

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Universeller Aufgabenclient

Erfahrungsbericht

Sven Koschnicke Christian Wulf Florian Fittkau

10.11.2010

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Outline

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement
- 5 Werkzeuge
- 6 Fazit

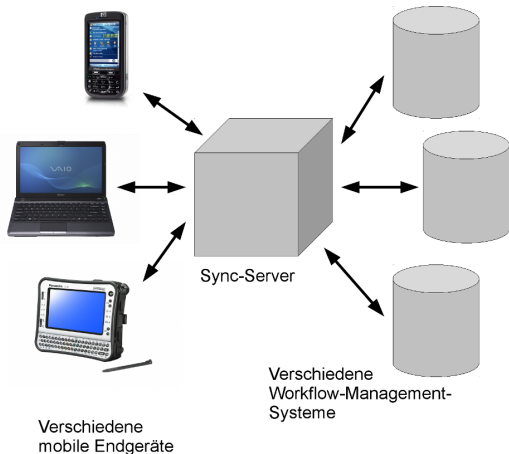
1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement
- 5 Werkzeuge
- 6 Fazit

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

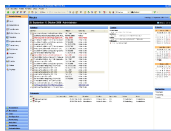
Projektziel

Erstellung eines universellen Aufgabenclients zum Workforce-Management im Feld.



1. Motivation
2. Rmobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Universell



1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement
- 5 Werkzeuge
- 6 Fazit

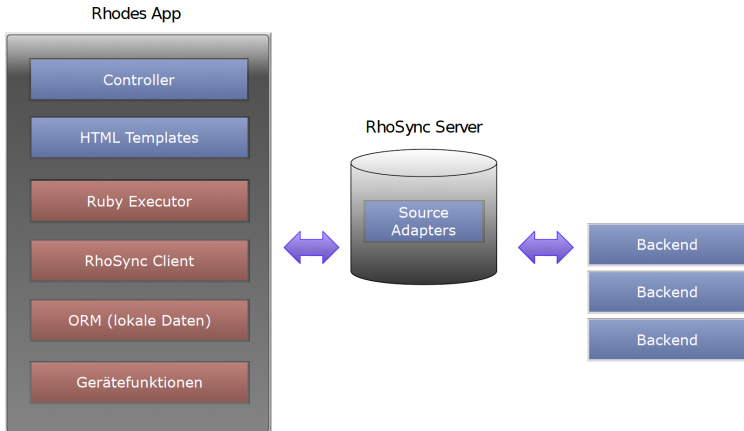
1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Das Rhomobile Framework

- Plattformunabhängige Entwicklung
- Flexible Darstellung durch Nutzung von HTML, CSS und nativen Elementen
- Synchronisations-Komponente mit flexiblem Datenschema

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Wie funktioniert Rhomobile?



1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

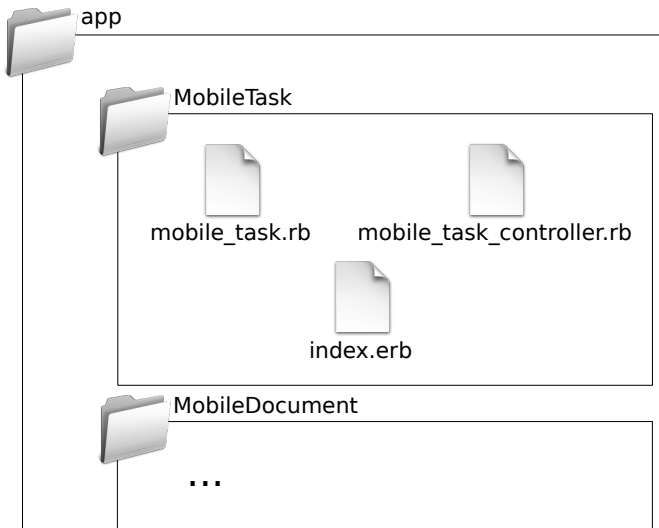
Beispiel

- Anzeige einer Liste von Aufgaben
- markieren einer Aufgabe als erledigt



1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Struktur des Frameworks



1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Model

mobile_task.rb

```
1 class MobileTask
2
3   include Rhom::FixedSchema
4
5   property :description, :string
6   property :status, :string
7
8   def open?
9     status == "open"
10  end
11
12 end
```

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Controller

mobile_task_controller.rb

```
1 class MobileTaskController < Rho::RhoController
2
3   def index
4     @mobiletasks = MobileTask.find_all
5
6     render :action => :index
7   end
8
9 end
```

View

index.erb

```
1 <% if @mobiletasks && @mobiletasks.length > 0 %>
2   <table>
3     <% @mobiletasks.each do |task| %>
4       <tr>
5         <td>
6           <% if task.open? %>
7             <%= link_to "[_]", :action => :mark_done,
8               :id => task.object %>
9           <% end %>
10          <% if task.done? %>[x]<% end %>
11        </td>
12        <td> <%= task.description %> </td>
13      </tr>
14    <% end %>
15  </table>
16 <% end %>
```

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Markieren einer Aufgabe als erledigt

mobile_task_controller.rb

```
1 class MobileTaskController < Rho::RhoController
2
3   def index
4     ...
5   end
6
7   def mark_done
8     mobiletask = MobileTask.find @params[id]
9     mobiletask.status = "done"
10    mobiletask.save
11
12    redirect_to :action => :index
13  end
14
15 end
```

1. Motivation
2. Rhomobile Framework
- 3. Desktop Client**
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client**
- 4 Dokumentenmanagement
- 5 Werkzeuge
- 6 Fazit

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

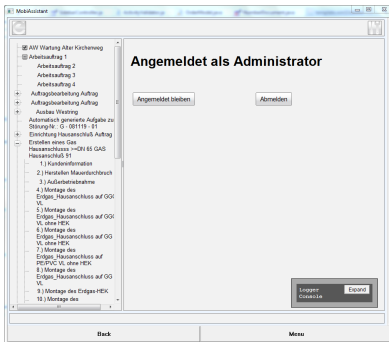
Anforderungen

- Eine zusätzliche Windows-Version für PCs
- Windows-ähnliches Aussehen inkl. besserer Bedienbarkeit
- Erweiterte Features (z.B. dynamische Kartennavigation)
- Gewünscht: Hohe Wiederverwendbarkeit des Quelltextes der mobilen Variante

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

1. Versuch

- Erweiterung der Win32-Debugging-Anwendung
- Nutzung des auf JS-basierenden Rich-Client-Web-Frameworks **Yahoo! User Interface Library (YUI)**¹
- Angepasste Views, geringe Änderungen in den Controllern



1. Motivation
2. RhoMobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

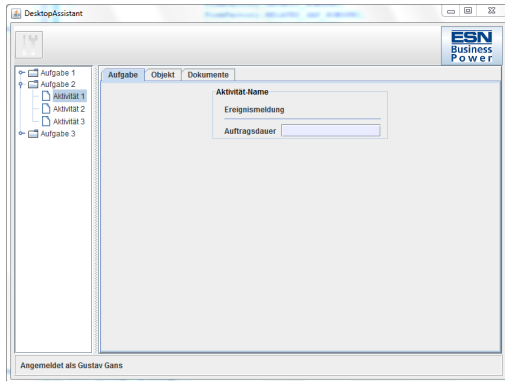
Probleme

- Aufwendig \Rightarrow langsame Entwicklungsgeschwindigkeit
- Ineffektiv (unnötige, unvorhergesehene Nebenwirkungen)
 - \Rightarrow RhoMobile-Framework ist nicht dafür gemacht
- Ineffizient

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

2. Versuch

- Native Java-Implementierung
- JS-Synchronisation-Bibliothek per Javas ScriptEngine
- Swing & jgoodies ⇒ Googles WindowBuilder Pro
- Work in progress...



1. Motivation
2. Rhomobile Framework
3. Desktop Client
- 4. Dokumentenmanagement**
5. Werkzeuge
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement**
- 5 Werkzeuge
- 6 Fazit

Dokumentenmanagement

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

- im Feld Dokumente bereithalten
- Fotos aufnehmen
- nicht vom Framework vorgesehen

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Dokumentenmanagement

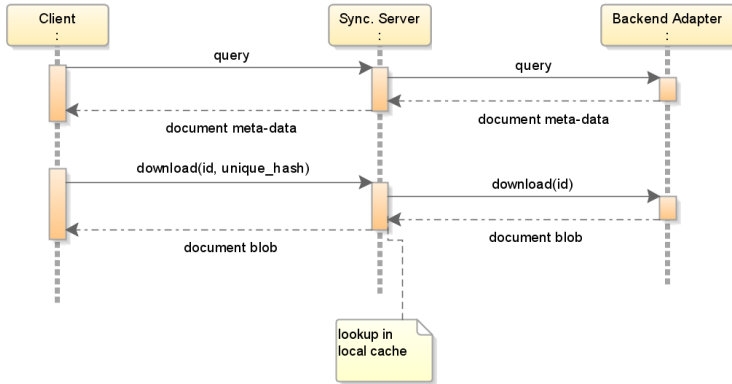


Abbildung 1: Synchronisationsprozess

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Übersicht

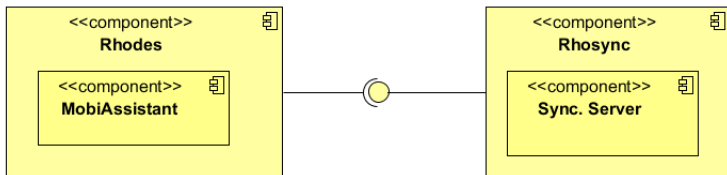
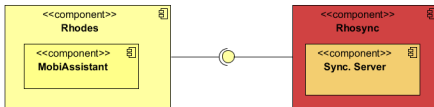


Abbildung 2: Architektur Übersicht

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Erweitern von RhoSync

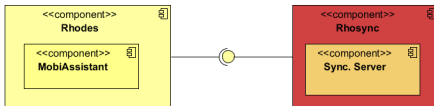


server.rb

```
1 get /application/download_file do
2   catch_all do
3     filename =
4       FileProcessor.download_file(current_client, params)
5     send_file(filename)
6   end
7 end
```


1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Erweitern von RhoSync

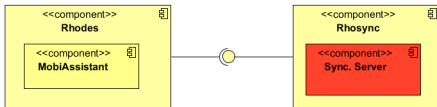


file_processor.rb

```
1 def download_file(client, params=nil)
2   client.app.download_file params[:id], params[:unique_hash]
3 end
```

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Sync. Server

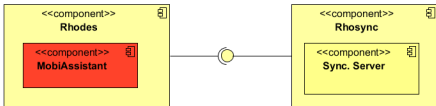


application.rb

```
1 def download_file(id, unique_hash)
2   file_name = File.join(get_storage_dir, id.to_s)
3   if not check_same_document(file_name, unique_hash, id)
4     get_document_from_backend(id, file_name)
5   end
6   return file_name
7 end
```

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Clientbefehl



Herunterladen einer Datei

```
1 Rho::AsyncHttp.download_file(  
2   :url => syncserver + /download_file?client_id= +  
3     client_id + "&id=" + id + "&unique_hash=" + unique_hash,  
4   :filename => filename,  
5   :headers => {Cookie => session},  
6   :callback => callback,  
7   :callback_param => params)
```

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. **Werkzeuge**
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement
- 5 **Werkzeuge**
- 6 Fazit

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

Werkzeuge

- Versionsverwaltungssystem: git (dezentral)
 - Pro: Fork-Feature (github), Gute Merge-Funktionalität, Eclipse-Integration
 - Contra: Komplizierter als Subversion, Dezentralisation nicht benötigt
- Emulatoren sehr hilfreich (kürzere Deploy-Zeit, schnellere Reaktionszeit)
- Test-Framework mspec
- Continuous-Integration-Server

1. Motivation
2. Rhomobile Framework
3. Desktop Client
4. Dokumentenmanagement
5. Werkzeuge
6. Fazit

- 1 Motivation
- 2 Rhomobile Framework
- 3 Desktop Client
- 4 Dokumentenmanagement
- 5 Werkzeuge
- 6 Fazit

1. Motivation
2. Rhomobile
Framework
3. Desktop Client
4. Dokumenten-
management
5. Werkzeuge
6. Fazit

Fazit

- junges Framework -> noch einige Bugs, Dokumentation lückenhaft und nicht immer aktuell
- aber: gute Zusammenarbeit mit den Entwicklern
- dank open source gut anpassbar und erweiterbar
- Test-Driven-Development
- keine volle Unterstützung für Desktops