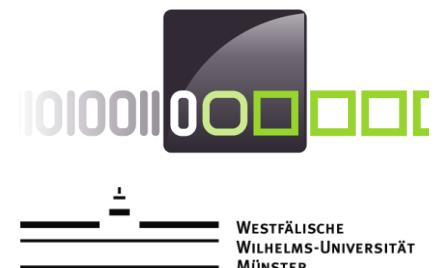




# PLATTFORMÜBERGREIFENDE ENTWICKLUNG VON APPS

DIPL.-WIRT.INFORM. HENNING HEITKÖTTER  
PRAKTISCHE INFORMATIK, UNIVERSITÄT MÜNSTER

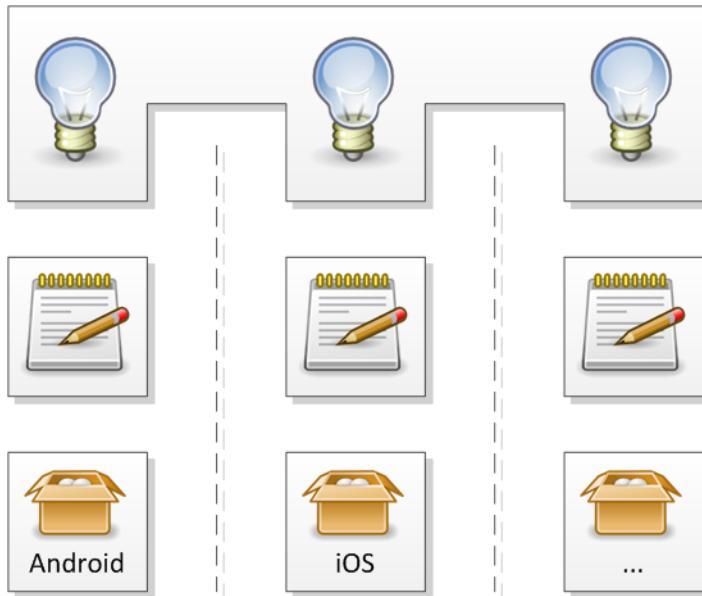


Themennachmittag „Business Apps“

10. April 2013

# GEGENÜBERSTELLUNG

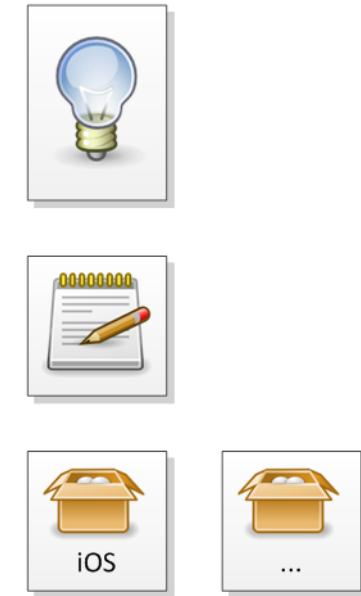
## NATIVE VS. PLATTFORMÜBERGREIFENDE ENTWICKLUNG



Definition &  
Entwurf

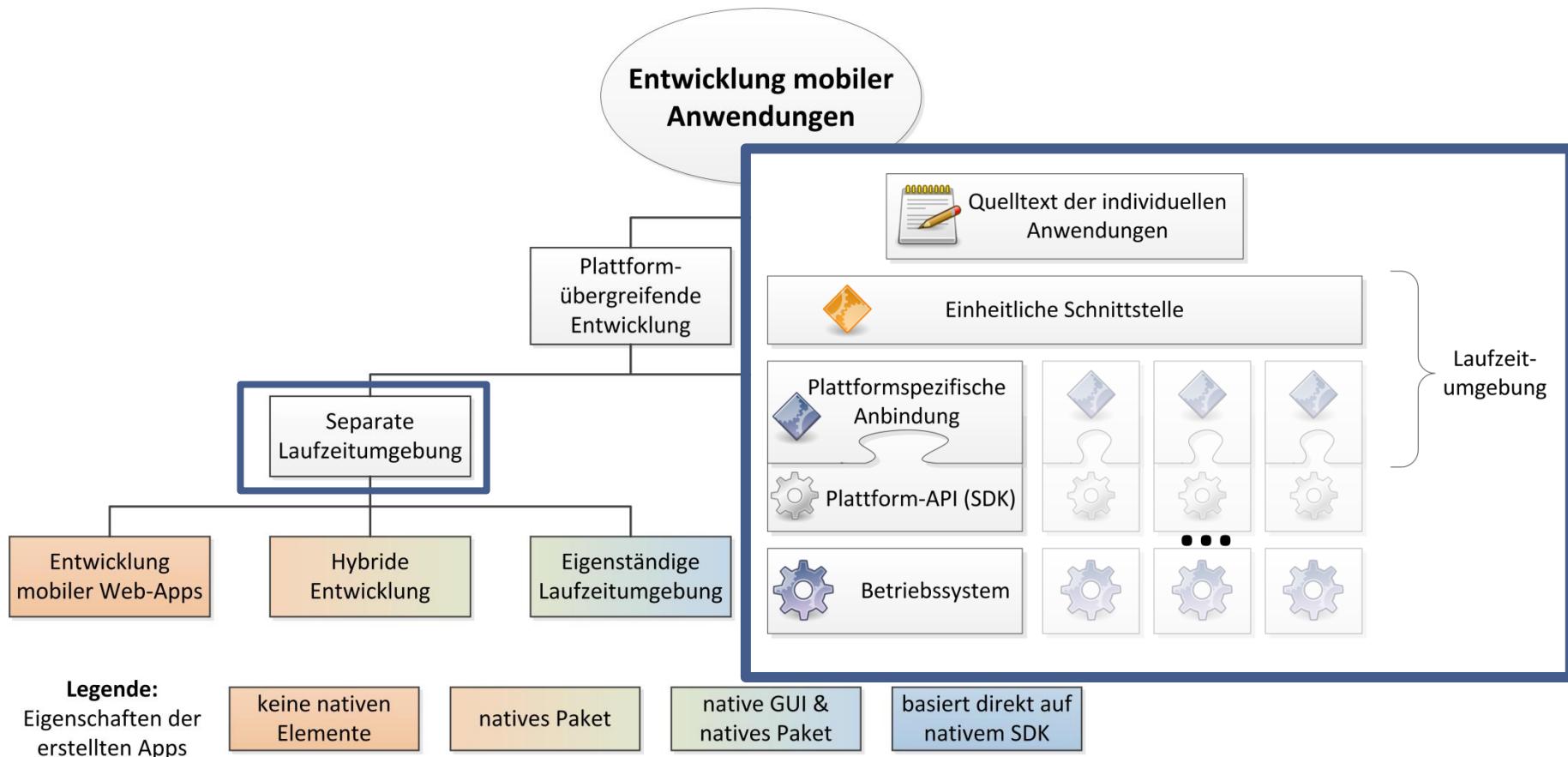
Implemen-  
tierung

Distribution  
& Testen



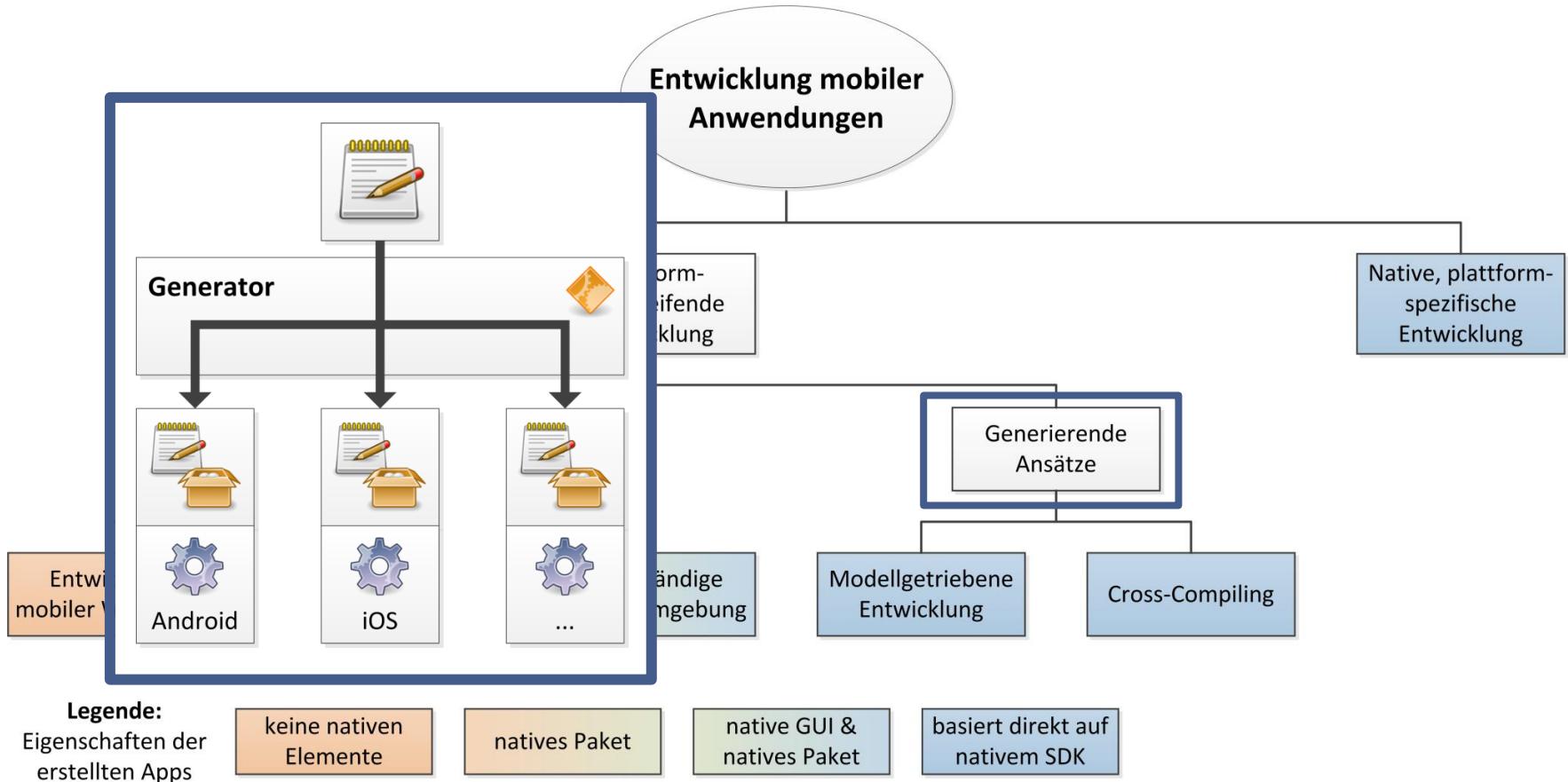
# ENTWICKLUNGSANSÄTZE

## KATEGORISIERUNG PLATTFORMÜBERGREIFENDER ENTWICKLUNG



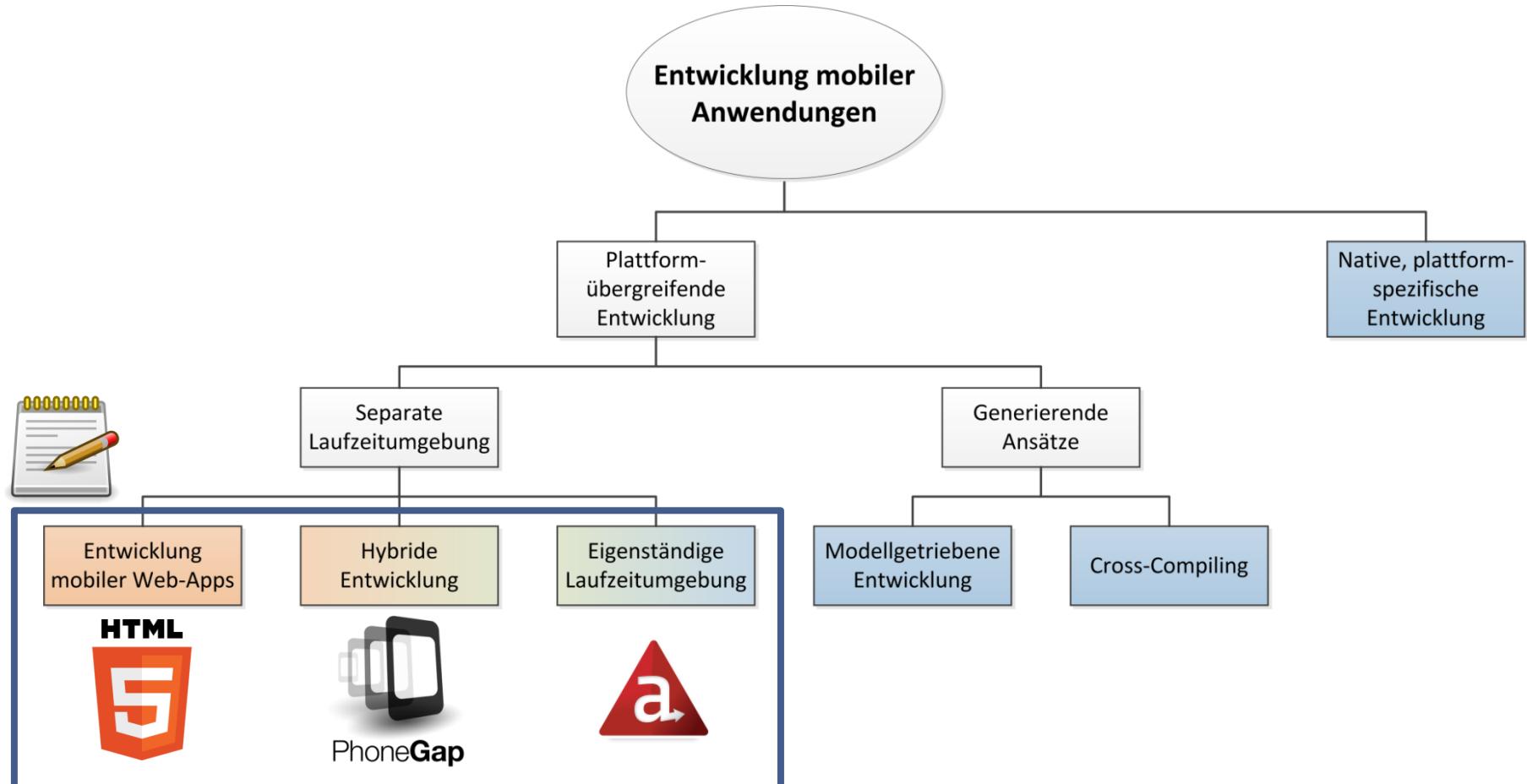
# ENTWICKLUNGSANSÄTZE

## KATEGORISIERUNG PLATTFORMÜBERGREIFENDER ENTWICKLUNG



# ENTWICKLUNGSANSÄTZE

## KATEGORISIERUNG PLATTFORMÜBERGREIFENDER ENTWICKLUNG



# KRITERIEN

## ANFORDERUNGEN AN PLATTFORMÜBERGREIFENDE ENTWICKLUNG

### Infrastruktur

- Lizenz und Kosten
- Unterstützte Plattformen
- Zugriff auf plattform-spezifische Funktionen
- Zukunftssicherheit
- Look & Feel
- Performance
- Distribution

### Entwicklung

- Entwicklungsumgebung
- GUI-Design
- Komfort des Entwicklungsprozesses
- Wartbarkeit
- Skalierbarkeit
- Migrationsoptionen
- Entwicklungsgeschwindigkeit und -kosten

# EVALUATION

## FRAMEWORKS IM VERGLEICH



Kriterium	Web-App	PhoneGap	Titanium	Nativ
Lizenz und Kosten	3	2	5	3
Unterstützte Plattformen	1	2	4	6
Plattformspezifische Funktionen	5	2	2	1
Zukunftssicherheit	1	2	3	1
Look & Feel	4	3	2	1
Performance	3	1	5	1
Distribution	3	2	2	2
Entwicklungsumgebung	2	2	3	2
GUI-Design	1	1	4	1
Komfort des Entwicklungsprozesses	2	2	3	2
Wartbarkeit	1	1	3	3
Skalierbarkeit	2	2	2	1
Migrationsoptionen	1	2	5	6
Entwicklungsgeschwindigkeit und -kosten	1	1	5	5

# PROBLEMATIK

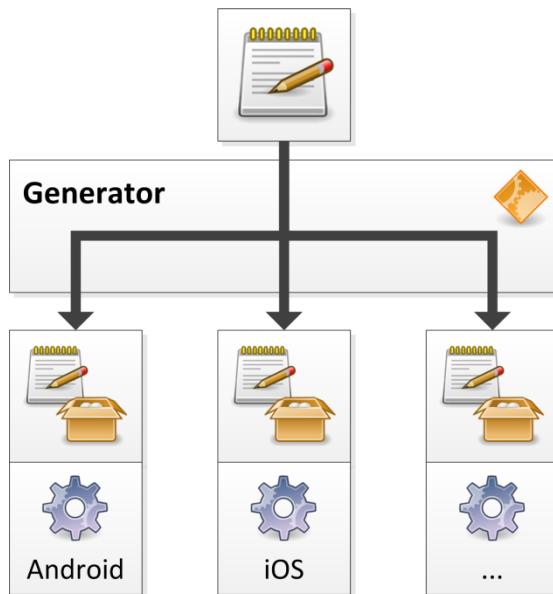
## VOR- UND NACHTEILE AKTUELLER ANSÄTZE

### Native Entwicklung

- + Natives Look & Feel
- Vielfacher Aufwand
- Niedriges Abstraktionsniveau

### Web-basierte Entwicklung

- Look & Feel einer Webseite
- + Einmaliger Aufwand
- ? Mittleres Abstraktionsniveau

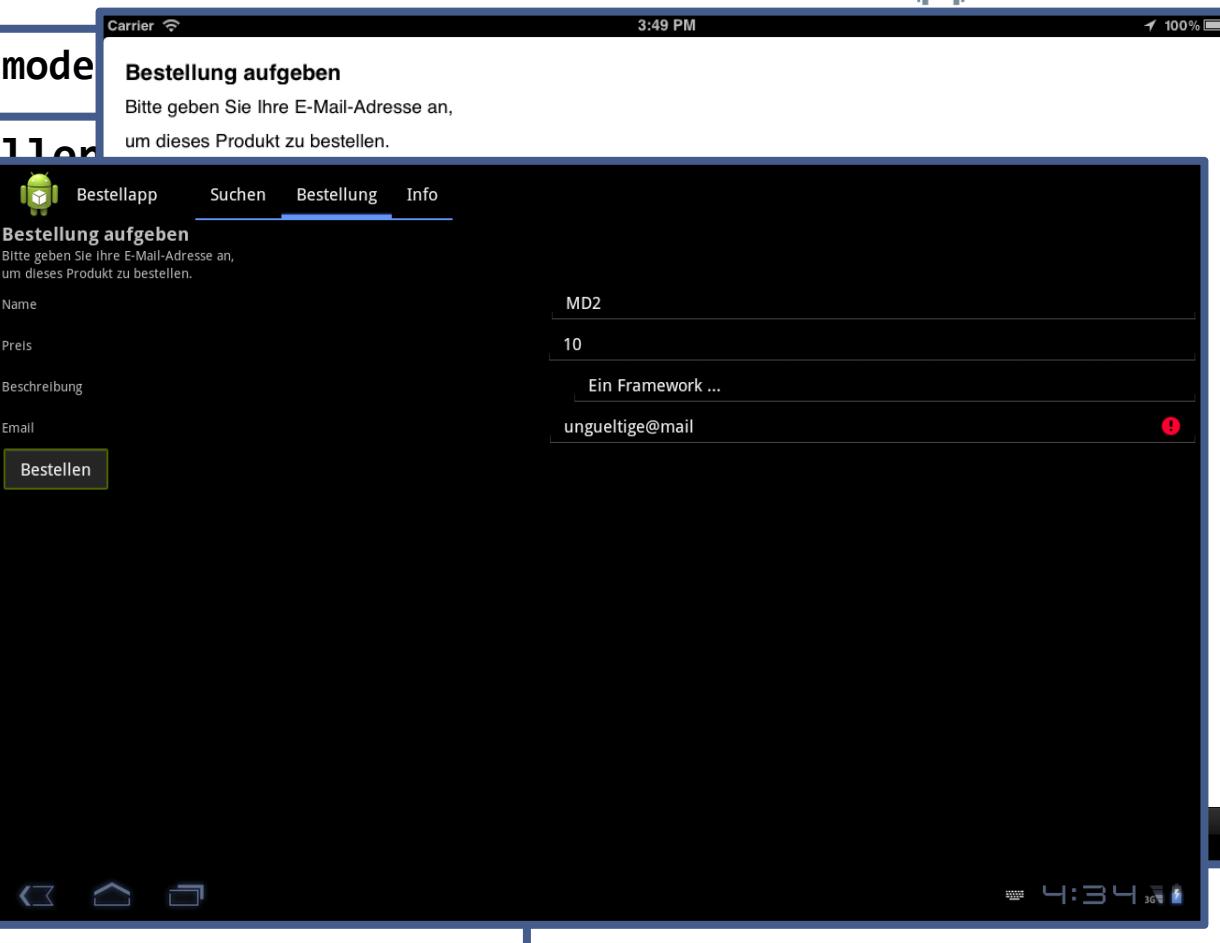


- Ein prägnantes textuelles Modell in domänenspezifischer Sprache
- Code-Generatoren erzeugen Apps für Android und iOS
  - Automatisch
  - Basierend auf nativem SDK
- Kompilierte Apps
  - Lauffähig ohne Anpassung
  - Verwenden nur native Elemente

# BEISPIEL-APP

## MD<sup>2</sup>-MODELL UND GENERIERTE APPS

```
package de.md2.bestellung.mode
en
en    FlowLayoutPane Besteller
        Label bestellenL
            text "Bestell
            style Gross
}
en
en    Label info("Bitte
        AutoGenerator be
            contentProvid
            bestellungPro
}
Button bestellen
}
```



# FAZIT

## PLATTFORMÜBERGREIFENDE ENTWICKLUNG?



- Verschiedene Ansätze zur plattformübergreifenden Entwicklung
  - Leistungsfähige Lösungen
  - Auswahl sollte wohlüberlegt geschehen
  - Insbesondere Web-basierte Frameworks vielversprechend
  - Kontinuierliche Weiterentwicklung
- 
- Lösung für natives Look & Feel bisher nicht verfügbar
  - Modellgetriebener Ansatz mit **M2D<sup>2</sup>** in der Erprobung
  - Machbarkeit demonstriert, Evaluation und weitere Forschung notwendig



## Weiterführende Informationen →



# VIELEN DANK FÜR IHRE AUFMERKSAMKEIT



DIPL.-WIRT.INFORM. HENNING HEITKÖTTER  
HEITKOETTER@WI.UNI-MUENSTER.DE

WWW.IAI.UNI-MUENSTER.DE/PUBLIKATIONEN/IAI4.PDF

THE IS RESEARCH NETWORK

[www.ercis.org](http://www.ercis.org)