

EDEN

D.4.2.1 Requirements for monitoring and measurement of services

Research Area: EDEN – Ecosystem Design and Engineering
Project Title: EDEN WP4 – User-driven Service Ecosystem Evolution
Document Type: I (Internal)

Document Identifier: FS-EDEN-D421
Document Title: D.4.2.1 - Requirements for monitoring and measurement of services
Editor: Seppo Törmä
Authors: Kari Tuomainen, Keijo Tuomainen, Seppo Törmä
Status / Issue: 1.1
Date Last Change: 31.05.2009
File: FS-EDEN-D421.doc
Delivery Date: 31.05.2009

Document History:

22.05.2009	Document created
27.05.2009	Version 0.1
28.05.2009	Version 0.2
31.05.2009	Version 1.0
25.08.2009	Version 1.1

Flexible Services

Executive summary

The objectives of EDEN Task WP4.2 “Tools for Monitoring and Measuring of Services” is to identify and develop tools to gain visibility into the activities taking place in a service ecosystem. While the goals are generic, the approach is pilot-oriented and aimed to support the overall work in the Flexible Services program.

In subtask 4.2.1 “Requirements for monitoring and measurement of services” the goal is thus to identify the requirements for monitoring and measurement of selected pilot services within Flexible Services program. The focus has turned into the LUCRE service platform and therefore also to the OtaSizzle services utilized in LUCRE platform. Consequently, this report documents the monitoring and measurement requirements identified in OtaSizzle and LUCRE platforms.

The requirements will serve as basis for the subsequent work in WP4.2 dealing with the definition of the architecture for monitoring and measurement of services, development of instrumentation tools for service creators, development of measurement services, experiments with instrumentation tools, and in evaluation of the experiments.

Flexible Services

Table of contents

1 Introduction	4
1.1 Monitoring environment	4
1.2 Monitoring Goals	5
2 System level monitoring	8
3 Application level monitoring	9
4 User level monitoring	12
5 Content level monitoring	13
6 High level monitoring information needs	14
6.1 System metalog	14
6.2 User aggregate log (individual users)	14
6.3 User action log - The action log includes detailed information on the user's actions in the service	14
6.4 Channel log - Channel Log contains contents of each channel	15
7 References	16

Table of figures

Figure 1: Service monitoring overview	3
--	----------

Flexible Services

1 Introduction

The objectives of EDEN Task WP4.2 “Tools for Monitoring and Measuring of Services” is to identify and develop tools to gain visibility into the activities taking place in a service ecosystem. The questions addressed are as follows:

How much are different services used? How active different users are? What kinds of actions users make and what kinds of sequences these actions create? What is the performance of the platform?

There are number of these kinds of questions that will be left totally unanswered without specific tools for gathering, processing, and visualization of data of service execution. Task WP4.2 will address these questions by developing monitoring and measurement tools for service ecosystems.

While the ultimate goals of WP4.2 are generic and aimed for broad range of service ecosystems, the approach adopted in the task is (1) pilot-oriented and (2) aimed to support the overall work in the Flexible Services program. As a pilot environment the focus is on LUCRE service platform and consequently also on the OtaSizzle services utilized in LUCRE platform. The choice can be justified on many grounds. The OtaSizzle platform is about to be launched to a large user population, which makes it realistic to expect large amounts of activity to monitor. Moreover, the LUCRE service platform will provide a variety of service compositions with specific monitoring requirements, e.g., with respect to the availability and integrity of component services.

In subtask 4.2.1 “Requirements for monitoring and measurement of services” the goal has been to identify the requirements for monitoring and measurement of OtaSizzle and LUCRE platforms. The requirements will be addressed in different levels: system, application, user, and content as well as the higher semantic level, as detailed below.

The requirements will serve as basis for the subsequent work in WP4.2 dealing with the definition of the architecture for monitoring and measurement of services, development of instrumentation tools for service creators, development of measurement services, experiments with instrumentation tools, and in evaluation of the experiments.

1.1 LUCRE/OtaSizzle pilot environment

In the LUCRE/OtaSizzle pilot environment monitoring is focused on observing a number of user services, some of which can be composite services consisting of a number of other services.

In the pilot environment the services store their data in OtaSizzle Common Services (COS), which is accessed through a REST API. The services can also have separate application databases, which are not covered by monitoring. COS is focused on serving social networking information and application-specific data has no semantics attached to it from the point of view of COS. COS also integrates infrastructure services such as session handling (single sign-on service) and updating users' location data. Ressi is the database containing the monitoring data made available to the researchers.

Flexible Services

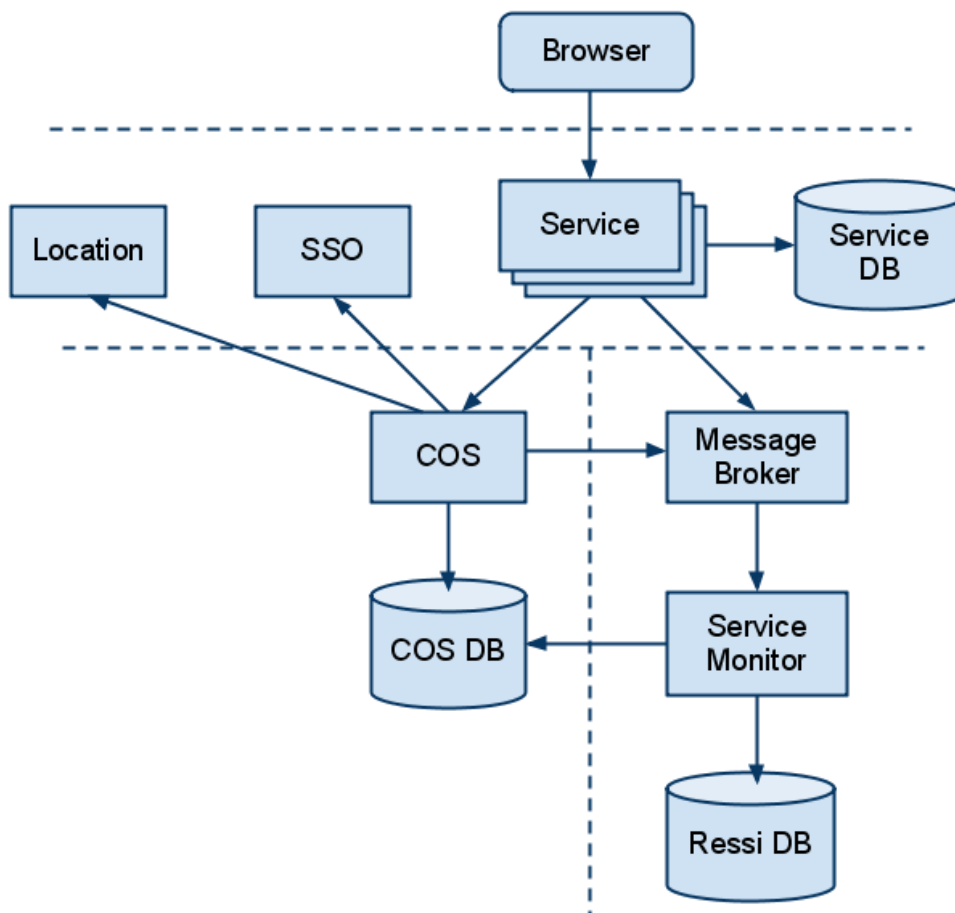


Figure 1: Service monitoring overview

Together the services create a ubiquitous social media platform for urban communities. The three most prominent services build around COS are

- Kassi -- a service enabling community based exchange of goods and services
- Ossi -- a browser-based social networking service enabling communication between users on mobile and web platforms
- Service Composer -- a tool enabling user created composite services

1.2 Monitoring Goals

The goal of monitoring is to create an infrastructure supporting different kind of ways of observing the service ecosystem. The focus areas of monitoring can be divided into four target levels as described in Table

Flexible Services

1.

Table 1: Monitoring target levels

Monitoring target level	Targets
System	Availability, load, response times and other information needed for system administration
Application	Usability issues, application level errors and other information useful in application development
Content	Channel contents, corpus
User	User actions, usage patterns, context information (time, location), social networks (friends, groups, channels)

System level monitoring focuses mainly on supporting administrative tasks and its role in research is limited. It should produce time series data that enables to observe trends in system utilization.

Application level monitoring is closely related to system level monitoring, but refers to monitoring actual functionality of the application, rather than just whether it is running or not. Application usage trends can be used to identify possible usability problems for instance.

The purpose of content monitoring is storing contents flowing through the system for later analysis. Textual content can be utilized for example with corpus-based methods.

User monitoring is considered as having the highest priority. It consists of storing user actions for usage pattern analysis as well as connecting relevant context information to user actions. This information can then be used for various purposes, including:

- Identifying user actions
- Predicting user goals
- Supporting service recommendation/discovery
- Tracking user evolution as a service consumer
- Analyzing social networks
- Reputation management.

Contextual information is an essential part of user monitoring. The context should include at least users *physical location, online location and time*. They are especially important since they link the user actions to the physical world. Other possible context information includes:

- User defined presence information: status (online, offline, busy, away), mood etc.
- Mobile phone activity, for instance active applications, music being played, active calls

Flexible Services

- Environment: lights, sounds, acceleration etc.

Note! Some of the context information is not real-time and may be missing from time to time.

Processing raw data without proper tools is time consuming and error prone, which is why monitoring tools are needed for obtaining a high-level overview of the various subsystems. Monitoring tools should cover the most basic use cases for observing the structure of the various social networks, that is

- Navigating structured data
- Exporting parts of the structured data
- Various visualizations, including:
 - Graphs
 - Time series
 - Dimensionality reduction.

Flexible Services

2 System level monitoring

System monitoring is needed to make sure all services are *available*, that *network connections* work, and there is *enough system resources* to satisfy the non-functional requirements of services. In addition, system monitoring can be used to *predict usage trends*. System monitoring helps to ensure that uptime is high, services answer quickly, and that other parts of service level agreement (SLA) are hold. System monitoring can be also used to perform automatic tasks, like restarting services that have stopper answering to user requests (inoperable).

System level monitoring can be *passive* monitoring, *synthetic* monitoring, or *combination* of both of these. On passive monitoring, real user actions and network traffic statistics are captured and analysed. On synthetic monitoring, monitoring system makes artificial request to the system that is being monitored.

One challenge for system monitoring are service mashups, i.e. composite services built from several existing services, usually implemented and provided by third parties. If one of the services fail or the application interface changes, a composite service that uses it, may also fail. System monitoring can be used to detect failure of composite services to give meaningful notifications to users, and to avoid irrelevant work from other components of the mashup.

Systems monitoring can also be used to detect some unwanted attempts to access or disrupt system. Denial-of-service attacks can be detected by logging network statistics. However, a host-based intrusion detection system is needed to detect that system has been compromised and modified without proper permissions.

There are several free and commercial service monitoring systems available (eg. Zabbix, Nagios). For a comparison of available monitoring systems see Comparison of network monitoring systems [Wikipedia 2009b].

System monitoring solution should support following features:

- Distributed monitoring, to ensure scalability
- Log information as time series
- Monitoring that service is available for users
- Measuring service respond times
- Monitoring network connectivity
- Enable predicting service utilisation trends
- Be extendable using custom scripts
- Sending alerts when error condition is detected

3 Application level monitoring

Application level monitoring refers to monitoring actual functionality of the application, not just whether it is running or not. In practice this means recording all user actions and calculating various metrics based on that. In the context of this project application level monitoring can be divided into three areas:

- Synthetic monitoring
- Passive monitoring
- Web analytics

Synthetic monitoring (or active monitoring) refers to simulating user actions using web browser emulation and storing various metrics in the process. In passive monitoring the same things are being monitored, but based on real user interaction. The most important metrics are response time and availability. The idea is to ensure that the application is working, as it should do. This is not limited to merely testing if the application functions correctly, but also monitoring that non-functional requirements such as response time requirements are met.

Synthetic monitoring focused on testing web application functionality is often referred to as internal or external web testing. The purpose of internal web testing is to monitor that the service works inside the corporate firewall, which is useful for ensuring that the intranet services are available. External web testing refers to testing the application availability outside the firewall and from different service providers. Being able to pinpoint the cause of service disruptions to possible network issues also shortens the time spent on resolving the issues.

Stress testing is analogous to web testing, but instead of using randomized simulations to verify correct functionality it tests how the application behaves under load beyond normal operational capacity. The purpose is to ensure that the service behaves in a predictable fashion under unusually high load stressing computational resources and high concurrency. Stress testing is also an useful tool in preparation for denial of service attacks.

On-site web analytics refers to application level monitoring targeting on-site visitor measurement. It consists of measuring, collecting and analyzing page view data in order to understand how visitors navigate inside an application. The data can be collected by logging user actions on the server (log analysis) or by notifying a monitoring server on page views from the client using JavaScript (page tagging). Commonly used metrics in web analytics are described in Table 2.

Flexible Services

Table 2: Common web analytic metrics [Wikipedia 2009a]

Metric	Description
Hit	A request for a file. Only available in log analysis and gives misleading numbers.
Page view	A request to view a page.
Visit (session)	Denotes a series of request from the same client, identified by the session ID.
First visit	A visit by a client that has no previous visits.
Visitor	An uniquely identified client, counted only once within a defined time period (day, week or month)
Repeat visitor	A visitor that has made at least one previous visit.
New visitor	A visitor that has not made any previous visits.
Impression	Advertisement loaded on a user's screen.
Singletons	Number of visits where only a single page is viewed.
Bounce rate	The percentage of singletons out of all visits.
Exit percentage	The percentage of users who exit from a page.
Visibility time	The time a single page is viewed.
Visit duration	Average amount of time that visitors spend on the site on a visit.
Page view duration (time on page)	Average amount of time that visitors spend on each page.
Page views per session (page depth)	Average number of pages views on a visit.
Frequency	Total number of visits divided by the total number of visitors.
Click path	The sequence of hyperlinks website visitors follow on the site.

Application level monitoring should support the following features:

- Simulating user actions

Flexible Services

- Measuring and collecting response times
- On-site web analytics

Flexible Services

4 User level monitoring

User level monitoring refers to monitoring user activities in the service ecosystem. It differs from other monitoring methods discussed earlier, such as web analytics, in that user identity and extensive contextual data related to the user are recorded.

Each user action should be recorded together with the contextual data available at the time. An action record should include at least the following information:

- User identifier
- Session identifier
- Action number within session
- Application identifier
- Action identifier
- Action source
- Action parameters
- IP Address
- HTTP Referrer [sic]

The context information should include at least the following dimensions:

- Action identifier
- Time
- Physical location, including last modified time
- User status message
- Other user properties at the time (see User aggregate log)

It should be possible to use this information to extract click streams for analyzing user action sequences. This can then be used in identifying user's goals and predicting user's needs. Additionally the context information can include detailed data from mobile phone activity, though this data may not be up-to-date.

User level monitoring also includes tracking social networks and user properties over time. These are described in *High level monitoring information*.

5 Content level monitoring

Content stored through the services, for instance in channels, should be archived for research purposes. Data mining methods can be applied to user input to reveal more semantic information than what can be obtained from click stream or context information. In the scope of this paper, content level monitoring is considered to be part of user level monitoring.

6 High level monitoring information needs

The needs for following high level statistics have been identified by HIIT. New high level log entry should be saved every time the information changes. Each log entry should also contain date and time information. [Lampinen 2009]

6.1 System metalog

- Number of users (gender / age distribution)
- Number of channels
- Number of private and public channels

6.2 User aggregate log - individual users

- Friend list
- Number of friends
- Number for private and public channels created
- Number of private and public channels visible to user
- Number of messages written
- Number of read messages
- Gender
- Time of birth

6.3 User action log

The action log includes detailed information on the user's actions in the service

- Channel actions
 - creating a channel
 - opening a channels for browsing
 - posting message
 - searching for a channel

Flexible Services

- Friend actions
 - inviting a new friend
 - accepting an invitation
 - rejecting an invitation
 - searching for a person
 - viewing friend's profile
 - removing a friend
- Status actions
 - updating status line
 - erasing status line
 - updating profile
- Other actions
 - entering the service
 - exiting the service
 - each time a user enters a new display

6.4 Channel log - Channel Log contains contents of each channel

- Contents of each channel

Flexible Services

7 References

- [Wikipedia 2009a] Web Analytics. Wikipedia, 2009.
Referenced 25.5.2009.
http://en.wikipedia.org/wiki/Web_analytics
- [Wikipedia 2009b] Comparison of network monitoring systems. Wikipedia, 2009.
Reference 25.5.2009.
http://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems
- [Lampinen 2009] A. Lampinen. Ressi -- Researcher Logs. HIIT, 2009.
Emailed 23.3.2009.