Web Services & Distributed Objects

Architectures for Decentralized Applications

Panelists: Rohit Khare, Henrik Frystyk Nielsen, & Mark Thomsen

ISR Research Forum
17 June 2003
UC, Irvine

Panelists

- **Rohit Khare**
  - ISR & KnowNow, Inc.
- **Henrik Frystyk Nielsen**
  - Microsoft, Inc.
- **Mark Thomsen**
  - Alodar Systems

A Tale of Two Technologies

- **Distributed Objects (CORBA)**
  - Component-oriented
  - Application-Programming Interfaces
- **Web Services (SOAP)**
  - “Service-oriented”
  - Application-Layer Protocol interfaces

Distribution vs. Decentralization

- **Managing inventory at multiple outlets**
  - Enterprise Resource Planning (ERP)
- **Merging with another retailer…**
  - Enterprise Application Integration (EAI)
- **Integrating directly with suppliers…**
  - Exchanges, Trading Webs, Directories, ???

Panel Discussion

1. Are these different technologies?

- Are protocols better than APIs? If so, how and why?
- What, exactly, is a service? Why is it so hyped?

2. Are these different problems?

- Is decentralization a genuine problem today? in the future?

3. How to choose the right hammer.

- What features of each technology support decentralization?
- Does it really matter which you choose, or it just a detail?

My Position

1. Protocols are better than APIs because they are designed with outsiders in mind.
2. A service is merely a component operated for others.
3. There are risks latent in misapplying distributed technology to a decentralized application domain.
4. There are fundamental advantages of Web Services over Dist. Objects for Internet-scale integration:
   - Better able to cope with network failures.
   - More likely to expose dependencies explicitly.