Early Warning System Installation Guide

Deliverable 1.3.4 – SMART MOVE

Version 1.1 - 007.03.2019

Produced in the Framework of the
SMART-MOVE Project

Management of Highly Variable Water Resources in semi-arid Regions - Israel (ISR), Jordanien (JOR), Palästinensische Gebiete (PSE); Teilprojekt 5: Geodateninfrastruktur zum nachhaltigen Datenmanagement

Produced by
Disy Informationssysteme GmbH

(Förderkennzeichen: 02WM1355E)

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

FONA
Forschung für Nachhaltige Entwicklung
BMBF

BETREUT VOM

PTKA
Projektträger Karlsruhe
im Karlsruher Institut für Technologie

disy

Legal Notice

Author:	Houssam Haddad (Disy Informationssysteme GmbH)
Co-Authors:	Vanessa Rojas
Project Lead:	Dr. David Riepl
Project:	SMART-Move
Filename:	D134 SMART Move – Early Warning System Installation Guide
Status:	Final

Disy Informationssysteme GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

Change Log

Version	Date	Change description	Author
0.5	02/2018	Initial creation of the Document	Houssam Haddad
0.9	02/2018	Editorial Review and Compilation	Dr. David Riepl
1.0	03/2018	Finalization	Dr. David Riepl
1.1	03/2019	Adding Disclaimer	Dr. David Riepl

Table of Contents

List of Figures

List of Tables

0 Disclaimer

This report represents the project deliverable **D1.3.4** of the SMART-Move Project and should be understood in the context of the full activity of the Workpackage **1.3: Data Management.**

Since the relevant scope, and thus the necessary tasks, changed in parts during the project phase the following table gives an overview of the activities and results this Workpackage:

Tab. 0: Planned and conducted activities and results in Workpackage 1.3: Data Management.

Activity	Planned	Conducted	Results
1.3.1	Requirement Analyses of Data Management and Data Model	Requirement Analyses of Data Management, Integration of Data	D1.3.1: **This report:** Requirement Analyses of Data Management, Integration of Data
1.3.2	Design and implementation of Data Import Interface for large data sets and telemetric sensors of climate and water quality stations	Design and development of ETL-processes to retrieve and load telemetric Sensor data of SEBA and PESSL Stations into a centralized Data Repository. Design and implementation of business logic of the Spring water early warning system	D1.3.2: **Report:** Early Warning System Concept and Technical Description **Software:** EWS-Prototype with focus on backend functionality (ETL and Database)
1.3.3	Design and Realization of an online dataportal and geodata warehouse for measurement data search and export (WEAP). Documentation.	Realization of an online portal with integration of data layers from previous projects. Full realization of Production-Ready EWS with Technical adaption iterations during 1-year testing phase of the EWS at Wadi Shueib springs.	D1.3.3: **Software:** Full realization of Production-Ready EWS Online information portal for project partners.
1.3.4	Capacity Development for transboundary Data Management in the region; Documentation; Training Material;	Adaption of the EWS for Cluster West springs (Auja) Official Inauguration of EWS in Wadi Shueib Jordan EWS Technical Training Workshop with Employees of the WAJ in Jordan	D1.3.4: **This Report:** Early Warning System Installation Guide

1 Preface

This document describes how to install and configure the system components of the Early Warning System (EWS) for the spring contamination. If the system infrastructure differs, deviating steps may be required.

The EWS was developed for the SMART-MOVE Project and is inspired by an estimate of contamination risk in karstic spring water sources. The estimate is based on an empirical correlation of continuously & automatically measurable observation parameters. The goals of the EWS are:

- Fast & reliable integration of monitoring data from variable sources
- Live calculation of contamination risk from input measurement data
- Push information on contamination risk to relevant decision makers
- Provide stakeholders with easy access to all relevant data for analyses
- Build system by mature & robust system components

2 System architecture

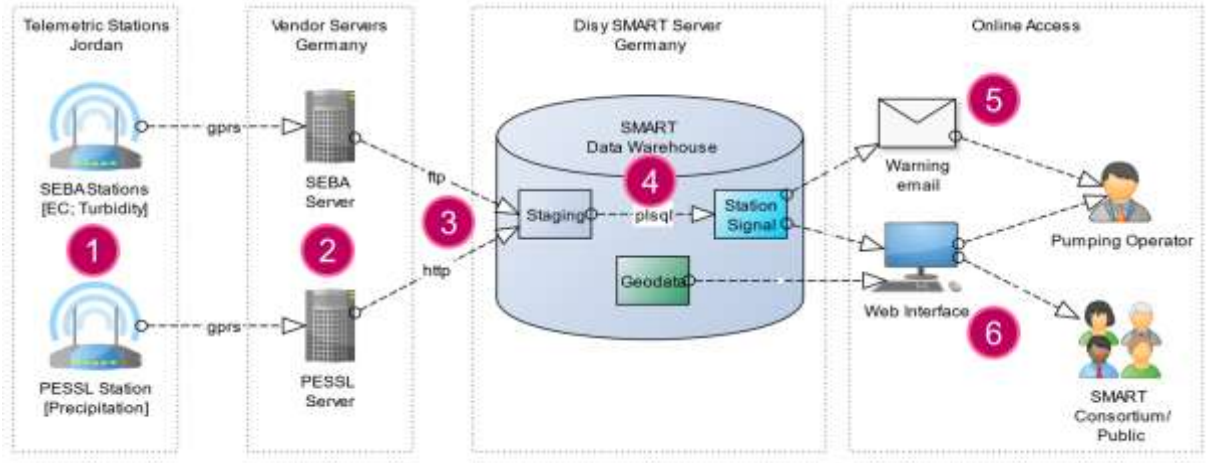


Figure 1 Functional components of the EWS

Figure 1 describes the functional components of the EWS:

1. Observation data is acquired at different stations (Rainfall & Hydro)
2. Stations send data to vendor-specific measurement Databases
3. Measurement data is integrated every 15min in SMART database (near real time)
4. New measurement data triggers updates of “Warning Levels” by EWS-algorithms
5. Changes of “Warning Level” trigger email-notifications to defined recipients
6. All available information can be accessed via a web interface

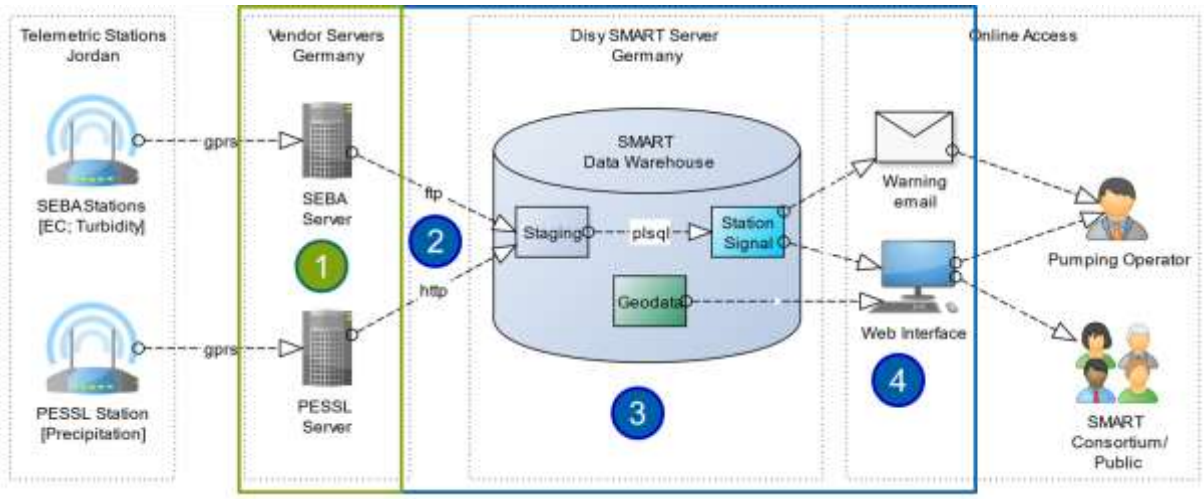


Figure 2 Technical components of the EWS

Figure 2 describes the technical components of the EWS:

1. Vendor-specific measurement databases (out-of-the-box). Data is transmitted telemtrically. These databases are not within the scope of this document.
    a. SEBA measures physiochemical parameters of spring monitoring stations and provides data in 15min-intervals as csv-files on an FTP-Server.
    b. PESSL measures precipitations and provides data on an http-URL-interface in a JSON format.
2. Measurement data import interfaces (java based). The data is
3. SMART database & EWS-procedures (Oracle DB)
4. Web interface (Disy Cadenza)

3 Minimum hardware Requirements

Processors:	1 quad core @ 3 GHz
RAM:	16 GB
HDD:	200 GB SAS
OS:	Windows Server 2008 R1 SP2 or higher, 64bit Linux 64bit

4 Setting up the data structure – Configuring the database

4.1 Installing the database

The Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit with the character set “AL32UTF8” was used during the project life cycle. For future implementations, it is recommended to also use an Oracle database with an equal or a more recent version.

Table 1 NLS parameters for the Oracle DB used during the project life cycle

NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	.,
NLS_CHARACTERSET	AL32UTF8
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD-MON-RR
NLS_DATE_LANGUAGE	AMERICAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY	$
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CONV_EXCP	FALSE
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_RDBMS_VERSION	11.2.0.1.0
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	.,
NLS_CHARACTERSET	AL32UTF8
NLS_RDBMS_VERSION	GREGORIAN

4.1.1 Install the database

Use the following paramers when creating installing the Oracle database:

1. Support email: --
2. Database-Only
3. Server-Class
4. Databaseinstallation with only one isntance
5. Language: English
6. Enterprise Edition (Default Settings)
7. Oracle-Base: C:\oracle11
8. Homedirectory: C:\oracle11\product\1.2.0\dbhome_1

4.1.2 Create an Oracle-Database-Instance

Use the following parameters when creating the Oracle-Database-Instance:

1. Global DB-Name: Choose a name for your DB. Example: “ICSNEUDB11“
2. Database configuration type: Choose the option “Single instance”
3. SID: Choose a name for the database SID. Example: “ICSNEUDB“
4. Create the database as a container database: No
5. Save type: Filesystem

4.1.3 Finalizing the installation

After installing the database and creating the instance, we need to make sure that the instance is accessible using the SID or the service name. Make sure that the service and TNS-listener have started and that they contain the correct parameters:

Example: **tnsnames.ora**:

```
ICSNEUDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = icsneuintdb.disy.net)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = icsneuintdb.disy.net)
    )
  )
```

Example: **listener.ora**:

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = icsneuintdb.disy.net)(PORT =
1521))
    )
  )

ADR_BASE_LISTENER = C:\oracle
```

4.2 Configuring the database

After making sure that our database-instance is working correctly, we need to prepare it for the SMART-MOVE schema.

1. Create a new tablespace with the name “SMARTMOVE”. The recommended file size is 1GB and the recommended max size is 32GB. The following is an example of the tablespace used during the project life cycle:

```
CREATE SMALLFILE TABLESPACE SMARTMOVE DATAFILE
  '/opt/oracle/oracle/oradata/oracle10/smartmove.dbf' SIZE 1G
  AUTOEXTEND ON NEXT 10M MAXSIZE 32767M
  LOGGING ONLINE PERMANENT BLOCKSIZE 8192
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE DEFAULT NOCOMPRESS SEGMENT SPACE
MANAGEMENT AUTO;
   ALTER DATABASE DATAFILE
```

```
  '/opt/oracle/oracle/oradata/oracle10/smartmove.dbf';
```

2. Create a new DB user with the name “SMARTMOVE” and the default tablespace “SMARTMOVE”.

4.3 Initializing the database

Follow these steps to initialize the newly created database:

1. Import the provided dump file. The import will populate the tables with the initial data and create the necessary database procedures. To import a dump file into a database, you may use a wizard with a graphical interface, such as the one provided with SQL Developer or use the following commands :

```
impdp <user-system>/<password-system> directory=<folder-containing-
dumpfile> dumpfile=<dump-file> logfile=<file-name> job_name=<job-name>
```

Example:

```
impdp SMARTMOVE/PASSWORD directory=DATAPUMP_DIR dumpfile=SMART-MOVE.dmp
LOGFILE=20181017_SMARTMOVE_IMP.log job_name=20181017_SMARTMOVE_IMP
```

2. If the initial import of the dump file is successful the schema “SMARTMOVE” will be populated with several tables, views and procedures.

4.4 Setting up the Early Warning System

4.4.1 Adding or editing stations

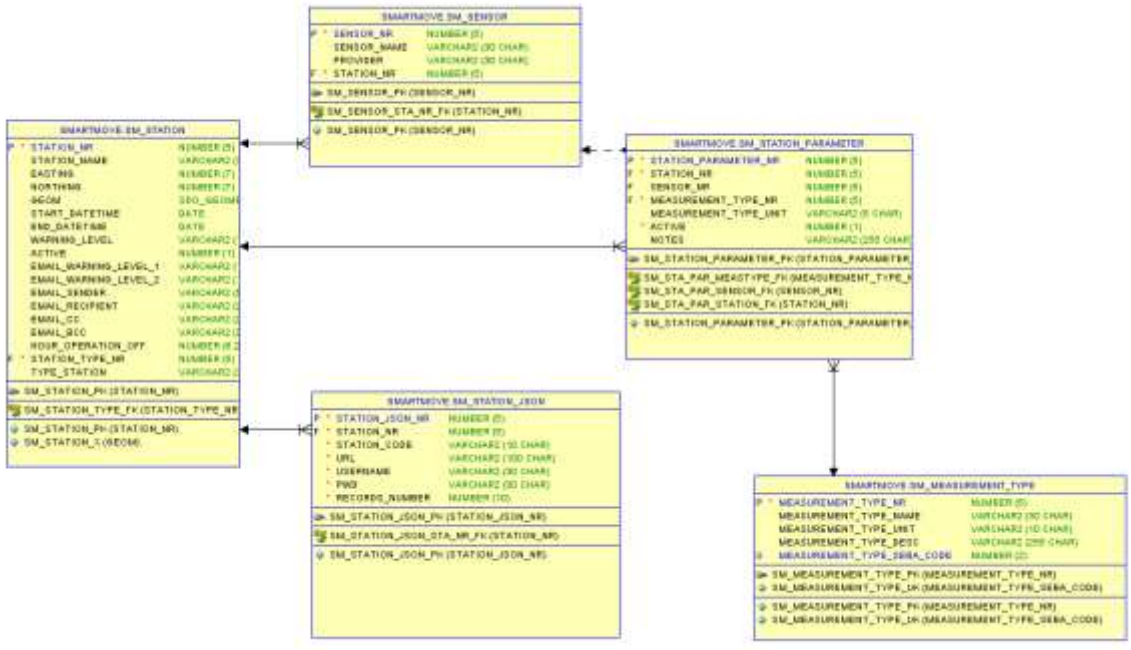


Figure 3 Diagramm Entity Relationship showing the main tables in the SMART_MOVE schema

4.4.1.1 Climate stations (Pessl Instruments)

Climate stations measure precipitation and deliver data as JSON files. To add or edit a climate station, you need to consider the following fields in the following tables. You may ignore fields written in **underlined bold**:

1. SM_STATION (Station information):
    a. **STATION_NR:** Automatically generated ID. Ignore this column.
    b. STATION_NAME: Set the print name of the station. Recommended: Add "(Climate Station)" as a suffix.
    c. EASTING/NORTHING: Set the coordinates for the location of the station if there is no geometry available.
    d. GEOM: Set the point geometry for the location of the station.
    e. **START_DATETIME/END_DATETIME:** These two columns are updated automatically by the procedures when there is a red signal. They indicate the start/end date time of the red warning signal. Ignore this field.
    f. WARNING_LEVEL: Set the default warning level. Possible values: 'GRAY', 'GREEN', 'YELLOW', 'RED'.
    g. ACTIVE: Set whether the station is active or inactive. Possible values: 0, 1.
    h. EMAIL_WARNING_LEVEL_1/EMAIL_WARNING_LEVEL_2: Set the warning levels for the automatic e-mail.
    i. EMAIL_SENDER/EMAIL_RECIPIENT/EMAIL_CC/EMAIL_BCC: Define the email addresses for the sender, the recipient, cc and bcc.
    j. HOUR_OPERATION_OFF: Set the amount of time the station will shut down after reaching a warning level. Unit is in hours. Required value: ≥1.
    k. STATION_TYPE_NR: Foreign key to the station type. Set the station type number to "1" (=Climate Station).

Example:

2. SM_STATION_PARAMETER (Parameter information):
    a. **STATION_PARAMETER_NR:** Automatically generated ID.
    b. STATION_NR: Foreign key to the station ID. Map the parameter to the corresponding station.
    c. **SENSOR_NR:** Foreign key to the sensor ID. Does not apply for climate stations. Leave this field as null.
    d. MEASUREMENT_TYPE_NR: Set this to "1" (=precipitation).
    e. ACTIVE: Set whether the parameter is active or not. Possible values: 0, 1.
    f. NOTES: You can add any additional notes here.
3. SM_STATION_JSON (Server information for data retrieval):
    a. **STATION_JSON_NR:** Automatically generated ID.
    b. STATION_NR: Foreign key to the station ID. Map the JSON-URL to the corresponding climate station.
    c. STATION_CODE: Provide the identifying code as set by the manufacturer. This will be appended to the end of the URL.
    d. URL: Set the main URL to fetch the JSON files.
    e. USERNAME/PWD: Provide the authentication details to logon to the JSON Server.

f. RECORDS_NUMBER: Choose the number of records to be retrieved. Unit is in hours. Required value: ≥1. For example, the recommended value “4” will fetch all data from the last 4 hours.

4.4.1.2 Spring Monitoring stations (SEBA)

Spring monitoring stations measure turbidity and electrical conductivity (EC). They deliver data as CSV files. To add or edit a climate station, you need to consider the following fields in the following tables. You may ignore fields written in **underlined bold**:

1. SM_STATION (Station information):
    a. **STATION_NR:** Automatically generated ID. Ignore this column.
    b. STATION_NAME: Set the print name for the station.
    c. EASTING/NORTHING: Set the coordinates for the location of the station if there is no geometry available.
    d. GEOM: Set the point geometry for the location of the station.
    e. **START_DATETIME/END_DATETIME:** These two columns are updated automatically by the procedures when there is a red signal. They indicate the start/end date time of the red warning signal. Ignore this field.
    f. WARNING_LEVEL: Set the default warning level. Possible values: ‘GRAY’, ‘GREEN’, ‘YELLOW’, ‘RED’.
    g. ACTIVE: Set whether the station is active or inactive. Possible values: 0, 1.
    h. EMAIL_WARNING_LEVEL_1/EMAIL_WARNING_LEVEL_2: Set the warning levels for the automatic e-mail.
    i. EMAIL_SENDER/EMAIL_RECIPIENT/EMAIL_CC/EMAIL_BCC: Define the email addresses for the sender, the recipient, cc and bcc.
    j. HOUR_OPERATION_OFF: Set the amount of time the station will shut down after reaching a warning level. Unit is in hours. Required value: ≥1.
    k. STATION_TYPE_NR: Foreign key to the station type. Set the station type number to “2” (=Spring Station).
2. SM_SENSOR (Sensor information for data retrieval) (A single spring monitoring station can have multiple sensors):
    a. **SENSOR_NR:** Automatically generated ID. Ignore this column.
    b. SENSOR_NAME: Set the name of the sensor. The name needs to have the prefix “UCL” (Example: UCL1022) because the data integration jobs are set to expect UCL* files.
    c. PROVIDER: Set the name of the sensor provider.
    d. STATION_NR: Foreign key to the station ID. Map the sensor to the corresponding spring monitoring station.
3. SM_STATION_PARAMETER (Parameter information):
    a. **STATION_PARAMETER_NR:** Automatically generated ID. Ignore this column.
    b. STATION_NR: Foreign key to the station ID. Map the parameter to the corresponding station.
    c. SENSOR_NR: Foreign key to the sensor ID. Map the parameter to the corresponding sensor.
    d. MEASUREMENT_TYPE_NR: Set the measurement type number to “2” (=turbidity) or to “1” (=EC).
    e. ACTIVE: Set whether the parameter is active or not. Possible values: 0, 1.
    f. NOTES: You can add any additional notes here.

4.4.2 Setting up EWS signals

4.4.2.1 Rain signals

The rain signals apply only to climate stations. To add or edit a rain signal, you need to consider the following fields in the SM_RAIN_SIGNAL. You may ignore fields written in **underlined bold**.

1. **RAIN_SIGNAL_NR:** Automatically generated ID. You can ignore this column.
2. RAIN_SIGNAL_NAME: Set the print name of the signal.
3. STATION_NR_RAIN: Foreign key to the station ID. Map the rain signal to the corresponding climate station.
4. MEASUREMENT_TYPE_NR_RAIN: Foreign key to the parameter ID. Set this to “1” to map the rain signal to the precipitation parameter.
5. RAIN_LIMIT_YELLOW: Set the threshold for the yellow warning level. Unit is in [mm].
6. RAIN_LIMIT_RED: Set the threshold for the red warning level. The value has to be greater than the yellow warning level. Unit is in [mm].
7. HOUR_RANGE_RAIN: Set the time window for the calculation of the warning signals. Unit is in hours.
8. ACTIVE: Set whether the parameter is active or not. Possible values: 0, 1.
9. NOTES: You can add any additional notes here.

Table 2 An example of a rain signal (not actual values)

Rain Signal NR RAIN_SIGNAL_ NR	Signal Name RAIN_SIGNAL_ NAME	Station Precipitation STTION_NR_ RAIN	Measurement Type MEASURENT_TYPE _NR_RAIN	Rain Yellow	Rain Red	Rain hour	Active
1	As-Salt	As-Salt (2)	Precipitation (1)	10	40	12	1
2	Fuhais	Fuhais (4)	Precipitation (1)	10	40	12	1

4.4.2.2 Combined signals

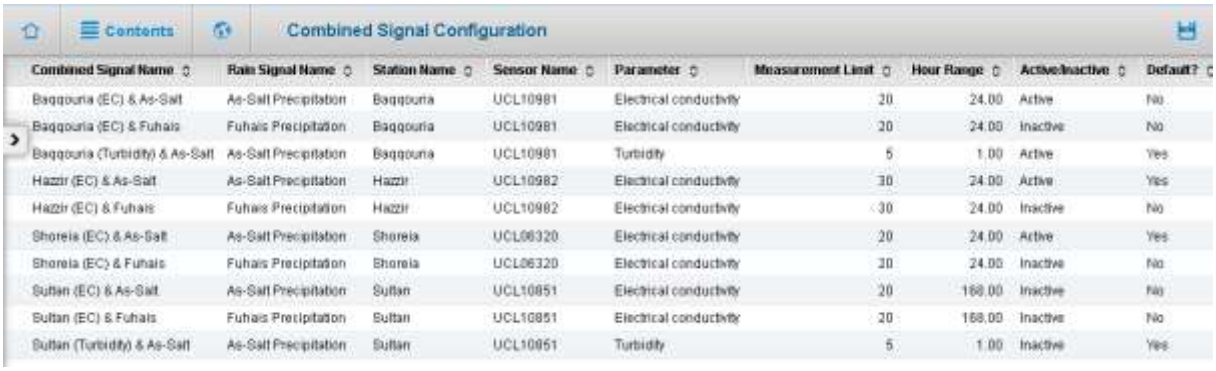
The combined signals combine a rain signal from a climate station with a turbidity or EC signal from a spring monitoring station. To add or edit a combined signal, you need to consider the following fields in the SM_COMBINED_SIGNAL. You may ignore fields written in **underlined bold**.

1. **COMBINED_SIGNAL_NR:** Automatically generated ID. You can ignore this column.
2. COMBINED_SIGNAL_NAME: Set the print name of the signal. This format is recommended: [Name of the spring monitoring station] ([Parameter]) & [Name of the climate station]. Example: “Baqquouria (Turbidity) & As-Salt”.
3. RAIN_SIGNAL_RAIN: Foreign key to the rain signal ID. Map the combined signal to a rain signal of choice.
4. STATION_NR: Foreign key to the station ID. Map the combined signal to a spring monitoring station.
5. SENSOR_NR: Foreign key to the sensor ID. Map the combined signal to a sensor.
6. MEASUREMENT_TYPE_NR: Foreign key to the parameter ID. Map the combined signal to a corresponding measurement type. The referenced parameter has to be either “Turbidity” or “EC” (Possible values: 2, 3)

7. MEASUREMENT_LIMIT: Set the Threshold for turbidity or EC to trigger a warning level. Unit in [FTU] for Turbidity and [µS] for EC. Required value: ≥0 (Example: 20)
8. HOUR_RANGE_RAIN: Set the time window for the calculation of the warning signals. Required value: ≥1 (Example: 24)
9. ACTIVE: Determines whether the parameter is active or not. Possible values: 0, 1.
10. DEFAULT_SIGNAL: Set the default signal. For every combination of two stations, only one combined signal is allowed. Enabling one will automatically disable the already enabled combined signals. Possible values: 0, 1.

Table 3 An example of a combined signal (not actual values)

Comb Signal NR	Signal Name	Rain Signal NR	Station NR Sensor NR	Signal	Meas Type	Limit	Hours	Active	Def
2	Baqqouria (Turbidity) As-Salt	As-Salt (1)	Baqquoria (1) UCL10981 (1)	Precipitation + Turbidity	Turbidity (2)	5	1	Y	Y
3	Hazzir (EC) & Fuhais	Fuhais (2)	Hazzir (6) UCL10982 (2)	Precipitation + EC	EC (3)	30	24	Y	Y
4	Sultan (Turbidity) As-Salt	As-Salt (1)	Sultan (7) UCL10851 (4)	Precipitation + Turbidity	Turbidity (2)	5	1	Y	Y
5	Shoreia (EC) & Fuhais	Fuhais (2)	Shoreia (3) UCL06320 (3)	Precipitation + EC	EC (3)	20	24	Y	Y
6	Baqqouria (EC) & Fuhais	Fuhais (2)	Baqquoria (1) UCL10981 (1)	Precipitation + EC	EC (3)	20	24	Y	N
7	Sultan (EC) & Fuhais	Fuhais (2)	Sultan (7) UCL10851 (4)	Precipitation + EC	EC (3)	20	=24*7 =168	Y	N



Combined Signal Name	Rain Signal Name	Station Name	Sensor Name	Parameter	Measurement Limit	Hour Range	Active/Inactive	Default?
Baqqouria (EC) & As-Salt	As-Salt Precipitation	Baqqouria	UCL10981	Electrical conductivity	20	24.00	Active	No
Baqqouria (EC) & Fuhais	Fuhais Precipitation	Baqqouria	UCL10981	Electrical conductivity	20	24.00	Inactive	No
Baqqouria (Turbidity) & As-Salt	As-Salt Precipitation	Baqqouria	UCL10981	Turbidity	5	1.00	Active	Yes
Hazzir (EC) & As-Salt	As-Salt Precipitation	Hazzir	UCL10982	Electrical conductivity	30	24.00	Active	Yes
Hazzir (EC) & Fuhais	Fuhais Precipitation	Hazzir	UCL10982	Electrical conductivity	30	24.00	Inactive	No
Shoreia (EC) & As-Salt	As-Salt Precipitation	Shoreia	UCL06320	Electrical conductivity	20	24.00	Active	Yes
Shoreia (EC) & Fuhais	Fuhais Precipitation	Shoreia	UCL06320	Electrical conductivity	20	24.00	Inactive	No
Sultan (EC) & As-Salt	As-Salt Precipitation	Sultan	UCL10851	Electrical conductivity	20	168.00	Inactive	No
Sultan (EC) & Fuhais	Fuhais Precipitation	Sultan	UCL10851	Electrical conductivity	20	168.00	Inactive	No
Sultan (Turbidity) & As-Salt	As-Salt Precipitation	Sultan	UCL10851	Turbidity	5	1.00	Inactive	Yes

Figure 4 An example of a combined signal configuration displayed on Cadenza online portal

4.4.3 Configuring SMTP

In this chapter we will configure the Oracle database to allow it to send warning e-mails through SMTP. We will go through the example used during the project life cycle.

4.4.3.1 SMTP server configuration and UTL_MAIL package creation:

1. Connect to the server and make executable the files below

```
$ chmod 755 $ORACLE_HOME/rdbms/admin/utlmail.sql
$ chmod 755 $ORACLE_HOME/rdbms/admin/prvtmail.plb
```

2. Connect as SYS and execute the scripts below

```
$ SQLPLUS CONN sys/password AS SYSDBA
SQL> @$ORACLE_HOME/rdbms/admin/utlmail.sql
SQL> @$ORACLE_HOME/rdbms/admin/prvtmail.plb

SQL> @ /opt/oracle/oracle/product/11.2.0/dbhome_1/rdbms/admin/utlmail.sql
SQL> @ /opt/oracle/oracle/product/11.2.0/dbhome_1/rdbms/admin/prvtmail.plb
```

3. In addition the SMTP_OUT_SERVER parameter must be set to identify the SMTP server. Set the parameter in the database startup file and restart the database.

```
SQL> ALTER SYSTEM SET smtp_out_server='smtp.disy.net' SCOPE=SPFILE;
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
```

4. Give grants to the new package to the respective users

```
SQL> GRANT EXECUTE ON SYS.UTL_MAIL TO SMARTMOVE;

-- Query to verify grants
SQL> SELECT GRANTEE , TABLE_NAME , PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME = 'UTL_MAIL';
```

4.4.3.2 Access Control List (ACL) configuration

We need to configure an Access Control List (ACL) and grant privileges to user SMARTMOVE.

1. Create ACL and grant privilege to SMARTMOVE (connect as SYS)
   Procedure DBMS_NETWORK_ACL_ADMIN.CREATE_ACL description:

Parameter	Description
acl	Name of the Access Control List. This is a XML file which will be created in /sys/acls directory by default.
description	Description of the ACL
principal	Principal (database user or role) to whom the privilege is granted or denied. Case sensitive.
is_grant	TRUE or FALSE, whether to grant access or deny access.
privilege	Network privilege to be granted or denied - 'connect/resolve' (case sensitive).Note: SMARTMOVE user needs both privileges.

   Execute the following script:

```
BEGIN
   DBMS_NETWORK_ACL_ADMIN.CREATE_ACL (
    acl          => 'smartmove.xml',
    description  => 'Permissions to access smtp.disy.net',
    principal    => 'SMARTMOVE',
    is_grant     => TRUE,
    privilege    => 'connect');
   COMMIT;
END;
/
```

2. Add privilege for user SMARTMOVE

```
BEGIN
   DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE (
    acl          => 'smartmove.xml',
    principal    => 'SMARTMOVE',
    is_grant     => TRUE,
    privilege    => 'resolve');
   COMMIT;
END;
/
```

3. Assign a network host to the Access Control List
   Procedure DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL description:

Parameter	Description
acl	Name of the Access Control List.
host	Host to which the ACL will be assigned. The host can be the name or the IP address of the host. A wildcard can be used to specify a domain or an IP subnet. The host or domain name is case-insensitive. Note: Only one ACL can be assigned to any host computer, domain, or IP subnet.
principal	Principal (database user or role) to whom the privilege is granted or denied. Case sensitive.
lower_port	Lower bound of a TCP port range if not NULL
upper_port	Upper bound of a TCP port range. If NULL, lower_port is assumed.

Execute the following script:

```
BEGIN
   DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL(
     acl         => 'smartmove.xml',
     host        => 'smtp.disy.net',
     lower_port  => 25);
END;
/
-- Select to verify privileges
SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT FROM DBA_NETWORK_ACL_PRIVILEGES;

-- Query to list ACL and host
SELECT ACL, HOST, LOWER_PORT, UPPER_PORT FROM DBA_NETWORK_ACLS;

-- Query to list domains for host
SELECT * FROM TABLE(DBMS_NETWORK_ACL_UTILITY.DOMAINS('smtp.disy.net'));
```

4.4.3.3 Execute email procedure

The procedure to send an email has the following format:

```
BEGIN

        UTL_MAIL.send(sender     => 'smartmove@disy.net',
                recipients => 'vanessa.rojas@disy.net,person2@domain.com',
                cc         => 'person3@domain.com',
                bcc        => 'person4@domain.com',
                subject    => 'UTL_MAIL SMARTMOVE Test',
                message    => 'This is a test for the warning signal'); */
END;
/
```

4.4.3.4 Further information

https://oracle-base.com/articles/10g/plsql-enhancements-10g#UTL_MAIL
http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_networkacl_adm.htm#CHDJFJFF

5 Fetching the live data – Configuring the automatic data integration

Data provided by the measuring stations can be automatically downloaded and integrated into the database for further analysis. “Talend Open Studio for Data Integration”, an ETL tool (Extract, Transform and Load) is deployed to achieve this goal.

5.1 Configuring Talend jobs

The Talend jobs are delivered alongside this document. Place the folder “talendjob” in a directory of choice on the production server. The folder contains the following:

1. java: Java Runtime Environment required for the jobs.
2. logs: Contains logfiles.
3. JSON_00_Master_0.1 (Job JSON Master): Contains the job and its necessary parameters for climate stations.
4. SEBA_00_Master_01 (Job SEBA Master): Contains the job and its necessary parameters for spring monitoring stations.

Before running the jobs we have to adjust some context variables:

5.1.1 Climate stations:

The data for climate stations are fetched by running the job “Job JSON Master”. The server URL and login parameters are stored in the table SM_STATION_JSON and are easily configurable. We still need to adjust some context parameters before running the jobs.

1. Navigate to:

   `talendjob/JSON_00_Master_0.1/JSON_00_Master/smartmove/json_01_read_rainsignal_0_1/contexts`

   and edit the database connection in the file Default.properties.

2. Navigate to:

   `talendjob/JSON_00_Master_0.1/JSON_00_Master/smartmove/json_02_read_values_0_1/contexts`

   and edit the database connection in the file Default.properties. Leave StationNr and JSONurl as is.

5.1.2 Spring monitoring stations:

The data for climate stations are fetched by running the job “Job SEBA Master”. The server URL and login parameters are hard coded in the Talend job and are configurable. The job is designed to run with SEBA servers (data.seba-hydrocenter.de). We still need to adjust some context parameters before running the jobs. Navigate to:

1. `talendjob/SEBA_00_Master_0.1/SEBA_00_Master/sm/seba_00_master_0_1/contexts`
2. `talendjob/SEBA_00_Master_0.1/SEBA_00_Master/sm/seba_01_ftp_0_1/contexts`

3. `talendjob/SEBA_00_Master_0.1/SEBA_00_Master/sm/seba_02_stg_job_0_1/contexts`
4. `talendjob/SEBA_00_Master_0.1/SEBA_00_Master/sm/seba_03_imp_seba_0_1/contexts`

and edit the following in the file Default.properties:

The database connection:

DataFTPPath: points to the directory where the FTP files will be downloaded to. Change it to a directory existing on the server.

DataPath: You may ignore this parameter.

JobNr: Leave as is.

5.2 Running the jobs

1. Make sure to check the java path within the executables.
2. Linux only: Make the *.sh files executables (chmod 755)
3. Schedule to run the following jobs as follows:

Jobs	Station	Parameter	Time Zone	Server	Time	Detail
Job JSON Master PESSL Instruments	As-Salt Station_nr=2 Fuhais Station_nr=4	Precipitation	UTC + 2	Production Server	XX:15 Once per hour	Talend Job
Job SEBA Master	Sensors y Measurements configured in SM_SENSOR, SM_STATION_PARAMETER	EC,Turbidity, Water Level	UTC + 2	Production Server	XX:30 Once per hour	Talend Job
Warning Signal	Combined Signal	Precipitation	UTC + 2	DB Server	XX:35 Once per hour	SCENARIO_PKG.rain_run_signal
Warning Signal	Combined Signal	Turbidity + Precipitation	UTC + 2	DB Server	XX:45 Once per hour	SCENARIO_PKG.turbidity_run_signal
Warning Signal	Combined Signal	EC + Precipitation	UTC + 2	DB Server	XX:55 Once per hour	SCENARIO_PKG.ec_run_signal

```
15 * * * * /home/disy/talendjob/JSON_00_Master_0.1/JSON_00_Master/JSON_00_Master_run.sh
30 * * * * /home/disy/talendjob/SEBA_00_Master_0.1/SEBA_00_Master/SEBA_00_Master_run.sh
```

Figure 5 Talend jobs running as cron jobs on Linux

```
35 * * * * /home/oracle/smartmove/runsm_rain.sh >> /home/oracle/smartmove/runsm_rain.log 2>&1
45 * * * * /home/oracle/smartmove/runsm_turbidity.sh >> /home/oracle/smartmove/runsm_turbidity.log 2>&1
55 * * * * /home/oracle/smartmove/runsm_ec.sh >> /home/oracle/smartmove/runsm_ec.log 2>&1
```

Figure 6 Oracle DB jobs running as cron jobs on Linux

6 Analyzing the data – Configuring Cadenza

Cadenza is a software platform developed by Disy Informationssysteme GmbH. It is the recommended tool to filter, analyze and visualize data for the SMART-MOVE project. Cadenza also provides a web interface to view the data remotely.

6.1 Installing Cadenza

Follow these steps to install the cadenza platform:

1. Depending on your operating system, extract the provided zipped file containing the installation files for Linux or Windows.
2. Depending on your operating system run Setup.exe or Setup.sh and follow the on-screen instructions.
3. Choose “Englisch” as the installation language
4. Read the installer welcoming text and click on next.
5. Read and accept the license agreement.
6. Browse for the destination directory and click on next.
7. Choose “custom installation” and click on next.
8. Choose the following options: Cadenza Desktop (mandatory), Cadenza Web, Access manager, Demodata (optional), User Documentation (mandatory), Uninstaller.
9. Check the box “Use and install provided JRE” and click on next.
10. Type in the hostname of the server and click on next.
11. Type in the ports on which Cadenza will run, the web alias and the email address for error reports and click on next.
12. Windows only: Choose if you would like to add Cadenza to the start menu and click on next.
13. Choose the name of the Cadenza Webservice and click on next.
14. Windows only: Set the min and max size of the JVM heap in MB. You can leave the default as it is as well. Clicking on next will start the installation. A summary of your installation parameters will be provided after the installation is successfully completed.

6.2 Configuring Cadenza

6.2.1 SMART-MOVE repository

After installing Cadenza, we need to configure it to work with SMART-MOVE data. Follow these steps to connect cadenza to the SMART-MOVE repository:

1. Extract the zipped file “repository_smartmove.zip” provided with this document.
2. Place the folder repository_smartmove under [Cadenza_Installation_Directory]/CadenzaDesktop/repository
3. Navigate to [Cadenza_Installation_Directory]/CadenzaDesktop/config and edit the file repositoryList.xml as follows:

```
<?xml version="1.0" encoding="UTF-8"?>

<repositoryList xmlns="http://www.disy.net/cadenza/repositoryList/schema">
  <mainRepository printName="Main">

<systemUrl>$VAR{REPOSITORY_BASE_SYSTEM_URL}/repository_smartmove</systemUrl
>

<publicUrl>$VAR{REPOSITORY_BASE_PUBLIC_URL}/repository_smartmove</publicUrl
>
  </mainRepository>
</repositoryList>
```

6.2.2 Installing cadenza as a service

You may want to install Cadenza as a Windows service or a SystemD service, so the online portal would stay online even when the server is restarted. Follow these steps to install Cadenza as a service:

6.2.2.1 Linux

1. Navigate to [Cadenza_Installation_Directory]/CadenzaDesktop/bin/services and edit the file install_systemd_services.sh.
2. Remove the following lines:
    a. uninstall_service cadenzawebstart.service
    b. install_service $CADENZABASE/CadenzaDesktop/bin/services/cadenzawebstart.service
3. Run the install_systemd_services.sh
4. Use systemctl to control the state of the Cadenza service

6.2.2.2 Windows

1. Navigate to [Cadenza_Installation_Directory]/CadenzaWeb/bin/ and run with administrative privileges the file “installCadenzaWebNTService.bat” then follow the in screen instructions.
2. Navigate to [Cadenza_Installation_Directory]/CadenzaDesktop/bin/services and run with administrative privileges the file “install_accessmanager_service.bat”
3. Use Windows services to control the state of the Cadenza service

6.3 Cadenza user management

Cadenza user management allows an administrator to add, edit or delete users that access the Cadenza platform. The user management is only available in Cadenza Desktop. Follow these steps to manage users:

1. Enable Cadenza user management for Cadenza Web by navigating to [Cadenza_Installation_Directory]/CadenzaWeb/webapps/cadenza/WEB-INF/config and editing the file plugins.xml and uncommenting the following lines: <plugin name="AccessManager"></plugin>
2. Enable Cadenza user management for Cadenza Desktop by navigating to [Cadenza_Installation_Directory]/CadenzaDesktop/config and editing the file plugins.xml and uncommenting the following lines: <plugin name="AccessManager"></plugin>
3. Make sure Cadenza user management service is started
4. Start Cadenza Desktop and login as an Admin (Default: Admin/Admin)
5. Click on the person icon to start cadenza user management

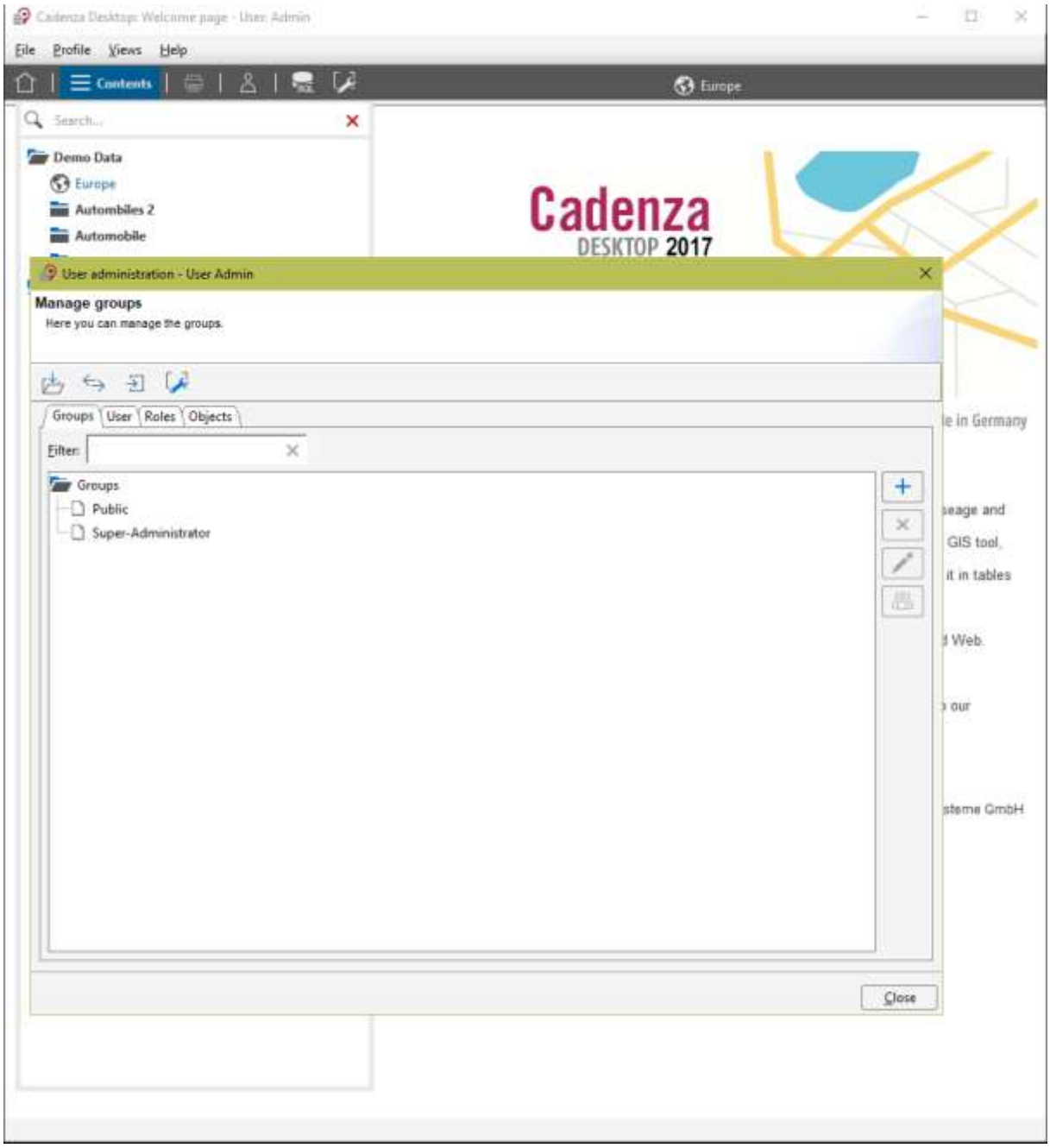


Figure 7 Overview of the Cadenza user management.

6.4 Using Cadenza

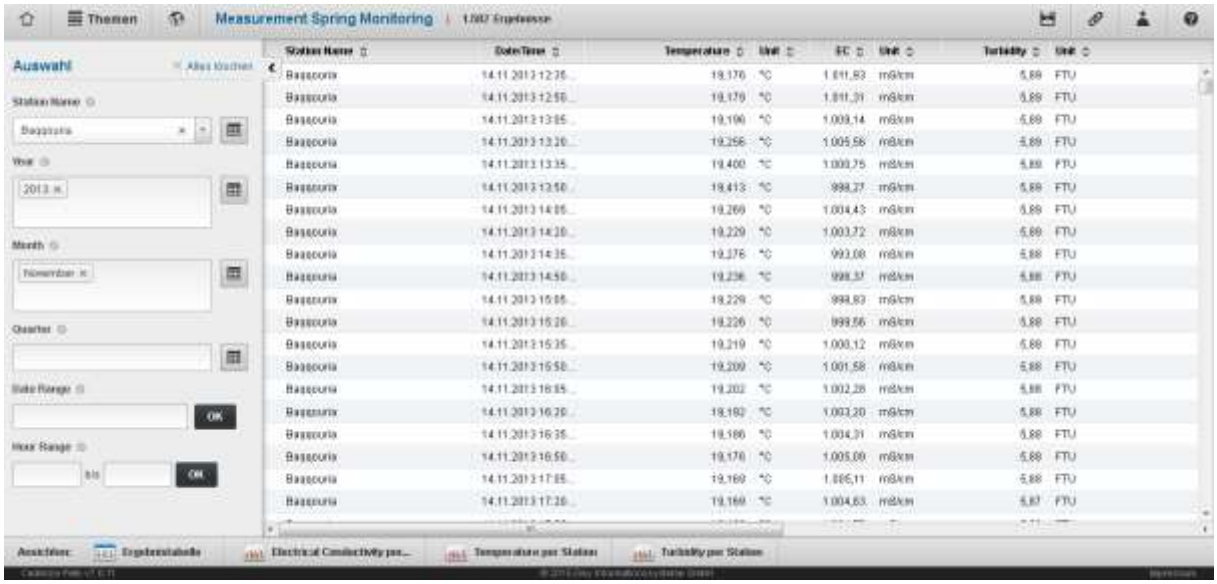
The station and measurement information could be accessed via an online portal. The online portal provides a list of different tables where the information can be filtered. It also allows the possibility to display diagrams of the filtered data or compare measurements between stations.

Figure 8 Overview of the Cadenza online portal

6.4.1 Tables and Diagrams

6.4.1.1 Table: Measurement Spring Monitoring

This table shows the measurements per spring monitoring station. On the left side is possible to filter the data. At the bottom is possible to select different diagrams per station.

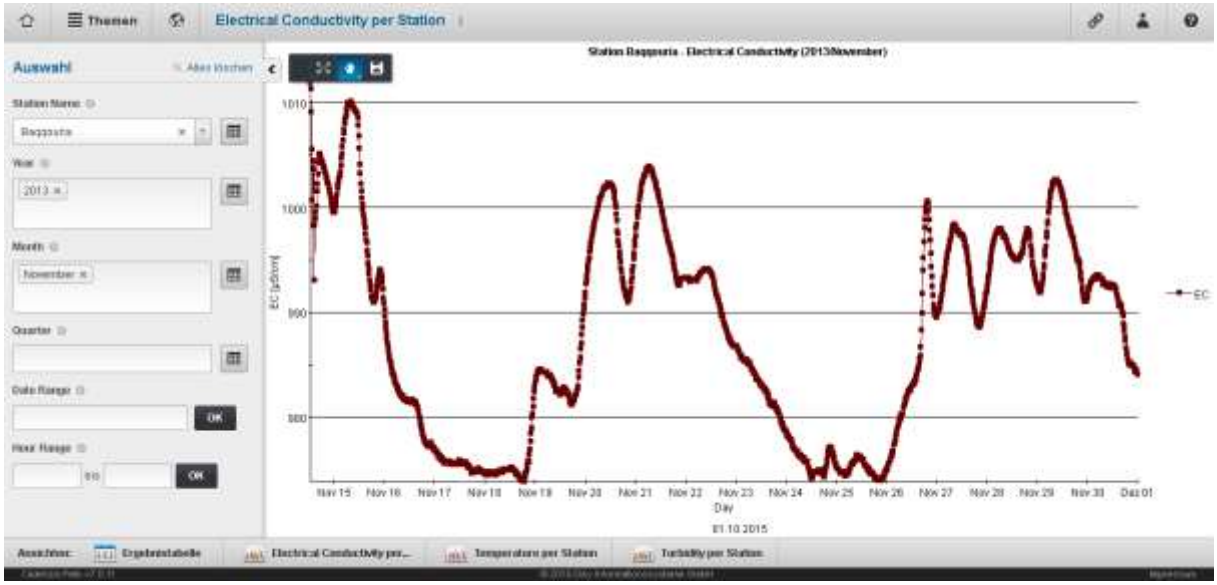


Filter	Description
Station Name	Station name for the measurements and diagrams.
Year	The available years will be listed in the combo box.
Month	The available months will be listed in the combo box.
Quarter	The filter gives the possibility to choose a quarter.

Date Range	Filter the data according to a date range.
Hour Range	Filter the data according to an hour range.

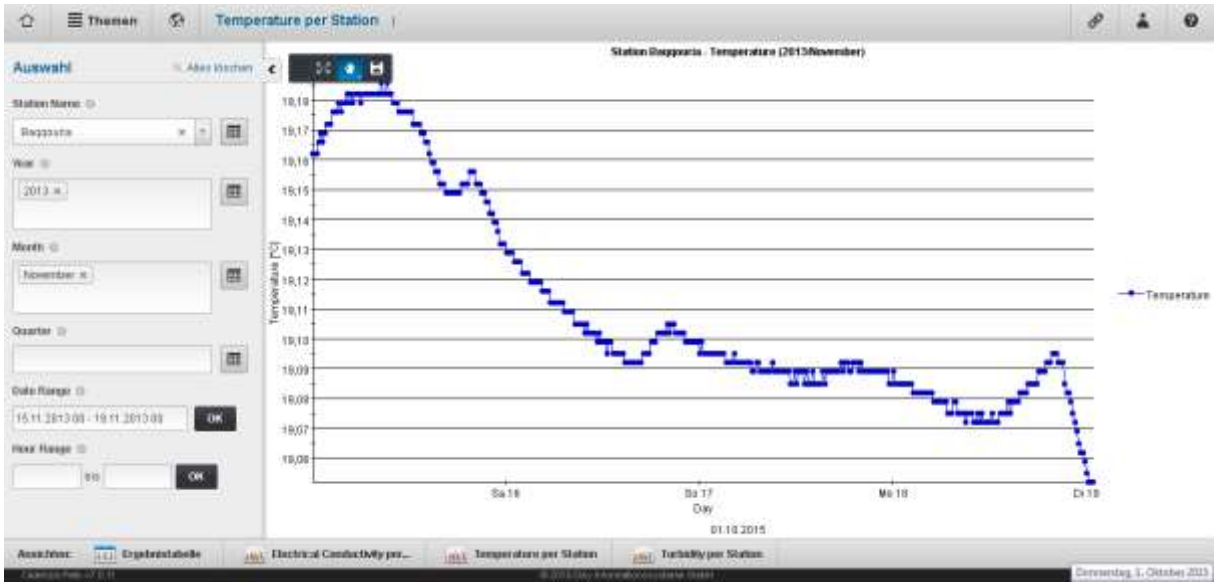
6.4.1.2 Diagram: Electrical Conductivity per Station

This diagram shows the variation of the electrical conductivity in the given range.



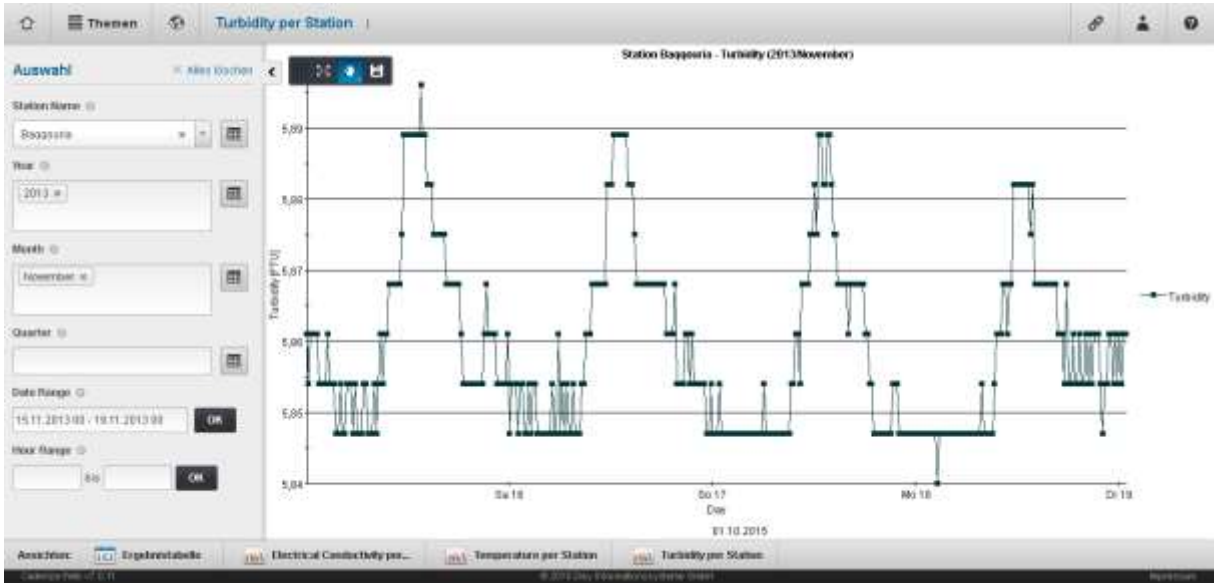
6.4.1.3 Diagram: Temperature per Station

This diagram shows the variation of the temperature pro station according to the filter.



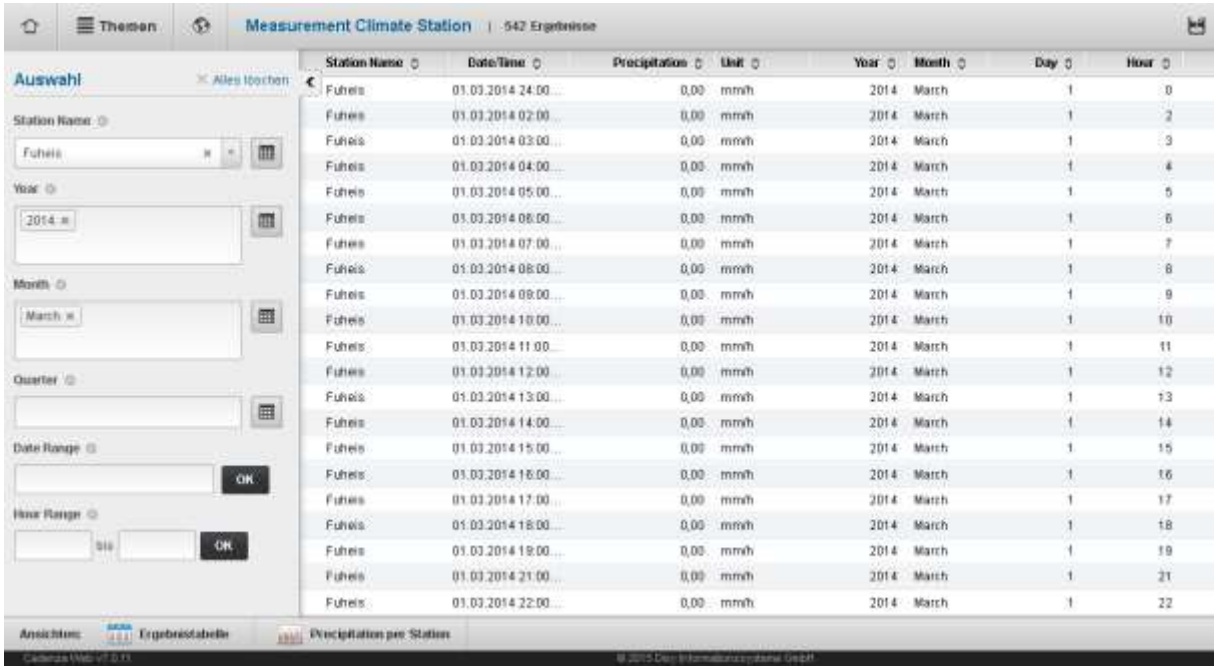
6.4.1.4 Diagram: Turbidity per Station

This diagram shows the turbidity variation according to the given filter.



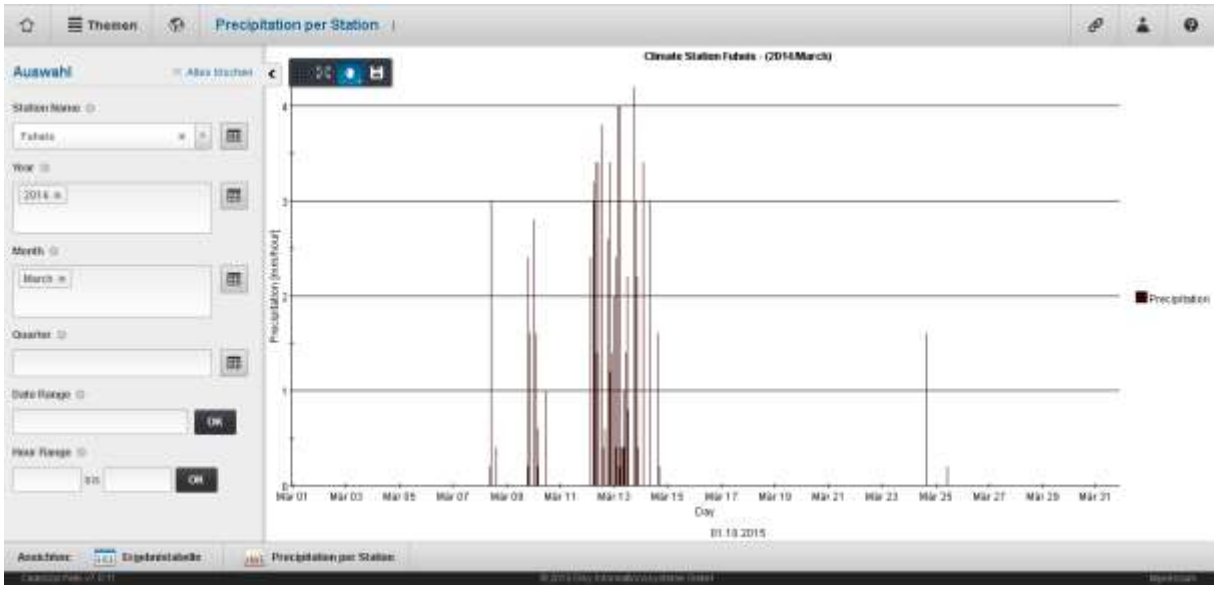
6.4.1.5 Table: Measurement Climate Station

This table shows the individual measurements for each climate station. The related precipitation diagram can be displayed with the option Precipitation pro Station at the bottom.



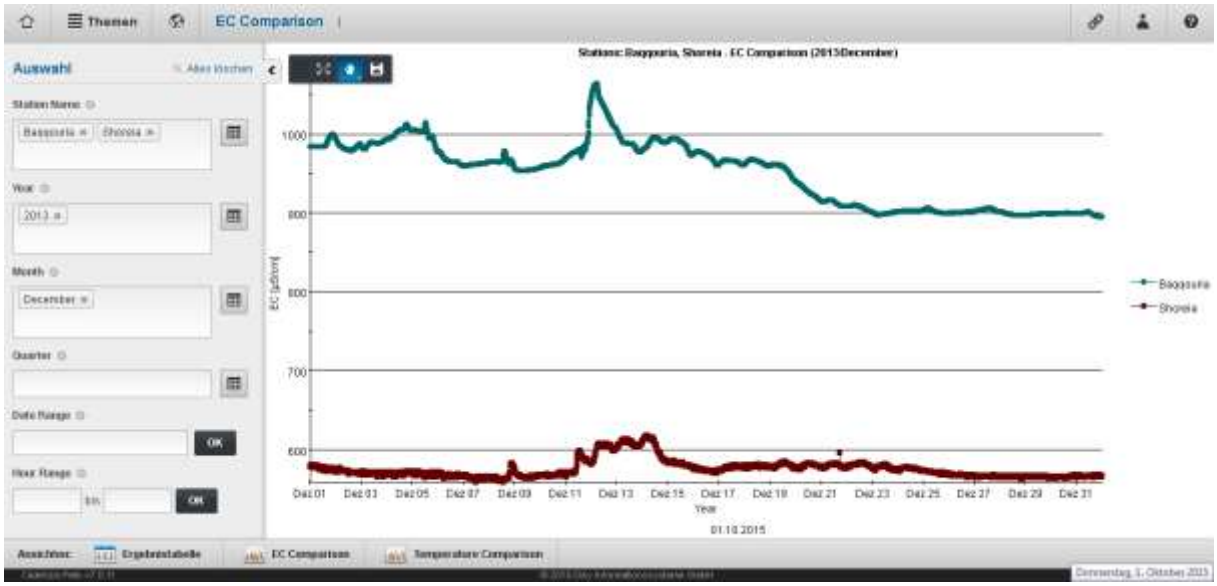
6.4.1.6 Diagram: Precipitation per Station

This diagram shows the precipitation for each station.



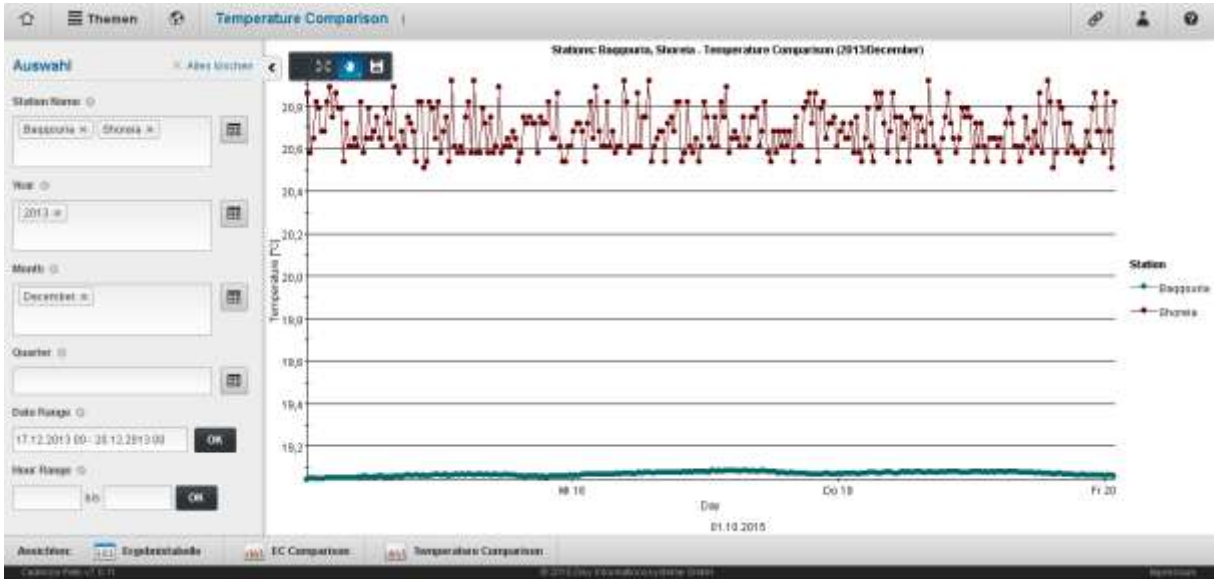
6.4.1.7 Diagram: EC Comparison

This diagram is based on the previous table and can be used to compare EC values between stations.



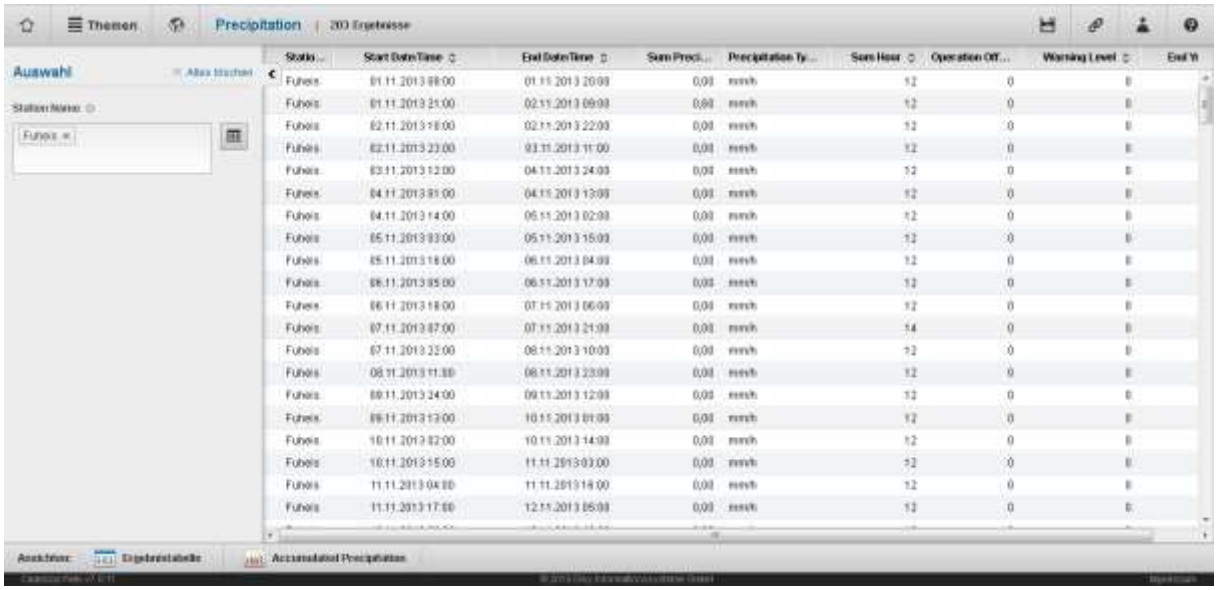
6.4.1.8 Diagram: Temperature Comparison

This diagram compares temperature measurements between stations.



6.4.1.9 Table: Precipitation

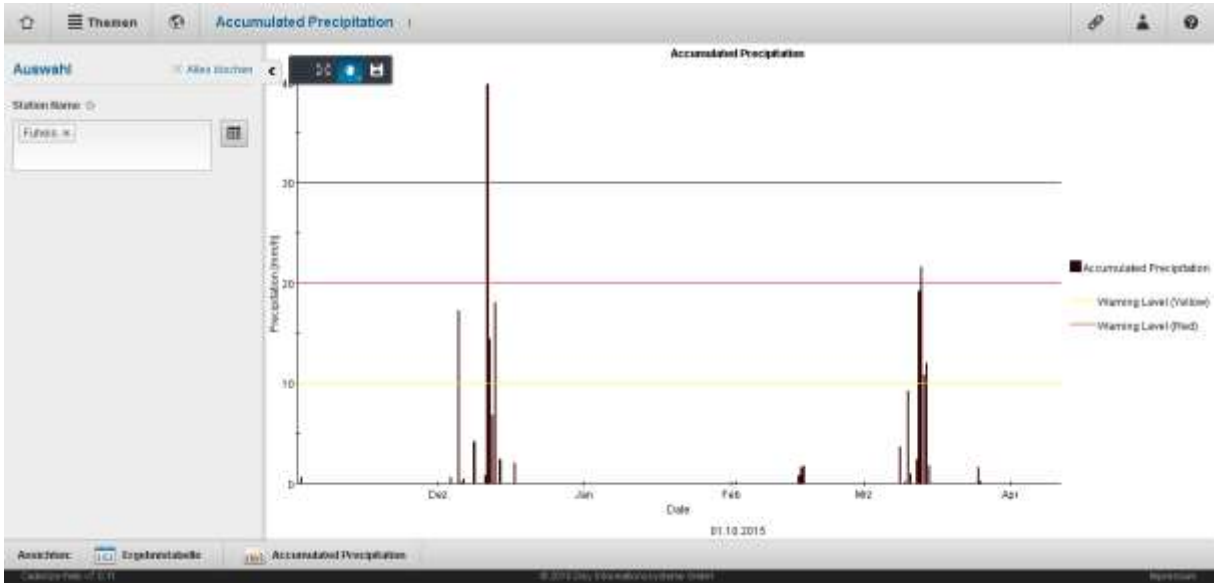
The Precipitation table displays the accumulated precipitation per station; it also indicates the warning level value according to the accumulated precipitation.



6.4.1.10 Diagram: Accumulated Precipitation

This diagram displays graphically the accumulated precipitation and when the measured values exceeded the limit values:

- Warning level (Yellow) 10mm/h
- Warning level (Red) 20 mm/h


Themen
Accumulated Precipitation
Auswahl
Accumulated Precipitation
30
20
10
0
Dez
Jan
Feb
Mrz
Apr
Date
01.10.2015
Accumulated Precipitation
Warning Level (Yellow)
Warning Level (Red)
Ansichten:
Accumulated Precipitation