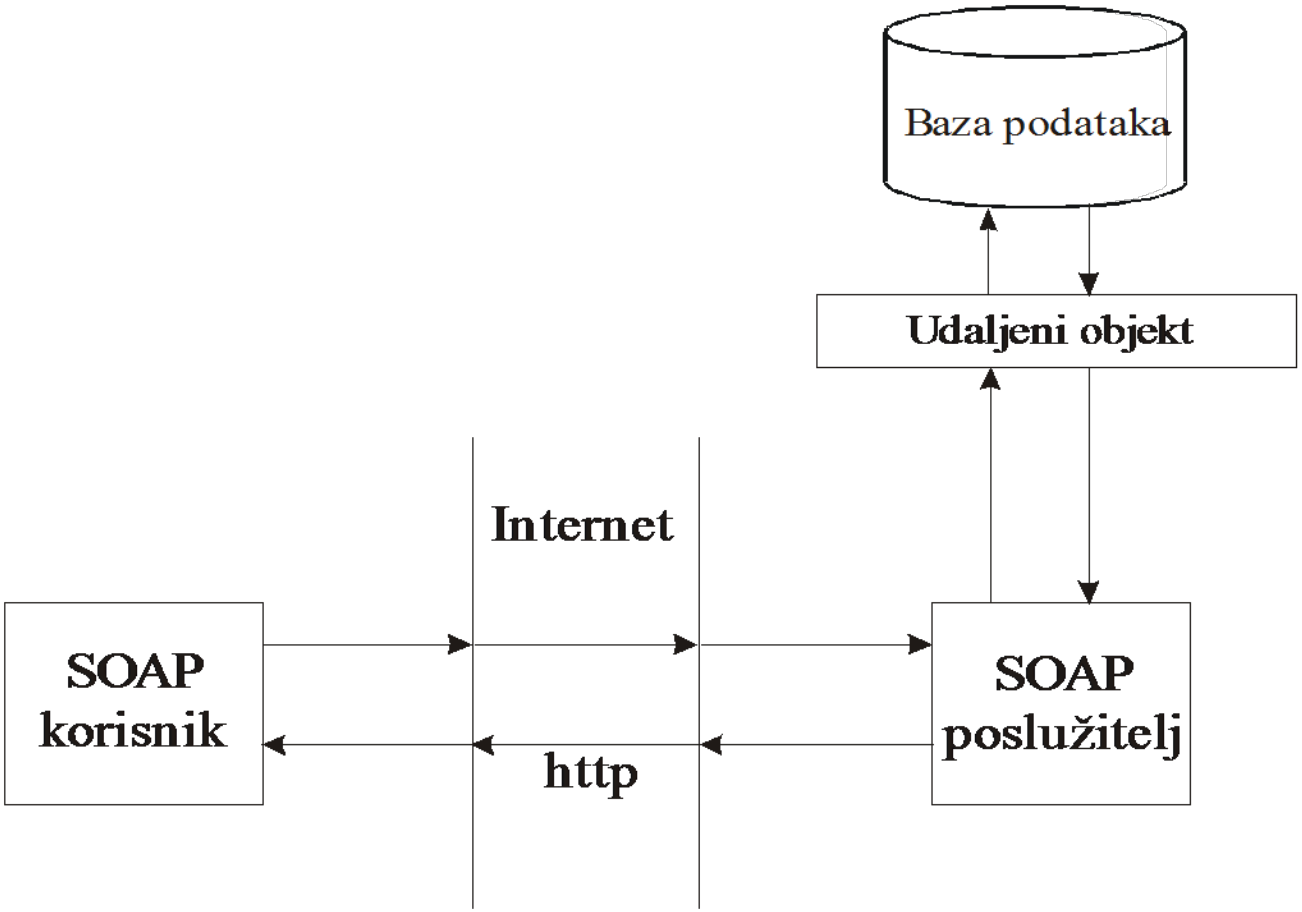
Servisno orijentirane arhitekture i njihov standardizacijski okvir

Web servisi

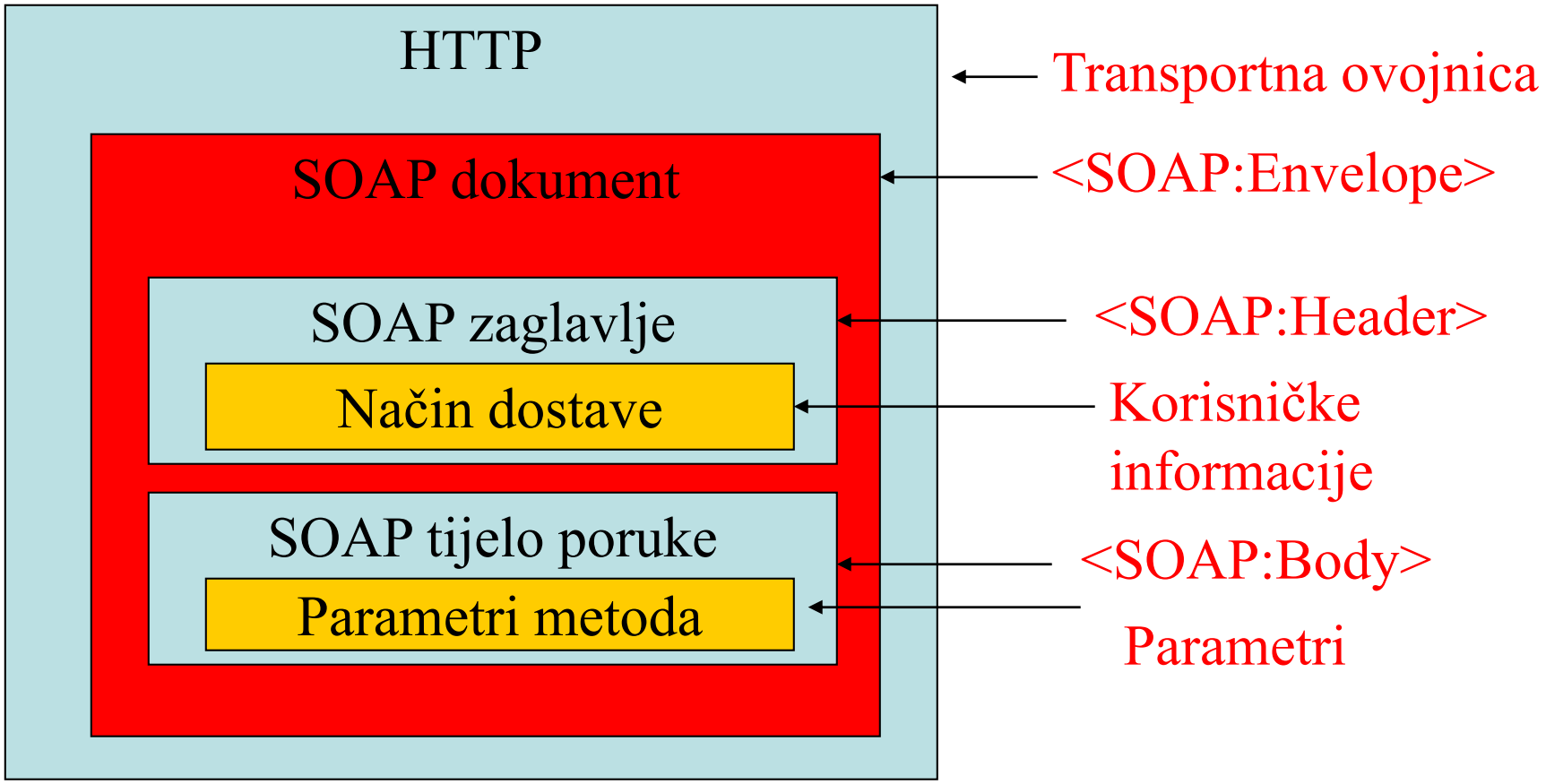
- Sve je počelo krajnje jednostavno



SOAP – kao što i ime kaže - jednostavno

- Simple Object Access Protocol
- počeo kao način pozivanja DCOM objekata labavije povezanih
- korištenje Internet protokola HTTP, FTP
- baziran na XML-u

SOAP Struktura poruke



- Ali nikada nije jednostavno, zahtjevi na arhitekturu su s vremenom rasli pa je iz Web servisa napravljena SOA

SOAP struktura poruke

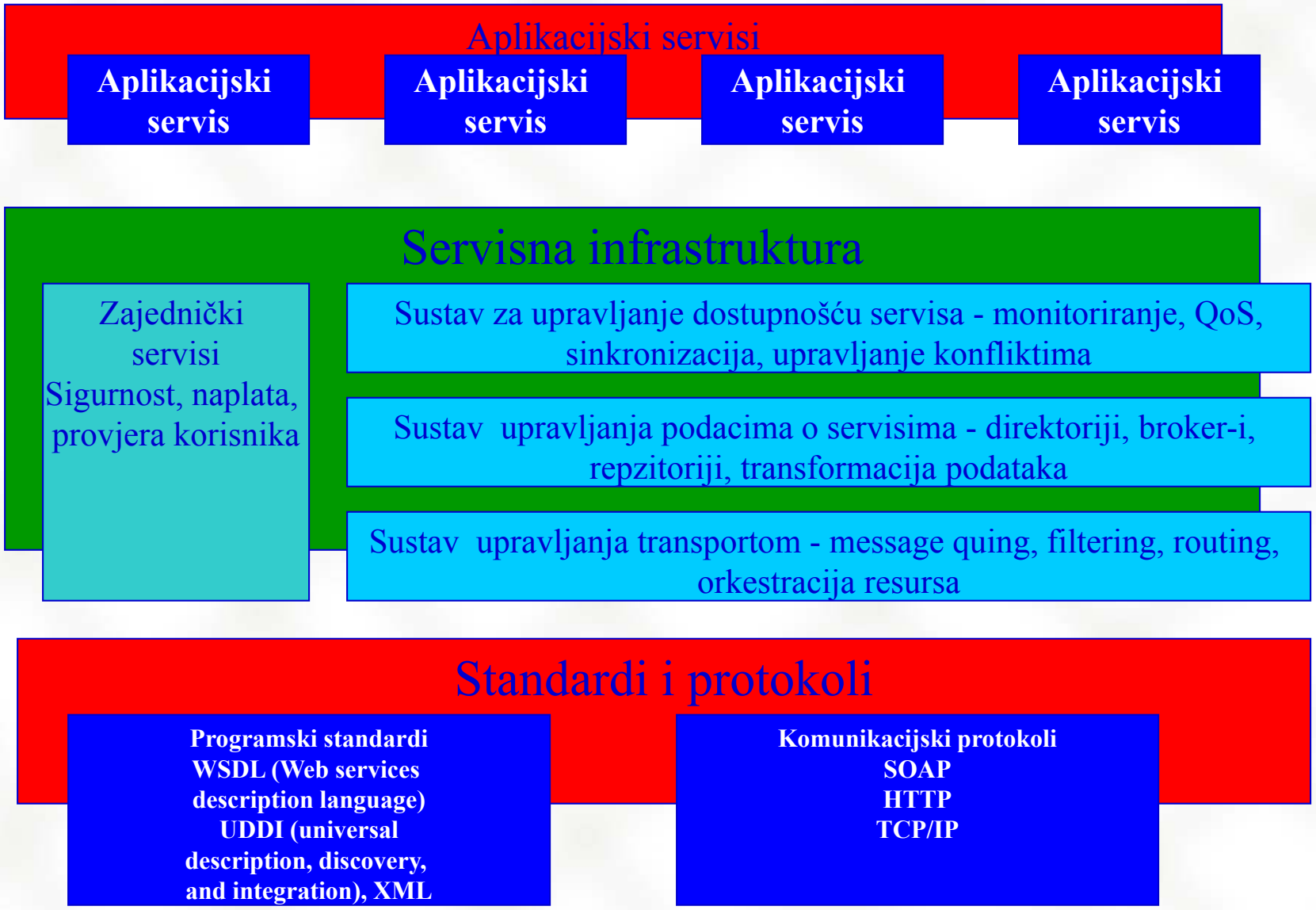
- Dva elementa poruke SOAP:Header i SOAP:Body
- SOAP:Header - informacije o transakciji - korisnički definirana informacija

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap.v1">
        <SOAP:Header>
                <trans:Transaction
                xmlns:trans="http://schemas.JDU.hr/transaction.xsd"
        SOAP:mustUnderstand="1">opis_transakcije</trans:Transaction>
        </SOAP:Header>
        <SOAP:Body>
                <Isporuci_podatke
                xmlns="http://schemas.JDU.hr/Isporuci_podatke.xdr">
                <ID_podatka>AA9999999</ID_podatka>
                </Isporuci_podatke>
        </SOAP:Body>
</SOAP:Envelope>
```

SOAP problemi

- Upravljanje transakcijama - potpuno programski
- XDR - nije prihvaćena od W3C
- Nedostatak standarda za namespaces
- Pristup servisima i dodjela ovlaštenja!
- Način naplate

Arhitektura Web servisa



SOA – definicije (1)

- SOA je informacijska i komunikacijska (ICT) arhitektura koja pruža fleksibilnost potrebnu za implementiranje elemenata poslovnog procesa i postavljanje potrebne ICT infrastrukture u obliku sigurnih, standardiziranih komponenti (servisa) koje se mogu višestruko koristiti i međusobno kombinirati kako bi zadovoljile različite poslovne potrebe.

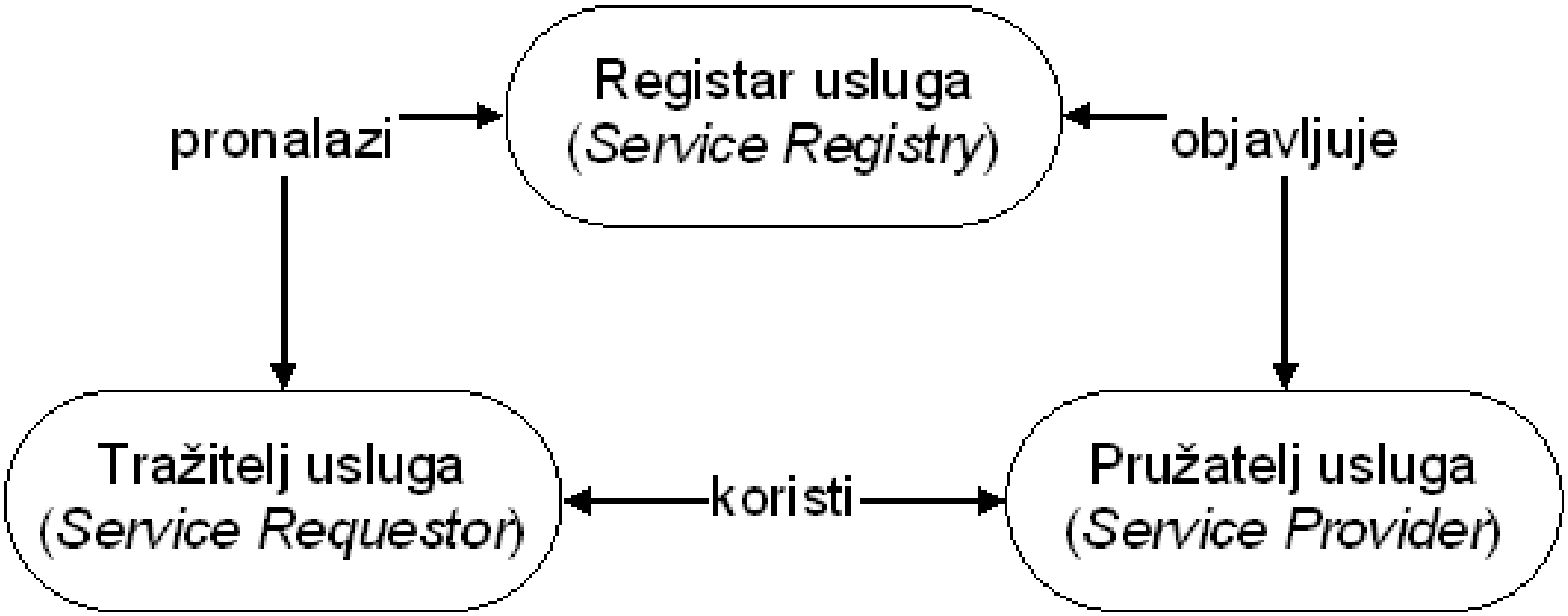
SOA – definicije (2)

- SOA je široko rasprostranjena ICT arhitektura koja se temelji na labavoj povezanosti (*loose coupling*), višestrukoj iskoristivosti (*reuse*) i interoperabilnosti između različitih programskih i organizacijskih sustava

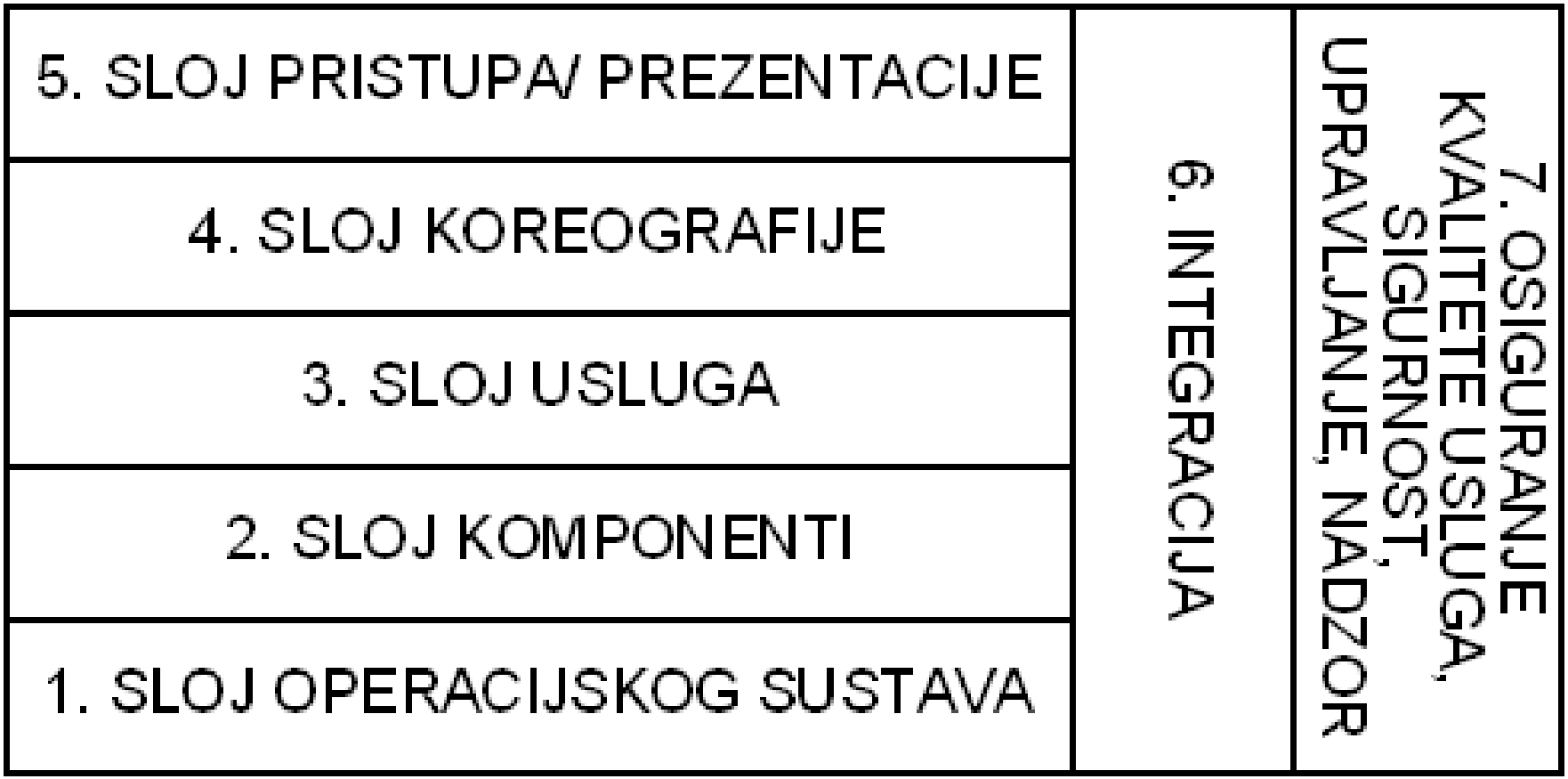
SOA – definicije (3)

- Servisno orijentirane arhitekture predstavljaju okvir za integraciju poslovnih procesa i odgovarajuće ICT infrastrukture u komponente (servise, usluge) koje se mogu višestruko koristiti i međusobno kombinirati kako bi odgovorili zahtjevima poslovanja u dinamičnom okruženju.

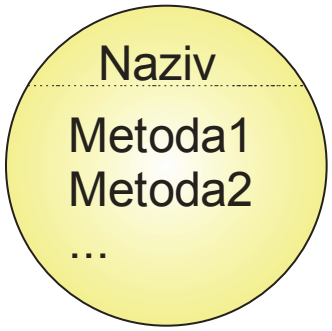
SOA – model

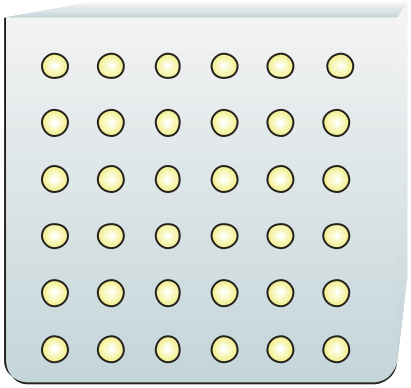

Registar usluga
(Service Registry)
pronalazi
objavljuje
Tražitelj usluga
(Service Requestor)
koristi
Pružatelj usluga
(Service Provider)

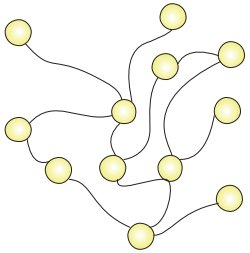
SOA – model

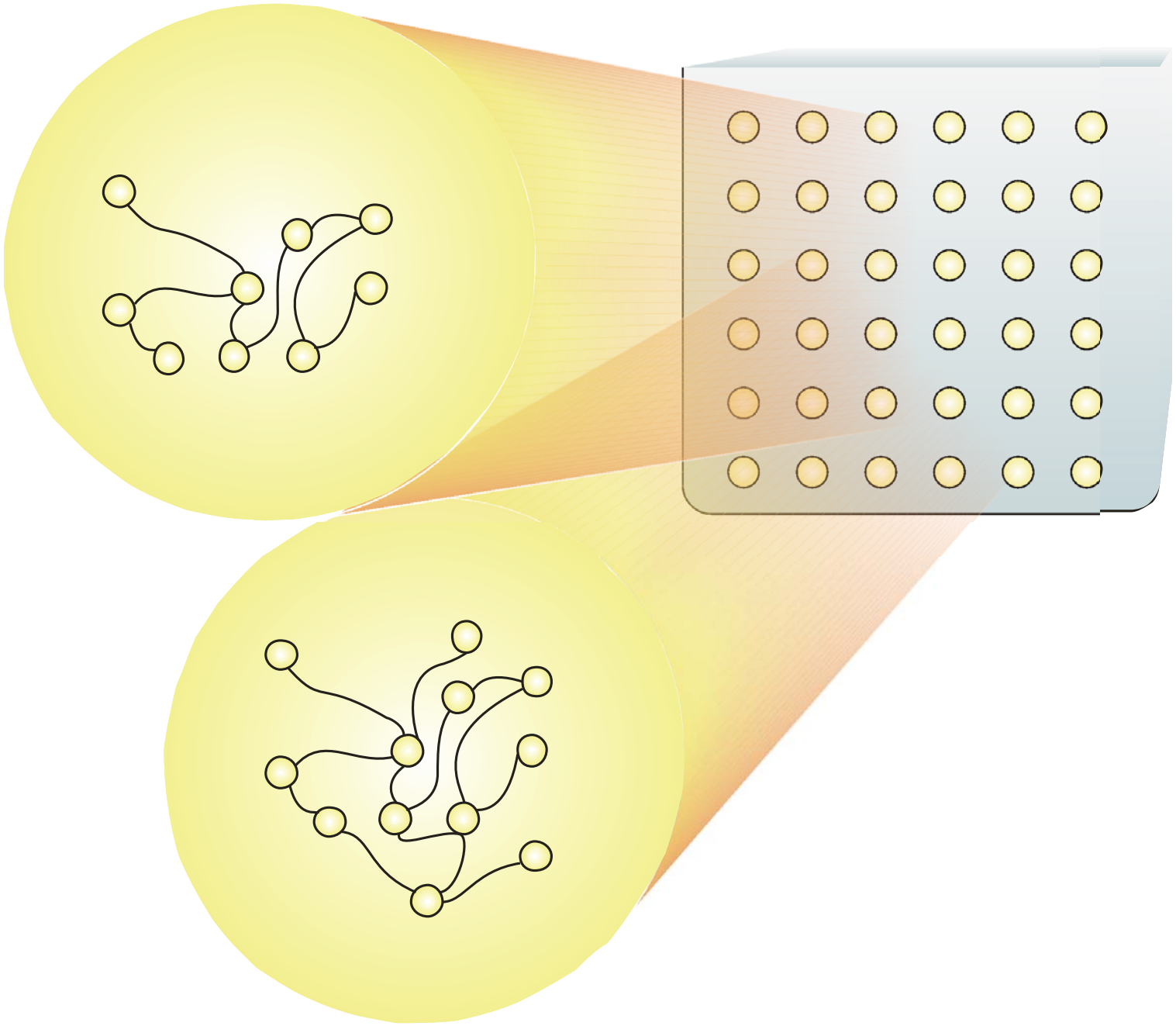

5. SLOJ PRISTUPA/ PREZENTACIJE
4. SLOJ KOREOGRAFIJE
3. SLOJ USLUGA
2. SLOJ KOMPONENTI
1. SLOJ OPERACIJSKOG SUSTAVA
6. INTEGRACIJA
7. OSIGURANJE KVALITETE USLUGA, SIGURNOST, UPRAVLJANJE, NADZOR

SOA komopozicija







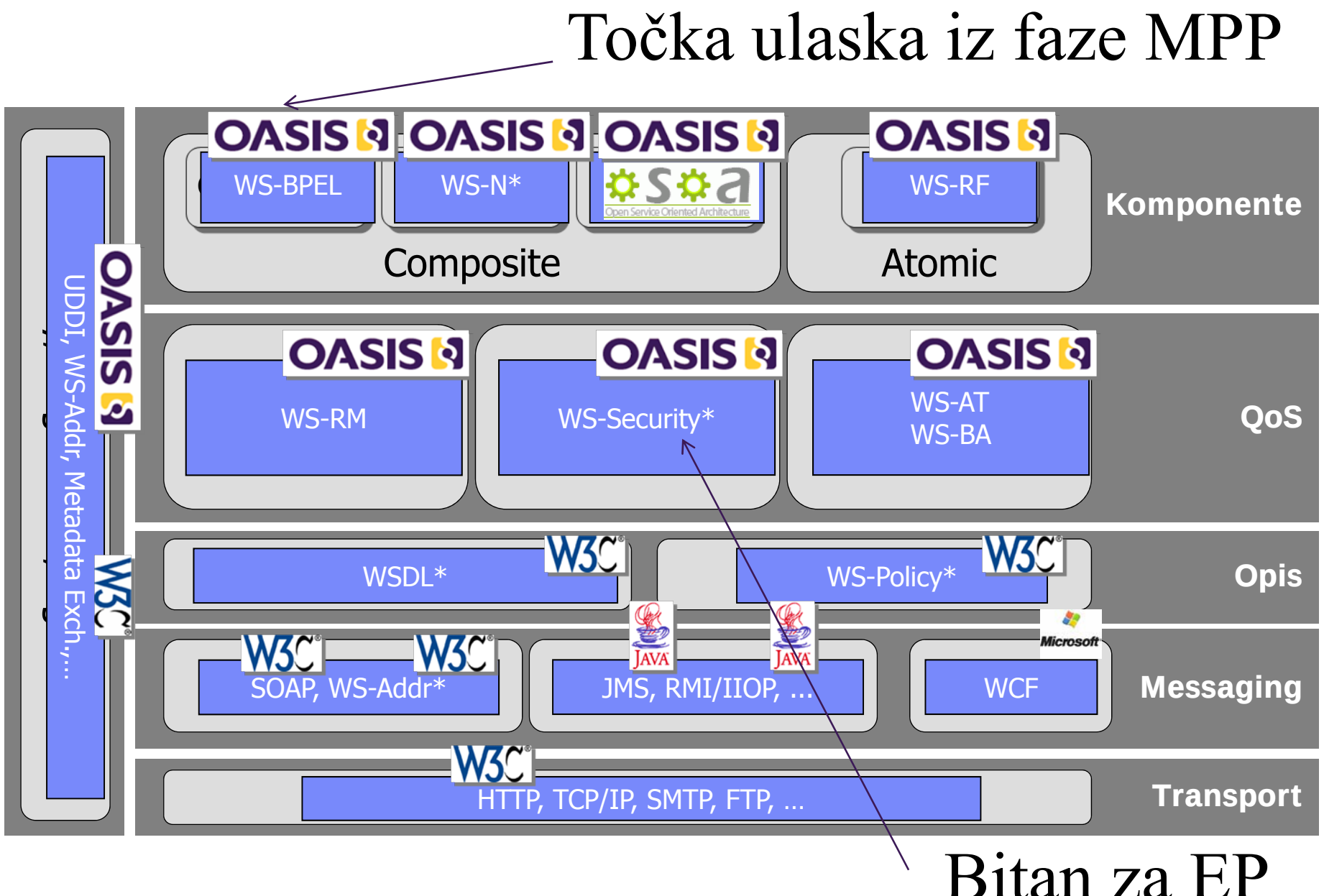


Standardi

- Toliko složena arhitektura ne moža raditi bez standarda
- Temelj interoperabilnosti – procesne, semantičke i podatkovne
- Ključni za povezivanje različitih subjekata
- Kako povezati model poslovne tehnologije s informacijskim sustavom?

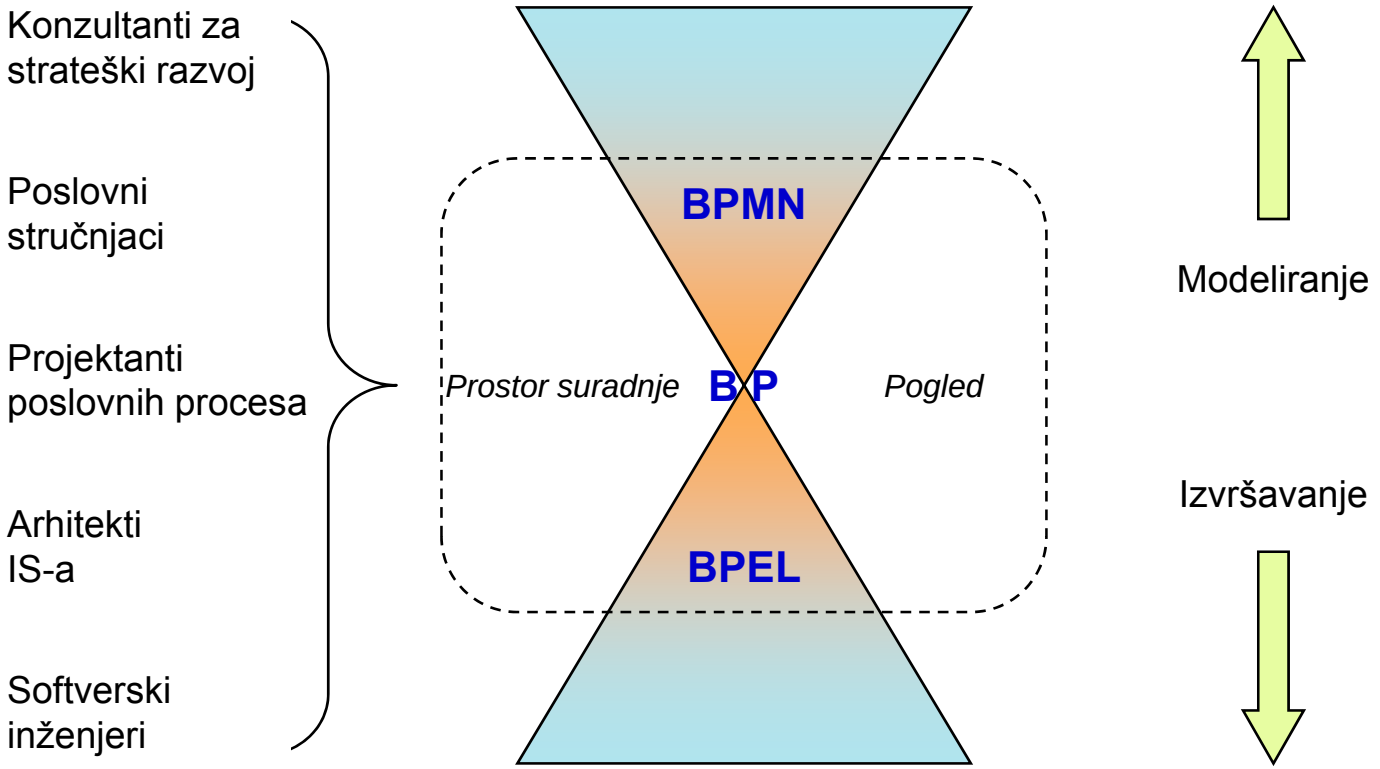
- **Izrazito složen standardizacijski okvir**

SOA standardi



Uloge kod BPM

Korisnici Poslovno okruženje Svrha

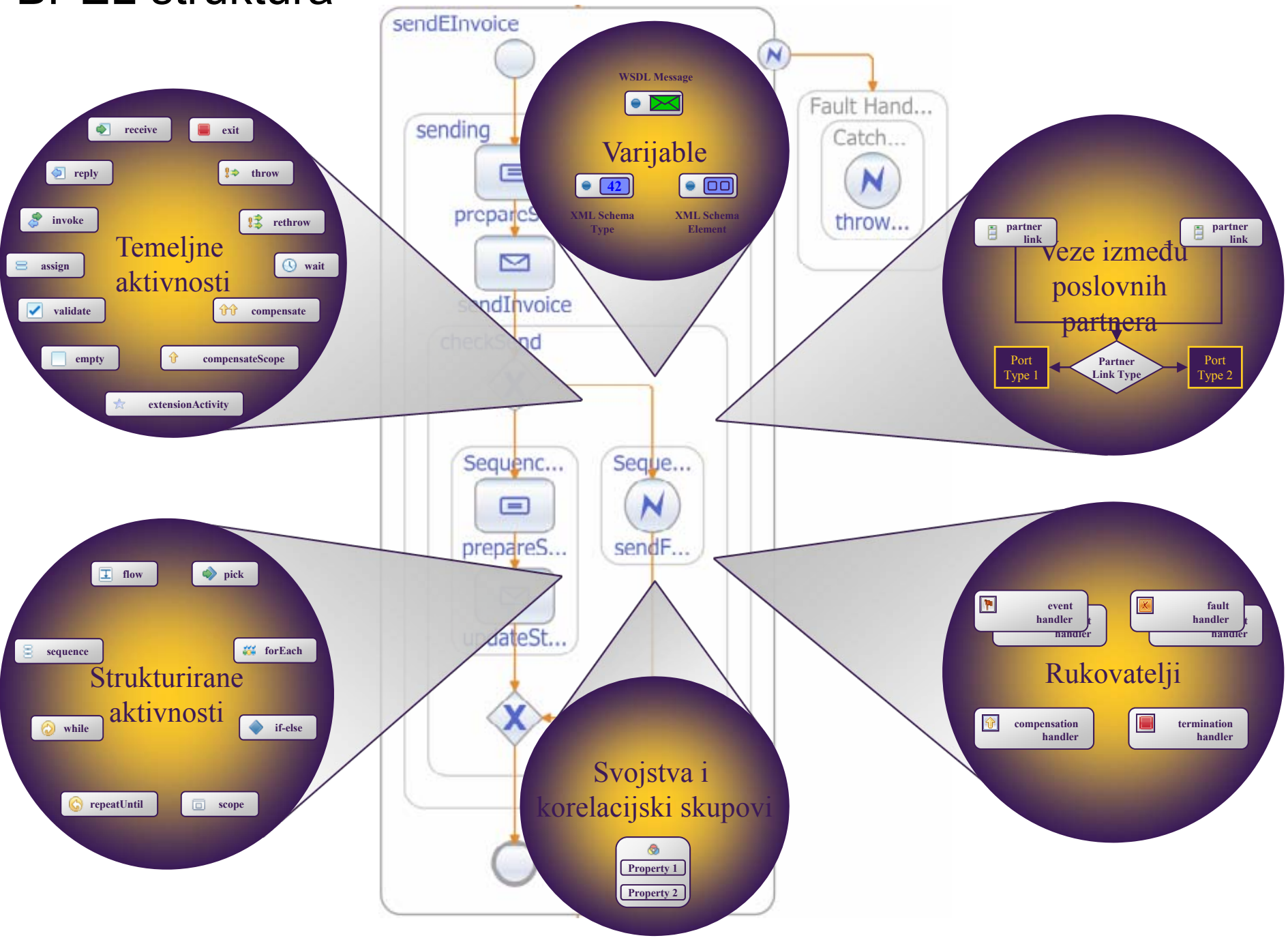


Primjena ICT

Značenja:
BPMN-Business Process Modeling Notation
BPEL-Business Process Execution Language

Prema: Stephen A. White
BPM Architect, IBM

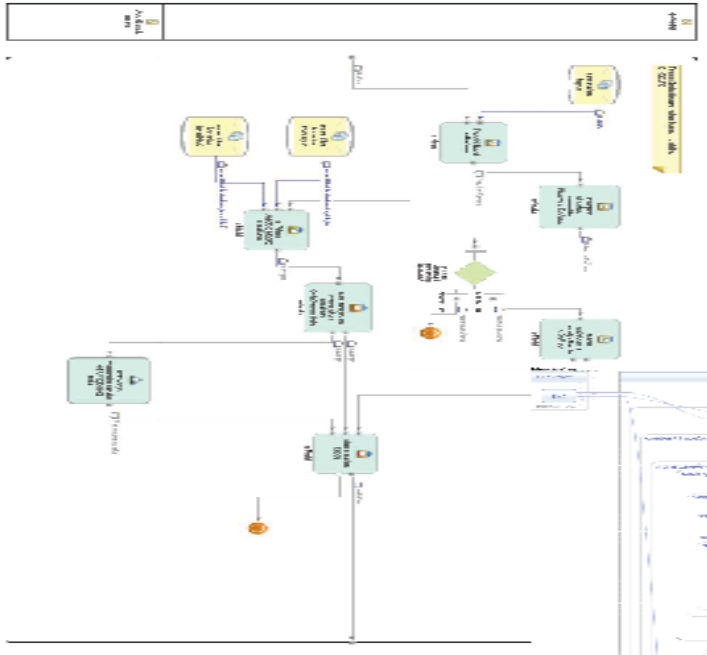
BPEL struktura



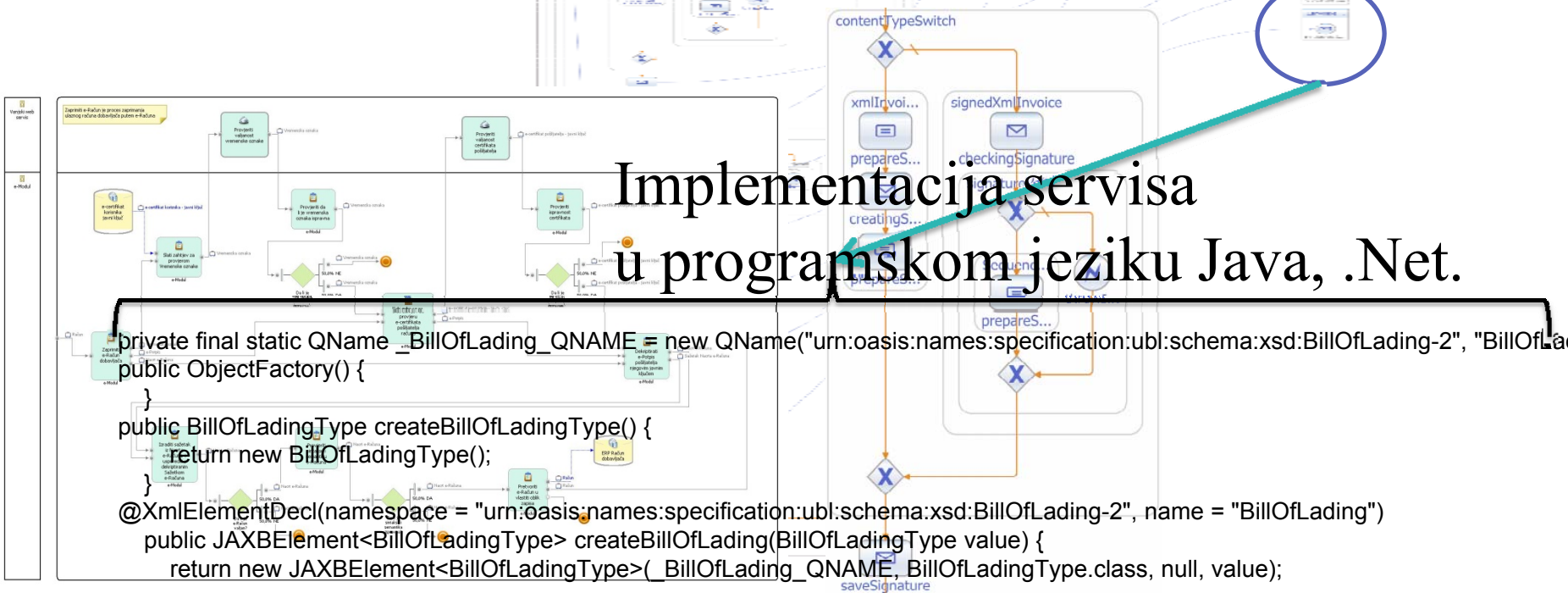
Preslikavanje procesnih aktivnosti u servise

Aktivnosti u servise

Podatkovni i kontrolni tokovi u orkestracijske tokove



Procesna logika postaje servisna logika pri čemu su moguće rekombinacije servisa u različite strukture što omogućuje fleksibilnost razvijene arhitekture. Stoga jedan skup servisa može podržati više načina izvođenja određenog poslovnog procesa što se i pokazalo na primjeru eRačuna.



Implementacija servisa u programskom jeziku Java, .Net.

```
private final static QName _BillOfLading_QNAME = new QName("urn:oasis:names:specification:ubl:schema:xsd:BillOfLading-2", "BillOfLading");
    public ObjectFactory() {
    }
public BillOfLadingType createBillOfLadingType() {
    return new BillOfLadingType();
    }
@XmlElementDecl(namespace = "urn:oasis:names:specification:ubl:schema:xsd:BillOfLading-2", name = "BillOfLading")
    public JAXBElement<BillOfLadingType> createBillOfLading(BillOfLadingType value) {
        return new JAXBElement<BillOfLadingType>(_BillOfLading_QNAME, BillOfLadingType.class, null, value);
```

Zašto nam uopće treba WS-BPEL?

- WSDL bazirani servisi imaju “stateless” koncepciju
  - Poruke se izmjenjuju
    - Sinkronim pozivima
    - Asinkronim pozivima
- Postoje i daleko složeniji modeli interakcije
- WS-BPEL omogućuje razradu dugotrajnih “stateful” transakcija
- WS-BPEL omogućuje agregaciju Web servisa korištenjem poslovnog modela koji se ponovno mogu agregirati u složenije komponente

Dvorazinski razvojni model

- Krupni pogled
  - Razvoj procesa – poslovni stručnjaci i informatičari
    - Tijek logike, definiranje temeljnih funkcija
- Detaljan pogled (programiranje)
  - Programeri implementiraju servise

Definicija procesa preko WS-BPEL prema OASIS

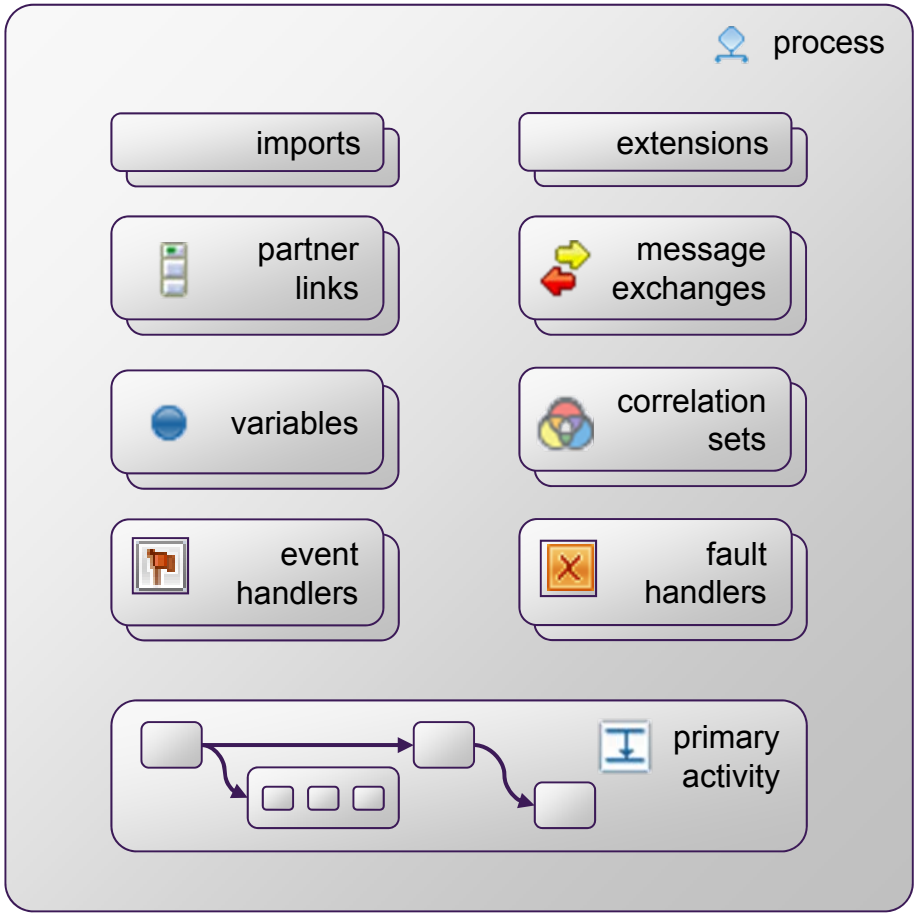
Deklarira zavisnost o vanjskoj XML shemi ili WSDL definiciji

Odnosi koje će WS-BPEL uspostaviti tijekom rada

Pohrana podataka unutar procesa ili u razmjeni između partnera

Konkurentna obrada ulaznih poruka ili vremenski okidanih događaja

Izvodi procesnu logiku i proizvoljan broj može biti rekurzivno ugniježđen

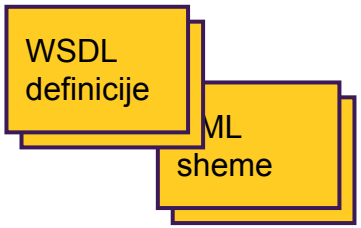


Imenički prostor WS-BPEL ekstenzija atibuta i elemenata

Odnos između dolaznih i odlaznih aktivnosti poruka

Polja s podacima koji identificiraju konverzaciju

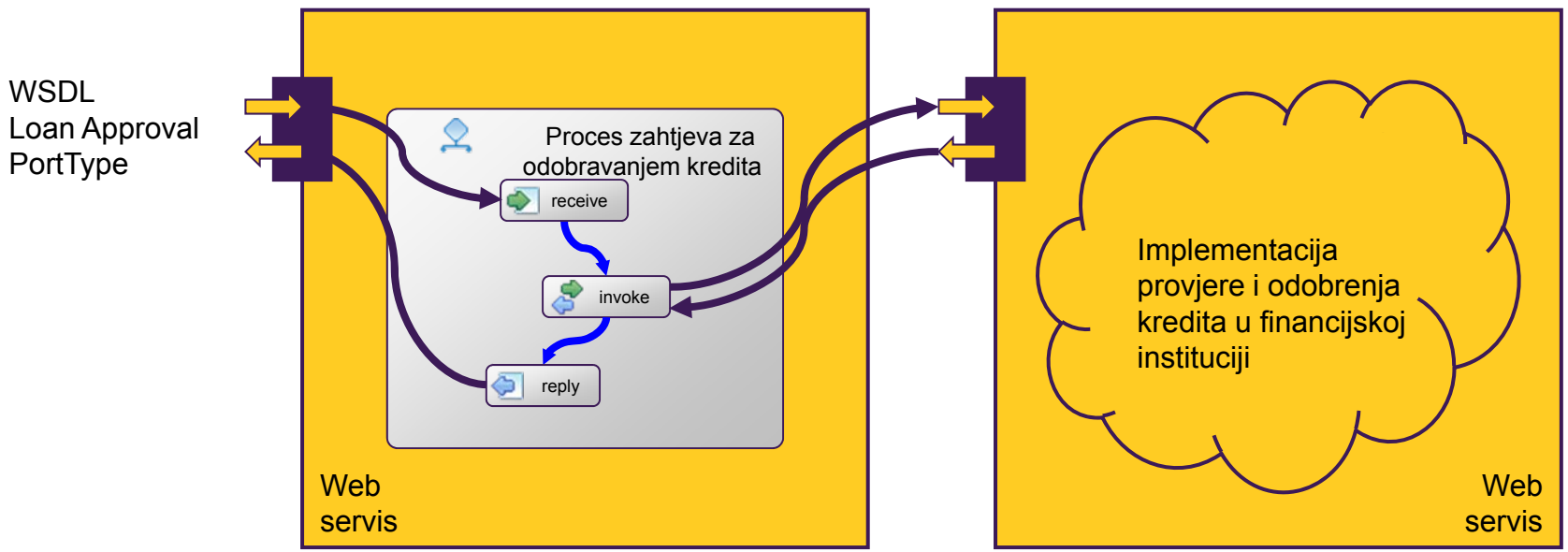
Rad s izuzecima u procesu



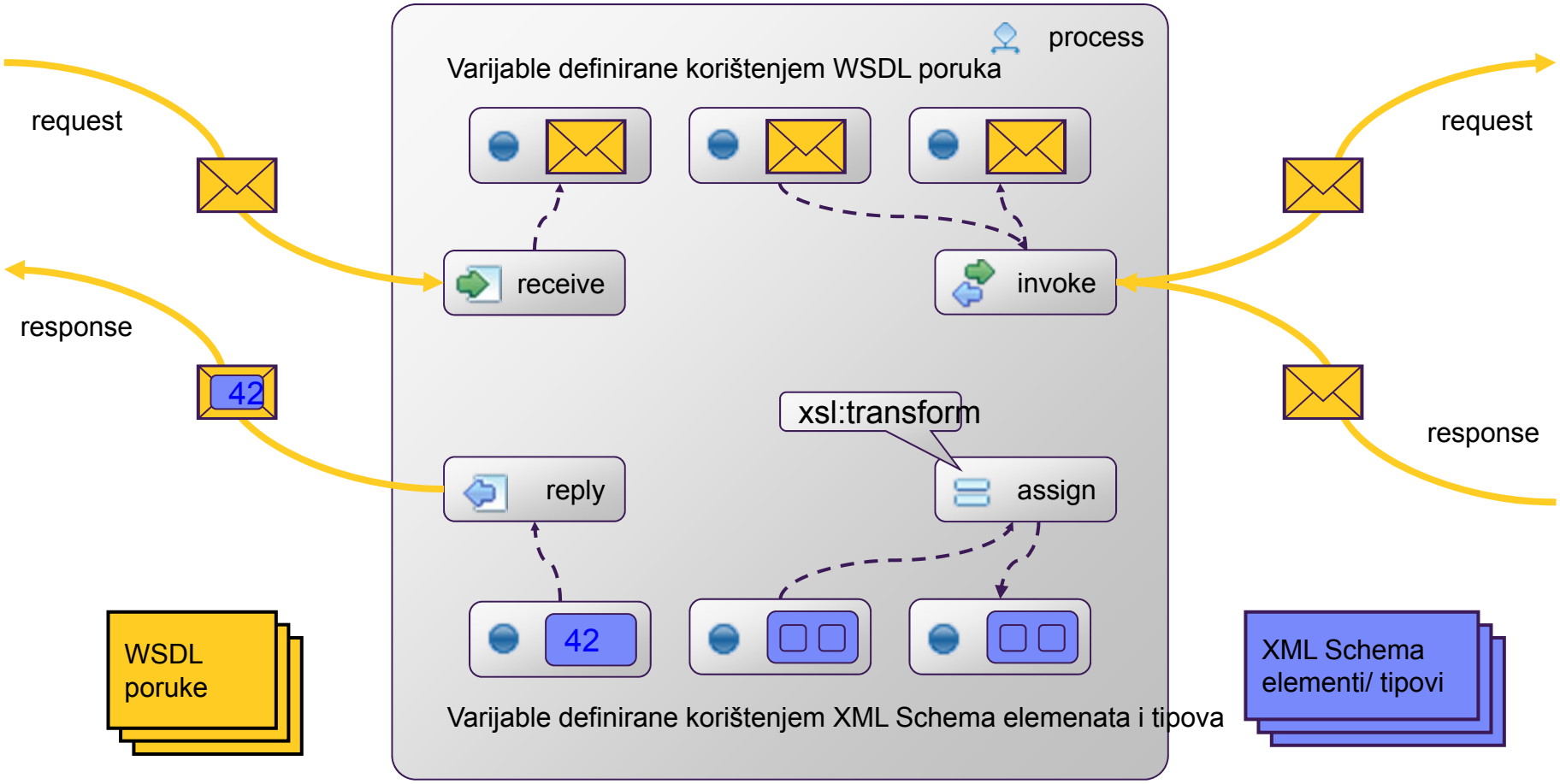
Recursive Composition Model

WS-BPEL procesi su prezentirani kao Web servisi prema poslovnim partnerima

WS-BPEL procesi su u interakciji s Web servisima poslovnih partnera pri čemu trajanje transkacije može biti relativno dugo

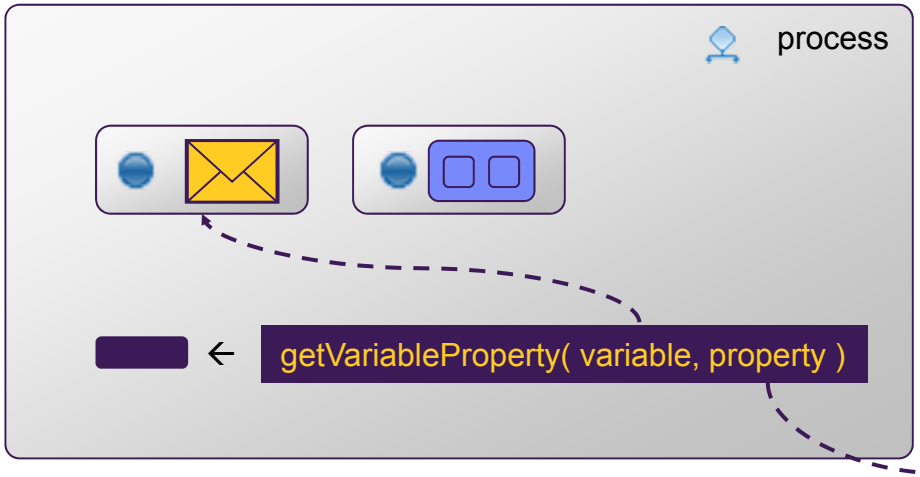


Varijable

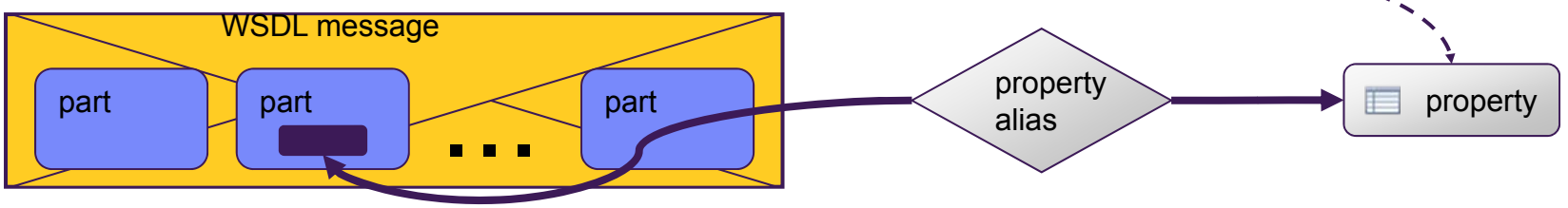

process
Varijable definirane korištenjem WSDL poruka
request
response
42
receive
invoke
request
response
xsl:transform
reply
assign
42
WSDL
poruke
Varijable definirane korištenjem XML Schema elemenata i tipova
XML Schema
elementi/ tipovi

Svojstva varijabli

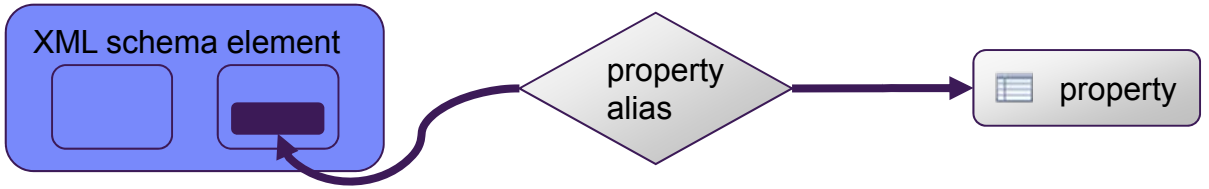
Svojstvo kreira ime koje ima semantičko značenje nad tipom definiranim XML shemom.



Svojstva izoliraju procesnu logiku od detalja definicije varijabli.

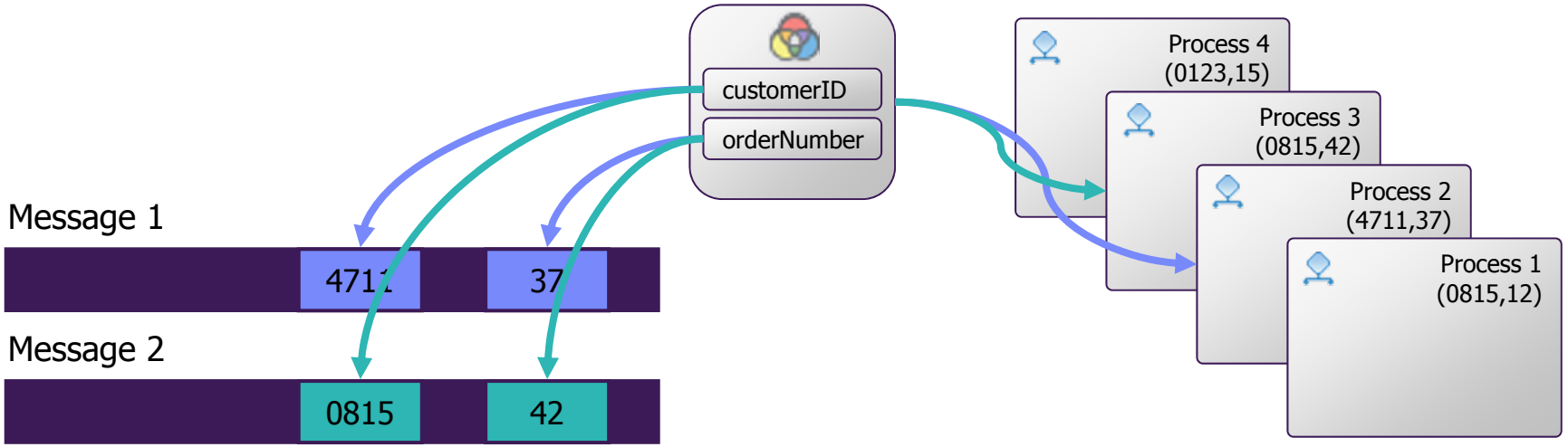


Svojstva se mapiraju na dijelove WSDL poruka ili elemente XML shema.

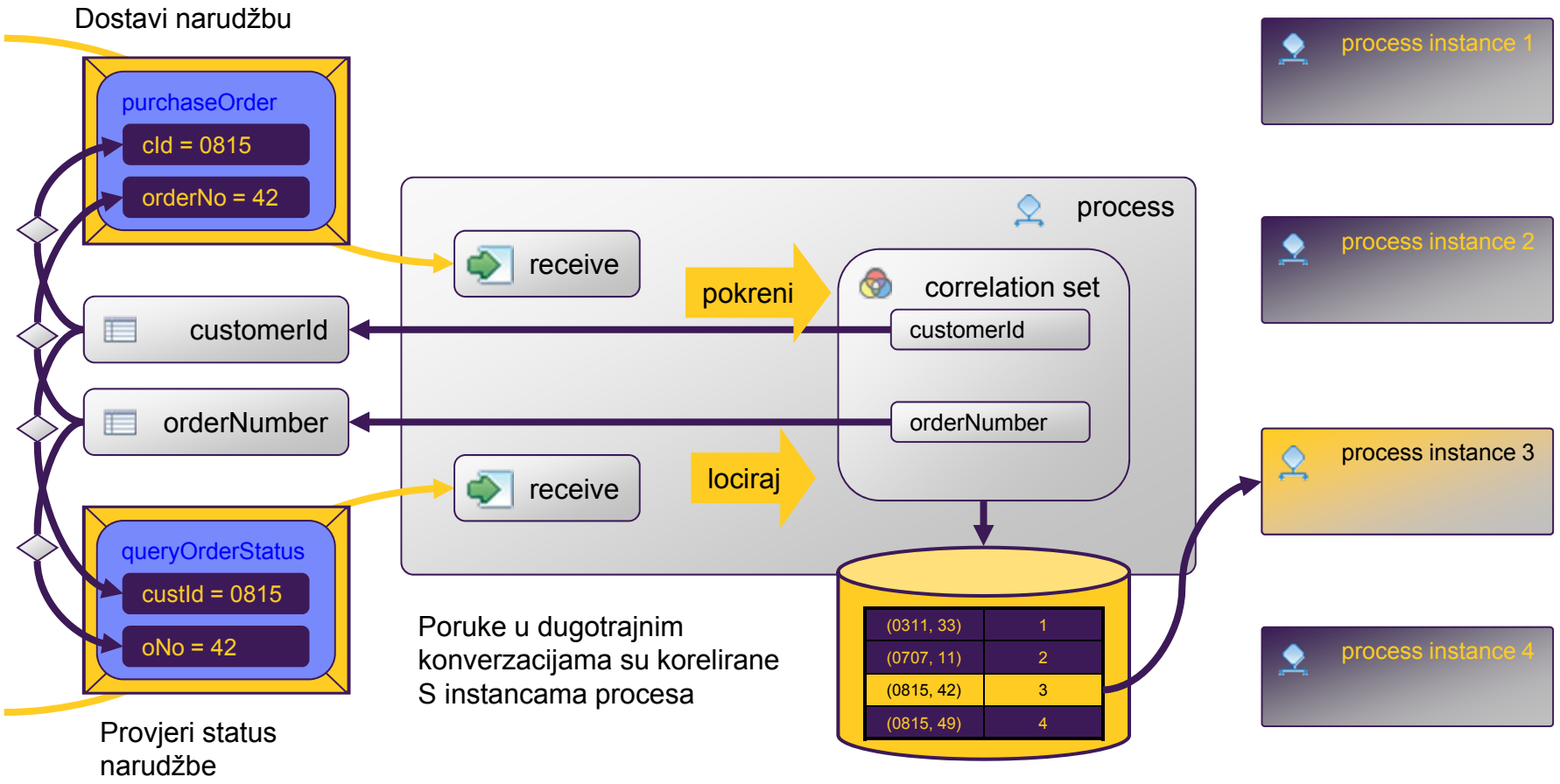


Svojstva i korelacijski skupovi

- Kako identificirati “stateful” procesne instance preko “stateless” WS sučelja?
- Instanci procesa se dodjeljuje jedan ili više ključeva
  - Poslovni podataka se koristi kao ključ npr. IDkupca
  - Ključ može biti i složen npr. IDkupca i IDNar
  - WS-BPEL naziva ključ korelacijskim skupom – tj. koristi se da bi se dolazeća poruka povezala s instancom procesa



Svojstva i korelacijski skupovi



Temeljne aktivnosti

Blokiraj tijek i čekaj da odgovarajuća poruka dođe. Pošalji odgovor na upit.

Pokreni request-response operaciju.

Ažuriraj vrijednosti varijabli ili veza s čvorovima/partnerima s novim podacima.

Provjeri XML podatke pohranjene u varijablama.

Generiraj grešku unutar poslovnog procesa.

Proslijedi pogrešku dalje unutar bloka obrade pogreške

Odmah obustavi izvršenje instance poslovnog procesa.

Pozovi kompenzacijsku proceduru za sve završene chile scopes

Pozovi kompenzacijsku proceduru za jedan završen child scope.

Čekanje na istek zadanog perioda.

No-op za poslovni proces.

Okvir za ekstenziju jezika

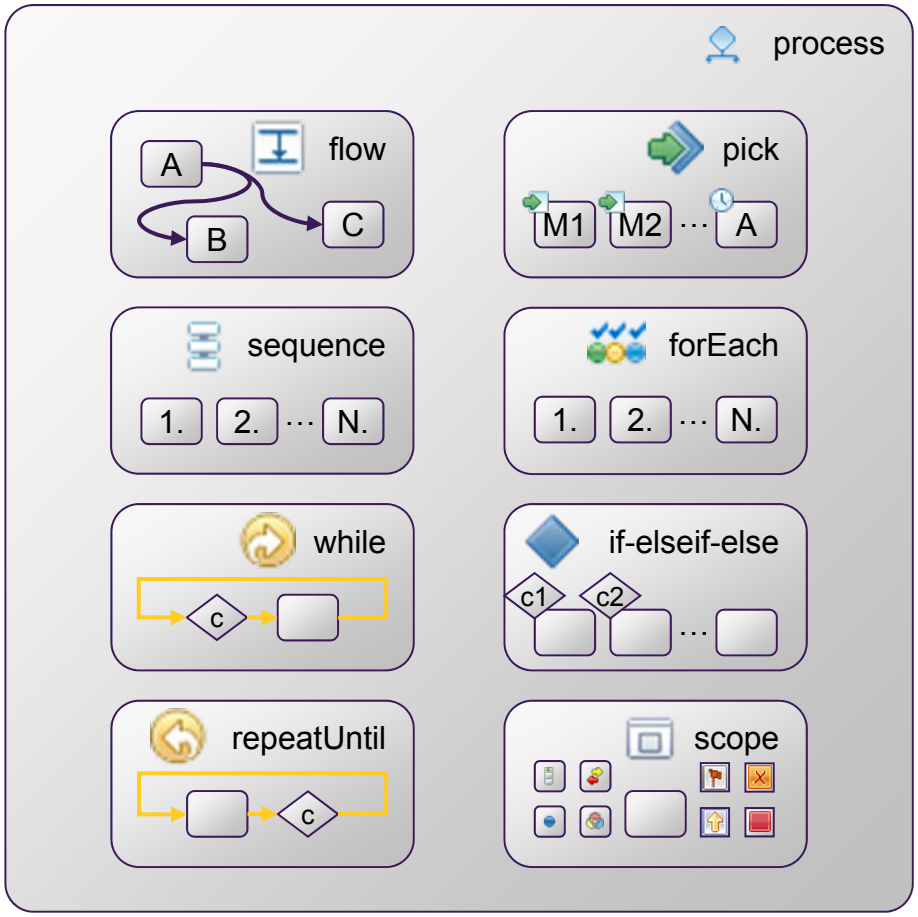
Strukturirane aktivnosti

Sadržane aktivnosti izvršavaju se paralelno ili slijedno upravljane kontrolnim tokovima.

Sadržane aktivnosti izvode se sekvencijalno prema leksičkom redoslijedu.

Sadržana aktivnost se ponavlja dok je ispunjen uvjet.

Sadržana aktivnost se ponavlja do ispunjenja uvjeta.



Zaustavi izvršenje i čekaj odgovarajuću poruku (ili time out)

Sadržana aktivnost se ponavlja sekvencijalno ili paralelno kontrolirana odgovarajućom varijablom

Odabir jedne grane od mogućih izbora

Povezivanje sadržane aktivnosti s njezinim lokalnim varijablama, partnerima itd.

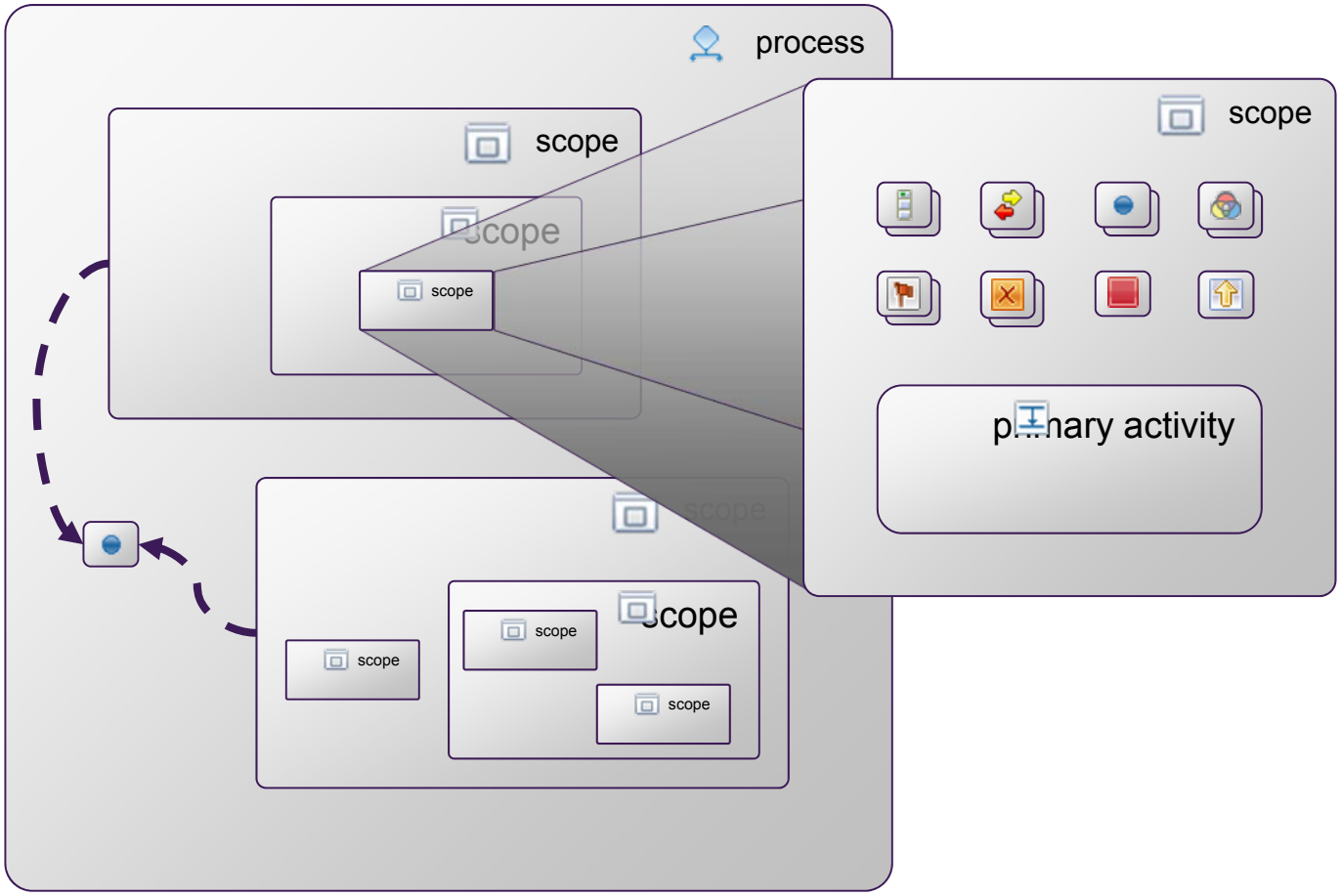
Doseg

Doseg (scope) opisuje kontekst izvršavanja aktivnosti koje obuhvaća

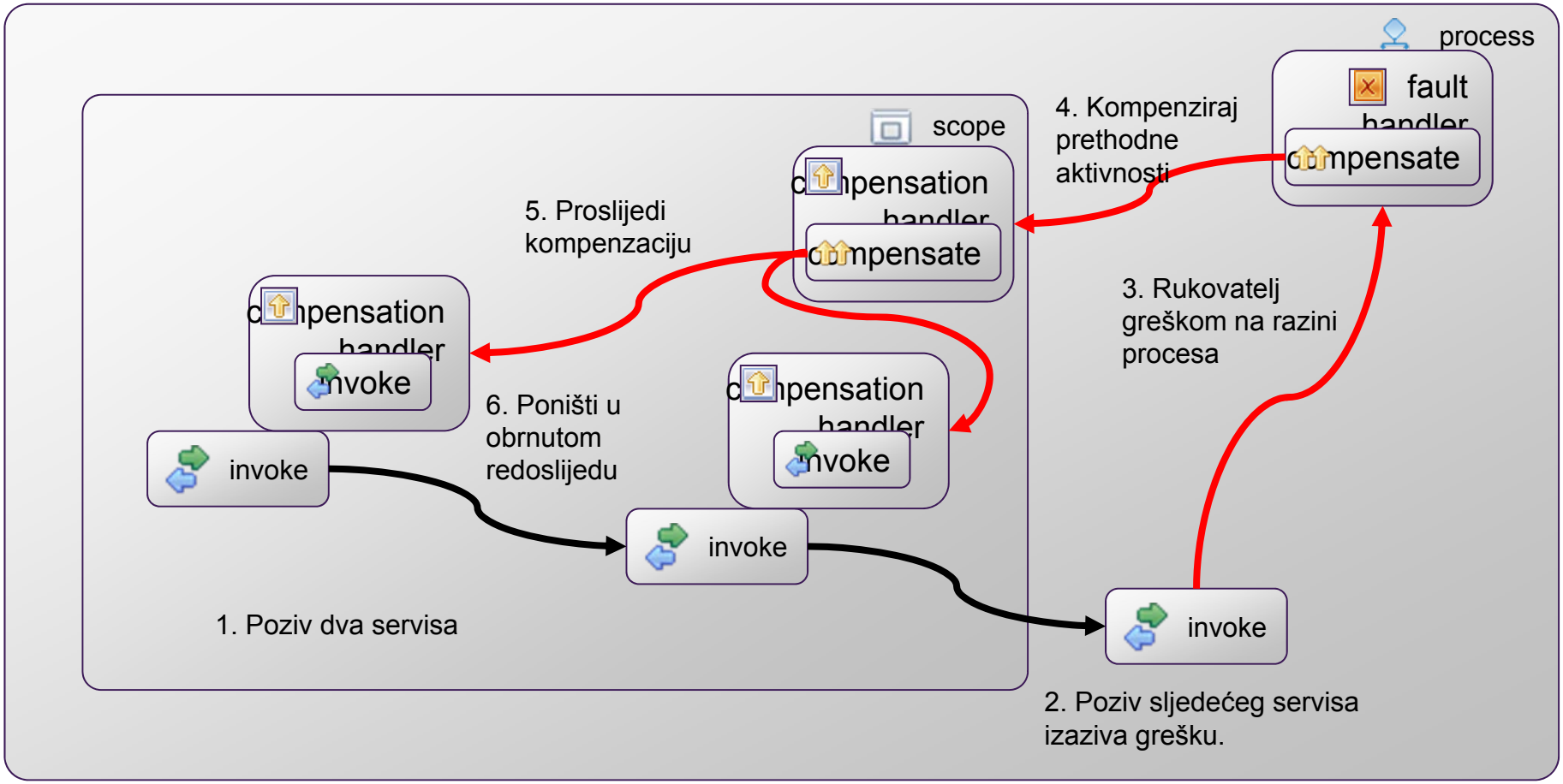
Lokalne deklaracije – partneri, varijable, razmjene poruka, korelacijski skupovi

Lokalni rukovatelji – događaji, greške, završetak, kompenzacija

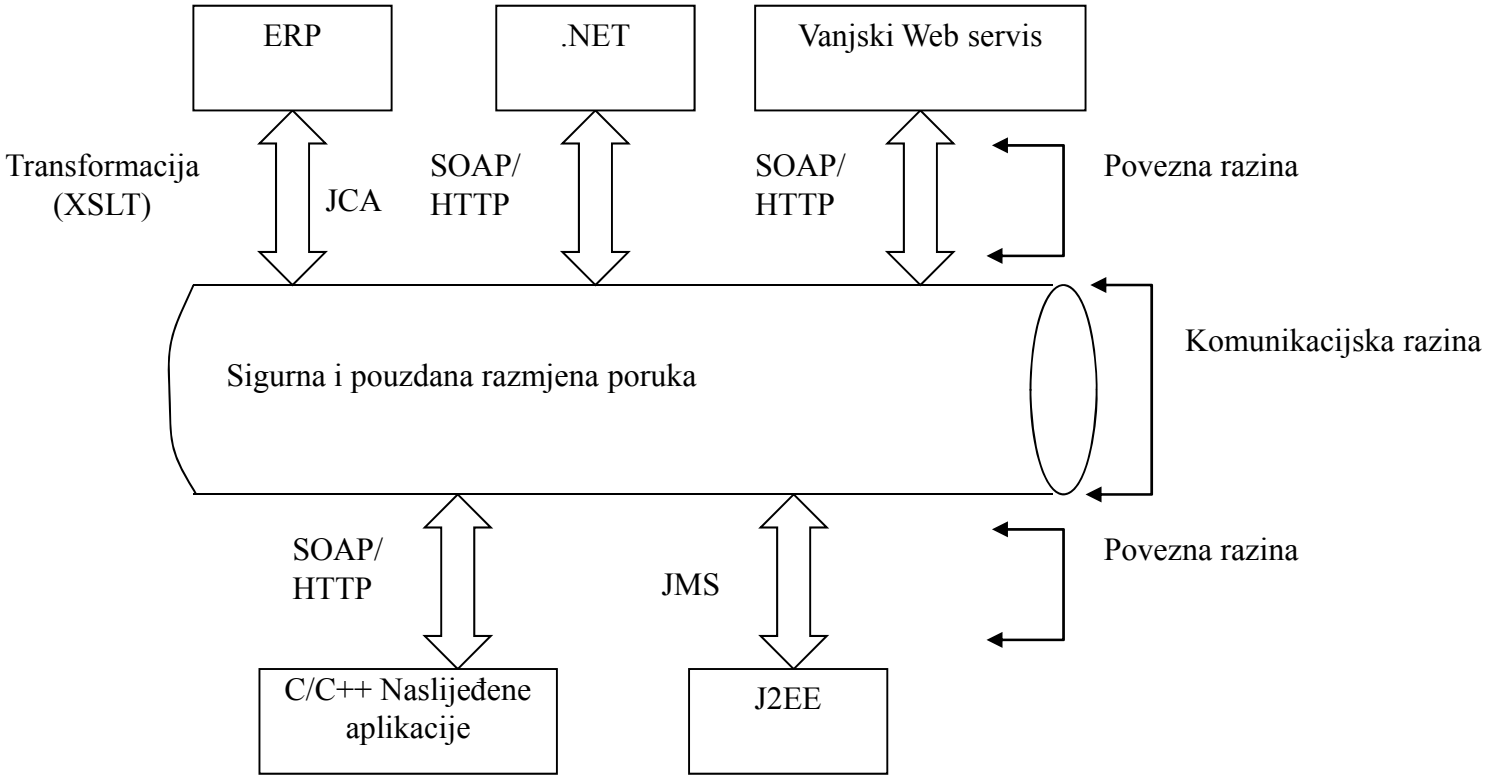
Izolirani dosezi omogućuju nadzor nad konkurentnim pristupom dijeljenim resursima



Kompenzacija



Poslovna sabirnica – Enterprise Service Bus



Komunikacija treba biti i sigurna

Temeljna sintaksa WS-Security standarda

- **<wsse:Security>** "obavija" sve sigurnosne informacije i oslanja se na niz drugih standarda
- Pozicija u zaglavlju poruke znači da se odnosi na cijelu SOAP poruku

```
<S: Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wsse:Security
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
      <wsse:UsernameToken>
        ...
      </wsse:UsernameToken>
      <EncryptedKey
            xmlns="http://www.w3.org/2001/04/enc-enc-enc#">
        ...
      </EncryptedKey>
      <Signature xmlns:"http://www.w3.org/2000/09/xmldsig#">
       ...
      </Signature>
    </wsse:Security>
  </S:Header>
```

Definicija i korištenje WS-Security imeničkog prostora

Autentikacija

Zaštita preko enkripcije

Integritet poruke preko digitalnog potpisa

Autentikacija

- **WS-Security definira sigurnosne oznake (token) koji mogu sadržavati različite zahtjeve prema zaštićenim servisima**
  - **Npr. korisničko ime i opcionalna zaporka, Kerberos ulaznica, X.509 certifikat**
- **Postoje dva tipa oznaka:**
  - **`UsernameToken` and `BinarySecurityToken`**
- **`UsernameToken` je najjednostavniji. Sadrži obvezno korisničko ime i opcionalnu zaporku.**
- **`BinarySecurityToken` sadrži enkodirane binarne podatke pogodne za pohranu X.509 certifikata ili Kerberos ulaznice**

Primjer korištenja username/password oznaka

```
<S: Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wsse:Security
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
        soapenv:mustUnderstand="1">
      <wsse:UsernameToken wsu:ID="mojaOznaka">
        <wsse:Username>Neven</wsse:Username>
        <wsse:Password>passw0rd</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

Drugi dijelovi poruke mogu se referencirati na taj UsernameToken preko ovog ID-a

Pozivatelj servisa dostavlja korisničko ime i zaporku poslužitelju kako bi se autenticirao. Poslužitelj mora razumjeti ovo zaglavlje ili vraća poruku greške.

- Slanje korisničkih podataka na ovaj način nije sigurno
- Mora se koristiti s WS-Security enkripcijom ili preko sigurnog kanala

Upravljanje pristupom

- X.509 i Kerberos
- Središnji čvor
  - Certificate authority (X.509)
  - Key Distribution Centre (Kerberos)
- U jedinici državne uprave
- Kritična točka sustava
- Visoko opterećenje

Kerberos zaglavlje (1)

```
<S: Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wsse:Security
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
        soapenv:mustUnderstand="1">
      <wsse:BinarySecurityToken wsu:ID="myToken"
        ValueType="wsse:Kerberosv5ST"
        EncodingType="wsse:Base64Binary>
        CGHJKOz88ZUbolfgImmJZc1 ..
      </wsse:BinarySecurityToken>
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

Korištenje Kerberos ulaznice

Binarni podaci Kerberos ulaznice se enkodiraju kao Base64

Kerberos ulaznica ili x.509 certifikat se smatraju potpisanim sigurnosnim elementima jer ih obično potpisuje određen autoritet

- Obzirom da treća strana može presresti certifikat ili ulaznicu i uključiti ju svoj poziv, samo certifikat ili ulaznica nisu dovoljan jamac za autentikaciju

XML enkripcija

- XML Encryption standard definira načine za enkripciju cijelih ili dijelova XML dokumenta
- Dozvoljena je enkripcija različitih dijelova istog dokumenta s različitim ključevima
- Moguće je kriptirati cijeli dokument ili jedan element

XML enkripcija

- **<EncryptedKey> element se smješta u sigurnosni dio zaglavlja**
  - **<EncryptionMethod>** Enkripcijski algoritam (simetrični)
  - **<KeyInfo>** Identifikator javnog ključa podrazumijevajući da pošiljatelj i primatelj razumiju značenje identifikatora
  - **<CipherData><CipherValue>** Kriptirana vrijednost simetričnog ključa
  - **<ReferenceList>** lista **<DataReference>** sadržaja kriptiranih simetričnim ključem

Primjer enkripcije (zaglavlje)

```
<S: Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wsse:Security ...>
      <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm =
         "http://www.w3.org/2001/04/xmlenc#rsa-1_5">
        </EncryptionMethod>
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmlsig#">
          <wsse:SecurityTokenReference>
            <wsse:KeyIdentifier>u3AA1M+DMOA1bX/vWJ ...
            </wsse:KeyIdentifier>
          </wsse:SecurityTokenReference>
        </KeyInfo>
        <CipherData>
          <CipherValue>cdck0cWh94oF5xBoEm ... </CipherValue>
        </CipherData>
        <ReferenceList>
          <DataReference URI = "#mojaOznaka">
          </DataReference>
        </ReferenceList>
      </EncryptedKey>
```

Simetričan ključ je kriptiran RSA-1.5 algoritmom korištenjem javnog ključa kao što je navedeno niže

Identifikator (ne i sam ključ) javnog ključa

Kriptiran simetričan ključ

URI se referira na dio poruke kada se se simetrični ključ koristi u drugim dijelovima poruke

Tijelo poruke

```
<S:Body>
  <po xmlns: ...>
    <wsse:Security ...>
      <EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#"
      Id="mojaOznaka"
      Type="http://www.w3.org/2001/04/xmlenc#Content">
        <EncryptionMethod Algorithm =
         "http://www.w3.org/2001/04/xmlenc#tripledes-cbc">
        </EncryptionMethod>
        <CipherData>
          <CipherValue>Ew7Zggr8z3 ... </CipherValue>
        </CipherData>
      </EncryptedData>
      <shipTo>
        <company>FOI Varaždin</company>
        <street>Pavlinska 2</street>
        <postalCode>42000</postCode>
      </shipTo>
        :
  </po>
</S:Body>
```

Korištenje simetričnog ključa kao što je navedeno u zaglavlju za kriptiranje ovog dijela

Enkripcijski algoritam je triple-DES simetrični algoritam

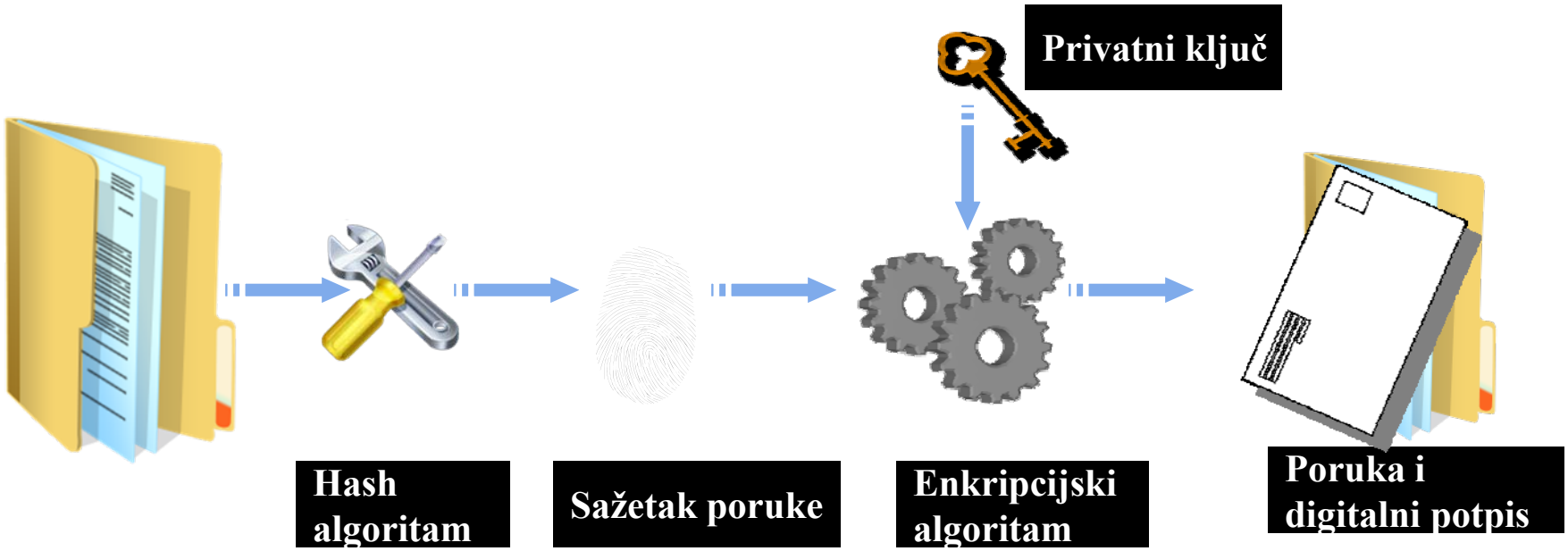
Kriptirani podaci npr. broj kreditne kartice

Dio koji nije osjetljiv nije kriptiran

W3C XML digitalni potpis

- XML digitalni potpis koristi se za potpisivanje strukturiranih digitalnih sadržaja poput:
  - XML elemenata
  - Eksternih URI-ja
  - Externih binarnih podataka
  - Binarnih podataka ugrađenih kao base 64 enkodiranih stringova unutar XML dokumenata
- Postoje 3 vrste XML digitalnih potpisa
  - Enveloped
  - Enveloping
  - Detached

Kreiranje digitalnog potpisa



XML Signature

- <Signature>
  - <SignedInfo>
    - <CanonicalizationMethod>
    - <SignatureMethod>
    - <Reference>
    - <SignatureValue>
  - <KeyInfo>

XML digitalni potpis

- Standard za kreiranje XML digitalnog potpisa definira pravila za kreiranje digitalnog potpisa i njegovu pohranu unutar XML dokumenta
- **<Signature>** element sastoji se od tri glavna dijela
  - **<SignedInfo>**

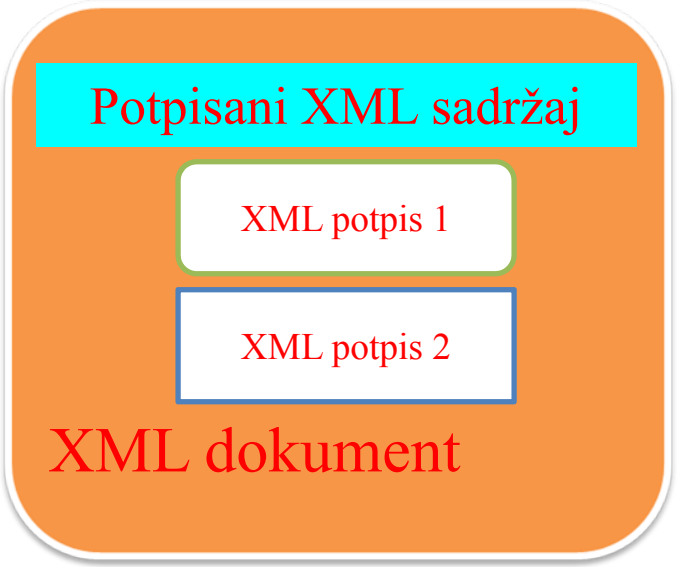
    Informacija o potpisanome (npr. algoritimi generiranje sažetaka i enkripcijski algoritimi)
  - **<SignatureValue>**

    Vrijednost digitalnog potpisa
  - **<KeyInfo>**

    Opcionalni javni ključ za provjeru potpisa
- W3C preporuka (http://www.w3.org/Signature)

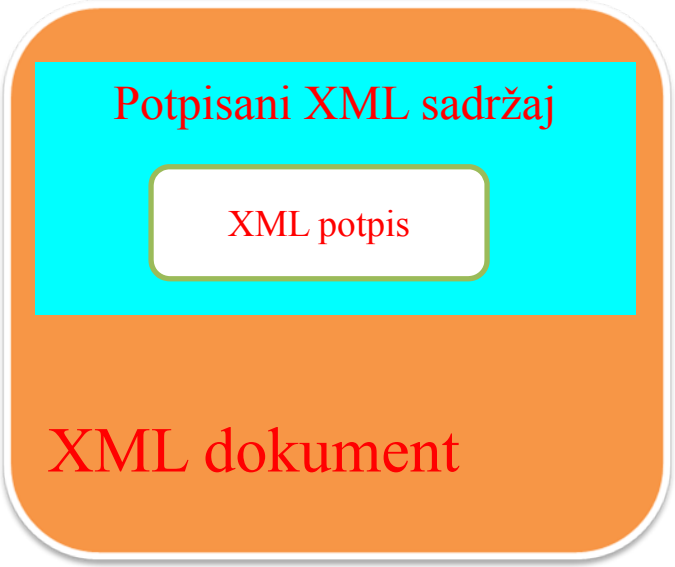
Enveloped XML digitalni potpis

Enveloped XML digitalni potpis kreira se na način da je potpis ugrađen u potpisani dokument.



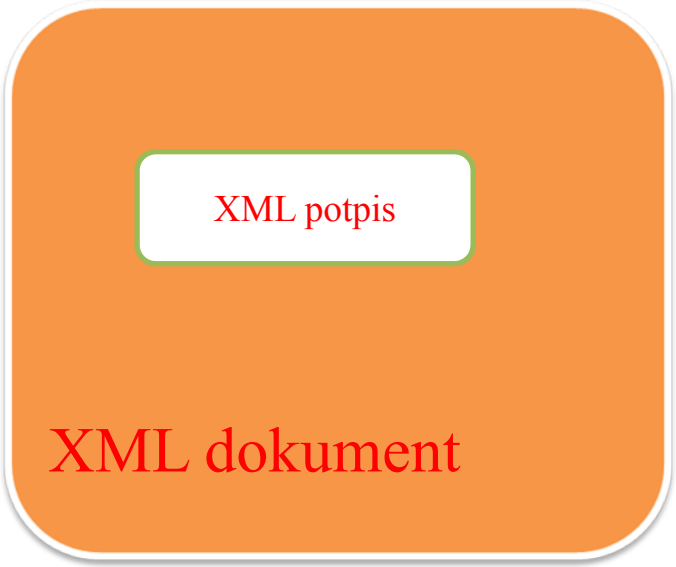
Enveloping XML Digital Signature

Enveloping XML digitalni potpis kreira se na način da je potisani sadržaj ugrađen unutar XML signature elementa.



Odvojeni (Detached) XML digitalni potpis

Odvojeni XML digitalni potpis je potpis gdje su potpisani sadržaj i XML digitalni potpis odvojeni.



Potpisani XML sadržaj

Ostali standardi

- Biraju se prema zahtjevima projekta
- Dio problema njihove implementacije rješavaju sofisticirane programske platforme
- Međutim kod detaljnijih razrada servisa i njihovih poveznica potrebno ih je ugrađivati izravno u programski kod

Zaključak

- SOA projekti su izuzetno složeni zbog cijelog niza tehnoloških i standardizacijskih zahtjeva kojima moraju udovoljiti
- Važno je odabrati i odgovarajuće standarde koji će jamčiti odgovarajuću kvalitetu usluge
- Razvija se cijela nova domena distribuiranih sustava (cloud, grid) koja jamči zanimljivu budućnost, ali i puno učenja novih koncepata