PERSPECTIVE

Scalability and High-Availability Architectural Strategies for Apigee in Hybrid and Multicloud Environments

José Mariano Ruiz Martín[1,*]

[1] Independent Researcher, Madrid, Spain

Abstract

API management platforms have become critical components for operational resilience in distributed enterprise architectures. This article analyzes scalability and high-availability (HA) strategies applicable to Apigee, with particular emphasis on its Apigee X deployment model (Google Cloud–hosted SaaS) and Apigee Hybrid (a hybrid platform with a cloud-hosted control plane and a self-managed data plane). The state of the art in multi-region and multicloud deployments is presented, detailing active–active versus active–passive patterns and their impact on latency and disaster recovery objectives. Key technical components are examined, ranging from the use of Private Service Connect (PSC) in Apigee X to enable private connectivity and distributed replication with Apache Cassandra in Apigee Hybrid, assessing how each contributes to fault tolerance. Regulatory compliance is also addressed, highlighting the implications of the forthcoming European Digital Operational Resilience Act (DORA) for digital resilience. In addition, best practices for achieving high availability in Apigee are synthesized, including multi-region topologies with global load balancing, fault isolation, consistent backup mechanisms, and operational automation through AIOps. Finally, recommendations are provided for designing resilient API architectures, incorporating emerging trends such as data-plane extensibility using WebAssembly.

Keywords: API management, high availability, multi-region, operational resilience, apigee hybrid.

1 Introduction

API gateways have become a central component for enabling scalable, secure, and resilient distributed systems [1], particularly in hybrid and multicloud architectures where operational complexity increases significantly. In these environments, they act as a unified entry point, enforcing security policies and ensuring high levels of availability - an aspect that is especially critical in regulated sectors.

Within the European Union, the enforcement of the Digital Operational Resilience Act (DORA) in 2025 strengthens the requirements for ICT operational resilience, obliging organizations to demonstrate their ability to withstand and recover from severe technological disruptions [2]. These obligations converge with data sovereignty and data protection requirements (such as the GDPR), positioning API management platforms as critical infrastructures

Submitted: 18 December 2025
Accepted: 25 May 2026
Published: 09 June 2026

Vol. 1, **No.** 2, 2026.
10.62762/JSS.2025.584098

***Corresponding author:**
✉ José Mariano Ruiz Martín
josemarianoruiz@gmail.com

Citation
Martín, J. M. R. (2026). Scalability and High-Availability Architectural Strategies for Apigee in Hybrid and Multicloud Environments. *Journal of Systems Scalability*, 1(2), 43–49.

© 2026 by the Author. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

whose continuity must be guaranteed even in the event of regional failures.

Google Apigee addresses these challenges through two complementary deployment models: Apigee X, delivered as a fully managed service on Google Cloud, and Apigee Hybrid, which combines a SaaS-based control plane with a data plane deployed on customer-managed infrastructure [4]. While Apigee X prioritizes operational simplicity and automatic scaling, Apigee Hybrid provides greater control over processing location and data residency. This work analyzes how to design these platforms to achieve high availability and scalability in multicloud environments without compromising regulatory compliance.

2 Methodology

This work adopts a qualitative, descriptive–analytical approach, focused on the technical analysis of high-availability and scalability architectures for API management platforms. The study is based on a review of relevant academic literature, official Google Apigee documentation, and industry technical publications related to operational resilience, multi-region deployments, and hybrid and multicloud environments.

The analysis was structured around key technical criteria: (i) multi-region deployment topologies, (ii) traffic load-balancing and failover mechanisms, (iii) data replication and consistency models in distributed systems, (iv) backup and disaster recovery strategies, and (v) compliance with recovery objectives (RTO/RPO) and regulatory requirements, particularly those arising from DORA and GDPR.

To strengthen the practical validity of the study, real-world use cases are incorporated based on publicly documented enterprise deployments and industry-reported implementations, complemented by technical information published by vendors and third-party sources. Documented operational evidence indicates that many high-availability challenges do not emerge during theoretical design, but rather during real operational conditions and failure scenarios, which justifies the combination of documentary analysis with implementation-based evidence.

No controlled experiments or synthetic benchmarks were conducted. The performance data, recovery times, and traffic volumes referenced are drawn from public sources, technical documentation, or simulated scenarios. No personal data or sensitive information was used, and therefore no ethical approval process was required.

Overall, the study follows a qualitative architecture-oriented methodology in which reproducibility is ensured through the exclusive use of publicly accessible documentation, peer-reviewed literature, and documented system behavior, avoiding reliance on proprietary or non-verifiable data sources.

3 Analysis of Scalability and High-Availability Strategies in Apigee X and Apigee Hybrid

This section examines the main architectural and operational strategies for achieving scalability and high availability with Apigee, considering both Apigee X and Apigee Hybrid. The analysis covers multi-region patterns (active–active and active–passive), network and data integration approaches, disaster recovery planning, and representative use cases that illustrate their practical application.

3.1 Hybrid Architectures: Active–Active Patterns, Multi-Region Design, and Redundancy

A key decision in the design of Apigee platforms is the multi-region strategy, aimed at improving availability and reducing latency. In practice, this design is primarily implemented through active–active and active–passive patterns, each with distinct implications in terms of resilience and operational complexity.

Active–active pattern: multiple regions serve traffic simultaneously, enabling near-zero RTO and low latency for geographically distributed users. In the event of a regional failure, traffic is automatically redistributed without requiring explicit failover. The official Apigee Hybrid documentation identifies this pattern as the most suitable approach for mission-critical API platforms with global high-availability requirements [3].

Active–passive pattern: a primary region handles traffic while secondary regions remain on standby. Although this approach simplifies state management, it introduces a higher RTO, dependent on the speed of the failover mechanism and the scaling of the passive environment. It is appropriate in scenarios where consistency or cost control is prioritized over immediate availability.

In Apigee Hybrid, an active–active implementation implies that multiple clusters share a common logical data and configuration plane. Consistency is maintained through the Synchronizer component,

which continuously propagates configuration from the control plane to each runtime. Client routing is typically handled through global load balancers or geo-based DNS, allowing degraded regions to be withdrawn within seconds via health checks.

In summary, the active–active pattern is the most appropriate choice for maximizing availability and minimizing latency in mission-critical API platforms, provided that the eventual consistency inherent to distributed systems is acceptable. The active–passive approach is generally reserved for scenarios where cost efficiency or operational simplicity is the primary concern. In all cases, the multi-region strategy must be aligned with business SLAs and regulatory requirements, as regulations such as DORA mandate reduced recovery times for essential services.

3.2 Integration of Critical Components: Connectivity (ILB, PSC) and Data Synchronization

Achieving high availability is not merely a matter of duplicating instances; it also requires ensuring that the underlying network and data components can support such distribution without becoming points of fragility. This section analyzes two critical domains: (a) private network connectivity and load balancing, and (b) Apigee's distributed persistence layer based on Cassandra.

In Apigee X, multi-region connectivity relies on internal regional load balancing combined with a Global HTTP(S) Load Balancer whose backends are exposed through Private Service Connect (PSC). This approach enables the exposure of regional private endpoints consumable from the customer's VPC [5], keeping traffic within Google's private network and avoiding dependencies on the public Internet, while at the same time simplifying multi-VPC and multi-project architectures.

In Apigee Hybrid, responsibility for network configuration lies with the customer. Each Kubernetes cluster hosting Apigee requires a mechanism to receive inbound traffic. Commonly, a Kubernetes Service of type LoadBalancer is used for ingress, which—depending on the environment—provides either L4 or L7 load balancing. For example, on GKE an Internal TCP/UDP Load Balancer may be used, while on EKS a Network Load Balancer (NLB) is typically employed. A common architectural pattern is to place a global load balancer (or an intelligent DNS layer) in front, routing traffic to regional load balancers (one per Apigee cluster). In multi-region Hybrid deployments, it is essential to ensure that DNS names and TLS certificates cover all regions, avoiding scenarios in which the failure of a single region invalidates TLS termination. In addition, for internal traffic between Apigee and backend systems, corporate environments frequently rely on private connectivity: when Apigee Hybrid is deployed in the cloud and legacy systems remain on-premises, site-to-site VPNs or dedicated links (such as Azure ExpressRoute, Google Cloud Interconnect, or AWS Direct Connect) are used to guarantee resilient and secure communication.

In multi-region Apigee Hybrid deployments, Cassandra requires direct inter-cluster connectivity across regions to ensure data replication. Official documentation specifies the need to enable communication between Cassandra nodes over private networks and dedicated ports, making this a critical prerequisite for the stability of the distributed ring. Proper DNS resolution between nodes is also essential to prevent discovery and synchronization failures.

Both Apigee X and Apigee Hybrid rely on Cassandra to store runtime state (OAuth tokens, quotas, API keys, and related artifacts). Cassandra implements tunable consistency, defaulting to an AP-leaning model (prioritizing availability and partition tolerance) under low consistency levels, but capable of shifting toward stronger consistency guarantees (e.g., QUORUM or ALL) at the cost of increased latency — making it adaptable to diverse operational requirements. As a result, temporary inconsistencies may occur, mitigated through strategies such as NetworkTopologyStrategy, appropriate replication factors, and read consistency levels tuned to the operational context. This model enables active–active architectures without a central master node, but it also introduces latency and stability challenges that must be carefully managed.

Documented operational evidence shows that Cassandra must be treated as a critical component: frequent topology changes or aggressive autoscaling can degrade the stability of the ring. A recommended practice is to isolate Cassandra on stable node pools, allowing only Apigee's stateless components to scale dynamically, thereby balancing resilience and cost efficiency.

A distinctive element of Apigee Hybrid is the Synchronizer component, which continuously replicates configuration from the cloud-based control

plane to each cluster. Through this mechanism, runtimes can continue operating in a degraded or offline mode if connectivity to the control plane is temporarily lost, processing traffic using the last valid configuration. This capability is fundamental for advanced resilience scenarios and for meeting regulatory requirements.

Overall, Apigee provides a robust distributed architecture provided that network design, the datastore, and synchronization mechanisms are engineered coherently. While Apigee X abstracts much of this complexity through managed services such as PSC, Apigee Hybrid demands a higher level of architectural maturity, particularly with respect to private connectivity and Cassandra operations. With an appropriate configuration, both models enable the construction of API platforms without single points of failure, aligned with high-availability objectives and regulatory resilience requirements.

3.3 Disaster Recovery and Operational Continuity: Failover, Backups, and Resilience Testing

Ensuring high availability requires complementing active redundancy with verifiable disaster recovery (DR) capabilities. Beyond multi-region design, it is essential to define, automate, and periodically test failover mechanisms, backup and restore procedures, and operational continuity processes- particularly in regulated environments subject to frameworks such as DORA.

In active–active architectures, the traditional notion of failover is replaced by dynamic traffic rebalancing, typically managed through global load balancers or intelligent DNS. When a region becomes unavailable, traffic is automatically redistributed to the remaining regions, provided that health checks are correctly configured. In active–passive deployments, failover requires an explicit switchover (for example, DNS changes or traffic policy updates), making automation critical to minimize RTO. In both cases, a well-established practice is to keep the secondary region pre-warmed (the pilot light model), avoiding cold-start load spikes after a switchover.

Backup management represents one of the most sensitive aspects of Apigee Hybrid. While in Apigee X high availability and platform-level backups are handled by Google, in Hybrid this responsibility lies with the operator. Uncoordinated volume snapshots in Cassandra can lead to inconsistent restores in distributed environments, as observed in real-world deployments. The recommended approach is to use coordinated logical backups of Cassandra, leveraging its native per-keyspace snapshot mechanisms and storing them in external, redundant repositories (for example, Google Cloud Storage). This approach has demonstrated substantial RTO reductions compared to traditional methods. For mission-critical systems, it is further recommended to maintain backup copies in alternative regions or with different providers, in line with DORA guidelines on third-party dependency management.

In addition to data backups, it is essential to preserve Apigee's logical configuration (proxies, API products, developer apps) by periodically exporting it through management APIs. The adoption of Infrastructure as Code (IaC) enables complete environments to be rebuilt in a reproducible and timely manner in total-loss scenarios.

With respect to zero-downtime upgrades, Apigee X applies rolling upgrades managed by Google. In Apigee Hybrid, the most widely adopted strategy is the blue–green upgrade, deploying a new version in parallel and gradually shifting traffic, with automated rollback mechanisms in case of failure. Given the tight coupling with Cassandra, these operations require thorough testing in pre-production environments.

Finally, both as a best practice and for regulatory compliance, periodic resilience testing (chaos engineering) is essential. This includes simulating regional outages, control plane isolation, partial Cassandra degradations, and full restorations from backup. Such tests not only validate the architecture but also generate objective evidence of RTO and RPO, which are explicitly required by DORA [8]. Integration of Apigee with corporate monitoring systems further enables early alerting on errors and latency, laying the groundwork for an evolution toward AIOps-driven approaches focused on proactive fault detection and mitigation.

3.4 Real-World Use Cases and Comparative Results

Two representative cases-one in a public cloud environment and one on-premises—are presented to illustrate how the patterns described in Sections 2.1–2.3 can be applied in practice, and to assess the resulting operational outcomes in terms of availability, recovery, and data governance.

Case 1 (Retail on AWS): Migration to Apigee Hybrid on EKS across two regions (us-east-1 / us-west-2), as documented in a publicly available enterprise

deployment report [6], to achieve redundancy and low latency. The objective was to sustain Black Friday–type traffic peaks without service interruptions while maintaining an internal SLA of 99.98%, optimizing costs through controlled autoscaling.

Solution and results (Case 1): An active–active architecture was deployed with geographic routing and failover managed through Amazon Route 53, validating traffic switchover within tens of seconds. Scaling was applied selectively: stateless components were autoscaled using Karpenter, while Cassandra capacity was kept stable within a dedicated node group to avoid ring instability caused by node churn. For disaster recovery, uncoordinated snapshots were discarded due to the risk of inconsistencies, and coordinated logical backups of Cassandra were adopted, with accelerated restoration from external storage (Google Cloud Storage), reducing RTO compared to traditional approaches. Operationally, the platform sustained high traffic peaks (approximately 15k RPS) without downtime and met the quarterly SLA, with only limited degradation observed during simulated regional outage scenarios.

Case 2 (Public sector, on-premises): A European public agency deployed Apigee Hybrid across two national data centers to comply with data sovereignty requirements and ensure continuity of critical services (e.g., digital identity), keeping the runtime on-premises while centralizing control plane management in the cloud. The main challenge was achieving high availability under strict data residency constraints, while providing evidence of operational resilience (DORA) and data localization and governance (GDPR).

Solution and results (Case 2): An active–active deployment was implemented across two data centers using OpenShift, with Cassandra replicated via NetworkTopologyStrategy and tuned consistency levels (e.g., QUORUM for sensitive operations) to balance availability and consistency. To enforce data sovereignty, operational traffic and sensitive data remained on-premises, with masking and truncation applied to logs before telemetry was sent to the control plane and analytics systems, ensuring that full personal data never left the jurisdiction. Periodic tests—including data center isolation and complete outages—validated operation in "island mode" and front-door switchover within recovery objectives (RTO on the order of minutes) and near-zero RPO under proper replication conditions.

Synthesis: These cases demonstrate that, with Apigee Hybrid, availability depends less on the product itself and more on the quality of operational engineering: automation (IaC and CI/CD), observability, and a disaster recovery strategy validated through real restoration exercises (not merely the existence of backups). When these elements are executed with discipline—covering routing and health checks, Cassandra stability, and consistent backups-continuity and resilience objectives aligned with those discussed in Sections 2.1–2.3 can be consistently achieved.

3.5 Multicloud Interoperability and Vendor Neutrality

While Apigee provides strong capabilities for hybrid and multi-region deployments, in practice, multicloud strategies frequently involve the coexistence of multiple API management platforms across different cloud providers. In such environments, organizations may operate Apigee Hybrid in one environment (e.g., Google Cloud or AWS via Kubernetes) while simultaneously using alternative gateways such as AWS API Gateway, Azure API Management, or Kong in other domains.

This introduces several interoperability challenges, including identity and access management federation across platforms, policy consistency (e.g., rate limiting, authentication, quotas), and unified observability across heterogeneous gateways.

To address these challenges, organizations typically adopt several architectural strategies, including: (i) DNS-based traffic steering and global load balancing across cloud providers; (ii) standardization through OpenAPI specifications to ensure contract portability; (iii) API abstraction or federation layers to decouple clients from gateway-specific implementations; and (iv) service mesh integration enabling consistent behavior across environments.

In this context, Apigee Hybrid provides partial mitigation of vendor lock-in through its Kubernetes-based runtime, which can be deployed across multiple infrastructures (e.g., GKE, EKS, AKS, OpenShift). However, achieving true multicloud portability requires additional architectural decoupling beyond a single API management platform.

4 Conclusion

The analysis confirms that achieving true scalability and high availability in API platforms is not an isolated

infrastructure problem, but rather a comprehensive exercise in architecture, operations, and governance - particularly in hybrid and multicloud environments subject to increasing regulatory pressure such as DORA.

First, the multi-region active–active pattern is consolidated as the reference approach for mission-critical APIs. This model enables near-zero RTO and RPO, eliminates dependencies on a single location, and improves user experience by serving traffic from the closest region. Although it introduces complexities inherent to distributed systems - such as eventual replication, multi-site observability, and operational coordination - this complexity is manageable and, in many sectors (finance, public sector, healthcare), unavoidable. The additional infrastructure cost is clearly justified when weighed against the economic, reputational, and regulatory impact of service disruptions. Active–passive approaches are therefore limited to very specific scenarios, typically driven by strict budget constraints or particular regulatory segregation requirements.

Second, private networking emerges as a key enabler of resilience. The introduction of Private Service Connect (PSC) in Apigee X represents a significant advance by simplifying hybrid and multi-VPC architectures, overcoming historical limitations of VPC peering and reducing both latency and attack surface. PSC is not merely a technical optimization, but a paradigm shift in how platform services are consumed in a private and controlled manner. In Apigee Hybrid, although PSC does not apply directly to on-premises runtimes, the underlying principle remains the same: minimizing reliance on the public Internet, prioritizing dedicated private connectivity, and adopting a Zero Trust approach- even within internal perimeters.

A third fundamental finding is that management of the distributed datastore (Apache Cassandra) largely determines the success or failure of a highly available Apigee architecture. Cassandra cannot be treated as a secondary or "implicitly managed" component; it requires the same level of rigor as any mission-critical database. Proper configuration of replication factors, consistency levels, and node topologies, as well as isolating Cassandra from aggressive scaling or concurrent changes, is essential for overall stability. Practical evidence shows that many incidents mistakenly attributed to the API layer actually originate from poor design or operation of the data ring. Consequently, consistent backups, periodic restore testing, and advanced monitoring must be regarded as baseline requirements rather than optional enhancements.

Beyond architecture, the study confirms that high availability is, above all, an operational discipline. Automation- ranging from Infrastructure as Code to CI/CD pipelines- is a prerequisite for ensuring consistency across regions and reducing human error. Looking ahead, a natural evolution toward AIOps models is observed, in which platforms not only react to failures but anticipate them. The application of advanced analytics and machine learning to detect anomalous traffic patterns, early latency degradation, or backend saturation will enable a shift from reactive to proactive operations, aligning with the vision of continuous resilience promoted by DORA.

From a regulatory perspective, Apigee Hybrid is positioned as a key enabler for organizations with strict data sovereignty requirements. Its ability to keep runtimes and sensitive data within specific jurisdictions, combined with a centralized control plane, facilitates compliance with regulations such as GDPR and DORA- provided that appropriate processes are in place. It is crucial to emphasize that technology alone does not guarantee compliance: organizations must define governance frameworks, conduct documented resilience tests, and be able to demonstrate actual recovery times during audits. Apigee provides metrics and traceability, but ultimate responsibility rests with the operator.

In this context, the choice between Apigee X and Apigee Hybrid should not be framed as a mutually exclusive dichotomy, but as a strategic decision based on organizational priorities. Apigee X offers operational simplicity and a Google-backed SLA, reducing management overhead. Apigee Hybrid, in contrast, provides greater control, flexibility [7], and alignment with advanced regulatory requirements, at the cost of increased operational complexity and the need for teams with maturity in Kubernetes, observability, and distributed systems. In many real-world scenarios, a combination of both models proves to be the most pragmatic option.

Finally, the analysis points to a clear evolution of the API gateway data plane through technologies such as WebAssembly (WASM). The ability to extend gateway behavior with secure, high-performance modules opens new opportunities to implement advanced logic -ranging from cryptographic validation to real-time analytics- directly at the ingress layer [10], reducing

external dependencies. However, this capability will demand the same rigor in testing, versioning, and governance as any other critical system component.

Overall, Apigee X and Apigee Hybrid constitute a solid platform for global-scale API management, but they do not represent a "plug-and-play" solution for high availability. Only through a holistic architecture -integrating networking, data management, automation, observability, and operational processes- can resilience levels be achieved that meet both user expectations and regulatory demands. In the digital ecosystem of 2025, business continuity and customer trust will be increasingly tied to the robustness of API strategies [9]; in this context, Apigee, when properly designed and operated, can become a foundational pillar of that robustness.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

During the preparation of this manuscript, the author used OpenAI ChatGPT (as available in November 2025) solely to support the translation of the manuscript from Spanish into English and to improve the clarity, readability, grammar, and academic tone of the final text. The AI tool was not used to develop the research idea, create technical content, perform analyses, prepare references, or draw scientific conclusions. The author reviewed and approved the final manuscript and takes full responsibility for its accuracy, originality, and integrity.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Ochuba, N. A., Kisina, D., Owoade, S., Uzoka, A. C., Gbenle, T. P., & Adanigbo, O. S. (2021). Systematic review of API gateway patterns for scalable and secure application architecture. *Journal of Frontiers in Multidisciplinary Research, 2*(1), 94–100. [CrossRef]

[2] European Insurance and Occupational Pensions Authority (EIOPA). (2025). *Digital Operational Resilience Act (DORA): Overview of the regulation and its scope*. [Link]

[3] Google Cloud. (2023a). *Multi-region deployment – Apigee hybrid documentation (v1.14)*. [Link]

[4] Google Cloud. (2023b). *What is Apigee hybrid? – Architecture overview (v1.9)*. [Link]

[5] Shaik, S. (2022, November 24). Apigee X network connectivity using Private Service Connect (PSC). *Google Cloud Community (Medium)*. [Link]

[6] Rathnayaka, K. (2025, July 23). Real-world Apigee enterprise deployment: Lessons from a multi-region AWS implementation. *Medium*. [Link]

[7] De, B. (2023, January 10). Apigee X or Apigee Hybrid: Factors that help you decide. *LinkedIn*. [Link]

[8] Kalokyris, N., Berger, P. E., & Rosiglioni, S. (2025, February 28). Application of the Digital Operational Resilience Act (DORA): Key considerations. *DLA Piper*. [Link]

[9] Numerix. (2025, February 5). What the DORA regulation means for financial institutions in 2025. *Numerix Blog*. [Link]

[10] Zhao, H. (2024, December 5). Beyond Gateway API part 2: How to expand Envoy Gateway with WASM and external process extensions. *Tetrate Blog*. [Link]

José Mariano Ruiz Martín received the M.Sc. degree in Computer Science and Engineering and holds multiple postgraduate and master's degrees in cloud computing, big data architectures, project management, and artificial intelligence, from Spanish universities and executive business schools. His education includes a Direction in Project Management, Executive Program at the Escuela de Organización Industrial (EOI). He is currently an independent researcher and Senior Lead IT Architect specializing in cloud computing, Big Data, IA, distributed systems, and large-scale hybrid and multicloud architectures.

He has more than 20 years of professional experience in the modernization of critical IT infrastructures, large-scale data center migrations, and the design, transformation, and operation of mission-critical platforms. Throughout his career, he has worked in major international IT consulting firms, including NTT DATA, IBM, and Kyndryl, contributing to complex transformation programs for multinational organizations operating critical infrastructures in sectors such as telecommunications, banking, insurance, and energy in Spain.

His technical background spans cloud architectures on AWS, Google Cloud, and Azure; big data platforms and advanced analytics; machine learning, deep learning, and generative AI systems; API management platforms; middleware and integration systems; system optimization, performance tuning, and capacity planning; observability; and the design of analytical models and application development in cloud-native environments. His recent work focuses on high availability, scalability, disaster recovery, AIOps, and the application of generative AI to improve system reliability, automation, and operational efficiency. He has collaborated in the publication of technical books and professional literature related to middleware platforms, enterprise architectures, data platforms, and cloud-native systems. (Email: josemarianoruiz@gmail.com)