OWASP The Open Web Application Security Project

IBWAS'10

2nd OWASP Ibero-American Web Application Security Conference

16.17 December 2010
ISCTE . Instituto Universitário de Lisboa
LISBOA . PORTUGAL

How cryptography can rescue the web

Carlos Ribeiro

Carlos.Ribeiro@ist.utl.pt

INSTITUTO SUPERIOR TÉCNICO

inesc id lisboa

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
    - back from the early 80’s: first IBM PC infection
    - but knowhow from mid 60’s

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - back from the early 80’s: first IBM PC infection
    - but knowhow from mid 60’s
  - Worms
    - Fast worms
      - Code Red [2001] (IIE servers) (8 months <0,5 Million)
      - Samy [2005] (MySpace) (20 hours; 1 Million)
    - Slow worms
      - Stuxnet [2010] (Windows, SCADA, PLC, Motor controls)

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
    - Authentication
    - Fast worms
      - Code Red [2001] (IIE servers) (8 months <0,5 Million)
      - Samy [2005] (MySpace) (20 hours; 1 Million)
    - Slow worms
      - Stuxnet [2010] (Windows, SCADA, PLC, Motor controls)

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
    - Authentication
    - Script attacks (XSS)
    - Request Forgery attacks (CSRF)
    - Confused Deputy problem

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
    - Script attacks (XSS)
    - Request Forgery attacks (CSRF)
    - Confused Deputy problem
  - Code-injection attacks
    - php injection
    - Shell injection
    - Sql injection

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
  - Code-injection attacks
    - php injection
    - Shell injection
    - Sql injection
  - Stolen credentials
    - Passwords (e.g. Dictionary attacks)

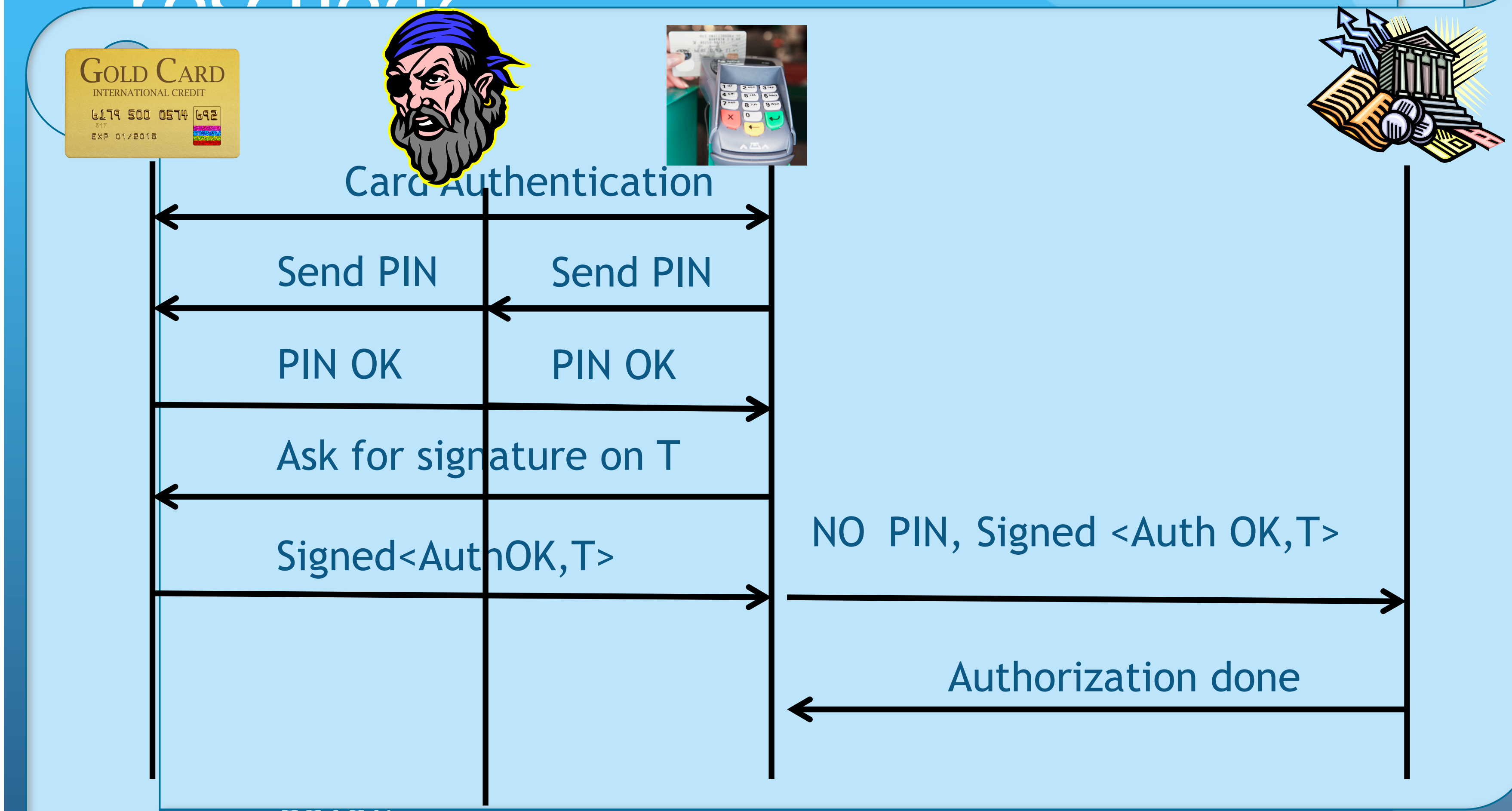
Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
  - Code-injection attacks
  - Stolen credentials
    - Passwords (e.g. Dictionary attacks)
    - Cookies (e.g. firesheep)

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
  - Code-injection attacks
  - Stolen credentials
    - Passwords (e.g. Dictionary attacks)
    - Cookies (e.g. firesheep)
    - Certificates (e.g. Stuxnet)

Why do the web needs to be rescued?



Card Authentication

Send PIN

Send PIN

PIN OK

PIN OK

Ask for signature on T

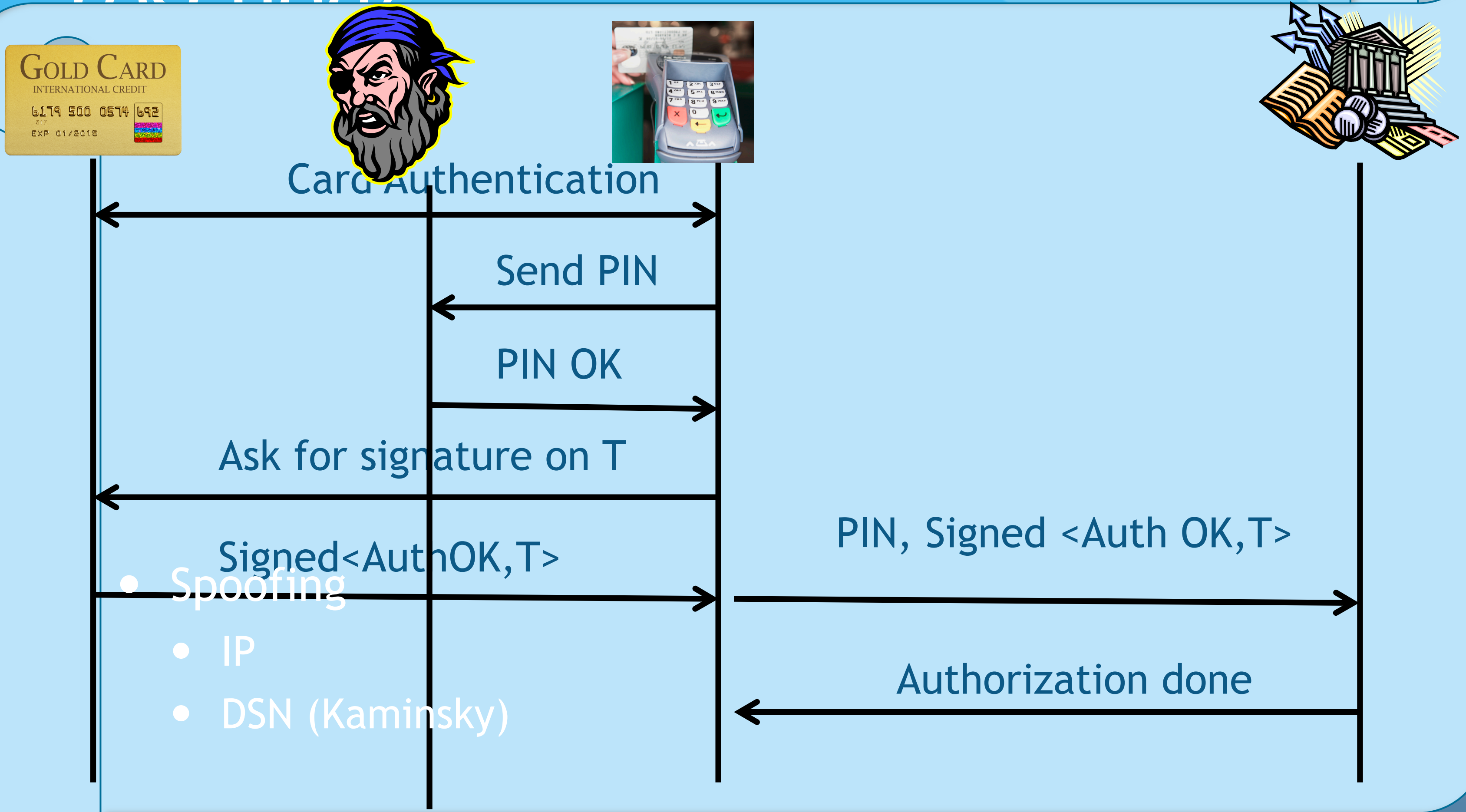
Signed<AuthOK,T>

NO PIN, Signed <Auth OK,T>

Authorization done

2010])

Why do the web needs to be rescued?



Card Authentication

Send PIN

PIN OK

Ask for signature on T

Signed<AuthOK,T>

PIN, Signed <Auth OK,T>

Authorization done

- Spoofing
  - IP
  - DSN (Kaminsky)

2010])

Why do the web needs to be rescued?

- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
  - Code-injection attacks
  - Stolen credentials
  - Spoofing
  - DDoS
    - Estonia attack 2007
  - IP
    - DSN (Kaminsky)

Why do the web needs to be Rescued?

Post Authentication

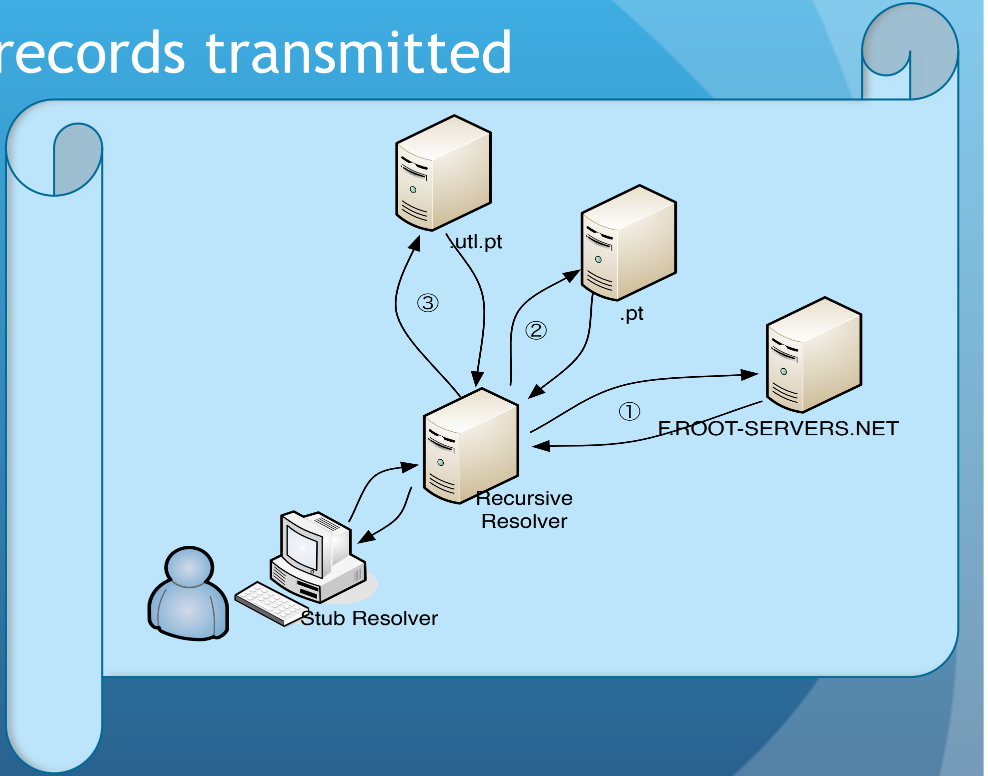
- The web is free … for all …
  - Virus
  - Worms
  - Phishing attacks
  - Cross-site
  - Code-injection attacks
  - Stolen credentials
  - Spoofing
  - DDoS
  - BotNet
    - Estonia attack 2007
    - How to use the web to run a Cmd&Ctrl

Good Authentication

- Prevents several know problems
- Big dissuasion factor
- Services authentication
  - Currently PKI with root certificates in browsers
  - Future also DNSSEC

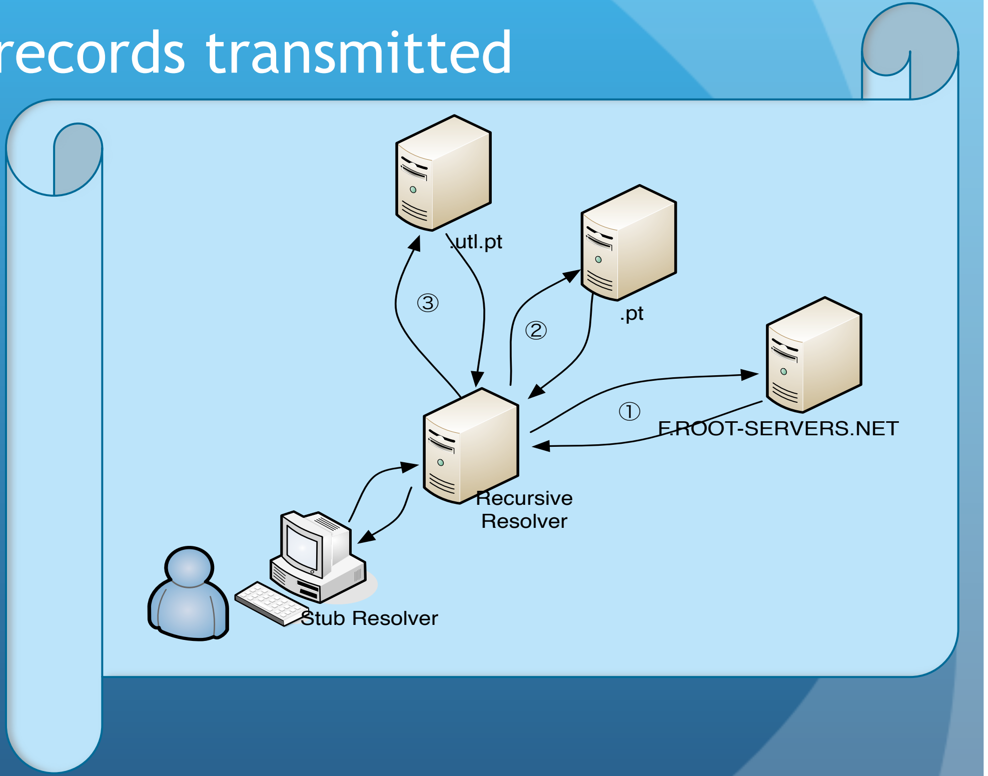
DNSSEC

- DNS Security Extensions
- Provides authentication for records transmitted between DNS resolvers
  - Root servers already signed
  - TLD domains being sign
  - No stub resolver



DNSSEC

- DNS Security Extensions
- Provides authentication for records transmitted between DNS resolvers
  - Root servers already signed
  - TLD domains being sign
  - No stub resolver



- Global PKI
  - Authenticate service names
  - Authenticate mail addresses
    - through DKIM
  - Authenticate machines
    - IPSec and SSH

What about persons?

CARTÃO DE CIDADÃO
CITIZEN CARD

PORTUGAL

DA CONCEIÇÃO ÁVILA

PAULA ANDREIA

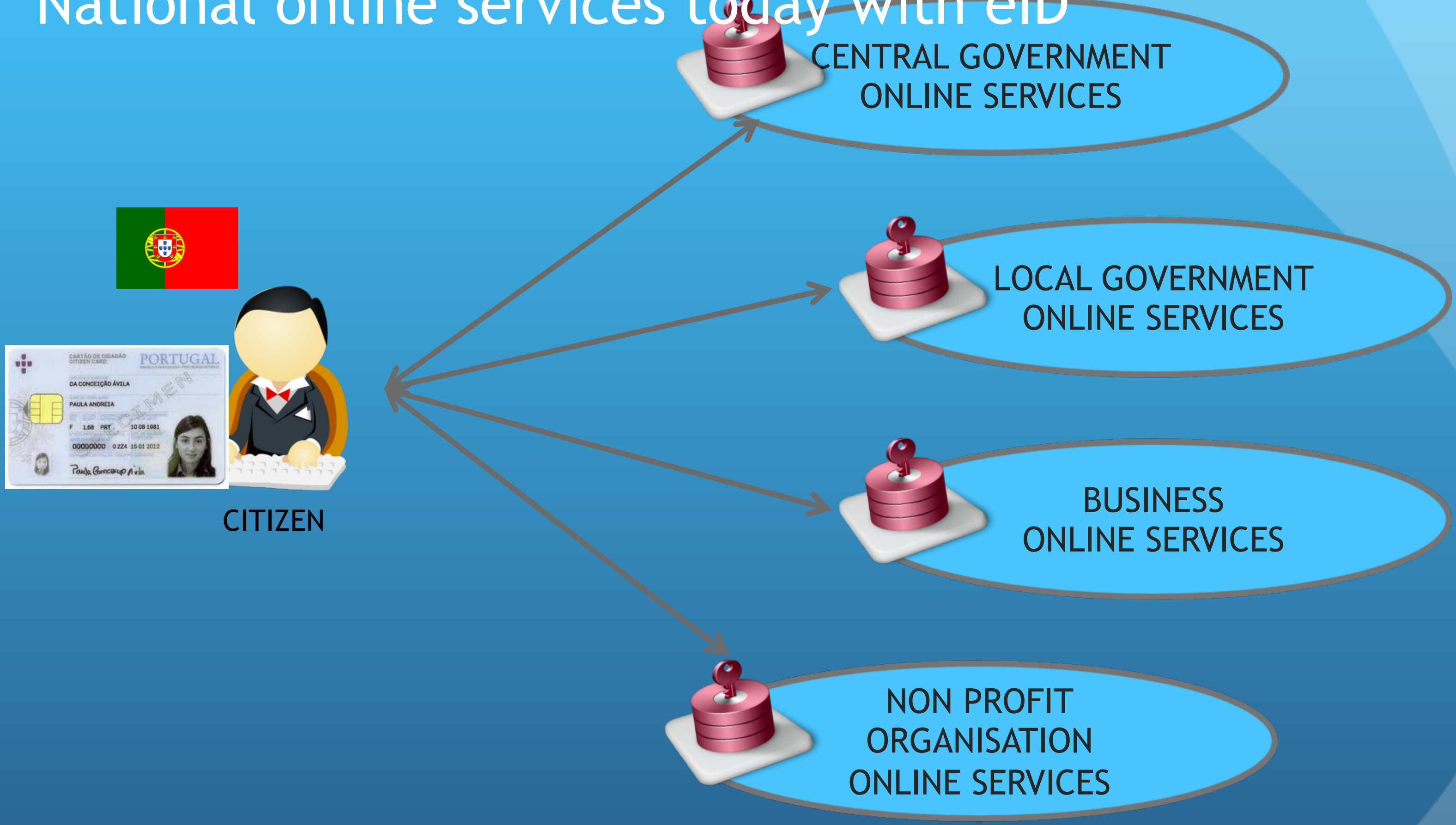
F 1,68 PRT 10 08 1981

00000000 0 ZZ4 15 01 2012

SPECIMEN

- Most sites manage their on registration services
  - Organizations use Single Sign On services
  - Some are federated through OpenID
- Persons are identified using passphrases and cookies
  - Some organizations require also tokens (e.g. Smartcard, RSASecurid)
- Financial institutions require two levels of authentication
- Every thing is very limited either in scope or in security strength
- Most countries already have or are deploying National eIDs

National online services today with eID



CITIZEN

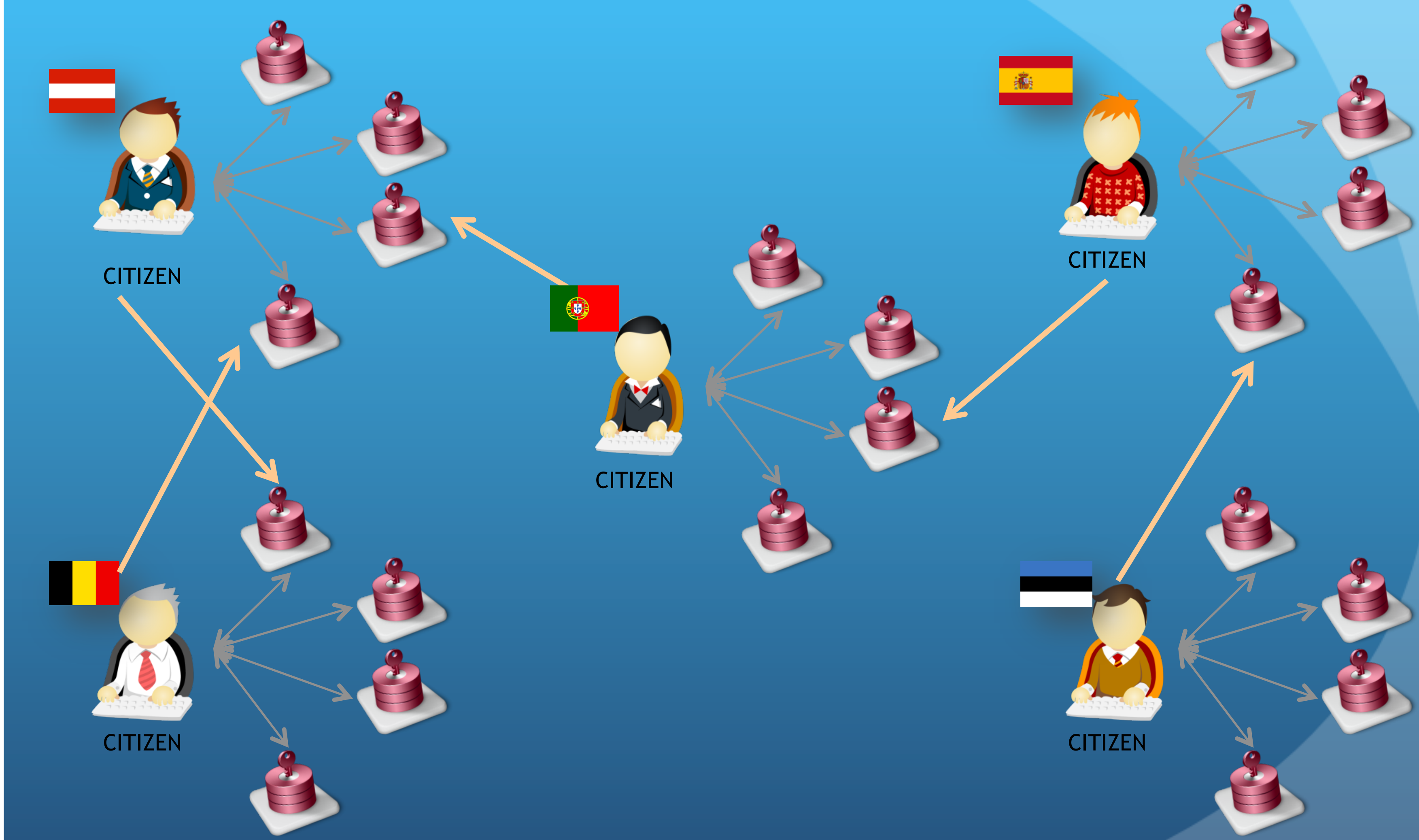
CENTRAL GOVERNMENT ONLINE SERVICES

LOCAL GOVERNMENT ONLINE SERVICES

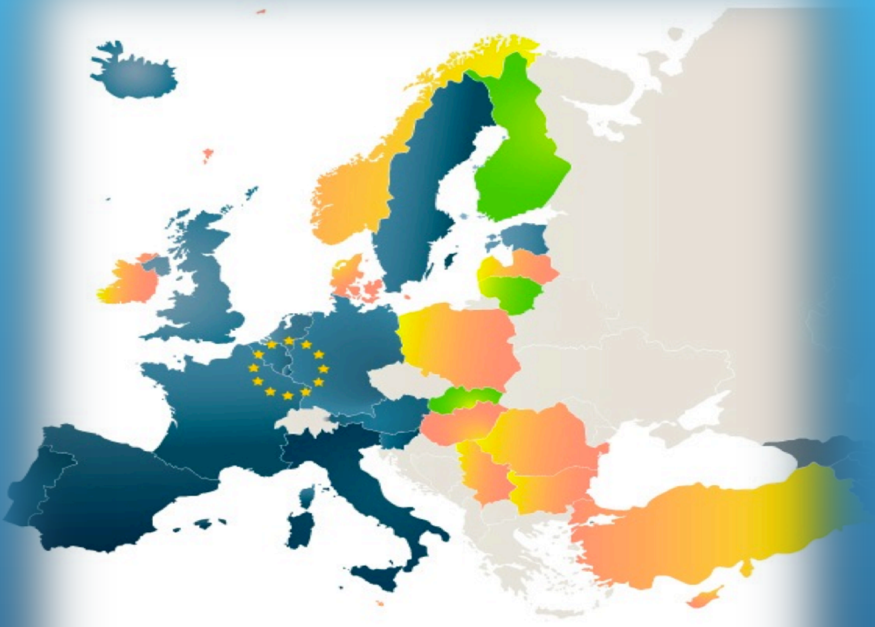
BUSINESS ONLINE SERVICES

NON PROFIT ORGANISATION ONLINE SERVICES

All Nations have their own eID infrastructure



STORK: Countries involved



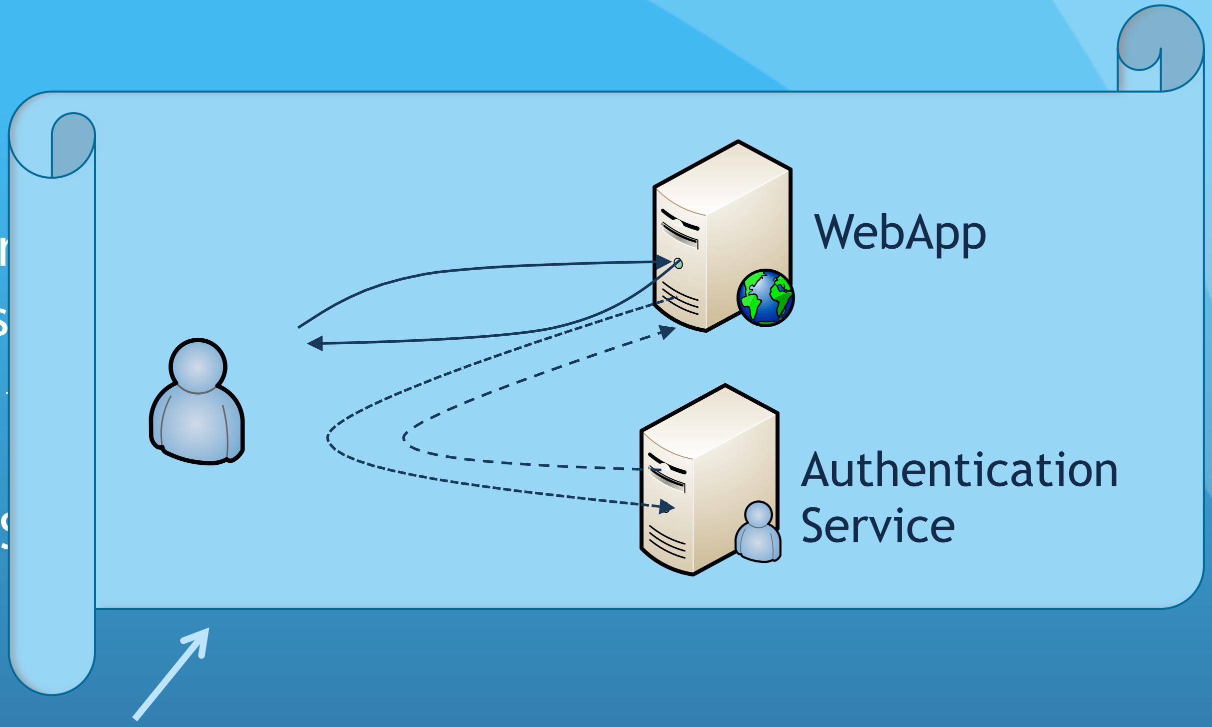
14 ORIGINAL PARTNERS

ENLARGEMENT:
3 ADDITIONAL MEMBERS

12 IN REFERENCE GROUP

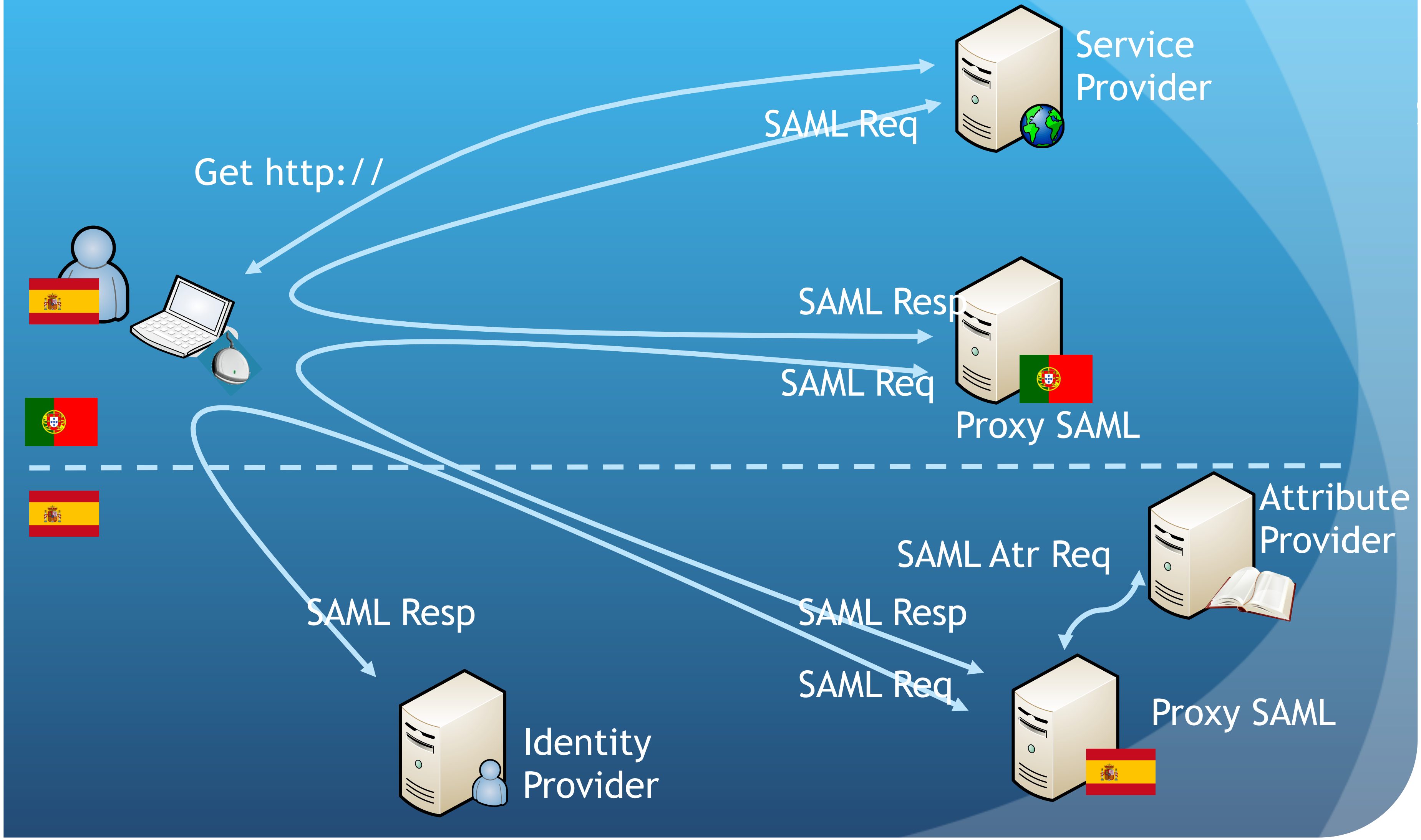
SAML 2

- Secure Asser
  - Assertions
  - Protocols -
  - Bindings -
  - Profiles - S



- Single Sign On profile
  - XML based SAML assertions
  - Over HTTPS binding
  - The authentication process depends on the Authentication Service

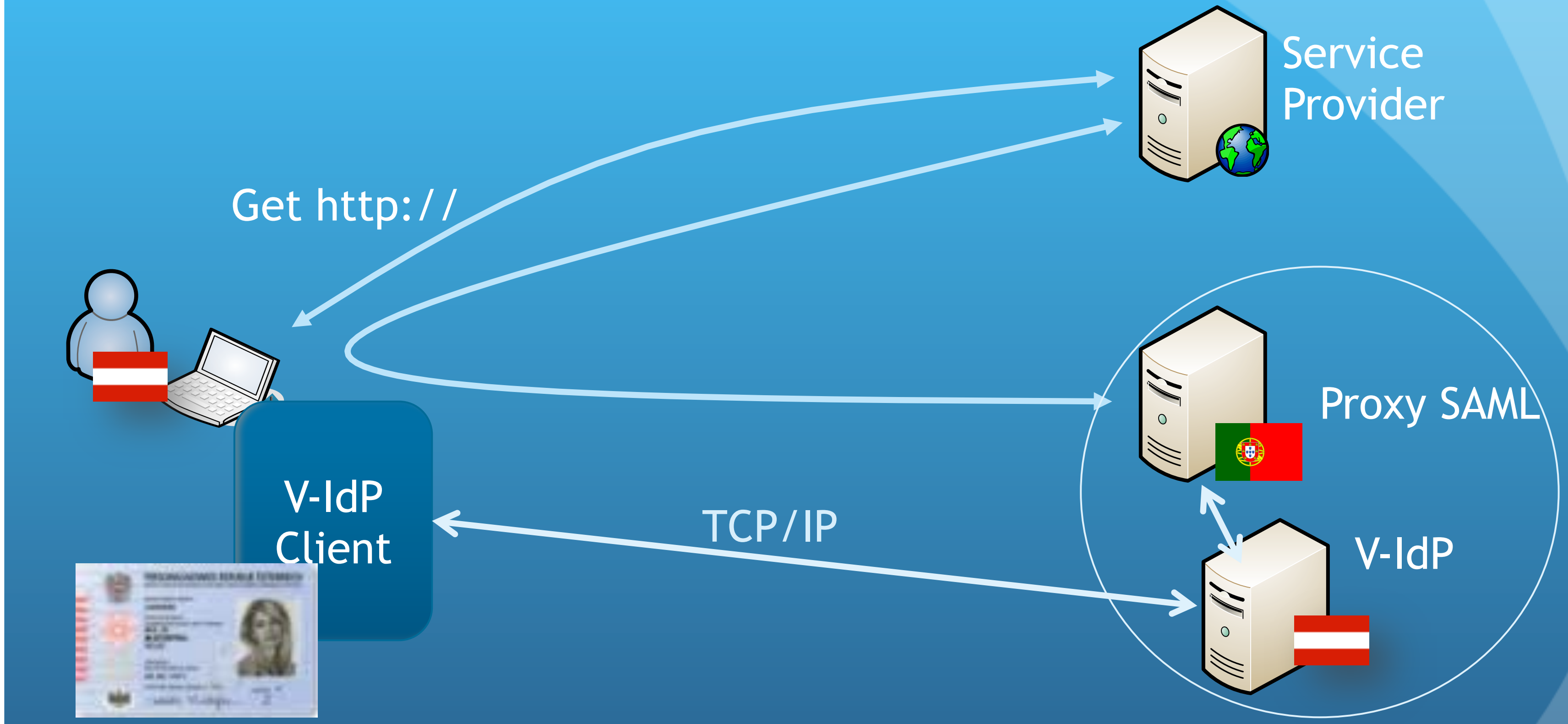
STORK Communication Architecture



Stork features

- User centric
  - Users are in control of release attributes
  - Countries may apply their regulation at Proxy level
- Privacy aware
  - An user identifier for each SP type
- Heterogeneity
  - Each Country may use it’s own identity management solution

Virtual-Identity Provider



Service Provider

Get http://

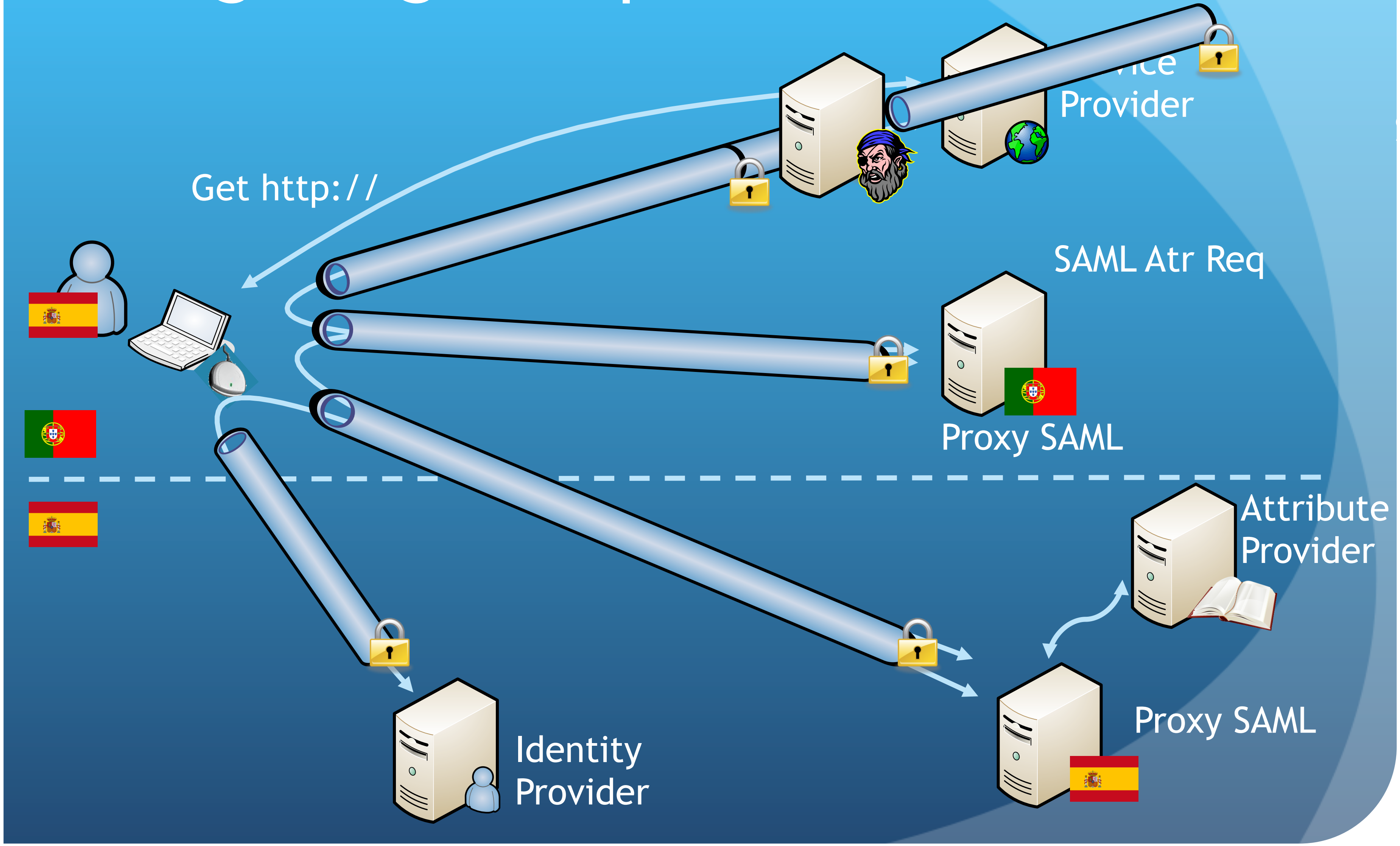
Proxy SAML

V-IdP Client

TCP/IP

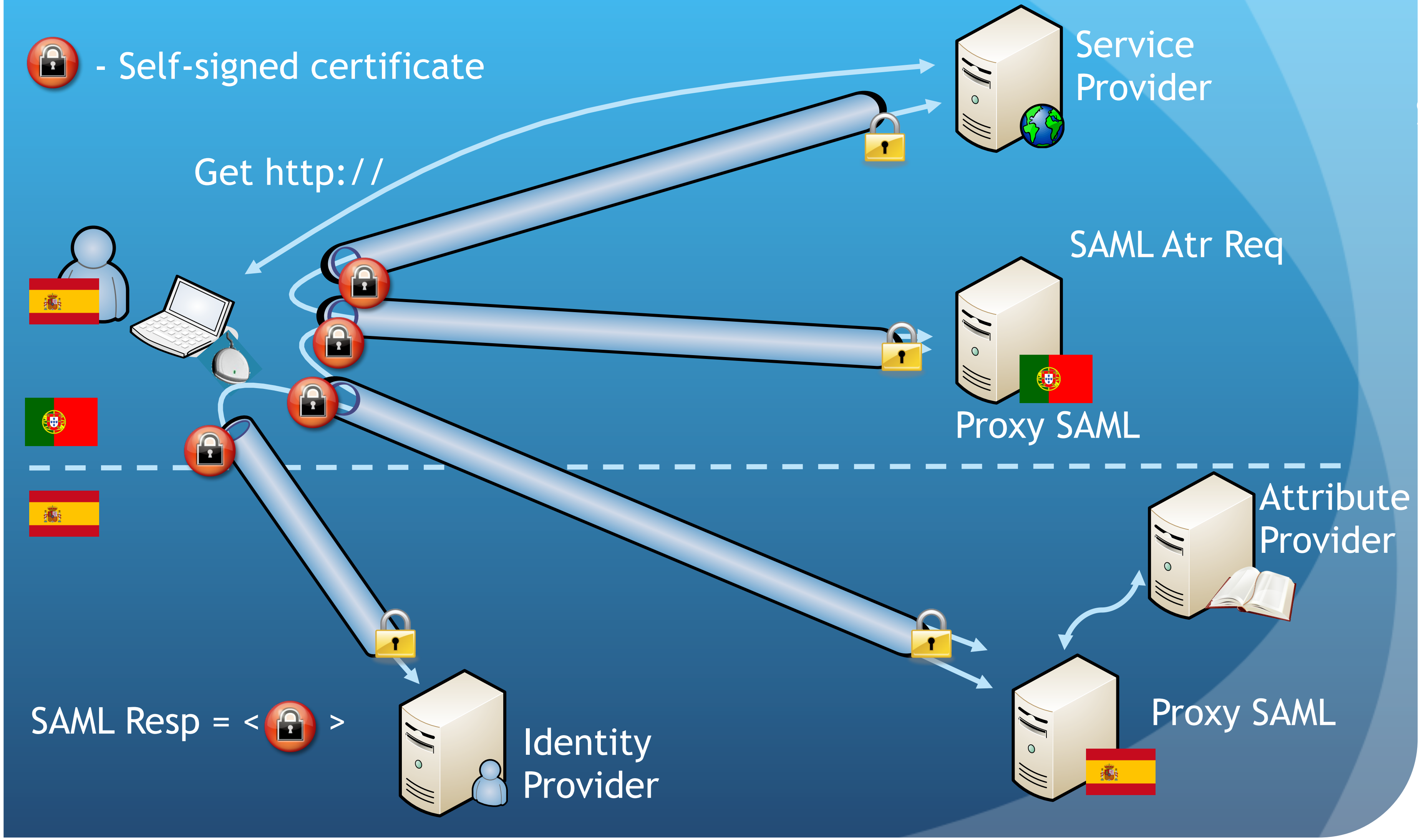
V-IdP

Single Sign On problem



Holder of key profile

🔒 - Self-signed certificate



Get http://

Service Provider

SAML Atr Req

Proxy SAML

Attribute Provider

Proxy SAML

Identity Provider

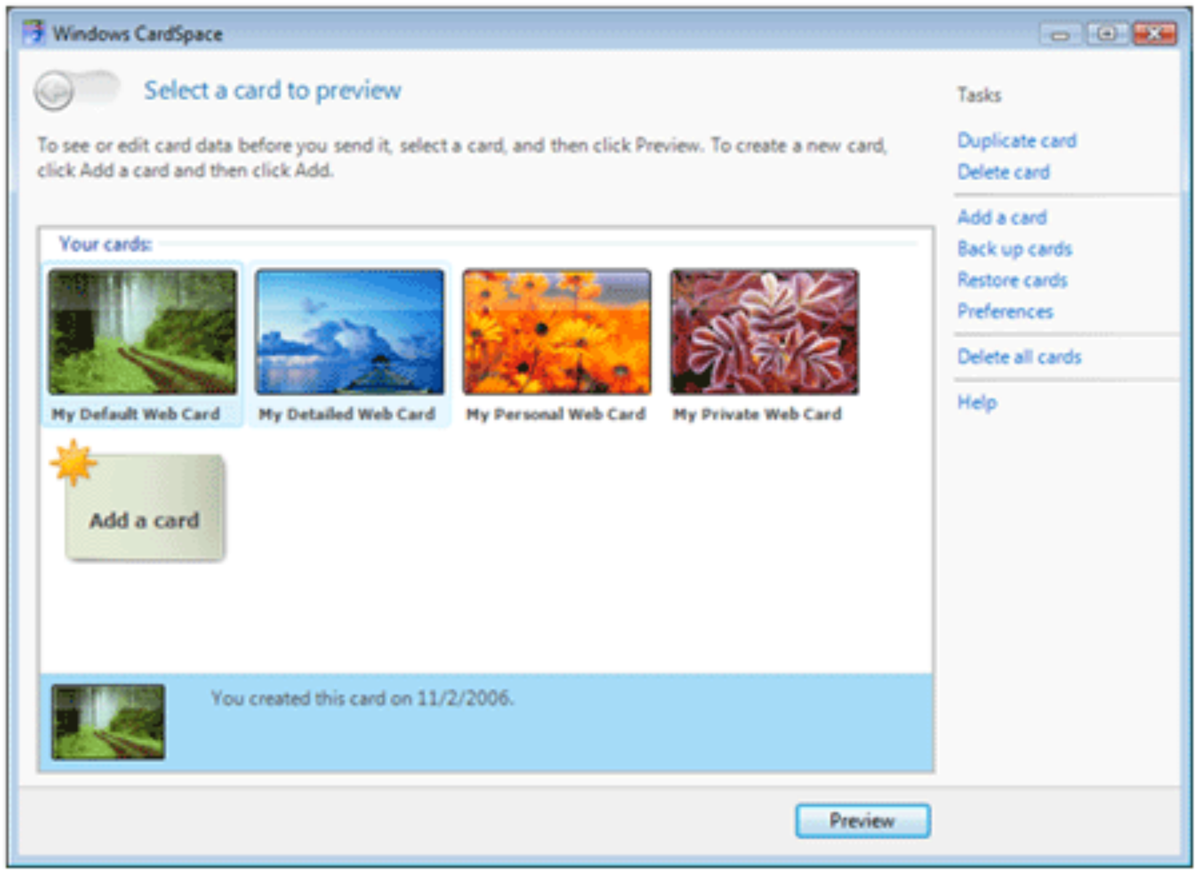
SAML Resp = < 🔒 >

Holder of key profile

- Not bearer tokens
- Token may only be used by someone that proves the possession of the private key of the certificate.
- Client certificates are self-signed and generated on spot for each service to preserve privacy
  - Unfortunately browsers don’t know how to do this efficiently
  - Browsers have poor computation power

Identity Selectors

- Extensions to b
  - Microsoft Car
  - Higgins
  - Several other
- Identity Metasy
  - Identity Selec
  - Identity Provi
  - Relying Partne



Select a card to preview

To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card and then click Add.

You created this card on 11/2/2006.

Identity Selectors

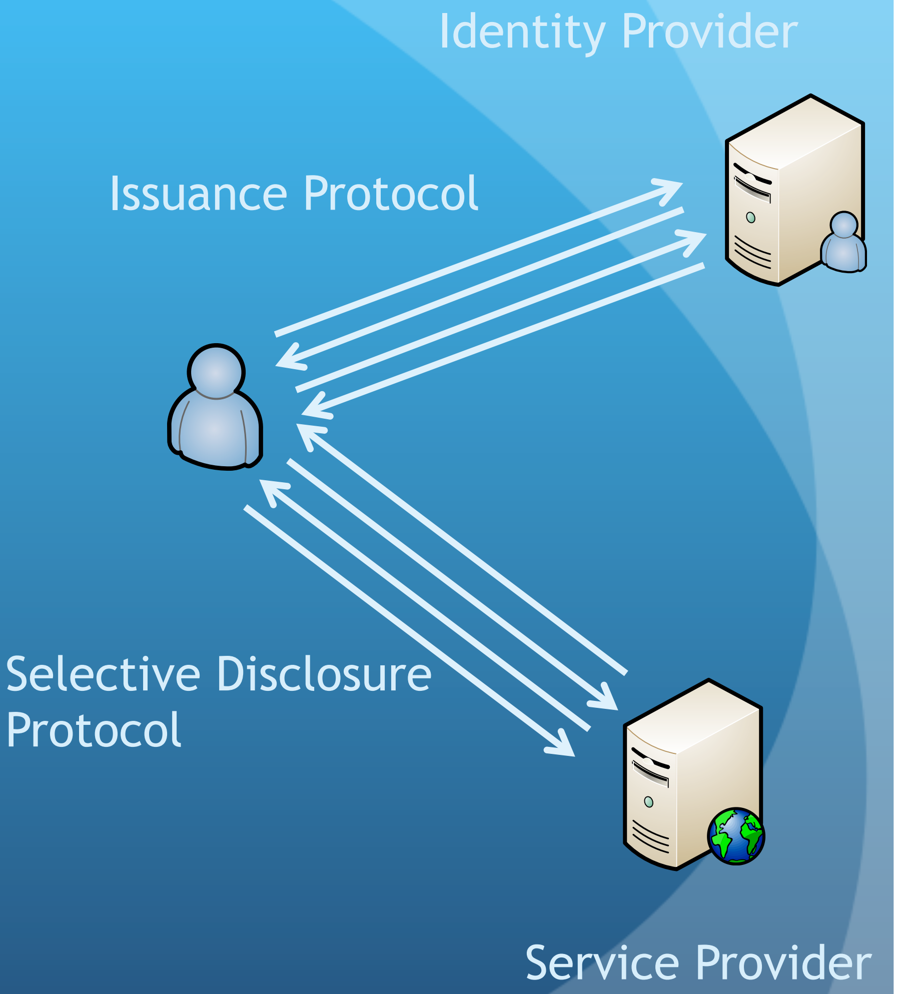
- Manage Cards with identities
  - SAML 2.0 tokens
  - WS-* tokens
  - OpenID tokens
  - U-Prove tokens

U-prove

- Special kind of tokens
  - May be encoded in WS-* claims (CardSpace 2.0)
  - May be encoded in SAML 2.0 tokens
- SPs only have access to the user attributes allowed by the user
  - selective disclosure
- IdP cannot get together with SP to know the full identity of the user
  - Untraceability
- IdP does not need to be online to allow selectively disclosure
  - Scalability

U-prove protocols

- Issuance protocol
  - Signed token with all the user attributes
  - <Name, Age, Address><signature>
  - IdP never sees <signature>
  - Untraceability
- Selective Disclosure Protocol
  - <Name, XXX, XXX><signature>
- The user must store the token
- Proof of possession
  - Prevents token replay



U-Prove credential

- <Name, age, address> = <x_1,x_2,x_3>
- For some set of generators $g_i = g_0^{y_i}$ of Z_p where p is a large prime

$$Credential = Cr =< g_1^{x_1} g_2^{x_2} g_3^{x_3} g_0^{\alpha} >$$

$$Signed\ Credential = \{Cr\}_{P_k}$$

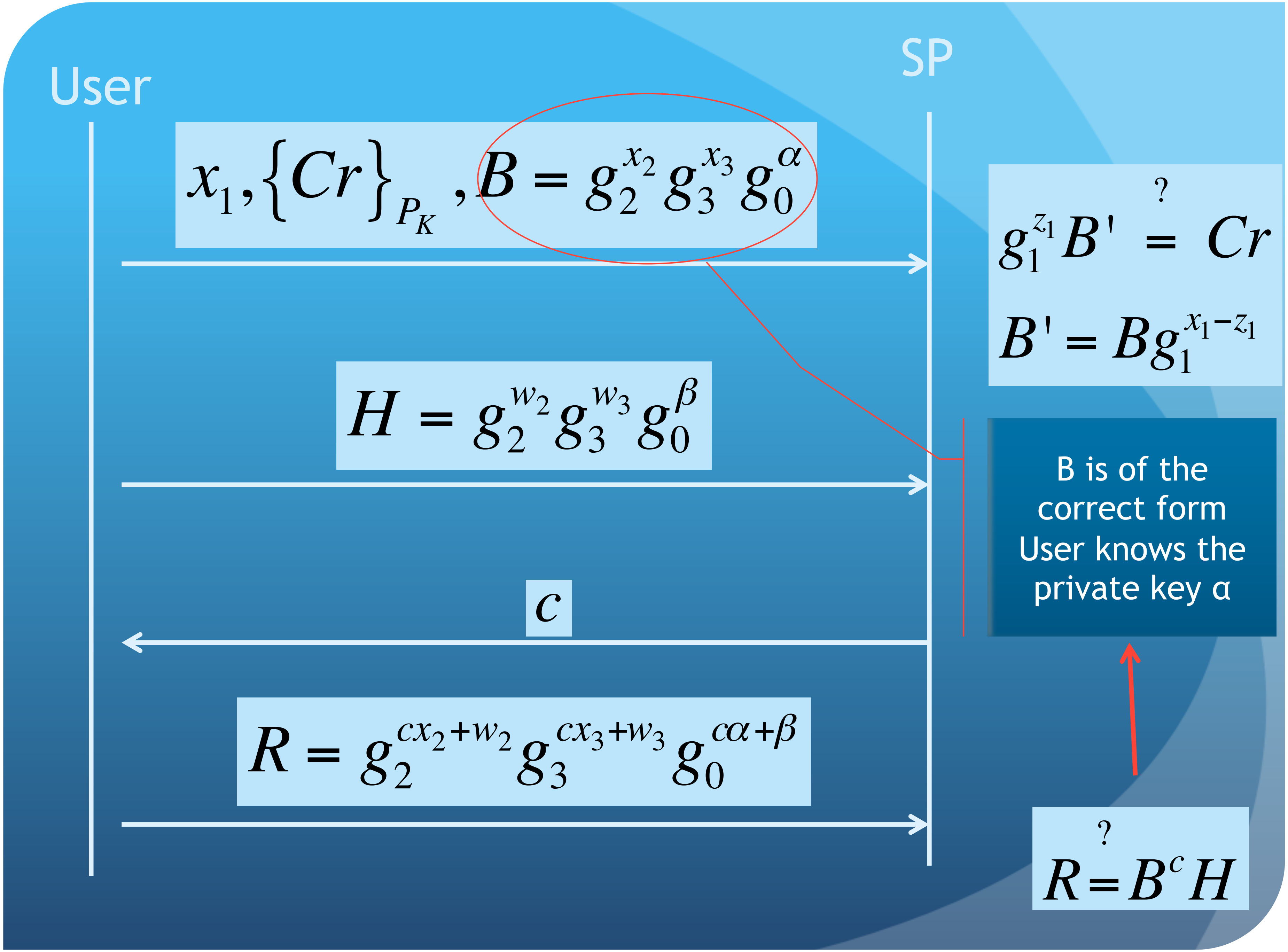
$$\alpha, g_0^{\alpha} are\ private\ user\ numbers$$

$$P_k\ Issuer\ private\ key$$

- Credential and signature can be public
  - Every one can verify the signature
  - No one can know x_i from the credential
- The private user numbers prevent dictionary and replay attacks

Selective disclosure Protocol

- If User provides x_1, x_2, x_3, α every service provider can verify the validity of the attributes by computing the credential and compare it with the sign one.
  - But the SP would no every thing about the user
  - But the SP could replay the attributes and the credential and fake to be the user
- How to disclose x_1 without disclosing x_2, x_3, α ?
- How to prove that you are the owner of the attributes ?



User

SP

$$x_1, \{Cr\}_{P_K}, B = g_2^{x_2} g_3^{x_3} g_0^{\alpha}$$

$$g_1^{z_1} B' \overset{?}{=} Cr$$

$$B' = B g_1^{x_1 - z_1}$$

$$H = g_2^{w_2} g_3^{w_3} g_0^{\beta}$$

$$c$$

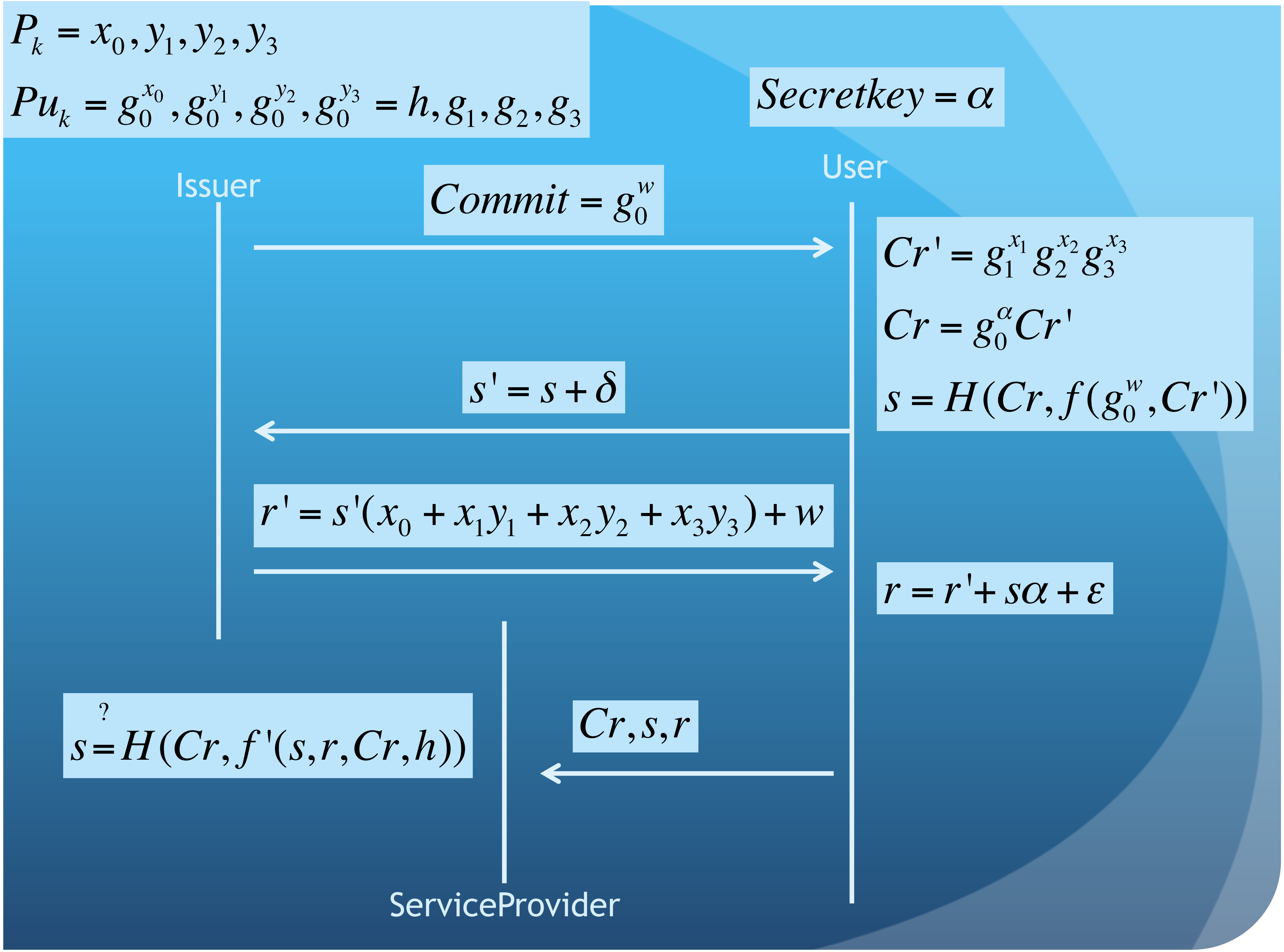
$$R = g_2^{cx_2+w_2} g_3^{cx_3+w_3} g_0^{c\alpha+\beta}$$

$$R \overset{?}{=} B^c H$$

B is of the correct form
User knows the private key α

Issuance Protocol

- Credential of the correct form $Cr = < g_1^{x_1} g_2^{x_2} g_3^{x_3} g_0^{\alpha} >$
- Credential = Cr and signature = <s,r> not know to the issuer
- α not know to the issuer
- x_1, x_2, x_3 know to the issuer



$P_k = x_0, y_1, y_2, y_3$

$Pu_k = g_0^{x_0}, g_0^{y_1}, g_0^{y_2}, g_0^{y_3} = h, g_1, g_2, g_3$

$Secretkey = \alpha$

Issuer

User

$Commit = g_0^{w}$

$Cr' = g_1^{x_1} g_2^{x_2} g_3^{x_3}$

$Cr = g_0^{\alpha} Cr'$

$s = H(Cr, f(g_0^{w}, Cr'))$

$s' = s + \delta$

$r' = s'(x_0 + x_1 y_1 + x_2 y_2 + x_3 y_3) + w$

$r = r' + s\alpha + \varepsilon$

$s \overset{?}{=} H(Cr, f'(s, r, Cr, h))$

Cr, s, r

ServiceProvider

U-prove properties

- Scalable
- Untraceability
- Selective Disclosure
- Hardware tokens support
    - If only the hardware token knows one of the x_i the user cannot create Cr’ without the token
- But how to know that what you are disclosing is what you want?
    - Is your computer with virus?
    - “What you see is what you sign” ?

User centric security

- Not the principal the real user
- For very sensitive applications we may have a secret channel between the user and the service provider
- Some solutions have been implemented for specific applications but none is generic
    - E.g. MarkPledge for e-voting stuff

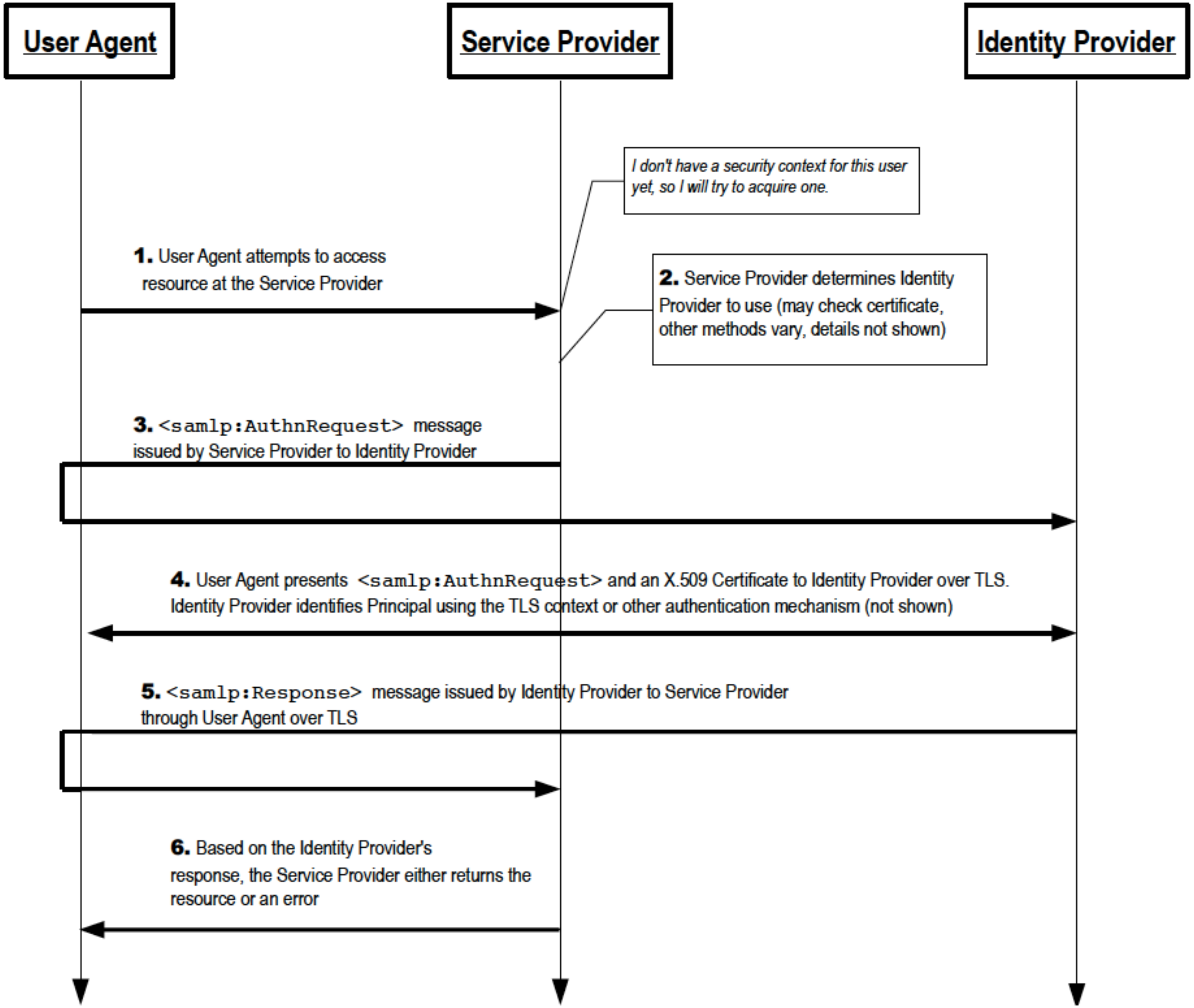
Conclusions

- A unsuspicious number of attacks to the web result from poor authentication
- Several solutions have been proposed
  - DNSSEC, STORK, U-Prove

Questions ?

Carlos.Ribeiro@ist.utl.pt

- We are still far from protecting the user from all authentication pitfalls, but we are getting closer



User Agent

Service Provider

Identity Provider

I don't have a security context for this user yet, so I will try to acquire one.

1. User Agent attempts to access resource at the Service Provider

2. Service Provider determines Identity Provider to use (may check certificate, other methods vary, details not shown)

3. `<samlp:AuthnRequest>` message issued by Service Provider to Identity Provider

4. User Agent presents `<samlp:AuthnRequest>` and an X.509 Certificate to Identity Provider over TLS. Identity Provider identifies Principal using the TLS context or other authentication mechanism (not shown)

5. `<samlp:Response>` message issued by Identity Provider to Service Provider through User Agent over TLS

6. Based on the Identity Provider's response, the Service Provider either returns the resource or an error