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?

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

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

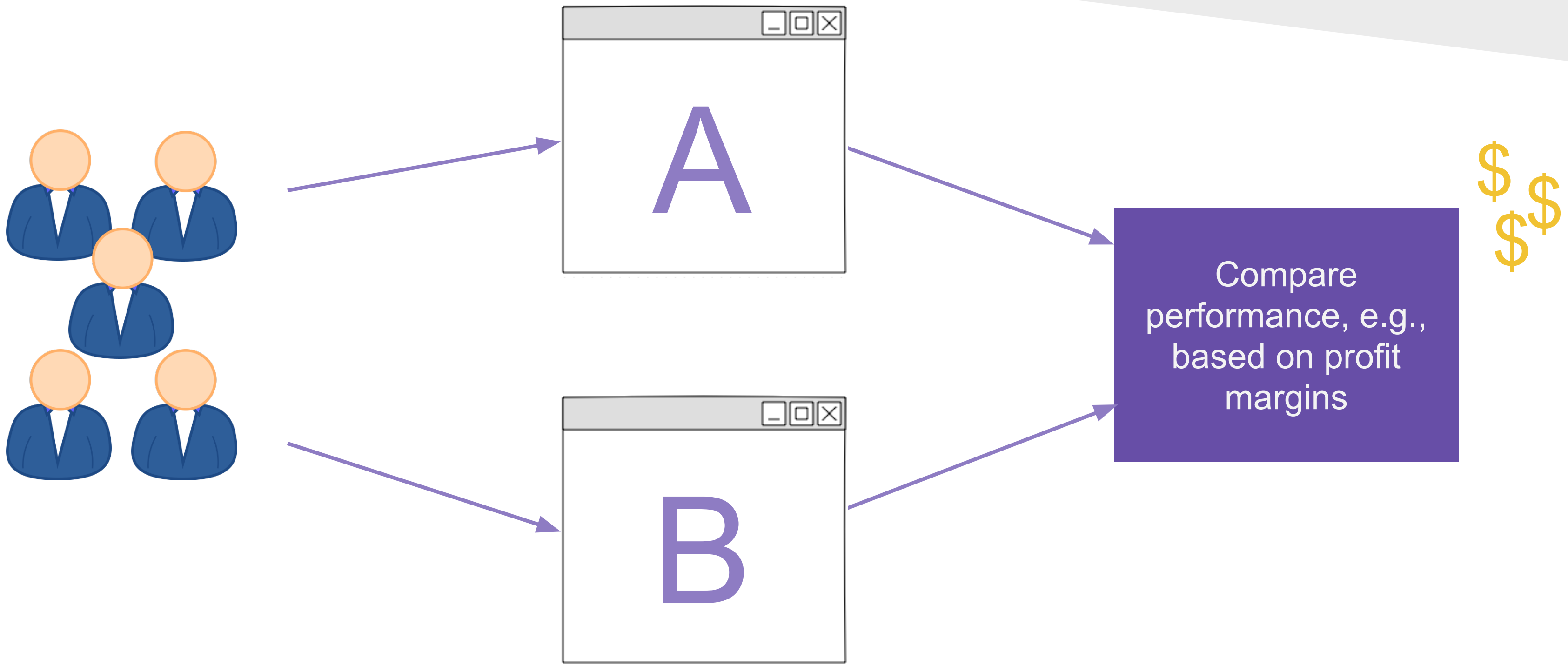
Drawbacks of offline evaluation

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

N. Ferro
"*CLEF 15th Birthday: Past, Present, and Future*," in SIGIR Forum, 48(2): 31-55, 2014.

CLEF 2016

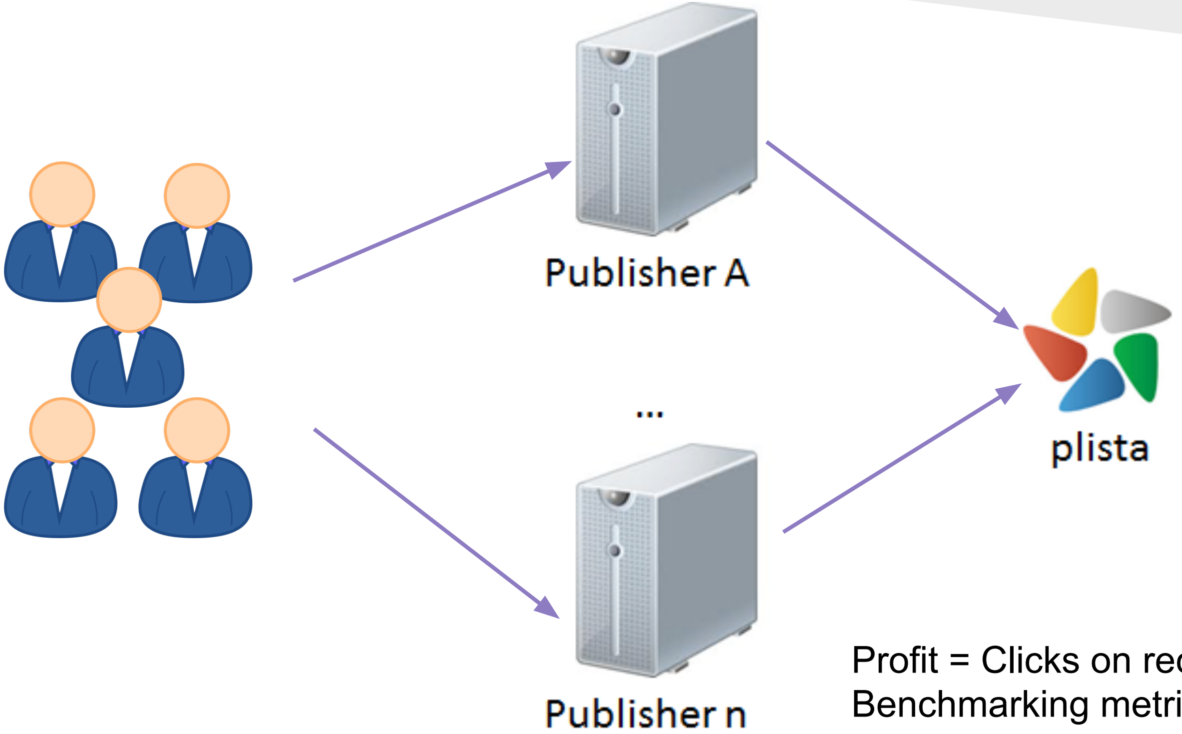
Evaluation Labs

- 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

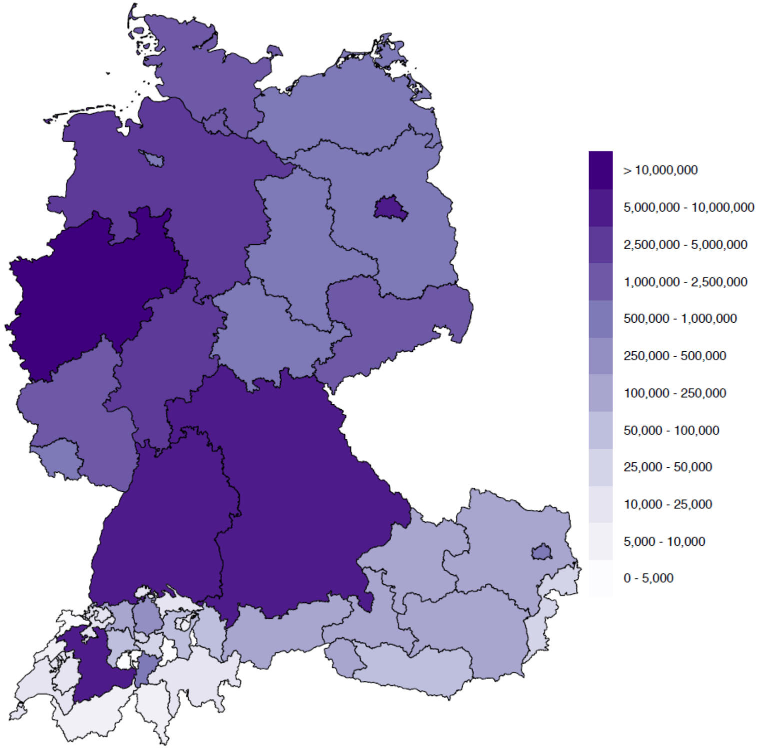
Image: Courtesy of T. Brodt (plista)

NewsREEL scenario



Profit = Clicks on recommendations
Benchmarking metric: Click-Through-Rate

Who are the users?



F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, T. Heintz,
"*Benchmarking News Recommendations in a Living Lab*," in Proceedings of CLEF 2014, Sheffield, UK, pp. 250-267, 2014.

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

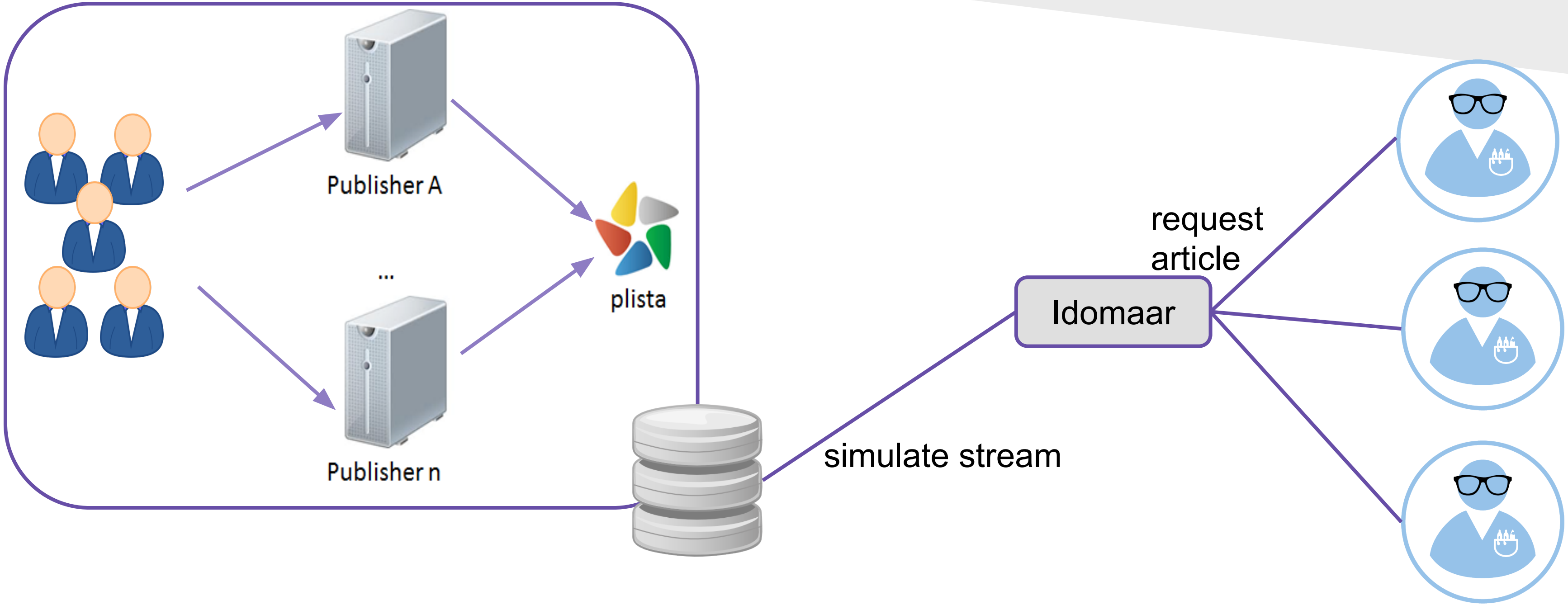
Dataset

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

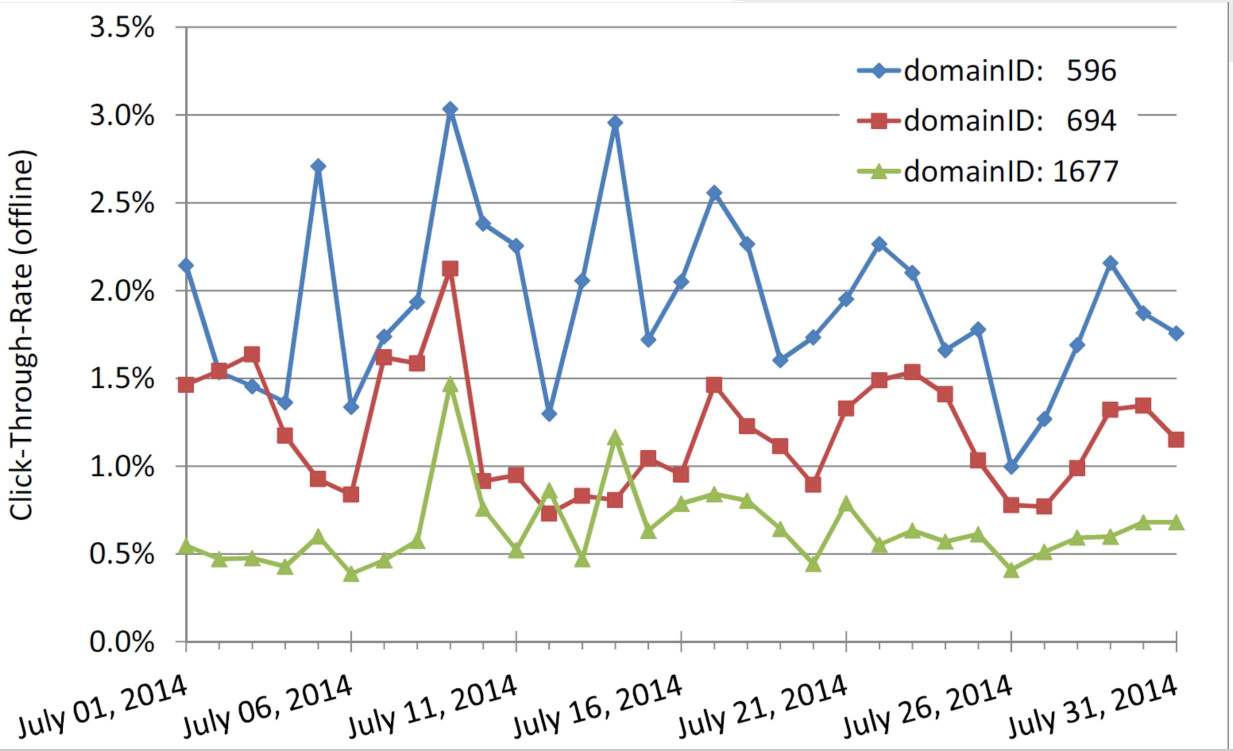
Evaluation

- **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)



B. Kille, A. Lommatzsch, R. Turrin, A. Sereny, M. Larson, T. Brodt, J. Seiler, F. Hopfgartner
"*Overview of CLEF NewsREEL 2015: News Recommendation Evaluation Lab*," in Working Notes of CLEF 2015, Toulouse, France, 2015.

Task 2: Online Evaluation

Recommend news articles in real-time

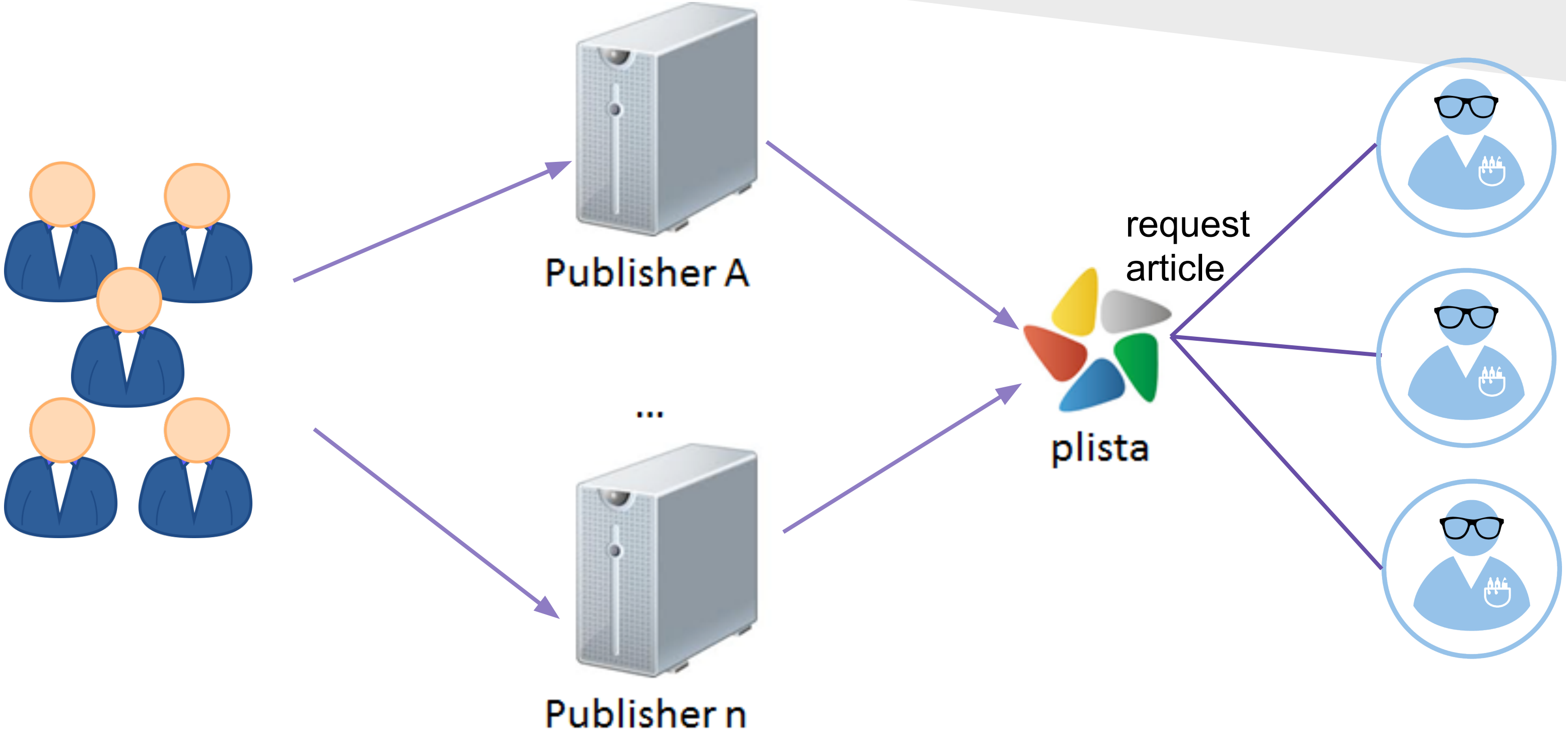
Living Lab

- 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)

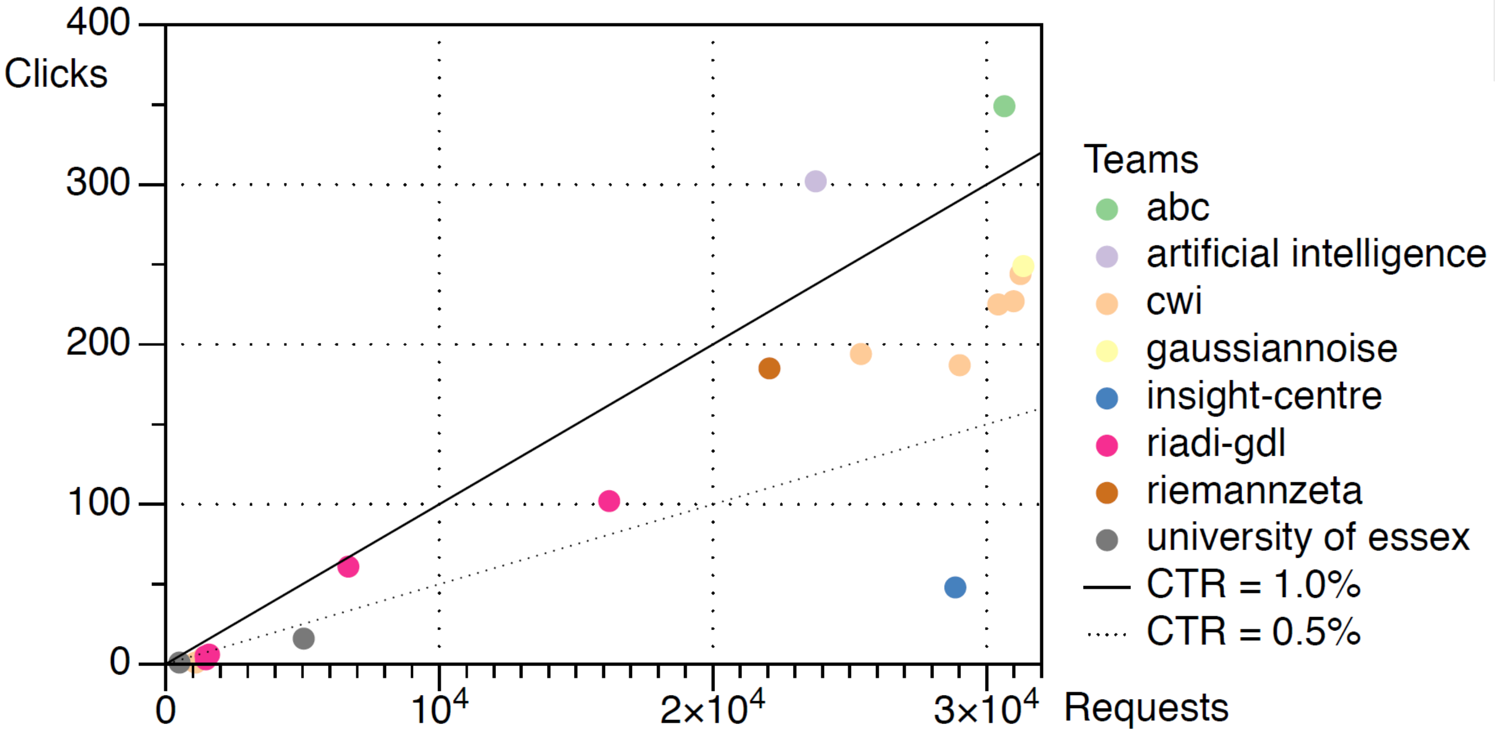
Evaluation

- 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)



B. Kille, A. Lommatzsch, R. Turrin, A. Sereny, M. Larson, T. Brodt, J. Seiler, F. Hopfgartner
"*Overview of CLEF NewsREEL 2015: News Recommendation Evaluation Lab*," in Working Notes of CLEF 2015, Toulouse, France, 2015.

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

amazon

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

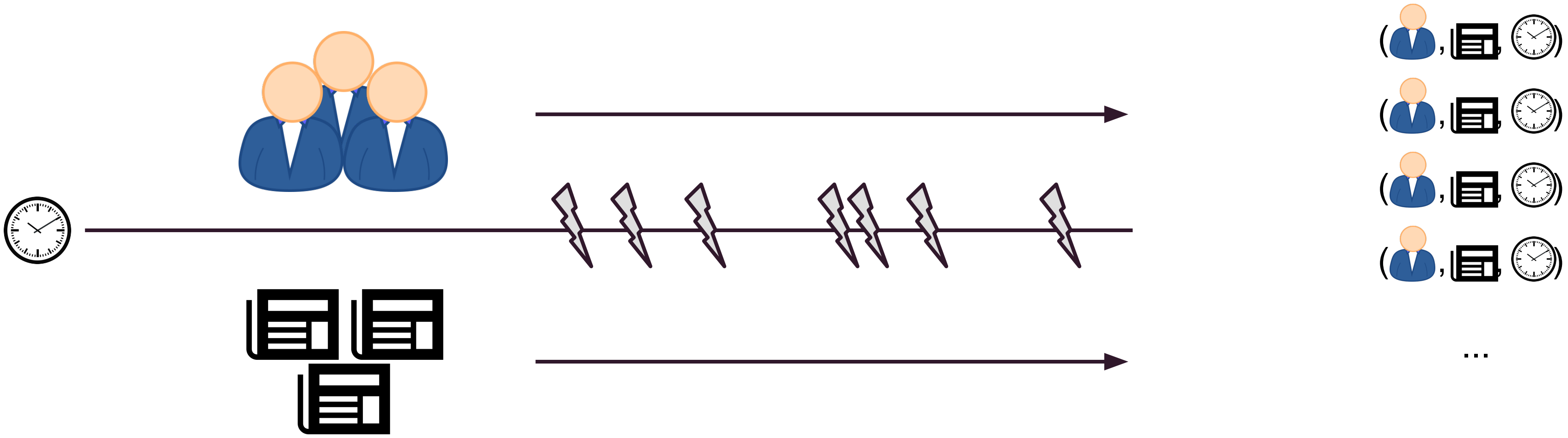
Data Stream

> "A sequence of timely ordered interactions between members of the dynamic collections of users and items."

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

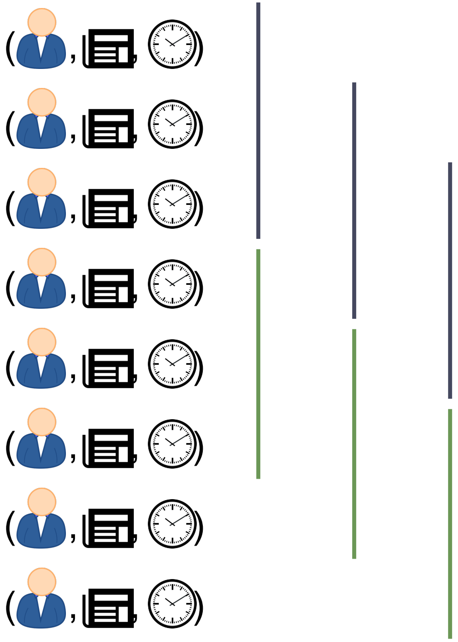
Data Streams

- time induced order
- new users
- new items

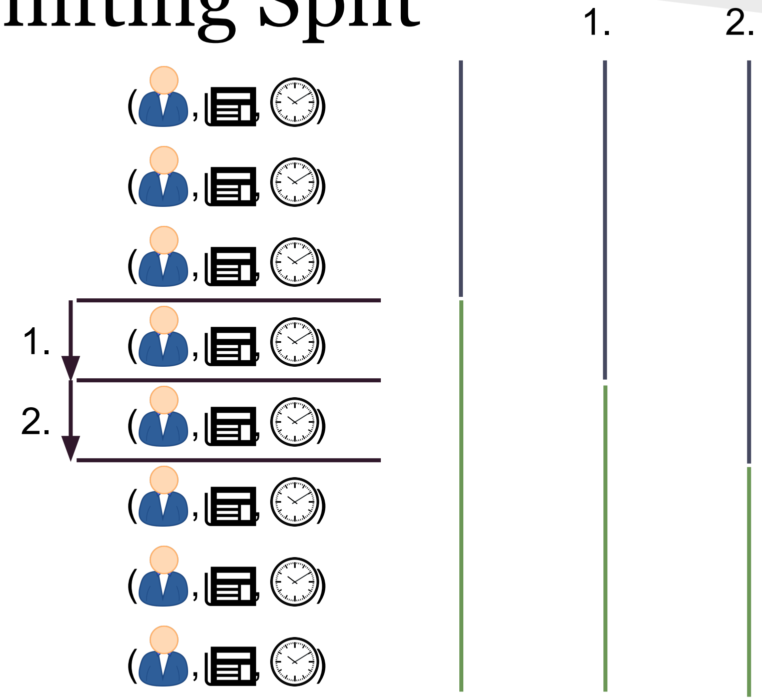


Data Streams

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 ⅓. 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 Θ and a user u and returns a list of items [v_1, v_2, …] 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, ...] ordered by decreasing count

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

A. Lommatzsch and S. Albayrak
"*Real-Time Recommendations for User-Item Streams*" in Symposium on Applied Computing 2015, Salamanca, Spain, 2015.

Popular Algorithms for Streams

Recency

INPUT: {(v,t)}

memorise when articles had been created

OUTPUT: [v_1, v_2, v_3, …] ordered by decreasing creation date

Assumption: visitors enjoy reading the most recently created articles.

A. Lommatzsch and S. Albayrak
"*Real-Time Recommendations for User-Item Streams*" in Symposium on Applied Computing 2015, Salamanca, Spain, 2015.

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, ...] 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, ...] 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, …] ordered by similarity to previously read articles

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

A. Lommatzsch and S. Albayrak
"*Real-Time Recommendations for User-Item Streams*" in Symposium on Applied Computing 2015, Salamanca, Spain, 2015.

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, …] ordered by predicted preferences

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

A. Lommatzsch and S. Albayrak
"*Real-Time Recommendations for User-Item Streams*" in Symposium on Applied Computing 2015, Salamanca, Spain, 2015.

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

D. Tikk, R. Turrin, M. Larson, D. Zibriczky, D. Malagoli, A. Said, A. Lommatzsch, V. Gál, and S. Székely
"*Comparative Evaluation of Recommendation Systems for Digital Media*," in Proceedings of the IBC conference 2014, Amsterdam, Netherlands, 2014.

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

D. Tikk, R. Turrin, M. Larson, D. Zibriczky, D. Malagoli, A. Said, A. Lommatzsch, V. Gál, and S. Székely
"*Comparative Evaluation of Recommendation Systems for Digital Media*," in Proceedings of the IBC conference 2014, Amsterdam, Netherlands, 2014.

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**
- Reproducibility
- Real world scenario
- 3D evaluation
- Streams of data

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

- 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?

A. Said and A. Bellogín
"*Comparative Recommender System Evaluation: benchmarking recommendation frameworks*," in Proceedings of the RecSys conference, ACM 2014.

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.

D. Loiacono, A. Lommatzsch and R. Turrin
"*An analysis of the 2014 RecSys Challenge*" in Proceedings of the RecSys Challenge, ACM RecSys 2014.

Why another framework?

Does the recommendation algorithm fit into my production environment?

- Language Agnostic
- Reproducibility
- **Real world scenario**
- 3D evaluation
- Streams of data

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

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

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

B.Kille, A.Lommatzsch, R.Turrin, A.Sereny, M.Larson, T.Brodt, J.Seiler, F.Hopfgartner
“*Stream-Based Recommendations: Online and Offline Evaluation as a Service*” in Proceedings of the 6th International Conference of the CLEF Initiative, Springer Verlag, 2015.

Why another framework?

What's beyond recommendation quality?

- Language Agnostic
- Reproducibility
- Real world scenario
- **3D evaluation**
- Streams of data

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

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

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

Why another framework?

How to treat real scenarios where data are continuously streamed?

- Language Agnostic
- Reproducibility
- Real world scenario
- 3D evaluation
- **Streams of data**

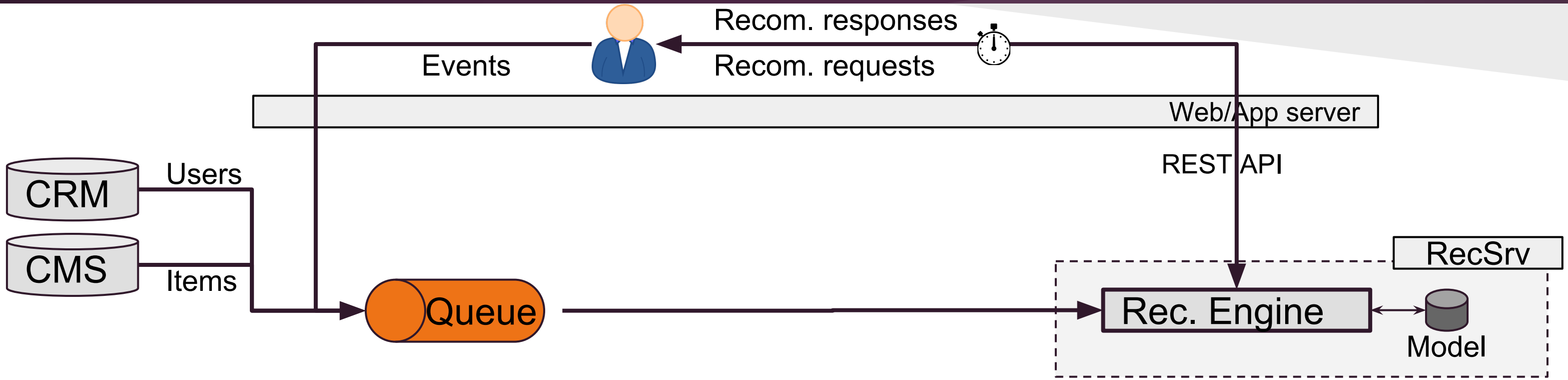
Recommendations should react to users events in real-time

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

- 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

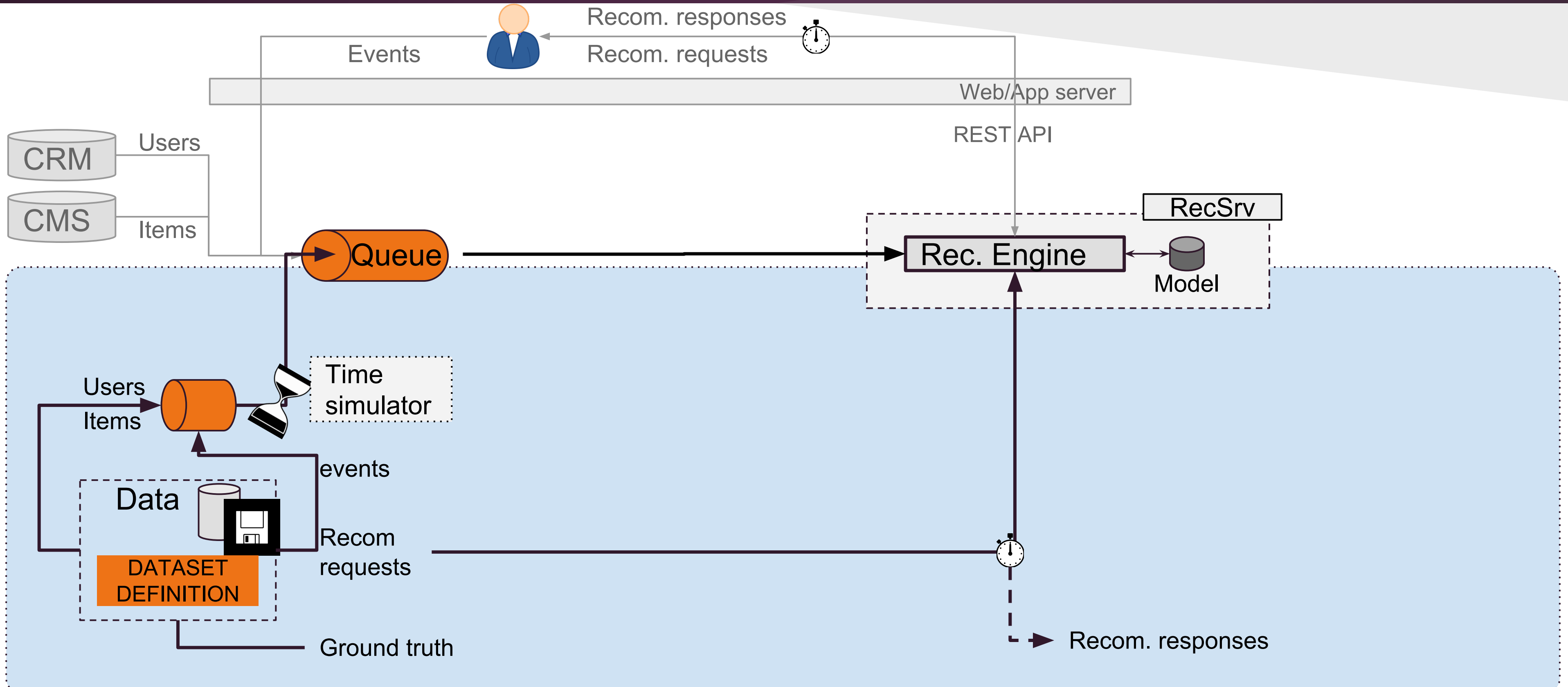
B.Kille, A.Lommatzsch, R.Turrin, A.Sereny, M.Larson, T.Brodt, J.Seiler, F.Hopfgartner
“*Stream-Based Recommendations: Online and Offline Evaluation as a Service*” in Proceedings of the 6th International Conference of the CLEF Initiative, Springer Verlag, 2015.

Workflow: real system

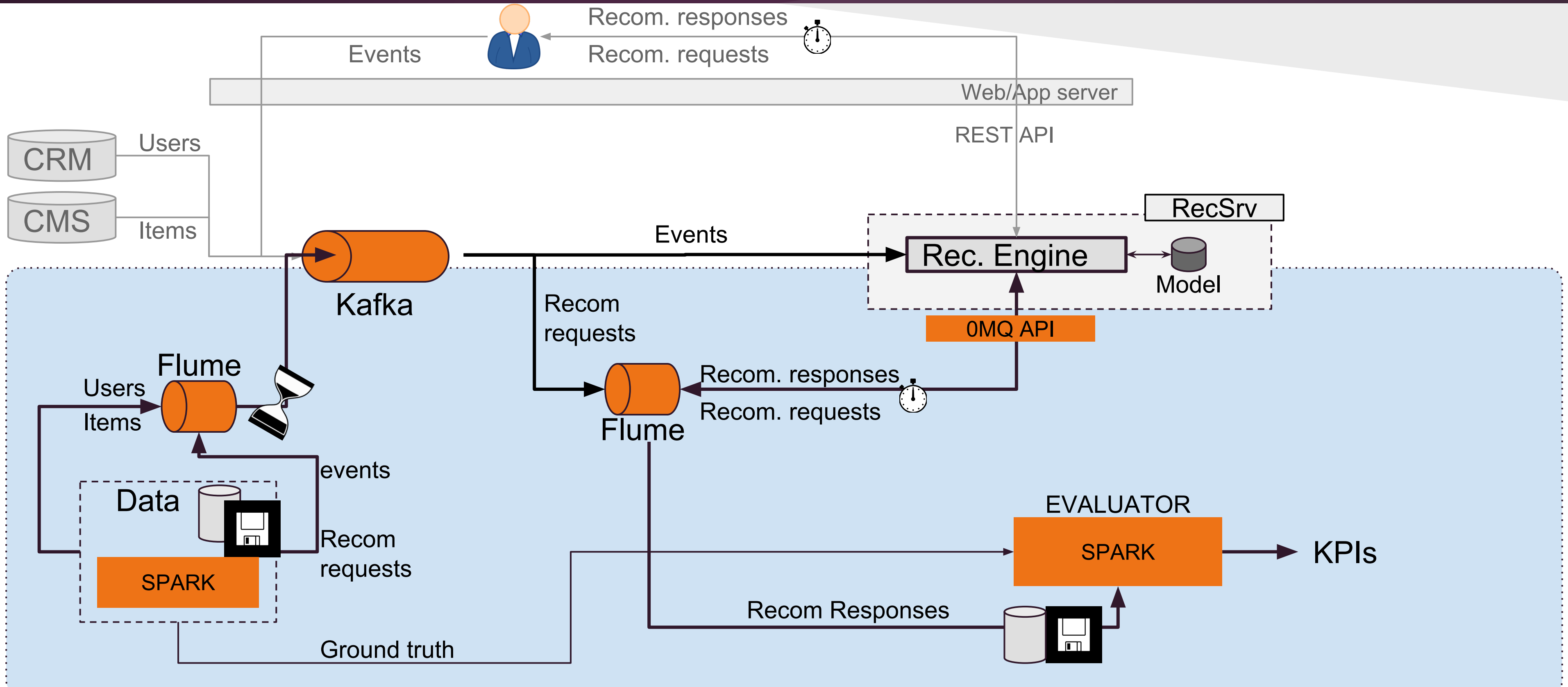


A. Said, B. Loni, R. Turrin and A. Lommatzsch
"*An extended Data Model Format For Composite Recommendation*" in Proceedings of the 8th RecSys conference, ACM RecSys 2014.

Workflow: Idomaar (1)



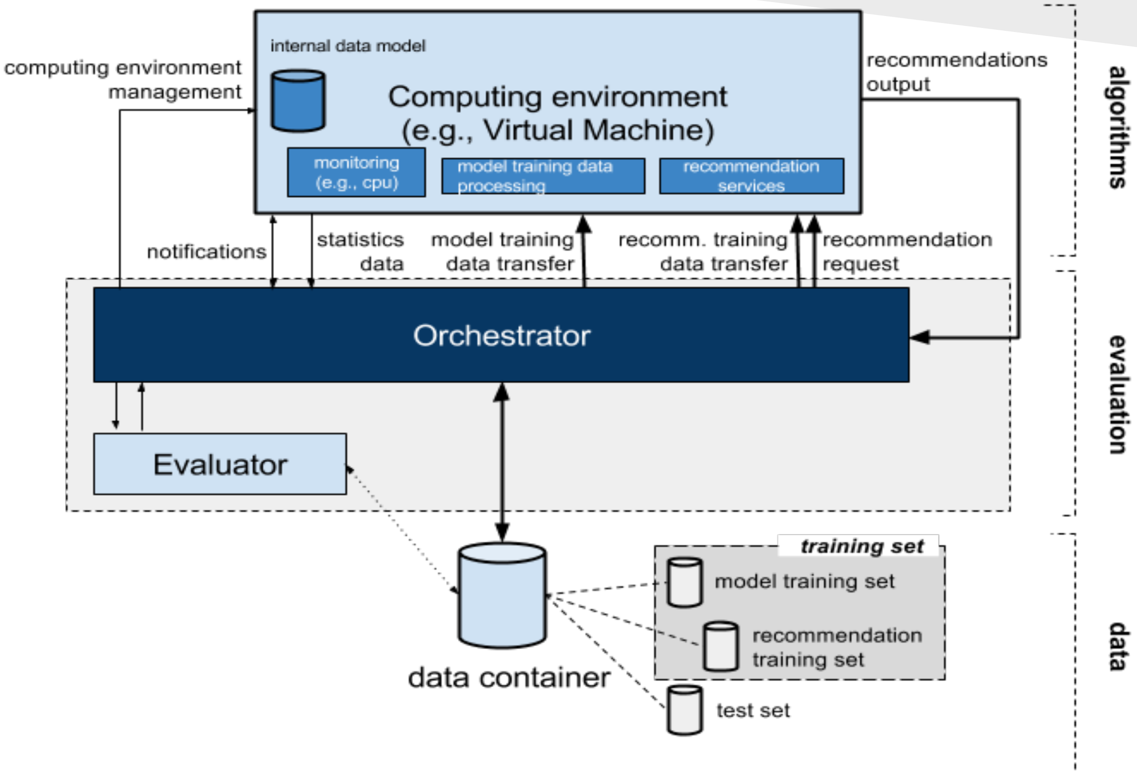
Workflow: Idomaar (2)



Architecture (1)

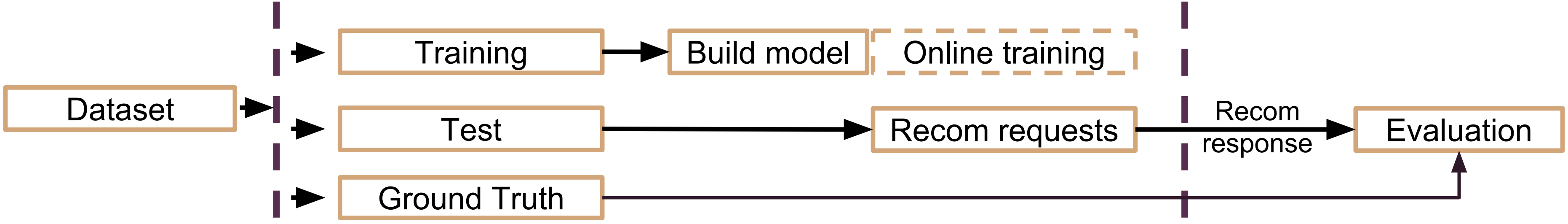
the **algorithms** to test, both state-of-the-art algorithms and new solutions implemented within the CrowdRec project	**Algorithms**
the **evaluation logic**, experimenting with the available algorithms to compute both quality (e.g., RMSE, recall) and system (e.g., execution and response time) metrics.	**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

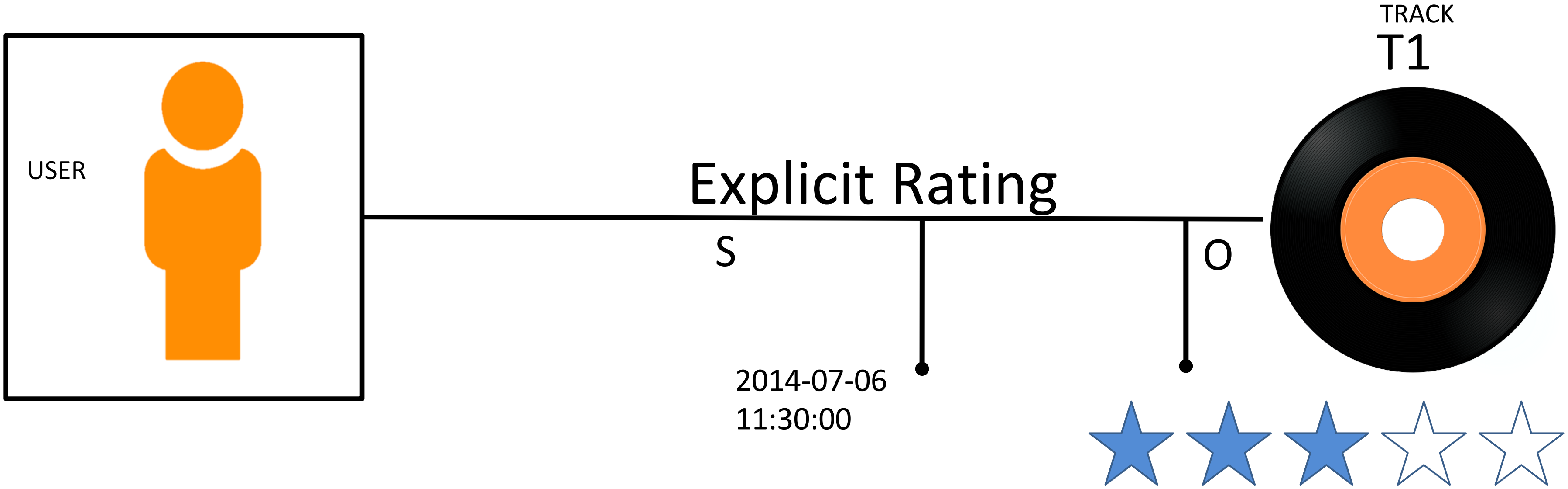
ts: 1363245118

properties: {"rating":9}

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

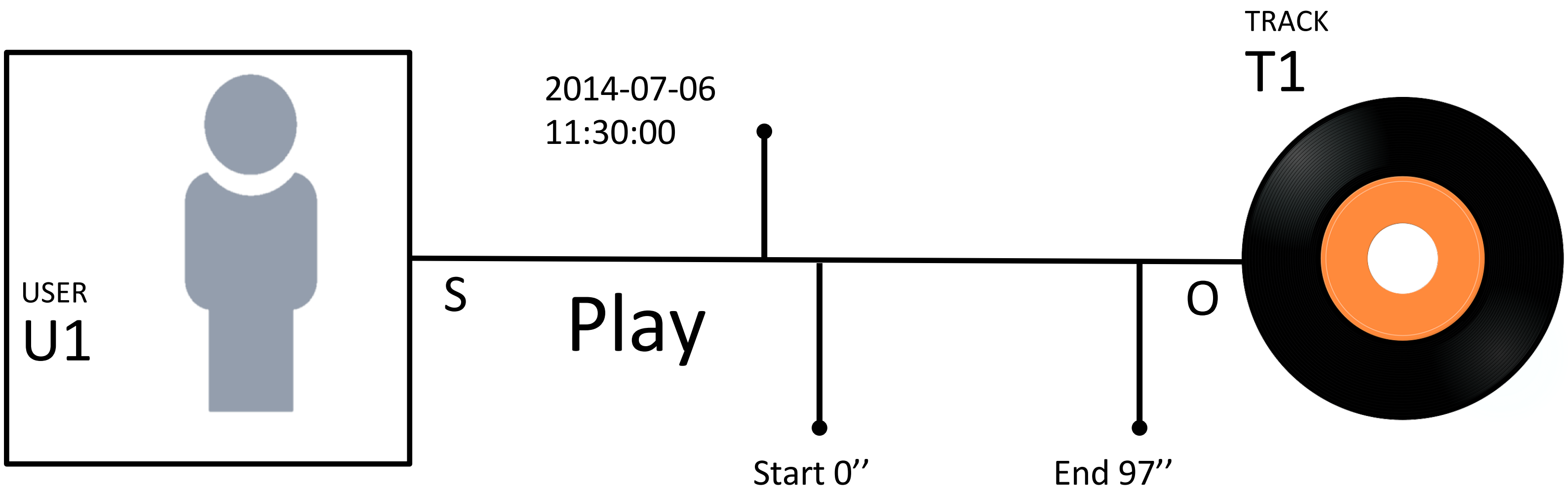
Data

Explicit Rating

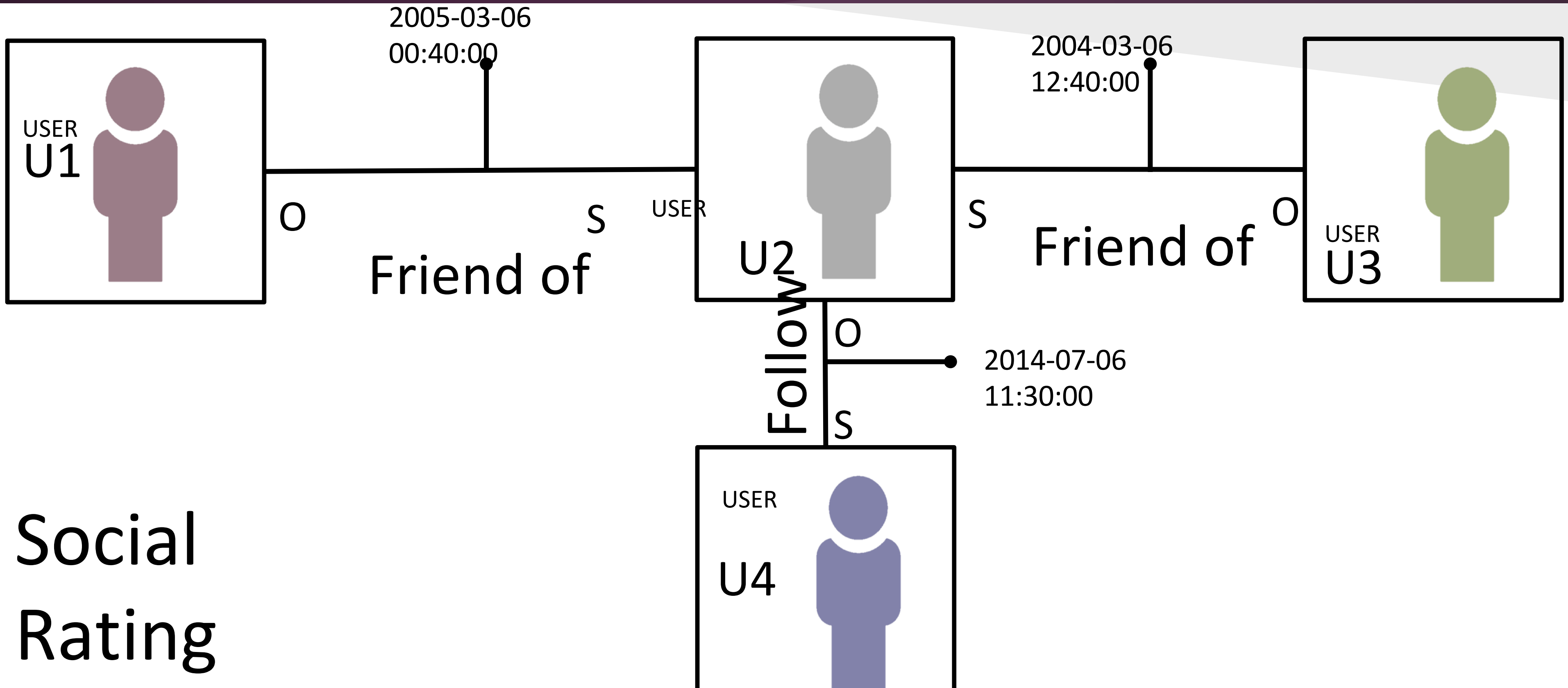


Data

Implicit Rating



Data

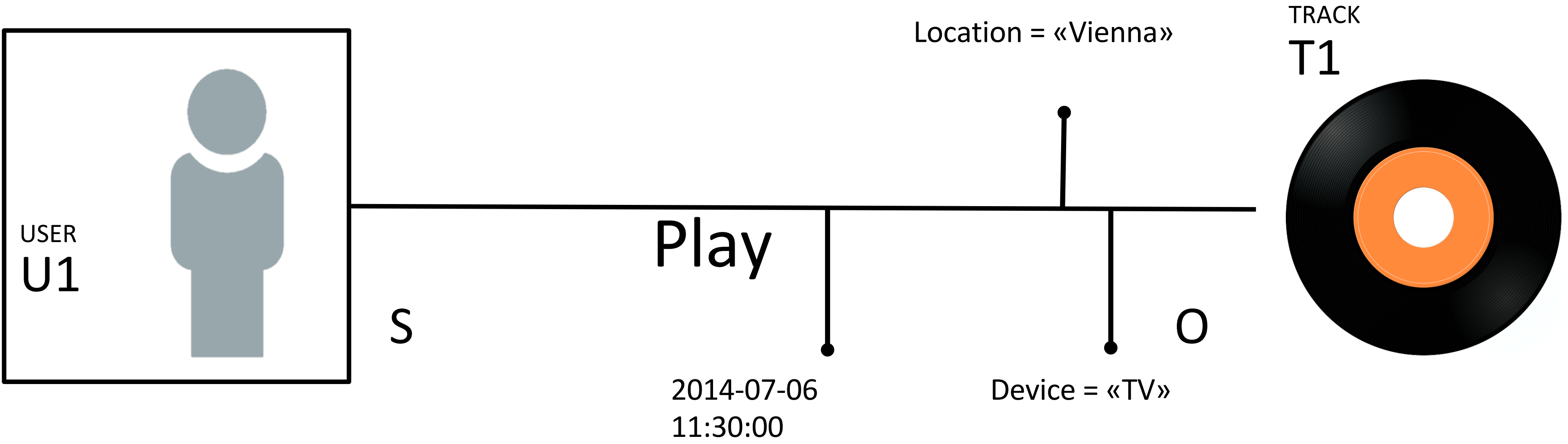


Social

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

puppet labs

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

Lightning-fast cluster computing

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/01.linux/01.centos/01.mahout*
5. Implement the interface with zeromq or http protocol following template from mahout machine (zeromq) or Clef machine (http, 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/01.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
- **Flume**: flume.apache.org
- **Spark**: spark.apache.org
- **Vagrant**: 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”

plista

plista who?

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

recommendation platform based on

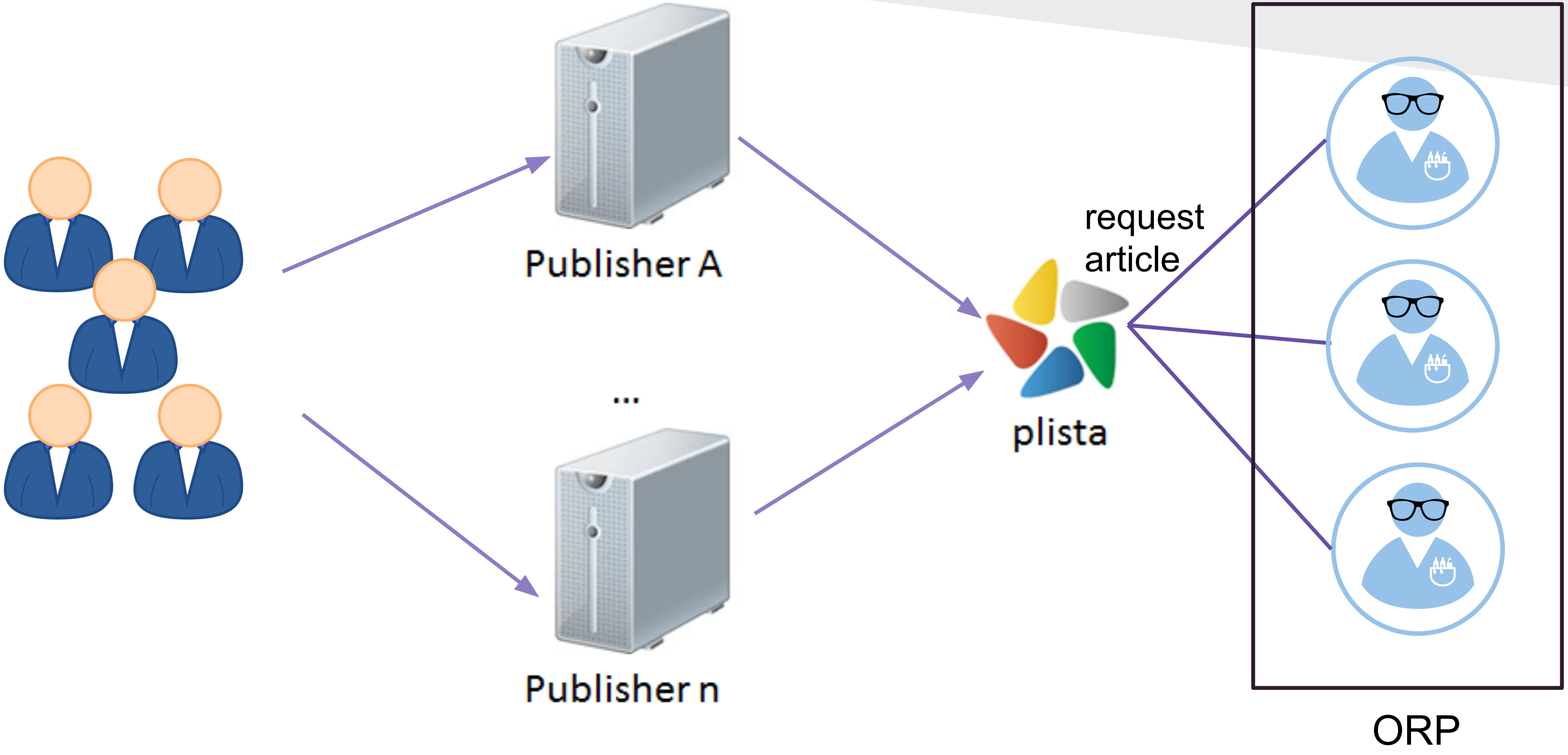
- Ensemble of unrelated, uncoupled algorithms
- Message bus to tie the architecture together

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

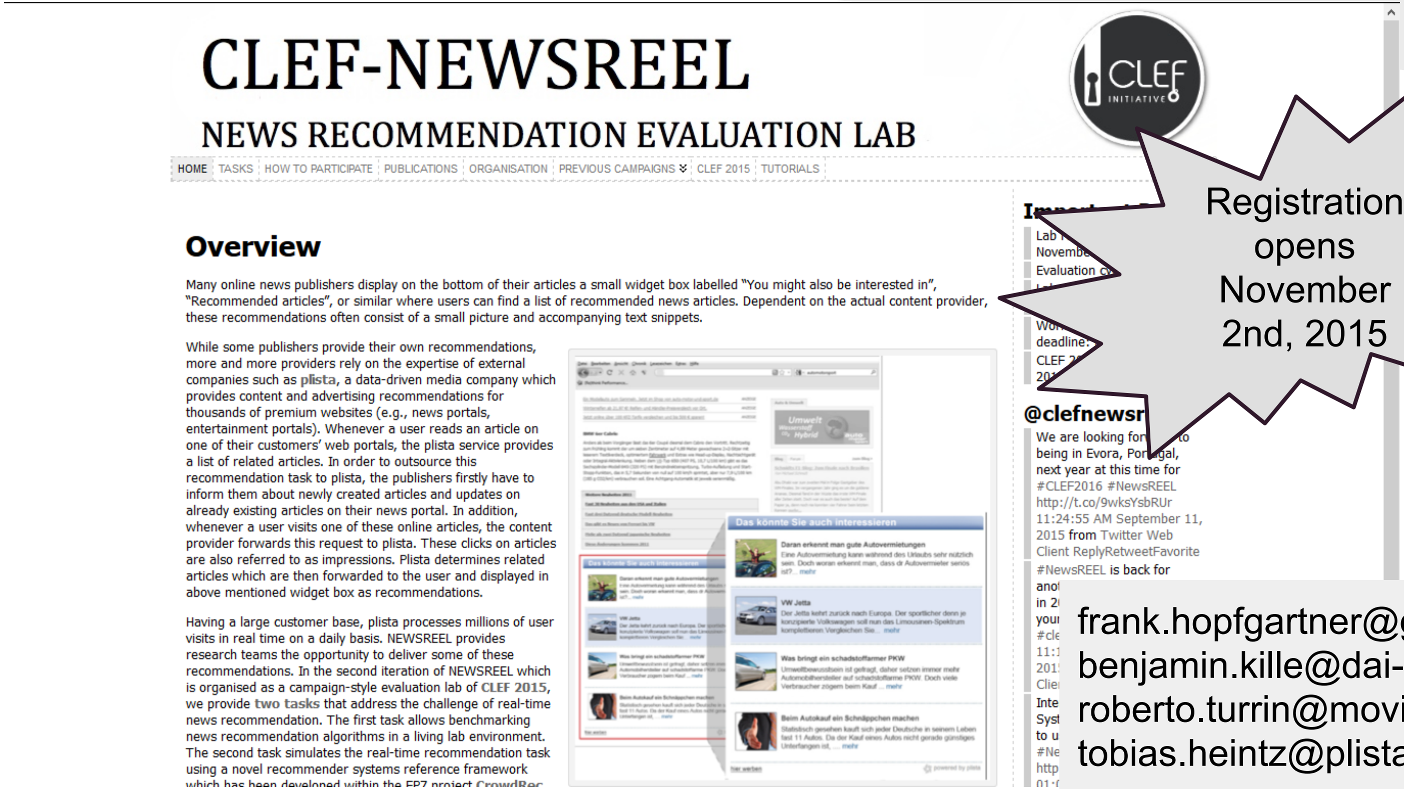
ORP - What kind of data?

- 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":{"26":[],"8":[18841,18842],"11":[81565]},"clusters":{"51":{"2":255},"65":{"1":255},"64":{"2":255},"66":{"8":255}}},"recs":{"ints":{"3":[238088342,238718487,238804226,225938570,232378516]}},"timestamp":1442388373786}

Concluding…

www.clef-newsreel.org



Registration opens November 2nd, 2015

frank.hopfgartner@glasgow.ac.uk
benjamin.kille@dai-labor.de
roberto.turrin@moviri.com
tobias.heintz@plista.com