Submitted to the EC on XXX/XX/20XX

Justice Action Grant

Me-CODEX II

Maintenance of e-Justice Communication via Online Data Exchange

Call identifier: JUST-2015-JACC-AG-1
Project full title: Maintenance of e-Justice Communication via Online Data Exchange
Grant agreement n°: 785818

e-CODEX infrastructure security recommendations

Deliverable Id:	n/a
Deliverable Name:	e-CODEX infrastructure security recommendations
Status:	Draft
Dissemination Level:	me-CODEX II – Management Board (after approval WP3)
Due date of deliverable:	NA
Actual submission date:	
Work Package:	WP 3
Organisation name of lead partner for this deliverable:	Austria
Author(s):	Jack Hanser, Huub Moelker
Partner(s) contributing:	Ministry of Justice and Security in the Netherlands

Abstract:
Security requirements and recommendations should bring the level of trust and security in e-CODEX information exchanges to a higher level. The build of trust will attract more participants to European digital business collaboration. The alignment of national security policies into a European security policy will ease the implementation efforts taken by the respective Member States. This document provides insights as to what, why and how building trust can be established.

History

Version	Date	Changes made	Modified by
0.1	19-3-2020	1st draft document	Jack Hanser
0.2	18-5-2020	Introduction (based on WP3 call recommendation)	Huub Moelker
0.3	14-9-2020	Review comments WP3 participants processed	Huub Moelker
0.9	21-10-2020	Version for e-CODEX Management Board approval	Huub Moelker
1.0	26-11-2020	MB approved	Huub Moelker

Table of contents

List of abbreviations

Acronym	Explanation
CA	Certificate Authority
e-CODEX	e-Justice Communication via Online Data Exchange
HSM	Hardware Security Module
MLS	Message Layer Security
SSL	Secure Socket Layer
TLS	Transport Layer Security
ICT/IT	Information and Communication Technology
Me-CODEX	Maintenance of e-Justice Communication via Online Data Exchange
MS	Member States of the European Union

Table 1: Abbreviations

1 Introduction

One of the most important components of online business collaborations is creating a trusted environment where the involved partners feel confident in exchanging information. Especially in the domain of civil and criminal justice where the information exchanged often is of sensitive nature, the trust assurance must be beyond any doubt. In order to protect information in and between IT systems, an adequate security policy must be defined. Moreover, the policy must be maintained over time in order to keep up with the latest technological developments and cyber threats. Deploying security measures in a dynamic environment such as IT is, and always will be a rat race.

Like all other aspects in the e-CODEX approach a common European e-Justice security policy should ensure security interoperability between participants, based on the EU core principles of subsidiarity and proportionality. Partner countries are most likely to have some form of internal security policy. An analysis of these partner specific policies, should lead to a common security policy on the European level.

Aiming to create a common European perspective on IT security a number of legal frameworks have been established.

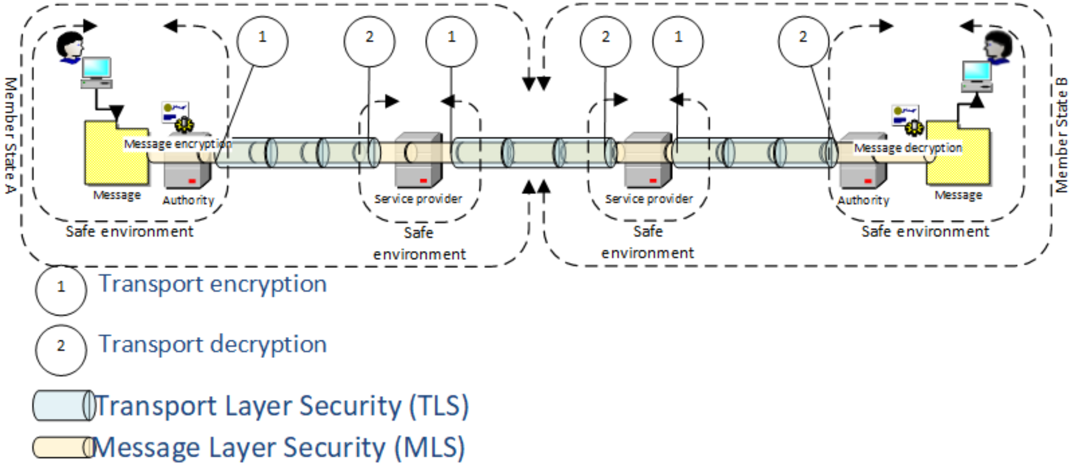
Cooperation by Member States should facilitate the technical interoperability of the notified electronic identification schemes with a view to fostering a high level of trust and security appropriate for the degree of risk. The exchange of information and the sharing of best practices between Member States with a view to their mutual recognition should help such cooperation. (source: EU regulation 910/2014 on electronic identification and trust services)

2 Security layers

In the context of e-CODEX information exchange, two levels on which security solutions can be deployed are described in this document; Transport Layer Security and Message Level Security.

1. **Transport Layer Security**: it secures the actual package that is transported between the nodes that provide the transport. As in e-CODEX the principle of the 4-corner model is adopted, information exchange between an ‘original sender’ and the ‘final recipient’ is serviced by ‘e-CODEX service providers’ for each of the respective business partners, the transport route consists of three so called ‘hops’[1];
    a. Hop #1: original sender to service provider of the sending Member State.
    b. Hop #2: service provider sending state to service provider receiving state.
    c. Hop #3: service provider receiving state to final recipient.

    For each ‘hop’ the channel between the nodes is encrypted. This kind of security is deployed by default in all of the current e-CODEX business collaborations. However, there are different ways of deploying the ‘hop-encryption’. This is further explained in Chapter 3.
2. **Message Level Security**: It encrypts the actual business payload. In e-CODEX terms, the case related information is encrypted in such a way that it can only be decrypted by the entity for which the information is intended. Distinction is made between business information and routing information. For the purpose of service providers who serve multiple back-end entities (final recipients) being able to determine for which final recipient the message is intended, but are not to be allowed access to message content. MLS is further explained in Chapter 4.



-

[1] HOP: term used to indicate that two business entities are interconnected through intermediary service providers. See: https://www.oasis-open.org/committees/download.php/22830/ebms_core-3.0-spec-wd-17-diff.pdf **(page 9-119)**

3 Transport Layer Security

A secure Transport Layer Security (TLS) configuration is important for securing network connections. So why TLS? TLS protects communication between a client and a server. Protecting communications is especially important when sending sensitive information over a connection. Information can be sensitive due to confidentiality and integrity constraints. In some cases, the use of encrypted connections is mandatory. This liability can be included in the policy of an organization, but can also be laid down in legislation and regulations. TLS offers strict and less strict settings. While we don't want to go for the less secure settings, we also don't want to go for the most strict settings because of the loss of interoperability.

Example:

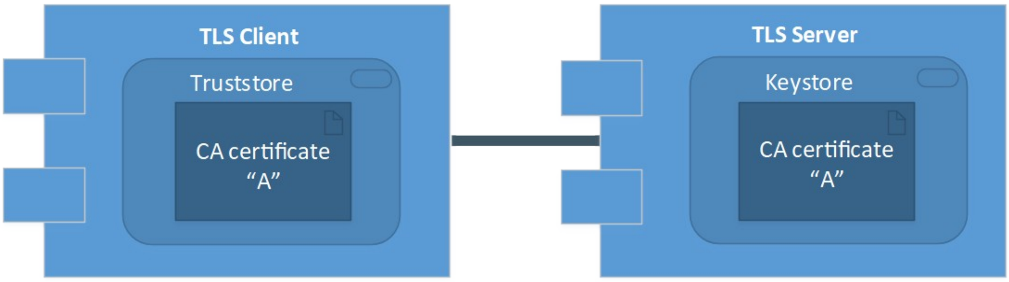
One of the most important attack vectors that we have to deal with in our case is data collection. With proper signing we always know from whom the message was sent and it cannot not be manipulated. With encryption we guarantee that, should the message be intercepted, it cannot be read. In combination with mutual TLS and the use of the latest ciphers, we ensure that a lot of other risks are mitigated; for example 'Man-in the-middle attacks' or execution. We also check the payloads and validate messages.

3.1 Server and client certificates

To guarantee a high level of security the use of mutual TLS is MANDATORY. For mutual authentication, client certificate authentication MUST be allowed.

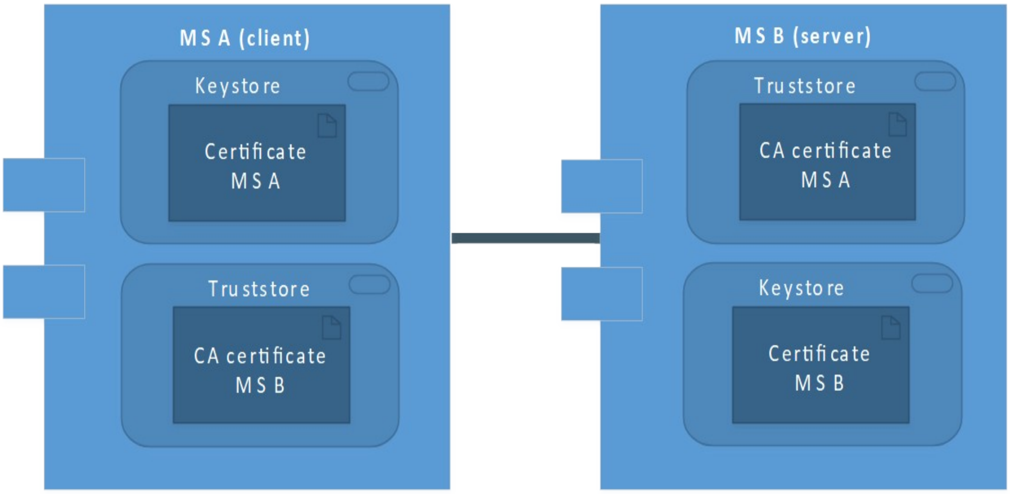
In One-way authentication, there's trust store on the client and a key store on the server containing the private key. This type of authentication is typical for browsing to a server on the internet.

Figure 2. One-way SSL communication



In system-to-system connection the use of mutual TLS is more common.
In Two-way authentication there's a trust store and a key store on both systems.

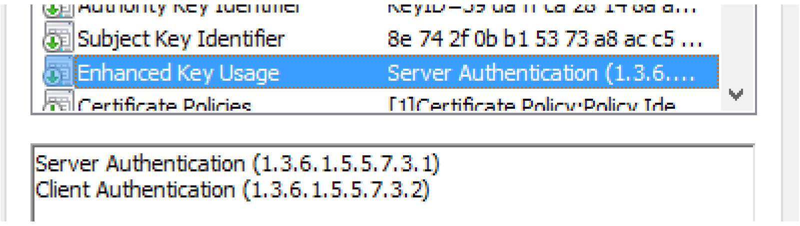
Figure 3. Client certificate authentication



With client certificate authentication, a higher form of security is implemented; the traffic between the systems is encrypted while both systems are authenticated. When the client (in this example the MS A) connects to the server that request client-certificate authentication, the server sends a list with the CA's it has in his trust store and is willing to accept. In the real world this is similar how passports work. You've never met the holder of the passport before but you trust the issuing authority of that passport.

It's possible to use the same certificate for server authentication and client authentication or to use separate certificates for the both features. Participants' exchange infrastructure administrators are free to follow their internal domain policy as long as both Object Identifiers (OID) are present in the certificate when choosing to use a single certificate for both features.
Example*:*
The OID for server certificate is "1.3.6.1.5.5.7.3.1" and for client certificate, it is "1.3.6.1.5.5.7.3.2".

Figure 4. The enhanced key usage showing both OID's



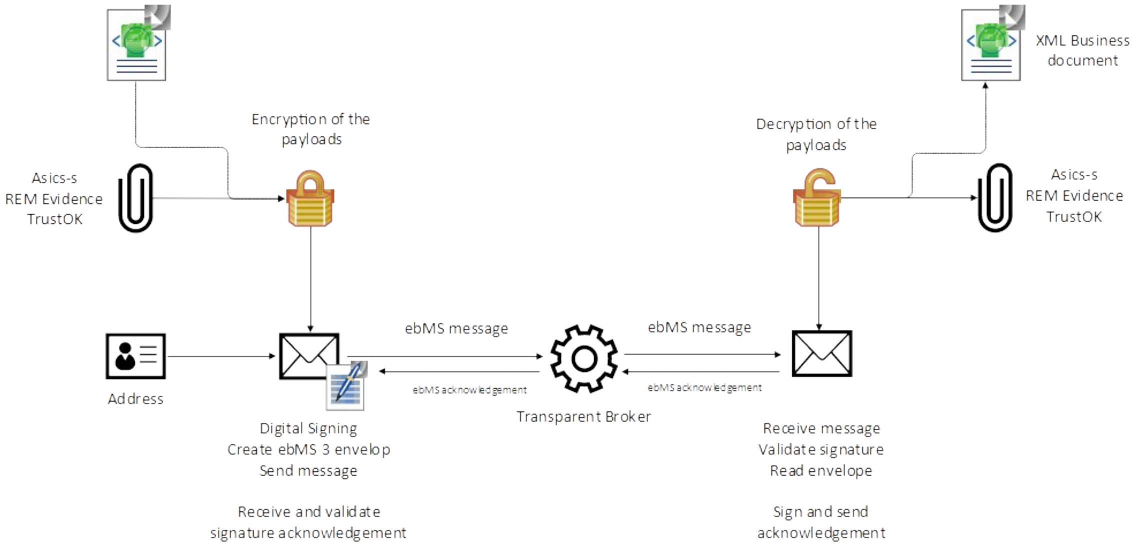
```
Subject Key Identifier      8e 74 2f 0b b1 53 73 a8 ac c5 ...
Enhanced Key Usage          Server Authentication (1.3.6....
Certificate Policies        [1]Certificate Policy:Policy Ide
```

```
Server Authentication (1.3.6.1.5.5.7.3.1)
Client Authentication (1.3.6.1.5.5.7.3.2)
```

4 Message Layer Security – signing and encryption

To keep confidentiality of the message from end-to-end, encryption and/or signing can be used, (TLS is hop-to-hop, chapter 2). The hop does not need to decrypt the message content just to encrypt it again for the next hop. Between hops only routing information needs to be exposed for further dispatch. We can use XML signing and XML Encryption. (Standardized in WS-Security).

Figure 5. The Confidentiality is end-to-end guaranteed



XML Signing

SHA-1 is insufficient for signing, so it is desirable to use an algorithm that is better like SHA-512, SHA-384 or SHA-256

XML Encryption

If payload level XML encryption is used, the FIPS 197 standard (AES) is to be used. AES128 AES256 are the two standards recommended at this point.

5 *Strength of encryption*

The strength of the encryption mechanisms meets the requirements of the time, this means that: the versions and the cryptographic algorithms that are used are open standards and are known to be robust. The key length is large enough to withstand, in the foreseeable future, successful attempts to have the keys retrieved while respecting the importance of the data that it protects.

5.1 Versions

Recent versions of TLS are more secure than older versions. The older TLS versions contain vulnerabilities that cannot be repaired. Use of most recent versions of the TLS protocol is therefore essential. Currently the oldest three versions of TLS (SSL 1.0, SSL 2.0 and SSL 3.0) are not safe to use. The best protection is provided by the most recent version of TLS: TLS 1.2 and TLS 1.3. The e-Justice community should always be up-to-date and adopt newer versions on release.

5.2 Cryptographic selection

For each connection, the client and server use the four cryptographic algorithms match. An algorithm for **key exchange**, an algorithm for digital signatures in the **certificate verification**, a **bulk encryption** algorithm and an algorithm for **hashing**. The four selected cryptographic algorithms are collectively called an algorithm selection.
The two cryptographic algorithms for bulk encryption and hashing are collectively referred to by the term cipher suite, used for protection of records.

The verification of certificates uses digital signatures to ensure the authenticity of the connection To ensure a reliable certificate verification algorithm, the algorithm used to sign a certificate is selected by the certificate supplier (CA). The certificate specifies the digital signature algorithm that becomes its owner during the key exchange used. The RSA-key length should be above 2048 bit, to keep the data save for the next couple of years.

Examples of these algorithms are:

- **Certificate verification**: RSA, ECDSA, etc.
- **Key exchange**: ECDHE, DHE, RSA, etc.
- **Bulk encryption**: AES-GCM, 3DES-CBC,etc.
- **Hashing**: SHA-1, SHA-256, etc.

	Key exchange	Certificate verification	Bulk encryption	Hashing
Good	ECDHE	ECDSA RSA	AES_256_GCM AES_128_GCM	SHA-384 SHA-256
Still Safe	DHE		AES_256_CBC AES_128_CBC	SHA-1

Deprecated	RSA*		3DES-CBC	

* Written as TLS_RSA_WITH_

Some possible cipher suites are:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA384

Deprecated cipher suites are:

- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

6 *Certificates*

6.1 Private key

Generate the private key on the system where the key will remain. A HSM is preferred.
The secret key of the own certificate must be adequately protected. An attacker who obtains this secret key can read or manipulate the intercepted communication traffic. A secret key can be stored in an HSM. An HSM is designed to provide physical protection against 'stealing' a secret key.

6.2 Certificates

There are a number of restrictions on the certificates to be used, below the explanation of which certificates are allowed and which are not.

Don’t use the following FQDN / Common names:

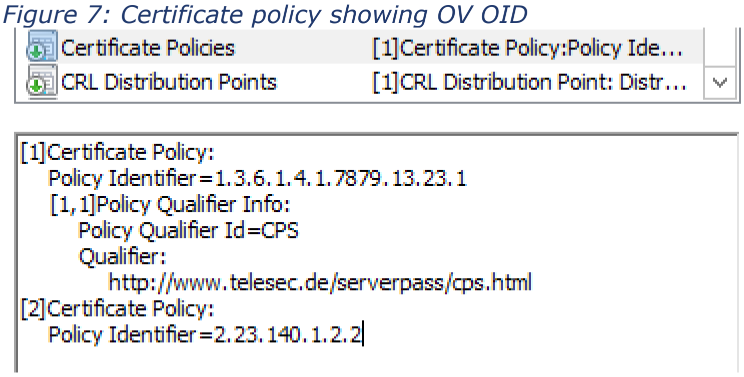
- Blanco
- Incomplete domainnames (e.g. Only hostname)
- IP-addresses as FQDN
- Internationalized Domain Names (IDN)
- The use of wild cards and spaces (e.g. *.bing.com)
  This also applies to the extension Alternative Name (SAN)
- Internal domainnames

The three types of certificates;

- Domain validated (DV). The domain holder of the domain is hereby validated as the applicant for the certificate. And the WHOIS registry is checked.
- Organization Validated (OV). The applicant is hereby validated as the owner of the domain and the organization is checked in the registry (such as the Chamber of Commerce). With this (company) a telephone validation takes place and the WHOIS registry is checked.
- Extended Validated (EV). The same requirements as DV + OV, but the applicant is also validated, for example by signing a form.

The extension "certificate policy" can show you what type of certificate you have.

Figure 7: Certificate policy showing OV OID



Certificate Policies	[1]Certificate Policy:Policy Ide...
CRL Distribution Points	[1]CRL Distribution Point: Distr...

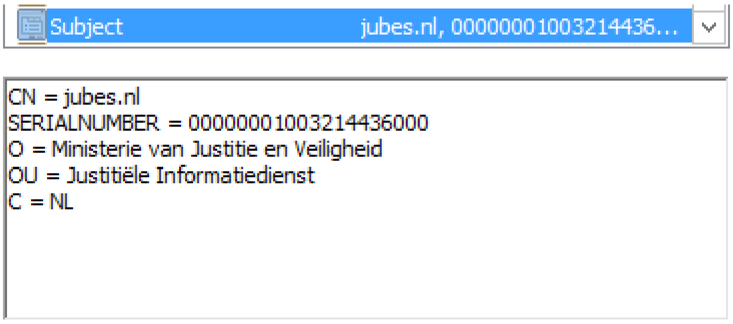
```
[1]Certificate Policy:
    Policy Identifier=1.3.6.1.4.1.7879.13.23.1
    [1,1]Policy Qualifier Info:
        Policy Qualifier Id=CPS
        Qualifier:
            http://www.telesec.de/serverpass/cps.html
[2]Certificate Policy:
    Policy Identifier=2.23.140.1.2.2
```

DV, OV and EV have the following object ID’s (OID)

Type	Policy Identifier
Domain Validated	2.23.140.1.2.1
Organization Validated	2.23.140.1.2.2
Extended Validated	2.23.140.1.1

Some countries don’t accept Domain validated certificates and only Organization Validated, Extended Validated or certifcates that have gone through the same authentication and validation process are accepted. The validation process for these certificates is longer and more extensive than Domain Validated.

Not all CA’s have implemented the use of the Certificate Policy. Another way to recognize the type is to check the organization data in the subject. This extension will display information about the domain name and the registered legal name. Additionally, it can contain the geographical location of the city, state, and country where the company is registered to do business.

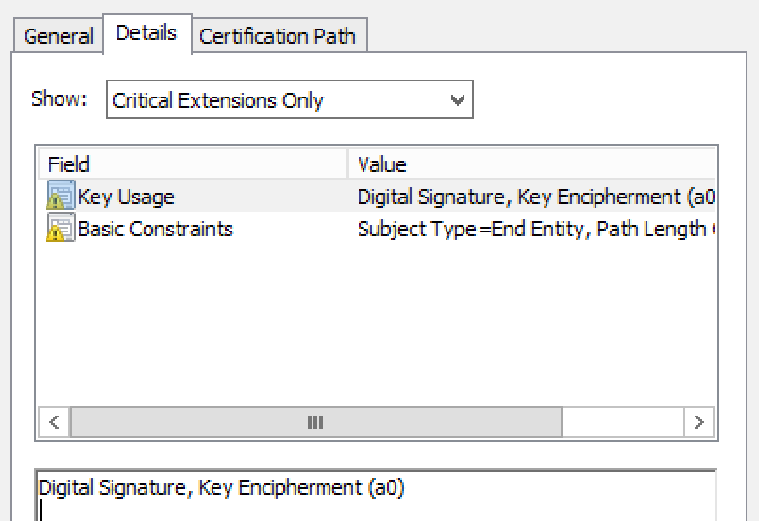


Subject jubes.nl, 0000000100321443 6...

CN = jubes.nl
SERIALNUMBER = 00000001003214436000
O = Ministerie van Justitie en Veiligheid
OU = Justitiële Informatiedienst
C = NL

The extension “Key Usage” defines what a certificate may be used for. It defines the purpose of the public key contained in a certificate. With it you can restrict the operations of the public. For example, if you have a key used only for signing or verifying a signature, enable the digital signature and/or non-repudiation extensions.

Figure 6 The Key Usage Extension



Show: Critical Extensions Only

Field	Value
Key Usage	Digital Signature, Key Encipherment (a0
Basic Constraints	Subject Type=End Entity, Path Length

Digital Signature, Key Encipherment (a0)

In this figure you also see extensions marked as critical (yellow triangle). If an extension is marked as critical the certifcate MUST be used for only that purpose. On the MSH we should only support certificates with the following extensions marked as critical: keyUsage, basicConstraints and policyConstraints. According to RFC5280[2][3]; every certificate-using system MUST reject the certificate if it encounters a critical extension it does not recognize.

[2] https://tools.ietf.org/html/rfc5280
[3] https://www.rfc-editor.org/info/rfc5280

7 Recommendations

The recommendations in this section apply not only to participants PRODUCTION ENVIRONMENTS, but also for the environment in which they perform integration tests with their cross-border partners. E.g. 'TEST ENVIRONMENT', PRE-PRODUCTION ENVIRONMENT', ACCEPTANCE ENVIRONMENT', et cetera.

Security policy and related recommendations should be an on-going concern for the European e-Justice community. Not only during the me-CODEX programme(s), but also after the programme handover to a sustainable service provider. The service provider and the Member States should institutionalise a joint infrastructure user council in order to maintain the appropriate security level over time, meeting both service provider and participants security policy requirements.

At this point this list is not exhaustive and in fact it will never be. Within e-CODEX community we should continually elaborate on which recommendations are applicable. Also distinction between 'recommended' and 'normative' should be allocated.

Subject	**Recommendation**
Application of the recommendations	The security recommendation should apply to all environments that are exposed to external (cross-border) partners.
Security policy interoperability	The common security policy may not infringe national or domain specific security policies.
Transport security	2-way SSL is to be applied to all cross-border e-Justice services
Message security	
Certificates	It should be possible to authenticate client certificates by tracing the chain to the CA
Certificate Authorities	Only certificates that originate from a Certificate Authority listed in the Trusted Services List (TSL) is accepted
Certificate validation	The RSA-key length should be above 2048 bit, to keep the data save for the next couple of years.
FQDN	Don't use the following FQDN / Common names: • Blanco • Incomplete domainnames (e.g. Only hostname) • IP-addresses as FQDN • Internationalized Domain Names (IDN) • The use of wild cards and spaces (e.g. *.bing.com) This also applies to the extension Alternative Name (SAN) • Internal domainnames

certificate policy	Extension should be used to indicate what type of certificate one is using
keyUsage	It must be defined for what usage the certificate is created/deployed. (signing/encryption/etc..) RFC5280
basicConstraint	RFC5280
policyConstraint	RFC5280