COMP3017
Service Computing

Answers to your feedback

- Based on your feedback, starting from the next lecture, I will post the ppt slides before the lecture, so that you can preview the materials before the class.
- To minimize your fatigue/time spent looking at the screen continuously, we will have several shorter sessions (3 or 4)

Answers to your feedback

- A student asked if our course will be similar to MIT’s 6.824 – the short answer is NO

MIT’s 6.824	Our course
Issues of distributed systems in general	Focus on services
Service computing is applied in many kinds of distributed systems	

Review

- Let us quickly review the concepts we learnt last time

Service

- Services represent a type of relationships-based interactions (activities) between at least one service provider and one service consumer to achieve a certain business goal or solution objective.

Multiple Choice(single) Points: 1

Setting

Consider an example of a Web Service: A company called Widget, Inc. sells parts through its website, enabling customers to submit purchase orders and check on order status. How do we call Widget Inc. in this scenario?

A Service requestor/ service consumer

[✓ B] Service provider

C Service registry

D I do not know

Submit

Open Question Points: 10

Setting

Let’ s continue with our example. Widget, Inc. sells parts through its website, enabling customers to submit purchase orders and check on order status. How do we call the customer of Widget Inc. in this scenario?

Answer

What is Service Computing?

- We said that Services Computing is a cross-discipline

Multiple Choice(multiple) Points: 1

Setting

Service computing aims to bridge 2 disciplines. Please select which ones.

[✓ A] Business

[✓ B] IT

C Math

D Finance

Submit

What is Service Computing?

- Services Computing is a cross-discipline that covers the science and technology of bridging the gap between business services and IT services.
- Supports integrating the business as linked, repeatable business tasks, or services.

Enterprise

- Enterprises (bank, aviation, restaurant)are made up of a set of Business Processes
- Most of these processes can be broken down into more fundamental discrete building blocks known as services.

Open Question Points: 10

Setting

Can you give examples of a few business processes common in many companies?

Open Question is only supported on Version 2.0 or newer.

Answer

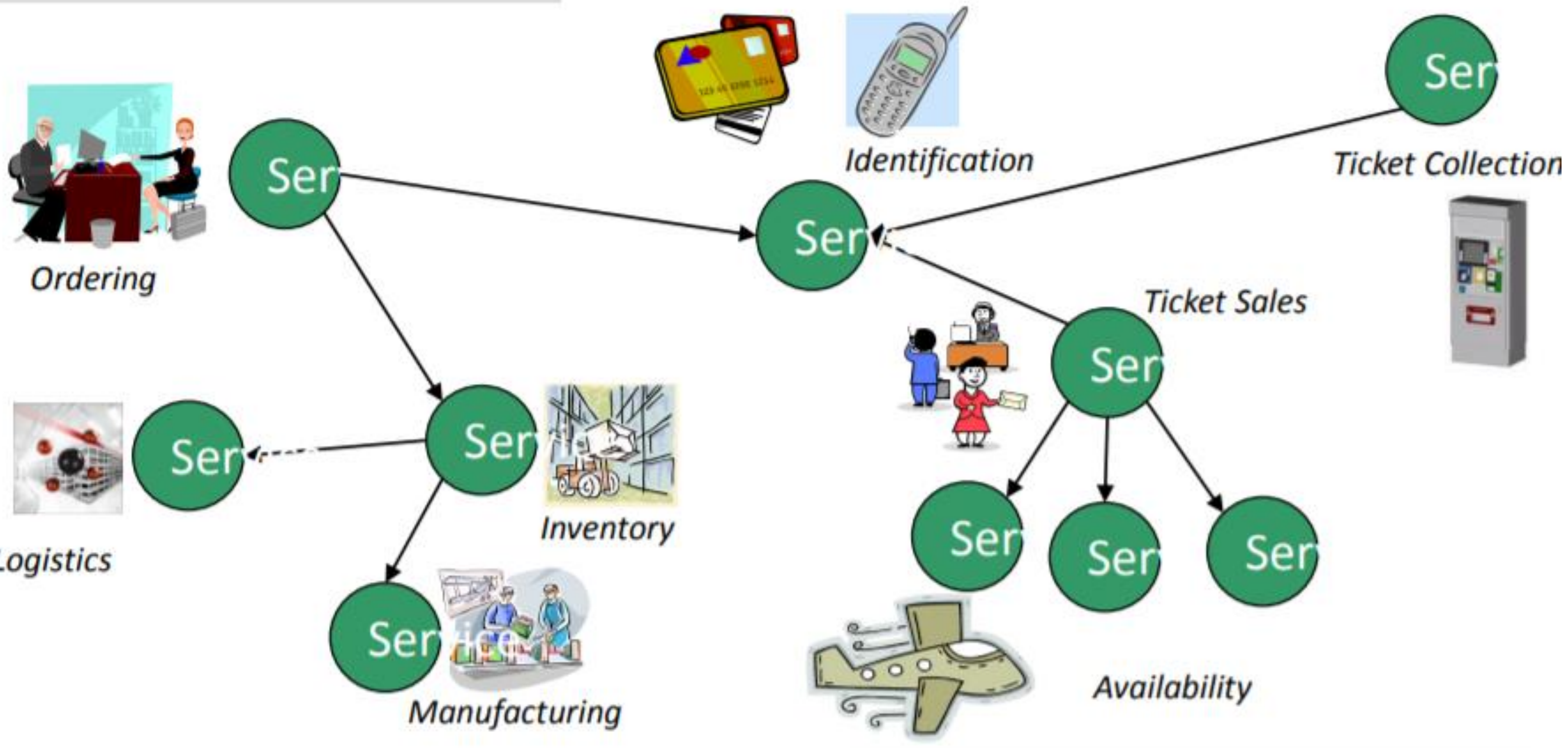
Motivation - ENABLE FLEXIBLE, FEDERATED BUSINESS PROCESSES

- Enable flexible, federated business processes

Enabling a virtual federation of participants to collaborate in an end-to-end business process

Enabling alternative implementations

Enabling reuse of Services



Enabling virtualization of business resources

Enabling aggregation from multiple providers

Module One: Introduction to Service Computing and XML-RPC

The goal of Service Computing

- Enable IT services and computing technology to perform business services more efficiently and effectively.
  - How to divide distributed systems to "services" which can be separately invoked through network requests and can provide independent functionalities
  - How to manage and evaluate existing services
  - How to reuse these services building more complex, composite services
  - How to assure quality of services

Activities in a service lifecycle

- business componentization
- services modelling
- services creation
- services realization
- services annotation
- services deployment
- services discovery
- services composition
- services delivery
- service-to-service collaboration
- services monitoring
- services optimization
- services management

Business benefits – decreased cost

- Decreased cost:
  - Add value to core investments by leveraging existing assets
  - New systems can be built faster for less money
    - Reducing integration expense
    - Built for flexibility
    - Long term value of interoperability

Business benefits – increased productivity

- Increased employee productivity:
  - Built on existing skills
  - Consolidate duplicate functionality

Business benefits - partnership

- Built for partnerships:
  - Standards based
  - Business relationships expressed via service interactions
  - Integration is driven by what is needed, not what is technically possible

Business benefits – agility

- Agility - Built for change
  - Helps applications evolve over time and last
  - Abstract the backend and replace over time
  - Focusing on core-competencies
  - Incremental implementation approach is supported
  - Service Outsourcing – new business model!

Technical Benefits

- Services Scale
  - Build scalable, evolvable systems
  - Scale down to mobile devices
  - Scale up to for large systems or across organizations
- Manage complex systems
  - Does not require centralized services
  - Empowers users with high end communication
- Platform independent use
- Loose Coupling allows flexibility

Service computing applications

- Web applications
- Mobile applications
- Cloud systems
- Big data applications
- IoT Systems
- Blockchain systems
- Workflow systems
- Other distributed systems

Underlying technology

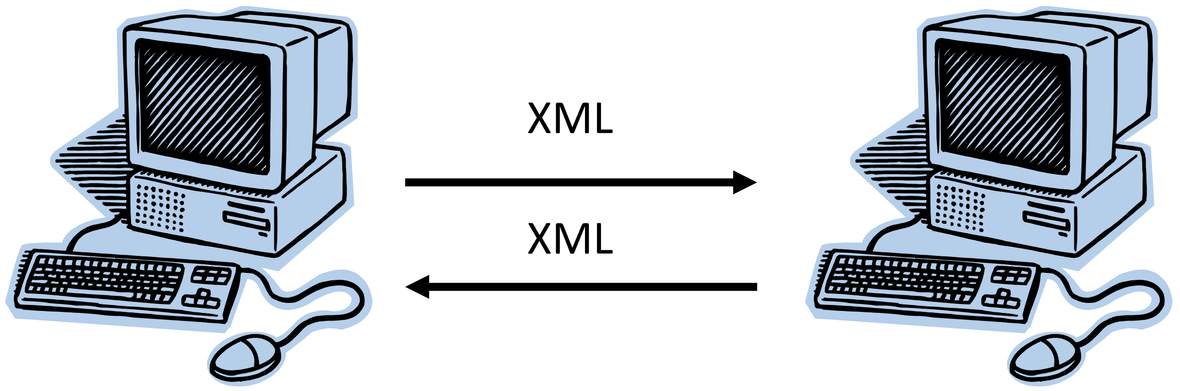
- Web services
- Service-oriented architecture (SOA)
- Cloud computing
- Business process modelling
- ….

What is a Web Service?

What is a Web Service?

- A Web Service is any service that:
  - Is available over the Internet or private (intranet) networks
  - Uses a standardized XML messaging system
  - Is not tied to any one operating system or programming language

A Basic Web Service

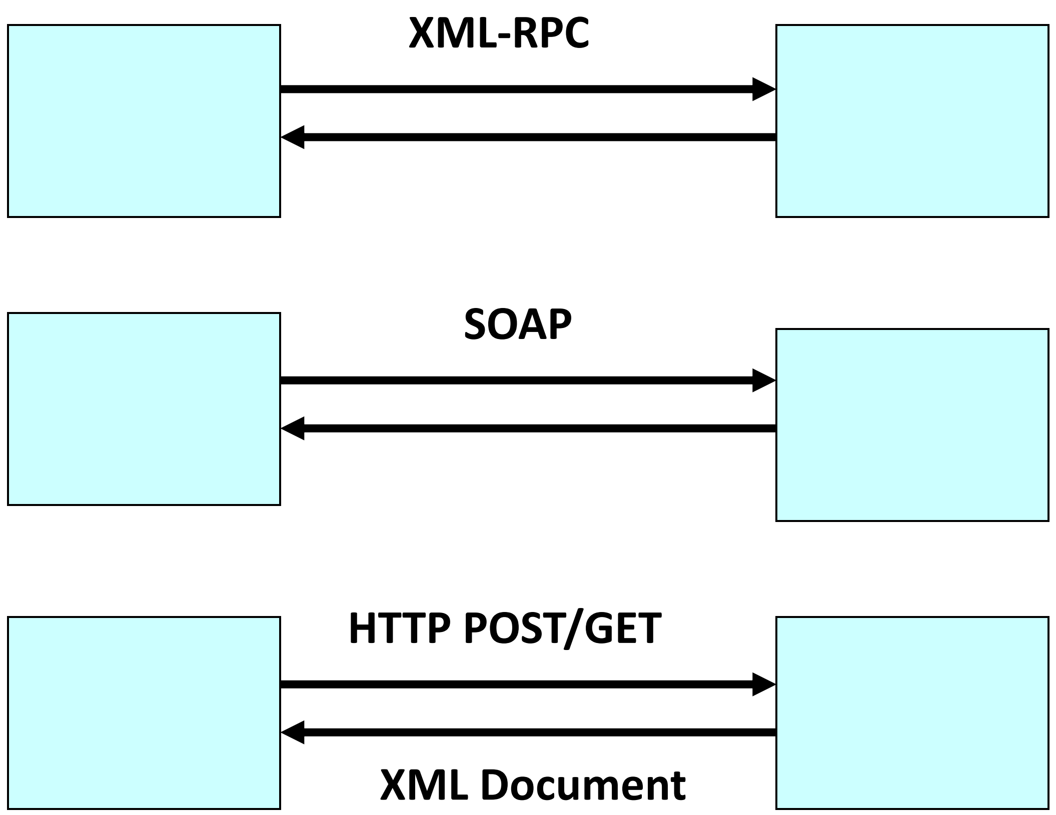


Computer A:
Language: Perl
Operating System: Windows 2000

Computer B:
Language: Java
Operating System: Linux

XML Messaging

- There are several alternatives for XML messaging:
  - XML Remote Procedure Calls (XML-RPC)
  - SOAP
  - Regular XML transported over HTTP
- Any of these options are valid.



Web Services Defined

- Although not required, a web service may also have two additional (and desirable) properties:
  - a web service should be *self describing*.
  - a web service should be *discoverable*.

Web Services: *Self Describing*

- If you publish a new web service, you should also publish a public interface to the service.
- At a minimum, you should include human-readable documentation so that others can easily integrate your service.
- If you have created a SOAP service, you should also include a public interface written in a common XML grammar.

Web Services: *Discoverable*

- If you create a web service, there should be a relatively simple mechanism to publish this fact.
- Likewise, interested parties should be able to easily discover your service.
- The discovery service could be completely decentralized or completely centralized.

Web Services: Summary

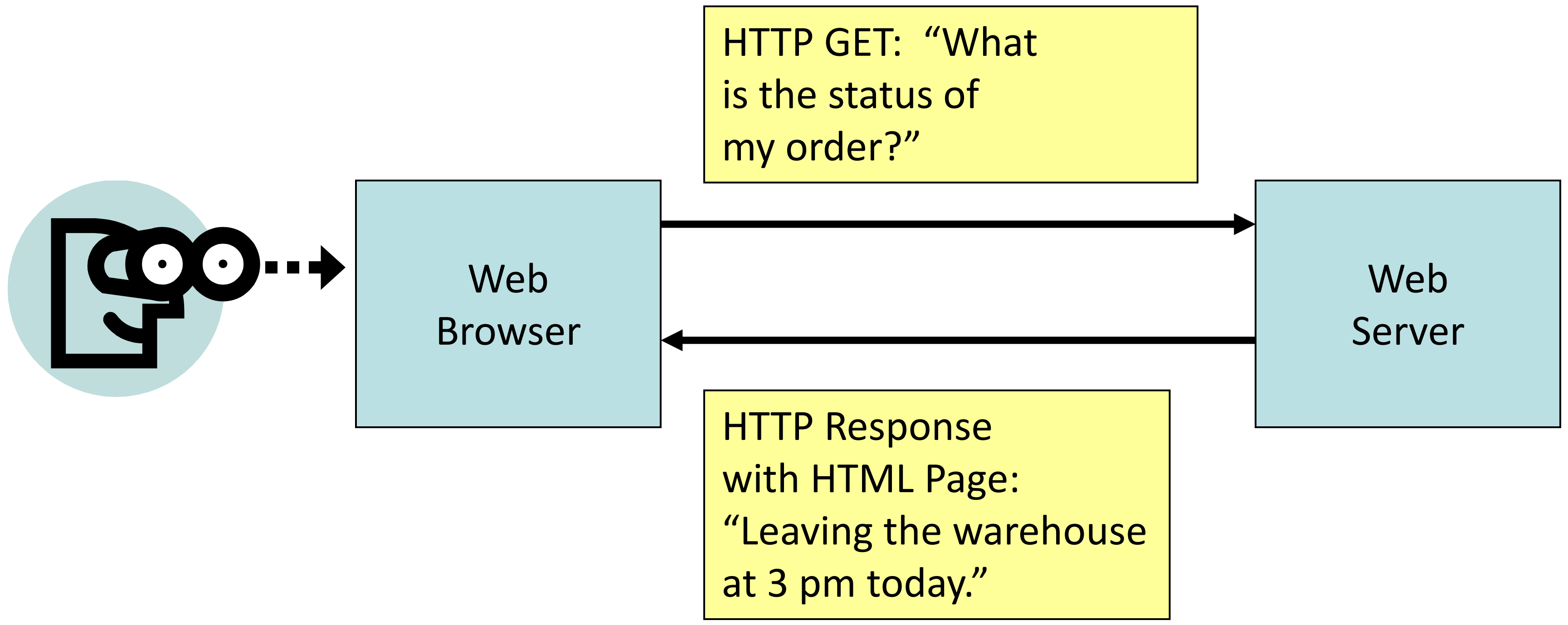
- To summarize, a complete web service is any service that:
  - Is available over the Internet or private (intranet) networks
  - Uses a standardized XML messaging system
  - Is not tied to any one operating system or programming language
  - Is self-describing via a common XML grammar
  - Is discoverable via a simple find mechanism

The Impact of Web Services

Web Services in Action

- To understand the impact of web services, consider basic e-commerce functionality.
- For example, Widgets Inc. sells parts through its web site.
  - Customers can purchase parts and check on order status.

The Human Centric Web

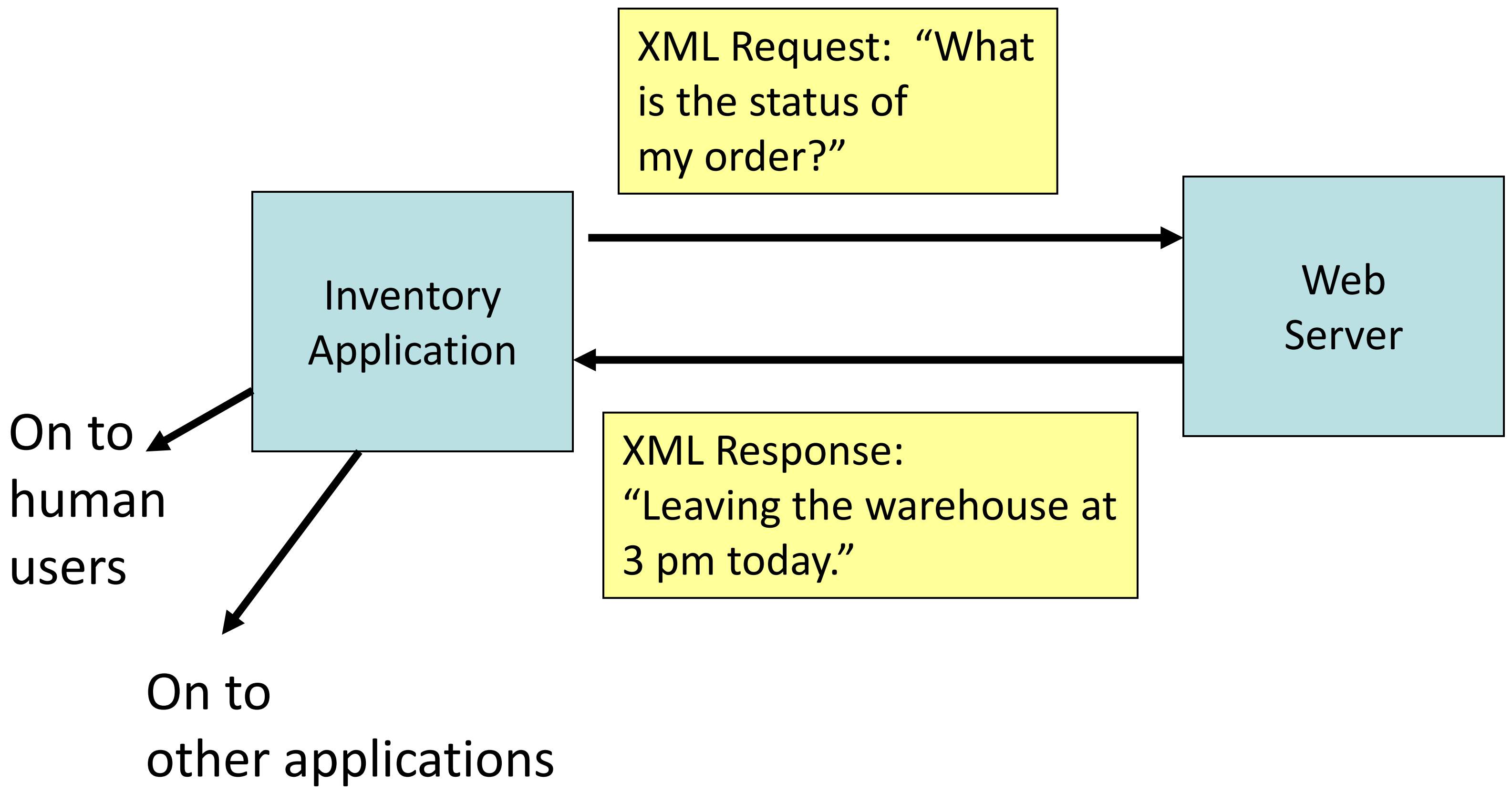

HTTP GET: "What is the status of my order?"
Web Browser
Web Server
HTTP Response with HTML Page: "Leaving the warehouse at 3 pm today."

- This illustrates a *human-centric web*, where humans are the primary actors initiating web requests.

Web Services: Application-Centric Web

- With web services, we move from a human-centric web to an *application-centric* web.
- In other words, conversations between applications occur as easily as conversations between web browsers and servers.

The Application-Centric Web


XML Request: "What is the status of my order?"
Inventory Application
Web Server
XML Response: "Leaving the warehouse at 3 pm today."
On to human users
On to other applications

- The order status is now a web service.
- Applications can therefore connect to the order status service directly.

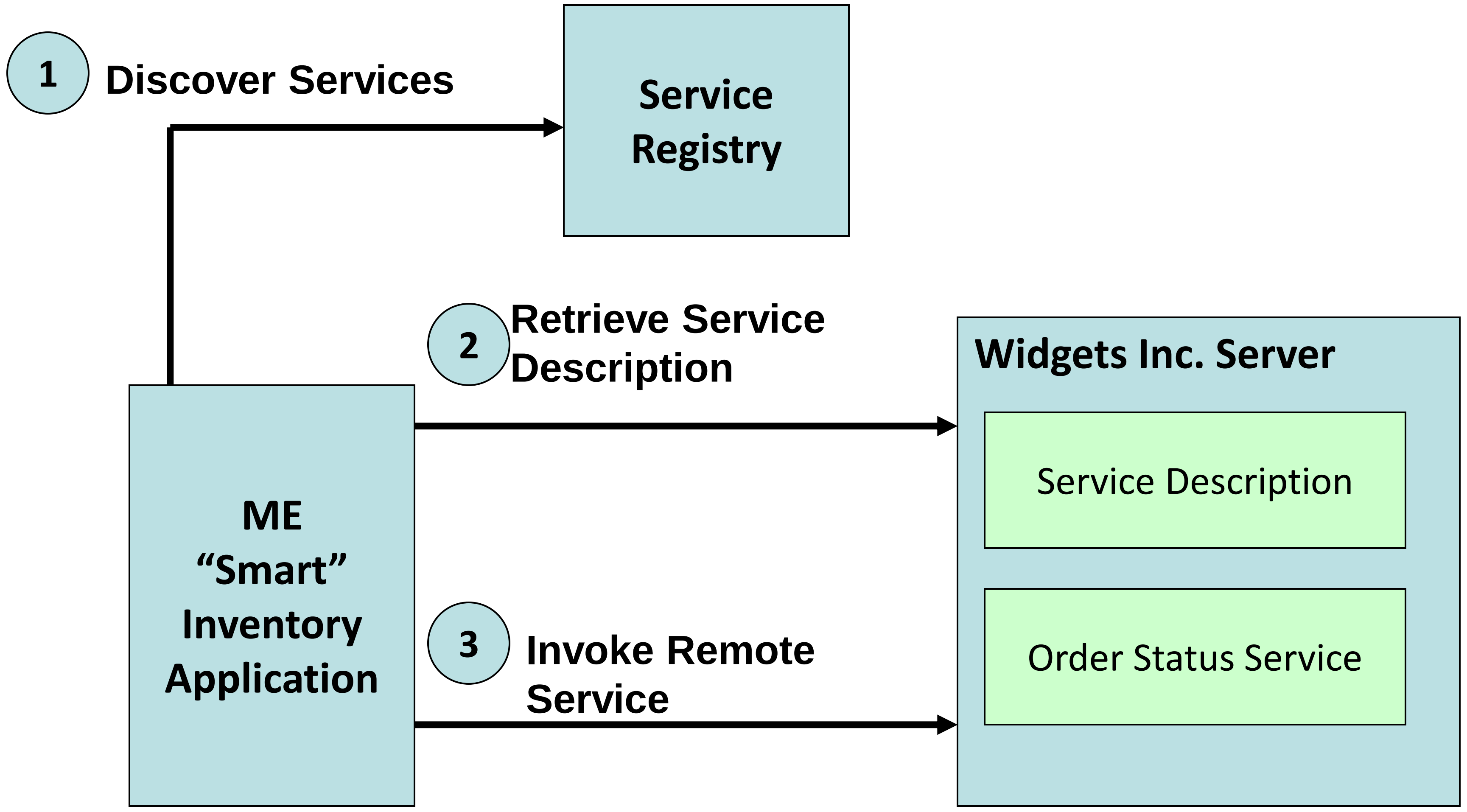
Application-Centric Web

- There are numerous areas where an application-centric web would be extremely helpful:
  - credit card verification
  - package tracking
  - shopping bots
  - single sign-on registration
  - calendar, email, etc.

The Long-term Vision: Automated Web

- In the long-term, web services offer the promise of the *automated Web.*
- "Just-in-time" integration:
  - If services are easily discoverable, self-describing, and stick to common standards, it is possible to automate application integration.
- For example, consider a company, Mega Electric (ME) that wants to buy parts from Widgets,Inc.
  - ME wants to automatically integrate inventory with Widgets, Inc. order status service.

Just-In-Time Integration



Standardization

- The World Wide Web Consortium is heavily involved in standardizing web services.
  - https://www.w3.org/standards/webofservices/

Web Services Architecture

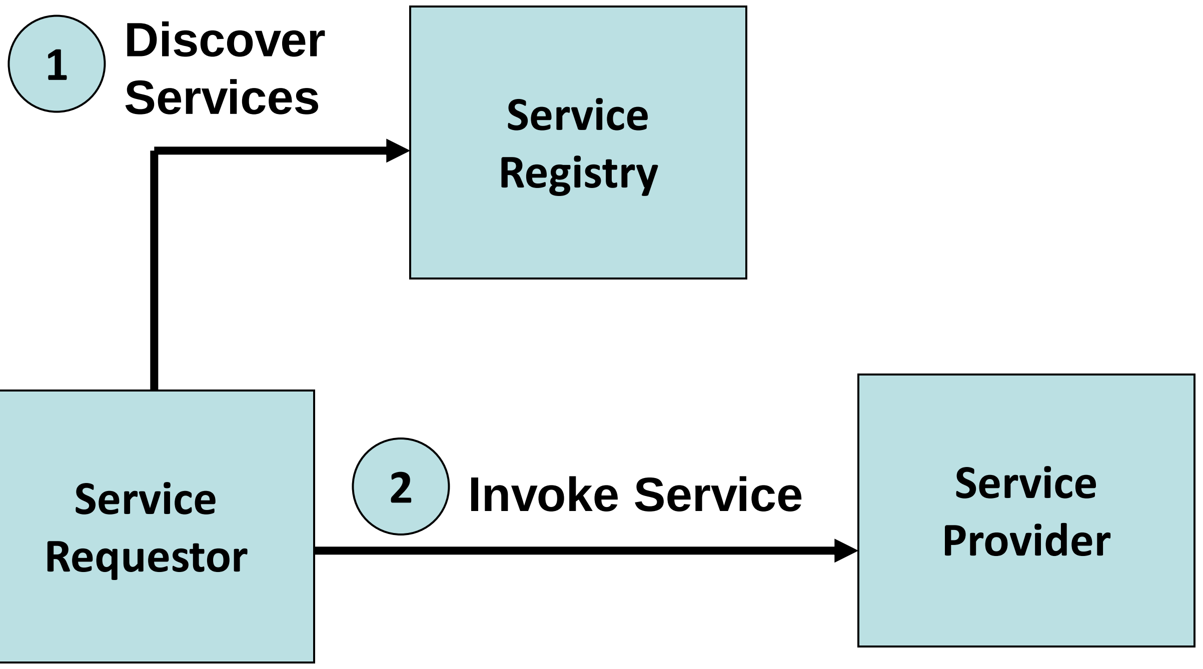
Web Service Architecture

- There are two ways to view the web service architectural framework:
  1) Examine individual roles of each web service actor
  2) Examine the emerging web service protocol stack.

Web Service Roles

- Three major roles in web services:
  - Service Provider: provider of the web service.
  - Service Requestor: any consumer of the web service.
  - Service Registry: logically centralized directory of services.

Web Service Roles



What to do next?

- Go to the Blackboard to find ppt slides with today's lecture

&

- See you on Monday in Tencent Meetings/Rainclassroom!