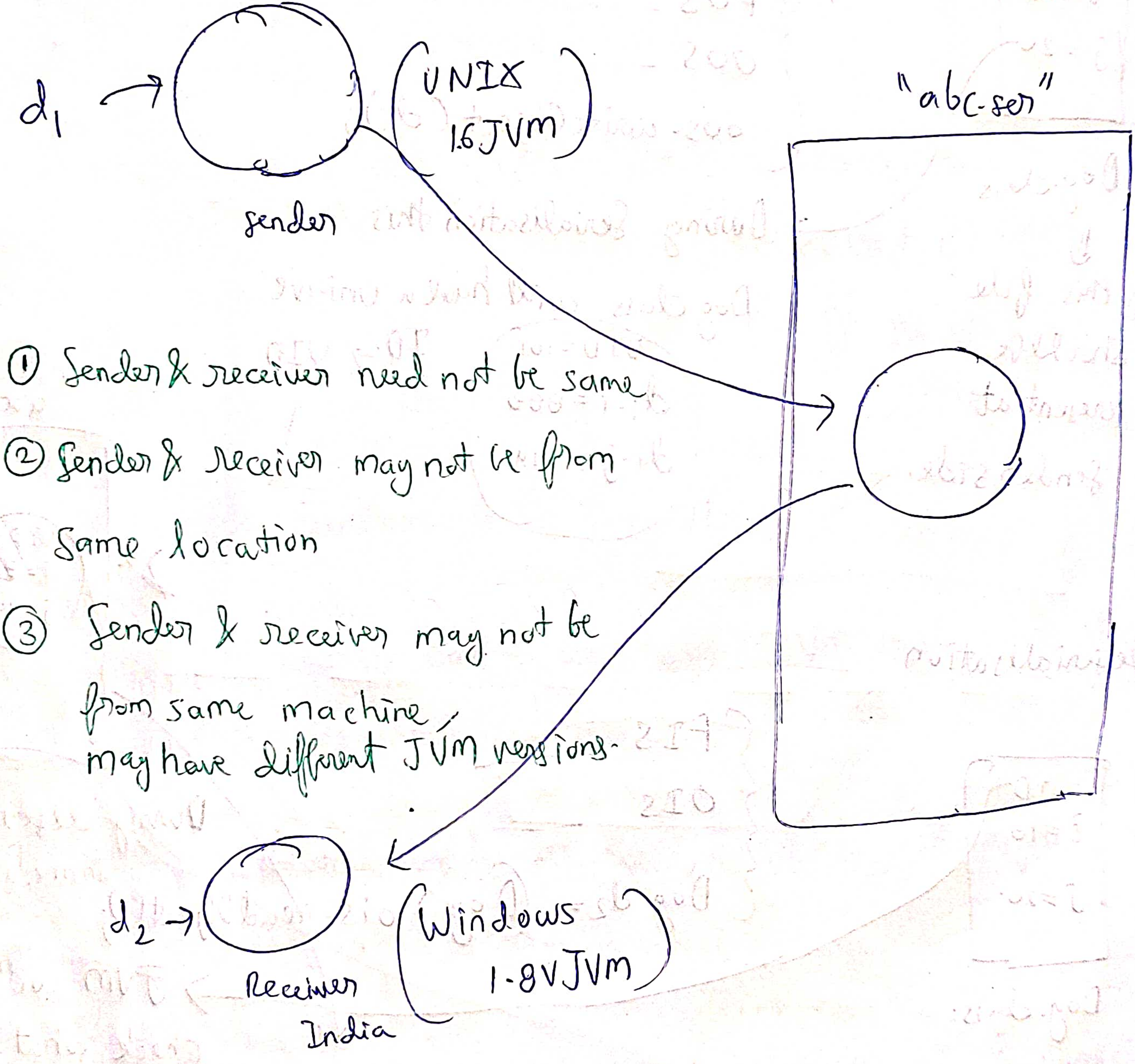
(Part 15)

Serial Version UID

```
private static final long serialVersionUID = 1L;
```



① Sender & receiver need not be same

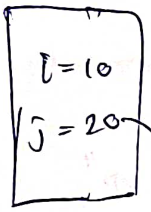
② Sender & receiver may not be from same location

③ Sender & receiver may not be from same machine, may have different JVM versions.

At sender side serialisation code will be executed

At receiver side deserialisation code will be executed

(Serialisation)



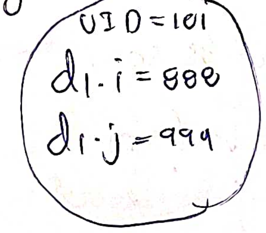
```
Dog d1= new Dog();
FOS ----
OOS ----
oos.writeObject(d1);
```

Dog.class

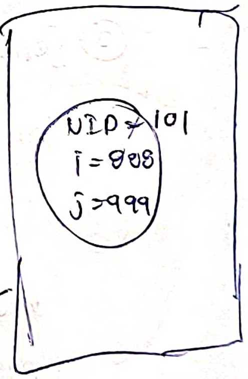
⇓

this file should be present at sender side

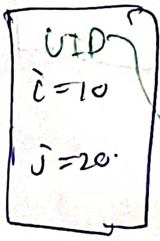
During Serialisation this Dog class will have a unique ID → UID



"abc.ser"



Deserialisation



```
FIS ----
OIS ----
Dog d2= (Dog) ois.readObject();
```

Dog.class.

⇓

this file should be present at receiver side as well.

During deserialisation immediately

JVM will check what is the UID of the class at receiver side.

If during deserialisation, JVM checks the UID in the Java class on receiver side, matches with the UID on the file or not.

If Matches: well and Good, object will get deserialised.

If does not match: JVM will throw Exception,
Saying (RE: Invalid Class Exception)
And hence state of the object will not be passed.

This (UID → serial Version UID) ⇒ {to identify whether its the correct object or not by receiver side JVM}

(If You Depend on JVM generation of Serial VersionUID)

[Problems] If JVM generates Serial Version UID

(1.) Both sender and receiver have to use same JVM version, or else serialVersionUID mismatch will happen.

(2) Both sender & receiver should use same Dog class file version.

If at sender side I make the change

```
Dog{ int i=10;
     int j=20;
     int K=30;  ⇒ this change
}.
```

will lead to change in generation of (Serial Version UID)

(3) Chance of Performance Also.
↳(Internally May use Complex Algo)

If we want to avoid any performance issues,

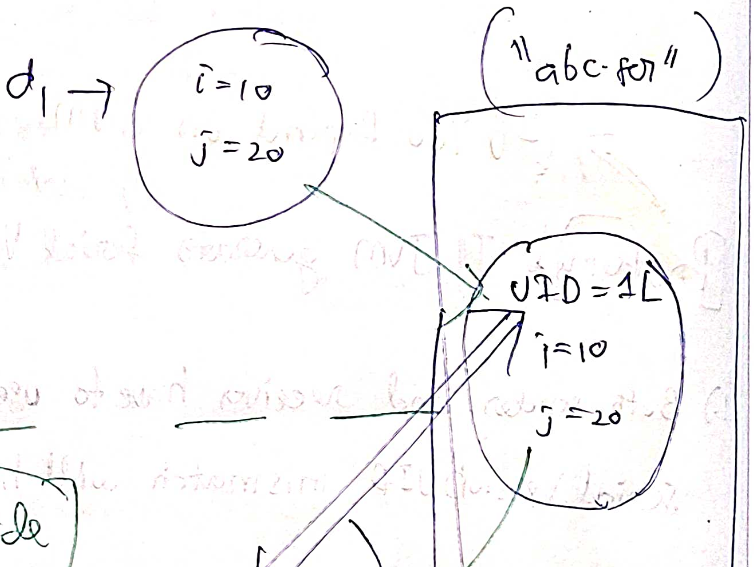
Its better to use custom serialisation.

(Part 16)

lets say that sender and receiver have different version of classes, and we don't let JVM generate the serialVersionUID, in that case,

```
class Dog{
 Private static final long serialVersionUID = 1L;
  int i=10;
  int j=20;
}
```

Sender Side



Receiver Side

```
Class Dog{
 private static final long serialVersionUID=1L;
 int i=10;
  int j=20;
 int k=30;
  int w=40;
}.
```

Proper serialisation, because serialversionUID matches.