# Enterprise computing

# Web Application Architectural styles

## 6.what is meant by the term "Service Oriented Architecture"?

Service oriented architecture is a collection of services. These services communicate with each other. The communication can involve either data passing or it could involve two or more services coordinating some activity. There are some concepts like different transport porotocol and using different message services involve.

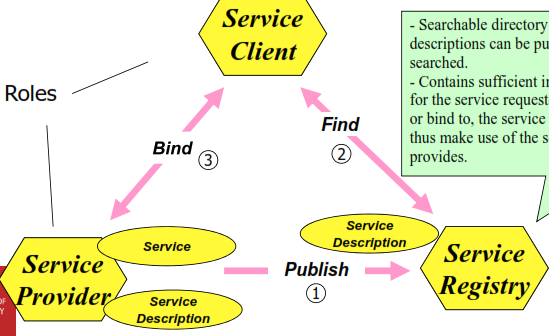
An SOA is based on

-the interaction between decoupled services and/or end-user applications

-associated with messages & governed by policies.

-services distributed in a network, via published and discoverable

interfaces.



## 7. what is differences between web services and SOA?f

A Web service is a software system *identified* by a URL,

whose public interfaces and bindings are *defined* and

*described* using XML.

Its definition can be *discovered* by other software

systems

These systems may then interact with the Web service

using XML based messages conveyed by Internet

protocols (not necessarily Web protocols!)

but .. diffinition of soa

## 8.what is the difference between stateful and stateless web services? Give an example of each

Services that can be invoked repeatedly without having to maintain context

or state are called *stateless*.

Simple informational services are stateless.

Stateless means that service does not record information of the user.-------

Services that require their context to be preserved from one invocation to

the next are called *stateful*.

Complex services (business processes) typically involve stateful interactions.

Statefull means that service does remember information of the user(by key or composite key or something like that)

## 9.explain how RPC(remote procedure call) and message styles of interaction differ?

1.RPC is synchronous

In RPC:

Object/component-oriented approach to distributed computing e.g. CORBA

Application elements use a request/wait-for-reply (**synchronous**) model of

communication.

RPC-style programming leads to *tight coupling* of interfaces and applications.

In an RPC environment each application needs to know the intimate details of the *interface* of every other application – the number of methods it exposes & the details of each method signature it exposes.

In message oriented service:

Most asynchronous messaging mechanisms follow the “fire-and-forget” messaging principle where the sending application can conduct its work as usual once a message was asynchronously sent.

The sending application assumes that the message will arrive safely at its

destination at some point in time.

This mode messaging does not preclude the necessity to perform

request/reply operations.

MOM is an infrastructure that involves the passing of data between applications using a common communication channel that carries self-contained messages.

Messages are sent and received asynchronously. The messaging system (integration broker) is responsible for managing the connection points between clients & for managing multiple channels of communication between the connection points.

## 10.explain what is meant by the term choreographic language?

Choreography is associated with the public (globally visible)message exchanges, rules of interaction & agreements that occur between multiple business process endpoints. It tracks the sequence of messages that may involve multiple parties & multiple sources, & described from the *perspectives of all parties*(common view).

Not necessarily activity-centric

Usually in any application we can come up with two structre:

1.Centralised orchestration bpel

2. decentralised choreography (application send message )

## 11.List 4 advantages of using middleware?

* MOM is an infrastructure that involves the passing of data betweenapplications using a common communication channel that carries self-contained messages.
* --Messages are sent and received asynchronously. The messaging system (integration broker) is responsible for managing the connection points between clients & for managing multiple channels of communication between the connection points.
* --Middleware functions in enterprise computing. As well assupporting distributed computing, middleware need to support

Inter-operability between ***heterogeneous***

systems

data formats

interaction styles

Management and monitoring

Identity and security

Business process enactment

Governance and policy

## 12.List 4 functions of message-oriented middlewar?

Second part of the q11 answer

## 13.what is an enterprise integration pattern?what type of architectural styles uses these patterns?give an example of such a pattern

Design patterns for the use of enterprise application integration

Enterprise Integration Patterns are implemented in many open

source ESB's

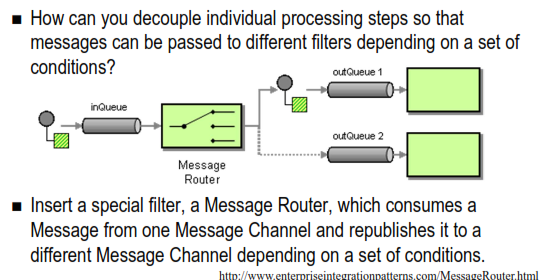
Spring Integration, Apache ActiveMQ/ Camel, ...

From book by Gregor Hohpe and Bobby Woolf

describing 65 design patterns

* Categories:
* Integration Styles
* Messaging Systems
* Message Construction
* Message Routing
* Message Transformation
* Messaging Endpoints
* System Management

Example of pattern:



## 14.list 2 advantages claimed for restful approach to web services as opposed to Ws\* approach?(also you can look at next question)

1. Standard simple operations, standard error codes *looser* ***interface*** *coupling (still need to know resourceURI)*

*Generic interface* 

1. Multiple representations Ability to support client preferences /

disparate clients and *looser* ***data*** *coupling*

1. No complex standards – HTTP well understood
2. Can be implemented with any server-side technology
3. Because REST is stateless (servers do not have to remember

state):

highly **scalable** suitable load balancing / cloud

**fast** caching: HTTP gets can be cached in reverse proxy reducing load on origin server, clients can also have their own caches avoid call to

6.Late binding / linked services – client does not need to know all

links in advance. Resource links can be changed on the fly

server

loose **interface** and **address** coupling!!

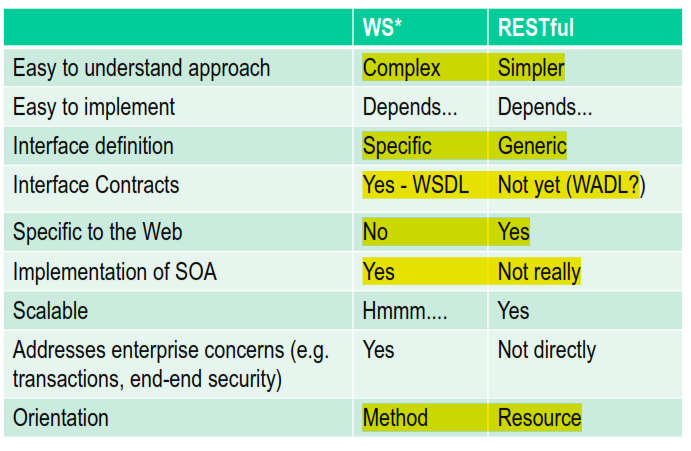
7Good for processes where clients drive the process decisions

(e.g. human’s involved in decision making)

loose **behavioural** (**temporal**) coupling!!

## 15.List 2 advantages claimed for a WS\* approach to web services as opposed to a restful approach?

1. it is not just specific to web
2. that is implementation of SOA



## 16.explain 4 main HTTP verbs are used in restful approach?

HTTP ‘verbs’ have specific meaning uniform CRUD interface for keeping

resources updated:

**GET** – retrieve information on a resource

**POST** – Create a new resource

**PUT** – Modify an existing resource

**DELETE** – Delete an existing resource

Also use **HEAD** (get meta-data), **OPTIONS** (check which verbs supported)

## 17.why are restful / resource-based architecture claimed to be more scalable that ws\* / middleware-based architectures. Do you agree?

Yes, Because rest is stateless and using url more elaboration as below:

Because REST is stateless (servers do not have to remember

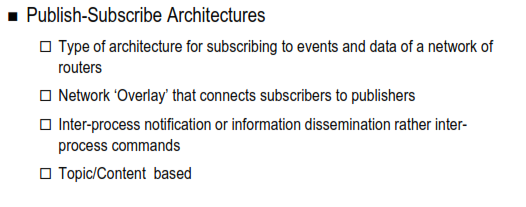
state):

highly **scalable** suitable load balancing / cloud

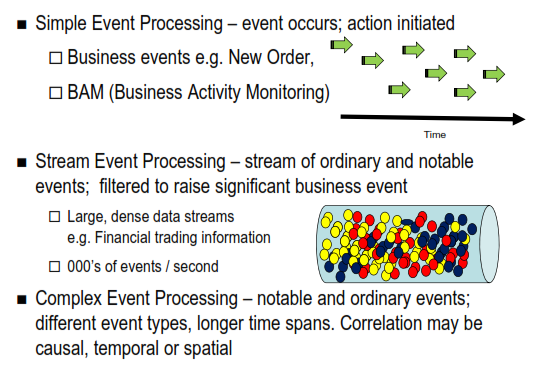
**fast** caching: HTTP gets can be cached in reverse proxy reducing load

on origin server, clients can also have their own caches avoid call to server

## 18. what is pub-sub overlay network? Describe the process by which subscribe receives information

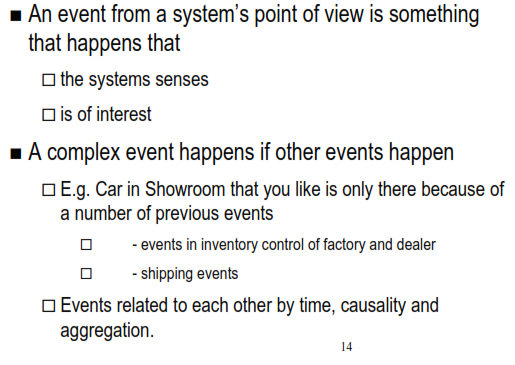


## 19. Briefly describe the 3 styles of event processing in EDA?(lec06)



## 20.what is complex event(lec06)?

Complex event is an event that is inferred from relationships between other events. More elaboration as below:



## 21.What is JMS? What are the two communication models it supports?(lec05)

The Java Message Service (JMS) API is a MOM API for sending messages between two or more clients.

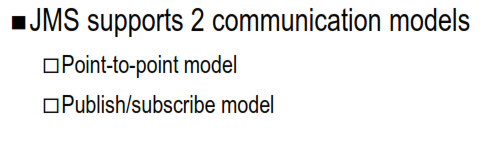
Allows application components based on Java EE to create, send, receive, and read messages.

loosely coupled

asynchronous

**JMS**

reliable



## 22.what are the key ESB capabilities?