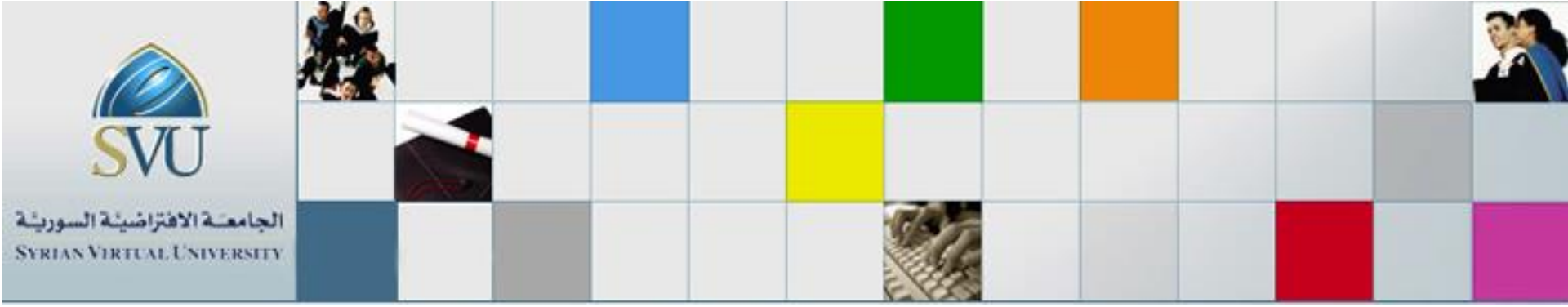


البرنامج: ماجستير الدراسات العليا في علوم الويب
رمز البرنامج: MWS
الفصل: **S20**

اسم المقرر: برمجة التطبيقات الشبكية والموزعة
رمز المقرر: ADP
الصف: C2

اسم المنسق: د. محمد الجنيدي
المدرس: د. وسيم الجنيدي

رقم الجلسة: الجلسة الأولى
موضوع الجلسة: مقدمة
تاريخ الجلسة: 2020-10-29

What Is Remoting?

- ***Remoting*** is the process of programs or components interacting across certain boundaries.
- These contexts will normally resemble either **different processes or machines**.
- We have to distinguish between ***remote objects*** and ***mobile objects*** .
- ***remote objects:*** provides the ability to **execute methods on remote servers**, passing parameters and receiving return values.
  - Always "stay" at the server.
  - Only a reference to it will be passed around among other machines.

What Is Remoting?

- ***mobile objects:***
  - They are **serialized** (**marshaled**) into a general representation-either a binary or a human readable format like XML-and then de-serialized in the other context involved in the process.

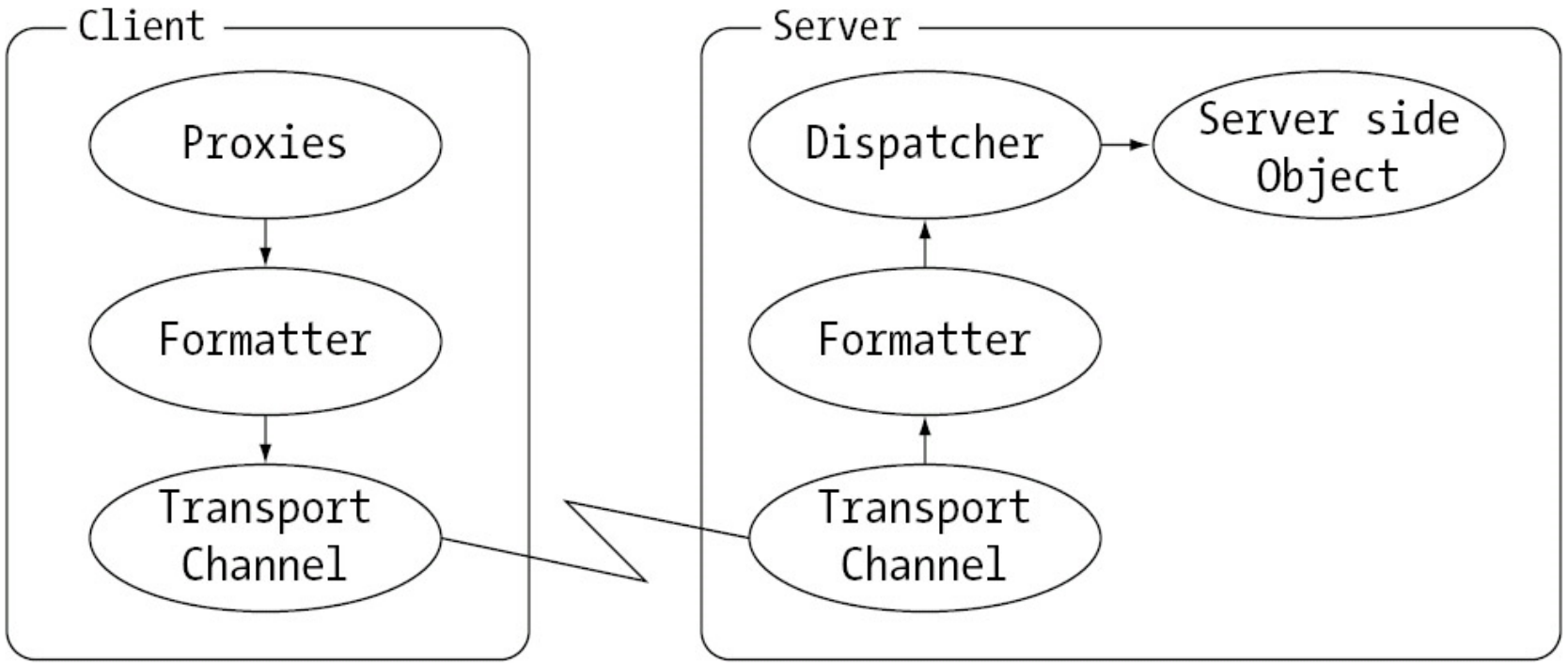
mobile objects

- Server and client both hold copies of the same object.
- Methods executed on those copies of the object will always be carried out in the local context, and no message will travel back to the machine from which the object originated.
- after serialization and deserialization, the copied objects are indistinguishable from regular local objects, and there is also no distinction between a server object and a client object.

.NET Remoting Basics

Your first remoting application

.NET Remoting



Your First Remoting Application
MarshalByRefObjects

- They allow the ability to execute remote method calls on the server side.
- These objects will live on the server and only a so-called **ObjRef** will be passed around.
  - think of the ObjRef as a networked pointer that shows on which server the object lives and contains an ID to uniquely identify the object

Your First Remoting Application
MarshalByRefObjects

- The client will usually not have the compiled objects in one of its assemblies; instead only an interface or a base class will be available.
- Every method, including property gets/sets, will be executed on the server
- The .NET Framework's proxy objects will take care of all remoting tasks, so that the object will look just like a local one on the client.

Your First Remoting Application
ByValue objects

- When these objects are passed over remoting boundaries (as method parameters or return values), they are serialized into a string or a binary representation and restored as a copy on the other side of the communications channel.
- After this re-creation, there is no notation of client or server for this kind of object; each one has its own copy, and both run absolutely independently.
- Methods called on these objects will execute in the same context as the origination of the method call.

Your First Remoting Application
ByValue objects

- For example, when the client calls a function on the server that returns a ByValue object, the object's state (its instance variables) will be transferred to the client and subsequent calls of methods will be executed directly on the client.
- This also means that the client has to have the compiled object in one of its assemblies.

Your First Remoting Application
ByValue objects

- The only other requirement for an object to be passable by value is that it supports serialization.
- This is implemented using a class-level attribute: **[Serializable]**.
- In addition to this "standard" serialization method, you'll also be able to implement **ISerializable**.