

Dieser Vortrag ist unter der Lizenz der Creative Commons Namensnennung-Nicht-kommerziell-Weitergabe unter gleichen Bedingungen 3.0, Deutschland veröffentlicht

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

# Die Android-Plattform

Java für unterwegs?

# Moritz Haarmann

Medieninformatik, HdM Stuttgart



Nolte&Lauth GmbH



@derwildemomo

# Benjamin Reimold

Angewandte Informatik, DH BW



aformatik



@elektrojunge

**Hallo Android!**

# Wer seid Ihr?

Hände hoch!

# Agenda

Was ist  
Android

Apps für  
Endkunden

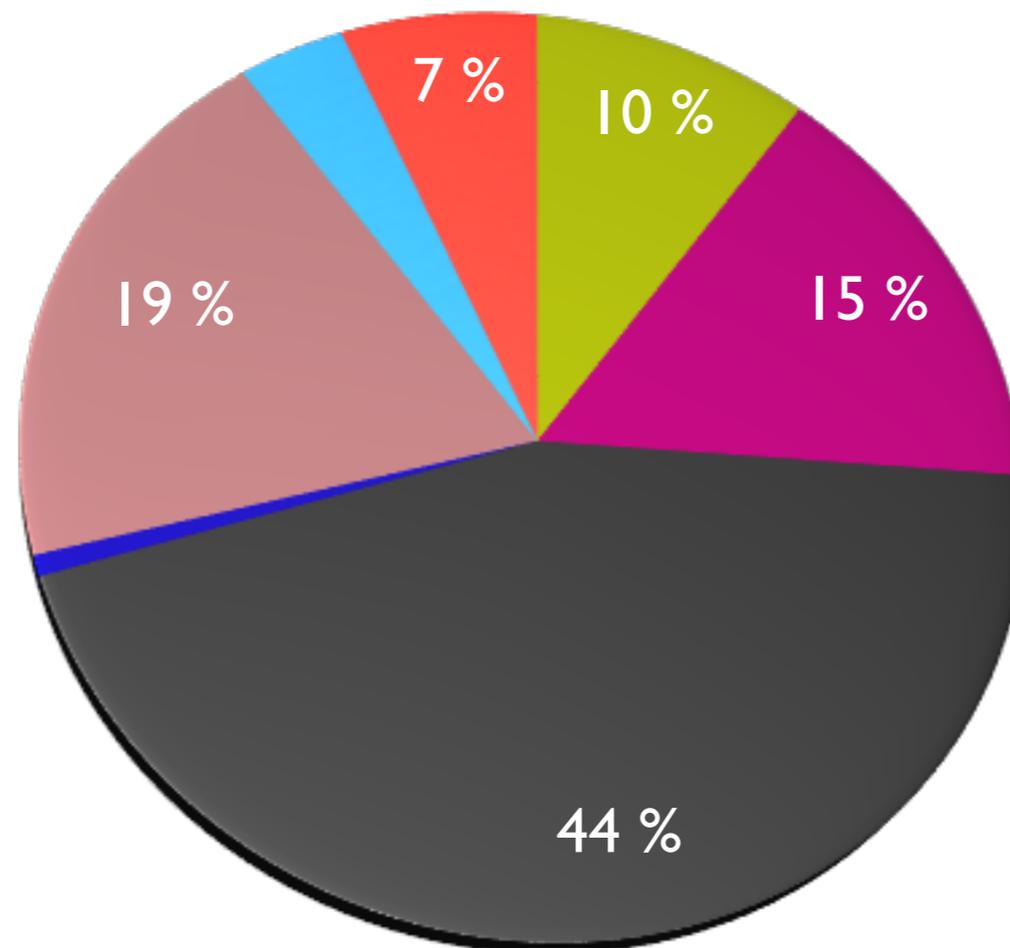
B2B Apps

Technologie

# Was fehlt

- eine Einführung in die Programmierung für Android
- Werbung für Google/Android
- Keine Codebeispiele
- Geräte verschenken
- auch nicht am Ende

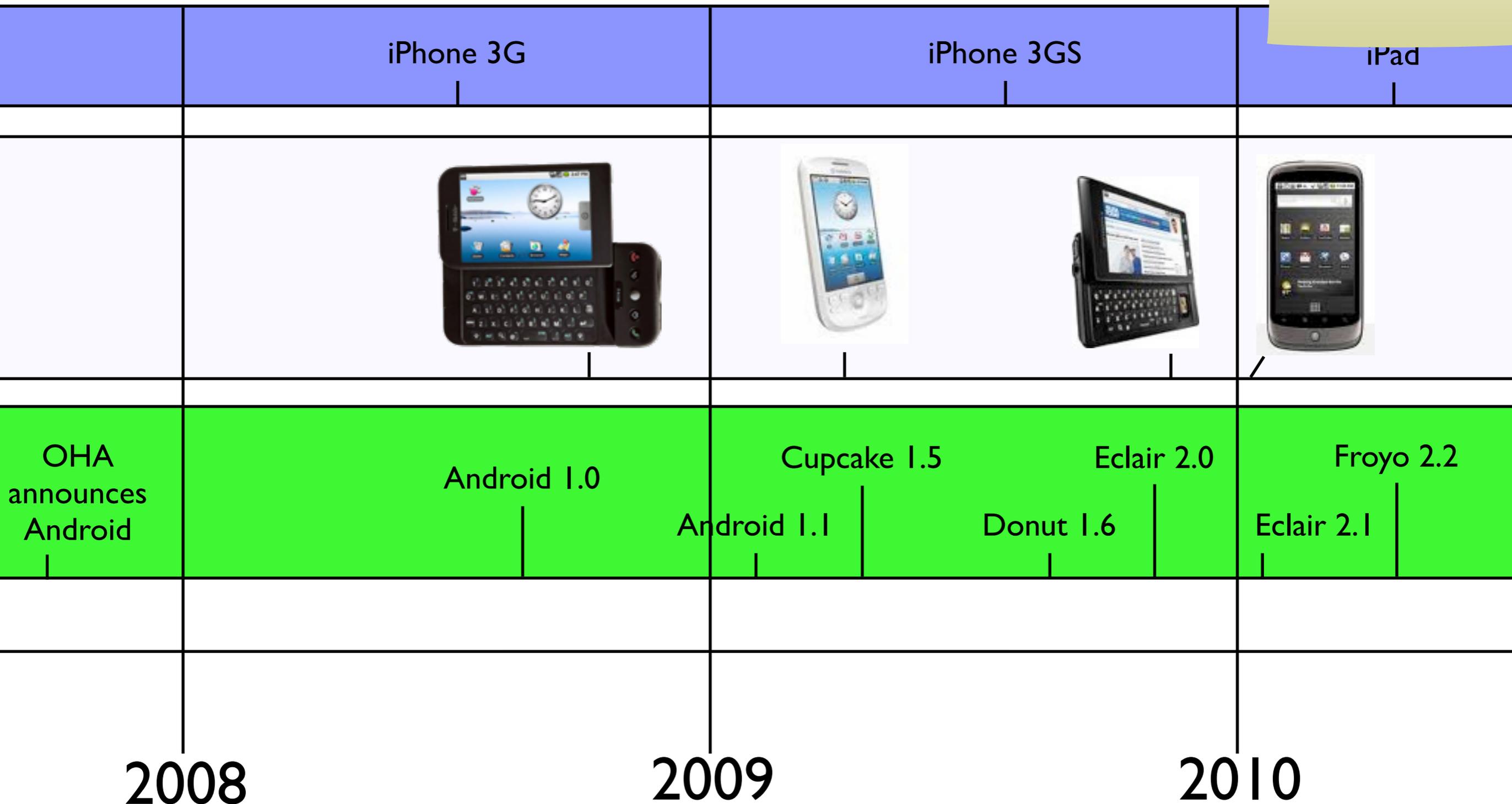
# Verkaufsanteil Mobiler Betriebssysteme



Quelle: Gartner, Weltweiter Verkauf mobiler Betriebssysteme, Q1/10

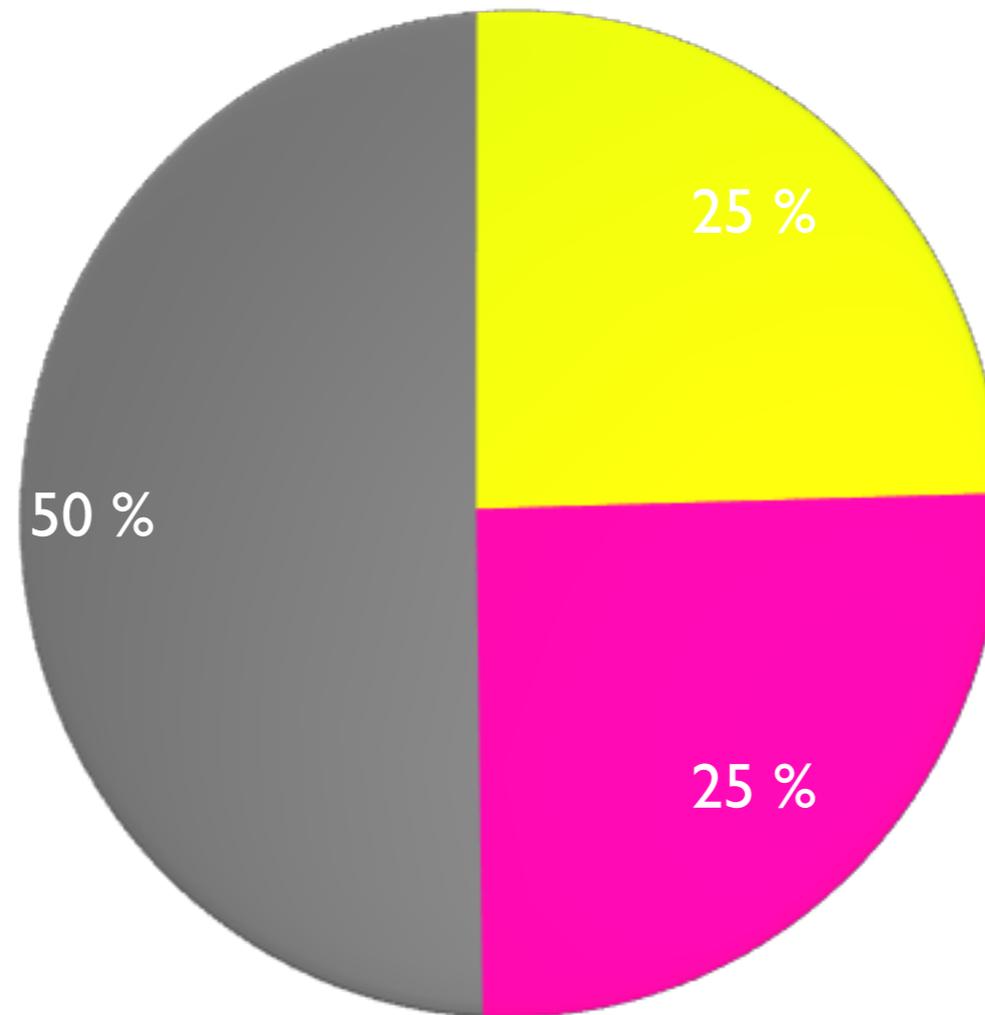
18 erhältlich, 67  
weltweit

# Versionsgeschichte



# Versionsverteilung

● Android 1.5      ● Android 1.6      ● Android 2.1



Quelle: Google, Zugriffe auf Android Market, Juni 2010

# Apps für Endkunden





# Apps für Endkunden

- Größter Anteil an Apps für Android
- Keine Zertifizierung wie bei Apples Appstore
- 25\$ Registrierungsgebühr für Market
- Regional begrenzte Verteilung möglich



# Monetarisierung Aktiv

- Aktiv: 30/70, davon 30% des Umsatzes an Google
- Zahlung nur über Google Checkout
- Apps können binnen 24h zurückgegeben werden, volle Rückerstattung inklusive



# Monetarisierung Passiv

- Google Mobile Ads, vorgestellt 2010
- In-App Werbung
- Dauerhafte Einnahmen auch nach Kauf
- Attraktiver für Endkunden?
- Nicht in jedem App-Konzept realisierbar



# Probleme

50% der iPhone Käufer  
21% der Android User  
Quelle: AdMob

- Im Moment sind die Kunden weniger kauffreudig als bei Apple
- Illegale Verbreitung von Apps?
- Paid Market nicht überall verfügbar!
- Kauf nur über Google Checkout
- Hardwarefragmentierung

# Android im Unternehmen?



# Erwartungen

- Einfache Entwicklung von Applikationen (Zeit & Kosten)
- Produktivitätssteigerung
- Erreichbarkeit
- Kontrolle
- Sicherheit

# Neue Möglichkeiten?

- Geolokalisation
- Internetzugang
- Kamera
- Schnelle Prozessoren
- Lagesensoren
- ?

# Entwicklung

- Kostenloses SDK, Plugin für Eclipse
- Einfacher Einstieg bei der Entwicklung, bekannte IDE (Eclipse), Verwendung von Java(-Syntax) & XML
- Plattformunabhängig (Linux, Mac, Windows)

# Android für Unternehmen?

- Programmier-/Laufzeitumgebung frei wählbar, z.B. auch Flash
- Philosophie der Offenheit (keine privaten APIs)
- Kein Desktop-Client à la iTunes notwendig
- Zugriff auf Gerätespeicher mittels Dateimanager
- Android ist im Kern Open Source (Security,...)

# Android für Unternehmen?

- Anbindung an Exchange
  - Sync von Adressen & E-Mails
  - Kein Sync von Kalender
- Unabhängige Verteilung von Apps ohne Google & Android Market
- Vollständig anpassbare GUI

# Froyo für Unternehmen?

- Starker Schritt in Richtung Unternehmenseinsatz
- Verbesserte MS Exchange Integration
  - Password Policies,
  - Address Lookup
  - Kalendersynchronisation
- Device Policy Management API

remote wipe -> aus der ferne löschen

# Froyo für Unternehmen?

push to phone,  
vertriebsbeispiel

- Remote Wipe
- Auto-Update von Applikationen
- Push-to-Phone Feature (Cloud 2 Device Messaging)

# Nachteile

- Starke, unbekämpfte Plattformfragmentierung
- Falls Nutzung mit Google-Account: Sicherheit, Datenschutz
- Google bietet keinen direkten Support
- Anpassungen verhindern native OS-Updates
- Keine native Vollverschlüsselung

# Zukunft

- Entwicklung des Marktes (iPhone, Blackberry, WinPhone7, Symbian, MeeGo, WebOS, Bada?)
- Android Integration mit Enterprise Backends (SAP, Lotus, ...)
- Web-Applikationen sollen sich immer mehr wie native Apps anfühlen, HTML5 (Zugriff auf Kamera, Lagesensor, etc.)
- Android 3.0 (Gingerbread)? Langsamere Release-Zyklen?

# Fazit

- Einfache Entwicklung von Applikationen (Zeit & Kosten)
  - Mit Froyo (2.2) gesteigerte Businessstauglichkeit
  - Android wird zunehmend interessant für die Anwendung im Enterprise-/Business-Umfeld
  - Anpassungen notwendig, nicht zu unterschätzen
- 

# Die Dalvik VM



# Hintergrund

- Dalvík: Hafenstadt in Island
- Entwickelt unter der Führung von Dan Bornstein, seit 2005 bei Google
- VM mit Fokus auf ressourcenbeschränkte Umgebungen
- Keine Java VM!

# Laufzeitverhalten

- Jede Applikation läuft in eigenem Prozess mit eigenem Speicher
- Sandboxing wichtig wegen
  - Stabilität
  - Sicherheit!

# Unterschiede zur JVM

- Dalvik benutzt DEX-Binaries, die aus .Class Dateien kompiliert werden
- .Class und .dex sind nicht binär kompatibel
- Lizenzfrage?

# DEX-Format

- „Captain Obvious“ meldet: DEX = Dalvik Executable
- Größte Optimierung statischer Art: globaler *Constant Pool* für alle Klassen eines Pakets.
- Andere Opcodes (2 byte statt 1 byte)
- Dx-tool konvertiert JAR nach DEX
- Bei der Installation wird noch optimiert, z.B. bei Byte Order, Inline Functions, Static linking...

# Garbage Collector

- Prinzipiell: Mark and Sweep
- Problematisch: Kommt der GC, ist für (geräteabhängig, da die Prozessorleistung variiert) ca. 200ms Pause
- GC fernhalten: Eigener Programmierstil
- Vgl. iPhone: Kein GC!

# Java für Android?



# Java für Android?

- Grundsätzlich vollkommen anderes Gesamtkonzept verglichen mit der Java ME
- „abgespeckte“ Java SE, basierend auf Apache Harmony (Open Sorce Java Runtime)
- Keine Java ME-Klassen
- Generics und Annotations
- Reflection

# Bestandteile



- Konsistente API auf allen Geräten mit gleicher Version
- Rasche Weiterentwicklung (von Google bestimmt)

# Java für Android?

- Exzessiver Gebrauch von `final`, `static`
- Vermeidung von Iteratoren, Collections, Enums, übermäßiger Objektallokation (Reuse, StringBuilder)
- Unterschied: Sehr bewusster Umgang mit Speicher notwendig, saubere Entwicklung.

# UI-Entwicklung

- Kein Swing, kein AWT
- Prinzipiell: Deklarativ mittels XML (wie in Silverlight oder JavaFX), oder programmatisch unter Benutzung von `android.*`
- Wenig Gemeinsamkeiten mit der traditionellen UI-Entwicklung in Java, abgesehen vom Event-Handling

# Fazit

- Bekannte Technologie, neue Anwendung
- Java als Köder?
- Revolutionär: Hochoptimierte VM inklusive eigenem Bytecode
- Nicht vergleichbar mit Java ME
- Ressourcen: <http://tinyurl.com/jfs2010>

**Fragen?**

**Danke!**



# Moritz Haarmann

# Benjamin Reimold

[m.haarmann@gmail.com](mailto:m.haarmann@gmail.com)

[@derwildemomo](#)

[benjamin.reimold@aformatik.de](mailto:benjamin.reimold@aformatik.de)

[@elektrojunge](#)



**XING** 

<http://gtugna.gtugs.org>

