



Java Business Integration szolgáltatásalapú architektúra JavaEE környezetben

Simon Géza

geza.simon@sun.hu

Zsemlye Tamás

tamas.zsemlye@sun.com

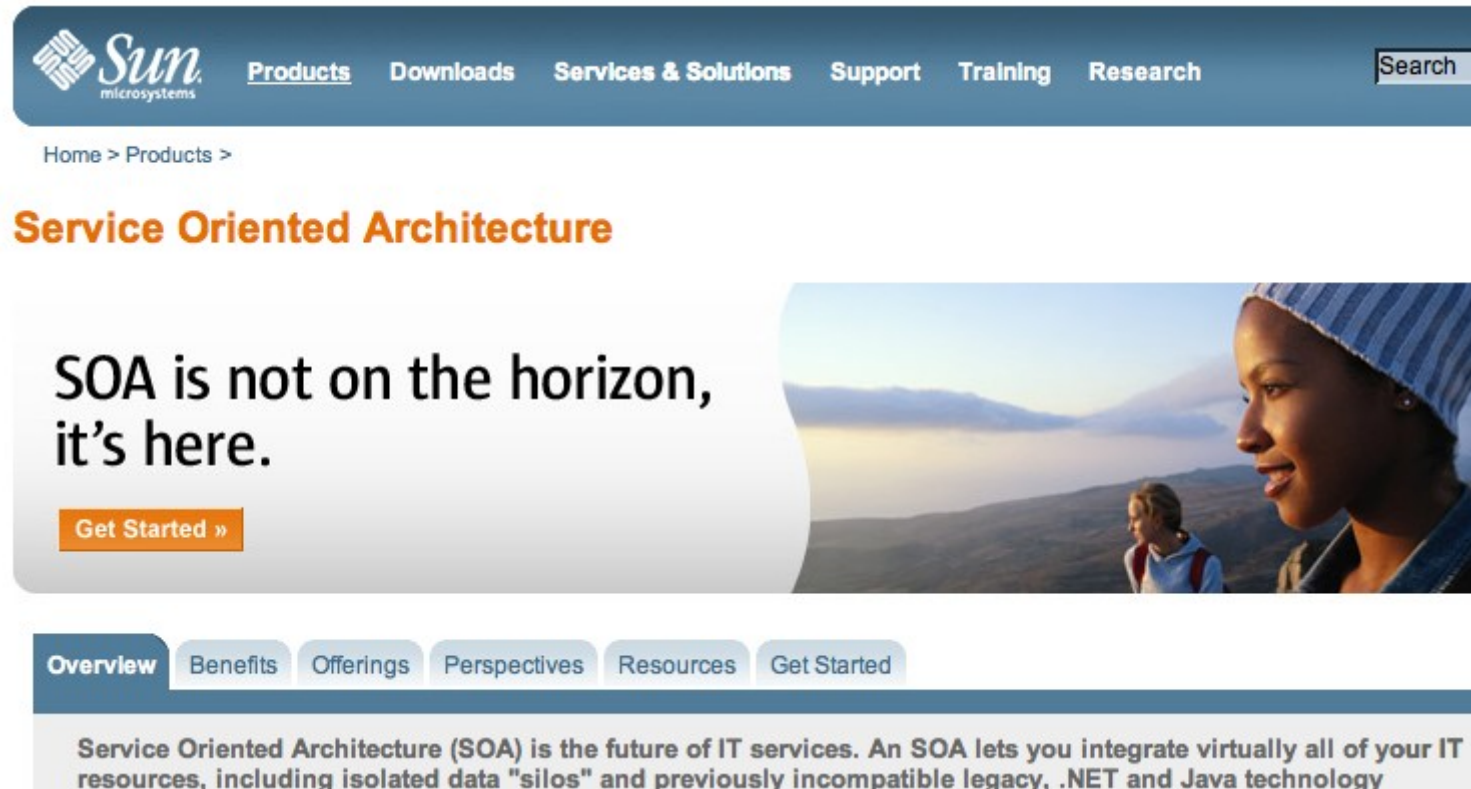


Témáim:

SOA architecture Webservice folyamat java
WS-addressing java **JavaEE** business process
execution **szinkron** aszinkron lazán csatolt
szolgáltatás **JB** integráció távoli metódushívás
dokumentum **XML** durva szemcsés orchestration
BPEL registry location transparent
relációs adatbázis MOM fejlesztőeszköz
szabványok NetBeans open **ESB**

SOA – Service Oriented Architecture

- Mi?
- Miért?
- Hogyan?



The screenshot shows the Sun Microsystems website page for Service Oriented Architecture. At the top, there is a navigation bar with the Sun Microsystems logo and links for Products, Downloads, Services & Solutions, Support, Training, and Research. A search box is located on the right. Below the navigation bar, the breadcrumb "Home > Products >" is visible. The main heading is "Service Oriented Architecture" in orange. The main content area features a large image of two people looking out over a landscape at sunset. Overlaid on the image is the text "SOA is not on the horizon, it's here." and a "Get Started »" button. Below the main content, there is a secondary navigation bar with tabs for Overview, Benefits, Offerings, Perspectives, Resources, and Get Started. The "Overview" tab is selected. Below the tabs, a paragraph of text reads: "Service Oriented Architecture (SOA) is the future of IT services. An SOA lets you integrate virtually all of your IT resources, including isolated data 'silos' and previously incompatible legacy, .NET and Java technology".

Mi a SOA?

- varázsszó, és azon kívül:
- tervezési elv (elosztott szoftverrendszerek)
- best practice (technológia- és platformfüggetlen)
- laza csatolás
- üzleti funkciók szolgáltatásként
- web service vagy bármi más (corba, rmi, ...)

Miért (kell/van/pont most) SOA?

- sok rendszer, bonyolultak
- heterogének

- integráció igénye
- pre-SOA (MOM, db-link, közös tábla, közös file, ...)

- gyorsan
- olcsón
- újrahasznosítható (eszköz / tudás)

Hogyan? SOA architektúrális szabályok

- megosztott szolgáltatások (durva szemcsézettség)
- leírás (WSDL) – regisztráció – megtalálás (find-bind-execute)
- többnyire **Aszinkron**
- párbeszédes (**C**onversational) (nem stateless)
- **DokumentumC**entrikus (xml) üzenetek
- megbízható, biztonságos
- vezénylés (orchestration)
- helyfüggetlen (location transparent)

Szolgáltatások:

Minden szolgáltatás!

- Transzformáció
- Át/irányítás
- Adatbázis/file írás/olvasás
- Üzleti logika végrehajtása
- Döntéshozatal (szabálykiértékelés)
- ...
- XSLT, EJB, BPEL, JDBC, MQ, ...

Web services

- Lazán csatolt
- Find – Bind – Execute
 - > Automatizálható felkutatás (**find**): UDDI
ehhez valahová regisztrálni kell (registry)
 - > Automatizálható kapcsolódás (**bind**): WSDL
ehhez interface-leírás kell (discovery)
 - > Automatizálható végrehajtás (**execute**): SOAP
- Megbízható, biztonságos
 - > WS-security
 - > WS-*

ACDC

Aszinkron: a kliens (hívó) szemével

- skálázhatóság
- hosszú folyamatok

Conversational: nem kell az állapotot (nagy xml)

- de/serializálni
- küldözgetni
- titkosítani

Document centric:

- a folyamatokhoz illeszkedik (opentravel)

JAX*: nemA, nemC, nemDC

Állapotmentes

Szinkron

RPC-centrikus

- lehet kézzel fejleszteni, vagy
- **BPEL**

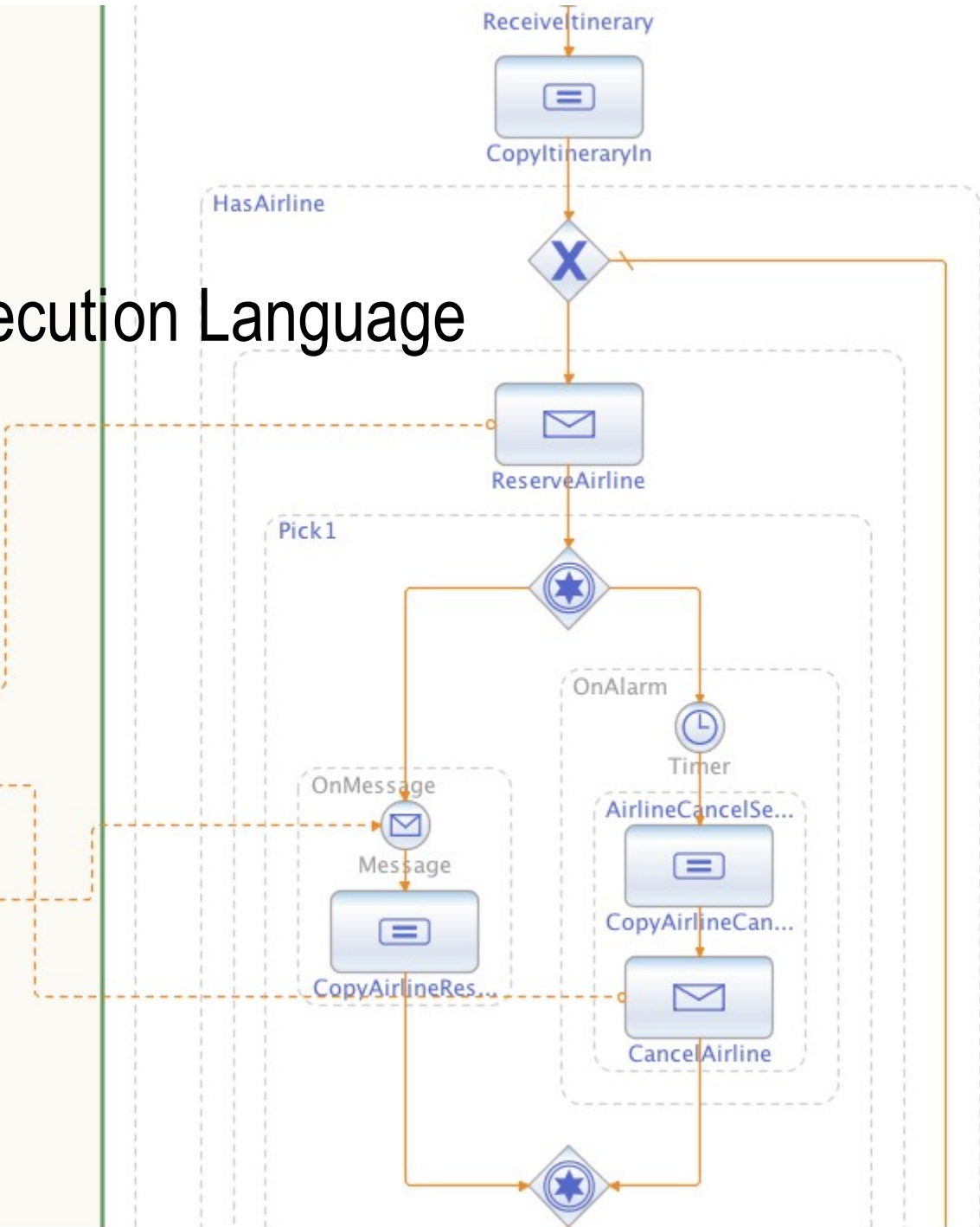
BPEL

Business Process Execution Language

- orchestration

- ACDC

- (nem coreography)



BPEL

BPEL4WS 1.0

IBM

WS-BPEL 2.0

OASIS-OPEN.org

(2007. feb.)

```
<flow>

  <links>
    <link name="ship-to-invoice" />
    <link name="ship-to-scheduling" />
  </links>

  <sequence>
    <assign>
      <copy>
        <from variable="PO" part="customerInfo" />
        <to variable="shippingRequest"
            part="customerInfo" />
      </copy>
    </assign>

    <invoke partnerLink="shipping"
            portType="lns:shippingPT"
            operation="requestShipping"
            inputVariable="shippingRequest"
            outputVariable="shippingInfo">
      <source linkName="ship-to-invoice" />
    </invoke>

  </sequence>
</flow>
```

Ismétlés: architekturális szabályok

- megosztott szolgáltatások (durva szemcsézettség)
- leírás (WSDL) – regisztráció – megtalálás (find-bind-execute)
- többnyire **Aszinkron**
- párbeszéd (conversation, stateless)
- **Document Center** (xml) üzenetek
- megosztott biztonság
- vezénylés (orchestration)
- **helyfüggetlen (location transparent)**

Location transparency: ESB

A folyamatoknak ne kelljen ismerni a hívott szolgáltatás

- helyét (URL)
- technológiáját

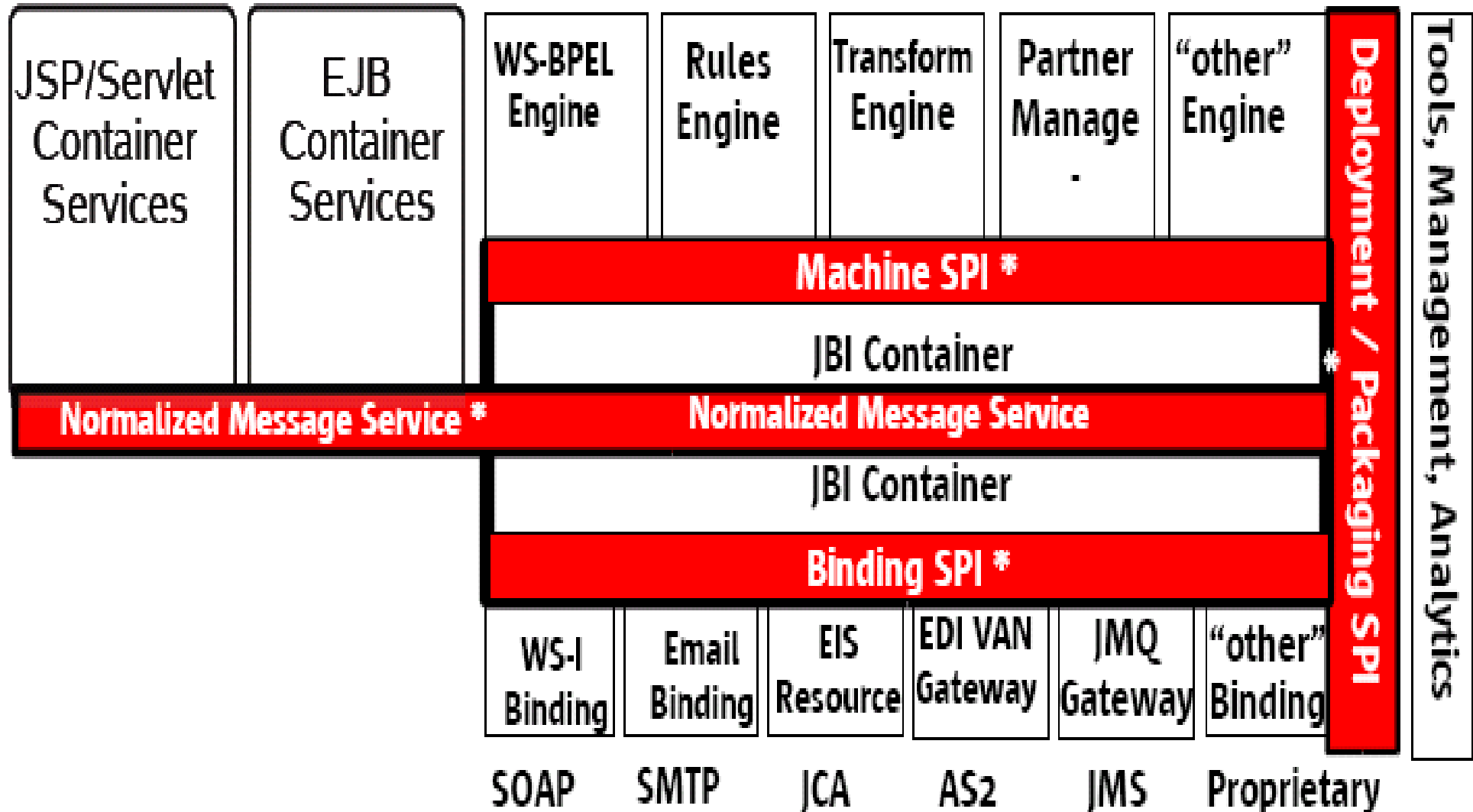
- elég a buszt megtalálni
- normalizált üzenetek

Java Business Integration: JBI

- Java Community Process (jcp.org)
- JSR-208, 2005. augusztus

- javax.jbi.*
- JavaEE vagy JavaSE
- JMX

Java Business Integration



* Defined by JSR 208

JBI komponensek

Enterprise Service Bus

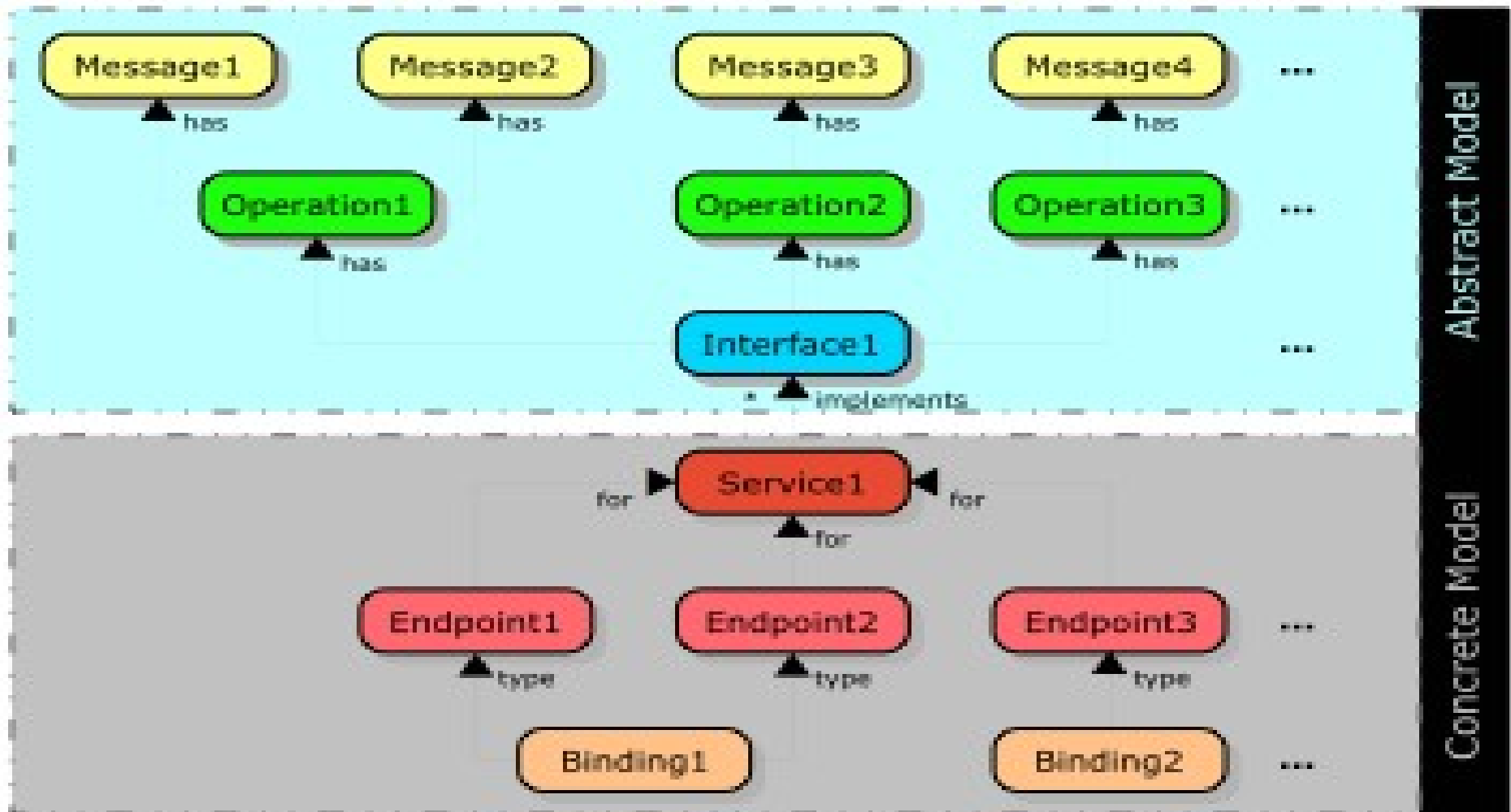
- 2 SPI:
 - > Service Engine-ek
 - > Binding Componentek

- +2 SPI:
 - > deploy (service unit – service assembly): composite application
 - > WSDL-alapú route-olás

A programozó tipikusan nem is látja!

Normalized message service

WSDL-alapú

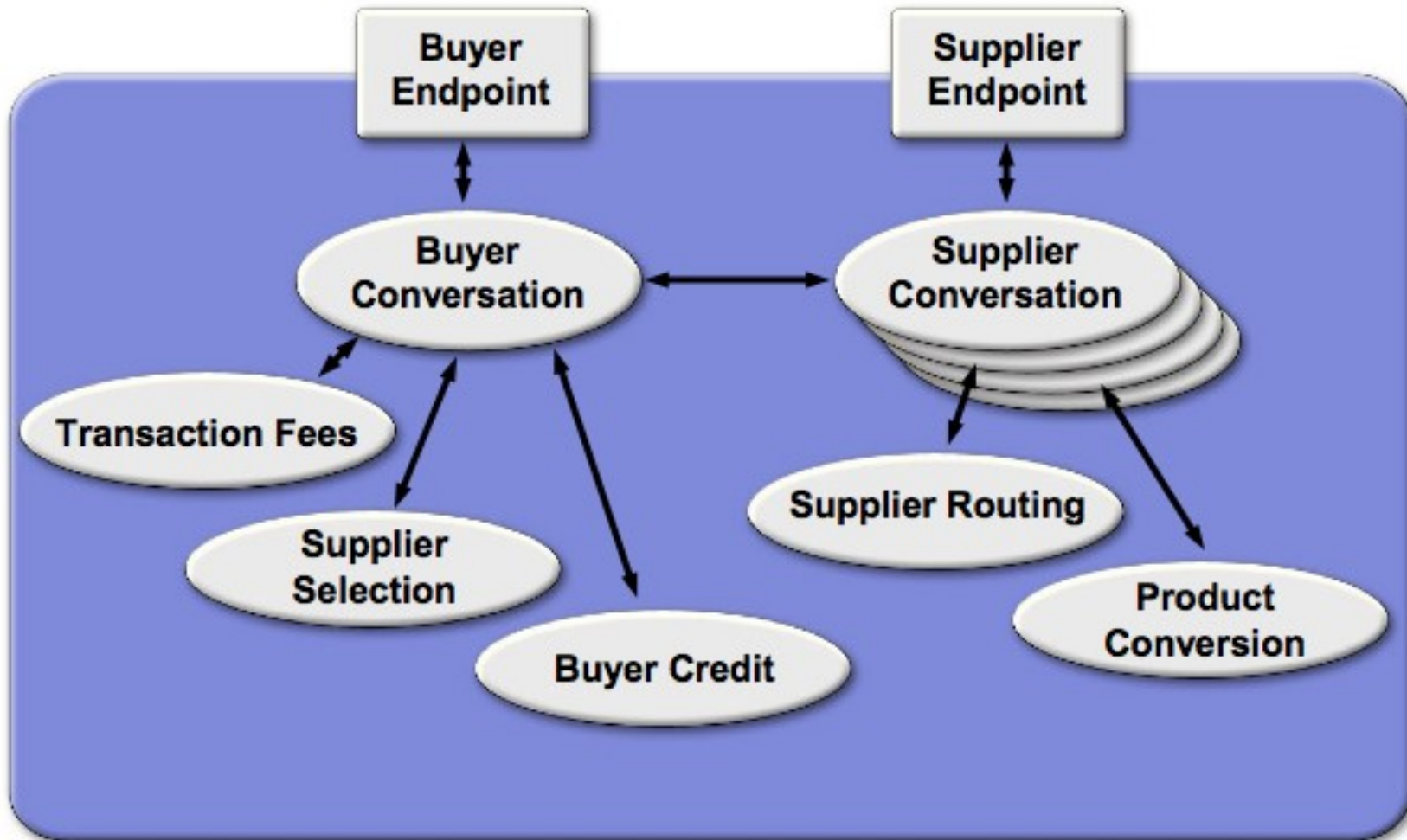


Normalized message service

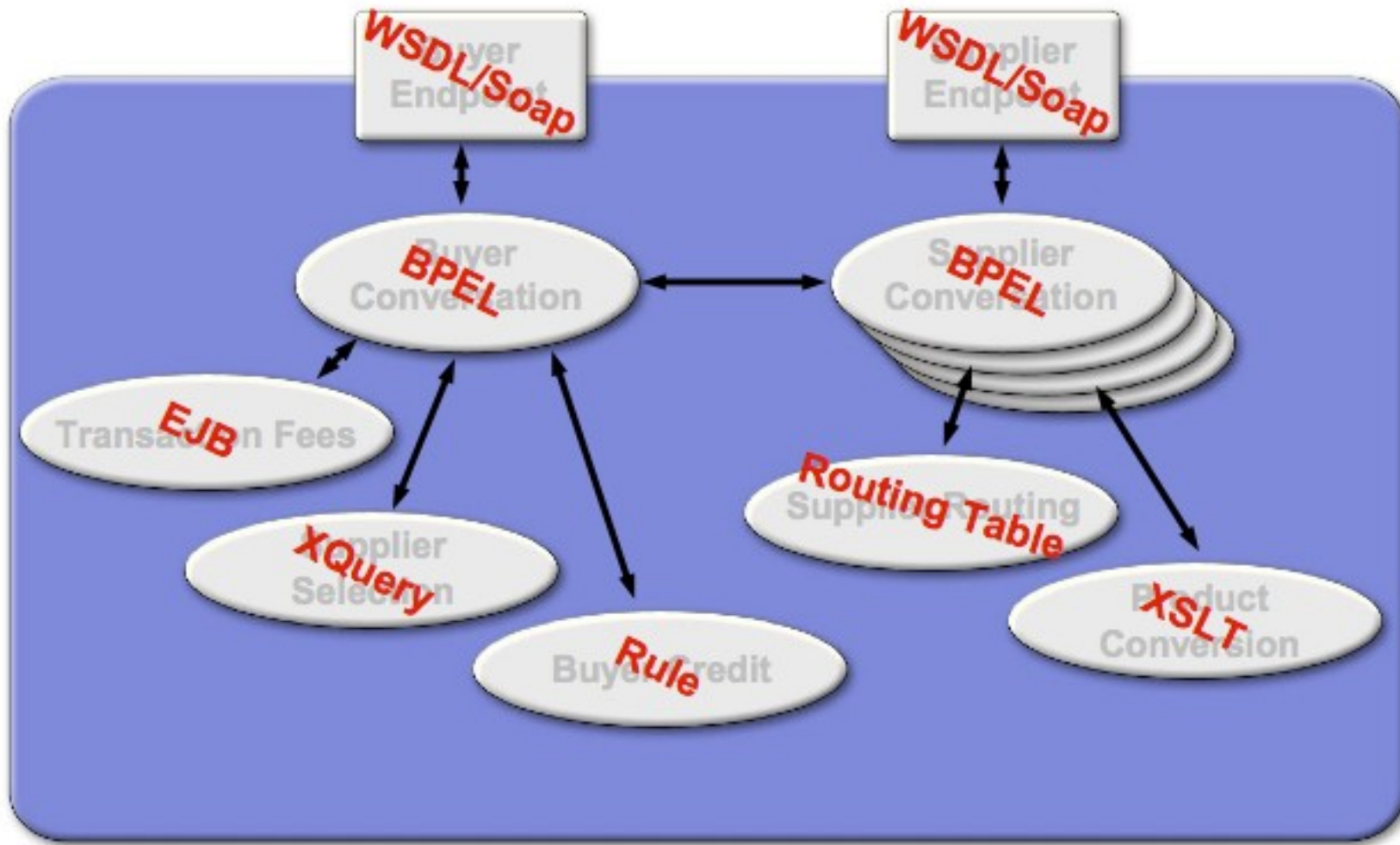
Kommunikáció-minták: (WSDL 2.0)

- In-Only (egyirányú WSDL 1.1)
 - > üzenet befelé, nincs hibajelzés (fault)
- Robust In-Only
 - > üzenet befelé, válaszban hibajelzés
- In-Out (kérdés-felelet WSDL 1.1)
 - > üzenet be, válasz-vagy-hiba kifelé
- In Optional-Out
 - > üzenet be, lehet hiba,
 - > válasz, arra is lehet hibajelzés

Composite application



Composite application



Service engine, binding component

- Service engine:
 - > XSLT
 - > rule engine
 - > EJB
 - > BPEL
 - > ...
- Binding component:
 - > HTTP
 - > JMS
 - > file
 - > ...

Szerepek

- Engine developer
- Binding developer
- JBI environment provider
- JBI application developer

JBI eszközök

Kell:

- JBI implementáció
- néhány SE
- néhány BC
- jó eszközök (IDE)
- JavaEE container

OpenESB 2.0 (2007. 03. 13.)

`open-esb.dev.java.net`

Open source JBI eszközök

- JBI implementáció: **OpenESB**
 - > néhány SE
 - > néhány BC
- jó eszközök: **NetBeans** enterprise pack
- JavaEE container: **Glassfish**

`open-esb.dev.java.net`
`glassfish.dev.java.net`
`netbeans.org`

open-esb.dev.java.net 2.0

Service Engines:

- > WS-BPEL 2.0 (2007. feb.)
- > JavaEE 5 (kétirányú)
- > XSLT
- > JDBC

Binding Components:

- > HTTP (SOAP és REST)
- > JMS, MQSeries
- > file
- > SMTP
- > FTP

NetBeans.org – enterprise pack

Editorok:

- BPEL (BPMN)
- WSDL
- XSD

Deploy:

- csomagolás (ant)
- service assembly deploy
- Teszt:
- BPEL debugger

Összefoglalás

SOA architecture Webservice folyamat java
WS-addressing java **JavaEE** business process
execution **szinkron** aszinkron lazán csatolt
szolgáltatás **JB** integráció távoli metódushívás
dokumentum **XML** durva szemcsés orchestration
BPEL registry location transparent
relációs adatbázis MOM fejlesztőeszköz
szabványok NetBeans open **ESB**

Kérdések/Demo/Címek

- `geza.simon@sun.hu`
`tamas.zsemlye@sun.com`
- `open-esb.dev.java.net`
- `glassfish.dev.java.net`
- `netbeans.org/products/enterprise`

`www.javagrund.hu`