ARCNet Architecture

ARCNet Architecture – HPC to Federated Edge AI Network for Autonomous Discovery

Introduction

ARCNet (Autonomous Resource Controller Network) is envisioned as a **global, open-architecture AI network** that bridges exascale computing with distributed edge systems. To accelerate time-to-market, ARCNet is pivoting from an initially planned 1 GW hyperscale build-out to a **federated architecture** leveraging both **national supercomputers and thousands of modular 1 MW data centers**. In this new design, the "brain" of ARCNet resides on Oak Ridge National Laboratory's **Frontier** (the world's first exascale supercomputer) and upcoming **Lux** AI cluster, while a network of ~1000 **1 MW Rubin-based nodes** acts as a distributed "nervous system" across the country. At the extreme edge, **autonomous machines** (e.g. fleets of additive manufacturing units in ARC's ADAM platform) serve as the network's "senses and actuators," executing tasks and feeding data back into the system. The result is a one-of-a-kind closed-loop infrastructure for **sensing, reasoning, and action** – analogous to how Ethernet standardized data exchange, **ARCNet provides a universal protocol linking HPC, distributed compute, and edge devices** for autonomous operations.

ORNL Supercomputers as the "Brain" of ARCNet

ORNL's Frontier and Lux supercomputers form the cognitive core of ARCNet. Through a public-private partnership (the DOE **Genesis Mission**), ARCNet can tap Frontier's exascale performance now and Lux's forthcoming AI-optimized capacity in 2026 (ornl.govornl.gov). Frontier and Lux provide **unmatched training and simulation capabilities** – for example, training large-scale AI models or running high-fidelity physics simulations for materials and manufacturing. ORNL's Lux AI cluster is explicitly designed to support **"large-scale AI training and distributed inference"** as a secure, open platform (ornl.gov), aligning perfectly with ARCNet's hybrid approach. In this architecture, **ORNL's supercomputers act as the training brain**, developing new AI models, global optimization strategies, or digital twins using massive datasets. This concept is demonstrated in the ARC–ORNL pilot "Alloy Design Agent," where **ORNL handles model training at scale while ARCNet handles real-time inference and agent orchestration**. A secure **ARCNet–Frontier bridge** enables **"workload bursting"** to Frontier for intensive jobs and governed data flow back to ARCNet. By leveraging ORNL's HPC as the brain, ARCNet dramatically accelerates development – avoiding the long lead time of building its own 100 MW datacenters – and ensures access to state-of-the-art computing for U.S. government and industry users from Day 1.

Federated 1 MW Nodes as the Distributed Network "Nervous System"

Surrounding the HPC brain is ARCNet's **geographically federated network of ~1000 modular 1 MW datacenter nodes**. Each 1 MW node is a high-density AI/HPC cluster (built on NVIDIA's next-gen **Rubin** architecture) contributing a slice of the total capacity. Despite their smaller size, these nodes are extraordinarily powerful – Rubin racks can deliver on the order of **8 exaflops of AI performance per rack** (576 GPUs with fast interconnect), enabling each micro-datacenter to handle demanding inference and localized training tasks. Crucially, the distributed design confers agility and economic advantages. **Sites under ~1.5 MW qualify for a 30% federal solar Investment Tax Credit**, and are simpler to permit and deploy, allowing ARCNet to **roll out capacity in parallel across many locations**.

In this federated network, **intelligent workload orchestration** is key. ARCNet's scheduling system treats the 1 MW sites plus the ORNL brain as a unified "meta-supercomputer." Jobs are dynamically placed to optimize for performance and latency: tasks requiring tight internode coupling (e.g. a real-time physics simulation) can be **confined to a single 1 MW site or the HPC** to exploit nanosecond-level internal networks, whereas **"embarrassingly parallel" or asynchronous workloads** are spread across many sites in parallel. Data locality is prioritized – computations are sent to the node where the needed dataset or instrument is located, minimizing cross-country data transfers. This approach mitigates the inherent WAN latency between sites, ensuring that ARCNet achieves near-hyperscale performance for distributed AI inference and analytics, while retaining the resiliency of many smaller nodes. If one node goes offline, the others continue operating – an advantage in reliability and uptime over a monolithic 100 MW center. The 1 MW nodes also serve as regional hubs connecting to local edge devices (factories, vehicles, labs), processing their data and dispatching AI models or commands with minimal latency to the field.

Edge Integration via the ADAM Platform – Closed-Loop in Action

At the network's edge, ARCNet interfaces with fleets of physical machines through platforms like **ADAM (Autonomous Discovery and Advanced Manufacturing)**. ADAM is ARC's flagship "Hello World" application for ARCNet – **"the world's first end-to-end AI materials discovery through advanced manufacturing platform," an AI orchestrator that controls physical hardware (e.g. binder-jet 3D printers) to run autonomous experiments in closed loops (**GitHub**)**. In the new architecture, an ADAM deployment might consist of several additive manufacturing machines, robotic handlers, furnaces, and sensors at a site like ORNL's Manufacturing Demonstration Facility (MDF). These are connected to a nearby ARCNet 1 MW node for on-site compute and data buffering, and through that node to the HPC brain at ORNL. The workflow is a continuous **sense–think–act loop**:

- **Sense:** Edge instruments (printers, ovens, microscopes, etc.) generate data – e.g. a printed alloy's dimensions, microscopic images, or magnetic performance metrics. This data streams into the local ARCNet node and onward to the HPC brain.

- **Reason:** At Frontier/Lux, high-level AI algorithms (such as an evolving materials model or a design-of-experiments planner) analyze the incoming data and suggest new candidate designs or process parameters. *Training* of these AI models happens here using the aggregated dataset of all experiments, leveraging exascale compute to discover patterns or optimal solutions that a single site alone couldn't compute. Meanwhile, the ARCNet distributed layer can handle the *inference* part – running the trained models to evaluate each candidate in real-time and orchestrating the sequence of experiments via multi-agent systems. ARCNet's secure design (built with export control compliance from Day 1) ensures that sensitive data and AI models move between ORNL and field nodes in a governed, auditable fashion.

- **Act:** The chosen next experiment or action is dispatched back through ARCNet to the appropriate edge device. For example, the ADAM orchestrator (the Nova multi-agent system) might direct a printer to fabricate a new material composition, then instruct a furnace to sinter it and a suite of sensors to test it (GitHub). The local 1 MW node executes low-latency control loops (possibly running a digital twin or doing quick feedback analysis), while coordinating with the central planner. This **closed-loop workflow – "design → simulate → make → measure → learn"** – can iterate dozens of times faster than traditional R&D cycles, enabling rapid discovery of optimized materials or components.

This **ARCNet+ADAM closed-loop** exemplifies the platform's power. By uniting HPC-driven learning and planning with edge-level execution, ARCNet can autonomously drive complex, goal-directed processes like **materials discovery** (as in the Genesis Magnet Pilot) or, in the future, a fleet of self-driving vehicles or factories. Each cycle's results are fed back into the AI models, continuously improving the system's knowledge. Importantly, because ARCNet nodes are distributed geographically, similar closed-loop testbeds can run in parallel at dozens of sites (for different projects or domains), all benefiting from the shared HPC brain that aggregates their insights.

Advantages of the HPC + Federated Edge Architecture

By combining ORNL's supercomputing might with ARCNet's distributed network, this architecture offers **unique advantages** for technically savvy stakeholders:

- **Unmatched Compute Scale and Efficiency:** ARCNet effectively weds **leadership-class supercomputers** to a **nationally distributed AI fabric**. The ORNL "brain" provides **massive-scale training** (exascale flops, high-bandwidth networks) for developing AI models and simulations, while the 1 MW Rubin nodes contribute additional exaflop-scale throughput at the edge for inference and data processing. This hierarchy ensures **the right compute at the right place** – global learning tasks use the big iron, while routine inference is handled closer to the source. The result is **faster AI iteration** and up to *50% lower inference cost* compared to running everything on public cloud GPUs, thanks to optimized workload placement and ARCNet's low-cost energy design (solar + battery at each node).

- **Closed-Loop Autonomy at Scale:** Unlike siloed HPC setups or cloud-only approaches, ARCNet enables **closed-loop autonomy on a continental scale**. Data from thousands of sensors and machines flows into a central intelligence that can immediately refine models and issue new directives – all without human-in-the-loop delays. This tight integration of **sensing, reasoning, and acting** creates a powerful virtuous cycle: the more experiments and operations ARCNet runs, the smarter it gets, and the more efficiently it can optimize subsequent actions. This is critical for domains like advanced manufacturing, where design iterations and learning cycles traditionally take months or years. ARCNet's pilot at ORNL aims for a **10× reduction in discovery-to-product time** for new magnet materials, illustrating the impact of closing the loop. Furthermore, the architecture's inherent feedback loop ensures that **safety and performance improve continuously** – a necessity for autonomous systems such as self-driving fleets or adaptive power grids.

- **Geographic Resilience and Edge Proximity:** The federated 1 MW node strategy gives ARCNet an unparalleled **footprint**. Compute power is not centralized in one location but spread across many sites – some colocated with key partners like national labs and universities, others near industrial centers or even embedded in mobile platforms. This distribution reduces single-point-of-failure risk and can **localize services** for latency-sensitive tasks. For example, a factory in the Midwest can rely on a nearby ARCNet node for real-time control and analytics, with only strategic use of the distant HPC for heavy training. Moreover, sub-1.5 MW sites can more easily exploit **on-site renewables and existing grid interconnects**, avoiding lengthy infrastructure build-outs. It means ARCNet can scale incrementally and quickly – adding capacity node-by-node in response to demand or mission needs, rather than mega-projects that take years. For government and prime contractor users, this agility translates to **faster deployment of AI capabilities in the field**.

- **Open, Universal Platform:** ARCNet is being built as an **open-architecture, model-agnostic network**, akin to a common protocol layer for AI-driven autonomy. Just as Ethernet standardized the way diverse computers communicate, ARCNet defines a universal way to connect intelligent computing to real-world devices. The system supports any AI models or agent frameworks (it's not tied to a specific vendor or algorithm), enabling integration of innovations from across the AI community. The partnership with ORNL further ensures **open standards** (for example, developing an "Agent-to-Instrument" API spec for lab equipment control) and a talent pipeline (AI technologist fellowships) to broaden adoption. By treating **AI compute as a public**

utility available for any innovator, ARCNet lowers the barrier for federal programs, startups, or researchers to deploy large-scale AI solutions. This open ethos and the **independent, public-benefit ownership** of ARCNet make it a trustworthy neutral platform for multi-party collaboration – a critical factor for government and industry partners who require transparency and control over their AI infrastructure.

Conclusion

The proposed ARCNet architecture fuses the best of centralized and distributed computing to create a **"brain-to-edge" nervous system for autonomous systems**. By situating its core intelligence on ORNL's Frontier and Lux supercomputers, ARCNet gains immediate access to world-leading AI training capabilities and a strong government partnership. By federating that intelligence out to thousands of 1 MW nodes and down to edge devices, ARCNet achieves global reach, resilience, and real-time responsiveness. This closed-loop design – spanning **exascale data centers to smart sensors on factory floors** – gives ARCNet a decisive advantage in tackling complex, real-world challenges. From **autonomous R&D labs** that iterate on materials or medicines, to **self-optimizing fleets** of vehicles and industrial facilities, ARCNet provides the common infrastructure to sense the environment, learn and reason at scale, and act efficiently – all under a unified, open framework. For technically savvy investors and strategic partners, this architecture represents a leap forward: a blueprint for an AI-driven **"industrial nervous system"** that is as foundational and transformative in the physical world as Ethernet was for digital communication.

Sources: The feasibility of ARCNet's distributed 1 MW node model is detailed in an internal analysis. ARC's partnership with ORNL through the Genesis Mission outlines the closed-loop manufacturing pilot and secure HPC integration. ORNL's announcement of the **Lux** AI supercomputer emphasizes its role in **AI for materials and manufacturing** and support for distributed inferenceornl.govornl.gov. The ADAM platform is documented as an example of ARCNet's end-to-end autonomous experimentation capabilityGitHub. These components together underpin the unique ARCNet architecture described above.

Feasibility of Federated 1 MW

Feasibility of Federated 1 MW Rubin-Based Data Centers vs 100 MW Hyperscale

Introduction

ARCNet's original plan calls for 10 hyperscale data centers of 100 MW each (total 1 GW) to power advanced AI and HPC workloads. An alternative approach is to distribute this capacity across ~1000 smaller sites (~1 MW each) using NVIDIA's **Rubin** architecture – NVIDIA's next-generation AI/HPC platform known for packing extreme performance into high-density racks[1][2]. Each 1 MW node would function as part of a geographically **federated supercomputer**, rather than a mere edge cache. Power would come from on-site hybrid energy systems (solar arrays, battery storage, possibly **small modular reactors** in the future) to maximize renewable usage and reliability. Crucially, keeping each site below ~1.5 MW ensures eligibility for the 30% federal solar Investment Tax Credit (ITC) under current rules, without needing to meet extra labor requirements (which only apply to projects over 1 MW)[3][4].

This report evaluates the **technical and economic feasibility** of this distributed model versus the hyperscale approach, focusing on U.S. deployments. Key considerations include performance and networking trade-offs, cost per MW (with tax incentives), power/grid interconnect issues, regulatory and sustainability advantages of sub-1.5 MW sites, multi-tenant and reliability implications, and real-world examples of <5 MW distributed supercomputing nodes. A comparison table is provided to summarize costs and benefits of the two models.

Performance, Latency and Networking Trade-offs

Intra-Cluster Performance: A single 100 MW hyperscale data center can house hundreds of high-density racks with low-latency internal networks (InfiniBand/NVLink). For example, NVIDIA's **Rubin** architecture enables ~8 exaflops of AI performance in one rack (576 GPUs) with a fast fabric[1][5]. Tightly coupled HPC workloads (like large physics simulations or distributed AI training) benefit from nanosecond-scale node proximity and massive internal bandwidth. In contrast, spreading the same capacity across 1000 separate 1 MW sites means any cross-site communication occurs over WAN links with **orders of magnitude higher latency** (milliseconds) and lower bandwidth. Even with dedicated fiber connections, inter-node latency and throughput in a geo-distributed cluster will be far inferior to a local cluster's. **Geographically federated supercomputing inherently faces latency penalties** – as noted in grid computing research, loosely coupled distributed systems cannot efficiently run certain traditional supercomputing tasks that require frequent synchronization (e.g. fluid dynamics simulations)[6]. To mitigate this, the federated Rubin network would need intelligent workload scheduling: jobs requiring intense internode communication might be confined to a single site or region, whereas embarrassingly parallel or asynchronous tasks can be split across sites. Techniques to **prioritize data locality** – assigning tasks to the site where needed data resides – become crucial to minimize cross-site traffic[7].

Latency & Throughput: Within a hyperscale facility, specialized interconnects (InfiniBand HDR/NDR, NVLink, or NVIDIA Spectrum-X) provide high throughput and *extremely low latency* (often <5–10 microseconds node-to-node). A distributed 1 MW node network, however, must rely on metro or long-haul fiber. Even with high-speed optical links, propagation delays (roughly 5 µs per km) and routing overhead mean **latency could be 1000× higher** between nodes in different cities versus within one data hall. For tightly coupled AI training (which often uses synchronous parameter updates), this latency would drastically slow convergence unless algorithms are redesigned (e.g. using pipeline parallelism or async updates to hide latency). Thus, **performance per job may degrade in the distributed model** for workloads not easily partitionable. Some emerging HPC use-cases might tolerate federation – for instance, multi-site inference serving or federated learning – but ultra-scale training runs or MPI-style computations would see slowdowns. In effect, the distributed approach trades raw inter-node performance for other benefits like resilience. Each 1 MW Rubin cluster remains very powerful on its own (capable of multi-petaflop to exaflop-scale compute in a single rack[8][2]), so tasks that fit on one or a few racks can run at full speed. The challenge is **coordinating large jobs over many sites** – the software stack (scheduler, MPI, NCCL, etc.) must evolve to manage high-latency links gracefully.

Networking Complexity: Building a "distributed supercomputer" demands a robust, high-bandwidth network backbone connecting all sites. This likely means leasing fiber or deploying private optical links with redundancy. Achieving network throughput comparable to intra-datacenter levels is infeasible economically; instead, the design might group sites into regional clusters with fast connectivity, and use a hierarchical scheduling approach. Each 1 MW module could act like an "availability zone" – high-speed internally (via NVIDIA Quantum InfiniBand or similar) and connected to other modules via Ethernet or optical transport. Networking gear like **NVIDIA Spectrum-X** Ethernet fabrics and ConnectX-9 smart NICs support multi-site scaling, but performance will still fall short of a single-site network fabric[9]. Redundancy in networking is a plus: multiple smaller sites mean the overall system can route around a fiber cut or site outage by redistributing workloads elsewhere (with some performance hit), whereas a single-campus cluster is more vulnerable to a *total* network failure (however unlikely). In summary, **hyperscale centers excel at low-latency, high-bandwidth computing**, whereas a federated 1 MW node architecture sacrifices some of that performance in exchange for other gains (redundancy, proximity to data/power).

Redundancy and Fault Tolerance: This is one area where the distributed model shines. Ten 100 MW mega-centers concentrate capacity – a failure (power outage, cooling issue, natural disaster) at one could knock out 10% of ARCNet's compute at once. In contrast, 1000 independent 1 MW sites create inherent redundancy: losing one only removes 0.1% of capacity. The federated design can be highly **fault-tolerant**, rerouting tasks to other nodes if a site goes offline. It's analogous to cloud availability zones – many small zones reduce single points of failure. That said, multi-site HPC jobs would need checkpointing and resilience because if one site in a tightly-coupled job fails, the whole job can fail. The system would need to handle partial failures gracefully, perhaps by dynamically excluding the failed node and continuing (for algorithms that support it) or restarting that job segment elsewhere. Overall, **resilience is improved at the system level** (no single calamity can cripple the entire 1GW of capacity), but individual job resilience will depend on advanced software fault-tolerance.

Deployment and Cost per MW Analysis

Capital Expense per MW: Large hyperscale data centers typically enjoy economies of scale, achieving lower construction cost per megawatt of capacity. Industry data shows that seasoned hyperscale builders can deliver facilities for as low as **\$6 million per MW** (and in some cases even ~\$3.6M/MW in regions like China)[10]. In contrast, smaller installations usually have higher unit costs – mid-sized data centers (5–20 MW) average \$8–9M per MW[11], and 1 MW "edge" or modular sites can be in the upper part of the \$8–12M/MW range[12]. This is due to fixed costs (power distribution, cooling equipment, security) being less amortized at small scale. Each 1 MW site requires its own infrastructure: container or building, cooling system, UPS/generators (or batteries), network gear, etc., which might cost nearly what a larger facility's modules cost, but without the volume discount. There is potential to mitigate this: using **prefabricated modular data center units** can drive some economies of production. Vendors like Vertiv and HPE offer 0.5–1 MW turnkey modules that are factory-built and simply installed on site, reducing on-site construction time and cost. These **modular pods** include built-in cooling, power distribution, fire suppression, etc., and can be replicated for scale. For example, Vertiv's MegaMod blocks start at 0.5 MW and can be deployed rapidly, with everything pre-tested – they note that standardized builds and prefabrication helped hyperscalers cut costs dramatically[13]. Still, even with modular units, deploying 1000 separate sites will involve significant overhead in aggregate (1000 pads, 1000 hookups, etc.). **In pure CapEx, a 100 MW x10 approach likely has an advantage**, perhaps requiring ~\$6–8B total (using ~\$6–8M/MW), whereas 1000 × 1 MW might cost on the order of \$10B if \$10M/MW each.

However, the distributed model benefits from **Investment Tax Credit (ITC) savings and local optimizations**. Each 1 MW site with solar can qualify for a 30% ITC on the solar/battery portion of the project. Because each is under the 1 MW AC threshold, they automatically get the full 30% credit without needing to meet labor requirements[3][4]. By contrast, a single 100 MW solar farm feeding a big site would be subject to prevailing wage/apprenticeship rules to get the 30%, or else fall to a 6% base credit – a risk if compliance isn't achieved[3]. Starting construction now locks in these credits before phasedowns begin (the IRA keeps 30% ITC at least until 2032, after which it steps down)[14]. So ARCNet's distributed plan could **recoup a substantial portion of capital** via tax credits. For example, if each 1 MW site includes, say, \$1.5M in solar panels and \$0.5M in batteries (just an estimate), that \$2M renewable investment yields ~\$600k ITC benefit per site. Across 1000 sites, that's ~\$600M – helping offset the higher base cost. Moreover, each small site might tap *additional* local incentives: many U.S. states and utilities have grants or credits for behind-the-meter solar, energy storage, or microgrids, especially if under certain sizes or in certain locations.

Operational Expense (OpEx) and Energy Cost: A hyperscale 100 MW data center often secures bulk power rates from utilities – on the order of a few cents per kWh – and may operate with very efficient cooling (PUE ~1.1–1.3 for modern designs). Small sites might have slightly higher PUE due to lower absolute efficiency of cooling at small scale, although modern modular designs boast compact efficiency (one vendor notes *compact modular DCs use available energy more efficiently, yielding low PUE*[15]). The big difference is energy sourcing: a distributed 1 MW node can be strategically placed near cheap or surplus renewable energy. For instance, sites could be located in regions with high solar irradiance or on land where wind power is available, and directly consume that power. This avoids some transmission costs and price markups. With on-site solar + storage, **energy costs per kWh can be significantly reduced** (after payoff of the solar CAPEX, solar provides power at very low marginal cost). Excess solar generation could even be sold back to the grid or used to charge batteries for nighttime. In contrast, a 100 MW site must buy most of its power from the grid at commercial rates, unless it also builds a massive solar farm (which would need hundreds of acres). Thus, the distributed model might save on electricity expenses in the long run by maximizing **local renewable usage** and avoiding demand charges that large consumers face. Furthermore, smaller sites might not incur the same level of backup generator fuel costs if they rely on batteries; large data centers often maintain diesel generators (with significant maintenance and testing costs). If each 1 MW site can run grid-independent for many hours via solar + batteries (and maybe a small generator or future microreactor for emergency), it could reduce downtime risk and fuel expense.

It's worth noting that managing 1000 utility bills and maintenance crews is more complex than 10, but those can be handled via centralized monitoring and local service contracts. **Labor costs** may be higher in aggregate for many sites (you need more technicians spread out), but each site's simplicity (possibly unmanned, automated modules) could mitigate this.

Deployment Speed: Building ten 100 MW facilities is a massive undertaking – site acquisition, permitting, constructing large buildings, bringing in huge power feeds, etc. Such projects often take 18–36 months from planning to commission per site, and may face delays for power hookup (discussed in the next section). Conversely, rolling out hundreds of small prefabricated units can be done in parallel, and **each unit might be installed and live within 6–12 months**[16]. Many could be deployed concurrently across different states. The **prefab approach** significantly accelerates deployment: for example, modular data centers have been delivered and made operational in a matter of months in some cases, since most assembly and testing is done off-site[13]. By using standardized 1 MW modules, ARCNet could adopt an assembly-line deployment model – while site A's slab and utilities are prepared, site B's container is being built in a factory, etc. This agility could be crucial given the **race to add AI capacity**. In short, although the distributed model has higher unit costs, it leverages tax credits and potentially faster deployment to even the field. We will compare concrete cost estimates in the table later.

Power Availability and Grid Interconnect Considerations

One of the biggest practical challenges for large data centers today is obtaining sufficient power from the grid in a timely manner. A 100 MW data center draws as much power as a small city[17], requiring new substations, high-voltage lines, and utility upgrades. In several U.S. regions, big projects are **literally ready but sitting idle for lack of grid capacity**. For example, in Santa Clara (Silicon Valley), nearly 100 MW of newly built data center space is unenergized because the local utility cannot supply power until extensive upgrades (targeted by 2028) are done[18][19]. Northern Virginia – the largest data center hub – faces multi-year delays (2–5 years) to hook up new large facilities[20]. This bottleneck is a critical risk for any hyperscale deployment: ARCNet's 100 MW sites might be built, but could wait years for full power, undermining business goals.

By contrast, **smaller 1 MW installations can be more nimble in grid interconnection**. A 1 MW load can often tie into the local distribution grid (e.g. at 480 V or 12 kV lines) with far less bureaucracy than a 100 MW load that needs transmission-level service. Utilities typically have "fast-track" interconnection processes for small generators or loads under certain thresholds. Many 1 MW sites could potentially be sited in areas with existing spare capacity – e.g. near substations serving industrial parks or near large solar farms. The distributed model may exploit pockets of available power that wouldn't support a hyperscale alone but are perfect for a 1 MW addition. **Timeline-wise**, connecting a 1 MW project might take on the order of months to a year (for standard commercial service upgrades), whereas 100 MW projects are often 2+ years with significant grid construction. Also, if a 1 MW site can run mostly on its own solar generation, it might only need a modest grid connection for backup or nighttime, easing the utility's burden. In remote areas, a 1 MW solar + battery data center could even operate *off-grid* for extended periods, only occasionally using a small grid link or generator – effectively a microgrid operation.

Permitting and regulatory approvals also favor small sites in many cases. A 100 MW center triggers extensive environmental impact studies: large land clearing, potential cooling water usage, air permits for dozens of backup generators, and public hearings (communities may oppose a huge power-hungry facility or its diesel exhaust, noise, etc.). **In contrast, 1 MW sites have a smaller footprint and can often fly under the radar**. They might be built on already-developed land (rooftops, parking lots, existing warehouses retrofitted) or in rural solar farms. Being under certain thresholds can exempt them from stringent environmental reviews or complex zoning. For instance, many locales have simpler permits for facilities under some MW or square-footage limit. Even generators: a single 2 MW diesel genset at a small site might not trigger the same Clean Air Act permit requirements that 50 gensets at a big site would. Some 1 MW pods might use only batteries as backup (zero emissions on site). **Community acceptance** is likely higher for dispersed small nodes, which might be barely noticeable (a couple of shipping-container-sized units with solar panels) compared to a gigantic data center building.

Another advantage: **geographic diversity of power sources**. Ten hyperscale sites concentrated in a few states could all be subject to the same regional grid stress or regulatory changes. Spreading 1000 sites across the country means ARCNet isn't reliant on any single grid or region. If one area faces electricity price spikes or rationing, only a portion of the network is affected. The rest can continue operating at full power in other regions. This also allows "chasing cheap electricity" – workloads can even be shifted (if latency-tolerant) to nodes in regions where power is currently cheapest or most abundant (a concept known as geographic load balancing[21][22]). For example, daytime jobs could run on nodes in high-solar regions taking advantage of peak solar output, then shift at night to other regions.

One must consider **coordination with utilities** for so many sites. While each small site is easier to handle, managing hundreds of interconnection agreements across dozens of utilities is non-trivial. ARCNet would need a dedicated team (or partner) for power procurement and interconnects nationwide. Still, compared to negotiating a single 100 MW connection (which may require big investments from the utility), negotiating many small ones might actually be easier since they fit within existing frameworks (just like connecting 1 MW of new commercial load, which utilities do regularly).

In summary, **power availability strongly favors the distributed model**: it avoids putting "all the eggs" in a few overloaded grid baskets, and can leverage faster, simpler hookups. It turns a huge grid challenge into bite-sized pieces. The trade-off is complexity of scale – many agreements and varied timelines instead of a few. But given the urgency of AI build-outs (with even tech giants hitting power shortages[23]), being able to add capacity wherever it can fit is a strategic edge.

Regulatory, Tax, and Sustainability Advantages (Sub-1.5 MW Hybrid Sites)

Keeping each site under ~1–1.5 MW unlocks a suite of **regulatory and tax benefits** that larger installations cannot fully exploit:

- **Investment Tax Credit (ITC):** As noted, solar or storage projects *under 1 MW AC* automatically qualify for the **30% federal ITC** without extra conditions[3]. Projects above that size must comply with prevailing wage and apprenticeship requirements or else receive only a 6% base credit[3][4]. By staying just below the threshold, ARCNet's sites ensure they maximize federal incentives with minimal administrative burden. If construction begins before the current ITC phase-down (expected in the mid-2030s), they lock in the full credit[14]. Additionally, if any sites integrate other clean technologies (e.g. fuel cells or micro wind), those may also be ITC-eligible. The **tax savings significantly improve ROI** for each node.

- **Accelerated Depreciation and Deductions:** Smaller renewable energy systems and energy-efficient equipment often qualify for accelerated depreciation (e.g. 5-year MACRS for solar). Also, the IRS Section 179D deduction for energy-efficient commercial buildings could apply if the data center container design meets certain efficiency criteria. These tax treatments further reduce the effective cost of the distributed model.

- **Sustainability and Carbon Footprint:** Federated 1 MW sites powered by hybrid solar/battery systems can achieve a very high fraction of renewable energy usage. Each node can be designed as a mini **green data center**, potentially net-zero for operational power. In contrast, powering a 100 MW site with on-site solar is extremely challenging – it would require a vast solar farm and still need grid power at night. The distributed approach aligns with corporate ESG goals and can be marketed as a network of "green AI micro-factories." For example, an Iceland data center achieved zero-emission status by using 100% renewable power and free cooling[24]; ARCNet can emulate this across many micro-sites using local clean power. Furthermore, **transmission losses** are reduced when generation is on-site – improving overall efficiency.

- **Regulatory Simplicity:** Many states have **simplified interconnect rules for small systems** (often under 2 MW or 5 MW). For instance, small solar+storage projects may qualify as distributed generation with expedited interconnection and less onerous grid studies. Large power plants, by contrast, trigger FERC oversight and complex grid impact analyses. Being below 1.5 MW keeps each site in the "small generator" category, avoiding federal siting hurdles and lengthy queue processes. Also, small power systems can sometimes skirt local moratoria that target big data centers (some communities have paused new large data centers due to concerns about power and water use – a 1 MW site might not even meet the definition of those bans).

- **Permitting and Zoning:** In many jurisdictions, a containerized 1 MW data center could be installed under a simpler commercial building permit, especially if sited on an existing industrial property. There's likely no need for extensive new building construction – these modules can be placed on a concrete pad or within a light steel shell. **Environmental permits** are easier: sub-1.5 MW of solar and a battery have minimal emissions (just some inverter noise perhaps). If no diesel generator, air quality boards have no major say. If including a backup genset, a small one might be below air permit thresholds for emergency generators (for instance, some air districts exempt emergency gensets under certain kW from strict rules). Additionally, these small sites could utilize **creative siting**: e.g. colocating at renewable energy sites (a solar farm could host a 1 MW data module using its output), or at manufacturing facilities where waste heat from the servers could be reused (since each site's heat is modest, it could supply a nearby building's heating). This gives sustainability synergies – reusing server waste heat to warm local facilities, improving overall energy utilization.

- **Small Modular Reactors (SMRs) & Microreactors:** While not available immediately, the mention of SMRs hints at future-proofing. The U.S. is actively exploring micro nuclear reactors in the 1–5 MW range for remote power. Companies like Oklo, NuScale, and others are developing **microreactors** that could power data centers with 24/7 carbon-free energy. For example, Standard Power (a data center provider) plans to build the first SMR-powered data centers in the U.S. in collaboration with reactor vendors[25]. Microsoft has also hired experts to integrate SMRs for its data centers[26][27]. If ARCNet's 1 MW sites are designed with hybrid energy in mind, they could later plug in a microreactor module when that tech matures. An SMR around 1–5 MW could serve as **always-on baseload**, with solar and batteries handling peak shaving and backup. Regulatory approval for SMRs is complex, but smaller reactors intended for private sites are being streamlined by the NRC. The advantage is **unwavering reliability and zero-carbon power**[28]. A microreactor on-site eliminates dependence on the grid entirely and runs irrespective of weather – increasing uptime and making each node a self-sufficient power island. SMRs also offer security benefits (they're usually underground and passively safe, and reduce exposure to grid outages or fuel supply issues). The **drawback** is heavy regulatory oversight and public concern – likely only certain locations (e.g. a secure campus) would host a reactor. But conceptually, ARCNet could deploy a few SMR-powered nodes at critical locations (perhaps where solar isn't as viable) to ensure firm capacity. The **NEXUS-DC initiative** by Idaho National Lab notes that small nuclear is well-suited to modular data centers, being **highly scalable and reducing initial capital per unit** for medium data centers[29].

In summary, the sub-1.5 MW strategy aligns neatly with federal incentives and the clean energy transition. It transforms the data center network into a patchwork of green microgrids. This not only yields tax credits and potential grants (e.g. DOE funding for innovative clean-powered computing), but also provides a narrative of sustainability (important for winning over local authorities and customers). Many enterprises would be attracted to ARCNet's "AI compute delivered from 100% renewable micro data centers," which could command a premium or meet compliance (for example, government or research clients needing low-carbon compute). **Regulatory risk is lowered** as well – it's easier to get a thousand small permits than one mega permit that can be a single point of failure if denied.

One must note that operating so many renewable energy systems does introduce operational complexity – keeping panels clean, batteries maintained, etc. But these can be standardized and remotely managed. Each site could include a smart microgrid controller; under the IRA, data centers using qualified microgrid controllers may even get ITC incentives[30]. The ability to island from the grid in an emergency (using solar/storage/SMR) also provides resilience against wider power outages – a sustainability and reliability win.

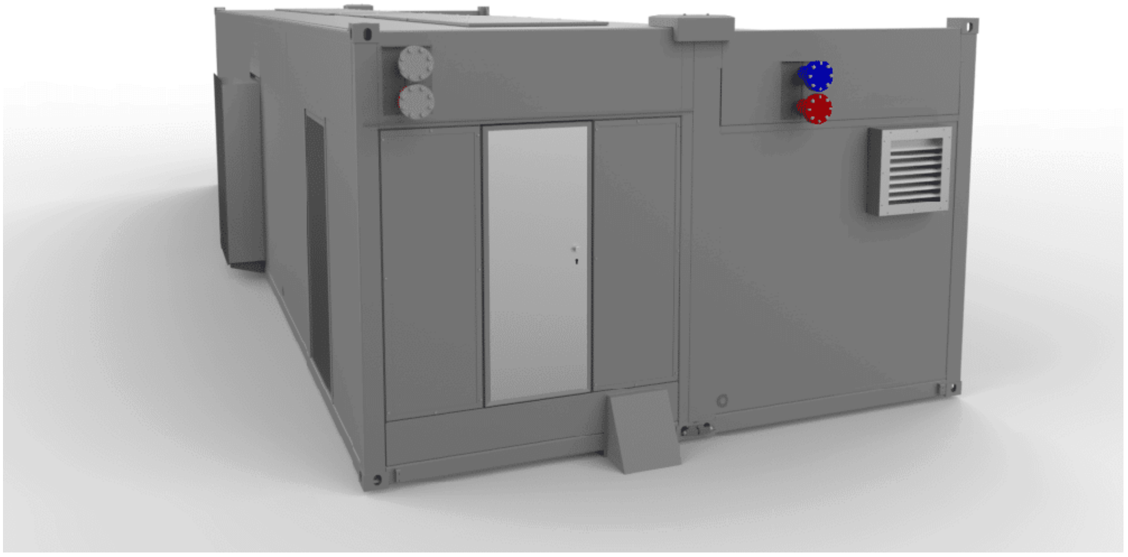


Figure: Example of a pre-fabricated 1.5 MW water-cooled data center module. Such modular units can be rapidly deployed on-site with integrated cooling and power equipment, enabling a scalable distributed architecture[31][16].

Multi-Tenancy, Reliability, and Auditability in a Federated Architecture

Transitioning from a few centralized data centers to a federated network of hundreds or thousands of nodes raises important considerations for multi-tenant operations, overall reliability, and auditability of the system:

Multi-Tenant Usage: In a hyperscale facility, multi-tenancy is straightforward – different customers' servers or VM instances share the same halls, and the operator enforces isolation via networking and access controls. In the distributed model, multi-tenancy could mean two things: (1) multiple tenants sharing a single 1 MW site, or (2) tenants distributed across different sites. Each 1 MW node could be partitioned (virtually) among tenants using cloud orchestration tools, just like a small Availability Zone. Modern software-defined networking and virtualization would allow ARCNet to carve up resources securely. However, the **network can become a bottleneck** in multi-tenant AI workloads if not managed well – HPC experts note that in multi-tenant AI/HPC environments, network performance and isolation are critical to efficiency[32]. Tenants might have different latency sensitivities; the system should allocate tenants to specific sites or groups of sites to meet their performance needs (e.g. a tenant running a large MPI job should be placed within one region, not scattered).

One benefit of many sites is **tenant locality options**. For example, a client worried about data sovereignty or low latency to a particular region could be assigned to a nearby 1 MW node, rather than their data going cross-country. It's analogous to edge cloud zones. Multi-tenant management software (like Kubernetes, Slurm for HPC, or cloud management platforms) would need to handle scheduling across geographically distributed nodes – essentially a federated cloud. This is technically achievable; it resembles how cloud providers distribute workloads across regions, though here at a finer granularity. **Security isolation** is also aided by physical separation – high-security tenants could even get dedicated nodes (1 MW pod just for them) which is easier to arrange when capacity comes in small increments.

Reliability and Uptime: With hyperscale, each site is designed for high availability (redundant power feeds, N+1 cooling, etc.), but if it fails, a huge chunk of capacity is lost. The federated model inherently provides *geographical redundancy*: one site's outage has limited blast radius. This is great for reliability of the *overall service*. ARCNet can promise a higher aggregate uptime – even if a few sites are down, the majority are up. Workloads could be live-migrated or quickly restarted on other nodes if one fails (assuming data is replicated or accessible). There's a parallel to content delivery networks: they use many nodes so that failures are localized and self-heal.

That said, coordinating reliability across sites means robust **orchestration and monitoring**. Each site must be equipped with sensors, remote reboot capabilities, and possibly an on-call maintenance contract. Sophisticated software will monitor performance and health of each node, performing automated failover of jobs. For tightly coupled HPC jobs, failure of one node (or site) mid-run is problematic; solutions include using checkpointing (periodically saving state so it can restart on a new node if needed) and using error-tolerant algorithms. Also, because the system relies on network connectivity, network outages between sites need contingency – if a site becomes isolated (network partition), its jobs might pause or run in degraded mode until connectivity resumes. **Overall, reliability at the system level should improve** due to no single point of failure, but application-level reliability will depend on building resiliency into software.

Auditability: In a multi-tenant, distributed setup, **auditing** refers to tracking resource usage, data access, and ensuring compliance across all sites. This is more complex than in one data center. ARCNet will need a unified logging and auditing system that aggregates logs from every node. For billing purposes, tracking each tenant's CPU/GPU hours per site is necessary (similar to cloud billing, but now across many edge locations). Fortunately, tools exist to centrally manage this kind of data. Another aspect is **data auditability** – some customers may require proof that their data was processed only in certain jurisdictions or that proper security controls were in place at each location. This would require certifying each 1 MW site to relevant standards (SOC2, ISO 27001, etc.) just as one would a larger data center. That's a lot of certifications, but if the modules are identical in design, the compliance process can be templatized.

Audit logs should capture which tasks ran where, who launched them, and any anomalies. The **federated nature introduces new audit challenges**: for instance, if there's an investigation (security or compliance), one might have to collect evidence from dozens of sites. Automated log collection and SIEM (Security Info and Event Management) tools would be critical to have in place from day one.

Security Isolation: Multi-tenant HPC also brings security concerns like potential side-channel leaks or interference if two tenants share hardware. In a big center, the solution might be strict job scheduling or hardware partitioning. In the distributed model, ARCNet could choose to **assign entire sites to sensitive tenants** (essentially single-tenancy at the physical level for those who need it), thus achieving strong isolation. Meanwhile, less sensitive workloads can co-exist on shared nodes with robust containerization/VM isolation. This flexibility can be a selling point: e.g., a government client could be given exclusive use of a particular group of nodes that are known to be on U.S. soil, powered by clean energy – meeting both security and sustainability requirements.

Finally, **auditing energy usage and carbon footprint** might be important for ESG reporting. With so many solar installations, ARCNet can audit and report precisely how much of each job's energy came from renewables vs grid. This level of detail could enable per-tenant carbon accounting (something enterprises increasingly want).

In summary, a federated architecture demands a "single pane of glass" management of a *distributed cloud*. Technologies from cloud computing (automation, orchestration, zero-touch provisioning) will be leveraged to make 1000 sites behave like one. It's challenging but achievable – essentially building a distributed cloud similar to how content delivery networks or telecom edge clouds operate. Once in place, it offers **granular control**: more ways to optimize (by moving workloads around for performance or cost) and more ways to isolate or dedicate resources as needed. The key is heavy investment in software – both scheduling algorithms that understand latency and energy, and monitoring systems that maintain security and performance visibility across the entire federated network.

Examples of Distributed Supercomputing (<5 MW Nodes)

While the extreme scale of ARCNet's plan is novel, there are precedents for **distributed supercomputing architectures with relatively small nodes**:

- **DiRAC (UK):** The DiRAC HPC facility in the UK is a "distributed supercomputer" spanning four university sites (Cambridge, Durham, Edinburgh, Leicester). Each site hosts a part of the overall HPC capability, and together they serve scientific users as a unified resource[33]. The systems are networked via national research networks, though jobs typically run on one site at a time. Notably, the new DiRAC-3 upgrade (circa 2021) increased performance 3–5× and made the distributed system 10× more energy efficient[34][35]. Each DiRAC site is on the order of a few hundred kilowatts to a couple MW – demonstrating that multi-site HPC can work for scientific computing. The UK's model shows how a federated setup can support diverse research workloads, and emphasizes energy efficiency improvements (something ARCNet seeks via solar power).

- **XSEDE / Open Science Grid (USA):** In the U.S., the XSEDE program (now succeeded by ACCESS) linked multiple supercomputers at different universities and national labs into a federated ecosystem. While not a single cluster, it allowed researchers to use a common interface to access compute time on various machines nationwide. Similarly, the Open Science Grid federates computing resources across over 100 sites (including many clusters well under 5 MW) for high-throughput computing in fields like particle physics. These are examples of *federation software* enabling distributed resources to function somewhat cohesively. They show that from a user perspective, distributed resources can be made to feel unified, though workloads are often embarrassingly parallel in these cases.

- **Distributed "Grid" Computing Projects:** Projects like Folding@home or SETI@home effectively create a distributed supercomputer out of many small nodes (PCs/GPUs in volunteers' homes). At its peak in 2020, Folding@home reached ~2.5 exaFLOPS by aggregating countless machines[36]. This highlights that enormous aggregate performance is possible with a distributed network – albeit for very loosely coupled tasks. ARCNet's scenario is more controlled (dedicated 1 MW nodes) but it's conceptually similar in harnessing many pieces for a collective goal.

- **Commercial Edge Networks:** Companies like Cloudflare and StackPath run distributed networks of data centers (though smaller scale in compute) to provide services. While not HPC, they manage thousands of locations as one infrastructure. Also, some blockchain cloud projects attempt to use geographically distributed servers as one "supercomputer" marketplace (e.g. Golem, etc.), though these are early-stage.

- **Modular Mini-Supercomputers:** Some vendors (HPE, Penguin Computing, etc.) have offered containerized HPC clusters (250 kW–1 MW modules) for quick deployment near data sources. For instance, HPE's Performance Optimized Data Center (POD) units can house high-density compute in a shipping container form factor, which Contour (cited earlier) integrates for clients[37]. These often serve remote or tactical HPC needs (like military deployments or oilfield analytics). While typically used standalone, one could deploy multiple and network them if needed.

- **Nuclear-Powered Data Center Pilots:** As mentioned, Standard Power plans SMR-powered data centers and is likely not building a single 100 MW reactor, but rather starting with a small reactor (perhaps ~5–10 MW) for a smaller facility[25]. This suggests a model of smaller independent data center units each with dedicated clean power.

Overall, direct analogues to ARCNet's vision (a tightly coupled distributed *supercomputer*) are limited, because historically supercomputing favors centralization for performance reasons. However, the trend is shifting as costs and energy push people to consider distributed setups[38]. The European **PRACE** initiative created a persistent distributed supercomputing infrastructure across countries[38]. And the concept of **"Jungle computing"** (listed in supercomputing references[39]) explores using heterogeneous, geographically dispersed systems together. With advances in networking and intelligent schedulers, we are beginning to see more *geo-distributed HPC* research (e.g. scheduling algorithms specifically for geo-distributed HPC clusters[40]).

In practice, ARCNet might be pioneering a new scale of distributed HPC if they proceed. It would serve as a flagship example if successful, likely drawing on techniques from cloud computing, edge computing, and traditional HPC to make it work.

Cost-Benefit Comparison Table

The following table contrasts key factors between the **original hyperscale model (10×100 MW)** and the **proposed distributed model (≈1000×1 MW)**:

Aspect	**10×100 MW Hyperscale Data Centers**	**1000×1 MW Distributed (Rubin) Nodes**
Compute Performance	Excellent intra-site performance with low-latency, high-bandwidth networks. Tightly-coupled HPC jobs scale efficiently within each 100 MW campus. *(All GPUs are local, enabling full NVLink/InfiniBand speed.)*	High performance per node (Rubin rack packs ~8 exaflops) but reduced performance for jobs spanning multiple sites due to WAN latency[6]. Some large parallel tasks may run slower or need redesign. *(Nodes act like separate clusters; cross-site communication is the bottleneck.)*
Latency & Networking	Intra-datacenter latency of microseconds; easy to support MPI,	Inter-site latency of milliseconds; requires high-speed fiber links. Likely limits tightly

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
	synchronous AI training at scale. *(Unified fabric within campus.)*	synchronous workloads to one site or region. *(Relies on scheduling to keep related tasks local; network is the "weak link.")*
Redundancy/Resilience	Lower system-wide redundancy. A single site outage (power failure, etc.) can remove 100 MW (10% of capacity) at once. Some sites might be in same region (common failure domain).	High redundancy and fault tolerance. Each 1 MW node failure only 0.1% of capacity lost. Geographic diversity protects against regional outages. *(No single point of failure; system "self-heals" by shifting load.)*
Deployment Speed	Slower to deploy at scale. Each 100 MW facility ~18–24+ months to build and energize; sequential projects may take years. Potential delays for utility power and permits can idle completed builds[18][19].	Faster, parallel deployment. Modular 1 MW units installed in months each[16]; hundreds can be built concurrently. Can start delivering capacity incrementally (e.g. 100 sites live within a year) rather than waiting for huge builds to finish.
CapEx per MW	Economies of scale yield lower cost per MW. Best-in-class ~$6M/MW (even $3.6M/MW in some cases)[10]. Typical hyperscale ~$7–8M/MW. Large shared infrastructure (substation, cooling	Higher unit cost per MW due to lost scale. Estimated ~$8–12M/MW for small modular sites[12]. Some savings via prefab standardization, but still more duplication of equipment. *($ per MW improves if*

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
	plant) amortized over many MW.	*many identical units are produced, but still ≥ hyperscale.)*
OpEx and Energy Cost	Generally low marginal cost per kW due to bulk utility rates and efficient PUE (~1.2). However, relies on grid power purchases (subject to price volatility) and may incur demand charges for huge load. Significant cooling/water costs for large heat loads; diesel generator maintenance for backup.	Potentially lower energy costs via on-site renewables. Solar + battery can offset a large portion of consumption – essentially producing power at <$0.05/kWh after incentives. No demand charges if mostly self-powered; minimal transmission losses. PUE could be low since waste heat is smaller-scale and modules are optimized[15]. More sites means more staff/maintenance visits, but also unmanned operation is feasible.
Tax Incentives	Limited by project size. Projects >1 MW must meet labor rules for 30% ITC, otherwise only 6%[3]. A 100 MW solar farm would require compliance management. Depreciation and other incentives still apply, but no bonus ITC for being small.	Every site qualifies for **30% ITC** on solar/storage by being <1 MW[3]. No prevailing wage requirement needed for full credit. Aggregate ITC savings are huge (multiple small credits). Possibly can also get production tax credits or other local incentives at many sites.

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
Grid Interconnection	*Challenging.* Requires new substations and high-voltage lines. Multi-year queue waits common[20]. Each 100 MW load = major grid impact (utility upgrades $100M+). Risk of completed data center sitting idle for power (as seen in CA)[18][19].	*More flexible.* 1 MW can tie into existing distribution in many areas. Shorter interconnect process, some under “fast track” rules. Many sites can be located where spare capacity exists. Easier to work with numerous small utilities than to get one big allocation. If grid is not ready, sites can run on local generation (solar, future SMR) to some extent.
Permitting & Zoning	Large footprint (hundreds of thousands of sqft) – triggers extensive environmental and zoning reviews. May face community opposition (noise, water use, visual impact). Many diesel backup gensets require air permits; large water-cooled systems need water rights or reuse.	Small footprint (few containers + solar panels) – often can use existing industrial or rural sites with minimal new disturbance. Easier to get zoned (sometimes qualifies as “accessory use” to a solar farm or commercial site). Usually no significant air/noise emissions if using batteries. Low profile helps avoid NIMBY issues.
Sustainability	Can procure renewable energy via PPAs, but physically often uses grid mix (which may include fossil fuels). On-site	High sustainability profile. Each site can run primarily on solar energy with battery firming – possibly achieving 80–100%

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
	solar limited by space (100 MW load would need ~500+ acres of solar). Huge backup generators burn fossil fuel during outages/tests. Large energy use in one place could strain local environmental resources (water, etc.).	renewable supply. Small modular reactors in the future could provide clean baseload[27][28]. Minimal reliance on diesel (many sites might be designed diesel-free). Waste heat from each site is modest and could be repurposed locally (heating nearby buildings, greenhouses). Overall carbon footprint per compute is very low.
Multi-Tenant Capabilities	Centralized location simplifies multi-tenancy (all resources pooled). Can allocate servers/VMs to tenants on one big cluster; low network latency between tenants' workloads. Data kept on-site for easy access. However, any security breach or noisy neighbor can affect many customers due to shared infrastructure.	Federated design allows assigning tenants to specific sites or distributing across sites. Can guarantee certain tenants' data stays in certain regions for compliance. Isolation can be stronger (e.g. give a tenant a dedicated 1 MW node for high security). Networking for multi-tenant must be carefully managed – cross-site traffic for a tenant adds latency, so tenants with large inter-node communication might be kept within one site.

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
Management Complexity	Easier to manage physically – 10 locations vs 1000. On-site staff can handle issues, and centralized maintenance is possible. Monitoring systems cover one big facility at a time. Fewer external relationships (10 utilities, a few local governments).	Significantly more complex to coordinate. Requires sophisticated centralized monitoring, automation, and remote management for 1000 sites. No staff on-site; must rely on sensors and dispatch field technicians when hardware fails. Many more utility accounts, leases, and regulatory jurisdictions to handle. Strong software-driven ops (AI Ops) needed to scale management.
Auditability & Compliance	All data and processes in a few sites – easier to maintain a physical chain of custody and perform audits (though the scale is large). Compliance audits (SOC2, etc.) done per site. Energy usage audit is straightforward aggregate.	Must maintain audit trails across distributed sites. Need unified logging of who accessed what, where computations ran. Compliance requires consistency across sites (e.g. each node meeting security standards). However, finer-grained audit possible, e.g. per-tenant energy consumption and carbon footprint at each node can be tracked. Data jurisdiction can be audited by restricting which nodes were used. More moving

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
		parts, but also more data to leverage for audits.
Scalability	Scales in big chunks – adding capacity means building another huge facility. Could overshoot demand if a 100 MW build comes online before fully needed. Conversely, difficult to add just a little capacity quickly.	Highly granular scalability. Can add a few 1 MW modules at a time as demand grows. Even decommission or relocate nodes if needed (modular units can be moved). Much more flexible – scale out in step with demand geographically. Technology refresh can also be rolling (upgrade one node at a time) rather than forklifting an entire giant center.
Cost Summary	**CapEx:** ~$6–8M per MW (~$600–800M per 100 MW site); total ~$6–8B for 1GW. **OpEx:** Bulk power ~$0.05/kWh, PUE 1.2, ~$40M/year per site power cost. Some staff per site. **Tax**: Limited ITC (needs labor compliance), depreciation available.	**CapEx:** ~$10M per MW (with solar/storage, etc.); ~$10B for 1GW (1000 sites). ITC could reduce effective cost ~30% on solar portion. **OpEx:** Solar reduces grid purchase; effective power cost potentially <$0.03–0.04/kWh net. More sites to maintain, but minimal on-site personnel. **Tax:** 30% ITC on maybe half the capex (energy systems), plus

Aspect	10×100 MW Hyperscale Data Centers	1000×1 MW Distributed (Rubin) Nodes
		accelerated depreciation.

Table: Comparison of the 100 MW hyperscale model vs. 1 MW distributed node model. Hyperscale offers peak performance and some cost efficiencies, whereas the distributed approach provides superior resilience, renewable integration, and deployment flexibility, at the expense of network latency and management complexity.

Conclusion

Replacing 10 giant data centers with a federated network of ~1 MW Rubin-powered installations is **technically feasible** but involves clear trade-offs. The hyperscale model wins on raw performance for tightly coupled tasks and simplicity of operations, whereas the distributed model excels in adaptability, sustainability, and fault tolerance.

Crucially, the economics of the distributed approach can be favorable **if the renewable energy incentives are captured effectively**. The 30% ITC, when multiplied across hundreds of solar-enabled micro data centers, becomes a game-changer – it helps offset the higher per-MW build cost. Moreover, distributed energy sourcing (solar, batteries, future SMRs) insulates ARCNet from rising grid costs and aligns with a carbon-free vision that may attract customers and avoid potential carbon taxes or mandates.

On the technical side, ARCNet would need to invest heavily in network infrastructure and distributed systems software. The success of a federated supercomputer hinges on smart scheduling – ensuring jobs run where they can perform best (minimizing high-latency communication) – and on robust federation of data (so that needed datasets are available at the nodes that process them, perhaps through a distributed filesystem or replication strategy). Fortunately, emerging networking solutions (like NVIDIA’s Quantum InfiniBand over metro distances, or satellite-linked HPC networks) and cloud-native HPC techniques are making this more achievable.

The comparison shows that **for many modern AI workloads, which are often more loosely coupled (think ensemble model training, distributed inference, hyperparameter searches, etc.), a distributed network might perform sufficiently well** – and the latency downsides are outweighed by the ability to bring compute to the data and power sources. Traditional supercomputing tasks (like exascale simulations) would likely still prefer a single site with a massive cluster, but ARCNet’s focus (implied by use of Rubin AI architecture) seems to be AI factories and possibly cloud-like services, which are more amenable to distribution. In fact, NVIDIA’s design of the Rubin platform anticipates multi-rack and multi-site scale-out via InfiniBand and Ethernet fabrics[9], indicating the hardware and software ecosystem will support disaggregated deployments.

From a regulatory and strategic standpoint, the federated model positions ARCNet as a **leader in green computing and resilient infrastructure**. It avoids the risks of big delays and community pushback that many hyperscalers are now encountering. Instead of battling for a single 300 MW substation in Virginia, ARCNet can sprinkle 1 MW modules across many states, some perhaps at renewable generation sites or areas needing economic development (which could yield local grants or tax breaks too). It's a decentralized philosophy aligning with trends in energy (distributed generation) and computing (edge cloud).

In conclusion, while challenging, the replacement of 100 MW behemoths with 1 MW federated nodes appears **economically viable when accounting for ITC savings and renewable energy cost reductions**, and **technically sound** provided significant engineering effort is devoted to networking and orchestration. The choice may ultimately depend on ARCNet's workload profile: if ultra-low-latency coupling is paramount, a hybrid approach might be used (e.g. a few regional hubs of 5–10 MW that aggregate some nodes in key locations). But if the workloads can be partitioned or are more AI/analytics-oriented, the federated supercomputer could deliver near-hyperscale performance with far greater agility and sustainability. This approach would be pioneering, turning ARCNet's infrastructure into something akin to an "AI computing CDN" – a competitive differentiator in the era where both compute power and clean energy are in short supply.

---

[1] [2] [5] [8] [9] NVIDIA Unveils Rubin CPX: A New Class of GPU Designed for Massive-Context Inference | NVIDIA Newsroom

https://nvidianews.nvidia.com/news/nvidia-unveils-rubin-cpx-a-new-class-of-gpu-designed-for-massive-context-inference

[3] How Can My Business Qualify for the Base 30% Federal Solar Tax Credit?

https://www.sunpeakpower.com/blog/qualifying-for-full-federal-tax-credit-for-solar

[4] [14] Inflation Reduction Act Creates New Tax Credit Opportunities for Energy Storage Projects - McGuireWoods

https://www.mcguirewoods.com/client-resources/alerts/2022/12/inflation-reduction-act-creates-new-tax-credit-opportunities-for-energy-storage-projects/

[6] [24] [36] [38] [39] Supercomputer - Wikipedia

https://en.wikipedia.org/wiki/Supercomputer

[7] [21] [22] [40] Task Scheduling in Geo-Distributed Computing: A Survey

https://arxiv.org/html/2501.15504v1

[10] [11] [13] DCD>Building at Scale: Hyperscalers aim to build at $6m per MW - DCD

https://www.datacenterdynamics.com/en/news/building-scale-hyperscalers-aim-build-6m-mw/

[12] Data Center Construction: Costs, Timeline, and Delivery Steps

https://www.mastt.com/guide/data-center-construction

[15] [16] [31] [37] Data Centres - Contour Advanced Systems

https://www.contouradvancedsystems.com/business-unit/datacenters/

[17] Data Centers Push the Limits of the Power Grid | ARC Advisory Group

https://www.arcweb.com/blog/data-centers-push-limits-power-grid-0

[18] [19] [20] [23] Silicon Valley data centers totalling nearly 100MW could 'sit empty for years' due to lack of power — huge installations are idle because Santa Clara can't cope with surging electricity demands | Tom's Hardware

https://www.tomshardware.com/tech-industry/data-centers-in-nvidias-hometown-sit-idle-as-grid-struggles-to-keep-up

[25] [26] [27] [28] Nuclear Powered Data Centers: Microsoft Bets on SMRs to Fuel the Cloud | Washington D.C. & Maryland Area | Capitol Technology University

https://www.captechu.edu/blog/nuclear-powered-data-centers-microsoft-bets-smrs-fuel-cloud

[29] [PDF] NEXUS-DC: Nuclear Energy eXpedition for US Data Centers

https://inldigitallibrary.inl.gov/sites/sti/sti/Sort_146874.pdf

[30] Energy Tax Benefits for Data Centers: In Brief | Congress.gov

https://www.congress.gov/crs-product/R48583

[32] Multi-Tenant HPC and AI: How the Network Can Make or Break the ...

https://www.hpcwire.com/2025/07/24/multi-tenant-hpc-and-ai-how-the-network-can-make-or-break-the-system/

[33] [34] [35] Distributed supercomputer DiRAC gets a £20m upgrade | Computer Weekly

https://www.computerweekly.com/news/252494409/Distributed-supercomputer-DiRAC-get-a-20-upgrade

Netflix guidlines

How and Why Netflix Built a Real-Time Distributed Graph: Part 1 — Ingesting and Processing Data Streams at Internet Scale

Netflix Technology Blog

Following

7 min read

·

Oct 17, 2025

1K

22

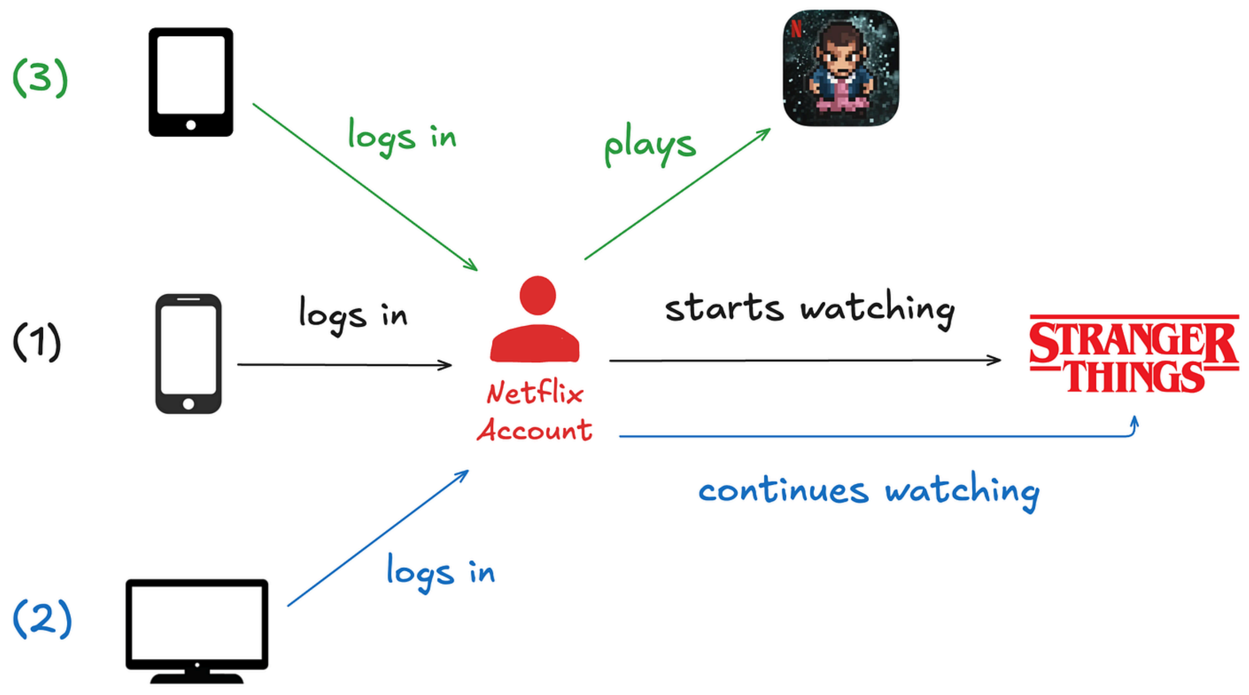
Authors: Adrian Taruc and James Dalton

This is the first entry of a multi-part blog series describing how we built a Real-Time Distributed Graph (RDG). In Part 1, we will discuss the motivation for creating the RDG and the architecture of the data processing pipeline that populates it.

Introduction

The Netflix product experience historically consisted of a single core offering: streaming video on demand. Our members logged into the app, browsed, and watched titles such as Stranger Things, Squid Game, and Bridgerton. Although this is still the core of our product, our business has changed significantly over the last few years. For example, we introduced ad-supported plans, live programming events (e.g., Jake Paul vs. Mike Tyson and NFL Christmas Day Games), and mobile games as part of a Netflix subscription. This evolution of our business has created a new class of problems where we have to analyze member interactions with the app across different business verticals. Let's walk through a simple example scenario:

Press enter or click to view image in full size



1. Imagine a Netflix member logging into the app on their smartphone and beginning to watch an episode of Stranger Things.
2. Eventually, they decide to watch on a bigger screen, so they log into the app on a smart TV in their home and continue watching the same episode.
3. Finally, after completing the episode, they log into the app on their tablet and play the game “Stranger Things: 1984”.

We want to know that these three activities belong to the same member, despite occurring at different times and across various devices. In a traditional data warehouse, these events would land in at least two different tables and may be processed at different cadences. But in a graph system, they become connected almost instantly. Ultimately, analyzing member interactions in the app across domains empowers Netflix to create more personalized and engaging experiences.

In the early days of our business expansion, discovering these relationships and contextual insights was extremely difficult. Netflix is famous for adopting a microservices architecture — hundreds of microservices developed and maintained by hundreds of individual teams. Some notable benefits of microservices are:

1. **Service Decomposition**: The overall platform is separated into smaller services, each responsible for a specific business

capability. This modularity allows for independent service development, deployment, and scaling.

2. **Data Isolation**: Each service manages its own data, reducing interdependencies. This allows teams to choose the most suitable data schemas and storage technologies for their services.

However, these benefits also led to drawbacks for our data science and engineering partners. In practice, the separation of business concerns and service development ultimately resulted in a separation of data. Manually stitching data together from our data warehouse and siloed databases was an onerous task for our partners. Our data engineering team recognized we needed a solution to process and store our enormous swath of interconnected data while enabling fast querying to discover insights. Although we could have structured the data in various ways, we ultimately settled on a graph representation. We believe a graph offers key advantages, specifically:

- **Relationship-Centric Queries:** Graphs enable fast “hops” across multiple nodes and edges without expensive joins or manual denormalization that would be required in table-based data models.
- **Flexibility as Relationships Grow:** As new connections and entities emerge, graphs can quickly adapt without significant schema changes or re-architecture.
- **Pattern and Anomaly Detection:** Our stakeholders’ use cases often require identifying hidden relationships, cycles, or groupings in the data — capabilities much more naturally expressed and efficiently executed using graph traversals than siloed point lookups.

This is why we set out to build a Real-Time Distributed Graph, or “RDG” for short.

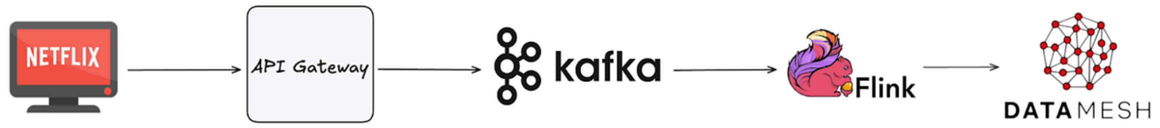
Ingestion and Processing

Three main layers in the system power the RDG:

1. **Ingestion and Processing** — receive events from disparate upstream data sources and use them to generate graph nodes and edges.
2. **Storage** — write nodes and edges to persistent data stores.
3. **Serving** — expose ways for internal clients to query graph nodes and edges.

The rest of this post will focus on the first layer, while subsequent posts in this blog series will cover the other layers. The diagram below depicts a high-level overview of the ingestion and processing pipeline:

Press enter or click to view image in full size



Building and updating the RDG in real-time requires continuously processing vast volumes of incoming data. Batch processing systems and traditional data warehouses cannot offer the low latency needed to maintain an up-to-date graph that supports real-time applications. We opted for a stream processing architecture, enabling us to update the graph's data as events happen, thus minimizing delay and ensuring the system reflects the latest member actions with the Netflix app.

Kafka as the Ingestion Backbone

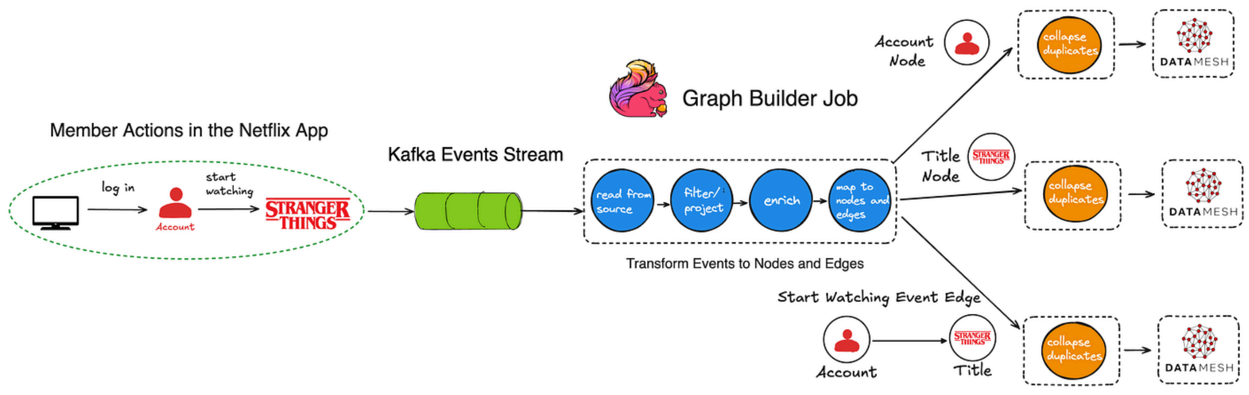
Member actions in the Netflix app are published to our API Gateway, which then writes them as records to Apache Kafka topics. Kafka is the mechanism through which internal data applications can consume these events. It provides durable, replayable streams that downstream processors, such as Apache Flink jobs, can consume in real-time.

Our team's applications consume several different Kafka topics, each generating up to roughly **1 million messages per second**. Topic records are encoded in the Apache Avro format, and Avro schemas are persisted in an internal centralized schema registry. In order to strike a balance between maintaining data availability and managing the financial expenses of storage infrastructure, we tailor retention policies for each topic according to its throughput and record size. We also persist topic records to Apache Iceberg data warehouse tables, which allows us to backfill data in scenarios where older data is no longer available in the Kafka topics.

Processing Data with Apache Flink

The event records in the Kafka streams are ingested by Flink jobs. We chose Flink because of its strong capabilities around near-real-time event processing. There is also robust internal platform support for Flink within Netflix, which allows jobs to integrate with Kafka and various storage backends seamlessly. At a high level, the anatomy of an RDG Flink job looks like this:

Press enter or click to view image in full size



For the sake of simplicity, the diagram above depicts a basic flow in which a member logs into their Netflix account and begins watching an episode of Stranger Things. Reading the diagram from left to right:

- The actions of logging into the app and watching the Stranger Things episode are ultimately written as events to Kafka topics.
- The Flink job consumes event records from the upstream Kafka topics.
- Next, we have a series of Flink processor functions that:

1. Apply filtering and projections to remove noise based on the individual fields that are present — or in some cases, not present — in the events.

2. Enrich events with additional metadata, which are stored and accessed by the processor functions via side inputs.
3. Transform events into graph primitives — nodes representing entities (e.g., member accounts and show/movie titles), and edges representing relationships or interactions between them. In this example, the diagram only shows a few nodes and an edge to keep things simple. However, in reality, we create and update up to a few dozen different nodes and edges, depending on the member actions that occurred within the Netflix app.
4. Buffer, detect, and deduplicate overlapping updates that occur to the same nodes and edges within a small, configurable time window. This step reduces the data throughput we publish downstream. It is implemented using stateful process functions and timers.
5. Publish nodes and edges records to Data Mesh, an abstraction layer that connects data applications and storage

systems. We write a total (nodes + edges) of **more than 5 million records per second** to Data Mesh, which handles persisting the records to various data stores that other internal services can query.

From One Job to Many: Scaling Flink the Hard Way

Initially, we tried having just one Flink job that consumed all the Kafka source topics. However, this quickly became a big operational headache since different topics can have different data volumes and throughputs at different times during the day. Consequently, tuning the monolithic Flink job became extremely difficult — we struggled to find CPU, memory, job parallelism, and checkpointing interval configurations that ensured job stability.

Instead, we pivoted to having a 1:1 mapping from the Kafka source topic to the consuming Flink job. Although this led to additional operational overhead due to more jobs to develop and deploy, each job has been much simpler to maintain, analyze, and tune.

Similarly, each node and edge type is written to a separate Kafka topic. This means we have significantly more Kafka topics to manage. However, we decided the tradeoff of having bespoke tuning and scaling per topic was worth it. We also designed the graph data model to be as generic and flexible as possible, so adding new types of nodes and edges would be an infrequent operation.

How and Why Netflix Built a Real-Time Distributed Graph: Part 2 — Building a Scalable Storage Layer

Netflix Technology Blog

Following

10 min read

·

Nov 14, 2025

345

6

Authors: Luis Medina and Ajit Koti

This is the second entry of a multi-part blog series describing how we built a Real-Time Distributed Graph (RDG). In Part 1, we discussed the motivation for creating the RDG and the architecture of the data processing pipeline that populates it. In Part 2, we'll explore how we designed the storage layer to handle billions of nodes and edges while maintaining single-digit millisecond latencies.

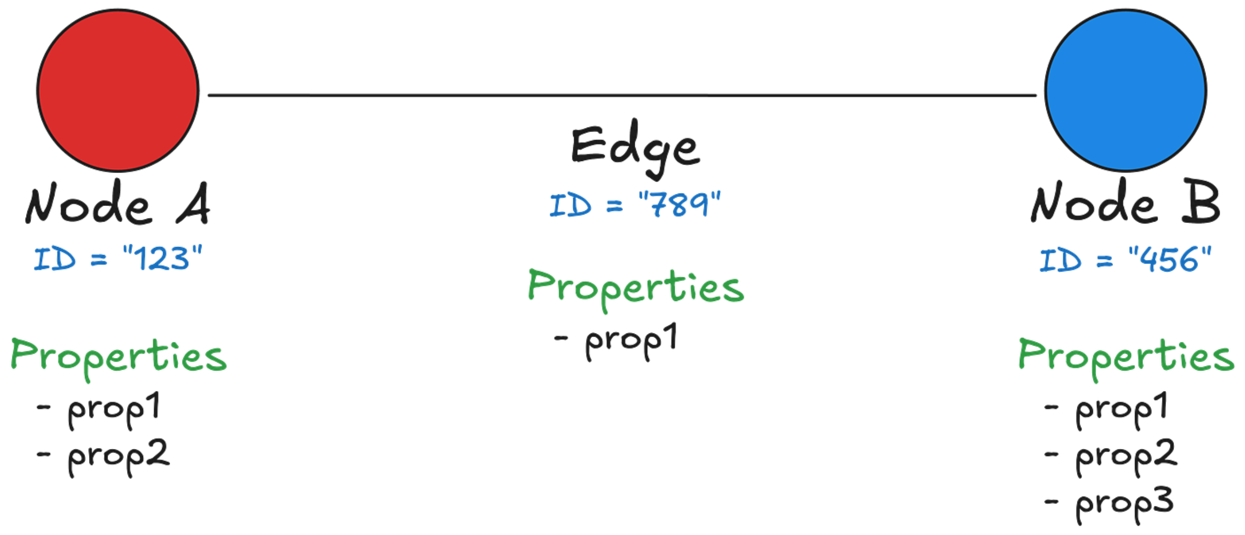
Introduction

In Part 1 of this series, we explained why Netflix needed a Real-Time Distributed Graph (RDG) and how we built the ingestion and processing pipeline using Apache Flink to transform streaming events into graph primitives. Now comes the following critical question: once we've created billions of nodes and edges from member interactions, how do we actually store them?

The Graph Data Model

Before diving into storage technology, let's clarify what we're actually storing.

Press enter or click to view image in full size

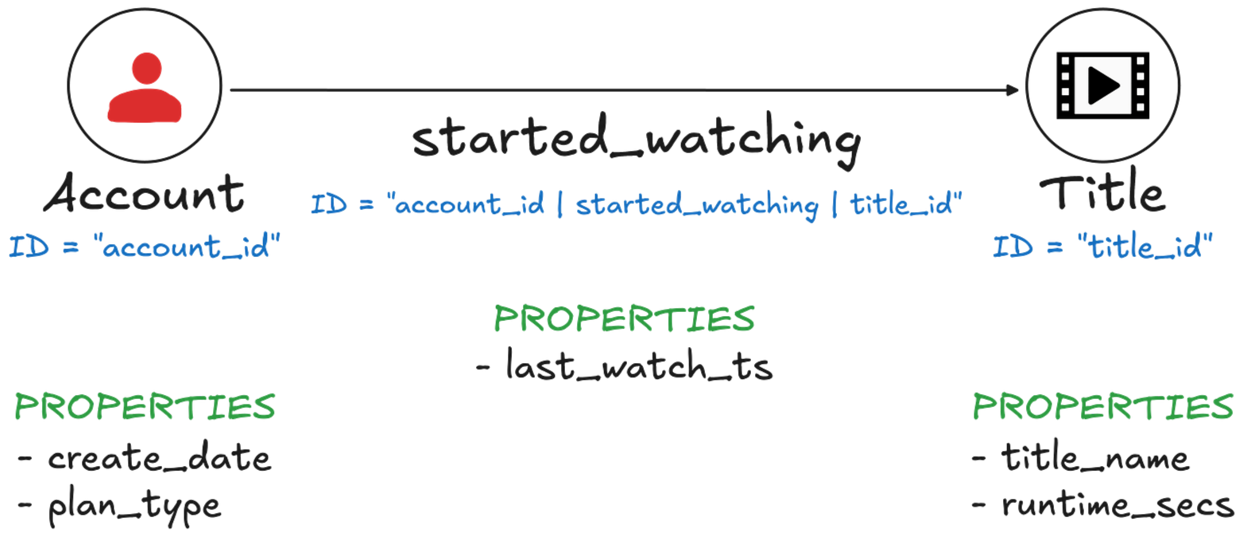


The RDG is a property graph consisting of:

- **Nodes**: Entities including member accounts, titles (such as shows/movies), devices, and games. Each node has a unique identifier and a set of properties containing additional metadata.
- **Edges**: Relationships between nodes, such as “started watching,” “logged in from,” or “plays.” Edges also have unique identifiers and properties, such as timestamps.

Let’s revisit our example from Part 1.

Press enter or click to view image in full size



When a member watches Stranger Things, we might create:

- An “account” node (with properties like creation_date and plan_type)
- A “title” node (with properties like title_name and runtime_secs)
- A “started watching” edge connecting the nodes above (with properties like last_watch_timestamp)

This simple abstraction allows us to represent incredibly complex member journeys across the Netflix ecosystem. But how do we efficiently store and query this graph structure at scale?

Graph Databases

The storage layer is the backbone of the RDG. It needs to efficiently store billions of nodes and hundreds of billions of edges while supporting both heavy read and write workloads with low latency.

In evaluating different storage options, we explored traditional graph datastores like Neo4J and AWS Neptune. While they do provide feature-rich capabilities around things like native-graph query support and data models to represent different types of graphs, they also pose a mix of scalability, workload, and ecosystem challenges.

Scale and performance limitations

Native graph databases, while excellent for exploring relationships, struggle to scale horizontally for large, real-time datasets. Their performance typically degrades with increased node and edge count or query depth (i.e., more hops). In our own early evaluations, Neo4j, for example, performed well for millions of records but became inefficient for hundreds of millions due to high memory requirements and limited distributed capabilities. AWS Neptune has similar limitations due to its single-writer, multiple-reader architecture, which presents bottlenecks when ingesting large data volumes in real-time, especially when spanning multiple regions. These limitations made them unsuitable for workloads demanding low-latency access, particularly in distributed systems where data sharding is required.

Workload complexity

Traditional graph database solutions often present significant challenges when faced with the demands of Netflix's high-volume, real-time data environment. These systems are not inherently designed or optimized for the continuous, high-throughput event streaming and real-time data ingestion workloads that are critical to our operations, and they frequently struggle with query patterns that involve full dataset scans, property-based filtering, and indexing. This made it clear that we needed an alternative capable of handling our internet-scale data.

Internal support

At Netflix, we have extensive internal platform support for relational and document databases, compared to graph databases. Consequently, non-graph databases are also easier for us to operate. As a result, we found it simpler to emulate graph-like relationships in existing data storage systems rather than adopting specialized graph infrastructure.

Due to the reasons mentioned above, we ultimately decided that the options we evaluated wouldn't meet our requirements at Netflix's scale. Therefore, we turned instead to an internal platform specifically designed for this type of challenge: the Data Gateway Platform. More specifically, its Key-Value Data Abstraction Layer (KVDAL).

Why KVDAL?

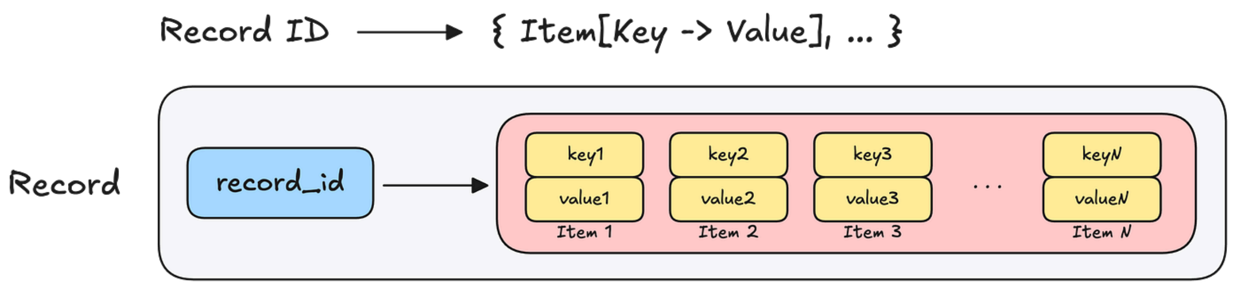
The Data Gateway Platform at Netflix provides several Data Abstraction Layers (DALs) that sit between applications and databases. These DALs support different data access patterns while delivering high availability, tunable consistency, and low latency, all without the operational overhead of managing underlying storage infrastructure directly.

For the RDG, we chose KVDAL as our primary storage solution. KVDAL is built on top of Apache Cassandra and provides a two-level map architecture that proved to be a perfect fit for graph data.

The KVDAL Data Model

KVDAL organizes data into **records** that are uniquely identified by a record_id. Each record can contain one or more **sorted items**, where an item is simply a key-value pair. This structure looks like the following:

Press enter or click to view image in full size



To query KVDAL, you:

- First, look up a specific record by its record_id
- And then (optionally) filter and retrieve only the subset of items you're interested in by their keys

This two-level architecture gives us both efficient point lookups and the flexibility to retrieve related data with minimal overhead.

Mapping Graphs to Key-Value Storage

The elegance of KVDAL for graph storage becomes clear when we examine how nodes and edges are mapped to records.

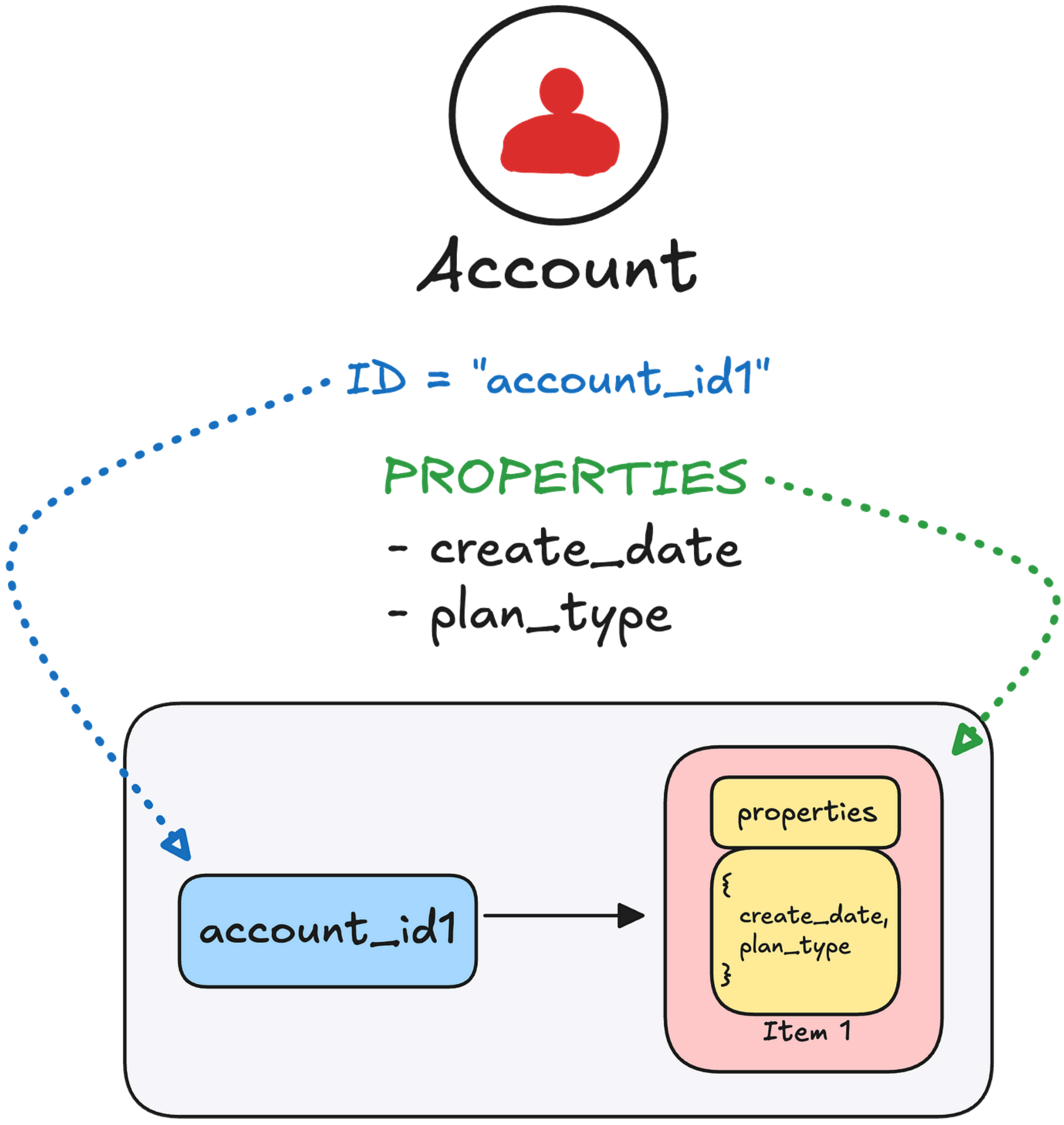
Storing Nodes

When storing nodes:

- The node's unique identifier becomes the record_id
- All properties for that node are stored as a single item within the record

For example, an account node might look like:

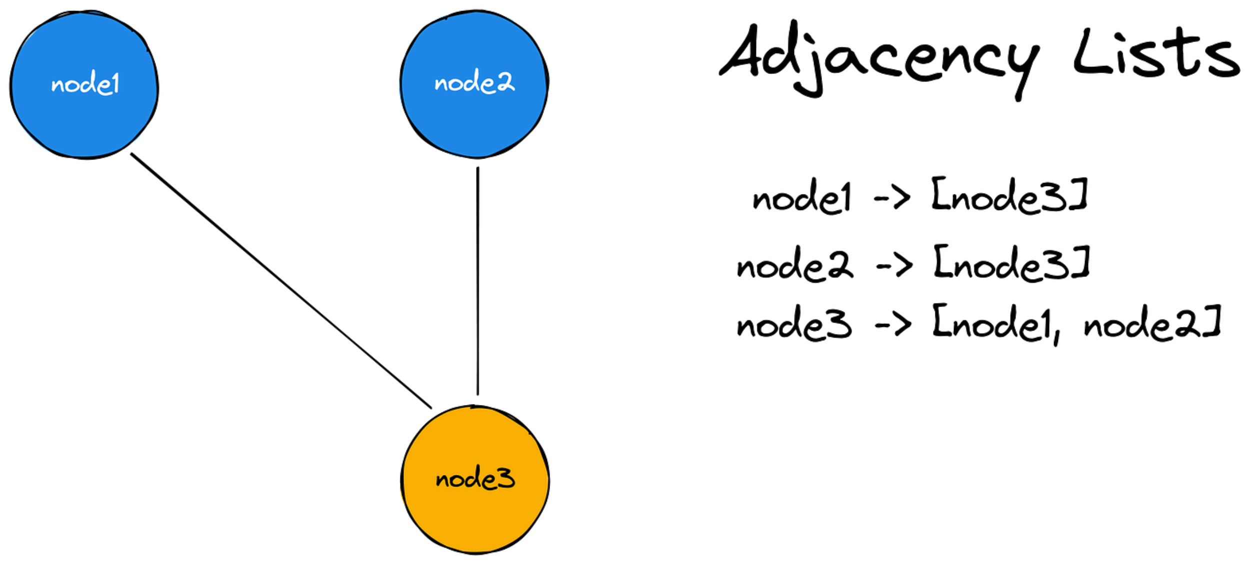
Press enter or click to view image in full size



Storing Edges as Adjacency Lists

When storing edges, we rely heavily on adjacency lists, which are a way of representing a graph where, for every node in the graph, we have a list that contains the nodes that are adjacent to it.

Press enter or click to view image in full size



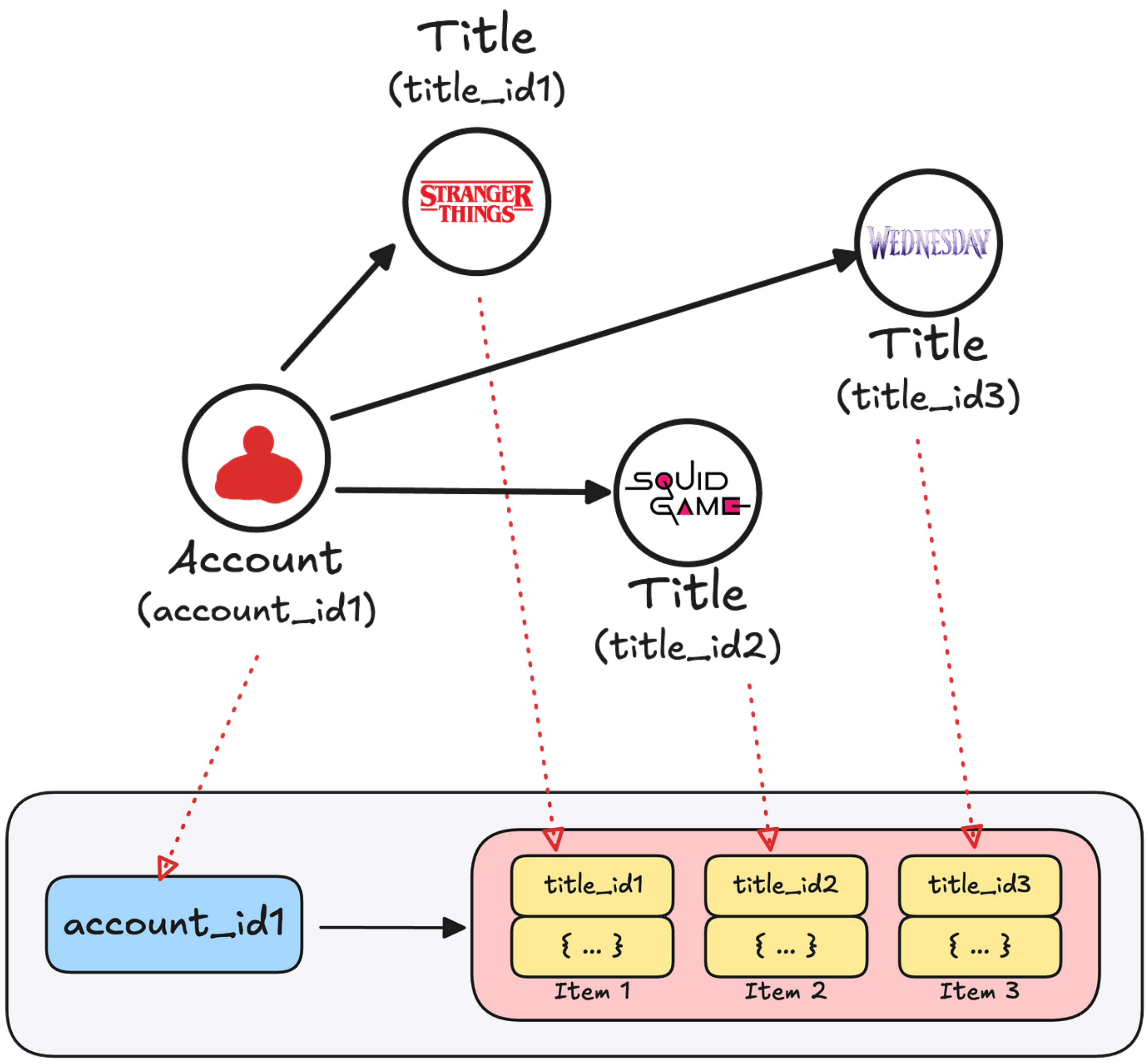
For our use case, this means that:

- The record_id represents the "from" node of the edge

- While the items within that record represent all the “to” nodes that the “from” node connects to
- Just like nodes, edges can also store properties within an item.

For example, if an account has watched multiple titles, the adjacency list might look like the following:

Press enter or click to view image in full size



This adjacency list format is vital for graph traversals. When we need to find all titles a member has watched, we can retrieve the entire record with a single KVDAL lookup. But we can also filter by specific titles using KVDAL's key filtering, avoiding the need to fetch and deserialize unnecessary data.

Data Lifecycle

The lifecycle of the node and edge data stored in RDG is relatively straightforward. In *Part 1*, we covered how we constantly ingest real-time event streams that are converted into node and edge objects before being stored in the graph. Whenever we ingest new nodes or edges that we haven't seen before, KVDAL simply creates new records for them. In cases where an edge exists with an existing "from" node but a new "to" node, we would create a new item within the existing KVDAL record. There are other times when we might ingest the same node or edge multiple times. When this happens, we simply overwrite the existing values within KVDAL, which means overwriting the entire record for nodes or the corresponding items for edges. This gives us the ability to continuously update the nodes and edges in the graph with the most recent property values, which is essential for keeping specific properties (e.g., timestamps) up to date.

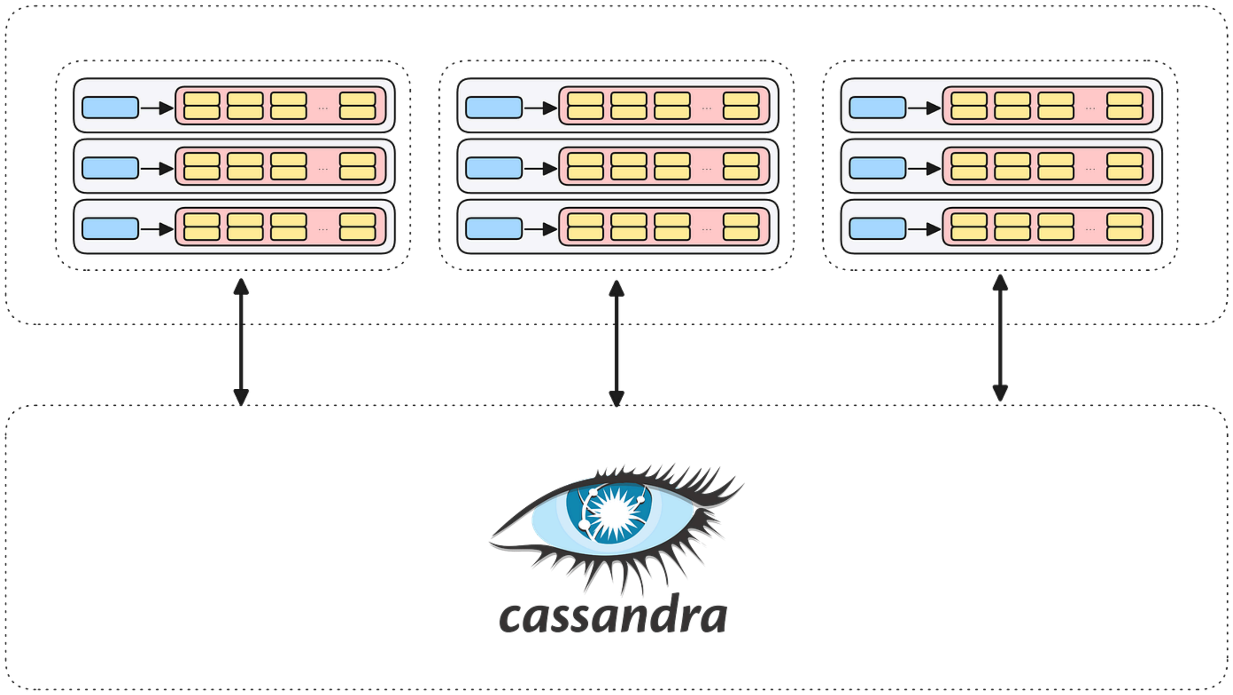
KVDAL can also automatically expire data after a specific period of time on a per-namespace, per-record, or per-item basis. This not only provides us with a lot of flexibility and fine-grained control over how we expire data from the graph on a per-node-type and per-edge-type basis, but it also allows us to create limits on how large different parts of the graph can get.

Namespaces: The key to flexibility and scale

One of the most powerful KVDAL features that we leveraged for RDG is the concept of **namespaces**. A namespace is similar to a table in a traditional database. It’s a logical grouping of records that defines where data is physically stored while abstracting away details of the underlying storage system.

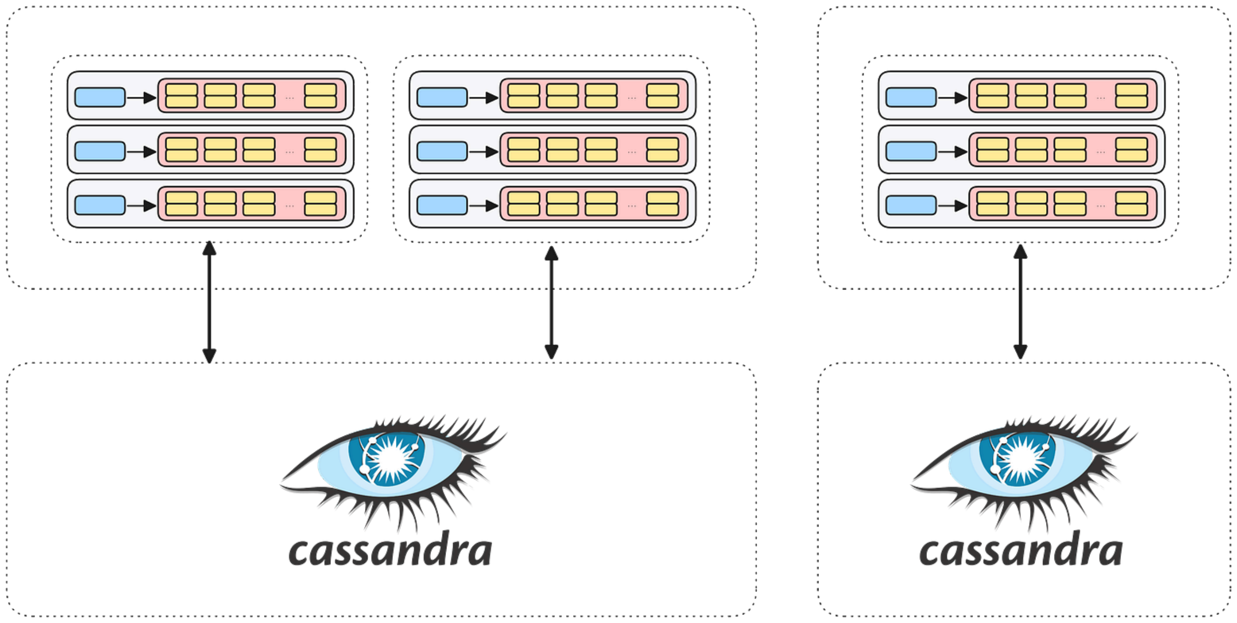
This means that you could start with a KVDAL setup where all of your namespaces are backed by the same Cassandra cluster.

Press enter or click to view image in full size



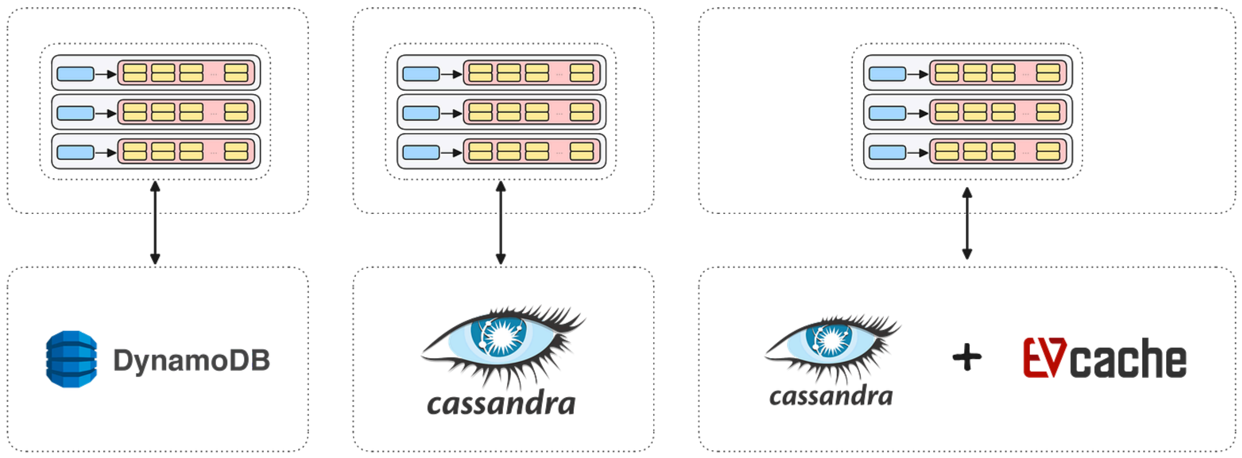
However, suppose one of these namespaces needs to store more data than the others, or it begins to receive a disproportionately larger amount of traffic compared to the others. In that case, we can simply move it to its own cluster as a way to isolate it from the rest of the setup, allowing it to be managed and scaled independently.

Press enter or click to view image in full size



Similarly, if your namespaces have different requirements around performance and scalability, we can further customize them by moving each namespace to its own dedicated storage setup in order to meet those needs.

Press enter or click to view image in full size



Because of this flexibility, KVDAL can scale to support **trillions** of records per namespace with **single-digit millisecond** access latencies.

One namespace per node and edge type

For the RDG, we provision a separate namespace for every node type and edge type in the graph. This might seem like overkill at first, but it provides us with a lot of benefits, including:

Independent Scaling: Different parts of the graph have vastly different data volumes and access patterns. By isolating each entity type into its own namespace, we can scale and tune each one independently from the others.

Flexible Storage Backends: Namespaces can be backed by different Cassandra clusters or entirely different storage technologies. For example:

- Low-latency data might use Cassandra with EVCache for caching
- Average throughput data with more lenient requirements around latency might use a single Cassandra cluster to host multiple namespaces
- Extremely high-throughput data might use dedicated Cassandra clusters per namespace

Operational Isolation: If one namespace experiences issues or needs maintenance, it doesn’t impact the rest of the graph. This isolation is critical for maintaining system reliability at scale.

Flexibility when adding new entity types

This namespace model also makes it straightforward to extend the RDG with new types of nodes and edges. When we need to support a new use case, say, adding support for tracking member interactions with live events, we would simply:

1. Define the new node and edge types in our data model
2. Provision new KVDAL namespaces for them
3. Update our Flink jobs to populate these namespaces (see Part 1 for details)
4. Deploy without impacting existing functionality

This flexibility was a key design goal from the beginning. We wanted to ensure the RDG could support unknown future use cases without significant re-architectural efforts.

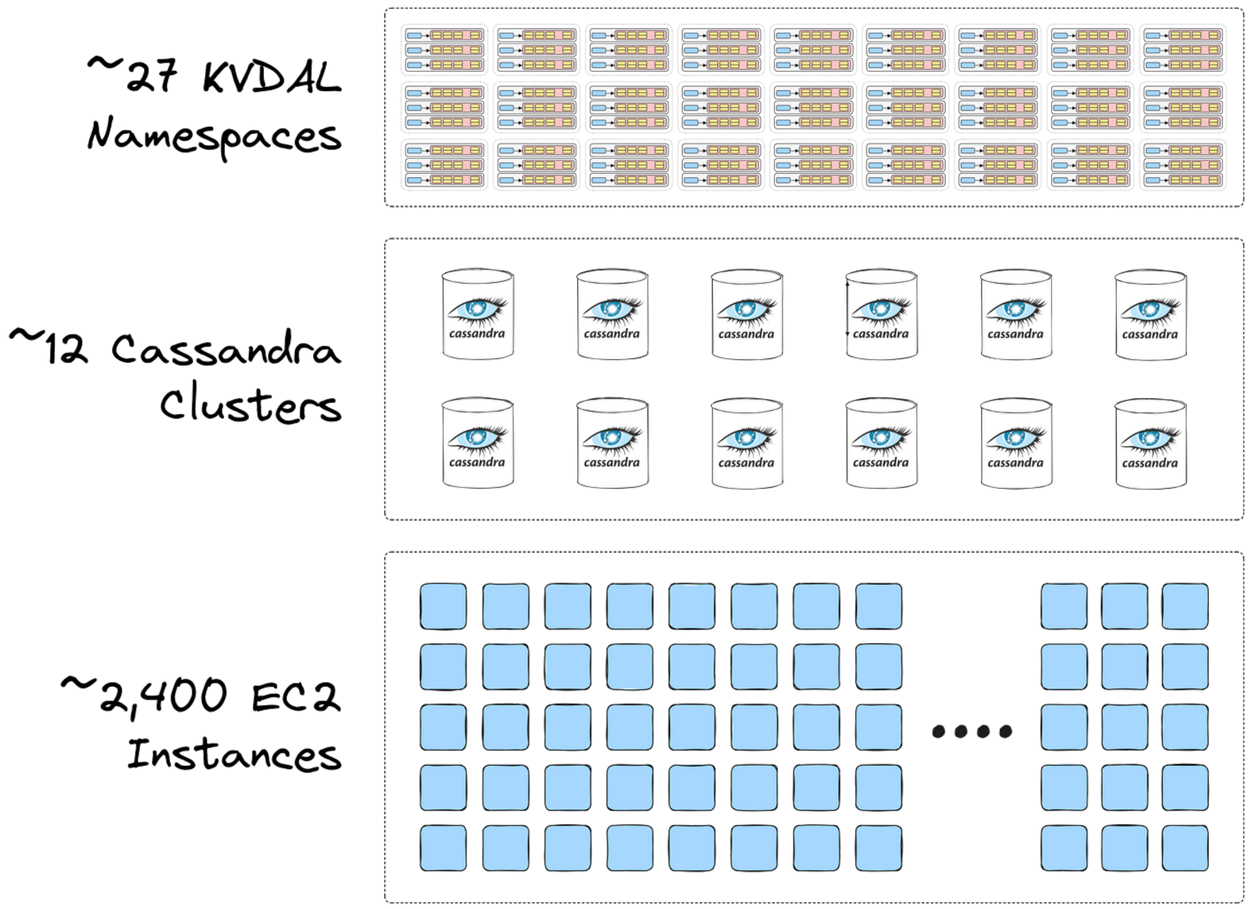
By the numbers

So how well does this architecture actually perform in practice?

Currently, our graph comprises over **8 billion nodes** and more than **150 billion edges** across all the different use cases we support. In terms of throughput, we have achieved concurrent reads and writes at sustained rates of approximately **2 million reads/sec** and **6 million writes/sec** across all nodes and edges in the graph.

To support this, we currently operate a KVDAL cluster that consists of approximately 27 different namespaces, backed by around 12 Cassandra clusters across 2,400 EC2 instances.

Press enter or click to view image in full size



These numbers represent one of the larger Cassandra deployments at Netflix. However, it's important to note that the numbers listed above aren't hard limits. Every component in our architecture can scale linearly. As the graph continues to grow, we can add more namespaces, clusters, and instances as needed.

Looking Ahead

Building the RDG's storage layer required significant engineering investment, and not everything was smooth sailing. Throughout its development, we encountered scaling challenges with various parts of the infrastructure and had to rework the data model several times. Through extensive iteration and performance tuning, we arrived at a solution that met our needs. These challenges also taught us valuable lessons about operating graph systems at scale.

Despite the bumps along the way, the RDG has delivered exceptional flexibility, scalability, and reliability. We were fortunate to build on the shoulders of giants by leveraging KVDAL's namespace model and the battle-tested reliability of Cassandra. Thanks to this robust infrastructure and unwavering support from our platform teams, we created a foundation that can evolve alongside Netflix's business needs.

Furthermore, our learnings benefited not only the RDG. In fact, they uncovered opportunities to generalize our solutions and, to that end, we partnered with our online datastores team to develop a broader graph abstraction platform that can support other graph use cases across Netflix. Stay tuned for a separate blog post that will provide more details about this exciting work.

—

Thanks for reading Part 2 of the RDG blog series. Stay tuned for Part 3, where we'll dive into the serving layer and explore how we make this massive graph accessible for use cases across Netflix.

Single-Tenant ARCNet

According to a document from **2025**, ARCNet's core plan already assumes a unified, centrally managed platform layer (portal/APIs + orchestration + model catalog + auditing) on top of Rubin-based "AI factory" infrastructure. Below is an updated version of the prior analysis **assuming ARC (Autonomous Resource Corporation / ARCNet operator) is the only infrastructure tenant**, and "users" are bringing **applications/workflows** that ARC runs *on their behalf*.

Updated Concept: Single-Tenant ARCNet Running Many User Applications

What changes (and what doesn't)

Still true

- **Federated 1 MW nodes** can be deployed as a tightly managed distributed system, with ARC operating the network as a cohesive unit (load balancing, failover, orchestration).
- ARC can standardize around **Rubin racks** (e.g., NVL144 delivering ~8 exaflops per rack) and leverage the same "AI factory" design philosophy (high density, low-latency internal networks per site).
- ARC's "trust layer" (audit trails, provenance, logs) remains central, and arguably becomes easier because ARC owns the entire operational envelope.

What changes materially

- "Multi-tenancy" becomes **multi-application**, not multi-customer infrastructure tenancy.
- You drop a whole class of complexity: noisy-neighbor isolation between unrelated tenants, per-tenant network segmentation across sites, and tenant-specific compliance boundaries at the hardware layer.
- ARC can optimize model placement, caching, routing, and replication **globally** because there's one operator and one control plane.

Operating Model: ARC as the Only Tenant

1) Users supply apps; ARC supplies the model + execution environment

Instead of customers "bringing their own cluster," ARC exposes:

- A **catalog of pre-optimized open models** and one-click deploy/fine-tune workflows.
- An orchestration layer that allocates resources across the network (now across many 1 MW sites instead of 10×100 MW).
- Full **auditability**: job-level logs, provenance, evaluation artifacts—explicitly positioned as core ARCNet value.

Result: users "feed in" applications (pipelines, datasets, prompts, eval suites, agent recipes), and ARC chooses *where* and *how* to run them.

2) Model selection becomes an internal ARC responsibility

Because ARC is the only infra tenant, ARC can run a **model selection router** as a first-class system component:

- Route per-request/per-task to the best available model (cost/latency/quality), including specialized inference hardware like **Rubin CPX** for long-context inference (ARC's roadmap highlights CPX's long-context focus and high throughput).
- Maintain "golden" evaluation harnesses and regression tests (ties directly into the "audit trail" ethos).
- Continuously retrain/fine-tune and roll forward/back model versions with logged deltas.

This is the cleanest way to deliver "application-specific LLMs" at scale without asking every user to become a platform engineer.

Architecture Implications for Federated 1 MW Nodes

A) Scheduling gets simpler (and more powerful)

With single-tenant infra, ARC can treat the network like one big fleet:

- **Regional packing:** keep tightly-coupled jobs within one site or metro cluster; only federate across sites when the workload is naturally partitionable.
- **Global failover:** ARC's roadmap explicitly anticipates the network behaving as a cohesive unit with multi-region failover; single-tenant control makes this easier to enforce consistently.
- **Energy-aware placement:** ARC can shift batch work to nodes with surplus solar/battery at that moment, without negotiating "tenant fairness."

B) Data plane becomes "replication-first," not "shared filesystem-first"

Because cross-site latency is unavoidable, the winning pattern is:

- Replicate "hot" application datasets and model artifacts to the sites where they'll run.
- Treat cross-site networking as a control/coordination fabric + occasional bulk transfers, not as a low-latency training fabric.

This aligns with ARCNet's emphasis on traceability and logged workflows—replication events and dataset versions are part of the audit trail.

C) Hardware specialization becomes a network-level advantage

ARC can mix clusters optimized for:

- training,
- inference,
- long-context inference (via CPX-class accelerators),

…and route workloads accordingly. This is much harder in true multi-tenant environments where customers demand static allocations.

Energy + 1 MW Design Still Works (and aligns with Rubin-era "AI factory" requirements)

Your internal, single-tenant model actually makes the hybrid-energy story cleaner because ARC can tune compute schedules to energy availability.

Key design elements from your own ARCNet docs remain directly applicable:

- **800 V DC architecture** for efficient high-density delivery and fewer conversion losses.
- **Integrated energy storage** at rack + facility level to buffer volatile AI power draw; the roadmap explicitly describes multi-layer storage and notes Rubin racks have substantially more integrated storage for surges.
- Hybrid supply strategy (solar + batteries + SMRs over time) is already positioned as the long-term baseload + sustainability plan.

If ARC is the only tenant, ARC can also implement “energy-first batch windows” (e.g., schedule large fine-tunes when local solar/battery is flush).

Updated: “Multi-tenancy” Section (Now Multi-Application)

What becomes easier

- No “tenant vs tenant” isolation battles. You can standardize:
  - identity & access (project-level),
  - quotas (application-level),
  - rollout governance (ARC-managed).
- Auditing is consistent because ARC owns the tooling end-to-end—matching the stated ARCNet goal that every job can produce an audit trail (datasets, params, outputs).

What you still must do (but it’s manageable)

- **Application isolation** (not tenant isolation): prevent one user application from exfiltrating another’s data.
  - Use strict project namespaces, encryption-at-rest per project, and signed artifact access.
- **Per-application cost controls**: since ARC is fronting the fleet, you need internal chargeback-style accounting (but it’s simpler than multi-tenant billing disputes).

Updated Cost-Benefit Table Row (Only the changed row)

Aspect	10×100 MW (Hyperscale)	1000×1 MW (Federated)
“Multi-tenancy” / platform complexity	Strong, but tenant isolation + scheduling + billing at huge scale	**Simpler under single-tenant ARC**: multi-*application* governance (namespaces, quotas, policy), ARC-managed model catalog and routing, unified audit trail controls

---

Bottom line (with the new assumption)

If ARC is the only tenant, the federated 1 MW architecture gets **meaningfully more plausible operationally**:

- You’re no longer trying to be “AWS-but-distributed,” with every customer demanding bespoke networking/compliance carveouts.
- You’re building a **single, optimized, audited AI execution fabric** that runs many user applications—with ARC controlling model selection, rollout, and placement.

If you want, I can produce a “v2 architecture spec” next: control plane / data plane, workload classes (train vs inference vs agentic workflows), and a recommended minimum viable rollout (first 10–20 nodes, then scale).

ARCNet V2 Architecture Spec

ARCNet V2 Architecture Spec (Single-tenant ARC operating multi-application workloads)

Assumptions

- ARC is the only infrastructure tenant; users submit *applications* (pipelines/workflows) and ARC selects models + placement.
- Nodes are ~1MW modular “AI factory” sites designed for rapid replication and hybrid energy.
- ARC’s platform layer includes a model catalog, orchestration, and audit trail as core primitives.

---

1) Node Architecture (what’s inside a 1MW site)

1.1 Compute unit

- **Rubin-era rack cluster** as the atomic “compute island” per site (1–N racks depending on power envelope).
- Each site is optimized to run *independently* at full speed (low-latency fabric inside the site), and participates in federation via WAN.

1.2 Power + energy layer

- **Hybrid energy**: solar + batteries (and future firm power like SMR/microreactor where appropriate).
- **High-efficiency DC power architecture** (your roadmap emphasizes 800V DC for efficiency and reduction of conversion losses).
- **Multi-layer buffering**: rack-level + facility-level storage to smooth AI load spikes (your roadmap calls out large integrated storage to handle surges).

1.3 Storage tiers (per site)

- **Tier 0 (hot)**: local NVMe cache for active model weights, KV cache, embeddings, and hot datasets.
- **Tier 1 (warm)**: object store / content-addressed artifact store for model versions, adapters, datasets, and evaluation outputs.
- **Tier 2 (cold)**: durable remote replication (regional or global) for compliance + recovery.

1.4 Networking

- **Inside-site fabric**: dedicated low-latency HPC/AI interconnect for scale-up/scale-out within the site.
- **Between-sites WAN**: bandwidth-optimized backbone (regional hubs + private links) designed for:
    - artifact replication,
    - dataset sharding/placement,
    - checkpoint transfers,
    - control-plane coordination
      (not for ultra-tight synchronous training across distant sites).

---

2) Control Plane (global brain)

Think of the control plane as "ARCNet OS": it decides *what runs where, on which model, with which data, under what policy*, and logs everything.

2.1 Core services

A) Identity, policy, and application registry

- **Project/application identity**: each user application has a project boundary (keys, policies, allowed datasets).
- **Policy engine** enforces:
    - dataset access rules,
    - allowed models,
    - geographic placement constraints (if needed later),
    - retention/audit requirements.

B) Model Catalog + Model Router

A first-class internal service that:

- tracks base models, fine-tunes, adapters (LoRA), evaluation scores, cost/latency profiles
- chooses the best model *per request* or *per workflow step*
  This aligns with ARCNet's concept of providing a platform layer (catalog + orchestration) rather than raw infrastructure.

Router inputs

- task type (chat, extraction, code, agent step)

- context length + latency budget
- safety/classification policy
- cost budget
- data locality (where the needed data already lives)

C) Global Orchestrator + Scheduler

- Accepts “Application Runs” (a workflow DAG).
- Breaks into jobs and places them across:
    - a single site (preferred),
    - a regional cluster,
    - multi-region (only for partitionable work).

Scheduling signals

- GPU availability
- network congestion
- storage/cache hit rate
- *energy availability* (solar/battery state)
- reliability (node health score)

D) Fleet & Site Ops (zero-touch operations)

- Provisioning: immutable images, declarative configs
- Health: node telemetry, automatic drain, remote remediation
- Upgrades: rolling updates site-by-site with canaries

E) Provenance + Audit Trail (always-on)

Your “trust layer” becomes simpler because ARC controls the entire execution fabric:

- Logs: model version, dataset hash, prompt templates, tool calls, outputs, evals
- Reproducibility: “rerun exactly” for regulated or scientific workflows
  This is a centerpiece of the ARCNet positioning.

---

3) Data Plane (movement + execution)

The data plane is everything that moves *tokens, tensors, datasets, and artifacts* efficiently.

3.1 Execution plane

- Runtime environments:
  - inference services (online)
  - batch jobs (fine-tunes, evals)
  - agent workflow workers (tools + state + memory)

3.2 Artifact plane (content-addressed)

- Every model, dataset slice, and eval output becomes an **addressable artifact** (hash/versioned).
- Enables:
  - deterministic rollbacks,
  - caching across sites,
  - global replication strategies.

3.3 Replication strategies

- **Model artifacts**: replicate "popular" models to all nodes; replicate niche models to regional nodes.
- **Datasets**: replicate based on application demand; keep sensitive datasets in a minimal set of trusted regions if needed.
- **Checkpoints**: store regionally; replicate globally for high-value runs.

3.4 Cache hierarchy

- Global index: "where is artifact X currently hot?"
- Site cache: NVMe and memory caches for weights + KV cache.
- Prefetching: when the scheduler assigns a job to a site, it triggers artifact prefetch before execution.

---

4) Workload Classes and Their Requirements

4.1 Training workloads (application-specific, not frontier)

Goal: fine-tune, adapt, or distill models for specific applications.

Typical jobs

- LoRA / adapters
- continued pretraining on domain corpora
- distillation from larger teacher models
- evaluation-driven iteration

Placement

- Prefer **single site** (or a small regional cluster) for high-bandwidth training.
- Multi-site training only for:
    - asynchronous / parameter-server-like designs,
    - data-parallel with infrequent sync,
    - or “swarm training” research modes.

Data needs

- Strong dataset governance + hashing + lineage
- Checkpoint durability and resumability

4.2 Inference workloads

Split into **online** and **batch**:

Online inference (latency-sensitive)

- SLA-oriented routing (p95/p99)
- geo-aware placement (closer to user if needed)
- aggressive caching (weights, KV, embeddings)

Batch inference (throughput / cost-sensitive)

- schedule when energy is abundant
- run in lower-cost nodes with high solar/battery state

4.3 Agentic workflows (multi-step, tool-using)

This is where ARCNet’s “augmented intelligence” thesis pays off: the network runs *workflows*, not just prompts.

A concrete reference model is ROBIN’s discovery loop: literature search agents + analysis agents coordinating over iterative steps (hypothesis → experiment plan → analysis → refine).

Agentic runtime requirements

- long-lived state (memory, scratchpads, tool outputs)

- tool execution sandboxing
- retrieval (RAG) across curated corpora
- deterministic logging for audit/replay (critical for scientific workflows)

Placement

- Keep an agent's *entire loop* within one site when possible (reduces latency and state fragmentation).
- Replicate only compact artifacts across sites (summaries, embeddings, intermediate results).

---

5) Minimum Viable Rollout Plan (10–20 nodes → scale)

Phase 0: Reference Stack (0–2 "lab nodes")

Purpose: harden the software platform before field replication.

- Implement:
  - Model catalog + router (basic rules-based → learned later)
  - Workflow orchestration (DAG execution)
  - Artifact store (content-addressed)
  - Provenance/audit logging baseline
- Validate:
  - model versioning + rollback
  - reproducible workflow runs
  - remote fleet management

Phase 1: MVP Federation (10–20 nodes)

Topology recommendation

- 2–3 regions (e.g., West / Central / East)
- Each region: 4–8 nodes (so you can do regional balancing + failover)

What to ship

1. **Inference-first** (online + batch)

- easiest to distribute without cross-site tight coupling

2. **Fine-tuning as a managed service**
- single-site training, artifact replication
3. **Agentic workflows (v1)**
- a ROBIN-like workflow runner for multi-step applications (science, legal, ops), emphasizing audit + replay.

Hard requirements

- Global scheduler that is aware of:
  - cache locality
  - energy availability (battery/solar state)
  - node health
- End-to-end provenance:
  - every run produces an auditable bundle (inputs → model → outputs)

Operational target

- "N-1 site loss" tolerance inside each region (lose one node, continue service)

Phase 2: Regional Clusters (50–100 nodes)

- Add more regions and densify the best-performing ones.
- Introduce:
  - automated dataset replication policies
  - smarter router (quality/cost prediction)
  - canary deployments of model updates
- Expand agentic workflows:
  - tool marketplace
  - standardized "workflow templates" for verticals (bio, mobility, gov ops)

Phase 3: Network Effects Scale (200+ nodes)

- Treat the network as a compute marketplace internally:
  - jobs routed to where energy + compute + data are optimal
- Add "specialist nodes":
  - long-context inference clusters
  - training-optimized clusters
- Introduce stronger guarantees:
  - per-application SLOs
  - carbon accounting per workflow run

---

6) What I’d implement first (if you want the shortest path to value)

1. **Artifact store + provenance** (because it underpins everything)
2. **Model catalog + router** (delivers immediate “ARC manages model selection” advantage)
3. **Inference federation (10–20 nodes)** (fastest proof of distributed viability)
4. **Agentic workflow runner** using a ROBIN-style loop pattern (high differentiation; hard for hyperscalers to “copy paste” without your distributed audit + control narrative)

If you want, I can turn this into:

- a **component diagram** (control plane vs data plane)
- and a **deployment bill of materials** for a 1MW node (compute, storage tiers, networking tiers, energy tiers) aligned to your roadmap themes (800V DC + integrated storage + modular replication).

Reference Design A

According to a document from **2025**, ARCNet’s hardware blueprint is explicitly built around (1) an **end-to-end 800V DC** backbone (utility 13.8kV AC rectified to 800V DC at the facility level and fed directly to rack busways), (2) **multi-layer integrated energy storage** (rack ultracaps + site BESS, with Rubin-era racks including dramatically more integrated storage), (3) **45°C warm-water liquid cooling**, and (4) **repeatable site blueprints** that can be scaled site-to-site and populated “module by module.”

Below are two concrete 1MW reference designs + how I’d map each into a 10–20 node MVP rollout (and what evolves by node #50).

Reference Design A: 1MW “Single-Rack Node” (purest replication unit)

What it is

A **single rack-scale Rubin “AI factory” unit** treated as the smallest deployable site: one pad, one microgrid package, one rack, one warm-water loop, one WAN edge. This matches the roadmap’s “1MW+ per rack” target enabled by 800V DC distribution.

A1) Compute module

- **1 × ~1MW rack-scale system**
    - Choose a Rubin rack configuration appropriate for your workload mix (training-heavy vs inference-heavy; the roadmap highlights Rubin + Rubin CPX combinations and rack-scale NVL144-class performance)
- **Rack-level fast buffering**
    - Integrated **ultracaps/supercaps** to absorb millisecond spikes locally (required for GPU surge stability)

A2) Storage tiers (minimum viable)

- **Tier 0 (hot)**: local NVMe cache for weights shards, KV cache, embeddings, hot dataset slices
- **Tier 1 (warm)**: local object/artifact store (content-addressed) for:
    - model weights + adapters
    - checkpoints
    - eval artifacts + run manifests
- **Tier 2 (durable)**: regional replication target (not necessarily on-site)

A3) Networking tiers

- **Intra-site fabric**: minimal—because you only have one rack (still need top-of-rack switching and internal connectivity)
- **WAN edge**: dual uplinks if available (or one + LTE/5G failover), encrypted overlay, replication acceleration

A4) Energy + power (aligned to 800V DC)

- **Utility entrance** (where available) + metering/protection
- **Facility rectification to 800V DC + 800V DC busway into rack**
  - “13.8 kV AC rectified to 800 V DC at the facility level and fed directly… into the racks”
- **Facility BESS** sized for:
  - ride-through + smoothing (seconds/minutes)
  - optional islanding (if designed)
  - roadmap calls this “shock absorber” between DC and grid
- **Solar PV + EMS / microgrid controller** (site-dependent, but standardized interface)

A5) Cooling (warm-water)

- **45°C warm-water loop** with CDU/LDU and dry cooler/fluid cooler
  - roadmap explicitly references 45°C warm-water cooling for Rubin racks

Why this is the “purest replication unit”

- Lowest site-to-site variance (ideal for a fleet)
- Simplifies capacity planning: “add one node → add ~1MW”
- Great for inference + agentic workloads + single-node training loops
- Hardest part is procurement: you’re betting early on rack-scale availability and 800V/DC supply chain consistency

---

Reference Design B: 1MW “Multi-Rack Node” (more flexible early ramp)

What it is

A 1MW site composed of **2–4 sub-MW racks** (e.g., 250–500kW each) behind the same 800V DC + warm-water + BESS envelope. This is still a single “node” operationally, but you can populate it incrementally (“module by module”)—a pattern the roadmap itself uses for scaling capacity over time.

B1) Compute module (more mix-and-match)

- **2–4 racks totaling ~1MW**
    - Mix training-optimized racks with inference/long-context racks (Rubin CPX emphasis) if you want specialization early
- **Benefit:** you can start at 250–500kW live capacity per site and grow to 1MW without redoing permits/interconnect

B2) Storage tiers (stronger locality options)

Same tier model as A, but typically:

- Larger Tier 1 (more local artifacts)
- Optional “site-local dataset shard” service (because multi-rack sites can host more concurrent applications)

B3) Networking tiers (real intra-site fabric)

- **Intra-site fabric becomes important** now:
    - you’ll want a true low-latency internal network for multi-rack training runs and fast weight movement inside the site
- WAN stays replication-optimized (same philosophy)

B4) Energy + power (aligned to 800V DC, but more forgiving)

- Same **13.8kV AC → 800V DC** facility rectification + DC busways
- **BESS** often slightly larger (multi-rack surges can stack)
- More options for staged commissioning (bring one rack online while finishing the rest)

B5) Cooling (warm-water)

- Same 45°C loop requirement, but typically:
    - slightly more robust CDU capacity
    - more plumbing complexity (still standardizable)

Why this is easier early

- Procurement flexibility (you can deploy whatever sub-MW racks you can secure first)
- Easier “first 10–20 nodes” timeline because each site can deliver partial capacity quickly
- Better for early multi-application load diversity (some apps inference-heavy, some training-heavy)

MVP rollout mapping (10–20 nodes), and what evolves by node #50

Below is the **pragmatic rollout plan** I’d use with both designs.

Phase 1: First 10–20 nodes (what must exist Day-1)

Day-1 platform requirements (both designs)

These are the non-negotiables to make federation work without multi-tenant infra complexity:

1. **Control plane basics**
- API/portal entrypoint + app/project registry
- Global scheduler (simple rules-based placement is fine)
- Model catalog + router (initially rules-based)
- Artifact store + replication controller (content-addressed)
- Fleet management + remote provisioning
- Provenance/audit logging “always on” (run manifests)
  This matches the roadmap’s “portal/APIs + orchestration + model catalog” platform layer
2. **Site “join” automation**
- Plug in node → attest identity → pull baseline images → prefetch core model set → start serving
3. **Energy-aware signals (minimum)**
- each node publishes: available power headroom, BESS state-of-charge, thermal headroom
- scheduler uses this *at least* for batch job gating
4. **800V DC + energy buffering**
- End-to-end 800V DC distribution is part of the design foundation
- Rack-level ultracaps + site BESS (smoothing + ride-through)
5. **Warm-water cooling**
- Treat 45°C warm-water loop as standard kit per site

Recommended mix for the first 10–20 nodes

- **If rack-scale 1MW racks are reliably available:** go heavier on **Single-Rack Nodes** (cleanest fleet).
- **If supply is uncertain (most likely early):** deploy **Multi-Rack Nodes first** to ramp capacity, then standardize toward single-rack later.

Suggested split (practical):

- Nodes #1–6: Multi-rack (fastest to get capacity live)
- Nodes #7–20: whichever procurement favors, but keep the *site envelope* identical (800V DC + BESS + warm-water + replication kit)

Phase 2: By node #50 (what can be phased in and becomes "v2 complete")

Platform upgrades (high leverage at scale)

- **Learned model router** (predict quality/cost/latency per task; not just rules)
- **Cache-aware placement** (prefer nodes that already have weights + datasets hot)
- **Replication policy automation**
  - "hot models everywhere"
  - "warm models per region"
  - dataset sharding rules per application
- **Canary + rollback pipelines** (model + runtime images)
- **Regional backbone optimization** (regional hubs, QoS lanes for replication vs serving)

Site upgrades (hardware/ops)

- **Tighter energy scheduling**
  - batch training windows keyed to solar/BESS state
  - auto-throttle non-urgent jobs under grid constraints
- **More aggressive "module-by-module" scaling**
  - Multi-Rack nodes become your flexible capacity "buffers"
  - Single-Rack nodes become your standardized, repeatable workhorses
    The roadmap explicitly frames scaling capacity incrementally ("module by module") and building repeatable blueprints across sites

---

Quick decision rules: Single-Rack vs Multi-Rack

Choose Single-Rack Node when:

- you want maximal replicability and minimal site variance
- your dominant workloads are inference + agentic workflows + single-site training loops
- you can procure rack-scale units reliably

Choose Multi-Rack Node when:

- you need faster early deployment and partial go-lives
- procurement is uncertain and you want mix-and-match flexibility
- you want internal site diversity (training racks + inference racks) without multi-site coupling

---

If you want, I can turn this into **two one-page "spec sheets"** (like vendor-ready briefs) with:

- target footprint, interfaces (power/network/water), commissioning checklist
- "minimum required" vs "optional" components
- acceptance tests per node type (power stability tests, thermal tests, replication tests, inference SLA smoke tests)

Spec Sheet A — ARCNet 1MW “Single-Rack” Node

According to documents from **2025**, here are two **vendor-ready, one-page spec sheets** for ARCNet 1MW nodes, aligned to your roadmap themes (**800V DC + integrated storage + modular replication**) and integrating **Power Innovations International (Pii) Tower of Power SST** and **Yotta Block** storage.

Spec Sheet A — ARCNet 1MW “Single-Rack” Node (Pure Replication Unit)

1) Purpose / Performance Target

- **Replication unit:** smallest “copy/paste” building block for rapid site multiplication.
- **IT load: 1.0 MW** max at rack, delivered over **800V DC** backbone (facility rectified to 800V DC, fed into busways/racks).
- **Cooling target:** warm-water liquid cooling compatible with **~45°C** supply architectures.

2) Target Footprint (Typical)

- **Compute:** 1× high-density AI rack (OCP/ORV3-class or equivalent), up to 1MW IT.
- **Power conversion (adjacent):** 1× 19” rack stack housing **Pii Tower of Power** modules and DC distribution.
- **Integrated storage (adjacent / roof / perimeter):** Yotta Block arrays + LVDC combiner cabinets.

> **Why this is “single-rack”:** one compute rack is the *replication identity*; BOS is standardized sidecar(s) with fixed interfaces.

3) External Interfaces (Power / Network / Water)

Power (facility side)

- **Service:** 480Vac 3-phase (typical), sized for 1MW IT + losses + charging (recommend ≥1.25MVA service headroom).
- **Primary architecture:** facility AC → **Pii Tower of Power (SST)** → **800V DC** busway into rack PDU/DC distro.
- **Pii Tower of Power capabilities (key constraints):**
  - Up to **<300 kW** per “tower,” multi-directional power management. (Power Innovations International)
  - Accepts **120V 1-ph up to 480V 3-ph AC**, and **DC/AC sources up to 800V** (solar/batteries). (Power Innovations International)
  - Outputs include **800V** and **±400V DC** plus **12/24/48V** options. (Power Innovations International)

    - **Grid-tied + islanded** operation and multi-port I/O. (Power Innovations International)
    - “Towers” can be **paralleled to scale**. (Power Innovations International)
- **1MW sizing rule: 4× Tower of Power** stacks (N) + optional 1× (N+1) depending on availability and desired redundancy.

Network

- **Uplinks (minimum):** 2× diverse fiber uplinks to site spine (A/B paths).
- **OOB management:** separate 1/10GbE mgmt drop to ARC NOC/edge control plane.

Water / Cooling

- **Facility liquid:** warm-water loop compatible with ~45°C supply and CDU distribution.
- **Rack:** direct-to-chip cold plates via CDU (rack or row-level).

4) Minimum Required vs Optional Components (Bill-of-Materials View)

Minimum Required

- **Compute**
    - 1× 1MW-class AI rack (compute + NVSwitch/scale-up fabric as required)
    - 1× rack management controller + node BMC integration
- **Power (800V DC backbone)**
    - Facility rectification/800V DC distribution architecture per ARCNet plan
    - **Pii Tower of Power SST modules** sized to ≥1MW total (e.g., 4×<300kW units) (Power Innovations International)
    - DC busway + rack DC PDU / branch protection
- **Storage tiers (energy)**
    - **Rack-level fast buffer** (supercaps/ultracaps or equivalent) to absorb millisecond surges (per ARCNet integrated storage strategy).
    - **Yotta Block** distributed LV storage blocks (see optional sizing guidance below)
- **Cooling**
    - CDU + warm-water interface (45°C-class)
- **Controls**
    - Node telemetry + safety interlocks (E-stop, leak detection, insulation monitoring for HVDC)

Optional / Phase-In

- **N+1 Tower of Power** for redundancy
- **Solar PV DC-coupled input** into Tower of Power (supports solar/battery DC inputs up to 800V). (Power Innovations International)
- **Expanded Yotta Block energy** for longer ride-through / peak shaving

- **Outdoor-rated rack** for Tower of Power with integrated thermal management (vendor option). (Power Innovations International)

5) Integrated Battery Storage — Yotta Block (How It Fits)

- **Device:** Yotta Block **48V nominal**, **1kWh** increments; LVDC architecture (<60V). (Solar Tech Solutions)
- **Electrical (datasheet highlights):** ~48V nominal, ~1kWh, 15S/1P LFP, max continuous current 20A. (Solar Tech Solutions)
- **Safety validation signal:** the attached UL9540A-style thermal runaway validation reports a **PASS** result for containment (no flames escaping; cheesecloth not charred).
- **Interface strategy:** aggregate Yotta Blocks on **48V DC** bus → **48V→400/800V DC/DC stage** into the node's HVDC backbone (this matches the "integrated storage + 800V DC" philosophy without putting large BESS inside the rack aisle).

Practical sizing: 1MW for 5 minutes ≈ 83 kWh ⇒ ~83 Yotta Blocks; for 15 minutes ≈ 250 blocks. (Use as a modular "dial," not a fixed requirement.)

6) Commissioning Checklist (Node-Level)

1. **Mechanical**
   - Rack leveling/anchoring; torque check; labeling; clearances verified
2. **Power**
   - Tower of Power module discovery + firmware baseline; verify multi-port configuration (Power Innovations International)
   - Energize HVDC busway; verify polarity/insulation resistance; branch breaker mapping
3. **Cooling**
   - Flush + pressure hold; CDU commissioning; flow and ΔT verification at 25/50/75/100% load
4. **Storage**
   - Yotta block strings recognized; LV combiner protection verified; DC/DC enable/disable tested
5. **Network**
   - Spine uplinks A/B; OOB reachable; time sync; logging/metrics path
6. **Software**
   - Node joins ARC control plane; pulls baseline images; runs smoke workloads

7) Acceptance Tests (Per-Node)

Power stability

- Step-load: 10%→90% IT load transient, measure 800V DC droop, recovery time, ripple (pass/fail thresholds set by rack PSU tolerance)

- Islanding: simulate grid loss; verify controlled shed or ride-through behavior via storage + Tower of Power mode switch (Power Innovations International)

Thermal

- 100% load for 60–120 minutes; verify GPU junction stability, CDU ΔT, no leak alarms

Replication

- Artifact replication test: push a container/model bundle; validate integrity + cross-node pull time SLA
- Chaos test: restart node services; verify automatic rejoin

Inference SLA smoke

- Fixed prompt suite; measure p50/p95 latency + error rate under sustained concurrency; verify "SLA green" at 30/60/90% utilization

---

Spec Sheet B — ARCNet 1MW "Multi-Rack" Node (Flexible Ramp / Easier Early Procurement)

1) Purpose / Performance Target

- **Ramp-friendly:** bring capacity online in increments (e.g., 250kW per rack) while keeping the same standardized energy and network interfaces.
- **Total node IT: 1.0 MW** across **2–6 racks** (configurable).

2) Target Footprint (Typical)

- **Compute racks:** 2–6× AI racks (e.g., 2×500kW or 4×250kW), phased in.
- **Shared power row:** 1–2× 19" racks for **Tower of Power** + HVDC distribution + metering.
- **Shared cooling row:** 1× CDU (row-level) sized for 1MW, with spare ports for growth.

3) External Interfaces (Power / Network / Water)

Power

- Same as single-rack: facility AC → **Tower of Power** → **800V DC** busway; modular towers paralleled. (Power Innovations International)
- **Distribution:** HVDC busway trunk with tap-offs per compute rack (supports "add rack" without re-engineering).

Network

- **Spine connectivity:** at least 2× diverse uplinks; leaf/spine inside node (multi-rack fabric).
- OOB management remains physically separate.

Water / Cooling

- 45°C-class warm-water loop compatibility, CDUs shared across racks.

4) Minimum Required vs Optional Components

Minimum Required

- **Power**
    - Tower of Power stacks sized for initial phase (e.g., 2×<300kW) with clear growth path to ≥1MW (Power Innovations International)
    - 800V DC distribution trunk + protection
- **Cooling**
    - Row CDU sized for "day-1 load + next increment"
- **Compute**
    - Minimum 2 racks (to validate multi-rack scheduling + fabric behavior)

Optional / Phase-In

- Additional compute racks (add in 250–500kW steps)
- Additional Tower of Power modules (add in <300kW steps) (Power Innovations International)
- Expanded Yotta Block capacity for stronger peak shaving / ride-through (Solar Tech Solutions)

5) Integrated Storage — Yotta Block (Multi-Rack Use)

- Use Yotta Blocks as **distributed "edge-BESS tiles"**: near-rack or roof-adjacent, aggregated per row.
- Same LVDC aggregation → DC/DC into HVDC backbone; scale storage with racks.
- Yotta datasheet attributes (1kWh increments, LVDC) support fine-grained scaling with node expansion. (Solar Tech Solutions)

6) Commissioning Checklist (Node-Level)

1. Commission **power row** first (Tower of Power modules, HVDC trunk, metering) (Power Innovations International)
2. Commission **cooling row** (CDU, leak detection, flow validation)
3. Bring up **rack #1 and #2**; validate scheduling + fabric

4. Add racks incrementally; repeat acceptance tests per added rack

7) Acceptance Tests (Per-Node Type)

Power stability

- HVDC trunk loading test with staged rack turn-on (ramp 0→250kW→500kW→750kW→1MW)
- Verify no nuisance trips; verify Tower of Power parallel sharing and telemetry coherence (Power Innovations International)

Thermal

- Verify row CDU capacity at each increment; confirm no hot-spot migration as racks are added

Replication

- Cross-rack artifact replication and “rack isolation” test (remove one rack; workloads reschedule)

Inference SLA smoke

- Latency under mixed tenancy-by-application (same ARC tenant), confirm predictable p95 as racks are added

---

Notes on Roadmap Alignment (Why these two designs match ARCNet’s themes)

- **800V DC backbone** is explicitly part of ARCNet’s reference architecture, with facility rectification feeding racks directly.
- **Integrated energy storage** is a first-class design principle (rack-level fast buffer + site/row storage), and Rubin-class designs emphasize more integrated energy storage for stability.
- **Modular replication** is operationalized as (a) “single-rack identity” replication, or (b) “multi-rack ramp” with the same standardized interfaces.

Included files (as requested):

Tab 8