

Package, Component, Deployment Diagram

PB007 Software Engineering I

Lukáš Daubner
daubner@mail.muni.cz

Package, Component, Deployment Diagram

PB007 Software Engineering I

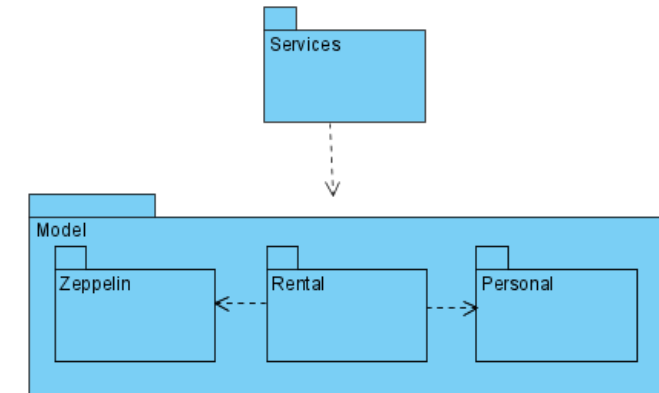
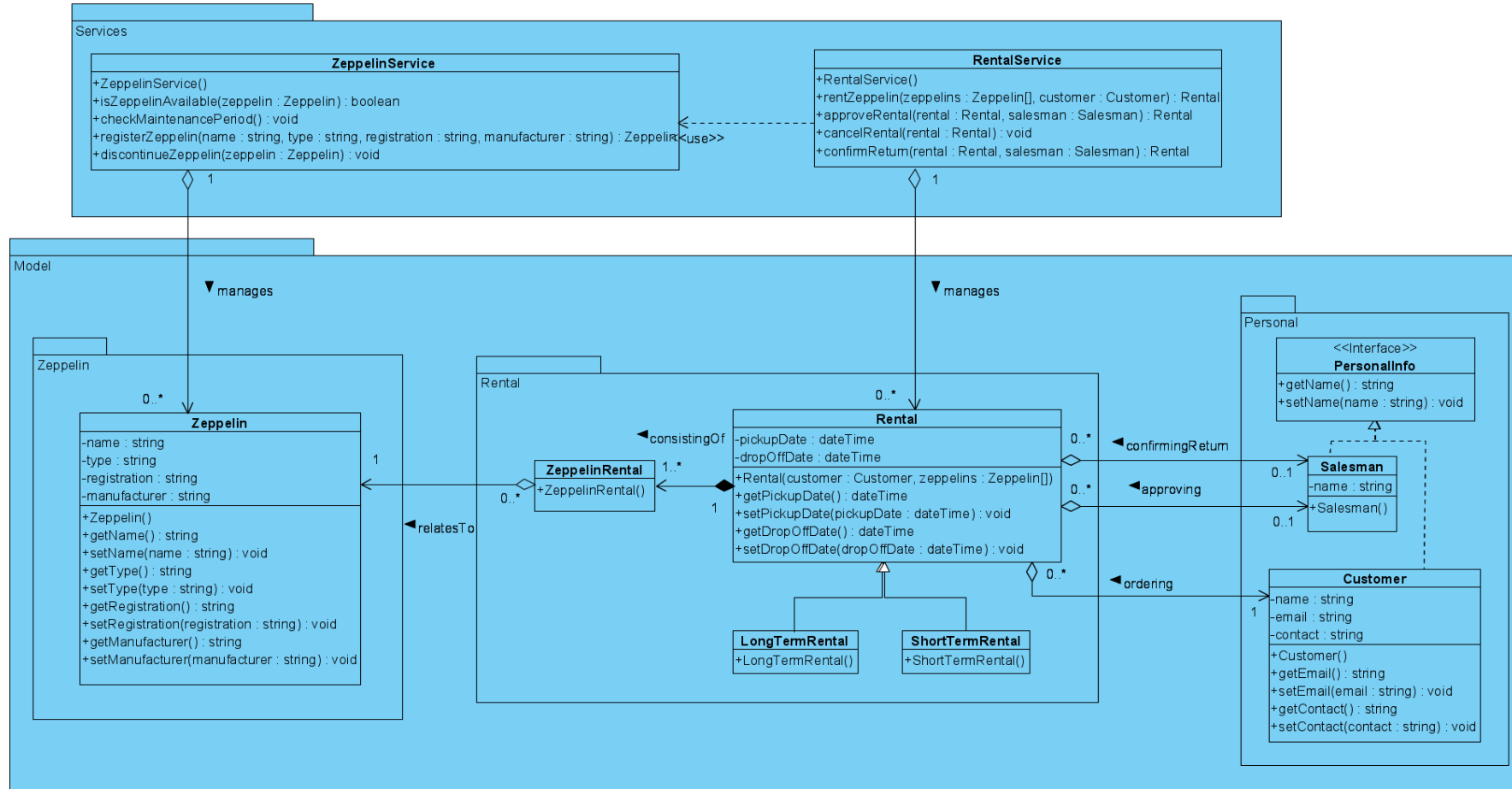
Lukáš Daubner
daubner@mail.muni.cz



Package Diagram

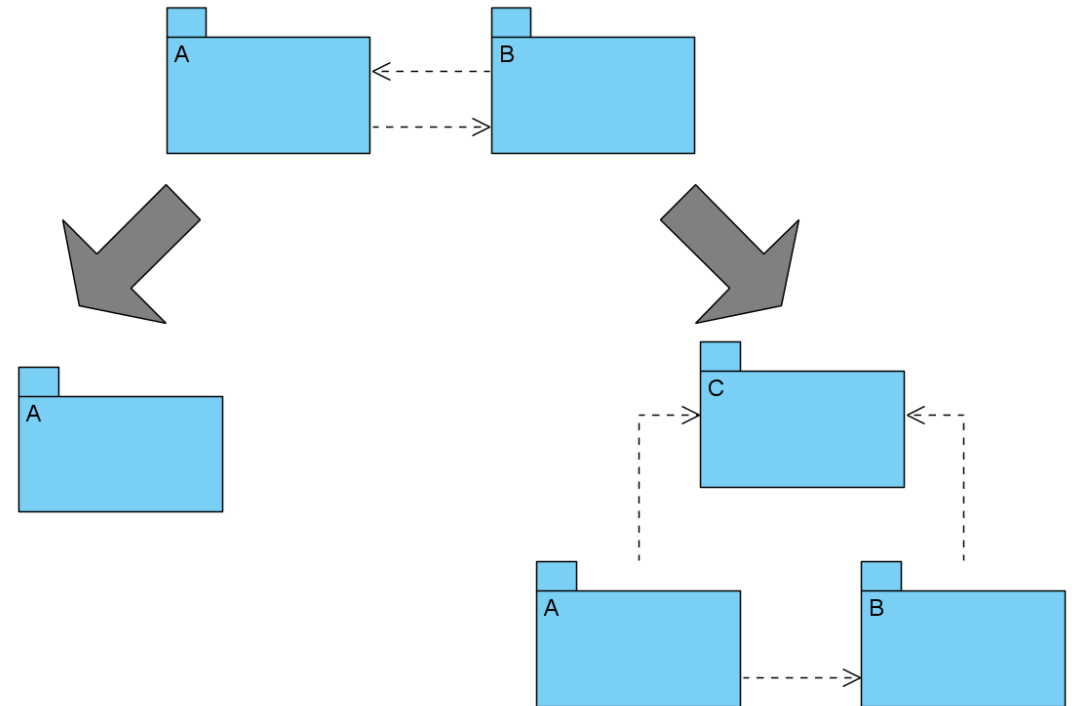
- Groups related elements and depicts dependencies between them
- The elements can be classes, use cases, etc...
- It includes:
 - Packages – represents logical grouping of elements and define their namespace
 - Dependencies – indicates dependency of elements from one package to another
- From object point of view, it is package/namespace like it is in OOP programming languages
 - Dependencies are then usage of “include/using” keywords

Package Diagram – Example



Package Diagram – Circular dependency

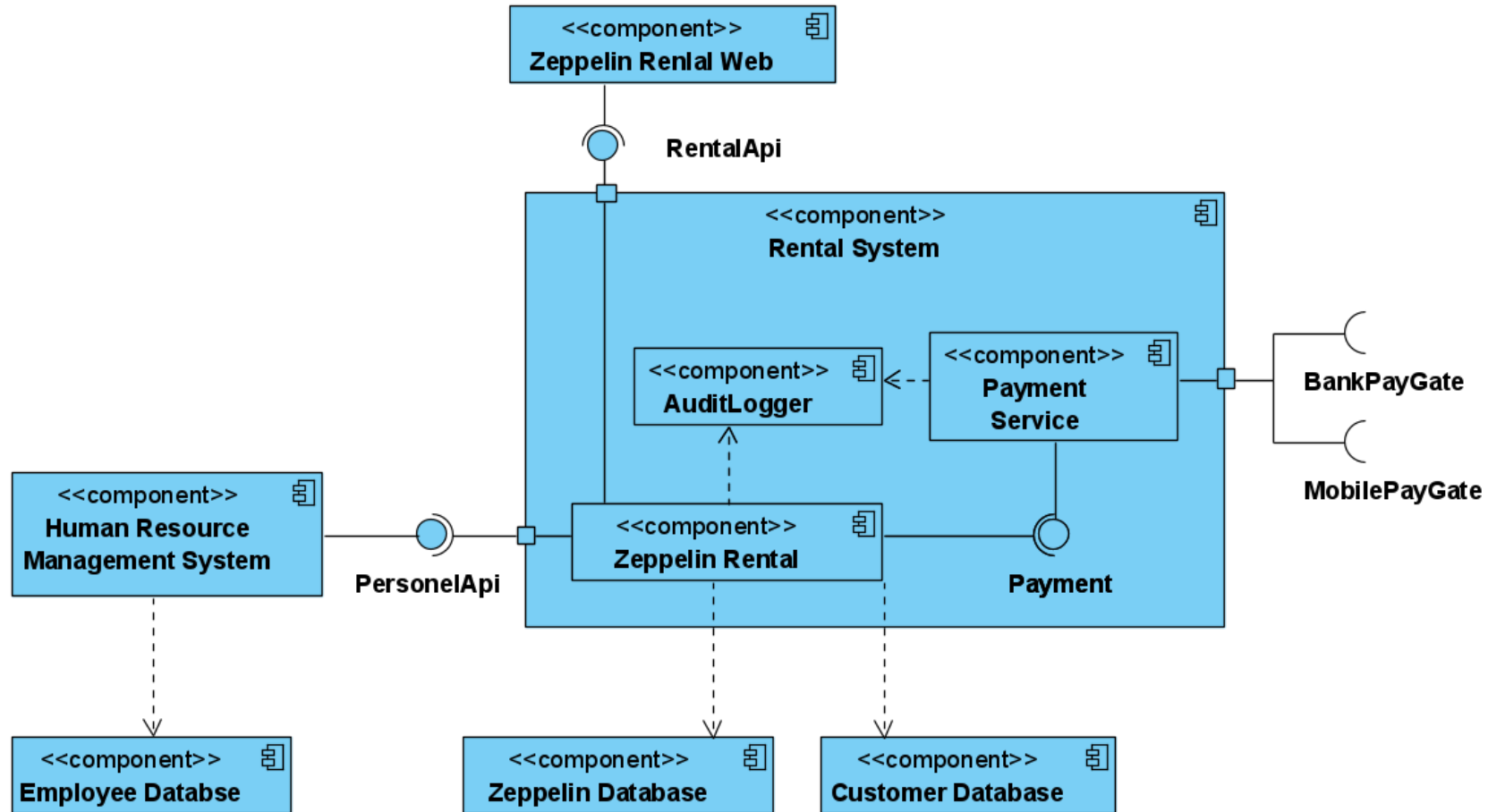
- All circular dependencies must be resolved
 - They can't be implemented
- Either by merge or split



Component Diagram

- Captures architecture of the system
- Modeled all parts of system
 - Executable files
 - Libraries
 - Static files
 - Database schema
 - Etc...
- Captures dependencies between components, especially interface
- Interface is contract used for communication between components
 - Provided interface – What the component can do
 - Required interface – What the component need

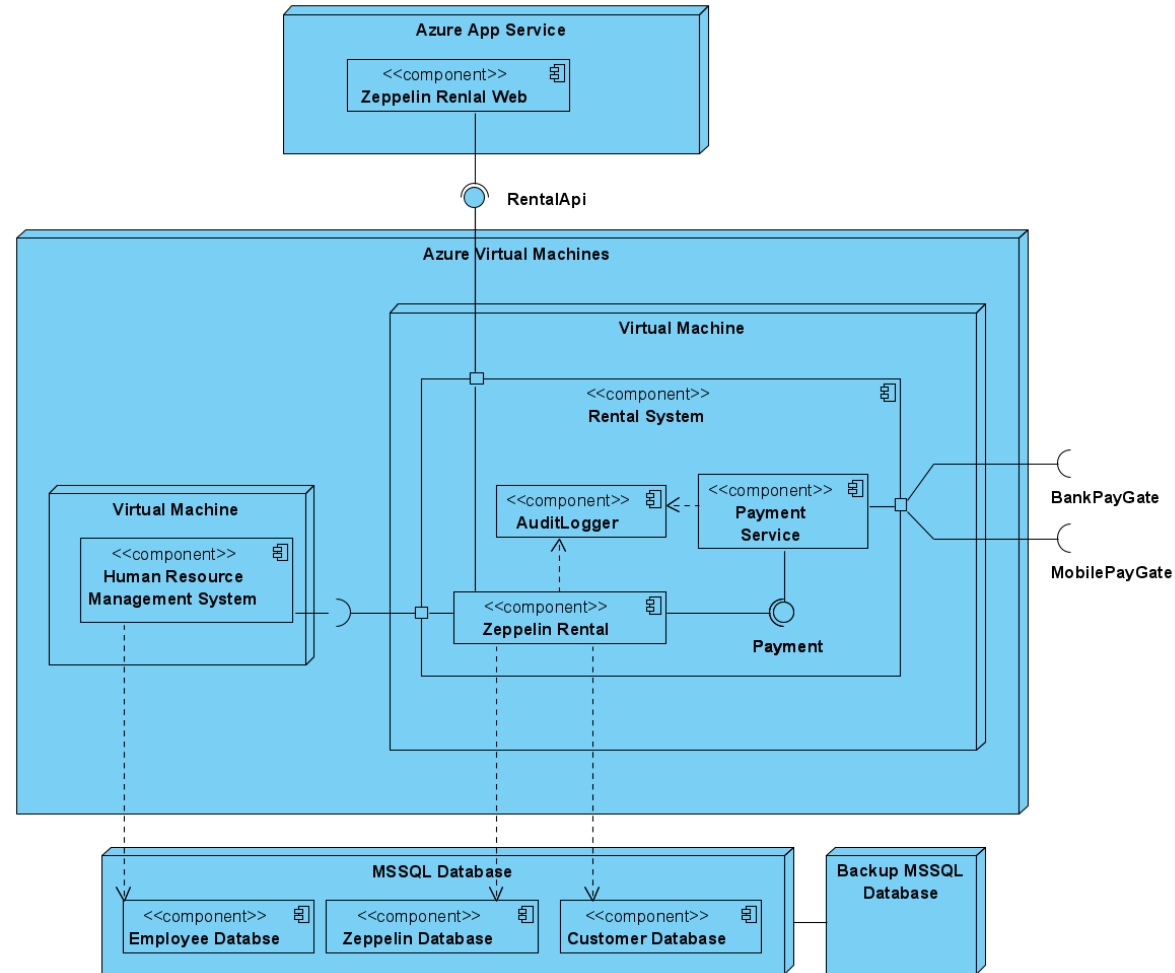
Component Diagram – Example



Deployment Diagram

- Depicts mapping of architecture on infrastructure
 - Hardware
 - Virtual servers
 - Containers
- You can model elements like backups, load balancing, mapping on cloud services, etc...
- It includes:
 - Nodes – General computational resources, specified by stereotypes
 - Components, Artifacts
 - Interfaces
 - Links, Dependencies – Connection between the components. They should include information about used communication protocol

Deployment Diagram – Example



Task for this week

You gotta do what you gotta do

- Process the feedback
- Copy the Design Class Diagram and create packages
 - Create Package Diagram
 - Mind the cycles! Directions of associations/dependencies will tell you
- Think about components of your system and how they communicate
 - You might need some database schema, caches, web pages, sensors, etc...
 - Interfaces are important
 - Create Component Diagram
- Create Deployment Diagram of the proposed system
 - The system must run/be hosted somewhere
- Do your part in peer review