WHITE PAPER

arm

Evolving Edge Computing

Contents

1 Why Evolve Edge Computing?

Edge computing is a term that has been in use for a long time. Throughout the industry, there are many references to edge and many pre-conceptions about what that might mean. The term 'edge' is typically used for devices that exist on the edge of a network and can cover a plethora of use cases, ranging from the router in your house, a smart video camera surveying a parking lot, to a control system managing a robot on a production line in a smart factory. It is hardly surprising then that 'edge' is a confusing term with so many use case examples to choose from.

So, what is happening that means that Arm is calling for an evolution in edge computing? This paper examines the convergence of several market trends that present new challenges and opportunities in this space and require us to rethink the way forward.

Firstly, edge devices are becoming connected to cloud services such that they are generally located close to the source of data. In turn, they generate insight that feeds new digital transformation services that are hosted in the cloud. In this context, we define 'the cloud' as being a centrally located compute resource, typically datacenter based, running high-level business services.

These services consume insight (data) from a vast number of remotely located edge devices. As this cloud-connected trend accelerates, we see a deepening of the 'relationship' between cloud and edge devices, such that the centrally located services consuming the data have an ever-increasing amount of control over the edge devices with the aim of driving ever high levels of efficiency in how these networks are deployed. Although the edge is distinctly different to cloud compute resources, we expect to see developers increasingly being able to develop applications at a high level that are 'pushed out' to the edge, enabling data insights to be refined and tuned for very specific use cases.

For the purposes of this paper, we focus on 'frictionless development' as a term that embraces high-level workloads with hardware abstraction, while allowing the developer to exploit the full benefits of the underlying hardware.

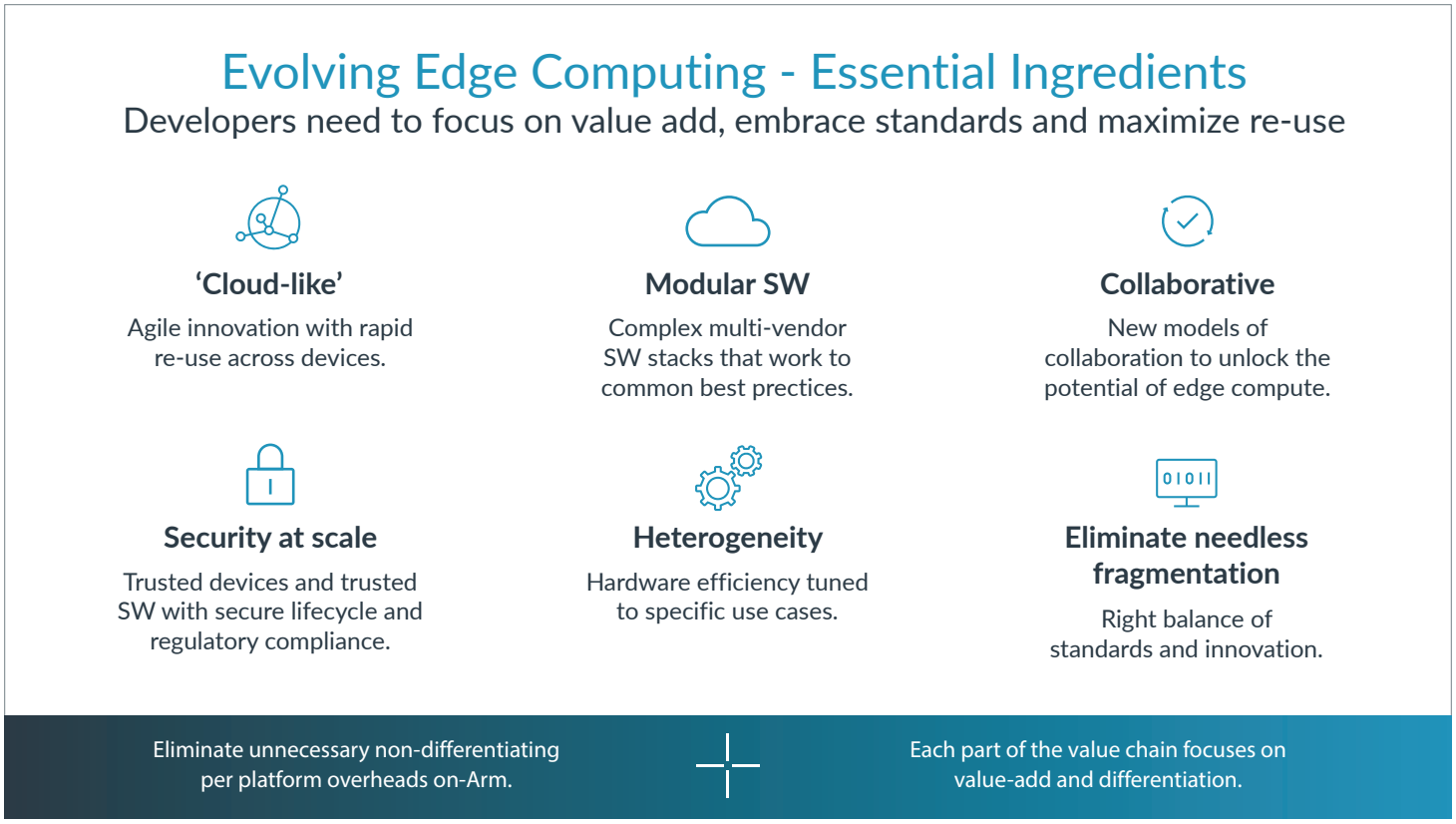


FIG. 1
Evolving Edge Computing – Essential Ingredients

Secondly, we see a huge shift in the market to driving insight through artificial intelligence. Typically, this means pushing AI models out to edge devices so they can deliver the insight needed for business-level services.

Finally, these devices need to be managed in a secure way. As described later in the paper, emerging regulations mandate software security and guaranteed updates, making it increasingly important to consider the full security model of edge computing. When deployed at scale, edge devices are performing a critical role in the delivery of high-value services and making them more vulnerable to bad actor manipulation.

Secure identity and secure lifecycle management are critical considerations for a best-practice edge computing approach.

In the context of this paper, edge computing and subsequently, edge AI, typically encompasses compute-rich devices that can be programmed in high-level abstracted languages that make them accessible to a broad range of developers. From an Arm architecture perspective, this currently relies on Arm Cortex-A as the principal processing element. The ability to support compute-intensive workloads and rich operating systems, including Linux, allows products based on Cortex-A based to address the widest possible set of use cases.

We can expect many edge AI use cases to be power-consumption and cost sensitive, so there is an ongoing need to balance these aspects across the ecosystem. With this in mind, we also look at the need for heterogeneity, i.e., moving compute-intense workloads to specialist types of compute that offer a more balanced approach.

2 Vision

As use-case complexity and the scale of smart connected edge devices deployment grows, almost exponentially, some technologies used in cloud-native [1] solutions are being embraced in edge computing. We see a future that empowers the next generation of application developers with frictionless 'cloud-like' development flows that fuel collaboration, maximize re-use, accelerate time to market, and reduce the total cost of ownership on Arm. The rapid advancement of AI use cases is expected to fuel most of the growth in the edge (or edge AI) market, with inference being deployed at scale across multiple architectures.

This rapid shift in edge compute represents several challenges, which Arm believes necessitate an evolved, best-practice approach to edge computing to enable the intelligent edge through:

- **Re-use of software components:** Applications are a key differentiator. The availability and re-use of the core underlying stack is critical as developers wish to focus on differentiation and maximize re-use elsewhere.
- **Embracing heterogeneity through abstraction of the complexity of differentiated hardware with a common software ecosystem:** Devices are use-case optimized based on cost, power, and performance, driving hybrid device architectures (CPU/GPU/NPU/ISP, and so on). The common software ecosystem needs to provide an integrated view of the system with levels of abstraction that reduce complexity.
- **Generic abstracted development flows that fuel collaboration, speed time to market, lower total cost of ownership and maximize re-use:** Use cloud-native derived methodologies, such as continuous integration/continuous deployment (CI/CD), to develop, test applications, and deploy efficiently to target hardware. Development flow efficiency is key in both the development phase, as well as in long-tail maintenance once the application is deployed.
- **Security at scale:** This is achieved through frictionless secure lifecycle management and regulatory compliance to reduce total cost of ownership for the deployed lifetime of the device.

2.1 Edge Versus Cloud

Beyond hardware constraints, there are several key differences between edge [2] and cloud as operational environments. Edge nodes and devices are purpose-built with different cost constraints, resulting in many different configurations deployed over multiple generations of underlying hardware components.

Nodes differ in hardware resources, such as CPU architecture, micro-architecture, core count, memory, storage, connectivity (latency and bandwidth), peripherals, and accelerators. Additionally, edge nodes and gateways are more likely to require dynamic frequency scaling (either because of battery conservation or thermal throttling). This high degree of hardware heterogeneity has implications on deployment, where multiple versions of an application may be required to support device differences.

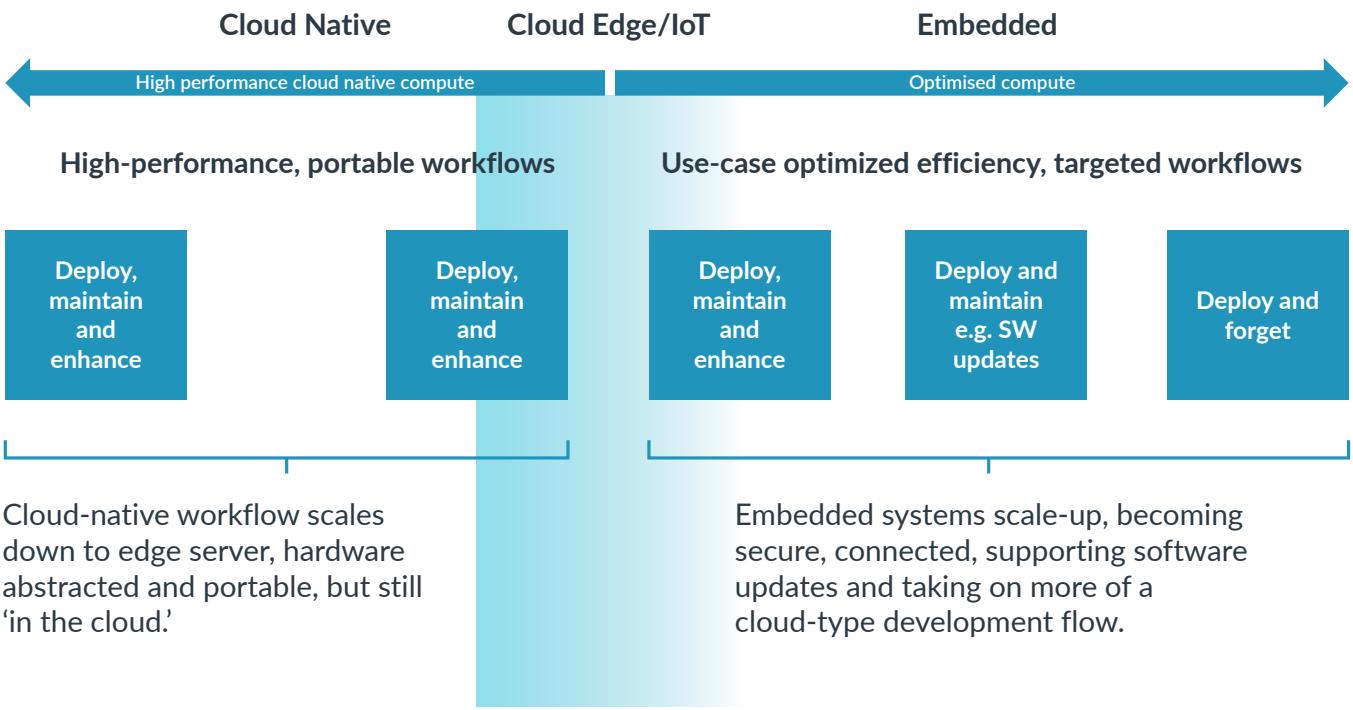


FIG. 2
Cloud transition to Edge

Organic growth and physical constraints, such as location and difficult or costly replacement, require multiple generations of nodes to coexist, leading to different SKUs of the device supported with the same application software during the system's lifetime.

The edge is likely to have a higher data storage and transmission cost compared to the datacenter. Few edge devices are likely to have

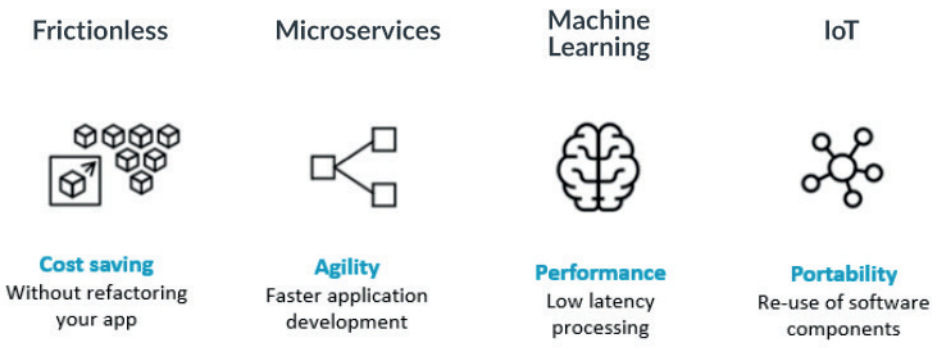
high-bandwidth network connections, constant connectivity is not necessarily a given, and transferring data to and from thousands of edge gateways is expensive. Virtual machine and container images magnify the data movement cost, amounting to close to a complete distribution download per application, due to existing packaging.

While layered container images are intended to reduce this overhead, third-party application packaging makes underlying layer re-use unlikely. For example, Arm developed a prototype healthcare application with machine learning, which used 17 Docker images, occupying about 2.3 GB of storage. Deploying this application to thousands of nodes over metered cellular networking would not have been practical. For this reason, as well as the somewhat more constrained compute capability, we do not see a pure 'cloud-native' deployment to edge computing devices, but rather a frictionless 'cloud-like' model which is aimed at delivering cloud benefits, such as portability and abstraction, in a more hardware-constrained environment.

2.2 Why 'Cloud Like' in Edge Computing?

The efficiencies resulting from minimizing the operational burden of developers, administrators, and users in cloud computing have led to other segments evaluating the use of technologies originating from the cloud in other environments.

FIG. 3
Benefits of Cloud Native



The driver behind this movement is based on the law of economics, namely that the cloud-native model of abstraction has been shown to accelerate time to market and save costs. Continuous development [1] is a major component of achieving a faster time to market. These advantages are rooted in several core properties of cloud-native technologies:

- Portable, hardware abstracted.
- Consistency across any installation/deployment.
- Timely updates without complex re-integration overheads.
- Speed time to market and maximize re-use.
- Fast application development times.
- Remove unnecessary industry fragmentation to eliminate siloed per platform costs.

2.3 What's Changing in Edge Computing?

Digital transformation across industries continues at pace, bringing with it new innovative business services and never-before realized efficiencies. From building the next wave of Giga Factories to low-carbon, energy-efficient cities, and the electrification of transport, a common theme underlies it all—data insight at a scale never-before realized.

Traditional views of data insight are built around a datacenter 'cloud centric' model. In this scenario, sensor data is shared with the cloud, in turn deriving insight at scale through techniques such as AI, to deliver the desired business and efficiency outcomes. The challenge comes with scale and the sheer number of connected devices, and corresponding compute drives the need to put processing close to the source of the data. Factors such as latency, power consumption, cost, privacy, and connectivity, all drive the need to deliver ever-more sophisticated edge computing, rather than simply **pushing** data to remote cloud-based server.

As well as frictionless compute where it is needed, other factors are required to meet the scale and demand of edge AI growth over the next few decades.

Scaling data insight and value: Simply connecting devices to the cloud brings neither scale, nor operational efficiency. Traditional cloud datacenters deliver generic compute for use by business-level applications. Conversely, edge devices form the 'real-world interface' and deliver massive insight at scale into those cloud-based services platforms. How insight is enabled at the edge and how these connected devices are securely managed becomes a critical success factor in scaling new applications and services.

Security at scale: There is growing regulation around the management of electronic data and products. The European Cyber Resilience Act, the UK Product Security and Telecommunications Infrastructure Act and the European Renewable Energy Directive are prime examples. With similar legislation progressing in the US, the regulatory landscape could pose a risk of financial penalties and lost reputation for those who fail to manage the security of digital hardware and software adequately across device lifecycles. Trust therefore becomes a significant factor in enabling scale. Edge devices do not benefit from being in a traditional datacenter setting and are installed wherever they are needed. Unlike traditional enterprise datacenter models where servers are deployed in secure locations with highly managed security, in edge deployments, we see very different deployment and threat models. Edge devices must be deployed in a wide variety of locations, with highly variable security threats, e.g., publicly located, susceptible to physical attack, connecting via public networks, to name just a few. Establishing the right level of security and trust for edge devices is critical to scale applications and realize the business benefits.

Operational efficiency: As we scale out edge compute, operational efficiency becomes a key consideration when considering total cost of ownership. We can think about this in two ways: Firstly, the development cost to create the application or service, and secondly, the operational or running costs once the service is deployed. Since edge compute devices typically have a long lifetime (5 to 10 years, or longer) the total cost of ownership becomes a critical consideration. The costs incurred to operate a device include factors such as power consumption (linked to running costs and carbon efficiency), as well as device maintenance costs related to managing software updates and overall product lifecycle. As the deployment of devices scales and use case complexity grows, device vendors and service providers increasingly look to **optimize** operational efficiency.

Agile innovation: Our traditional view of cloud compute is built around agile development. This delivers tremendous efficiency both in terms of cloud accessibility to a vast number of developers via consistent and hardware abstracted development flows, and an agile mindset in product development. As use cases become more complex, developers are looking to embrace the benefits of 'cloud-like' innovation in edge use cases. Examples include abstracting hardware differences as much as possible and supporting an agile development flow that facilitates rapid innovation, fast virtual prototyping and continuous development and improvement (CI/CD flows).

2.4 Challenges to Overcome

As we have seen, the demand for edge compute is relentless, but so too is the need for efficiency at all levels if we are to realize the vision at scale. Traditional IoT-connected devices that we see today go some way to solving these challenges, but a step change in how edge devices are enabled must

happen across all industries. We can summarize the key challenges as follows:

Develop a 'cloud-like' mindset at the edge: The traditional datacenter model of 'write once and run anywhere' does not map directly to edge devices for practical reasons, however elements of that model are critical for an effective edge computing evolution. Edge devices tend to be application specific (e.g. a smart camera) but must embrace elements of frictionless development for specific benefits. As we think about edge computing as an extension of the datacenter, we need a whole new mindset in terms of how accessible these edge devices are to developers, and how they support agile development, virtual prototyping, and continuous improvements. To deliver this vision also requires a significant mindset shift for traditional embedded developers. Gone is the traditional 'linear' development flow of specifying, implementing, testing, and deploying applications. Instead, we shift to CI/CD/delivery flow to speed time to market, maximize software re-use and ultimately reduce cost. To do this, the market must build common abstracted programming models to open the accessibility of edge devices to developers across platforms, abstracting complexity and limiting hardware dependencies exclusively to where these add value, such as for performance and power optimization.

Security and privacy at scale: A bedrock of scaling the cloud out to the edge is ensuring robust security and privacy. Building devices that have a trusted and consistent approach to security is critical for their lifecycle management and ensuring trust around the device, connection, software lifecycle, data, and services. With software stacks becoming increasingly complex and multivendor, we see greater a need for composable software, whereby each party owns only the portion of software that they care about. Within this model, each software component essentially has its own secure lifecycle. Underpinning this is the need for consistent platform security capabilities, such as secure boot, secure updates, secure storage,

and trusted crypto. How each of the software components can access these secure platform services to manage their lifecycle is critical.

Eliminate needless fragmentation: Needless fragmentation holds back innovation and slows the pace of adoption at scale. It is therefore essential to seek out commonality that removes needless non-differentiation so the supply chain can focus only on the differentiation that adds value to their business and the market. An obsessive attention to efficiency is needed both in the development of the device, as well as the operational costs.

A modular approach to software deployment: Fragmentation challenges extend to software as we consider the increasingly complex use cases for edge devices. It is commonplace for multivendor software stacks to run on an edge device with many third-party components needing to come together and interoperate. Increasingly, end-market deployments care about what software is running on edge devices. Fleet managers, for example, want to know what operating systems are deployed, what security patches are pushed out, and where different software assets are coming from. The desire for choice, coupled with growing complexity, is driving the need for modular, interoperable software that can be maintained throughout its deployed lifetime.

Balance standardization and differentiation: The market must embrace standards and commonality where necessary to speed time to market, reduce total cost of ownership, and eliminate needless fragmentation. Collaborating on Arm can bring the right level of standardization, while allowing hardware innovation and differentiation to thrive. There is no single 'recipe' for edge devices from an Arm platform point of view. Instead, we consider ***'the set of hardware and software interfaces needed to minimize the cost of booting, running, and maintaining operating systems and other system software through the lifetime of the device'***.

Benefits of this approach include:

- Reduces time, cost, and effort from getting software to install and work for device lifetimes.
- Removes non-differentiating cost from the ecosystem.
- Allows the ecosystem to invest more time and money on work that adds value.

Today, initiatives like **PARSEC** for standardized hardware-abstracted security services are becoming essential, as is a consistent approach to security, which is provided by **PSA Certified**. Plus, through **Arm SystemReady**, we look at how operating systems are supported on edge devices as a critical factor, alongside the need to offer and maintain operating system distributions on devices for their complete lifecycle, while eliminating per-platform porting costs.

Heterogeneity in edge AI: When thinking about cloud native, we imagine containerized compute workloads that can run in a fully portable manner in cloud datacenters. As we established early in this document, edge computing tends to be application specific and optimized for certain workloads and power/performance budgets. Over the last few years, we are seeing a deepening trend for 'accelerated compute,' whereby hardware acceleration is applied to common and compute-intensive workloads. Accelerated compute takes many forms but principally falls into two areas:

01 In-line acceleration that occurs as part of the CPU operation (e.g., Arm Scalable Matrix Extensions).

02 Offload acceleration (e.g. hardware that sits alongside the CPU, such as an NPU, bproviding heterogeneity in the programming model).

Accelerated compute is used to improve performance, reduce power consumption for specific workloads, or sometimes both. Examining how developer experiences scale across heterogeneous platforms is essential to avoid needless fragmentation and siloed developments becoming deeply entwined to specific hardware variants. As we look towards the evolution of edge devices as outlined in this paper, the partial decoupling of hardware and application as a trend moves us toward an 'app-like' model that facilitates the abstracted deployment of microservices. Making use of the 'goodness' of the underlying heterogeneity with the right level of abstraction across platforms becomes critical for these services and maximizes developer reach.

Industry collaboration models need to evolve: Moving toward a frictionless cloud-like mindset at the edge is a clear example of how the industry can work collaboratively to address the opportunities and the challenges we have considered and create an environment in which everyone can differentiate and thrive. When we imagine the cloud-like edge future, we know that it can only be successful through collaboration. As we have seen in other markets, such as telecoms, industries come together around standardization and interoperability. For the evolving edge, this interoperability is not simply to check standards compliance, but also to foster software interoperability across complex multivendor stacks to eliminate needless differentiation, reduce the total cost of ownership, and achieve deployment at scale.

2.5 Summary

Traditional cloud native in the datacenter is a highly controlled compute environment, but we cannot say the same of 'edge.' How compute scales outside of the datacenter is critical in allowing evolved IoT use cases to scale. This includes areas such as security, secure software lifecycle, and application deployment.

As a result, we can see why compute needs to move outside the datacenter (in a complimentary way) and the net effect in three key sub trends:

- **Use cases are getting more complex and the corresponding software stacks are also getting more complex and multivendor:** Traditional embedded development approaches (develop, test, deploy) are not working and no longer scaling, so the market is shifting toward more agile CI/CD flows with rapid prototyping (virtual platforms) and the rapid deployment of portable and modular 'cloud-native-like applications and services.' Developer experience needs to become frictionless in these complex systems.

- **Fleet management at scale:** How are devices connected, managed, and maintained? When we look at the sheer scale of IoT deployments, the management/control of these devices needs to be efficient and at ultra-low cost. Managing the secure lifecycle of the device, secure updates, and trusted credentials is critical, as compute is no longer in a trusted environment.

- **Total cost of ownership:** All edge devices that are feeding cloud services must fit within an operational 'budget' that fits the business models of the overarching services (think about the security camera example). Major factors for total cost of ownership include installation costs (e.g. zero-touch self-installation), running costs, such as power consumption, and cost of back haul (this means minimizing per-device power and reducing the backhaul data, but performing more inference at the edge). Software lifecycle costs are also a major consideration, for instance, the cost of maintaining/patching the device throughout its lifetime.

As you can see there are multiple success factors that the industry must address to enable a dynamic edge that it is fit for scale. When we consider the transition of compute to the edge to 'enhance' cloud compute, we must be mindful of those wider factors.

Bibliography

[01] B. Mitchell, “Cloud Native Software Engineering,” arXiv, vol. 2307, no. 01045v1, 2023.

[02] LFEDGE, “Open Glossary of Edge Computing,” State of the Edge, 2019.

[03] S. Foster, “what-is-continuous-development,” 2023. [Online]. Available: https://www.perforce.com/blog/kw/what-is-continuous-development. [Accessed 2023].

arm