

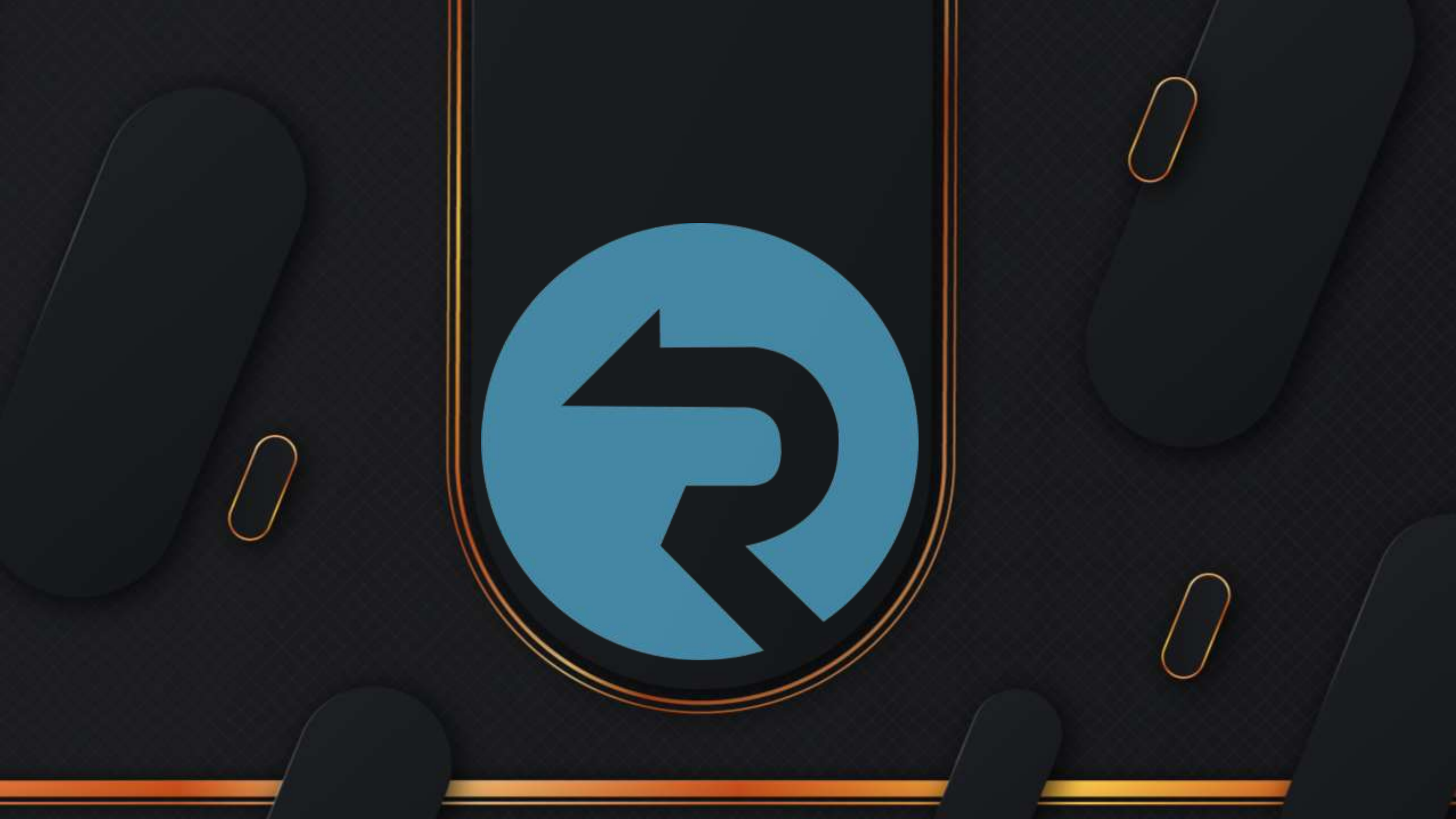
SignalR

THE COMPLETE GUIDE



BHRUGEN PATEL

www.dotnetmastery.com



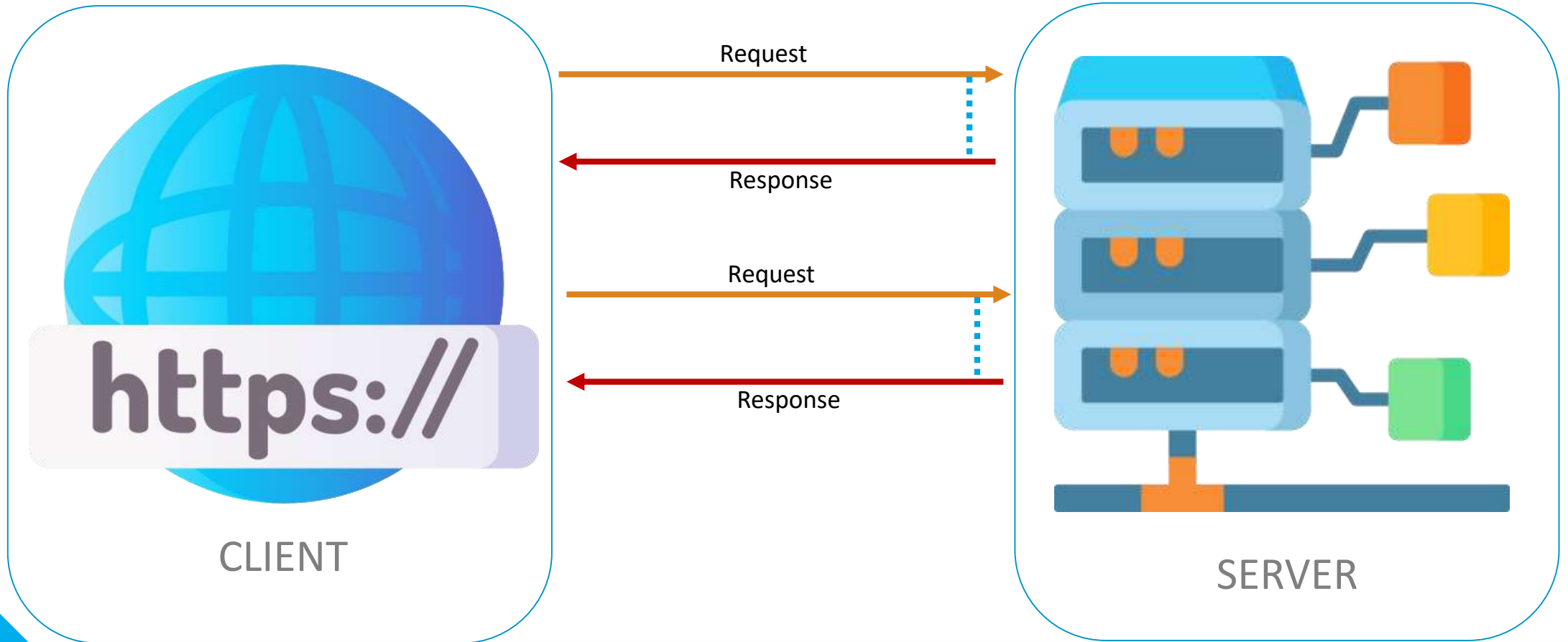
Signal **R**

Topics Covered

- ✓ SignalR Fundamentals
- ✓ SignalR .NET Client
- ✓ SignalR Identity User
- ✓ SignalR Groups
- ✓ 8 Different Projects



HTTP



SignalR Transport Type

01

Web Sockets

02

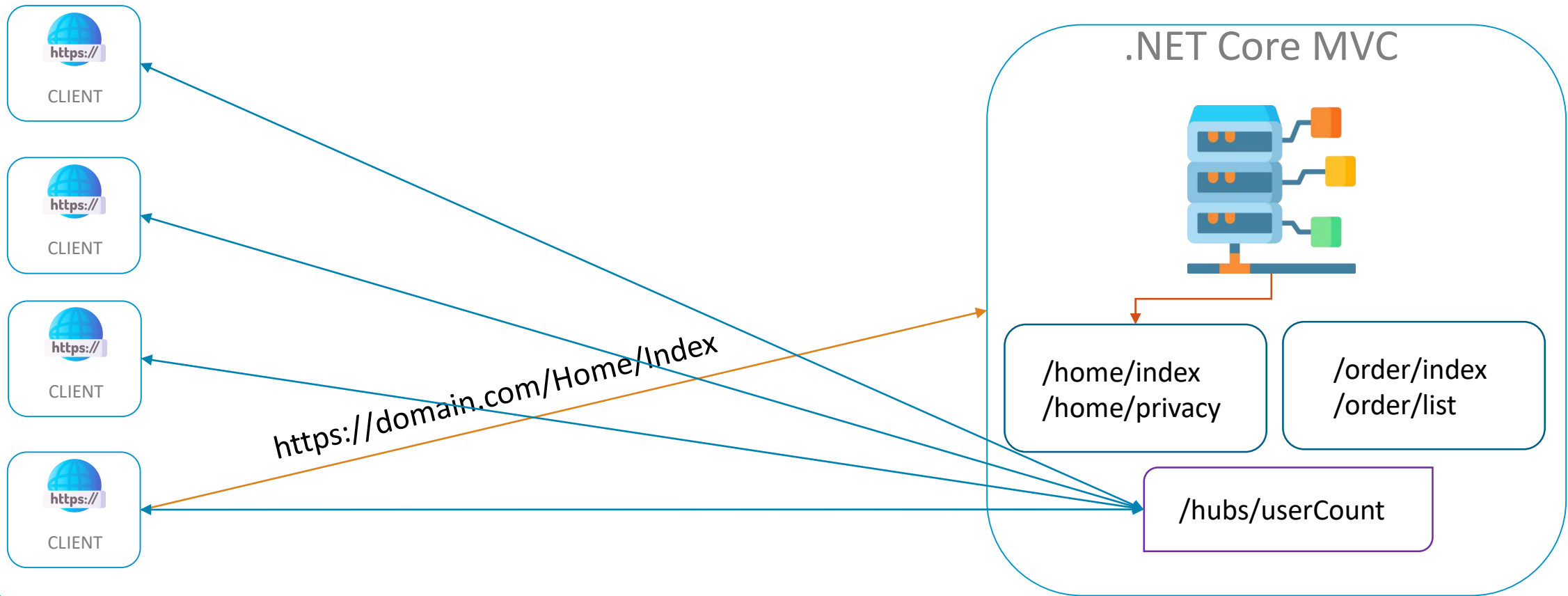
SSE (Server Sent Events)

03

Long Polling

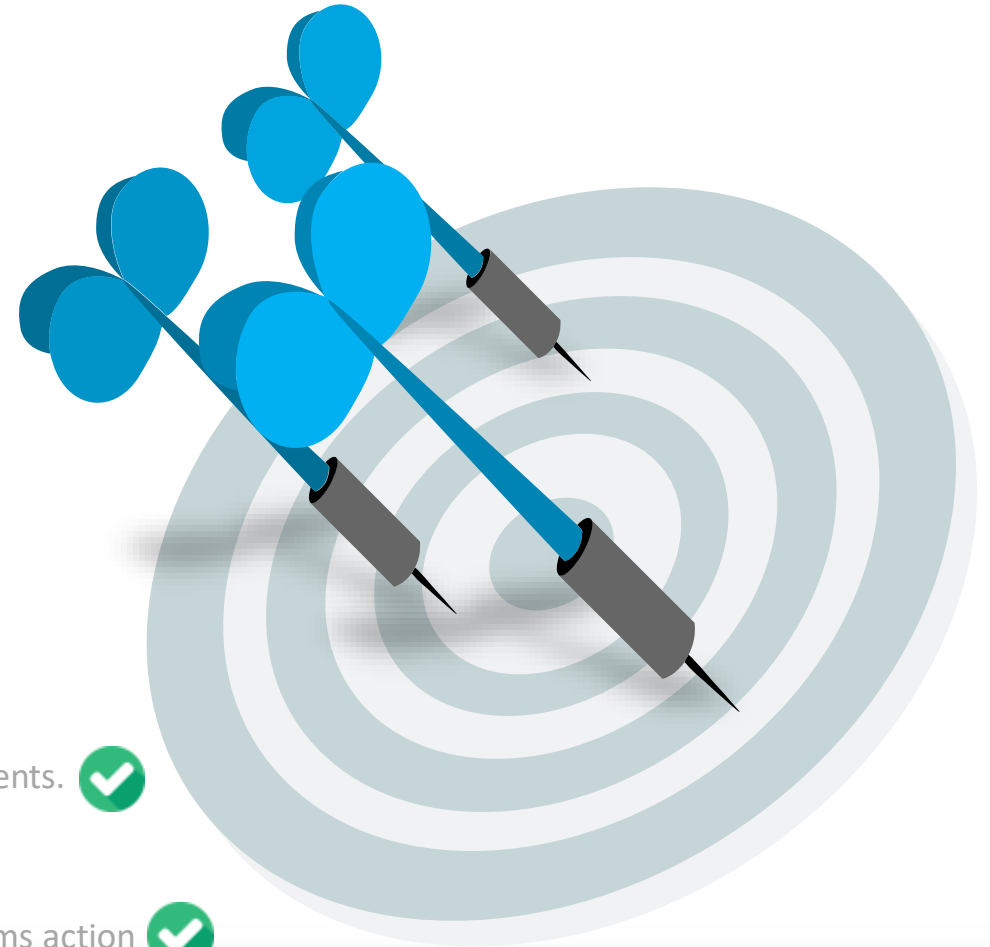


Hubs SignalR

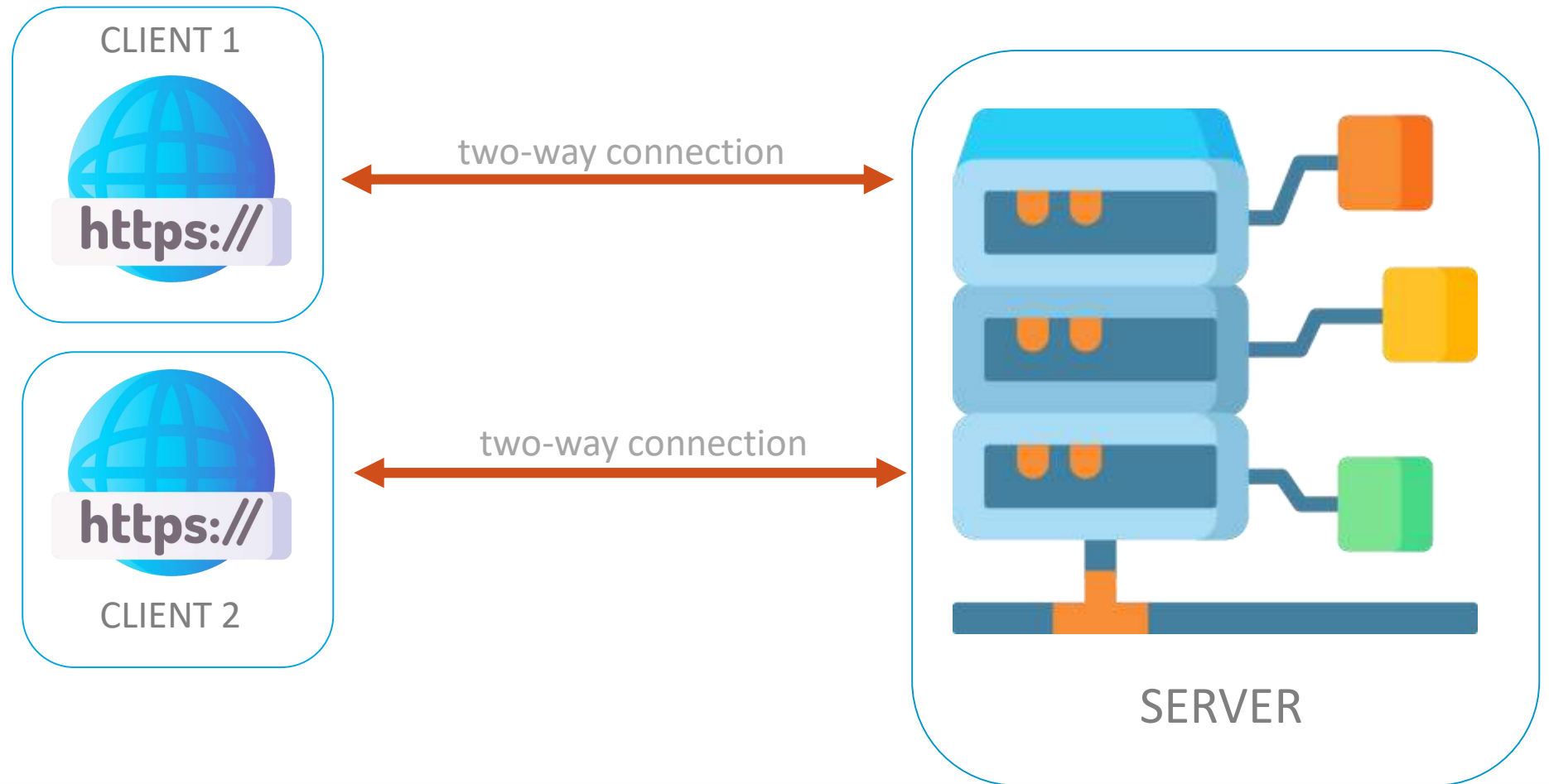


SignalR Typical Flow

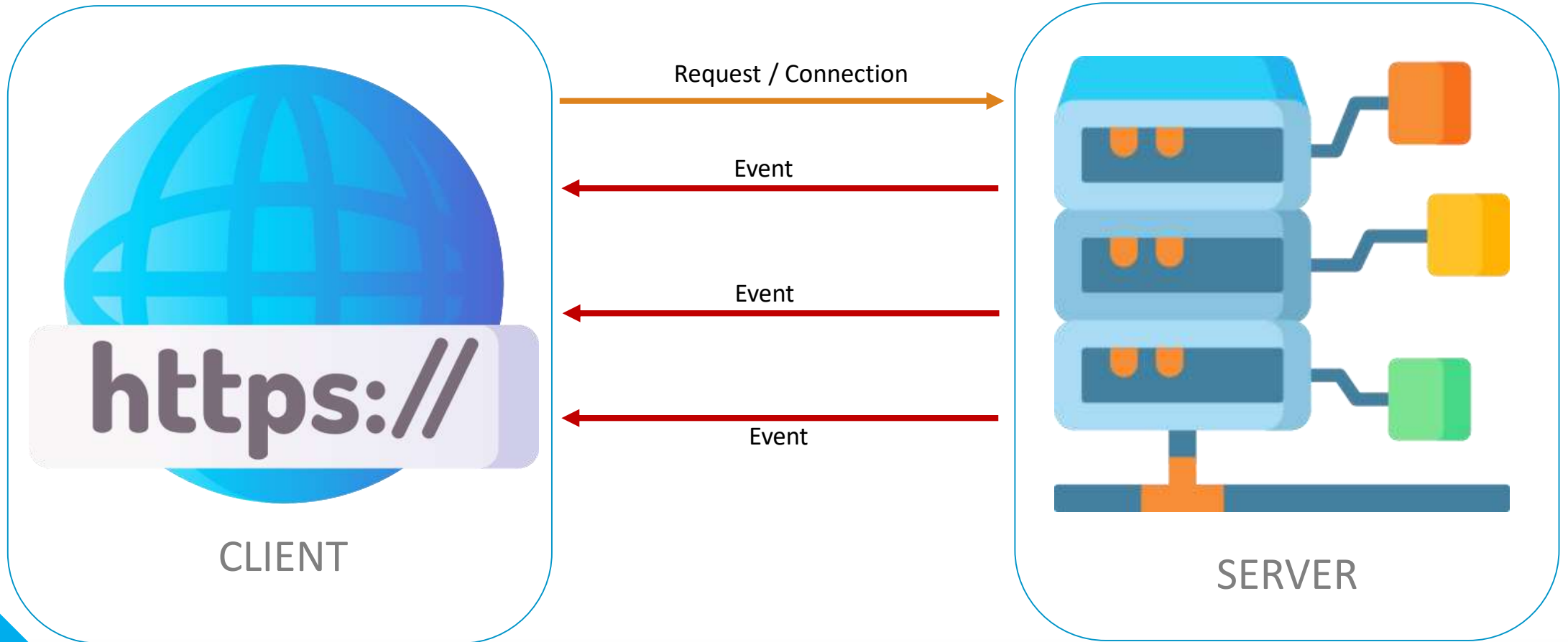
- 1 Create SignalR Hub ✓
- 2 Add Methods to Hub ✓
- 3 Add Client side SignalR ✓
- 4 Connect to SignalR Hub from Client JS ✓
- 5 Call SignalR Hub method ✓
- 6 SignalR Hub invokes method in Client JS to notify clients. ✓
- 7 Client receives update from SignalR hub and performs action ✓



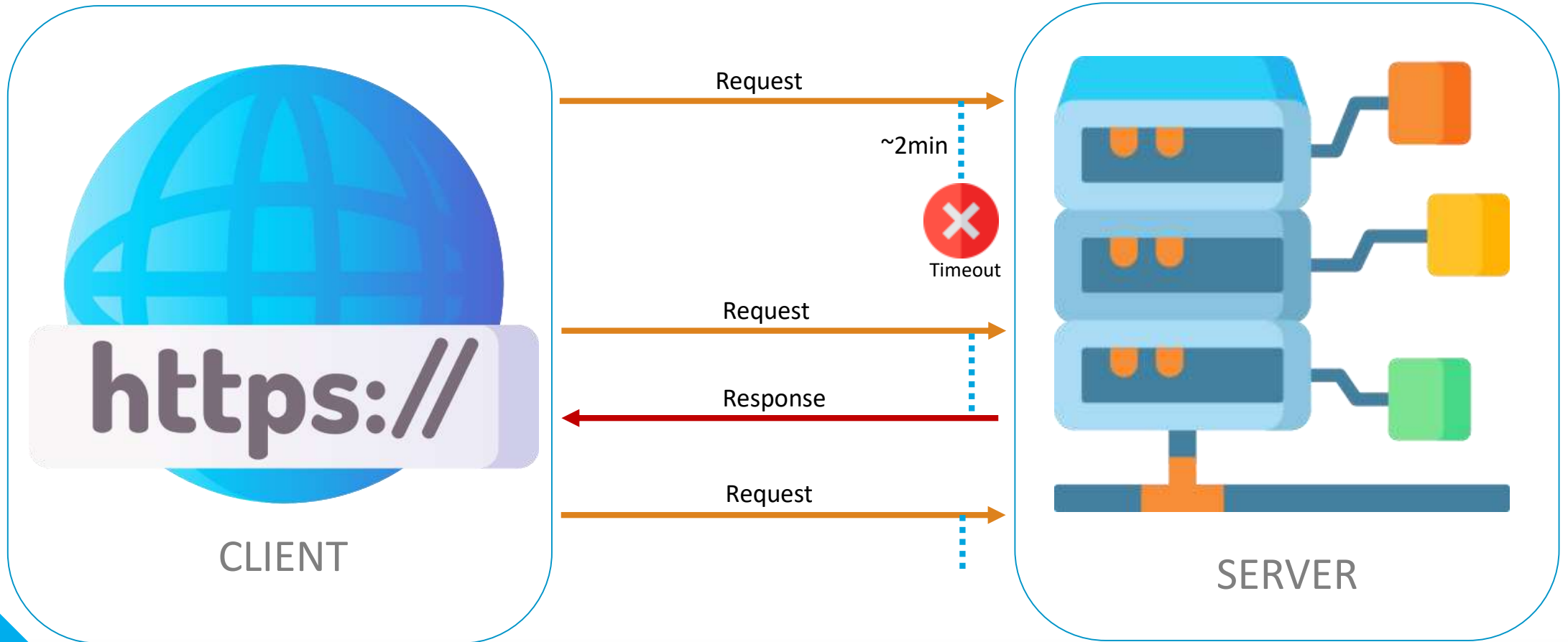
WEB SOCKETS



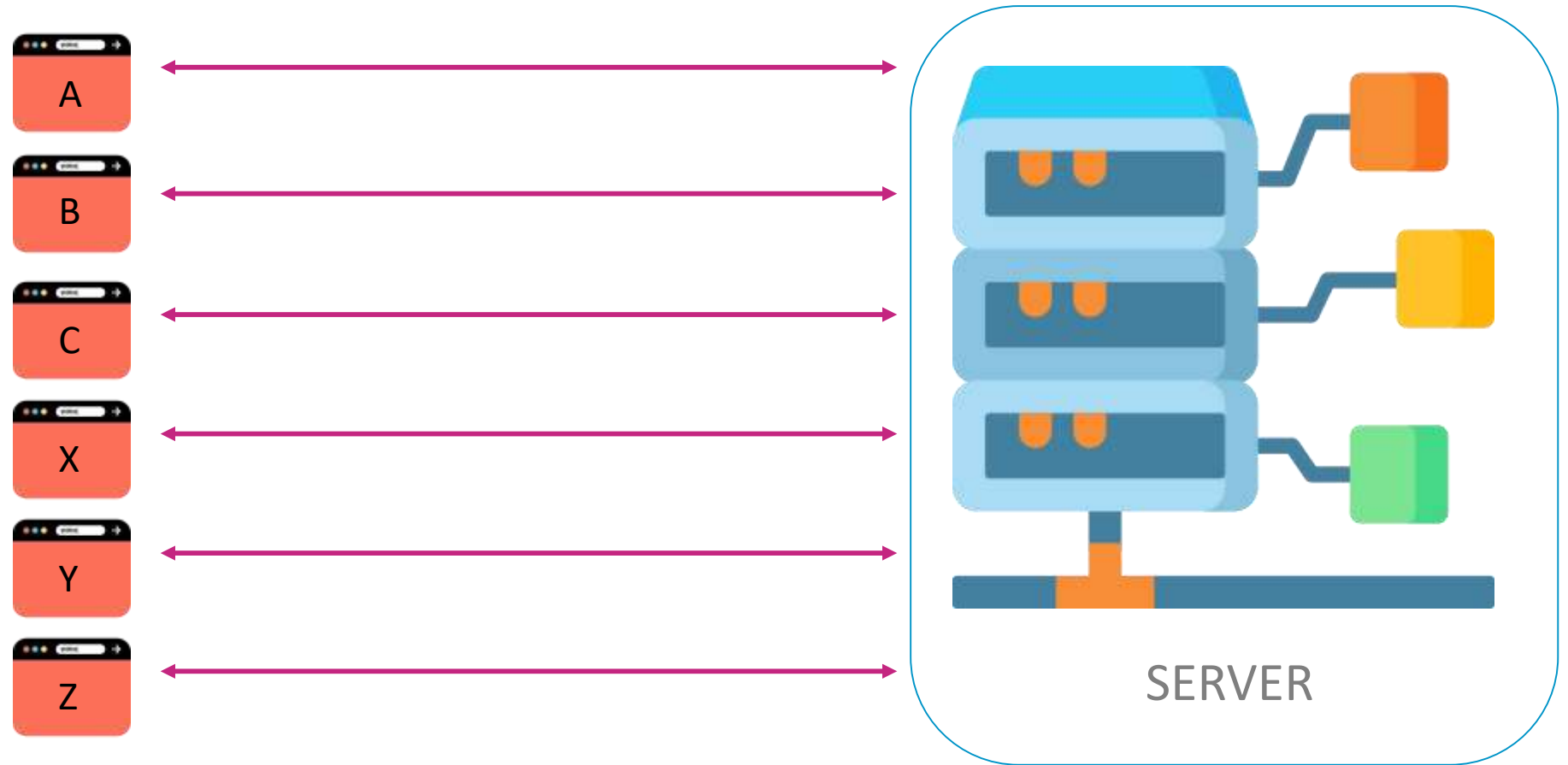
Server Sent Events



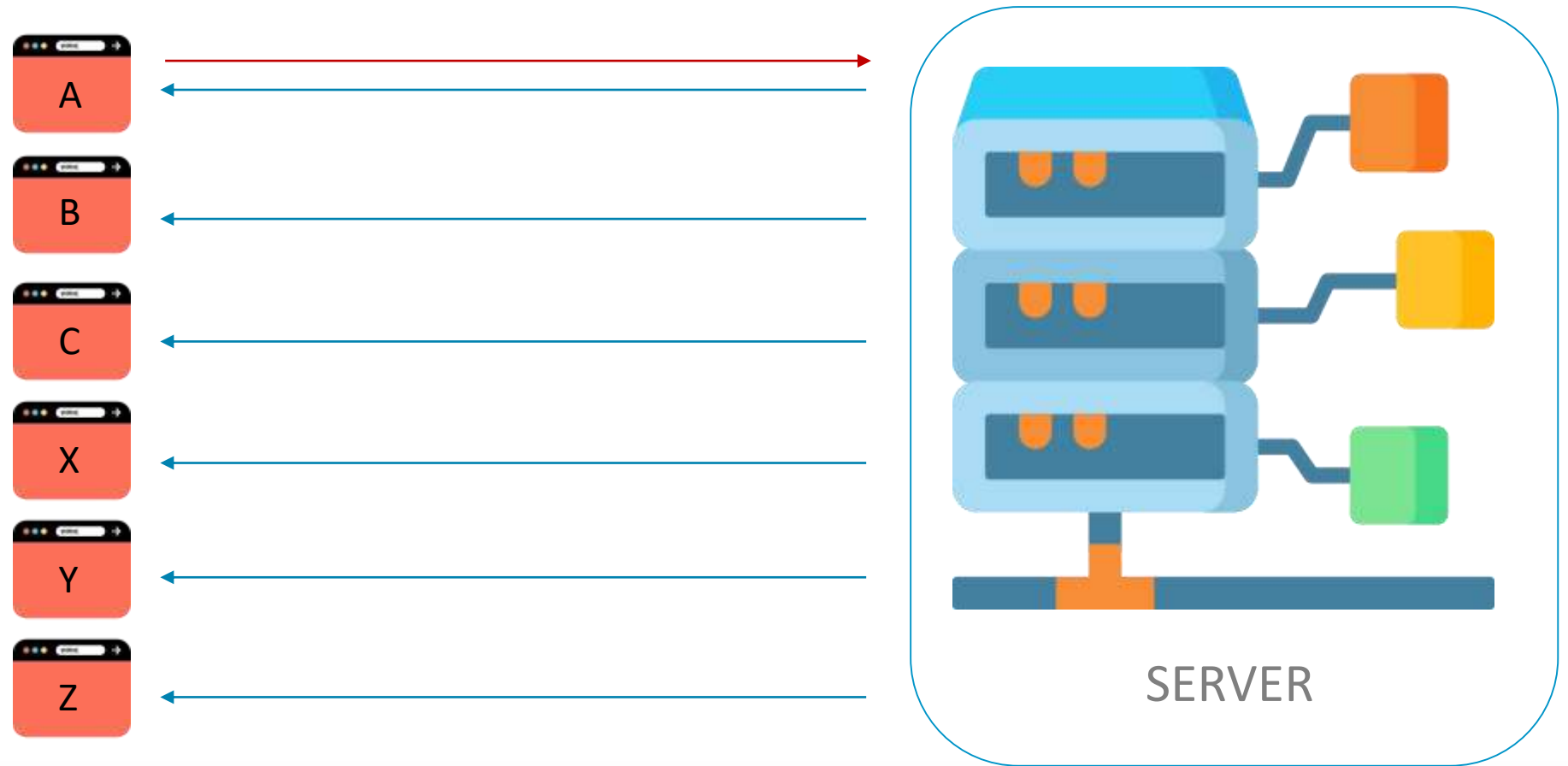
LONG POLLING



SIGNALR CONNECTION

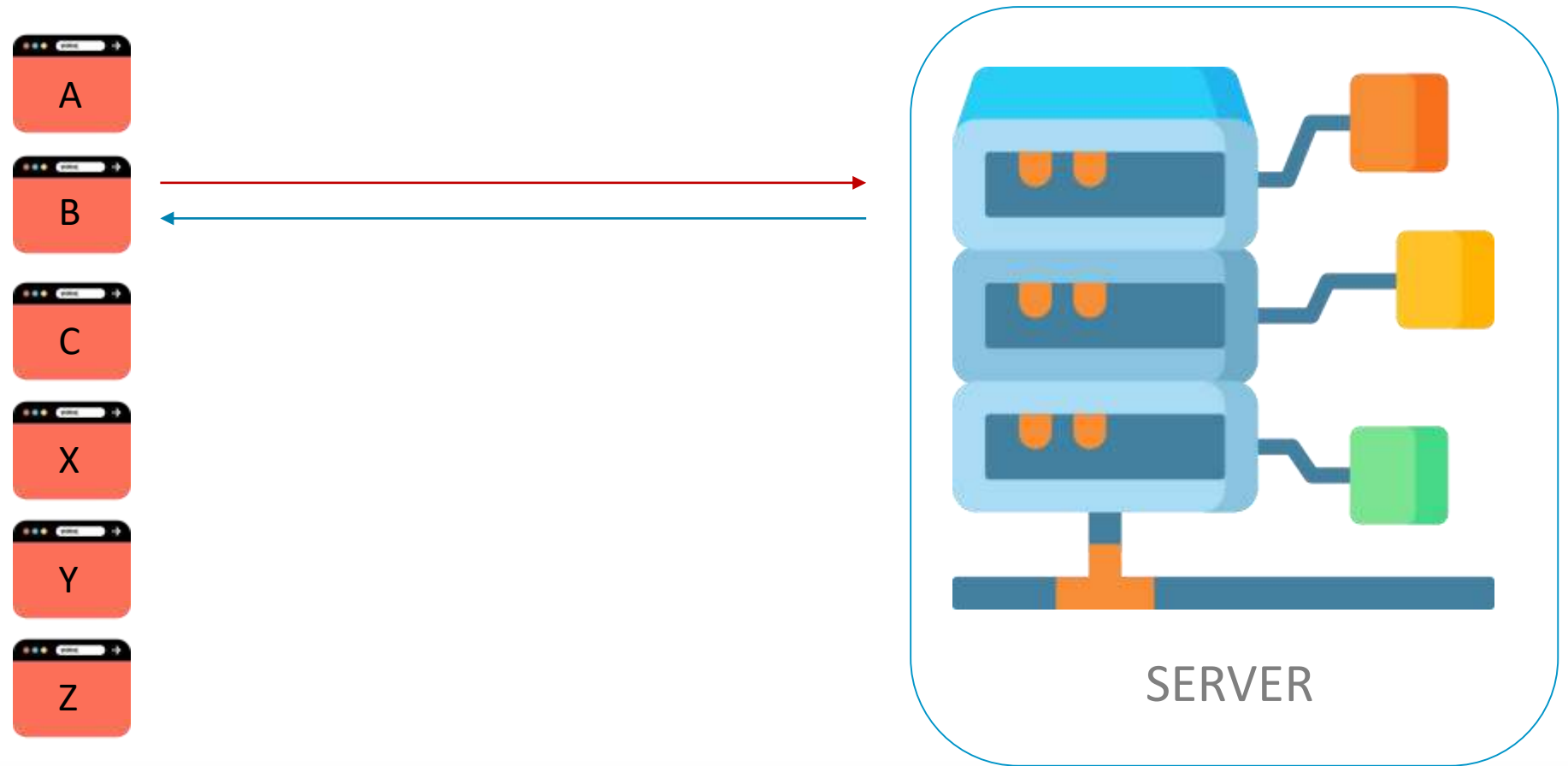


SIGNALR CONNECTION



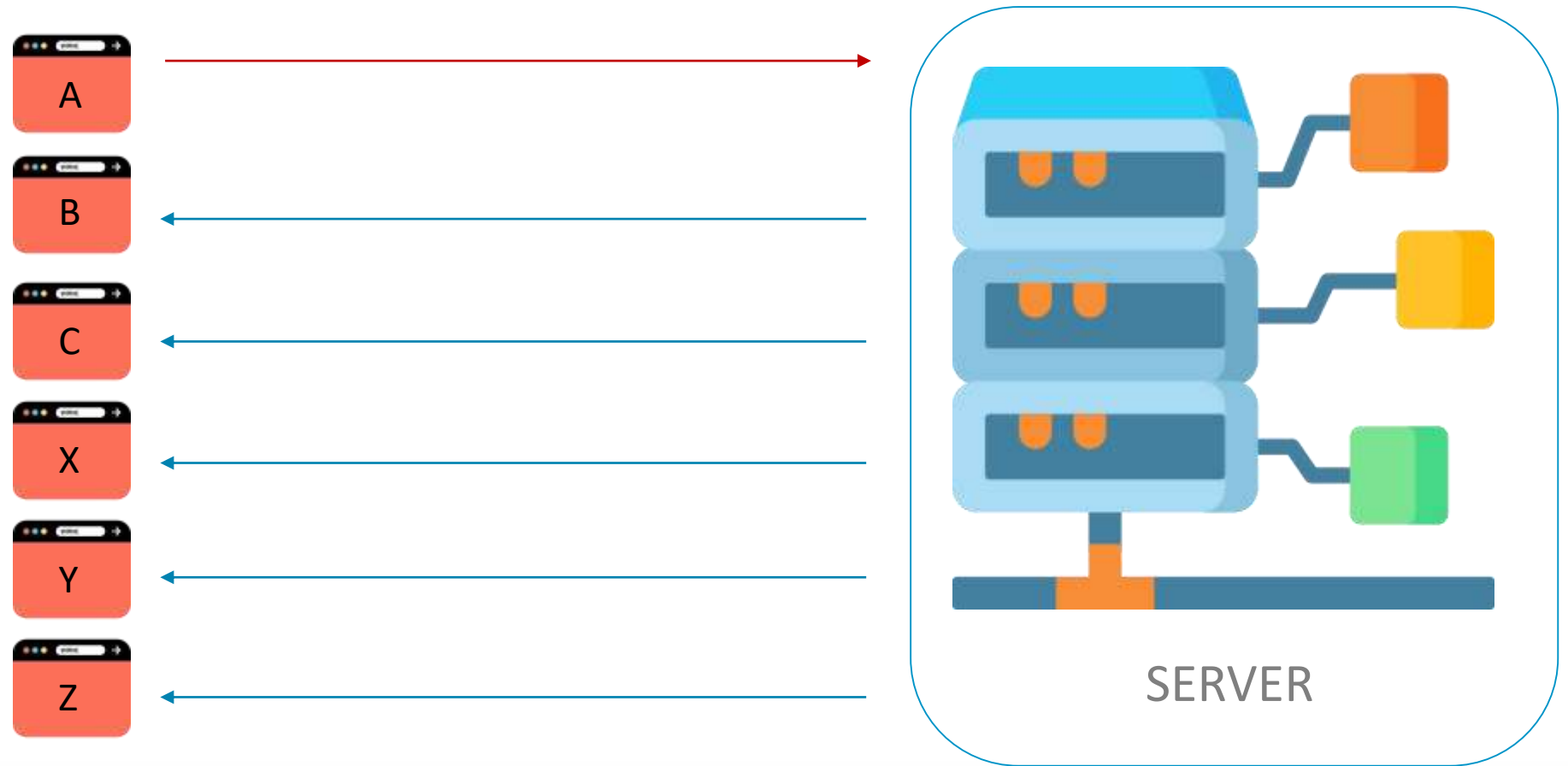
```
await Clients.All.SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



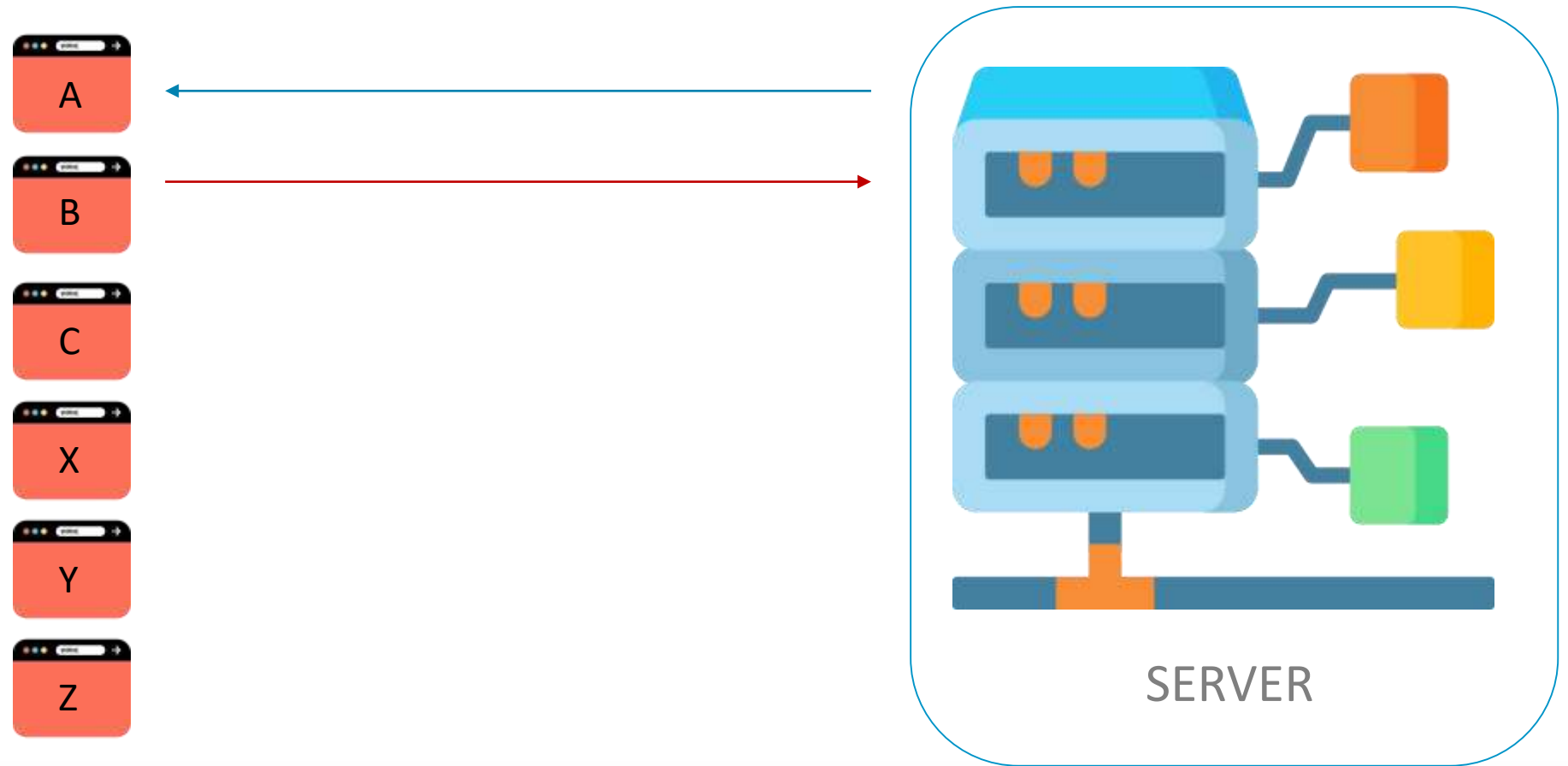
```
await Clients.Caller.SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



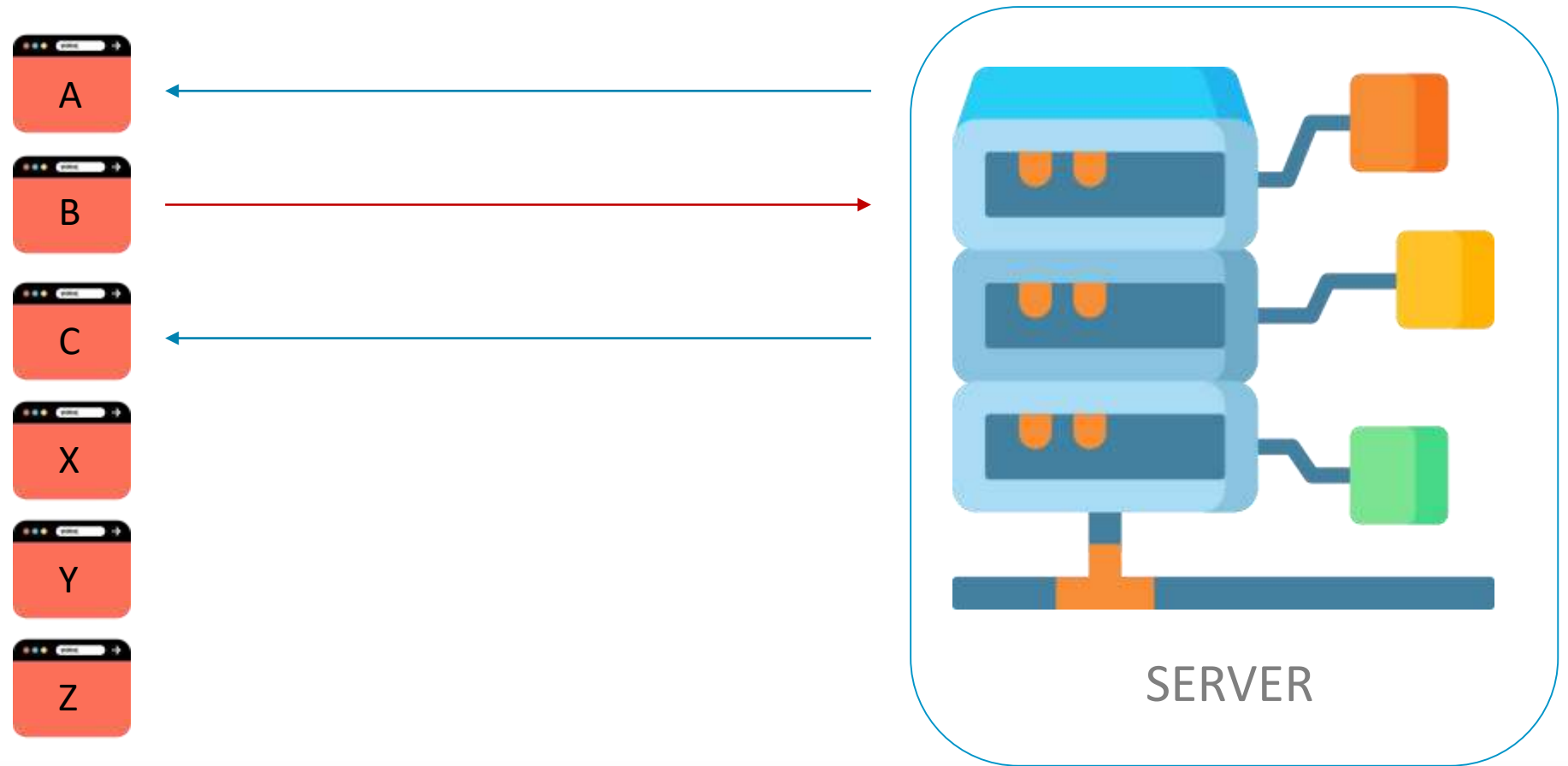
```
await Clients.Others.SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



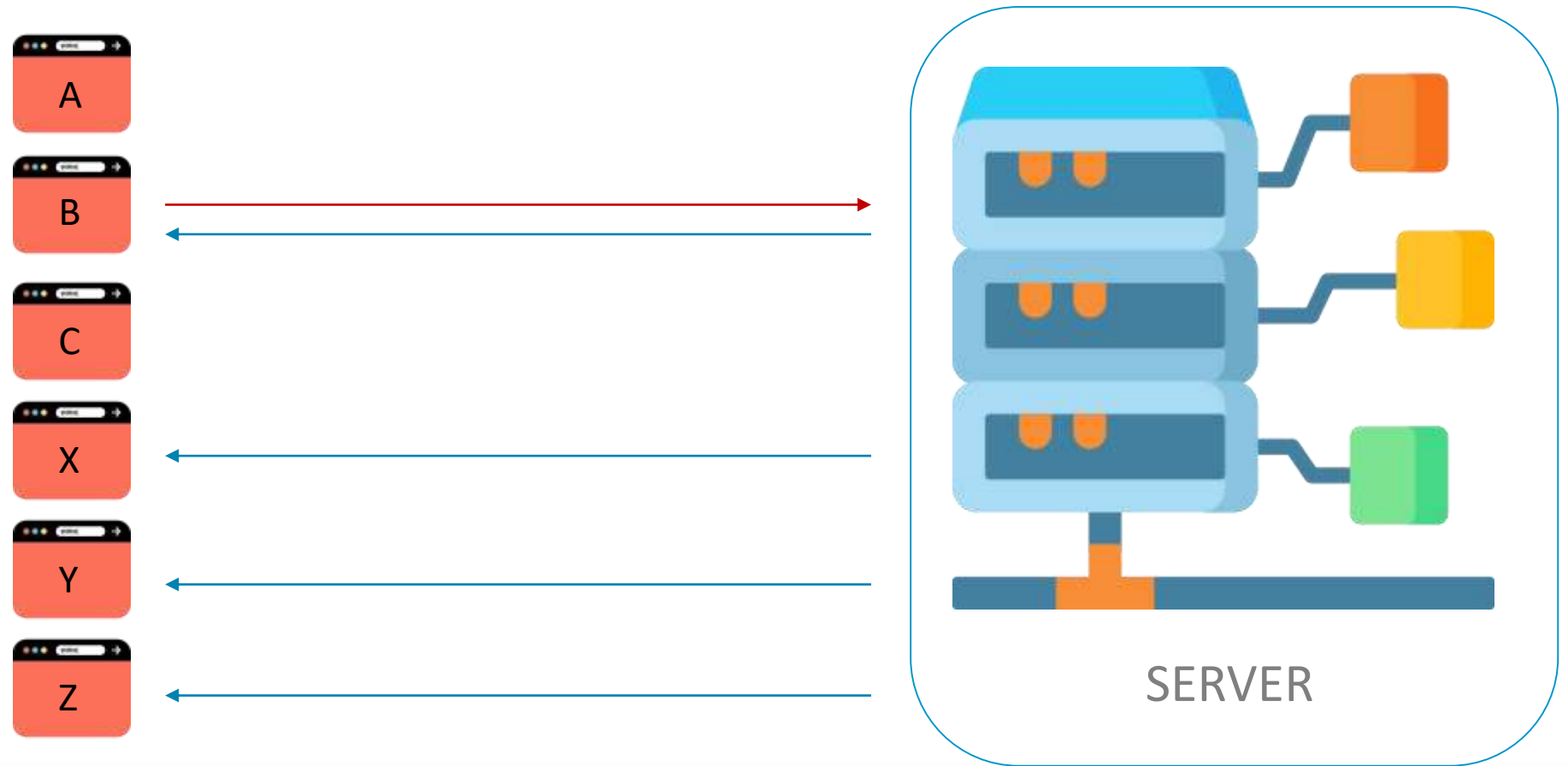
```
await Clients.Client("Connection Id - A").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



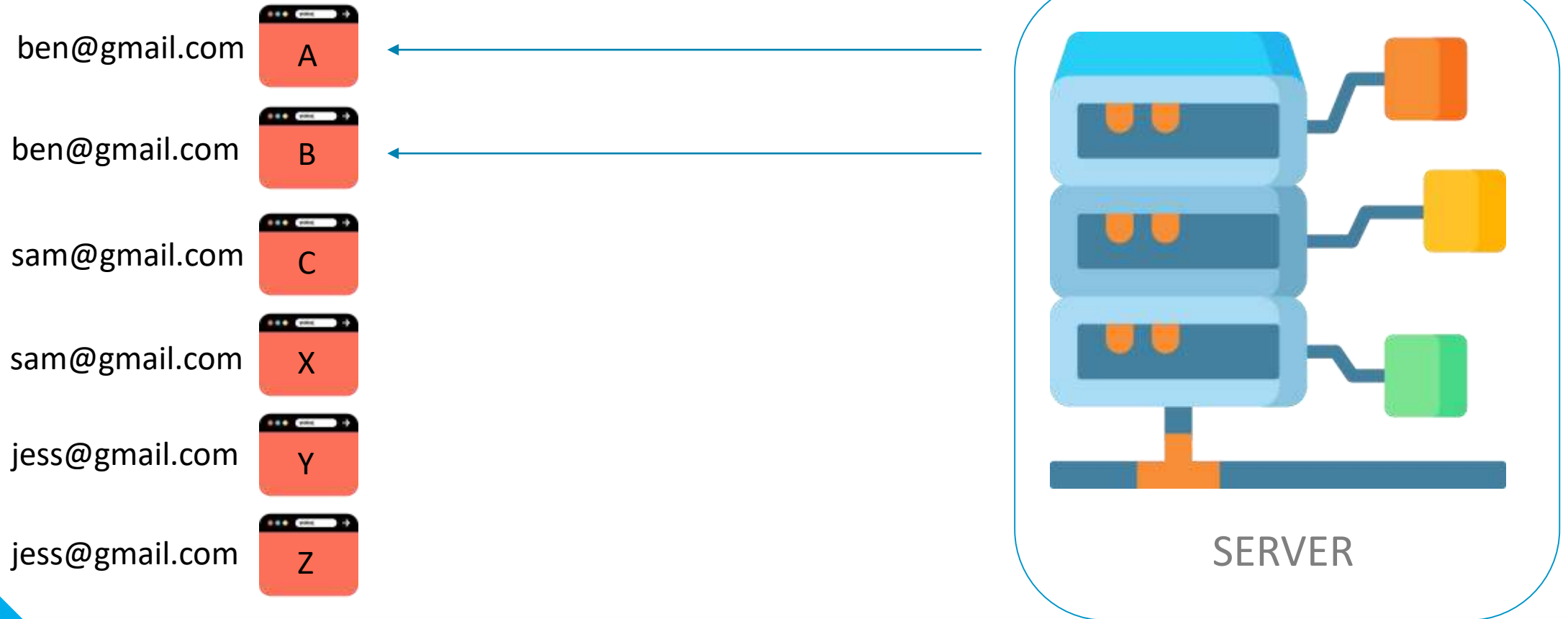
```
await Clients.Clients("Connection Id - A", "Connection Id - C").SendAsync("ReceiveMessage", user, message);
```


SIGNALR CONNECTION



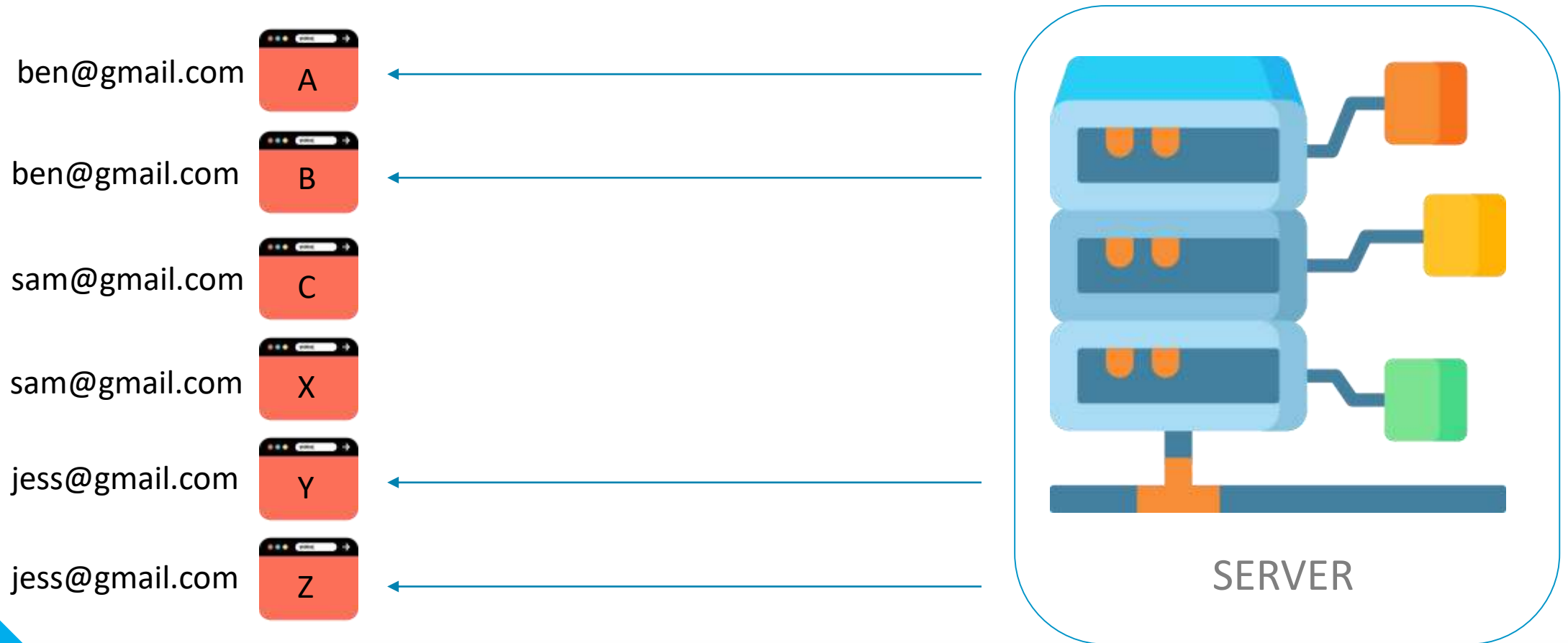
```
await Clients.AllExcept("Connection Id - A", "Connection Id - C").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



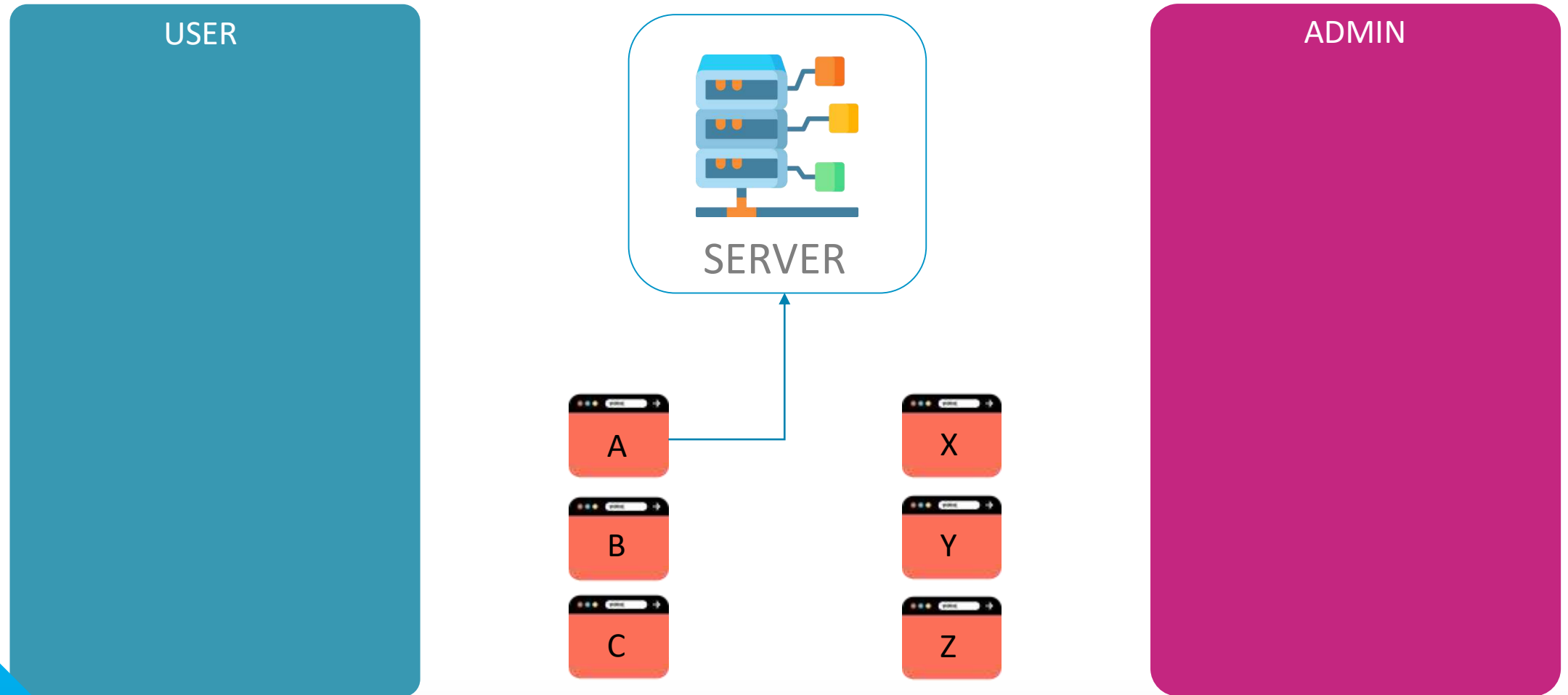
```
await Clients.User("ben@gmail.com").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



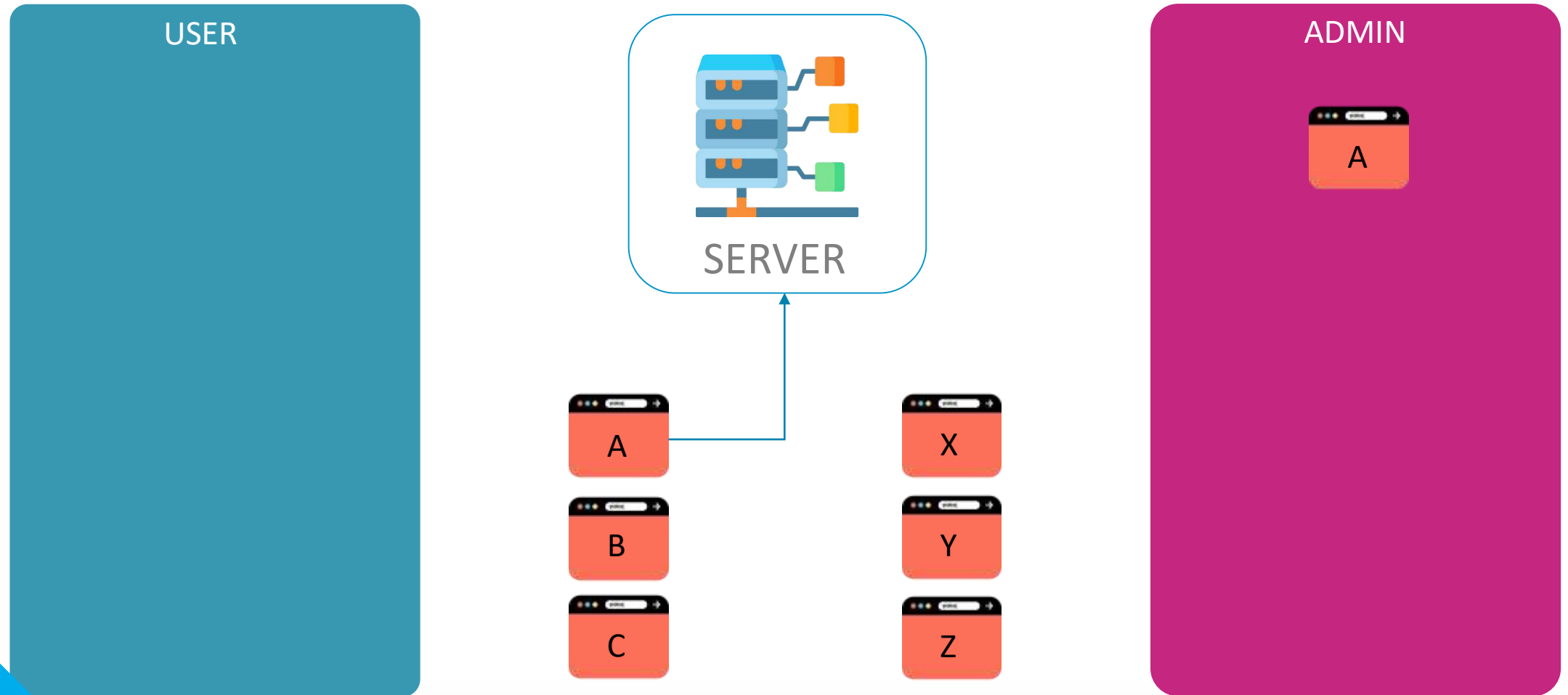
```
await Clients.Users("ben@gmail.com","jess@gmail.com").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



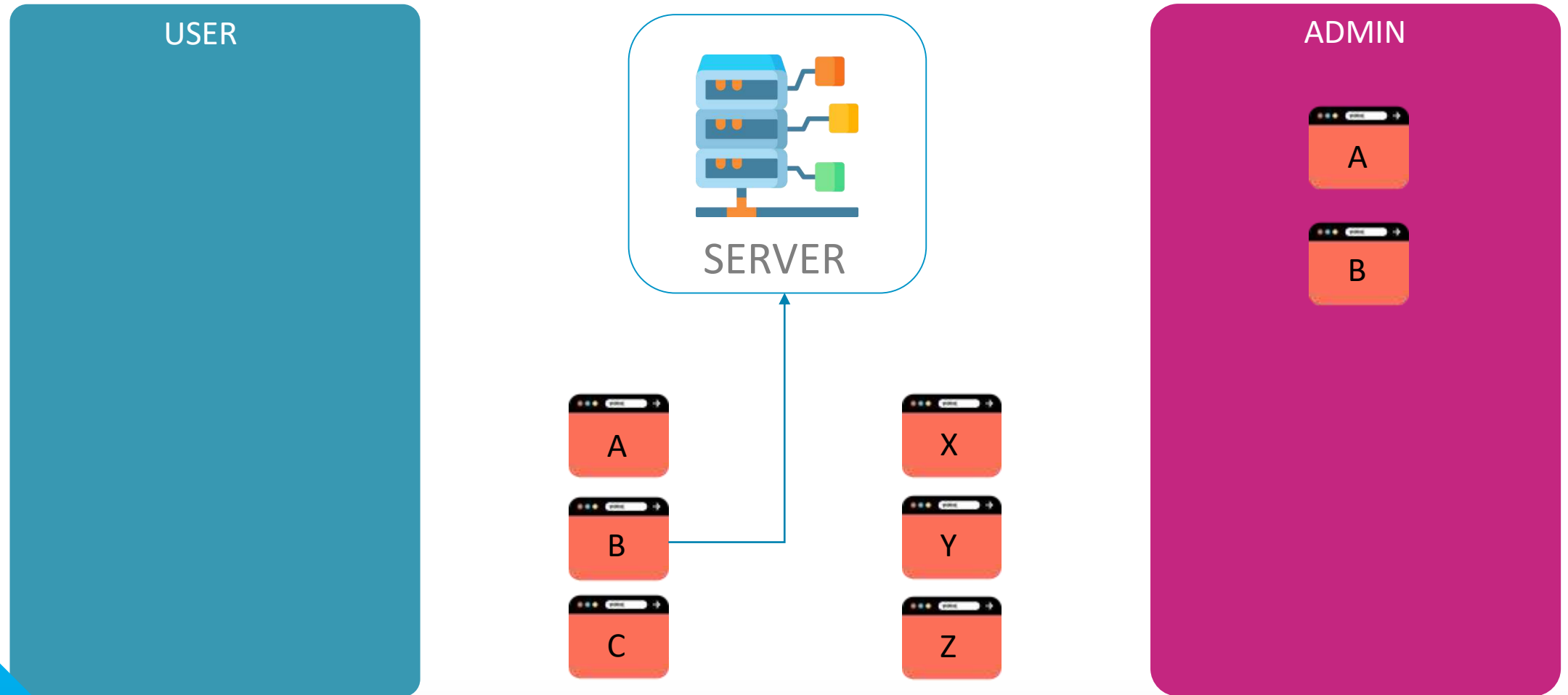
```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");
```

SIGNALR CONNECTION



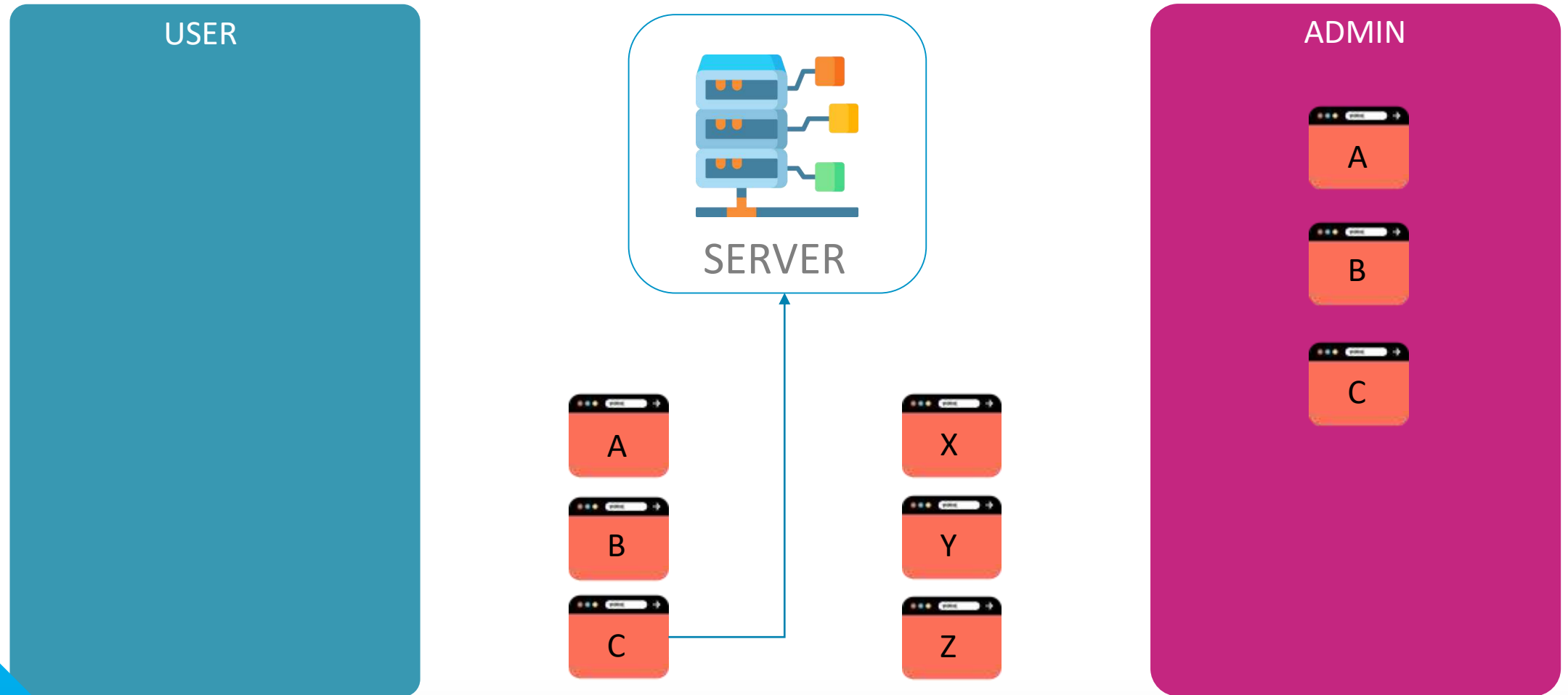
```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");
```

SIGNALR CONNECTION



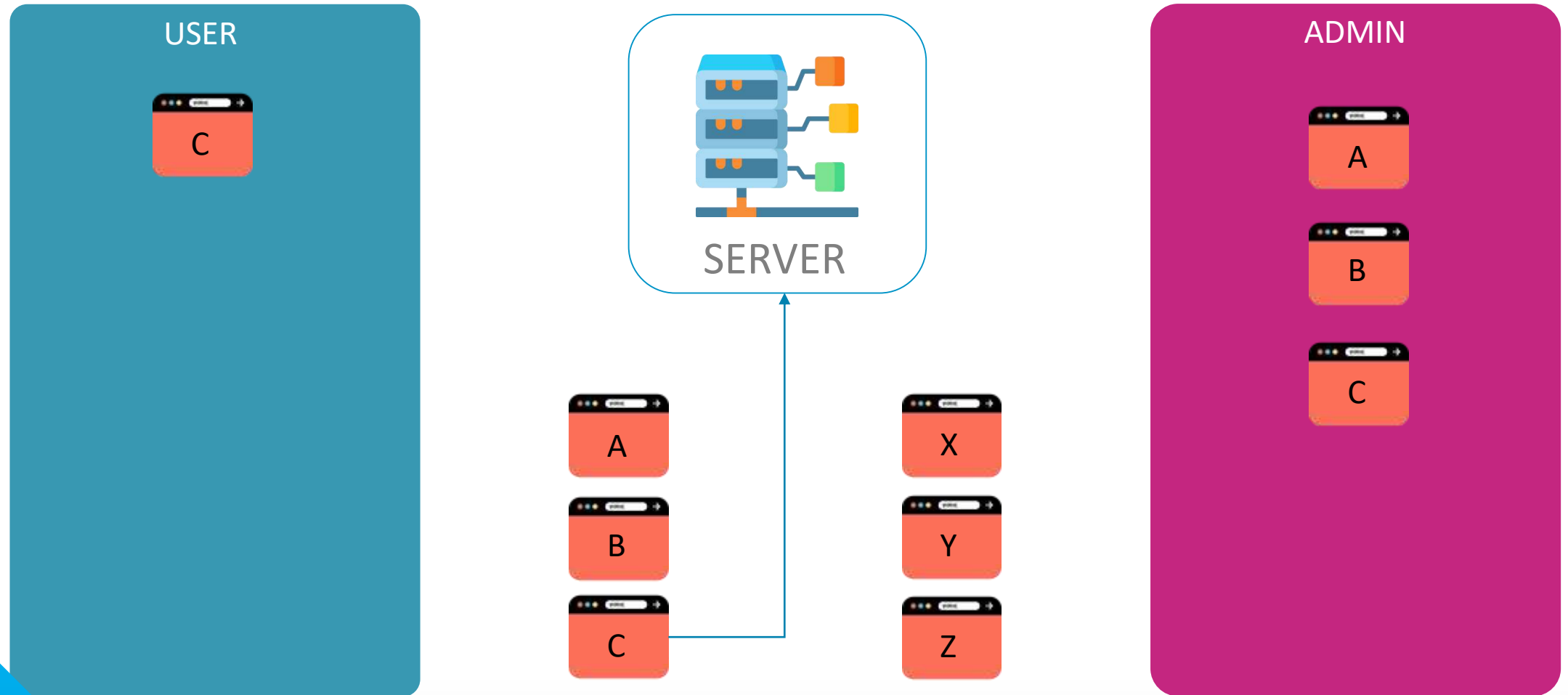
```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");
```

SIGNALR CONNECTION



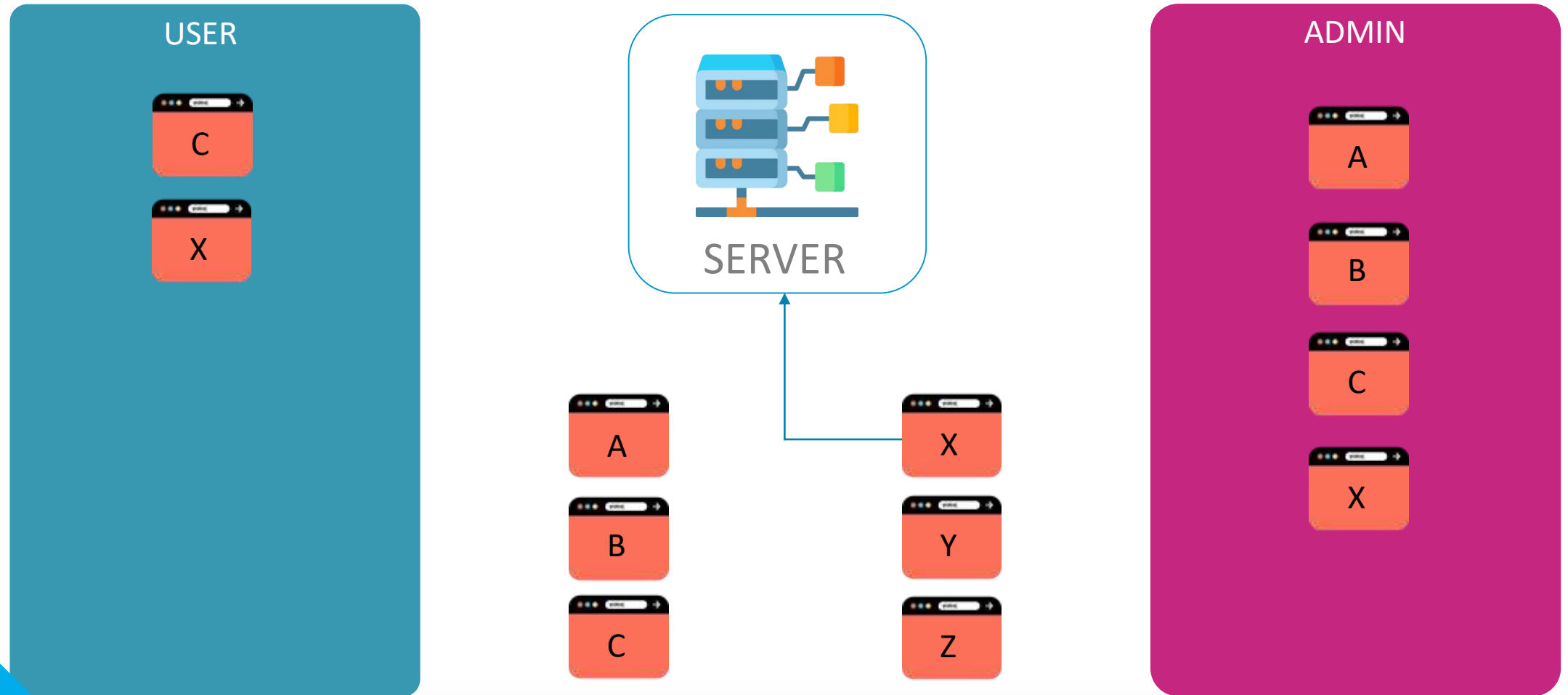
```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");
```

SIGNALR CONNECTION



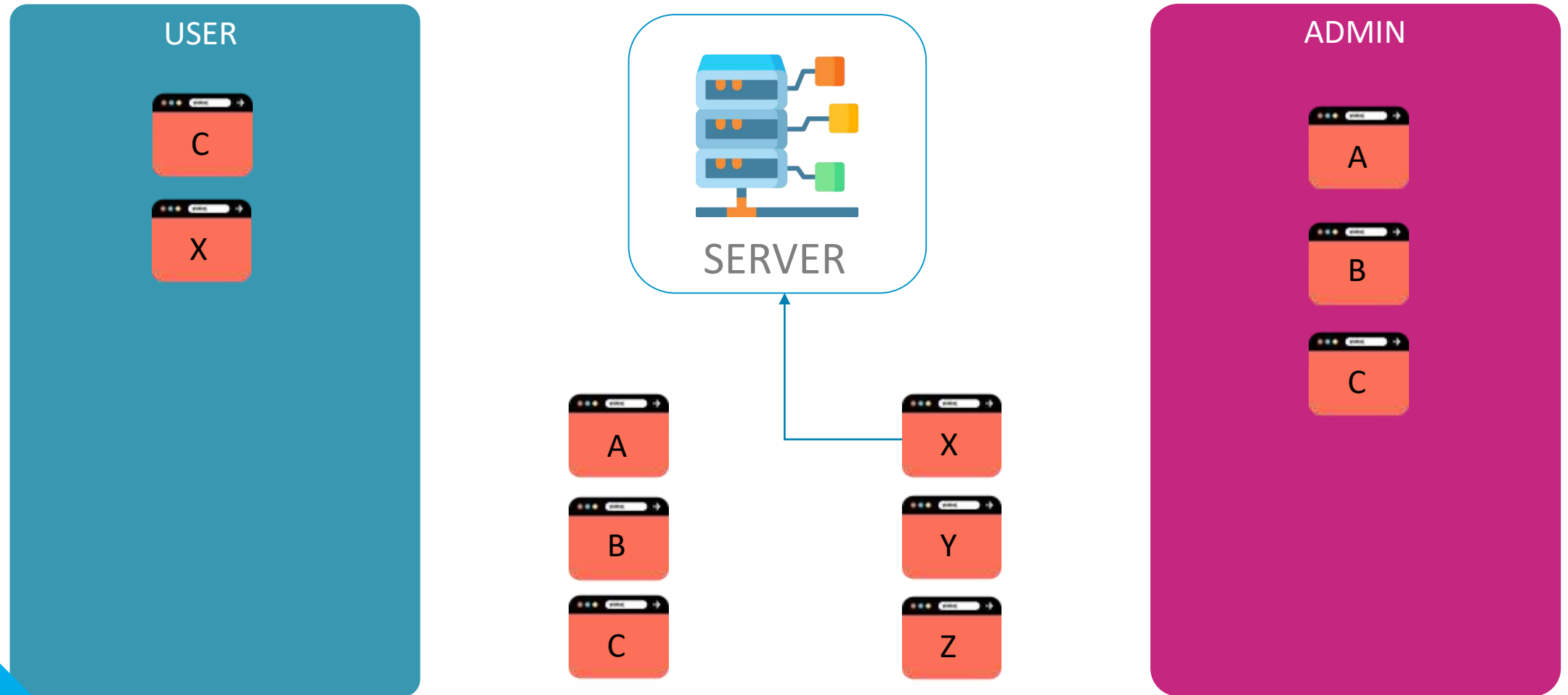
```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");  
await Groups.AddToGroupAsync(Context.ConnectionId, "user");
```


SIGNALR CONNECTION



```
await Groups.AddToGroupAsync(Context.ConnectionId, "admin");  
await Groups.AddToGroupAsync(Context.ConnectionId, "user");
```

SIGNALR CONNECTION



```
await Groups.RemoveFromGroupAsync(Context.ConnectionId, "admin");
```

SIGNALR CONNECTION

ADMIN

ben@gmail.com A

ben@gmail.com B

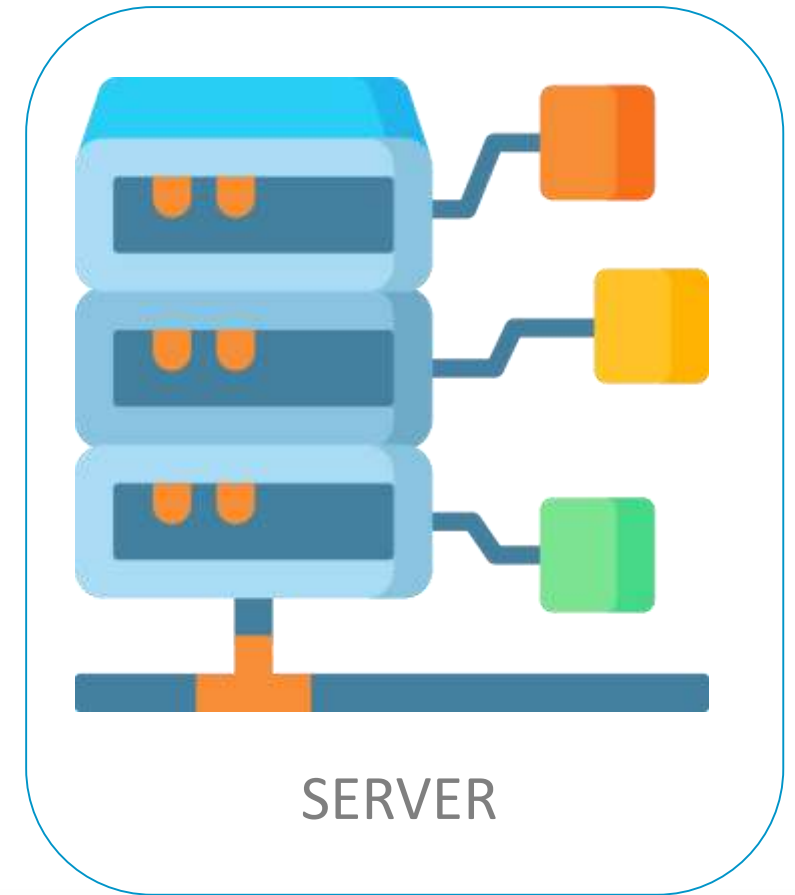
USER

sam@gmail.com C

sam@gmail.com X

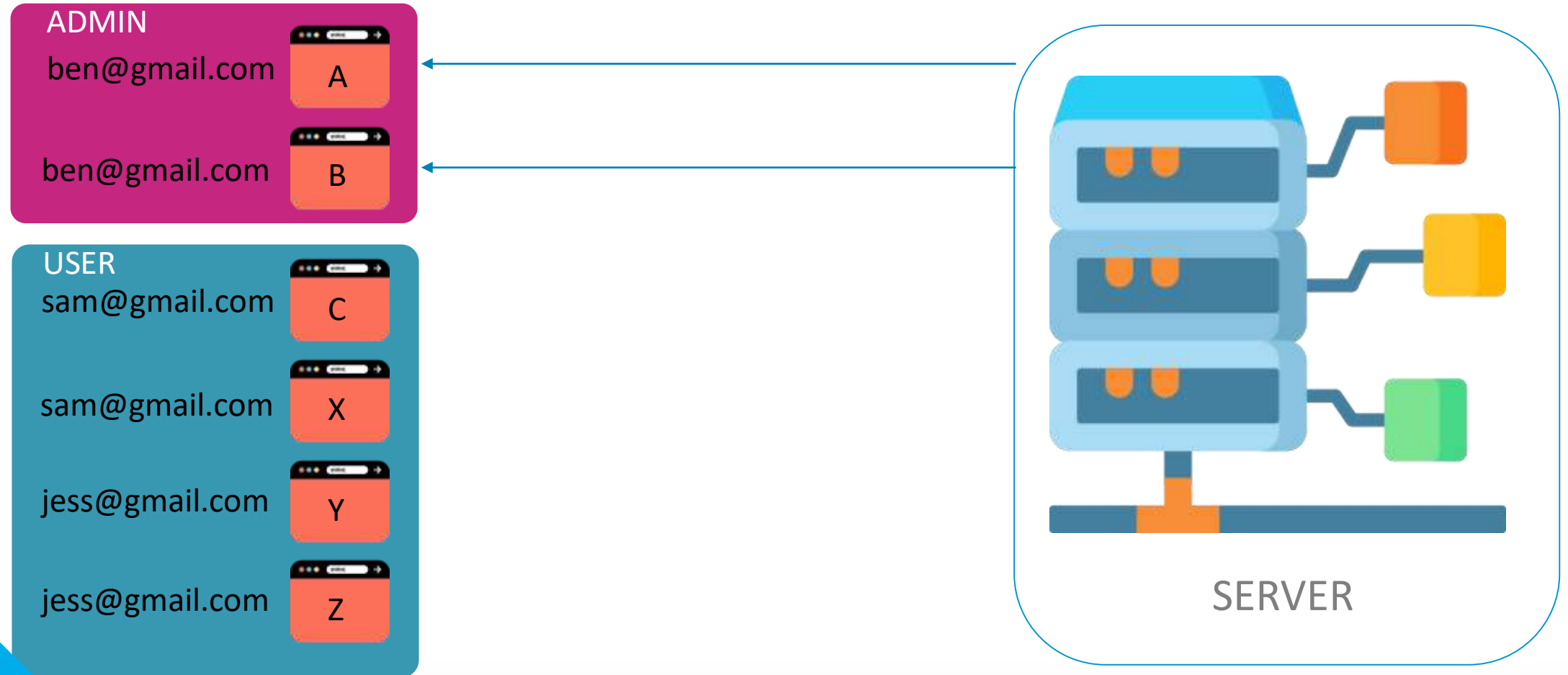
jess@gmail.com Y

jess@gmail.com Z



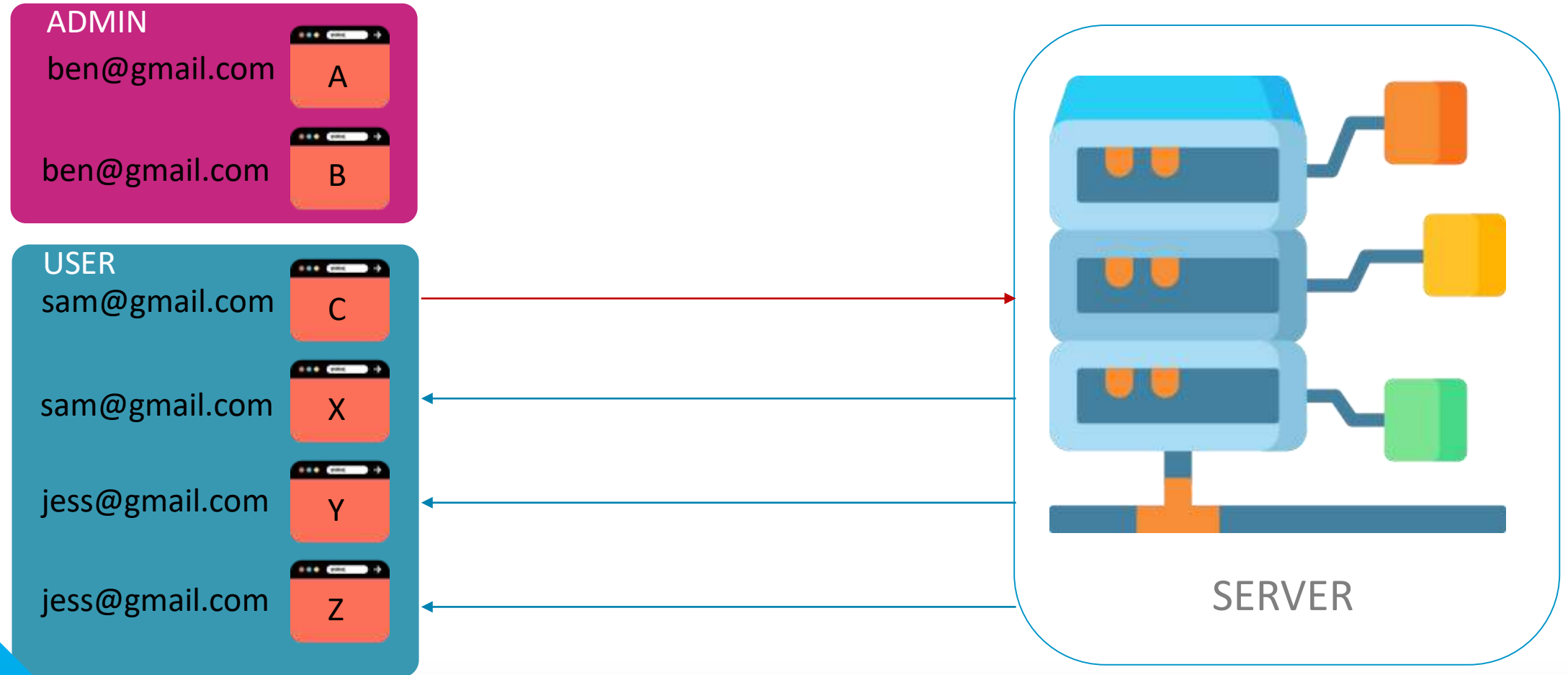
```
await Groups.AddToGroupAsync("Context.ConnectionId","admin");  
await Groups.RemoveFromGroupAsync("Context.ConnectionId", "admin");
```

SIGNALR CONNECTION



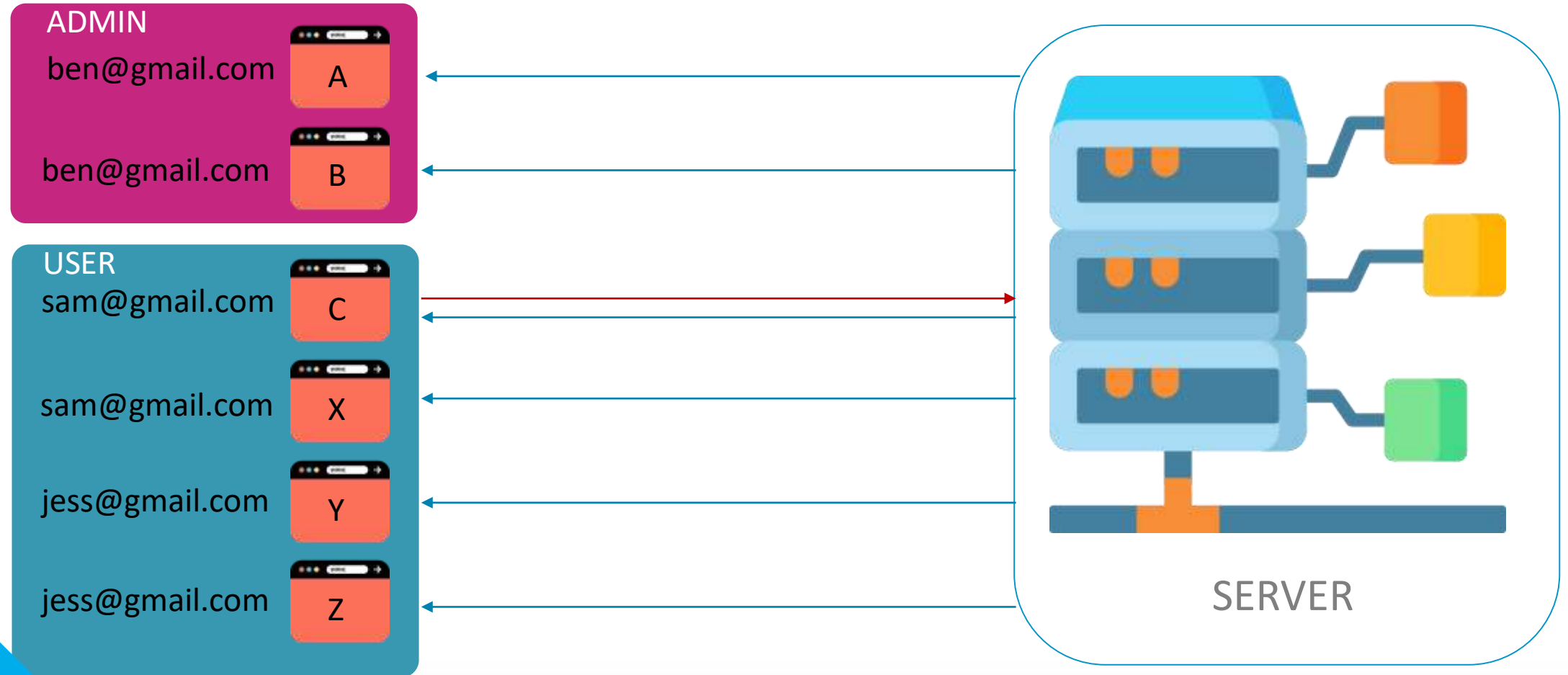
```
await Clients.Group("admin").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



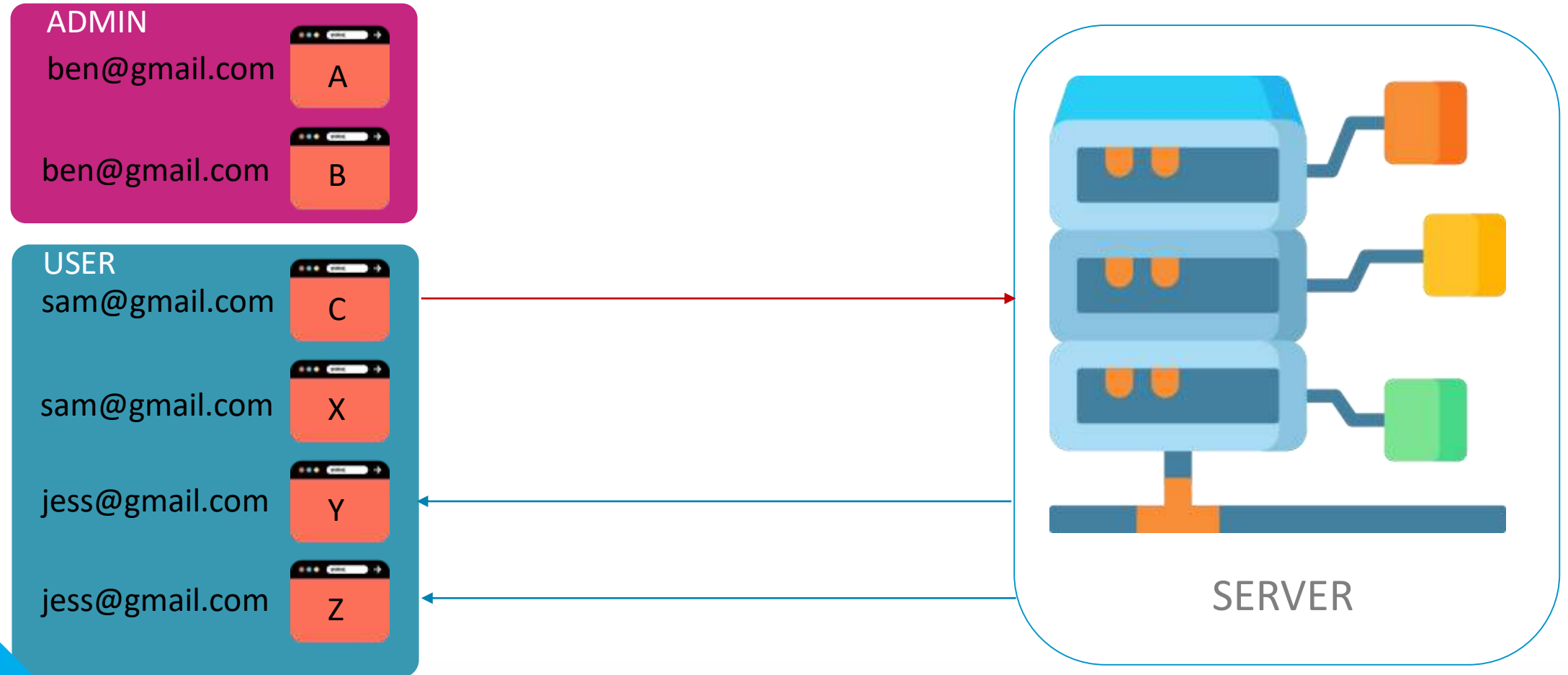
```
await Clients.OthersInGroup("user").SendAsync("ReceiveMessage", user,message);
```

SIGNALR CONNECTION



```
await Clients.Groups("admin","user").SendAsync("ReceiveMessage", user, message);
```

SIGNALR CONNECTION



```
await Clients.GroupExcept("user", "sam@gmail.com").SendAsync("ReceiveMessage", user, message);
```


SIGNALR CONNECTION

```
await Clients.All.SendAsync("ReceiveMessage", user, message);

await Clients.Caller.SendAsync("ReceiveMessage", user, message);

await Clients.Others.SendAsync("ReceiveMessage", user, message);

await Clients.Client("Connection Id - A").SendAsync("ReceiveMessage", user, message);

await Clients.Clients("Connection Id - A", "Connection Id - B").SendAsync("ReceiveMessage", user, message);

await Clients.AllExcept("Connection Id - A", "Connection Id - C").SendAsync("ReceiveMessage", user, message);

await Clients.User("ben@gmail.com").SendAsync("ReceiveMessage", user, message);

await Clients.Users("ben@gmail.com", "jess@gmail.com").SendAsync("ReceiveMessage", user, message);

await Groups.AddToGroupAsync(Context.ConnectionId, "GroupName");

await Groups.RemoveFromGroupAsync(Context.ConnectionId, "GroupName");

await Clients.Group("admin").SendAsync("ReceiveMessage", user, message);

await Clients.OthersInGroup("admin").SendAsync("ReceiveMessage", user, message);

await Clients.Groups("admin", "user").SendAsync("ReceiveMessage", user, message);

await Clients.GroupExcept("admin", "sam@gmail.com").SendAsync("ReceiveMessage", user, message);
```