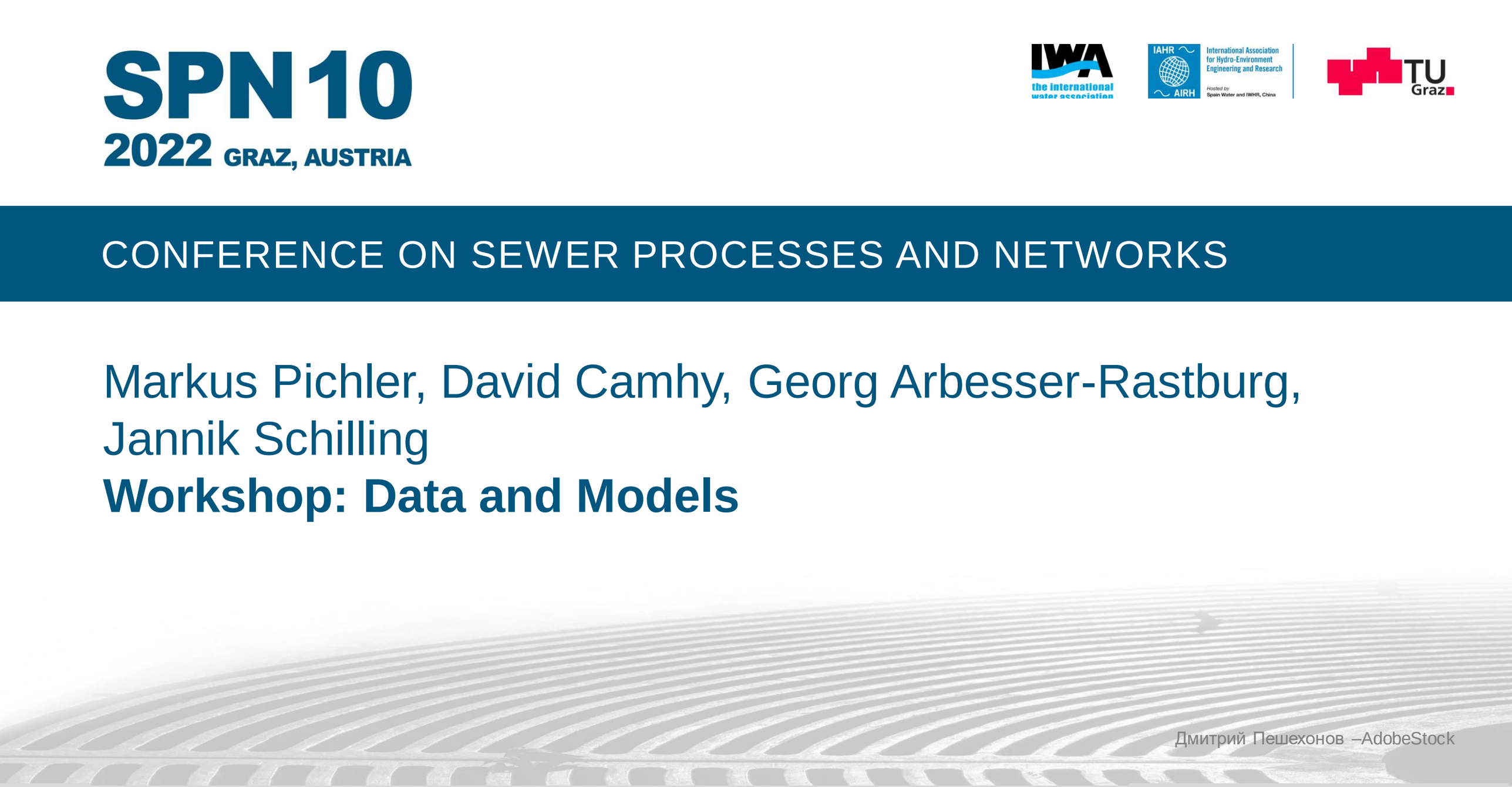
SPN10
2022 GRAZ, AUSTRIA

CONFERENCE ON SEWER PROCESSES AND NETWORKS

Markus Pichler, David Camhy, Georg Arbesser-Rastburg, Jannik Schilling

Workshop: Data and Models



Дмитрий Пешехонов –AdobeStock

Agenda

David Camhy and Georg Arbesser-Rastburg

- Data management
- Model and algorithm management

Markus Pichler

- Python interaction with SWMM (swmm-api)

Jannik Schilling

- Generate_SWMM_inp: An Open-Source QGIS Plugin to Import and Export Model Input Files for SWMM

Data Management

Collected Data Types

- **Time series data:** Measurement data and simulation results
  - Several hundred stations —> billions of data points (precipitation, water quality, pressure, flow, temperature, infrastructure monitoring, …)
  - Different temporal resolution (100 ms and upwards)
  - Long time series (up to 70 years)
  - Complex values (Arrays/Spectra)
  - Simulation results (several hundred values / timestamp)

- **Geographic Information System** data
- **Arrays** (climate forecast data)
- **Media** (pictures and videos)
- **(Arbitrary analysis results)**

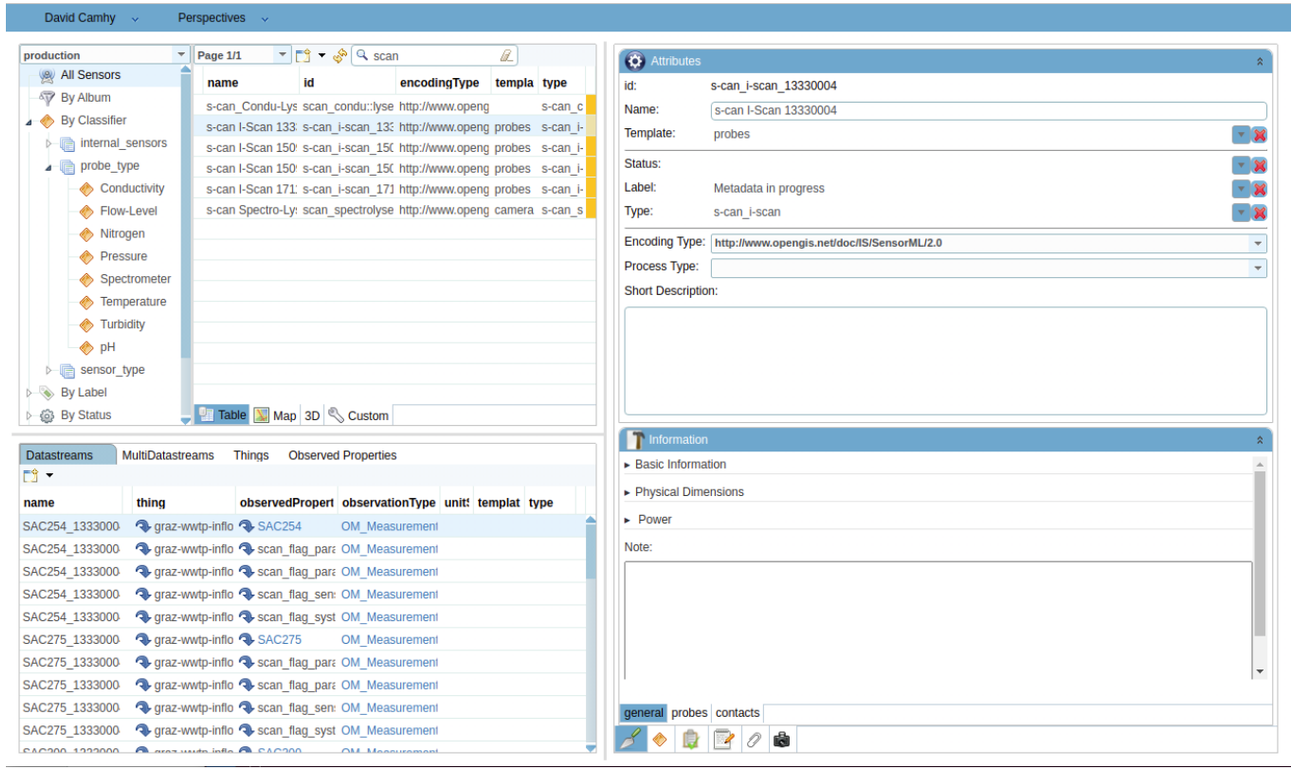
The Right Tool for the Job

- **Time series data: InfluxDB**
  - Easy visualization and alerting possibilities using Grafana
  - Very fast aggregation and analysis, efficient storage
- **Array data: TileDB and Parquet**
  - TileDB: Dense and sparse (!) array data support, parallel r/w support, versioning, fast subsetting
  - Parquet: Efficient storage format for local and distributed analysis
- **Vector GIS data: PostGIS (based on PostgreSQL)**
  - Geo-relational database
- **Dissemination/Publication: netCDF4**
  - Scientific data formats with standardized metadata (Climate and Forecast Metadata Conventions)

Measurement Metadata

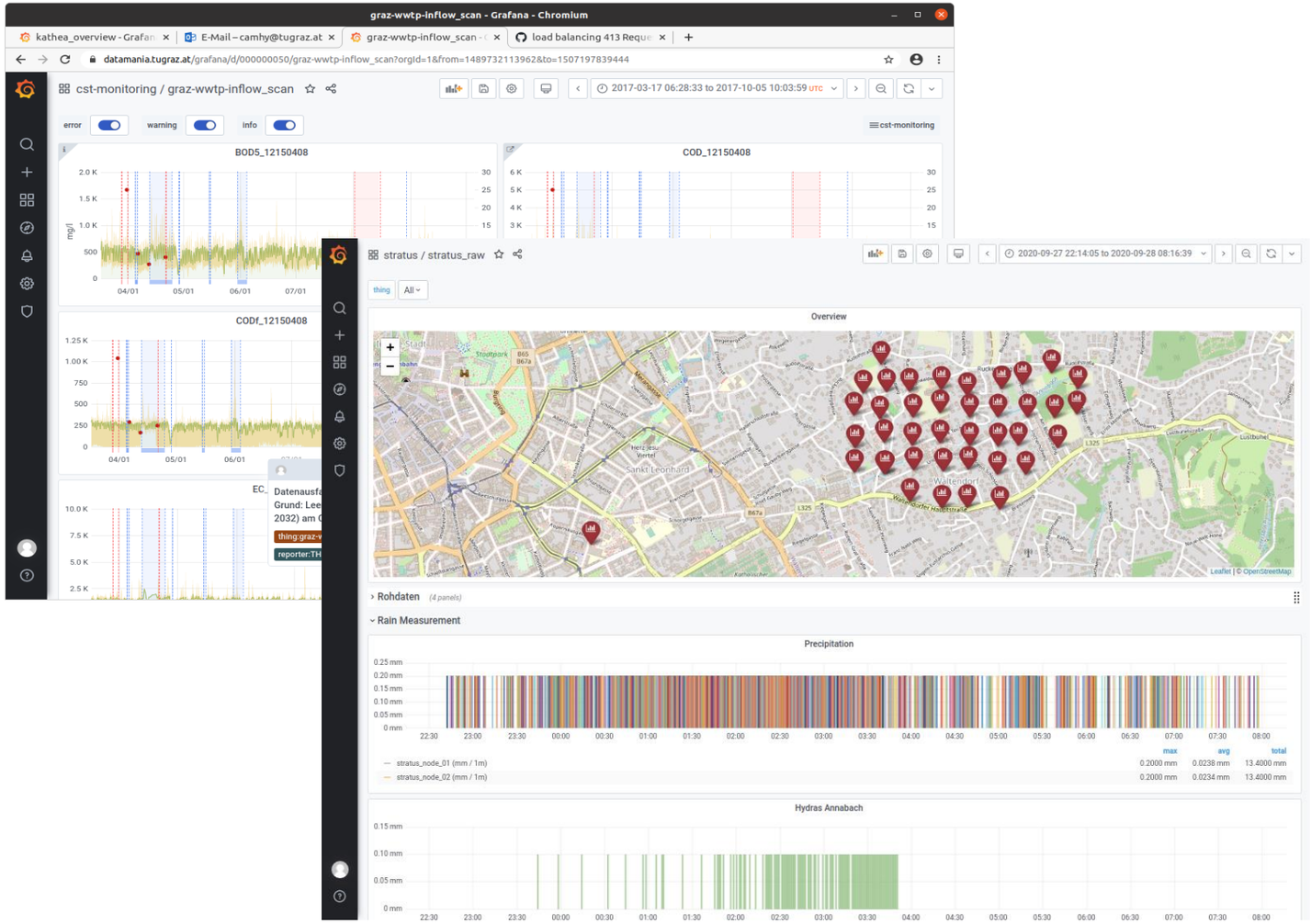
based on sensorthings (OGC Standard)

- Station and sensor metadata management
  - Geolocation, units, operational data, project metadata, ...
- Custom developed web application (UI) based on Fraunhofer's FROST Server
- Standardized description of things, sensors, datastreams, observed properties,...



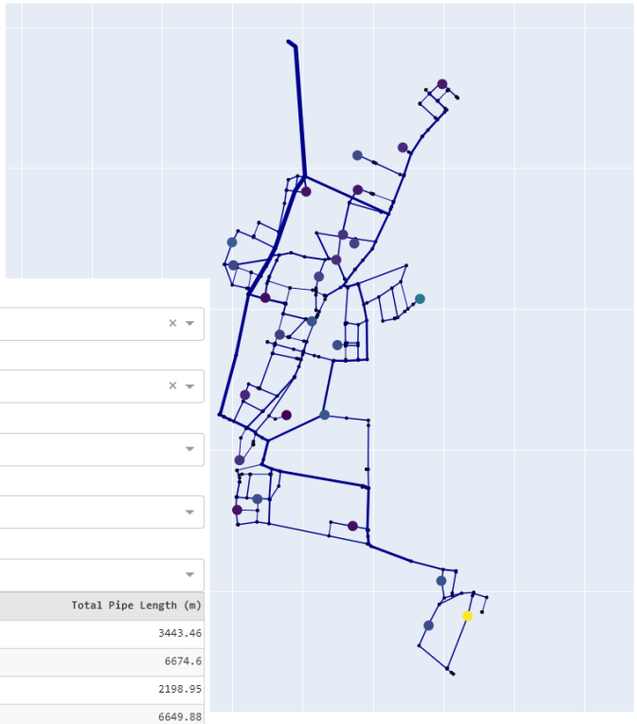
Data and Analysis Visualization – Grafana

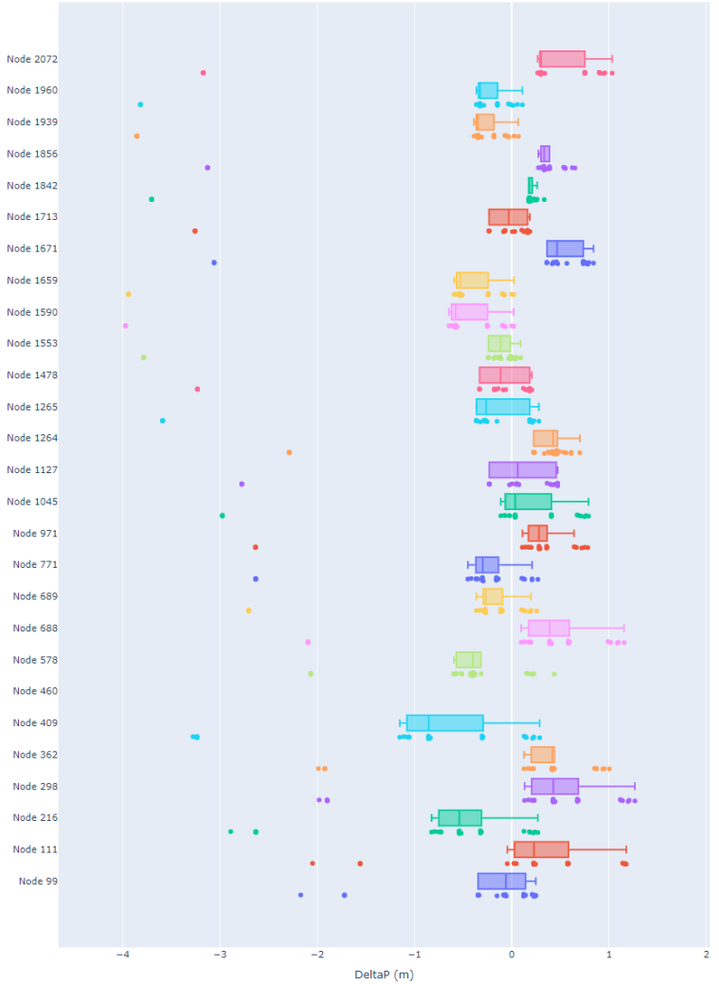
- Powerful web-based time series visualization solution
- Easy to setup and use
- Supports alerting and annotations



Data and Analysis Visualization – Dash

- Python toolkit
- Generates interactive visualization and analysis web-apps





Network

WienGruenII_2019_MN_ElevAdj_skel.inp

Metadata

WienGruenII_2019_skel.meta

Material

YES

Age

Select...

Type

Select...

Diameter

Select...

Zone

Select...

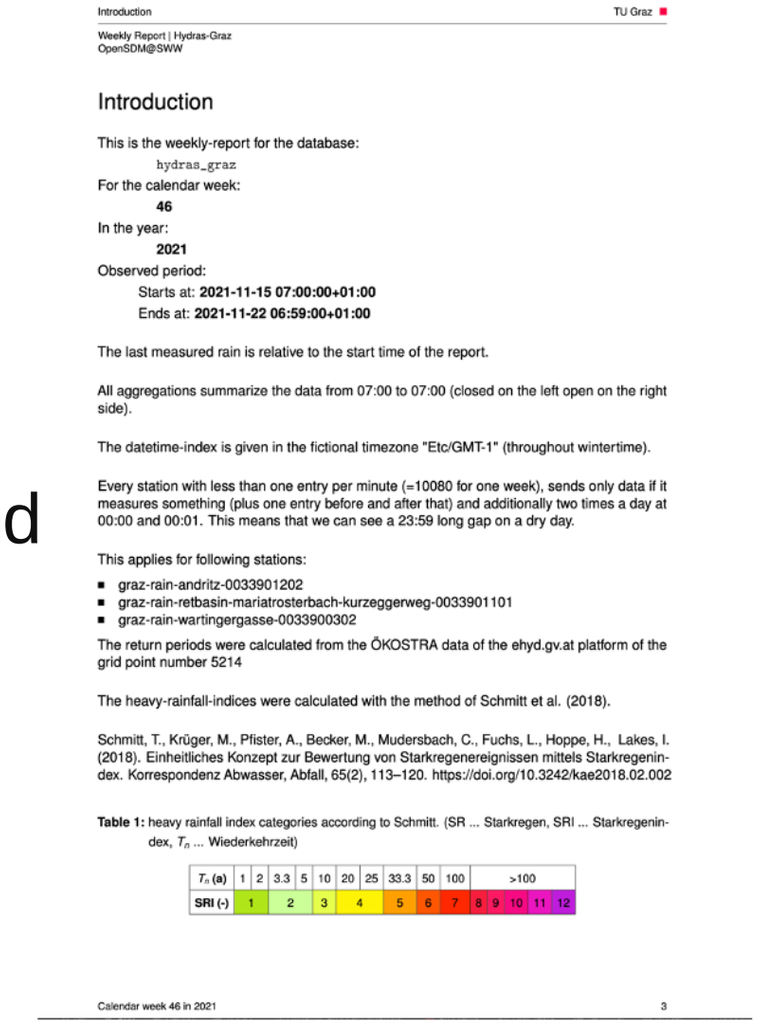
Installation Density

Select...

Group No.	Group	Number of Pipes	Mean Pipe Length (m)	Mean Pipe Diameter (mm)	Total Pipe Length (m)
1	AZ	32	107.61	356.25	3443.46
2	GG	88	75.85	173.58	6674.6
3	PE	28	78.53	154.89	2198.95
4	SG	80	83.12	176	6649.88
5	SGzm	247	54.37	161.04	13429.48
6	St	11	43.29	177.27	476.22

Reporting

- Weekly report of all precipitation measurements in Graz
- Multiple visualizations and statistical analysis
- Sent to all partners automatically once per week



Weekly Report | Hydras-Graz
OpenSDM@SWW

Introduction

This is the weekly-report for the database:

`hydras_graz`

For the calendar week:

46

In the year:

2021

Observed period:

Starts at: **2021-11-15 07:00:00+01:00**
Ends at: **2021-11-22 06:59:00+01:00**

The last measured rain is relative to the start time of the report.

All aggregations summarize the data from 07:00 to 07:00 (closed on the left open on the right side).

The datetime-index is given in the fictional timezone "Etc/GMT-1" (throughout wintertime).

Every station with less than one entry per minute (=10080 for one week), sends only data if it measures something (plus one entry before and after that) and additionally two times a day at 00:00 and 00:01. This means that we can see a 23:59 long gap on a dry day.

This applies for following stations:

- graz-rain-andritz-0033901202
- graz-rain-retbasin-mariatrosterbach-kurzeggerweg-0033901101
- graz-rain-wartingergasse-0033900302

The return periods were calculated from the ÖKOSTRA data of the ehyd.gv.at platform of the grid point number 5214

The heavy-rainfall-indices were calculated with the method of Schmitt et al. (2018).

Schmitt, T., Krüger, M., Pfister, A., Becker, M., Mudersbach, C., Fuchs, L., Hoppe, H., Lakes, I. (2018). Einheitliches Konzept zur Bewertung von Starkregenereignissen mittels Starkregenindex. Korrespondenz Abwasser, Abfall, 65(2), 113–120. https://doi.org/10.3242/kae2018.02.002

Table 1: heavy rainfall index categories according to Schmitt. (SR ... Starkregen, SRI ... Starkregenindex, T_n ... Wiederkehrzeit)

T_n (a)	1	2	3.3	5	10	20	25	33.3	50	100	>100				
SRI (-)	1		2		3		4		5	6	7	8	9	10	11	12

Table 4: daily precipitation dur in HH:MM

	Mon. 15.11.	Tue. 16.11.	Wed. 17.11.	Thu. 18.11.	Fri. 19.11.	Sat. 20.11.	Sun. 21.11.	SUM
Feldkirchen-Airport	00:01	00:00	00:00	00:00	00:00	00:00	00:00	00:01
Andritz	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
Annabach-Lang	00:01	00:00	00:00	00:00	00:00	00:00	00:00	00:01
Gösting-Thalstrasse	00:01	00:00	00:00	00:00	00:00	00:00	00:00	00:01
Petersbach	–:–	–:–	–:–	–:–	–:–	–:–	–:–	–:–
Petrifelderstrasse	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
Prochaskagasse-School	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
Bründlbach-Krottendorferstr-RB	–:–	–:–	–:–	–:–	–:–	–:–	–:–	–:–
Einödbach-Schererpark-RB	–:–	–:–	–:–	–:–	–:–	–:–	–:–	–:–
Mariatrosterbach-Kurzeggerweg-RB	00:03	00:00	00:00	00:00	00:00	00:00	00:00	00:03
Stufenbach-Ziegelstr-RB	–:–	–:–	–:–	–:–	–:–	–:–	–:–	–:–
Thal-Erlenbach-RB	00:01	00:00	00:00	00:00	00:00	00:00	00:00	00:01
Sankt-Johann-School	06:38	05:03	06:30	08:38	07:19	07:50	07:16	45:14
Stiftingbach-II	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
Strassgang-Bath	00:04	00:00	00:00	00:00	00:00	00:00	00:00	00:04
Strassgang-Zamg	00:03	00:00	00:00	00:00	00:00	00:00	00:00	00:03
Stremayrgasse	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
University	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
Wartingergasse	00:03	00:00	00:00	00:00	00:00	00:00	00:00	00:03
Zusertalgasse	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
Andritzbach-Hügelweg-RB	00:01	00:00	00:00	00:00	00:00	00:00	00:00	00:01
Stattegg-Höllbach-RB	00:02	00:00	00:00	00:00	00:00	00:00	00:00	00:02
Weinitzen-Schöcklbach-RB	00:03	00:00	00:00	00:00	00:00	00:00	00:00	00:03

Markus Pichler

Discussion

What solutions do you use to store which kind of data?

How do you store your measurement metadata?

Do you use web based visualization and analysis solutions?

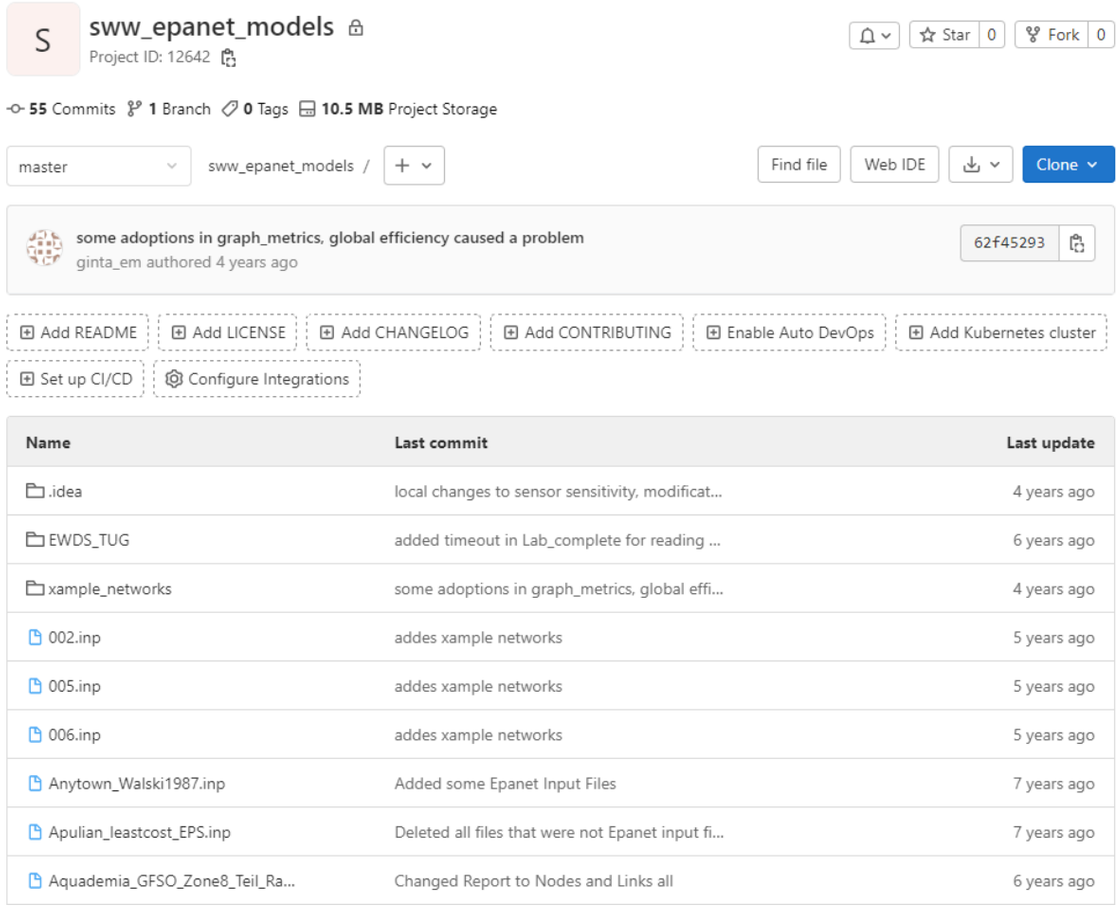
How do you exchange data with project partners?

Which formats do you use for dissimination and publication?

Model and Algorithm Management

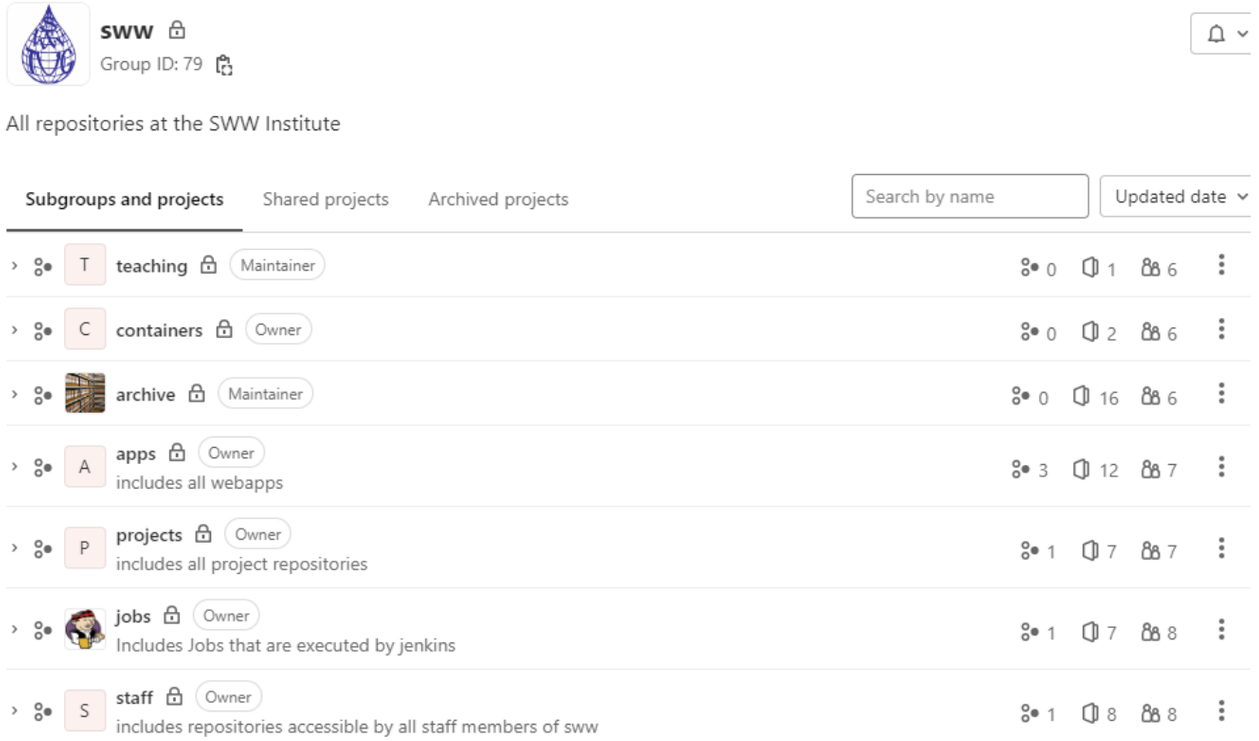
Model Storage

- Models
  - EPANET models (inp-files)
  - SWMM models (inp-files)
- Current solution
  - Models stored on network drives
  - Models stored in Git repositories
  - Models only locally available on workstations
- No central model storage, versioning only in Git repositories



Algorithm Management

- Algorithms managed in Git repositories
  - General algorithm and script repository
  - Project-related repositories
  - User-related repositories
  - Algorithm-related repositories



Algorithm Execution

- **HTCondor**
  - Supports many (in our case up to several 100000) long running simulations
  - Allows execution on user workstations (Linux and Windows)
- **Jenkins**
  - Scheduled and interactive jobs
  - Data collection and analysis, educational jobs
- **Argo Workflows (work in progress)**
  - Complex pipelines for distributed cluster workflows
- **External Clusters/Supercomputers**
  - Is used when Institute infrastructure is not sufficient (RAM, CPUs, …)

Issues we face

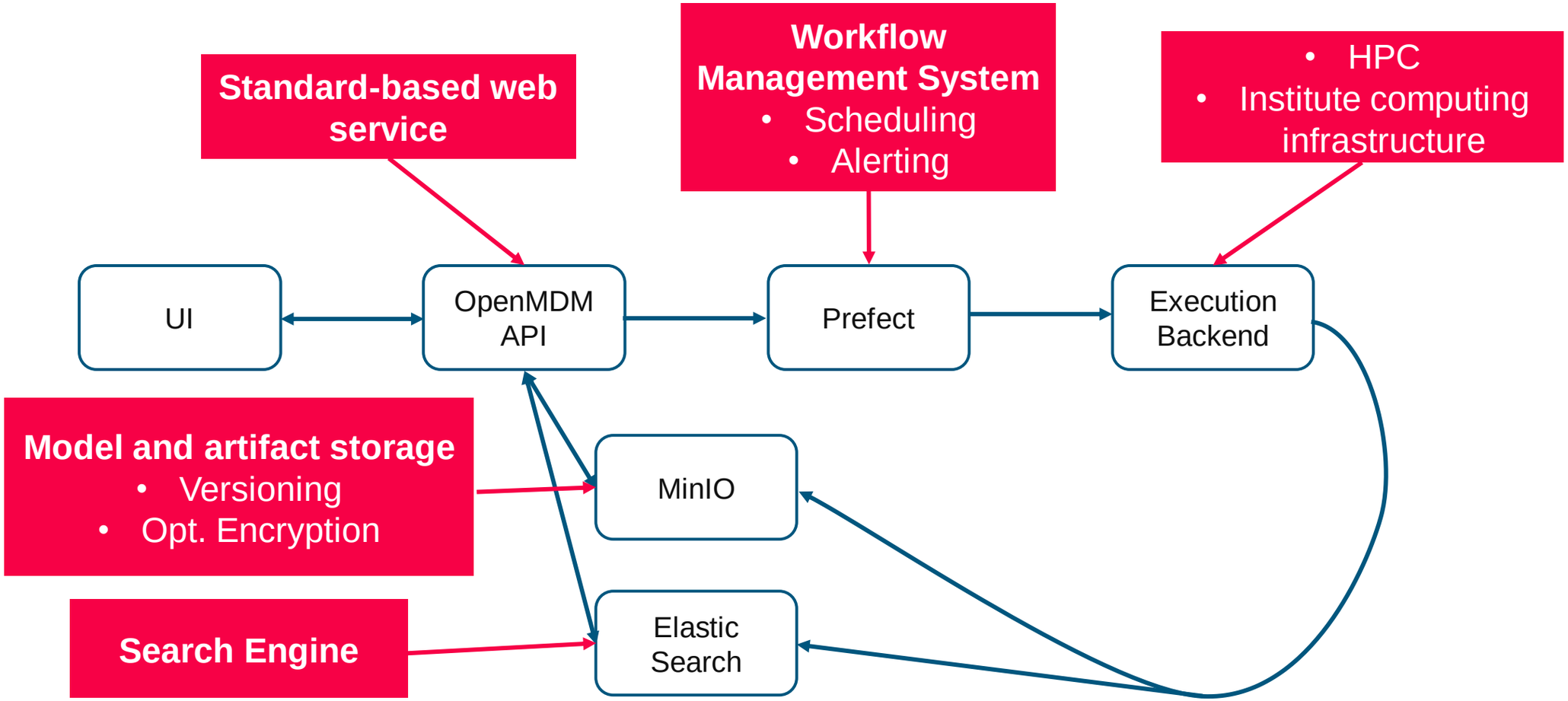
- Results are stored (different databases, files on a server, …) but there is no link to the algorithms and models used
- Running old algorithms again takes time to set up (unclear dependencies, no dependency versions)
- There is no central model storage → models are scattered and hard to find
- No or not standardized model metadata

Open Model Data Management - OpenMDM

- Tool with web-based user interface for
  - Managing hydraulic water distribution and sewer models, model metadata and model versions as well as results
  - Starting containerized computations in a computing cluster

- **Goals**
  - Creating a framework that allows for containerized workflows that
    - Extract metadata from models
    - Store (versioned) models and the corresponding metadata in a central storage
    - Start modelling tasks in computing clusters
    - Support different execution backends like HTCondor or Argo Workflows
    - Store simulation artifacts in a central storage

OpenMDM – Current Concept



Discussion

How do you store models and simulation artifacts?

How do you manage your algorithms?

How do you ensure reproducible workflows?

What kind of model and simulation metadata would you need for your scientific work?

swmm-api

Markus Pichler

Why?

Why SWMM?

- Open Source
- Widely used in science (so do we)

EPA SWMM

Why Python?

- also widely used in science
- reasonably fast
- relatively easy to learn

python™

Why another one?

Bryant E. McDonnell (UDM 2022): “Everyone was just wrapping SWMM Input files and SWMM Output files with Python”

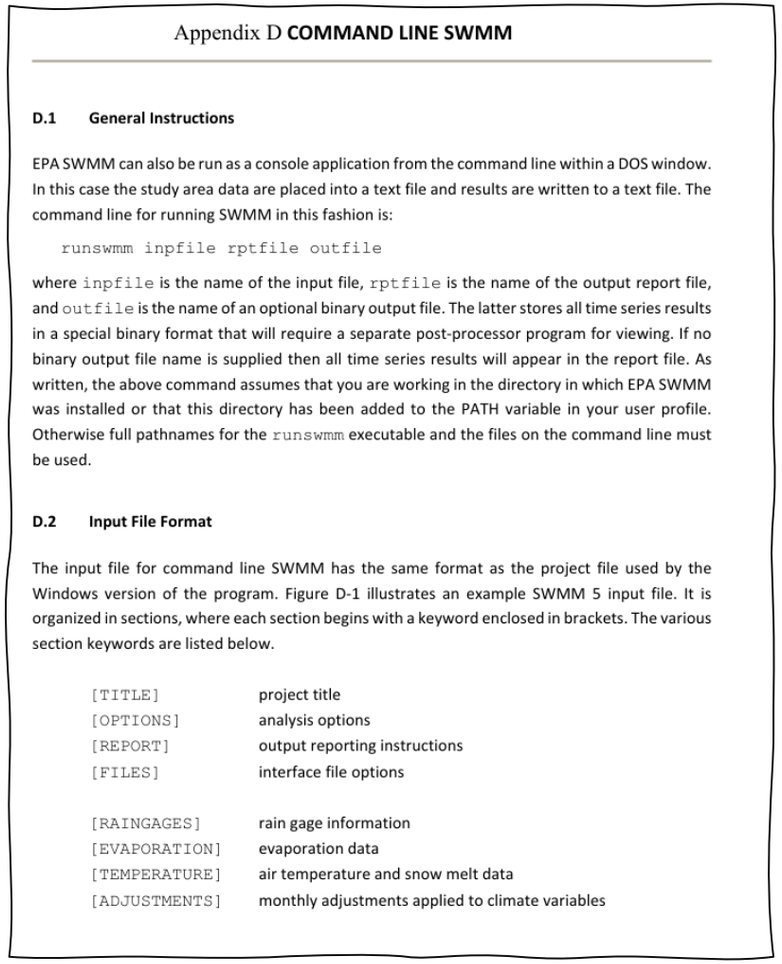
Aren’t there already a lot of them?

- yes, but …
  - availability
  - speed
  - usability
  - extensible

Based on the SWMM command line interface

What information is in the .inp-file

How the .inp-file is structured



Appendix D **COMMAND LINE SWMM**

D.1 General Instructions

EPA SWMM can also be run as a console application from the command line within a DOS window. In this case the study area data are placed into a text file and results are written to a text file. The command line for running SWMM in this fashion is:

```
runswmm inpfile rptfile outfile
```

where `inpfile` is the name of the input file, `rptfile` is the name of the output report file, and `outfile` is the name of an optional binary output file. The latter stores all time series results in a special binary format that will require a separate post-processor program for viewing. If no binary output file name is supplied then all time series results will appear in the report file. As written, the above command assumes that you are working in the directory in which EPA SWMM was installed or that this directory has been added to the PATH variable in your user profile. Otherwise full pathnames for the `runswmm` executable and the files on the command line must be used.

D.2 Input File Format

The input file for command line SWMM has the same format as the project file used by the Windows version of the program. Figure D-1 illustrates an example SWMM 5 input file. It is organized in sections, where each section begins with a keyword enclosed in brackets. The various section keywords are listed below.

`[TITLE]`	project title
`[OPTIONS]`	analysis options
`[REPORT]`	output reporting instructions
`[FILES]`	interface file options
`[RAINGAGES]`	rain gage information
`[EVAPORATION]`	evaporation data
`[TEMPERATURE]`	air temperature and snow melt data
`[ADJUSTMENTS]`	monthly adjustments applied to climate variables

Main functionalities

- Reading SWMM input files (.inp)
- Modifying model components and settings
- Simulating models using SWMM command-line tool or PySWMM
- Reading results (.out and .rpt) as pandas data objects

Use cases

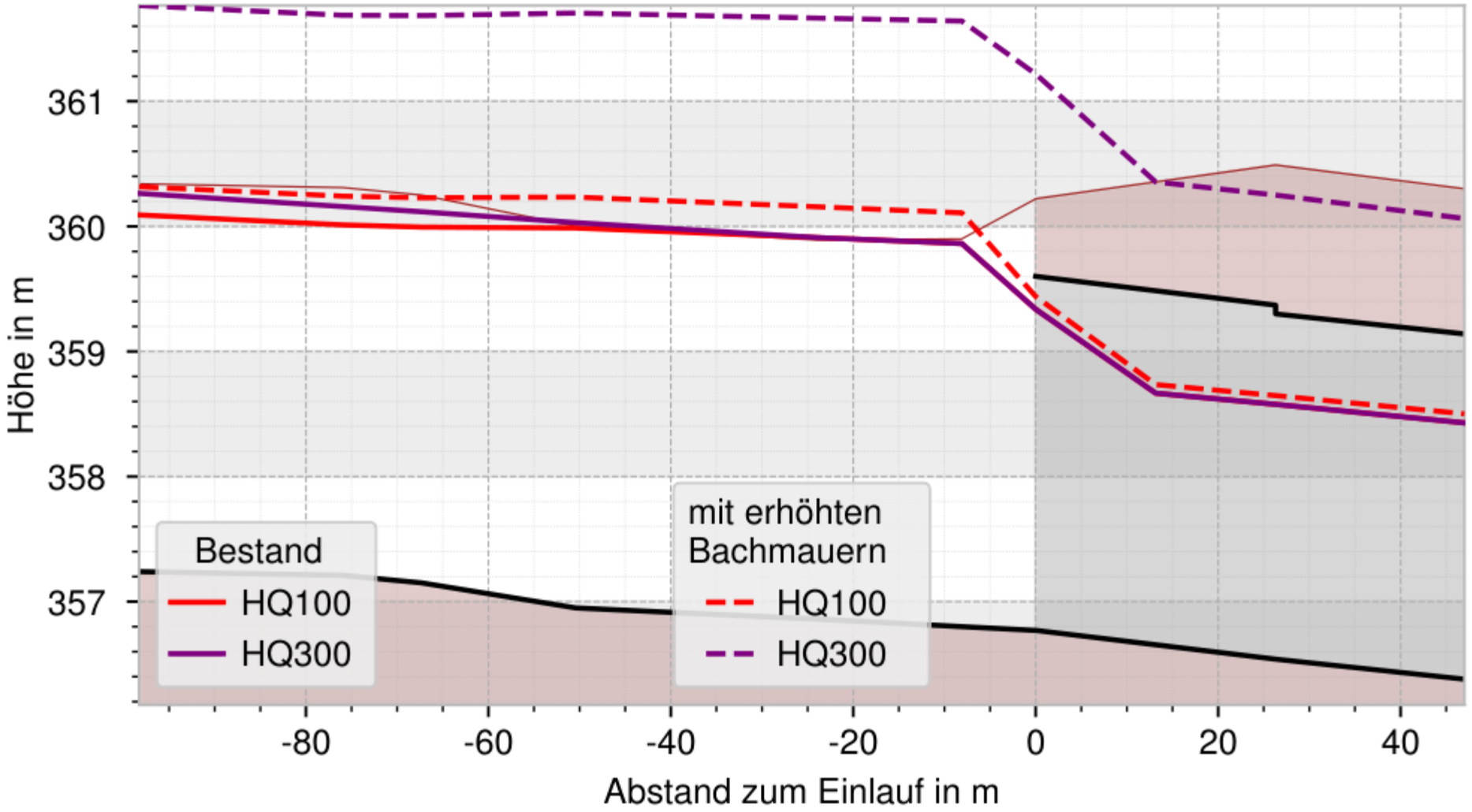
event based long term simulations

sensitivity analysis

creating a SWMM model with GIS data

working with a modified SWMM

creating beautiful plots for publications


361
360
359
358
357
Höhe in m
-80
-60
-40
-20
0
20
40
Abstand zum Einlauf in m
Bestand
HQ100
HQ300
mit erhöhten
Bachmauern
HQ100
HQ300

Links

Installation

- `pip install swmm-api`
- https://pypi.org/project/swmm-api/

Documentation

- https://markuspichler.gitlab.io/swmm_api

Examples / Tutorials

- https://markuspichler.gitlab.io/swmm_api/examples/

Files in this workshop

https://gitlab.com/markuspichler/swmm_api/-/tree/master/examples/spn10