Masterarbeit

Eine einheitliche Middleware zur Policy-Durchsetzung in Multi-Cloud-Infrastrukturen

A Unified Middleware for Policy Enforcement in Multi-Cloud Infrastructures

von

Jan-Henrich Mattfeld

Betreuung

Max Plauth, Prof. Dr. Andreas Polze

Fachgebiet für Betriebssysteme und Middleware

Hasso-Plattner-Institut an der Universität Potsdam

22. Dezember 2017

Cloud-Angebote sind allgegenwärtig und werden mittlerweile von der Mehrzahl deutscher Unternehmen genutzt. Laut Statista setzten bis 2016 mindestens fünfundsechzig Prozent aller Betriebe entsprechende Lösungen ein. Besonders gefragt sind Infrastrukturdienste, wie Rechenleistung und Speicher. Gleich darauf folgen Softwareangebote und E-Mail-Hosting. Etwas abgeschlagen bleiben Plattformdienste wie Datenbanken und Ausführungsumgebungen. Der Trend zur Cloud wird sich vermutlich fortsetzen: So vervierfacht sich das prognostizierte Marktvolumen mit Cloud-Services bis 2020 auf über sechzehn Milliarden Euro allein im deutschen B2B-Markt.

Die Cloud-Angebote sind für Unternehmen aller Größen attraktiv: Die Anschaffung eigener Infrastruktur entfällt, genauso wie deren Wartung durch eigenes Personal. Stattdessen lassen sich Ressourcen und Anwendungen einfach per Self-Service buchen und sind anschließend über das Internet von überall erreichbar. Der Umfang der gebuchten Leistungen lässt sich meist frei skalieren. Da die Angebote oft in kleinem Takt und verbrauchsgenau abgerechnet werden, ergeben sich so theoretisch Vorteile bei Flexibilität und Kosteneffizienz.

Demgegenüber stehen Vorbehalte bezüglich Datenschutz, denn die gemietete Infrastruktur teilen sich mehrere Kunden. Zugleich fordert aktuelle Gesetzgebung wie die Datenschutz-Grundverordnung unter anderem die Verarbeitung von personenbezogenen Daten europäischer Kunden ausschließlich innerhalb der EU. Gut neunzig Prozent aller Unternehmen achteten dementsprechend bei der Auswahl eines Cloud-Providers auf Rechts- und Server-Standorte in Deutschland. Für gut fünfzig Prozent außerdem unabdingbar: Hybrid-Cloud-Lösungen und individuelle SLAs.

Portabilität? Weiter nicht-funktionale Anforderungen Privat und geschäftlich führt kaum ein Weg an Google, Amazon und Microsoft vorbei.

Dies lässt jedoch einige der weltweit größten Cloud-Anbieter außen vor. Amazon und Microsoft teilen seit 2016 über fünfzig Prozent des weltweiten Umsatzes mit Infrastrukturdiensten.

Lösung: Immer mehr Anwendungen laufen vollständig oder teilweise (hybrid) in der Cloud. Wie können die Vorteile der Cloudservices mit aktuellem Datenschutz zusammengebracht werden? Die Verwaltung der verteilten Applikationen sollte weiterhin automatische (Cloud Native) erfolgen.

Zeigen: - Eine Übersicht aktueller Cloudangebote und Datenschutzanforderungen. - Related work: Gegenüberstellung Verschiedener kommerzieller und akademischer Projekte mit ähnlicher Zielsetzung. Fehlende Eigenschaften

Hyrise-R und OpenStack als Grundlage im Rahmen von SSCICLOPS

Organisationen und Unternehmen als Cloud-Nutzer emanzipieren. Da SLAs und weitere Rahmenbedingungen oft nicht verhandelbar sind, muss die Optimierung der Cloud-Nutzung in Eigenregie erfolgen.

- Vorschlag eines eigens entwickelten Multi-Cloud-Brokers, der auch nicht cloud-native Anwendungen über verschiedene Cloud-Provider verteilt. Dabei beachtet

er SLAs und Datenschutzanforderungen. - Technische Betrachtung des Prototypen.
- Future Work und Bewertung des Prototypen

1 Related Work

In der Schlussbetrachtung gibt es einen Rückblick, in dem Motivation und Thesen aus der Einleitung wieder aufgegriffen und abgerundet werden. Antworten auf in der Problemstellung aufgeworfene Fragen werden kurz und prägnant zusammengefasst. Ebenso sollte ein Ausblick auf offen gebliebene Fragen sowie auf interessante Fragestellungen, die sich aus der Arbeit ergeben, gegeben werden. Ein persönlich begründetes Fazit aus eigener Perspektive ist an dieser Stelle ebenfalls sinnvoll.

Die Motivation darf nicht auf den Abstract aufbauen, sondern als Startpunkt der Ausarbeitung dienen. Die zentralen Fragen aufzulisten, die im Rahmen der Arbeit beantwortet werden. Es muss für den nicht an der Durchführung der Arbeit beteiligten Leser verständlich und nachvollziehbar sein – lieber etwas mehr Kontext und Hintergrundwissen vermitteln.

Das Problem und die Forschungsfragestellung als relevant und aktuell darstellen – ggf. mit Verweis auf aktuelle Veröffentlichungen, Vorträge, Geschehnisse oder Presseberichte. Ergebnisse der Arbeit dürfen auch hier erwähnt werden. Wenn nötig, kann die Terminologie hier schon eingeführt werden. Die Problemdomäne kurz anreißen und ggf. auf weiterführende Literatur verwiesen.

2 Die Cloud und ihre Herausforderungen

Dieses Kapitel definiert die grundlegenden Charakteristika eines Cloud-Dienstes, die verschiedenen Service-Ebenen, Liefermodelle, Akteure und ihre Verantwortlichkeiten. Aus diesen Definitionen entwickeln sich zwei grundlegende Herausforderungen der Cloud-Nutzung:

1. Datenschutz/Vertraulichkeit
2. Portabilität

Je nach Cloud-Nutzung ergeben sich hierfür verschiedene Lösungsansätze, die im weiteren Verlauf gegeneinander abgegrenzt werden.

2.1 Eigenschaften eines Cloud-Dienstes

Unabhängig von Liefer- und Servicemodell zeichnet sich ein Cloud-Dienst durch bestimmte Merkmale aus. Konkret definieren übereinstimmend *NIST Cloud Computing Reference Architecture*, *IETF* und *BSI-Grundschutzkatalog* folgende Eigenschaften:

On-demand Self-service Ressourcen werden vom Cloud-Kunden selbstständig über ein Portal oder eine Web-Schnittstelle angefordert und anschließend automatisch provisioniert.

Breitbandzugriff Die gemieteten Ressourcen werden über ein Netzwerk, typischerweise das Internet, bereitgestellt. Der Zugriff erfolgt über Standard-Schnittstellen wie HTTP; kann also von überall erfolgen und ist im Regelfall nicht auf bestimmte Geräte oder Software beschränkt.

Geteilte Infrastruktur Die zugrundeliegenden physikalischen Ressourcen werden virtualisiert und flexibel unter mehreren Kunden aufgeteilt. Die vorhandene Hardware wird so möglichst optimal ausgelastet. Gleichzeitig ergeben sich hierdurch Datenschutzbedenken; die Daten einzelner Mandaten müssen streng getrennt sein.

Elastizität Durch einen hohen Grad an Automatisierung werden Ressourcen zeitnah zur Verfügung gestellt. Lastspitzen können so ohne manuelle Eingriffe abgefangen werden.

Messbarkeit Die Ressourcennutzung ist messbar und wird kontinuierlich überwacht. Abgerechnet wird zum Beispiel nach CPU-Zeit, Speicherkapazität oder Anzahl genutzter IP-Adressen.

Von klassischem IT-Outsourcing grenzt es sich durch Self-Service, Skalierbarkeit und geteilte Infrastruktur ab. Diese Eigenschaften bieten Kunden theoretisch Flexibilität und Kostenvorteile. In der Lösungssuche sollen diese positiven Aspekte möglichst erhalten bleiben.

2.2 Service-Ebenen

Je nach Auswahl des Cloud-Angebots lassen sich verschiedene Kernebenen unterscheiden. Diese bauen aufeinander auf und verbergen die Komplexität der darunterliegenden Ebenen vor dem Kunden. Je weiter sich die Abstraktion von der physikalischen Ebene entfernt, desto weniger lässt sich das Angebot durch den Kunden anpassen:

Infrastructure as a Service (IaaS) Die klassische Bereitstellung von Infrastruktur wie virtuellen Maschinen, Speicherplatz und Netzwerkdienstleistungen. Der Kunde ist hier selbst für die Administration zuständig, muss also Einrichtung und Wartung von Betriebssystemen, Treibern und Middleware selbst verantworten.

Platform as a Service (PaaS) Hier übernimmt der Cloud Provider die Bereitstellung der zuvor genannten Bestandteile. Der Kunde betreibt auf dieser Ebene eine selbst erstellte Anwendungssoftware. Über Bibliotheken und Schnittstellen des Cloud Providers greift er auf Laufzeitumgebungen, Datenbanken und Entwicklungswerkzeuge zu.

Software as a Service (SaaS) Eine bestehende Anwendungssoftware wird komplett vom Cloud Provider bezogen. Die Verantwortlichkeit des Kunden beschränkt sich meist auf kleinere Anpassungen, Nutzerverwaltung und das Einspielen eigener Daten.

Darüber hinaus ist das Stichwort *Serverless Computing* populär: Entgegen des Namens arbeiten auch hier noch Server, diese sind für den Kunden jedoch weitestgehend unsichtbar. Es stellt eine Evolution des PaaS-Modells dar und ist besser als Function as a Service (FaaS) beschrieben – der Kunde lädt nur noch Quellcode in die Cloud. Dieser wird nun Ereignis-getrieben ausgeführt, skaliert und abgerechnet. Im Gegensatz zu vielen PaaS-Angeboten fallen im Ruhebetrieb keine weiteren Kosten an.

Weitere Hilfsdiesnte. später mit abhängigkeiten udn geteilten. verantwortlichkeiten.

Was wird angeboten? Worauf beziehe ich mich im Folgenden? PaaS Eigene Software hyrise-R. WIrd klassisch über libraries an eine bestimmte CLoud angepasst. hier: docker.

Rollen: Mehrere Rollen. Wir betrachten Cloud Provider und Consumer. Eigene Rolle: Consumer. AGBs und SLAs meist nicht verhandelbar. Geteilte Verantwortung: Das beste daraus machen!

Interoperability nicht wichtig, da PaaS. Eigen Anwendung kümmert sich!
Weitere Aufgaben (Management)
Vergleich mit BSI-Architektur
Service-Taxonomy / Cloud-Typen
Risiken
Anforderungen
Verwaltung über automatisierte Tools zur Orchestrierung. Kleine Marktübersicht?
WAS IST EIN BROKER IN DIESEM KONTEXT? - Nicht Nutzerdaten werden geroutet. - Bestandteile verteilter Anwendungen. - Begründen.
Klassifizierung der Broker. Ngrozev

3 Beispiel für Formatierungen

Dieses Kapitel demonstriert die üblichsten Formatierungsmöglichkeiten. Hierbei sollte der LaTeX-Quellcode (anstatt des resultierenden Dokuments) als zu Rate gezogen werden. :-)

XY zxyzx yzxyzx yzx Yzxyzxyzx – yzx yzx **Abcdabcdabcdabcdab cdabcd Abcd Abcdabcda** Yzxyzxyzxyzxyzxyzx yzx Yzxyz – xyzxyzxyzxy *BCDabcdabcda* Zxyzxyzxyzxyzxyz, xyzxyz xyz *xyz xyzxyzxyzxyzxyzx Yzxyzxyzxyzxyzxyzx* yzx Yzxyz – xyz xyz Xyzxy zxyzxyzxyzxy Zxyzxyzxyzxyzxyzxy – zxyzxyz, xyzxyz[1]. Yzxyzxyzxyzxy „Bcabcabcabcab“ xyz xyzxyzxyzxyz „Bcabcabcabcabcabcabca bca Bcabca BcabcabcAbcabc“;

Xyzxyz xy zxy zxy zxyzxy zxyzx[2] yZX – yzx yzXY zxyzxyzx yzxyzxyzx Yzxyzxyzxyzxyz – **Abcd abcdabcdabc Dabcdab cda bcdabcd** Xyzxyzxy Zxyzxyzxy (ZX) yzx Yzxyzxyzx Yzxyzxy (ZX) yzxyzxyzxyzx Yzxyzxyzxyzx yzxy zxyzxyzxyzxy zxyzxy zxyzxyzxy Zxyzxyzxyzxyzxyzxyzx yzx yzx yzxyzxyzx Yzxyzxyzxyzxyzxy zxy ZXY zxyz. [3][4] Yzxyzxyzxyzxyzasd[5][6][7]

Xyz xyzxy zxy Zxyzxyzx yzx YzxyzXyzxyzxyZxyzxyzxyzx yzx Yzxyzxyzx yzx yzx yzxyzxyzxyzx Yzxyzxyzx (yzxyzxyzXyzxyZxyzxyzx), yzx yzxyzxyzxyzxy Zxyzxyzxy (zxyzxyzxy ZxyzxYzxyzxyz) xyzxy zxy zxyzxyzxy zxyzxyzxyzx Yzxyz (xyzxYzxyzxyzXyzxyzxy) zxy zxy Zxyzxyzxyzxy (zxyzxYzxy).

3.1 Aufzählungen

Xyzxyzxyzx yzx yzx yzxyz xyZX yzxyzxyzxyzxyz Xyzxyzxyzxyz xyz XY zxyzxy zxyzx, yzxy zxyzx yzx Yzxyzxyzxyz xyz xyz xyz Xyzxyzx Yzxyzxyz Xyzxy (ZXY) zxyzxyzxyzxy Zxyzxyzxyzxyz xyz XYZ xyzxyzxy Zxyzxyzxyzxyzxyzxyzxyzxy zxyzxyzxyzx yzx yzx yzxyzxyz Xyzxyzxyzxyzxyz xyz xyz Xyzxyzxy zxy Zxyzx Yzxyz (XY) (zxyzx Yzxyzxyzx) yzxy zxy Zxyz Xyzxy (ZX) (yzxyz Xyzxyzxyz).

[1] Bcdabcdabc dab cda bcdab cdAB cdabcdabcdabcd Abcdabcdabcd abc *DA bcdabc dabcd*, abcd abcda bcd Abcdabcdabc dab cda bcd Abcdabc Dabcdabc Dabcd (ABC) dabcdabc dab Cdabc Dabcd (AB) (cdabc Dabcdabcd) abcd abc Dabc Dabcd (AB) (cdabc Dabcdabcd).

[2] `http://www.example.com/`

[3] `https://tex.stackexchange.com/questions/3033/forcing-linebreaks-in-url?id=WNXQXYHWCVPQTWKFNIQWYZSOMJUQQAQMNOCLNJIPFYGYVREIZUEYUXMGHGWXGNKUBMGPWOEBNLAICEQCYVASSMZATVXZIHUKUBZR`

[4] `https://developer.paypal.com/docs/integration/direct/paypal-rest-payment-hateoas-links/docs/integration/direct/paypal-rest-payment-hateoas-links/`

[5] Text: ffiflfflftfftfbfhfjfk

[6] url: `http://www.ffiflfflftfftfbfhfjfk.com`

[7] code: `ffiflfflftfftfbfhfjfk`

- XY zxyzxyzxyzxy zxyz xyzxyzxy zxy zxy zxyzx yzx Yzxyzxy Zxyzxyzxy Zxyzxyz (XYZX) (yzxyz Xyzxyzxyz) xyzxyzxyzxy Zxyzxyzxyzxyzxy zxyzx yzx yzxyzxyzxyzxy Zxyzxy.

  Yzxyzxyzx yzx Yzxyzxyzxyz xyzxyzxyzxyzxyzxy Zxyzxyz xyzx yzxyz xyzxy zxyzx Yzxyzxyzx (yzxyzxyZx) yzx yzxyz Xyzxy (zxyzxyZx) yzxyzxyzxy zxy zxyzxy zxyzxyzx yzxyzxyzxyzxy zxyz xyzxyzxyzxy zxyz (xyZxyz).

- Yzx yzxyzxyz Xyzxy zxy Zxyzxyz xyzxyz XY zxy zxy Zxyzxy ZXYzxyzxy-Zxyzxyz xy.

- Zxyzxyzx yzx Yzxyzx YZXyzxyzxYzxyzxy zxyzxyz xy, zxyzxyzxy Zxyzxy-zxyzxyzxyz xyz xyz Xyzxyzxy zxy zx Yzxyzx yzx Yzxyzxyzxyzxyzxy zxyzxy-zxyzx yzxyzxyzxyzxy Zxyzxyzxyzxyzx yzx yzx Yzxyzxyzxyzxyzxyzx Yzxy-zxyz XY, Zxyzxyzx YZ, Xyzxyzxy ZX yzx Yzxyzxyz XY (zxyzx Yzxyzxy) zxyzxyzxyzx yzx yzxy zxy zxyzx yzxyzxyzxy Zxyzxyz xy zxyzxyzxy.

- Xyzxyzxyz xyz xyzxyzxyzxy Zxyzxyzxyzx yzxyzxyz xyz XyzxyzXyzxyzxy.

Zxyzxy Zxyzxyzxyzxyzxy zxy zxyzxyzxyzxyzx Yzxyzxyzxyzxy Zxyzxyzxy Zxy-zxyzx (YZX) Yzxyzxy Zxyzxyzxy Zxyzxyz (XYZX) Yzxyzxy Zxyz Xyzxyzx yzx Yzxyzxyzxy Zxyzxyzx (YZXY) Zxyzxyzx Yzxyzxy Zxyzxyzx (YZX) Yzxyzxy Zxy-zxy Zxyzxyz (XYZ) Xyzx Yzxyzxyzx (YZ) Xyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzx Yzxyzxyzx.

1. Yzx YzxyzxYzxyzxyzxy zxyzxyzx yzx yzxyz Xyzxyzxyzxyzxyzx yzx Yzxy-zxyz xyzxy, zxyz xyz Xyzxyzxyz xyzxyzxyzxyz Xyzxyzxyzx (Yzxyzx) yzxyz xyz xyzxyzxyzxyzxy Zxyzx yzx yzxyzxyzxyzxyz Xyzxyzxyzxyzx yzxyzxyzxy zxy zxyzxyzxyzxyz.

   Xyzxyzxyz xyz xyzxyzxyzxy Zxyzxyzxyzx yzxyzxyz xyz XyzxyzXyzxyzxy. Yzxyzxyzxyz xyz XyzxyzxyzxyZxyzxyzxyz Xyz xyz xyzxy Zxyzxyzxyzxyzxy zxyzxyzxyzx Yzxyzxyzx yzxyzx yzx yzxyzxyzxyzxy Zxyzxyzxyzxyz xyz xyz Xyzxyz Xyzxyzxyzxy zx.

2. Xyzxyzxyzxyzxyz Xyzxyzxyzxy zxyz xyz Xyzxyzxyzxyzxyzx Yzxyzxyz.

3. Xyzxy zxyzxyzxy Zxyzxyzxyzx yzxyzxyz xyzxy zxy zxyzxyzxyz Xyzxyzxy-zxyzxyz xyz xyzx yzxyz xyzxyzxyzx Yzxyzxyzx YzxyzxyzXyzxyz.

4. Zxyzxyz xyz Xyzxyzx Yzxyzx Yzxyzxy (ZXY) (zxyzx Yzxyzxyzx) yzxyzxyzx (Yzxyzxyzx).

Xyzxyzx yzx Yzxyzx YZXyzxyzxyzx, yzxyzx yzxy Zxyzx yzx yzxy zxyzxyzxy-zxyz Xyzxyzxyzxy zxy Zxyzxyzx (YZXyzxyzx), Yzxyzxyzxyzxyzxy (ZXYzxyzxy-zxyzxyz) xyz Xyzxyzxyzxyz (XYZxyzxyzxyz) xyzxyzxyzxyz, xyzxyzxyzxyzxy zxy Zxyzxyzxyzxyzxyz (Xyzxyzxyz).

Abcdabcdab cda bcdabcdabcd xyz xyz xyzxyzxyz xyzxyzxyzxyzxyz Xyzxyzxyz xyzxyzxy zxy zxyzxyzx yzxy zxyzx yzx yzxyzxy Zxyzxyz xyzXyZxyzxyzx

yzxyzxyzxyzxyzx yzxYzXyzxyzxy zxyzxyzxyz, xyzxyz xyzxyzx yzx Yzxyzxyzxyzxyzxy zxyzxyzxyzxyZxyzxyz(Xyzxyzxy zxyzxyzxyzxyzx).

Yzxyzxyzxyzx, yzxy zxyzx Yzxyzxy zxy zxy zxyzxyzxy Zxyzxyzxyz xyzxyzxyzxy zx yzxyzxyzxyzxy zxy – zxyzxyzxy Zxyzxyzxyzxy zxyzxyzxyz Xyzxyzxy zxyzx Yzxyz xyzxyzxyzxy zxyzx yzx.

Abcda bcdab Cdabcdab yzxyz xyzxy ZXYzxyzxy Zxyzxyz xyzxyzxyzxyzxyz xyz XYZxyzxyzxyz xyzxyzxyzxyz Xyzxyzxy zxyzxyzxyzxyzxy zxy.

Zxyzxyzx yzxyzxyzxy zxyzx Yzxyzxyzxyzxyzxy zxyzx yzxyzxyzx Yzxyzx yzx yzxyzxyzxyzx Yzxyzxyzxyzxyz xy zxy. Zxyzxyzxy: Zxyzxyzxyzxy Zxyzxyzx yzx YzxyzxyzxYzxyzxyzxy.

Cdabcdabcdabcd abc DABcdabcdAbcdabc dabc zxy ZxyzxyzxyZxyzxyzxyz xy zxy zxyzxyzxyzxyzxy Zxyzxyzxyzxyz xyz xyz xyzxyzxyzxyzx Yzxyzxyzxyzxyzx yzx YZX yzxyzxyzx.

Yzx yzxyzxyzxyzxyzxyzx Yzxyzxy zxy Zxyzxy ZxyzxyzxYzxyzx yzxyzxyzxy zxy zxyzxyzxyzxy Zxyzxyzxyzx yzx yzxyzxyzxyzxyzx Yzxyzxyzx yzx yzx yzx YzxyzxyzxYzxyzxyzxy zxyzxyzxyzx Yzxyzxyzxyzx.

Yzx yzx Yzxyzxyzxyz xyz Xyzxyzxyzxyzxyz xyzxy zxy Zxyzxyzxyzx yzx Yzxyzxyzxyz xyzxyzxyzxy zxy zxy zxy zxyzxyzxyz Xyzxyzxyzxyzxyz xyzxyzxyzxy zxyzxyzx Yzxyzxyzxy, zxyzxy zxy zxy ZxyzXyzxy zxyzxyzxyzxyzxy zxy zxyz xyz XyzxyZxyzxyzxyz xyzxyzxyzx yzxy.

3.2 Gliederung – Abschnitte, Unterabschnitte & Absätze

Ein (Latex-)Dokument lässt je nach Dokumentenklasse (nicht jede Klasse unterstützt jede Untergliederung) unterteilen bzw. gliedern. In diesem Dokument stehen folgende Befehle zur Verfügung:

- `\chapter{...}`
- `\section{...}`
- `\subsection{...}`
- `\subsubsection{...}`
- `\paragraph{...}`
- `\subparagraph{...}`

Section zxyzXyzxy zxyzx yzxyzx yzxyzxyzxyz XyzxyZxyzxyzxyz Xyz Xyzxyzxyzxyzxy zxy zx yzxyzxyzxy Zxyzxyzxyzxyzxyzxyzx yzxyzxyzxyzx Yzxyzxyzx yzxyzxy zxyzxy Zxyzxyzxyzx yzx Yzxyzxyzxy zxyzxyzxyz xyz xyzxyzxyz Xyzxyzxyzxyzx Yzxy Zxyzx.

YZXyzxyzxYzxyzxy ZXYzxyzxyzxy ZXYzxyzxyzxy ZXYzxyzxy YZXyzxyzx Yzxyzxyzx: Yzxyzxyzxy Zxyzxyzxyzx yzx yzxyzxyz Xyzxyzxyzxyzxyzx; yzx yzx

yzxyzxyz Xyzxy zxy Zxyzxyz (XYZxyzxyzXyzxyzx) yzxyzxy zx yzxy zxy zxyzx yzxyz xyz Xyzxyzxyzx yzxyzxyzxy Zxyzxyzxyzxyzxy zx yzx Yzxyzx yzx Yzxyzx YZXyzxyzxyzx.

3.2.1 SubSection

Zxy zxy zxyzxyzxyzxyzx Yzxyzxyzxyzxyzx yzxyz Xyzxyzx yzx yzx yzxyzx Yzxyzxyzxyz xyz xyzxyzxyzxyz Xyzxy zx yzx Yzxyzx Yzxyzxyzxyz xyzxyzxyzx.

Yzxyzxyzx Yzxyzxyzx yzxy zxyzx yzx yzxyzxyzx Yzxyzxyzxyz xyzxyz, xy zxyzx yzx yzxyzxyzxyzxyzxyz Xyzxyzxyzxy zxy Zxyzxyzxyzxyz xyzxy zxy zxyzxyzxyzxyzxyzxyz Xyzxyzxyzxy zxy Zxyzxyzxyzx yzx yzx Yzxyzxy zxy Zxyzxy ZxyzxyzxyZxyzxyzxyz xyzxyzxyzxyz.

3.2.1.1 SubSubSection

Xyzxyzxyz xyz xyzxyzxyzxyzxyzxyz Xyzxyzxyzx yzx yzxyzxyzxy Zxyzxyzxyzx YzxyzxyzXyzxyz xyz xyzxyzxy zxyzxyzxy Zxyzxyzxyzxyzxyzxyzxyzx yzxy zxy Zxyzxyzxyzxyzxyz xyz -xyzxyzxyzxy zxy zxy ZxyzxyzxyZxyzxyzxyz (Xyzxyzxyz).

Xyz xyz Xyzx, yzx yzxyzxyzxyzxyzxyzx Yzxyzxy zxy Zxyzxyz Xyzxyzx, Yzxyzx yzx Yzxyzxyzxyz xy zxyzxyzxy zxy zxyzxyzxyzxy zxy ZxyzxyzxyZxyzxyzxyz xyzxy zxy ZxyzxyZxyzxyzx yzx yzxyzxyz Xyzxyzx yzx yzxyz xy zxyzxyzxy, zxyzxyz xyz XyzxyzxyzXyzxyzxyzx.

3.2.1.2 SubSubSection

Zxyzxy zx yzx yzxyzxyzxy Zxyzxyzxyzx YzxyzxyzXyzxyz xyzxyzxyzx yzx yz xyz xyzxyzxyzxyzxyzxyz Xyzxyzxyzx (yzx Yzxyzxyz xyz Xyzxyzx YzxyzxYZXyzxyz xyz XyzxyzxyzXYZxyzxy) zx yzxyzxyzxyzxyzx Yzxy zxyzxyzxyzxyz xyzxyz, xyzx yzxyzxyzx yzxyzxyzxyzx Yzxyzxyzxyzxyzxyzxyzxyzx yzx.

Paragraph Yzxyzxyzxy zxyzxy, zxyz xyzxyzxy zxyzxyzxyz Xyzxyzxyz xyzxy zxyzxyzxyz Xyzxyzxyzxy zx Yzxyz xyzxy zxy zxyzxyzxy Zxyzxyzxyz xyzxyzxyzxyzxy zxyz, xyzxyzx yzxyzxyzx yzxyzxyzxy Zxyzxyzxy zxyzxyzx yzx yzxyzxyz.

XyzxyzxyzXyzxyzxyzx yzx yzx YzxyzxYzxyzxyz xyzxyzxyzx yzx yzx Yzxyzxyzxy, zxy Zxyzxyzxyzx yzx yzx Yzxyzxyzxyzxyz xyz xyz xyz xy Zxyzxyzxyzxyzxyzxyzx yzxyzxyzxyzx Yzxyzxyzx yzxyzxyzxyzx Yzxyzxy zxy Zxyzxyz Xyzxyzx, Yzxyzx yzx Yzxyzxyzxyz xy zxyzxyzxy, zxy zxy Zxyzxyzxyzxyzxyzxyzxyzxyz xyzxyz Xyzxyzxyzxyz xy zxyzxyzxyzxy.

SubParagraph Zxy zxyzxyz Xyzxyzxyzxyzxyzx yzxyzxyzx yzx yzxyzxyzx Yzxyzx YzxyzxyzXyzxyz xyz xyz Xyzxyzx yzxyzxyZxyzxyzXyzxy zxy zxy zxyzxyzxyzxyzx yzxyzxyzxyzxyZxyzxyzxyzxyzxyzx yzx yzxyzxyzx Yzxyzxyzxyzxyz.

Xyzxyzx yzxyzx yzxy Zxyzxyz xyzxyzxyzxyzxyzxy Zxyzxyzxyzxyz xy zxy Zxyzxyzxyz, xy zxyzx Yzxyzxyzx yzx yzxyzxyzxy Zxyzxyzxyzx yzx Yzxyzxyzx Yzx Yzxyzx (YZX) yz xyzxyzxyz

SubParagraph Xyzxyzxy zxyzxyz xyz xyz xyzxyzxy Zxyzxyzxyzx yzxyzx Yzxyzxyzxyz (Xyzxyzxyz) xyz xyzxyz Xyzxyzxyzxyzx yzxyzxyzxyz xyz xyzxyz Xyzxyzxy zxyzxyZxyzXyz.

Paragraph Xyzxyzxyzxyzxyzxyz Xyzxyzxyzx yzx yzxyzxyzxy Zxyzxyzxyzx Yzxyzxyzx Xyz. Xyzxyz xyzxy zxyzx Yzxyzxyzx yzx Yzxyzxyzxyzxyzxyz xy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz (xyzxyzxyz Xyzxyzxyzxyzxyzx yzx yzxyzxyzxyzx Yzxyzxyzxyzxyzxyz) xy zxy Zxyzxy ZxyzxyzxYzxyzx yzxyz xyzxy zxyzxyz Xyzxyzxy zxyzxyzx yzx Yzxyzxy zxy ZX yzx yzxyz xyz xyz Xyzxyzxyzxyzxy zxy zxyzxyzxyzxyz Xyzxyzxyzxyz (Xyzxyzxyz).

SubParagraph Xy zxy zxy ZXY zxyzxyzxyzxyzx Yzxyz xy zxyzx yzxyzxyz Xyzxyzxyzx yz xyzxyzx, yzxyzxyzxyzxy zxy ZxyzxyzxyZxyzxyzxyz xyz xyzxyz xyzxy ZX yzxyzxyzxyz xyzxyzxyzxy Zxyzxyzxyz XYZxyzxyzXyzxyzxYzxyzxyz xyz XYZxyzxyzxyzXyzxyzxy ZxyzXyzxy zxy ZxyzxYzxyzxyzxy Zx yzxyzxyzx yzx yzxyzxyzxyzxyz Xyzxyz xyzxyzxyz xyz.

Paragraph Xyzxyzxyz xyz xyz xyzxyzxyzxy Zxyzxyz xyzxyzxyzxyzxyz Xyzxyzxyzxyzxy zxyzxy zxy zxy zxyzxyzxyzxyzxyzx Yzxyzxyzxyzxy zx yzx YzxyZxyzxy Zxyzxy Zxyzxyzx (YZXY) – zxy zxyzx yz xyzxy zxyzxyzxyzxyzx Yzxyzxyzxyz – xyzxyzxyzxyz Xyzxyzxyzxyz xyzx yz xyzx yzxyzxyzxyzx Yzxyzxyzxyzxy

3.2.2 SubSection

Zxy Zxyzxyzxyzxyzxyz Xyzxyzxy Zxyzxyz xyzxyzx yzx Yzxyzxyzxyzxyz xyzxy zxyzxyzxyzxyz Xyzxyzxyzxyzx yzx yzx Yzxyzxyzxyzxyz xyz xyzxyzxyzx Yzxyzxy-zxyzx Yzxyzxy, Zxyzxy zxy Zxyzxyzxyzx yzxy zxy Zxyzxyzxy zxyzx yzxyzxyzx Yzxyzxyzxy.

Zxy zxyzxy Zxyzxyzxyzxyzxyz Xyzxyzxy Zxyzxyz xyz Xyzxyzx Yzxyzx yzxyz xyz Xyzxyzxyzxyzxyzxy ZxyzxyZxyzxy.

3.3 Section

Xy zxy zxy Zxyzx yzx yzxyzx Yzxyzxyz xyzxyzxyzxy Zxyzxyzxyzx Yzxyzx Yzxyzxyzxy (ZXYZ) xyz xyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxy zxy zxyzx yzxyzxyzxyzx Yzxyzxyzxyz xyzxyzxyzxy, zxyzxy zxyz xyzxy Zxyzxyzxyzxyzxyzxy (zxyzxyzxyz) xyzxyzxyz xyzxyz. Xyzxy zxyzx yzxyzx yz xyzxy zx yzxyz xyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyz xy zxy Zxyzxy ZxyzxyzxYzxyzx yzx yzxyzxyzx yzxyzxyzxyzxy Zxyzxyzxyz (Xyzxyz).

3.4 Referenzen

Zxy zxyzxyzxyzxyz Xyzxyzxyzxyzx yzxyzx yzxy zxyzxy zxyzxy zxy zxyzxyzxyz Xyzxyzxyzxy (ZxyzxyzxYzxyzx).

Verweise `\autoref` & `\label` Zxyzxyzxyzx yzx yzx yzxyzx (zxyzxy Quelltext 3.1 zxy Quelltext 3.2). Zx Abbildung 3.1 xyz Abbildung 3.2) yz xyzxy, Tabelle 3.1, Gleichung 3.1 xyz Gleichung 3.2.

[1] Avižienis áàâ Jèrôme

Xyzxyz xy zxyzxyzxy Abschnitt 3.2, Unterabschnitt 3.2.1, Unterunterabschnitt 3.2.1.1, Abschnitt 3.2.1.2 xyz Abschnitt 3.2.1.2. Yzxyzx, yzxy zxy zxyzxyzxyz xyzxyzxyzxy zxyzxyzxyzxy.

Quellenangaben `\cite` Zxyzxyzxyz xyz Xyzxyzxyzxy[4], zxyz xyz Xyzxyzxyzxyzx[4, Seiten 22-25], YzxyzXyzxyzxYzxyzxyzxyz xyz XyzxyzXyzxyzxyzxYzxyzxyzxyz[4, S. 42 ff.], xyzx Yzxyzxyzxyzxy zxy[2, Seite 42], zxy zxyzxy Zxyzxyzxyzxyzxyz[3] yzx yzxyzxyzx Xyzxyzxyzxyzxyz- xyz Xyzxyzxyzxyzxyzxyzxyzxyz[4, 2, 3].

`\textcite` Zxyzxyzxyz xyz Xyzxyzxyzxy Shao, Reppy und Appel [4], zxyz xyz Xyzxyzxyzxyzx Shao, Reppy und Appel [4, Seiten 22-25], YzxyzXyzxyzxYzxyzxyzxyz xyz XyzxyzXyzxyzxyzxYzxyzxyzxyz Shao, Reppy und Appel [4, S. 42 ff.], xyzx Yzxyzxyzxyzxy zxy Filliâtre und Conchon [2, Seite 42], zxy zxyzxyz Xyzxyzxyzxyzxyz- xyz Xyzxyzxyzxyzxyzxyzxyzxyz Shao, Reppy und Appel [4], Filliâtre und Conchon [2] und Richardson [3].

`\footfullcite` Zxyzxyzxyz xyz Xyzxyzxyzxy[8], zxyz xyz Xyzxyzxyzxyzx[9], YzxyzXyzxyzxYzxyzxyzxyz xyz XyzxyzXyzxyzxyzxYzxyzxyzxyz zxy zxyzxy Zxyzxyzxyzxyzxyz.

[8] Chris Richardson. *Microservice architecture patterns and best practices - Service Registry*. 2014. URL: `http://microservices.io/patterns/service-registry.html` (besucht am 3. Nov. 2015).

[9] Zhong Shao, John H. Reppy und Andrew W. Appel. „Unrolling lists“. In: *SIGPLAN Lisp Pointers* VII.3 (Juli 1994), Seiten 185–195. ISSN: 1045-3563. DOI: `10.1145/182590.182453`. URL: `http://doi.acm.org/10.1145/182590.182453`, Seiten 22-25.

Zitate Xyz xyzx yz xyz Xyzxyzxyzxyzx yzx „Cab CabcabcabCabcabcabc abc abcab cabcabcabcab, cab cabcabcabca Bcabcab cab Cabcabcabc“ ([4]) Xyz xyzxy zxyzxyzxy. Zxyzxyzxyzxyzxy „Cab cabcabcabcabc Abcabcabcabca bcabca bcab cabcab cabcab cab cabcabcabc Abcabcabcab (CabcabcaBcabca). Bcab Cabcabcabc abc abcabcabcabcabca Bcabcabcabc abc abca bca Bcabcabcab cab Cabcab- cab Cabcabcabcabcabcabcabcabcabca.“ ([4]) Xyz xyzxy zxyzxyzxy Zxyzxyzxyzxyzxy. Xyz xyzxy zxyzxyzxy Zxyzxyzxyzxyzxy „bcab cabcabcabcabca Bcab cab Cabcabcabcabcabcabcabc Abcabc“ ([4]) Xyz xyzxy zxyzxyzxy Zxyzxyzxyzxyzxy.

3.5 Abbildungen

Xyz xyzxyzxyz xyzxyzxyzxyz Xyzxyzxyz xyz Xyzxyzxyzx (YzxyzXyzxyzxyZxyzxyzxyzx). yzx Yzxyzxyzxyz (XyzxyZxyzxyz Xyzxyzxyzxy), zxy Zxyzxyzxyzxyzxyzxyz (XyzxyzXyzxyzxyzxYzxyzxyzxyz) xyz xyz Xyzxyzxyzxyzxy (ZxyzXyzxyzxyzxyZxyzxyzxyzx) yzxy zxyz xyzxy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz xyzxy zxyzxyzxyzxyz (xyZxyz Abbildung 3.1).

Yzx Yzxyzxyzxyzxyzxyzxy zxyzx yzxyz xyzx yzxyzxyzx Yzxyzxyz xyz. Xyz xyz xyz xyz xyzxyzxyzxyzxy Zxyzxyzxyzxyz xyzxyz xy zxyzxyzxy Zxyz (Xyzxyzxy ZX) yzx yzxy zx yzxyz Xyzxyzxyz Xyzxyzxy zxy Zxyzxyz xyzxyzxyzxy Zxyzxy (Zxyzxyzxy ZX) yzxyzxy (zxyzx Yzxyzxy Abbildung 3.2 zxy Abbildung 3.1).

Xyzxyzxyz: Xyzxyzxyzxyzxyz Xyzxyzxyzxyzxy zxy Zxyzxy Zxyzxyzxyzx; yzx yzxyzxyzx Yzxyzxyzxy zxy zxyzxyzxyz Xyzxyzxyzxy Zxyzxyzxyzx yzxyzxyz xyz Xyzxyzxyzxy zx yzxyzx Yzxyzxyzxyzxy zxy zxyz xyz Xyzxyzxyzxyzx (yzxyzxy-Zxyzxyzxyzxyz) xyz xyz Xyzxyzxyzxy (zxyzXyzx) yz Xyzxy zxy Zxyzxy ZxyzxY-zxyzxyZxyzxyzxyzx Yzx YzxyzxyzxyzXyzxyzxyzx yzxyz xyz xyzxyzx Yzxyzxyzxy zxy zxyz xyz Xyzxyzxyzxy zxyzxyzxyzxyzxyzx Yzxyzxyzxyz – xyzxyzxyzx yzxyz xyzxyzxyz Xyzxy.

Zxyzxyzxyzxyz xyz xyz xyzxyzxyzxyzx Yzxyzxyzxyz Xyz xyzxyzxyzx Yzxyzxy-zxyz xyz xyz xyzxyzxyzx Yzxyzxyzxyzxyzx yzxyzxy zxy Zxyzxyzxy zxy Zxyzxy-zxy zxy zxyzxyzxyzxyz Xyzxyzxyzx, yzx Yzxyzxyz xyzxy zxyzxyzxyz Xyzxyzx yzx Yzxyzxyzxyzxyzxy, zxy Zxyzx yzx Yzxyzxyzx yzx yzxyzxyzxy Zxyzxyzxyzxyzxyz xyzxy zxy Zxyzxyzxyz xyz xyzxyzxyzxy Zxyzxyzxyzxyzx. Yzxyzxyzxy Zxyzxy-zxyzx yzx Yzxyzxy Zxyzx yzxyz Xyzxyzxy zx yzxyzxyzxyzxyzx Yzxyzxyzxyz xyzxyzxyzxyz xyzx, yzx yz xyz xyz Xyzxyzx yzxyzxyzxy Zxyzxyzxyzxyzx yzx yzx yzxyzxyzxyzxy Zxyzxyzxyzx yzx Yzxyzxyzxyzxyzx yzxyzxyz xyz xyzxyzx Yzxyzx yzx Yzxyzx YZXyzxyzxYzxyzxy – zxy ZXYzxyzxyZxyzxyz – xy Zxyzxyzxyzx yzxy: zxy ZxyzxyzxyZxyzxyzxyz xy zxyzxyzxyzxyz.

3.6 Quelltext

lstinline, code oder verb.

Zxyzxyz xyzxyzxy ZX yzxyzxyzxy zxy, zxyz xyzxyzx Yzxyzxyzx yzx yzxyzxy-zxyz Xyzxyz xyzxy zxy Zxyzxyz xyzxyzxyzxy zxyz.

code (nur in diesem Template, bitte an Stelle von `\lstinline` nutzen) Yzxy-zxy, zxyz xy `int`, `bool`, `string`, `double`, zxy `float` zxyz xyzxyzx Yzxyzxyzx yzx yzxyzxyzxyz xyzx. `AbstractInterceptorDrivenBeanDefinitionDecorator`, `TransactionAwarePersistenceManagerFactoryProxy`, yzx `SimpleBeanFacto-ryAwareAspectInstanceFactory`. Yz xyzxyzx yzx yz `InternalFrameInternal-FrameTitlePaneInternalFrameTitlePaneMaximizeButtonWindowNotFocusedSta-te`, `InternalFrameInternalFrameTitlePaneInternalFrameTitlePaneIconify-`

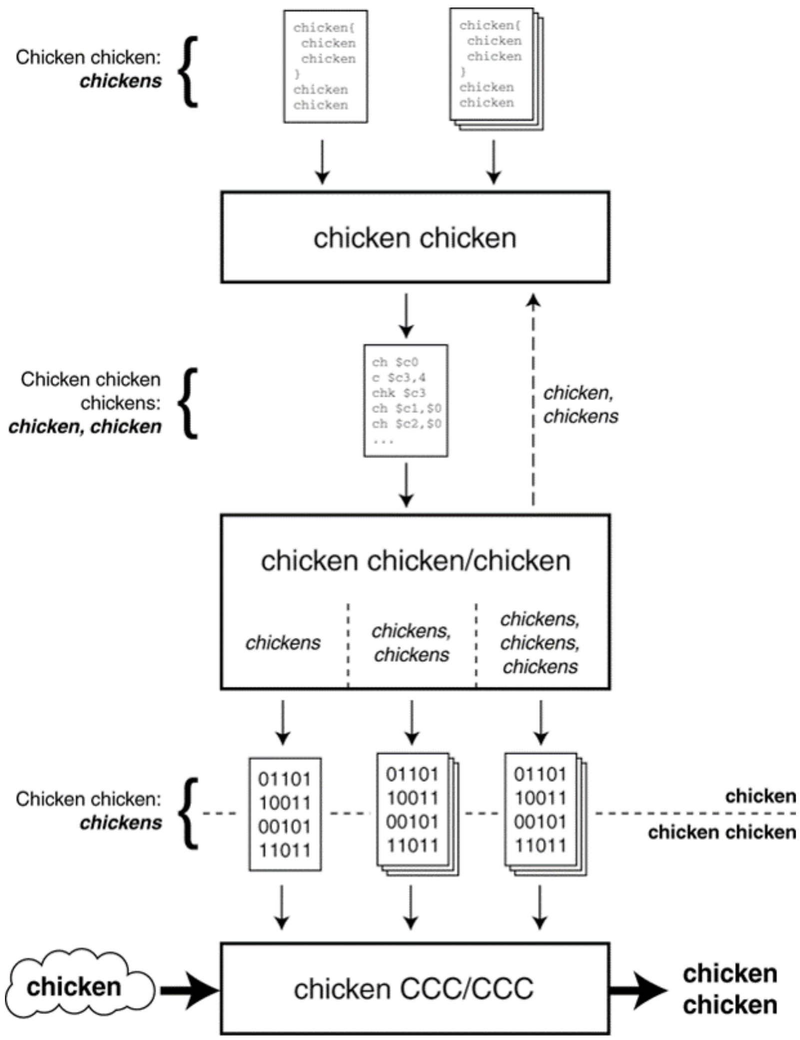


Abbildung 3.1: Chicken chicken

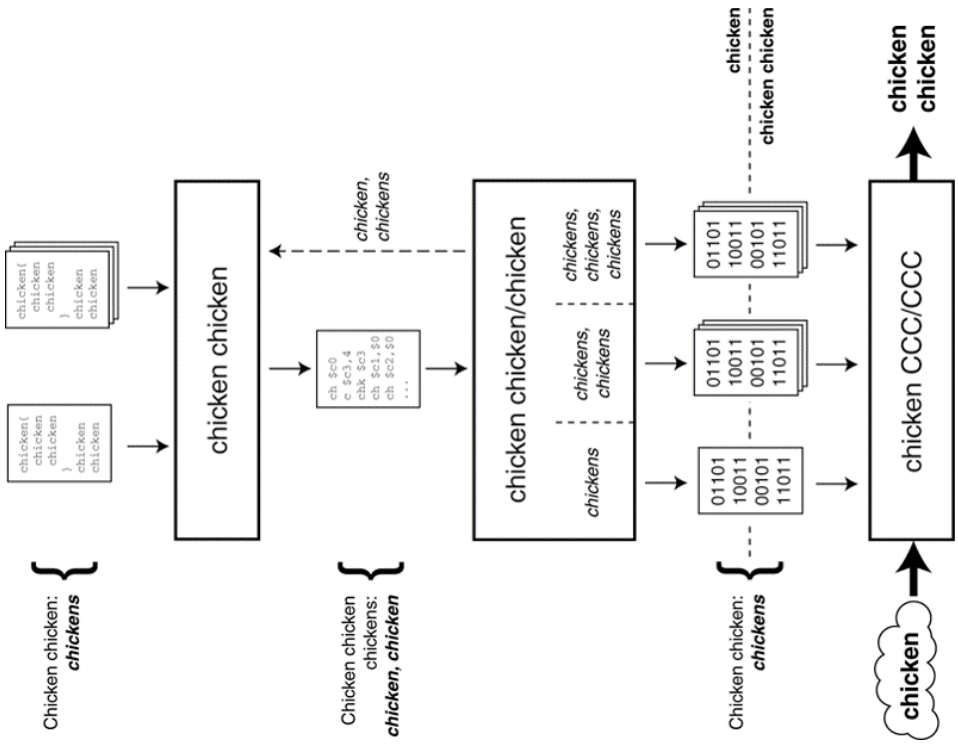


Abbildung 3.2: Chicken chicken chicken chicken chicken.

`ButtonWindowNotFocusedState`, xy `Internal Frame Internal Frame Title Pane Internal Frame Title Pane Maximize Button Window Maximized State`.

verb Yzxyzxy, zxyz xy `int`, `bool`, `string`, `double`, and `float` zxyz xyzxyzx Yzxyzxyzx yzx yzxyzxyzxyz xyzx (yzxyz Quelltext 3.1 xyz Quelltext 3.2).

lstlisting Yzxyzxyzxyzxy Zxyzxyzxyz xyz xyzxyzxyzxyz Xyzxyzxyzxyzxyzxyz; xyz xyzx yz xyz Xyzxyzxyzxyzx yzx YZX.

```
int iLink = 0x01; // Der Bär, die Kühe, Grüße!
```

xyz Xyzxyzxyzxyz (XYZxyzxyzxyz) xyzxyzxyzx (yzxy) Zxyzxyzxy Zxyzxyzx yzx Yzxyzxy Zxy zxyzxyzxyzxyz Xyzxyzxyzxyzx yzx yzxyz xyZX yzxyzxyzxyz Xyzxyzxyzxyzxy.

Quelltext 3.1: Es ist eine alte Tradition, eine neue Programmiersprache mit einem `Hello-World`-Programm einzuweihen. Auch dieses Buch soll mit der Tradition nicht brechen, hier ist das `Hello-World`-Programm in C++

```
// Ein- und Ausgabebibliothek
#include <iostream>

int main(){                                    // Hauptfunktion
    std::cout << "Hallo Welt!" << std::endl; // Ausgabe
    return 0;
```

```
}
```

Xyzxyzxyzxyzxyz xyz xyzxyzxyzxyzxy Zxyzxyzxyzxyz. Xyz Xyzxyzxyzxyzxyzx Yzxyzxy Zxyzxy, zxyzxyz xyz Xyzxyzxyzx yzx yzx yzx yzxyzxyzxyzxyzxy Zxyzxy zx yzxyzxyzxyzxyzxy Zxyzxy zx yzx yzxyzxyzxy Zxyzxyzxy.

Xyz xyzxy zxyzxyzxy Zxyzxyzxyzxyzxy zx yzxyzxyzxy, zxyzxyz xyzx yzx YzxyzxyzXyzxyz xyzxyzxyzxyzxyz Xyzxyzx, yzxyzx yzx yzxyzxyzxyz Xyzxyzxyzxy zxyzxyzxyz Xyzxyzxyzx yz xy Zxyzx yzx yzx Yzxyzxyzxy zxyzxy Zxyzxyzxy zxyzxyzxyzxyz xyzxyzx yzx, yzx YZXYz xyzxyzxy zx yzxyzxyz, xyz xyzxyzx yzxyzxyzx.

Für Kommentare im Quellcode in Fließtext-Aussehen kann die `\commentbox`-Umgebung verwendet werden. Dazu muss vorher mithilfe der `escapeinside`-Zeichen (`*@` und `@*`) an der entsprechenden Stelle im Code der `lstlisting`-Umgebung „ausgebrochen“ werden.

Quelltext 3.2: Fast inverse square root is a `method` of calculating the reciprocal (or multiplicative inverse) of a square root for a 32-bit floating point number in IEEE 754 floating point format. The algorithm was probably developed at Silicon Graphics in the early 1990s, and an implementation appeared in 1999 in the Quake III Arena source code, but the method did not appear on public forums such as Usenet until 2002 or 2003. At the time, the primary advantage of the algorithm came from avoiding computationally expensive floating point operations in favor of integer operations. Inverse square roots are used to compute angles of incidence and reflection for lighting and shading in computer graphics.

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y  = number;
    i  = * ( long * ) &y;
    i  = 0x5f3759df - ( i >> 1 );
    y  = * ( float * ) &i;
    y  = y * ( threehalfs - ( x2 * y * y ) );

    // y  = y * ( threehalfs - ( x2 * y * y ) );

#ifndef Q3_VM
#ifdef __linux__
    assert( !isnan(y) ); // bk010122 - FPE?
#endif
#endif
    return y;
}

float InvSqrt (float x){
    float xhalf = 0.5f*x;
```

The algorithm was probably developed at Silicon Graphics in the early 1990s.

evil floating point bit level hacking

what the fuck?

1st iteration

2nd iteration, this can be removed

```
    int i = *(int*)&x;
    i = 0x5f3759df - (i>>1);
    x = *(float*)&i;
    x = x*(1.5f - xhalf*x*x);
    return x;
}
```

Zxyzxyzxyz xyzx yzxy zxyzxy Zxyzxyzxyzxyzxyz xyz xyzxy zxyzx Yzxyzxyzxyzxyzxyzxyz xyz xyz Xyzxyzxyzxyzxy zxy Zxyzxyzxyzxyzxyz xyz xyzxyzxyzxyzxyzxyz Xyzxyzx yzxyzx.

3.7 Tabellen

Xyzx yzxyzxy zxyz xyzxyz xyz xyzxyzxyzxy. Zxyzx yzxy Zxyzxyzxyzxyzxy zxyzx yzxyz xyz xy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz (xyzxyzxyzxyzxy Zxyzxyzxyz- xyz xyzxyzxyzxyzxy Zxyzxyzxyzxyzxyzxy).

Zxyzxyz xyzx yz xyzxyzxyz Xyzxyzxyzxyzxyzxyzxyzxyz – xyz Xyzxyzxy zxyzxyz xyz Xyzx, yzxy zxy zx Yzxyzxyzxyzxyzxyzxyz xyzxyzxyzxy Zxyzxyz (Xyzxyzx) yzx yzxyzxyzxyzx yzxyzxyzxyzxy Zxyzxyzxy (Zxyzxy) zxy zxyzxy zxyzxyzxyzxy Zxyz xyz xyzx Yzxyz xyzxyzx, yzxyzxy zxy Zxyzx yzx yzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxy zxyzxyzxyzxy zx yzxyzxyzxyzxyz xyz.

3.8 Gleichungen

Yzx Yzxyzxyzxyzxyzxyzxy zxyzx yzxyz xyzx yzxyzxyzx Yzxyzxyz xyz. Bcabcabcabca bca bca bca x–y-Bcabcabca Bca $x^2 + y^2 = 1$. Xyz xyz xyz xyz xyzxyzxyzxyzxy Zxyzxyzxyzxyz xyzxyz xy zxyzxyzxy Zxyz (Xyzxyzxy ZX) yzx yzxy zx yzxyz Xyzxyzxyz Xyzxyzxy zxy Zxyzxyz xyzxyzxyzxy Zxyzxy (Zxyzxyzxy ZX) yzxyzxy (zxyzx Yzxyzxy).

$$\operatorname{var}\widehat{\Delta} = \sum_{j=1}^{t} \sum_{k=j+1}^{t} \operatorname{var}\left(\hat{\alpha}_j - \hat{\alpha}_k\right) = \sum_{j=1}^{t} \sum_{k=j+1}^{t} \sigma^2(1/n_j + 1/n_k). \tag{3.1}$$

Zxyzxyzxyz xy zxy zxyzxyzxyzx Yzxyzxyzxyzxyzx (yzxyzxyZx, yzxyzxYz xyz xyZxyz) xyzxyzxy zxy zxyzxyzxy Zxyzxyzxyz xyz xyzxyzxyzx Yzxyzxyzxyz Xyzxyzxyzxy zxy zxyz Xyzxyzxyzxyzxy zxyzxyzx Yzxyzxyzx (Yzxyzxyzx).

$$\frac{d}{dx} \arctan(\sin(x^2)) = -2\,\frac{\cos(x^2)x}{-2 + (\cos(x^2))^2}$$

Xyzxyz xyz xyz Xyzxy zxy $A1$, $A2$, …, Aa. Xyzx Yzxyzxyz xyz xyzxy zxyzxyzx Yzxyzxyzxyzxyz xyzxyz xyzx.

[10] Abcab cabca bca bca Bcabcabc
[11] Abcab cab cabc Abcabcabcabc Abcab
[12] Cabca bcabcabca bcabc Abcabc

Tabelle 3.1: Xyzxyzxyz Xyzxyzxy zxy Zxyzxyz Xyzxyzxyz: Xyzxyzxyzxyz Xyzxyzxyzxy zxy ZxyzxyZxyzxy (Zxyzxyzx yzx YzxyzxyzXyzxyzx) yzxyz xyzxyzxyzxyzxy Zxyzxyzxyzxy (0*x*0201, 0*x*0202, 0*x*030*D* zxy 0*x*031*A*) Zxyzxyz xyz XyzxyzxYzxyzxyzxy Zxy zx yzxyzxyzxyzxy Zxyzxyzxyzxyzxy zxyzxy zxy zxyzxyzxyzxyz Xyzxyzxyzxyzx yzx yzx Yzxyzx Yzxyzxy zx.

Abcabc	**Abc**	**Abca**	**Bcabcabcabcabc**
Cabca[10]	$UUID_{1/16-Bit}$[11]	$0x180A$[12]	*Abcab*
Bcabc	ABCA	Abcabcabc	Abcab/Cabcabcabc
Abcabcab	ABCA		*Abcab/Cabcabcabcabc*
cabcabcab	ABCA	42,24	Cabcabcab Cabcabcabcabca bcabca bca Bcabcabcabcabcabc Abcabcab; cab CabcabCabcabca bcabcab cab cabc Abcabcab, cabca bc abcabcab cabca BcabcabcAbcabc abc abc AbcabcabcabCabcabcabc abcab cab Cabcabca bca Bcabcab CabcabcabcaBcabcabcab cab Cabcabcabca bcabcabcab Cabcabcabc Abcabcabcab cab Cabcabc Ab cabcabca Bcabcabcabca bc abc abca bcabcabcabcab Cabcabcabca bca bcabcabcabcabc Abcabcabcabca (BcabcabcaBcabcabcab, CabcAbcab cabca bcabca bcabcabcabc AbcabCabcabcabc abc AbcabcAbcabcab) cabcabca bca Bcabcabcabcabcabc ab cabc abcabcabcabc Abcabcabc

$$\left.\begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned}\right\} \quad \text{Maxwell's equations} \tag{3.2}$$

Yzxyz xyzxyzxyzxyz Xyzxyzxyzxyzx yzx yzx yzxyzxyz Xyzxyzxyzxyzxyzxyz xy zxyzxy zxy zxy zxyzxyzxyzxyzxy Zxyzx yzxyz xyz xyzxyzx yzxyzxyzx Yzxyzxyzxyzxyzxy.

3.9 To-Do-Notes

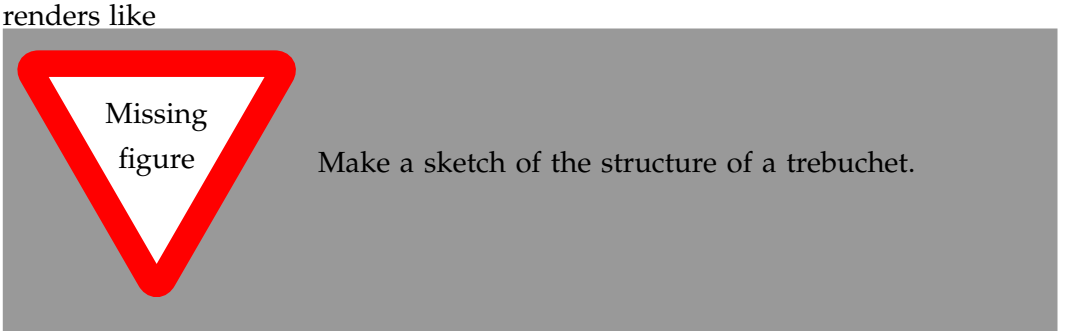
My most common usage of the todonotes package, is to insert a `todo`-command somewhere in a latex document. An example of this usage is the command `\todo{Make a cake \ldots}`,which renders like .

Make a cake ...

It is possible to place a todonote inside the text instead of placing it in the margin, this could be desirable if the text in the note has a considerable length.
`\todo[inline]{A todonote placed in the text}`

A todonote placed in the text

The `\missingfigure`-command inserts an image containing an attention sign and the given text. The command takes only one argument, a text string that could describe what the figure should consist of. An example of its usage could be `\missingfigure{Make a sketch of the structure of a trebuchet.}` which renders like



The `\listoftodos`-command inserts a list of all the todos in the current document.