



## Zentrum für sichere Informationstechnologie – Austria Secure Information Technology Center – Austria

A-1030 Wien, Seidlgasse 22 / 9  
Tel.: (+43 1) 503 19 63-0  
Fax: (+43 1) 503 19 63-66

A-8010 Graz, Inffeldgasse 16a  
Tel.: (+43 316) 873-5514  
Fax: (+43 316) 873-5520

DVR: 1035461

<http://www.a-sit.at>  
E-Mail: [office@a-sit.at](mailto:office@a-sit.at)  
ZVR: 948166612

UID: ATU60778947

# PDF-OVER 4.0 ANFORDERUNGSDOKUMENT VERSION 0.5, 19. JUNI 2012

**Andreas Fitzek** - [andreas.fitzek@iaik.tugraz.at](mailto:andreas.fitzek@iaik.tugraz.at)  
**Tobias Kellner** - [tobias.kellner@iaik.tugraz.at](mailto:tobias.kellner@iaik.tugraz.at)  
**Vesna Krnjic** - [vesna.krnjic@iaik.tugraz.at](mailto:vesna.krnjic@iaik.tugraz.at)

**Zusammenfassung:** PDF-OVER ist ein plattformunabhängiges Werkzeug zum elektronischen Signieren von PDF-Dokumenten mit der österreichischen Bürgerkarte. Ein Usability Experten Review der Software ergab einige Mängel im Bereich der Benutzerfreundlichkeit. Es stellte sich heraus, dass die Benutzerinnen bzw. Benutzer nicht ausreichend durch den Signaturvorgang geführt werden und daher den Signaturprozess häufig vorzeitig abbrechen. Die Neuentwicklung der Software soll zum Ziel haben, sowohl für die Durchschnittsnutzerin und den Durchschnittsnutzer einfach, verständlich und gut bedienbar zu sein, als auch effizient für den Power-User. Folglich soll PDF-OVER 4.0 mit einer durchgängigen Benutzerinnen- und Benutzerführung und mit konfigurierbaren Voreinstellungsmöglichkeiten für wiederkehrende Schritte neu entwickelt werden. Bei der Umsetzung soll besonderes Augenmerk auf eine Entkoppelung von Benutzerschnittstelle und Dokumentenverarbeitung gelegt werden. Damit wird auch eine Vorbereitung der Schnittstellen auf Erweiterung wie z.B. eine Integration des PAdES-Standards erreicht. PDF-OVER 4.0 ist als Java-Anwendung geplant, die plattformunabhängig unter Windows, Mac OS und Linux verfügbar ist.

## Inhaltsverzeichnis

Inhaltsverzeichnis	1
Abbildungsverzeichnis	2
1. Einleitung	2
2. Projektziele	3
2.1. Zielsetzung	3
2.2. Prozessbeschreibung	4
2.2.1. Auswahl des Dokuments	5
2.2.2. Positionierung des Signaturblocks	6
2.2.3. Signieren des Dokuments	7
3. Systemkomponenten	9
3.1. PDF-Signatur (Middleware)	9
3.2. GUI	9
3.3. Externe Komponenten	9
4. Arbeitspakete	10
4.1. Spezifikation	10
4.2. Design	10
4.3. Implementierung	11
4.4. Dokumentation	11
4.5. Testdesign und Testen	11
4.6. Inbetriebnahme der Software	11
5. Aufwand	12
Revisionsverlauf	12

# Abbildungsverzeichnis

Abbildung 1 - Prozessbeschreibung .....	4
Abbildung 2 - Entwurf der Benutzeroberfläche (Schritt 1) .....	5
Abbildung 3 - Entwurf der Benutzeroberfläche (Schritt 2).....	6
Abbildung 4 - Entwurf der Benutzeroberfläche (Schritt 3).....	7
Abbildung 5 - Abschlussdialog.....	8
Abbildung 6 - Softwarearchitektur.....	9

## 1. Einleitung

Das Signieren von PDF-Dokumenten hat sich im E-Government-Bereich als eine wichtige Anwendung etabliert, beispielsweise bei der Kommunikation zwischen Behörde und Bürgerin und Bürger. Behörden und Parteien haben die Möglichkeit, PDF-Dokumente mit einer elektronischen (Amts-) Signatur zu versehen, und somit offizielle Dokumente, wie etwa Bescheide, auf elektronischem Weg austauschen. Zielgruppen sind somit sowohl Bürgerinnen und Bürger als auch Behörden und Unternehmen.

Im Rahmen dieses Projekts soll die schon vorhandene Applikation PDF-OVER neu entwickelt werden. Der Funktionsumfang der aktuellen Version soll bis auf die Möglichkeit einer textuellen Signatur und der Freitexteingabe übernommen und um weitere Kernkomponenten erweitert werden. Es stellte sich heraus, dass die textuelle Signatur und die Freitexteingabe kaum genutzt werden, jedoch zusätzliche Komplexität generieren, die der Benutzerfreundlichkeit des Gesamtprozesses abträglich ist. Beabsichtigt ist die Entwicklung eines vereinfachten Signaturprozesses, der gut verständlich ist und effizient abgewickelt werden kann.

Für eine effiziente elektronische PDF-Signatur bietet sich eine plattformunabhängige Applikation an, die einerseits ein einfaches Benutzerinterface hat und eine unerfahrene Benutzerin bzw. einen unerfahrenen Benutzer durch den gesamten Prozess führt, mit ausreichendem Feedback über den aktuellen Zustand. Andererseits besteht die Notwendigkeit, wiederkehrende Schritte im Ablauf zu automatisieren, damit Power-User mit weniger Interaktion das gewünschte Ergebnis erhalten. Entsprechend diesen Anforderungen beinhaltet das neue Konzept eine einfache, klar strukturierte Benutzerschnittstelle mit konfigurierbaren Einstellungsmöglichkeiten. Es soll beispielsweise möglich sein, die Position der Signatur, die verwendete Bürgerkartenumgebung (BKU), sowie den Speicherort des signierten PDF-Dokuments in der Konfiguration festzulegen.

Ebenso soll die Architektur der Software eine saubere Entkoppelung der Benutzerschnittstelle und der Dokumentverarbeitung vorsehen, wobei die Schnittstelle zur Dokumentverarbeitung möglichst generisch gehalten wird, um zukünftige Erweiterungen zu vereinfachen. Eine solche Erweiterung wäre beispielsweise der Austausch der PDF-AS-Komponente durch eine Bibliothek, die PAdES-kompatible Signaturen erstellt.

Es soll eine einheitliche, plattformunabhängige Möglichkeit geben, die Software zu installieren. Für ein automatisiertes Deployment unter Windows-Plattformen (beispielsweise bei Behörden) soll zusätzlich noch ein MSI-Installer-Paket zur Verfügung gestellt werden.

Dieses Dokument umreißt konzeptionell die Implementierung und das mögliche Design einer neuen Version von PDF-OVER 4.0. Grundlegende Anforderungen an die neue Version und mögliche Lösungsansätze werden diskutiert und dargestellt.

Der geschätzte Aufwand für dieses Projekt beträgt 72 PT. In weiterer Folge (Kapitel 5, Seite 12) wird eine detaillierte Abschätzung des Aufwands gegeben.

## 2. Projektziele

### 2.1. Zielsetzung

Ziel dieses Projekts ist die Neuentwicklung der Software PDF-OVER, welche zum elektronischen Signieren von PDF-Dokumenten mit der österreichischen Bürgerkarte eingesetzt wird. Das Signieren selbst erfolgt über eine Bürgerkartenumgebung (lokale BKU oder Handy-Signatur).

Die vorgeschlagene Neuentwicklung der Software soll möglichst benutzerfreundlich umgesetzt werden, damit der Signaturprozess für die Durchschnittsnutzerin bzw. den Durchschnittsnutzer verständlicher wird. Ebenfalls sollen die Ansprüche eines Power-Users berücksichtigt werden. Es soll möglich sein, in wenigen Schritten eine PDF-Signatur durchzuführen. In der Konfiguration sollen Vorauswahlen über die Bürgerkartenumgebung, Positionierung des Signaturblocks, den Speicherort des signierten Dokuments o.ä. getroffen werden können, damit die Benutzerin bzw. der Benutzer diese Auswahl nicht wiederholt treffen muss. Des Weiteren soll besonderes Augenmerk auf eine möglichst wiederverwertbare Architektur der Software gelegt werden. Die einzelnen Komponenten sollen modular aufgebaut werden und über gut definierte Schnittstellen miteinander kommunizieren. Ein Adapter soll für die Kommunikation zwischen externen Signaturtools (PDF-AS) und der Benutzeroberfläche eingesetzt werden. Damit soll gewährleistet werden, dass eine Anbindung an ein anderes Signaturtool (beispielsweise PAdES) mit wenig Aufwand möglich ist. Ebenso sollen die Schnittstellen zum Benutzerinterface so definiert sein, dass ein Anbinden an eine andere Benutzeroberfläche (beispielsweise HTML5-Interface) einfach möglich ist.

Folgende Punkte werden bei der Umsetzung besonders berücksichtigt werden.

- Die Auswahl des zu signierenden Dokumentes soll mittels „Drag und Drop“ auf das PDF-OVER Symbol möglich sein.
- Das Öffnen des zu signierenden Dokuments soll über einen Dateisuchdialog möglich sein.
- Die Auswahl des Dokuments soll nach dem Starten des Programms über „Drag und Drop“ möglich sein.
- PDF-OVER soll über einen Kommandozeilenaufwurf geöffnet werden können und das zu signierende Dokument soll als Argument mitgegeben werden können.
- Die Überprüfung, ob es sich um ein PDF-Dokument handelt, soll nicht über den Dateinamen erfolgen (PDF-Dokumente die keine .pdf Dateiendung haben sollen akzeptiert werden). Falls es sich um kein PDF-Dokument handelt, soll dementsprechend eine Fehlermeldung angezeigt werden.
- Die Benutzerin bzw. der Benutzer soll ein ausgiebiges Feedback über den Fortschritt des Signaturvorgangs erhalten.
- Die Benutzerin bzw. der Benutzer soll stets die Möglichkeit haben, einen Schritt zurück im Signaturprozess gehen zu können.
- Die Position des Signaturblocks soll im Dokument frei wählbar sein, optional soll der Signaturblock automatisch am Ende des Dokuments oder auf einer eigenen Seite eingefügt werden können (Konfigurationsoption).
- Die Positionierung des Signaturblocks soll mit Pfeiltasten wählbar sein.
- Die Seite soll ebenfalls mit Tastatur (Bild Auf / Bild Ab Tasten) wählbar sein.
- Die Signatur soll die binäre Methode verwenden.
- Optional soll es die Möglichkeit geben, die Bürgerkartenumgebung vorauszuwählen (lokale BKU, Handy-Signatur).
- Optional soll der Speicherort festgelegt werden können, sodass die Speicherung des signierten Dokuments automatisch erfolgt.
  
- Konfigurationsmöglichkeiten:
  - Bildmarke
  - Hinweistext
  - Schriftart des Signaturblocks

- Proxy
- Handy-Signatur-Einstellungen
- Bürgerkartenumgebung
- Automatische Speicherung des Dokuments
- Automatische Positionierung des Signaturblocks am Ende des Dokuments / auf einer eigenen Seite

## 2.2. Prozessbeschreibung

In diesem Abschnitt werden die drei wichtigsten Schritte beschrieben, die bei einem Signaturprozess durchlaufen werden. Der gesamte Ablauf wird in der Abbildung 1 in einem Zustandsdiagramm dargestellt.

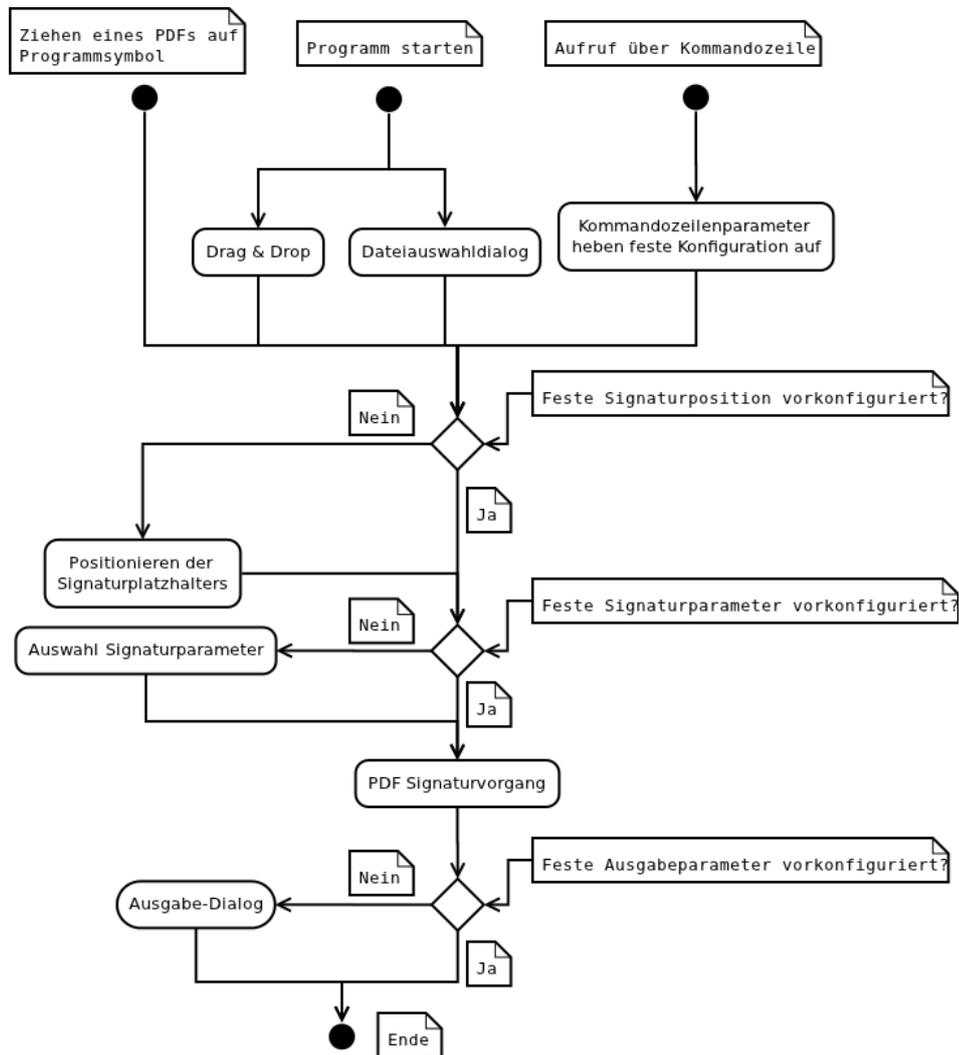


Abbildung 1 - Prozessbeschreibung

## 2.2.1. Auswahl des Dokuments

Die Auswahl des Dokuments kann wie in Abbildung 1 dargestellt auf drei unterschiedliche Arten erfolgen:

- Es besteht die Möglichkeit das zu signierende PDF Dokument auf das PDF-OVER Symbol zu ziehen. Der PDF-Signaturvorgang wird entsprechend den vom Benutzer konfigurierten Parametern fortgesetzt (siehe Abbildung 1). Sind keine entsprechenden Parameter konfiguriert, wird die Benutzerin bzw. der Benutzer aufgefordert, die Position für den Signaturplatzhalter auszuwählen.
- Wird das PDF-OVER Programm direkt gestartet, kann die Benutzerin bzw. der Benutzer das PDF-Dokument in ein Feld im Programm ziehen (drag and drop) oder über einen Dateiauswahldialog auswählen. Wie dieses Interface aussehen könnte, wird in Abbildung 2 skizziert.
- Des Weiteren kann PDF-OVER über die Kommandozeile gestartet werden. Dabei können bestimmte Parameter wie die Bürgerkartenumgebung, die Positionierung oder der Speicherort angegeben werden.

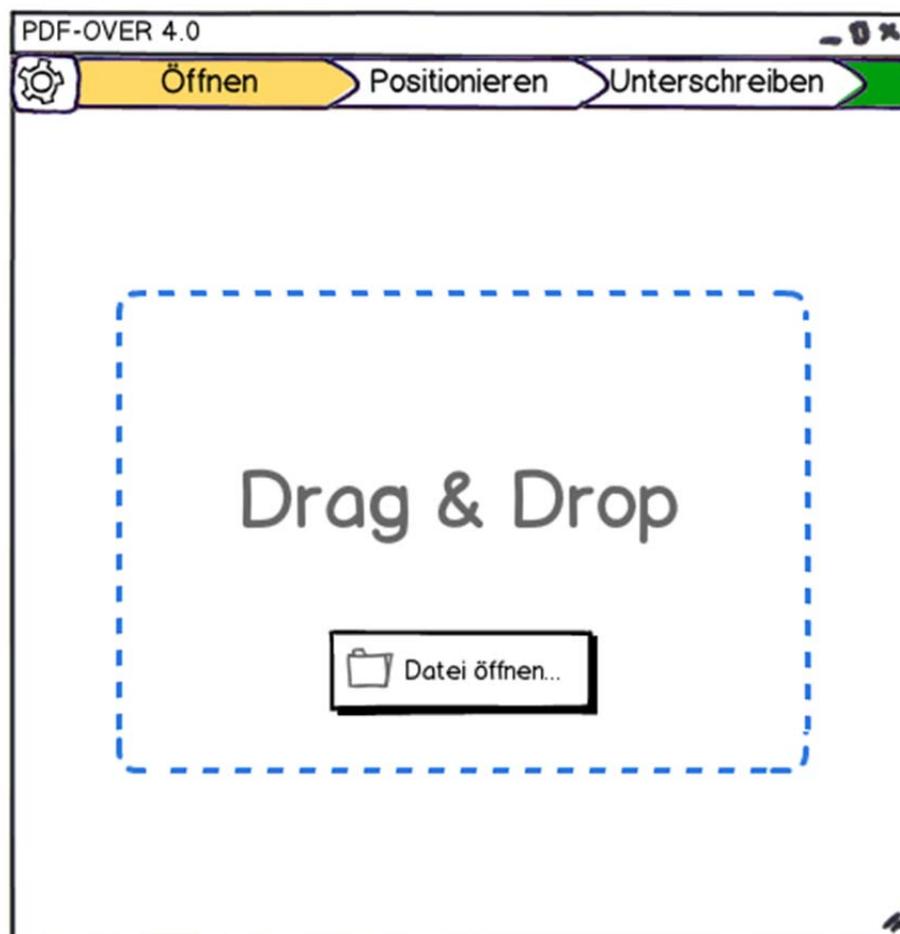


Abbildung 2 - Entwurf der Benutzeroberfläche (Schritt 1)

## 2.2.2. Positionierung des Signaturblocks

Nach dem Öffnen des Dokuments wird (sofern die Positionierung des Signaturblocks nicht im Vorhinein festgelegt wurde) das Dokument angezeigt. Dort kann dann die Position des Signaturblocks frei gewählt werden. Die Positionierung erfolgt über die Tastatur (mit den Pfeiltasten) oder mit der Maus. Es soll außerdem die Möglichkeit geben, den Signaturblock automatisch am Ende des Dokuments oder auf einer neuen Seite zu platzieren. In diesem Fall wird dieser Schritt übersprungen.

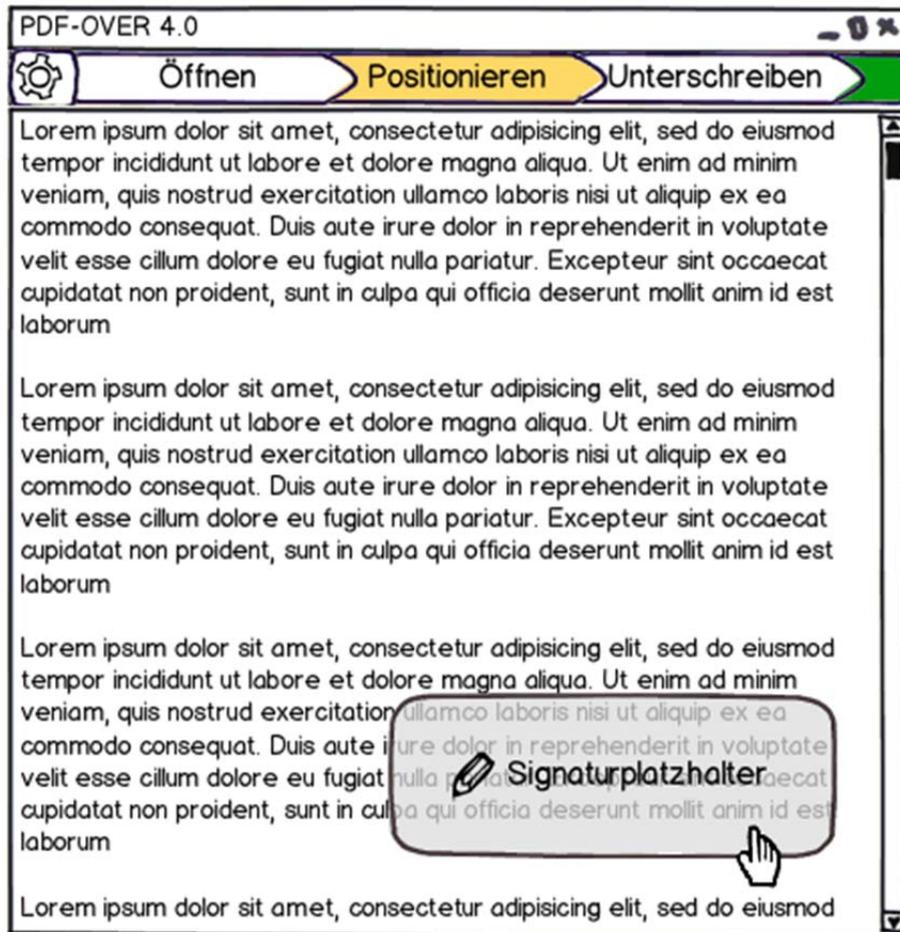


Abbildung 3 - Entwurf der Benutzeroberfläche (Schritt 2)

### 2.2.3. Signieren des Dokuments

In diesem Schritt wird, wenn nicht schon in der Konfiguration festgelegt, die Bürgerkartenumgebung ausgewählt, mit der die Signatur erfolgen soll. Danach erfolgt der eigentliche Signaturprozess über die gewählte BKU.

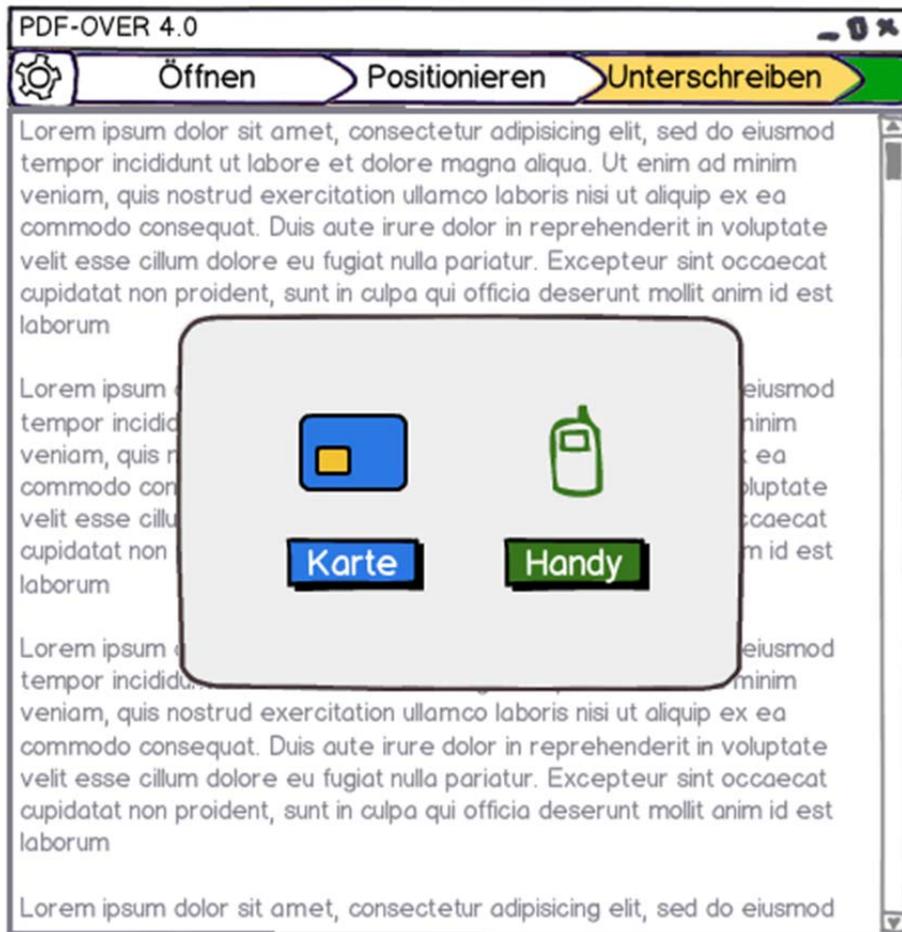


Abbildung 4 - Entwurf der Benutzeroberfläche (Schritt 3)

Nach dem Signieren wird die Benutzerin bzw. der Benutzer nach dem Speicherort gefragt, falls der Speicherort nicht schon per Konfiguration oder Kommandozeilenparameter festgelegt wurde. Abschließend wird der Benutzerin bzw. dem Benutzern ein Link zum gespeicherten Dokument angeboten. Dieser Abschlussdialog kann per Konfigurationsparameter deaktiviert werden.

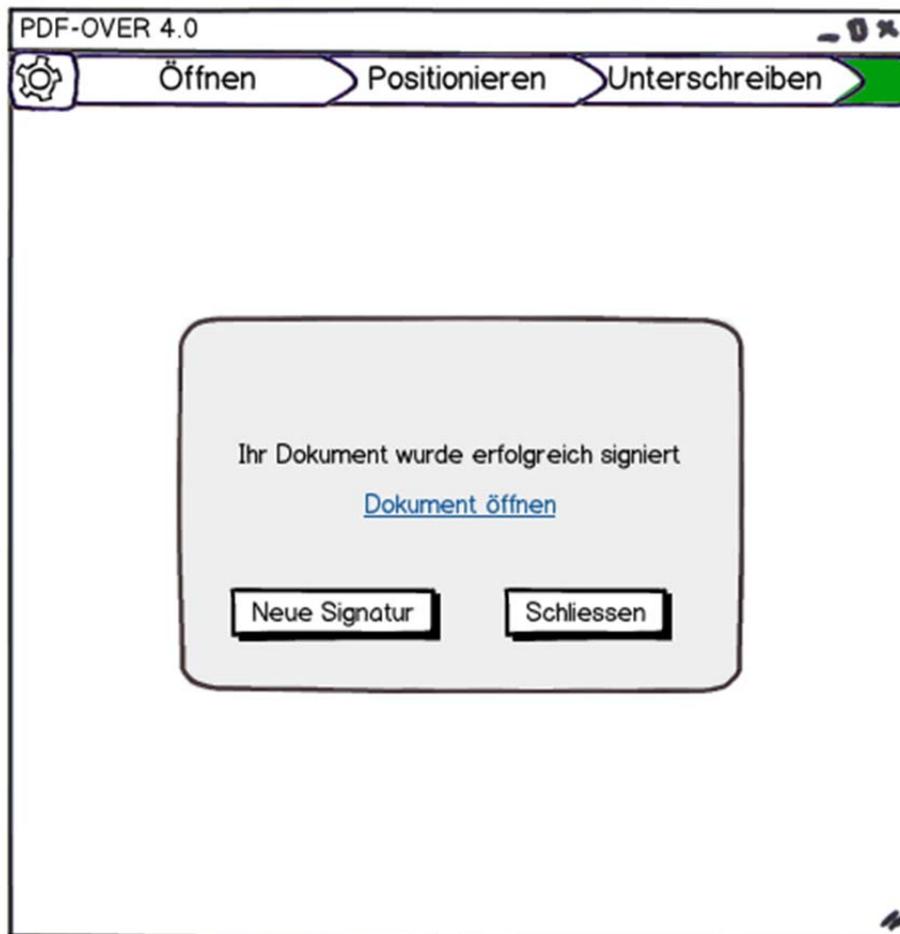


Abbildung 5 - Abschlussdialog

### 3. Systemkomponenten

Entsprechend den oben genannten Anwendungsfällen kann die PDF-OVER Applikation in folgende Komponenten gegliedert werden:

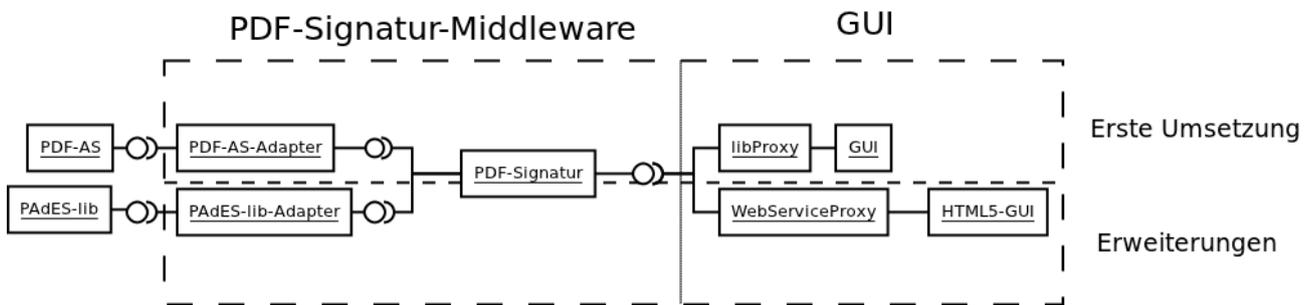


Abbildung 6 - Softwarearchitektur

#### 3.1. PDF-Signatur (Middleware)

Die Middleware der PDF-OVER Applikation abstrahiert den Signaturprozess, und bietet dem vorgeschalteten GUI ein möglichst einfaches Interface, um Dokumente zu signieren. Dazu implementiert sie die Schnittstellen zur PDF Amtssignatur Applikation, welche für das Aufbringen der Amtssignatur auf das PDF-Dokument zuständig ist. Diese Schnittstelle soll möglichst einfach austauschbar sein, um zukünftige PDF-Signatur-Bibliotheken (z.B. PAdES) zu unterstützen.

#### 3.2. GUI

Das Grafische Benutzerinterface (GUI) bedient sich der PDF-Signatur-Middleware. Diese Schnittstelle soll dabei sowohl direkt, als auch beispielsweise als Webservice angesprochen werden können. Dadurch werden zukünftige Erweiterungen (wie beispielsweise ein HTML5 Online-Signatur Interface) erleichtert.

#### 3.3. Externe Komponenten

Da die Umsetzung aus Gründen der Plattformunabhängigkeit in Java erfolgt, muss auf dem Zielsystem ein Java Runtime-Environment verfügbar sein. Zur Signatur wird eine Bürgerkartenumgebung (lokale BKU oder Handy-Signatur) eingesetzt. Die lokale BKU muss auf dem Zielsystem bereits installiert sein. Die PDF-Signatur-Middleware spricht die entsprechende PDF-Signatur-Bibliothek, in diesem Fall PDF-AS, an. Diese ist Teil der PDF-OVER Installation.

## 4. Arbeitspakete

Der aus der Durchführung des Projekts entstehende Arbeitsaufwand kann in folgende Arbeitspakete unterteilt werden.

Arbeitspaket 1: Spezifikation

- Schnittstellen
- Architektur
- Build-Prozess
- Softwareverteilung (Deployment)

Arbeitspaket 2: Design:

- Klassenentwurf (Klassendiagramm)
- Grafische Benutzeroberfläche (GUI)

Arbeitspaket 3: Implementierung

- PDF-Signatur-Middleware
- Grafische Benutzeroberfläche (GUI)
- PDF-AS- Schnittstelle
- Signaturschnittstelle (GUI-Schnittstelle)

Arbeitspaket 4: Dokumentation

- Anwenderdokumentation
- Entwicklerdokumentation

Arbeitspaket 5: Testen

Arbeitspaket 6: Inbetriebnahme

### 4.1. Spezifikation

**Ziel:**

- Präzise Anforderungen an die Software sollen festgelegt werden.
- Spezifizierung der Schnittstellen der einzelnen Softwarekomponenten und Erstellung einer Grobarchitektur.
- Festlegen der Implementierungssprache und die Entwicklungsumgebung.
- Definieren des Build-Prozesses und der Softwareverteilung.

**Tätigkeiten:**

- Spezifikation und Entwurf der Softwaremodule und ihrer Schnittstellen.

**Aufwand:**

10PT

### 4.2. Design

**Ziel:**

- Die Architektur der Software wird entsprechend den Anforderungen erstellt.
- Die Komponenten und ihre Schnittstellen werden entsprechend der Implementierungssprache definiert. Hier wird besonders darauf geachtet, dass die einzelnen Komponenten modular aufgeteilt werden.

**Tätigkeiten:**

- Das Design der Software wird in einem Klassendiagramm abgebildet (Klassenentwurf).
- Der Entwurf der Benutzeroberfläche wird erstellt.

- Die Benutzerfreundlichkeit des Designs soll mit Mockups (Entwürfen) schon in der Designphase überprüft werden.

**Aufwand:**

10 PT

### **4.3. Implementierung**

**Ziel:**

- Umsetzung der vorher definierten Architektur für PDF-Signatur Middleware, Benutzeroberfläche, PDF-Schnittstellen, Signatur-Schnittstellen

**Tätigkeiten:**

- Konzeption von Datenstrukturen und Algorithmen.
- Strukturierung des Programms durch geeignete Abstraktionsebene.
- Dokumentation der Problemlösungen und der Implementierungsentscheidung.
- Umsetzung der Komponenten in der oben definierten Programmiersprache.
- Gleichzeitiges Erstellen von Unit-Tests.
- Abschließendes Testen des Programms.

**Aufwand:**

40 PT

### **4.4. Dokumentation**

**Ziel:**

- Anwenderdokumentation

**Tätigkeiten:**

- Erstellung der Dokumentation.

**Aufwand:**

4 PT

### **4.5. Testdesign und Testen**

**Ziel:**

- Sicherstellung der korrekten Funktionalität der implementieren Anwendung.

**Tätigkeiten:**

- Durchführung ausgiebiger Tests schon während der Programmierung.
- Durchführung ausgiebiger Abschlusstests.

**Aufwand:**

6 PT

### **4.6. Inbetriebnahme der Software**

**Ziel:**

- Herstellung einer funktionstüchtigen Instanz der Implementierung auf einem Produktivsystem.

**Tätigkeiten:**

- Konfiguration des Zielsystems und Portierung der Anwendung.
- Abschließende Tests am Produktivsystem.

**Aufwand:**

2 PT

**5. Aufwand**

Entsprechend den in Abschnitt 4 beschriebenen Tätigkeiten ergibt sich für die Durchführung des Projekts der in Tabelle 1 zusammengefasste geschätzte Aufwand.

Arbeitspaket	Aufwand
Spezifikation	10 PT
Design	10 PT
Implementierung	40 PT
Dokumentation	4 PT
Testen	6 PT
Inbetriebnahme	2 PT
<b>Summe</b>	<b>72 PT</b>

*Tabelle 1 - Aufwandsabschätzung*

**Revisionsverlauf**

Version	Datum	Autor(en)	
0.1	15.05.2012	Vesna Krnjic	Dokumentstruktur
0.2	12.06.2012	Vesna Krnjic, Tobias Kellner	Zielsetzung, Aufwandsabschätzung
0.3	14.06.2012	Vesna Krnjic, Tobias Kellner Andreas Fitzek	Systemkomponenten
0.4	15.06.2012	Vesna Krnjic Tobias Kellner	Überarbeitung Ergänzung UI Mockup, Komponenten
0.5	19.06.2012	Vesna Krnjic Tobias Kellner	Überarbeitung, Fertigstellung Kommentare Herbert Leitold