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.