

# *Swing: Daten prüfen*

Karsten Lentzsch



# *Ziel*

Ansätze kennenlernen  
wie man in Swing  
Fachdaten prüfen kann

Lösungen bewerten können

# *Vorstellung*

- Ich baue Swing-Anwendungen, die viele Leute elegant finden
- arbeite seit 1990 mit Objekten
- helfe Anderen über und unter der Haube
- biete Bibliotheken, die Swing ergänzen
- biete Swing-Beispiele zu Architekturen
- und schreibe über Desktop-Themen

# *Gliederung*

- Einleitung
- Grundlagen
- Wer kann prüfen?
- Wann und wie prüfen?
- Wie anzeigen?
- Erfahrungsbericht

# I - Einleitung

# Swing-Bausteine



# Swing-Bausteine



# *Fragen*

- Worum geht's bei der Datenprüfung?
- Wer kann und soll Daten prüfen?
- Wann kann ich prüfen?
- Wie schreibe ich den Prüfcode?
- Wo zeige ich Prüfergebnisse?
- Wie markiere ich ungültige Eingaben?
- Wie weise ich hin auf gültige Werte?



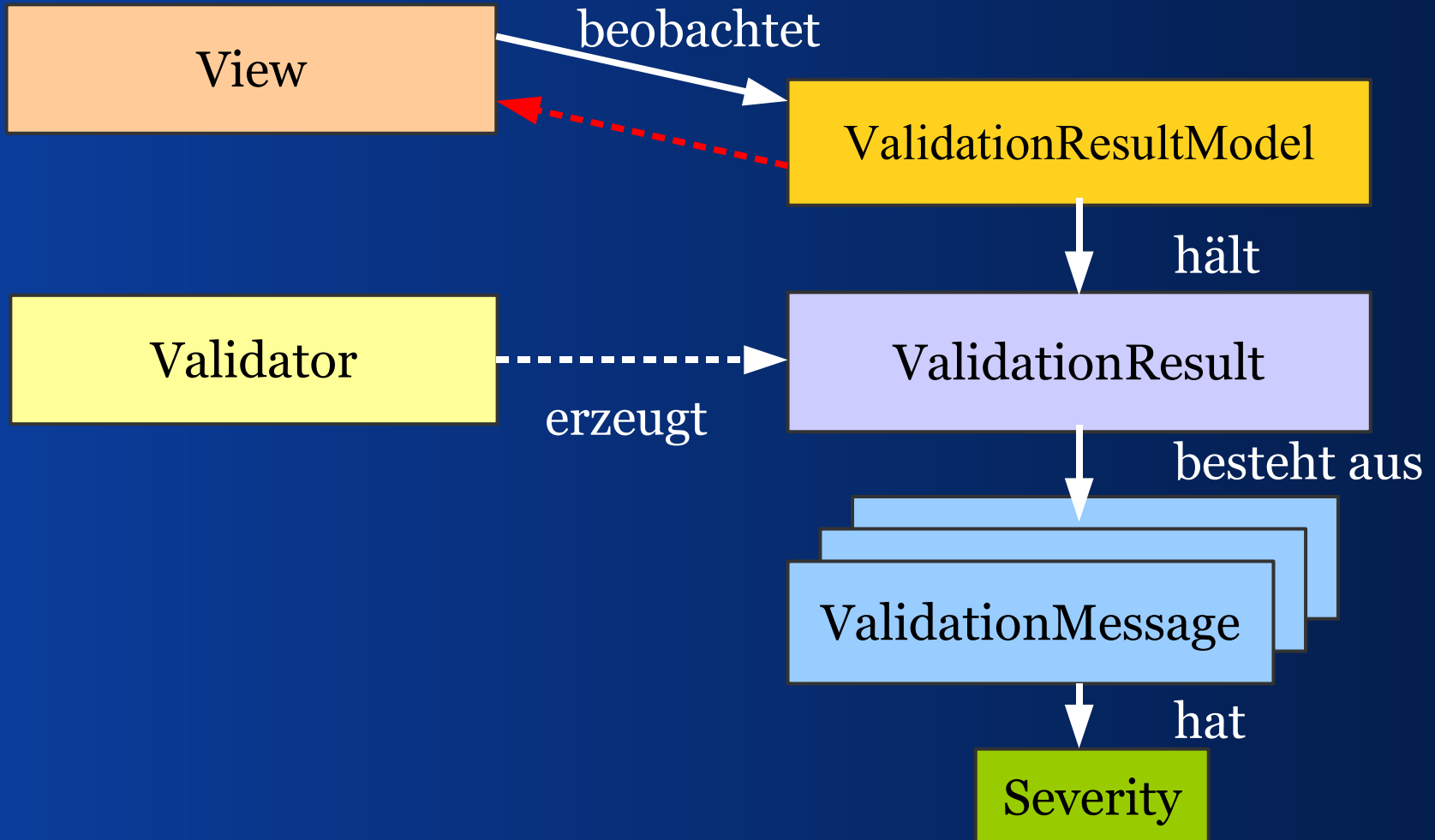
# II - Grundlagen

*Was brauche ich, um Daten zu prüfen?*

# *Aufbau*

- Jemand kann prüfen - Validator
- Er liefert ein Ergebnis - ValidationResult
- Das besteht aus ValidationMessages
- Severity zeigt den Schweregrad an
- ValidationResultModel meldet Änderungen
- ValidationComponentUtils markiert Views
- ValidationResultViewFactory liefert Views

# Prüfraahmenwerk



# *Entwurfsziele*

- Arbeitet mit Standard-Swing-Komponenten
- Arbeitet mit eigenen Komponenten
- Braucht keine speziellen Komponenten
- Braucht keine speziellen Panels
- Passt zu verschiedenen Datenbindestilen
- Passt zu verschiedenen Prüfstilen
- Ist erweiterbar

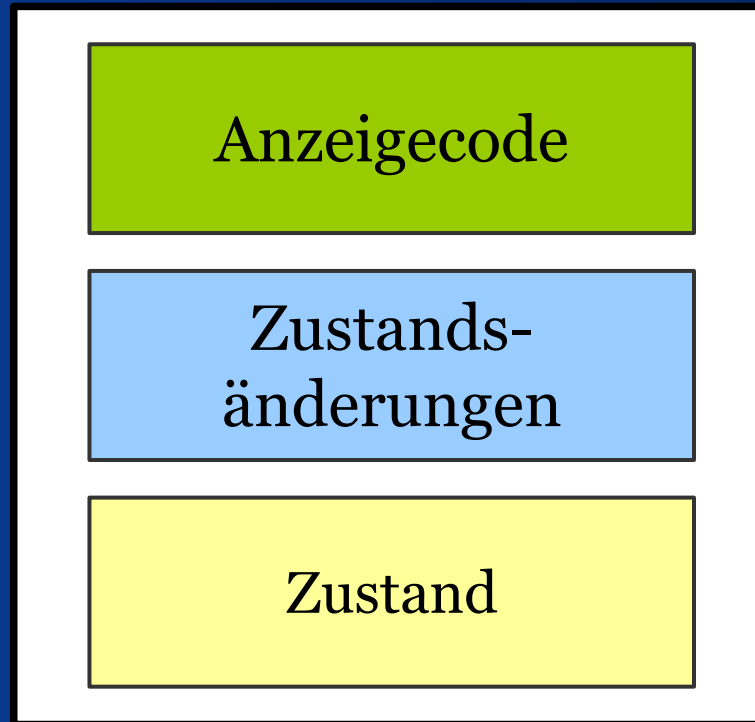
# *Was bietet JGoodies Validation?*

- Implementiert den vorgestellten Ansatz
- Ist allgemein gehalten
- Braucht keine anderen Bibliotheken
- Kann binär genutzt werden
- Kann als Grundlage dienen für eigene Validierungslösungen

# III – Wer?

*Wer prüft wen?*

# *Client und Server*

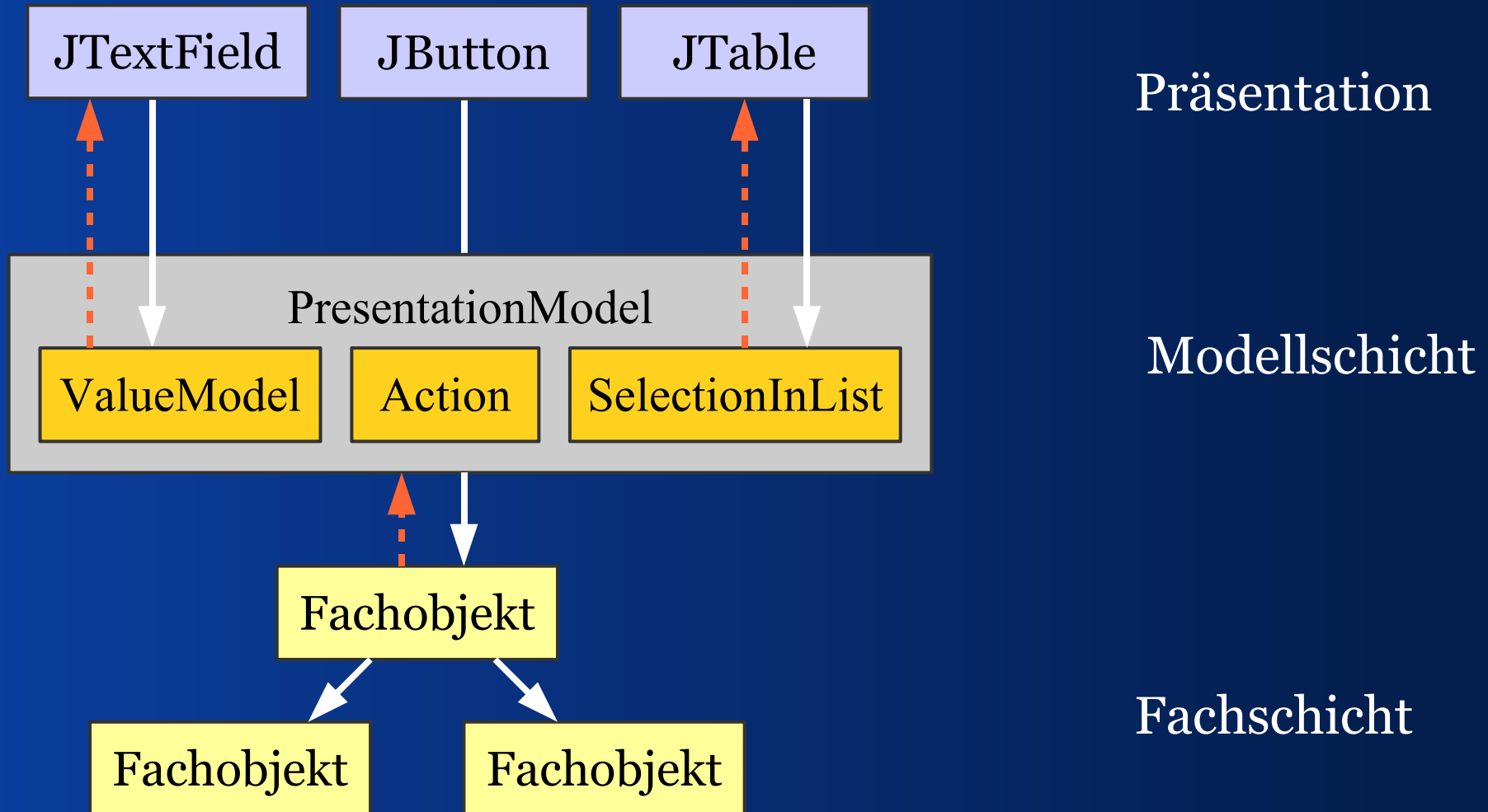


Client



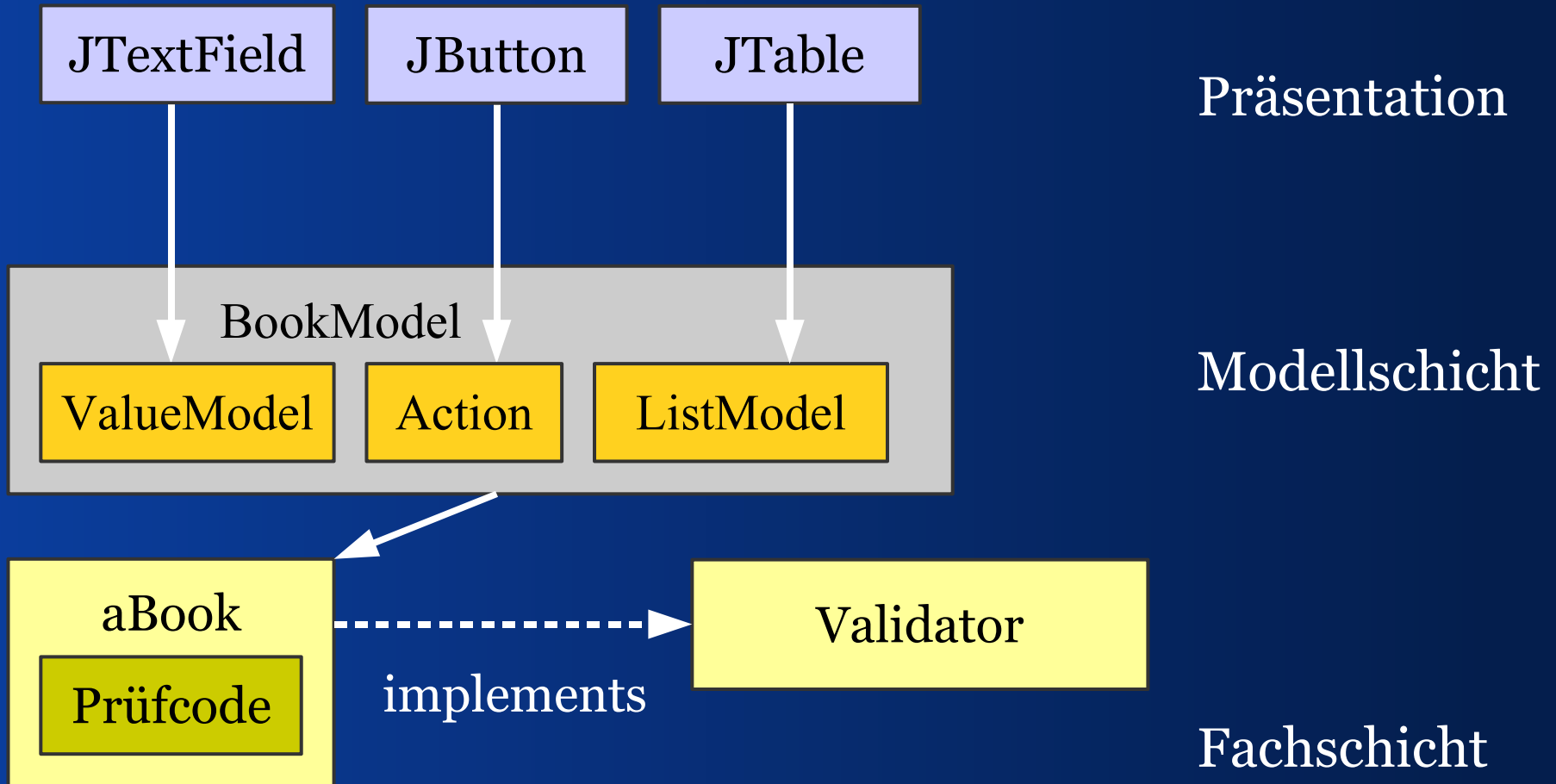
Server

# 3-Schichten-Client-Architektur

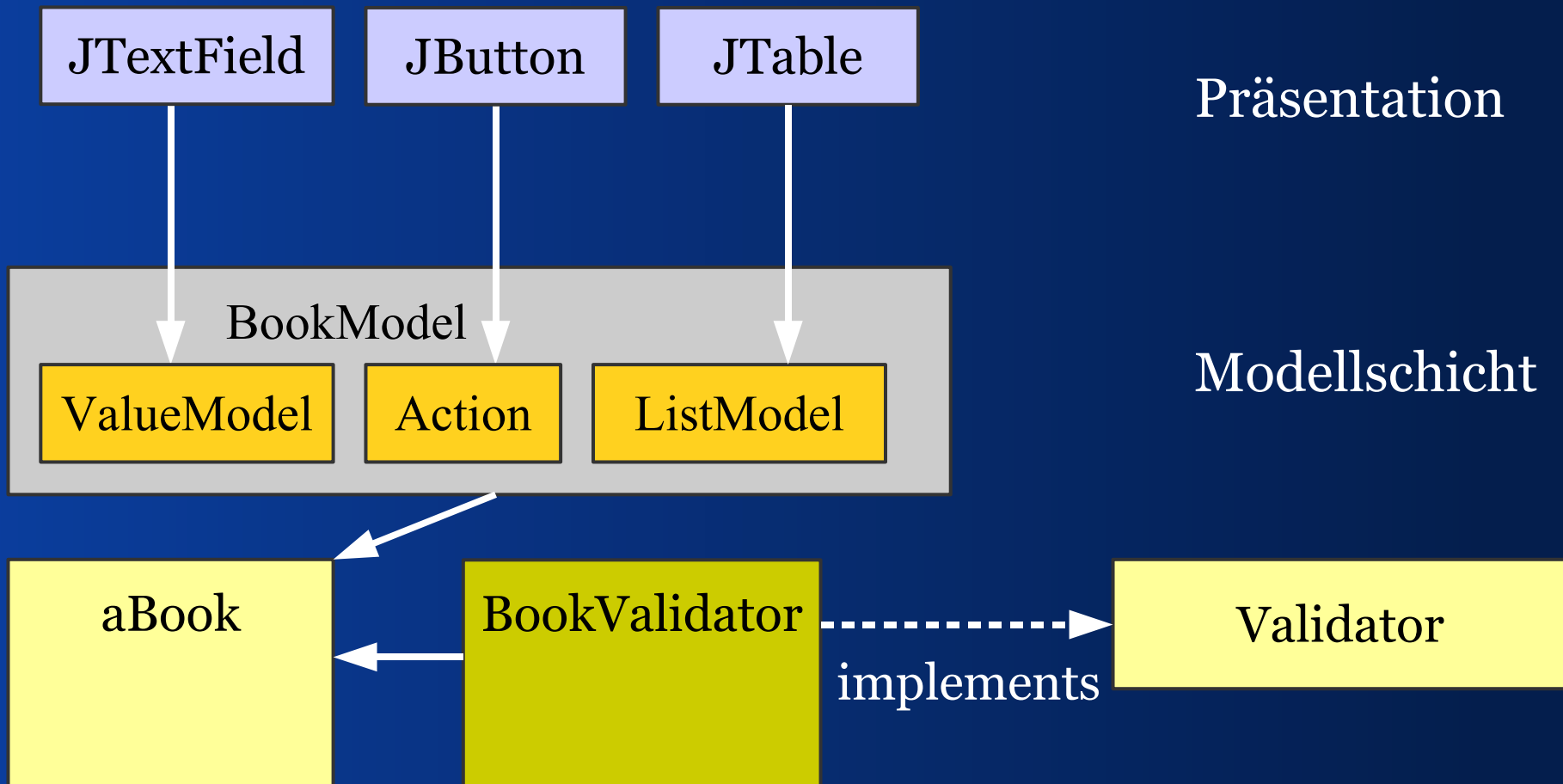




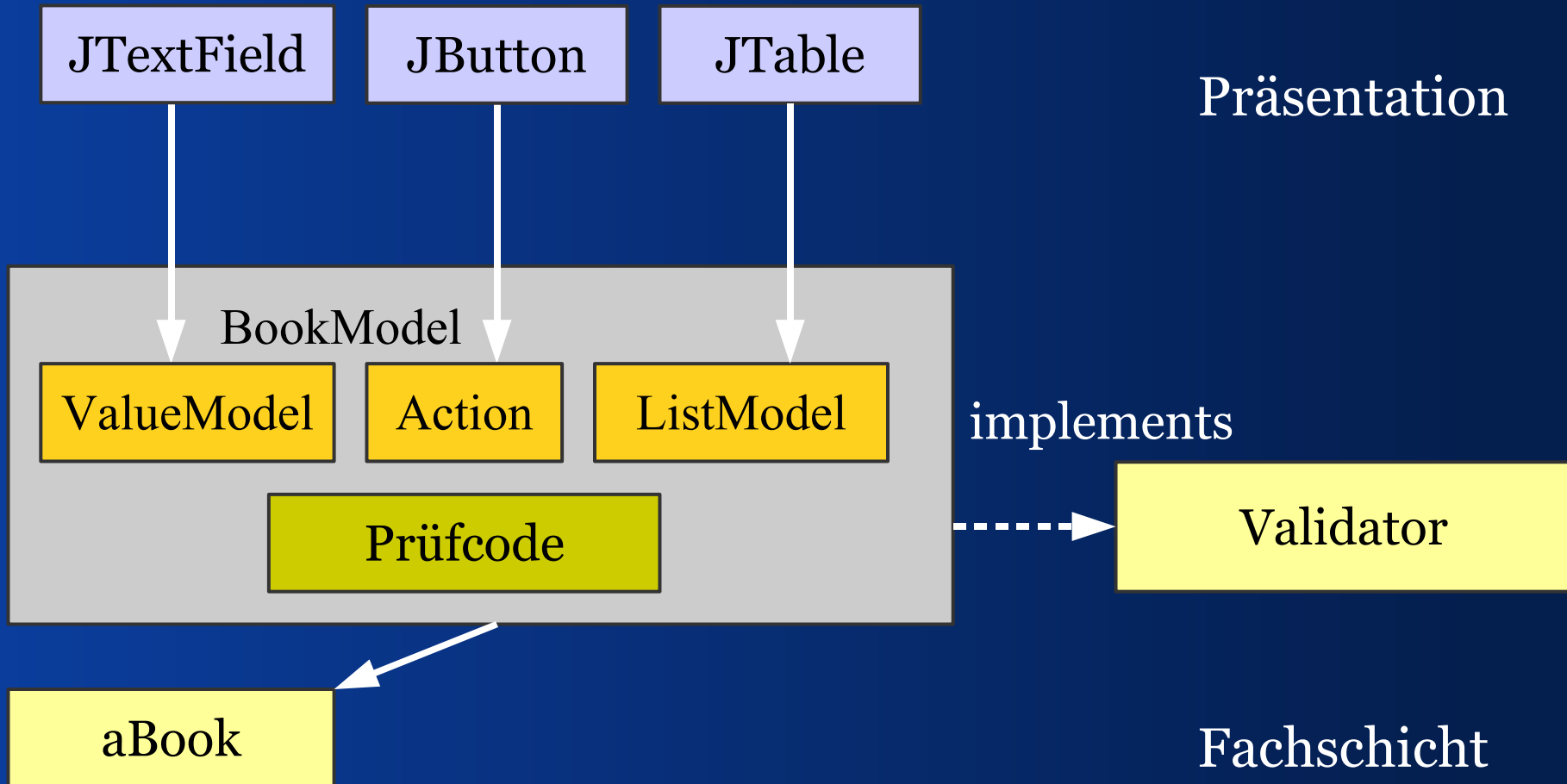
# *Fachobjekt prüft*



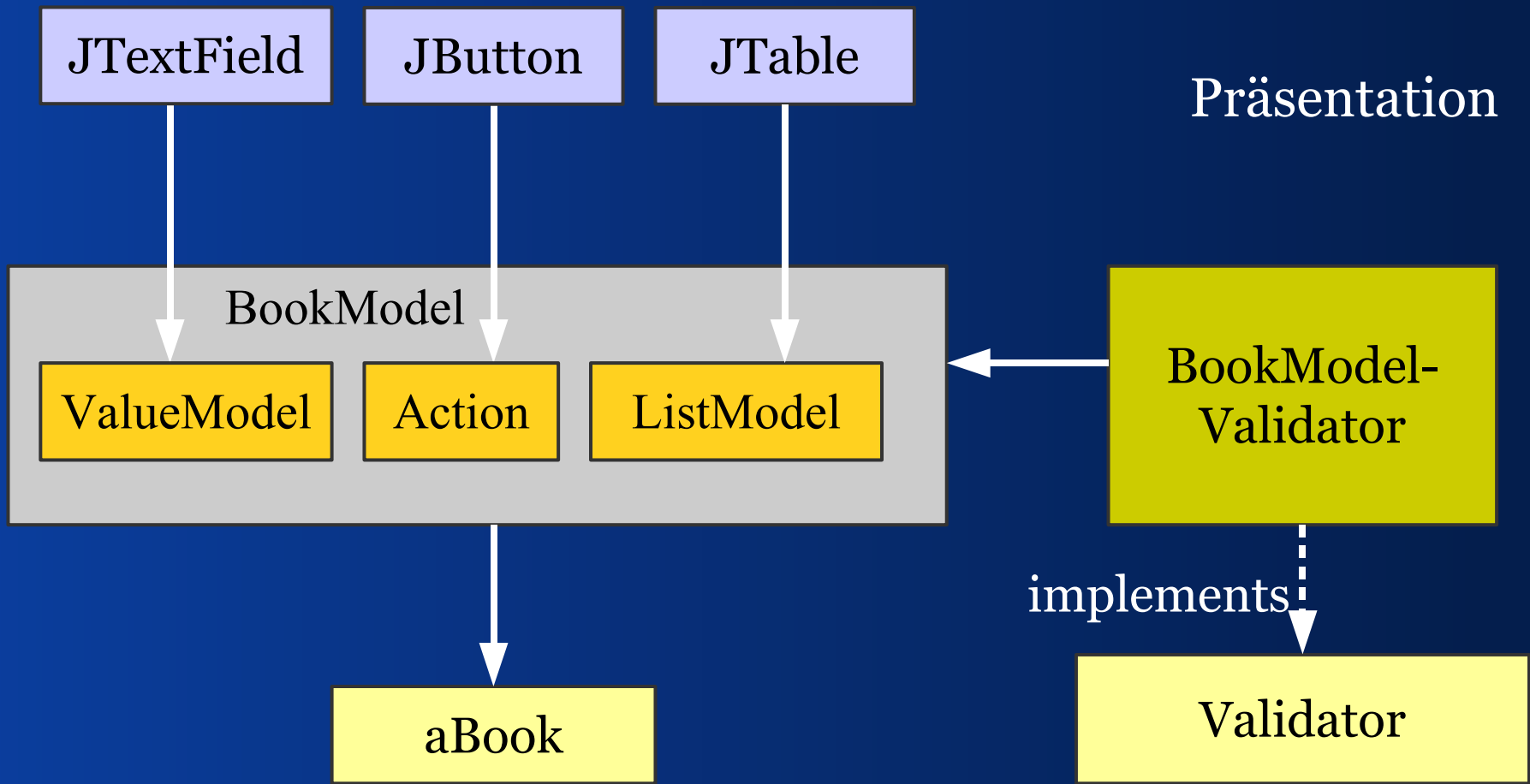
# *Validator zu Fachobjekt*



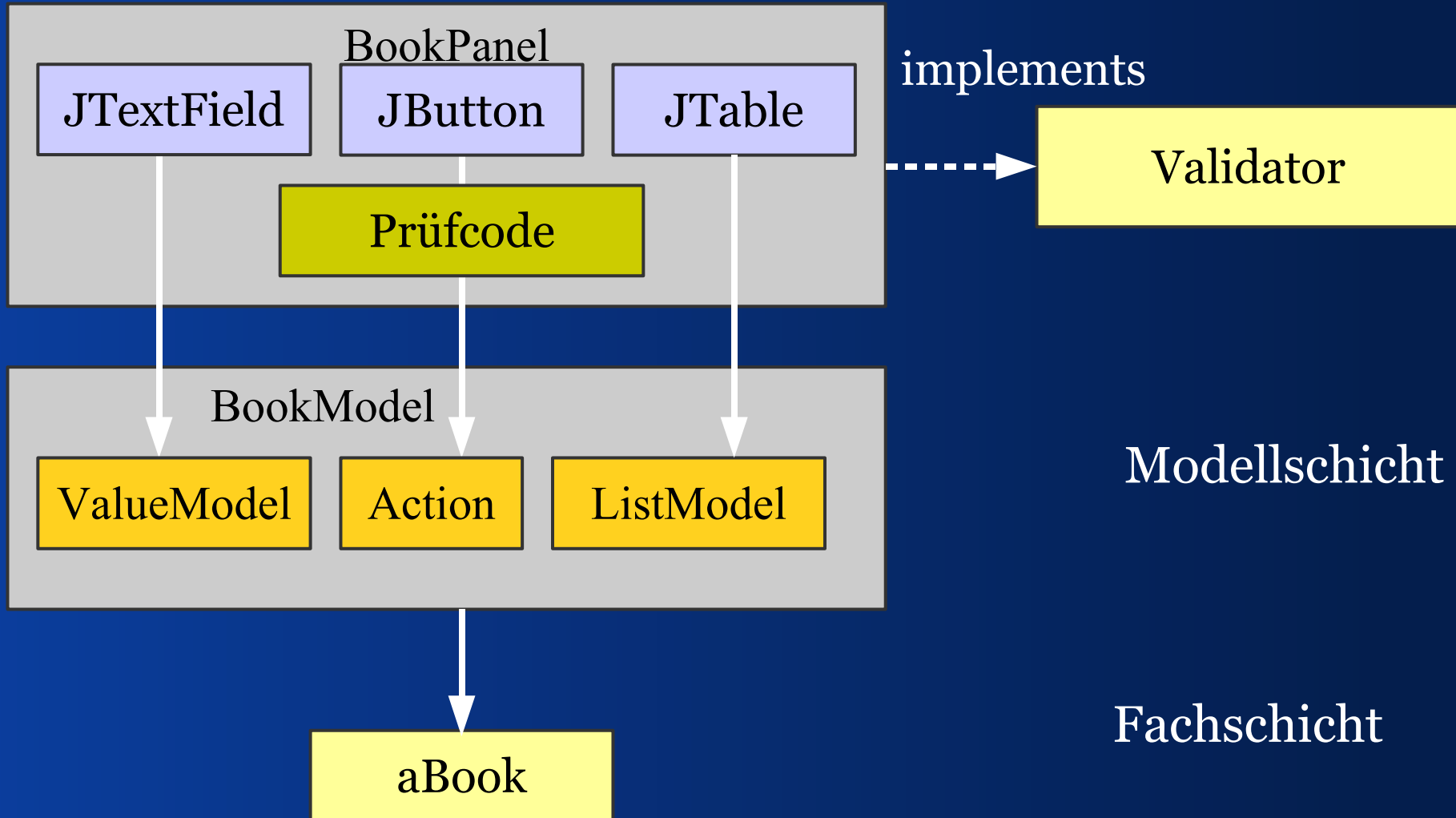
# *PresentationModel prüft*



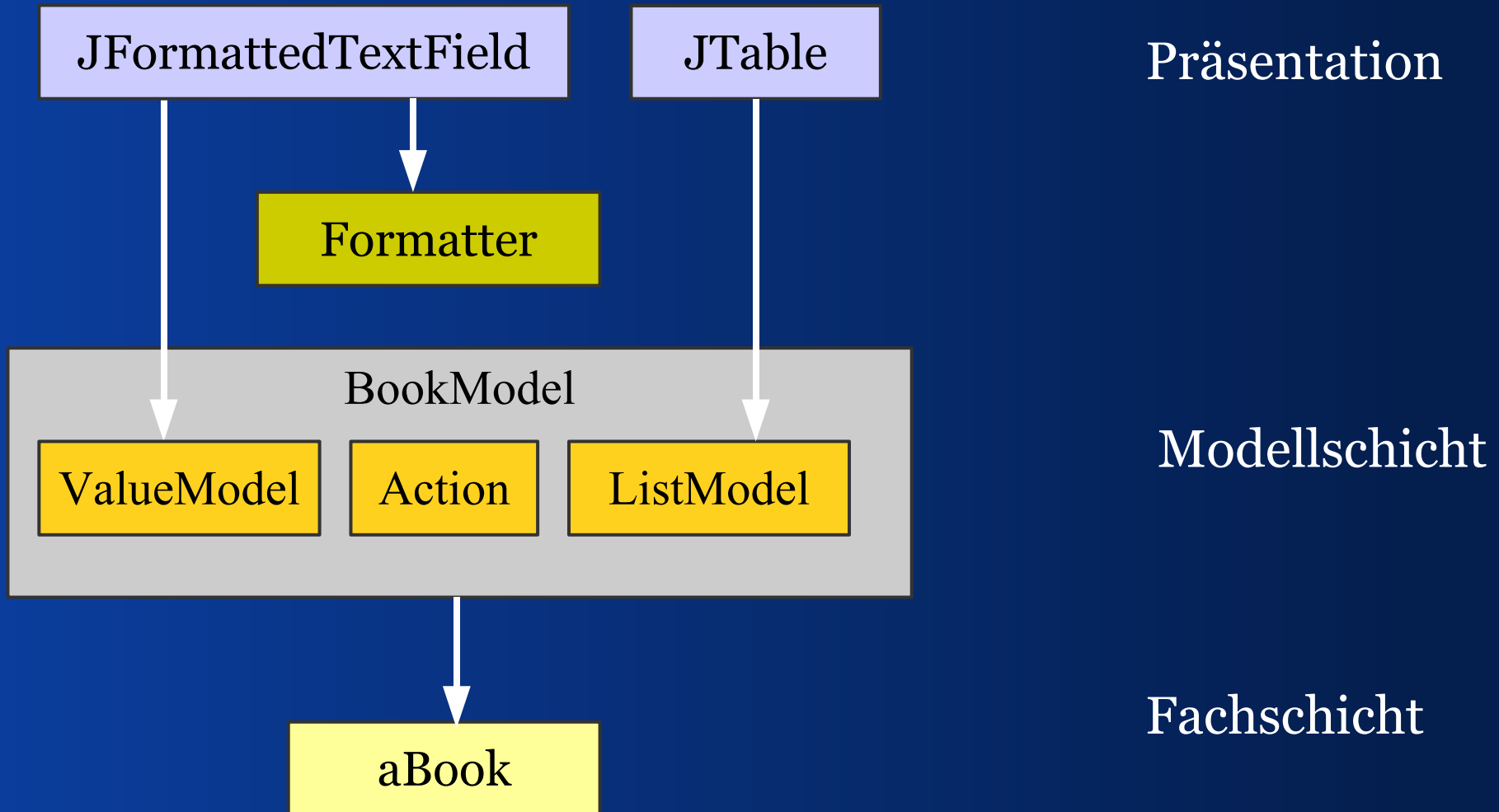
# Validator zu Module



# Präsentation prüft



# *View filtert ungültige Daten*



# IV – Wann?

*Wann prüfe ich? Wer löst das aus?*

# *Prüfzeitpunkte*

- Zeichen getippt (key typed)
- Fokus gewechselt (Focus lost)
- Eingabe übernehmen (OK/Apply)
- Daten speichern/drucken/sendern
  
- Siehe Beispiele in der Validation Demo

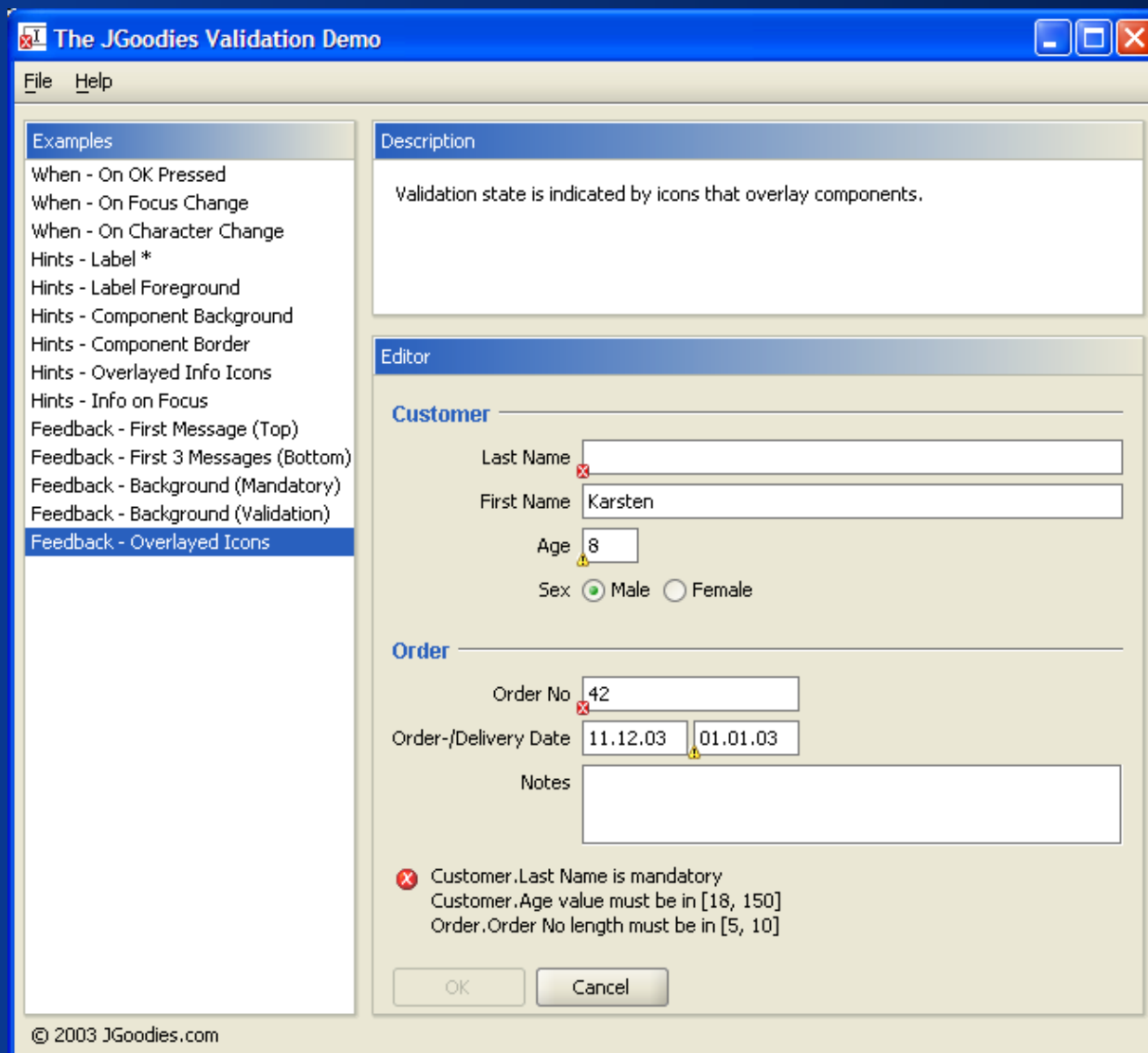


# *Beispielanwendung:*

## JGoodies Validation Demo

*Aufgaben und Ansätze zur Datenprüfung*

[www.JGoodies.com/freeware/validationdemo](http://www.JGoodies.com/freeware/validationdemo)



# V – Wie anzeigen?

*Wie kann ich Prüfergebnisse anzeigen  
und auf ungültige Eingaben hinweisen?*

# *Prüfergebnisanzeigen*

- Einzelnes Label mit Prüftext und Warn-Icon
- TextArea für mehrere Prüfergebnistexte
- JList/JTable für mehrere Ergebnisse
- Überlagerter, halb-transparenter Text
  
- Siehe Beispiele in der Validation Demo

# *Ungültige Eingaben markieren*

- Textfeldhintergrund
- Komponentenrahmen
- Icon schwebt über Komponente
- Transparenter Text schwebt neben Komp.
  
- Siehe Beispiele in der Validation Demo

# VI - Erfahrungsbericht

*Wie funktioniert dieser Prüfstil im Alltag?*

# *Allgemeines*

- Jakarta Commons nutzen
- Möglichst auf ungepufferten Werten prüfen
- Eigene Validatoren schreiben, nicht die Fachobjekte oder Modelle prüfen lassen
- In der Fachschicht prüfen (wenn's geht)
- Meide Präsentationsprüfungen, aber nutze Filter (JFormattedTextField)

# *Wo steht JGoodies Validation?*

- Architektur ist stabil seit Dezember 2003
- Keine Fehler oder Probleme bekannt
- Tutorial und Demo vorhanden
- Wenige vordefinierte Prüfregeln



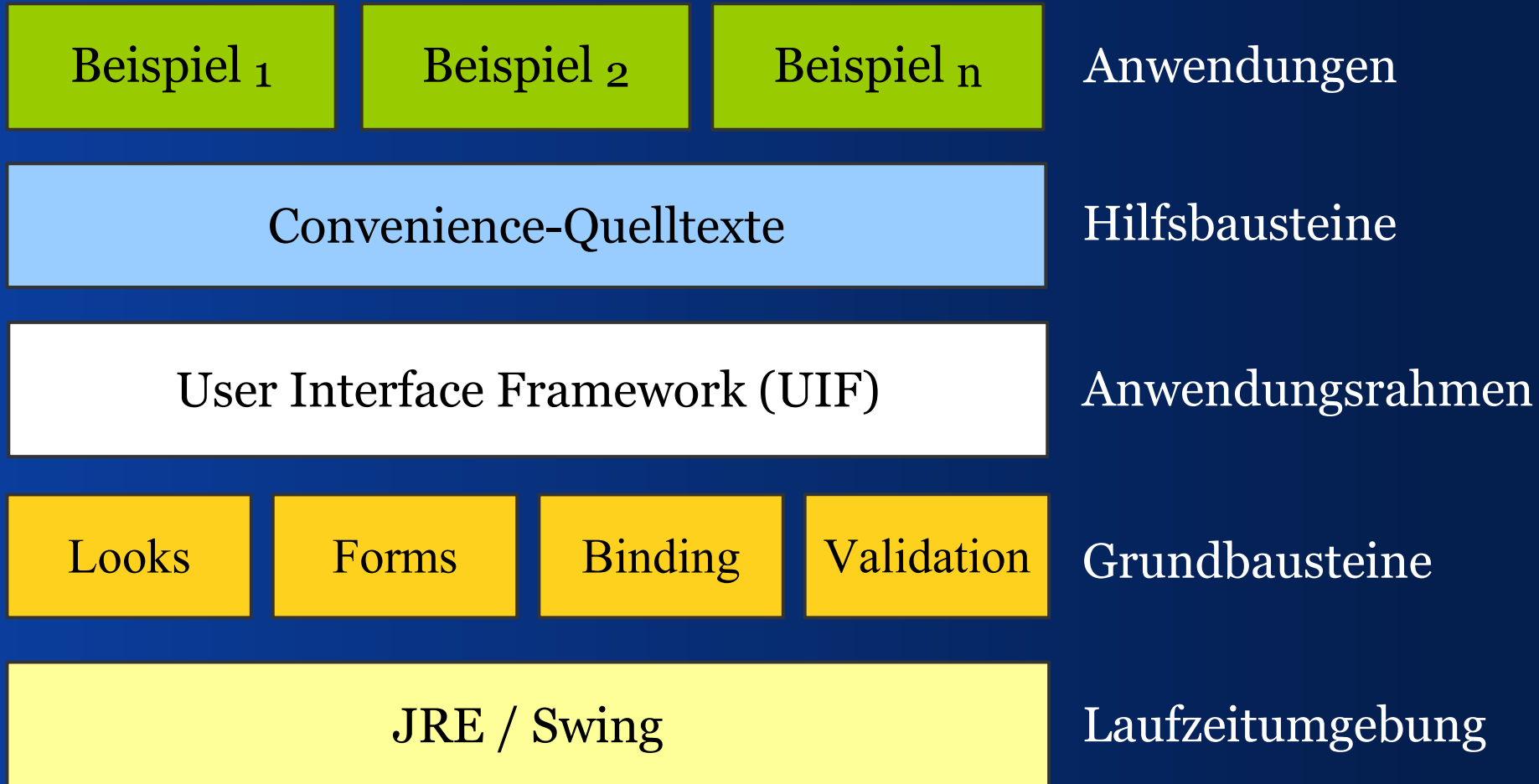
# Schluss

*Zusammenfassung, Referenzen, Literatur*

# *Zusammenfassung*

Wir haben Prüfaufgaben identifiziert und gelernt, wer, wann, wie prüfen kann und wie man Prüfergebnisse anzeigen kann.

# *JGoodies Swing Suite*



# *Referenzen I*

- JGoodies Validation  
[validation.dev.java.net](http://validation.dev.java.net)
- JGoodies-Artikel  
[www.JGoodies.com/articles/](http://www.JGoodies.com/articles/)
- JGoodies-Demos  
[www.JGoodies.com/freeware/](http://www.JGoodies.com/freeware/)

# Referenzen II

- Oracle's JClient und ADF  
[otn.oracle.com/](http://otn.oracle.com/), nach 'JClient' suchen
- Spring Rich Client Project  
[www.springframework.org/spring-rcp.html](http://www.springframework.org/spring-rcp.html)
- JDNC (Java Desktop Network Components)  
[jdnc.dev.java.net](http://jdnc.dev.java.net)

# Fragen und Antworten

*Ende*

Hoffentlich hilft's!

Viel Erfolg!