White Paper

Navigating the Challenges of AI Infrastructure Design

Balancing Power, Latency, Reliability, and Data

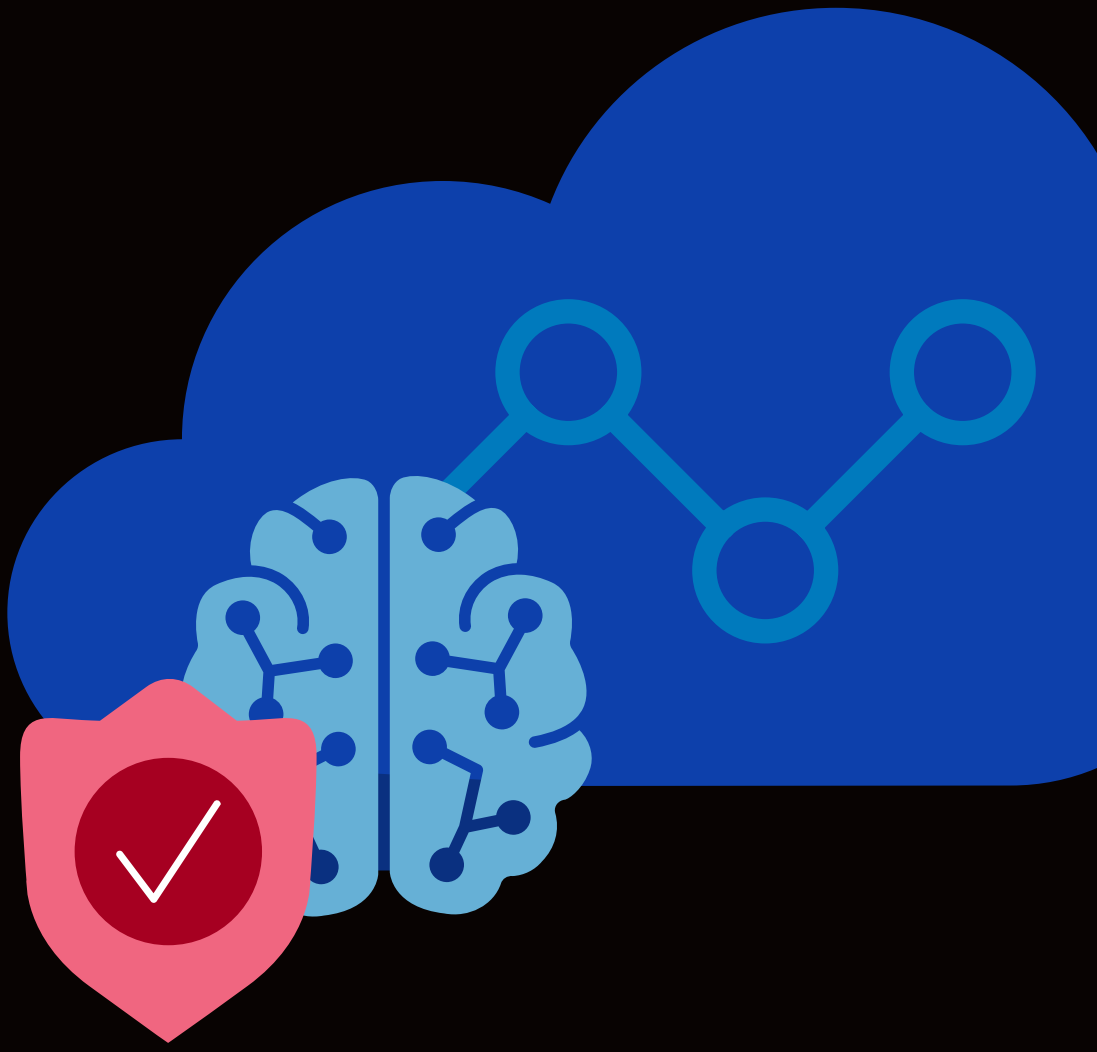

Table of Contents

Executive Summary

Designing and deploying AI applications at scale entails addressing intricate challenges to strike a careful balance between power availability and cost, latency requirements, data gravity, and system reliability. This document employs foundational concepts such as "power gravity" and "data gravity" to elucidate the driving forces behind AI infrastructure design. It discusses the trade-offs and synergies between these factors. Furthermore, it examines various deployment models—SaaS-hosted, cloud-hosted, self-hosted, and edge-hosted—highlighting their distinctive challenges and providing insights into achieving optimal performance, scalability, and sustainability for organizations.

Advanced strategies such as model optimization, federated learning, and hybrid approaches are examined to address various business and technical requirements of AI workloads. Regulatory compliance and environmental considerations in designing sustainable AI solutions are discussed. By presenting case studies and exploring future trends like nuclear energy and emerging technologies, this document aims to provide a guide for organizations targeting efficient and scalable AI applications. It highlights the need for informed design decisions that align with technological advancements and business objectives, ensuring AI infrastructure can adapt to changing market demands.

Introduction

The development and operation of AI applications at scale present unique challenges, particularly when it comes to balancing the often-competing needs for affordable and reliable power with service-level agreements (SLAs), and—in today's world of AI apps—the geographic constraints on GPU and data locations. AI factories (massive storage, networking, and computing investments serving high-volume, high-performance training and inference requirements) being designed and deployed today have immense power needs. In fact, the term "power gravity" has recently been coined to refer to the need to locate computational resources where power is abundant and cost effective. These power considerations can be at odds with SLA requirements, especially when there are tight latency constraints, such as for applications that require real-time inferencing, like augmented reality, autonomous vehicles, and smart manufacturing.

A third critical factor arises if there are large data sets needed by the application. Data owners use the term "data gravity" to refer to the bias for large data corpora to attract applications and services to where the data is generated or resides. Yet another input in this mix of design considerations comes from the reliability requirements for mission-critical applications. This white paper explores the relationship between these different forces and provides insights into how organizations can navigate these challenges, identifying areas of synergy and areas where trade-offs must be made.

Fundamental Application Architecture Challenges

Designing and deploying AI applications at scale involves navigating a complex landscape of architectural challenges. These challenges stem from the need to balance goals in the areas of cost and availability of power, latency requirements of the application, and overall system reliability while managing the business and regulatory constraints imposed by the locality of data and compute/GPU resources. While these challenges are true for applications generally, the recent emergence of AI applications—which are often more power and data intensive while simultaneously becoming increasingly mission critical across many industries—makes understanding and addressing these fundamental architecture challenges crucial for ensuring optimal performance, scalability, and sustainability.

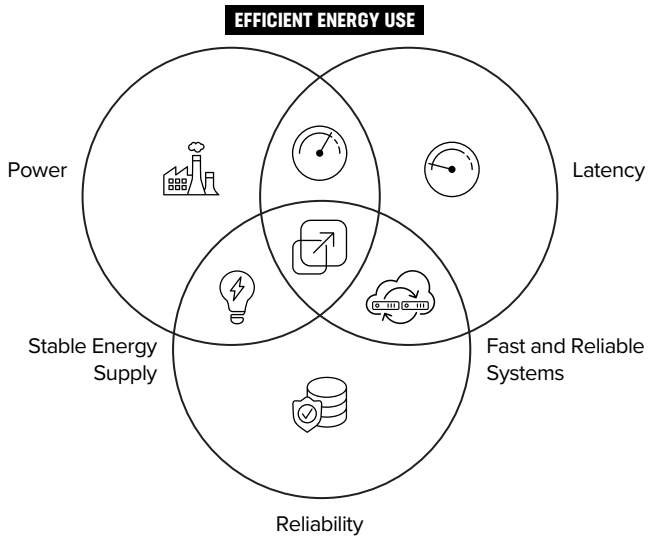

Figure 1: Balancing architecture challenges

Latency

The end-to-end latency experienced by an application's client is primarily determined by the application's problem domain and business objectives. While most applications typically set a target latency goal, the prescribed value can be as tight as tens of milliseconds or could be as long as a couple of seconds. Highly interactive multimedia applications, such as real-time language translation, augmented reality, or connected autonomous vehicles, typically require the lowest latency values. Clients of interactive consumer applications that are more transactional, such as e-commerce or banking, tend to be latency tolerant, though less so on mobile devices.[1] Finally, large, automated workflows, such as backup and database analytics, tend to be viewed more as “batch” jobs, with latency being less of a concern (compared to factors like cost efficiency.)

In addition to considering the metric of average latency, some applications also have requirements on the “worst-case” (for example, 99th slowest percentile) latency or the variability in latency (also known as “jitter”) that an application client observes. Citing the aforementioned examples, the consequence of an unfortunate delay would vary widely—a one-second slower translation 1% of the time might be mildly annoying, but a one-second delay in returning information about a neighboring car on a smart highway could have life-threatening consequences (or, in a more fault-tolerant design, an anxiety-inducing user experience from the vehicle making an emergency maneuver).

Other applications—notably augmented reality and some online games—can be more tolerant of modest latency, but only if that latency is consistent. For those applications, the “jitter” can cause more user experience issues than a few extra tens of milliseconds (ms) of latency. In fact, one study found that for modest latency (below 150 ms), each 10 ms increase in jitter resulted in twice the quit rate than a 10 ms increase in average latency.[2]

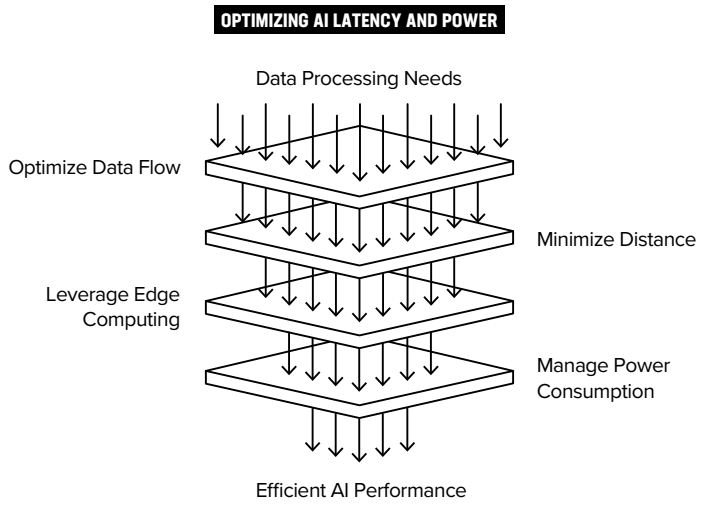

Figure 2: Latency optimization

In the context of generative AI (GenAI) applications, training generally is viewed as a “batch” job and therefore tolerates latency. In contrast, inferencing is used in more interactive workflows and will usually have latency requirements in the tens to hundreds of milliseconds, depending on the modality of the interaction (text vs. video response, for example), with worst-case and jitter requirements being dependent on the application, such as autonomous driving compared to real-time audio translation.

Understanding an application's latency requirements—average, worst case, and jitter—is crucial for deployment decisions. It's a good starting point as these requirements are usually independent of power and data gravity concerns.

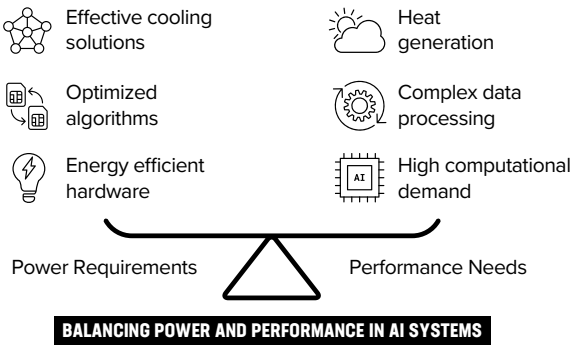

Figure 3: Balancing power and performance

Power

The rapid expansion of AI applications has led to significant power challenges that directly impact application architecture and delivery. Power availability is a primary concern, as AI data centers require substantial and reliable electricity to support high-performance computing tasks. For instance, data centers can consume as much power as a mid-sized city based in the United States, with projections indicating that demand will triple within three years, consuming 12% of the country's power supply.[3] This escalating demand necessitates careful planning to ensure uninterrupted power supply, as any instability can lead to application downtime and degraded performance.

The cost of power is another critical factor influencing AI infrastructure decisions. The energy-intensive nature of AI workloads leads to substantial operational expenses. For example, the increasing energy consumption of GenAI poses significant environmental concerns. AI data centers require immense processing power, leading companies to explore sustainable solutions to mitigate costs.[4] Consequently, organizations are exploring more efficient energy management strategies to optimize power usage and reduce expenses.

Addressing the environmental impact of increased power consumption is also paramount. The surge in energy demand from AI data centers contributes to higher carbon emissions and resource depletion, conflicting with global sustainability goals. For instance, AI workloads have sent data center emissions skyrocketing, prompting experts to explore ways to reduce energy use and promote sustainable AI.[5] Organizations are now prioritizing the integration of renewable energy sources and implementing energy-efficient technologies to minimize their ecological footprint.

Finally, navigating regulatory compliance related to power consumption and environmental impact is becoming increasingly complex. Governments and regulatory bodies are imposing stricter guidelines to ensure sustainable energy practices. For example, in Virginia, the rapid expansion of data centers has raised concerns about overburdening the electrical grid, leading to discussions on regulatory measures to manage this growth.[6] Organizations must adapt their application architectures to align with these evolving regulations, ensuring that their operations remain compliant while meeting performance and scalability requirements.

Reliability

The growing importance of AI applications has intensified the need for highly reliable infrastructure and architecture. AI systems are increasingly embedded in mission-critical business processes—from financial forecasting and fraud detection to clinical decision support and industrial automation. As organizations scale AI adoption, they are also placing these systems under business-driven SLAs that demand consistent uptime, performance, and correctness. Reliability—defined as the ability of a system to function correctly and consistently over time—is no longer just an operational ideal; it is a contractual requirement. Failure to meet reliability expectations can result in missed SLAs, financial penalties, reputational damage, and, in some cases, regulatory exposure.

The complexity of modern AI systems compounds this challenge. High-performance AI applications often span distributed clusters, GPU-intensive inference pipelines, real-time data streams, and hybrid deployments across cloud, edge, and on-premises environments. Each of these components represents a potential point of failure. A lapse in data freshness, a stalled inference engine, or a failed component in the orchestration layer can all degrade service reliability, leading to downstream impacts on users or automated systems that depend on the AI output. In systems under tight SLAs, even brief disruptions can cascade into significant business consequences. For example, if a recommendation engine fails during a high-traffic sales event, the loss of customer engagement and revenue can be immediate and measurable.

What makes reliability a particularly acute problem in AI is that failure modes are often non-obvious and data dependent. Unlike traditional applications, AI systems can produce subtly incorrect results—hallucinated outputs, misclassifications, or biased inferences—without crashing or logging errors. These “silent failures” can go undetected, quietly eroding user trust or introducing systemic risk, especially in high-stakes environments like healthcare, finance, or security. These characteristics challenge traditional monitoring strategies and elevate the risk profile of AI systems. Without strong reliability engineering practices and visibility into model behavior, organizations risk deploying systems that behave unpredictably in real-world conditions.[7]

Finally, as AI systems take on more decision-making responsibility, the consequences of unreliability become increasingly existential to the business. Unreliable AI can undermine automation initiatives, expose gaps in compliance, or cause downstream systems to behave incorrectly. This has led many organizations to adopt human-in-the-loop practices as a short-term safety measure. While effective for reducing the impact of AI misbehavior, this introduces manual effort that undermines the efficiency gains AI is meant to deliver. Over time, failure to address reliability at the architectural level leads to brittle systems that are costly to operate, difficult to scale, and ultimately unfit for business-critical use cases.[8]

In conclusion, navigating the challenges of AI infrastructure design requires a delicate balance between power, latency, reliability, and data requirements. As AI applications continue to evolve and become more integral to various industries, it is crucial to address these factors to ensure optimal performance and scalability. By understanding the unique demands of AI workloads and leveraging strategies such as model optimization, federated learning, and hybrid approaches, organizations can build efficient and scalable AI applications. Additionally, considering the regulatory and environmental impacts of power consumption and data governance will be essential in creating sustainable and compliant AI solutions. Ultimately, the key to success lies in making informed decisions that align with both business objectives and technological advancements.

Understanding the Effects of Application Deployment Model Choices

Power, latency, and reliability challenges vary significantly depending on where an AI solution is deployed. F5's AI Reference Architecture[9] includes four deployment models: SaaS-hosted, cloud-hosted, self-hosted, and edge-hosted. Each model presents unique challenges and strategies for balancing power gravity and data gravity. Understanding these challenges is crucial for designing and deploying AI applications that meet performance, scalability, and sustainability requirements.

SaaS AI

The AI solution is provided as a **fully managed service** by a third-party provider. Customers can access and use the AI capabilities over the internet without worrying about the underlying infrastructure, maintenance, or updates, making it a **convenient and scalable option**.

- Examples: Microsoft CoPilot, Salesforce Einstein, Microsoft Azure OpenAI Service w/ GPT 4o, Meta Llama 3.2 in Amazon Bedrock

Cloud-Hosted AI

The AI solution runs on cloud infrastructure provided by cloud service providers such as AWS, Google Cloud, or Azure. It offers **flexibility, scalability, and ease of integration** with other cloud services, while the **customer maintains control** over the configuration and management of their AI systems.

- Examples: vLLM running Llama 3.2 on AWS infrastructure

Self-Hosted AI

The AI solution is **deployed on the customer's own infrastructure**, such as on-premises servers or private data centers. This provides maximum control and customization options but **requires significant resources** for setup, maintenance, and management of the hardware and software components.

- Examples: NVIDIA Triton Inference Server running Llama 3.2 on bare metal on-premises

Edge-Hosted AI

The AI solution in an edge environment, **outside traditional cloud or data center infrastructure**. This model **reduces latency, enhances privacy, and ensures real-time processing** by bringing the computation closer to the data source or end-user.

- Example: Tesla Full-Self Driving, John Deere See & Spray, industrial IoT solutions

Figure 4: Four AI deployment models

SaaS

In a SaaS-hosted deployment model, the AI solution is provided as a fully managed service by a third-party provider. AI application owners can access and use the AI capabilities over the Internet without needing to worry about the underlying infrastructure, maintenance, or updates. This relief for the application owner comes with trade-offs. First, the total cost may be significantly higher, especially at a large scale, similar to what has been observed in the past for large public cloud deployments. Second, the set of capabilities and SLA parameters are set by the SaaS provider, implying that specific AI architecture patterns may not be possible (for example, fine-tuning, distilled models) and that there may be mismatches between the application's SLA needs and the provider's capabilities.

POWER, LATENCY, AND RELIABILITY CONSIDERATIONS

Power:

- Limited control over power efficiency improvements

Latency:

- Network latency due to data traveling to and from the cloud
- Potential performance issues during peak usage times
- Varied latency depending on the user's geographic location

Reliability:

- Dependency on the service provider's uptime and maintenance schedules
- Risk of outages affecting multiple users simultaneously
- Limited control over disaster recovery protocols

Cloud-hosted

In a cloud-hosted deployment model, the AI solution runs on cloud infrastructure (compute, networking, object storage, AI-specialized compute, AI model repository, vector database, etc.) from cloud service providers such as AWS, Google Cloud, or Microsoft Azure. This deployment model offers flexibility in choosing, configuring, and operating large language models (LLMs). Benefits include scalability, ease of integration with other cloud services, and the ability to select the most suitable language model for specific tasks.

Unlike SaaS, the application owners have much more control over both how the application is architected and where/how it is deployed, only limited by the building blocks provided by the cloud provider(s) and their set of deployment choices. However, it also presents challenges, such as managing data transfer costs, ensuring latency is minimized, especially if data is distributed across multiple cloud regions, and maintaining optimal performance configurations. Additionally, operating a custom LLM environment demands costly specialized skills, including expertise in machine learning, data engineering, and cloud architecture, to effectively manage and optimize the system.

POWER, LATENCY, AND RELIABILITY CONSIDERATIONS

Power:

- Power is controlled by the cloud service provider

Latency:

- Network latency based on the distance between the user and data center
- Impact of bandwidth limitations and network congestion
- Variability in latency depending on cloud provider's infrastructure

Reliability:

- Reliability depends on the cloud provider's infrastructure and SLAs
- Risk of widespread outages affecting all hosted services.
- Robustness of the cloud provider's disaster recovery and backup solutions

Self-hosted

In a self-hosted deployment model, the AI solution is deployed on the customer's own infrastructure, such as on-premises servers or collocated data centers. This model allows for high control and significant customization but requires considerable up-front capital resources for setup, maintenance, and management of the hardware and software components. These considerable resources include skilled personnel who can handle complex configurations and ongoing technical support, specialized hardware that can efficiently process AI workloads, and substantial financial resources to cover the costs of purchasing and maintaining this equipment. In addition, the customer must ensure adequate power availability and manage the environmental impact of increased power consumption. Where there are critical latency requirements specified by the AI application owner and/or limitations to the ability of local municipalities to deliver on the power demands of the AI data center, it may be required to set up multiple data centers in different locations, each of which must manage their power and environmental impacts.

POWER, LATENCY, AND RELIABILITY CONSIDERATIONS

Power:

- Customer responsibility for managing and optimizing power usage
- Need for backup power solutions (such as generators)
- Energy consumption depends on the scale and efficiency of the local infrastructure

Latency:

- Generally lower latency as data processing is closer to the source
- Network latency minimized by local infrastructure
- Latency variability dependent on local network configuration

Reliability:

- Reliability depends on the quality of the local infrastructure
- Customer responsibility for maintaining uptime and handling outages
- Need for robust backup and disaster recovery solutions

Edge-hosted

A primary challenge in edge ecosystems is ensuring power availability and reliability, as edge devices often operate in environments with limited power supply. Efficient power management is crucial for continuous operation, requiring optimized AI models capable of running on low-power hardware.

Backhaul reliability is a crucial factor for maintaining stable data transmission to central servers or cloud infrastructures for processing. In scenarios where real-time decision-making is essential, such as with autonomous vehicles or industrial automation, any disruption in backhaul connectivity can lead to substantial delays or operational failures. Therefore, solution design must account for potential non-trivial connectivity interruptions. Real-time decision-making defines edge computing, providing immediate insights and actions essential for applications like smart healthcare or retail, where milliseconds matter. Edge devices must manage event data streams and perform complex computations with minimal latency. Balancing power and data gravity presents unique challenges, demanding power-efficient AI models and robust backhaul connections.

POWER, LATENCY, AND RELIABILITY CONSIDERATIONS

Power:

- Limited power availability in some edge devices and locations
- Energy efficiency is crucial due to constrained power resources
- Potential need for battery solutions or renewable energy sources

Latency:

- Lower latency due to proximity to data sources to support real-time applications and services
- Minimized data transfer times and network congestion

Reliability:

- Reliability can vary based on the robustness of edge devices
- Potential challenges with maintaining and updating distributed edge infrastructure
- Dependency on local network conditions and connectivity

An Application Priority Choice: Data-Centric vs. Compute-Centric

Armed with an understanding of deployment models and their unique considerations, we now turn to how an application owner or a DevOps engineer can approach making the deployment decision for specific applications. We distinguish between existing applications and newly proposed or developing applications, as the latter will often have additional design options that are not possible for existing applications.

General Design Considerations

Some general properties are broadly true for most applications and deployment models, so they can be considered as ground rules. The first is that governance factors, especially compliance and business continuity, are often non-negotiable and are likely to be very difficult to impossible to design around, thereby making them forcing functions. The most common regulatory driver is data management. Specifically, data sovereignty requirements may entirely preclude some application deployment locations or models. Similarly, hard requirements around disaster recovery will usually necessitate some form of redundant deployment model. Lastly, from the perspective of all else being equal, it is important to remember that, because of economies of scale, large monolithic data centers that co-locate big compute, CPU, and storage will be more efficient than smaller ones. Of course, all else is rarely equal, which is the focus of the remainder of this section.

Existing Application Deployment

Start with latency

Millions of applications currently exist with varied user bases and reliability requirements that may not have been considered when the application was first designed. For these existing applications, latency becomes a third non-negotiable constraint, alongside compliance and business continuity. Latency is an important factor in deployment decisions as it is a function of the application's business needs, independent of power gravity. The application's latency requirements, if present and aggressive (as described earlier, highly dependent on the application's domain), are suited to deployment models that optimize network transit—edge-hosted or SaaS services that are distributed and can be directly accessed by the client. When control over worst-case latency and jitter is needed, self-hosted deployment may be a better option due to allowing direct management of the network stack. Similarly, applications with specific data requirements that are sensitive to latency will benefit from collocating data and compute resources to minimize data transfer delays, which often makes SaaS deployment impractical.

From the lens of GenAI applications, aggressive latency requirements will be more common for inferencing, especially when the output is real-time multimedia, such as real-time audio language translation. AI training workflows don't have the same needs, as these long-running activities can tolerate latency.

For existing applications that have aggressive latency requirements, edge-hosted deployments are usually the best fit, though this may require data to be moved to or (more likely) cached at the edge. Where that is not possible, applications with slightly less aggressive latency goals and/or those with significant overlap between self-hosted locations and the application customer base can use self-hosted deployment. This model can also be used to relieve possible data constraints. While less ideal, SaaS services that employ a distributed edge architecture can be used if worst-case latency and jitter are not primary concerns and the application client device can directly invoke the SaaS service without requiring a data center module to mediate the interaction.

Then think of data needs

After latency, the next decision is the trade-off between power gravity and data gravity. While both these factors are intertwined, and the trade-offs can be iterated upon, we recommend starting with data, as it more often has hard compliance constraints around privacy and sovereignty, whereas power constraints tend to mostly limit upside scalability, for which there are additional architectural options.

The hard data constraints will typically force more black-and-white decisions on deployment. For example, if regulatory requirements mandate that some data be kept within a specific geography, that is likely to force a decision to go with a distributed deployment model with nodes in each regulated geography or to forego other geographic markets entirely. On the other hand, the overhead of building data centers that require complex certifications (such as FedRAMP) may strongly bias business leaders towards fewer, larger data centers.

Another important consideration is the adjustable operational expense (OpEx) of data movement required by the application—specifically, how much data must be available wherever the application runs. The data requirements vary by application type:

- For chatbot applications, the primary data often comes directly from the client request[1]
- For geo-specific services like food delivery or ride sharing, most required data is naturally location-based and doesn't need high-speed transfer
- For applications like weather forecasts, the necessary data may be relatively small or easily cacheable

While some applications do require access to large volumes of non-cacheable data, in most cases the cost and latency of data movement are minor compared to compute costs. For example, the costs of moving data for a chatbot leveraging retrieval-augmented generation (RAG) backed by a large document store are still likely lower than the cost of the GPU running an LLM.

[1] It is important to note the AI models are also an avenue of data, but in most cases the models are fairly static, relatively modest in size, and easily cacheable; therefore, the cost of pushing the model to multiple locations (such as an edge-hosted environment) is relatively minimal.

Depending on the application, the deployment biases arising from the data requirements may align or be at odds with the latency-driven decisions. Where they align—such edge computing bias as for geographically-based applications (such as ridesharing)—the decisions are easy. Similarly, if the application does not have strong latency requirements, then the data-driven deployment decisions will not cause conflict. Only those applications that have both strong latency requirements and large monolithic data create tension. Fortunately, such applications are uncommon, and when they do occur, the usual trade-off is to focus first on meeting the latency constraints and then accepting additional OpEx costs associated with data motion.

Power decisions impact OpEx and scale limits

The third primary dimension of application deployment trade-offs is power, including:

- The cost of each client request
- Power efficiency of the application as measured by watts per client request
- The maximum number of concurrent users that the application can support based on maximum available power and efficiency

For data centers that are not self-hosted (such as hyperscaler cloud providers and SaaS providers), the monetary cost of power is implicitly included in the cost of the exposed resource, including CPU, disk, memory, or GPU.

For applications that are especially power intensive per client (typically either large database apps or ML/GenAI heavy apps), the cost of power can be an important factor in the overall OpEx of the application. For power-intensive apps that do not require decentralization (such as apps that are latency tolerant), the preferred option is to deploy those apps in large data centers where power is inexpensive. However, if latency and/or data locality needs drive decentralization, those needs should take precedence. Given the choice, the distributed nodes should still take advantage of the lowest available priced (reliable) power within the required geographic region.

In addition to the cost, the maximum available power can be a consideration. This is usually not an issue until one of two things occurs. First, if an application has a huge user base or becomes especially power intensive (such as GenAI model training), it can outgrow the local utility's customer quota or even ability to supply reliably. Alternately, a data center may be serving many smaller apps that in aggregate exceed the power supplier's quota or capacity. The solution to the first set of issues is to find a way to globally load balance across multiple data centers. This occurs naturally for edge-deployed apps. Techniques such as DNS load balancing can be used when the application deployment is not already distributed.

The second issue—power limitations arising from an excessive portfolio of applications—can be addressed by deploying to a small number of data centers, each built at a location with relatively inexpensive power. This is the strategy used by today's largest cloud providers.

The final consideration is the reliability of power. Most large data centers have backup power for short-duration outages, but if outages become common (such as rotating blackouts) or the area is prone to natural disasters that could result in long outages, an N-1 or N-2 redundancy strategy should be employed. Sufficient excess aggregate power capacity should be available across all data centers so that the loss of any one (or perhaps two) can be shouldered by the remaining data centers.

In short, power capacity concerns can often be addressed by distributing the workload across multiple processing locations using DNS on a per-application basis. This approach often aligns with the need to distribute workloads for other reasons like latency or data geo-locality. The primary exception is for non-latency-sensitive but power-intensive applications that also require data to be centralized. This trifecta of conditions is fortunately rare, with the most common case being AI training. In such cases, performing the work in a single large data center, with cheap abundant power, is the recommended strategy, offloading as many other applications as needed to ensure the power-hungry application's needs are met. When even that is insufficient, application refactoring may need to be the last resort.

New Application Development

The development of new applications has more options to optimize the trade-offs between latency needs, data constraints, scalability goals, and the availability of reliable, inexpensive, and abundant power. While many of these trade-offs were discussed earlier relative to existing applications, here they are summarized with a focus on their application in the design process. We will also present how these ideas can be specifically executed within the context of GenAI application development.

Latency

When developing new applications, consider which business requirements demand low latency (or minimal jitter) and whether your architecture can separate these components into distinct microservices or applications that clients can access independently.

For example, many multimedia applications have a "control plane" that establishes connections and authenticates and authorizes the user and a "data plane" that delivers the media content. The control plane has relaxed latency constraints and thus more freedom in deployment location, even if the data plane will have latency-driven deployment considerations. The nuance around direct client access is mentioned because it enables latency-sensitive services to be decoupled from the latency-tolerant ones from the user's perspective.

In addition, if much of the data required to deliver responses with low latency is infrequently updated, or if the application allows update to that data to be updated in a more relaxed manner, then caching the data can solve a potentially difficult trade-off between centralized deployment (for data reasons) and distributed deployment (for latency). Consider traffic monitoring for a navigation application. Traffic information for remote locations is latency

tolerant, while only the data for the local location is latency sensitive. In that case, the distributed application architecture can cache information for remote application nodes and allow that cache to be lazily updated.

Data

Using the same strategy as for latency, new application development should start by identifying any hard requirements (such as regulatory and compliance[10]) for data locality. If the exercise uncovers a limited number of localities that have such constraints, then the business owner should consider if those specific locations are of sufficient importance to drive deployment location decisions.

Assuming the hard data constraints are met, and that latency requirements do not force a distributed deployment model where data and compute must be co-located, then the primary data consideration is understanding the cost of transporting the data to compute resources, if the two are not co-located. This computation primarily depends on:

- The amount of data that must be shipped for each request or unit of compute
- The cost of moving each byte of data (with acceptable data transfer latency)

Taking a consumer energy monitoring application as an example, if there are 1 million monitored households, each sending 10KB of data per hour at $0.10 per GB, the data transmission cost would be $1.00 per hour. This cost is relevant for business owners who must make trade-offs around data transmission costs versus power costs.

Power

Primary power considerations are the cost per unit, reliability, and abundance/ability to scale. Assuming DevOps teams follow a practice of distributing applications to different data centers to keep each data center within its capacity, the main scalability concern of any single application is whether it is intended to scale past the capacity of a single data center. If this is a requirement for a new application, that application should be architected so that load can be distributed to multiple data centers, perhaps leveraging a DNS-based load balancing and routing technology. In the rare case of an application that has high upside scalability needs but cannot be deployed across multiple data centers, then the design should be examined to see if the power required per request or per client can be reduced.

The reliability needs of the application should be evaluated against the reliability capabilities of the deployment locations. When these locations cannot meet reliability requirements, the application design should incorporate load distribution methods—similar to those used for scalability—to enhance overall system reliability.

Finally, minimizing the OpEx of the application means minimizing the sum of the power costs (for compute and data) and any data motion costs. New applications will typically have freedom regarding where the necessary compute resources, such as virtual CPUs, GPUs, and

Kubernetes clusters, are located. Application owners will typically choose the location that has the lowest power cost as long as reliability and maximum power requirements are met.

In the case of legacy data, it should be co-located if possible to eliminate data motion costs. When the home of the data cannot be moved, the OpEx owner should compute the marginal increased cost of power of moving the compute to the location of the data, versus the cost of transmitting the data to the low power cost deployment location.

Going back to the energy monitoring example, where the data transmission costs were $1.00 per hour, let's assume that the difference in power cost is $0.02 per hour (between the low-cost power location and the location that is the data "home"). Then, if the power used for computation is over 50W/hour, moving the data to the low-cost compute data center is more cost effective than running compute resources in the same data center as the data.

In the Context of GenAI Applications

The recent explosion of applications leveraging GenAI technologies, and specifically the data considerations and power attributes of specialized AI processors has made the trade-offs mentioned especially relevant. Therefore, we provide some specific guidance in the context of developing of new GenAI applications.

The distinction between training workflows and inferencing workflows is a key one. Inferencing (the classification of an input or the creation of new content) is typically less compute-intensive than training, which involves the creation of a new model, either from scratch or derived from a prior model.

Inferencing

Inferencing applications typically create a response, whether text, audio, image, or video, based on a user request. The limitations of most of today's AI inferencing hardware often preclude latencies of under 100 milliseconds, but it is expected that this number will improve significantly over the next 12-24 months. Therefore, while current hardware disallows achieving latencies for which deployment location matters, we believe this will change quickly. Consequently, understanding the business needs for newly developed AI apps will be increasingly important in the next one to two years. Latency-sensitive AI applications will likely emerge in real-time language translation or autonomous navigation.

Two relevant emerging GenAI technologies are worth noting for latency, data, power. First, is RAG technology to improve AI response quality, augmenting the prompt with additional contextual data judiciously selected from a specialized data corpus. When RAG data is remote (such as a Microsoft SharePoint document store in the cloud), obtaining that data to enhance the prompt can incur additional latency. Some latency-intolerant apps may need to forego RAG in favor of alternate techniques, such as specialized or fine-tuned models (described later).

Second, if the size of the RAG data is large, the additional data transmission cost may become relevant. However, because the point of RAG is to choose the additional data judiciously, we expect the additional cost to typically be manageable.

Also relevant is the use of more specialized models. Several technical approaches exist, but the most relevant for inferencing deployment decisions is model distillation, which creates a relatively small (compared to a general-purpose LLM) model that embodies only the knowledge needed for the application's problem domain. If done well, this smaller model can be used to reduce both latency and power—smaller models typically have lower latency and consume less power. Therefore, new GenAI applications that address only a limited knowledge domain should consider specialized models, especially distilled models if latency or power are primary concerns for those applications.

Training

The creation of a new model tends to be much more compute and data intensive than tuning, typically taking days or weeks. As a result, training is almost always latency tolerant without deployment location constraints for compute. However, training typically ingests very large amounts of data—OpenAI GPT-4 is believed to be trained on over 60TB of data.[11] With multiple iterations of that data used in a single training run, it is highly desirable to have the data present in the same location as the training compute hardware for the entire training duration.

Another benefit of placing training data near training hardware is improved efficiency, as GPUs perform better when they can access data with minimal latency. Note that if the training data does not change throughout the process (true in most cases), the cost of making a one-time copy of the data to the training compute location is likely to be relatively small. For example, the cost of transferring a training set of 60TB, at a transmission cost of $0.10 per GB is $6,000.00, which is usually less than 0.1% of the compute cost of training.

In short, for most GenAI training use cases, a centralized deployment model is the best choice, as it then co-locates data with compute in a location that can provide inexpensive power in sufficient quantities. If multiple models are to be trained, they can usually be treated as independent, so each training event can be assigned to a different data center if needed because of power availability issues. The primary exception is for scenarios where the training data has hard data residency requirements, in which case either the compute location must be chosen to honor those constraints, or the data must be handled separately using other training augmentation techniques.

Fine-tuning is an important GenAI technique for creating specialized models that better align with application objectives than their generic base models. Unlike other methods, fine-tuning achieves this alignment without adding runtime costs during inference. Training data for fine-tuning may contain sensitive data with location limitations,[2] dictating the location of where fine-tuning can occur.

Model optimization

AI model optimization addresses the tension between power gravity and data gravity to reduce computational and power requirements. This can involve techniques such as purpose-built models, model pruning, quantization, and federated learning.

[2] In the case of the fine-tuning corpus, the data constraint may be either regulatory or data sensitive due to the intellectual property value of the training set.

Federated learning

This technique allows AI model training to be partitioned across multiple decentralized devices or servers, reducing the need to centralize data and computational resources. It can help manage the peak demands of power and data gravity by distributing the computational load.

Hybrid approaches

Organizations can adopt hybrid approaches that combine centralized and decentralized strategies. For example, training large models in centralized data centers where power is abundant, while deploying optimized models to the edge for inference.

Optimizing data transfer and storage

Reducing data transfer and storage needs through techniques such as data compression and deduplication helps minimize power consumption.

New Data Center Investment

Locating data centers near abundant, reliable, and cost-effective power

Strategic placement of data centers in regions with access to cheap and reliable power sources helps reduce operational costs and ensure a stable power supply.

Investing in renewable energy sources

Utilizing renewable energy sources such as solar, wind, and hydroelectric power ensures a sustainable power supply and reduces the carbon footprint of data centers.

Implementing energy-efficient technologies

Employing energy-efficient hardware and cooling technologies can significantly reduce power consumption.

Case Studies

AI Factories

The rapid expansion of AI-driven workloads has dramatically increased the demand for high-performance data centers, making power availability a critical constraint in site selection. While data gravity pulls AI infrastructure toward locations rich in data and connectivity, power gravity—the availability and distribution of energy resources—places hard limits on where these facilities can operate.

This tension is evident in Taiwan's recent moratorium on large data centers in the north,[12] where the government acknowledged that the existing grid infrastructure could not support further expansion. Similarly, Google Ireland's proposed data center in Dublin faced rejection,[13] in part due to concerns over power supply constraints. These cases highlight how AI-driven data centers must balance the benefits of data locality with the reality of power availability, ensuring that infrastructure can scale without overwhelming regional energy grids.

Beyond site selection, power gravity can shape the regulatory landscape for AI data centers, influencing long-term operational feasibility. In the United States, the Federal Energy Regulatory Commission (FERC) recently blocked an agreement that would have allowed additional power allocation from a nuclear plant to an Amazon data center, citing risks to grid stability and public energy costs.[14]

As AI data centers proliferate, businesses must align data gravity and power gravity, ensuring that workloads are deployed where both data and power resources are sustainable. Ignoring this balance can lead to inefficiencies, operational disruptions, and regulatory setbacks. Moving forward, enterprises must collaborate closely with utilities and policymakers to integrate AI infrastructure with energy planning, optimizing not just for computational performance but also for grid resilience and sustainability.

Edge Computing and IoT

When designing AI and ML solutions for edge and IoT deployments, balancing latency, security, and power requirements is essential for creating viable and efficient systems. Unlike traditional cloud-based AI, which benefits from abundant computational resources, edge deployments must operate within localized power constraints while still meeting performance expectations. NEC's walkthrough face recognition system[15] exemplifies this challenge by processing image data locally at the edge. This approach mitigates the effects of power gravity by reducing the inefficiencies of continuous cloud offloading. At the same time, data gravity plays a crucial role—biometric data captured at the edge has inherent security and regulatory concerns, making local processing not just a performance optimization but a necessity. By keeping inference local, NEC's system significantly reduces latency for real-time authentication while also enhancing security by limiting the movement of sensitive data across networks to reduce exposure to potential interception or breaches.

The interplay of power gravity and data gravity makes balancing latency, security, and power efficiency particularly complex in edge AI deployments. Devices such as security cameras, industrial sensors, and smart city infrastructure must maintain high levels of performance while minimizing both energy consumption and the risks associated with moving sensitive data.[16] To achieve this balance, developers can leverage specialized AI accelerators that optimize inference efficiency, deploy lightweight neural networks that reduce computational overhead, and implement adaptive power management strategies to sustain long-term operation.

NEC's solution demonstrates how a well-architected edge AI system can align with latency, security, and power requirements, keeping computation and sensitive data where they naturally reside while delivering real-time, secure, and energy-efficient AI. As edge AI adoption grows, ensuring that designs carefully account for these interdependent forces will be key to delivering scalable, sustainable, and secure AI solutions.

Future Trends and Considerations

The Role of Nuclear Energy

The International Energy Agency (IEA) estimates that global investment in grid infrastructure was nearly $400 billion in 2024, with projections to rise to around $600 billion annually by 2030.[17] This surge in investment is driven by the decarbonization of electricity generation, the growing share of electricity in energy consumption, and the need to fortify grids against extreme weather events.

Advances in Power Efficiency

Schneider Electric will leverage its expertise in data center infrastructure and NVIDIA's advanced AI technologies to introduce the first publicly available AI data center reference designs.[18] These designs are set to redefine the benchmarks for AI deployment and operation within data center ecosystems, marking a significant milestone in the industry's evolution.

As AI applications gain traction across industries and demand more resources than traditional computing, the need for processing power has surged exponentially. The rise of AI has spurred notable transformations and complexities in data center design and operation, prompting companies to swiftly construct and operate energy-stable facilities that are both energy-efficient and scalable. This collaboration between Schneider Electric and NVIDIA exemplifies the growing trend toward advanced power efficiency solutions for AI applications, paving the way for a more efficient, sustainable, and transformative future.

"We're unlocking the future of AI for organizations," said Pankaj Sharma, Executive Vice President, Secure Power Division & Data Center Business, Schneider Electric. "By combining our expertise in data center solutions with NVIDIA's leadership in AI technologies, we're helping organizations to overcome data center infrastructure limitations and unlock the full potential of AI."

Regulatory and Governance Impacts

As regulatory frameworks surrounding data sovereignty and privacy continue to evolve, AI applications and infrastructure must adapt to an increasingly complex landscape of governance requirements. Policies such as the GDPR in the EU and China's PIPL impose strict controls on data residency, influencing where AI models can be trained, deployed, and executed. This has led to a regionalization trend in cloud and AI infrastructure, where providers like Microsoft Azure and AWS have established sovereign cloud offerings to ensure compliance with local data protection laws.

Additionally, financial and healthcare regulations—such as HIPAA in the United States and emerging data localization requirements in India[19]—are prompting businesses to prioritize localized data storage and processing. These trends reinforce the concept of data gravity, where large volumes of regulatory-bound data necessitate that AI and compute resources be collocated in specific jurisdictions. As data accumulates in compliance-driven locations, it exerts a gravitational pull on the associated AI workflows, driving further investments in localized compute infrastructure to maintain operational efficiency.

Looking ahead, AI governance will not only dictate data storage locations but also influence how AI models are trained and deployed, particularly in regulated industries like finance, healthcare, and national security. For example, the EU AI Act introduced ethical AI requirements that could limit the geographic scope of model training and inferencing based on risk assessments. Additionally, as energy consumption and sustainability concerns rise, jurisdictions like the Netherlands[20] and Singapore[21] are imposing moratoriums on new data centers, amplifying the impact of power gravity—the tendency for AI compute infrastructure to cluster in regions where power availability, cost, and sustainability align with regulatory and economic constraints. These forces will drive AI infrastructure strategies toward a balance between regulatory compliance, sovereignty, and efficiency, fostering innovations in edge computing, federated learning, and on-premises AI solutions to navigate restrictions while maintaining performance and scalability.

Emerging Technologies

Modular nuclear power and private energy generation are emerging technologies that may provide new opportunities for organizations to address power gravity challenges in AI applications and data center infrastructure. Small modular reactors (SMRs), with their modular design and reduced land requirements, present a scalable and flexible energy source ideal for creating localized power for data centers, reducing reliance on traditional power grids.

For instance, Oracle is designing a gigawatt-scale data center powered by a trio of SMRs, aiming to meet the substantial energy demands of AI operations.[22] Similarly, Google has entered a strategic partnership to synchronize new clean power generation with data center growth, accelerating the transition to a carbon-free future for AI.[23]

Companies are also exploring private energy generation through renewable sources to create localized power solutions. Google's recent $20 billion partnership with Intersect Power and TPG Rise Climate exemplifies this approach, focusing on co-locating data centers with solar, wind, and battery storage facilities to ensure an efficient and sustainable energy supply.[24] These initiatives aim to reduce reliance on traditional power grids, enhance energy resilience, and address the escalating power demands driven by AI advancements. By integrating localized power sources, organizations can mitigate power gravity challenges, ensuring energy supply aligns closely with consumption needs, thereby optimizing performance and sustainability in their AI and data center operations.

Conclusion

Navigating the challenges of AI infrastructure design requires a delicate balance between power, latency, reliability, and data requirements. As AI applications continue to evolve and become integral to many industries, it is crucial to address these factors to ensure optimal performance and scalability. By understanding the unique demands of AI workloads and leveraging strategies such as model optimization, federated learning, and hybrid infrastructure approaches, organizations can build and operate efficient, scalable, and transformative AI applications. Additionally, considering the regulatory and environmental impacts of power consumption and data governance will be essential in creating sustainable and compliant AI solutions. Ultimately, the key to success lies in making informed decisions that align with both business objectives and technological advancements.

GLOSSARY

AI Factory: A massive storage, networking, and computing investment serving high-volume, high-performance training and inference requirements.

CapEx (Capital Expenditure): Up-front costs for acquiring or upgrading physical assets or equipment, such as building new data centers or purchasing servers.

Data Gravity: The tendency for large datasets to "pull" compute and services toward their location, often due to bandwidth, latency, or regulatory factors.

DNS Load Balancing: A technique that distributes network traffic across multiple servers or data centers using the Domain Name System, helping balance load and improve reliability.

Edge-hosted Model: Deployment of AI services where inferencing and responses are generated closer to users or data sources, reducing latency and bandwidth.

Federated Learning: A collaborative machine-learning approach in which models train across multiple decentralized devices or servers, allowing training data to be distributed without centralizing raw data in one place.

Fine-Tuning: The process of refining a pretrained model with domain-specific datasets to improve performance on targeted tasks, as an alternative to creating a new model from scratch.

Generative AI: AI systems that produce new content (text, images, audio, etc.) rather than just analyzing existing data or making classifications.

GPU (Graphics Processing Unit): A specialized processor originally designed for graphics rendering, now essential for parallel computation in AI model training and inference.

LLM (Large Language Model): A large-scale foundation model, such as GPT, trained on vast text corpora and capable of understanding and generating human-like language.

Model Distillation: The practice of creating a smaller, more efficient AI model from a larger one—maintaining key capabilities but reducing computational overhead.

OpEx (Operational Expenditure): Ongoing costs for running a solution or infrastructure (e.g., cloud usage fees, maintenance, utilities), as opposed to more significant up-front capital expenses.

Power Gravity: The tendency for AI compute resources, like GPU clusters, to be positioned in locations offering abundant, renewable, or affordable power to manage operational costs and scale.

RAG (Retrieval-Augmented Generation): An AI technique in which models retrieve external text or documents during inference to enhance the relevance and depth of generated responses.

SaaS-hosted or Cloud-hosted Model: AI solutions delivered by third-party providers (SaaS for fully managed services; cloud-hosted for more architectural control) while still relying on external infrastructure.

SLA (Service-Level Agreement): A contractual commitment specifying the performance, availability, or uptime targets that a service provider guarantees to meet.

[1] Marketing Dive, Google: 53% of mobile users abandon sites that take over 3 seconds to load, Sep 2016
[2] Huang, P., & Lei, C.-L. (2009). Effect of network quality on player departure behavior in online games. IEEE Transactions on Parallel and Distributed Systems, 20(5), 593–606, Aug 2008.
[3] Reuters, Big Tech's data center boom poses new risk to us grid operators, Mar 2025
[4] Wired, Is ai more sustainable if you generate it underwater?, Sep 2024
[5] MIT Sloan School of Management, AI has high data center energy costs — but there are solutions, Jan 2025
[6] Reuters, Data center build-out stokes fears of overburdening biggest US grid, Mar 2025
[7] Google Cloud, AI and ML Perspective: Reliability, Oct 2024
[8] Trigyn Technologies Limited, Ethical considerations when designing AI Solutions, Sep 2024
[9] F5, AI / ML Reference Architecture Overview, Jan 2025
[10] DataBank, Compliance in Data Centers: Navigating Regulatory Requirements for businesses, May 2024
[11] KDnuggets, GPT-4 Details Have Been Leaked!, July 2023
[12] Data Center Dynamics, Taiwan to stop large data centers in the North, cites insufficient power, Aug 2024
[13] Network World, Google Ireland bid to build new data center rejected, Aug 2024
[14] Utility Dive, FERC rejects interconnection pact for Talen-Amazon Data Center deal at nuclear plant, Nov 2024
[15] NEC, Case Studies of Edge Computing Solutions, Oct 2017
[16] Grzesik, P., & Mrozek, D., Combining Machine Learning and Edge Computing: Opportunities, Challenges, Platforms, Frameworks, and Use Cases, Feb 2024
[17] The Economist, A new electricity supercycle is under way, Jan 2025
[18] Schneider Electric, Schneider Electric collaborates with NVIDIA on designs for AI Data Centres, Mar 2024
[19] Carnegie Endowment for International Peace, Understanding India's new Data Protection Law, Oct 2023
[20] Data Center Dynamics, The ongoing impact of Amsterdam's data center moratorium, Aug 2024
[21] LightReading, Singapore ends data center pause as it seeks sustainable growth, Jul 2023
[22] Power Engineering, Oracle designing data center to be powered by trio of small modular reactors, Sep 2024
[23] Google, A new approach to data center and clean energy growth, Dec 2024
[24] Reuters, AI boom spurs Big Tech to build clean power on site, Feb 2025