Writing Samples

Charlo Mouton

Healthcare IT & compliance · Web3 & fintech technical writing

1. Post-Quantum Security, Explained for People Who Sign the Checks
2. Building for More Than One Rulebook: Compliance Design in Web3 and Fintech
3. What Home Health and Hospice Agencies Actually Need From Software

Post-Quantum Security, Explained for People Who Sign the Checks

A plain-English briefing for business leaders

By Charlo Mouton

Most security problems announce themselves. A breach shows up in the logs, a customer calls, a regulator sends a letter. The quantum problem is different. The damage can start years before anyone notices, and by the time it is obvious, the data is already gone.

This piece explains what the risk is, what has already changed, and what a sensible business should do about it in the next 12 to 24 months. No physics degree required.

The short version

Almost everything your company does online is protected by two kinds of math. One kind keeps data secret while it moves between systems (encryption and key exchange). The other kind proves who sent something and that it was not altered (digital signatures). Today, most of that work is done by algorithms called RSA and elliptic-curve cryptography.

Those algorithms are safe against today's computers. They are not expected to be safe against a large, working quantum computer. Nobody can tell you the exact year one will exist. But the people responsible for setting security standards decided they could not wait to find out, and they have already published the replacements.

Why "not yet" is not the same as "not a problem"

The risk most leaders miss is called **harvest now, decrypt later**.

An attacker does not need a quantum computer today to cause you trouble later. They only need to copy your encrypted traffic or files now and store them. Storage is cheap. When the decryption capability arrives, they open the archive.

So the real question is not "when will quantum computers arrive?" It is "how long does my data need to stay private?"

- A marketing email: a few days.
- A customer's financial history, a health record, a merger file, a long-term contract: ten years or more.

If the answer for some of your data is "longer than the time until a capable quantum computer exists," that data is already exposed in a practical sense. That is the whole

argument for acting early.

Signatures have a related problem. If a signature scheme can be broken, someone could forge approvals, software updates, or financial instructions that look legitimate. Anything that relies on long-lived signing keys, such as firmware, identity systems, or blockchain accounts, needs a plan.

What has already changed

This is not a forecast. In August 2024, the U.S. National Institute of Standards and Technology (NIST) published its first three post-quantum cryptography standards:

- **ML-KEM** (FIPS 203), for establishing shared encryption keys. This replaces the key-exchange role RSA and elliptic curves play today.
- **ML-DSA** (FIPS 204), a general-purpose digital signature scheme.
- **SLH-DSA** (FIPS 205), a signature scheme built on hash functions. It is slower and produces larger signatures, but it rests on very conservative assumptions, which makes it a good backup.

NIST has also selected an additional key-establishment algorithm (HQC) as a backup to ML-KEM, and its draft transition guidance proposes phasing out today's vulnerable algorithms over the next decade, with deprecation around 2030 and disallowance around 2035 for many uses. Large vendors are already shipping hybrid modes in browsers, messaging apps, and cloud services.

In other words, the replacement parts exist. The work now is installing them.

What this means in business terms

You do not need to understand lattice math to manage this risk, the same way you do not need to understand metallurgy to replace the locks on a building. You need to know four things.

1. Where your cryptography lives. Most companies cannot answer this today. Encryption is buried in vendor products, libraries, VPNs, payment gateways, identity providers, and backup systems. The first project is an inventory: what systems encrypt or sign data, with which algorithms, and who controls the upgrade.

2. Which data has a long shelf life. Rank your data by how long it must stay confidential. Health, financial, legal, and identity data usually sits at the top. That ranking tells you where to start.

3. Who you depend on. For most businesses, the bulk of the migration happens inside vendors' products. Ask your critical vendors a direct question: "What is your post-quantum roadmap, and which NIST algorithms are you adopting?" A vague answer is useful information.

4. How quickly you can swap algorithms. Security teams call this *crypto-agility*. A system that hard-codes one algorithm deep in its design is expensive to fix. A system that treats the algorithm as a replaceable part can be upgraded with a configuration change and a test cycle. Agility is the real goal, because the standards will keep evolving.

A practical 12-to-24-month plan

1. **Name an owner.** Someone in security or IT should own the quantum transition, with a short quarterly report to leadership.
2. **Build the inventory.** Start with internet-facing systems, identity, payments, and anywhere long-lived data is stored.
3. **Prioritize by data lifespan.** Protect the long-shelf-life data first.
4. **Use hybrid modes where available.** Hybrid key exchange combines a classical algorithm with ML-KEM, so you are protected as long as either one holds. It is the cautious middle path most major vendors have taken.
5. **Put it in contracts.** Add post-quantum readiness questions to vendor reviews and renewal terms.
6. **Test before you need to.** Post-quantum keys and signatures are larger than today's. That can affect bandwidth, storage, hardware tokens, and older devices. Find those limits in a pilot, not during an emergency.

Being honest about what "quantum-ready" means

A word of caution about marketing language. Using a NIST-standardized algorithm is not the same as being certified. Formal validation (for example, FIPS 140-3 module validation) is a separate, lengthy process. And a system can use post-quantum algorithms in one layer while still relying on classical cryptography in another.

I have seen this up close. At Galaxia, the ecosystem I founded, we built post-quantum tooling using ML-KEM, ML-DSA, and SLH-DSA at the application layer through an open-source library, running on a public testnet. We are also clear in our own documentation that this implementation is not FIPS-validated and that on-chain signatures on the underlying network remain classical. Saying that plainly costs nothing and earns trust. Buyers should expect the same honesty from every vendor.

When someone tells you a product is "quantum-proof," ask three questions:

- Which algorithms, and in which part of the system?
- Is the implementation validated, or just built on the standard?
- What still depends on classical cryptography?

The bottom line

Post-quantum security is a migration, not a product you buy once. The standards are published, the deadlines are taking shape, and the biggest risk, data stolen today and read later, is already in play. Companies that start with an inventory, prioritize long-lived data, and push their vendors for specific answers will handle this calmly. Companies that wait for a headline will be doing it in a hurry.

Charlo Mouton is a writer and the founder of Galaxia, a Web3, fintech, and AI ecosystem with post-quantum and compliance tooling on public testnet. He writes about security, compliance, and healthcare operations for business readers.

Building for More Than One Rulebook: Compliance Design in Web3 and Fintech

How to design a product that can live under MiCA in Europe and a patchwork of rules in the U.S.

By Charlo Mouton

Every founder in digital assets eventually hits the same wall. The product works. Users like it. Then someone asks, "Can we launch this in Europe?" or "What happens when a customer in New York signs up?" and the honest answer is a long pause.

The usual response is to bolt compliance on at the end: a KYC vendor here, a geo-block there, a legal memo in a shared drive. That approach works until the second jurisdiction. After that, every new market means rewriting logic that was never meant to change.

There is a better way to build, and it starts by treating regulation as a design input rather than a launch checklist.

Two very different rulebooks

The European Union now has a single framework for crypto-assets: the Markets in Crypto-Assets Regulation, known as MiCA (Regulation (EU) 2023/1114). Its stablecoin rules took effect in mid-2024 and the rest applied from 30 December 2024. MiCA sorts the world into a few categories, including asset-referenced tokens (ARTs), e-money tokens (EMTs), and crypto-asset service providers (CASPs), and attaches requirements to each: authorization, white papers, reserve rules, governance, conduct, and disclosure. A CASP authorized in one member state can passport its services across the EU.

Alongside MiCA, the EU's recast Transfer of Funds Regulation applies the "travel rule" to crypto transfers. Providers must collect and pass along originator and beneficiary information, in line with FATF Recommendation 16.

The United States has no single equivalent. A typical product touches several layers at once:

- **Federal anti-money-laundering rules.** Many crypto businesses register with FinCEN as money services businesses and must run AML programs, file reports, and follow the federal travel rule.
- **State licensing.** Money transmission is licensed state by state, and some states add their own regimes, such as New York's BitLicense.

- **Securities and commodities questions.** Whether a token or product falls under the SEC, the CFTC, or neither depends on facts and continues to be debated in courts and in Congress.
- **Stablecoin rules.** The GENIUS Act, signed in July 2025, created the first federal framework for payment stablecoins, with reserve, redemption, and supervision requirements.

The EU gives you one big rulebook. The U.S. gives you many smaller ones that overlap. A product designed for only one of these worlds will struggle in the other.

Principle 1: Separate the rules from the code that runs them

The most expensive compliance mistake is hard-coding a jurisdiction's rules into business logic. "If user is in Germany, cap transfers at X" works once. Repeat it for 30 countries and 50 states, and nobody can say with confidence what the system actually does.

A cleaner design separates three things:

1. **Jurisdiction profiles.** Structured data describing what each market requires: identity checks, transaction thresholds, disclosure text, reporting duties, restricted products.
2. **A rules engine.** A component that reads the user's context, applies the matching profile, and returns a decision with reasons.
3. **The product.** Wallets, payments, issuance, and trading call the engine instead of carrying their own copy of the rules.

When a regulation changes, you update a profile and run your tests. You do not hunt through a dozen services looking for every place someone wrote "if EU."

Principle 2: Every decision should leave a receipt

Regulators, auditors, and banking partners do not ask whether you have a policy. They ask you to prove that the policy ran, for this customer, on this date. Systems built for more than one jurisdiction need a record of each compliance decision: which profile applied, which checks ran, what the result was, and which version of the rules was in force.

This is one place blockchain infrastructure can help. Anchoring a hash of a decision record on a public ledger creates a timestamp that cannot be quietly edited later. Personal data stays off-chain; only a fingerprint goes on it.

Principle 3: Collect the minimum, keep it close

Multi-jurisdiction products face a privacy problem. Travel-rule and KYC obligations require collecting personal data, while GDPR in Europe and a growing set of U.S. state privacy laws restrict how you store and move it. Design for data minimization from the start: collect what the strictest applicable rule needs, store it in the region it belongs to, and pass attestations ("this user is verified") between systems instead of raw documents wherever the rules allow.

Principle 4: Keep the engine and the license separate in your head

Software can enforce rules. It cannot hold a license. A CASP authorization under MiCA, a state money transmitter license, or an exemption opinion from counsel comes from regulators and lawyers, not from code.

This distinction matters in how you describe your product. "Built to support MiCA requirements" or "MiCA-aligned" describes design. "MiCA-compliant" or "authorized" describes a legal status that only a regulator can grant. Mixing them up is the fastest way to lose a bank partner's trust.

At Galaxia, the ecosystem I founded, we built a compliance engine on public testnet that carries rule profiles for 13 regional frameworks plus the FATF travel rule, and asset-issuance modules designed around MiCA's ART, EMT, and CASP categories alongside U.S. and UK rule sets. Every report the engine produces is explicitly marked as not certified and not authorized. That label is a feature. It tells partners exactly what the software does and what it does not claim to be.

Principle 5: Plan for the rules you do not have yet

Regulation in this space is still moving. U.S. market-structure legislation, stablecoin implementing rules, and EU technical standards will keep changing the details. A system built on profiles, versioned rules, and auditable decisions absorbs those changes. A system built on scattered if-statements breaks a little more each time.

Practical steps for teams starting now:

- **Map your products to categories first.** Before writing code, decide which MiCA category and which U.S. regimes each product could fall into, and get counsel to pressure-test it.
- **Version your rules.** Every rule set should carry a version and an effective date, and every decision should record which version it used.

- **Test against scenarios, not just code paths.** Build test cases like "EU resident sends 1,500 euros to an unhosted wallet" and "Texas user redeems a stablecoin" and run them on every release.
- **Write disclosures with the same care as code.** White papers, risk statements, and marketing claims are regulated outputs under MiCA. Review them like production changes.
- **Choose your words carefully.** Describe what your system is designed to do, not legal outcomes you have not been granted.

The takeaway

Compliance in Web3 and fintech is not one problem. It is the same set of problems, identity, reporting, disclosure, and proof, expressed differently in every market. Teams that build the rules as data, record every decision, and stay precise about what their software can and cannot claim will enter new markets in weeks. Teams that hard-code their first jurisdiction will spend years undoing it.

Charlo Mouton is a writer and the founder of Galaxia, a Web3, fintech, and AI ecosystem whose compliance and identity tooling runs on public testnet. Nothing in this article is legal advice.

What Home Health and Hospice Agencies Actually Need From Software

A buyer's-eye view from the administration side of the office

By Charlo Mouton

Software vendors love to demo the dashboard. Agency administrators rarely care about the dashboard. What they care about is whether the claim goes out on time, whether the surveyor finds the signed order, and whether anyone can tell them, at 4:45 on a Friday, which visits are still missing documentation.

I spent more than a decade in home health and hospice administration: operations, billing, and compliance. This is a practical list of what agencies need from software, written for the people who build it and the people who buy it.

1. Deadlines the system tracks for you

Home health and hospice run on clocks. Many of them have real money attached.

- **Notice of Admission (home health).** Since 2022, Medicare home health agencies must submit a Notice of Admission within five calendar days of the start of care. File late, and payment is reduced for each day you missed.
- **Notice of Election (hospice).** Hospices face a similar five-day window after the election date, with consequences for late filing.
- **30-day billing periods.** Under the Patient-Driven Groupings Model (PDGM), home health billing runs in 30-day periods. Each period needs complete documentation before the final claim.
- **Signed orders and plans of care.** Physician signatures on plans of care and orders are a constant chase, and unsigned orders can hold up billing.
- **Assessment timing.** OASIS assessments, recertifications, and hospice recertifications all have defined windows.

Good software does not just store these dates. It surfaces them before they are missed, names the person responsible, and escalates when something slips. A spreadsheet of due dates is a record. A system that tells the intake coordinator on day three that the NOA has not gone out is a safeguard.

What to ask a vendor: "Show me what happens on day four when an NOA hasn't been filed. Who gets notified, and how?"

2. Billing that catches problems before the payer does

Most cash-flow problems in home health start upstream of billing. A visit wasn't documented. An order wasn't signed. Authorization for a non-Medicare payer expired. By the time a denial comes back, the work to fix it is three times harder.

Agencies need:

- **Pre-billing checks** that flag missing visit notes, unsigned orders, or documentation gaps before a claim is built.
- **Clear claim status** without logging into three different portals.
- **Denial tracking by reason**, so a pattern ("we keep getting denied for X with this payer") becomes visible in weeks, not quarters.
- **Payer-specific rules** for Medicaid managed care and commercial plans, which rarely match Medicare's.

What to ask a vendor: "What percentage of your customers' claims go out clean on the first pass, and how do you measure it?"

3. HIPAA safeguards that are built in, not bolted on

Every vendor will say "HIPAA" in the first five minutes. Buyers should push past the word.

HIPAA does not have a government certification. There is no such thing as being "HIPAA certified." What matters is whether the vendor will sign a **Business Associate Agreement (BAA)**, whether their own hosting and subcontractors will sign BAAs with them, and whether the product actually supports the Security Rule's safeguards.

In practice, look for:

- **Role-based access**, so an aide sees their own patients' visit details and not the entire census.
- **Audit logs** that record who viewed or changed a record and when, kept long enough to answer questions during an investigation.
- **Multi-factor authentication** and automatic session timeouts, especially on shared or field devices.
- **Encryption** of data in transit and at rest.
- **Secure messaging**, so staff stop texting patient details from personal phones.

What to ask a vendor: "Will you sign a BAA, and which of your subcontractors have signed BAAs with you?" Then ask to see the audit log for a test record.

4. Field tools for the people doing the visits

Clinicians and aides do their work in kitchens, cars, and driveways, often with poor signal. Software that assumes a desk and a good connection will be worked around, and workarounds are where documentation gaps begin.

Field staff need:

- Mobile documentation that works **offline** and syncs when service returns.
- **Electronic Visit Verification (EVV)** that meets state Medicaid requirements. The 21st Century Cures Act required EVV for Medicaid personal care and home health services, and states have implemented it in different ways.
- Visit notes that are quick to complete accurately, so charting does not spill into evenings.

What to ask a vendor: "Let me try your mobile app in airplane mode."

5. Survey readiness on demand

State surveys and accreditation visits can arrive with little notice. Administrators need to pull records, policies, training files, and evidence of corrective actions quickly.

Agencies need:

- **Personnel file tracking**: licenses, TB tests, competencies, in-service hours, with expiration alerts.
- **Quality and incident tracking**, tied to follow-up actions with owners and due dates.
- **A clean audit trail** showing who did what and when, so the agency can prove its process ran, not just that it exists.

What to ask a vendor: "If a surveyor asked for every record touched by a specific nurse in the last 60 days, how long would it take to produce?"

6. Back-office work that doesn't live in email

A surprising amount of agency operations still runs through inboxes, sticky notes, and shared spreadsheets: referral follow-ups, staffing changes, payroll questions, vendor invoices, and policy updates. These are not clinical tasks, but they decide whether the clinical work gets done.

The best systems give every task an owner, a deadline, and a home tied to the right record. That idea shaped Nebula, the back-office workspace I am building at Galaxia, now in early access. It is designed for office and staff operations and is built so patient information stays out of it.

7. Honest implementation and support

Switching systems is one of the riskiest things an agency does. Data migration, staff training, and parallel billing can strain a small office for months.

Buyers should ask for:

- A **written implementation plan** with named contacts.
- **References from agencies of similar size and service mix**, not just the vendor's largest customer.
- A clear answer on **data export**: if you leave, how do you get your records out, and in what format?

The bottom line

Agencies don't need more features. They need fewer missed deadlines, fewer denied claims, fewer surprises at survey time, and staff who are not charting at midnight. Vendors who build around those outcomes, and buyers who test for them during the demo, will both come out ahead.

Charlo Mouton is a writer and the founder of Galaxia. He writes about healthcare operations, compliance, and fintech for business readers. This article is general information, not legal, billing, or compliance advice. Check current CMS and state rules before acting.