Web Services Overview

Sang Shin
www.JPassion.com
"Learn with JPassion!"

Agenda

- What is a Web Service?
- Why Web Services?
- Where are & where are Web Services going?
- Web services over Java platform

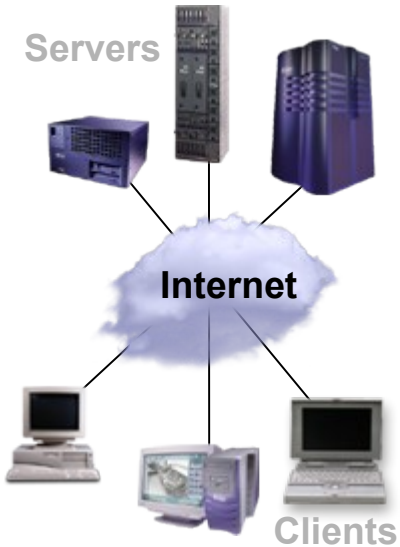
What is a Web Service?

Web Services Definition by W3C

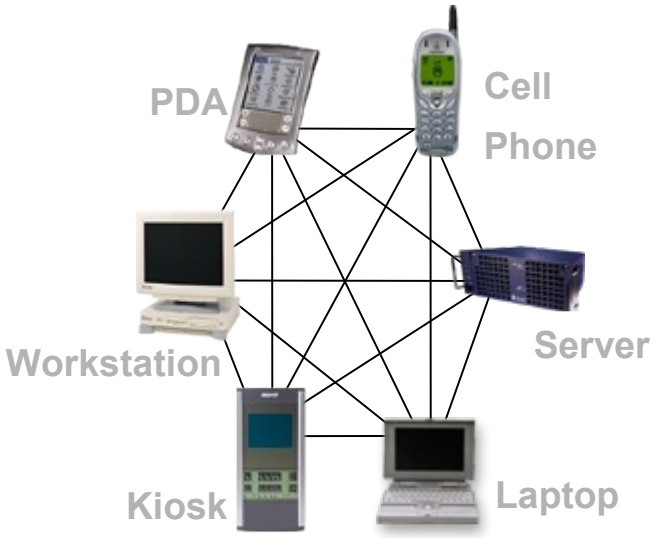
- A Web service is a software application
- identified by a URI,
- whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and
- supports direct interactions with other software applications
- using XML based messages
- via Internet-based protocols

Distributed Computing Evolution

Client-Server(C/S) silos



Web-based computing



Web Services/Peer-to-Peer

Traditional C/S vs. Web Services

Traditional C/S

- Within enterprise
- Tied to a set of programming languages
- Procedural
- Usually bound to a particular transport
- Tightly-coupled
- Efficient processing (space/time)

Web Services

- Between/within Ent.
- Program language independent
- Message-driven
- Easily bound to different transports
- Loosely-coupled
- Relatively not efficient processing

Web Application vs. Web Services

Web Application

- User-to-program interaction
- Static integration of components
- Monolithic service

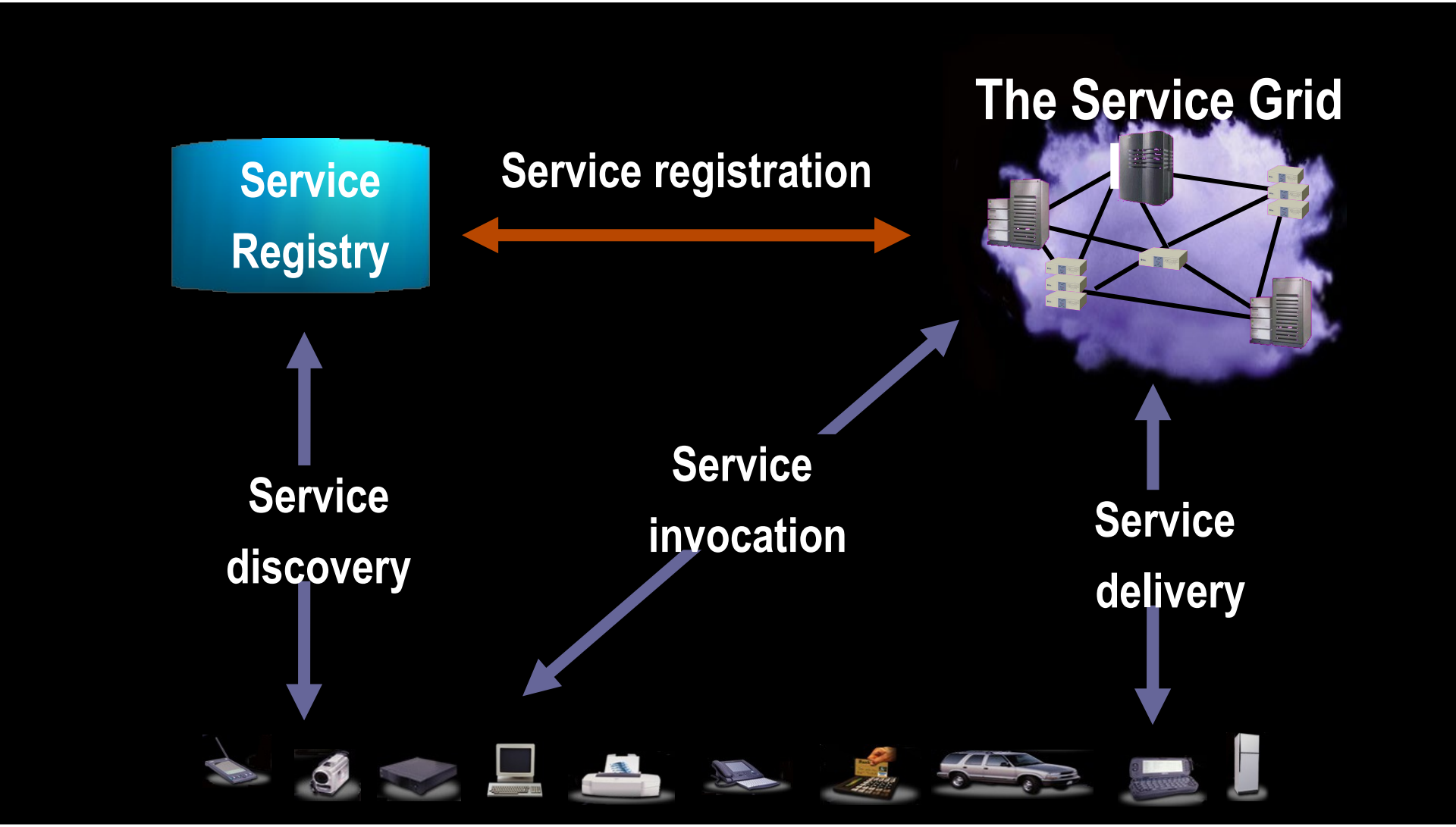
Web Services

- Program-to-program interaction
- Possibility of dynamic integration of components (in the future)
- Possibility of service aggregation (in the future)

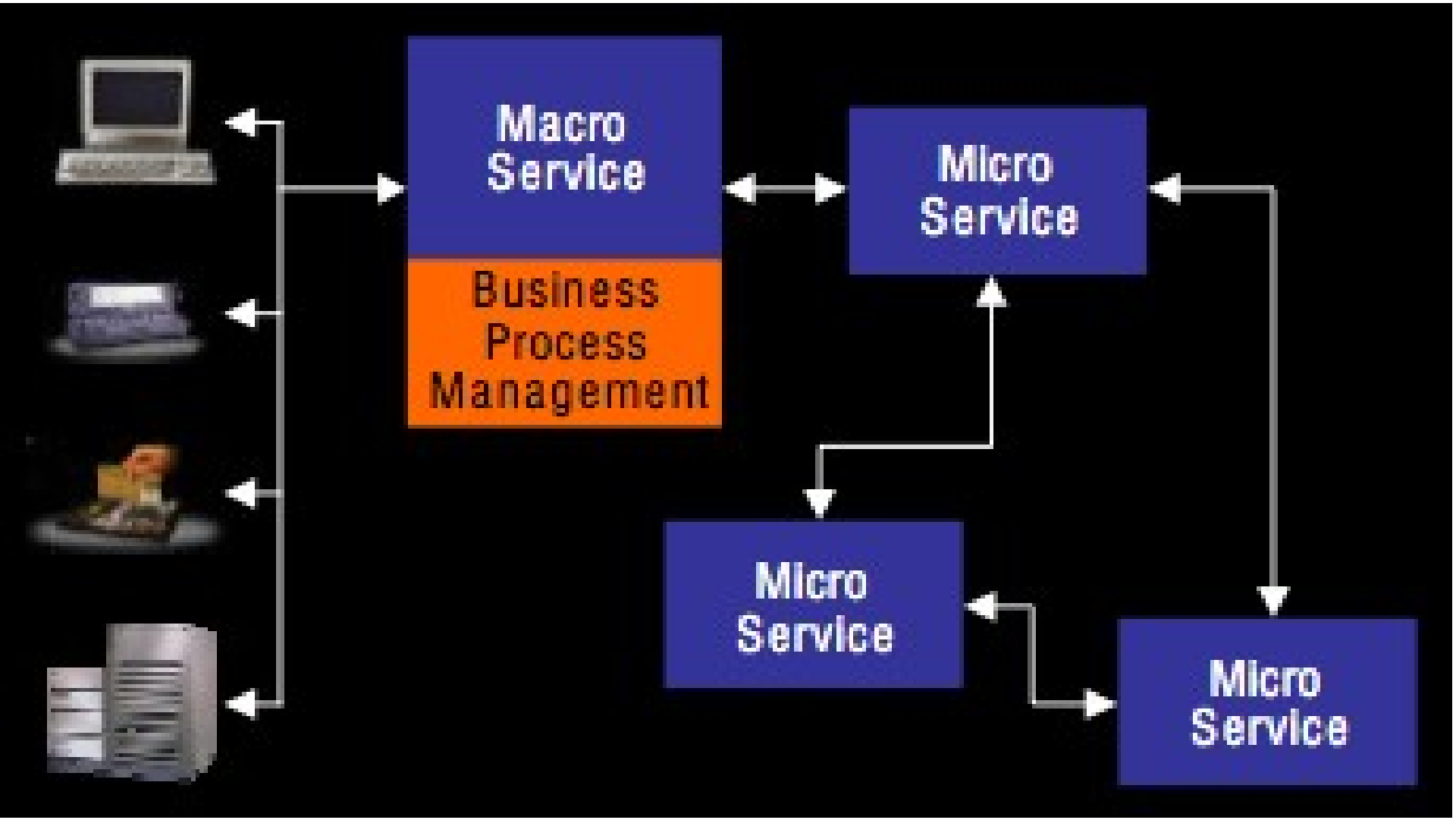
Characteristics of Web Services

- XML Based everywhere
- Message-based
- Programming Language independent
- Could be dynamically located
- Could be dynamically assembled or aggregated
- Accessed over the Internet
- Loosely coupled
- Based on industry standards

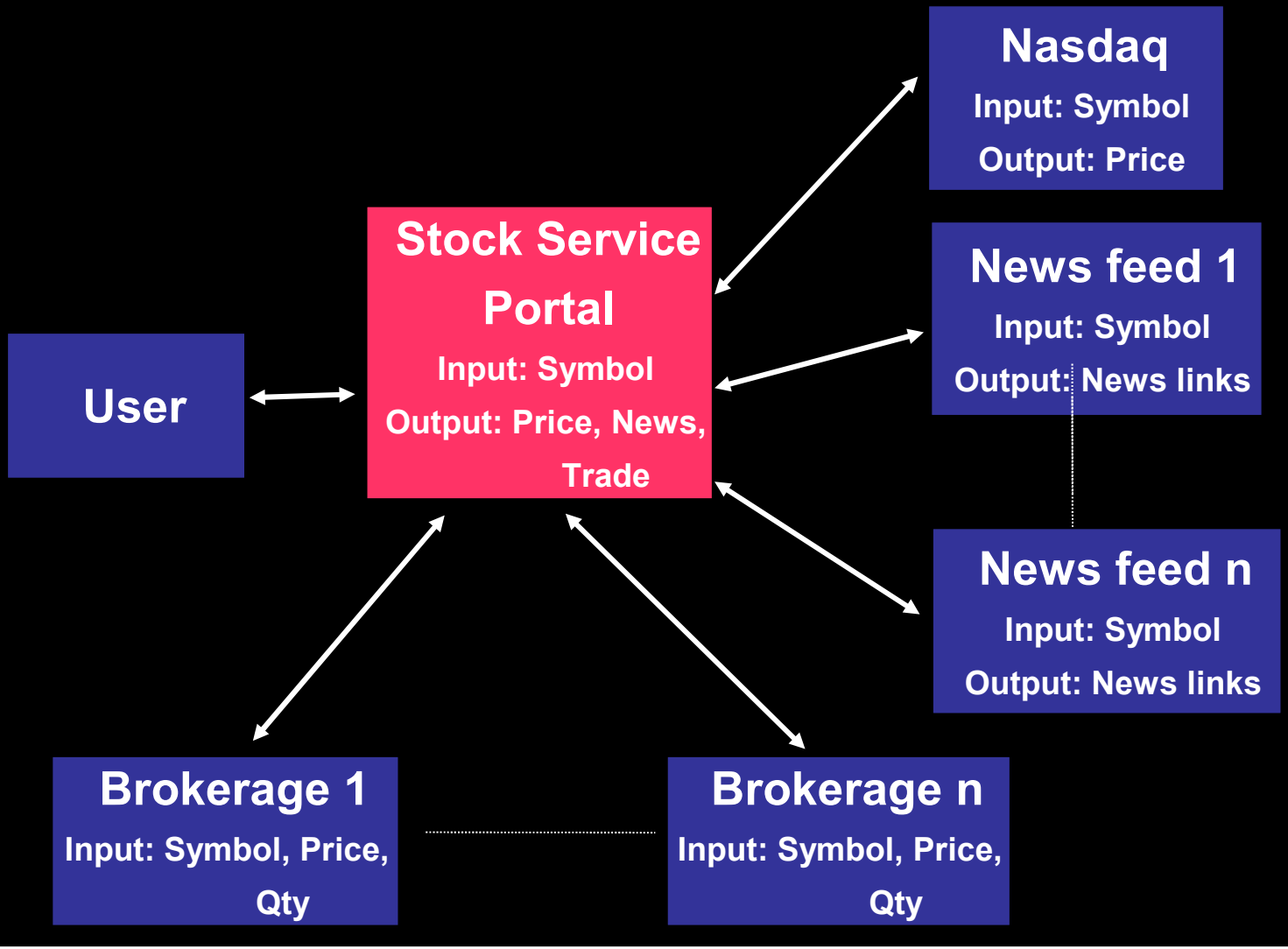
Web Services



Service Aggregation

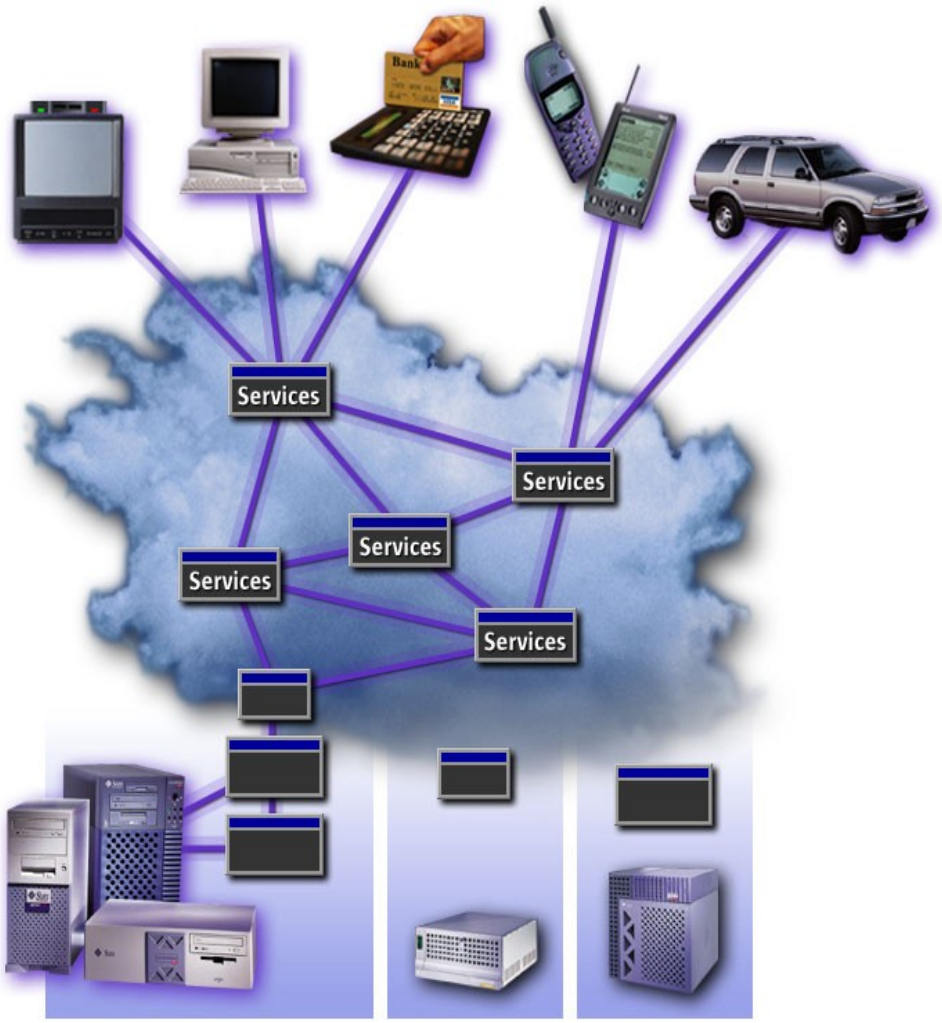


Service Aggregation Example



Why Web Services?

Why Web Services?



Web Services are:

- Are platform neutral
- Are accessible in a standard way
- Are accessible in an interoperable way
- Use simple and ubiquitous plumbing
- Simplify enterprise integration

Why Web Services?

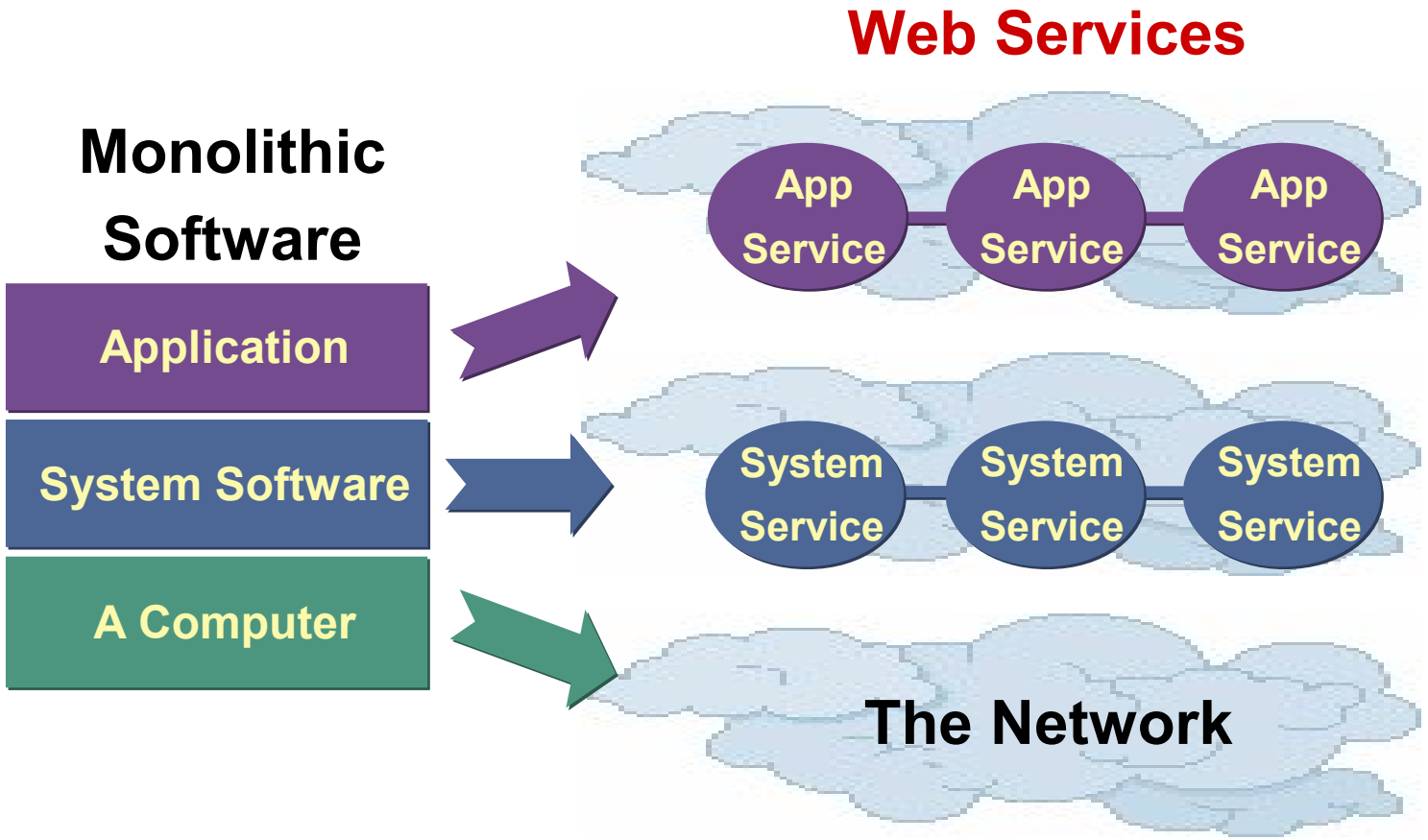
- **Interoperable** – Connect across heterogeneous networks using ubiquitous web-based standards
- **Economical** – Recycle components, no installation and tight integration of software
- **Accessible** – Legacy assets & internal apps are exposed and accessible on the web
- **Available** – Services on any device, anywhere, anytime
- **Scalable** – No limits on scope of applications and amounts of heterogeneous applications

State of Web Services

State of Web Services

- Basic Technology/Standards are well established
  - Service definitions, protocols – SOAP, WSDL
  - Security, transaction, state, and user context
  - Workflow, Identity management
- Abundant implementations
- Business Web Services are the next big thing, but more works are needed in:
  - Quality of Service, management
  - Provisioning, Accounting
- SOA is becoming reality

Impact of Web Services on Software: "Application Dis-Integration"



Web Services Support over Java Platform

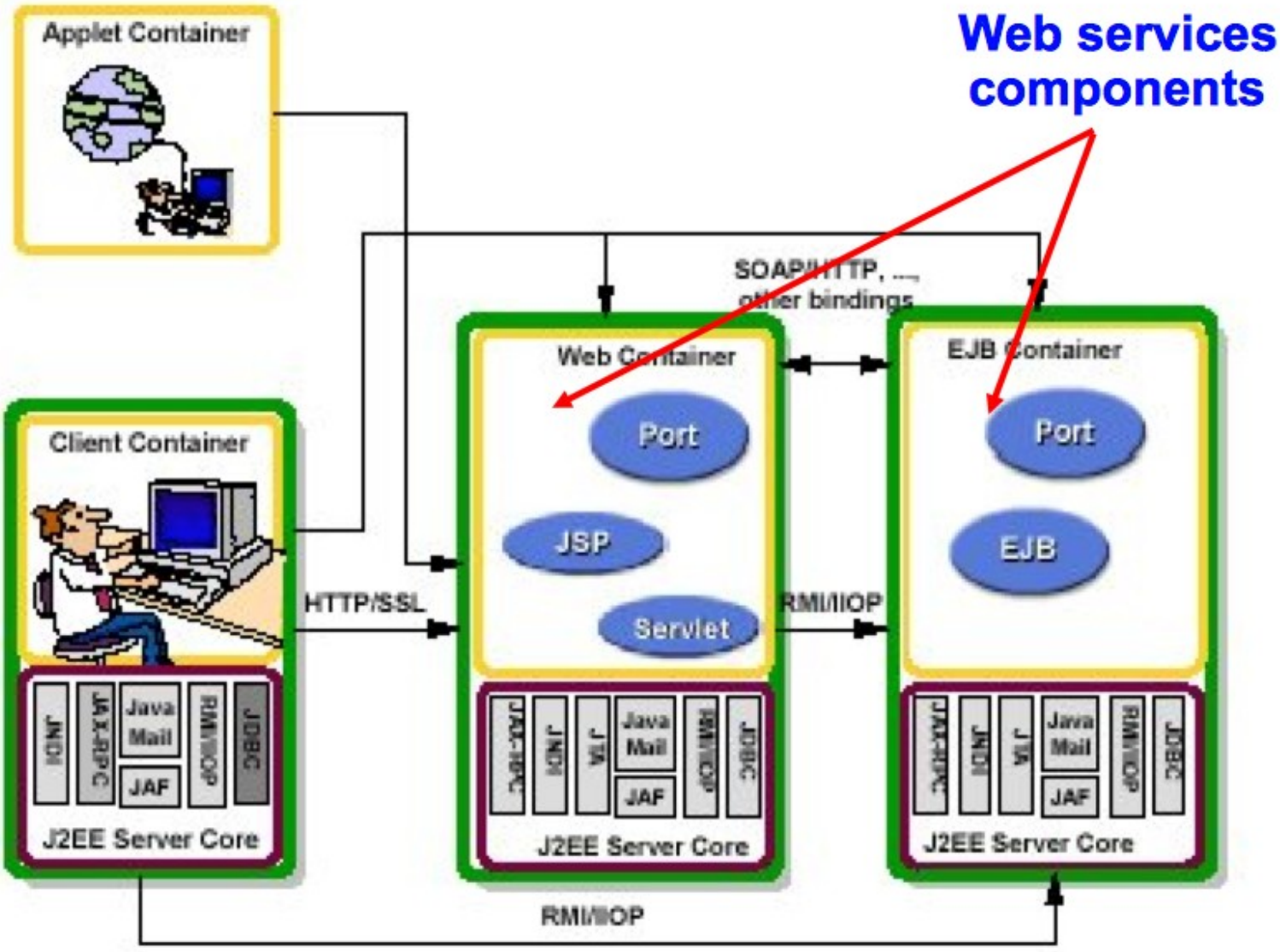
Why Java EE for Web Services?

- Web Services are just one of many service delivery channels of Java EE
  - No architectural change is needed
  - Existing Java EE componenets can be easily exposed as Web Services
- Many benefits of Java EE are preserved for Web Services
  - Portability, Scalability, Reliability
  - No single-vendor lock-in

Where are We now?

- Java APIs for Web Services are well-accepted
  - JAX-WS, JAXB
- Tools are available now for exposing existing Java EE components as Web Services
- Java EE community has defined overall framework for Web Services (Java EE 1.4, JSR 109)
- Java SE 6 can host Web services

Web Service Components



Source: Web Services for J2EE (JSR 109), V1.0

Thank you!

Sang Shin
http://www.JPassion.com
"Learn with JPassion!"