omd-systems

SuperKiGaV Jour fixe #1

Technische Architektur

Fachliche Architektur

Vision

omd-systems

Technische Architektur

Warum web?

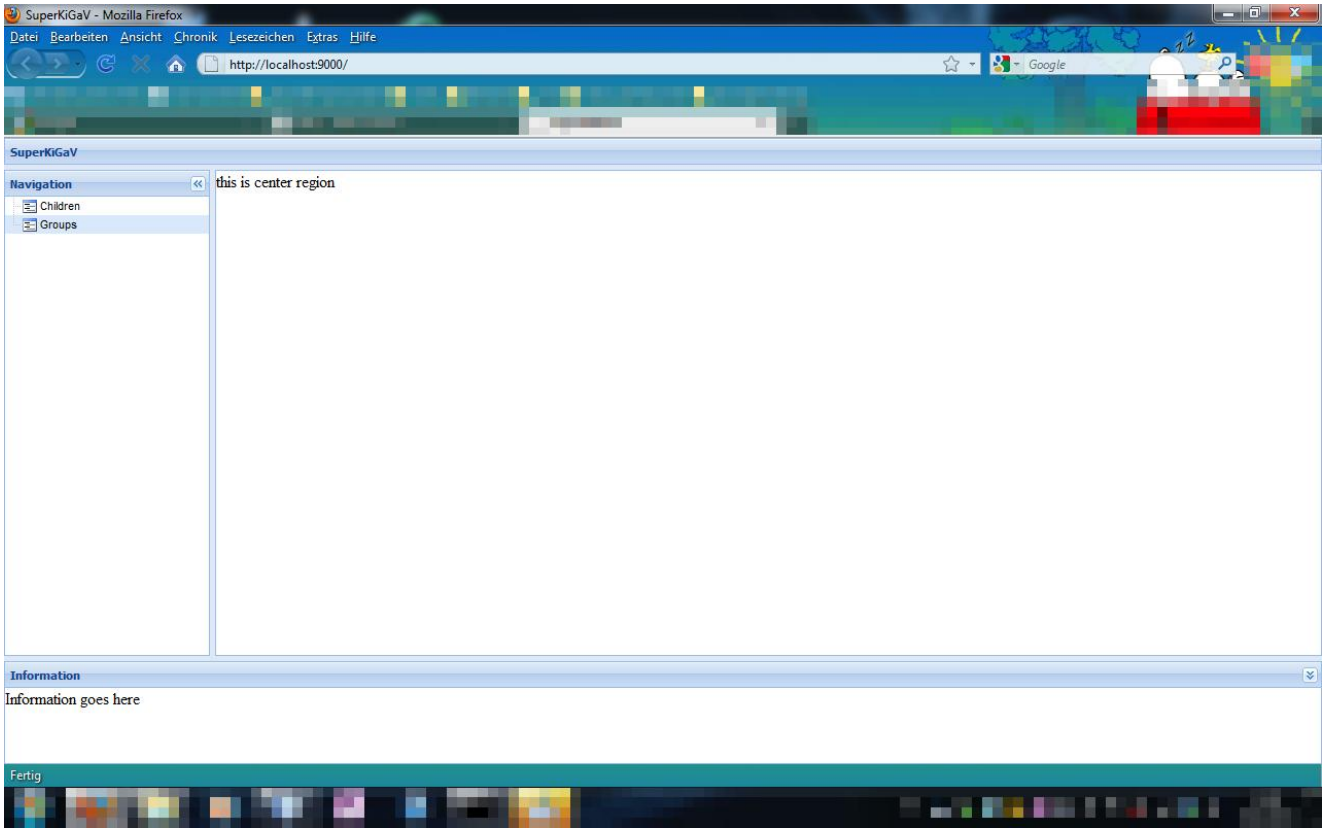
Play!

- MVC-Model
- Routing
- Tests

ExtJs

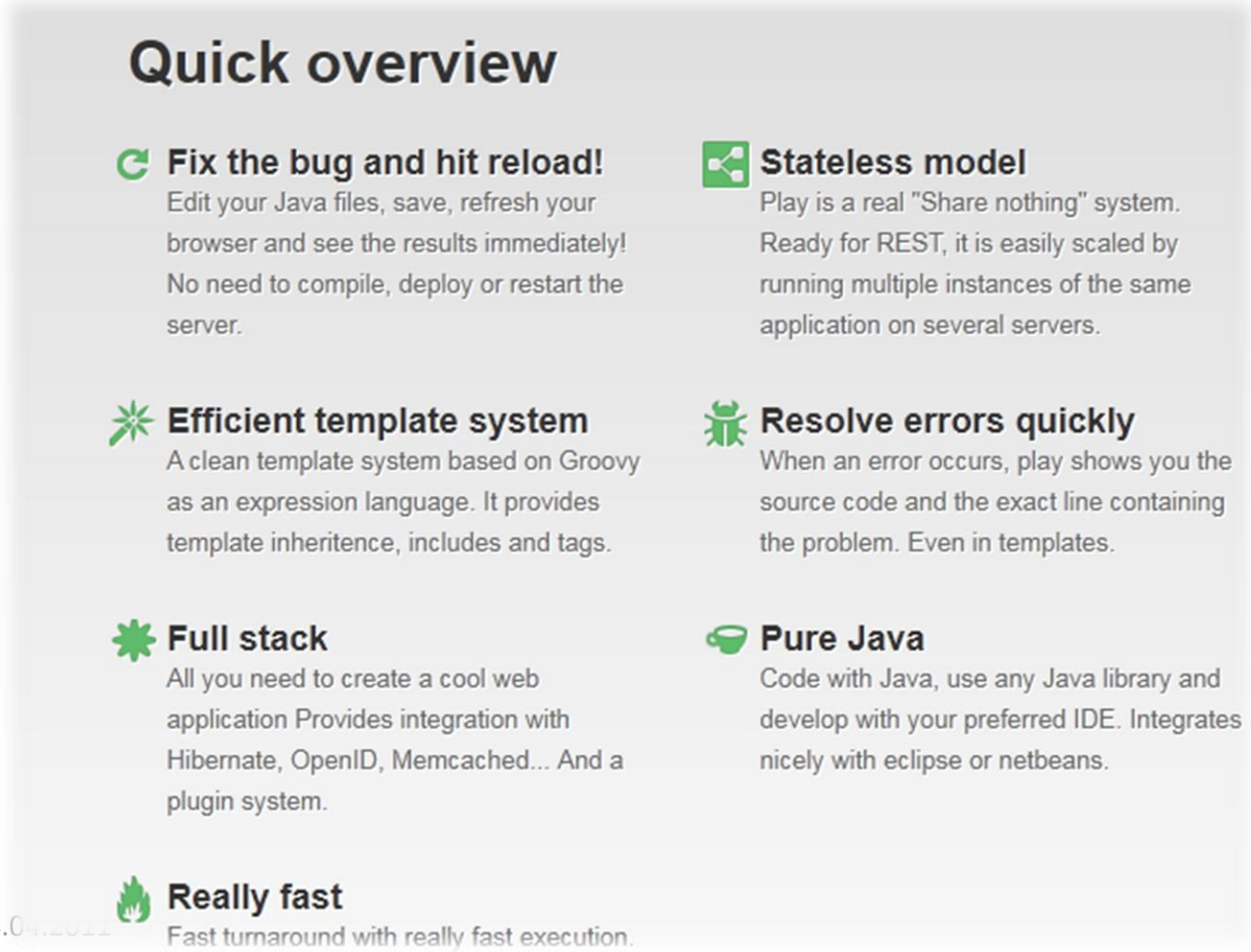
omd-systems

Warum web?



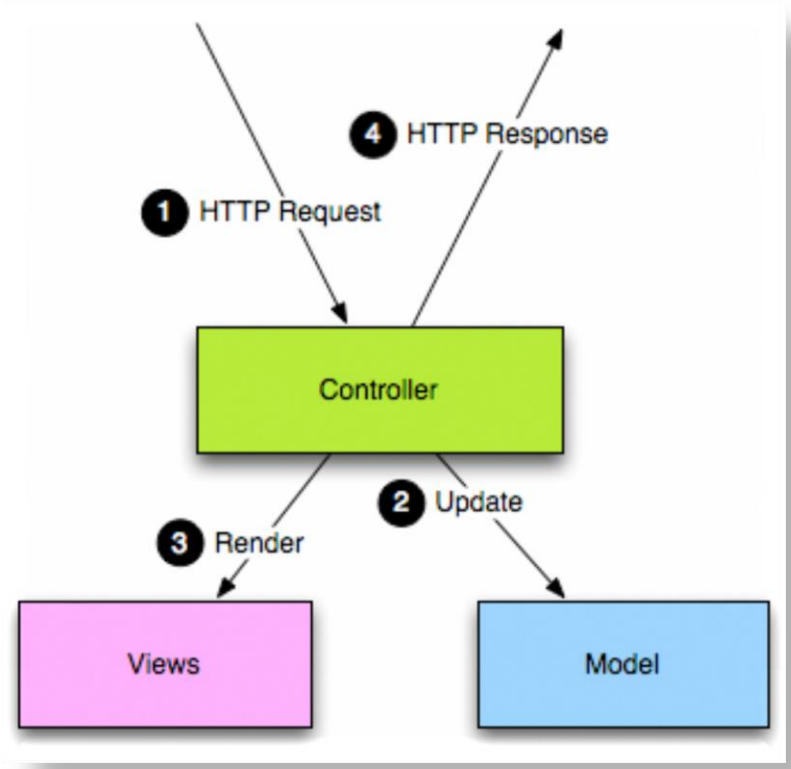
omd-systems

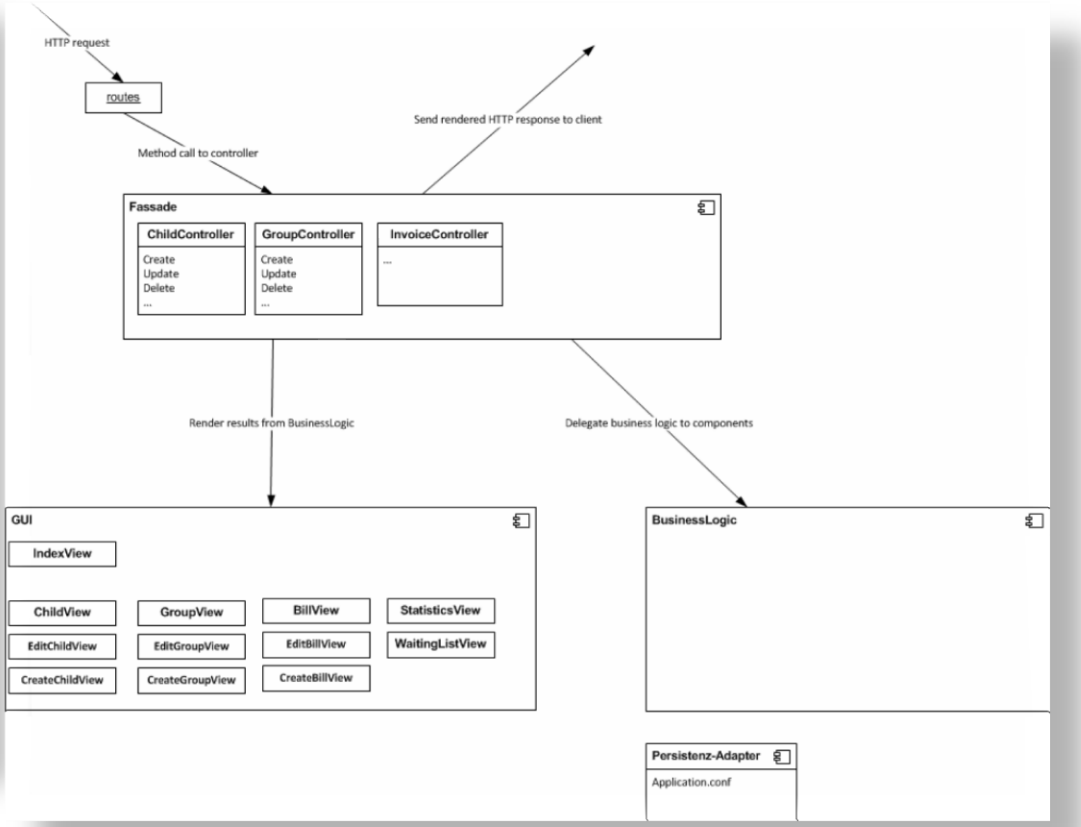
Play! – ein „fullstack“ Java web framework



omd-systems

MVC-Model





omd-systems

MVC-Model

Ein Model:

```
@Entity
class ChildEntity extends Model implements IChildData{
   private String name;
   private String familyName;
   private Date dateOfBirth;
   private String allergies;
   private AdressType adress;
   private List<Long> groups;

public ChildEntity(String name, String familyName, Date
dateOfBirth, String allergies, AdressType adress) {
      this.name = name;
      this.familyName = familyName;
      this.dateOfBirth = dateOfBirth;
      this.allergies = allergies;
      this.adress = adress;
      this.groups = new LinkedList<Long> ();
   }
}
```

Zugriff auf Persistenz

```
boolean deleteChild(long id) {
   if (!existsChild(id)) {
      return false;
   }else{
      ((ChildEntity)ChildEntity.findById(id)).delete();
      return !existsChild(id);
   }
}

private boolean existsChild(long id){
   return ChildEntity.findById(id) != null;
}
```

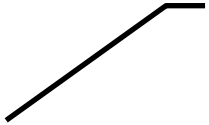
Lesen über static Methoden der Model-Klasse

Löschen, Speichern und Updaten direkt auf Model-Objekt

omd-systems

Kapselung der Persistenz

```
# Database configuration
# ~~~~~
# Enable a database engine if needed.
#
# To quickly set up a development database, use either:
#   - mem : for a transient in memory database (HSQL in
memory)
#   - fs  : for a simple file written database (HSQL file stored)
db=mem
#
# To connect to a local MySQL5 database, use:
# db=mysql:user:pwd@database_name
#
# If you need a full JDBC configuration use the following :
# db.url=jdbc:postgresql:database_name
# db.driver=org.postgresql.Driver
# db.user=root
# db.pass=secret
```



/conf/application.conf

omd-systems

Controller und routing

Router-Klasse

/conf/routes

```
public class Application extends Controller {

   public Application(){
      super();
   }

   public static void index() {
      render();
   }

}
```

```
# Routes
# This file defines all application routes (Higher priority
routes first)
# ~~~~

# Home page
GET     /                           Application.index

# Map static resources from the /app/public folder to the
/public path
GET     /public/                    staticDir:public

# Catch all
*       /{controller}/{action}      {controller}.{action}
```

Views

Javascript: ExtJs

```
var MainView = function() {
  new Ext.Viewport({
    layout: 'border',
    items: [{
      region: 'north',
      html: '<h1 class="x-panel-header">SuperKiGaV</h1>',
      autoHeight: true,
      border: false,
      margins: '0 0 5 0'
    }, {
      region: 'west',
      collapsible: true,
      title: 'Navigation',
      xtype: 'treepanel',
      width: 200,
      autoScroll: true,
....
```

Aufgerufen aus template

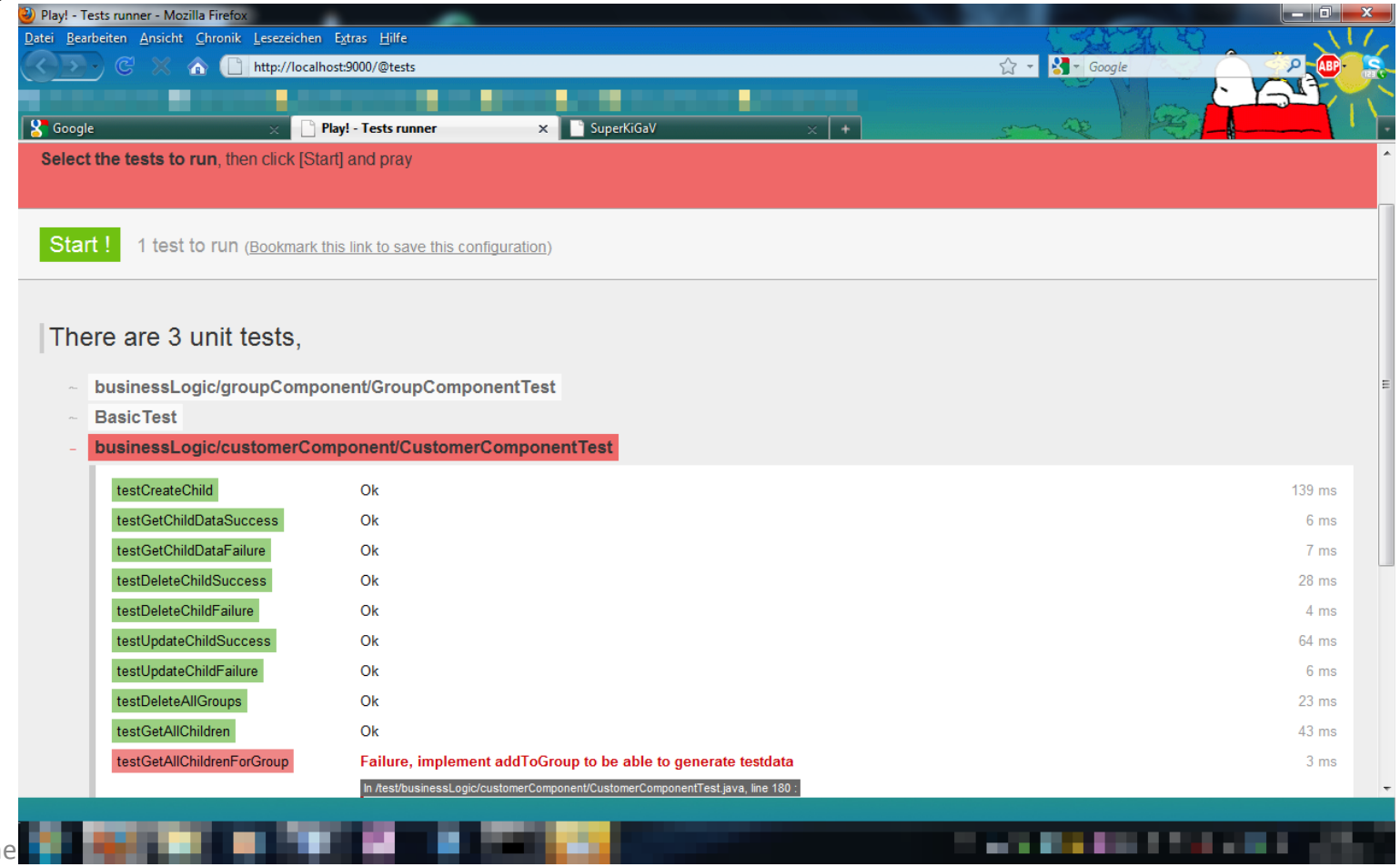
```
<!DOCTYPE html>

<html>
  <head>
    <title>#{get 'title' /}</title>
    <meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
    <link rel="stylesheet" type="text/css"
media="screen"
href="@{'/public/stylesheets/main.css'}">
    #{get 'moreStyles' /}
    <link rel="shortcut icon" type="image/png"
href="@{'/public/images/favicon.png'}">
    <script src="@{'/public/javascripts/jquery-
1.4.2.min.js'}" type="text/javascript" charset="utf-
8"></script>
    #{get 'moreScripts' /}
  </head>
  <body>
    #{doLayout /}
  </body>
</html>
```

public class CustomerComponentTest extends UnitTest {

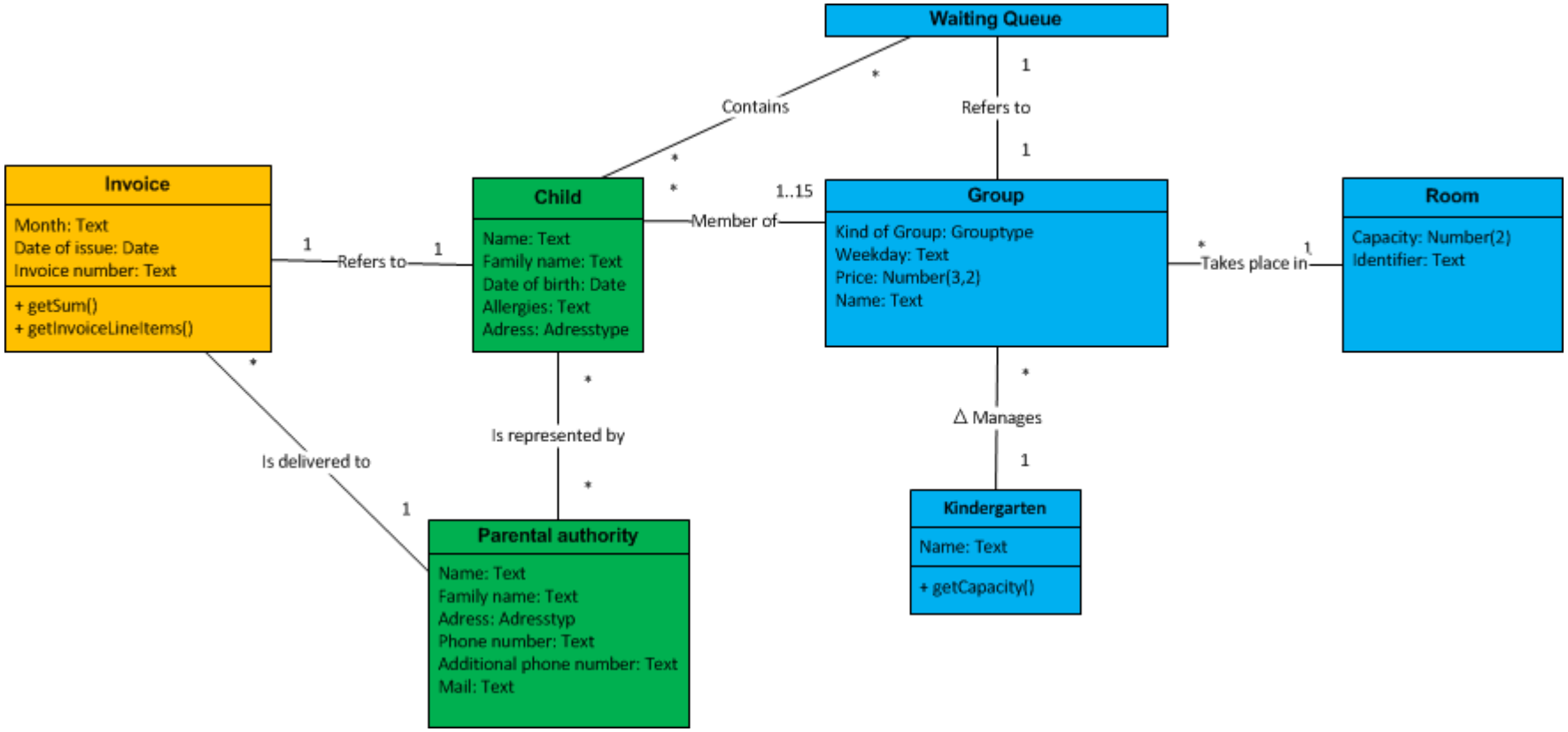
omd-systems

Tests in play!

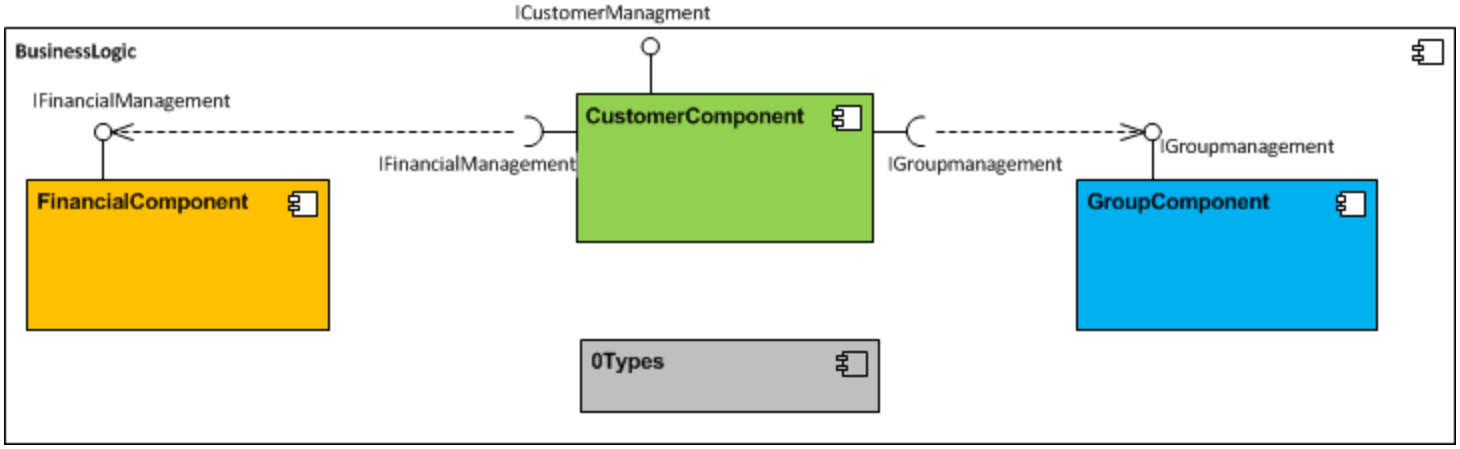


omd-systems

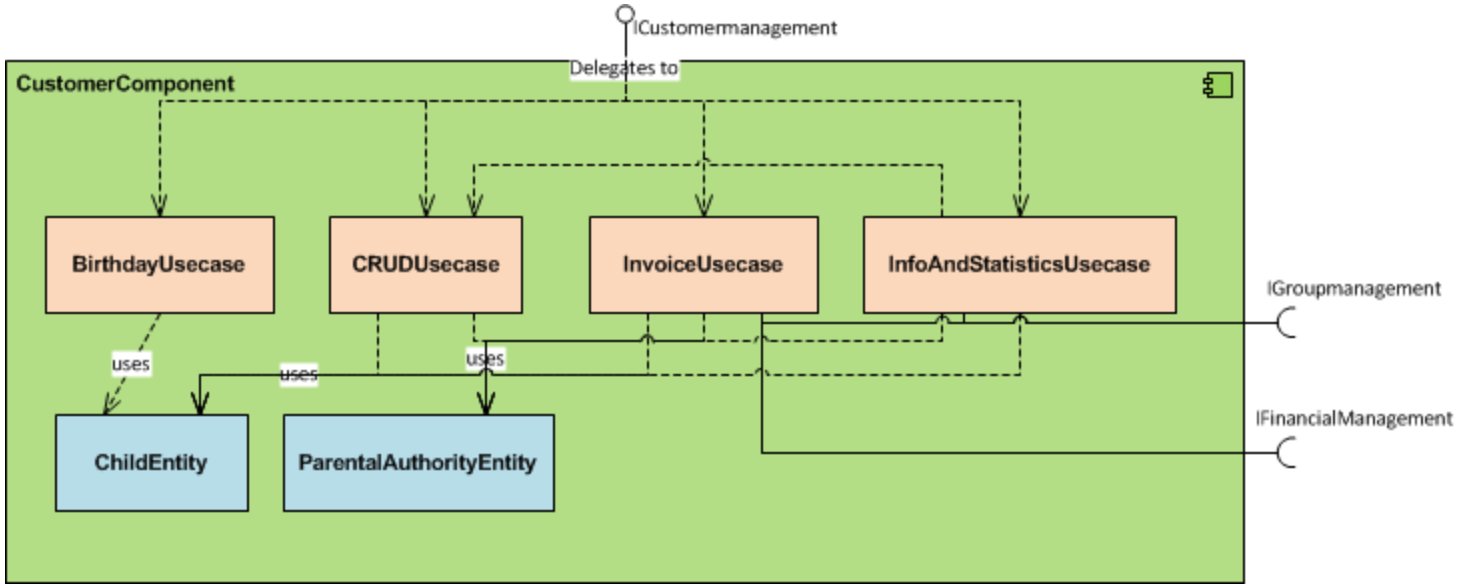
Fachliche Architektur



Fachliche Architektur



Fachliche Architektur



Fachliche Architektur

UseCase „Assign child to group“

Title: Assign child to group
Actor: Kindergartenmanager (below: manager)
Goal: Child can visit group or is in waiting queue
Trigger: Parents wants child to visit a group
Precondition: Child is in system
Postcondition: Child is in group or waiting queue

Success:

1. Manager navigates to form “assign child to groups”
2. System shows form with group overview and current occupancy for each group
3. Manager selects group and clicks on “assign”
4. System checks if slot is free
5. System assigns child to group and saves data to database and shows message to manager: “Child successfully assigned”
6. Continue with 2.

Advanced:

4.
   a. Not enough slots free
      i. System shows message “Assign child to waiting queue instead? Y/N”
      ii. System shows current size of waiting queue
      iii. Yes
         - Systems appends child to waiting queue
      ii. No
         - Continue with 2.

Error:

5.
   a. Database not available
      i. Systems shows message “technical problem. Try again? Y/N”
      ii. Yes
         - Continue with 5.
      iii. No
         - Continue with 2.

omd-systems

Vision

- Intuitive Bedienung
-„Innere Schönheit“
    - Testdriven development – dadurch hohe Abdeckung
- Personalverwaltung mit aufnehmen (viel später)
- Tagesausflüge planen können
- Allergien der Kinder für Küche bereitstellen
- Seriöses Design
- Ajaxafien

Danke für die Aufmerksamkeit

omd-systems

Oliver, Marvin, Dario