Assurance Is Not Compliance (and Why That Distinction Matters)

Dr. Liz James

Table of Contents

Author Context and Perspective

This paper is written from the perspective of a practitioner operating at the intersection of cybersecurity, safety, and complex socio-technical systems.

I am a Managing Security Consultant at NCC Group, a global cybersecurity and software resilience organisation. Across its practices, NCC Group's work can be understood holistically as supporting customers with assurance: helping organisations justify confidence in their products, systems, services, and operations in the face of evolving threats, regulatory pressure, and real-world complexity.

I sit within NCC Group's Consulting and Implementation practice, with a primary focus on Transport, Operational Technology (OT), and Cyber-Physical Systems (CPS). I have worked in this domain for over six years, supporting manufacturers, operators, and infrastructure owners across automotive, rail, maritime, and broader critical infrastructure contexts.

My route into industry was through academia rather than traditional IT or security operations. I completed an Engineering Doctorate (EngD) at WMG, University of Warwick, following an MPhys background. That path provided a particular induction into engineering as an applied, socio-technical discipline: one that spans innovation, systems thinking, regulation, and the often-messy interface between academic theory and industrial reality.

On entering the cybersecurity industry, I was trained initially through a condensed penetration-testing pathway and undertook a period of conventional technical engagements. This grounding was valuable, but it quickly became clear that many of the most significant risks in transport and OT environments do not arise from isolated technical vulnerabilities alone. They emerge instead from the interaction between technology, people, process, organisational structure, and regulatory interpretation. These industries were, and in some cases still are, replete with serious technical weaknesses. However, what proved more concerning in practice was not merely the existence of vulnerabilities, but their invisibility to those responsible for managing, governing, and taking accountability for the systems.

Over time, my work has centred increasingly on complex systems where assurance cannot be reduced to testing outcomes or compliance artefacts alone. In these environments, cybersecurity, safety, resilience, and trust are engineering properties that must be reasoned about explicitly and baked into every layer, from product design to organisational capability, to supply chain interfaces, to real-world operation and change.

The ideas in this paper therefore do not originate from abstract theory or regulatory analysis in isolation. They are shaped by repeated practical encounters with the limits of compliance-led assurance, the pressures faced by practitioners on all sides of the interface, and the need to explain, justify, and defend confidence in systems whose behaviour matters to people, safety, and society.

The position taken here is a pragmatic one: assurance is not a replacement for compliance, testing, or standards. It is the engineering discipline that gives those activities meaning, and it is increasingly essential for systems that must remain trustworthy as they scale, integrate, and evolve.

Executive Summary

Organisations routinely describe themselves as “assured” when they are, in practice, compliant. Compliance is concrete: named standards, auditable activities, and portable certifications. Assurance is different. It is the structured discipline of reasoning about trust: why should we believe this system, service, or organisation is trustworthy in its intended context, and what would change that judgement?

Compliance can contribute evidence to assurance, but it does not bound the assurance problem. Compliance-led approaches tend to collapse the reasoning space into a convex hull defined by standards and regulations, leaving emergent interactions, socio-technical harms, and novel misuse outside scope. This is why organisations can be compliant and still fail in the real world.

Effective assurance must span both product/system assurance and organisational assurance, and critically the interfaces between them. It depends on a credible Item Definition that includes technical boundaries and roles, responsibilities, lifecycle states, and operational context through ConOps and ODD. Defining the item is itself non-trivial, and many assurance failures originate from implicit or incomplete scoping rather than from a lack of controls.

The paper examines where assurance breaks down in practice: conflicting specialist lenses (safety, security, resilience), failures of reasonable foreseeability, and harm arising from the malicious use of intended functionality where systems behave exactly as designed yet produce unacceptable outcomes at scale or under changed conditions.

To address this, the paper advocates principles-based assurance, using stable principles (e.g., NCSC Technology Assurance Principles [1], and by analogy GDPR principles) as anchors that survive regulatory churn. Principles are translated into context-specific claims, which are then justified through explicit Claims–Arguments–Evidence (CAE) reasoning. CAE is presented as an aid to clarity and review, not an overhead, enabling peer critique, explicit caveats where evidence is substitution-based, and clear ownership of claims even when evidence is distributed across large organisations and supply chains.

Finally, the paper challenges absolute language such as “secure” or “safe”. There is no such thing as simply secure; these are risk judgements in context. Assurance does not eliminate risk. It enables defensible confidence under change, by making assumptions, trade-offs, and residual risk explicit.

1. Assurance as a Space for Reasoning

At its core, assurance is the structured space in which we reason about trust.

It exists to answer a deceptively simple question:

- Why should we believe this system, product, or organisation is trustworthy in its intended context?

That question is not rhetorical, nor is it answered once. It must be answered repeatedly as systems evolve, contexts change, and new information emerges. Assurance therefore exists not as a point-in-time judgement, but as a living reasoning space in which confidence is constructed, tested, and revised as context evolves.

1.1. Assurance Is About Justified Confidence

Assurance is concerned with **justified confidence**, not assertion. To say that something is trustworthy is to claim that there are good reasons to believe it will behave acceptably, even under stress or uncertainty.

Those reasons must be articulated. They must connect:

- what the system or organisation is intended to do,
- how it has been designed and implemented,
- how it is operated and governed,
- and what evidence exists about its real-world behaviour.

Without this chain of reasoning, confidence collapses into optimism, habit, or reputation. Assurance replaces intuition with explicit justification.

1.2. Assurance Is Not a Deliverable

Assurance is often treated as an output: a report, a certificate, a sign-off. This framing is convenient, but misleading.

Reports and certifications are *artefacts* of assurance activity. They are snapshots of reasoning at a moment in time. The assurance itself lies in the reasoning that produced them - and in the organisation's ability to revisit and revise that reasoning as conditions change.

When assurance is reduced to a deliverable, it becomes brittle. When the context shifts, the artefact remains static, and confidence decays unnoticed.

1.3. The Role of Judgement

Unlike compliance, assurance cannot be fully automated or reduced to procedure. It requires professional judgement.

Judgement enters when:

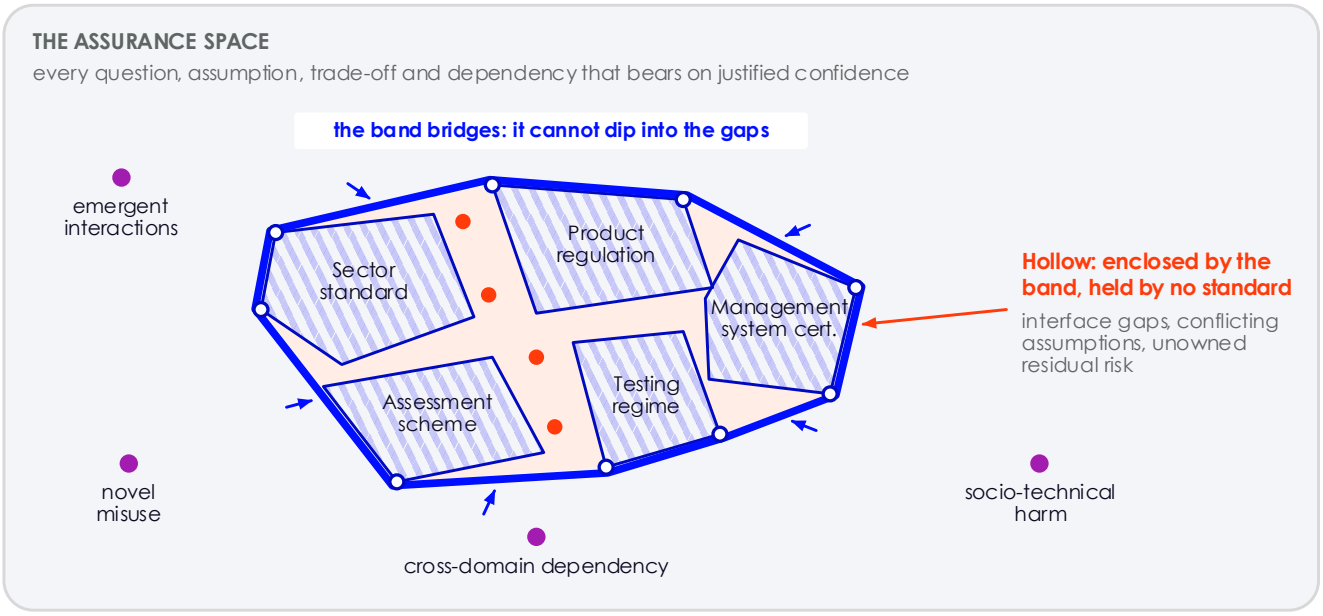
- trade-offs must be made between competing objectives,
- evidence is incomplete or indirect,
- assumptions must be accepted or rejected,
- or uncertainty cannot be eliminated.

Making this judgement explicit is not a weakness. It is what allows assurance to be examined, challenged, and defended.

1.4. Compliance as an Input to Assurance

Figure 1 Compliance is a band drawn taut around the standards

Pulled tight, the band rests on each standard at a few points and spans the gaps between them.The enclosed hollows are counted as covered . They are not.



Compliance activities such as audits, testing, assessments and certifications are often treated as synonymous with assurance. In practice, they are better understood as inputs into assurance reasoning rather than substitutes for it.

A useful way to visualise this distinction is spatial.

Assurance can be thought of as a reasoning space: the full set of questions, uncertainties, assumptions, trade-offs, and risks that must be considered in order to justify confidence in a system, service, or organisation.

Compliance-led approaches tend to collapse this space into a convex hull defined by regulations and standards. The questions that are asked, the evidence that is gathered, and the conclusions that are permitted are bounded by what is explicitly required or recognised by those frameworks. Anything outside that hull is implicitly treated as out of scope: emergent interactions, socio-technical effects, novel misuse, cross-domain dependencies.

Within this convex hull, compliance can be highly effective. It provides shared baselines, comparability, and enforceability. However, risks do not respect regulatory boundaries. Real-world harm often arises in the gaps between standards, at the interfaces between domains, or in behaviours that were never explicitly prohibited because they were not anticipated.

Assurance-led approaches resist this collapse. Instead of shrinking the reasoning space to fit existing frameworks, they use regulations and standards as shapes within a larger space. Compliance artefacts become evidence points inside a broader argument, rather than the outer boundary of what is considered.

This distinction explains why organisations can be fully compliant and yet demonstrably unprepared for real-world failure modes. The convex hull of compliance may be intact, while the surrounding assurance space remains unexplored.

Compliance activities audits, testing, assessments, certifications operate at three distinct levels, each of which contributes differently to assurance reasoning.

First, there are policies, processes, and procedural artefacts. These include documented policies, standards, procedures, plans, and process descriptions that define intent and expected behaviour. Reviewing these artefacts provides evidence about how an organisation intends to manage risk but says little on its own about whether those intentions are realised in practice.

Second, there are outputs of systems and processes. These include logs, test results, metrics, configurations, build artefacts, monitoring data, and assessment findings. These outputs provide evidence about what actually happened or how a system behaves under specific conditions. They are often more probative than policies, but remain bounded by scope, environment, and time.

Third, there are outcomes and certifications. These include audit opinions, certifications, attestations, and formal approvals that result from successful execution of compliance processes. These outcomes provide signals of baseline conformance and comparability, but they abstract away detail and judgement in order to be portable and recognizable.

Each of these layers has value, but they are not interchangeable. Assurance requires reasoning about how intent (policies and processes), execution (system and process outputs), and judgement (outcomes and certifications) relate to one another.

Used correctly, compliance provides structured evidence across these layers within the assurance space. Used incorrectly, any one layer may be treated as a proxy for the others for example, assuming that a certification implies effective operation, or that documented process implies real-world control.

Assurance gives these artefacts meaning by situating them within a coherent argument about trust, behaviour, and risk.

1.5. Assurance Under Change, Novel Threats, and Going Concern

A defining strength of an assurance framework is its ability to reason about novel and emerging threats before they are fully understood, standardised, or widely experienced.

Compliance regimes are necessarily backward-looking. They encode lessons from past incidents, established threat models, and negotiated consensus about acceptable practice. This makes them valuable for baseline control, but poorly suited to anticipating new forms of risk that arise from technological change, scale, or novel combinations of systems.

Assurance, by contrast, provides a framework for reasoning forward. By grounding confidence in principles, explicit claims, and articulated assumptions, assurance allows organisations to ask:

- what would change if this system were scaled, automated, or repurposed,
- which assumptions would fail first under stress or adversarial pressure,
- and where emerging threats might manifest even if no explicit vulnerability or failure is yet observable.

This is particularly important in domains where threats are evolving faster than standards, such as cybersecurity, AI-enabled systems, autonomous technologies, and highly connected socio-technical infrastructures. In these contexts, waiting for threats to be codified into regulation or harmonised standards is often equivalent to waiting until harm has already occurred.

Assurance does not require perfect foresight. It does not predict specific future attacks or failures. Instead, it creates the conditions for early understanding: making dependencies visible, surfacing fragile assumptions, and identifying where confidence is contingent rather than robust. This enables organisations to recognise when their assurance posture is becoming stale, even if formal compliance status remains unchanged.

In this respect, assurance closely parallels the going-concern assumption in accountancy. Financial statements are not prepared on the basis that an organisation is perfect or risk-free; they are prepared on the assumption that the organisation will continue to operate for the foreseeable future. Auditors do not certify that failure is impossible.

They assess whether there is reasonable confidence that the organisation can continue as a going concern, given its structure, controls, risks, and resilience.

Assurance serves a similar function for systems, services, and organisations. It is not a claim that nothing will go wrong, but a reasoned judgement that the entity can continue to operate acceptably in the face of change, stress, uncertainty, and emerging threat.

Change is inevitable: technologies evolve, threats adapt, organisations restructure, supply chains shift, and societal expectations change. An assurance approach that is valid only for a static snapshot a single architecture, a single deployment, or a single audit cycle is fragile by design.

By treating assurance as an ongoing reasoning activity rather than a checklist or a point-in-time verdict, organisations create the conditions for confidence that can be revisited, challenged, and renewed. Like going-concern assessments, assurance must be sensitive to early warning signals, emerging risks, and changes in underlying assumptions, rather than waiting for formal failure.

This perspective reinforces a central theme of this paper: assurance is not about eliminating risk, but about maintaining justified confidence as conditions evolve.

1.6. Control Mapping, Standards Churn, and the Illusion of Progress

A persistent pathology in compliance-dominated environments is the reflexive mapping of controls between regimes.

A new standard is published, a prescriptive regulation comes into force, or a certification scheme is revised. The immediate organisational response is familiar: take the existing control set, map it line-by-line to the new one, identify gaps, and close the minimum number required to restore a compliant state. This activity is often framed as rigorous analysis. In practice, its outputs are almost always ephemeral.

There is thinking involved in control mapping. Teams reason implicitly about equivalence, intent, scope, and overlap. But that reasoning is rarely retained. Once the mapping table exists, the argument that justified it disappears. What remains is a static artefact that answers only one question: are we compliant again?

This behaviour is understandable in a system dominated by compliance incentives. Budgets are constrained. Deadlines are fixed. Regulatory exposure is binary. When the objective is to minimise disruption and cost, the rational strategy is to do the least work necessary to regain compliance. The result is a cycle of standards churn, where effort is expended repeatedly without a corresponding increase in understanding.

From an assurance perspective, this is a lost opportunity.

If compliance frameworks are to be leveraged at all, they should be mapped to principles, not to each other. Controls exist because, at some point, someone believed they supported a more fundamental property: confidentiality, integrity, availability, safety, resilience, accountability. Mapping controls directly between regimes collapses that reasoning into surface similarity. Mapping controls to principles preserves intent.

A principles-first approach changes the nature of the work:

- Controls from different standards that support the same underlying principle can be grouped explicitly.
- Apparent gaps can be examined as missing arguments, not merely missing controls.
- Redundancy becomes visible, not as waste, but as defence-in-depth supporting a claim.
- Novel requirements can be assessed by asking whether they introduce a genuinely new principle, or simply a new instantiation of an existing one.

Crucially, this mapping is then retained as argumentation. Instead of a control-to-control crosswalk that is discarded once compliance is achieved, the organisation retains a structured explanation of why its existing activities support particular assurance claims, and where they do not.

In this model, compliance mapping becomes a by-product of assurance reasoning, not its driver. Standards and regulations are treated as sources of evidence and structure within an enduring argument, rather than as targets to be hit and forgotten. This allows organisations to respond to regulatory change without repeatedly re-deriving understanding and without mistaking the restoration of compliance for the restoration of confidence.

Forward reference. The purpose of this mapping is not the map itself. In Claims, Arguments, and Evidence, compliance artefacts only contribute to assurance when they ultimately terminate in explicit **claims**, supported by **arguments** and **evidence**. Mapping that does not resolve into claims and argumentation merely relocates compliance effort; it does not increase justified confidence.

2. A Structural Model for Assurance

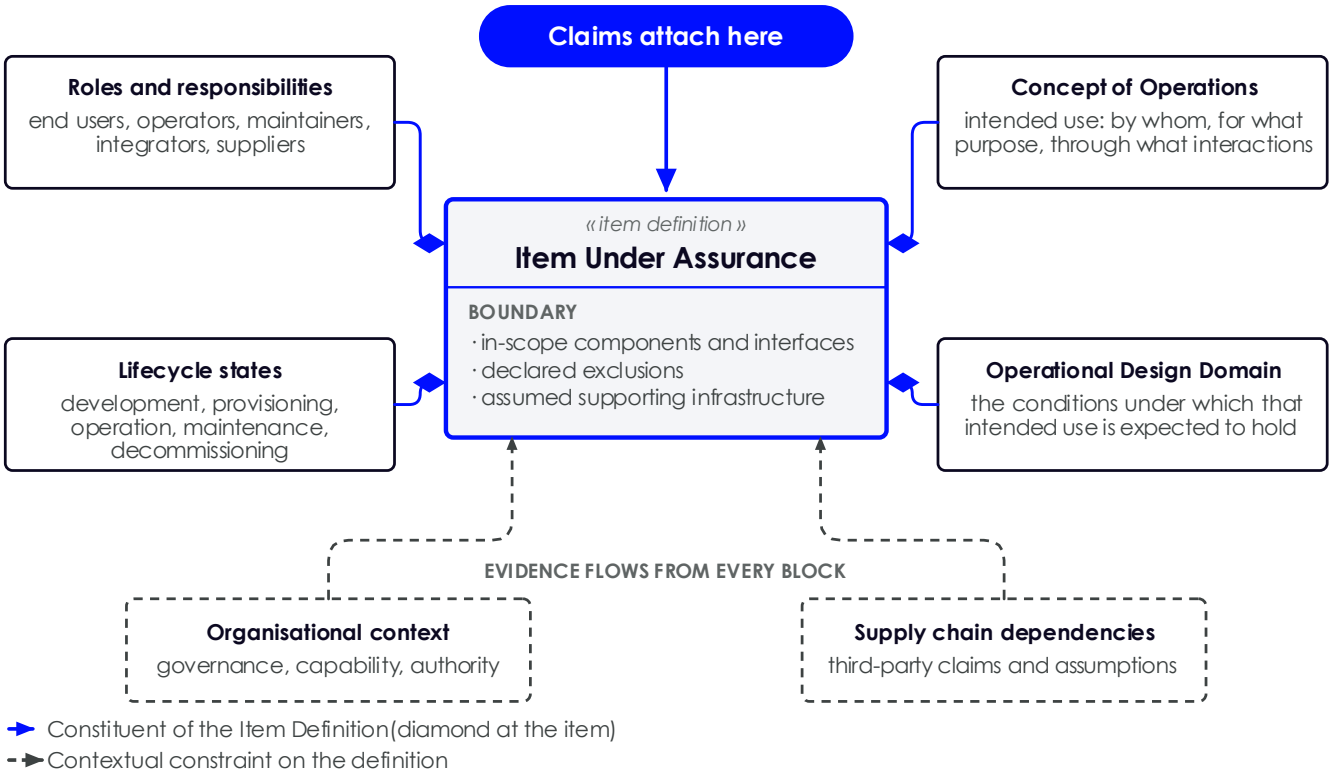
If Assurance as a Space for Reasoning, establishes what assurance is, this section clarifies what assurance must reason about.

Assurance does not operate in the abstract. It must account for concrete systems, real organisations, and critically the interfaces between them. Many assurance failures occur not because individual elements are poorly designed or governed, but because their interactions are insufficiently reasoned about.

2.1. Product / System Assurance: Defining the Item Under Assurance

Figure 2 The Item Under Assurance as a system

An Item Definition integrates every associated element . Claims are asserted of the item, never of its parts alone.



An effective **Item Definition** is the foundation of product and system assurance. It establishes what is being reasoned about, under what conditions, and with which assumptions. Without a clear item definition, assurance claims are necessarily ambiguous.

Crucially, the Item Definition is not limited to a list of components or boundaries. It must integrate roles, lifecycle, and operational context including the Concept of Operations (ConOps) and Operational Design Domain (ODD) into a coherent description of the object of assurance.

Product and system assurance is frequently underestimated in its complexity. Modern products and services are rarely static artefacts; they are evolving socio-technical constructs that exist across multiple environments, lifecycle stages, and usage contexts.

Assurance therefore begins with a deceptively difficult task: **defining the item under assurance**.

This requires explicit answers to questions such as:

- Is the item a product, a deployed system, a service, or a system-of-systems?
- Which components, interfaces, and dependencies are in scope?
- What assumptions are made about the operational environment and supporting infrastructure?
- Which lifecycle phases are included or excluded?

Ambiguity at this stage propagates forward. Poor item definition leads to vague claims, fragile arguments, and evidence that appears sufficient until challenged.

2.1.1. Roles, Responsibilities, and Relationships as Part of Item Definition

An effective Item Definition must explicitly capture the **roles, responsibilities, and relationships** associated with the product, system, or service.

A single item typically involves multiple actors with materially different relationships to it, including:

- end users,
- operators,
- maintainers and integrators,
- manufacturers and suppliers.

Each role interacts with the item differently, under different privileges, incentives, and threat models. These relationships shape how functionality is accessed, how failures propagate, and how misuse or harm can arise.

Roles and responsibilities are therefore not contextual background; they are intrinsic properties of the item under assurance. Many risks emerge precisely at the interfaces between roles for example, between operator and maintainer, or supplier and integrator rather than within any single perspective.

If roles and relationships are left implicit, assurance claims become role-blind. Evidence may demonstrate that the item is acceptable for one class of actor while remaining unsafe, insecure, or misuse-prone for another.

Explicitly defining roles and relationships as part of the Item Definition is therefore essential for credible, multi-perspective assurance.

2.1.2. Lifecycle States and Shifting Threat Models

Products and systems move through multiple lifecycle states, including development, deployment, operation, maintenance, and decommissioning. Each state introduces different attack surfaces, failure modes, and acceptable behaviours.

Controls appropriate in one state may be dangerous in another. Diagnostic access essential during maintenance may be unacceptable during live operation. Assurance must therefore be explicit about which lifecycle states are in scope, and how transitions between them are managed.

In addition, modern regulation and real-world risk increasingly demand a feedback loop between deployment reality and product development.

A developer or manufacturer's assurance case is rarely invalidated by design changes alone. It is more commonly undermined by contextual drift: how the product is actually used, integrated, scaled, or repurposed once it leaves the intended operating environment described in the Item Definition (including ConOps and ODD).

This creates a through-life obligation that is both practical and epistemological: organisations must remain aware of whether their assurance assumptions continue to hold. That awareness may come from incident data, vulnerability reports, customer deployments, telemetry, integrator feedback, regulatory changes, or observed misuse patterns.

Where those signals indicate that products are being used outside assumed conditions or that emergent threats are materialising at scale the appropriate response is not simply to update documentation, but to update the assurance claims themselves.

For product development lifecycles, this implies traceability across versions: showing how new real-world information led to revised claims, revised arguments, and therefore revised design choices. In other words, secure by design becomes an explicit, auditable feedback mechanism:

- operational reality informs assurance reasoning,
- assurance reasoning updates claims,
- updated claims drive design and control changes,
- and those changes produce new evidence in subsequent releases.

This traceability is increasingly important in regimes that emphasise secure-by-design and through-life support. It allows organisations to demonstrate not only that they shipped a compliant product at time of release, but that they sustain justified confidence as real-world use evolves.

2.1.3. Concept of Operations and Operational Design Domain as Part of Item Definition

The Concept of Operations (ConOps) and Operational Design Domain (ODD) are not ancillary artefacts. They are core elements of an effective Item Definition.

The ConOps describes how the item is intended to be used: by whom, for what purpose, with what responsibilities, and through what interactions between people and technology.

The ODD defines the conditions under which that intended use is expected to be safe, secure, and acceptable. This may include environmental conditions, operational modes, user competence, connectivity assumptions, workload, scale, and interaction with other systems.

Together, ConOps and ODD bound the assurance problem space. They define what constitutes appropriate use, foreseeable misuse, and operation outside the defined design assumptions. Where these are implicit or incomplete, assurance claims become unstable and are easily invalidated by deployment, scaling, or repurposing.

Making ConOps and ODD explicit as part of the Item Definition is therefore a prerequisite for credible assurance, not a refinement step.

2.2. Organisational Assurance: Capability Over Time

Organisational assurance is often perceived as simpler because it is mediated through familiar artefacts such as policies, procedures, and audit cycles. In reality, modern organisations are complex systems in their own right.

Organisations operate across multiple teams, geographies, suppliers, and regulatory regimes. Responsibility, authority, and knowledge are distributed rather than centralised.

2.2.1. Capability Versus Documentation

At an organisational level, assurance is not about the existence of documented processes but about sustained **capability**.

Capability includes the ability to:

- make informed design and risk decisions,
- manage change without eroding control,
- detect and respond to incidents,
- and learn from failure.

Documentation provides evidence of intent. Capability is demonstrated through behaviour, particularly under stress.

2.2.2. Supply Chain and Dependency Interfaces

Most modern products and services depend on complex supply chains, platform providers, integrators, and third-party components. As a result, assurance claims rarely sit entirely within the control of a single organisation.

Confidence in a system therefore rests on assumptions about the behaviour, capability, and processes of suppliers and partners. These assumptions are often weakly evidenced, displaced into contractual language, or reduced to the presence of third-party certificates.

From an assurance perspective, this is inadequate. A certificate may demonstrate that a supplier's management system is compliant with a standard, but it does not demonstrate that a specific product, variant, or integration has been developed, configured, and assessed in a way that supports the claims being made by the risk-holding organisation.

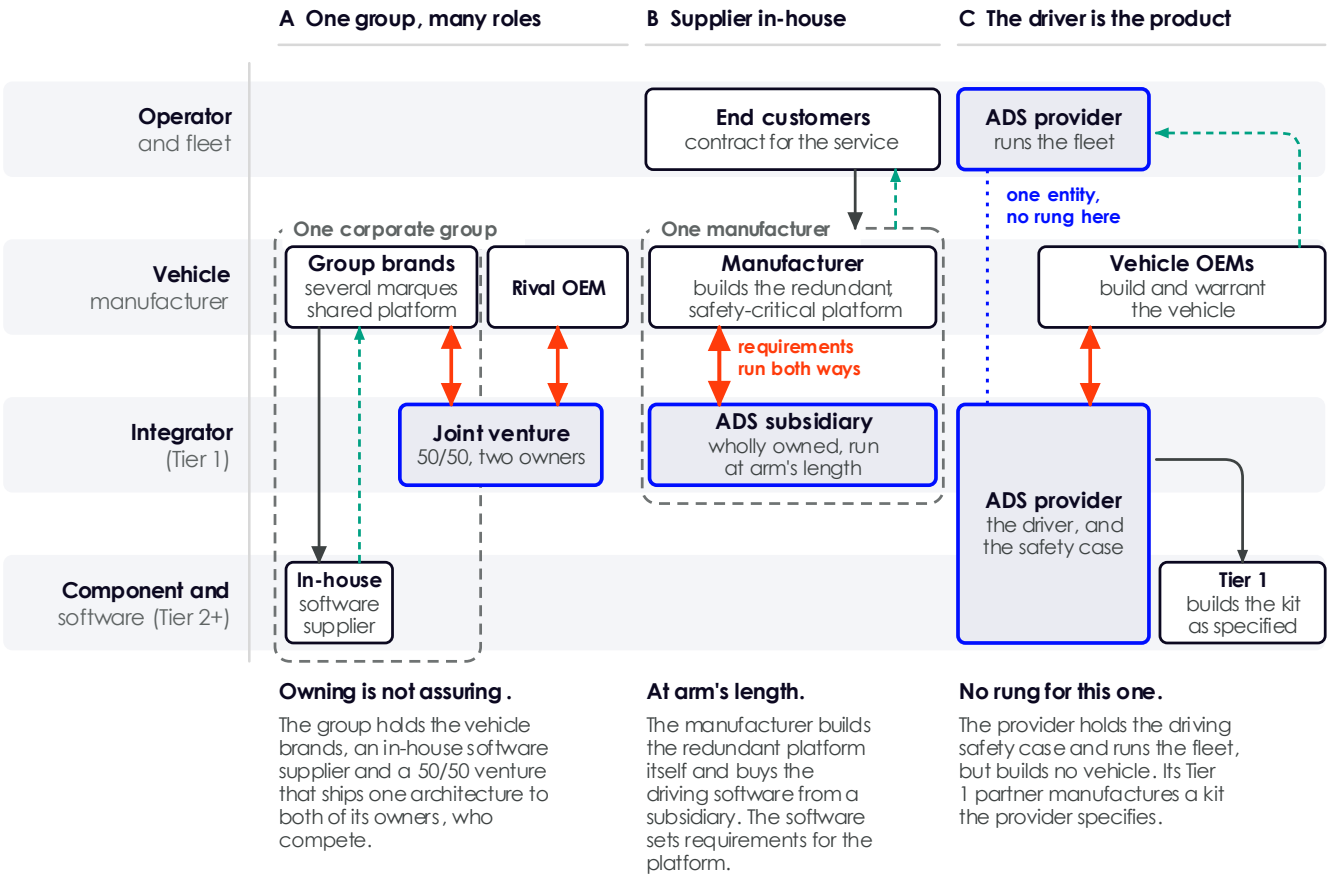
Principles-based assurance requires that such dependencies are made explicit, bounded, and reflected in the scope of claims being made. This includes clarity about what is being relied upon from suppliers, what assumptions are being made about their processes and artefacts, and where responsibility for residual risk ultimately sits.

Locating that responsibility is harder than the language of tiers suggests. A tier number describes a position in a flow of work rather than an organisation, and real chains rarely place one organisation in one position. A group may hold the vehicle brands, an in-house software supplier and a jointly owned venture whose architecture is fitted by two competing owners. A manufacturer may build the platform itself while buying the driving software from a subsidiary it deliberately runs at arm's length. A developer of an automated driving system may hold the safety case for the driving task and operate the vehicles commercially without building a vehicle at all.

In arrangements of this kind the requirements do not simply descend. The organisation nearest the top is constrained by design objectives set from below, and the same interface carries obligations in both directions. Assurance interfaces are therefore better identified from the claims being made and the evidence that supports them than from the position an organisation occupies in a supply chain diagram. Three such arrangements are set out in Figure 3.

Figure 3 Positions in the chain are not organisations

A tier number says where work sits , not who is accountable for it. Real chains put one organisation in several positions at once , place partners on both sides of the same interface, and let requirements run upward as well as down .



Each panel is a real arrangement in current automotive practice , drawn generically .

2.2.3. Organisational Drift and Change

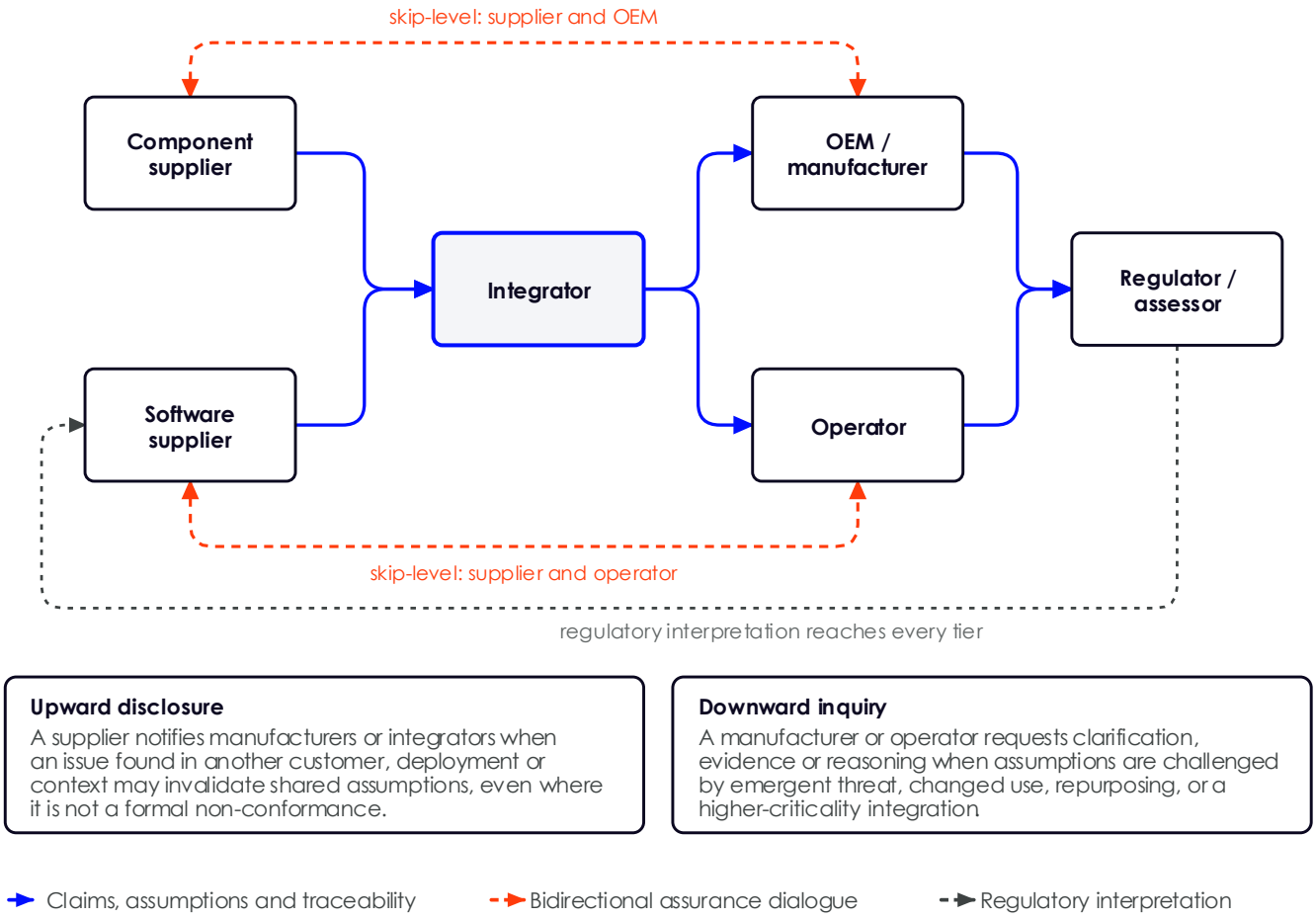
Over time, organisations change: people move roles, priorities shift, and processes adapt. This leads to organisational drift, where actual practice diverges from documented intent.

Assurance must therefore consider not only whether controls exist, but whether the organisation can detect and correct drift as conditions change. In this respect, organisational assurance closely mirrors the going-concern concept introduced in Assurance Under Change, Novel Threats, and Going Concern.

2.2.4. Assurance Interface Agreements

Figure 4 The assurance interface is a graph, not a chain

Assurance-relevant reasoning flows both ways and skips levels. Accountability for risk stays exactly where it sits.



Where assurance depends on suppliers, integrators, or platform providers, there is a need for a more explicit construct than traditional compliance-focused interface agreements.

An Assurance Interface Agreement (AIA) can be understood as an evolution of the Cybersecurity Interface Agreement (CIA) concept, extending it beyond the exchange of final artefacts toward the exchange of assurance-relevant reasoning.

In practice, many CIAs focus narrowly on the delivery of compliance work products, such as an Item Definition, a Threat Analysis and Risk Assessment (TARA), or a third-party certificate asserting conformance to standards such as ISO/SAE 21434 [2]. While these artefacts may be necessary, they are rarely sufficient to support meaningful assurance.

Common failure modes include:

- Item Definitions that are generic, incomplete, or misaligned with the actual deployment context.
- TARAs that reflect an off-the-shelf product rather than a tailored or integrated variant.
- Reliance on third-party certificates as proxies for product-specific assurance.

From the perspective of the risk-holding manufacturer or operator, these artefacts do not provide the means to reason about *how* assurance was achieved. They obscure critical questions such as:

- How does the supplied variant differ from the off-the-shelf product?
- Which assumptions in the original threat model no longer hold?
- What parts of the supplier's process were exercised for this specific integration?
- Which claims are being relied upon, and which are no longer valid?

An Assurance Interface Agreement addresses this gap by focusing on **claims, assumptions, and traceability**, rather than on artefact delivery alone. At a minimum, it should make explicit:

- Which assurance claims the supplier is asserting, and in what context.
- Which assumptions underpin those claims, including intended use, scale, and operational environment.
- How deviations, tailoring, or integration affect the original threat model and assurance reasoning.
- What evidence exists for the specific product, configuration, or variant being supplied.
- How changes, vulnerabilities, or incidents will be communicated and fed back into the assurance case.

The purpose of an AIA is not to shift risk away from the manufacturer or operator; risk ownership remains unchanged. Instead, it enables the risk holder to reason about supplier contributions to assurance with clarity and confidence, and to integrate those contributions coherently into a system-level assurance case.

Critically, an Assurance Interface Agreement must also enable bidirectional information flow in response to events that impact assurance.

Where an event, vulnerability, incident, or emerging threat affects confidence in a claim, the AIA should provide an explicit mechanism for:

- **Upward disclosure**: enabling a supplier to notify manufacturers or integrators when an issue is identified in another customer, deployment, or context that may invalidate shared assumptions, even where the issue does not constitute a formal non-conformance or reportable incident.
- **Downward inquiry**: enabling manufacturers or operators to request additional clarification, evidence, or reasoning from suppliers when assurance assumptions are challenged, for example, due to emergent threats, changes in end-user behaviour, repurposing of the product, or integration into a higher-criticality environment.

This exchange may not always result in design change or remediation. In many cases, the objective is to **inform business and risk decisions**, update assurance reasoning, or consciously accept revised residual risk. What matters is that the request for information, and the response to it, are recognised as legitimate assurance activities rather than exceptional or adversarial actions.

By explicitly supporting bidirectional assurance dialogue, an AIA helps prevent a common failure mode in supply chains: where suppliers treat assurance as a one-time delivery obligation, and manufacturers discover too late that critical assumptions have been invalidated elsewhere. Instead, assurance becomes a shared, through-life reasoning activity across organisational boundaries, even while accountability for risk remains clearly assigned.

A practical way to assess whether an interface agreement is **fit for purpose** is to ask a simple question:

If something materially changes in the assurance landscape: an incident in another customer deployment, an emergent threat, a change in scale or use, or new regulatory interpretation, does this agreement enable proactive, timely support to reason about the impact on our claims?

If the answer is no, the agreement is functioning as a compliance interface rather than an assurance interface. It may satisfy contractual or audit requirements, but it will not support justified confidence when conditions change. An Assurance Interface Agreement exists precisely to ensure that when the unexpected happens, the parties involved can reason together about what has changed, what still holds, and what decisions now need to be made.

By treating supplier interfaces as assurance interfaces rather than compliance boundaries, organisations can avoid a common and costly illusion: that conformance to a standard somewhere in the supply chain is equivalent to assurance for a specific product used in a specific context.

2.3. Compliance Frameworks as Structural Artefacts

Compliance frameworks should be understood as structural artefacts within this broader assurance model.

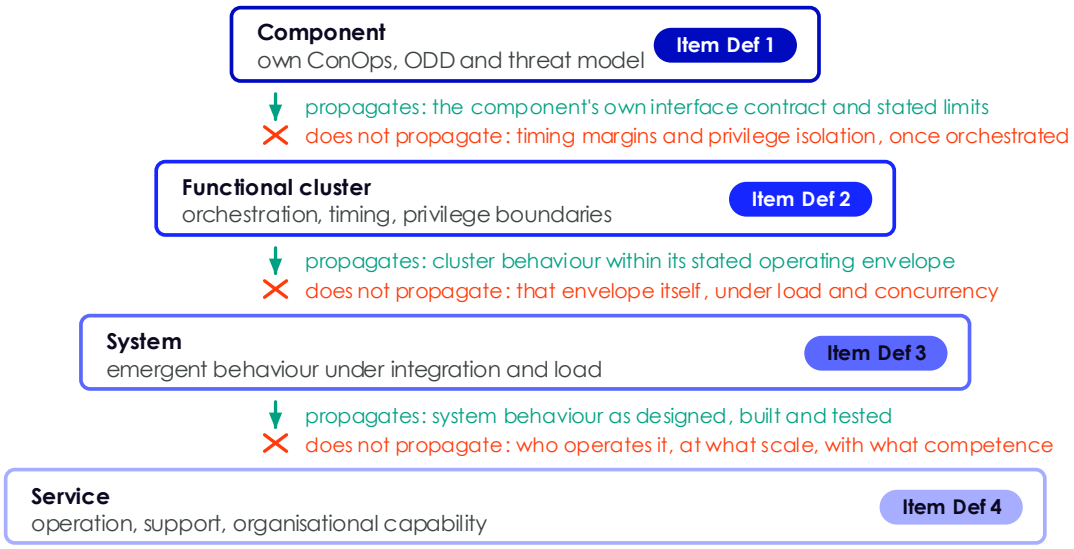
Some frameworks primarily address product or system behaviour; others focus on organisational processes. Still others sit at the interface. None of them, individually or collectively, fully bound the assurance problem space.

Assurance reasoning is required precisely to understand what falls between frameworks, how their assumptions interact, and where residual risk remains.

This structural model sets up the next problem: how different assurance lenses can legitimately reach conflicting conclusions even when systems are formally compliant.

Figure 5 Every abstraction layer needs its own Item Definition

Component definitions are inputs to the layer above , never a substitute for it. Only what stays valid in the new context propagates .



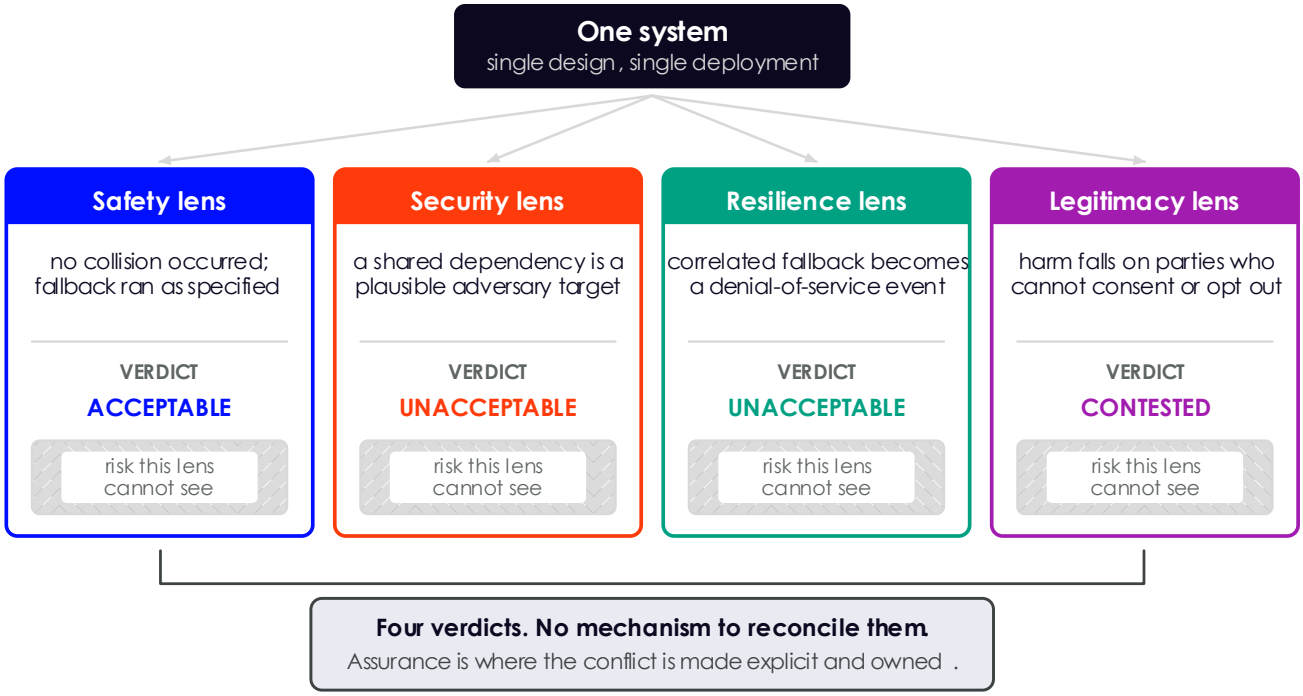
The sum of component Item Definitions is not an Item Definition for the system

Just as a set of individually safe components does not compose into a safe system, a set of component-level definitions does not describe the behaviour, risks or assumptions of the layer above. Treating them as additive encourages over-claiming and masks emergent risk.

3. The Problem of Conflicting Lenses

Figure 6 The same system through four lenses

Each lens is valid and none is complete. Compliance rarely requires the contradictions between them to be reconciled.



Modern systems are assessed through multiple specialist lenses: safety, security, resilience, privacy, usability. Each lens is valid, but none is complete.

In systems-of-systems, these lenses often produce contradictory conclusions. A design choice that appears safe through one lens may introduce unacceptable risk through another. Compliance regimes rarely require these conflicts to be reconciled.

3.1. Case Study: Autonomous Vehicles and Urban Blackout

On 20 December 2025, a fire at a Pacific Gas and Electric substation in San Francisco's Mission district cut power to roughly 130,000 customers, about a third of the city [3]. It was not a cyberattack and not malicious interference; the cause was formally undetermined. Traffic signals went dark across a wide area.

From a safety perspective, the individual vehicles behaved correctly. Waymo's autonomous fleet treats a dark signal as a four-way stop, degrading gracefully exactly as designed. The binding constraint lay elsewhere. Each such encounter triggers a confirmation step in which a remote-assistance operator authorises the vehicle to proceed. That fallback is sized for the normal rate of ambiguous encounters, not for a city-wide simultaneous spike. With a few dozen operators against thousands of newly ambiguous junctions, the queue backlogged [4].

The consequences were operational rather than collisional. Waymo recorded 1,593 stalls of two minutes or longer; 64 vehicles required staff dispatch or towing; the company suspended service some five hours into the outage. San Francisco's emergency dispatch centre called Waymo 31 times, with one dispatcher held for 53 minutes [5]. Fire service witnesses later testified that vehicles encroached on emergency scenes, and fire appliances and transit buses were obstructed. Waymo subsequently acknowledged that its response did not meet its own standards [4], and cited degraded local connectivity as a contributing factor.

This is the important structural point, and it is sharper than a simple lens conflict. Through the safety lens the design worked: no collisions occurred, and each vehicle executed its specified minimum-risk behaviour. Through a

resilience lens the fleet became a city-scale obstruction. But the failure was not merely that two lenses disagreed. The mitigation itself failed in common cause: a human fallback shared across the entire fleet was saturated by precisely the event it existed to handle. Correlated demand on a shared recovery mechanism is invisible to any analysis that reasons about one vehicle at a time. Note also that “behaved exactly as designed” holds at the level of the vehicle, not of the service, a distinction the operator itself conceded.

This distinction is important, but it is not reassuring. The failure mode emerged in the absence of malicious intent, and a mechanism that fires without an adversary is a mechanism an adversary can fire. Once such systems operate at city scale, the same dependencies that failed benignly become strategically relevant infrastructure. Power distribution, traffic control systems, and mobile communications are not neutral backdrops; they are part of the operational fabric on which autonomous systems rely.

At this scale, it is no longer reasonable to assume that these dependencies will fail only accidentally. Infrastructure that can immobilise transport networks through indirect effects would, in other contexts, be considered a plausible target for nation states or advanced persistent threat (APT) actors. The assurance challenge therefore exists regardless of adversarial intent: benign failure is sufficient to cause harm, and malicious exploitation would only lower the bar.

Crucially, this was not a failure of compliance. Each component behaved in accordance with its specifications and regulatory expectations. It was a failure of assurance across lenses, a failure to reason explicitly about scale, coupling, and dependency on shared infrastructure.

The case illustrates a class of risk that is increasingly common in modern systems: malicious or harmful use of intended functionality, including scenarios where no attacker is present. A difference in scale became a difference in class, transforming correct safety behaviour into a system-level failure with societal impact.

3.2. What Reconciliation Requires

Conflict between lenses is not a defect to be engineered away. Each lens asks a different question about the same item, and a design that satisfies all of them without tension is often one whose claims are too weak to be interesting. The failure is not that specialists disagree. It is that the disagreement is rarely brought to a point at which someone has to decide.

Reconciliation begins with a shared object. Competing positions must be expressed as claims about the same defined item, under the same Concept of Operations and Operational Design Domain, as set out in Product / System Assurance: Defining the Item Under Assurance. Until that is done the lenses are not genuinely in conflict; they are describing different objects and talking past one another. A good deal of what presents as an irreconcilable difference between safety and security dissolves at this step, because the two assessments turn out to have assumed different boundaries, different lifecycle states, or different populations of user.

What remains after that step is a real trade-off, and it can be argued rather than settled by seniority. This is the work that structured assurance reasoning supports, as set out in Claims, Arguments, and Evidence. Expressed as claims, “the system is safe” and “the system is resilient” can both be asserted with their assumptions, exclusions and operating conditions visible. Where the two cannot both hold, the incompatibility is recorded as bounded residual risk rather than resolved by whichever lens carries the most regulatory weight.

Reconciliation also requires an owner. A trade-off between lenses that belongs to no one will be rediscovered at the next incident, which is why claim ownership, discussed in Building Blocks as an Aid, Not an Overhead, matters here as much as the notation does. Compliance rarely forces any of this, because each framework is scoped to a single lens: a safety standard does not ask whether the safe behaviour remains resilient when every vehicle performs it at once, and a security standard does not ask whether the secure configuration is operable under load. The space between them is nobody's deliverable. It is the same structural point Figure 1 makes about the hollows the band encloses, seen from the position of the specialists standing on either side.

4. Techno-Social Systems and the Meaning of Safety

The limitations of compliance are even more pronounced in socio-technical systems, where outcomes depend on human interpretation, power dynamics, and contested values.

Such systems have a recurring failure mode that does not fit neatly into safety, security, or misuse categories, and it is central to how socio-technical harm actually manifests. It is defined and examined in Malicious Use of Intended Functionality (MUIF).

Consider the claim:

"Body-worn cameras are used to keep everyone safe."

This claim is easy to make compliant. Hardware can be certified, data protected, policies approved. Assurance, however, requires that we argue what safe means and for whom.

4.1. Safety Is Not Singular

Safety encompasses physical harm, psychological impact, dignity, and social legitimacy. Evidence for physical safety effects is difficult and often inconclusive. Evidence for psychological and social impacts is rarer still.

In the absence of strong evidence, organisations often substitute capability for outcome: the presence of a tool becomes proof of benefit.

This substitution becomes especially problematic when the same capability can be deliberately leveraged to cause harm. Intended functionality, deployed without careful reasoning about scale, aggregation, and incentive alignment, becomes a vector for malicious use even while remaining compliant.

4.2. Stakeholders and Trade-offs

Different stakeholders experience safety differently:

- Wearers may gain physical protection.
- Subjects may experience loss of privacy or intimidation.
- Bystanders may be subjected to incidental surveillance.
- Society absorbs long-term effects on trust and consent.

The mechanism that protects one group can harm another. Compliance frameworks rarely require these trade-offs to be articulated. Assurance does.

4.3. Malicious Use of Intended Functionality (MUIF)

A recurring failure mode in modern systems is harm arising not from malfunction or unauthorised access, but from the deliberate leveraging of functionality that operates exactly as intended. This is referred to here as malicious use of intended functionality (MUIF).

MUIF describes scenarios where systems behave correctly, remain compliant, and satisfy design requirements, yet are exploited to produce unacceptable outcomes once scale, incentives, or context change. These risks often fall between safety, security, and misuse frameworks, and are therefore poorly addressed by compliance-led assurance.

The word *deliberate* does the work here, and it attaches to the initiating condition rather than to every step that follows. An actor who induces a condition whose consequences are foreseeable has acted deliberately, even where the harmful behaviour is the system operating exactly as designed and no component is subverted. Arson is the familiar case: the arsonist does not control combustion, only the ignition, and the building catching is the expected consequence rather than an accident.

This is why a failure mode that first appears benignly still belongs in this category. If correlated but entirely legitimate behaviour can produce an unacceptable outcome with no one intending it, the same outcome is available to someone who does intend it, and the initiating condition becomes the thing assurance has to reason about. The absence of an adversary in a given incident is evidence about that incident, not about the mechanism.

4.3.1. MUIF in Privacy and Pseudonymous Data

The privacy domain provides a clear illustration of MUIF.

Many systems are explicitly designed to release or share pseudonymous or aggregated data under the assumption that individual harm is mitigated once direct identifiers are removed. Such designs are often compliant with data protection requirements, supported by privacy impact assessments, and accompanied by assurances that the data is “non-identifiable”.

This reasoning draws directly on well-established GDPR principles [6]: lawfulness, fairness and transparency; purpose limitation; data minimisation; accuracy; storage limitation; integrity and confidentiality; and accountability. These principles already embody decades of thinking about proportionality, harm, and trust in information systems.

However, the intended functionality of releasing pseudonymous data can itself be leveraged to cause harm when combined with auxiliary information, repeated queries, or external datasets. The system operates exactly as designed; the harm arises from inference rather than breach.

Healthcare datasets provide well-known examples. Even when direct identifiers are removed, combinations of attributes such as age, location, admission time, and treatment type can be sufficient to re-identify individuals or infer sensitive information. In small populations or specialised contexts, the ability to infer that someone in a small group has undergone a particular treatment may itself be harmful, even without full re-identification.

This is not a failure of access control or data leakage. It is the malicious or harmful exploitation of intended data release, often through correlation and aggregation over time. A difference in query volume or auxiliary knowledge becomes a difference in class.

Techniques such as differential privacy exist precisely because traditional anonymisation is insufficient at scale. Differential privacy does not eliminate risk; it makes explicit, measurable trade-offs between data utility and disclosure risk, acknowledging that privacy loss accumulates with repeated use.

From an assurance perspective, claims such as “the data is anonymised” are too coarse to justify confidence. Meaningful claims must specify how GDPR principles are interpreted in context, what assumptions are made about adversarial knowledge and scale, and how residual privacy risk is bounded and monitored.

This example reinforces a broader point: regulatory principles already provide a strong foundation for assurance reasoning. What is often missing is not principle, but explicit claims and arguments about how those principles hold under scale, aggregation, and adversarial use.

4.3.2. Insider Threat as Malicious Use of Intended Functionality

The **insider threat** provides another clear and often under-analysed example of malicious use of intended functionality, particularly when viewed through the lens of roles, responsibilities, and capabilities defined as part of the Item Definition.

In many products and services, certain roles, such as servicing, warranty support, customer operations, incident response, or platform administration, are intentionally granted **elevated privileges** relative to ordinary users. These privileges may include access to aggregated datasets, the ability to execute bulk queries, override controls, modify configurations, or access diagnostic interfaces that are deliberately restricted in normal operation.

From a compliance or vulnerability-centric perspective, this is not a defect. It is a **feature**, introduced to enable legitimate business functions such as fault resolution, fraud investigation, customer support, safety intervention, or regulatory reporting. The system is behaving exactly as designed.

However, from an assurance perspective, this design choice introduces a distinct class of risk that sits uncomfortably between traditional categories:

- It is not an external attack.
- It is not a software vulnerability.
- It is not unauthorised access.

Instead, it is the risk that authorised roles, exercising legitimate functionality, can cause harm when intent, incentives, oversight, or context change.

This is often described informally as a “business logic failure”. The logic of the system is correct for its intended purpose but insufficiently reasoned about when placed in a broader socio-technical context. The assurance gap arises not from the existence of the capability, but from the assumptions about trust that surround the entity holding the role and the tokens, credentials, or permissions that embody that trust.

Common contributing factors include:

- Roles whose access scope is significantly broader than any single user or asset.
- Credentials or tokens protected by weaker controls than those applied to high-impact user actions.
- Legitimate workflows that permit population-scale queries or actions with minimal friction.
- Oversight mechanisms focused on misuse detection rather than on the *reasonableness* of use.

Crucially, these risks are rarely visible at the component or control level. They emerge at the interface between:

- role definition,
- organisational process,
- technical enforcement,
- and real-world incentives and pressures.

This is why insider threat cannot be treated solely as a human resources issue, a monitoring problem, or a compliance checkbox. It is an **assurance problem**, rooted in how the Item Definition captures roles and relationships, how ConOps describes legitimate use of privileged functions, and how the Operational Design Domain bounds acceptable scale, frequency, and aggregation of actions.

An assurance-led approach therefore requires explicit claims such as:

- which roles are permitted to exercise population-scale or cross-context functionality,
- under what conditions such use is considered reasonable and proportionate,
- how misuse or drift would be detected and challenged,
- and how trust in those roles is justified, limited, and revisited over time.

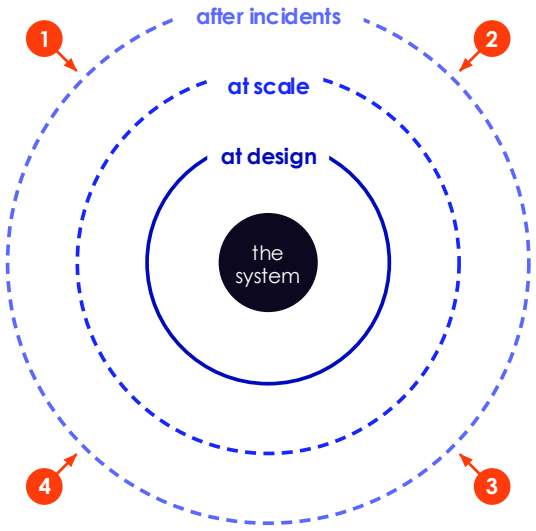
Without such reasoning, organisations may remain fully compliant while carrying a latent risk surface that sits just outside the traditional security boundary. The system remains secure in the narrow sense, yet fragile in the face of insider misuse that is entirely consistent with its intended design.

This example further reinforces a central theme of this paper: many of the most serious assurance failures arise not from broken controls, but from **unexamined trust relationships** embedded in roles, privileges, and business logic that operate correctly until context, scale, or intent shifts.

5. Reasonable Foreseeability

Figure 7 Reasonable foreseeability moves outward

What an organisation can be expected to have anticipated expands with scale, integration, incident history and learning.



WHAT MOVES THE BOUNDARY OUTWARD

- greater scale and exposure
- deeper integration and coupling
- incidents in adjacent domains
- accumulated operational learning

HARMS CROSSING INTO FORESEEABILITY

1. **Misuse at population scale**
   individually authorised actions, aggregated
2. **Insider use of legitimate privilege**
   a feature, exercised by the role it was built for
3. **Correlated fallback under shared outage**
   safe behaviour, triggered en masse
4. **Harm to parties who cannot consent**
   bystanders with no route to refuse

WHAT THE BOUNDARY IS NOT

A prediction.
Assurance does not forecast specific attacks or failures.

A shield.
"It was outside our threat model" rarely survives scrutiny once harm occurs

Fixed.
What was obscure at inception becomes obvious as exposure grows.

Single-perspective.
Technically unlikely and socially obvious are different tests.

> The test is whether you can explain , credibly and transparently, why a harm was or was not considered, and what would change that judgement.

Reasonable foreseeability defines the practical boundary of assurance. It answers the question of which harms, misuse cases, and failure modes an organisation can legitimately be expected to have considered, reasoned about, and addressed.

Foreseeability is unavoidable. No system can be assured against every conceivable outcome. At the same time, foreseeability is frequently misunderstood as a purely technical or probabilistic concept. In practice, it is social, contextual, and retrospective.

5.1. Foreseeability as a Boundary, Not a Shield

In theory, reasonable foreseeability provides a pragmatic boundary. It recognises that designers and operators cannot anticipate every novel misuse or emergent interaction, and that assurance must be proportionate to available knowledge, capability, and context.

In practice, foreseeability is often misused as a **defensive shield** rather than a reasoning boundary. Arguments such as "this scenario was not identified" or "this was outside the threat model" may appear technically valid, but they rarely survive external scrutiny once harm has occurred.

Foreseeability is judged not by whether a specific scenario appeared in a hazard log or threat register, but by whether it *should have been anticipated* given:

- the nature of the system,
- the scale and criticality of its deployment,
- the incentives and capabilities of plausible actors,
- and prior analogous failures in adjacent domains.

Assurance therefore requires that foreseeability is argued explicitly, not assumed implicitly. Claims must be bounded by clear statements of what was considered foreseeable, what was excluded, and why those exclusions were reasonable at the time.

5.2. Foreseeability Is Contextual and Role-Dependent

What is reasonably foreseeable depends strongly on perspective.

Engineers may focus on component failure modes and technical vulnerabilities. Security practitioners may focus on attacker capability and intent. Operators may focus on procedural deviations and human error. End users and the public, however, often reason about foreseeability through lived experience and intuition.

These perspectives do not align neatly. A scenario that appears technically unlikely may be socially obvious. Conversely, a scenario that is salient to engineers may be invisible to users.

Foreseeability must therefore be reasoned about across roles defined in the Item Definition. Assurance that reflects only one perspective is fragile.

This role-dependence also applies over the lifecycle. What is foreseeable during design may differ from what is foreseeable during operation, scaling, or repurposing. As systems accumulate users, dependencies, and exposure, new misuse pathways become obvious in hindsight even if they were obscure at inception.

An assurance approach that does not revisit foreseeability as context evolves accumulates hidden risk.

5.3. Foreseeability, Scale, and the Malicious Use of Intended Functionality

Foreseeability interacts with the failure mode defined in Malicious Use of Intended Functionality (MUIF) in a way that narrow analysis routinely misses.

Individually, such actions may be compliant, authorised, and well within design assumptions. When exercised at scale, or combined strategically, they can produce qualitatively different outcomes.

This is where differences in scale become differences in class. Rate limits that are safe for one user become attack vectors when multiplied across thousands. Safety mechanisms that work for isolated incidents become denial-of-service mechanisms when triggered en-masse.

Foreseeability arguments that focus narrowly on component behaviour or individual actions fail to capture these emergent effects. Assurance must consider aggregation, automation, and incentive alignment, particularly where functionality is exposed through APIs, remote access, or automated control.

Malicious use of intended functionality sits at the boundary between safety, security and misuse, and it is worth being exact about where that boundary runs. Functional safety addresses hazards arising from malfunctioning behaviour [7]. The safety of the intended functionality addresses hazards that arise with no malfunction at all, from functional insufficiencies and from reasonably foreseeable misuse [8]. Cybersecurity threat models reach the actor, but generally assume the system has been made to do something it should not. None of the three is aimed squarely at an actor who induces an initiating condition so that correct behaviour, at scale, produces the harm. Treating foreseeability seriously requires acknowledging these gaps and reasoning about them explicitly, even where mitigation options are limited or imperfect.

5.4. Case Study: Smart Motorways, When Compliance Survives and Assurance Fails

This case illustrates how narrow item definition, implicit ODD assumptions, and role-blind foreseeability reasoning can coexist with formal compliance while undermining real-world safety and trust.

UK Smart Motorways provide a clear illustration of how compliance can coexist with a failure of assurance, particularly when product/system behaviour, organisational decision-making, and public perception interact.

Smart motorways were designed to increase road capacity by converting the hard shoulder into a live running lane and managing traffic dynamically through variable speed limits, signage, and monitoring. From a regulatory and engineering perspective, the programme complied with applicable highway standards [9] and safety assessments at the time of deployment.

However, the assurance failure did not stem from a single technical defect or procedural lapse. It emerged from a combination of assumptions that were individually reasonable, yet collectively fragile.

First, the **item definition** was implicitly narrow. The system was treated primarily as a traffic management and control system, rather than as a socio-technical system involving stranded drivers, emergency responders, maintenance crews, and the public. The lived experience of a driver stopped in a live lane was not central to the assurance narrative.

Second, the Concept of Operations and Operational Design Domain relied on assumptions about detection and response that were not met at deployment. The configuration at issue is All Lane Running (ALR), in which the hard shoulder becomes a permanent running lane. Confidence rested on stopped vehicles being identified quickly and lanes closed promptly, but radar-based stopped vehicle detection was not in place when the schemes opened. Its rollout ran roughly six years late and was completed across the ALR network only in September 2022. Early schemes therefore depended on CCTV and third-party reports, and the Transport Committee found that CCTV was not routinely monitored, being used reactively to verify incidents already reported [9]. An inquest heard National Highways staff accept that the assumption cameras would spot a stationary vehicle was not practicable. This is the cleanest form of the failure: the safety argument rested on a detection capability that had been promised but not yet delivered. Operating outside the assumed conditions was not non-compliance, but it hollowed out the basis for confidence.

Third, **role-based perspectives diverged**. From an operator's perspective, compliance metrics focused on system availability and response processes. From a driver's perspective, the absence of a hard shoulder translated directly into acute psychological and physical risk. From an emergency responder's perspective, lane access and predictability were degraded. Assurance reasoning did not reconcile these perspectives into a single, defensible claim about safety.

Fourth, **reasonable foreseeability was treated narrowly**. While individual failure modes were assessed, the scenario of a vehicle breaking down in a live lane was perceived as statistically low likelihood and operationally manageable. To the public, however, this scenario was intuitively foreseeable and unacceptable. After fatalities occurred, hindsight collapsed the distinction between unlikely and obvious.

Finally, **organisational drift and change** compounded the problem. As smart motorways expanded and operational practices evolved, assumptions embedded in earlier safety cases were no longer uniformly valid. Assurance was not sufficiently revisited as scale increased and context changed.

The result was a collapse of trust. Despite formal compliance, public confidence eroded to the point where the programme was curtailed: in April 2023 all future smart motorway schemes were cancelled, and a retrofit programme added more than 150 emergency areas to existing roads, completing in spring 2025 [10, 11]. It is worth being precise about what did not happen. Existing all lane running motorways were not converted back, and the hard shoulder has not been reinstated. The remedy addressed the consequences of the design while leaving the design in place, which is itself an assurance judgement.

A caveat matters here, because the casualty data does not support the simplest reading. Between 2015 and 2019 the fatal casualty rate on all lane running motorways was lower than on conventional motorways. The defensible finding is narrower and more interesting: fatal and serious injury rates rose after conversion, and the specific hazard of being struck while stopped in a live lane increased. Controlled Motorway, which retains the hard shoulder,

recorded the lowest casualty rate of any motorway type [9]. The assurance failure is therefore not that the roads were simply more dangerous in aggregate, but that a design satisfying its aggregate safety case redistributed risk onto a small group of people in a situation they experienced as indefensible. Aggregate metrics can be met while a foreseeable harm concentrates.

Smart motorways demonstrate that assurance cannot be reduced to demonstrating that a system functions as designed. It must also address how that design is experienced by different stakeholders, under stress, and at scale.

5.5. Foreseeability as an Assurance Discipline

Treating foreseeability seriously requires:

- explicit articulation of assumptions,
- consideration of scale, aggregation, and misuse,
- engagement with non-technical stakeholders,
- and humility about uncertainty.

Foreseeability is not about predicting the future perfectly. It is about being able to explain, credibly and transparently, why certain harms were or were not considered, and what would change that judgement.

Assurance does not eliminate hindsight. It prepares organisations to withstand it.

6. Principles-Based Assurance

By this point, we can take as given that compliance alone is insufficient to justify trust in complex, evolving systems. The preceding sections have established why assurance must reason across product, organisational, and socio-technical boundaries; why harm often emerges from scale, interaction, and misuse rather than defects; and why confidence must survive change rather than snapshot inspection.

This section explains why principles are the correct abstraction layer for assurance, and how they allow reasoning to remain coherent across regulatory churn, organisational boundaries, and system evolution. Principles are not an alternative to regulation or standards. They are the stable layer that allows assurance reasoning to persist when controls, frameworks, and certification mechanisms change.

6.1. Regulatory Intent vs Compliance Instrumentation

Many contemporary regulatory regimes are principles-led in intent but compliance-led in execution. This is not an accidental flaw; it is a structural property of regulation at scale.

Legislators increasingly express obligations in outcome-oriented language: appropriate security, proportionate risk management, reasonably foreseeable misuse, through-life responsibility. However, regulators must operationalise these obligations through inspection, audit, and enforcement mechanisms that are necessarily repeatable, comparable, and resource-efficient. The result is an inevitable translation from assurance intent to compliance instrumentation.

The EU Cyber Resilience Act (CRA) [12] is a clear illustration. Its essential requirements are framed around achieving an appropriate level of cybersecurity relative to risk, intended purpose, and reasonably foreseeable use across the lifecycle. These are assurance questions. In practice, the CRA is enforced through conformity assessment routes, harmonised standards, and presumptions of conformity. These mechanisms do not define the obligation; they approximate it in a form regulators can scale.

This pattern is neither new nor unique. UNECE R155 [13] exhibits the same structure. It explicitly combines an organisational obligation (the Certificate of Compliance for a Cyber Security Management System, Annex 4) with product-facing type approval requirements. Annex 5 describes minimum mitigation categories, but these are acknowledged as a floor, not a complete expression of cybersecurity assurance. The regulation's centre of gravity

is organisational risk capability, yet market confidence is often sought through discrete, inspectable artefacts such as penetration tests or witnessed assessments.

A useful contrast can be drawn with emissions regulation. There, regulatory confidence is anchored to highly standardised, repeatable tests against quantitative limits. Assurance is collapsed almost entirely into measurement. Cybersecurity and socio-technical risk do not permit this collapse without loss. Attempting to treat cybersecurity like emissions testing leads to misplaced confidence in artefacts that are, at best, partial proxies for trust.

The critical point is this: compliance artefacts are not the obligation; they are an enforcement convenience. Principles-based assurance is the layer that allows organisations to reason coherently across the gap between regulatory intent and compliance instrumentation, without mistaking the proxy for the property being regulated.

6.2. Principles as the Stable Layer

Principles occupy a distinct epistemological role in assurance. They express what must be true for trust to be justified, independent of how those truths are implemented at any given point in time.

Controls, standards, and certification schemes encode how principles were instantiated under particular assumptions: about technology, threat models, organisational structure, and enforcement capability. These assumptions inevitably age. When assurance is anchored directly to controls, confidence decays silently as context shifts.

Principles provide stability across:

- • regulatory churn and evolving standards,
- • technological change and increasing system scale,
- • organisational restructuring and supply-chain fragmentation,
- • and the emergence of novel threat classes not anticipated by existing controls.

Assurance fails when principles and controls are conflated: when satisfying a checklist is treated as evidence that the underlying principle has been met. In reality, conformance demonstrates only that one instantiation of a principle was followed, not that the principle itself continues to hold.

This distinction explains why control-to-control mapping between standards so often feels hollow. Such mapping preserves surface equivalence while discarding intent. Mapping controls to principles, and then reasoning from principles to claims, preserves meaning and allows confidence to persist even as the control landscape evolves.

6.3. The NCSC Technology Assurance Principles as an Example

The UK National Cyber Security Centre's Technology Assurance Principles (TAPs) [1] provide a concrete illustration of principles functioning as a stable assurance layer.

The TAPs deliberately separate concerns into three enduring classes:

Product development [14]

- Design for user need
- Enable your developers
- Manage your supply chain risk
- Secure your development environment
- Review and test frequently
- Manage change effectively

- Build for through-life

Product design and functionality [15]

- Usability of the product
- Restrict access to authorised users
- Protect sensitive data when in transit
- Protect against unauthorised access and modification
- Protect against compromise from connected technology
- Security events should be logged and monitored

Through-life [16]

- Enable people to manage their risks
- Protect the product from unauthorised access or modification
- Monitor services used by a product
- Review and improve the security offered by a product
- Be prepared to respond to external events

What is notable is not the individual content of these principles, but their distribution across time and responsibility. They explicitly resist the common failure mode of time-slice assurance, where confidence is asserted at design or release and implicitly assumed thereafter.

The TAPs do not prescribe specific controls. Instead, they articulate what *good must look like* while allowing context-sensitive interpretation. It is worth noting that the NCSC does not treat these principles as an endpoint. They sit within a wider programme, Principles Based Assurance (PBA) [17, 1], described as “a new approach to Assurance, one which gives due consideration to context and risk, providing a security assessment which is appropriate to the context in which a system will be used.” PBA is explicitly framed as “a risk-based rather than a compliance-driven approach”, and its three assessment pillars: development; design and functionality; and through-life, map directly onto the three principle groups above.

More significant for the argument developed here is the mechanism PBA uses to get from principles to assessment. The NCSC introduces a set of artefacts it calls Assurance, Principles and Claims (APCs), which restructure the published principles using a Claims, Argument, Evidence approach borrowed from safety assurance and adapted for cybersecurity. A claim is defined as a statement about a property of a particular object; an argument is the rule that bridges evidence to that claim; evidence is the artefacts that establish the supporting facts. That is the same translation step described in From Principles to Claims in the Presence of an Item Definition and the same structuring model adopted in Claims, Arguments, and Evidence, arrived at independently by the UK national technical authority. The convergence matters: it suggests that principles-based, claims-led assurance is not a private methodological preference but the direction in which national assurance practice is already moving. PBA remains a programme under development rather than settled practice, and this paper does not depend on it but readers encountering these ideas for the first time should know that the regulator's own thinking runs along the same lines.

6.4. From Principles to Claims in the Presence of an Item Definition

Principles do not become assurance claims in the abstract. A principle can only be transformed into a meaningful, defensible claim **in the presence of a defined item**.

An Item Definition – as described in Product / System Assurance: Defining the Item Under Assurance, establishes *what the claim is about*. It bounds scope, identifies roles and relationships, fixes lifecycle state, and makes explicit the Concept of Operations (ConOps) and Operational Design Domain (ODD) within which the claim is asserted. Without this grounding, claims drift toward generality and are easily over-extended beyond the context for which confidence was actually justified.

This distinction matters because principles are intentionally broad and enduring. They express what must be true for trust to be justified, but they do not specify *where*, *for whom*, or *under what conditions* that truth is expected to hold. Item Definition supplies those missing dimensions.

For example, the same principle may give rise to materially different claims depending on how the item is defined:

- **Principle:** Protect against unauthorised access and modification.
- **System-level claim:** The deployed system prevents unauthorised modification of safety-critical configuration data within its defined operational environment and lifecycle state.
- **Service-level claim:** Operational processes prevent unauthorised modification of live system configuration during maintenance and support activities.
- **Organisational-level claim:** The organisation sustains controls that prevent unauthorised modification of production systems across suppliers, environments, and through-life change.

These are not alternative phrasings of the same claim. They are distinct commitments that arise because the *item under assurance* is defined differently in each case. Attempting to assert them without explicit item definition leads to category errors: evidence gathered for one object (for example, a component or development process) is assumed to support confidence in another (such as a deployed service or socio-technical system).

A weak or implicit Item Definition is therefore one of the most common root causes of assurance failure. It allows claims to be asserted broadly while evidence remains narrow, creating the appearance of confidence that collapses under scrutiny. Conversely, a disciplined Item Definition constrains claims deliberately. It makes clear where assurance applies, where it does not, and what assumptions would need to change before claims must be revisited.

In a principles-based assurance approach, this transformation step is where intent becomes commitment. Principles express what matters. **Claims state what is being promised, for a specific item, in a specific context.** Only once that transformation has occurred can arguments and evidence meaningfully justify confidence.

It is also important to recognise a frequent but subtle failure mode at this stage: **the sum of Item Definitions for components does not constitute an Item Definition for a functional cluster or a system**.

In the same way that a collection of components that are individually “safe” does not guarantee that the system formed from them is safe in combination, a collection of component-level Item Definitions does not automatically describe the behaviour, risks, or assumptions of a higher-level item. System-level properties emerge from interaction, orchestration, timing, privilege boundaries, and operational context, none of which are fully captured by component descriptions alone.

Component-level Item Definitions are therefore *inputs* to higher-level assurance, not substitutes for it. Only those assumptions, limitations, interfaces, and behaviours that remain relevant in the new context should propagate upward. Others may be invalidated, amplified, or transformed by integration.

This is particularly acute for functional clusters and systems-of-systems, where components are combined in ways that were not anticipated by their original designers, operate under different threat models, or are subject to new roles, scale, and incentives. Treating component Item Definitions as additive encourages over-claiming and masks emergent risk.

Effective assurance instead requires that each abstraction layer (component, functional cluster, system, service) has its **own explicit Item Definition**, informed by but not reducible to those below it. This disciplined re-definition is what allows principles to be translated into claims that remain valid as complexity increases.

6.5. Claims That Do Not Derive Directly from Principles

While principles provide a stable anchor for assurance reasoning, it is important to recognise that not all meaningful assurance claims arise directly from principles.

Principles articulate what must be true for trust to be justified across a class of systems, services, or organisations. However, many claims that matter operationally, commercially, and societally emerge from contextual commitments, architectural choices, or deployment expectations rather than from foundational principles alone.

Examples include claims such as:

- the system can be deployed without dependence on a specific cloud or infrastructure provider,
- the service maintains defined availability characteristics under stated operational conditions,
- the system remains operable or recoverable within agreed time bounds,
- the service can be exited, migrated, or replaced without disproportionate disruption,
- or the product can be integrated into heterogeneous environments without privileged dependencies.

These are not abstract principles. They are assurance-relevant claims that shape trust, risk, and dependency, particularly at scale.

Service Level Agreements (SLAs) provide a useful illustration. From a narrow technical security perspective, availability might be treated as one of the classic CIA properties. In practice, SLAs are far more than technical uptime metrics. They formalise assumptions about operational behaviour, response time, recovery, escalation, and accountability, all of which directly affect the Operational Design Domain (ODD) of a system or service.

An availability commitment expressed through an SLA therefore functions as an assurance claim about acceptable operational degradation, recoverability, and continuity under stress. These properties are not always derived cleanly from core security principles, but they are essential to justified confidence in real-world deployment.

Similarly, claims about vendor neutrality, portability, or deployment independence are rarely traceable to a single principle. They emerge from architectural and organisational decisions intended to manage concentration risk, supply chain fragility, and systemic dependency. In critical or high-impact systems, such claims can be central to resilience, sovereign capability, and long-term trust, even where no regulation explicitly requires them.

What matters from an assurance perspective is not whether such claims can be traced directly to a named principle, but whether:

- they are made explicit rather than assumed,
- they are grounded in a clear Item Definition (including lifecycle and ODD),
- the arguments supporting them are visible,
- and the evidence used to justify them is appropriate to the claim.

In emerging markets and novel system classes, this distinction becomes even more important. Principles often lag innovation. The claims that differentiate a product, service, or system, and that communicate its unique value proposition, may exist outside any formalised principle set. That does not make them weaker; it makes them more in need of explicit assurance reasoning.

This also reinforces an important asymmetry: principles inform claims, but claims are not constrained to principles. Attempting to force all claims to collapse into an existing principle taxonomy risks suppressing precisely the reasoning that assurance exists to surface, particularly around dependency, scale, integration, and socio-technical impact.

For this reason, good assurance practice allows claims to emerge from:

- principles (where they exist),

- Item Definition and ODD commitments,
- architectural and organisational intent,
- and explicit promises made to customers, regulators, or the public.

Claims that arise in this way are still subject to the same discipline. They must be justified through arguments and evidence, bounded by assumptions, and revisited as context changes. The fact that they are not "principle-derived" does not exempt them from scrutiny; it makes that scrutiny more important.

This framing also helps resolve a common confusion: assurance is not about selecting principles that fit the item, but about reasoning honestly from the item, its context, and its commitments toward defensible claims. Principles provide stability. Claims provide specificity. Assurance exists in the disciplined relationship between the two.

6.6. Principles Across Product, Service, and Organisation

Principles-based assurance makes explicit that claims exist at multiple, interdependent levels:

- **Product/System claims** address behaviour, failure modes, and technical properties.
- **Service claims** address how systems are operated, supported, and integrated.
- **Organisational claims** address governance, capability, and sustained control.

Confidence at any one level depends on the others. Product assurance depends on organisational capability to develop, deploy, and maintain systems. Organisational assurance depends on the properties and behaviours of the products and services delivered.

This interdependence explains why many complex regulatory and compliance obligations sit at interfaces: secure development lifecycle requirements, supply-chain assurance, vulnerability handling, and change management. Principles prevent assurance gaps at these boundaries by providing a common frame of reference across roles and responsibilities.

6.7. Handoff to Structured Assurance

Principles define what must be true for confidence to be justified. Claims make that intent explicit for a defined item and context.

There are multiple established ways of expressing structured assurance reasoning. Approaches such as Goal Structuring Notation (GSN) [18], safety cases, and related argument-based methods are widely used across safety-critical and regulated industries. Each provides mechanisms for making goals, assumptions, context, and justification explicit, and each can be effective when applied with discipline.

In this paper, Claims–Arguments–Evidence (CAE) is used deliberately as the primary structuring model. The preference is pragmatic rather than doctrinal. CAE's separation of claims, arguments, and evidence aligns closely with how practitioners reason about assurance in practice, and its minimalism makes it more intuitive for mixed audiences spanning engineering, security, safety, governance, and regulation.

Importantly, the assurance principles described here are not tied to CAE as a notation. They can be realised using GSN or other structured argument approaches without loss of intent. What matters is not the diagramming style, but the explicit articulation of claims, the reasoning that justifies them, and the evidence on which that reasoning depends.

The remaining question, therefore, is not which notation to use, but how claims are justified in practice which requires explicit arguments and evidence. This is addressed in Claims, Arguments, and Evidence.

7. Claims, Arguments, and Evidence

With principles-defined claims in place, assurance turns on a simple but demanding question:

Why should we believe these claims hold in the real world?

Answering that question requires more than evidence collection. It requires explicit reasoning that links intent to reality. This is the role of Claims, Arguments, and Evidence (CAE).

CAE is not a notation exercise or a documentation burden. It is a discipline for making assurance reasoning visible, reviewable, and challengeable.

7.1. Claims: Contextual Assertions About a Defined Item

All assurance claims are claims **about a defined item**.

Claims are only meaningful when their scope, assumptions, and context are anchored to a clear Item Definition, as set out in Product / System Assurance: Defining the Item Under Assurance. Without this anchor, claims drift toward abstraction or slogan, and evidence appears relevant without actually supporting the conclusion being asserted.

A claim is a statement that must be true for trust to be justified.

Good claims are:

- specific to context and scope,
- bounded by assumptions and operating conditions,
- meaningful to decision-makers.

Poor claims are vague, absolute, or borrowed directly from standards.

For example:

- Weak claim: “The system is secure.”
- Stronger claim: “The system prevents unauthorised modification of safety-critical configuration data within its defined operational environment.”

Claims derived from principles provide the anchor points of the assurance case. They define what confidence actually means in practice.

7.2. Arguments: Making Reasoning Explicit

Arguments explain **why** a claim should be believed.

They articulate:

- how design decisions support the claim,
- how organisational practices sustain it,
- how risks, trade-offs, and constraints are managed,
- and under what assumptions the reasoning holds.

Without arguments, evidence becomes a pile of artefacts with no coherent meaning. Arguments are where professional judgement lives.

In practice, arguments often follow recurring patterns, such as:

- decomposing a claim into necessary sub-claims,
- substituting a proxy system or environment where direct evidence is unavailable,
- bounding claims through assumptions and exclusions,
- or demonstrating defence-in-depth across multiple controls.

Making these patterns explicit prevents accidental over-claiming and exposes gaps that would otherwise remain hidden.

7.3. Evidence: Supporting, Not Defining, Confidence

Evidence consists of artefacts that support or challenge an argument. Examples include:

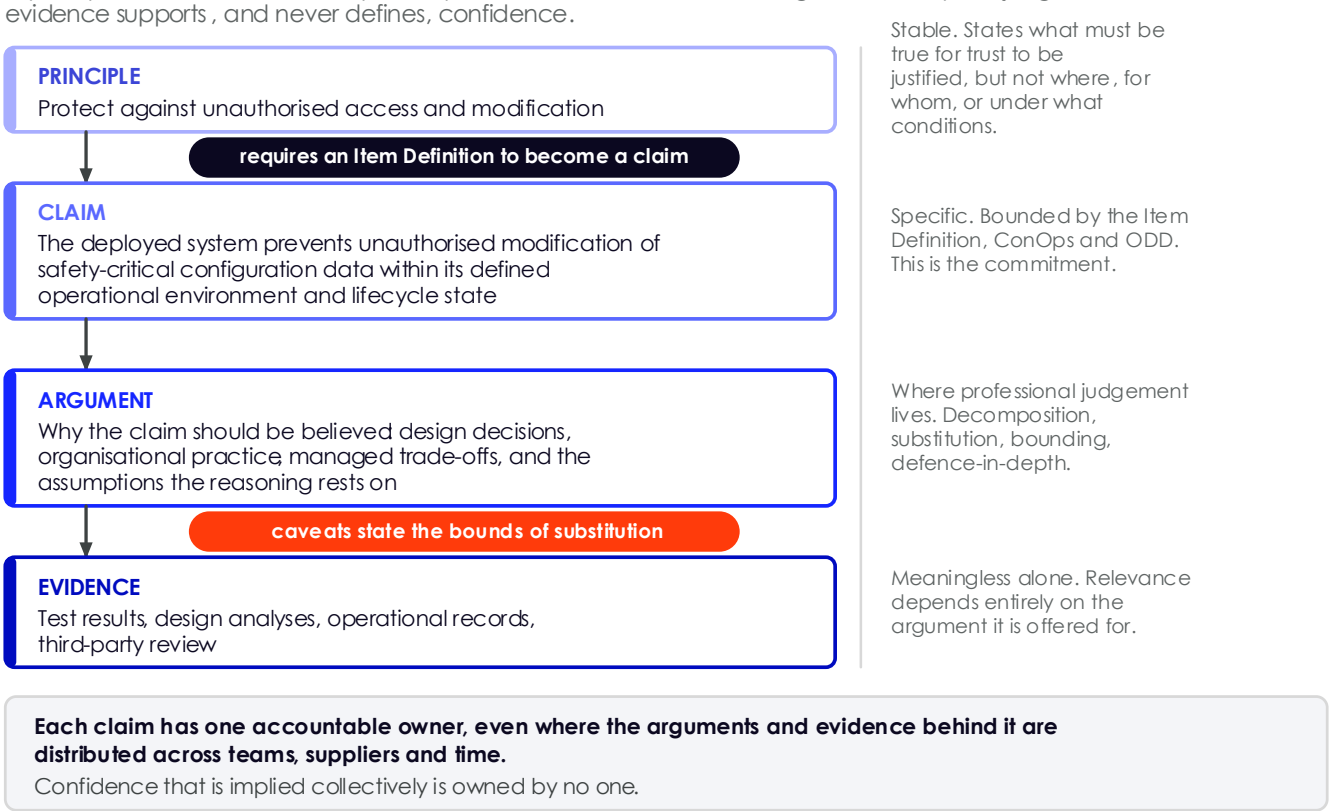
- test results and assessment reports,
- design documentation and analyses,
- operational records and metrics,
- third-party reviews and audits.

Crucially, evidence does not speak for itself. Its relevance depends entirely on the argument it is intended to support.

Standards and certifications therefore contribute evidence. They do not, on their own, justify claims. Treating evidence as a substitute for reasoning is the core failure mode of compliance-led assurance.

Figure 8 From principle to evidence

A principle becomes a claim only in the presence of a defined item . Arguments carry the judgement; evidence supports , and never defines, confidence.



Each claim has one accountable owner, even where the arguments and evidence behind it are distributed across teams, suppliers and time.
Confidence that is implied collectively is owned by no one.

7.4. When Claims Are Too Complex or Abstract

Not all claims lend themselves to straightforward argumentation using simple decomposition or direct evidence attachment. In complex socio-technical systems, some claims are inherently broad, abstract, or multi-layered, encompassing technical behaviour, organisational capability, human interaction, and evolving context.

Examples include claims such as:

- The system operates securely under all reasonably foreseeable conditions.
- The organisation sustainably manages cyber risk across its portfolio.

- The service will not cause unacceptable harm as it evolves over time.

These claims may be meaningful at a decision-making level, but they are too coarse to be supported directly by individual tests, audits, or assessments. Attempting to do so often results in large collections of evidence with no clear line of reasoning.

7.4.1. Using Argument Building Blocks

To manage this complexity, assurance practice makes use of **argument building blocks**: reusable, semantically well-defined fragments of reasoning that can be composed to structure complex assurance arguments.

Well-established building block taxonomies exist within the Claims–Arguments–Evidence discipline, including:

- The CAE building blocks and connection rules described by the Claims–Arguments–Evidence community [19, 20].
- The CAE mini-guide on argument blocks and patterns [21].
- The building block approach to assurance cases described by Bloomfield and Netkachova [22].

These sources describe blocks such as **Concretion**, **Decomposition**, **Substitution**, **Calculation / Proof**, and **Evidence Incorporation**, alongside side-claims and justification constructs that make inference conditions explicit.

7.4.2. When Building Blocks Are Appropriate

Structured argument building blocks are particularly valuable when claims cannot be supported by a single, static line of reasoning, but instead depend on **how confidence is constructed across lifecycle states, abstraction layers, and integration boundaries**.

A common trigger for building-block use is the temptation to treat assurance as additive: to assume that confidence in a system can be constructed by aggregating confidence in its constituent parts. This assumption is routinely false. The same non-additivity set out for Item Definitions in From Principles to Claims in the Presence of an Item Definition applies to arguments: a collection of component-level arguments does not automatically justify a system-level claim.

Building blocks become appropriate precisely when this non-additivity must be made explicit and reasoned about.

In practice, this most often arises in the following situations:

First, **claims that span multiple lifecycle states**. Confidence in a product or system is rarely justified in a single state. During development, arguments may rest on design intent, threat modelling, and architectural analysis. During manufacturing or provisioning, confidence may depend on flashing processes, secure enclaves, key management, and production environment controls. During operation, arguments shift again toward monitoring, incident response, update mechanisms, and organisational capability. Building blocks allow these distinct but related arguments to be composed without collapsing them into a single, misleading narrative.

Second, **claims that rely on component, functional cluster, and system abstraction layers**. Component-level Item Definitions and arguments can legitimately contribute to higher-level assurance, but only through explicit reasoning about what propagates upward. Assumptions, limitations, interfaces, and threat models often change when components are integrated into functional clusters or systems. Building blocks such as decomposition and concretion make these transitions visible, preventing implicit over-claiming that arises when component assurance is treated as directly transferrable.

Third, **claims that involve supplier-provided, off-the-shelf, or pre-configured components**. Where assurance depends on artefacts supplied by third parties, building blocks help distinguish between reliance on supplier claims, reliance on supplier evidence, and new claims introduced by configuration, integration, or assembly. This is particularly important where responsibility shifts across organisational boundaries, and where only a subset of supplier assumptions remain valid in the receiving context.

Fourth, **claims that depend on configuration, assembly, or provisioning states**. Many critical security properties are not fixed at design time but emerge during configuration, assembly, flashing, or secure provisioning. Examples

include trust anchor establishment, key injection, secure boot configuration, and privilege separation between manufacturing and operational modes. Building blocks allow assurance arguments to explicitly reflect these state-dependent properties rather than implying that design-time reasoning alone is sufficient.

Fifth, **claims that must remain valid through operation, maintenance, and PSIRT (product security incident response team) activity**. Through-life assurance necessarily involves substitution and bounded confidence. Evidence gathered at one point in time is used to justify confidence at another. Building blocks provide a disciplined way to express these substitutions, the caveats they introduce, and the conditions under which claims must be revisited.

Two conditions cut across all five. Confidence often rests on proxy evidence, simulation, or environments that are not representative of deployment; and evidence frequently has to be integrated across safety, security, resilience, and organisational domains. Both make the inference steps load-bearing rather than incidental, which is precisely what building blocks are for.

Across all of these cases, the purpose of building blocks is not to add formalism for its own sake. It is to make visible where confidence is constructed, where it is inferred, and where it depends on assumptions that are sensitive to lifecycle state, integration context, or organisational control.

Used in this way, building blocks help ensure that assurance arguments scale with system complexity without collapsing into either unjustified generality or unmanageable detail.

Structured this way, building blocks allow assurance reasoning to be expressed as a composition of arguments, rather than a single linear narrative. This improves clarity, reviewability, and robustness under challenge.

7.5. Building Blocks as an Aid, Not an Overhead

The use of building blocks does not require specialised tooling or formal notation. Their primary value lies in providing a shared mental model for structuring reasoning.

Explicitly acknowledging that some claims require such structure helps avoid two common failure modes:

- treating abstract claims as self-evident and therefore under-justified, and
- attempting to support complex claims with undifferentiated evidence.

Recognising when a claim is too complex to be argued informally is itself an assurance skill.

Equally importantly, the use of building blocks provides a **shared narrative and vocabulary** for discussing assurance across different roles and organisations. By making the structure of reasoning explicit, building blocks help align engineers, safety specialists, security practitioners, governance teams, regulators, auditors, and suppliers around *why* confidence is being asserted, not just *what* artefacts exist.

This shared language reduces misunderstanding at interfaces, supports constructive challenge, and enables assurance discussions to move beyond role-specific viewpoints toward a coherent, system-level understanding.

In practice, this also raises an important governance question: **who owns the claims**.

As with information management systems and asset ownership, effective assurance requires clear accountability for claims, even where the supporting arguments and evidence are drawn from across the organisation. Individual teams may own evidence or contribute specialist reasoning, but the claim itself must have an accountable owner who is responsible for asserting it, defending it, and revisiting it as context changes.

Making claim ownership explicit helps avoid a common failure mode in large organisations, where confidence is implied collectively but owned by no one. It also supports more effective interaction with external stakeholders, regulators, and assessors, who need to understand not just the content of an assurance case, but who stands behind it.

A useful comparison can be drawn with **homologation and type approval functions within OEMs**. These teams are often deliberately organisationally distinct from design and engineering groups. Their role is not to generate detailed technical knowledge, but to *own the approval claims*, challenge the sufficiency of arguments, and critique whether the available evidence justifies those claims in the eyes of regulators.

This separation is intentional. It creates constructive tension between those who build systems and those who must assert that the systems meet required properties. Principles-based assurance benefits from a similar model: claim owners may draw on expertise and evidence distributed across the organisation, but they retain independent responsibility for the coherence, defensibility, and integrity of the assurance case as a whole.

7.6. Substitution and Caveats

In practice, a large proportion of cybersecurity and assurance deliverables rely, implicitly or explicitly, on **substitution**. Evidence is gathered from systems, environments, organisations, or timeframes that are not fully representative of the real-world deployment for which confidence is ultimately being asserted.

This is particularly common in:

- penetration testing and vulnerability assessments,
- design and architecture reviews,
- security maturity or gap assessments,
- supplier or component evaluations,
- and pre-deployment assurance activities where live systems cannot be tested directly.

In these contexts, assurance reasoning is necessarily *indirect*: confidence about one item is inferred from evidence about another. For example:

- a penetration test of a staging environment is used to infer properties of a production deployment,
- a review of an off-the-shelf component is used to support confidence in a tailored or integrated variant,
- or a point-in-time assessment is used to justify through-life claims about an evolving system.

An assurance-led approach requires that this substitution is made explicit and reasoned about, rather than treated as an unfortunate footnote.

7.6.1. Caveats as Assurance Reasoning, Not Disclaimers

In the cybersecurity industry, caveats are often treated as defensive disclaimers: text included to limit liability, signal uncertainty, or protect the assessor rather than to inform the reader. As a result, caveats are frequently undervalued, skimmed, or ignored.

From an assurance perspective, this is a missed opportunity.

Properly constructed caveats are **part of the argument**. They articulate the boundaries of substitution and therefore define the scope of justified confidence. A good caveat answers questions such as:

- What exactly was tested, reviewed, or observed?
- In what ways does that differ from the real system of concern?
- Which assumptions are being made to bridge that gap?
- Under what conditions would those assumptions no longer hold?

For example, in a penetration test report:

- A weak caveat might state that “findings are limited to the scope tested.”

- A stronger, assurance-relevant caveat would explain that confidence about production behaviour is being inferred from a staging environment, that certain classes of operational risk (such as scale-dependent abuse or insider misuse) were not observable, and that the claim supported is therefore bounded accordingly.

This reframing turns caveats from liability shields into **decision-support tools**.

7.6.2. Improving the Value of Common Cybersecurity Deliverables

Many common cybersecurity deliverables already contain the raw material needed for stronger assurance; what is often missing is explicit reasoning about substitution.

For example:

- **Penetration tests** typically demonstrate that *no easily exploitable vulnerabilities were found under specific conditions*. They rarely justify broader claims such as "the system is secure." By explicitly linking findings to bounded claims (e.g. resistance to a defined attacker model within a defined environment), penetration tests can contribute meaningfully to assurance without over-claiming.

- **Architecture and design reviews** often reason about intended behaviour rather than observed behaviour. Making the substitution explicit, *this argument relies on the design being implemented as described*, clarifies where configuration drift, integration complexity, or operational deviation could undermine confidence.

- **Security reviews and maturity assessments** frequently aggregate evidence across teams or products. Explicit caveats about heterogeneity, sampling, or uneven control maturity help asset owners understand which parts of the organisation the claims genuinely apply to.

When substitution is acknowledged explicitly, deliverables become more valuable, not less. They provide a clearer signal of what confidence is justified today, and where further evidence would most effectively strengthen the assurance case.

7.6.3. Substitution, Risk Appetite, and Honest Assurance

It is important to recognise that substitution is not inherently wrong. In many OT, CNI, and safety-critical environments, it is unavoidable due to safety constraints, availability limitations, or operational risk.

What matters is not whether substitution occurs, but whether it is **honest**.

An assurance-led approach allows organisations to say, explicitly:

- this claim is supported by proxy evidence,

- this level of confidence is acceptable given current risk appetite,

- and these are the conditions under which we would need to revisit it.

This transparency enables better decision-making. Asset owners can distinguish between *unknown risk* and *known residual risk*. Regulators and assessors can see where confidence is bounded rather than overstated. Engineers and security practitioners can target effort where it meaningfully improves the assurance case, rather than simply expanding test scope without clarity of purpose.

By contrast, when substitution remains implicit, caveats are minimised, and claims are overstated, organisations accumulate assurance debt. Confidence appears higher than it truly is, until real-world deployment, scale, or misuse exposes the gap.

Explicit substitution reasoning therefore does not weaken cybersecurity deliverables. It makes them **fit for purpose as assurance artefacts**, aligned with the realities of complex systems and constrained environments.

7.7. Why CAE Matters

Explicit CAE reasoning enables:

- meaningful peer and third-party review,
- accumulation of assurance across engagements and time,
- resilience to regulatory and contextual change,
- and transparent communication of uncertainty to decision-makers.

Most importantly, it prevents the illusion of confidence that arises when compliance artefacts are mistaken for assurance.

CAE does not require specialised tooling or formal notation. It requires discipline.

8. Implications in Practice

A common objection to principles-based assurance is that it appears resource-intensive, abstract, or achievable only by large organisations with dedicated assurance teams. This concern is understandable, but it rests on a misconception about where the effort in assurance actually lies.

Many organisations already practice a form of *implicit assurance*. Engineers apply professional judgement. Teams document processes. Compliance evidence is gathered. Systems are tested and reviewed. The problem is rarely an absence of effort. It is that these activities are often:

- fragmented across domains (for example, safety, cybersecurity, privacy, and reliability),
- framed in discipline-specific language,
- anchored in local artefacts such as tests, checklists, or certificates,
- and difficult to communicate, review, or scale beyond the immediate team.

As a result, coherence is lost. Confidence becomes assumed rather than reasoned, and assurance exists only implicitly in the heads of practitioners rather than explicitly in a form that can be challenged, transferred, or sustained.

The argument in this paper is not that organisations need to *do more*. It is that they need to make existing assurance reasoning **explicit, structured, and contextual**.

It is important to be clear about what does *not* work. Retrofitting structured assurance onto a legacy system with an unclear item definition, undocumented assumptions, and fragmented evidence is typically a high-effort, low-yield activity. Attempting to reconstruct claims, arguments, and assumptions after the fact often exposes just how much tacit knowledge has already been lost. For this reason, retroactive assurance should be approached cautiously and selectively.

However, when principles-based assurance is **built in from the outset**, it is both achievable and proportionate, even for resource-constrained organisations.

Embedding assurance early means:

- defining the item under assurance clearly,
- anchoring key design and architectural decisions to principles,
- and allowing explicit claims to accumulate gradually over time as the system evolves.

This is not documentation heavy. It is alignment heavy. The effort is spent aligning:

- roles (engineering, operations, governance),

- lifecycle stages (design, deployment, evolution),
- and expectations (engineering confidence, regulatory obligation, user and societal trust).

Done well, this alignment reduces duplication, clarifies responsibility, and makes the assurance posture visible and challengeable. In practice, this *reduces* wasted effort by preventing teams from working at cross purposes or rebuilding confidence repeatedly for different audiences.

8.1. What Resource-Constrained Teams Can Do Today

Even with limited time and budget, teams can begin applying assurance thinking incrementally:

- **Define the item clearly.** Be explicit about what is being assured, for whom, and under what conditions.
- **Start with a single claim.** State one thing that must be true for confidence to be justified.
- **Attach one argument and one piece of evidence.** Identify why you believe the claim holds, and what artefact supports that belief.
- **Make one assumption explicit.** Record a condition that must remain true for confidence to hold.

This small discipline scales naturally. Over time, claims accumulate, assumptions are revisited, and evidence is refreshed as systems change. Structured assurance emerges organically, without waiting for a wholesale transformation or a dedicated assurance programme.

Crucially, this approach aligns with the broader theme of this paper: assurance is not a deliverable to be bolted on, but a way of reasoning that becomes more valuable as systems grow in impact, complexity, and societal consequence.

An assurance-led approach has concrete implications for how systems are designed, assessed, operated, and governed. It does not require abandoning compliance activities, but it does require reframing their purpose and limits.

8.2. For Engineers

For engineers, principles-based assurance means designing with claims in mind and treating design choices as arguments.

Design choices should be traceable to explicit claims derived from core principles. Architectural decisions, control selections, and failure-handling mechanisms are not guarantees of safety or security; they are **arguments** for why particular claims should hold under stated assumptions.

This has a practical implication that is increasingly important in through-life regimes: engineers and product teams need a structured **feedback loop** from real-world deployment back into the assurance case and design.

In practice, this means:

- being explicit about which risks are being mitigated and which are being accepted,
- documenting the assumptions that design decisions rely on (scale, usage, threat capability, operational behaviour),
- monitoring whether real-world deployments violate those assumptions (including integration into higher-criticality contexts),
- and ensuring traceability across versions showing how new operational information led to updated claims and updated design decisions.

8.3. For Security and Safety Practitioners

For security professionals, safety engineers, and risk practitioners, assurance-led practice shifts the focus from control enumeration to **claim interrogation**.

Rather than asking only whether a control exists or a test has been performed, practitioners should ask:

- what claim this activity is intended to support,
- whether the available evidence is sufficient for that claim,
- where substitution is occurring and on what assumptions it relies,
- and how scale, misuse, or changing context could undermine confidence.

This framing aligns directly with the treatment of caveats and substitution in Substitution and Caveats**,** and should be read as a practical application of that discussion rather than a restatement of it. The objective here is not to re-argue the limits of common cybersecurity activities, but to operationalise them: to turn everyday deliverables into clearer inputs to assurance reasoning.

Applied consistently, this approach legitimises constructive challenge of inherited architectures, vendor assertions, and historical decisions without requiring perfect hindsight. It provides a principled basis for writing caveats that inform decision-making, articulating residual risk honestly, and recommending proportionate improvement that strengthens arguments rather than merely expanding control coverage.

8.4. For Organisational Leaders and Asset Owners

For organisational leaders, asset owners, and accountable executives, assurance is fundamentally about **owning risk-informed claims**.

Effective assurance governance requires:

- clear ownership of top-level claims,
- mechanisms to challenge whether arguments remain valid as context changes,
- and willingness to accept or reject residual risk consciously rather than implicitly.

This mirrors established practice in safety cases, information asset ownership, and homologation functions: confidence is asserted deliberately, not assumed collectively.

8.5. For Regulators, Assessors, and External Stakeholders

For regulators, auditors, and independent assessors, a principles-based assurance approach supports more meaningful engagement.

Rather than focusing solely on artefact completeness, external stakeholders can:

- examine whether the claims being asserted are appropriate to the risk and context,
- assess whether arguments are coherent, proportionate, and internally consistent,
- and understand the limits of confidence implied by the available evidence.

This shifts assessment away from checklist sufficiency toward the *quality of reasoning*, while still allowing compliance artefacts to play their intended role as evidence.

Importantly, this closes the loop with the distinction set out in **Regulatory Intent vs Compliance Instrumentation**. Regulatory obligations are expressed in terms of outcomes and justified confidence but enforced through inspectable proxies. A principles-based assurance view allows assessors to judge whether those proxies genuinely serve regulatory intent in the specific context at hand, rather than mistaking conformance to instrumentation for satisfaction of the underlying obligation.

This enables constructive challenge without conflating non-compliance with failure, or compliance with safety.

8.6. What Changes Day-to-Day

In day-to-day practice, assurance-led working results in:

- clearer scoping conversations focused on claims rather than activities,
- reports that distinguish between evidence strength and confidence scope,
- caveats that explain reasoning boundaries rather than disclaim responsibility,
- and recommendations that strengthen arguments instead of merely adding controls.

The result is not more documentation, but better reasoning and confidence that can withstand scrutiny when systems behave in unexpected ways.

8.7. Real-World Monitoring, Learning from Events, and Assurance Renewal

Assurance does not end at deployment or certification. For systems and services that operate in the real world, **operational monitoring and learning from events are themselves assurance activities**, not merely operational hygiene.

This continuous assurance loop is particularly important for risks that arise from the **malicious or harmful use of intended functionality**, including insider threat scenarios discussed in Insider Threat as Malicious Use of Intended Functionality. In such cases, traditional vulnerability discovery or compliance monitoring may never trigger an alert, because systems are operating exactly as designed. Operational monitoring, near-miss analysis, and event learning therefore become the primary means by which organisations detect when trust assumptions embedded in roles, privileges, scale, or business logic are no longer holding in practice.

Modern regulatory regimes increasingly recognise this implicitly. The EU Cyber Resilience Act (CRA) [12] introduces explicit obligations around vulnerability handling, incident reporting, and through-life security support. NIS2 [23] requires entities to handle incidents (Article 21(2)(b)) and report significant incidents (Article 23); reporting of near misses is voluntary (Article 30), reflecting an expectation that risk understanding evolves through operation rather than being fixed at design time.

From an assurance perspective, these obligations are best understood as inputs into a **continuous assurance feedback loop**:

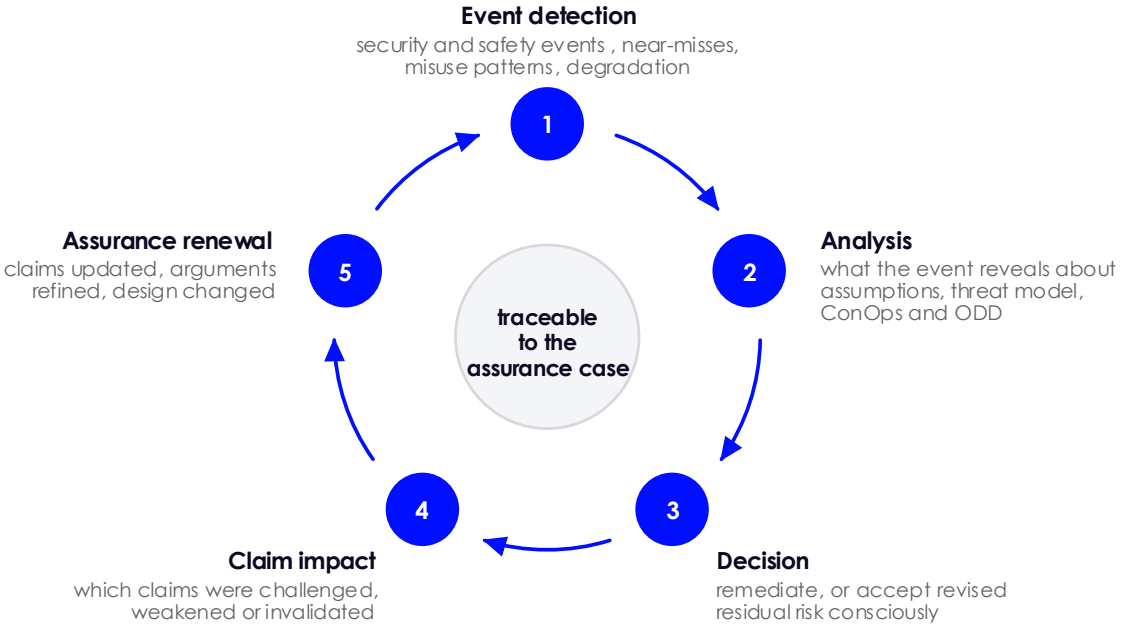
- **Event detection and observation**: security events, safety incidents, near-misses, data breaches, anomalous behaviour, misuse patterns, and operational degradation are identified through mechanisms such as SIEM, SOC monitoring, safety reporting systems, and operational telemetry.
- **Analysis and interpretation**: events are analysed not only for immediate containment, but for what they reveal about underlying assumptions, threat models, ConOps, ODD, and system interactions. Near-misses are particularly valuable, as they expose latent risk without requiring harm to have occurred.
- Decision and response: remedial action may be taken, but it is equally important to recognise that not every risk can be reduced to a level that is as low as reasonably practicable (ALARP) [24, 25, 26] for a given organisation or product. The ALARP judgement weighs the sacrifice, in money, time or trouble, against the risk averted, and UK practice accepts that bringing legacy systems up to modern standards may not be reasonably practicable. The same bounding appears in other regimes: United States nuclear regulation defines as low as is reasonably achievable with explicit regard to the state of technology and to economic and societal factors [27]. Both are judgements to be justified, not targets to be met. Some risks may be accepted consciously due to feasibility, proportionality, or competing objectives. Assurance requires that such decisions are explicit, justified, and recorded.

- **Claim impact and traceability**: events and decisions must be traceable to the assurance case. This includes identifying which claims were challenged, weakened, or invalidated by the event, and whether arguments or assumptions require revision.

- **Assurance renewal**: where necessary, claims are updated, arguments refined, and evidence refreshed. For products, this may feed directly into design changes, revised secure-by-design decisions, updated documentation, or changes to supported operating environments. For organisations, it may drive changes to governance, capability, or supplier management.

Figure 9 Monitoring and learning as assurance activities

Operational evidence is not only proof that controls run . It is evidence about the continued validity of the assurance case itself .



WHY THIS LOOP IS THE PRIMARY CONTROL FOR HARM FROM INTENDED FUNCTIONALITY

Nothing else fires.
Where systems operate exactly as designed, vulnerability discovery and compliance monitoring may never raise an alert.

Near-misses are the cheap signal
They expose latent risk without requiring harm to have occurred, long-standing practice in safety, converging now in security and privacy.

Not every risk must be driven down.
Some are accepted for feasibility or proportionality. Assurance requires that the decision is explicit, justified and recorded.

Without traceability, debt accrues.
Monitoring decoupled from the assurance case satisfies incident-handling duties while confidence quietly decays.

This traceability from event, to analysis, to decision, to revised claims is critical. Without it, monitoring becomes decoupled from assurance, and organisations accumulate assurance debt even while meeting formal incident handling requirements.

Safety disciplines have long recognised the value of **learning from near-misses** as a core assurance mechanism. Cybersecurity and privacy domains are increasingly converging on the same insight. Incidents that do not cross reporting thresholds, or that are resolved operationally, may still carry significant assurance value by revealing how systems behave under stress, scale, or adversarial pressure.

In a principles-based assurance model, real-world monitoring is therefore not merely evidence of control operation; it is evidence about the *continued validity of the assurance case itself*. It provides early warning when assumptions embedded in the Item Definition, ConOps, or ODD no longer hold, and it enables justified confidence to be renewed rather than assumed.

Treating monitoring, incident handling, and learning as assurance activities aligns regulatory intent with operational reality. It allows organisations to demonstrate not only that they respond to events, but that they **reason from them**, updating claims and design decisions as systems and contexts evolve.

9. There Is No Such Thing as “Secure”

A final but essential clarification is required.

As an industry, we should collectively acknowledge that there is no such thing as a system that is simply “secure”, “safe”, or “resilient” in any absolute sense. These terms describe *aspirations*, not states.

Security, safety, and resilience are not binary properties that can be achieved and then possessed. They are judgements about **risk**: about what harms are plausible, which are acceptable, and which are intolerable in a given context.

Treating security as a state encourages false certainty. Treating it as a risk-based judgement encourages continual reasoning.

9.1. Security as Risk, Not Status

When organisations say a system *is secure*, they usually mean one of three things:

- it complies with a recognised standard,
- it has passed a defined set of tests,
- or it has not yet failed in an observable way.

None of these statements imply the absence of risk. They imply only that certain risks have been addressed to a degree that was acceptable at a particular point in time.

Assurance exists precisely because risk cannot be eliminated. Its purpose is not to declare safety, but to justify confidence *despite uncertainty*.

9.2. Why Absolutes Are Dangerous

Absolute language masks trade-offs when it is used imprecisely. Claims such as “secure by design” or “failsafe” often conceal:

- residual risk that has been accepted implicitly,
- harms that were deprioritised rather than mitigated,
- or assumptions that have not been revisited as context changed.

In socio-technical systems, these hidden trade-offs resurface as loss of trust when reality diverges from the narrative.

This is not an argument against concepts such as *secure by design* in themselves. Used correctly, secure by design can be a powerful assurance construct, but only when it is anchored to explicit principles, traceable claims, and defensible arguments.

In a principles-based, CAE-driven approach, design choices are not assertions of safety; they are **arguments**. They explain *why* particular architectural, functional, or organisational decisions were made in order to satisfy specific claims derived from core principles. When secure by design is treated in this way, it strengthens assurance by making trade-offs visible and reviewable, rather than obscuring them.

A principles-based, CAE-driven approach therefore does not reject secure by design; it disciplines it.

This distinction matters not only for engineers, but for **security practitioners, safety engineers, assessors, and risk professionals** whose day-to-day work often sits downstream of design decisions they did not make.

For these practitioners, terms such as *secure by design* are frequently encountered as assertions rather than explanations. They appear in architectures, procurement documentation, vendor claims, and regulatory narratives, yet are rarely accompanied by a clear account of:

- which principles the design is intended to satisfy,
- which claims it is meant to support,
- what threats, hazards, or misuse cases it was explicitly designed to address,
- and which trade-offs were consciously accepted.

A principles-based assurance approach reframes secure by design as something that can be **examined, challenged, and reasoned about**, rather than taken on trust.

For security professionals, this means being able to ask constructive questions that go beyond control coverage:

- Which claims does this design decision support?
- What assumptions does it rely on about users, attackers, scale, or operation?
- How does this design behave when those assumptions are violated but the system remains compliant?

For safety engineers, it provides a mechanism to connect design intent to foreseeable misuse, human behaviour, and socio-technical interaction, not just component failure modes.

For assurance and risk practitioners, it creates a common frame in which design choices can be evaluated alongside organisational capability, operational constraints, and regulatory expectations.

In all cases, treating design decisions as arguments rather than guarantees enables practitioners to surface residual risk, articulate uncertainty honestly, and maintain confidence as systems evolve. This is essential if secure by design is to function as an assurance-strengthening concept rather than a comforting slogan.

9.3. What Good Assurance Actually Provides, and to Whom

This section addresses *audience-relative assurance*: what kinds of claims assurance can legitimately make, and how those claims relate to different decision-makers.

Good assurance does not claim that a system *is* secure, safe, or resilient in any absolute sense.

Instead, good assurance makes **contextual, audience-aware claims about confidence**, and it does so deliberately. It recognises that different stakeholders require different forms of justification, even when they are ultimately relying on the same underlying reasoning.

At its core, assurance claims answer the question:

Why should this particular audience believe that this system, service, or organisation is trustworthy enough for the decision they are being asked to make?

This immediately exposes a reality that compliance-led approaches tend to obscure: **evidence does not persuade all audiences equally**.

The same body of evidence may be technically rigorous yet socially unconvincing or intuitively reassuring yet technically weak. Assurance therefore requires not only that evidence exists, but that it is *appropriate to the audience and decision context*.

For example:

- Engineers and security practitioners may reasonably place weight on source code analysis, architectural reasoning, formal methods, or MISRA coding-standard compliance data [28].

- Organisational leaders and asset owners may rely on arguments about governance, capability, incident learning, and sustained control.
- Regulators may require traceability, proportionality, and consistency with statutory intent.
- Members of the public, passengers, or service users are rarely persuaded by technical artefacts at all. Their confidence is shaped by observable behaviour, transparency, accountability, and whether systems behave acceptably when things go wrong.

A bundle of source code, test reports, or certificates, however rigorous, does not, on its own, justify public confidence. A certificate stating that a system is fully MISRA compliant with all deviations documented may be meaningful to an assessor, but it cannot plausibly reassure a road user that a vehicle will behave safely in an emergency, or a citizen that a data-driven system will not harm them.

This does not make technical evidence irrelevant. It means that **technical evidence must be mediated through argument**: translated into claims that speak to the concerns, values, and risk perceptions of the audience in question.

Good assurance therefore does not produce a single, universal claim. It produces a **coherent set of related claims**, each grounded in the same underlying reasoning, but framed appropriately for different stakeholders.

Crucially, these claims must remain consistent with one another. Assurance fails when:

- public-facing narratives over-promise relative to technical reality,
- internal technical confidence is asserted without regard to societal impact,
- or regulatory assurances are treated as substitutes for operational trust.

A principles-based, CAE-driven approach makes this alignment possible. Principles provide the stable intent. Claims express what confidence means for a given audience. Arguments explain *why* that confidence is justified. Evidence supports those arguments in ways that are meaningful to the decision being taken.

In this sense, assurance is not about persuading everyone with the same artefacts. It is about ensuring that **every claim of trustworthiness is defensible, proportionate, and honest for the audience to whom it is made**.

That is why good assurance avoids absolute language. It does not say “this system is safe.” It says, explicitly:

- *for whom* the system is considered safe enough,
- *under what conditions* that judgement holds,
- *what risks remain*, and
- *what would cause that confidence to be revisited*.

This framing becomes especially important once we recognise that **not all stakeholders are able to consent to, interrogate, or even be aware of the assurance arguments being made.**

9.4. Why Consent Cannot Substitute for Assurance

In many modern socio-technical systems such as autonomous vehicles, urban sensing platforms, algorithmic decision systems, or large-scale digital services, it is sometimes assumed that confidence can be grounded in *user consent*. A purchaser may choose to buy an autonomous vehicle. An operator may choose to deploy a system. A customer may accept terms of service.

However, assurance cannot stop at explicit users.

There are often **bystanders and indirectly impacted parties** who have no practical opportunity to consent to the risks, trade-offs, or assumptions embedded in the system’s design and operation. Pedestrians sharing the road with

autonomous vehicles, individuals captured incidentally by surveillance systems, or communities affected by infrastructure automation do not meaningfully participate in procurement decisions, technical reviews, or certification processes.

For these stakeholders, confidence is not established through access to evidence or formal consent. It is established or lost through observed behaviour, perceived legitimacy, and whether systems fail in ways that respect human safety, dignity, and fairness.

This is where the socio-technical nature of assurance becomes unavoidable. Claims that may be acceptable to engineers, regulators, or customers can still fail socially if they do not account for those who are affected without agency. A system can be technically justified and procedurally compliant yet still erode trust when bystanders experience harm or risk that was never articulated in the assurance narrative. This reinforces the need for assurance claims that explicitly account for non-consenting affected parties, consistent with the role-aware Item Definition discussed in Case Study: Autonomous Vehicles and Urban Blackout and the foreseeability analysis in

Reasonable Foreseeability.

Good assurance therefore cannot be scoped solely around contractual relationships or explicit users. It must also confront a deeper limitation: **in many modern systems, meaningful consent may not be possible at all**, even where formal mechanisms appear to exist.

In theory, consent is often treated as a mechanism for transferring risk acceptance. Users are presented with a Terms of Service, a privacy notice, or an explicit “Accept” button, and this interaction is assumed to legitimise downstream consequences. In practice, this assumption is increasingly untenable.

Modern socio-technical systems, particularly those incorporating machine learning, large-scale data aggregation, or adaptive behaviour, are often complex enough that even their designers struggle to explain all aspects of behaviour, interactions, and emergent effects. If the engineers and architects closest to the system cannot fully characterise its behaviour under all conditions, it is unreasonable to expect end users to do so.

Meaningful consent requires more than a formal opportunity to agree. It presupposes:

- that the subject can reasonably understand what they are agreeing to,
- that the risks and trade-offs are explainable in accessible terms,
- and that refusal is a genuine option rather than a practical impossibility.

These preconditions are frequently not met.

Terms of Service documents that run to tens or hundreds of thousands of words are a familiar illustration. While users may technically be given the opportunity to read them, it is widely recognised, including in legal and regulatory contexts, that it is not reasonable to assume that an average user has read, understood, or meaningfully consented to such material. A single click on “Accept” cannot plausibly be treated as informed acceptance of complex, evolving socio-technical risk.

This limitation becomes even more acute when systems affect non-users and bystanders.

Pedestrians sharing space with autonomous vehicles, individuals captured incidentally in data-driven systems, or communities affected by infrastructure automation are not parties to any contract at all. For them, neither consent nor refusal is meaningfully available.

From an assurance perspective, this matters because **consent cannot be used as a substitute for justification**. Where understanding is infeasible, assurance must carry more of the burden. Claims about acceptability must be defensible on their own terms, grounded in principles, reasonable foreseeability, and explicit treatment of harm, not deferred to formal agreement mechanisms that do not reflect real comprehension or agency.

This limitation is not merely philosophical. Both **data protection law and consumer protection law explicitly reject the idea that formal agreement mechanisms imply meaningful consent**.

Under the GDPR, consent must be *freely given, specific, informed, and unambiguous* (Article 4(11)) [6]. Regulators have clarified that consent obtained through overly complex, opaque, or legalistic mechanisms does not meet this standard (GDPR Recitals 32 and 42; EDPB Guidelines on Consent [29]). Where system behaviour cannot be reasonably explained, even by its designers, the preconditions for informed consent are not satisfied.

Consumer protection law reaches similar conclusions from a different direction. EU and UK consumer law recognises that acceptance of lengthy, complex standard-form terms does not imply meaningful agreement where material behaviour or risk cannot reasonably be understood by the average user. Excessive length, opacity, and imbalance undermine fairness and enforceability.

Cookie consent is the most heavily worked example, and it is worth being precise about what actually changed. The legal test did not. Consent has had to be freely given, specific, informed and unambiguous since the GDPR applied, and refusal has had to be possible with the same degree of simplicity as acceptance since CNIL's recommendation of September 2020 [30]. What changed is the granularity with which regulators examined the interface that presents it. The EDPB's Cookie Banner Taskforce concluded in January 2023 that a banner offering no refusal option on the layer carrying the accept button cannot produce valid consent, and that a reject link rendered at minimal contrast is defective for the same reason [31]. The EDPB's guidelines on deceptive design patterns,

revised the following month, named the interface patterns that undermine a free choice, among them *continuous prompting*, where users are asked repeatedly until they give in [32]. The ICO and the CMA reached a compatible conclusion from the consumer-protection direction in August 2023 [33]. In December 2024 CNIL issued formal notices to publishers whose reject option was a low-contrast link [34]. The ICO's guidance on storage and access technologies, finalised in April 2026, requires that users can refuse as easily as they can accept [35].

The most recent movement runs in a different direction again, and it is the more telling one. The European Commission's Digital Omnibus proposal of November 2025 would move terminal-equipment consent into the GDPR and oblige controllers to respect a machine-readable expression of the user's choice, on the stated reasoning that a regulatory answer to consent fatigue and the proliferation of banners is long overdue [36]. The EDPB and EDPS strongly supported that aim, while objecting to other parts of the same package [37]. It remains a proposal. The direction of travel is away from adjudicating the design of individual banners and towards removing the per-site interaction altogether, which is a considered admission that more than a decade of interface scrutiny did not reliably produce meaningful consent.

That is the pattern this paper has been describing, in a domain where it has been tested to exhaustion. Throughout, systems remained formally compliant: banners were present, consent was recorded, audits were passed. What was absent was any defensible claim that the consent obtained meant what it was taken to mean. That is the shape of

Case Study: Smart Motorways, When Compliance Survives and Assurance Fails. It is worth keeping distinct from the earlier case of the urban blackout, which was not a compliance failure at all: each component met its specification, and what fell short was the scope of the reasoning rather than the discipline of following a rule. Compliance-led thinking is one route into an assurance failure. It is not the only one, and treating every failure as though it were would be its own kind of narrow reasoning.

This reinforces the socio-technical argument made throughout: trust in complex systems cannot be reduced to contractual artefacts or procedural acknowledgements. It must be earned through transparent reasoning about behaviour, impact, and limitation, including for those who cannot meaningfully consent yet are nonetheless affected.

That is the minimum standard for trust in complex socio-technical systems.

10. Conclusion: Assurance, Pressure, and the Cost of Systemic Failure

This paper has argued that compliance and assurance are not interchangeable. Compliance defines enforceable baselines; assurance is the structured reasoning that justifies trust in the presence of uncertainty, scale, and change. Treating compliance as a substitute for assurance creates an illusion of control that becomes increasingly fragile as systems integrate into society.

A core takeaway follows directly from this distinction: the greater the potential impact of a system, service, or organisation, the less sufficient compliance-led reasoning becomes, and the more essential assurance-led reasoning is. As scale, coupling, and societal dependence increase, the cost of unexamined assumptions, proxy evidence, and retrospective standards rises sharply. For low-impact systems, baseline compliance may bound risk acceptably. For high-impact systems, it cannot.

Naively, this gradient points first to Critical National Infrastructure (CNI): energy, transport, communications, health, and essential digital platforms, where failure modes propagate rapidly across society and the economy. In these contexts, reliance on baseline compliance as the primary source of confidence is structurally inadequate. Assurance intensity must scale with impact: more explicit claims, stronger interrogation of assumptions, rigorous treatment of interfaces and supply chains, and continuous renewal of confidence under change. Compliance remains necessary, but it is a floor. Assurance determines whether that floor is remotely sufficient.

Returning to the spatial metaphor introduced in Compliance as an Input to Assurance, compliance regimes form a convex hull around acceptable practice: rigid, bounded, and necessarily retrospective. As systems scale, interconnect, and accumulate novel use, misuse, and dependency, pressure builds inside the broader assurance space. New risks do not disappear simply because they fall outside existing standards. They press outward, concentrating stress at interfaces, assumptions, and organisational boundaries.

When that pressure is not relieved through explicit assurance reasoning, it does not dissipate harmlessly. It accumulates. And when failure occurs, the hull does not deform gently: it tears.

We have already seen what those tears look like.

Cyber incidents routinely demonstrate how a technically local compromise propagates into socially and economically damaging cascades across supply chains: halted production, disrupted logistics, energy shortages, delayed healthcare, and loss of public confidence. In many such cases, each individual organisation involved was compliant with its obligations. What failed was not control execution, but system-level reasoning about dependency and consequence.

Similarly, large digital platforms and social media systems have operated within evolving regulatory baselines while still producing well-documented harms to users, particularly children and adolescents, through scale effects, incentive misalignment, and malicious use of intended functionality. Compliance did not prevent these outcomes because the harms emerged in the spaces between standards: in behavioural feedback loops, amplification dynamics, and socio-technical interactions that were not explicitly reasoned about at design time.

In safety-critical infrastructure, such as smart motorways or autonomous transport systems, we have seen how systems can behave exactly as specified, and still fail society. Safe fallback behaviours, minimum-risk manoeuvres, or conservative shutdown logic may satisfy safety requirements in isolation yet become city- or network-scale denial-of-service events when combined with loss of connectivity, power, or human coordination. These outcomes were not the result of negligence or non-compliance, but of assurance gaps between lenses, domains, and responsibilities.

These failures matter because they are not merely technical. They undermine societal resilience, economic stability, and sovereign capability. Energy systems, transport networks, communications platforms, and increasingly autonomous services are now foundational to daily life. In such contexts, the cost of failure is borne by the public, not just by the organisations that were audited or certified.

Compliance alone cannot contain these cascades because it is typically scoped to organisational boundaries rather than consequence boundaries; to artefact verification rather than dependency reasoning; and to individual responsibility rather than system-of-systems interaction.

A supplier may be compliant. An integrator may have performed due diligence. Contracts may have been satisfied. Yet harm still propagates across the very links that made the system valuable in the first place. This is not a failure of regulation in intent, but a limitation of compliance as a conceptual model of trust.

Principles-based assurance offers a way to relieve pressure before rupture occurs. By anchoring reasoning in what must be true for trust to be justified, and by making claims, arguments and evidence explicit, it enables organisations, regulators, and authorities to interrogate assumptions, anticipate cascade paths, and adapt as context changes. It provides a way to reason about novel threats, scale effects, and misuse before they manifest as crisis.

This is not an argument against standards, certification, or enforcement. It is an argument against mistaking them for the outer boundary of responsibility.

10.1. Assurance Is Not a Brake on Innovation

Engineering is a discipline of constraint optimisation. Assurance is what makes the constraints explicit.

It is also important to be explicit about what this position is not. Calls for principles-based assurance are sometimes mischaracterised as anti-innovation, or as an attempt to constrain technical progress through abstract oversight. This narrative is familiar from parts of the “big tech” playbook, where regulation is framed as an external brake on creativity. That framing is misleading.

Innovation does not occur in a vacuum. In a vacuum, development tends toward pure technology push: capabilities are built because they are possible, and return on investment is sought after the fact. Engineering, by contrast, is a discipline of constraint optimisation. Meaningful innovation emerges when practitioners work within and across multiple constraints: safety, security, privacy, usability, resilience, cost, scale, social acceptability, and regulatory obligation.

For impactful systems, these constraints are not incidental; they are defining. Innovation occurs not by ignoring them or attempting to blow them up, but by discovering new strategies, architectures, trade-offs, or operating models that satisfy them simultaneously in ways that were previously thought infeasible. This is as true in transport, energy, and infrastructure as it is in digital platforms and autonomous systems.

Principles-based assurance does not suppress innovation. It creates the conditions for responsible innovation by making constraints explicit, surfacing trade-offs early, and enabling reasoned disagreement about what must be optimised, accepted, or deferred. It provides a shared language in which regulators, engineers, customers, and society can articulate what matters and why without collapsing immediately into prescriptive control or retrospective blame.

In that sense, assurance is not opposed to innovation. It is the mechanism by which innovation remains aligned with purpose, trust, and societal legitimacy as systems scale and their consequences become harder to contain.

Compliance can stabilise systems.

Assurance is what prevents them from failing catastrophically when the world changes.

The choice is not whether modern systems will be stressed: they will be.

The choice is whether we notice, reason, and adapt before that stress becomes socially and economically damaging.

Works Cited

[1] National Cyber Security Centre, “White paper: The future of NCSC Technology Assurance,” National Cyber Security Centre, 2021.

[2] International Organization for Standardization and SAE International, “Road vehicles — Cybersecurity engineering,” ISO/SAE, 2021.

[3] Pacific Gas and Electric Company, “Mission Substation Outage Event, 20 December 2025 — De-Energization Community Event report,” Pacific Gas and Electric Company, 2025.

[4] The San Francisco Standard, “Waymo apologises to city officials over stalled cars during December blackout,” 3 March 2026. [Online]. Available: https://sfstandard.com/2026/03/03/waymo-san-francisco-december-blackout-stalled-cars-hearing/. [Accessed 17 August 2026].

[5] Axios, “Waymo outage stalled emergency response,” 9 January 2026. [Online]. Available: https://www.axios.com/local/san-francisco/2026/01/09/waymo-outage-stalled-response-delay. [Accessed 17 August 2026].

[6] European Union, “Regulation (EU) 2016/679 General Data Protection Regulation,” Official Journal of the European Union, 2016.

[7] International Organization for Standardization, “Road vehicles — Functional safety,” ISO, 2018.

[8] International Organization for Standardization, “Road vehicles — Safety of the intended functionality,” ISO, 2022.

[9] House of Commons Transport Committee, “Rollout and safety of smart motorways,” House of Commons, 2021.

[10] Department for Transport, “All new smart motorways scrapped,” 2023. [Online]. Available: https://www.gov.uk/government/news/all-new-smart-motorways-scrapped. [Accessed 17 August 2026].

[11] National Highways, “National Emergency Area Retrofit,” [Online]. Available: https://nationalhighways.co.uk/roads-and-travel/road-projects/smart-motorways-evidence-stocktake/national-emergency-area-retrofit/. [Accessed 17 August 2026].

[12] European Union, “Regulation (EU) 2024/2847 of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements (Cyber Resilience Act),” Official Journal of the European Union, 2024.

[13] United Nations Economic Commission for Europe, “UN Regulation No. 155 — Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system,” UNECE, 2021.

[14] National Cyber Security Centre, “Technology assurance — Principles: Product development,” [Online]. Available: https://www.ncsc.gov.uk/collection/technology-assurance/principles-product-development. [Accessed 17 August 2026].

[15] National Cyber Security Centre, “Technology assurance — Principles: Product design and functionality,” [Online]. Available: https://www.ncsc.gov.uk/collection/technology-assurance/principles-product-design-and-functionality. [Accessed 17 August 2026].

[16] National Cyber Security Centre, “Technology assurance — Principles: Through-life,” [Online]. Available: https://www.ncsc.gov.uk/collection/technology-assurance/principles-through-life. [Accessed 17 August 2026].

[17] National Cyber Security Centre, “Principles Based Assurance (PBA),” [Online]. Available: https://www.ncsc.gov.uk/information/principles-based-assurance. [Accessed 17 August 2026].

[18] Assurance Case Working Group, “Goal Structuring Notation Community Standard,” Safety-Critical Systems Club.

[19] National Protective Security Authority, “CAE: Blocks and Connection Rules,” National Protective Security Authority, 2022.

[20] Adelard LLP, “CAE Mini-Guide 4: CAE Connection Rules,” Adelard LLP, 2020.

[21] N. Chozos and R. Bloomfield, “CAE Mini-Guide 5: CAE Building Blocks,” Adelard LLP, 2020.

[22] R. E. Bloomfield and K. Netkachova, “Building Blocks for Assurance Cases,” in *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Naples, Italy, 2014.

[23] European Union, “Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS2),” Official Journal of the European Union, 2022.

[24] Health and Safety Executive, “ALARP at a glance,” [Online]. Available: https://www.hse.gov.uk/enforce/expert/alarpglance.htm. [Accessed 17 August 2026].

[25] *Edwards v National Coal Board,* 1949.

[26] Nuclear Institute, “The Application of ALARP to Radiological Risk,” Nuclear Institute.

[27] United States Nuclear Regulatory Commission, “10 CFR 20.1003 — Definitions (ALARA),” US Nuclear Regulatory Commission.

[28] MISRA, “MISRA C:2023 — Guidelines for the use of the C language in critical systems,” The MISRA Consortium Limited, 2023.

[29] European Data Protection Board, “Guidelines 05/2020 on consent under Regulation 2016/679,” European Data Protection Board, 2020.

[30] Commission Nationale de l'Informatique et des Libertes, “Deliberation n 2020-092 du 17 septembre 2020 portant adoption d'une recommandation proposant des modalites pratiques de mise en conformite en cas de recours aux cookies et autres traceurs,” CNIL, 2020.

[31] European Data Protection Board, “Report of the work undertaken by the Cookie Banner Taskforce,” European Data Protection Board, 2023.

[32] European Data Protection Board, “Guidelines 03/2022 on deceptive design patterns in social media platform interfaces,” European Data Protection Board, 2023.

[33] Information Commissioner's Office and Competition and Markets Authority, “Harmful design in digital markets: how online choice architecture practices can undermine consumer choice and control over personal information,” Digital Regulation Cooperation Forum, 2023.

[34] Commission Nationale de l'Informatique et des Libertes, “Dark patterns in cookie banners: the CNIL issues formal notices to website publishers,” CNIL, 2024.

[35] Information Commissioner's Office, “Guidance on the use of storage and access technologies,” [Online]. Available: https://ico.org.uk/for-organisations/direct-marketing-and-privacy-and-electronic-communications/guidance-on-the-use-of-storage-and-access-technologies/. [Accessed 17 August 2026].

[36] European Commission, “Proposal for a Regulation amending Regulations (EU) 2016/679 and others as regards the simplification of the digital legislative framework (Digital Omnibus), COM(2025) 837 final,” European Commission, 2025.

[37] European Data Protection Board and European Data Protection Supervisor, “Joint Opinion 2/2026 on the Proposal for a Regulation as regards the simplification of the digital legislative framework (Digital Omnibus),” EDPB, 2026.