# Ziele

**Kontextabgrenzung von Systemen erläutern und anwenden**

Softwarearchitekten können:

* Kontext von Systemen z.B. in Form von Kontextdiagramm mit Erläuterungen darstellen
* externe Schnittstellen von Systemen in der Kontextabgrenzung darstellen
* fachlichen und technischen Kontext differenzieren.

# Kontextabgrenzung

Quelle Basiswissen für Softwarearchitekten, 4. Auflage, Kapitel 4.3.4

## Inhalte der Kontextabgrenzung

Die Kontextsicht (auch: Kontextabgrenzung) ist ein wesentliches Bindeglied zwischen der textuellen oder grafischen Anforderungsbeschreibung und der späteren Architektur. Sie beschreibt das Umfeld eines Systems und die Beziehungen bzw. Zusammenhänge mit diesem Umfeld und dient damit allen Beteiligten als Einstiegspunkt und Landkarte für das zu beschreibende System.

Bei der Darstellung der Kontextsicht liegt somit der Schwerpunkt auf Schnittstellen zu den umliegenden Systemen (Nachbarsysteme). Für eine weiter gehende Beschreibung der fachlichen, technischen und organisatorischen Aspekte der Umsetzung dieser Schnittstellen soll auf das entsprechende Schnittstellenkonzept verwiesen werden.

Für die Kontextsicht sind folgende Elemente von Bedeutung:

* Externe Akteure (Nachbarsysteme und Benutzer)
* Das zu entwickelnde System selbst
* *Alle* Schnittstellen zu externen Akteuren (alle Nachbarsysteme bzw. Benutzer) einschließlich:
	+ Art der Schnittstelle: z.B. Online, Batch, USB oder Datei sowie die über diese Schnittstelle übertragenen Daten oder Ressourcen sowie ggf. genutzte Services oder Funktionen,
	+ verwendete Kommunikationsprotokolle sowie
	+ verwendete Kommunikationsmuster, z.B. synchron, asynchron.

Die Kontextsicht grenzt auf dieser Basis den Bereich des zu betrachtenden Softwaresystems ab.

## Beschreibungselemente der Kontextsicht

Beschreibungen der Kontextsicht erfolgen vor allem durch

* Kontextdiagramme und
* Listen von Nachbarsystemen mit deren Schnittstellen.

Die Kontextsicht kann (und wird häufig) sowohl fachlich als auch technisch sein. Damit kann ein zu beschreibendes Softwaresystem fachlich zu anderen Softwaresystemen abgegrenzt werden, aber auch technisch in bestehende oder zu beschaffende Infrastrukturen eingegliedert werden.

Besonders der fachliche Teil einer Kontextsicht kann statisch, aber auch dynamisch sein. Da die dynamische Sicht eher für Test und Betrieb, die statische Sicht dagegen eher für Architektur, Design und Entwicklung geeignet ist, konzentrieren wir uns auf die statische Sicht.

Durch die Kontextsicht muss sichergestellt werden, dass die Schnittstellen mit allen relevanten Aspekten (Was wird übertragen? In welchem Format wird übertragen? Welches Medium wird verwendet? Etc.) spezifiziert wird, auch wenn einige populäre Diagramme (wie z.B. das UML-Use-Case-Diagramm) nur ausgewählte Aspekte der Schnittstelle darstellen.

Diagramme der Kontextsicht werden in vielen Notationen erstellt. Wenn UML-Diagramme eingesetzt werden, dann sind für fachlich orientierte Diagramme dieser Sicht UML-Komponenten- und UML-Kompositionsstrukturdiagramme besonders sinnvoll. In dem Fall bietet es sich an, das zu beschreibende System mit wohldefinierten Schnittstellen als »Blackbox in der Mitte« anzusiedeln.

Bei technisch orientierten Kontextsichten können Sie UML-Komponenten auch gut durch UML-Knoten ergänzen. Hinzu kommen bisweilen UML-Paketsymbole. Auch Netzwerksymbole (z.B. aus Visio), die außerhalb der UML-Notation liegen, werden gerne genutzt. Andere eher informelle Notationsformen im

»Boxes and Arrows«-Stil von PowerPoint werden ebenfalls des Öfteren in der Kontextsicht eingesetzt.

Wichtig ist, dass die jeweilige Darstellung für die Kommunikation mit Ihren primären Stakeholdern gut passt. In der Kontextsicht werden abstrakte Darstellungen der Baustein-, Laufzeit und Verteilungssicht gezeigt, daher dürfen ergänzend auch deren jeweilige Notationen und Diagramme verwendet werden.

In der Regel enthält die Kontextsicht folgende Elemente

|  |  |
| --- | --- |
| Typ | Beschreibung |
| UML-KomponenteUML-Part | UML-Komponenten und UML-Parts stellen als wesentliche Top-Level- Elemente Bausteine dar, für die klare (ggf. extern sichtbare) Schnittstellenssenziell sind. Sie sind die wichtigsten Symbole der Kontextsicht. |
| UML-Knoten | Speziell für eine (ergänzende) Darstellung des (technischen) Verteilungs-bzw. Infrastrukturkontexts des zu beschreibenden Softwaresystems können auch UML-Knotensymbole genutzt werden. Vor allem zur Verbindung von Knoten dürfen hier neben UML-Abhängigkeitsbeziehungen auch UML-Assoziationen genutzt werden. |
| UML-Akteur | Zur Darstellung des Bezugs des zu beschreibenden Softwaresystems zu wichtigen Benutzerrollen wird der Typ des UML-Akteurs genutzt. |
| Schnittstellen zur Außenwelt(»Abhängigkeitsbeziehung«) | Diese dienen der Darstellung des Daten- oder Kontrollflusses zwischen den Systemen und Stakeholdern der Außenwelt. Verwenden Sie UL Beziehungen (»Abhängigkeiten«). Diese beinhalten bei Bedarf Informationen über Schnittstellenart, Kommunikationsprotokolle, Kommunikationsmuster und übertragene Objektart. Jede Schnittstelle sollte in der Kontextsicht einen aussagekräftigen Namen tragen. |
| Legende/Kommentar | Verbale Legenden bzw. Erläuterungen erscheinen als Kommentar imDiagramm. |

## Technische und fachliche Kontextabgrenzung

Die Kontextabgrenzung existiert in zwei Ausprägungen, dem fachlichen Kontext (unbedingt erforderlich) und dem technischen Kontext (optional). In beiden Fällen können Sie zur übersichtlichen Darstellungm des Systems in seiner Umwelt Kontextdiagramme verwenden,

### Fachlicher Kontext

Der fachliche Kontext beschreibt die Funktion und das Umfeld des Systems aus einer geschäftlichen oder fachlichen Perspektive (business context). Oftmals finden Sie im fachlichen Kontext Verweise auf die wichtigsten Anwendungsfälle (use cases) der Anforderungsanalyse.

Die Kontextabgrenzung sollten Sie zumindest statisch beschreiben z. B: mit einem UML-Paket- oder Komponentendiagrammen. Falls Sie hier (ergänzend zu den Anforderungsdokumenten) auch Abläufe zwischen den Nachbarsystemen und Ihrem System zeigen möchten, eignen sich beispielsweise Sequenzdiagramme.

Abbildung Fachliche Kontextabgrenzung

### Technischer Kontext

Als Ergänzung zum fachlichen Kontext können Sie hier technische Kanäle und/oder Übertragungsmedien zwischen dem System, seinen Nachbarn und deren Umgebung beschreiben. Verwenden Sie UML Verteilungsdiagramme mit ergänzenden Erläuterungen der dargestellten Elemente.

Die Beschreibung des technischen Kontextes ist optional, da diese Darstellung der technischen Infrastruktur auch in der Verteilungssicht aufgezeigt wird. Insbesondere in heterogenen Systemumgebungen halten wir den technischen Kontext jedoch für sehr wichtig.

Abbildung Technische Kontextabgrenzung

Jetzt müssen Sie noch die fachliche und technische Kontextabgrenzung miteinander in Beziehung setzen, indem Sie die fachlichen Schnittstellen den technischen Kanälen („Leitungen“) zuordnen. Das geht mit einer Tabelle recht einfach – hier ein Beispiel:

|  |  |
| --- | --- |
| Fachliche Schnittstelle | Technischer Kanal |
| Kundendaten | Gigabit Ethernet (sftp) |
| Reports | Reports Internet ADSL (http) |
| Status | Wlan 802.11.g (http) |
| Aufrufe | Wlan 802.11.g (http) |

Quelle Softwarearchitektur kompakt, 2. Auflage