# Pressure Stabilisation System

Facilitator guide

**Your job:** help the team see where pressure is spreading to other teams — then put a circuit-breaker on the worst handoffs, so it stops cascading. You're containing the spread, not making anyone tougher.

**The one idea:** pressure becomes dangerous when it spreads — the damage is in the spread, not the load. An incident hits one team; under pressure they skip the handoff, and by the next day three teams are scrambling. The cascade came from the handoff, not the incident. Contain the spread at the worst handoff and the whole system steadies.

WHEN DO I USE THIS?

## Use this when

**GOOD FIT**

Pressure on one team keeps becoming other teams' problem. An incident or crunch sets off reacting far beyond its source, and rushed handoffs are creating errors downstream. You can name where one team's pressure is becoming another team's problem.

**WRONG TOOL**

One hard call to make under pressure › Decide Under Pressure. One person needs to stay composed when something lands hard › Steady Receive. Recovering after the setback has already happened › Recovery Playbook. The load genuinely sits inside one team › Execution Pacing System.

**The one test before you start:** can you name where one team's pressure is becoming another team's problem? If the pressure sits inside a single team, this isn't the tool yet — that's a load problem, not a spread one.

WHAT WILL THE TEAM LEAVE WITH?

## What you'll leave with

**WHAT YOU GET**

A map of where pressure spreads — not just where it starts — with the two or three worst handoffs marked, a circuit-breaker placed on each, and the first one owned and dated for this week.

**WHY IT MATTERS**

The team sees the source wasn't where most of the damage was — and that containing the spread calmed the system faster than reducing the load. The pressure stops becoming everyone else's problem, without anyone having to work harder.

© Witmer

## PART 1 BEFORE THE ACTIVITY

get ready

1 · WHAT IS MY JOB HERE?

### Your job

You are running this to one outcome: a circuit-breaker on each of the worst handoffs, with the first one owned and dated. Your job is to:

- keep the focus on the **handoff, not the people** — the pile-up is a broken handoff, never a weak team
- make the cascade **physical** — pass the tokens and let the pile grow, so the spread is seen, not argued
- let the team **find the worst spread-point themselves** — don't name it for them; the surprise is the reveal
- **contain before reducing** — circuit-breakers first; the original load is a separate conversation

THE ONE RULE

**Stop pressure spreading before you try to reduce it.** The whole reveal depends on the tokens being passed and piling up where the team can see them. Reduce the load instead, and the spread simply finds the next handoff. Contain it at the handoff first.

2 · WHO DOES WHAT?

### Roles

| Role | What they do |
|---|---|
| You (facilitator) | Run the flow, pass the tokens, keep the focus on the handoff not the people, and capture the circuit-breakers and owners. Make the cascade physical; let the team find the spread-point. |
| Participants (4–10) | The people from across the handoffs — not just the team at the source. They map the dependencies, watch the cascade, and place the circuit-breakers. 45–60 minutes. |

3 · HOW DO I OPEN?

### Opening script — 2–3 minutes

In your own words, never read aloud. It sets the frame without giving away the reveal.

SAY — IN YOUR OWN WORDS

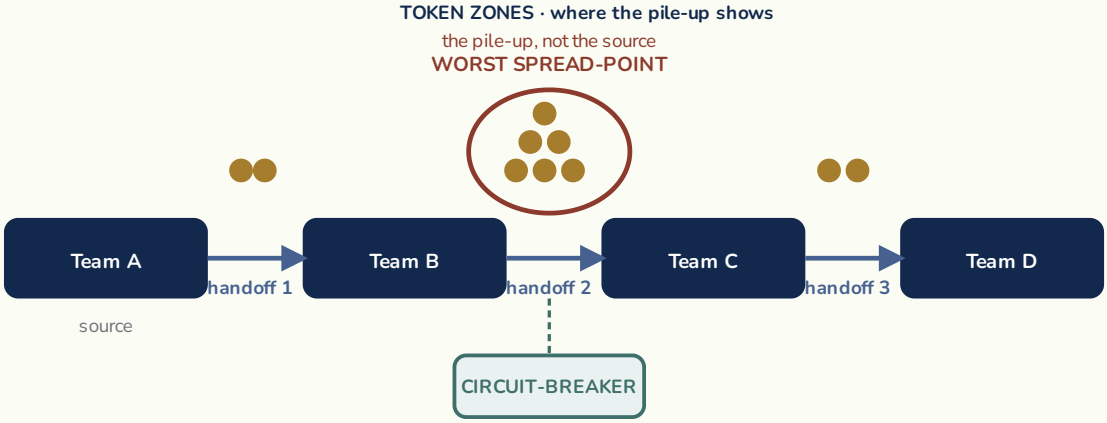
"We're not here to reduce anyone's workload or hand out blame. We're going to take one real pressure situation and follow how it travels between our teams — and I think we'll find the damage isn't where the pressure started. It's in the handoffs, where one team's pressure gets passed to the next. Then we'll put something small and concrete on the worst ones to stop that."

© Witmer

4 · HOW DO I PREPARE THE SPACE?

## Setup and materials

THE DEPENDENCY BOARD — TOKENS PILE UP AT THE HANDOFF, NOT THE SOURCE




**One or two tokens start at the source; each handoff adds tokens for the stress it creates downstream.**

| Component | What it is, and when you use it |
|---|---|
| **Dependency board** | The team's work as a chain of who hands what to whom — the path pressure travels, with a token zone above each handoff. Step 2. |
| **Pressure tokens** | Passed along the chain; added at each handoff for the stress it creates downstream. They make the cascade visible. Steps 3–4. |
| **Circuit-breaker cards** | Ways to contain spread at a handoff — a protected channel, a comms rule, a decision boundary. Step 5. |
| **Starter-scenario cards** | Optional. Example pressure situations — use one only if the group has no real situation of its own. |

**Get the right people in the room.** Bring people from across the handoffs, not just the team at the source. The spread only shows up if the teams it spreads to are present.

## PART 2 DURING THE ACTIVITY

run it

**Before you begin — 30-second check**

- ✓ There's one real pressure situation that's spreading across teams — live or recurring
- ✓ The people from across the handoffs are in the room, not just the source team
- ✓ The board is ready to map the chain, with token zones above each handoff
- ✓ You're ready to hold the rule — contain the spread before reducing the load

© Witmer

5 · HOW DO I RUN IT?

## How to run it

Brief the team (2–3 min), then run these in order. The bold part is what you say or do; the note under it is where you earn your keep.

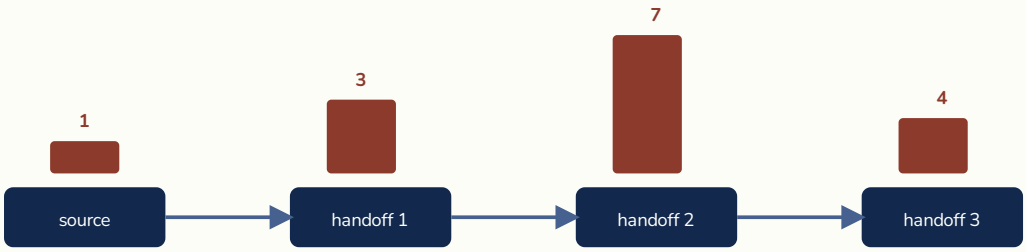
| | | |
|---|---|---|
| **1. Set up** | **5 min** | *"What's the one real pressure situation we're working on?"* Appoint yourself facilitator. Name a specific incident, crunch or slip — one where one team's pressure became another's problem. |
| **2. Map the flow** | **9–11 min** | *"Who hands what to whom — lay it out as a chain."* Only the path the pressure actually travels; don't map the whole org. Map what actually happened, not the process everyone wishes they followed. |
| **3. Pass the pressure** | **9–11 min** | *"Drop tokens at the source, then pass them along."* Add one token each time a handoff forces another team into extra or reactive work. Make it physical and visible — the pile-up is the point. |
| **4. Find the worst spread-points** | **9–11 min** | *"Where did the tokens pile up most?"* Read the piles by relative size — the biggest pile is the worst spread-point; you don't need exact counts. Ignore the original source; focus only on where pressure multiplied most. Mark the two or three worst handoffs, and let the team notice it's not the source — don't say it for them. |
| **5. Break the spread** | **11–13 min** | *"What single rule here would stop this pressure being passed to the next team?"* Place a circuit-breaker on each marked handoff — a channel, a rule, a boundary. Contain the spread; don't start fixing the original load. |
| **6. Commit** | **5–6 min** | *"Which one goes in this week, who owns it — and how will we know it held?"* Name each circuit-breaker with an owner and when it goes in. For the first one, name the signal that will tell you it worked next time pressure hits — e.g. that handoff stops generating downstream firefights. Photograph the board before anyone leaves. |

**These times are indicative** and will depend on the nature of the problem and the size of the team; the whole activity runs 45–60 minutes. The step that overruns is pass the pressure (step 3) — it gets lively. The step that gets squeezed is break the spread (step 5); protect it. A vivid cascade with no circuit-breaker changes nothing.

© Witmer

**What a circuit-breaker looks like — make it concrete.** Not "communicate better", but a specific rule at a specific handoff. For a protected channel: *during the incident window, the B›C handoff goes through one named coordinator — not five ad-hoc pings.* For a comms rule: *no cross-team requests during a Sev-1 except through the incident lead.* For a decision boundary: *Team C decides what it drops; upstream doesn't get to reassign its work.* If it couldn't be written on a sticky note and enforced next week, it's too vague.

THE CASCADE — PRESSURE MULTIPLIES AS IT CROSSES HANDOFFS




*The biggest pile is at handoff 2 — that broken handoff, not the incident, is exporting pressure worst.*

**When to step in — and when to stay out** · a principle, not a step

**STEP IN**

The room blames a team instead of looking at the handoff → redirect to the handoff.

They map the whole org, not the pressure path → pull it back to the path.

They jump to reducing the load before containing the spread → hold the rule.

They want to fix every handoff → push them to the worst two or three.

**STAY OUT**

The team is surprised the source isn't the worst point → let it land.

They debate which handoff is worst → that's the work.

A circuit-breaker feels small → small and real beats big and vague.

The cascade gets lively → that energy is the tool working.

**The clearest tell it's working:** someone looks at the token pile-up and says "the incident isn't the problem any more — it's that handoff." That's the reveal arriving. If all the tokens sit at the source, the chain wasn't followed far enough — keep passing.

© Witmer

## PART 3 AFTER THE ACTIVITY

make it stick

6 · HOW DO I CLOSE?

### Closing script — 2 minutes

Reconstruct the discovery — don't summarise it. Replay where the team started so the insight lands as something they lived.

> SAY — IN YOUR OWN WORDS
>
> "When we started, we'd have said the problem was *[the incident]*. But look at the board — one token at the source, and a pile here, at *[the worst handoff]*. That's where the pressure was being exported worst. We've put one circuit-breaker there. The incident is still live — but the system steadies, because the spread is contained."

7 · DID WE GET A GOOD RESULT?

### What good looks like

**A GOOD RUN**

Tokens pile up away from the source; the team names the worst handoff itself; two or three concrete circuit-breakers, with owners. The team can say "the damage was in the spread, not the load."

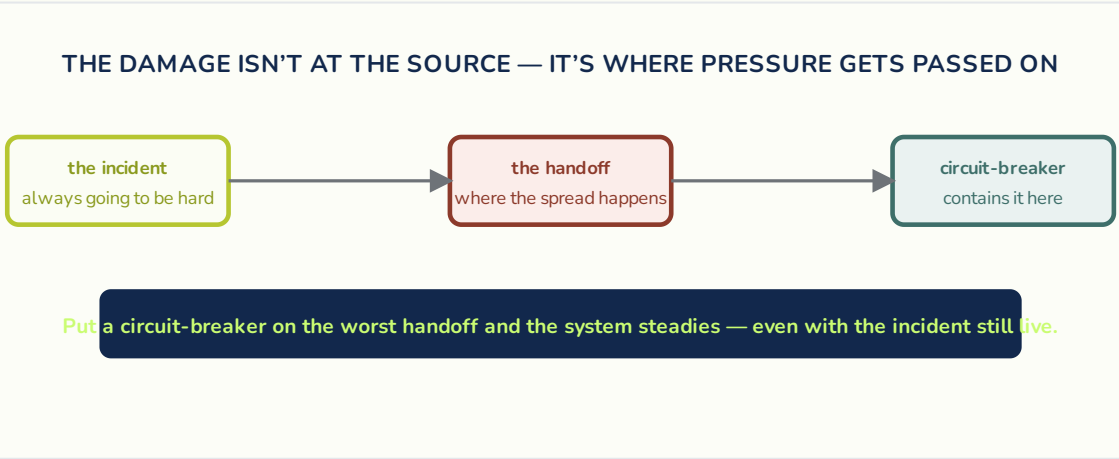
**A HOLLOW RUN**

Tokens stay at the source (the chain wasn't followed); the room blames a team; "everyone just needs to stay calm" with no circuit-breaker. If it looks like this, keep passing the tokens and redirect to the handoff.

8 · WHY DOES THIS WORK?

### The idea underneath

Pressure itself isn't what overwhelms a delivery. The real damage happens when pressure spreads from one team to the next through rushed handoffs and reactive decisions — the cascade does more harm than the source. Containing the spread at a few key handoffs steadies the whole system, faster than trying to reduce the pressure itself, because the pile-up — not the incident — is what's destabilising the teams around it.




© Witmer

**Before you finish — the check**

- ✓ The chain was followed far enough that tokens piled up away from the source
- ✓ The team named the two or three worst spread-points itself
- ✓ A concrete circuit-breaker is placed on each — a channel, a rule, a boundary
- ✓ The first circuit-breaker has a named owner and a date for this week
- ✓ The original load was left as a separate conversation — not solved here
- ✓ The first circuit-breaker has a signal that will tell you it held next time pressure hits
- ✓ The board has been photographed before anyone leaves

9 · HOW DO I MAKE THIS A HABIT?

## Take it back to work

**THE HABIT**

Under pressure, ask: “where is this becoming someone else’s problem?” — and put a circuit-breaker there.

The lasting change: the team stops treating pressure as something to push through and starts watching where it gets passed on — containing the spread at the handoff before it cascades. Pressure spreads faster than work. Stop the spread first.

© Witmer

## Appendix · Reference

You should not need this while you run the activity. Read it before, or use it if a run goes sideways.

10 · A WORKED RUN

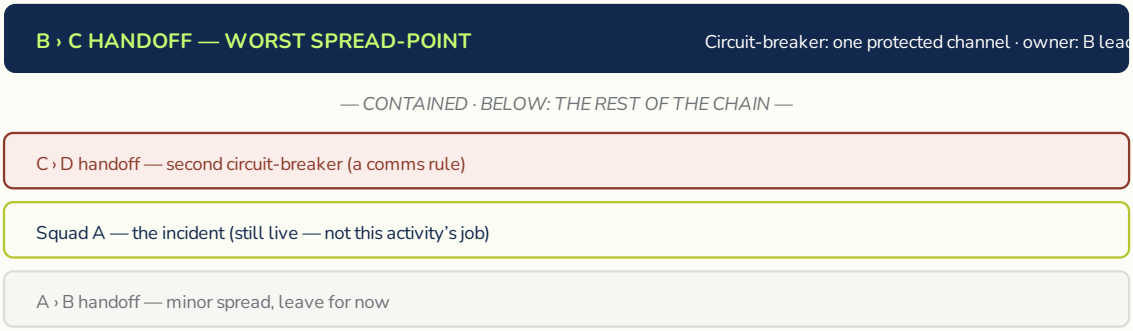
# Watch the reveal land

A production incident lands on Squad A mid-release. Under pressure they skip their handoff notes and pull in Teams B, C and D as they go. The team maps the chain and passes the pressure.

### THE MOMENT THE REVEAL LANDS

One token starts at Squad A. By the B›C handoff the pile has multiplied — far more than sits at the source. Someone says: **“the incident isn’t the problem any more — it’s that handoff.”** The incident was always going to be hard, but most of the cascade came from one rushed handoff.

### FINISHED SPREAD MAP — CIRCUIT-BREAKERS ON THE WORST TWO HANDOFFS




*Two circuit-breakers steady the whole chain — the incident is a separate conversation.*

**How the facilitator made it land:** made the cascade physical (tokens passed and piling up, not a discussion about pressure); let the team find the spread-point rather than naming it; held the focus on the handoff, not the people; and contained before reducing — a single protected channel on the B›C handoff, with the original incident left as a separate conversation. The system steadied even though the incident was still live.

**The question that links it to real work:** before they leave, ask — “which of our recent incidents spread exactly like this, and where did it cross over?” Naming a real one turns the model into a pattern the team will recognise next time.

© Witmer

11 · WHAT GOES WRONG MOST OFTEN?

## Common mistakes

| Mistake | Why it fails the run |
|---|---|
| **Mapping the whole org** | Map only the path the pressure travels, or the cascade gets lost in detail. The chain is the tool. |
| **Blaming the team, not the handoff** | The pile-up is about a broken handoff, not a weak team. Keep it on the handoff. |
| **Reducing the load first** | Slower and often impossible mid-crunch. Contain the spread first; the load is a separate conversation. |
| **Fixing every handoff** | Two or three circuit-breakers steady the system; ten is a project no one finishes. |
| **Solving the original incident here** | That's not this activity's job. Solve the spread; leave the incident to the people already on it. |

12 · WHAT IF SOMETHING GOES WRONG?

## Edge cases

| What happens | What to do |
|---|---|
| All tokens stay at the source | The chain wasn't followed far enough. Keep passing — add tokens for every downstream effect until the spread shows. |
| The room blames a team | Redirect to the handoff: "what about the handoff let this through?" Never "who dropped it?" |
| Pressure really is all at the source | Rare, but legitimate — then this is a load problem, not a spread one. Note it and use a pacing or capacity tool. |
| Everything looks like a spread-point | Rank by pile-up; take only the worst two or three. The rest can wait. |
| A circuit-breaker needs authority the team lacks | Capture it as an ask with an owner — naming it is the first containment. |
| Time runs out before circuit-breakers | Cut a token-passing round, never the break step. Commit one circuit-breaker now. |

© Witmer

13 · WHERE DOES THIS FIT?

## Where this sits in the Gym

| If instead... | Use this |
|---|---|
| there's one sharp call to make inside the pressure | **Decide Under Pressure** — make the call well under load. |
| one person needs to stay composed when something lands hard | **Steady Receive** — hold steady in the moment. |
| the spread is contained and you want to build recovery in for next time | **Recovery Playbook** — a reusable recovery for next time. |
| the load genuinely sits inside one team | **Execution Pacing System** — set a sustainable pace (not a spread problem). |

**Rule of thumb:** Pressure Stabilisation System is about containing spread between teams — stop the cascade at the handoff before reducing anything. It's not a single hard call (Decide Under Pressure), not one person staying composed (Steady Receive), and not a single-team load problem (Execution Pacing System).

© Witmer