Backstage

Keeping Backstage plugins current

with official codemods

Automate known edits. Keep the judgment.

Paul Schultz · Red Hat
Alex Bit · Codemod



Migrations are a people problem.

Confidence

What will change?

→

Trust

Can we review it?

→

Coordination

Who owns the rollout?

A shared rule helps. People still own the rollout.

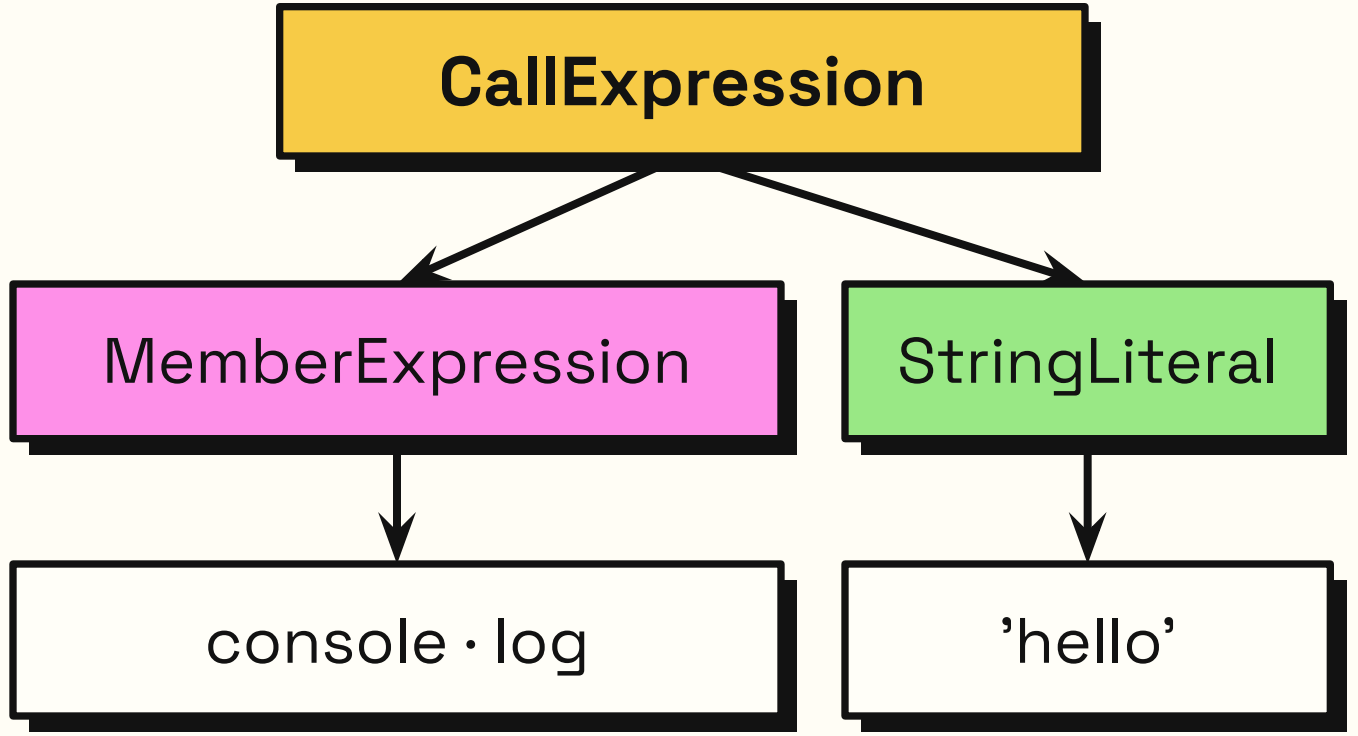
Capture the change once.



Keep expertise in code—not in every maintainer's memory.

Match structure, not strings.

```
console.log('hello');
```



An import, a call, and a string are different nodes.

Change the call.
Leave the message.

- Before

```
- console.log(
    'console.log'
  );
```

+ After

```
+ logger.info(
    'console.log'
  );
```

The matched call changes. The string stays intact.

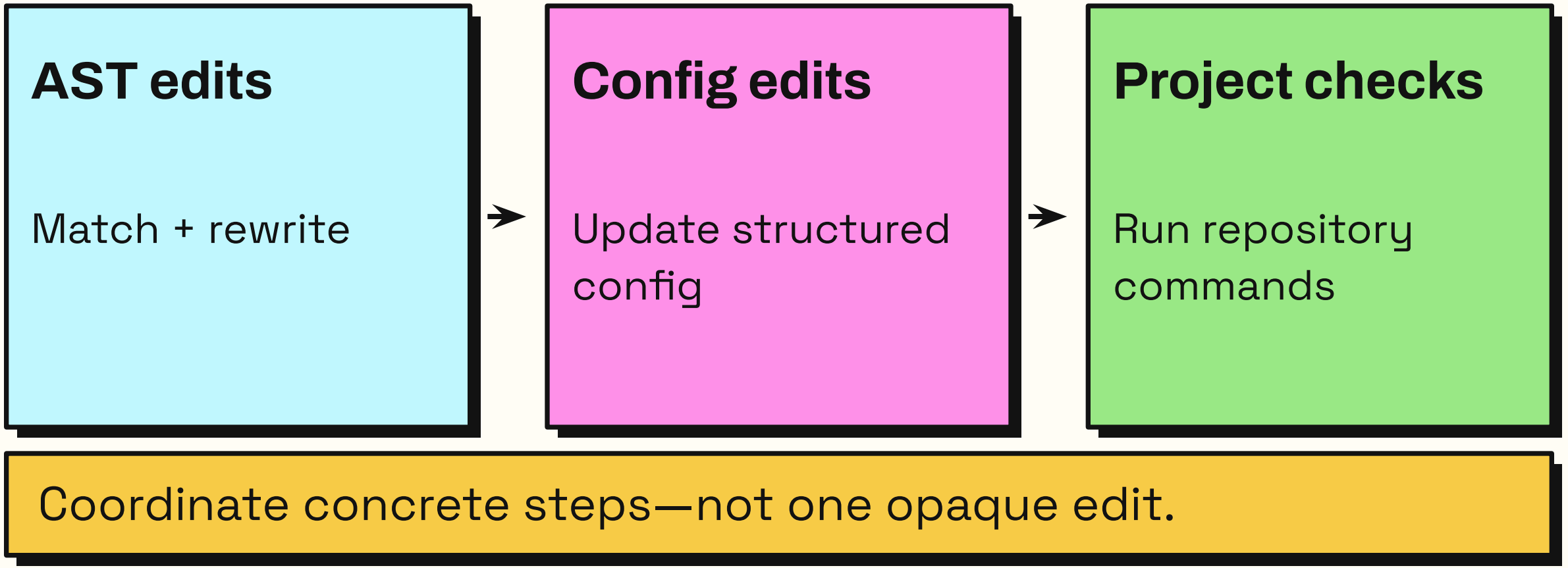
Repeatable.
Not infallible.

AI can help author a rule.

Deterministic execution applies known edits.

Neither replaces review.

One migration.
Different step types.



Author once.
Share the next rule.



Search existing rules before writing one.

Backstage moves.
Plugins must keep up.

You opened a feature PR.

First: update the host and affected APIs.

Keep the judgment. Automate the repetition.

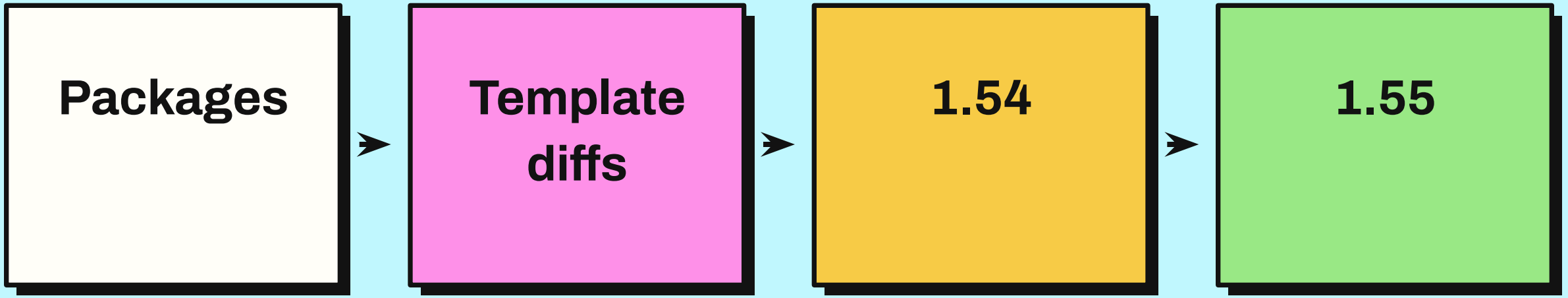
Introducing official Backstage codemods.

github.com/backstage/codemods

Versioned recipes + opt-in migrations

Community-built. Core-team approved.

One plugin.
1.53 → 1.55.



Prepare the host. Then run source recipes in order.

Two migration paths. Different timing.

Release upgrade

Versioned recipes

Follow release changes.

UI migration

MUI 4 → Backstage UI

Opt in when ready.

BACKSTAGE × CODEMOD

A plugin, in motion.

Badges · MUI 4 → Backstage UI

- 01 Dry run
- 02 Apply
- 03 Review
- 04 Agent
- 05 Cleanup
- 06 Test

Review stays with the maintainer.

01 / PREVIEW

See the changes first.

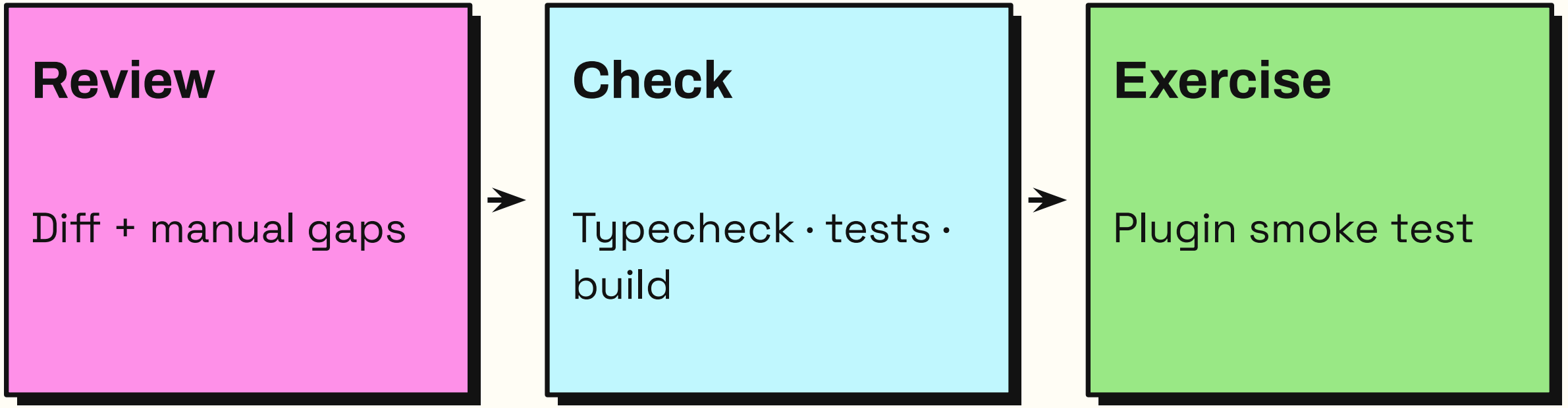
Terminal — badges / workspace

```
$ codemod run
  @backstage/mui4-to-bui-migration-recipe@0.4.1
  --dry-run

Resolving recipe · previewing changes…
```

Preview the recipe before touching the code.

A changed diff is not done.



Exit code ≠ merge criterion.

Codemod + skill.
Two jobs.

Codemod

Known source edits

Repeatable transforms.

Agent skill

Guided follow-up

Check · escalate · document.

ILLUSTRATIVE · PROPOSED WORKFLOW

From release to review

A possible maintenance loop for Backstage plugins

01 / RELEASE

Backstage release

A new version is published

CONCEPT, NOT A RECORDING OF A REAL RUN

RELEASE → ISSUE → AGENT → REVIEW

Help us find the edge cases.

Run a recipe on real code.

Share a minimal example + expected behavior.

Turn feedback into tests and better rules.

Backstage

Run a codemod.
Move Backstage forward.

github.com/backstage/codemods

Start with one covered upgrade.

Paul Schultz · Alex Bit

