

The background is a solid dark blue color. On the left side, there is a vertical strip of colorful, pixelated gears in shades of orange, yellow, and white. Overlaid on the blue background are several semi-transparent gears in various shades of blue and grey, arranged in a cluster. The text is centered in the middle of the page.

Υποστήριξη Ροής Εργασιών

Υποστήριξη Ροής Εργασιών

- Ροή εργασιών (workflow)
 - αυτοματοποίηση του συνόλου ή μέρους μιας επιχειρηματικής διαδικασίας
 - έγγραφα, πληροφορίες ή και αναθέσεις εργασιών περνούν από έναν συμμετέχοντα στον άλλο
 - σύμφωνα με ένα σύνολο προαποφασισμένων κανόνων
- Επιχειρηματική διαδικασία (business process)
 - σύνολο δια-συνδεδεμένων δραστηριοτήτων που συλλογικά πραγματοποιούν έναν επιχειρηματικό στόχο
 - καθορίζουν λειτουργικούς ρόλους και σχέσεις
- Μία δραστηριότητα (activity)
 - είναι ένα από τα βήματα σε μία διαδικασία

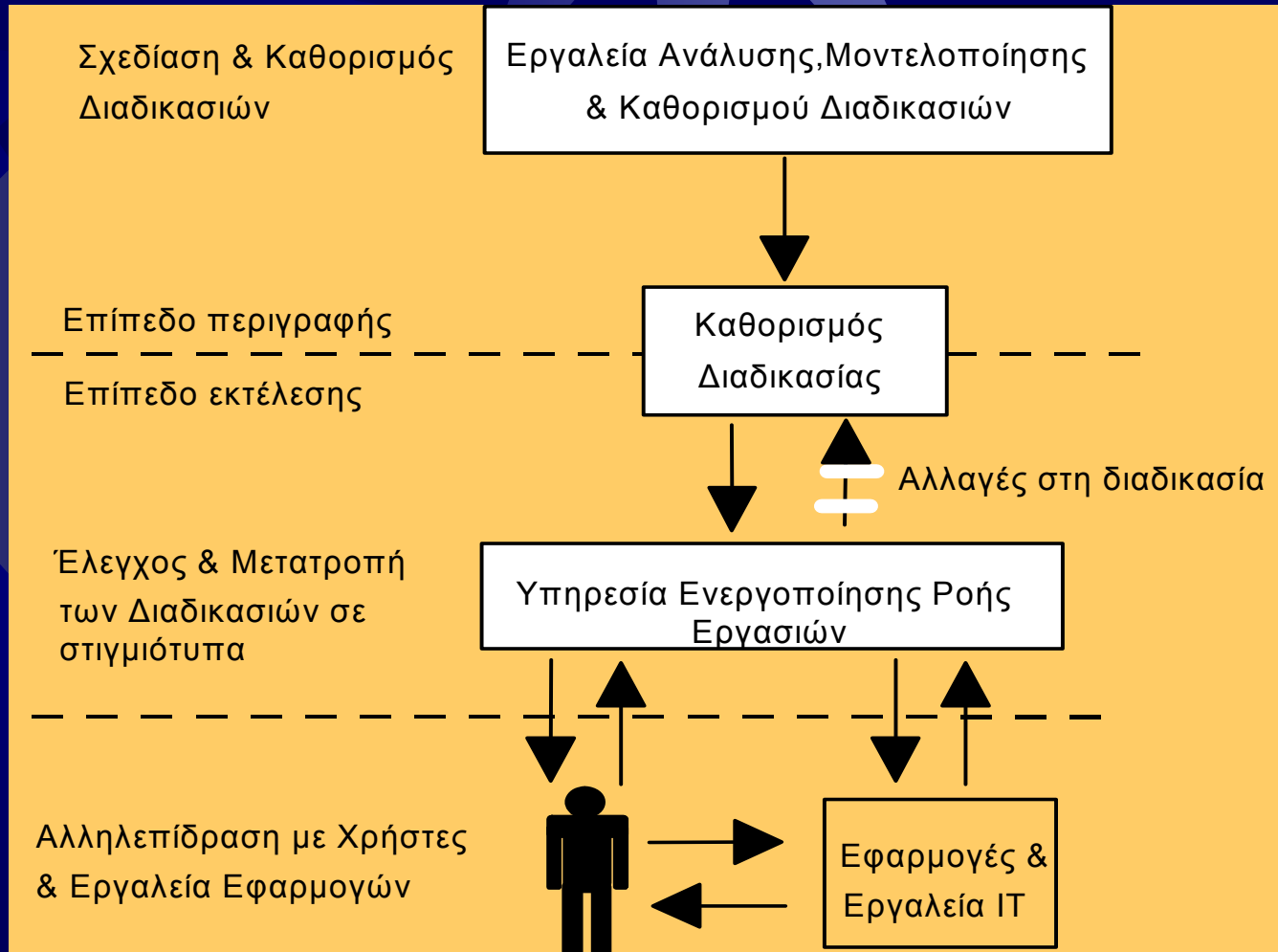
Συστήματα Διαχείρισης Ροής Εργασιών (ΣΔΡΕ)

- Πληροφοριακά συστήματα τα οποία
 - καθορίζουν
 - μοντελοποιούν
 - διαχειρίζονται και
 - εκτελούντμήματα διαδικασιών ενός οργανισμού
- Επιχειρηματικές διαδικασίες μπορούν να καταταμηθούν σε δραστηριότητες
 - αυτόματη αποστολή μηνυμάτων
 - αυτόματη δρομολόγηση εγγράφων
- Όταν ολοκληρωθεί η εκάστοτε δραστηριότητα, το σύστημα εξασφαλίζει ότι οι υπεύθυνοι για την επόμενη δραστηριότητα ενημερώνονται και λαμβάνουν τα απαραίτητα δεδομένα για την εκτέλεση αυτού του σταδίου της διαδικασίας
- Εργασίες μοντελοποιούνται
- ΣΔΡΕ αναλαμβάνει την εκτέλεση του μοντέλου
- (video)

Επίπεδα ΣΔΡΕ

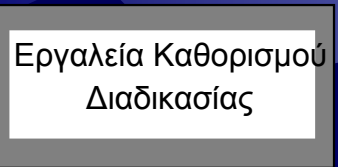
- Επίπεδο περιγραφής και μοντελοποίησης των διαδικασιών
 - περιγράφει τα στοιχεία που καθορίζουν τη λειτουργία του ΣΔΡΕ:
 - ποιες διαδικασίες θα εκπληρώσει ή θα παρακολουθήσει
 - τι είδους δεδομένα θα αποθηκεύσει
 - με ποιες διεπιφάνειες θα επικοινωνεί με τους χρήστες στο επίπεδο ενεργοποίησης Κ.Ο.Κ.
- Επίπεδο εκτέλεσης των διαδικασιών
 - εκτέλεση των δραστηριοτήτων που έχουν καθοριστεί
 - παρακολουθούνται οι δραστηριότητες που εκτελούνται και τα πρόσωπα που τις εκτελούν
 - παρέχεται υποστήριξη για την εκτέλεση αυτών των δραστηριοτήτων Κ.Ο.Κ.

Επίπεδα ΣΔΡΕ

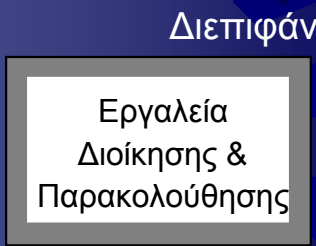


Μοντέλο Αναφοράς Ροής Εργασιών (WfMC)

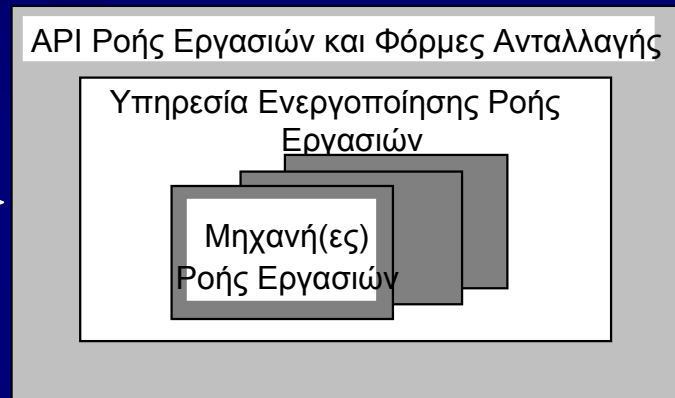
Διεπιφάνεια 5 κοινή για διαφορετικά ΣΔΡΕ, επιτρέπει σε πολλές Υπηρεσίες Ενεργοποίησης να έχουν πρόσβαση σε Εργαλεία Διοίκησης και Παρακολούθησης, τα οποία παρέχουν μία ολοκληρωμένη εικόνα της ροής εργασιών σε έναν οργανισμό, ανεξάρτητα από το σύστημα στο οποίο βρίσκεται



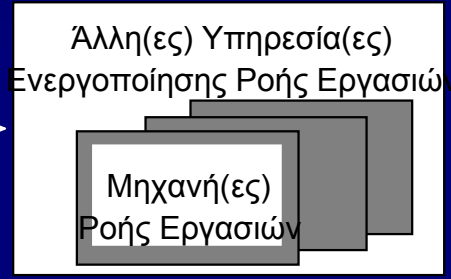
Διεπιφάνεια 1 επιτρέπει την ανταλλαγή καθορισμένων διαδικασιών μεταξύ των Εργαλείων Καθορισμού Διαδικασιών και της Υπηρεσίας Ενεργοποίησης



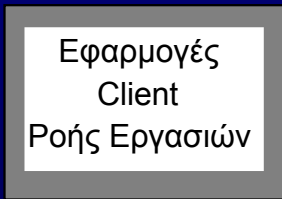
Διεπιφάνεια 5



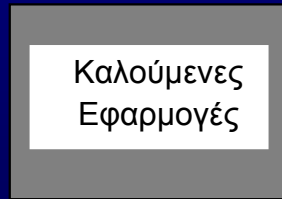
Διεπιφάνεια 4



Διεπιφάνεια 2



Διεπιφάνεια 3



Διεπιφάνεια 2 περιλαμβάνει την έννοια της λίστας εργασίας, χρησιμοποιείται για την παρουσίαση στο χρήστη εργασιών μέσω των εφαρμογών Client ροής εργασιών και της διεπιφάνειας χρήστη

Διεπιφάνεια 3 κάνει εφικτή την αλληλεπίδραση με εξωτερικές εφαρμογές του ΣΔΡΕ

Διεπιφάνεια 4 χρησιμοποιείται για την αλληλεπίδραση με άλλες Υπηρεσίες Ενεργοποίησης Ροής Εργασιών, δηλαδή με άλλα ΣΔΡΕ

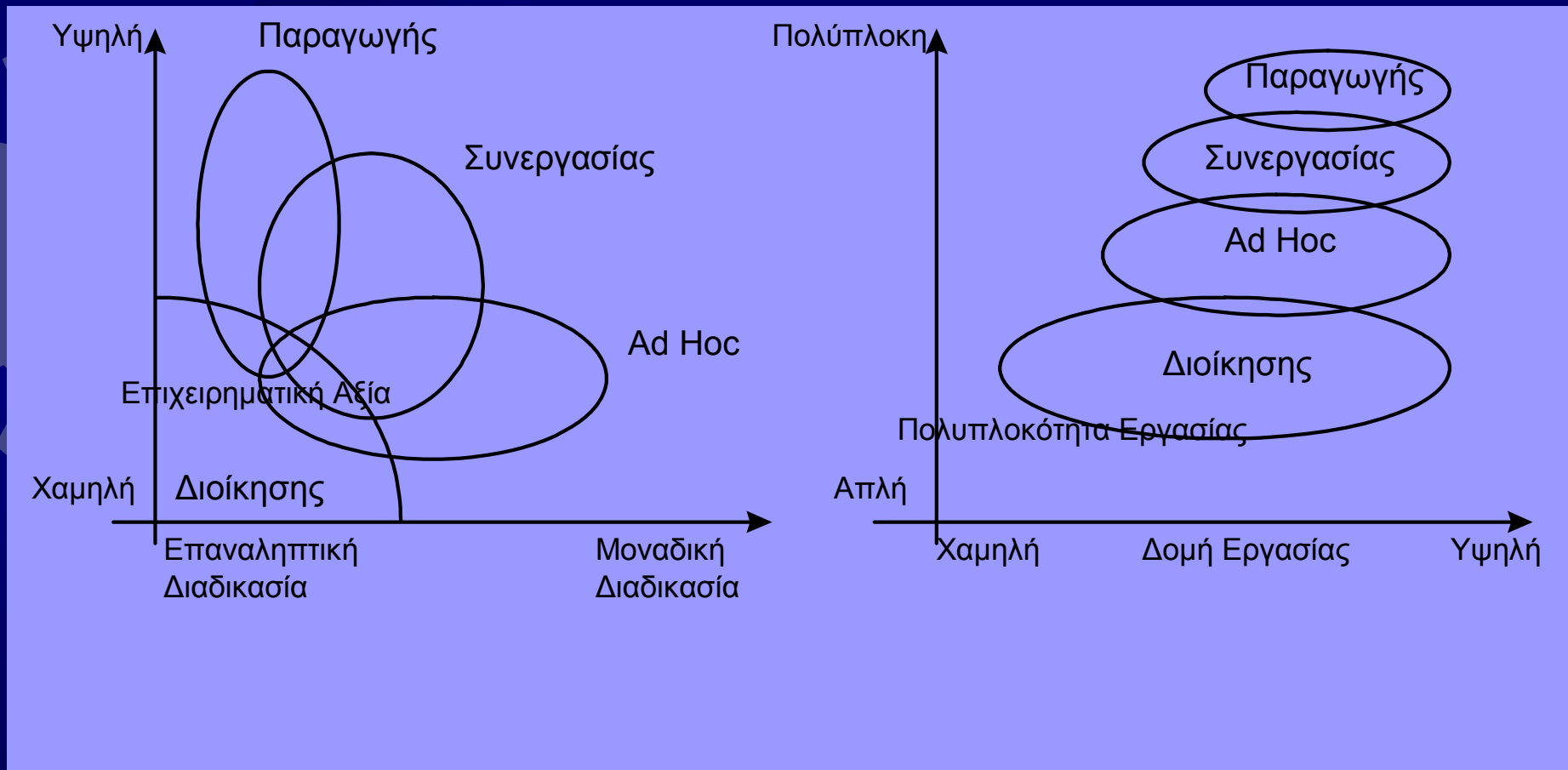
Οφέλη των ΣΔΡΕ

- Αυξάνεται η αποδοτικότητα και η τυποποίηση των διαδικασιών
 - Δίνεται έτσι η δυνατότητα για περισσότερη εργασία,
 - μειώνονται τα νεκρά διαστήματα στη διάρκεια της διαδικασίας
 - βελτιώνεται η ποιότητα μέσω της τυποποίησης
- Βελτιώνει τη διαχείριση του ελέγχου
 - παρακολούθηση της τρέχουσας κατάστασης μιας ροής εργασιών και του τρόπου με τον οποίο αυτή εξελίσσεται σύμφωνα με τον προγραμματισμό
 - εξασφαλίζοντας την ολοκλήρωση των εργασιών και την ελαχιστοποίηση των προβλημάτων
- Εργασίες ανατίθενται αποτελεσματικά στα στελέχη
 - μέσω της εξισορρόπησης του φόρτου εργασίας και της αντιστοίχισης εργασιών και στελεχών
- Γίνονται προσβάσιμες από τους υπαλλήλους οι πηγές πληροφοριών όπως και όταν απαιτούνται
- Ενθαρρύνεται η σκέψη με κέντρο τη διαδικασία μέσω του σχεδιασμού των διαδικασιών
- Η ευελιξία που αποδίδει η χρήση τους σε έναν οργανισμό οδηγεί στην δυνατότητα εύκολου και γρήγορου ανασχεδιασμού διαδικασιών (Process reengineering)

Περιορισμοί και Προβλήματα ΣΔΡΕ

- Οι τεχνικές μοντελοποίησης ροής εργασιών που ήδη υπάρχουν καλύπτουν μόνο μερικώς τις επιχειρηματικές διαδικασίες
 - πολυπλοκότητα διαδικασιών απαιτεί τεχνικές μοντελοποίησης με πιο ευρύ πεδίο δράσης.
 - αποτελεί αντικείμενο έρευνας στις μέρες μας
- Ένα βασικό πρόβλημα σε κάθε μεγάλο οργανισμό είναι η ετερογένεια των ΣΔΡΕ που χρησιμοποιούνται από τα διαφορετικά τμήματα
 - Για να είναι αποτελεσματικά, πρέπει να είναι συνδεδεμένα σε ένα κοινό σύστημα, κάτι που είναι δύσκολο να γίνει με τα υπάρχοντα ΣΔΡΕ
- Η έλλειψη κλιμάκωσης αποτελεί πρόβλημα στα υπάρχοντα συστήματα
 - Έχοντας θεωρηθεί αρχικά σαν εργαλεία συνεργασίας, τα περισσότερα ΣΔΡΕ που υπάρχουν έχουν σχεδιαστεί για μικρές ομάδες
 - Οι κατασκευαστικοί περιορισμοί (απλή βάση δεδομένων, φτωχή υποστήριξη επικοινωνίας, έλλειψη προνοητικότητας στο σχεδιασμό, τα προβλήματα που δημιουργήθηκαν από ετερογενείς σχεδιασμούς) εμποδίζουν τα υπάρχοντα συστήματα να αντιμετωπίσουν ένα μέρος του αναμενόμενου φορτίου
- Τέλος, ένα από τα σημαντικότερα προβλήματα στη χρήση των συστημάτων αυτών είναι ότι ο έλεγχος, η παρακολούθηση και ο συντονισμός των διαφόρων διαδικασιών είναι κεντρικός
 - η χρήση τους σε κατανεμημένα συστήματα όπου οι διαδικασίες συνήθως ξεπερνούν τα στενά όρια ενός οργανισμού να καθίσταται αρκετά προβληματική

Ταξινόμηση ΣΔΡΕ



ΣΔΡΕ διοίκησης

- Ασχολούνται με διοικητικές διαδικασίες όπου τα βήματα που ακολουθούνται είναι καλά καθορισμένα και βασίζονται σε κανόνες που είναι γνωστοί σε όλους τους χρήστες
- Παραδείγματα
 - εγγραφή σε μαθήματα σε ένα πανεπιστήμιο
 - αίτηση για βαθμό μετά το τέλος της διατριβής
 - αίτησης έκδοση άδειας οχήματος
- και σχεδόν οποιαδήποτε άλλη διαδικασία στην οποία χρειάζεται να συμπληρωθεί ένα σύνολο αιτήσεων και να προωθηθούν μέσα από μια σειρά βημάτων
- Θα πρέπει να σημειωθεί ότι αυτός ο τύπος των ΣΔΡΕ υποστηρίζει την ιδέα της επεξεργασίας εντύπων, έναν νέο όρο για την παλαιότερη έννοια του γραφείου “χωρίς χαρτιά” και επίσης σχετίζεται με συστήματα μεγάλης κλίμακας, όπου ο αριθμός των διαδικασιών που περιλαμβάνονται τείνει να είναι πολύ υψηλός

Ad-hoc (περιστασιακά) ΣΔΡΕ

- Αντιμετωπίζουν εξαιρέσεις και μοναδικές καταστάσεις
- Ο λόγος που χρησιμοποιείται ένα ΣΔΡΕ με τέτοια χαρακτηριστικά, δεν είναι η δυσκολία παρακολούθησης κάθε ξεχωριστής διαδικασίας, αλλά η δυσκολία να παρακολουθηθούν όλες οι διαδικασίες ταυτόχρονα

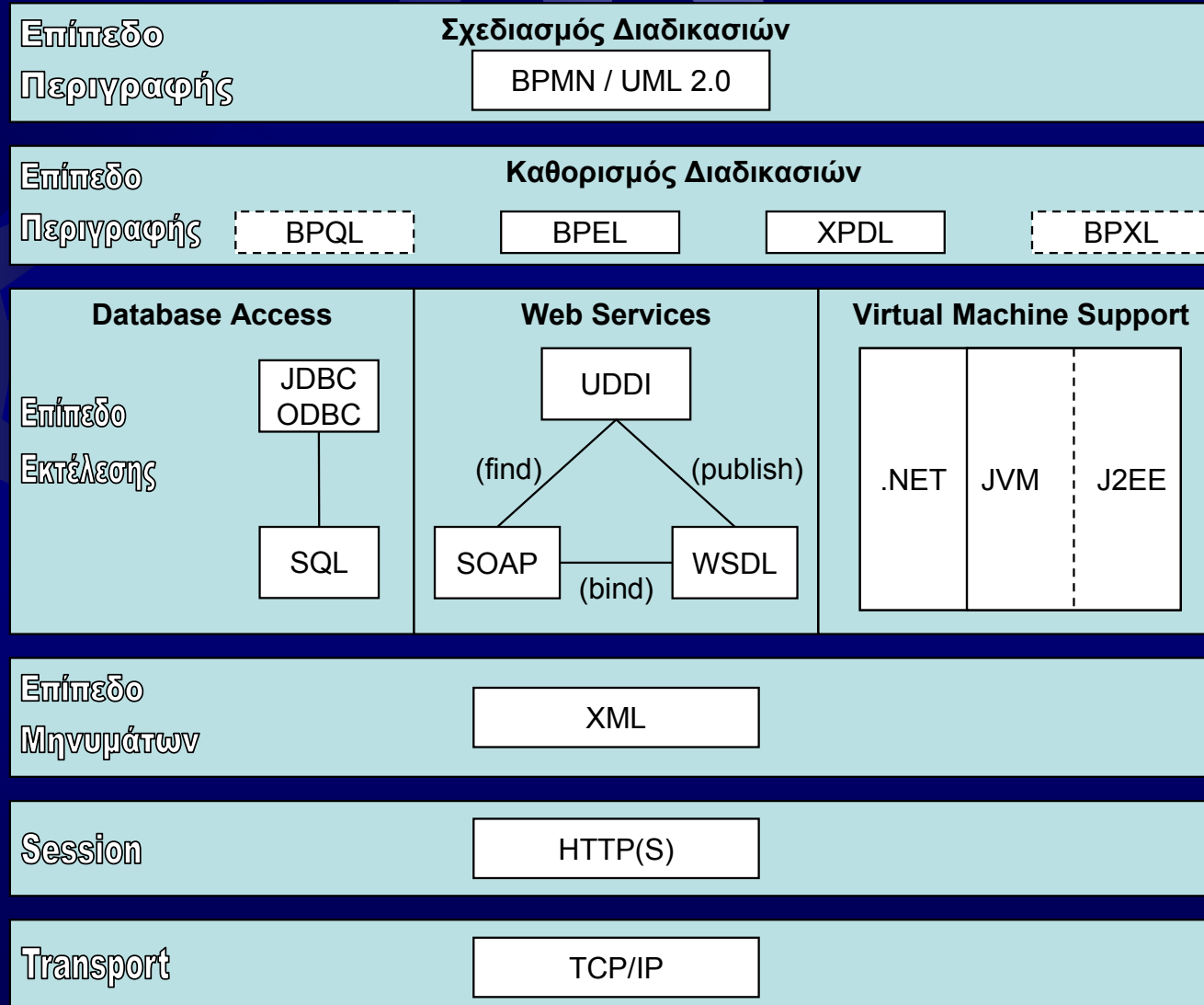
ΣΔΡΕ συνεργασίας

- Χαρακτηρίζονται από τον αριθμό των συμμετόχων και τις αλληλεπιδράσεις μεταξύ τους
- Σε αντίθεση με άλλους τύπους ΣΔΡΕ που βασίζονται στο γεγονός ότι υπάρχει συνεχώς πορεία προς τα εμπρός, ένα ΣΔΡΕ συνεργασίας μπορεί να περιλαμβάνει αρκετές επαναλήψεις του ίδιου βήματος μέχρις ότου κάποια μορφή συμφωνίας επιτευχθεί ή ίσως να περιλαμβάνει επιστροφή σε ένα νωρίτερο στάδιο
 - Π.χ. γράψιμο μιας εργασίας από αρκετά άτομα
 - Δύσκολο να μοντελοποιηθεί μια τέτοια διαδικασία χρησιμοποιώντας εργαλεία που δεν είναι προσαρμοσμένα για συνεργασία μιας και είναι σχεδόν αδύνατο να καθοριστούν τα βήματα που θα ακολουθηθούν
- Είναι δυναμικά, με την έννοια ότι επανα-ορίζονται καθώς εξελίσσονται

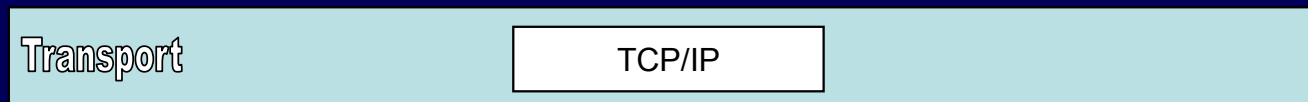
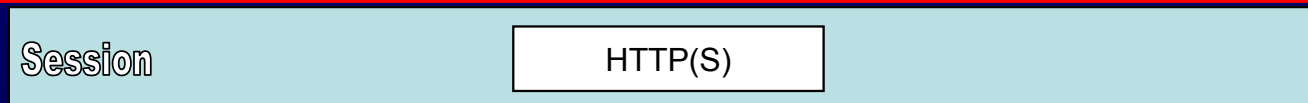
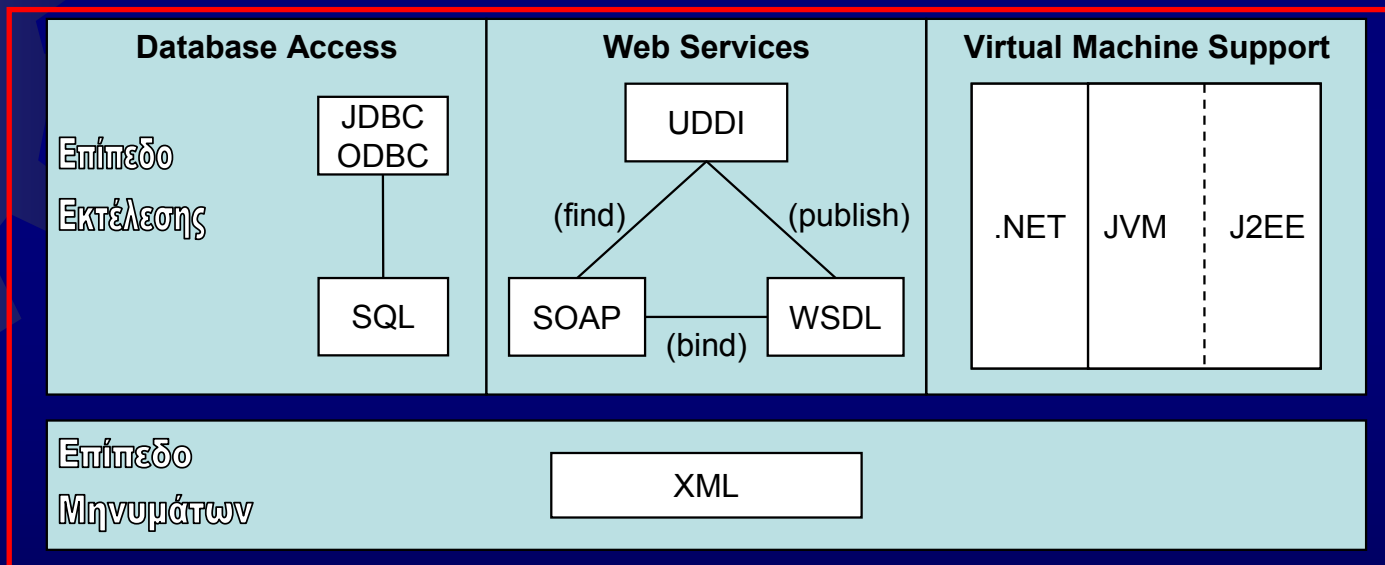
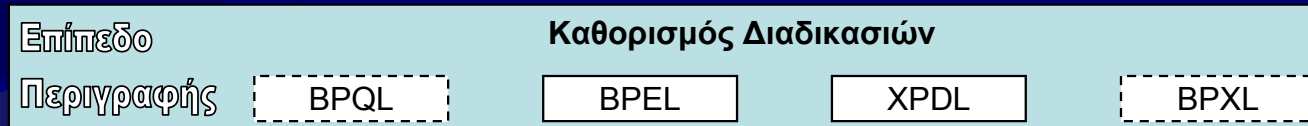
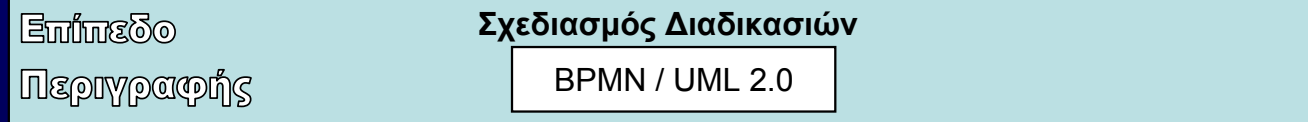
ΣΔΡΕ παραγωγής ή συναλλαγής

- Πιο πολύπλοκο είδος ΣΔΡΕ
- Αυτοματοποιούν περίπλοκες επιχειρηματικές διαδικασίες που είναι επαναληπτικές, καλά δομημένες και μεγάλου όγκου
- Παραδείγματα
 - Αιτήσεις δανείων και πίστωσης
 - Ασφαλιστικές απαιτήσεις
- Διαφορά ανάμεσα σε ΣΔΡΕ διοίκησης και παραγωγής είναι μερικές φορές θέμα προοπτικής
 - ΣΔΡΕ παραγωγής, κύρια σημεία
 - η μεγάλη κλίμακα,
 - η πολυπλοκότητα και
 - η ετερογένεια του περιβάλλοντος όπου αυτά εκτελούνται,
 - η ποικιλία των ανθρώπων και των οργανώσεων που εμπλέκονται
 - η φύση των εργασιών

Τυποποιημένες Web τεχνολογίες υποστήριξης ΣΔΡΕ



Τυποποιημένες Web τεχνολογίες υποστήριξης ΣΔΡΕ



Επίπεδο εκτέλεσης

- Υπάρχουν διάφορες τεχνολογίες για προγραμματισμό και διάθεση αυτοματοποιημένης λειτουργικότητας
 - Έμφαση σε πρόσβαση σε βάσεις δεδομένων
 - JDBC, ODBC, SQL
 - Πρότυπα συγκεκριμένων προμηθευτών
 - Π.χ. .NET
 - Web Services
 - Δημιουργός της τεχνολογία IBM αλλά θεωρείται de-facto standard

Τι είναι Web Services

- “Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. ... Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.”

IBM web service tutorial

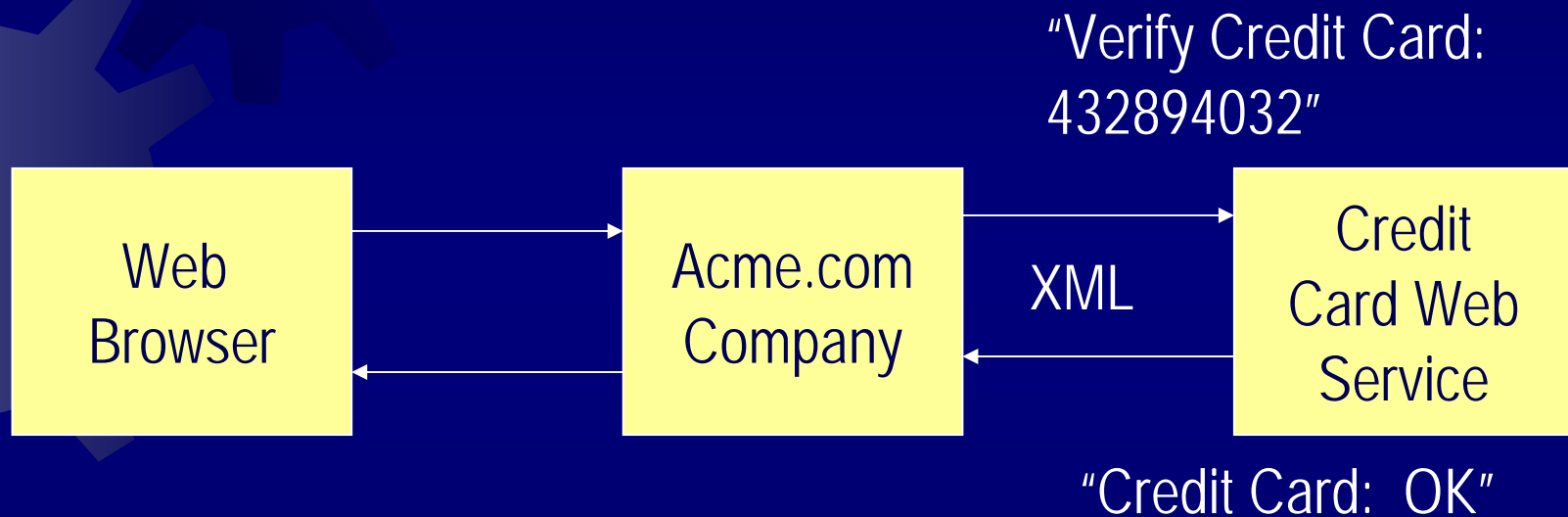
- Web Service είναι:
 - κάθε service διαθέσιμη μέσω του web
 - κάθε service που επιτρέπει σε δύο εφαρμογές λογισμικού να ανταλλάξουν δεδομένα
 - κυρίως, αλλά όχι απόλυτα, βασίζεται σε:
 - XML για data encoding
 - HTTP για data transport

Ρόλος Web Services

- Τα Web Services επιτρέπουν την πρόσβαση σε εφαρμογές, όμως δεν αποτελούν τα ίδια αυτές τις εφαρμογές ούτε τις κατασκευάζουν
- Είναι απαραίτητο να χρησιμοποιηθεί μια γλώσσα προγραμματισμού για το σχεδιασμό των client και server μερών της επικοινωνίας
 - Java, C/C++, ASP, Perl, etc.
- Τα Web services αποτελούν απλώς την «κόλλα» μεταξύ των εφαρμογών client και server

Ένα απλό παράδειγμα

- Επιβεβαίωση πιστωτικών καρτών (credit card verification)



- Αυτή η λειτουργικότητα υπάρχει ήδη
- Ωστόσο, στόχος των Web services είναι η τυποποίηση της διαδικασίας, υποστηρίζοντας τη λειτουργικότητα “plug and play”

Χαρακτηριστικά Web Services

■ Modular

- τα συστατικά στοιχεία (υπο-υπηρεσίες) μπορούν να επαναχρησιμοποιηθούν και είναι δυνατή η σύνθεσή τους για τη δημιουργία μεγαλύτερων συστατικών στοιχείων

■ Available

- τα services πρέπει να εκτίθενται έξω από την συγκεκριμένη τεχνολογία ή το σύστημα που είναι διαθέσιμες
- τα services μπορούν να αποκεντρωθούν εντελώς και να κατανεμηθούν στο Internet

■ Described

- η περιγραφή των services είναι αναγνώσιμη από μηχανές
- η περιγραφή μπορεί να χρησιμοποιηθεί για τον προσδιορισμό της διεπαφής («παράθυρο στο κόσμο) του service

■ Implementation-independent

- η διεπαφή (interface) του service είναι ανεξάρτητη από την τελική υλοποίηση

■ Published

- οι περιγραφές των services είναι διαθέσιμες σε ένα repository όπου οι χρήστες (άνθρωποι και υπολογιστές) μπορούν να βρουν το service και να χρησιμοποιήσουν την περιγραφή για να έχουν πρόσβαση σε αυτό

The Web Services Stack

UDDI: Protocol for discovering services

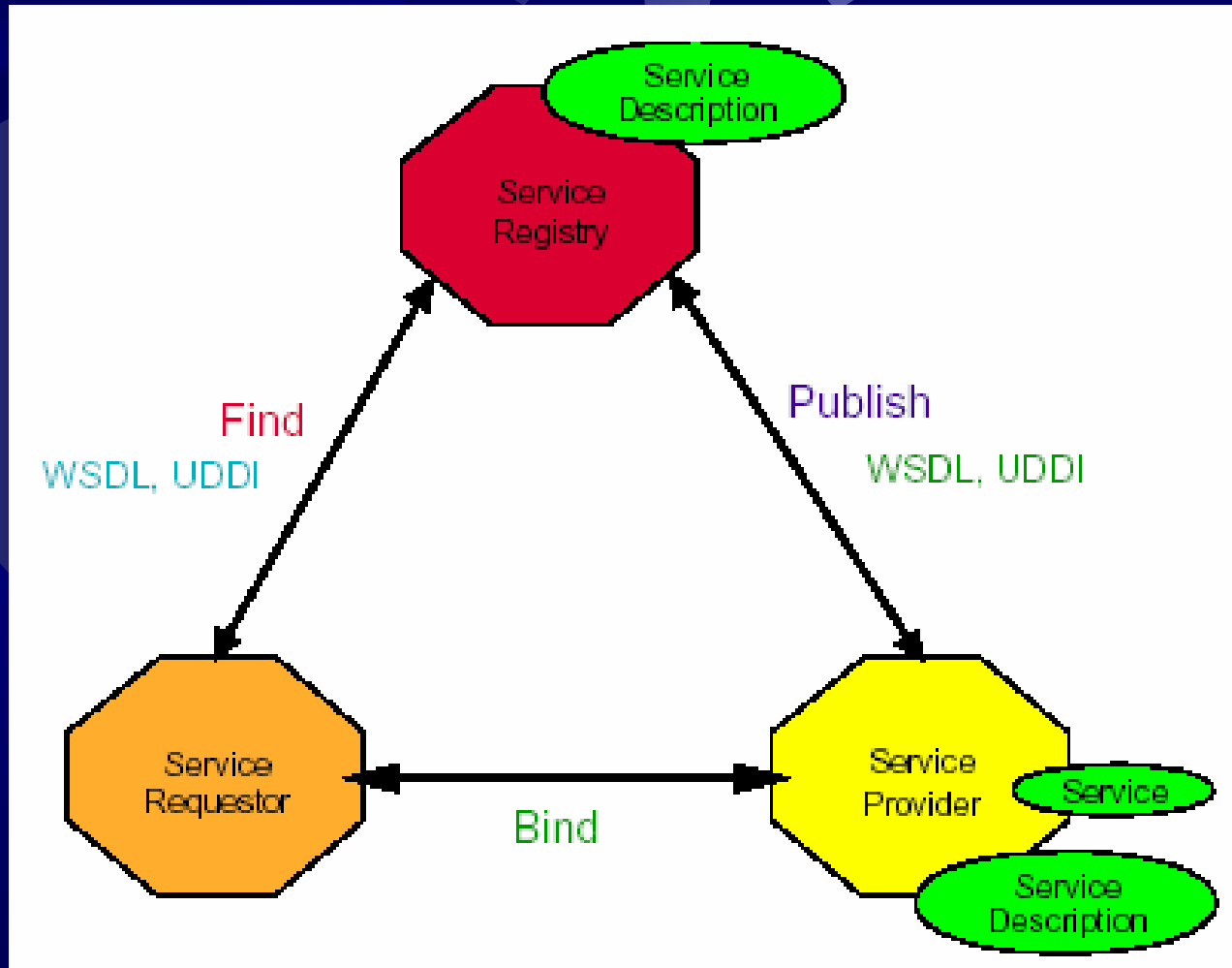
WSDL: Format for describing services

SOAP: Protocol for Remote Procedure Calls

XML: Data Encoding

HTTP: Data Transport

Web Services Model



SOAP

UDDI: Protocol for discovering services

WSDL: Format for describing services

SOAP: Protocol for Remote Procedure Calls

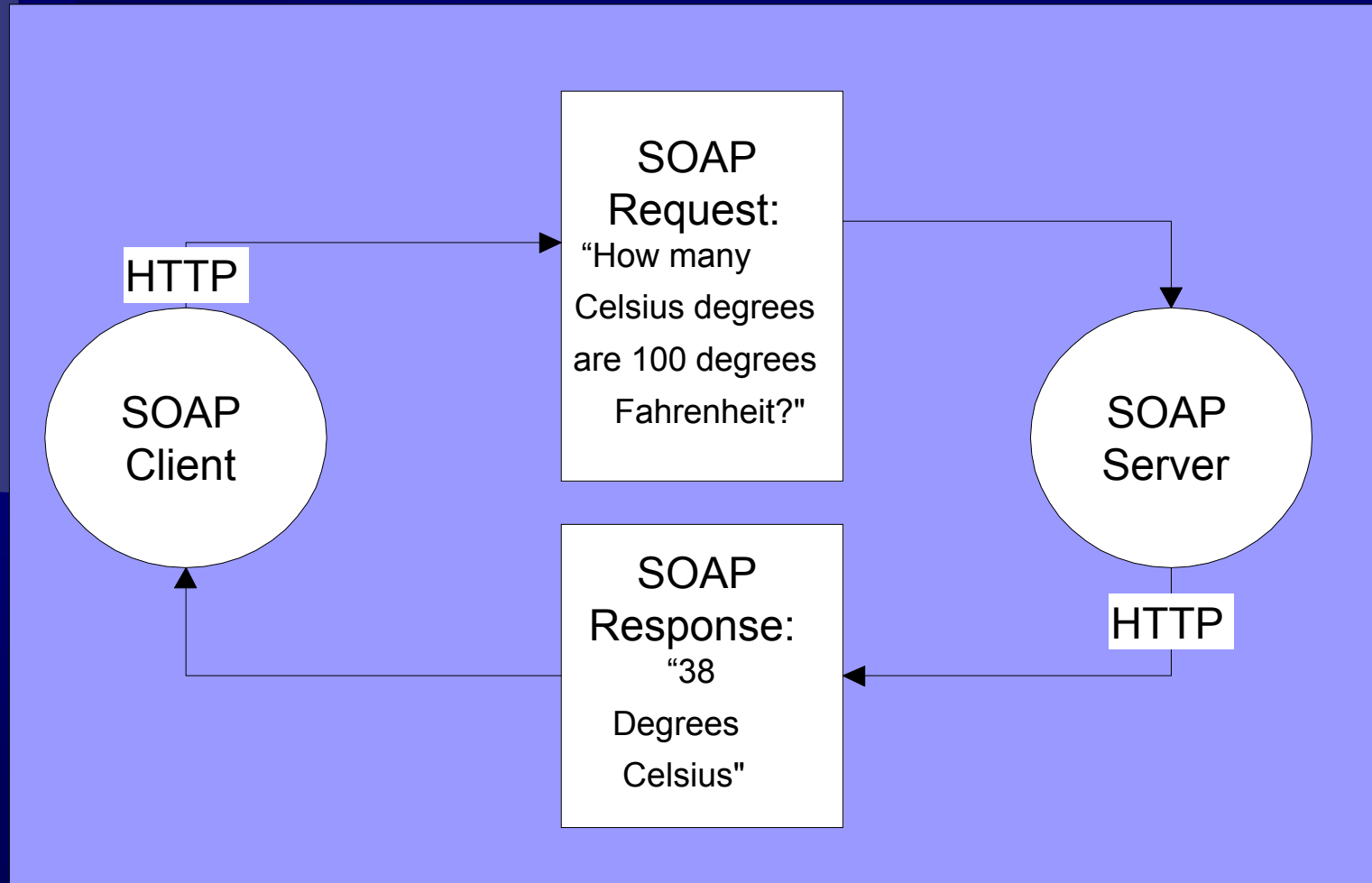
XML: Data Encoding

HTTP: Data Transport

- Το Simple Object Access Protocol (SOAP) είναι ένα πρωτόκολλο ανταλλαγής πληροφοριών μεταξύ Η/Υ που βασίζεται στην XML
- Ο βασικός στόχος του SOAP είναι τα Remote Procedure Calls (RPC) που μεταφέρονται μέσω του HTTP
- $SOAP = RPC + XML + HTTP$

Παράδειγμα SOAP

- Έλεγχος καιρικών συνθηκών



SOAP Messaging

- Ένα SOAP Conversation περιλαμβάνει:
 - SOAP Request: προσδιορισμός method name, method parameters, κλπ.
 - SOAP Response: προσδιορισμός return values ή error conditions.
- Όλα τα SOAP messages κωδικοποιούνται σε XML.

XML - eXtensible Markup Language

- markup language σχετικά όμοια με την HTML
- σχεδιάστηκε να περιγράφει δεδομένα
- αποτελεί ένα εργαλείο μεταφοράς / μετάδοσης δεδομένων ανεξαρτήτως πλατφόρμας και λογισμικού
- Υποστηρίζει / διευκολύνει την ανταλλαγή δεδομένων μεταξύ μη συμβατών συστημάτων

Παράδειγμα σε XML

```
<book>  
  <title>Nonmonotonic Reasoning: Context- Dependent  
Reasoning</title>  
  <author>V. Marek</author>  
  <author>M. Truszczynski</author>  
  <publisher>Springer</publisher>  
  <year>1993</year>  
  <ISBN>0387976892</ISBN>  
</book>
```

XML - eXtensible Markup Language

- τα XML tags (ετικέτες XML) ΔΕΝ είναι προκαθορισμένα
 - ο χρήστης θα πρέπει να ορίσει τα δικά του XML tags
- ο χρήστης θα πρέπει να ορίσει το δικό του XML Schema για να περιγράψει
 - τη δομή των δεδομένων και
 - τη δομή των XML εγγράφων που δημιουργεί
- Για να αποφεύγονται conflicts μεταξύ ορισμών διαφορετικών developers χρησιμοποιούνται prefixes:

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

```
<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

- Τα prefixes μπορούν να έχουν μοναδικά ονόματα:

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
```

...

Skeleton SOAP Message

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
  ...
  ...
</soap:Header>

<soap:Body>
  ...
  ...
  <soap:Fault>
    ...
    ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

The required SOAP Envelope element is the root element of a SOAP message. It defines the XML document as a SOAP message.

The optional SOAP Header element contains application specific information (like authentication, payment, etc) about the SOAP message.

The required SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message.

Παράδειγμα SOAP 1.1

FahrenheitToCelsius Web Service

(http://www.w3schools.com/webservices/ws_example.asp)

SOAP request

```
POST /webservices/tempconvert.asmx HTTP/1.1
```

```
Host: www.w3schools.com
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
SOAPAction: "http://tempuri.org/FahrenheitToCelsius"
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <f:FahrenheitToCelsius xmlns:f="http://tempuri.org/">  
      <f:Fahrenheit>100</f:Fahrenheit>  
    </f:FahrenheitToCelsius>  
  </soap:Body>  
</soap:Envelope>
```

Παράδειγμα SOAP 1.1

SOAP response

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <f:FahrenheitToCelsiusResponse xmlns:f="http://tempuri.org/">  
      <f:FahrenheitToCelsiusResult>38</f:FahrenheitToCelsiusResult>  
    </f:FahrenheitToCelsiusResponse>  
  </soap:Body>  
</soap:Envelope>
```

Παράδειγμα Web Service σε ASP.NET

```
<%@ WebService Language="VBScript" Class="TempConvert" %>
Imports System
Imports System.Web.Services
Public Class TempConvert :Inherits WebService

<WebMethod()> Public Function FahrenheitToCelsius
(ByVal Fahrenheit As String) As String
    dim fahr
    fahr=trim(replace(Fahrenheit,",","."))
    if fahr="" or IsNumeric(fahr)=false then return "Error"
    return (((fahr) - 32) / 9) * 5)
end function

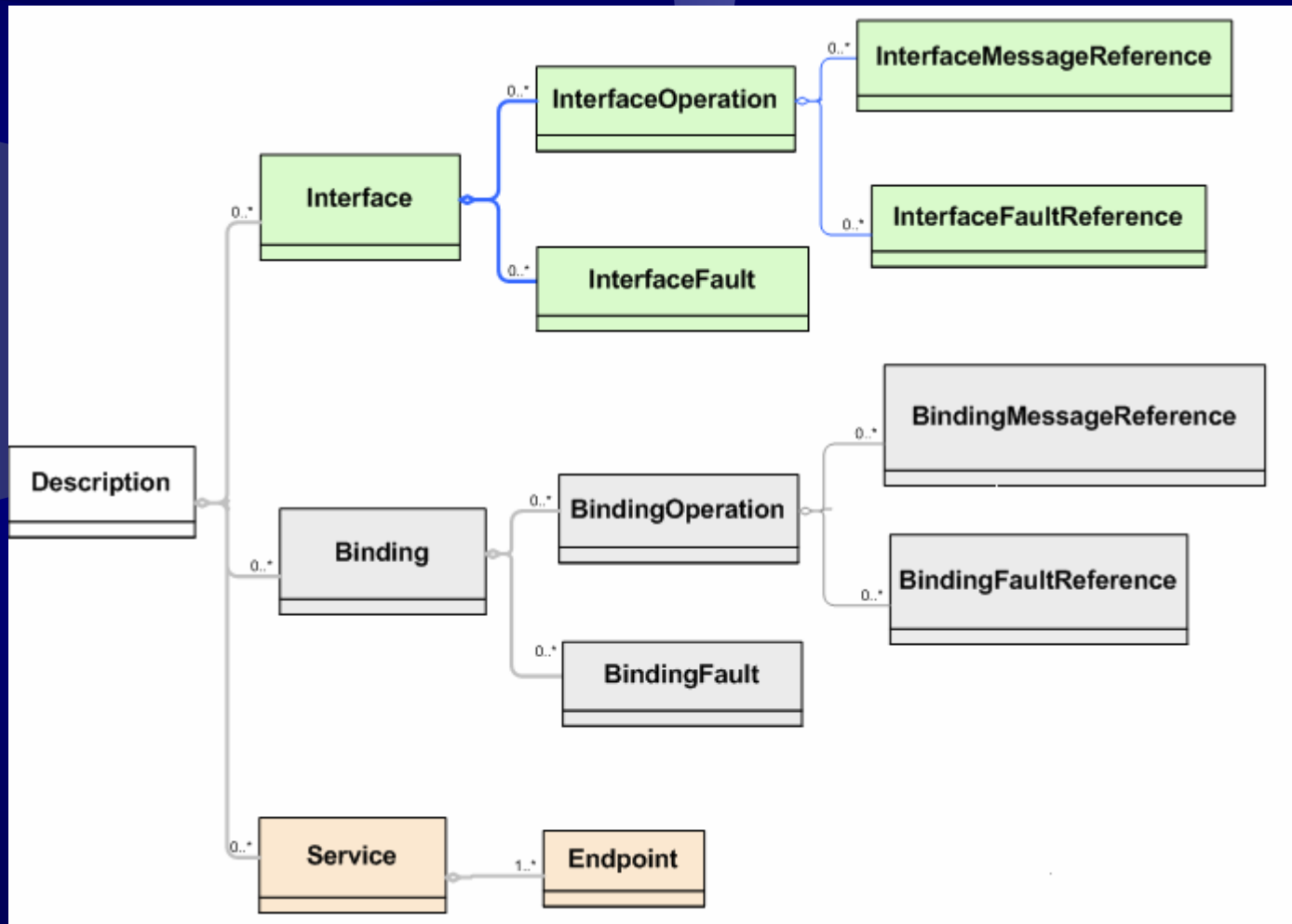
<WebMethod()> Public Function CelsiusToFahrenheit
(ByVal Celsius As String) As String
    dim cel
    cel=trim(replace(Celsius,",","."))
    if cel="" or IsNumeric(cel)=false then return "Error"
    return (((cel) * 9) / 5) + 32)
end function
end class
```


Web Services Description Language 2.0

- Ένα WSDL κείμενο χρησιμοποιεί τους παρακάτω τύπους δεδομένων για την περιγραφή ενός Web Service.

Στοιχείο	Προσδιορίζει
<types>	Προσδιορίζει τους τύπους δεδομένων που χρησιμοποιούνται από το Web Service.
<interface>	Προσδιορίζει μια σειρά μηνυμάτων που ανταλλάσσονται.
<binding>	Καθορίζει τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται.
<service>	Καθορίζει που βρίσκεται η υπηρεσία.

WSDL 2.0 Structure



WSDL 2.0 Components

```
<wsdl:description xmlns:wsdl="http://www.w3.org/ns/wsdl">  
  <wsdl:types/>  
  <wsdl:interface/>  
  <wsdl:binding/>  
  <wsdl:service/>  
</wsdl:description>
```

WSDL 2.0 Types - Interface

- Το στοιχείο <types> περιλαμβάνει όλο το σχήμα XML περιγραφής ενός element και δηλώσεις τύπου που περιγράφουν τα μηνύματα του Web Service.
 - Η WSDL 2.0 είναι ανοικτή ως πρότυπο για χρήση και άλλων σχημάτων για την περιγραφή των μηνυμάτων πέραν αυτού της XML που χρησιμοποιείται ως προεπιλογή.
- Το στοιχείο <Interface> περιγράφει μια σειρά μηνυμάτων που ανταλλάσσονται ομαδοποιώντας τα συσχετιζόμενα μηνύματα σε λειτουργίες (operations).
 - Τα operations είναι μια ακολουθία εισερχομένων και εξερχόμενων μηνυμάτων.
 - Ένα interface είναι ένα σύνολο από operations.

WSDL 2.0 Types Example

```
<types>
  <xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://greath.example.com/2004/schemas/resSvc"
    xmlns="http://greath.example.com/2004/schemas/resSvc">
    <xs:element name="checkAvailability" type="tCheckAvailability"/>
    <xs:complexType name="tCheckAvailability">
      <xs:sequence>
        <xs:element name="checkInDate" type="xs:date"/>
        <xs:element name="checkOutDate" type="xs:date"/>
        <xs:element name="roomType" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="checkAvailabilityResponse" type="xs:double"/>

    <xs:element name="invalidDataError" type="xs:string"/>
  </xs:schema>
</types>
```

WSDL 2.0 Interface Example

```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wsd1"
  targetNamespace= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:tns= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:ghns = "http://greath.example.com/2004/schemas/resSvc"
  . . .
  xmlns:wsd1x="http://www.w3.org/ns/wsd1-extensions">
  . . .
  <types>
  </types>
  <interface name = "reservationInterface" >
    <fault name = "invalidDataFault"
      element = "ghns:invalidDataError"/>
    <operation name="opCheckAvailability"
      pattern="http://www.w3.org/ns/wsd1/in-out"
      style="http://www.w3.org/ns/wsd1/style/iri"
      wsdlx:safe = "true">
      <input messageLabel="In"
        element="ghns:checkAvailability" />
      <output messageLabel="Out"
        element="ghns:checkAvailabilityResponse" />
      <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
    </operation>
  </interface>
  . . .
</description>
```

WSDL 2.0 Components

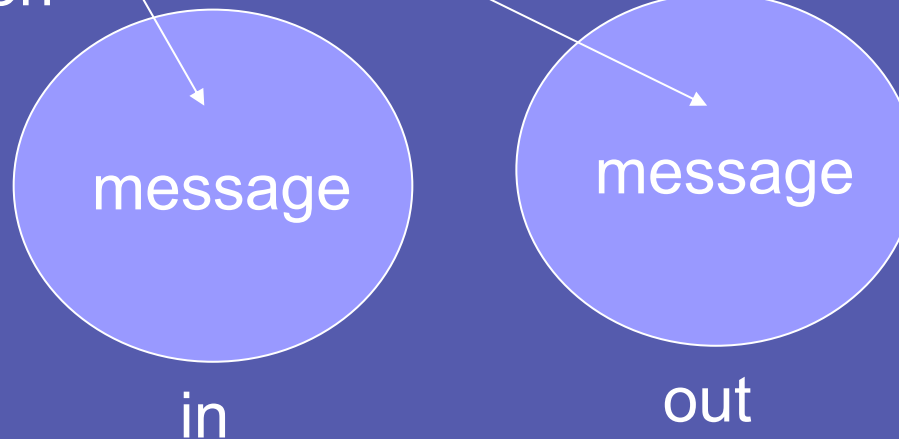
Data Types describing the message

```
<types>  
  definition of types.....  
</types>  
<message>  
  definition of a message....  
</message>
```

For maximum platform neutrality, WSDL uses XML Schema syntax to define data types.

The <message> element defines the data elements of an operation.

Operation



Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.

WSDL 2.0 Binding - Service

- Το στοιχείο `<binding>` καθορίζει το πρωτόκολλο επικοινωνίας του client με το Web Service.
- Το στοιχείο `<service>` καθορίζει την διεύθυνση της υπηρεσίας παρέχοντας το URL αυτής καθώς και ένα συγκεκριμένο `<interface>` όπως και αντίστοιχο `<binding>`.

WSDL 2.0 Binding Example

```
<description
  xmlns="http://www.w3.org/ns/wsd1"
  targetNamespace= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:tns= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:ghns = "http://greath.example.com/2004/schemas/resSvc"
  xmlns:wsoap= "http://www.w3.org/ns/wsd1/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  . . .
  <types>
    . . .
  </types>

  <interface name = "reservationInterface" >
    . . .
  </interface>
  <binding name="reservationSOAPBinding"
    interface="tns:reservationInterface"
    type="http://www.w3.org/ns/wsd1/soap"
    wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">
    <operation ref="tns:opCheckAvailability"
      wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>

    <fault ref="tns:invalidDataFault"
      wsoap:code="soap:Sender"/>
  </binding>
  . . .
</description>
```

WSDL 2.0 Components

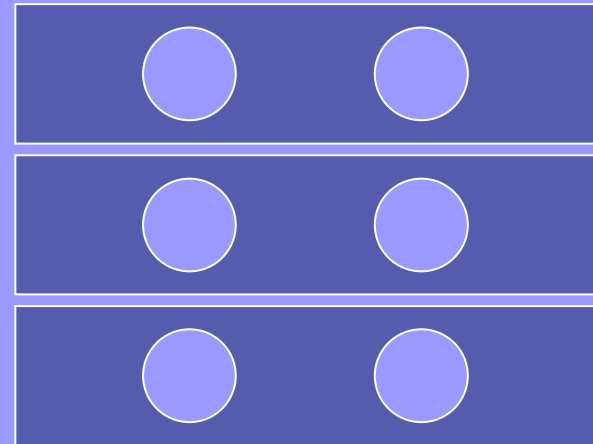
Bindings define the communication protocols used by the web service

```
<binding>
```

```
  definition of a  
  binding....
```

```
</binding>
```

Interface



Binding

+ Network address

WSDL 2.0 Service Example

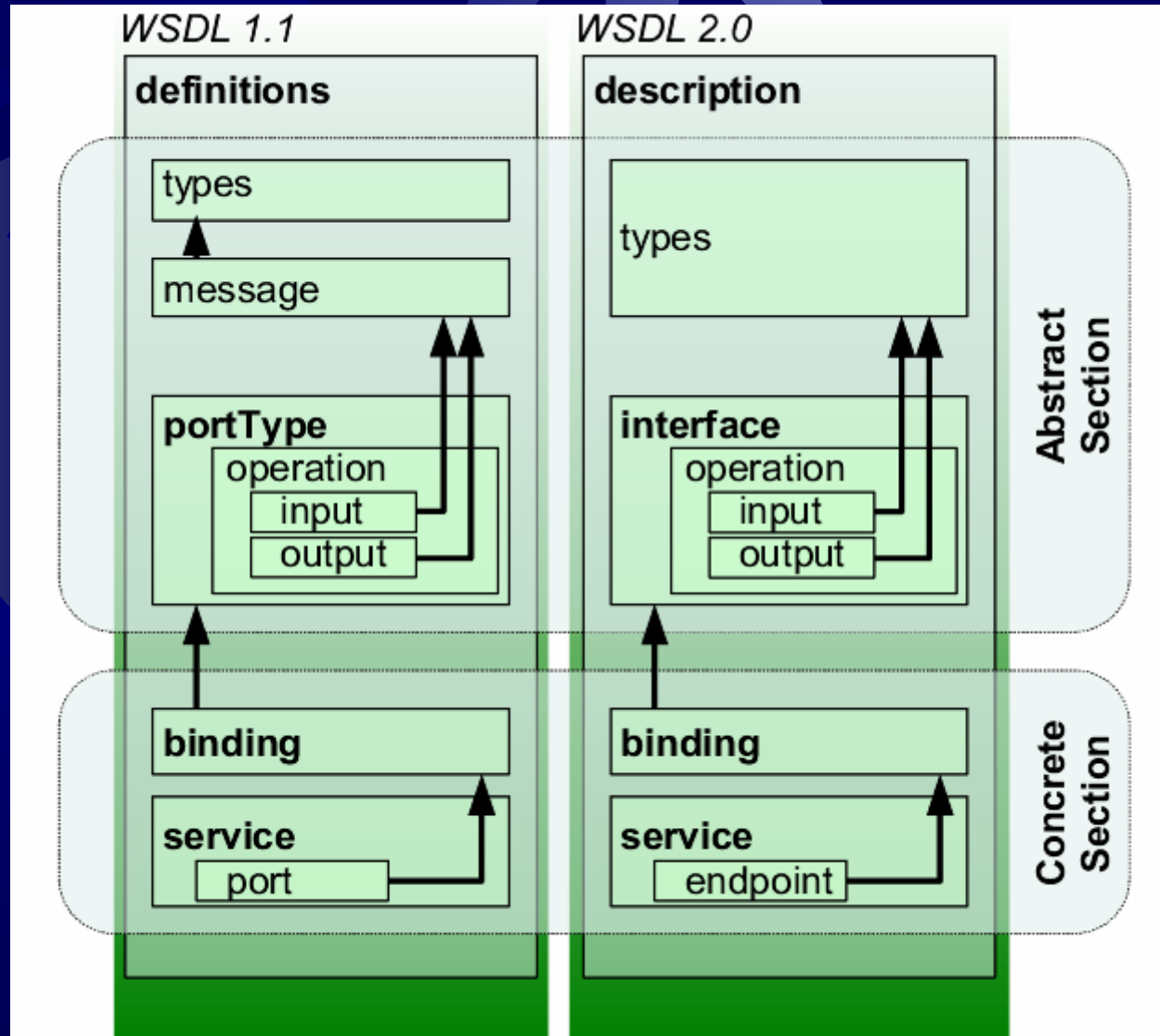
```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wsd1"
  targetNamespace= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:tns= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:ghns = "http://greath.example.com/2004/schemas/resSvc"
  xmlns:wssoap= "http://www.w3.org/ns/wsd1/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  . . .
  <types>
    . . .
  </types>

  <interface name = "reservationInterface" >
    . . .
  </interface>
  <binding name="reservationSOAPBinding"
    interface="tns:reservationInterface"
    . . . >
  </binding>
  <service name="reservationService"
    interface="tns:reservationInterface">
    <endpoint name="reservationEndpoint"
      binding="tns:reservationSOAPBinding"
      address = "http://greath.example.com/2004/reservation"/>
  </service>
</description>
```

WSDL 1.1 vs WSDL 2.0

- Η πλέον πρόσφατη έκδοσή της WSDL είναι η 2.0 η οποία αποτελεί πρότυπο της W3C από το 2007.
- Παρόλα αυτά η πρώτη έκδοση της WSDL ήταν η 1.1 .
- Η βασική διαφορά μεταξύ των δύο εκδόσεων είναι ότι η 1.1 χρησιμοποιούσε για την δημιουργία του binding μόνο τις μεθόδους POST και GET της HTTP ενώ η 2.0 τις χρησιμοποιεί όλες.
- Ωστόσο η υποστήριξη με εργαλεία ανάπτυξης για την έκδοση 2.0 είναι ακόμα μικρή ενώ αντίθετα για την 1.1 αρκετά μεγάλη.

WSDL 1.1 vs WSDL 2.0



Service Discovery: UDDI

UDDI: Protocol for discovering services

WSDL: Format for describing services

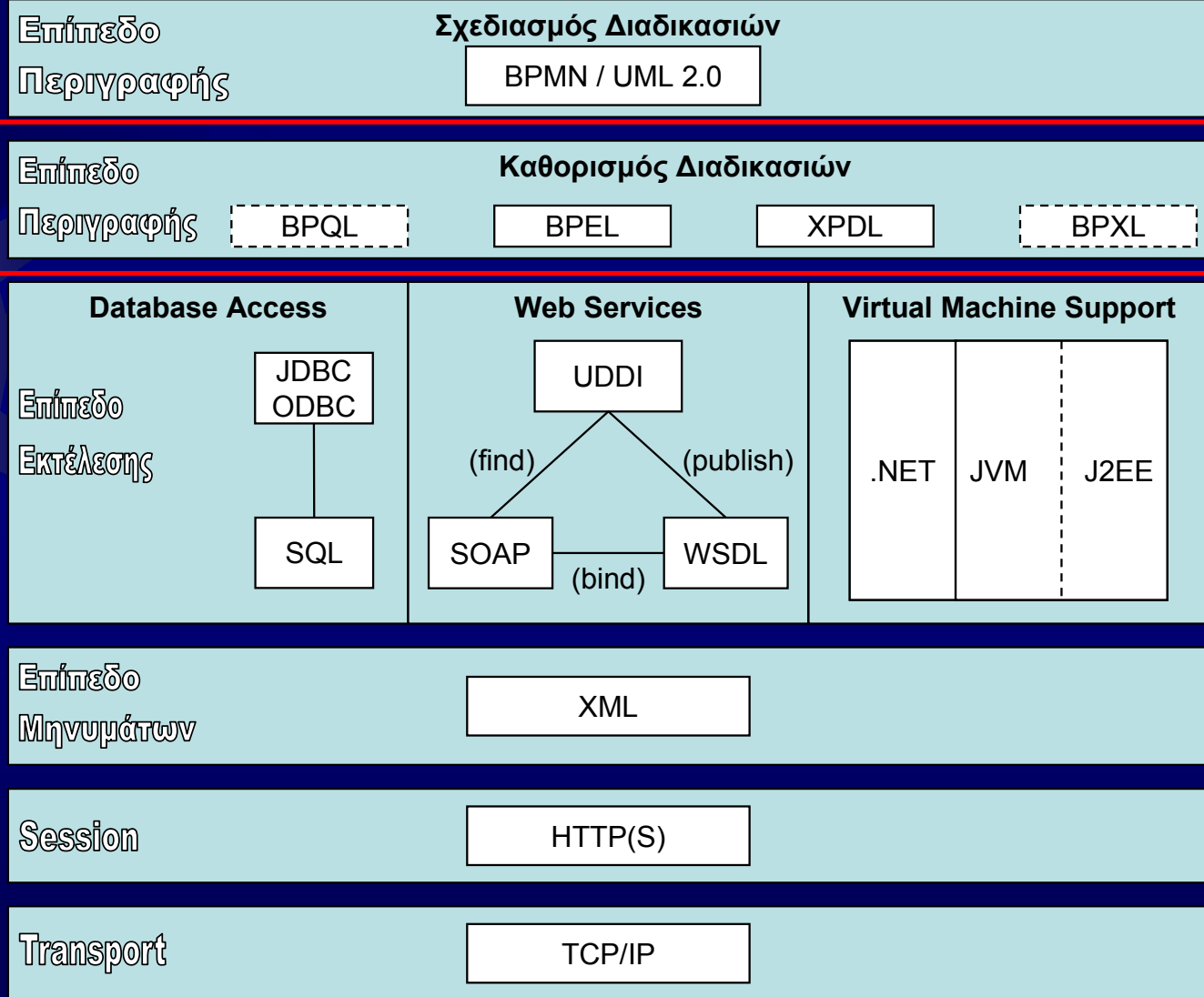
SOAP: Protocol for Remote Procedure Calls

XML: Data Encoding

HTTP: Data Transport

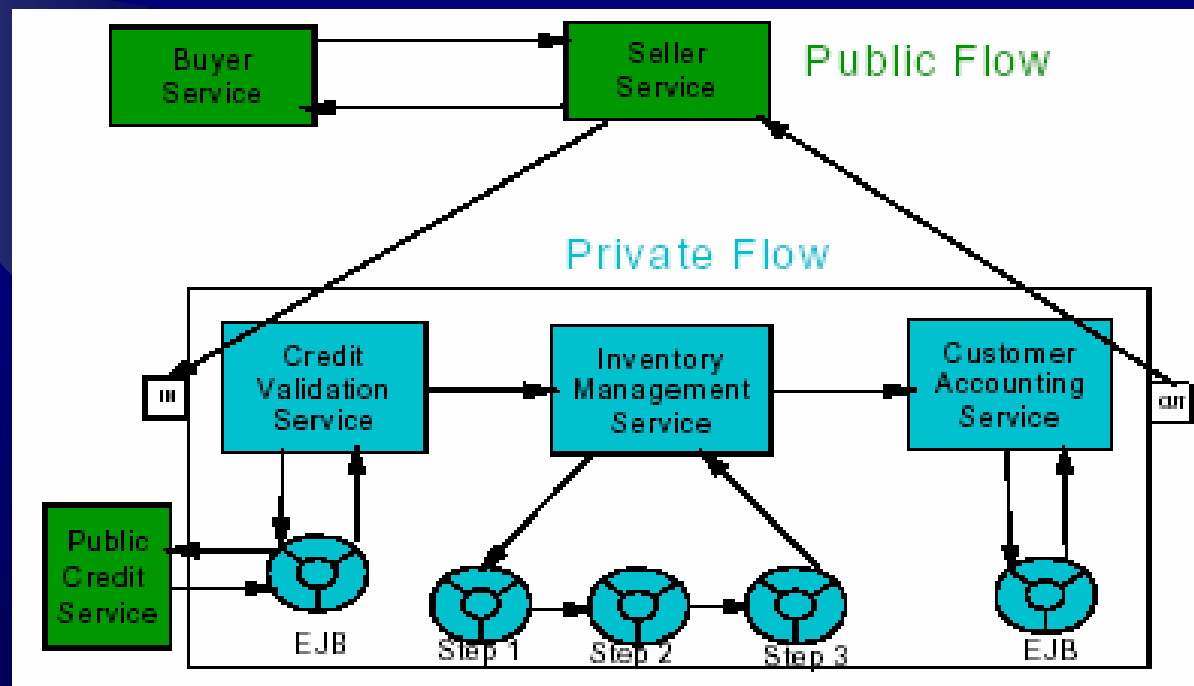
- UDDI: Universal Description, Discovery and Integration registry
 - Ανάλογο του Χρυσού Οδηγού για τα Web services
 - Στόχος είναι η δημιουργία ενός μοναδικού μητρώου που περιλαμβάνει όλα τα web services
 - Για παράδειγμα, “I want a web service that provides a tax calculator. Who offers such services?”
- Δημιούργημα τριών εταιρειών: Microsoft, IBM και Ariba
 - Συμμετέχουν: Dell, Fujitsu, HP, Hitachi,, Intel,, Oracle, SAP, Sun κ.α.

Τυποποιημένες Web τεχνολογίες υποστήριξης ΣΔΡΕ



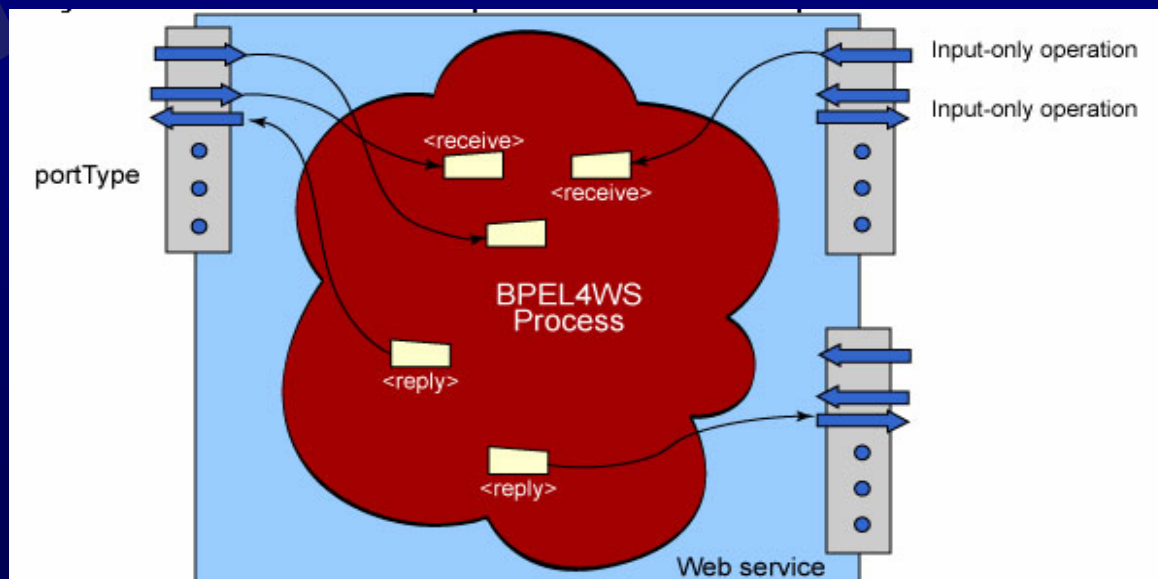
Καθορισμός Διαδικασιών

- Γιατί χρειάζεται;
 - Συντονισμός (choreography) των αλληλεπιδράσεων μεταξύ των εμπλεκόμενων βημάτων της διαδικασίας
 - Έλεγχος ροής (split, join, parallel, etc.)
 - Συνδυασμός των βημάτων της διαδικασίας για την υποστήριξη μιας ολοκληρωμένης επιχειρησιακής διαδικασίας



BPEL - Εισαγωγικά Στοιχεία

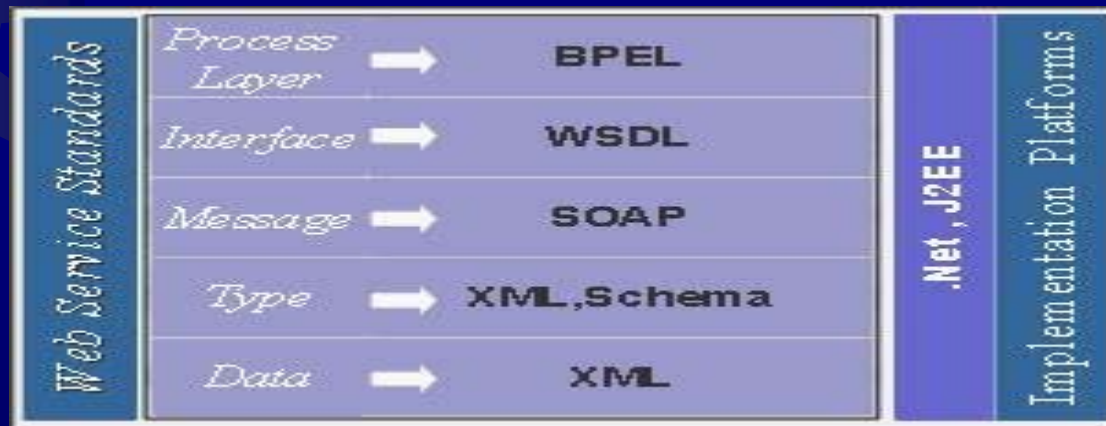
- Η BPEL είναι μια γλώσσα περιγραφής και εκτέλεσης επιχειρηματικών διαδικασιών με χρήση Web Services.
- Καθορίζει:
 - την πιθανή σειρά εκτέλεσης των λειτουργιών ενός συνόλου Web services
 - τα δεδομένα που ανταλλάσσονται μεταξύ των Web services
 - ποιοι είναι οι συμμετέχοντες και πως αυτοί εμπλέκονται στην επιχειρησιακή διαδικασία
 - ένα κοινό χειρισμό των εξαιρέσεων (joint exception handling) για τα σύνολα των Web services
 - και θέματα που αφορούν τα multiple services και τη συμμετοχή οργανισμών
- Αποτελεί την επιτομή της Service Oriented Architecture μέσω της :
 - Σύνθεσης
 - Ενορχήστρωσης
 - ΣυνεργασίαςWeb Services μέσω μιας νέας δομής που ονομάζεται Business Process.



BPEL - Εισαγωγικά Στοιχεία

Η Bpel είναι μια γλώσσα βασισμένη στην XML που υποστηρίζει την τεχνολογία των Web Services και συμπληρώνει την SOA επιπρόσθετα των παρακάτω τεχνολογιών:

- SOAP
- WSDL
- UDDI.



BPEL - Ιστορική Αναδρομή

- Η Bpel αντιπροσωπεύει την εξέλιξη δύο παλαιότερων γλωσσών περιγραφής ροών εργασίας, της WSFL, η οποία αναπτύχθηκε από την IBM και της XLANG η οποία αναπτύχθηκε από την Microsoft.
- Η πρώτη έκδοση της BPEL αναπτύχθηκε το 2002 και μέχρι το Μάρτιο του 2003 είχε υιοθετήσει η έκδοση 1.1 στην οποία πολλοί vendors βελτίωσαν και πρόσθεσαν νέα χαρακτηριστικά .
- Τέλος το 2007 ο οργανισμός OASIS παρουσίασε τη τελευταία έκδοση 2.0 .

BPEL

- Η bpel υποστηρίζει δύο τρόπους για την περιγραφή ενός process:
 - Τα εκτελέσιμα process που επιτρέπουν τον καθορισμό λεπτομερειών ως προς την εκτέλεση της διαδικασίας
 - Τα abstract process που δεν περιλαμβάνουν λεπτομέρειες των διαδικασιών που περιγράφουν και δεν είναι εκτελέσιμα. Είναι κυρίως αφαιρετικές γενικές διαδικασίες που αποτελούν πρότυπα του τρόπου ανταλλαγής της πληροφορίας μεταξύ των μελών μιας πιο συγκεκριμένης διαδικασίας.
- Ένα Bpel process μπορεί να είναι
 - Synchronous
 - Asynchronous
- Μπορεί να εκτελείται :
 - Σειριακά (sequential)
 - Παράλληλα (parallel)

BPEL - Εξωτερικές Επαφές (Partner Links)

- Ένα βασικό χαρακτηριστικό της BPEL είναι η δυνατότητα επικοινωνίας με Web Services και ο συντονισμός τους ώστε να συνθέσει μια ορισμένη επιχειρηματική λογική.
- Κάθε επαφή που γίνεται για την κλήση μιας υπηρεσίας ονομάζεται Partner link.
- Ένα Partner link είναι ένα στιγμιότυπο στο οποίο ορίζονται τα αναγκαία WSDL port types που προσφέρονται από την BPEL και απαιτούνται από την υπηρεσία για να γίνει η επαφή με αυτήν.

```
<partnerLinks>  
<partnerLink name="ClientStartUpLink"  
partnerLinkType="wsdl:ClientStartUpPLT" myRole="Client" />  
</partnerLinks>
```

BPEL - Μεταβλητές

```
<variables>  
<variable name="myVar1" messageType="myNS:myWSDLMessageDataType" />  
<variable name="myVar1" element="myNS:myXMLElement" />  
<variable name="myVar2" type="xsd:string" />  
<variable name="myVar3" type="myNS:myComplexType" />  
</variables>
```

Οι μεταβλητές μπορούν να δηλωθούν είτε μέσα σε ένα scope είτε εντός του process στην πρώτη περίπτωση είναι τοπικής εμβέλειας ενός στην δεύτερη global.

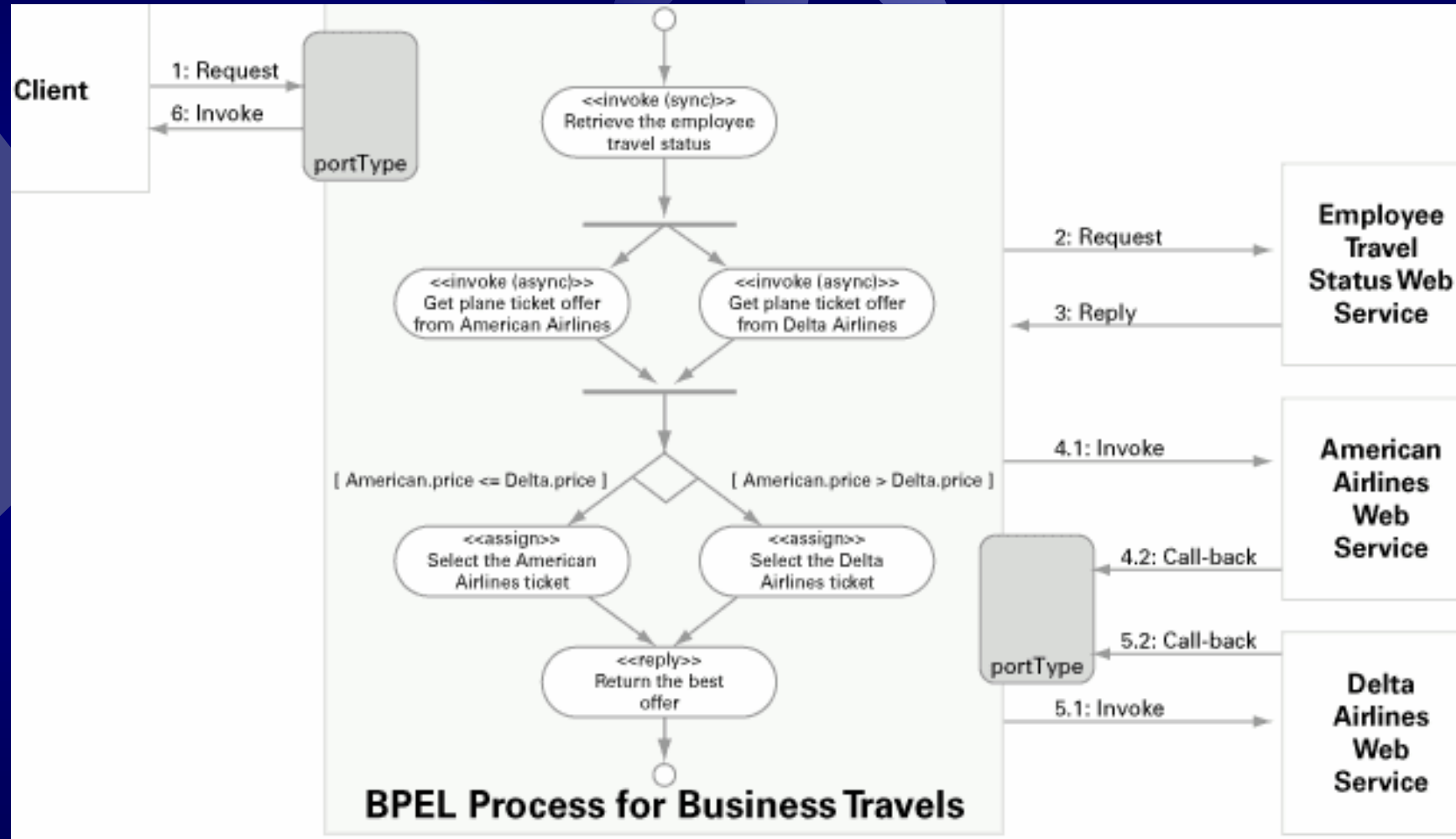
BPEL - Basic Activities

- Κλήση άλλων Webservice <invoke>
- Αναμένοντας τον χρήστη να κάλέσει το process και να αποστείλει ένα μήνυμα δεδομένα <receive> (receiving a request)
- Δημιουργία μιας απάντησης στην περίπτωση synchronous operations <reply>
- Χειρισμός δεδομένων μεταβλητών <assign>
- Χειρισμός λαθών και εξαιρέσεων <throw>
- Αναμονή για ένα συμβάν <wait>
- Τερματισμός ολόκληρου του process <terminate>

BPEL - Structure Activities

- Μπορούμε να συνδυάσουμε τις προηγούμενες βασικές δραστηριότητες και να δομήσουμε μια περιπλοκή επιχειρηματική λογική.
- Στο ρεπερτόριο της Bpel για τον σκοπό αυτό υπάρχουν οι structure activities που μας επιτρέπουν με την χρησιμοποίησή τους να συνθέσουμε πολύπλοκες δομές.
 - `<sequence>` που μας επιτρέπει να ορίσουμε μια σειρά δραστηριοτήτων που θα εκτελεστούν σειριακά.
 - `<flow>` μας επιτρέπει να ορίσουμε ποιες δραστηριότητες θα εκτελεστούν παράλληλα.
 - Case-switch ,if-else construct (`<switch>`) για την αλλαγή της πορείας εκτέλεσης κατά περίπτωση.
 - `<while>` για την δημιουργία επαναλήψεων.
 - `<pick>` για την δυνατότητα επιλογής ανάμεσα σε εναλλακτικές διαδρομές.

BPEL - ένα τυπικό παράδειγμα



BPEL - Invoke

Η δραστηριότητα αυτή χρησιμοποιείται για την κλήση ενός Web Service που παρέχεται από μια εξωτερική επαφή.

Μια invoke δραστηριότητα μπορεί να καλέσει μια μονόδρομη λειτουργία και να μην περιμένει απάντηση είτε μια αμφίδρομη ή οποία θα σταματήσει την εξέλιξη του process και θα αναμένει απάντηση από την εξωτερική επαφή

```
<invoke name="RequestResponseInvoke"  
partnerLink="BusinessPartnerServiceLink"  
operation="RequestResponseOperation"  
inputVariable="Input "  
outputVariable="Output " />
```

Απαραίτητα στοιχεία που πρέπει να ορισθούν είναι το partnerLink (το όνομα του Web Service) και ένα operation του web service που θα κληθεί.

BPEL - Receive

Ο σκοπός μιας receive activity είναι να λαμβάνει μηνύματα από μια εξωτερική επαφή. Για αυτό το λόγο πάντα σε μια receive activity ορίζεται το αναγκαίο Partner link και το operation , ενώ θα πρέπει να ορισθεί επίσης και μια μεταβλητή (ή ένα σύνολο από μεταβλητές) στη οποία θα αποθηκεύονται τα ζητούμενα από την εξωτερική επαφή δεδομένα.

```
<receive name="ReceiveRequestFromPartner"  
createInstance="yes"  
partnerLink="ClientStartUpPLT"  
operation="StartProcess" ... />
```

BPEL - Reply

Κάθε receive activity μπορεί να συνδυαστεί με μια reply activity στην περίπτωση που θέλουμε να έχουμε μια επικοινωνία με την εξωτερική επαφή της μορφής request-response.

```
<reply name="ReplyResponseToPartner"  
partnerLink="ClientStartUpPLT"  
operation="StartProcess"  
variable="ClientAnswer"  
faultName="ClientAnswer".../>
```

BPEL - Assign

Η δραστηριότητα αυτή χρησιμοποιείται για την αντιγραφή των δεδομένων που προέρχονται από Partner Links και που θα πρέπει μεταφέρονται στις μεταβλητές του process για τον χειρισμό τους.

```
<assign>  
<copy>  
<from variable= "SomethingMessage" />  
<to variable="SomethingElseMessage" />  
</copy>  
</assign>
```

BPEL - Throw

Η Throw χρησιμοποιείται όταν το process πρέπει να επισημάνει μια λάθος που συμβαίνει εσωτερικά.

```
<throw xmlns:FLT="http://example.com/faults"  
faultname=FLT:"OutOfStock" />
```

BPEL - Wait

Η δραστηριότητα αυτή ορίζει την καθυστέρηση για μια συγκεκριμένη χρονική περίοδο μέχρις ότου συμβεί κάτι.

Μια τυπική χρήση αυτής της δραστηριότητας είναι να γίνει invoke ένα operation σε μια συγκεκριμένη χρονική στιγμή.

```
<sequence>  
<wait>  
<until>'2012-12-21T00:00'</until>  
<invoke> partnerLink="SaveUsPlease" portType="AutomaticSavingOfSouls"  
operation="PleaseForgiveMeFather"  
inputVariable="NumberOfSeans" />  
</sequence>
```

BPEL - Terminate

Η δραστηριότητα αυτή χρησιμοποιείται για τον άμεσο τερματισμό του process χωρίς να γίνεται χειρισμός λαθών ή να υπάρχει διαδικασία να ξεκινήσει το process από την αρχή.

BPEL - Sequence

Η ιδιότητα αυτή μας επιτρέπει να συμπεριλάβουμε μια σειρά από άλλες δραστηριότητες οι οποίες θα πρέπει να εκτελεστούν με μια ορισμένη σειρά .

```
<sequence name="InvertMessageOrder">  
<receive name="receiveOrder" ... />  
<invoke name="checkPayment" ... />  
<invoke name="shippingService" ... />  
<reply name="sendConfirmation" ... />  
</sequence>
```

Στο παραπάνω παράδειγμα οι δραστηριότητες

- receiveOrder ,
- checkPayment,
- shippingService ,
- sendConfirmation

εκτελούνται με την σειρά που αναγράφονται εντός της sequence.

BPEL - Flow

Η δραστηριότητα αυτή χρησιμοποιείται για τον συγχρονισμό και την παράλληλη εκτέλεση δραστηριοτήτων.

```
<flow ...>
<links> ... </links>
<documentation>
check availability of a flight, hotel and rental car concurrently
</documentation>
<invoke name="checkFlight" ... />
<invoke name="checkHotel" ... />
<invoke name="checkRentalCar" ... />
</flow>
```

Οι τρεις δραστηριότητες

(*checkFlight*, *checkHotel* and *checkRentalCar*)

εκτελούνται παράλληλα ενώ και οι τρεις ξεκινούν ταυτόχρονα όταν η flow ξεκινά η οποία τερματίζει όταν τελειώσουν οι δραστηριότητες την εκτέλεσή τους.

BPEL - Flow links

Τα flow links χρησιμοποιούνται για τον συντονισμό στην εκτέλεση των υποκείμενων εντός του flow δραστηριοτήτων.

```
<flow ...>
<links>
<link name="checkFlight-To-BookFlight" />
</links>
<documentation>
check availability of a flight, hotel and rental car concurrently
</documentation>
<invoke name="checkFlight" ...>
<sources>
<source linkName="checkFlight-To-BookFlight" />
</sources>
</invoke>
<invoke name="checkHotel" ...
<invoke name="checkHotel" ... />
<invoke name="checkRentalCar" ... />
<invoke name="bookFlight" ...>
<targets>
<target linkName="checkFlight-To-BookFlight" />
</targets>
</invoke>
</flow>
```

Η σύνδεση 2 δραστηριοτήτων που πρέπει να συγχρονιστούν πραγματοποιείται με τη χρήση των στοιχείων **source** και **target**

BPEL - if-else

Η δραστηριότητα if-else activity μας επιτρέπει να δημιουργήσουμε κλαδιά στην δομή του process ανάλογα αν οι συνθήκες αλήθειας που έχουμε είναι αληθείς ή ψευδείς.

```
<if name="isOrderBiggerThan5000Dollars">
  <condition>
    $order > 5000
  </condition>
  <invoke name="calculateTenPercentDiscount" ... />
<elseif>
  <condition>
    $order > 2500
  </condition>
  <invoke name="calculateFivePercentDiscount" ... />
</elseif>
```

BPEL - While

Η δραστηριότητα αυτή είναι γνωστή από τις κλασσικές γλώσσες προγραμματισμού και σκοπός της είναι η επανάληψη μιας διαδικασίας όσο η συνθήκη αληθείας ικανοποιείται.

```
<while>  
  $iterations > 3  
</condition>  
  <invoke name="increaseIterationCounter" .../>  
</while>
```

BPEL - Repeat Until

Η δραστηριότητα επανάληψης έχει την διαφορά σε σχέση με την while ότι η υποκείμενη δραστηριότητα θα εκτελεστεί τουλάχιστον μία φορά γιατί η συνθήκη αληθείας βρίσκεται στο τέλος της δομής της.

```
<repeatUntil>  
<invoke name="increaseIterationCounter" ... />  
<condition>  
$iterations > 3  
</condition>  
</repeatUntil>
```

BPEL - forEach

Σε αυτή την δραστηριότητα επιλέγουμε να εκτελέσουμε την υποκείμενη δραστηριότητα (μία η περισσότερες) N φορές. Υπάρχουν δύο παραλλαγές της forEach δραστηριότητας , η παράλληλη εκτέλεση και η σειριακή.

```
<forEach parallel="no" counterName="N" ...>
<startCounterValue>1</startCounterValue>
<finalCounterValue>5</finalCounterValue>
<scope>
<documentation>check availability of each item ordered</documentation>
<invoke name="checkAvailability" ... />
</scope>
</forEach>
```

Ένας περιορισμός που υπάρχει στην δραστηριότητα αυτή είναι ότι ενώ στις άλλες δραστηριότητες μπορούμε να έχουμε σαν υποκείμενες προς επανάληψη άλλες δραστηριότητες , εδώ μπορούμε να έχουμε μόνο μια scope δραστηριότητα.

BPEL - pick

Η δραστηριότητα αυτή χρησιμοποιείται για την αναμονή κάποιου εκ των καθορισμένων συμβάντων. Ενεργοποιείται με την πραγματοποίηση του πρώτου χρονικά συμβάντος από το προκαθορισμένο σύνολο.

Εκτελεί τη δραστηριότητα που συσχετίζεται με το συμβάν και παύει να αναμένει τα υπόλοιπα συμβάντα του συνόλου.

```
<pick>
  <onMessage partnerLink="buyer" portType="orderEntry"
    operation="inputLineItem" variable="lineItem">
    <!-- activity to add line item to order -->
  </onMessage>

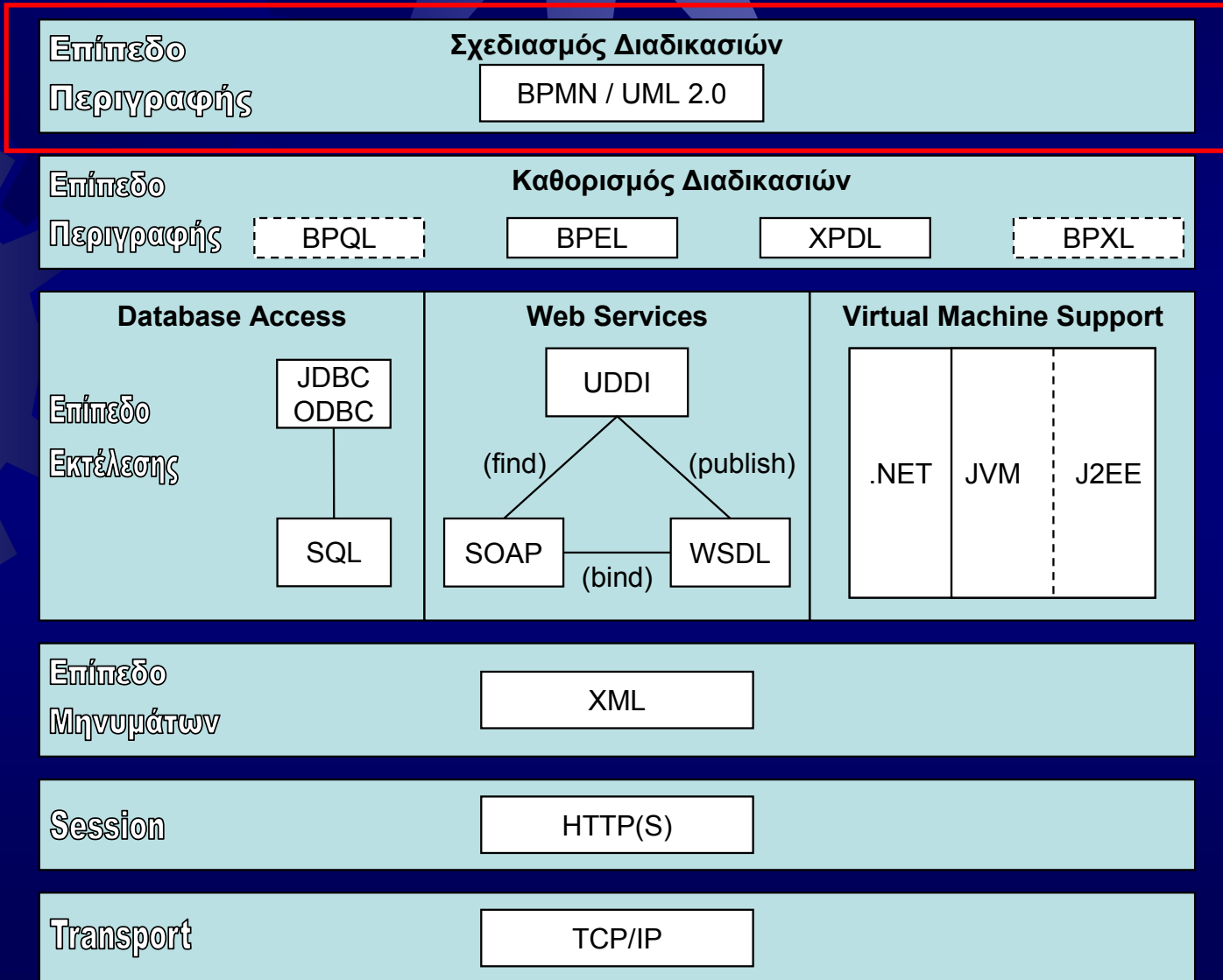
  <onMessage partnerLink="buyer" portType="orderEntry"
    operation="orderComplete" variable="completionDetail">
    <!-- activity to perform order completion -->
  </onMessage>

  <!-- set an alarm to go after 3 days and 10 hours -->
  <onAlarm for=" 'P3DT10H' ">
    <!-- handle timeout for order completion -->
  </onAlarm>
</pick>
```


ΕΠΙΛΟΓΟΣ - ΠΟΥ ΧΡΕΙΑΖΕΤΑΙ ΤΕΛΙΚΑ Η BPEL;

- Η χρήση της Bpel εντός ενός οργανισμού γίνεται με σκοπό καθοριστεί βάσει κοινών προτύπων η συνεργασία των εφαρμογών καθώς επίσης και για να βελτιωθεί η λειτουργικότητα παλαιών stand alone εφαρμογών.
- Η Bpel βοηθάει έναν οργανισμό να καθορίσει με τον πλέον βέλτιστο τρόπο της επιχειρηματικές του διαδικασίες .
- Η χρήση της Bpel δεν έρχεται να αντικαταστήσει ήδη υπάρχουσες λειτουργικές εφαρμογές αλλά να βελτιώσει το ήδη υπάρχων λειτουργικό περιβάλλον.
- Η BPEL είναι η τεχνολογία που χρειάζεται κάθε οργανισμός ο οποίος σκοπεύει ή ήδη δημοσιοποιεί την λειτουργικότητά του μέσω Web Services.

Τυποποιημένες Web τεχνολογίες υποστήριξης ΣΔΡΕ

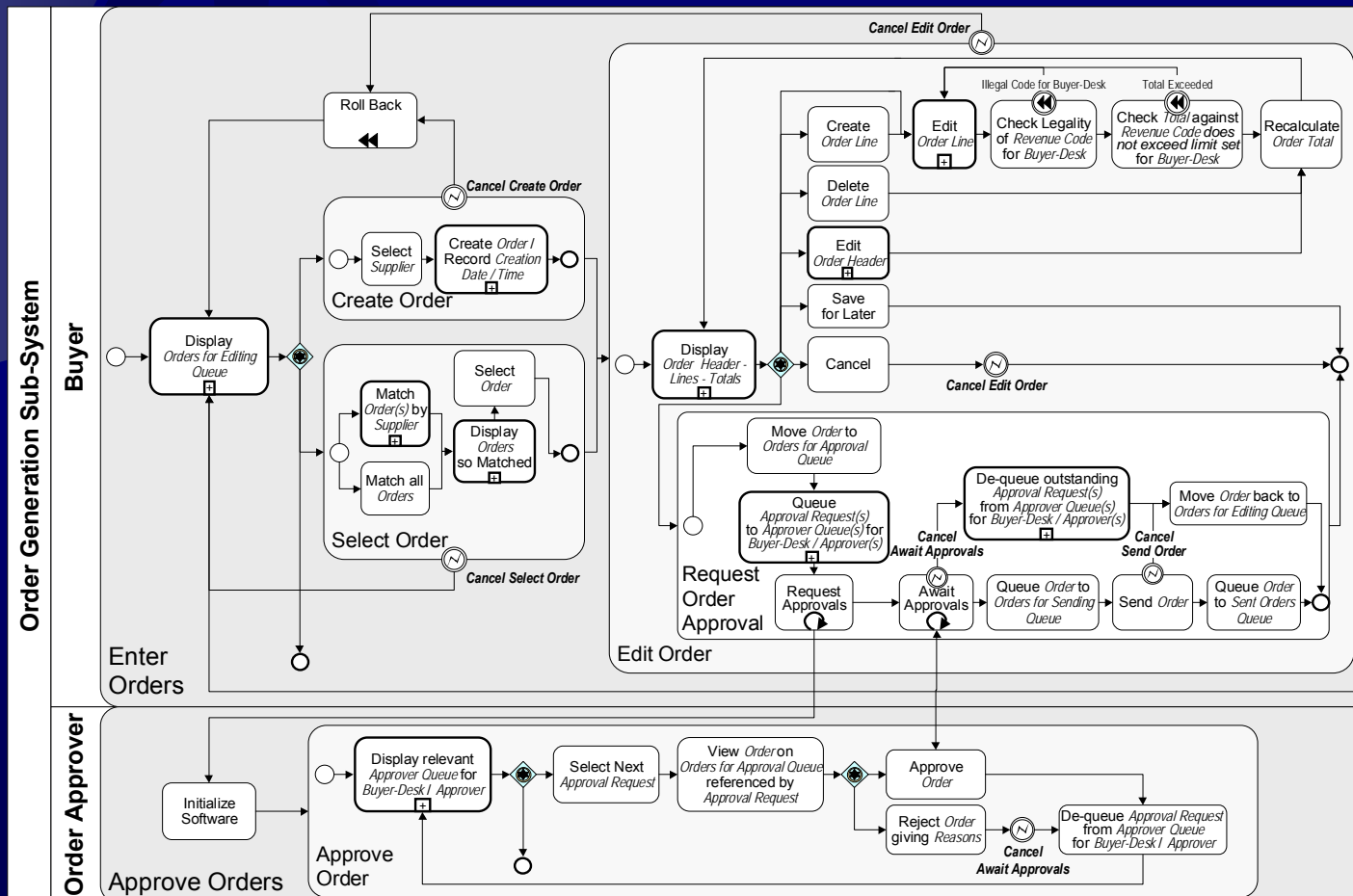


Επίπεδο περιγραφής (Presentation Layer)

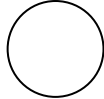
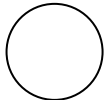
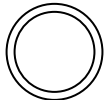
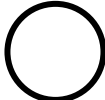


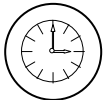

















- Υπάρχουν διάφορες γλώσσες μοντελοποίησης, ορισμένες περισσότερο γενικές ως προς τη χρήση τους και άλλες εξειδικευμένες
 - Στην περίπτωση της γενικής μοντελοποίησης η γλώσσα Business Process Modeling Notation (BPMN) χρησιμοποιείται ευρέως από τους εταιρικούς χρήστες
 - Η γλώσσα Universal Modelling Language (UML) χρησιμοποιείται ευρέως από τους προγραμματιστές (αν και μπορεί να γίνει σχετικά εύκολα κατανοητή και από τους εταιρικούς χρήστες)

Business Process Modelling Notation (BPMN)

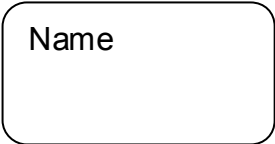



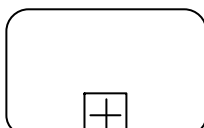
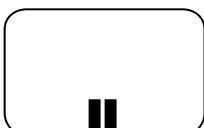
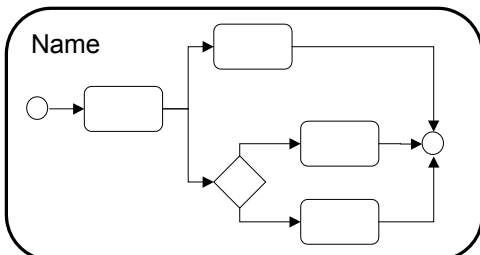


- Κατανοητός συμβολισμός (εύκολος στη μάθηση)
- Ιδιαίτερα εκφραστικός (μπορεί να απεικονίσει πολλές πολύπλοκες καταστάσεις)




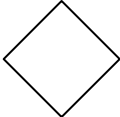





Σύμβολα BPMN: Γεγονότα (Events)

Element				Description							
<h2>Event</h2>  <p>Name or Source</p>				<p>An Event is something that “happens” during the course of a business process that affects the flow of the process. An event has a cause (trigger) and/or an effect (result). There are three basic types of event - start, intermediate, end.</p>							
	Start	Int.	End		Start	Int.	End		Start	Int.	End
General				Exception				Timer			
Link				Cancel				Terminate			
Message				Compensation							
Multiple				Rule							

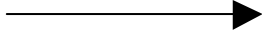
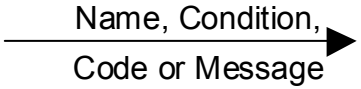
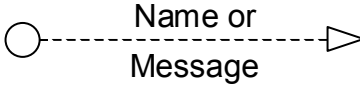
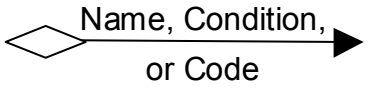
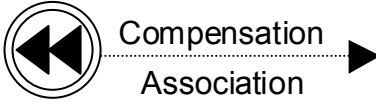
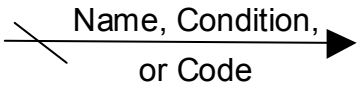
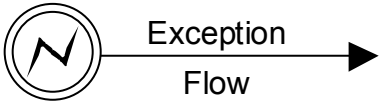

Σύμβολα BPMN, Διαδικασίες / Εργασίες (Processes / Tasks)

Element	Description
<p>Process / Task</p> 	<p>A Process is a compound activity that can be broken down into a finer level of detail, such as another process (a sub-process) or a task. A Task is an atomic activity for which it is decided that it does not need to be broken down further.</p>
<p>Task</p> 	<p>Transaction</p> 
<p>Activity Looping</p> 	<p>Collapsed Sub-Process</p> 
<p>Multiple Instance Looping</p> 	<p>Expanded Sub-Process</p> 
	<p>Ad-Hoc Sub-Process</p> 
	<p>Compensation Sub-Process</p> 



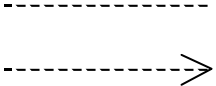

Σύμβολα BPMN, Gateways

Element	Description				
<p style="text-align: center;">Gateway</p> 	<p>A gateway is used to control the divergence and convergence of multiple Sequence Flows. Thus, it will determine branching, forking, merging and joining of paths.</p>				
Exclusive OR Data-Based			Inclusive OR		
			Complex		
Exclusive OR Event-Based			Parallel AND		

Σύμβολα BPMN, Ροές (Sequence Flows)

Element	Description		
<p style="font-size: 2em; text-align: center;">Sequence Flow</p> 	<p>A Sequence Flow is used to show the order that activities will be performed in a process. For each Sequence Flow a “Token” will flow from the source object(s) to the target object(s) - like a baton in a race.</p>		
Normal / Un-controlled Flow		Message Flow	
Conditional Flow		Compensation Association	
Default Flow (Data-Based Decisions Only)			
Exception Flow		Data Object	

Σύμβολα BPMN, Διάφορα (Miscellaneous)

Element	Description						
<p style="text-align: center;">Miscellaneous</p>	<p>Aside from Events, Process / Tasks, Gateways and Sequence Flows, there are a variety of other miscellaneous symbols used in a BPMN diagram.</p>						
<p>Group (for documentation purposes)</p>  <p>Off-Page Connector</p>  <p>(General) Association</p> 	<p>Text Annotation</p>  <p>Pool</p> <table border="1" data-bbox="1043 938 1854 1129"> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Name</td> <td></td> </tr> </table> <p>Lanes</p> <table border="1" data-bbox="1043 1161 1854 1353"> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Name</td> <td>Name</td> </tr> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Name</td> <td>Name</td> </tr> </table>	Name		Name	Name	Name	Name
Name							
Name	Name						
Name	Name						

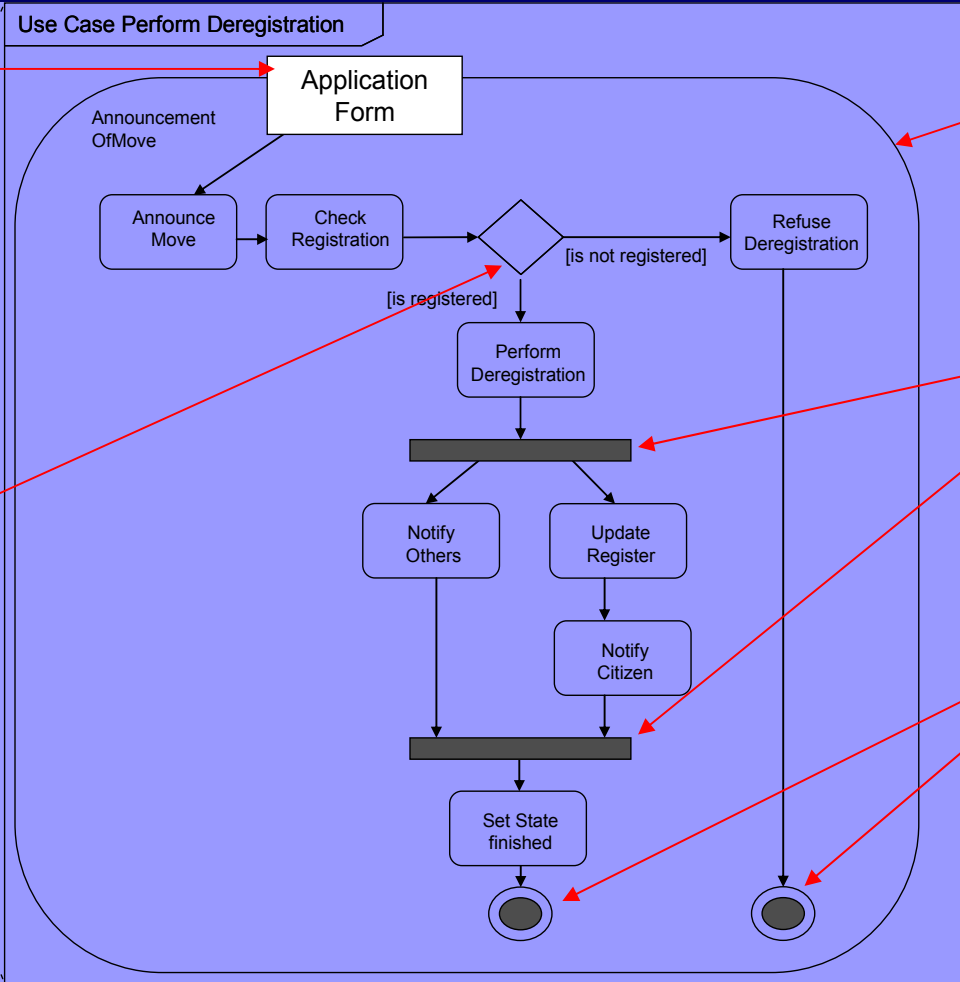
Universal Modelling Language 2.0

- Η UML παρέχει τη δυνατότητα μοντελοποίησης της σχέσης μεταξύ των Use Cases και της αναλυτικής περιγραφής των δραστηριοτήτων τους

Αντικείμενο
(Object Node)

Perform Deregistration

Σύνδεση
explicit OR



Δραστηριότητα

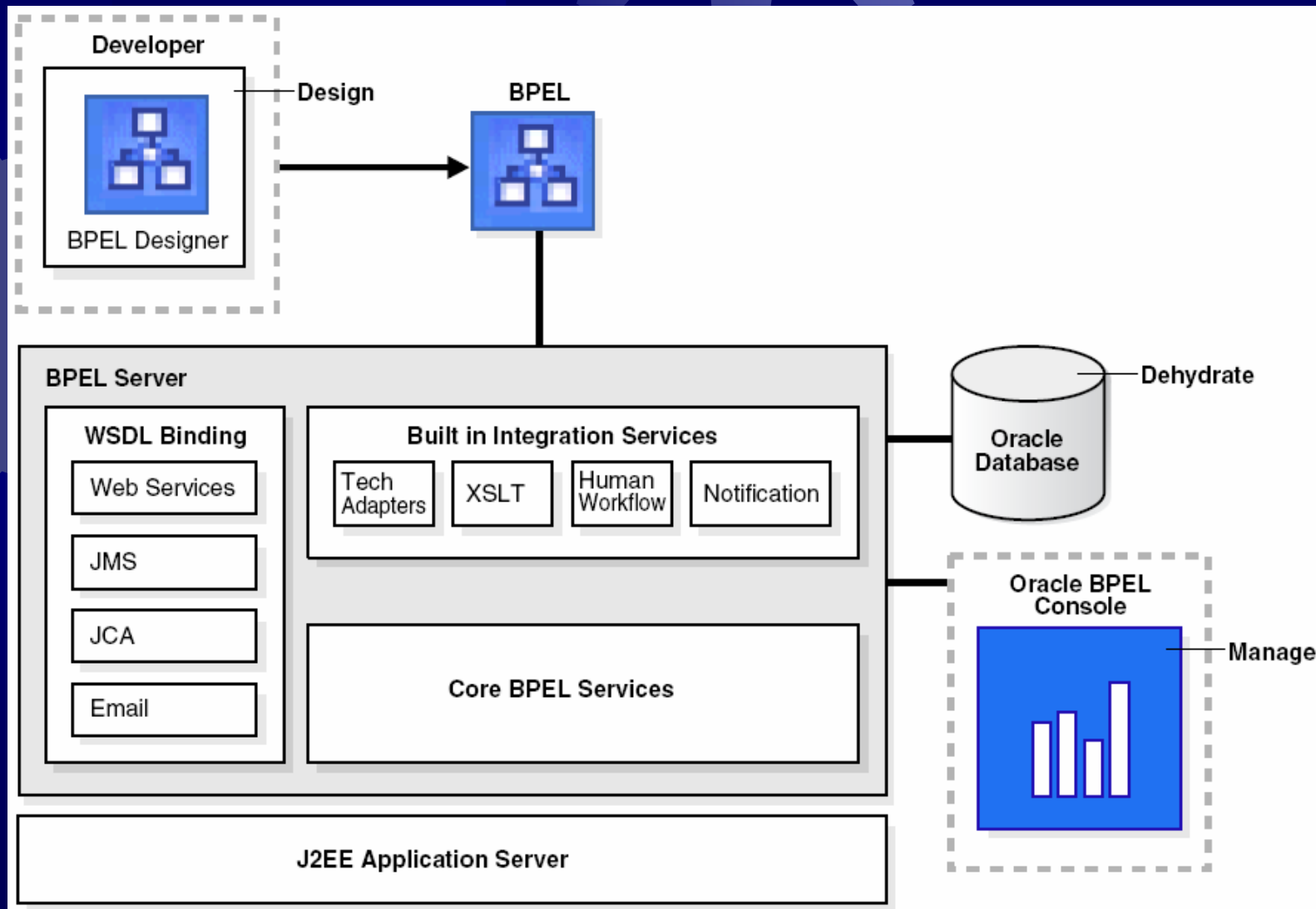
Σύνδεση AND

terminators



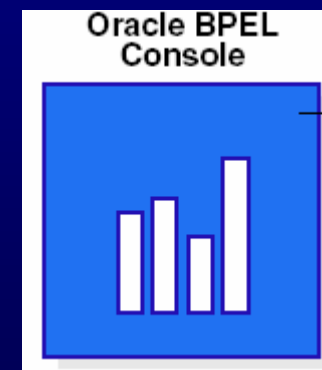
BPEL Demo

Οι τεχνολογίες του DEMO



Ο σκοπός του DEMO

- Σχεδιασμός, ανάπτυξη και έλεγχος της ροής εργασίας που σχετίζεται με τη πιστοληπτική αξιολόγηση ενός ατόμου (Credit Rating)
- Όταν εκτελούμε τη ροή εργασίας:
 - μας ζητείται να εισάγουμε το ΑΦΜ μας (μέσα από μία HTML φόρμα)
 - το σύστημα επιστρέφει τη πιστοληπτική αξιολόγηση μας
- Κατά τη διάρκεια του DEMO θα σχεδιάσουμε λειτουργίες (διαδικασίες) που επικοινωνούν με τη λειτουργία CreditRating
 - η λειτουργία αυτή υπάρχει ήδη
 - με `obant` κάνουμε `deploy and start` τη λειτουργία
 - με το `console` ελέγχουμε αν τρέχει

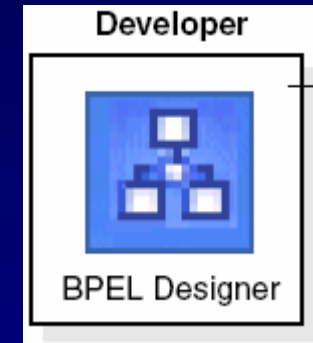


Εκκίνηση

- Εκκινούμε JDeveloper BPEL Designer
- Εκκινούμε Oracle BPEL Server
- Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager 10.1.2 > Developer Prompt to open up a command prompt
- Change directories
 - `cd utils\CreditRatingService`
- Enter the following command:
 - `obant`
- Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager 10.1.2 > BPEL Console
 - Password BPEL

Δημιουργία νέου Project και Workspace

- με τον developer δημιουργούμε:
 - BPEL process source (projectname.bpel)
 - Web services description language client interface (projectname.wsdl)
 - BPEL process deployment descriptor (bpel.xml)



Δημιουργία νέου Project και Workspace

- 1. Return to JDeveloper BPEL Designer
- 2. Select File > New from the main menu
- 3. Double-click Workspace in the Items window to display the Create Workspace window
- 4. Enter myBPELworkspace in the Workspace Name field and accept the default path in the Directory Name field
- 5. Deselect the Add a New Empty Project check box
- 6. Click OK
- 7. Right-click myBPELworkspace in the Applications Navigator section
- 8. Select New Project to define a new BPEL process project
- 9. Double-click BPEL Process Project in the Items window to display the BPEL Process Project window
- 10. Enter myCreditFlow in the BPEL Process Name field. All other fields default to the correct values for creating an asynchronous BPEL process
- 11. Click OK. The BPEL process files are created in the Oracle_Home\integration\jdev\jdev\mywork\myBPELworkspace\myCreditFlow directory
- 12. The following sections appear. If they do not appear, click Diagram View in JDeveloper BPEL Designer and double-click myCreditFlow.bpel in the Applications Navigator section. See the Oracle BPEL Process Manager Developer's Guide for a description of JDeveloper BPEL Designer sections.

Viewing the WSDL source code

- Double-click myCreditFlow.wsdl in the Applications Navigator section
- A myCreditFlowProcessRequest element is accepted as input
- A myCreditFlowProcessResponse element is returned as output

```
<element name="myCreditFlowProcessRequest">
  <complexType>
    <sequence>
      <element name="input" type="string"/>
    </sequence>
  </complexType>
</element>
<element name="myCreditFlowProcessResponse">
  <complexType>
    <sequence>
      <element name="result" type="string"/>
    </sequence>
  </complexType>
</element>
```

Ports and operations

- One operation initiates the asynchronous process

```
<portType name="myCreditFlow">
  <operation name="initiate">
    <input
      message="client:myCreditFlowRequestMessage" />
    </operation>
</portType>
```

- The other operation calls back the client with the asynchronous response

```
<portType name="myCreditFlowCallback">
  <operation name="onResult">
    <input
      message="client:myCreditFlowResponseMessage" />
    </operation>
</portType>
```

Partners and roles

- The `partnerLinkType` for this asynchronous process has two roles
 - one role is for the service provider, and
 - the other is for the requester

```
<plnk:partnerLinkType name="myCreditFlow">
  <plnk:role name="myCreditFlowProvider">
    <plnk:portType name="client:myCreditFlow"/>
  </plnk:role>
  <plnk:role name="myCreditFlowRequester">
    <plnk:portType
name="client:myCreditFlowCallback"/>
  </plnk:role>
```

- `</plnk:partnerLinkType>`

Αλλαγή του WSDL αρχείου

- Αλλαγή του `input` σε `ssn`.
- Αλλαγή του `result` σε `creditRating`.
- Αλλαγή του `string` σε `int`.

Δημιουργία και configuration εξωτερικών λειτουργιών

- Δημιουργία Credit Rating Service Partner Link
 - Τα Partner Link προσδιορίζουν τις εξωτερικές λειτουργίες με τις οποίες η λειτουργία που δημιουργούμε αλληλεπιδρά.
- Δημιουργία Scope Activity
 - Τα Scope Activity είναι ανάλογα με κομμάτια { } σε γλώσσες προγραμματισμού
- Δημιουργία Invoke Activity στο εσωτερικό του Scope Activity
 - Τα Invoke Activity επιτρέπουν το προσδιορισμό του Operation που θέλουμε να καλέσουμε
- Δημιουργία Initial Assign Activity στο εσωτερικό του Scope Activity
 - Τα Assign Activity επιτρέπουν απλούς μετασχηματισμούς δεδομένων. Εδώ παίρνει το ssn και καλεί το Credit Rating Service για να δώσει το Credit Rating

Έλεγχος, compiling και ανάπτυξη (deployment) της ροής εργασίας

Τρέξιμο και έλεγχος ροής εργασίας