

# Why Organizations Become Rigid (and What Actually Helps)

*A systems lens on rigidity, adaptation, and organizational design*

Michael Voorhaen

Organizations rarely become rigid because of one obviously bad decision. They become rigid because a series of reasonable choices start reinforcing each other.

Teams narrow their focus to reduce cognitive load, communication gets more local, goals tighten, routines stabilize, and leaders centralize under pressure. Each move makes sense on its own, or at least it does while you are making it. Together, they make adaptation harder.

The basic point is simple: rigidity is usually a system, not a single failure. Once you can see that system more clearly, you can also start to see which design moves help keep it from hardening.

I came to this through organizational design rather than psychology. I had been spending a lot of time with design frameworks, trying to understand why some of them felt clarifying and useful while others made me uneasy even when I could see their appeal.

Around the same time, I was reading Steven Hayes' [\*A Liberated Mind\*](#) and David Marquet's [\*Leadership Is Language\*](#), and something about the two started to click for me.

[Hayes](#) gave me a language for what happens when rules outlive the reality they were meant to serve. [Marquet](#) sharpened the difference between execution and decision-making, and why the same habits do not serve both. A little later, Daniel Kahneman's [\*Thinking, Fast and Slow\*](#) ([Kahneman, 2011](#)) made the attraction of rules, routines, and familiar interpretations look even more understandable: they reduce effort, create clarity, and spare us from having to reconsider the situation from first principles each time.

That combination stayed with me because, if it is even partially right, the consequences for agility are significant. Organizations that continue to act from yesterday's assumptions do not just make poorer decisions; over time, they can lose the capacity to notice when adaptation is needed at all.

This is not a formal study or a complete theory. It is a simplified lens I find useful for bringing several patterns into one picture. It is incomplete by design, and that is fine. It is not meant to be right in every detail. It is meant to make a recurring pattern easier to see, and easier to challenge before it hardens.

## Why organizations become rigid

### Rules outlive reality

You can see it in a team that keeps honoring a boundary or a rule even after the real problem has clearly started crossing it.

In *A Liberated Mind*, [Hayes](#) gives a name to one part of the pattern: the insensitivity effect. People can keep following rules even when direct experience suggests the rules no longer fit. Research on teams shows the same thing at group level. When teams fail to update their shared understanding, they adapt badly in changing environments ([Uitdewilligen et al., 2013](#)).

That matters because organizations are full of rules. Some are explicit, some are structural, some are conversational, and some are just habits that no one notices anymore. The problem is usually not structure itself. It begins when structure stops being questioned.

Once rules settle in, attention narrows around them. Familiar interpretations get easier to defend. Routines form around them. [Gersick and Hackman \(1990\)](#) show how quickly a useful solution can become a preferred pattern simply because it reduces uncertainty and effort.

What [Marquet](#) offers here is a useful distinction between redwork and bluework. Redwork benefits from reducing variation. If the work is execution, safety, or repeatability, rules help. Bluework is different. It includes noticing change, questioning assumptions, and reconsidering whether the current approach still fits reality. The problem starts when organizations bring a redwork logic into bluework and then wonder why learning slows down.

## Expertise hardens

Rules are only part of the story. Over time, expertise hardens the pattern further.

Research on expertise shows the trade-off clearly. Deep experience makes people faster and more effective inside a domain, but it can also make reframing harder when the situation changes ([Dane, 2010](#)). At team level, we see the same drift. Long-tenured groups reduce outside communication, drift into Not Invented Here (NIH) patterns, defend their own ways of working, and adapt more slowly when they stop revising their shared understanding ([Katz & Allen, 1982](#); [Uitdewilligen et al., 2013](#)).

That is one reason narrower scope feels so attractive. It reduces cognitive load. Repetition builds fluency. Stable local work feels efficient. Teams get rewarded for the very behaviors that later reduce adaptability. [De Smet and Grcic \(2023\)](#) make the counterpoint explicit: cognitive load is real, but narrowing scope is not the only way to deal with it. You can also manage load through better information flow and better design.

## How the pattern reinforces itself

None of these patterns fully explain rigidity on their own. The real explanatory value appears when you connect them.

One way to make sense of this is to look at it as a system. Not in a formal modeling sense, but in the simpler idea that recurring outcomes are produced by structure: by the relationships between parts, the way information moves, and the feedback loops through which actions create conditions for more of the same. Systems thinking, in that sense, asks a different question. Not just what happened, but what pattern is being produced, what structure is producing it, and where the leverage points for changing it might be.

The useful move here is to trace how the variables interact over time, how they reinforce or counteract one another, and where the structure offers leverage for changing the pattern. That is the move behind the systems language in this essay, and it is also the thread that runs through books like [Senge's \*The Fifth Discipline\* \(1990\)](#) and [Meadows's \*Thinking in Systems\* \(2008\)](#). A team narrows scope to cope. That reduces outside signals. With fewer outside signals, updating gets weaker. Once updating gets weaker, even more narrowing can start to feel sensible. Seen that way, rigidity stops looking like one mistake and starts looking like a reinforcing pattern.

Different fields describe different parts of this pattern. Seen together, they tend to cluster around a few broad themes:

- **Threat rigidity** Pressure narrows control, constricts information processing, and pushes organizations toward familiar responses ([Staw, Sandelands, & Dutton, 1981](#)).
- **Goal myopia** Targets narrow attention, reward local optimization, and crowd out non-goal signals.
- **Insensitivity / updating failure** Signals stop getting noticed or integrated well, and outdated interpretations persist.
- **Cognitive entrenchment** Routines, expertise, and familiar schemas harden over time.
- **Social identity dynamics** Structural boundaries can acquire social-identity significance, making in-group favoritism and out-group differentiation more likely ([Tajfel & Turner, 1979](#)).

Figure 1 summarizes those themes before the full model.

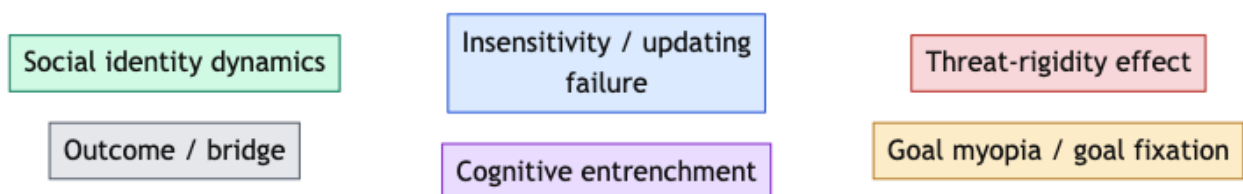


Figure 1. Legend: the main themes behind the rigidity model.

When those themes are translated into model nodes, some labels are shorthand. Threat / pressure, Stable composition / bounded domains, and Goal pressure / narrow targets describe the starting conditions. Cross-boundary information flow, Attention to outside cues / signals, and Mental-model updating describe what the system still notices. Routine persistence, Cognitive entrenchment, Social identity / in-group salience, and NIH / boundary defense / external rejection describe how it hardens.

If you connect these patterns, you get something like this:

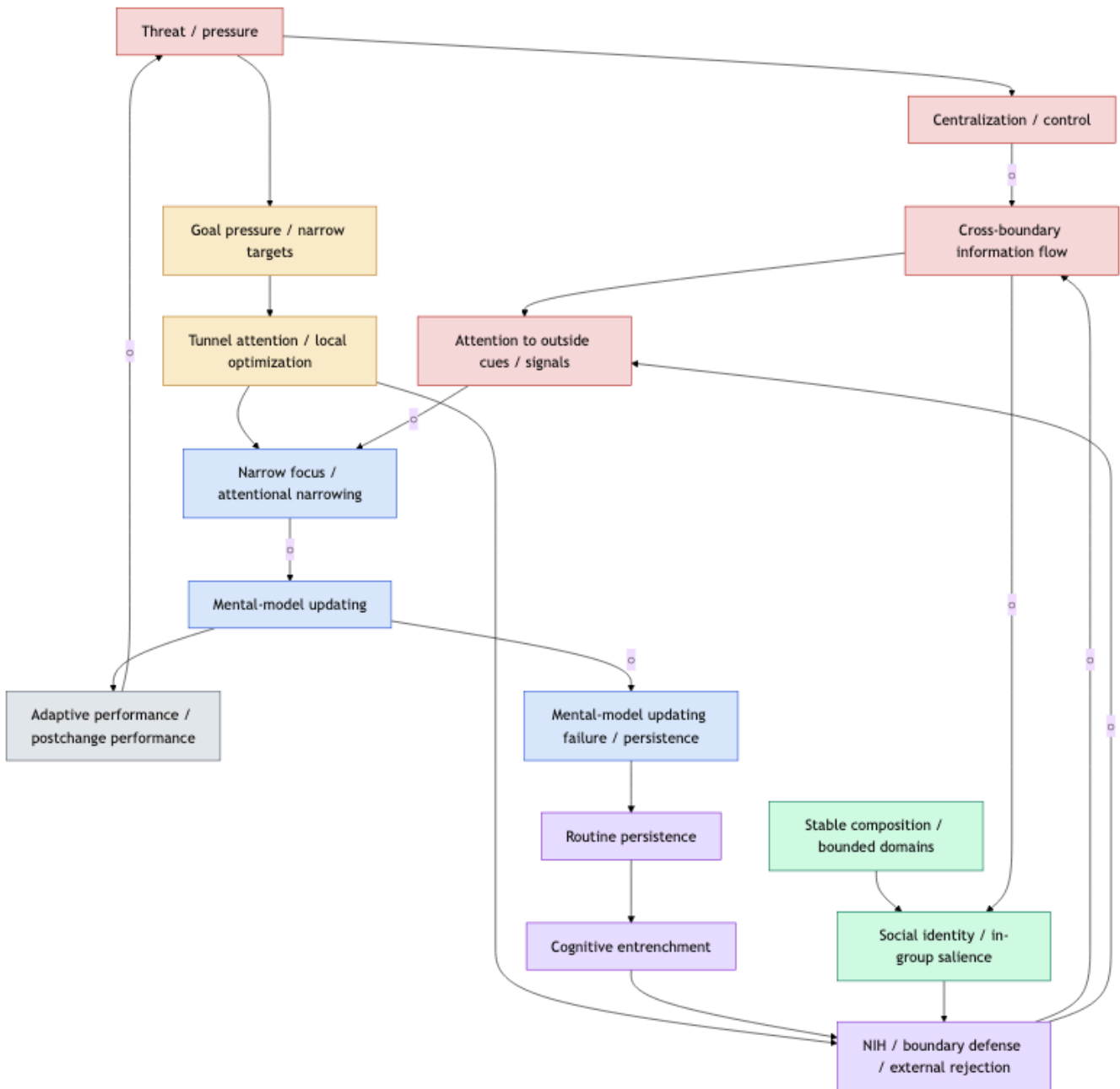


Figure 2. A simple model of how rigidity reinforces itself.

The easiest way to read Figure 2 is one loop at a time. Start with Threat / pressure. Under pressure, organizations tend to centralize. More Centralization / control usually means less Cross-boundary information flow. With less information moving across boundaries, there is less Attention to outside cues / signals. Attention narrows. Mental-model updating drops. Adaptive performance / postchange performance gets worse. Poorer performance creates more pressure. By the time the loop closes, the system has made the original problem harder to escape.

In compact form, that loop looks like this:

- more Threat / pressure → more Centralization / control → less Cross-boundary information flow → less Attention to outside cues / signals → more Narrow focus / attentional narrowing → less Mental-model updating →

poorer Adaptive performance / postchange performance → more Threat / pressure

The other loops follow the same reinforcing logic:

- more Mental-model updating failure / persistence → more Routine persistence → more Cognitive entrenchment → more NIH / boundary defense / external rejection → less Cross-boundary information flow → less Mental-model updating
- more Stable composition / bounded domains → more Social identity / in-group salience → more NIH / boundary defense / external rejection → less Cross-boundary information flow → more Social identity / in-group salience
- more Goal pressure / narrow targets → more Tunnel attention / local optimization → more Narrow focus / attentional narrowing → less Mental-model updating → poorer Adaptive performance / postchange performance → more Threat / pressure → more Goal pressure / narrow targets

That is why rigidity can grow gradually, feel locally rational, and still become systemically harmful. A sociotechnical reading points in the same direction: the design problem is never only technical. Organizations are held together by patterns of communication and decision-making, and those patterns are themselves part of what gets reproduced over time. For our purposes the implication is simple but useful: the way people coordinate is part of the system producing rigidity, not just a wrapper around it.

## What actually helps

If this pattern is real, then a few design moves matter more than others. None of them remove structure or expertise. What they do is make it harder for useful structure and expertise to harden into blindness.

In practice, six moves matter most:

- Reopen the design regularly, so boundaries stay provisional instead of turning into defended identities.
- Keep information moving across boundaries, so teams still see disconfirming signals before local certainty hardens.
- Broaden what teams own, so local optimization has a harder time crowding out whole-product judgment.
- Interrupt routine persistence, so expertise stays useful without becoming the only frame available.
- Lead against closure under pressure, so threat does not automatically collapse the system into tighter control and narrower sensing.
- Design goals for learning, so targets guide attention without turning into tunnel vision.

### Reopen the design regularly

The first and most important move is to reopen the design before rigidity fully sets in. Design should not be treated as a one-off intervention followed by years of interpretation. It should be treated as an

ongoing move: map, assess, debate, redesign, learn, and repeat.

One practical implication is that boundaries stay provisional rather than becoming part of the unquestioned structure. Teams need clear accountabilities, but they also need regular opportunities to reconsider whether the current boundaries still fit the work. Without that, bounded domains become part of group identity and then become harder to reconfigure.

Boundary review should not be exceptional. It should be built into the operating model through recurring design conversations, scheduled boundary reviews, architectural retrospectives, value-stream reviews, cross-team experiments, or similar moves. Without this meta-practice, even good structures drift. With it, many of the other design moves become easier to sustain.

Main pressures it relieves:

- Stable composition / bounded domains
- Social identity / in-group salience
- NIH / boundary defense / external rejection

## **Keep information moving across boundaries**

Adaptation depends on what the system is still capable of noticing.

The most reliable way to weaken insensitivity is to keep information moving across boundaries. Cross-team reviews, shared planning, refinement, experiments, and other recurring interactions widen the informational field. They keep weak ties alive and increase the chance that disconfirming evidence gets seen before it is ignored for too long. One of the simplest practical rules is also one of the most important: do not let teams operate only inside their own local frame for too long.

What it strengthens:

- Cross-boundary information flow
- Attention to outside cues / signals
- indirectly, Mental-model updating

## **Broaden what teams own**

Broader ownership changes what teams pay attention to.

When responsibility is organized around a broader product slice rather than a narrow local mandate, teams are less able to optimize their own area while ignoring the larger system. Whole-product ownership does not reduce complexity. It redistributes where that complexity has to be handled. That is exactly why it matters: it weakens some of the conditions that make local optimization and defensive boundary behavior feel normal.

Main pressures it relieves:

- Stable composition / bounded domains
- Tunnel attention / local optimization

- NIH / boundary defense / external rejection

What it strengthens:

- Cross-boundary information flow

Whole-product ownership also changes the shape of the loop itself:

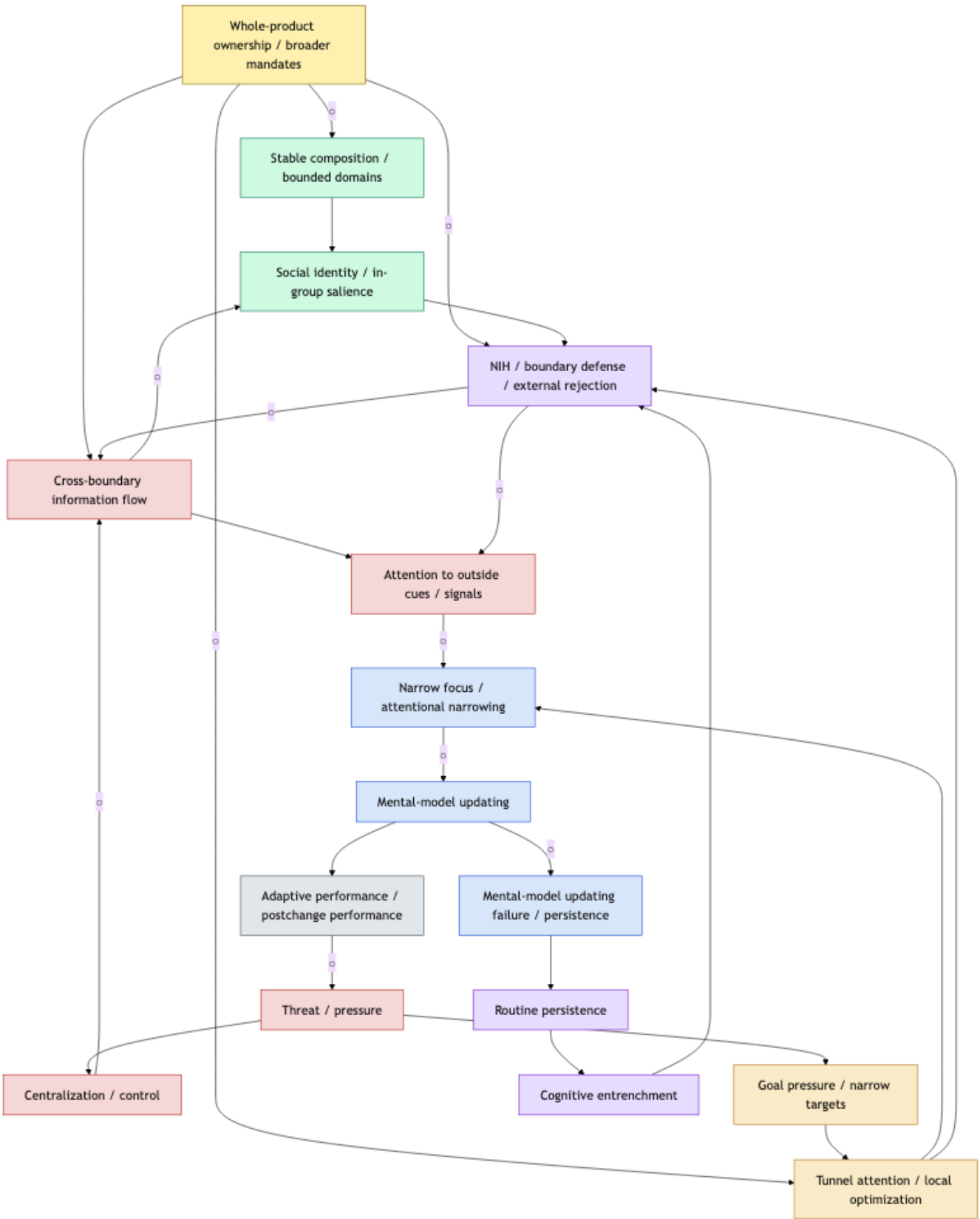


Figure 3. Whole-product ownership weakens bounded domains, local optimization, and boundary defense.

### Interrupt routine persistence

Entrenchment grows when the same people solve the same kinds of problems in the same way for too long.

For that reason, rotation, shared responsibility, exposure to varied problems, and fresh evaluation criteria matter. These moves do not eliminate expertise. They stop expertise from becoming the only frame available. Their role is to introduce enough novelty that existing routines and schemas have to be re-examined.

Main pressures it relieves:

- Routine persistence
- Cognitive entrenchment

## **Lead against closure under pressure**

Leadership matters here, though not because leaders directly update other people's mental models.

It matters because leadership changes the climate around information, dissent, and control. Under pressure, organizations often centralize and close down. That is exactly the terrain described by threat-rigidity research. Leaders who intensify command-and-control behavior strengthen that drift. Leaders who invite challenge, outside input, and debate weaken it.

Research also gives this point some practical shape. [Bonini et al. \(2024\)](#) find a positive overall relationship between leadership and adaptive performance. [Ma et al. \(2024\)](#) find that empathetic leadership supports innovative behavior partly through career adaptability. [Leone and Reiter-Palmon \(2022\)](#) add a concrete move on the evaluation side: especially when leaders are not domain experts, they can work with teams to define evaluation criteria and reduce bias toward tried-and-true solutions, so established patterns have to earn their place rather than be assumed.

Main pressures it relieves:

- Centralization / control
- NIH / boundary defense / external rejection

What it strengthens:

- Attention to outside cues / signals

Leadership also changes the loop by changing the information climate:

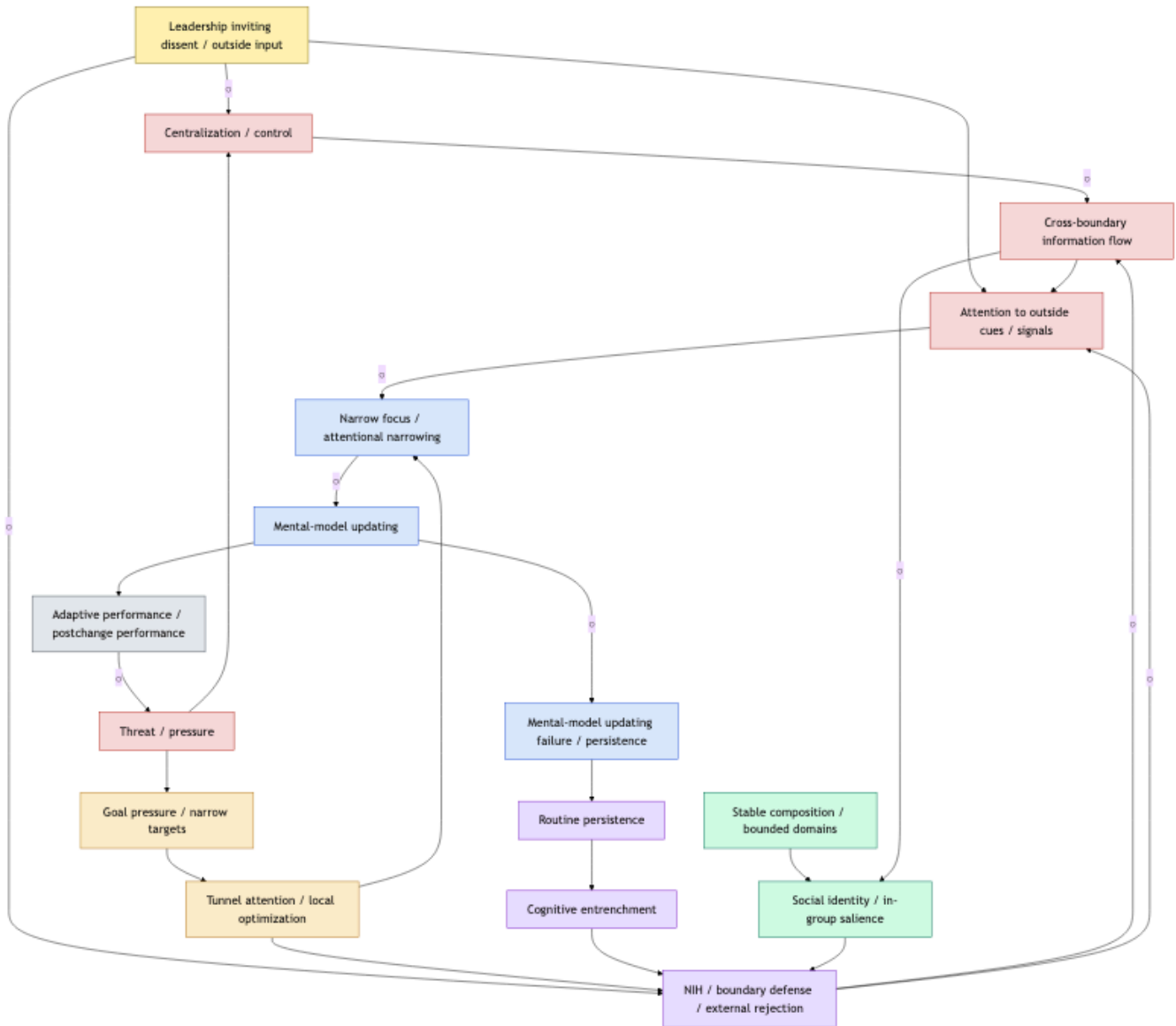


Figure 4. Leadership opens information flow and weakens control-driven closure.

## Design goals for learning

Goals are not neutral. They shape attention.

Research on goals shows that narrow targets can produce tunnel attention and non-goal neglect ([Ordonez et al., 2009](#)). In interdependent work, that means teams can end up protecting a narrow target even when the broader system is changing around them.

The practical implication is not to abandon goals, but to design them differently: more whole-product oriented, more provisional, and more balanced with learning signals and counter-metrics.

Main pressures it relieves:

- Goal pressure / narrow targets
- Tunnel attention / local optimization

# What this lens says about common frameworks

This is not a definitive ranking. It is what these approaches look like when you care specifically about avoiding long-term rigidity.

The question is not which framework is universally best. The more useful question is what each one tends to strengthen, what it under-supports, and what it leaves to good judgment when you look at it through this lens.

## Team Topologies

Through this lens, [Team Topologies \(Skelton & Pais, 2019\)](#) under-supports several of the design moves that keep rigidity from hardening over time.

Its language is stabilizing, which helps explain its popularity. It gives organizations a clean way to think about boundaries, interaction modes, and flow. Cognitive load and architecture alignment do matter. The problem is that TT leans on them so heavily that other parts of adaptability get pushed to the edge: broader ownership, recurring redesign, strong cross-boundary sensing, and deliberate moves that interrupt local hardening before it becomes normal.

The design logic matters here. TT strongly encourages aligning team boundaries with software and domain boundaries, and it makes cognitive-load reduction a major reason for doing so. That is attractive because it makes local clarity easier to achieve. It also fits an inverse-Conway style instinct: shape teams around the architecture you want, and shape the architecture around teams that can own clearer slices, which still traces back to Conway's original argument about the relationship between system design and communication structure ([Conway, 1968](#)). But once those moves become dominant, they can also become narrowing mechanisms. They can harden local ownership, reduce the felt need for whole-product exposure, and turn architecture-fit into a reason not to revisit whether the boundaries themselves should change.

That leads to an important nuance. TT may indeed create some room for adaptation by reducing cognitive load and improving local coordination. Under moderate change, that can help teams notice cues and respond. But that benefit is fragile. As routines stabilize, teams can become entrenched in the very interaction patterns that made them efficient, and under pressure organizations often narrow attention further and fall back on familiar responses rather than reopen the design ([Uitdewilligen et al., 2013](#); [Staw et al., 1981](#)). At that point, TT's redesign vocabulary is also relatively narrow: it can support reshuffling within its own grammar of team types, interaction modes, and boundary choices, but it is less helpful when the right move is to question that grammar itself.

On this measure, TT under-supports these design moves:

- **Broaden what teams own** It reinforces narrower ownership around bounded domains more than whole-product ownership.
- **Keep information moving across boundaries** It allows interaction modes, but it does not build strong recurring forcing functions that keep information and challenge moving across boundaries.

- **Reopen the design regularly** It makes bounded domains and team identity highly salient, but leaves most of the forcing functions for revisiting boundaries and redesigning the structure underspecified.
- **Interrupt routine persistence** Its emphasis on cognitive-load reduction through narrower focus can unintentionally strengthen the conditions under which expertise calcifies.

There is an obvious counterargument. TT does not literally say teams should never evolve. It discusses team interactions, platform support, and the need to revisit designs as contexts change. Skilled practitioners can certainly use TT more lightly than the rhetoric of its core patterns may suggest. The issue is that these adaptive elements remain weaker than the book's main structural signal. The strongest forcing functions in TT still point toward clearer boundaries, cleaner ownership, and reduced load. The forcing functions for reopening the design, broadening ownership, and deliberately increasing cross-boundary challenge are much less concrete.

That is why TT still worries me through this lens. It may help local execution, but it gives too little structural support to the design moves that weaken rigidity over time. Practitioner critiques reinforce the same concern. [Aas \(2025\)](#) argues that TT can over-index on mirroring the current architecture rather than questioning whether the architecture itself should change, which aligns with the inverse Conway maneuver discussed by [LeRoy and Simons \(2011\)](#). [Oost \(2024\)](#), [Krivitsky \(2022\)](#), and [Drobniuch \(2024\)](#) likewise question whether TT's treatment of boundaries and cognitive load is too easily turned into a rigid template. I do not think those critiques carry the whole argument on their own, but they do point in the same direction.

## LeSS

By contrast, [LeSS \(Larman & Vodde, 2016\)](#) supports more of the anti-rigidity design moves directly.

That contrast with TT matters. LeSS also relies on stable, semi-permanent teams, so the difference is not simply permanence versus fluidity. The difference is what the stability is built around. In TT, stability is tied more strongly to clearer bounded ownership and a smaller interaction grammar across teams. In LeSS, stability is tied more strongly to a shared product, one backlog, and recurring joint sensemaking. Because teams remain oriented to a broader common goal rather than a narrower slice, they have more room to shift work, re-coordinate, and adjust interaction patterns when task demands change.

- **Broaden what teams own** Whole-product focus and one backlog reduce the ease of narrow local optimization.
- **Keep information moving across boundaries** Shared reviews, planning, and refinement keep weak ties and disconfirming signals alive.
- **Interrupt routine persistence** Shared responsibility and varied work reduce the chance that expertise calcifies in one narrow area.
- **Reopen the design regularly** Principles, rules, guides, and experiments keep adaptation alive as an ongoing process rather than a one-off setup.

- **Design goals for learning** Whole-product orientation makes it easier to frame success in broader system terms rather than only local targets.

That does not make LeSS automatically adaptive. It can still become ritualized if shared events stop functioning as real sensing, challenge, and redesign mechanisms and become mere ceremony. But through this lens it looks stronger because it embeds recurring mechanisms for information flow, challenge, and redesign rather than treating them as optional extras.

LeSS can also harden in a more specific way. One backlog, shared cadence, and stable teams do not guarantee fresh thinking on their own. If the product group starts treating its current framing as settled, if refinement becomes convergence too early, or if shared forums become places where teams align around the dominant interpretation rather than test it, LeSS can still drift into collective closure. The mechanism is different from TT's narrower boundary logic, but the risk remains: strong shared context can become shared unquestioned context.

## Org Topologies

[Org Topologies \(Krivitsky & Flemm, 2021-\)](#) is useful in a different way. It is not really a framework in the same sense as LeSS or Team Topologies. It works better as a design tool or design language for making structural trade-offs visible. That makes it especially useful for the meta-move of reopening the design regularly.

It is also useful because it makes the trade-offs between its three topologies easier to discuss explicitly. OT distinguishes between a Resource Topology, a Delivery Topology, and an Adaptive Topology. Through this lens, those are not just categories to sort organizations into. They are different structural bets with different strengths and failure modes. A Resource Topology can improve utilization but fragment ownership. A Delivery Topology can improve the flow of outputs but still separate directing from doing. An Adaptive Topology tries to merge directing, doing, and delivering into a broader unit oriented toward outcomes, learning, and easier change. That makes OT a good companion for this essay's systems view: it gives you a way to evaluate those trade-offs in terms of what they open or close in the larger rigidity pattern.

Its value shows up most clearly here:

- **Reopen the design regularly** The MADE cycle explicitly reopens design through mapping, assessment, redesign, and elevation, including review of mandate scope and boundaries.
- **Broaden what teams own** By discussing scope of work mandate and scope of skills mandate, it helps leaders see when current boundaries are too narrow for the work.
- **Keep information moving across boundaries** Not by prescribing one fixed event set, but by making misalignment between structure and real coordination needs visible and then introducing recurring practices where they are needed.

That is where OT's idea of elevating katas becomes especially useful. They are the bridge from diagnosis to action: recurring practices that start operationalizing some of the levers above. In practice, they can look a lot like shared reviews, planning, refinement, and other cross-boundary forums that help

teams shift from narrower output ownership toward broader outcome responsibility. In that sense, OT can complement LeSS here: LeSS builds several of those forums into the operating model by default, while OT gives you a way to introduce and adapt them more intentionally when the diagnosis shows they are missing.

Used well, OT supports the habit of reflection that many organizations otherwise lack. Its weakness is different from TT's: it can become a language for mapping the current state without creating enough pressure to change it. Used poorly, it can become a language for defending the status quo rather than reopening design. But through this lens its main value is clear. It helps people revisit the current structure before it hardens. In that respect it also fits broader systemic design work that treats organizational design as an ongoing leadership responsibility rather than a one-time structure choice ([Ramos & Pavlichenko, 2022](#)).

That is also where OT can fail more concretely. A team can map mandates, discuss topology fit, and surface structural trade-offs accurately, yet still stop short of redesign if no one has the authority, appetite, or timing to act on what the diagnosis reveals. In that case OT improves the quality of the conversation without changing the structure that keeps reproducing the problem. It becomes a language of diagnosis rather than a forcing function for redesign.

## What to watch for

The point of this lens is not to settle a framework debate once and for all. It is to keep a familiar failure mode visible.

Organizations do not usually become rigid because of one bad choice. They become rigid because several locally sensible choices start reinforcing each other. That is why rigidity is so easy to miss while it is forming, and why it can look like good management right up until the system as a whole stops adapting.

That is also the trap with this lens. If you turn it into a new doctrine, you repeat the same mistake. The value is in the questions it helps you ask. Are boundaries still serving the work? Is information still moving? Are teams optimizing locally? Are routines still alive? Are goals narrowing attention? Are leaders opening the system or closing it?

The real test is simple: does the organization have recurring ways to stop useful patterns from hardening into unquestioned rules?

## References

- Aas, P. (2025). *The Fundamental Misunderstanding in Team Topologies*. [Blog post]. [https://patricia.no/2025/05/24/team\\_topologies.html](https://patricia.no/2025/05/24/team_topologies.html)
- Bonini, S., et al. (2024). *Leadership and adaptive performance: A systematic review and meta-analysis*. PLOS One. <https://doi.org/10.1371/journal.pone.0304720>

- Conway, M. E. (1968). *How do committees invent?* *Datamation*, 14(4), 28–31.  
[https://www.melconway.com/Home/Conways\\_Law.html](https://www.melconway.com/Home/Conways_Law.html)
- Dane, E. (2010). *Reconsidering the trade-off between expertise and flexibility: A cognitive entrenchment perspective.* *Academy of Management Review*, 35(4), 579–603.  
<https://doi.org/10.5465/amr.35.4.zok579>
- De Smet, J., & Grcic, V. (2023). *Navigating Complexity of Cognitive Load in Software Product Development.* Leanpub.  
<https://leanpub.com/cognitiveload>
- Drobniuch, B. (2024). *From Conway's Law to Cognitive Load: Rethinking Organizational Agility (Talk LeSS 2024).* [Video].  
[https://www.youtube.com/watch?v=rug7\\_fHYpsE](https://www.youtube.com/watch?v=rug7_fHYpsE)
- Gersick, C. J. G., & Hackman, J. R. (1990). *Habitual routines in task-performing groups.* *Organizational Behavior and Human Decision Processes*, 47(1), 65–97.  
[https://doi.org/10.1016/0749-5978\(90\)90047-D](https://doi.org/10.1016/0749-5978(90)90047-D)
- Hayes, S. C. (2019). *A Liberated Mind: How to Pivot Toward What Matters.* Avery.  
<https://www.goodreads.com/book/show/43263358-a-liberated-mind>
- Kahneman, D. (2011). *Thinking, Fast and Slow.* Farrar, Straus and Giroux.  
<https://us.macmillan.com/books/9780374533557/thinkingfastandslow/>
- Katz, R., & Allen, T. J. (1982). *Investigating the "Not Invented Here" (NIH) syndrome: Performance, tenure, and communication patterns of 50 R&D project groups.* *R&D Management*, 12(1), 7–20.  
<https://doi.org/10.1111/j.1467-9310.1982.tb00478.x>
- Krivitsky, A. (2022). *How Adaptive Are Team Topologies?* Org Topologies blog.  
<https://www.orgtopologies.com/post/how-adaptive-are-team-topologies>
- Krivitsky, A., & Flemm, R. (2021–). *Org Topologies™.*  
<https://www.orgtopologies.com>
- Larman, C., & Vodde, B. (2016). *Large-Scale Scrum: More with LeSS.* Addison-Wesley.  
<https://less.works/less/framework/index.html>
- Leone, S. A., & Reiter-Palmon, R. (2022). *Leading creative teams: A process-perspective with implications for organizational leaders.* *Translational Issues in Psychological Science*, 8(1), 90–103.  
<https://doi.org/10.1037/tps0000296>
- Ma, H., et al. (2024). *Empathetic leadership and employees' innovative behavior: examining the roles of career adaptability and uncertainty avoidance.* *Frontiers in Psychology*.  
<https://doi.org/10.3389/fpsyg.2024.1371936>

- Marquet, D. L. (2020). *Leadership is Language: The Hidden Power of What You Say and What You Don't*. Penguin.  
<https://www.goodreads.com/book/show/42774083-leadership-is-language>
- Meadows, D. H. (2008). *Thinking in Systems: A Primer*. Chelsea Green Publishing.  
<https://www.chelseagreen.com/product/thinking-in-systems/>
- Mohammed, S., Ferzandi, L., & Hamilton, K. (2010). *Metaphor No More: A 15-Year Review of the Team Mental Model Construct*. *Journal of Management*, 36(4), 876–910.  
<https://doi.org/10.1177/0149206309356804>
- Oost, M. (2024). *Stop Team Topologies*. Medium.  
<https://martyoo.medium.com/stop-team-topologies-fd954ea26eca>
- Ramos, C., & Pavlichenko, I. (2022). *Creating Agile Organizations: A Systemic Approach to Develop People and Teams for Superior Performance*. Addison-Wesley.  
<https://www.informit.com/store/creating-agile-organizations-a-systemic-approach-to-9780136523263>
- Skelton, M., & Pais, M. (2019). *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution.  
<https://teampologies.com/book>
- Senge, P. M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. Currency Doubleday.  
<https://www.penguinrandomhouse.com/books/163984/the-fifth-discipline-by-peter-m-senge/>
- Staw, B. M., Sandelands, L. E., & Dutton, J. E. (1981). *Threat Rigidity Effects in Organizational Behavior: A Multilevel Analysis*. *Administrative Science Quarterly*, 26(4), 501–524.  
<https://doi.org/10.2307/2392338>
- Tajfel, H., & Turner, J. C. (1979). *An integrative theory of intergroup conflict*. In W. G. Austin & S. Worchel (Eds.), *The social psychology of intergroup relations* (pp. 33–47). Brooks/Cole.
- Uitdewilligen, S., Waller, M. J., & Pitariu, A. H. (2013). *Mental model updating and team adaptation*. *Small Group Research*, 44(2), 127–158.  
<https://doi.org/10.1177/1046496413478205>
- LeRoy, J., & Simons, M. (2011, February 3). *Dealing with creaky legacy platforms*. JonnyLeRoy.com. Originally published in the December 2010 issue of *Cutter IT Journal*.  
<https://jonnyleroy.com/2011/02/03/dealing-with-creaky-legacy-platforms/>
- Ordonez, L. D., Schweitzer, M. E., Galinsky, A. D., & Bazerman, M. H. (2009). *Goals Gone Wild: The systematic side effects of overprescribing goal setting*. *Academy of Management Perspectives*, 23(1), 6–16.  
<https://doi.org/10.5465/amp.2009.37007999>