Deep Science Publishing
https://doi.org/10.70593/978-93-49910-08-9_4

DeepScience
Open Access Books

Chapter 4: Best practices in building secure, compliant, and resilient cloud-native architectures for artificial intelligence workloads

4.1. Introduction to Cloud-Native Architectures

Cloud-native architectures are characterized as a strategy for building and running applications that exploits the advantages of the cloud computing delivery model. Using cloud-native technologies, developers and organizations can build and run scalable applications in an environment that overcomes the constraints of a traditional data center and of proprietary technology stacks. Today, we can give even stronger definitions and more accurate terms to characterize these architectures, with even increased levels of common agreement. In particular, we posit that cloud-native architectures are distributed application architectures, involving a set of usual patterns such as service-oriented or microservices design, API-first design, developed using standard protocols and data in a bi-modal approach that exploits the co-evolution of both shared, common data and distributed, specifically-oriented data structures. This said and done, the landscape of cloud-native development is quickly changing, with the introduction of several enabling technologies such as containers and container orchestration systems. Indeed, during the last ten years we have witnessed an increasing democratization of cloud services, with the proliferation of several types of first-party cloud resources, from serverless computers to managed data storage and machine learning services. On the other hand, we witness the increased availability of third-party solutions from the community or third-party vendors that enrich the cloud ecosystem. These evolutions are affecting the motivations and drivers for organizations to adopt cloud-native development and have been pushing to the increasing convergence of cloud-native development practices. By adopting an increased number of cloud services or of third-party products, organizations

are actually reducing the implementation and operational burdens required for building cloud-native architectures (Celeste & Michael, 2021; Bauskar, 2025; Jorepalli, 2025).

Consequently, both types of applications deployed in cloud-native architectures must take heightened care of special requirements for resiliency, performance, and security due to their differences from typical enterprise workloads. Different types of AI workloads such as AI training and validation, and AI inferencing deployed in edge and fulfillment applications must undergo more scrutiny around resilience, security, and compliance due to their higher risk factor association (Theodoropoulos et al., 2023; Ugwueze, 2024).

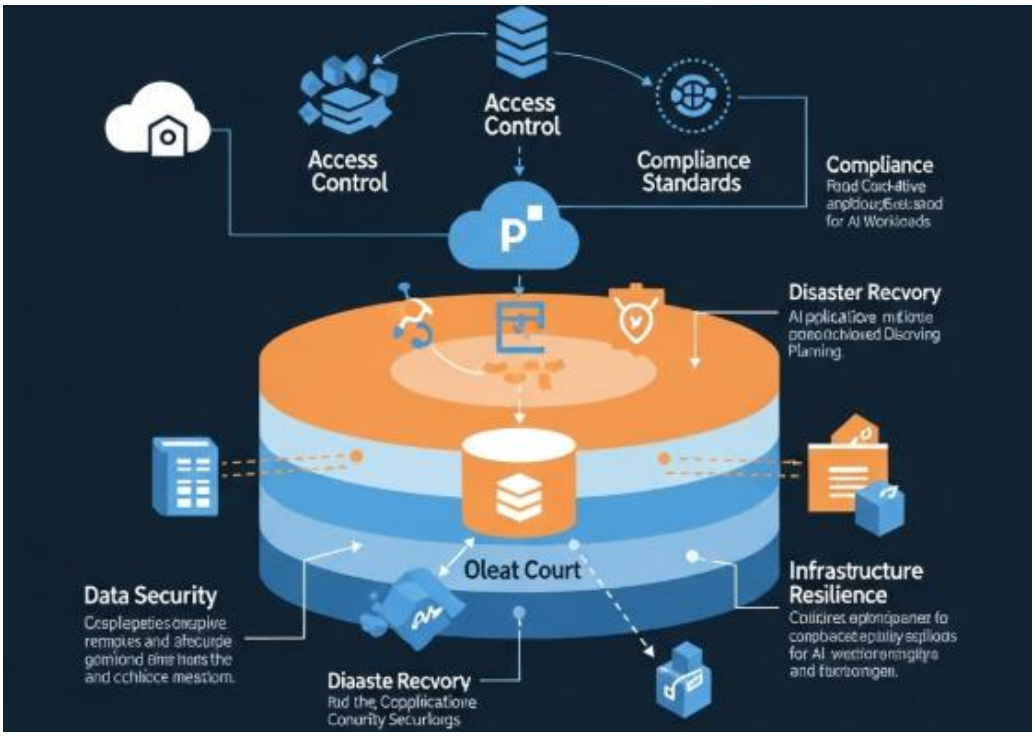

Fig 4.1: Building Secure, Compliant, and Resilient Cloud-Native Architectures

4.1.1. Background and Significance

Cloud-native architectures have taken center stage in both public and private sectors due to their capacity to empower next-generation applications, significantly enhance customer experience, optimize processes, and support the rapid processing of enormous volumes of data generated by engineering, business operations, and user activities. Cloud-native architectures are not limited to augmenting customer-facing applications. Governments and enterprises leverage this architecture to increase the resiliency of

business workflows, enhance service quality to citizens and users, and offer new capabilities and services. Financial services are working diligently to implement their cloud-native system architectures to guarantee resiliency against outages and attacks. Healthcare services are transitioning to the cloud to ease the burden on users seeking immediate capability delivery.

AI workloads have distinct processing patterns, characteristics, and requirements that differ from traditional enterprise applications. Most enterprise workloads are business transaction-driven and used to incrementally process enterprise activities and maintain a history for compliance and auditing. In contrast, AI workloads periodically leverage the abundance of data to run models and take actions in applications that are typically then either activated or deactivated—or provide insights that are sometimes sent back into business transaction applications for implementation.

4.2. Understanding AI Workloads

This section will help you get a better understanding of the cloud-native AI workloads, their differing characteristics, their foundational dependencies, and the implications. This understanding will help you make informed decisions on your AI architecture choices, required policy controls, relevant compliance mandates, and how to run a principled, risk-based approach to actually achieving compliance.

Artificial intelligence (AI) drives our next-generation customer experiences and business decisions across industries, but operationalizing AI is hard. That's because at the heart of AI is a new class of workloads that requires equal amounts of data, science, and infrastructure engineering. They're constantly evolving with the state of the art in generative AI, where we have large models that are trained with massive volumes of unstructured data, and then fine-tuned with costly, discerning human efforts. There are several additional engineering principles, practices, and considerations that go into model training workloads: workload distribution architecture, accelerated hardware, hyperparameter optimization, mixed precision math, high-throughput data movement from storage to training compute, retraining triggers, production serve-stop SLAs, serve redistribution during retraining, and monitoring observability around drift for both training and serving workloads.

But none of it matters if the data that's either being used to train or fine-tune the model, or being ingested during model production serving, is sensitive. That's the primary reason why companies cannot produce AI workloads on the public cloud secure as the on-prem infrastructure. This is especially true for the initial forecast collaborative and personalized AI models in sensitive domains. By definition, federated and differential privacy-based training and fine-tuning strategies do not work for all model types.

4.2.1. Data Encryption Strategies

Data encryption is the cornerstone of any compliance program and sound security practice. Proven industry standards in cryptography have long established encryption as the basic mechanism to reliably protect data. Organizations need to determine whether they will implement their own custom encryption solution or whether their cloud provider already can offer one that satisfies legal, operational, or cost considerations. If no managed encryption solution is available, it might be necessary to implement a custom encryption solution. Organizations need to determine and implement a suitable data encryption strategy before storing and transmitting data. Only then, with careful planning and verification of the actual implementation later on, would organizations mitigate the risk of suffering a significant data breach or sensitive data exposure. Such incidents can happen if sensitive data is stored in plaintext or, even worse, transmitted in plaintext. It is crucial to check for common cryptographic design mistakes early and often and to champion the importance of proper cryptography within the organization's culture. Data encryption has become relatively easy to deploy at scale and cheap given the computational infrastructure available, yet organizations still don't do enough. Encryption tools in cloud services still are used relatively infrequently, often because users want low latency or performance estimations in evaluating the ciphers used or when building secure channels. Planning in terms of latency or availability needs to address not only key access, retrieval, and replacement during key rollover operations but also with updates or notifications of key changes in key management services, but also potential hot spots of encrypted data blocks when the data stored is not large enough, as with other caching or temporary latency evaluation considerations.

4.3. Security Considerations in Cloud-Native Environments

“Identity and access management policies and practices that support compliance with specific regulations are often not well-defined in initial cloud use cases, giving rise to gaps in compliance for the data being processed in the cloud, especially when non-production cloud resources are not properly governed. In addition, shared resources with specific regulatory requirements may not be completely enforceable. Those gaps can lead to serious violations of compliance standards. Organizations must become very clever about developing identity and access management policies that prevent adverse consequences from occurring.”

Threats to the cloud are multi-faceted, requiring a complex but thoroughly documented set of threat models to identify what can happen to data moving into and out of the cloud. Development of these threat models must be comprehensive, taking business needs, application inheritance, operational and functional security support, operational risk for development and production support, regulatory compliance, and external threats into

account. Security policy decisions must address both active vulnerabilities and passive operations and must also recognize that inexperience in designing secure cloud architectures can threaten all data, especially personally identifiable data. Security policies need to reflect compliance with established regulations without needing to duplicate compliance work effort. Successful security policies will also provide sufficient latitude to document architecture sufficiently to indicate when specific regulations are not applicable because their conditions are not met, without stifling creativity in developing cloud-native architectures for AI."

Next, the policies and tools implemented should allow rapid onboarding of new users to allow them to work and request access seamlessly. Preferably, they would use automated infrastructure as code tools to add users and their associated access requests to the tool, connecting the disparate systems and policies in place. Tools are great for that, allowing for policy as code to be integrated into any number of workflows. They allow for the automation of the approval process, thereby enforcing policy as code and enabling simple auditing of any sensitive data access requests alongside their utilization. They also enable integration with access request processes on other systems, allowing shared information workers to have automated approvals for shared projects.

Fig 4.2: Security Considerations in Cloud-Native Environments

4.3.1. Threat Modeling

In this section we dive into Threat Modeling, something we did not explicitly mention in the previous chapter. The main reasons are they can be very time consuming to do well and are designed to be consumed by many different types of people on the project. They help to formalize the process of threat modeling each single aspect of a system and why it serves a threat involved with a specific type of work product. At the end of an analysis of this type a series of mitigations are produced that are then reviewed during the by-in process to make sure everyone is on the same page and why.

Threat Modeling is a fairly reproducible process that can help to identify weak links in your architecture, and your implementation artifacts. Using a matrix of Threat/Asset you can build a long list of various security issues with your cloud architecture, cloud native best practice, workloads, operational processes etc. These are then prioritized either by checking against a secondary severity / risk matrix or some other prioritization method. The final step usually is some form of evidence that each issue is being mitigated either in an architectural or process way such as reviews or by implementing some kind solution. This then provides a priority list of items for that cloud workload project developer team to do over their work process and make sure it's on spooling in either validation, or the implementation and is focused towards a most severity statement at the time of the project.

4.3.2. Identity and Access Management

We have previously discussed how important it is to decouple policy from code to prevent vulnerable code from compromising underlying sensitive data and system configuration. This is also a best practice in enforcing access controls, specifically in separating the process of requesting access from that of granting it. Control systems or product-owner teams need to first clearly define their security and compliance policies, alongside the type and sensitivity of data and system configurations that risk being compromised. The policies should exclude as much of the development, DevOps, and DevSecOps teams as possible and create rules for requests for every kind of sensitive data access. They also need to allow easy ways to exchange security, compliance, and configuration-related data without compromising security or sensitive information. Popular user management systems for this purpose are great tools, but need to be additionally managed for secure use by developers and infrastructure teams as well as Secrets Management solutions, which enable secure decoupling of access and code from common development and application integration tools.

4.4. Compliance Frameworks for Cloud-Native Solutions

A major challenge with cloud architectures hosting enterprise workloads and, possibly, the most crucial step for the deployment of production applications are issues of data security. Data provides the competitive advantage to organizations or the value-added services users seek. With its storage, management, and processing having moved to the cloud, its loss or compromise can pose dire consequences for organizations. Incident and risk management frameworks and recommendations can help organizations use cloud-hosted infrastructure sensibly. While incident and risk management provide the common basis for enterprise data security, industry- and domain-specific regulations supplement these frameworks with their focuses on sensitive data types and specific classes of organizations. Whether merchant, healthcare provider, or financial entity, many organizations have some government oversight about how they handle different types of sensitive data. Mishandling of sensitive data can come with long-lasting consequences. Fines, organizations being barred from operation, license revocation, etc. Compliance for organizations can be a problem, having to prove that certain predicates are always true. Enforcement of generalized security frameworks is not feasible since those do not touch upon the unique questions raised by the large variety of domains of the regulated organizations.

Moving organizational workloads and, in particular, sensitive data to the cloud raises the question of compliance. Whether the organization has a certifiable system to handle sensitive data remains a constant question for regulators. This does not mean that cloud solutions do not lend themselves to certification and approval. Rather solutions offering the claimed properties must be carefully constructed. Cloud-native solutions additionally need to take into account specific configurations exposing user-sensitive data or reducing the visibility of the cloud infrastructure used for workload-hosted applications that do operate on sensitive data.

4.4.1. Overview of Regulatory Standards

Technology globalization and economic liberalism have made markets, consumers, resources, and services more interconnected and accessible; however, they have also increased information sharing across parties that may not have long-term operational relationships. These are the backdrop of the emergence of new security threats and the challenging task of an organization to become compliant with regulatory laws and standards, so that customer and market trust are built and maintained. These tasks become more challenging when cloud-native architectures are adopted, because they lessen the operational boundaries of data and computation resource frameworks across regions and markets.

A regulatory standard is a set of guidelines to ensure compliance with a particular standard within a certain operational area. Regulatory standards map closely to groups of domains, like accessibility, information sharing, and secure data governance, who are affected by specific market, industrial, or government-oriented sector characteristics and specific security property objectives of systems handling sensitive information such as health care and financial systems. Noncompliance, whether deliberate or not, may result in punishment or penalties in the form of fines, revocation of right to operate business, or jail sentences for involved human actors. The use of regulatory standards should not be confused with best-practices templates intended for design, development, management, manufacturing, and operation but that are not tied to legislation or threat and risk tool stakeholders' approval.

Several areas have been targeted by compliance frameworks across the world. Regulatory frameworks define how private data should be handled to ensure privacy of data subjects. Specific regulations specify the security and privacy of healthcare data. Federal standards are established to provide a foundation for assuring the quality, security, and interoperability of systems used within government agencies. Regulations protect records of students attending colleges and schools. Security requirements of information systems supporting the federal government are also outlined.

4.4.2. Implementing GDPR in Cloud Architectures

General Data Protection Regulation has many requirements that directly relate to Security and Cloud Security Frameworks. Technologies, Practices, and Mechanisms that align with various Cloud Security Standards, Privacy Frameworks, and Compliance Mandates significantly ease compliance. Let's consider Data Providers are ingesting Only Health Data. GDPR allows the Data Provider to submit Data to the Data Ingesting and Storing Layer without identifying the Data Originator. Data Providers can get their Data encrypted when they reach the API/Streaming Endpoint which terminates HTTPS and provides Cloud-Managed Secure Sockets Layer or virtual private cloud-level encryption. On the Managed VPC, Private buckets receive Data at ingested Periodic Intervals, which are encrypted with Unique Keys per Batch of Data as Local Data Encryption. These unique keys are sent to the Secret Management Service with expiry intervals corresponding to the Data Ingestion Frequencies. This Security Model satisfies the Data Minimization Principle of the GDPR - the Data Provider can choose to not include the Data Subject Identifiers in exposed Payloads.

The Service is then Built around this User-Centric Design. The Cloud Key Storage service uses the Unique Keys for Batch Decryption upon each Data Processing cycle. Crypto Module Checks if the Key is still Valid; If not, the current crypto module generates a Key Rotating Request with the Data Management Service. Data

Management sends a Batch Decryption_Not Available Notification, which is hosted on a Temp File Store Service Bucket. The Service then Breaks the Computation Pipeline, Victimizing all Functions List the Bucket. Each Function at a Set Time Interval, checks if the Notebook File/ Every Model is Stalled. If Yes, It Processes all Resources with the Notify. It then exits and is Timed Out, Shutting down and Calling Batch Processing Function Removal and Restoring of Temporary Files.

4.5. Building Resilience into Cloud-Native Architectures

Cloud-native applications are often built for elasticity and scalability. However, outages and disasters can still happen in cloud environments. For example, distributed NoSQL data stores have been known to lose at least 30 percent of traffic due to outages. Furthermore, cloud service providers undergoing maintenance of the datacenter region will cause a traffic drop close to zero on certain availability zones. A major cloud service provider suffered a 90-minute long outage of its system that was supposed to have four-nines availability. Such availability features are not unique to one provider. Other providers also offer global cloud services that provide such availability guarantees. Cloud service customers cannot load balance across providers to mask a provider-wide outage due to the practical issues involved in multi datastore transactions and data consistency.

For enterprise class cloud-native workloads, you have to ensure resilience in all aspects of your design. Simply trusting cloud providers will ensure resilience is likely fraught with some issues, as complexity increases the likelihood of bugs in service software stacks, faults may affect multiple datacenter regions simultaneously, and errors may be introduced even from the defense-in-depth design practice. Designing systems for higher availability usually incurs extra costs, although distributor software developers can make these available. We will review design concepts to increase the resilience of your workloads on cloud-native architectures. In this section, we will review best practices related to high availability and disaster recovery, especially with regards to enterprise business requirements. Outages or disasters can happen without warning, such as an outage affecting customers relying on centralized email and support ticket services. Using storage services with strong consistency behavior could offer some protection, but could be less performant and more expensive than competitors.

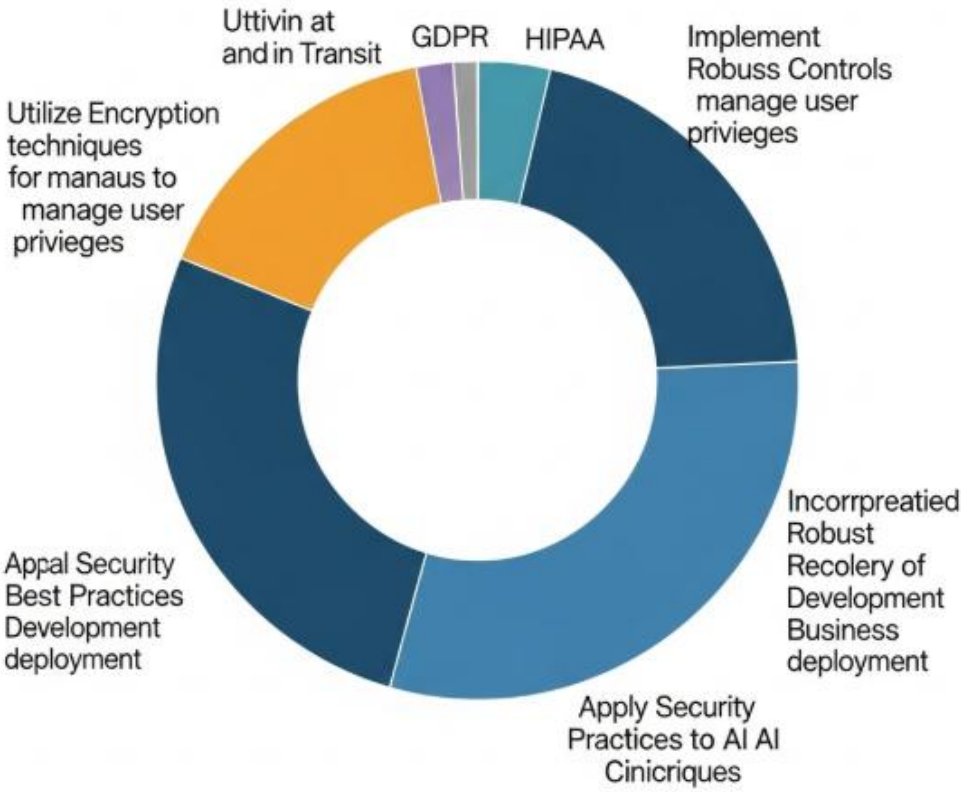

Fig : Resilient Cloud-Native Architectures for AI Workloads

4.5.1. Designing for High Availability

High availability refers to a system's ability to remain operational for the majority of business hours without prohibitively expensive use of redundant hardware. A good practice of ours is to build core infrastructure services with 4 nines availability. To achieve this goal, we often put backup redundancy around single points of failure. If one node fails, there is another node that can seamlessly take over without affecting routine operations. Our database sharding architecture will have shards replicated on several nodes.

Within our infrastructure services, we will accept short periods during which the availability of some components falls below our threshold. For example, it is normal for a backend service to be unavailable for less than a minute if it is being upgraded or a few minutes during a failover and recovery stage if a node fails. While subsystems can experience occasional short periods of low availability, the infrastructure services as a whole must remain continuously operational. Architectural wisdom requires that any sharded backend service has at least one replica shard that is in sync with the primary

shard. We will not deploy new features or bug fixes that have the potential to take a shard offline for an extended period of time without first ensuring such a fallback mechanism is in place. In addition, we will not starve replica shards from updates. Log replication throughput must be sufficient to keep the replica up to date with the primary while not being so high as to overwhelm the storage subsystems for the log.

The replica of a shard needs to be completely idle or under very low load during the switching over of shared ownership to minimize the switching over time. To achieve these goals, we deploy a shard that is known to support extra usage. We would not deploy an embedded or a very phase-shifted backend around the shard, as this would adversely affect the load on the primary and on its replica. In addition to redundancy and fallbacks, we also put a stress on predictive signaling of impending problems.

4.5.2. Disaster Recovery Planning

This section focuses on disaster recovery planning (DRP) for AI workloads with the underlying assumption that discussion on maintaining AI workloads in an "Available" state is existing infrastructure-based high-availability based capabilities that are brought to bear on service and component uptime over time. DRP focuses – going further – on service recovery timelines, i.e. duration from failure detection until recovery. Such discussions are critical to defining acceptable RTO/RPO for different AI components, the pace of business, and the choice of tools and tech industry has made to determine the current state of the lifecycle/service on failing over and recovering to software, service, and component close to real time – in turn affecting budgets for building our DRP strategy and creating our game plan – strategy and runbook – for DRP testing and iterations.

Existing DRP practice tends to be focused more on the domain of computing be it for classical or ML workloads, but that also has relevance for the rest of the AI software lifecycle components, namely, for the data ingestion pipelines for training, periodic/drift retraining, and inferencing either in production or on-demand for scorecarding, etc. These act as the data fabric for the communication between microservices and services for providing the intended operationalization and are in their own right AI operationalization lifecycle extensions that need to be integrated into DRP since they interact bi-directionally with other aspects of DRP. DRP such as actual systems on which code runs or shared storage or network infrastructures for code execution or synchronous communications during run for AI inference processes are being endangered or needing rational delivery design for failure prevention and mitigation strategies with DRP goals are extended through the addition of security and business need state information.

4.6. Conclusion

The Cloud Computing Security Reference Architecture provides a framework to secure cloud computing while addressing key challenges including managing users, maintaining data integrity, and securing interfaces in cloud environments. Building upon this framework, the Best Practices in Cloud-Native Architecture describes models for overcoming these challenges when building and deploying cloud-native architectures for foundational services, emerging accelerator services, and for specific use cases and domain-specific needs centered around AI workloads. These best practices recognize a need for secure foundational cloud-native architecture designs and patterns which support the usability of high performance accelerators like Graphics Processing Units and Tensor Processing Units in secure and compliant environments.

Accelerators have undergone rapid advancements within cloud-native architectures for AI workloads, enabling discoveries and key investments in critical domains. Additional momentum from start-ups, well-established architects and the media have catapulted cloud-native AI workloads to the forefront, securing interest at the highest levels of global leadership. The latest versions of developer frameworks have embraced models specifically designed for Large Language Model workloads like diffusion models and the foundation models which underpin novel capabilities like AI-augmented creativity and knowledge. The intuitive use and access to cloud-native generative AI workloads for voice, video, art, and text creation drive investment and engagement from developers and technologists alike. While cloud-native LLM workloads stand at the forefront, making decisions based on trade-offs amongst capabilities, costs, responsiveness, and quality is critical. The architectural best practices in building secure foundational and accelerated cloud-native architectures serve as a reference to drive these decisions.

4.6.1. Emerging Trends

Cloud native computing offers better return on investment (ROI), upsurges productivity with increased agility and innovation velocity, and convenient and portable workloads to move as and when needed. With Artificial Intelligence/ Machine Learning (AI/ML) occupying center stage due to its increasing adoption across industries for a variety of use cases in Cloud Native across the spectrum of build, deploy, and run. They occupy the Cloud Native Landscape with a significant footprint across the stack of Open Source technologies. The use of open source technologies for Cloud Native AI has several advantages. It democratizes access to powerful tools and technologies and fosters collaboration and sharing of innovations. It also leverages strengths and experiences from the developer community. With the tooling maturing on multiple fronts – We have Cloud Native for AI/ML, MLOps, and CI/CD for GenAI. Exploratory, development, deployment, and lifecycle management for GenAI with various types of models from

foundation, encoder-decoder, latent diffusion, text to 2D, text to 3D, multi model and multimodal; and applications manifesting in a spectrum from chat, agents, image generation, audio/text generation, code generation, and hallucination management.

The use of Cloud Native technology by tech giants for their GenAI models and applications and as a productivity layer for developers and users is leading to a transformation in the way enterprise applications are developed, deployed, and run. All of this is happening with Data Locality, Privacy and Security becoming first class citizens. As this space is rapidly evolving, the opportunities are immense for ISVs, SaaS providers and Internal Developers / DevOps. The tooling available today and being built for tomorrow is enabling organizations to harness the capabilities of Cloud Native and GenAI to maximize business value.

References

Ugwueze, V. (2024). Cloud Native Application Development: Best Practices and Challenges. International Journal of Research Publication and Reviews, 5, 2399-2412.

Jorepalli, S. K. R. (2025). Cloud-Native AI Applications Designing Resilient Network Architectures for Scalable AI Workloads in Smart Education. In Smart Education and Sustainable Learning Environments in Smart Cities (pp. 155-172). IGI Global Scientific Publishing.

Theodoropoulos, T., Rosa, L., Benzaid, C., Gray, P., Marin, E., Makris, A., ... & Tserpes, K. (2023). Security in cloud-native services: A survey. Journal of Cybersecurity and Privacy, 3(4), 758-793.

Celeste, R., & Michael, S. (2021). Next-Gen Network Security: Harnessing AI, Zero Trust, and Cloud-Native Solutions to Combat Evolving Cyber Threats. International Journal of Trend in Scientific Research and Development, 5(6), 2056-2069.

Vivian, M. Secure and Compliant AI Workloads Across Cross-Border Cloud Platforms.

Bauskar, S. R. (2025). Optimizing Multi-Cloud Environments Advanced Database Technologies for Scalable and Resilient Education and Training Systems. In Integrating AI and Sustainability in Technical and Vocational Education and Training (TVET) (pp. 189-206). IGI Global Scientific Publishing.