Sistemi Distribuiti
Corso di Laurea in Ingegneria

Prof. Paolo Nesi

PARTE 10: DCE-COM-DCOM

Department of Systems and Informatics

University of Florence

Via S. Marta 3, 50139, Firenze, Italy

tel: +39-055-4796523, fax: +39-055-4796363

Lab: DISIT, Sistemi Distribuiti e Tecnologie Internet

nesi@ingfi1.ing.unifi.it, nesi@dsi.unifi.it nesi@computer.org

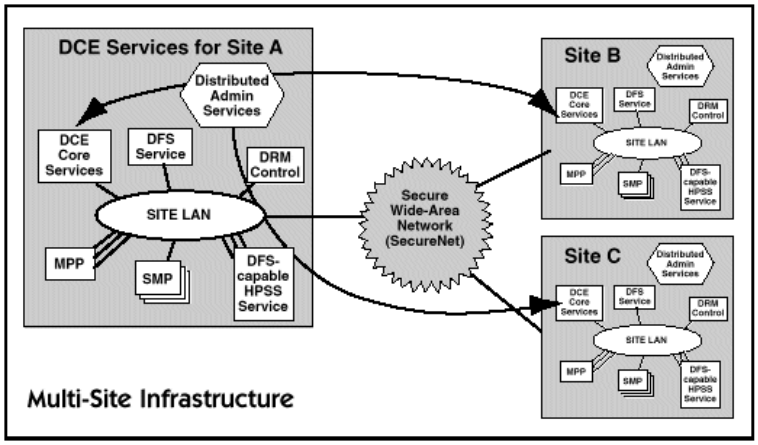
www: http://www.dsi.unifi.it/~nesi

DCE

- Message Passing/Queuing

Distributed components architectures for web application

DCE (Distributed Computing Environment)



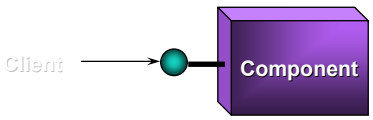
Support only C/C++ languages

COM

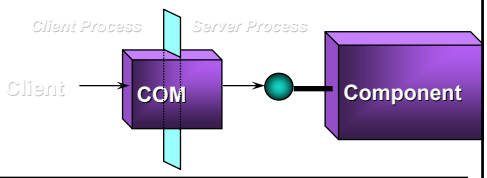
- Questioni di base COM: classi e server
- Interfacce di COM
- Interazione con COM

Accessing COM Services

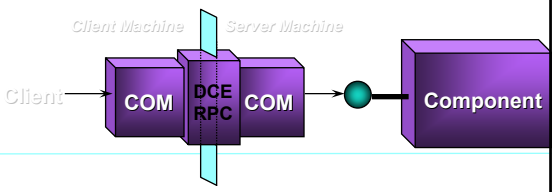
In the same process
Fast, direct function calls



On the same machine
Fast, secure IPC



Across machines
Secure, reliable and flexible DCE-RPC based *DCOM* protocol



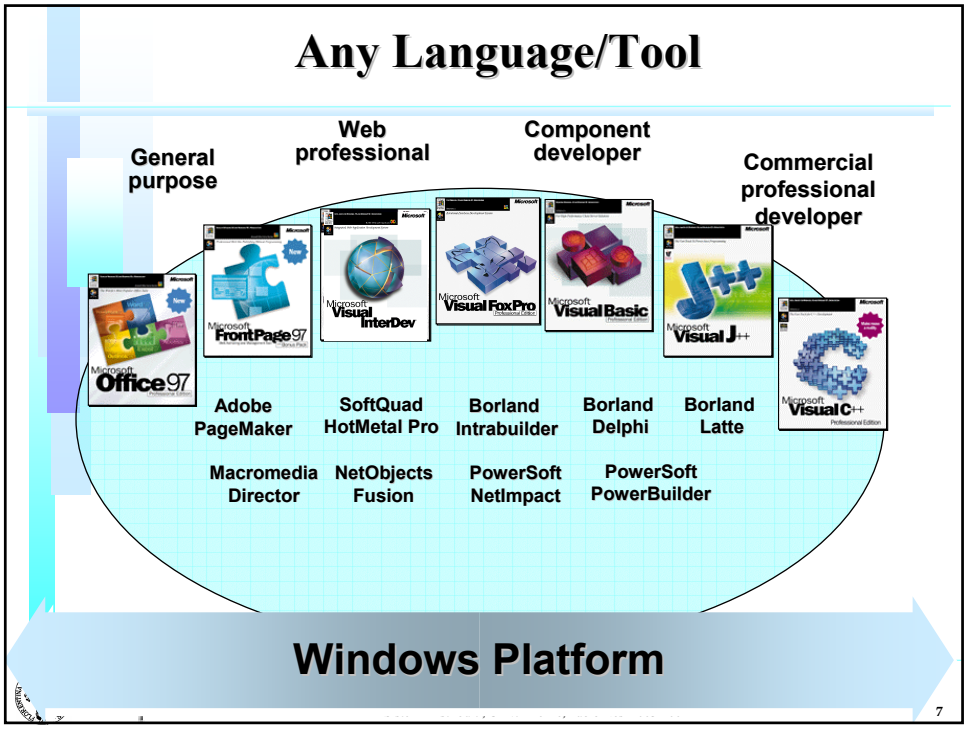
Major COM services

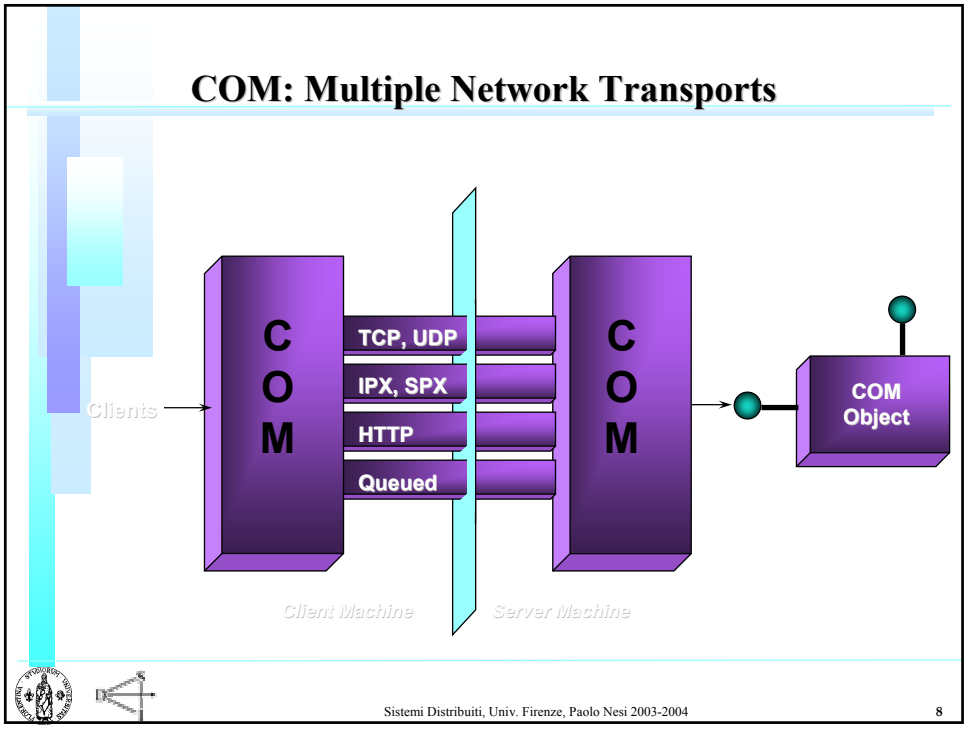
Shipping now:

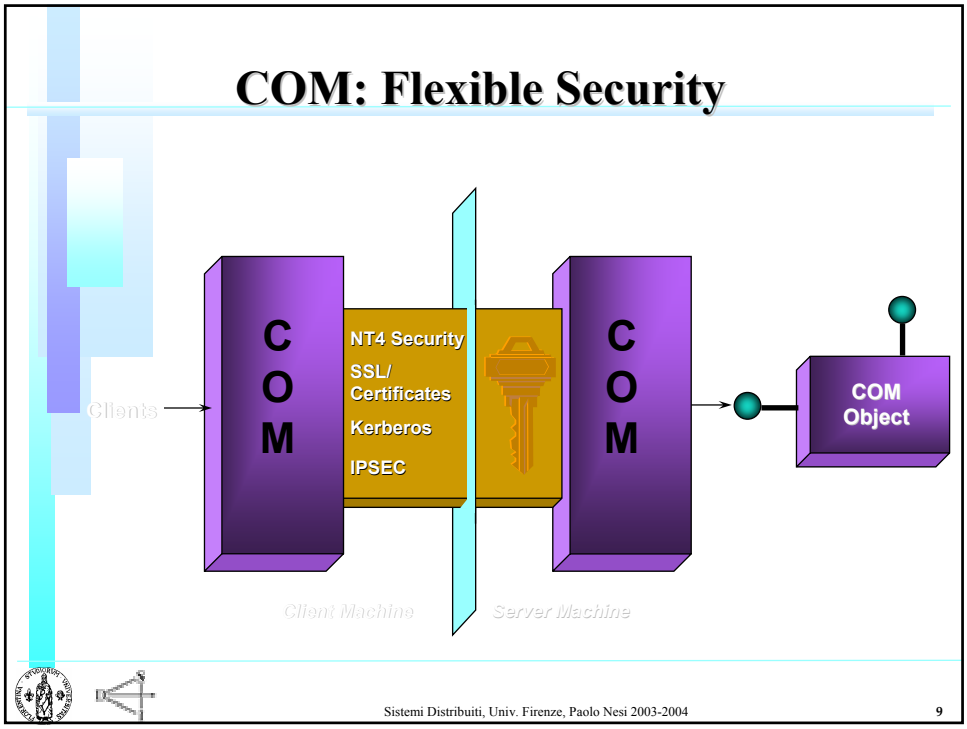
- Security
- Lifecycle management
- Type information
- Monikers (Naming)
- Database access
- Data Transfer
- Components
- Transactions
- Asynchronous communications
- Registry
- Automation (Dynamic Invocation)

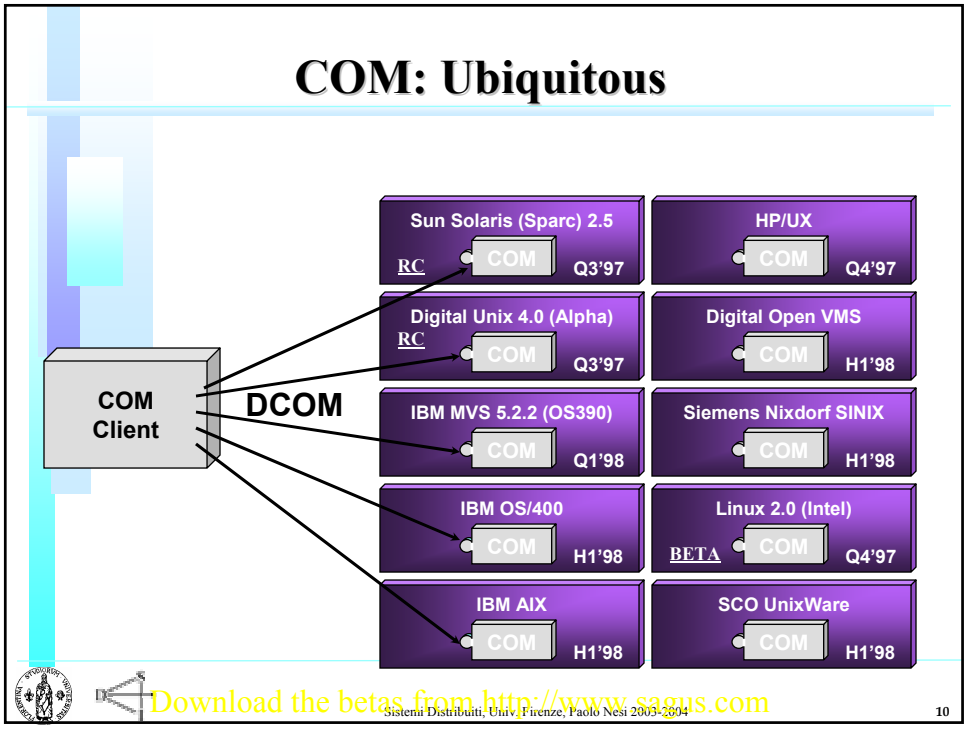
Shipping soon:

- Directory (NT 5.0)


Any Language/Tool
General purpose
Web professional
Component developer
Commercial professional developer
Office 97
FrontPage 97
Visual InterDev
Visual FoxPro
Visual Basic
Visual J++
Visual C++
Adobe PageMaker
SoftQuad HotMetal Pro
Borland Intrabuilder
Borland Delphi
Borland Latte
Macromedia Director
NetObjects Fusion
PowerSoft NetImpact
PowerSoft PowerBuilder
Windows Platform
7


COM: Multiple Network Transports
Clients
COM
TCP, UDP
IPX, SPX
HTTP
Queued
COM
COM Object
Client Machine
Server Machine
Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2003-2004
8


COM: Flexible Security
Clients
C O M
NT4 Security
SSL/ Certificates
Kerberos
IPSEC
C O M
COM Object
Client Machine
Server Machine
Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2003-2004
9

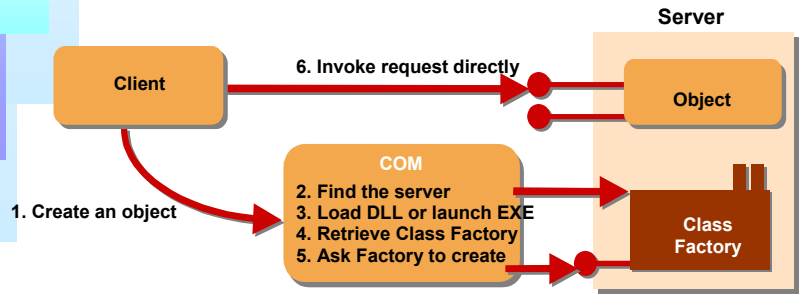

COM: Ubiquitous
COM Client
DCOM
Sun Solaris (Sparc) 2.5
RC
COM
Q3'97
HP/UX
COM
Q4'97
Digital Unix 4.0 (Alpha)
RC
COM
Q3'97
Digital Open VMS
COM
H1'98
IBM MVS 5.2.2 (OS390)
COM
Q1'98
Siemens Nixdorf SINIX
COM
H1'98
IBM OS/400
COM
H1'98
Linux 2.0 (Intel)
BETA
COM
Q4'97
IBM AIX
COM
H1'98
SCO UnixWare
COM
H1'98
Download the betas from http://www.sagus.com
Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2003-2004
10

COM Classes and Servers

- COM class: body of source code that implements COM interfaces
- provides real functions in any supported programming language for each interface method it supports
- each COM class has a unique identifier (CLSID)
- client ask COM to create an object and return interface pointer
- client applications interact with COM objects through interface pointers

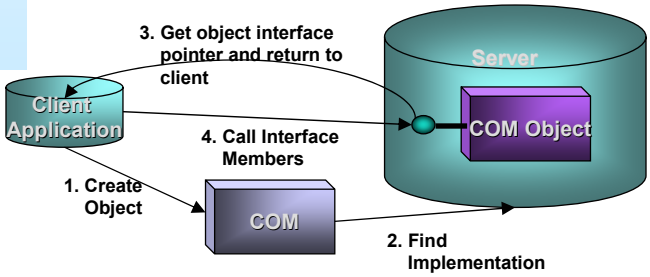

- client not dependent on implementation details of COM
- COM servers:
  - in-process server: DLL loaded into client process calls go directly to object created in the client's process
  - out-of-process server: separate executable, either on same machine as a client or on remote machine; calls go first to an in-process proxy which uses RPC; in the server, stub object receives each incoming call and dispatches to appropriate COM object
- ActiveX control is in-process COM server object


Object Creation

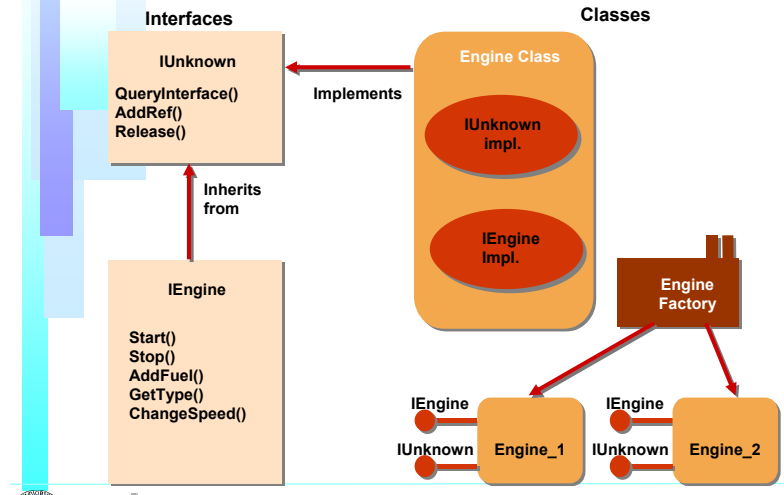


Object and Component Middleware

- COM
  - The client connects to the object through the COM server, transparently.



Interfaces, Classes and Factories




COM Interfaces

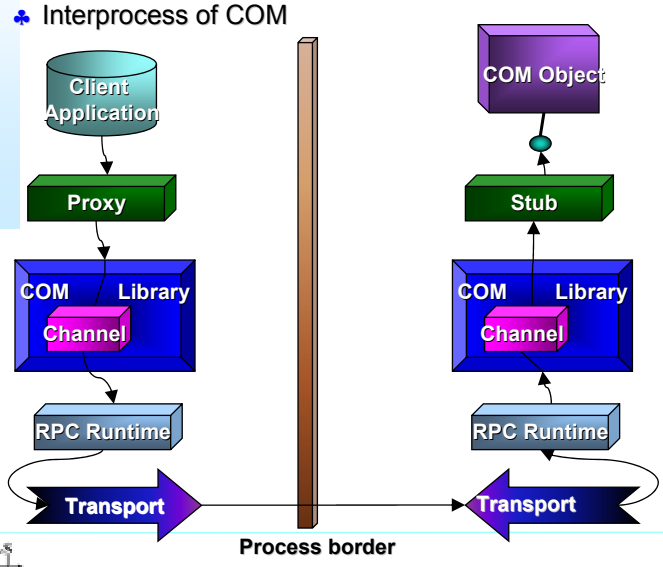
- COM interface defines behavior or capabilities of software component as a set of methods and properties
- interface is contract that guarantees consistent semantics
- each COM object must support at least one interface (IUnknown)


COM pros/cons

- PROs
  - Access to OS functionality
  - Faster and easier to write apps
  - Third-party COM components
- CONs
  - Requires infrastructure and tools
  - Client/server kept separate (e.g. different strings implementations)
  - DLL hell


Object and Component Middleware

- Interprocess of COM




DCOM

- Questioni di base DCOM
- Interfacce di DCOM
- Architetture di DCOM
- DCOM e Active X
- CORBA e COM


DCOM

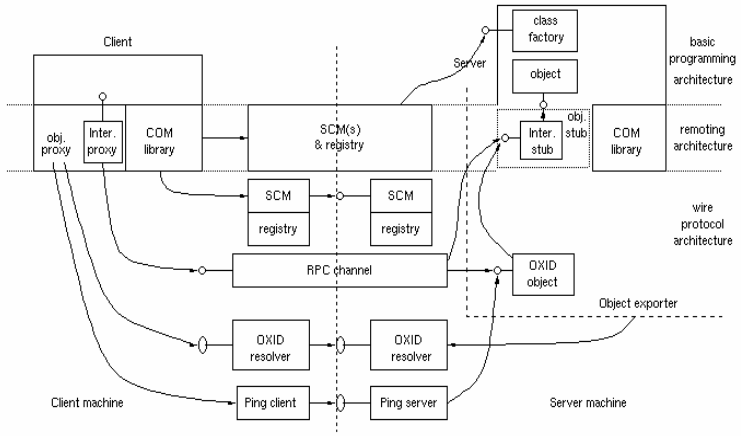
- DCOM distributed extension to COM
- builds an ORPC layer on top of DCE RPC
- COM server can create object instances of multiple object classes
- COM object supports multiple interfaces, representing different view or behavior of the object
- interface consists of a set of functionally related methods


DCOM Interfaces

- interfaces described using MIDL
- MIDL compiler generates proxy and stub code in C or C++ from interface definition
- generated proxy code provides client-side API
- stub objects decode incoming client requests and deliver to appropriate object in the server


- COM client interacts with COM object by acquiring a pointer to an object's interface and invoking methods through that pointer, as if the object resides in the client's address space
- interface follow standard memory layout, same as C++ vtable
- specification at binary level
- integration of binary components in different languages (C++, Java, Visual Basic)


DCOM Architecture



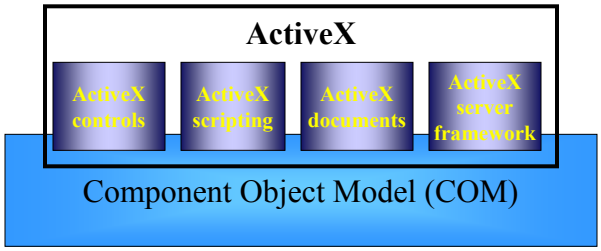

DCOM Overview

- proxy and stub code interact with appropriate runtime libraries to exchange requests and responses
- each interface has UUID
- QueryInterface method of IUnknown
- QueryInterface returns an interface pointer
- interface pointer points to COM binary data structure
- client application must know CLSID and IID for an interface
- standard dictates interface functions calling conventions

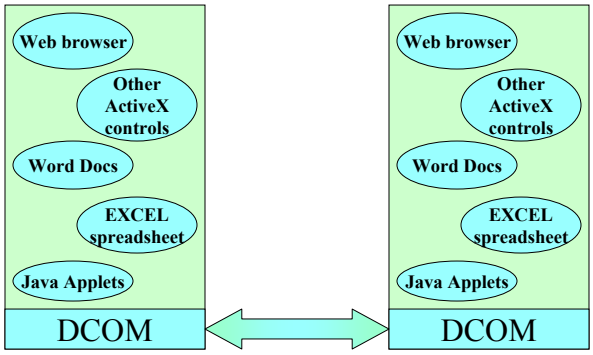

Distributed components architectures for web application
DCOM/ActiveX

Can be viewed as an integration technology (inter-operability standard) that permits different application software components to interact with each other as part of an integrated, interactive web application, including Java Applets. DCOM (Distributed Component Object Model) is a set of Microsoft concepts and program interfaces in which client program objects can request services from server program objects on other computers in a network. While it is similar to CORBA, it only operates between Microsoft programs.

ActiveX major components:



Distributed components architectures for web application
DCOM/ActiveX



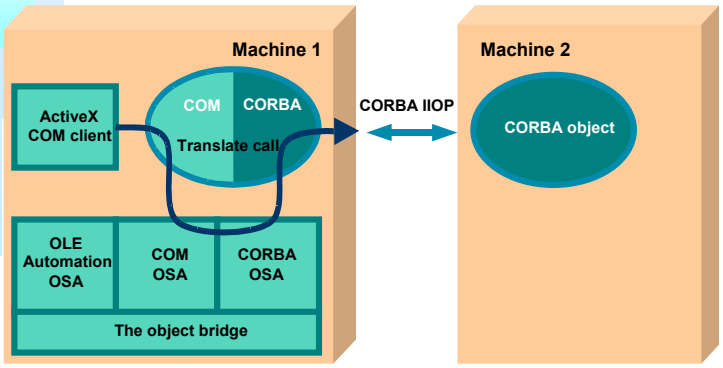
Client Workstation (Active desktop)

Server machine (Active server)

CORBA / COM interoperability

- Naming of communication endpoints:
  - CORBA: Interoperable Object Reference
  - DCOM: OBJREFs (include reference counting)
- Support for multiple interfaces (only in DCOM)
- Format of payload parameter values:
  - DCOM: Network Data Representation
  - CORBA: Common Data Representation


COM-CORBA Interoperation




CORBA and DCOM limitations

- DCOM platform limitation
- CORBA, subtle incompatibilities require ORB from same vendor
- Reliance on closely administered environments
  - IIOP must cross firewalls
- Programming difficulties in data alignment and data types


- http://www.microsoft.com/com/tech/com.asp

THE site for COM (and the new COM+)

- http://www.microsoft.com/Com/news/drgui.asp

  - A tutorial about COM, DCOM and COM+

- http://www.microsoft.com/com/presentations/

  - Some presentations from Microsoft regarding COM

- http://www.microsoft.com/com/presentations/realtour.zip

- This presentation gives an overview of COM and its features. Also included are some quotes from the press showing that COM is preferred over CORBA and IIOP