

# Telecommunication Services Engineering (TSE) Lab

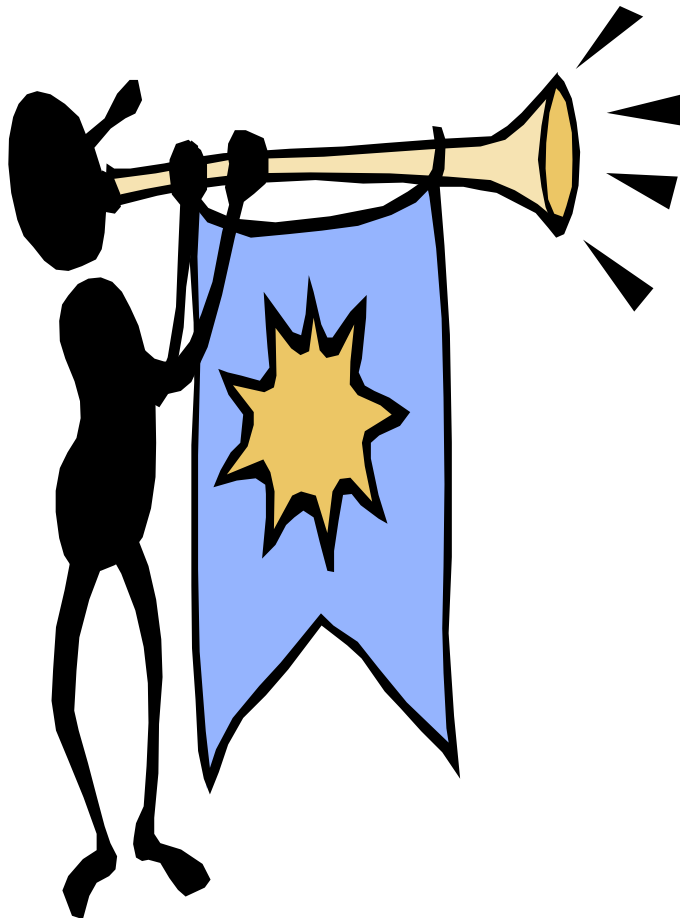


## Chapter VII-

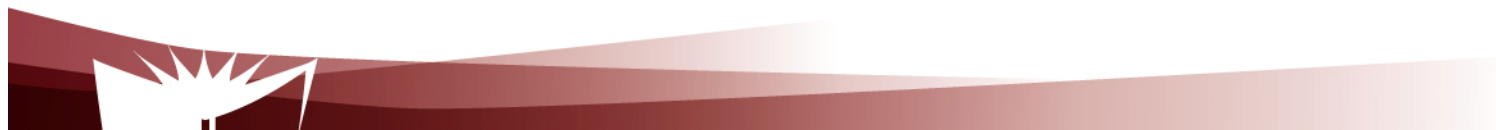
### SOAP Based - Web Services For Value Added Services (VAS) in NGNs

<http://users.encs.concordia.ca/~glitho/>

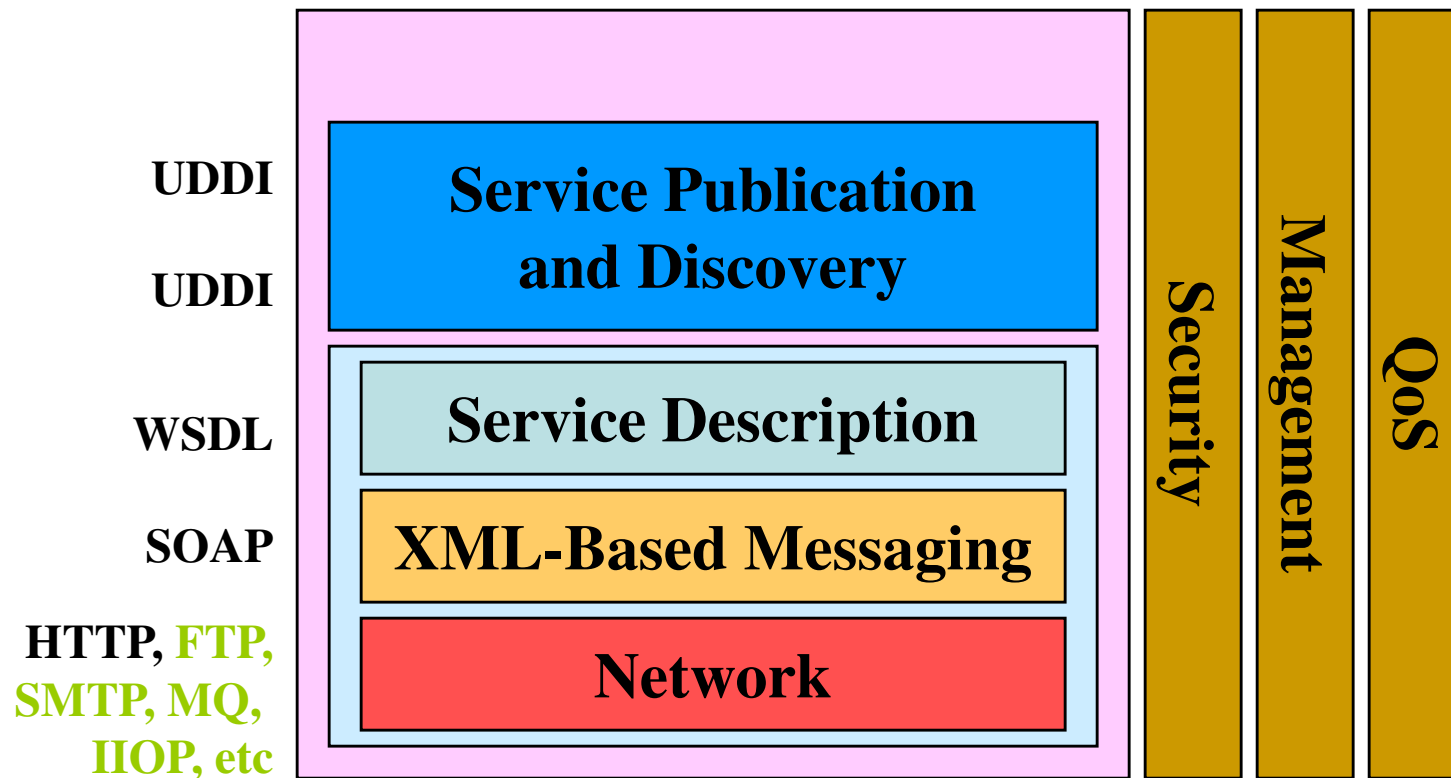
## Outline



1. Technologies
2. Applications to NGN



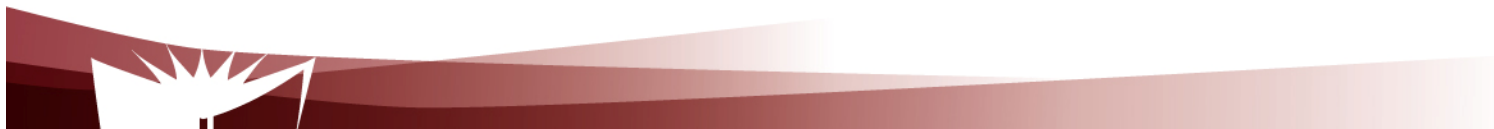
## Technologies



## Technologies

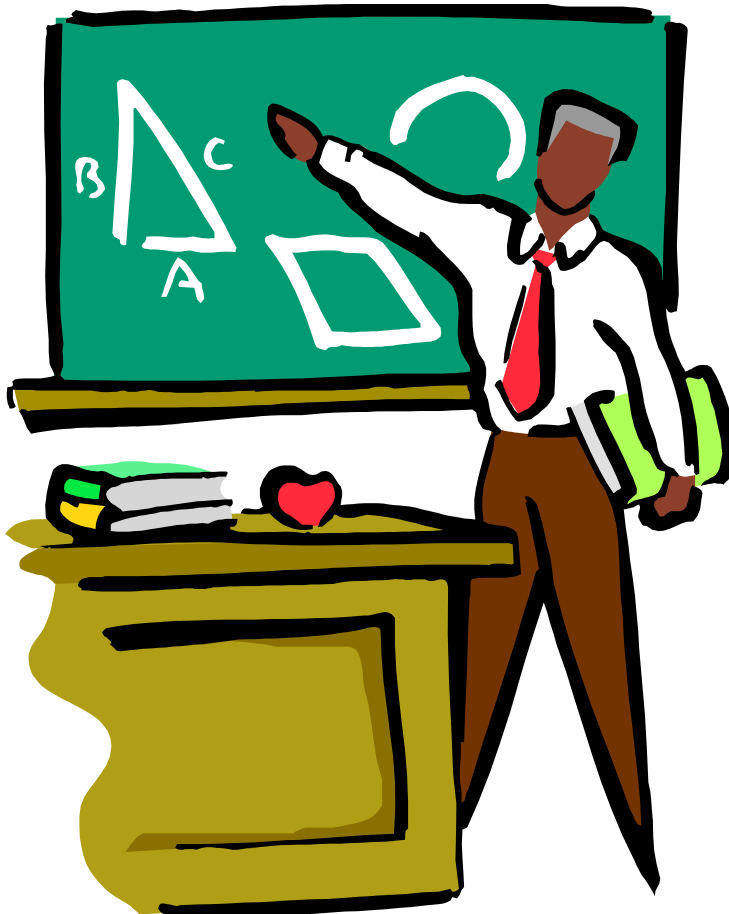


2. XML (already discussed)
3. HTTP (already discussed)
4. WSDL
5. SOAP
6. UDDI



# Telecommunication Services Engineering (TSE) Lab

## SOAP



- Introduction
- Message structure
- Bindings

## Introduction

### SOAP is

- A simple XML based communication protocol between applications
- Platform and language independent

**Purpose: Get the XML data from one point to another point over the network**

- Provider / UDDI
- Requestor / UDDI
- Provider / Requestor

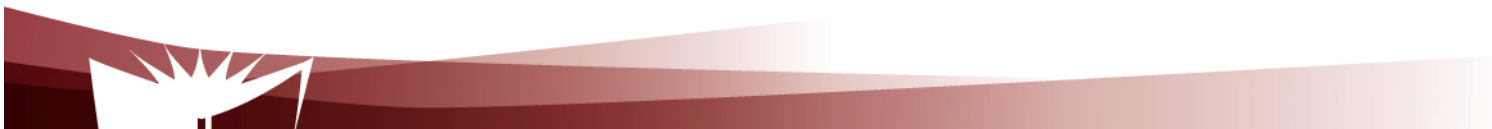
### W3C recommendation

- Effort initiated by IBM and IONA

## Introduction

**Purpose: Get the data from one point to another point over the network**

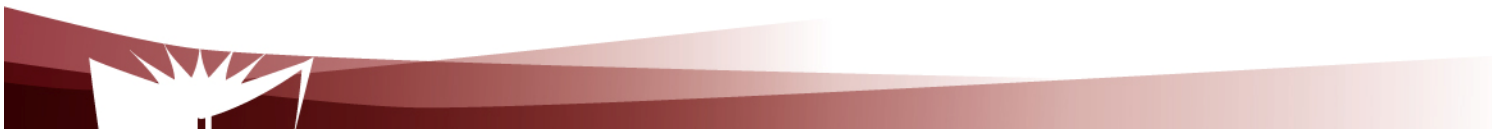
- One way XML messaging protocol that can be used to build models such as
  - Request / reply
  - Asynchronous messaging
  - Event notification
- Entities
  - Sender
  - Receiver
  - Intermediary



## Message structure

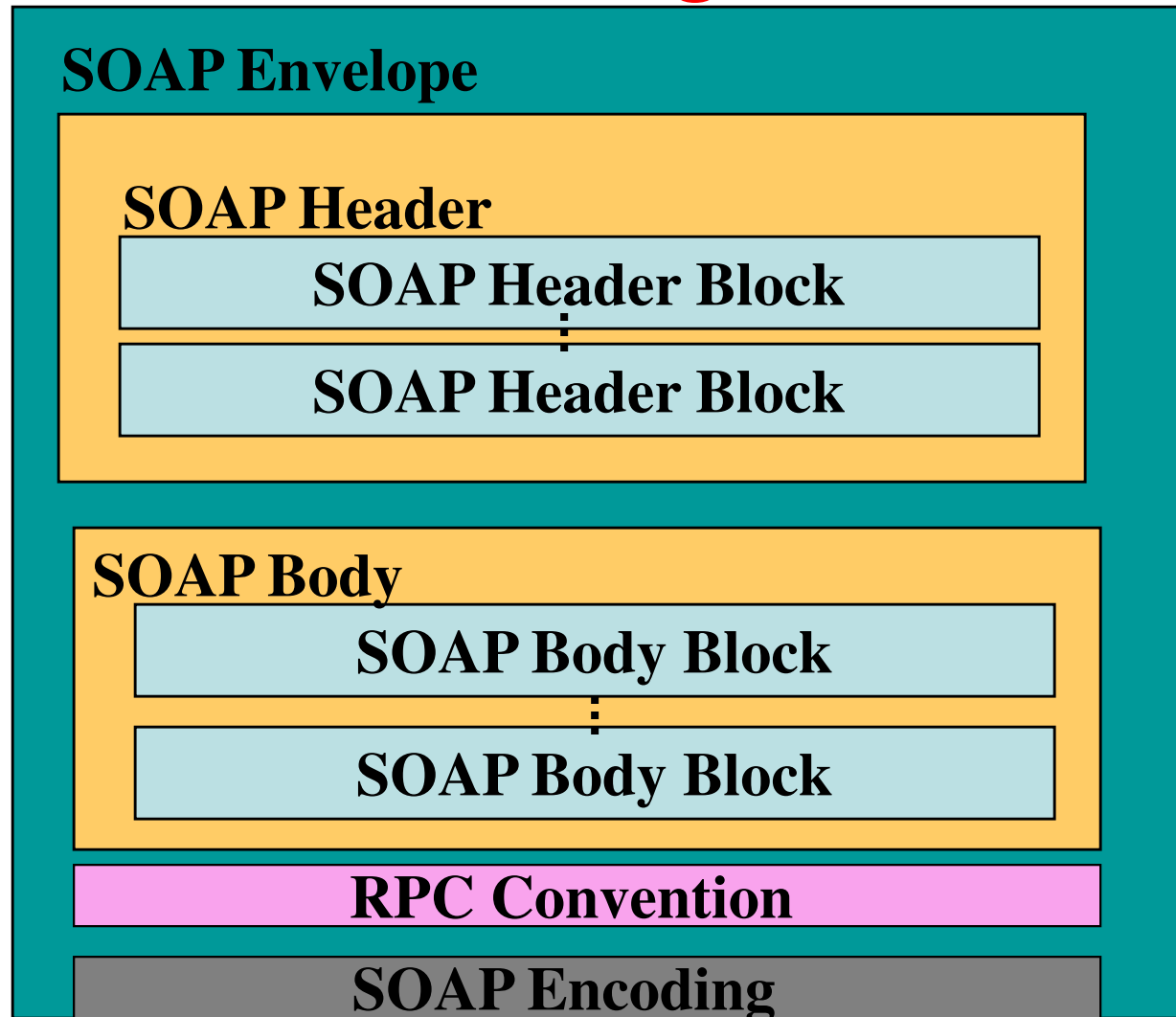
### Several parts

- **Envelope** (mandatory): Start and end of message
- **Header** (optional): Optional attributes used in the processing
  - May be negotiated
  - Examples: transactions, priority, QoS, security
- **Body** (mandatory): Message being sent
  - Actual message
  - Fault codes
- **Attachment** (optional) : Self-explanatory
- **RPC convention** (optional) : Requirements for RPC mapping
  - Target URI for the SOAP node, procedure name/signature
- **SOAP Encoding** (optional) : How to represent data being transmitted in the message
  - Encoding scheme





## Message structure



## Bindings

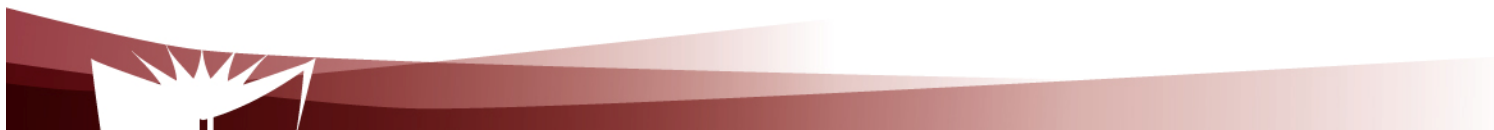
**Purpose: Specification of how SOAP messages may be passed from one node to another node using a concrete lower layer protocol**

Existing bindings

- HTTP
- SOAP over email

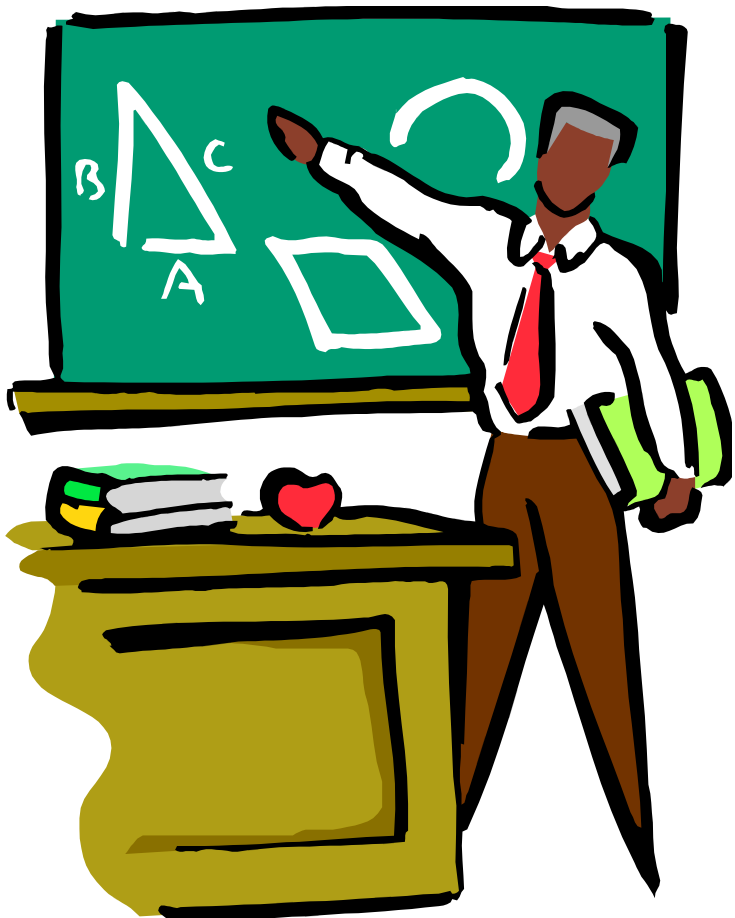
HTTP binding

- HTTP Request URI used to identify SOAP node
- Commonly used HTTP request for carrying SOAP messages: HTTP Post



# Telecommunication Services Engineering (TSE) Lab

**Additional information on SOAP ...**



# Telecommunication Services Engineering (TSE) Lab

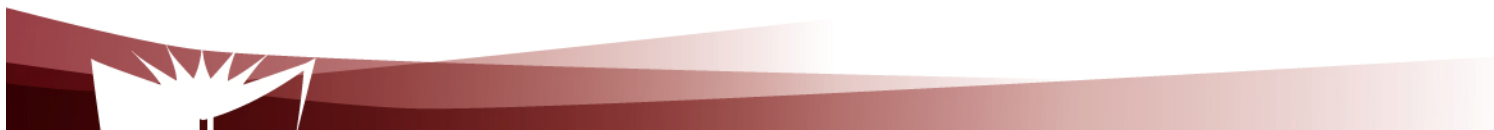
## Reminder: SOAP message sender and receiver ...

### Concepts

- **Sender**
  - Initial sender
  - Intermediary sender
- **Receiver**
  - Intermediary receiver
  - Ultimate receiver

### Nodes

- **Sender**
- **Intermediary (Intermediary sender + intermediary receiver)**
- **Ultimate receiver**



# Telecommunication Services Engineering (TSE) Lab

**Why SOAP? Why not just send XML documents in HTTP  
or via Email ...**

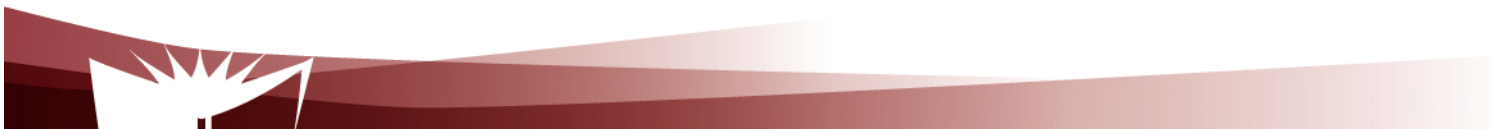
**Give application level control:**

- Priority
- Security
- And other ....

**Via header processing by intermediaries:**

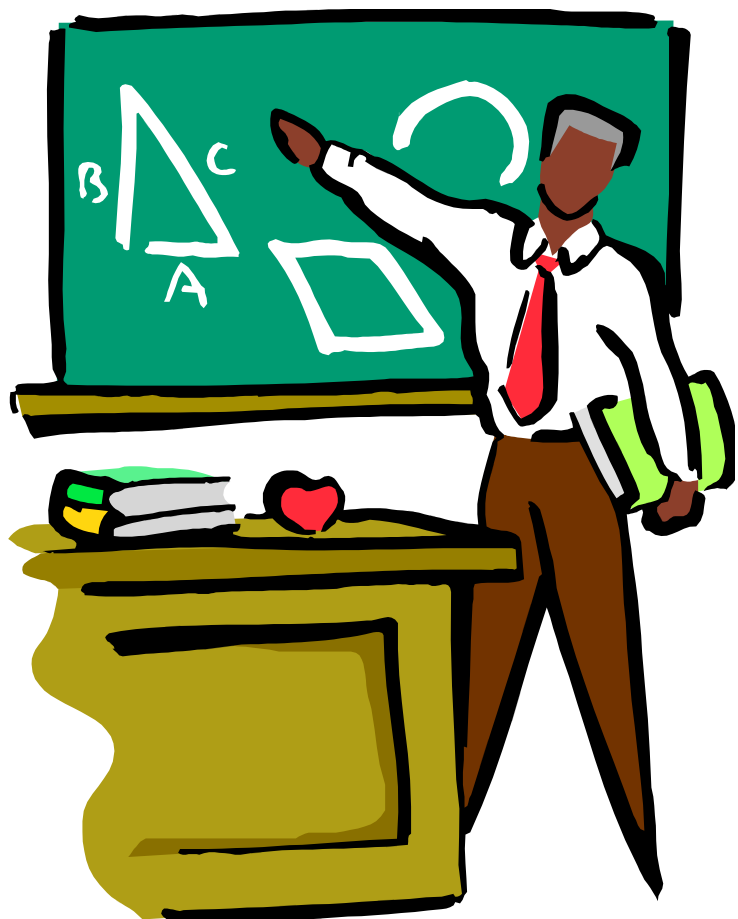
- Take actions according to headers
- Replace headers
- And others ..

**And independently of the binding (e.g. email,  
HTTP):**



# Telecommunication Services Engineering (TSE) Lab

WSDL



# Telecommunication Services Engineering (TSE) Lab

## WSDL

WSDL is an XML-based language for describing Web services and how to access them

**Purpose: XML grammar for describing a Web service**

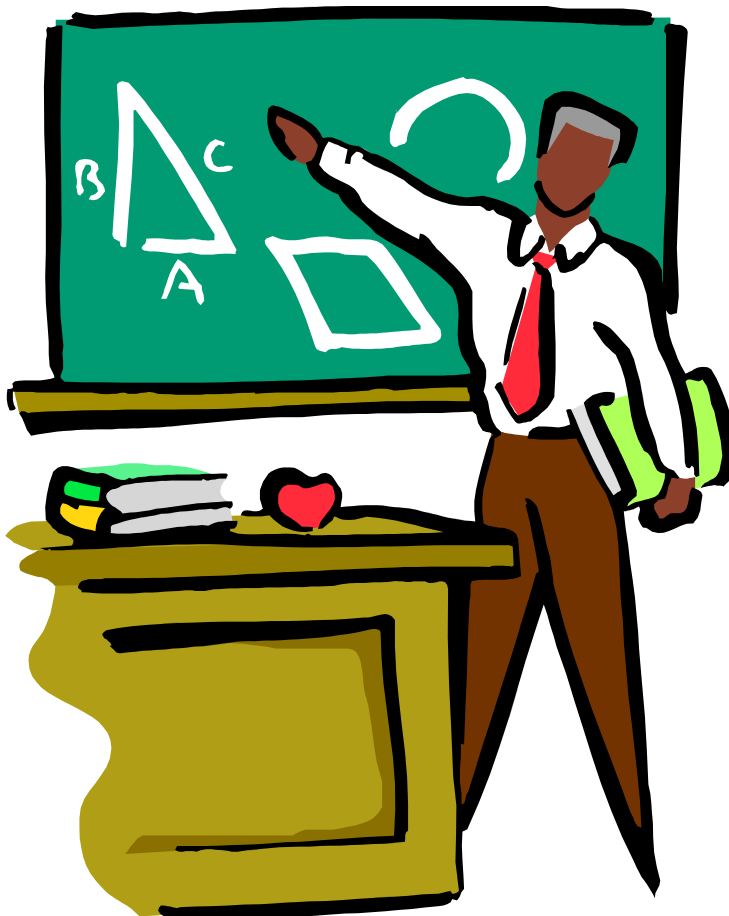
- Formats and protocols
  - Input data to the Web service
  - Operations to be performed on the data
  - Binding to a transport protocol

**Initially developed by a handful of companies (e.g. IBM, Microsoft)**

**Now a W3C recommendation**

# Telecommunication Services Engineering (TSE) Lab

## UDDI ...



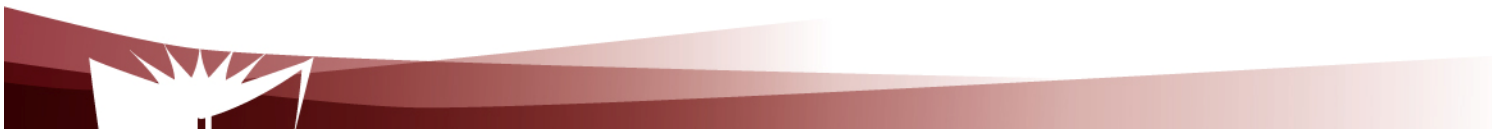
- Introduction
- Content
- Publishing APIs
- Inquiry APIs



## Introduction

**Purpose: Enable the publication, the discovery and the usage of Web services**

- Integral part of the Web services infrastructure
  - Public
  - Semi-public (e.g. circle of trust)
  - Private (e.g. enterprise)
- Data bases accessible via SOAP APIs
  - Publishing API
  - Inquiry APIs



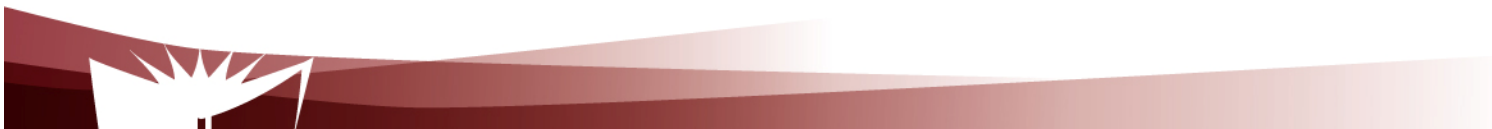
## Introduction

### UDDI.ORG

- Initiated by a a handful of companies (e.g. IBM, Microsoft)
- Now open to all companies
- Produce specifications for UDDI

### Initial public UDDI repository

- Operated by founders of UDDI.ORG, later joined by HP and SAP
- Synchronized data bases called operator sites (one at each site)
- Test UDDI
  - Allow requestors and providers to test their UDDI clients
- Production UDDI
  - Allow providers to actually publish Web Services and requestors to actually inquire about Web services
    - Need to register with one of the operators for publishing services (authorization)



## The content ...

### White pages

Business address

Contact person / number

### Yellow pages

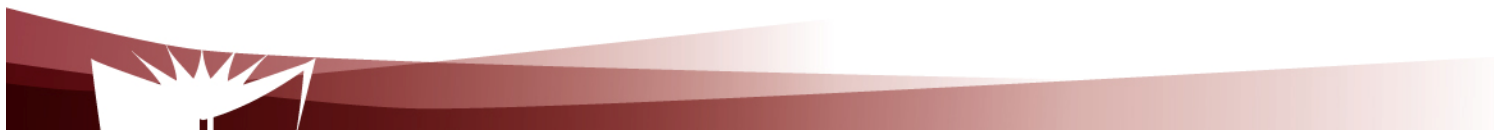
More info about the business

- Type of business
- Industry type
- Products / services

### Green pages

Technical information about the services

- Service features/functionality
- Pointer to the WSDL file



## The content ...

### UDDI data model

#### Business entity

- Top level structure
- Description of the entity for which information is being registered
- Include the list of Web services provided by the entity

#### Business services

- Name and description of services being published
- Include binding templates

#### Binding templates

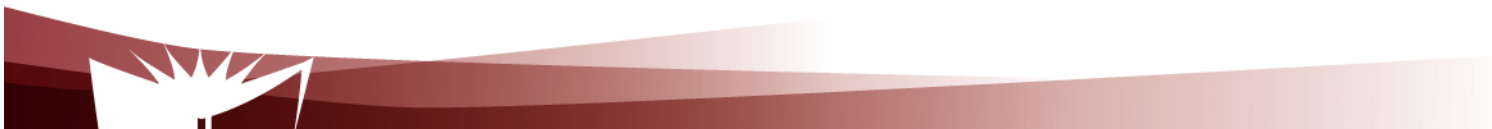
- Information about the services
- Include entry point for accessing the services

#### tModel

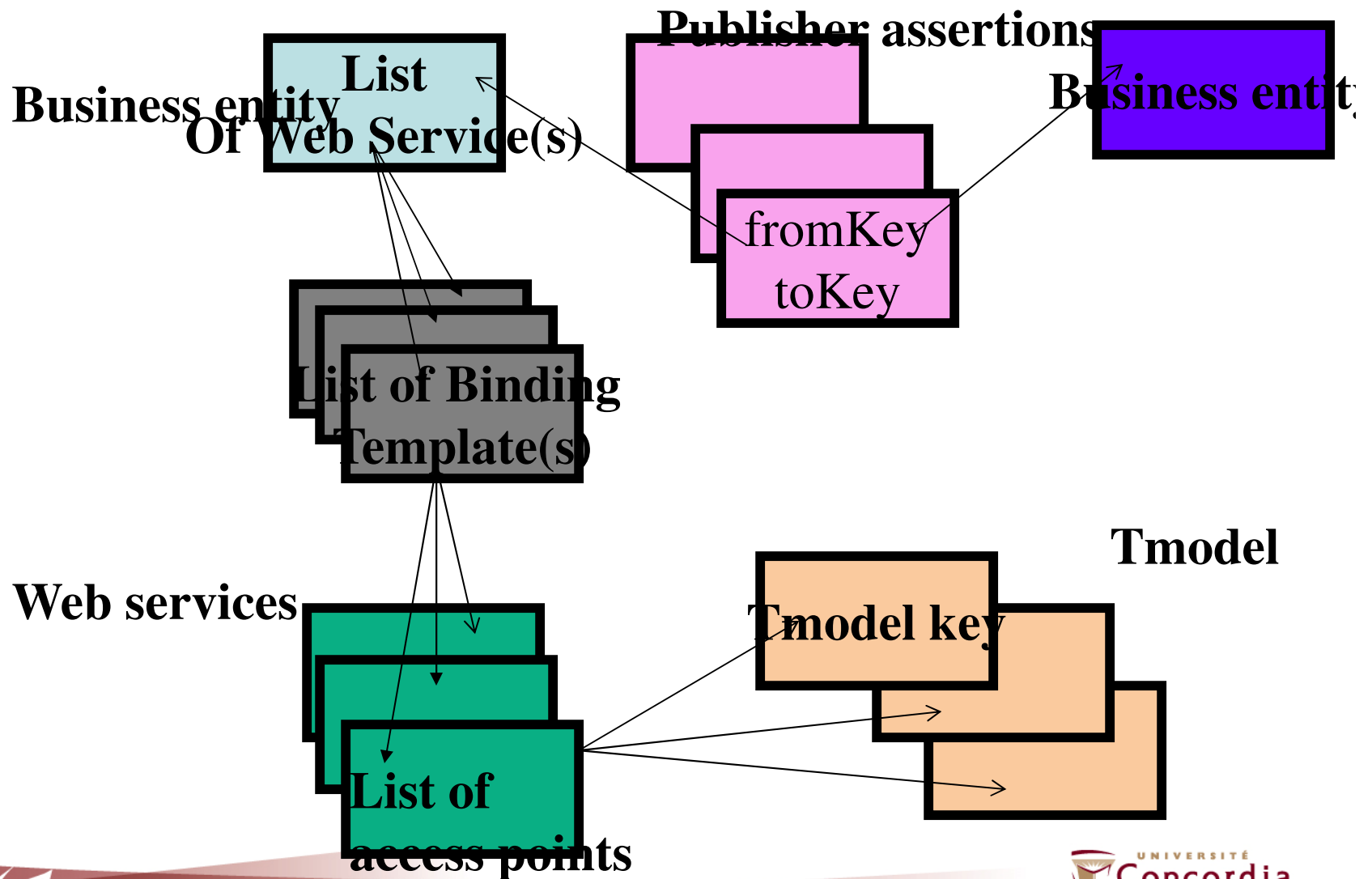
- Fingerprint, collection of information that uniquely identify the service

#### Publisher assertion

- Business relationship between business entities (e.g. subsidiary of ..)



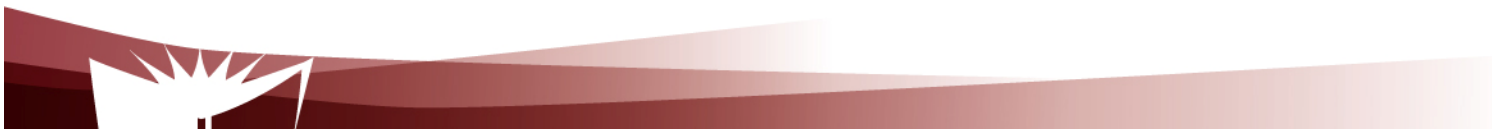
## Data model ...



## Publishing APIs

### Some examples

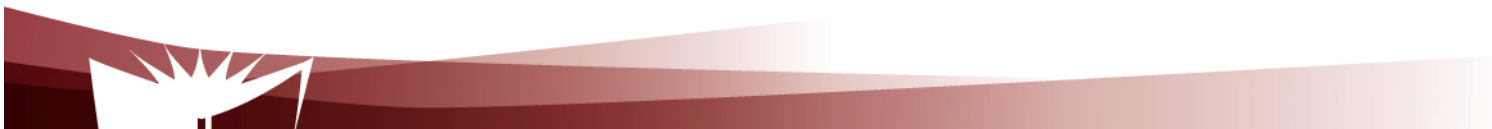
- Add Publisher Assertions
- Save/Delete binding
- Save/Delete Business
- Save/Delete Service
- Save/Delete tModel
- Set/Get Publisher assertions
- Delete\_Publisher\_Assertion
- Get Registered assertions
- Get Assertions status report (used by UDDI operators)



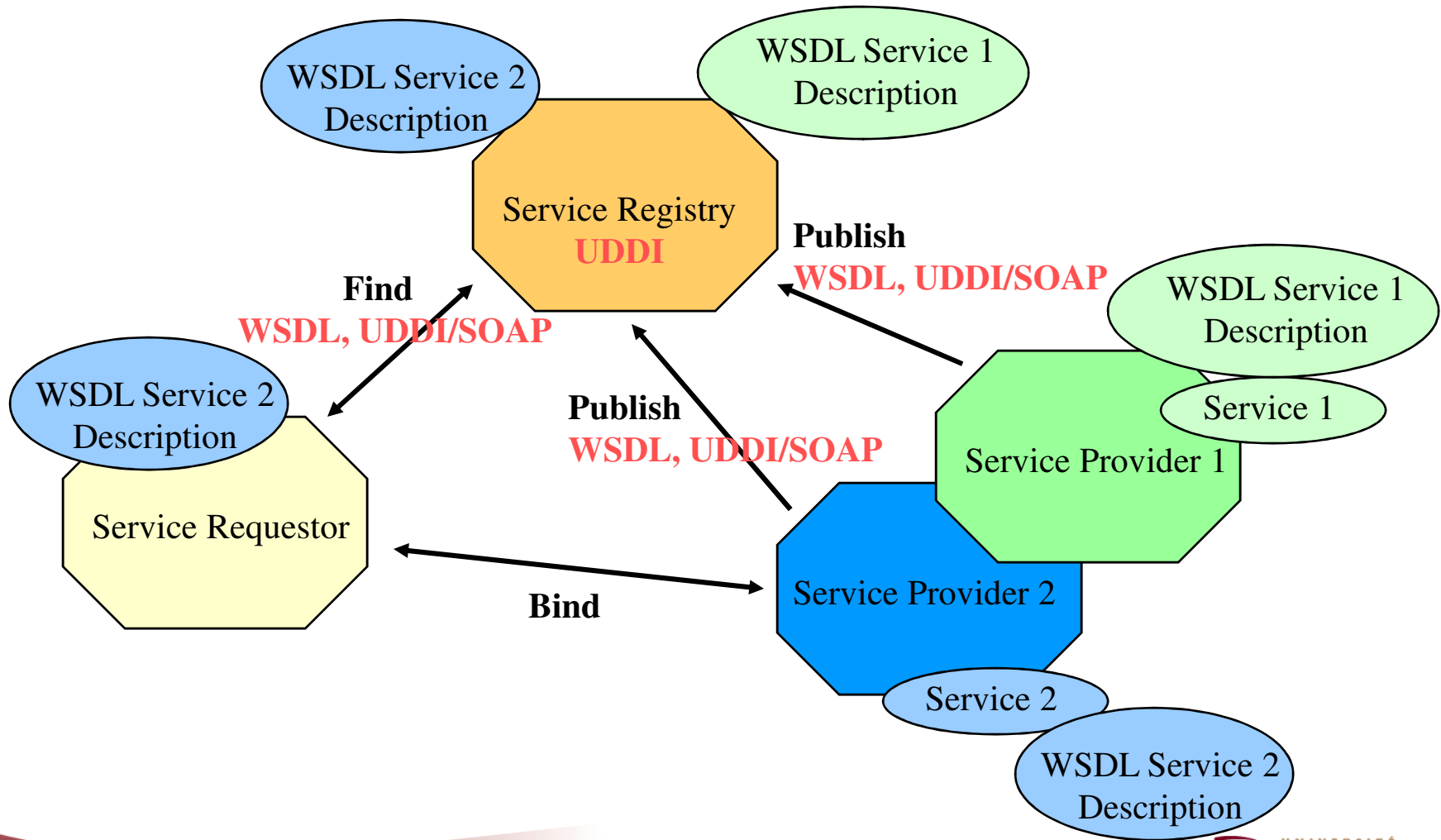
## Inquiry APIs

### Some examples

- Find binding
- Find business
- Find related business
- Find service
- Find tModel
- Get binding details
- Get business details
- Get tModel details



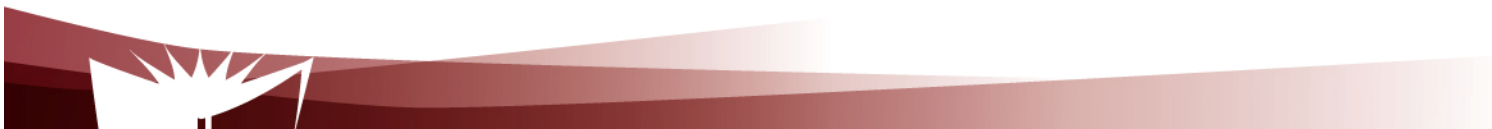
## Putting it together ...





## Examples of tool kits

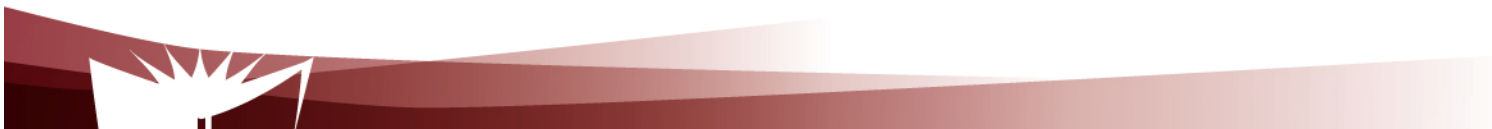
- Examples of tool kits
  - Apache / Axis
  - BEA Weblogic
  - SunOne
  - .Net
  - Systinet
  - Get tModel details
  
- Usage simplicity depends on:
  - Friendly user interface
  - Detail level required



# Telecommunication Services Engineering (TSE) Lab

## To probe further ...

- F. Curbera et al., **Unraveling the Web services Web: An Introduction to SOAP, WSDL and UDDI**, IEEE Internet Computing, Vol. 6, No2, March-April 2002, pp. 86-93
- E. Newcomer, **Understanding Web Services: XML, WSDL, and UDDI**, Addison Wesley, 2002
- W3C specifications
- OASIS specifications (UDDI)
- <http://www.projectliberty.org/>
- <http://www.bea.com/>

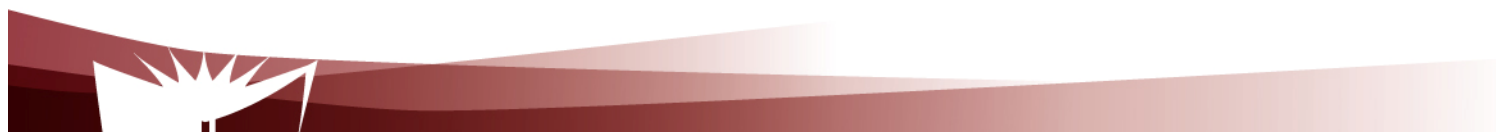


# Telecommunication Services Engineering (TSE) Lab

## Applying Web services to value added service engineering in NGN

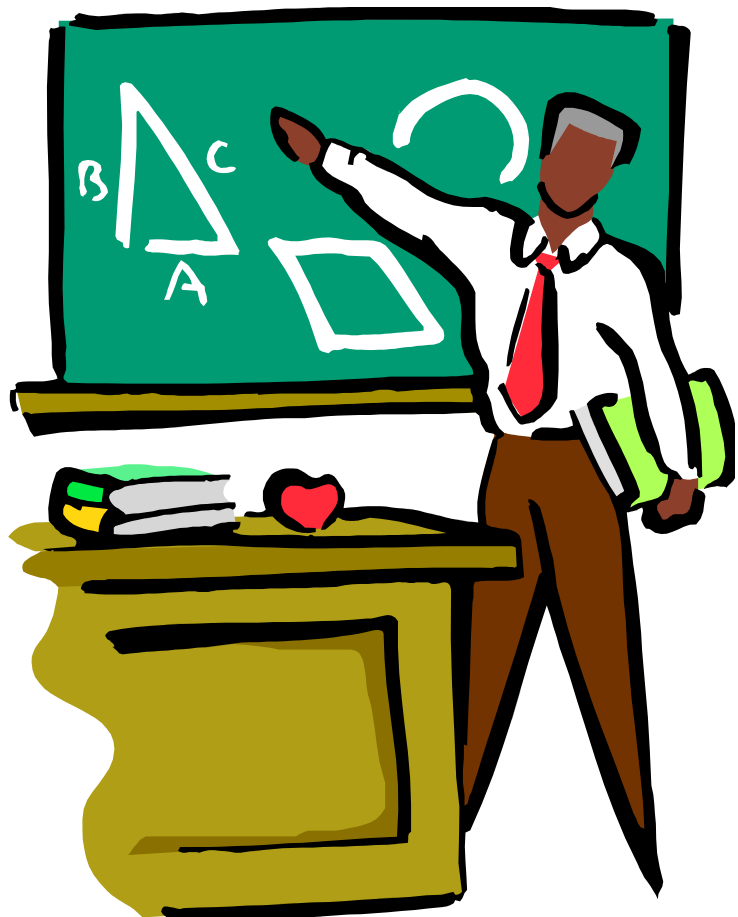


1. Parlay-X
2. OMA Deployment patterns



# Telecommunication Services Engineering (TSE) Lab

## Parlay-X ...

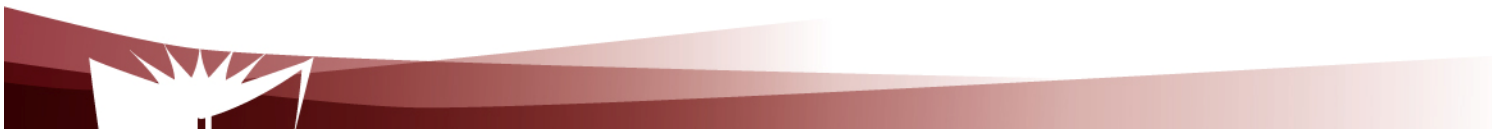


1. Introduction
2. Architecture
3. The services

## Introduction

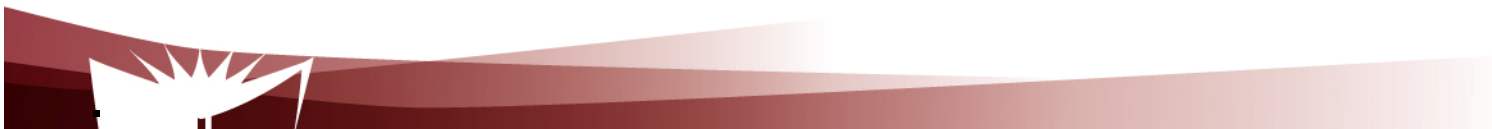
### Application interfaces

- Aim at covering all telecommunication capabilities
  - Stand alone capabilities (e.g. presence, call control)
  - Combined capabilities (presence + call control)



## The standardized services

1. Call control
2. Messaging
  - SMS
  - MMS
3. Payment (e.g. volume charging)
4. Account management (e.g. account credit expiration date query)
5. User status (online / offline)
6. Terminal location



## Parlay-X Call Control ...

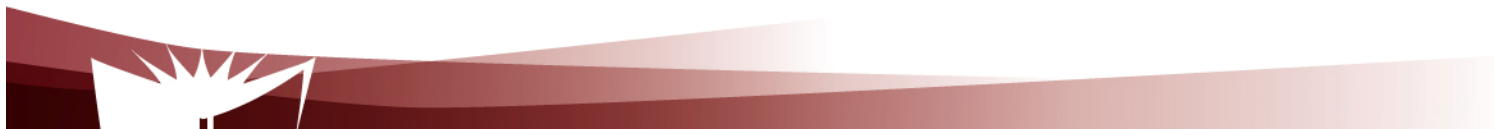
Make a call

Get call information

End call

Cancel call request

.



## Parlay-X Call Control ...

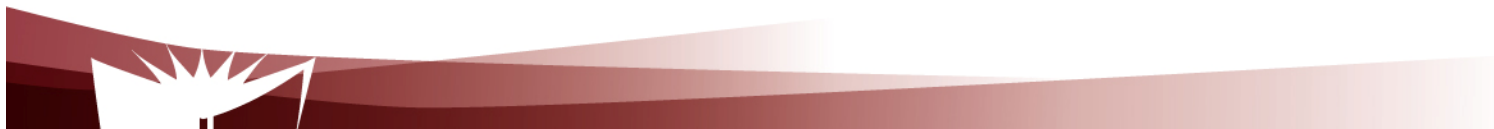
Handle busy

Handle Not reachable

Handle No answer

Handle off Hook

.



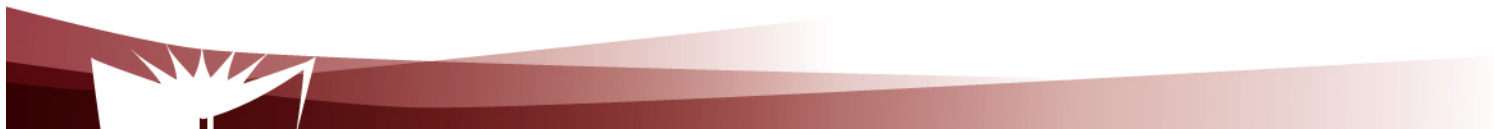


# Parlay-X Conferencing Basics...

Allow the creation of a multimedia conference call and the dynamic management of:

- Conference
- Participants
- Media

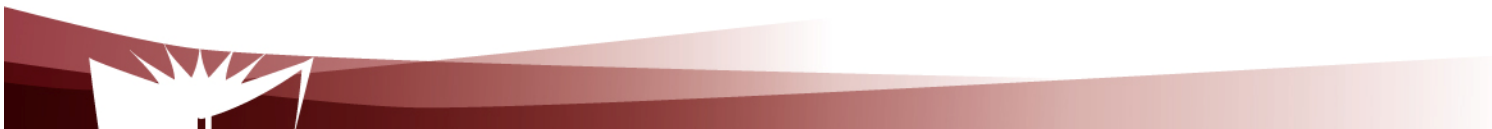
.



# Parlay-X Conferencing Basics...

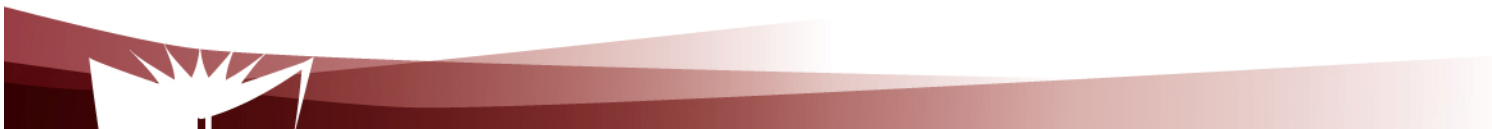
### Service model entities

- Conference
  - “Context / virtual room” to which participants can be added
- Participants
  - Parties involved in the conference
- Media
  - audio/video/chat



# Parlay-X Conferencing Basics...

- Conference
  - “Context / virtual room” to which participants can be added
- Participants
  - Parties involved in the conference
- Media
  - audio/video/chat



# Parlay-X Conferencing Basics...

Create conference

- Create a multimedia conference with initially no participant

GetConference Info

- Information on status (e.g. active, terminated)

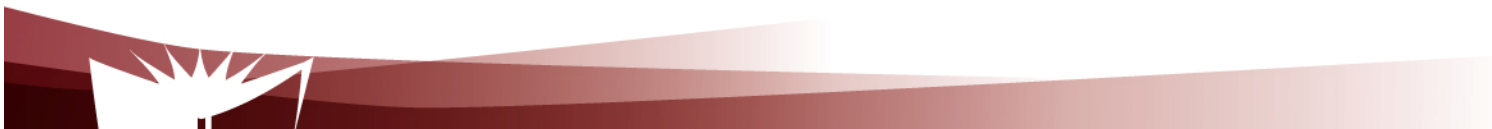
EndConference

Several possibilities

Maximum duration has expired

All participants have left

▪



# Parlay-X Conferencing Basics...

inviteParticipant

- Add a new participant to the conference

disconnectParticipant

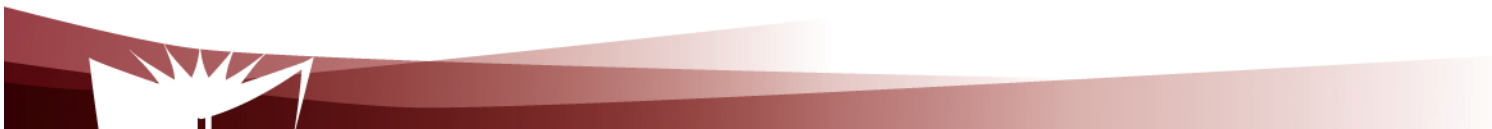
- Disconnects the participant

addMediaForParticipant

Executed on a single participant

- Add a media stream to the media set used by participant

.

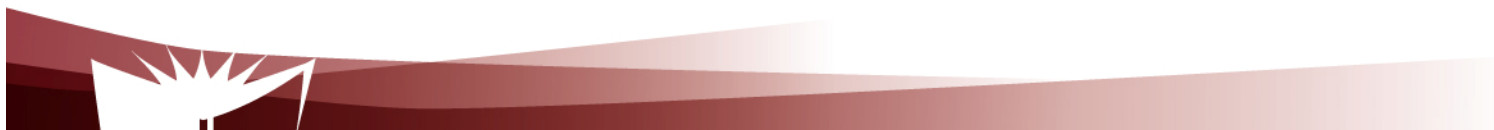


# Parlay-X Conferencing Basics...

deleteMediaForParticipant

disconnectParticipant

getParticipantInfo



## Parlay-X MMS ...

Send Message

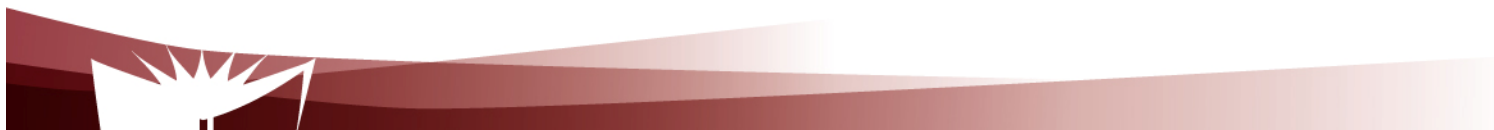
Get Message Delivery Status

Get Received messages

Get messages URIs

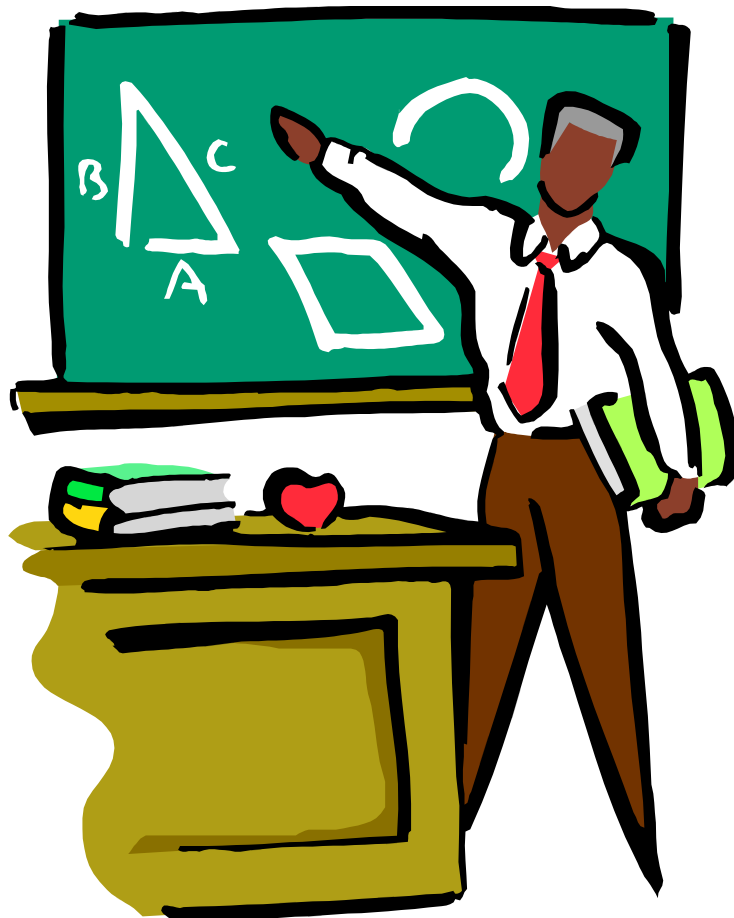
Notify message reception

.



# Telecommunication Services Engineering (TSE) Lab

OMA ...



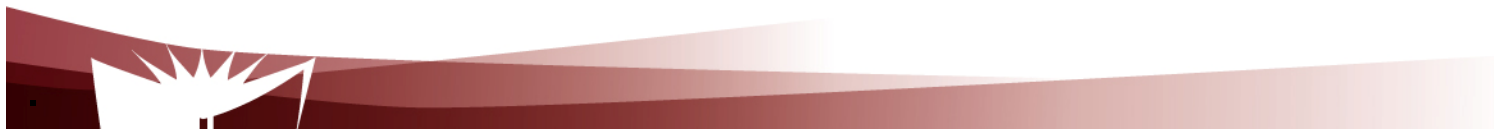
Examples of deployment patterns



## Deployment patterns

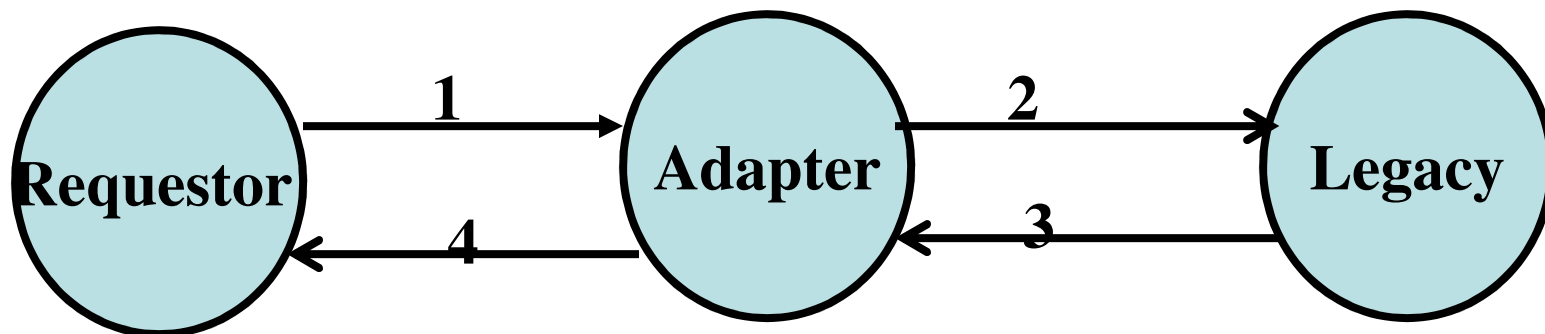
### OMA

- Industry association created in 2002
- Focus on mobile services
- Aims at:
  - Consolidating standards for wireless services (e.g. 3GPP/PP2, IETF, W3C)
  - Producing new standards if needed-
  - Tackling the two issues



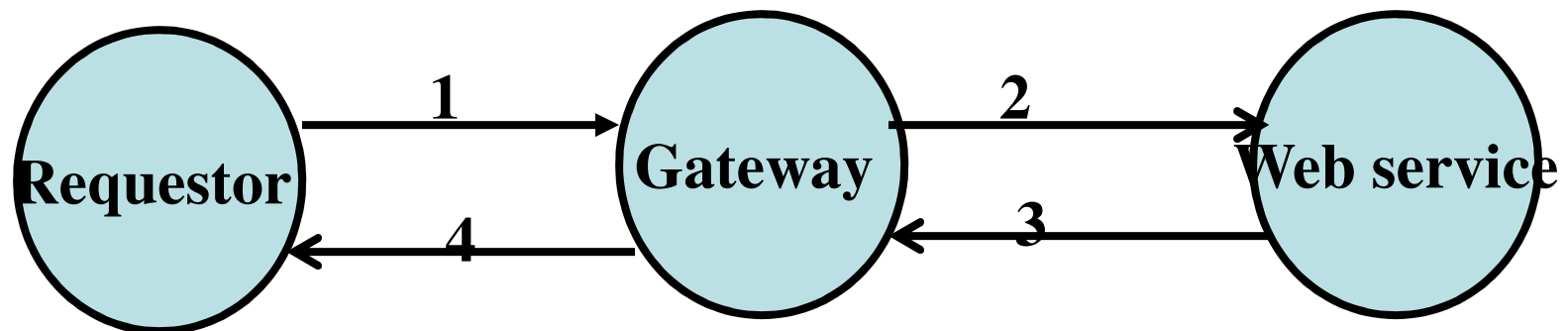
# Examples of deployment patterns

## The adapter pattern



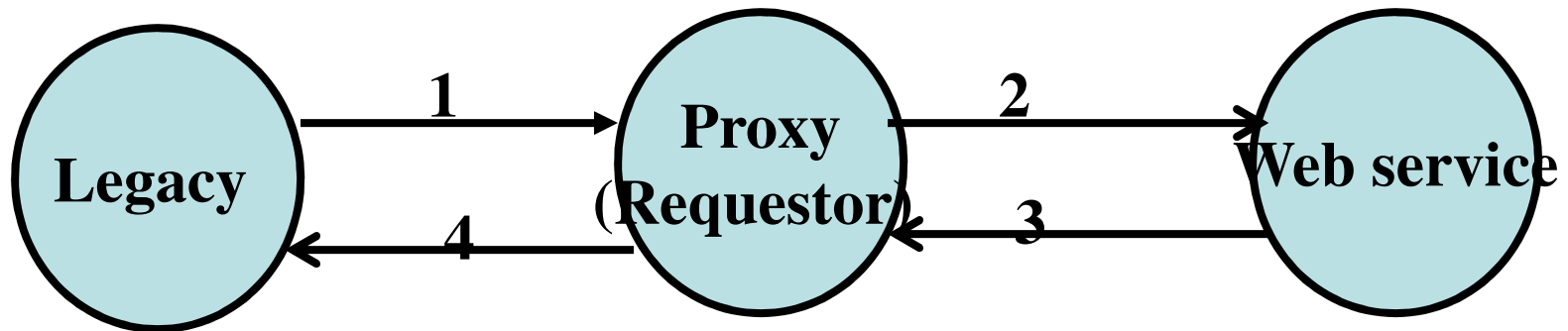
## Examples of deployment patterns

### The gateway pattern



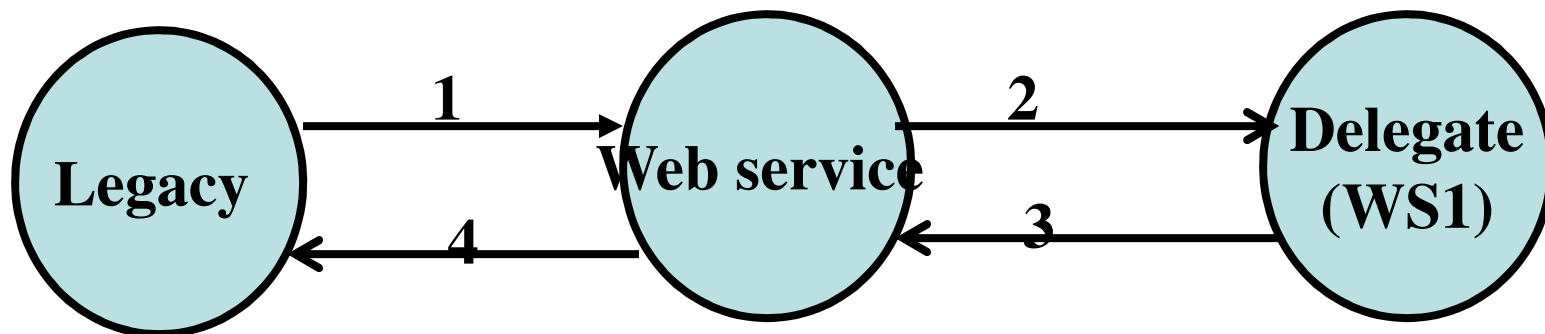
## Examples of deployment patterns

### The proxy pattern



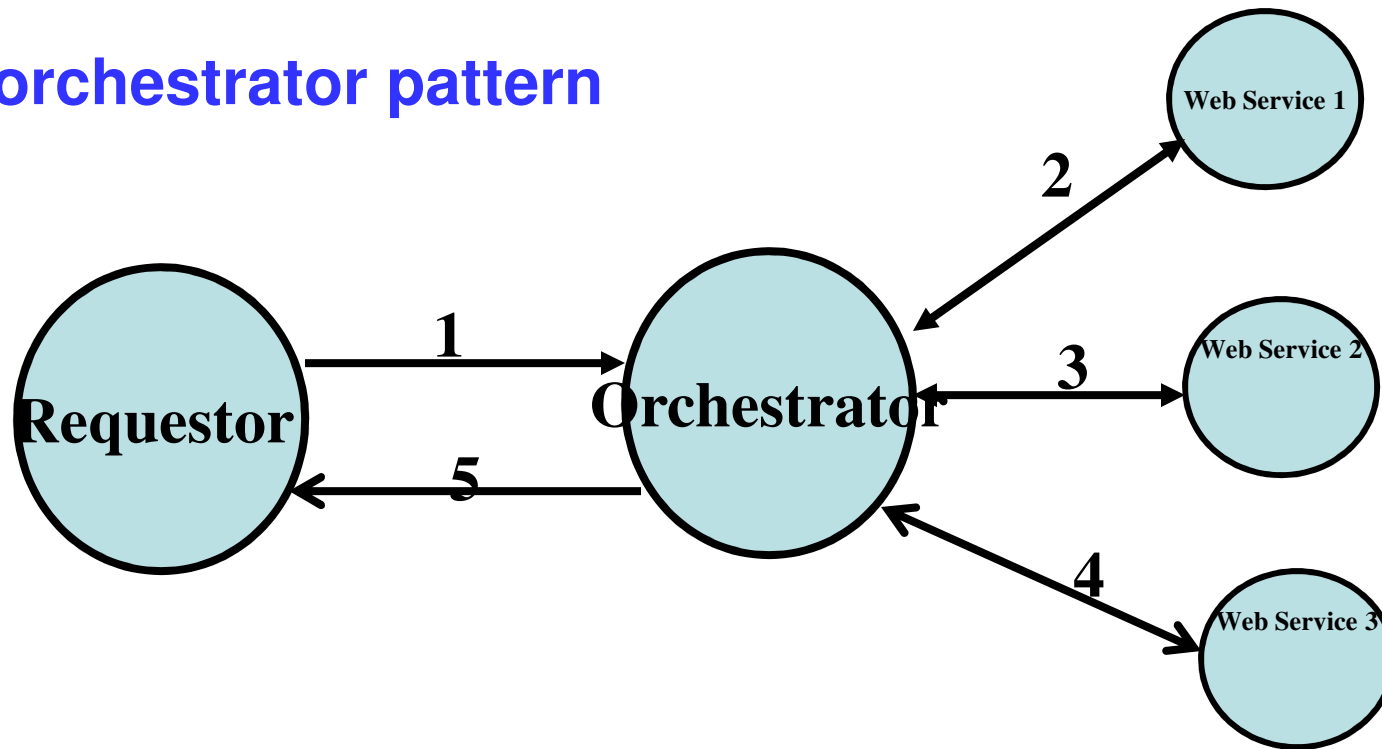
## Examples of deployment patterns

### The delegate pattern



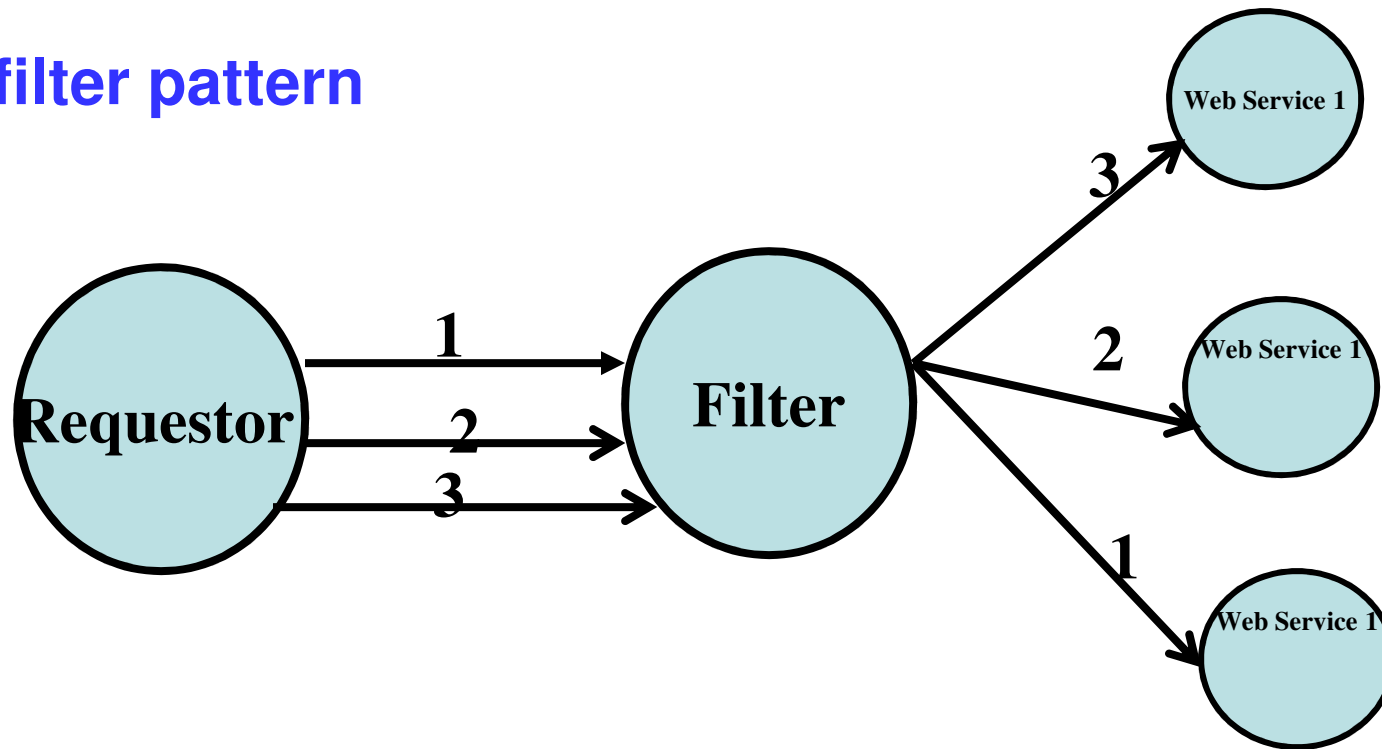
## Examples of deployment patterns

### The orchestrator pattern



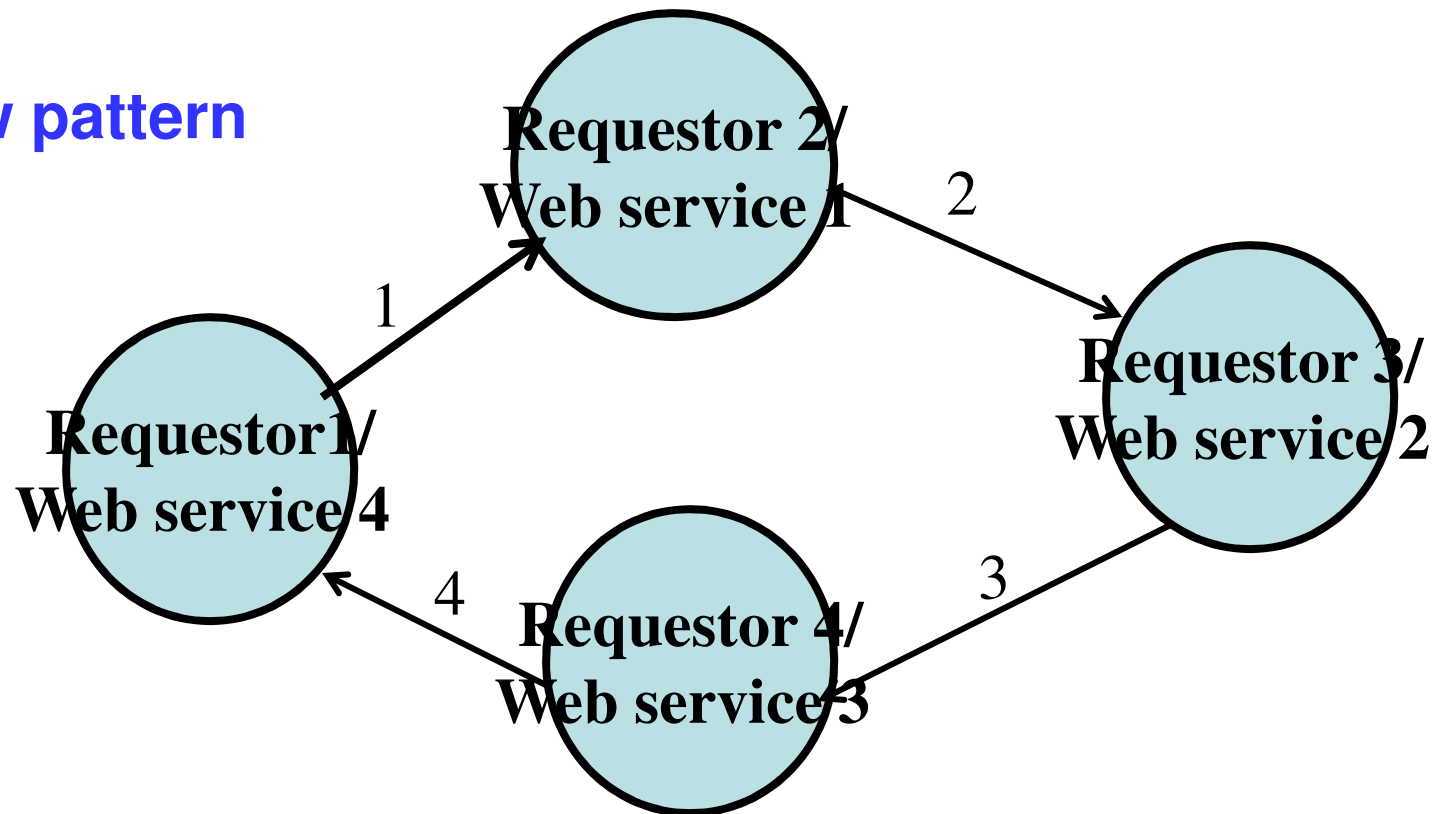
## Examples of deployment patterns

### The filter pattern



## Examples of deployment patterns

### The workflow pattern





# Telecommunication Services Engineering (TSE) Lab

## To probe further ...

- F. Curbera et al., **Unraveling the Web services Web: An Introduction to SOAP, WSDL and UDDI**, IEEE Internet Computing, Vol. 6, No2, March-April 2002, pp. 86-93
- E. Newcomer, **Understanding Web Services: XML, WSDL, and UDDI**, Addison Wesley, 2002

