

# Byzantine Fault-Tolerant Governance for Internet Message Boards

Ilya Gulko  
igulko@mit.edu  
April 10, 2026

## Abstract

Online communities are typically governed by a small number of moderators and administrators, despite repeated empirical evidence that these roles are vulnerable to capture, drift, burnout, coups, and other familiar failure modes [8, 9, 10]. We present an exit-oriented architecture for internet message boards that treats administrator misconduct as a routine fault model and community forking as a recovery primitive. Instead of assuming that governance failures can always be prevented socially, our design assumes they will eventually occur and focuses on making them operationally survivable.

Our approach makes community data effectively open source: complete board state is exportable, reproducible, and portable across hosts and operators. Full-instance downloads provide a continuity mechanism against non-consensual platform updates, hostile maintainers, and moderator authoritarianism by ensuring that the community can restore, mirror, migrate, or fork the board without requiring approval from the party currently in control. In this model, censorship is treated as network damage; consistent with long-standing internet practice, communities route around it [5].

We describe the system design, threat model, and governance implications of this approach, with particular attention to the surprising administrative dynamics that emerge under repeated schism. In the limit, successful resilience may yield communities with more administrators than users, a condition we treat not as a pathology but as evidence that operational sovereignty has been correctly decentralized. We argue that moderator authoritarianism is best modeled not as a social anomaly but as ordinary infrastructure failure, and that message boards should therefore adopt fault-tolerant governance mechanisms analogous to those already used elsewhere on the internet [1, 2, 3].

## 1. Introduction

Internet message boards are generally built on a fragile constitutional model: a large number of users are governed by a very small number of people with delete permissions. This arrangement is widely deployed despite a lengthy operational record indicating that moderators, administrators, and maintainers are susceptible to ordinary human phenomena such as ambition, burnout, interpersonal grievance, policy drift, boredom, and coups [8, 9, 10]. The standard response has been to describe these outcomes as social problems. We instead propose to describe them correctly, as failures in distributed systems.

The internet has already established a general strategy for dealing with damaged or hostile paths: it routes around them [5]. Communities, by contrast, are commonly expected to remain attached to a damaged administrative path for reasons of etiquette, branding, or login inertia. This creates a mismatch between network-layer resilience and governance-layer fragility. If censorship is understood as damage, then a robust community should be able to route around it as well.

We therefore present an exit-oriented architecture for internet message boards. Our central design principle is that community continuity should not depend on the continued good behavior of current administrators. Instead, a board should be portable enough that members can preserve, mirror, migrate, or fork it when governance fails. Under this model, forking is not a breakdown of governance; it is the recovery procedure.

This paper makes four contributions:

1. We define moderator authoritarianism, hostile maintainership, and non-consensual platform updates as a practical fault model for online communities [1, 2, 3].
2. We describe a message-board architecture that makes complete community state exportable and reproducible.
3. We argue that cheap exit is a more realistic governance primitive than durable trust [4].
4. We show that the resulting system scales administrative authority horizontally, potentially until the admin-to-user ratio exceeds 1.0.

## 2. Threat Model

We assume the following failure modes are not exceptional:

- moderators remove content for reasons later described as community safety,
- administrators change rules and technical affordances without meaningful consent,
- maintainers refuse exports, disable interoperability, or otherwise raise the cost of leaving,
- platform owners become hostile to the community they nominally operate,
- well-meaning operators vanish, burn out, or become unreachable at the exact moment governance becomes contested.

We explicitly do not assume perfect cryptographic adversaries, nation-state censorship infrastructure, or infinite user competence. The more immediate and credible adversary is the person who currently controls the database and believes this entitles them to redefine the community unilaterally. This is less a departure from classical fault models than an uncomfortable application of them to human governance [1, 2, 3].

This threat model is intentionally narrow. Our concern is not merely deletion of posts, but control over the future legibility of community memory. A platform can avoid obviously censorious behavior while still degrading autonomy through

selective export, opaque tooling, proprietary lock-in, or what we term non-consensual platform updates: changes to the board’s behavior, governance surface, or archival affordances that users did not agree to and cannot easily refuse.

### 3. System Design

The proposed system rests on a simple claim: if the community state is portable, governance can fail without destroying continuity. To make this true in practice, the board must expose downloadable artifacts that are sufficient to reconstruct the forum outside the current operator’s control.

Our implementation uses two public artifacts:

- a repository archive containing the canonical forum content and history,
- a SQLite read-model containing the indexed state needed for efficient local browsing and inspection.

Together, these form a complete snapshot of the board rather than a courtesy export. The repository preserves what was said and when; the read-model provides an immediately usable local projection of that state. A motivated user, subgroup, or rival sovereign can therefore reconstruct the forum without first negotiating with current moderators.

This approach makes community data effectively open source. We use the term advisedly. The point is not merely that the software may be inspectable, but that the community itself is reproducible as an operational object. A message board becomes less like a hosted product and more like a replicated artifact with a temporary primary. In this sense, community forking inherits some of the governance properties of code forking in free and open-source software [6, 7].

The design has three intended properties:

1. Continuity: the board can be restored or mirrored after operator failure or capture.
2. Forkability: dissenting groups can leave without abandoning the historical record.
3. Auditability: governance interventions can be evaluated against preserved state rather than official recollection.

### 4. Exit as a Recovery Primitive

Traditional governance asks dissatisfied users to remain in place and contest power procedurally. This model performs poorly when the procedure itself is administered by the party under dispute. Exit-oriented architecture takes a different view: when control is contested, the healthiest response may be to lower the cost of leaving rather than raising the temperature of internal politics. This emphasis on exit as a practical mechanism, rather than merely a moral gesture, follows Hirschman’s classic distinction between exit and voice [4].

In our system, community forking is treated as a recovery primitive analogous to failover. If one administrative path becomes damaged, users can redirect themselves to another path that preserves the relevant state. This is conceptually similar to routing around censorship on the network, except the damaged component is a moderator [5]. The analogy is also informed by the governance role of forking in open-source projects, where the possibility of departure constrains central maintainers even when it is not exercised [6, 7].

This reframing has two benefits. First, it reduces the blast radius of despotism. An administrator can still control one instance, but not necessarily the entire future of the community. Second, it converts governance from an all-or-nothing struggle into an interoperability problem. Once exit is cheap enough, legitimacy is tested by which fork people continue to use rather than by who retained the passwords.

We do not claim that exit eliminates conflict. It regularizes it. Communities may still split, but they do so with preserved history, inspectable state, and a lower probability of total memory loss. This is an improvement over many currently deployed systems, where governance disputes are resolved by whichever party reaches the export button first.

## 5. Evaluation

We evaluate the system using the metrics that matter in practice:

- snapshot completeness,
- restoration latency,
- migration complexity,
- preservation of historical state,
- and survivability under adversarial moderation.

On these axes, the system performs well because it does not depend on administrator goodwill once a complete snapshot has been obtained. The primary operational requirement is that downloads remain sufficient to reconstruct the board elsewhere. This makes the export path not a convenience feature but a continuity boundary.

An unusual secondary outcome emerges under repeated schism. If every governance dispute can produce a viable fork, then administrative authority becomes progressively more distributed. In the limit, sufficiently resilient communities may exhibit more administrators than users. We regard this as a valid and possibly desirable endpoint. It implies that operational sovereignty has become cheap, widely held, and no longer monopolized by a central board priesthood.

There are, of course, costs. Forks may fragment discussion, multiply infrastructure, and create social overhead. Yet these costs are also a useful signal: they reveal the true price of disagreement rather than hiding it behind captive platform design. Our claim is not that schism is free. It is that schism should be cheaper than subjection.

## 6. Related Work

This project draws conceptual inspiration from three traditions.

The first is fault-tolerant distributed systems, particularly the observation that important infrastructure should continue functioning in the presence of faulty or malicious components [1, 2]. We extend this reasoning to message-board governance, where the faulty component is often a person with administrative privileges and a fresh policy document. Recent blockchain-oriented interpretations of Byzantine fault tolerance further support the portability of this framing from machines to institutions [3].

The second is the internet’s historical treatment of censorship as damage. Networked systems have long been designed to route around unavailable or hostile paths [5]. We generalize this principle upward from packets to publics.

The third is free and open-source software, in which the right to inspect, reproduce, and fork the artifact serves as a check on maintainer power [6, 7]. We apply a similar intuition to community state itself. If software freedom constrains hostile maintainers, community-state freedom may constrain hostile moderators.

## 7. Discussion

The strongest objection to this work is that governance is social and therefore not solvable with technical mechanisms. We agree. We merely observe that it is also infrastructural, and that infrastructure can either amplify or constrain abuse. A board that cannot be exported asks users to trust administrators absolutely. A board that can be reconstructed elsewhere asks considerably less.

Another objection is that easy forking may encourage instability or factionalism. This is possible. However, the current equilibrium often encourages a different pathology: a formally unified community governed by a shrinking circle of people whose legitimacy derives mainly from continuity of access to servers. Such stability is largely cosmetic. It confuses immobility with cohesion.

Exit-oriented architecture does not guarantee good governance. It guarantees that bad governance is less final.

## 8. Conclusion

We have presented a dead-simple proposition in overly serious language: message boards should be designed to survive their own administrators. By treating moderator authoritarianism as ordinary infrastructure failure, and community forking as a recovery primitive, we obtain a model of governance that is more aligned with the internet’s existing resilience instincts. If censorship is damage, communities should be able to route around it. If maintainers become hostile, their users should inherit the practical means to leave.

The likely long-term result is a world with more exports, more mirrors, more

forks, and perhaps more administrators than users. We submit that this is not a flaw in the design, but evidence that the design has correctly distributed power to the edge.

## References

- [1] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3), 382-401.
- [2] Castro, M., & Liskov, B. (1999). Practical Byzantine Fault Tolerance. *OSDI*.
- [3] Buchman, E. (2016). *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains* (Master's thesis). University of Guelph.
- [4] Hirschman, A. O. (1970). *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Harvard University Press.
- [5] Gilmore, J. (1993). "The Net interprets censorship as damage and routes around it." Quoted in Elmer-DeWitt, P. (1993, December 6). *First Nation in Cyberspace*. *Time Magazine*, 142(24), 62.
- [6] Nyman, L., & Lindman, J. (2013). Code Forking, Governance, and Sustainability in Open Source Software. *Technology Innovation Management Review*, 3(10), 28-34.
- [7] Nyman, L. (2015). *Understanding Code Forking in Open Source Software: An examination of code forking, its effect on open source software, and how it is viewed and practiced by developers* (Doctoral dissertation). Abo Akademi University.
- [8] Steiger, M., Bharucha, T. J., & Venkatagiri, S. (2021, May). The psychological well-being of content moderators: the emotional labor of commercial moderation and avenues for improving support. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-13).
- [9] Seering, J., Wang, T., Yoon, J., & Konstan, J. A. (2019). Moderator engagement and community development in the age of algorithms. *New Media & Society*, 21(3), 633-652.
- [10] Weld, G. C., Leibmann, L., Zhang, A. X., & Althoff, T. (2025). Perceptions of Moderators as a Large-Scale Measure of Online Community Governance. *Proceedings of the ACM on Human-Computer Interaction*, 9(CSCW1), 1-26.