



**advania**

Welcome to IT

Reykjavík 14.01.2016

**Advania PDF Signing service**

**Signet Forms**

*Version 1.0*

## TABLE OF CONTENTS

1	Scope and introduction .....	3
2	System overview .....	3
3	URL'S.....	3
3.1	Preproduction environment.....	3
3.2	Production environment .....	4
4	Authentication.....	4
4.1	certificate chains .....	4
4.2	Authentication problems .....	4
5	Website .....	4
5.1	Forms.....	4
5.2	Customized forms.....	5
5.3	Company interface .....	5
6	web service.....	5
6.1	Classes .....	5
6.1.1	ApplicationRequest .....	5
6.1.2	ApplicationData .....	5
6.1.3	Attribute .....	6
6.1.4	Applicant.....	6
6.1.5	TokenRequest.....	7
6.1.6	FormInfo .....	7
6.1.7	ApplicationNotification .....	8
6.2	Methods .....	8
6.2.1	ApplicationPDF - POST.....	8
6.2.2	ApplicationData - POST .....	8
6.2.3	Applications - POST.....	9
6.2.4	ApplicationData - POST .....	9
6.2.5	DeleteApplication - POST .....	9
6.2.6	GetForm - GET .....	9
6.2.7	GetForms – GET .....	10
6.2.8	GetToken – POST.....	10
6.2.9	TestNotify – POST.....	10
6.2.10	GetApplication – GET.....	10
6.2.11	GetApplicationData – GET .....	10
7	Status changes.....	11
7.1	Code example.....	11
8	Processes.....	11
8.1	Basic user flow.....	11

8.2	Preauthenticated user process.....	12
8.3	User fills form at service provider .....	13
9	Examples and instructions.....	14
10	Epilogou.....	14

## Version history

Date	Version	Description	Author/Approv.
13.01.16	1.0	First version	Sveinbjörn Óskarsson
Url for the document:			

## Table of figures

Figure 1. Basic system overview.....	3
Figure 2. Basic user flow for filling and signing forms.....	12
Figure 3. Pre-authenticated user sent to form.....	13
Figure 4. User fills out form and service provider .....	14

## 1 SCOPE AND INTRODUCTION

Signet Advania is a signing solution suite from Advania which has been in development since 2007 using technologies from companies like Ascertia and iText Software. Signet Forms is one component of the suite where companies can easily add PDF forms which can be filled and signed online.

This document describes the web and web services and its usage.

## 2 SYSTEM OVERVIEW

Signet is hosted on several servers behind a load balancer (LB). Below on picture 1 you find a rough overview of the Signet servers:

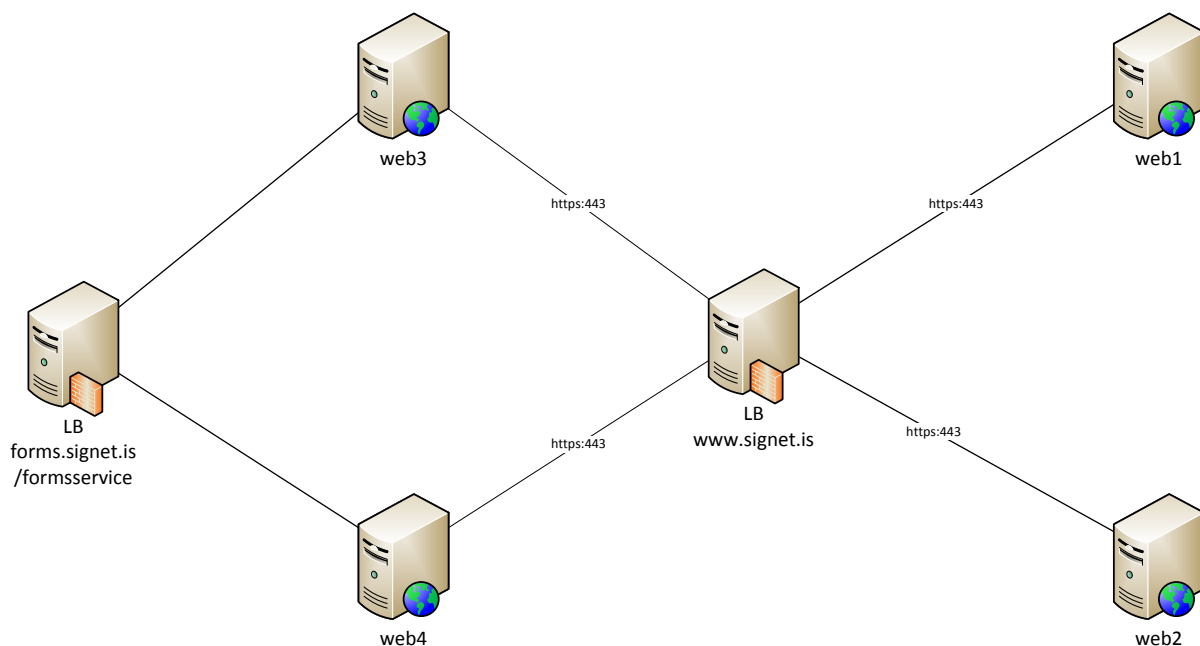


Figure 1. Basic system overview

## 3 URL'S

Signet Forms has both preproduction and production environment, hosted at signet.is. The web services and the Signet web require both electronic certificates as authentication. For preproduction, Advania can provide certificates for the web services authentication.

### 3.1 PREPRODUCTION ENVIRONMENT

The preproduction environment is accessible from following URLs:

- Signet Forms Web : <https://prufa.signet.is/forms2/>
- Web services: <https://prufa.signet.is/formsservice/> (REST Web API)
- Web service endpoint: <https://prufa.signet.is/formsservice/api/forms> (REST Web API)

## 3.2 PRODUCTION ENVIRONMENT

Production environment is accessible from following URLs:

- Signet Forms Web : <https://forms.signet.is>
- Web services: <https://forms.signet.is/formsservice/> (REST Web API)
- Web service endpoint: <https://forms.signet.is/formsservice/api/forms> (REST Web API)

## 4 AUTHENTICATION

The web services require authentication using electronic certificates at the transport layer.

### 4.1 CERTIFICATE CHAINS

Following certificate chains are trusted for authentication certificate to web services:

- Íslandsrót – Fullgildur búnaður
- Auðkennisrót – Traust auðkenni – Traustur búnaður
- Framtíðarrót – Signet skilríki (Only for preproduction environment)

### 4.2 AUTHENTICATION PROBLEMS

A common error in authenticating with digital certificates is:

```
The HTTP request was forbidden with client authentication scheme 'Anonymous'.
```

This error is usually saying that you tried to authenticate with a digital certificate but were unsuccessful in applying the private key. Two common causes are:

- The user (app pool etc) does not have permission to use the private key. This can be sorted by giving the user permission on the certificates private key (in Windows right click the certificate in the certificate manager and select manage private keys).
- The certificate chain is incomplete and the machine is not able to correctly select the certificate. This can be sorted by adding the root and intermediate certificates to the certificate store. The chain for test authentication certificates can be downloaded from <https://prufa.signet.is/documentation>.

## 5 WEBSITE

The website is available at the URLs described in chapter 3 above. The website contains a few different processes which will be described below.

### 5.1 FORMS

The Forms are available at <url>/Home/Forms. You can see a list of only your forms by adding the query parameter ssn=SSN of company, i.e. <url>/Home/Forms?ssn=5902697199 to see a list of Advania Forms. The path to a form is normally /Home/SubmitAuth/<form id>

## 5.2 CUSTOMIZED FORMS

Advania can customize the forms if needed to add logic that will hide fields depending on conditions, rearranging of fields etc. If the form has been customized the path to the form is /home/submit/<form id>.

## 5.3 COMPANY INTERFACE

Registered company users can access an interface for company users where you can see registered forms, registered applications as well as delete applications and fetch application data. The interface is available at <url>/Company.

## 6 WEB SERVICE

The webservice is a RESTful webservice which accepts JSON or XML messages at /api/forms. This chapter describes the webservice interface and classes.

### 6.1 CLASSES

Following is a description of the classes used in the webservice interface.

#### 6.1.1 APPLICATIONREQUEST

When requesting application info the request should contain an ApplicationRequest model which is as follows

- AppID
  - integer
  - The application ID
- SignetDocumentID
  - string
  - The is of application document ID in Signet.
- FormID
  - integer
  - The ID of the form used
- UserSSN
  - string
  - National registry number of user.
- FromTime
  - date
  - Application submit from date
- ToTime
  - date
  - Application submit to date

#### 6.1.2 APPLICATIONDATA

When requesting application data the response is given as an ApplicationData model which is as follows:

- AppID

- integer
  - ID of application.
- FormID
  - integer
  - ID of form used
- Filename
  - string
  - Filename of PDF
- Deleted
  - boolean
  - Has the application been deleted
- SubmitTime
  - date
  - Time of submit.
- SignedTime
  - date?
  - Time of signature (nullable) if signed.
- DeleteTime
  - date?
  - Time of deletion (nullable) if deleted
- Applicants
  - Collection of Applicant
  - List of applicants.
- Attributes
  - Collection of Attribute
  - List of application attributes

### 6.1.3 ATTRIBUTE

The forms contain attributes which are of the model class Attribute which is as follows:

- Key
  - string
  - The HTML id/name of attribute.
- DisplayName
  - string
  - The display name of attribute
- Visible
  - boolean
  - Is the attribute visible in the form or not
- Type
  - string
  - The attribute type (CheckBox, Combo, List, None, PushButton, RadioButton, Signature or Text)

### 6.1.4 APPLICANT

The forms contain applicants which are of model class Applicant which is as follows:

- SSN
  - string
  - National registry number of applicant.

- Name
  - string
  - Name of applicant.
- SigningText
  - string
  - Applicants signing text.
- Signed
  - boolean
  - Has the applicant signed the application.
- SigningTime
  - date?
  - The time of signature (nullable) if signed.

#### 6.1.5 TOKENREQUEST

When requesting SSO tokens for sending users straight to the form or for submitting the form remotely the request should contain an TokenRequest model which is as follows:

- Destination
  - string
  - The URL of action to submit to
- Mobile
  - string
  - The mobile phone of user (which contains his digital certificates).
- returnUrl
  - string
  - The URL where the user should be redirected to after signing
- Ssn
  - string
  - SSN of user/applicant

#### 6.1.6 FORMINFO

When requesting info about Forms the info is returned as a FormInfo model which is as follows:

- ID
  - integer
  - The ID of the form.
- Name
  - string
  - The name of the form.
- Description
  - string
  - The description of the form
- FileName
  - string
  - The filename of PDF
- FormString
  - string
  - The HTML form string if available
- Enabled
  - boolean



- Is the form enabled (visible) or not.
- Attributes
  - Collection of Attribute
  - List of attributes in form

### 6.1.7 APPLICATIONNOTIFICATION

When Signet Forms notifies of applications it will send the notification as an ApplicationNotification model which is as follows:

- AppID
  - integer
  - The ID of the application.
- FormID
  - integer
  - The ID of the form.
- NumSigners
  - integer
  - The number of signers on form
- SignersLeft
  - integer
  - The number of signers left to sign
- SubmitTime
  - date
  - Application submit datetime
- SignedTime
  - date?
  - Application signature datetime (nullable) if signed
- Status
  - ApplicationStatus
  - The status of form/application (New, InSigning or Signed)

## 6.2 METHODS

Following is a description of the available methods. Short description of methods are available at <https://forms.signet.is/formsservice/Help>. All methods are accessible at <url>/api/forms/<method>.

### 6.2.1 APPLICATIONPDF - POST

The method ApplicationPDF allows you to fetch the signed (or only filled if not signed yet) PDF document giving the application ID in an ApplicationRequest request.

```
public byte[] ApplicationPDF(ApplicationRequest value)
```

- value
  - ApplicationRequest
  - The application request with application ID

The method returns the PDF in byte array.

### 6.2.2 APPLICATIONDATA - POST

The method ApplicationData allows you to fetch application data giving the application ID in an ApplicationRequest request.

```
public ApplicationData ApplicationData(ApplicationRequest value)
```

- value
  - ApplicationRequest
  - The application request with application ID

The method returns the data in an ApplicationData form (see above).

### 6.2.3 APPLICATIONS - POST

The method Applications allows you to fetch all applications PDF that fulfil the ApplicationRequest request.

```
public byte[] ApplicationData(ApplicationRequest value)
```

- value
  - ApplicationRequest
  - The application request with search parameters

The method returns the data in a zipped byte[] array.

### 6.2.4 APPLICATIONDATA - POST

The method ApplicationData allows you to fetch application data giving the application ID in an ApplicationRequest request.

```
public List<ApplicationData> ApplicationsData(ApplicationRequest value)
```

- value
  - ApplicationRequest
  - The application request with search parameters

The method returns the data in a ApplicationData collection form (see above).

### 6.2.5 DELETEAPPLICATION - POST

The method DeleteApplication allows you to delete all application that fulfil the ApplicationRequest request.

```
bool DeleteApplication(ApplicationRequest value)
```

- value
  - ApplicationRequest
  - The application request with search parameters

The method returns true if the delete was successful.

### 6.2.6 GETFORM - GET

The method GetForm allows you to fetch form information given the form ID.

```
FormInfo GetForm(int id)
```

- id
  - integer
  - The form ID

The method returns the form info as a FormInfo model.

### 6.2.7 GETFORMS – GET

The method `GetForms` allows you to fetch all forms registered to companies

```
List<FormInfo> GetForms()
```

The method returns a collection of `FormInfo` models.

### 6.2.8 GETTOKEN – POST

The method `GetToken` allows you to get an authentication token to post a user directly to a form to fill (`/Home/SubmitToken`) or submit a filled form (`/Home/SubmitTokenModel`).

```
string TestNotify(ApplicationRequest value)
```

- value
  - `ApplicationRequest`
  - The token request with appropriate parameters

The method returns a base64 encoded SAML 2.0 authentication token to send user to a form or action.

### 6.2.9 TESTNOTIFY – POST

The method `TestNotify` allows you to test your registered endpoint were the service will send a notification message with the status of the application with the ID sent.

```
string GetToken(TokenRequest request)
```

- request
  - `TokenRequest`
  - The application request with application ID

The method returns a string with a result message.

### 6.2.10 GETAPPLICATION – GET

The method `GetApplication` allows you to fetch the signed (or only filled if not signed yet) PDF document giving the application ID.

```
byte[] GetApplication(int id)
```

- id
  - integer
  - The application ID

The method returns the PDF in byte array.

### 6.2.11 GETAPPLICATIONDATA – GET

The method allows you to fetch application data giving the application ID.

```
GetApplicationData GetApplicationData(int id)
```

- value
  - id
  - The application ID

---

The method returns the data in an ApplicationData form (see above).

## 7 STATUS CHANGES

Advania can register a RESTful endpoint which receives the status changes (POST) of applications/forms. Status changes are sent with ApplicationNotification class as described above webservice chapter. Signet Forms supports endpoints with a) no authentication and b) endpoints with username and password authentication (basic) and c) endpoints with electronic certificates as an authentication method. The status can be POSTed as either XML or JSON.

### 7.1 CODE EXAMPLE

An example web service for status changes can be found at the site where the examples and instructions can be found (<https://prufa.signet.is/documentation/>).

## 8 PROCESSES

Signet Forms supports different processes for filling and signing of forms. This chapter describes the three basic processes.

### 8.1 BASIC USER FLOW

The figure below describes the flow of a basic form filling and signing where there is only one signer.

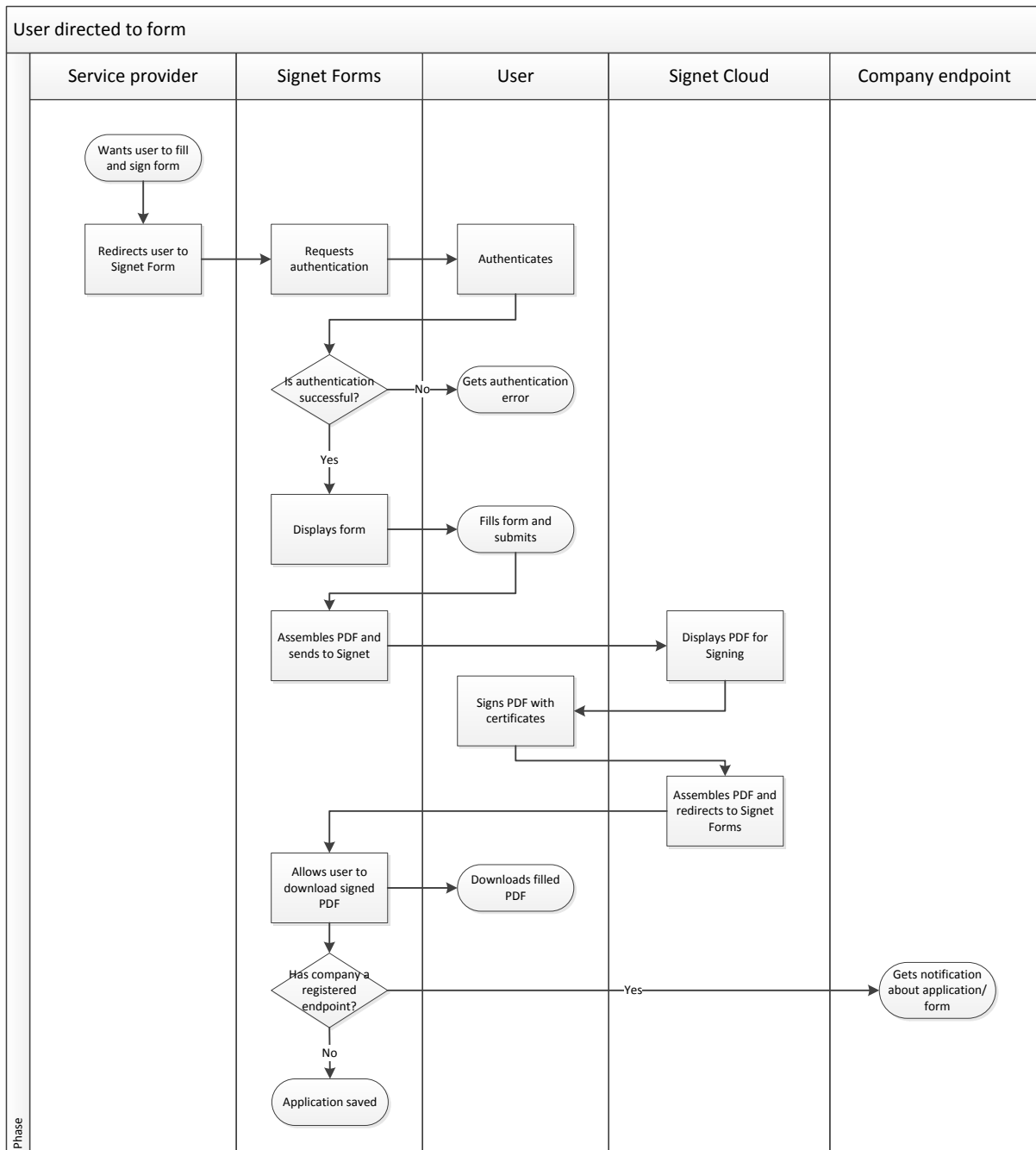


Figure 2. Basic user flow for filling and signing forms

## 8.2 PREAUTHENTICATED USER PROCESS

The figure below describes the process when user is pre-authenticated at service provider and sent straight to form for filling signing.

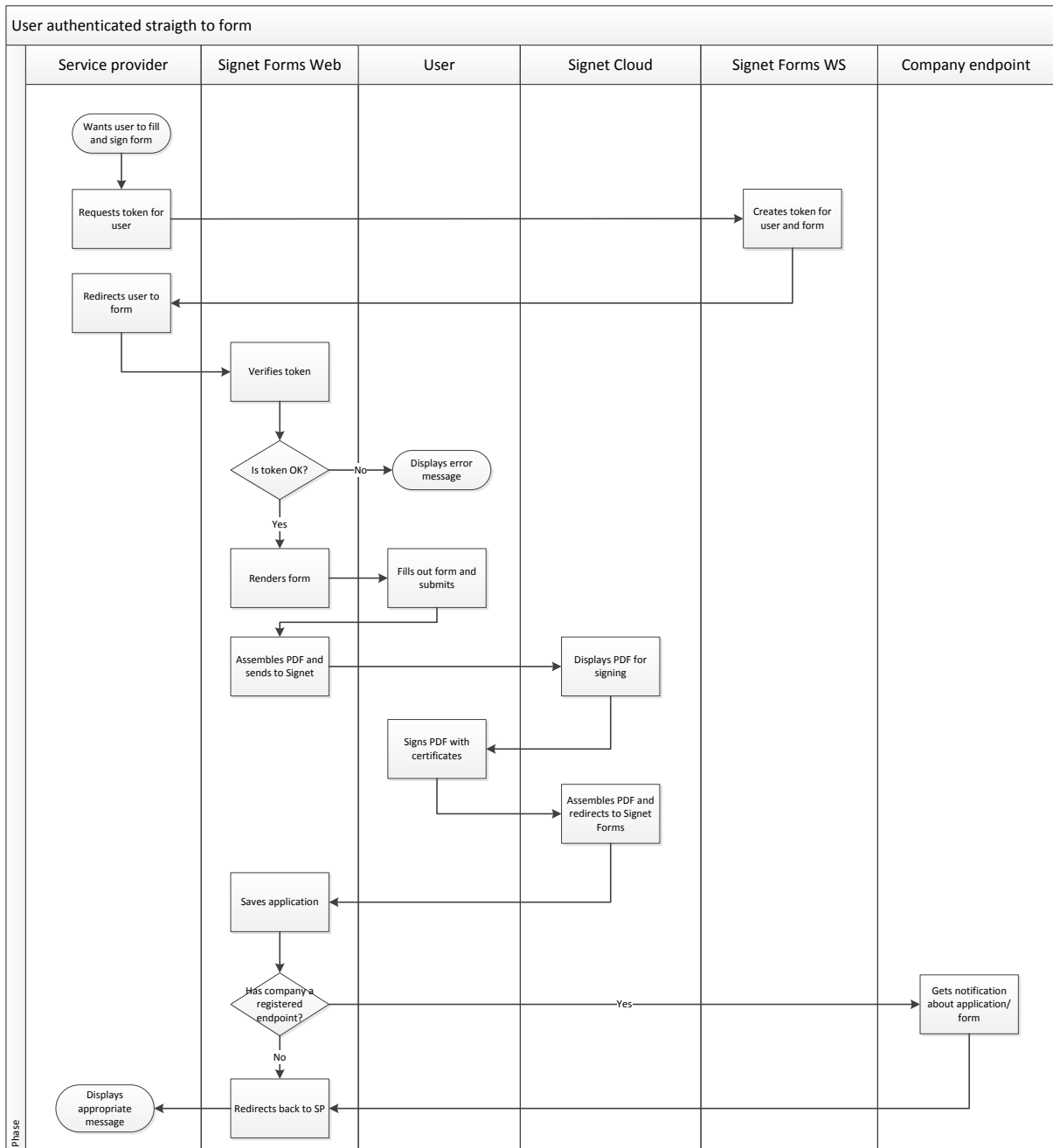


Figure 3. Pre-authenticated user sent to form

### 8.3 USER FILLS FORM AT SERVICE PROVIDER

The figure below describes the process when the user is authenticated and fills the form at service provider and Signet Forms only handles assembly and signing.

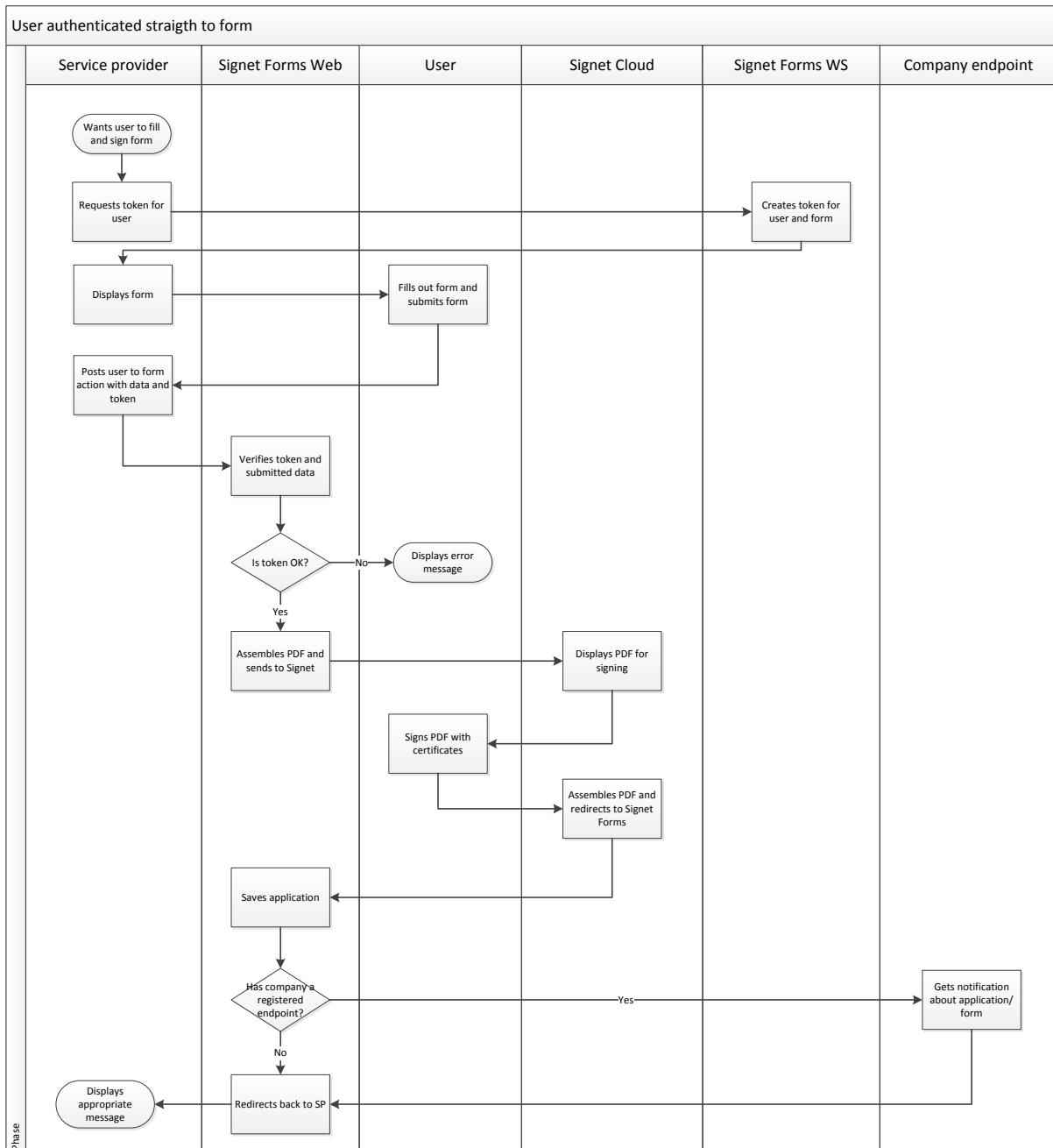


Figure 4. User fills out form and service provider

## 9 EXAMPLES AND INSTRUCTIONS

The code examples and instructions can be found at: <https://prufa.signet.is/documentation>.

## 10 EPILOGOU

For further information about the solution and technical support contact Security solutions at Advania, [oryggislausnir@advania.is](mailto:oryggislausnir@advania.is).