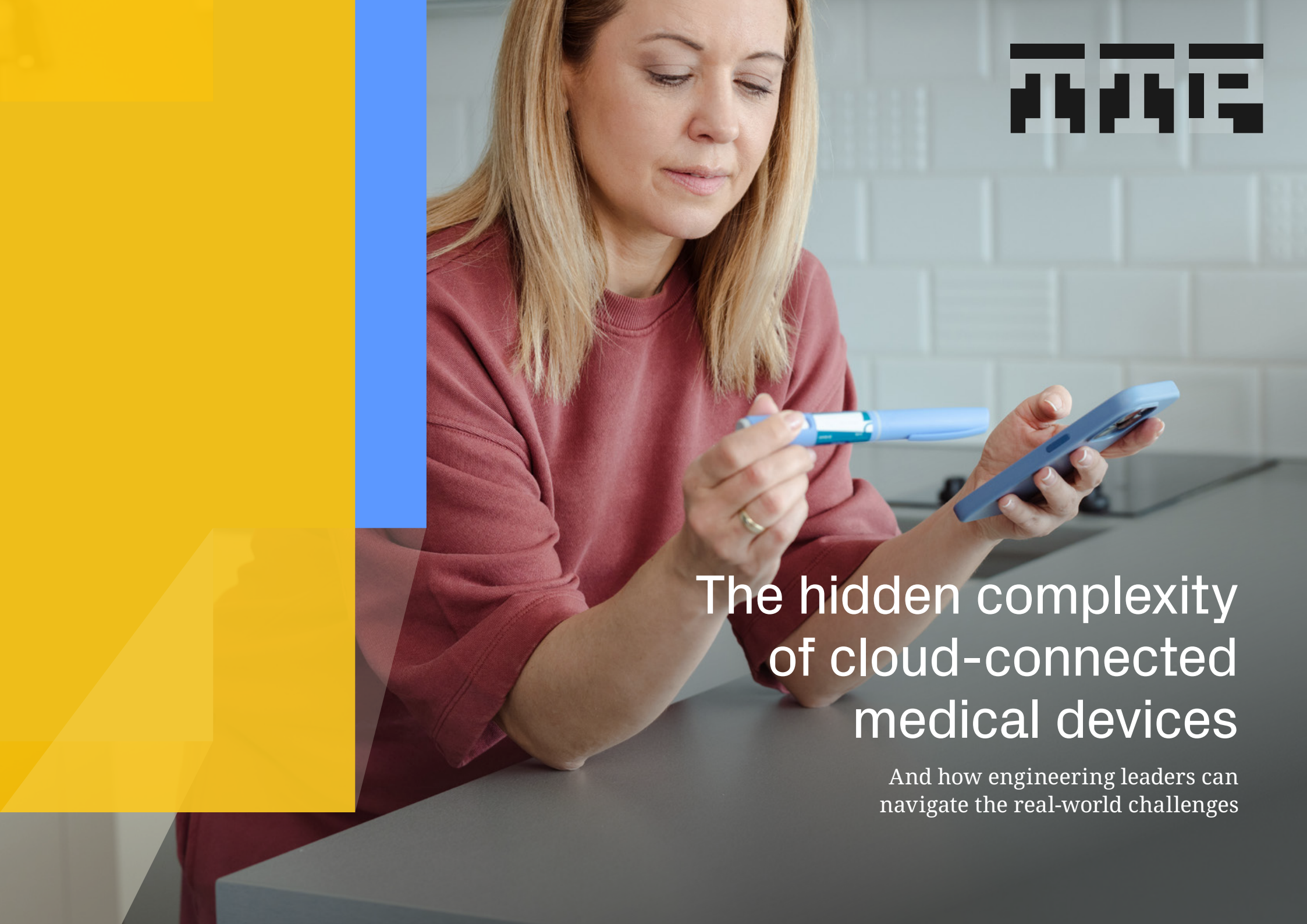


The hidden complexity of cloud-connected medical devices

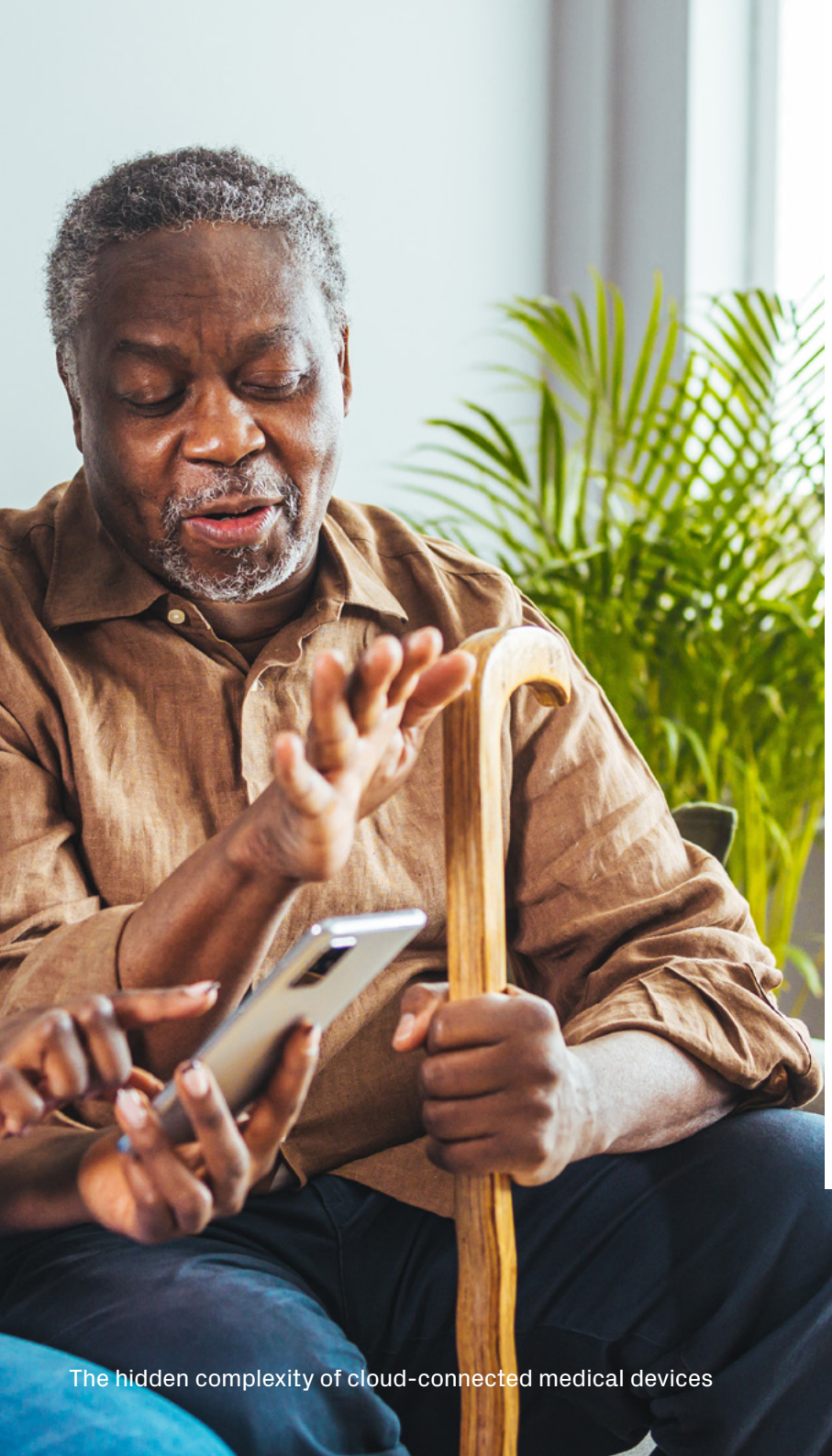
And how engineering leaders can navigate the real-world challenges

Executive summary

Cloud-connected medical devices promise significant value: better uptime, richer insights, smoother maintenance, and new digital services. Yet, teams often underestimate how difficult it is to turn that promise into a reliable and valuable regulated product. The underlying challenge is not simply technical complexity but a collision of worlds: safety-critical device engineering and fast-moving cloud software. When these domains meet, assumptions break, risks multiply, and projects slow down.

Many organisations begin with the idea that connecting a device is a natural extension of their existing development processes. What they discover – often late – is that cloud-connected devices behave like ecosystems, not products. Device and cloud teams frequently work from incompatible mental models. Cloud features are added late and struggle to align with real workflows. Security and lifecycle obligations expand far beyond initial expectations. The result can be delays, avoidable redesigns, regulatory friction, and products that fail to deliver the intended value.

This paper outlines why cloud-connected medical devices are more complex than they appear and explains how to approach them in a way that avoids common pitfalls. It draws on real-world project experience and aligns with the expectations of modern regulatory frameworks in the US and EU. Organisations that succeed do so by approaching the device, cloud, operations, security, and lifecycle as one coherent system. They design for real clinical and operational needs. And they treat firmware and software as long-term assets that evolve safely, not prototypes that happen to ship.



Contents

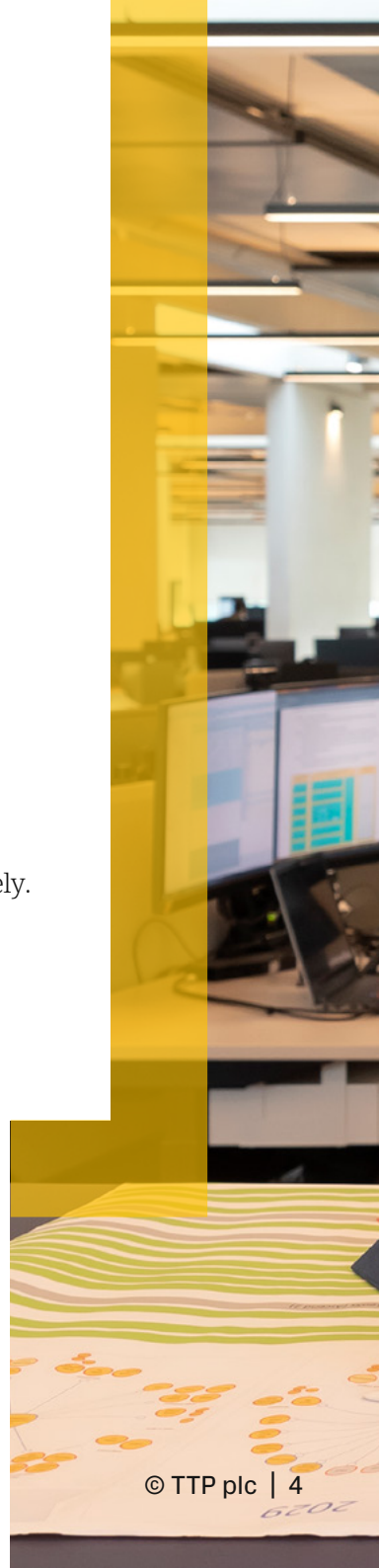
The challenge of cloud-connected medical devices

Medical device companies increasingly face pressure to offer connected, data-enabled systems. Fleet management, predictive maintenance, remote diagnostics, and digital services all rely on devices that communicate reliably with cloud infrastructure. For teams with strong device engineering backgrounds, this can appear straightforward: add connectivity, store data, and expose insights. But connectivity transforms a device into part of a continuously evolving distributed system, whose behaviour depends on environments the manufacturer does not control.

Regulatory expectations are also changing. In the United States, the FDA's guidance "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions" makes cybersecurity inseparable from safety, and requires manufacturers to demonstrate secure design, provide Software Bill of Materials (SBOMs), and show robust processes for updates and vulnerability handling. In the EU, the Medical Device Regulation (MDR) mandates general safety and performance requirements for software, and MDCG guidance further clarifies how clinical evaluation applies to medical-device software, including components delivered via cloud infrastructure.

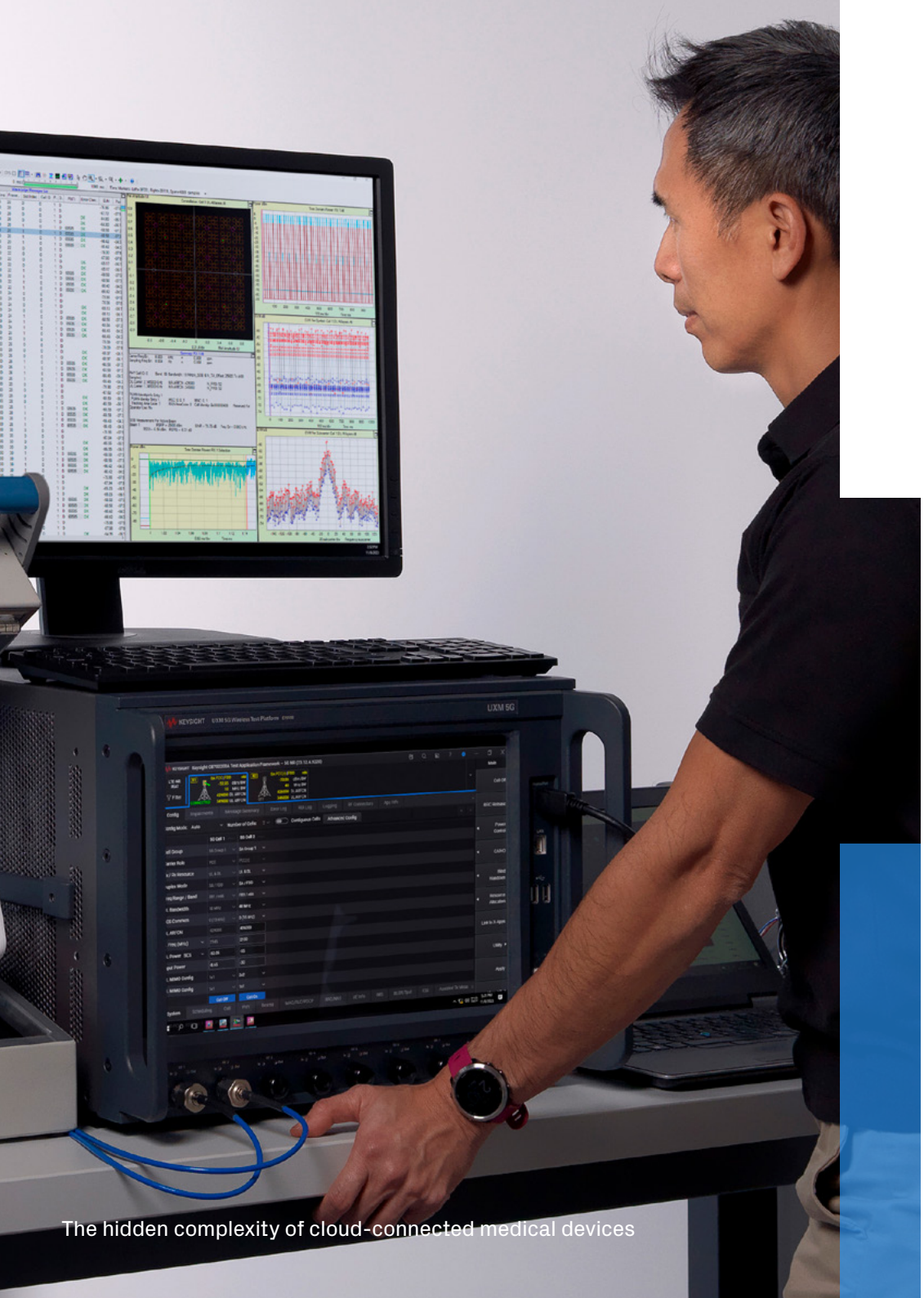
Additional horizontal standards such as EN/IEC 82304-1 (health software product safety) and IEC 81001-5-1 (security processes for health software) set expectations for lifecycle governance, secure development, and operational safety. These apply not just to devices but to the broader system in which they operate.

Regulated products – primarily class II and III – that integrate embedded firmware, cloud services, secure connectivity and long-term operational commitments are challenging. By following the principles outlined in this paper, they can be delivered effectively.



Real-world challenges and failure modes

Despite the promise of connected systems, many programmes struggle in similar ways once development moves beyond early prototypes. The challenges are rarely caused by a single technical failure. Instead, they emerge from misaligned assumptions, late-stage trade-offs, and gaps between how systems are designed and how they are actually used and maintained. The following sections describe common failure modes observed in real-world projects.



Fragmented assumptions between device and cloud teams

Device engineers and cloud engineers often begin with fundamentally different assumptions. Device teams expect deterministic behaviour, constrained resources, and the need to validate systems under fixed configurations. Cloud teams expect elasticity, continuous deployment, and infrastructure that evolves weekly. When these assumptions meet, subtle integration failures appear: devices assume the cloud is always reachable; cloud systems assume devices behave like browsers or mobile apps. These mismatches, if not addressed early, lead to integration delays and brittle behaviour in real deployments.

Engineering pressure pushes data value to the backseat

Device development programmes are demanding. Safety-critical firmware, manufacturing readiness, hardware optimisation, and regulatory documentation consume significant effort. In this environment, cloud and data features are often added late. But without early investment, cloud components struggle to align with the device's capabilities or the customer's workflows. Teams may meet technical milestones but fail to realise the value that connectivity was supposed to deliver.

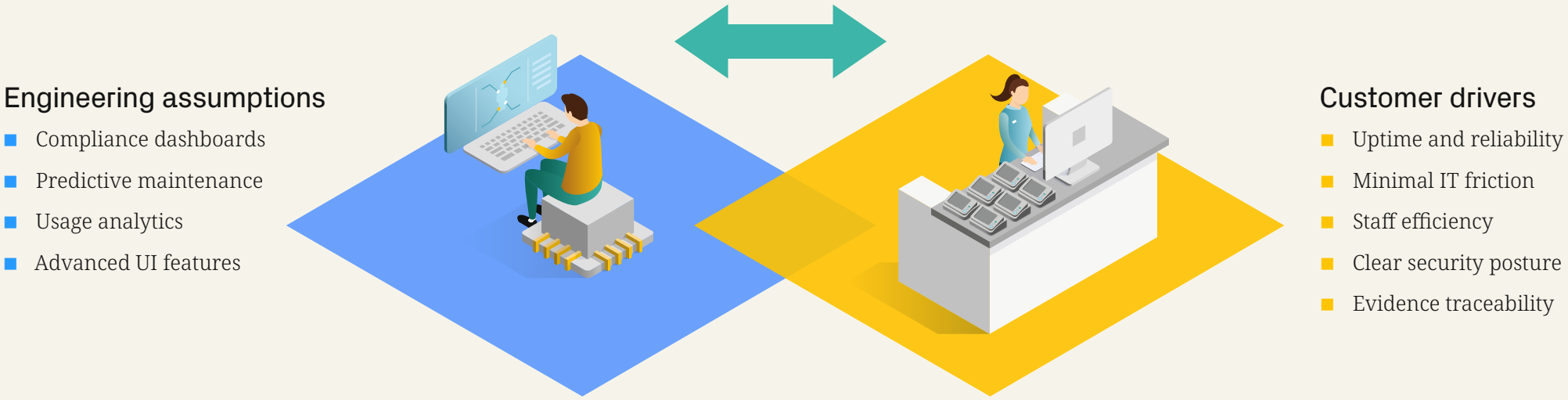


Figure 1: The mismatch between internal engineering expectations and actual hospital or laboratory purchasing drivers.

Customer value propositions for cloud often miss the mark

Internally, connectivity is frequently justified through abstract value propositions: compliance dashboards, predictive maintenance, usage analytics. But actual buyers – hospitals, labs, clinicians – prioritise reliability, minimal IT burden, security assurances, and workflow integration. Staffing costs are particularly severe, with buyers under strong pressure to minimise these. If cloud features introduce friction or cost, or if the value does not translate into real operational benefit, adoption stalls. As a result, devices can be technically impressive but commercially unsuccessful.

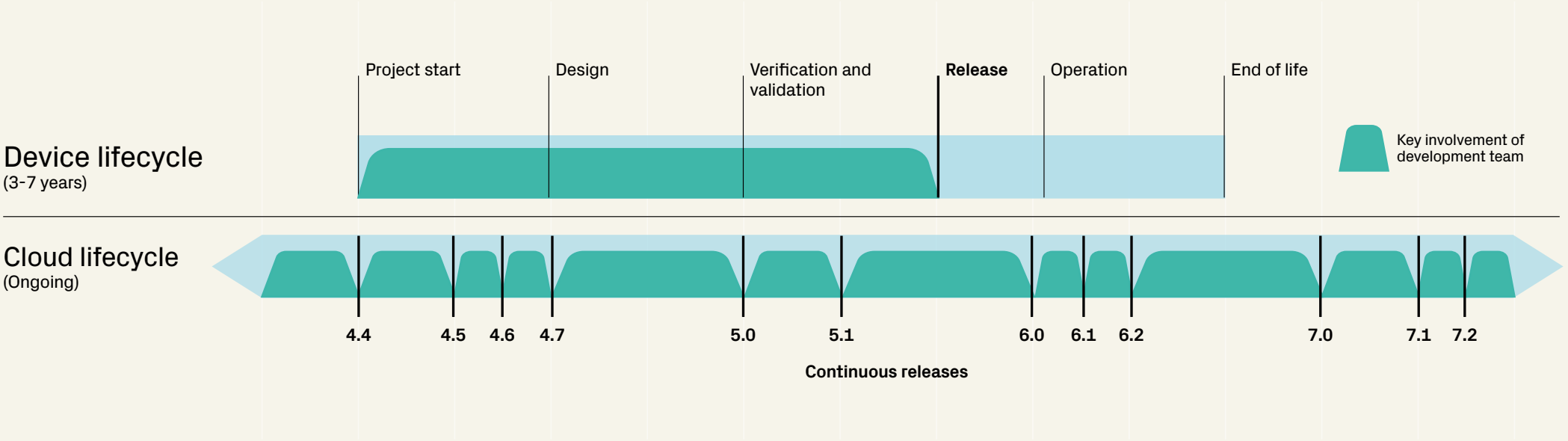


Figure 2: A visual comparison of the long medical-device lifecycle versus the fast iteration cadence of cloud services.

Rapidly evolving cloud and cybersecurity expectations clash with device verification and validation cycles

Modern regulatory expectations treat cybersecurity as a continuous process, not a one-time deliverable. The FDA's cybersecurity guidance requires manufacturers to demonstrate ongoing management of threats, vulnerabilities, and updates. IEC 81001-5-1 reinforces this with requirements for secure development, configuration management, and operational oversight. Medical devices, however, follow long validation cycles and must maintain evidence of safety for years.

When cloud platforms evolve every few months, manufacturers face a widening gap between regulatory expectations and the operational reality of maintaining a distributed system. Operations directors, faced with the reality of using connected devices for critical uses, need assurances that regular updates won't compromise safety and that devices will continue to work as expected.

Why common approaches often fall short

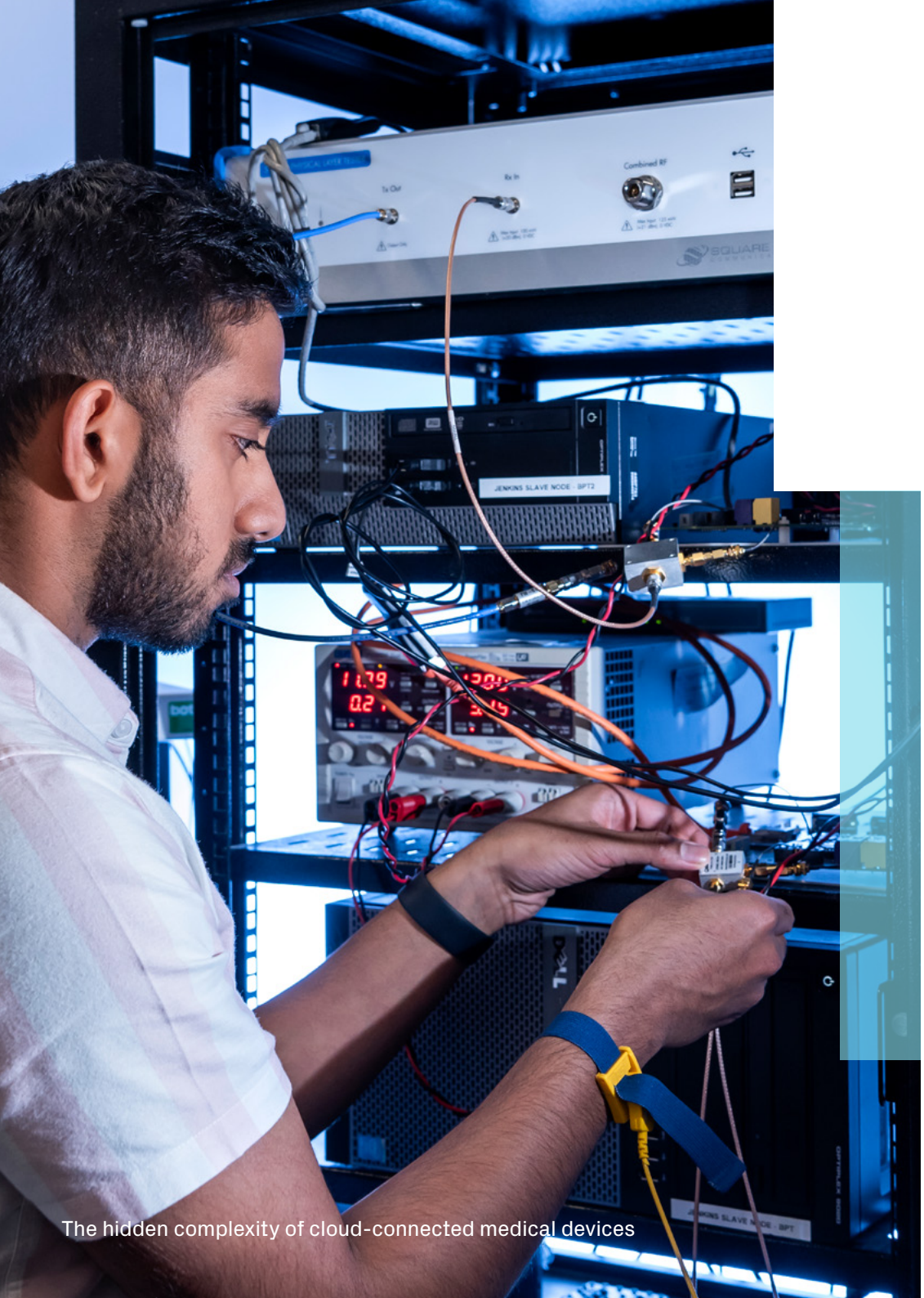
Many organisations encounter these challenges not because they lack technical capability, but because they apply development models that were never designed for cloud-connected, regulated systems. Common approaches reflect assumptions that work well for standalone devices or enterprise software, but break down when safety, regulation, and long-term operation intersect.

Treating connectivity as a late bolt-on

When connectivity is added late in development, it tends to inherit whatever technical and architectural constraints have already been set for the device. Interfaces become fragile, cloud features remain underspecified, and the system lacks the documentation and traceability required for modern regulatory submissions. The FDA's 2023 guidance "Content of Premarket Submissions for Device Software Functions" expands expectations for software documentation even for standalone software. For connected systems, these expectations extend further: update strategies, security assurances, failure handling, and architectures must all be accounted for. When connectivity is bolted on, these elements rarely arrive in time.

Treating cloud as an optional extra

Some organisations treat cloud functionality as a secondary feature that follows after the "real engineering" is done. Yet connectivity fundamentally changes the nature of the device and how value is delivered. Without integrating cloud requirements into early design, teams struggle to articulate a value proposition that resonates with customers. Features that do not directly support clinical or operational needs, or that disrupt established workflows, are unlikely to gain traction - even if they demonstrate technical sophistication. The result is cloud systems that work acceptably in isolation but fail to integrate into the larger product strategy.



Device-centric mindset that undervalues system complexity

Device-focused engineering teams may see the physical device as the centrepiece of value creation. In reality, cloud-connected systems shift value toward data, analytics, and long-term operational services. When teams prioritise the device above the system, they risk designing architectures that are difficult to update, maintain, or secure. The device may operate flawlessly in isolation but falter when deployed into the messy reality of hospital networks, variable connectivity, and diverse clinical workflows.

Cloud-centric mindset that ignores real-world constraints

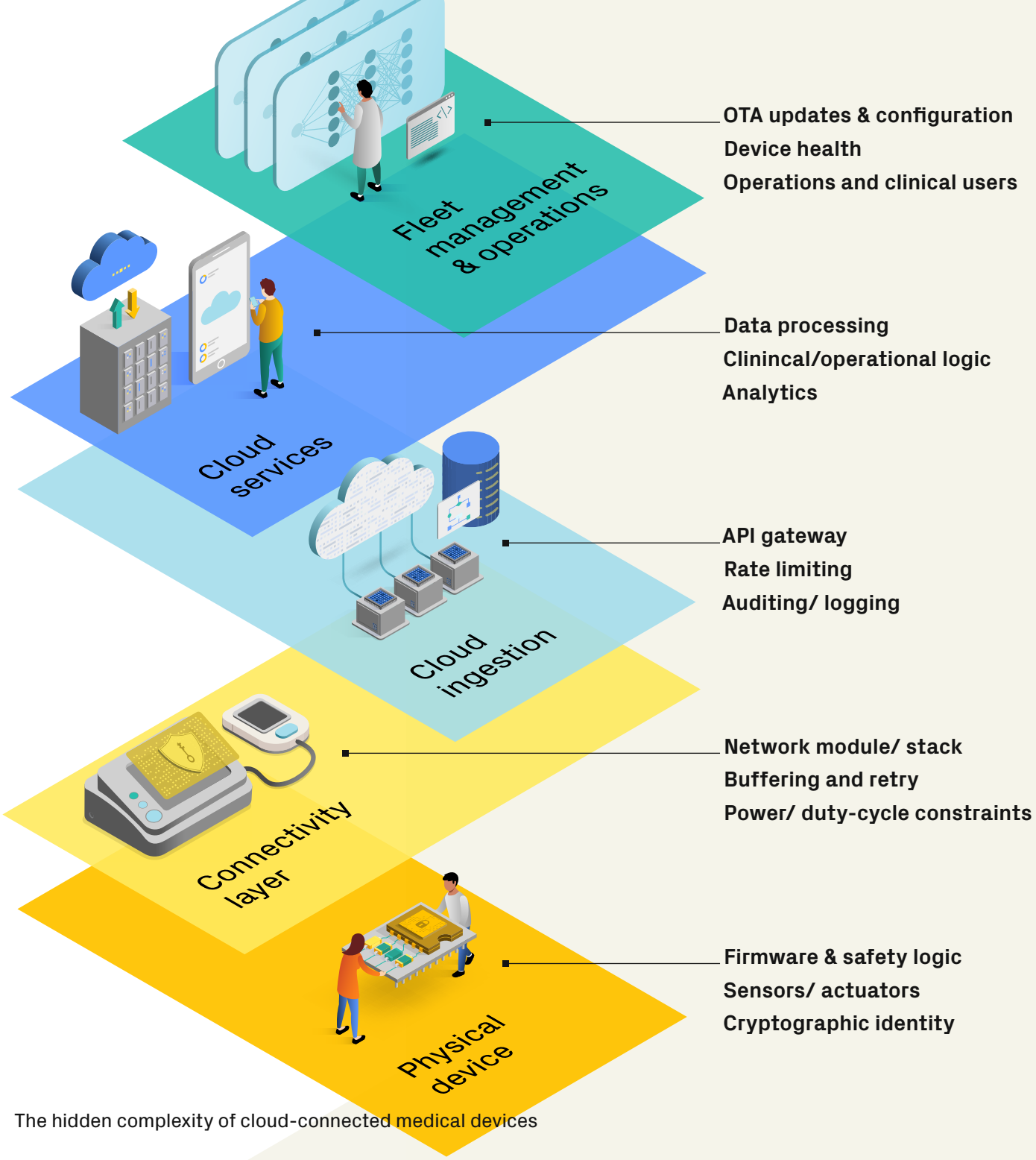
Cloud engineers often think in terms of rapid iteration, stateless services, and elastic scaling. Medical devices do not behave this way. They experience intermittent connectivity, have fixed resources, and must maintain safety-critical behaviour regardless of cloud conditions. Cloud teams may assume that updates can be deployed continuously or that devices can rely on high-bandwidth, low-latency connections. These assumptions break down in real-world deployments. When these assumptions go unchallenged, systems may satisfy technical feasibility but fail to meet the clinical, operational, and risk expectations required by frameworks such as IMDRF's SaMD guidance.

Key principles for success

Programmes that succeed take a different approach from the outset. Rather than treating connectivity as an add-on, they recognise cloud-connected devices as complex, regulated systems and design accordingly. The principles below reflect practices that consistently help teams deliver reliable, compliant, and valuable connected products.

Think broad: system-of-systems architecture and early prototyping

Connected systems behave as ecosystems rather than discrete products. Firmware, connectivity, cloud ingestion, processing, fleet management, IT infrastructure, and clinical workflows all interact to determine system performance. Viewing the system holistically uncovers dependencies that are easy to overlook when components are designed in isolation. For example, assumptions about network reliability shape how the device must buffer and prioritise data; assumptions about clinical workflow shape when data must be transmitted or retrieved.



Early, end-to-end prototyping is one of the most effective ways to expose risks. Building minimal paths through the system allows teams to observe real behaviour under real constraints – including offline operation, jittery networks, authentication edge cases, and update workflows. These prototypes reveal integration challenges before architectural commitments become costly or irreversible.

Figure 3: How firmware, connectivity, cloud ingestion, cloud services, fleet management, IT infrastructure and clinical workflows form one integrated system.

Design for how devices will be used – not just how they are built

Clinical and operational environments create the constraints within which the device must succeed. Staff are busy, IT teams are stretched, and security expectations are non-negotiable. Connectivity must earn its place by reducing workload, simplifying processes, and preventing downtime. When connectivity helps automate routine checks, streamline configuration, and detect faults early, it becomes a clear operational asset. Cloud features that complicate tasks or add overhead are quickly rejected.

Engaging early with hospital IT and clinical users helps ensure that cloud features are grounded in real-world needs. It also aligns with MDCG guidance, which emphasises demonstrating clinical value and fitness-for-purpose for software-based functions, including cloud services.

Treat software and firmware as long-term assets with governance

Medical-device software is never truly "finished." Libraries evolve, cloud services deprecate APIs, vulnerabilities emerge, and customers expect updates. Meeting these expectations safely requires disciplined software governance. Configuration control, change management, and traceability must extend from initial requirements through long-term maintenance. A clear update strategy, covering development, testing, rollout, rollback, and evidence generation becomes essential.

SBOM and supply-chain processes help manage third-party risk, ensuring that vulnerabilities are detected early and handled consistently. Devices must remain verifiable when their dependencies change. Aligning with IEC 81001-5-1 helps teams establish lifecycle practices that maintain safety and compliance over time.

Build security and evidence lifecycle into architecture

Security is not a feature – it is a continuous process that touches every stage of the device lifecycle. It begins with secure provisioning and identity assignment during manufacturing. It continues through mechanisms such as secure boot, authenticated updates, and monitored deployment. Effective security architectures anticipate the need for staged rollouts, rollback mechanisms, and real-time monitoring of anomalies.

Regulators increasingly expect not only secure behaviour but also a traceable account of how security is maintained. Threat modelling, SBOM maintenance, incident response processes, and vulnerability-handling records all contribute to a defensible regulatory submission. Designing the system so that this evidence is produced naturally reduces overhead and improves trust with customers and auditors.

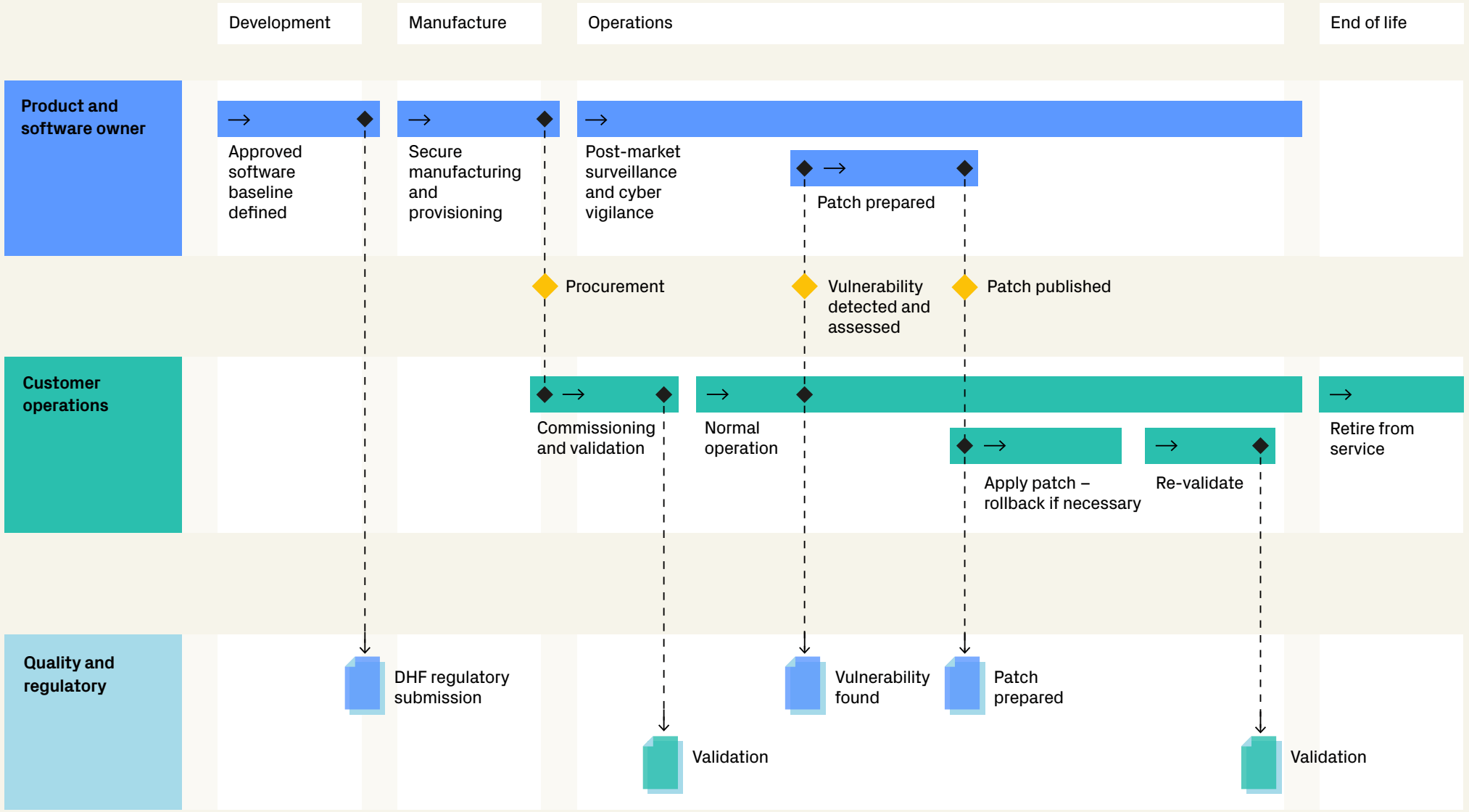


Figure 4: High-level swimlane diagram showing provisioning, signing, verification, deployment, rollback and evidence creation.

Risks, trade-offs, and mitigations

Architectural trade-offs

Developing a cloud-connected medical device inevitably requires navigating competing priorities. Architectural decisions must balance structure and flexibility. A rigid architecture can simplify documentation, validation, and long-term governance, but may limit the ability to incorporate new workflows, security requirements, or data-driven features later in the product's life. Conversely, architectures that remain highly flexible can accelerate development and experimentation but risk creating ambiguity during verification or introducing pathways that are difficult to validate under regulatory expectations. There's a need to balance the architecture between areas that are rigid and verifiable, and areas that are designed to flex with evolving cloud practices, services and security. Safety, and operational assurances, depend on Operations customers being able to trust that evolving software loads won't compromise operations.

Iteration speed versus regulatory evidence

Teams also face tension between the faster iteration cycles of modern cloud platforms and the slower, evidence-heavy processes of regulated medical-device software. Rapid deployment of cloud features can bring clear benefits but may introduce inconsistencies in documented behaviour or raise questions during regulatory review if the supporting evidence is incomplete. Balancing iteration speed with robust documentation and traceability is essential to maintaining both agility and compliance.

Security trade-offs

Security creates its own set of trade-offs. Strong protections are vital, but if implemented without considering real operational contexts, they may add friction or even disrupt clinical workflows. Successful teams approach these tensions with a systems mindset: by considering the device, cloud, users, and regulatory landscape together, they can make deliberate choices rather than reactive compromises. Early identification and explicit discussion of these trade-offs helps prevent costly late-stage redesigns and ensures the final system performs reliably in the real world.

Mitigating risk in practice

Mitigating these risks depends on broad collaboration and clear lifecycle intent. Multidisciplinary architecture reviews – bringing together device, cloud, QA/RA, and operations teams – help ensure assumptions are aligned and that all parts of the system can be validated consistently. Full-stack prototyping provides early insight into integration challenges and user impact, long before final design decisions are locked in. Lifecycle and configuration-management processes aligned to standards such as IEC 81001-5-1 strengthen governance and reduce long-term risk. SBOM processes and supply-chain monitoring help manage vulnerability exposure, while regular engagement with hospital IT and security teams ensures the device aligns with operational expectations and constraints.

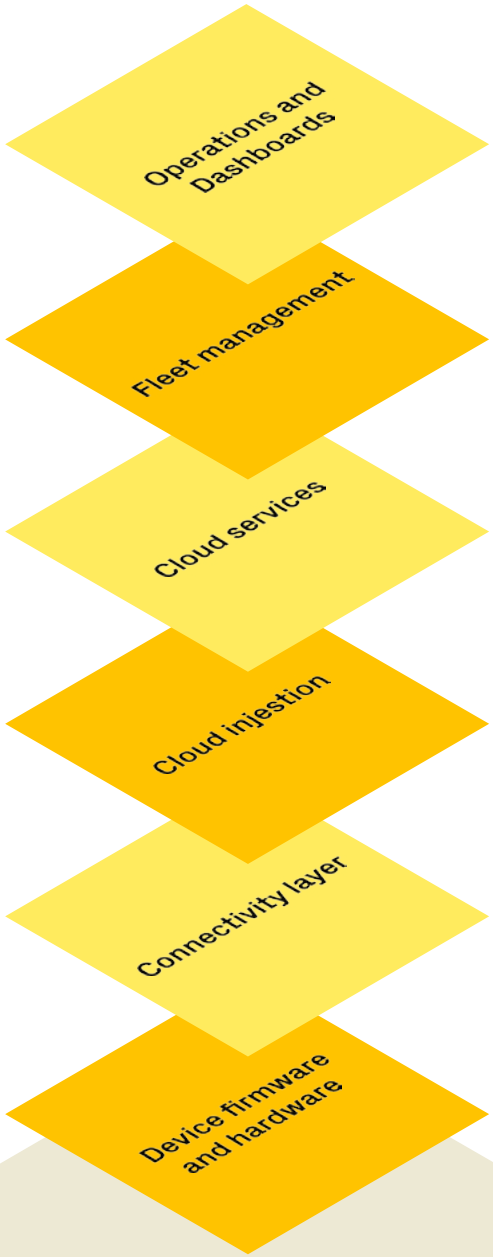
6 key mitigations

- Make the core trade-offs explicit early (structure vs flexibility, speed vs evidence, security vs usability, where & how safety is delivered).
- Define “non-negotiables” up front (regulatory needs, safety, security baselines) and what can evolve.
- Treat documentation and traceability as product features, not afterthoughts.
- Validate the system end-to-end early (device–cloud–user workflow), not just components in isolation.
- Design security around real clinical operations to avoid adding avoidable workflow friction.
- Establish lifecycle ownership (change control, configuration, SBOM/supply chain) from the start.

Example architecture and update pathway

A connected medical device architecture typically comprises several interdependent layers. At the foundation lies the device layer, where real-time firmware and safety logic enforce deterministic behaviour and ensure patient safety. Secure boot mechanisms and cryptographic identity anchor the device's trustworthiness. Above this sits the connectivity layer, responsible for handling unreliable networks, buffering data during outages, managing retries, and ensuring that clinical information is delivered reliably even when conditions are suboptimal.

Cloud ingestion services authenticate devices, enforce rate limits, and implement auditing to maintain system integrity. The cloud services layer processes incoming data, applies clinical or operational algorithms, and generates the insights required by clinicians, service teams, or regulatory evidence processes. Device fleets are supported by dedicated management services that monitor health, orchestrate secure updates, and manage configuration at scale. Operational dashboards then give IT staff, service engineers, and clinical users visibility into the state of devices, enabling better decision-making and more proactive support.



A reliable update pathway overlays this architecture. Devices are provisioned at manufacture with secure identities and cryptographic material. Updates are produced through controlled pipelines, signed to establish provenance, and deployed in stages to reduce risk. Devices verify updates before installation and can revert safely if necessary. Each update event must be integrated into the organisation's risk-management and clinical-evaluation processes, especially when changes could affect safety, intended use, or system performance. A clear and well-documented update lifecycle ensures that the device evolves safely over time while preserving the integrity of regulatory submissions.

Figure 5: A simplified layer view showing firmware, connectivity, cloud ingestion, cloud services, fleet management and operational layers.

Self-assessment checklist

The challenges and trade-offs described above often emerge late, when they are costly to address. The questions below summarise the core considerations that consistently shape successful cloud-connected medical device programmes. Use this list to evaluate whether your organisation is addressing the core challenges of connected-device development:

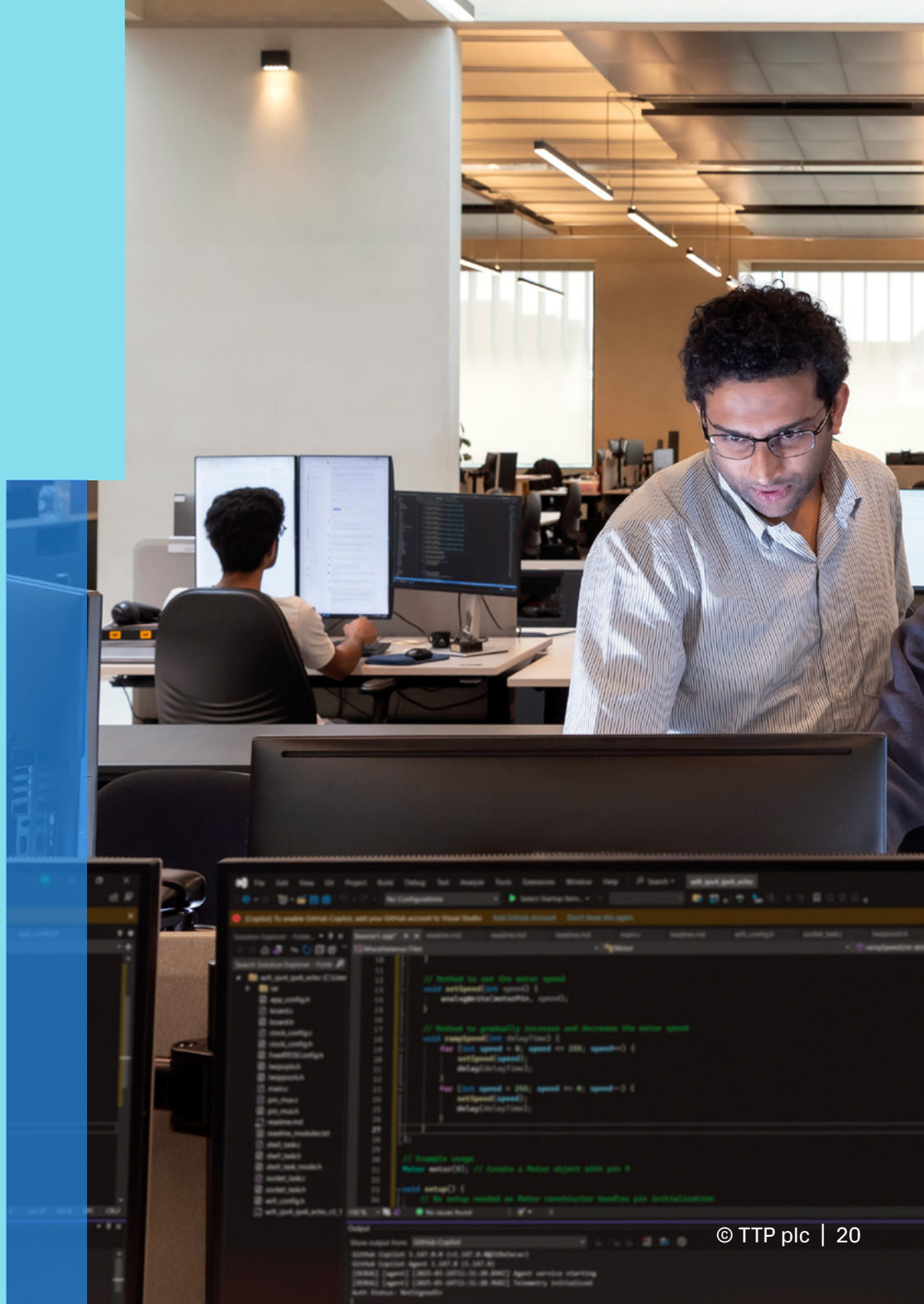
- Have you modelled the device + cloud + operations as a single system rather than separate silos?
- Have you prototyped complete end-to-end workflows under realistic conditions (e.g. flaky networks, offline periods, manufacturing provisioning)?
- Do your proposed cloud features map to explicit clinical or operational value, consistent with MDR/MDCG expectations for clinical evaluation of medical-device software?
- Does your design incorporate SBOMs, update strategies and software documentation at the level expected by FDA's "Content of Premarket Submissions for Device Software Functions"?
- Are your security processes and lifecycle controls aligned to FDA cybersecurity guidance and IEC 81001-5-1?

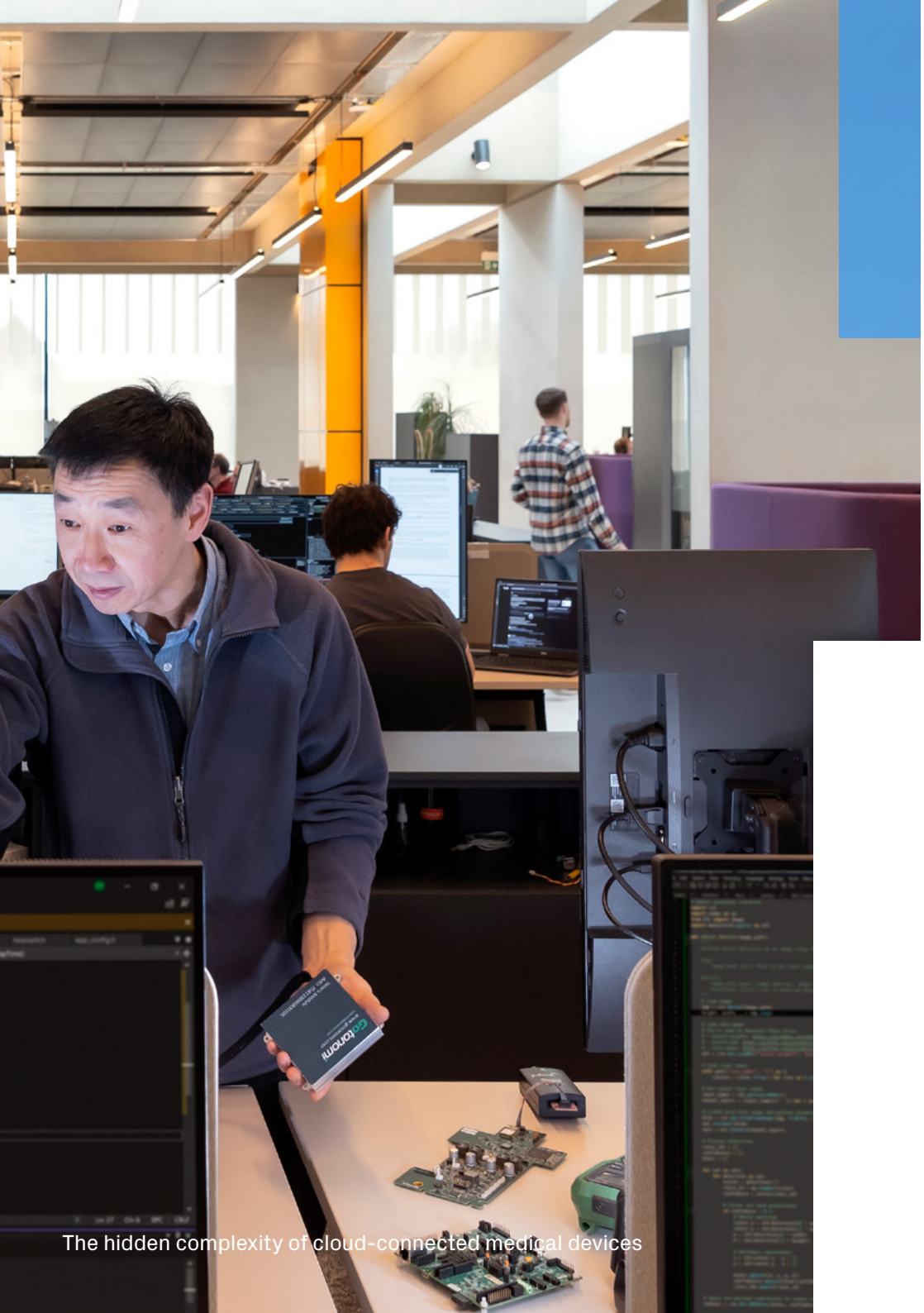
Conclusion

Connected medical devices introduce challenges that are easy to underestimate. The complexity does not arise from connectivity itself but from the need to integrate safety-critical systems with fast-moving cloud infrastructure, cybersecurity expectations, operational workflows, and long-term maintenance obligations. Success depends on treating the entire system – device, cloud, operations, security, and lifecycle – as a coherent whole rather than a set of components.

Teams that succeed do three things consistently. They treat the entire product – device, cloud, operations, security, and maintenance – as one integrated system. They align design decisions with real operational and clinical value, as expected by MDR, FDA, and IMDRF SaMD guidance. And they embed lifecycle governance and security into their architecture from the outset, ensuring their processes align with expectations from FDA cybersecurity guidance, FDA device-software submissions requirements, EN/IEC 82304-1, and IEC 81001-5-1.

Delivering such systems is demanding, but the organisations that approach them with a strong system architecture, clear lifecycle intent, and close engagement with clinical and operational users consistently outperform those that treat connectivity as an add-on. When designed with foresight and discipline, cloud-connected devices deliver durable technical, clinical, and commercial advantage.





The hidden complexity of cloud-connected medical devices

How TTP can help

Cloud-connected medical devices demand joined-up thinking across embedded systems, cloud software, cyber security and regulatory governance. If your internal bandwidth is stretched, or you need specific expertise to de-risk key architectural decisions, TTP can integrate alongside your team and provide targeted, senior engineering support where it matters most.

We help startups establish scalable, regulator-ready architectures from the outset, and support established manufacturers modernising platforms without disrupting validated products.

With experience across device engineering, secure cloud systems and lifecycle management, we design connected medical technology that is robust, updateable and built for long-term operation in real clinical environments.

About TTP's Software team

Our software architects and developers have the broad expertise to help you capture and evolve sophisticated software products, putting you in control of how, when, and who develops them.

We can help across the product lifecycle from early concept through to new product introduction, across risk scales up to safety-critical, and anywhere from handling specific engineering problems through to full product development responsibility. TTP consultants support you throughout your software development lifecycle.

References

1. FDA – Cybersecurity in Medical Devices: Quality Management System Considerations and Content of Premarket Submissions
2. EU – Regulation (EU) 2017/745 on medical devices (MDR) and associated implementing guidance.
3. FDA – Content of Premarket Submissions for Device Software Functions.
4. IMDRF / FDA – Software as a Medical Device (SaMD): Clinical Evaluation (IMDRF/SaMD WG/N41FINAL:2017, adopted by FDA).
5. MDCG 2020-1 – Guidance on Clinical Evaluation (MDR) / Performance Evaluation (IVDR) of Medical Device Software.
6. EN 82304-1:2017 / IEC 82304-1:2016 – Health software – Part 1: General requirements for product safety.
7. IEC 81001-5-1:2021 – Health software and health IT systems safety, effectiveness and security – Part 5-1: Security

About the author

Dan Talmage

Dan brings over 25 years' experience in end-to-end complex system product development. As a leader, systems engineer and project manager, Dan has broad technical expertise across multiple fields, and a track record of successful delivery of the largest scale projects in the Life Sciences and Consumer sectors.

Dan's expertise covers technical project management and leadership, systems engineering, medical and regulatory, software and firmware and signal processing.

He has developed PCR instruments, blood gas analysis, rail safety and food packaging systems, and has over 60 smartphone type approvals.

Get in touch at ttp.com/software

TTP plc
TTP Campus, Cambridge Road,
SG8 6HQ
+44 1763 262626

ttp.com