
	<div> HTBLuVA Wien XX  </div> <div> Abteilung: Informationstechnologie  Ausbildungsschwerpunkt: Systemtechnik </div>
---	--

## DIPLOMARBEIT

### Soil Moisture Exploitation Platform (SMEP)

Ausgeführt im Schuljahr 2013/14 von:

Betreuer/Betreuerin:

Ahmed Aly            5AHITT

Michael Borko, Bakk. techn.

Alexander Garber    5AHITT

Dr. Walter Rafeiner-Magor

Michaela Lipovits    5AHITT

.

Patrick Mühl        5AHITT

Wien, am 8.5.2014

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit — einschließlich Tabellen, Karten und Abbildungen —, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Vienna, 8.5.2014

---

Michaela Lipovits  
(Product Owner)

---

Ahmed Aly

---

Alexander Garber

---

Patrick Mühl

**Abstract**

The department for photogrammetry and remote sensing at the Technical University of Vienna works a lot with soil moisture data measured by satellites. These measurements are currently saved in binary files, alongside to additional data for the countries in a relational database as well as longitude and latitude values in plain-text files. The large amount of data has to be on each machine, which costs the user a lot of storage.

In order to save the data in a structured way and provide a modular expandable application programming interface which accesses the data remotely from a server, the project Soil Moisture Exploitation Platform (SMEP) was started.

**Kurzfassung**

Die Abteilung für Photogrammetrie und Fernerkundung der Technischen Universität Wien arbeitet sehr viel mit Bodenfeuchtedaten, welche von Satelliten gemessen werden. Diese Messungen werden zur Zeit in binären Dateien gespeichert. Parallel dazu werden die dazugehörigen Ländernamen in einer relationalen Datenbank gespeichert und die Daten für die jeweiligen Längen- und Breitengrade werden in einfachen Textdateien gespeichert. Diese große Menge an Daten muss auf jeder Maschine lokal vorhanden sein. Dies kostet viel Speicherplatz.

Um die Daten strukturiert zu speichern und eine modulare und erweiterbare Programmierschnittstelle, welche die Daten von einem Server abrufen, zu erstellen, wurde das Projekt Soil Moisture Exploitation Platform (SMEP) gestartet.



# Contents

<b>Abstract/Kurzfassung</b>	<b>3</b>
<b>1 Introduction</b>	<b>9</b>
<b>2 Problem description</b>	<b>11</b>
2.1 Current situation . . . . .	11
2.1.1 Timeseries . . . . .	13
2.1.2 Images . . . . .	14
2.2 Requirements . . . . .	16
2.3 Problem solving approach . . . . .	16
<b>3 Project organization</b>	<b>17</b>
3.1 Team . . . . .	17
3.2 Scrum . . . . .	18
3.2.1 Roles . . . . .	18
3.2.2 Product Backlog . . . . .	18
3.2.3 Sprints . . . . .	19
3.2.4 Meetings . . . . .	19
3.2.5 Experience . . . . .	20
<b>I Technologies</b>	<b>21</b>
<b>4 Git</b>	<b>23</b>
4.1 Clone . . . . .	23
4.2 Common Commands . . . . .	24
<b>5 Python</b>	<b>27</b>
5.1 Why is Python used at the Institute . . . . .	27
5.2 General . . . . .	27
5.2.1 Syntax . . . . .	28
5.2.2 Data Structures . . . . .	33
5.2.3 Data Types . . . . .	35
5.3 Libraries . . . . .	37
5.3.1 pip . . . . .	37
5.3.2 NumPy . . . . .	38
5.3.3 nosetests . . . . .	40

<b>6</b>	<b>Relational database</b>	<b>43</b>
6.1	Filling Database on the Server . . . . .	43
6.1.1	Datamodel . . . . .	43
6.2	Methods to write into the Database . . . . .	44
6.2.1	psycopg2 . . . . .	44
6.2.2	SQLAlchemy . . . . .	45
6.2.3	Performance Test . . . . .	46
<b>7</b>	<b>Data Warehouse</b>	<b>47</b>
7.1	Introduction . . . . .	47
7.2	Schema models . . . . .	48
7.3	Aggregation . . . . .	51
7.4	Advantages and Disadvantages . . . . .	51
7.5	Point descriptions . . . . .	52
7.5.1	Conclusion . . . . .	54
<b>8</b>	<b>NoSQL</b>	<b>55</b>
8.1	SciDB . . . . .	55
8.1.1	General . . . . .	55
8.1.2	Usage . . . . .	58
8.1.3	Advantages and Disadvantages . . . . .	61
8.2	Couchbase . . . . .	65
8.2.1	General . . . . .	65
8.2.2	Usage . . . . .	68
8.2.3	Advantages and Disadvantages . . . . .	73
8.2.4	Conclusion . . . . .	74
<b>9</b>	<b>Code Documentation Tools</b>	<b>75</b>
9.1	Doxygen documentation tool . . . . .	75
9.1.1	Usage of Doxygen . . . . .	75
9.1.2	Pro . . . . .	75
9.1.3	Contra . . . . .	76
9.2	Sphinx documentation tool . . . . .	76
9.3	Usage of Sphinx . . . . .	76
9.3.1	reST field Lists . . . . .	77
9.3.2	Headers . . . . .	78
9.3.3	Making a list . . . . .	79
9.3.4	Making a table . . . . .	80

9.3.5	Code explanation . . . . .	80
9.4	Other ways to edit the documentation . . . . .	80
9.4.1	Pro . . . . .	80
9.4.2	Contra . . . . .	80
9.5	Conclusion . . . . .	81
<b>II</b>	<b>Implementation</b>	<b>83</b>
<b>10</b>	<b>Server</b>	<b>85</b>
10.1	Initial setup . . . . .	85
10.1.1	OS Setup . . . . .	85
10.1.2	IPTables . . . . .	87
10.1.3	SSH . . . . .	90
10.1.4	Fail2Ban . . . . .	91
10.1.5	FTP . . . . .	93
10.1.6	PostgreSQL . . . . .	94
10.2	Webaccess . . . . .	96
10.3	Gunicorn . . . . .	96
10.3.1	Needed dependencies . . . . .	96
10.3.2	WSGI Web Server Gateway Interface (WSGI) . . . . .	97
10.3.3	Error & Access logs . . . . .	99
10.3.4	Timeouts . . . . .	100
10.4	Nginx . . . . .	100
10.4.1	Configuration . . . . .	100
<b>11</b>	<b>Datastore</b>	<b>103</b>
11.1	Reading the provided data . . . . .	103
11.1.1	Reading the .dat and .idx files . . . . .	103
11.1.2	Reading lon/lat . . . . .	104
11.1.3	Reading the country information . . . . .	105
11.2	Datastructure . . . . .	108
11.3	Setup . . . . .	109
11.4	Populating Couchbase . . . . .	109
11.4.1	Writing the data . . . . .	109
11.4.2	Deleting the data . . . . .	112
11.5	Reading data from Couchbase . . . . .	113
11.5.1	Views for Timeseries and Image . . . . .	113
11.5.2	Views for gathering GPIs . . . . .	114

11.5.3 Python Code for reading Views . . . . .	116
<b>12 API</b>	<b>117</b>
12.1 Design . . . . .	117
12.1.1 Datatier API . . . . .	117
12.1.2 Core API . . . . .	118
12.1.3 Client API . . . . .	119
12.2 Datatier API . . . . .	121
12.2.1 Datatier . . . . .	121
12.2.2 Datatier Exceptions . . . . .	121
12.2.3 Couchbase Datatier . . . . .	123
12.3 Core API . . . . .	137
12.3.1 Core Configuration . . . . .	137
12.3.2 Core Modules . . . . .	139
12.4 Client API . . . . .	145
12.4.1 urllib2 vs. requests . . . . .	145
12.4.2 Client API Implementation . . . . .	146
12.4.3 Functions . . . . .	148
<b>13 Conclusion</b>	<b>153</b>
13.1 Achieved Requirements . . . . .	153
13.2 Review . . . . .	154
13.3 Future work . . . . .	154
<b>Table of Figures</b>	<b>155</b>
<b>Table of Listings</b>	<b>157</b>
<b>References</b>	<b>161</b>
<b>Glossary</b>	<b>167</b>