COMP3017
Service Computing

Hot topic study task 2 - individual

- Think about your group's topic from the perspective of services or service computing
  - How would we define services within your topic?
  - Can you come up with any examples?
  - Applications?
- Submit your answers by creating a new entry in the Blackboard journal by April 2nd (Saturday)

Module 1-3 review

Open Question Points: 10

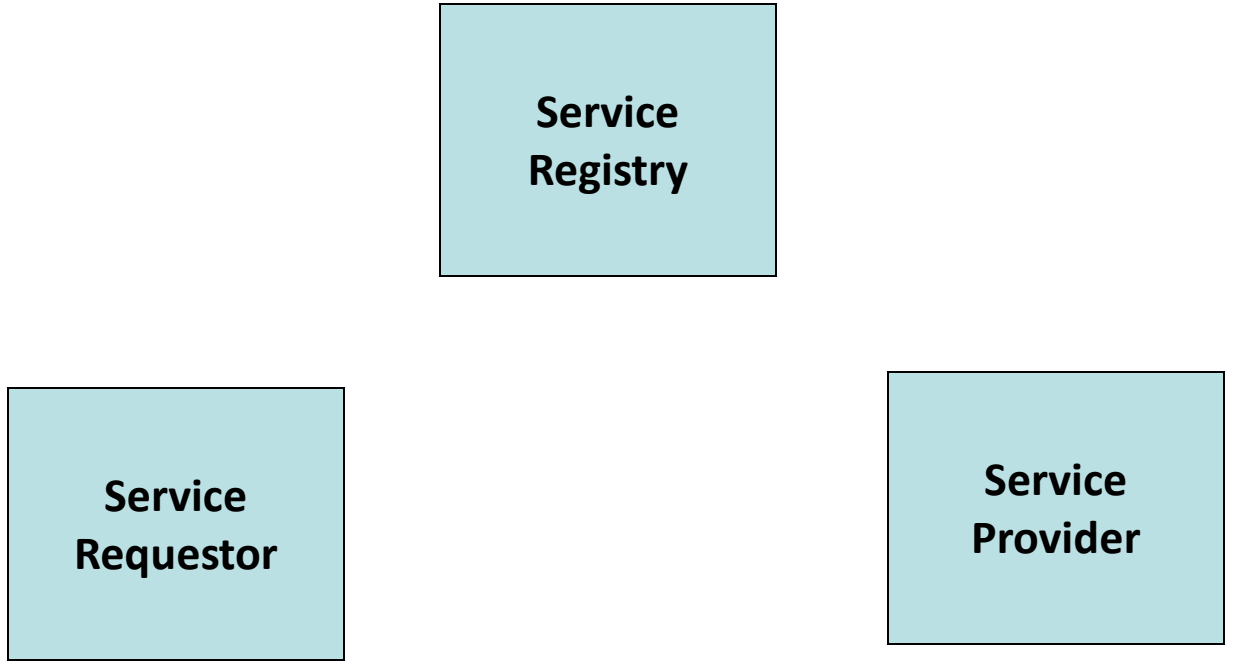
Setting

What are the names of the three actors in web services architecture?

Open Question is only supported on Version 2.0 or newer.

Answer

Web Service Roles



Open Question Points: 10

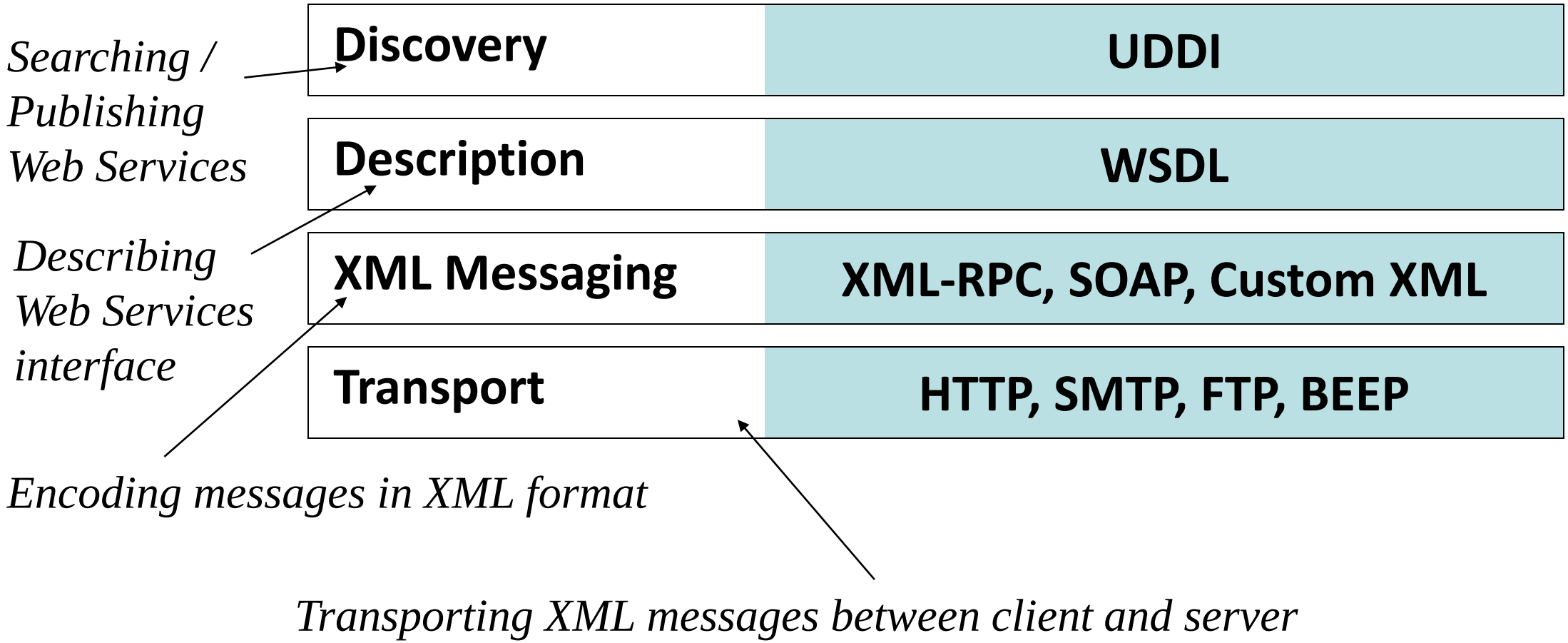
Setting

What is the purpose of web service protocol stack?

Open Question is only supported on Version 2.0 or newer.

Answer

Web Service Protocol Stack



Open Question Points: 10

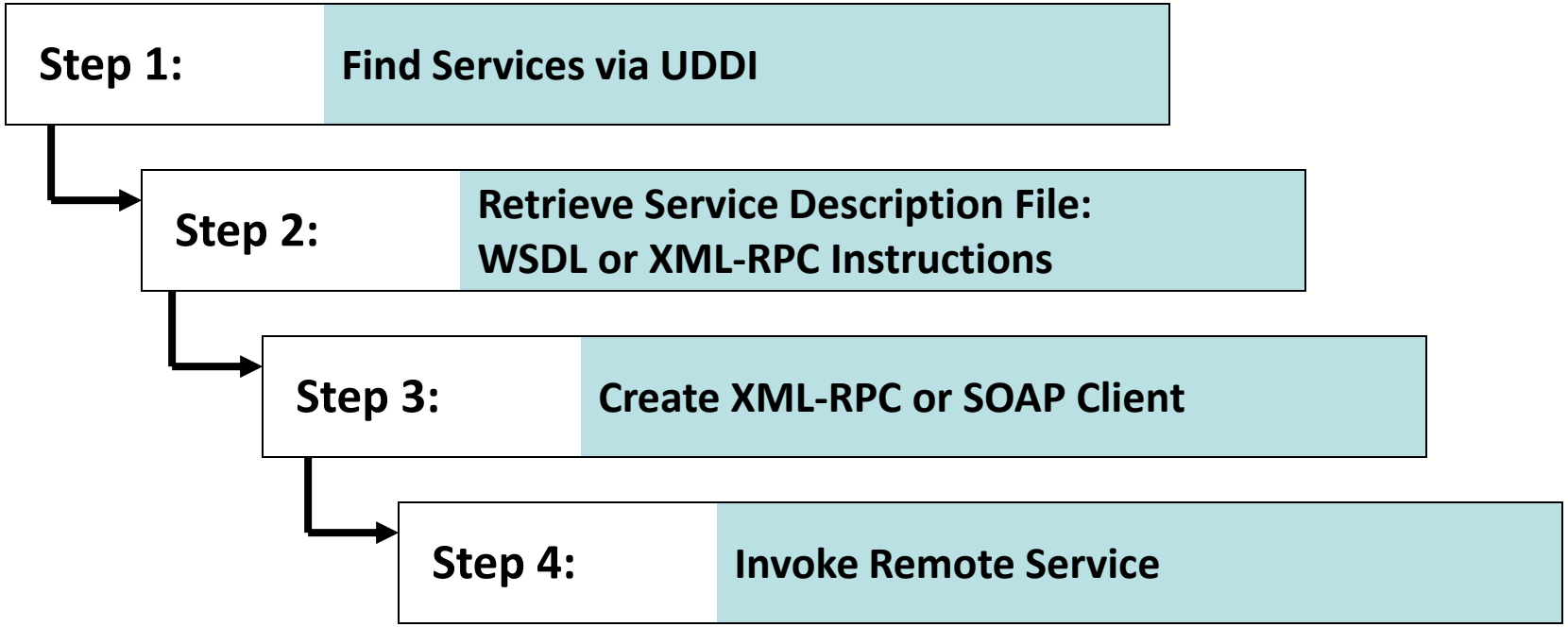
Setting

How service requestor uses the protocols in web service protocol stack?

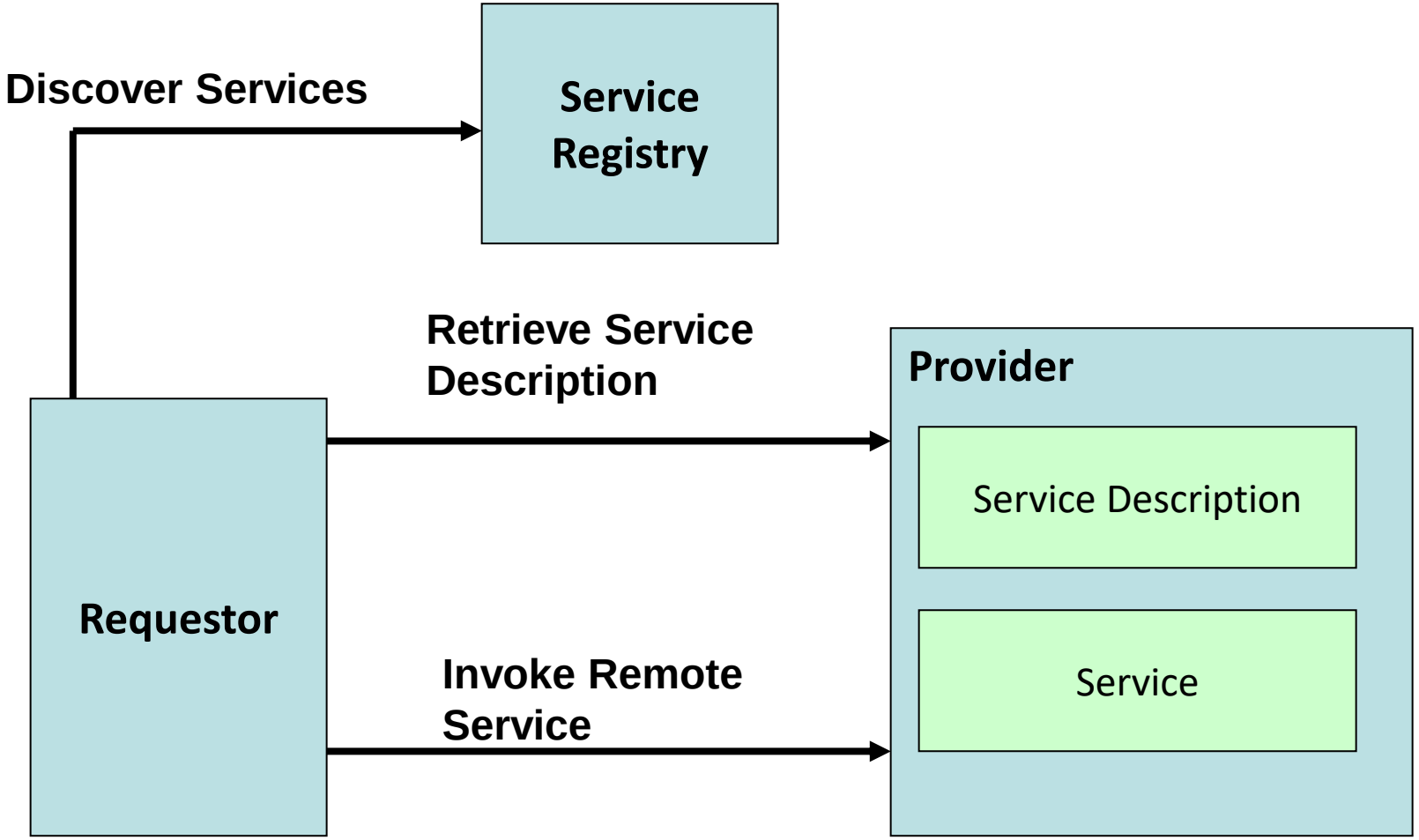
Open Question is only supported on Version 2.0 or newer.

Answer

Using the Protocols Together – service requestor perspective



Service requestor



Open Question Points: 10

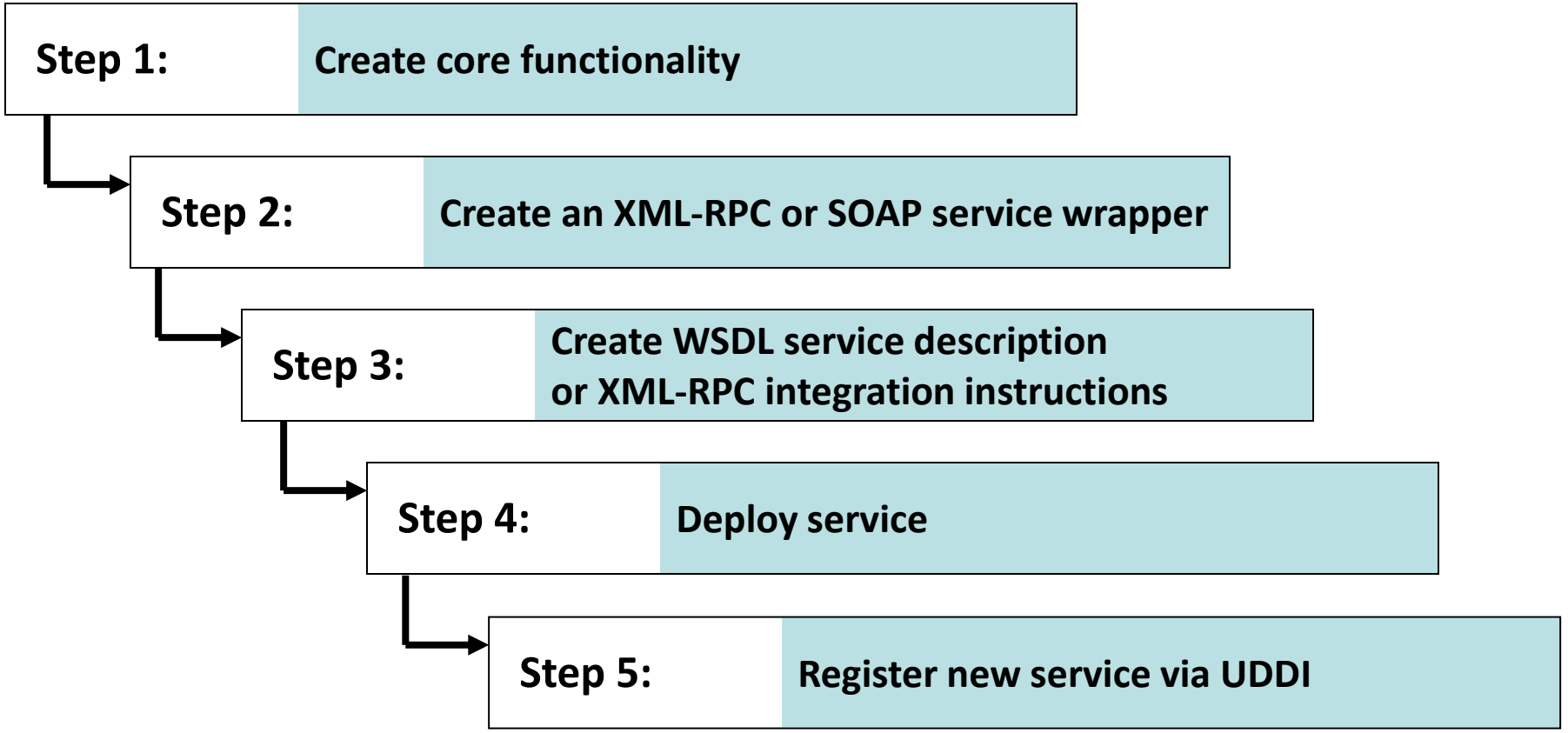
Setting

How service provider uses the protocols in web service protocol stack?

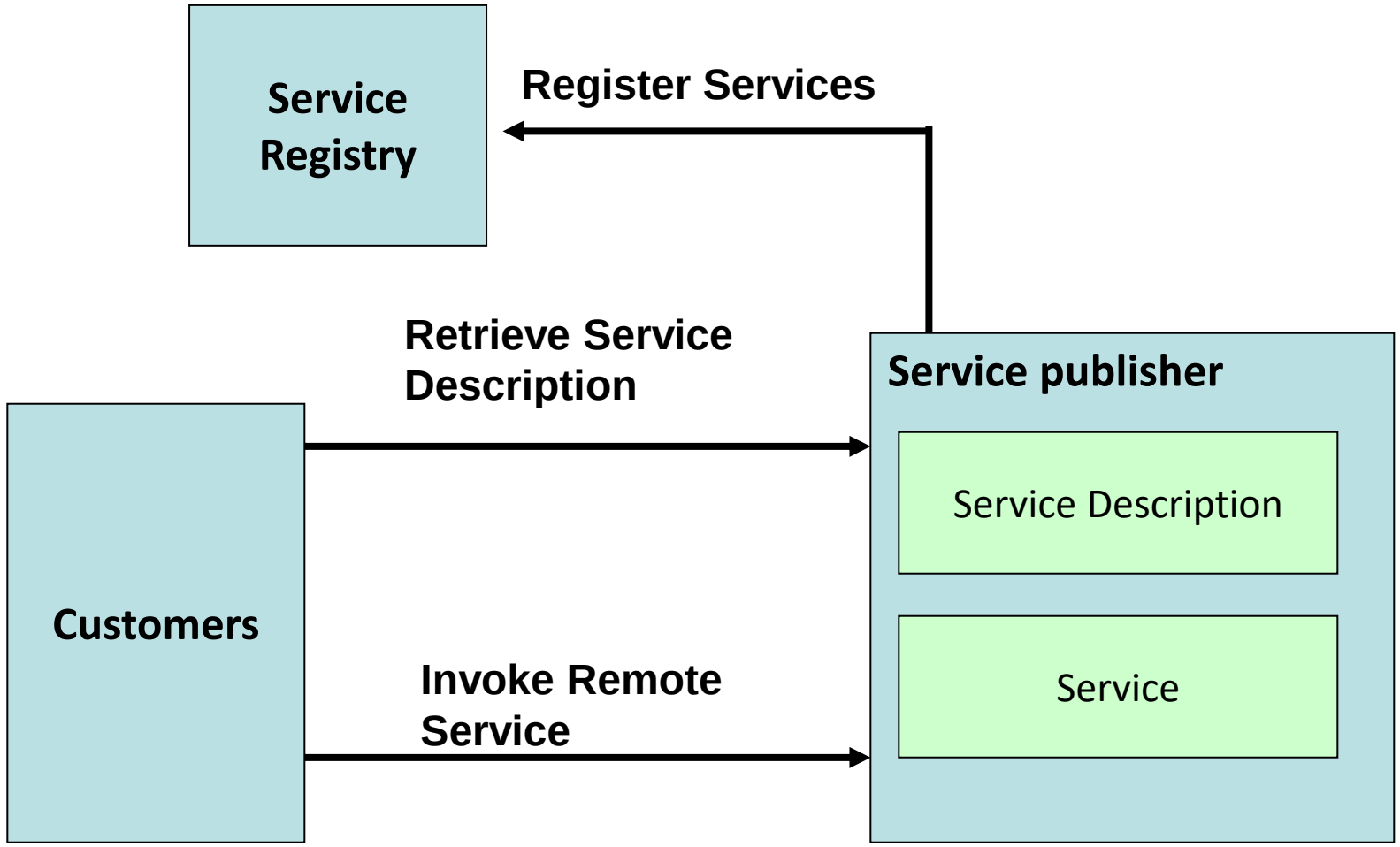
Open Question is only supported on Version 2.0 or newer.

Answer

Using the Protocols Together – service provider perspective



Service Provider



Module Four: Basics of web service programming; developing web services with JEE platform

Our materials for this module:

- Tutorial document
  - You can find it in the Blackboard
- Java guide document
  - Also available in the Blackboard, additional help for the students who are not familiar with Java

Module 4 Learning Outcomes

- be familiar with JAX-WS
- create a SOAP web service using two ways: top-down approach and bottom-up approach
- use *wsimport* tool
- generate web service source files from WSDL
- publish the web service end points
- connect to the web service
- create a JAX-WS client
- make web service calls

Developing Web Services with Java EE platform

Java EE – Java Enterprise Edition

- Java Platform, Enterprise Edition
- Formerly known as
  - Java 2 Platform, Enterprise Edition (J2EE)
- The current version is known as
  - Jakarta EE
- A set of specifications, extending Java SE 8 with specifications for enterprise features such as distributed computing and web services
- Java EE is defined by its specification
  - It defines APIs and their interactions

Java EE specifications

- Web specifications
  - Servlet
  - WebSocket
  - Java Server Faces
  - Unified Expression Language
- Web Service specifications
  - Java API for RESTFful Web Services
  - Java API for JSON Processing
  - JAVA API for JSON Binding
  - Java Architecture for XML Binding
  - Java API for XML Web Services
- Enterprise specifications
  - Contexts and Dependency Injection
  - Enterprise JavaBean
  - Java Persistence API
  - Java Transaction API
  - Java Message Service
- Other specifications
  - Validation
  - Batch Applications
  - Java EE Connector Architecture

JAX-WS

- A part of the Java EE platform
- Java API for XML Web Services
- A Java programming language API for creating web services, particularly SOAP services
- One of the JAVA XML programming APIs
- It can be used in Java SE starting with version 6

JAX-WS 2.2 specification

- https://jcp.org/en/jsr/detail?id=224
- Defines a standard Java-to-WSDL mapping which determines how WSDL operations are bound to Java methods when a SOAP message invokes a WSDL operation
  - Determines which Java method gets invoked and how that SOAP message is mapped to the method's parameters
  - Determines how the method's return value gets mapped to the SOAP response

Using JAX-WS to create Web Services

- In this module, you will use JAX-WS to
  - Create a SOAP Web Service
  - Connect to a SOAP Web Service
- Please follow the tutorial document uploaded to our WeChat group
- If you have any problems or questions, contact us on WeChat

Creating and Connecting to SOAP Web Service using JAX-WS

- Java API for XML Web Services (JAX-WS)
  - a standardized API for creating and consuming SOAP web services
  - a framework that simplifies using SOAP
- In this Tutorial, we'll create a SOAP web service and connect to it using JAX-WS

Tools

- A Java Development kit(JDK).
- The Inject java package

Web services built on JAX-WS

- The web service built on JAX-WS can be divided into two parts
  - Server
  - Client

Server

- There are two ways of building SOAP web services.
  - Top-down approach (contract-first)
  - Bottom-up approach(contract-last)
- In this tutorial, we'll take a look at both approaches.

Top-down approach (contract-first)

- WSDL document is created first
- The necessary Java classes are generated from the WSDL

Bottom-up approach(contract-last)

- Java classes are written first
- WSDL is generated from the Java classes

Comparison of top-down and bottom-up approaches

- Bottom-up approach is an easier option in case of complex web services
  - Writing a WSDL file can be quite difficult depending on how complex your web service is
- Since WSDL is generated from the Java classes, any change in code might cause a change in the WSDL
  - This is not the case for top-down approach

Client

- We can use wsimport tool, a web service tool provided by Java, to generate the client code automatically.

Tutorial review

Multiple Choice(multiple) Points: 1

Setting

(Please select 3 answers) What will you do in this tutorial?

A Create a SOAP web service using top-down approach

B Create a SOAP web service using bottom-up approach

C Use the WSDL file from the web service to generate the client code

D Write the client code manually

Submit

Multiple Choice(single) Points: 1

Setting

In JAX-WS, can we generate the client code automatically?

A YES

B NO

Submit

Multiple Choice(single) Points: 1

Setting

What is the name of the web service tool provided by Java, that allows us to generate the client code from the WSDL file automatically?

A JAX-WS

B wsimport

C SOAP

Submit

Programming assignment

- Please follow the tutorial document uploaded to the Blackboard
- If you have any problems or questions, contact us on WeChat

Tutorial Assessment questions

- After completing the tutorial, as your assignment, you will need to answer the following questions:
  - In the top-down service, can the client use getEmployee, updateEmployee methods? Please explain.
  - If we want the client connected to the bottom-up web service to use countEmployees method, what modification should we do in the EmployeeService.java and EmployeeServiceImp.java ?
  - What is GlassFish?

Tutorial assignment

- For completing the tutorial and submitting answers to the questions, you can get 10 points towards your final grade – 1/2 of “Exercises” component
  - Complete exercises in the “Module 4 web service tutorial”
  - Prepare the document with your answers to the tutorial assessment questions
  - Submit the tutorial answers document in the Blackboard assignment box by 10th April

Module 4 Summary

- Basics of web service programming
- Introduction to JAX-WS

Assignment deadlines reminder

- April 2nd – hot topic study journal
- April 10th – web service programming tutorial

Scalable Service Composition and Reconfiguration in Pervasive Computing Environments

(this topic is not included in the final examination)

Publications

Book Chapter

- Joanna Izabela Siebert, Jiannong Cao, “Scalable Service Composition in Pervasive Computing Environments,” In Scalable Computing and Communications: Theory and Practice, S.U. Khan, L.Wang, A.Y. Zomaya (Eds.), John Wiley \& Sons, Ltd., 2013.
- Jiannong Cao, Joanna Izabela Siebert, “Service Management in Pervasive Computing Environments,“ In Pervasive Computing and Networking, M.S. Obaidat, M.Denko, I.Woungang (Eds.), John Wiley \& Sons, Ltd., 2011.

Journal Paper

- Joanna Izabela Siebert, Jiannong Cao, Steven Lai, Peng Guo, and Weiping Zhu, “LASEC: A Localized Approach to Service Composition in Pervasive Computing Environments,” submitted to IEEE Transactions on Parallel and Distributed Systems.

Conference Paper

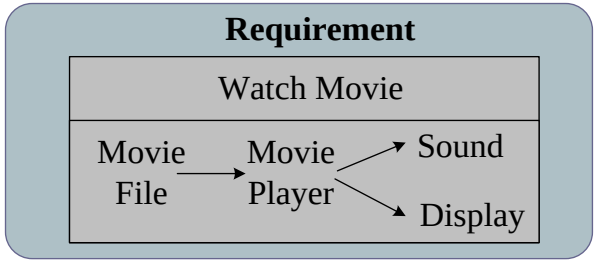
- Joanna Izabela Siebert, Jiannong Cao, Long Cheng, Edwin Wei, Canfeng Chen, and Jian Ma, “Decentralized Service Composition in Pervasive Computing Environments,“ In Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC '10), pp. 1258-1262, June 28-July 02, 2010. Caen, France.
- Joanna Izabela Siebert, Jiannong Cao, "Service Composition in Pervasive Computing Environments: a Survey,“ In Proceedings of 3rd International Interdisciplinary Technical Conference of Young Scientists, May 19-21, 2010. Poznan, Poland.
- Joanna Izabela Siebert, Jiannong Cao, Yu Zhou, Miaomiao Wang, and Vaskar Raychoudhury, “Universal Adaptor: A Novel Approach to Supporting Multi-protocol Service Discovery in Pervasive Computing,”, In Proceedings of International Conference on Embedded and Ubiquitous Computing (EUC’07), pp. 683-693, December, 2007, Taipei, Taiwan.

Outline

- Overview of Service Composition and Reconfiguration in PvCEs
- LASEC: Localized Approach to Service Composition
- LASER: Localized Approach to Composed Service Reconfiguration
- LASEH: Localized Approach to Service Heterogeneity
- Conclusion and Future Work

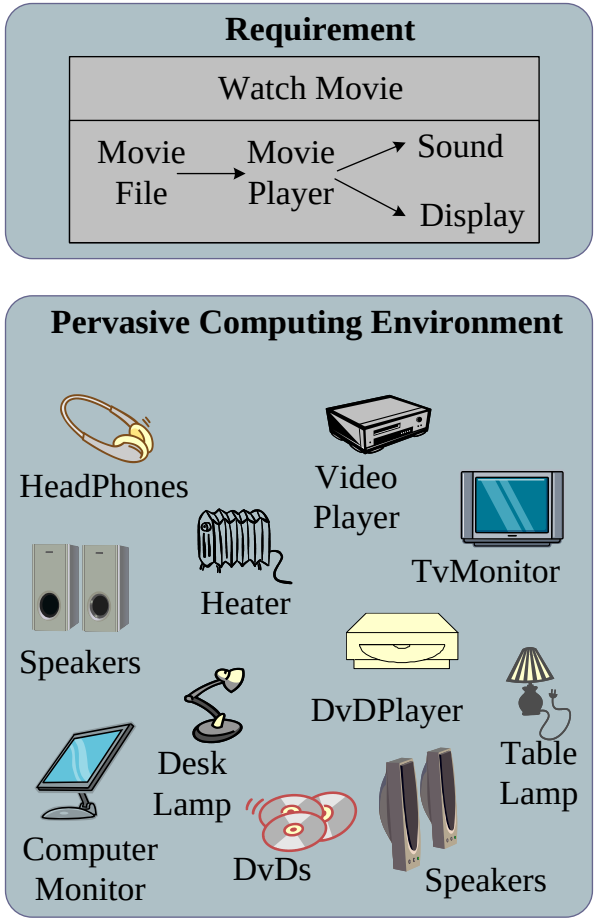
UIO-based PvCEs

- Pervasive Computing Environment
  - A medium that provides user with all the functionality he needs to satisfy his requirements. It is built on physical world with embedded computing devices.
- UIOs
  - Smart objects augmented with various processing capabilities
  - Mirrors can help with shopping
  - Objects at home can serve as interfaces for controlling robots
  - Drinking cup can monitor and remind about healthy drinking habits



Service Composition in PvCEs (1)

- Service
  - A functionality of a computational entity whose execution satisfies the requestor's requirement.
- Service Composition
  - A process of identifying and combining component functionalities to compose a higher level functionality and provide means to perform the requested functionality.



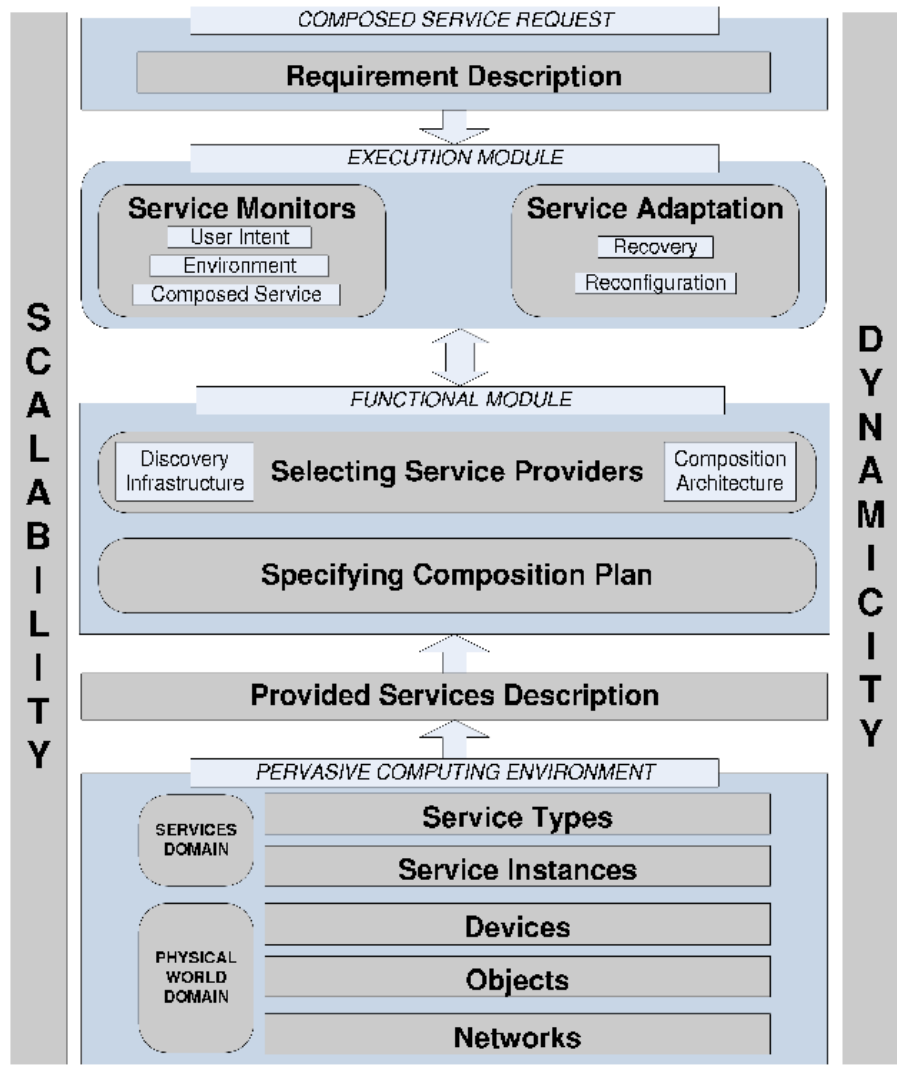
Service Composition in PvCEs (2)

- Characteristics of the UIO-based PvC environment
  - Service requestors require services instantaneously to accomplish their goals
  - Service requestors have no prior knowledge about the available services
  - Service providers join and leave environments at run-time
  - The demand of service requestors changes together with changes in context

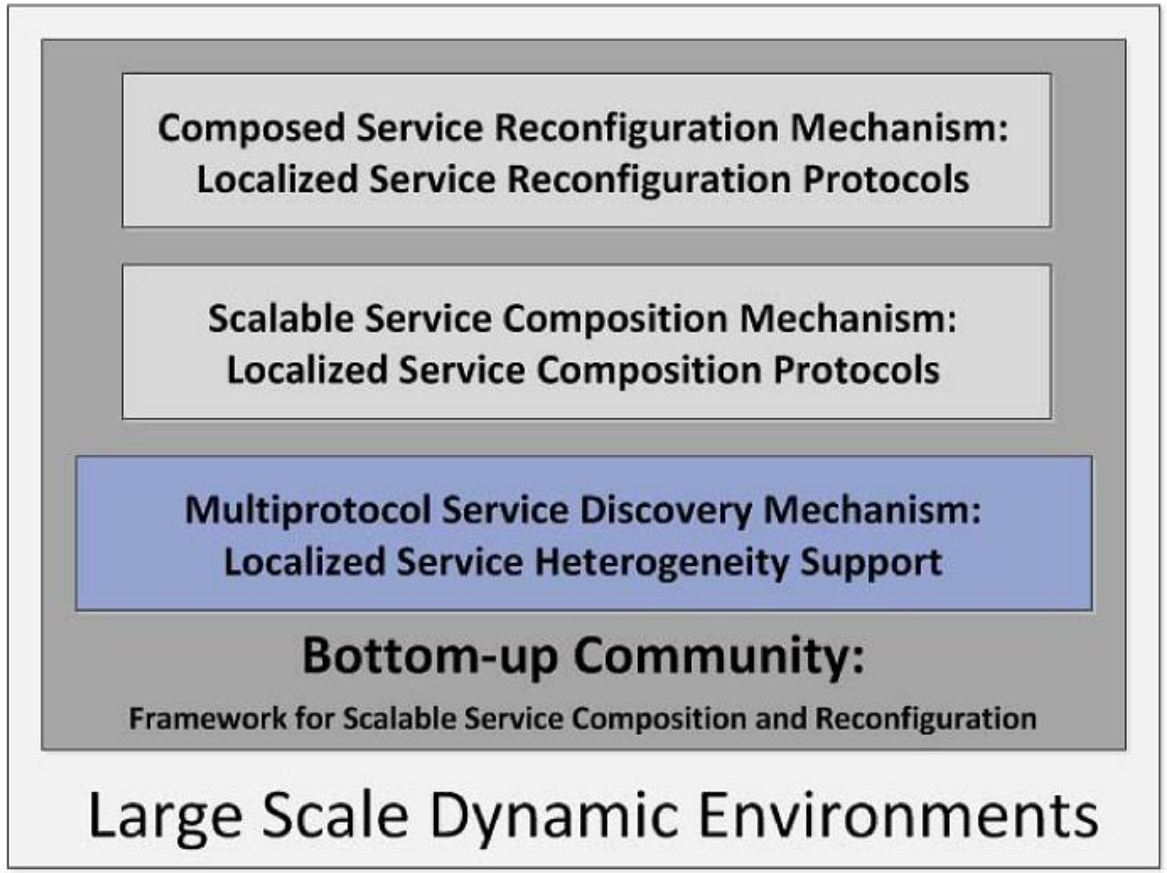
Key Issues for Service Composition and in PvCEs

- Bottom-up UIO collaboration support
  - In UIO based PvCE, there is very large number of resource constrained UIOs and no guarantee of having a central entity capable of managing all other devices.
  - To achieve desirable system behaviors, autonomous interaction of UIOs is required - enable to coordinate the UIOs and make them serve people in a less obtrusive manner.
- Handling changes in environments
  - Assist people with computation support everywhere and all the time.
  - PvCE is extremely dynamic in nature and UIOs frequently join and leave the environment.
- Heterogeneity support
  - Heterogeneity in computing systems will not disappear in the future, but instead will increase as the range of computing devices widens.
  - In UIO based PvCE heterogeneity is particularly high.

Service Composition Framework



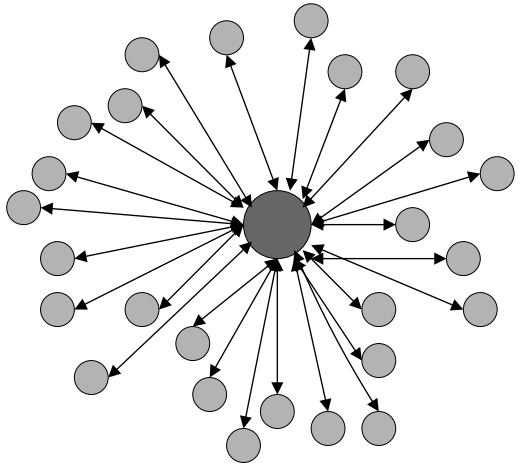
Framework for Scalable Service Composition and Reconfiguration



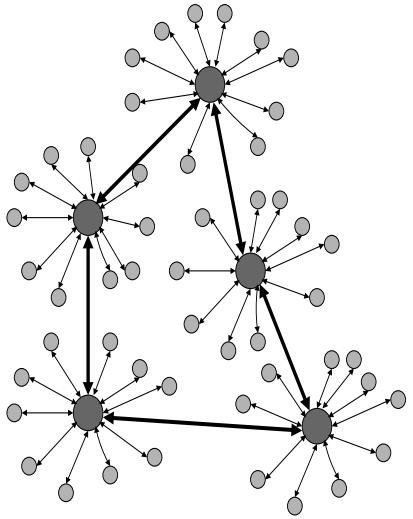
LASEC: Localized Approach to Service Composition

Background

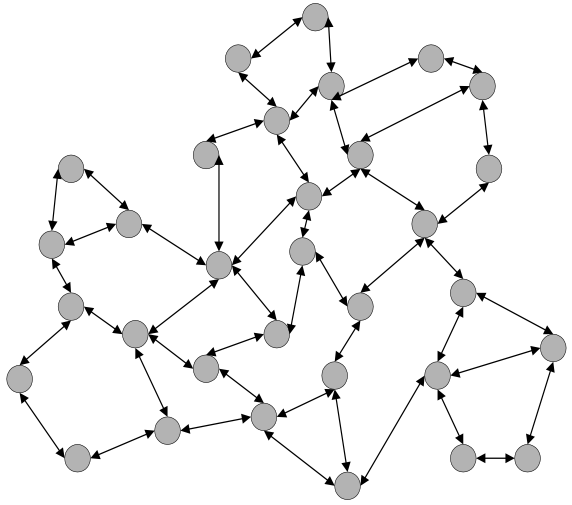
- Existing approaches
  - Centralized - as the service composition environment may be dynamic and large scale, centralized service composition algorithm is usually inefficient due to message cost.
  - Decentralized with coordinators - a decentralized approach, which employs pre-determined coordinators to search and compose service, may have high cost as well, if the coordinators are far from those devices that should be composed.



Centralized



Hybrid



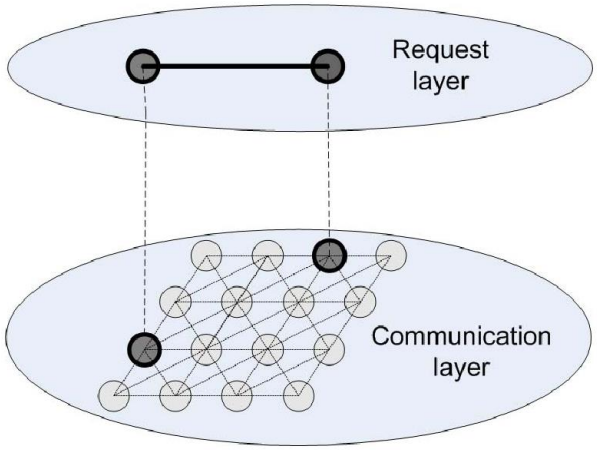
Decentralized

Motivation

- Common feature - globalized nature of algorithms
  - At least one node needs to maintain global information about the current task and about network status.
  - Unsuitable for highly dynamic and large scale environments - huge communication overhead, especially with frequent changes of topology.
  - Global knowledge is not always necessary, since in service composition for PvCE, it is beneficial when the distance between the services used in composition is as small as possible.
- To remedy the problem of maintaining global information, we have opted to adopt a localized approach, which naturally tends to select compositions with high degree of composition locality.

Problem Statement

- Given
  - A set of nodes each providing certain services, the nodes are interconnected and can communicate with each other via wireless network,
- Assume
  - A user specified requested composite service with specified types of requested services and relationships between them.
- Objective
  - Design a localized protocol for the nodes to collaborate in the composition process
- Such that
  - The communication cost is minimized.

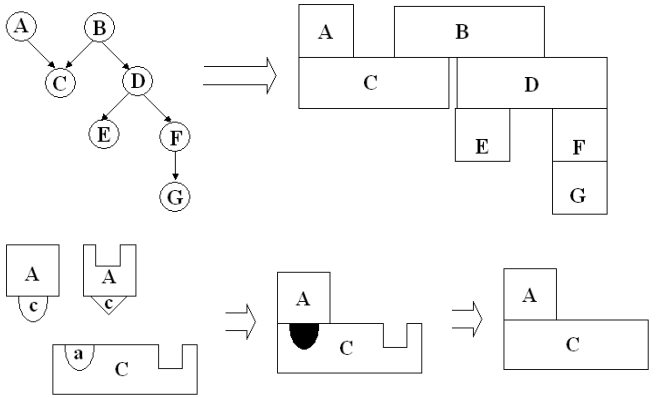


Overview of the Solution (1)

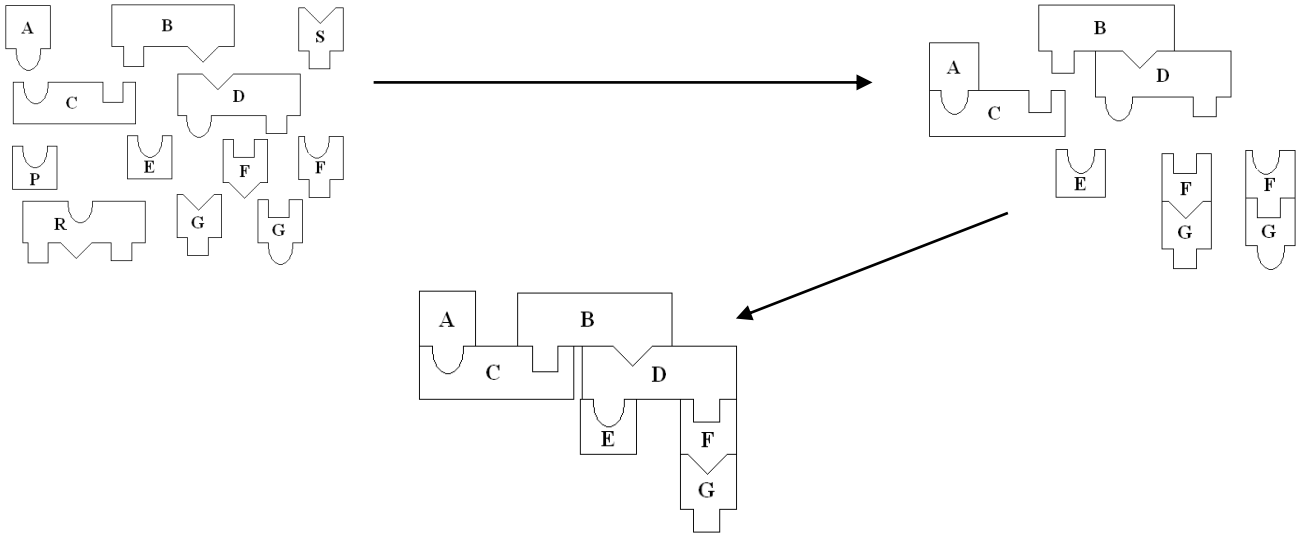
- Requirements for new solution
  - There should be no special entity to manage service composition process
  - Service providers can communicate only with their local neighbours, not all other service providers
  - No service provider knows the full global information or gathers it
  - Initially, service providers are atomic, in sense that they provide some functionality and are not coupled with any other service provider to provide a composed service

Overview of the Solution (2)

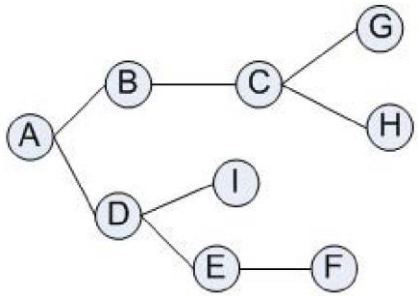
- Inspired by the strategies adopted in solving jigsaw puzzles.

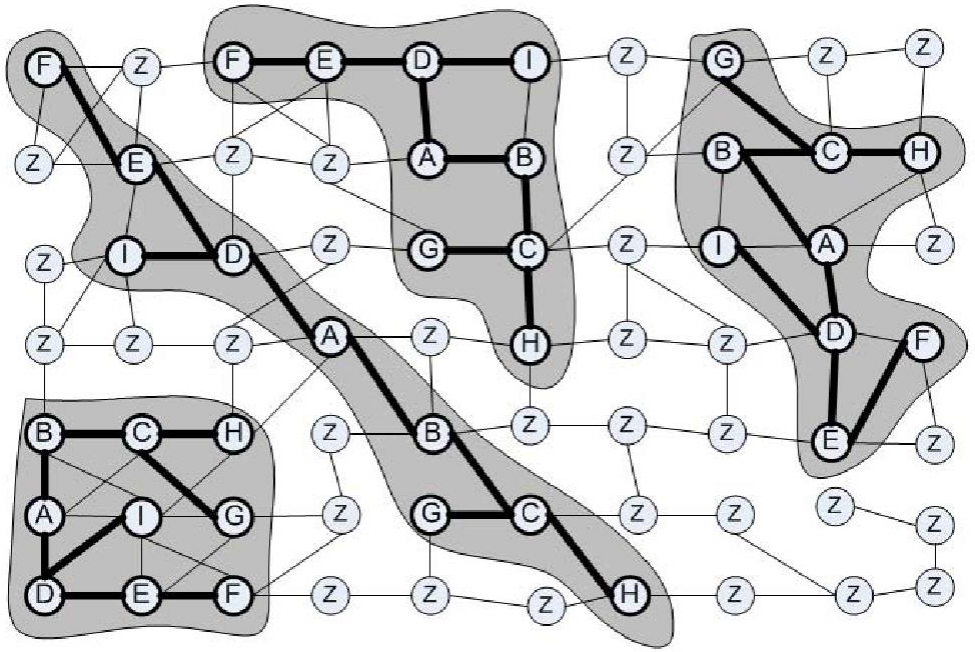


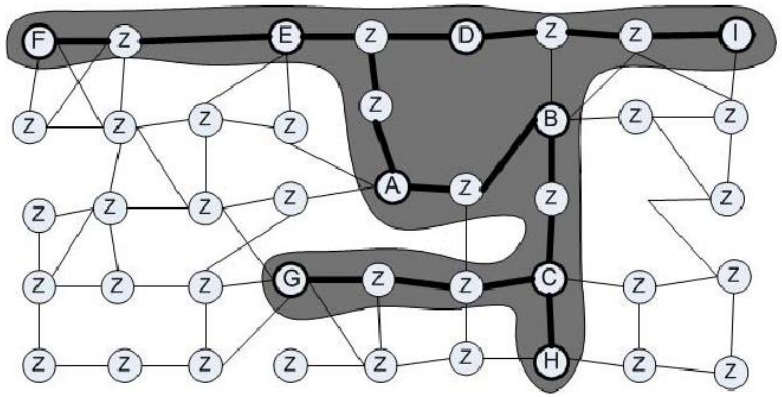
- Forming service sections: Service provider identifies what service type it needs to interlock with through his output and then searches for appropriate service provider with matching input type.
- Growing service: At later stages these sections are joined together with other sections to gradually construct the completed solution.



Overview of the Solution (3)







- We will continue this topic on Wednesday