International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 11
Article Received: *25 July 2022* ***Revised:*** *12 September 2022* ***Accepted:*** *30 November 2022*

Adopting DevOps in SAP Product Development: Innovative Methods for Continuous Integration and Delivery

Rahul Ranjan

Lead Development Architect & Solution Architect, Product Engineering, SAP America Inc, Newport Beach, Orange, California

Nationality: Indian

Email: fromrahulranjan@gmail.com, ORCID: https://orcid.org/0009-0002-0754-3270

Abstract

The main purpose of this research is to look into the adoption of DevOps in SAP product development from the perspective of issues about continuous integration (CI) and continuous delivery (CD). The study highlights substantial gaps, such as those stemming from manual-filtered monolithic systems and security compliance. Results show that automation with Jenkins, GitLab, and SAP gCTS increases the speed and reliability of deployments, while compliance and risk management are improved through the use of DevSecOps and shift left security testing. An integrated approach to automation, security, and agile collaboration as put forth by this study, establishes an efficient software delivery system within a scalable SAP DevOps infrastructure. This research underlines the importance of having well-defined strategic frameworks for DevOps to allow for governance, agility, and performance-centred SAP product development.

Keywords: DevOps, SAP, CI/CD, Automation, Security, Compliance, Integration, Deployment, Framework, Scalability

Introduction

Implementing DevOps helps improve CI and CD within SAP product development. In the past, SAP development processes were constrained by lengthy release cycles and manual test procedures. DevOps introduces mechanisms for automation, collaboration, and agile development to ensure dependable and timely deployments. SAP monolithic case hampered the adoption of CI and CD pipelines. Newer models make use of SAP Cloud Platform, Containers, and Microservices (JAMPANI *et al*. 2021). CI tools like Jenkins, GitLab, and Azure DevOps facilitate automated building processes. Errors are minimized through the use of automated testing frameworks such as Selenium, Tricentis, and SAP Solution Manager. A uniform environment is guaranteed by an implementation of IaC with Terraform, Ansible, and BTP. Compliance checks are automated in the early stages of the development process in DevOps Security (DevSecOps) (Hsu, 2018). Rapid feedback loops benefit ABAP and Fiori development. Automatic transport management and rollback features are made possible through the use of CI/CD pipelines. Cloud native deployment strategies are supported by SAP BTP and Kubernetes providing extensive functionality. Toggle features and canary deployments improve flexibility during deployment. Development, Test, and Operations merge roles which removes inefficiencies in processes. Scrum and Agile work approaches are complementary to the principles of DevOps. Versioning is made easier with the use of SAP accelerators such as CTS. Quality is incorporated in the first steps of the process with shift left testing. Monitoring is improved using observability tools such as Dynatrace and Splunk. The risk of downtime is minimized using rollback features and blue-green deployments. Data security and integrity are always of top importance. Adherence to GDPR and regulations that surround the industry is crucial. Synthetic monitoring along with automated performance testing helps in preventing failures, while real-time analytics along with dashboards help in the decision-making process. In SAP cloud ecosystems, scalability is achievable through increased use of the cloud. Workflows are optimized through edge computing and AI automation (Tikkinen-Piri *et al*. 2018). Team productivity is improved through automated documentation and knowledge sharing. Success is driven by cultural transformation and executive sponsorship. Long-term benefits are guaranteed through skill enhancement in SAP DevOps. A competitive edge is maintained through innovative and continuous learning.

Problem statement

Traditional processes of product development in SAP have issues integrating DevOps for continuous integration (CI) and

continuous delivery (CD). Innovation is impaired by protracted release cycles, manual testing, and inflexible system architectures. Deeply rooted deficiencies in automation of transport management and integration of CI/CD tools result in losses of productivity. Inefficient collaboration between teams and gaps within the security compliance framework create delays in deployment. Adopting DevOps and cloud-native technologies is slowed down by SAP's monolithic structure. Inconsistencies in environments and rollback problems elevate risks of failure (JAMPANI *et al*. 2021). Gaps in observability and automation of testing raise concerns about the quality of the product. These hurdles are further worsened by the skills gap in DevOps for SAP. Solving these problems is imperative for establishing a scalable, secure, and efficient SAP product development system.

Research Aim

This research aims to explore innovative DevOps adoption methods in SAP product development to enhance continuous integration (CI) and continuous delivery (CD) by addressing automation, security, agility, and collaboration challenges.

Research Objectives

- To identify key barriers to DevOps adoption in SAP product development.
- To evaluate automation and CI/CD tools for improving SAP deployment efficiency.
- To analyze the impact of security and compliance integration in SAP DevOps.
- To develop a framework for agile, scalable, and collaborative SAP DevOps implementation.

Literature review

The integration of DevOps within SAP product development has received considerable attention concerning its possible improvements in continuous integration (CI) and continuous delivery (CD) processes. Numerous researchers observe that traditional SAP environments suffer from poor agility, automation, and deployment effectiveness due to their monolithic building blocks and heavily manual processes. Research suggests that these include older systems, complex transport management, and insufficient skills or knowledge relevant to DevOps practices. The absence of automation in alarm SAP CI/CD pipelines lowers the speed at which systems are available and increases risks associated with the operational processes (Reddy, 2021). Scholars emphasize automation tools, such as Jenkins, GitLab, and SAP gCTS, as accelerators of build, test, and release cycles. Research centered around CI/CD in SAP landscapes shows that the adoption of Infrastructure as a Code (IaC), containerization, and the cloud are key enablers of greater deployment efficiency. There is a significant gap with regards to integration of security and compliance issue within SAP DevOps. Research suggests the implementation of some approaches, such as those of DevSecOps, integrated compliance shift-left testing, and automated compliance checking, to address those issues. Researchers recommend using CI pipelines and tools such as SAP Charm or Tricentis to automate workflows for implementing security policies (Voruganti, 2021). Automating SAP transport management is needed in agile environments to support reliable change tracking and automated rollback functionality. Scholars also elaborate the challenges related to collaboration across development, operations, and testing disciplines, which diminishes the effectiveness of SAP DevOps. Agile and Scrum have been observed to improve collaboration, but unless there is a cultural shift coupled with organizational buy in, the progress will be limited. As the literature suggests, real-time observability, synthetic monitoring, and AI-driven automation allow for the optimization in SAP system performance. Research on cloud SAP deployment shows the enhancements for scalability and flexibility provided by SAP BTP, Kubernetes, and Microservices. Further research discusses the use of feature toggles, blue-green deployments, and canary releases as methods for minimizing downtime while maximizing deployment safety (Munoz *et al*. 2020). It is stated that this increases the need for skills development in SAP DevOps, thus training courses, knowledge bases, and constant education opportunities are recommended. Dealing with those problems, researchers give out guidelines all-encompassing towards scalable and secure SAP DevOps adoption focusing on tool integration, automation, security, and agile methodology (Kulkarni, 2019). This work demonstrates that there must be a mix of automation and security enhancement, collaboration improvement, and agile strategy to deal with the barriers of SAP DevOps adoption.

Methodology

This study uses a secondary research method to look into DevOps adoption in SAP product development. Using scholarly articles, industry literature, and case studies already published, this approach helps understand the integration of security and automation tools into CI/CD processes fully. Secondary research provides a low cost, wide availability of data, and analysis of historical metrics. It makes it possible to discern optimal practices, frameworks, and strategies without the hassle of gathering first-hand information. The method improves the reliability and comprehensiveness of the supportive SAP DevOps adoption implementations. The synthesized and multi-sourced information presented in this

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 11
Article Received: *25 July 2022* ***Revised:*** *12 September 2022* ***Accepted:*** *30 November 2022*

research provides efficient, scalable, and secure solutions for SAP product development.

Result and Discussion

Key Barriers to DevOps Adoption in SAP Product Development

Multiple obstacles stand in the way of integrating DevElOps in SAP product development, including aging monolithic systems, outdated manual workflows, and organizational divisions. SAP's architecture is traditionally made up of rigid, siloed systems that do not support Continuous Integration (CI) and Continuous Delivery (CD) (Bobrovskis and Jurenoks, 2018). SAP systems, unlike modern, cloud-native applications, use a transport-based deployment model which renders automation impractical.

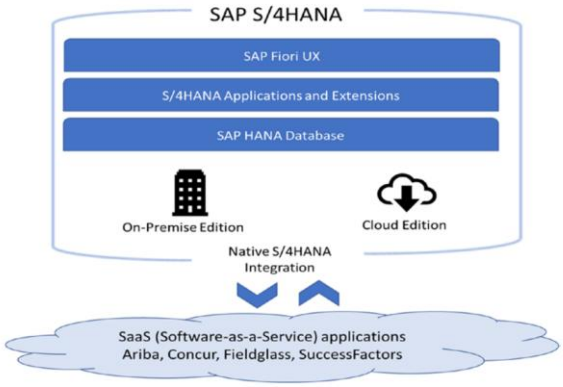

Figure 1: SAP Architecture

(Source: Kulkarni, 2019)

Manual transport management is a significant impediment because standard SAP change control relies on sequential transport request processing which results in delays, errors, and complexities surrounding rollbacks. The level of automation in place results in inefficient deployments and operational slowdowns. In addition, SAP's reliance on centralized development environments hampers agile parallel development and testing. Further delays in the adoption of SAP DevOps tools and automation practices are attributed to gaps in skills within the organization. Many SAP teams lack basic competencies with C1/CD tools such as Jenkins, GitLab, and gCTS. Unless there is proper cultural and educational transformation, integration of DevOps processes will continue to be fragmented. Security and compliance are other equally challenging barriers. SAP applications deal with sensitive enterprise data which is under strict regulatory compliance (GDPR, ISO 27001, SOC 2). However, traditional security tends to be applied too late in the development lifecycle, resulting in greater risks (Woolf, 2018). The absence of DevSecOps, including shift-left security checks and automated compliance checks, increases the gap in SAP DevOps implementation. Furthermore, collaboration between developers, operations, and security teams is hindered by team silos along with resistance to change. The integration of DevOps with SAP is only adopted to a limited extent due to a lack of efficient communication and shared responsibility. These barriers can be solved with automation and security integration, which transforms the culture to make the adoption of SAP DevOps secure, efficient, and scalable.

Impact of Automation and CI/CD Tools on Deployment Efficiency

The enhancement in automation and CI/CD tools has had a great impact on SAP product development in terms of deployment speed, reliability, and consistency. Traditional SAP deployment methods suffer from prolonged release cycles and inefficient operations due to their dependence on manual transport management systems. The integration of CI/CD enables error and delay minimization through automated build, testing, and deployment cycles. Code versioning, change tracking, and transport automation are made seamless with the Jenkins, GitLab CI/CD, and SAP gCTS tools (Jollien, 2021).

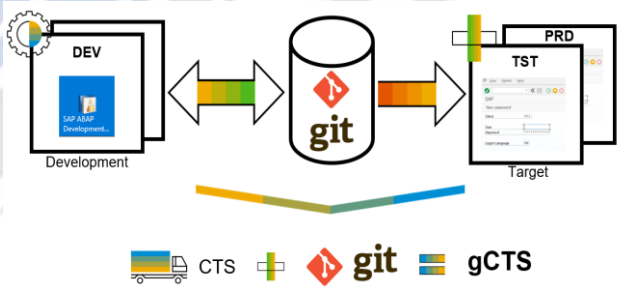

Figure 2: GitLab CI/CD, and SAP gCTS tools

(Source: Karin, 2019)

The ability to work in parallel improves collaboration and productivity among team members, which is made even easier through automated transport approvals which decreases the risk associated with manual interventions. Managing consistent development, testing, and production environments is achieved through the use of Infrastructure as Code (IaC) tools like Terraform, Ansible, and BTP (SAP Business Technology Platform). The mitigation of configurational drift and deployment failure is made possible through the automation of environment provisioning with these tools. Resource optimization and scalability in SAP landscapes is met by containerization using Docker and orchestration with Kubernetes. Automated testing frameworks such as Selenium, Tricentis, and SAP Solution Manager lead to improved code quality through enabling

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 11
Article Received: *25 July 2022* ***Revised:*** *12 September 2022* ***Accepted:*** *30 November 2022*

continuous testing. Costs post-deployment issues can be avoided using shift left testing which allows security vulnerability and functional defect detection during the earliest stages. Enhanced deployment flexibility is achieved through rollback mechanisms such as blue-green deployments and canary releases (Madupati, 2021). These methods allow instant rollback and gradual rollouts, minimizing disruption and downtime. Performance monitoring and failure detection are provided by Dynatrace, Splunk, and Prometheus. These tools, alongside the CI/CD automation technologies, allow Divison of SAP Product Development to deploy new releases faster, more stable, and with lower operational risk. These tools are instrumental in creating agile and efficient DevOps SAP environments.

Security and Compliance Integration in SAP DevOps

Security and compliance aspects have a direct impact on the adoption of SAP DevOps owing to the sensitive nature of the enterprise data handled by SAP applications and the necessity to comply with regulations such as GDPR, ISO 27001, and SOC 2 (Woolf, 2018). In most traditional SAP development environments, security is often treated as a distinct phase, which increases detection latency with regards to vulnerabilities and escalates the risk to production. Integration of security at early stages of the development lifecycle enables continuous security and compliance monitoring, as well as disaster recovery validation and mitigation. Shift left security testing is an important element for the identification of vulnerabilities in SAP CI/CD pipelines.

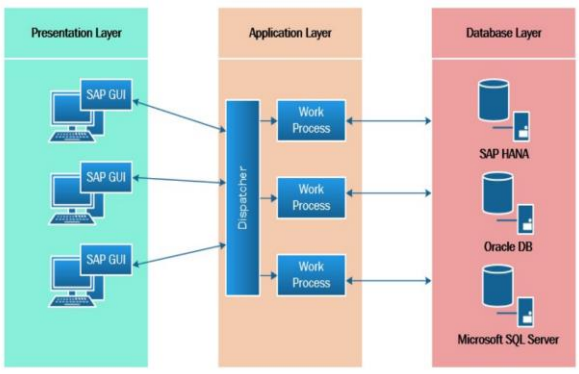

Figure 3: Web-Based SAP Architecture

(Source: Amini *et al.* 2020)

SonarQube, Checkmarx, and OWASP Dependency-Check are examples of tools that automate static application security testing (SAST) and dynamic security application testing (DAST) to find operational and structural weaknesses in the code prior to its deployment. DevOps has the ability to prevent production security breaches by integrating security scans into build pipelines prior to delivering code. Automated compliance validation allows the SAP application to confirm that it meets the requirements set by the external industry regulations and the internal policy documentation. Open Policy Agent (OPA) and HashiCorp Sentinel are examples of Policy-as-Code frameworks that automatically implement security policies in CI/CD pipelines (Gopireddy, 2020). Monitoring enforce compliance therefore acts as a regulation, audit, and logs analysis further guarantees compliance with set standards. SAP Solution Manager and Tricentis tools enable Transport security automation which removes unauthorized alteration and misconfiguration of SAP environments. SAP transports and code repositories are protected through role-based access control (RBAC), multi-factor authentication (MFA) and encryption standards. The gCTS method of secure transport validation considerably improves versioning as well as auditability. With tools such as Splunk, Dynatrace, and SAP ETD (Enterprise Threat Detection) security monitoring can now be done in real-time which helps with proactive threat mitigation. Such tools correlate log data, network activity, and system behavior to identify cyber threats prior to them inflicting any damage. In addition, rollback and disaster recovery procedures guarantee secure deployment. Organizations can deal with security breaches or compliance failures due to immutable infrastructure, automated backups, and blue-green deployments. Through the implementation of DevSecOps practices, SAP DevOps achieves gapped and validated compliance, complete security monitoring, minimalistic product development risk, and maximal effort towards mitigating cyber threats and regulatory breaches (Miller, 2019).

Framework for Agile, Scalable, and Collaborative SAP DevOps

Combining automation, security, scalability, and collaboration enables the effective and efficient delivery of software as required in an SAP DevOps framework. Along with traditional SAP product development, product releases are accompanied with excessive manual processes and rigid architectures which prolong the integration of DevOps immensely. Integration of security and compliance enables an organization to implement Controlled Continuous Integration (CI) and Continuous Delivery (CD) along with an agile and scalable framework (Arachchi and Perera, 2018). Automation is the primary step to enabling the workflow in an SAP DevOps framework. Jenkins, GitLab CI/CD, and SAP gCTS serve as better options of code versioning, transport management, and unit testing respectively.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 11
Article Received: *25 July 2022* ***Revised:*** *12 September 2022* ***Accepted:*** *30 November 2022*

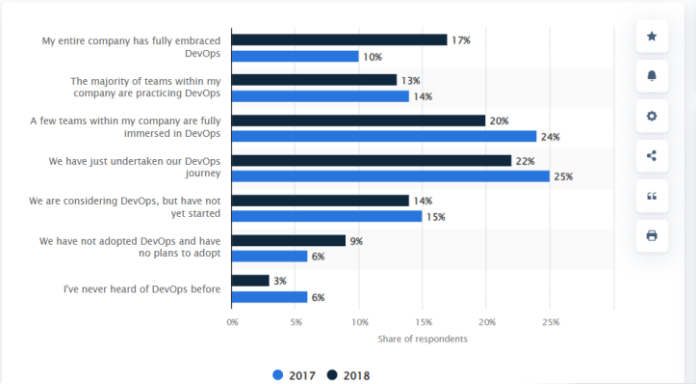

Figure 4: Rate of DevOps adoption by software developers

(Source: Lionel Sujay Vailshery, 2022)

Consistent environment provisioning across development, testing, and production is achieved through containerization using Docker and Kubernetes orchestration. Implementation of Infrastructure as Code (IaC) through Terraform, Ansible, and SAP BTP ensures consistency across different environments. Running SAP applications both in the cloud and on premise is made possible through use of containerization with Docker and Kubernetes orchestration, which provides enhanced resource efficiency and scalability. With CI/CD pipelines and implementing Scrum and Kanban, incremented improvements and faster cycles are achieved in SAP DevOps (Gonzalez Alzate, 2021). Safe deployment strategies such as feature toggles, blue-green deployments, and canary releases manage the risks that accompany downtimes. Proactive maintenance, alongside log analysis and performance monitoring, are automated through AI which guarantees stable and scalable SAP operations. Having security and compliance integrated is important in a scalable DevOps framework.

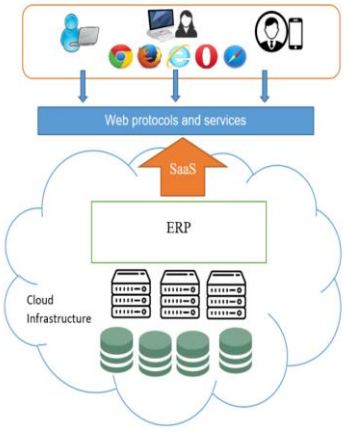

Figure 5: Cloud Based ERP

(Source: Amini *et al*. 2020)

Shift-left testing, automated compliance enforcement, and real-time monitoring of security threats ensure cyber protection for SAP landscapes. Compliance monitoring for GDPR, ISO 27001, SOC 2, and security threat monitoring are taken care of by SAP Enterprise Threat Detection (ETD), Splunk, and Dynatrace. Effective collaboration among developers, operations, and security personnel is essential in adopting SAP DevOps. Cross-functional teams working with automated documentation and knowledge management systems tend to be more productive and innovative. Executive sponsorships and training investments expedite adoption of these practices and technology, ensuring teams are fully equipped to traverse skills gaps, advancing SAP DevOps maturity. The integration of automation, agile, integrated security, and cross-team collaboration creates a solid SAP DevOps framework that improves the speed of software delivery, operational resilience, and business agility (Vadapalli, 2018). This method achieves effective, efficient, and secure SAP product development while reducing risks and optimizing performance.

Conclusion

This study emphasizes the difficulties as well as the advantages from automation, security integration, and framework construction pertaining to DevOps adoption in SAP product development. Findings validate that the existence of monolithic architectures, manual processes, and skill gaps are barriers to the efficient implementation of CI/CD. Automation tools such as Jenkins, GitLab, and SAP gCTS increase the speed and the reliability of deployments, while Infrastructure as Code (IaC), containerization, and adoption of cloud computing improve scalability. Practicing DevSecOps, employing shift-left security testing, and automating compliance measures reduce security and regulatory issues. An effective SAP DevOps framework enables efficient, secure, and scalable deployment by seamless integration of automation, security, agile principles, and multidisciplinary collaboration. Overcoming these factors through automation and enhancement of security and cultural change guarantees continuous delivery and operational agility in SAP environments. The study justifies the existence of unstructured DevOps silos which focus on automation for productivity gains, compliance, and agility for SAP innovation and sustainability.

References

[1] Amini, M. and Abukari, A.M., 2020. ERP systems architecture for the modern age: A review of the state of the art technologies. *Journal of Applied Intelligent Systems and Information Sciences*, *1*(2), pp.70-90.

[2] Arachchi, S.A.I.B.S. and Perera, I., 2018, May. Continuous integration and continuous delivery pipeline automation for agile software project management. In *2018 Moratuwa Engineering Research Conference (MERCon)* (pp. 156-161). IEEE.

[3] Bobrovskis, S. and Jurenoks, A., 2018. A Survey of Continuous Integration, Continuous Delivery and Continuos Deployment. In *BIR workshops* (pp. 314-322).

[4] Gonzalez Alzate, J.D., 2021. Implementación de CI/CD en Kubernetes usando Kaniko y Tekton.

[5] Gopireddy, S.R., 2020. Automated Compliance as Code for Multi-Jurisdictional Cloud Deployments. *European Journal of Advances in Engineering and Technology*, *7*(11), pp.104-108.

[6] Hsu, T.H.C., 2018. *Hands-On Security in DevOps: Ensure continuous security, deployment, and delivery with DevSecOps*. Packt Publishing Ltd.

[7] JAMPANI, S., MUSUNURI, A., MURTHY, P. and GOEL, O., 2021. Optimizing Cloud Migration for SAP-based Systems.

[8] Jollien, Y., 2021. Analyse et développement ABAP simple sur SAP Cloud Platform (SCP) dans un contexte projet avec l'État du Valais.

[9] Karin, 2019. *Learn how to extend and personalize SAP applications. Follow the SAP technology blog for insights into SAP BTP, ABAP, SAP Analytics Cloud, SAP HANA, and more.* Accessed from https://community.sap.com/t5/technology-blogs-by-sap/gcts-is-here/bc-p/13444472

[10] Kulkarni, S., 2019. Implementing SAP S/4HANA. *Implementing SAP S/4HANA*.

[11] Lionel Sujay Vailshery, 2022. *Extent of DevOps adoption by software developers worldwide in 2017 and 2018* Accessed from https://www.statista.com/statistics/673505/worldwide-software-development-survey-devops-adoption/

[12] Madupati, B., 2021. Kubernetes: Advanced Deployment Strategies-* Technical Perspective.

[13] Miller, S., 2019. DevOps Tools and Technologies: A Comparative Study.

[14] Munoz, R., Vázquez-Gallego, F., Casellas, R., Vilalta, R., Sedar, R., Alemany, P., Martinez, R., Alonso-Zárate, J., Papageorgiou, A., Catalan-Cid, M. and Moscatelli, F., 2020, June. 5GCroCo barcelona trial site for cross-border anticipated cooperative collision avoidance. In *2020 European Conference on Networks and Communications (EuCNC)* (pp. 34-39). IEEE.

[15] Reddy, S., 2021. Regression Testing for Continuous Deployment: Strategies to Keep Up with Rapid Changes. *Journal of Multidisciplinary Research (JOMR)*, *7*(01), pp.44-52.

[16] Tikkinen-Piri, C., Rohunen, A. and Markkula, J., 2018. EU General Data Protection Regulation: Changes and implications for personal data collecting companies. *Computer Law & Security Review*, *34*(1), pp.134-153.

[17] Vadapalli, S., 2018. *DevOps: continuous delivery, integration, and deployment with DevOps: dive into the core DevOps strategies*. Packt Publishing Ltd.

[18] Voruganti, K.K., 2021. Implementing Security by Design practice with DevSecOps Shift Left Approach. *Journal of Technological Innovations*, *2*(1).

[19] Woolf, C., 2018. All AWS Services GDPR Ready. *AWS Security Blog. h ps://aws. amazon. com/blogs/security/all-awsservices-gdpr-ready*.

[20] Woolf, C., 2018. All AWS Services GDPR Ready. *AWS Security Blog. h ps://aws. amazon. com/blogs/security/all-awsservices-gdpr-ready*.