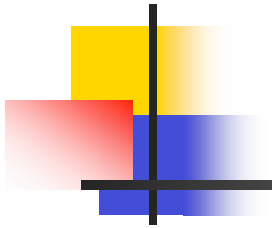
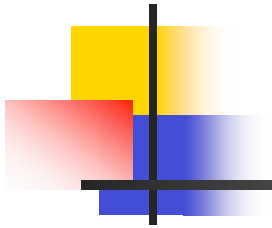


Developments in Web Service Programming

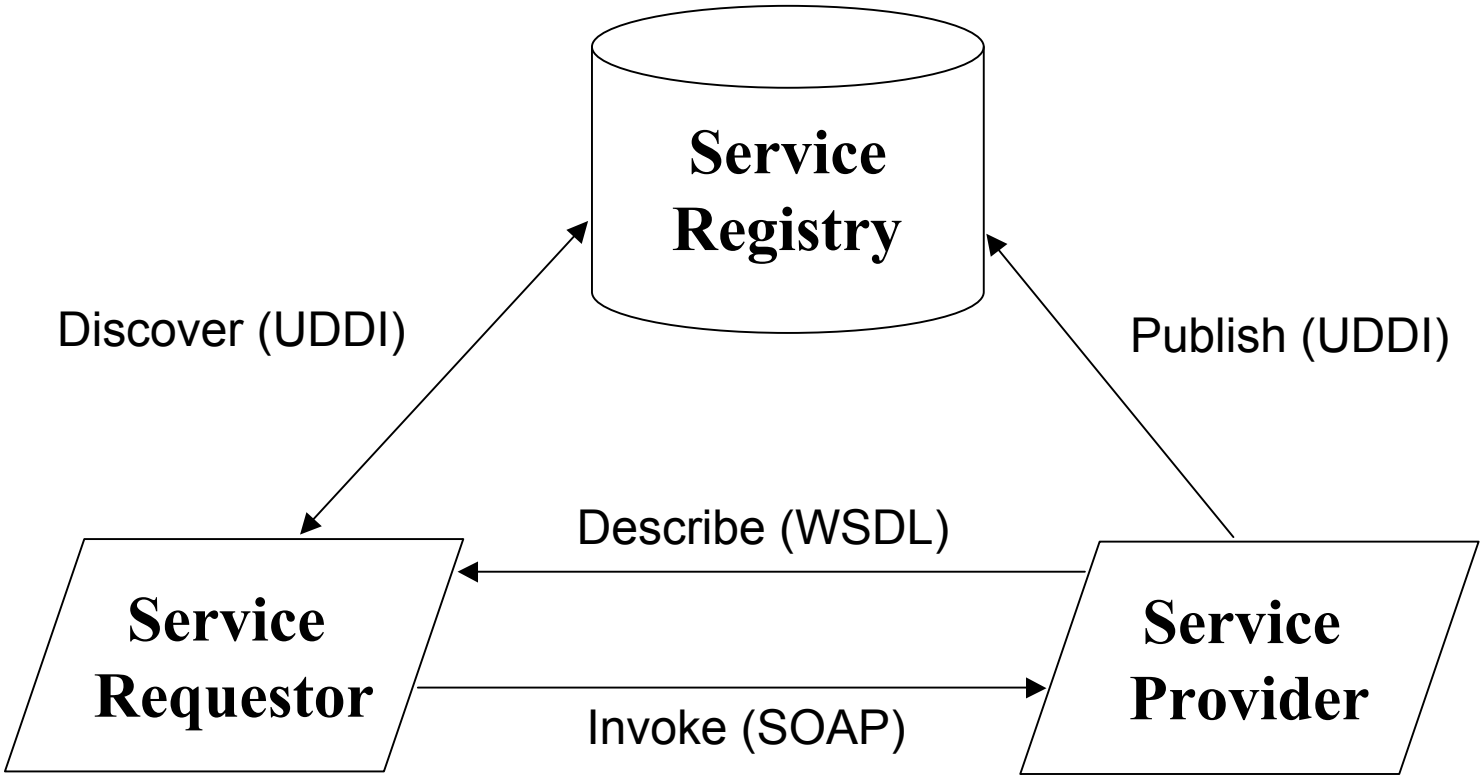
(Week 12 Lecture 2)

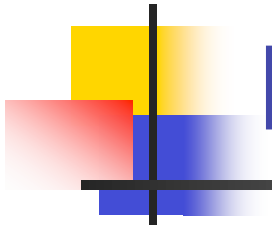


Lecture Outline

- Today's lecture is just a short one to
  - Wrap up the series on Web Services.
  - Look at some current development issues on Web Services.

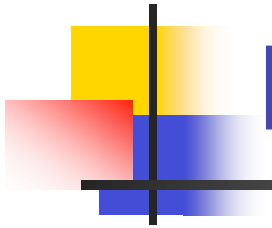
Example Web Service Infrastructure





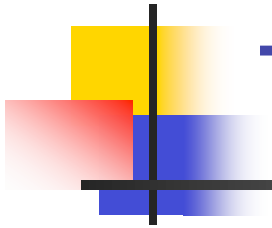
Motivations for Web Services

- Revisiting week 10 lecture 1:
  - Why not stick with normal programs as a basis for providing services?
  - Answers
    - Programs not accessible over the Internet.
    - Must develop for particular OS and in particular languages.
    - No program-to-program communication.



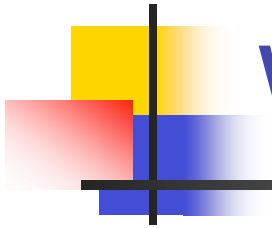
Motivations for Web Services

- Revisiting week 10 lecture 1:
  - Why not stick with current familiar HTTP/HTML/CGI as a basis for providing services?
  - Answers
    - No program-to-program communication.
    - Services are not self-describing.
    - Services are not discoverable.
    - Services are not easily integrated.



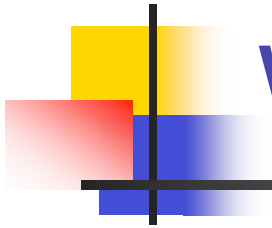
Towards Integration

- The ideal in a network environment is for individual components to be able to take advantage of everything else in the environment.
  - Eg. I want to develop a spread-sheet program that can make use of an on-line currency converter, and automatically store the user data into an intranet database server, and can link to text in a word processor program, etc.



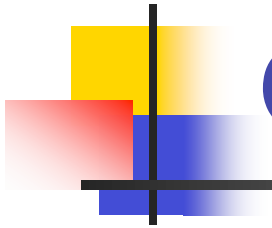
Web Services at W3C

- W3C's current work in Web Services standards concentrating on:
  - XML Protocol
    - Defining the various layers of Web Services, and how they fit together.
  - Refinement of SOAP
    - Security extensions, Attachments, etc.
  - WSDL
    - From Working Draft to full Recommendation.
  - Web Service Choreography
    - Defining how to describe the interactions between service requestor and provider.



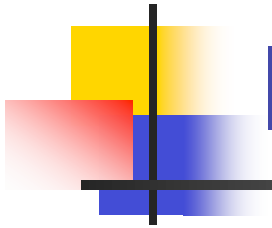
Web Service Developers

- Developers of Web Services are split mainly into two camps:
  - J2EE
  - Microsoft .NET
- However, organizations are hinging their bets on both camps.
  - Recent Gartner Research reports that "more than 90 percent of medium to large organizations that develop applications for their own projects will likely use a mixture of both Microsoft and Java technologies through 2005".



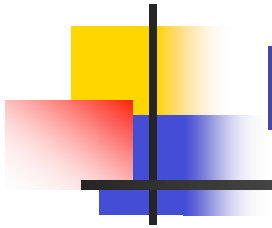
Other Environments

- Also support for Web Services development under various applications paltforms.
  - Eg. Oracle databases, IBM WebSphere, etc.



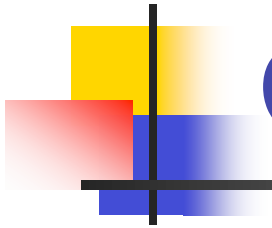
Ideal versus Reality

- The ideals of Web Services is to be platform and environment independent.
  - It shouldn't matter how you developed the service, they will be able to communicate with each other once they are deployed.
  - J2EE and Microsoft .NET, as well as all other development environments, are implement standard SOAP, WSDL, UDDI and XML.



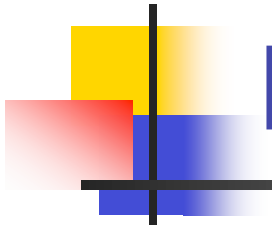
Ideal versus Reality

- However, the reality is that the development platform is critical to due to:
  - Cost of tools
  - Ease and cost of development
  - Ease and cost of deployment
  - Skill set of available personnel
  - etc.



Closing the Gap

- The gap in benefits between the J2EE and .NET are no longer as obvious as it was a few years ago.
  - .NET has matured in terms of it's functionalities and support for large-scale, mission-critical services.
  - Sun's upcoming Java Studio Creator is targeted at providing similar levels of ease of development of Microsoft Visual Studio .NET.
  - Which one is better in **total** cost of development and deployment is disputable.



References

- Current developments in standards and protocols
  - http://www.w3.org/2002/ws/
- J2EE
  - http://java.sun.com/j2ee/
- Microsoft's .NET
  - http://www.microsoft.com/net/
  - http://www.microsoft.com/net/basics/whatis.asp
- More on the concept of Web Services, .NET and J2EE in ICT345 Internet Science and Technology