

# Übungsbeispiel

Maximilian Irlinger, Tobias Kain

# Verwendete Technologien

- Java
- JavaFX
- Mozart Spaces
- PostgreSQL

# Alternative Implementierung

Gewünschte Features: Transactions, Notifications, Write/Read/Delete, Queries, wahlfreier Zugriff

- MongoDB
  - Transactions
- Oracle DB
  - Konnten wir uns nicht leisten.
- JMS
  - Konzept der Queue selten verwendet
  - Aufgabenstellung eher “datenbankmäßig”
  - Wollten keine 2. uns “unbekannte” Technologie nutzen
- PostgreSQL

# Koordinationsmodelle

- AnyCoordinator
- QueryCoordinator
- FifoCoordnator
- LabelCoordinator

# Architektur (1)

- Modul **Shared**

- Abstrakte Klassen für Roboter (außer Monitoring) und Services
- Abstrakte GUI Klassen
- Datenklassen

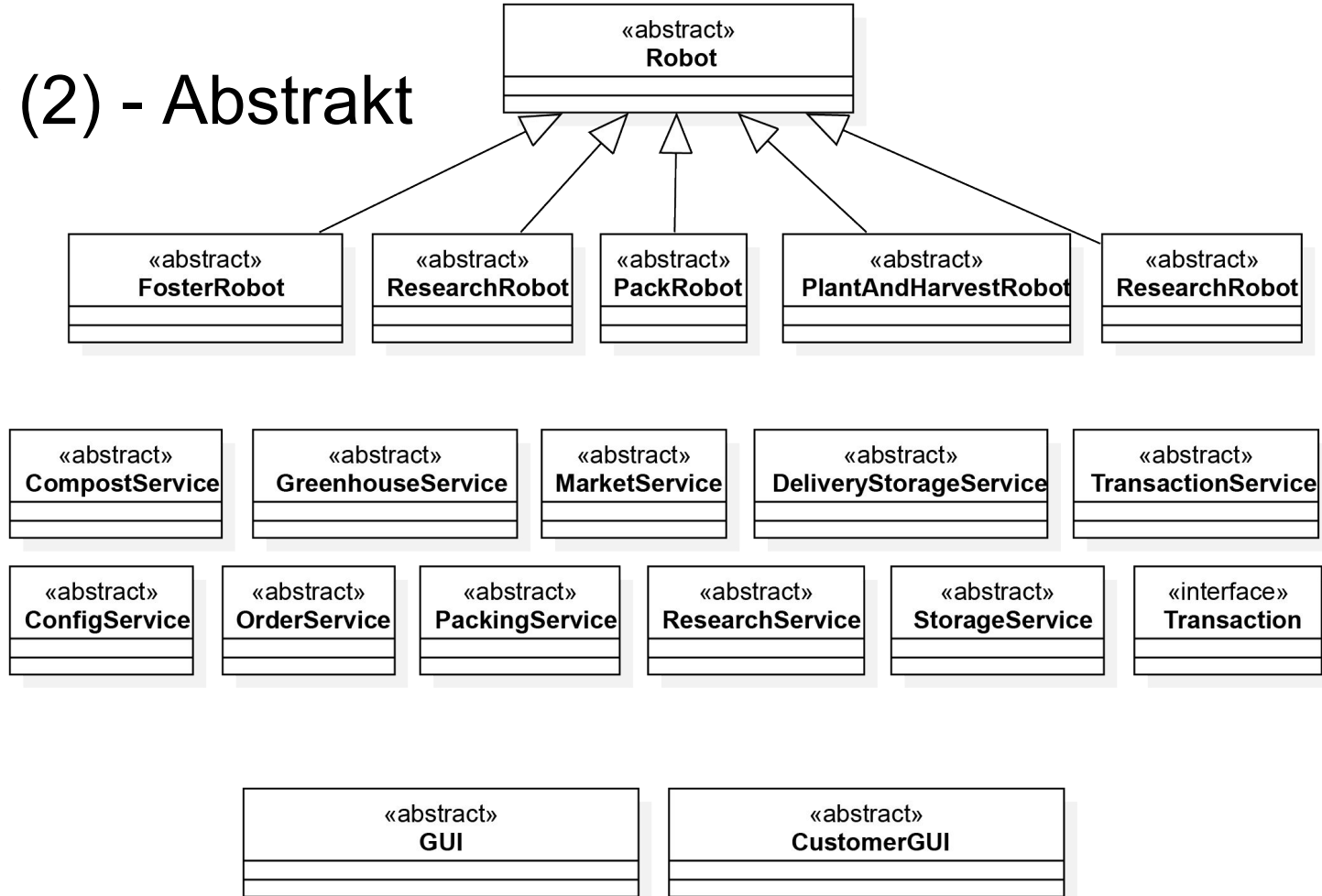
- Modul **Space Implementation**

- Jeder Roboter + GUI eine Main Methode
- Space Server für Wasser Aspekt

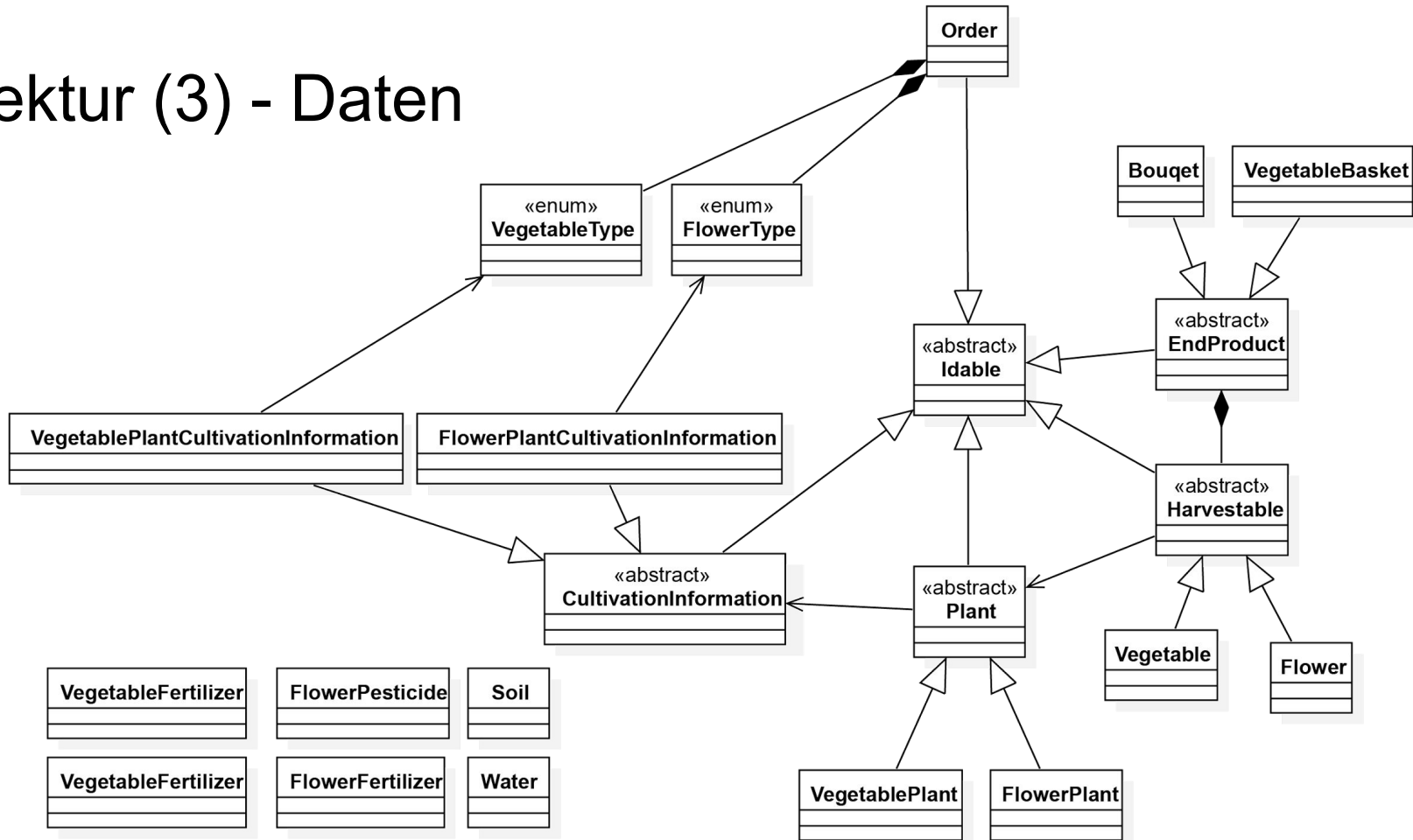
- Modul **Postgres Implementation**

- Jeder Roboter + GUI eine Main Methode
- Monitoring Roboter als große Query

# Architektur (2) - Abstrakt



# Architektur (3) - Daten



# Aufwand

- Aufwand in Stunden
  - Teil 1: 80 Stunden pro Person
  - Teil 2: 35 Stunden pro Person
  - Insgesamt: 115 Stunden pro Person
- Aufwand in LOC
  - Space spezifisch: 2500 LOC
  - Postgres spezifisch: 2700 LOC
  - Insgesamt: 11000 LOC



# Schwierigkeiten bei Task 1

- Bounded Containers in MozartSpaces
- Postgres 3rd Party Alpha Treiber
- Multithreading concurrency
  - synchronized

# Schwierigkeiten bei der Erweiterung

- keine

Danke für Ihre Aufmerksamkeit

Maximilian Irlinger, Tobias Kain