JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Ausarbeitung

Dokumentbasierte Kommunikation / Datenbankreplikation

Systemtechnik
5BHITT 2015/16

SSteinkellner

Version 0.2
Begonnen April 2016
Beendet Juni 2016

Betreuer: M.Borko

Inhaltsverzeichnis

Dokumentbasierte Kommunikation

JSON

Allgemein

JSON ist die Abkürzung für JavaScript Object Notation und dient zur Definition von Objekten.[js1, js2]
In Javascript gibt es verschiedene Datentypen:

- String beinhaltet einen text und wird von Anführungszeichen begrenzt.
- Integer beinhaltet eine zahl und
- Boolean beinhaltet die Werte true oder false, wobei beide nicht begrenzt werden müssen.
- Ein Array wird von [begrenzt] und kann durch beistrich getrennt Variablen aller Datentypen enthalten.
- Ein Objekt wird begrenzt von { und }, und kann durch beistrich getrennte key-value-paare enthalten. Der Key kann ein String oder ein Integer sein, der Value kann jeder JSON-Datentyp sein.

Ein JSON-String kann ein Objekt oder ein Array enthalten, wobei weitere Daten verschachtelt enthalten sein können.

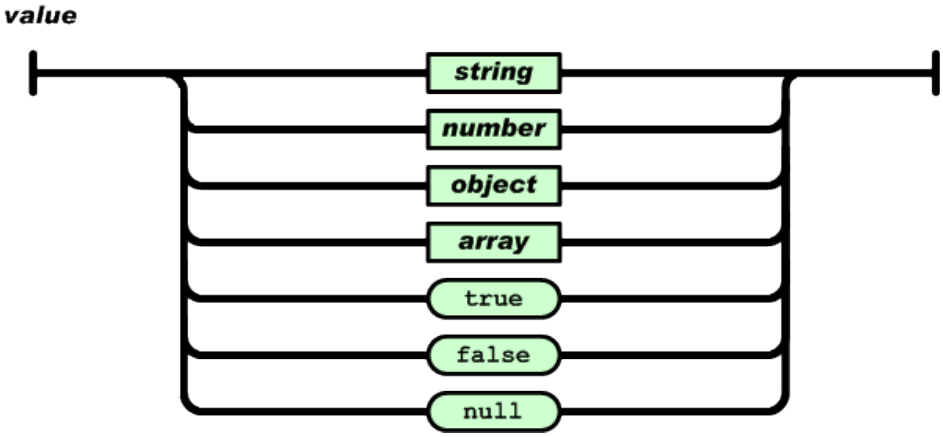

Abbildung 1: Datentypen in JSON [js1]

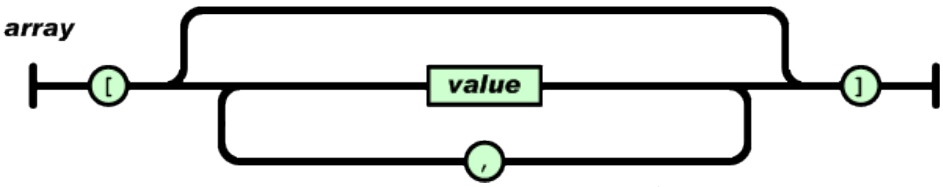

Abbildung 2: Arrays in JSON [js1]

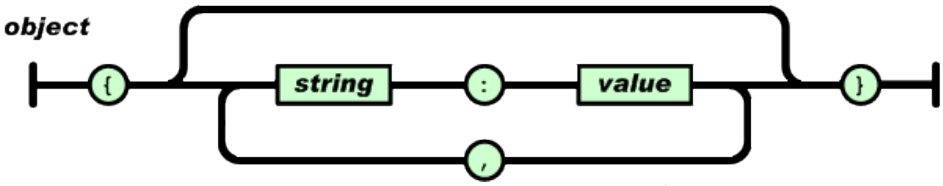

Abbildung 3: Objekte in JSON [js1]

Funktions/Ablaufdefinitionen

Mit der Funktion eval(text), bzw evalJSON(text) kann ein String angesehen werden, als wäre er direkt in Javascript programmiert worden. Dadurch kann code je nach Notwendigkeit erweitert werden, oder auch übers Netzwerk neue Funktionen gesendet, schnell eingebunden und sofort ausgeführt werden. [js3, js4, js5] Desweiteren können in dieser Form einzelne Funktionen eines Ablaufs definiert und in einer Liste gespeichert, bzw. als Liste gesendet werden. Somit kann von einem Gerät zum nächsten oder von einem Server direkt an ein Gerät ein bestimmter Ablauf gesendet und Schrittweise abgearbeitet werden.

Callbacks

In Javascript können Methoden wie Variablen behandelt werden, aber zusätzlich über Variablenname([Parameter]) aufgerufen werden. Dadurch entsteht die Möglichkeit, Funktionen in Listen zu speichern und nacheinander aufzurufen. Über eine Vorher definierte Funktion können Ergebnisse gespeichert, gesendet oder anders verarbeitet werden.
Desweiteren gibt es die Möglichkeit, in einem Objekt als Key gewisse Grenzwerte zu verwenden und als Value dann einen oder mehrere Parameter zu hinterlegen, die bei erreichen dieser Grenzwerte angepasst werden, beziehungsweise Funktionen die ausgeführt werden, wodurch eine einfache Regelung implementiert werden kann. [js6, js7, js8]

XML

Die "eXtensible Markup Language" ist eine Auszeichnungssprache für Daten, die sowohl von Menschen leicht gelesen als auch Maschinen leicht geparsed werden kann.[xml1]

Aufbau

Das Document Object Model besteht aus einem Header und mehreren Elementen. Der Header enthält die Information, dass es sich um ein XML-Dokument handelt, sowie die verwendete XML-Version. Die Elemente enthalten eine Typbezeichnung und können Attribute enthalten, in denen Eigenschaften gespeichert sind. Falls mehrere Daten gleicher Art gespeichert werden sollen, können diese als Kind-Elemente geschrieben werden. Unterhalb des Headers darf nur ein einziges Element in der äußersten Ebene existieren, das sogenannte Root-Element.[xml1, xml2]

Format

Das Root-Element kann Informationen über das Schema enthalten, nach dem die XML-Datei aufgebaut ist. Das Schema ist ebenfalls ein XML-Dokument, dass definiert, wie XML-Dokument die dieses Schema verwenden aufgebaut sein müssen.[xml3]

Verarbeitung

Zur Verarbeitung müssen XML-Datein zuerst in den Arbeitsspeicher geladen und dann von einem Parser ausgelesen werden. Für viele Programmiersprachen gibt es bereits Frameworks und funktionsfähige Parser zum herunterladen oder direkt integriert.

Der Inhalt der Dokumente kann mithilfe von XSLT (eXtensible Stylesheet Langugage Transformation) grafisch im Browser dargestellt werden. In der XSLT-Datei, die selbst auch wieder ein XML ist, wird das Aussehen der Darstellung definiert und mit IF-Anweisungen, Switches und Schleifen entsprechend der Daten eine Anzeige auf XML-Basis (zB.: XHTML) generiert. Der speicherort der XSLT-Datei kann im Root-Element angegeben werden.[xml1]

Grundeigenschaften

Zum Austausch von XML zwischen mehreren Systemen sind folgende Eigenschaften notwendig:

- ein Dokument ist w**ohlgeformt**, wenn es alle Regeln für XML-Datein erfüllt. In den Regeln ist festgelegt, dass nur ein einziges Root-Element vorhanden sein darf, Elemente sich nicht überlappen dürfen, wie Attributnamen und Inhalte aussehen dürfen, sowie die 3 möglichen Formen von Tags (öffnend <tag>, schließend </tag>, geschlossen <tag />).[xml1]
- ein Dokument ist g**ültig**, wenn zusätzlich zur Wohlgeformtheit auch die Definitionen des angegebenen Schemas erfüllt sind.[xml1]

Übertragen von Files

proprietär

Zur Übertragung von JSON-Strings können Sockets verwendet werden. Auf diese Art bekommen die beteiligten Programme sofort bei Vollendung der Übertragung die neuen Informationen und können darauf reagieren. Die Persistierung muss dabei selbst programmiert werden, kann aber auch die Festplatte entlasten, falls nicht der gesamte übertragene Inhalt sondern nur ein Teil davon gespeichert werden muss. Um auf die erhaltenen Daten reagieren zu können, müssen diese auf irgendeine Art in Objekte umgewandelt werden, außer sie bestehen nur aus einer einzigen Variable.

Distributed File System

Ein DFS hat den Vorteil, dass die Daten von einer, im OSI-Schichten-Modell weiter unten liegenden, Schicht bereits auf alle angebundenen Systeme verteilt wird und dadurch bei Ausfall eines Teils auf den anderen bereits vorhanden ist und nach einem Neustart aus dem "Schwarmbewusstsein" wiederhergestellt werden kann. Die Reaktion auf übertragung der Daten kann je nach verwendetem DFS durch Callbacks sofort erfolgen. Falls das eingesetzte DFS keine Callbackmöglichkeit besitzt, kann durch ständiges Abfragen des letzten Änderungszeitpunktes, je nach zeitlichem abstand der Überprüfungen relativ zeitnah, aber manchmal auch sehr verzögert auf eine Änderung reagiert werden. Open-source Systeme können angepasst werden und mit eigenen Callbackfunktionen ausgestattet werden, wodurch eine enge Kopplung zwischen DFS und der, die Befehle ausführenden, Software geschaffen werden kann.[dfs1, dfs2, dfs3]

Integration neuer Systeme

Translator/Content Enricher

Um die Daten auf mehreren verschiedenen Systemen konsistent zu halten, können gemeinsame Datenbanken verwendet werden. Wenn eine Änderung an den Systemen nicht möglich ist, müssen die Daten über Translatoren und Content Enricher ausgetauscht werden. Der Translator hat die Aufgabe, das Format übertragener Daten so anzupassen, dass der Empfänger den vom Sender gesendeten Inhalt interpretieren und damit arbeiten kann. Falls die gesendeten Daten nicht ausreichen, damit der Empfänger sie verwenden kann, so können über einen Content Enricher aus verschiedenen Datenquellen weitere relevante Daten gesammelt und gemeinsam mit den gesendeten Daten im vom Empfänger benötigten Format an ebendiesen Empfänger gesendet werden.

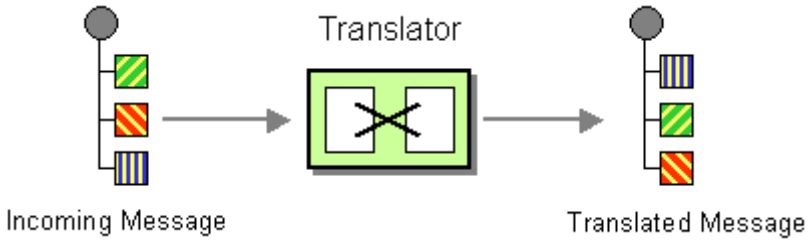

Abbildung 4: Translator [j3]

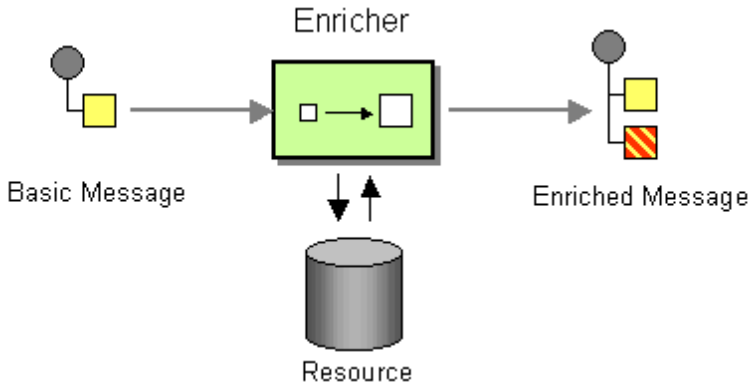

Abbildung 5: Content Enricher [j3]

Abgleich der Berechtigungen

Um Systembenutzer einfach verwalten zu können, sollten die Account- und Authentikationsinformationen Zentral gespeichert werden. Die Autorisationsinformationen sollten zwar abhängig von verwendeten Systemen, aber wenn möglich auch systemübergreifend nach gleichen Kriterien definiert werden. Bei der Integration müssen dann die neuen Berechtigungen an der zentralen Stelle hinterlegt werden. Die Speicherung der Berechtigungen und Accounts kann als ein File pro User mit Authorization- und Authentication-Informationsdaten umgesetzt werden, wodurch eine einfache Erweiterbarkeit erreicht werden kann.[db6]

Datenbankreplikation/Lastverteilung

Struktur/Datentypen

passende Datentypen für Ablaufdefinitionen

In einer Datenbank kann man Aublaufdefinitionen auf die folgenden 2 Arten speichern. Zum einen kann man Object-Methoden serialisiert als String in der Datenbank ablegen und sobald sie gebraucht werden, aus der Datenbank abrufen, deserialisieren und ausführen. Das Problem dabei ist, dass bei änderungen am Code die verschiedenen gespeicherten Methoden möglicherweise nicht mehr kompatibel sind. In dem Fall kann es zu Fehlern und dadurch unerwünschten bzw. unerwarteten Verhaltensmustern kommen.[j1]
Die Lösung dafür ist Gleichzeitig auch der zweite Ansatz. Die Methoden werden direkt im Programmcode ausprogrammiert, wobei relevante Variablen in der Datenbank abgelegt werden. Die Anpassbarkeit wird dadurch zwar eingeschränkt, aber auch die Möglichkeit, dass Fehler auftreten. Durch die Variablen kann über If-Anweisungen, Switches und Schleifen die Ausführung der Methode kontrolliert und gesteuert werden. Bei diesem Ansatz gibt es Ähnlichkeiten zu RMI, wo Funktionsname und Parameter für die Ausführung an ein anderes Gerät übertragen werden.

RMI

Remote Method Invocation ist ein Kommunikationsprotokoll für Aufrufe der Funktionen von Java-Objekten auf anderen Geräten. Die verwendbaren Funktionen werden in einem Interface definiert, auf der Aufrufer-Seite per TCP an die Server-Seite durchgeschleift, dort berechnet/abgearbeitet/etc. und das Ergebnis über den selben Kanal wieder zurück gesendet. Für den Aufrufer sieht es so aus, als wäre die Funktion lokal ausgeführt worden. Beim Aufruf können Parameter mitgegeben werden, die dann mit dem Request übers Netzwerk an den Server übertragen werden und die Ausfürhung beeinflussen können.[j2]
Um eine synchrone Durchführung eines Programmteiles auf mehreren Geräten zu erreichen, kann ein Befehl per RMI an alle betroffenen Geräte übertragen und ausgeführt werden.

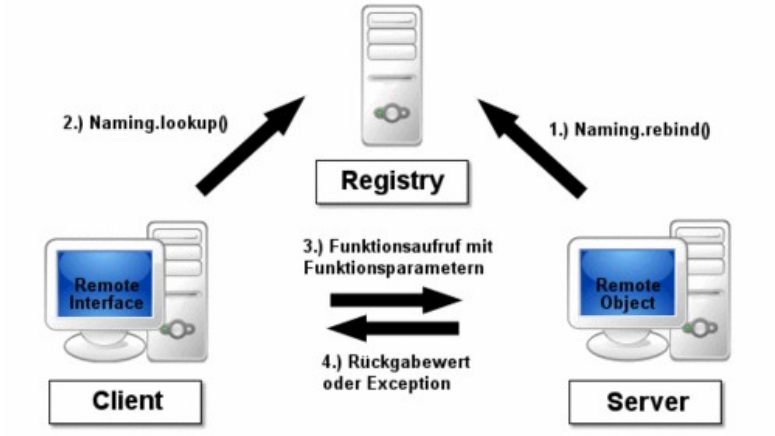

Abbildung 6: RMI Verbindungen [j2]

Replikation

Erreichbarkeit bei Ausfall einzelner Nodes

Weder der User noch extern angeschlossene Systeme bekommen vom Selbstmanagement und den ausgefallenen Nodes etwas mit, da das System nach außen hin wie ein einziges abgeschlossenes System aussieht. Sobald das System einen Ausfall bemerkt, wird es versuchen eine neue Instanz (falls vorhanden) zu starten, um die Schwachstelle zu überbrücken und einen Administrator vom Ausfall benachrichtigen, um bei Hardwaredefekten repariert oder ersetzt zu werden.[db7]

2-Phase-Commit

Das 2PC-Protokoll ist ein zuständig für die Gewährleistung der ACID-Eigenschaften einer verteilten Datenbank. Es ist in 2 Phasen unterteilt:

- Request-Phase [db2]
  Ein Server ("Koordinator") teilt allen Clients ("Teilnehmer") mit, welche Änderungen durchzuführen sind. Die Teilnehmer führen die Änderungen durch bis zu der Stelle, an der sie die Transaktion durch einen Commit abschließen oder durch einen Rollback rückgängig machen können. Dann, oder wenn während der Durchführung ein Fehler/Problem auftritt, melden sie sich beim Koordinator.
- Commit-Phase [db2]
  Sobald alle Clients alle Aktionen der Transaktion fertiggestellt haben, sendet er an alle Clients den Befehl, die Transaktion per Commit zu persistieren. Sollte ein Client ein Problem melden, wird an alle Clients der Befehl gesendet, die Änderungen der Transaktion per Rollback zurückzusetzen.

3-Phase-Commit

Das 3PC-Protokoll ähnelt dem 2PC, mit der Differenz, dass eine weitere Phase, die Precommit-Phase. Der Vorteil dabei ist, dass beim Ausfall des Koordinators ein neuer Koordinator ernannt werden kann, der den Commit abschließt oder zurücksetzt, da beim 2PC die Teilnehmer bis zur Bestätigung des Koordinators blockieren, was bei einem Ausfall zu endlosen Wartezeiten führen kann. [db3]

Verteilte Datenbanken

Fragmentierung

Bei der hoizontalen Fragmentierung werden die Daten einer Tabelle anhand gewisser Inhalte auf mehrere Tabellen geteilt, die alle die selben Spalten enthalten, aber unterschiedliche Daten. Die Daten können durch simples Aneinanderhängen der Zeilen der Fragmente auf den Gesamtstand zusammengebracht werden.[db9]

Diese Art der Fragmentierung ist sinnvoll, wenn auf mehreren geografisch optimal positionierten Servern alle Daten einer gewissen Gruppe von Benutzern, Produkten, etc. gebraucht wird.

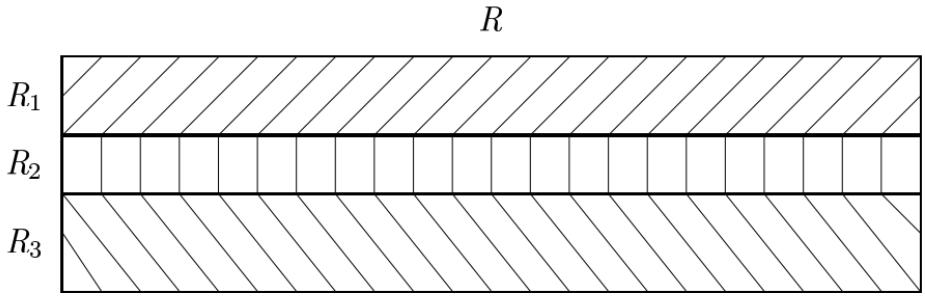

Abbildung 7: horizontale Fragmentierung [tgm4]

Bei der vertikalen Fragmentierung werden verschiedene Spalten einer Tabelle zusammensortiert und dann auf verschiedene Server abgelegt. Die Daten können per Primary Key, der in jedem Fragment enthalten sein muss, wieder zusammengefügt werden.[db9]

Ein Vorteil dieser Art der Fragmentierung ist, dass in einem Fragment der Tabelle mehrere Datensätze zu einem Datensatz in einer anderen Tabelle gespeichert werden kann. Es ermöglicht also (durch geringe Anpassungen der Primary Keys) 1-zu-N und M-zu-N Beziehungen.

Diese Art der Fragmentierung ist sinnvoll, wenn ein Teil der Daten in einer Tabelle sicherheitsrelevant ist (zB.: Passwörter, Kontodaten, Adressen) oder einfach manche Spalten öfter und andere seltener gebraucht werden.

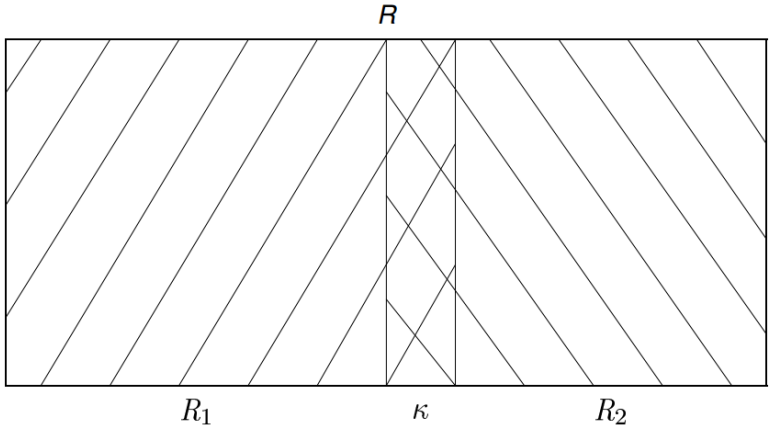

Abbildung 8: vertikale Fragmentierung [tgm4]

Schema

Im Zuge der Modellierung einer Datenbank werden aufgrund von verschiedenen Eigenschaften Relationen Festgelegt und in ein Relationenmodell übertragen. Im Relationenmodell wird jede Relation mit ihren Eigenschaften aufgelistet. Daraus kann das Datenbankschema erstellt werden, das die Tabellen, Spalten, Abhängigkeiten (in Form von Foreign Keys) und erlaubte Datentypen definiert.[db10]

Bei Verteilten Datenbanken gibt es mehrere voneinander abhängige Schema. Zur Fragmentierung gibt es das Fragmentierungsschema, dass die Aufteilung der Daten kennt. Darunter liegt das Zuordnungsschema, dass weiß, welche Daten auf welchem Server abgelegt sind. Auf den Einzelnen Servern gibt es lokale Schema, in denen jeweils der Teil des Fragmentierungsschemas abgebildet ist, der für den Server wichtig ist. Von oben gesehen, ist das Gesamtsystem Transparent, was bedeutet, dass der Client (solang er beim hauptserver, der das globale Schema besitzt, nachfragt), nicht merkt, dass es sich um eine verteilte Datenbank handelt.[tgm4]

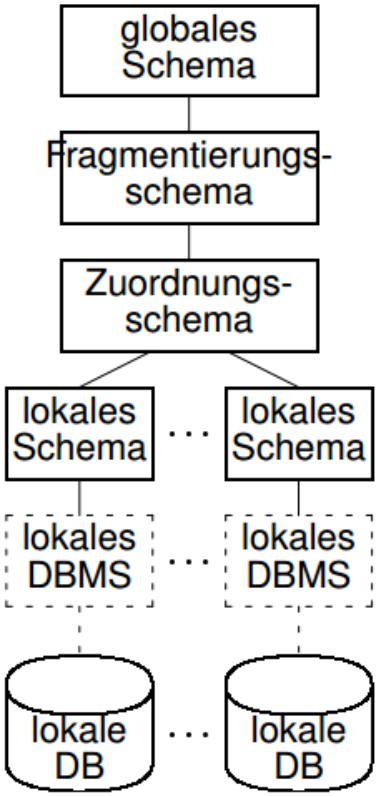

Abbildung 9: Fragmentierungsschema, Zuordnungsschema [tgm4]

Allokation

Die Zuordnung der einzelnen Fragmente zu Stationen, anhand von gewissen Faktoren, wird Allokation genannt. Daten können ohne Replikation, nur auf einer einzigen Maschine, oder mit Replikation, auf mehreren Maschinen, abgelegt werden.[tgm4]

Bei Verwendung der replizierten Allokation entsteht mehraufwand für das DBMS, da die Konsistenz über mehrere Systeme garantiert werden muss, dafür sind die Daten im Fall eines Ausfalls immer noch erreichbar. Bei Verwendung einer nicht replizierten Allokation sind die Daten nur auf einer Maschine vorhanden und bei einem Ausfall nicht mehr erreichbar, aber auch schwerer zu entwenden. Beim Speichern von vertraulichen Daten sollte die replikationslose Allokation verwendet werden, da die Angriffsmöglichkeiten proportional zur Anzahl der Server steht.

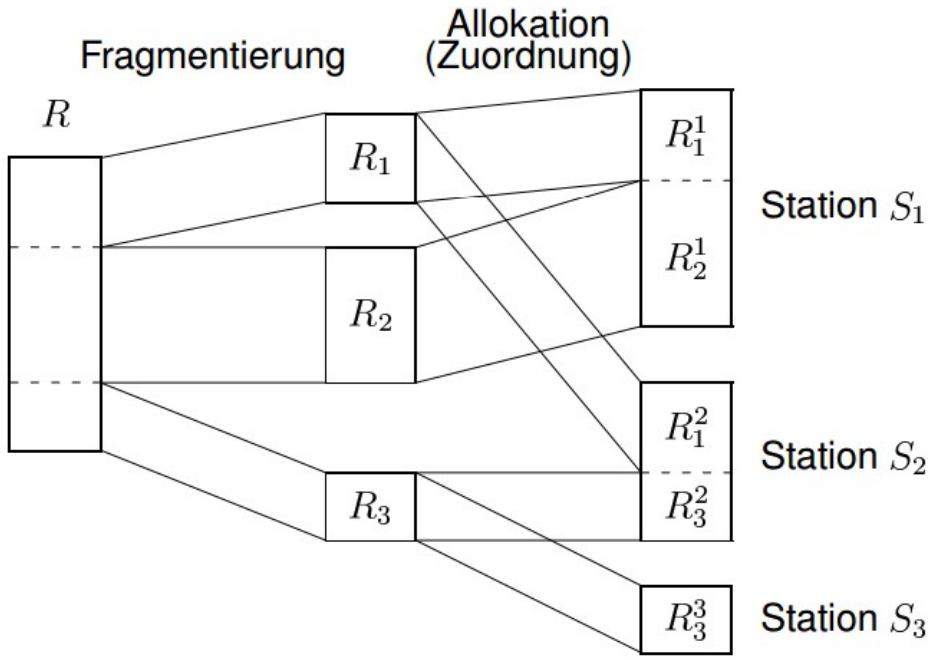

Abbildung 10: Fragmentierung & Allokation [tgm4]

Sicherheit

Schutz vor Hackerangriffen

Vor Hackerangriffen können Systeme auf verschiedenen Ebenen im OSI-Schichten-Modell geschützt werden. Die meisten Applikationen unterstützen zumindest eine Art der Authentication und Authorization. Desweiteren können durch minimierung der geöffneten Ports auf die für die Applikation notwendigen, sowie durch Verwendung von proprietären Protkollen, die Angriffsmöglichkeiten und damit auch das Risiko verringert werden.

Bot-Netze

Wenn Teile eines verteilten Netzwerks Stück für Stück übernommen wird, wird die Availability langsam beeinträchtigt, da die Geräte andere Aufgaben zugeteilt bekommen, als eingentlich vorgesehen. Meist werden Botnetze für Denial of Servie Attaken oder zum Knacken von Passwörtern durch Brute Forcing verwendet. Während die Denial of Service Attacke für die Betreiber unangenehm ist und Kosten verursachen bzw. Gewinnen verringern kann, ist ein Brute-Force-Attacke relativ gefährlich, da ein angegriffener Server nur schwer feststellen kann, dass es sich um eine Brute-Force-Attacke handelt, da jeder Versuch von einem anderen Rechner mit einer anderen Adresse stammt.[bot1, bot2]
Rechtlich gesehen sind Botnetze nur schwer zu bekämpfen, da die unterschiedlichen Rechtslagen in den verschiedenen Ländern meist nicht auf moderne Bedrohungen wie Botnetze ausgelegt sind. Daher werden Klagen durch Landesgrenzen meist komplett ausgebremst. Auf rechtlichem Weg können Botnetze also kaum bis garnicht behoben werden, wesshalb mittlerweile Cyber-Security-Firmen aber auch das FBI und Interpol versuchen Botnetze zu übernehmen, den Usern hinweise auf Reperatur zu geben und die Netzwerke dadurch zu entschärfen bzw. langsam abzubauen.[bot3]

Integration neuer Systeme

Konsistenz

ACID ist ein Konsistenzansatz, bei dem der Schwerpunkt auf Atomarität, Isolation und Dauerhaftigkeit liegen. Eine Transaktion wird dabei ganz oder garnicht ausgeführt und kann keine andere Transaktion beeinflussen. Das Ergebnis einer Transaktion ist immer Dauerhaft und sollte nicht zurückgesetzt werden, muss daher aber auch Konsistent sein.[db5]

BASE ist ein Konsistenzansatz, bei dem der Schwerpunkt auf Verfügbarkeit liegt. Der konsistente Zustand sollte schlussendlich irgendwann erreicht werden, muss aber nicht sofort erzwungen werden. Die Konsistenz kann dabei durch Versionsnummern erreicht werden, da jeder Zustand gültig ist und im Nachhinein auf jede Version zugegriffen werden kann. Diese Vorgehensweise füllt zwar die Datenbank, aber ermöglicht ein Zurücksetzen im Fall von korrupten Daten.[db5]

Wenn mehrere Systeme zusammengeführt werden, sollten sie den selben Ansatz verfolgen, da sonst möglicherweise 2 Datenbanken mit unterschiedlichen Datenständen Zustande kommen können und aufwendig verwaltet werden müssen.

DBMS-Wechsel

Anstatt neue Systeme zu integrieren, können bestehende Systeme auf ein gemeinsames System umgestellt werden. Dazu müssen die Daten der bestehenden Systeme so exportiert werden, dass sie im neuen System wieder importiert werden können. Dies kann über Translator geschehen, die die Daten auslesen und sofort wieder in das andere System eintragen, oder über weitverbreitete Datenformate wie XML. Im Fall eines Totalausfalls kann das System auf den Zustand der letzten exportierten Daten zurückgesetzt werden, falls diese auf einem externen Speichermedium persistiert wurden. Falls nur eine Instanz ausfällt, können beschädigte und fehlende Daten nach einem Neustart ebendieser Instanz von den anderen Instanzen erneut integriert werden, bzw. durch ein gemeinsames verteilte DBMS automatisch überspielt werden, sofern die Daten nicht nur auf dieser einen Instanz gespeichert waren.[db7]

Quellen

[js1] JSON Dokumentation/Übersicht
online: http://www.json.org/json-de.html
zuletzt aufgerufen: 27.05.2016

[js2] Selfhtml-Artikel zu JSON
online: https://wiki.selfhtml.org/wiki/JavaScript/JSON
zuletzt aufgerufen: 27.05.2016

[js3] Wikipedia-Artikel zu JSON
online: https://de.wikipedia.org/wiki/JavaScript_Object_Notation
zuletzt aufgerufen: 27.05.2016

[js4] Javascript Function eval
online: http://www.w3schools.com/json/json_eval.asp
zuletzt aufgerufen: 27.05.2016

[js5] API-Dokumentation
online: http://api.prototypejs.org/language/String/prototype/evalJSON/
zuletzt aufgerufen: 27.05.2016

[js6] Tutorial: "inkludieren einer Funktion in einen JSON-String"
online: http://solutoire.com/2008/06/12/sending-javascript-functions-over-json/
zuletzt aufgerufen: 27.05.2016

[js7] Wikipedia-Artikel zu Callbacks
online: https://de.wikipedia.org/wiki/Hook_(Informatik)
zuletzt aufgerufen: 28.05.2016

[js8] Wikipedia-Artikel zu Cross-Site-Callbacks in Javascript
online: https://en.wikipedia.org/wiki/JSONP
zuletzt aufgerufen: 28.05.2016

[dfs1] Wikipedia-Artikel zu Clustered File Systemen
online: https://en.wikipedia.org/wiki/Clustered_file_system#Distributed_file_systems
zuletzt aufgerufen: 28.05.2016

[dfs2] Wikipedia-Artikel zu Distributed File Systemen
online: https://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems
zuletzt aufgerufen: 28.05.2016

[dfs3] Wikipedia-Artikel zum Quantcast File System
online: https://en.wikipedia.org/wiki/Quantcast_File_System
zuletzt aufgerufen: 28.05.2016

[j1] Artikel zur Serialisierung
online: http://www.zdnet.de/39154667/wissenswertes-zur-serialisierung-von-java-objekten/
zuletzt aufgerufen: 28.05.2016

[j2] Wikipedia-Artikel zu RMI
online: https://de.wikipedia.org/wiki/Remote_Method_Invocation
zuletzt aufgerufen: 28.05.2016

[j3] Integration Patterns
online: http://www.enterpriseintegrationpatterns.com/patterns/messaging/index.html
zuletzt aufgerufen: 28.05.2016

[db1] Wikipedia-Artikel zu Replikation
online: https://de.wikipedia.org/wiki/Replikation_(Datenverarbeitung)
zuletzt aufgerufen: 29.05.2016

[db2] Wikipedia-Artikel zu Commit Protokollen
online: https://de.wikipedia.org/wiki/Commit-Protokoll
zuletzt aufgerufen: 29.05.2016

[db3] Wikipedia-Artikel zum Three-Phase Commit-Protokoll
online: https://en.wikipedia.org/wiki/Three-phase_commit_protocol
zuletzt aufgerufen: 29.05.2016

[db4] Digitalisiertes Buch Kapitel "Replizierte Datenbanken"
online: http://dbs.uni-leipzig.de/buecher/mrdbs/mrdbs-91.html
zuletzt aufgerufen: 29.05.2016

[db5] Wikipedia-Artikel zu Konsistenz
online: https://de.wikipedia.org/wiki/Konsistenz_(Datenspeicherung)
zuletzt aufgerufen: 29.05.2016

[db6] Blog-Eintrag über Konsistenz
online: https://martinzimmermann1979.wordpress.com/tag/konsistenz/
zuletzt aufgerufen: 29.05.2016

[db7] Blog-Eintrag über das CAP-Theroem
online: http://www.johndcook.com/blog/2009/07/06/brewer-cap-theorem-base/
zuletzt aufgerufen: 29.05.2016

[db8] Artikel zu Data Independence
online: http://www.tutorialspoint.com/dbms/dbms_data_independence.htm
zuletzt aufgerufen: 29.05.2016

[db9] Wikipedia-Artikel zu Fragmentierung
online: https://de.wikipedia.org/wiki/Denormalisierung#Fragmentierung
zuletzt aufgerufen: 03.06.2016

[db10] Wikipedia-Artikel zu Relationalen Datenbanken
online https://de.wikipedia.org/wiki/Relationale_Datenbank
zuletzt aufgerufen: 03.06.2016

[bot1] Wikipedia-Artikel zu DOS
online: https://de.wikipedia.org/wiki/Denial_of_Service
zuletzt aufgerufen: 30.05.2016

[bot2] Forschungsbericht zu Botnetzen
online: http://www.fkie.fraunhofer.de/de/forschungsbereiche/cyber-defense/projekt-botnetz-bekaempfung/fkie-studie-enisa-botnet.html
zuletzt aufgerufen: 30.05.2016

[bot3] Zeitungs-Artikel zu Botnetzen
online: http://derstandard.at/1224256061444/Botnets-Millionen-Zombie-PCs-machen-das-Web-unsicher
zuletzt aufgerufen: 30.05.2016

[xml1] Wikipedia-Artikel zu XML
online: https://de.wikipedia.org/wiki/Extensible_Markup_Language
zuletzt aufgerufen: 03.06.2016

[xml2] Wikipedia-Artikel zu DOM
online: https://de.wikipedia.org/wiki/Document_Object_Model
zuletzt aufgerufen: 03.06.2016

[xml3] Wikipedia-Artikel zu XSD
online: https://de.wikipedia.org/wiki/XML_Schema
zuletzt aufgerufen: 03.06.2016

[tgm4] TGM E-Learning SYT vierte Klasse

[tgm5] TGM E-Learning SYT fünfte Klasse