

# Hinweis

Die Original-Präsentation basiert auf dem HTML-Präsentations-Framework reveal.js.

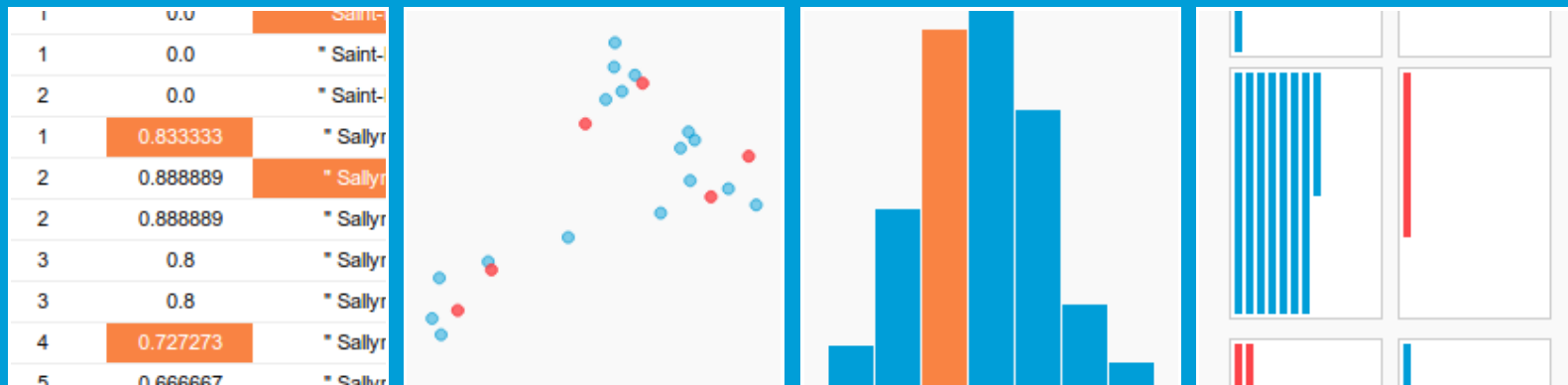
Bei diesem PDF-Dokument handelt es sich um eine mit Qualitätseinbußen konvertierte Datei.

# Implementation of a Browser-based Interface for Provenance Analysis

## Bachelor Thesis

Janek Bettinger

2014/10/24



# Data Provenance

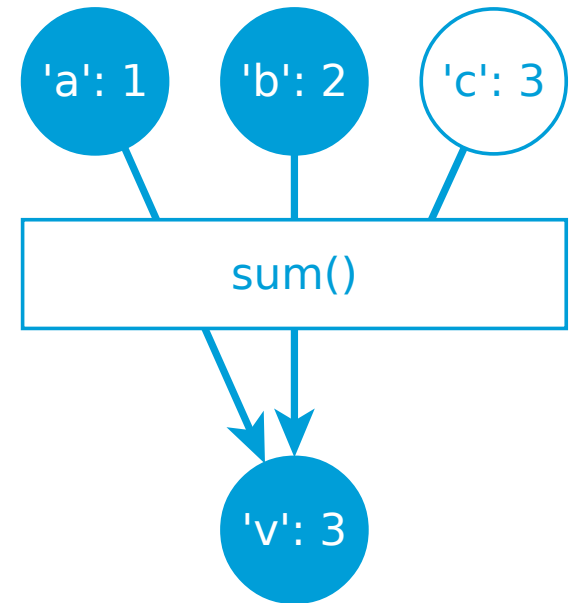
derivation history of generated data

# Provenance Analysis Tool

computes provenance relationships of Python programs

# Sample

```
def sum(arg):  
    res = []  
    for d in arg:  
        res.append({  
            'v': d['a'] + d['b']  
        })  
    return res  
  
data = [ {'a': 1, 'b': 2, 'c': 3} ]  
summed = sum(data)    # [ {'v': 3} ]
```



# Analysis Results

in the relational data format (→ PostgreSQL)

id	callid	varname	containerid	subscript	value	atomic	varid
28	3		2	5		f	2
29	3		28	article_score	0.351335	t	2
30	3		28	author_exp	1	t	2
31	3		28	author_score	0.0	t	2
32	3		28	author_name	" -jha- "	t	2
33	3		2	6		f	2
34	3		33	article_score	0.332234	t	2
35	3		33	author_exp	2	t	2
36	3		33	author_score	0.142857	t	2
37	3		33	author_name	" -jha- "	t	2

id	argumentvalueid	returnvalueid
198043	28	219342
372586	28	219343
198045	29	219342
372585	29	219347
372588	29	219343
372590	29	219346
372591	29	219344
372595	29	219345
372593	30	219345
198044	31	219342

# Interface

web application that illustrates:

- function calls
- processed data
- relationships

# Kinds of Visualization

Table — plain values

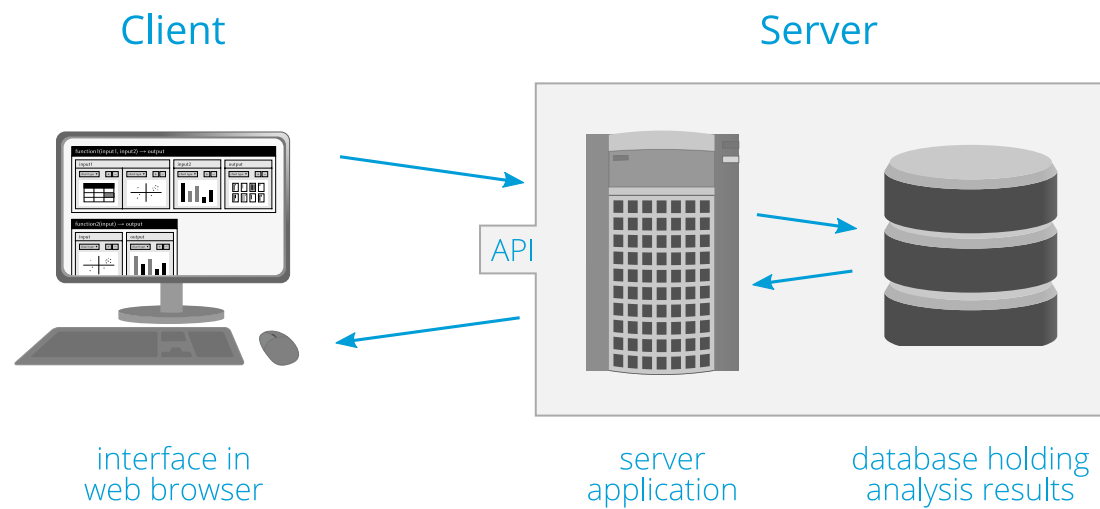
Scatter Plot — coordinate system

Histogram — distribution of values

Line Mosaic Plot — relationships between variables



# Setup



# Live Demos

- ① — table, scatter plot, histogram
- ② — line mosaic plot

# Server

## languages:

**Python** — application

PostgreSQL — database access

JSON — data exchange

## Python framework:

Bottle — web server & API

# API

`/api/calls` — function calls incl. parameters

`/api/values` — values and IDs of related ones

# Client

## languages:

**JavaScript** — behaviour

HTML — content

CSS — presentation

SVG — graphic

## JavaScript libraries:

jQuery — manipulation

RequireJS — modules

Handlebars — templates

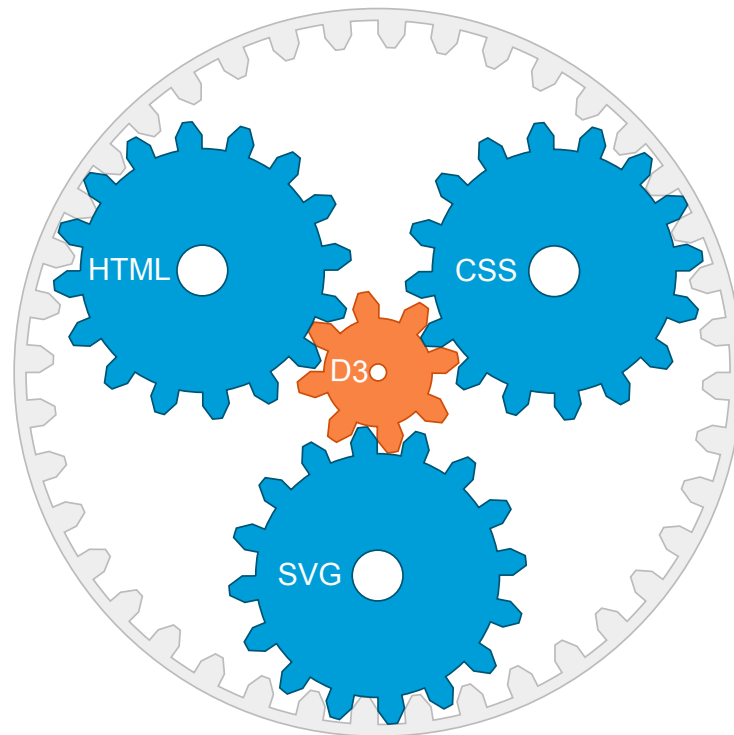
...

# DataTables

lazy loading of (large) tables

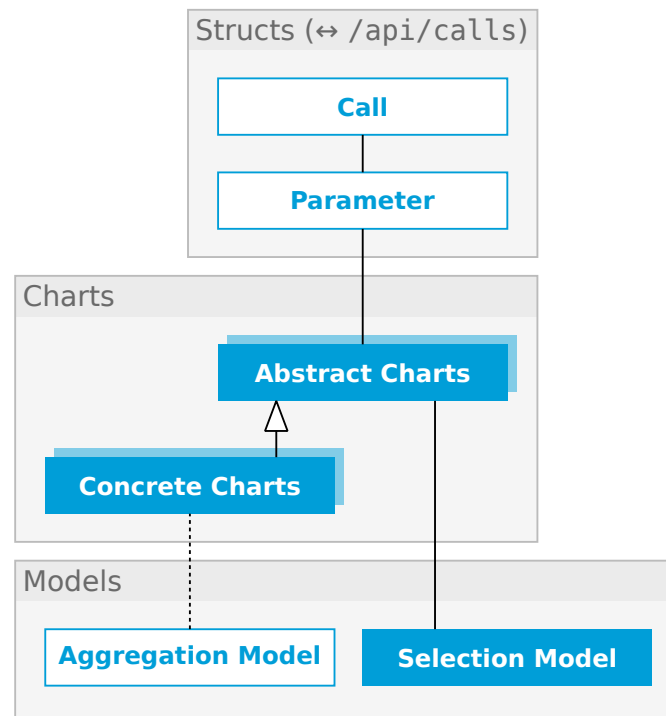
# Data-Driven Documents

— D3.js —



(based on work by Mike Bostock)

# Structure



(simplified)

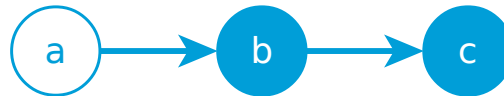
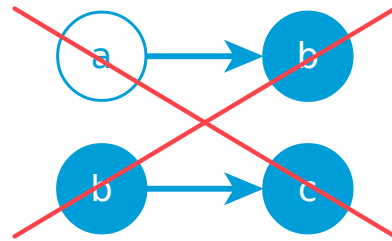


major challenge:

**Performance**

# Database

pre-calculation of relationship chains



# Further Measures

- caching — spare calculations & lookups
- data structures — e. g. sets instead of arrays
- statements — efficient and native ones

# Conclusion

implementation has been **successful**

*... of course, software is never finished*

*“ A picture is worth a thousand words.  
An interface is worth a thousand pictures.”*

— Ben Shneiderman (expert on human-computer interaction, 2003)