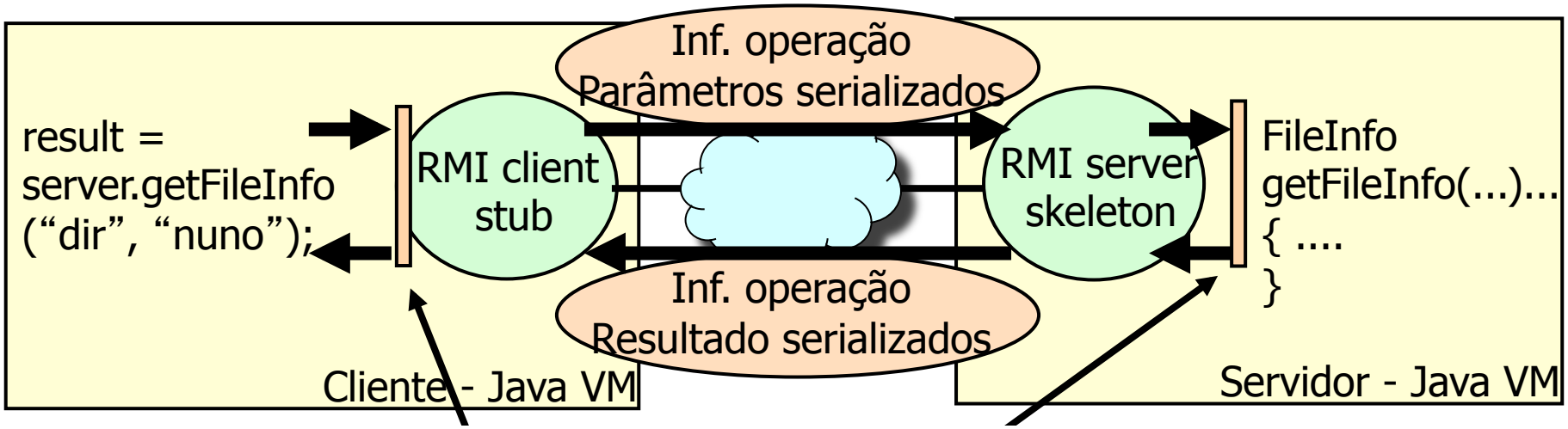
JAVA.RMI

RMI (Remote Method Invocation)

Mecanismo pelo qual um objecto pode invocar métodos de objectos não residentes na mesma JVM

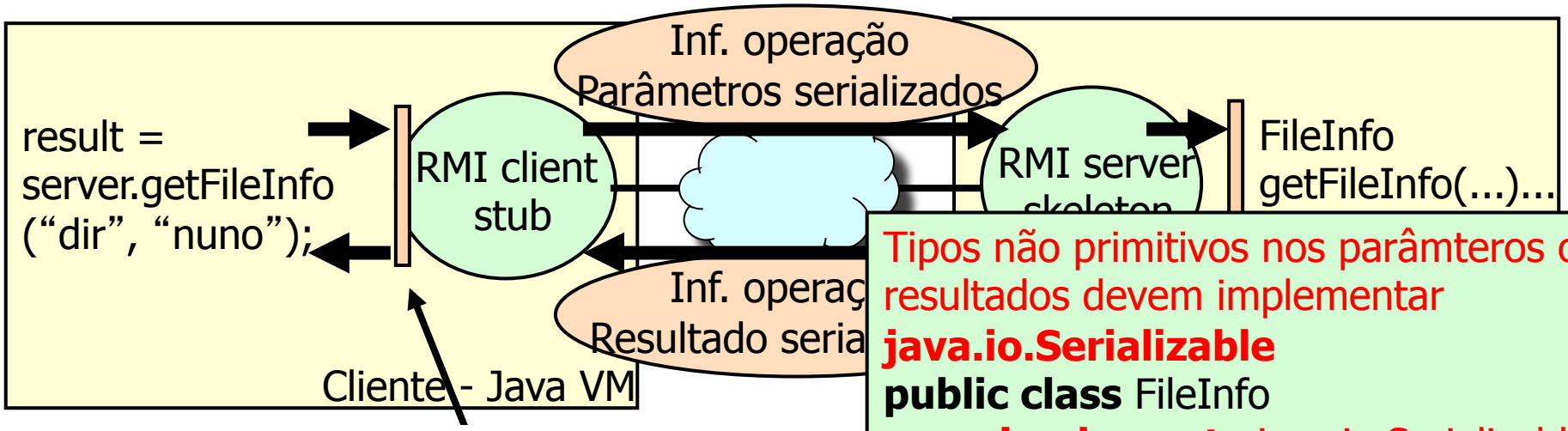



```
public interface IFileServer extends java.rmi.Remote {
    public FileInfo getInfo( String path, String name)
                    throws RemoteException, InfoNotFoundException;
    ...
}
```

JAVA.RMI

RMI (Remote Method Invocation)

Mecanismo pelo qual um objecto pode invocar métodos de objectos não residentes na mesma JVM




```
public interface IFileServer extends java.r
    public FileInfo getInfo( String path, Strin
                    throws RemoteException,
    ...
}
```

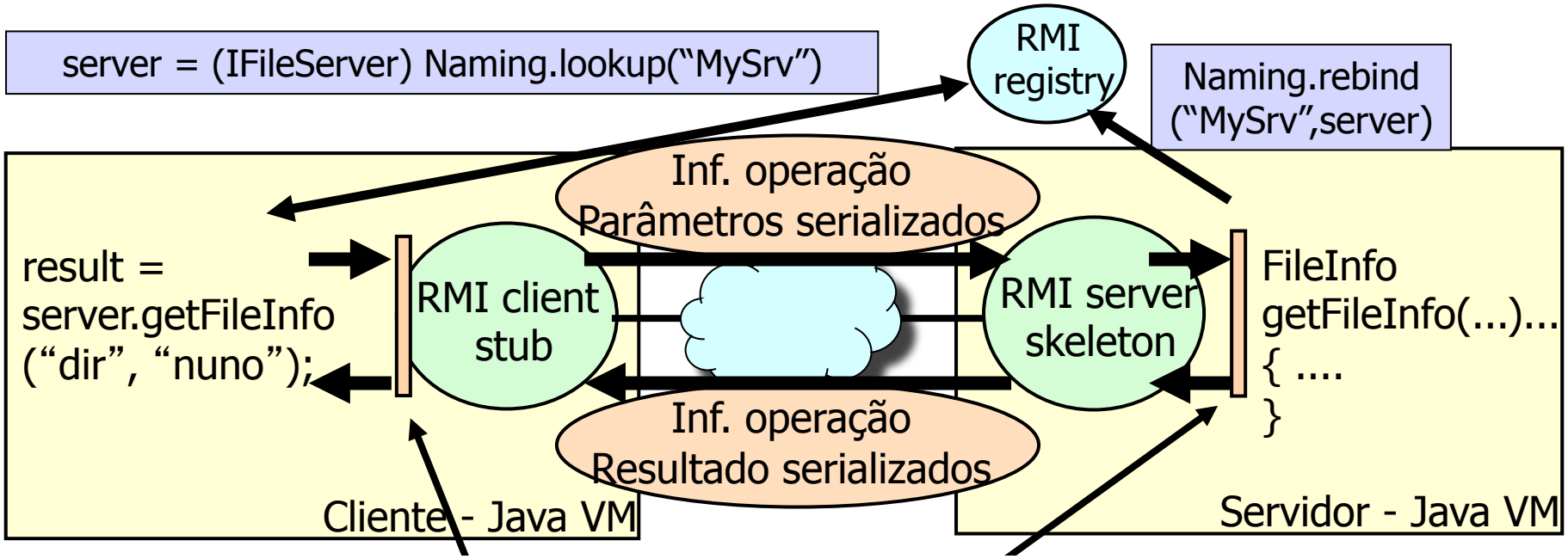
Tipos não primitivos nos parâmteros ou resultados devem implementar **java.io.Serializable**

```
public class FileInfo
        implements java.io.Serializable
{
    public String name;
    public Date modified;
    ...
}
```

JAVA.RMI

RMI (Remote Method Invocation)

Mecanismo pelo qual um objecto pode invocar métodos de objectos não residentes na mesma JVM




```
public interface IFileServer extends java.rmi.Remote {
    public FileInfo getInfo( String path, String name)
                    throws RemoteException, InfoNotFoundException;
    ...
}
```

Invocação remota vs. invocação local

- Invocação efectuada através dum referência para objecto remoto
  - Necessidade de obter referência
    - Directório externo (rmiregistry)
    - Código pode estar disponível localmente ou ser carregado dinamicamente
  - Referência remota *(stub)* gerada automaticamente ou explicitamente (rmic)

- Passagem de argumentos e resultado por valor (cópia)
  - O servidor apenas pode influenciar o cliente pelo envio do resultado – alterações nos parâmetros não são reflectidas no cliente
  - Objectos propagados são serializados (é feita uma cópia e criado um novo objecto)
    - Mais detalhes numa próxima aula

- Necessidade de tratar excepções de comunicações
  - Todos os métodos remotos podem lançar a excepção *RemoteException* (para indicar erro de comunicações)

FERRAMENTAS

Directório de objectos:

rmiregistry

```
Naming.rebind( “//servername/objectname”, srv);
IFileServer srv = (IFileServer)Naming.lookup( “//servername/
objectname”);
```