for Service Oriented Architecture. It follows from the concepts and relationships defined in the OASIS Reference Model for Service Oriented Architecture. While it remains abstract in nature
ROSA (Referentie Onderwijs Sector Architectuur)	(Dutch) Reference Architecture site for Education Sector.
SOA Reference Architecture	This specification presents a SOA Reference Architecture (SOA RA)
Software Assurance Maturity Model	The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.
The Chromium Architecture (Google)	Complete architecture and design documentation of the Google Chromium Projects.
VMware Infrastructure Architecture Overview	VMware Infrastructure architecture Overview (PDF whitepaper).
VMware vSphere	VMware vSphere

Industry Architectures

Link Name	Link Description
BIAN (Banking Industry Architecture Network)	The BIAN model is a Service Oriented Architecture with consistent service definitions, level of detail and boundaries. This makes it easier to choose and integrate commercially available products of different vendors.
Distributed Management Task Force	DMTF's Systems Management Architecture for Server Hardware (SMASH) standard is a suite of specifications that deliver industry standard semantics, protocols and profiles to make data center

	resource management interoperable.
EURIDICE	EURIDICE is an EU funded project which deals with the development and implementation of new concepts in the area of intelligent Cargo.
EURIDICE	Overview of the EURIDICE architecture main concepts and components. (Link to all public architecture documents). EURIDICE Integrated Project Euridice is an Integrated Project funded by EU's Seventh Framework Programme ICT for Transport Area. The basic concept of Euridice is to build an information services platform centred on the individual cargo item and on its interaction with the surrounding environment and the user.

Cloud

Link Name	Link Description
Cloud computing patterns	CloudPatterns.org is a community site dedicated to documenting a master patterns catalog comprised of design patterns that capture and modularize technology-centric solutions distinct or relevant to modern-day cloud computing platforms and business-centric cloud technology architectures. Part of this catalog is comprised of compound patterns that tackle contemporary cloud delivery and deployment models (such as public cloud, IaaS, etc.) and decompose them into sets of co-existent patterns that establish core and optional feature sets provided by these environments.
Eucalyptus Cloud Reference Architectures	A catalog of private cloud reference architecture and architectural layers that can be used as the foundation for production deployments.
IBM Cloud Computing Reference Architecture 2.0	Currently adopted by the Open Group
NIST Cloud Computing Reference Architecture (Version 2)	NIST Cloud Computing Reference Architecture.

Foundation Architectures

Link Name	Link Description
EAM Pattern Catalog	The objective of the EAM Pattern Catalog is to complement existing enterprise architecture (EA) management frameworks, which provide a holistic and generic view on the problem of EA management, and to provide additional detail and guidance needed to systematically establish EA management in a step-wise fashion within an enterprise.
FSAM (Federal Segment Architecture Methodology)	The Architecture and Infrastructure Committee released the Federal Segment Architecture Methodology (FSAM) v1.0 in December 2008. The FSAM features easy-to-use templates that expedite architecture development and maximize architecture use. The FSAM includes step by step guidance based on business-driven

Architecture magazines

Link Name	Link Description
Architectuur en Governance magazine	Architecture & Governance Magazine is a publication of Troux Technologies.
DoDAF V2.02 Journal	The DoDAF Journal is a community of interest based discussion board. The Journal includes descriptions of best practices.
Journal of Enterprise Architecture (AOGEA Journal)	The Journal of Enterprise Architecture (JEA) is published quarterly by the Association of Enterprise Architects. It is a peer-reviewed international quarterly publication for the Enterprise Architecture community. Standard magazine for all TOGAF certified architects...
Journal of Information Architecture	The Journal of Information Architecture is an international peer-reviewed scholarly journal. Its aim is to facilitate the systematic development of the scientific body of knowledge in the field of information architecture.
The Architecture Journal (Microsoft)	The Architecture Journal is an independent platform for free thinkers and practitioners of IT architecture. New editions are issued quarterly with articles designed to offer perspective
XR Magazine	XR Magazine is Dutch online platform and magazine for managers and architecten. (Language dutch).

Security architecture

Link Name	Link Description
Cybersecurity Framework (NIST)	The Framework Core offers a way to take a high-level Security Framework.
Mobile Security Reference Architecture	The Mobile Security Reference Architecture (MSRA) is a deliverable of the Digital Government Strategy (DGS). A key objective of the DGS is to procure and manage mobile devices, applications, and data in smart, secure, and affordable ways. The MSRA has been released by the Federal CIO Council and the Department of Homeland Security (DHS) to assist Federal Departments and Agencies (D/As) in the secure implementation of mobile solutions through their enterprise architectures
Open Security Architecture (OSA)	OSA distills the know-how of the security architecture community and provides readily usable patterns for your application. OSA shall be a free framework that is developed and owned by the community.
Open Web Application Security Project (OWASP)	The Open Web Application Security Project (OWASP) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software.
Oracle Reference Architecture Security Release 3.1	This document (2010) provides a reference architecture for designing an enterprise security framework. This framework supports the security needs of business solutions and helps to unify the disparate security resources commonly found in IT today. It offers security services that are critical to the integrity of modern distributed and service-oriented solutions, and beneficial to legacy systems as well.
Privacy Management Reference Model and Methodology (PMRM)	The Privacy Management Reference Model and Methodology (PMRM, pronounced “pim-rim”) provides a model and a methodology for: · understanding and analyzing privacy policies and their privacy management requirements in defined use cases; and · selecting the technical services which must be implemented to support privacy controls. It is particularly relevant for use cases in which personal information (PI) flows across regulatory, policy, jurisdictional, and system boundaries.
SABSA (Sherwood Applied Business Security Architecture)	SABSA is a proven methodology for developing business-driven

Software Assurance Maturity Model	The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

Architecture Methods

Link Name	Link Description
Archimate	Archimate 1.0 version: The ArchiMate enterprise architecture modeling language offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies. In a short time
ARIS (Architecture of Integrated Information Systems)	Long time a industry default standard. Not anymore however -) ARIS is now more a tool of Software AG's. ARIS Business Process Analysis Platform is ideal for organizations that want to document
Cloud Computing Patterns	Describing good solutions to reoccurring problems as patterns is a common practice in research and industry alike. While the development of cloud applications faces many new challenges
DYA	Primary Dutch EA method. Used and owned by Sogetti.
DYA Infrastructure Repository	DYA|Infrastructure brings business agility
EAM Pattern Catalog	The objective of the EAM Pattern Catalog is to complement existing Enterprise Architecture (EA) management frameworks
EAM Pattern Catalog	The objective of the EAM Pattern Catalog is to complement existing enterprise architecture (EA) management frameworks, which provide a holistic and generic view on the problem of EA management, and to provide additional detail and guidance needed to systematically establish EA management in a step-wise fashion within an enterprise.
FSAM (Federal Segment Architecture Methodology)	The Architecture and Infrastructure Committee released the Federal Segment Architecture Methodology (FSAM) v1.0 in December 2008. The FSAM features easy-to-use templates that expedite architecture development and maximize architecture use. The FSAM includes step by step guidance based on business-driven
FSAM (Federal Segment Architecture Methodology)	The Architecture and Infrastructure Committee released the Federal Segment Architecture Methodology (FSAM) v1.0 in December 2008. The FSAM features easy-to-use templates that

	expedite architecture development and maximize architecture use. The FSAM includes step by step guidance based on business-driven
GERAM (Generalised Enterprise Reference Architecture and Methodology)	The scope of GERAM encompasses all knowledge needed for enterprise engineering / integration. Thus GERAM is defined through a pragmatic approach providing a generalised framework for describing the components needed in all types of enterprise engineering/enterprise integration processes.
TOGAF	TOGAF is a framework - a detailed method and a set of supporting tools - for developing an enterprise architecture.

Architecture organizations

Link Name	Link Description
Association of Enterprise Architects (AEA)	The Association of Enterprise Architects (AEA) is the definitive professional organization for Enterprise Architects. Our goals are to increase job opportunities for all members and increase their market value by advancing professional excellence
BIAN (Banking Industry Architecture Network)	The BIAN model is a Service Oriented Architecture with consistent service definitions, level of detail and boundaries. This makes it easier to choose and integrate commercially available products of different vendors.
Center for Enterprise Architecture (EA)	The purpose of the Center for Enterprise Architecture is to gather intellectual resources across Penn State to address open and important research concerns and questions that span the design
Distributed Management Task Force	DMTF's Systems Management Architecture for Server Hardware (SMASH) standard is a suite of specifications that deliver industry standard semantics, protocols and profiles to make data center resource management interoperable.
Enterprise Architecture Center of Excellence (EACOE)	The mission of the Enterprise Architecture Center of Excellence (EACOE) is to be the definitive source for all aspects of Enterprise Architecture
IASA (Global IT Architects Association)	Iasa is the premier association focused on the architecture profession through the advancement of best practices and education while delivering programs and services to IT architects of all levels around

	the world. Our mission is to make IT architecture the most recognized profession in the world.
ITAG – Information Technology Architecture Group	The MIT Enterprise Architecture Guide (EAG) documents MIT's architectural principles and goals, the current state of MIT's enterprise architecture, and a future state architectural vision. The EAG also includes information regarding the ITAG architecture review process. Since this document serves to inform developers about available enterprise tools and services, we expect the EAG will be useful to enterprise system developers across the institute.
NAF (Nederlands Architectuur Forum)	Dutch organization for promoting working with the IT architecture discipline. (EA driven)
NGI architectuur	Dutch department under NGI-NGN (Dutch non-profit IT organization for IT professionals). This architecture group works with couple with other Dutch EA groups when organizing meetings and creating workgroups.
Software Engineering Institute (SEI)	The Software Engineering Institute (SEI) is a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD). It is operated by Carnegie Mellon University. The SEI offers many (free) publication on all aspects of software architecture.
The Information Architecture Institute	The Information Architecture Institute is a 501(c)6 professional organization
The Jericho Forum	The Jericho Forum
The MOD research group	Research group on model-driven software engineering at SINTEF The MOD research group is part of the Department of Networked Systems and Services within the Division of Information and Communication Technology. This group is located in Oslo
The Open Group	The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations
Via Nova Architectura	Dutch EA platform. Via Nova Architectura wil de primaire informatiebron zijn op het gebied van architectuur en iedereen bereiken die in het onderwerp ge

Appendix:General tips for creating an architecture

Creating an architecture document in a complex environment, very demanding stakeholders, a complex innovative product or an old undocumented legacy system can be hard.

The best way to deal with problems is communication, over and over again. However the following tips derived from real world architecture challenges can speed up creating your architecture.

- Document economically (“Less is often more”).
- Focus on explanation and rationale, not only facts.
- Highlight and focus on business objectives of the system.
- Make a clear distinguish between architecture, design and implementation descriptions. Architecture is mostly on the ‘WHAT’ and ‘HOW’. A good architecture SHOULD give strict guidelines and constraints regarding the implementation.
- Always work with explicit quality requirements. Or if no quality requirements exist, make the the chosen assumptions explicit.
- Show the context graphically. A picture is most of the time easier to understand than plain text!
- For external interfaces to other systems, create discrete interface documents. Only mention the interface in the main architecture document.
- Always keep version control of your documentation in mind. A MUST when you distribute documents or offer the option to download architecture documents of a site.

Appendix: Maintenance of this architecture playbook

This is a living document and will continue to be updated with more helpful information and examples as they become available.

The latest version of this architecture playbook is always on: https://nocomplexity.com/architecture-playbook/

PDF and ePUB versions are derived from this online version.

Send any errors, remarks, suggestions and improvements to info@nocomplexity.com

This book is open source. See Github https://github.com/nocomplexity/ArchitecturePlaybook for the book content, the accounts infrastructure, and I welcome issues and pull requests.

-

Appendix:About Maikel Mardjan

Maikel is a business architect and loves to make designs for complex IT systems in a simple way. Maikel is always in for improving your security architecture and is not afraid to make his hands dirty when things get rough during implementation. Maikel has more than 20 years of relevant experience on various IT roles in famous (international) companies. Maikels loves to create good architectures and designs that really bring new success to companies.

Maikel holds both a Master (Msc) Business Studies of University of Groningen and a Master degree (Msc) Electrical Engineering, of Delft University of Technology. Maikel is TOGAF 9 Certified and CISSP (Certified Information Systems Security Professional) certified. Maikel holds his on innovative IT company https://nocomplexity.com. Despite privacy concerns, Maikel can be found on Twitter too https://twitter.com/maikelmardjan

Dutch speaking designers are invited to visit https://organisatieontwerp.nl This is my home for all my native dutch speaking collegues and friends!

Appendix:How to Contribute

Help make this website the best resource for business IT Architecture!
Best way to contribute is the simplest one: Spread the Word! Share this eBook or the site link of the book: https://nocomplexity.com/ with your friends.

I will make regular updates to this book and will make sure the content stays up-to-date.
I like to receive your contributions, mail your contributions to: info@nocomplexity.com
If you are using github, you can also visit the github page for this book at:
https://github.com/nocomplexity

Appendix:Copyright and notices

This book is provided “as-is” and expresses the author's views and opinions. The views, opinions, and information expressed in this book, including URL and other Internet website references, may change without notice. All trademarks are property of their respective owners.

Version August 2016
License cover image: CC0

- No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.