

# Tamino Mobile

---

**Ralf Rutke**  
**Director Mobile Computing**  
**Software AG**

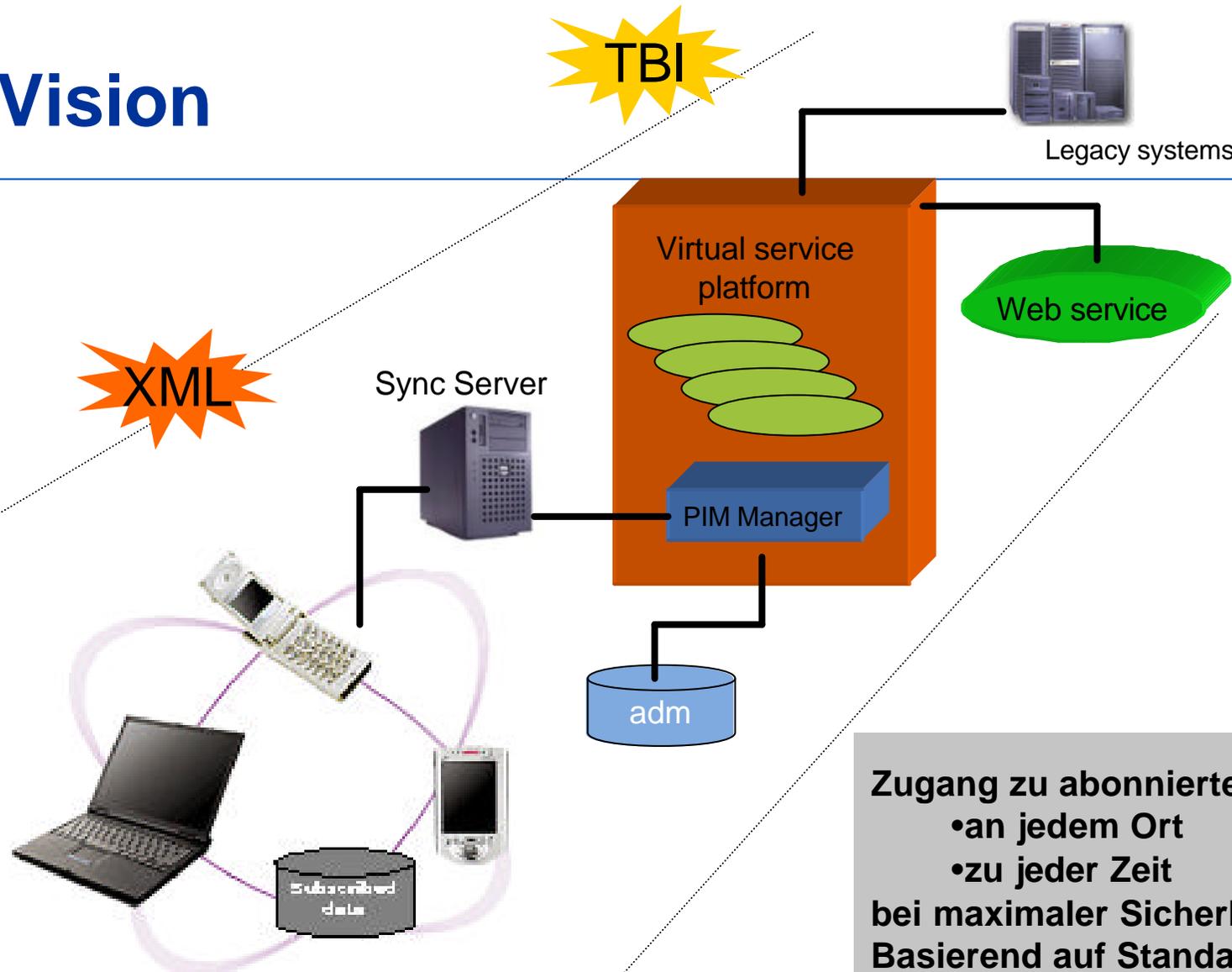
[Ralf.rutke@softwareag.com](mailto:Ralf.rutke@softwareag.com)

# Überblick

---

- **Mobile Datenbanken**
  - ◆ Vision
  - ◆ Herausforderungen
  - ◆ Kritische Erfolgsfaktoren
- **Tamino Mobile**
  - ◆ Footprint
  - ◆ Performance
  - ◆ Programmer value
  - ◆ Robustness
- **Tamino Mobile im Kontext**

# Vision



Zugang zu abonnierten Informationen  
•an jedem Ort  
•zu jeder Zeit  
bei maximaler Sicherheit  
Basierend auf Standardtechnologien

Step into the wireless future

## Herausforderungen: Mobile Datenbanken

---

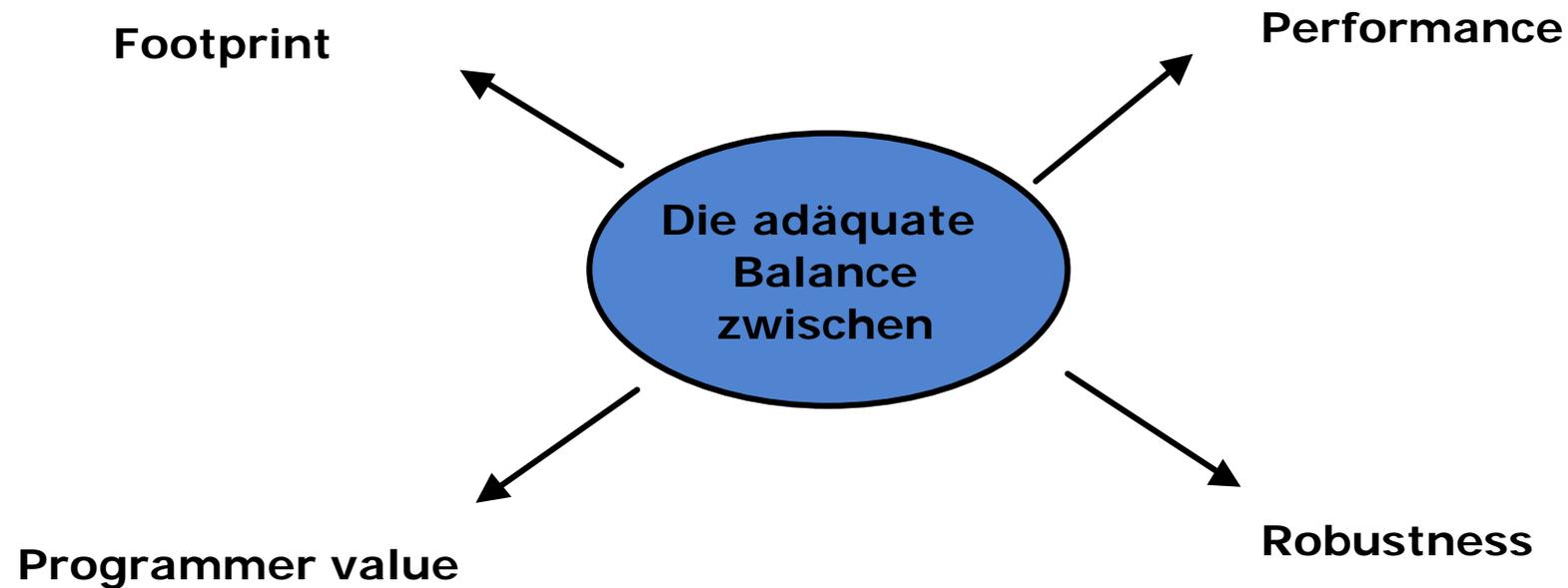
- **Nutzung von Applikationen und Informationen unabhängig von der Verfügbarkeit von Netzwerken.**
- **Unterstützung unterschiedlicher Datenformate**
- **Optimierte Nutzung limitierter Ressourcen**
- **Breite Plattformunterstützung**
- **Universell verwendbare, standardisierte APIs**
- **Applikations- und Synchronisierungsmodell**

# Herausforderungen: Server kontra PDA

<b>Server</b>	<b>PDA</b>
<b>Multi user</b>	<b>Single user</b>
<b>Heavy I/O</b>	<b>No or little I/O</b>
<b>Abundant memory</b>	<b>Limited memory</b>
<b>Powerful processor(s)</b>	<b>Limited processor capacity</b>
<b>Always on-line</b>	<b>Often off-line</b>

Step into the wireless future

# Kritische Erfolgsfaktoren



# Kritische Erfolgsfaktoren

---

Tamino Mobile 3.0

# Tamino Mobile: Footprint

---

## ■ Eigenschaften

- ◆ Kompakter und modularer Code
- ◆ Geringer Bedarf für administrative Daten

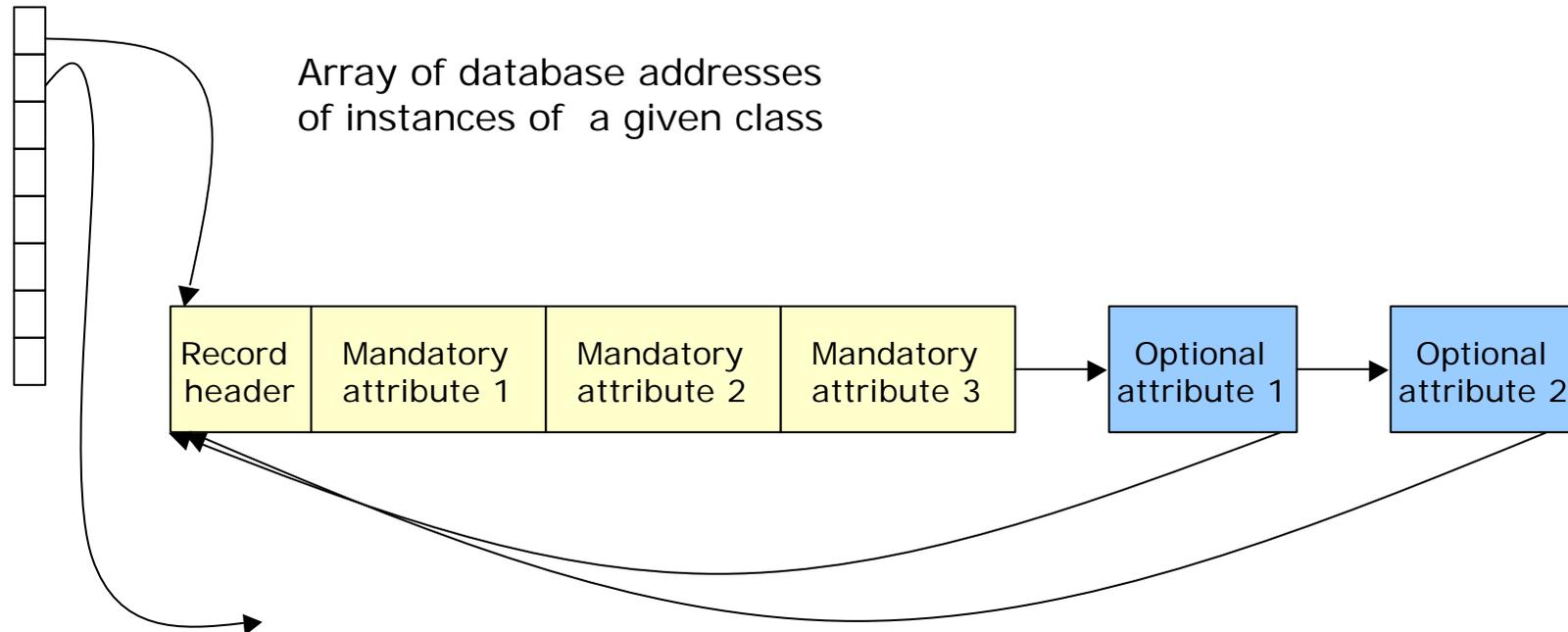
## ■ Vorteile

- ◆ Optimale Ausnutzung limitierter Ressourcen auf mobilen Endgeräten
- ◆ Verbleibende Ressourcen stehen der Anwendung zur Verfügung

## Tamino Mobile: Footprint / Implementierung

- Tamino Mobile speichert mandatory Attribute eines Objekts zusammenhängend ab.
- Instanzen optionaler Attribute werden dynamisch erzeugt und zerstört. Die physische Allokation erfolgt daher außerhalb vom abhängigen Objekt.
- Optionale Attribute werden physisch als Link-Liste implementiert. Der Listen-Header verbleibt im Header des Objekts.

# Tamino Mobile: Performance Objektlayout



- Blobs und Texte werden außerhalb des Objekts allokiert. Alle weiteren Datentypen werden innerhalb allokiert.
- Das Datenbank Schema speichert den Informationen über die relative Startadresse eines mandatory Attributs.
- Tamino Mobile nutzt balanced binary trees.

Step into the wireless future

# Tamino Mobile: Performance/Indexierung

## ■ Drei Typen der Indexierung

- ◆ Ein einzelnes Attribut
- ◆ Eine Attributgruppe
- ◆ Ein Kombination von Attributen unterschiedlicher Klassen

## ■ Definition von Indices

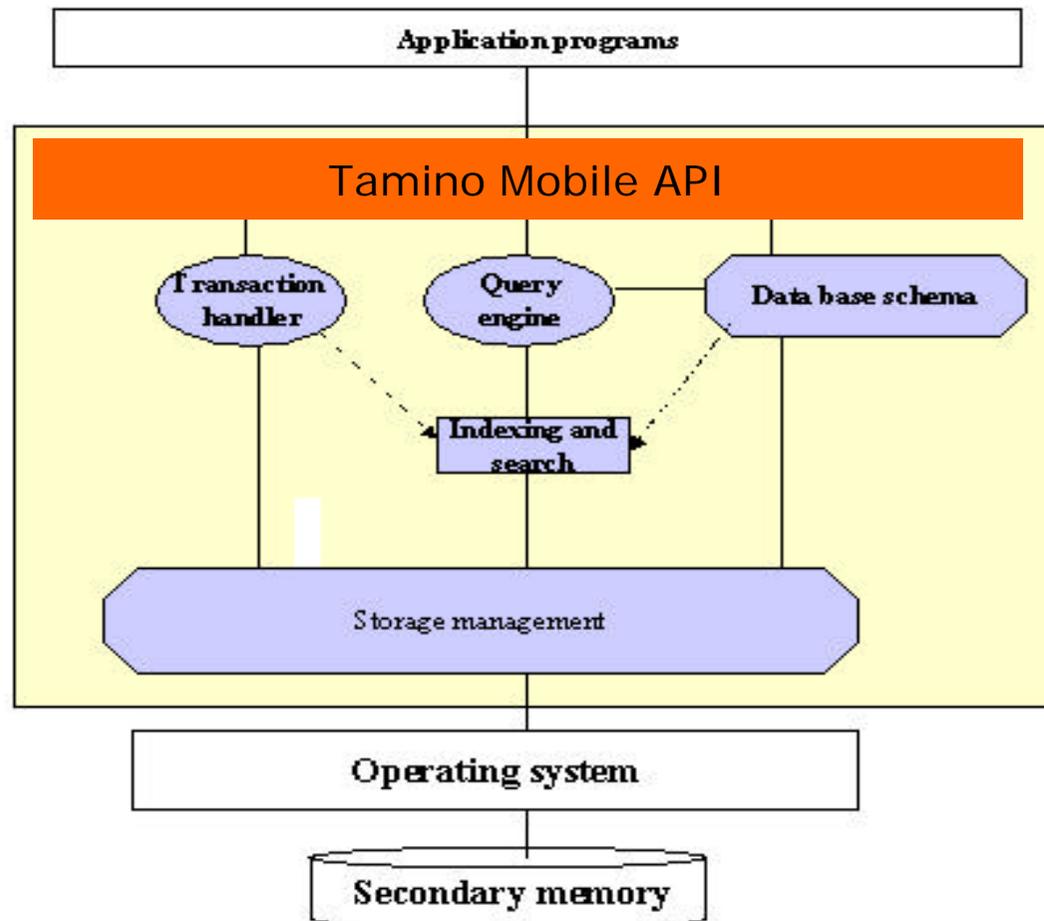
- ◆ Ein Index wird im Datenbankschema definiert
- ◆ Jede Datenmanipulation führt zu einem automatischen Update der Indexstruktur.
- ◆ Indices können dynamisch eingerichtet werden

## Tamino Mobile: Programmer value

---

- Unterstützung unterschiedlicher Datenformate
- Unterstützung mehrerer Programmiersprachen
- Natives C++ Interface
- Direkte XML (SAX, DOM, XPATH API) Unterstützung
- HTTP Interface

# Tamino Mobile: Robustness



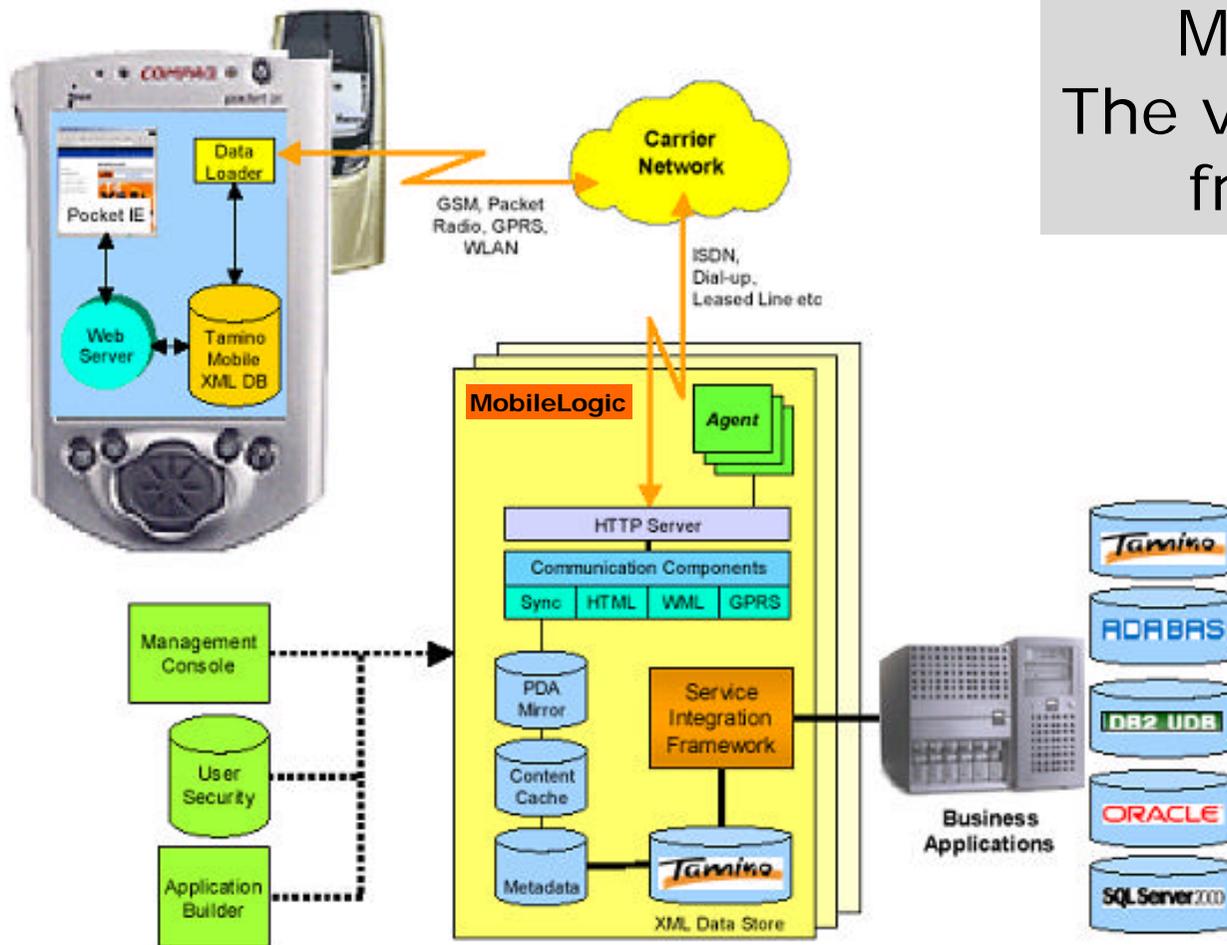
Step into the wireless future

# Kritische Erfolgsfaktoren

---

Tamino Mobile  
Im  
Kontext

# Tamino Mobile im Kontext



MobileLogic  
The virtual Service  
framework

Step into the wireless future