

Compliance by design helps founders build regulatory requirements into product workflows from the start.

Compliance by Design: What Independent Founders Can Learn From Building in Regulated Markets

By Muzammil Posted on September 30, 2026

For founders, freelancers, developers, and product teams, compliance by design means treating legal and regulatory requirements as product requirements before workflows, data models, and automation are built.

> ***Quick answer: What is compliance by design?*** *It is the practice of identifying regulatory obligations early and translating them into product requirements, evidence trails, identity controls, configurable rules, and update processes. In regulated software, compliance is not a final legal review. It is part of the architecture.*
>
> *Compliance is often treated as something that happens after a product has been designed. Build the software. Test it. Get customers. Then ask a lawyer whether everything complies.*

In a regulated market, that order creates problems. Compliance is not a final review. It is a product requirement.

When I founded CyberActive while attending law school, I entered a business where technology, education, and regulation were inseparable. As the company moved traffic school, driver education, and other regulated educational programs into online environments, we could not simply reproduce a classroom experience on a website. The technology had to satisfy rules established by states, courts, agencies, schools, and other institutions.

That changed the way the product had to be designed. Questions that might look like legal questions from the outside quickly became software questions: How do you verify who completed a course? What information must be recorded? When is a student officially finished? Which records need to be preserved? Who receives completion information? What happens when one jurisdiction requires a process that another does not?

For independent developers, consultants, designers, product managers, and founders working with regulated businesses, understanding this connection between regulation and product design can prevent expensive redesigns, incomplete scopes, and difficult client relationships.

1. Why should regulatory requirements be studied before the workflow is designed?

One of the most expensive mistakes in regulated technology is designing the ideal user experience first and reviewing the rules later.

A product team might create a fast onboarding process only to discover that the client must collect additional documentation. A developer might remove steps that appear unnecessary, only to learn that an agency requires them. A freelancer might automate an approval process without realizing that a qualified person must review certain decisions.

The better approach is to identify regulatory requirements during discovery. The U.S. Small Business Administration notes that licensing and permit requirements vary based on business activity, location, and government rules. The same principle matters for technology projects: a workflow that works for one client or jurisdiction may not satisfy another.

Official reference: U.S. Small Business Administration, licenses and permits

Before building, ask the client:

- Which agencies or organizations regulate this activity?
- Which licenses or approvals affect the product?
- Are there state-specific or local requirements?
- Which actions require documentation or record retention?
- Does a human need to approve any part of the process?
- Are there required notices, disclosures, forms, or completion rules?

> ***The goal is not for the freelancer to become the client's lawyer. The goal is to understand which legal requirements have technical consequences while applying practical freelancing legal tips when working with clients and regulated projects.***

2. How do you turn regulatory language into product requirements?

Regulations are rarely written like software specifications. They do not normally say, "Create this database field" or "Display this notification after step four." Someone has to translate the obligation into a functional workflow.

Imagine a rule requiring a business to document that a user completed a particular process. That single requirement creates a chain of product questions:

- What qualifies as completion?
- Which events need timestamps?
- Does the system need to preserve an audit trail?
- Does the user receive a certificate or confirmation?
- Does an administrator need access to the record?
- Does the record need to be transmitted somewhere else?
- How long must it remain accessible?

A broad compliance requirement can affect database architecture, user interfaces, reporting tools, permissions, notifications, and administrative dashboards. If an obligation affects how the product works, it should appear in the product requirements themselves, not only in a legal memorandum that the engineering team never sees.



Compliance by design translates regulatory obligations into practical product requirements.

3. Why must regulated software be designed for evidence, not only functionality?

Most software teams focus on whether an action works. Regulated businesses often also need to prove that the action happened. There is an important difference.

A system might successfully send a required notification. A regulated client may also need evidence showing when the notification was generated, who received it, which version was sent, and whether any later action occurred.

That means documentation becomes part of the product. Depending on the industry, the system may need:

- Timestamps
- User activity logs
- Version histories
- Completion records
- Administrative notes
- Approval records
- Document storage
- Identity records
- Reporting history
- Data exports



An audit trail helps regulated software preserve evidence of important actions and approvals.

For independent technology professionals, this changes project scope. A client asking for a simple workflow may actually need a workflow plus the administrative infrastructure required to demonstrate compliance later. That should be discovered before the estimate is finalized.

4. When do identity requirements become product requirements?

Identity is another area where legal and technical requirements quickly overlap. A regulated process may require greater confidence about who is using the system, who completed an activity, or who approved a transaction.

The National Institute of Standards and Technology Digital Identity Guidelines address identity proofing, enrollment, authentication, authenticator management, federation, and related processes. Although businesses operate under different legal requirements, the framework illustrates an important product-design principle: identity is not one feature. It is a series of risk decisions.

Official reference: NIST SP 800-63-4, Digital Identity Guidelines

- Does the system only need an email and password?
- Is multifactor authentication appropriate?
- Does identity need to be verified before someone performs a specific activity?
- What happens when credentials are lost?
- Which administrator can change user information?
- How are suspicious events handled?

Freelancers building authentication or onboarding workflows should understand the client's actual risk and compliance requirements instead of selecting an identity process based only on convenience.

5. How should a product handle multi-jurisdiction compliance?

A process operating in one jurisdiction is easier to manage than a product operating across many. Rules can differ by state, county, city, licensing body, court system, or other authority.

I saw this repeatedly as CyberActive expanded regulated education programs. A requirement that appeared universal could turn out to have important jurisdiction-specific variations.

This creates a product architecture question: Do you hard-code every difference, or build a system capable of managing different rule sets?

Hard-coding every exception may work early. It becomes increasingly difficult as the business grows. A scalable system should separate configurable rules from the underlying product whenever practical.



Configurable jurisdiction rules can help regulated products adapt without rebuilding the core system.

Configurable jurisdiction rules can help regulated products adapt without rebuilding the core system.

Configuration may need to account for:

- Required fields
- Course or process requirements
- Notices
- Completion rules
- Reporting destinations
- Document templates
- Approval steps
- Retention requirements
- Jurisdiction-specific language

Independent developers should ask about expansion plans early. If a client operates in one state today but expects to enter ten more, the architecture decision made during the first project could determine whether growth requires configuration or a major rebuild.

6. Which compliance tasks should be automated, and which should stay human?

Regulated businesses have strong incentives to automate. Manual compliance processes are slow, create administrative work, and increase the possibility of missed steps. But automation works best when the rule is clear.

Software is well suited to repeatable actions such as:

- Sending scheduled notices
- Recording timestamps
- Generating routine documents from approved templates
- Tracking deadlines
- Flagging incomplete records
- Routing information to the correct department
- Producing standardized reports

Automation becomes more complicated when a decision requires interpretation. A system may identify a possible exception, but a qualified person may still need to determine what that exception means.

Automation becomes more complicated when a decision requires interpretation. A system may identify a possible exception, but a qualified person may still need to determine what that exception means.

The question should not be, “How much compliance can we automate?” A more useful question is, “Which parts of this process are consistent enough to automate safely, and where does the client require human judgment?”

For freelancers and independent developers, documenting that distinction protects both the product and the working relationship.

> *For independent professionals, freelance project management tools can also help organize deadlines, task ownership, documentation and repeatable project workflows alongside compliance processes.*



Automate repeatable compliance tasks while keeping human judgment for decisions requiring interpretation.

7. Who should be responsible for legal interpretation?

Technology consultants working with regulated clients should be especially careful about responsibility. A developer may understand how to implement a requirement without being qualified to decide what the law requires. Those are different jobs.

The project scope should identify who supplies the compliance requirements, who interprets them, and who approves them. A clear statement of work can help document these responsibilities, deliverables, approvals, and change requirements before development begins.

- The client supplies applicable regulatory requirements.
- The client's legal or compliance team approves interpretations.
- The technology provider converts approved requirements into product functionality.
- Regulatory changes requiring additional development are handled through an agreed change process.

This prevents an informal conversation from turning into an expectation that the freelancer is responsible for monitoring or interpreting an entire regulatory environment. For independent professionals, these responsibilities should be reflected in the broader freelance contract essentials that define the working relationship before development starts.

For an independent professional, this distinction belongs in the project conversation before development starts.

8. How should regulated products be designed for changing rules?

Compliance is not a one-time development task. Rules change. Agency procedures change. Forms change. Reporting requirements change. Client policies change. Technology introduces new risks that regulators eventually address.

That means regulated products need a process for updates. When designing the system, consider:

- Can administrators update required text without a software release?
- Are templates version controlled?
- Can jurisdiction-specific rules be modified independently?
- Is there a record of previous configurations?
- Who receives notice when requirements change?
- Is there a testing process before a new rule goes live?

A product becomes easier to maintain when change is expected instead of treated as an exception.

9. What should you ask before accepting a regulated-technology project?

Freelancers often receive requirements that sound simple: "We need an online form." "We need users to complete this process." "We need a dashboard." "We need to automate these approvals."

In regulated industries, each sentence deserves follow-up questions. Before agreeing to the scope, ask:

1. What regulation or policy drives this requirement?
2. Does the requirement differ by jurisdiction?
3. Who determines whether the implementation complies?
4. What evidence must the system retain?
5. Does the client need reporting or audit capabilities?
6. Which actions require human approval?
7. Are there identity verification requirements?
8. How long must records be maintained?
9. Who receives reports or completion information?
10. What happens when the underlying rule changes?

These questions do more than reduce compliance risk. They help produce a more accurate estimate, a clearer statement of work, and fewer surprises during development.

A compliance-by-design checklist for founders and independent developers

Area	Question
Requirements	What rules affect the workflow?
Responsibility	Who interprets and approves those rules?
Implementation	How will each approved requirement appear in the product?
Evidence	What must the system record to demonstrate that the required process occurred?
Change	How will the system adapt when requirements are updated?

If one of those answers is missing, there is probably more discovery work to do.

Frequently asked questions about compliance by design

What is compliance by design in software development?

Compliance by design means identifying legal, regulatory, licensing, reporting, and evidence requirements early enough to shape the workflow, data model, permissions, audit trail, and administrative tools. It treats compliance as a product input rather than a final review.

Why is compliance important in product design?

In regulated markets, a legal requirement can change how users register, how identity is verified, what records are retained, which steps require human approval, and how completion is reported. Discovering those requirements late can force expensive redesigns.

What is an audit trail in regulated software?

An audit trail is a reliable record of relevant system events, such as timestamps, user actions, approvals, document versions, notices, and completion events. The exact evidence required depends on the industry and applicable rules.

Can compliance be automated?

Repeatable actions such as notices, timestamps, deadline tracking, routing, and standardized reports are often good candidates for automation. Decisions that require legal interpretation, discretion, or professional judgment may still require human review.

How should software handle different state or local rules?

When practical, products should separate configurable jurisdiction-specific rules from the core product. This can make it easier to change required fields, notices, completion rules, reporting destinations, templates, and retention settings without rebuilding the entire system.

Compliance should shape the product early

The most useful lesson I learned from building technology in regulated markets is simple: compliance becomes expensive when it is discovered late.

When requirements are understood early, they can inform the architecture, workflow, data model, documentation, administrative tools, and project scope.

For founders, that produces a stronger product. For freelancers and independent technology professionals, it also creates a better client relationship. You understand what you are building, the client understands what it is responsible for providing, and both sides have a clearer picture of what happens when regulations change.

Regulation does not sit outside the product. In the industries where it matters most, regulation is one of the things the product is built to handle.

About the author

Sharon Ourian is an attorney, technology entrepreneur, and founder and CEO of CyberActive. She founded the company while attending UCLA School of Law and has spent more than two decades building technology for regulated online education and compliance-sensitive workflows. She is also co-founder and CEO of MELKpm and a principal of the Ourian Family Office, where she has applied her technology and operating experience across real estate and multi-entity businesses.

Official references

U.S. Small Business Administration: Launch your business – licenses and permits

NIST SP 800-63-4: Digital Identity Guidelines