

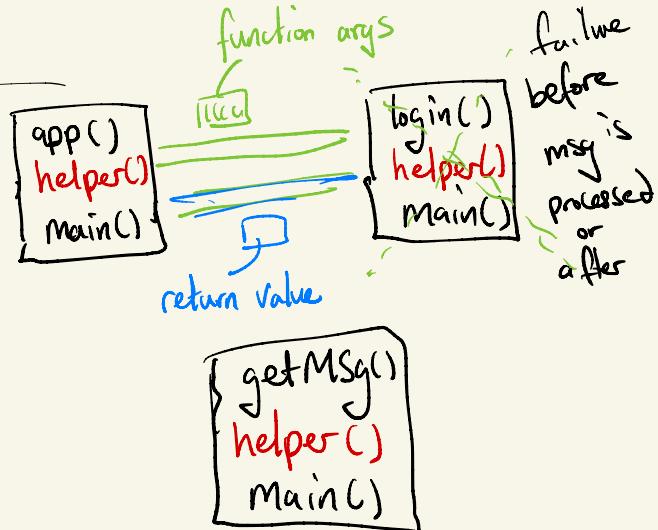
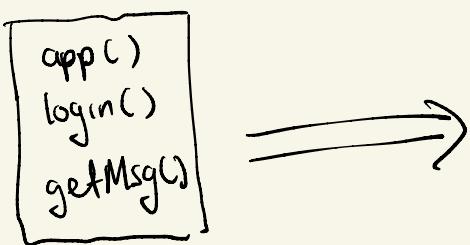
GS|380

DAY 4 [Feb 2]

Theophilus Benson



Last Class



helper Code (app.)

create socket

(line 74 - 101)

convert args into byte array
Sent array over socket to "other server" (line 106)

Wait()

(line 108)

get data over socket

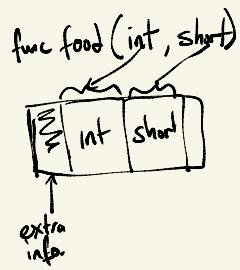
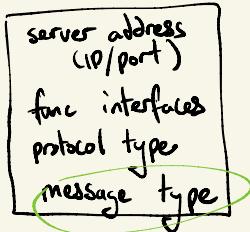
(line 112)

convert data into return value

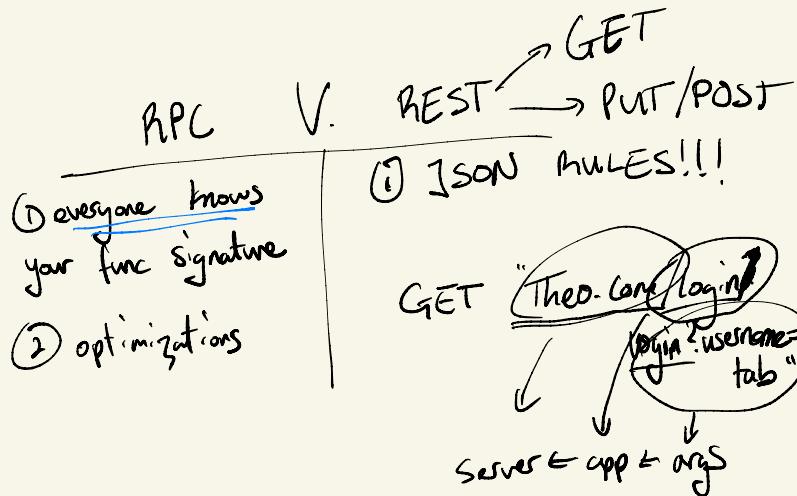
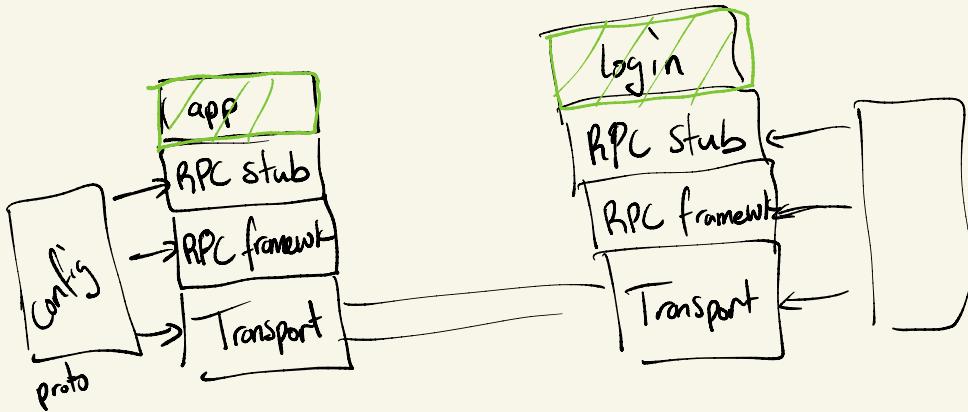
RPC (remote procedure call)

- ① make network connections (if keep track)
- ② take func arguments & convert into packets
- ③ packets into return values
- ④ failures: what are they? How to react!

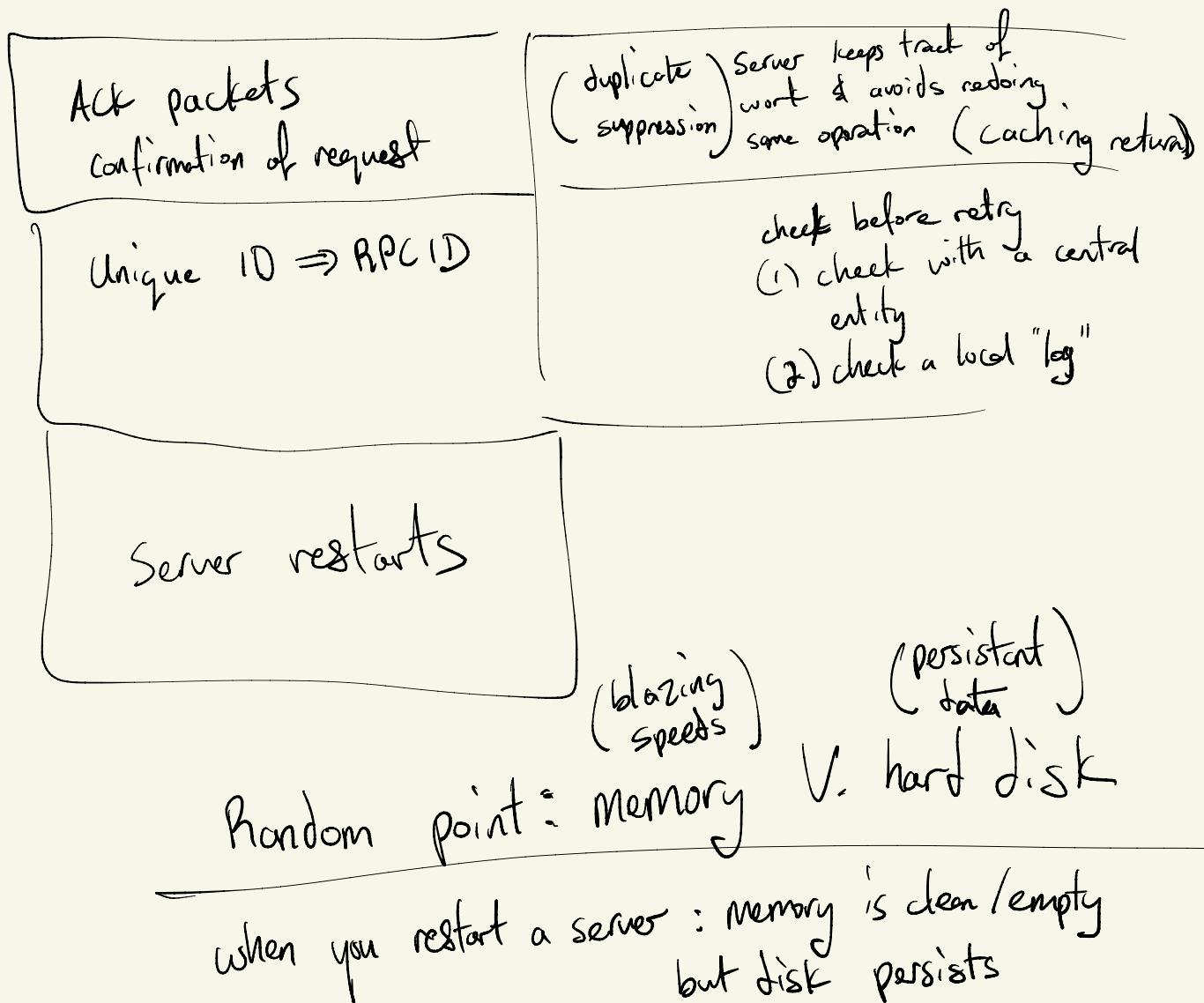
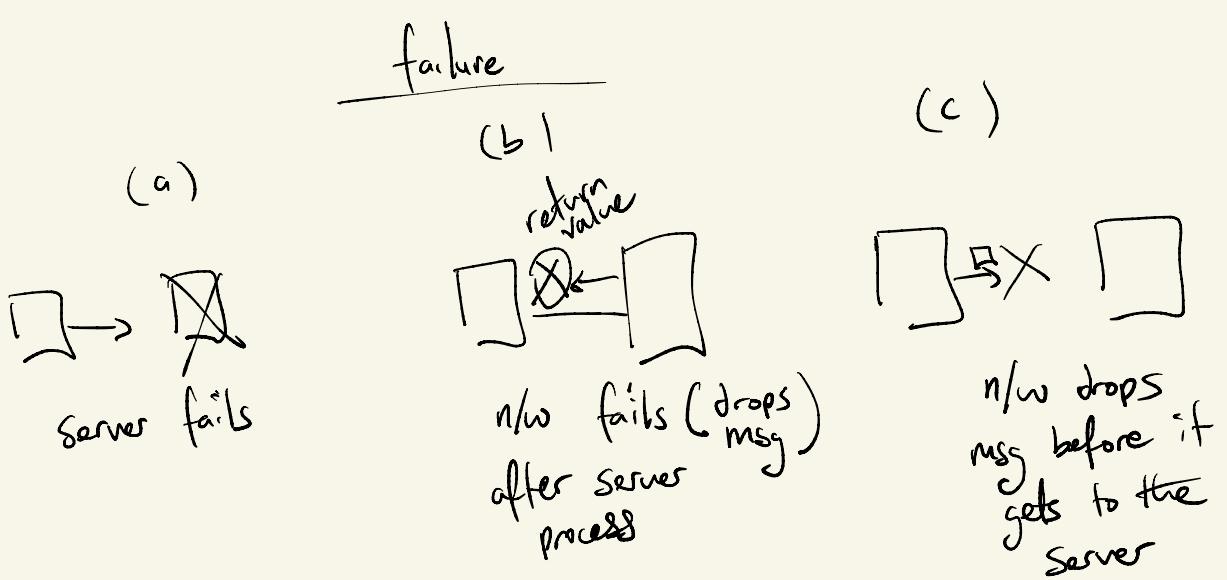
proto (protobuf)



arg / return \leftrightarrow msgs / packets



RPC - framework ; failures
HTTP error codes



Request retry: set a timeout & if a reply isn't received within that time then retry the request

Suppress duplicates: maintain a "log" of processed requests. Check every request against the log; if you notice a duplicate then drop & ignore this request.

Response replay: maintain a "log" of processed requests & their responses. Check every request against the log; if you notice a duplicate then reply with the cached response

Response replay
identify duplicates &
respond with cache
response

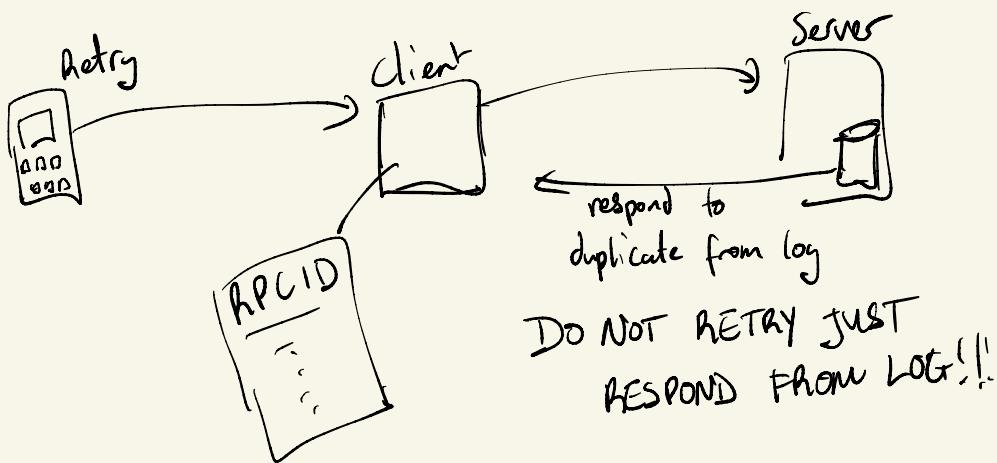
✓. Duplicate suppression
identify duplicates &
ignore them

Both prevent the system from doing duplicate work

RPC (Semantics)

(Impossible)

	at-least-once (1 or more time)	at-most-once (0 or 1)	exactly-once 1
Retry	✓	✓	✓
Suppress duplicates	Yes (3) Many (NB)	✓	✓
Replay (server side) response	Yes (Many) No (4)	✓	✓



I dempotency : doesn't matter how many times you call this function \Rightarrow only happens once

append	✓	write	location in file
append (file, "hello")		write (file, (20), "hello")	
20x		20x	
Not Idempotent		Yay! Idempotent	

RPC

- (1) RPC V REST
- (2) RPC Config file (proto)
- (3) failures (3 semantics)
- (4) Idempotency (at-least-once)
- (5) illustrated some of benefits of RPC
(writer less code)