Programmierpraktikum C und C++

Einführung

TECHNISCHE UNIVERSITÄT DARMSTADT

Anthony Anjorin

anthony.anjorin@es.tu-darmstadt.de

ES Real-Time Systems Lab

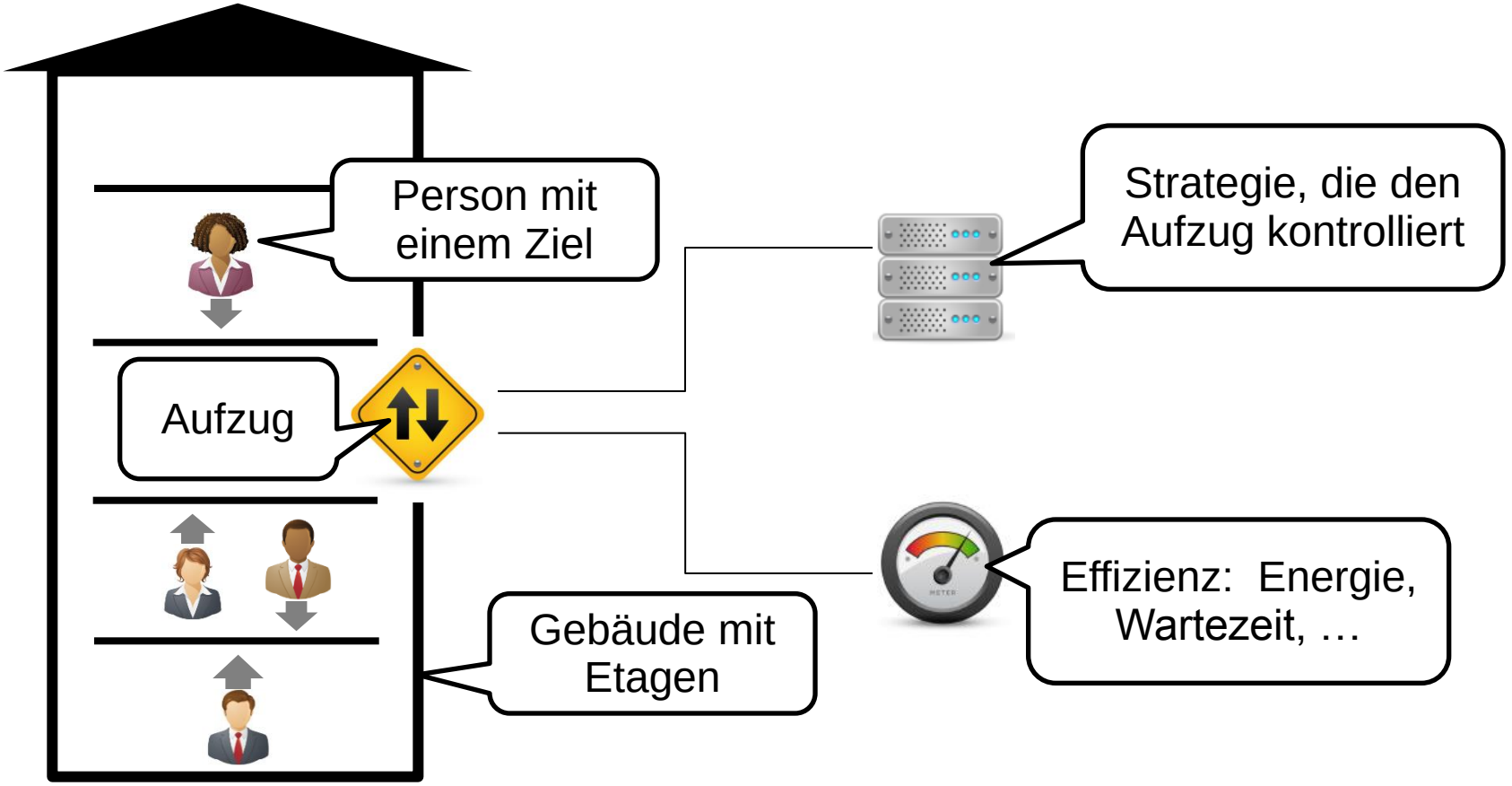
Prof. Dr. rer. nat. Andy Schürr

Dept. of Electrical Engineering and Information Technology

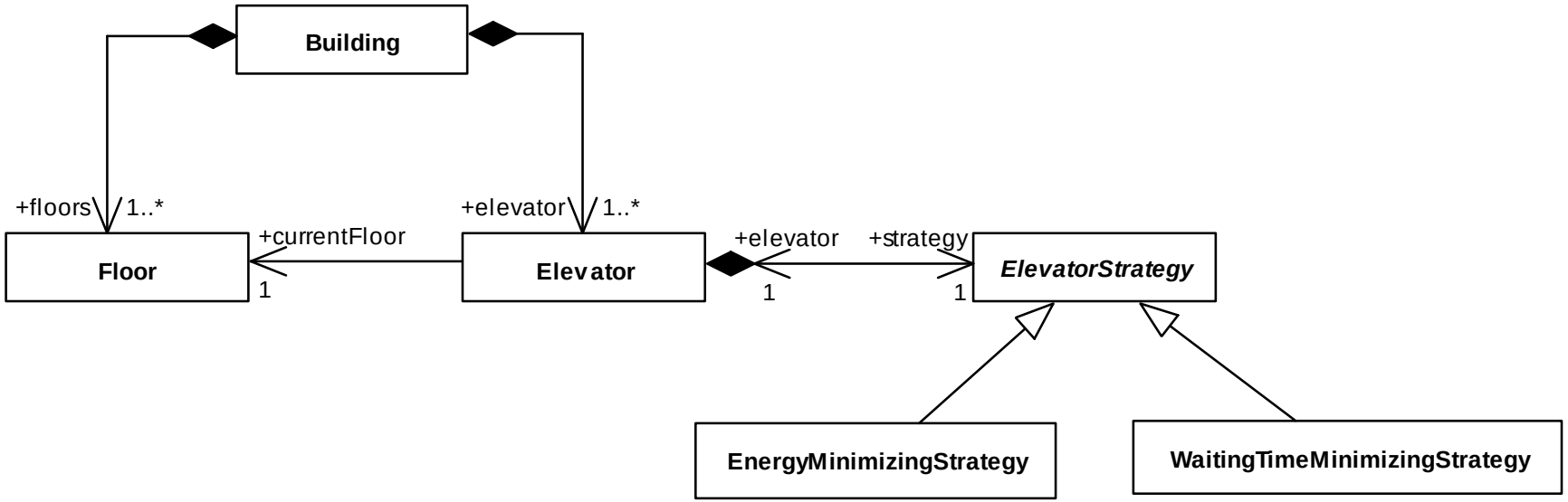
Dept. of Computer Science (adjunct Professor)

www.es.tu-darmstadt.de

Implementierung einer Aufzugsimulation



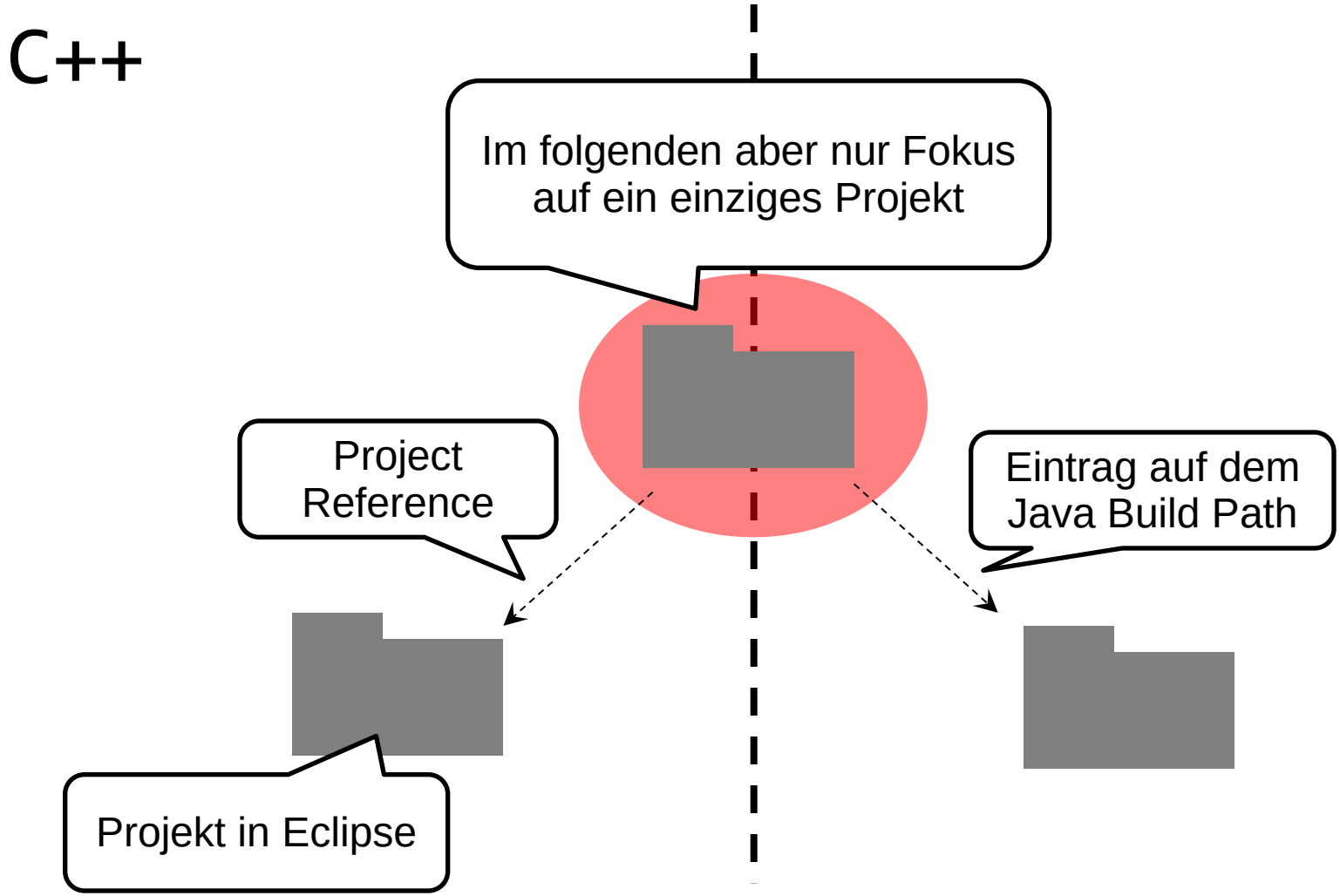
Statische Struktur des Systems (Klassendiagramm / Metamodell)



Implementierung mit C++ / Vergleich mit Java

1.Projektabhängigkeiten

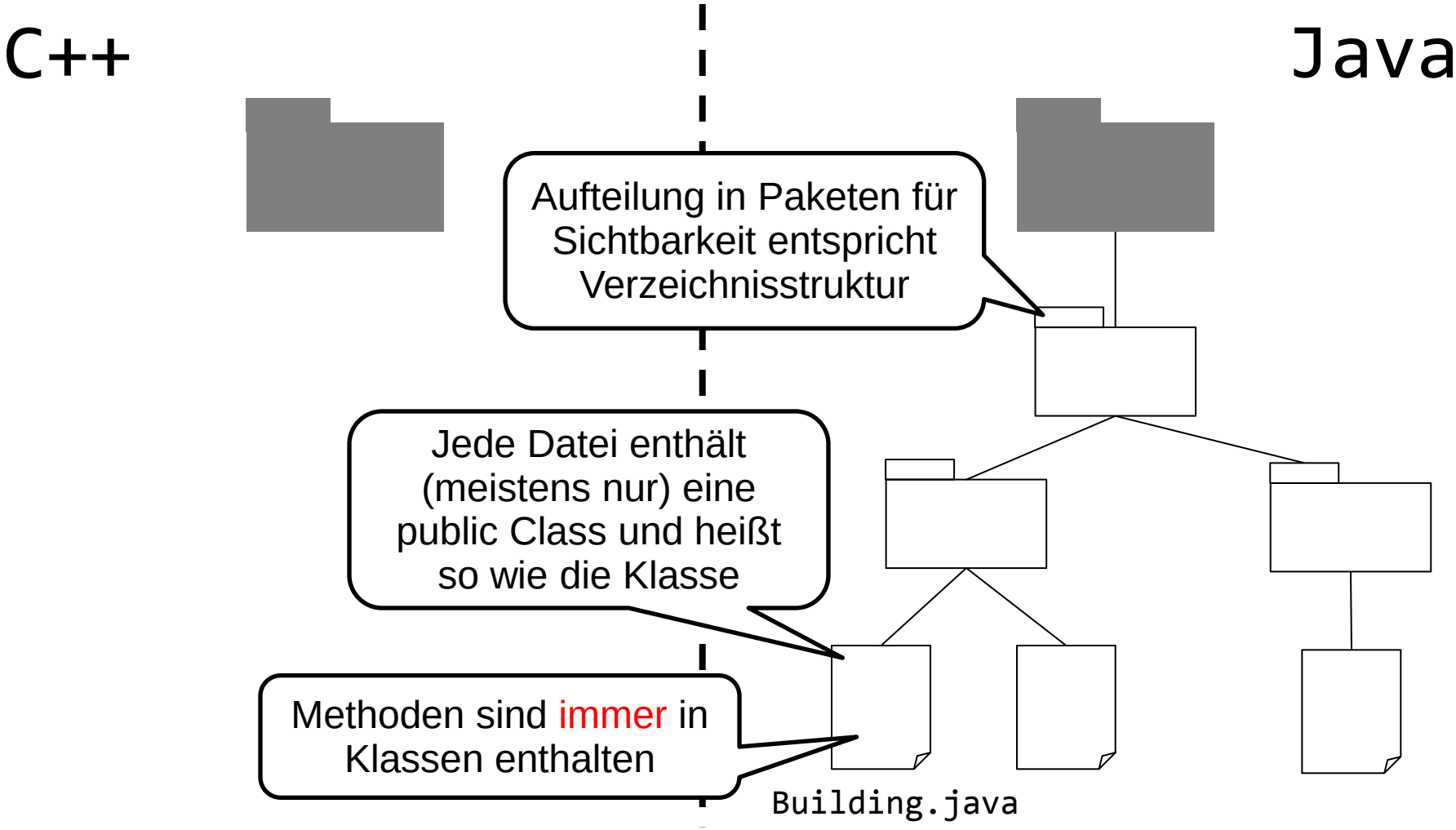
Projektabhängigkeiten (mit Eclipse CDT)



Implementierung mit C++ / Vergleich mit Java

2. Projektstruktur

Projektstruktur



Intermezzo

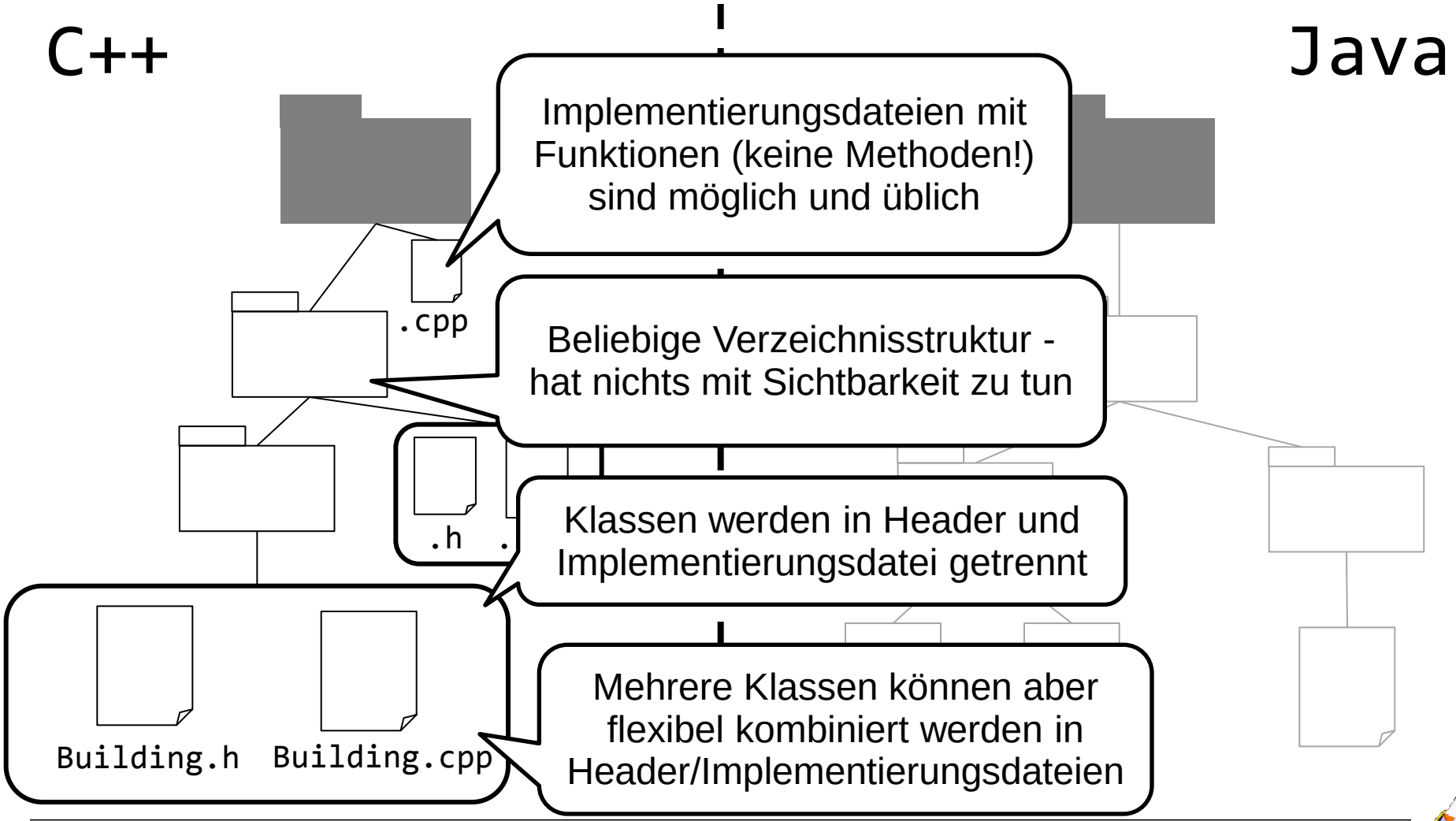
TECHNISCHE UNIVERSITÄT DARMSTADT

Ist es sinnvoll zu verlangen, dass jede „Funktion“ in einer Klasse sein MUSS?

Ist es sinnvoll die Paketstruktur an der Verzeichnisstruktur zu binden?

Darf man in Java mehrere Klassen in einer Datei implementieren?

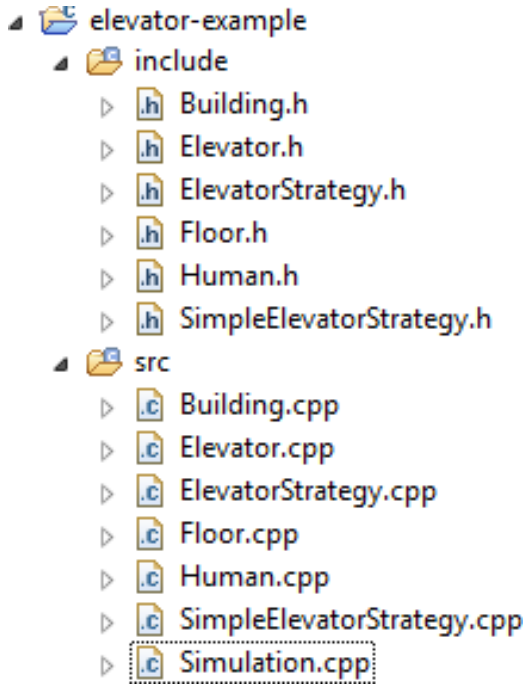
Projektstruktur



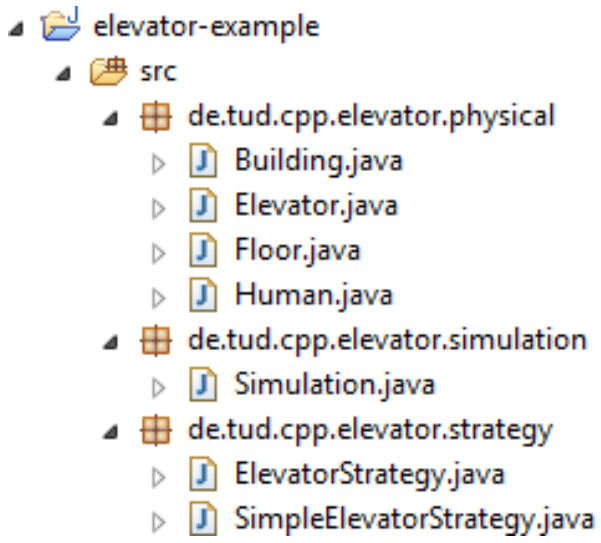
Projektstruktur

TECHNISCHE UNIVERSITÄT DARMSTADT

C++



Java



Header und Impl-Dateien

TECHNISCHE UNIVERSITÄT DARMSTADT

```
/*
 *
 * Part of the elevator si
 * A Building is a container for
 * Floors and the Elevator
 */

#ifndef BUILDING_H_
#define BUILDING_H_

#include <vector>

#include "Floor.h"
#include "Elevator.h"


class Building {
public:
  Building(int numberOfFloors);
  ~Building();

  void runSimulation();

private:
  std::vector<Floor> floors;
  Elevator elevator;
};

#endif /* BUILDING_H_ */
```

Kommentare wie in Java

Include-Anweisungen wie Import-Befehle in Java.

< > wird für Bibliotheken verwendet (kein eigener Code)

Deklaration der Klasse ist wie ein Interface in Java (Details kommen später)

Header und Impl-Dateien

TECHNISCHE UNIVERSITÄT DARMSTADT

```
#include <iostream>
using std::cout;
using std::endl;
```

Using-Befehle sind wie statische Imports in Java (die namespaces können ohne vollständige Angabe verwendet werden)

```
#include "Building.h"
```

Header-Datei wird eingebunden

```
Building::Building(int numberOfFloors) :
  floors(numberOfFloors, Floor()) {
    cout << "Creating building with "
         << numberOfFloors << " floors."
         << endl;
}

Building::~Building() {
  cout << "Destroying building." << endl;
}

void Building::runSimulation() {
  cout << "Simulation running ..." << endl;
  // ...
}
```

Methoden werden implementiert (Details später)

Intermezzo

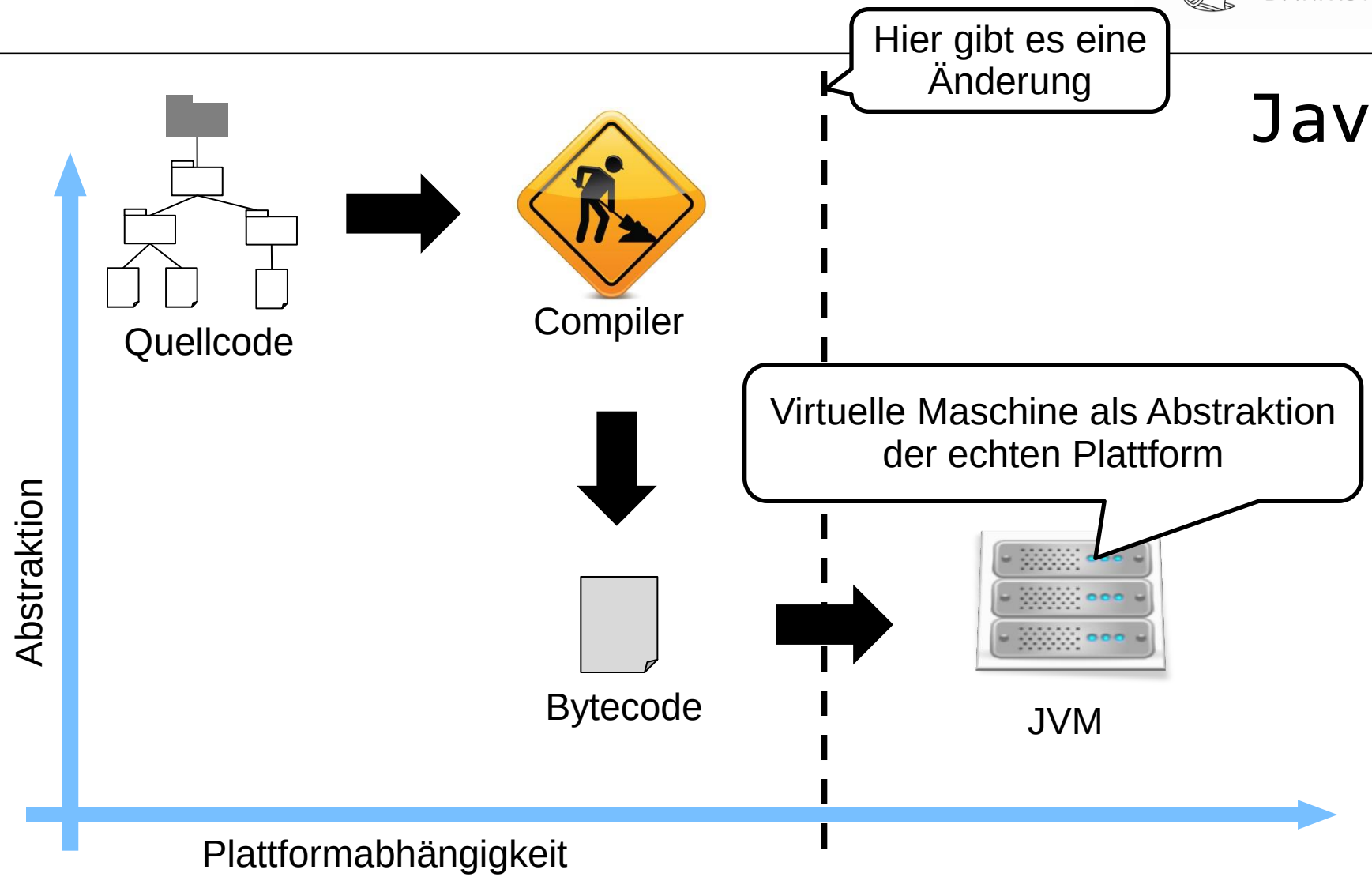
TECHNISCHE UNIVERSITÄT DARMSTADT

Ist die Trennung in Header- und Impl-Dateien wirklich hilfreich? Oder nur nervig…

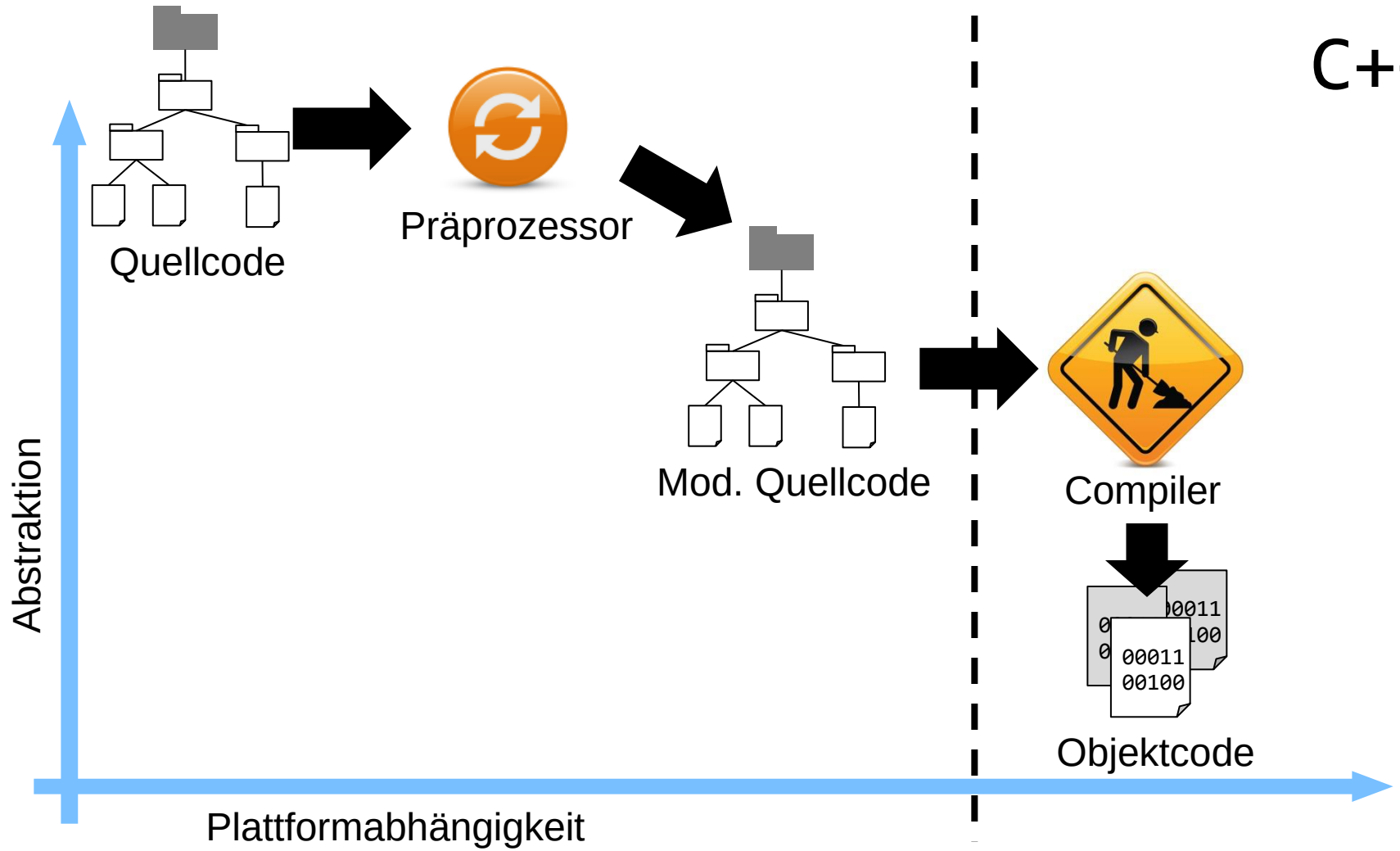
Implementierung mit C++ / Vergleich mit Java

3. Kompilierung

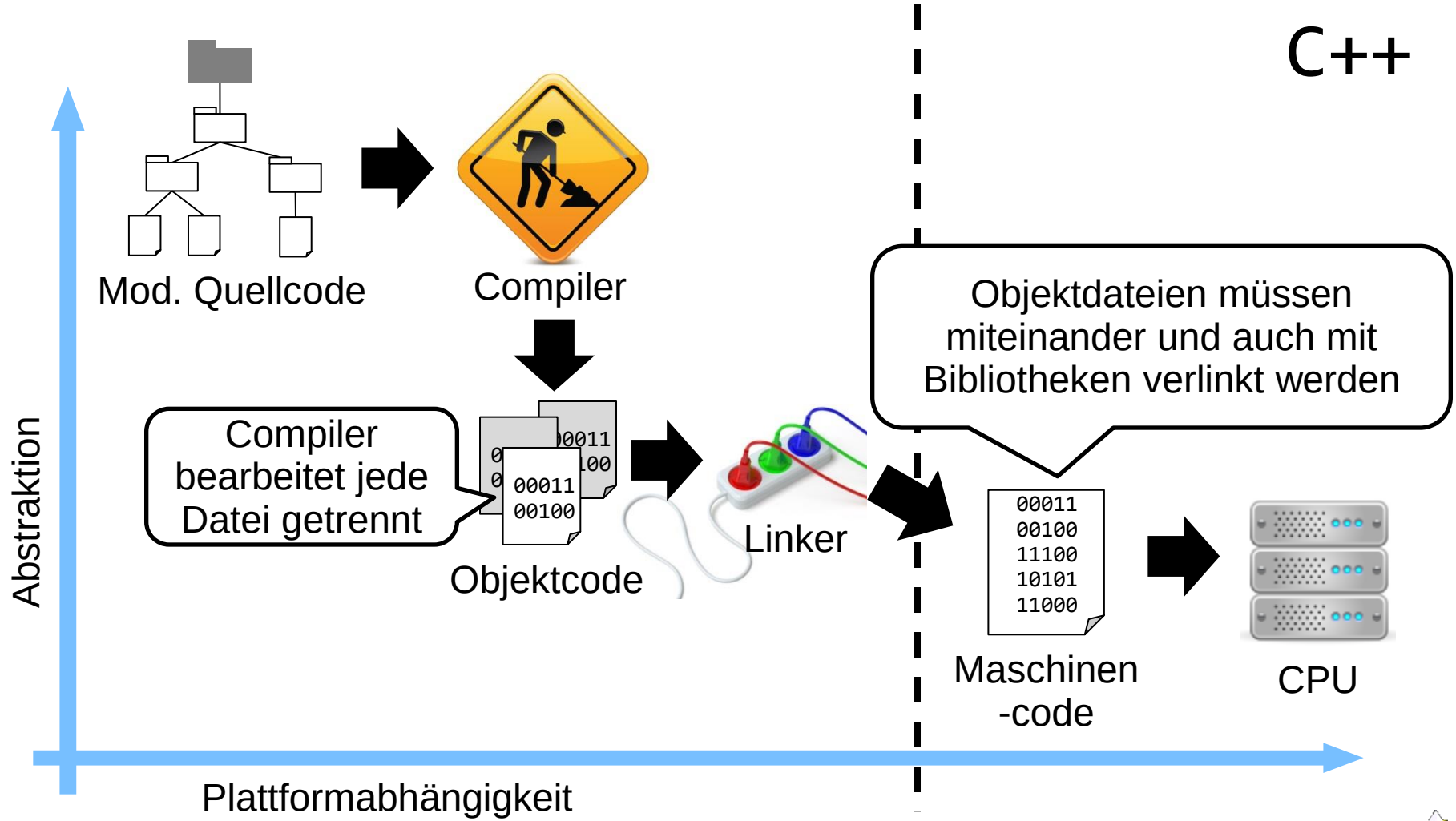
Kompilierung



Kompilierung



Kompilierung



Was genau macht der Präprozessor?

TECHNISCHE UNIVERSITÄT DARMSTADT

```
#ifndef BUILDING_H_
#define BUILDING_H_

#include <vector>

#include "Floor.h"
#include "Elevator.h"


class Building {
public:
  Building(int numberOfFloors);
  ~Building();

  void runSimulation();

private:
  std::vector<Floor> floors;
  Elevator elevator;
};

#endif /* BUILDING_H_ */
```

Schützt davor, dass Building.h mehrmals eingebunden wird

Diese Konvention macht es möglich, ohne Bedenken immer alle benötigten Header überall einbinden zu können

Der Präprozessor kann viel mehr, aber seine Verwendung für C++-Programme (über das gezeigte hinaus) ist weder notwendig noch zu empfehlen

Intermezzo

TECHNISCHE UNIVERSITÄT DARMSTADT

Stimmt es wirklich, dass Java „plattformunabhängig“ ist und C++ nicht?

Ist es möglich, dass man erfolgreich kompilieren aber nicht linken kann? Wie?

Ist der Präprozessor wirklich „böse“? Wieso? Ist dies bei allen Sprachen der Fall?

Implementierung mit C++ / Vergleich mit Java

4. Systemstart

Systemstart

TECHNISCHE UNIVERSITÄT DARMSTADT

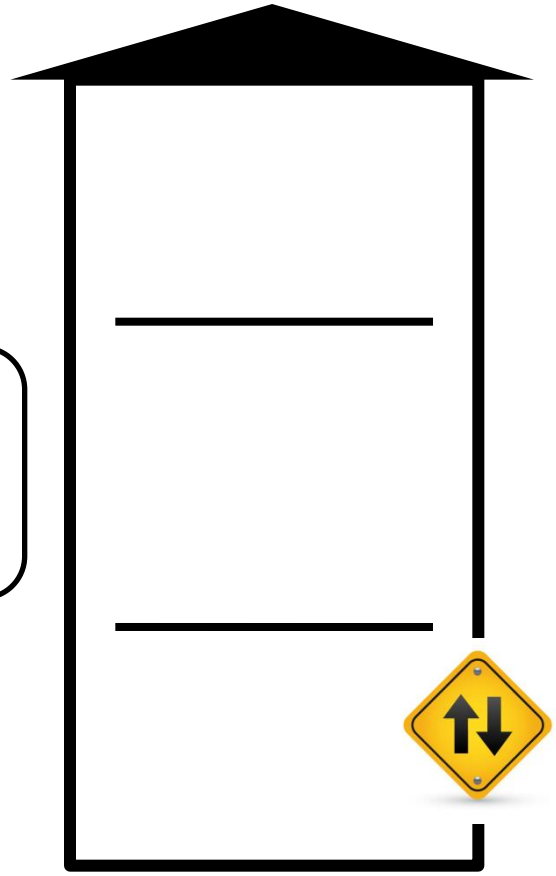
```
//====================================
// Name: elevator-example-lecture.cpp
//====================================

#include "Building.h"

int main() {
  Building building(3);
  building.runSimulation(
}
```

Main-Funktion entspricht Main-Methode in Java (Argumente auch möglich aber nicht nötig)

Kein Rückgabewert nötig (implizit 0 für „alles ordnungsgemäß durchgelaufen“), zumindest bei gcc



Intermezzo

TECHNISCHE UNIVERSITÄT DARMSTADT

Java vs. C++: Stärken und Schwächen?