

# Laufzeitsicht



Stand: April 2020

© Diese Unterlagen sind urheberrechtlich geschützt von Dr. Peter Hruschka und Dr. Gernot Starke.  
Jede Verwendung außerhalb der engen Grenzen des Urheberrechts ist ohne schriftliche Zustimmung der Autoren unzulässig und strafbar.  
Dies gilt insbesondere für Vervielfältigungen, Übersetzungen sowie Speicherung und Verarbeiten in elektronischen Systemen.

[www.arc42.de](http://www.arc42.de)

A decorative blue curved line in the bottom right corner of the page.

# Ziele und Inhalt



## Sie lernen

- den Unterschied zwischen Bausteinen und Laufzeitobjekten kennen
- wissen, wie man Laufzeitszenarien dokumentiert
- Szenarien auf verschiedenen Abstraktionsebene entwickeln können

## Lernziele gemäß iSAQB CPSA-F:

- LZ 2-2: Softwarearchitekturen entwerfen (R1)
- LZ 3-2: Softwarearchitekturen beschreiben und kommunizieren (R1)
- LZ 3-3: Notations-/Modellierungsmittel für Beschreibung von Softwarearchitektur erläutern und anwenden (R2)
- LZ 3-8: Architekturentscheidungen erläutern und dokumentieren (R2)
- LZ 3-9: Dokumentation als schriftliche Kommunikation nutzen (R2)
- LZ 3-10: Weitere Hilfsmittel und Werkzeuge zur Dokumentation kennen (R3)

## Sie üben

- die Erstellung von Laufzeitmodellen



# Laufzeitsicht

Zeigt das Zusammenspiel von (Instanzen von) Bausteinen zur Laufzeit

## Szenarien

- Besonders wichtig für objektorientierte Systeme, wo Bausteine mehrere Instanzen haben können
- bzw. für Embedded Real-Time Systems, wo Prozesse (Tasks, Threads, ...) neben den Bausteinen eine große Rolle spielen
- Zur Modellierung dynamischer Aspekte wie Lastverteilung, kritische Kommunikation, Einhaltung von vorgegebenen Protokollen, ...

## Beispielhafte Abläufe

# Zweck der Laufzeitsicht

1. Entdecken vergessener Bausteine
2. Verbessern/erweitern/modifizieren vorhandener Bausteine
  - Mehr Funktionalität
  - Andere Schnittstellen
  - ...
3. Verifizieren vorhandener Bausteine
4. Erläutern der Architektur anhand von beispielhaften Abläufen



# Entwicklung von Laufzeitszenarien

Wie sucht man Laufzeitaspekte aus?

- Oft nützlich: System-Start-up-Verhalten, Bootstrapping
- Immer nützlich: Der Normalfall der wichtigsten Use-Cases
  
- Dann:
  - Stress-Szenarien
  - Grenzszenarien
  - Szenarien für wichtige Qualitätsmerkmale (-> Ziele der Architektur)
  
- Nicht immer „vollständige Szenarien“ (quer durch das System)
  - auch Verhalten und Umgang mit externen Partnern (Schnittstellenabstimmung)
  - Zusammenspiel kritischer Komponenten, spezielle Protokolle

# Bausteinsicht vs. Laufzeitsicht

- + Vollständige Modelle  
(Abstraktionen des gesamten Source Codes mit adäquatem Tiefgang)
- + Alle möglichen Services und alle Schnittstellen
- Zusammenhänge sind schwer zu sehen

## Passende UML-Notation

- Klassenmodelle,  
Paketdiagramme,  
Komponentendiagramme

- + wesentliche Laufzeitaspekte werden hervorgehoben und exemplarisch diskutiert und analysiert
- beispielhafte Abläufe (Szenarien), nie alle Möglichkeiten

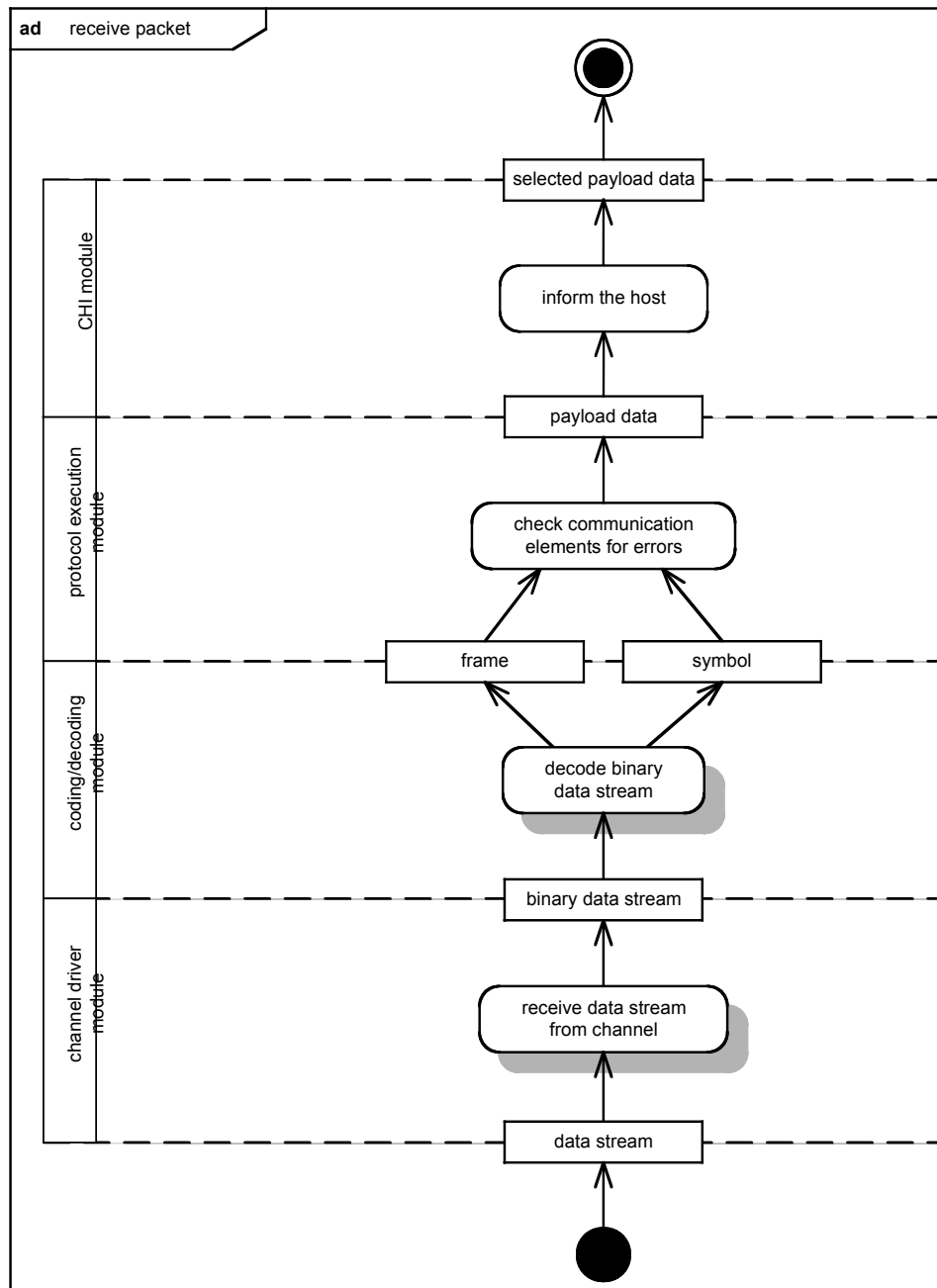
### Weitere Notationen:

- Flussdiagramme, BPMN, EPKs
- Nummerierte Listen
- ...

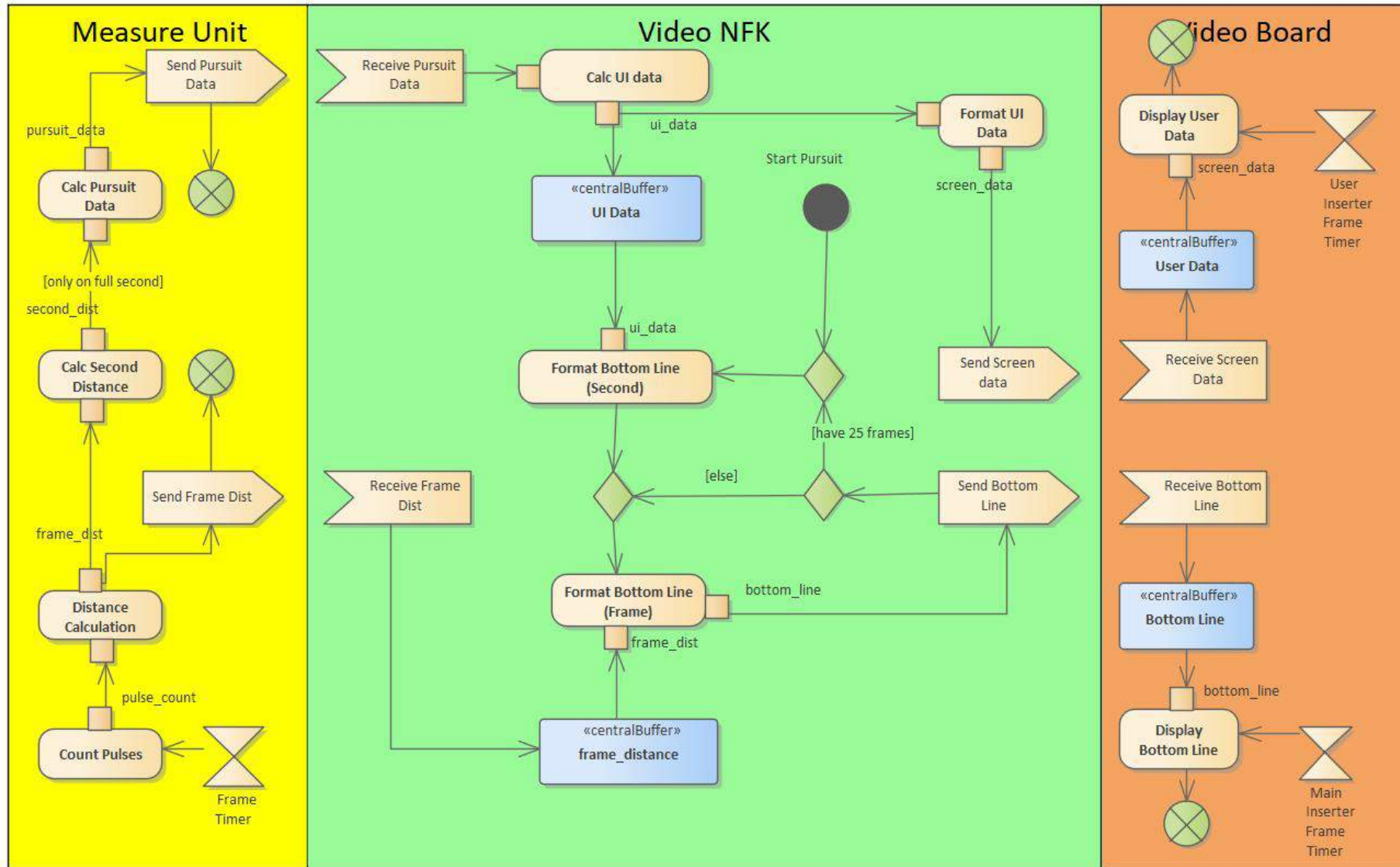
## Passende UML-Notation

Aktivitätsdiagramme (mit Swimlanes),  
Sequenzdiagramme,  
Kommunikationsdiagramme,  
Interaktionsübersichtsdiagramme,  
Timing Diagrams

# Aktivitätsdiagramm mit Bahnen (Swimlanes)



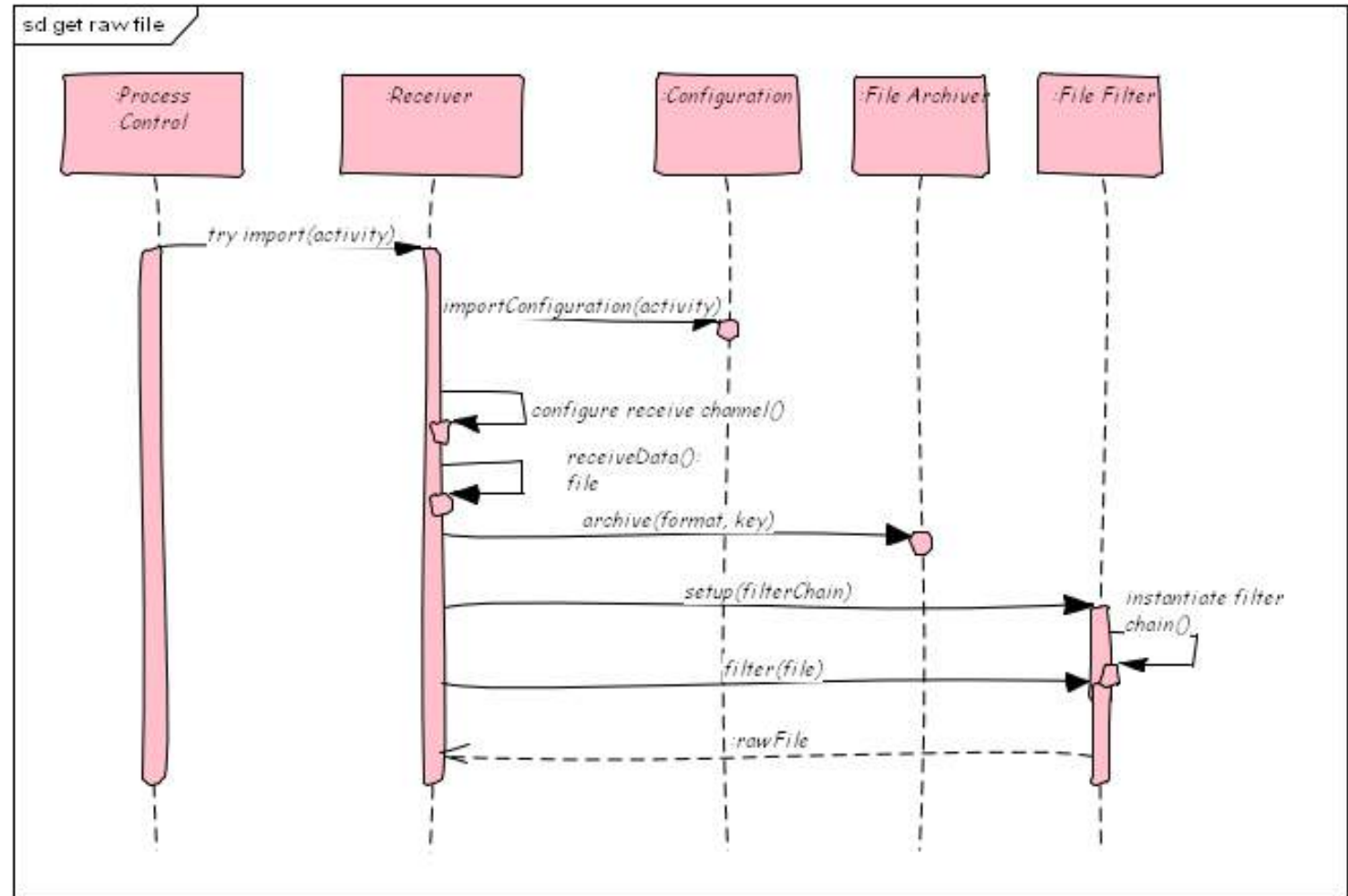
# act AD Frame Data Dispatch-Extended



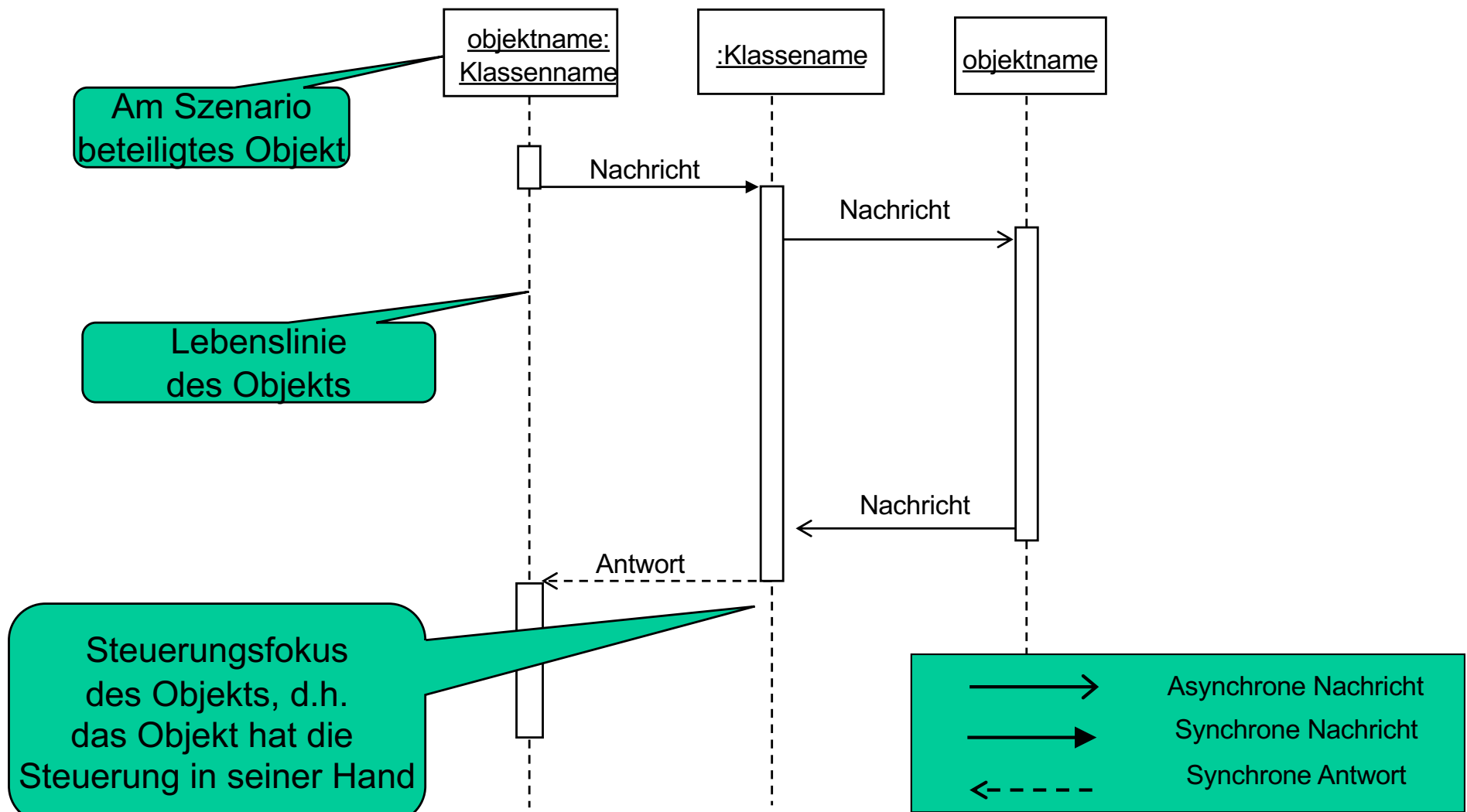


# Sequenzdiagramme ...

... zeigen Ereignisse in ihrer zeitlichen Reihenfolge auf (bzw. die Nachrichten), die zwischen Objekten ausgetauscht werden,

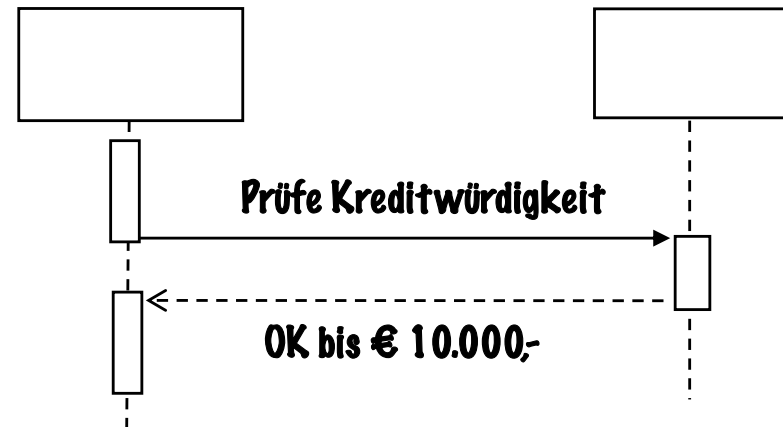


# Elemente von Sequenzdiagrammen

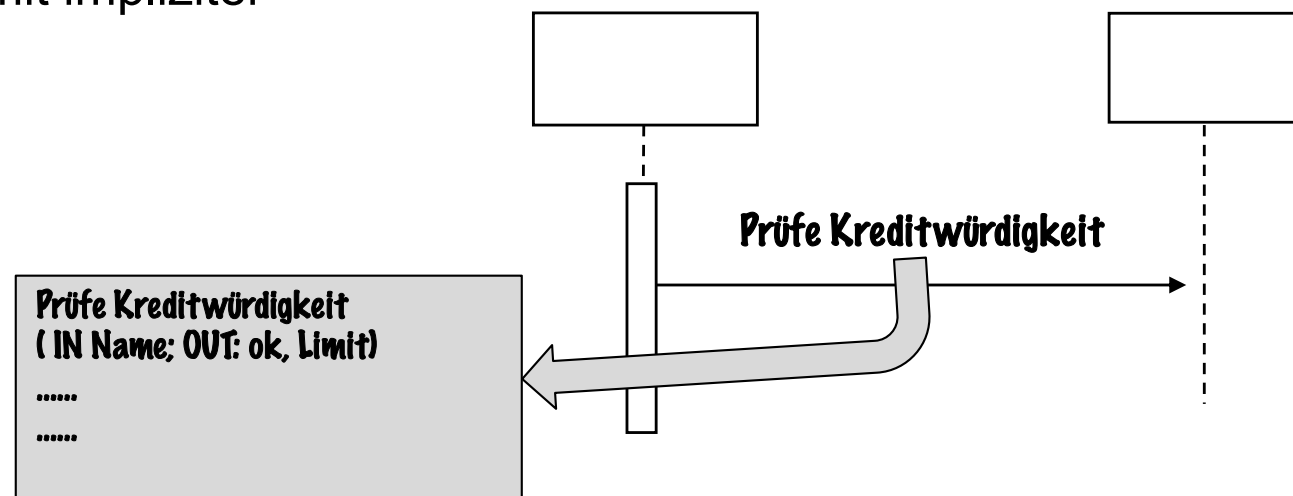


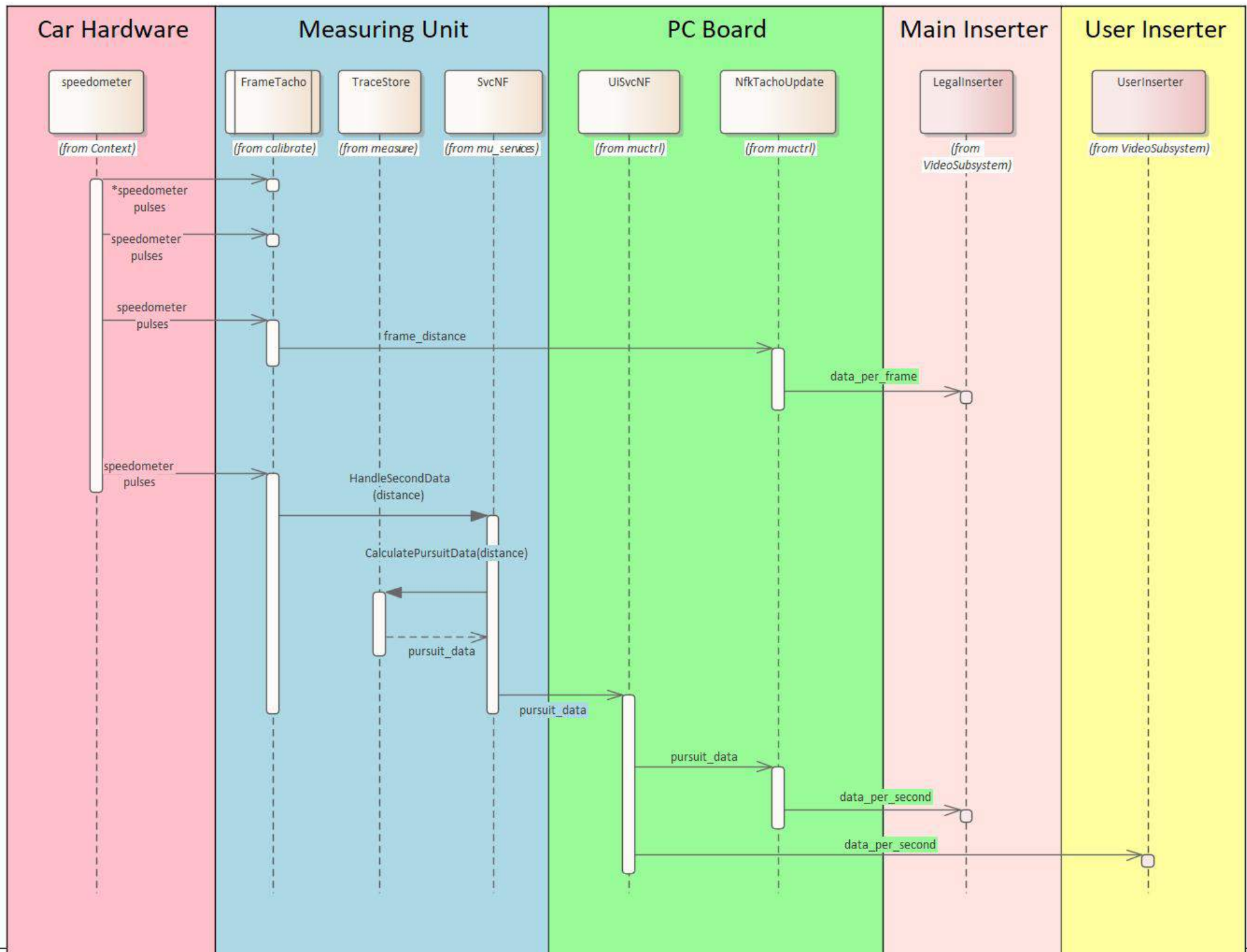
# Stil für synchrone Nachrichten

- Nachrichten und Antworten



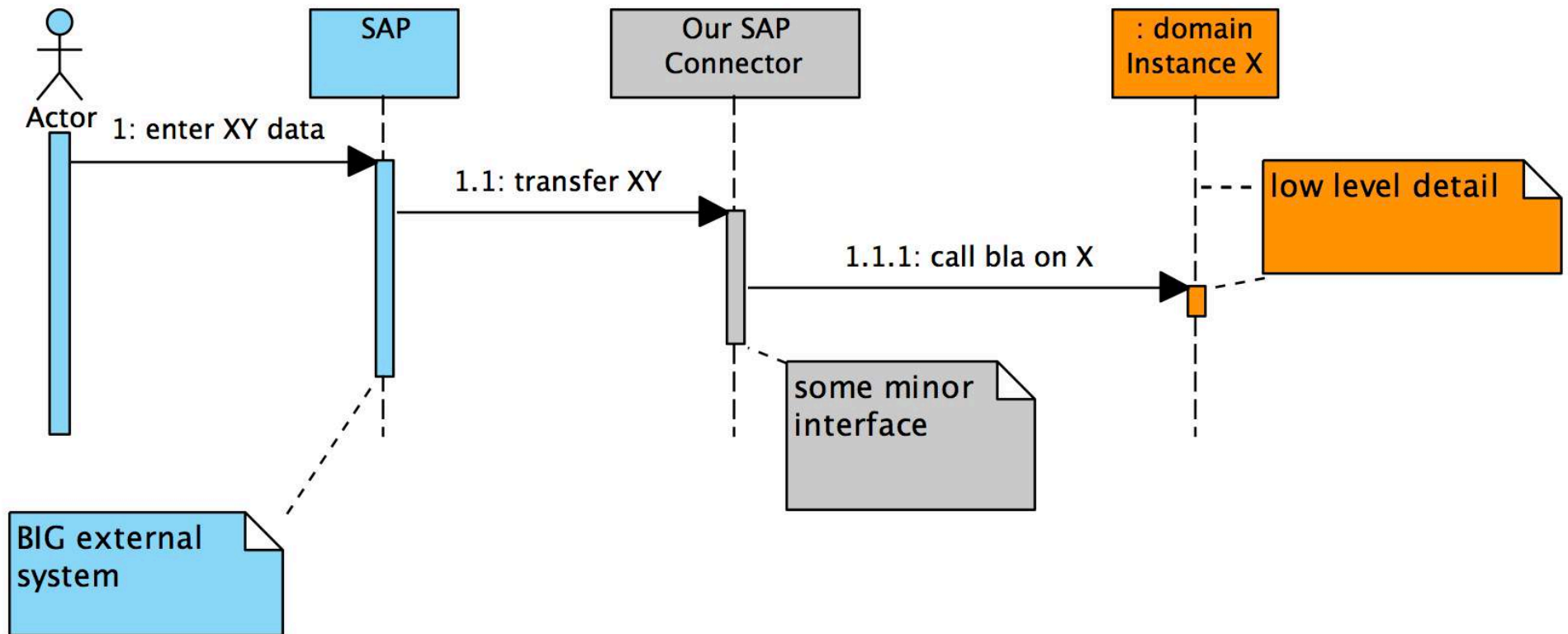
- Nachrichten mit impliziter Antwort



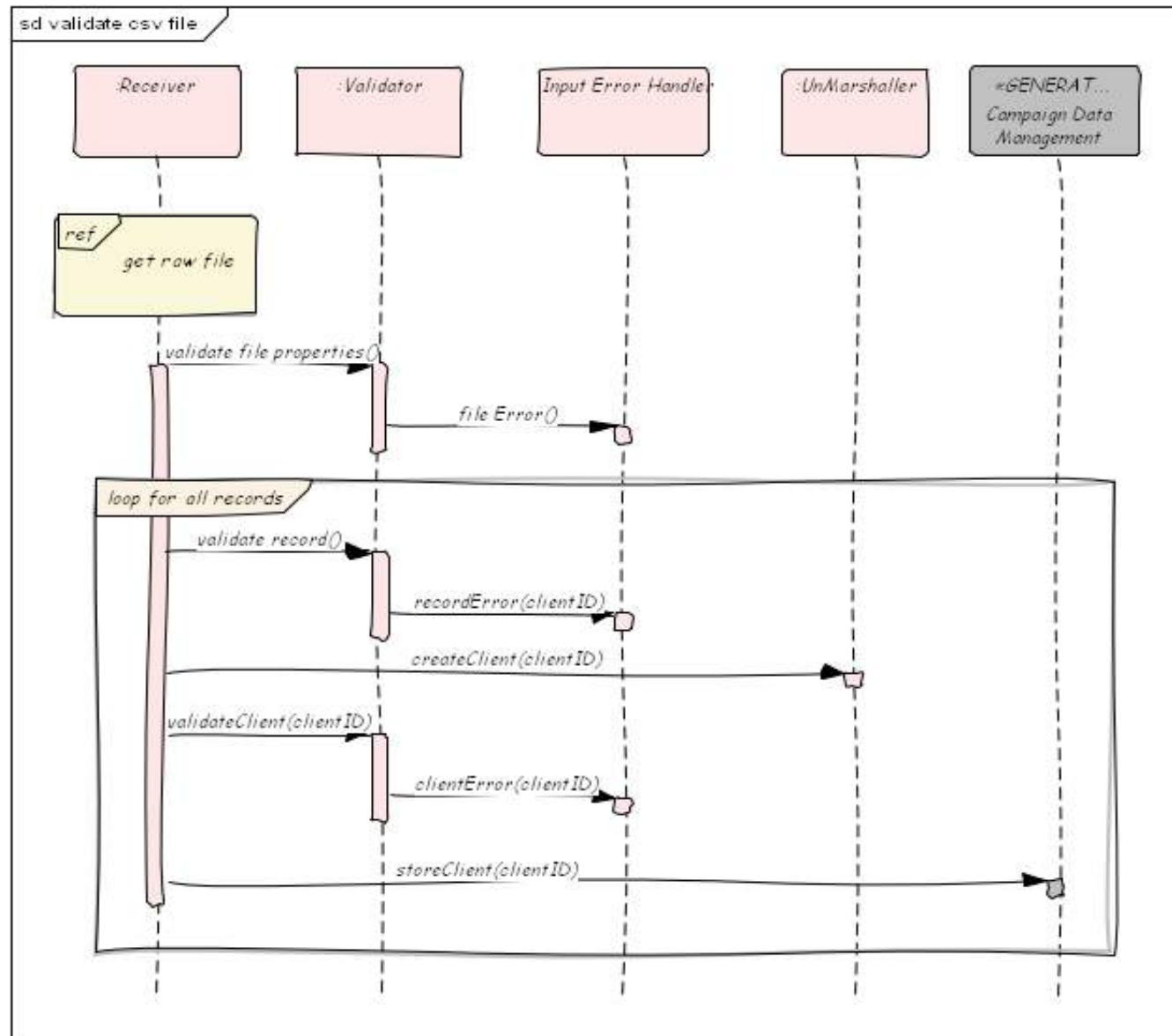


# Verschiedene Ebenen verwenden

## sd Different-Level-Illustration

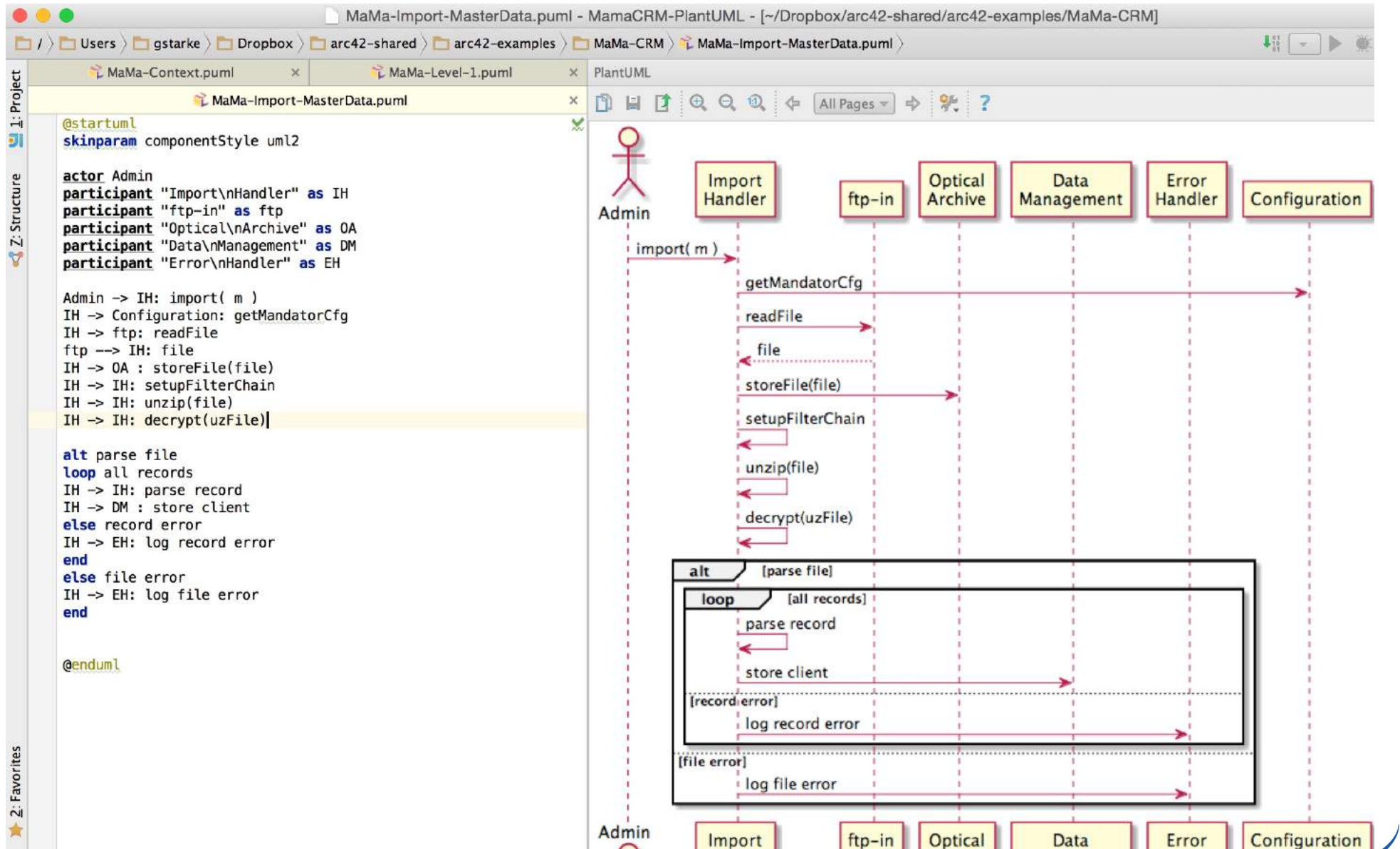


# Sequenzdiagramm (UML2): Schachtelung von Teilabläufen



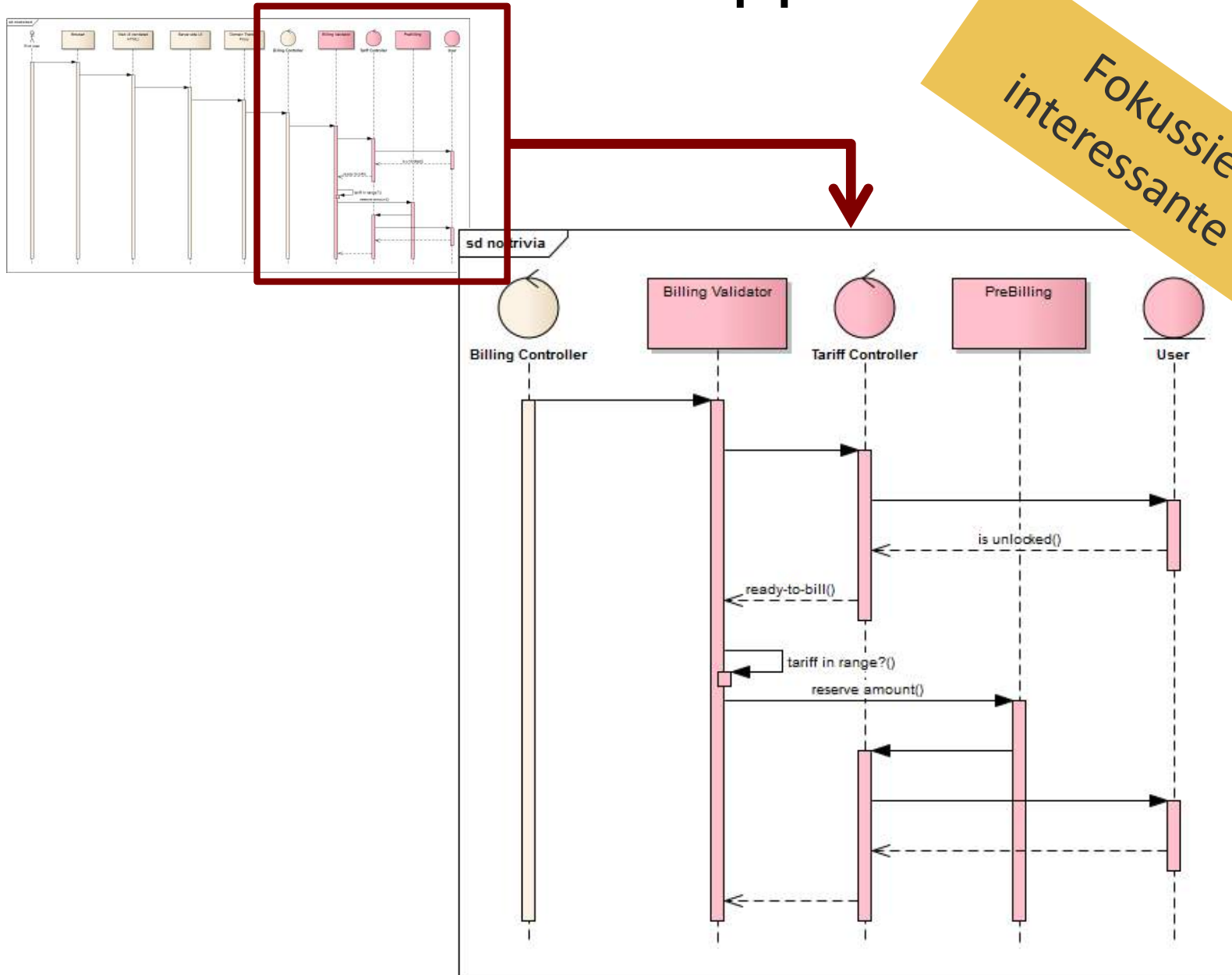
# Szenarien eher zur Diskussion...

weniger zur Dokumentation





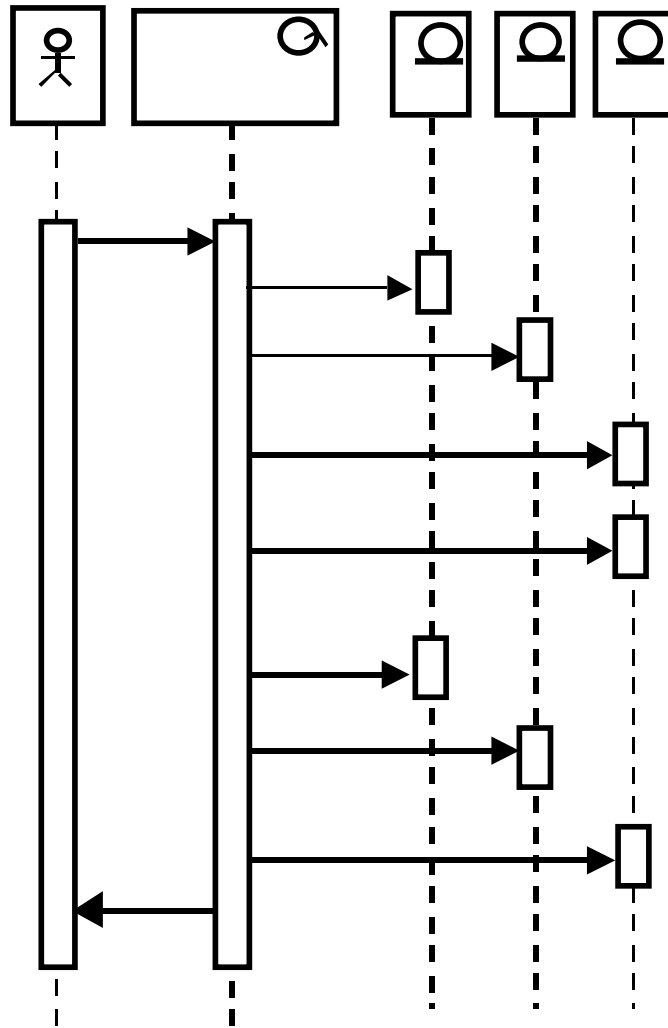
# Praxistipps Laufzeitsicht (2)



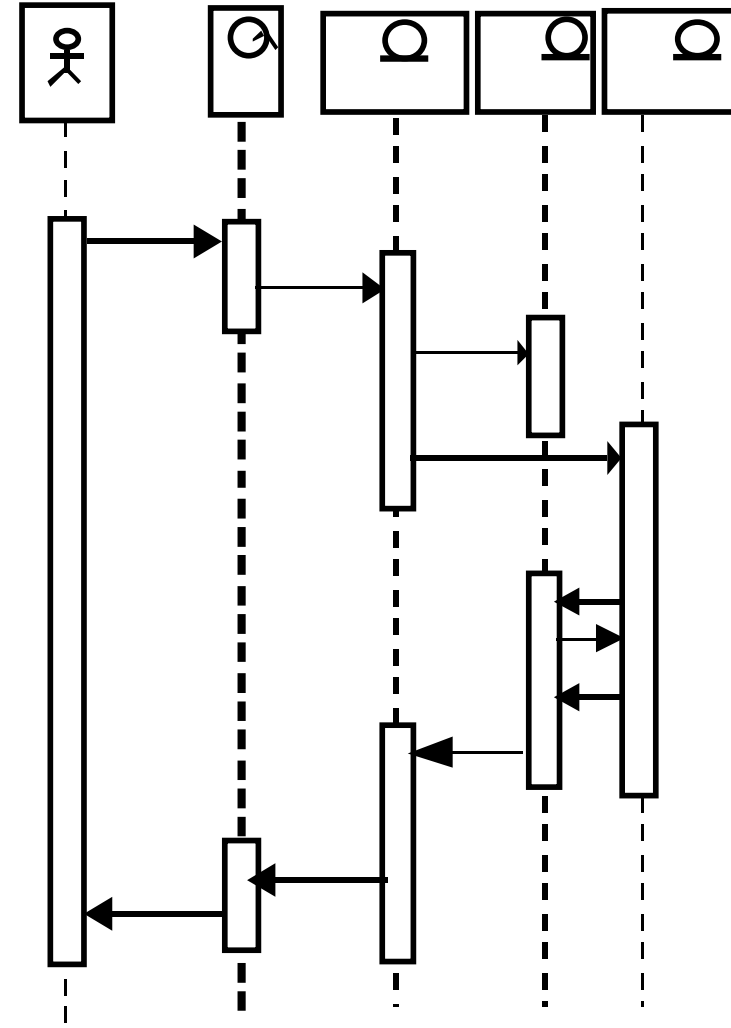
Fokussiere auf interessante Teile...



# Stile der Zusammenarbeit



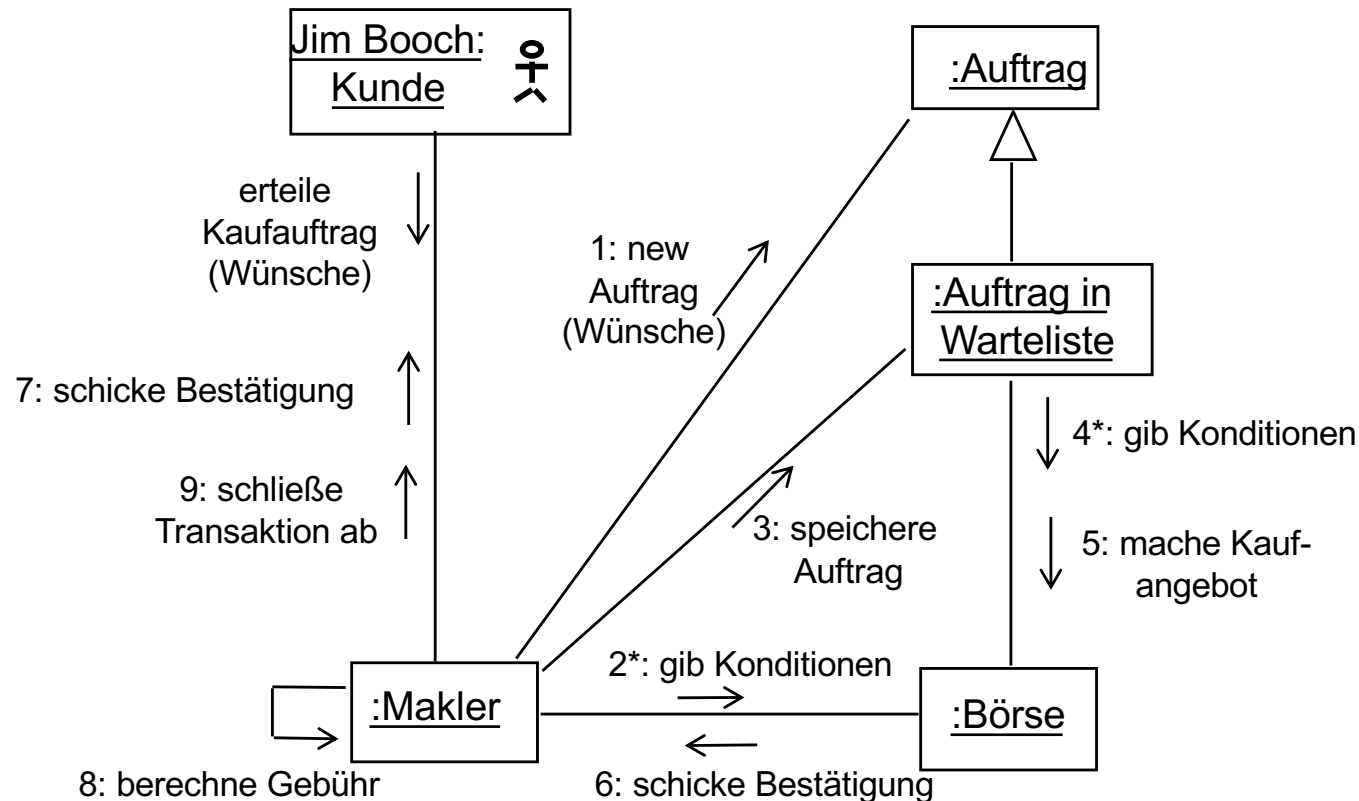
Wenig Delegation

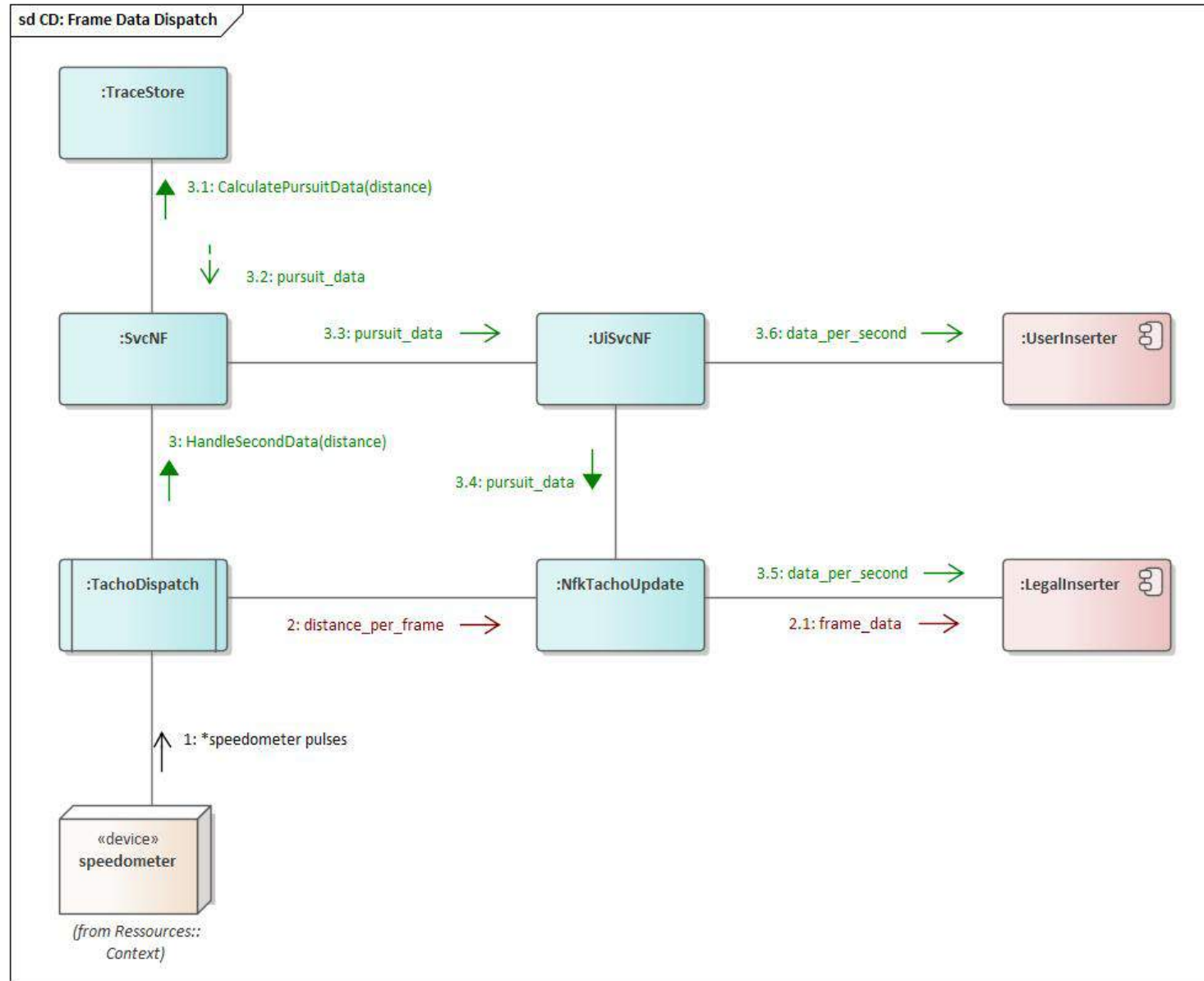


Viel Delegation

# Kommunikationsdiagramm = Objektdiagramm mit Nachrichten

- Beispiel Börsensystem:





# Wohin mit den Ergebnissen?

## 1. Einführung und Ziele

- 1.1 Aufgabenstellung
- 1.2 Qualitätsziele
- 1.3 Stakeholder

## 2. Randbedingungen

- 2.1 Technische Randbedingungen
- 2.2 Organisatorische Randbedingungen
- 2.3 Konventionen

## 3. Kontextabgrenzung

- 3.1 Fachlicher Kontext
- 3.2 Technischer- oder Verteilungskontext

## 4. Lösungsstrategie

## 5. Bausteinsicht

- 5.1 Ebene 1
- 5.2 Ebene 2
- ....

## 6. Laufzeitsicht

- 6.1 Laufzeitszenario 1
- 6.2 Laufzeitszenario 2
- ....

## 7. Verteilungssicht

- 7.1 Infrastruktur Ebene 1
- 7.2 Infrastruktur Ebene 2
- ....

## 8. Konzepte

- 8.1 Fachliche Struktur und Modelle
- 8.2 Typische Muster und Strukturen
- 8.3 Persistenz
- 8.4 Benutzeroberfläche
- ....

## 9. Entwurfsentscheidungen

- 9.1 Entwurfsentscheidung 1
- 9.2 Entwurfsentscheidung 2
- ....

## 10. Qualitätsszenarien

- 10.1 Qualitätsbaum
- 10.2 Qualitäts-/Bewertungsszenarien

## 11. Risiken

## 12. Glossar



# Zusammenfassung



- Nutzen Sie Szenarien zum Finden fehlender Bausteine und zum Verbessern existierender Bausteine
- Nutzen Sie Szenarien auch zu Kommunikation Ihrer Architektur
- Nutzen Sie Abläufe/Szenarien auf allen Abstraktionsebenen
- Achten Sie darauf, dass Laufzeitschritte von Bausteinen verantwortet werden
- Nehmen Sie nur so viele Szenarien in Ihre Doku auf, wie Sie konsistent halten können!
- Beliebte Notationen für Szenarien:
  - Aktivitätsdiagramme
  - Sequenzdiagramme
  - Flußdiagramme
  - Nummerierte Listen (Textuelle Beschreibungen)