DOI 10.37539/2949-1991.2026.41.6.060

Kovalev Dmitry Andreevich,
Head of System Design Department,
Integration Technologies and Security LLC

PRACTICAL APPROACHES TO SECURE SOFTWARE DESIGN AND PROTECTION OF DISTRIBUTED ENTERPRISE SERVICES

Abstract. Contemporary enterprise software environments present a security design challenge that perimeter-hardening cannot resolve: distributed architectures expose authentication and authorization state to propagation delays, concurrent updates, and consistency trade-offs that collectively undermine the guarantees a point-of-entry credential mechanism was designed to provide. The article reviews the structural inadequacy of password-based session models in asynchronous multi-service environments, analyzes the post-password authentication ecosystem and its deployment realities, examines zero-trust architecture as an operational design constraint, explores the security semantics of concurrent transaction processing, and evaluates the Adaptive Consistency-Oriented Protocol as a mechanism for aligning performance with security requirements. These contributions are positioned against the broader literature on zero-trust migration, FIDO2 deployment barriers, and DevSecOps pipeline integration.

Keywords: Secure software design, distributed systems security, zero-trust architecture, FIDO2. behavioral authentication, microservices, transaction consistency, post-password authentication.

Introduction

Security in enterprise software has long been treated as a boundary condition: perimeter controls are hardened, credentials are validated at entry points, and the session that follows is presumed to inherit the identity asserted at login. The accumulated evidence from two decades of production incidents contradicts that presumption. Session hijacking, privilege escalation through stale tokens, lateral movement following credential reuse across services, and insider-threat exploitation of the gap between initial verification and subsequent use represent a class of attacks that perimeter-oriented design cannot address, because each attack begins after authentication has already succeeded.

Polezhaev (2026a) provides the formal analytical grounding for this problem. Authentication in distributed systems is a continuous property depending on the coherence of state propagation across services, shaped by the causal structure of event ordering rather than by the moment a credential was issued. When an identity artifact is evaluated by a downstream service, its meaning reflects the system state visible at that component's evaluation point, which may differ from the state at the time of original issuance. The theoretical basis lies in Lamport's formalization of logical clocks: in asynchronous distributed systems, event ordering is defined by causal dependencies, and identity assertions evaluated at different points in these timelines correspond to different system states (Lamport, 1978). A single credential leakage event can therefore initiate lateral movement across multiple services simultaneously, because each service evaluates the compromised artifact against its own local authorization state before revocation propagates.

The Structural Authentication Vulnerability in Distributed Environments

The central contribution of Polezhaev (2026a) to the authentication security literature is a reframing of where the failure occurs. Prior work, including Ahmad et al. (2015), treated secure software design primarily as a protocol selection and threat modeling problem, with the assumption that a correctly chosen credential mechanism would suffice. The monograph shifts the analytical frame to the consistency layer, demonstrating that any authentication model relying on a persistent

artifact evaluated against a static reference exhibits security degradation proportional to the degree of state asynchrony in the system, independent of the cryptographic strength of the credential itself. The monograph formalizes this through a taxonomy of four failure propagation paths. Credential leakage initiates lateral movement because a compromised credential is accepted by multiple services lacking synchronized knowledge of its status. Session token replay follows the same mechanism: a captured token is accepted by services whose revocation state has not yet incorporated the invalidation event. Delayed revocation propagation allows a principal whose access has been terminated locally to retain access in remote services until the revocation event reaches them. State inconsistency produces conflicting authorization decisions when access control data is being synchronized and different replicas carry different versions. Each pathway shares one structure: a local security event fails to propagate instantaneously to all enforcement points, producing measurable divergence between intended policy and observed behavior.

The practical implication is that security-relevant state, including role assignments, permission records, device compliance status, and session tokens, must be treated as distributed data subject to the same consistency guarantees applied to any other critical data in the system. Where consistency is relaxed for performance reasons, the resulting window of divergence carries a direct and quantifiable security cost, bounded below by propagation latency and extending upward with queue depth and replication lag. A WebAuthn-based passwordless flow eliminates credential reuse but leaves the established session carrying an identity artifact evaluated asynchronously across services. Lammers and Lichter (2025) address one facet of this problem by proposing automated pattern-based guidance for secure architecture decisions; the runtime consistency gap identified in the preceding analysis operates at a layer that design-time recommendations cannot fully anticipate.

The Post-Password Authentication Ecosystem and Its Operational Gaps

Chapter 3 of Polezhaev (2026a) provides an operational analysis of WebAuthn and FIDO2 deployments that revises the optimistic framing common in protocol-level security literature. The FIDO2 standard eliminates shared secrets by binding identity to device-resident asymmetric keys: the relying party stores only public key material and issues a challenge that the authenticator signs with the private key, which never leaves the device. The cryptographic security of this model under a compliant client is well established. Real deployments diverge from this model in structurally reproducible ways that generate measurable security gaps independent of the cryptographic layer.

Three gaps receive detailed treatment. The first concerns attestation validation: production deployments frequently accept a wide range of authenticators without strict differentiation between hardware-backed and software-only credentials, which means credentials created under substantially different security conditions are treated as equivalent by the relying party. The second concerns fallback and recovery mechanisms. Auxiliary authentication paths, including account recovery by email, administrative override, and support-desk procedures, are almost always present in enterprise deployments. Kuchhal et al. (2023) empirically confirm that these paths are targeted preferentially, because they require less effort to exploit than the primary FIDO2 flow. The composite security level of an account is therefore determined by the weakest mechanism that can produce access, not by the primary cryptographic channel. The third gap concerns multi-device identity expansion: users who register multiple authenticators create multiple independent entry points, none of which the system ranks by assurance level, with platform authenticators used at higher frequency and consistently lower authentication strength as a result.

Lassak et al. (2024), drawing on 28 semi-structured interviews with CISOs and authentication managers, independently confirm that enterprise passkey deployment faces structural barriers corresponding precisely to these gaps: account recovery design, fallback authentication, legacy system integration, and incomplete tooling for credential lifecycle management. The barriers arise from the structural properties of identity in heterogeneous distributed environments. Ghorbani

Lyastani et al. (2020) establish in a controlled usability study that FIDO2 offers genuine advantages over passwords in the primary authentication flow; Kuchhal et al. (2023) demonstrate that malware-based attacks can bypass those protections by exploiting the session layer that follows authentication, the same post-authentication window characterized through the failure propagation taxonomy in Polezhaev (2026a). Kepkowski et al. (2023) independently identify similar deployment challenges, finding that the most technically sophisticated authentication paths are avoided in favor of more convenient alternatives, consistently lowering the effective assurance level of deployed systems.

Zero-Trust Architecture as a Design Constraint

Zero-trust architecture resolves the implicit-trust problem of perimeter-oriented designs by eliminating the assumption that network location confers identity validity, replacing it with continuous evaluation of identity, device posture, and contextual signals for each request. Rose et al. (2020) formalize this in NIST SP 800-207 through tenets organized around policy decision points, policy enforcement points, and continuous monitoring. The analytical challenge that this framework leaves unaddressed is identified in Chapter 4 of Polezhaev (2026a): the temporal structure of trust evaluation in distributed deployments, where the signals used for continuous verification are themselves subject to propagation delays.

Zero-trust compresses the temporal window of identity validity to the scope of a single request, which means identity is reconstructed at each evaluation point by aggregating the current values of a continuously updated signal set. This reconstruction depends on the consistency of those signals across the distributed enforcement points participating in the evaluation. When a device compliance update, a contextual risk score, and a role attribute are each maintained by different services with independent update cycles, the access decision at any given enforcement point reflects a combination of signal values that arrived at different times. The result is structurally reproducible divergence in access outcomes across enforcement points, regardless of how rigorously the zero-trust policy itself is defined.

Teerakanok et al. (2021) examine the migration challenge, identifying consistency management and legacy system integration as the primary technical obstacles to zero-trust adoption, though without a formal account of the underlying dynamics. Buck et al. (2021) arrive at a similar finding across a broader multivocal review: zero-trust research is concentrated at the conceptual and architectural levels, with limited empirical investigation of deployed system behavior under real-world load. Fernandez and Brazhuk (2024) carry the critique further, arguing that zero-trust implementations frequently replicate structural vulnerabilities of earlier models at a higher level of abstraction, and that continuous verification requirements are often satisfied by superficial signal checks.

Adanigbo et al. (2024), examining zero-trust in multi-cloud microservice environments, independently identify integration complexity and lack of centralized visibility as critical failure points, consistent with the signal propagation analysis. Cao et al. (2024) address the automation and orchestration challenge, proposing AI-driven orchestration as a mitigation for operational cost at scale. These contributions address the policy management layer; the architectural problem of consistency in state propagation across enforcement points remains a distinct open challenge in the literature.

The Adaptive Consistency-Oriented Protocol

Chapter 8 of Polezhaev (2026a) presents the Adaptive Consistency-Oriented Protocol (ACOP), an architectural contribution that addresses the security-consistency trade-off identified in preceding chapters. The protocol co-locates a validation sidecar with each microservice instance. This sidecar intercepts write operations through a loopback interface before they reach persistent storage and converts each state transition into a verifiable transaction artifact containing the transition payload, its causal context, a cryptographic hash, and prediction entries derived from a continuously updated probabilistic model of peer state.

Commitment in ACOP proceeds through distributed endorsement accumulation, with no central coordinator. Each transaction artifact is propagated to peer sidecars, which evaluate the

proposed transition against their local state and issue endorsements when the transition satisfies both structural consistency and domain security constraints. A transaction reaches committed state once a configurable endorsement threshold is satisfied. Because endorsement collection proceeds in parallel across peers, the latency overhead of validation is substantially lower than in coordinator-based protocols, where each transaction traverses a sequential bottleneck.

The performance comparison reported in Chapter 8 is quantitatively specific. Classical two-phase commit protocols exhibit end-to-end commit times of 120 to 250 milliseconds under moderate load. Leader-based consensus protocols such as Paxos and Raft reduce this to approximately 80 to 150 milliseconds, but introduce throughput degradation as leader queues saturate under high concurrency. The ACOP distributed endorsement architecture maintains more than 94 percent of baseline throughput under high concurrency, compared to 65 to 80 percent for coordinator-based systems under equivalent load, and reduces average commit latency by approximately 35 to 50 percent relative to leader-based consensus. Prediction accuracy exceeds 90 percent under sustained load, allowing dependent operations to proceed without blocking on peer confirmation in the majority of cases, with rollback applied only to the mispredicted fraction.

The security semantics of ACOP differ from those of eventual consistency models in a consequential way. In eventually consistent systems, intermediate states are externally visible to authorization components, and convergence is guaranteed only over time. In ACOP, state transitions become externally visible to the authorization layer only after endorsement and distributed validation complete. Authorization decisions are therefore anchored to endorsed state, eliminating the class of vulnerabilities arising from authorization components consuming unverified intermediate state. The distinction is between a temporal convergence guarantee and a structural endorsement guarantee: state has been independently validated by multiple participants before it participates in any access decision.

Two limitations of the current evaluation are worth noting. The performance figures derive from a controlled experimental environment; behavior under network partitions, high-churn membership, and Byzantine fault conditions is not characterized in the published results. Additionally, the endorsement threshold is presented as a design parameter without a formal analysis of how threshold selection affects the latency-fault-tolerance trade-off under varying replica counts. These are natural targets for follow-on empirical work.

Lombardi and Fanton (2023) address a related but distinct problem: the integration of security validation into CI/CD pipelines at the pre-deployment stage. Their CyberDevOps architecture embeds vulnerability assessment and compliance enforcement into the build process, covering the phase before services reach production. ACOP covers the runtime phase, where authorization decision quality depends on the consistency of live system state. The two contributions address non-overlapping stages of the security lifecycle and are accordingly complementary.

Continuous Behavioral Authentication as a Post-Authentication Security Layer

The article by Polezhaev (2026b) approaches the post-authentication vulnerability window from a direction orthogonal to the consistency-oriented architecture. The article develops a multi-modal behavioral authentication framework that provides continuous identity verification during an active session by monitoring keyboard and pointer interaction patterns, without requiring additional user interaction under ordinary conditions.

The theoretical grounding is in motor learning neuroscience: extensively practiced input interaction sequences consolidate into individual-specific low-level control circuits whose output characteristics are stable within subjects and discriminative between subjects. The framework integrates five behavioral signal channels, specifically keystroke timing dynamics, mouse trajectory, pointer dwell, scroll rhythm, and application transition timing, into a 114-dimensional feature vector evaluated at five-second intervals. A two-tier stacked ensemble classifier combines five modality-specific one-class support vector machines and five modality-specific LSTM networks encoding 60-

second temporal behavioral context. A gradient-boosted meta-classifier trained on a population including adversarially generated mimicry samples produces a Platt-scaled confidence score. Per-user threshold calibration sets the upper threshold at the 5th percentile of each enrolled user's own in-enrollment score distribution, replacing the population-level fixed thresholds that generate high false rejection rates in prior single-modality systems.

Evaluation across 64 users yielded a False Acceptance Rate of 0.28 percent, a False Rejection Rate of 1.76 percent, an Equal Error Rate of 1.02 percent, and a mean identity substitution detection latency of 31.4 seconds. The mimicry attack FAR of 0.003 percent is notable for the adversarial realism of the test condition: trained human mimics with prior observation access to target users' behavioral patterns were used, as opposed to naive impostor sessions typical of behavioral biometrics evaluations. The EER reduction from the keystroke-only baseline of 3.50 percent to 1.02 percent, a 70.9 percent improvement, indicates that the five behavioral channels contribute substantially independent discriminative information.

A relevant limitation of the evaluation is the sample of 64 users with controlled session lengths. The behavioral individuality literature supports inter-individual distinctiveness at this scale, but deployment in enterprise environments with thousands of users engaging in diverse task patterns, including irregular sessions, extended absences, and injury-related behavioral change, would require longitudinal validation that the current study does not provide.

Implications for Enterprise Security Architecture

The analytical framework reviewed here yields engineering principles that reposition security as a runtime consistency problem as much as a protocol selection problem. Authentication must be treated as a system property coupled to consistency mechanisms that propagate revocation and permission changes with bounded latency, with authorization decisions anchored to endorsed state. The post-password authentication ecosystem must be evaluated at the level of the entire authentication surface, including fallback mechanisms and recovery paths: the weakest path determines the composite assurance level. Concurrent updates to authorization-relevant state must be resolved through mechanisms that embed domain security constraints, not generic recency or merge rules.

These principles align with and extend the research of Rose et al. (2020), who supply the normative zero-trust framework but do not address runtime consistency semantics. They qualify the critical analysis of Fernandez and Brazhuk (2024), who correctly identify that naively implemented zero-trust reproduces structural vulnerabilities, by providing a formal account of the mechanism and a concrete architectural response in the form of ACOP. The empirical work of Lassak et al. (2024) and Kuchhal et al. (2023) on FIDO2 deployment barriers confirms the operational relevance of the failure propagation taxonomy in Polezhaev (2026a): the deployment problems practitioners report correspond structurally to the categories identified in that work.

The competitive dimension of the argument in Polezhaev (2026b) merits independent examination. The claim is that continuous behavioral authentication, implemented as an ambient background process, provides verifiable post-authentication identity continuity that point-of-entry mechanisms cannot supply, and that this constitutes a measurable differentiator in procurement contexts where security posture is a purchasing criterion. The argument is internally consistent, but its market validity depends on procurement practices that vary significantly across sectors and jurisdictions. In highly regulated environments such as financial services and healthcare in the EU, where audit requirements specify what security controls must be documented and reported, the argument carries weight.

Conclusions

The security of distributed enterprise services depends on properties that conventional authentication models are not designed to provide: the consistency of authorization-relevant state across asynchronous services, the bounded propagation of revocation and policy updates, the

semantic validity of concurrently resolved state transitions, and the continuity of identity verification throughout active sessions. The body of work reviewed here supplies both the formal characterization of these requirements and a set of concrete architectural responses, positioned within a broader literature that establishes normative frameworks but leaves the runtime consistency problem largely unaddressed. Existing research on zero-trust architecture establishes normative frameworks and identifies migration challenges but does not provide a formal account of how distributed state dynamics generate security gaps within correctly designed zero-trust policies. Research on FIDO2 deployment documents operational barriers without connecting them to the consistency properties of the underlying distributed infrastructure. The connecting analysis, showing that empirically observed deployment gaps are structurally necessary consequences of the consistency trade-offs distributed architectures make, is the primary theoretical contribution of the work reviewed here.

The critical analysis of Fernandez and Brazhuk (2024), that zero-trust implementations can reproduce structural vulnerabilities of earlier security models, receives both support and qualification from this framework. The formal support lies in the consistency dynamics that explain precisely how superficially compliant implementations fail. The qualification lies in the architectural response: ACOP provides the tools, specifically endorsed state commitment, predictive consistency management, and semantic conflict resolution, that can resolve the structural problem at the enforcement layer. Future research should address the behavior of ACOP under network partitions and Byzantine fault conditions, the formal verification of the endorsed-state commitment security model, and the interaction between ACOP and existing service mesh infrastructures such as Istio and Linkerd. Longitudinal evaluation of the behavioral authentication framework across diverse enterprise user populations, particularly under behavioral change events such as injury and extended absence, would substantially strengthen the empirical basis of the approach.

References:

1. Adanigbo, O. S., Adekunle, B. I., Ogbuefi, E., Odofin, O. T., Agboola, O. A., & Kisina, D. (2024). Implementing zero trust security in multi-cloud microservices platforms: A review and architectural framework. International Journal of Advanced Multidisciplinary Research and Studies, 4(6), 2402–2409. https://doi.org/10.62225/2583049X.2024.4.6.4357

2. Ahmad, Z., Asif, M., Shahid, M., & Rauf, A. (2015). Implementation of secure software design and their impact on application. International Journal of Computer Applications, 120(10), 8–15. https://doi.org/10.5120/21261-3355

3. Buck, C., Olenberger, C., Schweizer, A., Völter, F., & Eymann, T. (2021). Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. Computers & Security, 110, 102436. https://doi.org/10.1016/j.cose.2021.102436

4. Cao, Y., Pokhrel, S. R., Zhu, Y., Doss, R., & Li, G. (2024). Automation and orchestration of zero trust architecture: Potential solutions and challenges. Machine Intelligence Research, 21, 294–317. https://doi.org/10.1007/s11633-023-1456-2

5. Fernandez, E. B., & Brazhuk, A. (2024). A critical analysis of zero trust architecture (ZTA). Computer Standards & Interfaces, 89, 103832. https://doi.org/10.1016/j.csi.2024.103832

6. Ghorbani Lyastani, S., Schilling, M., Neumayr, M., Backes, M., & Bugiel, S. (2020). Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication. In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 268–285). IEEE. https://doi.org/10.1109/SP40000.2020.00047

7. Kepkowski, M., Machulak, M., Wood, I., & Kaafar, D. (2023). Challenges with passwordless FIDO2 in an enterprise setting: A usability study. In 2023 IEEE Secure Development Conference (SecDev) (pp. 37–48). IEEE. https://doi.org/10.1109/SecDev56634.2023.00017

8. Kuchhal, D., Saad, M., Oest, A., & Li, F. (2023). Evaluating the security posture of real-world FIDO2 deployments. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (pp. 2381–2395). ACM. https://doi.org/10.1145/3576915.3623063

9. Lammers, D., & Lichter, H. (2025). SecuRe: An approach to recommending security design patterns. In Proceedings of the 22nd IEEE International Conference on Software Architecture (ICSA 2025), 186-189. https://doi.org/10.1109/ICSA-C65153.2025.00037

10. Lassak, L., Pan, E., Ur, B., & Golla, M. (2024). Why aren't we using passkeys? Obstacles companies face deploying FIDO2 passwordless authentication. In Proceedings of the 33rd USENIX Security Symposium (pp. 7231–7248). USENIX Association.

11. Lombardi, F., & Fanton, A. (2023). From DevOps to DevSecOps is not enough. CyberDevOps: An extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. Software Quality Journal, 31, 619–654. https://doi.org/10.1007/s11219-023-09619-3

12. Polezhaev, A. (2026). Secure software design: Engineering applications for the post-password era. LAP Lambert Academic Publishing.

13. Polezhaev, A. (2026). Security as a feature: The next competitive advantage in software products. Austrian Journal of Technical and Natural Sciences, (5–6).

14. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

15. Teerakanok, S., Uehara, T., & Inomata, A. (2021). Migrating to zero trust architecture: Reviews and challenges. Security and Communication Networks, 2021, 9947347. https://doi.org/10.1155/2021/9947347.