DLR Knowledge for Tomorrow

Detecting and Visualizing Privacy Issues of Self-Tracking using Provenance Comics

Andreas Schreiber
German Aerospace Center (DLR)

Introduction

Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

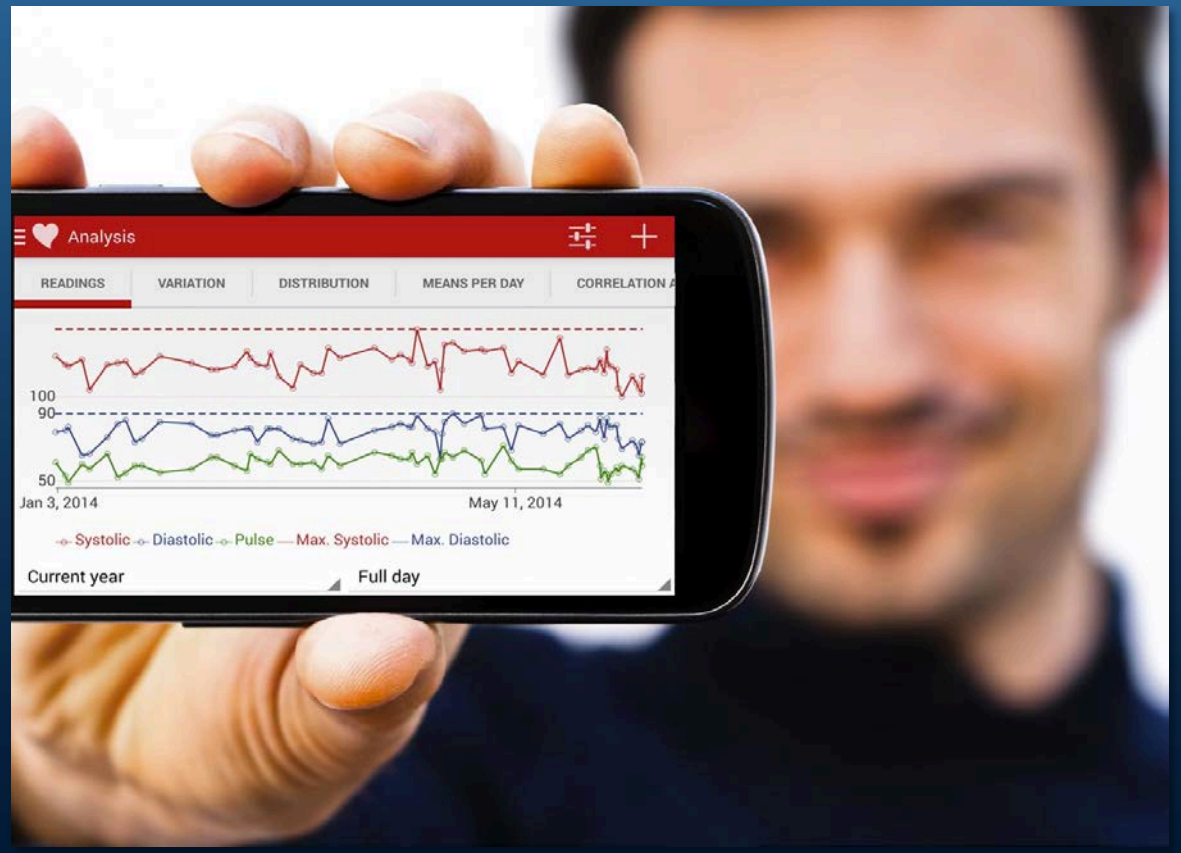
Simulation and Software Technology, Cologne/Berlin
Head of Intelligent and Distributed Systems department

Institute of Data Science, Jena
Head of Secure Software Engineering group

Co-Founder
Data Scientist
Patient

Motivation

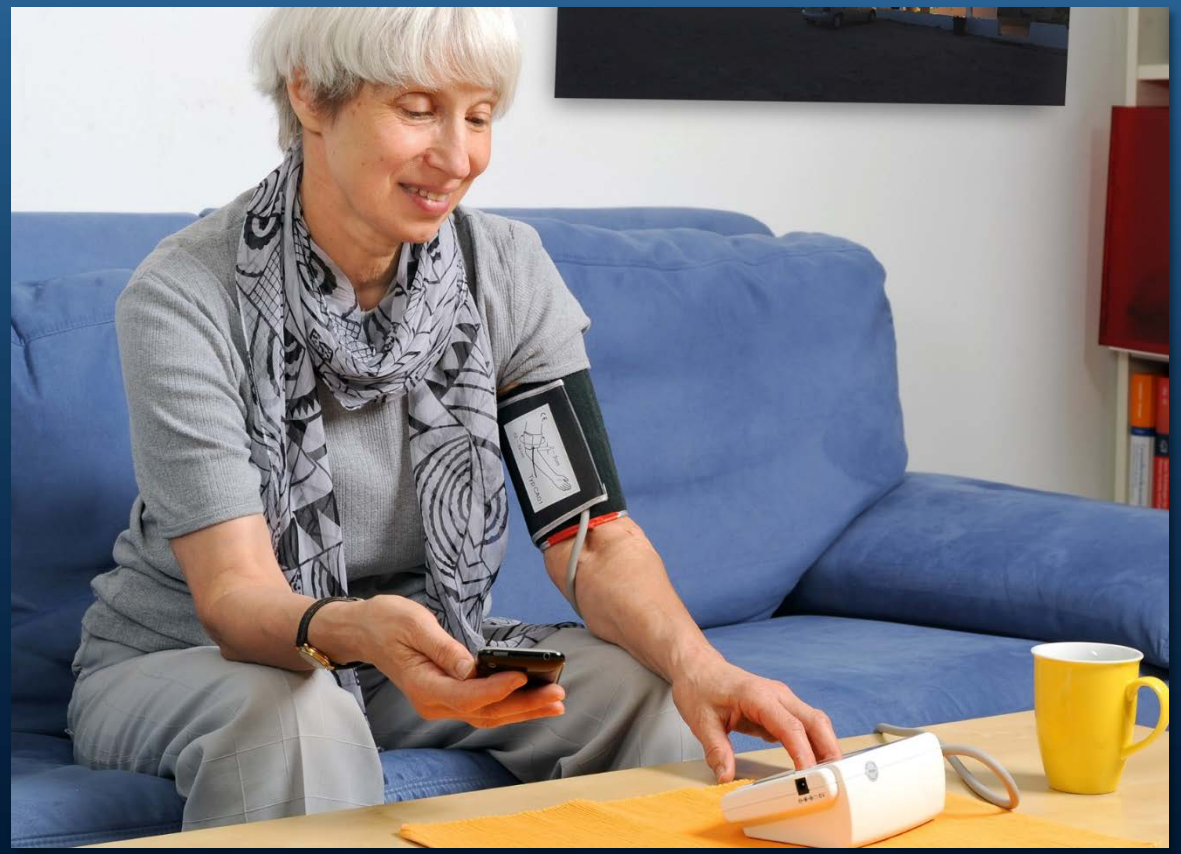
Quantified Self (n = 1)



Medical Trials (n > 1)

Motivation

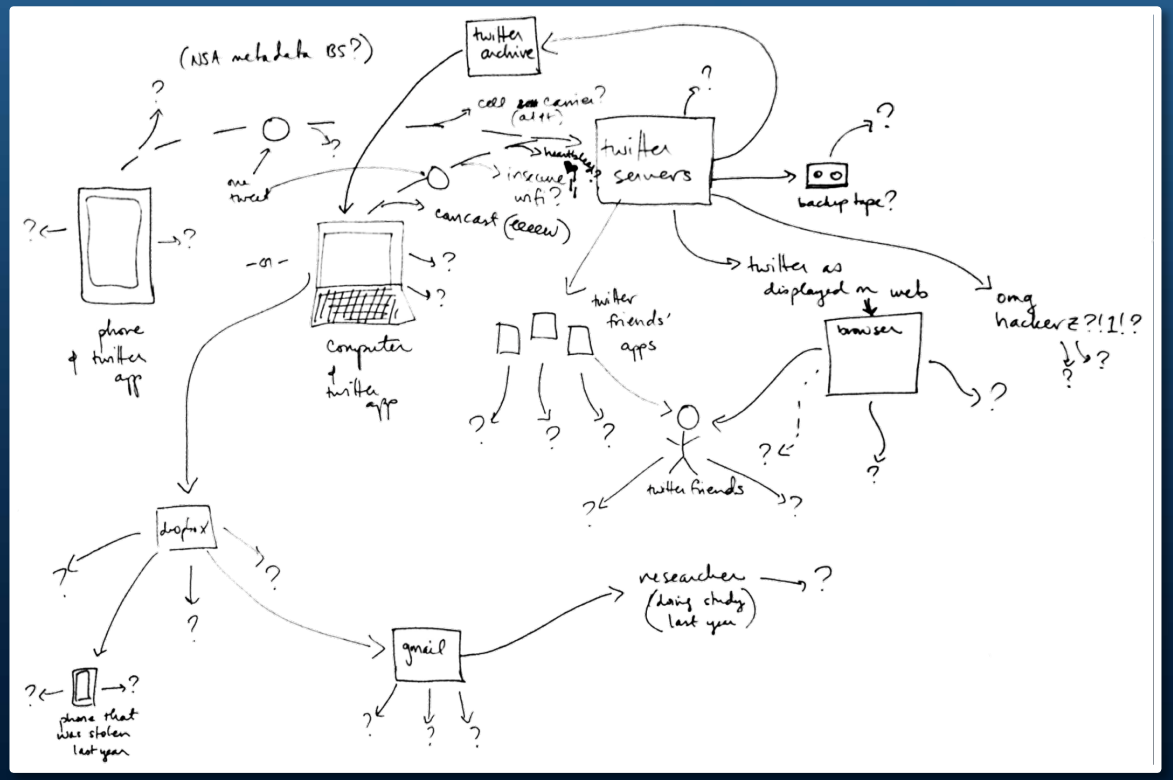
Telemedicine

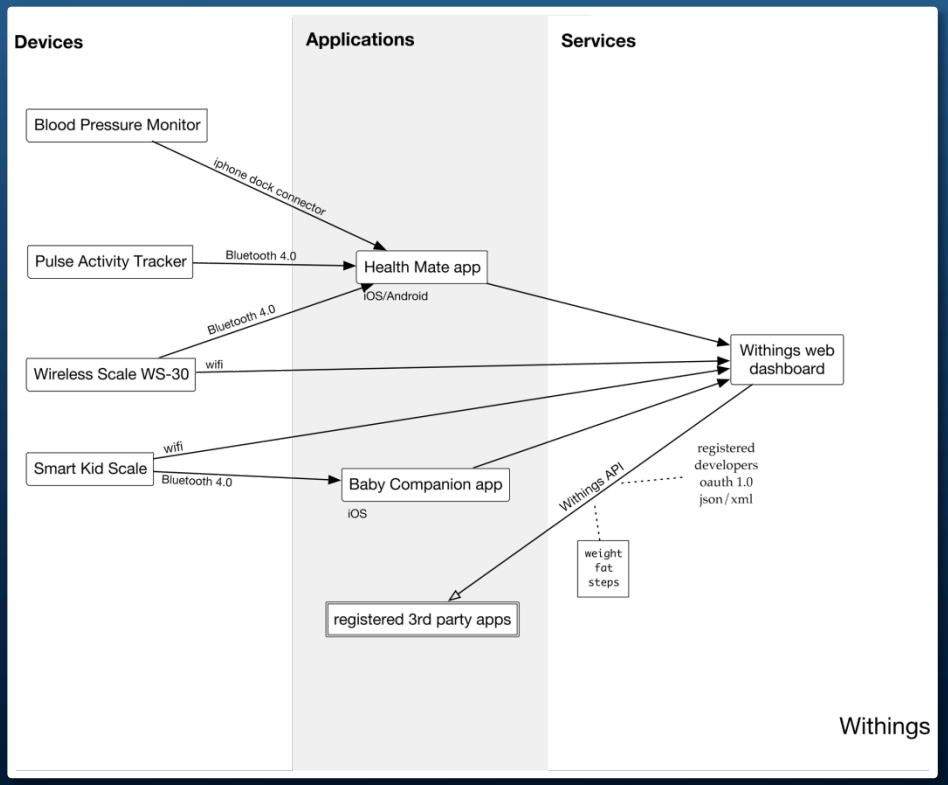


Medical experiments

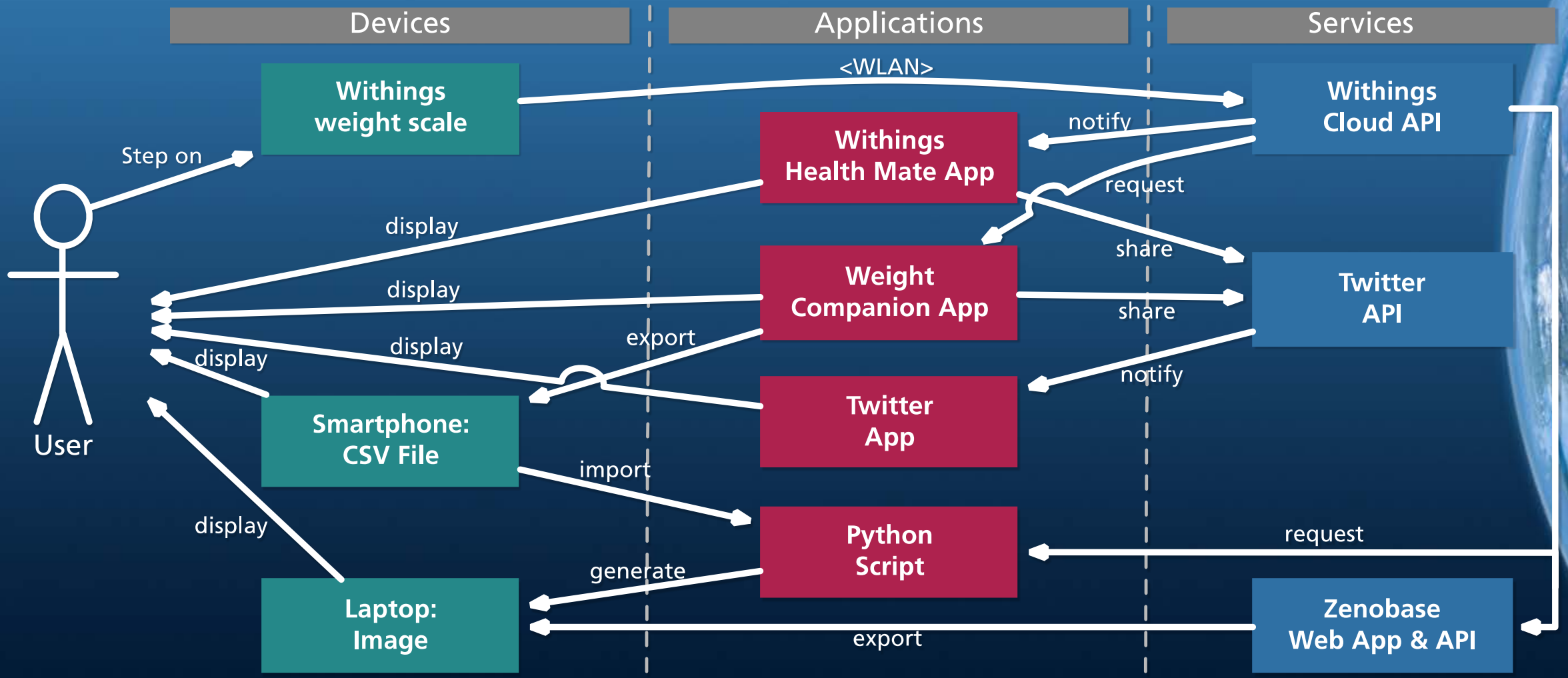
Understand, how Quantified Self data has been produced, processed, stored, accessed, …

Pictures from *Breakout Session on Mapping Data Access* (2014 QS Europe Conference, Amsterdam)
`https://forum.quantifiedself.com/t/breakout-mapping-data-access/995`





Example: Weight Tracking Workflow



Questions related to Quantified Self

Data

- What data about the user were created during the activity X?
- What data about the user were automatically generated?
- What data about the user were derived from manual input?

Apps and Services

- Which activities support visualization of the users data?
- In which activities can the user input data?
- What processes are communicating data?

Access and Privacy

- What parties were involved in generating data X?
- What parties got access on data X?
- Can other parties see user’s data X?

Provenance

Provenance is

information about entities, activities, and people involved in

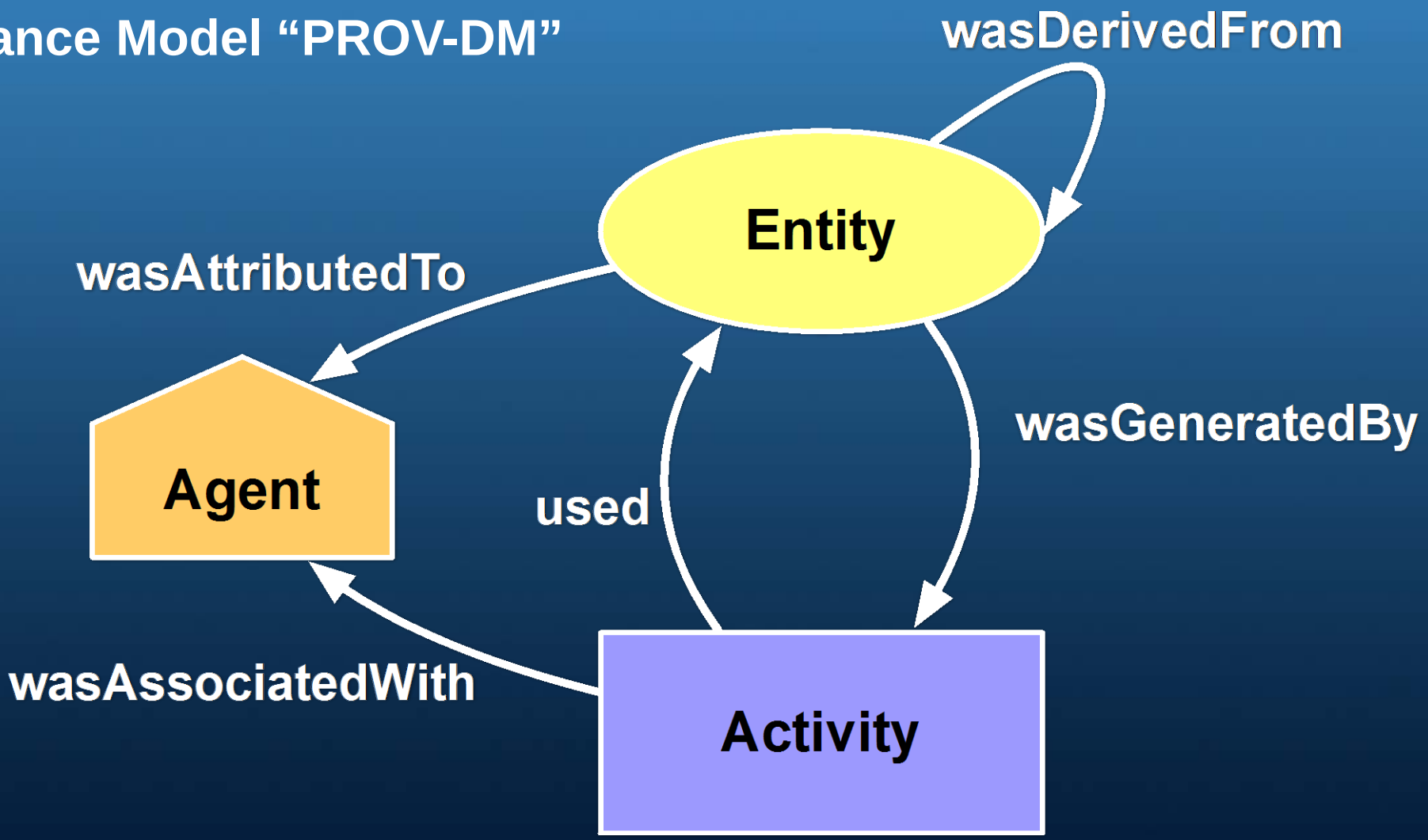
producing a piece of data or thing,

which can be used to form

assessments about its quality, reliability or trustworthiness.

PROV W3C Working Group
https://www.w3.org/TR/prov-overview

Provenance Model “PROV-DM”



Provenance Model for Quantified Self

Sub models for basic QS *Activities*

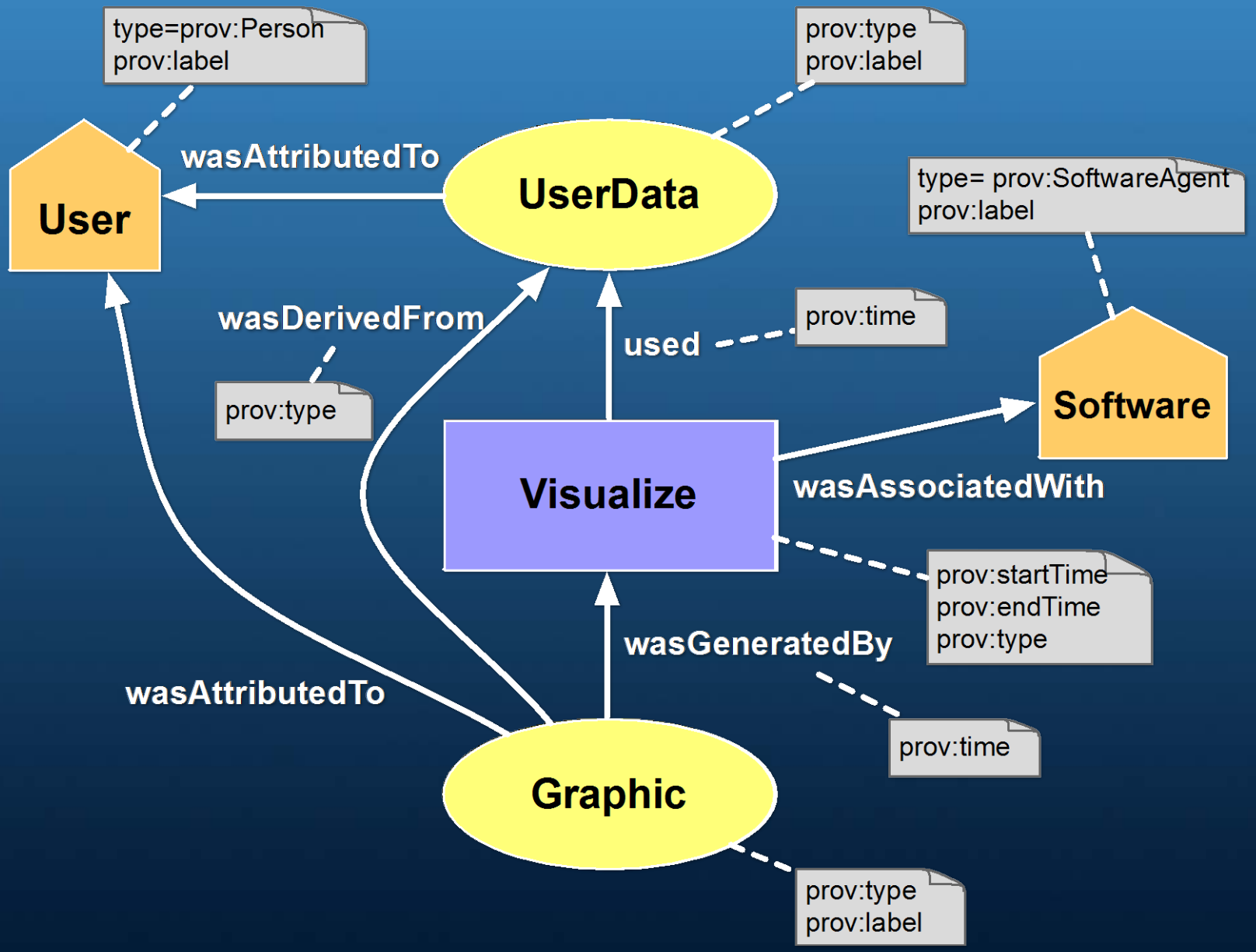
- Input
- Sensing
- Export
- Request
- Aggregate
- Visualize

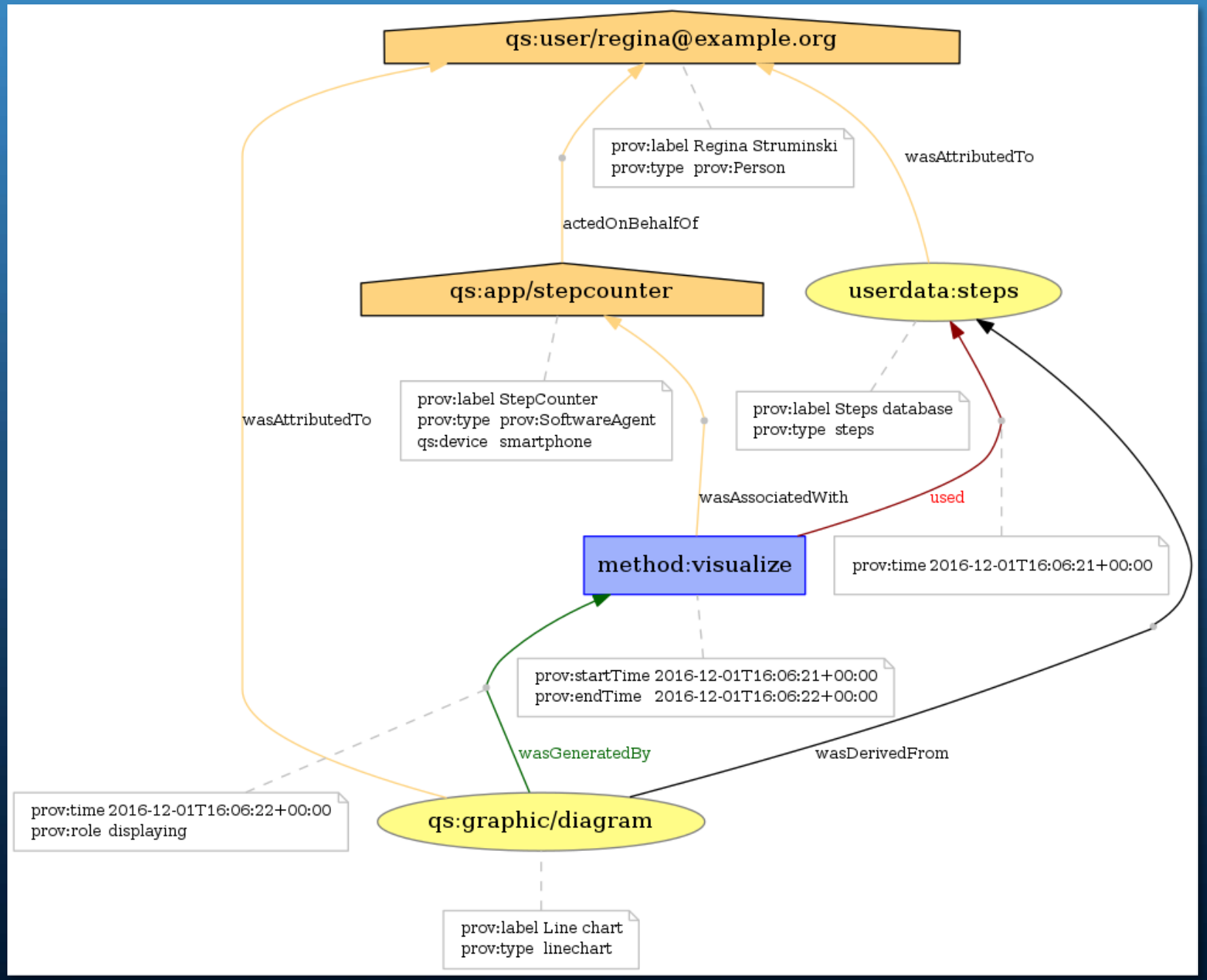
The activities generate or change data that is associated or attributed to *Agents*

- Users
- Software
- Organizations

References

- Schreiber, A. (2016) ***A Provenance Model for Quantified Self Data***. In: Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices: 10th International Conference, UAHCI 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings, Part I, Springer, 382-393
- Schreiber A., Seider D. (2016) ***Towards Provenance Capturing of Quantified Self Data***. In: Provenance and Annotation of Data and Processes. IPAW 2016. Lecture Notes in Computer Science, vol 9672. Springer, Cham


type=prov:Person
prov:label
prov:type
prov:label
wasAttributedTo
User
UserData
type= prov:SoftwareAgent
prov:label
wasDerivedFrom
prov:type
used
prov:time
Software
Visualize
wasAssociatedWith
prov:startTime
prov:endTime
prov:type
wasGeneratedBy
wasAttributedTo
prov:time
Graphic
prov:type
prov:label


qs:user/regina@example.org
prov:label Regina Struminski
prov:type prov:Person
wasAttributedTo
actedOnBehalfOf
qs:app/stepcounter
userdata:steps
wasAttributedTo
prov:label StepCounter
prov:type prov:SoftwareAgent
qs:device smartphone
prov:label Steps database
prov:type steps
wasAssociatedWith
used
method:visualize
prov:time 2016-12-01T16:06:21+00:00
prov:startTime 2016-12-01T16:06:21+00:00
prov:endTime 2016-12-01T16:06:22+00:00
wasGeneratedBy
wasDerivedFrom
prov:time 2016-12-01T16:06:22+00:00
prov:role displaying
qs:graphic/diagram
prov:label Line chart
prov:type linechart

```
document
  prefix userdata <http://software.dlr.de/qs/userdata/>
  prefix qs <http://software.dlr.de/qs/>
  prefix graphic <http://software.dlr.de/qs/graphic/>
  prefix app <http://software.dlr.de/qs/app/>
  prefix user <http://software.dlr.de/qs/user/>
  prefix device <http://software.dlr.de/qs/device/>
  prefix method <http://www.java.com>

  wasGeneratedBy(qs:graphic/diagram, method:visualize, 2016-12-01T16:06:22+00:00, [prov:role="displaying"])
  activity(method:visualize, 2016-12-01T16:06:21+00:00, 2016-12-01T16:06:22+00:00)
  entity(qs:graphic/diagram, [prov:type="linechart", prov:label="Line chart"])
  entity(userdata:steps, [prov:type="steps", prov:label="Steps database"])
  agent(qs:user/regina@example.org, [prov:type="prov:Person", prov:label="Regina Struminski"])
  agent(qs:app/stepcounter, [prov:type="prov:SoftwareAgent", qs:device="smartphone", prov:label="StepCounter"])
  wasAttributedTo(qs:graphic/diagram, qs:user/regina@example.org)
  wasAttributedTo(userdata:steps, qs:user/regina@example.org)
  actedOnBehalfOf(qs:app/stepcounter, qs:user/regina@example.org, -)
  used(method:visualize, userdata:steps, 2016-12-01T16:06:21+00:00)
  wasDerivedFrom(qs:graphic/diagram, userdata:steps, -, -, -)
  wasAssociatedWith(method:visualize, qs:app/stepcounter, -)
endDocument
```

Standard Graph Visualizations and Textual Representations of Provenance Data are not Easy to Understand by Non-experts

Idea: Provenance Visualization Using Comics

Provenance Comics

- Presenting the provenance of processes in visual representation that people can understand without prior instructions or training (“Provenance for people”)
- Assumption
  - People are familiar with comics from every day life
  - See *daily strips* in newspapers etc.

Provenance Comics

Mapping provenance graph to comics

- We generate a comic strip for each basic activity in the provenance graph
- Each strip consists of a varying number of panels, which are small drawings that provide further details about the activity
- The complete set of comic strips shows the “story” of the data

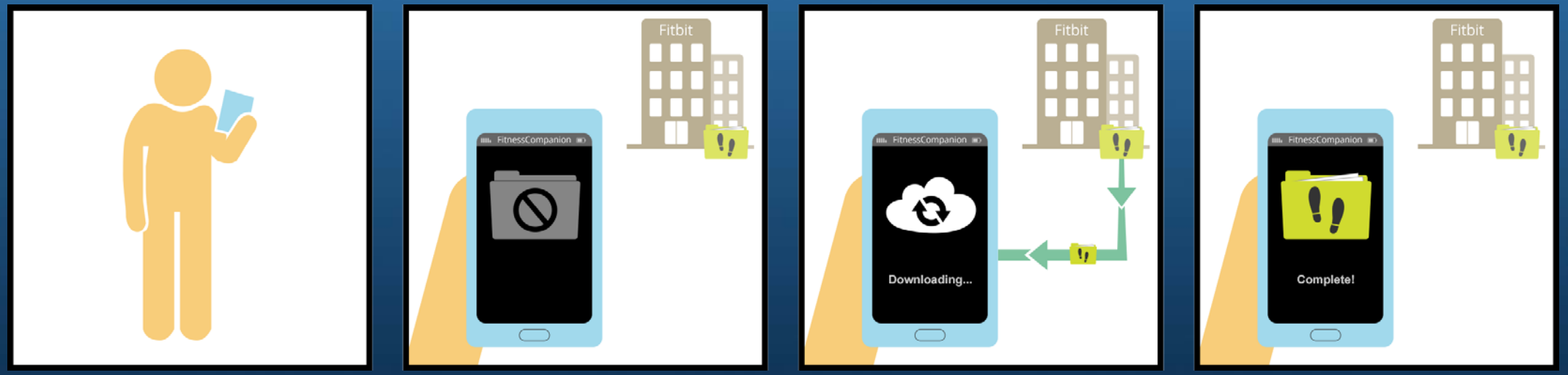
References

- Schreiber, A., Struminski, R. (2017) ***Visualizing Provenance using Comics*.** In: 9th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2017), Seattle, WA, USENIX Association
- Schreiber, A., Struminski, R. (2017) ***Tracing Personal Data Using Comics***. In: Universal Access in Human-Computer Interaction. Design and Development Approaches and Methods: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9--14, 2017, Proceedings, Part I, Springer, 444-455

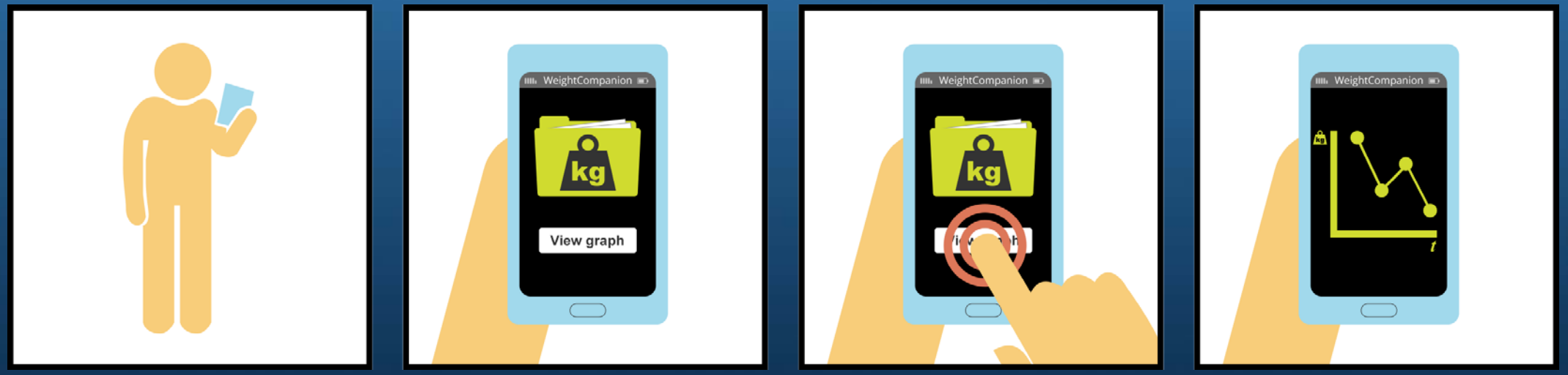
Graphical Style

Single Comic Strip Shows a Single Data-related Action

Communicate to People Where Data is Stored

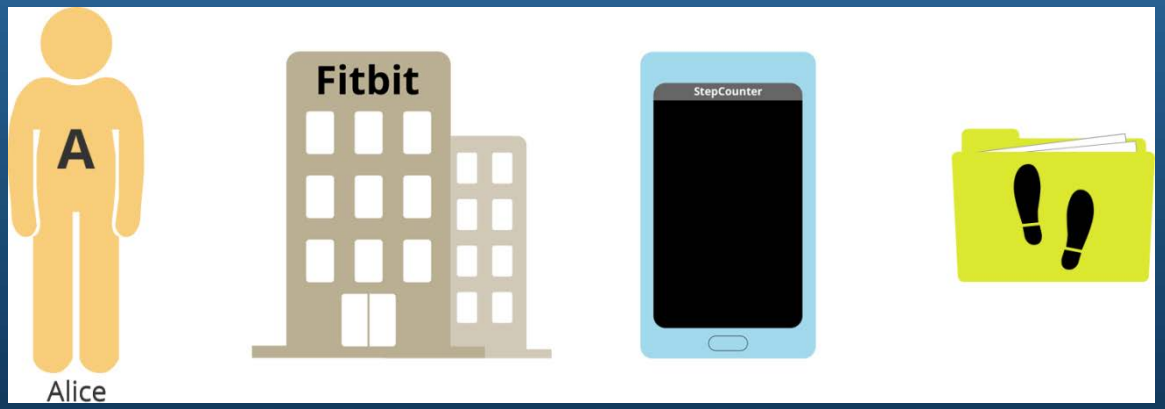


Understand How Data is Analyzed

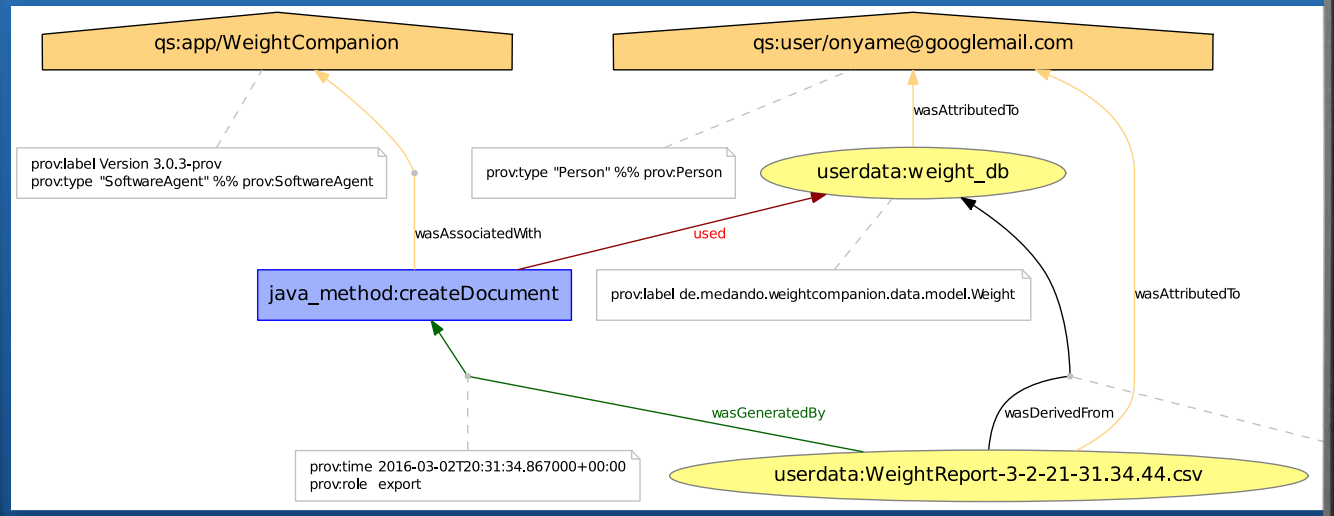


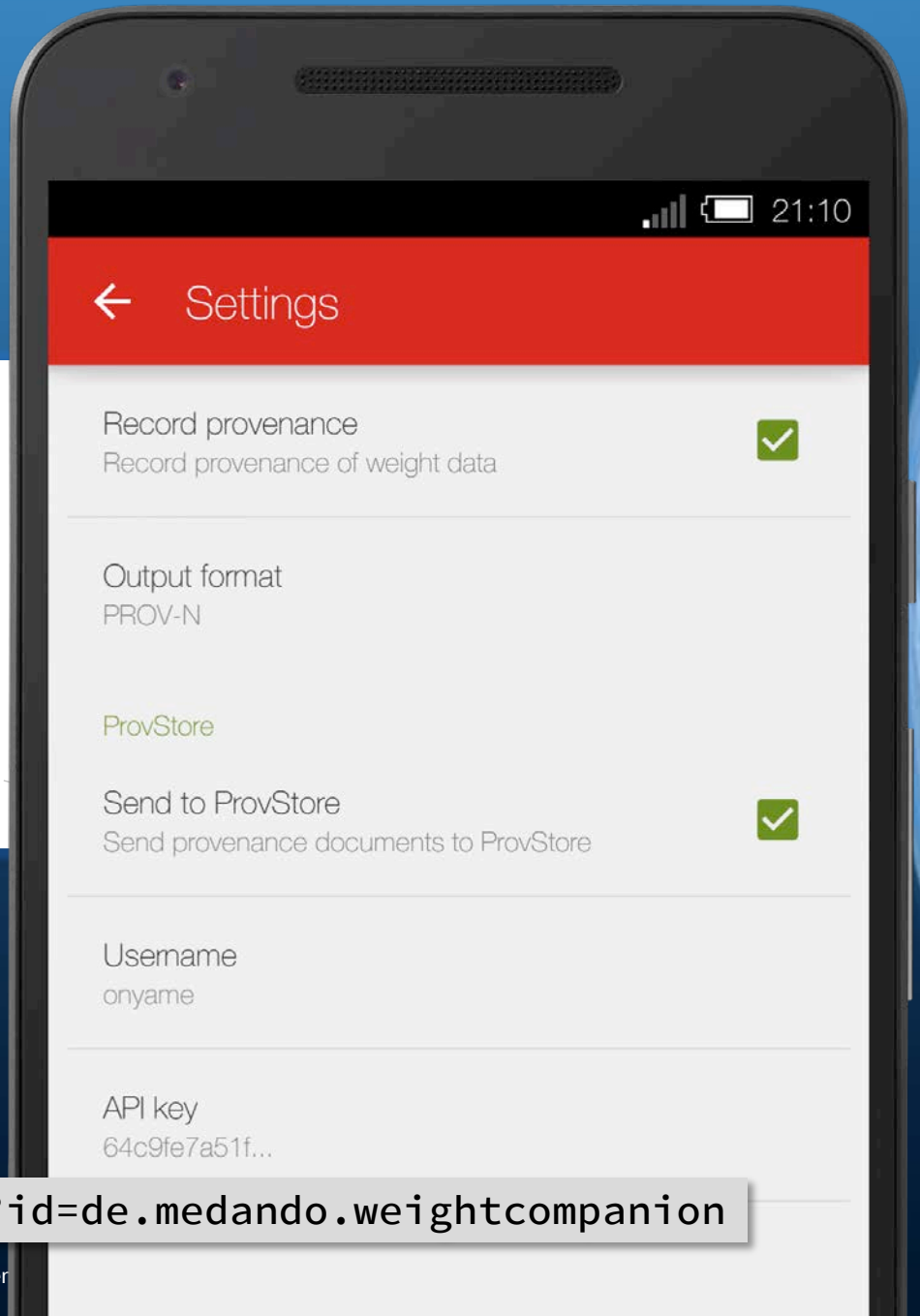
Distinctive Features

- Shapes
- Colors
- Icons
- Letters
- Labels



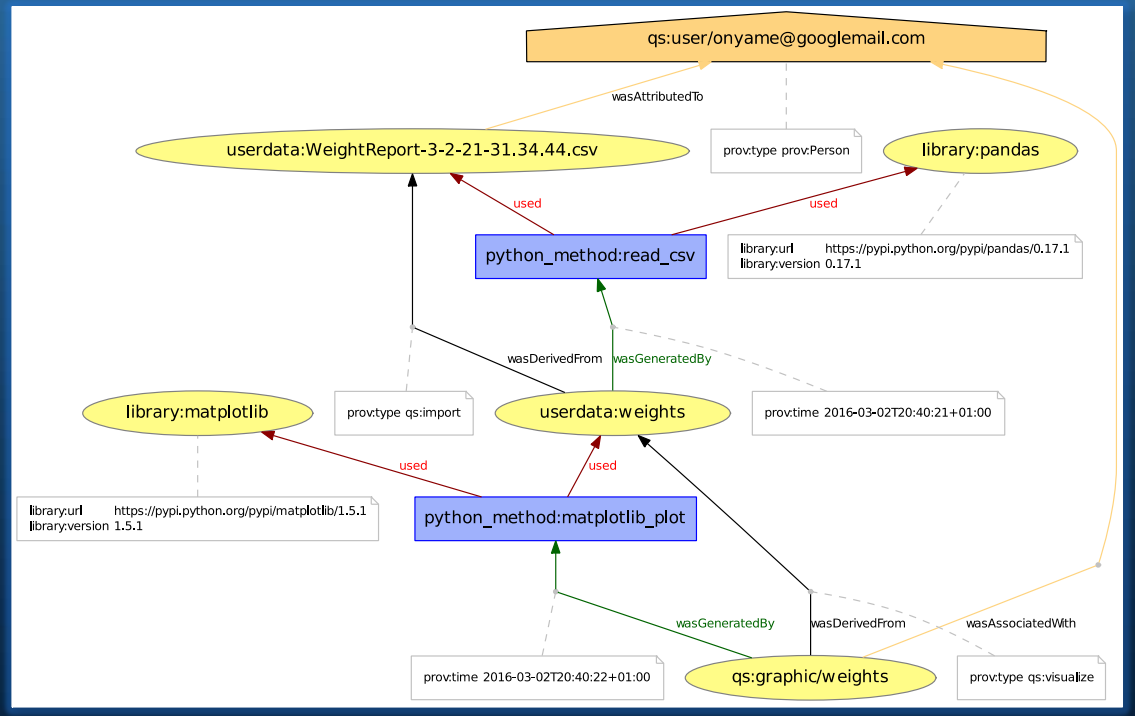
Collecting QS Provenance
Weight Tracking App



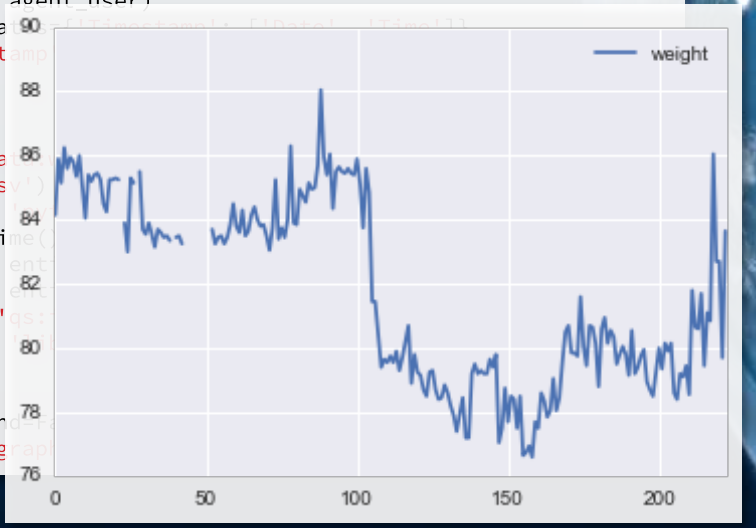


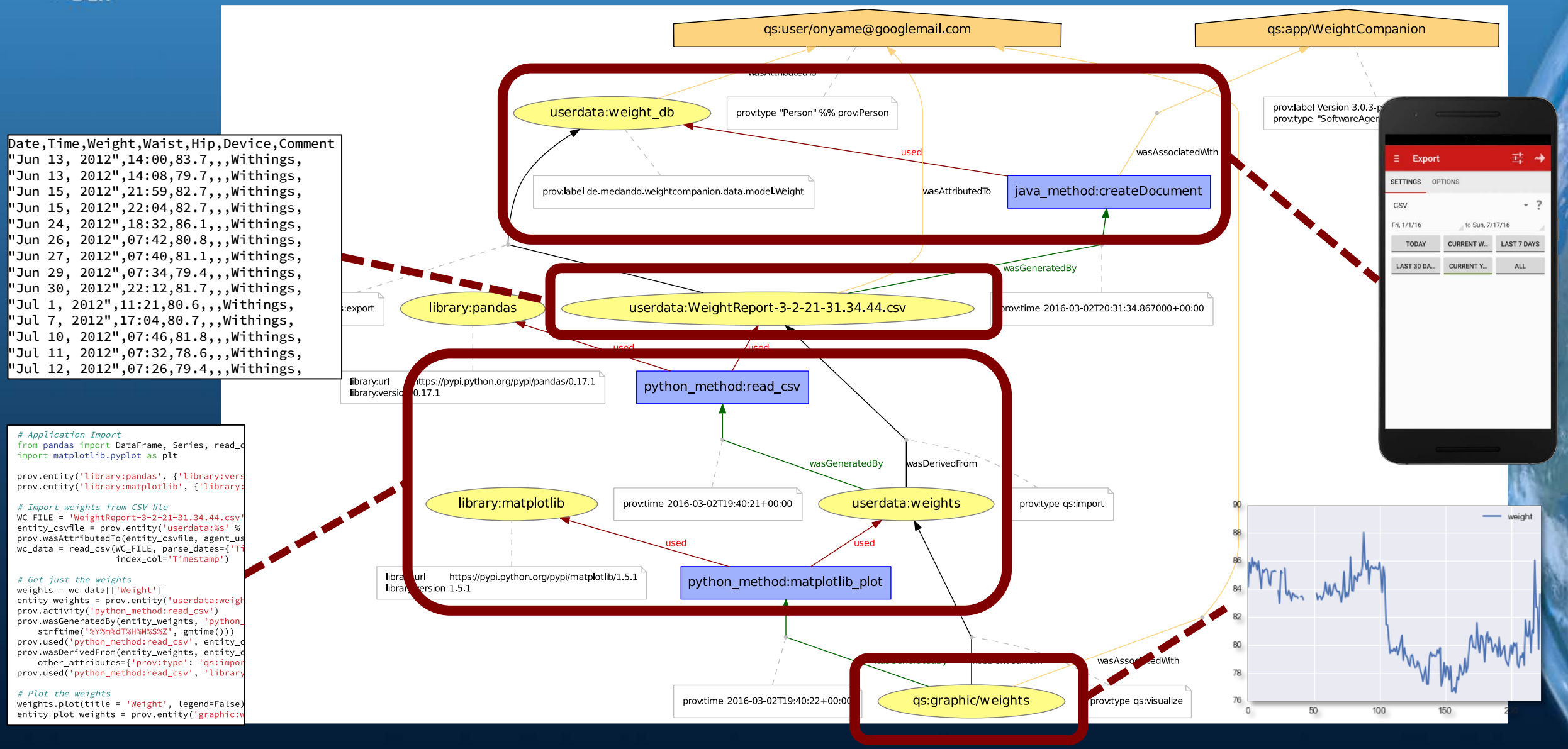
https://play.google.com/store/apps/details?id=de.medando.weightcompanion

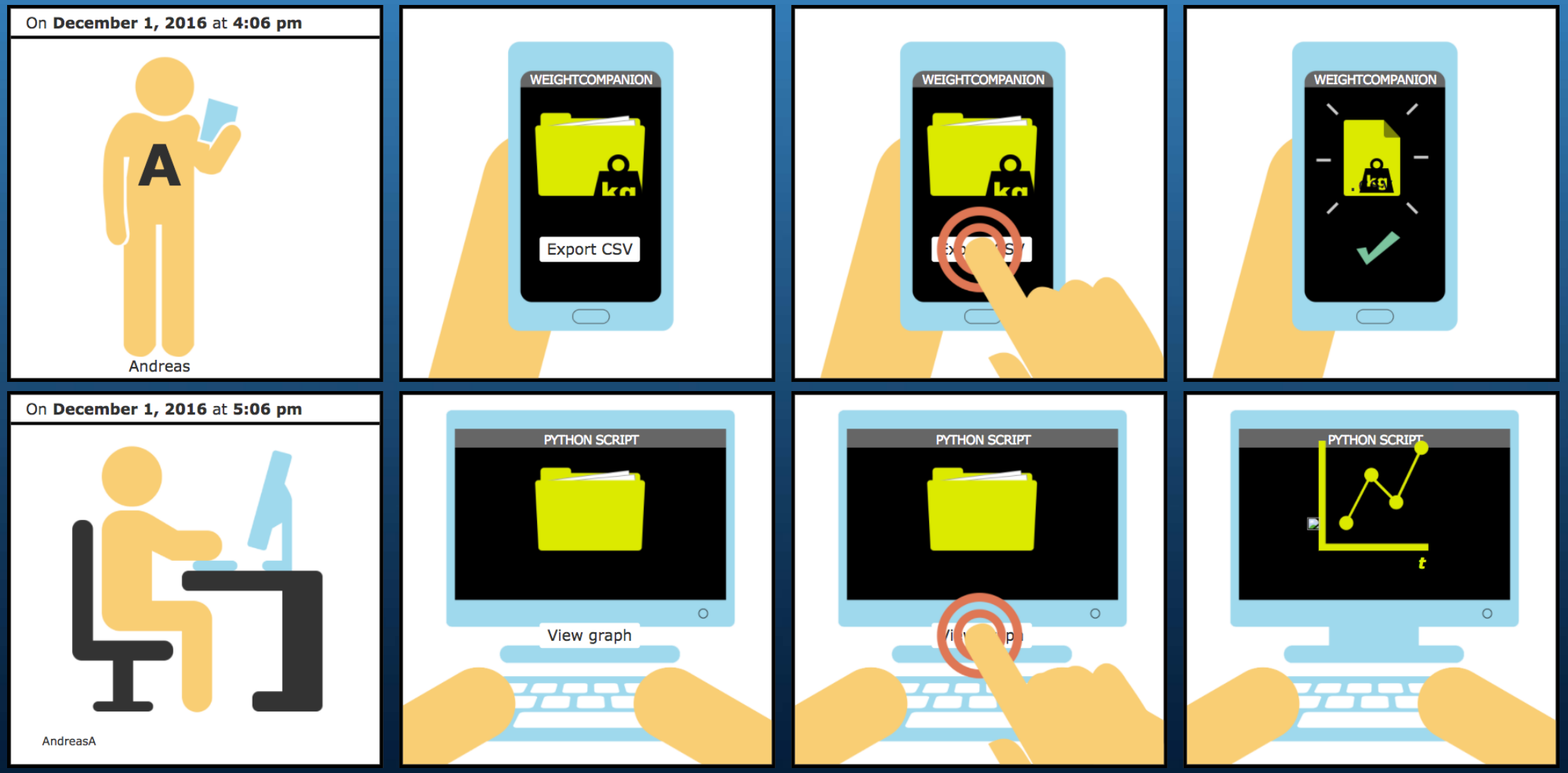
Collecting QS Provenance
Visualization with Python Script




```
# 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/qs/userdata/')
prov.add_namespace('user', 'http://software.dlr.de/qs/user/')
prov.add_namespace('graphic', 'http://software.dlr.de/qs/graphic/')
prov.add_namespace('library', 'https://pypi.python.org/pypi/')
prov.add_namespace('python_method', 'http://www.python.org')

# The user
agent_user = prov.agent('user:onyame@googlemail.com', {'prov:type': PROV['Person']})

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

prov.entity('library:pandas', {'library:version': pd.__version__})
prov.entity('library:matplotlib', {'library:version': matplotlib.__version__})

# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:%s' % WC_FILE)
prov.wasAttributedTo(entity_csvfile, agent_user)
wc_data = read_csv(WC_FILE, parse_da
                   index_col='Timest

# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userda
prov.activity('python_method:read_cs
prov.wasGeneratedBy(entity_weights,
    strftime('%Y%m%dT%H%M%S%Z', gmti
prov.used('python_method:read_csv',
prov.wasDerivedFrom(entity_weights,
    other_attributes={'prov:type': '
prov.used('python_method:read_csv',

# Plot the weights
weights.plot(title = 'Weight', leger
entity_plot_weights = prov.entity('g
```



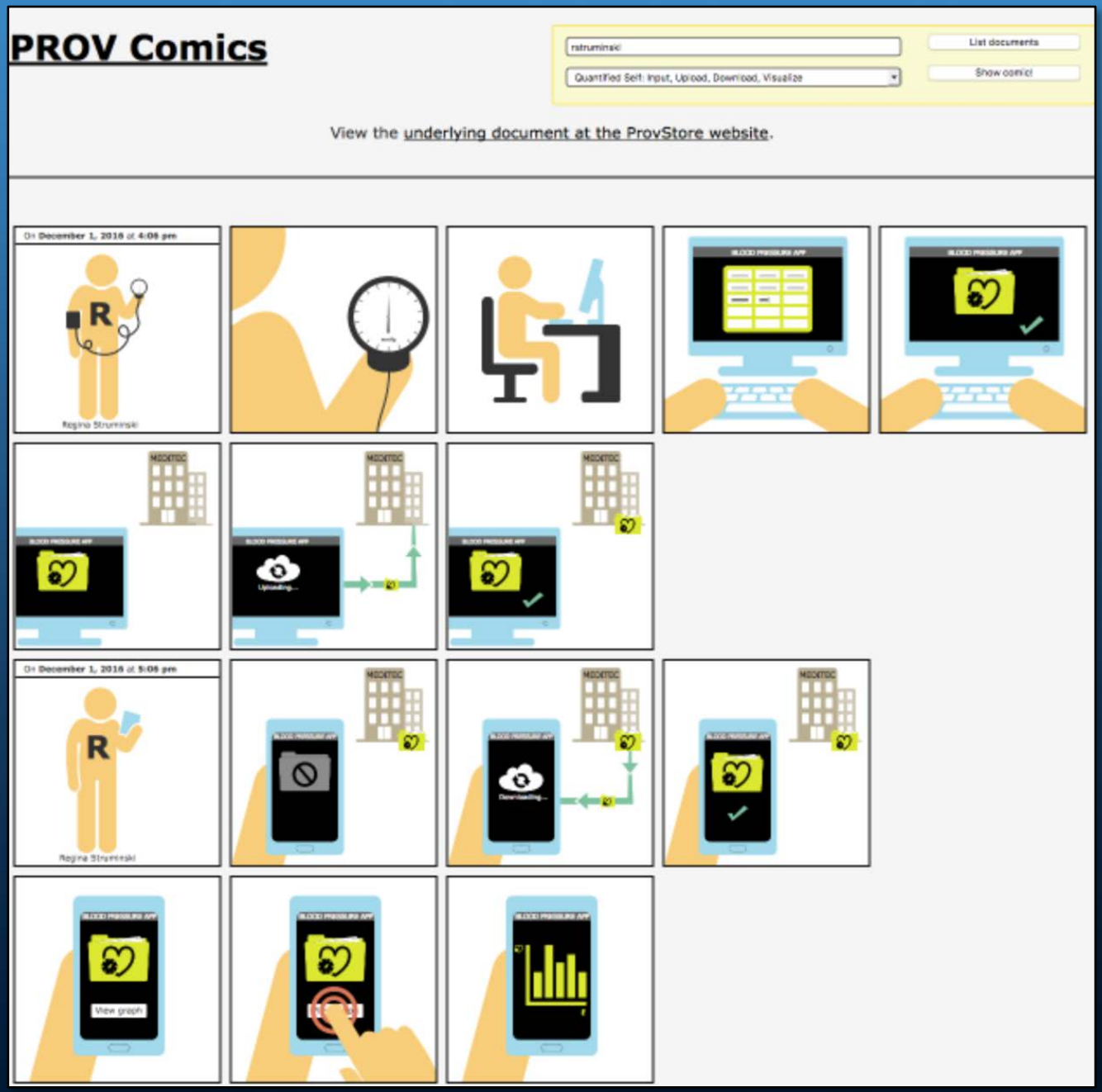

Date,Time,Weight,Waist,Hip,Device,Comment
"Jun 13, 2012",14:00,83.7,,,Withings,
"Jun 13, 2012",14:08,79.7,,,Withings,
"Jun 15, 2012",21:59,82.7,,,Withings,
"Jun 15, 2012",22:04,82.7,,,Withings,
"Jun 24, 2012",18:32,86.1,,,Withings,
"Jun 26, 2012",07:42,80.8,,,Withings,
"Jun 27, 2012",07:40,81.1,,,Withings,
"Jun 29, 2012",07:34,79.4,,,Withings,
"Jun 30, 2012",22:12,81.7,,,Withings,
"Jul 1, 2012",11:21,80.6,,,Withings,
"Jul 7, 2012",17:04,80.7,,,Withings,
"Jul 10, 2012",07:46,81.8,,,Withings,
"Jul 11, 2012",07:32,78.6,,,Withings,
"Jul 12, 2012",07:26,79.4,,,Withings,
qs:user/onyame@googlemail.com
qs:app/WeightCompanion
userdata:weight_db
prov:type "Person" %% prov:Person
prov:label de.medando.weightcompanion.data.model.Weight
used
wasAssociatedWith
wasAttributedTo
java_method:createDocument
prov:label Version 3.0.3-p
prov:type "SoftwareAgen
Export
SETTINGS
OPTIONS
CSV
Fri, 1/1/16
to Sun, 7/17/16
TODAY
CURRENT W...
LAST 7 DAYS
LAST 30 DA...
CURRENT Y...
ALL
library:pandas
userdata:WeightReport-3-2-21-31.34.44.csv
wasGeneratedBy
prov:time 2016-03-02T20:31:34.867000+00:00
used
library:url https://pypi.python.org/pypi/pandas/0.17.1
python_method:read_csv
wasGeneratedBy
wasDerivedFrom
library:matplotlib
prov:time 2016-03-02T19:40:21+00:00
userdata:weights
prov:type qs:import
used
used
library:url https://pypi.python.org/pypi/matplotlib/1.5.1
library:version 1.5.1
python_method:matplotlib_plot
prov:time 2016-03-02T19:40:22+00:00
qs:graphic/weights
wasAssociatedWith
prov:type qs:visualize
# Application Import
from pandas import DataFrame, Series, read_c
import matplotlib.pyplot as plt
prov.entity('library:pandas', {'library:vers
prov.entity('library:matplotlib', {'library:
# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:%s' %
prov.wasAttributedTo(entity_csvfile, agent_us
wc_data = read_csv(WC_FILE, parse_dates={'Ti
index_col='Timestamp')
# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userdata:weigh
prov.activity('python_method:read_csv')
prov.wasGeneratedBy(entity_weights, 'python_
strftime('%Y%m%dT%H%M%S%Z', gmtime()))
prov.used('python_method:read_csv', entity_c
prov.wasDerivedFrom(entity_weights, entity_c
other_attributes={'prov:type': 'qs:impor
prov.used('python_method:read_csv', 'library
# Plot the weights
weights.plot(title = 'Weight', legend=False)
entity_plot_weights = prov.entity('graphic:w
weight
90
88
86
84
82
80
78
76
0
50
100
150


On December 1, 2016 at 4:06 pm
A
Andreas
WEIGHTCOMPANION
Export CSV
WEIGHTCOMPANION
WEIGHTCOMPANION
On December 1, 2016 at 5:06 pm
AndreasA
PYTHON SCRIPT
View graph
PYTHON SCRIPT
PYTHON SCRIPT
t

PROV Comics
Web Application

`http://provcomics.de`

- Implemented in JavaScript
- Single page website
- Reads provenance graph from ProvStore
- Uses ProvStore jQuery API
- Code: `http://github.com/DLR-SC/prov-comics`



Future Work and Use Cases

Future Work

- Different comic styles
  - Comparative user studies
- Quantitative comics
  - Geographical information
  - Glyph-based depiction
- Technical improvements
  - Large Provenance graphs
  - Provenance templates
  - “Intelligent” generation of pictures

Possible Use Cases

- Journalism
- Generation of handbooks
- Communicating security incidents
  - Smart Home

Wang, Q.; Hassan, W.U.; Bates, A.; Gunter, C. (2017) ***Provenance Tracing in the Internet of Things***; TaPP’17, USENIX Association: Seattle, WA

Thank You!

Andreas Schreiber

www.DLR.de/sc/ivs

andreas.schreiber@dlr.de

@onyame