

Service Models for Cloud Computing: Search as a Service (SaaS)

Predrag Dašić^{*1,2}, Jovan Dašić², Bojan Crvenković²

¹High Technical Mechanical School of Professional Studies, Radoja Krstića 19, 37240 Trstenik, Serbia

²SaTCIP Publisher Ltd. Tržni Centar Pijaca 101, 36210 Vrnjačka Banja, Serbia

*dasicp58@gmail.com

Abstract— *Search as a Service (SaaS) is a cloud service model whose main focus is on enterprise search or site-specific web search. Modern companies require fast and accurate information from their internal databases, internal document stores, or through the content of a website. Having a reliable searching mechanism is essential for both internal company staff and for external customers. In this paper the overview of current state and technological advances of Search as a Service (SaaS) cloud service is given, as well as its security issues on current internet service platforms.*

Keyword - Cloud computing, Search as a Service (SaaS), internet technologies.

I. INTRODUCTION

The vast amount of data available in electronic form wouldn't be of much use if they couldn't be searched for specific information. It is an essential function for any business database function, either through internal databases, internal document stores, or through the content of a website. Search as a Service (SaaS) is a cloud based service which is a branch of Search as a Service (SaaS), with a purpose of performing enterprise search or site-specific web search. Search as a Service (SaaS) is a sophisticated method of retrieving specific information with great complexity behind it. Although it can perform the same search as common search engines the main differentiation is that it is mainly intended to search through private resources that are not visible to the public web.

Company that provides Search as a Service (SaaS) delivers the hardware and software resources to the client to support searches of their data repositories. For content that needs to be searched the client sends the indexing metadata to the service through the application programming interface (API). In the paper we will describe this process in more detail and show security recommendations for this highly evolving cloud computing feature as well.

II. LITERATURE OVERVIEW

Existing literature offers very little insight on Search as a Service (SaaS) functioning; however, there are numerous publications on cloud related technologies development. Authors of [1] constructed a system for trusted data sharing through untrusted cloud providers to address the security of data, especially the integrity and confidentiality of data of cloud service providers. In their study authors [2] provide a baseline for many objective visual analytics (MOEA) effectiveness, efficiency, reliability and controllability across multiple problem formulations for Search as a Service (SaaS) optimization. Authors of [3] propose a model to discover Web service based on semantic and search engine, and design its architecture. At the same time, we put forward the algorithm of splitting words and the algorithm of query expansion, develop a Web Service search prototype. In paper by [4] an algorithm which solves the problem of semantic input-output message structure matching for web service composition is given.

In paper by [5] authors propose Oblivious Term Matching (OTM) which unlike existing systems, OTM enables authorized subscribers to define their own search queries comprising of arbitrary number of selection criterion. OTM ensures that cloud service provider obliviously evaluates encrypted search queries without learning any information about the outsourced data. Another research by [6] shows that by utilizing well known indexing scheme such as inverted file and R-tree indexes over Web services attributes, the Earth Mover's Distance (EMD) algorithm can be used efficiently to find partial matches between a query and a database of Web services.

Authors of [7] introduced the Lethe indexing workflow to improve query and update efficiency in secure keyword search, while in research [8] they examined different datasets based on the empirical statistics of a document sharing system and alternative theoretical distributions and applied Lethe to generate indexing organizations of different tradeoffs between the searches and update cost. Research done by authors [9] has a goal to present a case study of forensic indexed Search as a Service (SaaS) and evaluate the feasibility of the service. Paper published by [10] presents an approach which integrates the access control mechanism with data encryption in order to comply with Access Control Aware Search (ACAS) principle.

Authors of [11] proposed a personalised search approach for web service recommendation. Interests are extracted from users' records, interest-similar users are selected using the criterion of cosine distance, and finally, services are ranked in decreasing order based on the recommendation from interest-similar users. In paper [12] authors investigate the vast stream of the state of the art in Everything as a Service (XaaS) and identify approaches for migrating applications to the cloud and exposing them as services. A conceptual framework of a service-ontology- based semantic service search engine was designed for Digital ecosystem (DE) by the authors of [13]. Authors of [14] discuss three searching encrypted data approaches that can be used in large cloud storage environment supported access control: IPU (Index-Per-User), CSI (Central-Single-Index) approach, and RBAC (Role-Based-Access-Control) approach. In paper [15] authors propose a searching mechanism to discover Semantic Web Services satisfying user requirements. They proposed to develop a Semantic Search Agent (SSA) to discover required web services from web. Authors of [16] propose an architecture for multi-domain queries through composition of search services. Authors of [17] present a hybridization of Searchable Encryption and Attribute Based Encryption techniques in order to satisfy the ACAS property. Authors of [18] discussed the idea of connecting powerful tools like Openlink Virtuoso, ElasticSearch and PostGIS within a single framework for cloud based framework for full text, geospatial and semantic search. In paper [19] applications of the Search as a Service (SaaS) are given. Search as a Service is used in other areas of cloud supported services such as various applications of Video Surveillance as a Service (VSaaS) [20,21].

III. FEATURES OF SEARCH AS A SERVICE (SAAS)

Search as a Service (SaaS) hosts search engine capable of full-text, numerical, and faceted searching that seamlessly delivers results in real-time even from the first entered character. Full-text search examine all words within full-text fields in order to find the most relevant records. This capability allows quickly searching and returning records from large volumes of data.

Numerical search is used for retrieving results from large datasets which consist mainly from numbers. This is generally the case with statistical data and calculation generated data sets. It uses numerical analysis methods which employs approximations and mathematical optimizations for obtaining approximate solutions while maintaining reasonable bounds on errors. Faceted search, also called faceted navigation or faceted browsing, is a technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters.

A faceted classification system classifies each information element along multiple explicit dimensions, called facets, enabling the classifications to be accessed and ordered in multiple ways rather than in a single, pre-determined, taxonomic order [22]. Table 1 shows some of the most important features of modern Search as a Service (SaaS).

TABLE I
 Features of Search as a Service (SaaS)

Relevance and ranking	SaaS offers customizable relevance and ranking of search results enabling the client to fine tune their product or service visibility.
Typo tolerance	In today's mobile environment it often happens that the user misspells a word or term which without an algorithm that automatically detects such errors would not yield a relevant result.
Smart highlighting	Search results have a highlighted search term or word which facilitates the users decision on selecting the right response.
Facets	Even with the first character the service offers search based facets which improves navigation, drill down, and refinements based on the user's query.
Geo awareness	Service reads the location, and/or language used, of the user and ranks the results which are closer to him. For example, can be highly useful for tourists seeking a coffee shop in Pairs.
Language support	Service has in-build support for all languages without any intervention from the client side. This is an essential function to have in order to attract more visitors.
Security	Prevention of crawling client data, hacking the account, human mistakes, access to your confidential data and other using Two-factor Authentication, Secure your Admin API Key, HTTPS, Unretrievable Attributes and API Key security
Analytics	This feature gives more insights into how the search engine is used. Client can observe factors like: most popular searches, average hits without typos and count, queries that returned no results, activity by countries, most popular filters.

Some of the derived search services are:

- Enterprise Search as a Service (ESaaS) is the implementation of enterprise search engine software into the

cloud environment whose purpose is internal indexing and searching of enterprise-type sources. Its main application is intended for governmental and large business usage.

- Elastic Search as a Service (ESaaS) is a concept that enables users to use various query features which facilitates the process of searching and with added freedom of obtaining desired results even with broader search terms. This term also occurs as an abbreviation of company Elasticsearch which provides Search as a Service (SaaS).

- Grid Search as a Service (GSaaS) is a different approach to distributing work load of search functions where computer resources manage queries in a non-interactive manner, each node performs a different task and delivers it as a common result.

- Intelligent Search as a Service (ISaaS) is a developing cloud service which is supposed to derive secure unification of data from disparate sources across the IT ecosystem (cloud and on-premise) and create ubiquitous search experience.

IV. SEARCH AS A SERVICE (SAAS)

Life cycle begins with importing data from the client's database into the service provider's database. There are usually two methods of importing data, either by dashboard or by using API. Whatever the case is it is recommended to upload the data in batches of 1000 to 10000 records at a time. Usual file formats are JSON, CSV, or a TSV. The data are then sent to the cloud (all servers of the service provider). The data that is stored inside the cloud can be updated and reindexed as necessary. From that point on the databases are always in sync through a service's API for any type of operation. Second step is the configuration of the index for precise ranking and relevance of the search results. The configuration is executed by the client in the Search as a Service (SaaS) control centre. Most Search as a Service (SaaS) providers handle semi-structured data well; however, formatting the data properly can have significant improvements towards speed, reliability and accuracy. There are several recommendations that should be followed:

- Organizing indices. Organize different indices for different types of data, which enables that each index has its own settings and ranking strategy.
- Organizing records. Most service providers support schemaless databases, index data attributes should be formatted to their correct type. For each object it is advisable to set ObjectID which facilitates removal or updating the record by its unique identifier.
- Indexing Relations. It is usually better to index each element of the array in a separate record, to get the best relevance. There are the three important guidelines in order to better index objects:
 - It is better to have several small objects than a big one. It will reduce the probability to have a wrong result.
 - When sharing information between several objects, it is better to use a different name for each attribute. This enables to use attributes to order matches by importance.
 - Finally to have an excellent ranking, we can use the customRanking index setting to introduce popularity of hits.

Last step of the process is the implementation of search interface to any application that is required to have search capabilities. The described steps are graphically shown in Fig. 1.

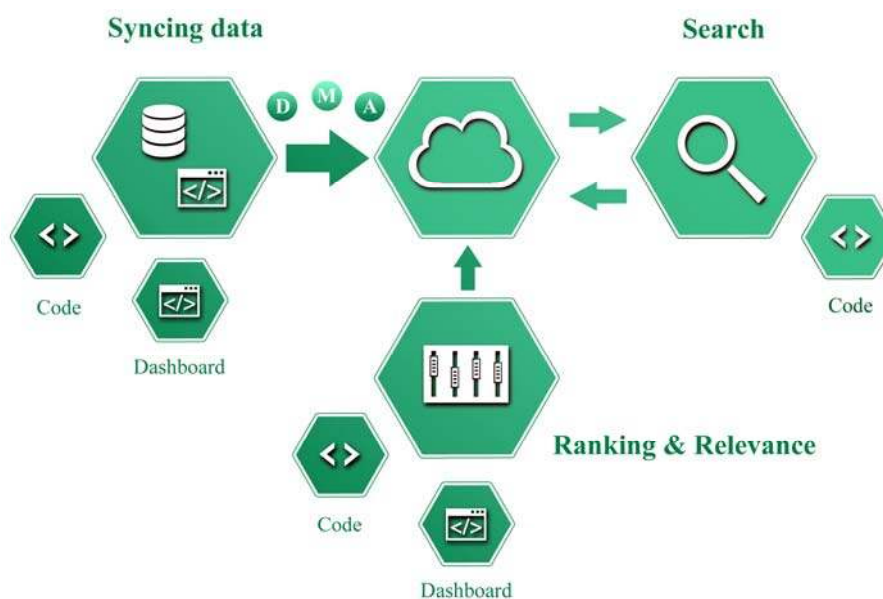


Fig. 1. Life cycle of SaaS. (Adapted from [23])

Some of the search capabilities that modern Search as a Service (SaaS) providers offer are:

- *Instant search results* page shows results, filters and pagination on the entire Web page and gets updated as a user types.
- *Result Filtering*. Filters can be used by date by converting dates to numeric values (i.e. Unix stamp), by numerical value (indexing of numerical values integers, doubles and boolean), by tag used when searching for a specific kind of objects and also by using facets which we explained earlier in the paper.
- *Multilingual search*. If there is a need for several languages indices need to be organized accordingly. General recommendation is to have one index per language to reduce the size of the index, although it is possible to use one index where each record contains all languages.

IV.I. Architectural process of SaaS

Base of operations is the index within the Search as a Service (SaaS) database. Index is the entity in which the data is stored or simply said where indexing and search queries are performed. That index is highly optimized for searching through special formatting and utilization of JavaScript Object Notation (JSON) schema. Every item that is imported to index represents a record which is translated into a JSON object which can be used for actions of searching, displaying, filtering, or ranking. These actions are called operations which can be divided into two types.

- Indexing is the process of adding, updating, deleting, or manipulating the data within the index.
- Searching is the process of querying the data stored in the index to return relevant search results.

This model of indexing and searching produces a valuable time saving process and ease of access by having one properly configured index which can be accessed by any number and type of clients.

Fig. 2 shows elements that constitute Search as a Service (SaaS) architecture.

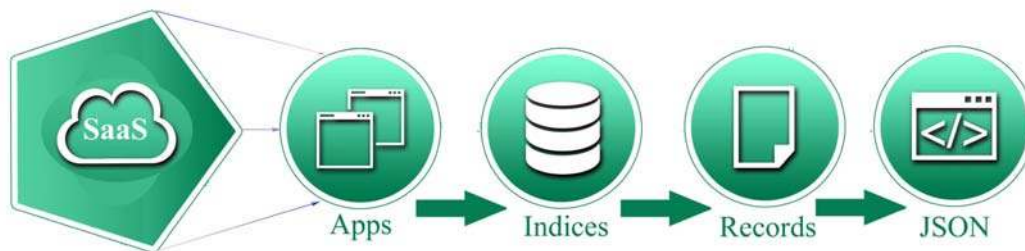


Fig. 2. Elements that constitute Search as a Service (SaaS) architecture

IV.II. Distributed Search Network

Distributed Search Network (DSN) was developed by the company Algolia in 2015 and can be described as a distributed consensus for high-availability and synchronization of data in different regions around the world for the routing of queries to the closest locations via an anycast DNS. It is a novel approach that gives instant response from the search engine to users located anywhere in the world by having a network of several dozen of servers which are geographically distributed all over the world. In Fig. 3 a graphical overview of Distributed Search Network (DSN) is given.

These servers act as synchronized master-to-master clones of the search engine, where all of the client indices are stored and updated in realtime, routing the users query to the physically closest server in the area. This is done by having several clusters of three servers per region. One cluster can host from one to several customers depending on the size of the data a customer has. It is possible to have several customers on a cluster except one customer can grow and change their usage dynamically.

In order to establish a Distributed Search Network it is necessary to adjust the processes to function automatically:

- If the cluster handles a large number of write operations or too much data, it is necessary to automate the migration of a customer to a different cluster.
- If the volume of queries is too large, it is necessary to automate the addition of a new machine to the cluster.
- If the volume of data is too large to be handled by a single cluster it is necessary to adjust the number of shards or divide a single customer over several clusters.

Implementation of these processes enables the assignment of a user to a cluster which is dynamic and changes upon the usage of a cluster. Once a user makes a query he is automatically being assigned a unique application ID which is linked to a DNS record, which uses a specific cluster of machines belonging to that DNS record. This process also performs load balancing using DNS.

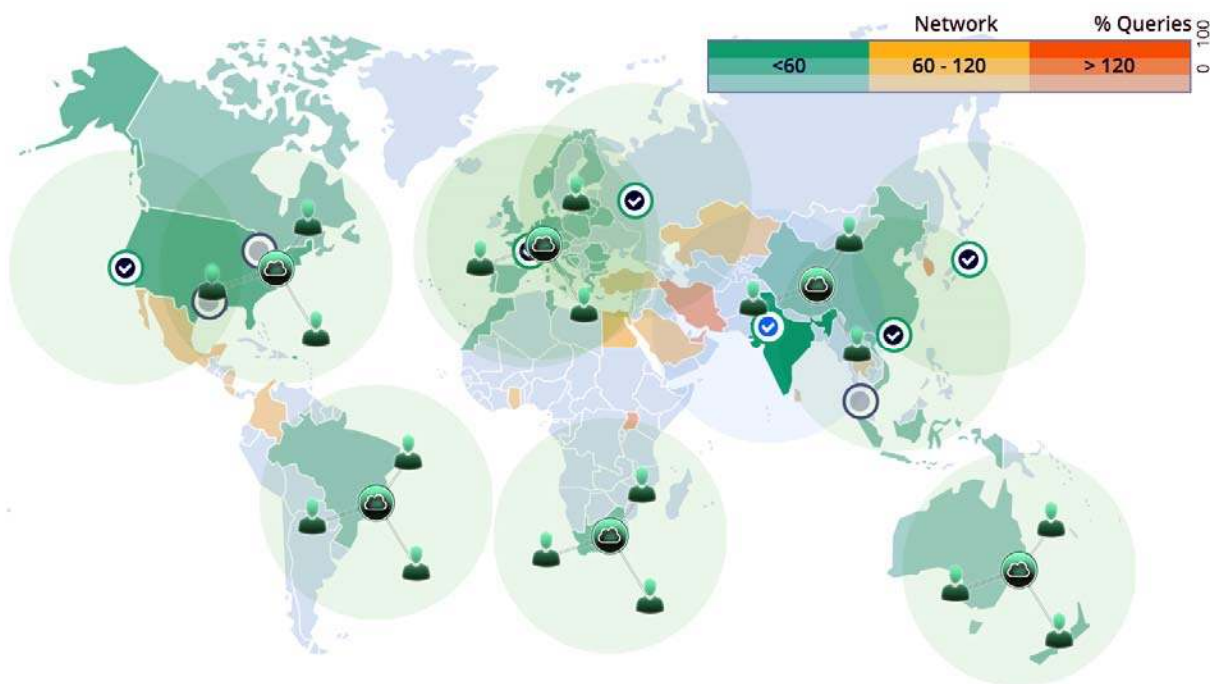


Fig. 3. Distributed Search Network (DSN)

IV.III. Search implementation

So far in traditional implementations of search logic the backend approach was used. It involves the user sending the query to the client’s server which performs the search on the database and redirects the user to the result. Modern Search as a Service (SaaS) architecture eliminates the need for backend intermediary by enabling the user to directly query the Search as a Service (SaaS) index in a cloud and receive the results directly in his browser. This improves overall search latency, enables “search as you type” feature and reduces client’s server load.

Fig. 4 shows the traditional and Search as a Service (SaaS) search processes where it can be clearly seen that the improvement in performance is up to 15 times faster than having the intermediary backend implementation.

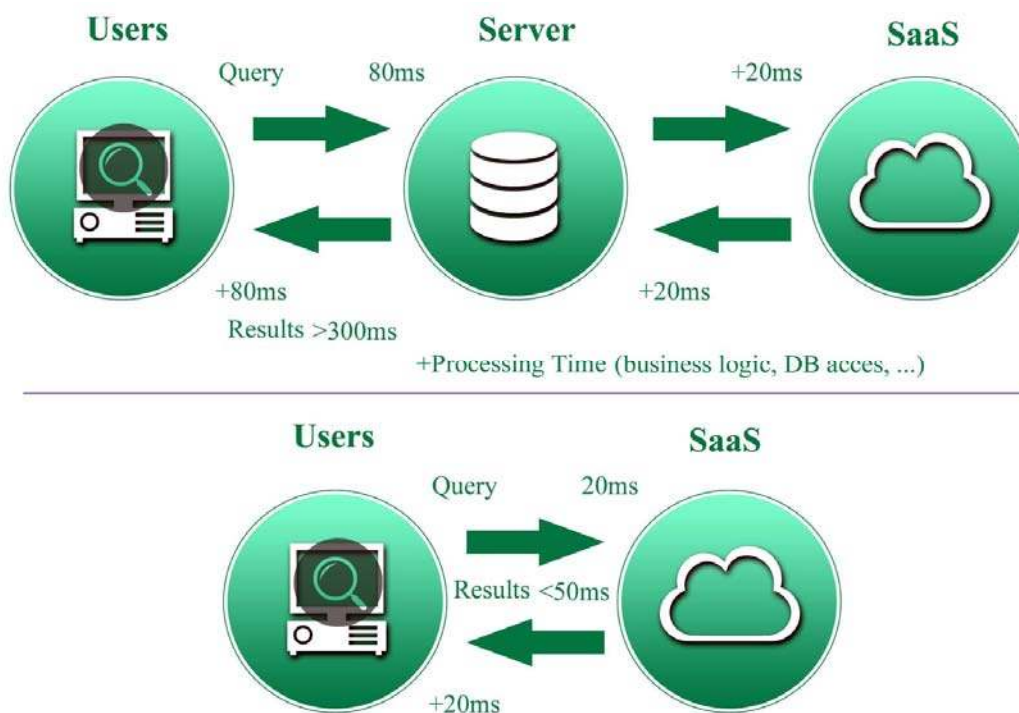


Fig. 4. Search implementation with backend (top) and with frontend (bottom)

V. SECURITY ISSUES OF SEARCH AS A SEARVICE (SAAS)

Data security is a general information technology issue but it also reflects onto the cloud service. The most common concern is the leakage of private data. Author of [24] lists the following concerns: usage data, sensitive information, personally identifiable information and unique device identities.

Constant availability of data is crucial especially in a business environment. Cloud architecture provides high availability through distributed networks but it is still an area that requires further improvements such as resilience to hardware/software malfunction and also defiance of service attacks.

Cloud environment may prove to be harder to monitor for unauthorized access since the servers are located all around the world and sometimes don't allow physical access. Search as a Service (SaaS) needs to implement policies to control connectivity such as discrete logarithm problems, secure one-way hash function, authorized IP addresses and access through virtual private networks.

In a Search as a Service (SaaS) deployment model, susceptible data is obtained from the enterprises, processed by the Search as a Service (SaaS) application and stores at the Search as a Service (SaaS) vendor end. Using strong network traffic encryption techniques such as Secure Socket Layer (SSL) and the Transport Layer Security (TLS) for security is mandatory. Network layer provides significant protection against traditional network security issues, such as IP spoofing, port scanning, packet sniffing, etc.

Regular backups along with advanced encryption methods of those backups are necessary for preventing possible leaching of receptive information. Cloud environment contains and manages data from multiple users and organizations which can potentially jeopardize collective data with just one outsider or insider breach.

Another security concern is integrity of data which can be a challenge to maintain in cloud environment considering different technologies and systems for manipulating data across different platforms. Author of [24] suggests following the ACID (atomicity, consistency, isolation and durability) properties to ensure data integrity and also designating hash values for sets of data.

Cloud environment supports application development and execution for various types of software components and frameworks. This imposes the problem of Web application security where risk of failure or security breach can potentially affect many users. One way of removing this potential risk is the security management on the application level by ensuring that the application is completely handled in the cloud.

In Fig. 5 the example of security layer which manages access rights, per index, per user; admin, search only and custom API keys with advanced access control lists and per user security filters is given.



Fig. 5. Security layer in Search as a Service (SaaS) (Adapted from [23])

VI. CONCLUSION

Search as a Service (SaaS) is a newly established cloud service which due to new technologies had positive spiralling results. Although it offers great benefits in the search area it is a surprising fact that there is only less than a dozen companies that provide this service. From the current state and available research it is clear that the

further development moves towards Intelligent Search as a Service (ISaaS) utilizing the power of the cloud environment such as Coveo Company is currently developing. Their contribution to these services has promising anticipations for developing of easily and securely unify data from disparate sources across the IT ecosystem (cloud and on-premise), create ubiquitous search experiences wherever people work, and recommend the most relevant insights from everywhere directly into the context of customers and employees. Search as a Service (SaaS) is rapidly being implemented in major Web site databases such as Vevo, IBM, Amazon, Netflix etc. This paper provides the contribution of better understanding the underlying principles and methods of service functioning with clear advantages that it has to offer to modern data searching and retrieval along with security concerns.

REFERENCES

- [1] G. Zhao, C. Rong, J. Li, F. Zhang and Y. Tang, "Trusted data sharing over untrusted cloud storage providers", In *Proc. CloudCom-2010*, November-December 2010, pp. 97-103.
- [2] M.J. Woodruff, T.W. Simpson and P. Reed, "Many-objective visual analytics: In search of search-as-a-service", In *Proc. AIAA 2014-2729*, June 2014, paper 2729.
- [3] G. Shu, X. Xu and H. Hua, "A Web service search approach based on semantic and search engine", In *Proc. ICPCA-2011*, October 2011, pp. 484-489.
- [4] P. Rodriguez-Mier, M. Mucientes and M. Lama, "Automatic web service composition with a heuristic-based search algorithm", In *Proc. ICWS-2011*, July 2011, pp. 81-88.
- [5] Z. Pervez, A.A. Awan, A.M. Khattak, S. Lee and E.-N. Huh, "Privacy-aware searching with oblivious term matching for cloud storage", *J. Supercomput.*, vol. 63, no. 2, pp. 538-560, 2013.
- [6] A.H.H. Ngu, J. Ma, Q.Z. Sheng, L. Yao and S. Julian, "ServiceExplorer: A similarity-based Web service search engine", In *Proc. SIGIR-2014*, July 2014, pp. 1251-1252.
- [7] E.C. Micheli, G. Margaritis and S.V. Anastasiadis, "Efficient multi-user indexing for secure keyword search", In *Workshop Proc. CEUR-2014*, March 2014, pp. 390-395.
- [8] E.C. Micheli, G. Margaritis and S.V. Anastasiadis, "Permission-based index clustering for secure multi-user search", *Trans. Data Priv.*, vol. 8, no. 1, pp. 29-53, 2015.
- [9] J. Lee and S. Un, "Digital forensics as a service: A case study of forensic indexed search", In *Proc. ICTC-2012*, October 2012, pp. 499-503.
- [10] A. Kaci, T. Bouabana-Tebibel and Z. Challal, "Access control aware search on the cloud computing", In *Proc. ICACCI-2014*, September 2014, pp. 1258-1264.
- [11] R. Hu, W. Dou and J. Liu, "A personalised search approach for web service recommendation", *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, no. 2, pp. 83-95, 2013.
- [12] Y. Duan, G. Fu, N. Zhou, X. Sun, N.C. Narendra and B. Hu, "Everything as a Service (XaaS) on the cloud: Origins, current and future trends", In *Proc. CLOUD-2015*, June 2015, pp. 621-628.
- [13] H. Dong, F.K. Hussain and E. Chang, "A service search engine for the industrial digital ecosystems", *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2183-2196, 2011.
- [14] F. Chen, J. Hong, R. Zheng and M. Jiang, "Research on access control during search encrypted data", *Lecture Notes in Electrical Engineering (LNEE)*, vol. 155, pp. 677-682, 2012.
- [15] D. Çelik and A. "Elçi A semantic search agent approach: Finding appropriate semantic web services based on user request term(s)", In *Proc. ICICT-2005*, December 2005, pp. 675-688.
- [16] A. Bozzon, M. Brambilla, F. Corcoglioniti and S. Vadacca, "A service-based architecture for multi-domain search on the Web", *Lecture Notes in Computer Science (LNCS)*, Vol. 6470, pp. 663-669, 2010.
- [17] T. Bouabana-Tebibel and A. Kaci, "Parallel search over encrypted data under attribute based encryption on the Cloud Computing", *Comput. Secur.*, vol. 54, pp. 77-91, 2015.
- [18] A. Both, A.-C.N. Ngomo, R. Usbeck, D. Lukovnikov, C. Lemke and M. Speicher, "A service-oriented search framework for full text, geospatial and semantic search", In *Proc. SEMANTiCS-2014*, September 2014, pp. 65-72.
- [19] P. Dašić, J. Dašić and B. Crvenković, "Applications of the Search as a Service (SaaS)", *Bulletin of the Transilvania, University of Braşov, Series I: Engineering Sciences*, vol. 9, no. 1, 2016.
- [20] P. Dašić, J. Dašić and B. Crvenković, "Some examples of video surveillance as a service applications", In *Proc. SIMPRO-2016*, October 2016, pp. 367-370.
- [21] P. Dašić, J. Dašić and B. Crvenković, "Service models for cloud computing: Video Surveillance as a Service (VSaaS)", *Bulletin of the Transilvania, University of Braşov, Series I: Engineering Sciences*, vol. 9, no. 1, 2016.
- [22] D. Tunkelang, "Faceted Search", *Synth. Lect. Inf. Concepts, Retrieval, Serv.*, vol. 1, no. 1, pp. 1-80, Jan. 2009.
- [23] Algolia technical documentation. 2016. Available on [<https://www.algolia.com/doc>].
- [24] A.A. Soofi, M.I. Khan, R. Talib and U. Sarwar, "Security issues in Search as a Service (SaaS) delivery model of cloud computing", *International Journal of Computer Science and Mobile Computing (IJCSM)*, vol. 3.no. 3. pp. 15-21, March 2014.

AUTHORS PROFILE

	<p>Predrag Dašić was born in Peć, Serbia, in 1958. He received the M.Sc. in Mechanical Engineering from the University of Pristina in 1982. He obtained postgraduate diploma for Candidate of sciences in mechanical engineering at the University of Belgrade in 1985. He was elected for Hon.D.Sc. (Honorary Doctor of Science) at Donbass State Engineering Academy (DSEA) from Kramatorsk (Ukraine) in 2010. He is currently working as a professor at High Technical Mechanical School of Professional Studies in Trstenik (Serbia) and as technical editor in SaTCIP Publisher Ltd. in Vrnjačka Banja (Serbia). His research interests include: production engineering, reliability, statistical analysis, database management system (DBMS), Web design and etc. He is a member of NYAS, UTA, AINU, IA-E, IAAST, ISAET, SKANU, WASET, WARSE, AAAS, INNS and ENNS, IASC, ISSI, IAEng, AMP and etc. He published more than 550 papers in various international journals and delivered presentations in various national and international conferences, author of 5 books, editor of 40 proceedings and 3 patents. His paper are cited in more than 100 articles in WoS and Scopus citation databases, and has more than 500 citations on Google Scholar (GS) and more than 120 citations in doctoral thesis and master thesis. He has developed 14 scientific software applications, for the application of theoretical and practical methods in solving practical engineering problems, which are successfully applied in practice and developed 4 Web sites. He is a chairman of organization committee of two international conferences RaDMI and EMoNT and a member of scientific committees of more than 50 international conferences. He is an editor-in-chief of two international journals JRaDMI and JEMoNT and a member of editorial board of more than 20 international journals.</p>
	<p>Jovan Dašić was born in Kruševac, Serbia in 1985. He graduated at Technical Faculty Čačak, University of Kragujevac in 2012. He is currently finishing M.Sc at Faculty of Machining Kraljevo, Serbia. He is currently working as Web designer in SaTCIP Publisher Ltd. in Vrnjačka Banja, Serbia. His research interests include: Web design, Internet technologies and media production. He has published 20 papers in national and international journals and conferences. He has developed 15 Web sites.</p>
	<p>Bojan Crvenković was born in Tetovo, Republic of Macedonia in 1985. He received his B.Sc at High Technical Mechanical School of Professional Studies in Trstenik in 2009. He is currently specializing in Java programming at IT Academy Belgrade, Serbia. He is currently working as a research assistant in SaTCIP Publisher Ltd. in Vrnjačka Banja, Serbia. His research interests include: OO programming, Web services, Internet technologies. He has published 10 papers in national and international journals and conferences. He has developed 6 Web sites.</p>