

# Consorci Administració Oberta de Catalunya

# Digicert token: instal·lació, inicialització, i signatura

Signatura de codi amb Signador





Realitzat per: Áurea Alcaide

Versió: 1.0 Data: 12/01/2024

Arxiu: DigicertToken.docx



# Índex

1	Objectiu	3
2	Compra i gestió del certificat	4
2.1	Compta de Digicert	
2.2	Certificat actual	5
3	Token criptogràfic	7
3.1	Instal·lació i inicialització	7
3.2	Passwords	7
4	Signatura de codi	8
4.1	Signar Ilibreries java (jarsigner)	8
5	Annexes	
5.1	Signador	10
5.2	Llibreries java (jarsigner)	10
5.3	Instal·lables (install4j v8.0.10)	12



# 1 Objectiu

El Consorci AOC utilitza un certificat de signatura de codi (*Code Signing Certificate*) per signar les llibreries java del Signador.

A partir de l'1 de juny de 2023 a les 00:00 UTC, els estàndards del sector exigeixen que les claus privades dels certificats de signatura de codi estàndard s'emmagatzemin en maquinari certificat com a FIPS 140 Nivell 2, Common Criteria EAL 4+ o equivalent. Aquest canvi reforça la protecció de la clau privada pels certificats de signatura de codi i l'alinea amb la protecció de la clau privada del certificat de signatura de codi EV (*Extended Validation*).

Fins ara fèiem servir certificat en software (p12 o pfx) per la signatura de codi. A partir d'ara cal fer-ho amb claus allotjades en un dispositiu criptogràfic hardware que acompleixi els requeriments anteriors.

El Consorci AOC ha adquirit un certificat de signatura de codi (en endavant CSC de *Code Signing Certificate*) a l'empresa Digicert. L'objectiu d'aquest document és la documentació dels passos que cal seguir per tal d'instal·lar el token criptogràfic, inicialitzar-lo, i un cop fet això, per signar codi amb ell.

#### Dcumentació de Digicert:

https://docs.digicert.com/en/certcentral/manage-certificates/code-signing-certificates.html



# 2 Compra i gestió del certificat

### 2.1 Compta de Digicert

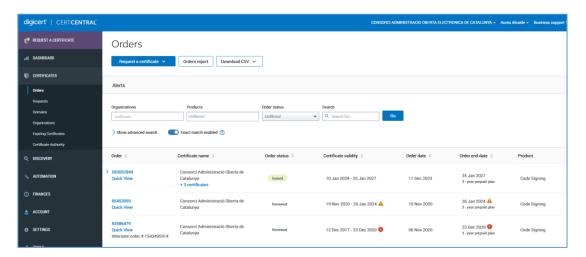
El Consorci AOC disposa d'una compta de Digicert en CERTCENTRAL, que és la web d'administració que permet la gestió tant de les compres com dels certificats un cop emesos.

#### http://www.digicert.com

Usuari: consorciaocat@aoc.cat

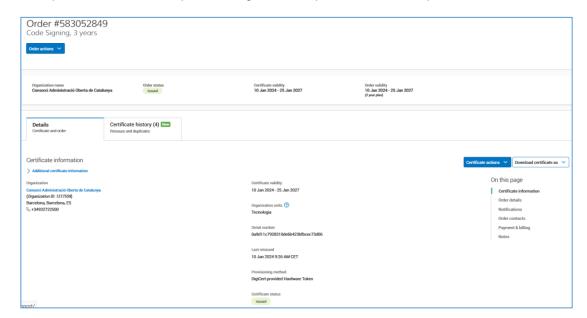
Account manager: Marc Kayumba (<u>marc.kayumba@digicert.com</u>)

Dins de "Certificates" > "Orders", podem veure els certificats contractats, inclòs els ja caducats:



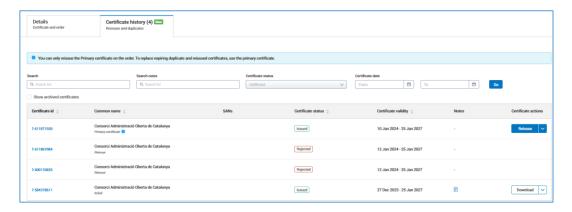
L'últim certificat emès és el primer CSC que hem contractat que segueix la nova normativa, és a dir, en token:

Si cliquem en la ordre, a la pantalla següent ens apareix el certificat que estem fent servir:





A la pestanya "Certificate history" podem veure les diferents reemissions sol·licitades sobre el mateix certificat:



En aquest cas va succeir lo següent:

- 1. 584378611: Vaig sol·licitar renovació del certificat anterior, el que caducava el 26 de gener de 2024, però al formulari vaig seleccionar "Use existint token", perquè desconeixia els canvis en la normativa. En intentar descarregar-me'l vaig veure que no tenia cap dispositiu, i des de Suport de Digicert em van informar de que necessitava un token.
- 2. 600176835: Donat que no tenim cap HSM amb el que fer plug&play al meu portàtil, vaig sol·licitar la reemissió del certificat, en aquest cas demanant que Digicert ens fes arribar un token. Segons em van dir des de Suport, la diferència entre enviament standard i expedited era de només 2 dies. Així que vaig seleccionar standard.
- 3. 611867984: Després de parlar amb Suport i que ja havien passat més de 10 dies i no ens enviaven el token, per fi vaig poder parlar amb el nostre *account manager*, el Marc Kayumba, i vam fer una nova sol·licitud de reemissió, aquest cop especificant enviament *expedited*.
- 4. 611871930: Aleshores quedaven en la pantalla com 2 enviaments pendents, l'standard i el expedited. Així que vaig anul·lar un dels dos, i aleshores van desaparèixer els dos enviaments pendents. Per tant, vam haver de sol·licitar una nova reemissió, amb token, i enviament expedited.
- 5. Un cop fet això, el token va arribar al dia següent (09/gener/2024).

### 2.2 Certificat actual

El certificat actual és aquest:





Des de la pantalla anterior podem gestionar les diferents accions que es poden dur a terme (reemissió, revocació, o descàrrega).



# 3 Token criptogràfic

### 3.1 Instal·lació i inicialització

Connectar el token USB a l'ordinador.

Per tal de procedir a la instal·lació, a la pàgina del certificat a instal·lar, clicar al desplegable "Certificate Actions". En el nostre cas el token està buit, i per tant hem de fer "Install Certificate". En la plana que s'obre, trobarem el codi d'inicialització del token, i un parell de links des d'on descarregar l'instal·lador del hardware i els drivers safenet.

A partir d'aquí, seguir aquestes instruccions:



Set\_Up\_Your\_DigiCert \_Provided\_eToken.pdf

Font: https://knowledge.digicert.com/solution/set-up-your-digicert-provided-etoken

### 3.2 Passwords

El SafeNet eToken utilitza diverses contrasenyes per a l'autenticació. Si una contrasenya d'administrador s'introdueix incorrectament cinc vegades, l'eToken es bloquejarà permanentment.

El SafeNet eToken utilitza els passwords següents:

- Password d'administrador:
   El password d'administrador predeterminat és "0" 48 vegades, tal com la proporciona
   el fabricant. Si es perd "aquest" password, l'eToken es bloquejarà permanentment i
   caldrà comprar-ne un de nou. DigiCert no configura aquest password.
- Password del token:
   Aquest password s'utilitza per accedir al magatzem de certificats de l'eToken. Si es perd, podeu resetejar l'eToken i tornar a instal·lar el certificat.
- Clau de desbloqueig personal (PUK o Personal Unlocking Key): el PUK predeterminat és 000000.

DigiCert no utilitza el PUK en el nostre procés.



# 4 Signatura de codi

Un cop instal·lades les claus al token, ja podem signar. El password que haurem de fer servir és el del token, cada cop que ens ho demani.

### 4.1 Signar Ilibreries java (jarsigner)

Instruccions:



SIGN\_JAVA \_JAR\_FILES\_WITH\_HAR

Font: https://knowledge.digicert.com/tutorials/sign-java-jar-files-with-a-hardware-token-based-code-signing-certificate-in-windows

Bàsicament, lo que hem de fer és:

Anar a la nostra JDK, per exemple: C:\Program Files\Java\jdk1.8.0\_281\bin

I crear un arxiu de nom eToken.cfg, amb aquest contingut:

name=eToken

library=c:\WINDOWS\system32\eTPKCS11.dll

#### Comandes:

Primer anem a la carpeta bin de la JDK:

### cd C:\Program Files\Java\jdk1.8.0\_281\bin

• Amb la següent comanda podem veure els certificats del token:

keytool -list -keystore NONE -storetype PKCS11 -providerclass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg

Introduzca la contraseña del almacén de claves:

Tipo de Almacén de Claves: PKCS11

Proveedor de Almacén de Claves: SunPKCS11-eToken

Su almacén de claves contiene 1 entrada

Consorci Administració Oberta de Catalunya, PrivateKeyEntry, Huella de certificado (SHA-256):

0E:CD:51:6A:48:CB:E4:1F:FE:B7:18:C0:9E:83:6F:61:80:F4:A3:0E:9F:AB:3C:AD:B4:C2:4F:8 A:12:99:65:E5

L'àlies del certificat és:

#### Consorci AdministraciÃ3 Oberta de Catalunya

Quan haguem de fer referència a ell, cal tenir escriure'l tal qual, amb el símbol "estrany".



 Per signar doncs un jar amb el token, afegint segell de temps (en aquest cas timestamp de Digicert):

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\path\to\file.jar" "Consorci Administració Oberta de Catalunya"

• I per comprovar les signatures (signatura i segell d'un jar, per exemple:

jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area\_treball\2024\appletCATCert.jar"

[.....]

- Signed by "CN=Consorci Administració Oberta de Catalunya, OU=Tecnologia, O=Consorci Administració Oberta de Catalunya, L=Barcelona, ST=Barcelona, C=ES"

Digest algorithm: SHA-256

Signature algorithm: SHA256withRSA, 4096-bit key

Timestamped by "CN=DigiCert Timestamp 2023, O="DigiCert, Inc.", C=US" on mié ene 10

08:54:07 UTC 2024

Timestamp digest algorithm: SHA-256

Timestamp signature algorithm: SHA256withRSA, 4096-bit key

Podem veure que té la signatura del Consorci, i un segell de temps de Digicert.



## 5 Annexes

### 5.1 Signador

### 5.2 Llibreries java (jarsigner)

Signatura dels jars del Signador:

=> Anem a la JVM on hem configurat el token (eToken.cfg):

cd C:\Program Files\Java\jdk1.8.0\_281\bin

=> Comprovació certificats en token:

keytool -list -keystore NONE -storetype PKCS11 -providerclass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg

=> Signatura (amb timestamp inclós) dels jars:



jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\appletCATCert.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\apsa-light.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\CATCertCMSlib.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\CATCertPDFlib.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\CATCertXMLlib.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesLinux\_Firefox.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesLinux\_Firefox\_v60.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesMacOS\_Firefox.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesMacOS\_Firefox\_v60.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesWin\_Firefox.jar" "Consorci Administració Oberta de Catalunya"

jarsigner -tsa http://timestamp.digicert.com -verbose -keystore NONE -storetype PKCS11 - providerClass sun.security.pkcs11.SunPKCS11 -providerArg ./eToken.cfg "C:\E\SIGNADOR\area\_treball\2024\librariesWin\_Firefox\_v60.jar" "Consorci Administració Oberta de Catalunya"



#### => Comprovació de la signatura i el timestamp de tots els jars:

```
jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\appletCATCert.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\appsa-light.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\CATCertCMSlib.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\CATCertPDFlib.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\CATCertXMLlib.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\CATCertXMLlib.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesLinux_Firefox.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesMacOS_Firefox.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesMacOS_Firefox_v60.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesWin_Firefox_jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesWin_Firefox_v60.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesWin_Firefox_v60.jar" jarsigner -verify -verbose -certs "C:\E\SIGNADOR\area_treball\2024\librariesWin_Firefox_v60.jar"
```

### 5.3 Instal·lables (install4j v8.0.10)

A install4j cal configurar el token per la signatura dels instal·lables de Windows. La llibreria PKCS#11 és:

c:\WINDOWS\system32\eTPKCS11.dll

Un cop el programa detecta el token, un cop inicialitzat (ens demanarà el password), podem seleccionar al desplegable, el certificat amb què volem signar. En el nostre cas només hi ha un certificat.

