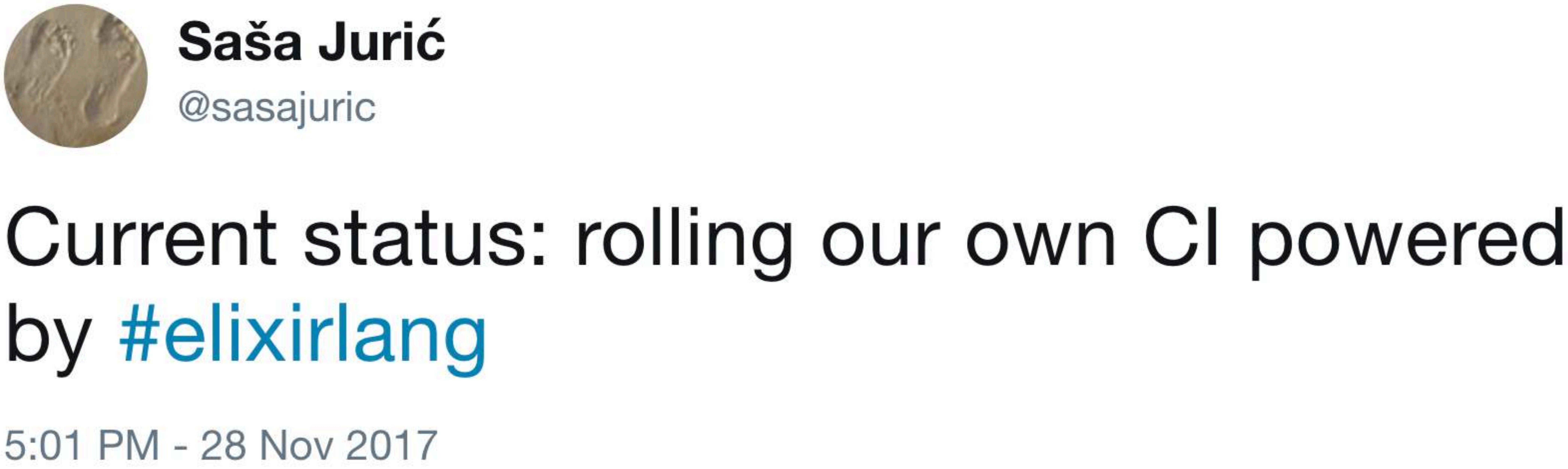
Such great heights

@sasajuric

aircloak.com

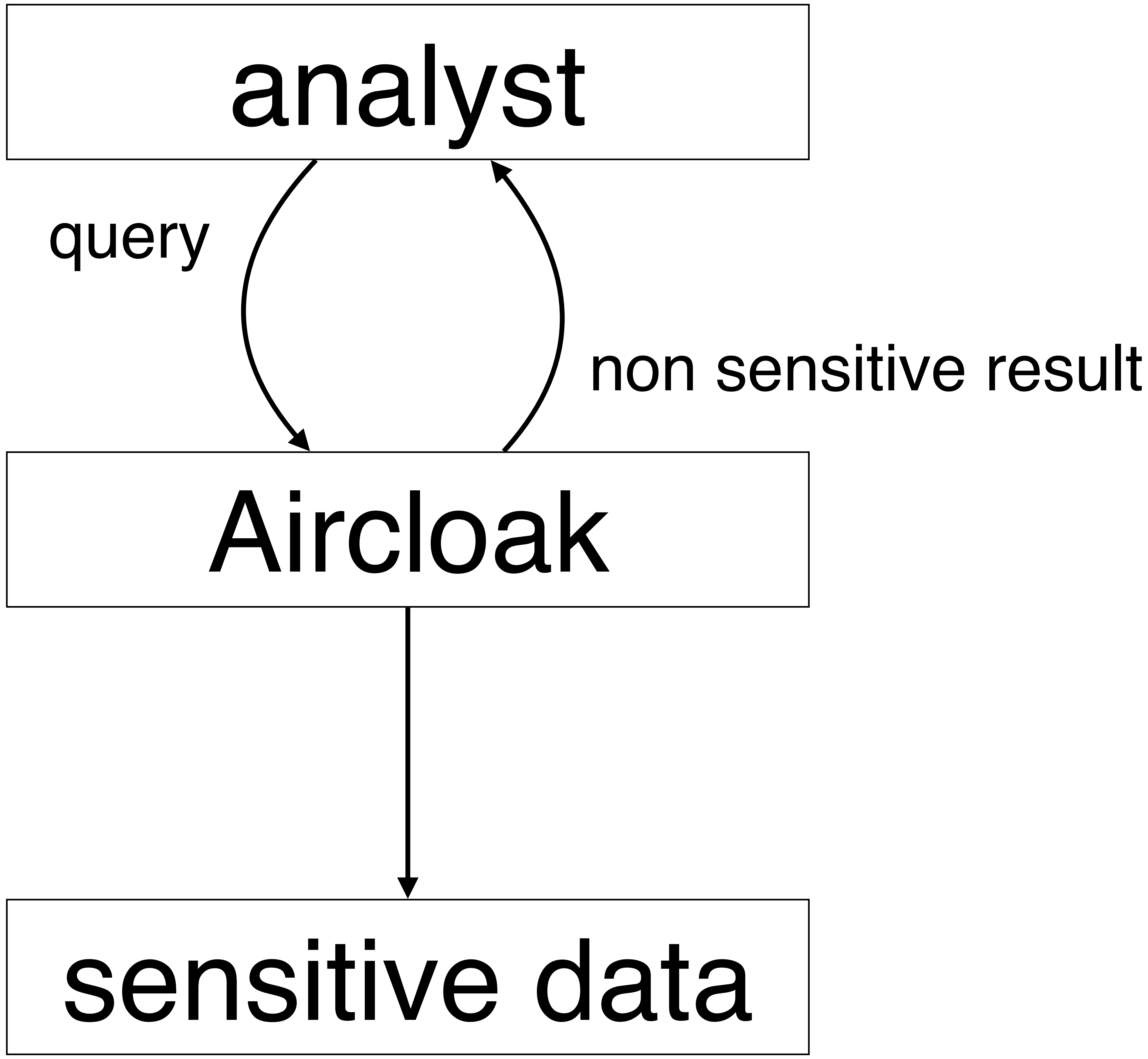


Saša Jurić

@sasajuric

Current status: rolling our own CI powered by #elixirlang

5:01 PM - 28 Nov 2017



analyst

query

non sensitive result

Aircloak

sensitive data

```
aircloak
  air
  cloak
  central
  common
  integration_test
  ...
```

dev flow

pull request based
mandatory code review
mandatory tests

outgrowing Travis CI

long build times
flakiness
low throughput

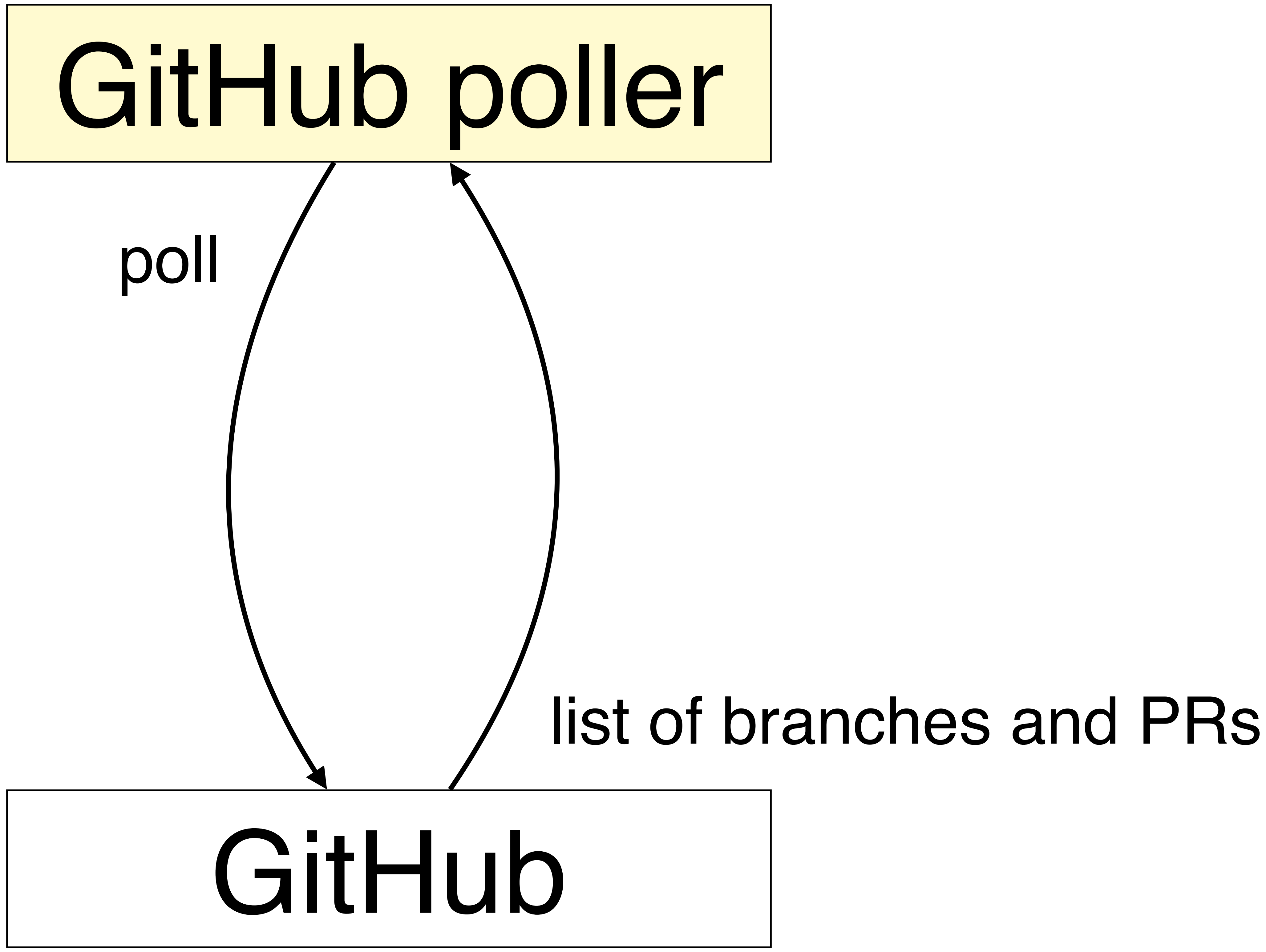
solution

self-hosting

BEAM powered CI

best tool for the job :-)
reduced technical overhead
total control and flexibility

```
aircloak
  air
  cloak
  central
  common
  integration_test
  ci
  ...
```

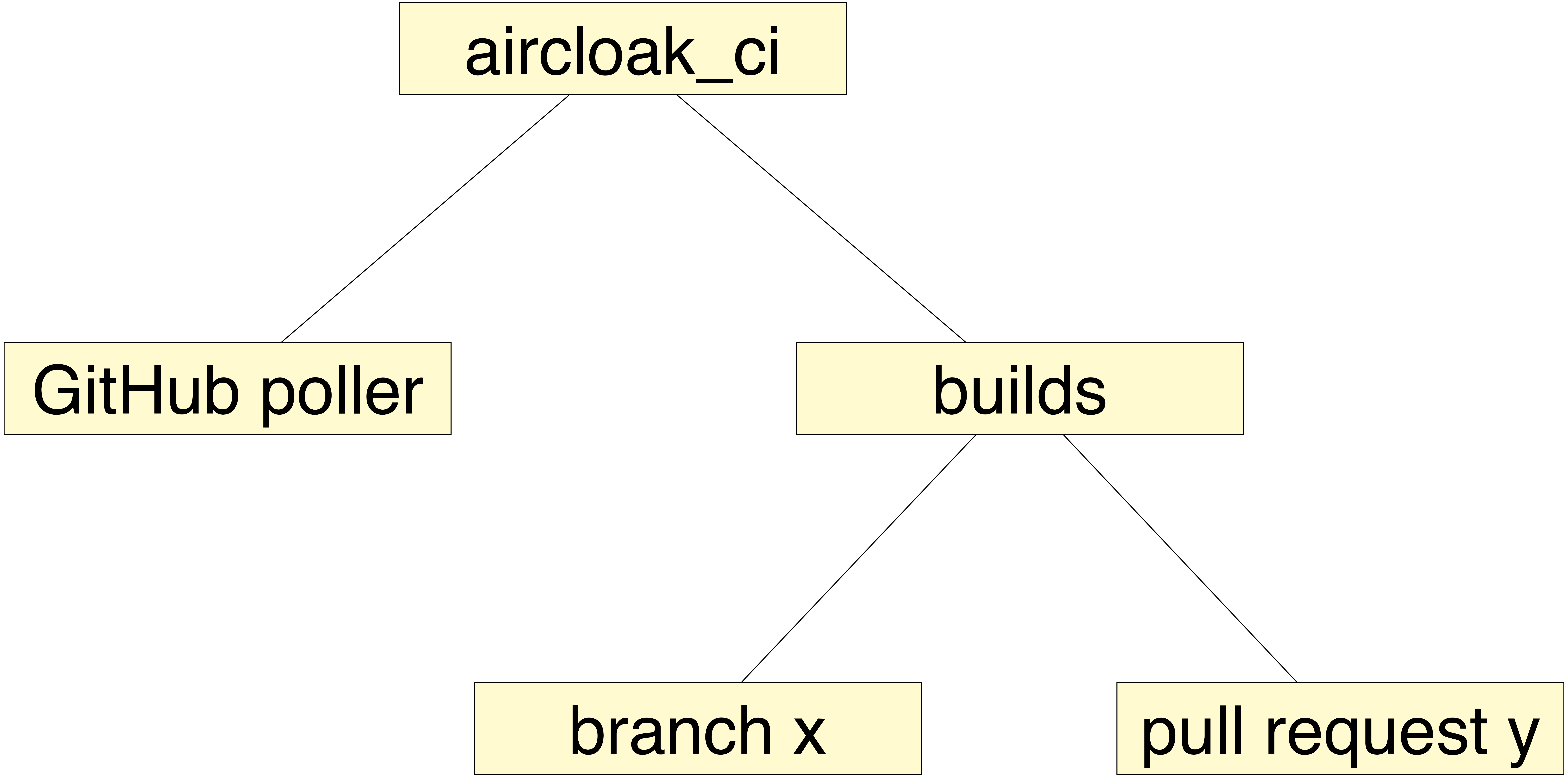


GitHub poller

poll

list of branches and PRs

GitHub



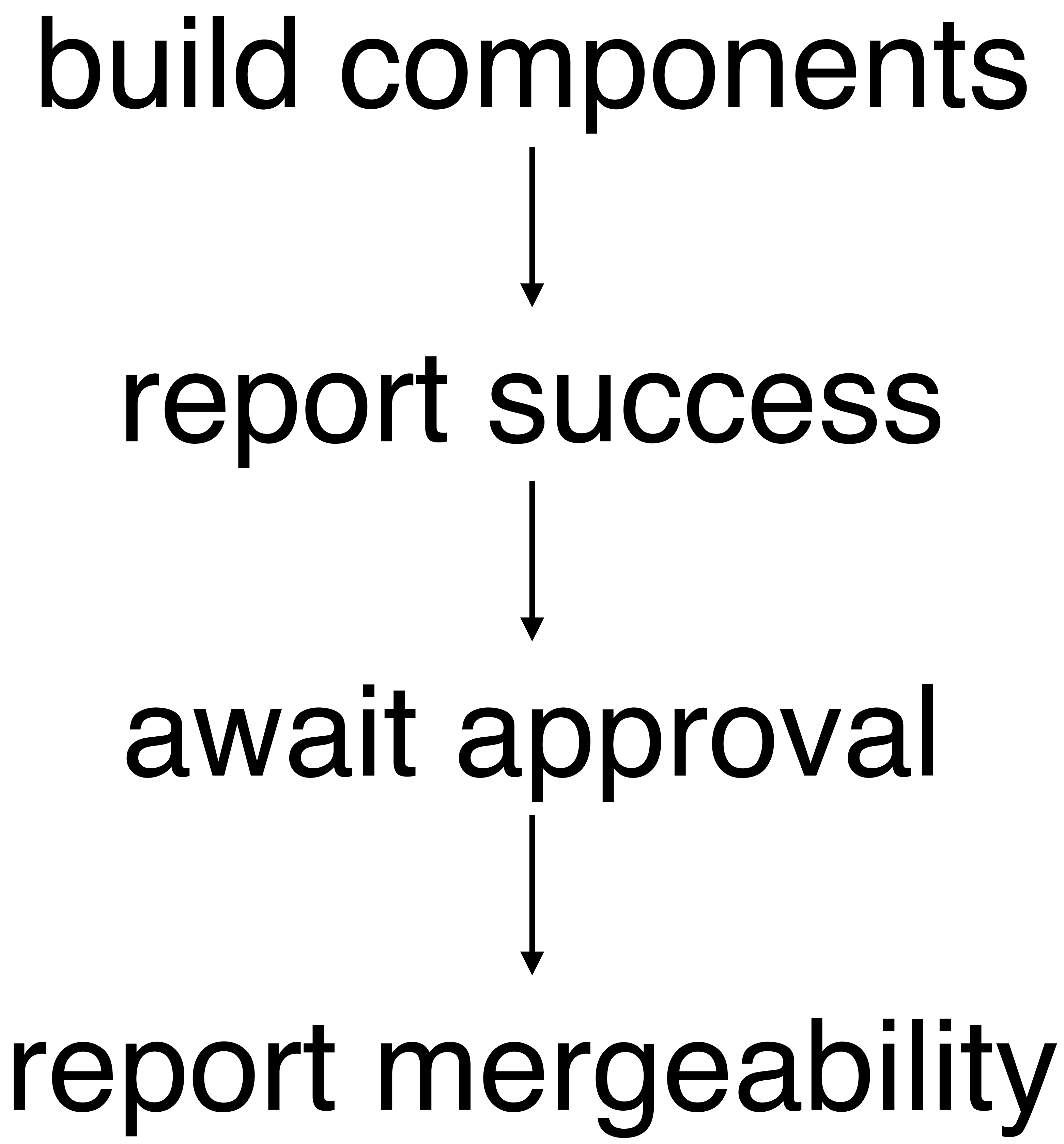
aircloak_ci

GitHub poller

builds

branch x

pull request y



build components

report success

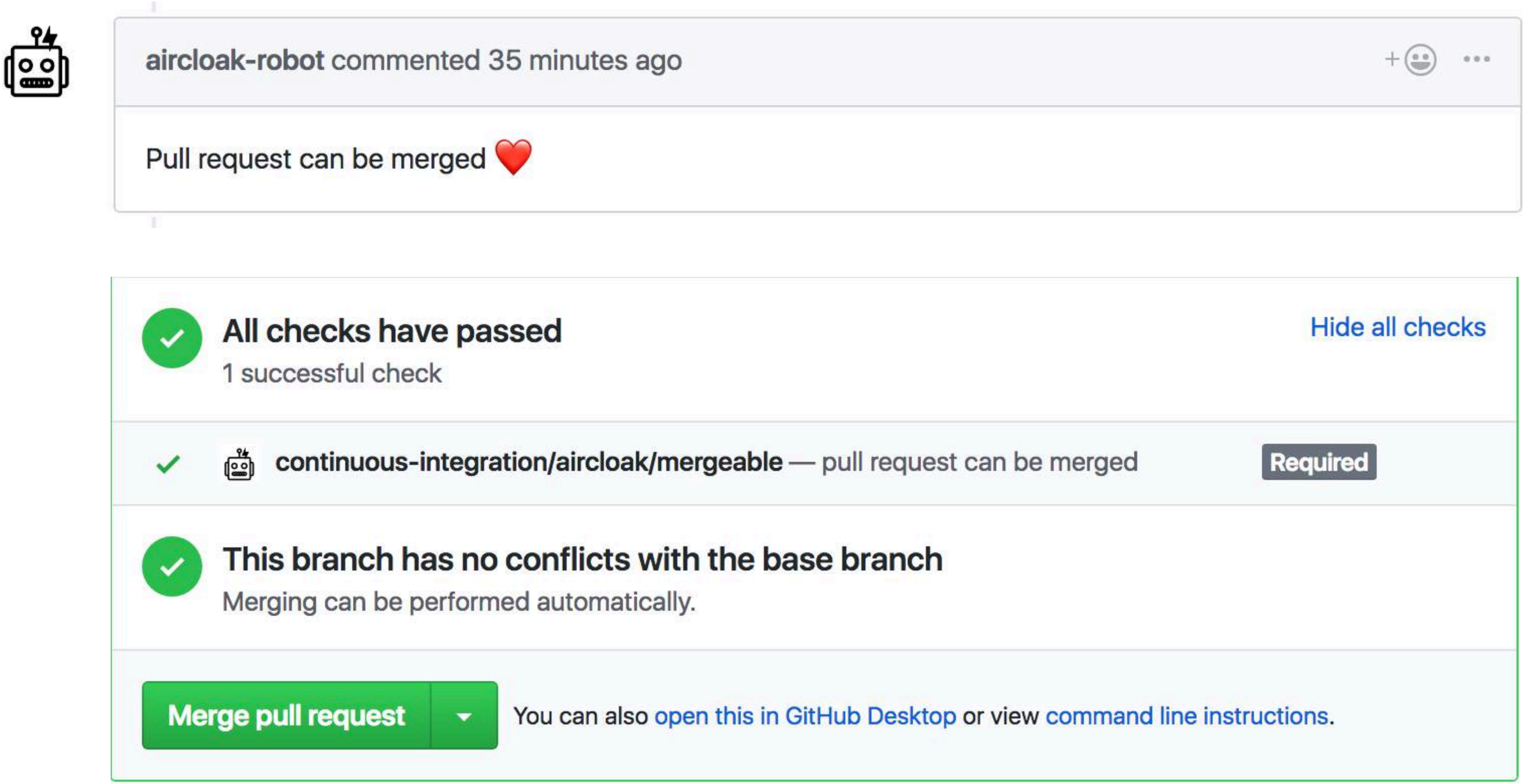
await approval

report mergeability

```
if not merge_conflict?(build_state) and
     tests_passed?(build_state) and
     pr_approved?(build_state)
   do: report_mergeable(build_state)
```

```
Github.comment_on_issue(
  build_state,
  "Pull request can be merged #{AircloakCI.Emoji.happy()}"
)

Github.put_status_check_state(
  build_state,
  "continuous-integration/aircloak/mergeable",
  "pull request can be merged",
  :success
)
```



aircloak-robot commented 35 minutes ago

Pull request can be merged ❤️

All checks have passed

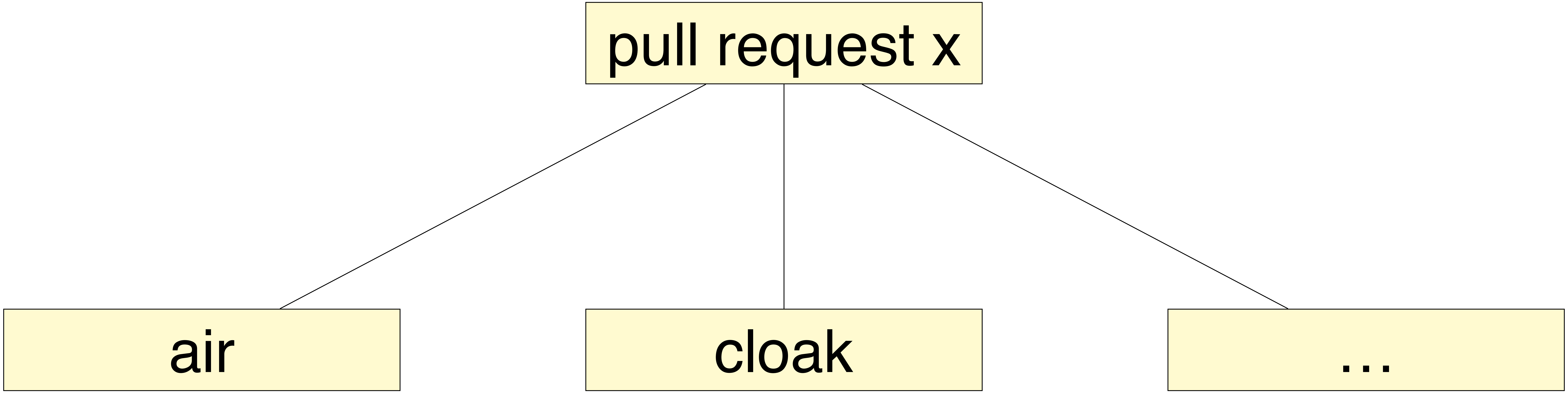
1 successful check

✓ continuous-integration/aircloak/mergeable — pull request can be merged

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.



pull request x
air
cloak
...

changed component	**tested components**
air	air integration_test
ci	ci
common	everything

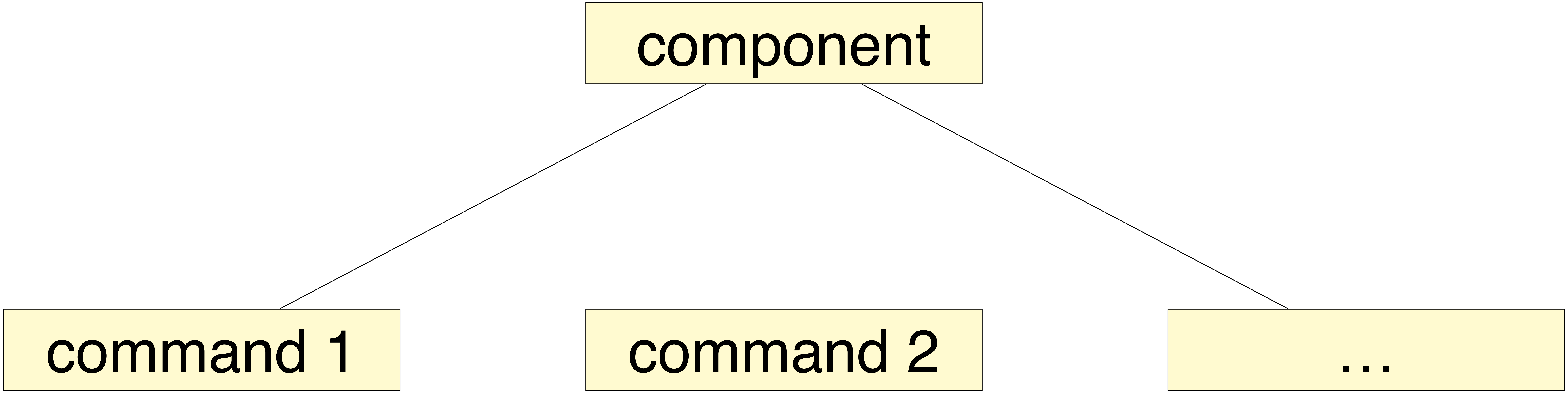
DSL

```
[
  "mix deps.get",
  "mix compile",
  "mix format --check-formatted",
  "mix test",
  "mix dialyze",
  # ...
]
```

```
{:sequence, [
  "mix deps.get",
  "mix compile",
  {:parallel, [
    "mix format --check-formatted",
    "mix test",
    "mix dialyze"
  ]},
  # ...
]}
```

```
test_commands = # ...
dev_commands = # ...
prod_commands = # …

{:sequence, [
  "mix deps.get",
  {:parallel, [
    test_commands,
    dev_commands,
    prod_commands
  ]}
]}
```

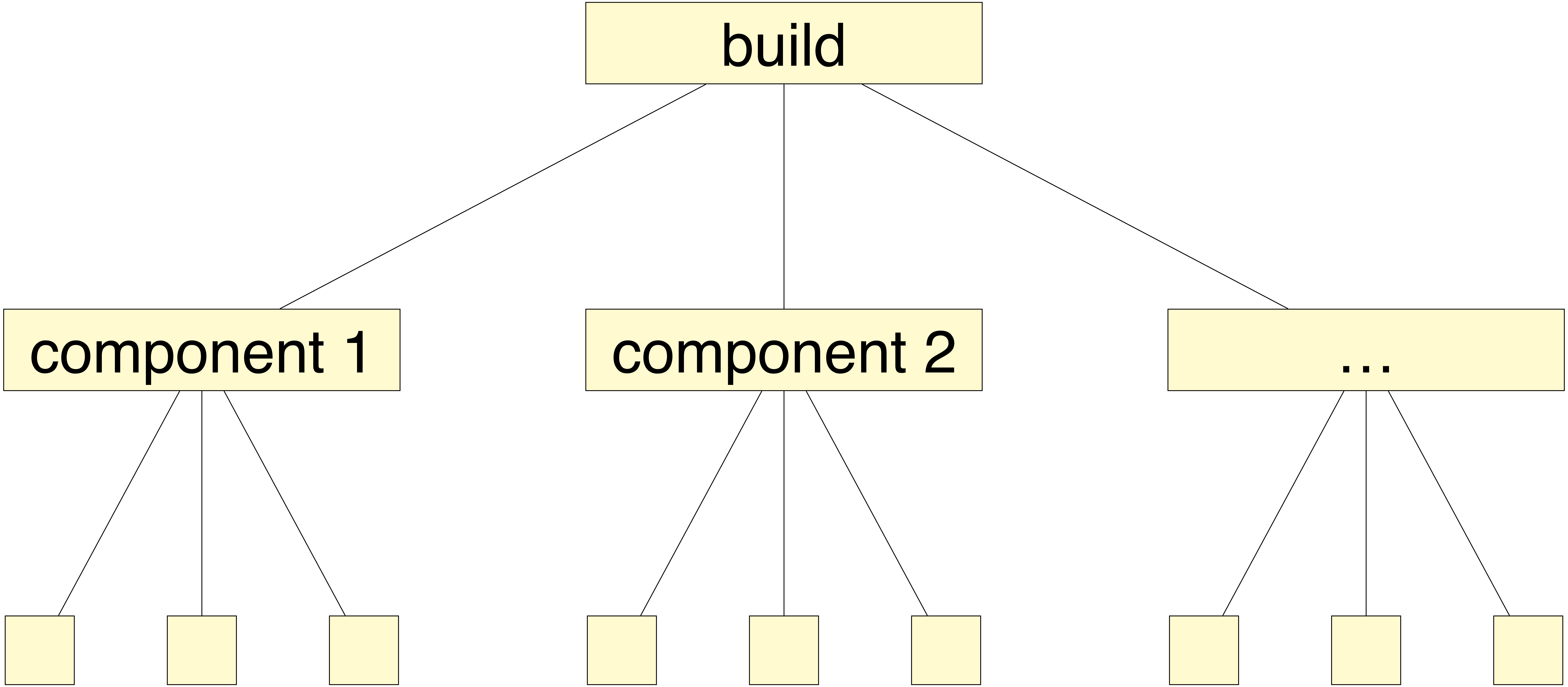


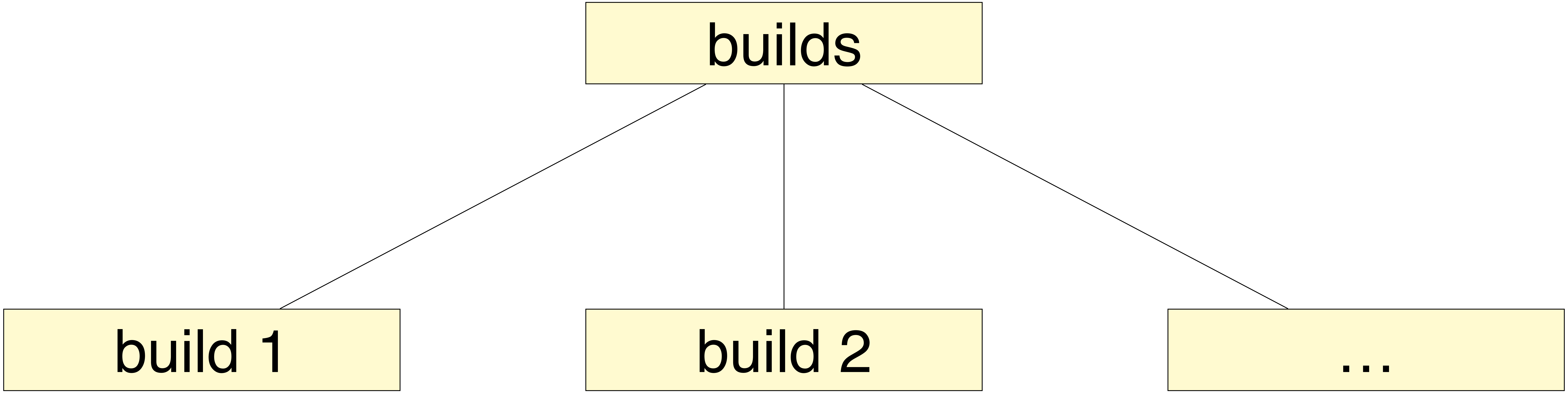
component

command 1

command 2

…


build
component 1
component 2
…



builds

build 1

build 2

…

not open-source (yet!)

Technical requirement	Server A	Server B
HTTP server	Nginx and Phusion Passenger	Erlang
Request processing	Ruby on Rails	Erlang
Long-running requests	Go	Erlang
Server-wide state	Redis	Erlang
Persistable data	Redis and MongoDB	Erlang
Background jobs	Cron, Bash scripts, and Ruby	Erlang
Service crash recovery	Upstart	Erlang

uniformity

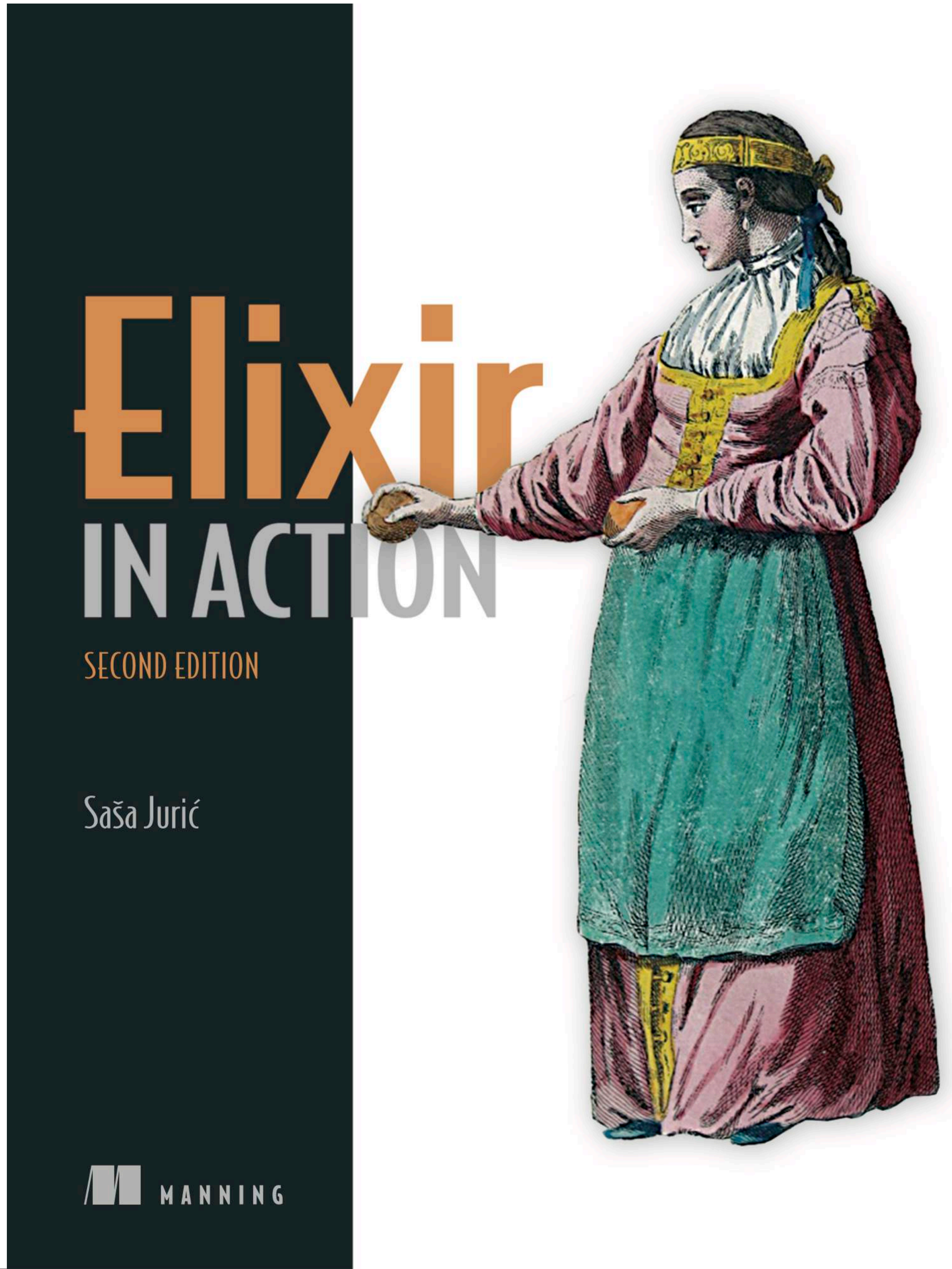
simpler development
simpler testing
simpler deployment
simpler maintenance
improved collaboration

technical feature	built-in options
services	GenServer, Task
service manager	Supervisor
service registry	Registry
k-v	ETS, Mnesia
web server	Phoenix, Plug, Cowboy
periodical jobs	:timer, Quantum, Periodic
load regulation	Jobs, Safetyvalve, Sbroker
concurrent flows	Flow, GenStage

distributed systems
persistence
partial deployment

They will see us waving from such great heights
Come down now, they'll say
But everything looks perfect from far away
Come down now, but we'll stay

The Postal Service


Elixir
IN ACTION
SECOND EDITION
Saša Jurić
MANNING