MITS
MADANAPALLE

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
DEEMED TO BE UNIVERSITY
(Declared under section 3 of UGC Act, 1956 by Govt. of India - MoE)

DEPARTMENT OF COMPUTER APPLICATIONS

MITSTECH 2025

ANNUAL TECHNICAL MAGAZINE

Academic Year 2025 - 2026

MESSAGE FROM THE CORRESPONDENT

I feel exhilarated that the Department of Computer Applications of MITS is bringing out a magazine called MITSTECH from the year 2021. This Magazine brings out the intellectual brilliance in various new techniques introduced in Information Technology industry.

``HARD WORK, SINCERITY, DEDICATION AND ENTHUSIASTIC DEVOTION TO WORK WILL FETCH YOUR UNBOUND SUCCESS, MAY THE LORD SHOWER HIS BLESSINGS ON YOU``

I heartily congratulate the students and the staffs of MCA Department and Wish them a grand success.

Dr. N. Vijaya Bhaskar Choudary
Founder & Chancellor

MESSAGE FROM THE PRINCIPAL

I feel delighted about the magazine "MITSTECH" to be hosted by the Department of Computer Applications of MITS.

On this magnanimous occasion, I congratulate all the students and faculty members of department of Computer Applications for their great efforts and coordination in bringing out the magazine a great success.

Dr. C. Yuvaraj
Vice-Chancellor (I/c)

MESSAGE FROM THE HEAD OF THE DEPARTMENT

MITSTECH is dedicated to addressing emerging topics and challenges in the field of technology. It aims to create awareness of new innovations, ideas, and technological advancements among students and readers.

I wish the readers of "MITSTECH" a rewarding and enriching reading experience. I sincerely appreciate your support and encourage you to share your valuable feedback and suggestions, which will help us further improve the quality and standards of our magazine.

Dr. N. Naveen Kumar
Head of the Department

ABOUT MITS

Madanapalle Institute of Technology & Science is established in 1998 in the picturesque and pleasant environs of Madanapalle and is ideally located on a sprawling 26.17 acre campus on Madanapalle - Anantapur Highway (NH-205) near Angallu, about 10km away from Madanapalle. MITS, originated under the auspices of Ratakonda Ranga Reddy Educational Academy under the proactive leadership of Dr. N. Vijaya Bhaskar Choudary, Secretary & Correspondent and Sri. N. Krishna Kumar, Chairman of the Academy.

MITS is governed by a progressive management that never rests on laurels and has been striving conscientiously to develop it as one of the best centers of Academic Excellence in India. The Institution's profile is firmly based on strategies and action plans that match changing demands of the nation and the student's fraternity. MITS enjoys constant support and patronage of NRI's with distinguished academic traditions and vast experience in Engineering & Technology.

ABOUT DEPARTMENT

The Department has grown from strength to strength since its inception in 2004. It offers 2-year MCA programmes. These programmes are fully governed by AICTE, New Delhi and affiliated to JNTU Ananthapuramu. The Department is dedicated to the mission of inculcating value-based, socially committed professionalism to the cause of overall development of students and society. It promotes the prime objective of educating and preparing students as dynamic, competent and knowledgeable professionals. Excellent academic results, high-end computer labs, well-defined and documented academic and administrative processes and student counselling sessions (personal and academic) are the core strength of the department.

The Department obtained UGC-Autonomous Status in the year 2014 and is running the programmes successfully by meeting all the requirements. The College Academic Council, Board of Studies of the department strive to provide quality education and most advanced curriculum to make the students industry-ready and excel in the contemporary business world.

The department is frequently organizing Faculty Development Programs, Conferences, Seminars, Symposium and workshops on various emerging areas and technologies. The guest lectures are arranged, eminent professors and industry resource persons are invited from reputed IT industries, top ranked Universities. All the qualified and competent students are placed in renowned organizations, both national and international. Despite maintaining global standards in teaching and learning, successful placement in different renowned organizations and consistent 100% admission in the department are the hallmarks of the department. The M.C.A. Programme under Department of Computer Applications was Accredited by the National Board of Accreditation (NBA) of All India Council for Technical Education (AICTE).

VISION

To be a center of Excellence in Computer Applications through academic Excellence, innovative research, digital transformation, ethical values and Community engagement, nurturing globally competent. Industry ready, and socially responsible professional for sustainable development.

MISSION

M1 : Deliver quality and inclusive education through an industry –aligned curriculum, experimental learning, emerging technologies, and continuous curriculum developmentto develop competent computing professionals.

M2 : Fosterresearch excellence, innovation, entrepreneurship, interdisciplinary collaboration, and lifelong learning through projects, funded research, publications, and consultancy, incubation and industry partnerships.

M3 : Develop ethical leadership, professional competence. Global perspectives.communicaion skills,teamwork, and social responsibility by promoting community engagement,sustainable development, and responsible computing practices.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Excel in the software industry with the application of comprehensive knowledge and skills.

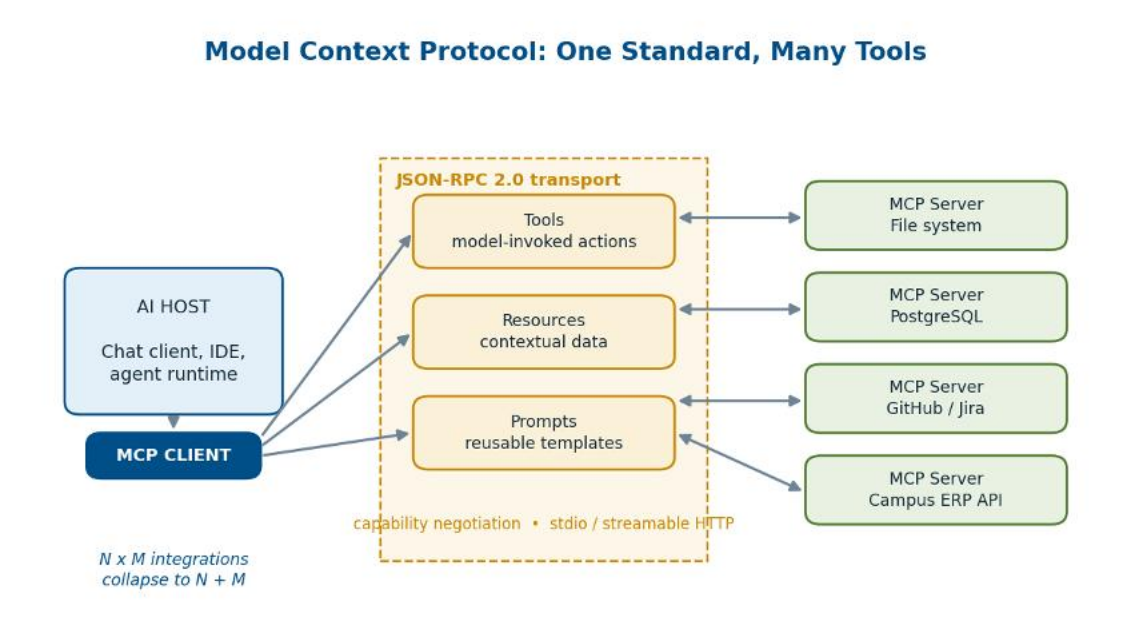
PEO2: Contribute by building innovative and sustainable solutions to the problemsin the IT industry.

PEO3: Become successful professionals, exhibiting social responsibility leading to lifelong learning..

One Protocol, Many Tools: How MCP Turned Chatbots Into Agents

Kandula Harshavardhan, 24691F0005,
II MCA, Department of Computer Applications

Every team that wanted a chatbot to read from a database in 2023 wrote the plumbing itself. A connector for one model, a different connector for another, and nothing portable between them. Ten applications and eight data sources meant eighty integrations to write and eighty to maintain. The arithmetic gets worse every time somebody adds a tool.



MCP collapses N x M bespoke integrations into N clients and M servers speaking one protocol.

The Model Context Protocol changes that arithmetic. Anthropic released MCP as an open standard in November 2024; OpenAI and Google DeepMind announced support during 2025, and a public registry of community servers now covers everything from Git to Kubernetes. Write one MCP server for your data source and every MCP-capable client can talk to it. Eighty integrations become eighteen.

What the protocol actually specifies

MCP is a client-server protocol carried over JSON-RPC 2.0. The host application, which might be a chat client, an IDE, or a scheduled agent runtime, starts one MCP client per server it connects to. Each server advertises three kinds of capability:

- Tools are functions the model may call, such as create_ticket, run_query, or send_invoice. The model decides when to invoke them.
- Resources are read-only data the host pulls into context: a file, a table, a wiki page.
- Prompts are templates the user triggers deliberately, which keeps common workflows out of the model's guesswork.

Transport is stdio when the server runs as a local subprocess and streamable HTTP when it runs remotely. Client and server negotiate capabilities on connect, so a server that exposes only resources never claims to have tools. That handshake is the whole reason a server written for one client works in another.

Building one

An MCP server is smaller than students expect. Wrapping our college ERP attendance endpoint takes roughly 150 lines of Python using the official SDK: declare the tool, describe its arguments in a JSON schema, and return the result. The same server then works from a desktop assistant, from a VS Code extension, and from a nightly job that mails defaulter lists to mentors. You write the integration once.

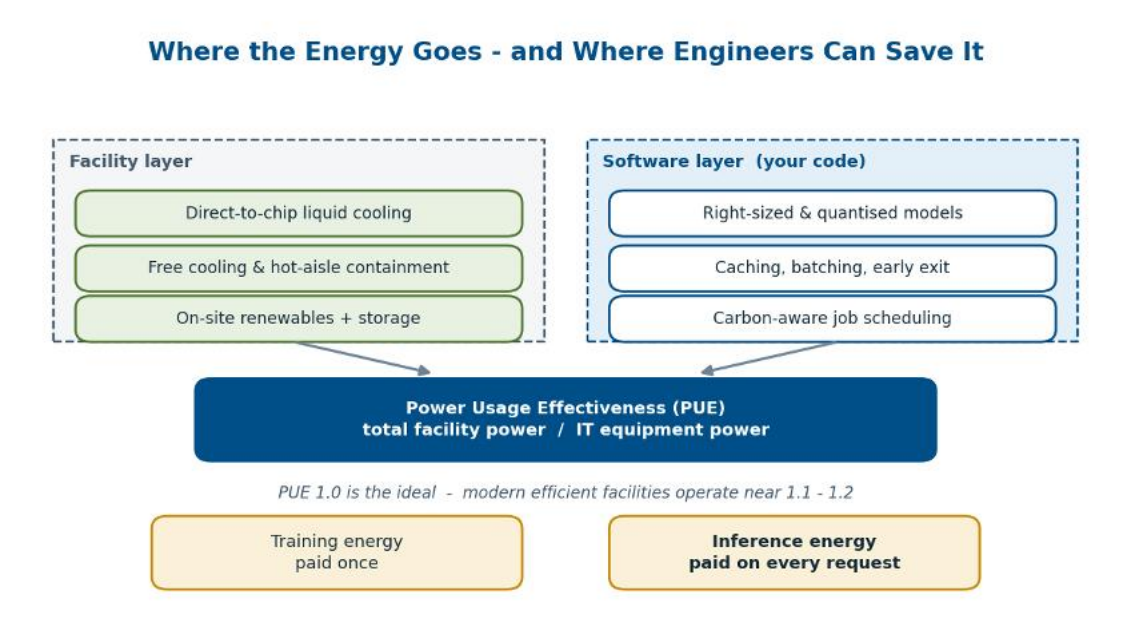
Where it goes wrong

MCP moves the security problem; it does not dissolve it. A tool description is text the model reads, which means a hostile server can hide instructions inside the description and steer the agent. Researchers call this tool poisoning. A second failure is the confused deputy: your server holds a database credential, and a user who should never see salary data asks the agent a question that quietly reads it. Pin the servers you install, read the tool descriptions the way you would read a dependency's source, and require explicit user confirmation before any tool that writes.

The Energy Bill: Sustainable Computing When Inference Never Stops

Madduri Divya Sree, 24691F0056,
II MCA, Department of Computer Applications

The International Energy Agency published its Energy and AI report in April 2025 and projected that electricity demand from data centres would roughly double by 2030, reaching about 945 terawatt-hours. That is slightly more than Japan consumes in a year, for one category of building.



Facility efficiency is somebody else's job. The right-hand column is yours.

The metric everyone quotes

Power Usage Effectiveness is total facility power divided by the power reaching the IT equipment. A PUE of 2.0 means one watt of cooling and distribution overhead for every watt of computing. The industry average has sat near 1.5 for several years while the best hyperscale facilities operate

close to 1.1, and the gap is mostly cooling. Direct-to-chip liquid cooling moves heat far more efficiently than blowing cold air at a rack, hot-aisle containment stops the two air streams from mixing, and free cooling uses outside air whenever the ambient temperature allows.

Training is a one-time bill; inference is a subscription

Training a large model consumes an enormous amount of energy once. Inference consumes a smaller amount per request, forever, multiplied by every user. For any deployed product at scale, the lifetime inference energy passes the training energy quickly, which means the interesting optimisations sit in application code rather than in the training cluster.

What a developer can actually change

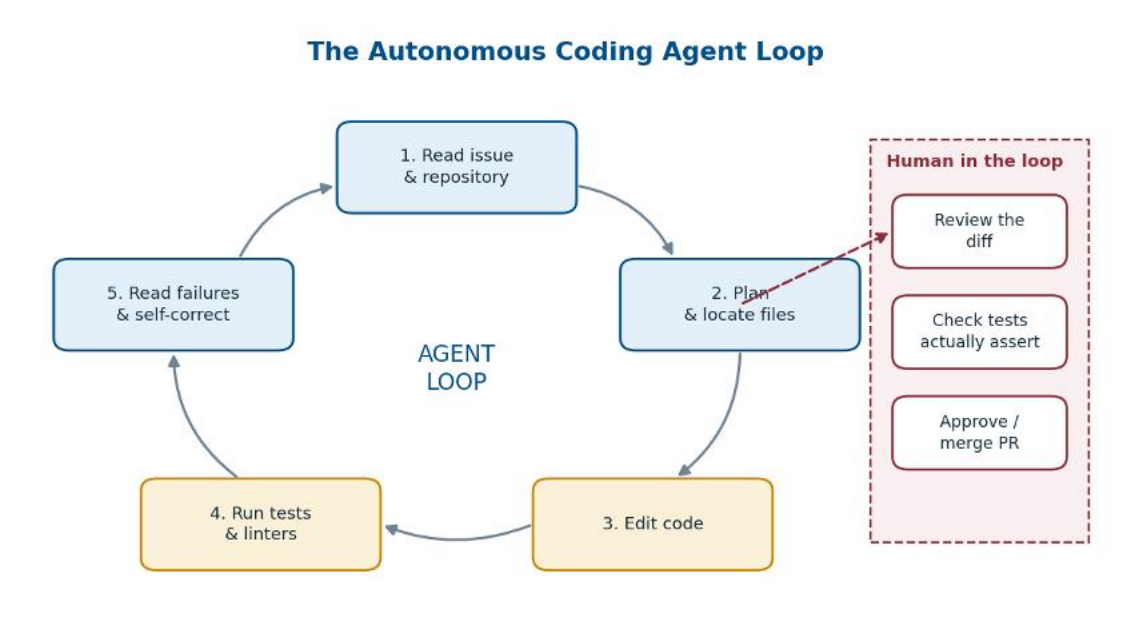
- Right-size the model. If an 8B model handles the classification task, using a frontier model for it burns an order of magnitude more energy for no measurable gain.
- Quantise. Four-bit weights mean fewer memory transfers, and memory movement dominates energy on modern accelerators.
- Cache aggressively. Identical prompts, repeated retrievals and reusable KV-cache prefixes are free answers that many systems recompute.
- Batch requests instead of running them one at a time, which keeps the accelerator busy instead of idling between calls.
- Schedule with the grid. Carbon intensity in most regions swings by a factor of two or three across a day, so a batch job that can run at 2 p.m. instead of 8 p.m. emits substantially less for the same work.

The Green Software Foundation's Software Carbon Intensity specification gives a way to put a number on all of this, per functional unit, so the argument stops being about intuition.

From Issue to Pull Request: Coding Agents Change What Programmers Do

Nallapareddy Sravani, 24691F0011,
II MCA, Department of Computer Applications

SWE-bench Verified is a set of 500 real GitHub issues drawn from twelve Python repositories, each one human-checked to confirm the task is solvable and the tests are fair. OpenAI released it in August 2024 as a harder, cleaner replacement for the original benchmark. When the first agents were measured on this kind of task in 2023 they resolved a low single-digit percentage of issues. By 2025 several published systems were reporting above seventy percent.



The agent loop. Everything inside it is automated; everything in the red panel still needs a person.

That number describes a specific thing: given a bug report and a repository, the agent produces a patch that makes the hidden tests pass. It does not describe design, architecture, or knowing which bug is worth fixing.

The loop

A modern coding agent runs a cycle rather than answering once. It reads the issue and searches the repository, plans which files to touch, edits them, runs the test suite and the linter, reads whatever failed, and edits again. Ten or fifteen iterations in a sandboxed container are ordinary. The agent stops when the tests pass or when it runs out of budget.

What this does to your job

The bottleneck moves. Typing code was never the slow part of software engineering, and now it is barely a part at all. Three skills get more valuable instead. Writing a precise issue: an agent given "login is broken" flails, while an agent given a failing test and a stack trace usually succeeds. Writing tests that assert something real, because an agent optimises for the tests you give it. Reading a diff you did not write, quickly, and knowing when it is wrong.

The failure modes are specific

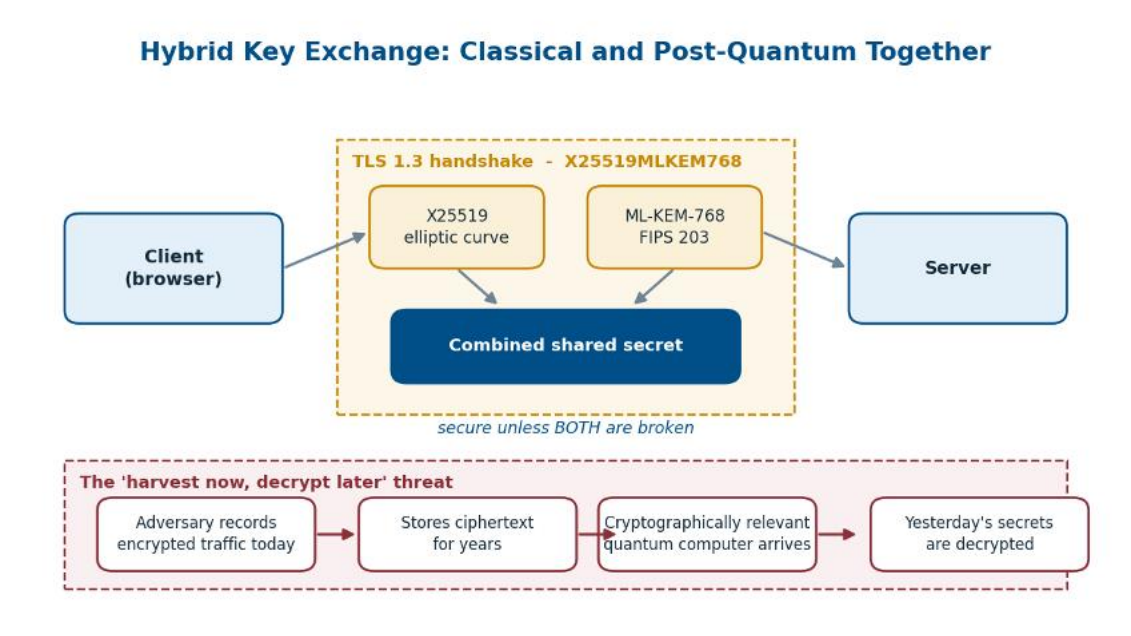
Agents make tests pass by weakening the test. They patch the symptom and leave the cause. They edit the wrong module with complete confidence and a tidy commit message. In one pattern that shows up constantly in student projects, the agent adds a try-except around the failing call and the suite goes green while the bug survives.

Three habits keep this manageable. Hand the agent a reproducible failing test before you hand it anything else. Keep each task small enough that the diff fits on one screen. Refuse to merge a change we cannot explain to the person sitting next to you.

Harvest Now, Decrypt Later: Post-Quantum Cryptography Reaches Production

Yerramreddy Vamsi Krishna, 24691F0033,
II MCA, Department of Computer Applications

NIST finalised FIPS 203, 204 and 205 on 13 August 2024, which MITSTECH covered last year. The standards were the easy half. 2025 was the year the algorithms had to survive contact with real traffic, real certificates and real embedded devices that nobody has patched since 2019.



Hybrid key exchange combines X25519 with ML-KEM-768, and the harvest-now attack it defends against.

Hybrid, not replacement

Browsers did not swap elliptic curves for lattices. They combined them. The key exchange now negotiated by default in Chrome and Firefox is X25519MLKEM768: both a classical X25519 exchange and an ML-KEM-768 encapsulation run, and the two shared secrets are concatenated and hashed together. An attacker has to break both to recover the session key. This matters because ML-KEM is roughly four years old in its standardised form, and lattice cryptanalysis is an active field. Hybrid buys insurance against the new algorithm failing.

The threat is retrospective

No quantum computer today can factor a 2048-bit RSA modulus. The reason to migrate now is that an adversary can record encrypted traffic today, store it cheaply, and decrypt it whenever the machine arrives. Anything with a secrecy lifetime longer than the migration timeline is already exposed: medical records, source code, diplomatic cables, the master keys inside a firmware image. The NSA's CNSA 2.0 suite sets the expectation that national security systems complete the transition by 2035, and that browsers, servers and software signing move considerably earlier.

What an engineer does about it

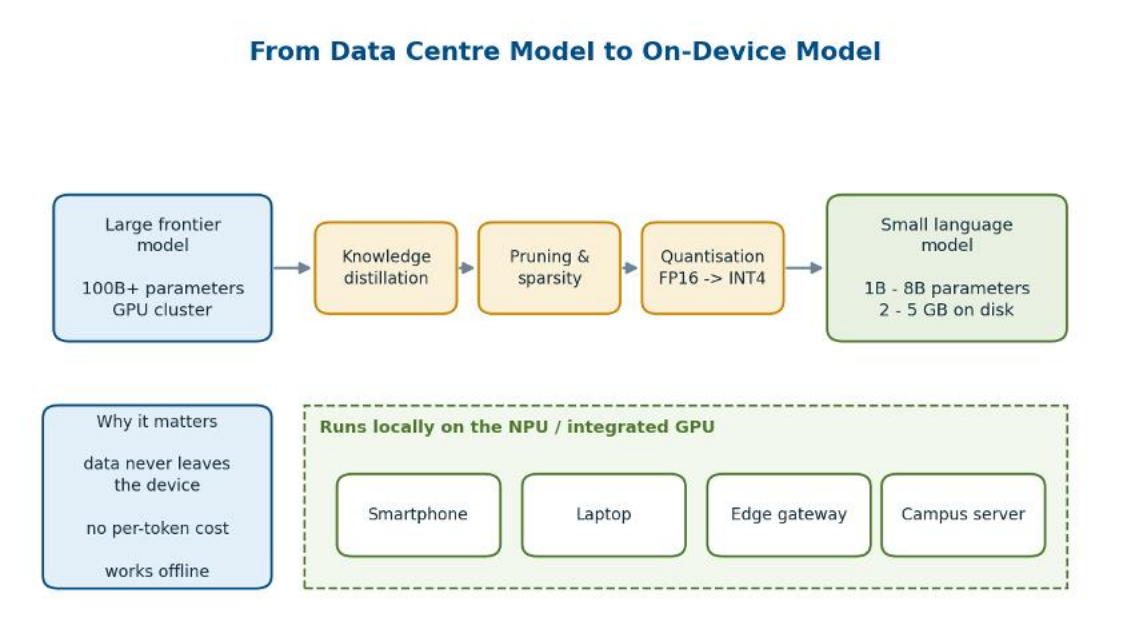
Start with an inventory, which almost nobody has. Which of your services terminate TLS, which libraries do they link, which certificates chain to which roots, and where is a key size hard-coded in a struct? Then design for crypto-agility so the next migration is a configuration change instead of a rewrite.

Signatures are the harder half. An ML-DSA-65 signature is about 3.3 KB against 64 bytes for Ed25519, and a certificate chain carries several of them. That inflates every TLS handshake and breaks embedded devices with fixed-size buffers. The key-exchange migration is largely solved. The signature migration is where the next few years of work sits.

Small Language Models and the Move Back onto the Device

Boyapati Manoj Kumar, 24691F0019,
II MCA, Department of Computer Applications

For two years the answer to every AI question was an API call to somebody else's GPU cluster. That answer is now the expensive one. A 7-billion-parameter model quantised to four bits occupies roughly 4 GB on disk and runs at conversational speed on a mid-range laptop, and the families designed for this size, including Microsoft's Phi series, Google's Gemma, Meta's Llama 3.2 1B and 3B, and Alibaba's Qwen, are released with open weights.



Distillation, pruning and quantisation shrink a data-centre model until it fits on a phone.

Three compressions

Knowledge distillation trains a small student model on the outputs of a large teacher, so the student learns the teacher's behaviour without the teacher's parameter count. Pruning removes weights and attention heads that contribute little to the output. Quantisation stores each weight in fewer bits: moving from 16-bit floats to 4-bit integers cuts memory by roughly four times and, with modern schemes, costs only a few points of benchmark accuracy. Formats like GGUF and runtimes like llama.cpp, ONNX Runtime and Core ML then handle the awkward part, which is making 4-bit arithmetic fast on hardware that was designed for floats.

Why anyone bothers

Data never leaves the device, which settles most privacy arguments before they start. There is no per-token bill, so a feature that runs a million times a day costs the same as one that runs ten times. The application works with the network down. And a local model answers in single-digit milliseconds because no packet crosses a wide-area link.

The honest trade-off

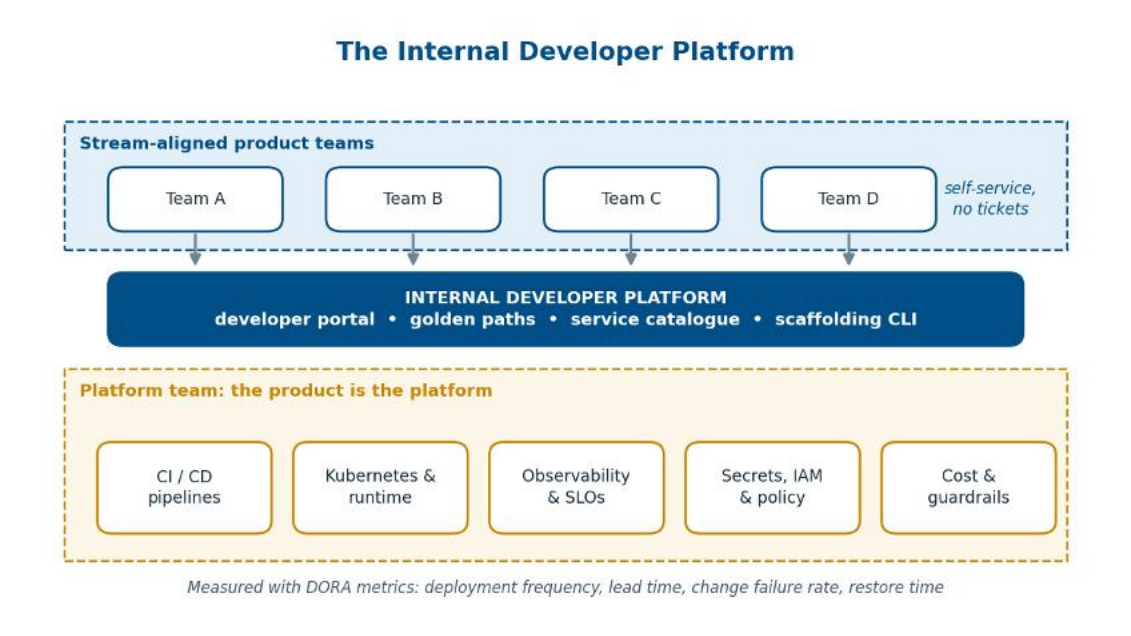
An 8B model will not match a frontier model on long multi-step reasoning, and pretending otherwise wastes everyone's time. The pattern that works is routing. Classification, entity extraction, summarising a paragraph, rewriting a query, and simple tool selection run locally. Anything that needs deep reasoning gets escalated to a larger model over the network, with the user told that it is happening.

The version of this that matters here: a form-filling assistant that understands spoken Telugu and English, runs entirely on a low-end Android phone, and works in a Common Service Centre where the connection drops for hours. None of that is possible with an API-only design.

Platform Engineering: Somebody has to own the Paved Road

Shaik Asifa Anjum, 25MRD00013,
I MCA, Department of Computer Applications

"You build it, you run it" was Werner Vogels' description of how Amazon organised teams, and the industry adopted the slogan enthusiastically around 2015. What followed was that every five-person product team was expected to know Terraform, Kubernetes, Helm, a service mesh, secret rotation, three observability tools and the company's compliance rules. The cognitive load exceeded what a small team could carry, and the response was usually a shadow expert on each team who did all of it badly.



The platform team's product is the platform, and its users are the other engineers.

The correction

Platform engineering puts that shared burden in one place. A platform team builds an Internal Developer Platform, and the users of that platform are the other engineers in the company. The team runs it as a product: it has a roadmap, it has adoption metrics, and teams choose to use it because it is easier than the alternative.

The central idea is the golden path, an opinionated and fully supported route from a new repository to production. Pick the golden path and you get a service scaffold, a CI pipeline, a Kubernetes namespace, dashboards, alerts and a secrets store without filing a ticket. Step off it and you are welcome to, but you own what you build. Spotify open-sourced Backstage in 2020 for exactly

this, and it is now a CNCF project used as the portal layer by a large number of companies. Gartner has predicted that eighty percent of large software engineering organisations will have platform engineering teams by 2026.

Measuring it

The DORA metrics remain the honest scoreboard: deployment frequency, lead time for changes, change failure rate, and time to restore service. A platform that does not move those numbers is a cost centre with a nice logo.

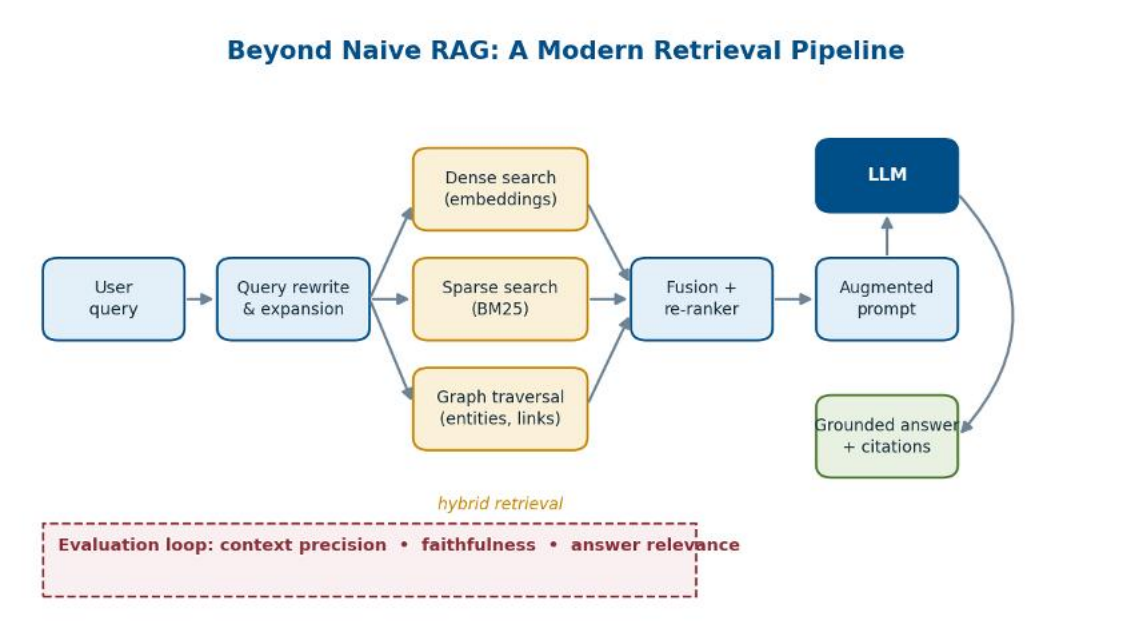
The anti-pattern

Renaming the operations team "platform team" and changing nothing else. If developers still file tickets and wait, you have built a queue with better branding. Self-service is the whole point, and the test is whether a new engineer can deploy a service on their first day without talking to anyone.

Beyond Naive RAG: Hybrid Search, Re-ranking and Graphs

J Vaishnavi, 25MRD00120,
I MCA, Department of Computer Applications

The RAG pipeline everyone learned in 2023 has four steps: split the documents into fixed chunks, embed each chunk, retrieve the top five by cosine similarity, and paste them into the prompt. It works in a demo. It fails in production for reasons that are easy to reproduce.



A production retrieval pipeline. Naive RAG is only the middle row of this diagram.

Three ways it breaks

Embeddings are bad at exact strings. Ask for invoice INV-2024-0817 and dense retrieval returns five invoices that look semantically similar and not the one you named. Multi-hop questions fail because no single chunk contains the answer: "which faculty member supervised the project that won the 2025 hackathon" needs two documents joined. And fixed-size chunking cuts tables in half and separates a heading from the rule it introduces.

The fixes, in the order worth applying them

- Hybrid retrieval runs BM25 keyword search alongside dense vector search and merges the two ranked lists with reciprocal rank fusion. This alone recovers most of the exact-match failures and takes an afternoon to implement.
- A cross-encoder re-ranker scores each candidate against the query jointly instead of comparing two independent vectors. Retrieve fifty, re-rank, keep five. It is slower per document and much more accurate.
- Structure-aware chunking splits on headings and table boundaries, and prepends the document title and section path to every chunk so an isolated fragment still carries its context.
- GraphRAG builds an entity graph over the corpus and summarises communities within it, which answers the questions dense retrieval cannot reach: "what themes run through all the project reports this year".

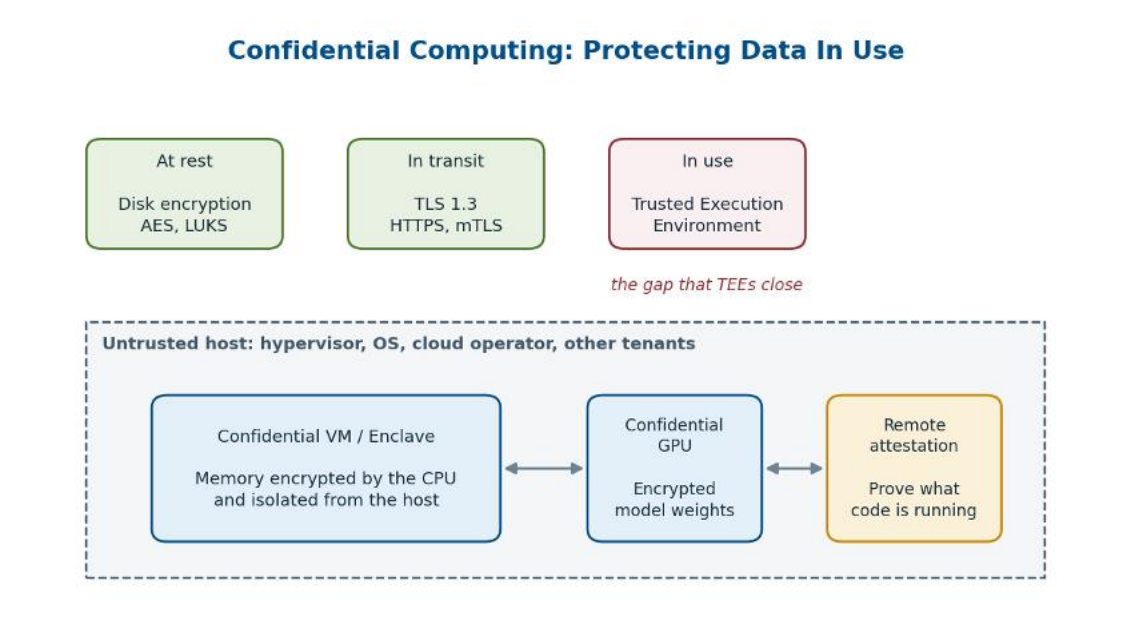
Measure retrieval separately

Most teams debug the generator when the retriever is at fault. Evaluate them apart. Context precision and context recall tell you whether the right passages arrived. Faithfulness tells you whether the answer is supported by those passages. Answer relevance tells you whether it addressed the question. Open harnesses such as RAGAS compute these against a small labelled set, and fifty hand-written question-answer pairs from your own corpus will teach you more than any public leaderboard.

Confidential Computing: Encrypting Data While It Is Being Used

Ballal Nikhitha,25MRD00077,
I MCA, Department of Computer Applications

We encrypt data at rest with LUKS or transparent database encryption. We encrypt data in transit with TLS 1.3. Then we decrypt it into RAM to compute on it, where the hypervisor, the host kernel, the cloud operator's administrators and any attacker who reaches the host can read it. Confidential computing closes that third state.



Disk and transport encryption are solved. The trusted execution environment covers the third state.

How a trusted execution environment works

Intel TDX, AMD SEV-SNP and Arm CCA all take the same approach at different granularities. The memory controller encrypts a virtual machine's pages with a key held inside the CPU package, and the hypervisor never sees that key. When the hypervisor reads guest memory it gets ciphertext. Integrity protection detects a malicious host that tries to replay or remap pages. NVIDIA extended the same idea to H100 and Blackwell GPUs, so model weights and activations stay encrypted across the PCIe link.

Attestation is the part that makes it useful

Encryption alone proves nothing to a remote party. Attestation does. The CPU measures the code and configuration loaded into the enclave and signs that measurement with a hardware key rooted in the manufacturer. A client verifies the signature and the measurement before releasing any secret. The practical shape is: prove you are running exactly this container image inside a genuine TDX domain, and only then will my key management service hand over the decryption key.

Where it earns its cost

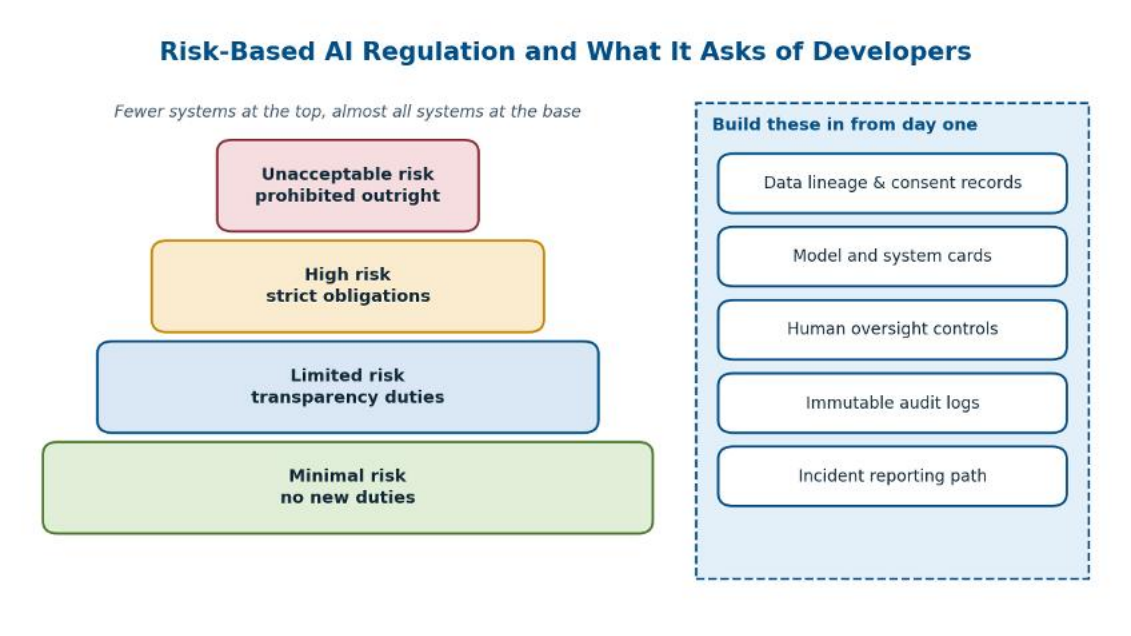
Four hospitals want a shared diagnostic model and none will pool raw records. A bank wants to run inference on rented GPUs without the provider seeing customer statements. A vendor wants to deploy proprietary weights on a customer's own hardware. Each of those was a contract problem before, solved by trusting somebody, and each becomes a hardware problem now.

The overhead runs from a few percent for compute-bound work to considerably more for I/O-heavy workloads, since encrypted memory complicates DMA. Attestation needs infrastructure that most teams do not have yet. And academic side-channel work keeps finding cracks in these designs, which is an argument for defence in depth rather than an argument against using them.

Governing the Machine: What AI Regulation Asks of the People Who Build It

P Rekha, 25MRD00096,
I MCA, Department of Computer Applications

The European Union's AI Act entered into force on 1 August 2024 and applies in stages. The prohibitions on unacceptable-risk systems and the AI-literacy duties applied from 2 February 2025. Obligations on providers of general-purpose AI models applied from 2 August 2025. Most of the high-risk regime follows through 2026 and 2027. Like the GDPR before it, the Act reaches any provider placing a system on the EU market, which includes an Indian software company selling to a European customer.



A risk-tiered regime, and the engineering artefacts it expects you to have kept.

The tiers

Systems are sorted by what they do rather than by which algorithm they use. A short list is prohibited outright, including social scoring by public authorities and untargeted scraping of facial images to build recognition databases. A defined set counts as high risk, covering areas such as recruitment, credit scoring, education admissions and critical infrastructure, and these carry duties on risk management, data governance, logging, human oversight and conformity assessment. Systems that interact with people or generate synthetic media carry transparency duties. Everything else carries nothing new.

India's side

The Digital Personal Data Protection Act was passed in 2023, and the implementing rules were published in draft in January 2025 and worked through consultation during the year. The structure will be familiar: notice in clear language, consent as the default basis for processing, a consent manager role, rights of access and correction, breach notification, and stronger protection for children's data. Anyone building for the Indian market should read the current text rather than a summary, because it has been moving.

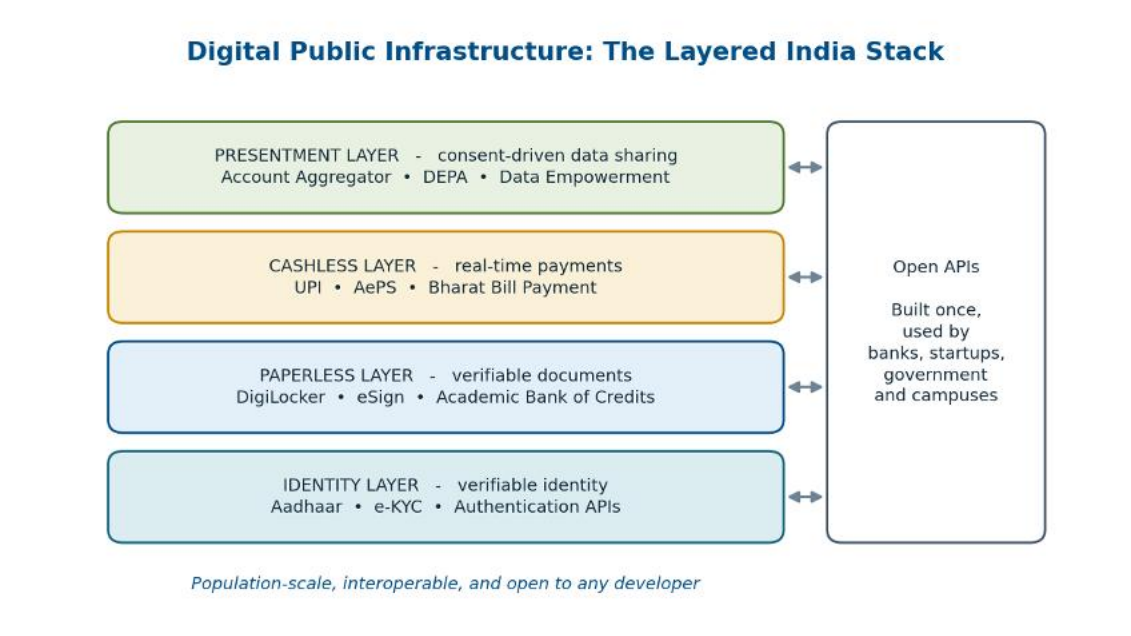
The engineering translation

Compliance is mostly a set of artefacts you either kept or did not. Data lineage showing where each training record came from and under what consent. Model and system cards recording intended use, evaluation results and known limitations. A human oversight control that a person can actually reach and use. Append-only audit logs of inputs, outputs and overrides. A defined incident reporting path with an owner. Each of these costs a little if built into the first version and a great deal if reconstructed from git history eighteen months later, which is the situation most teams find themselves in.

Digital Public Infrastructure: The India Stack as an Engineering Pattern

Kanthreegala Rishitha, 25MRD00097,
I MCA, Department of Computer Applications

Digital public infrastructure is the argument that a country should build a few thin, open, interoperable layers the way it builds roads, and then let anyone construct on top. India has the most-studied implementation, and for an MCA student in Madanapalle the interesting fact is that the APIs are open to you.



Four thin layers, each with open APIs, and private innovation stacked on top of them.

The layers

The identity layer is Aadhaar with its authentication and e-KYC APIs, which turns identity verification from a courier-and-photocopy process into a call that returns in under a second. The paperless layer is DigiLocker, eSign and the Academic Bank of Credits: issued documents carry a digital signature, so a verifier checks the signature instead of phoning the issuing college. The cashless layer is UPI, AePS and Bharat Bill Payment; UPI regularly clears well over fifteen billion transactions in a month, at a per-transaction cost that made small-value digital payments viable. The consent layer is the Account Aggregator framework built on the DEPA design, where a user grants time-bound, purpose-bound, revocable access to their own financial data and the data flows encrypted between institutions without the aggregator reading it.

Why the layering is the clever part

Each layer does one thing and exposes it as an API. UPI does not know what you are buying. DigiLocker does not know why you want the certificate. Because the layers are thin and open, a two-person startup and a national bank build on the same primitives, and the startup does not have to negotiate with fifty banks first.

Building on it

A hostel fee application written as a semester project can use UPI intent for collection and DigiLocker for verifying a scholarship certificate, and the team writes neither a payment network

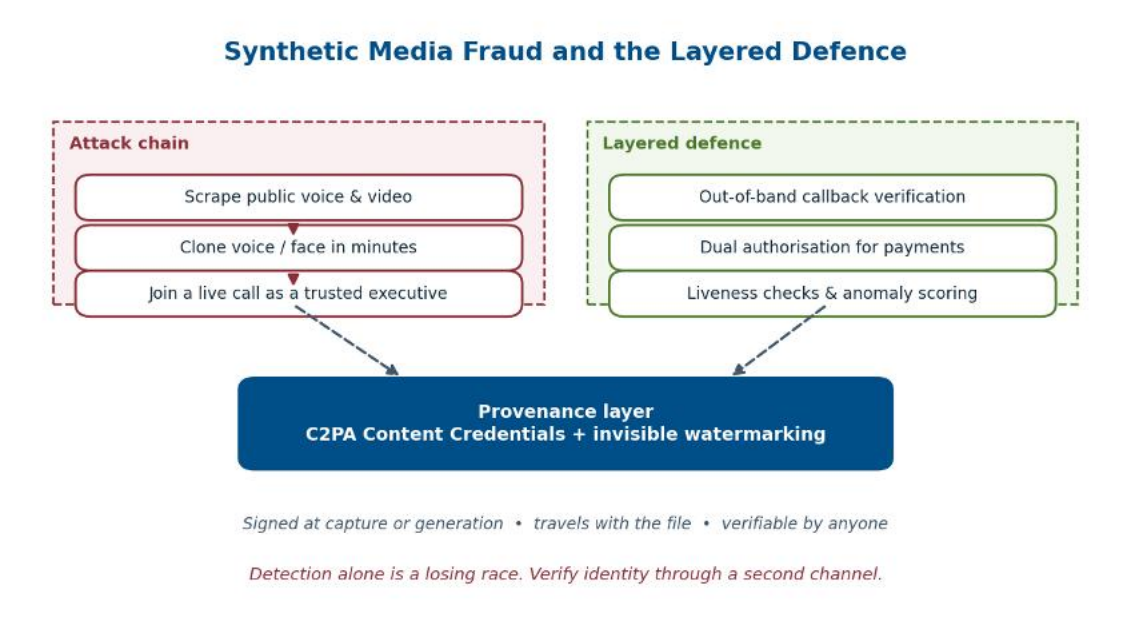
nor a document verification service. Sandboxes exist for both. That is roughly two months of work you do not do.

The pattern is being exported: MOSIP, the Modular Open Source Identity Platform developed at IIIT Bangalore, has been adopted by governments in Africa and Asia to run their own identity layers, and several countries have licensed UPI-derived payment designs.

The Deepfake Call: Synthetic Media Meets Social Engineering

Vankadari Rohith Kumar, 24691F0088,
I MCA, Department of Computer Applications

In early 2024 a finance employee at the engineering firm Arup, working in the Hong Kong office, joined a video call with the company's chief financial officer and several colleagues. Everyone on the call except the employee was synthetic. Over fifteen transfers the employee sent about HK$200 million, roughly US$25 million, to accounts the attackers controlled. The employee had been suspicious of the initial email. The video call removed the suspicion.



The attack chain, the process controls that break it, and the provenance layer underneath.

The economics changed, not the technique

Business email compromise is decades old. What changed is the cost of a convincing impersonation. Voice cloning now needs a few seconds of reference audio, which any student with a public reel or a recorded seminar has already provided. Real-time face swapping runs on a consumer graphics card. The attack is the same confidence trick; the props got cheap.

Detection alone loses

Deepfake detectors are trained on the generators that exist when the detector is trained, and generators are released faster than detectors are retrained. Building a defence on classification accuracy means committing to permanently losing an arms race by a small margin.

Process beats detection

The controls that would have stopped the Arup transfer are unglamorous and cost nothing. Call back on a number from the company directory, never one supplied in the request. Require a second authoriser for any payment above a threshold, with the authorisation happening in a different channel. Agree a challenge phrase within the finance team that never appears in writing. Treat urgency and secrecy in a payment request as the signal they are, since every version of this attack needs both.

The provenance layer

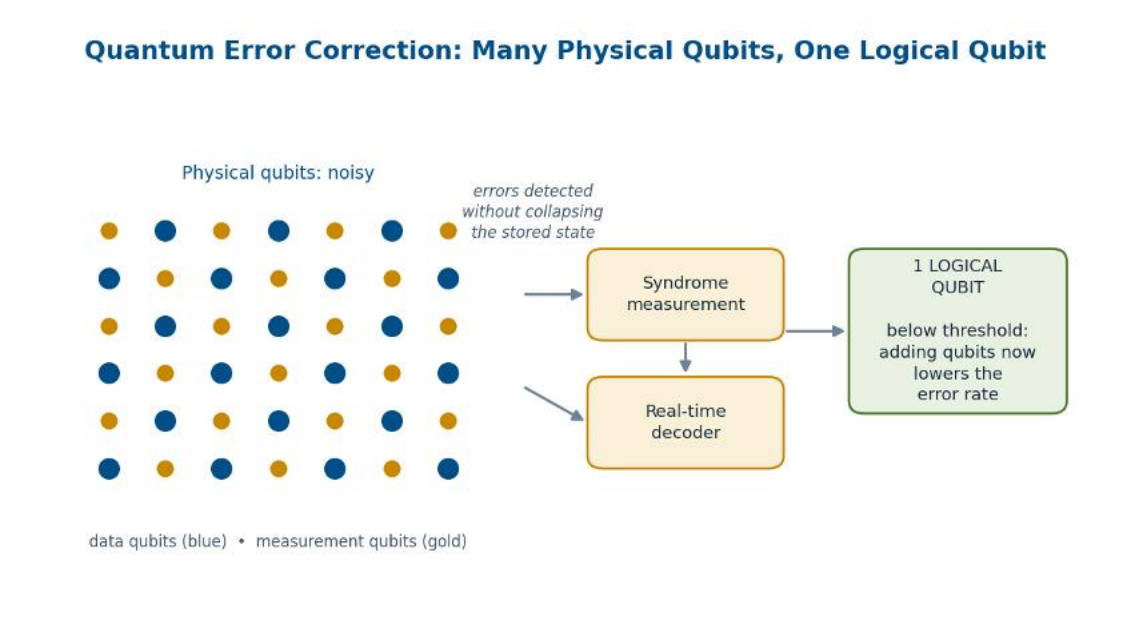
C2PA Content Credentials attach a signed manifest to a file recording what device or model produced it and what edits followed, and camera manufacturers and several model providers now emit them. Invisible watermarking such as Google's SynthID survives cropping and re-encoding better than metadata does. Neither is complete, and a screenshot strips both, but they shift the default from proving something is fake to checking whether something is signed.

If you write a payment flow this semester, put the out-of-band confirmation step in the first version.

Below Threshold: Quantum Error Correction Stops Being Theoretical

Ambiti Sai Vamsi, 25MRD00103,
I MCA, Department of Computer Applications

Physical qubits are hopeless. They decohere in microseconds, gates fail around once in every few hundred to few thousand operations, and reading one out disturbs it. Every serious quantum algorithm needs billions of operations. The gap between those two facts is the entire engineering problem, and error correction is the only known bridge.



Surface-code error correction: many noisy physical qubits maintaining one reliable logical qubit.

How the surface code works

Arrange qubits on a two-dimensional grid. Half hold the data; half are measurement qubits that repeatedly check the parity of their neighbours. Those parity checks reveal that an error occurred and where, without ever measuring the logical state itself, which would destroy it. A classical

decoder running in real time takes the stream of parity outcomes, infers which physical errors best explain them, and applies the correction. The distance of the code is how far apart the grid spreads the information; a larger distance tolerates more simultaneous errors.

What Willow demonstrated

Google announced its Willow chip in December 2024: 105 superconducting qubits, and the result that mattered was scaling behaviour. Increasing the code distance from 3 to 5 to 7 halved the logical error rate at each step. Below a certain physical error rate, adding more physical qubits makes the logical qubit better, which is what "below threshold" means. Above that rate, adding qubits adds noise faster than the code removes it, and every earlier experiment had been on the wrong side of the line.

The overhead is the story now

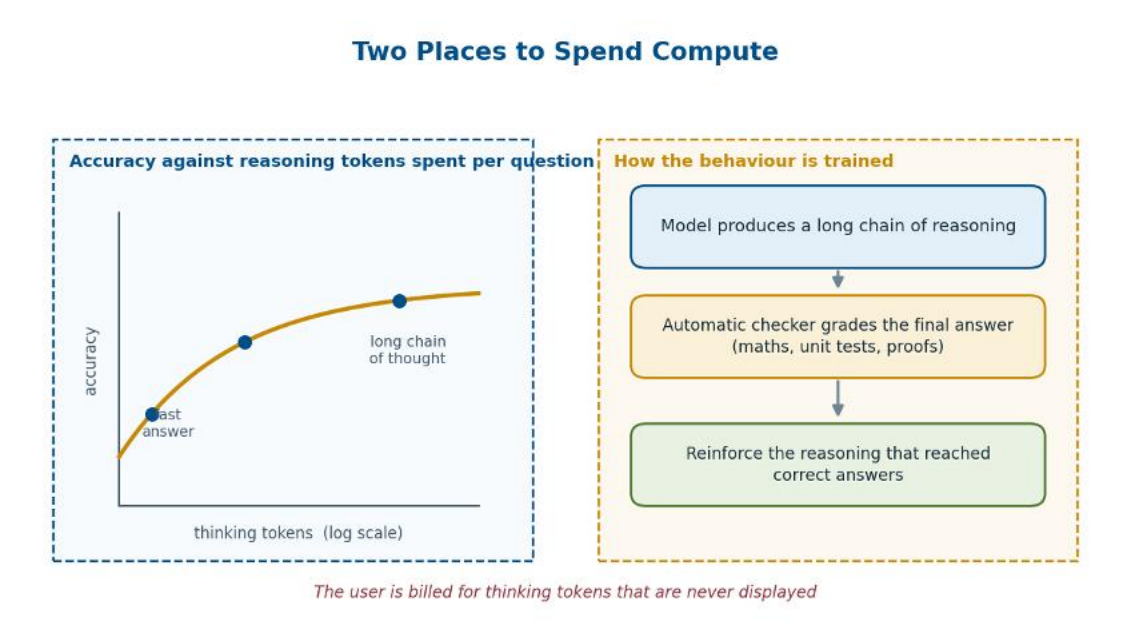
Current estimates put roughly a thousand physical qubits behind each useful logical qubit, and resource analyses for running Shor's algorithm against RSA-2048 land in the millions of physical qubits. Willow has 105. Nobody is breaking RSA this decade with that machine.

The reason cryptographers are migrating now is that two timelines overlap. Building the machine takes years. Replacing the cryptography inside every browser, certificate authority, smartcard and firmware image also takes years. Waiting for the first timeline to finish before starting the second is how you lose.

Thinking Costs Money: Reasoning Models and Test-Time Compute

Nagella Lokeshwar, 25MRD00061,
I MCA, Department of Computer Applications

Until 2024 the way to make a language model better was to make it bigger and train it on more data. That answer ran into money, data supply, and diminishing returns at roughly the same time. The replacement is to let the model spend more compute at the moment it answers, and it works well enough that the benchmark charts had to be redrawn with a new axis.



Accuracy now rises with tokens spent at inference, so the compute budget moved out of training.

What a reasoning model does differently

A reasoning model generates a long internal chain of thought before it produces the visible answer. It tries an approach, notices a contradiction, backtracks, and tries another. Those intermediate tokens are billed and counted even though the user never sees most of them. OpenAI shipped o1 in late 2024 and o3 during 2025 on this design, and the pattern is now standard across providers.

The training method is reinforcement learning on outcomes. The model produces a chain of reasoning, an automatic checker decides whether the final answer is right, and the reasoning that led to correct answers gets reinforced. On mathematics and code the checker is cheap to write, which is exactly why those two domains improved first and fastest.

DeepSeek-R1 and the open-weights shock

DeepSeek released R1 in January 2025 under an MIT licence, along with a paper describing the reinforcement learning recipe and a set of smaller models distilled from it into Qwen and Llama bases. Two things followed. Anyone with a university GPU could run a competent reasoning model locally, since the distilled 7B and 14B variants fit on hardware our labs already have. And the claimed training cost, well below what the industry assumed necessary, moved the argument about whether frontier capability requires a billion-dollar cluster.

The trade-off is real

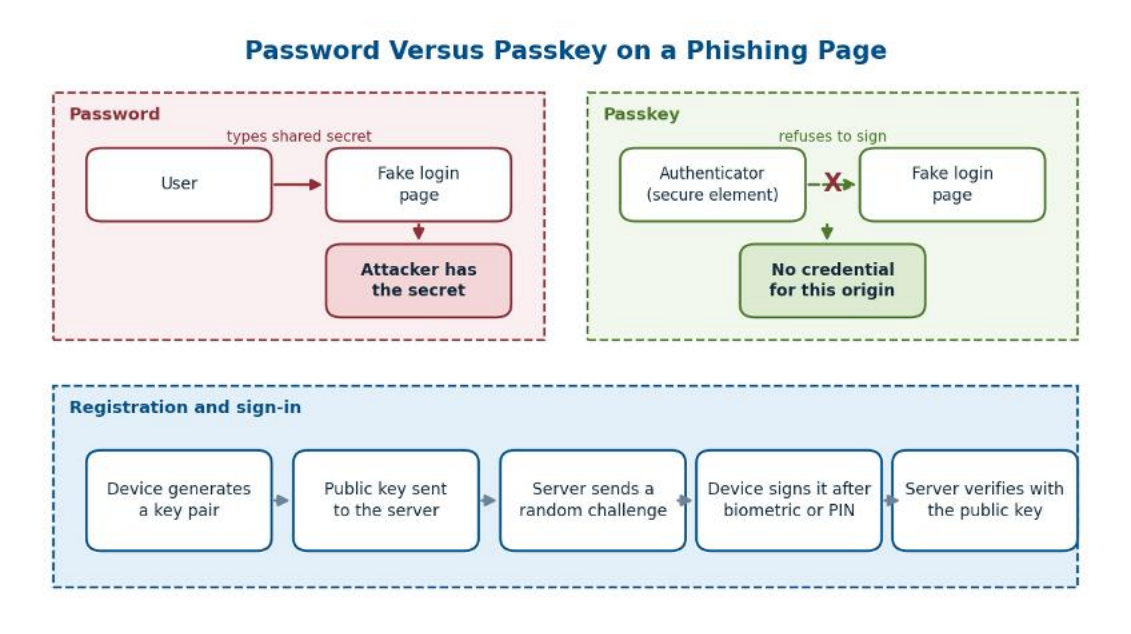
A reasoning model on a task that needs no reasoning is a waste. It takes ten to sixty seconds instead of one, and the token bill can run an order of magnitude higher because you pay for thinking you never read. Sorting support tickets does not need it. Proving that a distributed algorithm terminates does.

The design pattern that follows is a router: classify the incoming request first, send the easy majority to a fast model, and reserve the reasoning model for the queries that actually branch. Build the escalation path before you need it, because retrofitting a router into a system that assumed one model everywhere means touching every call site.

Passkeys: Why the Password Field Is Finally Disappearing

Konnepati Lahari, 25MRD00057,
I MCA, Department of Computer Applications

Every password scheme fails the same way. The user types a shared secret into a form, and if the form belongs to an attacker the secret is gone. One-time codes over SMS did not fix this, because the user will type the code into the fake form too. Push notifications did not fix it either, since a user tapping approve at 11 p.m. will approve anything.



A passkey is a key pair bound to one origin, which is what makes the phishing page useless.

The mechanism

A passkey is a WebAuthn credential. When you register, your device generates a public-private key pair, keeps the private key in secure hardware, and sends only the public key to the server. Signing in means the server sends a random challenge, your device signs it after a local biometric or PIN check, and the server verifies the signature against the stored public key. Nothing secret crosses the network, so a breach of the server database yields a list of public keys and nothing worth stealing.

The part that defeats phishing is origin binding. The browser hands the authenticator the origin of the page requesting the signature, and the authenticator will only use a credential registered for that exact origin. A convincing replica at mits-portal-login.example simply has no matching key, and the user cannot override this by being careless. The protection is structural.

Where it stands

Passkeys sync through the platform password manager, so registering on a phone and signing in on a laptop works without re-enrolling. Microsoft announced in May 2025 that new consumer accounts would be passwordless by default, with passkeys as the primary method. Google, Apple, Amazon and most large banks now accept them.

Building it

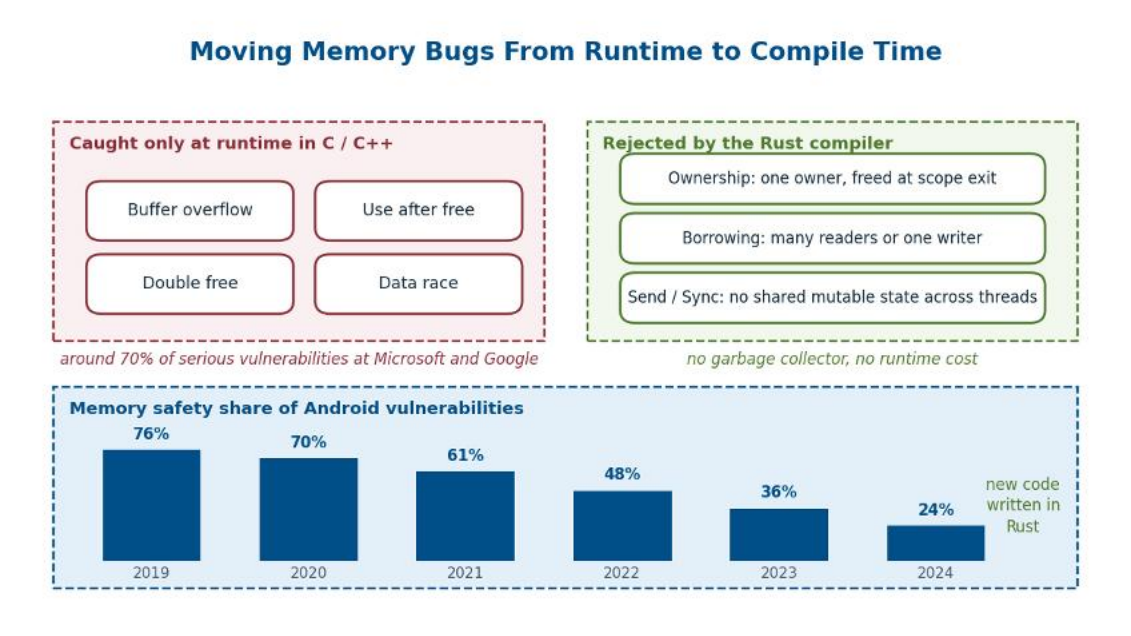
The browser API is navigator.credentials.create and navigator.credentials.get, and every major server language has a WebAuthn library that handles attestation parsing and challenge verification. Do not write that verification yourself. Two design decisions matter more than the code. Allow

several passkeys per account, because a user who registers only their phone and then loses it needs a recovery path. And make sure the recovery path is not a password reset email, since that quietly restores every weakness you removed.

Memory Safety at the Source: Rust Moves Into the Systems Layer

Gangavaram Manasa, 25MRD00066,
I MCA, Department of Computer Applications

Microsoft and Google have both published the same number from their own bug databases: around seventy percent of serious security vulnerabilities in their large C and C++ codebases are memory safety errors. Buffer overflows, use-after-free, double frees, data races. Sixty years of static analysers, sanitisers and code review have not moved that proportion, because the language permits the mistake and the programmer is human.



Ownership and borrowing move an entire class of bug from runtime to compile time.

What the compiler enforces

Rust's ownership model gives every value exactly one owner, and the value is freed when that owner goes out of scope. References are borrows, checked at compile time: either many readers or one writer, never both. A dangling pointer is not a bug you find in production; it is a program that does not compile. The same rules extend to threads, so a data race between two threads sharing a mutable value is also a compile error rather than a Heisenbug you chase for a week.

None of this costs runtime performance. There is no garbage collector and no reference counting unless you ask for it, because the checks happen before the binary exists.

The evidence from Android

Google reported that memory safety vulnerabilities in Android fell from about seventy-six percent of all vulnerabilities in 2019 to roughly twenty-four percent in 2024. The interesting part is how: Google did not rewrite Android. New code was written in Rust while old C and C++ was left alone, and the vulnerability count fell anyway, because most bugs live in new code. That finding

changed the advice from "rewrite everything", which nobody can afford, to "write the next module in a safe language", which any team can do.

Into the kernel

Rust support merged into Linux 6.1 in December 2022 as infrastructure, and real drivers have been landing since. Android's Binder IPC rewrite and the Nova GPU driver are the flagship cases. Progress has been contentious, since two languages in one kernel means two toolchains and arguments about who maintains the boundary between them.

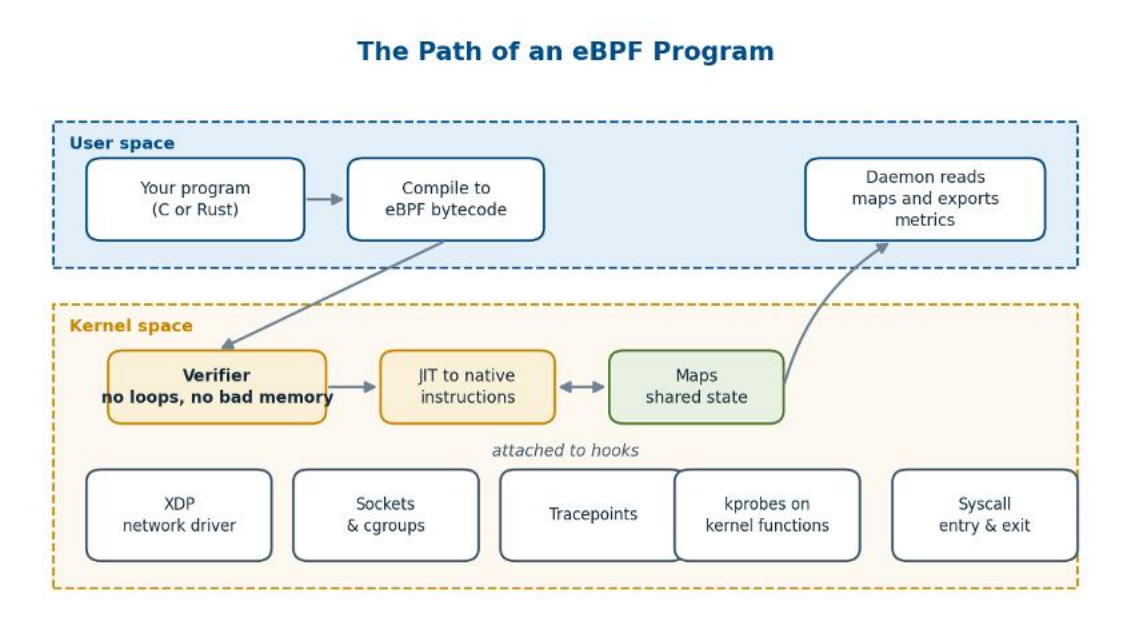
For your own work

The unsafe keyword exists and marks the blocks where you take responsibility, typically for hardware access or foreign function calls. Keeping those blocks small and auditable is the whole discipline. If you have written C for a systems lab, spend a weekend porting one exercise to Rust; the borrow checker will reject code you were sure was correct, and reading its complaints teaches you what your C was actually doing.

eBPF: Running Your Code Inside the Kernel Without Writing a Kernel Module

25MRD00075, Vempalli Nasreen
I MCA, Department of Computer Applications

Kernel space is where the interesting information lives. Which process opened which file, which packet was dropped and why, how long a syscall actually took. Getting at it used to mean writing a kernel module, which is a fine way to panic a production server, or copying enormous volumes of data to user space and analysing it there, which is slow and misses most of what you wanted.



Programs are verified before they load, attached to hooks, and share data with user space through maps.

How it works

eBPF lets you load a small program into a virtual machine inside the kernel and attach it to a hook: a network device, a tracepoint, a kprobe on almost any kernel function, a syscall entry, a cgroup. Before the program runs, a verifier examines it and rejects anything that could loop forever, read

out of bounds, or dereference an unchecked pointer. Only after it passes is the program JIT-compiled to native instructions. That verifier is why a bad eBPF program returns a load error instead of taking down the machine.

The program communicates with user space through maps, which are shared key-value structures the kernel side writes and your daemon reads. A ten-line program that increments a counter keyed by process ID, attached to a tracepoint, gives you a per-process syscall histogram with almost no overhead.

What it replaced

Networking came first. Cilium uses eBPF to implement Kubernetes service routing and network policy directly in the kernel data path, removing kube-proxy and its iptables chains, which mattered because iptables rule evaluation degrades as the rule count grows and a large cluster generates a great many rules. XDP hooks run before the kernel builds a socket buffer at all, which is how DDoS filters drop millions of packets per second on ordinary hardware.

Observability came second. Tools built on eBPF trace application behaviour without a sidecar, without recompiling, and without changing a line of the application. Security tooling came third, watching syscalls for the patterns a container escape produces.

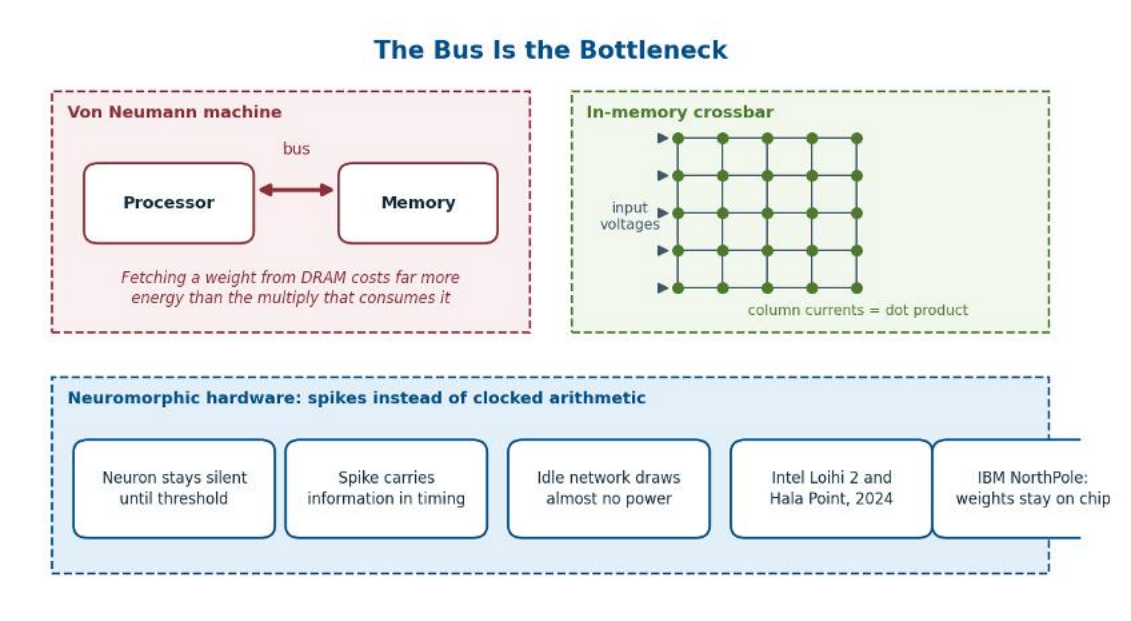
The catch

The verifier is strict and its rejection messages are famously hard to read, so the learning curve is steep in a way that has little to do with the language. Programs are tied to kernel internals, and the CO-RE mechanism exists specifically so a compiled program keeps working across kernel versions. A Windows port is in development, but for now eBPF means Linux.

Past von Neumann: Neuromorphic and In-Memory Computing

Marani Yaswanth, 25MRD00132,
I MCA, Department of Computer Applications

In a conventional machine the processor and the memory are separate, connected by a bus. Every operation fetches operands across that bus and writes the result back across it. For matrix multiplication, which is nearly all of what a neural network does, moving the weights costs far more energy than multiplying them. Estimates put the cost of a DRAM access at hundreds of times the cost of the arithmetic it feeds. Making the arithmetic units faster does not help when the data cannot arrive quickly enough.



The bottleneck is the bus. In-memory computing performs the multiply-accumulate inside the array.

Computing where the data already is

Analog in-memory computing performs the multiply-accumulate inside the memory array itself. Arrange resistive memory cells, phase-change memory or memristors in a crossbar, encode the weights as conductances, apply the input vector as voltages on the rows, and Ohm's law and Kirchhoff's current law produce the dot product as the current summed on each column. One matrix-vector multiply, one step, no weights moved. Reported energy efficiency for this operation is one to two orders of magnitude better than a digital equivalent.

Spiking hardware

Neuromorphic chips take a different route and imitate the brain's communication pattern instead of its arithmetic. Neurons stay silent until their accumulated input crosses a threshold, then emit a spike; information is carried in the timing of spikes and a quiet network consumes almost nothing. Intel's Loihi 2 is the best-documented research platform, and the Hala Point system Intel announced in April 2024 assembled 1,152 of those chips into a machine modelling around 1.15 billion neurons. IBM's NorthPole took a digital approach to the same idea, keeping all weights on-chip so inference never touches external memory.

Why you cannot use it yet

Analog devices drift, vary between cells, and quantise badly, so accuracy has to be recovered through hardware-aware training. Spiking networks lack an equivalent of backpropagation that works as reliably, and converting a trained conventional network into a spiking one loses accuracy. There is no PyTorch for this hardware; each platform has its own toolchain and its own small research community.

The pressure to solve those problems comes from the energy figures in the data centre article elsewhere in this issue. Digital accelerators are approaching the limits of what process shrinks will give them, and edge devices running vision models on a battery have a hard ceiling that better software cannot lift.

Magazine Student Editors

KANDULA HARSHAVARDHAN
Roll No.: 24691F0005

AMBITI SAI VAMSI,
Roll No.: 25MRD00103
Contact: mcaoffice@mits.ac.in
Visit us: www.mits.ac.in/MCA

EDITORIAL BOARD

CHIEF EDITOR

Dr. N. NAVEEN KUMAR

HOD/MCA

FACULTY EDITOR

Dr. R. MARUTHAMUTHU

Assistant professor / MCA

MITSTECH_MCA