

Edge-to-Edge Resource Discovery using Metadata Replication

Iliir Murturi, Cosmin Avasalcai, Christos Tsigkanos and Schahram Dustdar

Distributed Systems Group

TU Wien

Vienna, Austria

{imurturi, c.avasalcai, c.tsigkanos, dustdar}@dsg.tuwien.ac.at

Abstract—Edge computing has been recently introduced as an intermediary between Internet of Things (IoT) deployments and the cloud, providing data or control facilities to participating IoT devices. This includes actively supporting IoT resource discovery, something particularly pertinent when building large-scale, distributed and heterogeneous IoT systems. Moreover, edge devices supporting resource discovery are required to meet the stringent requirements prevalent in IoT systems including high availability, low-latency, and privacy. To this end, we present a resource discovery platform for IoT resources situated at the edge of the network. Our approach aims at providing a seamless discovery process that is able to (i) extend the covered area by deploying additional edge nodes and (ii) assist in the development of new IoT applications that target already available resources. Within our proposed platform, devices located in a certain proximity connect and form an edge-to-edge network that we call an edge neighborhood – our edge-to-edge metadata replication platform enables participating devices to discover available resources. Our solution is characterized by absence of centralization, as edge nodes exchange metadata about available resources within their scope in a peer-to-peer manner.

Index Terms—Edge Computing, Internet of Things, Resource Discovery

I. INTRODUCTION

The vision of Edge Computing and Internet of Things (IoT) is an composite system where interconnected edge nodes provide and manage a huge number of resources. The convergence between two concepts changes the way we today conceptualize IoT as a large number of things connected to the Internet, which are often distributed [1]. Hence, in the future the IoT needs to be thought of as an ecosystem of discrete computing resources connected to the Internet-enabled devices [2] - at the edge of network [3]. When this vision will be implemented fully, millions of IoT resources will be available and linked. This brings new challenges in edge computing such as performing resource discovery and resource management in a fast, convenient and user-friendly manner.

Resource discovery [4] is an important challenge in any distributed system such as distributed clouds [5], peer-to-peer networks [6], and grids [7]. It is essential to achieve pervasive behavior, in which it devices and services making up the Internet of Things perceive available resources and utilize them. In IoT systems, such pervasive behavior is only considered locally and achieved by taking into account available

resources located in the vicinity of the user (i.e., a meeting room) or in a local context (i.e., the overall apartment). Moreover, stringent requirements (i.e., high availability, low latency, privacy [8]) of IoT systems have recently suggested the architectural placement of a computing entity closer to the network edge [9]. Hence, edge computing is seen as a promising approach to support the IoT in scenarios where it is required to achieve low communication latency and high availability between the edge node and the end-users [10].

To overcome these shortcomings, edge devices can be introduced as an intermediary entity between applications and surrounding IoT resources. These edge entities are heterogeneous devices which may offer various computational resources to local devices [11]. In addition, an edge device may collect information about the other resources available in their environment. These resources can be referred to as computational, sensing, context data or other types of domain-specific resources that software-intensive devices may take advantage of to achieve their goals. IoT applications may utilize different capabilities of an edge device benefiting from high connectivity and awareness of other surrounding resources. However, in order to meet the stringent and unpredictable requirements of IoT applications, therefore a wider view of available resources is required. Thus, edge devices may collaborate by exchanging information about their surrounding resources and providing a service discovery for a local IoT applications.

We assume that resources share their functionality descriptions to the nearest edge device. We define two types of resources on the basis of their access types: public resources (i.e., resources can be shared) and private resources (i.e., resources can only be accessed locally). Each edge device collects resource descriptions in its vicinity and shares the public ones with the rest of the edge devices. Hence, other edge devices receive public descriptions and store them locally. Since all edge nodes are connected in a Peer-to-Peer (P2P) fashion and performing a resource discovery algorithm on the entire network is computationally demanding, we propose a solution where all devices found in a certain proximity will connect, forming an edge-to-edge network that we call an edge neighborhood. By applying our proposed approach in such a neighborhood, all resources shared in that specific area

can be discovered. For example, edge devices deployed in a building will form such a neighborhood – similarly, devices on a certain street, intersection or district can form such a group neighborhood. Moreover, such edge neighborhoods are expected to collaborate to fulfill IoT application requirements.

In this paper, we propose a novel resource discovery ecosystem for resources in an IoT network. We define an ecosystem as a collaboration of edge devices, information and people with the purpose of creating an IoT platform that supports the development of new IoT applications. The system aims to (i) aid developers in creating IoT applications specific for the available resources at the edge of the network, (ii) provide service discovery for IoT applications at the edge of the network, (iii) offer a platform to find available resources distributed over large areas and (iv) help public institutions (e.g., a fire department) to identify areas where there is a need of deploying new edge devices to enable required resources. Furthermore, we describe a new decentralized edge-to-edge metadata replication framework at the core of our ecosystem.

The rest of the paper is structured as follows. Section II gives an overview of our approach along with a motivating example used throughout the paper. Section III gives an overview of the proposed resource discovery ecosystem. Section IV defines a metadata model to describe network-based resources, and Section V describes in detail the proposed resource discovery using metadata replication across the edge nodes. Related work is considered in Section VI. Finally, Section VII concludes the paper.

II. MOTIVATING SCENARIO

We consider a smart city [12] environment as a motivating scenario for our decentralized resource discovery framework. Such a smart city is the result of combining successfully the Edge Computing paradigm with IoT devices such that new IoT applications can be created. Generally speaking, the backbone of such a city is represented by IoT applications able to take advantage of all edge resources deployed at the edge of the network and provide solutions for improving the daily lives of people and monitoring environmental conditions [13]. Concrete instantiations of this vision include applications like a smart traffic lights and smart energy. However, with no possibility of discovering the available resources at the edge of the network, developing applications for different areas of the city is considerably harder.

To motivate our proposed resource discovery framework, we consider the following use case. In a smart city, there is a fire department in charge of emergency scenarios involving smart buildings. A smart building provides different resources to the residents – each apartment has smart devices and sensors (i.e., temperature, presence, fire alarm, smoke detectors etc.). Since resources may collect private data of a specific person (e.g. a presence sensor), the data should not be accessed by other users from a remote location. Hence, resources in the network are divided into private and public resources. Furthermore, each apartment is equipped with an edge node which controls its surrounding resources (e.g., is a controller for a smart

home) and connects in a P2P manner with other edge devices from that specific building, forming what we call an edge neighborhood.

In our example scenario, we assume that every apartment has a fire detector resource designed to detect and respond to the presence of fire. The fire detector is a public resource, which means that all neighbors have the knowledge of which apartment has this resource. The fire department can access our proposed resource discovery platform that contains a list of all available resources in the city and query to find all buildings that already have a fire detector resource. Based on this information, the department can create an application that monitors these buildings by accessing the available resources and in case of emergency react and dispatch a fire brigade personnel for intervention. Furthermore, the fire department can deploy new edge devices in buildings that lack from these resources.

III. RESOURCE DISCOVERY ECOSYSTEM

An overview of the resource discovery framework is presented in Figure 1. The framework consists of three layers with the purpose of providing a seamless resource discovery for (i) extending the covered area with more deployed edge devices such that more resources are available and (ii) developers to create IoT applications for a specific area.

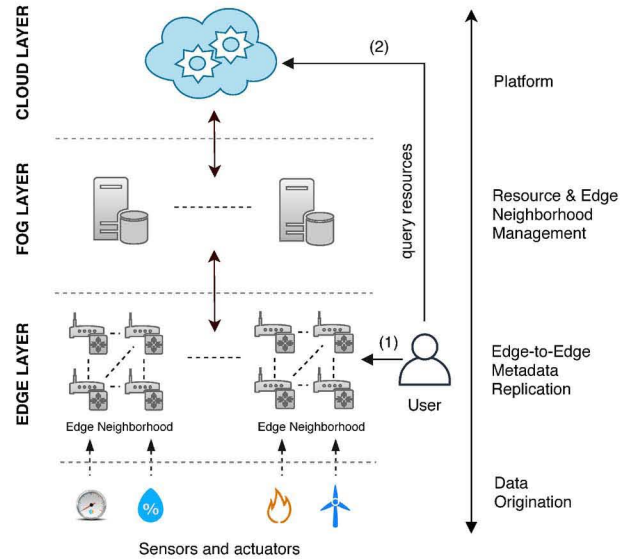


Fig. 1. Resource Discovery framework overview.

The first layer consists of multiple static edge devices deployed in a certain area (i.e., building, home, street, etc.). Each such device contains sensors and resources that could aid at building different IoT applications. Since privacy is imperative to such a framework, we classify resources on every edge device as public (i.e., the resources are shared with other neighbors) and private (i.e., resources can only be processed locally). By applying our edge-to-edge decentralized metadata replication system in such a neighborhood, we can discover all resources shared in that specific area. For example, all

edge devices deployed in a building or devices on a certain street, intersection or district can form such a neighborhood. At this level, each node in his group contains a local storage of shared resources (e.g., sensors, actuators, etc.) of each other node in that particular group. This layer also provides a local search mechanism for locally stored content where user IoT device may realize queries and select resources the edge neighborhood.

The second layer represents the fog layer and is in charge of offering the possibility of accessing the data available at layer one to different parties. Fog Computing [14] extends the Cloud Computing paradigm to the edge of the network, thus enabling a new breed of applications and services. In this layer, there is a fog node that manages one or more neighborhoods. Each such fog node collects information about deployed edge devices and the available resources from each neighborhood and creates (i) a description of the resources discovered for each particular group, (ii) an IP address of the group that gives the opportunity of other edge devices (i.e., a laptop) of users to connect and use the available resources, (iii) a list of keywords that describe each resource in a group and help to build easier queries, and (iv) assists on forming new edge mesh networks to fulfill application requirements. Hence, this layer assists the third layer with the possibility of doing queries and select only the neighborhoods that will have the resources from one particular area (e.g., find all neighborhoods that have fire detectors in the city). Note that we ignore the effects of network routing and assume that each device can connect to each other.

The third layer is called the Cloud layer and represents the location where a resource discovery platform is deployed. Such a platform contains a list of all discovered resources in the city and their description collected from the fog layer. Users can connect to it and find resources by creating specific queries. The user can create queries to find available resources with specific requirements (e.g., fire detector, smoke detector, etc.). Finally, when the results are returned, the user can connect directly to each neighborhood, create and deploy an IoT application.

IV. IOT RESOURCE DESCRIPTION

A resource can be described by providing certain core information about the functionality and its properties. This type of description is known as the metadata of the resource, which may be provided by the manufacturer or application developer. In order to accomplish our goal of describing resources in a pervasive environment, we adopt a similar approach to Barnaghi et al. [15]. We propose a resource representation structure which provides seven main properties as described below:

Resource identification provides basic information on the resource such as resource name, a unique resource identification ID, and the edge node ID which handles the resource description. Resources are described by a number of keywords which helps developers to search resources within the edge node. For example, knowing the location of the resources

is a way for the user to locate resources – in smart cities for instance, the location parameter may be used to discover resources such the temperature sensors in a certain area. In addition, it provides a product description (e.g., device brand, category, description, etc.).

Resource connectivity provides information such as the IP address and port number to reach the resource through the overlay network. In addition, it provides an information related to the connectivity status (e.g., connected or disconnected). As an example, a user application might request a specific available resource from the edge node which in turn gets the information how to reach the desired resource.

Resource capability describes the ability of the resource perform some action (e.g., the resource may provide storage capabilities to others). In addition, specific resources may provide sensing operations to change the state of the resource (e.g., increasing/decreasing temperature).

Resource accessibility provides information related to the resource access policy. We define two types of resources that participate in the edge network: i) public resources and ii) private resources. All edge nodes that join the network can discover and replicate public resources. Private resources are not discoverable by other edge nodes – such resources are accessible only locally.

Resource output provides information related to the output of the resource (i.e., unit). For example, a specific user application might require a temperature sensor which provides values only in Fahrenheit. However, we note that metadata does not contain any current value provided by the resource.

Resource location describes the properties of the resource where it is located. It can be described through the given geographical position or by a logical location. A resource can only have a link to one location instance.

Resource administration domain defines the owner of the resource shared in the edge network. For example, the fire department can monitor the activity within the administrative domain to ensure that the monitored resources remain consistent.

Representation of resources is mainly focused on representation models using ontologies. We propose that resource description to be formatted in a JavaScript Object Notation (JSON). The rationale of exchanging metadata over JSON is:

- Metadata using JSON may have a very rich description of the resource while being lightweight and machine readable.
- Size of metadata in JSON is small and replication process across many nodes is possible. Hence, JSON documents considerably reduce the network traffic between edge nodes compare to the other formats (e.g., XML [16]).

The proposed framework focuses on taking a set of resources around an edge node and producing a document with public resource descriptions that surround the edge node. Resource descriptions can be constructed by the application developer or the manufacturer itself.

V. EDGE-TO-EDGE METADATA REPLICATION

Our motivating example differs from current IoT resource discovery in two major ways. First, resource discovery provided by this framework is decentralized. Second, edge nodes found in a certain proximity will connect forming an edge-to-edge network that we call an edge neighborhood. By applying our proposed approach in such a neighborhood, we can discover all resources shared in that specific area. Hence, the focus of this framework is to discover resources near a user and the resources which are shared in the edge neighborhood. In the rest of this section, we discuss in detail the proposed approach for resource discovery using metadata replication across the edge nodes. Firstly, we elaborate the interaction of each component in a proposed system and secondly, we provide a detail description of each component of the architecture.

A. System architecture

Our system consists of three main components: the Edge-to-Edge Communication, the Metadata Container, and the Local Search Engine facility as presented in Figure 2. We assume that an edge node before joining the edge network contains metadata descriptions provided by resources in its surroundings. In this paper, we consider only built-in resources in the edge device and stationary resources that support IP based communication.

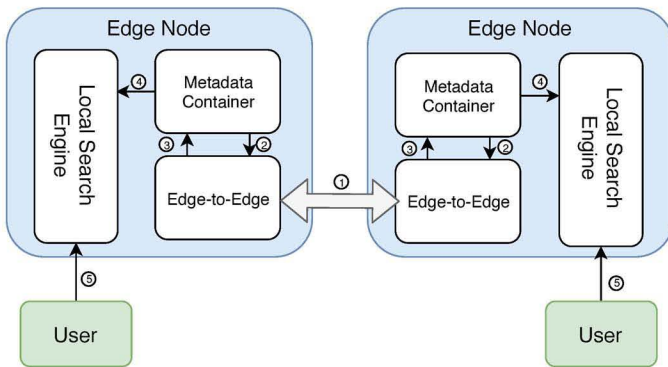


Fig. 2. Edge-to-edge Metadata replication: High level architecture.

Communication (1) between edge nodes is achieved through the Edge-to-Edge Communication component. This component specifies the structure of the network and their exchange information (i.e., edge node ID, IP, port) through node lookups. Each edge node stores locally exchanged information and updates it whenever a new node joins the network. The Metadata Container is responsible to manage data in the edge node. It manages which resource descriptions can be shared in the network and which can be accessed only locally. We merge all descriptions of public resources into a single metadata document and name it with the edge node ID. The metadata document is published by the Edge-to-Edge Communication (2) to the edge neighborhood.

Edge-to-Edge Communication is also responsible to obtain (3) the public metadata documents of all active edge nodes.

Metadata exchange begins by asking each active node to provide its public resources. The local search engine component (4) subsequently allows edge nodes to store data locally and provides a search mechanism for stored content. The user interacts (5) with the Local Search Engine Component via a RESTful APIs.

B. Edge-to-Edge Communication

The communication between edge nodes in the proposed architecture is realised through implementing the Kademlia Protocol [17]. Our study revealed several distributed hash table solutions to consider for our framework such as Chord [18], Pastry [19], and Tapestry [20]. We propose to use Kademlia as the edge communication protocol due to its automatic spreading of contact information and minimal Remote Procedure Calls. Moreover, one of the most important features of the Kademlia network is the $\mathcal{O}(\log(n))$ lookup time. The process of finding nodes and resource descriptions in this type of network is fast and efficient [17].

Kademlia is a distributed hash table (DHT) for decentralized peer-to-peer computer networks. A Kademlia network consists of nodes where each of the nodes has a unique 160 bit ID as an identifier. Nodes in the Kademlia network communicate using User Datagram Protocol. Moreover, the participating nodes exchange their information through node lookups. An overlay network is formed where each node is identified by its own node ID. The provided ID will serve as a direct map to the file hashes and where to obtain the requested resource. Beside the unique ID, it maintains a routing table and a DHT. A routing table maintains a list for each bit of the node ID (e.g., node ID consist 160 bits means that will keep 160 such lists). A routing table is divided into created lists known as buckets - each bucket contains contact information and the distance from the current node. Contact information reside into one of the buckets in which it contains the node ID, IP address, and port number of the other node. Buckets in the routing table are updated every time when a new node join the network. In addition, new edge nodes can be bootstrapped by knowing basic contact information of any other reachable DHT nodes in the network. The DHT segment stores key/value pairs where the key is the name of the public metadata document and the value is the network location of the document.

C. Metadata Container

This component is responsible for managing data in an edge node. It provides a mechanism which synchronizes data information with the Local Search Engine and the Edge-to-Edge Communication. As discussed at the beginning of the section, we assume that each node before joining the edge network contains (1) the metadata descriptions provided by the resources in its surroundings, as presented in Figure 3. We assume the metadata is transmitted only once to the nearest edge node.

Metadata is collected from each resource supplied to the edge node as a single JSON document. The process begins by storing (2) found metadata to the Local Search Engine. Each

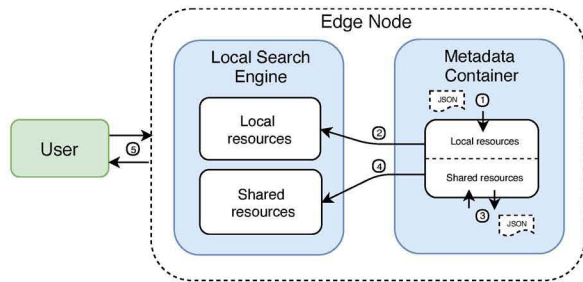


Fig. 3. Interaction between Local Search Engine and Metadata Container.

metadata description is analyzed based on their accessing type such as public or private. Private metadata documents will remain locally in the edge node and no further operation will be required. Public metadata documents are merged into a single JSON document and shared (3) with other edge nodes. For the first interaction between nodes, we conclude that providing a single metadata document for all public resources reduces unnecessary communication and search operations. This document is updated whenever new public resources are added.

The Metadata container is responsible to monitor the availability of the local resources in its surroundings (i.e., connected/disconnected). The edge node transmits a message to the rest of the nodes to update the availability of a specific resource whenever the resource becomes unavailable. If the resource remains inactive for a while, each edge node removes it from its local storage. In addition, when a newly public resource is added to the edge device, it broadcasts a message to the others to obtain and store it locally. This broadcast message provides basic information on how to obtain a description of the resources from the edge device.

D. Local Search Engine

Supporting queries such as showing all temperature sensors in a certain area is becoming highly desirable for IoT applications. The local search engine may assist in the development of new IoT applications that target already available resources at the edge neighborhood. In addition, it may also support the discovery of resources at the edge of the network for a user IoT application. Our study revealed several search engines such as Apache Solr [21] and MongoDB [22]. As the local search engine, we identify Elasticsearch [23] as a likely candidate to meet the desired component objective.

ElasticSearch is a document-oriented database which allows queries to be made through using JSON language and makes full-text search accessible via RESTful API, while a JSON-style domain-specific language allows writing complex queries. Elasticsearch resource descriptions are inserted (3), (4) based on their accessing types, as presented in Figure 3. Moreover, locally stored content helps a roaming IoT device to realize queries and select resources in the edge network. Hence, such requests (5) are therefore submitted and processed by the edge node.

VI. RELATED WORK

Resource discovery is a critical challenge for IoT application performance, where it must ensure that performance requirements are met and some quality of service constraint is satisfied. We discuss related work in three parts. The first part provides an overview on the replication of metadata, while the second part outlines related work on nodal collaboration techniques. Finally, we discuss resource discovery in IoT.

Our resource discovery approach is based on browsing the resources locally rather than using a keyword to search the entire network [24]. By replicating metadata between the edge nodes, we enable users to perform locally various complex queries (i.e., location-based, keyword-based, etc.). A P2P-based distributed database solution that supports bidirectional replication is provided by CouchDB [25]. Our architecture has some similar components, but we emphasize metadata replication as a single document, as opposed to database replication. A metadata replication process has many benefits such as enabling high availability by removing a single point of failure [26] or to achieve higher reliability and workload balancing among peers [27].

Researchers focus primarily on nodal collaboration in Fog Computing [14], considering three main techniques such as cluster [28], peer-to-peer and master-slave [29] conceptions. Generally, in Fog Computing communication between nodes is realized in P2P manner [30]. Due to their fully distributed nature, P2P architectures have been shown to be both scalable and reliable [31].

In IoT systems, selecting appropriate resources and services that satisfy users requirements is a challenge. Tanganelli et al. [24] proposed an Edge-centric distributed solution to federate different IoT gateways deployed in fog nodes close to IoT domains. The main goal of the proposed approach is providing applications a service for global discovery and access of IoT resources in the federated domain without taking into account their location. DHT based resource discovery has also been proposed, for instance, see [32]. OpenIoT [33] proposes an open service framework for the Internet of Things, which facilitates entrance into the IoT-related mass market, establishing a new ecosystem for the Internet of Things with the widespread adoption of IoT-related devices and software. Wang et al. [2] propose an architecture for service discovery in smart cities. The proposed framework focuses on taking a set of devices around a citizen, and proactively producing a list of services that surround the user using his preferences.

VII. CONCLUSION AND FUTURE WORK

Resource discovery is an important challenge in many contemporary distributed systems. Building a network of edge nodes where each stores information about its surrounding resources, affords an unprecedented way to discover and utilize them in applications. We introduced a three-layer resource discovery platform for resources at the edge of the network. The platform provides a seamless resource discovery process that application developers can access and use to extend the covered area by deploying more edge nodes and assists in

developing new IoT applications that target already available resources. We proposed a solution where all devices found in certain proximity connect and form an edge-to-edge network. We presented a new metadata model to describe the functionality of the edge device resources and IP-enabled resources. To discover all resource shared in such area, we proposed a decentralized edge-to-edge metadata replication framework.

This paper is only a small step towards a general framework for resource discovery in edge networks. Regarding future work, we first plan to implement the proposed decentralized edge-to-edge metadata replication framework discussed. It remains an important open question what is a reasonable number of edge nodes forming an edge neighborhood. We plan to carry out a series of experiments taking into account various factors, such as the number of metadata resource descriptions and the number of concurrent requests that can be supported. Beside a detailed evaluation, the platform itself as well as the implementation of the fog node functionality remains to be built. In addition, more work is needed to investigate current technologies that enable IoT applications to access resources. We ignored network routing, but we consider this an important aspect to be treated in future work.

ACKNOWLEDGMENT

Research partially supported by the TUW Research Cluster Smart CT and the EU H2020 Marie Skłodowska-Curie grant No. 764785 FORA–Fog Computing for Robotics and Industrial Automation.

REFERENCES

- [1] Christos Tsigkanos, Laura Nenzi, Michele Loreti, Martin Garriga, Schahram Dustdar, and Carlo Ghezzi. Inferring analyzable models from trajectories of spatially-distributed internet-of-things. In *1th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2019, Montreal, Canada, May 25-26, 2019*. IEEE Computer Society, 2019.
- [2] Edward Wang and Richard Chow. What can i do here? iot service discovery in smart cities. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.
- [3] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.
- [4] Soumya Kanti Datta, Rui Pedro Ferreira Da Costa, and Christian Bonnet. Resource discovery in internet of things: Current trends and future standardization aspects. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 542–547. IEEE, 2015.
- [5] Stefan Nastic, Sanjin Sehic, Duc-Hung Le, Hong-Linh Truong, and Schahram Dustdar. Provisioning software-defined iot cloud systems. In *2014 2nd International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 288–295. IEEE, 2014.
- [6] Elena Meshkova, Janne Riihijärvi, Marina Petrova, and Petri Mähönen. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer networks*, 52(11):2097–2128, 2008.
- [7] Nima Jafari Navimipour, Amir Masoud Rahmani, Ahmad Habibzad Navin, and Mehdi Hosseinzadeh. Resource discovery mechanisms in grid systems: A survey. *Journal of Network and Computer Applications*, 41:389–410, 2014.
- [8] Nianyu Li, Christos Tsigkanos, Zhi Jin, Schahram Dustdar, Zhenjiang Hu, and Carlo Ghezzi. Poet: Privacy on the edge with bidirectional data transformations. In *2019 IEEE International Conference on Pervasive Computing and Communications, PerCom 2019, Kyoto, Japan, March 11-15, 2019*. IEEE, 2019.
- [9] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

- [10] Marjan Gusev and Schahram Dustdar. Going back to the roots: the evolution of edge computing, an iot perspective. *IEEE Internet Computing*, 22(2):5–15, 2018.
- [11] Ju Ren, Hui Guo, Chugui Xu, and Yaoyue Zhang. Serving at the edge: A scalable iot architecture based on transparent computing. *IEEE Network*, 31(5):96–105, 2017.
- [12] Annalisa Cocchia. Smart and digital city: A systematic literature review. In *Smart city*, pages 13–43. Springer, 2014.
- [13] Xhevahir Bajrami and Ilir Murturi. An efficient approach to monitoring environmental conditions using a wireless sensor network and nodemcu. *Elektrotechnik und Informationstechnik*, 135(3):294–301, 2018.
- [14] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.
- [15] Suparna De, Payam Barnaghi, Martin Bauer, and Stefan Meissner. Service modelling for the internet of things. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 949–955. IEEE, 2011.
- [16] Audie Sumaray and S Kami Makki. A comparison of data serialization formats for optimal efficiency on a mobile platform. In *Proceedings of the 6th international conference on ubiquitous information management and communication*, page 48. ACM, 2012.
- [17] Petar Maymounkov and David Mazieres. Kademia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.
- [18] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.
- [19] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 329–350. Springer, 2001.
- [20] Ben Yanbin Zhao, John Kubiatowicz, Anthony D Joseph, et al. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. 2001.
- [21] Apache solr, <https://lucene.apache.org/solr/>.
- [22] MongoDB, <https://www.mongodb.com/>.
- [23] Elasticsearch, <https://www.elastic.co/>.
- [24] Giacomo Tanganelli, Carlo Vallati, and Enzo Mingozzi. Edge-centric distributed discovery and access in the internet of things. *IEEE Internet of Things Journal*, 5(1):425–438, 2018.
- [25] Couchdb, <http://couchdb.apache.org/>.
- [26] Feng Wang, Jie Qiu, Jie Yang, Bo Dong, Xinhui Li, and Ying Li. Hadoop high availability through metadata replication. In *Proceedings of the first international workshop on Cloud data management*, pages 37–44. ACM, 2009.
- [27] Benjamin Ahlborn, Wolfgang Nejdl, and Wolf Siberski. Oai-p2p: A peer-to-peer network for open archives. In *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 462–468. IEEE, 2002.
- [28] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.
- [29] Wangbong Lee, Kidong Nam, Hak-Gyun Roh, and Sang-Ha Kim. A gateway based fog computing architecture for wireless sensors and actuator networks. In *Advanced Communication Technology (ICACT), 2016 18th International Conference on*, pages 210–213. IEEE, 2016.
- [30] Heng Shi, Nan Chen, and Ralph Deters. Combining mobile and fog computing: Using coap to link mobile device clouds with fog computing. In *Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on*, pages 564–571. IEEE, 2015.
- [31] Stratis Ioannidis and Peter Marbach. Absence of evidence as evidence of absence: A simple mechanism for scalable p2p search. In *IEEE INFOCOM 2009*, pages 576–584. IEEE, 2009.
- [32] Simone Cirani, Luca Davoli, Gianluigi Ferrari, Rémy Léone, Paolo Medagliani, Marco Picone, and Luca Veltri. A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE Internet of Things Journal*, 1(5):508–521, 2014.
- [33] Jaeho Kim and Jang-Won Lee. Openiot: An open service framework for the internet of things. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 89–93. IEEE, 2014.