

# LOKALES WLAN BAISERTES MULTIPLAYER SPIELEFRAMEWORK FÜR ANDROID

## B A C H E L O R E A R B E I T

zur Erlangung des Grades eines Bachelore-Informatikers  
im Fachbereich Elektrotechnik/Informatik  
der Universität Kassel

eingereicht am

xx.xx.20xx

bei

Prof. Dr.-Ing. Albert Zündorf

Prof. Dr.-Ing. Lutz Wegner

Universität Kassel

von

Alexander Gerb

Liegnitzerstr. 6

34123 Kassel

# Zusammenfassung

Diese Arbeit umfasst die Implementierung eines Frameworks für Multiplayerspiele auf Basis des P2P-Frameworks Alljoyn und zwei weiteren Spielen, welche mithilfe des Frameworks erstellt worden sind und als praktische Beispiele dienen. Das Framework wird auf Basis des Android SDK realisiert und sollte eine Kommunikation zwischen mehreren Geräten soweit vereinfachen, dass man sich bei der Implementierung weiterer Spiele nur um die Spielmechanik Gedanken machen muss und die Lobby-Funktionalität, wie das Erstellen und das Verbinden zum Spiel vom Framework übernommen wird.

Als weiteres Kriterium gilt die Fähigkeit, Spiele im lokalen Wlan sichtbar zu machen und so eine Mehrspielerpartie zu ermöglichen ohne einen dedizierten externen Server, sowie einer Internetverbindung zu benötigen. Es sollte zum Spielen nur mindestens zwei Androidgeräte benötigen, die sich im selben Netzwerk befinden.

**Schlagwörter:** Multiplayer, Android, Lokal, Peer-to-Peer

# Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig angefertigt und mich fremder Hilfe nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichtem oder unveröffentlichtem Schrifttum entnommen sind, habe ich als solche kenntlich gemacht.

Kassel, xx.xx.20xx

---

Alexander Gerb

# Einleitung

Multiplayer Spiele auf mobilen Endgeräten gibt es unzählige. Jedoch sind die meisten dieser Spiele entweder Singleplayer oder sind Serverbasierte Multiplayer Spiele. Dies bedeutet, dass zum Spielen mit Anderen eine aktive Internetverbindung vorhanden sein muss. Trotz der mittlerweile gut ausgebauten Netzabdeckung und guter Internetgeschwindigkeit auf Smartphones, kann das Spielerlebniss durch den hohen Roundtrip getrübt werden. So sind z.B. Actionspiele meist nur über eine Hotspot Verbindung spielbar. Außerdem gibt es Situationen bei denen man keine oder nur eine beschränkte Internetverbindung hat, z.B. wenn man mit dem Zug unterwegs ist oder sich gerade in Ausland befindet und sich die Roaminggebühren sparen will. Gerade in solchen Situationen ist die Möglichkeit miteinander eine Mehrspielerpartie zu starten ohne eine Internetverbindung zu haben sehr willkommen. Um die Entwicklung solcher Spiele vorran zu treiben und es den Entwickler soweit wie möglich zu erleichtern, wollte ich ein Framework dafür zur Verfügung stellen. Ich wollte die ganzen Kommunikationsbedingen Herausforderungen mit dem Framework lösen, sodass die Entwickler sich nur um die Implementierung der Spiele selbst gedanken müssen. Die Entwickler hätten nur die Aufgabe sich um die Synchronisierung der spielbezogenen Daten zu kümmern und die Übertragen der Daten an die einzelnen Gerät würde sozusagen vom Framework übernommen. Damit die Entwicklung solcher Spiele für die Hobbyentwickler keine finanziellen Hürden stellt, habe ich mich für das Android SDK entschieden. Da das Android SDK kostenlos für jeden Entwickler zur Verfügung steht, kann jeder Entwickler, der über ein Android Gerät verfügt gleich mit der Entwicklung des Spieles loslegen. Natürlich lässt sich das Framework auch für praktische Anwendungen, wie z.B. Textmessenger, verwenden, jedoch werden ich im Rahmen dieser Ausarbeitung nur auf die Realisierung von Spielen eingehen.

# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>ii</b>
<b>Erklärung</b>	<b>iii</b>
<b>Einleitung</b>	<b>iv</b>
<b>Abbildungsverzeichnis</b>	<b>vi</b>
<b>Tabellenverzeichnis</b>	<b>vii</b>
<b>Quellcodeverzeichnis</b>	<b>viii</b>

<b>1 Einleitung</b>	<b>1</b>
1.1 Konzepte . . . . .	2
1.2 Ziel und Aufgabenstellung der Arbeit . . . . .	2
1.3 Lösungsweg der Aufgabenstellung . . . . .	3
1.4 Gliederung . . . . .	3
<b>2 Grundlagen</b>	<b>4</b>
2.1 Android SDK . . . . .	4
2.1.1 Activity . . . . .	5
2.1.2 Service . . . . .	5
2.1.3 Application . . . . .	5
2.1.4 Context . . . . .	5
2.2 AllJoyn . . . . .	6
<b>3 Versuchsdurchführung</b>	<b>7</b>
3.1 Theoretische Vorüberlegungen . . . . .	7
3.2 Versuchsaufbau . . . . .	7
3.3 Versuchsinstrumente . . . . .	8
3.4 Versuchsvorbereitung . . . . .	8

---

3.5	Praktische Versuchsdurchführung . . . . .	8
<b>4</b>	<b>Versuchsergebnisse</b>	<b>10</b>
<b>5</b>	<b>Diskussion der Messergebnisse</b>	<b>11</b>
<b>6</b>	<b>Weitere Problemstellungen / Ausblick</b>	<b>12</b>
<b>7</b>	<b>Fazit</b>	<b>13</b>
<b>A</b>	<b>Herleitung der Formel</b>	<b>I</b>
<b>B</b>	<b>Über das Problem X</b>	<b>II</b>
<b>C</b>	<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>D</b>	<b>Literaturverzeichnis</b>	<b>IV</b>

# Abbildungsverzeichnis

1.1	Von Text umflossenes Bild . . . . .	2
2.1	Von Text umflossenes Bild . . . . .	4

# Tabellenverzeichnis



# Quellcodeverzeichnis

3.1	Eine klassische <code>main.m</code> -Datei aus Objective-C . . . . .	7
-----	--	---

# 1 Einleitung

Spiele auf mobilen Endgeräten gibt es unzählige. Jedoch sind die meisten dieser Spiele entweder Singleplayer oder sind Serverbasierte Multiplayer Spiele. Dies bedeutet, dass zum Spielen mit Anderen eine aktive Internetverbindung vorhanden sein muss. Trotz der mittlerweile gut ausgebauten Netzaabdeckung und guter Internetgeschwindigkeit auf Smartphones, kann das Spielerlebniss durch den hohen Roundtrip getrübt werden. So sind z.B. Actionspiele meist nur über eine Hotspot Verbindung spielbar. Außerdem gibt es Situationen bei denen man keine oder nur eine beschränkte Internetverbindung hat, z.B. wenn man mit dem Zug unterwegs ist oder sich gerade in Ausland befindet und sich die Roaminggebühren sparen will. Gerade in solchen Situationen ist die Möglichkeit miteinander eine Mehrspielerpartie zu starten ohne eine Internetverbindung zu haben sehr willkommen. Man denke an die Situation man sitze im Zug mit paar Freunden und möchte eine Runde Karten spielen, aber man hat keinen Platz für die richtigen Karten. Um die Entwicklung solcher Spiele vorran zu treiben und es den Entwicklern soweit wie möglich zu erleichtern, wollte ich ein Framework dafür zur Verfügung stellen. Ich wollte die ganzen Kommunikationsbedingen Herausforderungen mit dem Framework lösen, sodass die Entwickler sich nur um die Implementierung der Spiele selbst Gedanken machen müssen. Die Entwickler hätten nur die Aufgabe sich um die Synchronisierung der spielbezogenen Daten zu kümmern und die Übertragen der Daten an die einzelnen Gerät würde sozusagen vom Framework übernommen. Damit die Entwicklung solcher Spiele für die Hobbyentwickler keine finanziellen Hürden stellt, habe ich mich für das Android SDK entschieden. Da das Android SDK kostenlos für jeden Entwickler zur Verfügung steht, kann jeder Entwickler, der über ein Android Gerät verfügt gleich mit der Entwicklung des Spieles loslegen. Natürlich lässt sich das Framework auch für praktische Anwendungen, wie z.B. Textmessenger, verwenden, jedoch werden ich im Rahmen dieser Ausarbeitung nur auf die Realisierung von Spielen eingehen.

## 1.1 Konzepte

Das Framework basiert auf dem Peer-to-Peer Prinzip und benötigt deswegen keinen weiteren Server. Weil durch einen Serverbasierten Ansatz der Zustand des Spiels nur auf dem Server organisiert wird ist es eine gewisse Herausforderung bei einem Peer-to-Peer Ansatz alle Geräte synchron zu halten. Der Entwickler hat nur Objekte, über die er die Kommunikation realisieren muss. Dabei handelt es sich zum einen um Objekte, die die eingehenden Signale behandeln und zum Anderen Objekte um Nachrichten an die anderen Spieler zu senden.

Jedes der Geräte hat zum einen einen Handler sowie einen Emitter. Der Emitter ist das Objekt welches die Signale an andere Geräte rausschickt und der Handler ist somit das Objekt, welches auf jedem Gerät die eingehende Nachricht behandelt. In den meisten Anwendungsfällen haben die Emitter und Handler auf allen Geräten die selbe Implementierung. Zusätzlich wird eine Session

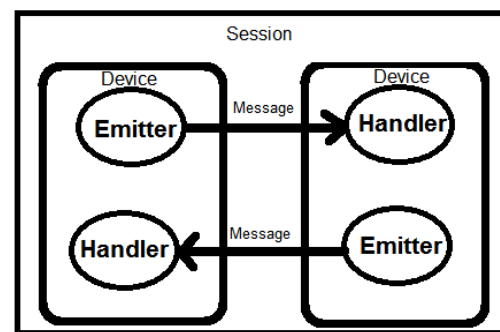


Abbildung 1.1: Kommunikations-Konzept

sitzungen oder Spielrunden in sich geschlossen zu handhaben. Somit erstellt ein Host eine Session zu der sich die die Klienten verbinden können um an der Spielrunde teilnehmen zu können. Dadurch wird außerdem die Möglichkeit gegeben mehrere Spiele simultan im selben Netzwerk zu spielen.

## 1.2 Ziel und Aufgabenstellung der Arbeit

Ziel dieser Arbeit ist es die Lokalisierung und Verbindung der Geräte vom Framework übernehmen zu lassen. Somit hat der Entwickler sich nur um die Implementierung der Schnittstellen-Objekte zu kümmern über die die Kommunikation abläuft. Das Framework soll auf dem Android SDK aufbauen, da es sich hierbei um eine kostenlos verfügbaren Entwicklungsframework handelt. Weiterhin sind zwei Spiele als praktische Beispiele mithilfe dieses Frameworks zu implementierung um zum einen die Vorgehensweise zu demonstrieren und zum Anderen die Realisierbarkeit zu testen.

### 1.3 Lösungsweg der Aufgabenstellung

Um sich die ganzen Herausforderung bei der Realisierung Endgerät-zu-Endgerät Kommunikation zu ersparen wurde ein weiteres Framework namens AllJoyn verwendet, welches die die Verbindung und die Kommunikation zum größten Teil schon übernimmt. Somit blieb zum Einen eine geschickte Integration von AllJoyn in das Framework, weiterhin mussten die vielen, gerade auf den ersten Blick komplizierten, Konfigurationen des AllJoyn Frameworks durch eine einfache Schnittstelle erweitert werden. Außerdem musste ein Multithreaded Handler-System verwendet werden um UI-Prozesse des Spiels nicht durch die Hintergrundprozesse der Gerätekommunikation zu behindern. Als praktische Anwendungsbeispiele wurde ein Echtzeitpuzzlespiel und ein Turnbased Kartenspiel namens MauMau implementiert.

### 1.4 Gliederung

Die Arbeit beinhaltet zu Anfang die Erklärung der verwendeten Frameworks AndroidSDK und AllJoyn. Dabei werden die Konzepte dieser Frameworks grob beschrieben. Als nächstes wird das Konzept und die Implementierung des Multiplayer Spieleframeworks, welches das Hauptaugenmerk dieser Arbeit ist, beschrieben. Daraufhin wird beschrieben wie die zwei Spielebeispiele realisiert wurden, sowie Herausforderung, die während der Implementierung entstanden. Zum Schluss gibt es noch eine theoretische Auseinandersetzung mit dem Prinzip der Wlan basierten P2P-Spiele.

## 2 Grundlagen

### 2.1 Android SDK

Android ist das Linux-basierte Betriebssystem für mobile Endgeräte, welches von Google 2011 offiziell zur Verfügung gestellt wurde. Android selbst gilt als sogenannte frei Software, welche bis auf den System-Kern unter der Apache-Lizenz steht. Diese Tatsache unter Anderen ermöglichte eine rasante Verbreitung dieses Betriebssystems auf vielen Geräten unterschiedlicher Hersteller.



Abbildung 2.1: Android

Somit waren im März 2013 etwa 750 Millionen Android End-Geräte aktiviert und man merkt schnell, dass die Popularität dieses Betriebssystems immer mehr zunimmt. Da ein Smartphone Betriebssystem auch von den angebotenen Apps lebt hat Google um die Entwicklung dieser eine Entwicklungswerkzeugsammlung zur Verfügung gestellt, welche die Entwicklung von Applikationen für Android möglichst einfach ermöglichen soll. Bei dieser Werkzeugsammlung handelt es sich um das Android SDK, welches auch als das Android Developer Tool kurz ADT verfügbar ist. ADT ist ein Plugin für das mittlerweile sehr weit verbreitete Entwicklungsumgebung Eclipse, welche die Entwicklung und die Übertragung der Applikation auf das Gerät mühelos ermöglicht. Weiterhin bringt das Android SDK einen Emulator mit sich, welches das Testen von Apps unter unterschiedlichen Konfigurationen ermöglicht ohne dass man ein Android-Gerät benötigt. Vorallem ermöglicht das Android SDK die Entwicklung der Apps in der Programmiersprache Java, welche sich immer höherer Beliebtheit erfreut und dank Eclipse zu einer höheren Produktivität beiträgt. Als nächstes gehe ich auf die einzelnen Grundlagen von Android SDK ein um die Funktionalität dieser zu beschreiben.

### 2.1.1 Activity

Eine Activity ist eine Klasse, welche einzelne UI-Fenster representiert. Somit bestehen die meisten Apps in Android aus mehreren Activities, welche mit einander verbunden sind. Ein Use-Case bei dem man mehrere Fenster hat, würde sozusagen mehrere Activities nacheinander aufrufen. Um eine Activity zu erstellen muss seine Klasse eine Unterklasse von Activity sein. Zusätzlich muss man die Methode ***onCreate()*** in seiner Klasse überschreiben. Diese Methode wird jedesmal aufgerufen wenn die Activity erstellt wird, also auf dem Bildschirm erscheinen soll. In diese Methode kommen meist Anpassungen an die UI hinein oder andere Operationen die beim Start notwendig sind.

### 2.1.2 Service

Ein Service ist eine Komponente, welche dazu gedacht ist Hintergrundprozesse zu übernehmen. Ein Service wird von z.B. einer Activity gestartet und läuft dabei im Hintergrund, selbst wenn die Activity nicht mehr existiert. Somit bietet sich ein Service gut an um z.B. die Netzwirkommunikation im Hintergrund zu behandeln ohne die Applikation selbst zu behindern. Weiterhin lässt sich ein Service auch an z.B. eine Applikation binden, sodass der Service auch beendet wird wenn die Applikation geschlossen wird.

### 2.1.3 Application

Application bietet zusätzlich zu den Activities die Möglichkeit während der ganzen Laufzeit der Applikation eine feste Instanz zu haben, welche den Zustand bestimmter Daten beinhaltet. So kann man es im Prinzip mit einem Singleton vergleichen der den Status der Applikation hält. Um an die Instanz zu kommen muss man aus dem Kontext heraus die Methode ***Context.getApplicationContext()*** aufrufen.

### 2.1.4 Context

Der Context beinhaltet Informationen über die Applikationsumgebung und lässt verschiedene Aktionen zu, wie z.B. das Aufrufen von weiteren Activities. Eine Activity ist eine Unterklasse vom Context und wird bei der Erstellung von z.B. einer View an diese übergeben.

## 2.2 AllJoyn

Duis[Lor09] sed lectus sem. Proin viverra venenatis tincidunt. Fusce eget turpis sit amet erat vestibulum pretium. In hac habitasse platea dictumst. Morbi eget massa et ante laoreet iaculis. Duis vitae nulla nulla. Suspendisse sit amet diam at enim accumsan consequat a eu sem. Proin venenatis ullamcorper gravida. Proin fermentum, metus vitae hendrerit bibendum, ligula dolor vestibulum eros, sed molestie lectus urna eget libero. Aenean ut sem nec metus tristique pretium.

## 3 Versuchsdurchführung

### 3.1 Theoretische Vorüberlegungen

Nulla fermentum velit quis elit volutpat auctor. Phasellus ante quam, mattis in bibendum vitae, elementum a lectus. Quisque id interdum arcu. Etiam eu nisl cursus nulla ornare placerat eu ac est. Pellentesque leo sapien, tempus ut ultrices quis, laoreet vel metus. Cras sed ultrices arcu. Aenean eu nisl eget mi interdum euismod eu vitae odio. Maecenas pharetra tempus interdum. Quisque at lacus ipsum, in vehicula leo. Curabitur commodo leo id mi posuere id interdum lectus volutpat. Aenean dapibus massa et mauris commodo molestie. Donec ac congue est. Mauris vel nisi odio.

```
1 #import <UIKit/UIKit.h>
2
3 int main(int argc, char *argv[]) {
4     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
5     int retVal = UIApplicationMain(argc, argv, nil, @"AppDelegate");
6     [pool release];
7     return retVal;
8 }
```

Listing 3.1: Eine klassische `main.m`-Datei aus Objective-C

### 3.2 Versuchsaufbau

In sed libero diam. Phasellus dignissim tincidunt tortor. Pellentesque sit amet mauris ac ipsum feugiat consequat. Curabitur fermentum libero sed lorem pulvinar molestie. Mauris sed est lacus. Aenean sed lacus sed nunc accumsan laoreet in nec tellus. Aenean sed consectetur lorem. Maecenas egestas magna id odio gravida vitae pretium erat dictum. Aliquam erat volutpat. Cras egestas tellus mattis mi tempus molestie.



Suspendisse quis libero orci, ac bibendum velit. Sed erat massa, sodales quis lacinia ac, faucibus et metus. Maecenas fermentum, massa ut volutpat laoreet, leo sem egestas massa, vitae tincidunt neque lacus ut ligula.

### 3.3 Versuchsinstrumente

Praesent erat sapien, convallis non aliquam sit amet, dignissim at lectus. Duis ac magna urna. Fusce vel risus non sem vestibulum gravida ut eget est. Vestibulum condimentum odio pharetra purus malesuada nec rhoncus erat ultrices. Mauris sagittis commodo aliquet. Pellentesque eget lorem sit amet enim aliquet dictum. Mauris luctus dictum dolor, at posuere mi condimentum ac. Mauris est enim, vehicula ut dapibus sed, tristique quis augue. Suspendisse laoreet dictum sapien, a convallis sapien porta quis. In commodo lacinia justo, vel bibendum felis pulvinar lobortis. Ut id tincidunt sapien. Donec pretium vulputate consequat. Donec ut bibendum metus. Maecenas at ipsum vel urna ultricies consectetur. Donec ligula tortor, convallis sed posuere vel, egestas nec sem. Maecenas sit amet orci mi. Sed sed arcu ut massa sagittis aliquam id non justo. Morbi pulvinar orci sed lorem eleifend aliquet. Donec rhoncus convallis turpis, in tristique mauris molestie sed.

### 3.4 Versuchsvorbereitung

Pellentesque ultricies, purus sed venenatis tincidunt, orci est pulvinar est, ut interdum nisl arcu a erat. Morbi ac risus id turpis malesuada eleifend lacinia non augue. Ut rutrum tristique consectetur. Vivamus dignissim varius lorem at hendrerit. Vivamus lacinia, libero et ultricies feugiat, erat enim luctus dui, vel ultricies dolor erat et diam. Praesent eget nisl nibh. Morbi commodo euismod dolor feugiat sodales. Morbi rutrum varius sapien nec pulvinar. Nullam et dolor arcu. Nulla facilisi. Etiam consequat elementum mauris, a viverra urna semper non. Nulla ornare ante eu turpis lobortis placerat. Donec vestibulum dolor at libero porta tristique.

### 3.5 Praktische Versuchsdurchführung

Suspendisse lacus lacus, volutpat vel luctus et, facilisis ac est. Ut tempor arcu vel urna rutrum rhoncus. Nam sodales dictum felis, id rhoncus nulla pharetra sit amet.

Nulla facilisi. Cras ultrices odio non mauris pretium condimentum. Curabitur vel leo sem, nec posuere erat. Pellentesque consequat tempor justo, id scelerisque erat congue a. Integer vel metus vel nulla tempor luctus sit amet at tortor. Curabitur imperdiet molestie elit. Curabitur hendrerit porttitor lorem posuere hendrerit. Nulla facilisi. Cras sodales metus ac augue facilisis ultricies. Vestibulum facilisis vehicula ultricies. Sed lobortis sodales odio, sit amet ullamcorper purus adipiscing sed. Donec molestie dictum turpis eget placerat.

## 4 Versuchsergebnisse

Vivamus ac hendrerit nisl. Morbi viverra sagittis urna, ac gravida diam posuere in. Nullam id enim nunc. Cras commodo eros ac mi iaculis ac aliquet eros molestie. Vestibulum semper, mauris a dapibus commodo, magna erat imperdiet ante, ac varius dolor ipsum et augue. Etiam non massa purus. Cras faucibus risus nec diam consectetur vestibulum. Donec luctus nunc eget diam commodo quis tempus augue mattis. Proin mattis mauris vehicula leo imperdiet tristique. Nullam a metus quam. Vivamus at turpis lorem, vel dapibus nisi.

## 5 Diskussion der Messergebnisse

Nulla ultrices accumsan turpis, at ultrices libero laoreet vitae. Nam ac ante in orci lobortis rutrum. Quisque metus diam, malesuada ac facilisis id, egestas eget dui. Donec id ante et ligula tincidunt congue. Ut eget neque eu sem elementum imperdiet. In euismod est id massa tristique eget vestibulum nisl lacinia. Nullam ullamcorper odio ut sem porta vulputate. Curabitur turpis turpis, tincidunt fermentum suscipit vel, vehicula quis nulla. Duis rhoncus sagittis condimentum. Ut molestie adipiscing mauris, vel imperdiet dui dignissim vel. Integer ac purus ante. Aenean euismod vulputate metus. Cras sit amet imperdiet ligula. Donec at diam diam. Donec posuere libero vel risus iaculis ullamcorper. Proin interdum pretium ante, eget rutrum felis elementum ut. Nam nisl nibh, sagittis in commodo vitae, venenatis non eros. Aenean pharetra vestibulum erat a tempor. Proin quam lacus, molestie a interdum id, consequat ac risus.

## 6 Weitere Problemstellungen / Ausblick

Praesent convallis ligula a mauris lacinia egestas. Praesent egestas, nunc in varius imperdiet, risus ante eleifend nulla, in feugiat ipsum lacus non metus. Cras lobortis metus ut leo consequat ut vulputate ligula ullamcorper. Nullam id arcu urna. Etiam non enim leo, a dictum augue. Sed quis tellus at lectus euismod sagittis. Proin a lectus quis nulla ultricies faucibus. Donec et odio id libero vulputate interdum at consequat diam. Praesent dictum pharetra lorem, eu fringilla erat porttitor sed. Etiam malesuada ipsum ac odio facilisis a viverra risus pharetra. Duis quis sem at sem luctus gravida. In hac habitasse platea dictumst. Donec eu metus et sapien molestie hendrerit in eget dolor.

## 7 Fazit

Nam accumsan leo ut magna adipiscing sollicitudin. Etiam sodales adipiscing molestie. Nulla auctor mattis dapibus. Aliquam molestie turpis pulvinar tortor porttitor at sagittis turpis faucibus. Sed ligula ipsum, placerat ut feugiat eu, condimentum eget mi. Aenean id egestas libero. Donec neque nisl, lacinia vitae condimentum quis, bibendum sit amet diam. Morbi in metus vitae lacus ullamcorper aliquam et eu risus. Nullam eros lorem, consectetur a laoreet molestie, volutpat ac dolor. Praesent ornare, mi vitae rutrum sollicitudin, turpis mi euismod nisi, sollicitudin viverra sem erat a mauris. Nam eget velit dolor. Vestibulum neque nibh, aliquet ac dictum eget, rhoncus et urna. Suspendisse ultricies hendrerit dignissim. Nullam hendrerit, sem interdum feugiat consequat, sem justo pretium lorem, faucibus iaculis dui diam commodo augue. Phasellus ut mauris leo. Cras quis leo nibh, eu convallis urna. In tristique condimentum laoreet. Nam ac nunc id ipsum elementum commodo. Morbi augue nulla, ultrices et venenatis sed, fringilla et risus.

## A Herleitung der Formel

Lorem ipsum (LI) dolor sit amet, consectetur adipiscing elit. Etiam quam sapien, mattis non varius eu, rutrum eget nisl. Morbi venenatis molestie ante, sed aliquet lectus aliquet id. Pellentesque consectetur nisl a massa ornare congue. Curabitur pellentesque hendrerit dolor eget faucibus. Etiam non risus arcu, id fermentum elit. Quisque suscipit posuere semper. Vestibulum sit amet dolor nec risus malesuada interdum aliquam in turpis. Maecenas mollis, magna at porttitor fringilla, risus libero commodo justo, non tempus nibh massa lacinia sapien. Aenean sodales ullamcorper massa, eu ullamcorper ipsum tempus sed. In adipiscing congue scelerisque. Pellentesque molestie, quam vel dictum iaculis, metus nunc mollis mi, nec venenatis tellus turpis eu arcu. Praesent at ultricies nibh. Proin neque libero, tincidunt dignissim ornare in, sagittis in ligula. Nunc sagittis sodales massa, a tempus felis vehicula id. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla adipiscing vestibulum eros, ut imperdiet augue scelerisque id. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Suspendisse aliquam pulvinar lectus id dictum. Etiam dictum sollicitudin elit sed scelerisque. Nullam sodales semper interdum.

## B Über das Problem X

Sed turpis erat, tincidunt eu sollicitudin eu, tempus quis lectus. Nullam orci leo, tempus vitae dictum eu, bibendum at ante. In placerat, mi eu consequat suscipit, turpis arcu dictum tellus, nec scelerisque turpis eros a enim. Mauris quis leo lacus. Vestibulum condimentum porttitor malesuada. Pellentesque nec dictum nisl. Donec eleifend libero sit amet urna dignissim in faucibus lorem ultricies. Vestibulum interdum egestas metus vel porta. In vehicula leo at nibh dignissim malesuada. Etiam non nunc ligula. Vivamus vitae nibh dolor, eget faucibus tortor. Suspendisse ac quam enim.



## C Abkürzungsverzeichnis

LI Lorem ipsum

## D Literaturverzeichnis

- [Mus09] Mustermann, Max: *Titel. Untertitel*. Auflage. Verlagsort: Verlag, Jahreszahl (= Reihe).
- [Lor09] Ebers, Robin (2009): „Lorem Ipsum“. URL: <http://de.lipsum.com>  
[Stand: 11.10.2009]