PANEVĖŽIO KOLEGIJA ISSN 2029-1280, eISSN 2669-0071. Taikomieji tyrimai studijose ir praktikoje – Applied Research in Studies and Practice, 2022, 18.

COMPONENT-BASED ARCHITECTURE OF IOT SEARCH ENGINE

Vaidas Giedrimas, Gediminas Bačkys[a]

[a] *Panevėžio kolegija/University of Applied Sciences, Lithuania*

Abstract. When the number of IoT devices is growing exponentially, the demand for the search engine of "things" is as big as for the search engine of web pages in the 1990s. Several pilot projects are already running, they differ by their internal architecture and the algorithms for discovery, search, and representation. Early results are as promising as unsettling because they expose the challenges and treats. When web pages are uniform in some extent, the IoT network is highly heterogeneous and dynamic. We argue that because of the nature of IoT, we cannot just reuse the principles of websites search engines. Moreover, the usage of only one algorithm for device discovery or its property search can become inefficient soon. The quality of IoT search can be improved either by using AI and machine learning techniques or dynamically reuse the elements of 2 or more IoT search engines. This paper presents the software architecture of IoT search engines, based on software components, which enable the combination of the advantages of different search engines and their algorithms.

Keywords: Search engine, Internet of Things, CBSE, Software services

INTRODUCTION

When the number of IoT devices is growing exponentially, the demand for the search engine of “things” is as big as for the search engine of webpages in the 1990s (Gubbi, Buyya, Marusic, Palaniswami, 2013; Fathy, Barnaghi, Tafazolli 2018; Tran, Sheng, Babar, Yao 2017). Early results are as promising as unsettling because they expose the challenges and threats. When webpages are uniform to some extent, the IoT network is highly heterogeneous and dynamic. The source of the data for a typical web search engine is the webpages, their metadata, and content. However, IoT content is far more complex as devices provide very dynamic data. Instead of static content they usually provide data streams. On other hand, even the type of data can change. For example, a temperature sensor can provide floated point values (temperature), when it is working, a boolean value (false) when it is out of service, or the textual content (via its representative). Because of the nature of IoT, we cannot just reuse the principles of web page search engines.

To deal with such dynamicity, in software engineering it is common to reuse software components and/or software services. Component-based software engineering (CBSE) uses modules of higher granularity – the components. This enables us to reduce the time required for the development and testing of component-based software significantly. Components are “hot swap” modules in software similar to the storage disks, power supplying and other modules are in contemporary hardware (mainly servers). As all the communication between software components is organized via interfaces, each software component can be easily and quickly replaced by another one, which implements the same interface. After this, there is no need to perform testing of the overall system. A component-based system can be considered highly reliable and easily maintainable (Vale et al. 2016).

The objective of the research is to present the software architecture of IoT search engines, based on software components, which enable the combination of the advantages of different search engines and their algorithms.

Research methods: analysis of scientific literature, data analysis.

The rest of the paper is organized as follows: Section 2 presents the state-of-art taxonomy of IoT search engines. Section 3 exposes our architecture of dynamic internet of things search engine based on existing IoTSE as software components. Section 4 shortly introduces related work in this area. Finally, the conclusions are made, and open questions are discussed.

TAXONOMY OF IOT SEARCH ENGINES

Each IoT search engine must be able to perform at least two activities: to discover an activity and to search for activity. During the process of discovery activity, the engine retrieves the list of IoT devices by scanning local/global environment, IoT device-related websites or cached databases. According to (Tranet al. 2019), more than 90% of the engines include discovery activities. After this list is made, the second activity - searching - is running. Its goal is to filter the former list by rejecting all the devices which properties do not conform with the initial query. IoT search engines can be classified according to their meta-path. Meta-path

reveals the source of the data for discovery and search and the target of these activities. Tranet et al. (2019) distinguish 8 classes of IoTSE

1) R → R class IoTSE
2) D + R → T → R class
3) D → D
4) R → T → R + D
5) D → T → R
6) F → F
7) R → T → F
8) S → S

where R is Representative of IoT device (e.g. dedicated website for its monitoring and control), D - Dynamic IoT content (e.g. not usual data stream), T - IoT device (aka Thing).

Table 1. IoT Search Engines

Engine	Class	Source	URL
Shodan	N/A		https://www.shodan.io
ThingFul	DRTR?		https://www.thingful.net
IoTCrawler	DD?		https://iotcrawler.eu
ForwarDS-IoT	RR	(Gomes et al., 2015)	
DiscoWoT	RR	(Mayer ,Guinard, 2011)	
ThingSeek	N/A	(Shemshadi, Sheng, Qin, 2016)	
IoT-SVK	DRTR	(Jin et al., 2011)	
CASSARAM	DD	(Perera et al., 2013)	
Snoogle	RT-R+D	(Wang, Tan, Li, 2010)	
ASAWoO	FF	(Mrissa, Medini, Jamont, 2014)	https://liris.cnrs.fr/asawoo
(no brand name)	RT	(Kamilaris, Papakonstantinou, Pitsillides, 2014)	
Microsearch	SS	(Tan, Sheng, Wang, Li, 2009)	

1st class (R-R) of IoT search engines is highly influenced by web search engines. The information about devices is taken from representatives only. No direct data gathering from IoT devices at the moment of the search. The second class (D+R-TR) of search engines gets data not only from representatives of things but (what is more specific) from the IoT devices directly as dynamic streams. The result of 2nd class IoTSE is the representatives. In other words, R-R class can be considered as a subset of D+R-TR, with zero D component. 3rd class (D-D) goes beyond D+R-TR class and gets the data from the very end sensors of Things, not just from the public observable states of things. 4th class (RT-R+D), according to (Tranet al. 2019), is an extension of the R-R class. What is interesting, that search engines from this class return not only links to IoT representatives but the data stream from them as well. 5th class (D-T-R) is a little bit similar to 2nd and 3rd classes. Typically, they work using SPARQL queries over contextual information. As (Tranet al. 2019) observed, in contrast with 2nd class IoT search engines, 5th class engines select things only by the data streams, not considering any other features (e.g., representatives). 6th class (F-F) is very interesting as the IoT devices are analyzed not by their representatives or data stream, but by their functionality. The functionality of things is exposed in a shared ontology. 7th class (RTF) of engines selects devices by its representatives. Its results include the functionality of selected things. As pointed in (Tranet al. 2019), these functionalities are presented as RESTful Web services and are capable of self-describing with specifications written in the Web Application Description Language (WADL). In our context –Component-based software engineering (CBSE) – it is a very promising fact. And 8th class of IoTSE (SS) queries on cached static information from IoT devices. In contrast to 3rd class, these IoTSE, do not acquire the newest data from IoT devices nor perceive dynamic data streams. However, then can be easily applied in the fields where the device state does not change often. Examplxes of IoT search engines are listed in Table 1.

COMPONENT-BASED ARCHITECTURE

The idea of combining IoT with software components and services is not very new. The group of (Liu, Morisset, Stolz 2010) has been working on software-hardware components fusion a long time before the rise of IoT concept. In another reference, Ruppen et al. (2015) presented the system based on model-driven architecture (MDA) which helps to bind IoT devices and RESTfull web services in a semi-automated way.

The component in (Ruppen et al. 2015) work is the pair of IoT device - RESTFull service. In both cases, the component is as a "piece" of the system having its hardware and software parts. In CBSE each component must have at least one interface (or Provided interface) which it is implemented, and (optionally) can have Required interface, which declares what interfaces are needed from other components to work properly. In notation provided interface often is represented as a "lollipop", while the required interface – as a sink. In most of the component-based technologies (e.g. .NET, EJB, CORBA) it exists also a Framework, or middleware, which encompasses main data structures, services (e.g. networking, DB) and ensures more smooth communication between components via their interfaces.

We focus on the architecture of IoT search engine, not on IoT system, that the concept of the component in our approach is defined differently. The component in our context is a reusable software module, having explicit interfaces, capable of discovering IoT devices and/or getting its data. It is the subject of third part composition and is strictly dependent on the middleware/framework.

Having this definition, we can describe the overall architecture of IoTSE (see Figure 1):

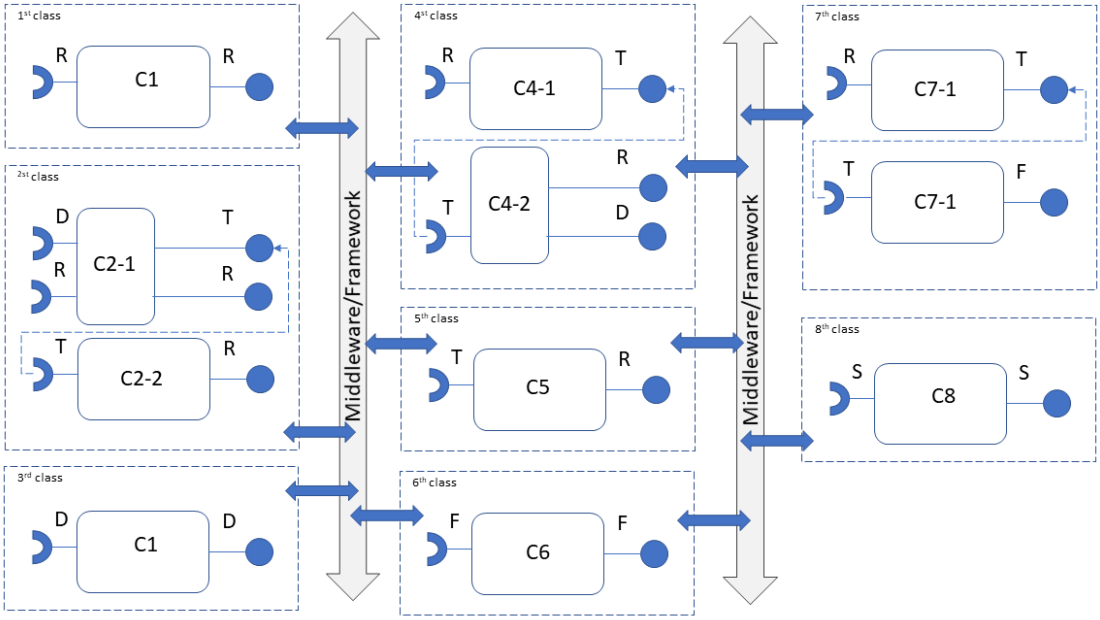

Figure 1. **The architecture of component-based IoTSE**

1) The architectural parts are components. Each component by itself is an IoT search engine. For example, we can include any IoTSE from Table 1 if only it meets the required interface.
2) The connector between the components are interface-to-interface calls. Here is worth mentioning that the source for the definition of the interfaces is the taxonomy of possible IoTSE's (see Section 2). Moreover, the Taxonomy just provides the Type of data. Bearing in mind that the data itself can be different, here we meet not only single interfaces but rather the sets of interfaces.
3) Search engine implements a subset of the interfaces in the set. Only in rare cases, the component will implement all the interfaces from the set. However, the number of IoTSE is big enough to orchestrate a proper combination of a few search engines to cover all required demand (defined by required interface).
4) In the most general case, we assume that some IoTSE can be considered not as an atomic component, but as the set of components. For example, the search engines belonging to the 2nd, 4th, and 7th classes can perform different tasks with different interfaces (see Fig. 1), so they can be used partially as well.
5) As it is common for component-based software systems, in our software architecture we will use a framework/middleware as well. This part of the system can be considered as a backbone, orchestrating overall communication and replacement procedures.
6) For the smooth usage of this architecture, we assume that the User interface of this "umbrella for IoTSEs" can be considered as a separate component (not shown in Figure 1).

As far as the authors of this paper know, at the time of writing paper, there is no system which has already implemented component-based architecture for the internet of things search engine.

RELATED WORK

During the time of writing this paper, we have no information about any other component-based approach on IoT search engines. However few papers touch this topic partially.

Glatard, T. et al. (2017) presents the software architectures for integration of the workflow engines in science gateways. Typically, a science gateway is used to share resources within a community and to provide increased performance and capacity through facilitated access to storage and computing power. Some science gateways (e.g. wiki-based stochastic programming and statistical modeling system for the cloud ((Giedrimas et al. 2016) or CodeOcean portal) enable even to collaborate remotely on the code level and elaborate scientific software in the crowdsourcing style. The workflow engine, as presented in (Glatard et al. 2017), typically describes applications using a domain-specific language including data and control flow constructs, parallelization operators, visual edition tools, links with application repositories, provenance recording, etc. The idea to use the (reusable) workflow engines in the overall science gateway is architecturally similar to our approach. Moreover, the authors of (Glatard et al. 2017) workflow engines called as software components while presenting 6 software architectures: Tight integration, Service invocation, Task encapsulation, Pool model, Nested workflows with service invocation and Conversion, and Workflow conversion with service invocation. In our approach, the overall backbone is the search engine of IoT (instead of the science gateways), and smaller parts - software components implementing different IoT search and discovery algorithms.

Other papers have weak (or negative) relations with software components; however, they are highly related to software architectures (Ozcan et al. 2012), (Cambazoglu et al. 2007) and/or search engines (Ozcan et al. 2012; Cambazoglu et al. 2007; Kejriwal 2021). The most interesting case is (Rhayem, Mhiri, Gargouri 2020) presenting the problem of IoTSE from the semantic web perspective. We believe that the results of Rhayem, Mhiri, and Gargouri (2020) could be blended with our approach and improve it.

CONCLUSIONS AND FUTURE WORK

After the analysis of the state of the art in IoT search engines, we came to the following conclusions:

1. Even though the market of IoTSE's is new and growing, a sufficient number of different search engines are developed already. Fortunately, IoTSEs are not unique and can be classified into a limited number of classes. Each class of the IoTSE can define an interface and each IoTSE from this class can be considered as a component implementing this interface.
2. Presented component-based architecture for the internet of things search engine improves current systems in flexibility, time-to-market, and robustness aspects.
3. From the software engineering perspective, our approach exposes one more possible use case of component-based technology.
4. Our future work includes the improvement of presented architecture by implementing the ideas presented in (Rhayem, Mhiri, and Gargouri, 2020) (the aspect of semantic web) and (Glatard et al., 2017) (the performance modeling and testing).

REFERENCES

Buyya, R., and Dastjerdi, A. V. (2016). *Internet of Things: Principles and Paradigms*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Cambazoglu B. B., and Baeza-Yates, R. (2016). Scalability and efficiency challenges in large-scale web search engines. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16. New York, NY, USA:ACM, p. 1223–1226.

Cambazoglu, B. B. et al. (2007). Architecture of a grid-enabled web search engine. *Information Processing and Management*, 43(3), 609–623, Special Issue on Heterogeneous and Distributed IR. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457306001877

Fagroud, F. Z. et al. (2019). What does mean search engine for IoT or IoT search engine. In *Proceedings of the 4th International Conference on Big Data and Internet of Things*, ser. BDIoT'19. New York, NY, USA: ACM.

Fagroud, F. Z. et al. (2020). IoT search engines: Exploratory data analysis. *Procedia Computer Science*, vol. 175, 572–577.

Fathy, Y., Barnaghi, P. and Tafazolli, R. (2018). Large-scale indexing, discovery, and ranking for the internet of things (IoT), *ACM Comput. Surv*., 51(2).

Giedrimas, V. et al. (2016). Wiki-based stochastic programming and statistical modelling system for the cloud. *International Journal of Advanced Computer Science and Applications*, 7(3).

Glatard, T. et al. (2017). Software architectures to integrate workflow engines in science gateways. *Future Generation Computer Systems*, 75, 239–255.

Gomes, P. et al. (2015). A federated discovery service for the internet of things. In *Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT*, ser. M4IoT 2015. New York, NY, USA: ACM, p. 25–30.

Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions, *Future Generation Computer Systems - The International Journal of E-Science*, 29(7), 1645–1660.

Hadjilambrou, Z. et al. (2019). Comprehensive characterization of an open-source document search engine. *ACM Trans. Archit. Code Optim.*,16(2).

Ismail, B. I. et al. (2017). Reference architecture for search infrastructure. In *7th IEEE International Conference on Control System, Computing and Engineering*, 115–120.

Jin, X. et al. (2011). Where searching will go in internet of things? In *2011 IFIP Wireless Days*, 1–3.

Kamilaris, A., Papakonstantinou, K. and Pitsillides, A. (2014). Exploring The Use of DNS as a Search Engine for the Web of Things. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 100–105.

Kejriwal, M. (2021). A meta-engine for building domain-specific search engines. *Software Impacts*, vol. 7, p. 100052.

Liu, Z., Morisset, C., and Stolz, V. (2010). RCos: Theory and tool for component-based model driven development. In *Fundamentals of Software Engineering*, F. Arbab and M. Sirjani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 62–80.

Lobo, J., Firmenich, S., Rossi, G., Defosse, N., and Wimmer, M. (2017). Web of things augmentation. In *Proceedings of the Eighth International Workshop on the Web of Things*, ser. WoT 2017. New York, NY, USA: ACM, 11–15.

Matheu, J. L. Hern´andez-Ramos, A. F. Skarmeta, and G. Baldini, (2020). A survey of cybersecurity certification for the internet of things. ACM Comput. Surv., 53(6), Dec.

Mayer S., and Guinard, D. (2011). An extensible discovery service for smart things. In *Proceedings of the Second International Workshop on Web of Things*, ser. WoT '11. New York, NY, USA: ACM.

Mrissa, M., Medini, L. and Jamont, J. (2014). Semantic discovery and invocation of functionalities or the web of things. In *2014 IEEE 23rd International WETICE Conference*, 281–286.

Ozcan, R. et al. (2012). A five-level static cache architecture for web search engines. *Information Processing and Management - Large-scale and Distributed Systems for Information Retrieval.*, 48(5), 828–840.

Perera, C. et al. (2013). Context-aware sensor search, selection and ranking model for internet of things middleware. In *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 1, 314–322.

Rhayem, A., Mhiri, M. B. A., and Gargouri, F. (2020). Semantic web technologies for the internet of things: Systematic literature review. *Internet of Things*, vol. 11, p. 100-206. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520300421

Ruppen, A., Pasquier, J., Meyer, S., and Ruedlinger, A. (2015). A component-based approach for the web of things. In *Proceedings of the 6th International Workshop on the Web of Things*, ser. WoT '15. New York, NY, USA: ACM.

Shemshadi, A., Sheng, Q. Z., and Qin, Y. (2016). Thingseek: A crawler and search engine for the internet of things. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '16*. New York, NY, USA:ACM, 1149–1152.

Shepherd, S. J. (2007). Concepts and architectures for next-generation information search engines. *International Journal of Information Management*, 27(1), 3–8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026840120600082X

Singh, R. et al. (2020). Internet of Things with Raspberry Pi and Arduino, 1st ed. CRC Press,.

Tan, C. C., Sheng, B., Wang, H., and Li, Q. (2009). Microsearch: When search engines meet small devices. In *Proceedings of the 6th International Conference on Pervasive Computing, ser.* Pervasive '08. Berlin, Heidelberg: Springer-Verlag, 93–110.

Tran, N. K. et al. (2017). Searching the web of things: State of the art, challenges, and solutions. *ACM Comput.Surv.,* 50(4).

Tran, N. K. et al. (2019). Internet of things search engine, *Commun. ACM*, 62(7), 66–73.

Tran, N., Sheng, Q., Babar, M. and Yao, L. (2017) A kernel-based approach to developing adaptable and reusable sensor retrieval systems for the web of things. In *Proceedings of 18th International Conference on Web Information Systems Engineering (WISE 2017)*, LNSC 10569. United States: Springer, Springer Nature, 315–329.

Vale, T. et al. (2016). Twenty-eight years of component-based software engineering. *J. Syst. Softw*., 111(C), 128–148.

Wang, H., Tan, C. C., and Li, Q. (2010). Snoogle: A search engine for pervasive environments. *IEEE Transactions on Parallel and Distributed Systems*, 21(8), 1188–1202.

