# OPENSIGN PROJECT - Concept

This document describes functionality that the code signing server provides. Section one gives an overview of entities and their interaction with the server. Section two pictures use cases. Following two sections describe the user interface from the web portal as well as the client side application.

## 1. Entities and Interaction



## 2. Use Cases

### 2.1 User Registration

For accessing the certification service a user has to go through the registration process. Starting point of this process is on the web-portal the "Register"- site. A new user has to enter some personal details and information of projects he was working on in the past. Note that this information is forwarded to the approver in the next step.

### 2.2 SPC Generation

Prerequisite to this step is that the user has entered some personal details and has downloaded the client application.

First, the user needs to request the commitment of an approver. This is done by selecting the desired approver form a drop-down list which highlights all approvers known to the system. This request will notify the approver, who will either accept or reject the request. In case of rejection the requesting user will not be able to generate a SPC. The only option would be to ask another approver. In case of acceptance the requesting user will be notified that he may generate a code signing certificate.

Generation of a SPC is triggered by the client application, which establishes a SSL tunnel to the server with server- and client-authentication.

The user has to enter his login-details which were set in step (2.1). As a response the required user-details for certificate generation are load from the server. After that the user can explicitly generate a signing key, which is stored in the local key store. Note that this key will be protected with the same password and username the user has entered before. This key is used for generating the PKCS#10 request. This request is posted to the server. In turn the certificate will be available at client side.

### 2.3 Code Signing

A dialog in the client application allows the user to select the code package he wants to sign. As the client application knows the currently valid certificate already no additional certificate selection is necessary. The user simply inputs his login-data, which will decrypt the signing key. After that the signed package, optionally with embedded certificate, will be created.

If the certificate has expired the user is requested to generate a new certificate before the signing operation.

### 2.4 Code Verifying

Code can be verified by making use of the client tool. The tool allows the selection of the signed code package in order to start the verification process. First, the attached signature of the code is verified my usage of the embedded certificate. In the second step the certificate is validated. This includes the signature, expiry date and if it has not been revoked. Additional information required for verification like certificate chain and revocation lists are loaded from the server on demand.

### 2.5 Certificate Revocation

Certificates may be revoked any time by the developer who owns the certificate and by the approver who is responsible for the developer. Other, also anonymous, users may access a form online to post a revocation request to the approver. The request includes the certificate serial number and a description of the reason why this certificate should be revoked. The approver has to validate this request and has to execute appropriate measures.

## 3. Web portal

Following sub-sections describe pages on the web portal, which are accessible to users determined by the authorisation level they have.

### 3.1 Home

This page holds a short summery of the functionality the server provides. Furthermore, a login-field for entering username and password is on this page.

### 3.2 User Profile (private)

On this site a user can maintain his personal details, which are only accessible by him and his approver and partly shown on the public profile.

**User Details:** This block contains user name, real name, date of birth and address of the user.

**Short CV:** Short introduction about the history of this developer.

**Project Involved:** A list of links and short description of projects the developer is contributing to.

### 3.3 Certificate Management

**Select Approver:** A drop down menu shows all assigned approvers. This field is only possible to change before the approver has accepted the request.

**Public Key List:** A list of all public keys a user has. Note that the user only may own one valid public key at a time - the others are revoked or have expired.

### 3.3 User Profile (public)

**Username:** This is the name the developer has chosen, which is unique in the system.

**Approver:** Link to the approver's public profile.

**Public Key:** Currently valid public key

### 3.4 Associated Members

This page is only accessible by the approver who is responsible for this group of developers. Mainly this page contains a list showing the most important values of the corresponding developer and a button whether to approve the developer or not. In case of approval the developer may generate a code signing certificate. If the approver cancels or denies the approval the developers code signing certificate is revoked and the developer is not able any more to have a new one generated.

### 3.5 Administration

The administrator maintains all data and acts as root of all approvers.

## 4. Client Side Application

TO BE CONTINUED

### 4.2 Generation of SPC

- Login at Code Signing Server

- Key generation

- PKCS#10 request generation

- Installation of SPC

### 4.2 Signing of Software

- Login

- Drag & Drop?

### 4.3 Verification of Software

- Drag & Drop?