Demo Abstract: Playground, A Safe Building Operating System

Xiaohan Fu*, Yihao Liu†, Jason Koh‡, Dezhi Hong§, Rajesh Gupta*, and Gabe Fierro¶
*University of California San Diego, USA, {xhfu, rgupta}@ucsd.edu
†Nanyang Technology University, Singapore, yihao002@e.ntu.edu.sg
‡Mapped, USA, jason@mapped.com
§Amazon, USA, hondezhi@amazon.com
¶Colorado School of Mines, USA, gtfierro@mines.edu

***Abstract*—This is an accompanying demo of a main article with the same title co-presented at ICCPS24 [1]. An accompanying poster is also prepared to provide context during the demo.**

Building operating systems (BOS), similar to classical OS, provide essential services to applications running on commercial buildings. The current state-of-the-art requires applications to be *trusted* and *carefully* monitored due to a lack of authorization, access control, and execution isolation mechanisms that could handle the complexity and scale of modern buildings in a *usable* manner. This impedes the adoption of "smart" applications that can enhance energy efficiency, occupant health, comfort, and productivity.

To fill this gap, we develop an operating system abstraction for smart buildings, PlayGround, that incorporates a structured semantic representation of the building to inform the safe, multi-tenant execution of untrusted applications. We use the semantic representation to implement (a) a novel graph-based capability mechanism for fine-grained and expressive access control management, and (b) a resource isolation mechanism with preemptive interventions and passive telemetry-based live resource monitoring. Our proposed solution ensures the practicality and long-term sustainability of adopting *untrusted* user-facing building applications with a focus on maintenance and management labor costs. In this demo, we demonstrate the user interaction with PlayGround with several real applications in a real building.

***Index Terms*—Brick, building, isolation, capability**

I. Introduction

Buildings play a vital role in the majority of human activities. An average American spends 90% of their time indoors [2]. A survey from 2022 highlighted that buildings are responsible for 40% of the total energy usage in the U.S. and contribute to 20% of the worldwide energy consumption, a figure that is increasing by 2% annually [3]. This situation underscores the urgent need to lower buildings' energy consumption while ensuring their occupants' comfort and productivity. The advent of the Internet of Things has transformed buildings into interconnected digital entities, presenting new avenues to meet these challenges through the implementation of intelligent, data-driven "applications."

Applications need to run somewhere. Existing Building Management Systems (BMS), which offer basic control and alarms, are proprietary and hard to program. Over ten years of research into building operating systems [4], [5], [6], application runtimes [7], [8], [9], and applications [10], [11], [12], [13] themselves has shown the need for programming abstractions to simplify the use of BMS and the complex subsystems within buildings. This is challenging due to the unique and incompatible mix of equipment and software in each building, leading to the development of standardized digital representations such as Brick [14] to improve programmability. However, the adoption of smart building applications is slow due to safety concerns. Building managers are cautious about allowing untrusted applications that could affect building safety or comfort, and the lack of clear application permissions complicates this more. We therefore propose two safety properties that a building application runtime should provide: 1. (principle of least privilege) ensure that applications interacting with digital and physical building resources only access what is necessary, 2. (resource isolation) enforce that the impact of applications on building resources are constrained within bounds set by the building managers, including potentially indirectly affected ones such as temperature, energy, and peak power due to the interconnected nature of building systems.

In light of this, we propose PlayGround, a "safe" building operating system (BOS) abstraction that enables the execution of *untrusted, multi-tenant* applications in modern buildings. PlayGround aims to encourage innovation and exploration of how modern building applications can provide value to occupants, managers, and other stakeholders with merely reasonable manual effort required to deploy them *safely*. We integrate a semantic representation of the building — Brick — to imbue our OS services with detailed knowledge of the building. We introduce two novel mechanisms in PlayGround, including 1) a graph-based capability mechanism for access controls; 2) a resource isolation mechanism with support of both pre-action interventions and live inspections.

This work is partly supported by the National Science Foundation under Award Number 1947050.

979-8-3503-6927-4/24/$31.00 ©2024 IEEE
DOI 10.1109/ICCPS61052.2024.00031

II. Demo

A. Demo Description: Design of Playground

PlayGround features a conceptual separation of user/kernel space analogous to classical OS. Our kernel design has five major components: 1) Resource Proxy, 2) Brick Oracle, 3) Permission Manager, 4) Resource Regulator, 5) App Steward. The Resource Proxy provides the hardware abstraction and handles actual access to various cyber-physical building resources such as sensors, actuators, data storage, and alarms. The Brick Oracle contains a queryable Brick model representing a semantic, graph-based representation of the building's structure, subsystems, and I/O points. The Permission Manager is an authorization service that manages and enforces the access control (capability) policies for various entities in the system through a novel graph-based mechanism. The Resource Regulator enforces two forms of resource isolation: 1) per-action value guards inspecting application writes, and 2) live resource tracing mechanisms tracking consumption constantly. Finally, the App Steward manages the life cycle of each app instance from registration to termination.

Applications are separated from the kernel services through a RESTful API service provided by the system. Some service endpoints are privileged (*e.g.,* configurations on these kernel components) and can only be performed by privileged users *i.e.,* building managers; all others require authorization by the Permission Manager.

Find more details of PlayGround in our main article [1].

B. User Experience

To demonstrate how each subcomponent of PlayGround works, we will set up several building applications on a real building. Participants may first play with these applications through the Web UI of PlayGround, just as normal building residents, to understand each application's functionality and how these applications make an impact on real buildings. Second, we will let participants experience the role of a building manager (with limited privilege). Participants may go through the process of designing capability policies, managing building applications, configuring validator mappings, writing new regulating policies and validators, updating the Brick representation, etc. Through this experience, participants shall intuitively understand how the Brick representation of buildings and our proposed mechanisms help make the building management process expressive and simple.

C. Demo Requirements

Due to the software nature of PlayGround, no special hardware is required. A monitor with fair resolution could help participants interact with the Web UI of PlayGround.

III. Conclusion

In this demo, we showcase how our novel dynamic and expressive access control and resource isolation mechanisms in PlayGround could significantly decrease the manual effort required to deploy building applications safely and securely. We envision that building managers with little or no programming background can easily adopt and maintain BOSes following our design.

References

[1] X. Fu, Y. Liu, J. Koh, D. Hong, R. Gupta, and G. Fierrom, "Playground: A safe building operating system," in *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, 2024.

[2] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann, "The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants," *Journal of Exposure Science & Environmental Epidemiology*, vol. 11, no. 3, pp. 231–252, 2001.

[3] U. Energy Information Administration (EIA), "Global energy consumption driven by more electricity in residential, commercial buildings — eia.gov." https://www.eia.gov/todayinenergy/detail.php?id=41753, 2023.

[4] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. Culler, "{BOSS}: Building Operating System Services," pp. 443–457, 2013.

[5] T. Weng, A. Nwokafor, and Y. Agarwal, "BuildingDepot 2.0: An Integrated Management System for Building Analysis and Control," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, (Roma Italy), pp. 1–8, ACM, Nov. 2013.

[6] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and P. Bahl, "An Operating System for the Home," pp. 337–352, 2012.

[7] G. Fierro, M. Pritoni, M. Abdelbaky, D. Lengyel, J. Leyden, A. Prakash, P. Gupta, P. Raftery, T. Peffer, G. Thomson, and D. E. Culler, "Mortar: An open testbed for portable building analytics," *ACM Trans. Sen. Netw.*, vol. 16, dec 2019.

[8] F. He, Y. Deng, Y. Xu, C. Xu, D. Hong, and D. Wang, "Energon: A data acquisition system for portable building analytics," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, e-Energy '21, (New York, NY, USA), p. 15–26, Association for Computing Machinery, 2021.

[9] A. Krioukov, G. Fierro, N. Kitaev, and D. Culler, "Building application stack (bas)," in *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, (New York, NY, USA), p. 72–79, Association for Computing Machinery, 2012.

[10] X. Fu, J. Koh, F. Fraternali, D. Hong, and R. Gupta, "Zonal air handling in commercial buildings," in *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '20, (New York, NY, USA), p. 302–303, Association for Computing Machinery, 2020.

[11] Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng, "Duty-cycling buildings aggressively: The next frontier in HVAC control," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 246–257, Apr. 2011.

[12] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, (Zurich, Switzerland), p. 1, ACM Press, 2010.

[13] B. Balaji, J. Koh, N. Weibel, and Y. Agarwal, "Genie: a longitudinal study comparing physical and software thermostats in office buildings," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, (New York, NY, USA), pp. 1200–1211, Association for Computing Machinery, Sept. 2016.

[14] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R. K. Gupta, M. B. Kjærgaard, M. Srivastava, and K. Whitehouse, "Brick : Metadata schema for portable smart building applications," *Applied Energy*, vol. 226, pp. 1273–1292, Sept. 2018.