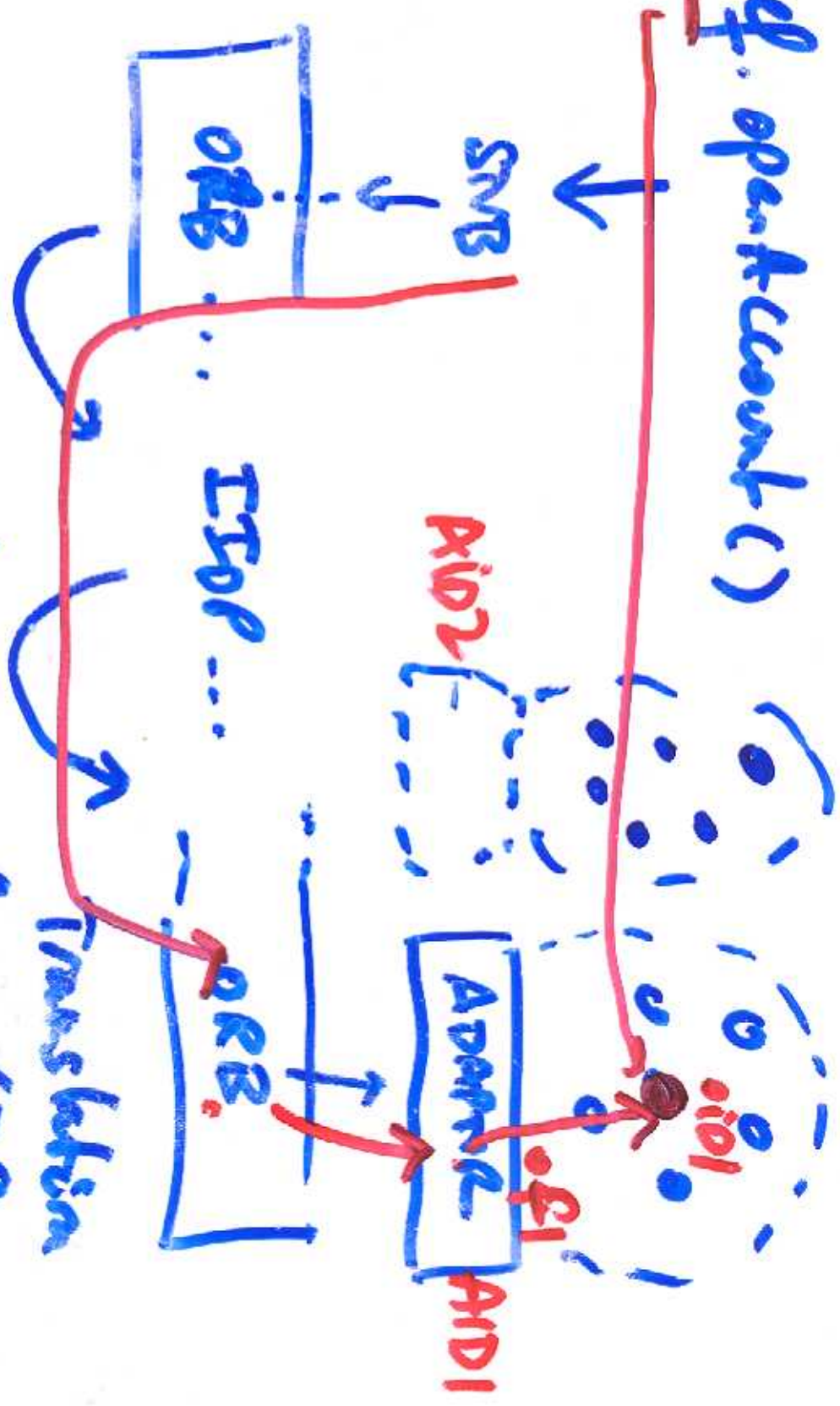






ref. open account (C)



Translation of

detached structures from

NATIVE representation to CDR

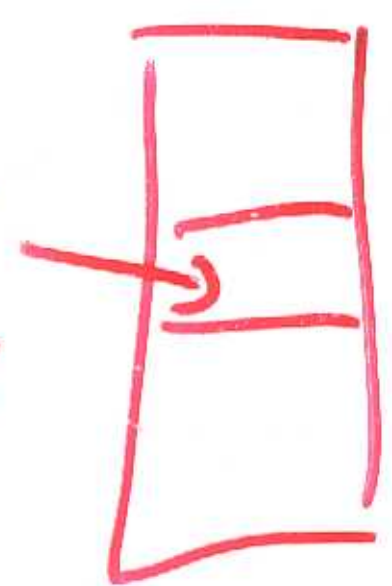
Translation

from CDR

to Native

Representation

ref



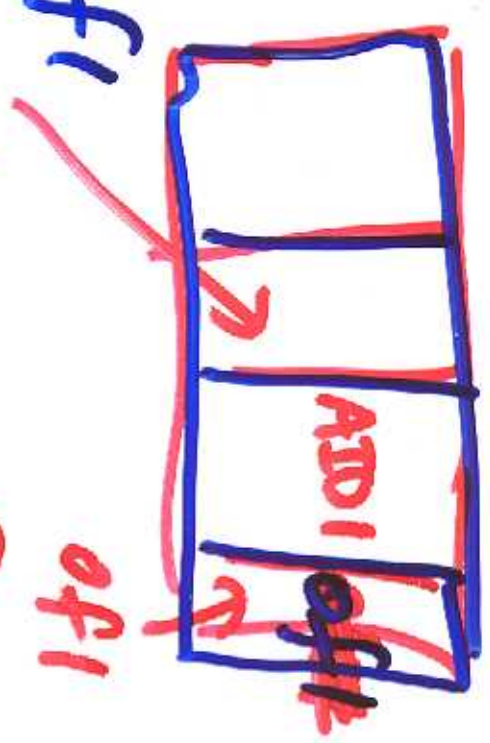
location

instance  
of CORBA  
OBJECT

USER  
SEES

ofi

location



instance  
CORBA OBJECT  
ID

[-] IMPLEMENTATION  
OBJECT ID

oid1  
e.g  
instance of C++ or  
Java object

Reference



CORBA WOELD

location

ADAPTER

INTER  
PARTION

CORBA

OBJECT

CORBA  
OBJECT

ADAPTER

MAP  
TABLE

IMPLEMENTATION

WOELD

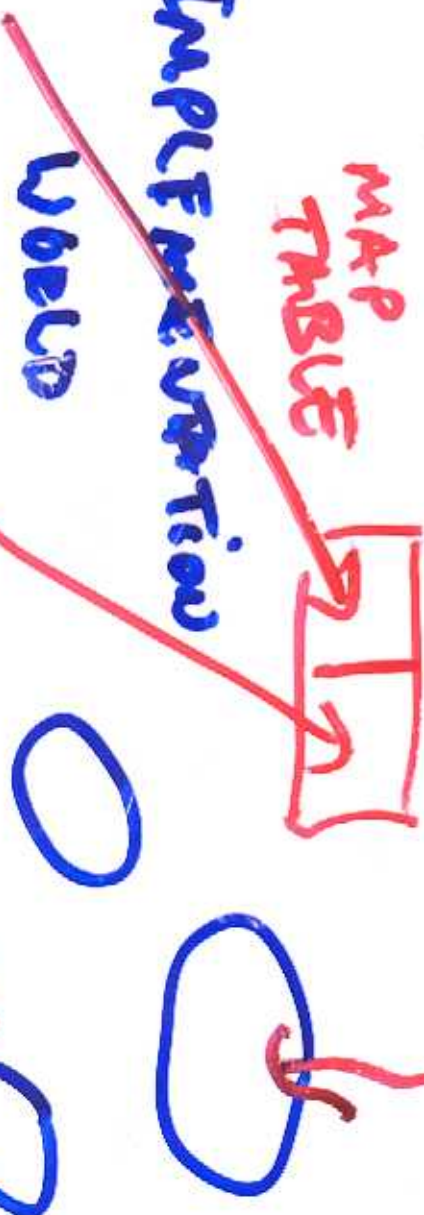
CORBA

(C++)

IMPLEMENTATION  
OBJECT

OID

IMPLEMENTATION OBJECTS



ref



+ operation

CorBA oid

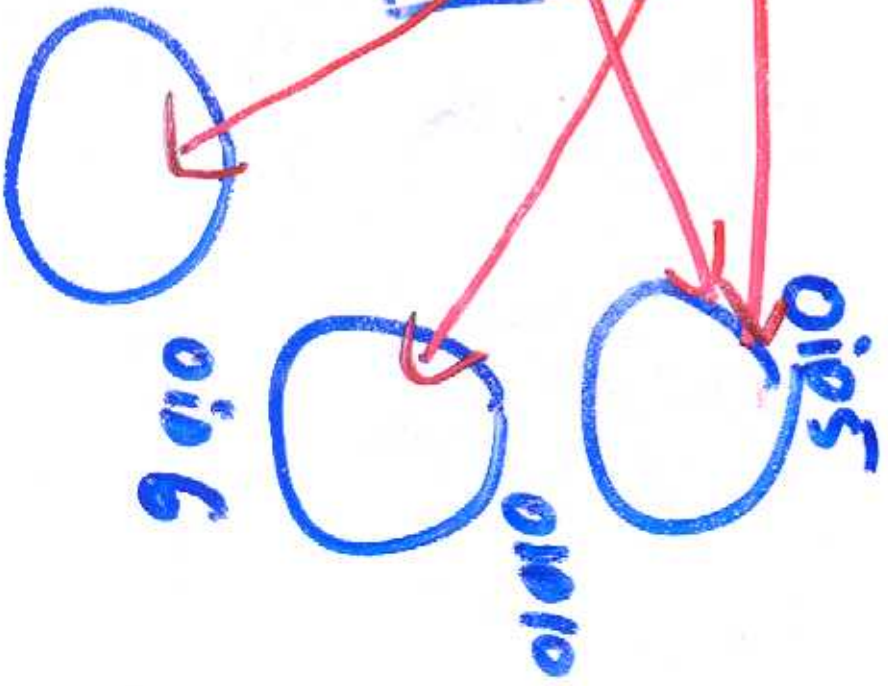
REQUEST

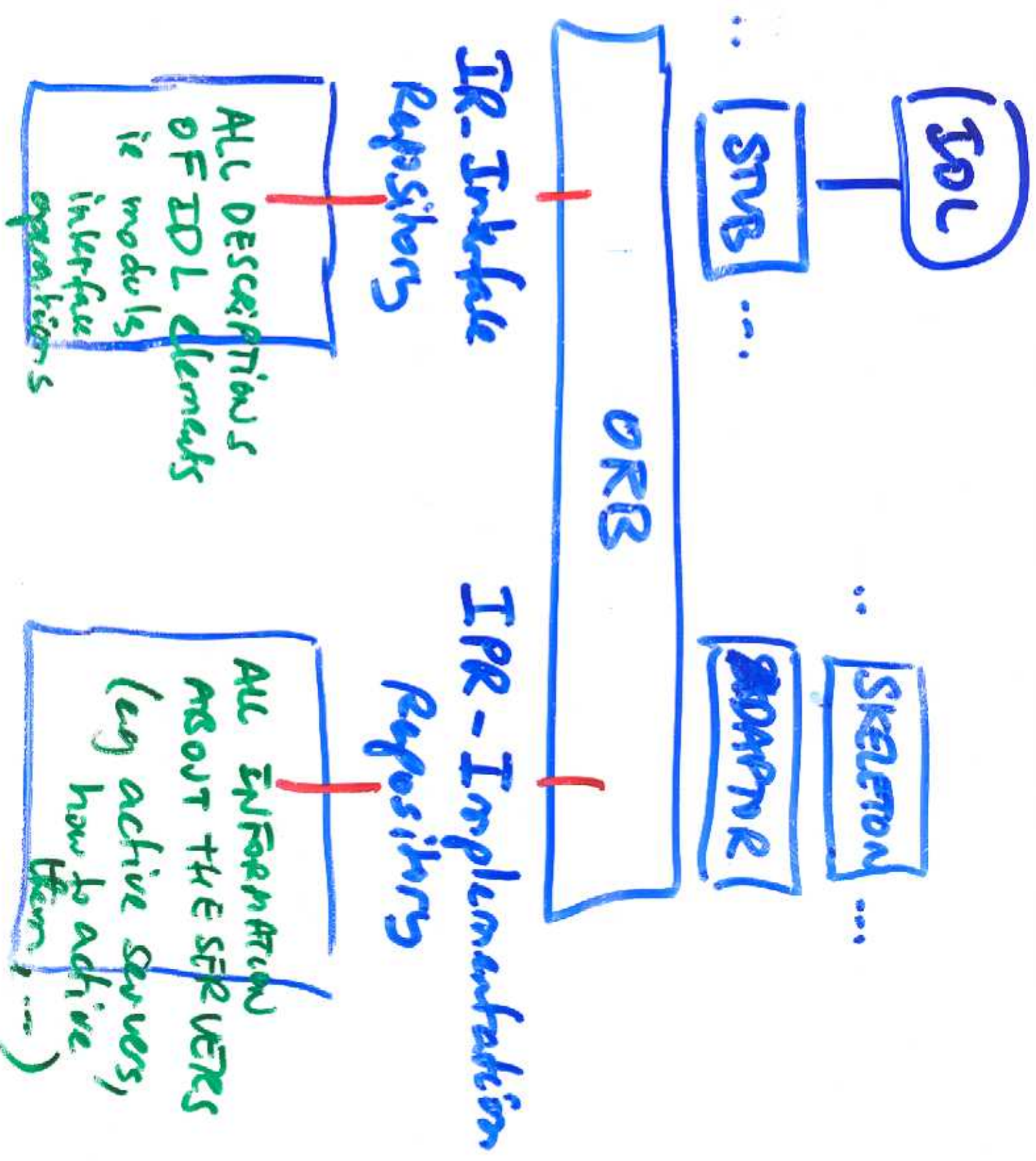


of1	oid5
of2	oid10
of10	oid5
of15	oid6

IMPLEMENTATIONS  
OBSERVER  
(C++)

MHP  
TABLE





IDL

SMR

ORB

SKELETON

DAPNR

IR-Interface

Repositories

IPR-Implementation

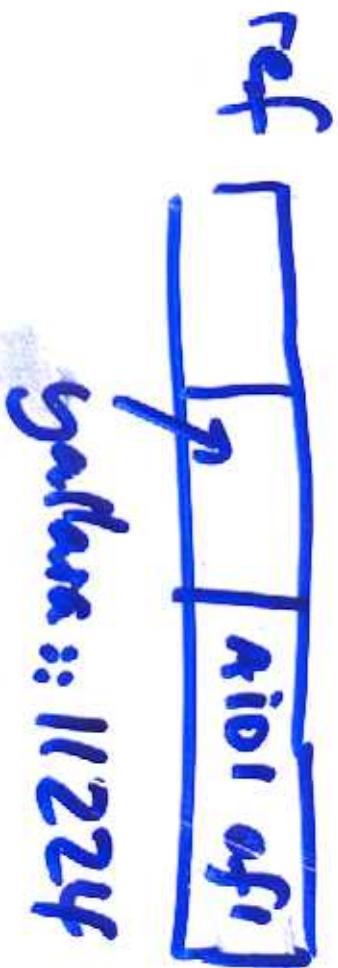
Repositories

ALL DESCRIPTIONS OF IDL elements ie module interface operations

ALL INFORMATION ABOUT THE SERVERS (eg active servers, how to activate them, ...)

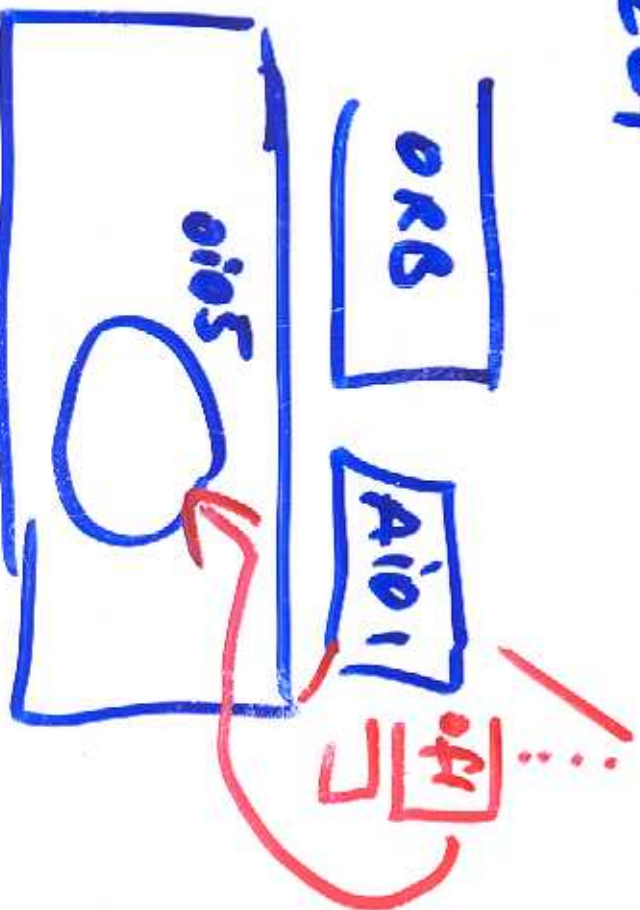
# ① "TRANSIENT" REFERENCES

- ORB does not have an implementation repository



ORB ...

SERVER  
is STILL  
RUNNING



LATER (loan) -- SERVER IS  
DEACTIVATED

LATER (1/2k) ...

SERVER IS  
ACTIVATED AT  
DIFFERENT

Port e.g

yellowa :: 10521

## ② Dynamic invocation (DII)

"You" (as user) can  
invoke operations that  
are NOT known at  
compile (i.e. SWIS  
does NOT know how  
to MARSHAL THE  
OPERATION)

Step 1 ...

interface Account

{ op1

op2 }

YOU WANT USE  
OPERATION @ CLIENT  
SIDE  
op3

DI

if knows

~~SOB~~

ONLY op1

and op2

Step 2

@ Server side,

in addition to  
implementations  
of op1 and op2  
you have introduced  
and another  
operation (say  
op3)

# DIJ

- create a 'DIJ' object
- assign values to this object
  - \* reference (you want to call op3)
  - \* input
  - \* output

# META-CLASS / META-TYPE

e.g. C++ class

