Configuration: Do you prove yours?

Infra Management Room

@abrianceau   in/abrianceau   alexandre@rudder.io

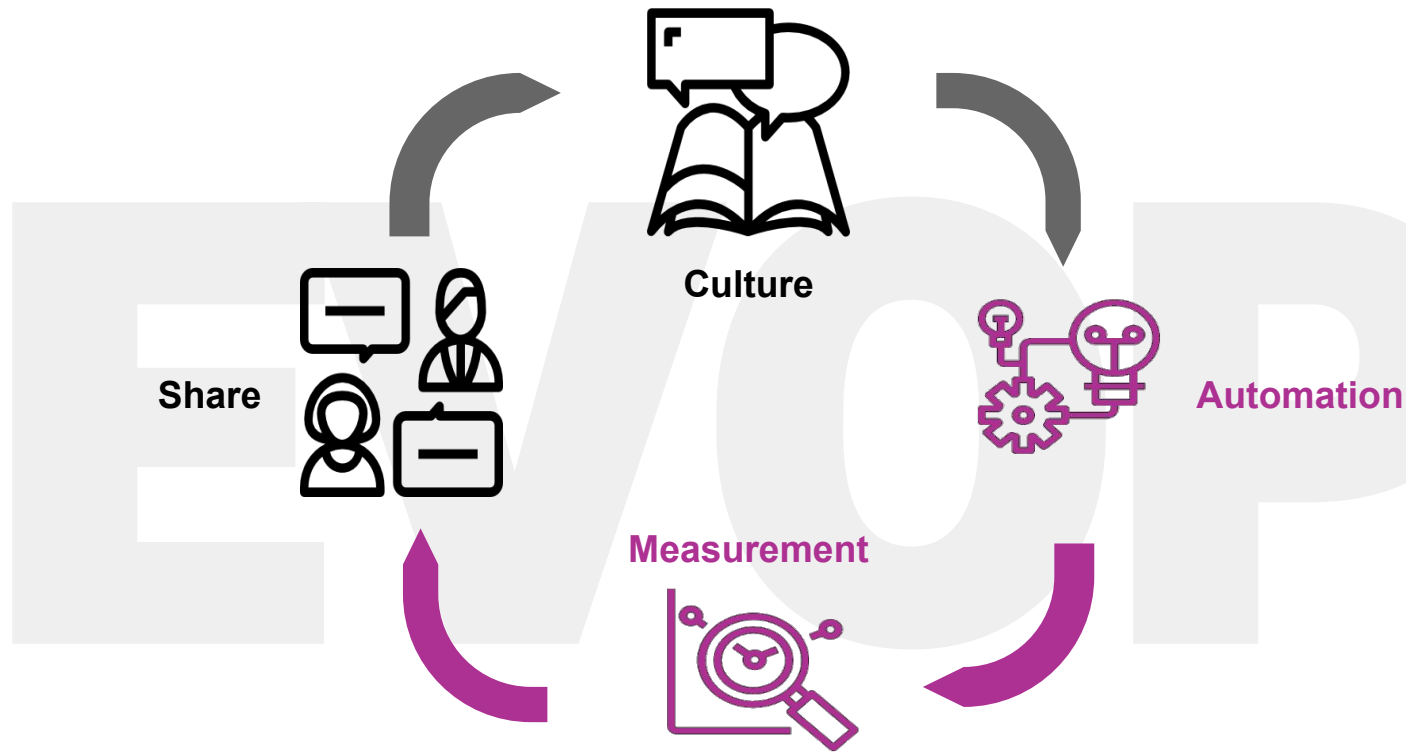# How are the servers doing?

"

Configuration management is a systems engineering process for establishing and maintaining **consistency** of a product [...] throughout its life.
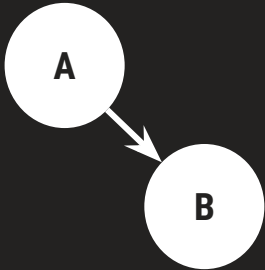
W Configuration_management

How devops can help to understand?

Culture

Automation

Share

Measurement

Why we need Observability in Configuration Management?

Causality

trust and prove
configuration states

Perspective

provide insights
relevant to different needs

Agency
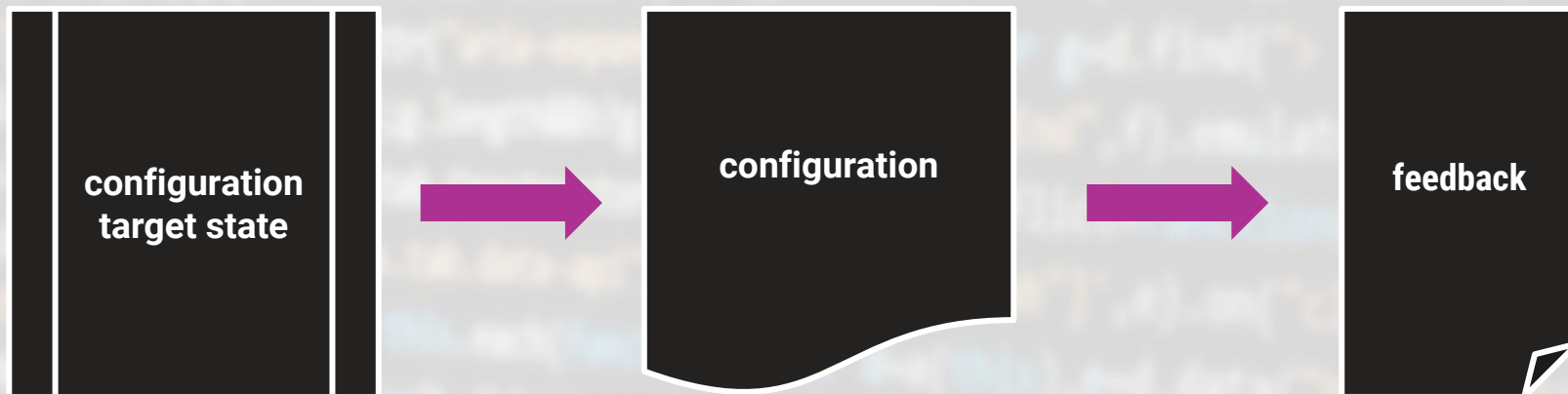
**help teams find
the best levers**
for their job

RUDDER

CONTINUOUS AUDITING & CONFIGURATION

Let's remember: What does configuration management do?

configuration target state → configuration → feedback

Let's remember: What does configuration management do?

configuration target state

configuration

configuration

configuration

feedback

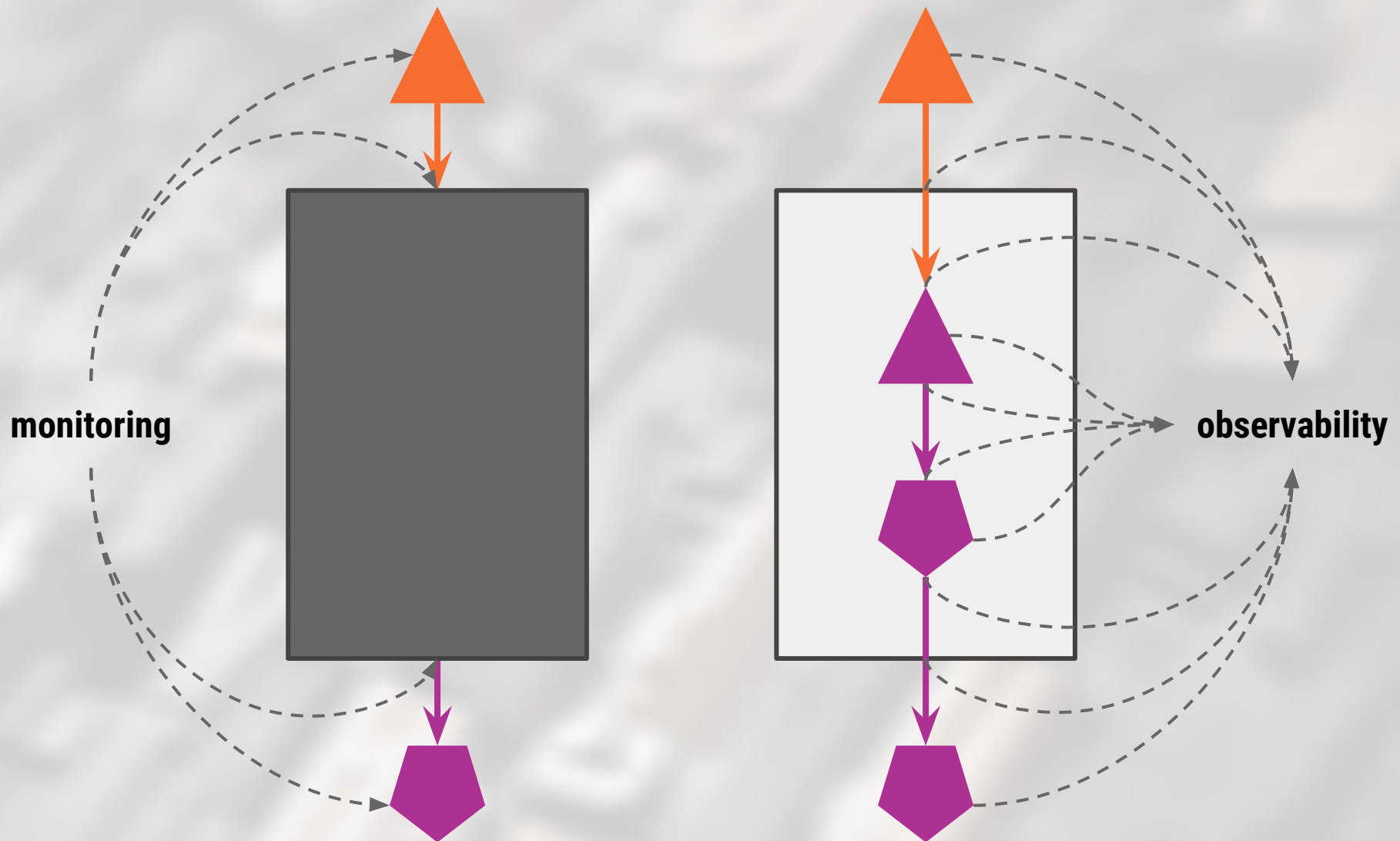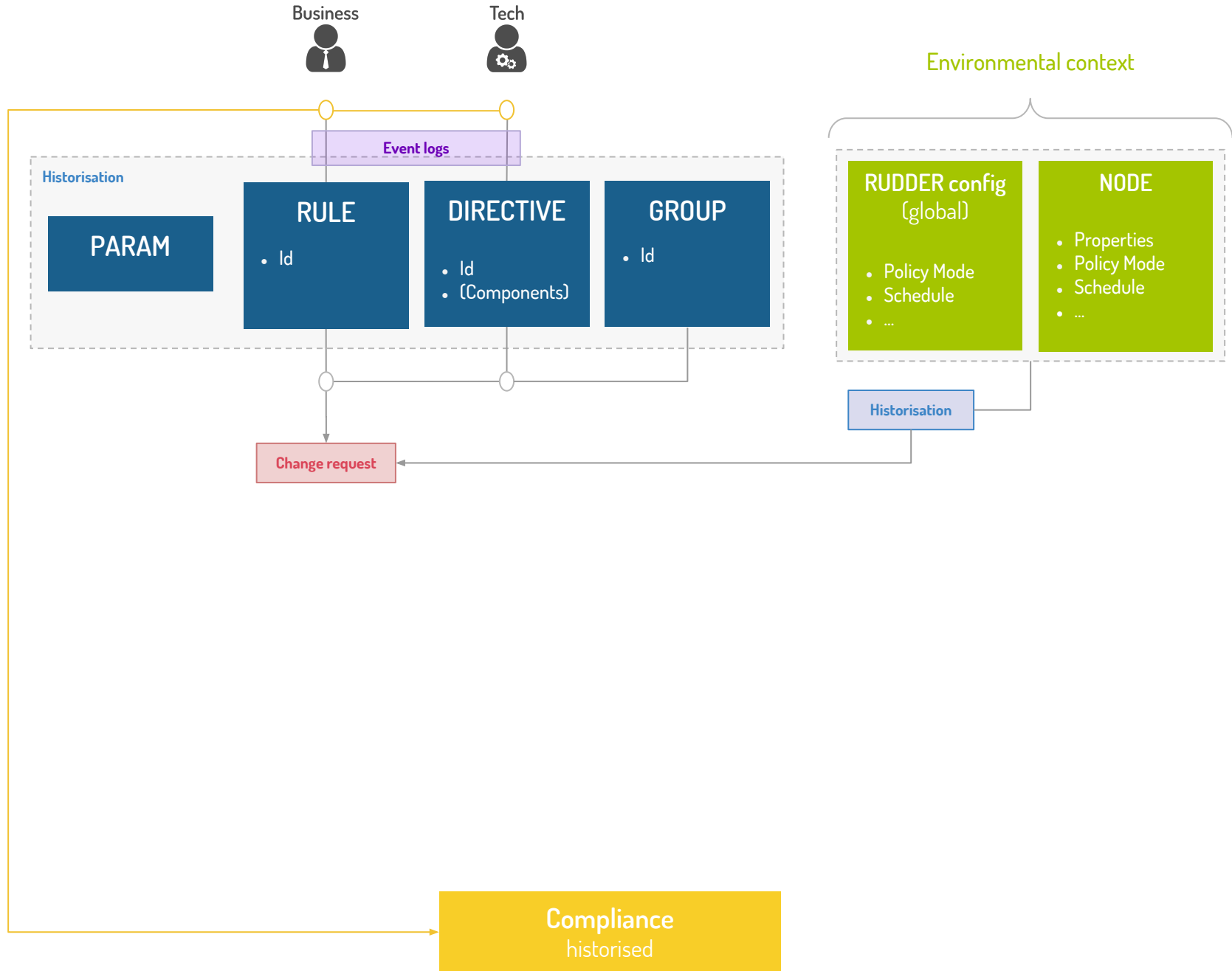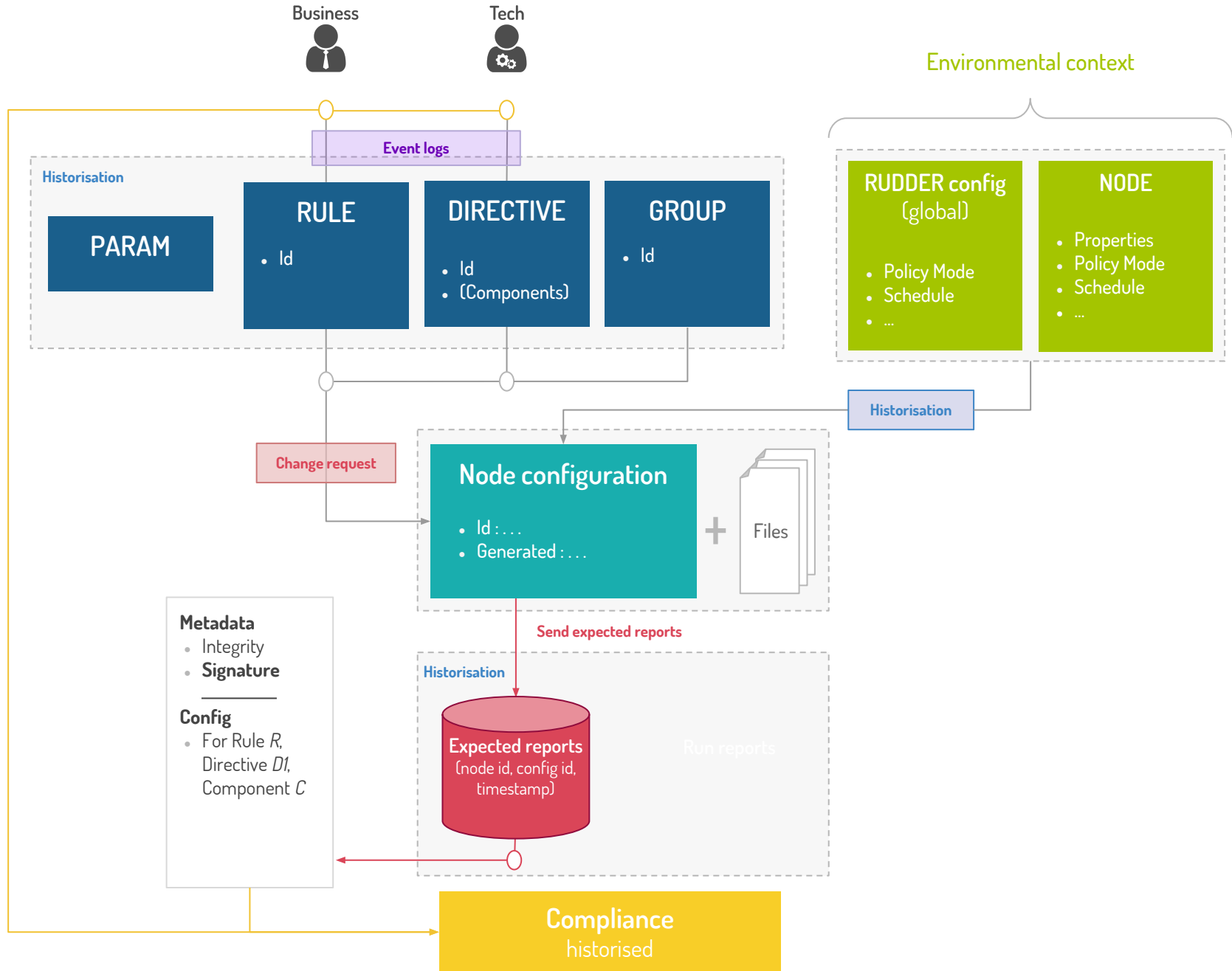**feedback**

feedback

"
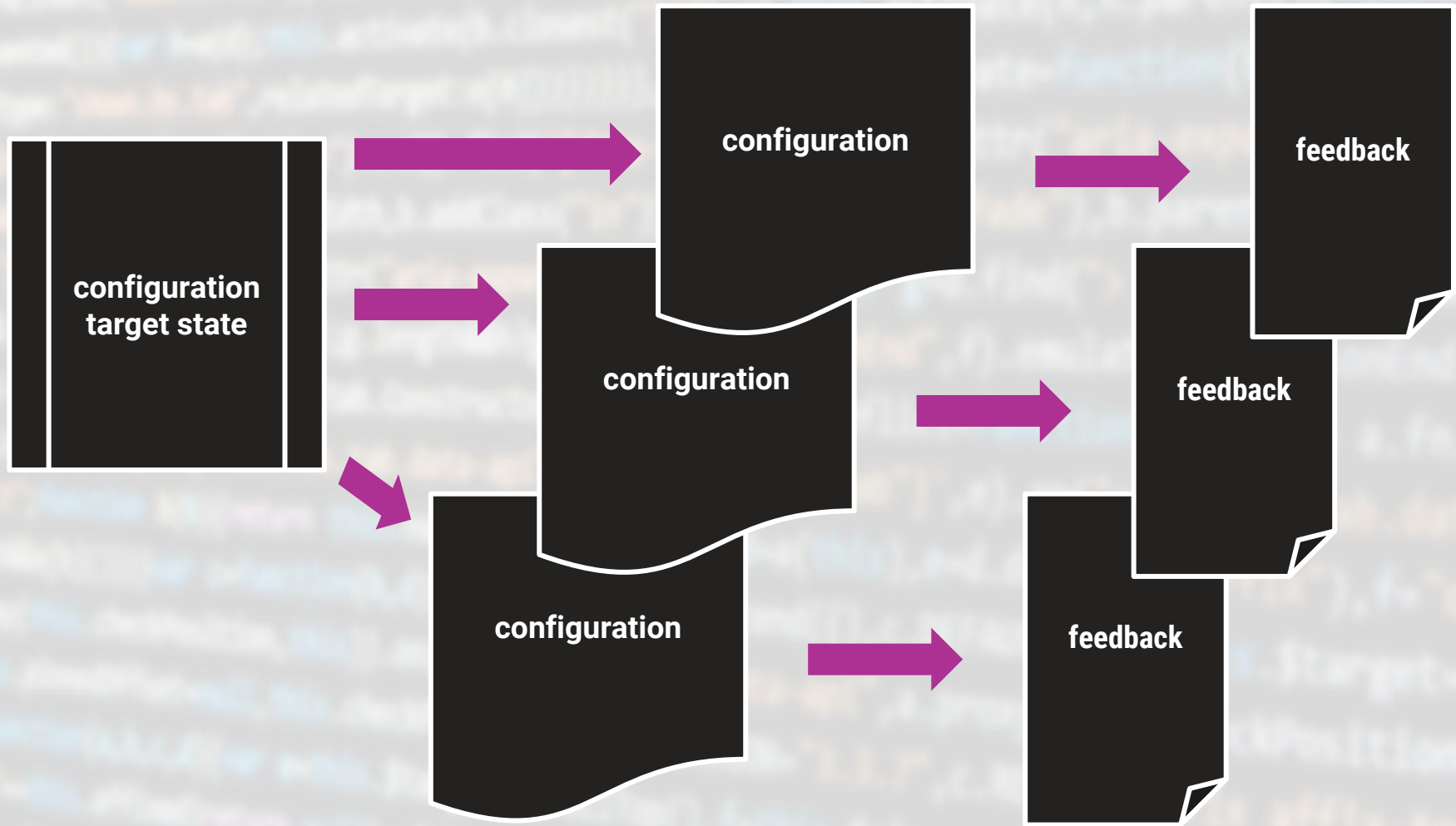
Observability is a measure of how well **internal states** of a system can be inferred from knowledge of its **external outputs**.

W Observability

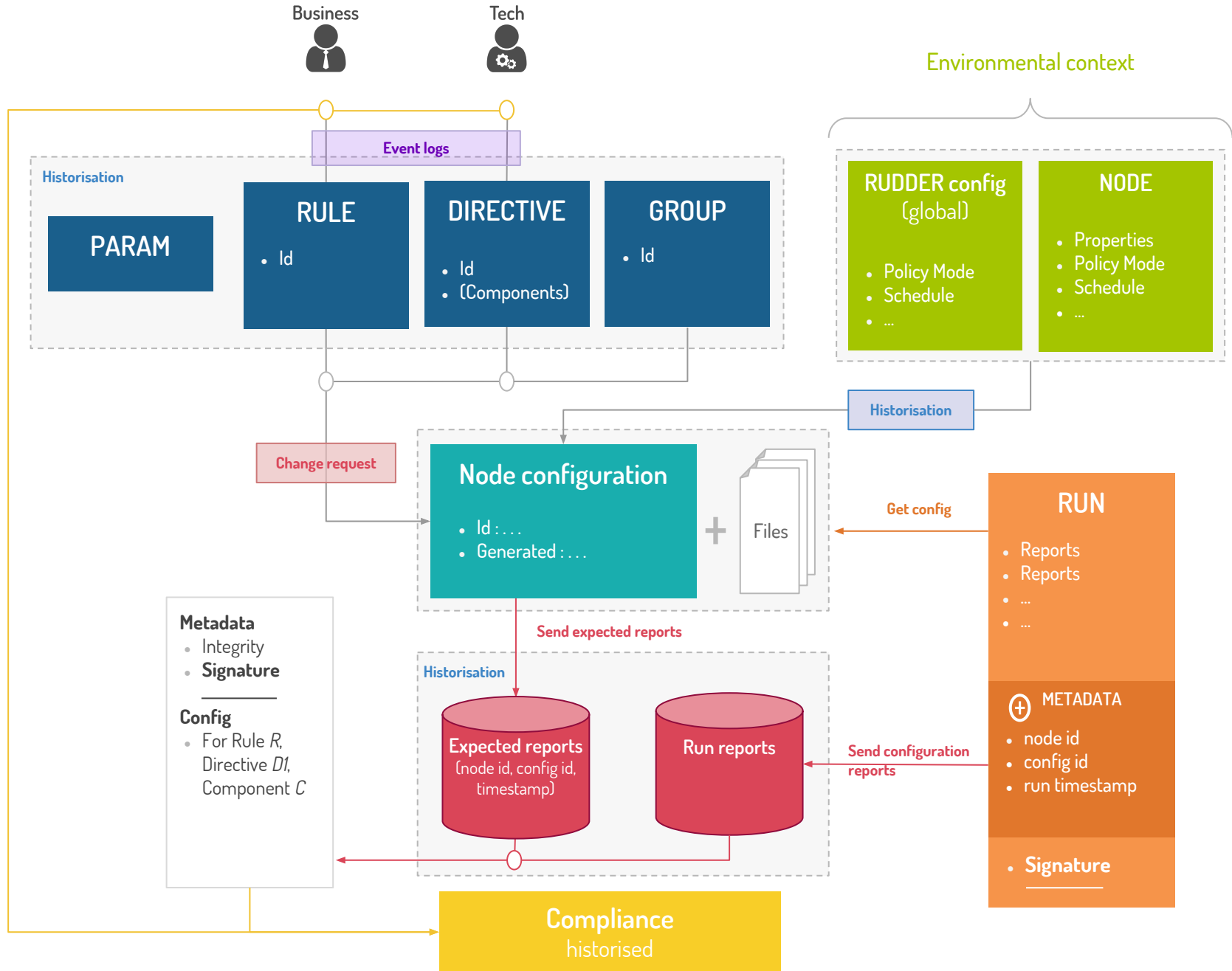**monitoring**

observability

Business Tech

Environmental context

Historisation

Event logs

PARAM

RULE
- Id

DIRECTIVE
- Id
- (Components)

GROUP
- Id

RUDDER config
(global)

- Policy Mode
- Schedule
- ...

NODE

- Properties
- Policy Mode
- Schedule
- ...

Historisation

Change request

Node configuration

- Id : . . .
- Generated : . . .

Files

Get config

RUN

- Reports
- Reports
- ...
- ...

Send expected reports

Metadata
- Integrity
- **Signature**

Config
- For Rule *R*, Directive *D1*, Component *C*

Historisation

Expected reports
(node id, config id, timestamp)

Run reports

Send configuration reports

⊕ **METADATA**
- node id
- config id
- run timestamp

- **Signature**

Compliance
historised

Impactful events category

Definition

of the target state

rules
node's groups
configurations

parameters
global
per nodes

environmental context
properties
policy mode...

Creation

of the configuration

**tracing the configuration
generation process**
OS specificity
template
variables...

predict configuration state
anticipation of any change
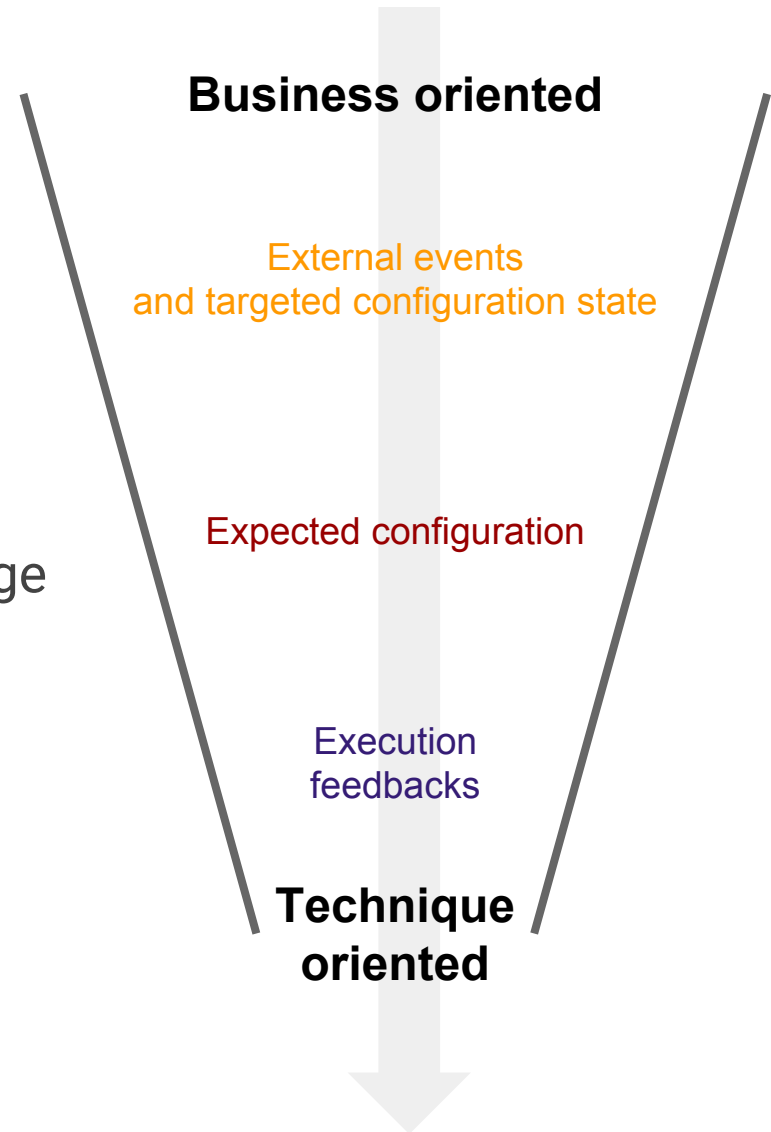used to compare with the execution

Execution

of the configuration

execution feedback
fine grained comparison with the
expectation
prevent any alteration

Are you sure that I didn't forget something ?

- **And on the node itself ?**

- **Observability should be agnostic**
  - Protocol normalization

- **Beyond 1st level of external output usage**
  - Data mining
  - Process mining and AI

Business oriented

External events
and targeted configuration state

Expected configuration

Execution
feedbacks

**Technique
oriented**

Configuration: Do you prove yours?

Infra Management Room

@abrianceau   in/abrianceau   alexandre@rudder.io