eObčanka veřejná


WARNING
VSTUP NA
VLASTNÍ
RIZIKO

Agenda

- Vlastnosti eObčanky a jak se používá
- Architektura
- Java knihovna k eObčance
- Inspirace
- Otázky

Vlastnosti eObčanky

- Vzdálená identifikace
- Podepisování kvalifikovaným certifikátem
- Obsahuje strojově čitelné údaje
- Obsahuje privátní klíč, kterým lze při znalosti PINu podepisovat.
- Neobsahuje fotografii

Elektronické služby finanční správy ČR

Portál eRecept - elektronická preskripce

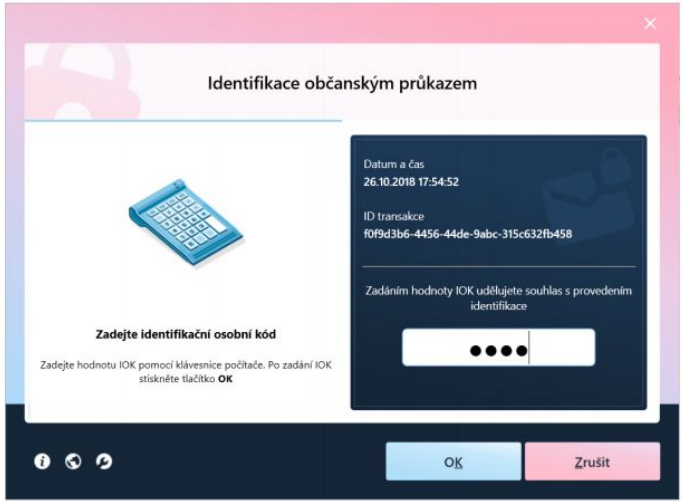
Portál národního bodu

ePortál ČSSZ

Portál občana

Aplikace a PINy

Identifikační klient

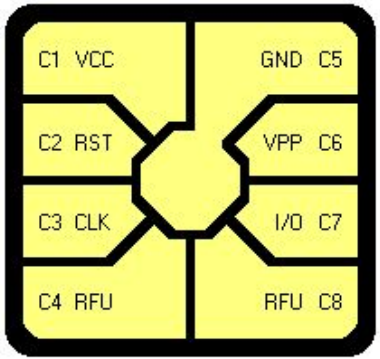


Správce karty

PINy

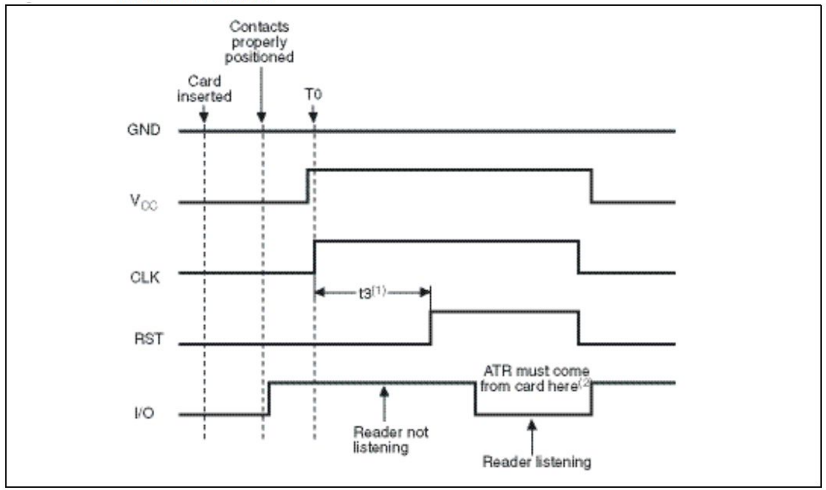
- BOK - Bezpečnostní osobní kód (4-10) pouze fyzick
- IOK - Identifikační osobní kód (4-10 číslic)
- DOK - deblokační osobní kód (4-10 číslic)
- PIN - pin (5-10)
- QPIN (asi od qualified PIN)
- PUK (PIN Unblocking Key) - (8-15 číslic)

Architektura - Čip



- Kontaktní čip M7892 G12 od společnosti Infineon
- 3 Applety:
  - Systémový
  - Identifikační
  - Podepisovací - IAS Classic V4.4 with MOC Server 1.1 on MultiApp V4 (Gemalto)
- Standard ISO 7816, CCID (Chip Card Interface Device)
- Memory 300 kBytes SOLID FLASH™ + 8 kBytes RAM
- CPU Dual 16-bit + šifrovaná RAM
- Symetrické šifrování DES, 3DES, AES up to 256-bit
- Asymetrické šifrování RSA up to 4096-bit, ECC up to 521-bit
- -25 to + 85 °C
- Pravděpodobně má ROCA zranitelnost

Answer to reset



1. t3 = 40 000 clock cycles.
2. ATR must be issued by card between 400 clock cycles and 40 000 clock cycles after RST goes high.

Architektura - APDU

Command APDU						
Header (required)				Body (optional)		
CLA	INS	P1	P2	Lc	Data Field	Le

`CLA` (1 byte): Třída.
`INS` (1 byte): Instrukce
`P1` (1 byte): Parametr 1
`P2` (1 byte): Parametr 2
`Lc` (1 byte): Volitelné pole, počet bajtů v poli `Data`
`Data` pole Toto volitelné pole obsahuje data
`Le` (1 byte): Volitelné. Určuje maximální počet bajtů v odpovědi.

Case 1:
No Command data,
No Response required

CLA	INS	P1	P2

Case 2:
No Command data,
Yes Response required

CLA	INS	P1	P2	Le

Case 3:
Yes Command data,
No Response required

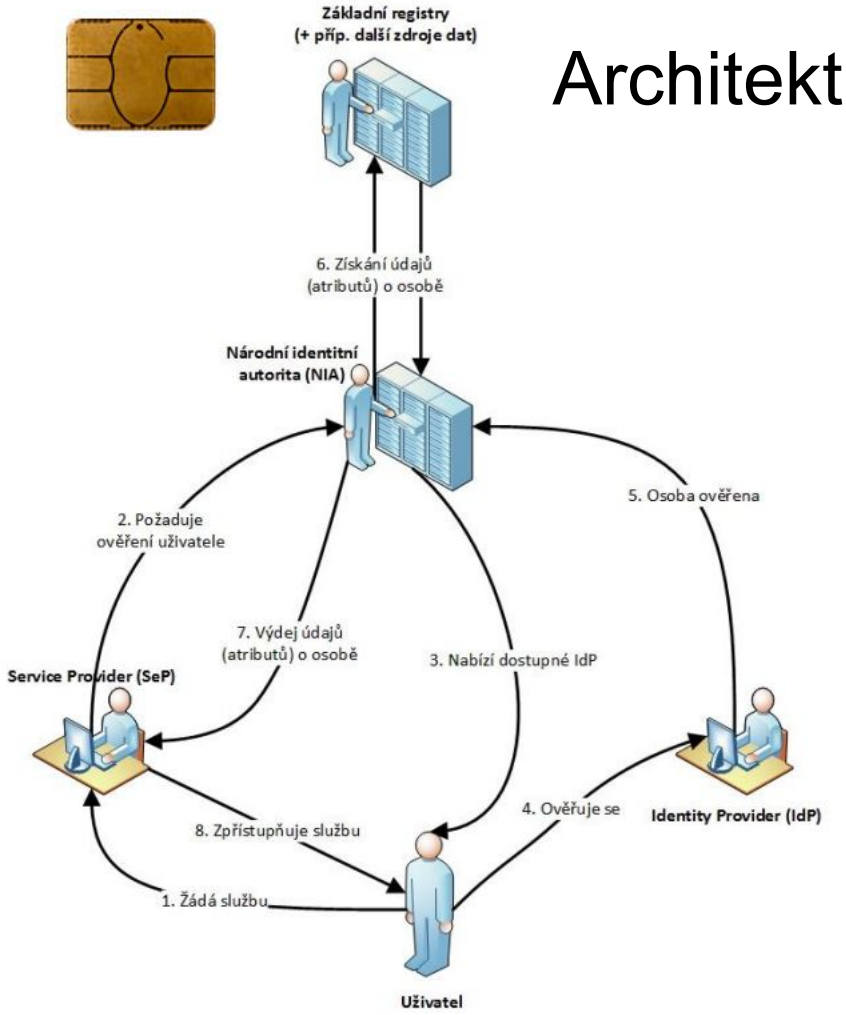
CLA	INS	P1	P2	Lc	Data Field

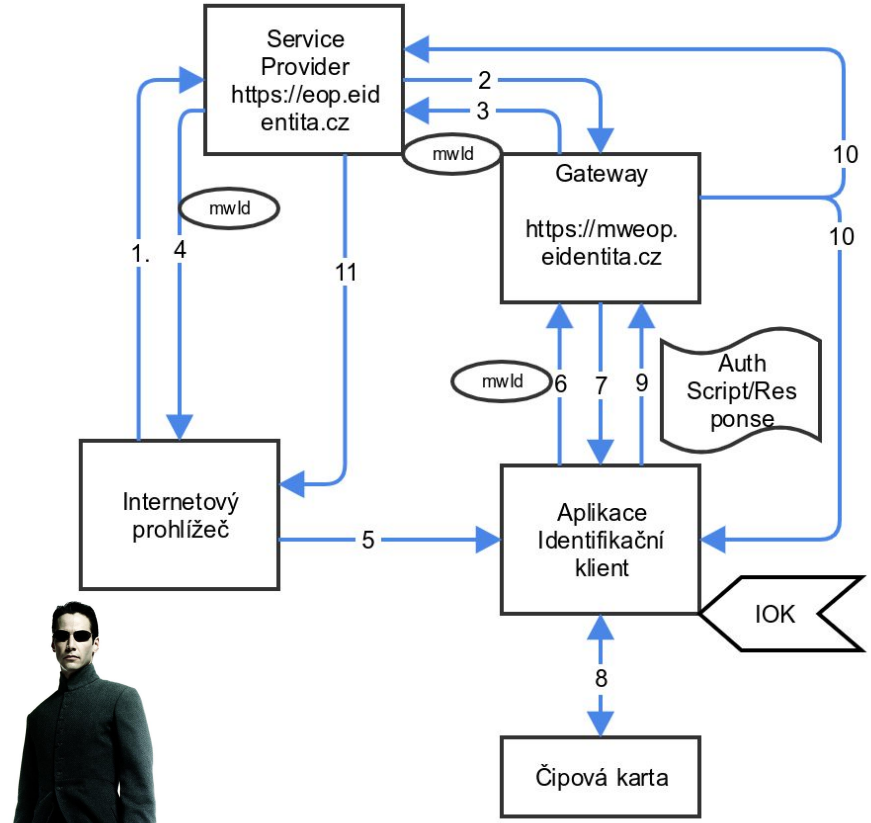
Case 4:
Yes Command data,
Yes Response required

CLA	INS	P1	P2	Lc	Data Field	Le

Response APDU		
Body (optional)	Trailer (required)	
Data Field	SW1	SW2

Architektura - Infrastruktura





Architektura - Skripty

Skript - zadání úlohy

ScriptEnvelope

Script

Label1: Command X, Param Y, Next Label = Label2

Label2: Command X, Param Y, Next Label = Label4

Label3: Command X, Param Y, Next Label = Label4

Script Hash, Signature, KeyId, Algorithm

Příkazy

Open Card

Close Card

Display Message

Send APDU

Goto Label

CmdPIN

Return

Odpověď - odeslání výsledků

Script Response

Label1: Response bytes

Label2: Response bytes

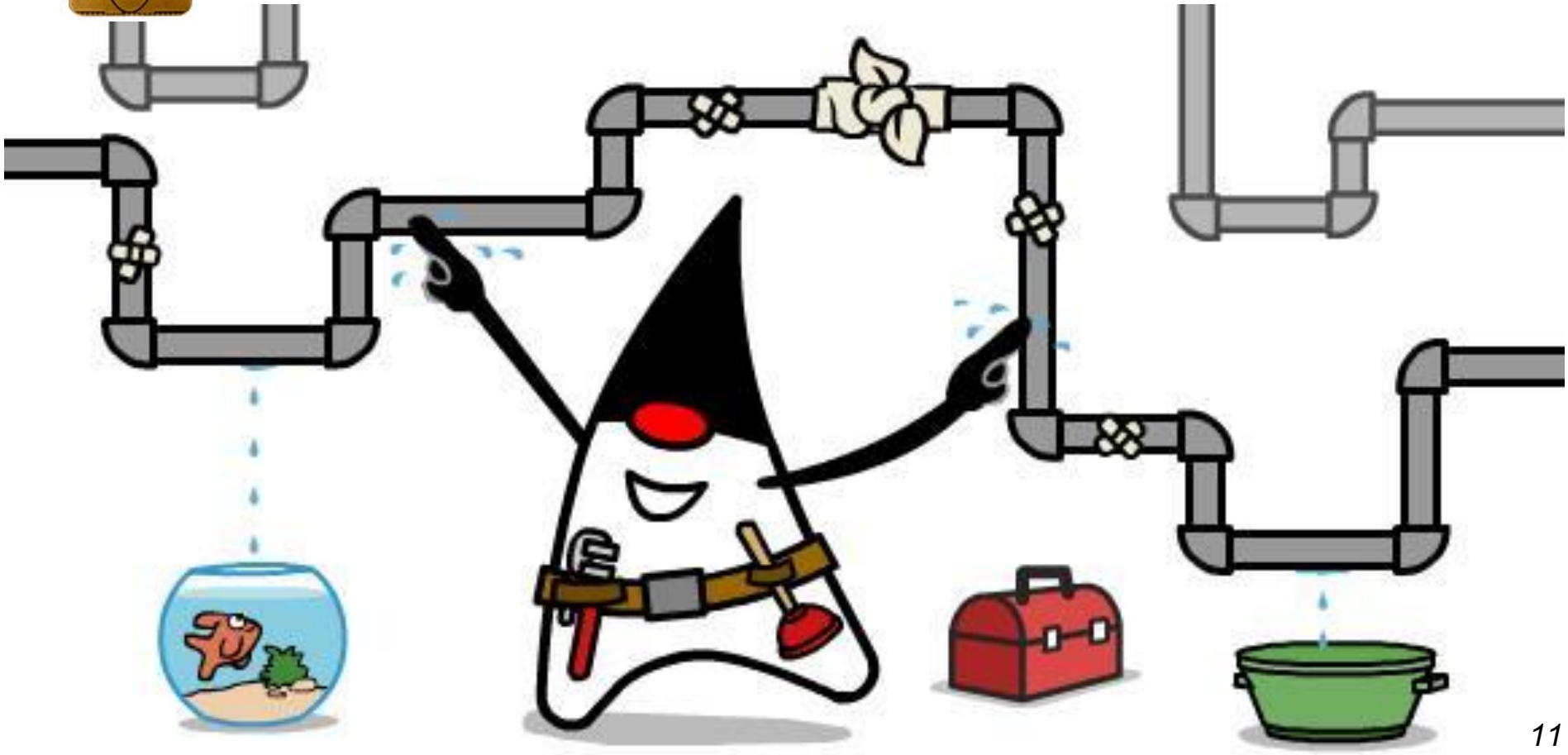
Label3: Response bytes

Hash, Signature, KeyId, Algorithm

Další obsah karty

- Tagy - počet pokusů, sériové čislo certifikátu, číslo klíče
- Identifikátory aplikací
  - 0xD2, 0x03, 0x10, 0x01, 0x00, 0x01, 0x00, 0x02, 0x02 - **Správa karty**
  - 0xD2, 0x03, 0x10, 0x01, 0x00, 0x01, 0x03, 0x02, 0x01, 0x00 - **Identifikace**
- Soubory - certifikáty
  - Krátký - Autorizační - sériové číslo (číslo občanky)
  - Dlouhý - Identifikační - sériové číslo, jméno, příjmení, pohlaví, stav, adresa, rodné číslo
- Secure Messaging

Java knihovna



Java knihovna

`https://github.com/ParalelniPolis/obcanka-public/java`

Repozitář si nejdříve forkněte, usnadní vám to vytváření pull requestů se změnami.

Členění knihovny:

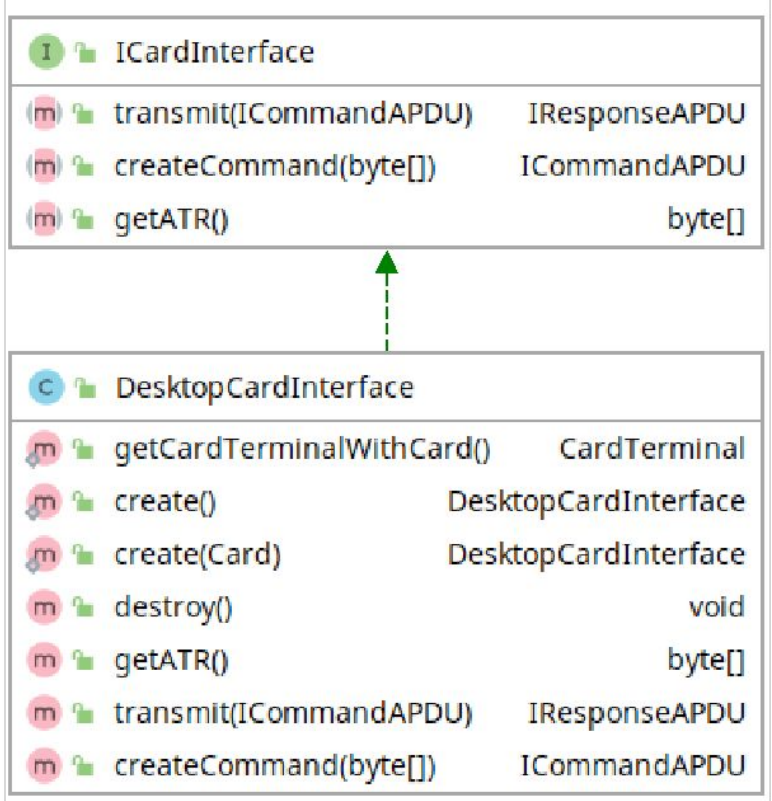
- core - základní funkce
- desktop_lib - funkce specifické pro desktop
- android_lib - funkce specifické pro android
- desktop_app - příkladové desktop aplikace
  - HelloWorld
  - Authenticate
- android_app - príkladová android aplikace
- tools - různé nástroje

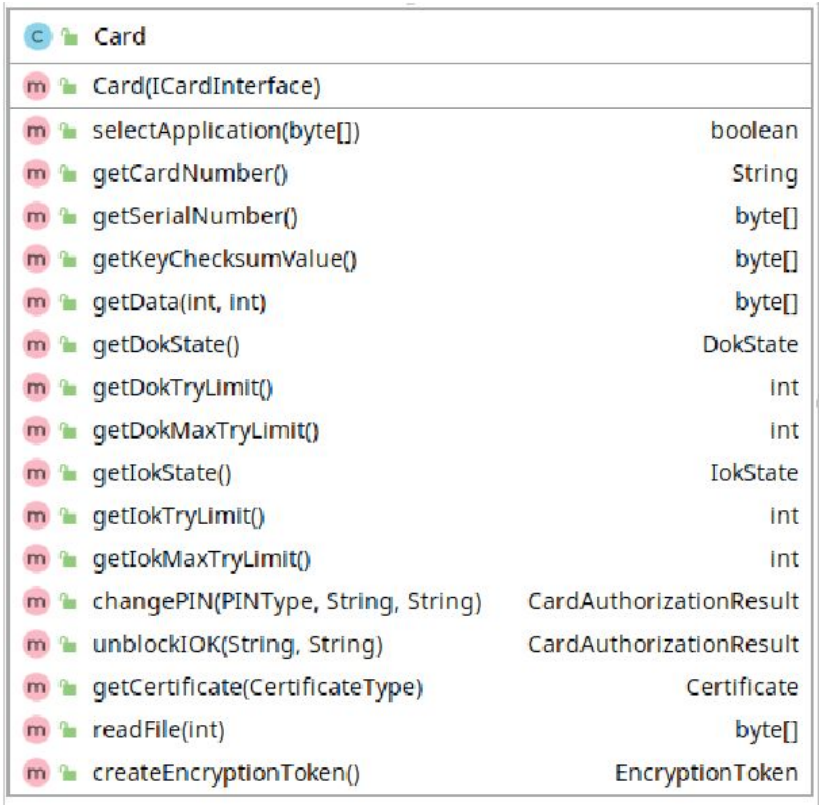
Apache Licence

Java knihovna - příklad

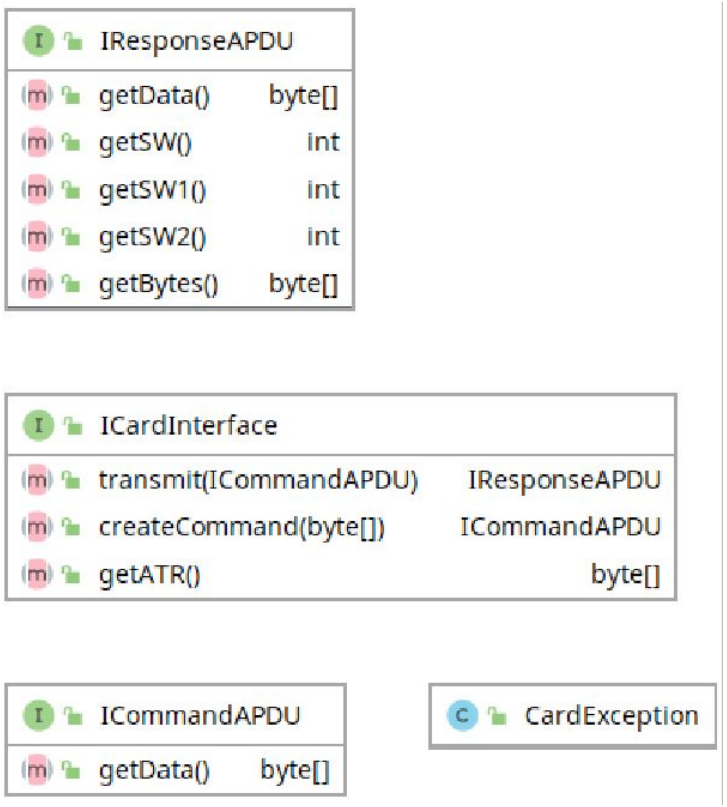
```
public class HelloWorld {
   public static void main(String[] args) {
       DesktopCardInterface ci = DesktopCardInterface.create();
       try {
           downloadCertificates(ci);
       } catch (CardException e) {
           e.printStackTrace();
       }
   }

   private static void downloadCertificates(ICardInterface ci) throws CardException {
       Card cm = new Card(ci);
       String cardID = cm.getCardNumber();
       System.out.println("cardID = " + cardID);
       System.out.println("iokState = " + cm.getIokState());
       System.out.println("cm.getIokTryLimit() = " + cm.getIokTryLimit());

       Certificate longCert = cm.getCertificate(Certificate.CertificateType.IDENTIFICATION);
       Certificate shortCert = cm.getCertificate(Certificate.CertificateType.AUTHORIZATION);
       try {
           FileOutputStream fos = new FileOutputStream("long.crt");
           fos.write(longCert.getData());
           fos.close();
       } catch (IOException e) {
           e.printStackTrace();
       }
   }
}
```

Java knihovna - důležité třídy





Java knihovna - důležité třídy



Inspirace

- Dokumentace
- Knihovny v jiném jazyce
- Podpora pro jiné platformy: iOS, OSX
- Open-source klon existující aplikace
- Zranitelnost
- Podpůrné nástroje: Emulátor karty, Síťová karta
- Infrastruktura: Síťová karta, Proxy, Gateway
- Přihlašování na web
- Javascript WebUSB podpora
- Zařízení používající občanku
- Volební systém