Project information

Customer/Project

General Relaying Party API

Document type

GRP Specification

Title

GRP API ver 1.1 for CGI eID service

Doc. ID	Rev.
	1.1
Produced by	Saved
CGI	2015-10-21
Approved by	Date / Signature
funktionstjanster@cgi.com	

Contents

Introduction

This document is the API description of the General Relaying Party (GRP) API that will take care of the new BankID infrastructure change that will be conducted during 2014 and also replace the previous Mobilt BankID API (MBI)

With the GRP api you will be able to handle authentication and signatures with "Mobilt BankID" as well as "BankID på fil" and "BankID på kort". But it is also designed to handle other new vendors of mobile authentication that may occur. Today we know that Telia during 2015 will launch TeliaEID App, and will then work with the GRP api.

The GRP api will handle all the communication to BankID´s RP Interface and work as a proxy. (In 2015 also with Telias RP-interface)

Two basic use case.

With the use of "BankID Säkerhetsprogram" BISP v5.1for PCs and "Mobilt BankID app" for mobile devices, this GRP api can be used in two different ways.

1. BankID on same unit: When the BISP are in the same PC/Mac you can get a "AutoStartToken" from the GRP api and then start the BISP localy on the PC/Mac
2. BankID on another unit: For Mobilt BankID or if BISP are on another PC/Mac than the one used. You have to ask the user for personalNumber (swe personnummer) and call GRP with that.

It is possible to also start "BankID säkerhetsapp"" with an "AutoStartToken", consider that if you develop an own native app but in most cases it is best to ask the user if the BankID are on the same device or on another device.

If you previous have used MBI api. Mobilt BankID

If you already have used the MBI api for Mobilt BankID, you will recognise the GRP api as an extension of the MBI where the ability to use "AutoStartToken" have been added and some error codes have been changed.

It will be very easy to change from MBI to the GRP api. Just import the new wsdl and change your code to use that api instead.

If you previous have used OSIF api. BankID på fil and BankID på kort

You will find some similarities to the OSIF 2.1 API in some parts, so there will be easier to complement or change an existing OSIF[1] solution with GRP. The GRP´s error handling follows normal SOAP error handling in contrast to the error handling i OSIF that is a bit special.

- You don´t have to control version of BISP or call on the Autentication or Signer2 plug-in/activeX and generate a challenge.

[1] OSIF 2.1, a web service API used for certificates (e-legitimation/BankID) on file and on cards in Sweden.

- If you support both PC and smartphone, you have to use different iframe to launch the BankID program.
- You will typically use “AutoStartToken” method in all places you use “BankID på fil” and “BankID på kort” today.

General Recommendations from eID-service

Frames, like this one, will contain specific information and comments regarding the GRP implementation in CGI:s eID service.. This concerns the normal customers connections to the eID service and use of the Swedish certificates. “Mobilt BankID” and “BankID på fil” and “BankID på kort”

There is a test eID environment and a production eID environment. How to communicate with these eID systems are described in the document “Teknisk Bilaga”.
GRP test you find at: https://grpt.funktionstjanster.se:18898/grp/v1 (recommended)
http://159.72.136.9:18899/grp/v1

How to get test certificates “Mobilt BankID” or “BankID på fil” and find more documentation about BankID in CGIs eID Portal.

https://funktionstjanster.primeportal.com/eID/

Please notice that you have to configure your BankID program/BankID App to communicate to the infrastructures test environment. You can´t use test and production environment at the same time from the same device.

Support and user accounts to eID portal ore ServiceID, please contact us at

funktionstjanster@cgi.com

Data types

This chapter describes some common compound data types used in the requests and responses.
The input and output data to the GRP server, is based on the use of ”*complex types*”, i.e. a data type that consists of a number of subordinated data types. For C programmers this corresponds to a *struct*, and for VB programmers it corresponds to a T*ype*. In this document however, we use the programming language neutral concept of *complex type*.

Some general recommendations

Recommendation using GRP

transactionId
A transaction ID in the request. Used for tracking purposes. The *transactionId* that is sent in the request will be returned in the response. If no *transactionId* was given in the request, the server will generate one and return it in the response
We recommend you to use *transactionId*. It will be helpful if tracing transaction between your application and eID service.
This is the same transactionId that are used in OSIF and MBI

Policy
Every customer in the eID-service receives a unique “ServiceID” that must be used in the GRP. Your ServiceID in GPR are the same that you already have if you use the OSIF 2.1 api.

In the eID-service environment you will receive different ServiceID to be used in test and production.. You will find your ServiceID in your “Teknisk Bilaga” or contact CGI to get one.

This is the same Policy that are used in OSIF and MBI. Use the same.

For GRP in test, you can use any serviceID logtest001, logtest002,.. to logtest010.

Provider

For future proofing we have added this component. Please use this parameter with the value “bankid”.

In OSIF you have provider=6 for BankID. When you call GRP you should instead use provider=bankid.

(In 2015 when TeliaEID App is launched, you should use provider=telia for Telia e-legitimatin)

displayName

The displayName will be displayed for the user in “BankID Säkerhetsprogram” or “BankID säkerhetsapp”. It should be your organizations brand or name.

This is the same as displayName in MBI. If you don´t have a displayName for your organization you have to order one from CGI. displayName max 30 character long.

PersonalNumber

```
Begin
        String    PersonalNumber
End
```

Description

Swedish civil registration number "personnummer" in 12 digits. (YYYYMMDDXXXX)

Schema

```
<xsd:simpleType name="PersonalNumberType">
        <xsd:annotation>
                <xsd:documentation>
                        An personal identity number should have length 12 and
contain only digits.
                </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
                <xsd:pattern value="\d{12,12}" />
        </xsd:restriction>

</xsd:simpleType>
```

Element description

Name	Data type	Occurs	Description
PersonalNumber	xsd:string	[0..1]	Must have 12 digits.

ProgressStatus

```
Begin
        String    ProgressStatus
End
```

Description

Status contains codes and descriptions that are returned by all operations.

Schema

```
        <xsd:simpleType name="ProgressStatusType">
                <xsd:restriction base="xsd:string">
                        <xsd:annotation>
                                <xsd:documentation>
                                        The values may be changed later.
                                </xsd:documentation>
                        </xsd:annotation>
                        <xsd:enumeration value="COMPLETE" />
                        <xsd:enumeration value="USER_SIGN" />
                        <xsd:enumeration value="OUTSTANDING_TRANSACTION" />
                        <xsd:enumeration value="NO_CLIENT" />
                        <xsd:enumeration value="STARTED" />
                </xsd:restriction>

        </xsd:simpleType>
```

Element description

Name	Data type	Occurs	Description
ProgressStatus	xsd:string	[1..1]	Status of transaction

The value for ProgressStatus can be any of following with corresponding suggested user text:

ProgressStatus	Description
OUTSTANDING_TRANSACTION	A: If ”AutoStartToken” is used and you tried to launch” BankID Säkerhetsapp” or” BankID Säkerhetsprogram”. Recommended text to the user. “Försöker starta BankID programmet” B: If you didn´t try to launch “BankID Säkerhetsapp” or” BankID Säkerhetsprogram”. Recommended text to the user. “Starta BankID-programmet”
USER_SIGN	The “BankID Säkerhetsapp” or “BankID Säkerhetsprogram” have received the authentication/signing request. Recommended text to the user: “Skriv in din säkerhetskod i BankID programmet och välj Legitimera eller Skriv under”
COMPLETE	The authentication / Signing are completed.
NO_CLIENT	Start of BankID failed. Users BankID was not available. Recommended text to the user. “Starta BankID-programmet”
STARTED	One BankID client have been started, but there are no usable BankID on the client. Recommended text to the user. “Du har inget BankID som går att använda för den här inloggningen/underskriften på den här datorn/enheten. Om du har BankID på kort, sätt in det i läsaren. Om du inte har något BankID kontakta din Bank.”

GrpFault

```
Begin
      FaultStatusType   faultStatus
      String            detailedDescription
End
```

Description

Error messages from GRP. If receive GrpFault, don't continue to call on the GRP service. Show the error message detaildDescription to the user.

```
<xsd:complexType name="GrpFaultType">
	<xsd:sequence>
		<xsd:element name="faultStatus"
type="tns:FaultStatusType" />
		<xsd:element name="detailedDescription"
type="xsd:string" />
	</xsd:sequence>
</xsd:complexType>
```

Element description

Name	Data type	Occurs	Description
FaultStatus	xsd:string	[1..1]	See FaultStatus below0
detailedDescription	xsd:string	[1..1]	Error text in Swedish that are recommended to the user.

FaultStatus

```
Begin
	String	FaultStatus
End
```

Description

Error codes from MBI.

Schema

```
	<xsd:simpleType name="FaultStatusType">
		<xsd:restriction base="xsd:string">
			<xsd:annotation>
				<xsd:documentation>
					The values may be changed later.
				</xsd:documentation>
			</xsd:annotation>
			<xsd:enumeration value="INVALID_PARAMETERS" />
			<xsd:enumeration value="ALREADY_IN_PROGRESS" />
			<xsd:enumeration value="ACCESS_DENIED_RP" />
			<xsd:enumeration value="RETRY" />
			<xsd:enumeration value="INTERNAL_ERROR" />
			<xsd:enumeration value="EXPIRED_TRANSACTION" />
			<xsd:enumeration value="USER_CANCEL" />
			<xsd:enumeration value="CLIENT_ERR" />
			<xsd:enumeration value="CERTIFICATE_ERR" />
			<xsd:enumeration value="CANCELLED" />
			<xsd:enumeration value="START_FAILED" />
		</xsd:restriction>
	 </xsd:simpleType>
```

Element description

Name	Data type	Occurs	Description

FaultStatus	xsd:string	[1..1]	Error message

FaultStatus	Description “recommended text to user”
INVALID_PARAMETERS	Error when calling the service. You have probably a internal technical problem. If you can´t solve this by your own, please contact CGI. If you use a OrderRef that previously resulted in COMPLEAT you get INVALID_PARAMETERS, the order cannot be collected twice.
ALREADY_IN_PROGRESS	An order for the user is already in progress. The order is aborted. No order is created. Recommended text to the user: “Åtgärden avbruten. Försök igen”
ACCESS_DENIED_RP	This is a internal access problem. Please, contact CGI. Recommended text to the user: “ Internt tekniskt fel. Försök igen vid ett senare tillfälle.”
RETRY	A temporary internal BankID error has occurred. Recommended text to the user: “ Internt tekniskt fel. Försök igen lite senare.”
INTERNAL_ERROR	A internal BankID error has occurred. Recommended text to the user: “Internt tekniskt fel. Försök igen.”
EXPIRED_TRANSACTION	No activity from the user and a time-out has occurred. (180 s) Recommended text to the user: “BankID-programmet svarar inte. Kontrollera att det är startat och att du har internetanslutning. Försök sedan igen.”
USER_CANCEL	The user have interrupt the authentication/signing. Recommended text to the user: “Åtgärden avbruten”
CLIENT_ERR	Internal BankID error. Recommended text to the user: “Intern tekniskt fel. Uppdatera BankID-programmet och försök igen”
CERTIFICATE_ERR	The users BankID are revoked, invalid or wrong security code have been entred to many times. Recommended text to the user: “Det BankID du försöker använda är för gammalt eller spärrat. Använd ett annat BankID eller hämta ett nytt hos din Bank”
CANCELLED	The order was cancelled. The system received a new order for the user. Recommended text to the

	user: “Åtgärden avbruten. Försök igen”
START_FAILED	Error starting BankID client after 30s. BankID not installed or problem with users computer. Recommended text to the user: “BankID-programmet verkar inte finnas i din dator eller telefon. Installera det och hämta ett BankID hos din Bank. Installera programmet från install.bankid.com”

endUserInfo

This is BankID specific

Name	Data type	Occurs	Comment
IP_ADDR			IP-address to the user´s computer/device.

requirementAlternativsType

This is BankID specific. Se BankID Relyning Party Guidelines v2.5, session 12.6 for more information. Don´t use this unless you have special requirements.

Name	Data type	Occurs	Comment
CardReader			Type of card reader that is allowed. “class1” (default) any card reader. “class2” a reader where the PIN-code must be entered on the reader. *Not recommended to use.*
CertificatePolicise			Use only if you want to limit the allowable methods: “1.2.752.78.1.1” – BankID på fil “1.2.752.78.1.2” – BankID på kort “1.2.752.78.1.5” – Mobilt BankID “1.2.752.71.1.3” – Nordea e-legitimation. In 2015 when TeliaEID App launch “1.2.752.35.1.3” TeliaEID on file “1.2.752.35.1.4” TeliaEID on card “1.2.752.35.1.5” Mobile TeliaEID In test enviroment “1.2.3.4.5” – BankID på fil “1.2.3.4.10” – BankID på kort “1.2.3.4.25” – Mobilt BankID “1.2.752.71.1.3” – Nordea e-legitimation.

			In 2015 when TeliaEID App launch “1.2.752.35.99.2” TeliaEID on file “1.2.752.35.99.1” TeliaEID on card “1.2.752.35.99.3” Mobile TeliaEID
IssuerCn			Use only if you want to restrict the use of BankID bases on Issuer Common Name. *Not recommended to use.*
AutoStartToken Required			If set to Yes, the client must have been started using the autoStartToken. To be used only if it is important that the BankID App is on the same device as your service.

Property

```
Begin
      String name
      String value
End
```

Description

Property is a compound type that consists of a name and its associated value. Values for name are defined in Attributes

Schema

```
<xsd:complexType name="Property">
        <xsd:sequence>
                <xsd:element name="name" type="xsd:string"/>
                <xsd:element name="value" type="xsd:string"/>
        </xsd:sequence>

</xsd:complexType>
```

Element description

Name	Data type	Occurs	Description
name	xsd:string	[1..1]	The name of the property.
value	xsd:string	[1..1]	The value of the property.

AuthenticateRequest

```
Begin
      String policy
      String dispayName
      String transactionId
      personalNumber personalNumber
      String provider
      String endUserInfo
```

```
      String requirementAlternatives
End
```

Description

Request an autentification of the user..

Schema

```
<xsd:complexType name="AuthenticateRequestType">
        <xsd:sequence>
                <xsd:element name="policy" type="xsd:string" />
                <xsd:element name="displayName" minOccurs="0" type="xsd:string"
/>
                <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
                <xsd:element name="personalNumber" type="tns:PersonalNumberType"
/>
                <xsd:element name="provider" minOccurs="0" type="xsd:string />
                <xsd:element name="endUserInfo" minOccurs="0" type="xsd:string"
/>
                <xsd:element name="requirementAlternatives" minOccurs="0"
type="xsd:string" />

        </xsd:sequence>
</xsd:complexType>
```

Element description

Name	Data type	Occurs	Comment	Optional
policy	xsd:string	[1..1]	Your customer specific ServiceID	
displayName	xsd:string	[0..1]	The text user will se in BankID Säkerhetsapp or BankID Säkerhetsprogram.	X
transactionId	xsd:string	[0..1]	A unique ID for the transaction.	X
personalNumber	String	[0..1]	Swedish “personnummer” in 12 digits. (ååååmmddxxxx) Optional if AutoStartToken method will be used.	X If not used, then “AutoStart-Token” method must be used.
provider	String	[1 1]	Use “bankid” (2015 “telia”)	
endUserInfo	String	[0 1]	Possible to add the users IP-adress. IP_ADDR	X
requirementAlternatives		[0 1]	Use if you have specific demands on: Cardreader CertificatePolicy IssuerCN AutoStartTokenRequired	X

General Recommendation
displayName
We recommend you to use displayName, even if you only have one displayName in

the beginning. (displayName are presented for the user in BankID Säkerhetsprogram and BankID Säkerhetsapp). In test there exist two possible displayNames "Test av Mobilt BankID" and the other "Funktionstjänster Test". The displayName should not exceed 30 characters. If you don´t have a displayName contact CGI. Delivery time for a displayName in production are 5-10 days.

Jag legitimerar mig mot: displayName

personalNumber
Usually used when BankID are on another device, but can also be used if you already know the users personalNumber and only want to allow that user to authenticate or sign. Can be useful in the use case where sign are followed after an authentication.

Provider

For BankID use "bankid".

In 2015 use "telia" for Telia

endUserInfo

Used for passing information relatated to the user and the users computer/device. (no controls are done by BankID in the current solution). If you are able to add the users IP-adress, we recommend you to do so.

```
<endUserInfo>
   <type>IP_ADDR</type>
   <value>192.169.0.1</value>  /*example IP-address */
</endUserInfo>
```

This function is a preparation for future extra security controls made by BankID. We recommend you to use this parameter, just to be prepared.

requirementAlternatives

We recommend that you don´t use this function unless you have very specific requirements. Only allow specific BankID certificate, specific issuer eg. (See BankID Relying party Guidelines v2.5, section 12.6

SignatureRequest

```
Begin
      String policy
      String displayName
      String transactionId
      personalNumber personalNumber
      String userVisibleData
      String userNonVisiableData
      String provider
```

```
        String endUserInfo
        String requirementAlternatives

End
```

Description

Request a digital signature.

Schema

```
<xsd:complexType name="SignatureRequestType">
        <xsd:sequence>
                <xsd:element name="policy" type="xsd:string" />
                <xsd:element name="displayName" minOccurs="0" type="xsd:string"
/>
                <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
                <xsd:element name="personalNumber" type="tns:PersonalNumberType"
/>
                <xsd:element name="userVisibleData" type="xsd:string" />
                <xsd:element name="userNonVisibleData" minOccurs="0"
type="xsd:string" />
                <xsd:element name="provider" minOccurs="0" type="xsd:string />
                <xsd:element name="endUserInfo" minOccurs="0" type="xsd:string"
/>
                <xsd:element name="requirementAlternatives" minOccurs="0"
type="xsd:string" />

        </xsd:sequence>
</xsd:complexType>
```

Element description

Name	Data type	Occurs	Comment	Optional
policy	xsd:string	[1..1]	Your customer specific ServiceID	
displayName	xsd:string	[0..1]	The text user will se in BankID Säkerhetsapp or BankID Säkerhetsprogram.	X
transactionId	xsd:string	[0..1]	A unique ID for the transaction.	X
personalNumber	xsd:string	[0..1]	Swedish “personnummer” in 12 digits. (ååååmmddxxxx)	X
userVisibleData	xsd:string	[1..1]	The text to be displayed and signed. Must be UTF-8 encoded, the value must be Base64 encoded. (Max 40.000 characters after Base64 encoding). The text can be formatted using CR=new line, LF=new line and CRLF=new line.	

userNonVisibleData	xsd:string	[0..1]	Data that is not displayed to the user at time of signature computation. The value should be Base64-encoded. (max 200.000 characters after Base64-encoding)	X
provider	xsd:string	[1..1]	Use “bankid” (2015 “telia”)	
endUserInfo	xsd:string	[0..1]	Possibel to add the users IP-adress. IP_ADDR	X
requirementAlternatives	xsd:string	[0..1]	Use if you have specific demands on: Cardreader CertificatePolicy IssuerCN AutoStartTokenRequired	X

Generall Recommenations

displayName

We recommend you to use displayName, even if you only have one displayName to begin with. (displayName are presented for the user in BankID Säkerhetsprogram and BankID Säkerhetsapp). In test there exist two possible displayNames “Test av Mobilt BankID” and the other “Funktionstjänster Test”. The displayName should not exceed 30 characters. If you don´t have a displayName contact CGI. Delivery time for a displayName in production are 5-10 days.

Jag signerar mot: displayName

personalNumber

Usually used when BankID are on another device. But can also be used if you already know the users personalNumber and only want to allow that user to authenticate or sign.

userNonVisibleData

When this data not is presented to the user, it should be used with care and with a god correlation to userVisibleData. Use only if you have special requirements. This data is not considered any legal value.

Provider

For BankID use “bankid”.

In 2015 use “telia” for Telia

endUserInfo

Used for passing information relatated to the user and the user's computer/device. (no controls are done by BankID in the current solution). If you are able to add the user's IP-adress, we recommend you to do so.

```
<endUserInfo>
   <type>IP_ADDR</type>
   <value>192.169.0.1</value>  /*example IP-address */
</endUserInfo>
```

This function is a preparation for future extra security controls made by BankID. We recommend you to use this parameter, just to be prepared.

requirementAlternatives

We recommend that you don´t use this function unless you have very specific requirements. Only allow specific BankID certificate, specific issuer eg. (See BankID Relying party Guidelines v2.5, section 12.6

If you use OSIF:

If you use OSIF today, you should use "AutoStartToken" in your migration to GRP. Using SignatureRequest / SignatureResponse replace the OSIF call "EncodeTBS", "GenerateChallenge" and "VerifySignature" and the "Signatur2 Plug-in" call.

userVisibleData = (BISP Signer2 plug-in) TextToBeSigned

userNonVisibleData = (BISP Signer2 plug-in) NonVisibleData

SignatureFileRequest – *deprecated end of life June 2016*

General Recommendations.

SignatureFileRequest are only possible on "BankID Säkerhetsprogram" and not on "BankID Säkerhetsapp". If you in OSIF used the "file signing method", you should use SignatureFileRequest instead of SignatureRequest.

With SignatureFileRequest you can add a .pdf or .txt document in the signing process.

We do not recommend this file signing method unless you already using this method in existing BISP solutions. BankID has announced end of life for the file signing function June 2016.. Other file signing methods exist, e.g. DSS api or different document signing services that are more general and modern.

```
Begin
      String policy
      String displayName
      String transactionId
```

```
      personalNumber personalNumber
      String userVisibleData
      String userNonVisiableData
      String provider
      String filenamne
      String fileContent
End
```

Description

Request a digital signature.

Schema

```
<xsd:complexType name="SignRequestType">
       <xsd:sequence>
              <xsd:element name="policy" type="xsd:string" />
              <xsd:element name="displayName" minOccurs="0" type="xsd:string"
/>
              <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
              <xsd:element name="personalNumber" type="tns:PersonalNumberType"
/>
              <xsd:element name="userVisibleData" type="xsd:string" />
              <xsd:element name="userNonVisibleData" minOccurs="0"
type="xsd:string" />
              <xsd:element name="filename" minOccurs="0" type="xsd:string" />

              <xsd:element name="provider" type="xsd:string" />

       </xsd:sequence>
</xsd:complexType>
```

Element description

Name	Data type	Occurs	Comment	Optional
policy	xsd:string	[1..1]	Your customer specific ServiceID	
displayName	xsd:string	[0..1]	The text user will se in BankID säkerhetsapp. Must correspond to a FP-certificate	X
transactionID	xsd:string	[0..1]	A unique ID for the transaction.	X
personalNumber		[0..1]	Swedish “personnummer” in 12 digits. (ååååmmddxxxx)	X
userVisibleData	xsd:string	[1..1]	The text to be displayed and signed. Must be UTF-8 encoded, the value should be Base64 encoded. Max 100 Kb (after Base64 encoding)	
userNonvisibleData	xsd:string	[0..1]	Data that is not displayed to the user at time of signature computation. The value	X

			should be Base64-encoded. Max 5Mb (after Base64-encoding)	
provider	xsd:string	[1..1]	Use “bankid”	
endUserInfo	xsd:string	[0..1]	Possibel to add the users IP-adress. IP_ADDR	X
requirementAlternatives	xsd:string	[0..1]	Use if you have specific demands on: Cardreader CertificatePolicy IssuerCN AutoStartTokenRequired	X
filename	xsd:string	[1..1]	String. Base 64-encoded. 5-340 characters. Printable ASCII-characters (32-127). The name must end with dot (“.”) and a suffix that describes the type of file. Allowed .pdf and .txt	
fileContent		[1..1]	MTOM/XOP-binary data. The filecontent is sent as binary attachment outside the SOAP-envelope using XML-binary optimized packing (XOP)	

CGI eID service

displayName

We recommend you to use displayName, even if you only have one displayName to begin with. (displayName are presented for the user in BankID Säkerhetsprogram and BankID Säkerhetsapp). In test there exist two possible displayNames “Test av Mobilt BankID” and the other “Funktionstjänster Test”. The displayName should not exceed 30 characters. If you don´t have a displayName contact CGI.

personalNumber

Usually used when BankID are on another device. But can also be used if you already know the users personalNumber and only want to allow that user to authenticate or sign.

userNonVisibleData

When this data not is presented to the user, it should be used with care and with a god

correlation to userVisibleData. Use only if you have special requirements. This data is not considered any legal value.

Provider

Use “bankid”.

endUserInfo

Used for passing information relatated to the user and the user’s computer/device. (no controls are done by BankID in the current solution). If you are able to add the user’s IP-adress, we recommend you to do so.

```
<endUserInfo>
   <type>IP_ADDR</type>
   <value>192.169.0.1</value>  /*example IP-address */
</endUserInfo>
```

requirementAlternatives

We recommend that you don´t use this function unless you have very specific requirements. Only allow specific BankID certificate, specific issuer eg. (See BankID Relying party Guidelines, section 11.6

If you use OSIF:

If you today use OSIF, you should use “AutoStartToken” in your migration to GRP. Using SignatureRequest / SignatureResponse replace the OSIF call “EncodeTBS”, “GenerateChallenge” and “VerifySignature” and the “Signatur2 Plug-in” call.

userVisibleData = (BISP Signer2 plug-in) TextToBeSigned
userNonVisibleData = (BISP Signer2 plug-in) NonVisibleData
filename = (BISP Signer2 plug-in) FileName
fileContent = (BISP Signer2 plug-in) FileContent

..

OrderResponse

General Information

AutenticationRequest, SignatureRequest and SignaturaFileRequest all gives the same OrderRespons

```
Begin
      String transactionId
      String orderRef
      Strinf AutoStartToken
End
```

Description

Schema

```
<xsd:complexType name="AuthenticateResponseType">
        <xsd:sequence>
                <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
                <xsd:element name="orderRef" type="xsd:string" />
                <xsd:element name="AutostartToken" type="xsd:string" />

        </xsd:sequence>
</xsd:complexType>
```

Element description

Name	Data type	Occurs	Comment
transactionId	xsd:string	[1..1]	An ID for the transaction. If you used transactionId in the request, it will be that parameter. Otherwise an transactinId will be created by GRP service.
orderRef	xsd:string	[1..1]	An identifier to be used later in the Collect method to collect the autentificate response and to query status. Save this on the server unique for the users session. UUID-string 36-50 characters
AutoStartToken	xsd:string	[1..1]	Use for launching ”BankID Säkerhetsprogram” or ”BankID Säkerhetsapp” UUID-string 36-50 characters

General Recommendation

AutoStartToken

Are used when BankID are on the same device and “peronalNumber” were not present in the request.
Using AutoStartToken to launch “BankID Säkerhetsprogram” or “BankID Säkerhetsapp” See BankID Relying Party Guidelines, section 3 Launching.

If you use OSIF:

If you use OSIF today, you should use “AutoStartToken” in your migration to GRP. Using AutenticationRequest / AutenticationResponse replace the OSIF call “GenerateChallenge” and “VerifyAutentication” and the “Authentication Plug-in” call.

CollectRequest

```
Begin
      String policy
      String transactionId
```

```
      String orderRef
      String displayName
End
```

Description

Collect query status and the authentication/signature.

Schema

```
<xsd:complexType name="CollectRequestType">
       <xsd:sequence>
              <xsd:element name="policy" type="xsd:string" />
              <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
              <xsd:element name="displayName" minOccurs="0" type="xsd:string"
/>
              <xsd:element name="orderRef" type="xsd:string" />
              <xsd:element name="displayName" type="xsd:string" />

       </xsd:sequence>
</xsd:complexType
```

Element description

Name	Data type	Occurs	Comment	Optional
policy	xsd:string	[1..1]	Your customer specific ServiceID	
transactionId	xsd:string	[0..1]	An ID for the transaction.	X
orderRef	xsd:string	[1..1]	The identifier returned by autentificateResponse or signatureResponse.	
displayName	xsd:string	[0..1]	Must be used if you have more than one displayName.	X

General Recommendation

For use case BankID on the same device:

After you received a orderRef you can start BankID Säkerhetsprogram or BankID Säkerhetsapp. See reference how to launch BankID programs on PC and mobile devices.

The automatic start can fail due to different reason:
* The user has not installed the BankID app
* Error in start command
* User did not allow the browser to launch the URL
The web browser will inform the user that the URL cannot be opened. Status START_FAILED will be delivered.

You can make a delay in 3 seconds before you start to call the Collect method.

After 3s you begin to make *CollectRequest* with a 3 seconds delay between each collect. The first answer will probably be USER_SIGN and then a COMPLETE. It is also possible to get OUTSTANDING_TRANSACTION, or after some time NO_CLIENT. In that case ask the user to start BankID client.

For use case BankID on another device:

After you have received a orderRef, you will inform the user to start BankID App "Starta BankID-programmet"

You can make a delay for 9 seconds before you start to call the Collect method.

After 9s you begin to make *CollectRequest* with a 3 seconds delay between each collect. First answer will probably be OUTSTANDING_TRANSACTION, then USER_SIGN and at last COMPLETE. (After six OUTSTANDING_TRANSACTION you get NO_CLIENT, just to indicate that the user not yet has started her BankID client.

After 180 seconds, you will at last get faultStatus EXPIRED_TRANSACTION

Please make notice of the different user messages that you can display to the user:
1. OUTSTANDING_TRANSACTION (no BankID client has yet received the order).
   "Starta BankID-programmet" or "Försöker starta BankID-programmet" if AutoStartToken were used.
2. NO_CLIENT "Starta BankID-programmet"
3. USER_SIGN (The BankID client has received the order)
   "Skriv in din säkerhetskod i BankID-programmet och välj Legitimera eller Skriv under"
4. COMPLETE (User has provided the security code and the order are compleated)
5. START_FAILED "BankID-programmet verkade inte finnas i din dator eller telefon.
   Installera det och hämta ett BankID hos din bank"
6. EXPIRED_TRANSACTION "BankID-programmet svarar inte. Kontrollera att det är startat och att du har internetanslutning. Försök sedan igen"

If you used displayName in the Authentication- or SignatureRequest, you must use the same displayName in the CollectRequest. If you have access to different displayName it is important that you use this parameter also in the CollectRequest.

CollectResponse

```
Begin
      String transactionId
      progressStatus progressStatus
      String signature
      Property attributes
End
```

Description

Schema

```
<xsd:complexType name="CollectResponseType">
       <xsd:sequence>
              <xsd:element name="transactionId" minOccurs="0"
type="xsd:string" />
```

```
                <xsd:element name="progressStatus" type="tns:ProgressStatusType"
/>
                <xsd:sequence minOccurs="0">
                        <xsd:element name="signature" type="xsd:string" />
                        <xsd:element name="attributes" minOccurs="0"
maxOccurs="unbounded" type="tns:Property" />
                        <xsd:element name="userInfo" type="xsd:string" />
                </xsd:sequence>
        </xsd:sequence>

</xsd:complexType>
```

Element description

Name	Data type	Occurs	Comment	Optional
transactionId	xsd:string	[1..1]	Identity of the transaction.	
progressStatus		[1..1]	OUTSTANDING_TRANSACTIOON, USER_SIGN, NO_CLIENT, STARTED, or COMPLETE	
signature	xsd:string	[0..1]	A XML signature (XMLDsig) that you should archive according to your organizations rules.	Only if signature
attributes	property	[0..unbounded]	A number of user attributes	

CGI eID Service
You can only collect COMPLETE status once. If you try to make an Collect on a orderRef that already resulted in COMPLEATE, you get Error code INVALID_PARAMETERS

Interpret user information

Attributes

Names that are used in attributes. You will recognise the same attributes in your OSIF implementation.

General information
The name of the user Attribute the GRP service give you are the same as used in OSIF specification and are also used in MBI. You will also recognize the naming as name in the client certificate.

Name	**Description**	**Comment**
cert.subject.cn	The given name and surname of the user *OSIF Subject.CommonName*	
cert.subject.givenname	All given names for the user *OSIF: Subject.GivenName*	
cert.subject.surname	Surename of the user. *OSIF: Subject.Surname*	
cert.subject.serialnumber	Swedish social security number (personnummer) *OSIF: Subject.SerialNumber*	
cert.serialnumber	The serial number of the users certificate. Unique for the used certificate and god to have in forensic purpose. *OCSP: SerialNumber*	
cert.notafter	Date when the users certificate expires. Can be used to alert the user to soon get a new. *OCSP: Expiredate*	
cert.notbefor	Creation day for the users certificate. Newly produced certificate have a higher risk of ID theft. *OSIF: notBefor*	
ipAddress	The IP-adress of the user agent as the BankID server discovers it. You can use this for extra validation against the IP-adress you have (used in endUserInfo in the request). UserInfoType - ipAddress	
security.level	Level 3 and 4 are used today.	

	3: used for “BankID på fil” and “Mobilt BankID” 4: used for “BankID på kort” Ändra: Vi måste ha ett internt register med alla utgivares CN och koppla till en level samt en description.	
securitylevel.description	Authentication Context for the OASIS Security Assertion Markup Language. Used today are: * SoftwarePKI * SmartcardPKI * MobileTwofactorContract	
validation.ocsp.response	The electronic receipt of the validation. String (b64) XML-signature signed by a certificate that has the same issuer as the certificate being verified. Should be saved for evidence purpose.	
cert.issuer.cn	Common Name of the issuer in the user certificate. *OCSP: Issuer.CommonName*	
cert.issuer.o	Organization Name of the issuer in the user certificate. *OCSP: Issuer.OrganizatonName*	
cert.issuer.c	Country of the issuer. SE *OCSP: Issuer.CountryName*	

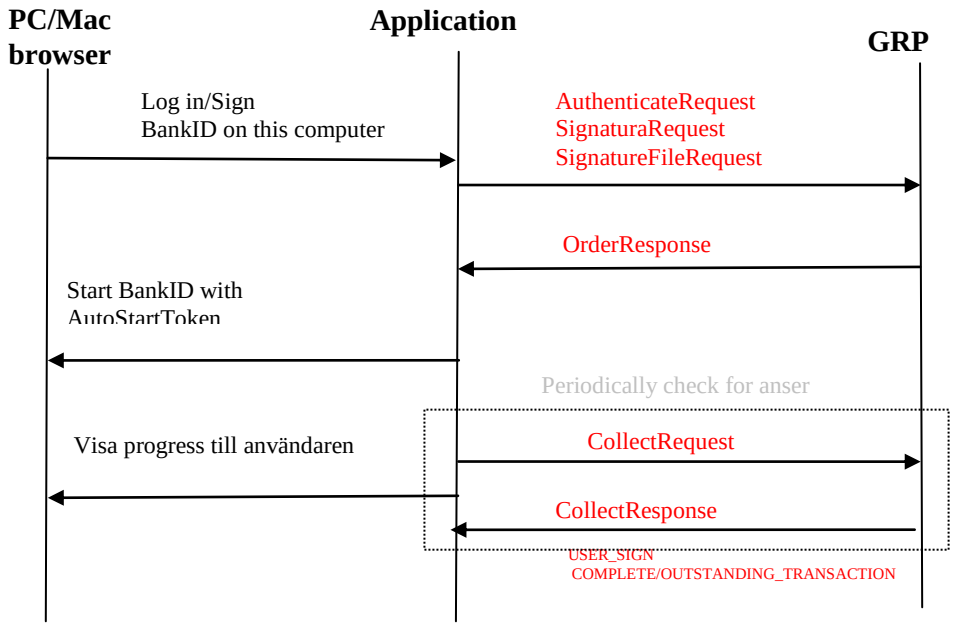
General information

For “Svensk e-legitimatin” are the following attributes normal of intrest.

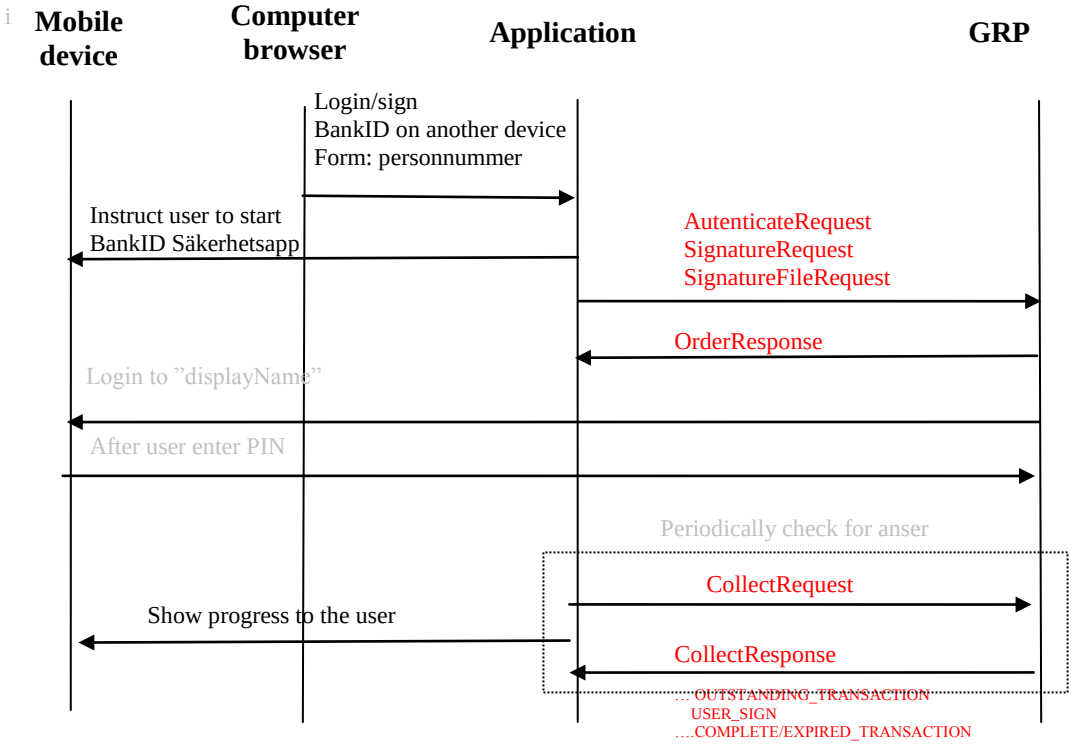
Attribut	*description*	*Example*
cert.subject.cn	Name	Agda Andersson
cert.subject.givenName	All given names	Agda
cert.subject.surname	Surname	Andersson
cert.subject.serialnumber	“personnummer”	188893099368

Basic flow

BankID on the same unit



BankID on another unit



Error handling

GRP´s error handling follows normal SOPA standards in contrast to the error handling in OSIF.

It is recommended to show *detaildDescription* and Errorcode "*faultStatus*" to the user. It could also be useful to print out time, *transactionId* or other useful information in a support situation.

WSDL

```
https://grpt.funktionstjanster.se:18898/grp/v1

Test:
https://grpt.funktionstjanster.se:18898/grp/v1
```

GrpService

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's
 version is JAX-WS RI 2.1.10-hudson-jaxws-ri-2.1.x-release-5-.
 -->
- <!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's
 version is JAX-WS RI 2.1.10-hudson-jaxws-ri-2.1.x-release-5-.
 -->
- <definitions xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
    1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://logic.grp.mobilityguard.com/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://logic.grp.mobilityguard.com/"
    name="GrpService">
  <import namespace="http://funktionstjanster.se/grp/service/v1.0.0/"
    location="http://eidt.funktionstjanster.se:18899/grp/v1?wsdl=1"
    />
- <binding xmlns:ns1="http://funktionstjanster.se/grp/service/v1.0.0/"
    name="GrpServiceServletPortBinding"
    type="ns1:GrpServicePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
    style="document" />
- <operation name="Sign">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
    </input>
- <output>
  <soap:body use="literal" />
    </output>
- <fault name="GrpFault">
```

```
  <soap:fault name="GrpFault" use="literal" />
    </fault>
    </operation>
- <operation name="Collect">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
    </input>
- <output>
  <soap:body use="literal" />
    </output>
- <fault name="GrpFault">
  <soap:fault name="GrpFault" use="literal" />
    </fault>
    </operation>
- <operation name="FileSign">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
    </input>
- <output>
  <soap:body use="literal" />
    </output>
- <fault name="GrpFault">
  <soap:fault name="GrpFault" use="literal" />
    </fault>
    </operation>
- <operation name="Authenticate">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
    </input>
- <output>
  <soap:body use="literal" />
    </output>
+ <fault name="GrpFault">
    </operation>
    </binding>
- <service name="GrpService">
+ <port name="GrpServiceServletPort"
    binding="tns:GrpServiceServletPortBinding">
    </service>
    </definitions>
```

Reference

For BankID some part of document “BankID Relying Party Guidelines” are god to read. Following parts are directly taken from that documentation.

You can also find that documentation on http://www.bankid.com/rp/info/

Launching

Launching the BankID app from a browser

When the BankID app is installed the schema “bankid” is registered in the operating system. When the bankid schema is requested from the browser the operating system launches the BankID app. The URL works in Android, iOS and Windows Phone 8 when the built-in web browser is used. The URL works in PCs with all commonly used browsers. Some differences exist on different platforms.
For PCs the URL syntax is:

```
bankid:///?autostarttoken=[TOKEN]&redirect=[RETURNURL]
```

Note that the redirect parameter must be last in the parameter list. The autostarttoken are optional.
Note that the parameter names must be lower case.

How to Start

We recommend using the following technique to start the BankID app from a browser.
1. Make the call to the web service, check the result.
2. **Try** to start the Bankid app in a frame with zero size
. <iframe src="bankid:///?autostarttoken=[TOKEN]&redirect=[RETURNURL]" height=0 width=0></iframe>"
If the BankID app is not installed the error presented by the browser will be in the zero size window and not visible for the user.
3. After 5 seconds display a link or button with the BankID URL. Use text RFA18.
<a href="bankid:///?autostarttoken=[TOKEN]&redirect=[RETURNURL]”>[RFA18]
Using a href this way is the most reliable method to start the BankID app.

If the start in 2 is successful, the link in 3 will be hidden by the BankID app. If the start in 2 is unsuccessful, the user will have a second chance to start. If the BankID app is started but no matching call to Authenticate or Sign has been done, an error message will be displayed in the app. To avoid this, we recommend that the link or button is inactivated immediately after the user presses it.

RFA18 = “Starta BankID-programmet”

Behavior in Different Browsers

Chrome

In current version of Chrome (version 31) a known problem blocks BankID from being started from a frame. Using the recommended method above solves the problem, see https://code.google.com/p/chromium/issues/detail?id=318788.

Internet Explorer

Internet Explorer manipulates the URL in the redirect parameter. In this specification we state that the RETURNURL must be URL-encoded. However, Internet Explorer decodes the content

prior passing it to the BankID app. This is why it must be last in the list of parameters. In the same way, Internet Explorer may decode the content of the RETURNURL when the BankID app passes the return URL back to the browser. If the RP includes session information that is affected by URL-encoders/decoders problems may occur. It is recommended to use only URL-encoding safe characters in the parameters.

Parameters in the start URL

Parameter	Description
autostarttoken	Optional. Holds the autoStartToken that was returned from the web service call. If the user ID number was not included in the web service call the autostarttoken must be provided. We strongly recommend to always use the autostarttoken when the URL is used to start the client. If it is not included and the user reloads the page or if the page erroneously repeats the start command the user may get an error claiming that the BankID is missing. The likelihood of this to happen is reduced if autostarttoken is used.
redirect	Mandatory. The BankID app uses the request parameter redirect to launch the RP web app after completed (including cancelled) authenticate or sign. The redirect URL must be UTF-8 and URL encoded and should match the web address the user is visiting when RP web app launches the BankID app. It may include parameters to be passed to the browser. For iOS and Windows Phone the redirect must have a value, but for all other platforms it may be empty (“redirect=”), or set to “null” (“redirect=null”). If it is empty or null the BankID app will terminate without launching any URL and the calling application will be in focus. **The general recommendation is to use redirect=null when it is possible.** **Note for Windows and Mac OS X** If redirect has a value the redirect parameter must be used together with autostarttoken. If autostarttoken is excluded, the content of redirect will be ignored and the behavior will be as if redirect=null. **Note for Android** If the user has several browsers installed on an Android device the user is sometimes presented with a question asking what browser to use. BankID recommends that redirect=null is used on Android. This ensures the user will return to the browser previously used. **Note for Windows Phone** When the browser on Windows Phone is started from an app it is considered a new session by the browser, hence any previous transient (session) cookies are unavailable. RP can use a persistent cookie or the RETURNURL to control the session.

Examples

The RP wants the BankID app to open a browser with the following URL after finishing execution:
https://demo.bankid.com/nyademobanken/CavaClientRedirReceiver.aspx?orderRef=bedea56d-7b46-47b1-890b-f787c650bc93&returnUrl=./CavaClientAuth.aspx&Environment=Kundtest.
The autostarttoken is included. The start URL is:

bankid:///?autostarttoken=a4904c4c-3bb4-4e3f-8ac3-0e950e529e5f&
redirect=https%3a%2f%2fdemo.bankid.com%2fnyademobanken%2fCavaClientRedirReceiver.
aspx%3forderRef%3dbedea56d-7b46-47b1-890b-
f787c650bc93%26returnUrl%3d.%2fCavaClientAuth.aspx%26Environment%3dKundtest

Launching the BankID app from Native App on Mobile Device

Android

```
Intent intent = new Intent();
intent.setPackage("com.bankid.bus");
intent.setAction(Intent.ACTION_VIEW);
intent.addCategory(Intent.CATEGORY_BROWSABLE); //optional
intent.addCategory(Intent.CATEGORY_DEFAULT); //optional
intent.setType("bankid");
intent.setData(Uri.parse("bankid:///?autostarttoken=<INSERT
AUTOSTARTTOKEN HERE>&redirect=null ")) ;
startActivityForResult(intent, 0);
```

In Android, the RP app does not need to register a URL scheme to be successfully re-launched by the BankID app. `onResume()` will be called when RP app is re-launched.
If the BankID app is not present on the device an `android.content.ActivityNotFoundException` is thrown. RP must inform the user. Message RFA2 should be used.

RFA2 = “Du har inte BankID-appen installerad. Kontakta din bank”

iOS

```
openURL:[NSURL URLWithString: @”bankid:///?
autostarttoken=<INSERT AUTOSTARTTOKEN
HERE>&redirect=fp_app_x://”]
```

If the BankID app is not present on the device `NO` is returned. RP must inform the user. Message RFA2 should be used.
The RP app must register a unique URL scheme to make it possible for the BankID app to re-launch RP app. In Xcode select the project and in the Info tab expand URL Types and add the URL scheme rp_app_x.
Note: rp_app_x is an example. The RP should use its own unique URL scheme.
The RP must implement the following function that will be called when the RP app is re-launched. –

```
(BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication
annotation:(id)annotation
```

RFA2 = “Du har inte BankID-appen installerad. Kontakta din bank”

Windows Phone

```
// Create the URI string
var uriToLaunch = string.Format(
      "bankid:///?www.bankid.com/autoStartToken={0}&redirect={1}
",
      <INSERT AUTOSTARTTOKEN HERE>,
      Uri.EscapeDataString("fp-app-x://bank_x"));
// Create the URI to launch from a string.
var uri = new Uri(uriToLaunch);
// Launch the URI.

bool success = await
Windows.System.Launcher.LaunchUriAsync(uri);
```

If the BankID app is not present on the device the operating system presents a dialogue asking to open Windows Phone store. RP must inform the user. Message RFA2 should be used. (RFA2 “Du har inte BankID-appen installerad. Kontakta din bank”)
The RP app must register a unique URL scheme to make it possible for the BankID app to re-launch RP app. In Visual Studio:

1. Open Package.appxmanifest
2. Open the tab Declarations.
3. Add a "Protocol". Under name enter ”rp_app_x”.
4. Enter a logo and a "Display name".

Note: rp_app_x is an example, RP should use its own unique URL scheme.

RP must also implement the following to be successfully re-launched by BankID Security App. In Visual Studio add the class AssociationUriMapper:

```
/// <summary>
/// The association uri mapper.
/// </summary>
internal class AssociationUriMapper : UriMapperBase
{
      /// <summary>
      /// When overridden in a derived class, converts a
requested uniform resource identifier (URI) to a new URI.
      /// </summary>
      /// <returns>
      /// A URI to use for the request instead of the value in
the <paramref name="uri"/> parameter.
      /// </returns>
      /// <param name="uri">The original URI value to be mapped
to a new URI.</param>
      public override Uri MapUri(Uri uri)
      {
            var tempUri =
      System.Net.HttpUtility.UrlDecode(uri.ToString());
            // URI association launch.
            if (tempUri.StartsWith("/Protocol"))
            {
                  // Here we can redirect to the correct page,
            but for now we don't
                  return new Uri("/MainPage.xaml",
            UriKind.Relative);
                  }
                  // Otherwise perform normal launch.
                  return uri;
            }

}
```

In App.xaml.cs, add AssociationUriMapper as UriMapper by adding the following line to the method InitializePhoneApplication:

```
// Assign the URI-mapper class to the application frame.
```

RootFrame.UriMapper = new AssociationUriMapper();

Test Enviroment (How to get BankID test certificate.)

Test version of BankID Security App for Android	http://www.bankid.com/rp/info
Installation instructions for Android	http://www.bankid.com/rp/info
Test version of BankID Security App for iOS	Uninstall any existing version of BankID Security App. Install BankID Security App from App Store. In Settings → BankID → Developer → Server enter businternal.test.bankid.com. BankID Security App will now connect to the test server. Please note that the app must be uninstalled to undo the change.
Test version of BankID Security App for Windows Phone 8	Uninstall any existing version of BankID Security App. Install BankID Security App from Windows Phone Store. Start BankID Security App, select Settings → Developer → Server and enter businternal.test.bankid.com. Save, exit BankID Security App and launch again. BankID Security App will now connect to the test server. Please note that the app must be uninstalled to undo the change.
Test version of BankID Security Application for PCs (Windows and OS X)	To be able to use the client for test you must configure it for test. If you change configuration your existing BankID:s may be blocked. Follow the backup-restore procedure to avoid that. Start by backing up the configuration for production: 1. Stop BankID Security Application (BISP). 2. Open BISP configuration folder (see below). 3. Copy all files in the folder to a separate location to be able to restore the production configuration later on. Create a set-up for test: 1. Stop BISP. 2. Open configuration folder. 3. Delete all files in the folder. 4. Create a plain text file in the folder named ”CavaServerSelector.txt”, containing the text ”kundtest”. The content must be plain text. The file may be created using notepad or the Terminal.app. Switch between production and test later on: 1. Stop BISP. 2. Replace files in the configuration folder with the earlier copied files for the desired environment. Location of BISP configuration folder □ Windows: %appdata%\BankID\Config □ OS X: /Users/<användare>/Library/Application Support/BankID/Config IMPORTANT: Only a BankID that is enrolled in the present environment can be used or administered in the present environment. Trying to use or administer a BankID from the other environment will block it.

BankID for test	Perform step 1-4 above and verify that your web service client can successfully communicate with the RP Interface. Verify that you have BankID Security App configured for test (iOS, WP8, Windows, OS X) or that you use a test version of BankID Security app (Android). Use a production or test BankID and log in to https://demo.bankid.com/nyademobanken. Note that if you configured the client for test, you cannot use a production BankID and vice versa. Select "Hämta BankID för test" and click the "Hämta BankID" button below the name in order to request a BankID for test. In the next step you can enter the ID ("personnummer") and name ("Namn") for the test BankID and request a Mobile BankID or a BankID on file. This will open a pop-up window. Your browser must not block pop-up windows. Follow the instructions in the pop-up window. If you are unable to follow the instructions above, please contact teknikinfo@bankid.com.
Network information test	The BankID app for mobile devices for test connects to the BankID server on the IP address 194.242.109.185 using the ports 4710, 443, 80, in the order mentioned. The BankID app for personal computers for test connects to the BankID server on the IP address 194.242.109.177 using port 443.
Network information productin	BankID Security Application for mobile devices in production connects to the BankID server on the IP address 194.242.109.204 using the ports 4710, 443, 80, in the order mentioned. BankID Security Application for personal computers in production connects to the BankID server on the IP address 194.242.109.207 using port 443.