

# Wiederverwendung komponentenbasierter Software

# Wiederverwendung - warum ?

- Softwareprojekte sind
  - langsam
  - teuer
- Vorteile für Hersteller und Kunden
- Übergang von der „freien“ Kunst zur Ingenieurskunst

# Voraussetzung zur Wiederverwendung

- Element ist
  - technischer Code (T)
  - Anwendungscode (A)
  - neutraler Code (0)
  - Codegenerator (R)
- Mischformen (AT) sind schwer wiederzuverwenden (z.B. JavaBeans)

# Elemente zur Wiederverwendung

- Algorithmen
- Klassen
- Komponenten
- Framework

# Elemente ... ctd

- Algorithmen
  - oft zu kleine Einheit
- Klassen
  - meist nicht möglich, da zu viele Klassen voneinander abhängig
- Framework
  - oft zu große Einheit
  - schlecht adaptierbar

# Komponenten

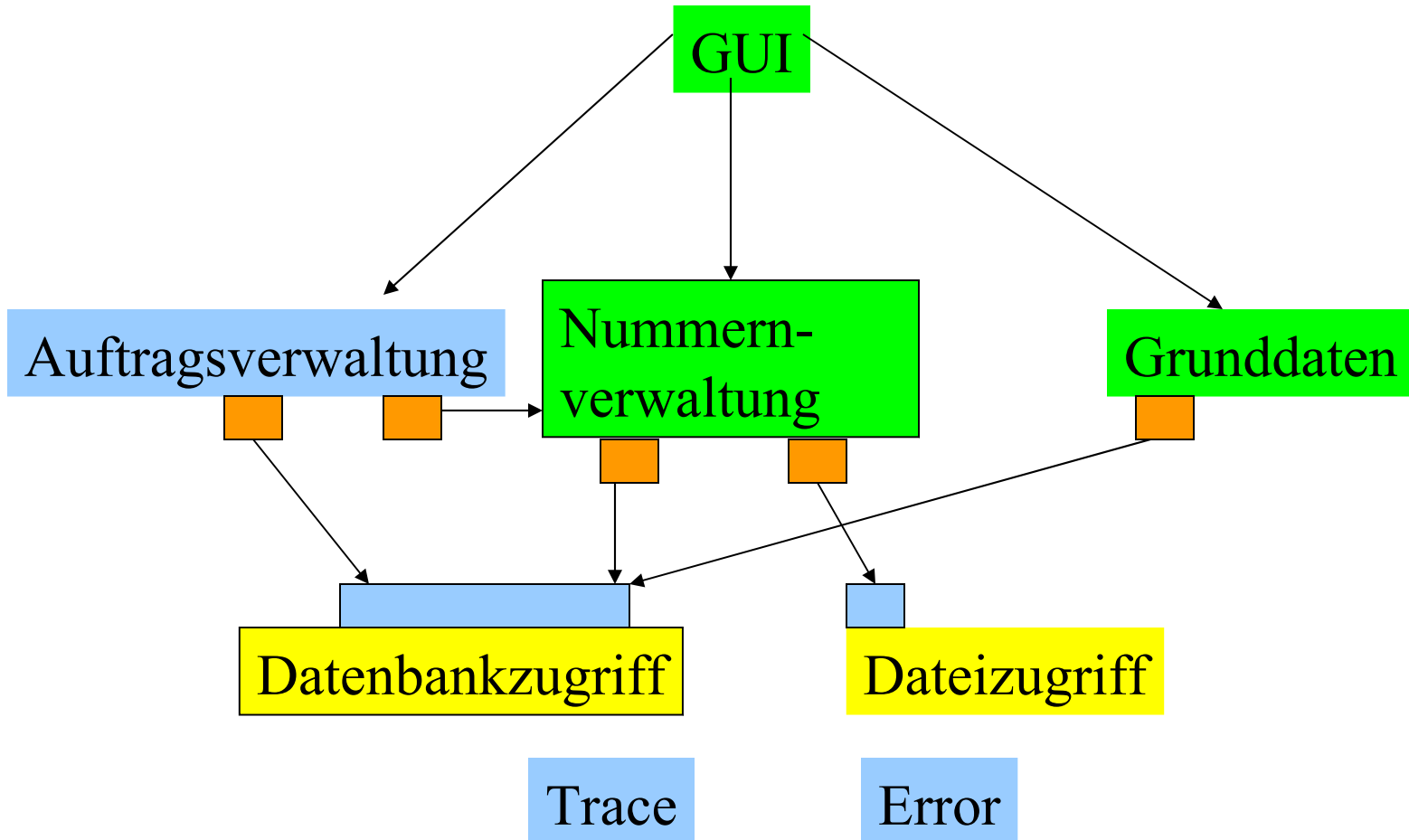
- funktional zusammengehörige Gruppe von Klassen
- werden über das Komponenteninterface angesprochen
- benützen andere Komponenten über Stützschnittstellen

# Komponenten (2)

bestehen aus

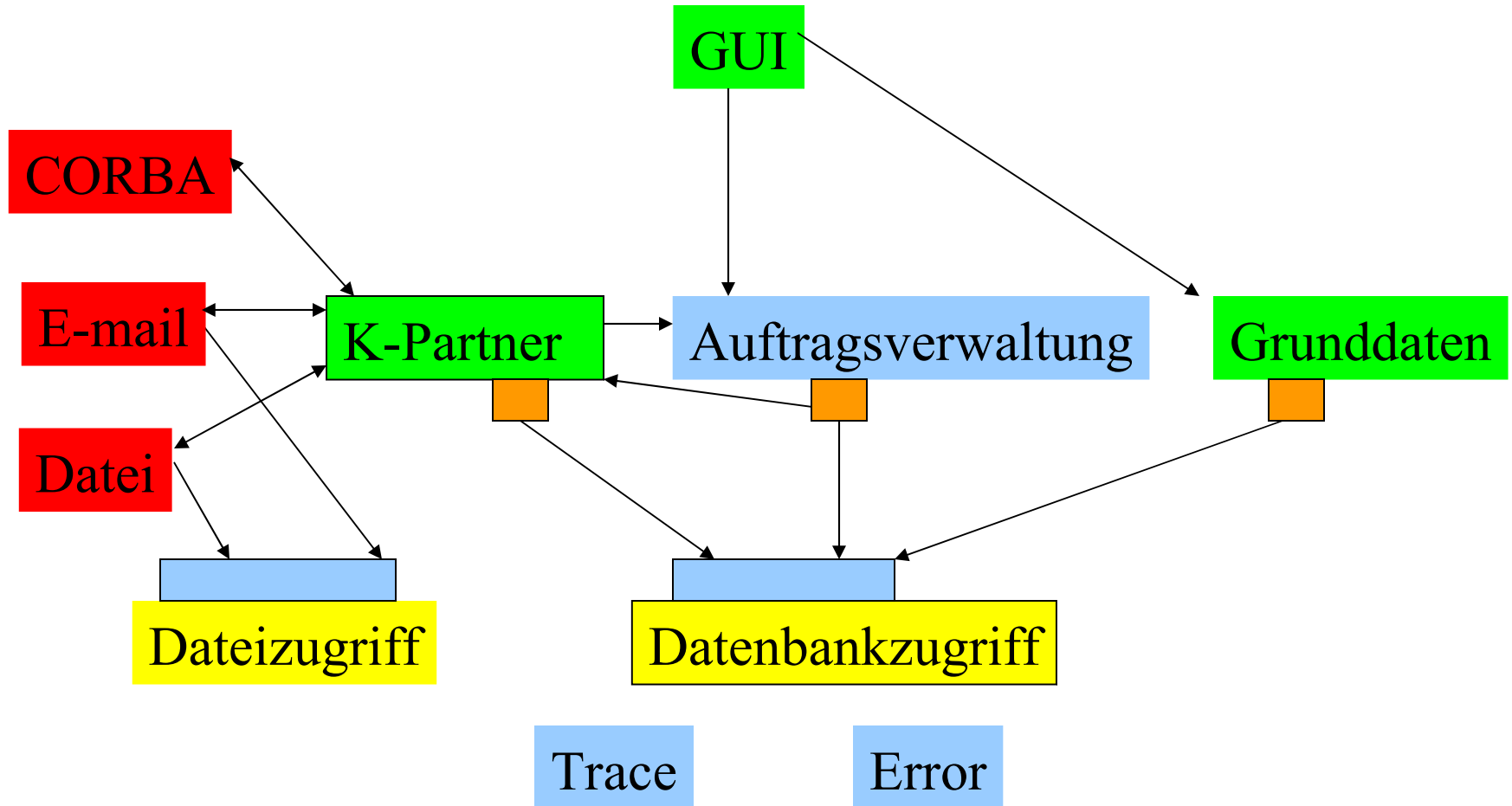
- Komponentenkern
  - technische Kapsel (z.B. Datenbank, GUI) (T)
  - Anwendungsproblem (z.B. Reisebuchung) (A)
  - generisches Problem (z.B. Kundenverwaltung, Auftragsverwaltung) (0)
- Adaptersoftware (R)
  - idealerweise generiert

# Rufnummernsystem





# Kommunikationssystem



# Resume

- Wiederverwendung war ausschlaggebend für die Gewinnung des Auftrags:
  - Durch Wiederverwendung konnte ein günstiges Angebot abgegeben werden
  - Mit der Abgabe des Angebots konnte eine erste Dummyversion gezeigt werden
  - Nach drei Wochen konnte zusammen mit dem ersten K-Partner getestet werden

# Literatur

- Brössler, Siedersleben „Softwaretechnik“, Hanser-Verlag 1999
- [www.wagnertech.de/Vorlesung](http://www.wagnertech.de/Vorlesung)