FKA The Toolbox
工具箱

REST based contracts in large systems

Anders Kalhauge - Rolf-Helge Pfeiffer - Jens Egholm Pedersen

cphbusiness
COPENHAGEN BUSINESS ACADEMY

Fall 2017

The goal of the day

cphbusiness
COPENHAGEN BUSINESS ACADEMY

- You all understand the toolbox as a sound alternative, somewhere between the extreme formalization in Design by Contract, and no formalization in natural language contracts.
- You will know different means to define contracts between subsystems
- You will master the central elements in the toolbox.

Agenda

cphbusiness
COPENHAGEN BUSINESS ACADEMY

- Presentation of Assignment 6
- Recap on the toolbox as a practical example to contract based software development
  - focus is on vertical contracts: front-end $\longleftrightarrow$ back-end
- Presentation of alternative contract formats

Formerly Known as The Toolbox

What's in the box?

cphbusiness
COPENHAGEN BUSINESS ACADEMY

Overview
What's in the box?

cphbusiness
COPENHAGEN BUSINESS ACADEMY

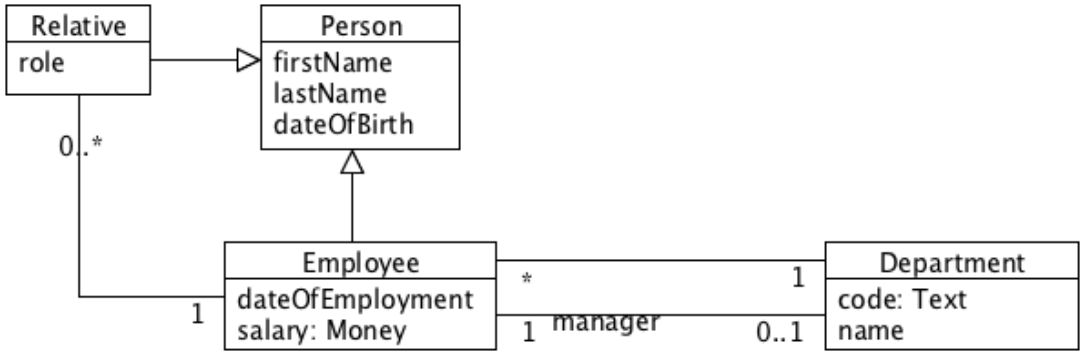
- Logical data model
- Use case model
  - Use case diagram(s)
  - Use case descriptions
  - System sequence diagram
  - System operation contracts
- Communication model
  - System operation contracts
  - Transfer objects
    - Data Transfer Objects (DTOs)
    - Exception Transfer Objects (ETOs)
- Verification strategy

Logical data model
What is a logical data model?

cphbusiness
COPENHAGEN BUSINESS ACADEMY

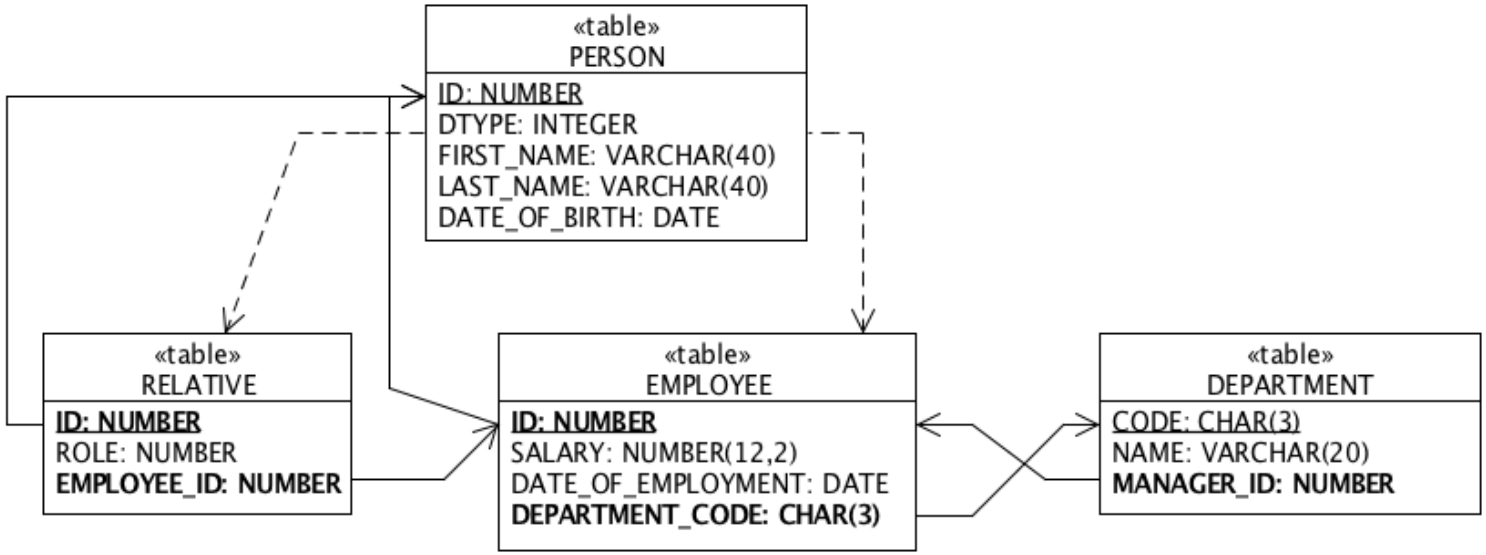
- It models the system state.
- Expresses valid pre- and postcondition states.
- Expresses possible system state changes.

Logical data model
An example



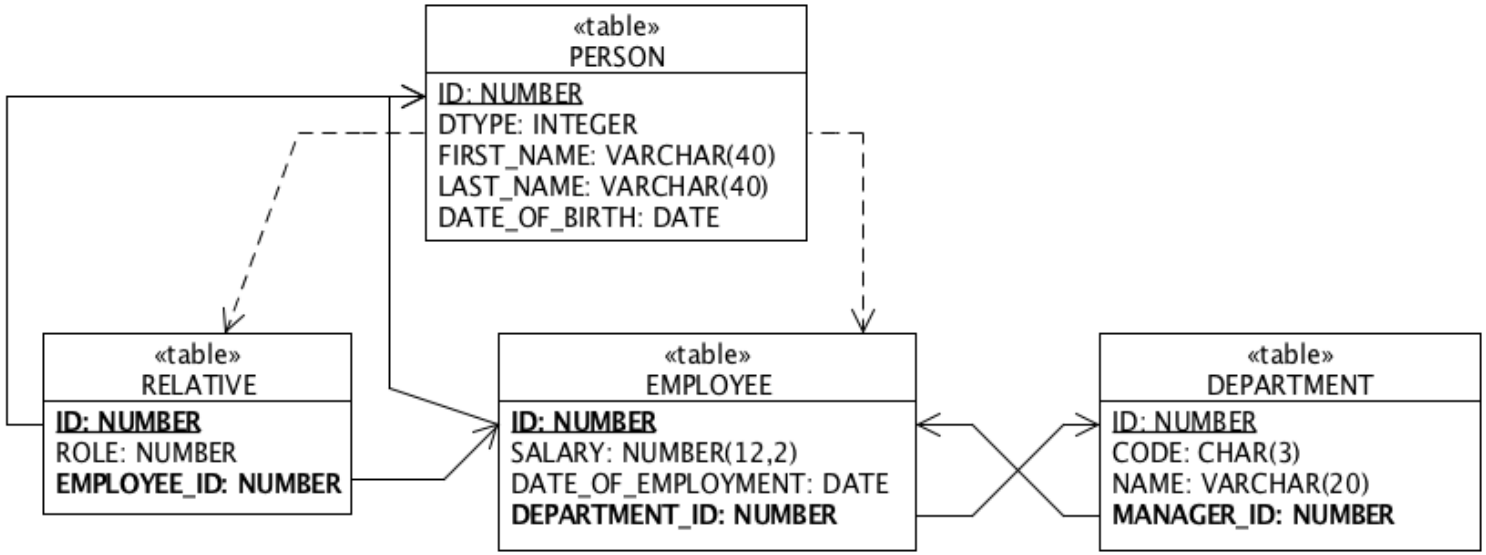
- What should be persisted
- Only entities
- No implementation details
  - No ids unless they contain data (not necessarily wise)
  - Only abstract types

Not a logical data model
but a relational implementation



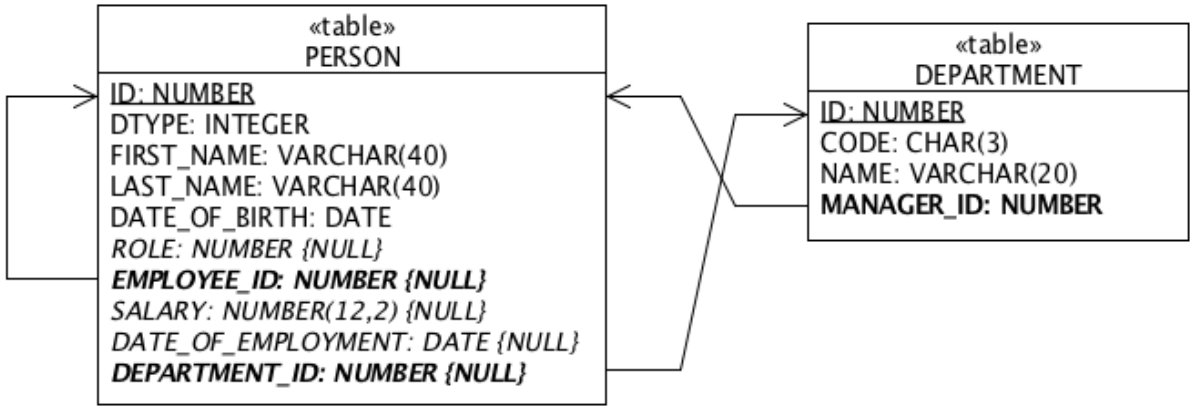
- Primary (underlined) and foreign (**boldfaced**) keys shown.
- Joined tables inheritance strategy, `DTYPE` discriminates between types.

Not a logical data model
but another relational implementation



- Same as I, with no data bearing primary keys ☺

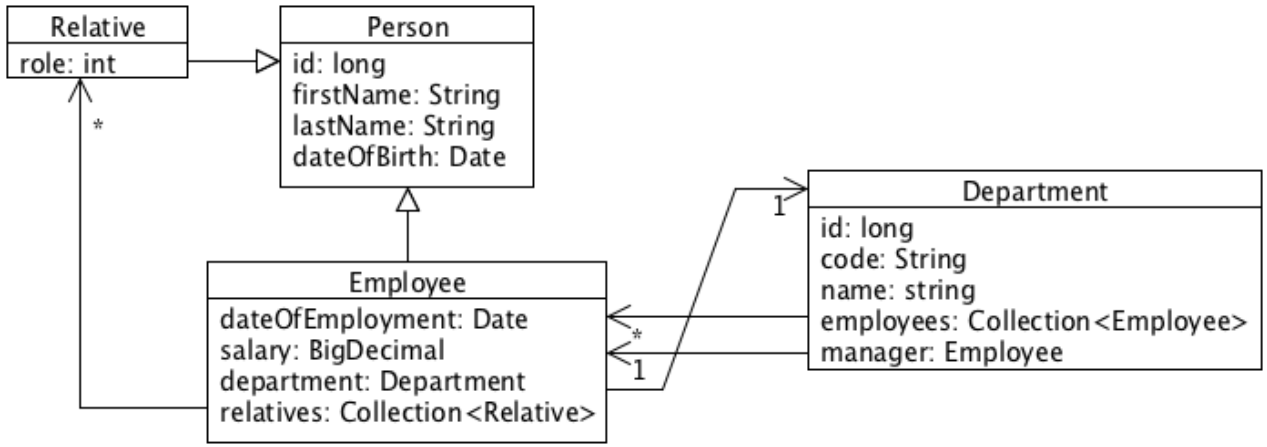
Not a logical data model
but a third relational implementation



- Single table inheritance strategy
- “Irrelevant fields are nulled

Still **not** a logical data model

but a implementation with objects



- No associations, only references
- Id's to support "Object Relational Mapping"

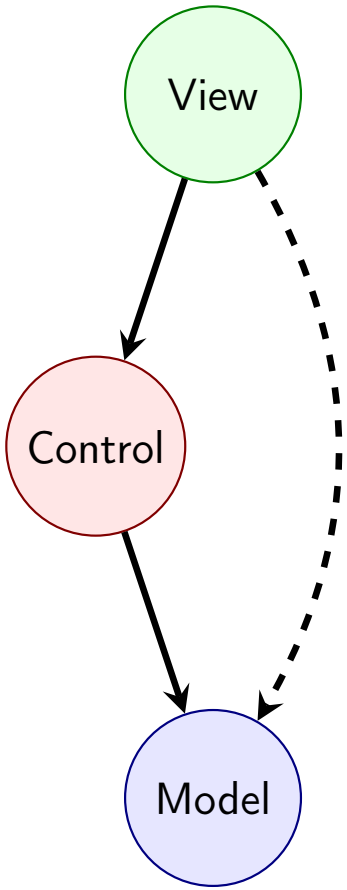
Use case model

cphbusiness
COPENHAGEN BUSINESS ACADEMY

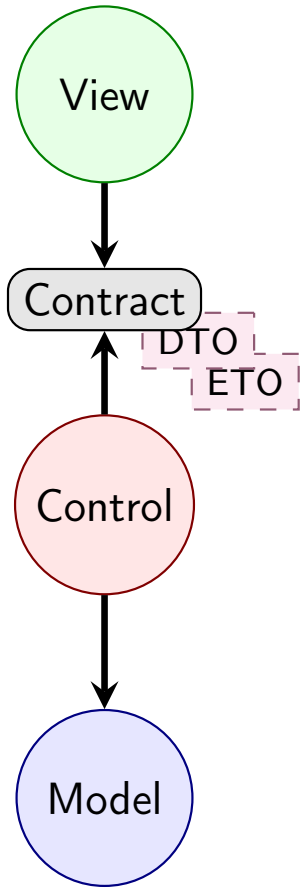
Goes here . . .

Communication model

System operation contracts - MVC pattern reviewed



(*a*)

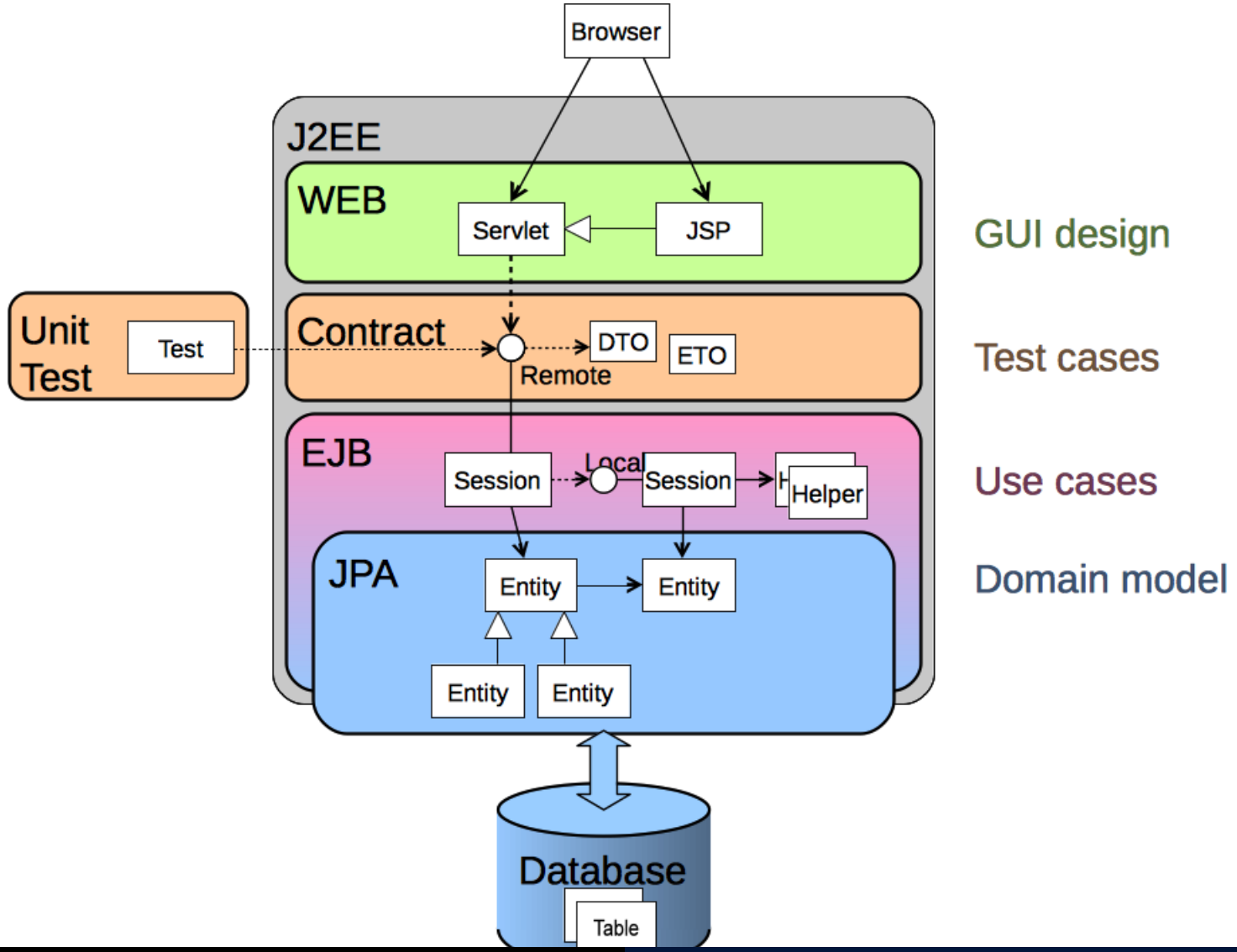


(*b*)

Communication model

System operation contracts - Layers in EJB

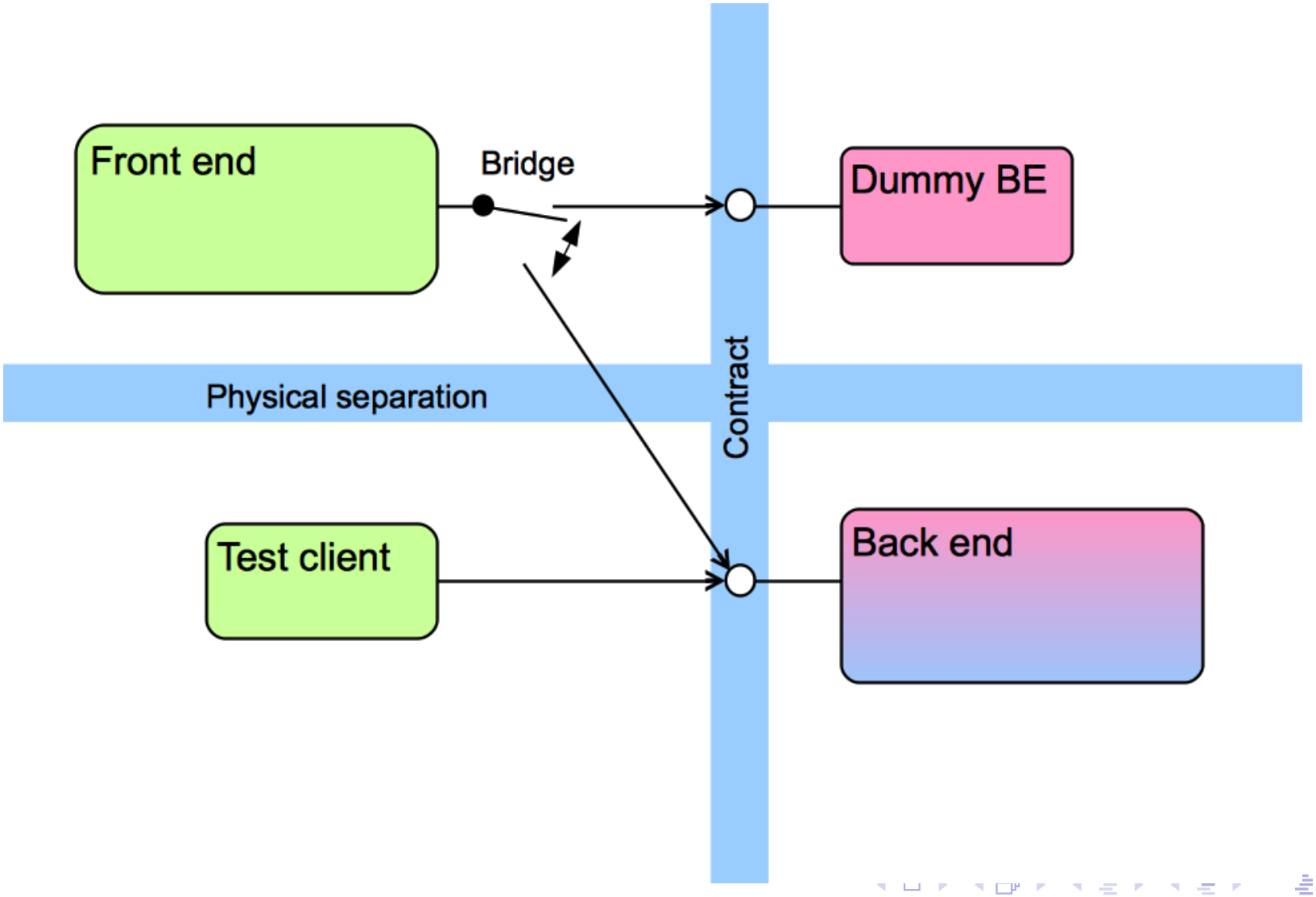
cphbusiness
COPENHAGEN BUSINESS ACADEMY



Communication model

Project setup with bridge

cphbusiness
COPENHAGEN BUSINESS ACADEMY



Communication model

System operation contracts - "Remote" interface

The interface is the code based operation contract.

- Use strong typing.
  - use DTOs instead of simple data types.
- Make inline documentation (JavaDoc)
  - have documentation close to code - easier to update.
  - generates written code contracts.
- Implement the interface with a Remote facade in the "backend".
  - Changes to the backend code or to the interface will have less impact.
- Reference the interface from a Factory in the "frontend".
  - Change of backend can be done with practically no code changes in "frontend"

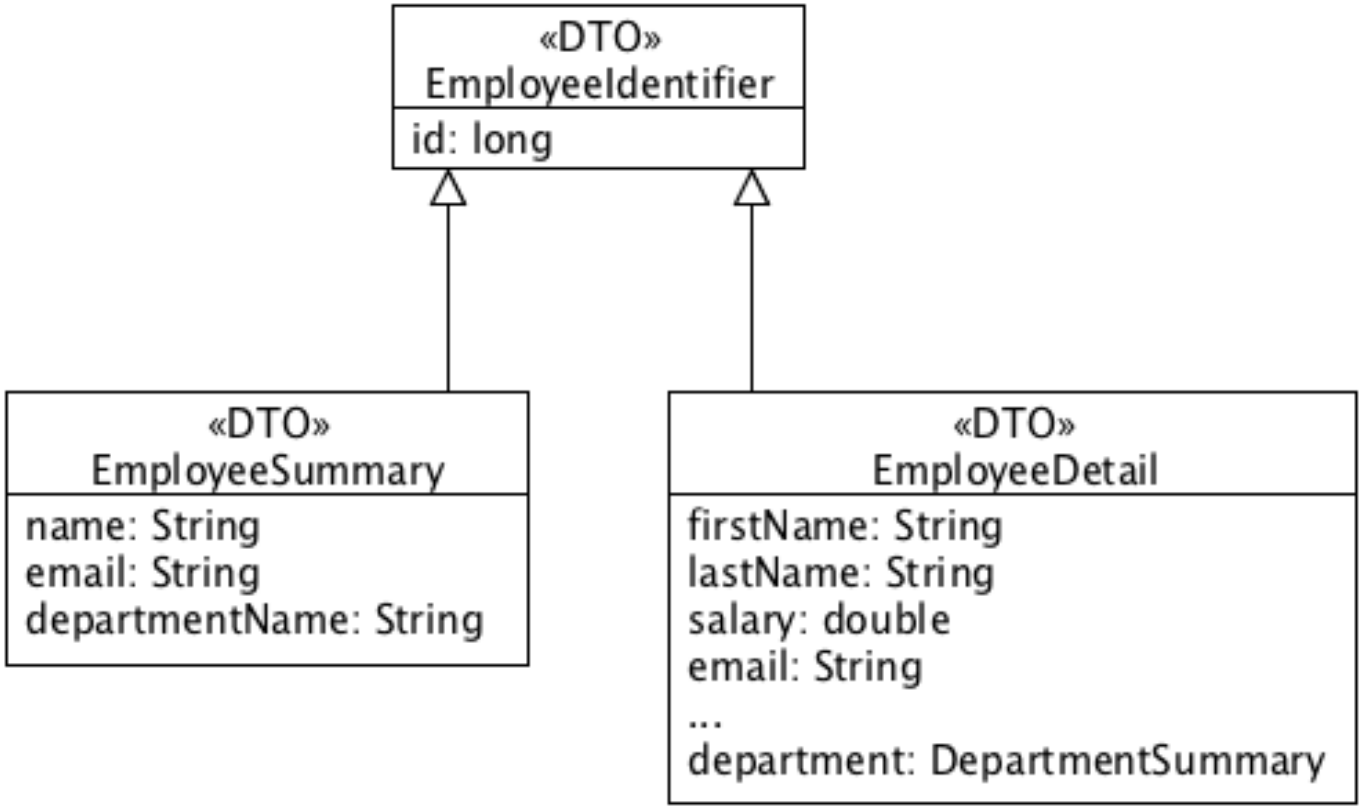
Communication model

System operation contracts - Data Transfer Objects

Data transfer objects should be as abstract as possible when still being concrete. Use DTOs for request and return values.

- Efficiency
  - Packing related data together
  - reducing calls - network calls are expensive to establish
  - reducing data - bandwidth is still an issue
- Encapsulation
  - by hiding irrelevant or secret data
  - **by hiding actual implementation**
- Serializable

Communication model
Data Transfer Objects - example



Communication model

System operation contracts - Exception Transfer Objects

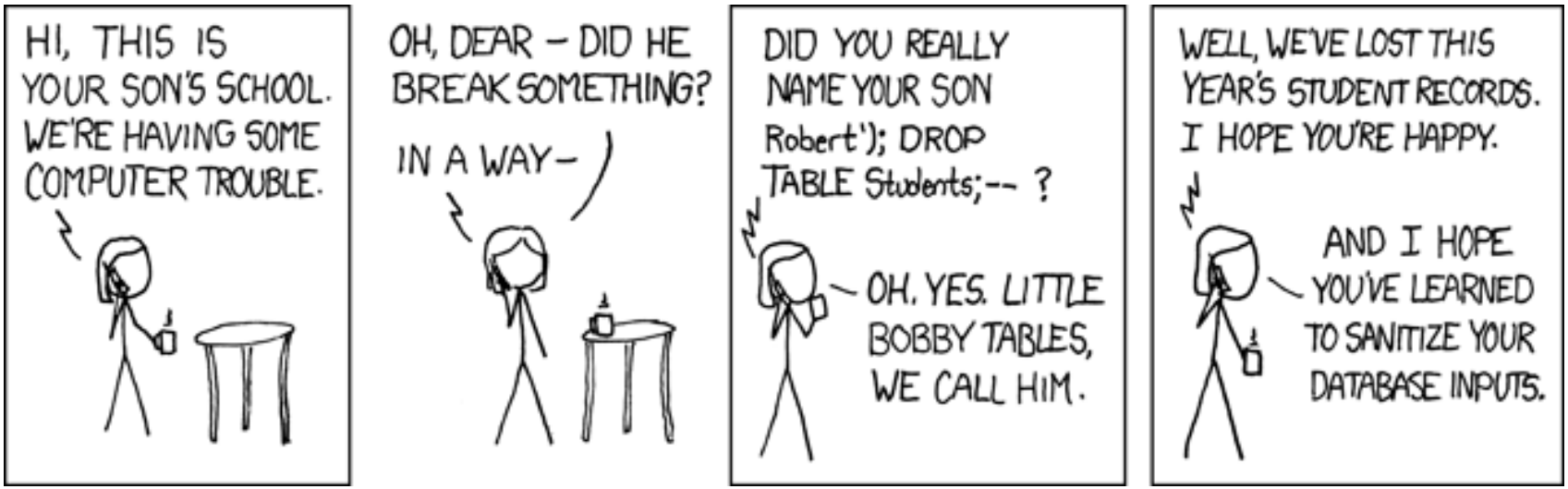
cphbusiness
COPENHAGEN BUSINESS ACADEMY

Exceptions are as valid, even less happy, return values from operations.

- User friendly - return only relevant information.
  - Preconditions: What precondition was violated (unchecked).
  - Postconditions: What went wrong (checked).
- Encapsulation
  - by hiding actual implementation
  - **revealing errors and their precise cause, is pleasing hackers**
- Serializable - in Java Exceptions are already Serializable

Communication model

Exception Transfer Objects - Alternative

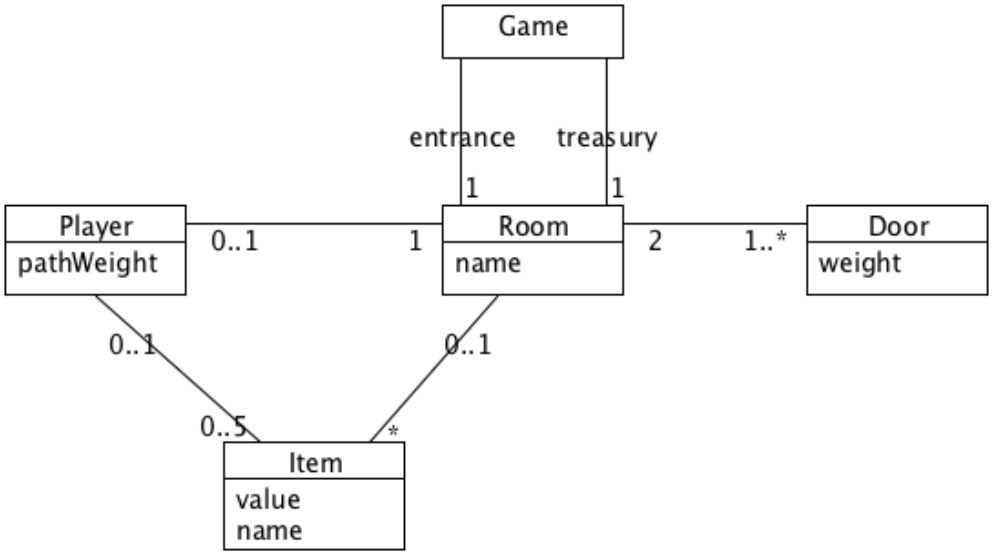


Dungeon game example
Requirements

cphbusiness
COPENHAGEN BUSINESS ACADEMY

We want a Dungeon game. The scenario of the game is a number of connected rooms or dungeons in a mountain. The player enters the mountain from the entrance room, and he/she should travel from dungeon to dungeon until he/she reaches the treasury room. All dungeon has doors that leads to at least one other dungeon. A dungeon can contain an unlimited number of items. Items have values. When a player is in the room he/she can see the items in the room, and he/she can see the doors leading from the room to other dungeons. The player can pick up and lay down items when he/she is in a room. But he/she can keep at most five items at a time. The quest is to reach the treasury room with the most expensive items through the shortest path. The game should run on a central server, and played throug a mobile phone connected to the server.

Dungeon game example
Logical data model

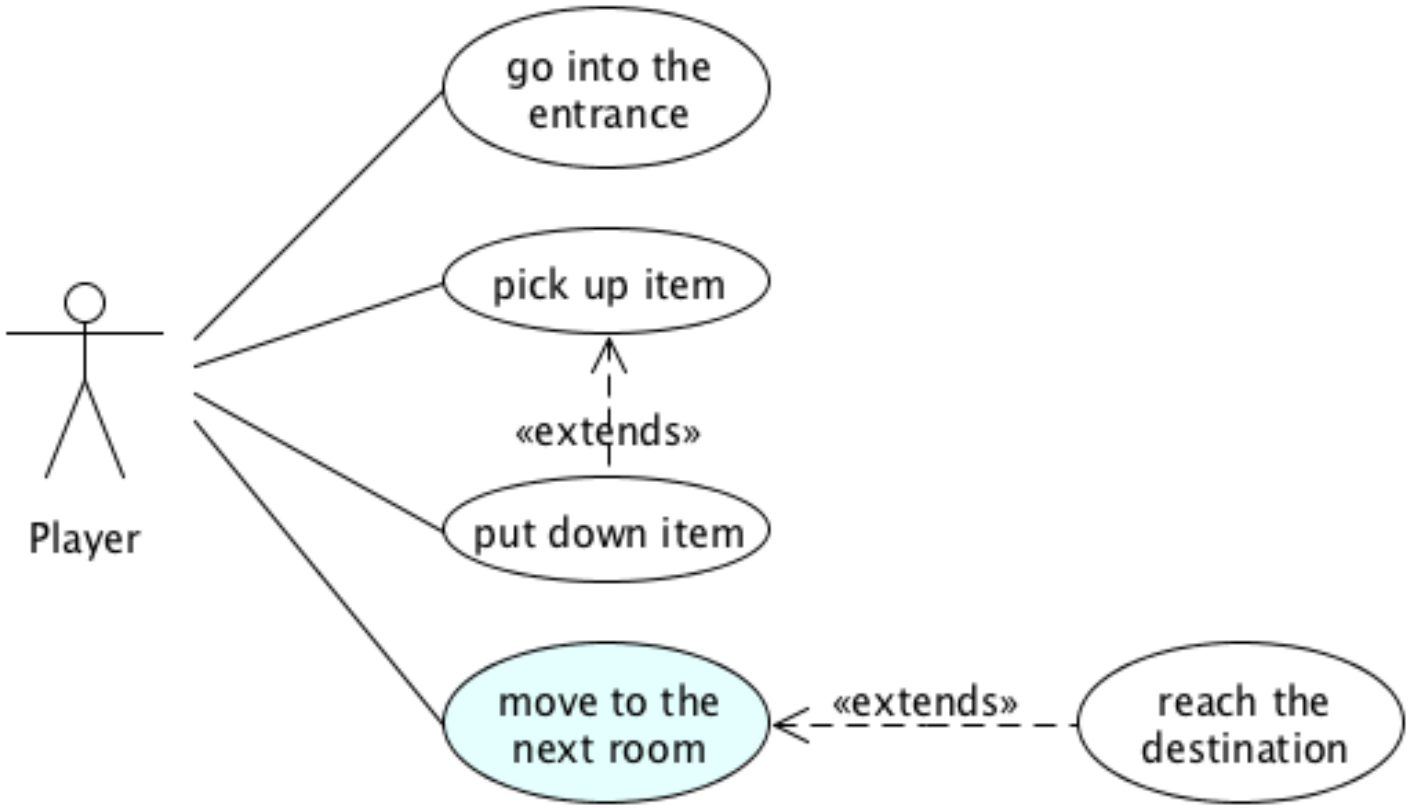


Again:

- Nouns from the requirements (glosary) are candidates
- What should be persisted
- Only entities
- No implementation details

Dungeon game example

Use Case Model - Use Case Diagram



Dungeon game example

Use Case Model - Detailed Use Case Description...

cphbusiness
COPENHAGEN BUSINESS ACADEMY

- **Name** Move to the next room
- **Scope** System under design (SuD)
- **Level** Goal: Move to the next room
- **Primary Actor** Player
- **Precondition** The player is in a room
- **Main succes scenario** ...
- **Success guaratees** The player is in a new room
- **Extensions** Reach the destination if room is treasury room
- **Special Requirements** NONE

Dungeon game example

Use Case Model - . . . Detailed Use Case Description

cphbusiness
COPENHAGEN BUSINESS ACADEMY

- **Name** Move to the next room

. . .

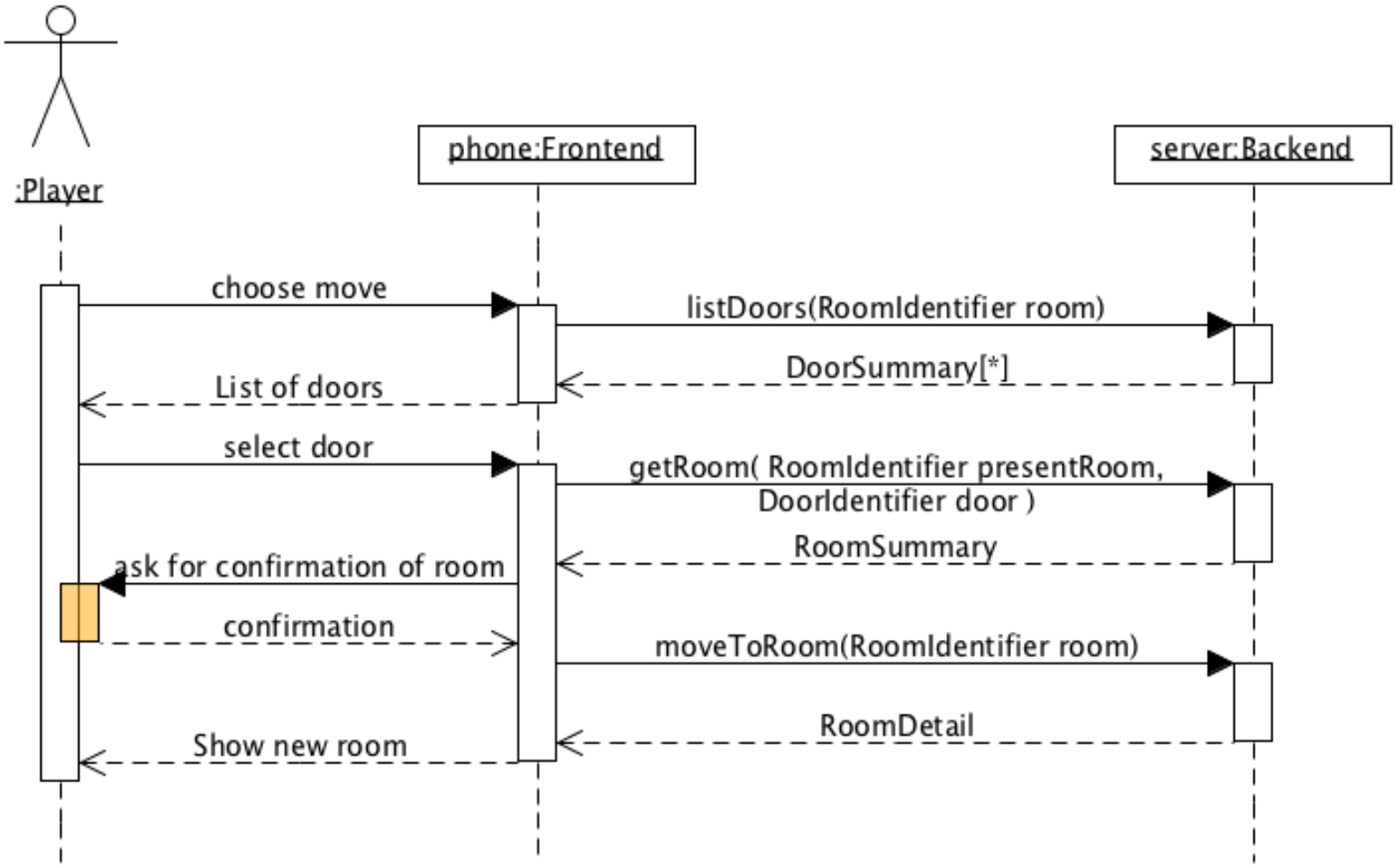
- **Main succes scenario**

1. Player chooses “move”
2. System shows a list with all doors to other rooms
3. Player selects the door he/she wants to move through
4. System shows the room name, and asks the player to confirm
5. Player confirms the selection of door
6. System moves the player to the room behind the selected door

. . .

Dungeon game example

Use Case Model - System Sequence Diagram



Dungeon game example

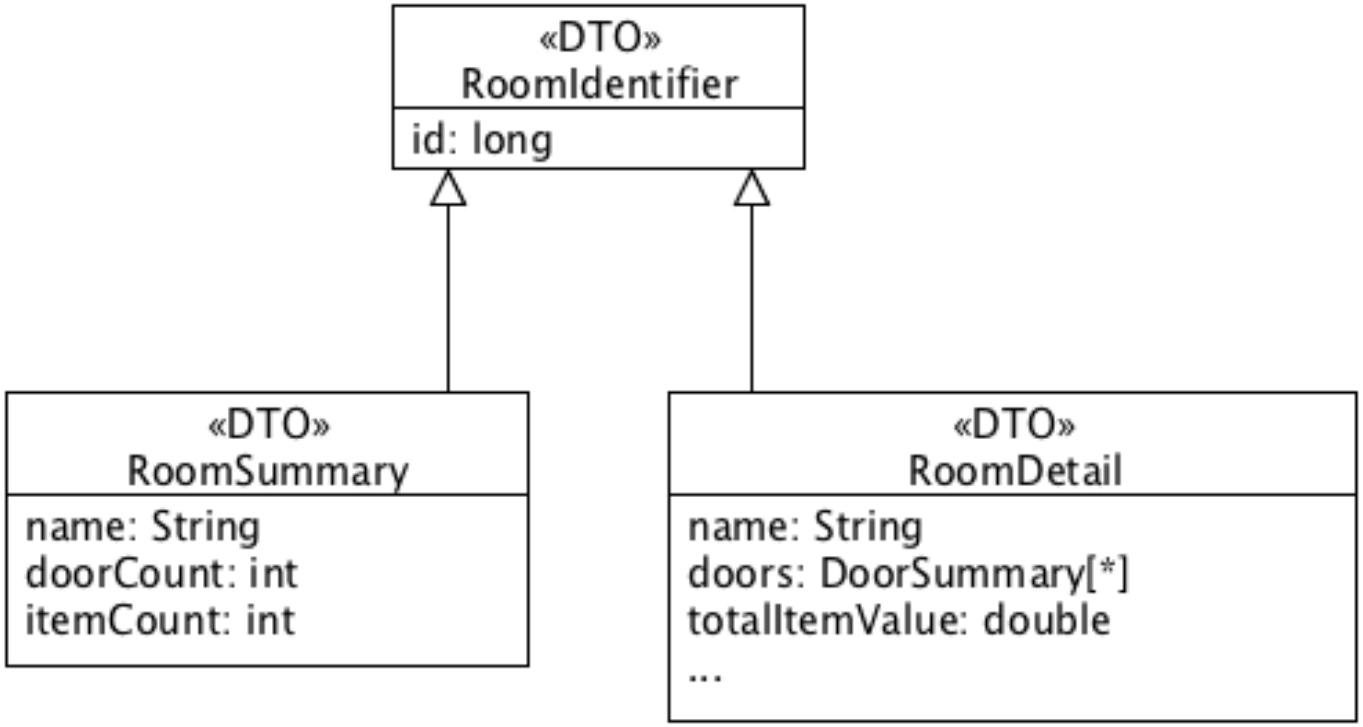
Communication model - "Remote" interface

cphbusiness
COPENHAGEN BUSINESS ACADEMY

```
@Remote
public interface DungeonManager {
  ...
  /**
   * List the doors leading from a given room.
   * @pre the room cannot be null and must exist.
   *    @throws NoSuchRoomException room doesn't exist.
   *    @param room the given room.
   * @post the doors in the given room is returned
   *    @return A collection of door summaries.
   */
  Collection<DoorSummary> listDoors(
      RoomIdentifier room
      );
  RoomSummary getRoom(
      RoomIdentifier room, DoorIdentifier door
      );
  RoomDetail moveToRoom(RoomIdentifier room);
  }
```

Dungeon game example

Communication model - Data Transfer objects



Dungeon game example

Communication model - Data Transfer objects

cphbusiness
COPENHAGEN BUSINESS ACADEMY

```
public class RoomIdentifier implements Serializable {
  private long id;

  public RoomIdentifier(long id) {
    this.id = id;
    }

  public long getId() { return id; }
  }
```

Dungeon game example

Communication model - Data Transfer objects

cphbusiness
COPENHAGEN BUSINESS ACADEMY

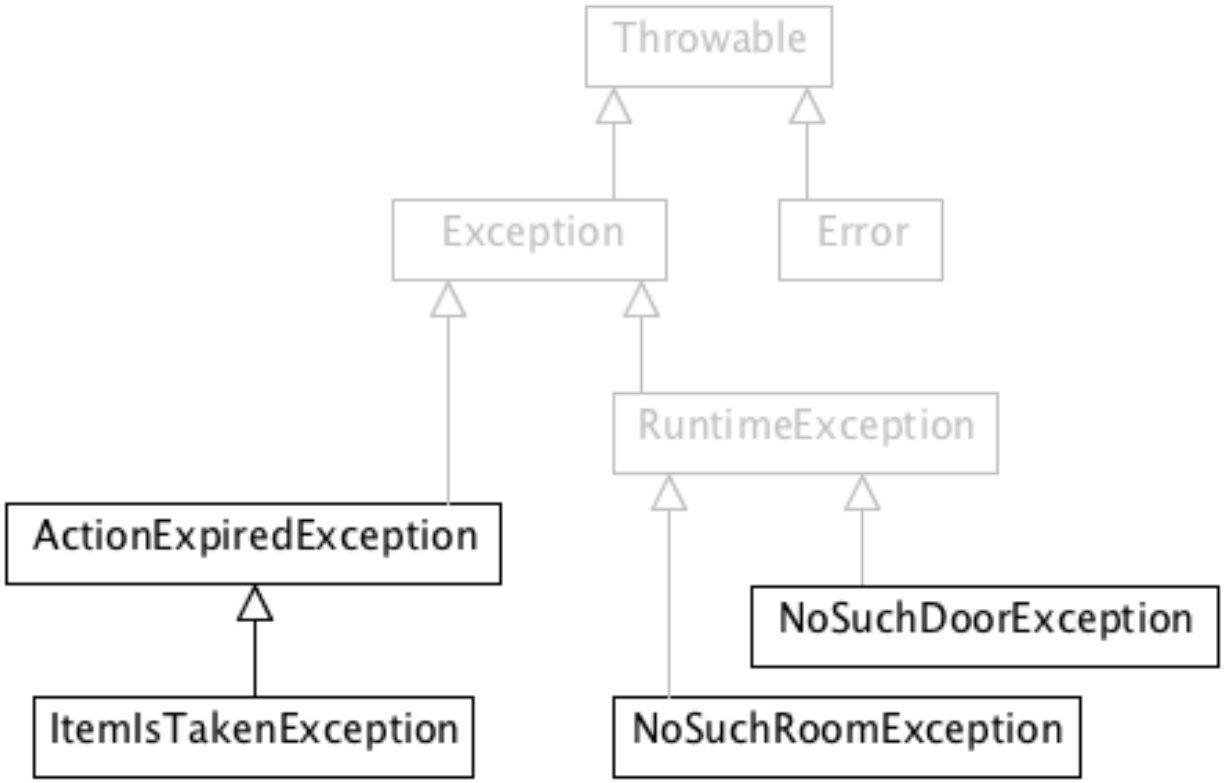
```
public class RoomSummary extends RoomIdentifier {
  private String name;
  private int doorCount;
  private int itemCount;

  public RoomSummary(
      long id, String name,
      int doorCount, int itemCount
      ) {
    super(id);
    this.name = name;
    this.doorCount = doorCount;
    this.itemCount = itemCount;
    }

  public long getName() { return name; }
  public long getDoorCount() { return doorCount; }
  public long getItemCount() { return itemCount; }
  }
```

Dungeon game example

Communication model - Exception Transfer objects



Dungeon game example

Communication model - Exception Transfer objects

cphbusiness
COPENHAGEN BUSINESS ACADEMY

```
public class ActionExpiredException
             extends Exception {

  public ActionExpiredException(String message) {
    super(message);
    }

  }
```

```
public class NoSuchRoomException
             extends RuntimeException {

  public NoSuchRoomException(String message) {
    super(message);
    }

  }
```

Contracts between sub-systems

cphbusiness
COPENHAGEN BUSINESS ACADEMY

- Remote Procedure Calls (RPC)
- CORBA
- SOAP - WSDL
- WADL
- OpenAPI

CORBA

Common Object Request Broker Architecture

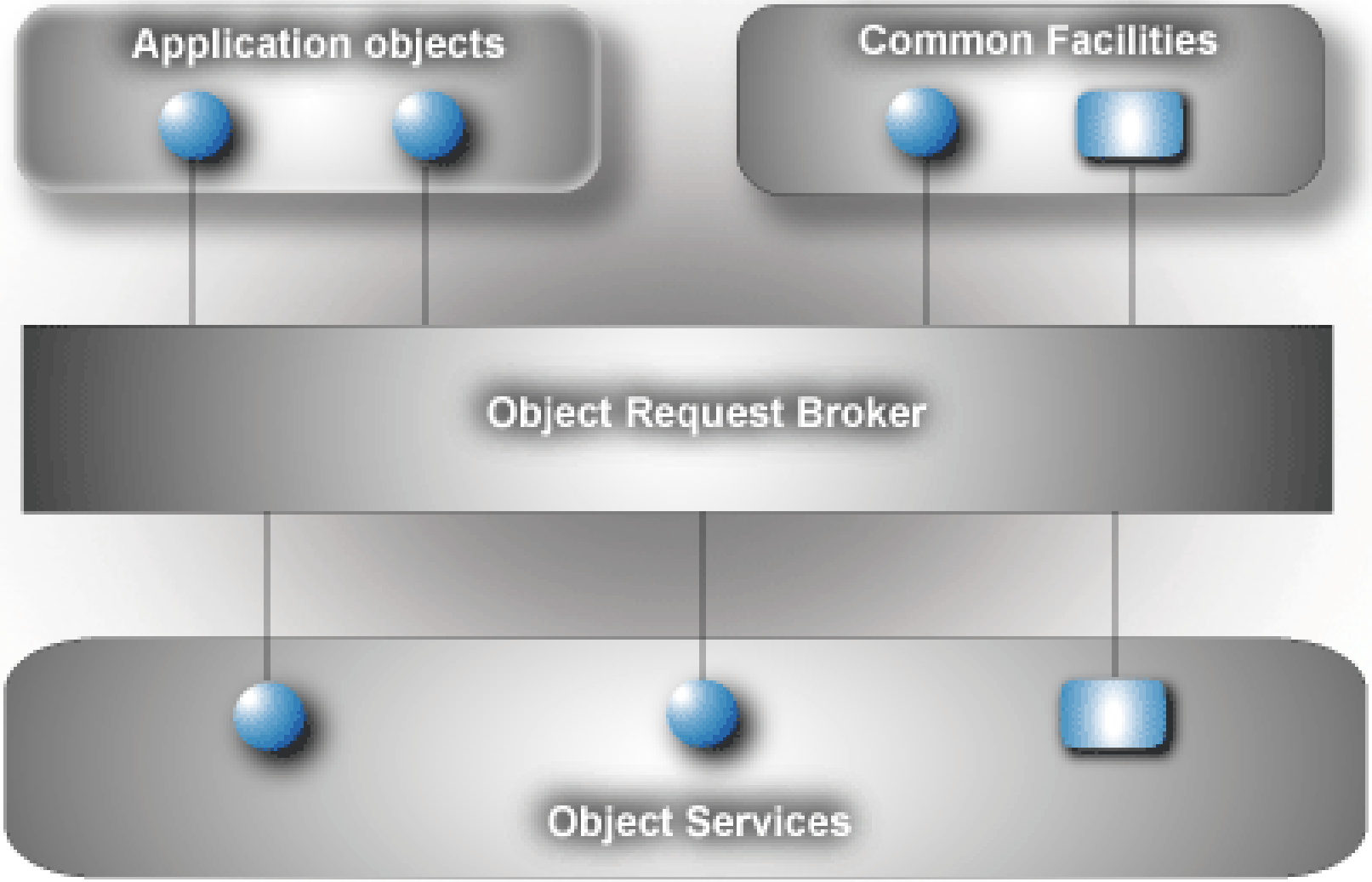
- IDL - Interface Definition Language
- Objects by Reference
- Data by Value

`http://www.ejbtutorial.com/corba/`

CORBA - IDL

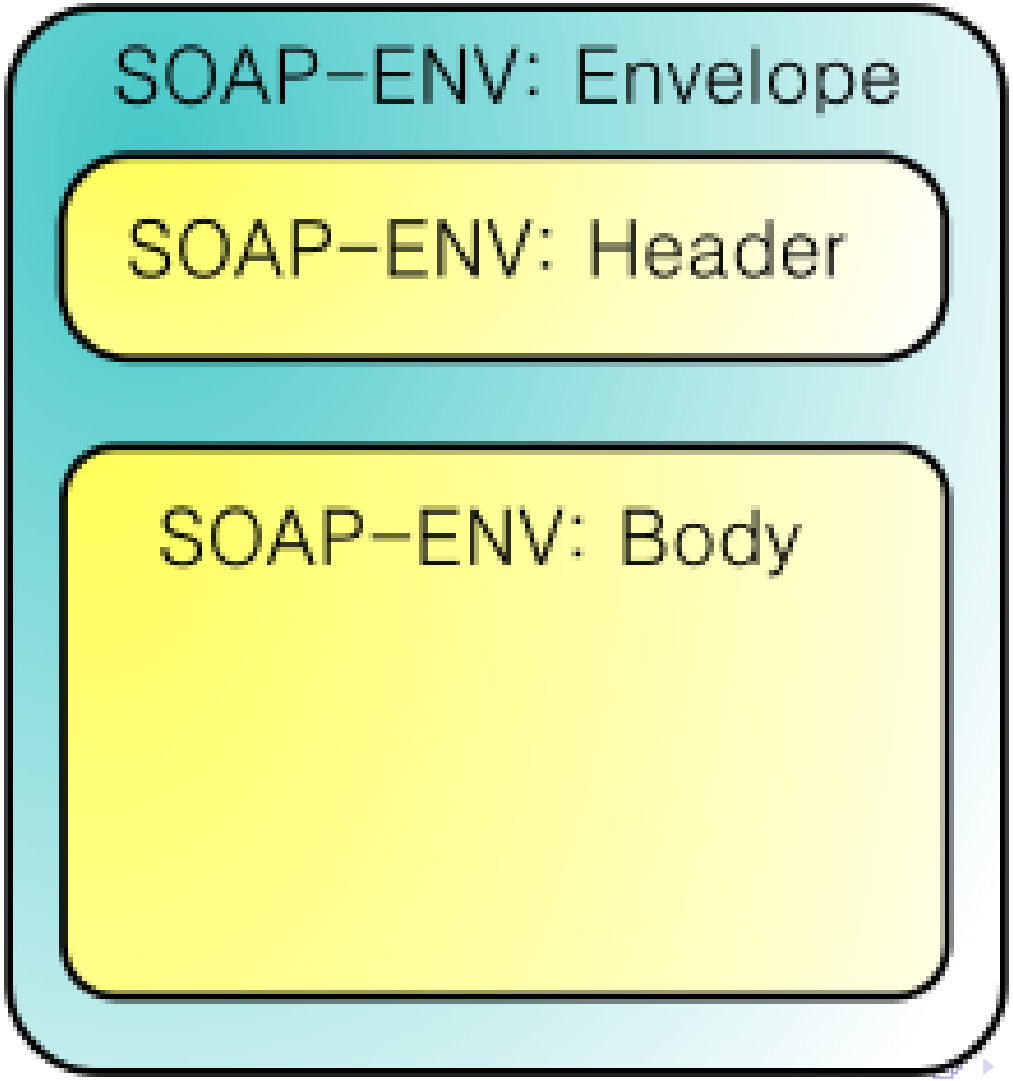

```
module finance{
  interface account {
    // operations
    void makeDeposit(in float amount);
    boolean makeWithdrawal(
        in float amount,
        out float balance
        );
    }
  }
```

CORBA


Application objects
Common Facilities
Object Request Broker
Object Services

SOAP

Simple Object Access Protocol



SOAP example


```
<?xml version="1.0"?>
<soap:Envelope
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:m="http://www.example.org/stock/Manikandan"
    >
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice>
      <m:StockName>GOOGLE</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

WADL

Web Application Description Language

- Driven by Oracle
- Xml-based
- Not very agile

OpenAPI with Swagger

cphbusiness
COPENHAGEN BUSINESS ACADEMY

Swagger `https://swagger.io`

C#

`https://docs.microsoft.com/en-us/aspnet/core/tutorials`

choose "ASP.NET Core Web API Help Pages using Swagger"

Java

`https://dzone.com/articles/swagger-make-developers-love`