TextFlows: web based text mining platform

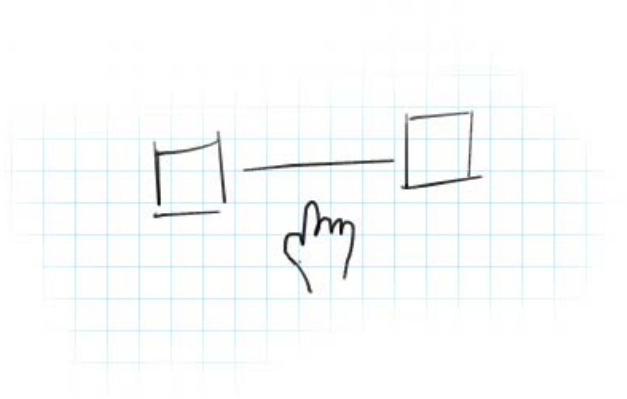
Authors: Matic Perovšek, Janez Kranj, Nada Lavrač et al.
Presentation by: Matej Martinc, Matic Perovšek, Senja Pollak

Textflows = fork of ClowdFlows, which is

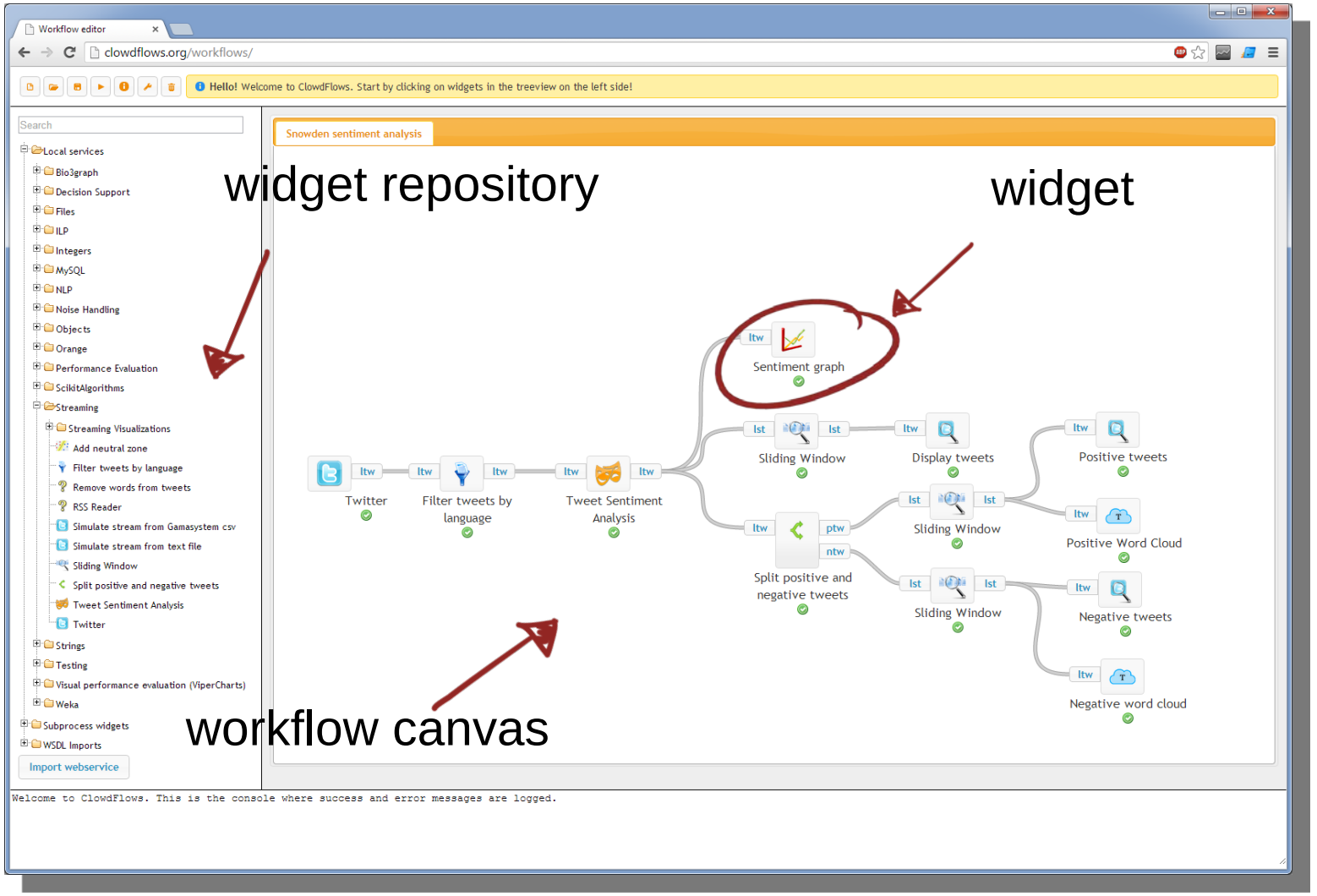
- A platform for:
  - composition,
  - execution,
  - and sharing of **interactive data mining workflows**
- Most important features:
  - A web based user interface for building workflows
  - Cloud-based architecture, service-oriented architecture
  - Big roster of workflow components
  - Visual programming paradigm
  - Open source
- Publicly available at clowdflows.org, source code available at https://github.com/xflows/clowdflows under MIT license

Building scientific workflows

- consists of simple operations on workflow elements
  - drag
  - drop
  - connect
- suitable for non-experts
- good for representing complex procedures
- allow users to publicly upload their workflows so that they are available to a wider audience, perfect for experiment replication



The user interface

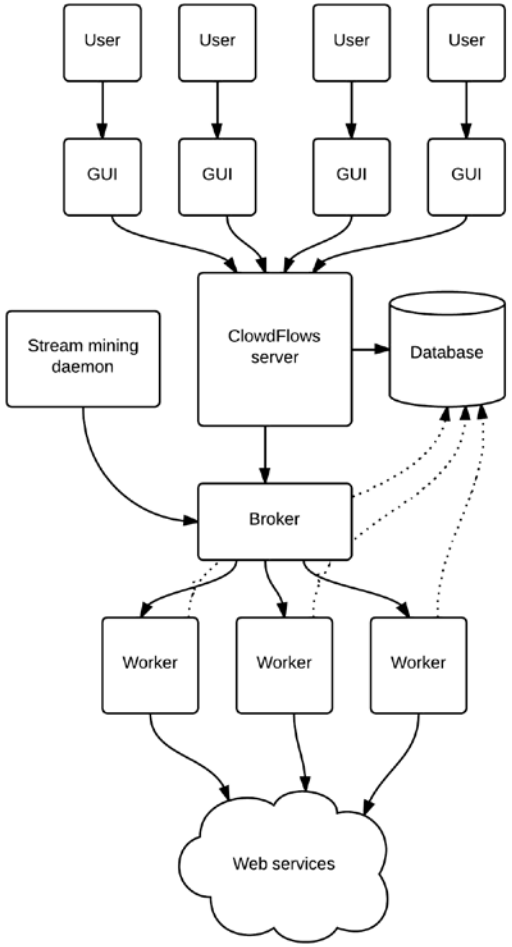


ClowdFlows platform

- Large algorithm repository:
  - Relational data mining,
  - Algorithms from other platforms such as Orange and WEKA
  - Data and results visualization
  - Social network analysis
  - Analysis of big data streams
  - Not many widgets for text mining
- Large workflow repository:
  - Enables access to our technology heritage

The architecture

- GUI
  - User constructs workflows by connecting widgets on the canvas
- ClowdFlows server
  - Serves the GUI, stores all changes to the database, emits tasks to execute widgets to the broker
- The broker
  - Delegates the tasks to workers.
- The workers
  - Headless instances of the ClowdFlows server (they do not serve the user interface)



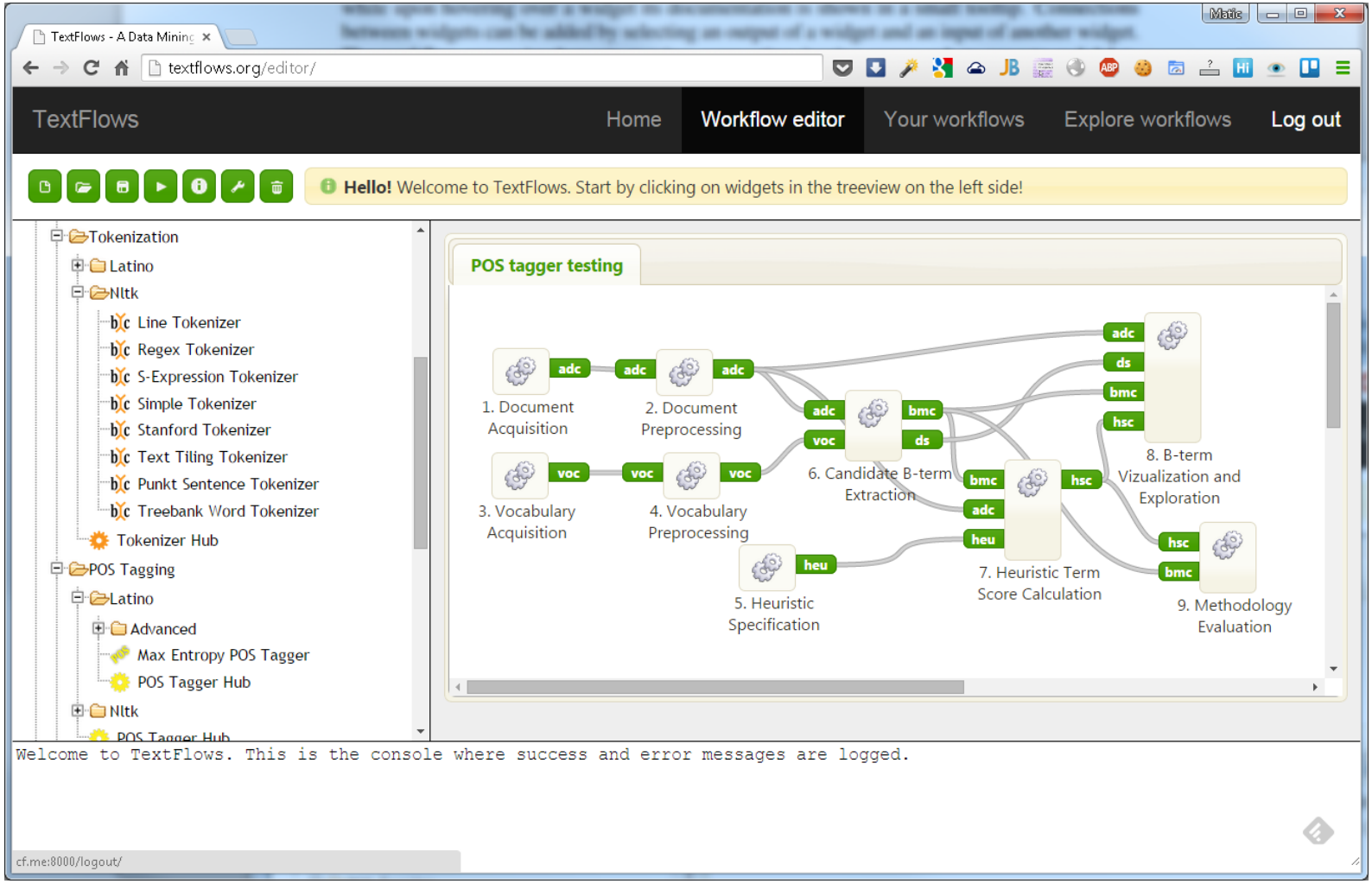
TextFlows

- TextFlows platform is a:
  - Fork of ClowdFlows
  - Specialized for the field of text analytics
- Widgets for:
  - Text preprocessing
  - Text categorisation
  - Literature based discovery
  - Relational data mining through wordification
  - And other
- Publicly available at textflows.org, source code available at https://github.com/xflows/textflows under MIT license

Comparison with ClowdFlows

- ClowdFlows:
  - Roster of not fully compatible widgets, developed separately by each workflow developer, non-systematic approach
  - Missing components for text mining and natural language processing
- TextFlows:
  - Includes numerous text mining and NLP widgets
  - Widgets grouped by their functionality
  - New common text representation structure

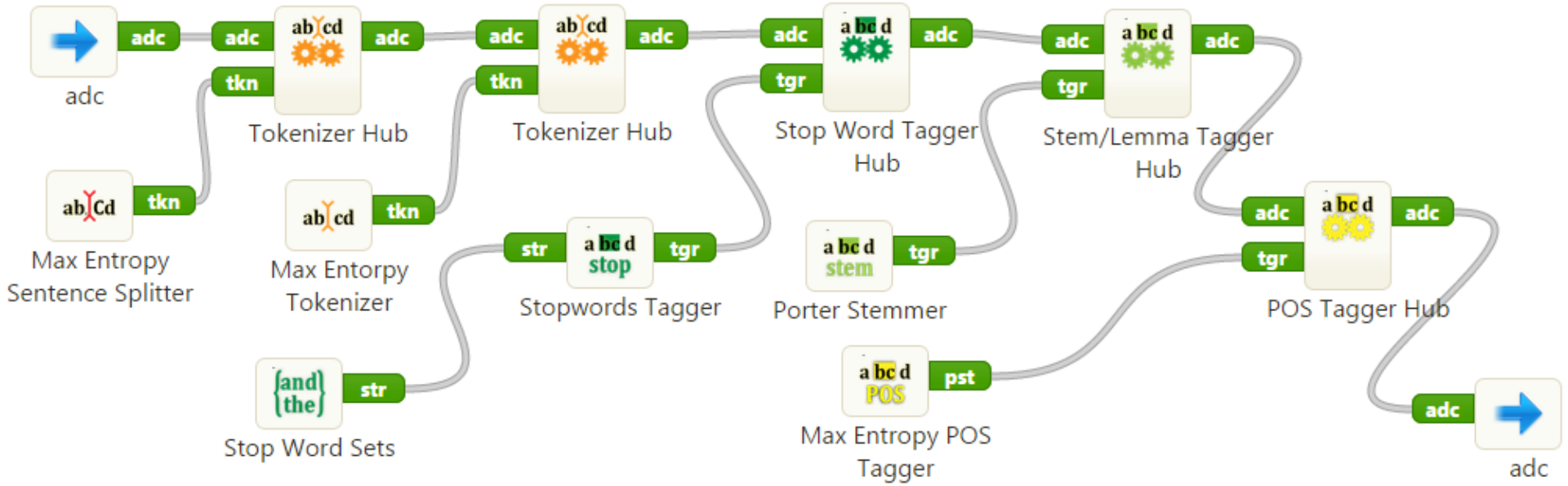
The user interface



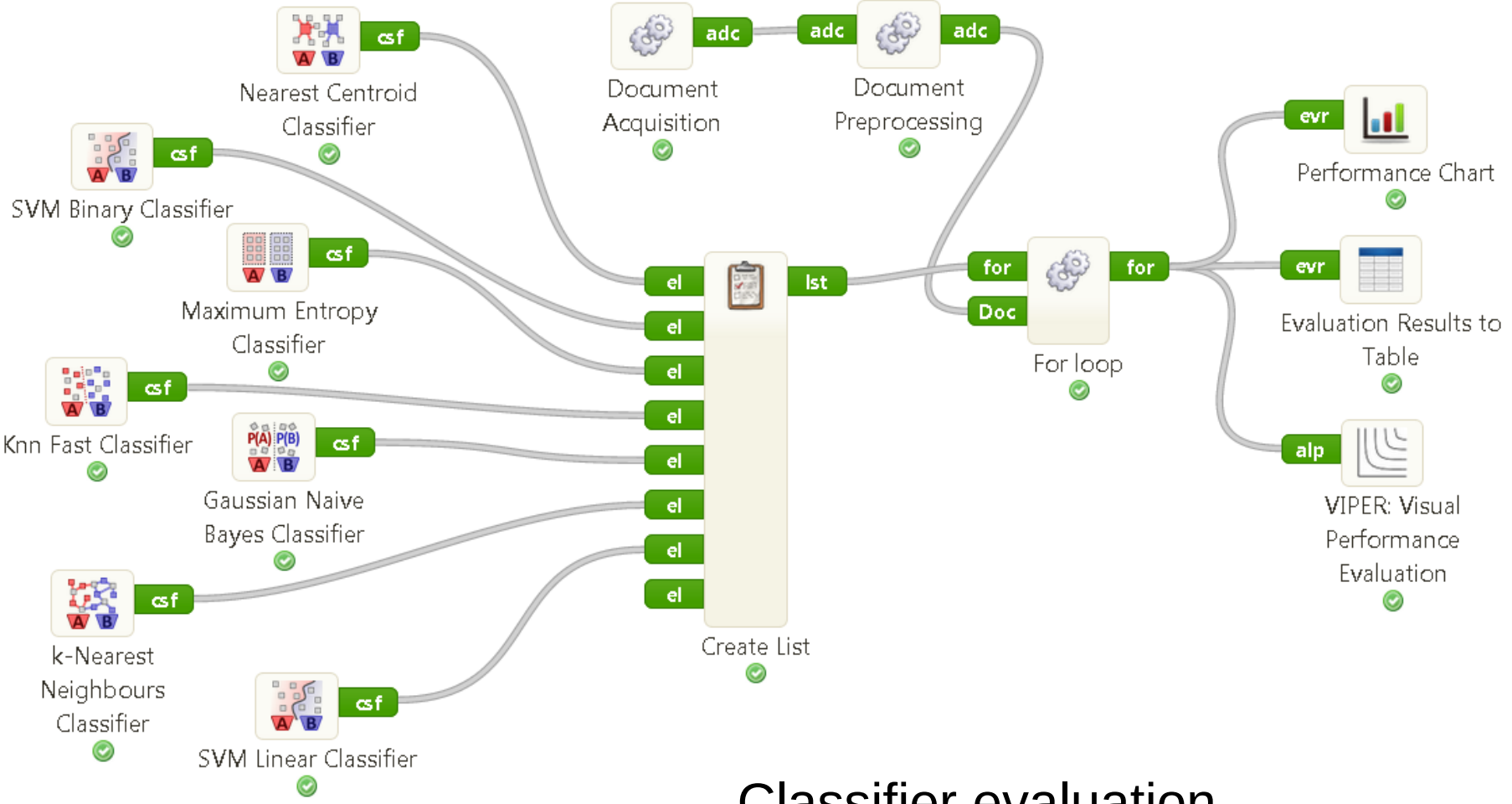
Widget types

- Regular widgets (tokenization, POS tagging, lemmatization, classification...)
- Visualization widgets for data visualization
- Interactive widgets
- Workflow control widgets (subprocesses, iteration through data)

Document Preprocessing



Classifier Evaluation workflow



Classifier evaluation
http://textflows.org/workflow/350/

Classifier Evaluation Results

Classifier evaluation
http://textflows.org/workflow/350/

Input and output formats

- No standardized I/O formats are imposed
- Unified format for corpora representation (AnnotatedDocumentCorpus Python class)
- AnnotatedDocumentCorpus(ADC) contains:
  - Collection of documents (AnnotatedDocument instances)
  - Features with additional info

Input and output formats cont.

- AnnotatedDocument instance contains:
  1. Text of the document
  2. Features with additional info about a single document
  3. Collection of annotation instances
- Annotation instance:
  1. Used to mark part of the document
  2. Pointers to the start and end of the annotation
  3. Type attribute for annotation grouping
  4. Features used by various taggers

Corpus aquisition

- Varius widgets for loading document corpora, labeling of documents with domain labels and converting them to ADC
- Multiple aquisition scenarios are supported:
  - Loading locally stored files in various formats
  - WSDL+SOAP web services
  - Selecting documents from SQL databases
  - Crawling the internet for gathering documents
  - Snippets returned from web search engines

ClowdFlows 2.0

- Addresses many current issues – for users and for devs
- Sometime in 2017
- Reintegration of TextFlows into ClowdFlows
- UX improvements:
  - Widget recommendation system based on the existing database of workflows
  - Faster workflow execution and scalability:
    - Optimized reads/writes of intermediate results
  - Improved error reporting
  - Integrated documentation

ClowdFlows 2.0 cont.

- Improvements for developers
  - ClowdFlows core will be completely separated from its widgets
    - modularity
    - Widget packages, e.g.: data_mining, weka, tf_core
    - We can focus on developing the core
  - Separation of front-end back-end
    - We implemented a ClowdFlows REST API
    - Front-end re-written in Angular 2 that consumes the API
    - Allows developers to reuse the UI for new backends, just by implementing the specified API endpoints
    - OR to consume the API for a new UI or even call the API programmatically from scripts

Thank you for your attention!

matej.martinc@ijs.si

REST service integration

- Create widget in Django admin
  - Add widgets attributes (name, action, description, package, category, input and output variable)
- Wrap REST service in a Python function
  - Example call to web service for sentiment analysis http://kt.ijs.si/MartinZnidarsic/webservices/sentana/sentana.php?sentence=What+a+lovely+day

```
def call_sentana(input_dict):
    import urllib2
    import json
    somesentence = input_dict['inp1']  # our only input is in input_dict['inp1'] , notice the variable name 'inp1'
    somesentence = somesentence.replace (" ", "+")
    url = 'http://kt.ijs.si/MartinZnidarsic/webservices/sentana/sentana.php?sentence=' + somesentence
    response = urllib2.urlopen(url).read()
    jsondata = json.loads(response)
    result = jsondata['data']['sentimentscore']
    output_dict = {}
    output_dict['out1'] = result  # result is put in the only output denoted with output_dict['out1']
    return output_dict
```

- Export the widget: python manage.py export_package -u workflows/$your_package_name$/db/package_data.json $your_package_name$