

# Wahlpflichtfach Design Pattern

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- miwilhelm@hs-harz.de
- <http://www.miwilhelm.de>
- Raum 2.202
- Tel. 03943 / 659 338

# Inhalt

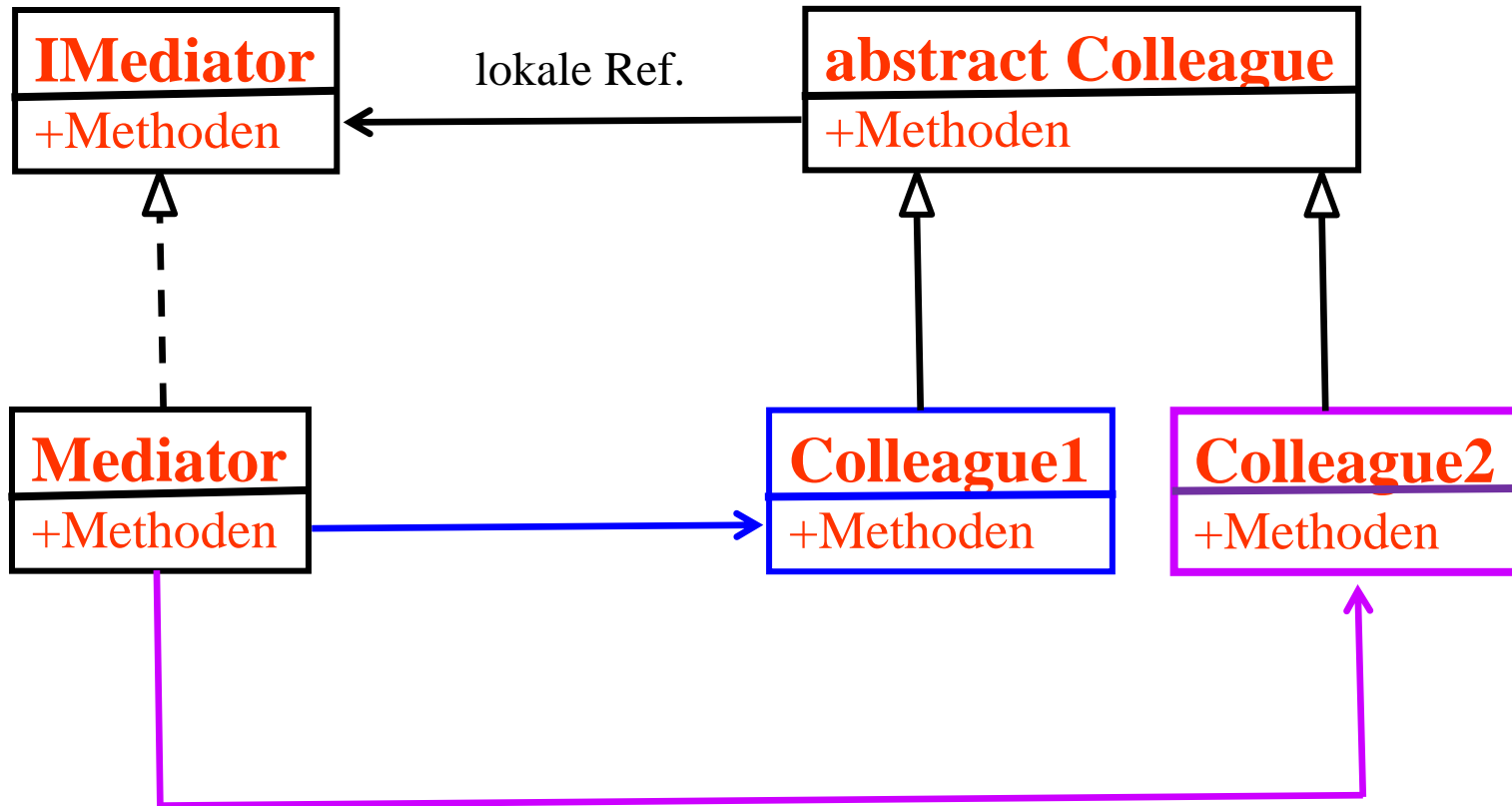
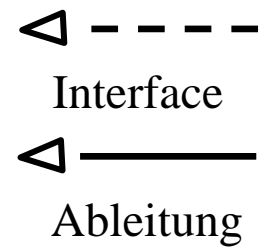
1. Einleitung
2. Singleton
3. Observer
4. Decorator
5. Abstract Factory
6. Adapter
7. Facade
- 8. Mediator**
8. Bridge
9. MVVM
10. Java Collection Framework
11. Command / Befehl

# Entwurfsmuster: „Mediator“

- **Mediator-Pattern**

- Verhaltens-Design-Pattern.
- Erlaubt die zentrale **Kommunikation** zwischen verschiedenen Klassen.
- Es ist eher eine lose Kopplung.
- Die Klassen sind meistens unabhängig voneinander.
- Eine **direkte** Kopplung zwischen zum Beispiel 100 Systemen ist sehr komplex.
- Durch das Mediator-Pattern entsteht eine Entkopplung zwischen den Systemen. Gleichzeitig entwickelt man eine einfachere Schnittstelle.
- Es gibt zwei Hauptklassen:
  - Mediator
  - Colleague

# Entwurfsmuster: „Mediator“



Die Colleague haben eine lokale Referenz auf dem Mediator  
Der Mediator hat Referenzen auf alle Colleagues

# Entwurfsmuster: „Mediator“

- **Beispiele**

- Chat-Programm:

- Kommunikation zwischen den Mitgliedern.

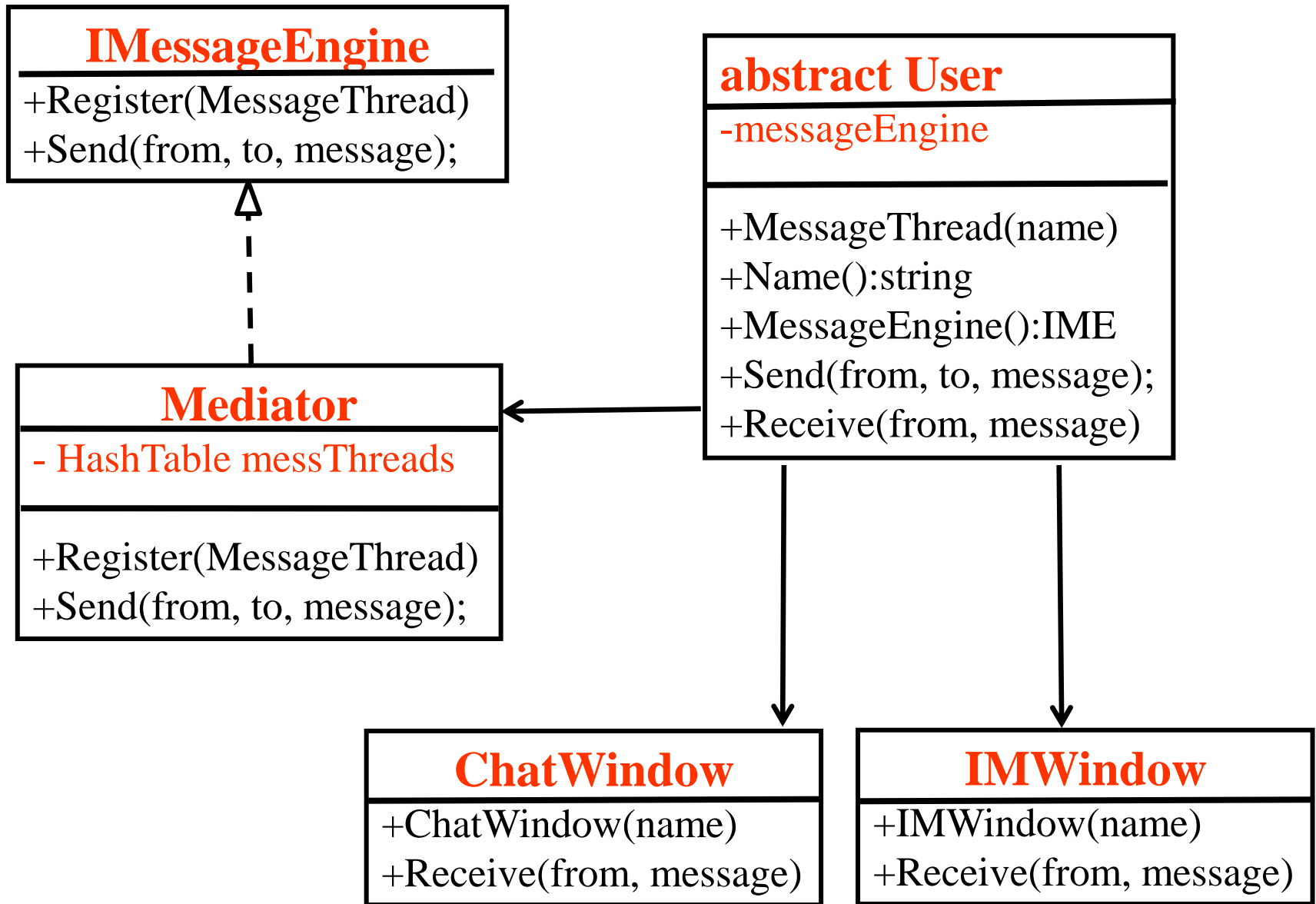
- Betriebssystem

- Kommunikation zwischen Fenstern.

- Netzwerk

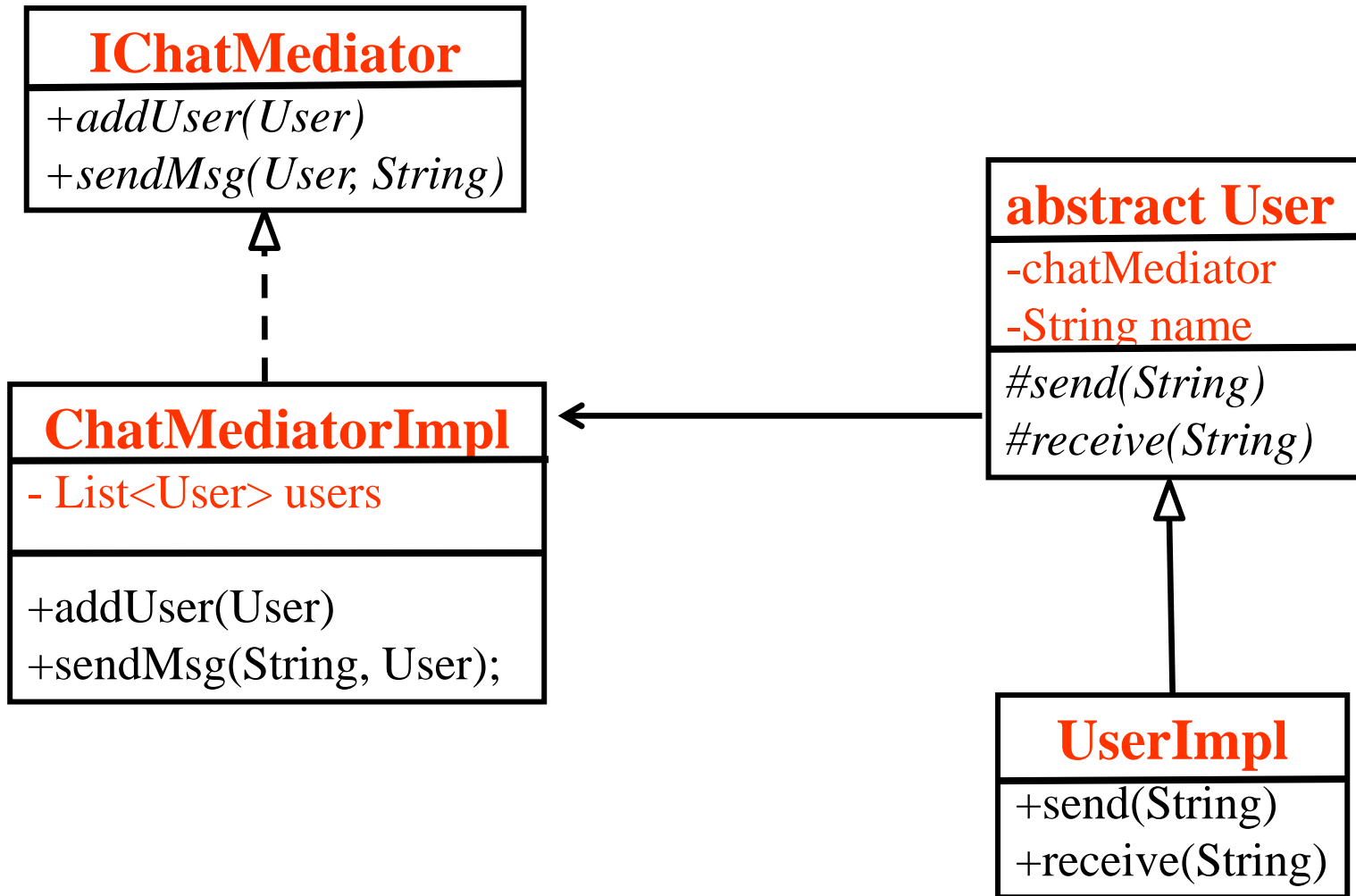
- Kommunikation zwischen Servern

# 1. Beispiel: Fensternachrichten



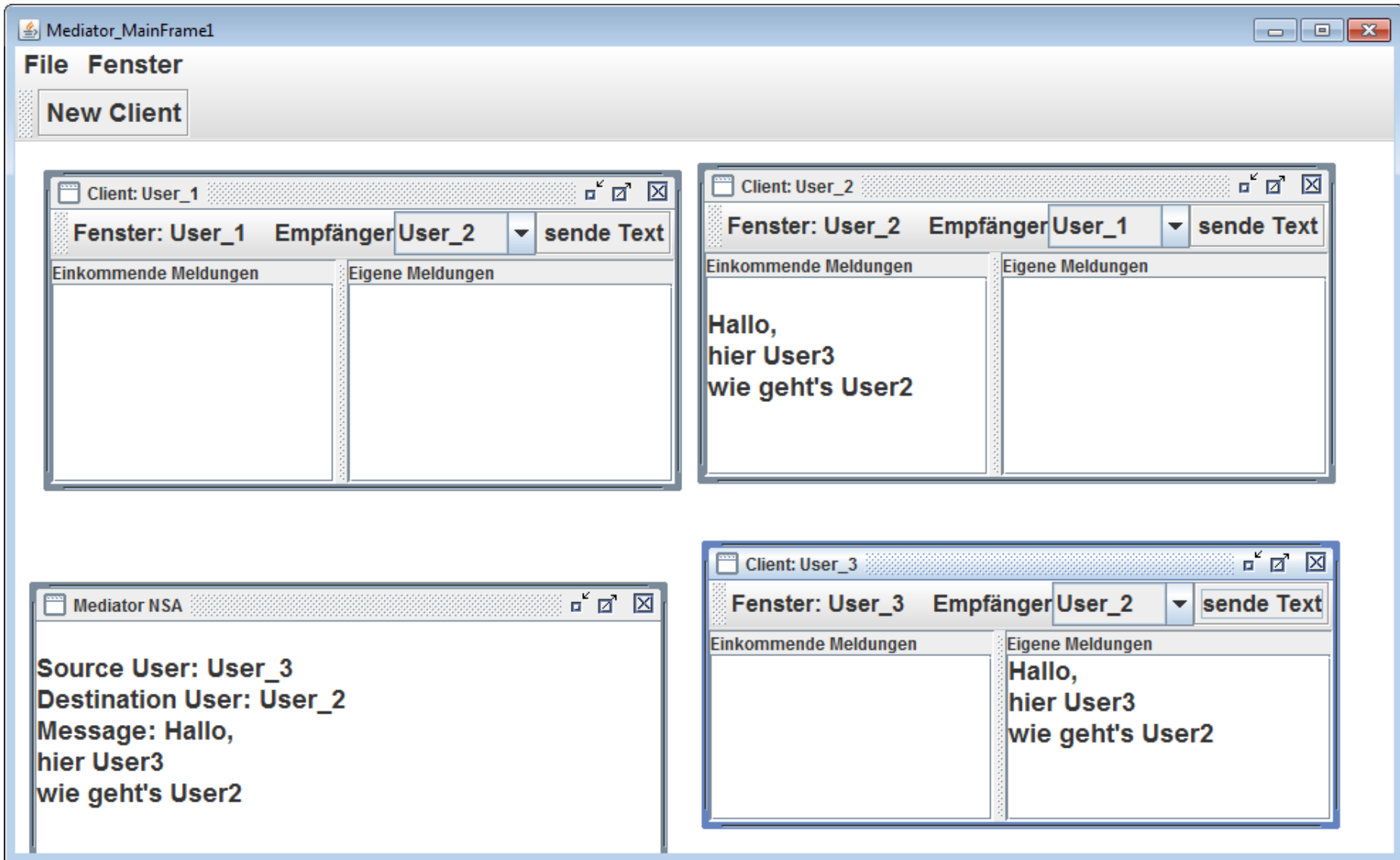
## 2. Beispiel: Fensternachrichten

# *abstract*



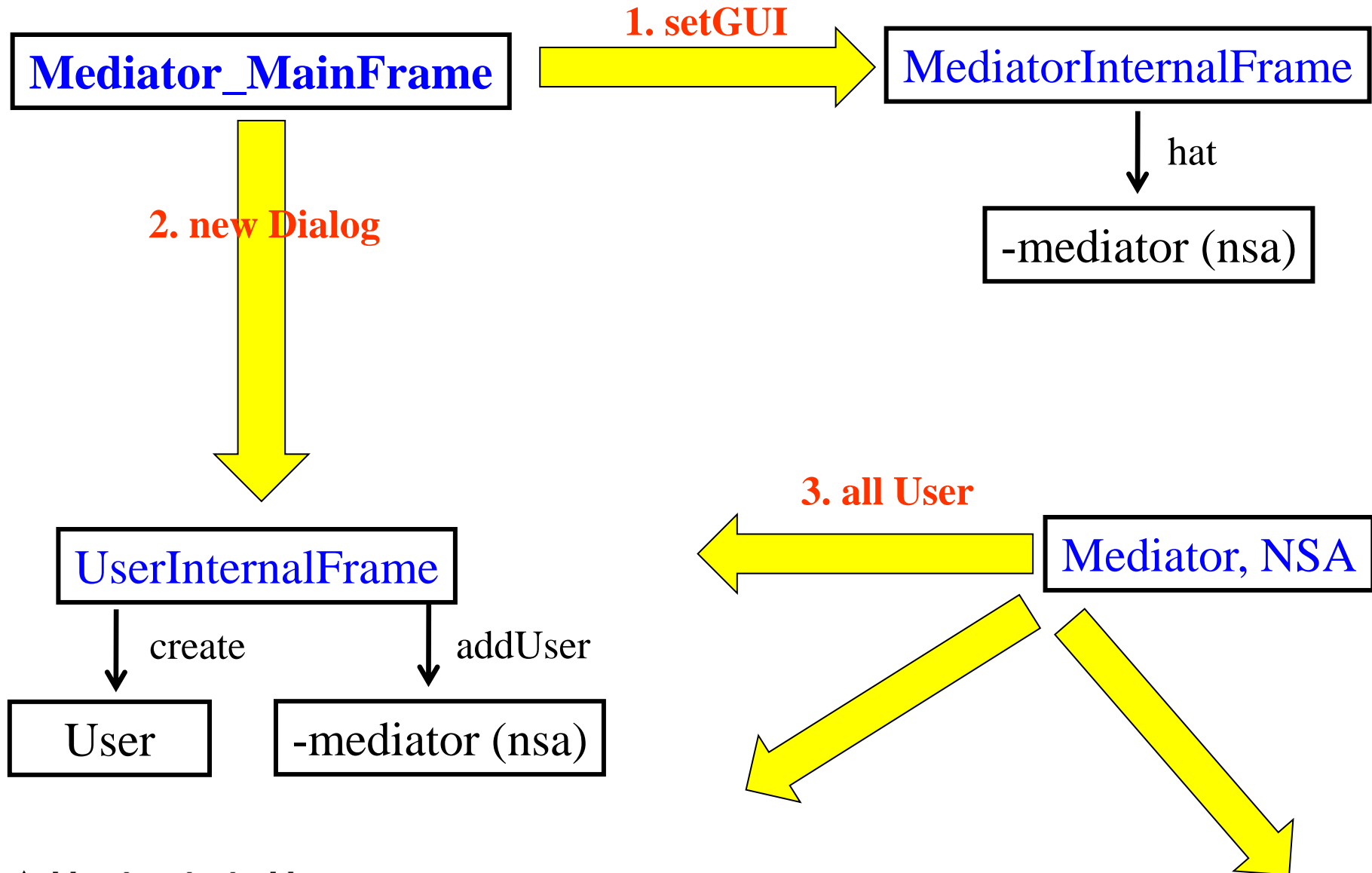
<http://www.journaldev.com/1730/mediator-design-pattern-in-java-example-tutorial>

# 2. Beispiel: Chat-Programm

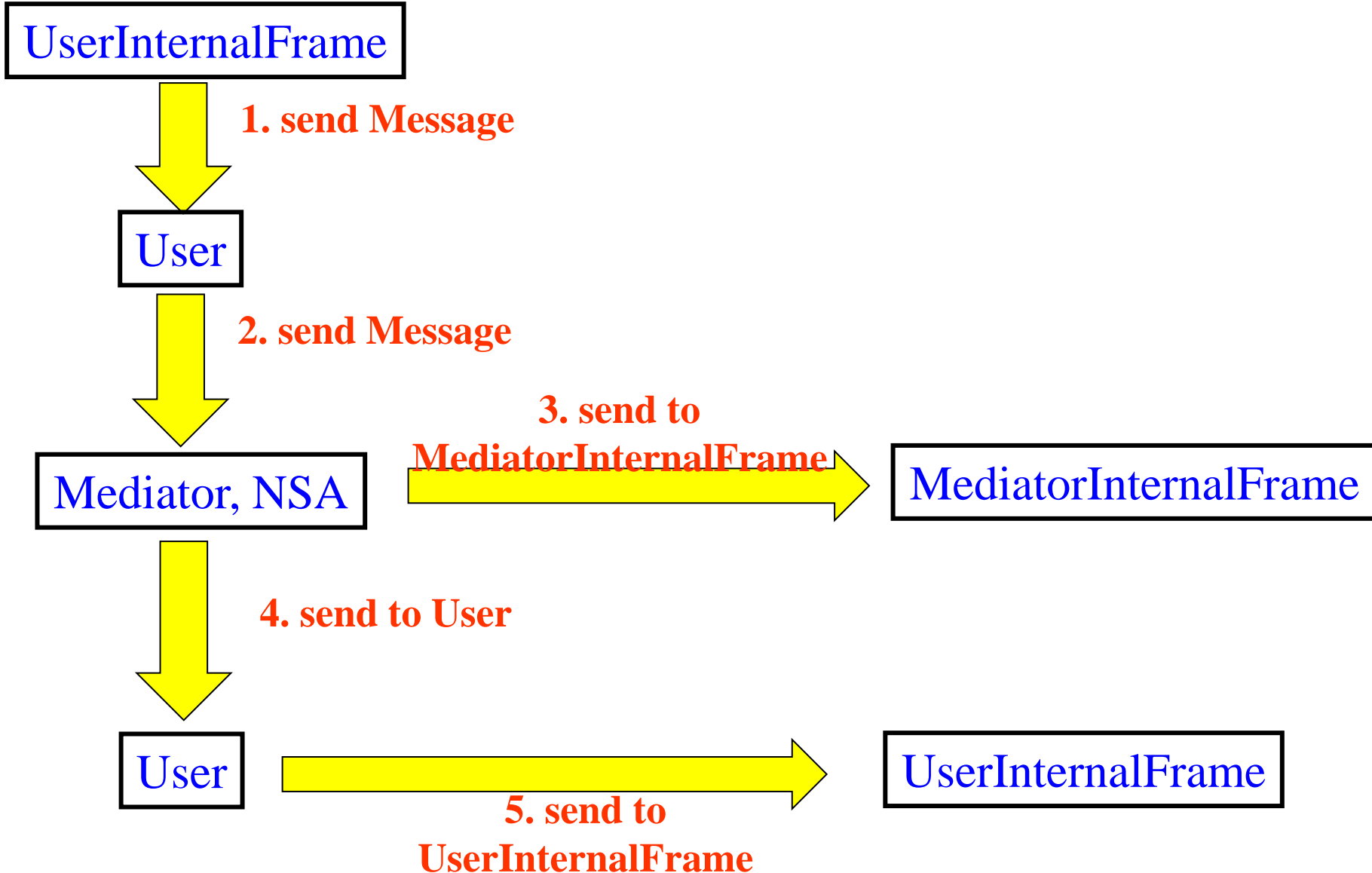


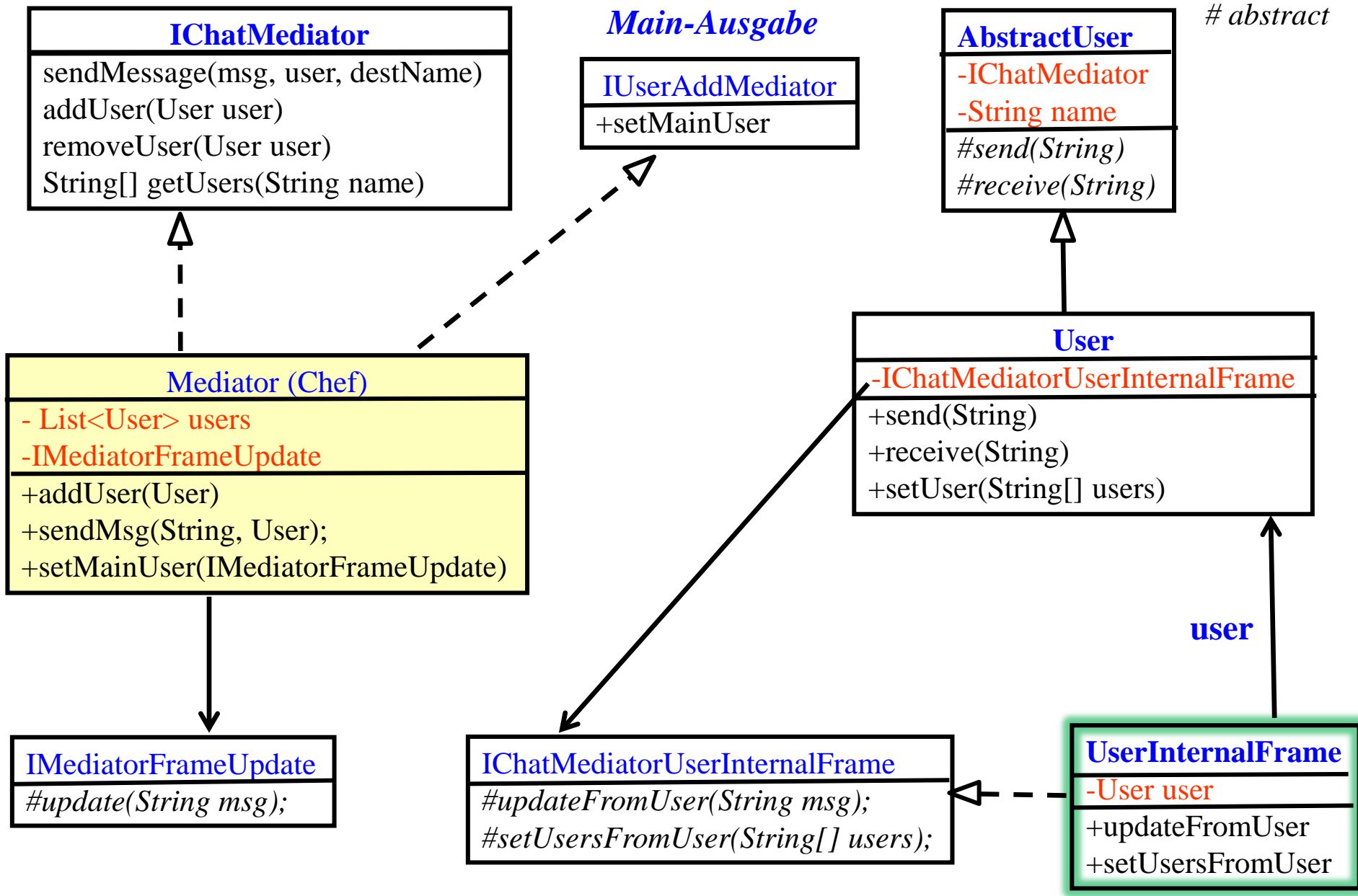


## 2. Beispiel: Aufrufverbindungen



## 2. Beispiel: Ablauf einer Sendmessage





# UML: MediatorInternalFrame

