From cloud AI to hybrid AI: The rise of model cascading

The next phase of enterprise AI is likely to be more distributed than the first. Rather than routing every workload to a large cloud-hosted model, organisations are increasingly combining on-device, edge and cloud-based AI models in cascaded architectures. This article examines the drivers behind model cascading, its role in enabling hybrid AI and the ecosystem challenges that must be overcome for it to scale.

Gabija Cepurnaite, Senior Consultant

May 2026

As enterprises scale AI, a new architectural pattern is emerging

The first wave of enterprise generative AI focused on accessing increasingly powerful large language models (LLMs) in the cloud. However, as organisations move from experimentation to operational deployment, many are discovering that relying on a single monolithic model is often too expensive, too slow and too inefficient for real-world environments. Instead, a new architectural approach is beginning to emerge: model cascading.

Model cascading involves orchestrating multiple AI models in sequence, with different models performing different tasks depending on complexity, latency requirements, resource availability or contextual triggers. Rather than sending every query or workload to a large and expensive model, lightweight models handle simpler tasks first and escalate only when deeper reasoning or richer outputs are required. This approach is increasingly becoming a foundational component of hybrid AI architectures: distributed AI systems spanning devices, edge infrastructure and cloud environments.

What is model cascading?

Model cascading is an architectural approach in which multiple AI models operate together in sequence, with each model handling a different level of complexity. Instead of routing every request to a single large model, the system first relies on smaller, lighter-weight models to process data locally: on-device or at the edge. Only when predefined trigger conditions are met – such as low confidence scores, ambiguous outputs or the need for deeper contextual understanding – is the task escalated to a more advanced and computationally intensive model.

In practice, this creates a hierarchy of inference. A lightweight model on a device may filter incoming sensor or video data, an edge-based model may perform richer inference, and a large cloud-hosted model may only be invoked for the most demanding reasoning tasks. The result is a distributed AI workflow that balances performance, latency and cost far more efficiently than relying on a single model architecture. Importantly, these cascades do not need to operate in one location. They can span devices, enterprise edge environments, telecom edge infrastructure and public cloud platforms. This is why model cascading is becoming closely associated with the broader emergence of hybrid AI.

Why model cascading matters now

Interest in model cascading is accelerating because enterprises are beginning to encounter the operational realities of scaling AI. During the initial generative AI boom, most organisations focused primarily on accessing increasingly powerful cloud-based models. However, as AI applications move into production environments, businesses are discovering that running every inference request through a large model is often prohibitively expensive and operationally inefficient. Inference, rather than training, is increasingly becoming the dominant long-term AI workload.

This is particularly important because many enterprise AI applications generate large volumes of inference requests. Customer service systems, industrial monitoring platforms, logistics environments and retail analytics applications may need to process thousands or even millions of interactions continuously. Using large models for every interaction quickly drives up GPU utilisation, cloud expenditure and energy consumption. Model cascading offers an alternative approach. Smaller models can resolve routine or low-complexity tasks locally, while only a minority of requests are escalated to larger models. In many scenarios, this allows organisations to preserve performance while reducing compute requirements. A recent research paper from Nvidia corroborates this view, presenting a view that smaller models are “sufficiently powerful, inherently more suitable, and necessarily more economical [...] and are therefore the future of agentic AI”.

At the same time, enterprise AI deployments are increasingly moving into environments where low latency and localised data processing matter. Industrial automation systems, smart city infrastructure, robotics platforms and video analytics applications often cannot tolerate the delays associated with sending all data to distant cloud environments. Many also involve sensitive operational data that enterprises prefer to keep locally. This is creating strong momentum behind edge AI deployments. In turn, it is increasing the importance of model cascading, since cascading architectures naturally align with distributed infrastructure. Initial inference can occur on-device, more advanced processing can take place at the edge, and only selected workloads need to be escalated to the cloud.

The result is a broader industry shift towards hybrid AI architectures in which inference workloads are dynamically distributed across devices, edge environments and cloud infrastructure depending on performance, cost and governance requirements.

The challenges of model cascading

Despite its potential, model cascading introduces a new layer of complexity that organisations must manage carefully.

1. **Orchestration.** A cascading system requires mechanisms to determine when a workload should be escalated, which model should be selected and where that model should run. These decisions must often be made dynamically and in real time. As a result, orchestration may become one of the most strategically important layers of the AI stack.
2. **Observability and governance.** In a traditional cloud AI deployment, organisations may only need to monitor a single model. In a cascading architecture, they may be managing several models distributed across devices, edge environments and cloud platforms. Ensuring consistent performance, reliability and regulatory compliance becomes significantly more difficult.
3. **Interoperability.** Many organisations will deploy models from multiple vendors, using different frameworks, optimisation techniques and hardware architectures. Creating seamless workflows between these models remains an immature area of the market.
4. **Optimisation complexity.** The economic value of model cascading depends on selecting the correct escalation thresholds. If lightweight models escalate too frequently, cost savings disappear. If they escalate too infrequently, accuracy may suffer. Determining the optimal balance between cost, latency and performance is likely to become a key area of innovation.

For this reason, the long-term winners in the model cascading ecosystem may not simply be model providers, but also the companies that can provide orchestration, monitoring and optimisation platforms capable of managing increasingly complex distributed AI environments.

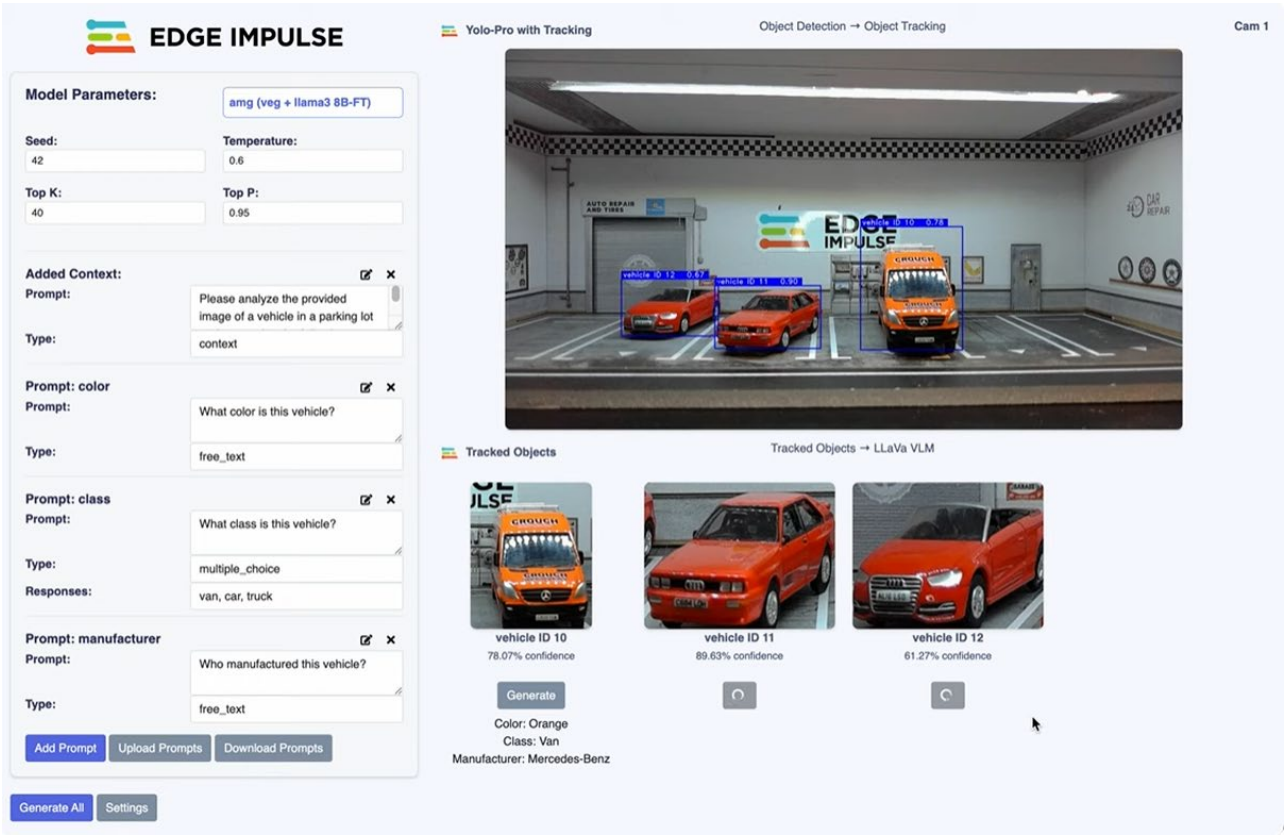
Example 1: Edge Impulse and cascaded edge AI

An example of model cascading in practice can be seen in Edge Impulse's edge AI architecture. The company has demonstrated a parking lot monitoring system that uses multiple models operating sequentially to show how more advanced AI capabilities can be deployed at the edge without relying on continuous cloud processing.

- **Stage 1: Lightweight object detection.** A camera captures images from a parking lot environment and passes them through an edge AI pipeline running on Qualcomm Dragonwing hardware. The first stage uses a lightweight object detection model to identify vehicles in the video feed. This model runs continuously and performs the relatively simple task of detecting whether a vehicle is present.

- **Stage 2: Triggered advanced inference.** The second stage is only triggered when the system identifies a new vehicle. At this point, the detected vehicle is cropped from the original image and passed to a more computationally intensive on-device vision-language model. By narrowing the input to only the relevant part of the image, the system improves the reliability of the smaller visual language model (VLM) and reduces unnecessary processing. The VLM then generates structured metadata about the vehicle, such as its type, manufacturer and colour.
- **Stage 3: Escalation and orchestration.** The system uses object tracking to follow vehicles within the frame, ensuring that the higher-powered model is not repeatedly invoked for the same vehicle. In a production environment, the structured outputs from the VLM could then be integrated into a wider parking lot management system, supporting use cases such as vehicle tracking, occupancy monitoring or automated metadata collection.

Figure 1: Edge Impulse parking lot monitoring demonstration



Source: Edge Impulse IQ9 Demo

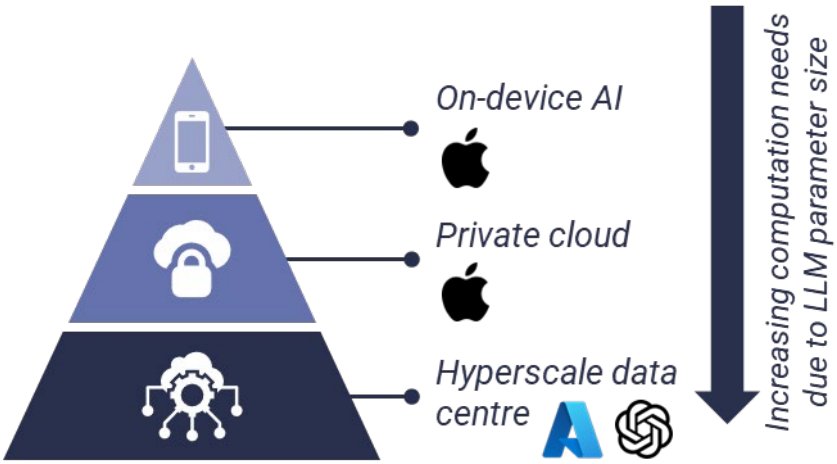
This architecture significantly reduces latency, bandwidth consumption, GPU usage and overall energy consumption. Importantly, it demonstrates how cascading architectures allow enterprises to balance resource constraints with sophisticated AI capabilities.

Example 2: Apple Intelligence's infrastructure hierarchy

Another illustrative example of model cascading is Apple's approach to its Apple Intelligence portfolio. In this case, some generative AI functions run on-device, typically on an iPhone, while others are escalated to Apple's private cloud infrastructure or even to Azure cloud infrastructure, where ChatGPT is used.

For example, Apple's AI writing tools use on-device models for proofreading and rewriting, escalate summarisation tasks to Apple's private cloud, and rely on ChatGPT for composing entirely new text. This provides a useful example of how generative AI can be deployed in a hybrid edge architecture, with tasks escalated according to the size and capability of the LLM required.

The infrastructure hierarchy of Apple Intelligence



Source: STL Partners

The future of model cascading depends on orchestration

Model cascading is emerging as a practical response to the challenges of scaling AI beyond experimentation and into production. As inference workloads grow, enterprises are increasingly looking for ways to balance performance, latency, cost and governance requirements. However, widespread adoption of model cascading will depend on the maturation of the supporting ecosystem. Organisations will need more sophisticated orchestration platforms capable of managing multiple models across distributed environments, as well as improved observability, governance and optimisation tools. Standards and interoperability will also become increasingly important as enterprises combine models, hardware and software from multiple vendors.

As AI deployments become more distributed and inference becomes the dominant operational workload, model cascading is likely to evolve from a niche architectural technique into a core design principle for enterprise AI. The organisations that can simplify the complexity of deploying, managing and optimising cascaded AI systems will be well positioned to capture value as the next generation of hybrid AI architectures emerges.

Gabija is a Senior Consultant at STL Partners, specialising in edge computing and AI-RAN.

Get in touch with the author to learn more
gabija.cepurnaite@stlpartners.com

Or visit STL Partners' Edge Hub
www.stlpartners.com/edge-computing