Konstantin Bork
Fachbereich Mathematik und Informatik

Freie Universität Berlin

Entwurf und Implementierung einer Architektur zur asynchronen Verarbeitung von Batch-Jobs in einem Unternehmen

Agenda

- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - Theorie
  - Architekturentwurf
- Evaluation
- Implementierung
- Fazit & Ausblick

Freie Universität Berlin

Problembeschreibung

- Geringe Produktivität des selbstentwickelten CMS
  - Lange Verarbeitung hochgeladener Dateien
  - Datenexporte benötigen viel Zeit
  - Timeouts
- Benutzer erhalten keine Rückmeldung vom System

Freie Universität Berlin

Agenda

- Problembeschreibung
- **Anforderungen**
- Anforderungsanalyse
  - Theorie
  - Architekturentwurf
- Evaluation
- Implementierung
- Fazit & Ausblick

Anforderungen

- Asynchrone Verarbeitung
- Priorisierung von Aufträgen
- Verteiltes System offen halten
- Nichtfunktional:
  - Modularer Aufbau
  - Wartungsfreundlich
  - Gut dokumentiert
  - Fängt Fehler ab

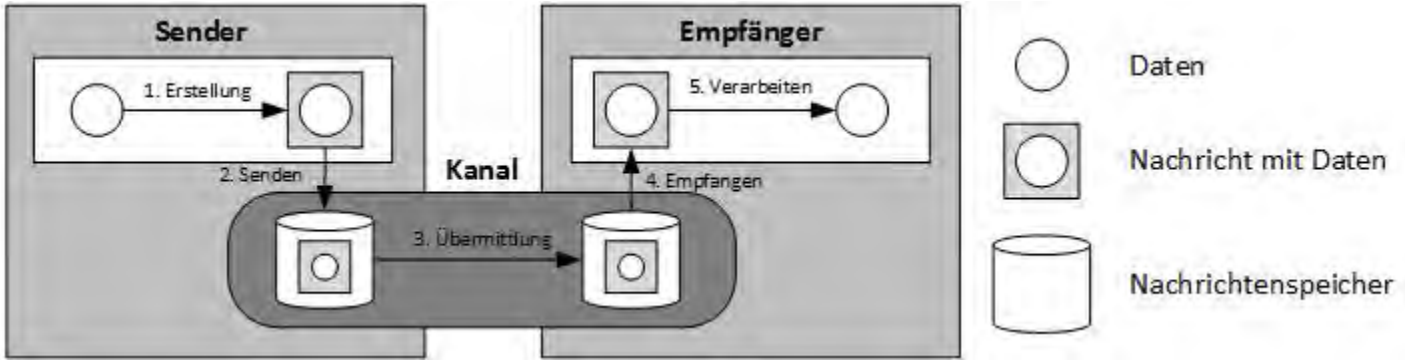
Freie Universität Berlin

Agenda

- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - **Theorie**
    - Architekturentwurf
- Evaluation
- Implementierung
- Fazit & Ausblick

Theorie: Messaging

- Schnelle asynchrone Kommunikation mit verlässlicher Zustellung



- Vorteile:
  - Verbindung separater Programme
  - Verlässliche Kommunikation
  - Nachrichten sind atomare Einheiten

Theorie: Middleware

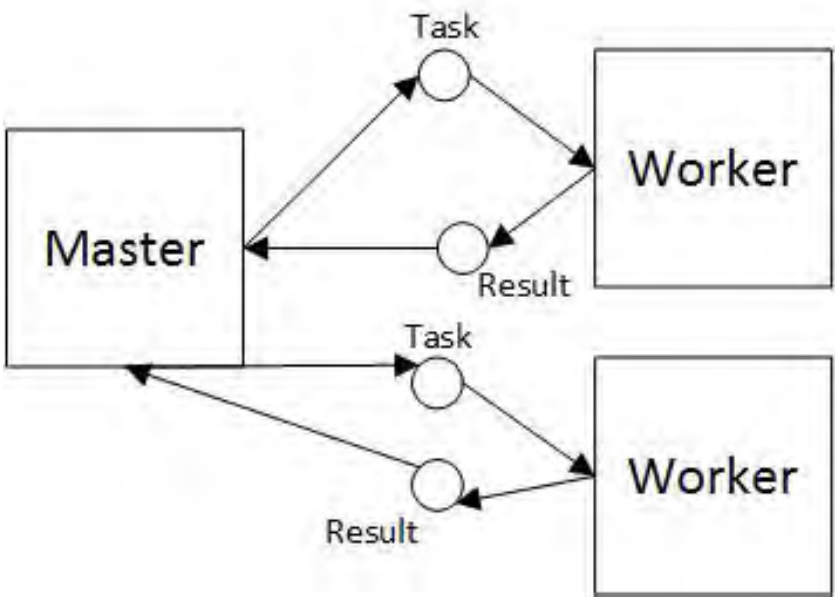
- Zusätzliche Schicht in Softwareumgebungen
- Ziel: einfache Zugriffe auf untere Schichten, höhere Produktivität
- Zwei Formen: Kommunikationsorientiert & Anwendungsorientiert

Theorie: Batch-Processing

- Jobs werden als Ganzes abgearbeitet
- Eigenschaften:
  - Große Mengen an Eingabedaten
  - Abarbeitung in *Batch Window*
  - Abarbeitung mehrerer Hundert bis Tausend Jobs
- Beispiel: Abbuchungen von Bankkonten

Theorie: Architekturmuster

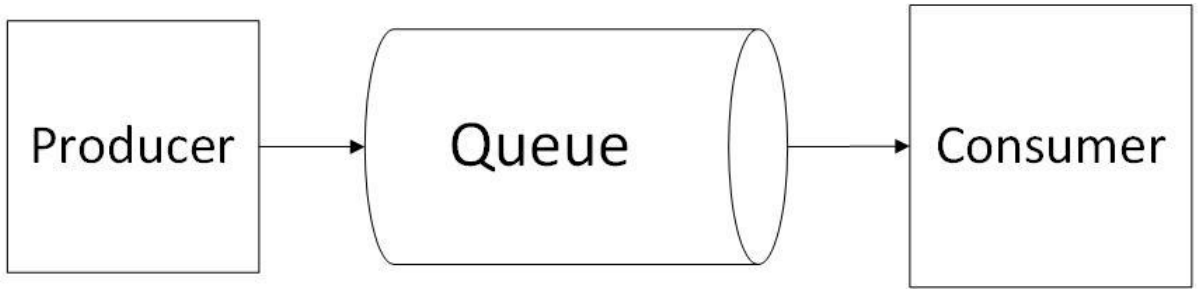
- Master-Worker



Freie Universität Berlin

Theorie: Architekturmuster

- Producer-Consumer



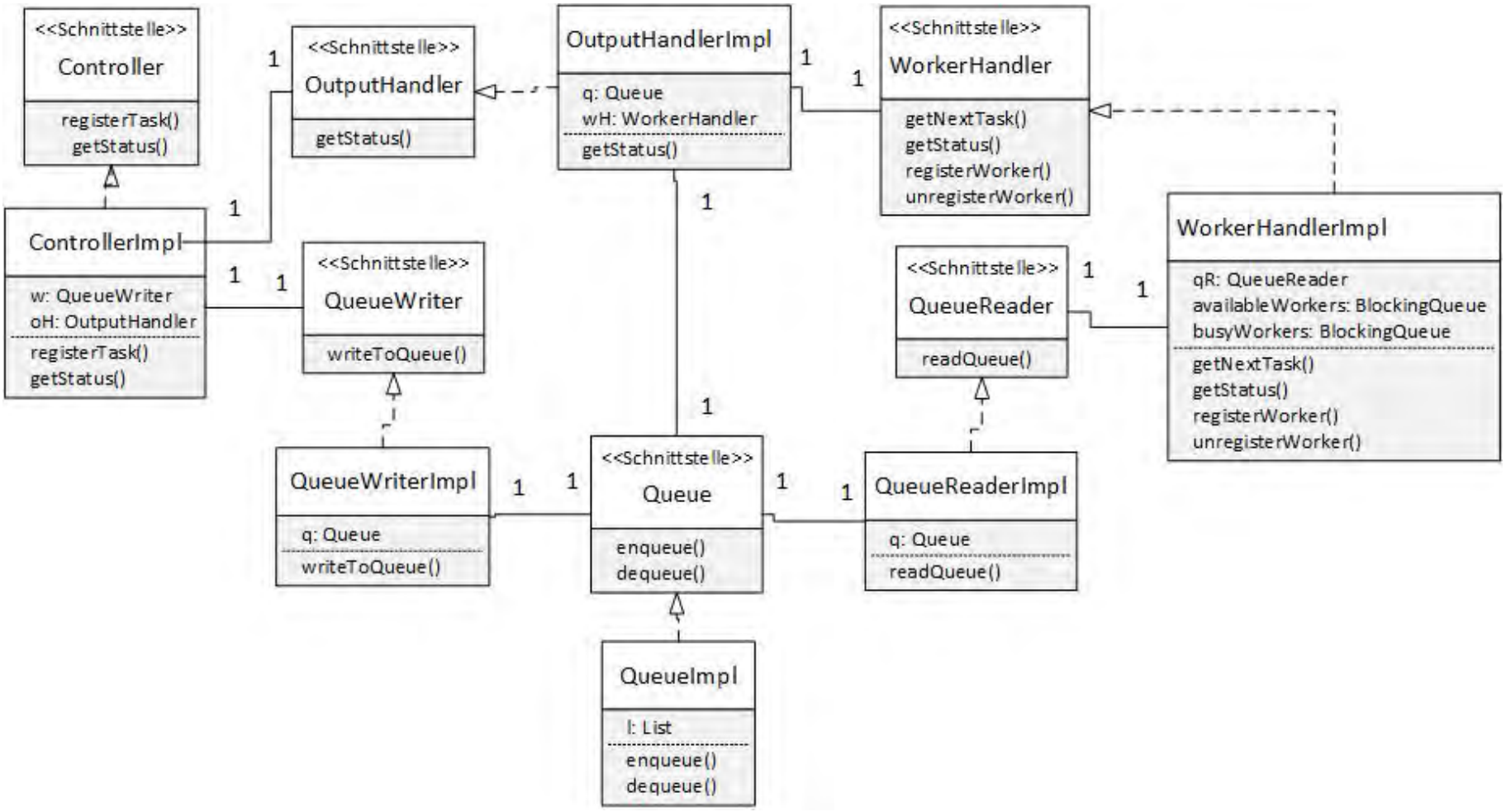
Freie Universität Berlin

Agenda

- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - Theorie
  - **Architekturentwurf**
- Evaluation
- Implementierung
- Fazit & Ausblick

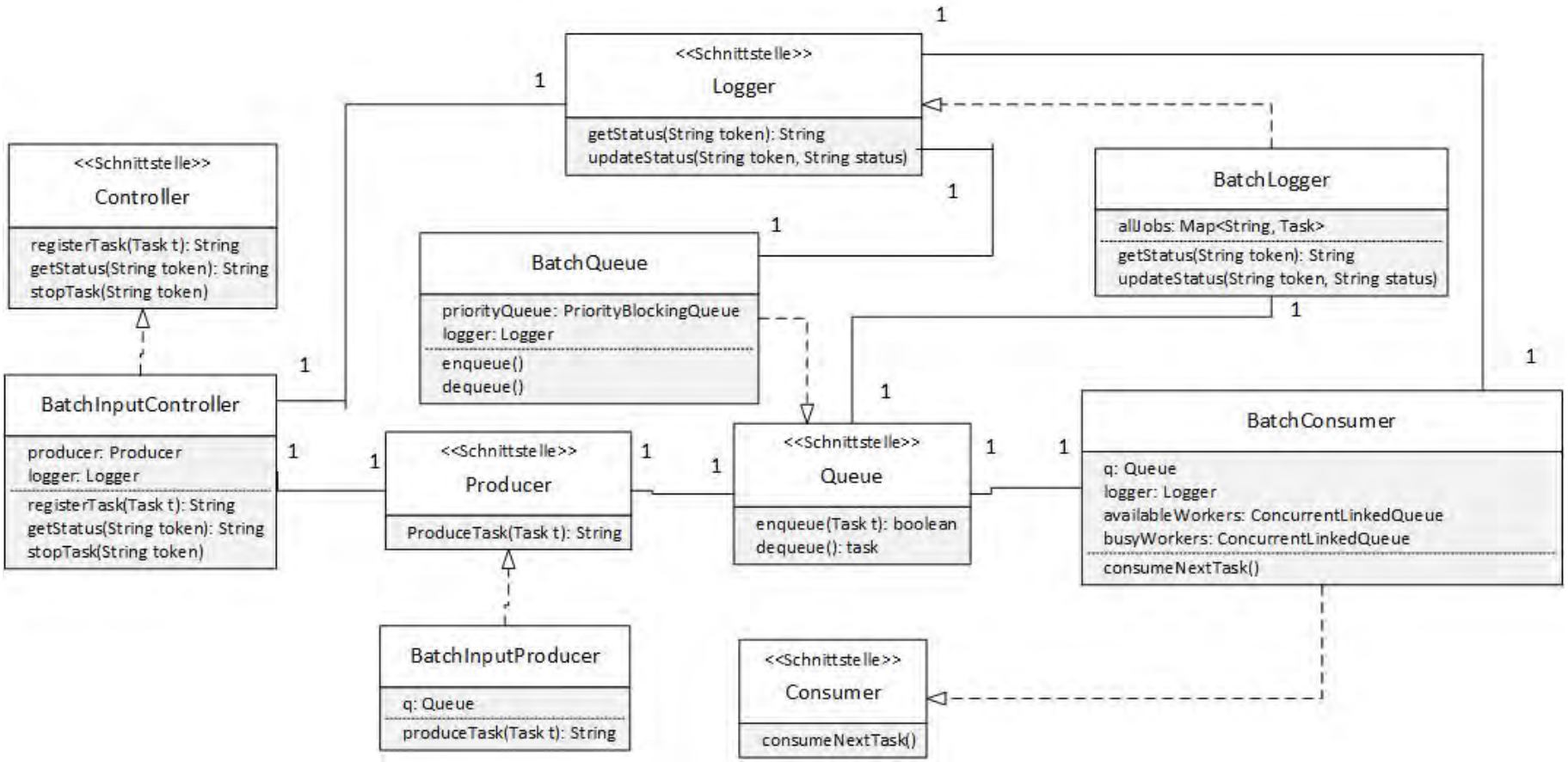
Konstantin Bork
Fachbereich Mathematik und Informatik

Architektur: Erster Entwurf



Freie Universität Berlin

Architektur: Überarbeitung



Agenda

- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - Theorie
  - Architekturentwurf
- **Evaluation**
- Implementierung
- Fazit & Ausblick

Freie Universität Berlin

Evaluation: Messaging-Systeme

Name	Apache ActiveMQ	HornetQ	Apache Kafka	Apache Qpid	RabbitMQ	Spread Toolkit
Entwickler	Apache Software Foundation	JBoss	Apache Software Foundation	Apache Software Foundation	Pivotal Software	Spread Concepts LLC
Lizenz	Apache 2.0	Apache 2.0	Apache 2.0	Apache 2.0	Mozilla Public License 1.0	Spread Open-Source License 1.0
Sprache	Java	Java	Scala	Java	Erlang	C
Unterstützte Sprachen	Java, C, C++, ...	Java	Java	Java, C, C++, Python	Java, Scala, Groovy, Erlang, ...	C, C++, Java, Python
Priorisierung	ja	ja	nein	ja	nein	nein
Grails-Unterstützung	ja	ja	ja	ja	ja	nein
Persistent	ja	ja	ja	ja	ja	ja
Queue-Modus	ja	ja	nein	ja	ja	nein
Dead-Letter-Queue	ja	ja	?	ja	ja	nein
Logs	ja	ja	ja	ja	ja	nein

Evaluation: Messaging-Systeme

- Beide Lösungen sind geeignet
- Allerdings bietet ActiveMQ mehr Vorteile
  - Speicherung von Nachrichten innerhalb des Programmspeichers
  - Mehr Möglichkeiten zur Erweiterung des Systems
  - Kein zusätzlicher Server notwendig
- Schlechte Erfahrungen mit ActiveMQ im Unternehmen

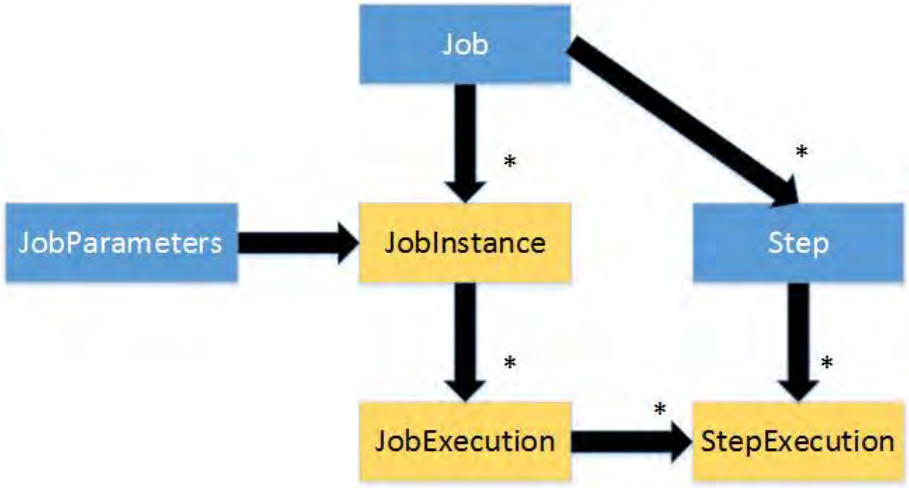
Evaluation: Enterprise-Integration-Frameworks

- Vergleich zwischen Apache Camel & Spring Integration
- Trotz schwierigeren Umgangs wird Spring Integration verwendet
  - Beide Lösungen bieten ähnlichen Funktionsumfang
  - Große Erfahrung mit Spring im Unternehmen

Freie Universität Berlin

Evaluation: Batch-Processing

- Schnelle Entscheidung für Spring Batch



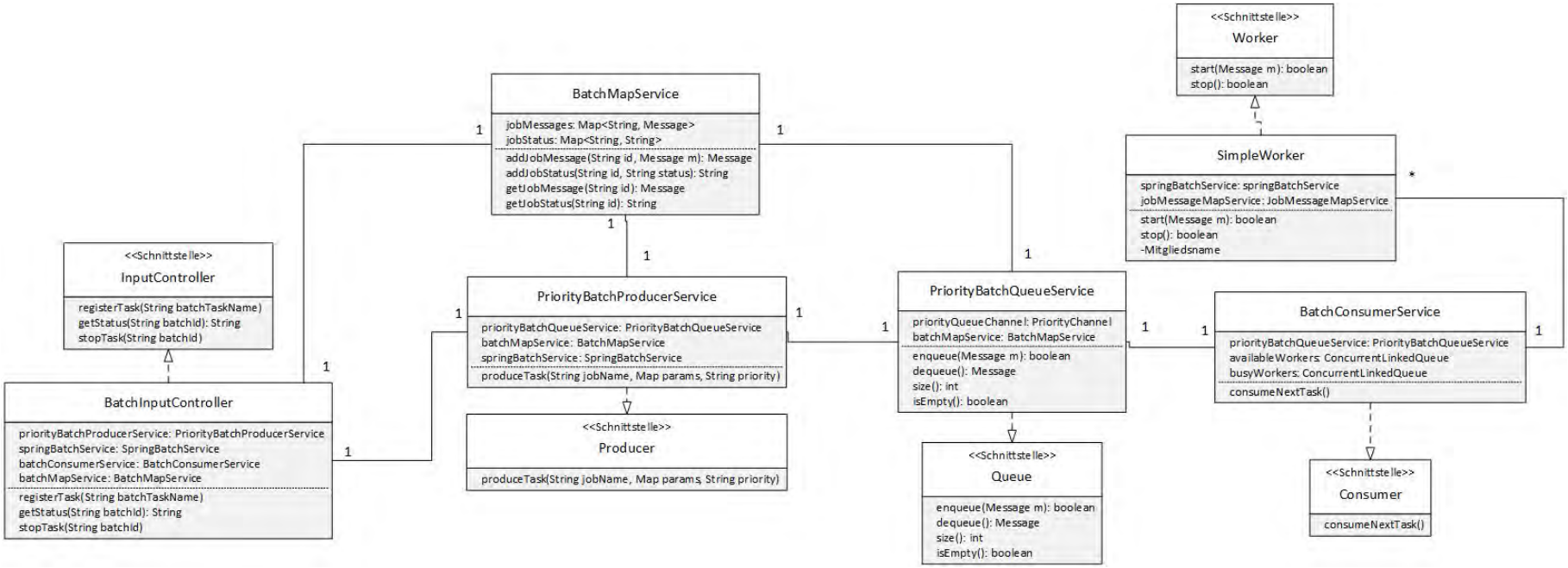
Freie Universität Berlin

Agenda

- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - Theorie
  - Architekturentwurf
- Evaluation
- **Implementierung**
- Fazit & Ausblick

Freie Universität Berlin

Implementierung: Architektur



Agenda

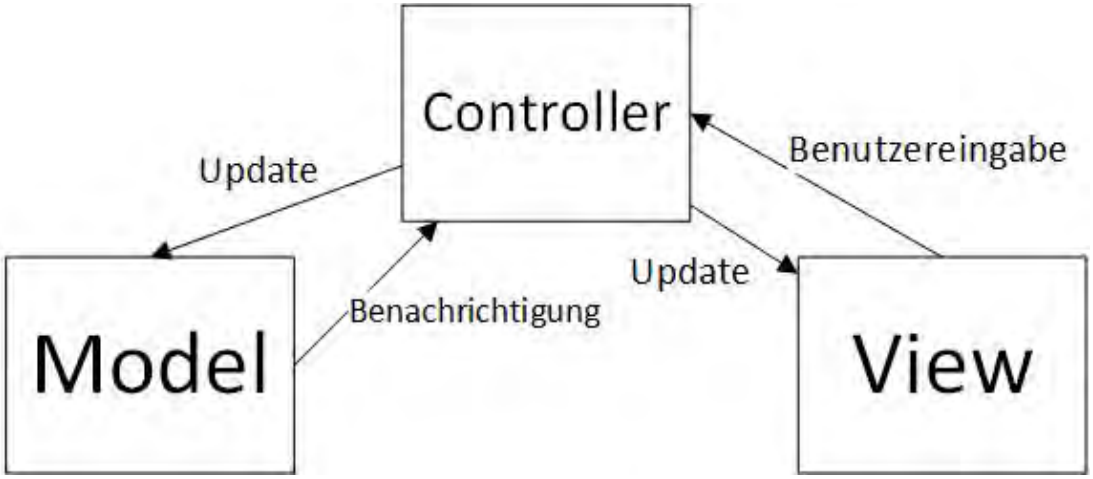
- Problembeschreibung
- Anforderungen
- Anforderungsanalyse
  - Theorie
  - Architekturentwurf
- Evaluation
- Implementierung
- **Fazit & Ausblick**

Fazit & Ausblick

- Anforderungen wurden erfüllt
  - Implementierung eines agentenbasierten Systems
  - Generisch für verschiedene Daten
  - Schnittstelle zur Steuerung der Auftragsverarbeitung
  - Spätere Verwendung eines Messaging-Systems möglich
- Weitere Optimierungen erforderlich
- Einbau in bestehende Systeme ist geplant
- Open Source Veröffentlichung ist geplant

Theorie: Architekturmuster

- Model-View-Controller



Implementierung: Vorgehen

- Auswahl der verwendeten Technologie
- Verwendung von Git
- Entwicklung wird in Teilprobleme zerlegt
- Unit- und Integration-Tests
- Test der Funktionsfähigkeiten in einem Beispielprojekt