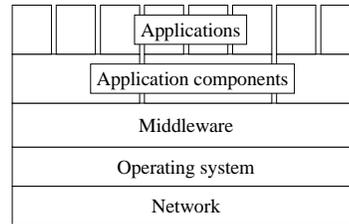


## Chapter 16

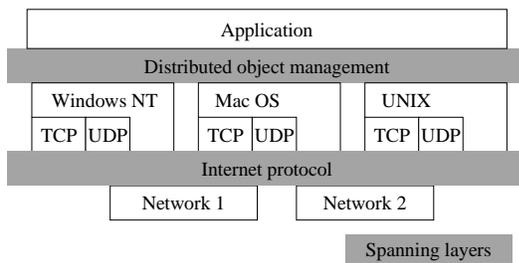
by  
David G. Messerschmitt

### Layering a computing infrastructure



2

### Spanning layer



3

### The new infrastructure: middleware

- Layer of software between OS and application
  - Hides heterogeneity
  - Provides generic common services
  - Increases level of abstraction
- By its nature, not single platform, nor bundled with equipment
  - Microsoft attempts to be an exception

4

### The new infrastructure

- Middleware is
  - where new capabilities are added by layering
  - where much of the experimentation and innovation happens in the infrastructure
  - where the successful approaches have a chance to become a spanning layer and/or integrated into a distributed OS
- Boundary to a distributed OS is fuzzy

5

### Middleware objectives

- Hide heterogeneity
- Location independence
- Common functionality needed by many applications
- Software portability and mobile code
- Help integrate legacy facilities
- Aid application interoperability
- Aid scalability

6

### Some middleware categories

- Transaction processing
  - Simplify the coordination of complementary resource managers
- Message-oriented middleware
  - Support message and queuing capabilities where resource managers are not available simultaneously (like workflow)

7

### Some middleware categories

- Distributed object management
  - Support applications that are distributed across heterogeneous platforms and organizations
- Mobile code
  - Allow application code to be moved and executed on heterogeneous platforms
  - Without prior software installation

8

### Infrastructure software today

- With networks, new emphasis on:
  - Portability: applications run across multiple platforms (avoid lock-in)
  - Interoperability: pieces of application must work together (benefit from network effects)

What are some examples of each?

9

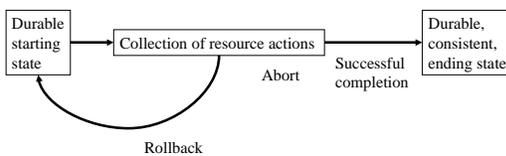
Understanding Networked Applications:  
A First Course

---

## Transaction processing

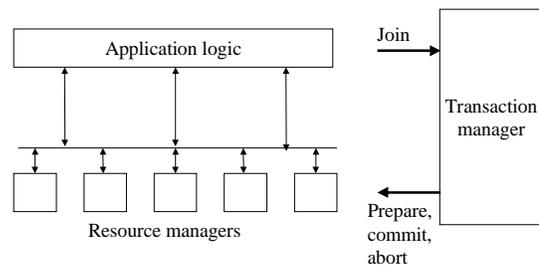
by  
David G. Messerschmitt

### The transaction



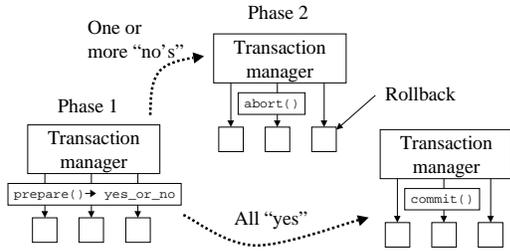
11

### Transaction architecture



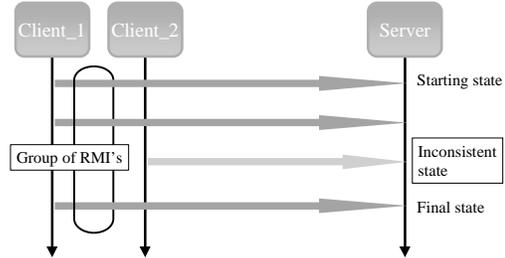
12

### Commit or abort



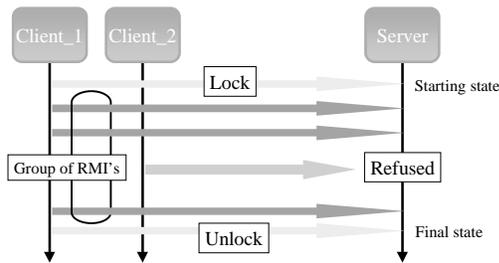
13

### Atomic series of resource actions



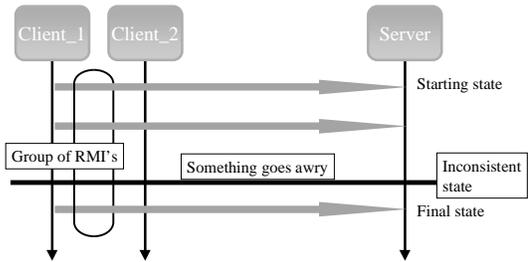
14

### Locking to prevent conflicts



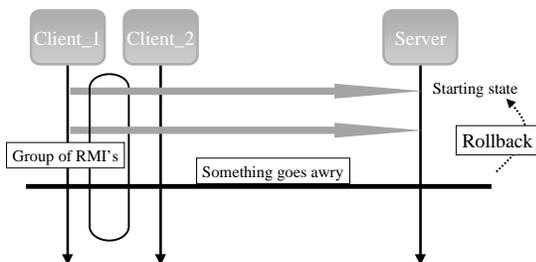
15

### Abort



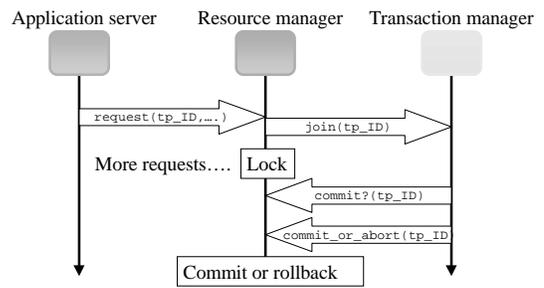
16

### Rollback



17

### Transaction protocol



18

## Mobile code and Java

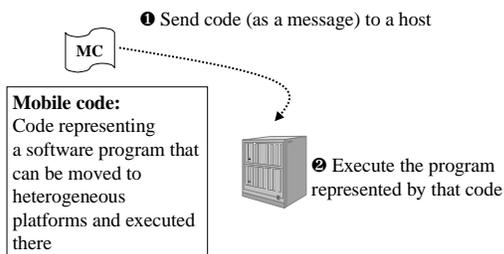
by  
David G. Messerschmitt

### Reminder: two key requirements

- With networks, new emphasis on:
  - Portability: applications run across multiple platforms (avoid lock-in)
  - Interoperability: pieces of application must work together (benefit from network effects)

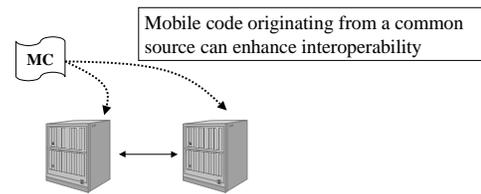
20

### Dynamic portability: mobile code



21

### Portability can aid interoperability



22

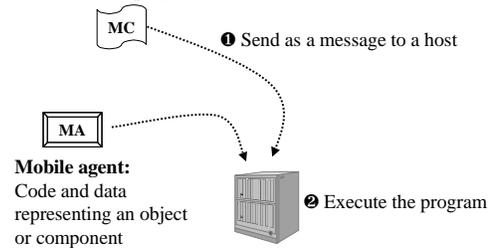
## Java

- Portability
  - “Write once, run anywhere”
- Programming productivity
  - Garbage collection (no memory leaks)
  - Multi-threaded
- Scalability
  - Move execution cycles
- Interoperability
  - Software components come from common repository

23

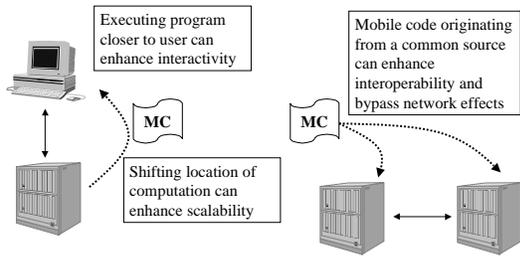
**Mobile code:**  
Code representing  
a software program

**Idea of mobile code**



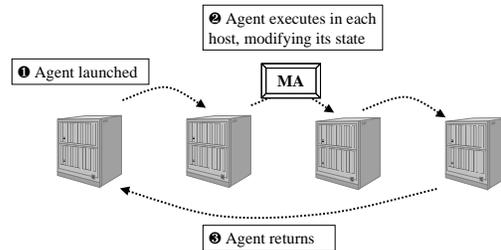
24

### Some mobile code advantages



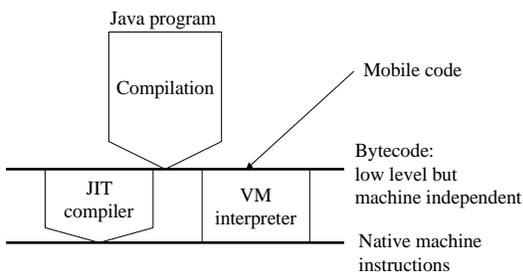
25

### Idea of mobile agents



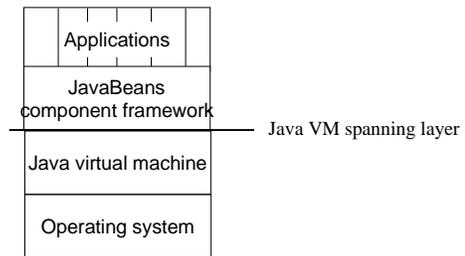
26

### Java virtual machine



27

### VM as spanning layer



28

### SUN/Java strategy

- License Java freely, even to rival Microsoft
  - Why?
- License terms give Sun a modicum of control over the “standard”
  - Why?
- How does Sun expect to make money?

29

Understanding Networked Applications:  
A First Course

### Distributed object management

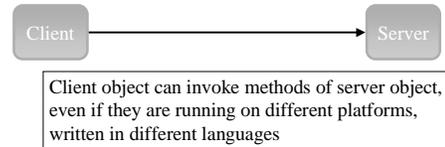
by  
David G. Messerschmitt

## Distributed object management

- Emphasis is on interoperability
  - Allows objects on one host to invoke methods of objects on another host
  - Platform, language independent
- CORBA vs DCOM
- Portability is not the emphasis

31

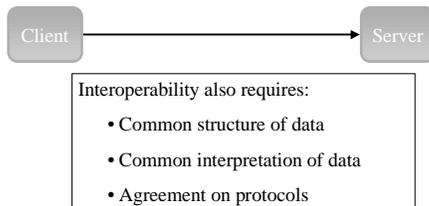
## Interoperability



What else is needed?

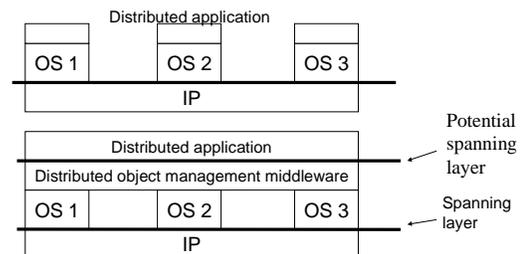
32

## Interoperability



33

## Before and after



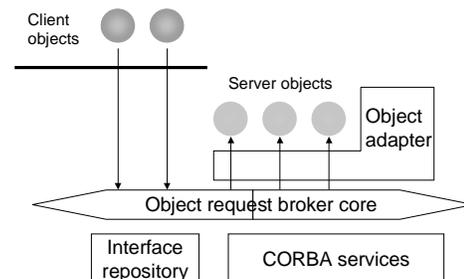
34

## What is the acronym?

- Common Object Request Broker Architecture
- or
- Concerned Off-Road Bicyclist Association?

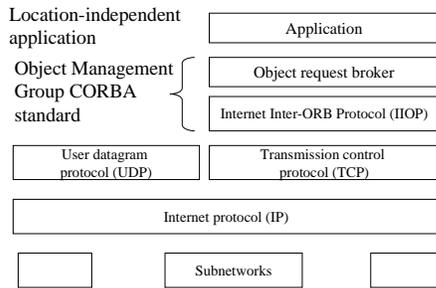
35

## CORBA architecture



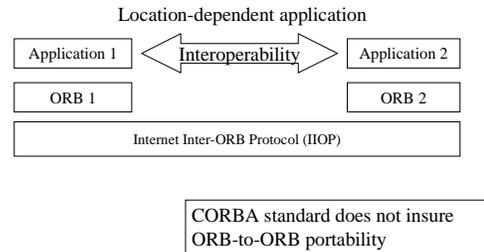
36

### Protocol layer



37

### Portability not promised



38

### OMG process

- Identify need
- Request for proposals
- Process to
  - choose best
  - or ask proposal advocates to work together

39

### Which is most effective?

- Industry de facto standard effort (CORBA) or
- Single vendor integrated solution (DCOM)?

40

### CORBA vs DCOM

CORBA	DCOM
<ul style="list-style-type: none"> <li>• Integrate best ideas</li> <li>• Multi-vendor support</li> <li>• Cross-platform and language</li> </ul>	<ul style="list-style-type: none"> <li>• Fast, no consensus required</li> <li>• No vendor interoperability issues</li> </ul>

41

### Two methods for application interaction

- CORBA and DCOM
- Exchange documents (XML)

What are their relative merits?

42

## CORBA vs XML

CORBA	XML
<ul style="list-style-type: none"> <li>• Natural OOP extension</li> <li>• No document interpretation</li> <li>• Good for back-and-forth protocols</li> </ul>	<ul style="list-style-type: none"> <li>• Flexible data-sharing</li> <li>• Good for document-like objects</li> <li>• Natural cross-platform capability</li> <li>• No protocol standardization</li> </ul>
<div style="border: 1px solid black; padding: 2px;">Both have need for standardization of data or document interpretation</div>	
43	44

## What CORBA provides

- Java
  - Language bindings
  - Transportable objects
- “Inter-galactic” software bus
  - Cross-platform and language
  - Interoperability (but not portability)
- High levels of abstraction
  - Remote method invocation on objects

## What CORBA provides (con’t)

- Run-time flexibility
  - Everything self-describing
  - Interface discovery
  - Dynamic data structures and binding
- Useful services
  - Naming
  - Security
  - Many others

45

## Importance of CORBA

- Inter-enterprise computing
  - Platform and language independence
  - Electronic commerce, network management, etc
- Reduction of network effects
  - Another spanning layer
  - Significance of platform reduced

46

## Are Java and CORBA competitive or complementary?

- Both offer interoperability across different platforms
- Java offers portability and transportability
- CORBA offers heterogeneous language bindings
- CORBA offers many services, metadata, etc.
- Bottom line: they are complementary!
  - (but some Java proponents may not agree)

47

## Who favors what?

Would these vendors be in favor of:

- Interoperability?
- Portability?

Microsoft  
Intel  
SUN  
Novell  
Iona

48