



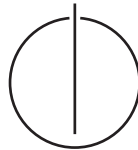
FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik

**Myriad – a mailmerge tool for massive  
parallel, yet individual email conversations**

Ludwig Schubert







# FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik

Myriad – a mailmerge tool for massive parallel, yet  
individual email conversations

Myriad – ein Serienbrief Email-Tool für hochgradig  
parallele, jedoch individualisierte Emailkonversationen

Author: Ludwig Schubert

Supervisor: Prof. Dr. Joe Don

Advisor: Dipl.Inf. Jon Doe

Date: August 15, 2013





Ich versichere, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 17. Juli 2013

Ludwig Schubert



---

## **Abstract**

This thesis introduces the myriad system for email mass communication.

Despite it's age, Email has remained the prevalent form of electronic communication, it's usage being wildly different from how it was imagined. The tools to handle it, however, are still stuck in their original UI metaphors.

Myriad aims at producing personalized communication on a comparable level to manually composed messages, while reducing user effort. A cross-over of mailmerge and customer support/helpdesk software, it enables managing big volumes of bidirectional email-based communication. It is based on a self-developed framework for separating information extraction, decision- making, and personalization steps in communication.

The myriad system consists of a server component that handles interfacing with email servers, a core logic system and a web frontend for users.

It can be tested at <http://myriad.ludwigschubert.de>.

## **Keywords**

Email, Workflow, Helpdesk, Mailmerge, Crowdsourcing, Personalized Communication, Assisted Templating, Generated Documents, Web-Application





---

## **Acknowledgments**

If someone contributed to the thesis... might be good to thank them here.



# Contents

<b>Disclaimer</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Table of Contents</b>	<b>xi</b>
 <b>Main Matter</b>	 <b>3</b>
<b>1. Introduction</b>	<b>3</b>
1.1. Motivation . . . . .	3
1.2. Goals and Background . . . . .	3
1.3. Outline . . . . .	3
 <b>2. Technical Backgrounds</b>	 <b>5</b>
2.1. Web Applications . . . . .	5
2.1.1. Why build a web application instead of a native one? . . . . .	5
2.1.2. Main Components of a Web Application . . . . .	5
2.1.3. Rails and Ruby . . . . .	5
2.2. Email Systems . . . . .	5
2.2.1. Working with RFC 2822 . . . . .	5
2.2.2. IMAP and SMTP . . . . .	5
2.3. Workflow Systems . . . . .	5
2.3.1. History of Workflow Systems . . . . .	5
2.3.2. Famous Examples . . . . .	5
 <b>3. Comparison with similar systems</b>	 <b>7</b>
3.1. Mailmerge Systems . . . . .	7
3.1.1. CRM Systems . . . . .	7
3.1.2. Dedicated Mailmerge Systems . . . . .	7

3.1.3.	Backend Services . . . . .	7
	Amazon SES . . . . .	7
	SendGrid . . . . .	7
3.2.	Customer Support Systems . . . . .	7
<b>4.</b>	<b>Concept</b>	<b>9</b>
4.1.	Functional Analysis . . . . .	10
4.2.	Product Functions . . . . .	10
4.3.	User Interface . . . . .	10
4.4.	Technical Analysis . . . . .	10
	4.4.1. Runtime Environment . . . . .	10
	4.4.2. Server Software Stack . . . . .	10
	4.4.3. Client Side . . . . .	10
	4.4.4. Backend Service Connections . . . . .	10
	Google Mail . . . . .	10
	Google Docs . . . . .	10
4.5.	System Design . . . . .	10
	4.5.1. Database Schema . . . . .	10
	Uncommon Patterns . . . . .	10
	4.5.2. Distribution of System Components . . . . .	10
<b>5.</b>	<b>Implementation</b>	<b>11</b>
5.1.	Preparation and Tools . . . . .	12
	5.1.1. Development Environment . . . . .	12
	5.1.2. Collaborative Development . . . . .	12
	git, gitflow and Github . . . . .	12
	5.1.3. Deployment . . . . .	12
	Deployment Tool Capistrano . . . . .	12
5.2.	Server Component . . . . .	12
	5.2.1. Core Classes and their Interaction . . . . .	12
	Contact . . . . .	12
	Template . . . . .	12
	Email . . . . .	12
	Campaign . . . . .	12
	Conversation . . . . .	12
	Key & Value . . . . .	12
	Search & KeyBinding . . . . .	12
	Notification . . . . .	12
	5.2.2. Workers and their Jobs . . . . .	12

5.2.3. Maintenance and Rake Tasks . . . . .	12
5.3. Backend Services Connection . . . . .	12
5.3.1. Email Fetching . . . . .	12
IMAP IDLEing . . . . .	12
5.3.2. Google Docs Syncing . . . . .	12
5.4. “Best Practices” . . . . .	12
5.4.1. Core Extensions . . . . .	12
5.4.2. Lean Workers with AbstractWorker . . . . .	12
5.4.3. Monitoring Services . . . . .	12
<b>6. Evaluation</b>	<b>13</b>
6.1. Comparison with Initial Goals . . . . .	13
6.2. Observed Use Cases . . . . .	13
6.2.1. Recruiting Exchange Students for one of the Most Desired Universi- ties of the World . . . . .	13
6.2.2. Requesting paper Reviews for a Journal . . . . .	13
6.2.3. Managing Incoming Class Administration Emails . . . . .	13
<b>7. Conclusion</b>	<b>15</b>
7.1. Conclusion of this work . . . . .	15
7.2. Discussion of results . . . . .	15
7.3. Future Work . . . . .	15
<b>Bibliography</b>	<b>17</b>
<b>List of Figures</b>	<b>19</b>
<b>Appendix</b>	<b>23</b>
<b>A. Detailed Descriptions</b>	<b>23</b>
<b>B. Colophon</b>	<b>25</b>



# **Main Matter**





# **1. Introduction**

This is the introduction.

## **1.1. Motivation**

## **1.2. Goals and Background**

## **1.3. Outline**

Hallo Latex: [1] Danke Latex!



## **2. Technical Backgrounds**

### **2.1. Web Applications**

#### **2.1.1. Why build a web application instead of a native one?**

#### **2.1.2. Main Components of a Web Application**

#### **2.1.3. Rails and Ruby**

### **2.2. Email Systems**

#### **2.2.1. Working with RFC 2822**

#### **2.2.2. IMAP and SMTP**

### **2.3. Workflow Systems**

#### **2.3.1. History of Workflow Systems**

#### **2.3.2. Famous Examples**



## **3. Comparison with similar systems**

### **3.1. Mailmerge Systems**

#### **3.1.1. CRM Systems**

#### **3.1.2. Dedicated Mailmerge Systems**

#### **3.1.3. Backend Services**

**Amazon SES**

**SendGrid**

### **3.2. Customer Support Systems**





## 4. Concept

### 4.1. Functional Analysis

### 4.2. Product Functions

### 4.3. User Interface

### 4.4. Technical Analysis

#### 4.4.1. Runtime Environment

#### 4.4.2. Server Software Stack

#### 4.4.3. Client Side

#### 4.4.4. Backend Service Connections

Google Mail

Google Docs

### 4.5. System Design

#### 4.5.1. Database Schema

#### 4.5.2. Distribution of System Components





## 5. Implementation

### 5.1. Preparation and Tools

#### 5.1.1. Development Environment

#### 5.1.2. Collaborative Development

`git`, `gitflow` and Github

#### 5.1.3. Deployment

Deployment Tool `Capistrano`

### 5.2. Server Component

#### 5.2.1. Core Classes and their Interaction

`Contact`

`Template`

`Email`

`Campaign`

`Conversation`

---

12

`Key & Value`

`Search & KeyBinding`

## **6. Evaluation**

### **6.1. Comparison with Initial Goals**

### **6.2. Observed Use Cases**

#### **6.2.1. Recruiting Exchange Students for one of the Most Desired Universities of the World**

#### **6.2.2. Requesting paper Reviews for a Journal**

#### **6.2.3. Managing Incoming Class Administration Emails**



## **7. Conclusion**

### **7.1. Conclusion of this work**

### **7.2. Discussion of results**

### **7.3. Future Work**



# Bibliography

- [1] Leslie Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.





## List of Figures



# Appendix



## **A. Detailed Descriptions**

Here come the details that are not supposed to be in the regular text.



## B. Colophon

This thesis is set in  $\text{\LaTeX}[1]$ , build by the continuous integration server Travis.