

ТЕХНОЛОГИЯ СОМ (COM-ТЕХНОЛОГИИ
COMPONENT OBJECT
TECHNOLOGY) –
ОБЪЕКТНО-
ОРИЕНТИРОВАННАЯ
ПРОГРАММНАЯ
СПЕЦИФИКАЦИЯ,
ПРЕДЛОЖЕННАЯ
MICROSOFT.

GUID
**(GLOBAL UNIQUE
INTERFACE
IDENTIFYER)**

АРХИТЕКТУРА COM
ЯВЛЯЕТСЯ РАСШИРЯЕМОЙ

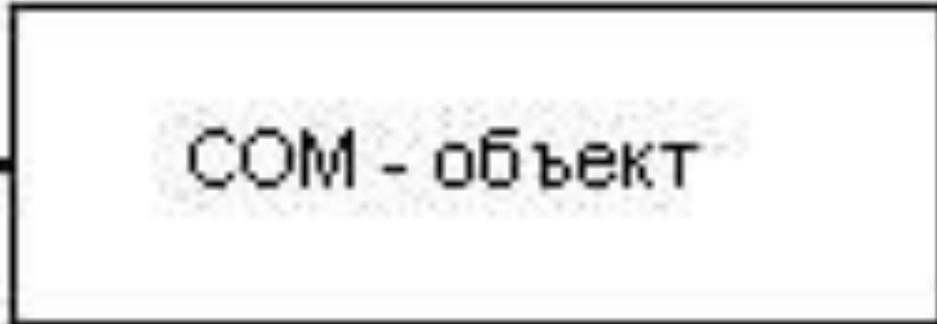
НА НЕЙ БАЗИРУЮТСЯ
ДРУГИЕ ТЕХНОЛОГИИ
MICROSOFT, ТАКИЕ
КАК *OLE* И *ACTIVEX*.

СОСТАВ СОМ-ОБЪЕКТА

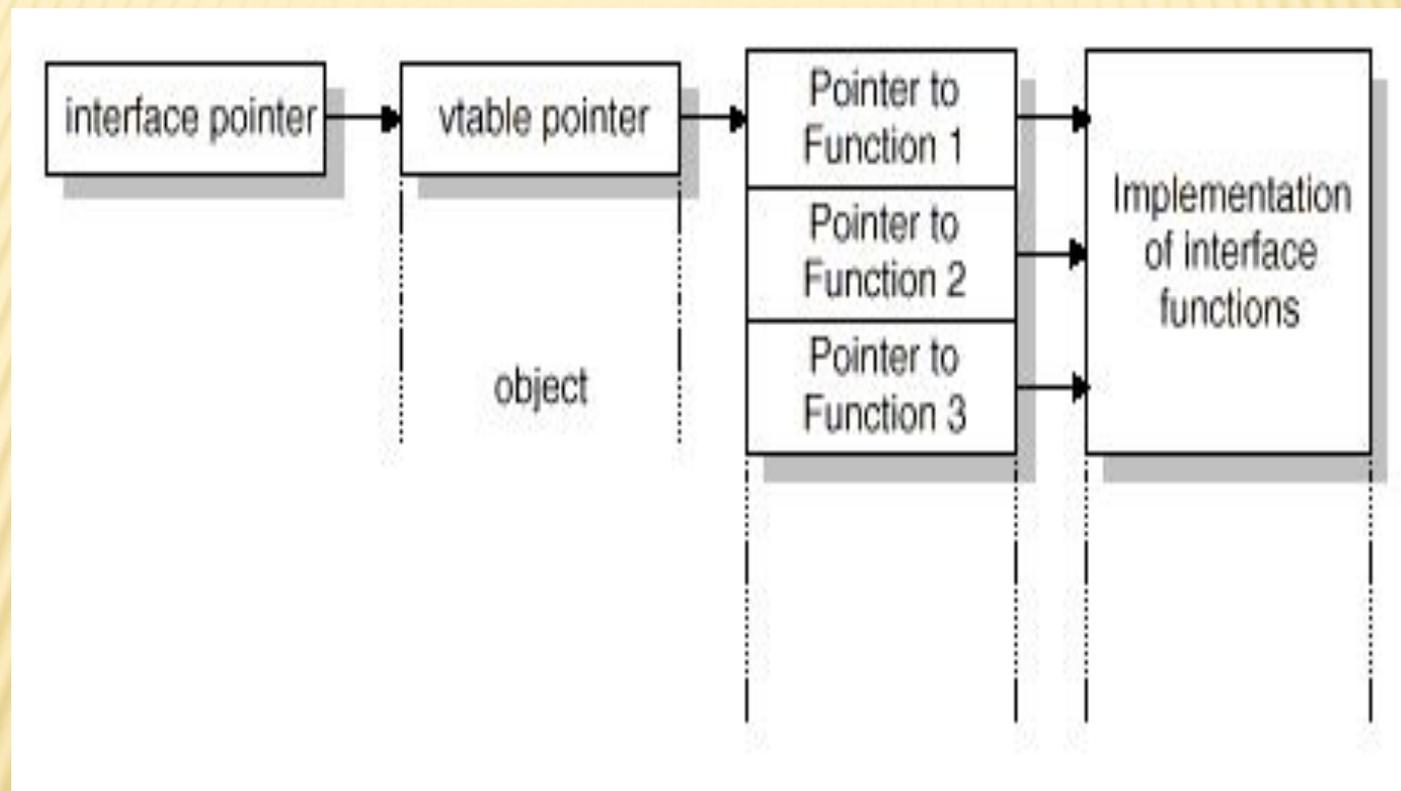
- ▣ **Interface** (СОМ-интерфейс) - множество прототипов функций (методов), чисто определенных. Термин «*чисто определенный метод*» или «*абстрактный метод*» исходит теории объектно-ориентированного анализа, и означает, что в определении класса отсутствует реализация метода, а присутствует только его определение. От такого класса нельзя создавать объекты. Его предназначение – описать фундаментальные общности для всех производных классов;
- ▣ **COM object** (СОМ-объект) – объект класса CoClass, который содержит реализацию СОМ интерфейса;
- ▣ **COM/ActiveX server** (СОМ сервер или ActiveX сервер) – модуль, такой как EXE, DLL или OCX, который содержит машинный код СОМ или ActiveX объектов;
- ▣ **Class factory** (фабрика классов) – объект, который может создавать СОМ-объекты из CoClass;
- ▣ **Type library** (библиотека типов) – файл, содержащий информацию о типах данных, которые использует СОМ/ActiveX сервер.

ИНТЕРФЕЙСЫ

Интерфейс



ИНТЕРФЕЙС ПОДЧИНЯЕТСЯ ДВОИЧНОМУ СТАНДАРТУ.



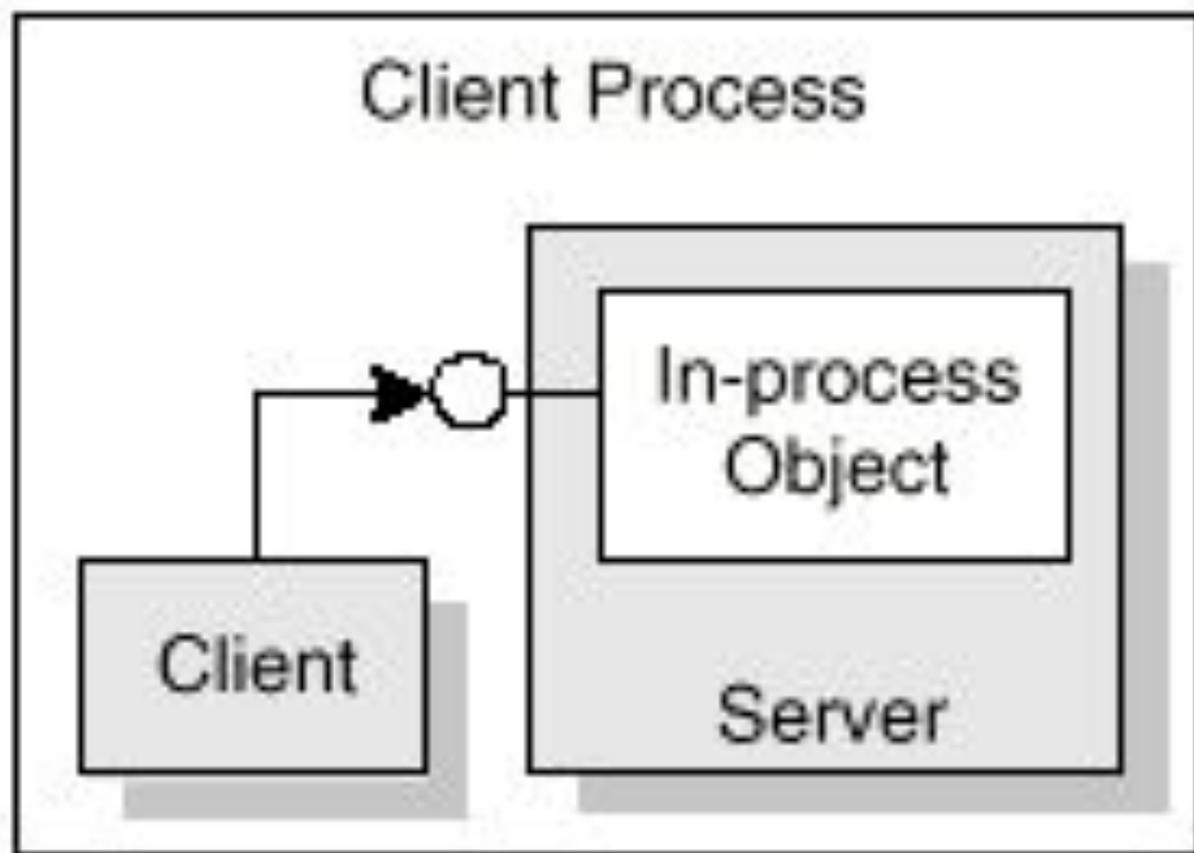
COM-СЕРВЕРЫ

- Объект COM-класса должен иметь в своем составе фабрику классов, и идентификатор класса CLSID (Class Identifier), так чтобы COM-объект мог быть создан на основе существующего модуля.

СОМ-СЕРВЕР МОЖЕТ БЫТЬ СЛЕДУЮЩИХ ТИПОВ:

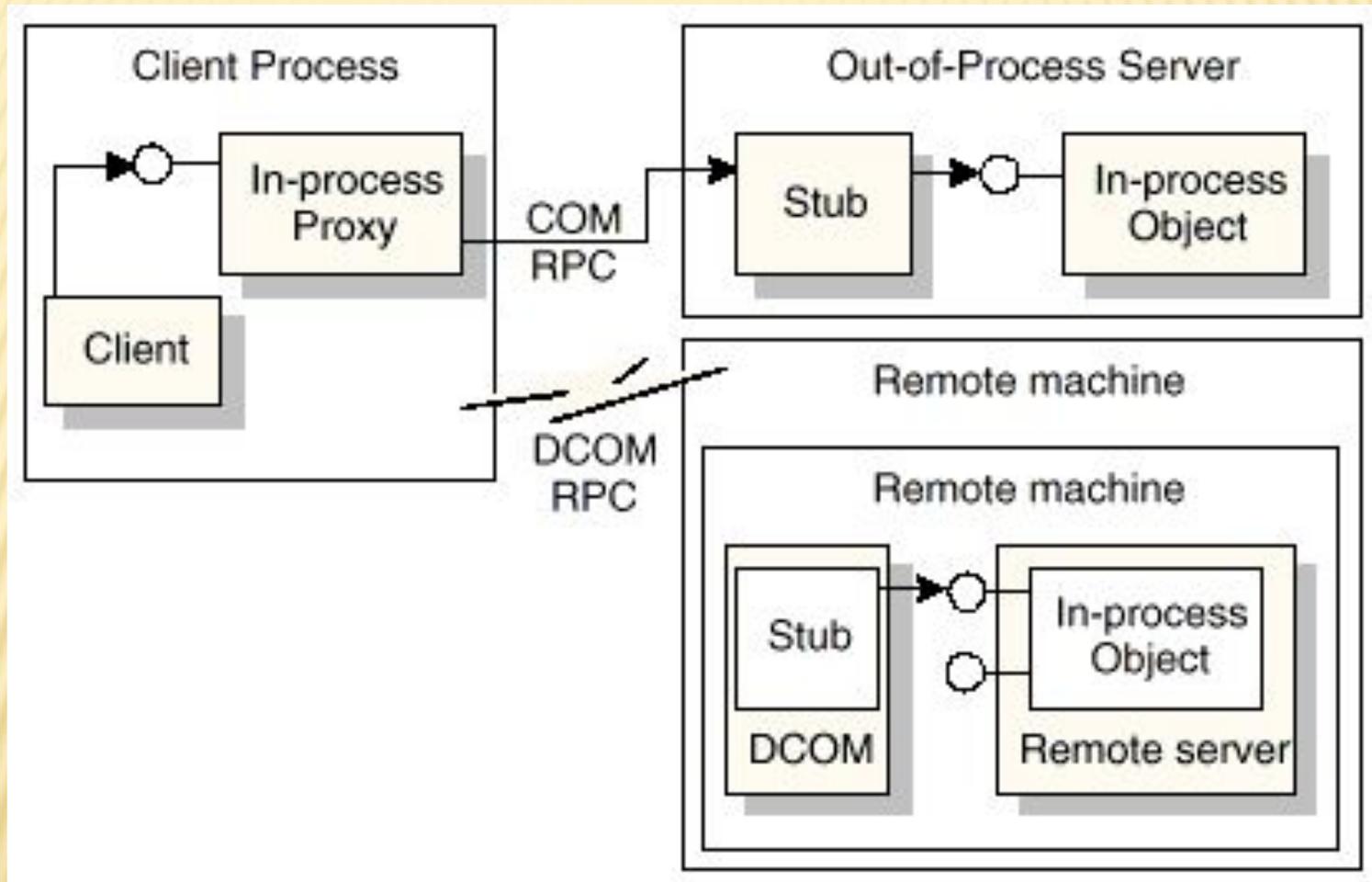
1. ***In-process server*** (внутренний сервер) – программный DLL модуль, работающий в рабочем пространстве памяти клиентского приложения:

Client Process



-
2. **Local server** (локальный сервер) – программный EXE модуль, работающий в отдельном адресном пространстве;
3. **Remote server** (удаленный сервер) – программный EXE модуль, работающий на удаленной машине:

МЕХАНІЗМ МАРШАЛІНГА (MARSHALLING).



ФАБРИКИ КЛАССОВ (CLASS FACTORIES)

Создание объекта класса производится посредством следующих действий:

- вызова глобальной **api**-функции **cogetclass**, которая ищет в системном реестре зарегистрированный класс с данным **clsid**, определяет путь к серверу, загружает сервер и выдает указатель на интерфейс производителя классов (обычно **iclassfactory**);
- УКАЗАТЕЛЬ НА **ICLASSFACTORY** МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ ВЫЗОВА МЕТОДОВ ПРОИЗВОДИТЕЛЯ КЛАССОВ, НАПРИМЕР: **COCREATEINSTANCE** (СОЗДАНИЕ ОБЪЕКТА);

БИБЛИОТЕКИ ТИПОВ (TYPE LIBRARY)

ПОСЛЕ СОЗДАНИЯ
БИБЛИОТЕКИ ТИПОВ, К НЕЙ
ОБЕСПЕЧИВАЕТСЯ ДОСТУП
ЧЕРЕЗ СПЕЦИАЛЬНЫЙ ТИП
ИНТЕРФЕЙСОВ: **ITYPELIB,**
ITYPEINFO И **ITYPECOMP.**