

From Annual Audits to Continuous Compliance: A Multi-Agent Architecture for Automated Regulatory Monitoring of AI Hiring Systems

Tony Rossi

Tony Rossi Consulting LLC, Poulsbo, WA | me@tonyrossi.dev

Abstract

AI-powered hiring systems are classified as high-risk under the EU AI Act and subject to overlapping regulations including NYC Local Law 144, CCPA/CPRA, BIPA, EEOC Title VII guidance, and WCAG 2.2 accessibility standards. Despite this regulatory pressure, compliance monitoring remains manual and periodic, typically limited to annual bias audits that recent research has shown to be broadly ineffective. No existing system provides continuous, automated compliance monitoring across multiple regulatory frameworks for a production hiring platform.

This paper presents a multi-agent CI/CD compliance architecture deployed on a production hiring platform. The system uses 15 AI agents organized into two teams: a 9-agent code review team (plus supervisor) that gates every deployment against regulatory requirements, and a 6-agent adversarial red team that conducts recurring security and compliance testing. Privacy and Accessibility agents hold mandatory veto authority over all deployments. The architecture integrates into GitHub Actions, gating promotions from development through test to production.

We ran this system against our own platform. It found 90 issues across two coordinated audits: 56 from a compliance sweep (14 Critical, 14 High, 24 Medium, 4 Low) and 34 from a WCAG 2.2 AA accessibility audit (6 Critical, 18 High, 10 Medium). Five compliance findings were legal blockers. The platform's WCAG conformance was estimated at 45–55%. The system issued a REJECT verdict against its own platform's codebase. We report all findings here, including the ones we would have preferred not to discover.

1 Introduction

If you build an AI system that scores job candidates, you are building a high-risk system under the EU AI Act's Annex III, with compliance obligations taking effect August 2, 2026 [1]. You are also likely subject to NYC Local Law 144's bias audit mandates, CCPA/CPRA data rights provisions, Illinois BIPA if you use biometric verification, and the EEOC's guidance on algorithmic adverse impact under Title VII [9]. As of April 2026, ADA Title II digital accessibility requirements add another layer [19]. That is six overlapping regulatory frameworks for a single product category, and the list is still growing.

The industry's answer to this has been annual bias audits. The evidence suggests this is not working. Gerchick

et al. [2] conducted the most comprehensive analysis to date, examining all 116 publicly available NYC Local Law 144 bias audits. They found that 83% were missing demographic data, that audits existed for roughly 2% of Fortune 500 companies, and that year-over-year comparison was essentially impossible due to shifting methodologies between audit cycles. Wright et al. [3] took a different approach, deploying 155 student investigators to directly examine employer compliance. What they documented was what they termed “null compliance”: employers exploiting the law's definitional ambiguity to self-classify their systems as out of scope. Groves et al. [4] interviewed 17 practitioners working within the regime and concluded it had failed to create effective accountability.

These are not edge cases. They point to a structural mismatch between the cadence of compliance assessment and the cadence of software development. A hiring platform that ships code three times a week can introduce a GDPR violation on Tuesday that will not surface until next year's audit. Accessibility regressions are particularly hard to catch: a developer adds a modal dialog, does not implement a focus trap, and keyboard-only users silently lose access to a feature for months. Nobody notices because the developer is not a keyboard user, the QA team is not a keyboard user, and there is no automated system checking.

We built a system to close this gap. It is a multi-agent compliance pipeline that runs inside the CI/CD process of a production hiring platform, monitoring six regulatory and standards frameworks continuously rather than annually. The architecture uses 15 AI agents organized into two coordinated teams, augmented by automated accessibility scanning, all integrated into GitHub Actions. The privacy agent and accessibility agent both hold hard veto power over deployments. To our knowledge, no published system does anything comparable. The closest work in the literature is O'Keefe and Frazier's proposal for “compliance-automating AIs” [15], which articulates the concept but provides no implementation.

This paper describes how the system works, reports what it found when we ran it against our own codebase (including findings that were, frankly, not what we hoped to see), and discusses the implications for AI governance more broadly.

2 Background and Related Work

2.1 Compliance and Auditing in AI Hiring

There is substantial academic literature on bias in AI hiring systems, but it overwhelmingly treats compliance as a research exercise rather than an engineering practice. Fabris et al. [5] produced the most comprehensive survey, covering over 250 AI tools used in HR across four pipeline stages. Raghavan et al. [6] established an early accountability baseline by studying claims and practices across 18 pre-employment assessment vendors. Glazko et al. [7] demonstrated disability bias in GPT-based resume screening, an important finding that directly connects AI hiring and accessibility. Xu et al. [8] contributed a novel result at AIES 2025, showing that LLMs systematically prefer AI-generated resumes over human-written ones at rates between 68% and 92%, a bias vector that only continuous testing would reliably detect.

What is missing from all of this work is the engineering layer: a system that performs compliance monitoring as part of the development process itself, not as a periodic external assessment. No paper at AIES, FAccT, or any comparable venue describes such a system for AI hiring platforms. The intersection of AI governance and digital accessibility is particularly neglected. Both communities have produced deep bodies of work, but no one has unified them within a single compliance pipeline.

2.2 The Regulatory Landscape

The regulatory environment for AI hiring is defined by jurisdictional fragmentation and tightening enforcement timelines. We summarize the six frameworks our system tracks.

The **EU AI Act** [1] explicitly classifies employment AI as high-risk under Annex III, Category 4. Obligations include risk management systems, technical documentation, data governance, transparency, human oversight, and accuracy requirements. Conformity assessment obligations take effect August 2, 2026, and the requirements are largely computational: logging, documentation, bias monitoring, and transparency reporting can all be automated.

NYC Local Law 144 requires annual bias audits for automated employment decision tools, with public posting and candidate notification. The empirical record on compliance is sobering [2, 3, 4], but the law’s core requirements — selection rate calculations, impact ratio computations, intersectional analysis — are inherently amenable to automation.

The **EEOC’s** May 2023 Technical Assistance Document [9] establishes that AI tools constitute “selection procedures” under the Uniform Guidelines. The four-fifths rule applies. **CCPA/CPRA** employment data exemptions expired January 1, 2023, with new ADMT regulations effective January 1, 2026. Academic treatment of CPRA applied to candidate data remains thin. **BIPA** applies to biometric identity verification in hiring platforms; *Deyerler v. HireVue* [10] upheld BIPA claims for facial geometry scans in AI video interviews.

WCAG 2.2 / ADA / Section 508: The ADA Title II final rule [19] mandates WCAG 2.2 AA conformance by April 2026. Section 508 covers federal technology. EN 301 549

aligns European standards. For hiring platforms specifically, inaccessible interfaces do not merely create legal exposure — they prevent candidates with disabilities from applying for jobs, which compounds exactly the discrimination that bias auditing is supposed to address.

The common thread is that most compliance obligations across these frameworks reduce to computational verification: does this data collection have a documented lawful basis? Is there a disclosure when AI influences a decision? Does a deletion request propagate through all subsystems? Does this interactive element have a keyboard handler? These are checks agents can run on every commit.

2.3 Multi-Agent Systems for Software Engineering

The multi-agent paradigm for software engineering has produced several notable systems, though none at the scale or with the compliance focus of our architecture. ChatDev [11] coordinates 7 agents for collaborative development. MetaGPT [12] uses 5 agents with SOP-based workflows. In production settings, HubSpot’s Sidekick [13] uses 2 agents for code review, and RepoAudit [14] uses 3 components for vulnerability detection. None of these systems exceeds 7 agents, combines code review with adversarial red teaming, or targets regulatory compliance.

The most relevant compliance-focused work includes GaaS [16], a modular governance layer with trust-factor scoring that was evaluated only in simulation across essay-writing and financial trading domains, and PCAS [17], a Datalog-based policy compiler that improved compliance from 48% to 93% on customer service benchmarks but was not deployed in production. Our system operates at a different scale — 15 agents across two teams — and against a different target: continuous regulatory compliance monitoring for a live platform handling real candidate data.

3 System Architecture

3.1 Platform Context

The system is deployed on a production hiring platform running React 18 with TypeScript, Cloudflare Workers (Hono framework), and Neon PostgreSQL. The platform uses AES-256-GCM field-level encryption on PII, a 4-state identity reveal machine for blind candidate screening, FaceTec biometric verification, and a 28-vector AI assessment engine with inference via NVIDIA NIM. The frontend uses Framer Motion for animations, 7 visual theme skins, and @dnd-kit for drag-and-drop interfaces.

This technology stack is representative of modern SaaS hiring platforms, and the compliance challenges it creates — multiple data stores, AI inference pipelines, third-party processor integrations, multi-jurisdictional users, complex interactive frontends — are generalizable beyond this specific system.

Figure 1 illustrates the overall pipeline architecture and the flow between the two agent teams.

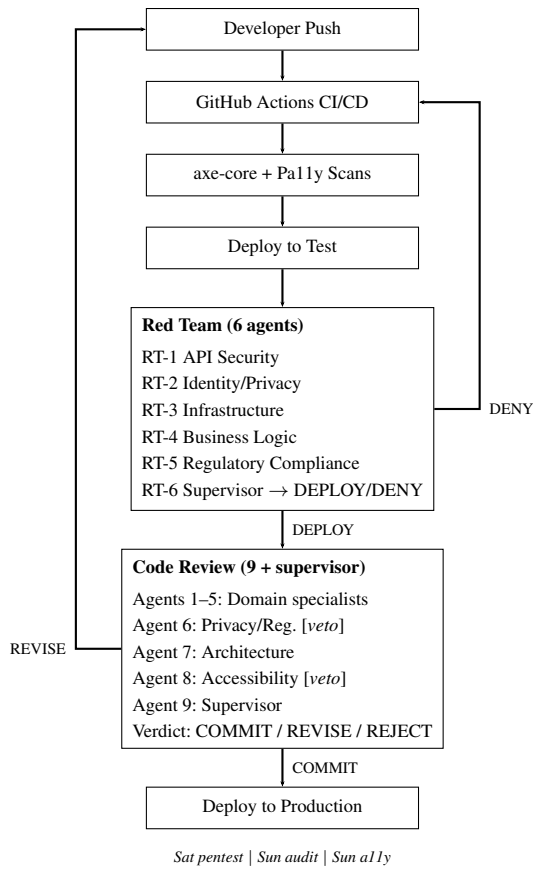


Figure 1: Pipeline architecture. Code flows through accessibility scans, adversarial red team testing, and multi-specialist code review before production. DENY and REVISE verdicts loop back. Agents 6 and 8 hold veto authority.

3.2 Code Review Team

The code review team gates every deployment promotion. Table 1 summarizes the nine specialists and the supervisor. Rather than describe each agent identically, we focus on the three that carry the most weight in the architecture and note how the cross-domain challenge protocol ties all of them together.

The Privacy & Regulatory Compliance agent (Agent 6) carries the most extensive mandate in the system. It holds mandatory veto authority: any regulatory gap triggers an automatic REVISE that blocks deployment. The agent tracks compliance across 17 regulatory frameworks organized into five primary categories (CCPA/CPRA, GDPR, EU AI Act, BIPA, NYC LL 144), plus 10 US state privacy laws, four international frameworks, FTC Act Section 5, and EEOC/Title VII.

What makes this agent consequential is the granularity of its checks. For every piece of personal data the platform collects, it verifies documented purpose, lawful basis, retention period, deletion mechanism, and privacy policy disclosure. Its output includes a compliance matrix (Regulation $\times$ Feature, rated COMPLIANT / GAP / AT RISK), specific reme-

Agent	Primary Domain
1. Frontend	React/TypeScript, theming, components
2. Backend	Cloudflare Workers, CORS, middleware
3. Database	Neon PostgreSQL, encryption, migrations
4. Auth/Security	Authentication, identity reveal, PII leaks
5. AI/NIM	Inference paths, assessment engine, coaching
6. Privacy	17 regulatory frameworks, <i>veto authority</i>
7. Architecture	Dependencies, coupling, team-readiness
8. Accessibility	WCAG 2.2 AA, <i>veto authority</i>
9. Supervisor	Final gate, cross-domain challenges

Table 1: Code review team roster. Agents 6 and 8 hold mandatory veto power over deployments.

diation guidance, DPIA recommendations, and a prioritized regulatory roadmap. In practice, this was the agent that discovered our platform had no deletion endpoint at all — a single gap that turned out to violate three regulations simultaneously.

The Accessibility agent (Agent 8) also holds veto authority. It audits against all 78 WCAG 2.2 AA success criteria, plus ADA Title II, Section 508, and EN 301 549. Its scope covers semantic HTML, keyboard navigation, screen reader compatibility, color contrast verified across all 7 theme skins, modal focus management, touch targets, motion preferences, form labeling, authentication accessibility (WCAG 3.3.8), and SVG chart accessibility. Every finding references a specific WCAG criterion and identifies which assistive technologies are affected. This agent found that our AI coaching interface — the platform’s core differentiating feature — was entirely invisible to screen readers. We did not know this until the agent reported it.

The Supervisor (Agent 9) waits for all eight specialists to complete, reads every finding, and actively challenges any agent that reported fewer than three issues (requiring evidence of thorough investigation). It runs the test suite, checks cross-domain consistency, and issues a verdict: COMMIT, REVISE, or REJECT. Any regulatory gap or Critical accessibility barrier is an automatic REVISE. On its first full run against our codebase, it issued REJECT.

The remaining agents (Frontend, Backend, Database, Auth/Security, AI/NIM, Architecture) operate on similar principles within their respective domains. The cross-domain challenge protocol is what connects them: when the database agent encounters a new PII field, the privacy agent must verify its lawful basis. When the AI agent changes scoring logic, the privacy agent checks EU AI Act disclosure requirements. When the frontend agent adds an interactive component, the accessibility agent verifies keyboard navigation and screen reader support. This prevents the kind of siloed analysis where each domain looks clean in isolation but the intersections harbor compound risks.

3.3 Red Team

The red team runs weekly and on every dev-to-test promotion. Where the code review team does static analysis, the red team tests the live environment adversarially. RT-1 (API Security) runs systematic penetration testing: authentication bypass, input fuzzing, CORS abuse, privilege escalation. RT-2 (Identity & Privacy) attempts to break blind screening, force identity-reveal state transitions, and extract PII through side channels. RT-3 (Infrastructure) stress-tests Workers, checks circuit breakers, and scans for CVEs. RT-4 (Business Logic) tests tier enforcement, payment gaming, prompt injection against the AI coach, and duplicate account creation.

RT-5 (Regulatory Compliance) is the red team’s most distinctive agent. It simulates actual enforcement actions: CCPA “Right to Know” requests checked for completeness and timeliness, GDPR erasure requests traced through every subsystem to verify nothing survives, consent withdrawal testing to verify processing actually stops, EU AI Act transparency checks, and dark pattern audits. RT-6 (Supervisor) compiles all findings, cross-references compound vulnerabilities across agents, and renders DEPLOY or DENY. Any regulatory finding from RT-5 rated High or Critical triggers an automatic DENY.

3.4 Automated Scanning and CI/CD Integration

The pipeline also runs axe-core and Pally against the live test deployment on every commit. These tools catch rendered-state accessibility violations — computed contrast ratios, actual DOM structure after JavaScript executes — that static code analysis misses. The AI accessibility agent, conversely, catches architectural patterns (missing ARIA live regions, absent focus management logic, motion preference handling) that automated scanners cannot reason about. The two approaches are complementary.

The full architecture integrates through GitHub Actions across three isolated environments, each with separate Cloudflare Worker deployments, Neon database branches, and secrets. Dev-to-test triggers accessibility scans followed by the red team. Test-to-production requires an explicit DEPLOY verdict with a linked report. When the red team issues DENY, the report feeds back to the code review team for remediation, creating a closed loop that repeats until DEPLOY.

The weekend audit schedule runs three comprehensive sweeps in sequence: red team penetration test (Saturday 02:00 UTC), codebase compliance audit (Sunday 03:00 UTC), and accessibility audit (Sunday 04:00 UTC). By Monday morning the development team has three reports covering security posture, regulatory compliance, and accessibility conformance. Critical or High findings automatically create labeled GitHub Issues.

4 Evaluation

4.1 Methodology

We report results from two audits conducted against the production codebase on March 29, 2026, approximately four months after the platform launched. The codebase had been

developed without continuous automated monitoring during that period, so the results represent the accumulated compliance and accessibility debt of the initial build phase.

The compliance audit deployed all nine review agents sequentially against the full codebase (not a diff), with each specialist sweeping their entire domain. The accessibility audit ran the WCAG specialist against all frontend code, supplemented by axe-core and Pally scans against the live deployment. Both supervisors compiled findings, challenged agents, and produced master reports.

4.2 Results Overview

Audit	Crit.	High	Med.	Low	Total
Codebase Compliance	14	14	24	4	56
Accessibility (WCAG)	6	18	10	0	34
Combined	20	32	34	4	90

Table 2: Combined audit findings by severity.

Ninety findings total. Twenty Critical, of which five were legal blockers (all regulatory) and six were complete barriers to assistive technology users. Estimated WCAG conformance: 45–55%. The supervisor issued REJECT.

4.3 Regulatory Findings

The compliance audit found violations across every framework we track.

GDPR produced five violations. The most consequential was the absence of any account deletion endpoint (Art. 17): no `DELETE /user/me` route existed, and the ML-derived data tables had no cascading deletes, meaning data would be orphaned even if someone manually attempted erasure. The remaining four: no DPIAs conducted for AI scoring, matching, or classification (Art. 35); no DPAs with any of our six third-party processors (Art. 28); no Standard Contractual Clauses for transatlantic transfers; and analytics events firing without consent verification (ePrivacy Directive).

EU AI Act: The AI assessment engine scores every candidate across 28 vectors and directly influences employer search rankings. At the time of the audit, there was no disclosure to candidates, no explanation of how scores were derived, no human review mechanism, and no opt-out. This is the scenario Annex III’s high-risk transparency requirements target. Separately, no high-risk AI documentation existed (training data provenance, bias testing methodology, risk management system).

BIPA: The biometric processor (FaceTec) lacked written data protection terms, and biometric photos were not deleted on consent revocation. **NYC LL 144:** No audit trail table existed for AI-influenced hiring decisions. **CCPA:** The missing deletion endpoint meant statutory deletion requests could not be fulfilled within required timeframes.

Two findings hit multiple frameworks simultaneously. The absent deletion endpoint violates GDPR Art. 17, CCPA §1798.105, and BIPA destruction requirements in a single gap. The undisclosed AI scoring violates the EU AI

Act, GDPR Art. 22, and NYC LL 144 AEDT notice requirements. These compound violations show why multi-framework monitoring matters: a single-domain auditor checking only GDPR would flag the deletion issue but miss the BIPA destruction timeline implication.

4.4 Accessibility Findings

The six Critical barriers clustered in the AI coaching interface, which is the feature candidates interact with most. The send button was implemented as a `<span>` with an `onClick` handler rather than a native `<button>`, making it invisible to assistive technology and unreachable by keyboard (WCAG 4.1.2, 2.1.1). The text input relied on placeholder text with no programmatic label (WCAG 1.3.1). The response area lacked an `aria-live` region, so the AI coach’s replies appeared silently without screen reader announcement (WCAG 4.1.3). The kanban board had no `KeyboardSensor` registered, preventing keyboard users from moving cards between pipeline stages (WCAG 2.1.1). Two additional chat-related components replicated the same labeling and announcement failures.

The result: the platform’s most important feature was completely inaccessible to the users who arguably need it most. This was not a known issue before the audit surfaced it.

Among the 18 High findings, several stood out. Over 20 files used Framer Motion animations without checking `prefers-reduced-motion` — fixable with a single line of code wrapping the app root in `<MotionConfig reducedMotion="user">`. Twelve modal dialogs lacked focus traps, allowing keyboard focus to escape to background content. Authentication form errors were not announced to screen readers. The AI assessment visualization (a radial SVG chart) had zero assistive technology support: no `role="img"`, no `<title>`, no `<desc>`, no keyboard access to interactive elements. Six search filter controls used placeholder text as their only label. And several modals used hardcoded Tailwind utility classes that bypassed the platform’s design token system, producing contrast ratios around 4.1:1 against the required 4.5:1.

4.5 Security and Cross-Domain Findings

The audit also surfaced security vulnerabilities with privacy implications: user email addresses written to production logs via `console.log()`; a CORS origin check using `.endsWith()` with `credentials: true`; 60+ database queries missing schema prefixes; 27 SQLite syntax fragments surviving a PostgreSQL migration; no rate limiting on the registration endpoint; and an authorization bypass on a reveal-tier endpoint.

The supervisor’s cross-domain analysis identified five compound risk clusters. The most dangerous was a single file (`admin-beta-invites.ts`) that simultaneously logged PII to production, contained a 40-step non-transactional user erasure process, and served as the platform’s only deletion path — in a system with no self-service delete endpoint. A 5,205-line monolithic module sitting outside the domain architecture contained 12 of the 27 SQLite

syntax violations. The AI assessment engine was technically sound (28 vectors validated, prompt injection defenses working, per-candidate persistence correct) but the entire disclosure layer was absent.

The most revealing cluster was the intersection of AI and accessibility. The coaching interface and assessment visualization were both working correctly from a functional standpoint and both completely inaccessible to assistive technology users. These features were developed on separate tracks with no cross-checking between them. The accessibility agent’s specific mandate to review AI-facing components exists to prevent exactly this pattern.

The audit also confirmed 15 areas of full compliance, including zero cross-domain imports, fully parameterized SQL, correct session and cookie security, a properly enforced identity reveal state machine, assessment framework integrity with bounds enforcement and anti-gaming flags, and prompt injection defenses on all user-facing AI endpoints. This positive evidence is essential for EU AI Act conformity assessment, which requires demonstrating both what works and what does not.

4.6 Remediation

Both audits produced prioritized fix schedules with specific file paths, line numbers, and concrete remediation steps. Table 3 breaks down the Critical compliance findings by regulation, estimated fix time, and the nature of the change required.

Finding	Est. Fix	Regulations
No deletion endpoint	1 day	GDPR 17, CCPA, BIPA
PII in production logs	2 hrs	GDPR 5(1)(c)
CORS wildcard + creds	2 hrs	Fetch spec, CSRF
AI scoring undisclosed	1 day	AI Act, GDPR 22, LL144
Missing schema prefixes	1–2 days	Data integrity
SQLite syntax in Postgres	1–2 days	Data integrity
No registration rate limit	3 hrs	Security
FaceTec DPA missing	4 hrs	BIPA
Analytics without consent	4 hrs	GDPR, ePrivacy
NIM key config error	2 hrs	Operational

Table 3: Critical compliance findings with estimated remediation time and applicable regulations. Total estimated: 5 working days.

The compliance audit estimated five working days for all Critical fixes. The fastest changes were surgical: removing a `console.log()` that printed user emails, or fixing a CORS origin whitelist. The most complex was the deletion endpoint, which required implementing cascading hard deletes across three ML-derived tables, adding `ON DELETE CASCADE` foreign key constraints, and building the API route itself.

The accessibility remediation showed a different pattern. Several Critical and High findings had dispropor-

tionately efficient fixes. Wrapping the application root in `<MotionConfig reducedMotion="user">` resolves the motion-preference issue across every animated file in a single line of code. Converting the chat send button from `<span>` to `<button>` fixes a Critical barrier in under a minute. Adding `aria-live="polite"` to the chat response container takes seconds and makes AI coaching responses audible to screen reader users for the first time.

Other accessibility fixes required more structural work. Implementing focus traps across 12 modals means either adopting a shared accessible modal component or retrofitting each one individually. Adding keyboard support to the @dnd-kit kanban board requires registering a `KeyboardSensor` with `sortableKeyboardCoordinates` and configuring accessibility announcements, which is a non-trivial change to the drag-and-drop interaction model.

This variation in fix complexity is itself informative. It suggests that continuous monitoring is most valuable not for the hard problems (which require planned engineering effort regardless of when they are discovered) but for the easy ones that persist simply because nobody knew they existed. A one-line fix that sits undiscovered for a year is a year of unnecessary non-compliance.

Flows Requiring Manual Testing The accessibility audit identified nine user flows that require hands-on testing with assistive technologies, beyond what automated analysis can validate:

1. AI coaching chat: send a message and hear the response (NVDA + Chrome, VoiceOver + Safari)
2. Kanban board: keyboard drag between columns (NVDA keyboard mode)
3. Professional DNA chart: understand assessment scores via screen reader (VoiceOver + Safari)
4. Sign-in OTP flow: email to code to passkey (NVDA, JAWS)
5. Job detail modal: open, navigate content, Escape to close (all three major screen readers)
6. Rejection modal: navigate and interact (NVDA)
7. All 7 theme skins: verify contrast ratios visually (Colour Contrast Analyser)
8. Assessment quiz: keyboard-only select and submit (VoiceOver)
9. Resume Lab: small action buttons on touch devices (VoiceOver on iOS)

These represent the boundary of what our automated system can verify. Full WCAG conformance will require completing this manual testing protocol, which we have scheduled but not yet executed at the time of writing.

5 Discussion

5.1 The Case for Continuous Monitoring

The five legal blockers and six Critical accessibility barriers existed in the codebase for four months before the system caught them. Under annual auditing, they could have

persisted for up to a year, silently affecting every user onboarded during that window. The continuous model catches violations either at the point of introduction (code review team, per-commit accessibility scans) or shortly afterward (weekly audit cycle), and the red team feedback loop drives iterative improvement that periodic assessment structurally cannot.

Accessibility regressions make an especially compelling case. A developer who does not use a keyboard adds a modal without focus trapping. A developer who can see the screen does not notice the missing `aria-live` region. These are not negligent developers; they are developers whose testing naturally reflects their own abilities. Continuous automated monitoring compensates for this gap by checking what the developer would not think to check.

5.2 On Giving Agents Veto Power

Granting hard veto authority to the privacy and accessibility agents was a deliberate and, at the time, somewhat uncomfortable decision. Bamberger and Mulligan [18] documented the organizational dynamic where compliance teams advise but cannot block. We inverted this: a missing consent flow, an undisclosed AI decision, or an inaccessible form control is treated the same as a failing test suite. The code does not deploy.

The first audit validated this choice. Every Critical finding the system caught had been in production for months, introduced by developers (ourselves included) who either did not know the requirement existed or assumed it could wait. Making compliance a hard gate rather than advisory feedback changes the calculus: you fix the issue before merge because the alternative is that nothing ships.

5.3 Implications for Regulatory Design

O’Keefe and Frazier [15] recently proposed the concept of “automatability triggers”: regulatory mechanisms that defer the effective date of compliance obligations until AI tools exist that can automate those obligations at reasonable cost. Their argument is that this resolves the timing dilemma in AI regulation — you can write strict rules without stifling innovation, because the rules only activate when compliance is cheap.

Our system provides empirical evidence for the feasibility side of that argument. The compliance checks we automate — deletion endpoint verification, consent flow tracing, AI disclosure presence, bias metric computation, WCAG criterion checking — are exactly the kinds of obligations that future regulations could condition on automatability. The remediation table (Table 3) shows that many violations, once identified by an automated system, can be resolved in hours. The bottleneck was never the fix; it was knowing the problem existed.

This suggests a concrete policy direction: regulations could require platforms to deploy continuous compliance monitoring systems as a condition of operating in high-risk AI domains, rather than relying on periodic third-party audits that the evidence shows are failing. The monitoring architecture itself becomes the compliance mechanism, with

audit artifacts serving as evidence of ongoing conformity rather than point-in-time snapshots.

5.4 Cost and Scalability

The full code review pipeline costs roughly 9x a single review session in API tokens. The red team adds additional costs for hitting live endpoints. `axe-core` and `Pa11y` add negligible overhead. We do not run the full 15-agent pipeline on every commit; routine changes get individual specialists, with the full sweep running weekly. This tiered approach is workable for an early-stage platform but would need re-evaluation as codebase size and deployment frequency scale.

5.5 Limitations

The most significant limitation is that these results come from a single production platform on a single technology stack. The architecture is designed to generalize — swap the regulatory mappings, keep the agent structure and CI/CD integration — but that generalizability is a design claim, not a tested one. We intend to evaluate deployment on additional systems.

Agent reliability presents a real and unresolved challenge. The agents produce false positives and they miss things. The supervisor’s challenge protocol helps surface false negatives, but we have not formally measured recall, and the supervisor itself can miss cross-domain issues. The regulatory interpretations encoded in Agent 6 reflect specific readings of regulations that remain genuinely ambiguous; reasonable attorneys would disagree with some of them. This system flags issues for human review. It does not replace legal counsel.

On accessibility, automated testing — even combining an AI agent with `axe-core` and `Pa11y` — cannot fully substitute for manual evaluation with actual assistive technologies. The nine user flows identified in Section 4.6 require hands-on screen reader testing that no automated tool can validate. Full WCAG 2.2 AA conformance assessment inherently requires human judgment on subjective criteria, and we have not yet completed that manual protocol.

5.6 Generalizability

The architecture is domain-independent in its structure. The agent team model (domain specialists + regulatory specialist + accessibility specialist + supervisor), the adversarial red team, and the CI/CD integration pattern all transfer without modification. What changes between domains is the regulatory mapping in Agent 6: healthcare AI would substitute HIPAA and FDA requirements for BIPA and LL 144. The accessibility agent’s WCAG mandate applies to any web-based system as-is.

The underlying insight is that regulatory compliance requirements decompose naturally into bounded computational checks that can be distributed across specialized agents and coordinated through a supervisor with veto authority. The question is not whether this decomposition is possible — the audit results demonstrate that it is — but whether it generalizes across different technology stacks and regulatory environments. That remains to be shown.

6 Conclusion

We built a 15-agent CI/CD compliance pipeline and pointed it at our own production hiring platform. It found 90 issues. Five were legal blockers spanning GDPR, the EU AI Act, BIPA, and CCPA. Six were accessibility barriers that made the platform’s core AI feature completely invisible to screen reader users. The system issued REJECT.

The cross-domain analysis revealed compound risks that no single-framework audit would surface, including a pattern where AI feature development and accessibility proceeded on entirely separate tracks with no mutual awareness. This is likely not unique to our platform.

Annual auditing is falling short. Eighty-three percent of LL 144 audits are missing demographic data. EU AI Act enforcement begins in August 2026. ADA digital accessibility deadlines have arrived. The distance between what regulations require and what the industry actually delivers is widening. Continuous multi-agent monitoring is one way to close that gap — not as a replacement for human judgment, legal expertise, and manual testing with assistive technologies, but as infrastructure that makes real compliance oversight possible.

Ethical Considerations

This paper describes a system deployed on a platform developed by one of the authors. We acknowledge the inherent conflict of interest in evaluating our own work and have addressed it by reporting all findings without exception: the legal blockers, the accessibility barriers, the REJECT verdict, and the 45–55% WCAG conformance estimate. The system found serious problems in our own code. We are publishing them because the contribution of this paper is the monitoring architecture, not the compliance of the platform it monitors.

The system analyzes code and system configurations, not individual user data. The audit results reported here describe system-level compliance states and contain no personally identifiable information.

We close with a caution. Automated compliance monitoring can produce false confidence if organizations treat agent verdicts as conclusive rather than advisory. Accessibility in particular demands ongoing engagement with users with disabilities, not just automated scanning. We advocate for continuous monitoring as a complement to legal expertise, human oversight, manual assistive technology testing, and external accountability — not a substitute for any of them.

References

- [1] European Parliament and Council. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act). *Official Journal of the European Union*, 2024.
- [2] M. K. Gerchick, R. Encarnación, C. Tanigawa-Lau, L. Armstrong, A. Gutiérrez, and D. Metaxa. Auditing the Audits: Lessons for Algorithmic Accountability from Local Law 144’s Bias Audits. In *Proc. ACM FAccT*, 2025.
- [3] L. Wright, R. M. Muenster, B. Vecchione, T. Qu, P. Cai, A. Smith, J. Metcalf, and J. N. Matias. Null

- Compliance: NYC Local Law 144 and the challenges of algorithm accountability. In *Proc. ACM FAccT*, 2024.
- [4] L. Groves, J. Metcalf, A. Kennedy, B. Vecchione, and A. Strait. Auditing Work: Exploring the New York City algorithmic bias audit regime. In *Proc. ACM FAccT*, 2024.
- [5] A. Fabris, N. Allhutter, and F. Pallas. Fairness and Bias in Algorithmic Hiring: a Multidisciplinary Survey. *ACM Transactions on Intelligent Systems and Technology*, 2024.
- [6] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy. Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices. In *Proc. ACM FAccT*, 2020.
- [7] K. Glazko, C. Kuo, and J. Forlizzi. An Audit of Ableism in Large Language Models for Resume Screening. In *Proc. ACM FAccT*, 2024.
- [8] Y. Xu et al. AI Self-preferencing in Algorithmic Hiring: Empirical Evidence and Insights. In *Proc. AAAI/ACM AIES*, 2025.
- [9] U.S. Equal Employment Opportunity Commission. Technical Assistance Document: Assessing Adverse Impact in Software, Algorithms, and AI Used in Employment Selection Procedures Under Title VII. May 2023.
- [10] *Deyerler v. HireVue*, No. 23-cv-5765 (N.D. Ill. 2024).
- [11] C. Qian, W. Liu, H. Liu, et al. ChatDev: Communicative Agents for Software Development. In *Proc. ACL*, 2024.
- [12] S. Hong, M. Zhuge, J. Chen, et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. In *Proc. ICLR*, 2024.
- [13] InfoQ. HubSpot’s Sidekick: Multi-Model AI Code Review. March 2026.
- [14] Z. Guo et al. RepoAudit: An Autonomous LLM-Agent for Repository-Level Code Auditing. In *Proc. ICML*, 2025.
- [15] C. O’Keefe and K. Frazier. Automated Compliance and the Regulation of AI. LawAI Working Paper Series, No. 1-2026, January 2026.
- [16] S. Gaurav, J. Heikkonen, and J. Chaudhary. Governance-as-a-Service: A Multi-Agent Framework for AI System Compliance and Policy Enforcement. arXiv:2508.18765, 2025.
- [17] N. Palumbo, S. Choudhary, J. Choi, P. Chalasani, and S. Jha. Policy Compiler for Secure Agentic Systems. arXiv:2602.16708, 2025.
- [18] K. A. Bamberger and D. K. Mulligan. *Privacy on the Ground: Driving Corporate Behavior in the United States and Europe*. MIT Press, 2015.
- [19] U.S. Department of Justice. Nondiscrimination on the Basis of Disability; Accessibility of Web Information and Services of State and Local Government Entities (Final Rule, ADA Title II). 89 FR 31320, April 2024.