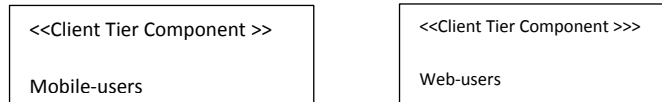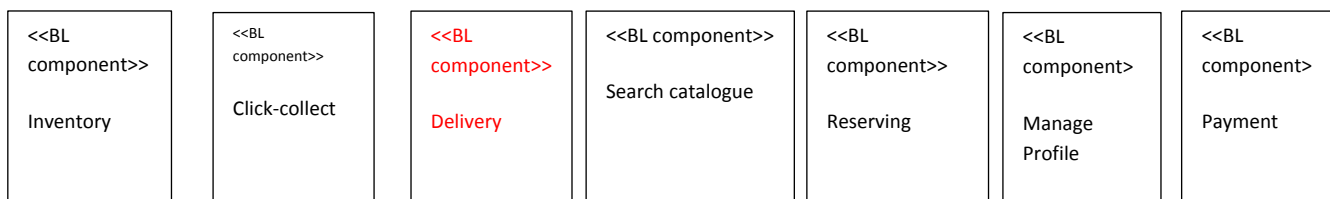**Argos example – A brainstorming exercise as part of the three-tier section of the recording:**

Three- tier architecture, compromising of client-tier, business-logic tier and database tier:
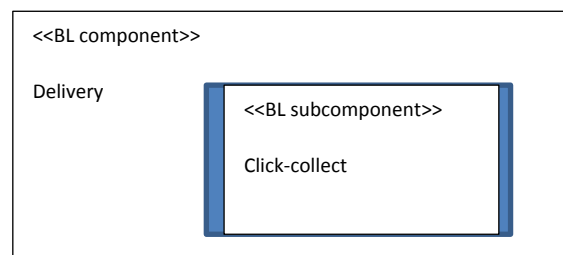
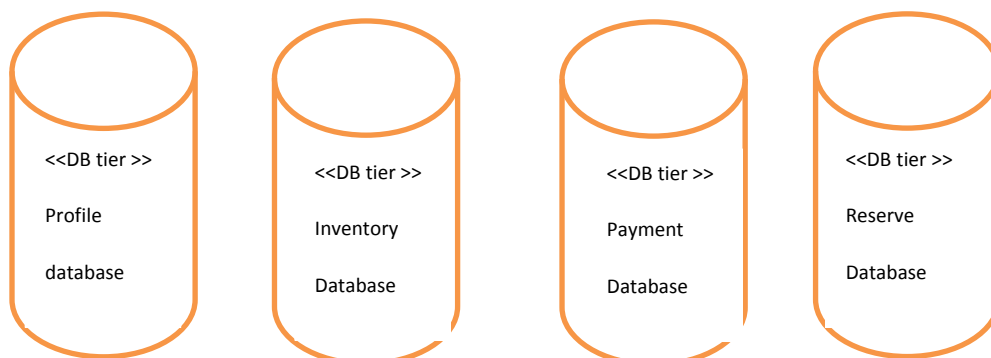First, brainstorm of possible client-tier component among the others:

<<Client Tier Component >>	<<Client Tier Component >>>
Mobile-users	Web-users

Second, brainstorm of possible Business Logic (BL) tier components among the others:

<<BL component>>	<<BL component>>	<<BL component>>	<<BL component>>	<<BL component>>	<<BL component>	<<BL component>
Inventory	Click-collect	Delivery	Search catalogue	Reserving	Manage Profile	Payment

Alternative model for << Business Logic component>> Delivery

<<BL component>>

Delivery

<<BL subcomponent>>

Click-collect

Third: Database Tier – one alternative design for Argos DB tier
(brainstorm of possible database components.

<<DB tier >>	<<DB tier >>	<<DB tier >>	<<DB tier >>
Profile database	Inventory Database	Payment Database	Reserve Database

Alternative design: Argos may decide to have one giant Database instead of multiple ones: this can be risky as it has one single point of failure and can suffer from performance bottlenecks, slowing down the system… can be less scalable if the system will grow…

You may also merge Payment and Reserve databases… less expensive but security, scalability and performance can be among the issues… Always think of the trade-offs: cost, security, scalability, performance, etc.