

A/A, then A/B

The philosophy of comparing a release with production

Two kinds of difference

	Between	Tells you
Noise	two runs of the <i>same</i> image	nothing about the release: a server-rendered date, a race, a header derived from the moment
Release difference	production and the candidate	what the release actually changes

You cannot read the second until you have removed the first.

A/A first

- Every night: **production against itself** (`--mode noise`).
- Anything that differs is noise by definition.
- Each piece of noise is either made deterministic in the app or **masked**, with a reason.
- The run writes `noise-status.json`: clean or not, verified or not, and when.

The right to fail

A release run may return **FAIL** only when the A/A is

1. **clean**: zero failing hunks,
2. **verified**: every image pulled and its signature checked,
3. **fresh**: no more than seven days old,

or someone waived it explicitly with `--noise skip`, which the report records.

Otherwise the same findings come back as **WARN**, exit 0, *"advisory only: ..."*. The right to fail is re-earned every night.

Masks are blind spots you chose

- `normalise/masks.yaml` lists every field deliberately not compared.
- Each mask has a reason of at least twenty characters: what noise it hides and why that noise cannot be a regression.
- As narrow as possible: a header by name, a series by anchored prefix.
- Adding or loosening one is **its own pull request**, so nobody reviews a change and the blind spot that hides it together.
- Every report lists which masks fired. A mask that never fires is a mask to delete.

Claims: what the release meant

```
claims:  
- artefact: dom  
  scope: "reader:topic"  
  reason: "Rule 0262: the topic page gains a visually hidden h1"
```

- Every failing hunk must be claimed; first matching claim wins.
- A claim that matches nothing is **stale**: the release did not change what it said.
- A broad claim (`*`) needs a named person in `approvedBy`.
- Claims live with the release in the monorepo, never in the harness.

Statistics where they belong

- Deterministic artefacts need one run. **Timing** needs several: Mann-Whitney U over at least four runs per side.
- Timing is **advisory**. Each journey's production runs form an **XmR control chart**:
centre $\pm 2.66 \times$ mean moving range
- A candidate outside the limits is a **special cause to investigate**, never a failure.

Three comparisons through a release

When	a	b	Purpose
Every night	production	newest <code>main</code>	Main to RC: a forecast, never a gate
Release candidate	production	<code>vX.Y.Z-rc.N</code>	The gate: every difference claimed
After deploy	what was judged	live production	Post-deploy: production is what we tested

Same engines, same claims, same right to fail.