

Thomas Uhrig: Design Patterns – Decorator & Fassade

Decorator

&

Fassade

- **Dekorator**

Strukturmuster der GoF

flexible Bildung von Unterklassen



Decorator

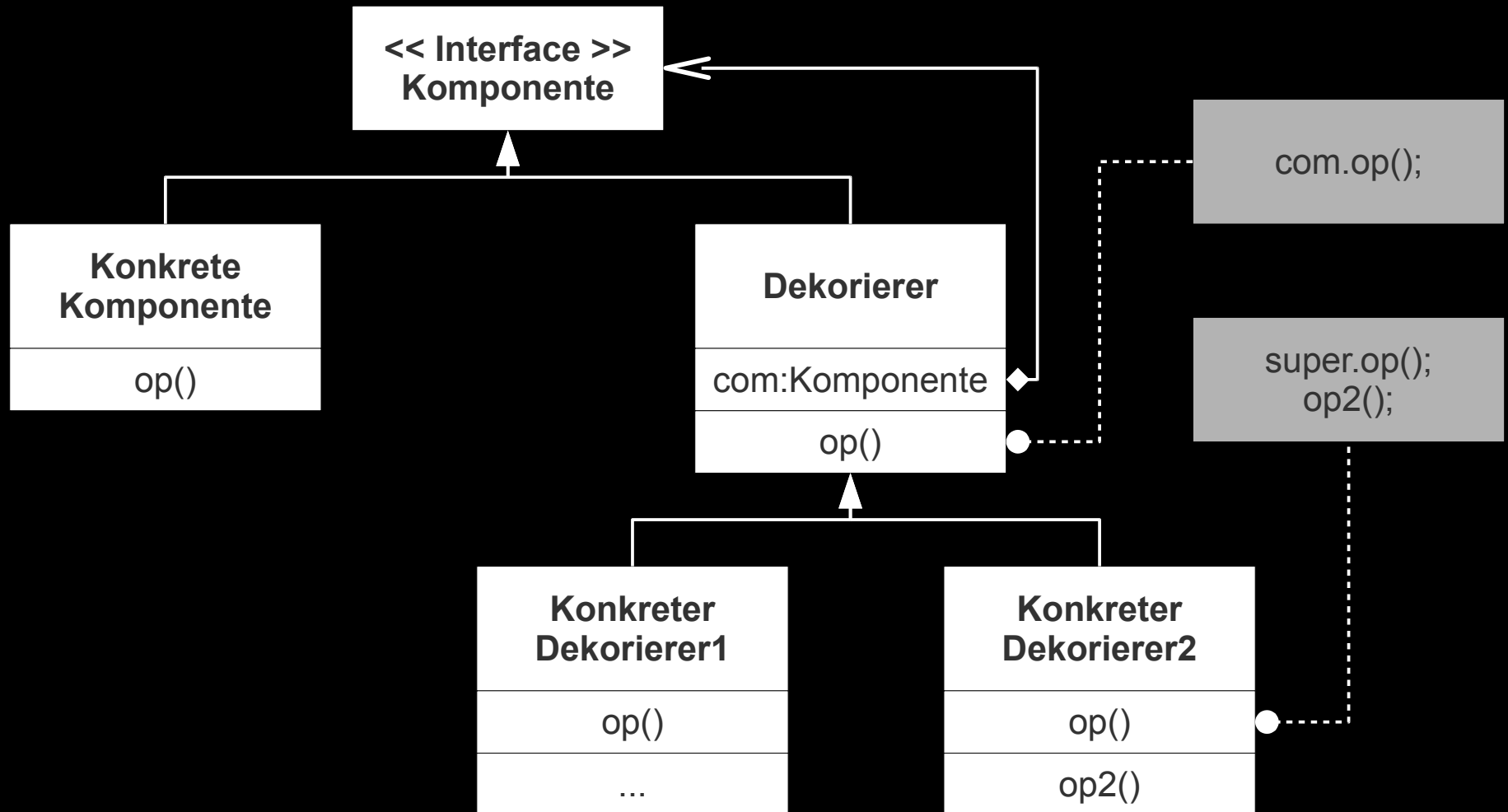
- **Zweck**

dynamisches Hinzufügen von Funktionalität

die Komponente ändert sich nicht

„Erweiterbarkeit“ zur Laufzeit

• Lösung



- **Vorteile**

Komposition statt Vererbung

zur Laufzeit möglich („Zusammenstecken“)

Funktionalität ist leicht „entfernbar“

transparent für den Client

- **Nachteile**

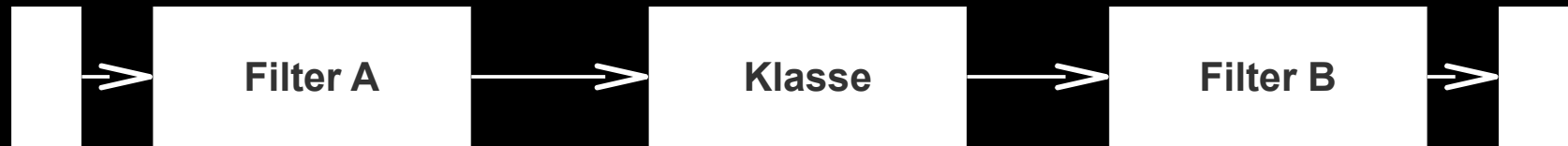
jede neue Funktionalität, ein neuer Dekorator

viele ähnliche Klassen (Übersichtlichkeit)

unklare Objektidentität

- **Varianten & Verweise**

(Decorating) Filter/ Pipes & Filter



Strategy (kapselt eine Funktion in einer Klasse)

- **Fassade**

Strukturmuster der GoF

Schnittstelle zu Subsystemen



Fassade

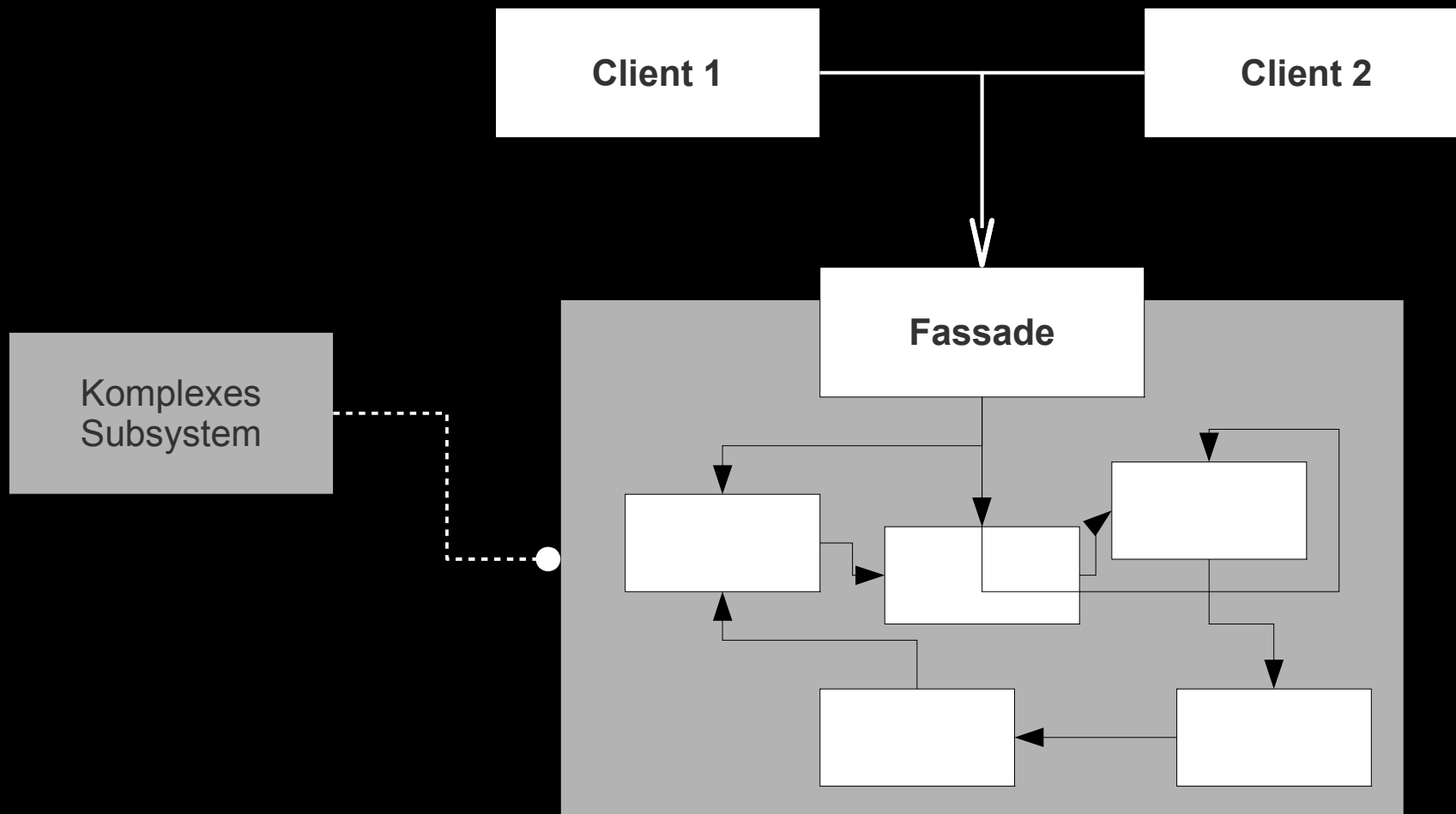
- **Zweck**

kapseln von komplexen Subsystemen

zentraler und einfacher Zugriff

Clients müssen das Subsystem nicht kennen

- Lösung



- **Vorteile**

entkoppelt Clients vom Subsystem (Kapselung)

zentraler Zugriff

lose Kopplung

- **Nachteile**

Clients dürfen die Fassade nicht umgehen

Fassade muss bei Änderungen angepasst werden

erhöhter Aufwand (Indirektionsstufe)

- **Varianten & Verweise**

Adapter (kapselt den Zugriff auf nur eine Klasse)

Proxy (Stellvertreter für eine Klasse)

Message-Fassade (zeitliche Entkoppelung)

- **Quellen**

Starke & Eilebrecht: Patterns kompakt. Spektrum Verlag 2010.

Wikipedia: <http://de.wikipedia.org/wiki/Viererbande>

Decorator

Klassifikation:

Strukturmuster

Zweck:

dynamisch zusätzliche Funktionalität

Szenario / Motivation:

Funktionen ändern, ohne Klassen zu ändern

Lösung / Struktur:

Komposition von Klassen mit gleichem Interface

Konsequenzen (Vor- & Nachteile):

Verkettung; viele ähnliche Klassen

Varianten & Verweise:

Filter; Strategy

Fassade

Klassifikation:

Strukturmuster

Zweck:

Kapselung komplexer Subsysteme

Szenario / Motivation:

lose und einfache Kopplung zw. Client und System

Lösung / Struktur:

zentraler Zugang

Konsequenzen (Vor- & Nachteile):

darf nicht umgangen werden; Indirektionsstufe

Varianten & Verweise:

Adapter; Proxy;