A Distributed Object Model for the Java System
Wollrath, Riggs, and Waldo, USENIX 1996

Goals:
  transparent RPC for object methods
  avoid NFS-style explicit object handles
  automatic association of relevant server w/ object ref
  allow passing of object references as arguments
    not just to object's home server (as in NFS)
    even to other client hosts
  distributed GC, needed for remote refs

Situations in which one client might pass remote object ref to another?
  lots of modules: shopping cart, item db, checkout, front end

Are there other approaches?
  distributed shared memory, would allow direct access to object data
  move the object to caller

first a simple call/return
  o = ???;
  o.fn("hello");
  which server to send to?
  what object on server?
  what about "hello"?
  what does RPC message contain?
  how does RMI s/w on server gain control? thread...
  how does server find the real object?
  where does server-side dispatch fn come from?

what does a stub object look like?
  type?
  contents?
  where did it come from?

is there anything special about the server-side "real" object?

is "hello" sent as a remote object ref?

how about passing an object as an argument?
  o1 = ???;
  o2 = ???;
  o1.fn(o2);
  what must o2 look like in the RPC message?
    server host, object ID
  what if o1's server already knows about o2?
    must have a table mapping object ID to ptr to o2
  what if o1's server does not know about o2?
    where does it get stub type, implementation?
    can stub stuff be generated purely by client?

there are probably type IDs, so client can re-use stub code
  an object ID must contain type ID, or an RPC to fetch it
  clients and servers must have tables mapping type ID to stub code

when can a server free an object?
  only when no client has a live reference

```
  server must somehow learn when new client gets a reference
  and when client local ref count drops to zero
  so clients must send RPCs to server to note first/last knowledge
  what if C1 passes to C2, C1 sends de-ref RPC before C2 sends ref?

what if a client crashes?
  will server ever be able to free the object?
```