International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.2 (2026) pp. 29-34
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Accessibility Under Distributed State: Engineering Account Access Experiences That Work for Everyone

Harshit Sunilkumar Vora*

Independent Researcher, USA
* **Corresponding Author Email:** harshit.vora143@gmail.com - **ORCID:** 0000-0002-0047-7008

Article Info:

DOI: 10.22399/ijcesen.5113
Received : 05 February 2026
Revised : 27 March 2026
Accepted : 29 March 2026

Keywords

Web Accessibility,
Account Authentication,
Distributed State Management,
Assistive Technology Compatibility,
WCAG Conformance

Abstract:

Account access experiences such as sign-in and authentication, account recovery, and device access enable users to manage their digital identity and to participate in civic, economic, and social aspects of life. Account access architecture is increasingly decentralized, with multiple changes from different identity providers, risk-scoring systems, and messaging services at nondeterministic times. The problems of accessibility are no longer about what is in the markup but about behavioral properties. These include a focus reset on re-render, communicating wait states only through animations, or status updates that never get exposed to assistive technology APIs. To eliminate these failure modes, this paper argues that accessibility must be treated as a first-class reliability property: specifiable, measurable, and stable under change. Accessibility contracts at the component level encode semantic, interaction, and announcement requirements for each component as testable defaults. The other two aspects are state modeling of engineering invariants in the transient, interrupted, and error states, and layered verification of both automated checkers and assistive technology-based manual checks, all along critical asynchronous paths. Release governance frameworks must connect these verification layers to deployment outcomes, ideally through explicit gates and incident-driven feedback loops, while conformance-driven measurement should be supplemented with user experience measures that automated tooling cannot capture. Account access flows that exclude people with disabilities deny them independent entry to the digital infrastructure through which banking, employment, healthcare, and civic participation are now primarily delivered.

1. Foundations of Accessible Account Access in Distributed Environments

Account access experiences, including sign-in, identity verification, credential recovery, device management, and security settings, are the foundational experience layer for modern digital participation: account access experiences determine whether a person can control their digital identity, access critical services, and respond to security challenges without support from another person. The outcome is more than inconvenience: users are functionally excluded from banking, communication, education, and employment services that have, in many cases, moved entirely online and have no offline equivalent.

The scale of this exclusion is substantial. The World Health Organization estimates that 1.3 billion people — approximately 16% of the global population — experience significant disability [1]. This includes individuals who rely on assistive technologies such as screen readers, switch access devices, refreshable Braille displays, and alternative pointing devices to navigate digitally presented content. When account access journeys are not designed to support these interaction modes, a significant share of the global population is denied independent access to essential digital infrastructure.

And then make reference [1] to the World Health Organization fact sheet, not the old cultural-overview source.

Research shows that there is enough justification for accessibility to exist as a separate area of engineering beyond overall usability. Research has specifically focused on the intersection between accessibility and situational disabilities and between web content accessibility guidelines and mobile web best practices. In this case, 19 design-related factors apply to a much larger group of

users than permanent disabilities do [2]. But for those with disabilities, accessible technology is not a nice-to-have or an avenue of convenience. It is the only way to participate[1].

Additionally, modern account access can feature a distributed architecture where state updates may originate from identity providers, risk-scoring systems, and messaging services with no guarantee of ordering and may have mid-flow session expiration. Network connectivity loss exposes behavioral failure modes in systems that lack explicit state-handling contracts — failures that were latent in the architecture regardless of connectivity conditions. For example, the focus is reset when components re-render, the only indication of a wait is an animation, and status changes are not communicated to assistive technology APIs. For mitigation to be successful, accessibility must be a first-class reliability property: specifiable, measurable, and stable under change [2].

2. WCAG 2.2 as a Behavioral Engineering Specification

The WCAG 2.2 specification — now the recommended conformance target per W3C — defines testable success criteria across four principles (Perceivable, Operable, Understandable, and Robust) that map directly onto the class of engineering failures characteristic of distributed account access flows [3]. WCAG 2.2 introduced several criteria of particular relevance here: SC 2.4.11 (Focus Not Obscured) requires that focused components not be entirely hidden by author-created content such as overlays or sticky headers; and SC 3.3.8 (Accessible Authentication) prohibits cognitive function tests as the sole authentication mechanism unless an accessible alternative is provided. These criteria, alongside the foundational four-principle framework and three conformance levels (Level A, Level AA, and Level AAA in order of priority), form a layered system of testable success criteria to be used in design specification, purchasing, regulation, and contracts [3].

Perceivable failures arise when status is conveyed using only color and motion, when a programmatic name is not associated with a form field, or when an inline error is presented but not programmatically associated with its form field. Account authentication fails at scale because a credential field without a programmatic name is unusable by a screen reader without guesswork, and an error message not programmatically associated with its form field provides no actionable information to a blind user navigating by form control.Keyboard-operable failures include any keyboard actions that violate accessibility norms, such as an illogical tab order or focus indicators obscured by overlays and sticky elements — the condition SC 2.4.11 directly addresses at the AA conformance level. SC 2.2.1 requires a mechanism to inform users that time limits exist and to give diverse users at least 20 seconds to extend the time limit, allowing users to extend the time limit at least 10 times [3]. In distributed authentication protocols where the session may be disrupted mid-flow, this criterion bounds the protocol behavior at the session level.

SC 3.3.8 responds to precisely this failure class by requiring that any cognitive function test used in authentication be accompanied by an accessible alternative, since a loop of repeated failures without a guaranteed resolution path is an effective denial of service for affected users.

Similarly, SC 4.1.3 (Status Messages) directly governs the announcement behavior that distributed authentication flows require: confirmation messages, error notices, and session state updates must be programmatically determinable so that assistive technology can surface them without the user's focus being redirected. This criterion closes the gap between dynamically updated content and the assistive technology APIs that expose it.

3. Component-Level Accessibility Contracts and Reusable Patterns

At scale, accessibility is operationalized most effectively through shared component systems. Account access experiences reuse a small set of interaction primitives across many screens and device classes: text inputs, credential inputs, selection controls, confirmation dialogs, status banners, device lists, and error summaries. When these primitives encode accessibility as a structural default, every consuming flow inherits those guarantees without requiring redundant implementation or per-feature auditing.

Component contracts must define requirements that go beyond visual specification. For a status or waiting banner, the contract must specify that state changes are announced programmatically without moving focus — implemented through ARIA live regions with an appropriate politeness level. WAI-ARIA defines three active live region priority properties — off, polite, and assertive — that govern how and when assistive technologies surface dynamic updates to users [5]. Testing against screen reader combinations has confirmed, however, that even basic live region behaviors such as nesting, prioritization, and update relevance produce inconsistent results across vendor combinations, with no screen reader fully passing

all live region test cases in evaluated configurations [5]. This empirical finding reinforces why component contracts must specify not only which live region property to use but also how the component should degrade when assistive technology support is partial.

Recovery screens deserve particular attention in component design. A recovery screen must present a meaningful heading upon entry so screen reader users receive immediate orientation, and the primary next action must be reachable by keyboard without traversing irrelevant content. The importance of heading structure as a navigation mechanism is documented through survey data: when asked how they navigate unfamiliar pages, screen reader users reported using headings whenever available at a rate of 52%, with an additional 24% reporting frequent use — making heading-based navigation the dominant strategy by a substantial margin [6]. By contrast, skip-to-main-content links — a commonly implemented alternative — were used whenever available by only 22% of respondents, with 19% reporting they seldom use them and 10% reporting they never do [6].Documenting these requirements as testable contracts — rather than design guidelines — creates verifiable artifacts. A component is not accessible because it passes a visual review; it is accessible because it satisfies a defined set of semantic, interaction, and announcement requirements verifiable through automated tooling, manual assistive technology validation, and regression tests. This shift in framing — from compliance to contract — is what makes accessibility stable as component systems evolve.

4. State Modeling and Focus Management Across Asynchronous Transitions

Asynchronous transitions are the highest-risk type of accessibility failures in any kind of distributed account access flow. This is when the next state of the flow depends on events outside the UI, like a second device confirmation, a risk signal from an identity service, or a reconciliation of session state between devices. Such transitions require the UI to react to nondeterministic timing signals. In these situations, the problem is not due to a markup error, but the fact that the specification does not define how the interface is expected to behave.The engineers must define the full state machine — all states visible to the user, interim and aborted states, and error states — before the interface is finalized. For each state, the engineers must agree on invariants: what its accessible name and role are, what its programmatic entry announcement is, whether to initially focus on content and where to focus, what is preserved and reset, and what recoveries are provided. These invariants are accessible, testable, and auditable, and they are naturally taken into account during design; as such, accessibility is built into the development process rather than added later.

This is corroborated by benchmarking of automated accessibility evaluation tools: human evaluators found 650 accessibility violations over 9 pages, and the best-performing tool identified 38% of them. The best-performing tool found a maximum of 50% of violated success criteria. The worst-performing tool found 14% of the violations identified by human evaluators [7]. This confirms that behavioral failures for dynamic state transitions, loss of focus, and asynchronous updates are beyond automated assessment and must instead be specified and validated through explicit state models and direct testing with assistive technology.The privacy and environmental costs of account access flow further complicate the matter. In transactional web use, experienced tracking was found to have the largest negative path coefficient (−0.346) of all conditions considered ($p < 0.01$), and the need for individual privacy can explain 45.7% of the variance in transactional web use behavior [8]. Recovery flows should therefore not only maintain structural continuity but also minimize perceived exposure by preserving session state, avoiding redundant requests for private information, and providing clear feedback about what information was retained or discarded during the recovery process.

5. Verification Strategies, Release Governance, and Regression Prevention

Automated verification is the only practical way to prevent accessibility regressions in a context of many teams frequently deploying updates, but this method must come with clear expectations of what it can and cannot provide, since the gulf between automated testing and thorough accessibility evaluation is well-documented.Lazar et al. (2007) surveyed 100 blind screen reader users and found that confusing page layouts and unlabeled form fields were among the most frequently reported barriers, confirming that the defects most harmful to assistive-technology users are routinely missed by automated checkers [12]. This pattern explains why accessibility regressions can persist across release cycles: the barriers exist in interaction patterns and state transitions that neither automated checks nor conventional defect triage reliably surface or prioritize.Sloan and Kelly (2011) reinforced this point from a measurement perspective, arguing that conformance-based approaches can never be sufficient on their own [9].

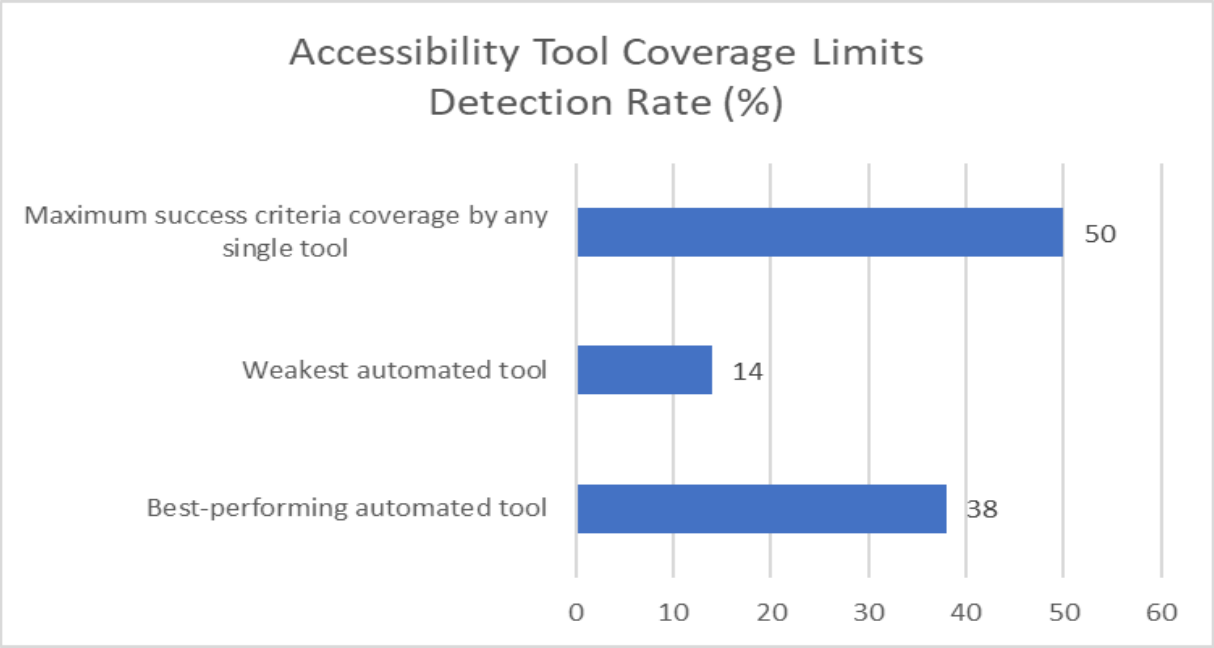
In order to create useful accessibility measures, the user experience must be part of the measure, not only compliance with technical requirements. Otherwise, conformance-based verification can produce false positives, because the outcome of automated checks may not correspond to the actual user experience.A defense-in-depth verification stack would therefore include static and runtime relationship and ARIA checks, keyboard reachability checks for every interactive state, focus-order and focus-visibility checks, semantic snapshot comparison tests against named journey states, and manual assistive technology tests for both critical asynchronous journeys and less commonly used recovery paths where behavioral failure is most likely to escape automation.Release governance binds all layers together with explicit gates and manual validations for critical journeys and feedback loops with incident management, with production accessibility failures treated as reliability incidents, ensuring that defects are episodic rather than chronic.

6. Social Implications and the Accessibility-Reliability Convergence

Socially, the case for defaulting account access experiences to a high level of accessibility is uncontroversial, as access to online services is required for engagement with modern society, whether banking, managing one's health, enrolling in education, applying for work, or accessing benefits from the public sector. All of these services have account access flows, which, as a class, are among the only pages some users may ever need to complete. The result of people with disabilities being consistently excluded from these flows is not inconvenience but enforced exclusion from infrastructure that non-disabled people can take for granted.The scale of the problem is demonstrated by Valtolina and Fratus (2022), who evaluated all 7,713 Italian municipal homepages against WCAG 2.0 and found only 12% with no conformance error and fewer than 5% with no possible WCAG 2.0 conformance error [10]. These are government service entry points — precisely the account access and authentication contexts where exclusion carries the most serious civic consequences. Most errors are highly concentrated: the single largest failure category, "Distinguishable" (insufficient color contrast and foreground-background separation), accounts for 53% of all identified errors, indicating that a small number of failure types produce the majority of access barriers across the evaluated corpus.There is regulatory recognition of these findings, with several jurisdictions enacting laws or regulations requiring digital accessibility and recognizing conformance to WCAG 2.1 Level AA as the applicable technical standard. Abou-Zahra et al. (2018) offer a more measured assessment: while rapid progress in AI-based image recognition, voice recognition, and natural language processing holds significant promise for web accessibility, current limitations in accuracy, accountability, and data sensitivity mean that AI cannot yet reliably replace the need for content authors to implement accessibility standards directly. The authors propose accessibility conformance evaluation — with its existing training data and commercial incentive — as a potential pathway to accelerate AI's gradual uptake in this domain, beginning with narrowly defined tasks before expanding to more complex interactions [11].The social dimension of this engineering work gives it urgency beyond technical conformance. Digital services have become the primary infrastructure for access to banking, healthcare, education, and employment, and account access flows are the gatekeeping layer of that infrastructure. Designing and operating those flows to be reliably accessible across the range of user abilities, input modalities, and network conditions — and to degrade gracefully when failures occur — is not a niche accommodation. It is a condition of sound systems engineering and a prerequisite for equitable digital participation[13].

Table 1: *Component Contract Requirements Beyond Visual Specification [5][6]*

Component Type	Contract Requirement	Verification Method
Status or waiting banner	State changes are announced programmatically without moving focus	Manual assistive technology validation
ARIA live regions	Specified politeness level appropriate to urgency	Automated tooling and screen reader testing
Recovery screen	Meaningful heading presented upon entry for immediate orientation	Semantic snapshot testing
Primary next action	Reachable by keyboard without traversing irrelevant content	Keyboard reachability checks
Interactive primitives	Accessibility encoded as a structural default	Regression testing and component auditing



Graph 2: *Automated Tool Detection Rate of Accessibility Violations [7, 8]*

Table 2: *Release Governance Components and Their Accessibility Functions [9][12]*

Governance Component	Function	Trigger Condition	Outcome if Bypassed
Explicit release gates	Enforce minimum verification before deployment	Every release affecting account access flows	Undetected regressions reach production
Manual validation requirements	Ensure that behavioral coverage automation cannot provide	Critical journeys and asynchronous state transitions	Dynamic failures invisible to automated checks ship undetected
Incident-driven feedback loops	Connect production failures to specific guardrail gaps	Accessibility barrier confirmed in production	Defect root cause unaddressed, failure recurs
Defect severity triage	Prioritize accessibility failures by user impact	On the identification of a production accessibility incident	High-impact barriers deprioritized against functional backlog
Guardrail strengthening	Update component contracts or test coverage after failure	Post-incident review	The same failure class recurs in future releases

7. Conclusions

Account access flows occupy a singular position in the accessibility risk landscape: a single inaccessible state — an unannounced focus reset, an error message without a programmatic association, a session timeout with no extension mechanism — can terminate an entire service relationship for a user with a disability, with no equivalent offline path available. This paper has argued that the engineering practices sufficient to prevent such failures are not novel, but they must be applied with the same specification discipline applied to functional correctness. Explicit state modeling that defines accessibility invariants for every reachable UI state, component-level contracts that encode semantic and announcement requirements as testable defaults, layered verification stacks that combine automated ARIA and keyboard checks with manual assistive technology validation, and release governance frameworks that treat production accessibility failures as reliability incidents are individually documented practices. Their value lies in their combination and consistent application across the full set of asynchronous, interrupted, and error states that distributed authentication flows produce.

The limitations of this framework are also specific. Automated tooling, even at its best-evaluated performance, detects fewer than half of the success criteria violations that human evaluators find [7], and behavioral failures in dynamic state transitions fall almost entirely outside its scope. Conformance measurement confirms the absence of known violation classes; it does not confirm that assistive technology users can complete critical journeys. Closing that gap requires user experience validation with assistive technology users on the specific interaction paths — asynchronous confirmations, mid-flow session recovery, error correction under

time constraints — where behavioral failure concentrates. The research agenda implied by this gap is clear: reliable methods for specifying, measuring, and governing the behavioral accessibility properties of distributed state systems remain underdeveloped relative to the engineering complexity of the systems they must cover. Progress on that agenda is both a technical necessity and a condition for the equitable delivery of services that now constitute essential digital infrastructure.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] World Health Organization [Online]. Available: https://iris.who.int/server/api/core/bitstreams/da982aa9-f135-4eff-9612-49590d58bf01/content

[2] Shawn Lawton Henry et al., "The role of accessibility in a universal web," ACM Digital Library, 2016. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2596695.2596719

[3] W3C Recommendation, "Web Content Accessibility Guidelines (WCAG) 2.2," [Online]. Available: https://www.w3.org/TR/WCAG22/

[4] Jonathan Lazar et al., "The SoundsRight CAPTCHA: An improved approach to audio human interaction proofs for blind users," ACM Digital Library, 2012. Available: https://dl.acm.org/doi/pdf/10.1145/2207676.2208385

[5] Peter Thiessen, "WAI-ARIA live regions and HTML5," ACM Digital Library, 2011. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1969289.1969324

[6] Yevgen Borodin et al., "More than meets the eye: A survey of screen-reader browsing strategies," ACM Digital Library, 2010. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1805986.1806005

[7] Markel Vigo et al., "Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests," ACM Digital Library, 2013. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2461121.2461124

[8] Ann D. Rensel et al., "Private Transactions in Public Places: An Exploration of the Impact of the Computer Environment on Public Transactional Web Site Use," Journal of the Association for Information Systems Vol. 7 No. 1, pp. 19-51/January 2006. [Online]. Available: https://www.researchgate.net/profile/June-Abbas/publication/220580492

[9] David Sloan and Brian Kelly, "Web accessibility metrics for a post-digital world," University of Bath, 2011. Available at: https://purehost.bath.ac.uk/ws/portalfiles/portal/223412/web-accessibility-metrics-2011.pdf

[10] Stefano Valtolina And Daniele Fratus, "Local Government Websites Accessibility: Evaluation and Findings from Italy," Digital Government: Research and Practice, 2022. https://doi.org/10.1145/3528380

[11] Shadi Abou-Zahra, et al., "Artificial Intelligence (AI) for Web Accessibility: Is Conformance Evaluation a Way Forward?" In Proceedings of the 15th International Web for All Conference, ACM Digital Library, 2018. https://dl.acm.org/doi/pdf/10.1145/3192714.3192834

[12]Jonathan Lazar et al., "What frustrates screen reader users on the web: A study of 100 blind users," International Journal of Human-Computer Interaction, 2007. https://web.archive.org/web/20100612034800id_/http://triton.towson.edu/~jlazar/IJHCI_blind_user_frustration.pdf

[13] Chetanya Rai . "Maintaining Michelin-star standards: A framework for pastry kitchen leadership and mentorship". *Review of Contemporary Philosophy*, Vol *23* No. 2, pp. 7940–7949. https://reviewofconphil.com/index.php/journal/article/view/1300