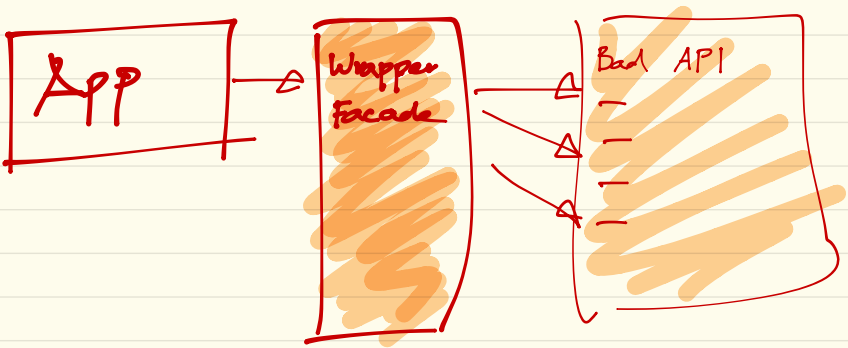
Wrapper Facade



- Kann auch mehr Logik enthalten
- Typensicherheit
- Legacy-Code verpacken für "Heute"
- Nachteile:
  Performance kann zum Problem werden
- Vorteile: Bessere API's
- Knackpunkte:
  Wrapper soll semantisch korrekt bleiben (zusammen was zusammen gehört

Anmerkungen P. Sommerlad:
- Error-Handling ist wichtig ⇒ Auch hier wrappen!
  ↳ Falls nötig Domain-spezifische Errors
- Wann nicht anwendbar?
  - Wrapper vom Wrapper vom Wrapper
- Stolperfalle:
  Async vs. Sync (Async ist schneller, da Bufferkopieren evtl. gespart werden kann)

Fault Tolerance Systems

Introduction: Zusammenhang Fault, Error & Failure

Fault: Bug, Ursache

Error: Zustand

Failure Effektives Problem

↳ Zu vermeidendes Problem

- Failure definieren sich im Normalfall durch Abweichung von der Spec

- Unterschiedliche Faults können zu gleichen Errors/Failures führen
- Coverage: Wahrscheinlichkeit dass sich ein System innert gegebener Zeit wieder erholen kann: Mean Time To Failure + Mean Time To Recover = Mean Time Between Failure

↳ Relaibility: $e^{-\frac{t}{MTTF}}$

- FIT: $\frac{\#\text{Failures}}{1 \cdot 10^9 h}$ ⇒ Failures in Time

⇒ Stichwort: Server-Zuverlässigkeit

Fail Silent: Bei Fehler übernimmt automatisch andere Komponente

Fail Consistency: Man muss herausfinden welche Systemkomponenten fehlerhaft sind

Malicious Failure: Man kann nicht einfach herausfinden welche Systeme fehlerhaft sind ⇒ Byzantinische Generäle zur Abstimmung