Documentatie Mobiele Authenticatie

# Visie

Via een smartphone zich identificeren met een eID is hedendaags veel te moeilijk. Het idee van een smartphone is juist de mobiliteit, maar als daar voor de identificatie nog een eID-reader bij komt, gaat die mobiliteit grotendeels verloren. Zou het dus niet mogelijk zijn om zonder eID reader te werken, en toch geen veiligheid te verliezen? Als we het mobiele toestel kunnen koppelen aan je identiteit, is het antwoord volmondig ja. Dit document is een studie over de verschillende mogelijkheden om deze koppeling te maken, en zal resulteren in de praktische uitwerking van één van deze mogelijkheden.

# Ontwerpsbeslissingen

## One-time password - Certificaat

De eerste keuze die we moeten maken is hoe we de identificatie juist doorvoeren. Er werd ons voorgesteld om via een one-time paswoord te werken enerzijds, en met een certificaat anderzijds. Het one-time paswoord is dan eigenlijk het bouwen van een software-variant op de apparaten die men onder andere bij home-banking gebruikt. Hiervoor moet je echter cijfers overtypen en met meerdere apparaten werken. Daarom zouden we eerder kiezen om een certificaat te installeren op de smartphone, eventueel in combinatie met een wachtwoord of passphrase om two-factor-authentication te verkrijgen.

## PhoneGap - Apache Cordova

We maken hier de keuze voor Apache Cordova. Dit omdat volgens online bronnen het voornaamste verschil is dat PhoneGap toegang geeft tot meer Adobe Services. Deze verwachten we echter niet nodig te hebben. (Apache Cordova is ontstaan toen het bedrijf achter PhoneGap zijn code doneerde aan Apache, vandaar héél gelijkaardig)

## InAppBrowser - Native Browser

Voorlopig zijn we geneigd om (zeker voor de initiële versie) met een InAppBrowser te werken omdat Cordova dit ondersteunt. Het is niet triviaal gebleken om de certificaten van de native browser aan te passen. De native browser kunnen we eventueel naartoe werken als de InAppBrowser goed werkt. Tenslotte is het wel gebruiksvriendelijker. Anderzijds kan het werken met de InAppBrowser leiden tot een “overheidsapp” waar alles voor de overheid mee kan geregeld worden.

## Unique Code - QR code

Ook hier kiezen we in de ontwerpsfase zeker voor een unieke code. Dit kan ons redelijk wat werkt besparen en toch een bruikbaar (doch onhandig) prototype opleveren. Het overstappen naar QR codes zal ook geen al te grote overstap zijn, en niet veel aanpassing vereisen.

## Device:certificate verhouding

### Verhouding

Apparaten en certificaten kunnen zich op verschillende manieren verhouden. Men kan een 1:1 verhouding gebruiken die per apparaat één certificaat toelaten en vice versa. Men kan ook een N:1 verhouding gebruiken die toelaat per certificaat meerdere apparaten te binden, eenzelfde certificaat kan dan gebruikt worden voor alle apparaten van een gebruiker. Het omgekeerde 1:N is ook mogelijk waarbij één apparaat meerdere certificaten heeft, een apparaat kan dan meerdere gebruikers of types certificaten hebben. Een laatste mogelijkheid is M:N wat toelaat per apparaat verschillende certificaten te hebben en ook toelaat een certificaat voor meerdere apparaten te gebruiken.

### Factoren

De voor- en nadelen van deze verhoudingen kunnen we aan de hand van enkele factoren vergelijken.

Transparantie naar de gebruiker toe: het concept van een certificaat en certificate authority is complex voor niet technische mensen dus de mate waarin uitleg nodig is om het systeem bruikbaar te maken voor de klant is van belang.

Meerdere devices per certificaat: dit laat toe dat voor meerdere apparaten hetzelfde certificaat kan gebruikt worden. Heeft als gevolg dat private keys moeten kunnen verwijderd worden van apparaten of doorgegeven worden naar andere apparaten wat vrij moeilijk kan zijn door security redenen. Anderzijds moet niet per apparaat een certificaat gegenereerd worden.

Meerdere certificaten per device mogelijk: meerdere users kunnen gebruik maken van eenzelfde apparaat om in te loggen op overheids websites. Het is ook mogelijk met meerdere types certificaten voor eenzelfde gebruiker te werken (al vereist dit meer kennis van certificaten door de gebruiker).

Indien kwijt/kapot/gestolen: wat moet een gebruiker doen indien zijn apparaat kapot is, gestolen werd of werd kwijt gespeeld. Ingewikkelde procedures zijn niet gebruiksvriendelijk.

Complexiteit van implementatie: is de complexiteit verantwoord in vergelijking met de toegevoegde functionaliteit.

### Vergelijking

1:1

+ Transparantie: gebruiker moet niet weten wat een certificaat is. Het volstaat dat de gebruiker weet dat hij zijn apparaat moet koppelen aan zijn account bij de overheid.

+ Kwijt/kapot/gestolen: één certificaat moet teruggetrokken worden wat overeenkomt met de gebruiker die het apparaat verwijdert uit de lijst van apparaten

- Geen meerdere devices per certificaat

- Geen meerdere certificaten per device: dus een apparaat kan niet door meerdere mensen gebruikt worden.

Complexiteit - functionaliteit

Dit geval is het eenvoudigst. Doordat er geen meerdere devices per certificaat zijn, moet het niet mogelijk zijn private keys te verplaatsen wat de implementatie sterk vereenvoudigd, ook moet er maximaal één certificaat teruggetrokken worden wanneer een apparaat verdwijnt. Dit is ten koste van functionaliteit aangezien er geen meerdere devices per certificaat of omgekeerd mogelijk zijn. Het systeem is ook makkelijk te begrijpen voor gebruikers.

1:N

+ Transparantie: het volstaat dat gebruikers weten dat ze ieder apparaat die ze hebben/gebruiken moeten koppelen aan hun account bij de overheid.

+ Meerdere certificaten per device mogelijk.

- Geen meerdere devices per certificaat.

- Kwijt/kapot/gestolen: alle certificaten die op het apparaat stonden moeten teruggetrokken worden (dus N in plaats van 1).

Complexiteit - functionaliteit

Doordat er geen meerdere devices per certificaat zijn, moet het niet mogelijk zijn private keys te verplaatsen wat de implementatie sterk vereenvoudigd. Anderzijds moet bij een kwijtgespeeld/kapot/gestolen apparaat wel meerdere certificaten ingetrokken worden. De complexiteit neemt dus toe maar blijft relatief eenvoudig terwijl de functionaliteit en daaruit volgend gebruiksgemak toeneemt. Het systeem blijft relatief gemakkelijk te verstaan voor de gebruiker tenzij we meerdere types certificaten toelaten (door dezelfde gebruiker).

N:1

+ Meerdere devices per certificaat mogelijk.

+ Kapot: de sleutel blijft bruikbaar.

- Geen meerdere certificaten per device.

- Kwijt/gestolen: één certificaat moet teruggetrokken worden maar wel op N apparaten vervangen worden.

- Transparantie: het wordt moeilijk om uit te leggen dat men voor alle apparaten de procedure opnieuw moeten doen als één van hun toestellen gestolen of kwijtgespeeld wordt. Het is ook niet gebruiksmakkelijk indien dit voorkomt.

Complexiteit - functionaliteit

De complexiteit stijgt sterk doordat het moet mogelijk zijn private keys te verplaatsen wat niet evident is omwille van security redenen. De transparantie naar de gebruiker toe neemt ook af. De functionaliteit stijgt.

M:N

+ Meerdere devices per certificaat mogelijk.

+ Meerdere certificaten per device mogelijk

+ Kapot: de sleutel blijft bruikbaar.

- Kwijt/gestolen: N certificaten moet teruggetrokken worden en ook op overeenkomstige N\*M apparaten vervangen worden.

- Transparantie: De gebruiksmakkelijkheid daalt sterk doordat een gestolen/kwijtgespeeld apparaat een groot effect heeft. Dit uitleggen aan de gebruiker wordt héél wat moeilijker. Dit wordt nog moeilijker als we ook met meerdere types certificaten werken.

Complexiteit - functionaliteit

De complexiteit is hier het grootst, het moet mogelijk zijn private keys te verplaatsen en bij kwijtspelen/gestolen apparaat is er een groter effect. De functionaliteit laat echter zowat alles toe. De transparantie naar gebruikers toe daalt ook het meest.

### Uiteindelijke keuze

We kiezen voor 1:N en specifieker kiezen we er voor dat meerdere certificaten afkomstig zijn van meerdere users (N certificaten, N gebruikers) en dat er geen meerdere types van certificaten zijn. Enerzijds omdat de transparantie naar de gebruiker toe daalt en anderzijds omdat er momenteel nog geen toepassing is voor meerdere types van certificaten voor eenzelfde gebruiker.

De redenen achter de keuze

Enerzijds laat het het gebruikerscomfort toe dat een apparaat door meerdere mensen kan gebruikt worden voor dezelfde applicatie (wat wel vaker gebeurt bij tablets bijvoorbeeld) en tegelijk zorgt het niet voor een moeilijke uitleg over wat een certificaat is, het blijft dus transparant. Ook is het dan niet nodig om een implementatie te voorzien voor het verwijderen of doorgeven van een private key op een smartphone en volstaat een withdrawal van het certificaat. De enige echte nadelen zijn dat bij het kapot gaan of kwijtspelen of gestolen worden van het apparaat er N certificaten moeten teruggetrokken worden (zonder weliswaar vervangingen op andere apparaten) en dat er geen meerdere devices per certificaat mogelijk zijn. In principe is dit laatste echter geen nadeel. Ten eerste omdat een klant nog steeds per apparaat het proces zal moeten uitvoeren dat zijn apparaat bindt aan zijn account. Dat in de achtergrond

geen nieuw certificaat wordt aangemaakt maar een bestaand certificaat gebruikt wordt zal de gebruiker een worst wezen en ten tweede is het zelfs zo dat dit eigenlijk enkel maar voor nadelen zorgt want dit zorgt voor een complexere uitleg naar de klant toe (transparantie)

en dat de private key moet kunnen verwijderd of doorgegeven worden (complexere implementatie) en dat indien het apparaat kwijt gespeeld wordt of gestolen wordt, er veel vervangingen nodig zijn op andere apparaten (die er niets mee te maken hebben in principe, leg dat maar eens uit aan de klant)

# Problemen

* Inloggen op de webapplicatie

# Technologiëen

Draw.io: voor alle figuren (UML, mockup, BPMN,...)

Ruby On Rails: de server

Apache Cordova: voor het ontwikkelen van de app op meerdere platformen

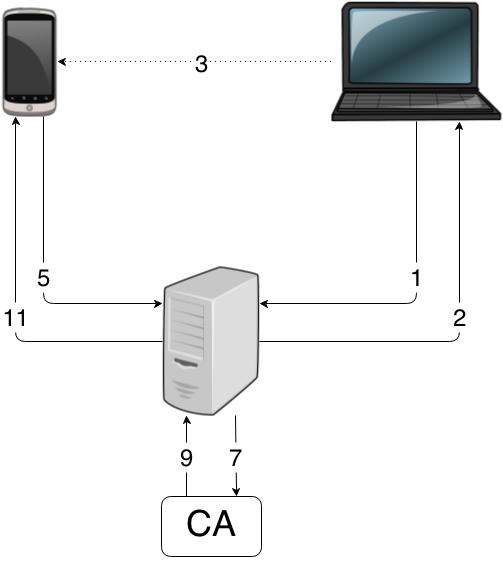
Virtual Box: alles zal gedraaid worden in Ubuntu (laatste versie)

Jenkins: managen van builds

Git/Mercurial: versioning

# Ontwerp

## Algemeen verloop

1. De gebruiker surft naar de webserver en logt in met zijn eID.
2. De webserver geeft de gebruiker een lijst van instructies, tesamen met een unieke code of QR code.
3. De gebruiker installeert en start de applicatie, geeft zijn unieke code in of scant de QR code met zijn smartphone.
4. De smartphone genereert zijn sleutelpaar.
5. De smartphone stuurt een CSR voor zijn sleutelpaar naar de server, tesamen met de unieke code of QR code.
6. De server legt de verbinding tussen de unieke code en het account van de gebruiker en zijn smartphone.
7. De server stuurt de CSR van de smartphone door naar de CA om te tekenen.
8. De CA tekent de CSR.
9. De CA stuurt het certificaat terug naar de server.
10. De server brengt de ingelogde gebruiker op de hoogte dat zijn certificaat getekend werd.
11. De server stuurt het getekende certificaat door naar de smartphone.
12. De smartphone slaat het certificaat op voor gebruik.

## Domain Model

## Op de server

#### Level 1

## Domain Model Server v2(3).png

* Een users stelt een burger voor met een account op eGov. Een user heeft geen, één of meerdere devices.
* Een device stelt een smartphone/tablet/… voor en kan door meerdere users gebruikt worden.
* Een certificate is een certificaat die ondertekend moet worden door een CA en die opgeslagen zal worden op het device. Dit certificaat moet bewijzen dat het device van een bepaalde gebruiker of meerdere gebruikers is. Een device kan meerdere certificaten hebben die afkomstig zijn van allemaal verschillende gebruikers.
* De eID wordt door de user gebruikt om te bewijzen dat hij of zij het device gebruikt of eigenaar is. Een user kan maar één eID hebben en omgekeerd.
* De website server is de website van de overheid waar de gebruiker op in logt met gebruik van de eID. Deze server genereert de unieke code die de gebruiker op zijn device moet invoeren. Indien dat de code correct wordt terug gegeven door het device zal de server de meegegeven (met de code) CSR doorsturen naar de CA. Deze ondertekent de CSR en stuurt die terug naar de server die ze doorstuurt naar het device.
* De unique code is de unieke code die de server genereert wanneer een user een device wil toevoegen. Deze wordt getoond aan de user die ze dan in zijn device invoert. Eenmaal de ingevoerde code verstuurd wordt naar de server waar controle gebeurt, stopt de code met bestaan en heeft ze geen betekenis meer.
* De CSR is een certificate signature request die het device genereert. Dit wordt naar de server gestuurd die ze dan naar de CA stuurt. De CA ondertekent deze en laat ze dan terugsturen.
* De CA is de certificate authority die certificates ondertekent of dus de CSR inwilligt.

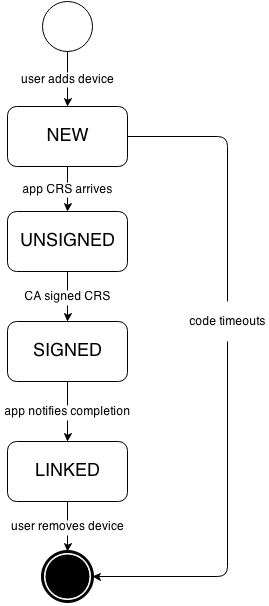
#### Level 2

TODO

### Op de smartphone

Hier krijg je enkel een certificaat (signed/unsigned) als concept.

## State diagram device



## Risicoanalyse

Zie externe spreadsheet.

## Use Cases

### User signs in

TODO: how?

### User sign in with certificate

TODO

### User adds device

**Beschrijving:** De gebruiker voegt een nieuw apparaat toe om te linken.

**Actoren:** De gebruiker, ht apparaat, de webserver, de CA.

**Precondities:**

* De gebruiker is ingelogd.

**Main Flow:**

1. De gebruiker vraagt om een nieuw apparaat toe te voegen.
2. De web server vraagt naar een naam voor het apparaat.
3. De gebruiker voorziet deze gegevens.
4. De webserver genereert een unieke code en geeft deze aan de gebruiker. De status van het apparaat wordt “pending”.
5. De gebruiker installeert de app op zijn apparaat.
6. De gebruiker geeft de unieke code in in het (fysieke) apparaat.
7. Het apparaat stelt een CSR op en stuurt dit met de code naar de webserver.
8. De web server zoekt de gebruiker via de code en zet de status van het apparaat op “unsigned”. Hij stuurt de CSR naar de CA. De code wordt verwijderd.
9. De CA tekent het CSR en stuurt het certificaat naar de webserver.
10. De web server zet de status van het apparaat op “signed” en stuurt het certificaat door naar het fysieke apparaat.
11. Het apparaat slaat het certificaat op en meldt het succes aan de webserver.
12. Op de webserver wordt het apparaat die gekoppeld werd in de lijst van apparaten getoond.

**Postcondities:**

* De gebruiker is ingelogd.
* Er is een nieuw apparaat gekoppeld aan het account van de gebruiker.

**Alternate Flows:**

* Overal: De gebruiker annuleert het aanmaken.
  1. De web server geeft de startpagina terug weer zonder het apparaat toe te voegen aan de lijst van apparaten.
* 3A: De gebruiker kiest de naam van een eerder aangemaakt apparaat.
  1. De web server weigert deze naam met een waarschuwing.
  2. Ga naar main flow stap 2.
* Na stap 4: De tijdslimiet op de code loopt af.
  1. De web server vergeet de code en notificeert dit, de webserver geeft de startpagina terug weer zonder het apparaat toe te voegen aan de lijst van apparaten.
* 7A: Het apparaat heeft geen verbinding met de webserver.
  1. Het apparaat vraagt de gebruiker om verbinding mogelijk te maken.
  2. Keer terug naar stap 6.
* 7A: De ingegeven code is niet geldig.
  1. De web server brengt het apparaat hiervan op de hoogte.
  2. Het apparaat brengt de gebruiker hiervan op de hoogte en waarschuwt dat de tijdslimiet mogelijk overschreden is.
  3. Keer terug naar stap 6.

### User removes device

**Beschrijving:** De gebruiker verwijdert de koppeling tussen account en apparaat.

**Actoren:** De gebruiker, de webserver, de CA.

**Precondities:**

* De gebruiker is ingelogd.
* De gebruiker heeft een gekoppeld apparaat.

**Main Flow:**

1. De gebruiker vraagt om het apparaat te verwijderen.
2. De web server vraagt aan de CA om het certificaat te ontkrachten.
3. De CA herroept het certificaat en meld dit aan de webserver.
4. De web server verwijdert het apparaat uit de lijst.

**Postcondities:**

* Het apparaat is verwijderd uit de lijst van apparaten.
* Het certificaat van het verwijderde apparaat is herroepen.

## Wireframes

Zie presentaties in `wireframes/mockups`.