FAST AND RESILIENT INTEGRATION TESTING

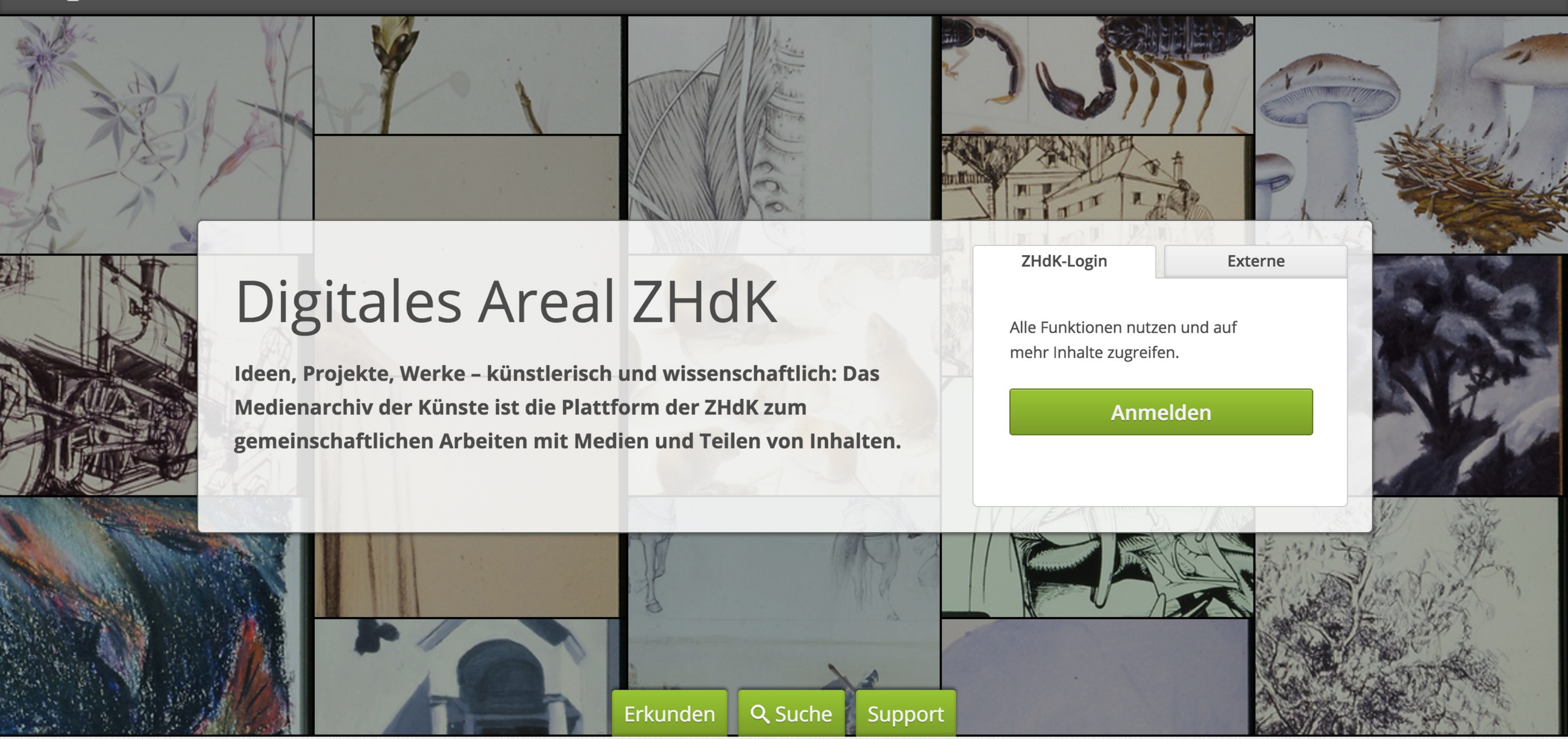
CONTINUOUS LIFECYCLE 2015

Dr. Thomas Schank

Max F. Albrecht

http://drtom.ch/talks/2015/CL

Version 1.1.0



Digitales Areal ZHdK

Ideen, Projekte, Werke – künstlerisch und wissenschaftlich: Das Medienarchiv der Künste ist die Plattform der ZHdK zum gemeinschaftlichen Arbeiten mit Medien und Teilen von Inhalten.

ZHdK-Login | Externe

Alle Funktionen nutzen und auf mehr Inhalte zugreifen.

Anmelden

Erkunden | Suche | Support

ZHdK-Katalog

Alle anzeigen →

MADEK TEAM & US

🏛 ZURICH UNIVERSITY OF THE ARTS

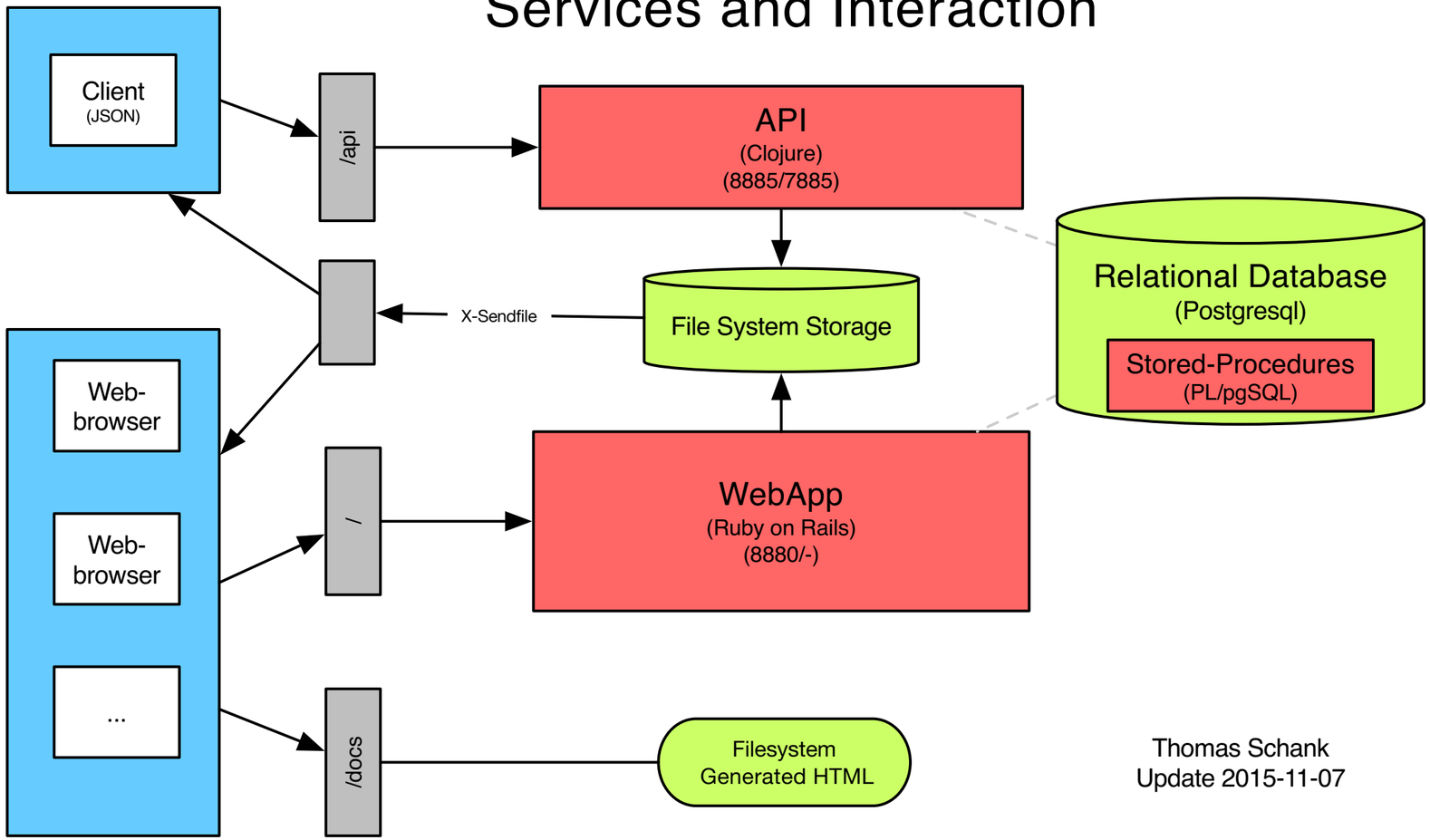
- Thomas: Software-Architect, Developer, CI-Infrastructure
- Max: Frontend Software-Engineer, Meta-Data Concepts

MADEK - MEDIENARCHIV DER KÜNSTE

ARCHITECTURE & TECHNOLOGIES

- Ruby on Rails, Clojure
- React with progressive enhancement
- 3-tier web-application
- towards micro-services
- deployment via Ansible to private cloud

Madek 3.0 Architecture
Services and Interaction



Thomas Schank
Update 2015-11-07

MADEK TESTING

"specification by example"

→ integration testing

→ components interaction

1. THE PROBLEM

MADEK PROJECT 2012

many new features, many new tests

- testing time 1 1/2 - 2 hours, increasing
- more and more failing tests: **false negatives**
- 1/8 builds pass

TRY TO IMPROVE TESTS

- very time and resources consuming
- improvement for some time
- new features and new tests made efforts futile

MANUAL RETRYING

automated tests, local retries

automatic → semi automatic testing

2. COMPREHENSION

PROBABILITY OF A FALSE NEGATIVE FOR A WHOLE TEST-SUITE

	Expression	Example
probability false negative single test	p_f	3%
probability "success"	$p_s = 1 - p_f$	0.97
number of tests	n	100
probability "success" whole suite	$\mathbf{P_s = p_s^n = (1 - p_f)^n}$	$\approx 5\%$

→ only one out of 20 will pass as it should

"succes" = true positive

WHY RETRYING WORKS SO WELL

let k number of independent retries per test

$$P_s(n) = (1 - p_f)^n \Rightarrow \mathbf{P'_s(n, k) = \left(1 - p_f^k\right)^n}$$

Expected successful outcome for $n = 100$ and $p_f = 0.03$

k	P'_s
1	5%
2	91%
3	99.7%

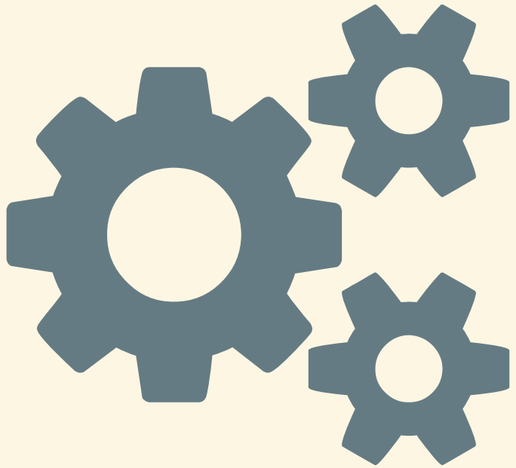
2. COMPREHENSION - CONCLUSION

- more tests → **exponential** increase of likeliness for false negatives
- compensate by **retrying** single tests just a **few times**

→ retrying is not an anti-pattern

→ it can be a necessity

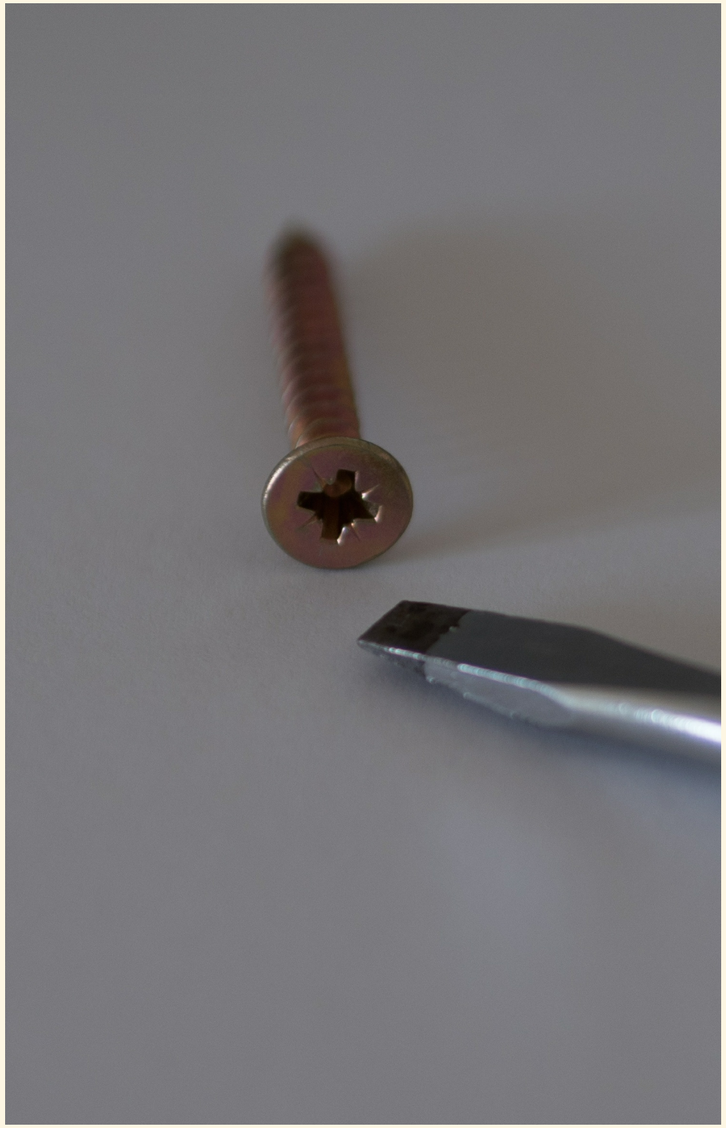
3. IMPLEMENTATION



JENKINS

- fall 2012
- build creates other builds via the Jenkins API
- last build aggregates
- solved false negative problem (partly)
- testing time: 15 - 25 minutes

→ it worked



- frequent code pushes interfere
- "REST-like style API" → not much like REST
- considerable effort and maintenance

→ Jenkins and "CI-X" just aren't made for this

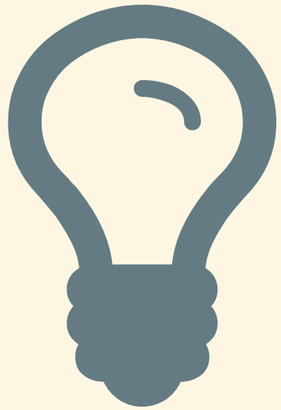
CIDER-CI

homegrown solution, started in spring 2013

- inherent support for **retries** and **parallelization**
- test **reproducibility**
- tight **integration** with **source code**
- **manage services** while testing
- support everything from **testing** to **deployment**

ready to use in fall 2013, **never looked back**

4. CONCEPTS IN CONTEXT

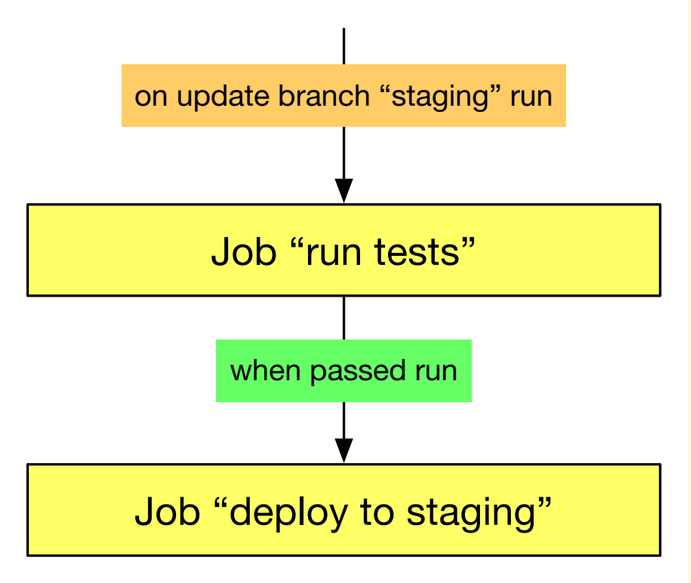


JOBS

EXAMPLES

- run test-suite
- perform static code checks
- build
- deploy

jobs can be **triggered** and can **depend on each other**



PROJECT CONFIGURATION

`cider-ci.yml` file in the project

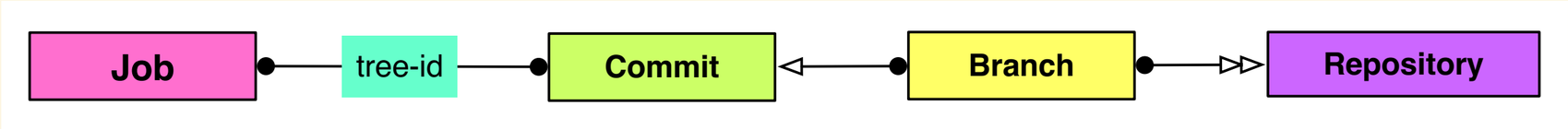
```
jobs:
  deploy_test:
    name: Deploy to test

    depends-on:
    - type: job
      job: integration-tests
      states: [passed]

    run-on:
    - type: branch
      include-match: ^master$

    # specify tasks etc
```

The source is the truth.

configuration: reproducible, reviews, audits ???

CIDER-CI AND THE SOURCE CODE

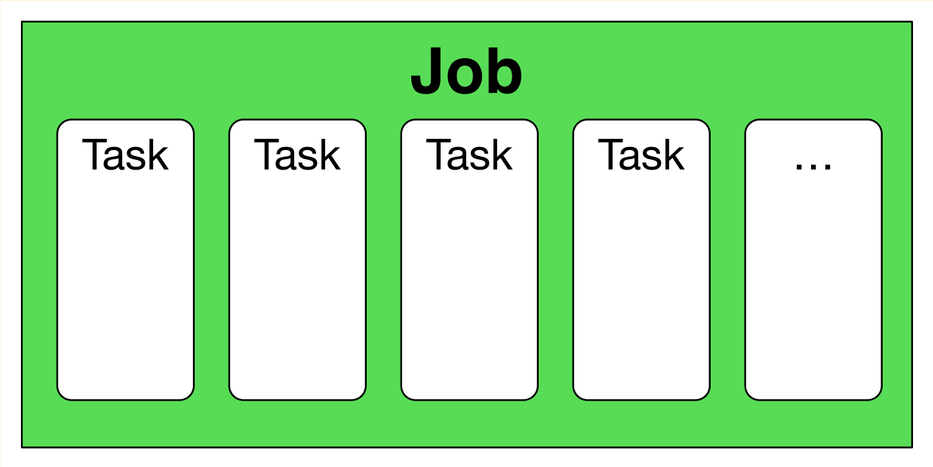
Cider-CI "knows" about commits, branches, submodules,…



`tree-id`: fingerprint of your source code

- **reproducibility**
- jobs can be **run at any time** (later)
- **binary search** for "bad" commits
- commit amends, squashing: **existing job remains valid**

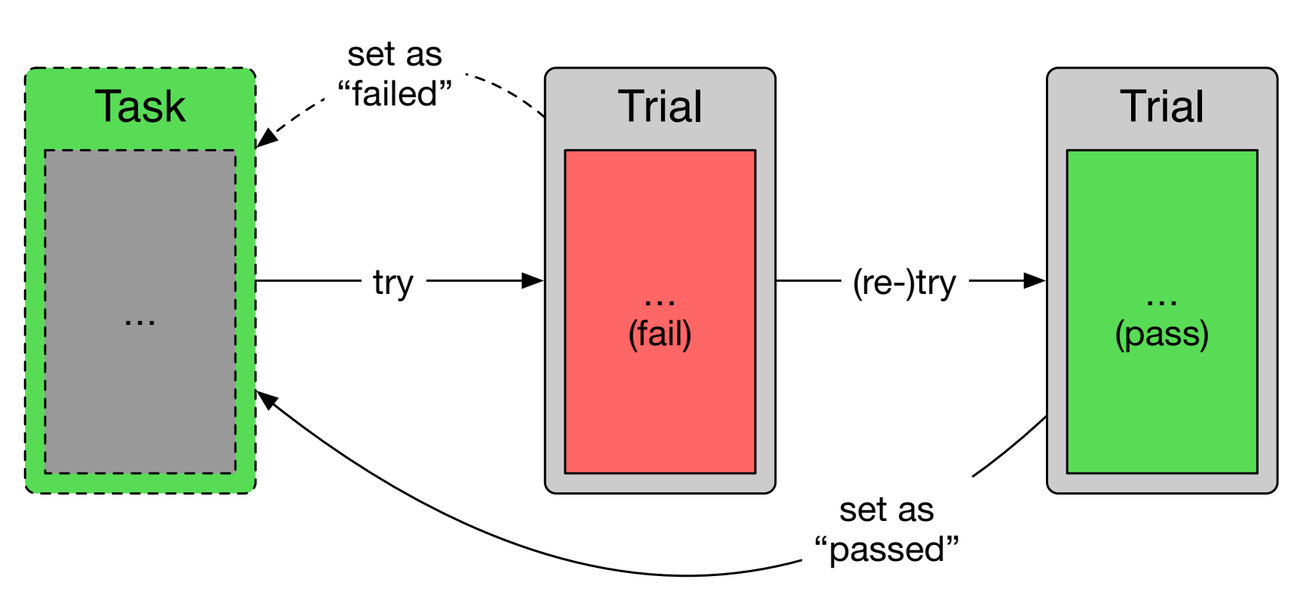
JOBS & TASKS



job: container and state aggregate for tasks

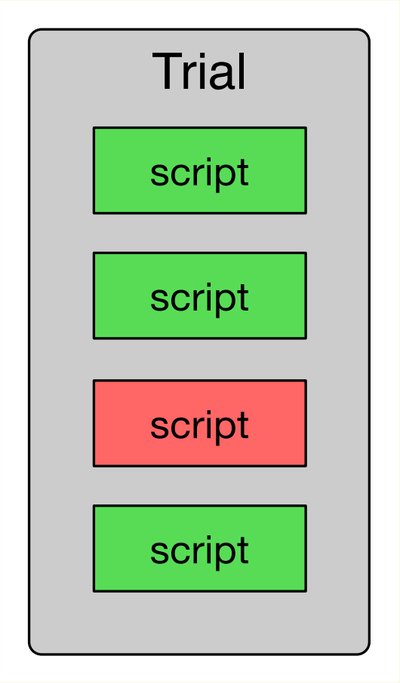
→ **parallelization**

TASKS & TRIALS



- **blueprint**
- container and state aggregate for trials

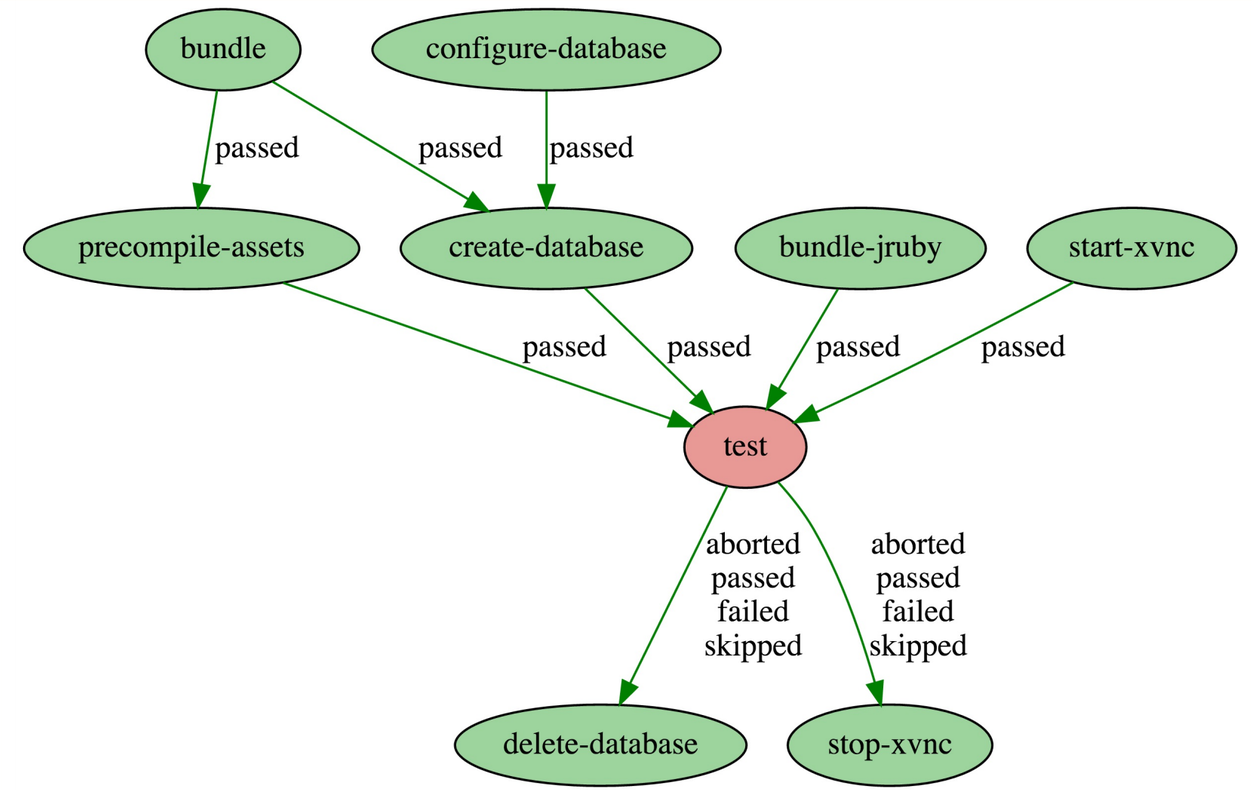
→ resilience



TRIAL & SCRIPTS

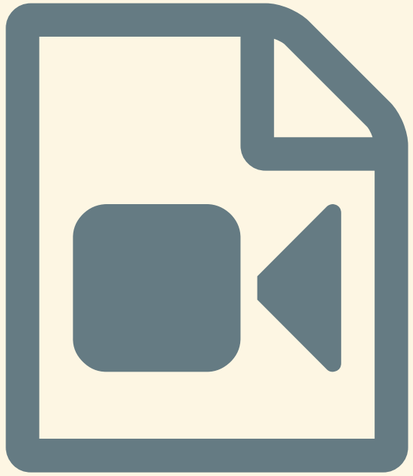
- actual unit of execution
- executed in the **same context**
- **depend** on each other

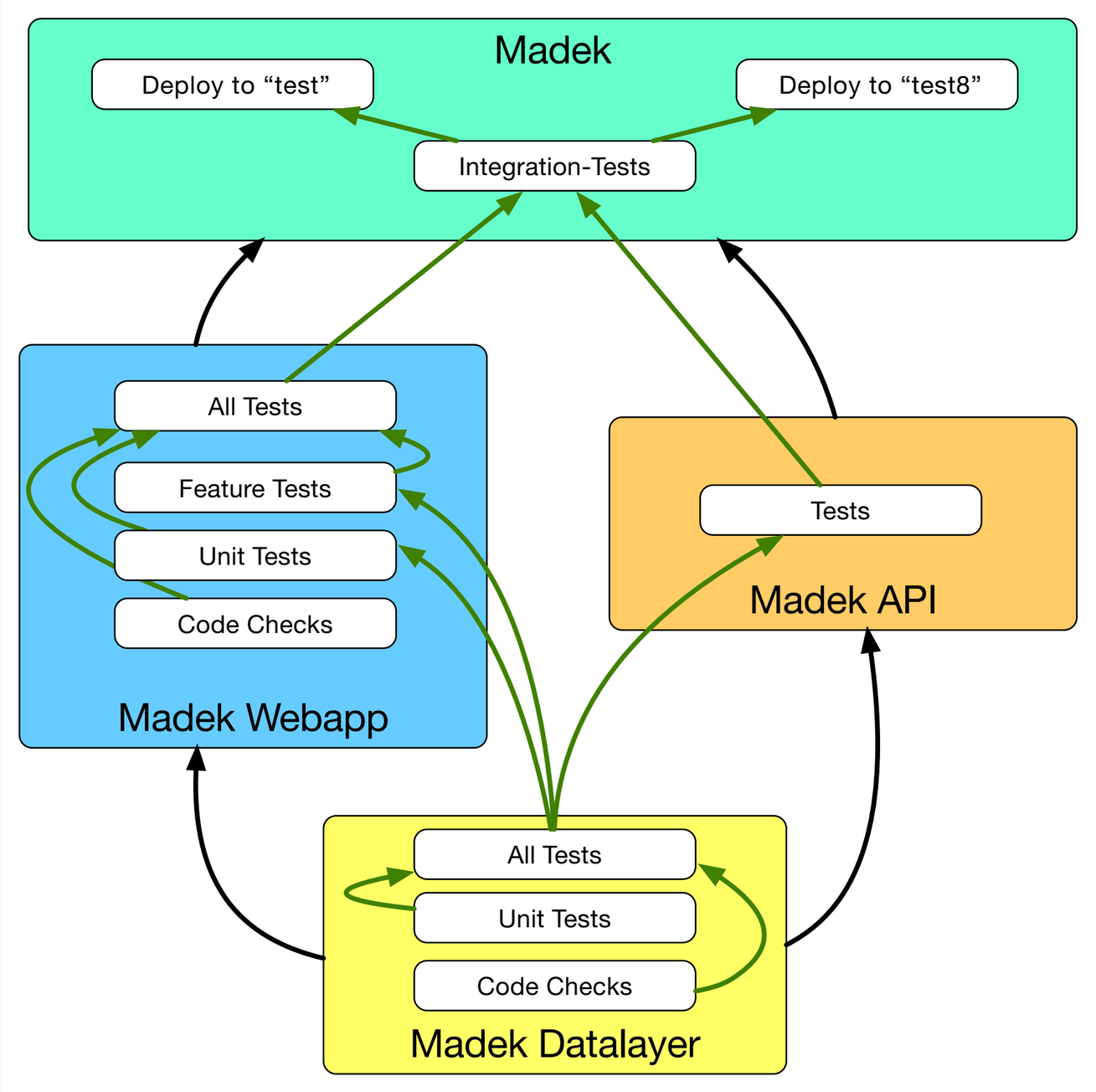
SCRIPT DEPENDENCIES

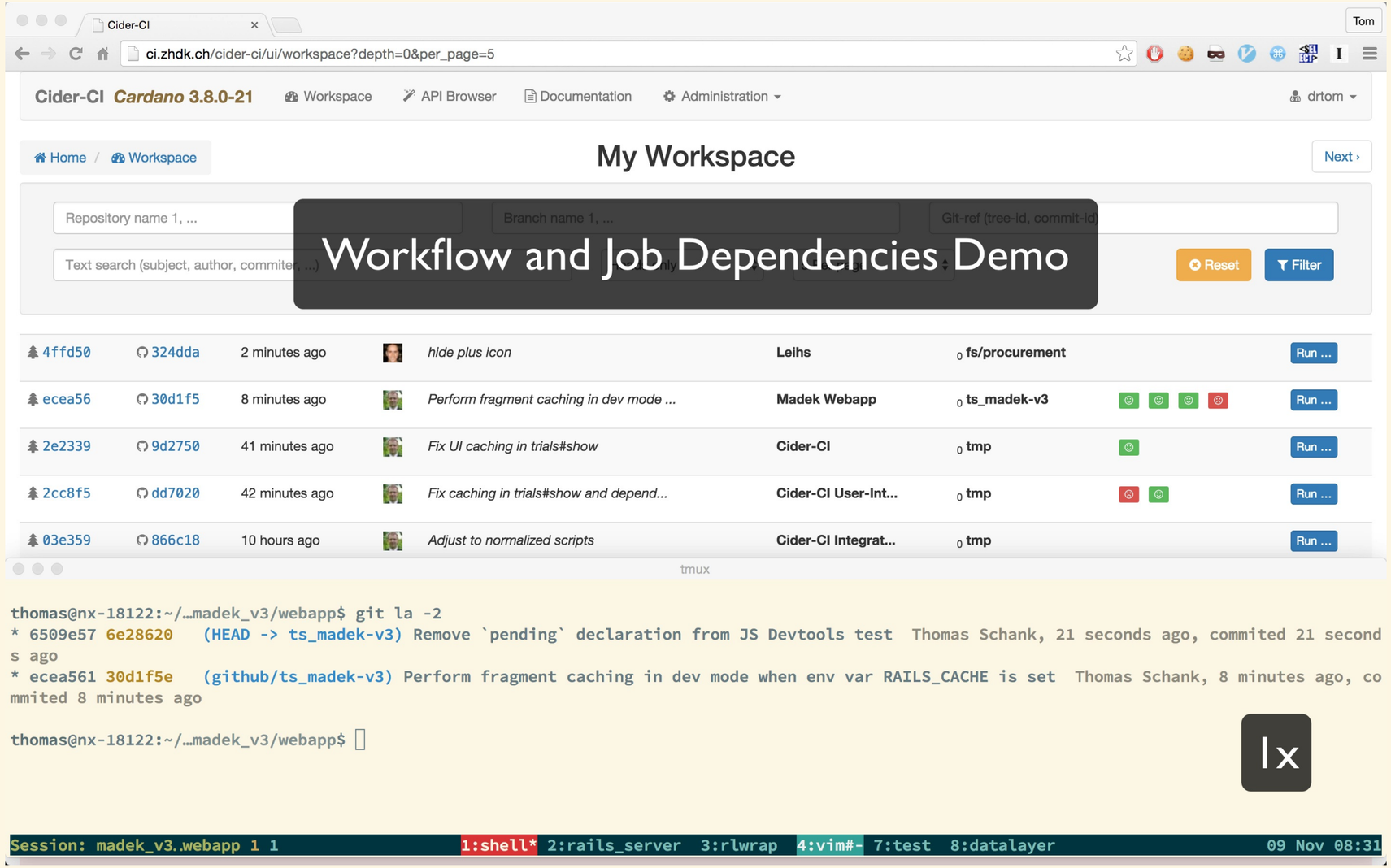


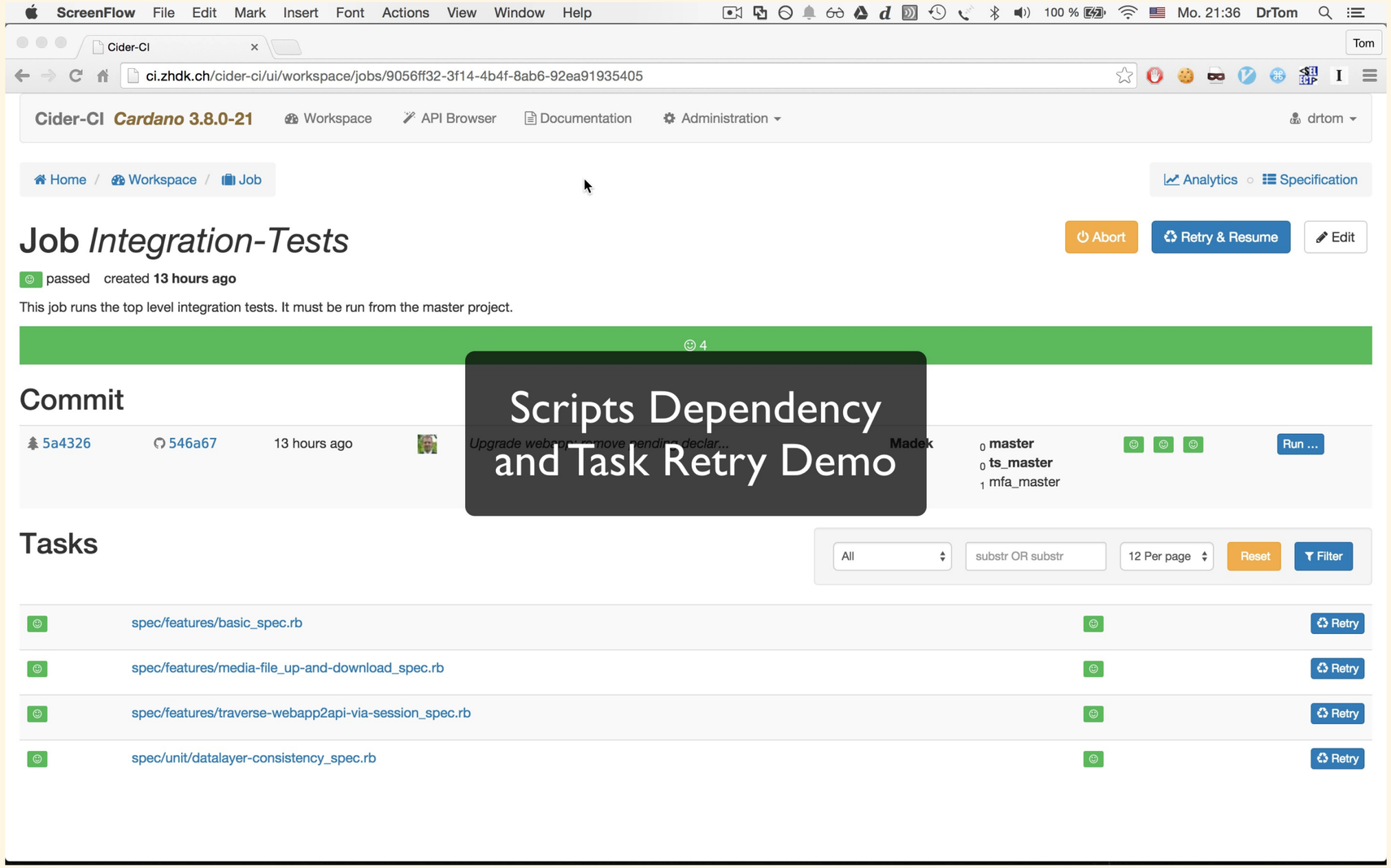
- traditional CI: one "build" ⇔ one script
- more modern: one main script + before and after "hooks"
- Cider-CI: **scripts with dependencies**

4. DEMOS




Madek
Deploy to “test”
Deploy to “test8”
Integration-Tests
All Tests
Feature Tests
Unit Tests
Code Checks
Madek Webapp
Tests
Madek API
All Tests
Unit Tests
Code Checks
Madek Datalayer


Cider-CI
Tom
ci.zhdk.ch/cider-ci/ui/workspace?depth=0&per_page=5
Cider-CI Cardano 3.8.0-21
Workspace
API Browser
Documentation
Administration
drtom
Home / Workspace
My Workspace
Next ›
Repository name 1, ...
Branch name 1, ...
Git-ref (tree-id, commit-id)
Text search (subject, author, commiter, ...)
Reset
Filter
Workflow and Job Dependencies Demo
4ffd50 324dda 2 minutes ago hide plus icon Leihs 0 fs/procurement Run ...
ecea56 30d1f5 8 minutes ago Perform fragment caching in dev mode ... Madek Webapp 0 ts_madek-v3 Run ...
2e2339 9d2750 41 minutes ago Fix UI caching in trials#show Cider-CI 0 tmp Run ...
2cc8f5 dd7020 42 minutes ago Fix caching in trials#show and depend... Cider-CI User-Int... 0 tmp Run ...
03e359 866c18 10 hours ago Adjust to normalized scripts Cider-CI Integrat... 0 tmp Run ...
tmux
thomas@nx-18122:~/…madek_v3/webapp$ git la -2
* 6509e57 6e28620 (HEAD -> ts_madek-v3) Remove `pending` declaration from JS Devtools test Thomas Schank, 21 seconds ago, commited 21 seconds ago
* ecea561 30d1f5e (github/ts_madek-v3) Perform fragment caching in dev mode when env var RAILS_CACHE is set Thomas Schank, 8 minutes ago, commited 8 minutes ago
thomas@nx-18122:~/…madek_v3/webapp$
lx
Session: madek_v3.webapp 1 1
1:shell* 2:rails_server 3:rlwrap 4:vim#- 7:test 8:datalayer
09 Nov 08:31


ScreenFlow File Edit Mark Insert Font Actions View Window Help
Mo. 21:36 DrTom
Cider-CI
Tom
ci.zhdk.ch/cider-ci/ui/workspace/jobs/9056ff32-3f14-4b4f-8ab6-92ea91935405
Cider-CI Cardano 3.8.0-21
Workspace
API Browser
Documentation
Administration
drtom
Home / Workspace / Job
Analytics
Specification
Job Integration-Tests
Abort
Retry & Resume
Edit
passed created 13 hours ago
This job runs the top level integration tests. It must be run from the master project.
4
Scripts Dependency and Task Retry Demo
Commit
5a4326
546a67
13 hours ago
Madek
master
ts_master
mfa_master
Run ...
Tasks
All
substr OR substr
12 Per page
Reset
Filter
spec/features/basic_spec.rb
spec/features/media-file_up-and-download_spec.rb
spec/features/traverse-webapp2api-via-session_spec.rb
spec/unit/datalayer-consistency_spec.rb
Retry

5. ADDENDUM

MANAGING FALSE POSITIVES

Retrying randomized tests can hide problems!

"Generative Testing" e.g.

SOLUTION:

- reproducibility by initializing the pseudo random generator (we use the `tree_id` e.g.)
- statistics

GIT SCM AND GIT ONLY

- don't compromise
- can't support everything with reasonable effort

SECURITY & TRUST

- Cider-CI server itself **never runs any code** from projects
- **"blessed" executors** only accept trials for a particular project (repository)

MATCHING TRIALS TO EXECUTORS

- task specifies **required traits**, e.g: [`bash`, `ruby-2.2`]
- executors advertise **available traites**, e.g. [`bash`, `maven`, `postgresql`, `ruby-2.1`, `ruby-2.2`, …]

Cider-CI will determine a suitable executor.

DEPLOYMENT

- Ansible
- Cider-CI deploy project, SCM managed, reproducible

CIDER-CI IS AN EXPERT SYSTEM

it is about **making the hard possible**, and not not about making the simple easy*

- for professionals
- no compromises
- steep learning curve
- high rewards

→ swiss army knife for devops

*see "Simple Made Easy" by Rich Hickey

CONCLUSION

- A false negative outcome becomes likely with an increasing number of tests.
- The problem must be solved by retrying single tests.
- Consider to build your own pipeline.
- Try Cider-CI, open source, installs with two commands:

  `http://docs.cider-ci.info/introduction/quick-start/`

Thank You!