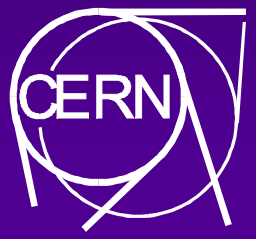


(D)COM
Microsoft's response to CORBA

Alessandro RISSO - PS/CO

Talk Outline

- DCOM
  - What is DCOM ?
  - COM Components
  - COM Library
  - Transport Protocols, Security & Platforms Availability
- Services Based on DCOM
- DCOM - Pros & Cons
- DCOM & CORBA - A Comparison

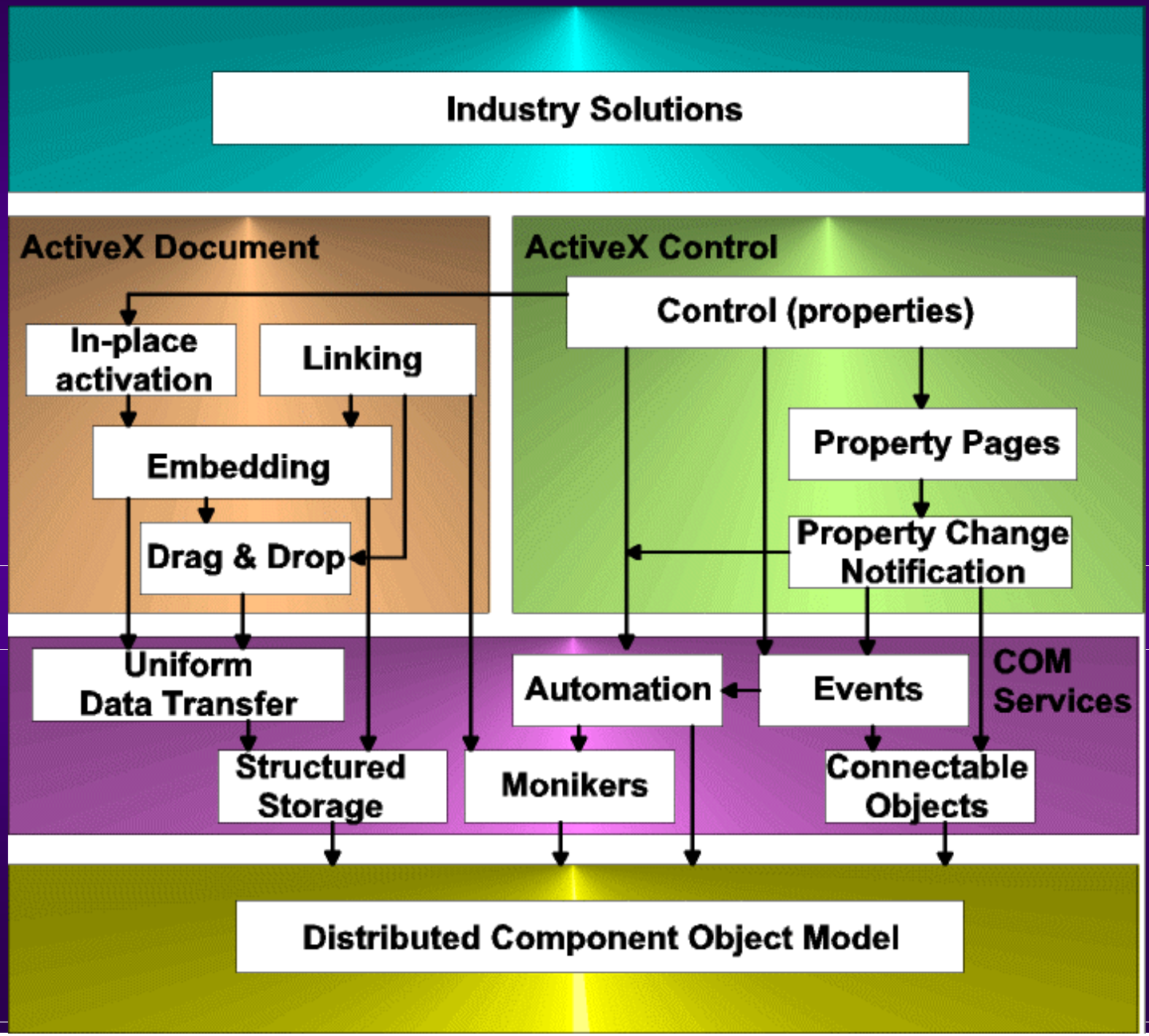
The Foundation

- COM is Microsoft's Component Model
  - COM defines a Binary Standard for Components Interoperability
  - COM Component is a piece of compiled code that provides some service to the rest of the system
- DCOM provides Distributed Services to COM
  - allows COM Components to transparently communicate across the network

Evolution of COM

- **1993** - OLE2 introduced COM
- **1994** - COM in Windows NT 3.5
- **1996** - DCOM in Windows NT 4
- **1996** - ActiveX, COM enabled for the Internet
- **1996** - Microsoft Transaction Server (MTS)
- **1998** - DCOM for W95, Sun Solaris, HP-UX, IBM AIX , …
- **200x** - COM+

COM Architecture

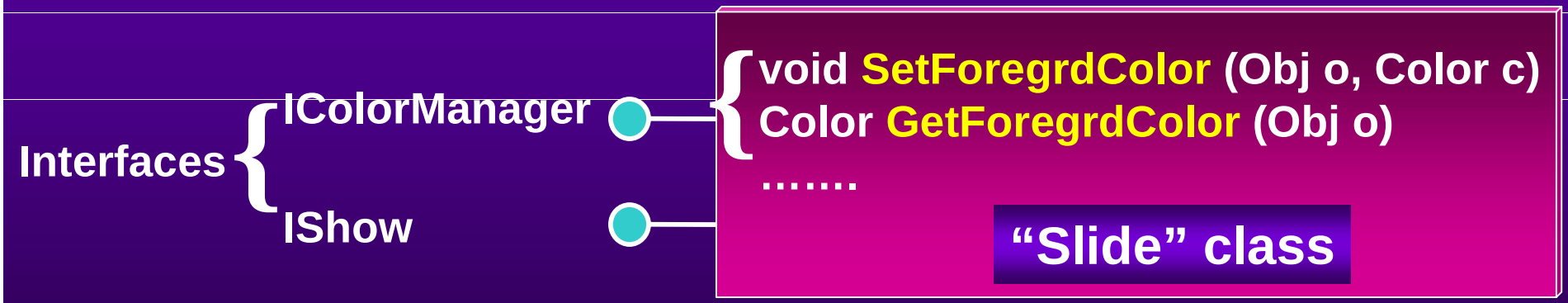


COM Components

- are language-neutral encapsulated objects
- expose a set of structured interfaces
- enable other applications or components to communicate in a standard way

COM Class

- named, concrete implementation of one or more interfaces
- identified by a class identifier - CLSID



CLSID = {55ca6522-e33e-22cf-55da-0091e2a3a124}

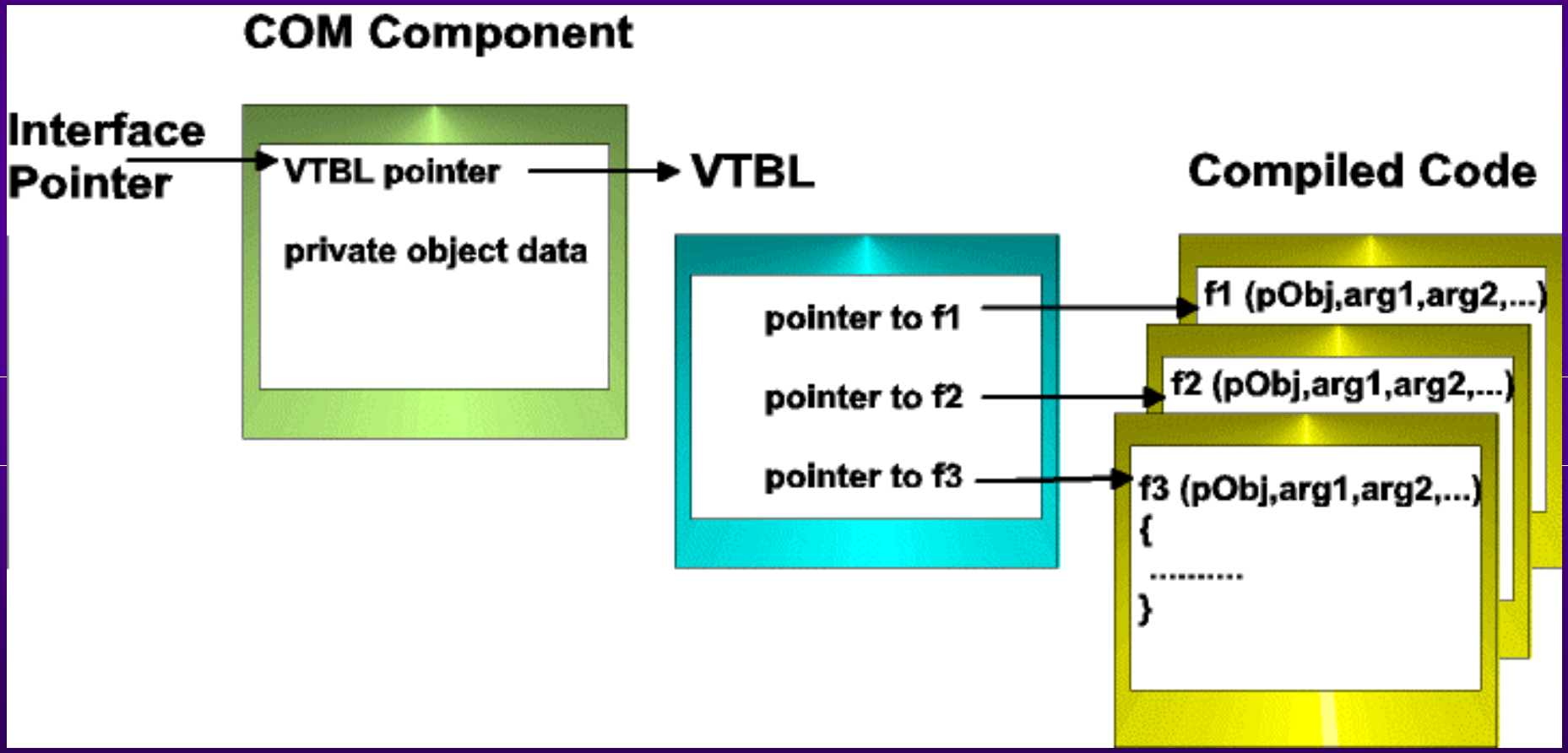
Interface

- a group of related functions that specifies a contract - name, interface signature and semantics
- denotes behavior only, not state
- are immutable - never versioned
- are strongly typed - unique identifier IID
- interfaces are described in MS-IDL - derived from the IDL of OSF/DCE

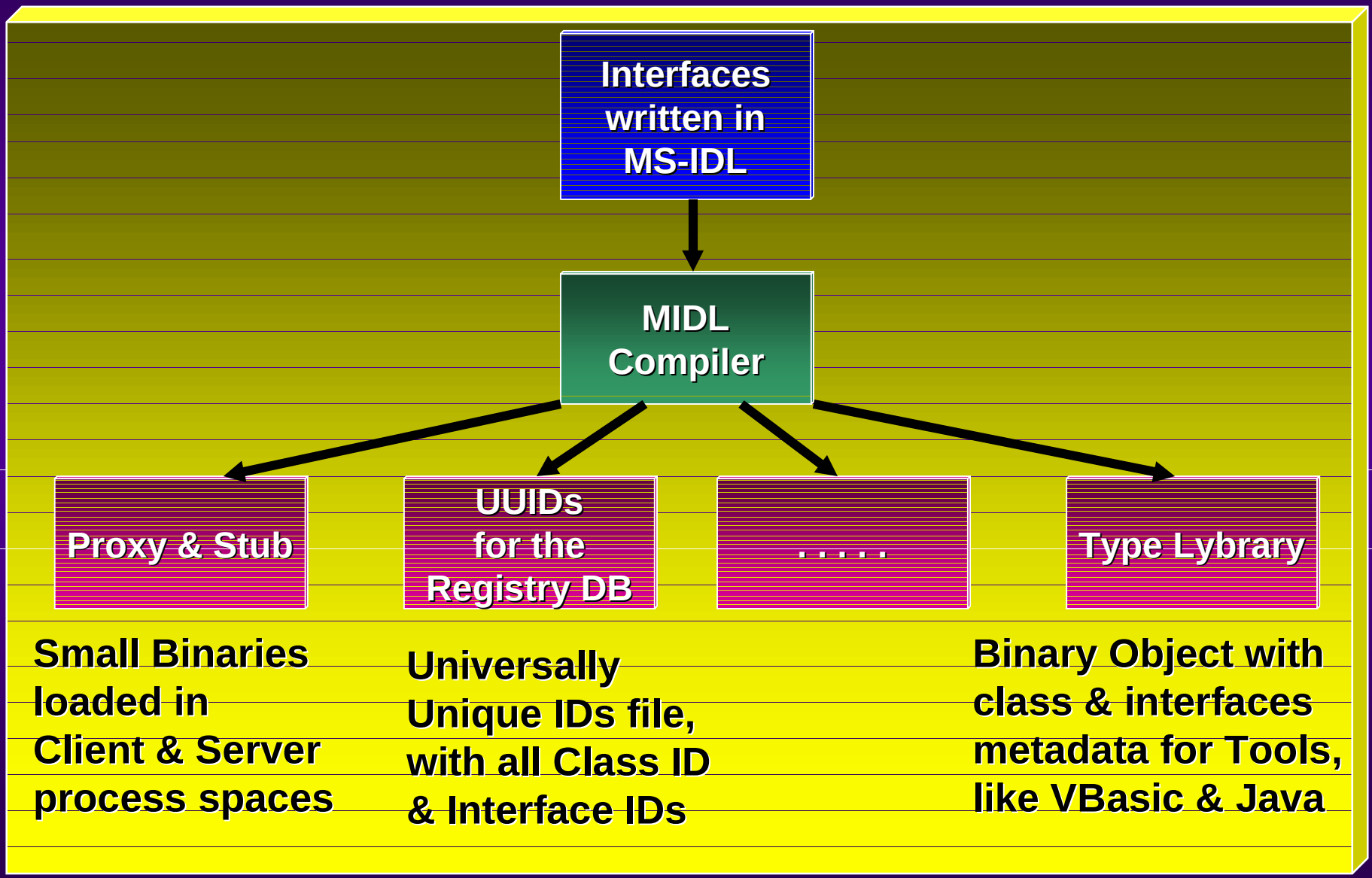
IUnknown Interface

- IUnknown defines a standard for communication with components
- All COM components are required to implement it
- IUnknown has three methods:
  - QueryInterface } Returns the Interface Pointer
  - AddRef
  - Release } Reference Counting Methods

Virtual Function Tables



IDL & COM


Interfaces written in MS-IDL
MIDL Compiler
Proxy & Stub
UUIDs for the Registry DB
. . . . .
Type Lybrary
Small Binaries loaded in Client & Server process spaces
Universally Unique IDs file, with all Class ID & Interface IDs
Binary Object with class & interfaces metadata for Tools, like VBasic & Java

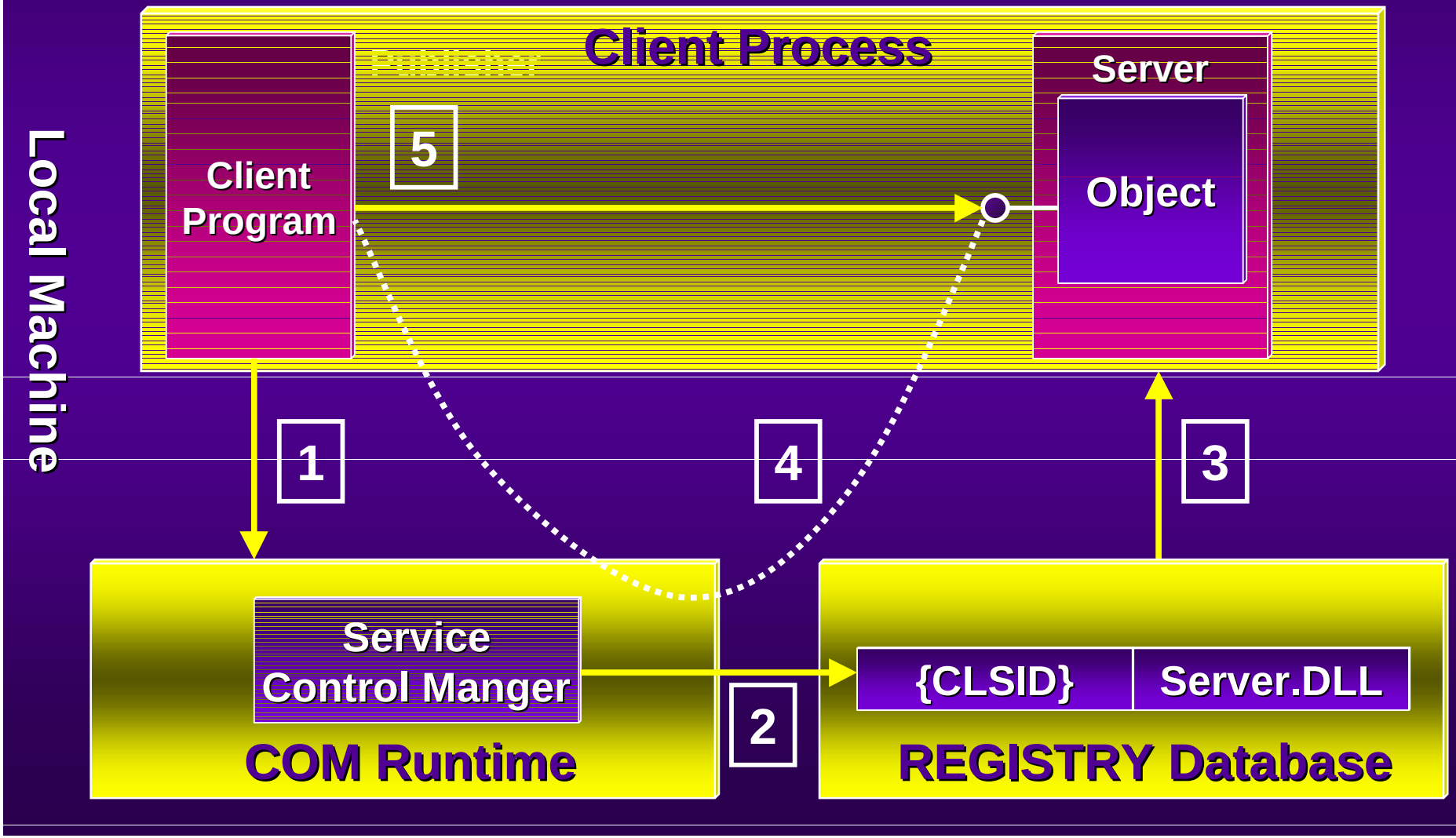
COM Library

A system component that provides the mechanism of COM

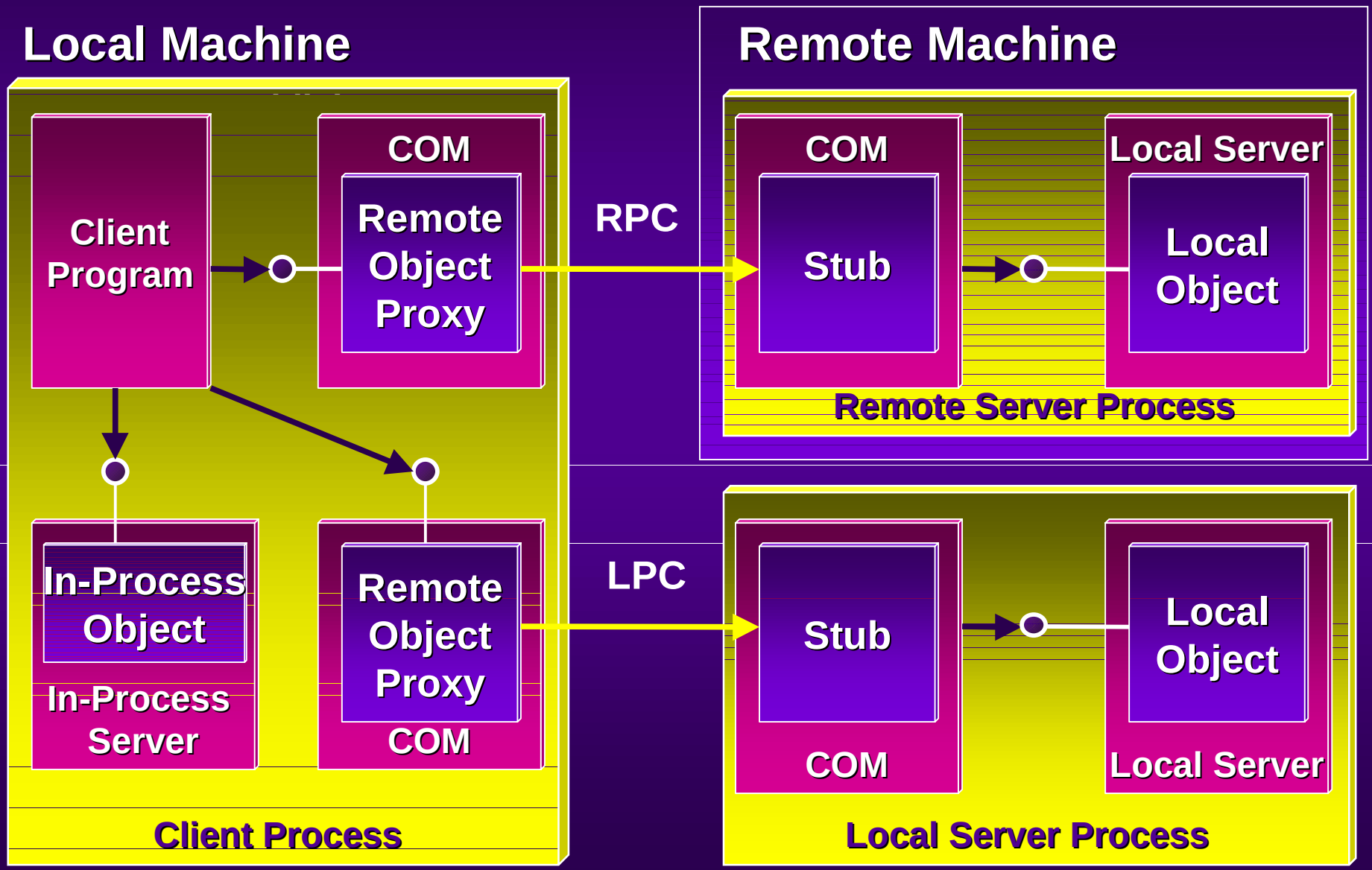
- Provides the ability to make **IUnknown** call across processes
- **Encapsulate** all work associated with the launching of components
- **Establish connections** between components
- In Win9x/WinNT is called OLE2.DLL

The COM Library

A System Component that provides the mechanism of COM



Accessing COM Services



Protocols & Security

- DCOM is based on MS-RPC
  - MS implementation of OSF/DCE-RPC specification
  - Tightly-Coupled & Synchronous
- DCOM can use all major Transport Protocols
  - default ones are TCP & UDP
- DCOM provides a Security Framework
  - It's based on WinNT Access Control Lists for components
  - ACLs indicate which users or groups of users have the right to access a component of a certain class

DCOM Administration

- For distributed applications, it's critical
  - To be able to centralize administration
  - To make client installation as straightforward as possible
- COM components administration is not easy:
  - COM components only increase in size (no versioning)
  - The biggest problem for maintaining "fat" clients is updating those clients to newer versions, by modifying the Registry file of each computer using them

DCOM Availability on non-Microsoft Platforms

- Sun Solaris 2.5 (Sparc)
- HP-UX
- Linux 2.0 Intel
- SCO UnixWare
- Siemens Nixdorf SINIX
- Digital Unix 4.0 (Alpha)
- IBM AIX
- Digital Open VMS
- IBM MVS 5.2.2 (OS390)
- IBM OS/400

EntireX - www.softwareag.com
COMsource - www.opengroup.org/comsource

Services Based on DCOM

- Microsoft Transaction Server
  - MTS manages Transactions, System Resources & Components' Life Cycle
  - MTS is a Standard for Server-Side COM Components
- Microsoft Message Queue
  - MSMQ implements asynchronous communication
  - MSMQ will be seen from DCOM as a new pluggable Transport Protocol (WinNT 5)

Domain Specific Services

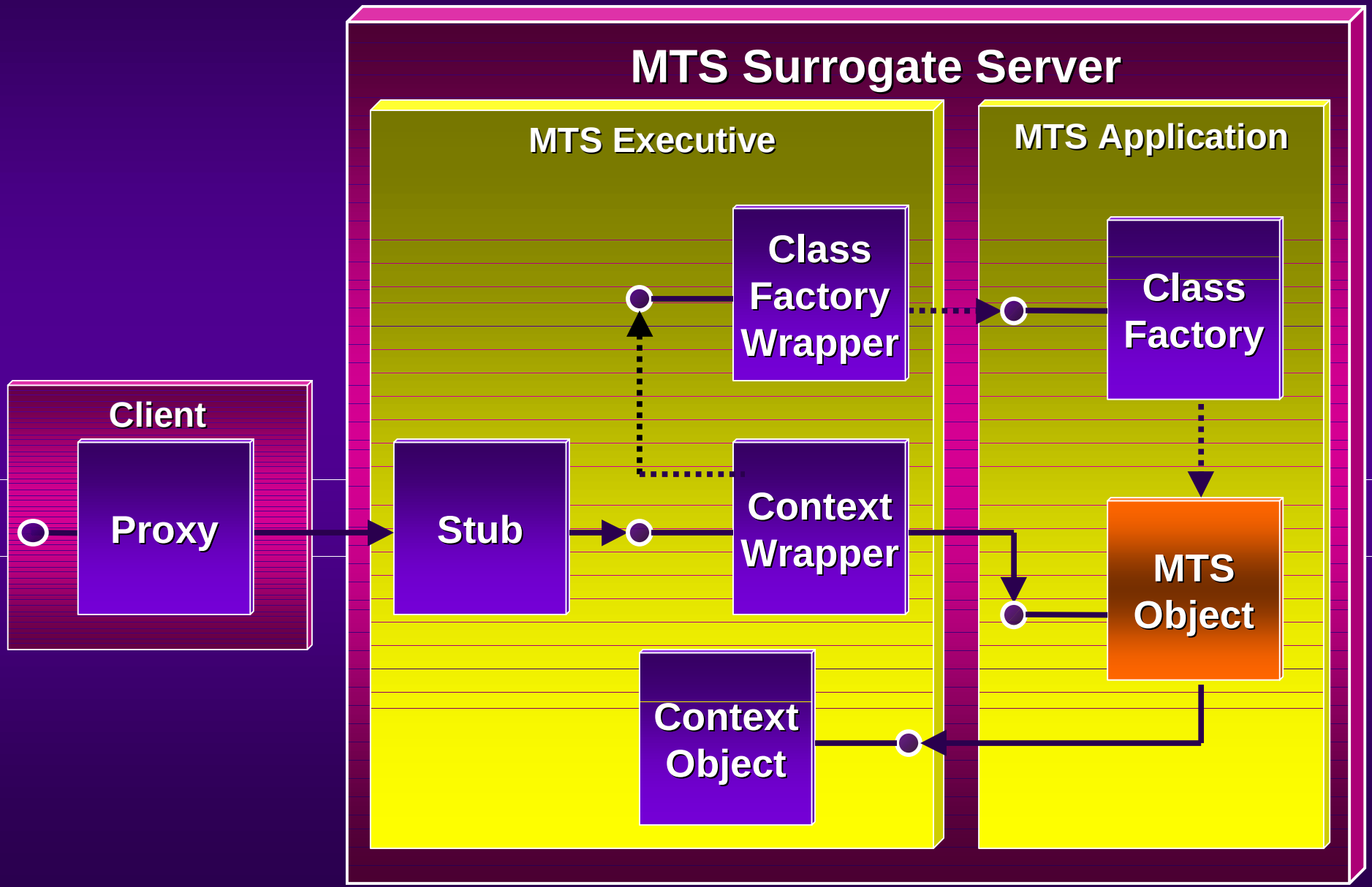
based on Windows DNA, the Windows Distributed interNet Applications Architecture

- Windows DNA is a framework for business solutions based on COM technologies
  - DCOM, MTS, MSMQ, IIS - MS Web Server, ...
- Available Industry Specifications & Products
  - OLE for Process Control (OPC)
  - Financial Services
  - Transportation & Distribution, ….

Microsoft Transaction Server

- DCOM only offers basic services for distributed computing
  - Building powerful COM servers can be daunting
- MTS integrates DCOM with
  - System Resources management
  - Server Component creation, execution & deletion management
  - Transactions Control & Security
- MTS is a standard for server-side components

MTS at work


MTS Surrogate Server
MTS Executive
MTS Application
Class Factory Wrapper
Class Factory
Client
Proxy
Stub
Context Wrapper
MTS Object
Context Object

MTS Constraints

- Constraints on COM Component
  - Be an in-process server (DLL)
  - Implement a Class Factory to create objects
  - Describe all of the component's interfaces in a Type Library
  - State-less objects only
- Available on NT platforms only

Microsoft Message Queue Server

- DCOM is based on MS-RPC
  - Tightly-Coupled & Synchronous
- MSMQ Communication Scenarios
  - Store-and-forward communication
  - Concurrent execution
  - Journaled communication
  - Connectionless communication
- MSMQ will be seen from DCOM as a new pluggable Transport Protocol

MS-Message Queue Server

- MSMQ is fully integrated with other Windows NT features such as
  - Microsoft Transaction Server (MTS)
  - Microsoft Internet Information Server (IIS)
  - Windows NT Security Environment
- MSMQ is available on Windows 9x & NT
- MSMQ is seen from DCOM as a new pluggable Transport Protocol

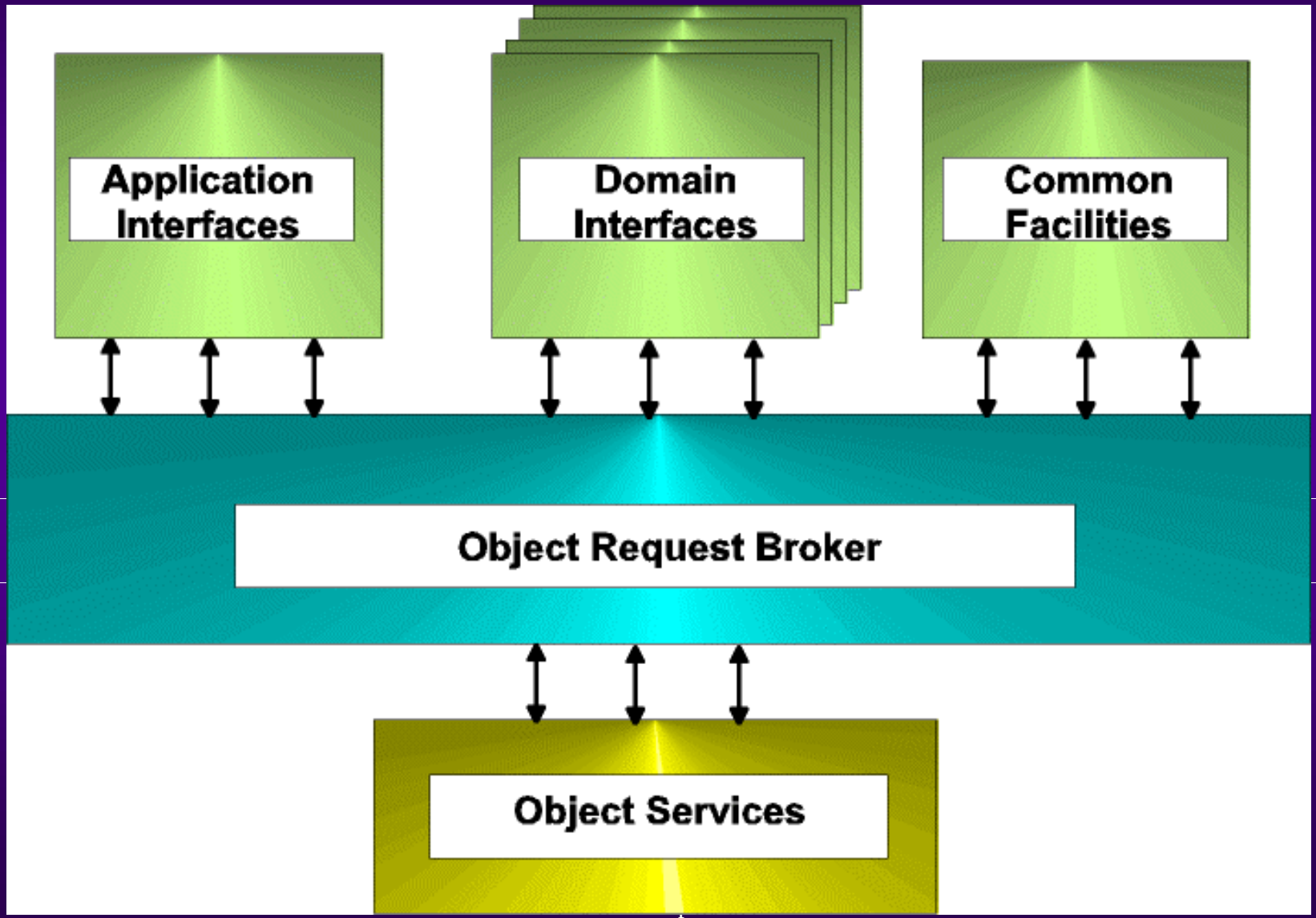
DCOM - Pros & Cons

- DCOM is a Mature Technology
  - Defines a Binary Standard for Components Interoperability
  - Broad Tools Support - VisualBasic, VC++, VJ++ & ActiveX
  - Is Programming-Language-Independent
  - Provides a transparent Remote Procedure Calls to both Local or Remote Servers
  - Can use all major Transport Protocols
  - MTS & MSMQ well integrate the basic DCOM functionality
  - Domain Specific Services are available on the market

DCOM - Pros & Cons

- COM components administration is not easy:
  - COM components update requires to modify the Registry file of each computer using them
  - COM components only increase in size (no versioning)
- DCOM Services are Windows-centric
- Only a few vendors are porting DCOM on other platforms
  - OpenGroup
  - Software AG Systems

CORBA Architecture



DCOM - CORBA
What's similar ?

- They provide:
- local and remote servers for distributed computing
- a framework with
  - ORB
  - Object Services
  - Common Facilities - horizontal services
  - Domain Specific Facilities - vertical services

DCOM - CORBA
What's different?

- **DCOM**
  - Microsoft & Open Group

    ~130 members
  - first implementation,

    then specification
- **CORBA**
  - OMG consortium

    ~ 800 members
  - first specification,

    then implementation

DCOM - CORBA
What's different?

- **DCOM**
  - Communication based on RPC protocol
    - synchronous communication
  - COM library with more built-in functions & less services
  - Built in OS
- **CORBA**
  - Communication based on IIOP or proprietary protocol
    - synch + asynchronous commun.
  - ORB with more services & less built-in functions
  - Built above OS

DCOM - CORBA
What's different?

- **DCOM**
  - Object consist of one or more **categories of interfaces**
    - each interface is named & has its own derivation hierarchy
    - Class identified by CLSID
  - Error reporting
    - methods return an integer value that indicate the success or failure of the call
- **CORBA**
  - Objects follow a **standard object model**
    - an interface is defined by its class & all the ancestors of that class
    - Class identified by Name
  - Error reporting
    - Exception mechanism, where errors are returned as a structure embedded within another object

Conclusions

The selection process between DCOM & others Middleware solutions revolves around the choice of server environments on which these systems will operate

References

- COM-DCOM-MTS & MSMQ Documentation
  - www.microsoft.com/com
  - www.microsoft.com/msmq
- EntireX
  - www.softwareag.com
- COMsource
  - www.opengroup.org/comsource