Indie Fog: An Efficient Fog-Computing Infrastructure for the Internet of Things

Chii Chang and Satish Narayana Srirama, University of Tartu

Rajkumar Buyya, University of Melbourne and Manjrasoft Pty Ltd.

Fog computing can help with some of the Internet of Things' limitations but also faces challenges of its own. The Indie Fog infrastructure could be a flexible, cost-effective way to cope with these challenges.

The Internet of Things (IoT) has created many technical and business opportunities but also has its limitations. For example, the traditional cloud-based IoT can experience more latency than many target applications can tolerate.

Fog computing, which moves computation and networking closer to the network edge, reduces the need to communicate via the cloud and thus decreases latency.

However, fog computing faces its own challenges, such as equipment that works only with the service providers' technologies or only with cellular networks.

In response, we have proposed the Indie Fog infrastructure, which utilizes consumers' networking devices—such as their Wi-Fi access-point routers—to provide IoT service providers with a fog-computing environment. This reduces the need for IoT service providers to deploy their own devices throughout a fog system.

Indie Fog would be flexible and cost-effective, support the IoT, and work with today's technologies.

THE IoT'S LATENCY CHALLENGE

The IoT represents the Internet's next stage, in which common objects such as vehicles, home appliances, office equipment, manufacturing facilities, and consumer electronic devices connect to the Internet.[1] This facilitates important new software-based services such as smart cities, smart factories, smart agriculture, and ambient assisted living (AAL).
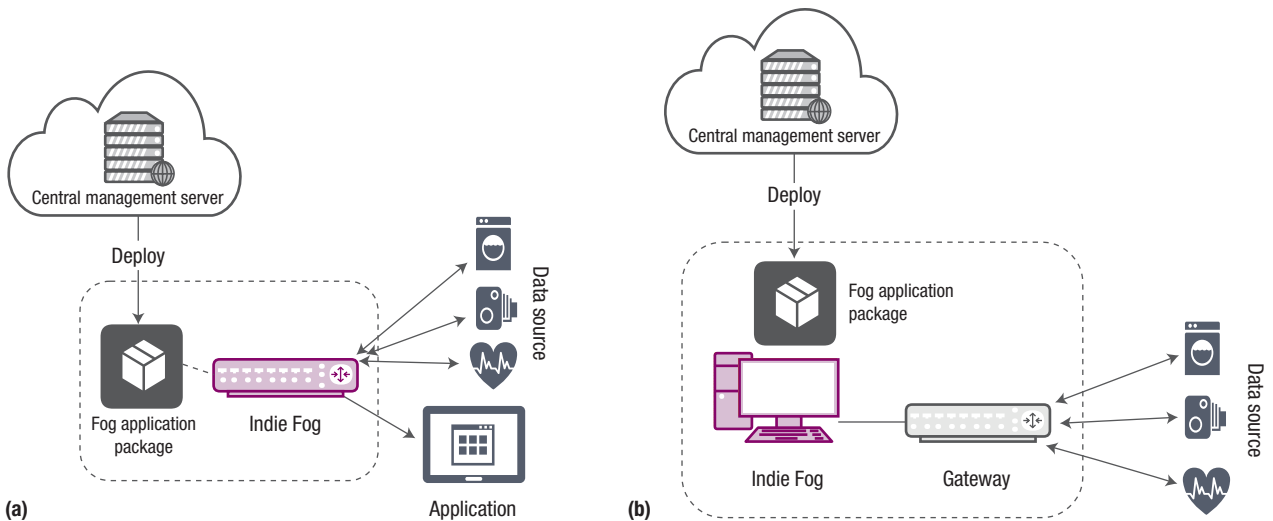
Figure 1. Two types of Indie Fog deployments. (a) In the integrated form, the functionality is provided by a gateway device integrated into the Indie Fog server, which connects to data sources and user devices. (b) In the collaborative form, the functionality is provided by a workstation in the same subnet as the data–source devices, to which it connects via an Internet gateway.

In the IoT's architecture, a cloud-based central server manages all activities of IoT devices residing at the network edge. This cloud-centric IoT (CIoT) requires these devices to send their data to the central server and wait for its response, thereby creating latency. Many emerging, data-intensive IoT-driven applications, such as AAL and smart-traffic services, can't tolerate latency. Researchers have thus developed fog computing.[2]

FOG COMPUTING'S CHALLENGES

By operating at the network edge, fog computing brings computation and networking closer to users and data sources. This reduces latency and increases computational and communication efficiency.

Typically, network-infrastructure providers, ISPs, or cloud-service providers offer fog services via customer-premises equipment (CPE) designed specifically for the providers' applications. For example, an ISP might enable fog services from the proprietary home Internet router the provider sends to the customer. This is different from public-cloud systems, in which third-party infrastructure companies offer generic platforms that work with multiple service providers.

Recently, the European Telecommunications Standards Institute (ETSI) introduced the Mobile Edge Computing (MEC) architecture,[3] which lets CIoT providers utilize cellular base stations' virtual machine (VM)-enabled servers to perform distributed computational or networking processes at the cellular-network edge. However, because MEC works only with cellular networks, its usefulness is limited.

To address these limitations, we propose Indie Fog.

CONSUMER AS PROVIDER

The popular Consumer as Provider (CaP) service-provision model[4] has triggered a new form of service provision in multiple domains. For example, the MQL5 Cloud Network distributed-computing project (cloud.mql5.com) uses consumers' equipment to perform various distributed computing tasks. And Fon (fon.com) utilizes CPEs to establish a global Wi-Fi network. These examples confirm that many consumers are willing to let providers pay to use their equipment for offering services.

We envision the CaP model extending to fog computing, and we use this approach in Indie Fog. Individuals, not just organizations, could configure their devices to provide public fog services. The independent nature of such a deployment inspired us to use the name Indie Fog, much as the term is used in areas such as indie art, indie music, indie design, and indie games.[5]

INDIE FOG'S ROLE

In general, Indie Fog services come in two forms: integrated and collaborative (see Figure 1). In the integrated version, the Internet router itself provides the virtual server that acts as the fog node. In the collaborative version, a computer connected to the Internet router provides the fog node. Both Indie Fog types operate at the
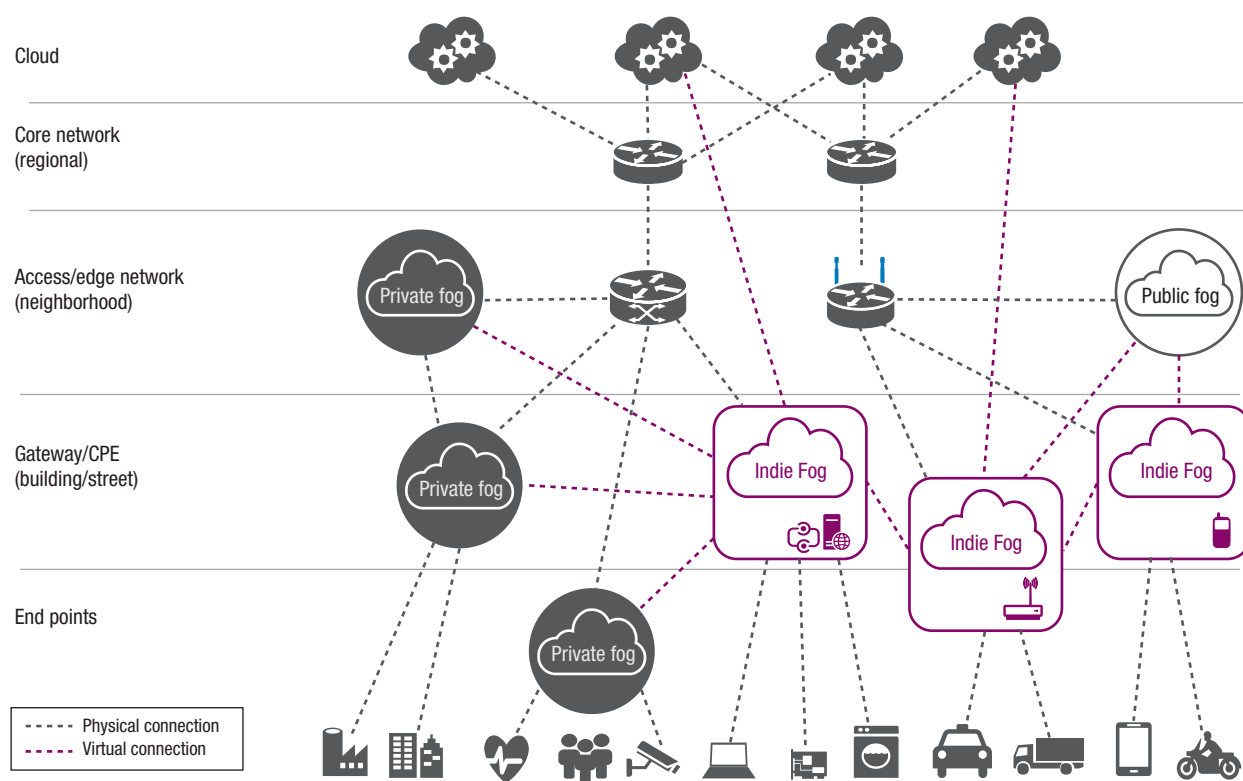
Figure 2. Customer–premises equipment (CPE), computers embedded in vehicles, or mobile devices with Internet connections can host an Indie Fog system. Because the technology is software defined, Indie Fog nodes provide virtual connectivity across different layers of the cloud–centric Internet of Things (IoT) architecture.

fog computing architecture's gateway layer.

According to the OpenFog Consortium,[2] the fog-computing architecture has multiple layers (see Figure 2). In general, the fog server can be hosted in devices at the access layer (such as a citywide Internet gateway), at the gateway layer (such as within a building or in the street), or within various objects (such as vehicles, laptops, or smartphones) that serve as IoT end points.

Commonly, service providers follow the private fog model, in which they configure a fog server within a device for use only by specified parties. In the public fog model, like that used in the MEC architecture, providers make fog services available via generic platforms to any paying customer with the proper equipment.

Figure 2 shows how Indie Fog servers can be in various consumer-based devices that have OSs and Internet connectivity—including consumer electronics, appliances, smartphones, and computers embedded in vehicles—at the gateway layer.

Indie Fog has the following characteristics:

› *Interoperability.* Indie fog nodes can interact with other fog nodes to create a software-defined network.
› *Scalability.* A CIoT system can form a scalable, software-defined cluster of Indie Fog nodes. Both public and private fog systems can use Indie Fog to extend their networks.
› *Cost efficiency.* Because Indie Fog uses customer equipment, it's less expensive, particularly for large deployments.
› *Security.* Indie Fog nodes provide a sandbox-based software-deployment environment using either hypervisor-oriented VM technologies or containerization technologies. A trusted cloud-based coordinator service manages the registered nodes.

INDIE FOG DEPLOYMENT
There are four basic Indie Fog deployment models.

Clustered Indie Fog
When a group of stationary Indie Fog nodes are in close proximity, such as within an ISP's subnet or in the same building, they are ideal for establishing a software-defined grid-computing cluster (see Figure 3a). Such a cluster can perform, in the edge network, high-performance preprocessing of data collected by sensors and other sources. This reduces the bandwidth needed to send the data to the cloud and the cost of processing it.
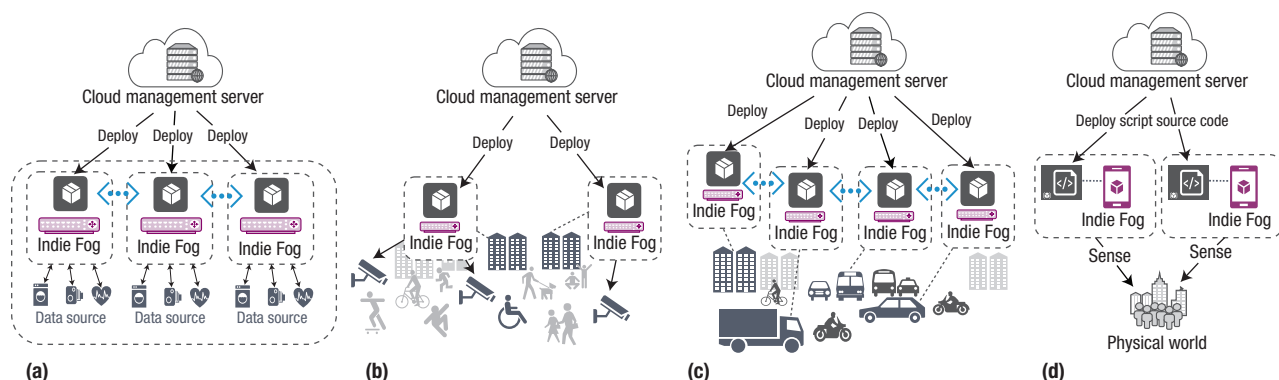
Figure 3. Indie Fog facilitates the deployment of four types of software-defined cloud-centric IoT systems: (a) clustered Indie Fog, (b) infrastructural Indie Fog, (c) vehicular Indie Fog, and (d) smartphone Indie Fog.

Infrastructural Indie Fog

Indie Fog deployed in stationary sensor devices, such as cameras, in an urban outdoor area could provide an infrastructure for ubiquitous applications that require temporary data acquisition and processing in a short period of time (see Figure 3b). For example, scientists could pay to use individuals' or businesses' Indie Fog servers with pre-installed sensor functions to collect and interpret urban data, such as a neighborhood's population density, for research.

Vehicular Indie Fog

Indie Fog is not limited to stationary nodes. In urban areas, vehicles could also host nodes. This would facilitate the Internet of Vehicles,[6] in which the CIoT system could dynamically deploy a software-defined vehicular fog[7] for extending the urban-sensing data-acquisition network, which collects traffic, metropolitan transit, and other information of interest to the public (see Figure 3c). Unlike stationary nodes, vehicular nodes could collect and process data on the move, thereby enabling sequential data streams and information on changing environments.

Smartphone Indie Fog

Smart mobile devices might not be able to form a stable cluster-computing environment because smartphone users move regularly and thus have dynamic, random connections. However, mobile-device-based server nodes could still be useful in urban sensing. Smartphones could collect many kinds of sensory data. If smartphone users install Indie Fog servers on their devices, the phones could provide and even preprocess their collected data as a service.[8]

Use cases

There are several promising Indie Fog use cases.

Ubiquitous care. Indie Fog nodes could communicate with standards-based smart devices in public urban areas to help, for example healthcare or eldercare services track their patients in real time.

Software-defined systems. By working with multiple Indie Fog nodes, the CIoT could configure software-defined networks. Similarly, by deploying grid computing across multiple Indie Fog nodes, the CIoT could configure a software-defined cluster-computing infrastructure.

Big data acquisition. Worldwide Indie Fog deployment could enable CIoT systems to stream large volumes of raw sensory data for continuous analysis.

Context brokering. Indie Fog could be used to collect, preprocess, and analyze data for context brokering, which is the collection and analysis of data to provide context for subsequent decision making.

Software-defined IoT orchestration. Deploying a distributed IoT-driven Business Process Management System (BPMS)[9] across a group of Indie Fog nodes could provide a composite IoT service for interacting with wireless sensors, actuators, and readers with minimal low-level programming required. In addition, if the IoT service uses semantic process execution, the BPMS could dynamically and autonomously configure the deployment and execution node to adapt to environmental changes, such as the failure of a fog server.

Commercial service. An Indie Fog node in a bus, for example, could allow an advertiser to dynamically change the content, application, or advertising it provides on seat-back monitors that passengers can view. For example, a seat-back camera and application could identify when a new passenger is in front of the monitor; analyze the person's age, gender, and other characteristics; and try to provide appropriate content, application, or advertising.

INDIE FOG ARCHITECTURE

Users can implement Indie Fog via a service-oriented architecture (SOA), as Figure 4 shows. In general, SOAs
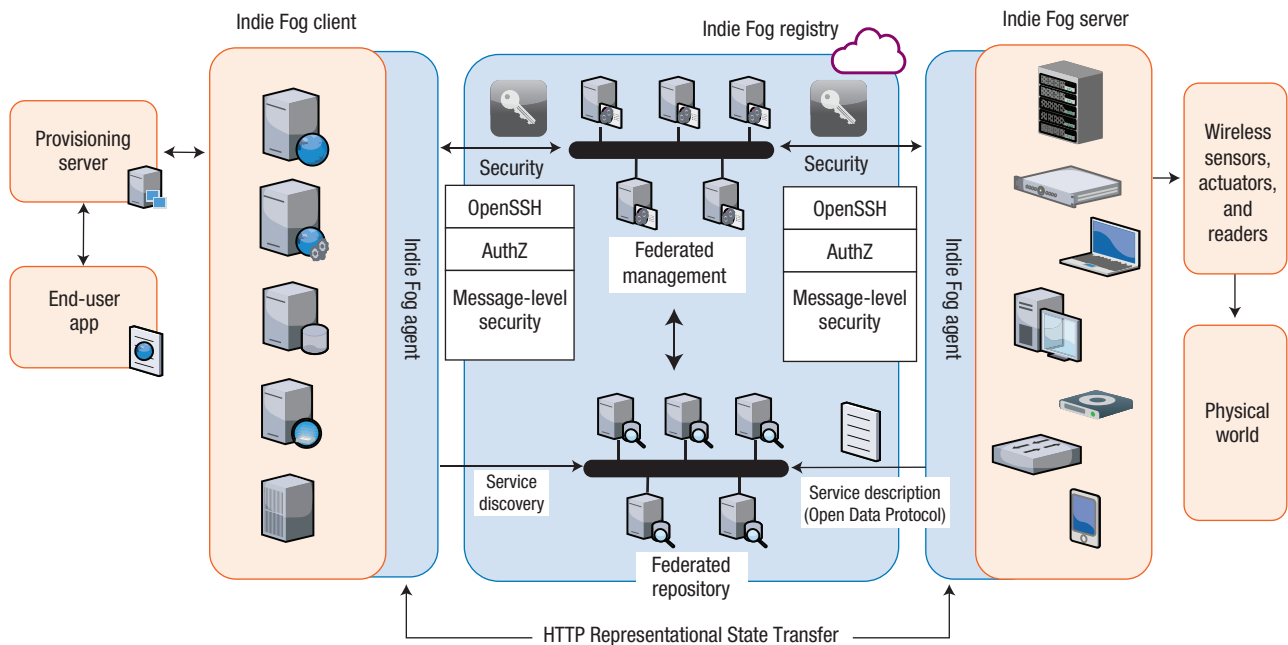
Figure 4. A service-oriented Indie Fog system consists of three core elements: the Indie Fog client; the Indie Fog server; and the Indie Fog registry, formed by federated management servers and federated service-description repository servers.

involve three core entities: the service registry, the service provider, and the service consumer. Based on the World Wide Web Consortium's Web service architecture (www.w3.org/TR/ws-arch), both provider and consumer must host an agent to support communication with other entities.

The registry, generally federated in nature, encompasses the repository of published services and handles identity and security management. After using the registry, the provider and consumer interact on a peer-to-peer basis.

Indie Fog service providers can generally publish their service descriptions to the repository in a standard format such as that called for by the Open Data Protocol (www.oasis-open.org /committees/odata). Consumers could then select their ideal providers based on the providers' location, as well as their server and networking capabilities.

The system's sandbox-based security management provides the trust that encourages consumers to access the software-deployment environment, as well as the permitted storage,

network, sensor, and other components from the provider. However, the provider can't observe consumer activities, which protects privacy.

Figure 5 illustrates a reference Indie Fog architecture with several core elements.

To provide secure services, each Indie Fog server must host an agent that manages service publication and secure access control, and that handles the deployment of consumer applications in the virtualization or containerization environment. The agent also handles hardware-access permissions.

Middleware and adaptors take care of the interaction between the consumer-deployed application and the Indie Fog host's hardware components.

The Indie Fog server describes its hardware specification and location in a standard-format service-description document. The Indie Fog agent manages the document's publishing and updating.

The agent uses its host interface to communicate with other Indie Fog agents. The host interface supports both the request-response and

publish-subscribe service-provisioning mechanisms.

Virtualization or containerization— via, for example, OpenStack++ (elijah .cs.cmu.edu/development.html), Linux Containers (linuxcontainers .org), Docker (www.docker.com), and Open Container Initiative (www .opencontainers.org) technology— enables the runtime environment. This lets Indie Fog clients deploy and operate their applications in the server's sandbox environment.

CHALLENGES AND FUTURE DIRECTIONS
Indie Fog faces fog computing's fundamental challenges,[10] such as dynamic process assignment among cloud- or edge-based resources, mobility awareness, fault tolerance, and security.

Standardization
Indie Fog servers must provide standard deployment environments without requiring users to install a specific API on clients for each server. However, finding a common deployment-platform standard is a challenge, as

the OpenFog Consortium hasn't released a fog-deployment standard.

A possible approach is extending existing service-oriented architecture and web-service standards—such as OData (Open Data Protocol) or MQTT (Message Queue Telemetry Transport)—to develop an Indie Fog specification.

Adaptive service discovery

Indie Fog relies on a global federated registry. The registry helps clients discover the best Indie Fog servers for their applications, which involves both context awareness and semantic descriptions of servers in terms of both spatiotemporal context and resource availability.

It's unlikely that a single registry could satisfy the entire Indie Fog ecology—particularly for organizations with far-flung facilities—because of the need for Indie Fog nodes to be close to one another to minimize latency. Perhaps in the future, there will be a federated registry, but this would increase service-discovery complexity.

Another service-discovery challenge is that Indie Fog servers might not use the same semantics as traditional webservers.

Quality of service

Indie Fog quality of service (QoS) faces three challenges: quality of connectivity, quality of performance, and quality of content or data delivery.

For example, Indie Fog providers might use heterogeneous servers with different computational and networking capabilities. Ensuring these servers' QoS could thus be challenging. The management system might need to provide trust and recommendation schemes to assist both clients and servers.

The quality of the data that the Indie Fog node delivers during information acquisition or context brokering also affects QoS.

Availability

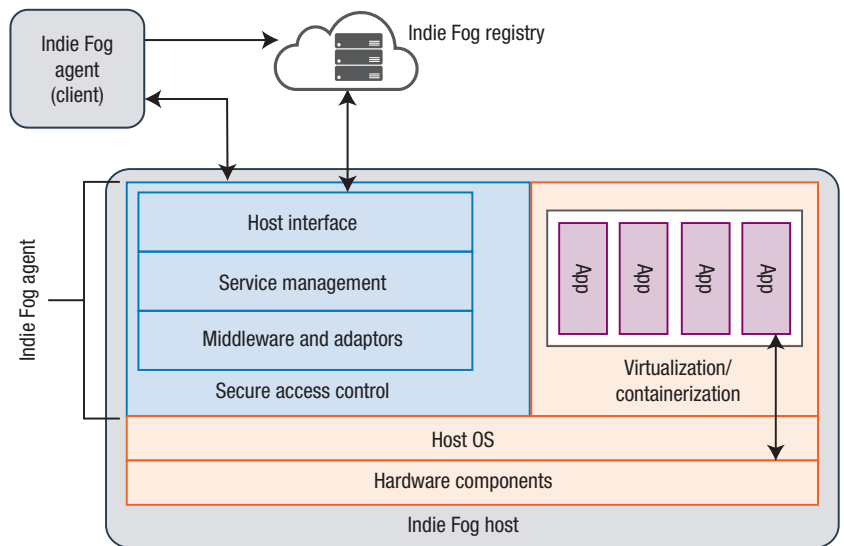Indie Fog might not provide as much availability as that called for by the

Figure 5. The Indie Fog server agent's software handles communications with external entities. The virtualization or containerization environment lets Indie Fog clients deploy and run their customized applications in the system's secure sandbox environment.

MEC telecommunications-industry standard. For example, some users might have mobile devices that don't have strong wireless signals. Also, because Indie Fog uses consumer-based equipment, location would affect its availability. For instance, the service might be less available in low-population suburbs or rural areas because their CPE density is much lower than in urban areas. Because MEC, on the other hand, is provided by equipment located in cellular base stations, it could support national or global availability.

Indie Fog could help CIoT providers reduce fog-deployment costs by using customer equipment and the CaP model to enable a flexible, software-defined infrastructure. The approach could thus be a promising solution for many CIoT service providers. It could also make fog computing more popular and provide business opportunities for CPE owners. ◻

REFERENCES

1. J. Gubbi et al., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645–1660.
2. "OpenFog Reference Architecture for Fog Computing," OpenFog Consortium, Feb. 2017; www.openfogconsortium .org/wp-content/uploads/OpenFog _Reference_Architecture_2_09_17 -FINAL.pdf
3. Mobile-Edge Computing–Introductory Technical White Paper, white paper, Mobile-Edge Computing (MEC) industry initiative, Sept. 2014; portal .etsi.org/portals/0/tbpages/mec /docs/mobile-edge_computing _-_introductory_technical_white _paper_v1%2018-09-14.pdf.
4. R. Sofia and P. Mendes, "User-Provided Networks: Consumer as Provider," *IEEE Communications Magazine*, vol. 46, no. 12, 2008, pp. 86–91.
5. M.Z. Newman, "Indie Culture: In Pursuit of the Authentic Autonomous Alternative," *Cinema J.*, vol. 48, no. 3, 2009, pp. 16–34.
6. M. Gerla et al., "Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds," *Proc. IEEE World Forum on Internet of Things* (WF-IoT), 2014, pp. 241–246.

7. X. Hou et al., "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Trans. Vehicular Technology*, vol. 65, no. 6, 2016, pp. 3860–3873.

8. M. Liyanage, C. Chang, and S.N. Srirama, "mePaaS: Mobile-Embedded Platform as a Service for Distributing Fog Computing to Edge Nodes," *Proc. 17th Int'l Conf. Parallel and Distributed Computing, Applications, and Technologies* (PDCAT 16), 2016, pp. 73–80.

9. C. Chang, S.N. Srirama, and R. Buyya, "Mobile Cloud Business Process Management System for the Internet of Things: A Survey," *ACM Computing Surveys*, vol. 49, no. 4, 2017; doi:10.1145/3012000.

10. A.V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," *Computer*, vol. 49, no. 8, 2016, pp. 112–116.

Computer

WWW.COMPUTER.ORG /COMPUTER

myCS Read your subscriptions through the myCS publications portal at **http://mycs.computer.org**

CHII CHANG is a research fellow in the Mobile & Cloud Computing Laboratory at the University of Tartu's Institute of Computer Science. Contact him at chii.chang@acm.org.

SATISH NARAYANA SRIRAMA is an associate professor at the University of Tartu and head of the Mobile & Cloud Computing Laboratory at the school's Institute of Computer Science. Contact him at srirama@ut.ee.

RAJKUMAR BUYYA is director of the University of Melbourne's Cloud Computing and Distributed Systems Laboratory and CEO of Manjrasoft Pty Ltd. Contact him at rbuyya@unimelb.edu.au.