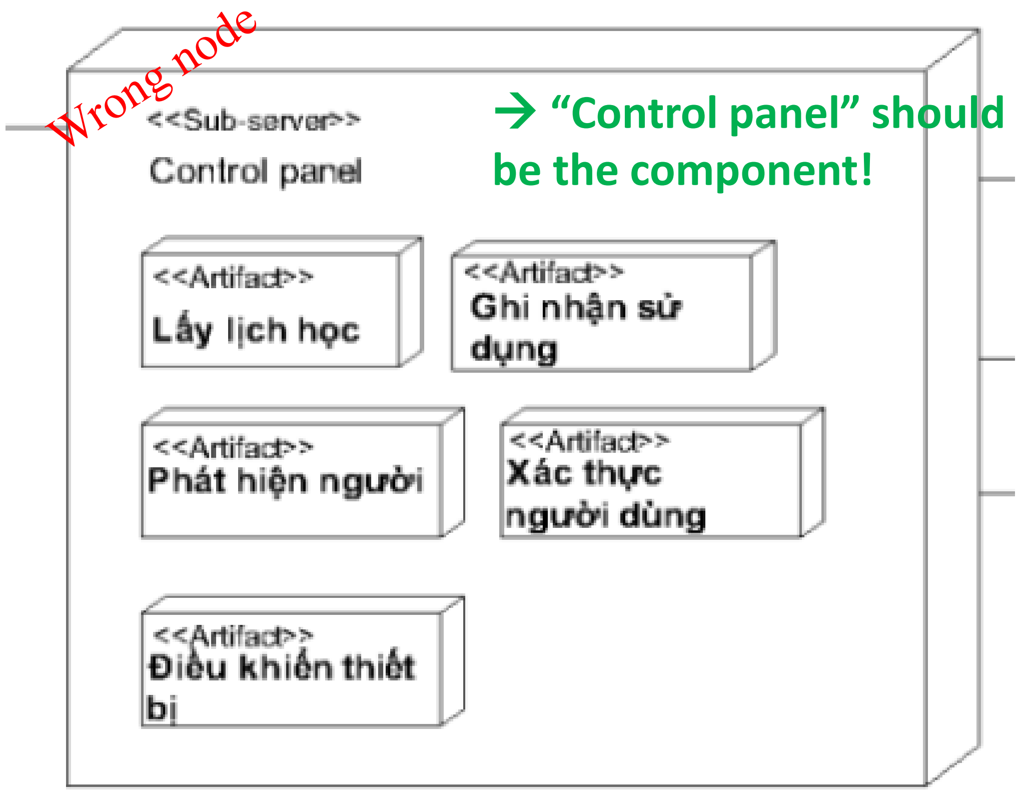
Review Project #3

11/2019

Tips

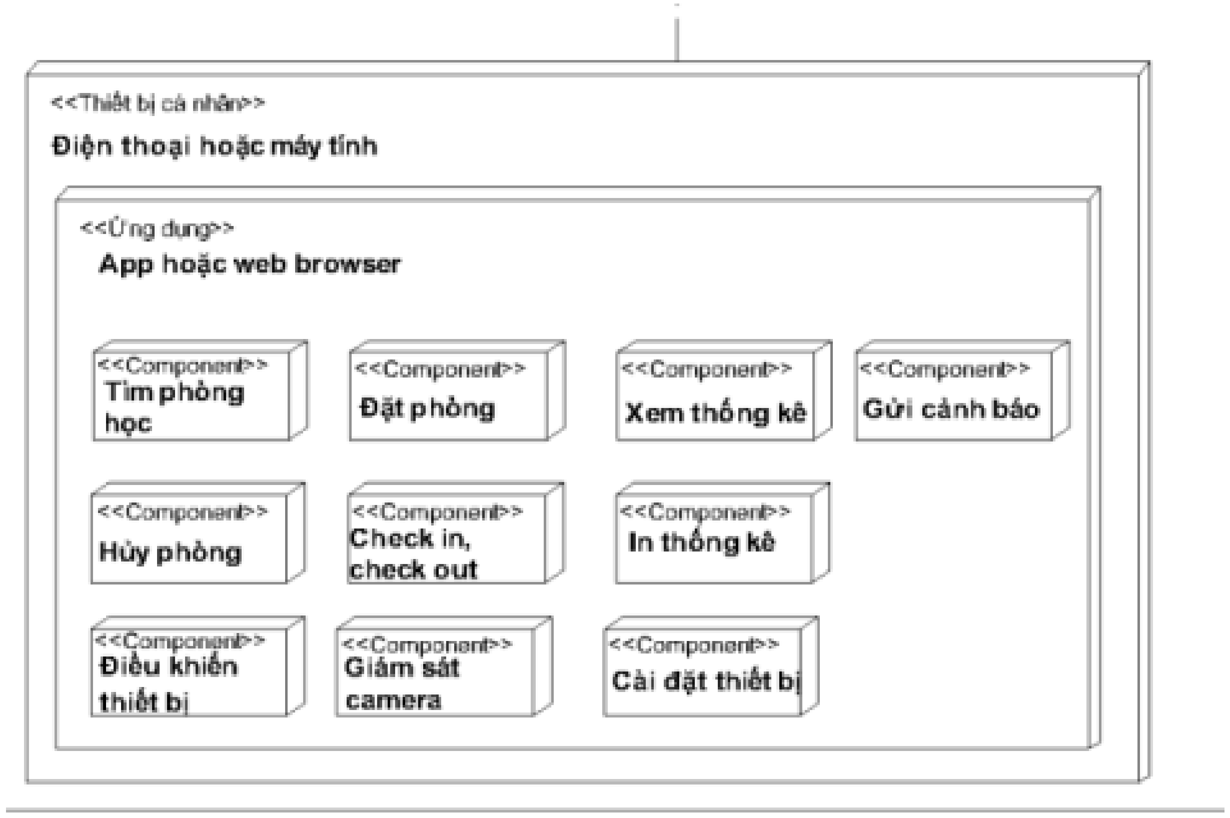
- Deployment view
  - Cần rõ các <<subsystem>>/component và vị trí vật lý (cài đặt)
  - Hoặc chỉ rõ thành phần nào sẽ deploy lên server, client, …
- Component diagram / Package diagram
  - Nên dùng các architectural design pattern
    - Layered, Repository, Client-Server, ...
  - Các thành phần bên trong component/package
    - Các component/package
    - Không là function/method

Use wrong symbol



→ "Control panel" should be the component!

Student's diagram



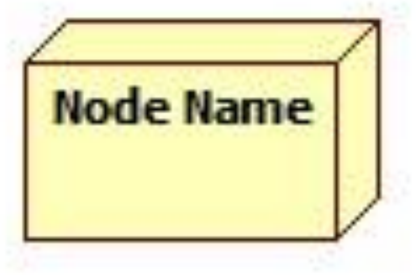
Wrong component symbol

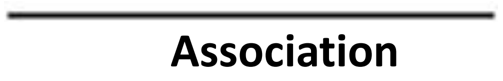
- Wrong architecture
- Wrong artifact symbol and meaning

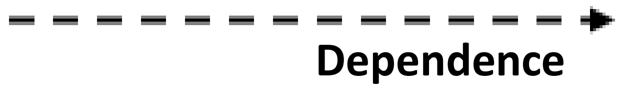
See next 3 slides

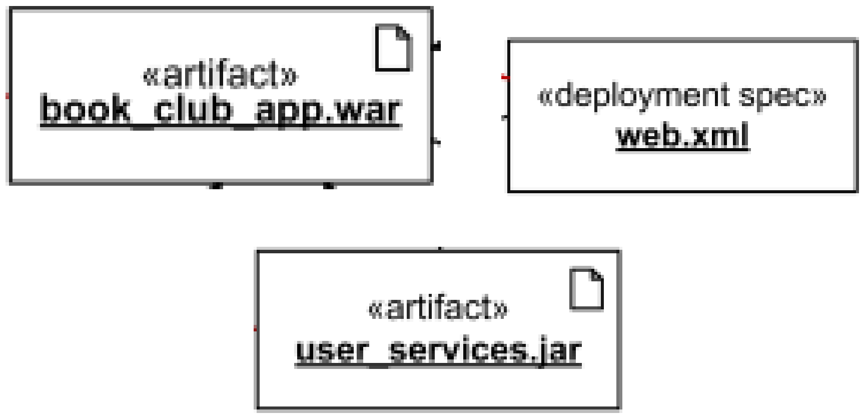
Deployment view

- **Artifacts** represent concrete elements in the physical world that are the result of a development process.
    - Examples of artifacts are executable files, libraries, archives, database schemas, configuration files, etc.
- **Deployment target** is usually represented by a **node** which is either hardware device or some software execution environment.
    - Nodes could be connected through **communication paths** to create networked systems of arbitrary complexity.









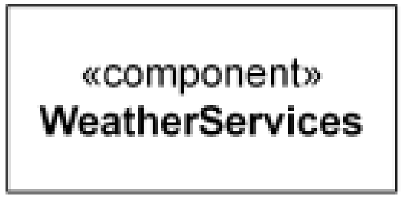
https://www.uml-diagrams.org/component.html

Deployment view

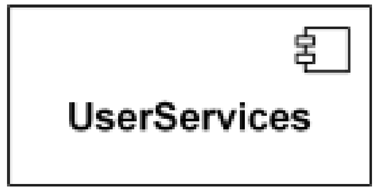
- A *component* is a **class** representing a modular part of a system with encapsulated content and whose **manifestation** is replaceable within its environment.

- A component has its behavior defined in terms of **provided interfaces** and **required interfaces** (potentially exposed via **ports**).

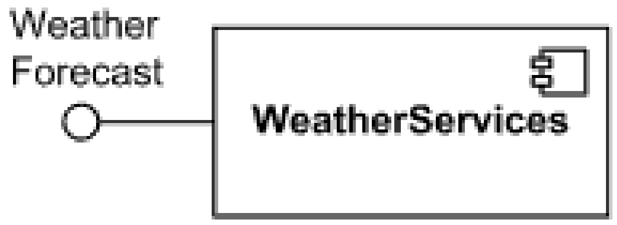
in UML 1.4



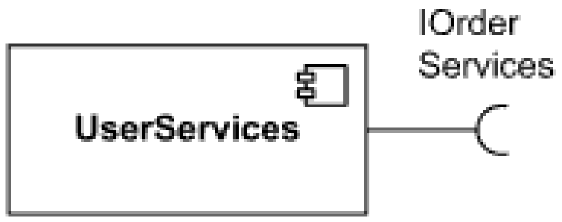
WeatherServices component



UserServices component



provided interface



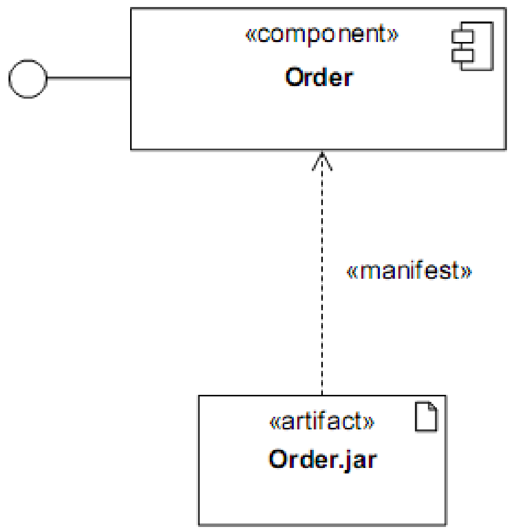
required interface

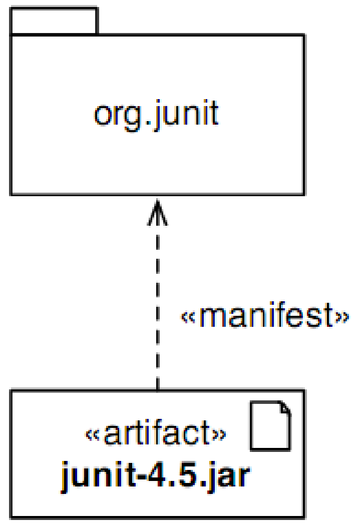
https://www.uml-diagrams.org/deployment-diagrams-reference.html

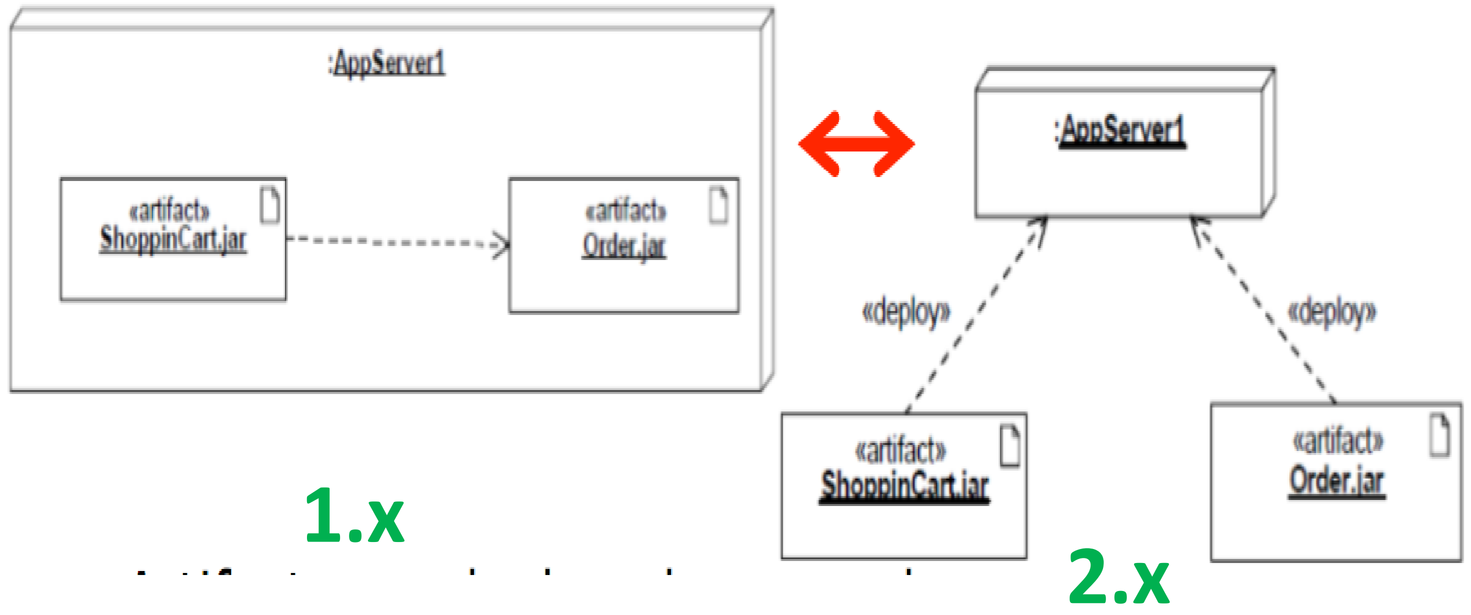
Note, that components were directly deployed to nodes.

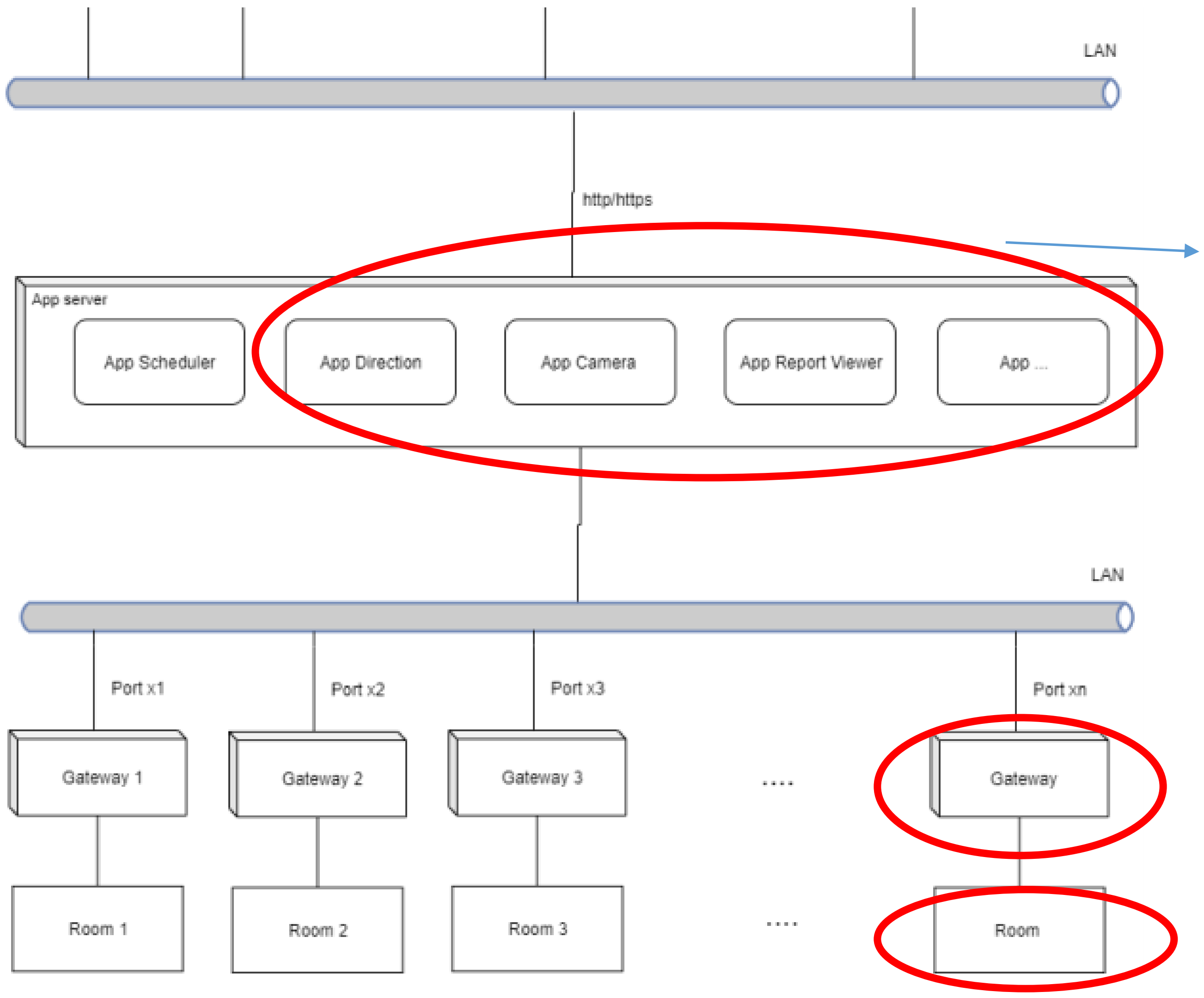
+ In UML 1.x deployment diagrams.

+ In UML 2.x artifacts are deployed to nodes, and artifacts couldmanifest (implement) components. Components are deployed to nodes indirectly through artifacts.









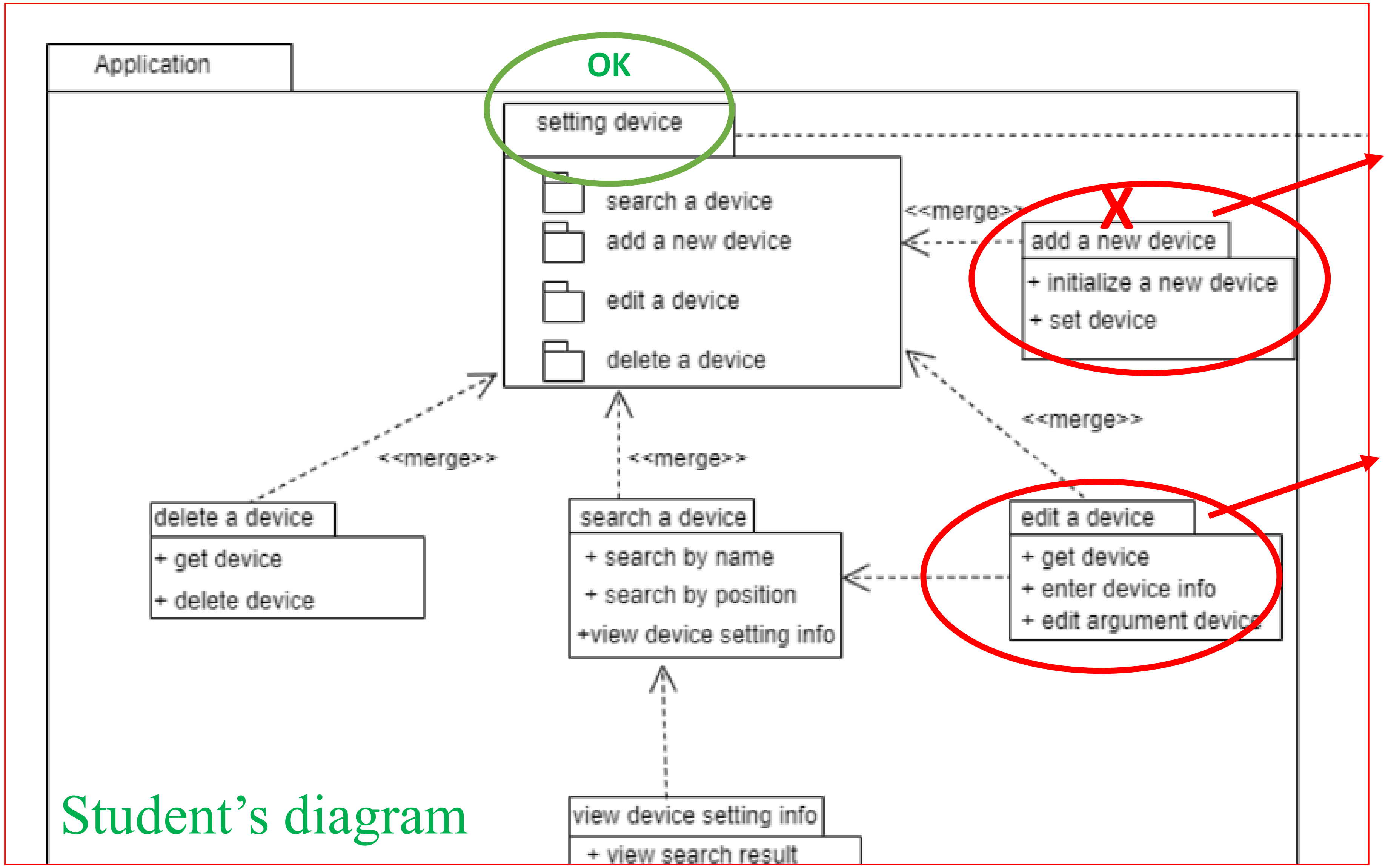
in one "App server" (physical server) ...need to install so many apps: "app Direction", "app Camera",.. (software) ??

access the Room via the Gateway?
+ gateway is a node?
+ room is a database schemas?

Why need too many gateways, ports and rooms?

Student's diagram

Package diagram



Not a package
→ Interface or Abstract class,..
+ have protocol specifications

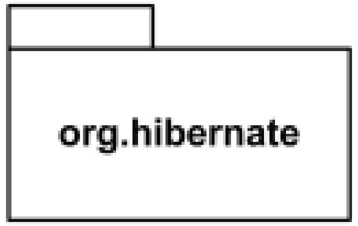
→ Interface detailed desc at project submission #4

Tips:
Các thành phần bên trong component/package
+ Các component/package
+ Không là function/method

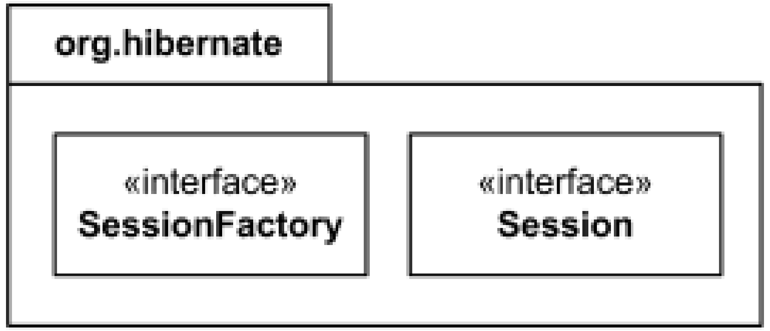
See next 1 slide

Package

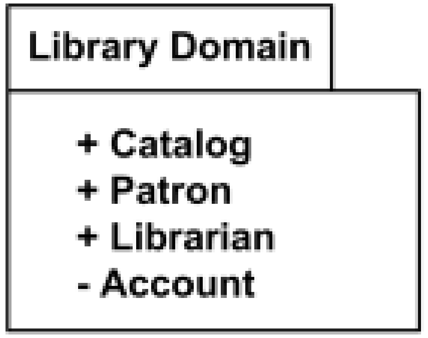
- **Package** is a **namespace** used to group together elements that are semantically related and might change together.
- It is a general purpose mechanism to organize elements into groups to provide better structure for system model.



Package org.hibernate

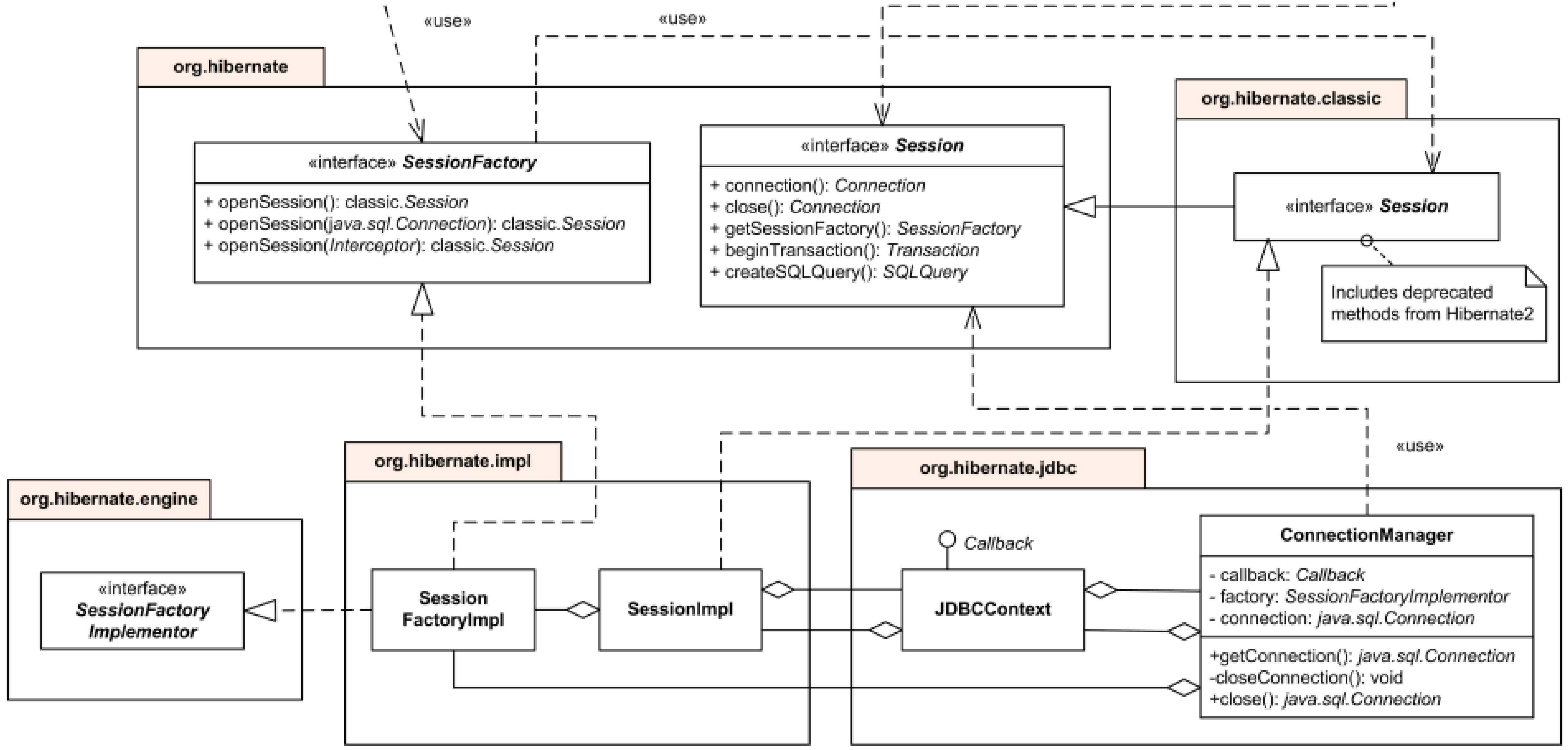


Package org.hibernate contains SessionFactory and Session



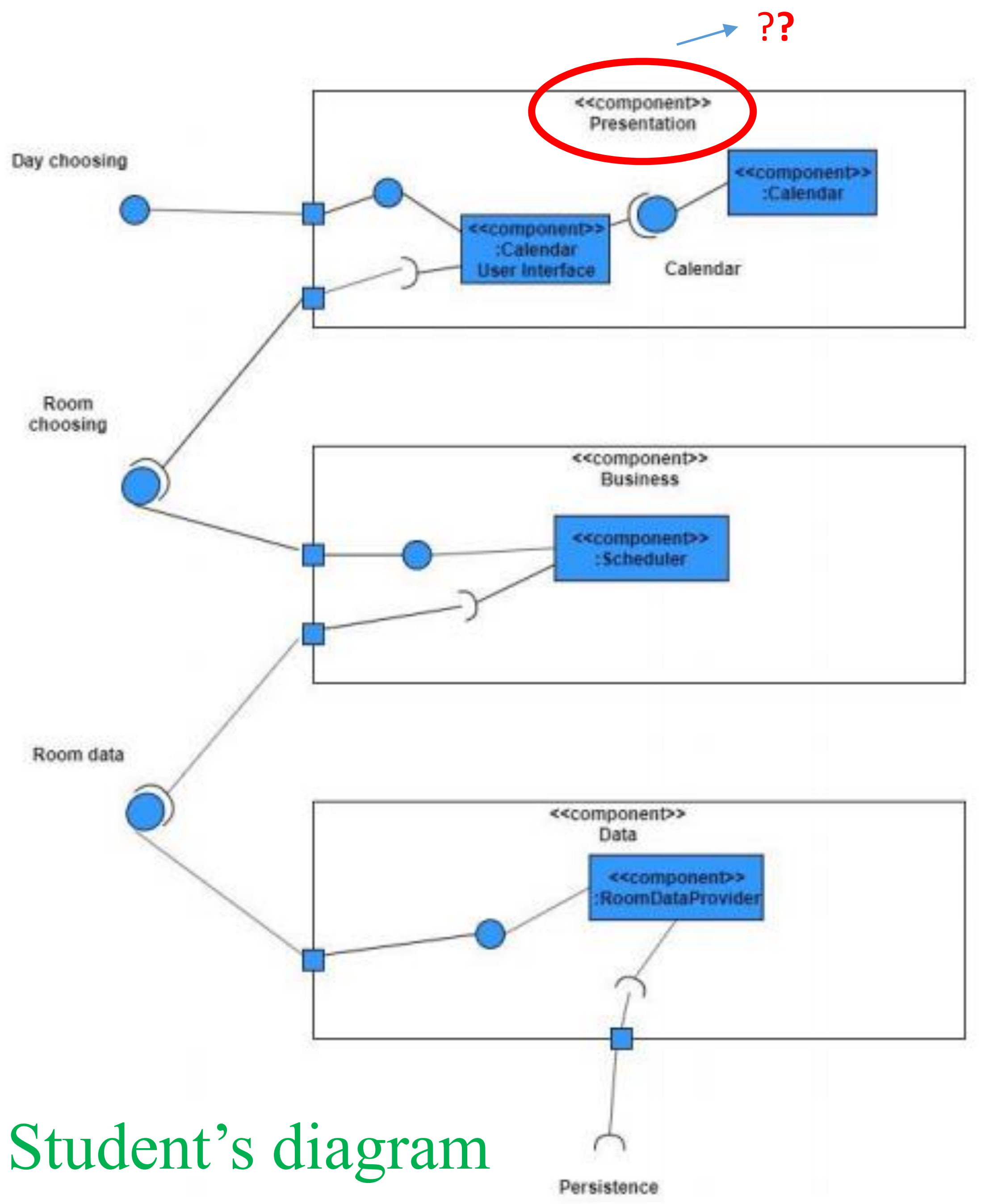
All elements of Library Domain package are public except for Account

https://www.uml-diagrams.org/package-diagrams.html#package


«use»
«use»
org.hibernate
«interface» SessionFactory
+ openSession(): classic.Session
+ openSession(java.sql.Connection): classic.Session
+ openSession(Interceptor): classic.Session
«interface» Session
+ connection(): Connection
+ close(): Connection
+ getSessionFactory(): SessionFactory
+ beginTransaction(): Transaction
+ createSQLQuery(): SQLQuery
org.hibernate.classic
«interface» Session
Includes deprecated methods from Hibernate2
«use»
org.hibernate.engine
«interface» SessionFactory Implementor
org.hibernate.impl
Session FactoryImpl
SessionImpl
org.hibernate.jdbc
Callback
JDBCContext
ConnectionManager
- callback: Callback
- factory: SessionFactoryImplementor
- connection: java.sql.Connection
+getConnection(): java.sql.Connection
-closeConnection(): void
+close(): java.sql.Connection

An example of UML package diagram for Spring and Hibernate data access classes.

https://www.uml-diagrams.org/spring-hibernate-uml-package-diagram-example.html



Student’s diagram

Three-Tier Pattern Applied to the Component Diagram ??

→ OK