Searching the Web of Things: State of the Art, Challenges, and Solutions

NGUYEN KHOI TRAN, The University of Adelaide
QUAN Z. SHENG, Macquarie University
MUHAMMAD ALI BABAR, The University of Adelaide
LINA YAO, The University of New South Wales

Technological advances allow more physical objects to connect to the Internet and provide their services on the Web as resources. Search engines are the key to fully utilize this emerging Web of Things, as they bridge users and applications with resources needed for their operation. Developing these systems is a challenging and diverse endeavor due to the diversity of Web of Things resources that they work with. Each combination of resources in query resolution process requires a different type of search engine with its own technical challenges and usage scenarios. This diversity complicates both the development of new systems and assessment of the state of the art. In this article, we present a systematic survey on Web of Things Search Engines (WoTSE), focusing on the diversity in forms of these systems. We collect and analyze over 200 related academic works to build a flexible conceptual model for WoTSE. We develop an analytical framework on this model to review the development of the field and its current status, reflected by 30 representative works in the area. We conclude our survey with a discussion on open issues to bridge the gap between the existing progress and an ideal WoTSE.

CCS Concepts: • **Information systems** → **Web search engines**; *Information retrieval*;

Additional Key Words and Phrases: Search, discovery, retrieval, Web of Things, Internet of Things

ACM Reference format:
Nguyen Khoi Tran, Quan Z. Sheng, Muhammad Ali Babar, and Lina Yao. 2017. Searching the Web of Things: State of the art, challenges, and solutions. *ACM Comput. Surv.* 50, 4, Article 55 (August 2017), 34 pages.
https://doi.org/10.1145/3092695

1 INTRODUCTION

Our world is becoming a resource library for software applications. Advances in embedded computing and low-power wireless communication bring Internet connectivity to physical objects, forming the Internet of Things (IoT) [30]. By reusing technologies and techniques of the World Wide Web, the information and services of these objects (e.g., sensor streams, actuating functions) can be provided on the Web as resources for human users and cyber-physical applications [13]. By bringing the familiarity of the Web to the interaction with physical objects, the emerging Web of Things (WoT) is expected to be the enabling factor to bring cyber-physical applications to the public and change the way we live [21, 52].

Authors' addresses: N. K. Tran; email: nguyen.tran@adelaide.edu.au; Q. Z. Sheng; email: michael.sheng@mq.edu.au; M. A. Babar; email: ali.babar@adelaide.edu.au; L. Yao; email: lina.yao@unsw.edu.au.

https://doi.org/10.1145/3092695

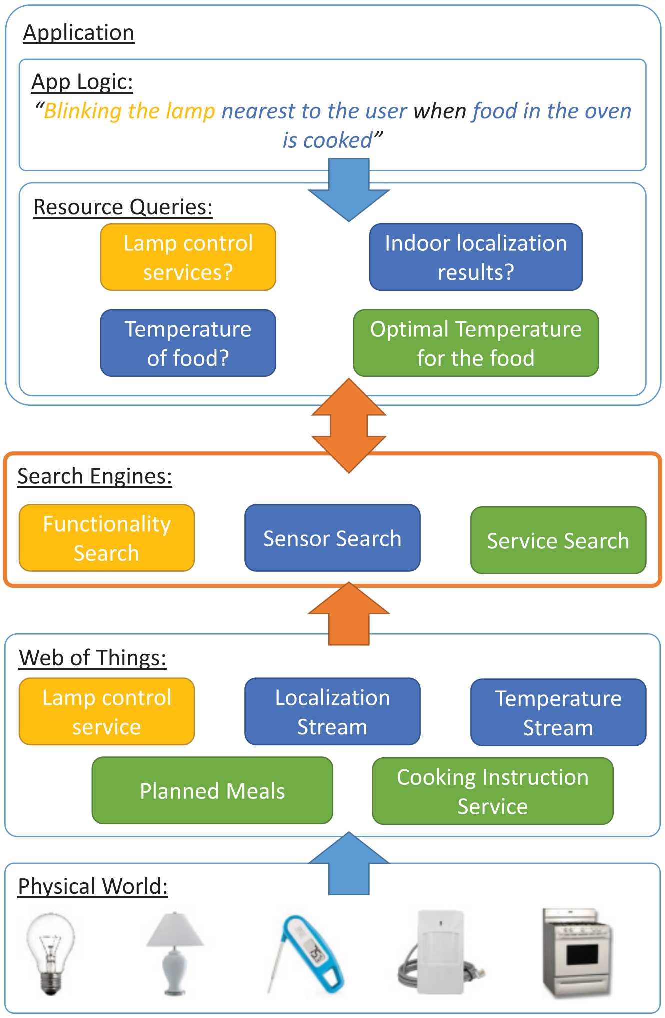


Fig. 1. Search engines as middle-ware to decouple application logic from resource retrieval in the Web of Things.

Web of Things Search Engines (WoTSE) are librarians of this emerging library. They bridge users and applications with required resources. For instance, consider a cyber-physical application in a smart home for elderly people that blinks the lamp closest to the house owner to notify that the meal in the oven is done (Figure 1). This application requires control service of lamps and lightbulbs in the house, a sensor stream from the meat thermometer, a stream of results from the installed indoor localization system, and a Web service showing the optimal temperature for the meal being cooked. Assuming that these resources are available, the task of developers is finding and linking them to the application logic. WoTSE decouples resource retrieval from the application. By querying an appropriate WoTSE, the application can retrieve resources needed for its operation without the manual configuration of developers. As long as the application has access to the WoTSE managing its deployment area, it can configure itself to work. And, as long as the WoTSE continues to manage changes of objects in its scope, the application always has access to the latest resources.

Research related to WoTSE begins from early 2000s and enjoys steady expansion ever since. It branches into different directions including object search, sensor search and functionality search. Essentially, WoTSE comprises of different types of systems, including unseen ones that will emerge when the adoption of WoT increases. This diversity complicates both the development of new WoTSE and the assessment of its state of the art. Therefore, a survey on WoTSE must focus on the whole field, not just only what happens within one type of WoTSE. Existing surveys either focus on one form of WoTSE [17, 46, 62] or focus on potential technical problems without considering the state of the whole field [65].

In this article, we perform a systematic survey on over 200 works related to WoTSE, with the focus on their diversity. Our contributions are fourfold:

- Proposing a conceptual model that describes WoTSE succinctly with resources involving in their query resolution process.
- Proposing a modular architecture as a reference to evaluate representative WoTSE prototypes.
- Reviewing the growth and state of the art of research and industrial efforts around WoTSE.
- Identifying open issues and potential solutions for the diversity challenge.

The remaining of this article is organized as follow. Section 3 introduces WoT and WoTSE concepts. Section 4 presents our proposed conceptual model and reference architecture for WoTSE. In Section 5, we present the analytical framework of our survey, which is built on our proposed models. We apply this framework on academic and industrial efforts and present the results in Sections 6 and 7, respectively. Finally, we discuss prominent open research issues around WoTSE in Section 8.

2 RELATED WORKS

A range of surveys on WoT and its closely related concept IoT exist in the literature. Early surveys [3, 4, 30, 34] serve as road-maps to realize IoT and WoT. They cover visions, definitions, enabling technologies and propose potential research directions. Later surveys focus on more specific usages. Reference [20] reviews WoT from the cloud computing perspective; Reference [42] reviews the enabling technologies to extend IoT into Cognitive IoT; Reference [38] surveys the integration of humans' social network into IoT to form a Social IoT; Reference [60] explores different use cases of IoT in smart cities and their enabling technologies; and Reference [48] approaches IoT from the politics and policy perspective. In these surveys, WoTSE either receives a brief discussion as a potential research topic [30] or a short introduction presenting some representative works [43].

A small number of surveys specifically on WoTSE exist in the literature. They either focus on one type of WoTSE or listing potential research problems without considering the state of the field. Reference [46] analyze seven prototypes on nine dimensions that focus on the ability to handle real-time, local sensor queries. Reference [62] performs a similar analysis with six prototypes on 14 dimensions. Reference [17] approaches WoTSE from perspective of EPCglobal's Discovery Service. They analyze five works on nine dimensions. Finally, Reference [65] evaluates 49 works, including both WoT-specific prototypes and results from other fields that are expected to be applicable to WoT, on nine dimensions. Selected works are analyzed on their basic operating principles, data representation, and type of searched content.

Our survey addresses the limitations of the existing surveys. We retrieve over 200 related works to build a flexible model for describing WoTSE and a modular architecture for assessing their implementation. The resulting analytical framework from our models is used to assess the growth and the state of all major types of WoTSE in the literature. Table 1 presents the comparison between our survey and the existing ones.

Table 1. Comparison Between Our Work and Existing Surveys on WoT Search Engine

	Prototypes	Dimensions	Assessment Focus
Romer, et al. 2010	7 prototypes	9 dimensions	Ability of WoTSE to handle real-time sensor data and perform local search
Zhang, et al. 2011	6 prototypes	14 dimensions	Ability of WoTSE to handle real-time sensor data and perform local search
Evdokimov, et al. 2010	5 prototypes	9 dimensions	Maturity of Discovery Service architectures and prototypes
Zhou, et al. 2016	49 projects	9 dimensions	Technologies and techniques transferable to WoTSE
Our Survey	214 prototypes	24 dimensions	Forms, Implementation of WoTSE and the current state of the field

3 BACKGROUND

3.1 The Web of Things

The Web of Things (WoT) emerges from applying Web technologies to the Internet of Things to access information and services of physical objects. In WoT, each physical object possesses a digital counterpart that is commonly referred to as "Virtual Object" [9] or "Web Thing" [22]. These objects are built according to Representational state transfer (REST) architecture and accessed with HTTP protocol via RESTful API. A Web Thing can have an HTML or JSON representation, REST API to access its properties and actions, and an OWL-based semantic description.

Web Things are integrated into the Web in three ways. They can be hosted directly by Web Servers embedded into physical objects. With clever optimization, a Web Server can operate on an embedded computer with only 200 bytes of RAM and 7KB of code [13]. For objects that cannot be modified, their virtual objects can be hosted by the Web Server embedded in a gateway device, or a cloud service. In these arrangements, the gateway device translates traffics in HTTP into the proprietary communication of the physical object. These three modes of integration are presented in Figure 2. An overview of enabling technologies for bridging physical objects to the Internet is provided in Reference [36]

In WoT, applications interact with physical objects with the familiar HTTP prococol and RESTful API. This simplifies the access to physical objects, allowing them to be used in Web applications and merged with existing Web resources [13]. It enhances the creation of value-added cyber-physical services by exposing sensing and actuating capabilities to a global open market [52]. Essentially, WoT turns the real-world into a library of software resources that is accessible via the Web.

3.2 Discovery and Search in the Web of Things

Web of Things Search Engines (WoTSE) are "librarians" of WoT. They *discover* and gather WoT resources in a specific scope and allow users to *"search"* on these resources. For brevity and consistency, we use the term WoTSE for both systems designed specifically for WoT and IoT or telemetric solutions that can be adapted to WoT.

WoTSE appear in different usage scenarios with different forms and implementations in the literature:

- *Locating Physical Objects:* In early projects, WoTSE are commonly used to locate physical objects, which are tagged with passive RFID tags [27, 59] or sensor nodes [18, 32, 55].

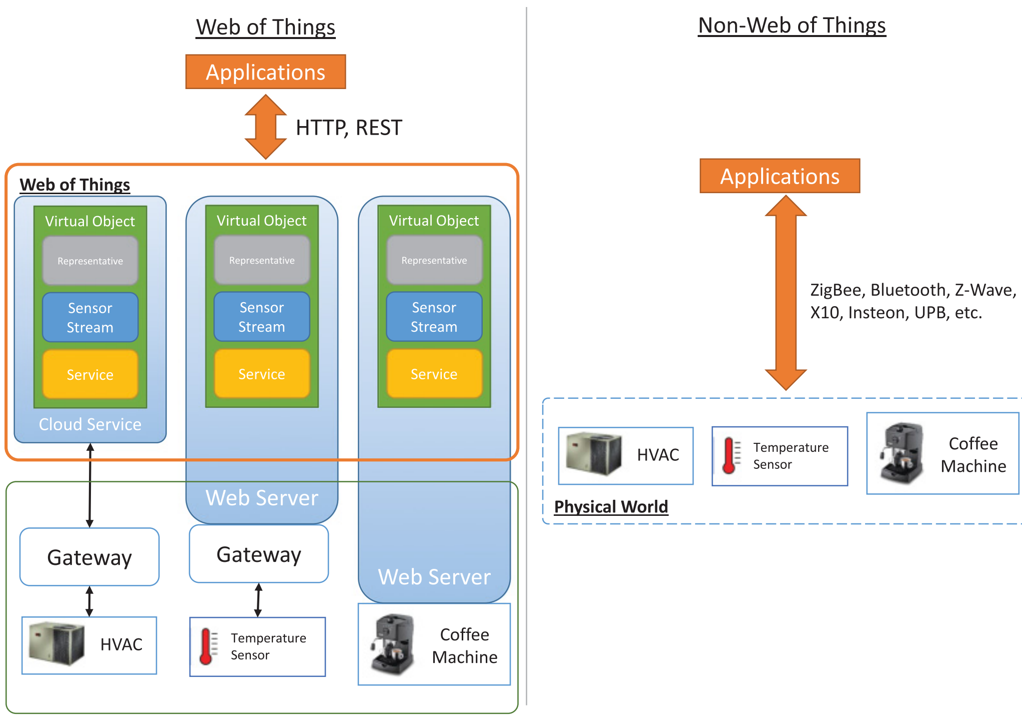


Fig. 2. Comparison between Web of Things and Non-Web of Things solution for accessing physical objects from an application.

- *Sensor Search:* GSN [1], CASSARAM [41] demonstrate the use of WoTSE for retrieving sensors based on their static meta-data and contexts, such as cost and reliability.
- *Finding Entity with Dynamic State:* Dyser [39] demonstrates a WoTSE that searches for physical objects (e.g., meeting room) based on their real-time states (e.g., "empty") derived from their sensor readings.
- *Finding Actuation Services:* Reference [10] demonstrates the use of WoTSE as a middle-ware for retrieving services offered by physical objects (e.g., changing lamp intensity).
- *Retrieving Data Records:* Prototypes of EPC Discovery Services [17] illustrate the use of WoTSE to retrieve data records relevant to an individual physical object. This problem was also investigated in Cooltown project [26].

Each form of WoTSE in the literature has its own characteristics and technical challenges. However, certain features are invariant between them. Therefore, we can build a common model to present majority of different WoTSE. We present this model in Section 4.1.

3.2.1 WoT Search vs Web Search. As content in WoT is accessible via the Web, WoT Search Engines are sometimes considered a minor extension of Web Search Engines. However, this is not the case, due to the unique features of WoT (Table 2). Existing Web Search Engines face following four issues in WoT.

First, WoT holds a vast amount of short, structured texts and non-text content (e.g., sensor streams, functionality), while Web Search Engines are optimized for long, unstructured texts. Therefore, text processing alone is not adequate for WoT. Second, WoT lacks the explicit links of the Web. Majority of relation between physical objects exists in the form of latent correlation

Table 2. Differences between the Web and the Web of Things

	Web	WoT
Content Type	Long, unstructured texts (i.e., Web Pages)	Numerical data; Short structured texts
Link Structure	Extensive, explicit link structure between pages (i.e., URL)	Latent links
Dynamicity	Stable; Long lifetime; Slow changing	Volatile; High update rate (up to 1,000,000 per second)
Scale	Over 1 billion Websites	Over 50 billion devices. Interactions happens in local areas

[58]. Therefore, both crawling and link analysis mechanisms (e.g., Page Rank) are not directly applicable to WoT. Third, WoT has a varying but high dynamicity. For instance, sensors in WoT update their content from once every several seconds to 1,000,000 times per second [43]. Therefore, storage and indexing mechanisms of Web Search Engines that assume slow changing content cannot cope with WoT. Finally, WoT is both larger and smaller than the Web. It is expected to contain over 50 billion devices by 2020 [61], while the Web currently holds only 1 billion Websites.[1] Yet, WoT applications interact with closely located resources for most of their life time. For instance, consider cyber-physical applications that interact with smart homes. Current Web Search Engines might not be able to scale up to serve over 50 billion devices. They are also not equipped to retrieve resources in the immediate vicinity of search users [21].

The stated issues imply that new techniques and mechanisms are required to realize WoTSE, despite the strong foundation laid by existing Web Search literature.

4 A MODEL FOR WEB OF THINGS SEARCH ENGINES

A model for WoTSE that provides succinct description of their operation and architecture is required to analyze their diverse literature. This model must fit naturally with majority of the existing projects and must be extensible to work with future, unseen types of WoTSE. We build our model base on over 200 existing works related to WoTSE in the literature. Section 4.1 presents "Meta-path"—our model for describing WoTSE. Section 4.2 presents our modular architecture for WoTSE.

4.1 Meta-path: The Signature of a WoTSE

The operation, usage purpose, and implementation of a WoTSE are decided by the type of resources that it uses for assessing query (i.e., *"Query Resource"*), for deriving search results (i.e., *"Result Resource"*), and the chain objects linking them. For instance, a search engine working with sensor streams uses different indexing and query assessment schemes comparing to a search engine working with physical functionalities. Therefore, they have different technical challenges and usage purposes. Based on this observation, we decided to model WoTSE with the types of resource that they use and the path between these resources. We call this model *"Meta-path"*:

$$QueryResType[Feature] + \ldots \Rightarrow Obj \rightarrow \ldots \rightarrow Obj \Rightarrow ResultResType + \ldots \quad (1)$$

Equation (1) presents the pattern of a meta-path, which consists of three parts. The first part ($QueryResType[Feature] + \ldots$ describes the types of resources utilized by a search engine to assess

[1] http://www.internetlivestats.com/total-number-of-websites/.

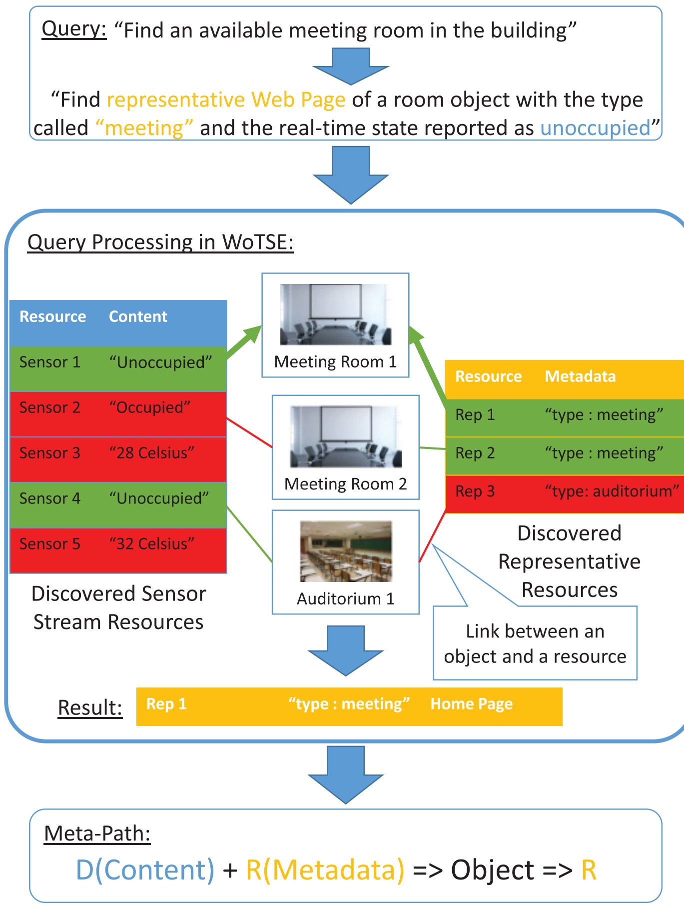


Fig. 3. Assessment of a query for available meeting room in a smart building and its related meta-path.

queries. For clarity, a meta-path also presents features of resources that involve in the query assessment. The part $\Rightarrow ResultResType + \ldots$ describes the types of resources used for building search results. The part $\Rightarrow Obj \rightarrow \ldots \rightarrow Obj$ describes the chain of objects linking query resources and result resources. The first object in the chain provides query resources, while the last object in the chain provides result resources. Links between objects can be extracted via their correlation [58]. This path can be zero-length, which denotes that a same set of resource is utilized for both assessing queries and building results. For instance, Web Search and Document Search systems have zero-length paths.

The actual query resolution of a WoTSE involves multiple concrete paths between discovered resources that are instantiated from its meta-path. Consider the resolution process for a query for available meeting room in a smart building (Figure 3) (e.g., Dyser system [39]). The result of this query is a set of digital representatives (e.g., Web page) of rooms in the building, which are of the type "meeting room" and have the state "available" reported by their sensors at the query time. The search engine has a set of sensor streams and digital representatives of physical objects created by a prior discovery process. The relations between sensor streams, digital representatives and objects are also recorded a priori. The first step of query resolution is matching the given query with content of sensor streams to find the ones reporting "available" state, and with metadata of representatives to find the ones belonging to meeting rooms. The second step is finding objects that have both matching sensor streams and digital representatives (i.e., "available meeting rooms"). Finally, representatives of these objects are selected as result resources to build search results. In this case, the search engine simply returns the list of Web pages. However, more complex processing such as aggregation or projection onto a map can be performed. The meta-path of our example search engine is $D(Content) + R(Metadata) \Rightarrow Object \Rightarrow R$. It is a common meta-path in the existing literature.

It should be noted that searching is more challenging in real-world scenarios. For instance, sensor streams might not report the state "available" explicitly, and the metadata of a room might not show its type as a meeting room. These problems must be countered by specific mechanisms of the search engine. However, the whole process is invariant.

A WoT resource is a mapping from a reference to a specific content in WoT (e.g., sensor stream, actuation service) at a specific instance of time. Formally, a resource is a four-dimensional vector $Res = (ID, Metadata, Representation, Content)$. ID denotes the reference assigned to a WoT content, which is commonly a URI. Alternatives are Electronic Product Code (EPC), Ubiquitous ID (uID) [28], and IPv6. $Representation$ denotes forms that a resource can represent itself, such as an HTML or JSON document. $Content$ denotes the content encapsulated by the resource. $Metadata$ describes this content with key-value pairs [41] or textual tags [59]. Semantic description [9] is an emerging form of metadata.

We organize resources into eight types according to their origin and content. A resource that is related to an individual object in the real world has physical origin. Otherwise, it has digital origin. The encapsulated content has four types:

- ***R**epresentative* denotes the virtual representation of physical entities in the digital world. Web pages are the most relevant for WoT. However, other forms of documents (e.g., XML, JSON) and database records are also acceptable.
- ***S**tatic Information* denotes the rarely-changed data held by an object. It can be archived sensor readings [57], files loaded by human users [53], or records of events related to an object.
- ***D**ynamic Information* denotes the frequently-changed data held by an object. Sensor readings are the most prominent form of dynamic information in WoT [43].
- ***F**unctionality* denotes the actuation services provided by an object.

By relying on the type of resources and the link between them, which dominates the operation and implementation of a search engine, our meta-path model provides a succinct description for WoTSE. Our model is also extensible by introducing new relations between objects. For instance, by giving WoTSE the ability to link room objects with human users, it can be extended to resolve queries for staffs who are using meeting rooms in a specific building. Because of these ability, Meta-path is used to model WoTSE in our survey.

4.2 An Architecture for WoTSE

The query resolution process that we introduced is common in existing WoTSE projects. Its implementation changes depending on the meta-path of each search engine, but its activities and their arrangement are invariant. Therefore, we model these activities as standalone modules for building an architecture for WoTSE. Figure 4 presents our modular architecture.

Modules in our architecture are organized into layers. Two lower layers handle discovery activities. Two upper layers handle search activities. Storage modules for resource collections and

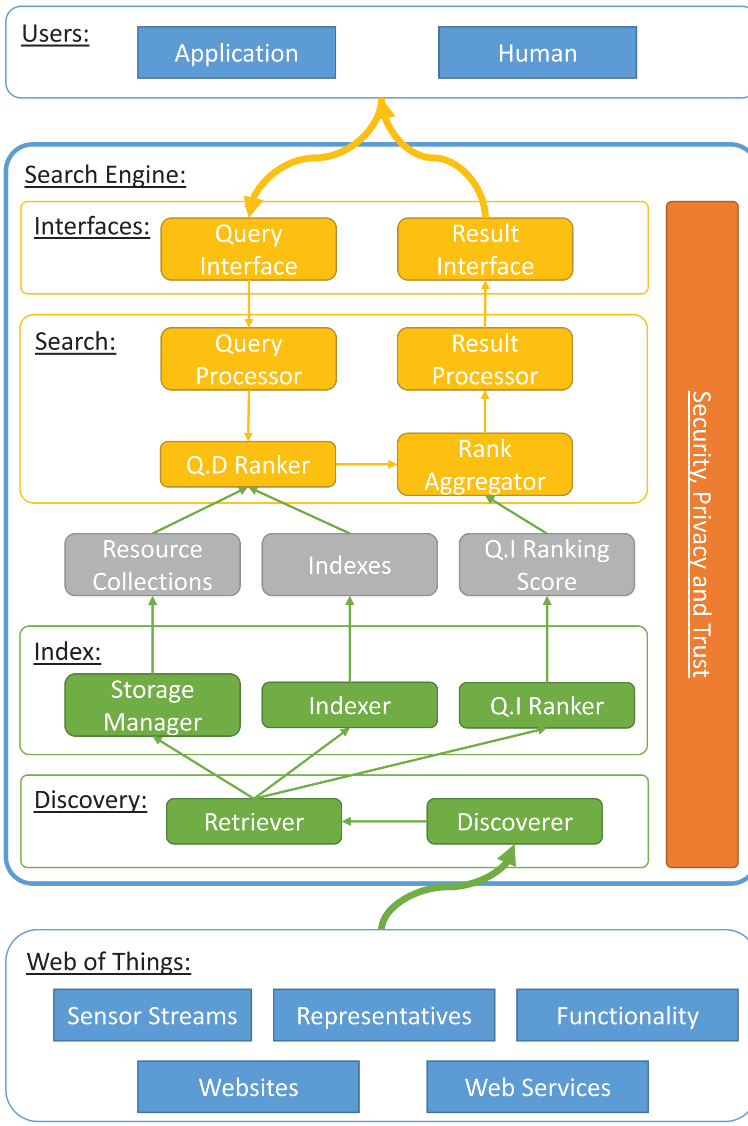


Fig. 4. A Modular Architecture for Web of Things Search Engines.

indexes link two set of layers. The whole system is protected by security, privacy, and trust assessment measures, which are grouped into a vertical layer.

Discovery Layer interfaces WoTSE with resources in WoT. The *Discoverer* module detects resources specified in the meta-path of the search engine, in a certain physical scope. It can also be extended to discover relations between objects and resources. The *Retriever* module collects the discovered resources and passes them to the upper layer.

Index layer stores and indexes resources with its *Collection Manager* and *Indexer* modules. This layer also possesses *Query Independent (Q.I) Ranker* to rank resources according to their natural order, independent from user queries. For instance, Page Rank [40] is a form of Q.I Ranking.

Depending on the timing between discovery and search activity, a WoTSE can push resources directly to the query resolution process, skipping the index and storage layer. For instance, the MAX search engine [59] discovers relevant objects in its vicinity during query resolution process by broadcasting the query. The set of responded objects forms its query resource collection, which is dropped after the query is resolved. We consider these WoTSE having *"virtual resource collection."* Majority of existing WoTSE actually have *"real resource collection."*

Search layer carries out the query resolution process. The *Query Processor* module transforms raw user queries into the form processable by the system. The *Query Dependent (Q.D) Ranker* scores discovered query resources with respect to the user query and utilizes the recorded links between resources to find their corresponding result resources. A Meta-path with multiple types of query resources can be implemented by multiple Q.D Rankers. The *Ranking Aggregator* module is responsible for combining different Q.D and Q.I ranking results into a final score for each resource. Finally, the *Result Processor* extracts and aggregates the information from matching result resources and produces search results.

User Interface (UI) layer interfaces WoTSE with users. It provides *Query Interface* and *Result Interface* to receive queries and return search results, respectively. Their forms and implementations vary depending on the meta-path of a WoTSE. It also depends on type of users targeted by the WoTSE. Naturally, a system designed for software applications needs a different interface than a system designed for human users.

The modular architecture provides a reference framework assessing the diverse implementation of existing WoTSE. It assesses the support that each module receives from the existing works and how it is commonly implemented. Together with meta-path, the modular architecture forms our WoTSE model.

5 ANALYTICAL FRAMEWORK

The framework of our survey consists of three parts (Figure 5). The data for our survey is collected from bibliographic data set of DBLP[2] (retrieved on Sep 14, 2016) and Scopus.[3] Works included in our dataset are either directly related to search and discovery in WoT, or highly referenced by directly related works. The preliminary selection is done by a tool that we developed (Algorithm 1). The final selection is done manually to remove highly cited works that are not related to search engines, such as general surveys on WoT and IoT. The complete list of works is available.[4]

The second part of our analytical framework is building a model for WoTSE based on the collected works. Results of this part are Meta-path and the modular architecture that we discussed previously. From these models, we identify 24 dimensions, organized into seven groups, to analyze existing works (Table 3). Dimensions from meta-path (1.1 and 1.2) assess the general operation of a WoTSE prototype. The remaining dimensions describe the way a WoTSE prototype implements modules in our architecture. We include additional dimensions that reflect non-functional requirements, such as scalability and adaptability. We also include Experiment Type and Experiment Scale to assess the evaluation carried out by the prototype.

The third part of our framework is analyzing the growth of research around WoTSE and its current state, reflected by classical and latest works in the field. We use the number of publication and *in-field* citations (i.e., references among over 200 WoTSE works) each year to assess the growth of the field. For the detailed analysis, we map a subset of works against the dimensions that we built

[2] http://dblp.uni-trier.de/xml/.
[3] https://www.scopus.com/.
[4] http://cs.adelaide.edu.au/~nguyen/publications.html.

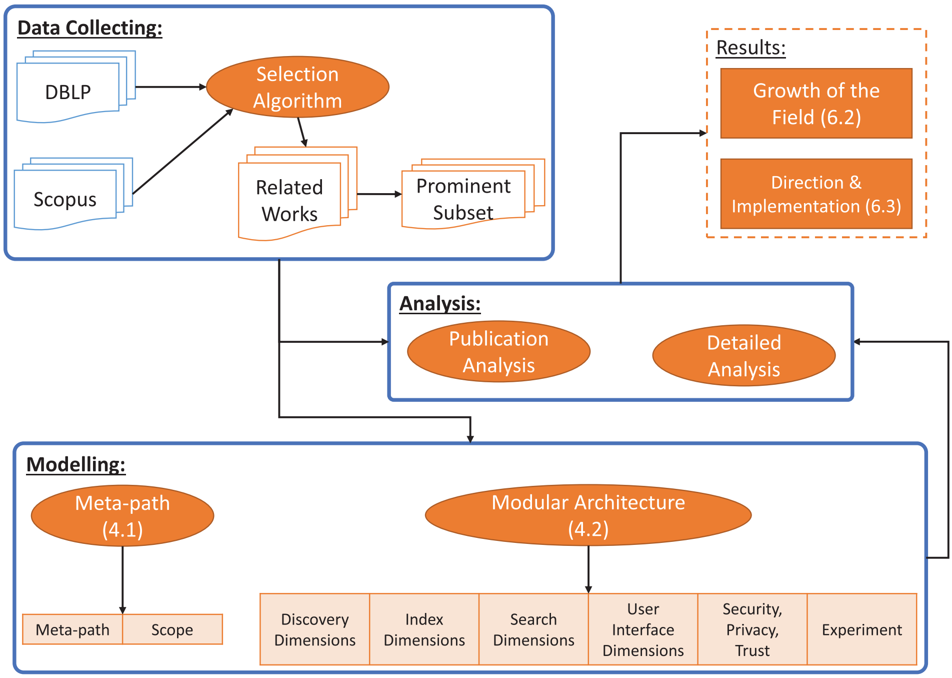


Fig. 5. Overview of the analytical framework. Oval objects represent components that we created. Dash arrows denote that the pointed object is derived from the pointing object. Solid arrows represent the link between inputs and outputs of our analysis.

in the second part of the framework. This subset of work is selected manually, with the attention on balancing the "classical" works with highest in-field citation count and latest works.

6 RESEARCH PROTOTYPES

6.1 Overview of Major Research Prototypes

6.1.1 MAX: A Human-centric Search Engine for the Physical World. MAX [59] is among the earliest works on building a search engine for the physical world. It is a standalone system that allows human users to provide a set of descriptive keywords to locate their tagged physical objects. To reflect its human-centric nature, MAX returns the location of the matching objects as landmarks instead of coordinates. MAX is organized into a three tier architecture, which is closely mapped into the organization of the physical locations in the real world. Located at the highest level are base stations, powered by the power line. These computers host the search application, host the security agent and act as gateways between the network of wireless tags and the backbone network (i.e., Internet). The middle tier consists of battery-powered RFID readers that are tied to large, rarely moved physical objects that represent landmarks. The bottom tier consists of passive RFID tags attached to small, mobile objects such as books and mugs. Queries are propagated through the network from the base stations, and the identity of the RFID readers detecting matching objects are returned as landmarks. To protect privacy of object owners, MAX allows them to specify their objects and physical spaces as private or off-limit to prevent unauthorized discovery and searches.

6.1.2 Discovery Mechanism of Global Sensor Network (GSN) System. GSN [1] is actually a platform for integrating wireless sensor networks over the Internet, not a search engine. However, it

ALGORITHM 1: Prototype Selection Algorithm

```
input  : DBLP: Bibliographic Dataset from DBLP
         SKW = {Discover, Search, Query}: Keywords related to search and discovery activity
         DKW = {WebofThings, WoT, InternetofThings, IoT}: Keywords to limit the domain of an article
output: Candidates: Set of candidate articles for manual selection
initialize Candidates list;
initialize References = title : count dictionary;
foreach article in DBLP do
    if article.name contains SKW and (article.name contains DKW or article.venue contains DKM) then
        append article to Candidates;
    end
end
foreach candidate in Candidates do
    extract the list of references refs from candidate;
    foreach ref in refs do
        increase References[ref] by 1;
    end
end
foreach reference in References do
    if count of reference is larger than 2 then
        append reference to Candidates;
    end
end
return Candidates
```

is highly referenced due to its sensor selection mechanism when processing sensor streams. GSN models each sensor node as a virtual sensor. Each virtual sensor is identified by a unique name and has a set of key-value pairs to represent its metadata. In other word, these virtual sensors can be consider the digital representative resource in our model. When processing sensor streams according to the declarative deployment description of users, GSN uses this information to retrieve sensors and perform the processing.

6.1.3 SenseWeb—An Infrastructure for Shared Sensing. SenseWeb [25] is a system for building applications and services based on the shared sensor data streams. Sensors in this system are connected to a centralized coordinator component through sensor gateway devices, which map their proprietary communication scheme into a standardized Web service API. An application requiring to use shared sensor data stream will interact with the tasking module of the coordinator a Web service API to express its sensing requests. The tasking module then searches on the static description of sensor streams to assess their capability, sharing willingness and other characteristics and return the relevant streams. This is a key distinction between the search service of SenseWeb and other search engines that also work with sensor streams such as DIS [57] and Dyser [39].

6.1.4 Distributed Image Search in Camera Sensor Network. DIS [57] is a system that performs general purpose image search on camera sensor network to recognize different types of objects. Its main difference comparing to the similar system is that it is for general purpose usage and can be used to recognize different types of objects instead of application specific, which will lock the whole camera sensor network into one task. It is one of the unique features of this search engine. To cope with the massive stream of captured images, which are enormous in scale and fast in

Table 3. Comparison Dimensions

Dimension	Description
1.1 Meta-path	Meta-path utilized by the WoT Search Engine under consideration.
1.2 Scope	The spatial range in which the WoT Search Engine can detect resources and interact with search users.
2.1 Discovery Scheme	Overall class of the discovery scheme utilized by the WoT Search Engine.
2.2 Mobility Support	Mechanisms utilized by the WoT Search Engine to detect and record the change in spatial locations of discovered entities.
2.3 Collector Type	Mechanisms to detect and collect resources.
3.1 Collection Type	The class of resource collections utilized by the WoT Search Engine.
3.2 Index Type	Mechanisms utilized by the WoT Search Engine to speed up the lookup process on resource collections.
3.4 Storage Scalability	Measures taken by the WoT Search Engine to ensure the scalability of its resource collections.
4.2 Query Model	The internal model of user queries utilized by the WoT Search Engine.
4.3 Result Model	The internal model of search results.
4.4 Q.D Ranking	Mechanisms utilized by the WoT Search Engine to assess the relevance of resources against a given query.
4.5 Adaptability	The ability of the WoT Search Engine to adapt its operations to different usage scenarios (e.g., different types of users).
4.6 Search Scalability	Mechanisms of the WoT Search Engine to ensure the scalability of its query processing.
5.1 User Type	Type of search users for which the WoT Search Engine is designed.
5.2 Interface Modal	The channel (i.e., "mode") on which the communication between search users and the search engine takes place.
5.3 Query Interface	The form of interface through which search users express their queries.
5.4 Result Interface	The form of interface through which search results are presented to users.
6.1 Security	The measures of the WoT Search Engines to protect itself against being breached by malicious parties.
6.2 Privacy	The measures of the WoT Search Engines to preserve the privacy of search users, sensor owners and sensed persons.
6.3 Trust	The measures taken by the WoT Search Engine to assess the trustworthiness of the discovered resources.
7.1 Experiment Type	The type of experiment carried out to evaluate the search engine prototype.
7.2 Experiment Scale	The scale of the carried out experiment, in terms of the number of data points or participants.

the generation rate, DIS employs a distributed search scheme, in which the discovery and search activities are carried out directly on each sensor nodes and results are combined into a single set of search result instead of having each sensor to transmit their readings to a centralized server for processing. Each image, either captured or supplied by search users are transformed into a set of 128 dimensional vectors using the Scale-Invariant Feature Transform (SIFT). These features are further clustered into Visual Words (i.e., "Visterms") to further reduce the space to represent these images. The matching between queried image and captured images are performed on visterms with TF-IDF score similarly to the matching between documents. DIS supports both ad hoc and continuous query. It can search on the newly captured image or the set of images stored in the camera sensor node.

6.1.5 Microsearch. Microsearch [53] is a scale-down information retrieval system that runs on sensor nodes with very limited computing and storage resources. It indexes small textual documents stored in the sensor node and returns the top-k documents that are most relevant to the query terms given by a search user. Documents are scored and ranked with the traditional Term Frequency (TF) and Inverse Document Frequency (IDF) metrics. While not being completely comparable to other works on this list, the unique approach of Microsearch provides an interesting alternative perspective on the problem. Therefore, it is included in our analysis.

6.1.6 Object Calling Home (OCH) System. OCH [18] system utilizes its participating mobile phones as a sensor network to locate missing physical objects. Each physical object is attached with a battery-powered Bluetooth transmitter, which is discovered by the phone's built-in Bluetooth discovery mechanism. To deal with the potential huge scale of the network, OCH utilizes a scoping mechanism that utilizes the association between things, humans and physical locations to reduce the number of sensors to pull during query resolution. The ideas proposed by OCH have been applied in commercial products (e.g., TrackR tag,[5] Tile[6]).

6.1.7 Dyser—A Real-Time Search Engine for Real-World Entities. The Dyser search engine [39] assesses queries of users against the real-world state of Web-enabled physical entities, which is reported by their attached sensors. The key challenge of Dyser is the dynamic nature of real-world states, which greatly surpasses the existing Web pages, rendering any indexes on these states outdated as soon as they are created. To solve this problem, Dyser assumes that sensor readings have a periodic nature and utilizes Sensor Rank algorithm [16] to predict sensor readings based on this assumption. The prediction result is used to order and minimize the sensor pull activity.

6.1.8 Ubiquitous Knowledge Base (uKB). uKB [47] is a distributed knowledge base whose assertion knowledge (i.e., knowledge about individual objects) are distributed over RFID tags attached to physical entities. The architecture of uKB consists of RFID readers that are inter-connected as a Mobile Ad hoc Network (MANET). In this survey, we focus on the discovery process in uKB, which discovers and gathers relevant pieces of assertion knowledge to a client to perform reasoning activities. The first step of discovery process is syntactical matching, in which syntactically relevant tags are detected based on their identity and the identity of the ontology that they use to describe themselves. The second step is semantic matching in which relevant tags are downloaded for further semantic-based assessment. Storing semantic description inside physical objects is an interesting and relevant idea for WoT. Therefore, discovery process of uKB is included in our analysis.

6.1.9 Snoogle—A Search Engine for Pervasive Environment. Reference [55] proposes that the pervasion of information stored in the networked sensors attached to physical entities will soon turn the world into a physical database. Snoogle is a search engine designed to look up information in such physical database. This search engine receives a set of keywords from a search user and returns a set of k objects having textually relevant description. To resolve query on a large number of sensors with limited computing and communication resources, Snoogle utilizes a distributed

[5] TrackR: https://www.thetrackr.com/.
[6] Tile: https://www.thetileapp.com/.

top-k query algorithm with pruning based on the characteristic of flash memory and Bloom filter to further reduce the transmission size. To preserve privacy of object owners, the textual content stored in private objects are encrypted with Elliptic Curve Cryptography (ECC).

6.1.10 DiscoWoT—Extensible Discovery Service for Smart Things. The heterogeneity of thing and service description is among the biggest challenges of WoT. DiscoWoT [31] is a semantic discovery service that aims to return the common representation form of any resource description given by a search user. DiscoWoT provides the common representation form and relies on strategies contributed by the community to translate resource description into this common form. The community effort lowers the entry-barrier for new WoT companies and products, and ensures that DiscoWoT is always up-to-date. While DiscoWoT appears to be very different from other WoT search engines, it is still mapped naturally into our model. If we consider each translation strategy as a function from a set of resource descriptions to the set of descriptions in the common representation form, then the union of these domains represents the set of all resource descriptions that DiscoWoT knows at a given point of time. In other words, this is the query resource collection of DiscoWoT. This collection is virtual, as it is not explicitly stored in the memory of the search engine.

6.1.11 IteMinder. IteMinder [27] is a search engine that allows users to locate their physical entities. Each entity participating in the system is attached with a passive RFID tag that stores its unique identity. Landmarks also attached with RFID tags for identification. IteMinder utilizes a physical robot, equipped with a laser rangefinder for navigation and an RFID reader for detection, to crawl the physical environment and record the location of physical objects into a database. Users are provided a Web interface mapping to look up in this database for their objects.

6.1.12 Searching the Web of Things. References [5, 10] develops a system for finding physical entities that have matching inputs and outputs to compose new applications and services. This system acts as a component in a larger WoT application framework instead of a standalone system. To describe physical entities, this system utilize five different ontologies to describe their finite state machines, their input and output structures, locations and owners. The entities are ranked by the similarity between their input structure and the output structure of the queried entity. This system also identifies the type of the interacting search user and adjusts its algorithm correspondingly.

6.1.13 Web of Things—Description, Discovery, and Integration. Reference [29] proposes to describe Web-enabled smart things with a common ontology, and register all smart things participating the network in a central Knowledge Base server for discovery purpose. The ontology describes entities based on their four basic capabilities: identity, processing, communication and storage. A user wishing to search for a smart thing sends his request to an Ambient Space Manager system, which in turn utilizes a Knowledge Base agent to query the Knowledge Base server for semantically matching entities.

6.1.14 Searching in a Web-Based Infrastructure for Smart Things. Reference [32] presents a distributed management infrastructure for environments populated with smart things, and a search engine prototype that allows users to perform look up for things in this infrastructure. These systems are organized as hierarchies according to logical identifiers of places that they cover to utilize the locality of smart things (i.e., things frequently interact with other things in their immediate environment). A user queries these systems by sending an HTTP GET request to one of their querying interfaces and providing the information for identifying a corresponding resource, along with spatial information to specify the query scope. Each query is modeled as a Web resource and assigned a unique URL. This URL is propagated through nodes in the infrastructure to build search results and returned to the search user.

6.1.15 Context-Aware Service Discovery for WoT. Reference [56] explores the use of contextual information collected from heterogeneous sources, including information about the physical world provided by networked sensors, to search for user-centric and situation-aware services to human and devices. This work models contextual information and relations among contexts with an ontology model that is extended to model uncertainty and temporal context. The contextual information is used to search on a service repository that contains both traditional Web services and real-world services provided by physical entities.

6.1.16 Context-Aware Sensor Search. CASSARAM [41] is a system that searches for connected-sensors using their contextual information, such as availability, accuracy, reliability, response time, and so on. It is motivated by the increasing number of sensors with overlapping capabilities deployed around the world and the lack of search functionality for these sensors. CASSARAM utilizes an extension of the Semantic Sensor Network Ontology (SSNO) [11] to describe the contextual information. A search user would query this ontology for sensors with a SPARQL query generated by the graphical user interface of CASSARAM. This interface also captures the references of the search user. The Euclidian distance between matched sensors and the user reference in a multi-dimensional space built from different types of sensor contextual information is used for ranking purpose. Top ranking sensors are returned as search results.

6.1.17 Content-based Sensor Search for Web of Things. Reference [54] defines content-based sensor search as the search for sensors that produce measurements within a certain range for a certain time period prior to the query. It is applied in WoT to find WoT-enabled physical entities that are in the queried real-time state. This work utilizes time-independent prediction models (TIPM) constructed for each individual sensor to rank them on based on their probability of having the queried state. This ranking activity reduces the communication overhead from validating the readings of matching sensors. TIPM is constructed from the assumption that a sensor reading that is frequently and continuously reported by a sensor in the past has a higher probability to be its current reading. To cope with the dynamic of sensor measurements, TIPM are continuously rebuilt and integrated into prior TIPM via a weighted sum. This method is evaluated by a combination of prototyping and simulation on a dataset of 162 sensors.

6.1.18 Ambient Ocean. Ambient Ocean [7] is a search engine that enables context-aware discovery of Web resources. Ambient ocean operates on Web resources attached with a data structure called Ocean Metadata that holds context metadata entities describing the current discoverability context of the Web resource. Each context metadata is backed by a context handler that provides mechanisms for comparison and indexing. Ambient Ocean relies on the community for the construction of context handlers and for adding context metadata to Web resources. A user interacts with Ambient Ocean server through a client application running on his mobile device. This client application utilizes readings from the local sensors as query terms to describe the current context of the user to the Ambient Ocean server. A list of URL pointing to Web resources having the relevant context is returned as the search result. Ambient Ocean can learn the association between context metadata entities to expand the given query.

6.1.19 Semantic Discovery and Invocation of Functionalities for the Web of Things. Reference [35] describes a search and discovery mechanism for functionalities of physical entities. It aims to discover and expose high-level functionalities of a physical entity that can be realized by a combination of its low-level physical capabilities and functionalities exposed by other entities in the immediate area. These functionalities and capabilities are described in a shared ontology. Each physical entity queries this ontology with a set of SPARQL queries encapsulated in Java functions. This work is a part of the avatar architecture from ASAWoO project, which aims to build an infrastructure for enhancing appliance integration into the Web and enable the collaboration between heterogeneous physical entities.

6.1.20 IoT-SVKSearch. IoT-SVK [12] is a hybrid search engine for WoT that is capable of resolving queries for WoT entities based on their textual description, their real-time sensor values, with respect to spatial and temporal constraints. IoT-SVK utilizes a uniform format to model the sampling data from WoT entities, which is distributed over multiple raw data storage for scalability. IoT-SVK utilizes three set of indexes. The full-text search on the description of WoT entities is handled by a B+ tree index. The spatial-temporal constraints of the queries are handled by an R-tree index, which is modified to support mobile entities. The value-based queries on sensor readings are handled a modified B+ tree index. Queries in IoT-SVK are processed as boolean expressions. The filtered search results are ranked according to an unspecified ranking mechanism. Evaluation of IoT-SVK is performed on a combination of real and simulated data from 352,000 sensors.

6.1.21 Gander. Gander [33] is a middle-ware and a search engine for pervasive computing environment, which is deployed directly on each node in the environment. It is designed for discovering and retrieving "datum," produced by these nodes, by propagating queries between Gander nodes in an ad hoc communication manner. Its prototype, however, is designed to work with virtual ad hoc networks deployed over the Internet. Gander allows users to fuse raw data into semantic, higher-level states in run-time with predefined rules in form of graphs. A query in Gander is modeled as a composition of three partial functions for filtering reachable, matching and constraint-satisfying nodes. Gander is evaluated with extensive case study and simulation.

6.1.22 Meta-Heuristic Approach for Context-Aware Sensor Search in the Web of Things. Reference [15] proposes a swarm intelligence method called AntClust to cluster sensors based on their meta-data and contexts for improving the scalability and efficiency of the search activity. The AntClust algorithm is inspired by the behavior of ants. It scatters all available sensors randomly on a sparse, two dimensional matrix and utilizes a set of agents to randomly pickup and drop sensors at different locations in the grid, biased by the relation between the selected sensor and its potential neighbors at the drop-off location. Performed experiments on a dataset of 100,000 sensors show that AntClust achieves notable gain in efficiency at the expense of accuracy, comparing to CASARRAM [41].

6.1.23 DNS as a WoT Search Engine. Reference [24] assumes that every WoT entity exposes their content as RESTful Web Services and proposes to use DNS to perform location-based search for these services. Each service is assigned with a URL in form of "sensorid.service.location.env." A user searching for sensors of a specific type at a specific location first queries the domain name "service.location.env" to retrieve a list of sensor URL, and then utilizes DNS to translate the collected URLs into IP addresses. Each service is assumed to return a self-description written in Web Application Description Language (WADL). Two experiments were performed on simulated data to assess the response time and storage requirement of this approach.

6.1.24 ForwarDS-IoT. ForwarDS-IoT [19] resolves queries for sensors and actuators whose semantic description stored on a federation of repositories. Each repository in ForwarDS-IoT has a domain-specific ontology, which is extended from SSN ontology. Users interact with ForwarDS-IoT via either its GUI or its RESTful API. Queries in ForwarDS-IoT specify conditions on metadata of physical objects, which are translated into a SPARQL queries and assessed against the stored semantic descriptions. ForwarDS-IoT supports both synchronous and asynchronous queries.

6.1.25 Extract-Cluster-Select (ECS). ECS [51] is a framework for producing relevant and diversified search results for the queries on physical entities in IoT. It utilizes the "Things Correlation Graph (TCG)," which represents a network of correlations between things, namely shared geographical locations and entity type. ECS consists of three steps. First, correlation between things are extracted to build TCG using $l1$-based graph construction method. Second, clusters of things are formed from TCG with spectral clustering techniques. Finally, things are selected based on the user's query and specified trade off between coherence and diversity of search results.

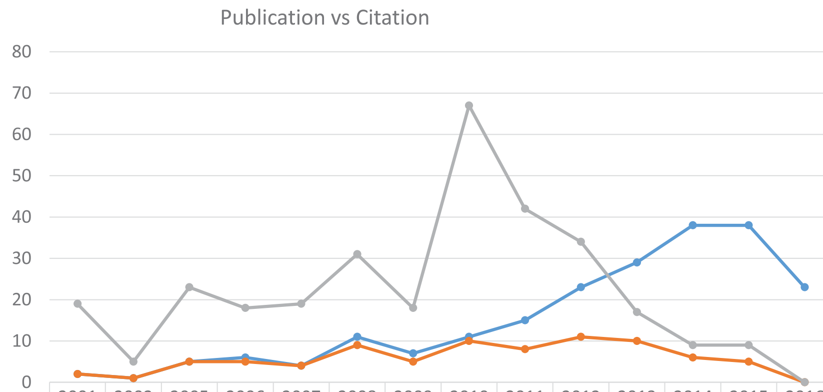
6.1.26 Query Processing for IoT. Reference [44] considers the process of searching for digital resources matching with real-world information reported by connected sensors. It utilizes statistical models to optimize the energy consumption of sensors and billing costs of cloud servers hosting these search applications analytically. The evaluation is carried out on a visual sensor network composed of multiple BeagleBone Linux embedded platforms and an application running on Amazon Web Service Elastic Compute Cloud (AWS EC2).

6.1.27 Context-aware Search System for IoT. Reference [8] presents a system that utilizes the contextual information extracted from IoT sensor data, namely user's identity, location, query time and current activity to search for physical entities and their related information. The activity recognition is performed by the combination of an online classifier based on Hidden Markov Model and pre-calculated probability distributions of different activities with respect to different time slots and locations. The detected user's activity, along with other contextual information, is used as the query to retrieve relevant entities and information from a pre-built context ontology.

6.1.28 ViSIoT. The Visual Search for Internet of Things system (VisIoT) [37] bridges sensor applications with public IoT cloud platforms by transforming the sensor information provided by public IoT clouds into the format required by the applications and exposing this information as virtual sensors via RESTful API. ViSIoT selects and ranks sensors on six "context properties" (i.e., battery, price, drift, frequency, energy consumption, and response time) using TOPSIS technique. The presented case study on Open Weather Map dataset shows that ViSIoT is capable of translating and deploying 100,000 sensors within 2min.

6.1.29 ThinkSeek. The ThinkSeek system [49] consists of a WoT crawler and a WoT search engine. The crawler extracts data about the physical world from public IoT cloud repositories on the Web, with the focus on live maps. Collected data is fed into a search engine and a Web-based visualization system. The ThinkSeek search engine supports both human users and smart devices. Humans utilize structured queries in the form of $(Location, \{Keywords\})$ to interact with the system, while smart devices utilize a CoAP RESTful API.

6.1.30 LHPM. LHPM [63, 64] is a prediction model for enabling searching on sensor content. LHPM consists of three parts. First, sensor readings are approximated with polynomials to lower the energy transfer cost. Second, sensor content is predicted with a multi-step, SVM-based method in which the new predictions are fed back as input. Finally, sensors are mapped into a two-dimensional vector space and ranked according to their cosine similarity with the given query. LHPM is evaluated with simulations with data from 54 temperature sensors from Intel Lab dataset and 78 water sensor from NOAA dataset. The experiment shows encouraging results on the approximation accuracy and energy consumption reduction.



	2001	2002	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Number of Publications	2	1	5	6	4	11	7	11	15	23	29	38	38	23
Number of Cited Works	2	1	5	5	4	9	5	10	8	11	10	6	5	0
Sum of citation	19	5	23	18	19	31	18	67	42	34	17	9	9	0

Fig. 6. Number of publications, in-field cited works, and in-field citations.

6.2 Publication and Citation Analysis

Figure 6 presents the number of published works, in-field cited works and in-field citations each year from over 200 selected works. The number of works related to WoTSE increases steadily each year since early 2000s, with a surge in 2009. Coincidentally, 2009 also marks the birth of the Internet of Things as the number of devices connected to the Internet becomes larger than World's population.[7] This surge of interest represents a gradual shift in focus of the community connecting things to the Web to finding and utilizing Web-enabled things. It also reflects the sharp perception of the community working with WoTSE.

However, the number of in-field cited works does not keep up with the number of publications. The gap between them expands at a steep rate from 2010. In fact, majority of in-field citations are held by a small set of works appearing around 2010. While citation count is not a perfect metric to evaluate the impact of research works, it can show that the existence of a research work is acknowledged by the the community. The lack of in-field citation is a possible indicator that majority of works around WoTSE are not detected by their peers.

6.3 WoTSE Form and Implementation Analysis

6.3.1 Comparison Result. The mapping of selected prototypes into dimensions defined in our analytical framework is presented in Figures 7, 8, and 9.

Operating scope of a WoTSE is presented in form of [*DiscoveryScope*] – [*SearchScope*]. **L**ocal scope denotes that the WoTSE can only find resources and provide search services to users in its vicinity, while **G**lobal scope denotes that it can operate across the Globe via the Web. A WoTSE can be tailored to work with **H**uman or **M**achine users. It's evaluation can be carried with **Prot**otypes on real devices or **Sim**ulation (e.g., network simulation with NS2)

The scheme of discovery process carried out by a WoTSE can be either **A**ctive or **P**assive. Active discovery means the search engine seek resources, while Passive discovery means resources are registered to the search engine. Depending on the scope and discovery scheme, a WoTSE uses different types of collector, including Web **Crawl**ers, resource **Reg**istration mechanisms and Local

[7] http://www.postscapes.com/internet-of-things-history/.

	Publication	Meta-path	Scope	Experiment Type	Experiment Scale
Max (Yap, et al. 05)	2005	R(Con) => E => R + D	L-G	Prot	10 participants
GSN (Aberer, et al. 07)	2007	D(ID) => D	L-G	-	-
SenseWeb (Kansal, et al. 07)	2007	D(Meta) => D	G-G	-	-
DIS (Yan, et al. 08)	2008	D(Con) => E => R	L-G	Prot	6 sensors
Microsearch (Tan, et al. 2008)	2008	S(Con) => S	L-L	Prot	1 sensor
OCH (Frank, et al. 2008)	2008	R(ID) => E => R + D	L-G	Prot,Sim	4 participants
Dyser (Ostermaier, et al. 10)	2010	D(Con) + R(Con) => E => R	G-G	Sim	385 sensors
uKB (Ruta, et al. 10)	2010	R(Con) => R	L-G	Sim	50 readers
Snoogle (Wang, et al. 10)	2010	R(Con) => E => R + D	L-G	Prot	8 sensors
WoT Discovery (Mayer, et al. 11)	2011	R(Con) => R	G-G	-	-
IteMinder (Komatsuzaki, et al. 11)	2011	R(Con) => E => R + D	L-G	Prot	1 robot
Christophe, et al. 11	2011	F(Meta) => F	G-G	Prot	-
Mathew, et al. 11	2011	R(Con) => R	L-G	-	-
WoT Search (Mayer, et al. 12)	2012	D(Con) + R(Con) => E => R	L-G	Sim	600 sensors
Wei, et al. 12	2012	F(Meta) => F	G-G	-	-

	Publication	Meta-path	Scope	Experiment Type	Experiment Scale
CASSARAM (Penera, et al. 13)	2013	D(Meta) => D	L-G	Sim	-
Content-based (Truong, et al. 13)	2013	D(Con) => E => R	G-G	Prot,Sim	163 sensors
Ambient Ocean (Carlson, et al. 14)	2014	R(Meta) => R	G-G	Sim	42,000 signal samples
(Mrissa, et al. 14)	2014	F(Meta) => F	L-L	-	-
IoT-SVK (Ding, et al. 14)	2014	D(Con) + R(Con) => E => R	G-G	Prot	10,000 Taxis
Gander (Michel, et al. 14)	2014	D(Con) + R(Con) => E => R	L-G	Prot	63 devices
DNS (Kamilaris, et al. 14)	2014	R(ID) => E => F	G-G	Sim	100 sensors
AntClust (Ebrahimi, et al. 15)	2015	D(Meta) => D	G-G	Sim	100,000 sensors
ForwarDS-IoT (Gomes, et al. 15)	2015	R(Meta) => R	G-G	Sim	-
ECS (Shemshadi, et al. 15)	2015	D(Con) + R(Con) => E => R	G-G	Sim	10,000 Taxis
Renna, et al. 16	2016	R(Con) => R	G-G	Prot	-
Chen, et al. 2016	2016	R(Meta) => R	G-G	Prot	8 participants
VisIoT (Nunes, et al. 16)	2016	D(Meta) => D	G-G	Sim	-
ThinkSeek (Shemshadi, et al. 16)	2016	D(Con) + R(Con) => E => R	G-G	Prot	-
LHPM (Zhang, et al. 16)	2016	D(Con) => E => R	L-G	Sim	132 sensors

Fig. 7. Meta-path types, scope, and experiment scale of selected prototypes.

	Discovery Scheme	Mobility Support	Collector Type	Collection Type	Index Type	Q.I Ranking	Storage Scalability	Search Scheme	Query Model	Result Model	Q.D Ranking	Adaptability	Search Scalability	User Type	Interface Modal	Query Interface	Result Interface	Security	Privacy	Trust
Max (Yap, et al. 05)	A	AHP	LD	V	-	-	V	AH	Txt	L	D(Rnk)	-	-	H	W	TBx	L	-	OAC,LAC	-
GSN (Aberer, et al. 07)	A	T	LD	R	-	-	-	AH	ID	Str	Ext	-	-	M	-	-	-	Crpt	OAC	-
SenseWeb (Kansal, et al. 07)	P	MP	R	R	U/S	-	-	AH	Cond	Str	D(Rnk)	-	Cch	M	API	API	-	-	OAC, Sum	R
DIS (Yan, et al. 08)	P	-	R	R	Txt	-	D	AH+C	Txt	L	D(Rnk)	-	D	H	App	-	L	-	-	-
Microsearch (Tan, et al. 2008)	P	-	R	R	Txt	-	-	AH	Txt	L	D(Rnk)	-	-	H	App	TBx	L	-	-	-
OCH (Frank, et al. 2008)	A	AHP	LD	V	-	-	V	AH	ID	S	Ext	-	Scp	H	App	Imp	S	-	OAC	-
Dyser (Ostermaier, et al. 10)	A	-	Crwl	R	PM	-	-	AH	Txt+ Cond	L	D(Rnk)+ Ext	-	Scp	H	W	TBx	L	-	-	-
uKB (Ruta, et al. 10)	A	AHP	LD	V	-	-	V	AH	Cond	L	Ext	-	-	M	-	-	-	-	-	-
Snoogle (Wang, et al. 10)	P	T+B	R	R	Txt	-	D	AH	Txt	L	D(Rnk)	-	D	H	App	TBx	L	Crpt	OAC	-
WoT Discovery (Mayer, et al. 11)	P	-	R	V	-	-	V	AH	ID	S	Ext	-	-	M	API	API	S	-	-	-
IteMinder (Komatsuzaki, et al. 11)	A	-	LD	R	-	-	-	AH	Txt	S	Ext	-	-	H	App	TBx	M	-	-	-
Christophe, et al. 11	P	-	R	R	U/S	-	-	AH	Cond	L	D(Rnk)	ReqT	-	H+M	App	Imp	-	-	-	-
Mathew, et al. 11	P	-	R	R	-	-	-	AH	Cond	L	Ext	-	-	H	App	TBx	L	-	-	-
WoT Search (Mayer, et al. 12)	A	AHP	LD	R	-	-	D	AH	Txt+ Cond	L	D(Rnk)	-	D	H+M	API	API	L	-	-	-
Wei, et al. 12	P	-	R	R	-	QoS	-	AH	Cond	L	Ext	-	-	M	API	API	L	-	-	-

Fig. 8. Implementation of selected prototypes. "-" denotes that the feature is not described, while "N/A" denotes that the feature is not implemented.

	Discovery Scheme	Mobility Support	Collector Type	Collection Type	Index Type	Q.I Ranking	Storage Scalability	Search Scheme	Query Model	Result Model	Q.D Ranking	Adaptability	Search Scalability	User Type	Interface Modal	Query Interface	Result Interface	Security	Privacy	Trust
CASSARAM (Penera, et al. 13)	A	-	LD	R	-	-	-	AH	Cond	L	D(Rnk)+ Ext	-	-	H+M	App+API	TBx+API	L	-	-	-
Content-based (Truong, et al. 13)	A	-	Crwl	R	PM	-	D	AH	Cond	L	P(Rnk)	-	D	H	W	F	L	-	-	-
Ambient Ocean (Carlson, et al. 14)	A	-	Crwl	R	U/S	R	-	AH	Txt	L	D(Rnk)	-	-	H	App	TBx+Sen	L	-	-	-
(Mrissa, et al. 14)	P	-	R	R	-	-	-	AH	Cond	L	Ext	-	-	M	-	-	-	-	F	-
IoT-SVK (Ding, et al. 14)	P	T	Reg	R	Txt,S,V	-	D	AH	Txt+ Cond	L	Ext	-	D	H	-	-	-	-	-	-
Gander (Michel, et al. 14)	A	AHP	LD	V	-	-	V	AH+C	Cond	L	Ext	-	D	M	API	API	L	-	-	-
DNS (Kamilaris, et al. 14)	P	-	R	R	-	-	D	AH	ID	L	Ext	-	D	M	API	API	L	-	-	-
AntClust (Ebrahimi, et al. 15)	P	-	R	R	Clst	-	-	AH	Cond	L	D(Rnk)	-	-	M	API	API	L	-	-	-
ForwarDS-IoT (Gomes, et al. 15)	P	-	R	R	-	-	D	AH+C	Cond	L	Ext	-	-	H+M	App+API	TBx+API	L	-	-	-
ECS (Shemshadi, et al. 15)	A	T	Crwl	R	Clst	-	-	AH	Txt+ Cond	L	D(Rnk)	-	-	H	W	TBx	M	-	-	-
Renna, et al. 16	-	-	-	R	-	-	-	AH	Txt	L	D(Rnk)	-	-	M	-	-	-	-	-	-
Chen, et al. 2016	-	-	-	R	Clst	-	-	AH	Txt+ Cond	L	D(Rnk)	-	-	H	App	TBx	L	-	-	-
VisIoT (Nunes, et al. 16)	P	-	R	R	-	-	-	AH	Cond	L	D(Rnk)	-	-	H	W	F	M	-	-	-
ThinkSeek (Shemshadi, et al. 16)	A	T	Crwl	R	Clst	-	-	AH	Txt+ Cond	L	D(Rnk)	-	-	H	W	TBx	M	-	-	-
LHPM (Zhang, et al. 16)	P	-	R	R	PM	-	D	AH	Cond	L	P(Rnk)	-	-	M	API	API	L	-	-	-

Fig. 9. (Cont) Implementation of selected prototypes.

Discovery (**LD**), which includes mechanisms to detect entities and resources in the immediate vicinity. The support for mobile objects by a WoTSE is organized into four groups. **T**imer denotes the continuous resampling of object's location after a predefined time period. **B**eacon denotes the mechanism in which the search engine continuously broadcasts beacons for receiving objects to register themselves. Ad hoc Pull (**AHP**) denotes that the location of objects are pulled every time a query is processed. Mobile Proxy (**MP**) denotes the use of spatially deployed proxies to query for resources at specific locations without having to keep track of their mobility [25].

The Collection Type includes **R**eal and **V**irtual collections. Index Type includes Text-based (**Txt**) indexes, which also include image-based indexes that treat images as a set of terms [57], **S**patial Indexes, numerical **V**alue Indexes, **Clust**ering mechanisms, Prediction Models (**PM**) (e.g., Sensor Rank [39]) and Unspecified Indexes (**U/S**) denoting indexing schemes that are mentioned but not described by the prototype. Q.I Ranking dimension includes the use of Quality-of-Service (**QoS**) and **R**atings from community. Storage Scalability support includes the use of **V**irtual resource collections to negate the need of actual storage and **D**istribution of resource storage over multiple instances of the search engine.

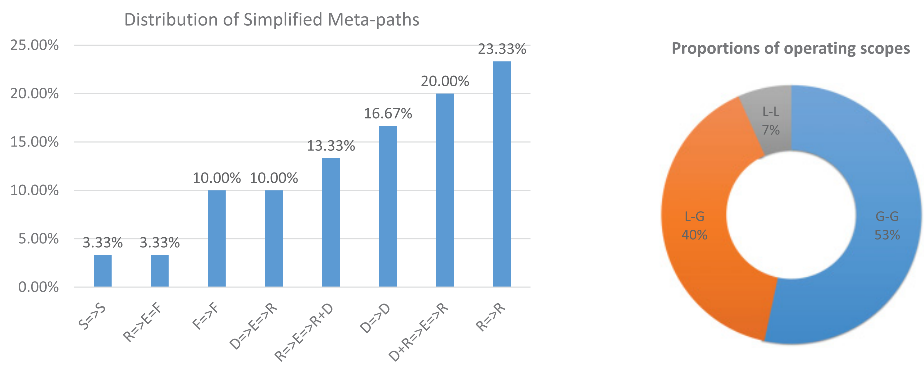
The Search Scheme dimension includes Ad hoc (**AH**) and **C**ontinuous search, denoting whether the given queries are matched one time against the current snap shot of the resource collection or continuously assessed against the updating collection. The Query Model dimension includes Text-based queries (**Txt**), Logical **Cond**itions and **ID**entity. The Search Result can be a **L**ist of matching data records, a **S**ingle record, or a **Str**eam of dynamic information (e.g., sensor readings). Q.D Ranking includes ranking based on the value of prediction models (**P(Rnk)**), distance-based ranking (**D(Rnk)**), which can be expressed by Euclidean distance, Jaccard index or Cosine similarity, and exact matching (**Ext**). We consider Text-based ranking (e.g., TF-IDF) a form of D(Rnk). Search Scalability mechanisms include the **D**istribution of query processing, caching (**Cch**) search results to reduce number of query sending to sensors (e.g., SenseWeb [25]) and scoping (**Scp**) to reduce the number sensors to assess. The adaptability dimension includes only one value—**ReqT**—which denotes the ability of a search engine to detect and adapt its algorithm to the type of user making the request.

The Interface Modal denotes the channel of communication between a search engine and search users, including **W**eb Interface, Web **API** and specialized **APP**lication. The form of interface on this channel to receive queries from users includes structured **F**orms, text boxes (**TBx**), **Sen**sors on client device, and **Impl**icit queries invoked by the interaction between users and client application (e.g., Reference [10]). Result interface includes the traditional **L**ist of records and the geographical **M**ap.

Security measures of WoTSE include encryption (**Crpt**). Privacy in WoTSE is protected by enforcing access control on objects (**OAC**) and spatial locations (**LAC**), **Sum**marizing sensor data and **F**iltering of search results to protect sensitive information of involved users. Finally, on Trust dimension, we have the value **R** denoting the use of ratings from the community.

6.3.2 *Form of WoT Search Engines.*

Figure 10(a) presents the distribution of meta-paths supported by the selected prototypes. Searching for objects based on their ID or metadata ($R \Rightarrow R$) is the most common form of WoTSE, followed closely by searching for objects using their real-time state (e.g., sensor readings, location) and searching for sensor streams ($D \Rightarrow D$). Familiarity is a possible explanation for the popularity of these meta-paths. For instance, $R \Rightarrow R$ is similar to Web search, while $R + D \Rightarrow Obj \Rightarrow R$ comes naturally with the idea of feeding real-world states into software applications. Surprisingly, searching for real-world functionality is not commonly supported even though it is crucial in the interaction with WoT-enabled smart environments.



(a) Meta-paths (b) Scopes

Fig. 10. Distribution of meta-paths and operating scopes.

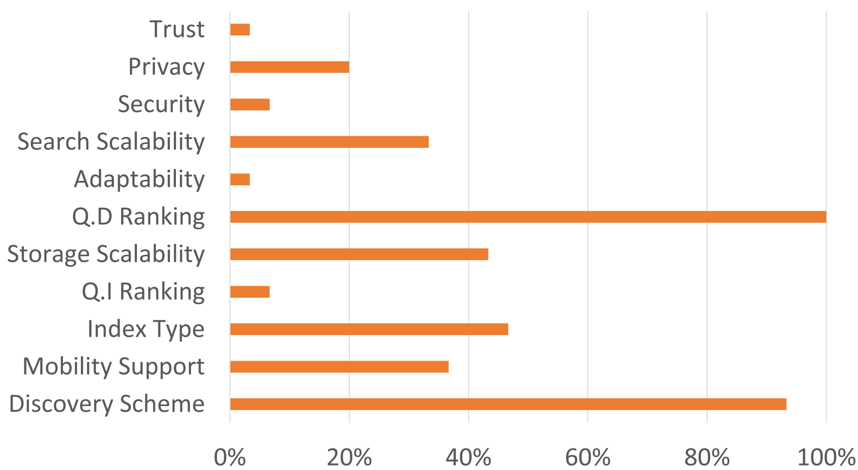


Fig. 11. Support of prototypes on key dimensions.

Figure 10(b) presents the distribution of operating scopes of selected prototypes. Local and global resource discovery are equally supported by the prototypes, which reflects the attention to both ends of the WoT scale. However, global scope dominates the search operation.

6.3.3 Implementation of WoT Search Engines. Figure 11 presents the support that key dimensions receive from the selected prototypes. Query Dependent (Q.D) ranking and discovery enjoy the strongest support, as they are the core of a WoTSE. Scalability of query processing and storage capability is supported by about half of prototypes. Most supporting prototypes scale up by utilizing virtual resource collections and distributing processing and storage across multiple computers. Mobility of physical objects is considered by less than 40% of the selected prototypes. The weak support for indexing is a surprising result, considering its crucial role in resolving queries. A possible explanation for this phenomenon is the simplicity of usage scenarios and involving resources in the selected prototypes.

Most prototypes do not support adaptability, which means that they cannot change their operations according to context, such as their current users. Query Independent (Q.I) ranking also lacks support, even though it plays a crucial role in the success of Web Search engines. It can be

Discovery Scheme	Count	Proportion
Active	13	48.15%
Passive	14	51.85%
Grand Total	**27**	**100.00%**

(a) Discovery Scheme

Collection Type	Count	Proportion
R	25	83.33%
V	5	16.67%
Grand Total	**30**	**100.00%**

(b) Collection Type

Search Scheme	Count	Proportion
AH	27	90.00%
AH+C	3	10.00%
Grand Total	**30**	**100.00%**

(c) Search Scheme

User Type	Count	Proportion
H	15	50.00%
H+M	4	13.33%
M	11	36.67%
Grand Total	**30**	**100.00%**

(d) User Type

Q.D Ranking	Count	Proportion
D(Rnk)	14	46.67%
D(Rnk)+Ext	2	6.67%
Ext	12	40.00%
P(Rnk)	2	6.67%
Grand Total	**30**	**100.00%**

(e) Q.D Ranking

Search/Result	L	S	Str	Grand Total
Cond	12		1	13
ID	1	2	1	4
Txt	6	1		7
Txt+Cond	6			6
Grand Total	**25**	**3**	**2**	**30**

(f) Query and Result Models

Fig. 12. Statistics of key dimensions.

contributed to the lack of natural order of WoT resources. Security, privacy, and trust are also not commonly addressed by prototypes.

Figure 12 presents the details of some interesting dimensions that receive high support. On discovery scheme, the active and passive schemes are equally utilized. This is a surprising result, because both Web Search and Sensor Search systems, which are frequently considered predecessors of WoTSE, rely on active discovery scheme. On collection-type dimension, real collections dominate, because it is the most straightforward and traditional solution in search engines. Search scheme is dominated by ad hoc search scheme, which is carried over from Web Search Engines. On targeted user type dimension, human users have a slight edge over machine users. Interestingly, some WoTSE are designed to support both types of users. On Q.D ranking dimension, distance-based ranking and exact matching are the two most common forms of ranking mechanisms among the selected prototypes. On the query model dimensions, logical conditions is the most common form of query, while list of resources is the most common result model.

7 INDUSTRIAL WORKS AND STANDARDS

7.1 Overview of Industrial Works and Standards

We define industrial works as publicly deployed and, optionally, commercialized products and services. We select two groups of industrial works for our evaluation based on references of research prototypes and IoT news sources. The first group is standalone WoT Search Engines. *Shodan*[8] proclaims to be "the world's first search engine for Internet-connected devices." It was designed and deployed by John Matherly in 2009. *Censys*[9] [14] search engine, deployed by the University of Michigan in 2015, and *Qadium*, which raised over 20 million dollars in fundings by 2016,[10] offer similar services. Essentially, these systems are tools for performing Internet-wide studies. However, they can be adapted to search for devices in WoT. Shodan and Censys can detect and access

[8] https://www.shodan.io/.
[9] https://censys.io/.
[10] http://www.forbes.com/sites/thomasbrewster/2016/06/05/qadium-iot-google-security-darpa-cia/#7289d6c42722.

(a) Industrial Works

Work	Metapath	Scope	User Type	Collection Type	Discovery Scheme	Mobility Support	Debut Year	Managing Organization	Cost
Censys	R	G	H	R	A	N/A	2015	University of Michigan	Free
Shodan	R	G	H	R	A	N/A	2009	John Matherly	From $19 USD/month
Thingful	R	G	H	R	A	N/A	2013	Umbrellium	Free
Qadium	R	G	-	-	A	-	2013	Qadium	-
AWS IoT	R	G	H,M	R	P	N/A	2015	Amazon	$5USD - $8USD per 1 million messages
Watson IoT	R	G	H,M	R	P	N/A	2015	IBM	$0.01USD per MB exchanged

(b) Standards

Work	Metapath	Scope	User Type	Collection Type	Discovery Scheme	Mobility Support	Debut Year	Managing Organization	Current State
EPCglobal Discovery Service	R=>S	G	H	R	P	N/A	2003	340 companies in 3 action groups	On-going
BRIDGE Discovery Service (WP2)	R=>S	G	H	R	P	N/A	2008	30 partners, coordinated by GS1	Ended. Deliverables include requirements and high-level design
Afilias ONS and Discovery Service [afilias:2008,rezafard:2008]	R=>S	G	H	R	P	N/A	2008	Afilias plc	On-going. Internet-draft submitted to IETF.

Fig. 13. Evaluation of WoT search engine industrial efforts (a) and standards (b).

a wide range of vulnerable network devices from Webcams, baby monitors, to ATM and medical devices.[11] *Thingful* search engine,[12] on the other hand, is designed specifically for WoT. Instead of pinging public IPv4 addresses, Thingful builds its dataset from sensor data sources on the Web. These resources are exposed for searching via a graphical map.

The second group is a search mechanism offered within commercial IoT Cloud Platform (e.g., Amazon Web Service IoT Platform (AWS IoT),[13] IBM Watson IoT platform (Watson IoT)[14]). These search mechanisms operate on objects and resources of the searcher, linked to the platform. The offered search capability is basic, such as filtering objects by their ID and metadata. It should be noted that while searched objects can be physically distributed across the globe, the scope of search capability offered to a user is still limited in his own "silo" of data.

We select standards for analysis based on the technical landscape of WoT Interest Group [23] and references of the research prototypes. We focus on standards that specify the whole discovery and search process, and select EPCglobal Discovery Service,[15] BRIDGE Discovery Service (WP2) [6], and Afilias Extensible Supply-chain Discovery Service (ESDS) [2, 45]. These standards revolve around "Discovery Service," which finds Information Systems in a network (e.g., Internet) that hold the information corresponding to a given object identifier. Therefore, from Meta-path perspective, these standards are very similar.

7.2 Evaluation

Figure 13 presents the evaluation result of industrial works and standards on a subset of our dimensions.

[11] https://blog.kaspersky.com/shodan-censys/11430/.
[12] https://thingful.net/.
[13] https://aws.amazon.com/iot/how-it-works/.
[14] http://www.ibm.com/internet-of-things/.
[15] http://www.gs1.org/epcrfid/epc-rfid-dci/1.

Comparing to the result of academic prototypes, industrial works and standards are considerably less "adventurous." They converge to searching for objects based on their static information such as ID and metadata, which is arguably the most natural step from the existing Web Search Engines. And, in the current state of WoT, this form might be all it takes to reap benefits from the emerging library of the real world.

8 DISCUSSION AND OPEN ISSUES

The goal of WoTSE is building an "ideal" search engine that can find *"anything,"* at *"anywhere"* and *"anytime."* "Anything" means it can work with any meta-path, involving any combination of WoT resources. "Anywhere" means it can utilize the spatial information to objects located at any specific location, in any specific area. "Anytime" means it can utilize the whole range WoT data, from the archived sensor readings to current sensing data to the data that will be produced in the future to match queries with resources. Resources returned by an ideal WoTSE not only have relevant content, but they also have that content at the relevant time, at the relevant place. An ideal WoTSE is the gateway to the Web of Things.

Moving toward this vision from the current state of the art requires us to address a wide range of issues. In this section, we discuss prominent ones.

8.1 Crawling WoT

Constructing resource collections automatically via crawling is desirable in WoTSE. However, this task is very challenging. The first issue is *detecting WoT data sources*. These sources can be organized into four groups [50]: cloud-based IoT platforms (e.g., Amazon Web Service IoT Platform, IBM Watson IoT platform), live-maps such as real-time transportation information services (e.g., FlightRadar24[16]), urban crowdsensing services (e.g., Waze[17]) and public environmental sensing services. To detect these sources automatically, their features must be formally defined and mapped into machine-detectable traits of Websites. These criteria are not straightforward, even for human operators. For instance, should a live-map of lightnings[18] around the world be considered a WoT data source?

The second issue is *extracting resources automatically*. Resources are commonly transferred by XML HTTP Request (XHR) responses in form of XML or JSON documents. Currently, the URL pattern of these XHR must be detected manually [49]. The third issue is *automatic integration of resources*. High degree of overlapping in coverage is observed in the WoT data sources (e.g., flight data [49]). However, the data that they provide is not completely identical. Aggregating reports from different data sources can reveal a complete picture of collected resources. However, this automation is challenging due to the diversity of resource data fields and formats.

8.2 Supporting Location-Based Search

Spatial information is crucial in searching WoT [32, 62]. The first challenge of providing location-based search is *identifying locations*. Latitude, longitude, and the height comparing to sea level together forms a potential location identifier. It is feasible in outdoor scenarios with sparse sensors. However, its granularity is challenged in indoor environments with dense distribution of objects. A potential solution is utilizing different coordinate systems with different granularity for different scenarios. However, this approach raises additional questions. For instance, how to

[16] http://flightradar24.com.
[17] https://www.waze.com.
[18] https://www.lightningmaps.org/?lang=en.

recognize the utilized coordinate system? How to integrate different coordinate systems into a single index structure?

The second problem is *associating coordinates with landmarks*. The role of landmarks to coordinates is similar to the role of domain names to IP addresses. For human users, landmark is preferable comparing to coordinates in both query and search results. For instance, a search result showing that the missing key chain is "under the desk in the dining room" is more intuitive than numerical coordinates. However, the query processing would be straightforward and unambiguous with numerical coordinates. Therefore, WoT Search Engines must be able to formally and semantically describe landmarks and translate between coordinates and landmarks.

8.3 Supporting the Dynamic Nature of WoT

Mobility of physical objects and changing sensor readings reflect the dynamic nature of WoT. Detecting and storing changes are key problems. In the context of WoTSE, we consider the problem of storing changes. The critical issue is indexing the changing data.

The first issue of storing changes is *indexing*. As indexes on sensor measurements are outdated as soon as they are created [39], a balance must be achieved between the freshness of stored data and the communication overhead of pulling the latest sensor measurements. For instance, a naive solution is pulling readings from all detected sensors for every query received. This approach does guarantee the freshness of data; however, the massive communication overhead negates any scaling-up possibility. An emerging solution is indexing prediction models of sensors instead. These prediction models can be built on the assumption of the periodic nature of sensor measurements [39] or from the density and scalability of each sensor reading within a time frame prior to the query [54]. Based on the result of the indexed prediction model, a search engine contacts can limit the number of sensors to validate before building search results.

The second issue is *storing and purging the collected data*. A WoT Search Engine must find a balance between the number of old readings stored for resolving historical queries and building prediction models, and the scale resources that it manages, because each set of past measurements duplicates the whole resource collection. As a result, mechanisms for ensuring the scalability of the data storage such as distribution deployment and purging strategies must be investigated.

The final issue is supporting subscription-based queries and continuous query processing. These abilities allow search users to register their interest for a specific real-world state and receive relevant search results in the future when they are detected. Subscription and continuous query processing are discussed but not implemented in the existing prototypes.

8.4 Supporting the Diversity

An ideal WoTSE must be able to work with many different types of resources and their combinations to resolve all given queries. Basic solutions are either building a search engine that is highly adaptable, or building a large number of specialized search engines for different resource combinations. The first solution might lead to "jack-of-all-trades" systems that are usable in many scenarios, but not particularly competent in any of them. In the second solution, the diversity of the search engines itself might become the problem.

A potential solution for this challenge is enabling modular construction of WoTSE, in which search engines are composed from a set of standardized modules according to the meta-path needed by a given query (Figure 14). The analysis of existing works reveals that meta-paths of different WoTSE overlap to a certain degree. By turning the whole discovery and search process of WoTSE into standardized modules, we can reuse them in other WoTSE that has (partial) overlapping meta-paths. This method facilitates specialization. Involving parties can focus on only components that align with their expertise instead of having to build the whole system. These

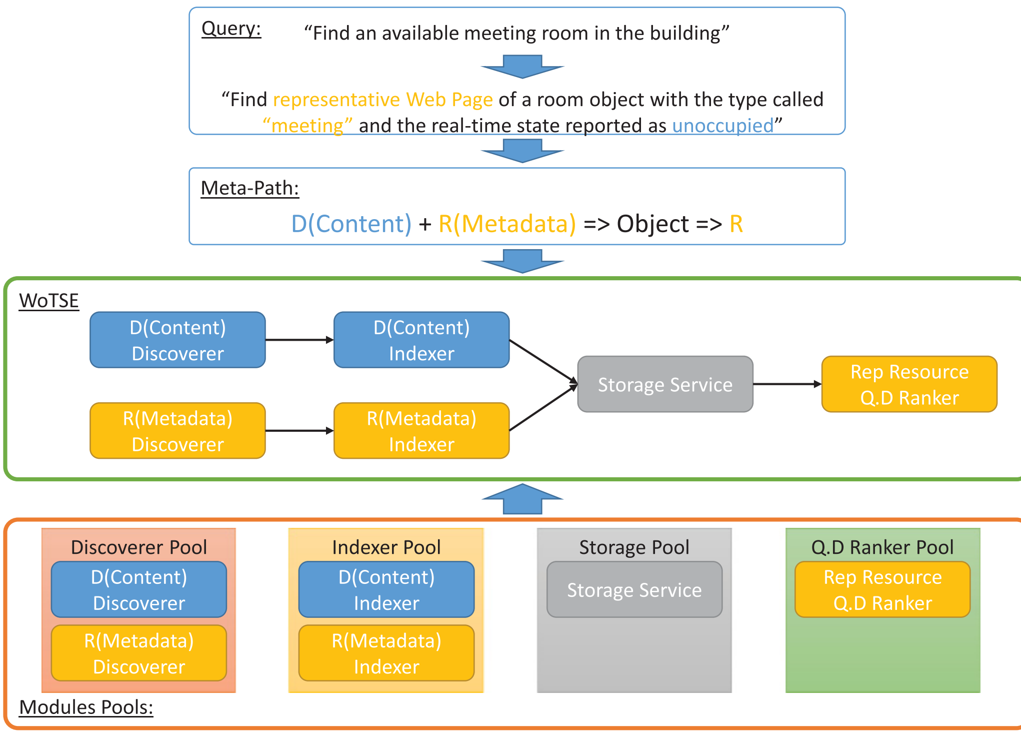


Fig. 14. Modular construction of Web of Things search engines.

components are then easily shared with the community to leverage the improvement and development of other components to ensure that the global WoTSE is always optimal and ready to cope with any combination of resource types in WoT.

Two major issues in enabling modular construction are standards and security. For modules that are independently developed to work together, we must provide standards for interfaces between modules, their operations, characteristics, and their arrangement as a system. We also need to ensure that modules actually do what they promise and ensure that they are not biased in their operation. These are challenging endeavors.

8.5 Supporting Scalability

The scale of WoT extends to both extremes of the spectrum: it is expected to be 50 or 100 times larger than the existing Web, yet majority of its interaction would be in small sets of co-located objects and resources. Therefore, WoTSE must fit the search activity in local scale naturally, and at the same time, they must also be able to scale up to reach billions devices on the world. Scaling up centralized search engines is not a preferable solution, because these systems are too far from physical world, making them insensitive to changes. Moreover, these systems must be able to identify a massive number of private locations (i.e., rooms inside smart homes) and associate private objects with these locations, which is challenging technically, socially, and politically.

An alternative solution is distributing WoTSE closer to the edge of WoT and linking them into a federation to provide global coverage. The WoTSE can be hosted on a computer, or even a smart phone to provide services to all authorized applications in its immediate vicinity (Figure 15).

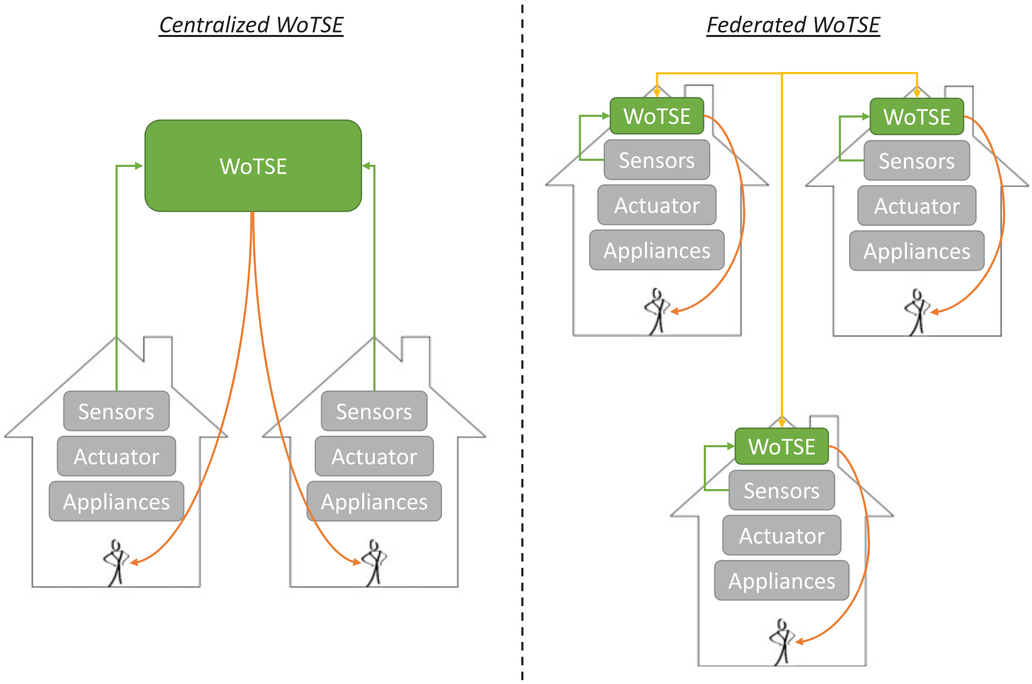


Fig. 15. Comparison between federated and centralized WoTSE.

Distributing WoTSE to the edge of WoT makes them naturally fit for local search activity, while linking them together provides the coverage to address the upper ends of WoT scale. Two major issues in enabling federated WoTSE are building effective and efficient methods to manage this massive federation and evaluating the trust of each member WoTSE in the federation.

8.6 Security, Privacy, and Trust

As the gateway to WoT that is capable of finding "anything," at "anywhere" and "anytime," WoT Search Engines represent unprecedented security and privacy risks. For instance, WoTSE can be used to track a person over a broad area and time period for surveillance or other malicious purposes. It can also be used to find and attack unprotected Web-enabled vehicles and medical devices, as demonstrated by Shodan and Censys. Such a system can also be used to spy on the stockpile and the transportation fleet of a company, resulting in massive economical damage. On the national and international scale, WoTSE can be a dangerous tool for espionage and sabotage. Therefore, a key issue of WoT Search Engine is *protecting the privacy of searchers, information owners and sensed people*. This is a challenging task, because a person cannot opt out of being sensed by sensors. Moreover, we lack the mechanisms to define ubiquitously accessible privacy policies and mechanisms to enforce them.

The second issue is *validating the discovered WoT content*. As real-world information in WoT is provided by exposed electronic tags and sensors that can be breached and forged, a malicious party can inject false information into WoT, which would be distributed by WoT Search Engines. For instance, consider a restaurant recommendation system that infers the crowdedness of restaurants with public sensors retrieved via WoT Search Engines. Rivals of a restaurant can sabotage it by planting forged sensors that report extreme noise and movement in its vicinity, causing the restaurant to be inferred as full and removed from the recommendation list. This type of attack can drive the restaurant out of business and damage the trust of users in both the recommendation system and the WoT Search Engine. A potential solution for this issue is validating the information received from sensors against past patterns and readings of their neighboring sensors. Another potential solution is building the audit-ability into WoTSE. Ensuring that one would be held accountable for his malicious activities is a powerful preventive mechanism.

9 CONCLUSION

The World is becoming a library of resources for software applications, thanks to the emerging Web of Things. This progress simplifies the development and adoption of cyber-physical applications, enabling WoT to realize its expected social and economical impacts. Web of Things Search Engines ensure the optimal utilization of this emerging library. Diversity of solution and the scale of WoT are main challenges facing WoTSE.

Our survey on over 200 academic and industrial works related to WoTSE confirms the continuous expansion of the field. It also reveals skewness in the attention that these works receives from their contemporaries. Searching for real-world objects, based on their real-world state is currently the most popular form of WoTSE. Bridging the gap from here to an ideal WoTSE that can find "anything," at "anywhere" and "anytime" requires us to address many issues, including diversity and scalability. For diversity issue, we propose to adopt a modular construction for WoTSE to facilitate better reuse of existing efforts to handle new forms of WoTSE. For scalability issue, we propose to distribute WoTSE to the edge of WoT and linking them into a federation. This arrangement provides WoTSE the natural ability to search locally, while also being able to scale up and serve 50 billion devices in 2020.

REFERENCES

[1] Karl Aberer, Manfred Hauswirth, and Ali Salehi. 2007. Infrastructure for data processing in large-scale interconnected sensor networks. In *Proceedings of the International Conference on Mobile Data Management, 2007*. IEEE, 198–205.

[2] Afilias. 2008. Afilias—Finding your Way in the Internet of Things. Retrieved from http://www.afilias.info/webfm_send/37.

[3] L. Atzori, A. Iera, and G. Morabito. 2010. The internet of things: A survey. *Comput. Netw.* 54, 15 (2010), 2787–2805. DOI: http://dx.doi.org/DOI 10.1016/j.comnet.2010.05.010

[4] Alessandro Bassi and Geir Horn. 2008. Internet of things in 2020: A roadmap for the future. In *Proceedings of the European Commission: Information Society and Media.*

[5] Mathieu Boussard, Benoit Christophe, Olivier Le Berre, and Vincent Toubiana. 2011. Providing user support in web-of-things enabled smart spaces. In *Proceedings of the 2nd International Workshop on Web of Things.* ACM, 11.

[6] BRIDGE. 2007. BRIDGE WP02—High Level Design for Discovery Services (17 August 2007).

[7] Darren Carlson and Andreas Schrader. 2014. Ambient ocean: A web search engine for context-aware smart resource discovery. In *Proceedings of the IEEE International Conference on Internet of Things (iThings'14)*. IEEE, 177–184.

[8] Yuanyi Chen, Jingyu Zhou, and Minyi Guo. 2016. A context-aware search system for internet of things based on hierarchical context model. *Telecommun. Syst.* 62, 1 (2016), 77–91.

[9] Benoit Christophe, Mathieu Boussard, Monique Lu, Alain Pastor, and Vincent Toubiana. 2011. The web of things vision: Things as a service and interaction patterns. *Bell Labs Tech. J.* 16, 1 (2011), 55–61.

[10] Benoit Christophe, Vincent Verdot, and Vincent Toubiana. 2011. Searching the "web of things." In *Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC'11).* IEEE, 308–315.

[11] Michael Compton, Payam Barnaghi, Luis Bermudez, Ral Garca-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semant.: Sci. Serv. Agents. WWWeb* 17 (2012), 25–32. DOI: http://dx.doi.org/10.1016/j.websem.2012.05.003

[12] Zhiming Ding, Zhikui Chen, and Qi Yang. 2014. IoTSVKSearch: A real time multimodal search engine mechanism for the internet of things. *Int. J. Commun. Syst.* 27, 6 (2014), 871–897.

[13] Simon Duquennoy, Gilles Grimaud, and Jean-Jacques Vandewalle. 2009. The web of things: Interconnecting devices with high usability and performance. In *Proceedings of the International Conference on Embedded Software and Systems (ICESS'09)*. IEEE, 323–330.

[14] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 542–553.

[15] Mohammad Ebrahimi, Elaheh Shafieibavani, Raymond K. Wong, and Chi-Hung Chi. 2015. A new meta-heuristic approach for efficient search in the internet of things. In *Proceedings of the 2015 IEEE International Conference on Services Computing (SCC'15)*. IEEE, 264–270.

[16] B. Maryam Elahi, Kay Romer, Benedikt Ostermaier, Michael Fahrmair, and Wolfgang Kellerer. 2009. Sensor ranking: A primitive for efficient content-based sensor search. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 217–228.

[17] Sergei Evdokimov, Benjamin Fabian, Steffen Kunz, and Nina Schoenemann. 2010. Comparison of discovery service architectures for the internet of things. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'10)*. IEEE, 237–244.

[18] Christian Frank, Philipp Bolliger, Friedemann Mattern, and Wolfgang Kellerer. 2008. The sensor internet at work: Locating everyday items using mobile phones. *Pervas. Mobile Comput.* 4, 3 (2008), 421–447.

[19] Porfirio Gomes, Everton Cavalcante, Taniro Rodrigues, Thais Batista, Flavia C. Delicato, and Paulo F. Pires. 2015. A federated discovery service for the internet of things. In *Proceedings of the the 2nd Workshop on Middleware for Context-Aware Applications in the IoT*. ACM, 25–30. DOI: http://dx.doi.org/10.1145/2836127.2836129

[20] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of things (IoT): A vision, architectural elements, and future directions. *Fut. Gen. Comput. Syst.* 29, 7 (2013), 1645–1660.

[21] Dominique Guinard, Vlad Trifa, Friedemann Mattern, and Erik Wilde. 2011. *From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices*. Springer, 97–129.

[22] W3C WoT IG. 2015. Web Thing Model (2015).

[23] W3C WoT IG. 2016. Web of Things—Technology Landscape (2016).

[24] Andreas Kamilaris, Koula Papakonstantinou, and Andreas Pitsillides. 2014. Exploring the use of DNS as a search engine for the web of things. In *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 100–105.

[25] Aman Kansal, Suman Nath, Jie Liu, and Feng Zhao. 2007. Senseweb: An infrastructure for shared sensing. *IEEE Multimed.* 4 (2007), 8–13.

[26] Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, and Howard Morris. 2002. People, places, things: Web presence for the real world. *Mobile Netw. Appl.* 7, 5 (2002), 365–376.

[27] Mizuho Komatsuzaki, Koji Tsukada, Itiro Siio, Pertti Verronen, Mika Luimula, and Sakari Piesk. 2011. IteMinder: Finding items in a room using passive RFID tags and an autonomous robot. In *Proceedings of the 13th International Conference on Ubiquitous Computing*. ACM, 599–600.

[28] N. Koshizuka and K. Sakamura. 2010. Ubiquitous ID: Standards for ubiquitous computing and the internet of things. *Pervas. Comput. IEEE* 9, 4 (2010), 98–101. DOI: http://dx.doi.org/10.1109/MPRV.2010.87

[29] Sujith Samuel Mathew, Yacine Atif, Quan Z. Sheng, and Zakaria Maamar. 2011. Web of things: Description, discovery and integration. In *Proceedings of the International Conference on Internet of Things (iThings/CPSCom'11) and 4th International Conference on Cyber, Physical and Social Computing*. IEEE, 9–15.

[30] Friedemann Mattern and Christian Floerkemeier. 2010. *From the Internet of Computers to the Internet of Things.* Springer, 242–259.

[31] Simon Mayer and Dominique Guinard. 2011. An extensible discovery service for smart things. In *Proceedings of the 2nd International Workshop on Web of Things*. ACM, 7.

[32] Simon Mayer, Dominique Guinard, and Vlad Trifa. 2012. Searching in a web-based infrastructure for smart things. In *Proceedings of the 3rd International Conference on the Internet of Things (IOT'12)*. IEEE, 119–126.

[33] Jonas Michel, Christine Julien, and Jamie Payton. 2014. Gander: Mobile, pervasive search of the here and now in the here and now. *IEEE Int. Things J.* 1, 5 (2014), 483–496.

[34] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. 2012. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* 10, 7 (2012), 1497–1516.

[35] Michael Mrissa, Lionel Mdini, and Jean-Paul Jamont. 2014. Semantic discovery and invocation of functionalities for the web of things. In *Proceedings of the IEEE 23rd International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'14)*. IEEE, 281–286.

[36] Anne H. H. Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Quan Z. Sheng. 2017. IoT middleware: A survey on issues and enabling technologies. *IEEE Internet of Things Journal (IoT-J)*, 4, 1 (2017), 1–20.

[37] Luiz Nunes, Julio Estrella, Luis Nakamura, Rafael de Libardi, Carlos Ferreira, Liuri Jorge, Charith Perera, and Stephan Reiff-Marganiec. 2016. A distributed sensor data search platform for internet of things environments. *arXiv:1606.07932* (2016).

[38] A. M. Ortiz, D. Hussein, Park Soochang, S. N. Han, and N. Crespi. 2014. The cluster between internet of things and social networks: Review and research challenges. *Int. Things J. IEEE* 1, 3 (2014), 206–215. DOI: http://dx.doi.org/10.1109/JIOT.2014.2318835

[39] Benedikt Ostermaier, K. Romer, Friedemann Mattern, Michael Fahrmair, and Wolfgang Kellerer. 2010. A real-time search engine for the web of things. In *Proceedings of the 1st International Conference on the Internet of Things (IOT'10)*. IEEE, 1–8.

[40] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the Web. Stanford InfoLab.

[41] Charith Perera, Arkady Zaslavsky, Peter Christen, Michael Compton, and Dimitrios Georgakopoulos. 2013. Context-aware sensor search, selection and ranking model for internet of things middleware. In *Proceedings of the IEEE 14th International Conference on Mobile Data Management (MDM'13)*, Vol. 1. IEEE, 314–322.

[42] Wu Qihui, Ding Guoru, Xu Yuhua, Feng Shuo, Du Zhiyong, Wang Jinlong, and Long Keping. 2014. Cognitive internet of things: A new paradigm beyond connection. *Int. Things J. IEEE* 1, 2 (2014), 129–143. DOI: http://dx.doi.org/10.1109/JIOT.2014.2311513

[43] Yongrui Qin, Quan Z. Sheng, Nickolas J. G. Falkner, Schahram Dustdar, Hua Wang, and Athanasios V. Vasilakos. 2016. When things matter: A survey on data-centric internet of things. *J. Netw. Comput. Appl.* 64 (2016), 137–153.

[44] Francesco Renna, Joseph Doyle, Vasileios Giotsas, and Yiannis Andreopoulos. 2016. Query processing for the internet-of-things: Coupling of device energy consumption and cloud infrastructure billing. In *Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI'16)*. IEEE, 83–94. DOI: http://dx.doi.org/10.1109/IoTDI.2015.37

[45] Rezafard. 2008. Extensible Supply-chain Discovery Service Problem Statement. IETF Internet-Draft, Nov 17, 2008. Retrieved from http://tools.ietf.org/html/draft-rezafard-esds-problem-statement-03.

[46] Kay Romer, Benedikt Ostermaier, Friedemann Mattern, Michael Fahrmair, and Wolfgang Kellerer. 2010. Real-time search for real-world entities: A survey. *Proc. IEEE* 98, 11 (2010), 1887–1902.

[47] Michele Ruta, Eugenio Di Sciascio, Giacomo Piscitelli, and Floriano Scioscia. 2010. A ubiquitous knowledge-based system to enable RFID object discovery in smart environments. *Pac. Asia J. Assoc. Info. Syst.* 2, 3 (2010).

[48] Chen Shanzhi, Xu Hui, Liu Dake, Hu Bo, and Wang Hucheng. 2014. A vision of IoT: Applications, challenges, and opportunities with china perspective. *Int. Things J. IEEE* 1, 4 (2014), 349–359. DOI: http://dx.doi.org/10.1109/JIOT.2014.2337336

[49] Ali Shemshadi, Quan Z. Sheng, and Yongrui Qin. 2016. ThingSeek: A Crawler and Search Engine for the Internet of Things (2016). DOI: http://dx.doi.org/10.1145/2911451.2911471

[50] Ali Shemshadi, Quan Z. Sheng, Yongrui Qin, Aixin Sun, Wei Emma Zhang, and Lina Yao. 2017. Searching for the internet of things: Where it is and what it looks like. Personal and Ubiquitous Computing. 1–16.

[51] Ali Shemshadi, Lina Yao, Yongrui Qin, Quan Z. Sheng, and Yihong Zhang. 2015. ECS: A framework for diversified and relevant search in the internet of things. In *Proceedings of the International Conference on Web Information Systems Engineering*. Springer, 448–462.

[52] Vlad Stirbu. 2008. Towards a restful plug and play experience in the web of things. In *Proceedings of the IEEE International Conference on Semantic Computing*. IEEE, 512–517.

[53] Chiu C. Tan, Bo Sheng, Haodong Wang, and Qun Li. 2008. *Microsearch: When Search Engines Meet Small Devices.* Springer, 93–110.

[54] Cuong Truong and Kay Rmer. 2013. Content-based sensor search for the web of things. In *Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM'13)*. IEEE, 2654–2660.

[55] Haodong Wang, Chiu C. Tan, and Qun Li. 2010. Snoogle: A search engine for pervasive environments. *IEEE Trans. Paral. Distrib. Syst.* 21, 8 (2010), 1188–1202.

[56] Qiang Wei and Zhi Jin. 2012. Service discovery for internet of things: A context-awareness perspective. In *Proceedings of the 4th Asia-Pacific Symposium on Internetware*. ACM, 25.

[57] Tingxin Yan, Deepak Ganesan, and R. Manmatha. 2008. Distributed image search in camera sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. ACM, 155–168.

[58] Lina Yao, Quan Z. Sheng, Byron J. Gao, Anne H. H. Ngu, and Xue Li. 2013. A model for discovering correlations of ubiquitous things. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM'13)*. IEEE, 1253–1258.

[59] Kok-Kiong Yap, Vikram Srinivasan, and Mehul Motani. 2005. MAX: Human-centric search of the physical world. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*. ACM, 166–179.

[60] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. 2014. Internet of things for smart cities. *Int. Things J. IEEE* 1, 1 (2014), 22–32. DOI: http://dx.doi.org/10.1109/JIOT.2014.2306328

[61] Arkady Zaslavsky, Charith Perera, and Dimitrios Georgakopoulos. 2013. Sensing as a service and big data. *arXiv:1301.0159* (2013).

[62] Daqiang Zhang, Laurence T. Yang, and Hongyu Huang. 2011. Searching in internet of things: Vision and challenges. In *Proceedings of the 2011 IEEE 9th International Symposium on Parallel and Distributed Processing with Applications*. IEEE, 201–206.

[63] Puning Zhang, Yuan-an Liu, Fan Wu, Suyan Liu, and Bihua Tang. 2016. Low-overhead and high-precision prediction model for content-based sensor search in the internet of things. *IEEE Commun. Lett.* 20, 4 (2016), 720–723. DOI: http://dx.doi.org/10.1109/LCOMM.2016.2521735

[64] Puning Zhang, Yuan-an Liu, Fan Wu, and Bihua Tang. 2015. Matching state estimation scheme for content-based sensor search in the Web of things. *Int. J. Distrib. Sensor Netw.* 2015 (2015), 221.

[65] Yuchao Zhou, Suparna De, Wei Wang, and Klaus Moessner. 2016. Search techniques for the web of things: A taxonomy and survey. *Sensors* 16, 5 (2016), 600.

Received January 2017; accepted May 2017