

# Real-Time Recommendation of Streamed Data

*Tutorial at ACM RecSys'15, Vienna, Austria*

Frank Hopfgartner, Benjamin Kille, Tobias Heintz, Roberto Turrin

# Overview

- Evaluating Recommender Systems
  - Offline vs. Online Evaluation
  - Case Study: CLEF NewsREEL
- Introduction to Stream Based Recommendation
  - Streamed Data vs. static sets
  - Challenges
  - Popular stream based recommendation algorithms
- Idomaar
- Open Recommendation Platform

# Why evaluate?

Amazon.com: Recommended for You

www.amazon.com/gp/yourstore/recs/ref=sv\_ys\_1

Apple Yahoo! Google Maps YouTube Wikipedia News Popular

amazon Try Prime

Shop by Department Search All

Frank's Amazon.com Today's Deals Gift Cards Sell Help

Your Amazon.com Your Browsing History Recommended For You Improve Your Recommendations

Your Amazon.com > Recommended for You  
(If you're not Frank Hopfgartner, click here.)

Just For Today

Browse Recommended

Recommendations

Amazon Instant Video

Amazon MP3 Store

Appliances

Appstore for Android

Arts, Crafts & Sewing

Automotive

Baby

Books

Books on Kindle

Camera & Photo

Cell Phones & Accessories

Clothing & Accessories

Computers

Electronics

Grocery & Gourmet Food

Home & Kitchen

Home Improvement

Kindle eBooks

These recommendations are based on items you own and view: All | New Releases | Coming Soon

1. **How to Sew: Basics**  
by Various (December 16, 2010)  
Average Customer Review: ★★★★☆ (37)  
Auto-delivered wirelessly  
Kindle Price: \$1.49

I own it Not interested Rate this item

Recommended because you purchased Big Data Now: Current Perspectives from O'Reilly Radar and more (Fix this)

2. **Big Data and Hadoop**  
by WAGmob (September 18, 2013)  
Average Customer Review: ★★★★☆ (6)  
Auto-delivered wirelessly  
Kindle Price: \$0.99

I own it Not interested Rate this item

Recommended because you purchased Big Data Now: Current Perspectives from O'Reilly Radar and more (Fix this)

More results

of July

# How to evaluate?

## Offline Evaluation

- Recommendation algorithms are often benchmarked **offline** using a static dataset.

## Online Evaluation

- Industry also performs **A/B testing** to benchmark their recommendation algorithms **in operation**.

# Offline Evaluation - Dataset construction

1. Chose time point  $t_0$  to split dataset
2. Classify ratings before  $t_0$  as training set
3. Classify ratings after  $t_0$  as test set

	Hundert-wasserhaus	Belvedere Complex	Hofburg Palace	Schonbrunn Palace	Wiener Rathaus
Paul	2		4	5	2
Peter			5		3
Susan			1	2	

# Offline Evaluation - Benchmarking Recommendation Task

1. Train rating function  $f(u,i)$  using **training set**
2. Predict rating for all pairs  $(u,i)$  of **test set**
3. Compute  $RMSE(f)$  over all rating predictions

$$RMSE(f) = \sqrt{\frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} (f(u, i) - r_{ui})^2}$$

# Drawbacks of offline evaluation



User is ignored

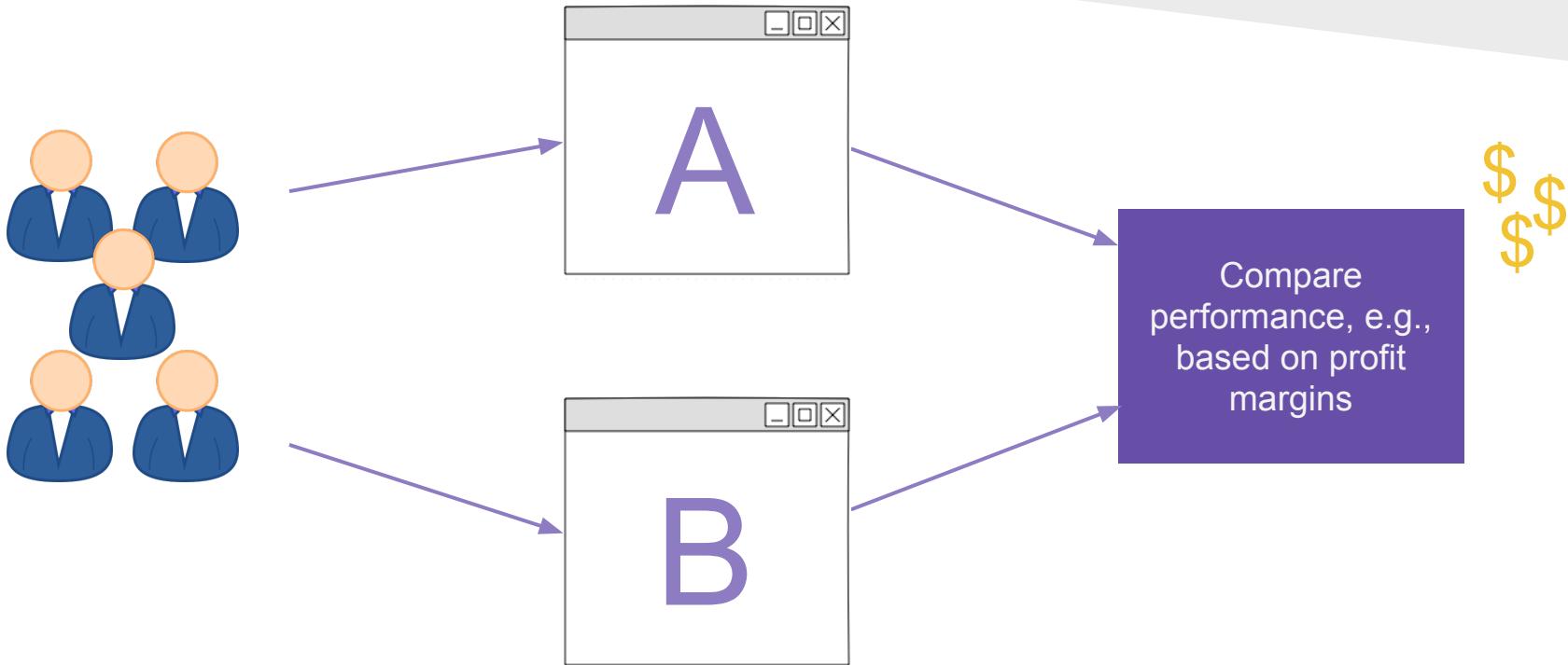
Previous interactions

Change of interest

Context

Technical challenges  
are ignored

# Online Evaluation (A/B testing)



# Drawbacks of online evaluation



Large user base and costly infrastructure required

Different evaluation metrics required

Comparison to offline evaluation challenging

# Case study: CLEF NewsREEL

In CLEF NewsREEL, participants can develop stream-based **news recommendation** algorithms and have them benchmarked (a) online by **millions of users** over the period of a few months, and (b) offline by **simulating a live stream**.

# Conferences and Labs of the Evaluation Forum (CLEF)

www.clef-initiative.eu

- Build and provide large test collections
- Foster interdisciplinary research towards grand challenges
- Build new research communities



The CLEF Initiative: Mission

**The CLEF Initiative is a self-organized body whose main mission is to promote research, innovation, and development of Information access systems with an emphasis on multilingual and multimodal Information with various levels of structure.**

- multilingual and multimodal system testing, tuning and evaluation;
- investigation of the use of unstructured, semi-structured, highly-structured, and semantically enriched data in information access;
- creation of reusable test collections for benchmarking;
- exploration of new evaluation methodologies and innovative ways of using experimental data;
- discussion of results, comparison of approaches, exchange of ideas, and transfer of knowledge.

CLEF 2013 and Onwards  
FIRE 2013, 5 December 2013, New Delhi, India

Nicola Ferro

16

# CLEF 2016

- eHealth
- ImageCLEF
- Living Labs for IR (LL4IR)
- **News Recommendation Evaluation Lab (NewsREEL)**
- Uncovering Plagiarism, Authorship and Social Software Misuse (PAN)
- Social Bookmark Search (SBS)

# NewsREEL scenario

The screenshot displays a web page with a dark header bar containing the title "NewsREEL scenario". Below the header is a large, bold, white title "NewsREEL scenario". The main content area features a news article about the BMW 6er Cabrio. To the right of the article is a sidebar with a section titled "Umwelt Wasserstoff CO<sub>2</sub> Hybrid" from "auto-motor-und-sport". Below this are sections for "Blog" and "Forum". A sidebar on the left lists recent news items. The bottom of the page contains a "Hier werben" (Here advertise) button and a "powered by plista" logo.

**BMW 6er Cabrio**

Anders als beim Vorgänger lässt das 6er Coupé diesmal den Cabrio den Vorbitz. Rechtzeitig zum Frühling kommt der um sechs Zentimeter auf 4,89 Meter gewachsene 2+2-Sitzer mit leichtem Kofferraum und einer verdeckten Motorhaube. Der 6er Cabrio ist mit Allradantrieb oder Integral-Aktivlenkung. Neben dem 3.0-Typ 650 (407 PS, 19,7 L/100 km) geht es das Sechszylinder-Modell 640 (320 PS) mit Benzin direktinspritzung, Turbo-Aufladung und Start-Stopp-Funktion, das in 5,7 Sekunden von null auf 100 km/h sprintet, aber nur 7,9 L/100 km (185 g CO<sub>2</sub>/km) verbrauchen soll. Eine Achtagung-Automatik ist jeweils serienmäßig.

**Weitere Neuheiten 2011**

- Fast 30 Neuheiten aus den USA und Italien
- Fast drei Dutzend Modell-Neuheiten
- Das sieht es Neues von Ferrari hin VW
- Helle als zwei Dutzend japanische Neuheiten
- Diese Änderungen kommen 2011

**Das könnte Sie auch interessieren**

**Daran erkennt man gute Autovermietungen**

Eine Autovermietung kann während des Urlaubs sehr nützlich sein. Doch woran erkennt man, dass dr Autovermieter seriös ist?... mehr

**VW Jetta**

Der Jetta kehrt zurück nach Europa. Der sportlicher denn je konzipierte Volkswagen soll nun das Limousinen-Spektrum komplettieren. Vergleichen Sie... mehr

**Was bringt ein schadstoffarmer PKW**

Umweltbewusstsein ist gefragt, daher setzen immer Automobilhersteller auf schadstoffarme PKW. Doch viele Verbraucher zögern beim Kauf ... mehr

**Beim Autokauf ein Schnäppchen machen**

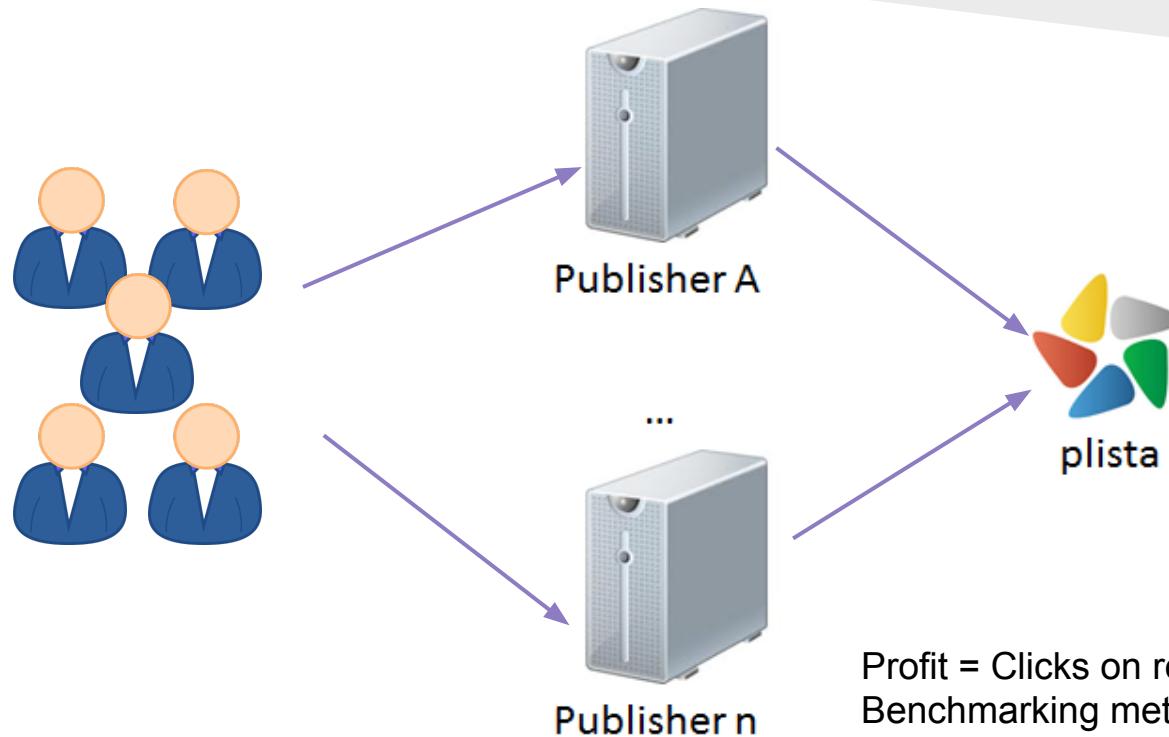
Statistisch gesehen kauft sich jeder Deutsche in seinem Leben fast 11 Autos. Da der Kauf eines Autos nicht gerade günstiges Unterfangen ist, ... mehr

**Hier werben**

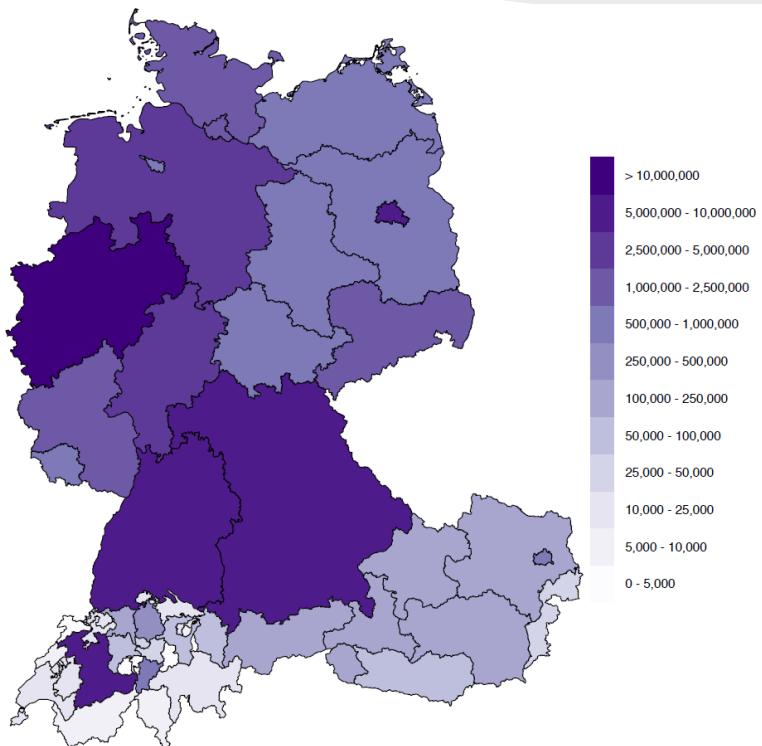
powered by plista

Image: Courtesy of T. Brodt (plista)

# NewsREEL scenario



# Who are the users?



# CLEF NewsREEL 2016

## Task 1

### Offline Evaluation

- Benchmark News Recommendations in a Simulated Environment

## Task 2

### Online Evaluation

- Benchmark News Recommendations in a Live System.

# Task 1: Offline Evaluation

## Predict interactions in a simulated data stream

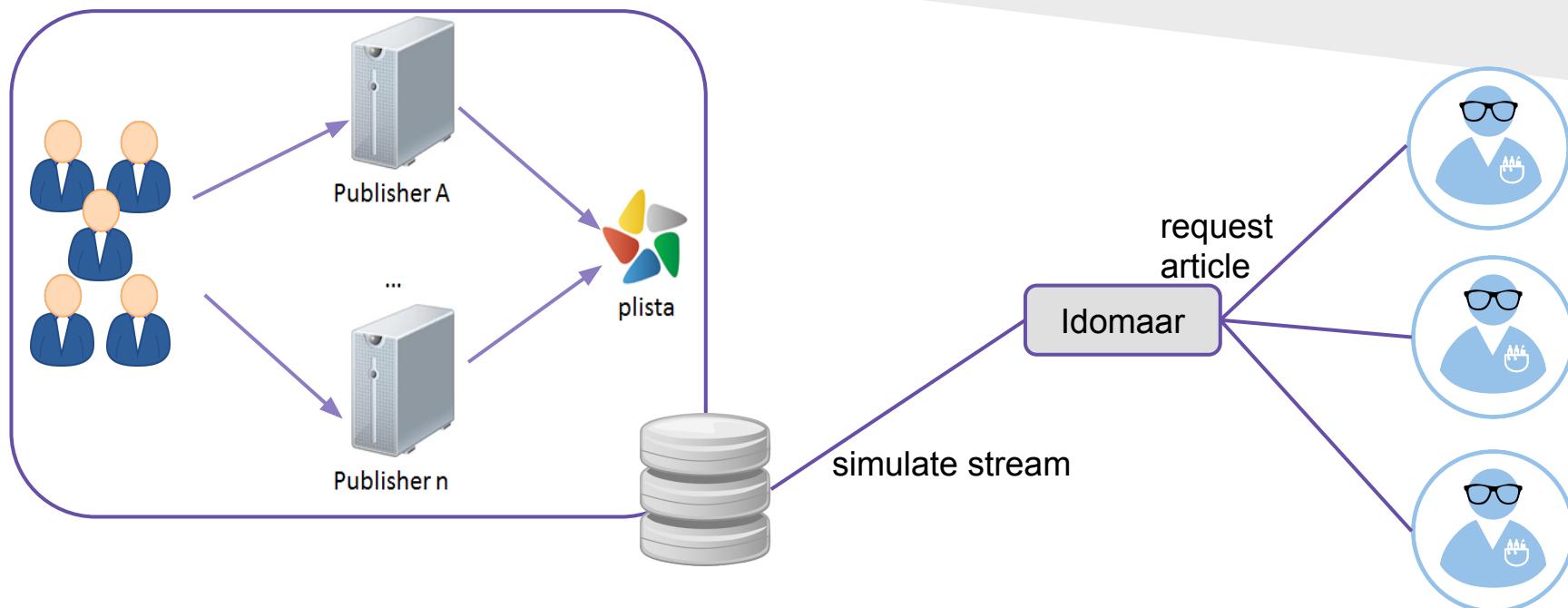
### Dataset

- Traffic and content updates of nine German-language **news content provider** websites
- **Traffic:** Reading article, clicking on recommendations
- **Updates:** adding and updating news articles

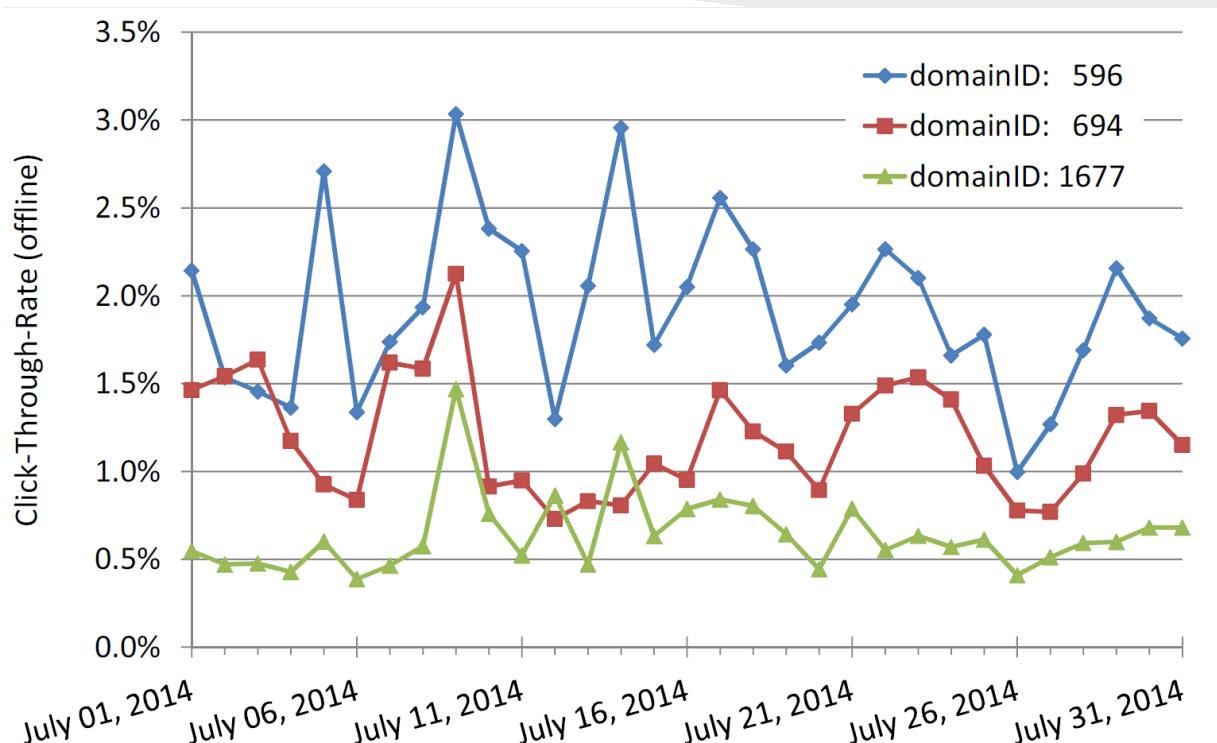
### Evaluation

- **Simulation** of data stream using **Idomaar framework**
- Participants have to **predict interactions** with data stream
- Quality measured by the ratio of successful predictions by the total number of predictions

# Simulation of data stream



# Example results (Task 1)



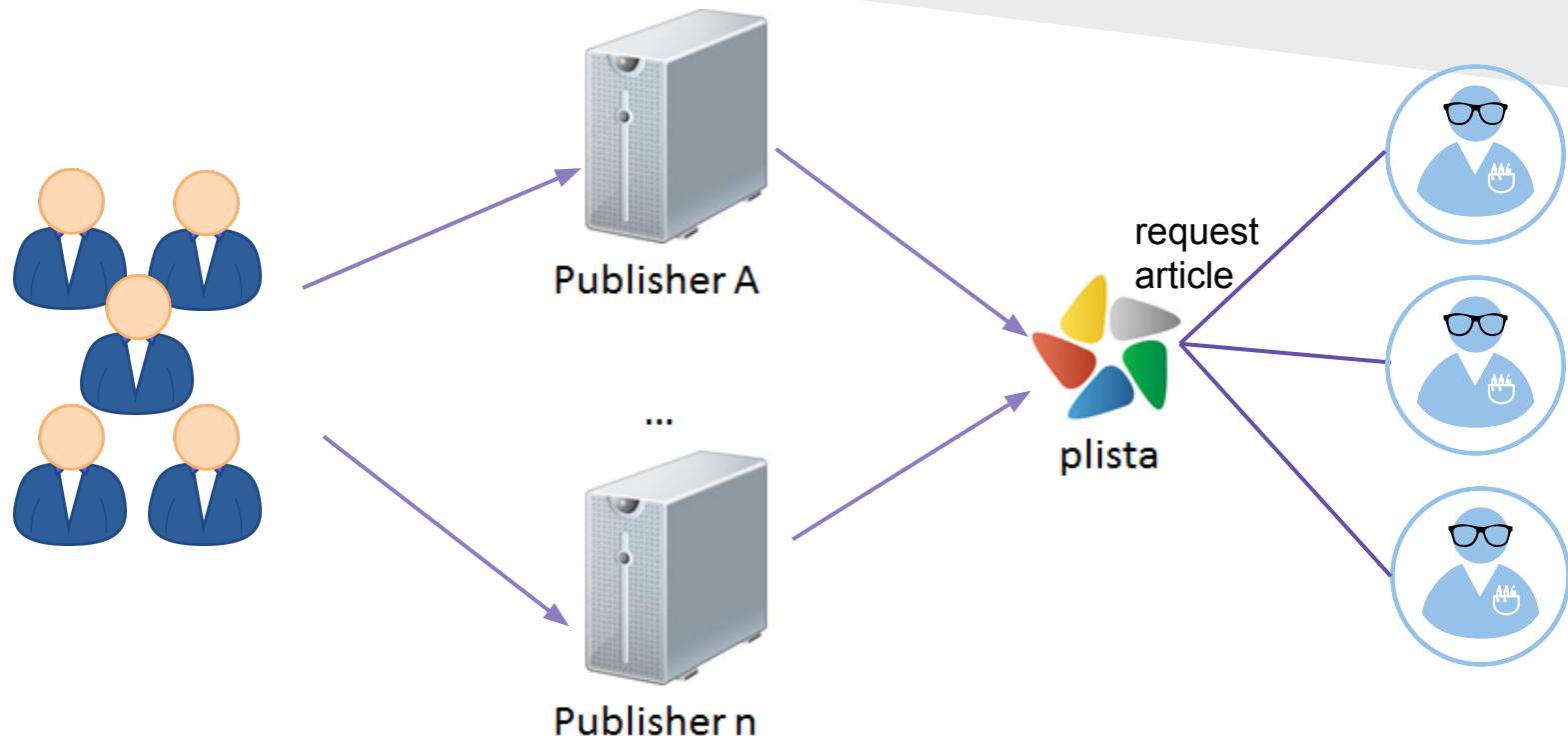
# Task 2: Online Evaluation

## Recommend news articles in real-time

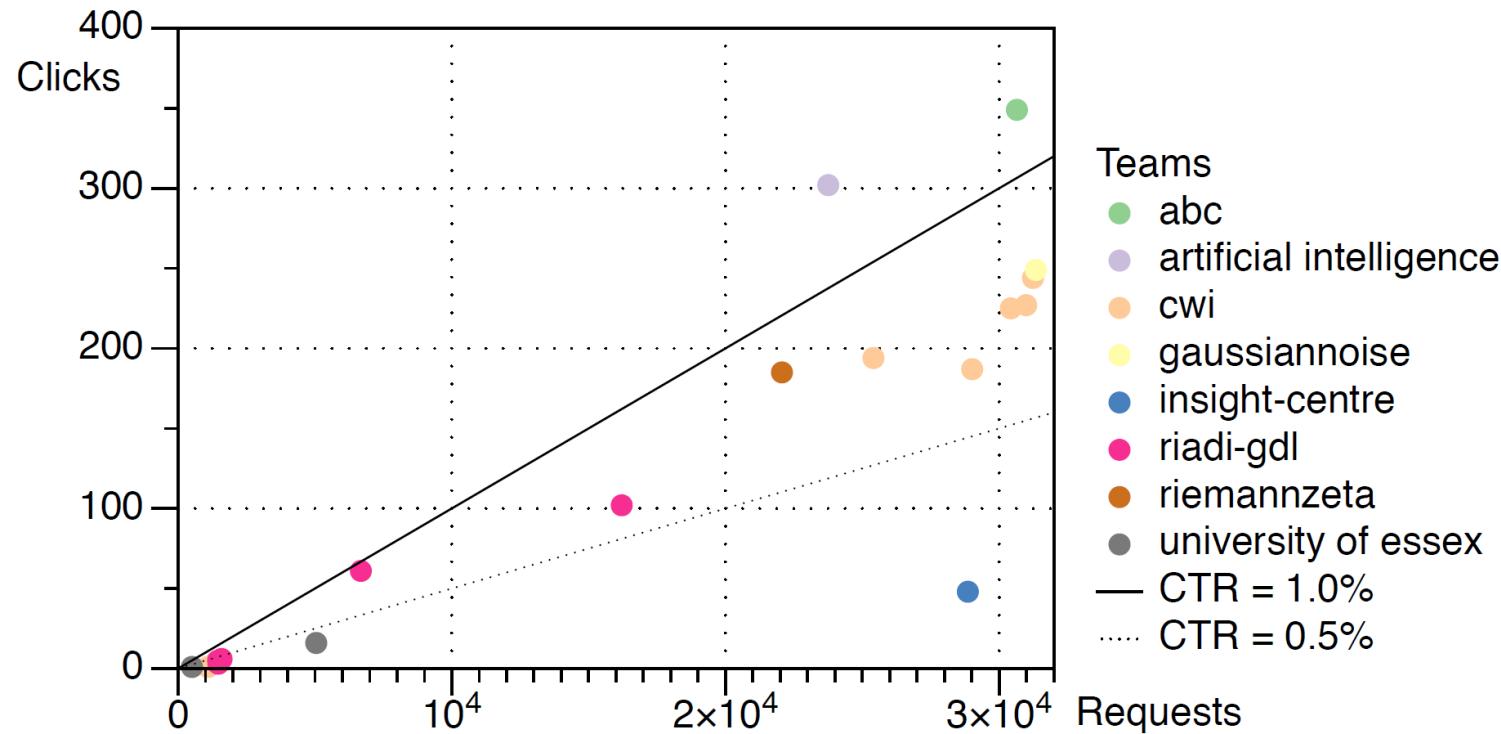
- Provide recommendations for visitors of the news portals of plista's customers
- Ten portals (local news, sports, business, technology)
- Communication via [Open Recommendation Platform \(ORP\)](#)

- Benchmark own performance with other participants and baseline algorithm
- We determine best performing algorithms during three pre-defined evaluation periods
- Standard evaluation metrics

# Live recommendation



# Example results (Task 2)



# Overview

- Evaluating Recommender Systems
  - Offline vs. Online Evaluation
  - Case Study: CLEF NewsREEL
- Introduction to Stream Based Recommendation
  - Streamed Data vs. static sets
  - Challenges
  - Popular stream based recommendation algorithms
- Idomaar
- Open Recommendation Platform

# Data Sets

We **record** interactions between the static collections of users and items obtaining a **fixed** data set.

# Data Sets

- matrix representation
- users ~ items
- preferences for known pairs
- sparsity

	Hundert-wasserhaus	Belvedere Complex	Hofburg Palace	Schonbrunn Palace	Wiener Rathaus
Paul	2		4	5	2
Peter			5		3
Susan			1	2	

# Data Sets

- Examples



NETFLIX

movielens

last.fm

# Reflection on Evaluation with Data Sets

## Advantages

- low complexity
- deterministic
- comparability
- re-producibility

## Disadvantages

- predicting previous interactions/preferences
- disregarding trends/shifting preferences
- ignoring technical requirements

# Data Streams

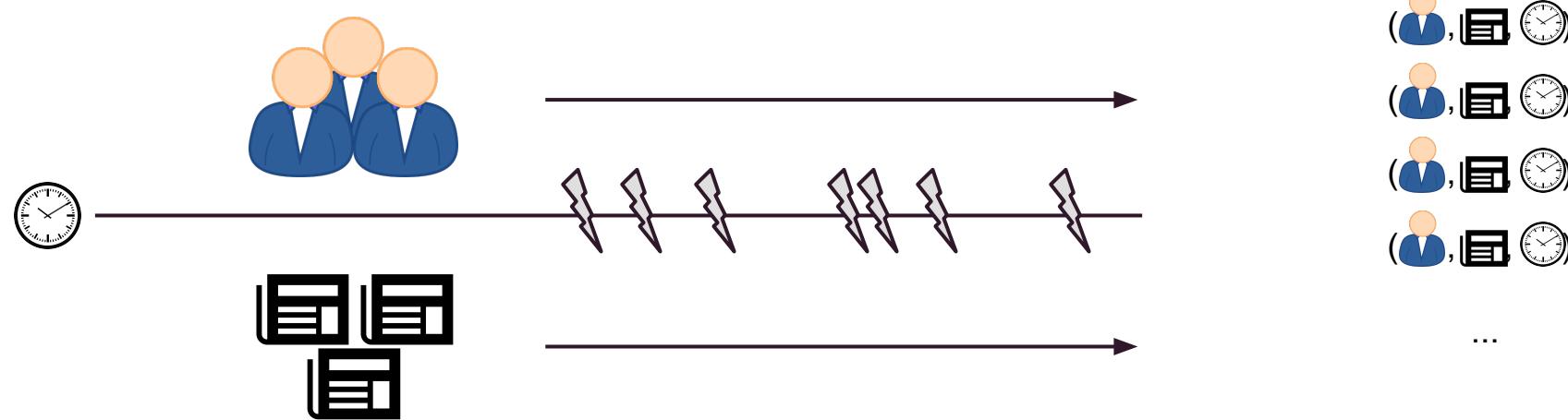
## Data Stream

“A sequence of timely ordered interactions between members of the dynamic collections of users and items.”

- order
- dynamic collections
- lack of explicit start/end

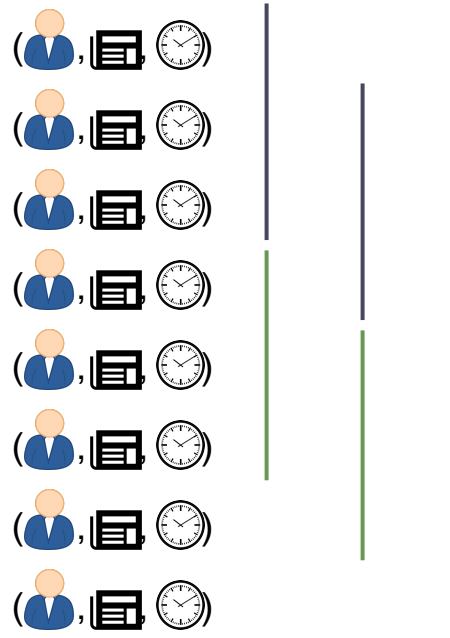
# Data Streams

- time induced order
- new users
- new items

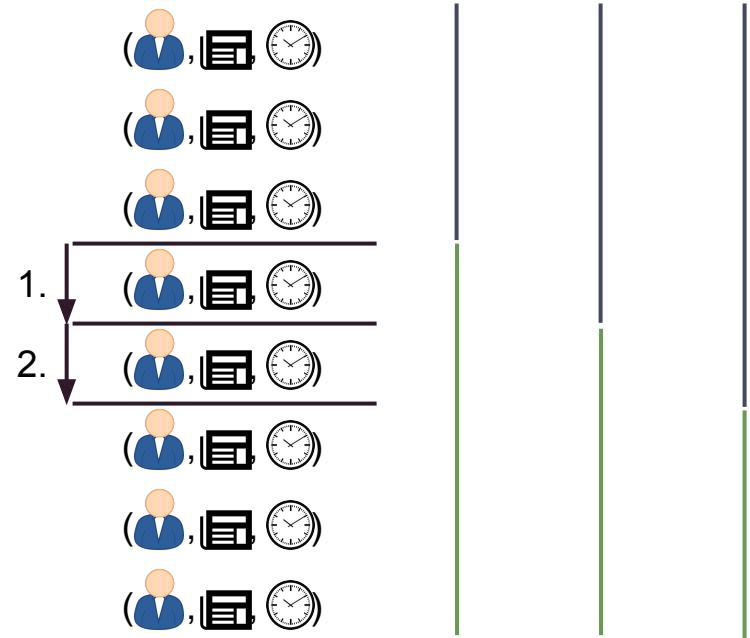


# Data Streams

# Sliding Window



# Shifting Split



# Data Streams

## Parameter

- **sliding window:**
  - window size (#data points vs time span)
  - slide rate
- **shifting split:**
  - initial split
  - shift

# Data Streams

## Advantages

- representative of real use case
- data continuously updated

## Disadvantages

- stochasticity
- hardly reproducible
- more complex

# [online/offline] ~ [data set/stream]

	Data Set	Data Stream
Online	$\{(u,v,t)\}$ observed interactions → disregard t for a sample of data points → obtain a temporary batch data set	$\{(u,v,t)\}$ observed interactions → respond to requests as they occur → discard interactions/discount their impact as time progresses  ⚠ access to interactive system required
Offline	$\{(u,v,t)\}$ observed interactions → disregard t for all data points → obtain an offline data set	$\{(u,v,t)\}$ observed interactions → re-iterate events according to t → choose between sliding window and shifting split or combination thereof

u := user  
v := item  
t := time

# Challenges

Which obstacles do we face as we evaluate streams?

- parameter tuning
- trends and shifting preferences
- exploration vs exploitation dilemma
- technical requirements
  - limited time to respond to requests
  - high volume of request with additional peaks
  - few time & limited resources to update modells

# Challenges

## Parameter Tuning

- training data
- evaluation data
- algorithm
- evaluation criteria
- meta parameter

# Challenges

## Trends and Shifting Preferences

- trends affect the performance of a recommender
  - existing models reflect the system's state at the time of their creation
- we have to re-train our models to keep up with user preferences and novel items

# Challenges

## Exploration vs Exploitation

Suppose we evaluate 3 algorithms A, B, & C. We split the traffic such that each receive  $\frac{1}{3}$ . Now someone comes up with a new algorithm D. How much traffic should we route to D? Whose traffic do we reduce?

→ we have to balance exploration vs exploitation

# Challenges

## Technical Requirements

- response time < 100ms
- requests arrive at rates of up to 100 messages/s
- re-training depends on the rate of new users & items

# Challenges

What do we get in return?

- scenario closer to actual use case
- actual feedback of users
- impression on algorithms' demand for resources

# Popular Algorithms for Streams

Generally, we consider a **recommendation algorithm** as a function that takes a **sequence of interactions**  $\{(u, v, t)\}$ , optionally additional parameter  $\Theta$  and a **user**  $u$  and returns a **list** of items  $[v_1, v_2, \dots]$  ordered by estimated preference.

# Popular Algorithms for Streams

- popularity
- recency
- trends
- random
- content-based filtering
- collaborative filtering
- combinations of above methods

# Popular Algorithms for Streams

## Popularity

INPUT:  $\{(u, v, t)\}$

compute frequency table for all  $v$

OUTPUT:  $[v_1, v_2, v_3, \dots]$  ordered by decreasing count

Assumption: visitors enjoy reading articles which are interesting to many other visitors.

# Popular Algorithms for Streams

## Recency

INPUT:  $\{(v, t)\}$

memorise when articles had been created

OUTPUT:  $[v_1, v_2, v_3, \dots]$  ordered by decreasing creation date

Assumption: visitors enjoy reading the most recently created articles.

# Popular Algorithms for Streams

## Trends

INPUT:  $\{(u, v, t)\}$ , segment size, number of segments

1. split stream in equally spaced segments
2. compute frequency table for each  $v$  in each segment
3. compute trend for each  $v$

OUTPUT:  $[v_1, v_2, v_3, \dots]$  ordered by increasing trend

Assumption: visitors enjoy reading the articles which continue to attract increasing numbers of views.

# Popular Algorithms for Streams

## Random

INPUT:  $\{v\}$

OUTPUT:  $[v_1, v_2, v_3, \dots]$  randomly ordered

Assumption: visitors enjoy reading surprising articles.

# Popular Algorithms for Streams

## Content-based Filtering

INPUT:  $\{(u, v, t)\}$ , similarity measure

1. determine articles target user has visited
2. compare their contents with all other articles

OUTPUT:  $[v_1, v_2, v_3, \dots]$  ordered by similarity to previously read articles

Assumption: visitors enjoy reading articles which are similar to articles they previously read

# Popular Algorithms for Streams

## Collaborative Filtering

INPUT:  $\{(u, v, t)\}$ , similarity measure

1. determine articles target user has visited
2. find users with similar preferences

OUTPUT:  $[v_1, v_2, v_3, \dots]$  ordered by predicted preferences

Assumption: visitors enjoy reading articles which like-minded visitors had enjoyed

# Overview

- Evaluating Recommender Systems
  - Offline vs. Online Evaluation
  - Case Study: CLEF NewsREEL
- Introduction to Stream Based Recommendation
  - Streamed Data vs. static sets
  - Challenges
  - Popular stream based recommendation algorithms
- Idomaar
- Open Recommendation Platform

# Idomaar

- Idomaar within CrowdRec
- Goals and Peculiarities
- Workflow:
  - Real systems
  - Idomaar emulation
- Architecture and Evaluation process
- Technologies

# CrowdRec - overview

- **The project:** CrowdRec improves recommendations for real-world use scenarios by *fusing information sources*, and *engaging users* to actively provide additional information.
- **The mission:** CrowdRec aspires to move from recommenders that fit the conventional “Wikipedia definition”... ...to a new breed of recommenders that predict connections between a **user in a use context** and **content**, **information**, relationships, products, services, actions, and tasks.



# CrowdRec - objectives

- **O1: Stream Recommendation**

Current systems fall far short of exploiting all available information, esp. in social network context.

- **O2: Crowd Engagement**

Recommendation improves as users are more engaged and contribute more information.

- **O3: Deployment and validation**

More useful algorithms result as recommender system research moves “out of the laboratory”.



# Idomaar - Goals (1)

Address real-world use cases and innovate:

- **Implementation benchmarking:**

An "agnostic" appraisal platform able to test both algorithmic effectiveness and implementation efficiency

- **Tendering – external evaluation:**

A benchmarking tool able to evaluate and compare different recommendation solutions for operators and assist business in selecting the one that fits best to its needs

# Idomaar - Goals (2)

Address real-world use cases and innovate:

- **Algorithm benchmarking – internal evaluation:**

A method that is capable of comparing recommendation algorithms working on either static or stream data

- **End-to-end measurement tool:**

A tool able to perform end-to-end measurements of several dimensions of a recommender algorithm (w.r.t. different implementations), such as: quality, robustness, and scalability

# Why another framework?

Many recommendation frameworks are already available,  
so, why another one?

Language Agnostic

Reproducibility

Real world scenario

3D evaluation

Streams of data

# Why another framework?

How to compare multiple algorithms implemented in different frameworks? (or even the same algorithm)

Language Agnostic

Existing evaluation frameworks are bound to given recommendation algorithms and use specific languages

Reproducibility

- What if I would like to test or **compare my own implementation in different languages?**
- How can I compare them and choose the best one?

Real world scenario

3D evaluation

Streams of data

# Why another framework?

How to grant the reproducibility of the same, identical test?

Language Agnostic

Reproducibility

Real world scenario

3D evaluation

Streams of data

The framework must allow anyone to reproduce the same identical test (e.g., for recommendation algorithm comparison), considering:

- Architecture
- Operative system
- Hardware

**Virtual machines and autoprovisioning** have been used.

# Why another framework?

Does the recommendation algorithm fit into my production environment?

Language Agnostic

Implementing a recommendation engine is much more than simply “run the algo”

Reproducibility

In a real production environment you need to use queue, web servers, databases, etc. that can influence the final results in terms of:

Real world scenario

- **Performance**
- **Technical limitation**

3D evaluation

Streams of data

# Why another framework?

## What's beyond recommendation quality?

Language Agnostic

Evaluation is the real objective and it should cover all the KPIs of a recommendation engine:

Reproducibility

- **End-user** requirements
- **Business** requirements
- **Feasibility**

Real world scenario

3D evaluation

Idomaar objective is to cover all the three aspects of the evaluation (3D benchmarking)

Streams of data

# Why another framework?

How to treat real scenarios where data are continuously streamed?

Language Agnostic

Recommendations should react to users events in real-time

Reproducibility

Stream of data, time, and context are key variables to be taken into account in the recommendation evaluation:

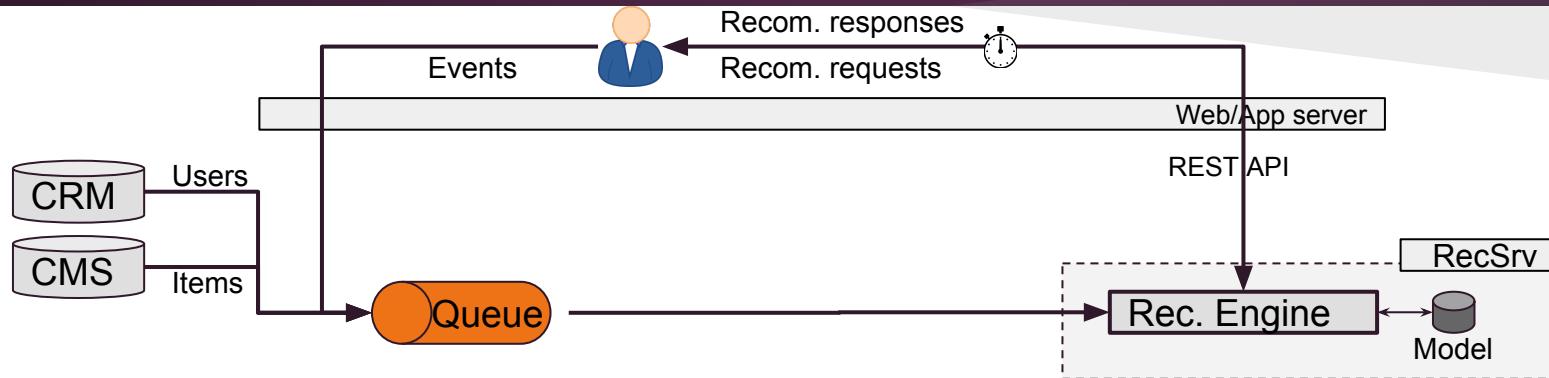
Real world scenario

- Architecture: the framework should be as similar as possible to a production environment
- Evaluation: data has to “flow” into the algorithm in a controlled way and kpi calculation must consider time and context

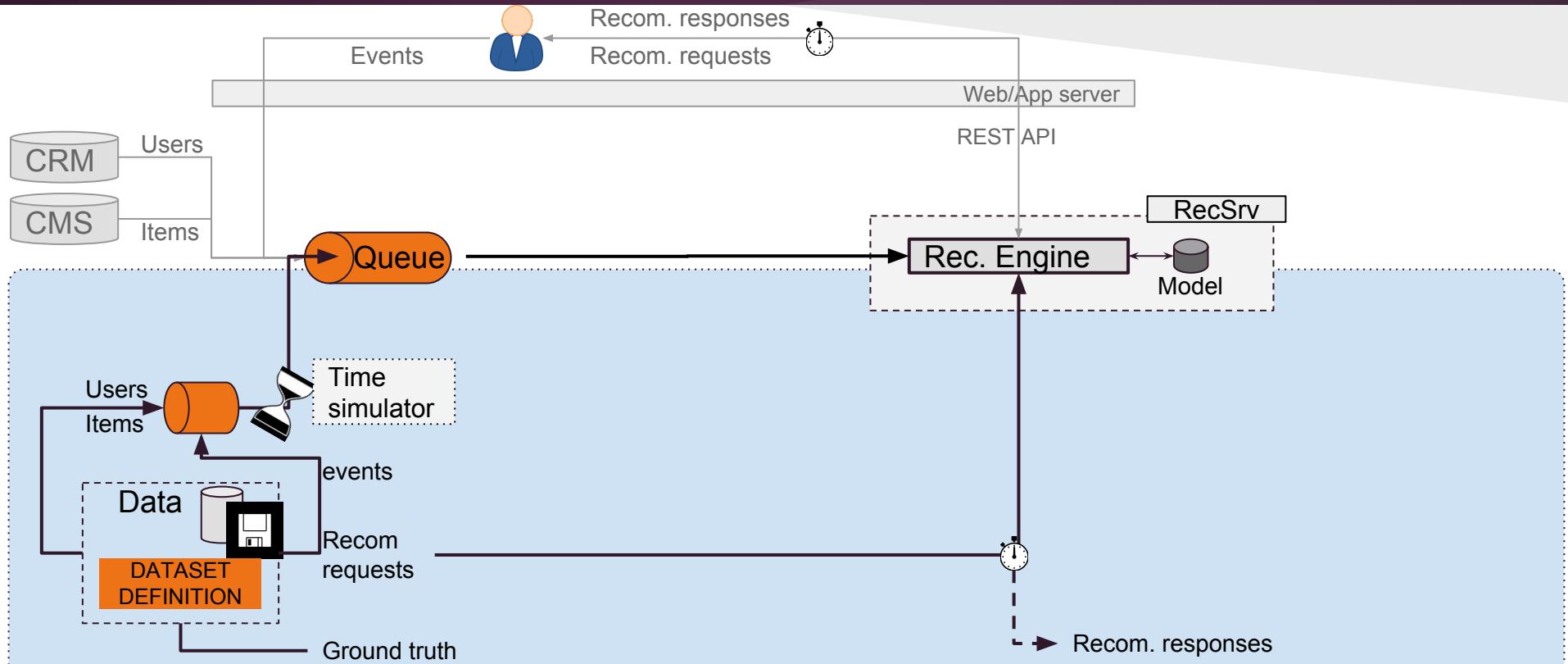
3D evaluation

Streams of data

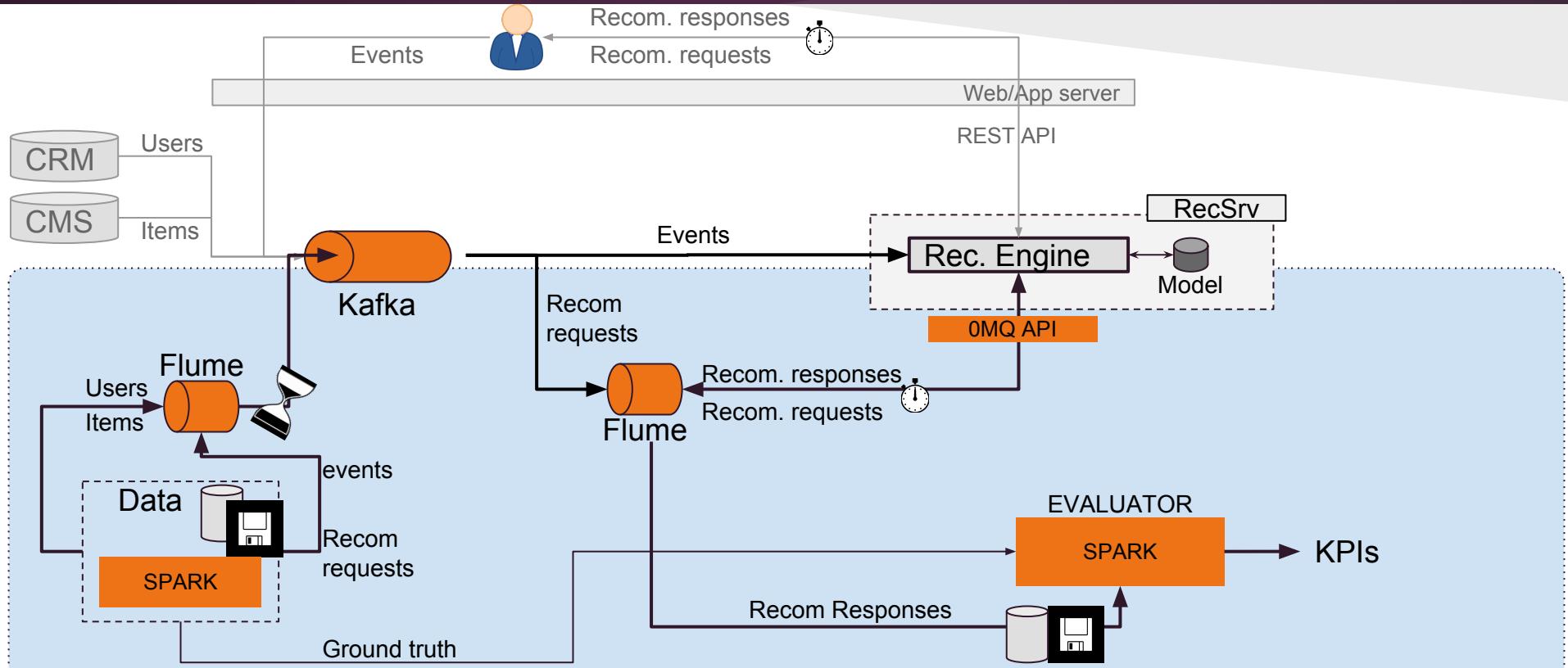
# Workflow: real system



# Workflow: Idomaar (1)



# Workflow: Idomaar (2)



# Architecture (1)

the **algorithms** to test, both state-of-the-art algorithms and new solutions implemented within the CrowdRec project

---

**Algorithms**

the **evaluation logic**, experimenting with the available algorithms to compute both quality (e.g., RMSE, recall) and system (e.g., execution and response time) metrics.

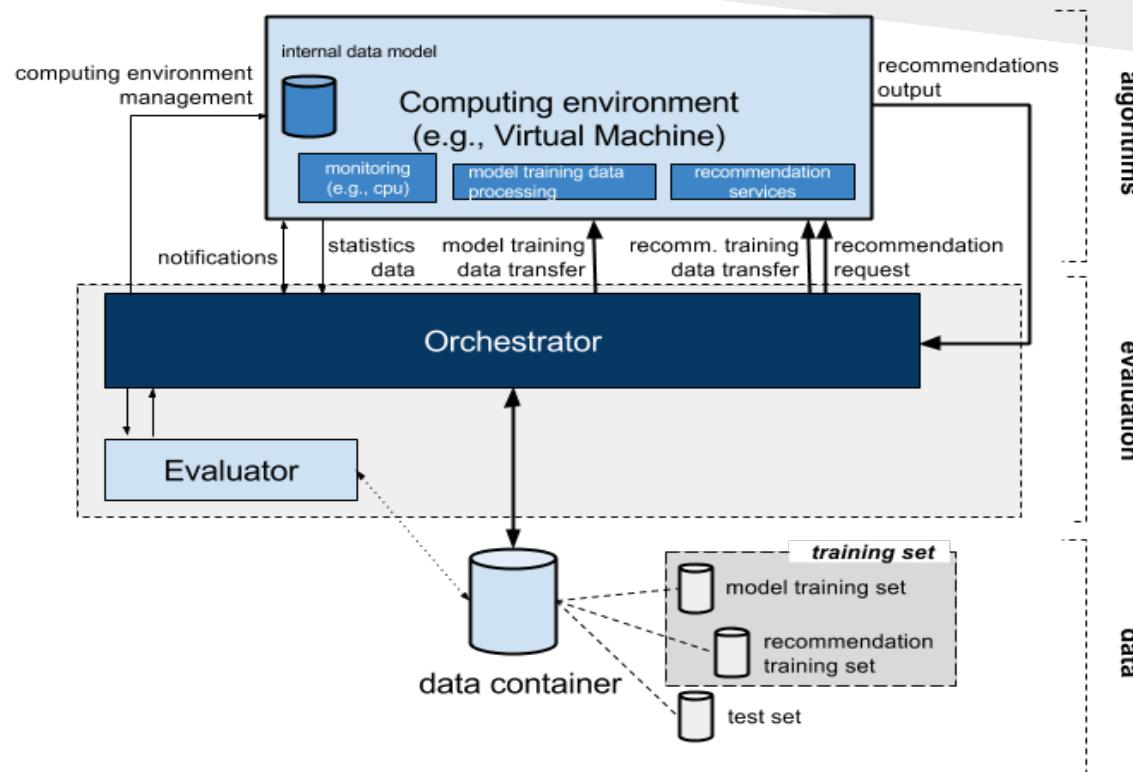
---

**Evaluation logic**

the **data**, i.e., the datasets made available to the practitioners

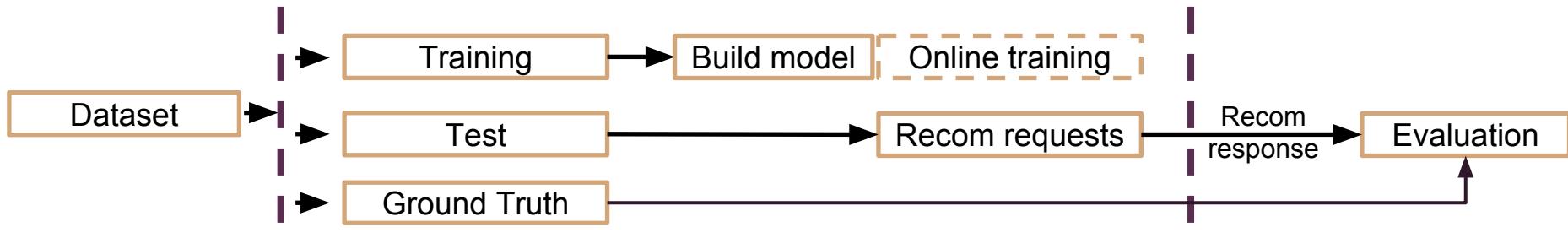
**Data**

# Architecture (2)



# Evaluation process

The orchestrator “feeds” the computing environment with training and test data and “coordinates” all the components



A common dataset format

Public or private available dataset can be used

The first operation is data splitting to create:

- training
- test
- ground truth.

- The recommendation engine bootstraps the models with training data  
- Recommendation requests can be served (while, optionally, further training data might be coming and recommendation model incrementally update)

Recommendation results are compared with ground truth to calculate the quality KPIs.

Additional data are collected to compute performance and non-functional KPIs

# Data

**5 attributes:** type, id, ts, properties, linkedEntities

**entity**  
**type:** movie

**id:** 2769592

**ts:** 1364002198

**properties:** {"title":"Kiss Shot Truth or Dare (2013)"}

**linked entities:** {"director": {"type": "person", "id": 1221}}

**relations**  
**type:** rating.explicit

**id:** 1

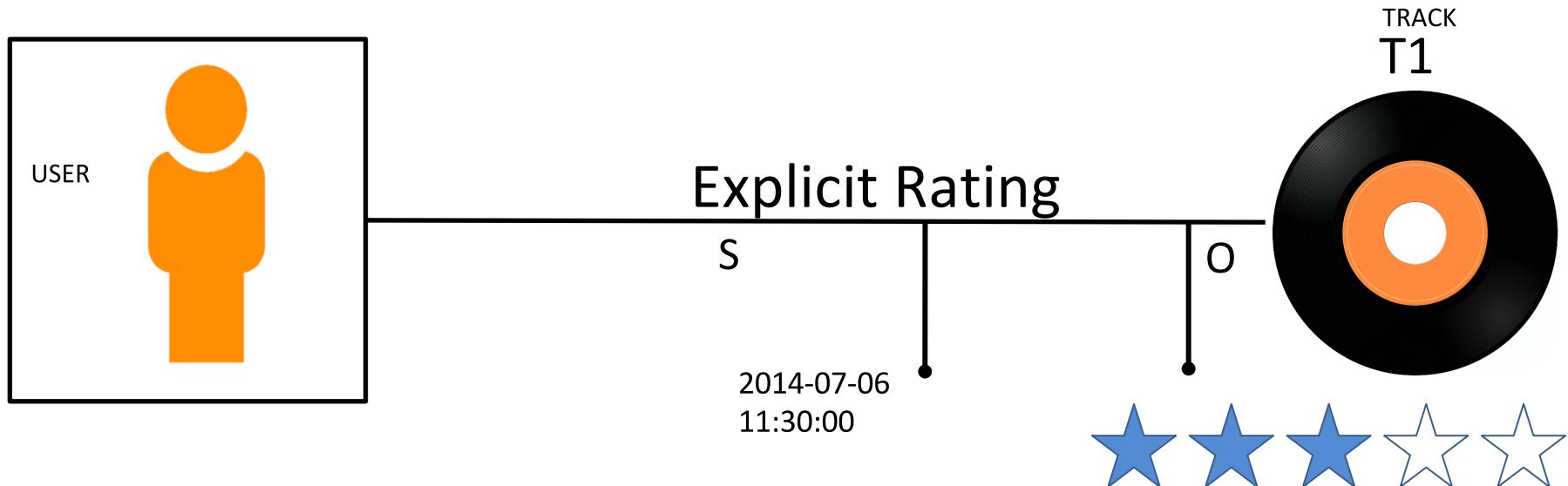
**ts:** 1363245118

**properties:** {"rating": 9}

**linked entities:** {"subject": "user:1", "object": "movie:0120735"}

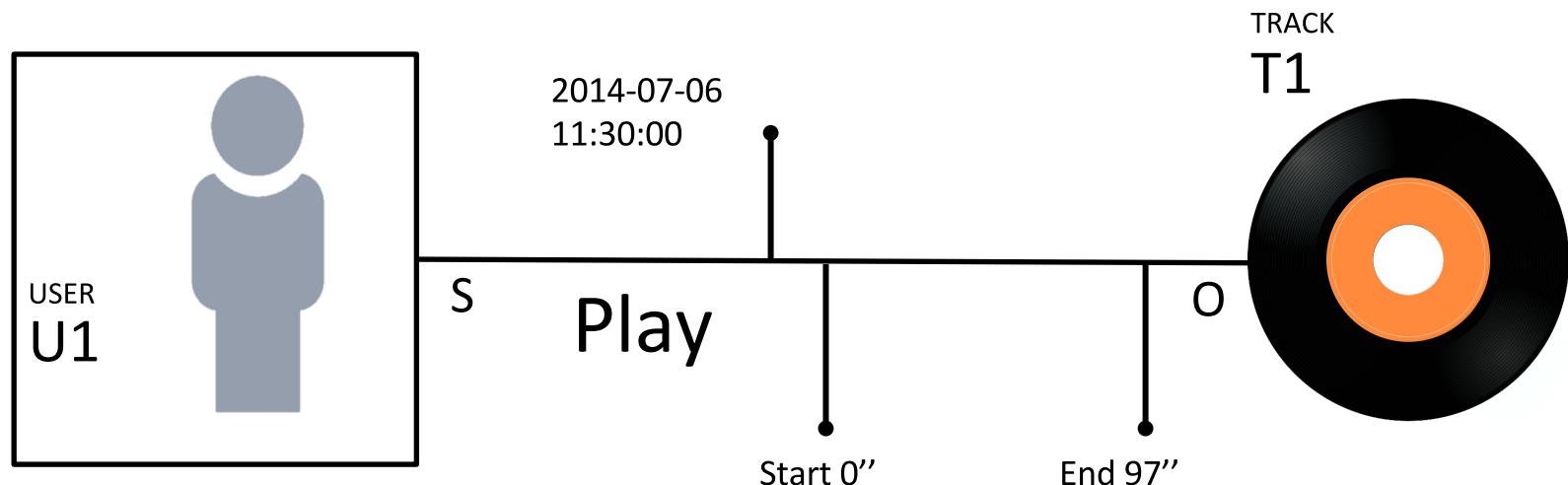
# Data

## Explicit Rating

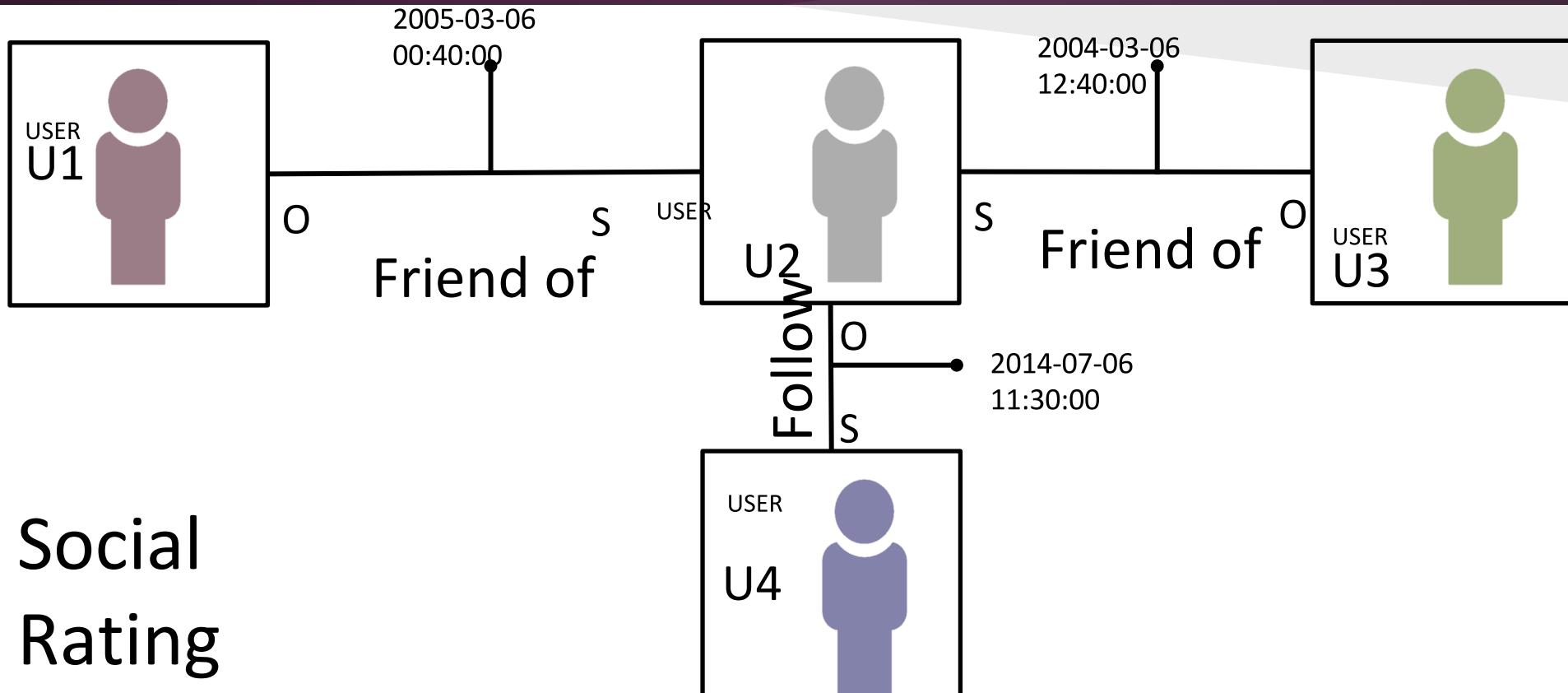


# Data

## Implicit Rating



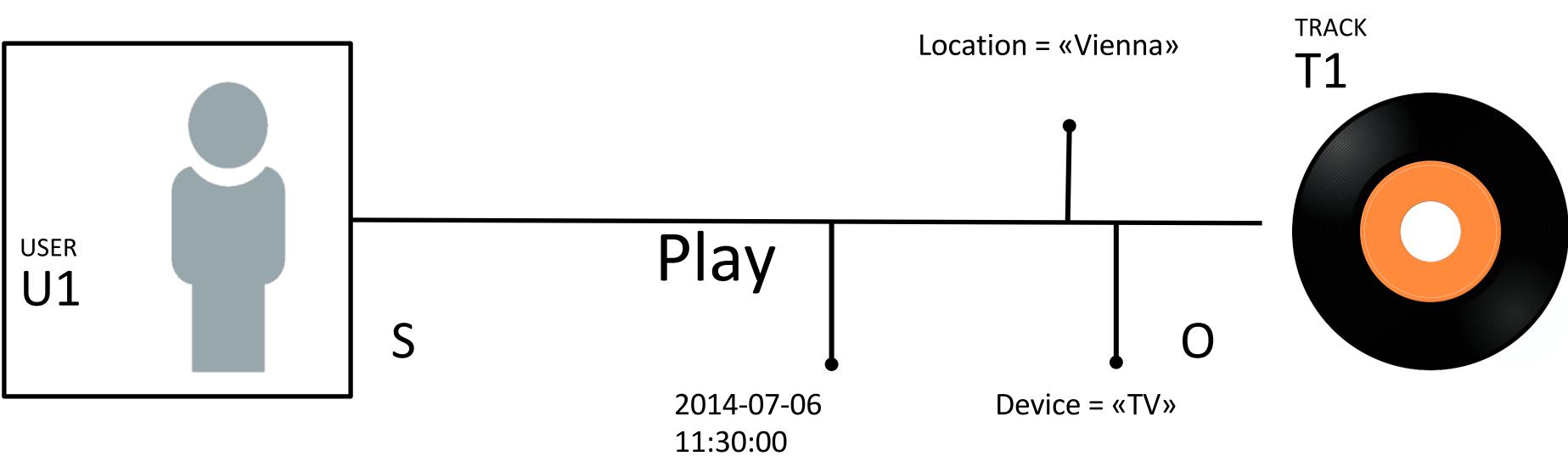
# Data



Social  
Rating

# Data

## Context



# Technologies: auto-provisioning

Computing environment should be reproducible for different user in different context, the idea: “using configuration based virtual machines”.



VAGRANT

Provides a way to “describe” virtual machine and manage their execution regardless of the virtualized infrastructure



The installation of the required software (operative system prerequisites, languages, algorithms) has to be reproducible, Puppet can automate the installation and the “configuration” can be saved on GitHub.

# Technologies: messaging



## ZeroMQ

control messages between components



## Flume

connect different data sources and sinks



## Kafka

send data to computing environment

# Technologies: data processing



Spark is a **programming language**:

- **Parallelized**: a correct functional programming grants Spark to work in parallel out of the box
- **Lazy**: it computes only when needed
- **Fast**: *faster* than map reduce or other tools
- **Accessible**: it offers *Python*, *Java* and *Scala* APIs out of the box
- **Expandable**: several *additional libraries* are available

# First steps into Idomaar

1. Visit the official website: <http://rf.crowdrec.eu/>
2. Read the documentation on the github project wiki
3. Start the quick start evaluation demo based on a Mahout implementation
4. Create your recommendation machine via puppet or custom script, starting from the Mahout template at: <https://github.com/crowdrec/idomaar/tree/master/computingenvironments/o1.linux/o1.centos/o1.mahout>
5. Implement the interface with zeromq or http protocol following template from mahout machine (zeromq) or Clef machine (<http://www.clef-newsreel.org/wp-content/uploads/2015/02/TutorialTask2.pdf>)
6. Prepare the evaluation data, following the schema at idomaar wiki <https://github.com/crowdrec/idomaar/wiki/DATA-FORMAT>. A sample evaluation dataset can be found at: [https://github.com/crowdrec/datasets/tree/master/o1.MovieTweetings/datasets/snapshots\\_10K/evaluation](https://github.com/crowdrec/datasets/tree/master/o1.MovieTweetings/datasets/snapshots_10K/evaluation)
7. Start the evaluation process using the start script in idomaar project, an example of the parameters can be found in 'idomaar-demo.sh'

# References

## Resources

- **Idomaar**: <http://rf.crowdrec.eu/>
- **Crowdrec**: <http://crowdrec.eu>

## Technologies:

- **0MQ**: <http://zeromq.org/>
- **Kafka**: [kafka.apache.org](http://kafka.apache.org)
- **Flume**: [flume.apache.org](http://flume.apache.org)
- **Spark**: [spark.apache.org](http://spark.apache.org)
- **Vagrant**: [www.vagrantup.com](http://www.vagrantup.com)

# Overview

- Evaluating Recommender Systems
  - Offline vs. Online Evaluation
  - Case Study: CLEF NewsREEL
- Introduction to Stream Based Recommendation
  - Streamed Data vs. static sets
  - Challenges
  - Popular stream based recommendation algorithms
- Idomaar
- Open Recommendation Platform

# plista

## plista who?

- “data-driven platform for content and ad recommendations”

The screenshot shows a web browser window with a dark header bar containing the title 'auto-motor-und-sport' and a search bar with the query 'automotorsport'. Below the header is a navigation menu with links like 'Datei', 'Bearbeiten', 'Ansicht', 'Chronik', 'Lesezettel', 'Extras', and 'Hilfe'. A small logo for 'Rethink Performance...' is visible.

The main content area displays a news article about the BMW 6er Cabrio. The article includes a headline, several paragraphs of text, and a small image of the car. Below the article is a sidebar with the heading 'Weitere Neuheiten 2011' and a list of recent car models: 'Fast drei Dutzend deutsche Modell-Neuheiten', 'Das gibt es dieses Jahr von VW', 'Mehr als zweit Dutzend japanische Neuheiten', and 'Diese Änderungen kommen 2012'.

On the right side of the page, there is a sidebar titled 'Auto & Umwelt' featuring a banner for 'Umwelt Wasserstoff CO<sub>2</sub> Hybrid' and a link to 'Blog'. Below the banner, there is a section titled 'Schmidtts E3 - Blog: Zum Finale nach Brasilien' with a link to 'zum Blog'.

The bottom right corner features a large blue banner with the text 'Das könnte Sie auch interessieren' and a list of five recommended articles, each with a thumbnail image and a brief description:

- Daran erkennt man gute Autovermietungen
- VW Jetta
- Was bringt ein schadstoffarmer PKW
- Beim Autokauf ein Schnäppchen machen

At the very bottom right, there is a small watermark that reads 'powered by plista'.

# plista

## plista who?

- “data-driven platform for content and ad recommendations”

## recommendation platform based on

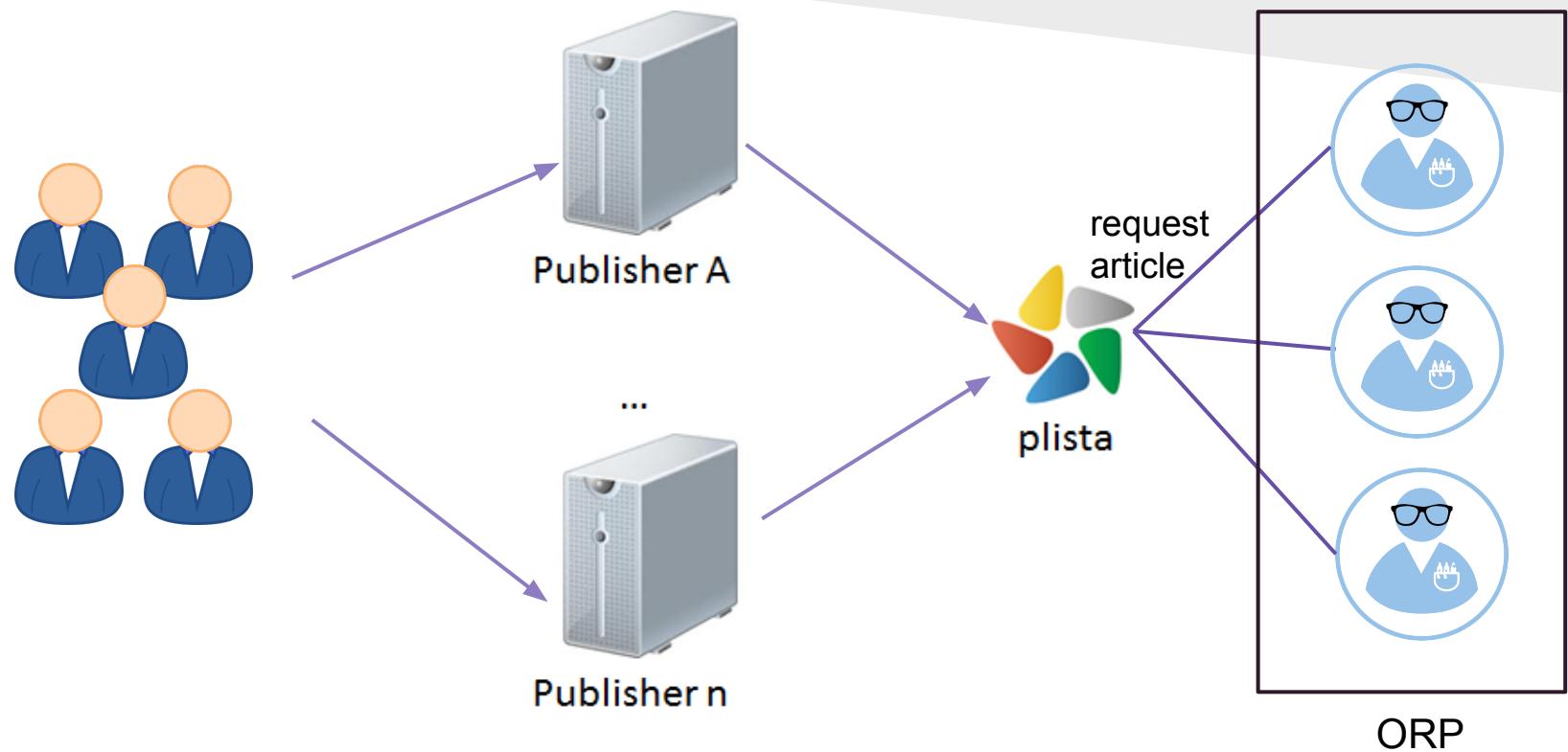
- Ensemble of unrelated, uncoupled algorithms
- Message bus to tie the architecture together

# Open Recommendation Platform

## What is it?

- API which hooks into our recommendation system
- Allows outside parties to receive events and calculate recommendations instead of us
- Live recommendations instead of static data set

# Open Recommendation Platform



# Open Recommendation Platform

Why should we do that?

- Open (source) culture
- Valuable input from scientists
- Create attention

# Open Recommendation Platform

## How does it work?

- Web-Frontend
- JSON-based HTTP-API
- more on that later

# ORP - How does it work?

- Plugs into Recommender Ensemble
  - each external recommender is treated as additional algorithm
  - mixed like our internal ones
- Plugs into internal message bus
  - each external recommender has own subscription on message bus
  - small script to forward over HTTP

# ORP - How does it work?

- Challenges
  - Dealing with high latency
    - Servers potentially outside of Europe  
→ Enforce hard timeout
  - Non-reliable recommenders
    - Errors can quickly clog up the system  
→ Automatic disabling of failing recommenders

# ORP - How to participate?

- How to sign up?
  - -> CLEF NewsREEL
- How to use the web interface?
  - Should be intuitive
  - Ask your friendly presenter
- How to set up a server?
  - Sadly beyond the scope of this talk

# ORP - How to participate?

- Provide an API endpoint and receive data
  - Get code running on the server
  - Submit public API endpoint URL
  - Activate recommender
  - Watch as data comes streaming in
- Restrictions
  - Timeout: 100 ms
  - High data volume: up to 100 msg/s

# ORP - What kind of data?

[Download Documentation](#)

- JSON and HTTP
  - small refresher needed?

# ORP - What kind of data?

- **Vectors**
  - data structures allowing us to describe an object by layering attributes
  - dense and expressive
- **Context**
  - holds all information we know about the current situation
  - constructed using vectors

# ORP - What kind of data?

- Different message types
  - Recommendation request
  - Event notification
  - Item update
  - Error notification

# ORP - What kind of data?

- Example message

```
{"type":"impression","context":{"simple":{"89":0,"29":17332,"27":35774,"85":7,"4":83,"56":1138207,"69":1851422,"63":1840689,"62":1788882,"83":50,"77":140,"52":1,"14":6576010,"39":26604,"68":1851468,"16":48811,"7":2878799,"88":9,"91":2864860,"96":0,"25":239164623,"42":0,"81":2101340,"24":0,"76":0,"84":31196728850247969,"44":1902613,"82":15917633,"47":504182,"75":1926071,"6":453642,"74":1919947,"17":49023,"61":869575223622174564,"15":15807917,"22":66770,"31":0,"5":317849,"101":0,"100":1,"67":1928652,"13":2,"9":26848,"23":9,"49":19,"97":1442388373,"80":449,"37":15979816,"98":3232180,"59":1275566,"18":5,"57":0},"lists":{},"clusters":{},"recs":{},"timestamp":1442388373786}
```

# Concluding...

## CLEF-NEWSREEL NEWS RECOMMENDATION EVALUATION LAB

[HOME](#) | [TASKS](#) | [HOW TO PARTICIPATE](#) | [PUBLICATIONS](#) | [ORGANISATION](#) | [PREVIOUS CAMPAIGNS](#) | [CLEF 2015](#) | [TUTORIALS](#)

### Overview

Many online news publishers display on the bottom of their articles a small widget box labelled "You might also be interested in", "Recommended articles", or similar where users can find a list of recommended news articles. Dependent on the actual content provider, these recommendations often consist of a small picture and accompanying text snippets.

While some publishers provide their own recommendations, more and more providers rely on the expertise of external companies such as plista, a data-driven media company which provides content and advertising recommendations for thousands of premium websites (e.g., news portals, entertainment portals). Whenever a user reads an article on one of their customers' web portals, the plista service provides a list of related articles. In order to outsource this recommendation task to plista, the publishers firstly have to inform them about newly created articles and updates on already existing articles on their news portal. In addition, whenever a user visits one of these online articles, the content provider forwards this request to plista. These clicks on articles are also referred to as impressions. Plista determines related articles which are then forwarded to the user and displayed in above mentioned widget box as recommendations.

Having a large customer base, plista processes millions of user visits in real time on a daily basis. NEWSREEL provides research teams the opportunity to deliver some of these recommendations. In the second iteration of NEWSREEL which is organised as a campaign-style evaluation lab of CLEF 2015, we provide two tasks that address the challenge of real-time news recommendation. The first task allows benchmarking news recommendation algorithms in a living lab environment. The second task simulates the real-time recommendation task using a novel recommender systems reference framework which has been developed within the FP7 project CrowdRec.

