

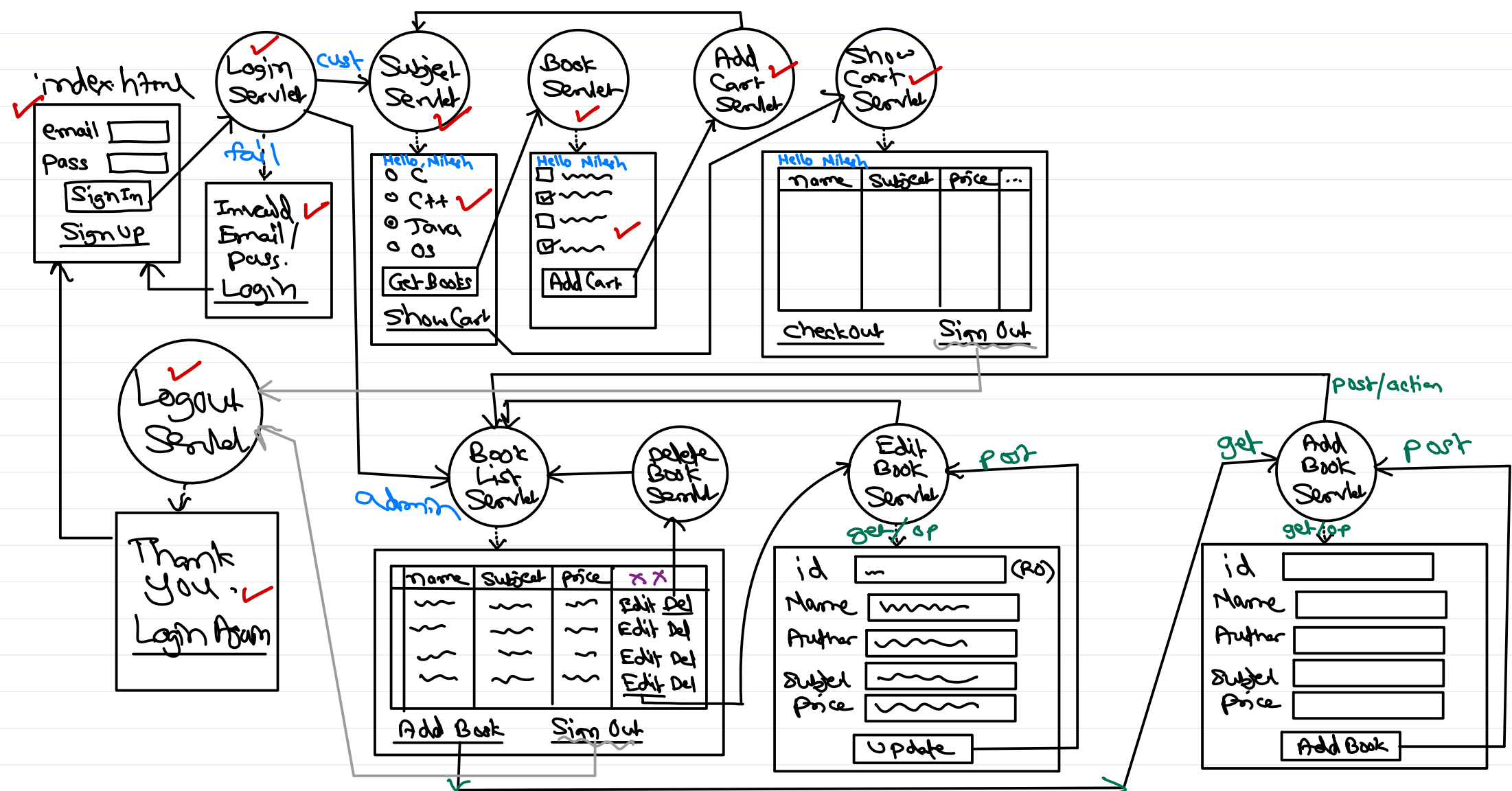


Advanced Java

Trainer: Nilesh Ghule

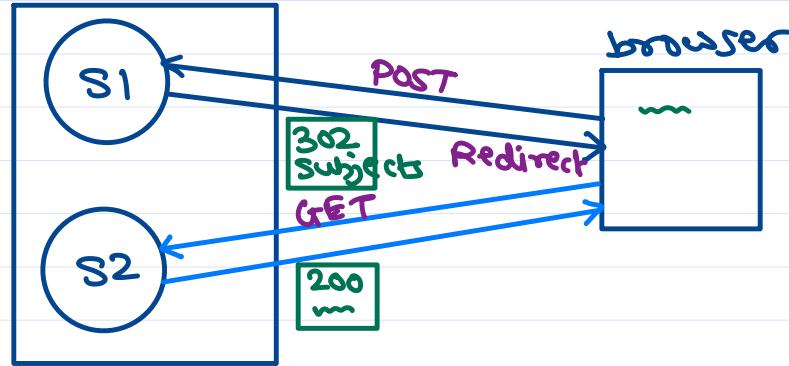


Bookshop using Servlets



HttpRedirection

```
resp.sendRedirect("subjects");
```



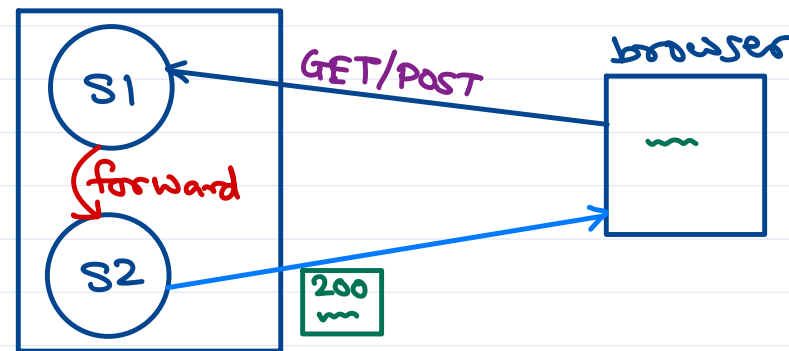
PRG design pattern.
= POST + Redirect + GET.

HTTP redirection:

`resp.sendRedirect()` generates a temp resp with status=302 & location of next page. Then browser makes GET req to next page to see final resp. Using this, we can go to any page in or out of our appn.

RequestDispatcher

```
rd = req.getRequestDispatcher("S2");  
rd.forward(req, resp);  
rd.include(req, resp);
```



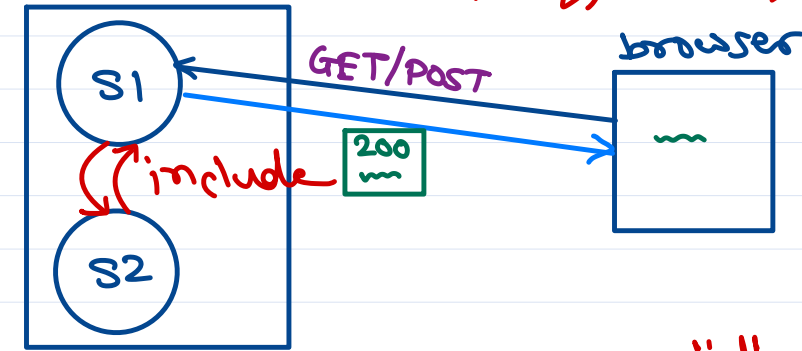
S1 process req partially & hand over same req to S2. Then S2 generates final response.

Using this we can navigate to any page within our appn.

In redirection, browser is aware of navigation (url change); but in req dispatcher browser is not aware of navigation (url of S1).

Since it is single req, it is faster.

If refresh, earlier post req might send data & may cause duplicate reqs on server.



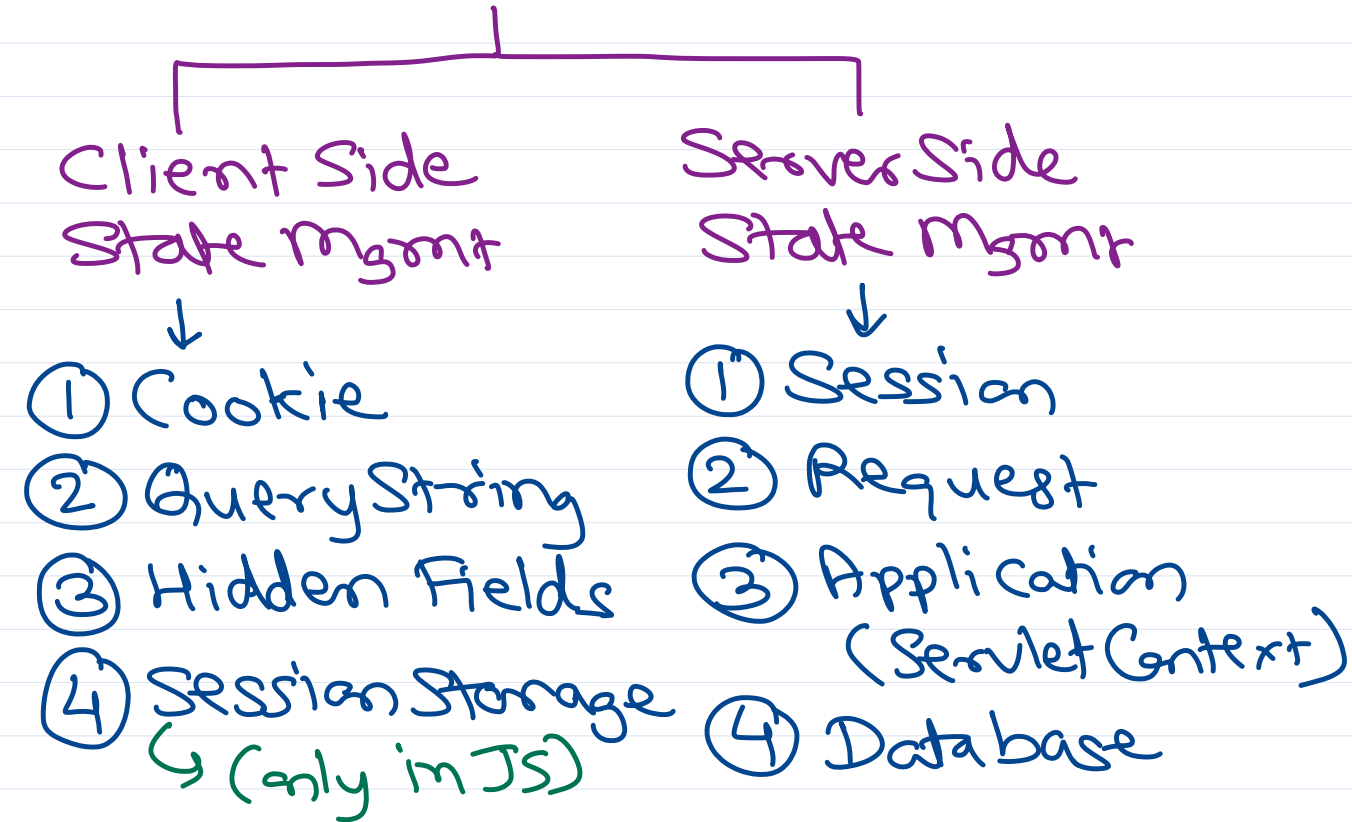
S1 process req partially & send to S2. It produces partial resp & return back to S1. Then S1 generate final resp.

Faster than redirect & slower than forward.

Req similar to fwd.

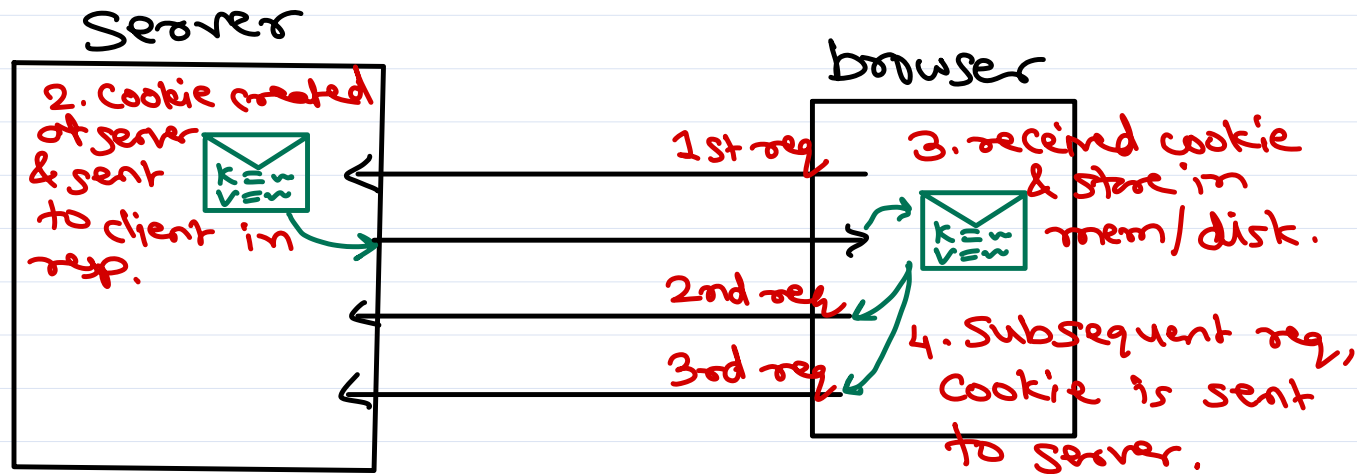
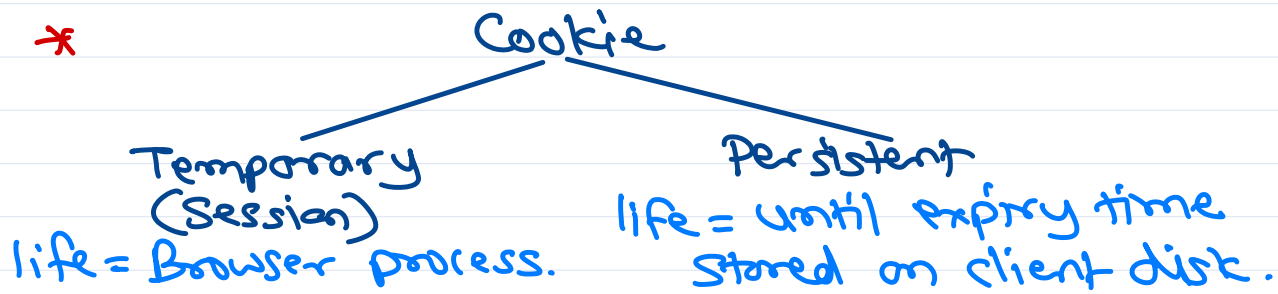
State Management

storing info about client
= state management.



Cookie 🍪

- * client side state management mechanism.
- * keeps data in key = value pair.
- * Cookie keeps text data (not binary).
- * max cookie size = 4 KB.
- *



* Send cookie to client

```
Cookie c = new Cookie("key", val);
c.setMaxAge(Seconds);
resp.addCookie(c);
```

to make cookie persistent.

if Seconds = -1 (-ve), cookie will be deleted from client disk.

* Receive Cookie from client

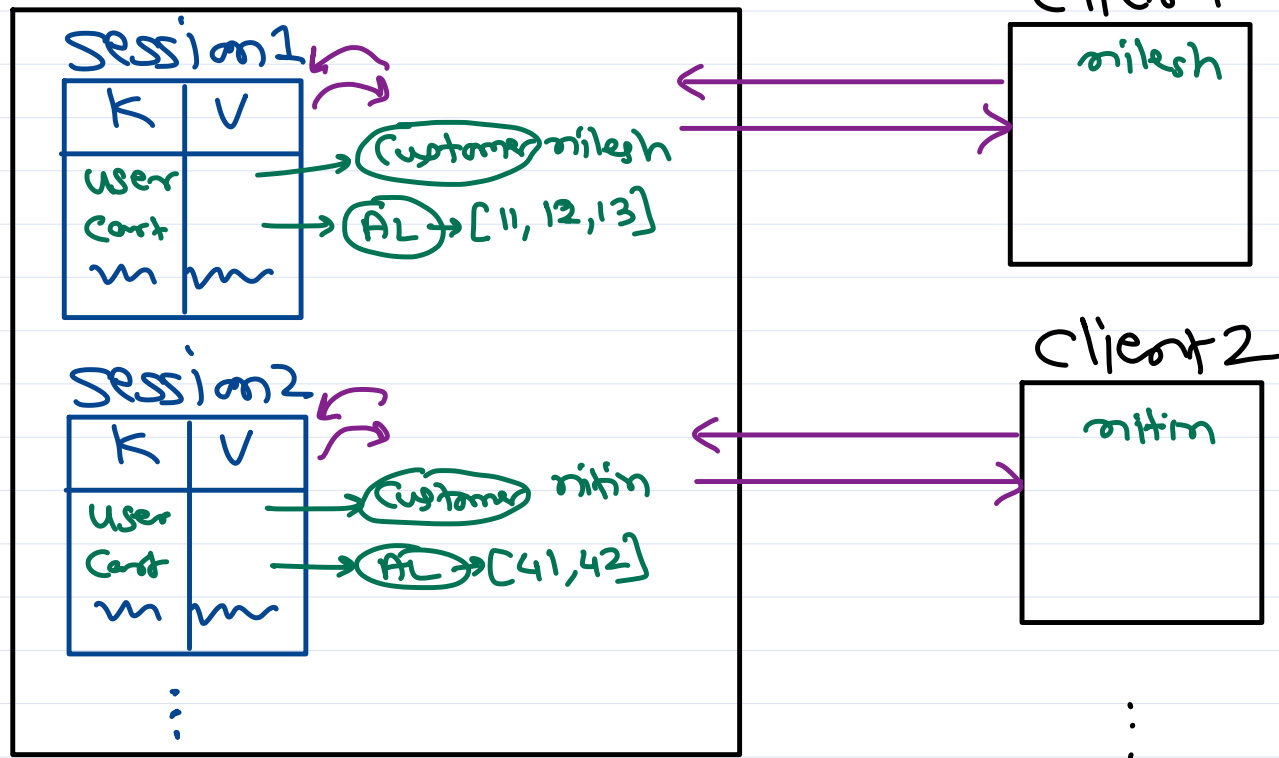
```
Cookie[] arr = req.getCookies();
for (Cookie c: arr) {
    if (c.getName().equals("key")) {
        val = c.getValue();
        break;
    }
}
```



Session

- * Server side state mgmt mechanism.
- * Like a hash map (key-value pairs) stored on server for each client.
- * Key (String) → value (Object).

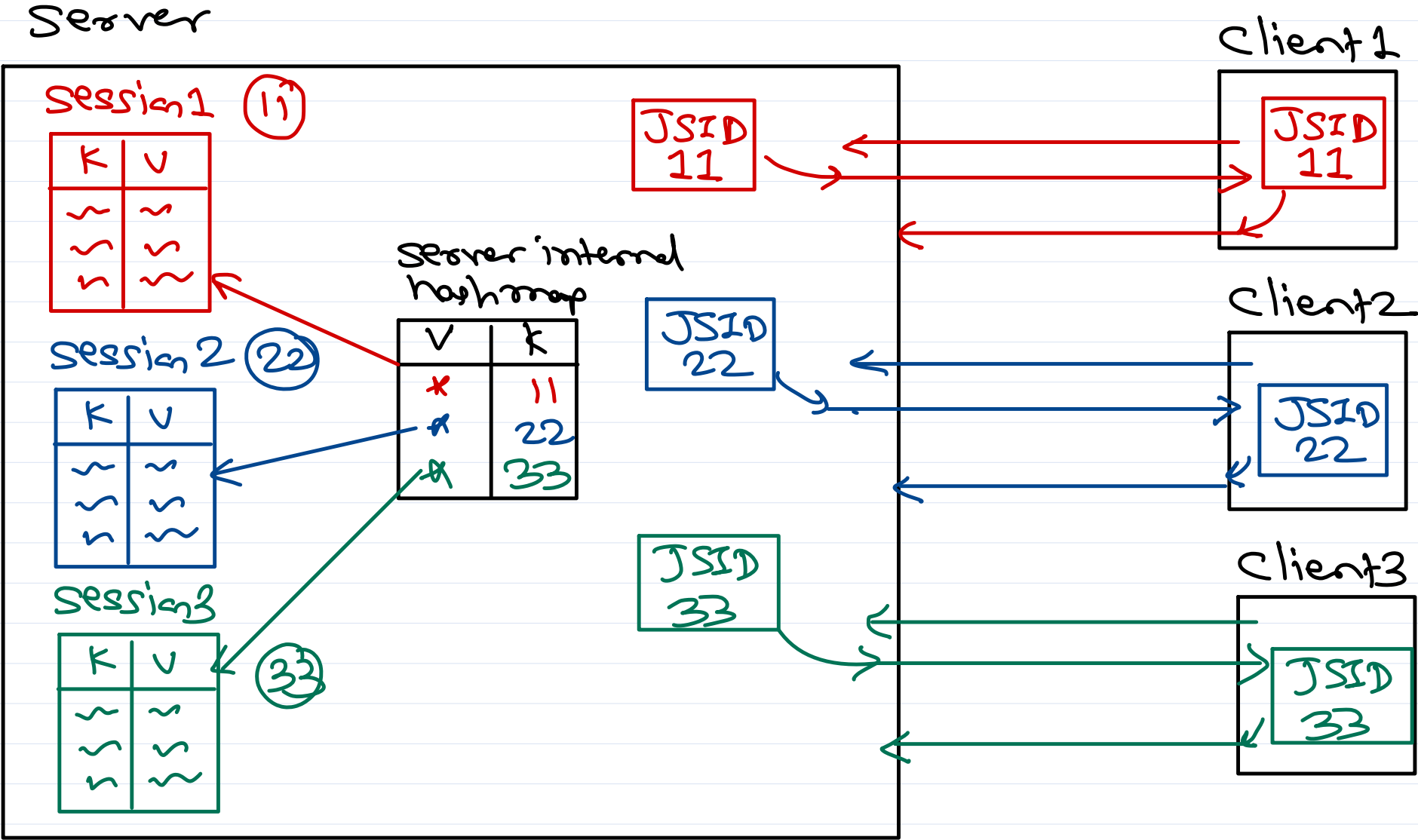
Server



- * Access current user session
`HttpSession session = req.getSession();`
- * To store key-value in session
`session.setAttribute("key", value);`
- * To retrieve value from session
`value = session.getAttribute("key");`
- * Destroy the user session
`session.invalidate();`



Session tracking





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

