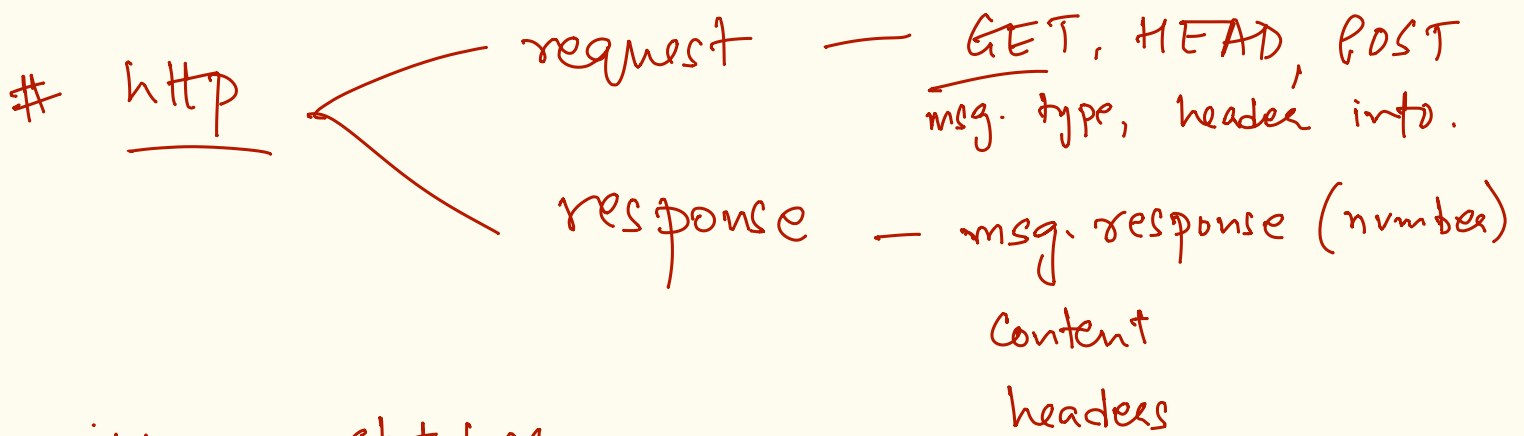


CS744

Lecture 22



- http ~ stateless

(i)

every req. is independent
& has to be completely specified
no implication on action due to
prev. requests.

(ii) performance axis

http 1.0	http 1.1	http 2	http 3
- basic specification - stateless - one connection per object/resource	- persistent connections - single TCP connection for multiple objects	~ parallel connections - per multi-streams on a single TCP.	

(iii) session & cookies

- session id

↑
negotiated via
a user authentication

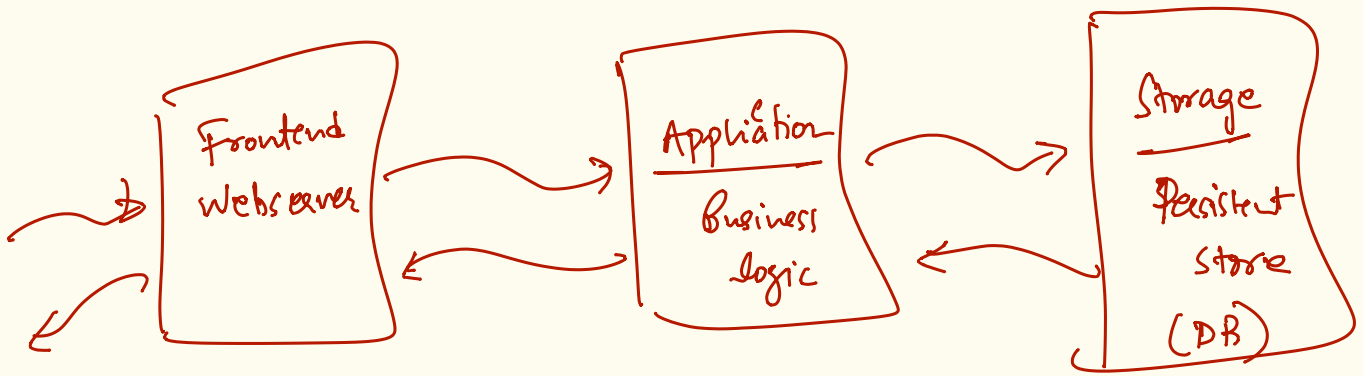
info about access stored on
client m/cs.

- http requests to a
server will be accompanied
by the cookie.

- ids fetched by http.
- applⁿ. interpret ids
from requests for
responses.

- cookie interpreted by
the applⁿ. for
client/user info.

(iv) http msg formats



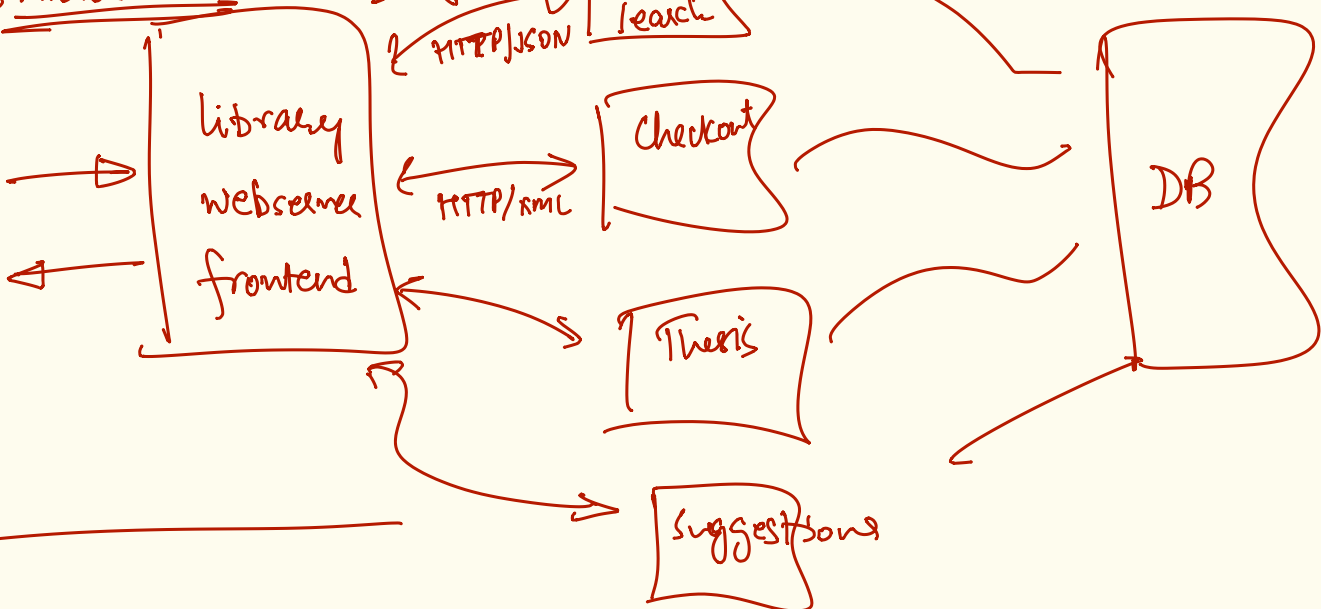
JSON ~ } text-based, key-value specifications for msg. content.
XML ~ }
Protobuf ~ } binary

development web, platforms

```
{ "name": "Dazzlightyear",  
  "destination": "infinity" }
```

web applⁿ design

microservices



② other examples — n fs } — protocols
a ftp }

③ platform / devp. primitive.

~ RPC — remote procedure call

client fn call is "remoted"
and executed on a server.

fn. info {
 which fn.
 arguments

 how ~~to~~ to send response back

 msg format.

browse("cs744");
 ↑
 fn. name

~ rpc library

— registering fns. / defining fns at the client & server.

— build state for fn. its, # arguments..

— rpc msg. format {
 serialized info. about
 the fn. call / response
 marshalling / unmarshalling.

— stubs at client & server
 {
 rpc wrapper / fn.

— manage n/w connections & transport.

eg: Flyt. ~ GPU virtualization



IPC

inter-process communication
primitive



```
graph TD; A[inter-process communication primitive] --> B[local]; A --> C[network]
```

- network sockets
 - unix domain sockets.
- Shared memory
- pipes / named pipes
 - signals
- message queues.