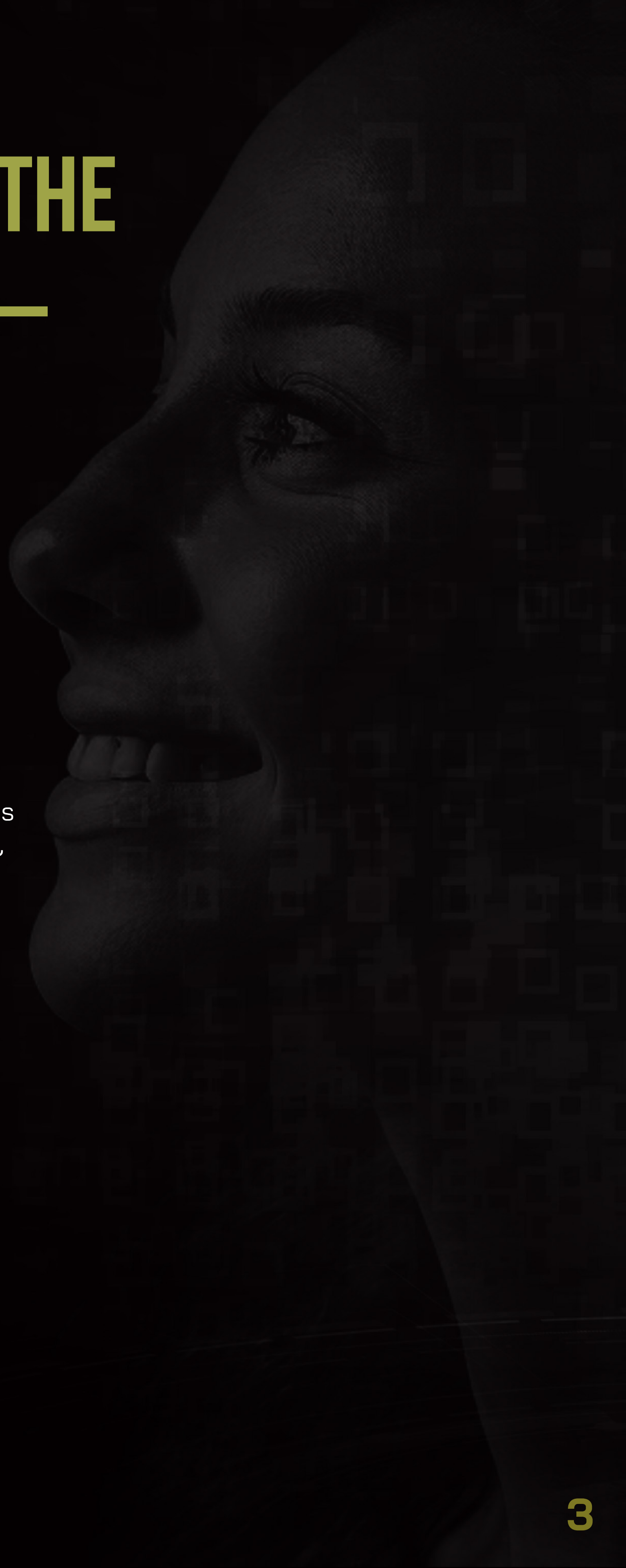
THE PROBLEMS WE CREATED BY SOLVING PROBLEMS

Why The Technologies Reshaping Our World Are Creating Challenges We Never Expected

EM360

CONTENTS

WE SOLVED THE PROBLEM — THEN THE SOLUTION GREW UP



Technology's greatest challenges increasingly come from success, scale, and the dependencies they create.

For a long time, technology felt like a story about improvement.

Faster communication.	Easier access to information.
Less paperwork.	Fewer queues.
Better logistics.	Smarter decisions.

More services available from a phone instead of a counter, a branch, or a call centre with hold music that made you question your life choices.

Those were real gains. They mattered. They still matter. The modern digital economy didn't appear because people fell in love with dashboards. It appeared because digital systems solved real problems well enough that individuals, businesses, and public institutions kept building around them.

That success changed the terms of the conversation. When a tool is optional, its flaws are annoying. When it becomes essential, those same flaws become structural.

The cloud is no longer just a convenient delivery model. Digital identity is no longer just a login box. Online services are no longer an extra channel. Automation is no longer just a productivity feature.

They have become the working assumptions underneath:	
Banking	Healthcare
Education	Logistics
Government access	Security operations
Supply chains	Everyday administration

That's the hinge point for this whole report. A lot of today's most serious technology problems aren't signs that technology failed. They're signs that technology worked, scaled, and became embedded deeply enough to produce **second-order consequences.**

- We made communication frictionless, then discovered that misinformation and overload also travel quickly.
- We made compute scalable, then ran into energy, water, land, and grid constraints.
- We made services digital first, then found that **"available" and "usable" aren't the same thing.**

We automated tasks, then learned that responsibility doesn't disappear just because the action is machine-driven. It moves. Sometimes it moves quietly. That's usually when the trouble starts. Across industries, policy discussions, infrastructure planning, technology strategy, and digital transformation initiatives, the recurring patterns are hard to miss:

- infrastructure that once felt invisible is becoming materially visible;
- scale is exposing environmental cost;
- trust is shifting from a security gate to a condition for growth;
- “connected” users are still being excluded from participation;
- and many supposedly mature systems are being held together by people doing extra unseen work.

These issues can look unrelated at first glance, but they’re connected by the same underlying force. **Dependency changes everything.** So this isn’t a doom piece, and it’s not an anti-technology piece either. It’s a maturity piece. The useful question is no longer only whether we can build more capable systems. Because we plainly can. The harder question is what those systems ask of the world once they work at scale.

What do they ask of power grids, water systems, workforce capacity, governance models, identity controls, public trust, and people who are already navigating systems that were supposedly designed to make life easier?

That last question matters because the next decade is going to look different from the last one. Capability will still matter, of course. No one is handing back fast networks, cloud infrastructure, digital payments, or machine learning because the consequences got complicated.

But capability on its own is no longer a convincing measure of maturity. If a system gets more powerful while becoming harder to govern, less understandable to humans, more resource-intensive, more exclusionary, or more fragile across dependencies, that’s not clean progress.

It’s progress carrying a bill that tends to arrive later, in smaller print, and at the worst possible moment. The central argument of this report is simple enough to say in one sentence, but large enough to change how organisations think about technology strategy.

We don’t simply live with technology anymore. We increasingly depend on it. And once dependency enters the room, the challenge is no longer creating powerful systems. It’s understanding what those systems require once people start relying on them.

THE CONSEQUENCES WE COULDN'T SEE

The more technology disappeared into the background, the easier it became to forget what it depended on.

For years, digital technology became easier to use by becoming harder to see. That convenience brought enormous benefits, but it also changed how people thought about scale, growth, and dependency. The systems underneath never disappeared. Most of us simply stopped noticing them.

Technology used to feel weightless

One of the most important things digital technology ever did was hide itself. That sounds like criticism, but it was actually part of the product. Cloud services were marketed as "elastic." Apps made complex processes feel like a tap.

Digital payments turned cash handling, reconciliation, and physical processing into something that happened elsewhere. Automation made work seem lighter because people no longer had to perform each step manually.

Intelligence started to feel like a feature you could layer into software. Ask a system. Get an answer. Move on. Very neat, frictionless and very convenient.

> *"Any sufficiently advanced technology is indistinguishable from magic." ~ Arthur C. Clarke, Science fiction writer and futurist*

Abstraction is the technical word for that move. In plain English, abstraction means hiding the messy parts so people can use a system without understanding every layer underneath it. That's often a good thing. You don't need to understand packet routing to join a video call, or storage architecture to upload a file.

The problem isn't abstraction itself.

The problem is what abstraction trained us to assume. It trained us to talk about digital growth as if it had no physical edge, no social cost, and no meaningful limit apart from budget. If more demand appeared, the answer was usually thought to be more cloud, more compute, more bandwidth, more integration, more tools.

Just scale it.

But digital systems were never weightless. They always depended on buildings, chips, cables, cooling systems, electricity, maintenance, labour, minerals, network hardware, software supply chains, regulation, procurement, and trust.

Abstraction didn't remove those dependencies. It mostly just removed them from view. That distinction matters because hidden dependency is still dependency. It's just dependency people stop planning for properly.

According to the **International Energy Agency**, data centres consumed approximately **415 terawatt-hours of electricity in 2024**, equal to roughly **1.5% of global electricity demand**. The IEA expects that figure to more than double by 2030 as AI and digital infrastructure continue expanding.

The significance isn't the percentage itself. It's that digital growth increasingly creates physical infrastructure requirements that somebody has to power, cool, finance, and manage.

This shift is becoming easier to see across the digital economy. Compute often feels remote and almost unreal until demand suddenly runs into power, cooling, or infrastructure constraints.

Connectivity feels effortless from the user side, yet every dashboard, security check, message, and meeting depends on data centres, fibre networks, devices, towers, and electricity operating behind the scenes.

Digital systems can feel detached from material reality even when they're anything but.

We should care about this because because invisible systems invite sloppy thinking.

If digital scale feels infinite...

- Growth becomes treated as a software challenge

If services feel instant...

- Recovery routes get underestimated

If automation feels clean...

- Oversight and exception handling get overlooked

If identity checks happen in the background...

- Trust becomes neglected

The first major problem created by successful digital technology wasn't the technology itself. It was the illusion that digital expansion was clean, detached, and endlessly available.

"Everything should be made as simple as possible, but not simpler." ~ Albert Einstein, Theoretical physicist

The more you depend on abstraction, the more painful the return to reality becomes. That return is already happening.

Energy markets notice it. Regulators notice it. Utilities notice it. Water planners notice it. Accessibility advocates have been noticing it for years.

Security teams definitely notice it, usually at 02:00, with three tabs open, six alerts blinking, and a great many opinions about whose "simple integration" caused the mess.

The old assumption that digital systems float above the real world is breaking down because the real world keeps sending invoices.

The physical cost of digital life is becoming impossible to ignore

If abstraction made digital expansion feel clean, AI has done the opposite. It has made compute visible again. The International Energy Agency's 2025 Energy and AI report was published because policymakers and operators no longer have the luxury of treating AI demand as background noise. The IEA states the point plainly: **"There's no AI without energy."** Specifically, electricity for data centres.

The report exists to model how much electricity AI could consume over the next decade and what that means for energy security, emissions, and affordability. The numbers made the issue harder to ignore.

The **International Energy Agency** estimates that electricity demand from data centres worldwide around

415 terawatt-hours **in 2024**

and could more than double to..

945 terawatt-hours annually **by 2030**

An amount greater than the total electricity consumption of many industrialised countries today.

The same reporting said **data centres** could account for **nearly half of electricity demand growth** in the **United States** through **2030**, while about **20% of planned projects face delays** because of **grid strain and infrastructure shortages.**

That's not a niche technical constraint. That's digital growth colliding with planning permission, substations, transmission, and time. The United States gives a very concrete example of this shift.

A U.S. Department of Energy-backed **Lawrence Berkeley National Laboratory report** estimated that **data centres** accounted for more than **4% of U.S. electricity use in 2023** and could **rise to between 6.7% and 12% by 2028**, driven largely by AI workloads.

In other words, a category that used to sit quietly behind the scenes is becoming a major planning variable for the grid itself. And electricity is only one piece of the story. Compute also needs cooling, land, hardware supply chains, and often large volumes of water.

A **2026 academic study on data-centre water capacity** warns that **cooling-related withdrawals** are becoming a meaningful **bottleneck for host communities**.

Its modelling suggests that if current water-use intensity persists, **U.S. data centres** could require **697 to 1,451 million gallons per day** of new water capacity through **2030**, roughly comparable to **New York City's average daily water supply.**

That's not a universal forecast for every region, and it's still modelling rather than a settled operational fact, but it usefully makes one point unavoidable: Digital infrastructure competes with physical infrastructure.

Research published in Nature estimated that global AI-related water consumption could reach **4.2 to 6.6 billion cubic metres annually by 2027**, depending on adoption patterns and infrastructure design. Water is increasingly becoming part of the digital infrastructure conversation, not just an environmental one.

There's also a localisation problem. Recent modelling of AI data-centre growth finds that capacity is concentrating heavily in North America, Western Europe, and the Asia-Pacific, with local grid vulnerability rising in places such as Oregon, Virginia, and Ireland.

This is a useful corrective to the lazy phrase "global demand." Demand is global. Pressure is local. That means the practical consequences show up first in particular communities, particular grids, and particular water systems, not in some abstract planetary spreadsheet.

This is where sustainability stops being something you talk about at the end of the slide deck.

If digital decisions have physical consequences, then sustainability becomes a design issue. Not a branding exercise or a quarterly ESG paragraph. A design issue.

- Which workloads need to run where?
- What cooling model is appropriate?
- Which power sources are realistic?
- How much redundancy is worth the environmental trade-off?
- What level of utilisation is being assumed?
- What happens if demand spikes harder than expected?

These questions are no longer peripheral because the cost of getting them wrong no longer stays hidden.

According to Uptime Institute's Global Data Center Survey, power availability has become one of the most significant constraints affecting new data centre development, with operators increasingly reporting difficulties securing sufficient electricity capacity in major markets.

Quantum is useful here, not because this report is "about quantum," but because it shows how future capability creates present planning pressure.

NIST finalised its first **three post-quantum cryptography standards** in **August 2024** and explicitly urged system administrators to start transitioning **"as soon as possible,"** noting that full integration will take time.

That's exactly the pattern this report is describing. A technology doesn't need to be fully mainstream before it starts creating operational consequences. Sometimes the consequence arrives first as:

Migration work	Standards work
Procurement work	Governance work

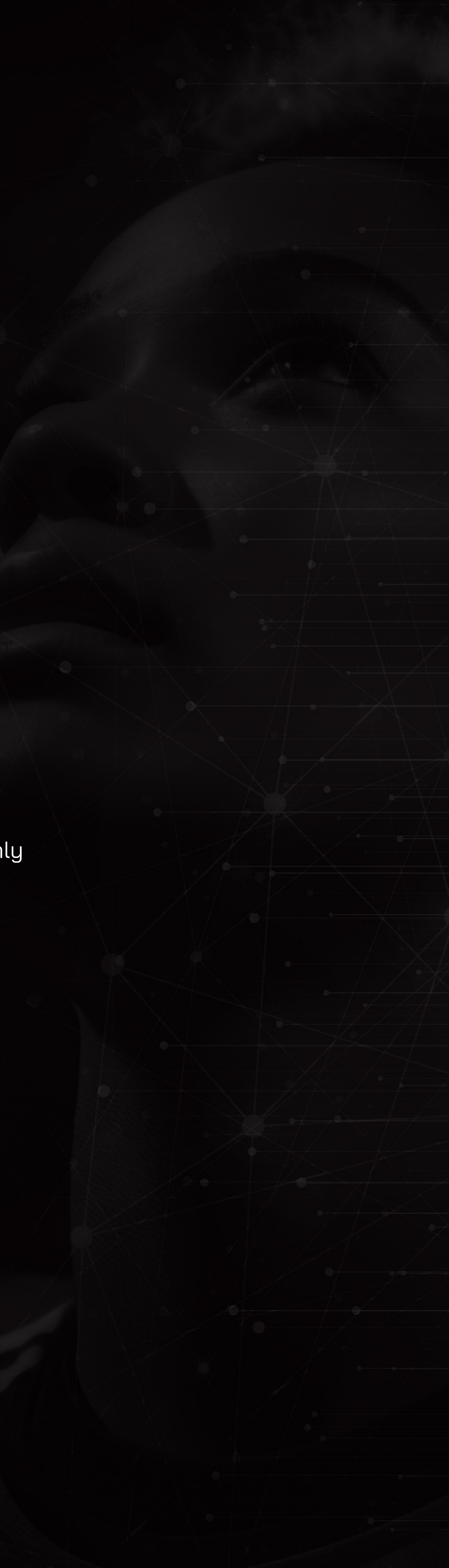
The future doesn't wait politely until your roadmap is ready.

So the argument here isn't **"AI uses energy."** That line is true, but too small. The larger point is that we solved limited access to computing by making compute scalable and abstracted. Now that solution has scaled far enough to confront us with its material footprint.

Compute still looks like a digital service. Increasingly, it behaves like a physical system. Or it **still feels like software at the point of use**. It just **no longer behaves like software only.**

THE CONSEQUENCES WE HANDED TO HUMANS

Automation often removes tasks more cleanly than it removes responsibility.



For decades, automation was framed as a way to reduce work. In many cases it has done exactly that. But as systems become more capable and autonomous, a different question emerges: when technology takes action, where does accountability go?

Automation didn't remove human responsibility

Automation is usually sold as subtraction.

- Fewer repetitive tasks.
- Less manual handling.
- Less waiting.
- Less human error.

Sometimes that's exactly what happens.

A well-designed automated process genuinely can remove drudgery and free people for more valuable work. But it's striking how often automation removes one kind of work while creating several others.

- Validation.
- Monitoring.
- Escalation.
- Exception handling.
- Policy design.
- Audit.
- Rollback.
- Explanation.

The task disappears. The responsibility doesn't.

That shift becomes much more serious when systems stop merely recommending and start acting. A chatbot that drafts text is one thing. An AI agent that triggers a payment, changes a configuration, makes a decision about access, or initiates a workflow is another.

As action becomes more automated, the practical question changes from: "Can we use this?" to "Under what conditions can we trust it enough to let it do something **real**?"

That's a governance question before it's a feature question.

According to Gartner, by 2028 at least 15% of day-to-day work decisions will be made autonomously through agentic AI, up from virtually none in 2024.

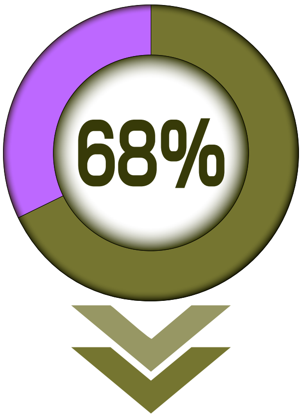
NIST's AI Risk Management Framework is helpful here because it treats "trustworthiness" as something that needs to be designed into the use, evaluation, and management of AI systems, not added afterwards.

NIST says the AI RMF aims to help organisations manage risks to individuals, organisations, and society, and to incorporate trustworthiness considerations into AI products, services, and systems.

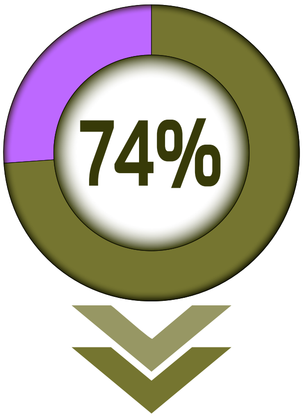
In **2026 NIST released a concept note for a critical-infrastructure profile**, which is a polite but unmistakable sign that autonomous and AI-enabled capability is already creating governance pressure in essential systems.

The same point appears in enterprise research on AI agents.

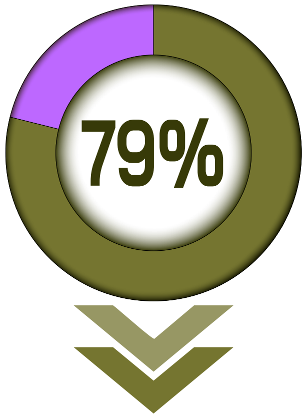
In a **2026 survey** reported by ITPro from research conducted by the Cloud Security Alliance and Aembit:



of organisations can't accurately distinguish AI agent activity from human activity



said agents often receive more access than necessary



believe agents create difficult-to-monitor access pathways

That's a remarkable sentence if you stop and sit with it for a second. Many organisations are moving towards systems that can act, while admitting they often can't clearly tell who did what and what those systems are allowed to reach.

This is why **"human in the loop"** matters, even if the phrase has become a bit battered from overuse. In plain language, it means a person still needs enough contextual understanding to know:

- When to trust a system
- When to interrupt it
- When to review it
- How to explain what happened afterwards

Human oversight isn't there because humans are old-fashioned. It's there because automated systems create accountability gaps the moment something important goes wrong.

If a machine acts on behalf of the organisation, the organisation still owns the action.

No one gets to point at the algorithm and pretend it was an act of Mother Nature.

There's a deeper change underneath this. Automation used to be discussed as labour reduction. Increasingly it's better understood as **"responsibility redistribution."**

Some responsibilities move to developers, some to operations teams, some to risk owners, some to managers, some to frontline staff who now have to explain system outcomes they didn't personally decide, and some to customers who must navigate automated processes with fewer human escape routes.

The **"**savings**"** are rarely distributed as neatly as the business case suggests.

A growing body of research into automation bias shows that people can become more likely to accept system recommendations even when warning signs are present. The risk isn't simply trusting automation too little. It's trusting it without maintaining the knowledge needed to challenge it.

That's why the dangerous version of automation isn't flashy failure. It's **"quiet over-delegation."** It's the moment a system becomes normalised enough that people stop noticing how much judgment has been outsourced without corresponding investment in accountability.

Technology solved slow, manual work in many contexts. The new problem isn't whether work can be automated. It's whether responsibility remains governable once it is.

Trust is becoming the thing everything runs on

Trust used to sound like a security department word. Something about authentication, permissions, compliance checkpoints, and awkward policy documents no one reads until an incident happens.

That picture is now too small. Once systems become more distributed, more automated, and more API-driven, trust stops being a perimeter issue and starts becoming infrastructure.

It becomes the condition under which action is allowed to happen safely.

The shift is becoming increasingly visible across modern enterprise systems. As systems grow more autonomous and interconnected, the challenge is no longer just whether a user can log in, but whether organisations can trust automated decisions enough to scale them safely. That's an important change in operational logic.

The question is no longer simply: **"Who may enter?"**

It is: **"Who or what may act, under what conditions, with which permissions, and with what evidence afterwards?"**

According to the World Economic Forum's Global Cybersecurity Outlook, digital trust is increasingly becoming a prerequisite for economic participation, with organisations reporting that trust influences adoption, customer behaviour, partnerships, and regulatory outcomes. So the issue is no longer simply preventing breaches. Instead it's enabling systems, services, and relationships to operate at scale.

This matters even more because non-human identities have exploded. Different research bases report different ratios, but the direction is consistent.

An **Obsidian Security report** said **machine identities** such as **API keys, service accounts, and certificates** now **outnumber human identities** by more than **100 to 1** in many enterprise environments and may **approach 500 to 1** in some sectors.

Other 2025 research cited in identity-security reporting placed **non-human identities at roughly 82 to 1 relative to humans.**

You don't need to worship any one specific number to grasp the real point. The centre of gravity has shifted. A great deal of modern trust is no longer about humans authenticating once a day. It's about machines, services, and agents continuously authenticating to one another.

CyberArk's 2025 Machine Identity Security Report found that machine identities are growing dramatically faster than human identities, with many organisations reporting difficulty maintaining visibility into certificates, service accounts, API credentials, and workload identities.

That changes what identity means. Identity is no longer just proof of who someone is. Identity now applies to a:

- Person
- Service
- Device
- Agent
- Workload

And whether they should be allowed to do something, in this context, right now, with these data, and with this level of risk. If you can't answer that clearly, trust becomes guesswork. And guesswork doesn't scale well in complex systems.

"What gets measured gets managed." ~ Peter Drucker, Management consultant and author

There's also a friction problem here. Too little trust creates risk because systems are exposed. Too much friction creates a different kind of risk because people find workarounds, share accounts, over-provision service access, or bypass controls entirely to get work done.

Mature trust design isn't about maximum restriction. It's about **precise, explainable, reviewable permissioning** that lets useful action happen without turning every workflow into a hostage negotiation with the helpdesk.

This is one of the clearest examples of the report's larger thesis. We solved the problem of access by connecting more people, systems, and tools. The consequence is that trust itself has become a scaling constraint.

Once everything is connected, proving what should be trusted becomes core infrastructure, not background admin.

The human brain became the backup system

There's a particular kind of modern workplace absurdity that almost everyone recognises.

- Too many dashboards.
- Too many alerts.
- Too many tabs.
- Too many approvals.
- Too many tools that are technically integrated but emotionally very much not.

Every system promises visibility, but the human being in the middle still has to notice what matters, reconcile conflicting signals, remember exceptions, chase decisions, and explain outcomes to other people.

Somewhere along the way, **"digital maturity"** became suspiciously good at producing more interfaces for humans to interpret.

Across enterprise operations, a recurring pattern is becoming harder to ignore. Mature systems reduce human effort, while immature systems redistribute complexity onto people.

Operational complexity can become a form of enterprise debt that builds up when systems become harder to understand, govern, and recover from than the organisation is designed to handle.

Put those together and the pattern is obvious: many systems now scale by leaning more heavily on human cognition, not less.

This isn't just a wellbeing story, though it's that too. It's a systems story. Human attention is finite. Context-switching is costly. Alert saturation makes weak signals harder to see rather than easier.

Research on **user attention** under cognitive load shows that **as cognitive load increases, people's ability to attend to interface elements decreases.** That finding comes from interface research, but it maps uncomfortably well onto the real world of enterprise operations and digital service environments.

If you build systems that require constant human interpretation to stay safe, those systems aren't especially mature. They're just politely outsourcing complexity to the nearest person with a pulse.

"Every system is perfectly designed to get the results it gets." ~ W. Edwards Deming, Statistician and management theorist

This is also why more observability isn't automatically more understanding. Many enterprises don't have a data shortage. They have a "meaning shortage."

That's a sharp distinction. More dashboards don't solve the problem if the actual burden of judgment still falls on overstretched humans trying to decide:

- Which alert is real
- Which anomaly matters
- Which workflow can wait

The practical maturity test is simple. If your system only looks stable because people are constantly compensating for it, it's not actually stable. If every increase in scale requires:

- More manual reconciliation
- More permissions review
- More exception meetings
- More Slack messages
- More "just quickly" interventions from staff

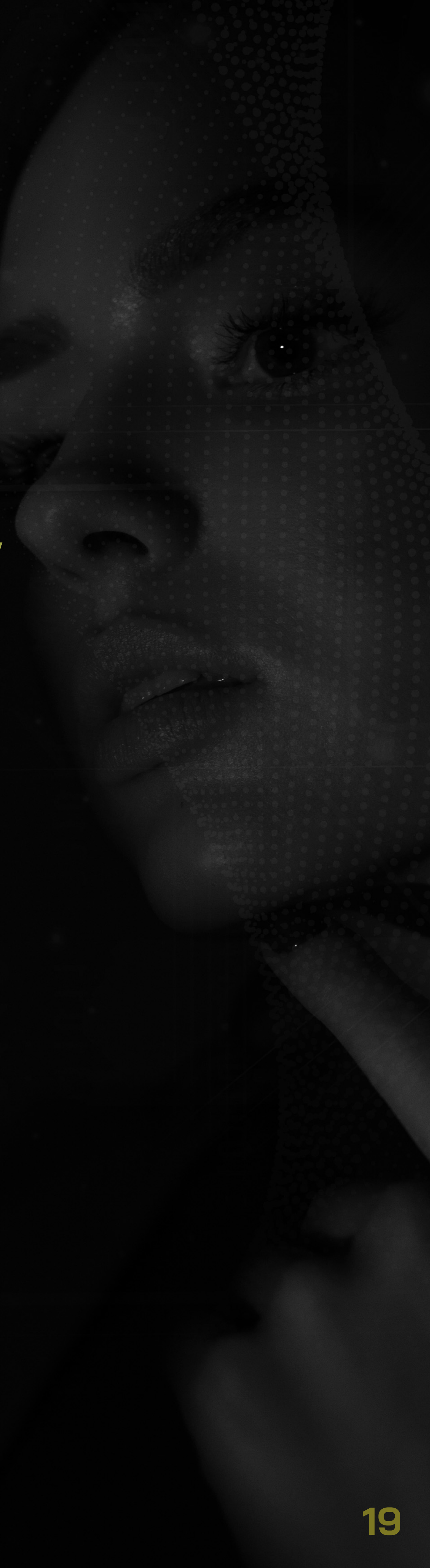
then scale isn't solving complexity.

It's moving complexity. Usually uphill, into the human brain, which was never designed to be the universal adapter for software sprawl.

Technology solved fragmentation in many areas by giving organisations more tools. The new problem is that people are becoming the integration layer between those tools. That's not a niche operations complaint.

It's one of the clearest signs that the consequences of digital success are now landing in human capacity.

THE CONSEQUENCES OF DEPENDENCY



Connecting people was the first challenge. Helping them participate is becoming the harder one.

Digital transformation is often measured by access, coverage, and adoption. Those metrics still matter. But as more of life moves online, a more important question emerges: what happens after someone connects?

Being connected isn't the same as being included

For years, the digital divide was framed as an access problem. Some people were online. Some weren't. Build more infrastructure, get coverage up, and you have solved the main issue.

That was never the whole picture, but it was useful as a starting point. It's much less useful now. Today the harder question isn't only whether someone can connect.

It's whether they can **"meaningfully participate"** once they do. The progress story is real.

According to **ITU** figures reported in late **2025**, around **6 billion** people were online globally while **2.2 billion** remained offline.

The same **ITU** data show that **5G** now covers **55%** of the world's population

though the gap between high-income and low-income countries is stark: **84%** coverage versus just **4%**

The report also notes that a typical user in a high-income country generates **nearly eight times more mobile data** than a user in a low-income country, and that affordability remains a barrier in **about 60% of low- and middle-income countries.**

GSMA reporting takes the argument a step further. More than **96% of the world's population** now lives **within range of mobile broadband,** but **more than 3 billion people remain offline** because of the "usage gap," not the "coverage gap."

The barriers include:

- Affordability
- Lack of electricity
- Lack of digital skills
- Content or services that don't meet users where they are

That's a very revealing shift. We are moving from a world where infrastructure availability was the main bottleneck to one where meaningful use is increasingly the bottleneck.

That matters because digital participation is no longer peripheral. Work applications, benefits claims, health appointments, education access, banking, identity verification, and customer support increasingly assume digital confidence as a baseline.

When systems are designed around always-on digital participation, exclusion is no longer just about being offline.

It can happen through:

- Cost
- Poor design
- Language mismatch
- Inaccessible verification
- Distrust
- Low digital literacy
- Unreliable service quality

A person can be technically connected and still effectively shut out.

This is where the conversation becomes bigger than enterprise architecture. Participation is about dignity as much as access.

If essential systems are designed for the confident, well-connected, well-provisioned user with recent hardware, stable data access, strong literacy, and plenty of time, then the system isn't neutral.

It's selecting for who gets to participate smoothly. Everyone else pays a **friction tax.** Sometimes in money. Sometimes in time. Sometimes in embarrassment. Sometimes in lost opportunity.

Research from the OECD has repeatedly found that digital skills are strongly correlated with employment opportunities, earnings potential, and access to public services. So as more systems become digital-first, participation increasingly depends on capabilities that extend beyond connectivity itself.

AI raises the threshold again. The question is no longer only whether people can access digital systems. Increasingly it's whether they can benefit from systems that are becoming more personalised, automated, and data-mediated.

If the quality of services, interfaces, or opportunities increasingly depends on AI-enabled systems, then "participation" starts to include the ability to understand, question, and benefit from those systems too.

That doesn't mean everyone must become a technical expert. It means the participation challenge is moving upward.

So yes, access still matters. Of course it does. But it's no longer enough. We solved part of the problem by getting more people online. The harder problem now is whether people can use digital systems to live, learn, work, access services, and make decisions with confidence.

That's what meaningful participation actually means.

A system can be online and still fail

One of the most misleading habits in technology is treating availability as success. The service is up. The site loaded. The app responded. The uptime target was met.

Great.

Except that a technically available service can still fail the person trying to use it. If they can't understand the interface, complete the journey, use assistive technology, recover from an error, or verify their identity, "online" isn't the same thing as "working."

> *"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."*
> *~ Tim Berners-Lee, Inventor of the World Wide Web and Director of the W3C*

This isn't an edge case. The **World Health Organization** estimates that **1.3 billion people, or 1 in 6 globally, experience significant disability**.

W3C's Web Accessibility Initiative says web accessibility is essential because poorly designed websites, apps, tools, and technologies create barriers that exclude people from using the web.

It also notes that accessibility benefits a much wider group than permanent disability alone, including:

- Older people
- Users on slow or expensive connections
- People with temporary impairments
- People facing situational limitations

- Older people
- Users on slow or expensive connections
- People with temporary impairments
- People facing situational limitations

In other words, accessibility isn't a compliance footnote. It's a service quality requirement. The real-world baseline isn't encouraging.

WebAIM's 2026 analysis of the top one million home pages found that

95.9% had detectable WCAG 2 failures

and the average page had

56.1 detectable accessibility errors.

Automated testing doesn't catch everything, so full conformance is likely lower than the already disappointing **4.1% implied by those findings.**

That should end the comforting fiction that accessibility is mostly solved or neatly contained within a few specialist sectors. It's not.

The enterprise implication is straightforward. As services become **"digital first,"** the fallback routes shrink. The old:

- Branch
- Receptionist
- Desk
- Phone support path
- Paper form

may still exist in theory, but often it's thinner, slower, or harder to reach than before. That means service design failures increasingly become **"access failures."** And when the service is essential, access failure becomes a form of exclusion.

This is why it helps to think of accessibility as infrastructure. Infrastructure has one basic job. It has to work for the people who depend on it. If the only route has become digital, then usability and accessibility are no longer "nice to have" qualities. They're part of whether the infrastructure is functioning at all.

A payment service that's technically online but impossible to navigate under pressure isn't resilient from the user's point of view. A public service portal that technically exists but excludes people through interface, language, verification, or compatibility barriers isn't actually fully available.

This point matters well beyond disability. It applies to anyone dealing with:

- Low bandwidth
- Stress
- Low confidence
- Fatigue
- Ageing devices
- Unfamiliar systems

Useful design isn't a cosmetic extra. It's part of whether digital-first systems deserve the trust we now place in them. We solved the problem of making services digital. The new problem is that digital services are becoming the only route, even when they don't work properly for everyone.

Failure doesn't stay where it starts

Connected systems are brilliant right up until they fail together. That's the trade. Integration removes friction when things are working and amplifies consequences when they aren't. A modern outage rarely stays contained inside one technical box because that box is connected to:

- Payment systems
- Authentication
- Customer support
- Logistics
- Partner APIs
- Cloud infrastructure
- Staff workflows

And the more efficient the chain, the farther failure can travel.

The **CrowdStrike incident in July 2024** is still one of the clearest demonstrations of this. **Microsoft said the faulty update affected about 8.5 million Windows devices**, which was **less than 1% of all Windows machines.** Yet the impact rippled across airlines, healthcare, finance, media, and government services.

The significance wasn't the percentage. It was the concentration. A single software failure reached deeply into already essential services because those services were operating atop shared dependencies.

That's what **failure propagation** looks like in practice. It starts somewhere specific and ends somewhere broader.

- We made communication frictionless, then discovered that An update problem becomes a flight disruption.
- An identity issue becomes a payments problem.
- A cloud dependency becomes a customer service outage.
- A telecom failure becomes a barrier to emergency access.
- The technical cause may be local. The lived consequence isn't.

This is why resilience can't mean only "prevent the outage." Prevention matters, obviously. But mature resilience also asks:

- Where failure goes
- Who depends on the dependency
- What degrades first
- What fallback actually exists
- Which groups have the fewest alternatives

Those are ecosystem questions, not just system questions. If your operations rely on external platforms, identity providers, payment rails, and shared infrastructure you don't control, then your resilience posture is partly a map of inherited dependencies.

There's a second uncomfortable point here. High availability stats can obscure social fragility. A system may be "up" almost all the time and still produce serious systemic risk because when it does fail, too much depends on it. In a world of essential digital systems, failure severity is shaped not only by frequency but by concentration.

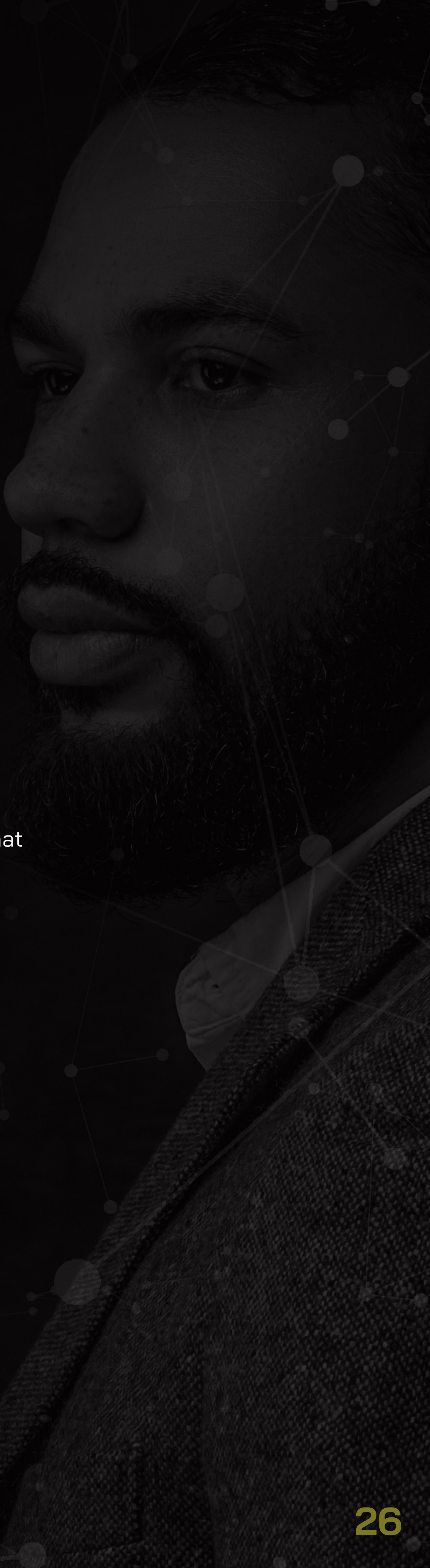
A rarely failing dependency can still be dangerous if half the operating model runs through it. That's a dependency problem, not an uptime problem.

The deeper lesson is that connection rearranges risk. We solved the problem of isolated systems by connecting them, standardising them, and integrating them. That often made services faster and better. It also created the conditions for failure to travel through the same connections that make success efficient.

The network that carries value also carries disruption. That's not a bug in the age of interdependence. It's one of its defining features.

THE CONSEQUENCES WE NEED TO DESIGN FOR

The next maturity challenge isn't building more powerful technology but managing what successful technology creates.



By this point, the pattern should feel familiar. The technologies themselves are rarely the problem. The challenge emerges when success creates dependencies, and those dependencies create new responsibilities that nobody planned for at the start.

The real problem isn't innovation — it's unplanned consequence

If you step back from all of this, a pattern emerges. Compute created physical responsibility. Automation created accountability responsibility. Connectivity created participation responsibility. Digital-first services created accessibility responsibility. Interconnection created resilience responsibility.

None of those responsibilities are accidental. They're what show up when useful technologies become normal enough to structure real life.

"We can't solve our problems with the same thinking we used when we created them." ~ Albert Einstein, Theoretical physicist

That's why it's too simple to frame current technology risk as "hype," "overreach," or "things moving too fast," though those can all be true in places. The larger issue is **unplanned consequence.**

We've been very good at building systems for capability, adoption, and scale. We've been less consistent at building for what those systems ask of the environments around them once they succeed. The result isn't an innovation crisis.

It's a **systems maturity gap.**

This is also where a lot of bad debates go wrong. People argue as if the choice is between embracing technology and resisting it. That's intellectually tidy and practically useless. The real question isn't whether innovation should continue.

It definitely will.

The question is whether organisations are treating **second-order effects** as core design requirements or as unfortunate side notes to be handled later by somebody else. "Later" usually turns out to mean after the dependency is already business critical, socially important, and unpleasantly expensive to redesign.

"An ounce of prevention is worth a pound of cure." ~ Benjamin Franklin, Founding Father, inventor, and statesman

In that sense, many emerging risks aren't signs that the original solution was wrong. The solutions were often right for the problem they addressed. The problem is that success scales. A good solution adopted widely becomes infrastructure. Infrastructure creates reliance. Reliance increases consequence.

And consequence forces new questions that didn't matter when the technology was merely helpful.

That's the conceptual move this report is trying to make. Stop asking whether a given technology is "good" or "bad" in the abstract. Start asking: "What happens when it works?"

Not in the demo. Not in the roadmap. In the real world, at scale, across dependencies, under pressure. And with uneven human capability, uneven access, uneven infrastructure, and imperfect governance.

That's where the next era of technology maturity will actually be tested.

What better technology maturity looks like now

If maturity is no longer just about capability, what does it actually look like? Not a consultant framework with seven coloured boxes, ideally. Something more useful than that.

A mature organisation now starts by asking what a technology depends on, not just what it promises.

It asks what physical systems sit underneath it, including:

- Electricity
- Water
- Hardware
- Networks
- Specialised labour

It asks what trust relationships it creates, including:

- People
- Devices
- APIs
- Service accounts
- Machine identities
- AI agents

It asks:

- What human work is genuinely removed?
- What human work is merely displaced?
- Who absorbs the new burden when the system becomes more complex?

It asks who can participate confidently and who is likely to be left behind by cost, design, access, or skill assumptions. It asks where failure will travel if one dependency degrades or disappears. And it asks whether the system remains understandable enough for people to challenge, govern, and recover.

That set of questions sounds deceptively simple because it is. But simplicity isn't the same thing as shallowness.

These questions push organisations back into contact with the real world.

They stop technology planning from drifting into the clean version that exists in a pitch deck and force it back into the messier version that exists in:

- Infrastructure
- Budgets
- Policy
- Service design
- Human behaviour

In practical terms, this means a few shifts.

1st **Sustainability has to move upstream**

If compute and connectivity have material costs, then energy, cooling, land use, hardware intensity, and reporting can't be treated as downstream clean-up exercises. They belong in architecture, procurement, and capacity planning.

2nd **Governance has to move closer to action**

The old pattern of letting capability sprint ahead while policy trails behind is becoming too risky in automated, API-rich, identity-dense environments. That doesn't mean slowing everything to a crawl.

3rd **Trust has to be treated as an operational architecture problem**

It means governance must be operational enough to shape permissions, oversight, rollback, and accountability in living systems, not just retrospectively document what went wrong.

4th **Human capacity must become a design metric**

Identity, permissions, credential hygiene, and auditability are no longer specialist back-office controls.

They're part of whether systems can safely scale at all.

If the organisation needs ever more meetings, approvals, manual checks, and alert triage to keep the stack safe, that's not proof of maturity.

It's a sign that complexity is outrunning governability.

5th Participation and accessibility have to be built in as service requirements, not added as late-stage fixes

As W3C makes clear, accessibility is essential to equal access and opportunity, and the web is increasingly important across education, employment, commerce, government, and healthcare.

If digital participation is now part of social and economic participation, exclusion is a core systems risk.

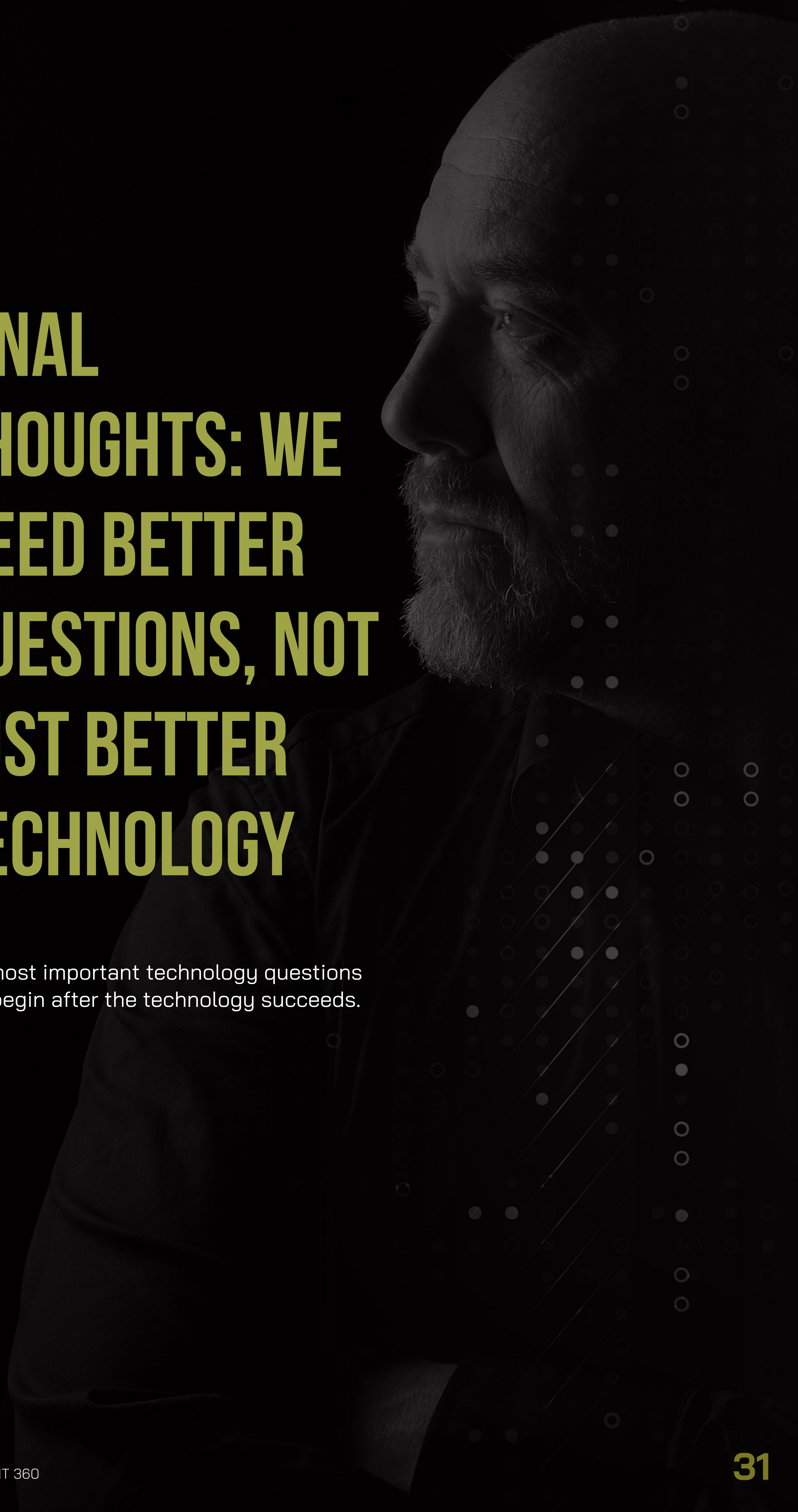
6th Resilience needs a broader definition

It should still include redundancy, recovery, security, and incident response. But it also has to include concentration risk, dependency mapping, fallback design, and the social consequences of digital-only pathways. Availability is necessary. It's not sufficient.

Put differently, the next maturity test isn't whether organisations can adopt more powerful technology. Many already can. The test is whether they can build systems that remain survivable, governable, understandable, inclusive, and worth the cost once those technologies work at scale. That's a harder test. It's also the one that matters now.

FINAL THOUGHTS: WE NEED BETTER QUESTIONS, NOT JUST BETTER TECHNOLOGY

The most important technology questions now begin after the technology succeeds.



Technology will keep advancing. That's not the problem. We will get better models, denser infrastructure, more automation, smarter sensing, sharper optimisation, and more services that move quietly from novel to normal. Some of that will be enormously valuable. Some already is.

The question is whether our way of evaluating success is keeping pace with what success now creates.

For years, most technology conversations were organised around capability.

- "What can it do?"
- "How fast is it?"
- "How accurate is it?"
- "How cheaply can it scale?"

Those are still useful questions. They're no longer enough. The strongest organisations in the next decade will not be the ones that only ask what technology can do. They will be the ones that also ask what technology asks of the world around it.

What does it ask of:

- Infrastructure
- Energy
- Water
- Trust
- Labour
- Public confidence
- Accessibility
- Fallback routes
- People who didn't design the system but still have to live inside it

That's the real lesson hiding under all of this. Technology created new problems not because it failed, but because it worked. It became:

- Useful enough to become essential
- Efficient enough to become default
- Scalable enough to become structural
- Familiar enough to disappear into everyday life

Once that happens, every design choice becomes more consequential. What was once a product decision can become a social decision. What was once an engineering trade-off can become a governance issue. What was once a small usability flaw can become exclusion. What was once a local dependency can become system-wide fragility.

So no, we don't need less innovation. We need innovation designed with its consequences in mind.

Future technology maturity will be measured less by raw capability and more by whether organisations understand the consequences of scale early enough to design for them properly.

Not after the outage. Not after the public backlash. Not after the regulator arrives. Not after the internal team is drowning in dashboards. Earlier than that. At the point where organisations start thinking beyond deployment and begin asking what success will require of the world around it.

That's where better systems begin. And it's why the most important technology question of the next decade is no longer

"What can we build?"

It is, much more uncomfortably and much more usefully,

"What happens when it works?"

EM360Tech

connects enterprise technology leaders with the insights, analysis, and conversations shaping the future of AI, data, security, and infrastructure.

For more insights,
visit www.em360tech.com
or contact us to collaborate on a thought leadership campaign:
info@em360tech.com