PB007 - Software Engineering I

Seminar 12 - Package, Component & Deployment

Tereza Kinská

FI MUNI - Lasaris

WS 2022

E-test

- At the beginning of each seminar (starting from the 3rd seminar)
- 2nd seminar only demo
- You can open them only at the seminar **after being told to do so**
- You can get 2 points for each test, and there are 10 tests in total
- 4 questions, 0.5 points each
- The test will cover the topic of that week's seminar

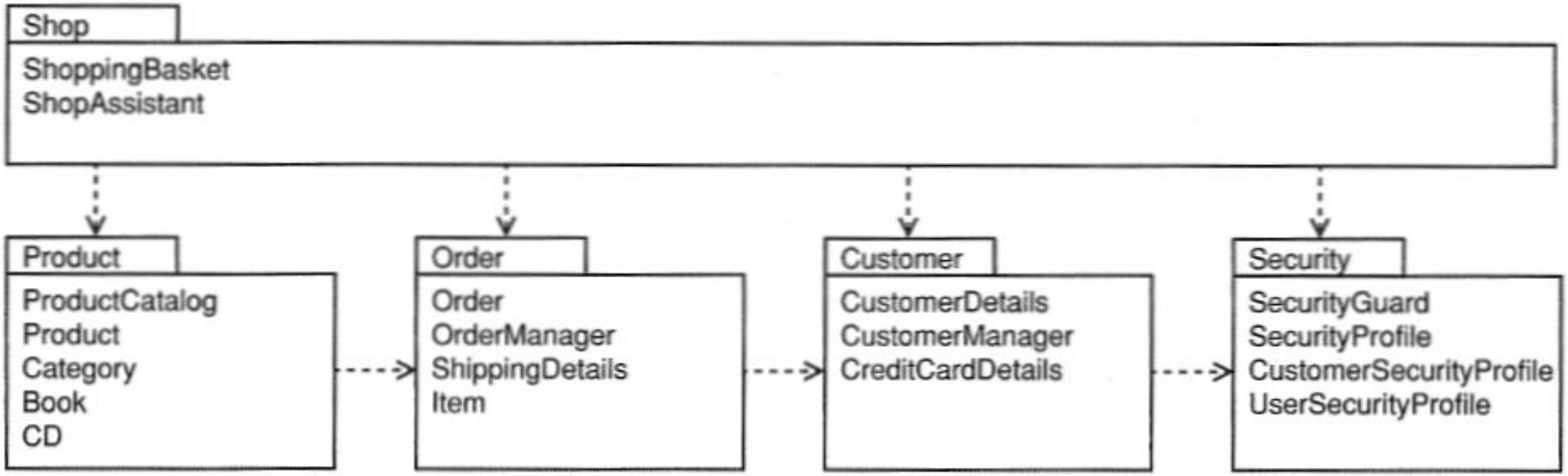
Package diagram

Package diagram models groups (packages) of related elements and their mutual relationships.

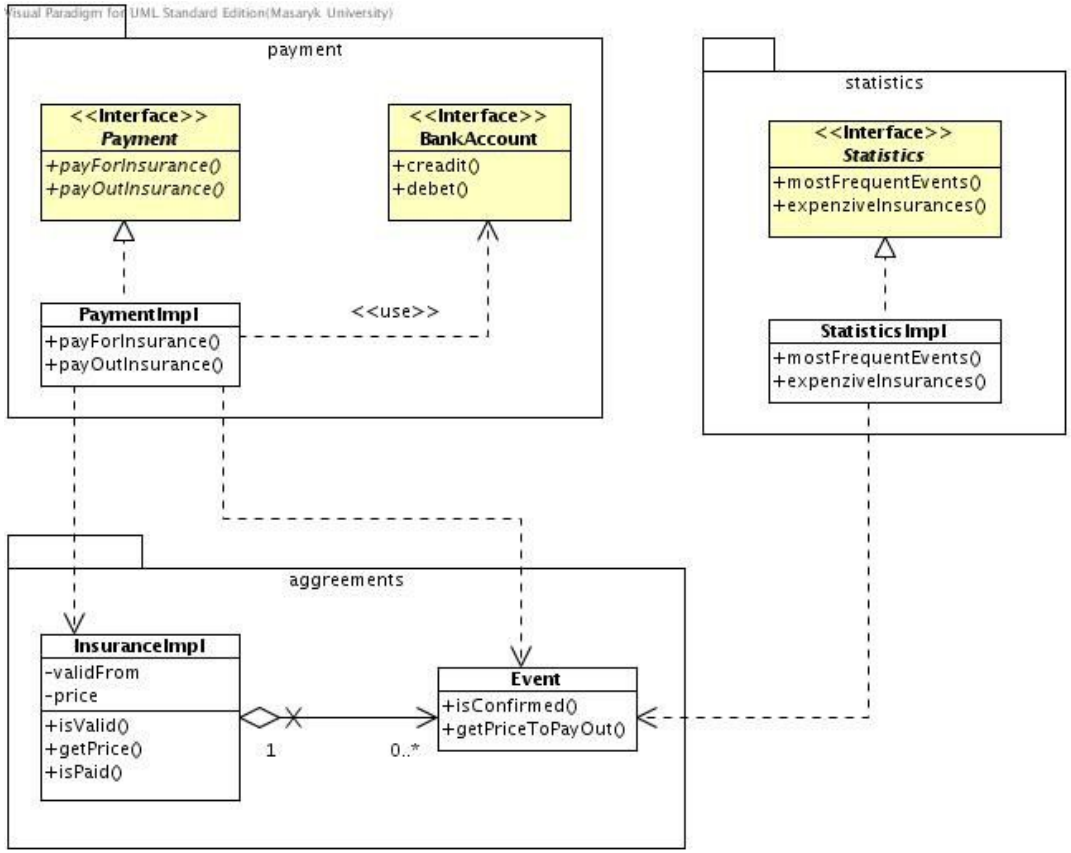
It consists of:

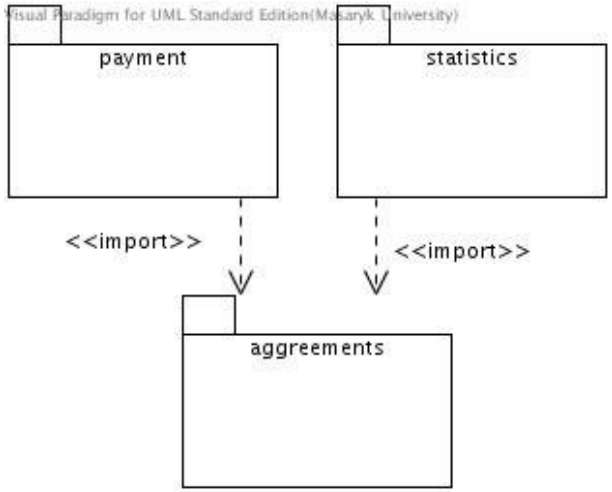
- **Packages** - represent a mechanism for logical grouping of related model elements (classes, objects, use cases,...); they define their namespace.
- **Dependencies** - indicate that elements from one package depend on the elements of another package. The type of dependency can be specified with stereotypes (`use`, `import`,...)

Package diagram - example 1

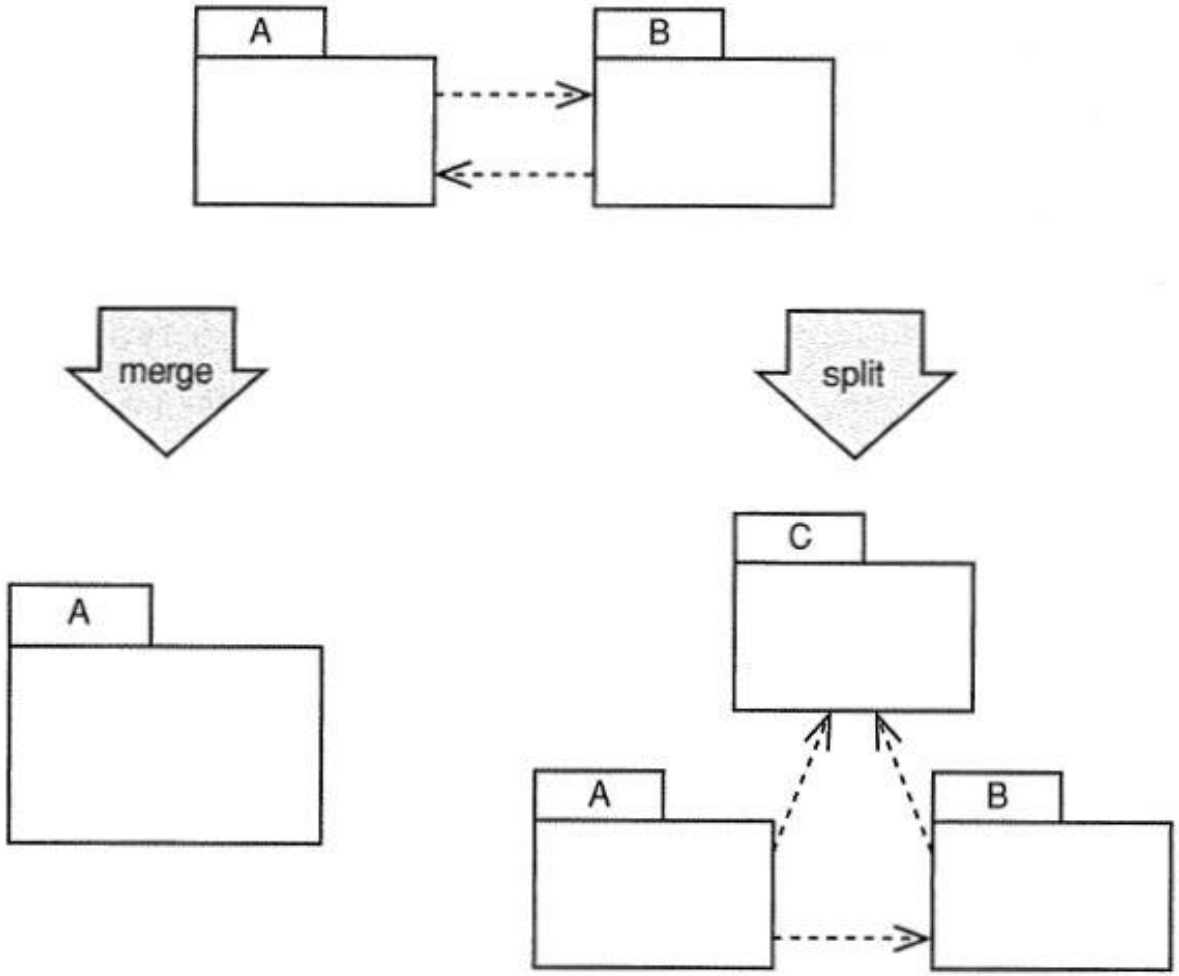


Package diagram - example 2





Package diagram - cyclic dependencies



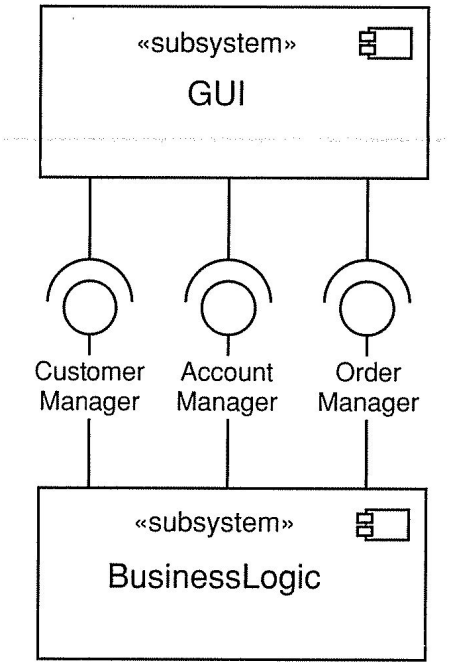
Component diagram

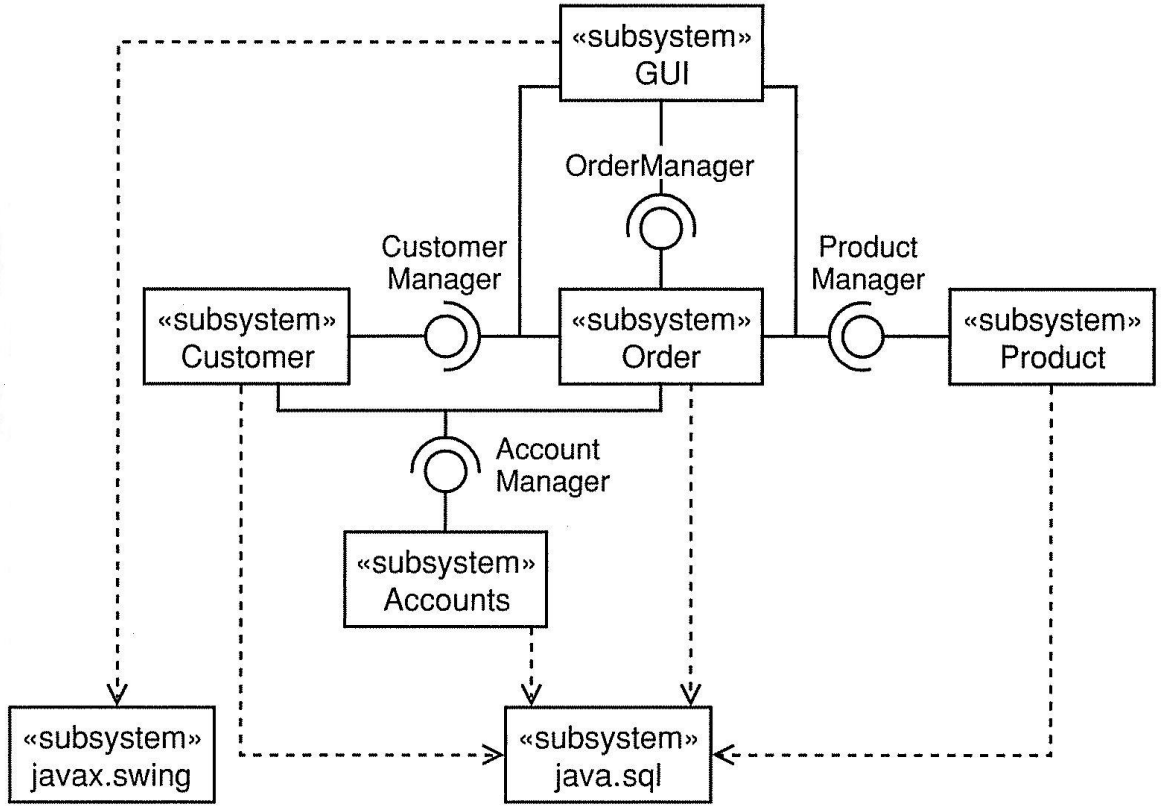
Component diagram models a hierarchical decomposition of system into separate parts and the communication relationships between them. It depicts a system's architecture.

It consists of:

- **Components** - software components (physically separate parts of the system), internally cohesive, externally communicate only via defined interfaces.
  - They can be *physical* (e.g. EJB) or *logical* (e.g. sybsystems)
  - They can consist of nested components.
- **Interfaces** - for communication between the components.
  - We distinguish *required* and *provided* interfaces
- **Relationships between interfaces** - connection of the *required* interface to the *provided* interface

Component diagram - example





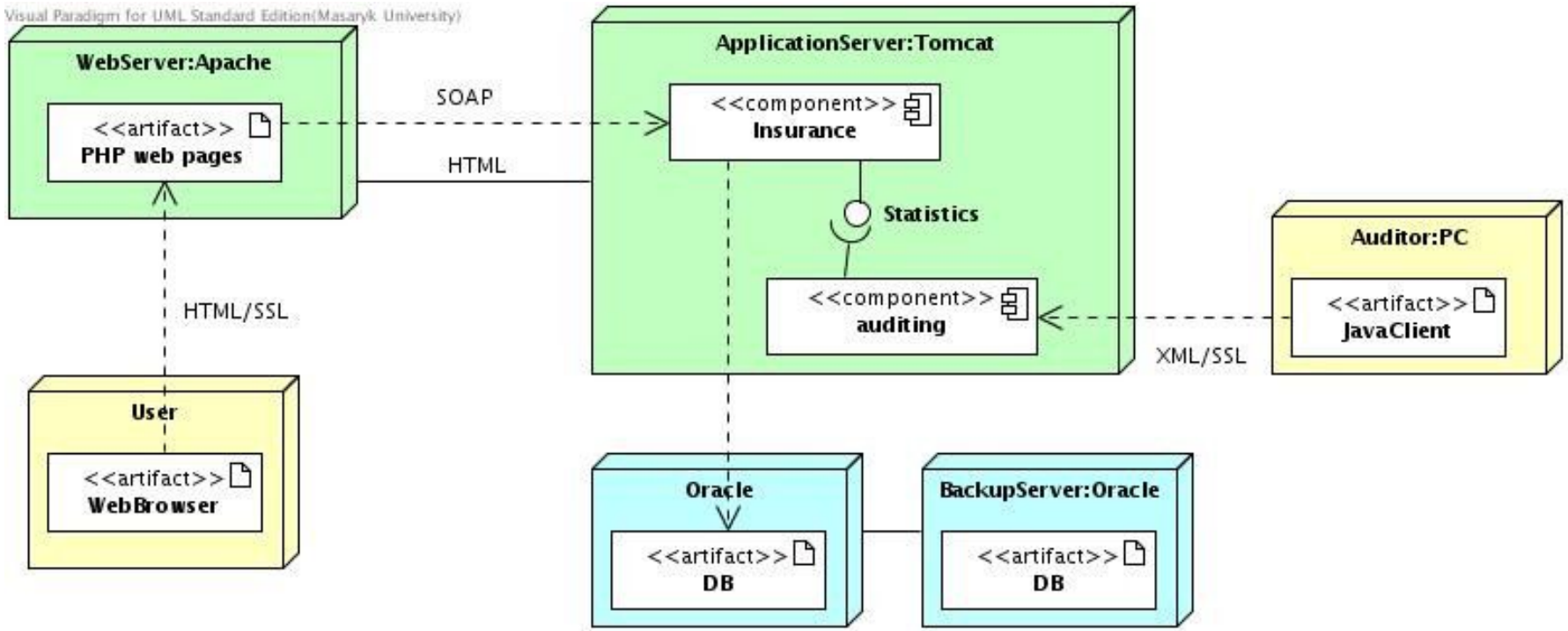
Deployment diagram

Deployment diagram models mapping of software architecture to a hardware architecture

It consists of:

- **Node** - computational resources, on which system parts will be deployed. They can be further specified with stereotypes, e.g. `device` or `execution environment`
- **Components/Artefacts**
- **Interfaces** – for communication between components
- **Associations/Dependencies** - connection between nodes (communication channels) and dependencies between components/artefacts. They can specify a communication protocol.

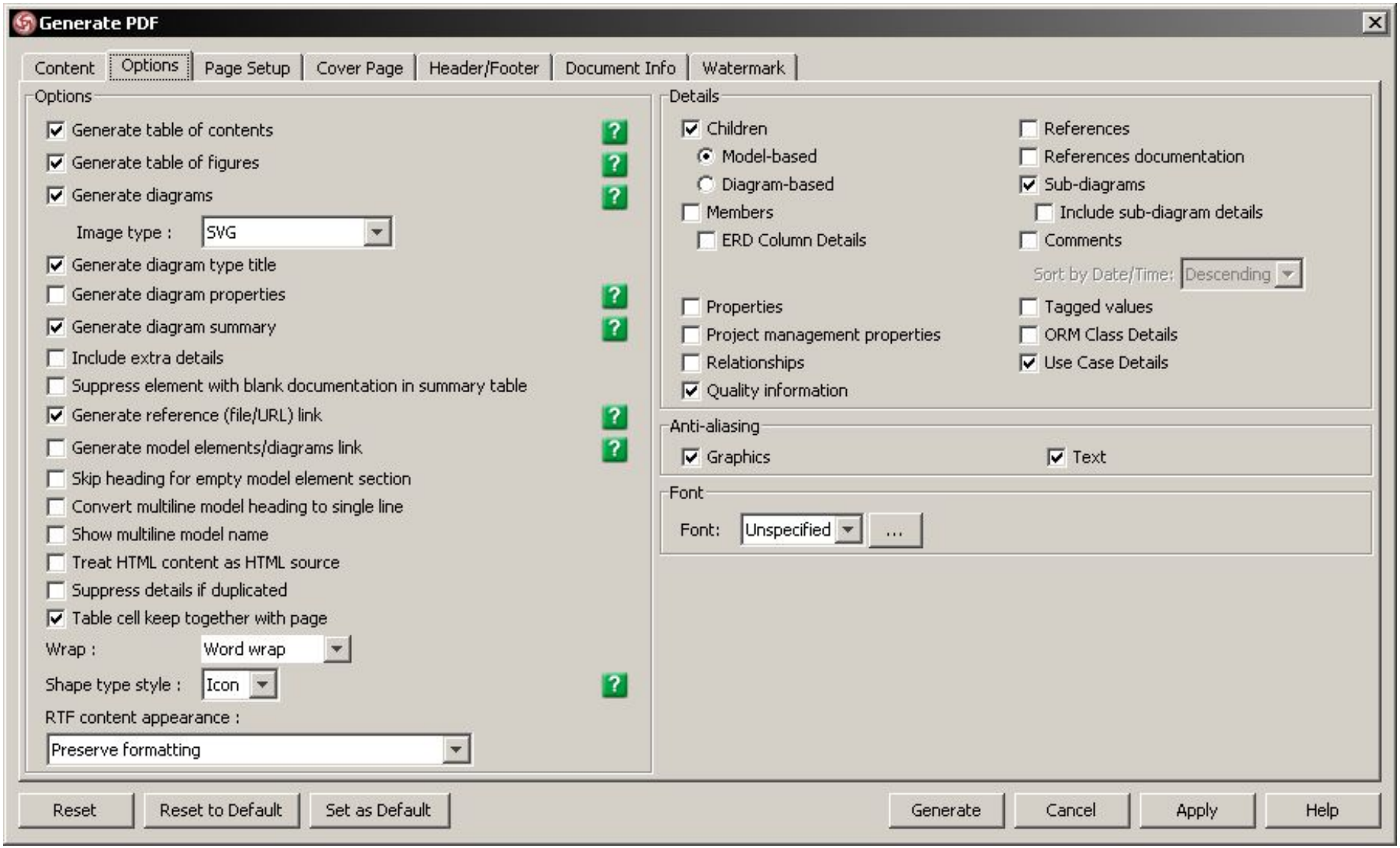
Deployment diagram - example



Rules for Report Submission

1. Submit the PDF report, not the VP source file and not an exported image.
2. PDF report must be created using the procedure shown on the seminars including the report settings.
3. The name of the PDF report file should be *lastname1-lastname2-lastname3* of the team members.
4. PDF report must contain all diagrams modelled until now.
5. PDF report must be uploaded to the homework vault by the specified deadline.
6. PDF report must be uploaded to the correct homework vault. The name of the homework vault is always specified on the slides.
7. Each team uploads only a single PDF report for the whole team.
8. Submitted diagrams must be clear and readable.
9. Submitted diagrams should not contain serious mistakes. At least, they should not contain mistakes mentioned in the *Catalogue of common mistakes*.

VP Report Settings



Tasks

- Adjust the previous diagrams according to the comments and to be consistent.
- Create a copy of the analytic class diagram. In the copy, add packages and move the classes into them.
- Draw dependencies between packages. Use stereotypes.
- 2 bonus points for early project completion (**until 17th December**; for exam on 14th December submit **until 10th December** and add word **final** to the file name)