

FP7 Project ASAP

Adaptable Scalable Analytics Platform

**D6.4 Dashboard****WP 6 – Information Visualization**

webLyzard technology

Nature: Report**Dissemination: Public****Version History**

Version	Date	Author	Comments
0.1	17 Jan 2017	A. Scharl	Initial Version
0.2	07 Feb 2017	W. Rafelsberger	Major Revision
0.3	16 Feb 2017	A. Brasoveanu	Statistical Indicator Integration
0.4	20 Feb 2017	A. Hubmann	System Architecture
0.5	22 Feb 2017	W. Rafelsberger	Revision and Review
0.6	25 Feb 2017	W. Rafelsberger	Revision and Review
1.0	28 Feb 2017	A. Scharl	Revision and Final Edits

Acknowledgement

This project has received funding from the European Union's 7th Framework Programme for research, technological development and demonstration under grant agreement number 619706.

Executive Summary

This software deliverable reports the work conducted in WP6 of the ASAP project between months 13 to 36. It describes the development process and reports significant results that led to final version of the ASAP Dashboard.

Objectives

The main goal of WP6 has been the development of a high-performance visual interface as part of the ASAP system architecture to help analysts navigate big data repositories across multiple dimensions (temporal, spatial, etc.). Special emphasis has been placed on the response time of the interface, and the ability to incorporate metadata as context information when analyzing complex relations in Web content (WP8) or telecommunications data (WP9) – using a multiple coordinated view approach.

Significant Results

The **ASAP Dashboard** integrates and synchronizes the visual analytics components reported in D6.2 and D6.3 into a unified Web application. It enables analysts to interactively explore collected Web content (product reviews, news media article, social media postings), extracted metadata, and statistical observations. The dashboard provides geospatial projections of these data types onto customizable base layers, as well as statistical charts and diagrams (line chart, bar chart, donut char, radar chart and scatter plot) with flexible mechanisms to render metadata attributes and interactive controls to select timescales. Year 3 activities focused on optimizing the event and notification system for intra-module communication, and providing adaptive tooltips and context menus as a user-friendly way to trigger on-the-fly query refinements. Substantial performance gains were achieved through improved data structures for document mapping, a revised indexing strategy, optimized query aggregations, and the use of a *Content Delivery Network* (CDN) to serve cacheable resources.

The **Application Programming Interface (API)** provides effective data interchange mechanisms for statistical indicator data and metadata extracted from textual Web content. In Year 3, the API has been extended to provide key functionalities for the Web content analytics and telecommunications use cases (WP8, WP9). These functionalities will play an important role in the ASAP Exploitation Plan (D10.6) by supporting a flexible and scalable *Visualization-as-a-Service* (VaaS) approach, which can be extended into a *Container-as-a-Service* (CaaS) approach by bundling together visualizations with data acquisition and transformation functionality.

Table of Contents

Executive Summary	2
Objectives	2
Significant Results.....	2
Introduction	4
Visual Dashboard Overview	4
Platform Setup and Configuration	5
User Interface Structure	6
Source and Configuration Management.....	7
Web Content Topics and Statistical Indicators	7
Semantic Associations	8
Trend Charts	8
Content View.....	8
Visualizing Context Dimensions.....	9
Statistical Indicator Integration	9
Statistics Sidebar	9
Integrated Visualization of Structured and Unstructured Data	10
Geographic Map	11
On-the-Fly Query Refinement	11
Temporal Controls.....	12
Adaptive Tooltips and Context Menus.....	12
System Architecture	13
Indexing Strategy and Deployment	13
Content Delivery Network Support and Migration	14
Application Programming Interface	15
Web Application Technology Stack.....	16
Embeddable Visualization-as-a-Service (VaaS) Architecture	17
Summary	18
References	19

Introduction

The ASAP Dashboard is intended to support free insight generation without prior modelling of a domain, embracing both unstructured (Web intelligence) and structured (linked data) sources. Shneiderman et al. (1996) present a taxonomy of data types in the context of visualization, including temporal and multivariate data. ASAP dynamically combines such data types on the fly - taking into account use case specifics and current user tasks. D6.1 to D6.3 focused on individual visualization components developed for the interactive visualization of heterogeneous data from multiple sources. The deliverables already described the concepts and requirements of data matching and integration. D6.4 documents the work conducted in T6.3 to integrate the components of T6.1 and T6.2 into the ASAP Dashboard.

The visual analytics components are building blocks for the full visualization dashboard and include interactive D3.js¹ modules for line charts, donut charts, bar charts, radar charts, scatter plots and geographic maps, and are synchronized within the ASAP dashboard. The dashboard offers an explorative interface using a set of visualizations to get a better sense of hidden patterns and trends. It enables analysts to investigate large quantities of data from heterogeneous data sources. As part of the reporting on the progress in M30, D6.3 already included preliminary results from T6.3, which were extended and revised for this final report. Since T6.3 builds upon the components developed in T6.1 and T6.2, D6.4 only references the corresponding deliverables, but does not provide a full documentation. This deliverable will be complemented by the Usability Report (D6.5).

Visual Dashboard Overview

ASAP integrates real-time data feeds from multiple sources via an open API (see Appendix C of D6.3),² which allows uploading structured and unstructured datasets, searching for specific sets of indicators or documents, and embedding individual visualizations in Web-based applications (T6.1, T6.2).

The ASAP dashboard³ combines several visualizations to represent the contextualized information space using a multiple coordinated view approach. Synchronized widgets show the various metadata dimensions (see Figure 1; the more lightweight look and feel in comparison to the mockup of D6.1 reduces interface complexity and highlights the actual content elements).

¹ www.d3js.org

² api.weblyzard.com

³ asap.weblyzard.com

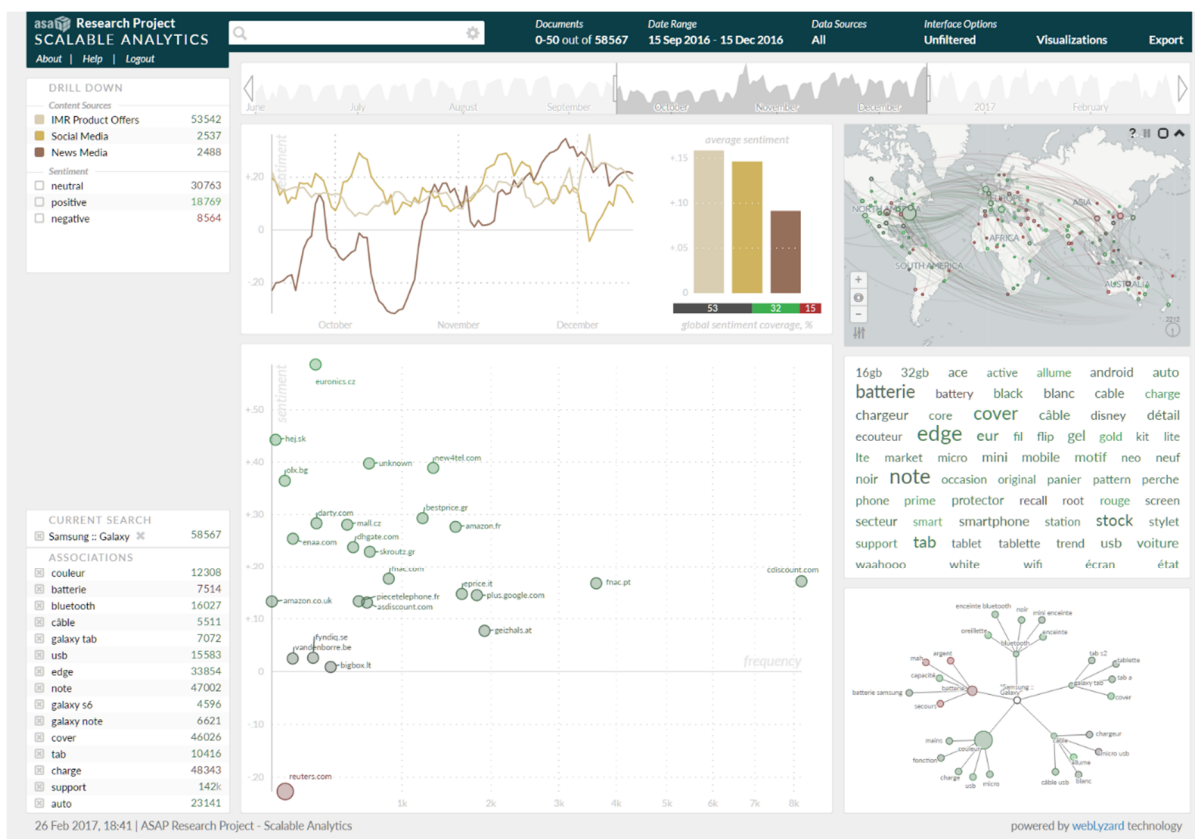


Figure 1. Screenshot of the ASAP Dashboard including the new timeline selector, line chart, bar chart, scatter plot and geographic map with aggregated location statistics

The interactive widgets help explore the storytelling potential of big data visualization, addressing calls for new methods to support the complementary relationship between the explorative and analytical dimensions of information visualization.

Platform Setup and Configuration

Work in T6.3 included the development, setup, configuration and testing of the ASAP Dashboard. This multi-step process consists of:

- **Content and Indicator Data Feeds.** The dashboard combines structured and unstructured data sources. The ASAP API framework (D6.3) was used to ingest data from external sources including product offers gathered by IMR and statistical telecommunications indicators from WIND. Additional news and social media streams are provided by the internal content acquisition pipeline of web-Lyzard. These feeds are streamed into the platform through the *Document API* and the *Statistical Data API*, which allows third-parties to provide both content as well as custom metadata annotations to the platform. The content sources are available for selection in the data sources menu of the dashboard's header

section, enabling users to explore the various sources separately or in combination. The indicator data is available via the new *Statistics* sidebar section (accessible via the *hamburger icon* of the upper right sidebar), complementing the *Topics* and *Drill Down* sections (see D6.3).

- *Content Customization and Disambiguation.* Selecting relevant news and social media content for the ASAP dashboard, in line with the use case requirements, required effective methods for content filtering and disambiguation. We used a domain-specificity measure based on a combination of blacklists and whitelists to assess the relevance of gathered news media articles in the context of telecommunications and big data, in addition to a geotagged Twitter feed and the domain-specific feeds provided by project partners.
- *Database Configuration.* The dashboard instance was activated using a separate user data schema and corresponding PostgreSQL tables, created by importing a general portal template with customized specifications for the ASAP project. This step includes the creation of administration accounts to maintain the platform, as well as regular user accounts to customize the dashboard and access specific data feeds.
- *Application Server Configuration.* Each dashboard instance runs on a dedicated Apache Tomcat application server. This includes creating override settings to adjust the ASAP dashboard including (i) title and layout, (ii) PostgreSQL, Elasticsearch, MongoDB and other service connection settings, (iii) authentication and user-specific feature configuration, (iv) URL schema adjustments, and (v) interface and visualization component configuration.
- *Indexer Setup.* While the application server hosts the dashboard interface itself, it is necessary to configure an indexing service for initial batch operations and create accounts for each use case partner and data source.
- *Deployment.* This step concludes the setup with deploying the configured components to the server infrastructure. This step needs to be repeated for incremental feature updates.

User Interface Structure

The ASAP dashboard builds on earlier insights gained from the *Media Watch on Climate Change* (Scharl et al., 2013), which initially served as a rapid prototyping platform to develop and test the widget synchronization, to gather user feedback (T6.4), and as an outreach channel (WP10). Rather than relying on simple statistical representations, the visual analytics dashboard supports the real-time synchronization of multiple coordinated views to convey context information along various semantic dimensions (Scharl et al., 2016). The dashboard distinguishes three types of context:

- *Lexical Context* – Specific vocabulary and sequence of words that precede or follow a statement (Fischl and Scharl, 2014; Wattenberg and Viégas, 2008);
- *Geospatial Context* – The author’s location and the geospatial references contained in a document (Niepold et al., 2008; Scharl and Tochtermann, 2007);
- *Relational Context* – Frequency distribution of named entities in the content repository, and co-occurrence patterns among these entities (Derczynski et al., 2015; Weichselbraun et al., 2015).

The dashboard provides visual means to analyze the content repository and the statistical indicators (Brasoveanu et al., 2017). It is divided into different content areas (see Figure 1), briefly summarized in the following sections – for additional details about these interface elements, please refer to the detailed online documentation.⁴

Source and Configuration Management

The upper menu shown in Figure 2 provides temporal controls, a selector for the Web content sources to be analyzed (news media, social media, IMR product offers), and access to a sidebar for exporting search results and other datasets in various formats.

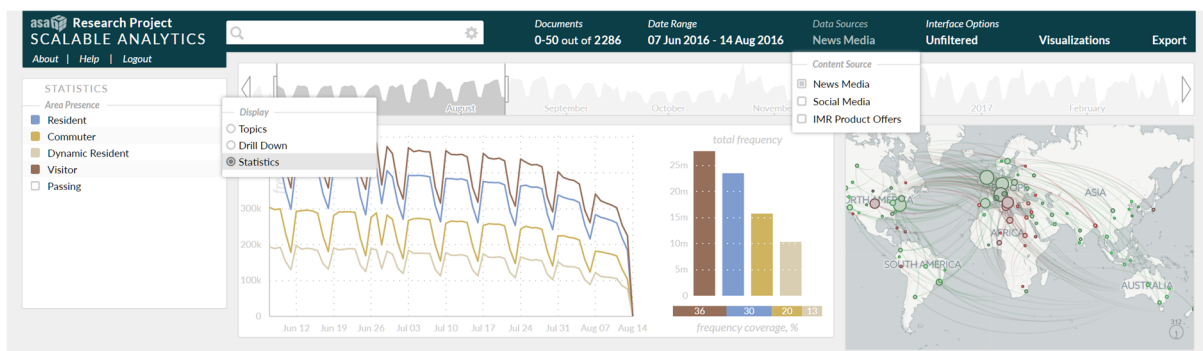


Figure 2. The upper section of the ASAP Dashboard with header, temporal controls and statistics sidebar showing selected indicators in the trend chart component (in this example, WIND area presence data from September to November 2016, classified by user type)

Web Content Topics and Statistical Indicators

The upper left window of the dashboard contains the topic and statistical indicator management section, which provides one-click access to the topics defined in the content customization and disambiguation step during the setup process. Clicking on a topic element triggers one of the following actions: (i) topic labels initiate a full-text

⁴ www.weblyzard.com/interface

search based on the topic's specific configuration; (ii) topic markers (= small rectangles) select topics to be shown in the charts; (iii) the settings symbol allows to configure or delete topics. As shown in Figure 2, this section can be switched to the statistical indicators view to explore data provided via the *Statistical Data API*.

Semantic Associations

The lower left view displays a list of associations with the search term, based on the selected source and data range. Specific functions that enable analysts to track and visualize these associations, as well as other metadata patterns (via the right sidebar), will be presented in the next sections.

Trend Charts

Developed in T6.1 and shown in Figure 3, the trend chart in conjunction with an optional donut or bar chart shows one of the following times series, selectable via a floating menu: (i) the share of voice, a comparative measure of attention based on the relative number of mentions, (ii) the frequency of selected topics in the specified time interval, (iii) the average sentiment regarding these topics, or (iv) the level of disagreement (standard deviation of sentiment). The charts support either the display of the raw values or can be switched to moving averages of 7, 30 or 60 days.

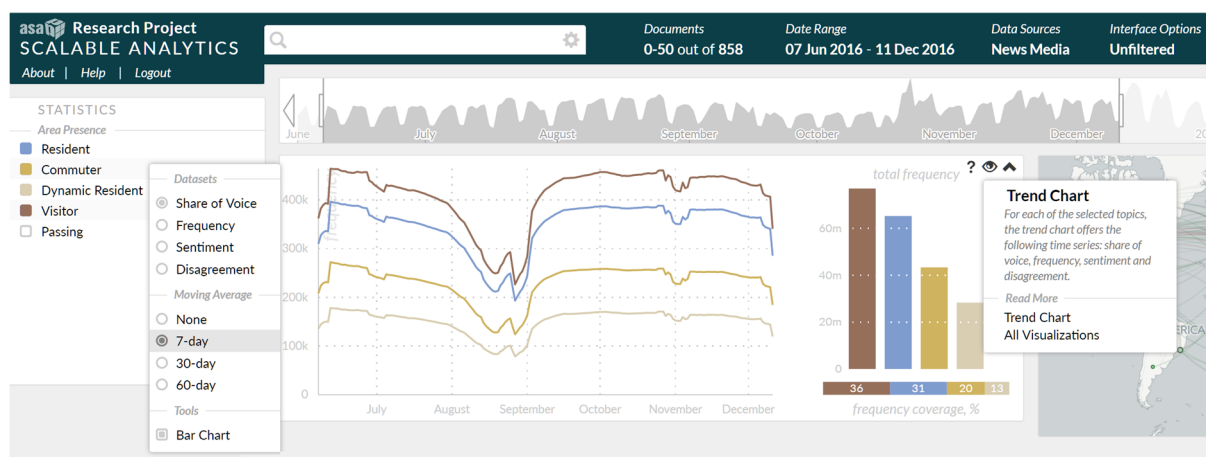


Figure 3. The trend chart component showing various statistical indicators including the contextual menu to the left as well as the help tooltip to the right.

Content View

The floating menu provides seven representations to show search results: documents, sentences, word tree, entities, relation tracker, sources, source map, and Twitter network. Interactive controls in the *document* and *sentence* views include: (i) *mouse-over* to preview documents; (ii) *click* to select a document and show its content in extended form; and (iii) *second click* to switch to full text view.

Visualizing Context Dimensions

The sidebar on the right features a configurable selection of visualizations aimed at providing insights into various context dimensions describing the currently selected information space (e.g. the selected topics, current search or selected statistical indicators). The visualizations include the geographic map developed in T6.2 (see below).

Statistical Indicator Integration

Statistics Sidebar

Gathering usability feedback on the GROOVE visualization reported in D6.2 showed that most users prefer more well-known and easier to interpret interface representations. Based on this feedback, we decided to discontinue the further integration of GROOVE and instead focused efforts in Year 3 of ASAP on a drill-down sidebar to show temporal views and convey temporal variations in metadata and statistical attributes. The drill down sidebar has the additional advantages of providing a clear separation of attribute values – e.g., positive vs. neutral vs. negative sentiment.

This new navigational structure forms the basis for the *Statistics Sidebar*, which is the interface to select statistical indicators provided via the Statistical API. The component enables the selection of distinct statistical observations for further analysis with visualization components within the dashboard. Each data point from a time-series corresponds to an observation.

This section of the left sidebar is enabled by using the menu in the topic sidebar header to switch between the different sidebar types. The options are (i) *topic management*, (ii) *metadata drill-down* and (iii) the *statistics* interface. The last one is the statistical observation exploration interface specifically developed for the ASAP Dashboard. The menu consists of observation categories and allows the trend chart to compare elements within the categories as well as comparison the current content search and its associations. The contents of the statistical observations sidebar reflect the data streams provided through the Statistical API. Each entry in the sidebar represents a particular slice from a dataset (e.g., calls or SMS data in Rome), the time interval being specified through the global *Date Range* selector in the upper menu or the new timeline interface presented in Figure 1.

Figure 2 shows a screenshot of the statistics menu, and Figure 3 the available moving average options. The statistical observations can be organized in custom categories. The most left column is used to select individual attributes for further analysis. On selection, a color from a predefined palette gets dynamically assigned to the statistical attribute, which is then used in other dashboard components to identify the attribute.

Integrated Visualization of Structured and Unstructured Data

The WYSDOM chart provides a dynamic assessment that allows real-time insights into the impact of events on communication patterns. It shows to what extent an event (e.g. a peak in the blue area that represents a statistical indicator) impacts observable patterns in Web content - in term of positive, neutral and negative coverage, or desired and undesired associations (see Figure 4). The set of terms to measure *desired* associations can be updated on the fly, for example to evaluate the degree of association between a telecommunications service with desired topics considered beneficial. The *undesired* category determines whether specific topics were avoided successfully. This computation should be considered adaptive and part of an iterative feedback cycle, customized to an organization's evolving dissemination goals.

Earlier versions of WYSDOM were limited to custom hard-coded integrations, for example server access statistics (visits, page views, etc.) from log file analysis tools such as *Google Analytics* or *Piwik*. For the ASAP Dashboard, a new version has been developed that supports all observations provided via the *Statistical API*. The dynamic integration ensures its applicability for a wide range of scenarios, as exemplified through the WIND telecommunications use case.

The stacked bar chart of Figure 4 consists of the following areas: Green areas above the horizontal axis represent the association with (i) desired topics and the number of (ii) positive references; blue areas reference statistical observations, i.e. trends in area presence; the grey area the number of (v) neutral references; red areas below the axis the number of (vi) negative references and the association with (vii) undesired topics. Adaptive tooltips described in a later section of D6.4 display additional context information; e.g., topics responsible for observable changes. The adaptive calculation enables analysts to assign weights to the above mentioned seven components to set their perceived importance.

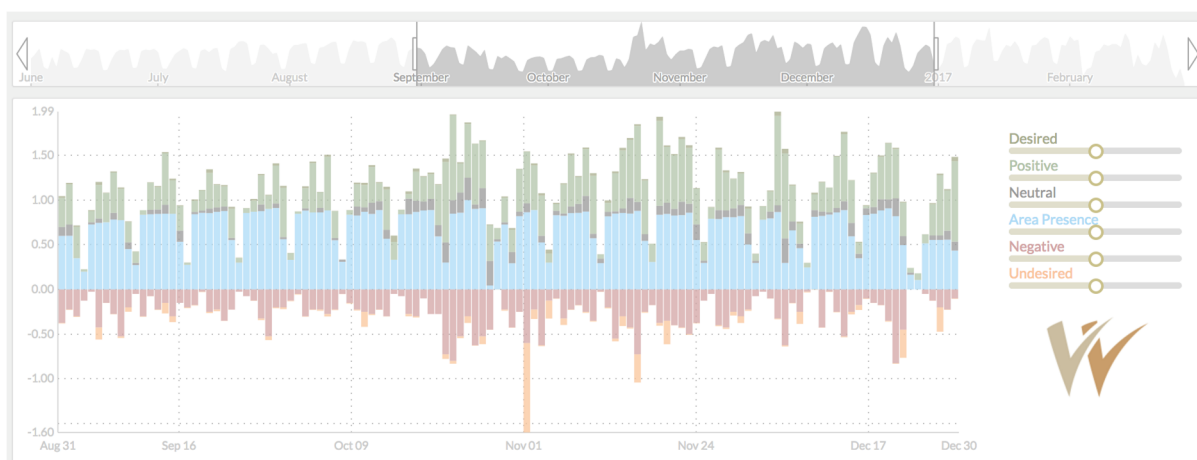


Figure 4. WYSDOM chart with accompanying timeline selector showing the area presence data of WIND between September and December 2016

Geographic Map

While the default implementation of the *Geographic Map* displays aggregated frequency and sentiment values, the library has been extended to support function overrides to display additional numeric metadata and statistical indicators on visual attributes (e.g. size, color, opacity) as shown in Figure 5. It supports custom base layers, dynamic clustering, and data overlays.

The scalability of the component in comparison to Year 2 has significantly been improved – instead of analyzing a result set of between 50 and 250 Web documents, the latest version aggregates the geotags of the entire result set (tested with queries up to 90 million documents), generates a frequency-sorted list of referenced locations, and projects the corresponding values onto the geographic map.

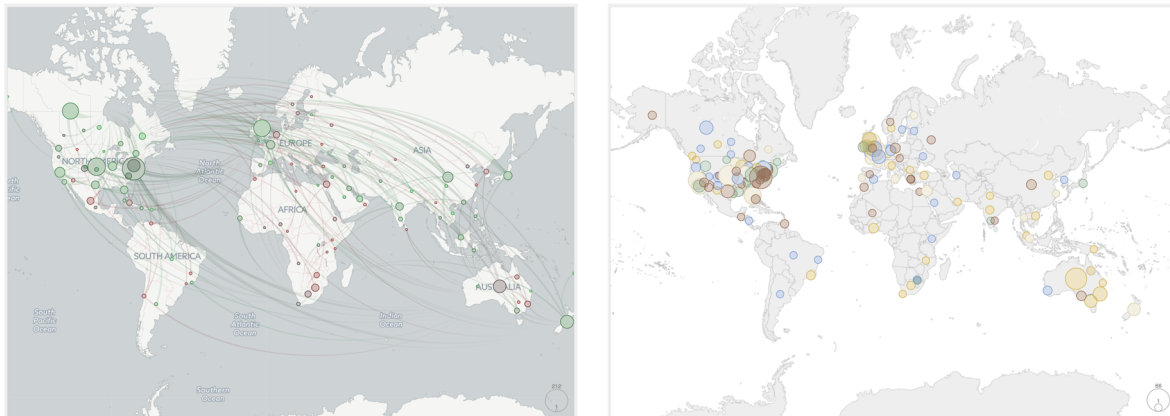


Figure 5. Comparison of two different geographic map modes: Sentiment distribution including trajectories linking source and target location (left); alternate color coding to compare multiple statistical indicators (right), using two different base layers.

On-the-Fly Query Refinement

The individual visual analytics components developed for the ASAP Dashboard as well as the original integration into the Dashboard have already been described in more detail in Deliverable D6.3. The *charts* are central components of the dashboard, using D3.js to render dynamic transitions – to show different source combinations, for example, or incremental changes in a dataset. In T6.1, we rewrote and extended the chart module to eliminate third-party library dependencies, increase rendering performance, and support a wider range of usage scenarios. T6.2 then extended the geographic map component, reported in Deliverable D6.3. The usability considerations and evaluations of these components are summarized in deliverable D6.5.

In the final phase of the ASAP project, special emphasis was placed on interactive controls for on-the-fly query refinements using these components – in terms of time interval for the analysis, Boolean operators, and component-specific search options.

Temporal Controls

Date selection is done using the *temporal controls* to investigate longitudinal data. The time slider element consists of a timespan bar in conjunction with a sliding window. From M30 to M36, these D6.3 components were further revised and integrated with the *Statistics* sidebar and the contextual tooltip menu described in the next section.

Adaptive Tooltips and Context Menus

The development work in WP6 resulted in a new layout and increased feature set. The more lightweight look and feel, as compared to the initial design reported in D6.1, reduces complexity and highlights the actual content; i.e. query results and visualizations. Instead of showing all the available options at once, many features of the user interface are now hidden until being activated via hovering above the corresponding interface element. To further reduce cognitive overhead, the new navigation structure distinguishes between global, view-specific and element-specific settings and action.

To support interactive query refinement while exploring the use case datasets with the ASAP dashboard, *adaptive tooltips* in conjunction with context menus were developed in

Year 2 of the project (representing the element-specific part of the multi-layer menu structure outlined above). In Year 3, they have been further extended, aligned between the different visualization tools, and fully integrated. The tooltips offer the most relevant actions and settings in the context of the current user interaction, and always include:

- a *query preview* in the form of a line chart,
- the ability to (i) *replace* the search term, to (ii) *extend* or (iii) *restrict* the search via Boolean operators, or to (iv) *exclude* documents containing a specific term from the result set.
- the *top keywords* associated with the hovered topic.

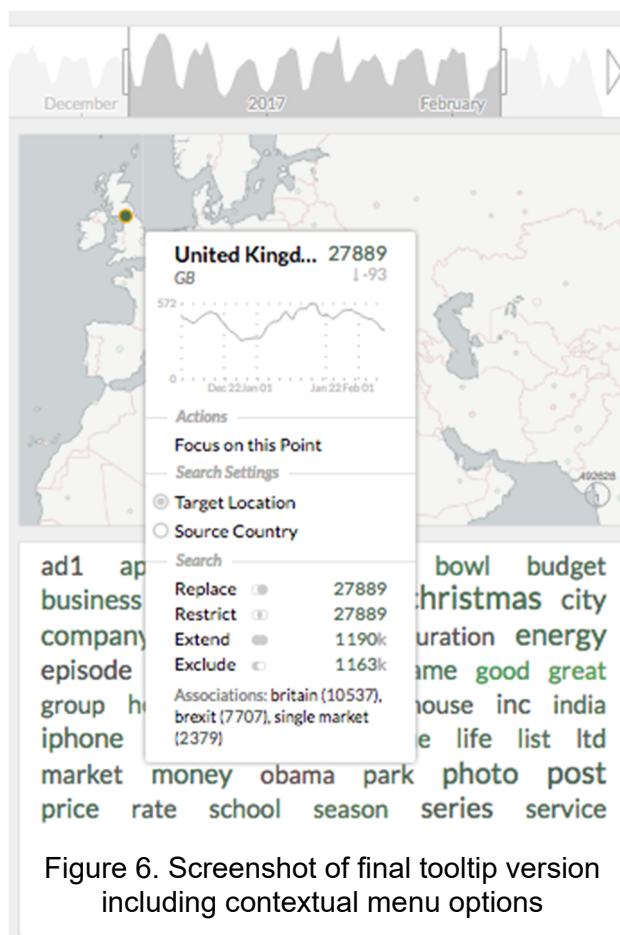


Figure 6. Screenshot of final tooltip version including contextual menu options

The design process including documentation on design and implementation iterations as well as usability considerations are described in more detail in Deliverable D6.5. The final version features a new contextual line chart as well as trending indicators. This enables detailed analysis of data points within the dashboard without triggering a new search so the user can stay within the context of the current information space. The search options are enriched with previews of the number of matches that an actual search would trigger. Hovering the options dynamically updates the line chart giving the user the option to quickly compare the frequency distribution of each search option.

The tooltips and contextual menu were developed considering the high-performance requirements of the ASAP project. Based on a test corpus of one million news media articles, the loading time for the full tooltip and contextual menu averages at 320ms.

System Architecture

This section describes the core architecture of the platform, the indexing and deployment strategy, as well as the API framework for (i) data and metadata ingestion and (ii) embedding search results and visualizations into third-party applications.

Indexing Strategy and Deployment

Elasticsearch.⁵ The visualization components of ASAP are based on high-performance queries on unstructured and structured data repositories. Before the start of the ASAP project, the webLizard content processing pipeline was based on Apache Lucene⁶ and largely built around the processing of large datasets in batch mode. We increased flexibility through a modularization strategy, together with a migration to Elasticsearch, a distributed search and analytics engine made available under an Apache 2 open source license. Elasticsearch provides a RESTful API using JSON over HTTP. It ensures the required scalability for real-time queries that provide the data for the dashboard through multi-tenancy and sharded indexing. To further improve query speed and limit Elasticsearch index sizes, each data source is mapped to multiple, time-sliced Elasticsearch indices. Depending on the data volume of the respective source, slices can be generated daily, weekly, monthly or yearly.

Elasticsearch not only speeds up accessing domain-specific repositories of unstructured content, but also facilitates slicing statistical linked data and integrating other data sources. Elasticsearch aggregations yield additional information for user queries

⁵ www.elastic.co/products/elasticsearch

⁶ lucene.apache.org

or API calls: (i) *top k answers* – highest ranked search results based on a simple or advanced query; (ii) *data slices* – most often across time, but in the case of complex datasets other dimensions can be included as well (e.g. analyzing tourism transactions to obtain all flights originating from a specific airport); (iii) *data summaries* – indicator performance is highlighted not just by color coding, but also through a data summary.

Vert.x.⁷ On top of Elasticsearch, the API specified in Annex C of deliverable D6.3 is implemented using Vert.x, a high-performance, lightweight and scalable application framework. Using its distributed event bus, larger applications can be broken down into micro-services, which can then be horizontally scaled over multiple hosts. As a non-blocking, asynchronous toolkit, Vert.x scales to several thousand requests per second. The modular processing strategy of WP6, in conjunction with the provision of Vert.x-based RESTful APIs, enabled the integration of intermediate results in T5.3.

Docker.⁸ For a more flexible deployment of the ASAP Dashboard, instead of running the application server directly on the host environment, the required dependencies and main application have been ported to be executed via Docker instances. Since the Docker engine container comprises just the application and its dependencies, this increases the portability and efficiency of the developed components.

Content Delivery Network Support and Migration

The dashboard architecture has been restructured to distinguish between static and dynamic resources. This extension allows to load static assets such as images, JavaScript files and style sheets from a dedicated domain, which was migrated to use Amazon Cloudfront⁹ as a Content Delivery Network (CDN) in February 2017. In addition, assets shared between different dashboard instances (geographic tiles, favicons, etc.) are stored only once. Assets of specific dashboard instances have a separate URL space plus a version string in the URL so that no stale cached resources are loaded in the portal. Cache lifetime for the CDN assets is set to one month.

Activating the CDN for the portal resulted in a speed improvement of 26.3% for the initial page load (as measured from a server in Canada, to assess impact on international distribution). As a result, the dashboard now ranks better in various speed tests, which positively impacts both usability and search engine ranking – using Google's PageSpeed Insights,¹⁰ for example, the system improved from 50% to 84%.

⁷ www.vertx.io

⁸ www.docker.com

⁹ aws.amazon.com/cloudfront

¹⁰ developers.google.com/speed/pagespeed

Special emphasis has been placed on the clear separation of content (static, dynamic) and the login framework to protect potentially sensitive information. The login server, for example, is exempted from CDN delivery. For the same reason, all data transfers of the dashboard are now served via HTTPS connections (and not only the login data), which will be highly relevant for the follow-up exploitation of project results (D10.6).

Application Programming Interface

For effective data interchange with the other technical work packages and the flexible integration of the D3.js-based visualizations developed in D6.1-D6.3 into third-party applications, an open *Application Programming Interface* (API) for structured and unstructured data was developed and first reported in D6.3. This API not only allows rendering multi-source data for the use case applications of WP8 and WP9, but also plays an important role in the ASAP exploitation strategy (D10.6). It supports a flexible *Visualization-as-a-Service* (VaaS) approach or, when bundled together with data acquisition and transformation services, even a *Container-as-a-Service* (CaaS) offering.

Prior to ASAP, the webLyzard dashboard existed in different versions, fully customized to client specifications. The manual effort of customizing each system quickly became a cost driver that does not scale well with an increasing number of projects. When aiming to support flexible on-the-fly visualizations based on dynamic datasets, as required by ASAP, this approach was no longer feasible. This led to the development of several open APIs as part of a uniform framework for the rapid integration of heterogeneous data sources into a common visualization processing pipeline.

The APIs are typically based on the principles *DRY – Don't Repeat Yourself* (i.e., all functionality related to a certain object is in a single place) and *CvC - Convention over Configuration* (i.e., naming conventions, date conventions, geolocation conventions) in order to enable fast computations (Fisher and Murphy, 2016). A lot of configuration options would have also added more processing steps (e.g., data conversions, additional data wrangling steps) and made the on-the-fly integration difficult.

The ASAP visualization pipeline for processing unstructured and structured content from multiple sources addresses the data matching problem described in D6.3. It is being made available via an open API for members of the consortium, and selected external partners. It consists of the following services:

- The *Document API* was used to process data from websites and social media, but also product offerings from IMR.
- The *Statistical Data API* is used for delivering statistical indicators to the dashboard (e.g., calls or SMS data from WIND).

- The *Search API* is used to process the queries for both documents and statistical observations.
- The *Visualization API* is used to process the visualization, both in the dashboard and in the stand-alone offerings.

The API specification and implementation has been improved continuously in alignment with the specific requirements of the consortium partners. Building on the earlier version of the APIs reported in Deliverable D6.3, the full PDF specification was converted to a more user-friendly online documentation¹¹ based on the *Swagger*¹² open source framework, where each API comes with its own conventions, code examples and documentation.

Web Application Technology Stack

The client-side JavaScript technology stack has been improved to make use of high performance frameworks while considering the complexity and long-term maintainability of the code base.

- *Bower and RequireJS*. Prior to ASAP, the platform's JavaScript code was organized in multiple files which were concatenated during the build process. This lead to maintenance issues especially when using individual modules in different context's, e.g. the full dashboard as well as the mobile and embeddable version. *Bower*¹³ is now used to manage third-party and internal referenced stand-alone modules as dependencies with version management. *RequireJS*¹⁴ enables to structure the JavaScript code in a modular way with a clear dependency tree. The build process takes this into account, analyzes the dependencies and creates an optimized production build of the JavaScript code.
- *Data-Driven Documents (D3)*. To ensure rapid prototyping of the visual methods developed, and a seamless integration into the ASAP Dashboard (T6.3) and external web-based applications, WP6 uses the D3 JavaScript library (Bostock et al., 2011). D3 is perfectly suited for the purposes of ASAP, since it is not focused on a new grammar for graphics, but rather on integrating existing standards to create effective visualizations.

¹¹ api.weblyzard.com

¹² www.swagger.io

¹³ www.bower.io

¹⁴ www.requirejs.org

- *RactiveJS*.¹⁵ The tooltip integration in Year 3 has replaced the classic stack based on *jQuery* and *Mustache templates* with *RactiveJS*, which makes use of efficient shadow DOM techniques to optimize the manipulation of interactive Web applications. Additionally, it enables a more concise separation of concerns and its API integrates well with the *Event and Notification System*.
- *Event and Notification System*. To allow the integration of tightly coupled views in multiple contexts, the newly developed event and notification system (Figure 7) described in D6.3 has been integrated as a core component of the ASAP Dashboard.

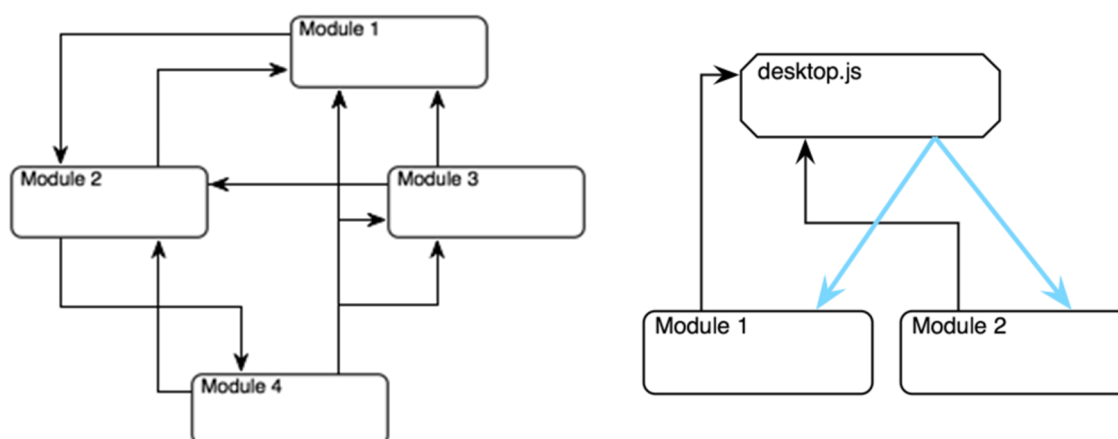


Figure 7. Example dependency graph of modules following the register/notify pattern (left) versus the events and notifications system developed within ASAP (right)

Embeddable Visualization-as-a-Service (VaaS) Architecture

To support the usage of the visual analytics components in multiple contexts, the previously described *event and notification system* was developed. In contrast to the legacy implementation used in previous versions for example of the dashboard and geographic map component, the new system does not require hard event dependencies inside single modules. This allows the rapid development of customized visualization dashboards based on the main platform. The updated event system was used in combination with the various visual analytics components to develop the embedded prototype for the *Telecommunication Data Analytics (TDA)* use case based on structured datasets provided by WIND (reported in D6.3)

¹⁵ www.ractivejs.org

Summary

Deliverable D6.4 summarizes the work conducted in T6.3 of the ASAP FP7 project. The result is the fully integrated ASAP Dashboard¹⁶ – aggregating, aligning and visualizing structured as well as unstructured datasets. It supports the analysis and visual exploration of information spaces along multiple semantic dimensions, based on data ingested via an open API and corresponding indexer component.

The dashboard builds upon a portfolio of visual analytics tools for the rapid rendering of complex datasets using visual cues to render metadata attributes, and interactive mechanisms to select specific time intervals (T6.1, T6.2). Among these tools are a versatile *chart library* and a geographic map to project the regional distribution of the aggregated datasets.

The open API allows integrating the WP6 visualization engine with the components developed in other WPs, and rendering multi-source data for the use case applications in WP8 and WP9. It connects the data streams to either individual visualizations, or to the entire ASAP dashboard. The API has helped to ingest the consumer product offers collected by IMR (D8.4) as well as anonymized, pre-clustered call data records provided by WIND (D9.4).

The fully functional ASAP dashboard has been deployed with adaptive tooltips and view-specific context menus to support on-the-fly query refinements, and provides various options for data highlighting, selection, and drill down. The Elasticsearch cluster powering both the dashboard and the API framework has been optimized for improved performance and resilience: (i) reduced memory load of individual nodes through data structure changes within the engine and the dashboards document mappings; (ii) automated query optimizations for resource-heavy aggregations; (iii) monthly index slices to reduce the size of active indices to answer queries, and improve horizontal scaling across multiple hosts for parallel queries.

The API specification has been continuously evolved, and plays an important role in the ASAP exploitation strategy (T10.2) by supporting a flexible and scalable Visualization-as-a-Service (VaaS) approach. Containerized visualizations support the integration of WP6 components into the data analytics workflow of T5.3 and the Telecommunication Data Analytics application of T9.3. The VaaS event and notification system offers a unified framework to manipulate intermediary and final data sets.

¹⁶ asap.weblyzard.com

References

- Bostock, M., Ogievetsky, V. and Heer, J. (2011). "D3: Data-Driven Documents", *IEEE Transactions on Visualization and Computer Graphics*, 17(12): 2301-2309.
- Brasoveanu, A.M.P., Sabou, M., Scharl, A., Hubmann-Haidvogel, A. and Fischl, D. (2017). Visualizing Statistical Linked Knowledge for Decision Support, *Semantic Web Journal*, 8(1): 113-137.
- Derczynski, L., Maynard, D., et al. (2015). "Analysis of Named Entity Recognition and Linking for Tweets", *Information Processing and Management*, 51: 32-49.
- Fischl, D. and Scharl, A. (2014). Metadata Enriched Visualization of Keywords in Context. *6th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS-2014)*. Italy, Rome: ACM Press: 193-196.
- Fisher, P. and Murphy, B.D. (2016). "Domain Model Fundamentals", *Spring Persistence with Hibernate*. Eds. P. Fisher and B.D. Murphy. New York, USA: Apress. 85-107.
- Joint Formulary Committee (2014). *British National Formulary (BNF) 67* London: Pharmaceutical Press.
- Niepold, F., Herring, D. and McConville, D. (2008). "The Role of Narrative and Geospatial Visualization in Fostering Climate Literate Citizens", *Physical Geography*, 29(6): 529-544.
- Scharl, A., Hubmann-Haidvogel, A., Sabou, M., Weichselbraun, A. and Lang, H.-P. (2013). From Web Intelligence to Knowledge Co-Creation – A Platform to Analyze and Support Stakeholder Communication, *IEEE Internet Computing*, 17(5): 21-29.
- Scharl, A. and Tochtermann, K., Eds. (2007). *The Geospatial Web - How Geo-Browsers, Social Software and the Web 2.0 are Shaping the Network Society*. London: Springer.
- Scharl, A., Weichselbraun, A., Göbel, M., Rafelsberger, W. and Kamolov, R. (2016). "Scalable Knowledge Extraction and Visualization for Web Intelligence", *49th Hawaii International Conference on System Sciences (HICSS-2016)*. Kauai, USA: IEEE Press. 3749-3757 [Best Paper Award].
- Wattenberg, M. and Viégas, F.B. (2008). "The Word Tree, an Interactive Visual Concordance", *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1221-1228.
- Weichselbraun, A., Streiff, D. and Scharl, A. (2015). "Consolidating Heterogeneous Enterprise Data for Named Entity Linking and Web Intelligence", *International Journal on Artificial Intelligence Tools*, 24(2): 1540008 | 1-31.

FP7 Project ASAP

Adaptable Scalable Analytics Platform



**End of ASAP
D6.4 Dashboard**

WP 6 – Information Visualization
webLyzard technology

Nature: Report
Dissemination: Public