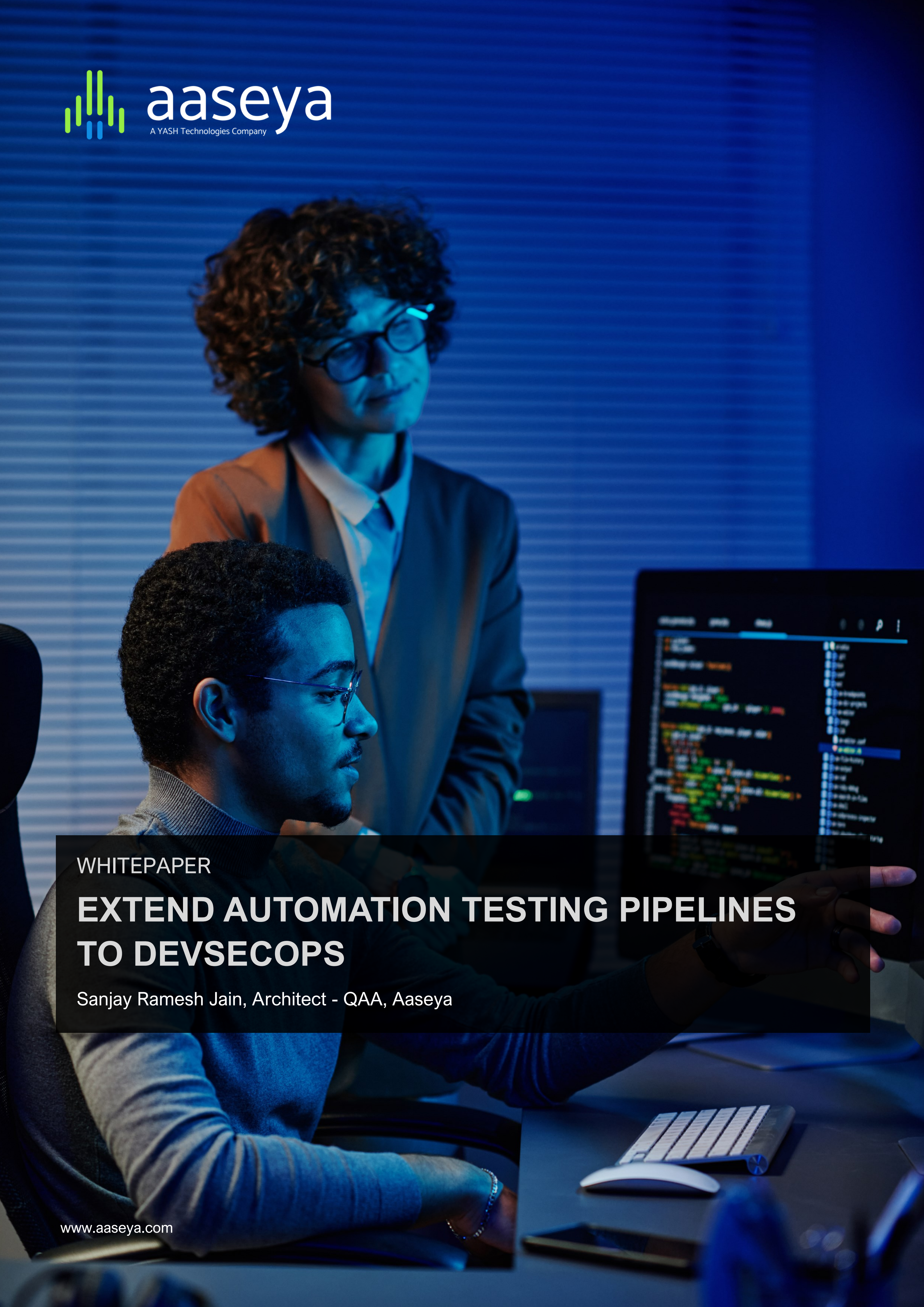
aaseya
A YASH Technologies Company



WHITEPAPER

EXTEND AUTOMATION TESTING PIPELINES TO DEVSECOPS

Sanjay Ramesh Jain, Architect - QAA, Aaseya

www.aaseya.com

ABSTRACT

This paper defines Web application security and forensics. It describes the testing techniques which can help to secure the web application, identify for any potential vulnerabilities, and defects which might lead to failure. It describes techniques which can help in achieving DevSecOps in a Software development environment.

INTRODUCTION

In the past decade, a great shift occurred in software development from creating Software as a Product (SAAP), that is executed as a single instance on customers' machines, towards providing Software as a Service (SaaS) where many users share instances that run on cloud infrastructure.

This provided software practitioners with the ability to continuously improve their product quality by releasing frequent updates. To manage these improvements efficiently, classical development (Dev) and operation (Ops) tasks were combined which resulted in a development concept termed DevOps. This concept is based on collaboration between the two former fields in all development stages and achieved by solving problems together, automating processes, and agreeing on mutual metrics to use when evaluating a system. This created the four pillars that guide teamwork in DevOps: culture, automation, measurement and sharing (CAMS). This agile development method enables software practitioners to test and deploy software versions at a much more frequent pace and hence respond to customers' demands rapidly. A prime example of this is Amazon, where a new version was released more than once per second.

While fast releases are considered to be beneficial to the quality of a product, they may also increase pressure on developers to finish their tasks more quickly. Studies such as Kraemer revealed that tight schedules or high workload can lead to the accidental introduction of security vulnerabilities into software systems. Kraemer also states that the reason for the presence of vulnerabilities is a lack of security knowledge in DevOps teams. This affects the quality of security tests and hence diminishes the security of a system. In addition to this, cybercrime is increasing in recent years. For instance, the number of stolen or compromised records has increased by 133% from 2017 to 2018. Furthermore, security and privacy regulations such as the General Data Protection Regulation (GDPR) have been implemented in the EU to enforce security standards and punish companies harshly if these regulations are violated. All these aspects show that security concerns have become increasingly important.

Continuous Integration (CI) and Continuous Delivery (CD) have become a well-known practice in DevOps to ensure fast delivery of new features. This is achieved by automatically testing and releasing new software versions, e.g. multiple times per day. However, classical security management techniques cannot keep up with this quick Software Development Life Cycle (SDLC). Nonetheless, guaranteeing high security quality of software systems has become increasingly important. The new trend of DevSecOps aims to integrate security techniques into existing DevOps practices. Especially, the automation of security testing is an important area of research in this trend. Although plenty of literature discusses security testing and CI/CD practices, only a few deals with both topics together. Additionally, most of the existing works cover only static code analysis and neglect dynamic testing methods. In this paper, we present an approach to integrate three automated dynamic testing techniques into a CI/CD pipeline and provide an empirical analysis of the introduced overhead. We then go on to identify unique research/technology challenges the DevSecOps communities will face and propose preliminary solutions to these challenges. Our findings will enable us to make informed decisions when employing DevSecOps practices in agile enterprise applications engineering processes and enterprise security.

This increased focus on security introduced a new field called DevSecOps, which attempts to integrate security (Sec) practices into DevOps. Traditionally, security experts were organized into separate silos and security concerns were addressed after the actual design and development stages. Similar to the inception of DevOps, DevSecOps attempts to promote collaboration between development, operations and security teams. DevSecOps establishes a proactive approach to limit the attack surface of the application and entails considering security from the very beginning of the project. However, the integration of security practices into modern software engineering creates several problems. Firstly, traditional security methods are not applicable because they cannot keep up with the agility and speed of DevOps. Secondly, very little is known about DevSecOps so far, as only a few studies were conducted on this topic. Especially the lack of knowledge of when and where to use (existing) tools in automation is a considerable problem that prevents software practitioners from integrating security into their DevOps activities such as continuous integration and continuous deployment (CI/CD)

Until now, research has identified the principles, priorities, and practices in DevSecOps. It appears that the automation principles.is equally significant in security. This enables security teams to keep up with the DevOps and establishes fast, scalable, and effective security tests.

However, most literature focuses on automatic security testing through Static scans of source code. Although important, static application security testing (SAST) cannot detect all security vulnerabilities in a system. In fact, SAST analysis is only able to find those vulnerabilities which are subset and are limited.

Index Terms: *DevSecOps, Dynamic Security Web Testing, Continuous Security, Continuous Integration.*

TESTING AND DEVSECOPS

A Functional expert usually gains business knowledge and becomes an SME. Nowadays, just SME wouldn't be fruitful but also need to enhance the skillset on test automation to achieve the goal of continuous deployment and testing. Shifting left in agile will be helpful only when a hybrid tester has a vision of achieving DevOps and include DevSecOps in the pipeline of testing.

Static testing and dynamic testing are already part of testing activities which is performed by a tester. But when there are tools like SonarQube which can ease the static testing and help in catching bugs early then it’s always better to plan for such activities to integrate with the deployment pipelines to achieve DevOps.

Recent times, non-functional testing is also automated and tagged with the DevOps to achieve quicker results especially for APIs and server-side testing. This leads a path to DevSecOps where a tester can include security testing tools & develop scripts in the pipeline to achieve DevSecOps. Security vulnerabilities are more costly if found at the later phase of the SDLC. Henceforth it is recommended to have DevSecOps implemented.

An automation tester can enable skillset to write up the security test scripts using tools which are SAST and DAST related like SonarQube, zapper or burp suite. These scripts can then be plugged in the DevOps pipeline so that a continuous deployment is achieved where functional, & non-functional testing is executed on every build which is deployed either in test environment or higher. The goal is to shift security left in the SDLC (Software development life cycle). The goal is to improve the coverage and effectiveness of security processes, increase software quality, shorten test cycles, and reduce the security debt. It is easier to fix the security bugs at the early stages like we do for bugs.

REAL-TIME IMPLEMENTATION CHALLENGES

The DevOps team who attempts to take ownership of the DevSecOps and implement them will foresee some of the challenges while implementation. Some of them are outlined below:

SKILLSET KNOWLEDGE

Automation team who implements the pipeline to integrate the automation suites with the deployment build usually doesn't have the knowledge about the development code. Likewise, developers who help in DevOps implementation don't have the knowledge of security testing and its standards. Developers who understand security concepts and best practices can start implementing them in every task.

CONTAINER SECURITY RISKS

Transient containers and microservices are difficult to monitor, while misconfiguration of container networking can leave your production environment vulnerable. Containers are often used to break down applications into microservices, which increases data traffic. Traditional server security solutions don't support containers; consider specialized security technology that can lock down containers with safe configuration, scan images to ensure they are safe, and monitor containers in production.

There are different vulnerabilities which a container can inherit from the base the base images or introduced during the image build process. Even for a container, attackers may exploit privileged containers to gain root access to the host system, leading to unauthorized access or control. Some Misconfigurations in container environments, such as weak access controls, exposed APIs, or insecure network configurations, can lead to unauthorized access or data breaches. Improper handling of sensitive data within containers can result in data leakage or unauthorized access, especially in multi-tenant environments.

Tools: Neu Vector

Solutions

- Regularly update and patch base images, scan images for vulnerabilities using security scanning tools, and adhere to best practices for image creation and management.
- Avoid running containers with unnecessary privileges, use least-privileged user accounts, utilize container security features like user namespaces, and implement role-based access control (RBAC) to restrict container capabilities.
- Regularly audit & review container configurations, follow security best practices for container deployment & orchestration platforms, & implement network segmentation and firewall rules to restrict access.

CLOUD DEPLOYMENT RISKS

Cloud deployment has its own benefits which also come along with its own security risks. The commonly seen risks are as follows:

- Unauthorized access to sensitive data stored in the cloud can lead to data breaches, resulting in financial loss, reputational damage, and regulatory penalties.
- Weak or misconfigured access controls may result in unauthorized users gaining access to cloud resources, leading to data exposure, data loss, or service disruption.
- Insecure cloud interfaces and APIs can be exploited by attackers to gain unauthorized access, manipulate data, or execute malicious activities.
- Denial of Service attacks can disrupt cloud services by overwhelming infrastructure resources or network bandwidth, resulting in downtime and service unavailability.

Tools: Cloud Formation, Azure (ARM)

Solutions

- Encrypt data both at rest and in transit, implement strong access controls and identity management practices, regularly audit, and monitor access logs, and use data loss prevention (DLP) tools to detect and prevent unauthorized access.
- Securely configure cloud interfaces and APIs, use encryption and authentication mechanisms to protect data in transit, implement API security best practices such as rate limiting and input validation, and regularly audit API usage and access logs.
- Implement DoS protection mechanisms such as rate limiting, traffic filtering, and load balancing, utilize distributed denial of service (DDoS) mitigation services provided by cloud service providers, and monitor for abnormal traffic patterns and anomalies.

APPLICATION SECURITY RISKS

A web application is built using open source and commercial libraries or framework. The web application works on the internet and has more security risks exposed. The commonly seen risks are as follows:

- Injection attacks, such as SQL injection and cross-site scripting (XSS), occur when attackers insert malicious code into input fields to manipulate or compromise the application's data or functionality.
- Weak IAM (Identity Access Management), usually giving more access to the users without any use.
- Broken Authentication, Improper session management, and insecure password storage can lead to unauthorized access to user accounts and sensitive data.
- Sensitive data exposure like credit card, account information, and personal information.
- Security misconfiguration of web servers, application frameworks and security settings.
- Unvalidated redirects and forwards can be exploited by attackers to redirect users to malicious websites or phishing pages, leading to credential theft or malware installation.

Tools: Burp Suite, OWASP, Zapper

Solutions

- Use parameterized queries and prepared statements to prevent SQL injection, validate and sanitize user input to mitigate XSS attacks, and implement web application firewalls (WAFs) to filter and block malicious requests.
- Implement strong password policies, use multi-factor authentication (MFA) for user authentication, enforce secure session management practices, and token-based authentication.
- Monitoring the logs for any suspicious activities.
- Encrypt sensitive data both at rest and in transit using strong encryption algorithms.
- Follow security best practices for configuring web servers and application frameworks, regularly update and patch software components.
- Conduct regular Security assessments and scans and use automation tools to identify the weak configuration.
- Implement whitelisting or predefined destinations for redirections.

TOOLS TO ADOPT DEVSECOPS

It’s always easy to adopt the security testing tools into the existing developer infrastructure. We can even set up guardrails to ensure that the developers cannot bypass these systems. The commonly used tools are as follows.

SAST TOOL LIST

- Open Web Application Security Project (OWASP)
- SonarQube
- Fortify
- SCA

SonarQube

SonarQube is one of the SAST tools. It helps in the static analysis of the code enabling systematically deliver and meet high code quality standards, for every project, at every step of the workflow. This integrates with the DevOps pipelines. Developers / QA can configure the critical security rules for different languages. Developers can integrate with the local IDE also to run them locally before pushing the code into the repository.

SonarQube includes a powerful secrets detection tool, one of the most comprehensive solutions for detecting and removing secrets in code.

Synk, A Software Composition Analysis (SCA)

Software composition analysis (SCA) is an open-source component management tool. It generates a report listing all open-source components in an application including direct and indirect dependencies. Using an SCA tool, development teams can quickly track and analyze open-source components introduced into a project.

Snyk is an SCA tool and provides automated security scanning for open-source dependencies and containers, offering vulnerability detection, license compliance, and dependency monitoring.

DAST tool list

Dynamic Application Security Testing (DAST) tools are used to identify security vulnerabilities in web applications by analyzing the application's runtime behavior and interactions.

Some of the commonly used are:

- **Burp Suite:** Burp Suite is a comprehensive web application security testing platform that includes a DAST scanner for identifying common vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR). Burp suite also helps in penetration testing and can perform web scanning for only commercial license.
- **OWASP ZAP:** OWASP ZAP is an open source DAST tool maintained by the Open Web Application Security Project (OWASP). It offers automated scanning capabilities for detecting vulnerabilities in web applications, along with a range of manual testing features.

KEY ELEMENTS FOR SUCCESSFUL DEVSECOPS

A web application is built using open source and commercial libraries or framework. The web application works on the internet and has more security risks exposed. The commonly seen risks are as follows:

- Integrate automated testing into the pipeline.
- Integrate security testing into workflows.
- Automated deployment and continuous deployment.
- Continuous monitoring and reporting any suspicious activities.
- Remediating application security vulnerabilities.
- Train engineers in Secure DevOps.

CONCLUSION

Continuous integration and deployment is common in agile environment to achieve a better quality product and better time to market. This can be extended to include non-functional testing where performance testing and security testing can be extended to void late identification of non-functional bugs or vulnerabilities. Adding a performance testing integrated with the DevOps will yield better results when considered the hardware and software cost involved. But Security testing extended to a DevOps can yield a better result and return to the investment (ROI) and make the testing left in the development process by even cutting down the time taken for security testing at the later stage. DevSecOps is one such approach is such way to implement the security test of DAST and SAST to pipeline with the build tools for better results.

REFERENCES

- **An Integrated Framework for DevSecOps Adoption. (n.d.).** Retrieved from https://www.researchgate.net/publication/361825047_An_Integrated_Framework_for_DevSecOps_Adoption
- **Atlassian. (n.d.). DevOps Tools: DevSecOps Tools.** Retrieved from https://www.atlassian.com/devops/devops-tools/devsecops-tools
- **Atlassian. (n.d.). Snyk Tool Implementation.** Retrieved from https://www.atlassian.com/devops/security-tutorials/jira-snyk-devsecops
- **OWASP. (n.d.). DevSecOps Maturity Model.** Retrieved from https://owasp.org/www-project-devsecops-maturity-model/

For more information contact
info@aaseya.com or visit www.aaseya.com

Aaseya Global Presence
AMERICAS | EUROPE | APAC | MEA

About Aaseya

Aaseya is a leading professional services company specializing in low-code and digital process automation. Our team of over 600 associates deliver agile implementations of cutting-edge technologies such as Pega, OutSystems, and Camunda, dramatically reducing the time-to-value for our clientele. Operating across 13 countries, Aaseya has achieved over 145 successful Go-Lives within a mere six-year period. As a subsidiary of YASH Technologies, a leading global system integrator with a workforce exceeding 8,000 employees and serving over 450 clients worldwide, Aaseya is dedicated to fostering innovation, achieving excellence, and ensuring client success on their digital transformation journey.

aaseya
A YASH Technologies Company

PEGA | Global Elite Partner

outsystems Partner DELIVERY

CAMUNDA | Gold Partner