Open Access

Online Journal of Robotics & Automation Technology

Iris Publishers

Review article

Reflections on Cyber Security Automation via SOA IoT Middleware

Alexandros Gazis[1]* and Stavros Stagkakis[2]

[1]*Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece*

[2]*Department of Cryptography, Security and Information Systems, Hellenic Military Academy, Greece*

***Corresponding author:** Alexandros Gazis, Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece.

Received Date: April 01, 2022

Published Date: April 13, 2022

Abstract

The Covid-19 pandemic forced companies to both alter their operation and adapt their activities and services provided. Indicatively, remote working, hybrid work models and teleconferences became the new reality, leading to the exponential increase of businesses' dependency on IT infrastructures, including internet usage and cloud computing. Moreover, the geopolitical environment affects cybersecurity and the likelihood of cyber-attacks on critical infrastructures. To address these issues, companies must update their digital infrastructure and reduce cyber vulnerabilities; this can be achieved through the use of middleware. In this article, we define the terms "middleware" and "Service Orientated Architecture" (SOA). Then, we explain how SOA can be fused with current middleware systems, using Automated Teller Machines (ATMs) as a case study. Finally, we review how a Cyber IoT Middleware uses SOA.

Keywords: Cyber security; Computer security; Cyber threats; Service Orientated Architecture; SOA middleware; Internet of things; IoT cybersecurity; Middleware

Introduction

Following the fourth industrial revolution, where physical, digital, and biological entities are expected to fuse into a single system, multiple scientific studies have focused on big data, AI and IoT [1]. Moreover, recent developments, including the Covid-19 pandemic and geopolitical relations (e.g. Russia - Ukraine) forced businesses to shift their focus to physical approaches in regards to customer and workforce experience [2]. The implementation of these changes put the companies' process flow and operation to the test. In other words, the maintenance, upgrade and expansion of their digital, cloud, and IT infrastructure put severe pressure on businesses. Simultaneously, cyber threats and security issues increased over 700% in recent years, according to Gartner [3], making companies more vulnerable to data breaches and ransomware attacks [4]. To address these issues, companies must both strengthen the architecture of their current computer infrastructures and embrace new technologies, such as middleware. Specifically, this term was first introduced during the 80s to describe the expansion of the traditional software design principles and upgrade existing legacy applications. Middleware's main purpose is to serve as a layer which provides a generic abstraction to an existing application. In Layman's terms, we should define middleware as a software solution that builds on the "separation of concerns" design principle; i.e. the development of an intermediate layer, serving as a "glue" software between software entities. For example, in IoT applications, the middleware layer is responsible for the communication between hardware (sensors or WSNs) and software layers. For accurate, consistent, and reliable machine-to-machine communication, these layers must interact flawlessly. To

achieve that, the middleware layer is responsible for coordinating, triggering and orchestrating all necessary services and processes to achieve optimal functionality [5]. In the following sections, we further define the term "middleware" and introduce the meaning of "Service Orientated Architecture" (SOA) design. Then, we examine a banking application for deposits to understand the importance of this architecture for the cybersecurity domain. Finally, we highlight the basic components of a Cyber Middleware layer that uses SOA and we propose future research, based on our study.

Defining Middleware and SOA

While there are numerous types of middleware (e.g. transactional, procedural, message-orientated, etc.), scholars typically distinguish them based on their use and structure logic. In this article, we categorize middleware based on its usage (i.e. general or service-specific). Given the agile nature of the software development and deployment life-cycle which is perpetually updated and modified to meet enterprises' requirements, middleware is used in conjunction with SOA and software design patterns [6]. Analytically, SOA is a pattern where software, hardware and networking are not rigidly integrated into a monolithic application; instead, they are separated into different entities that act independently. A software engineer approaches SOA as an autonomous system due to the unique properties of each entity. Thus, if a system is developed with a SOA, future modifications are applied to a specific service/module and not to the components of the overall system. The separation of entities and services enables us to break down each component into smaller services (microservices) and thus to both decrease the system's complexity and increase its modularity. As a result, the segregation to smaller software components simplifies businesses' focus on securing and expanding their applications. This way, in case of a cyber-attack, emphasis is placed on small (and autonomous) components that can be easily replaced or removed, without the need to perform an overall impact analysis.

The Automated Teller Machine (ATM) Example

Most core banking systems were developed in the 1950s; this means that they were built from the ground up as single-tiered software applications. When customers deposit or withdraw money from their accounts, they must input their request via a user interface (e.g. ATM screen) that translates it into a single service which queries a database and returns the results to the customer. This application is described as monolithic because it provides an end-to-end action that does not use multiple layers or modules. Moreover, if new functionalities were to be considered (e.g. new products for deposits, new activation/verification/authentication credentials or other e-banking features), these modifications need to be implemented from end-to-end. In order to make changes with ease and allow for applications to scale up and expand, we need to restructure applications in line with SOA, in a 4-tier architecture. The tiers would be composed of a User Interface, a Front-End controller, a Channel, and a Middleware layer. This way, instead of an application performing direct actions, each tier will be autonomous with a preset of input and output conditions. Lastly, the different layers will communicate via Application Programming Interfaces (APIs) through the Middleware tier. This layer will act as a "set of services below the application environment (i.e. below application-level APIs)" [7]. It will also handle requests for approval, block or view-only access mode, by clearly defining the application scope and usage, thus increasing its security. Moreover, the middleware will further categorize events, based on the business logic and IT operations to reduce synchronous to asynchronous states. This method will decrease the time of response for cyber operations (e.g. network trafficking, information, cyber events, etc.). After the presentation of a case study to understand how SOA can increase modularity and security, we shall review an important middleware for businesses to reduce cyber threats, i.e. a cyber IoT middleware that uses SOA.

Cyber IoT Middleware as SOA

Nowadays, the features of a traditional client-server, request-response model are relatively limited. This is the case because clients may only send data, after they submit their request to the server. This situation limits developers in terms of creating dynamic applications. One of the main architectural patterns that we studied aimed at surpassing static monolithic programs and developing full-duplex communications between different middleware layers. We propose the design of a layer that would act as a tool for a stable, bidirectional, full-duplex communication channel which would operate over HTTP/TCP/IP connections. To develop a Cyber middleware layer, emphasis should not be given to the protocol responsible for the persistent connection between a client-server, allowing real-time data flow (i.e. no AJAX requests). Middleware developers should focus on software abstraction to allow for a continuous data stream that instantaneous passes data as well as events among the client and server. Specifically, to define a cyber-middleware as an IoT SOA component, we need to develop a layer for handling the communication between systems and applications, as well as identifying and authorizing access to the layers, modules, services, and components. Moreover, this middleware will act as rails to securely transport information between the systems ("SSL Handshakes"). It will also propose diagnosing capabilities to acknowledge and resolve issues for each module. Last but not least, it will determine the distribution of applications in design, in addition to the development, integration, test, and production environments.

Conclusion

Data breaches, ransomware attacks and cyber threats are not a trend or a buzzword; they are a perilous reality that businesses must acknowledge and face with due diligence. Cyber security specialists and companies need to reduce their exposure points and upgrade the structure of their applications to smaller and separate layers. Moreover, end-to-end action should be broken down further via the use of web sockets (i.e. push paradigm) instead of server intensive tasks, such as long polling techniques (i.e. extension of the time that clients' connections remain open, until the server receives their data). This way, SOA shortens the response time for users and addresses resource limitations (i.e. firewall). Lastly, for future uses of the SOA design to middleware development, this architecture will help the independent study of each module. Consequently, future research should focus on modifying the necessary services, thus facilitating the development, maintenance, and safeguarding of critical information and digital data.

Acknowledgement

None.

Conflict of Interest

No Conflict of interest.

References

1. Wang J (2022) Influence of Big Data on Manufacturing Accounting Informatization. In: Xu Z, Alrabaee S, Loyola-González O, Zhang X, Cahyani NDW, Ab Rahman NH (Eds.), Cyber Security Intelligence and Analytics. CSIA 2022. Lecture Notes on Data Engineering and Communications Technologies: 125
2. Chen QC, Narasimhan L (2021) The potential of IoT-based smart environment in reaction to COVID-19 pandemic, Association for Computer-Aided Architectural Design Research in Asia (CAADRIA). 709-718.
3. Goasduff L (2018) Protect Your Organization from Cyber and Ransomware Attacks - Gartner.
4. Panneta K (2021) The Top 8 Security and Risk Trends We're Watching - Gartner.
5. Gazis A, Katsiri E (2022) Middleware 101: What to know now and for the future. Acm Queue 20: 10-23.
6. Al-Jaroodi J, Mohamed N (2012) Middleware is STILL everywhere!!! Concurrency and Computation: Practice and Experience. 24: 1919-1926.
7. Aiken B, Strassner J, Carpenter B, Foster I, Lynch C, et al. (2000) RFC2768: Network Policy and Services: A Report of a Workshop on Middleware.