Decentralized Access Control with Distributed Ledgers

Using Blockchain to manage IoT access

Ralph Deters
Computer Science
University of Saskatchewan
Saskatoon Canada
deters@cs.usask.ca

***Abstract*— The Internet of Things (IoT) aims to integrate physical devices (aka "things") on the Internet. Enabling (physical) devices/things to form loosely-coupled connections with each other and Internet services/resources enables new and rich interactions between devices, internet enabled services/resources and users. However, this, in turn, leads to the question of how to manage data, services, and interactions of the physical and cyber components. One possible way of managing the services and data and their interactions is by use of distributed ledgers like Blockchain. This paper presents the concept of using privately distributed ledgers as a means for managing the digital ecosystems of IoT.**

Blockchain Patterns, Blockchain, Access Control, IoT, Fog Computing, Edge Computing, Scripts

I. IoT

IoT aims to integrate physical devices (aka "things") on the Internet. Enabling (physical) devices to form loosely-coupled connections with each other and Internet services/resources allows new and richer interactions between devices, internet enabled services/resources and users. The idea of enhancing the capabilities of physical devices by connecting them to remotely hosted software components is also at the center of the cyber-physical system (CPS) paradigm. However, unlike IoT that supports loose-coupling between physical and cyber components, CPS favors " … tight conjoining of and coordination between computational and physical resources. …" [1].

In their IoT review, Gubbi et al. [2] differentiate between a thing-centric and cloud-centric view. The thing-centric view centers on the enhancement of a thing and rich user experiences when engaging it. Smart objects [3] or enchanted devices [4] are the most prominent examples in this category.

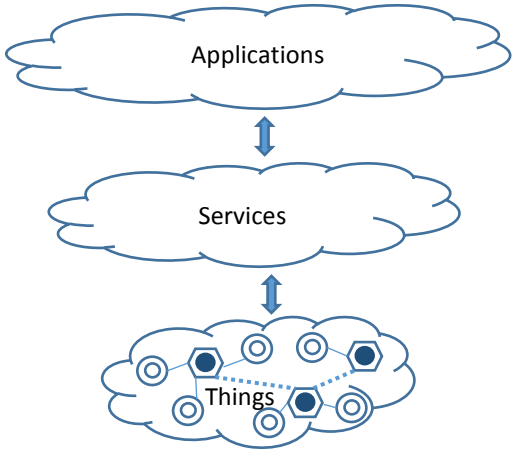

Figure 1. Three layers of cloud-centric IoT[1].

The cloud-centric approach [2,5,6] moves the focus away from the thing towards the services and applications that process large data streams. This view is primarily concerned with the requirement to scale e,g, handle/manage large numbers of connected devices. This model/paradigm implies three core layers. The things layer compartmentalizes the "network(s) of things" [2]. The services layer hosts all core IoT services, e.g., data storage, analytics, and storage.The applications layer is the host for applications like surveillance, monitoring, managing. The advantage of this design is the utilization of cloud computing for the higher IoT functions. When focussing on IoT "big data" scenarios, e.g., smart city, this approach is favorable. However, this approach has drawbacks namely bandwidth, latency, and weak interactions. Since all data is processed in the cloud, the upload requires's significant bandwidth. By placing the computation several network hops away from the physical devices, there is a noticeable latency which in turn is a disincentive for engaging the devices. These are of course well-

known issues of a cloud-centric solution [7], and that can be solved by moving the computation closer to the edge of the IoT system, e.g., via fog-computing [8,9] or edge-computing [10]. However, moving third-party computation into vehicles adds new challenges. While it is safe to assume that vehicles will continue to have increased computational and network resources that can be shared, the issue of multi-tenancy, e.g., multiple parties sharing the resources introduces novel challenges.
This paper focusses on the use of distributed ledger technology (aka Blockchain technology) as means to deal with the access management issues in IoT. The remainder of the paper is structured as follows. Section two discusses the multi-tenancy issues followed by a brief access control review section. Blockchain and blockchain design patterns are discussed in sections 4 & 5. This is followed by an evaluation in section 6 and an outlook and sumary in section 7.

II. Multi-Tenancy

Multi-tenancy [11] refers to an architecture that supports multiple user groups (tenants) to share one or more applications or services. To support the logical separation of tenants sharing applications or services, they must operate within different instances/contexts. Multi-tenancy has been extensively explored within the context of data [12] and cloud-hosted services [13,14,15]. Cherrier et al. [16] identified control flow, access rights [17,18,19] and different settings for actuators as critical challenges for multi-tenancy in IoT. Software-defined networking (SDN) [20, 21] is a management concept that centers on using abstraction to enable the decoupling the control plane (determine destinations of traffic) and data plane (forwarding traffic). Adopting this concept in the IoT space has led to the rise of Software-defined IoT (SD-IoT) [22]. SD-IoT uses abstraction to simplify provisioning and customization of its components. Network Function Virtualization (NFV) [23] goes beyond SDN by focussing on the virtualization of all elements resulting in the ability to define customized virtual networks. Virtualization is used within IoT, e.g., virtual sensors [24,25,26], but these approaches focus on combining or abstracting individual components not defining virtual IoT systems. It is essential to recognize the costs regarding required bandwidth and processing power needed to support the overhead introduced by this approach. Similar to a CPS, physical and virtualized components must be able to communicate. This is relatively easy when resource-rich single board computers (e.g., Raspberry Pi) or computer on a module like the Intel Edison are used. These are de facto Linux systems that have enough CPU, Ram and networking capabilities to stay connected with their virtual twins. However, these resource-rich platforms are relatively expensive and more importantly not low-energy solutions. This, in turn, limits their deployment in IoT. When using low-energy System on a Chip (SoC) IoT platforms things change. The TI CC2541 that is designed to run on a single coin cell battery for years (depending on usage scenario) and the Nordic nRF52832 are good examples of this class of IoT platforms. While these single-chip microcontrollers offer for example 32-bit ARM Cortex processors, they do not provide the ability to run multiple programs or even a single multi-threaded executable. Consequently, the virtualization approach is not useful for this growing class of IoT nodes.

Figure 2&3. Intel Edison on custom Arduino Board

However, it is important to note that these platforms are capable of hosting a single program that can monitor inbound low-energy connections, send data via these connections and of course interact simple sensors and actuators.

Figure 4. TI CC2541 Keyfob

A particularly exciting aspect of these platforms is their ability to host interpreters like Javascript (Espruino, http://www.espruino.com/). Rather than virtualizing IoT nodes, it becomes possible to allow third parties to push scripts onto the nodes.
This, of course, offers a radically different approach on multi-tenancy. Rather than providing a costly virtualization, scripts can be executed directly. Sandboxing is achieved by simply limiting the capabilities of the underlying interpreter.

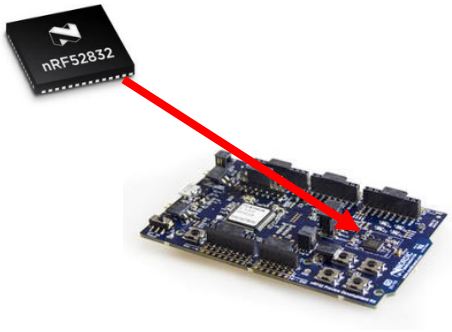

Figure 5. Nordic nRF52832 on development board

By representing the IoT nodes as RESTful web services [27], it becomes possible to not only access them via Robinson's [28] CRUD pattern [29] but also move towards Eherenkartz's computational Rest (http://www.erenkrantz.com/CREST/) in which computational expressions are exchanged. Naturally, this introduces the need for a robust access control.

III. Access Control

The two classical approaches for access control are MAC and DAC.The mandatory access control (MAC) policy grants access based on subjects and objects, which will be assigned security labels. Sandhu et al. combined and extended these two approaches and introduced 2000 the now dominant Role-based access control (RBAC) [31, 32,33,34,35,36]. RBAC uses the data abstraction concept. Instead of using the default operating system's permissions such as read, write, and execute, data abstraction allows the definition of abstract permissions [37]. Attribute-Based Access Control (ABAC) [26,27,28,29,30] is an extension of RBAC. However, inter-organizational access control remains a challenge with RBAC and ABAC due to their more or less centralized design.

Particularly in the context of access control within IoT, where multiple cooperative parties "own" components the centralized access control tends to be difficult to achieve. Blockchain has emerged in recent years as a fully decentralized alternative that seems well suited for the IoT space.

IV. Distributed Ledger Technology (aka Blockchain)

A blockchain is a decentralized ledger that contains connected blocks of transactions. The fundamental concept behind the blockchain is that tamper-proof storage of approved transactions. Valid/verified transaction are stored in the form of blocks that are linked to each other.

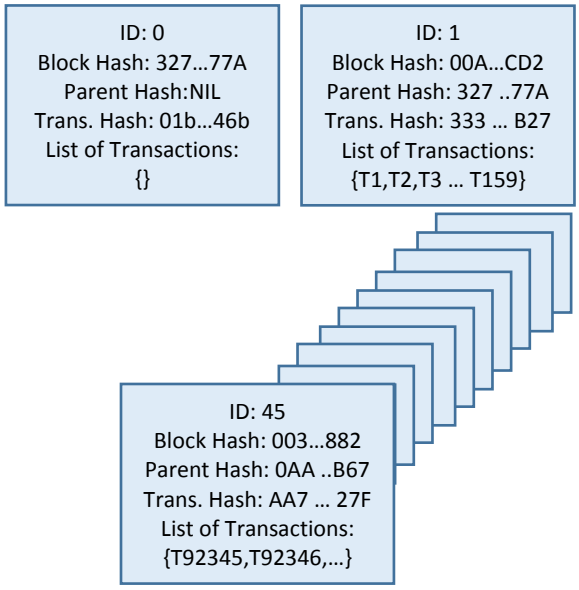

Figure 6. Sample Blockchain.

Upon creation of a new block, the hash value of the preceding block is entered. Once a new block is formed, any changes to a previous block would result in different hash code and would thus be immediately visible to all participants in the blockchain. Consequently, blockchains are considered tamper-proof distributed transaction ledgers. Originally designed as the distributed transaction ledger for BitCoin, the concept has spread and is now also making its way into enterprise software (e.g., Microsoft Azure's Coco).

An excellent example of blockchain technology is IBM's ADEPT system [42] that uses IBM's Bluemix platform. ADEPT can store the configuration of IoT devices and as a mechanism for pushing code onto devices.

However, as demonstrated by Verizon, Blockchain technology can also be used to trade access keys and therefore be used to build a decentralized and fully distributed infrastructure for access control. In this model, the owner of a resource, can securely transfer/modify or revoke one or more access key to another party via the blockchain.

It is imperative to note that Blockchain technology is not a silver bullet. Blockchain assumes that each node needs to see all transactions and that all nodes need to store a full copy of the data. Obviously, this approach does not scale. Distributed Ledger Technology, a superset of Blockchain Technology, offers more scalable solutions. First, a permission-based or private DLT is needed since identity needs to be established. Second, only those nodes that have a legitimate interest in transactions should be informed and consulted.

V. Design-Patterns for Distributed Ledger Technology (DLT) and Blockchains

An often overlooked issue in the deployment of DLT/Blockchains is the underlying design-pattern. In the context of access control for IoT two basic patterns can be identified:

- Announcement
  The announcement pattern is used to make a tuple consisting of user-id and access privilege known to others. The most common form is by use of granting a unique access key to an entity. This can be done by having the owner of a resource submit a transaction to the DLT/Blockchain indicating that entity X is given the access key Y. If the name of the entity and the access key are visible to all nodes of the Blockchain, a public announcement is made. However, entity name and access key may be encrypted via a secret known to the owner of the resource and the resource manager.

- Contract
  The contract (Smart Contract / Transaction Function) focusses on the use of a deterministic finite-state machine. Again the code can be visible to all nodes on the network (public smart contract / public transaction function) or encrypted so that only a selected group can execute the code. Requests from an entity to access a resource are now evaluated by the contract/function which in turn allows for more advanced access control, e.g., entity X may only do five reads, or 2 write operations.

VI. Evaluation

To evaluate the feasibility of using blockchain to govern the distribution of scripts onto low-power IoT components a basic hub-spoke IoT system was used. As shown in figure 7 Raspberry PI 3 are used as middleware components. The top layer devices are acting as entry points and the lower level devices as masters for the low-power IoT devices (Nordic nRF52832 on a development board). Requests are sent to the entry point devices, that in turn forward them to the raspberry PI's that are connected directly via BLE to the endpoints. Please note that the DLT/Blockchain MultiChain is used in the experiments.

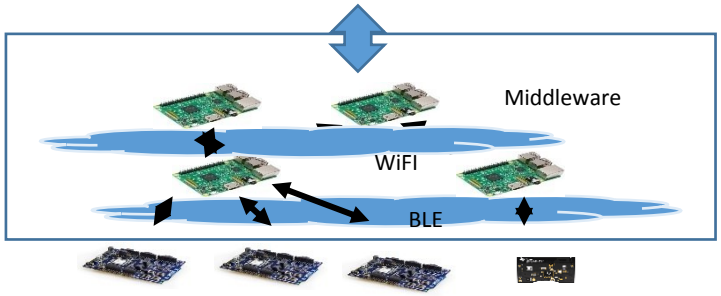

Figure 7. Connecting IoT endpoints with Raspberry PIs

A. Performance of GET (Reads)

The first set of tests focuses on external clients retrieving state, e.g., reading the temperature. Please note that we represent concurrent clients by threads. Two threads refer to threads in a load generator issuing GET requests at the specified intervals. Different colors refer to different threads. Each setting was run three times. The endpoints (e.g., components in a vehicle or roadside installation) host JavaScript code that is handling the read/writes to/from the underlying sensors and actuators. Since GET requests are cachable, these experiments show the performance of the cache that is hosted in the top layers of the Raspberry PIs. The cache is updated every second by writes that emanate from the IoT endpoints.

Figures 7 – 9 show that at 1000 ms arrival rates up to 5 concurrent clients do not impact the middleware. However, as the number of concurrent clients and the arrival rate is increased (more message in shorter time periods) we can see a dramatic decline in the middleware performance. Since all request are sent to the same Raspberry PI, we suggest using a basic load balancer to distribute the loads across multiple machines. The key factor is primarily the number of messages a single Raspberry has to process. Apparently choosing a more powerful compute node to process the requests would delay the need of a load balancer.

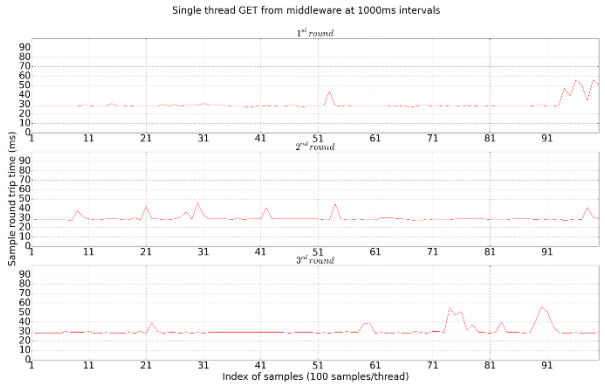

Figure 8. One Client sending 100 GET requests (1 sec delay)

Figure 9. Two Clients sending 100 GET requests (1 sec delay)

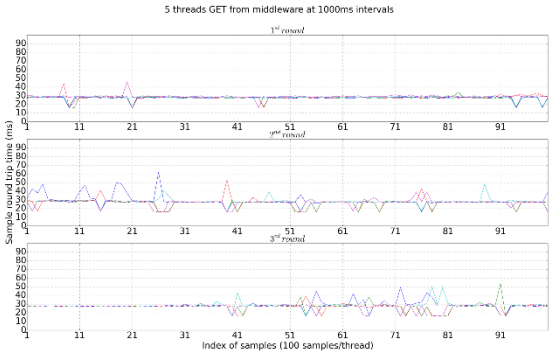

Figure 10. Five Clients sending 100 GET requests (1 sec delay)

B. Performance of POST (Writes)

The second set of tests focuses on external clients sending data (changing settings on the sensor) to the IoT endpoints. POST messages cannot be cached, and the request must be sent from the first layer of Raspberry PIs to the second and finally to the endpoint. Given that more machines are involved in processing the POST request it is not surprising that latency increases. Please note that all POST requests were sent to the same IoT endpoint which explains the dramatic decline in performance at higher loads.

Figure 11. 10 Clients sending 100 GET requests (125 ms delay)

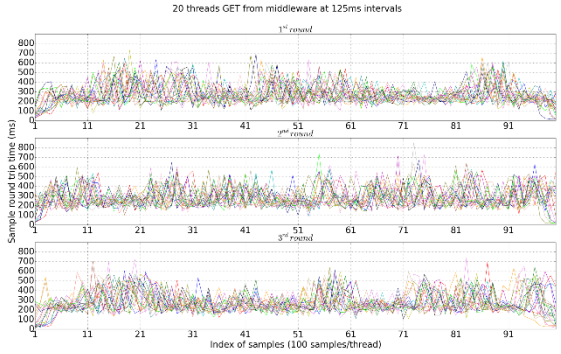

Figure 12. 20 Clients sending 100 GET requests (125 ms delay)

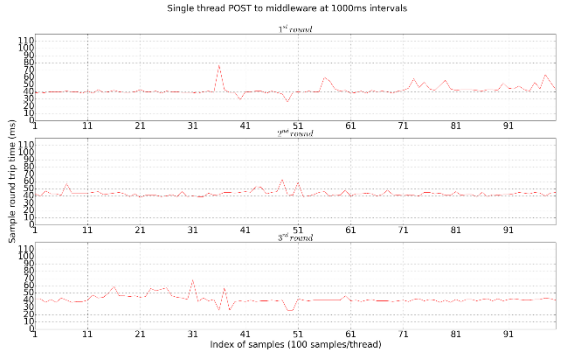

Figure 13. One Client sending 100 POST requests (1 sec delay)

C. Performance of Raspberry Pi hubs

To evaluate the delay caused by the devices, 100 write and 100 read requests were sent to an endpoint. As can be seen in figure 16, changing the state of the endpoint requires around 200 ms while reading from the IoT devices requires on average only 140 ms.

Figure 14. Two Clients sending 100 POST requests (1 sec delay)

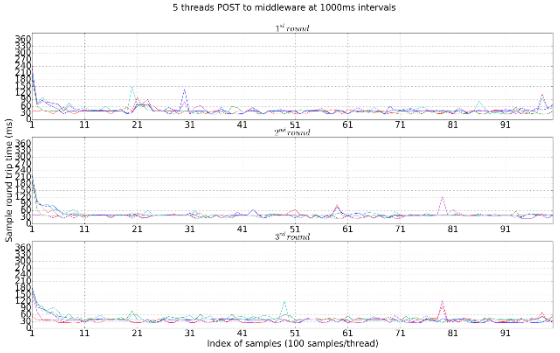

Figure 15. Five Clients sending 100 POST requests (1 sec delay)

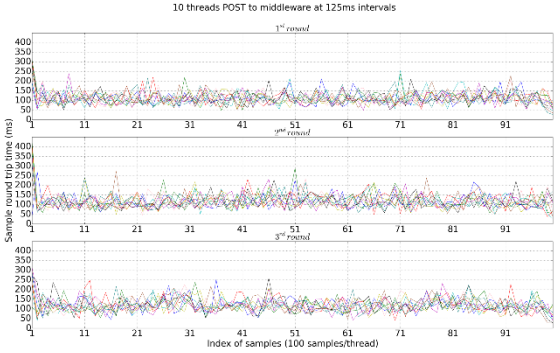

Figure 16. 10 Clients sending 100 POST requests (125 ms delay)

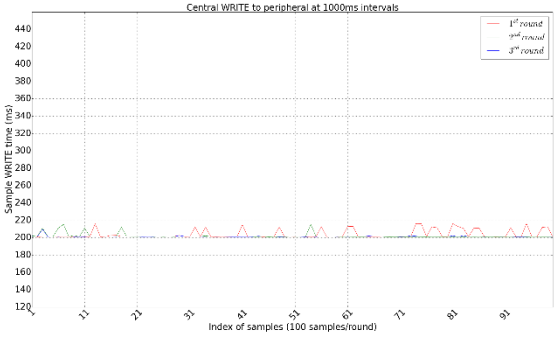

Figure 17. 100 sequential Writes to IoT endpoint

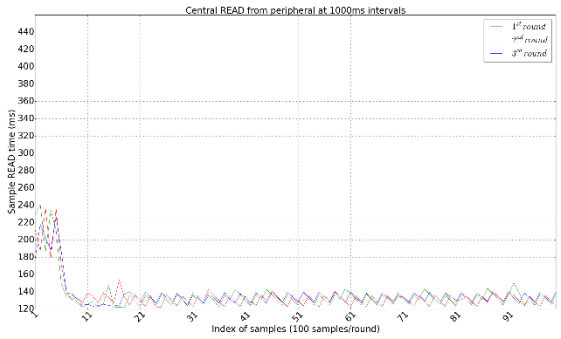

Figure 18. 100 sequential Reads from IoT endpoint

D. Performance of Blockchain in high throughput environment

To test the performance of the blockchain that controls the access privileges, e.g., if sending a JavaScript file is acceptable if a request can be served etc. we used two scenarios.To simulate high-speed connections we used wired connections.

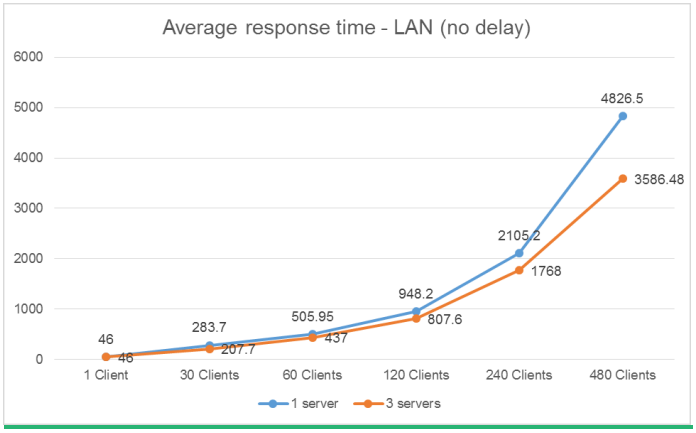

Figure 19. Average response time with simulated clients

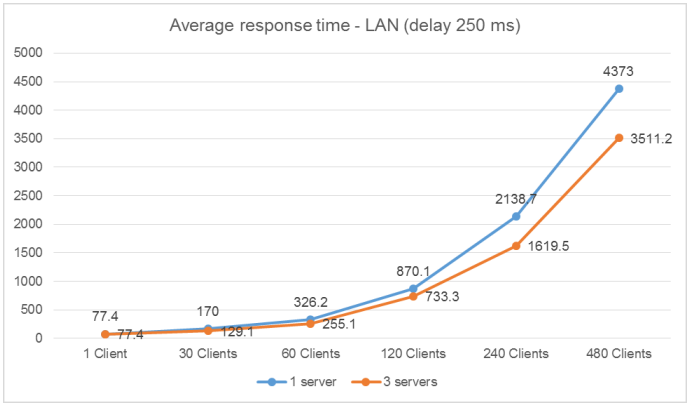

Figure 20. Average response time with simulated clients (250 ms delay)

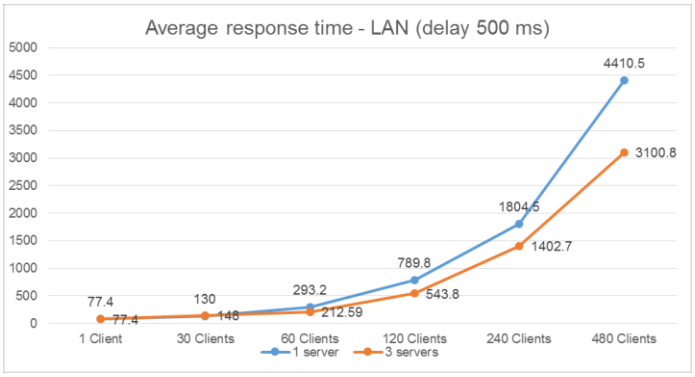

Figure 21. Average response time with simulated clients (500 ms delay)

E. Performance of Blockchain in Amazon EC2 cloud

Finally, the experiments are repeated in the Amazon EC2 cloud to test the effects of high-performance computing environments and high latency.

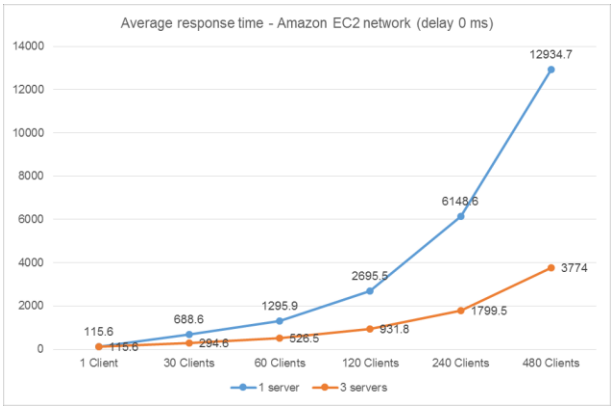

Figure 22. Average response time with simulated clients (no delay)

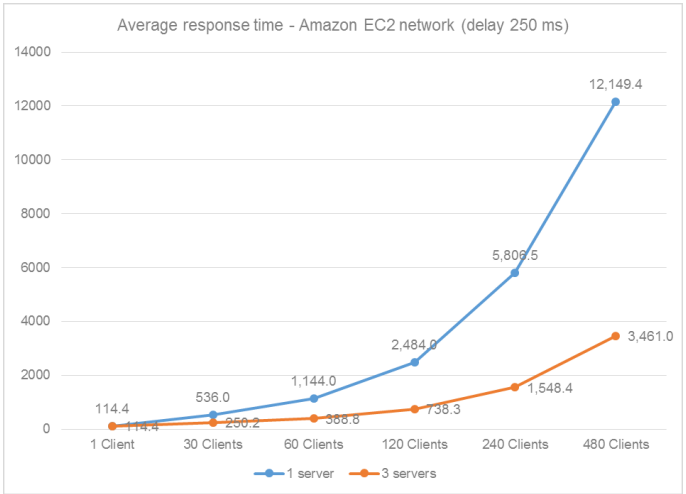

Figure 23. Average response time with simulated clients (250 ms delay)

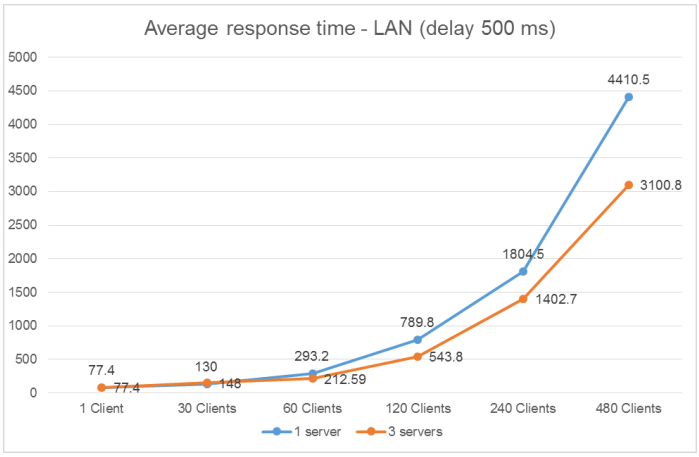

Figure 24. Average response time with simulated clients (500 ms delay)

As can be seen in figures 22-24, the actual workload on the blockchain nodes is minimal, and thus the added latency erases any possible gains of the cloud.

VII. Summary & Outlook

This paper focusses on combining two techniques to support multi-tenancy within IoT edge-computing environments. By pushing script engines onto nodes and allowing third parties to push code onto these nodes a very useful way of sharing low-energy nodes is possible. To overcome the oblivious security challenges we deployed a blockchain for access control. Treating access tokens as digital assets and exchanging them via a blockchain is a practical approach to controlling the distribution of scripts onto low-energy components. Future work will focus on the enhancing the reconfigurability of the Espruino platform, e.g., controlling the API that a given JS program can execute.

References

[1] NSF, Cyber-physical systems (CPS), 2010, https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm

[2] Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future Generation Computer Systems* 29, no. 7 (2013): 1645-1660.

[3] Sanchez, Tomas, D. C. Ranasinghe, Mark Harrison, and Duncan McFarlane. "Adding sense to the internet of things—an architecture framework for smart object systems." *Pers Ubiquitous Comput* 16, no. 3 (2012): 291-308.

[4] Rose, David. Enchanted objects: Design, human desire, and the Internet of things. Simon and Schuster, 2014.

[5] Doukas, Charalampos, and Ilias Maglogiannis. "Bringing IoT and cloud computing towards pervasive healthcare." In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pp. 922-926. IEEE, 2012.

[6] Biswas, Abdur Rahim, and Raffaele Giaffreda. "IoT and cloud convergence: Opportunities and challenges." In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp. 375-376. IEEE, 2014.

[7] Bonomi, Flavio, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. "Fog computing: A platform for internet of things and analytics." In *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 169-186. Springer International Publishing, 2014.

[8] Bonomi, Flavio, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. "Fog computing: A platform for internet of things and analytics." In *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 169-186. Springer International Publishing, 2014.

[9] Vaquero, Luis M., and Luis Rodero-Merino. "Finding your way in the fog: Towards a comprehensive definition of fog computing." *ACM SIGCOMM Computer Communication Review* 44, no. 5 (2014): 27-32.

[10] Grieco, Raffaella, Delfina Malandrino, and Vittorio Scarano. "SEcS: scalable edge-computing services." In *Proceedings of the 2005 ACM symposium on Applied computing*, pp. 1709-1713. ACM, 2005.

[11] Bezemer, Cor-Paul, Andy Zaidman, Bart Platzbeecker, Toine Hurkmans, and Aad'T. Hart. "Enabling multi-tenancy: An industrial experience report." In *Software Maintenance (ICSM), 2010 IEEE International Conference on*, pp. 1-8. IEEE, 2010.

[12] Jacobs, Dean, and Stefan Aulbach. "Ruminations on Multi-Tenant Databases." In *BTW*, vol. 103, pp. 514-521. 2007.

[13] Computing, Cloud. "Toward a multi-tenancy authorization system for cloud services." (2010).

[14] Guo, Chang Jie, Wei Sun, Ying Huang, Zhi Hu Wang, and Bo Gao. "A framework for native multi-tenancy application development and management." In *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pp. 551-558. IEEE, 2007.

[15] Mietzner, Ralph, Tobias Unger, Robert Titze, and Frank Leymann. "Combining different multi-tenancy patterns in service-oriented applications." In *Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International*, pp. 131-140. IEEE, 2009.

[16] Cherrier, Sylvain, Zahra Movahedi, and Yacine M. Ghamri-Doudane. "Multi-tenancy in decentralised IoT." In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pp. 256-261. IEEE, 2015.

[17] Bonomi, Flavio, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. "Fog computing: A platform for internet of things and analytics." In *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 169-186. Springer International Publishing, 2014.

[18] Xu, Xun. "From cloud computing to cloud manufacturing." *Robotics and computer-integrated manufacturing* 28, no. 1 (2012): 75-86.

[19] Botta, Alessio, Walter De Donato, Valerio Persico, and Antonio Pescapé. "On the integration of cloud computing and internet of things." In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pp. 23-30. IEEE, 2014.

[20] Nunes, Bruno Astuto A., Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. "A survey of software-defined networking: Past, present, and future of programmable networks." *IEEE Communications Surveys & Tutorials* 16, no. 3 (2014): 1617-1634.

[21] Kirkpatrick, Keith. "Software-defined networking." *Communications of the ACM* 56, no. 9 (2013): 16-19.

[22] Nastic, Stefan, Sanjin Sehic, Duc-Hung Le, Hong-Linh Truong, and Schahram Dustdar. "Provisioning software-defined iot cloud systems." In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pp. 288-295. IEEE, 2014.

[23] Chowdhury, NM Mosharaf Kabir, and Raouf Boutaba. "A survey of network virtualization." *Computer Networks* 54, no. 5 (2010): 862-876.

[24] Alam, Sarfraz, Mohammad MR Chowdhury, and Josef Noll. "Senaas: An event-driven sensor virtualization approach for internet of things cloud." In *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on*, pp. 1-6. IEEE, 2010.

[25] M. Samaniego, R. Deters: Hosting Virtual IoT Resources on Edge-Hosts with Blockchain, IEEE CIT 2016, 4 pages.

[26] M. Samaniego, R. Deters: Using Blockchain to push Software-Defined IoT Components onto Edge Hosts, BDAW 2016, 8 pages.

[27] Fielding R.: "Architectural Styles and the Design of Network-based Software Architectures", Dissertation University of Irvine, 2000

[28] Robinson, L.: "Richardson Maturity Model"]http://martinfowler.com/articles/richardsonMaturityModel.hml

[29] CRUD: "Create Read, Update and Delete", http://en.wikipedia.org/wiki/Create,_read,_update_and_delete

[30] Bell, D.E., La Padula, L.J.: Secure computer system: Unified exposition and multics interpretation. (1976)

[31] Crispo, B., Sivasubramanian, S., Mazzoleni, P., Bertino, E.: P-hera: Scalable fine-grained access control for p2p infrastructures. In: Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on. pp. 585–591 (2005)

[32] Sandhu, R., Ferraiolo, D., Kuhn, R.: The NIST model for role-based access control: towards a unified standard. In: ACM workshop on Role-based access control. pp. 1–11 (2000)

[33] Baldwin, R.W.: Naming and grouping privileges to simplify security management in large databases. In: Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on. pp. 116–132 (1990)

[34] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E., Sandhu, R.: Role-Based Access Control Models. IEEE Comput. 29, 38–47 (1996)

[35] Park, J.S., Sandhu, R., Ahn, G.-J.: Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4, 37–71 (2001). doi:10.1145/383775.383777

[36] Chen, L., Crampton, J.: Inter - domain Role Mapping and Least Privilege. (2007). doi:10.1145/1266840.1266866

[37] Attribute Based Access Control (ABAC) Overview, http://csrc.nist.gov/projects/abac/index.html

[38] Hu, V.C., Ferraiolo, D., Kuhn, R., Friedman, A.R., Lang, A.J., Cogdell, M.M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., Schnitzer Booz, A., Hamilton, A., Cybersecurity, S.: Draft Special Publication 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations. (2013)

[39] Sun, L., Wang, H.: A Purpose Based Usage Access Control Model. (2010)

[40] Ardagna, C.A., De Capitani Di Vimercati, S., Neven, G., Paraboschi, S., Preiss, F.-S., Samarati, P., Verdicchio, M.: Enabling Privacy-Preserving Credential-Based Access Control with XACML and SAML. (2010)

[41] DoD, U.S.: Department of defense trusted computer system evaluation criteria (orange book). (1985)

[42] https://www-935.ibm.com/services/multimedia/GBE03662USEN.pdf