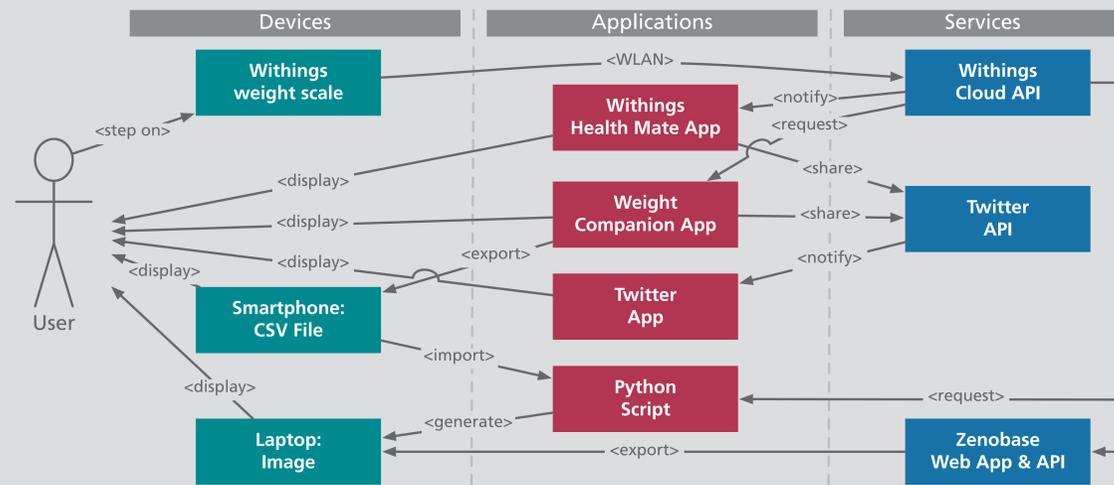


Towards Provenance Capturing of Quantified Self Data



Example Quantified Self workflow for weight tracking using a Withings online weight scale.

Quantified Self

Self-tracking with wearable devices, smartphone apps, or desktop applications became popular in recent years. Such a self-surveillance is called *Quantified Self* (QS). It is a community of people who record and analyse data about themselves for medical reasons, self-improvement, technological interests, or other reasons.

Capture Methods for Provenance

The following techniques are available for capturing provenance of „real-world“ QS workflows:

- **Wearable devices and other sensors** are usually very closed embedded systems. Adding functionality is not designated by their manufacturers. Adding functionality for capturing provenance would only be possible for devices with programming interfaces.

- **Smartphone, desktop, or web-based applications** usually need specific techniques for capturing provenance. Depending on their architecture and availability, changing the application's code itself or adding functionality via an API are most desirable. Other techniques are:
 - Wrapping the application and capture information from files or databases.
 - Grabbing information from the communication infrastructure.
 - Dynamic instrumentation of code or byte code.
- **Web Services or Cloud Storage** are often provided by manufacturers of tracking devices. Capturing provenance on Cloud resources is often not possible. Nevertheless, one could capture some provenance information via their APIs. For example, with capturing requests to the Web Service on client side or registering a callback listener for change notifications and capture all changes locally.

Example Provenance

We show the feasibility of some approaches with a practical „slice of life“ example from the weight tracking workflow: the weight data is exported from an Android application to a CSV file, then this CSV file is imported and visualized by a Python script.

- We added provenance capturing to the *WeightCompanion*. The application generates provenance documents using the Java library *ProvToolbox*, which we ported to Android.
- The Python script imports a CSV file and visualizes the weights data. The CSV file is imported into a *pandas* DataFrame object. Then the data is visualized using *matplotlib*. Provenance is captured using a Python library for PROV-DM.
- Both examples send the PROV document to the *ProvStore* directly. We merged the individual PROV documents using *ProvToolbox*.

```

# Provenance-related Imports
from prov.model import ProvDocument, PROV
from provstore.api import Api
from time import gmtime, strftime

# Create a new provenance document
prov = ProvDocument()

# Add namespaces
prov.add_namespace('qs', 'http://software.dlr.de/qs')
prov.add_namespace('userdata', 'http://software.dlr.de/userdata')
prov.add_namespace('prov', 'http://software.dlr.de/prov')
prov.add_namespace('graphic', 'http://software.dlr.de/graphic')
prov.add_namespace('library', 'https://pypi.python.org/pypi/%(name)s')
prov.add_namespace('python_method', 'http://www.provstore.de/python_method')

# The user
agent_user = prov.agent('user:onyame@googlemail.com')

# Application Import
from pandas import DataFrame, Series, read_csv
import matplotlib.pyplot as plt

prov.entity('library:pandas', {'library:version': '0.17.1'})
prov.entity('library:matplotlib', {'library:version': '1.5.1'})

# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:ts' % WC_FILE)
prov.wasAttributedTo(entity_csvfile, agent_user)
wc_data = read_csv(WC_FILE, parse_dates=['Timestamp', 'index_col=1Timestamp'])

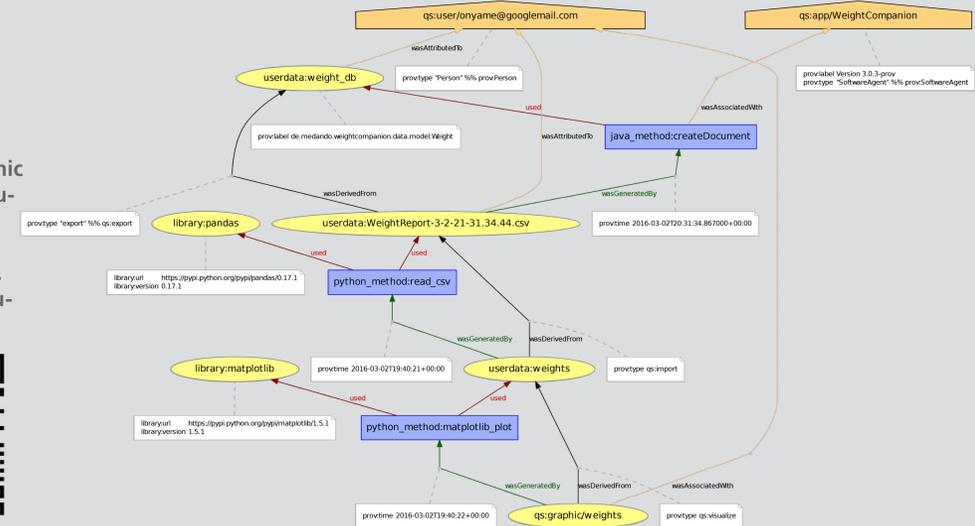
# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userdata:weights')
prov.activity('python_method:read_csv')
prov.wasGeneratedBy(entity_weights, 'python_method:read_csv', strftime('%Y%m%dT%H%M%S%Z', gmtime()))
prov.used('python_method:read_csv', entity_csvfile)
prov.wasDerivedFrom(entity_weights, entity_csvfile)
other_attributes = {'prov:type': 'qs:import'}
prov.used('python_method:read_csv', 'library:pandas')

# Plot the weights
weights.plot(title = 'Weight', legend=False)
entity_plot_weights = prov.entity('graphic:weights')
    
```

Recording the provenance of a Python script and within the Android app *WeightCompanion* (<https://play.google.com/store/apps/details?id=de.medando.weightcompanion>).



Merged provenance of a graphic of weight measurements (<https://provenance.ecs.soton.ac.uk/store/documents/113794/>).



Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

Andreas Schreiber
German Aerospace Center (DLR)
Distributed Systems and Component Software
Linder Höhe, 51147 Köln, Germany
andreas.schreiber@dlr.de

Doreen Seider
Medando UG (haftungsbeschränkt)
50670 Köln, Germany
doreen.seider@medando.de

