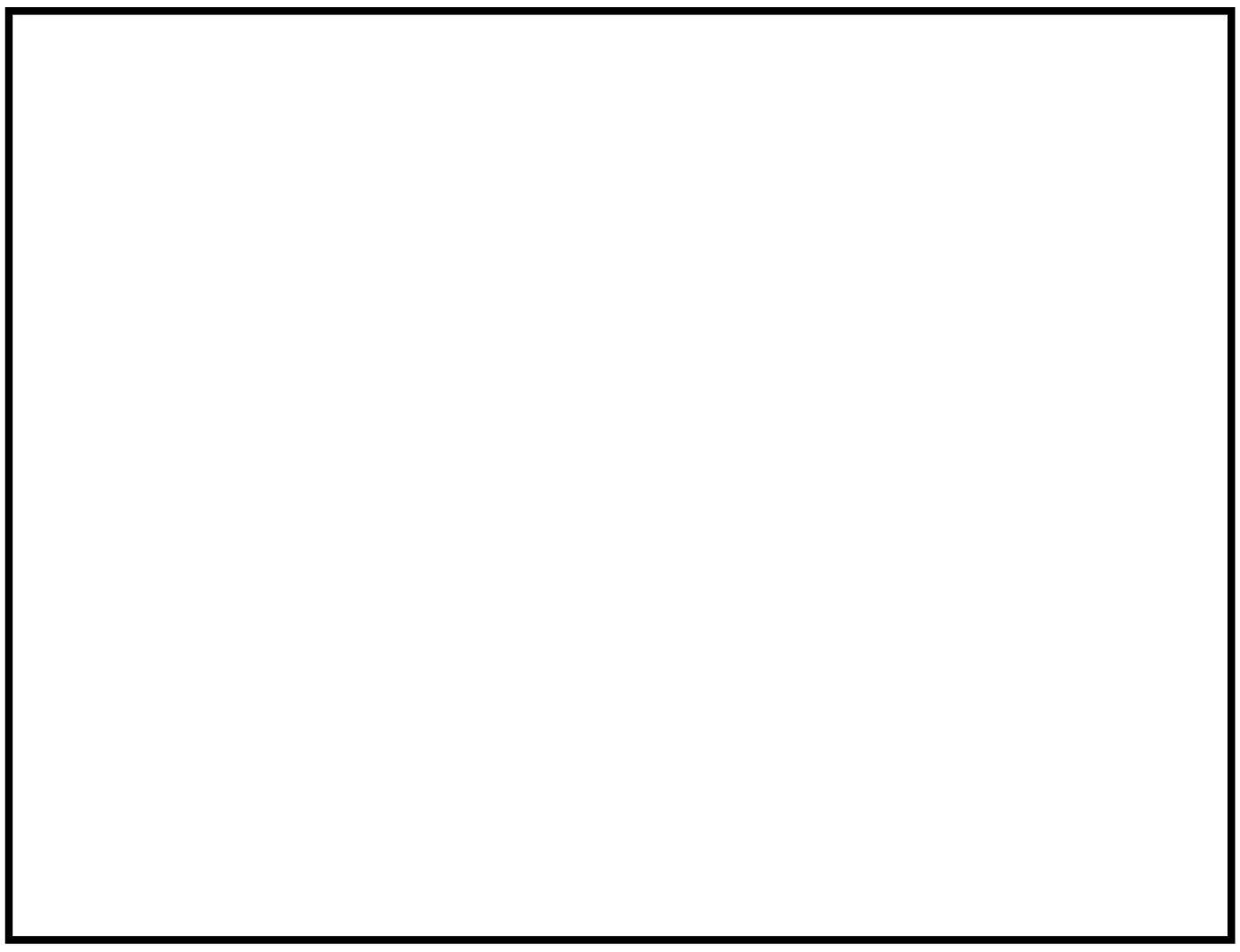
DCOM Overview

Outline

- COM overview
- DCOM overview
- Comparison DCOM and Corba

COM overview

- Standard for component interoperability
- binary standard
  - specifies how the component should be represented in the executable format
- language independent
- Designed for centralized systems
  - components within a process or across processes can communicate
  - DCOM generalizes this for the case where components are distributed
- Ideal for Windows based systems



Interfaces

- Interface specifies a collection of logically related functions
- Fundamental entity in COM
- Methods only – no variables or implementation
- An interface could be implemented by many components
- A special interface **IUnknown**

Interfaces (continued)

- Each interface has its own unique interface identifier (I-ID)
- 128 bit globally unique identifier (GU-ID)
- Based on the Distributed Computing Environment (DCE) standard
- Generated using the ethernet card number and exact time of creation to ensure uniqueness
  - No name collisions
  - Interface atomicity
  - Interfaces are immutable; if the interface is modified, it is saved as a new interface

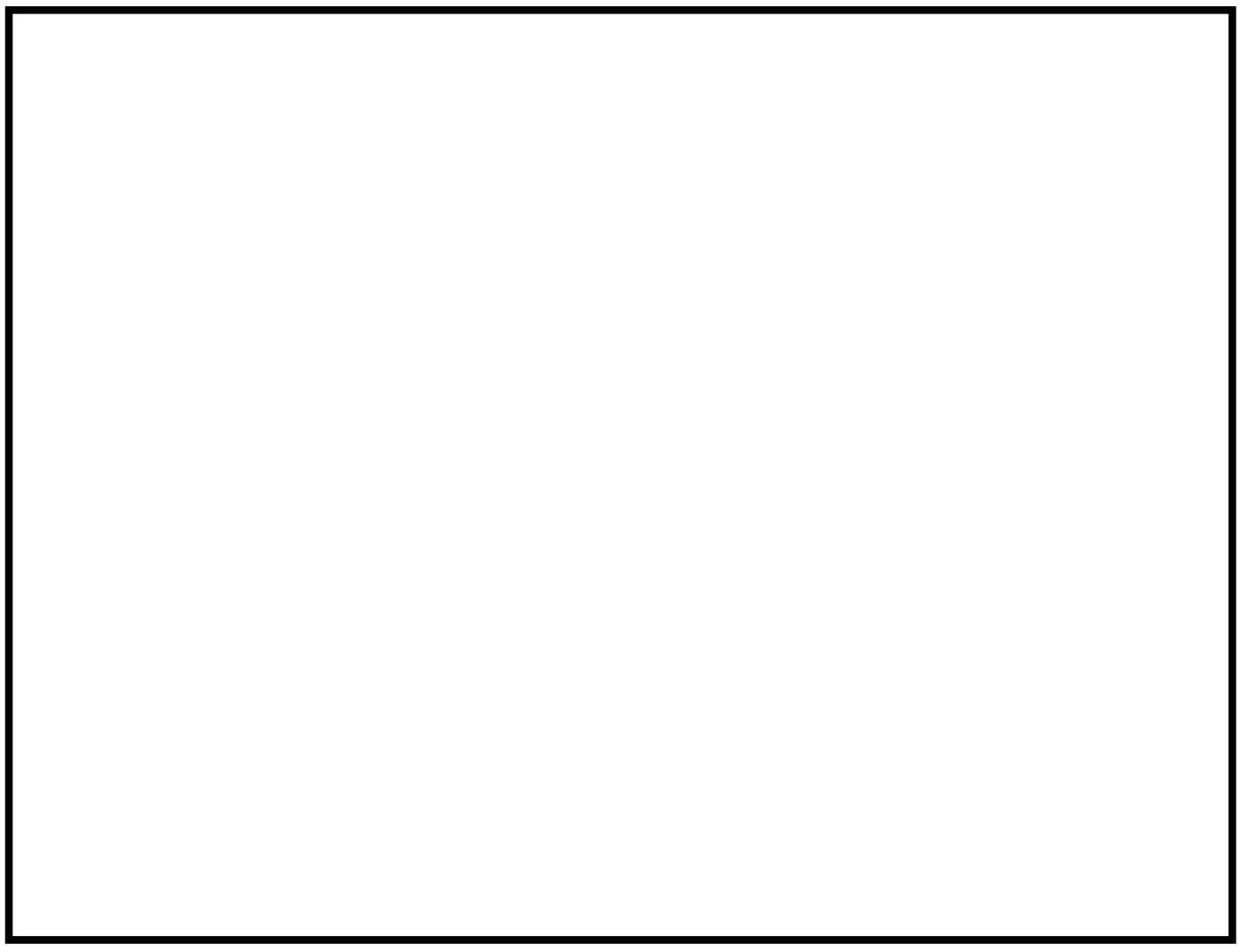
Interface IUnknown

- A special interface **IUnknown**
  - Every interface must inherit from IUnknown
  - Contains three methods: AddRef, Release and QueryInterface
    - used for reference counting and managing the lifetime of an object

```
interface IUnknown
{
   virtual HRESULT QueryInterface(IID &, void **) = 0;
   virtual ULONG AddRef() = 0;
   virtual ULONG Release() = 0;
}
```

Interface IUnknown (continued)

- Used for interface negotiation; the client can ask an object about the interfaces it can provide
- Given a particular IID, the client can determine if the object of that interface supports the required interface
- The value of type HRESULT is used to denote whether the operation was successful or not.
  - 0 = success
  - other values = (warning or error)

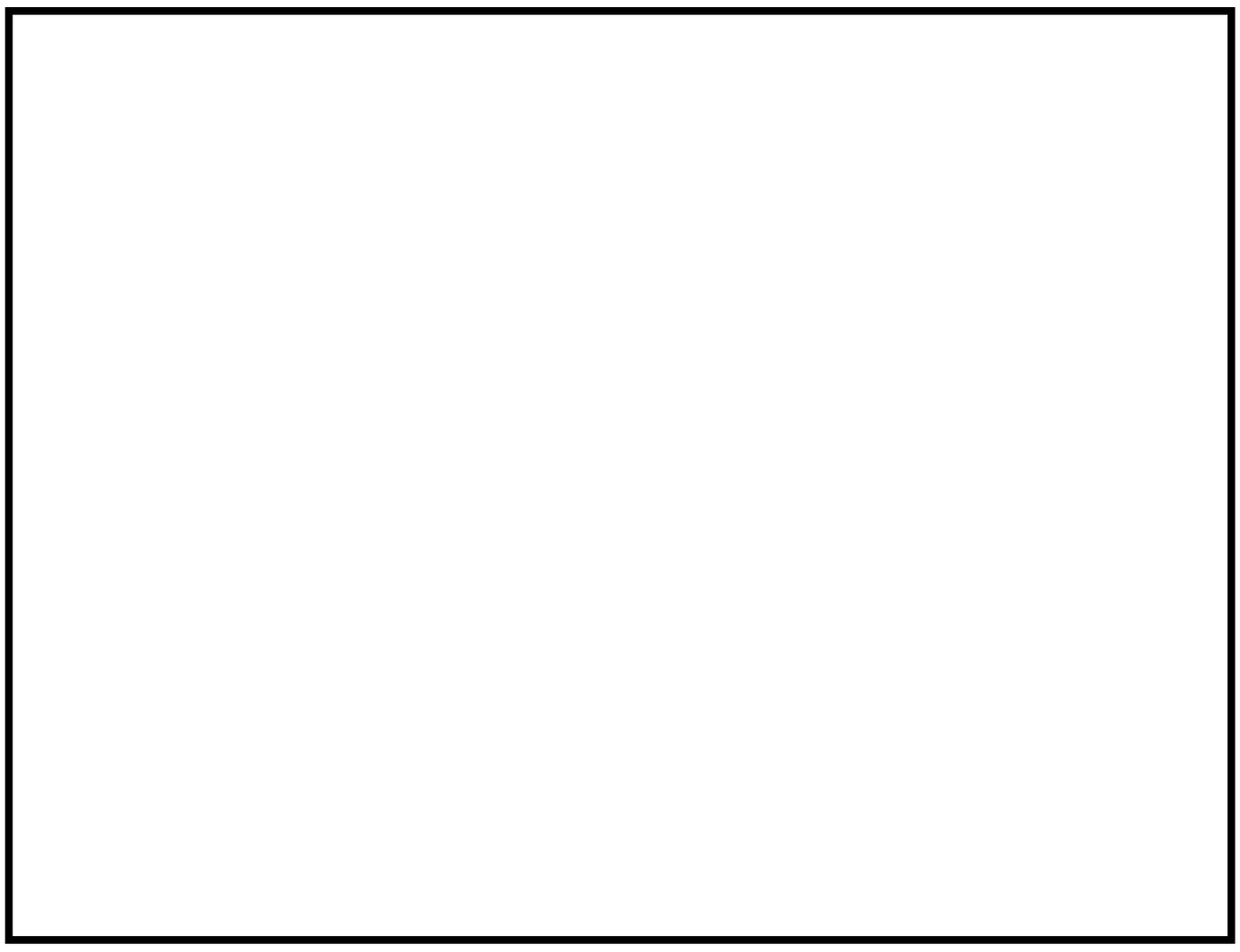


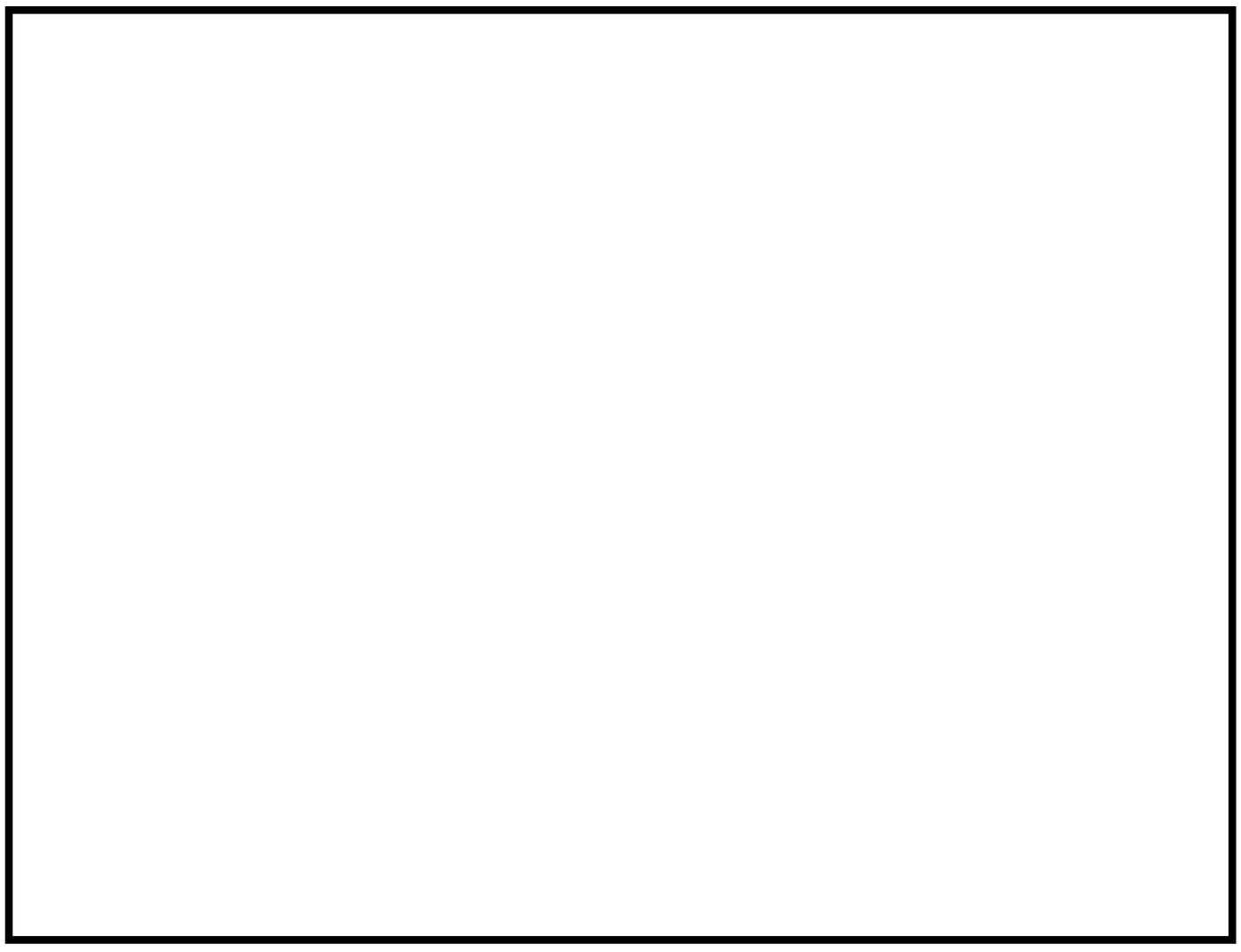
Binary standard for COM

- The interface consists of a pointer to an interface node
- The interface node contains a reference to a `vtable`
- `vtable` contains references to the methods supported by an interface (including QueryInterface, AddRef and Release)

Binary standard (continued)

- Efficiency
  - A call to COM method is as efficient as a C++ virtual method call
- Language independent
  - The source program may be in any language
  - The compiler output should conform to the required memory layout



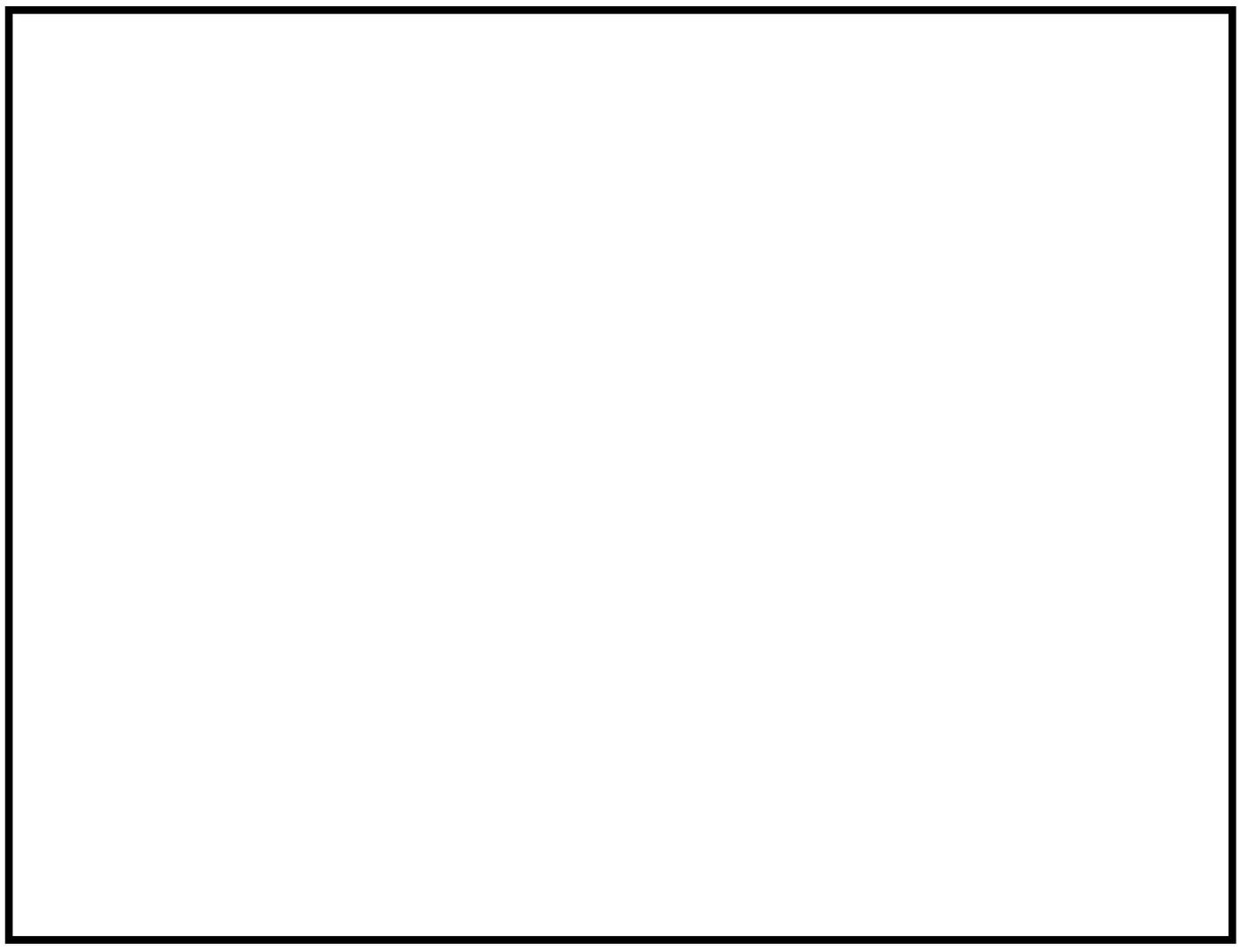


Class Identifier (CLS-ID)

- Each component also has a globally unique class identifier (CLSID)
- Based on the interfaces it supports and the development time
- A class implements one or more interfaces
  - clients interact with COM objects through the interface
  - clients must have a separate pointer to each interface they want to use
  - no access to object state
  - only functions in the interface may be called

Creating COM objects

- A COM object is a runtime instantiation of a COM class
- Every class must provide a class factory that deals with creating new instances of that class
- Given a CLSID, a new instance can be requested through COM library using
  - CoCreateInstance
  - CoGetInstance



Typical Execution

- Client uses the COMAPI to obtain information about the DLL that contains the class factory
- Using the DLL, it obtains the reference to the Classfactory
- The client calls CoCreateInstance on the class factory to obtain an object with the given interface
- The client uses the object
- The client releases the object
- The client releases the classfactory

Version Control

- Not supported by COM directly
- Multiple interfaces are used to solve the version problem
- If an interface is modified, the object may support both the older and the newer version.

Reference Counting

- AddRef and Release methods from IUnknown used for this purpose
- The clients are responsible for reference counting and doing memory management.
  - Improper use can lead to memory leaks, dangling pointers,
- Can fail to deal with cyclic structures

DCOM : Goals

- COM - with a longer wire
- Enable software components to communicate directly over a network in a reliable, secure and efficient manner
- Designed for use across multiple network transports including Internet protocols such as TCP, HTTP
- Intended to extend COM, work with Java applets and ActiveX components
- e.g., Java may be used for front-end graphic interface that communicates with other databases, other applications on the network.

DCOM : Issues

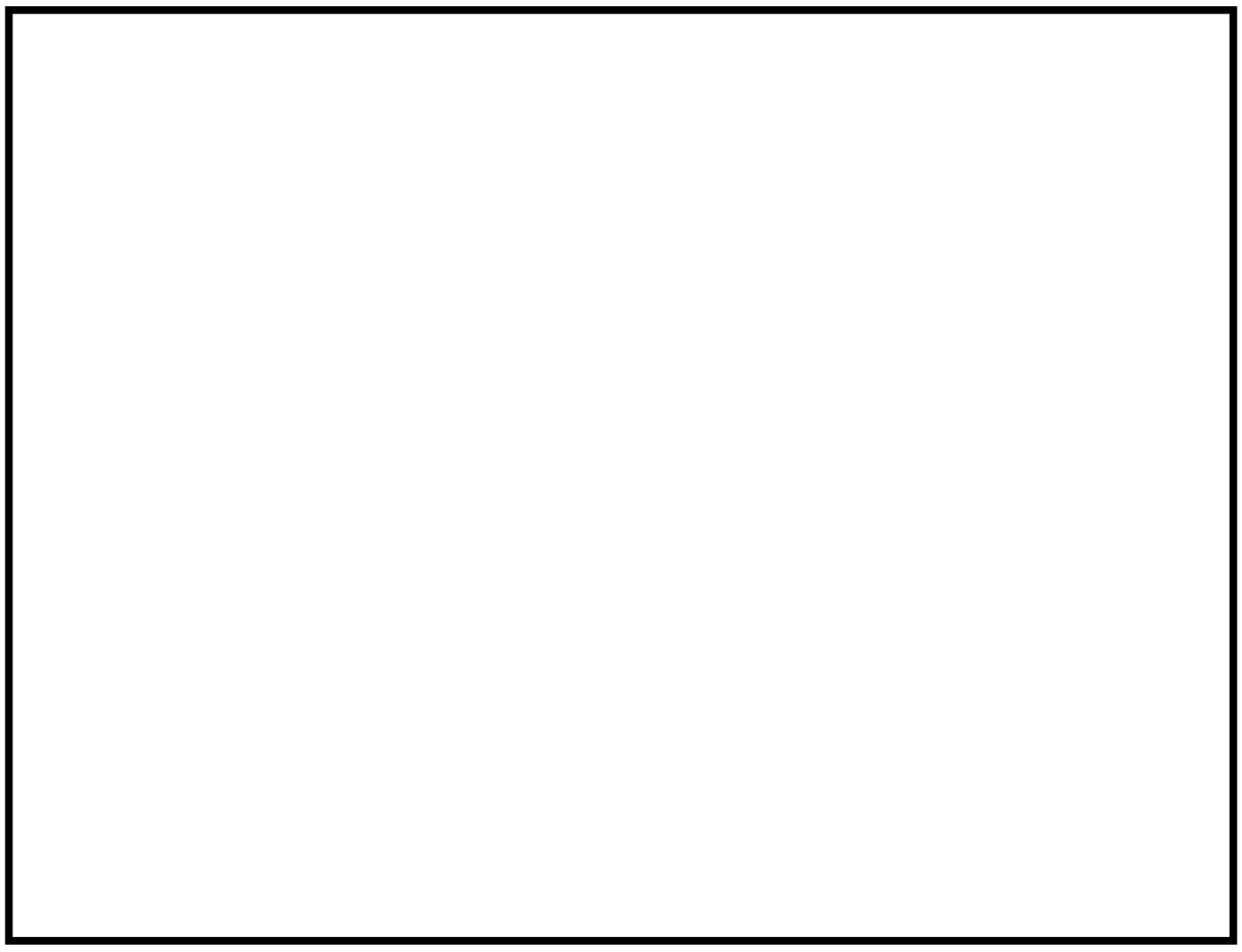
- Interoperability across network
- Versioning: updating some components without requiring everything else to be updated!
- Permit components written in different languages to communicate
- Transparency

MS-IDL

- DCOM is also RPC based
  - Object Remote Procedure Call
- DCE-RPC defines a standard for converting in-memory data structures and parameters into network packets

Object Remote Procedure Calls (ORPC)

- Microsoft's object oriented extension of the distributed computing environment (DCE) remote procedure call (RPC)
- Specifies
  - how calls are made on an object
  - how object references are communicated, maintained,
- Multiple network protocols
  - TCP/UDP/HTTP/..



Marshaling and Unmarshaling

- required to pass function calls, parameters, and return values
- Marshalling
- Unmarshalling
- Stub and proxy objects

Security

- Based on DCE RPC that provides for authentication and authorization
- Based on access control lists (ACLs) of COM components
- Different security protocols may be used