Examples

Software Architecture VO/KU (706.706/706.707)

Roman Kern

Version 1.1.0

Institute for Interactive Systems and Data Science, TU Graz

Outline

Example Pattern - Disruptor

Example Project - wissen.de

Example Project - EEXCESS

Example Pattern - Disruptor

Basic Problem

Concurrency

- Two or more tasks happen in parallel
  - ... and also contend on access to resources
- Where resources might be files, database access, memory, ...
- The two main concepts here:
  - Mutual exclusion - manage access to resource
  - Visibility of change - controlling, when changes become visible to other threads

Basic Problem

Mutual Exclusion

- Is typically achieved via **locking**
- … but locks are expensive
- and require support by the operating system
- Some platforms support **CAS** (Compare and Swap)
- … far less expensive
- but will only work for words and within a single machine

Note: Developing concurrent programs that make use of lock is hard, developing programs that integrate CAS is extremely hard

Basic Problem

- Ideally, there would be a single thread for all writes
- … and all other thread just reading the results
- Today, CPUs are multi-core and free to conduct out-of-order execution
- … therefore the reading/writing needs to be coordinated
- by the use of **memory barriers**

Queue

Queue Pattern

- Pattern to decouple **producers** from **consumers**
- Producers write into a queue (head)
- Consumers read out of the queue (tail)
- If the consumers handle more items that the producers generate
- … the queue will be empty and the system inefficient
- If the producers generate more items than the consumers can handle
- … the (unbounded) queue will explode

Note: Typically queue do not work well with CAS and other properties of modern architecture (cache lines)

Pipeline

Pipeline Pattern

- Pattern to decouple a series of **data transformation steps**
- Data is passed from one **filter** to another through **pipes**
- In a simple case the data is passed in a series of transformations
- … in a more complex case there will be branches and parallel transformations
- In between the transformations there will typically be queues

Note: A mixture of queue and pipeline are common for complex systems

Disruptor Motivation

Problem Setting

- Many incoming events
- Process in parallel
- Maximise resource utilisation
- Maintain sequence of events

Disruptor Motivation

Basic Structure

- Ring buffer (instead of a queue) with index
- Producers populate the buffer with items
- Consumers take out items from the buffer
- … where multiple consumers may process an item in a sequence
- Optimisations: Preallocation of items, size of ring buffer 2^n

Disruptor Motivation

Separation of Concerns

- Storage of items (being exchanged)
- Coordination of producers (claiming the next sequence for exchange)
- Coordination of consumers (being notified that a new item is available)

Disruptor Motivation

Barriers

- Producer barrier
- … items are stored in the correct sequence
- And an additional claim strategy
- … which decides what producers should should do (e.g. blocked, busy wait)
- Consumer barrier
- … consumers take out items in correct sequence
- With an additional waiting strategy

Disruptor

Disruptor Conclusion

- Complex implementation
- Easy to use
- At least one order of magnitude faster than e.g. an ArrayBlockingQueue
- Improved latency
- Plays well for garbage collectors

Example Project - wissen.de

Wissen.de - Host

- Wissen.de is a web-site hosted by **wissenmedia**
- Wissenmedia is owned by Bertelsmann SE & Co. KGaA
- Wissenmedia owns brands: Brockhaus, Bertelsmann, WAHRIG, CHRONIK, JollyBooks
- The brand Brockhaus is over 200 years old and is known by 93% people (in Germany)

Wissen.de - Scope

- Wissen.de is a **free service**
- Content is added and curated by **editors**
- Does not follow the Wikipedia model
- Free content is not taken from Brockhaus
- wissen.de articles differ from printed articles
- In their style and their life-cycle

Wissen.de - Motivation

The project started out as an **innovation platform**:

- Be innovative in terms of **business models**
- Wissen.de is just a single portal to a complex system
- → Another example is a cooperation with a set-top box manufacturer
- Be innovative in terms of **technologies**
- → Try out new functionality

Wissen.de - Software Architecture

In terms of software architecture this puts an emphasis an specific quality attributes:

- **Flexibility**
  - quickly try out new features
- **Evolvability**
  - add new features without interfering with existing infrastructure
- **Scalability**
  - need to manage millions of articles (more than the German Wikipedia)
  - need to serve many users

Wissen.de - Software Architecture

Focus on specific quality attributes has implications on others:

- **Configurability**
  - Need to be high as well
- **Testability**
  - Suffers, as the system is changing at a high pace

Wissen.de - Software Architecture

- More importantly, the architecture needs to be flexible
- And foresee possible directions
- Typically use YAGNI (“You ain’t gonna need it”) - as a guideline
- **Complexity**
  - The system has a high level of complexity
  - → very hard for new developers in the project

Wissen.de - Software Architecture

Role of the software architect

- Identify the main use cases
- Derive requirements from the use cases
  - In terms of functionality
  - In terms of not directly functional requirements
  - ... identify quality attributes
- Assess risks in the project
- Communication with project partners (iterate, document)
- Decide on programming language, frameworks, ...
- Decide on actual architecture (e.g. patterns, (a)synchronous, ...)
- Plan the development of the individual aspects (project manager)

Wissen.de - Software Architecture

High flexibility is achieved by

- **Loose coupling**
  - Individual components do not depend on other components
- **Generic interfaces and protocols**
  - Thus components can be easily swapped out and replaced
- But this have an impact on:
- **Performance**
  - System needs to be as generic as possible
  - → no option to fine-tune algorithms

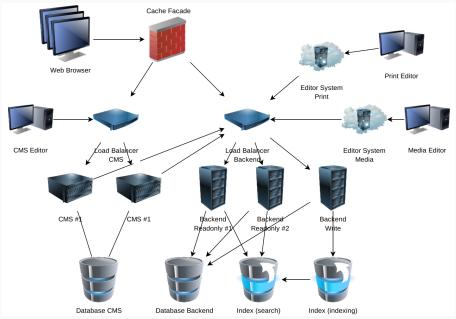
Wissen.de - Backend

- Wissen.de is only one of multiple web sites
- The whole infrastructure contains many sub-systems and components
- Another part is the interface to the other systems (e.g. editor systems)
- It is embedded into an existing landscape of tools → integrability

Wissen.de - Team

- Developed by separate teams
- Teams are from **different companies**
- Know-Center, Key-Tec, EDELWEISS72, wissenmedia, arvato, Nionex...
- Teams are **geographically dispersed**
- Graz, Munich, Gütersloh

Wissen.de - Overview



Wissen.de - Detail

- Will focus on the backend part only
- It is run on multiple (virtual) machines
- Used by multiple components
- Main tasks: Store articles, index articles, present articles

Articles

- Articles are stored as XML
- Combination of data and meta-data
- Meta-data are title, date, category, ...
- Data is XML, not restricted to a single format
- Links between articles

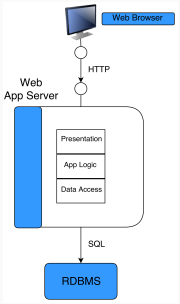
Main Architecture

- **Web service** as main interface
- → Client-server architecture
- Main architecture: **n-tier style**
- Typical example for a heterogeneous architecture style

n-Tier Architecture

- **Conceptual architecture**: 3-tier
- Database layer
- Application logic layer
- Presentation layer

Architecture: 3-layer applications



n-Tier Architecture

- **Implementation architecture**: 2-tier
- Framework library
- Presentation libraries
- E.g. web-service library, command line library

Database Layer

- Object-relational mapping (ORM)
- No direct interaction with the relational database
- Schema can be derived from the business objects

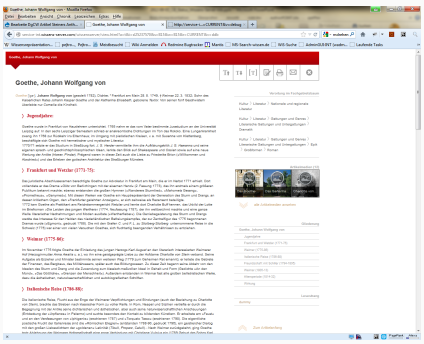
Example of ORM

```
@Entity
@Table(name = "ARTIFACT")
@NamedQueries( { @NamedQuery(name = "artifactById", query =
 "SELECT x FROM ARTIFACT x WHERE x.ARTIFACT_ID = :artifactIdParam AND x.BOOK_ID = :bookIdParam")
})
@XmlJavaTypeAdapter(XmlArticle.Adapter.class)
public class Artifact implements Serializable {
    @EmbeddedId
    private ArtifactPK              id;

    @ManyToOne
    @MapsId
    @JoinColumn(name = "BOOK_ID", referencedColumnName = "BOOK_ID")
    private Book                    book;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "CONTENT_ID", referencedColumnName = "CONTENT_ID", nullable = false)
    private Content                 content;
```

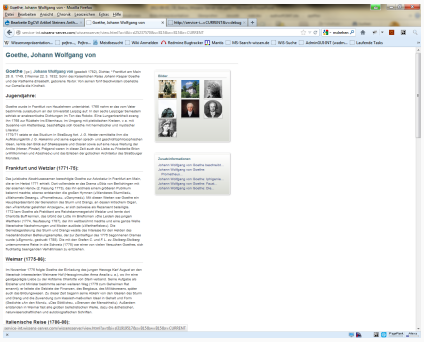

Presentation Layer

- XSLT scripts to transform the output into the target media
- Not only articles are transformed
- E.g. search results, error messages
- Different output target media
- E.g. mobile version, version for set-top boxes, product specific renderings

Presentation Layer



Presentation Layer



Example of XSLT

```
<xsl:stylesheet version="2.0" xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xpath-default-namespace="http://www.w3.org/1999/xhtml">

<xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml">
        <head>
            <link type="text/css" rel="stylesheet" href="{$url-prefix}static/{$version}/{$template-dir}/hilfe-texte/web/default.css" />
        </head>
        <body>
            <div class="artikel-inhalt">
                <xsl:copy-of select="$textbaustein-header"/>

                <!-- Hier wird der eigentliche Artikel gerendert -->
                <xsl:apply-templates />

                <!-- Inhaltsuebersicht fuer ge-chunkte Artikel -->
                <xsl:if test="not(/*[1]/ws:kontext/ws:ws-intern) and //chunk">
                    <xsl:call-template name="chunks-table"/>
                </xsl:if>
```


Web Service Interface

- Multiple interfaces, for different use cases (e.g. read-only access, administrative access, ...)
- Stateless
- Hybrid of REST and RPC style service
- Output is either XML or JSON

Information Extraction - Preprocessing

- Task: Transform an **XML into a textual representation**
- Three stages:
- Input XML
- → transformed into XHTML
- → transformed into plain text

Information Extraction - Preprocessing

- Style: Pipeline
- Batch-sequential, the next filter starts once the previous has finished
- The output of the previous filter is the input to the next

Pipes and filters

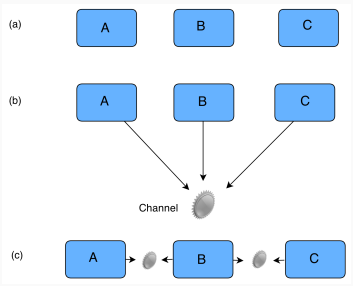


Figure 1: Pipe and filters style

Information Extraction - Execute

- Task: **Extract information** out of text
- Multiple sub-tasks:
  - Split the text into sentences
  - Split a sentence into token (words)
  - Mark certain words as stop-words (should be ignored)
  - Assign word groups to individual tokens
  - Detect named entities (E.g. person names)

Information Extraction - Execute

- Realisation: **Pipeline with shared repository**
- First the text is filled into a special data-structure
- Each filter (sentence chunker, stop-word detection, ...) modifies the data-structure
- Using so called annotations
- Each annotation is a span (start, end) with addition features
- Caveat: filters depend on the output of preceding filters

Example of Information Extraction Pipeline

```
public List<ExtractedInformationAnnotation> process(String text) throws InformationExtractionException {
    AnnotatedDocument doc = new DefaultDocument();
    doc.setText(text);
    for (Annotator annotator : annotators) {
        annotator.annotate(doc);
    }
```

Event Framework

- Components can register to listen for events
- Components can trigger events
- Typically all events should be handled asynchronously (the sender is not blocked)
- Architectural style: **publish-and-subscribe**

Notification Architectures

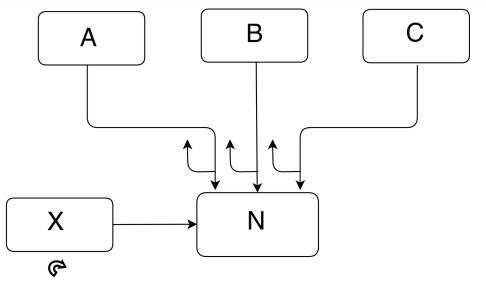


Figure 2: Notification architecture

Example of an Event Listener

```
eventListener = new EventListener() {
    @Override
    public void onEvent(Event event, Task task) {
        if (event instanceof MediaAddedEvent) {
            isDirty = true;
        }
    }
};

eventManager.registerEventListener(eventListener);

// somewhere else
eventManager.fireEventAsync(new MediaAddedEvent(name));
```


Cluster Communication

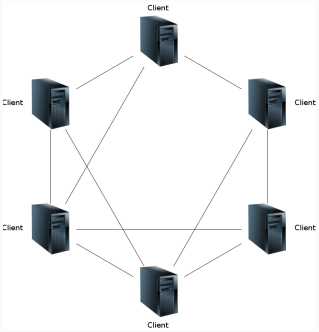
- Need to **scale out** (horizontally) to cope with the demand
- Add redundancy to increase the **availability**
- $\rightarrow$ instead of a single machine, have a cluster of machines
- Works transparently with the event framework

Cluster Communication

- Dynamically detect all cluster members on start-up (discovery)
- Communication is based on either broadcast/multicasts (UDP) or direct communication (TCP)
- All cluster nodes need to know each other
- Architectural style: **peer-to-peer**

47

Peer to peer



48

Cluster Communication - Synchronous

- Only asynchronous communication facilities
- Create synchronous communication via callbacks
- Each synchronous message contains a unique id and sent asynchronously
- Once the message has been processed by the remote note, a notification is sent back passing the id
- Processing then can be continued at the sender side

49

Indexing

- The **search index needs to be updated** once articles have been changed
- The component responsible to update the content of articles fires an event as soon as an article has changed
- The index components listens for these events
- → Decoupling of components, as one component does not know the other components
- Disadvantage: no direct control of the process flow, hard to track the progress of operations

50

Request Tracking

- **Track long running operations**
- For example: batch import of articles, which might take hours
- Idea: collect all information regarding an operation in one place, called **task**
- Store this information in the database
- Notify user once the operation is done

Request Tracking - Task

- Task consists of
- ID: Unique ID of the task
- Status: running, finished
- Result: success, failed, cancelled
- Messages: List of messages for the user
- Attributes: Track the progress (→ progress bar in the UI)
- Properties: Store internal state information

Request Tracking - Task

- A single task might spawn multiple machines
- Synchronisation via the database
- Administration console list all tasks
- Helps to detect the root of problems

Logging

- **Common logging infrastructure**
- Logging is also collected in the tasks
- Logging output also contains the task-id
- Log output is collected in files
- Log files are rotated

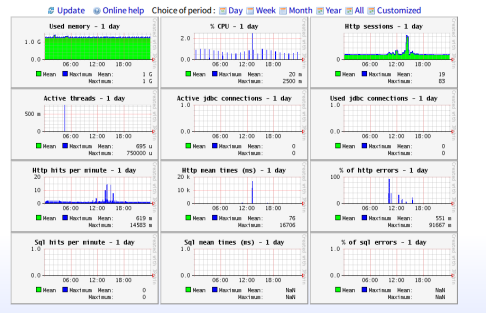
Error Handling

- Each layer produces its own **type system of errors**
- The presentation layer is responsible to report the error to the user
- For each error an unique ID is generated
- The ID is reported to the user and logged
- Thus no internal state is reported to the outside

Monitoring

- **Monitor the current state of the system**
- Web-based tool to monitor the state
- Current resource consumption, e.g. memory used
- List of recent error logs
- Support of administrative/analysis operations

Monitoring



Runtime Performance

- **Improve performance by use of caching**
- Caches need to be in-sync across the multiple machines
- Therefore all changes need to be reported to all machines
- The event framework propagates these changes to all nodes and components
- Changes in the file-system need to be detected as well

Improve Flexibility

- **Improve flexibility by increase of configurability**
- Level of configurability rises with the power of the configuration language
- Highest level if the configuration itself is some sort of programming language
- → **Interpreter** architectural style

Support Infrastructure

- Version control system
- Bug/Issue tracking system
- Continuous integration system
- Documentation system

Rollout

- **Development system**
- Virtual machine with all tools installed
- **Staging system**
- Replica of the production system
- **Production system**
- Only versions are deployed on the production system, which have been tested on the staging system
- Only a few people are allowed to deploy on the production system

Project Management

- Agile project development
- Short cycles, working software
- Project communication via periodic conference calls
- Additionally e-mail and via issue tracker

Example Project - EEXCESS

EEXCESS - Overview

EU Research Project (funded by the EU)

- "EEXCESS is a research project which aims to make cultural, scientific and educational content easily accessible and available"
- Runs over multiple years, started in 2013
- High number of involved partners:
- Austria (3), Germany (3), UK (2), France (1), Switzerland (1)
- Different types of partners
- Technology partner, data providing partners, test bed partners, associated partners

EEXCESS - Overview

EU projects basics

- Base is a document, the description of work (DoW)
- This document outlines the project
- Typically organised in work packages
- Tracked via milestones and deliverables
- The project gets reviewed by the EC

EEXCESS - Stakeholders

Stakeholders:

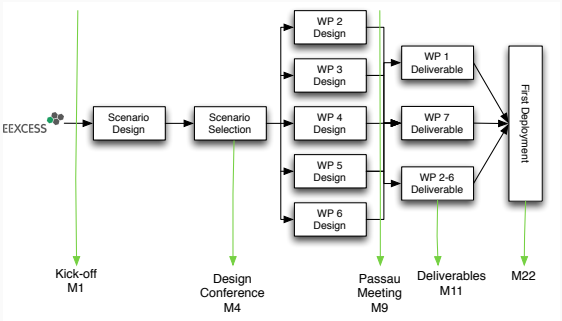
- European Commission
- Partners (scientific and commercial interests)
- Main problem: No clearly defined goal, but many different ideas

EEXCESS - Approach

Procedure:

- Demo scenario (to create a common understanding)
- Ask partners for scenarios (got 25 scenarios)
- Agree on use cases (4 scenarios)
- Ask partners for functional requirements (non-technical)
- Priorities and risks for requirements
- Ask the work package leaders for requirements (technical)
- Develop initial architecture

EEXCESS - Timeline Y1

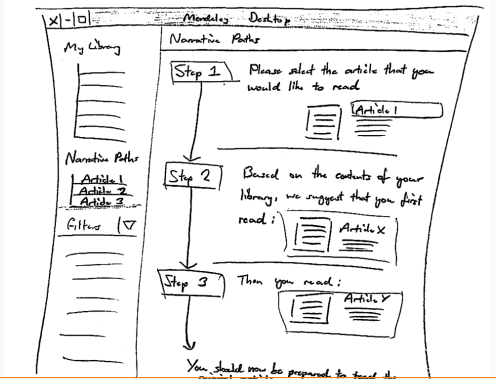


EEXCESS - Scenarios

Use case:

- Includes a persona
- Description of the scenario
- Mock-ups
- Relation to the project

EEXCESS - Example MockUp



EEXCESS - Scenario Analysis

ID	short description	scenario type	support type	recommended data	injection technology	language	target group	type of user context	platform type
B1	preparing lecture in e-learning system	content creation	educational support	cultural and scholarly data	easywiki plugin (SITOS)	german	teachers	explicit + implicit	web
B2	writing discussion board entry in e-learning system	content creation	educational support	websites, maps, pictures	browser plugin	german	pupils	explicit + implicit	web
B3	improving e-learning lecture	content creation	educational support	cultural and scholarly data	browser plugin	german	teachers	explicit + implicit	web
B4	recommendation of additional learning material	content consumption	educational support	websites	browser plugin	german	pupils	explicit + implicit	web
K1	wikipedia extension with museum objects	content consumption	educational support	local museum objects	browser plugin	german	teachers	implicit	web
K2	search in local library catalogue	content consumption	scholarly communica tion	data set information, related exhibitions, related institutions	browser plugin, server side deployment	german	scientists	implicit	web
K3	lecture preparation with Wikipedia	content consumption	educational support	local events, local museum objects	browser plugin	german	teachers	implicit	web
K4	Catalogue Maintenance in a Museum	content analysis	general public education	metadata of relevant resources	imdas plugin	german	scientists	implicit	web
M1	writing grant proposal	content consumption	scholarly communica tion	scientific papers	desktop tool, word-processor plugin	english	scientists	explicit + implicit	desktop
M2	understanding topics with narrative paths	content consumption	educational support or scholarly communica tion	scientific papers	mendeley desktop plugin	english	scientists	explicit + implicit	web
M3	what's new dashboard	content consumption	scholarly communica tion	scientific papers, blogs	mendeley desktop plugin, mendeley website plugin	english	scientists	explicit + implicit	mobile
P1	blog extension with scientific resources	content consumption	educational support	scientific papers	browser plugin	english	students	implicit	web
P2	twitter feed generation	content consumption	scholarly communica tion	twitter feeds	twitter bot	english	scientists	implicit	web
P3	bookmarking web app	content consumption & content creation	scholarly communica tion	bookmark snippets of scientific papers, web pages	web app + browser plugin	english	scientists	implicit	web
W1	preparing golf-club speech	content consumption	general public education	cultural and scholarly data	cms plugin	german	general public	implicit	web
W2	authoring encyclopedic	content	general	scientific papers	cms plugin	german	general	implicit	web

EEXCESS - Scenario Risk Identification

ID	short description	scenario type	injection technology	estimated difficulty from WP view				
				WP2	WP3	WP4	WP5	WP6
B1	preparing lecture in e-learning system	content creation	easywiki plugin (SITOS)	medium	medium		medium	
B2	writing discussion board entry in e-learning system	content creation	e-learning software plugin?	high	medium		medium	
B3	improving e-learning lecture	content creation	e-learning software plugin?	medium	?		medium	
B4	recommendation of additional learning material	content consumption	e-learning software plugin?	medium	medium		medium	
K1	wikipedia extension with museum objects	content consumption	browser plugin	low	low		low	
K2	search in local library catalogue	content consumption	browser plugin, direct access from library search service to eexcess recommender	low	?		low	
K3	lecture preparation with Wikipedia	content consumption	browser plugin	low	medium		low	
K4	Catalogue Maintenance in a Museum	content analysis	imdas plugin	medium	high		medium	
M1	writing grant proposal	content consumption	desktop tool, word-processor plugin	high	high		high	
M2	understanding topics with narrative paths	content consumption	mendeley desktop plugin	medium	high		medium	
M3	what's new dashboard	content consumption	mendeley desktop plugin, mendeley website plugin	high	medium		medium	
P1	blog extension with scientific resources	content consumption	browser plugin	medium	medium		medium	
P2	twitter feed generation	content consumption	twitter bot	low	high		low	
P3	bookmarking web app	content consumption & content creation	web app + browser plugin	low	high		low	
W1	preparing golf-club speech	content consumption	wissenmedia website plugin	medium	medium		medium	
W2	authoring encyclopedic article	content creation	wissenmedia cms plugin	medium	medium		medium	
W3	searching online encyclopedia	content consumption	wissen.de website plugin	high	?		medium	
W4	exploring information on mobile platform	content consumption	contextR web app extension?	high	high		high	
Z1	professor's scientific literature search in econbiz	content consumption	econbiz extension, browser plugin?	medium	low		medium	
Z2	expert suggest in econbiz	content consumption	econbiz extension, browser plugin?	medium	medium		medium	
Z3	bookmark organization for scientific literature	content consumption & content creation	econbiz extension, browser plugin?	medium	high		medium	
Z4	blog entry creation	content creation	wordpress plugin	medium	?		medium	
Z5	twitter listener	0	0	low	?		low	
Z6	library management	0	0	high	?		high	

Entscheidungsgrundlage für WP5
- plugins für CMS (bitmedia, ZBW econbiz), wo wir die sachen reinpfuschen müssen in existierende infrastrukturen --> MEDIUM
- reine browser plugins brauchen wir für die seiten,wo wir keinen zugang haben (wikipedia z.B.) --> LEICHT (im vergleich)
- mobil oder desktop (ms word) ist SCHWER

Entscheidungsgrundlage für WP2
- ausgehend von WP5 Entscheidungen (aufgrund injection technologie)
- Komplexität von Visualisierungen kann von LOW auf MEDIUM oder von MEDIUM auf HIGH ändern, wenn die Vis schwer zu generieren ist

EEXCESS - Requirements (excerpt)

II. Context-related Events

II.1.	Listen on search queries for deriving profiles. Possible sites are Google, Wikipedia etc.	H	L
II.2.	Listen on browsing history and events for deriving profiles. Browsing history will allow to detect web-site changes and browser activities.	H	L
II.3.	Include location information in user profiles/contexts. This information is needed for geo-context queries and visualisations.	M	H
II.4.	Support the learning of topics of interest for topics of short and long term profiles.	H	L
II.5.	Support the automatic learning of expertise levels on the client.	H	H

III. Visualisation

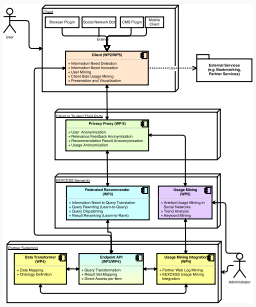
III.1.	Visualise single recommender results and allow enhanced navigation and exploration of those results.	H	L
III.2.	Visualise a set of consecutive recommender results in order to show relatedness among recommender results.	M	M
III.3.	Provide explorative visualizations that facilitate exploration of recommendation results.	H	M
III.4.	Share recommendations with other users to stimulate community based annotations.	L	M
III.5.	Share recommendations within user groups to stimulate community based annotations.	L	M

EEXCESS - Overview

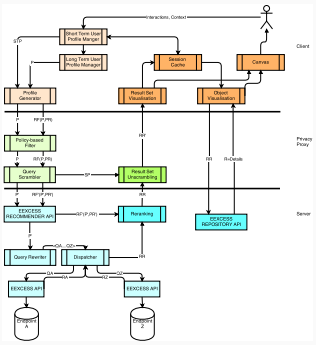
- Initial architecture draft based on scenarios
- Top down: starting with high level goals
- Bottom up: each work package has its own understandings

Note: The first draft of the overall architecture existed before requirements have been made explicit

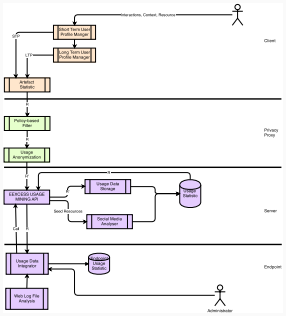
EEXCESS - Architecture Overview



EEXCESS - Architecture Detail - Query



EEXCESS - Architecture Detail - Usage Mining



EEXCESS - Project Management

- Project management by (scientific) coordinator
- Monthly telephone conferences
- Yearly architecture/scenario review
- About four meetings per year (e.g. common hackathon)
- Mailing list for the whole project
- Common content management system for project documentation
- Multiple source code control systems (SVN, Github)
- Common issue tracking system (Jira)

EEXCESS - Lesson Learnt

- Hard to create a common understanding/architecture
- Hard to get the involved partners to be motivated
- Limited project horizon has implications on architecture
- Work package organisation strongly influences architecture

The End