# I 4-736 DISTRIBUTED SYSTEMS

RMI (KESDEN, SPRING 2019)

# DISTRIBUTED OBJECTS

- Maintain an understanding of the identities of stateful objects
- Invoke methods upon these remotely accessible objects
- Obtain and pass around references to these remotely accessible objects
- Note: Objects versus Classes
  - There can be many instances, some remotely accessible, some note

# <sup>3</sup> DISTRIBUTED OBJECTS



• Common organization of a remote object with client-side proxy.

## JAVA IS MY FAVORITE RMI EXAMPLE: SIMPLE SOLUTIONS, EASY USE

- Interfaces provide common reference type for proxy and remote instances
- Serializable vs Remote interface
  - Remote: Send Remote-Object-Reference (ROR) and localize to proxy reference
  - Does not implement Remote: Needs to be Serializable (and not Remote):
    - Send copy and deserialize
  - Neither: Error
- Registry: Trade name for ROR

# JAVA RMI

- Original version:
  - "rmic" generated proxy and skeleton classes from the base .class file
- Step to simplicity:
  - The skeletons were really formulaic. They all had the same code. All they did was invoke a local method
    - Replace with a dispatcher that parses the incoming invocation and dispatches it locally
  - Proxies are formulaic. Interfaces provide all the methods, arguments, etc. ROR provides server information
    - Automatically generate them dynamically.
    - No more need for rmic. All dynamic.

### JAVA RMI EXAMPLE: INTERFACE

#### HelloInterface.java:

interface HelloInterface extends Remote {
 public String sayHello(String name)
 throws RemoteException;

### JAVA RMI EXAMPLE: SERVER

#### Hello.java

```
class Hello extends UnicastRemoteObject implements HelloInterface {
    private static final String serverName = "hello";
    public Hello() throws RemoteException { }
    public String sayHello(String name) throws RemoteException {
        return "Hello World! Hello " + name;
        }
    public static void main (String []args) {
        try {
            Hello server = new Hello();
            Naming.rebind (serverName, server);
            System.out.println ("Hello Server ready");
    }
}
```

```
} catch (Exception e) {
```

```
e.printStackTrace();
```

### JAVA RMI EXAMPLE: CLIENT

#### HelloClient.java

```
class HelloClient {
```

```
static void main (String []args) {
  try {
     Ctring | |a||a Convert | D|
     convert| D|
     convert | D|
```

```
String HelloServerURL = args[1];
```

```
System.setSecurityManager (new RMISecurityManager());
```

```
HelloInterface hello = (HelloInterface) Naming.lookup(HelloServerURL);
```

```
String theGreeting = hello.sayHello (args[0]); System.out.println (theGreeting);
} catch (Exception e) {
    e.printStackTrace();
}
```