

ZXID.org SAML 2.0 ja ID-WSF 2.0 / TAS³ open source -implementaatio

Sampo Kellomäki (sampo@symlabs.com), Symlabs

Haka- ja Virtu-käyttäjien kokoontuminen

3.2.2010 Helsinki



Esittely

Sampo Kellomäki (sampo@iki.fi) +351-918.731.007

- ZXID.org pääohjelmoija
- TAS³ Pääarkkitehti
- Liberty / Kantara / OASIS standadointi
 - Paljon interop kokemusta
- Symlabs: yksi perustajista

Symlabs

- Identiteetin hallintaan keskittynyt softa toimittaja
 - Virtuaalihakemisto-tuotteet
 - Federoitu identiteetti-tuotteet (IdP, SP, ...)
 - Konsultointi
- Tuotteet aluasta asti Liberty Sertifioituja (olimme mukana suunnittelemassa sertifiointi prosessia)
- Pienyritys joka toimii Lissabonista, Madridista, ja Chicagosta
- Kaupallinen ZXID.org tuki

TAS³

- Trusted Architecture for Securely Shareable Services
- EU FP7 tutkimus projekti
- Tunnistus, valtuutus, auditointi
- Kertakirjautuminen, webbi palvelut, auditointi paneli käyttöliittymä
 - ulkoistetut palvelut, SOA, pilvi
- SAML 2.0, XACML 2.0, ID-WSF 2.0
- ZXID.org on TAS³ referenssi-implemентаatio
- <http://zxid.org/tas3>
- <http://www.tas3.eu>

ZXID.org

- Open source, Apache2 lisenssi
- SAML 2.0: SP, IdP
- XACML 2.0: valtuutus (PEP, PDP-tynkä)
- ID-WSF 2.0: WSC, WSP, Discovery Bootstrap, Discovery Client, Discovery Palvelin

ZXID.org moduilit

- mod_auth_saml apache moduli: SP, PEP, Bootstrap
- ZXID SSO Servlet, esim. Tomcat ympäristöön: SP, Bootstrap
- ZXID ja TAS³ API: SP, PEP, WSC, WSP, Bootstrap, Discovery Client
 - C / C++
 - Java
 - PHP
 - perl (Net::SAML moduli)
- zxididp: IdP, Attribuutti palvelin, Bootstrap, Discovery Palvelin
- Tulevia ominaisuuksia
 - Axis2 moduli: WSC, WSP, Discovery Client
 - IIS filteri: SP, Bootstrap
 - C# / .Net API: SP, PEP, WSC, WSP, Bootstrap, Discovery Client

Mikä Bootstrap?

- Silta kertakirjautumisen ja webbipalveluiden välillä
- Webbipalveluiden välillä tulisi käyttää Discovery Palvelua
- SAML 2.0 kertakirjautumisen yhteydessä välitettävä erityinen attribuutti joka osoittaa Discovery palvelun, ja mahdollisesti muita palveluita.
- Bootstrapin tuottaminen vaatii IdP:n puolella erityistukea
 - zxididp tukee, monet kaupalliset, esim. Symlabs, tukevat
 - Valitettavasti Shibboleth / OpenSAML ei tue vielä

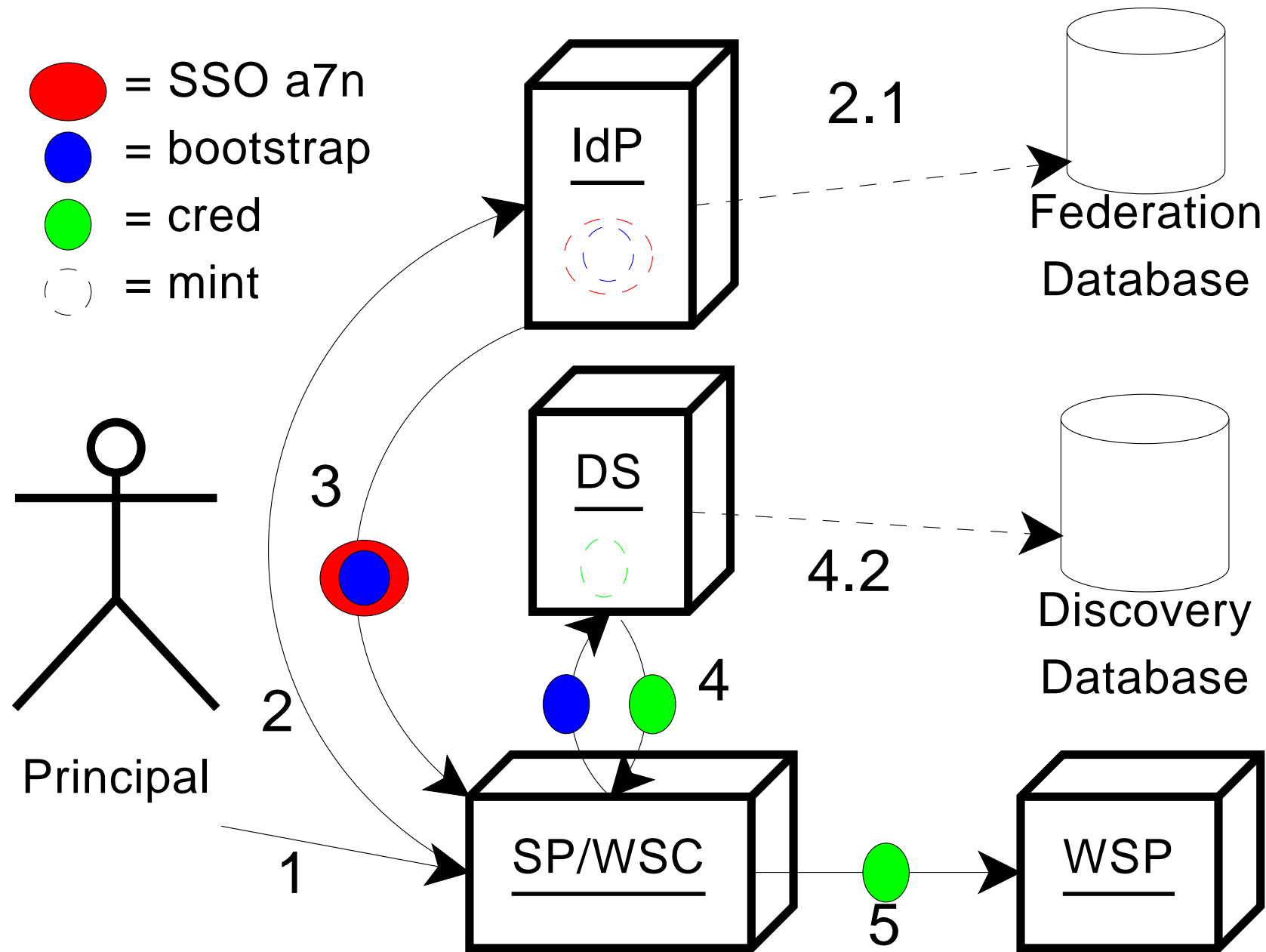


Figure 1: Single Sign-On (2,3), Discovery (4), and call to WSP (5). The blue ball represents discovery bootstrap.

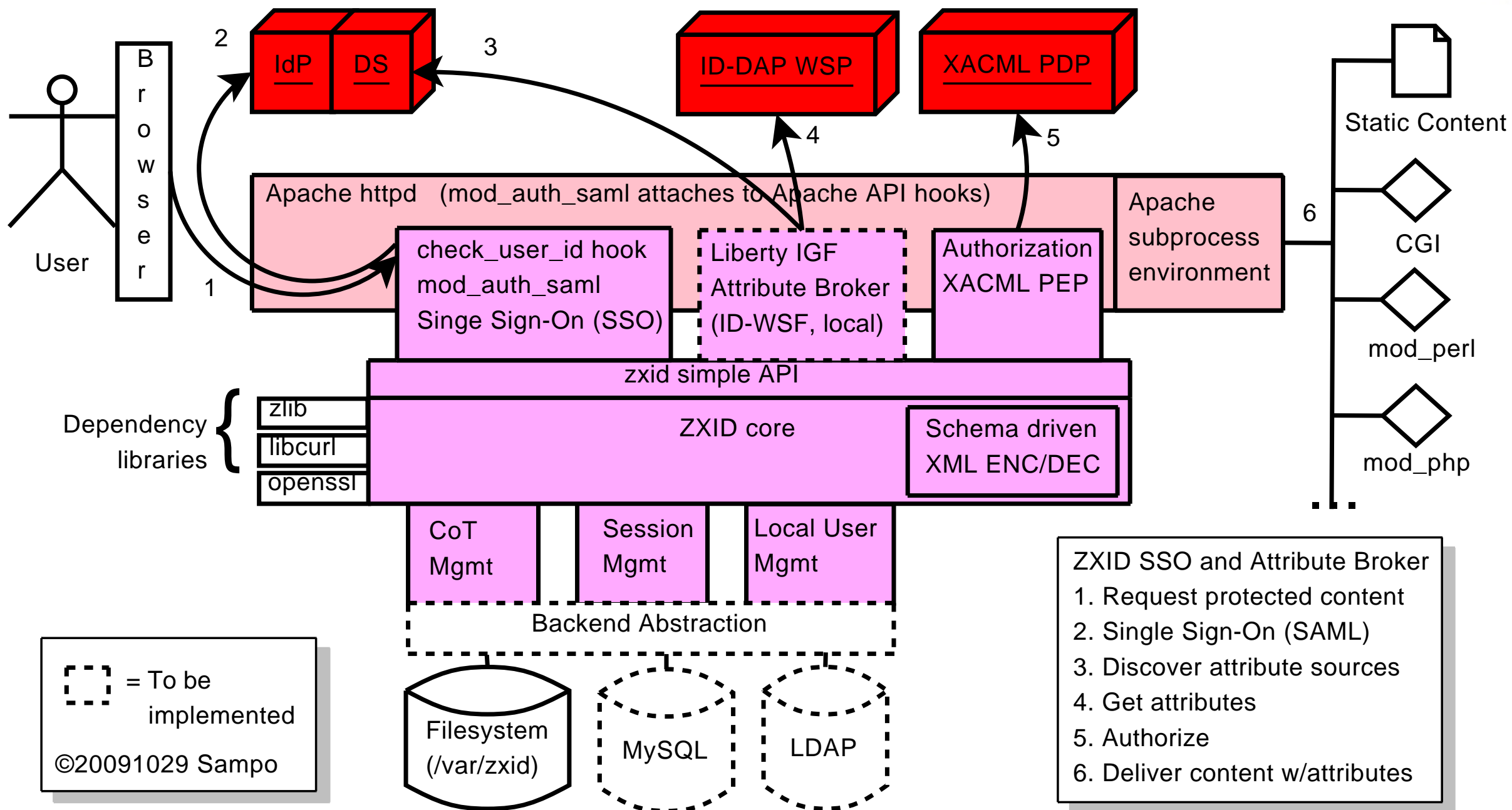
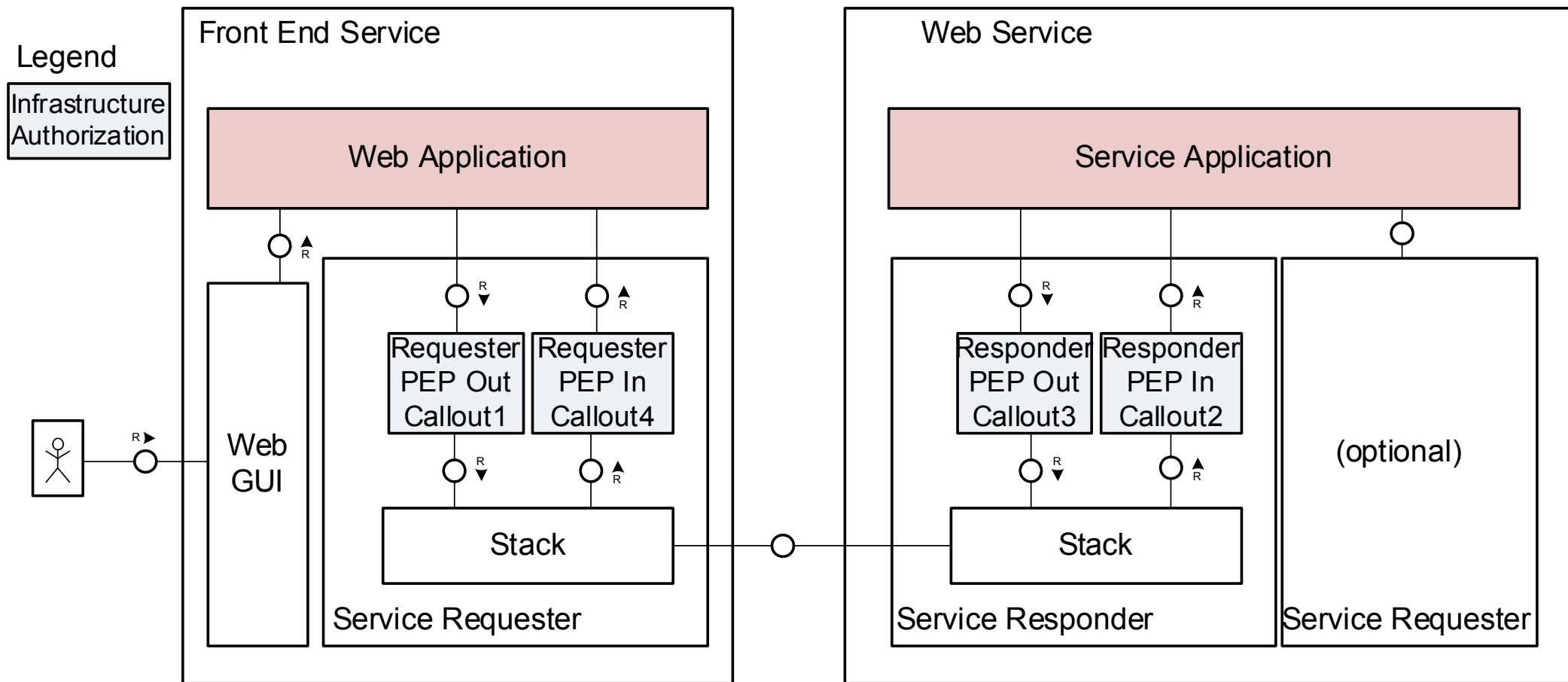
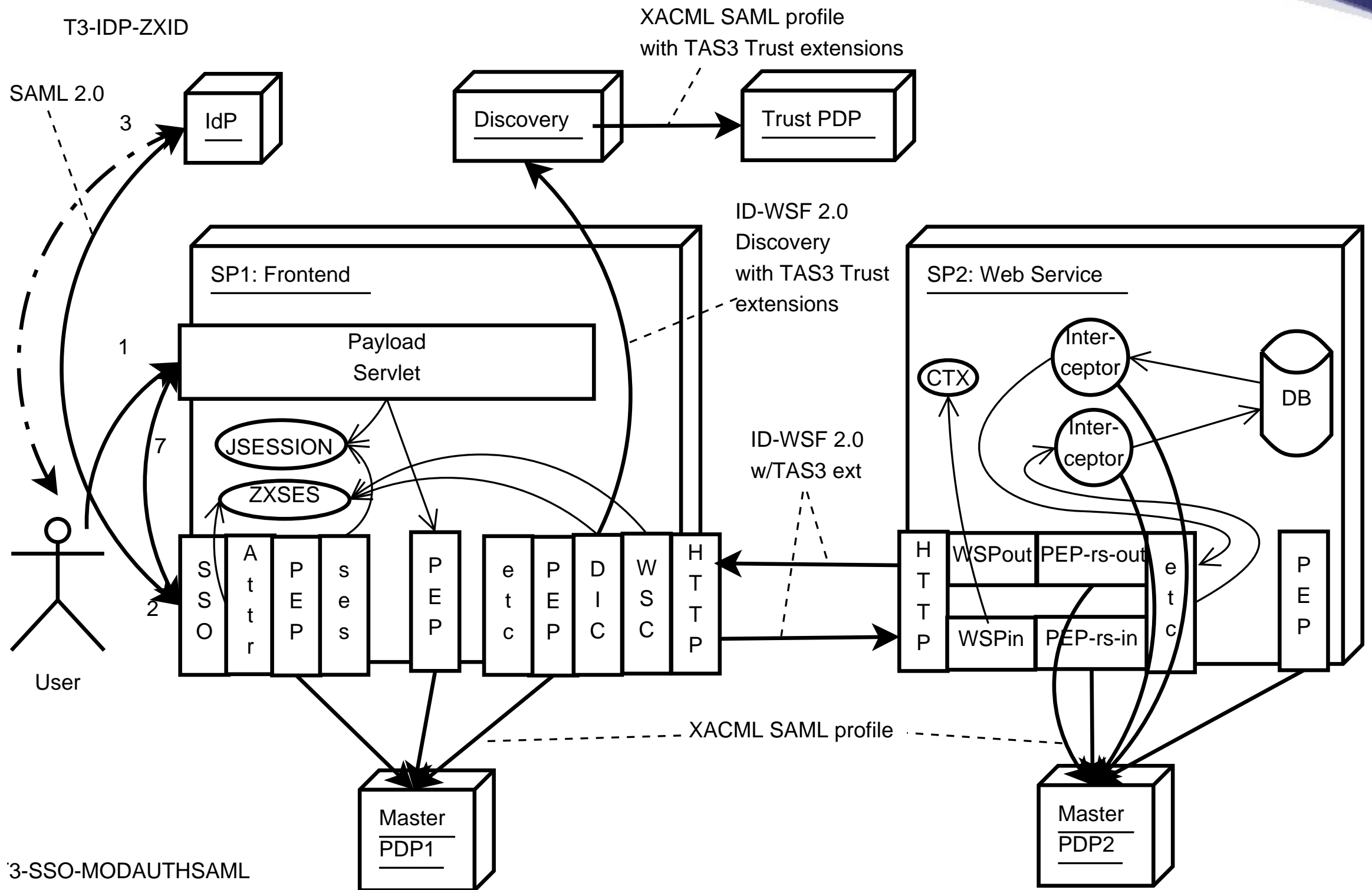


Figure 2: Software Architecture of `mod_auth_saml`.





3-SSO-MODAUTHSAML

Demo

Jos demoa ei ole, seuraavat kalvot suunnilleen kattavat demon

IdP:n valinta

ZXID SP Federated SSO (user NOT logged in, no session)

Login Using New IdP

A new IdP is one whose metadata we do not have yet. We need to know the IdP URL (aka Entity ID) in order to fetch the metadata using the well known location method. You will need to ask the administrator of the IdP to tell you what the EntityID is.

IdP URL

Entity ID of this SP (click on the link to fetch the SP metadata): <https://sp1.zxidsp.org:8443/zxidhlo?o=B>

Login Using Known IdP

Technical options

Create federation, NID Format:

zxid.org, 0.18 1178728139 libzxid (zxid.org)

Tunnistus IdP puolella

symLABS

e-nabling your business

Synlabs Federated Identity Access Manager

DirectoryScript

Welcome to Id Provider "IdP3 A" Home Login

You may login using various methods (pick your poison)

(be sure browser accepts cookies from the same domain)

1. Cookie login

Username:

Password:

If any web site (SP) asks...

The *IdP URL* (Provider ID/Entity ID) of this IdP is <https://a-idp.liberty-iop.org:8881/idp.xml>

You can cut and paste the above URL to any web site that allows Single Sign-On using *IdP URL* or "Any IdP" or "Other IdP". This mechanism allows the web site (SP) to dynamically join the Circle of Trust of this IdP. This is called *Auto-CoT*.

Onnistunut kertaalleen kirjautuminen: Suojattu sisältö

ZXID HELLO SP Management (user logged in, session active)

Local Logout

Single Logout (Redir)

Single Logout (SOAP)

Defederate (Redir)

Defederate (SOAP)

[sid\(Snlg5j2nB\)](#) [nid\(Ple9OQMhOpLCkz72rTbJv\)](#) [Reload](#)

[zxid.org](#), 0.18.1178728139 libzxid (zxid.org)

Webbipalvelu kutsut

(ei kuvaa)

mod_auth_saml Apache ympäristössä

- Ei ohjelmointia. Riittää että Apachen asetukseen lisätään:

```
LoadModule auth_saml_module modules/mod_auth_saml.s

<Location /protected>
    Require valid-user
    AuthType "saml"
    ZXIDConf "URL=https://sp1.zxidsp.org:5443/protect
    ZXIDConf "REDIR_TO_CONTENT=1"
</Location>
```

- Kaikki sovellukset jotka käyttävät HTTP Basic Auth:ia toimivat REMOTE_USER headerin kautta

SSO Servlet Tomcat ympäristössä

```
01 import zxidjava.*;    // Pull in the zxidjni.az() API
05
06 public class zxidappdemo extends HttpServlet {
07     public void doGet(HttpServletRequest req, HttpSe
08     throws ServletException, IOException
09     {
10     String fullURL = req.getRequestURI();
11     if (req.getQueryString() != null)
12         fullURL += "?" + req.getQueryString();
13     System.err.print("Start ZXID App Demo GET("+ful
14     HttpSession ses = req.getSession(false); // Im
15     if (ses == null) { // In
16         res.sendRedirect("sso?o=E&fr=" + fullURL);
17     return;
```

```
18     }
19
20     res.setContentType("text/html");
21     res.getOutputStream().print("<title>ZXID Demo A");
22
23     // Render logout buttons (optional)
24
25     print("<a href=\"sso?gl=1&s="+ses.getValue("se
```

Kertakirjautuminen PHP:ssä, *zxid_simple()*

- 38 riviä PHP koodia josta 22 asiaa (loput kommentteja tai HTML:ää)
- Kokonainen ratkaisu
 - Tuki kaikille SAML profiileille
 - Kertalleen uloskirjautuminen mukana (SLO)
 - Well Known Location (WKL) metadata exchange -tuki
- Ei tarpeen tuntea SAML protokollaa detajli tasolla
- Voit leikata ja liimata tämän omaan PHP sovellukseesi

Alustus (kerran)

```
01 <?
02 dl("php_zxid.so"); # Pull in module (.so file)
03 # CONFIG: You must have created /var/zxid directory
04 # CONFIG: You must edit the URL to match your domain
05 $conf = "PATH=/var/zxid/
           &URL=https://sp1.zxidsp.org:8443/zxidhlo.ph
06 $cf = zxid_new_conf_to_cf($conf);
07 ?>
```

- PATH configuration means multiple instances of ZXID can coexist (e.g. virtual hosting of web sites)
- URL configuration determines provider ID, can also be configured via `/var/zxid/zxid.conf`

Per suojattu sivu tai kunnes sessio on saatu käynnistettyä

```
08 <?
09 $qs = $_SERVER['REQUEST_METHOD'] == 'GET'
10     ? $_SERVER['QUERY_STRING']
11     : file_get_contents('php://input');
12 $res = zxid_simple_cf($cf, -1, $qs, &ses, 0x1814);
13
14 switch (substr($res, 0, 1)) {
15 case 'L': header($res); exit;
16 case '<': header('Content-type: text/xml'); echo $re
```

- Read input and call *zxid_simple()* to handle SAML protocol details
- Act on outcome of *zxid_simple()* as indicated by the first letter
 - L: protocol requires redirect, perform it
 - <: Send out XML data (such as Metadata or SOAP response)

IdP valinta

```
17 case 'n': exit;    # Already handled, do nothing furt
18 case 'e':
19 ?>
20 <title>Please Login Using IdP</title>
21 <h1>Please Login Using IdP</h1>
22 <?=zxid_idp_select_cf($cf, null, 0x1800)?>
23 <?
24 exit;
```

- e: indicates that IdP Selection page needs to be rendered
- *zxid_idp_select()* generates the ZXID standard form
- Alternatively you could supply your own HTML for the form as long as you respect the form field naming convention

Onnistunut kertakirjautuminen

```
25 case 'd': break; # Logged in case -- continue after
26 default: die("Unknown zxid_simple() res($res)");
27 }
28
29 # Parse the LDIF in $res into a hash of attributes $
30
31 foreach (split("\n", $res) as $line) {
32     $a = split(":", $line);
33     $attr[$a[0]] = $a[1];
34 }
35 ?>
```

- d: login successful, return data is LDIF entry with attributes of SSO

Suojattu sisältö, kertauloskirjautumis-nappulat

```
36 <title>Protected content, logged in</title>
```

```
37 <h1>Protected content, logged in as <?=$attr['cn']?>
```

```
38 <?=zxid_fed_mgmt_cf($cf, null, -1, $attr['sesid'], 0
```

- *zxid_fed_mgmt()* generates the Single Log-Out buttons
- This is the place to bootstrap your application's own session

Onnistunut kertakirjautuminen: attribuutit LDIF:inä

```
dn: idpnid=Pa45XAs2332SDS2asFs,affid=https://idp.dem
objectclass: zxidsession
affid: https://idp.demo.com/idp.xml
idpnid: Pa45XAs2332SDS2asFs
authnctxlevel: password
sesid: S12aF3Xi4A
cn: Joe Doe
```

- The LDIF entry is used as convenient format for passing attribute-value pairs from *zxid_simple()* to application
- Some "attributes" are synthesized, others come actually from assertion

Webbi palvelut

TAS3 API

tas3_sso() SSO (with optional application independent authorization)

tas3_az() Application Dependent Authorization

tas3_call() Web Services Client: call a web service and validate response

tas3_wsp_validate() Validate that web service request can be processed

tas3_wsp_decorate() Create a web service response

Attribuuttien käyttö

```
39     res.getOutputStream().print("<pre>HttpSession d
40     String[] val_names = ses.getValueNames();
41     for (int i = 0; i < val_names.length; ++i) {
42         res.getOutputStream().print(val_names[i] +
43     }
44
45     res.getOutputStream().print("</pre>");
46 }
47 }
```

Webbipalvelu kutsu (WSC)

```
ret = zxidjni.call(cf, zxidjni.fetch_ses(cf, sid),  
                  "urn:x-foobar", null, null, null,  
                  "<foobar>Do it!</foobar>");
```

Webbipalveluiden tarjoaminen (WSP)

```
01 zxidjava.zxid_ses ses = zxidjni.alloc_ses(cf);
02 String fullURL = req.getRequestURI();
03 zxidjni.url_set(cf, fullURL); // Virtual host support
04
05 String buf;
06 int len = req.getContentLength(); // Java / Servlet
07 byte[] b = new byte[len]; // way of reading
08 int here, got;
09 for (here = 0; here < len; here += got)
10     got = req.getInputStream().read(b, here, len - here);
11 buf = new String(b, 0, len);
```

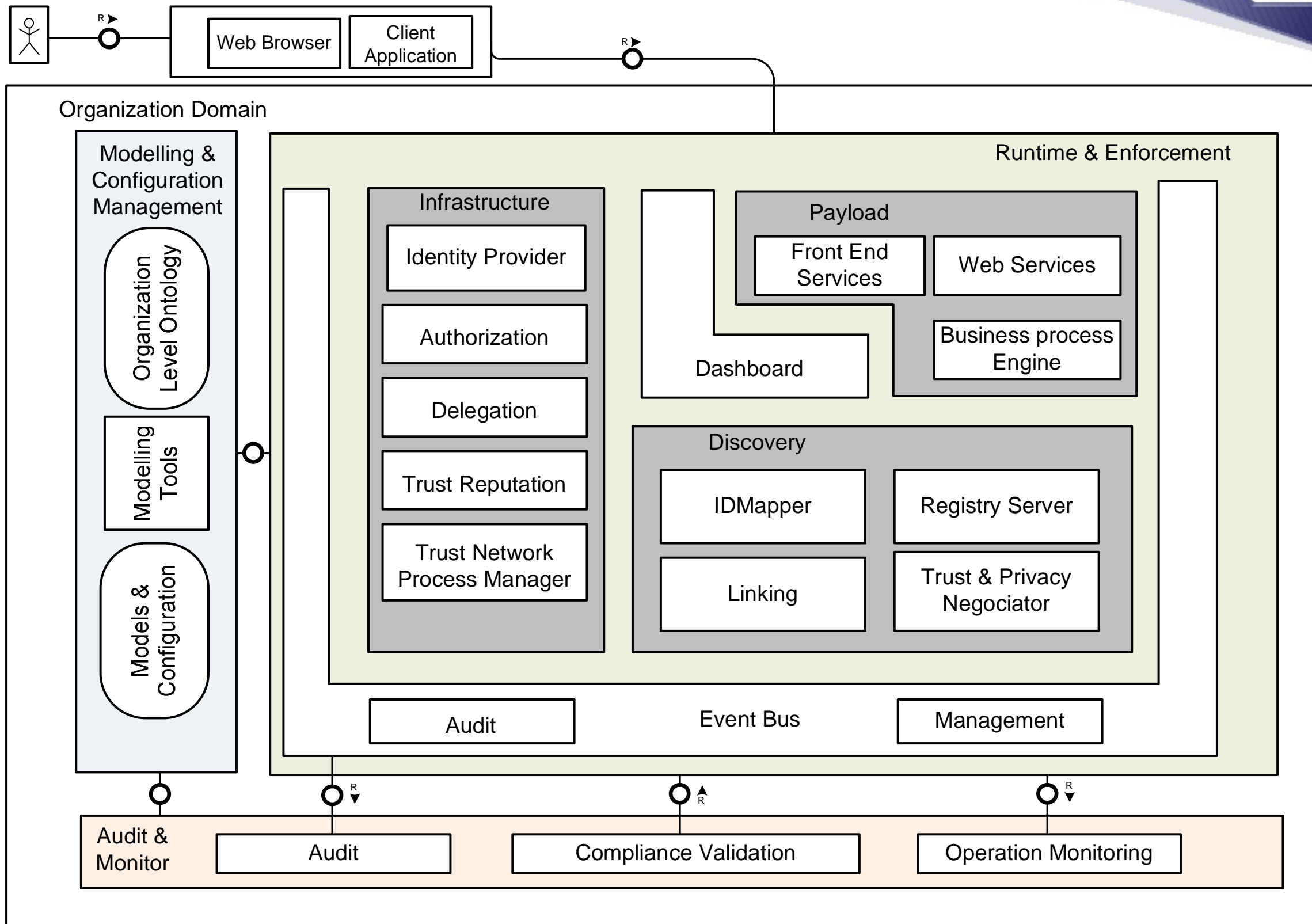
Pyynnön validointi

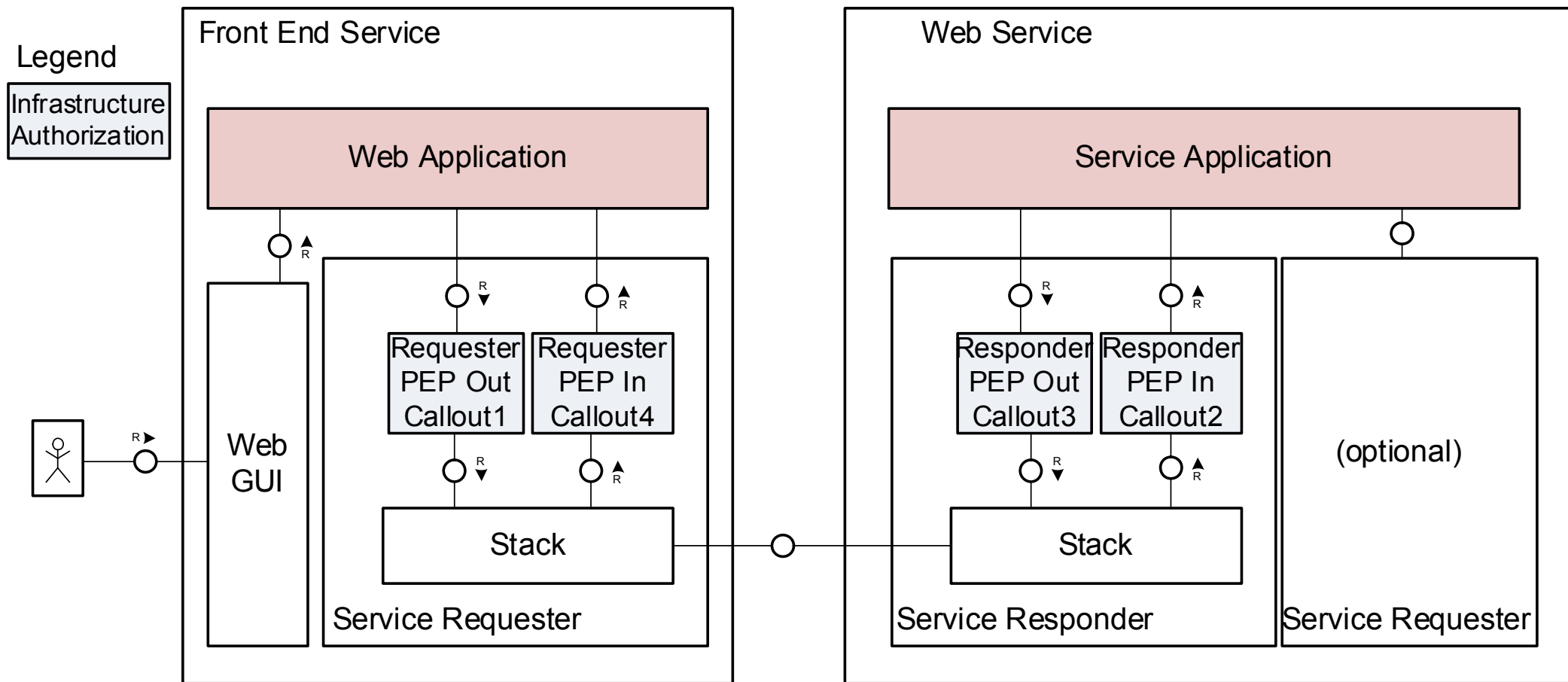
```
13 String nid =  
14     zxidjni.wsp_validate(cf, ses, null, buf);
```

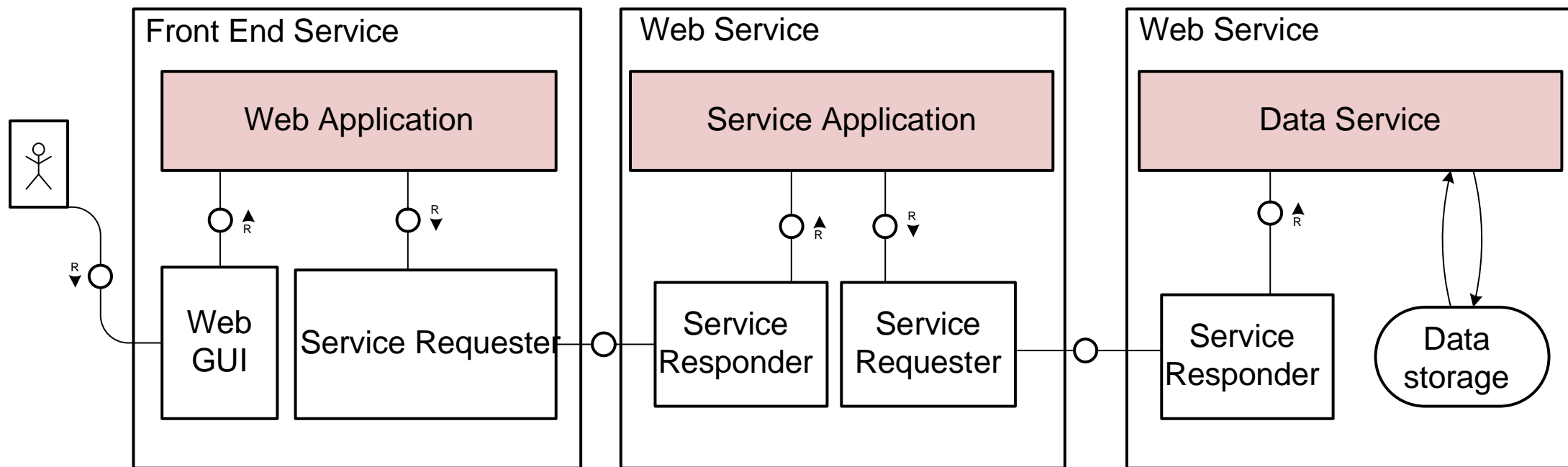
Valtuutus

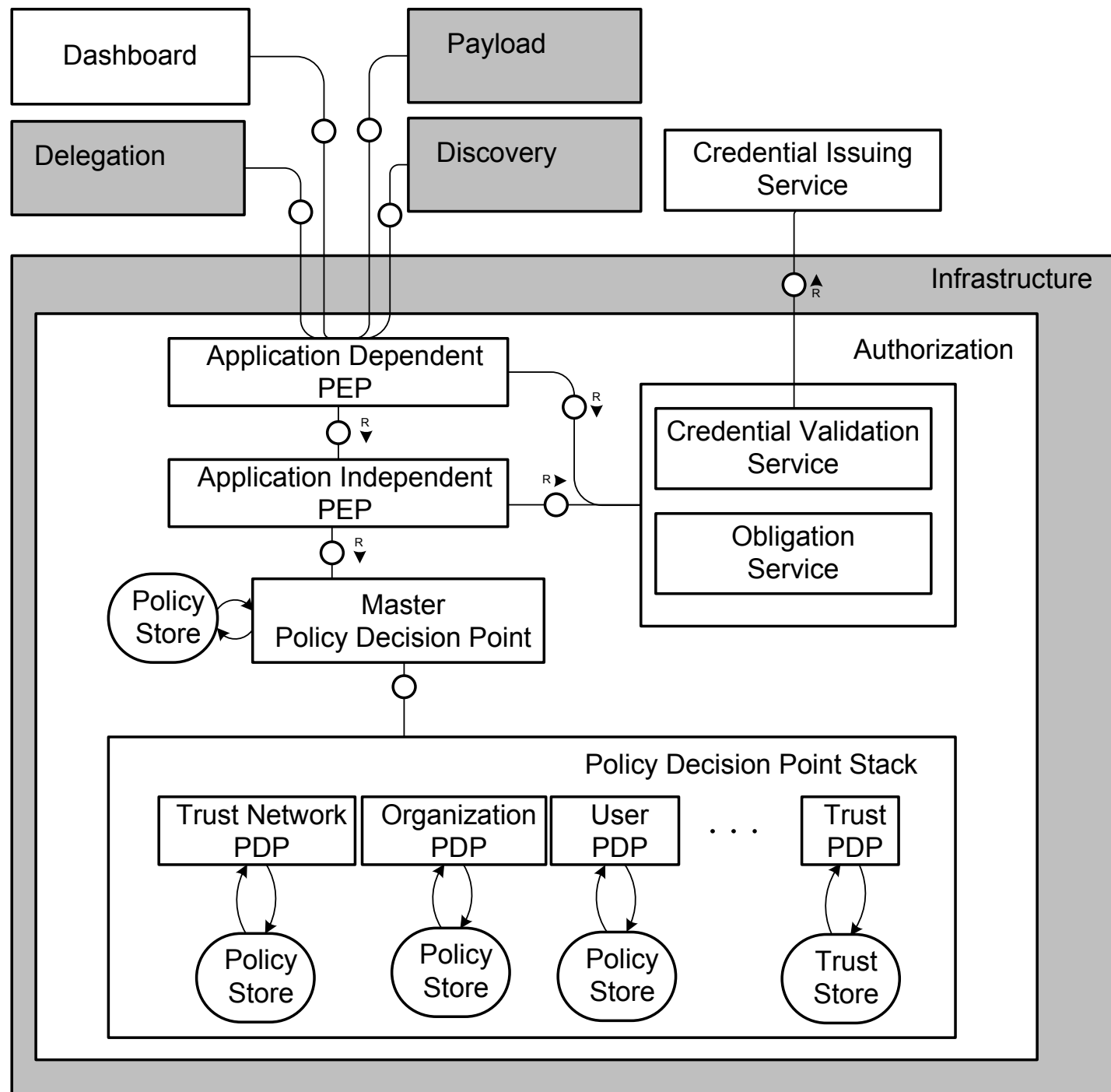
```
15 if (zxidjni.az_cf_ses(cf, "Action=Call", ses) == 0)
16     ret = zxidjni.wsp_decorate(cf, ses, null,
17         "<barfoo>" +
18         "<lu:Status code=\"Fail\" comment=\"Denied\">" +
19         "<data>Denied: nid="+nid+"</data>" +
20         "</barfoo>");
21 } else {
22     ret = zxidjni.wsp_decorate(cf, ses, null,
23         "<barfoo>" +
24         "<lu:Status code=\"OK\" comment=\"Permit\">" +
25         "<data>nid="+nid+"</data>" +
26         "</barfoo>");
27 }
28 res.getOutputStream().print(ret);
```

TAS³ Arkkitehtuuri









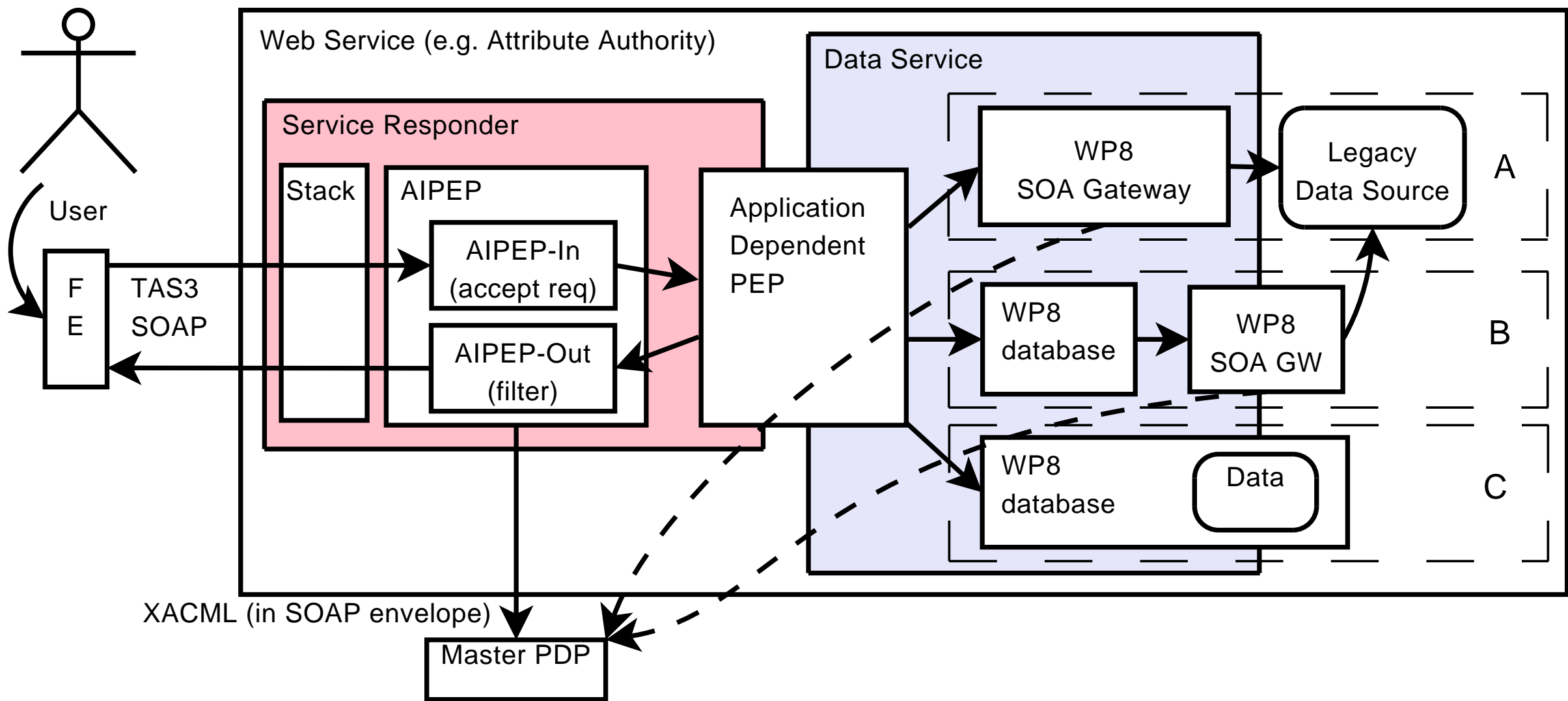


Figure 3: Application Integration using ADPEP and (A) Risaris SOA Gateway, (B) Fedora as frontend to Risaris, (C) Fedora database.

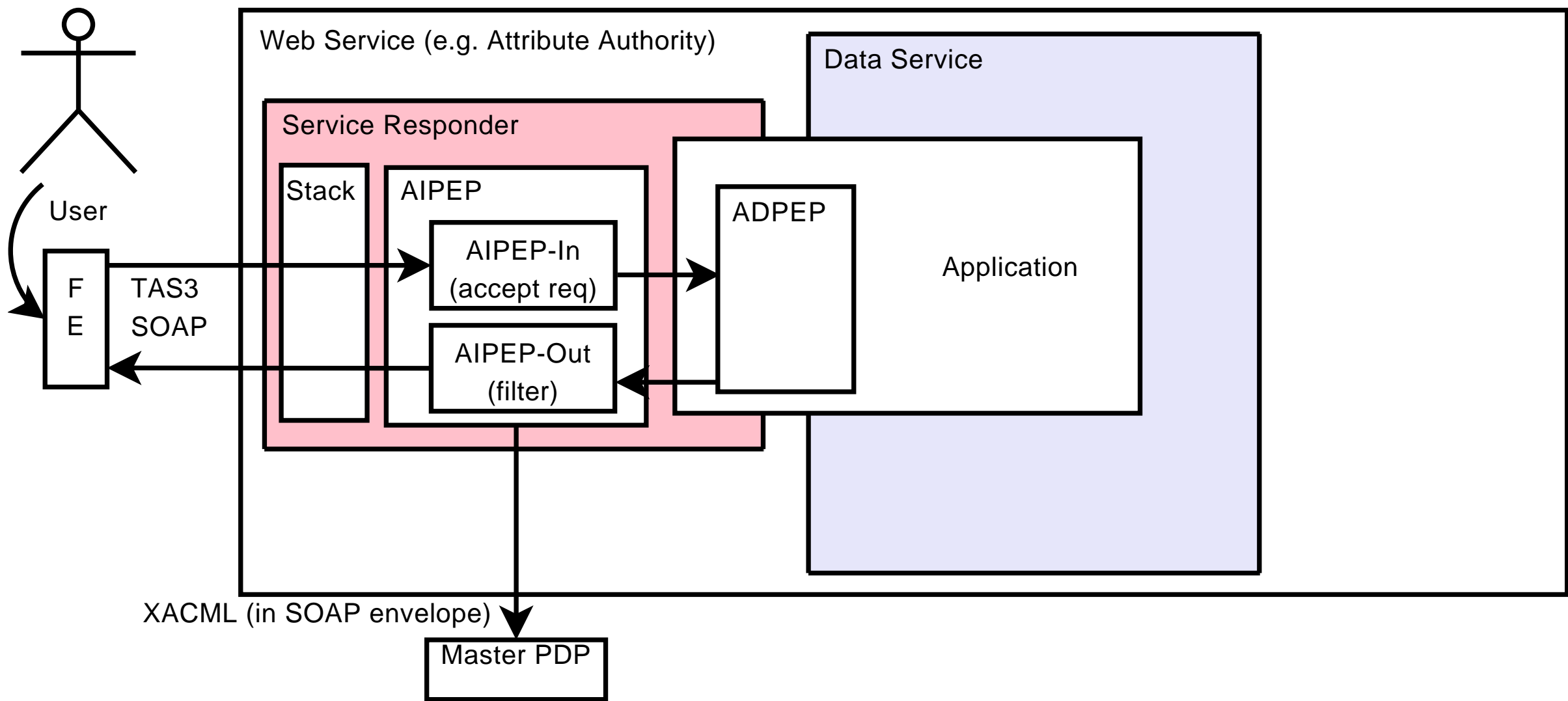


Figure 4: Application Integration: ADPEP implemented in application itself.

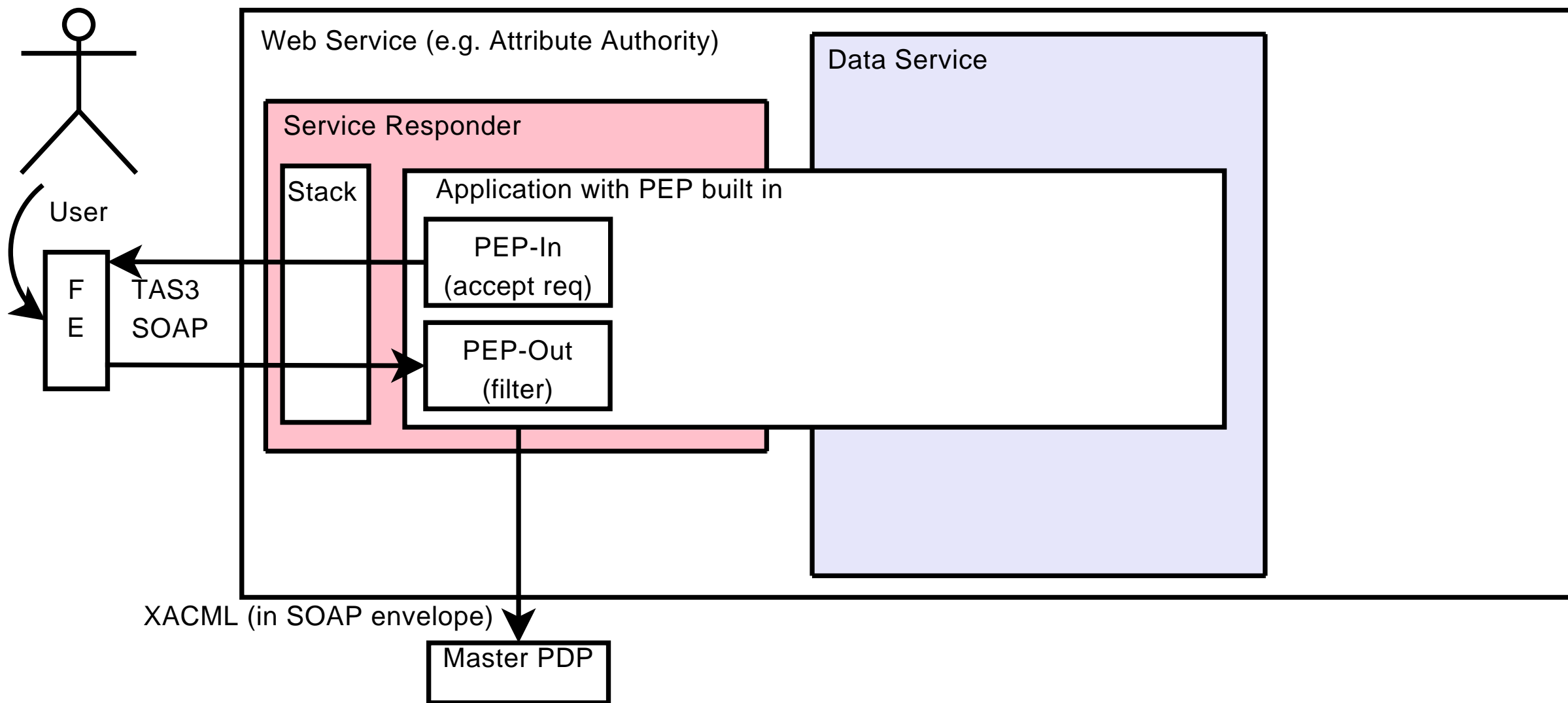
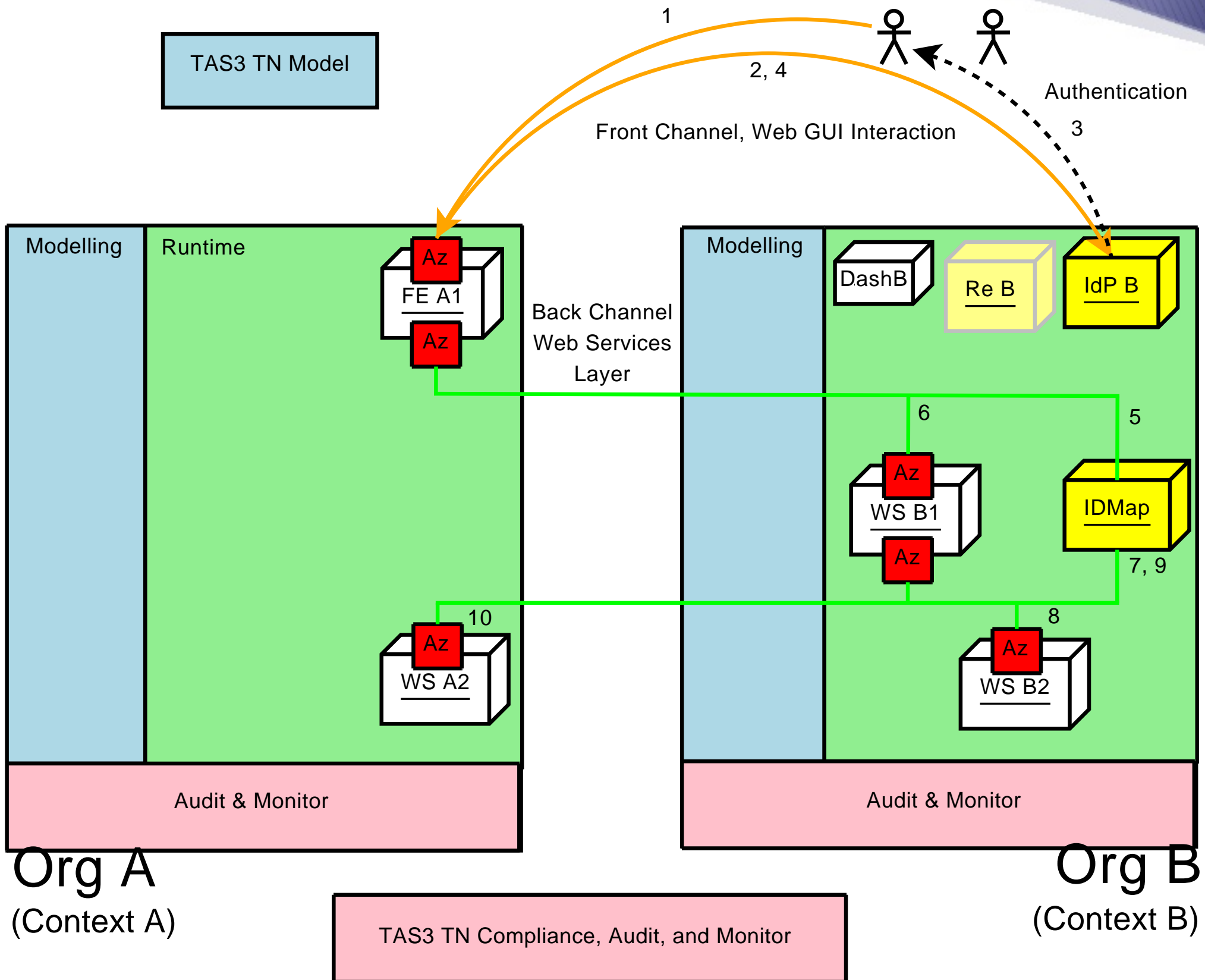


Figure 5: Application Integration: PEP implemented directly in application.



TAS3 TN Model

