










Software Design and Architecture

Engr. Abdul-Rahman Mahmood

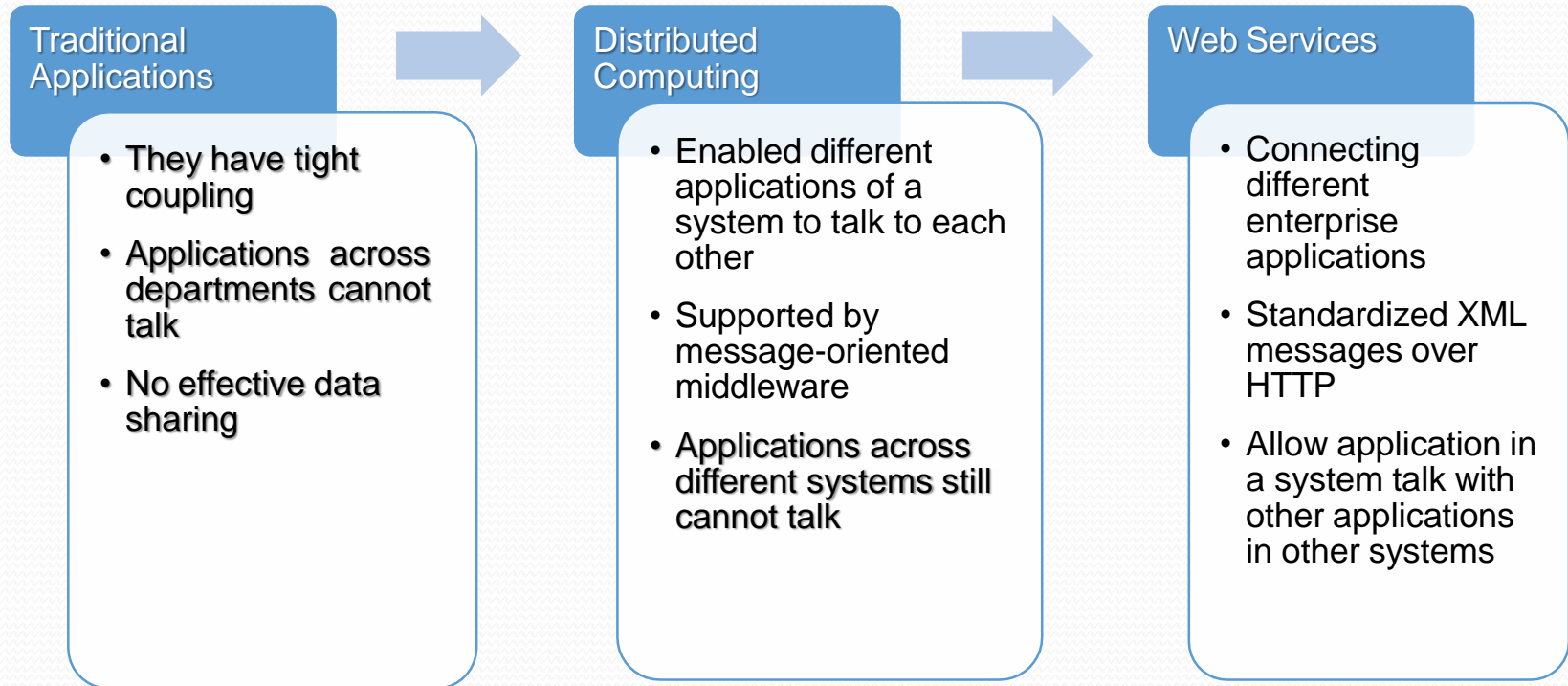
DPM, MCP, QMR(ISO9001:2000)

 armahmood786@yahoo.com
 alphapeeler.sf.net/pubkeys/pkey.htm
 pk.linkedin.com/in/armahmood
 www.twitter.com/alphapeeler
 www.facebook.com/alphapeeler
 abdulmahmood-sss  alphasecure
 armahmood786@hotmail.com
 <http://alphapeeler.sf.net/me>

 alphasecure@gmail.com
 <http://alphapeeler.sourceforge.net>
 <http://alphapeeler.tumblr.com>
 armahmood786@jabber.org
 alphapeeler@aim.com
 mahmood_cubix  48660186
 alphapeeler@icloud.com
 <http://alphapeeler.sf.net/acms/>

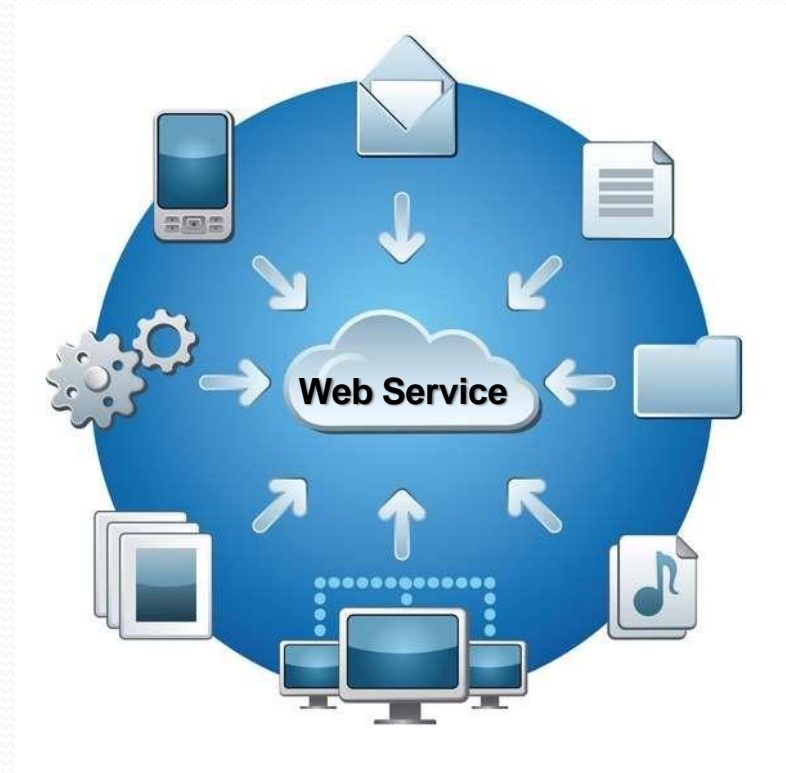
Web Services SOAP & RESTful

Paradigm Shift: Moving to SOA with Web Services

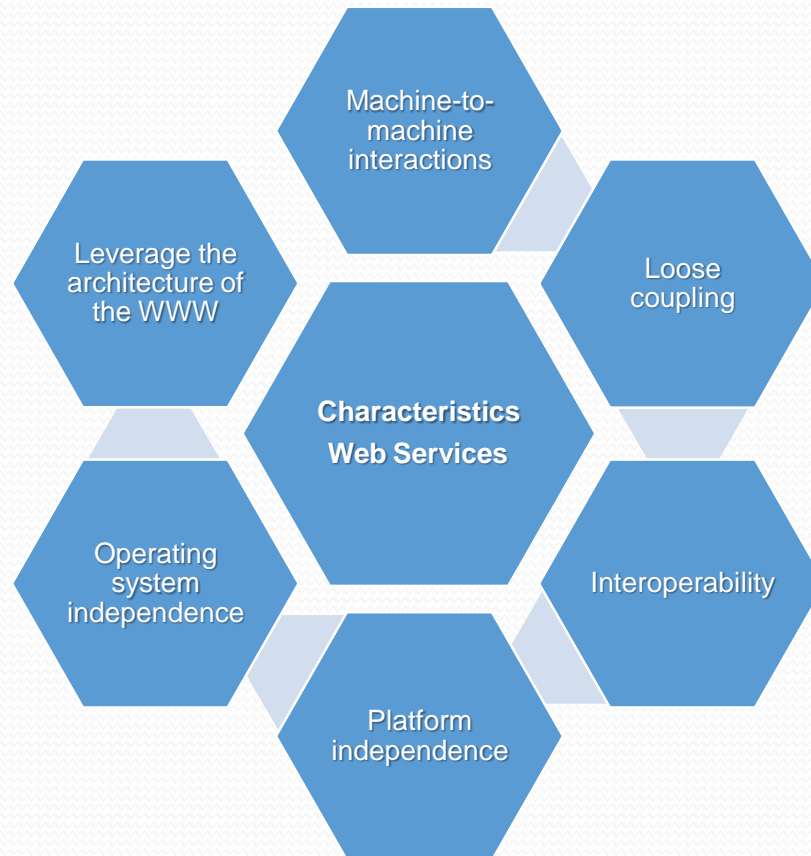


What is a Web Service?

- A **software system** designed to support interoperable **machine-to-machine** interaction over a **network**.
 - Self-contained, modular, distributed, dynamic
 - Standardized messaging system
 - Can be described, published, located and/or invoked over the network.
 - Language-agnostic
 - Vendor and transport neutral



Characteristics of Web Services

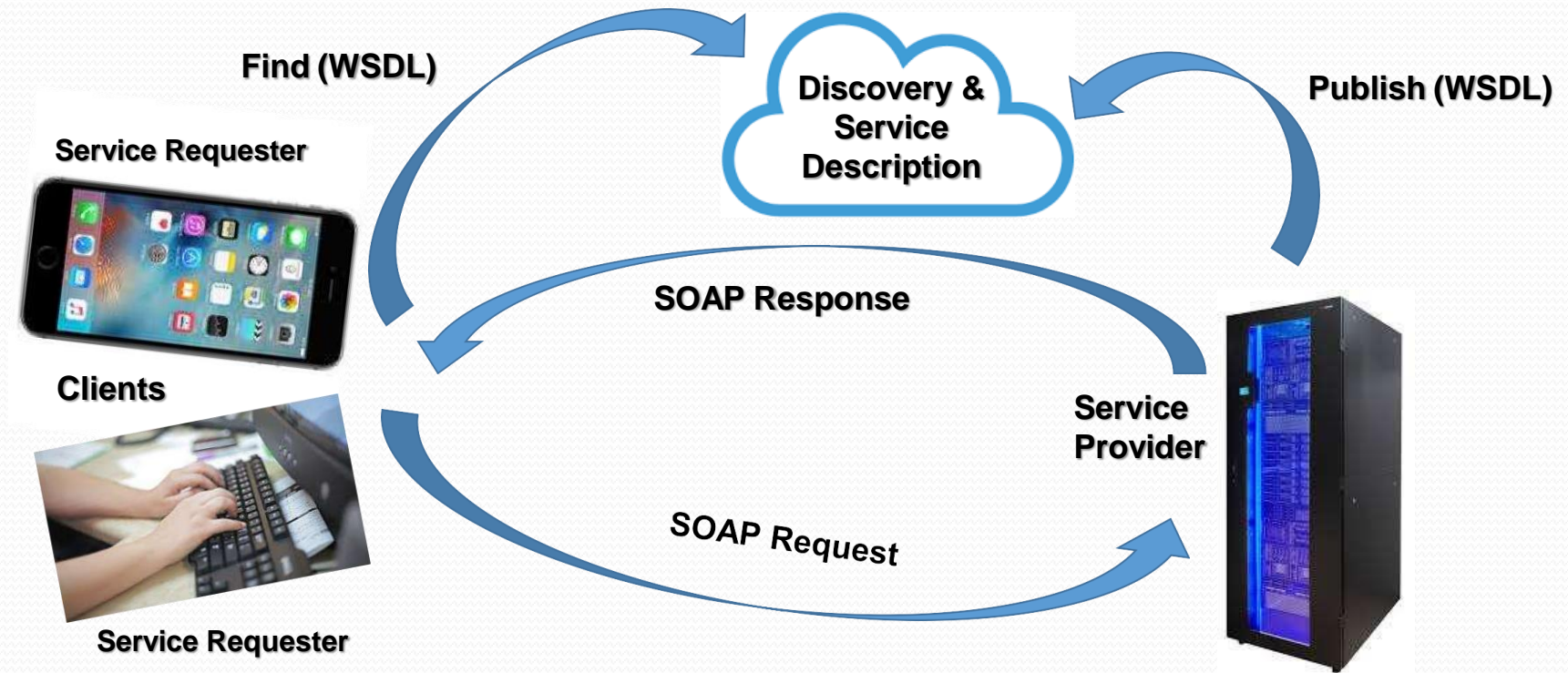


SOAP Web Services

SOAP (**S**imple **O**bject **A**ccess **P**rotocol)

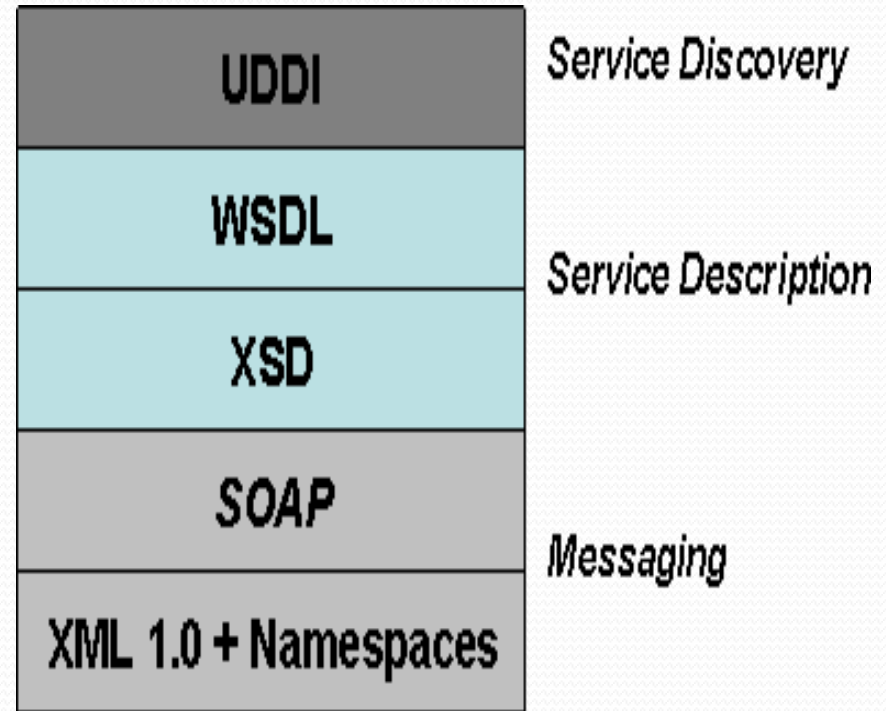
- SOAP is an XML-based messaging protocol for exchanging information among computers. SOAP is an application of the XML specification.
- W3C specification for XML distributed computing infrastructure.
- Enables client applications to connect to remote services and invoke remote methods
- Driven by standard specifications
 - Basic: UDDI, WSDL, SOAP, XML & Namespaces
 - Extended: WS-Security, WS-Policy, WS-I (Interoperability) etc.

SOAP - Architectural Overview



SOAP: Basic Specifications

- **UDDI** (Universal Description, Discovery, and Integration): Platform-independent way of describing and discovering Web services and Web service providers.
- **WSDL**: Defines services as collections of network endpoints.
- **SOAP**: Simple and lightweight mechanism for exchanging structured and typed information.
- XML + Namespaces



Pros &

Cons

SOAP – Pros & Cons

- SOAP Pros

- Standard protocol for exchanging information in a decentralized and distributed environment.
- Platform independent & Vendor neutral.
- Simple compared to RMI, CORBA, and DCOM etc
- Decouples the encoding and communications protocol.
- Anything that can generate XML can communicate through SOAP.
- Additional Security in addition to HTTP authentication or HTTPS.
- Supported by most languages and tools.

- SOAP Cons

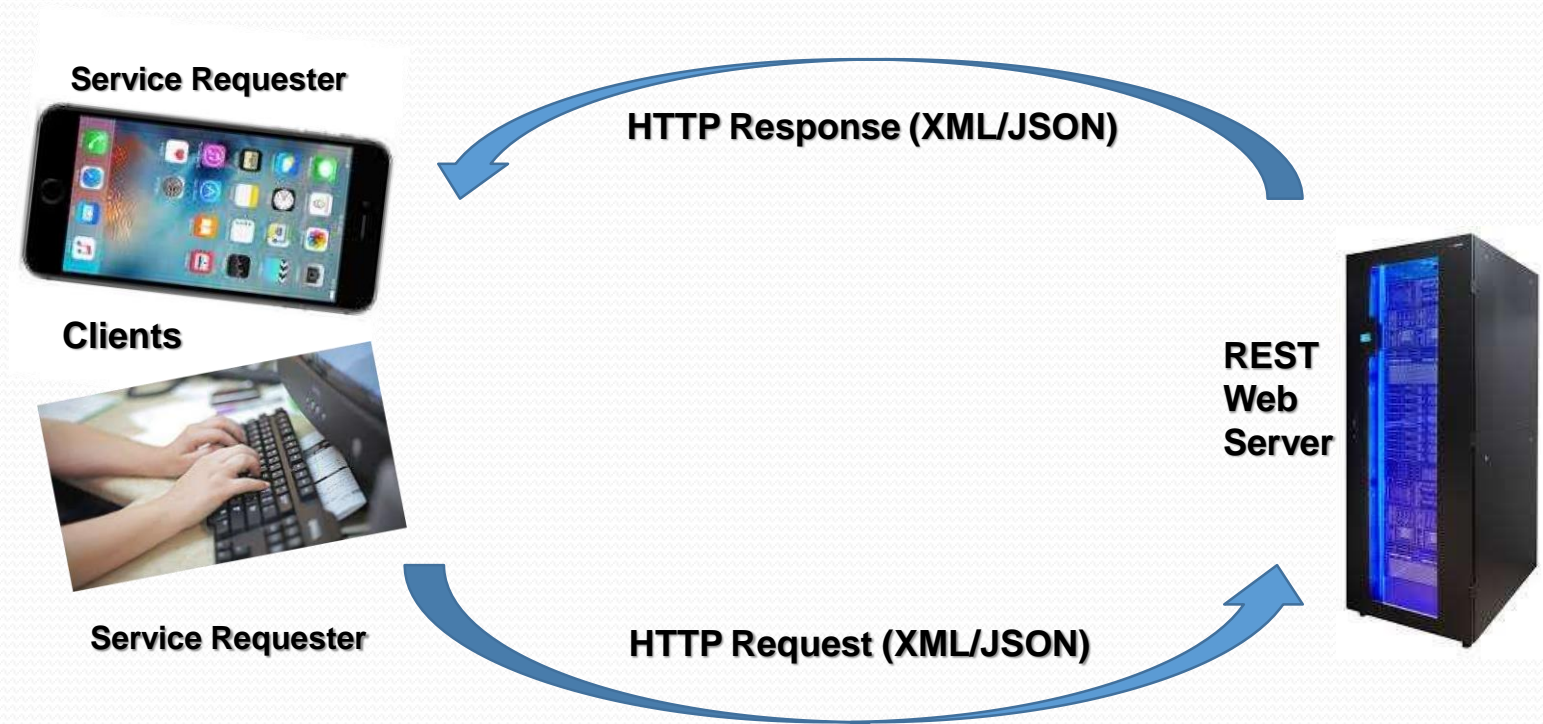
- Complex compared to RESTful Services
- Higher learning curve

RESTful Web Services

RESTful Web Service

- **R**epresentational **S**tate **T**ransfer
- REST is an architecture all about the Client-Server communication
- Guided by REST constraints (design rules).
- Based on Resource Oriented Architecture
 - > A network of web pages where the client progresses through an application by selecting links
- Uses simple HTTP protocol and service methods:
 - > GET: Return data, nothing is changed on server
 - > POST: Create, update, or delete data on server
 - > PUT: Replace referenced resource(s)
 - > DELETE: Delete referenced resource(s)

RESTful – Architectural Overview



Specifications (Constraints)

Client-Server

- Separation of concerns - user interface vs data storage
- Client and server are independent from each other

Stateless

- Each request from client to server must contain all of the information
- No client session data or any context stored on the server

Cacheable

- Specify data as cacheable or non cacheable
- HTTP responses must be cacheable by the clients

Uniform Interface

- All resources are accessed with a generic interface (HTTP-based) which remains same for all clients.

Layered System

- Allows an architecture to be composed of hierarchical layers
- Each component cannot “see” beyond the immediate layer.

Code On-Demand

- REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts.

Comparing: SOAP vs RESTful

SOAP	RESTful
XML based Messaging Protocol	REST is an architectural style
Uses WSDL for communication between Consumer and Provider	Uses XML or JSON to send or receive data
SOAP is Service Oriented – Invokes services by calling RPC methods	REST is Resource Oriented - uses (generally) URI and methods like (GET, PUT, POST, DELETE) to expose resources
SOAP supports for stateful implementation	REST follows stateless model
Transfer is over HTTP as well as other protocols such as SMTP, FTP, etc	REST is over only HTTP
SOAP is Distributed Computing style implementation	REST is Web Style (Client Server) Implementation
SOAP can be called from JavaScript but difficult to implement.	Easy to call from JavaScript.

Which to choose, SOAP or RESTful?

- Choosing between SOAP and RESTful – Architectural Decision, not just a matter of simplicity or performance.

When to use SOAP	When to use RESTful
<ul style="list-style-type: none">• Complex applications• Stateful operations• Asynchronous processing and invocation• Distributed applications in peer relationship• Additional security via WS Security<ul style="list-style-type: none">• Authentication Tokens• XML Encryption• XML Digital Signatures	<ul style="list-style-type: none">• Relatively Simpler Applications• Stateless operations• Limited bandwidth and resources• Caching situations• Client Server mode• Only HTTPS security