Synergy: A Smart and Scalable Energy Measurement Platform for Electricity Consumers

Sarah Johnson*, Pearce Ropion†, and Behnam Dezfouli‡
Department of Computer Science and Engineering, Santa Clara University, Santa Clara, USA
*s5johnson@scu.edu, †propion@scu.edu, ‡bdezfouli@scu.edu

***Abstract*—The key to becoming a more sustainable society is first learning to take responsibility for the role we play in energy consumption. Real-time energy usage gives energy consumers a sense of responsibility over what they can do to accomplish a much larger goal for the planet, and practically speaking, what they can do to lower the cost to their wallets. *Synergy* is an energy monitoring and visualization system that enables users to gather information about the energy consumption in a building – small or large – and display that data for the user in real-time. The gathered energy usage data is processed on the edge before being stored in the cloud. The two main benefits of edge processing are issuing electricity hazard warnings immediately and preserving user privacy. In addition to being a scalable solution that intended for use in individual households, commercial offices and city power grids, Synergy is open-source so that it can be implemented more widely. We present the design and implementation of Synergy as well as initial finding based on the data collected by this platform.**

***Index Terms*—Energy Efficiency, Energy Monitoring, Internet of Things (IoT), Modularity, Privacy**

I. Introduction

The world runs on electricity. Whether it be lighting homes, pumping water, powering factories, or even, in some cases, opening a door. However, energy usage is difficult to gauge. Energy bills often get paid without knowledge of how much energy was used and where. For example, many large complexes leave their lights and devices on throughout the night. On-demand electricity is expensive; both to our wallets and our planet. On average, American households spend $112 per month on their electricity bill [1]. Often times, electrical bills can spike due to devices that have been left powered on or malfunctioned, and can go undetected for years. Evidence has shown that monitoring domestic electricity usage plays a large role in reducing consumption [2]. Energy monitoring solutions allow people to be conscientious about how much energy they are using. These solutions are essential in keeping energy costs down for the consumer by revealing noninvasive lifestyle changes to lower energy consumption, such as making a habit of turning off lights or devices when not in use [2]. However, there are a multitude of challenges in implementing universal energy monitoring.

The challenges of energy monitoring stems from a problem with energy management. While monitoring specifies the process of keeping track of the energy usage, management refers to the coordinated efforts to habituate a reduction in energy consumption [3]. At any scale, energy management requires a coordinated effort between all actors. Actors could include the residents of a single household, or the employees within an office building. There must be consensus among the actors to realize lowered energy usage; be that an understanding that employees must turn off their monitors and devices at the end of the workday, or that all lights must be turned off in the home before leaving. Without proper energy management, energy monitoring would have much less meaning. At the same time, without energy monitoring, it would be much more difficult to assess how reductions in energy consumption could be realized.

Beyond the financial gain that individuals and businesses can gain from employing an energy monitoring platform, there is also an ethical duty of stewardship to consider. In 2017, about 34% of the total U.S. energy-related carbon dioxide (CO2) emissions were emitted by the U.S. electric power sector [4]. CO2 emissions contribute to global warming, which has the potential to be extremely disruptive to our global climate [5]. It is essential that CO2 emissions are reduced, especially given that the "impact of carbon emissions persists longer than that of nuclear waste" [6]. If the issue of greenhouse gases is not addressed now, it may not be addressed in time. Energy monitoring can play a role in minimizing greenhouse gases by giving individuals the extra push to reduce their carbon footprint - even if their intentions are to save money. Seeing their usage in real-time may also influence more users to pursue alternative forms of energy - such as solar power - which has a smaller carbon footprint than natural gas or coal, which are the two largest energy sources [7].

Monitoring energy usage is not a new concept. Electricity usage is consistently monitored by power companies in order to charge consumers for the energy that they use. However, there has been very little effort to deliver these monitoring capabilities to the individual. Instead, consumers are charged a lump sum without any distinction as to where that energy went or knowledge of how to reduce its usage. Placing energy management solutions in the hands of consumers enables them to actively institute measures for saving energy. The difficulty of this solution comes when one considers how these solutions could be enabled within households. Over 70% of the residential homes within the US were built prior to 2000 [8], meaning that they simply do not have the infrastructure to adequately equip energy monitoring and management devices without remodeling.

In order to accommodate some of the challenges that come with energy monitoring, many existing solutions have made

978-1-7281-1780-5/19/$31.00 ©2019 IEEE

their products compatible with other technologies in an effort to attract a larger audience. Sense [9] is a monitoring device that connects to a home's electrical panel. The device monitors electrical throughput in order to determine how much electricity is being used by the household at any given time. Sense also provides a cloud-based visualization service accessible via a web-based and smart phone application. Sense allows for precise detection of home activity including when electricity is being used and which device is using it. It can also notify users when it detects unnecessary energy usage from a device. However, precise detection that Sense provides introduces privacy and security concerns with regards to home safety. Data of every electrical action is monitored and immediately uploaded to Sense's cloud servers. If bad actors were to intercept this data stream it could give a general idea of when a user is home or not, leaving homes vulnerable [10], [11]. Constantly uploading data also means that the home's bandwidth is clogged with the hundreds of megabytes that Sense uploads to the cloud every day. Furthermore, the device itself is rated to a maximum of 200 amps meaning it can only be placed in the electrical panel of a residential home. Placing the device in a larger complex, such as an office building would cause it to overload, making the scalability of such a system impossible. Neurio [12] is another system which is installed in the home's electrical panel. Similar to Sense, it provides cloud-based data visualization that requires a constant connection to the internet. Neurio can also be integrated with solar panels mounted on the home's roof and supports submetering of major appliances like electrical vehicles. However, these additional features fall victim to the same drawbacks as Sense; primarily that none of the home's private usage data is stored locally, and the system has not been designed to be used outside of a standard residential home. There are also some alternative energy monitoring systems such as TP-Link's Smart Plugs [13]. These devices can be plugged into an existing power outlet on the wall or on a power strip. They can then monitor energy usage of a single appliance which can be plugged in the front side of the smart plug. Similar to both Sense and Neurio, TP-Link Smart Plugs stream their monitoring usage data to the cloud. Users can then access this visualization data through their smart phone and web applications. Furthermore, the plugs are also smart home enabled, meaning they can connect to existing smart home systems such as Amazon's Alexa and Google Home to be turned on and off remotely. Unfortunately, each smart plug can only monitor a single device at a time and costs $20 each, which can become expensive given how many devices reside in the modern home. This presents issues when it comes to the scalability of the product.

Other existing solutions have attempted to solve problems with measuring energy usage in low power devices [14] and power fluctuations in networking devices [15]. Monitoring in a low power environment involves precise monitoring devices that are unsuitable for consumer scalability. These types of devices would be best suited for monitoring the energy usage of the monitoring devices themselves or other similar sensors. Similarly, networking devices such as load balancers, switches, routers and modems use a lot of energy on a daily basis and these devices are always on and always processing. Although simple monitoring on these devices would be helpful, the real use case comes when attempting to detect security vulnerabilities or cyber-attacks. A denial of service (DoS) attack spams a networking device with hundreds of thousands of packets in order to attempt to block traffic to the services provided. Generally, when a DoS starts, the device's processor picks up, increasing its power usage. Energy Monitoring on networking devices could help detect DoS attacks when they start, instead of finding them hours later when it is too late.

In order to combat the security and scalability issues of existing systems, we have developed *Synergy*, a reliable, scalable, open-source, and privacy-protecting energy measurement tool. The data is generated between various subsystems for processing and storage, and only after initial processing it is periodically uploaded to the cloud. The system has an integrated electricity usage visualization dashboard that displays real-time power usage statistics. Furthermore, the system is able to detect sudden spikes or irregularities in usage, informing users to the possibility of a blown fuse or other energy leakage by enabling quick alert generation. Keeping the system on a local area network (LAN) enables the system to not require a constant internet connection. This allows the system to be scaled for larger complexes by allowing multiple devices to communicate wirelessly and has the added benefit of working in remote areas that have decreased Internet access. For example, using a LAN is much more sensible if a person leaves their stove on and needs an immediate warning. Synergy does not need to upload data to feed it back to the person. Instead, the real-time data is transmitted directly to the user. While there are plenty of ways that already exist to monitor energy usage, most of these methods fall short in security and scalability due to their reliance on cloud computing. By processing on the edge, our system is able to avoid these issues and empower users to take responsibility for their energy consumption more securely and on a much larger scale than is currently possible.

The rest of this paper is organized as follows: Section II presents the platform components and explores the design decisions that were made in the process. Then we study some of the real-world applications of Synergy in section III. Specifically, we show how the system can be used to reduce energy usage and costs, how it could be used in the future of smart grid technology, and how cyber-attacks could be averted by monitoring the energy usage of Internet of Things (IoT) devices like Alexa and Google Home. We conclude the paper in Section IV.

II. Synergy: Design & Implementation

Synergy is a universal energy monitoring platform designed to monitor the AC current of a device or set of devices. The system is composed of three micro-services: (i) AC Current Monitoring, (ii) System Management, and (iii) Web Server/UI.

The micro-services can be split into three groups: front-end, back-end, and hardware-level services. Front-end services

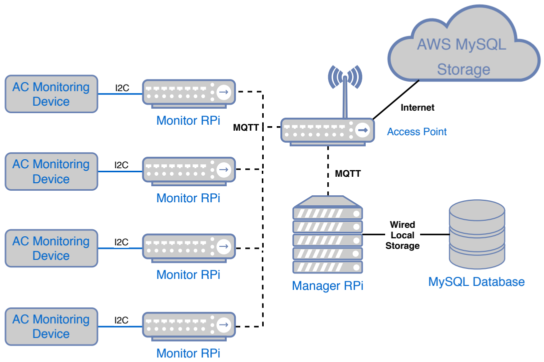

Fig. 1. High-level architecture of Synergy and the connection between AC current monitors, the system manager, and the databases.

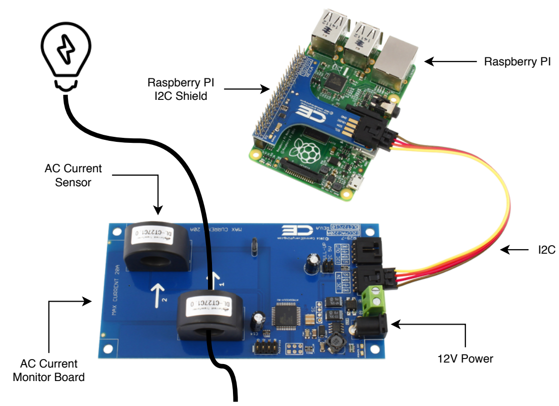

Fig. 2. Hardware implementation of an AC current monitor connected to a Raspberry Pi.

interface the system with the rest of the world. The client driven UI enables the users to interact with the system and visualize the energy usage data. The UI is populated by the web framework, which implements both a web socket and REST APIs. Synergy's back-end services expose hooks to allow two-way communication between the front-end and the individual monitoring devices. The manager is a single hub for all processing done on the data collected. Individual monitoring devices can broadcast their readings to the hub which then stores these readings in the database. Finally the monitoring devices can have up to 12 channels and broadcast real-time current readings to connected systems. Figure 1 maps the relationships between each of these micro-services.

A. AC Current Monitoring

The energy monitoring portion of Synergy is built using one or more AC current monitoring devices. These devices are designed to measure the real-time current in amperes (Amps) up to a maximum of 20 amps. Each device can have between 1 and 12 separate channels to measure the current. An individual channel can measure energy usage as long as the cumulative current passing through the channel does not exceed 20 amps. Therefore, a single channel could measure the energy usage of a single device, or the energy usage of an entire room. Synergy allows the users to set up the AC monitoring device's channels in a variety of configurations. For example, a single 12-channel board could be set up inside an electrical panel to monitor the energy usage of a building. Likewise, the system could be implemented in a power strip fashion. This would allow a user to place a multi-channel board near their desk to measure the energy usage of office appliances, such as a computer or printer, or measure specific areas of their home.

Each channel on an AC current monitor reads the passing current every half a second with 95% accuracy and can sense currents as low as 50 milliamps. Readings are encoded and forwarded via I2C - a serial interface designed for low speed devices such as sensors and microcontrollers [16] - to a hard-wired Raspberry Pi which is then used to relay energy usage information back to the manager through a MQTT channel as seen in figure 2. MQTT is a machine-to-machine (M2M2) connectivity protocol [17] that allows each Raspberry Pi to publish its readings. Synergy's Manager (which will be discussed in the next section) then subscribes to the monitoring device in order to receive the energy readings to store in the database. By using a publish and subscribe model, the monitor can stream the data it generates to the manager without having to be requested. Furthermore, each monitoring device, automatically generates a universally unique identifier (uuid), which is stored on the device and used each time a new reading is published. A python script handles reading and publishing the device's current continuously. The script first establishes a connection with the AC monitoring device, then reads the byte stream into a buffer before publishing the data along with the time it was read and the device's uuid as a stringified JSON object. With each iteration, the previous AC current readings are saved and compared against new readings. If the new readings are different from the previous readings, then the new readings are processed by the manager. Synergy then assumes that if there are no additional readings stored in the database between two timestamps that are more that 1 second apart, then the current did not change during that time. By limiting how many device readings are stored in the database, Synergy can reduce its overall footprint in terms of per-device storage requirements. The byte conversion, parsing, and publishing process for a multi-channel AC current monitor can be found in Algorithm 1.

Synergy monitoring devices are completely independent and self contained. They do not know about the existence of the Synergy manager or any other monitoring devices that have been added to the system. Instead, these devices can be thought of as workers in the system with the manager as their master. Any number of workers can be added to the system, each with the capability of transmitting within the MQTT channel. This means Synergy can be installed in a single residency with multiple monitoring devices placed throughout the home and a large complex such as an office building with a monitoring device placed in every office. It can also be used by power companies to become more knowledgeable about how consumers are using their energy by placing a monitoring device within each user's home.

Algorithm 1: Data collection and publishing

```
previousCurrents = []
while 1 do
    currents = []
    for i ← 0 to numChannels do
        Read current from monitoring sensor using I2C
        currents[i] = current
        if currents[i] ≠ previousCurrents[i] then
            publish to MQTT
    previousCurrents = currents
```

B. Manager

Once the energy usage readings have been converted into currents, they are sent to the manager whose primary role is to store energy usage readings into the database. The manager runs on a Raspberry Pi that is connected to the network. This device is called the *hub*, and all of the connected monitors are called *workers*. In order to receive the data from numerous workers at once, the manager opens MQTT channels which broadcast on the local area network (LAN). Instead of publishing new data to these channels like the workers do, the manager subscribes to any incoming messages of a specific type. The type in this case would be a new current reading. Other message types that the manager can receive include warnings and errors emitted from the workers as well as new connection messages from workers that have just been added to the system.

When the manager receives new readings, the readings are passed through a series of test cases to request feedback about the system in real-time. The test cases can include whether the energy readings of a device are above a certain threshold, or whether a device is using more power than expected at a certain time. Tests can be both user-defined and system-administered. In all cases, when a new energy reading fails a test case more than three times in a row, it is considered an abnormality. When an abnormality is detected, notifications are issued to users about potential errors in the system. Furthermore, Synergy keeps snapshots of a monitoring device's usage throughout the day. If the system detects that a device is consuming more than its average, an alert could be issued saying that the device is either consuming more power than normal or is currently malfunctioning. Synergy also has the ability to generate reports of energy usage over a period of time. These reports contain information including the total amount of energy used and the cost of energy. Synergy also attempts to identify areas that are consuming too much energy and provides recommendations on how to reduce their energy usage.

Once a new reading is past the test cases, it is inserted into the database. Synergy's database keeps track of every change in energy usage of a device for a configurable maximum time period from the date the current was read. After this period has ended, the manager uploads the historical energy usage to connected AWS Aurora MySQL instances for further storage and backup capabilities. The size of these instances depends on how much energy usage data is being generated. On average, Synergy's monitoring devices generate approximately 500 KB per 4 channels per hour. Therefore, a 12 channel device generates approximately 36 MB per day. However, the most recent data is stored locally on a 1 terabyte storage, which would allow a single 12 channel worker to function for over 76 years before an AWS upload is required.

TABLE I
DEFINITION OF USAGES TABLE

Column	Type	Notes
channelID	VARCHAR(36)	-
timestamp	BIGINT(13)	epoch in milliseconds
ch1	DOUBLE(5,3)	current in Amps
...	...	...
ch12	DOUBLE(5,3)	current in Amps

New readings received by the manager not only contain the current value of individual channels on a device, they are also transmitted with some metadata including the time the reading was made and the device id the readings are from. Table I describes the database table and the values it can receive. Before inserting a new reading, the manager checks if the device has previously sent readings, and if not, it generates IDs for all of the channels on the device. Synergy's workers only keep track of their device ID while the currents themselves are transmitted as an array. When adding new channels and their associated IDs to the database, the manager also records their channel number which is the position consisting of a number between 1 and 12, corresponding to the position of that channel on the device.

C. Web Server and UI

Once data from Synergy's energy monitoring devices is available within the database, the web framework becomes active. Synergy's web infrastructure is written in Node.js for its wide scale availability and community support. The framework serves two purposes: to serve the user interface's site and to provide real-time usage data to the user. Its main functionality comes in the form of an open REST API for the UI to request specific datasets. However, it also has the ability to stream results from the database given certain parameters. The web server divides the system into 5 main components: channels/devices, groups, users, notifications, and usages. In order to make the system user-friendly, Synergy allows users to name and edit any component within the system. It also allows for resetting of certain components if, for example, a user would like to reset their system when moving to a new house. Furthermore, users have the option to group any combination of either channels, monitoring devices or other groups into new groups whose combined energy usage can then be viewed.

Viewing the energy usage can be done in two forms: real-time energy usage,and historical energy usage. Real-time energy usage displays charts detailing how much energy the system is currently using. The data is streamed to the UI in real-time so if, for example, while viewing the current energy

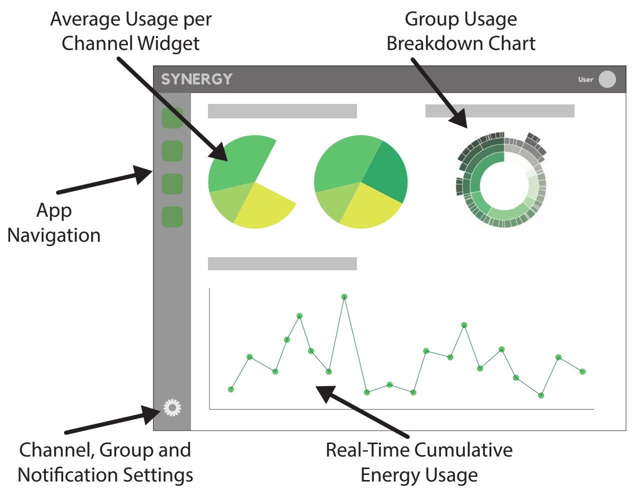

Fig. 3. Synergy UI showing interactive dashboard in which users select what information they want displayed.

Algorithm 3: Group member retrieval

```
function getMembers(group_id, members = [])
    groups = []
    devices = []
    channels = []
    get members of group_id
    foreach item of members do
        if item type is "group" then
            result = getMembers(item_id, members)
            add result to groups array
        else if item type is "device" then
            result = getDevices(item_id)
            add result to devices array
        else if item type is "channel" then
            result = getChannels(item_id)
            add result to channels array
        else
            throw Error
    return groups, devices, channels
```

usage, a user turns on a monitored device, the total energy usage would go up. In order to achieve this real-time feedback, the UI must first communicate with the web server in order to start streaming the correct set of data. To do so, the UI requests that a new socket be opened containing a set of channels, devices, or groups. The web server then creates a new socket and simultaneously pulls the latest results from the database. It will then publish this dataset on the socket which the UI can subscribe to. Finally, the web server can detect when the UI is no longer connected and stop streaming the data when that happens. The pseudo-code for this process can be found in Algorithm 2. The second usage type is historical data in which the web server serves a static snapshot of a specified set of channels, devices, or groups. Grouping data is held in two tables: a groups table which stores information about a group, and a groupings table which stores the members associated with each group. In order to decrease the amount of requests required to retrieve all of a group's direct and indirect members, the web server iterates through all available groups, devices, and channels before returning the dataset to the UI. The pseudo-code of this operation can be found in Algorithm 3.

Algorithm 2: Usage data request and streaming logic

```
function streamUsages()
    receive POST request containing a list of channel IDs
    create a socket
    respond to client with socket ID
    while client is connected do
        foreach channel do
            query DB for usages of channel within the last second
        publish usage data to socket
```

Unfortunately, energy usage in currents is not the easiest to understand as a consumer. Thus, Synergy's web services convert all the current readings to kilowatt hours (kWh). Additionally, users have the option to view usage data in a multitude of formats. Figure 3 depicts how different energy usage charts can be laid out and structured in order to illustrate energy usage data the the user in a multitude of ways to facilitate communication of key usage statistics. For example, users can customize their UI to include a line graph of the current cumulative power usage, bar charts displaying the energy usage at specific intervals, and pie charts which illustrate which channel, device, or group is using what percentage of power. Finally, the system allows users to enter the cost per kWh of power used from their energy bills in order to calculate how much the energy that is being used will cost and predict how much can be saved by reducing the users' energy usage.

III. Applications

Synergy has numerous applications across the board in terms of both environmental and economic impacts. Buildings are responsible for 40% of the energy usage on the planet and up to 30% of the carbon dioxide (CO_2) [18] released. Increased energy usage monitoring could lead to reductions in energy usage and the costs of energy usage, particularly by informing consumers about their energy consumption behaviors and informing producers of patterns in their devices' energy consumption that could be altered for a smaller carbon footprint. Furthermore, it could also influence smart grid systems in an effort to modernize the currently outdated electrical infrastructure.

A. Energy and Cost Reductions

Synergy allows a clear view into the metrics of different energy-demanding appliances. For example, Synergy is able to reveal the difference in performance metrics when altering your lighting settings. As shown in Table II, by dimming the lights of an overhead light source by 10% in an otherwise dark room, the illuminance is only reduced by 4.71% while the energy consumption is reduced by 10%. This results in a net decrease of 10% in terms of the cost of the electricity used (assuming

TABLE II
DECREASE IN LIGHT BRIGHTNESS TO KWH/COST RATIO

Brightness	Lux	Current	kW (1h)	kW (7h/day per month)	Cost
100%	170	0.442	0.05304	11.1384	$1.168
90%	162	0.398	0.04776	10.0296	$1.052
80%	152	0.355	0.04260	8.9460	$0.938
75%	146	0.323	0.03876	8.1396	$0.854
50%	107	0.204	0.02448	5.1408	$0.539
25%	50	0.086	0.01032	2.1672	$0.227

$0.1.488 per kilowatt hour (kWh)) over a 30 day period. The table only measures the consumption of a single room using smart LEDs. However, incandescent or florescent light bulbs would use more energy thus increasing the costs. If this metric were taken into account in a larger setting, such as an apartment (avg. 1200sqft), household (avg. 2600sqft), or office building (avg. 19,000sqft), potentially hundreds of dollars could be saved by just dimming the lights by 10%.

Furthermore many office buildings generally require lights to be left on at night, and without smart lighting, this means that despite vastly different internal lighting demands during the night and day, the same amount of energy is consumed at any time. By switching to smart lighting, managers of these buildings would be able to configure their lights specifically for the different demands of lighting throughout the day. Likewise, since Synergy relies on open protocols, it can be interfaced with different IoT devices such as cameras, Amazon Alexa, and thermostats to detect user activity and adjust light intensity automatically.

For developing countries, Synergy enables the energy distributors and users to monitor electrical appliances and avoid energy loss due to malfunction. By reducing energy waste in these areas, energy access could be offered to a larger number of people to align with The United Nations Development Programme (UNDP) [19].

B. Smart Grid

Synergy's platform is focused on providing more information to the consumer, enabling better communication about how users can save on their energy usage as well as giving utility providers increased insight into how consumers are using the energy provided. This concept has been coined as *Smart Grid*, which consists of a series of new computing and automation technologies that monitor the energy grid in order to provide improvements in energy management [20], energy costs and increased integration of renewable energy sources like solar and wind power [21]. The current power grid in the United States was designed and built over 100 years ago during a time when most residences only needed enough electricity to power a few light bulbs and a radio. "Today, an electricity disruption such as a blackout can have a domino effect – a series of failures that can affect banking, communications, traffic, and security" [21].

The Smart Grid engages in two-way communication between a consumer's home and the energy grid. Synergy is designed to seamlessly integrate with existing systems so its wide breadth of communication technologies coupled with its modularity and scalability make it a prime target for usage in a Smart Grid setting. Energy monitoring by the consumer allows users to see exactly how much power is used and how much it costs. Furthermore, consumers will be more inclined to decrease power consumption when they can see how much the energy is costing them while using it instead of a final sum at the end of the month. Smart Grid technologies also help to significantly reduce the peak demand for energy usage which is a period of time when electricity is expected to be sustained at a higher than average level [22]. Many electricity providers charge consumers based on peak demand pricing. However, the introduction of smart monitoring systems like Synergy allows for dynamic pricing [23] because energy providers can more easily predict and provide for the energy requirements of peak demand [24]. Moreover, individual contributors can better manage their individual energy usage during peak times in order to cut down on their costs.

C. Denial-of-Service Detection

Another possible application of Synergy is denial-of-service detection. DoS attacks attempt to flood a network with an overwhelming amount of traffic with the aim of making the services or resources provided by the network unavailable. Early stage DoS detection is extremely difficult, despite an abundance of research into detection techniques [25]. Energy monitoring introduces new opportunities to DoS detection in IoT devices. Because of the increased load on the device, more power is consumed. While the increase is too small to be noticed on the AC monitor that Synergy currently uses, it is possible that a monitor designed for reading low-power devices could be substituted and accurately detect irregular increases in energy consumption, indicating a DoS attack. This is something we hope to develop in the future.

IV. CONCLUSION

Synergy is an energy monitoring solution that is designed to be scalable for any situation in which energy monitoring could be used to decrease energy usage and energy costs. Although this paper discusses how *Synergy* was implemented using Raspberry Pi board, it could be implemented using any Linux-based hardware components. Furthermore, its system is not locked to any one environment as it utilizes libraries and frameworks that are available across most programming languages. In addition to this, *Synergy* is open-source which allows any user to adapt it to their needs or contribute to improving it. As mentioned previously, *Synergy*'s reliance on open protocols enables it to be integrated with various IoT devices, which is an ability that open-source development could expand upon.

It is also worth noting that *Synergy* can be deployed with any number of monitors, managers, and servers, just like the majority of modern web-based platforms. It can be used in residential homes, but also in larger complexes such as university, campuses, and office buildings. It can also be used by appliance producers to reduce their per device power usage requirements. Likewise, it could be implemented by local utility companies to obtain a better understanding of how electrical energy is being used in their community. There are very few limits to where *Synergy* can be used the effect that it can have on the community that uses it. Despite its passive nature, its use attempts to change behavioural patterns so as to reduce the carbon footprint that is indicative of electrical energy usage across the globe.

ACKNOWLEDGMENT

We would like to acknowledge and thank our financial support from the Engineering Undergraduate Program Senior Design Project Funding at Santa Clara University.

REFERENCES

[1] "Electric Sales, Revenue, and Average Price," report, US Energy Information Administration, 2019.
[2] T. L. Webb, Y. Benn, and B. P. Chang, "Antecedents and consequences of monitoring domestic electricity consumption," *Journal of Environmental Psychology*, vol. 40, pp. 228–238, 2014.
[3] C. Steel, "The Challenges of Energy Management." https://www.linkedin.com/pulse/challenges-energy-management-cameron-steel/, 2017.
[4] US Energy Information Administration, "How much of U.S. Carbon Dioxide Emissions are Associated with Electricity Generation?." https://www.eia.gov/tools/faqs/faq.php?id=77&t=11, 2018.
[5] T. Dietz and E. A. Rosa, "Effects of population and affluence on co2 emissions," *Proceedings of the National Academy of Sciences*, vol. 94, no. 1, pp. 175–179, 1997.
[6] D. W. Keith, "Why capture co2 from the atmosphere?," *Science*, vol. 325, no. 5948, pp. 1654–1655, 2009.
[7] US Energy Information Administration, "Electricity in the United States is Produced with Diverse Energy Sources and Technologies." https://www.eia.gov/energyexplained/index.php?page=electricity_in_the_united_states, 2018.
[8] N. Zhao, "Half of us homes built before 1980," tech. rep., 2018.
[9] "Sense." https://sense.com/, 2019.
[10] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic," *arXiv preprint arXiv:1708.05044*, 2017.
[11] B. Copos, K. Levitt, M. Bishop, and J. Rowe, "Is anybody home? inferring activity from smart home network traffic," in *IEEE Security and Privacy Workshops (SPW)*, pp. 245–251, IEEE, 2016.
[12] "Neurio." https://www.neur.io/energy-monitor/, 2019.
[13] "TP-Link." https://www.kasasmart.com/us/products/smart-plugs/kasa-smart-plug-energy-monitoring-hs110, 2019.
[14] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor networks at scale," in *2007 6th International Symposium on Information Processing in Sensor Networks*, pp. 186–195, IEEE, 2007.
[15] B. Dezfouli, I. Amirtharaj, and C. Li, "EMPIOT: An Energy Measurement Platform for Wireless IoT Devices," *Journal of Network and Computer Applications*, vol. 121, pp. 135–148, 2018.
[16] "Inter-Integrated Circuit (I2C)." https://i2c.info/, 2019.
[17] "Message Queuing Telemetry Transport (MQTT)." http://mqtt.org/, 2019.
[18] M. W. Ahmad, M. Mourshed, D. Mundow, M. Sisinni, and Y. Rezgui, "Building energy metering and environmental monitoring - a state-of-the-art review and directions for future research," *Energy and Buildings*, vol. 120, pp. 85–102, 2016.
[19] Gwnalle Legros, Ines Havet, Nigel Bruce, and Sophie Bonjour, "The Energy Access Situation in Developing Countries: A Review Focusing on the Least Developed Countries and Sub-Saharan Africa," report, UNDP and World Health Organization, 2009.
[20] S. Blumsack and A. Fernandez, "Ready or not, here comes the smart grid!," *Energy*, vol. 37, no. 1, pp. 61–68, 2012.
[21] "What is the Smart Grid." https://www.smartgrid.gov/the_smart_grid/smart_grid.html, 2019.
[22] E. Watch, "Understanding peak load and base load electricity." https://energywatch-inc.com/peak-load-base-load-electricity/, 2017.
[23] "Time Based Rate Programs." https://www.smartgrid.gov/recovery_act/time_based_rate_programs.html, 2019.
[24] N. J. Peck, "Peak power control with an energy management system," tech. rep., Naval Post–Graduate School, Monterey CA, 2013.
[25] B. K. Joshi, N. Joshi, and M. C. Joshi, "Early detection of distributed denial of service attack in era of software-defined network," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–3, IEEE, 2018.