European
Commission

SAT
for
eHI

[EIRA extension]

e-HI (Human Interface) Solution Architecture Template (SAT)

v1.0.0 Beta

Change control

Modification	Details
Version 1.0.0 Beta	

TABLE OF CONTENTS

1 INTRODUCTION

This document contains the description for a Solution Architecture Document (SAT) for e-HI, Human Interface.

This SAT is based on EIRA v1.1.0.

SAT eHI v1_0_0
Beta.archimate

SAT eHI v1_0_0 Beta
- ArchiMate.xml

1.1 Purpose of this document

Enterprise and Solution architects can use this document and the related SAT model to design solution architectures that have a Human Interface (as opposed to a machine-to-machine interface) as part of the solution. Note that the proposed solution architecture template is EC centric.

1.2 List of acronyms used in this document

ABB	Architecture Building Block
AJAX	Asynchronous JavaScript and XML
CORS	Cross-origin resource sharing
CSS	Cascading StyleSheet
DOM	Document Object Model
EIRA	European Interoperability Reference Architecture
HI	Human Interface
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
JWT	JSON Web Token
OWASP	Open Web Application Security Project
PE	Progressive Enhancement
PWA	Progressive Web Application
RIA	Rich Internet Application
RWD	Responsive Web Design
SAT	Solution Architecture Template
SBB	Solution Building Block
SEO	Search Engine Optimisation
SERP	Search Engine Results Page
UI	User Interface
URI	Uniform Resource Identifier
UX	User eXperience
W3C	World Wide Web Consortium
WAI-ARIA	Web Accessibility Initiative - Accessible Rich Internet Applications
WCAG	Web Content Accessibility Guidelines
XLIFF	XML Localisation Interchange File Format
XML	eXtensible Markup Language

2 GOAL, DESCRIPTION AND TARGET AUDIENCE

This chapter provides the goal of this SAT as well as a description on (web-based) human interfaces and indicates the target audience and the potential use of this Solution Architecture Template (SAT).

2.1 Goal

The purpose of this SAT is to provide guidance by defining a minimal, but holistic (legal, organisational, semantic and technical) interoperability architecture to develop an EC-centric web-based solution. The SAT should allow businesses, citizens and public administrations to have a common understanding of the most salient building blocks.

2.2 What is a User Interface

A User Interface is the result of design activities over some technologies (devices, user agents, frameworks and libraries) offering all the interactions between a human user and an information system to achieve some business goals. The execution path of these interactions constitutes the user experience.

In the scope of the European Commission, specific requirements are often captured in specific frameworks, such as eUI which is developed by DIGIT AOB (Architecture Office). Those frameworks facilitate the build of UI by providing sample applications, reusable components and widgets.

This SAT describes aspects of the Human Interface, it does not say anything about User Experience (UX), which is a strongly related topic, as it deals with responsive design, responsiveness, the fact that data must be 'recognisable', look and feel, templating etc. UX is considered out-of-scope for this SAT.

Although the term and technologies still relatively new, we briefly discuss "Progressive Web Applications (PWAs)" in the "Technical view – application" of this SAT. It is not a requirement as such, but is considered very relevant to the future of web-based user interfaces.

2.3 What is a solution architecture template (SAT)

A Solution Architecture Template (SAT) is a specification extending the EIRA providing support to solution architects in a specific solution domain. An SAT contains a motivation (principles, requirements), a goal and a description of the supported functionalities, a sub-set of the EIRA core Architecture Building Blocks (ABBs) covering the four views, a set of specific ABBs extending EIRA's views enabling specific functionalities to be provided by implementations derived from the SAT and the interoperability specifications of selected ABBs and a narrative for each EIRA view.

The benefits of a SAT are the following:

- Provides architects with a common approach to cope with a specific interoperability challenge. It also places the focus on the key-points you need to consider.

- An architect can create a solution architecture by mapping existing Solution Building Blocks (SBBs) to an SAT, based on the interoperability specifications that are provided. This is done by providing SBBs for the ABBs identified in the SAT.

- When an architect creates an SAT, he/she can define the interoperability specifications for the SAT's ABBs and moreover recommend specific SBBs which produces faster and more interoperable results.

- An SAT can be created within and across the different views of the EIRA. An SAT can then support architects specialised in different interoperability levels."

2.4 Target audience

This document has the following target audience:

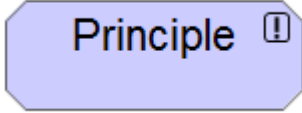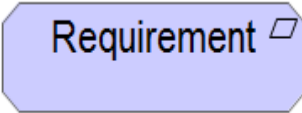| Audience	Description
Architect	Enterprise/solution architects in the need of understanding, implementing, or describing a Human Interface/ Rich Internet Application, as it is used within the European Commission and within the Member States.
Public Administration / Members States	Public Administrations of the European Union that need to have a holistic view in terms of architecture in relation to ensuring interoperability around a human to machine interface.

3 HUMAN INTERFACE INTEROPERABILITY MAPPED TO THE EIRA

This chapter contains for each EIRA view the corresponding ArchiMate model and narrative. Next to the SAT's EIRA architecture building blocks, the ArchiMate model includes, where applicable, the related specifications, principles and requirements.

The models have been scaled down to fit with the text, they are included in bigger format in the appendix.

3.1 ArchiMate motivation extension

In addition to the standard EIRA concepts, the diagrams use the following concepts coming from the ArchiMate motivation extension

Non-EIRA concept	Description
Principle ⚠	A principle is defined as a normative property of all systems in a given context.
Requirement ⬜	A requirement is defined as a statement of need that must be realized by a system.

The following principles and requirements are used in this SAT:

Requirement: Responsive Web Design (RWD)

Responsive web design (RWD) is an approach to web design aimed at crafting sites to provide an optimal viewing and interaction experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from desktop computer monitors to mobile phones).

Principle: Web Accessibility

Web Content Accessibility Guidelines (WCAG) 1.0 consist of 14 guidelines—each of which describes a general principle of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints that describes how to apply that guideline to particular webpage features.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use mark-up and style sheets, and do so properly
- Guideline 4: Clarify natural language usage
- Guideline 5: Create tables that transform gracefully
- Guideline 6: Ensure that pages featuring new technologies transform gracefully
- Guideline 7: Ensure user control of time sensitive content changes
- Guideline 8: Ensure direct accessibility of embedded user interfaces

- Guideline 9: Design for device independence
- Guideline 10: User interim solutions
- Guideline 11: Use w3c technologies and guidelines
- Guideline 12: Provide context and orientation information
- Guideline 13: Provide clear navigation mechanisms
- Guideline 14: Ensure that documents are clear and simple

3.2 How to use this SAT

An architect that uses this SAT typically wants to perform a gap-analysis or conformance check between an existing solution and this Solution Architecture Template, or he/she wants to model a solution of which part that interacts with the user through a 'Rich Internet Application' and uses this document as guidance.

Although this SAT takes a lot of inspiration from the e-UI initiative, which examines UX, browser compatibility and UI frameworks (for example: "Kendo UI", jQuery, bootstrap, Vaadin, …), it is written with a broader scope, it explains how any Rich Internet Application (RIA) can be modelled, without making statements about the frameworks that could or should be used.

3.2.1 Gap Analysis or conformance check

Using this SAT for gap analysis, the architect can map the building blocks of the solution to the ones in this SAT and identify which building blocks are missing or to ensure that the correct SBBs and IoP Specs are employed. These building blocks can either indicate missing functionality or missing interoperability specifications.

3.2.2 Building a solution

When building a solution, the architect is expected to use the four different EIRA views and provide a solution in the form of Solution Building Blocks (SBBs) for the Architecture Building Blocks (ABBs) that are indicated. This is done by replacing the Architecture Building Block with an annotated Solution Building Block. The existing SBBs in this SAT should not be removed and replaced, however, the acknowledgement of reusing these building blocks can be done by removing the ABBs which they specialise.

Interoperability Specifications (IoP specs) are added as specialisation of an Interoperability ABB, implemented in the form of an SBB and attached to an ABB as interoperability requirements. The final solution should only contain the implementation (the SBB) of the IoP Spec.

The result will be a solution architecture that will contain only SBBs, all ABBs should have been removed (in the case this SAT already provides SBBs for this ABB) or replaced by SBBs (solutions that implement that ABB).

The SAT is a document describing the needed Architecture Building Blocks for a desired solution. This should not be taken as restrictive but as advisory. When an Architecture Building Block (ABB) is present for which there is no implementation foreseen in the form of a Solution Building Block (SBB), it is *strongly* recommended, but not mandatory, to take this ABB into consideration in the final solution.

3.3 Legal View

The Legal view of the SAT consists of the following sub-set of EIRA Architecture Building Blocks (ABBs) as well as a number of predefined Solution Building Blocks (SBBs):
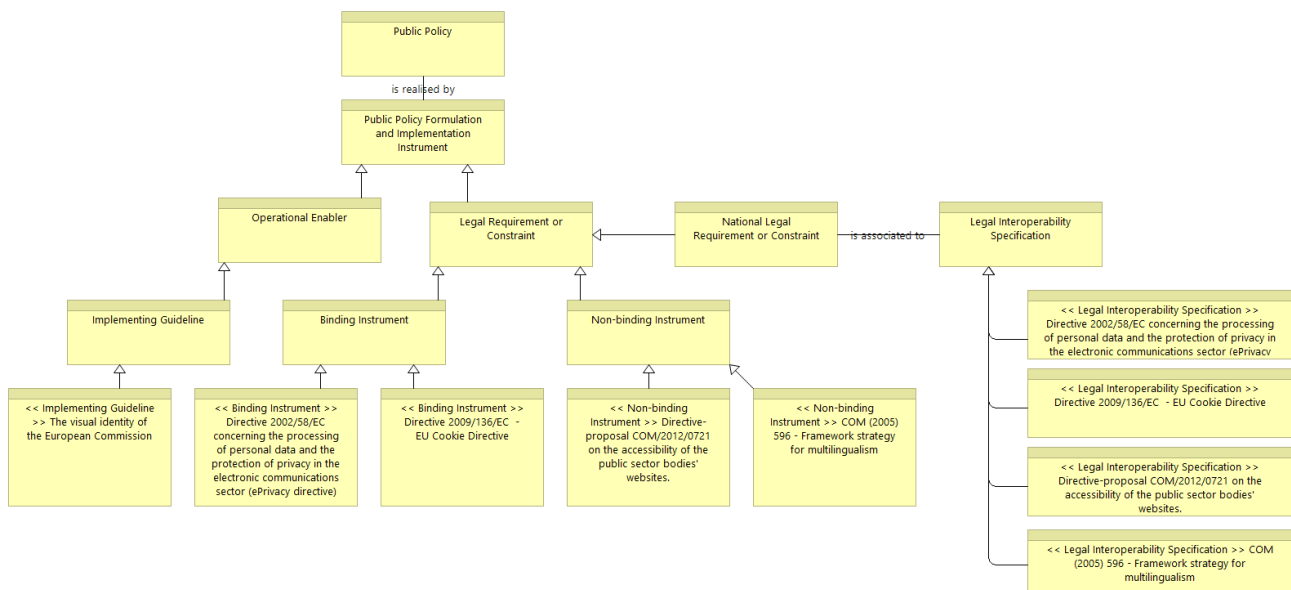
Directive 2002/58 on Privacy and Electronic Communications, otherwise known as E-Privacy Directive, is an EU directive on data protection and privacy in the digital age. It presents a continuation of earlier efforts, most directly the Data Protection Directive. It deals with the regulation of a number of important issues such as confidentiality of information, treatment of traffic data, spam and cookies. This Directive has been amended by Directive 2009/136, which introduces several changes, especially in what concerns cookies, that are now subject to prior consent.[1]

The Directive COM/2012/0721 (proposal) supports Member States to achieve their national commitments regarding web-accessibility as well as their commitment to the United Nations Convention on the Rights of Persons with Disabilities regarding websites of public sector bodies.

Web-accessibility refers to principles and techniques to be observed when constructing websites, in order to render the content of these websites accessible to all users, in particular those with disabilities.

Multilingualism refers to both a person's ability to use several languages and the co-existence of different language communities in one geographical area. The Commission's multilingualism policy has three aims:

---

[1] https://en.wikipedia.org/wiki/Directive_on_Privacy_and_Electronic_Communications

- to encourage language learning and promote linguistic diversity in society;
- to promote a healthy multilingual economy;
- to give citizens access to European Union legislation, procedures and information in their own languages.

Communication COM (2005) 596 sets out various activities for promoting language learning and linguistic diversity.

The Visual Identity of the European Commission is an implementing guideline which gives the European Commission a recognisable and coherent image. The visual identity is constructed around the European Commission's logo. This is based on two key elements: the European flag and a graphic element inspired by the headquarters of the European Commission.

3.4 Organisational View

The Organisational view of the SAT consists of the following sub-set of EIRA Architecture Building Blocks (ABBs)
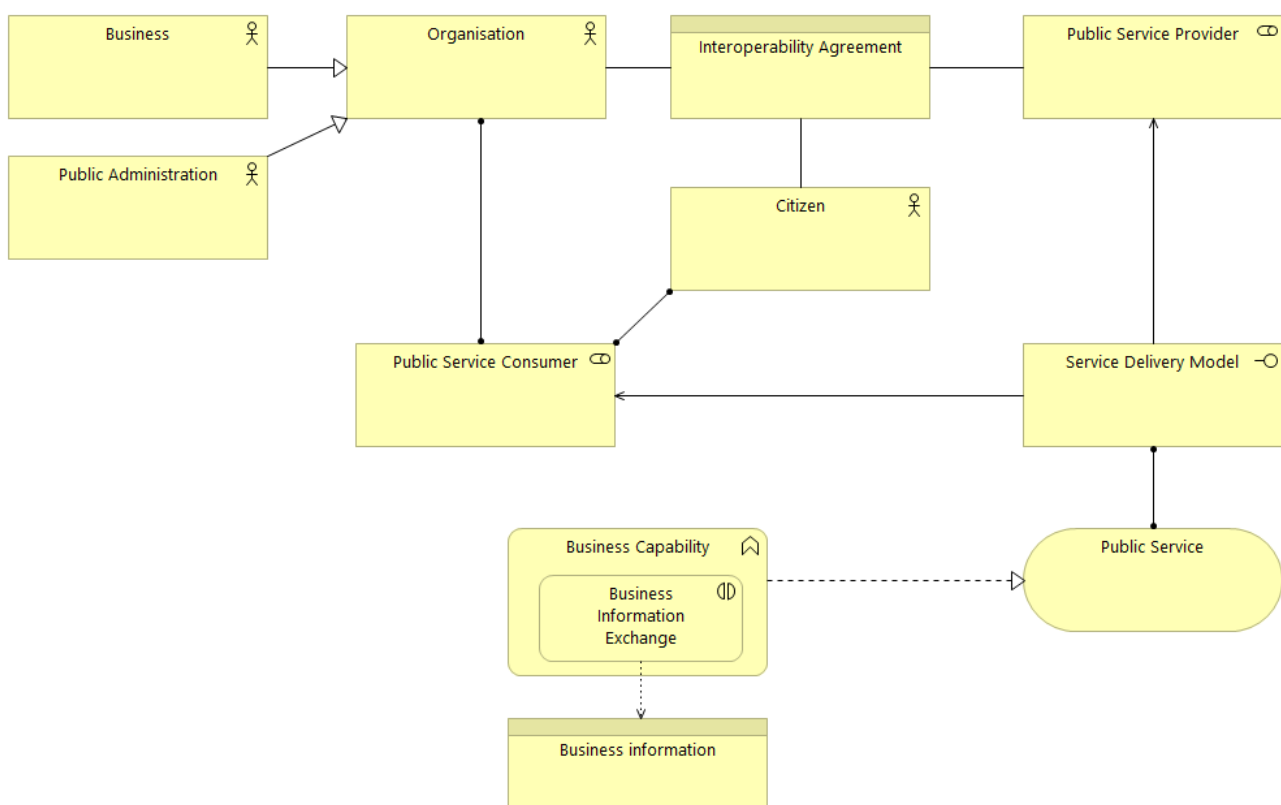
Considering that the definition of User Interface, as it is used in this SAT, is totally agnostic of business for which the information is displayed, this view does not prescribe any solution building blocks. It tells the user that any solution needs to consider interoperability agreements with its consumers (either business, public administrations or citizens) and a public service provider. This interoperability agreement will be described in a "Service Delivery Model" which defines how the organisation arranges the delivery of its services to service consumers. This (public) service is realised by a business capability which is the expression or the articulation of the capacity, materials and expertise an organization needs in order to perform core functions. The business information listed is an interaction between two or more public administrations, businesses or citizens.

3.5 Semantic View

The Semantic view of the SAT consists of the following sub-set of EIRA Architecture Building Blocks (ABBs) as well as a number of predefined Solution Building Blocks (SBBs):
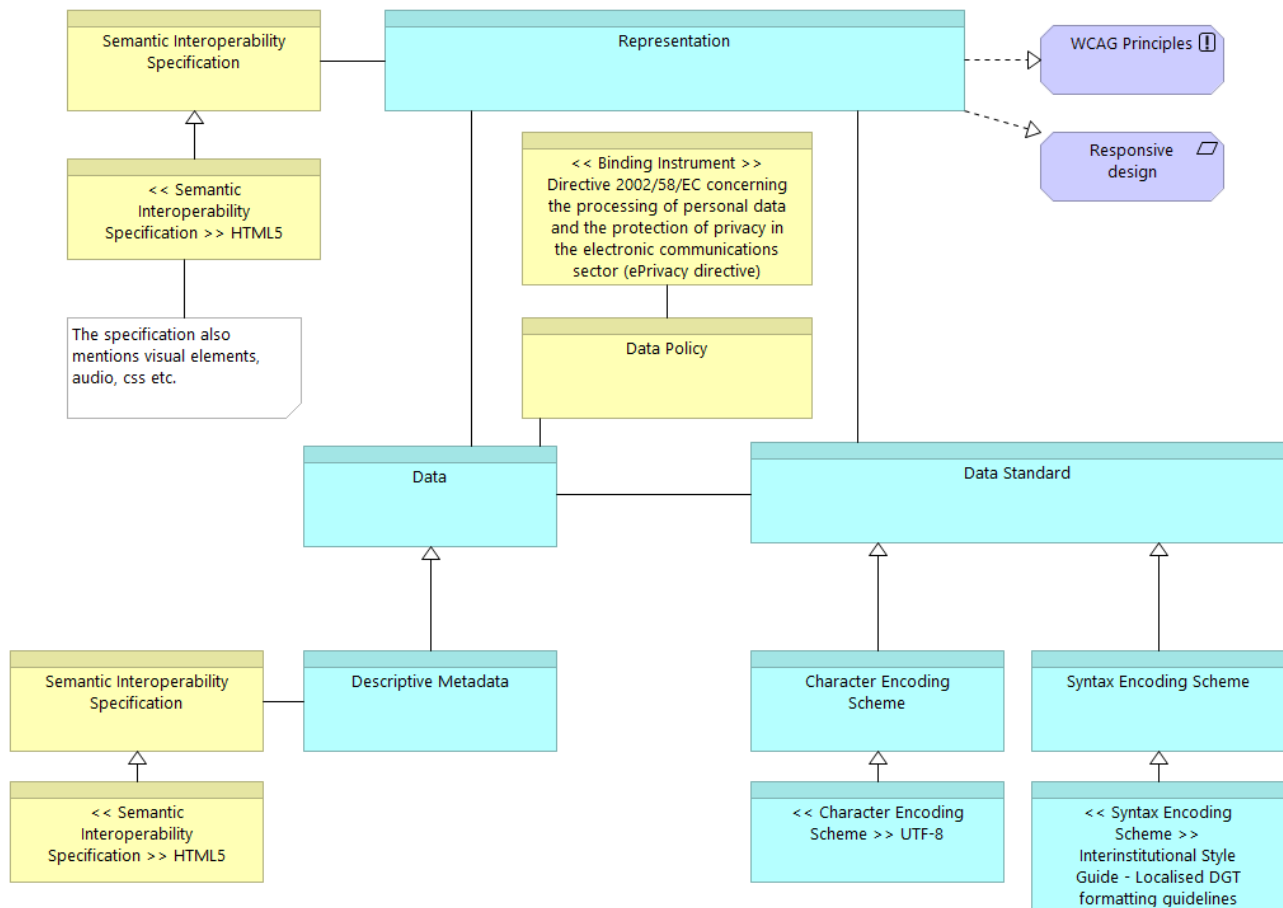
HTML5 is used as umbrella specification, containing pointers to many other specifications related to styling, multimedia etc. The status of the HTML5 specification is still 'Recommendation', but all major browsers (IE starting from IE9) now support this standard, as can be seen in the following image (source: http://fmbip.com/litmus/)

HTML5 Graphics & Embedded Content	MAC						WIN						
	SAFARI	FIREFOX		OPERA	CHROME		SAFARI	IE				FIREFOX	CHROME
	5.1	8	9	11.1	15	17	5.1	6	7	8	9	8	15
Canvas	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Canvas Text	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG Clipping Paths	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG Inline	✔	✔	✔	✘	✔	✔	✔	✘	✘	✘	✔	✔	✔
SMIL	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
WebGL	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Audio	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Video	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔

Note that this image does not indicate CSS support nor "HTML Web Applications", those charts are included in a dedicated appendix of this document.

3.5.1 Search Engine Optimisation (SEO)

This meta-tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable. Meta elements are typically used to specify page description, keywords, the author of the document, last modified time and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

Some search engines may display the meta description as a part of the search results, but the meta keyword tags should not appear in search results. The meta description tag serves the function of advertising copy. It draws readers to a website from the Search Engine Results Page (SERP) and thus, is an extremely important part of search marketing. Crafting a readable, compelling description using important keywords can improve the click-through rate for a given webpage.

3.5.2 Accessibility metadata

Accessibility metadata is, put simply, metadata that describes the accessibility of resources and services, usually those on, or available through, the web.

It is realised that much of the web content was not accessible to people who did not use standard web GUI browsers. In particular, low-vision and blind people, people with motor coordination problems, in fact, people including those who could not use a mouse on a computer screen for one reason or another, were not able to use their computers as their life-style-support machines.

The World Wide Web Consortium responded by establishing a Web Accessibility Initiative (WAI) program to work on what was making web content inaccessible.[2]

3.5.3 Wizards

Wizards are an implementation of a UI design pattern with the same name. They allow the user to perform a task step by step, allowing to save an intermediate state, which may be subsequently available via a direct link. The information remains persistent throughout multiple HTTP sessions, i.e. distinct user connections that the user would make, with the information stored in these steps allowing subsequent invalidation, thus resembling long-running transactions. We did not model them as such, since there is no notion of automatic roll-back, which needs to be implemented. There are multiple ways to implement this, manually for light processes, or using workflow/business process engines for more complicated flows.

---

[2] http://dublincore.org/groups/access/standards.html

3.6 Technical View – Application

The Technical view – application of the SAT consists of the following sub-set of EIRA Architecture Building Blocks (ABBs) as well as a number of predefined Solution Building Blocks (SBBs):
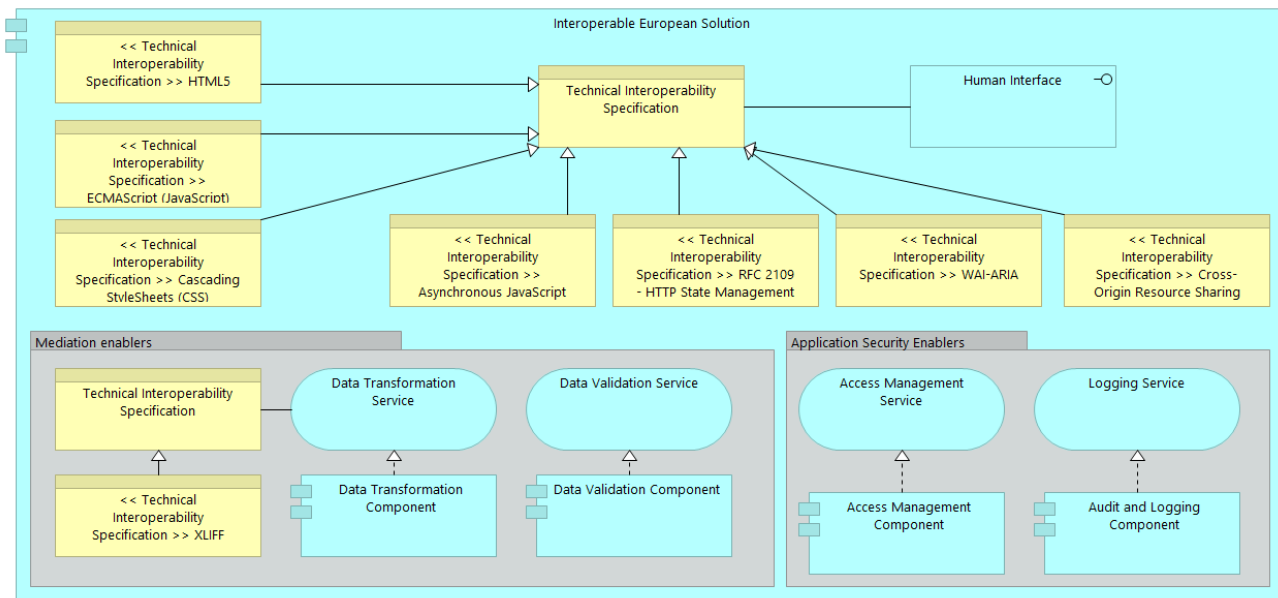
3.6.1 Asynchronous JavaScript and XML (AJAX)

Although AJAX (Asynchronous JavaScript and XML) is not a specification in the strictest sense of the meaning, it enables interoperability from a user interface perspective. AJAX is not a technology, but a group of technologies. HTML and CSS can be used in combination to mark-up and style information. The DOM is accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads. Note that nowadays, most AJAX communication is done using the JSON (JavaScript Object Notation) format instead of XML, but this is not presented as a requirement.

AJAX is a set of web development techniques using numerous client-side web to create asynchronous Web applications. With AJAX, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. By decoupling the data interchange layer from the presentation layer, AJAX allows for web pages, and by extension web applications, to change content dynamically without the need to reload the entire page and thus ensuring a much more streamlined user experience.

3.6.2 Cookies

An HTTP cookie (also called web cookie, Internet cookie, browser cookie or simply cookie) is a small piece of data sent from a website and stored in the user's web browser while the user is browsing. Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items added in the shopping cart in an online store) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited in the past).

3.6.3 Web Accessibility Initiative – Accessible Rich Internet Applications

WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) is a technical specification published by the World Wide Web Consortium (W3C) that specifies how to increase

the accessibility of web pages, in particular, dynamic content and user interface components developed with AJAX, HTML, JavaScript and related technologies.

3.6.4 Cross-origin resource sharing (CORS)

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts) on a web page to be requested from another domain outside the domain from which the resource originated. A web page may freely embed images, stylesheets, scripts, iframes, videos and certain plugin content (such as Adobe Flash) from any other domain. However, embedded web fonts and AJAX (XMLHttpRequest) requests have traditionally been limited to accessing the same domain as the parent web page (as per the same-origin security policy). "Cross-domain" AJAX requests are forbidden by default because of their ability to perform advanced requests (POST, PUT, DELETE and other types of HTTP requests, along with specifying custom HTTP headers) that introduce many cross-site scripting security issues. CORS defines a way in which a browser and server can interact to determine safely whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests.[3]

3.6.5 Application security and mediation enablers

The application security enablers are added as supporting services for any application, although it is important to realise that the logging of user-interactions may conflict with the ePrivacy regulation, it is important to examine what may be logged and what may not be logged.

As mediation enablers, the Data Validation (a UI should always be considered untrusted) and data transformation service, where XLIFF is added as technical specification to data transformation. XLIFF (XML Localisation Interchange File Format) is an XML-based format created to standardize the way localizable data are passed between tools during a localization process.

3.6.6 Progressive Web Applications (PWAs)

Progressive Web Apps are experiences that combine the best of the web and the best of apps. They are useful to users from the very first visit in a browser tab, no install required. As the user progressively builds a relationship with the App over time, it becomes more and more powerful. It loads quickly, even on flaky networks, sends relevant push notifications, has an icon on the home screen and loads as a top-level, full screen experience.[4]

Progressive Web Apps are:

- Progressive - Work for every user, regardless of browser choice because they're built with progressive enhancement as a core tenet.

- Responsive - Fit any form factor: desktop, mobile, tablet, or whatever is next.

- Connectivity independent - Enhanced with service workers to work offline or on low quality networks.

- App-like - Feel like an app to the user with app-style interactions and navigation because they're built on the app shell model.

- Fresh - Always up-to-date thanks to the service worker update process.

---

[3] https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

[4] https://developers.google.com/web/fundamentals/getting-started/your-first-progressive-web-app/

- Safe - Served via HTTPS to prevent snooping and ensure content hasn't been tampered with.

- Discoverable - Are identifiable as "applications" thanks to W3C manifests and service worker registration scope allowing search engines to find them.

- Re-engageable - Make re-engagement easy through features like push notifications.

- Installable - Allow users to "keep" apps they find most useful on their home screen without the hassle of an app store.

- Linkable - Easily share via URL and not require complex installation.

Progressive Web Applications are based on the Progressive Enhancement strategy which uses web technologies in a layered fashion that allows everyone to access the basic content and functionality of a web page, using any browser or Internet connection, while also providing an enhanced version of the page to those with more advanced browser software or greater bandwidth.

Graceful degradation is the practice of building your web functionality so that it provides a certain level of user experience in more modern browsers, but it will also degrade gracefully to a lower level of user in experience in older browsers. This lower level is not as nice to use for your site visitors, but it does still provide them with the basic functionality that they came to your site to use; things do not break for them.

Progressive enhancement is similar, but it does things the other way round. You start by establishing a basic level of user experience that all browsers will be able to provide when rendering your web site, but you also build in more advanced functionality that will automatically be available to browsers that can use it.

Progressive Enhancement consists of the following core principles:

- Basic content should be accessible to all web browsers

- Basic functionality should be accessible to all web browsers

- Sparse, semantic mark-up contains all content

- Enhanced layout is provided by externally linked CSS

- Enhanced behaviour is provided by unobtrusive, externally linked JavaScript

- End-user web browser preferences are respected

Progressive Web Applications are typically targeted to browser mobile web apps, it brings features that we expect from native applications to the mobile browser experience in a way that uses standards based technologies that are described in this SAT.

3.7 Technical View – Infrastructure

The Technical view – infrastructure of the SAT consists of the following sub-set of EIRA Architecture Building Blocks (ABBs) as well as a number of predefined Solution Building Blocks (SBBs):
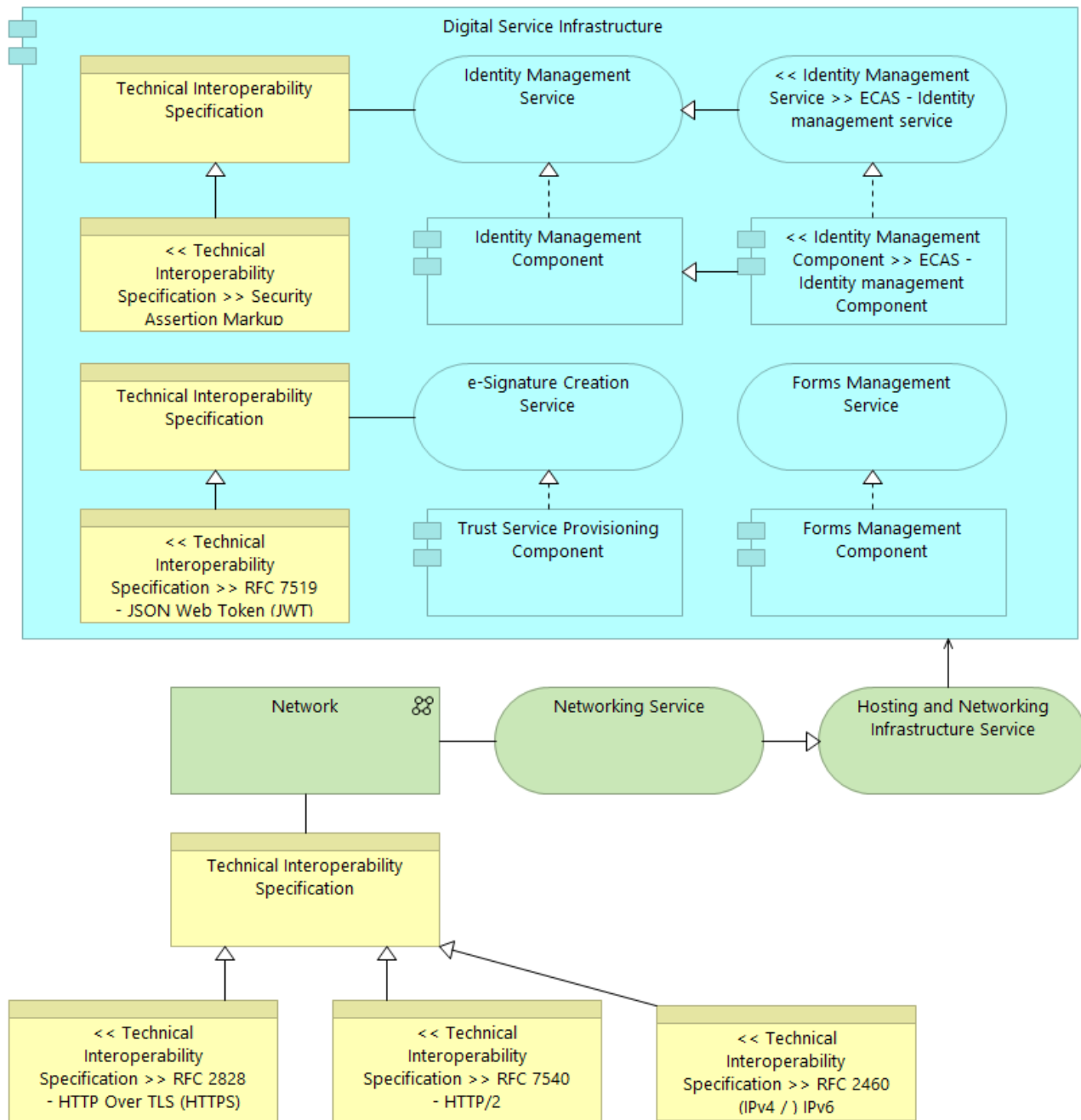
JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA.

Form management are included to tackle document upload which often is problematic. Identify management is added with an implementation attached; ECAS. The reason why ECAS is listed as both service and component is related to the fact that often, a server needs to identify itself to another server, on behalf of a client, a scenario that typically is related to AJAX calls which retrieve

information based on an identity. For this, a component needs to be added that acts as a user proxy.

Although using ECAS as an identity management solution is not strictly necessary for Member States, they will need to use Security Assertion Markup Language (SAML) as a technical specification for identity management. SAML is a format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider, the most important requirement that SAML addresses is web browser single sign-on (SSO).

4 SECURITY

Although security is not explicitly mentioned in the EIRA, it is a very important factor to seriously consider when designing a solution.

The Open Web Application Security Project (OWASP) has a list of 10 most critical web application security risks:

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross-Site Request Forgery (CSRF)
- Using Known Vulnerable Components
- Unvalidated Redirects and Forwards

Note that this list changes on a regular basis, the list above is the latest version dating of 2013. A data call to update this list has been announced, which will result in the OWASP top 10, 2016 version.

5 FUTURE WORK

The current release of this SAT is considered as a Beta version in anticipation of future improvements in terms of identifying additional interoperability specifications for its defined ABBs. This activity is foreseen to be structured as follows:

1. Identify the set of ABBs that, in the domain of this SAT, are considered as key interoperability enablers.

2. Ensure that for each such key ABB there is an identified interoperability specification. The identification of such specifications shall be the result of expert analysis and review.

Feedback and input in relation to this work, as well as any other suggestions for improvement, are more than welcome. Please direct any such input you may have as an email to DIGIT-EIRA@ec.europa.eu.

6 ACKNOWLEDGEMENTS

7 REFERENCES

- European Interoperability Reference Architecture (EIRA)

  https://joinup.ec.europa.eu/asset/eia/

- European Interoperability Framework (EIF)

  http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf

- ArchiMate®

  http://www.opengroup.org/subjectareas/enterprise/archimate

- Archi®

  http://www.archimatetool.com/

- OWASP

  https://www.owasp.org/

7.1 Legislative references

- Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (ePrivacy directive)

  http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML

- The visual identity of the European Commission

  http://ec.europa.eu/dgs/communication/services/visual_identity/index_en.htm

- COM/2005/0596 - A New Framework Strategy for Multilingualism

  http://eur-lex.europa.eu/legal-content/GA/ALL/?uri=celex:52005DC0596

  http://europa.eu/pol/mult/index_en.htm

- Directive-proposal (COM/2012/0721) on the accessibility of the public sector bodies' websites.

  http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P7-TA-2014-0158+0+DOC+XML+V0//EN

  http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52012PC0721&from=EN

7.2 Organisational references

7.3 Semantical references

- HTML5

  https://www.w3.org/TR/html5/

- Accessibility Metadata Project

  http://www.a11ymetadata.org/

- Accessibility Metadata: a rich mix of standards

  Liddy Nevile, ACRI, Latrobe University, Australia

  http://dublincore.org/groups/access/standards.html

7.4 Technical references

- XLIFF

  http://docs.oasis-open.org/xliff/xliff-core/v2.0/xliff-core-v2.0.html

- WAI-ARIA

  https://www.w3.org/TR/wai-aria/

- RFC 2828 - HTTP Over TLS (HTTPS)

  https://tools.ietf.org/html/rfc2818

- RFC 7519 - JSON Web Token (JWT)

  https://tools.ietf.org/html/rfc7519

- RFC 2109 - HTTP State Management Mechanism (Cookies)

  https://tools.ietf.org/html/rfc6265

- Security Assertion Markup Language (SAML)

  https://www.oasis-open.org/committees/security/

- RFC 7540 - HTTP/2

  https://tools.ietf.org/html/rfc7540

- CORS - Cross-Origin Resource Sharing

  https://www.w3.org/TR/cors/

- AJAX - Asynchronous JavaScript and XML

  http://adaptivepath.org/ideas/ajax-new-approach-web-applications/

- HTML5

  https://www.w3.org/TR/html5/

- Standard ECMA-262, ECMAScript® 2016 Language Specification

  http://www.ecma-international.org/publications/standards/Ecma-262.htm

- Cascading StyleSheets

  https://www.w3.org/Style/CSS/Overview.en.html

- Progressive Web Applications

  https://developers.google.com/web/progressive-web-apps/

8 APPENDIX: LEGAL VIEW

Public Policy

is realised by

Public Policy Formulation and Implementation Instrument

Operational Enabler

Legal Requirement or Constraint

National Legal Requirement or Constraint

is associated to

Legal Interoperability Specification

Implementing Guideline

Binding Instrument

Non-binding Instrument

<< Implementing Guideline >> The visual identity of the European Commission

<< Binding Instrument >> Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (ePrivacy directive)

<< Binding Instrument >> Directive 2009/136/EC - EU Cookie Directive

<< Non-binding Instrument >> Directive-proposal COM/2012/0721 on the accessibility of the public sector bodies' websites.

<< Non-binding Instrument >> COM (2005) 596 - Framework strategy for multilingualism

<< Legal Interoperability Specification >> Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (ePrivacy

<< Legal Interoperability Specification >> Directive 2009/136/EC - EU Cookie Directive

<< Legal Interoperability Specification >> Directive-proposal COM/2012/0721 on the accessibility of the public sector bodies' websites.

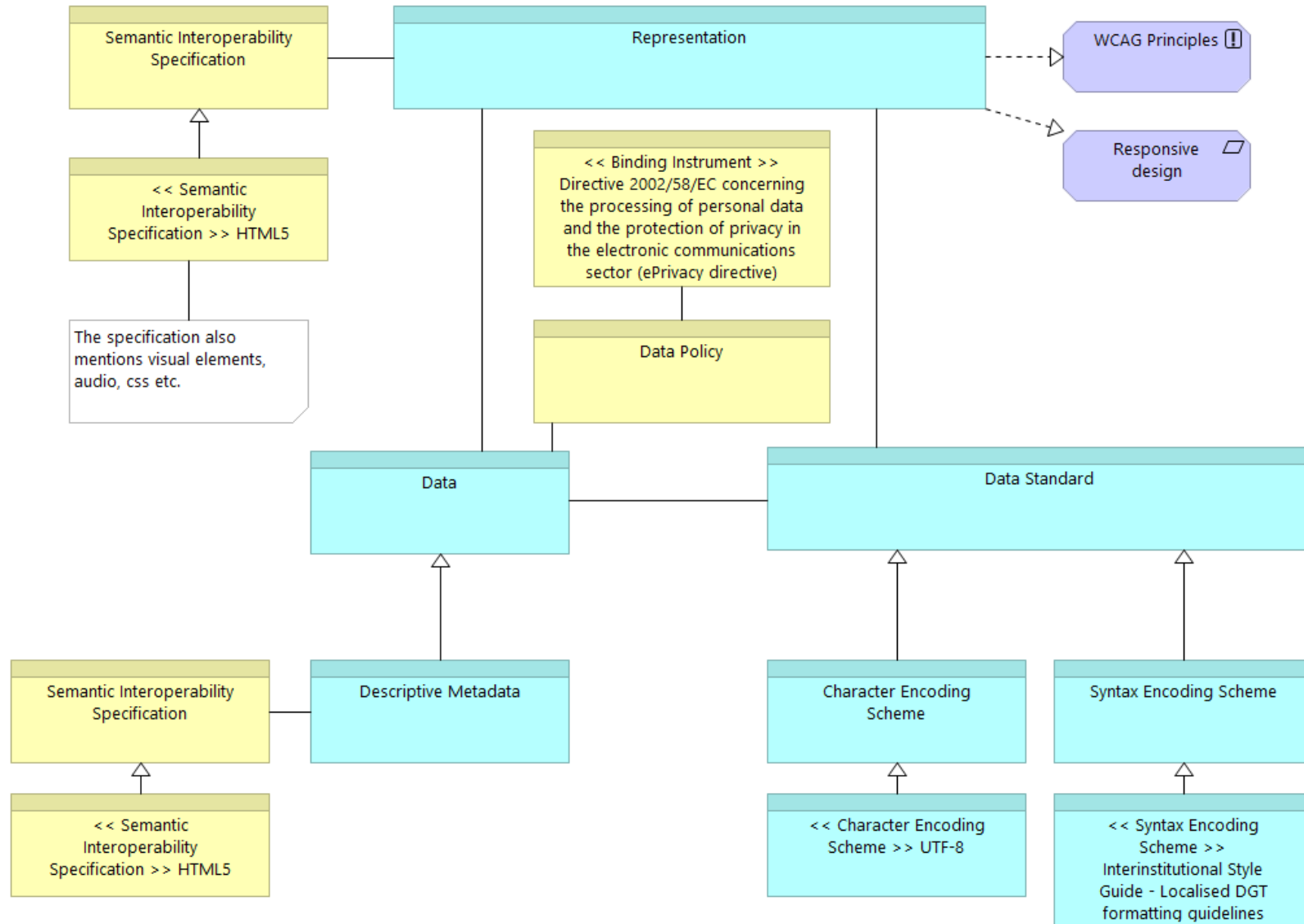<< Legal Interoperability Specification >> COM (2005) 596 - Framework strategy for multilingualism
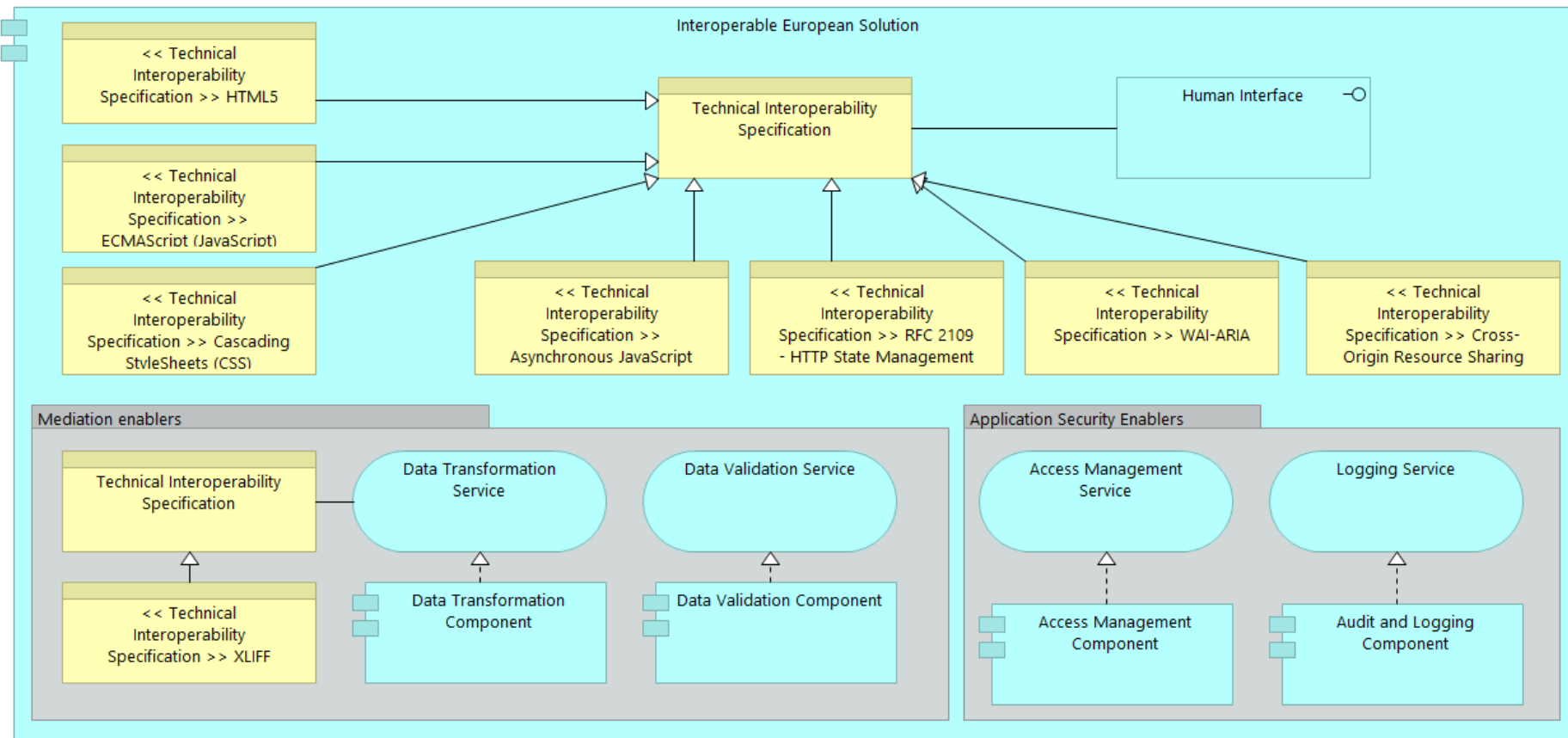
9 APPENDIX: ORGANISATIONAL VIEW
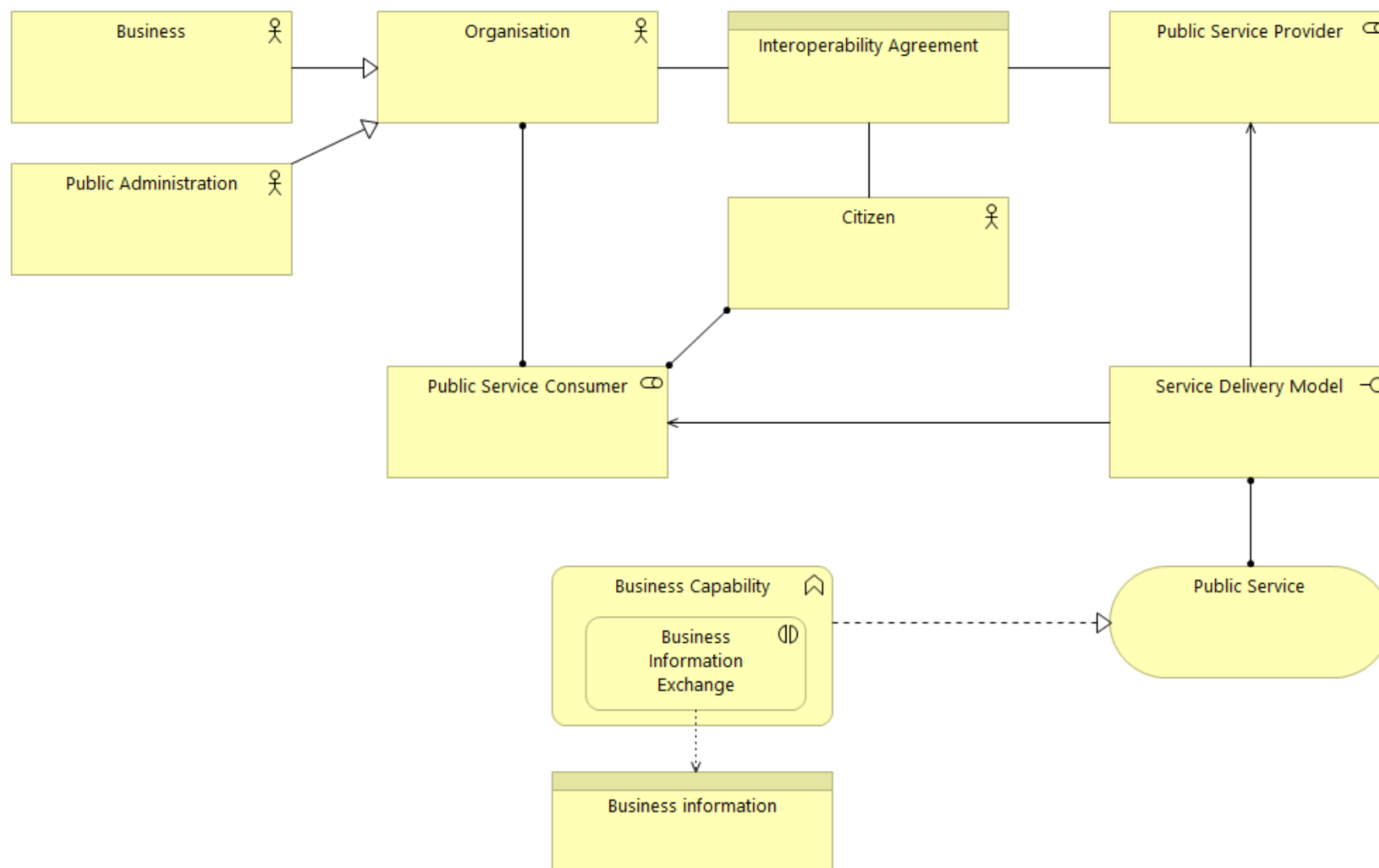
10 APPENDIX: SEMANTIC VIEW

11 APPENDIX: TECHNICAL VIEW – APPLICATION

12 APPENDIX: TECHNICAL VIEW – INFRASTRUCTURE
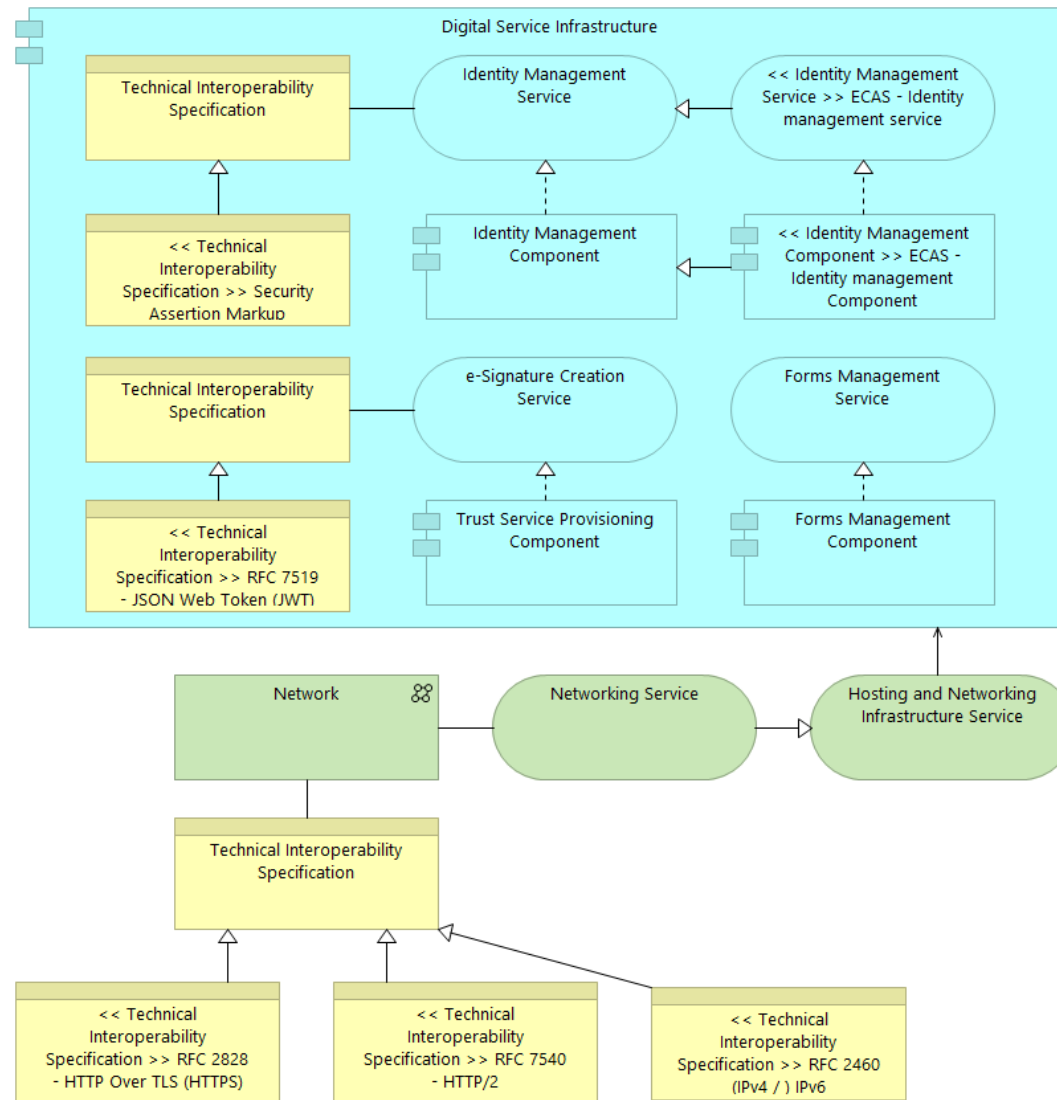
13 APPENDIX: HTML5 BROWSER COMPLIANCE

HTML5 Graphics & Embedded Content

	MAC					WIN						
	SAFARI	FIREFOX		OPERA	CHROME		SAFARI	IE				FIREFOX	CHROME
	5.1	8	9	11.1	15	17	5.1	6	7	8	9	8	15
Canvas	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Canvas Text	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG Clipping Paths	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
SVG Inline	✔	✔	✔	✘	✔	✔	✔	✘	✘	✘	✔	✔	✔
SMIL	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
WebGL	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Audio	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Video	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔

(full information available here: http://fmbip.com/litmus/)

HTML5 Web Applications

	MAC				WIN				
	SAFARI	FIREFOX	OPERA	CHROME	SAFARI	IE		FIREFOX	CHROME
	5.1	8	9	11.1	15	17	5.1	6	7	8	9	8	15
Local Storage	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
Session Storage	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
Post Message	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
Offline Applications	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✘	✔	✔
Workers	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✘	✔	✔
Query Selector	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
WebSQL Database	✔	✘	✘	✔	✔	✔	✔	✘	✘	✘	✘	✘	✔
IndexDB Database	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
Drag and Drop	✔	✔	✔	✘	✔	✔	✔	✔	✔	✔	✔	✔	✔
Hash Change (Event)	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
History Management	✔	✔	✔	✘	✔	✔	✔	✘	✘	✘	✘	✔	✔
WebSockets	✔	✘	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	✔
GeoLocation	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	✔
Touch	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘

(full information available here: http://fmbip.com/litmus/)

CSS3 Properties

	MAC								WIN							
	CHROME	FIREFOX	OPERA		SAFARI				CHROME	FIREFOX	OPERA		IE			
	25	20	12	15	1	5.1	6	25	15	12	6	7	8	9	10	
RGBA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	93%
HSLA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	93%
Box Sizing	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	55%
Background Size	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	92%
Multiple Backgrounds	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	92%
Border Image	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✘	✘	87%
Border Radius	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	93%
Box Shadow	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	93%
Text Shadow	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✘	✔	82%
Opacity	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✔	93%

(full information available here: http://fmbip.com/litmus/)