Toward Building a Safe, Secure, and Easy-to-Use Internet of Things Infrastructure

Yuvraj Agarwal and Anind K. Dey, Carnegie Mellon University

Carnegie Mellon University is leading a multi-institutional effort to build an open infrastructure to support the Internet of Things.

Sal glances at the display near her office door and sees that her next meeting is in 10 minutes. One participant is out of town and the other two people are running late, but the meeting room is still occupied by several people. The display also suggests it might be a good time to get coffee because the lines are short at the cafe downstairs. Her good friend Joe happens to be at the cafe, too. Sal checks an app she recently built and sees that the coffee is freshly brewed. "That simplifies things," she thinks to herself as she heads toward the cafe.

A safe and secure world enabled by the Internet of Things (IoT) promises to lead to truly connected environments, where people and things collaborate to improve the overall quality of life. The IoT will give us actionable information at our fingertips, without us having to ask for it or even recognizing that it might be needed. Consider this example that combines many simple uses of the IoT to cumulatively form an omnipotent assistant:

This is the unique promise of a successful IoT, and is what we are aiming for with GIoTTO, the IoT program at Carnegie Mellon University (CMU) named after the famous Renaissance painter.

NEED FOR AN OPEN INFRASTRUCTURE

Although numerous commercial and academic programs focus on building IoT systems, it's clear that for any IoT stack to be widely adopted, it must be open—without a

0018-9162/16/$33.00 © 2016 IEEE

EDITOR **ROY WANT**
Google; roywant@google.com

singular organization claiming ownership. We must involve the community with the IoT's design, development, and deployment—that means truly open source development, as exemplified by Linux and Android. We also believe that an IoT stack must provide immediate value to anyone wanting to deploy and use it, without requiring substantial integration work upfront. Practically, this means that it must provide important first-class features such as robust machine learning, easy end-user programming, security, and privacy. Our vision of the GIoTTO stack, which we are developing at CMU, is shown in Figure 1.

GIoTTO is an open source infrastructure intended to support the construction, maintenance, and use of IoT-enabled environments. We formed our team at CMU shortly after Google held an open call for proposals on the Open Web of Things. We responded and received the lead award on what is now known as the IoT Expedition (www.iotexpedition.org), which includes partners at Cornell Tech, the University of Illinois, and Google. The IoT Expedition's goals match those of GIoTTO, and the project has adopted GIoTTO as its software infrastructure. Each partner will contribute to and build on GIoTTO to demonstrate its value through a series of living laboratories at each site.

IOT CHALLENGES

The number of IoT-connected devices is expected to grow to 21 billion by 2020 (www.gartner.com/newsroom/id/3165317), presenting a major market opportunity for established hardware (such as Intel, Apple, Qualcomm, ARM, Samsung, and LG) and software vendors (such as Google, Microsoft, and IFTTT) across the world, in addition to spawning new entrepreneurship opportunities. These companies are working on producing IoT devices, software, and services to develop an interconnected world. Although the vision of an IoT-enabled future is enormously compelling, several key challenges must be addressed before it can become a reality. These challenges are related to three critical questions:

- How can we build an IoT infrastructure that is safe, secure, and private from the ground up? Safety implies that IoT devices won't do anything unexpected or unintended. Security implies that IoT devices only allow authorized entities, whether computer programs or humans, to access their services. Privacy implies that IoT devices don't access or leak private user data either directly or indirectly without a clearly defined, and verifiable, purpose being presented to and accepted by users.
- How can we leverage the huge amount of data being collected by sensors embedded in all objects? This calls for machine learning and data analytics to be integrated at every level from sensors and actuators to end users.
- How can we enable end users of varying technical ability to manage, interact with, and even control and program IoT-enabled environments? For the IoT to be truly pervasive, IoT systems must be accessible to end users, or they might be discarded along with the multitude of other technologies that showed much promise only to be ignored after deployment.

FROM THE EDITOR

Building on our inaugural column from February, this month's article presents an open program that sets out to explore the Internet of Things' (IoT's) value proposition. Just as the Internet belongs to all of us, I believe this program embodies the principles we hope will be the driving forces behind an equally successful IoT architecture. In this column, researchers from Carnegie Mellon University eloquently outline the tenants and goals of the GIoTTO software stack. *—Roy Want*

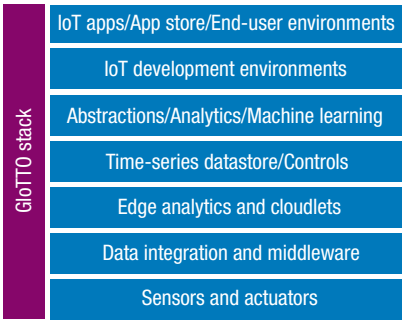

Figure 1. The GIoTTO open source stack.

SECURITY, PRIVACY, AND SAFETY

A key design decision we're addressing in our GIoTTO software stack is to face these concerns from the ground up rather than retrofitting later. Although these concerns have some natural overlaps (for example, safety concerns sometimes imply security issues), we're looking to devise solutions for each one. Safety in IoT means being able to reason about the behavior of IoT devices, especially actuators, and being able to detect and prevent unintended or unexpected behavior. This is a very difficult problem because it requires not only understanding "normal" behavior, but also being able to develop models for device interactions and devising mechanisms to enforce

safety properties. The challenges in ensuring the safety of IoT devices are also due to extreme heterogeneity, lack of standardization, and ineffectiveness of traditional defenses like end-host firewalls and antivirus software. We believe the underlying network provides a key vantage point to not only observe these device-to-device and device-to-cloud interactions, but also to enforce safety and security using techniques from network middleboxes. We're exploring methods to represent these IoT device interactions (for example, crowdsourcing) and devise models for safety policies that can be disseminated to IoT users.

IoT privacy challenges stem from sensors directly or indirectly leaking private information about users, often unbeknownst to them. Although useful for controlling appliances when users aren't home, occupancy sensors can also be used by attackers to determine how often homeowners are out of town. Information about users and their behavior can be inferred from sensors indirectly or by combining information from seemingly unrelated sensors. With the GIoTTO stack, we propose several ideas to help manage privacy. First, GIoTTO provides multiple tiers of access to sensor data, from the most sensitive (highest granularity, such as microphone data) to the least sensitive (low granularity, such as processed audio data to extract amplitude and frequency features). Second, GIoTTO provides a set of core services along with models for people, places, and things—app developers can access these shared data models, which are updated constantly in the stack. Although there might be a raw sensor for a user's exact location, a more conservative data model might only reveal whether a user is "on campus" or "in a meeting." This approach controls who has access to which tier of data, but also supports privacy policies and audits. Our goal is to eventually expose mechanisms where IoT applications must specify a clearly defined set of purposes of data access that can be checked by our GIoTTO stack and reported to users who own that data.

> A key design decision we're addressing in our GIoTTO software stack is to face security and privacy concerns from the ground up rather than retrofitting later.

Security in IoT means providing access control mechanisms and policies and being able to enforce them, particularly in the face of the tremendous number of heterogeneous devices. In the GIoTTO stack, we've implemented a robust access control layer to allow flexible security policies to be expressed as well as abstractions to manage the number of rules that must be specified. Done naively, one would need to have as many rules as the number of devices multiplied by the number of users. We leveraged ideas from role-based access control, and mechanisms for grouping users and sensors to reduce the number of rules that need to be specified. Furthermore, we developed a flexible tagging (key-value attributes) system for users and sensors or actuators to allow access control rules to be expressed more concisely and be verified at runtime with minimal performance overhead. We also use industry standard protocols (such as SSL and OAuth) to secure the other layers of the network protocol stack.

MACHINE LEARNING AND DATA ANALYTICS

With the huge amount of data that will be captured and stored in IoT infrastructures like GIoTTO, it's apparent that people won't be able to consume all the data. Most IoT efforts support sensing, simple computation, communication, distribution, and actuation, but not analytics and machine learning. In contrast, we treat all these aspects as core functions of any IoT infrastructure.

At the lowest level (collecting data from sensors embedded in the environment or objects), data analytics can be applied to reduce overall power usage. Instructions sent to sensors to collect data at a particular rate can be configured in real time using power consumption analytics. The configuration must satisfy all current requests for data (including just storing the data), while optimizing for power consumption.

As another example, we're currently working on developing algorithms to identify novel patterns present in sensor data and in people's behaviors (as captured by sensors). Identifying these patterns creates new, higher forms of data from low-level data. We call these components that can capture complex high-level patterns—or even ones that perform simple analyses such as averaging across sensors—*virtual sensors*. Virtual sensors take input from one or more physical or virtual sensors and produce some new output. An IoT infrastructure must be able to support the production of simple and sophisticated virtual sensors.

A third example we've implemented is support for programming by demonstration. Typical end users might not have the technical know-how to build their own virtual sensor, but GIoTTO provides a tool that allows end users to demonstrate a phenomenon that they want their environment to capture, and direct the tool to automatically build a classifier. Let's say a user wants to know whether she left her window open. Without knowing what sensing exists in her environment, she can simply launch the tool and provide examples of the window being open and the window being closed. The tool examines all data being collected during

both situations and identifies the sensors that are most predictive of the window's state. It calculates statistical features on the sensor data and trains a model in near real time. Within seconds, the end user has a virtual sensor that can detect window state.

Much like the rest of GIoTTO, the machine-learning components are intended to be pluggable, allowing a system administrator to select the appropriate components to deploy, whether they're widely available (such as TensorFlow or Azure) or bespoke.

END-USER EXPERIENCES

In addition to allowing end users to build their own virtual sensors, a core tenet of GIoTTO is strong user support for the installation, maintenance, and control of an IoT environment, whether it's a one-room office, a multiroom home, or a large factory. In all cases, the user experience should be simple and seamless.

In addition to supporting virtual sensors, we're also working to support virtual actuators. A virtual actuator can represent multiple actuators where users can specify the action they want to occur, but don't have to specify the object that performs the action. Instead, the infrastructure would select the object based on a set of predefined criteria. Similarly, a virtual actuator could cause collections of actuators to perform an action simultaneously (for example, lights blink and a phone buzzes), where the specification should be as simple as possible for end users.

We're actively working to support end users in programming their IoT environments, including providing a range of visually based programming platforms that would allow users to create their own simple if–then rules, and building applications with more complex logic. In addition to user-created applications, through our living labs at CMU we're supporting a number of scenarios for the campus environment that GIoTTO users can replicate and use at their own sites (such as the coffee example described earlier).

A future goal is making installation as simple as possible, where new objects are automatically added to the IoT environment and communicate seamlessly with GIoTTO through simple discovery protocols. Similarly, for maintenance, built-in data analytics support should identify sensors and actuators that are malfunctioning or need new batteries, and alert users about how to fix them.

We currently have a single installation of GIoTTO on the CMU campus, supporting four living laboratories spread over three different buildings. The living laboratories include two academic research labs, one office, and one public indoor space. We're continuing to develop the GIoTTO infrastructure to address the challenges laid out in this article. The first version of the infrastructure was released in March 2016, and updates will follow quarterly. Our future plans also include adding more academic and industrial partners to the IoT Expedition—we hope you'll consider joining.

YUVRAJ AGARWAL is an assistant professor of computer science in the School of Computer Science at Carnegie Mellon University. Contact him at yuvraj.agarwal@cs.cmu.edu.

ANIND K. DEY is the Charles M. Geschke Professor and director of the Human–Computer Interaction Institute at Carnegie Mellon University. Contact him at anind@cs.cmu.edu.