BOOSTING TEST COVERAGE

FOR MICROSERVICES

GoLab

Florence

20.11.2023

Laura Vuorenoja
Technology Strategist
OP Lab

@lauravuo
@vuorenoja
@lauravuo@fosstodon.org
/lauravuorenoja

AGENDA

findy-network.github.io

JANE OF ALL TRADES

@lauravuo

@vuorenoja

@lauravuo@fosstodon.org

/lauravuorenoja

Laura Vuorenoja
Technology Strategist
OP Lab
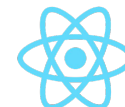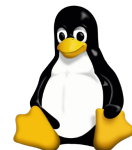
2005   **2010**   **2015**   **2020**

VERIFIABLE DATA EXCHANGE

CRYPTOGRAPHY AS AN ENABLER FOR DECENTRALIZED IDENTITY

findy-network.github.io

HOLDER

controls

HOLDER'S AGENT

Issue Credential Protocol

DIDComm

Present Proof Protocol

DIDComm

ISSUER

controls

VERIFIER

controls

read

write

read

VDR

ISSUER'S AGENT

VERIFIER'S AGENT

trust
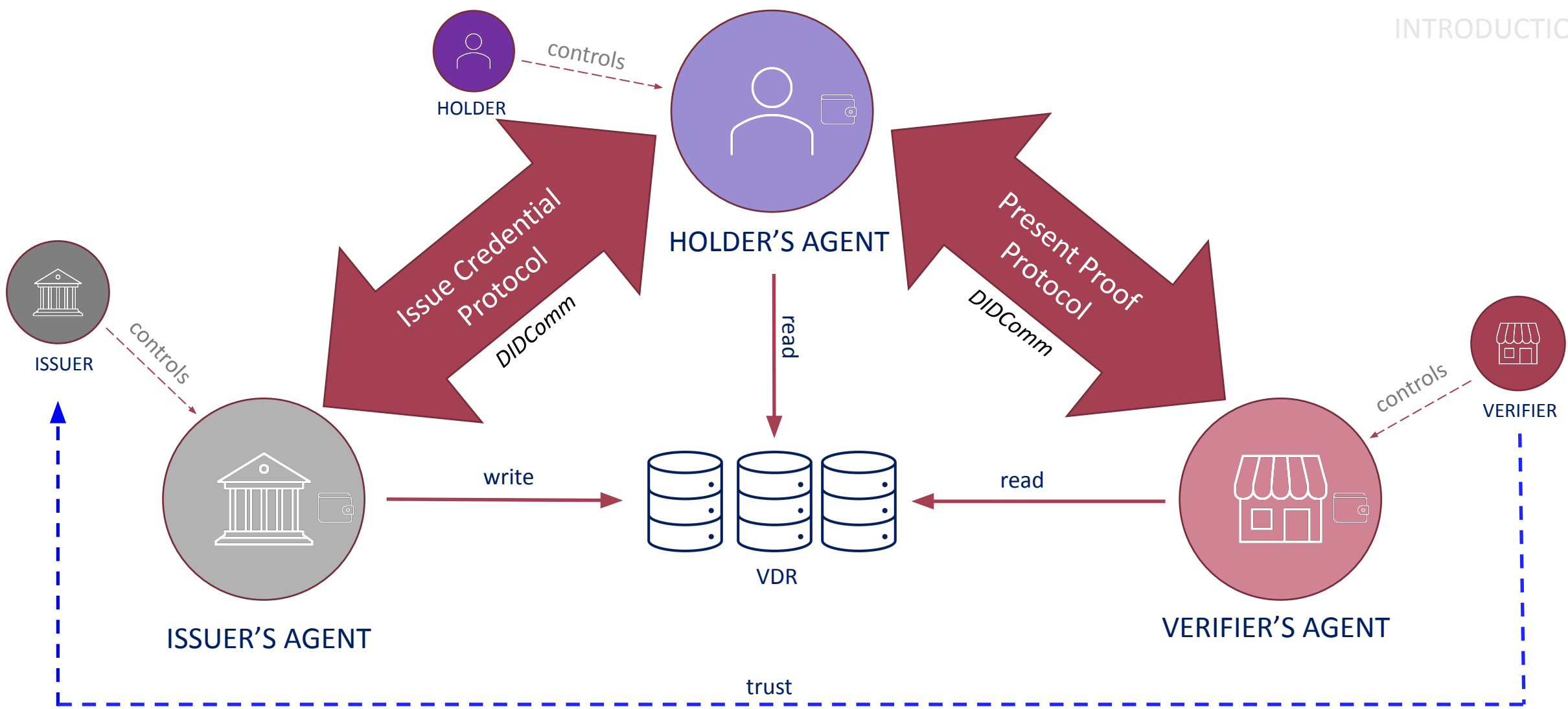
DECENTRALIZED DOMAIN

increases complexity

FINDY AGENCY
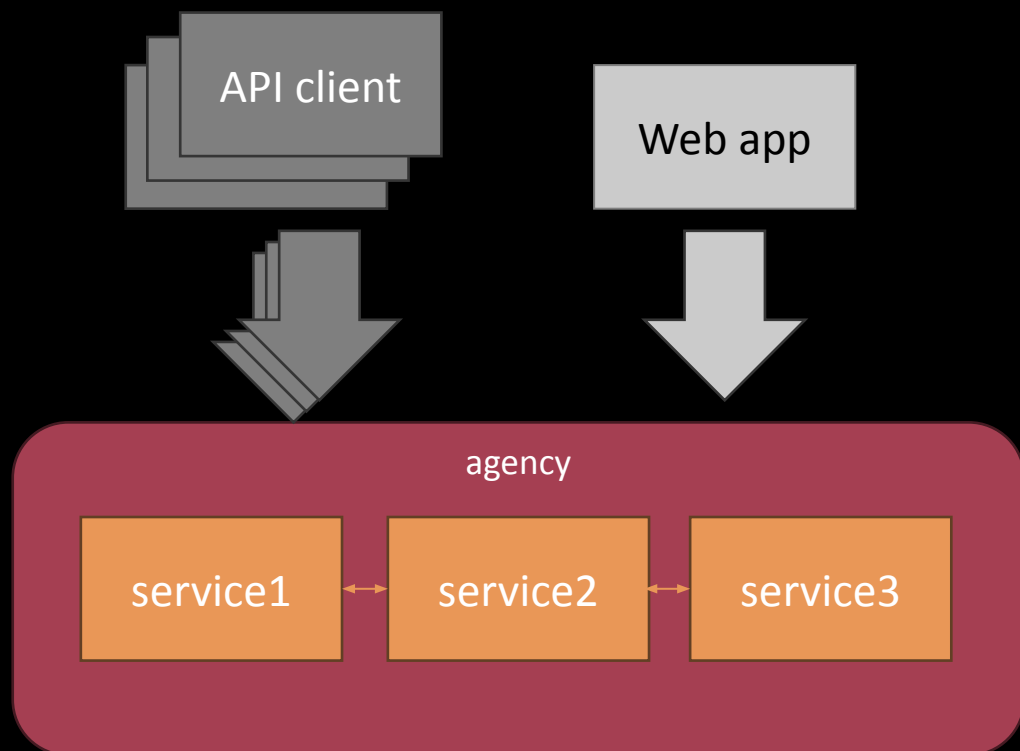
microservice architecture
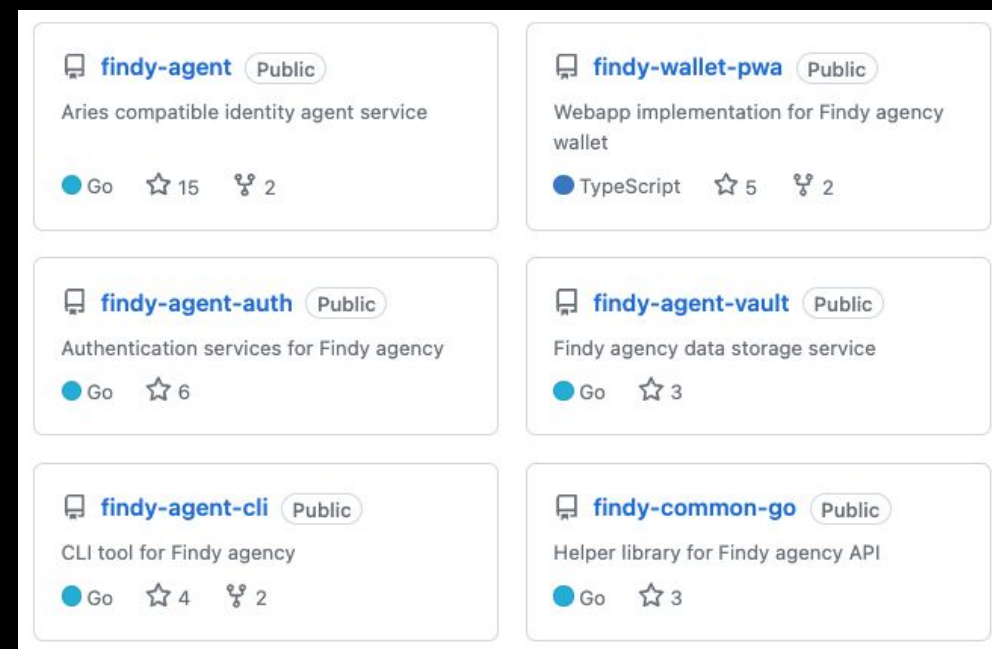
multiple repositories

findy-network.github.io

AUTOMATED TESTING = TESTS RUN IN CI

programmatic tests are just another form of manual testing if **NOT** run in CI

AUTOMATED TESTING

TIRELESS KNIGHT AGAINST REGRESSION

regression = previously working functionality or feature in an application starts malfunctioning or breaks after making changes or updates to the software

UNIT TESTS: IDEAL WORLD

```go
package abs

import "testing"

func TestAbs(t *testing.T) {
    got := Abs(-1)
    if got != 1 {
        t.Errorf("Abs(-1) = %d; want 1", got)
    }
}
```

UNIT TESTS: REALITY

network

System Under Test

file system

time

that complex
3rd party API

database

MOCKS

STUBS

FAKES

SPIES

…
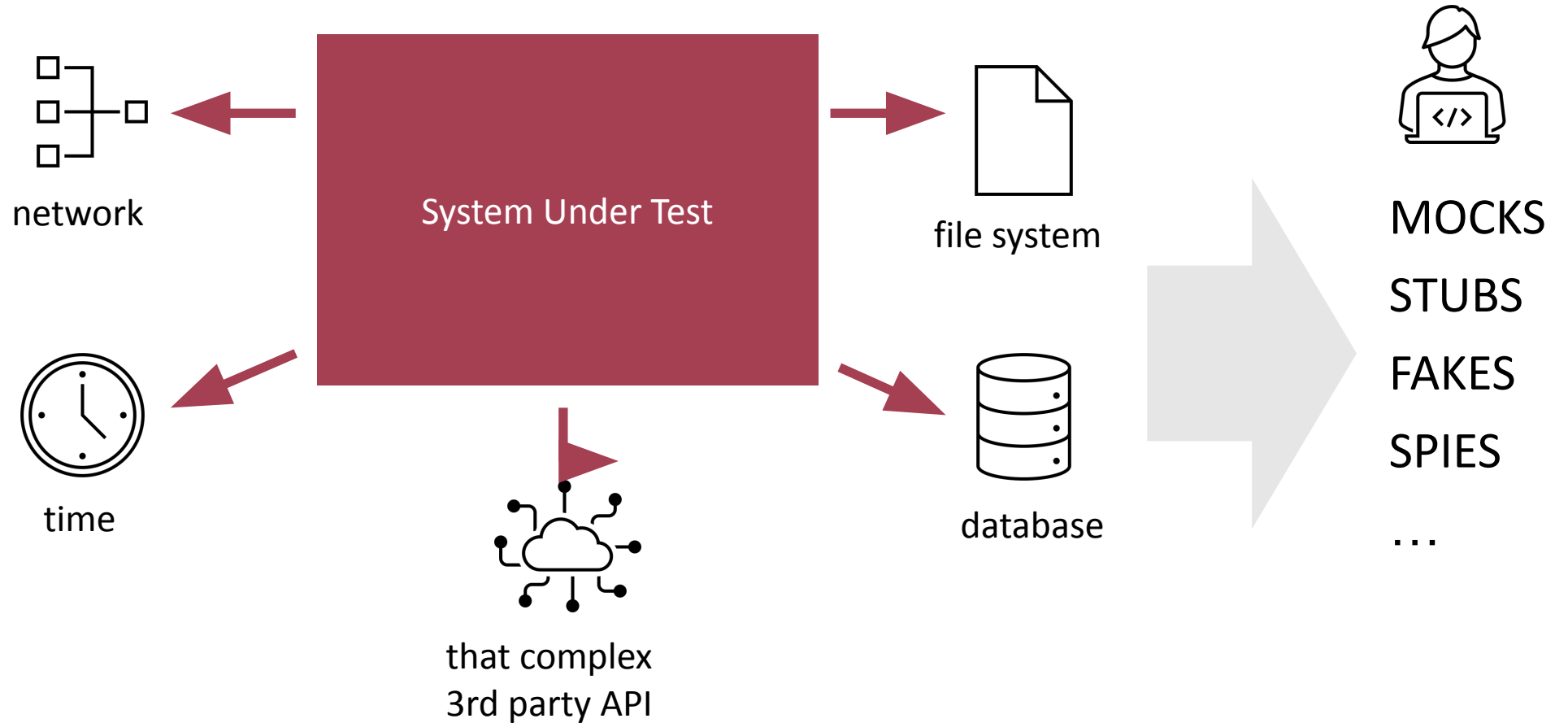
APPLICATION (INTEGRATION/E2E) TESTS

SOFTWARE RUN WITH ITS EXTERNAL DEPENDENCIES, WITHOUT UNIT TESTING TOOLING

FINDY AGENCY APPLICATION TESTS

gh-worklow.yml

docker-compose.yml

BROWSER

CLI

INTEROP
TEST APP

service1

service2

service3

service4

db1

db2

db3

TEST COVERAGE

findy-agent / protocol / connection / **connection_protocol.go**   **Uncovered** ⚠ | **Partial** ! ⋮ Covered

```go
102    func startConnectionProtocol(ca comm.Receiver, task comm.Task) {
103            defer err2.Catch(func(err error) {                                        !
104                    glog.Error("ERROR in starting connection protocol:", err)          ⚠
105            })                                                                          ⚠
106
107            deTask, ok := task.(*taskDIDExchange)
108            assert.That(ok)
109
110            invMsg := aries.MsgCreator.Create(didcomm.MsgInit{
111                    Type: deTask.Invitation.Type(),
112                    AID:  deTask.Invitation.ID()}).(didexchange.PwMsg)
113
```
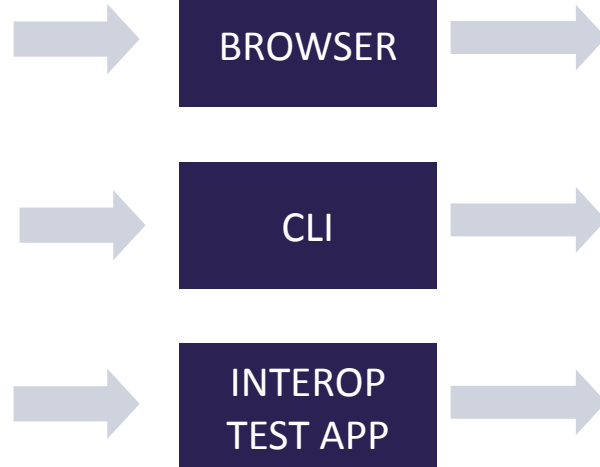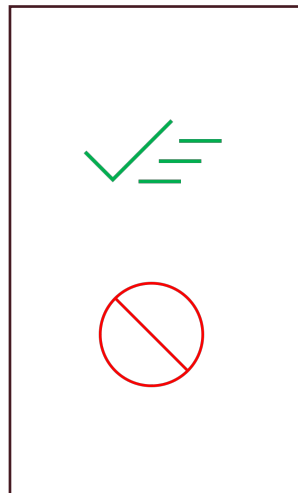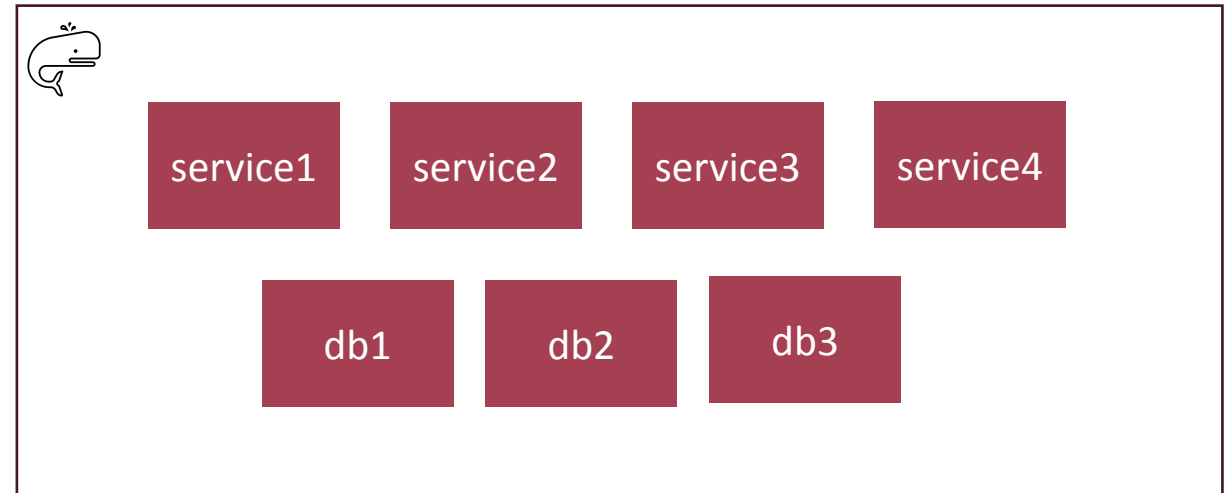
13

PURPOSE OF COVERAGE MEASUREMENT

IS NOT TO BE THE BONUS CRITERIA FOR YOUR BOSS

it determines how well the tests cover
the various aspects of the software

AUTOMATED MONITORING

- new PR
- tests in CI
- result file upload
- analysis
- PR check OK/NOK
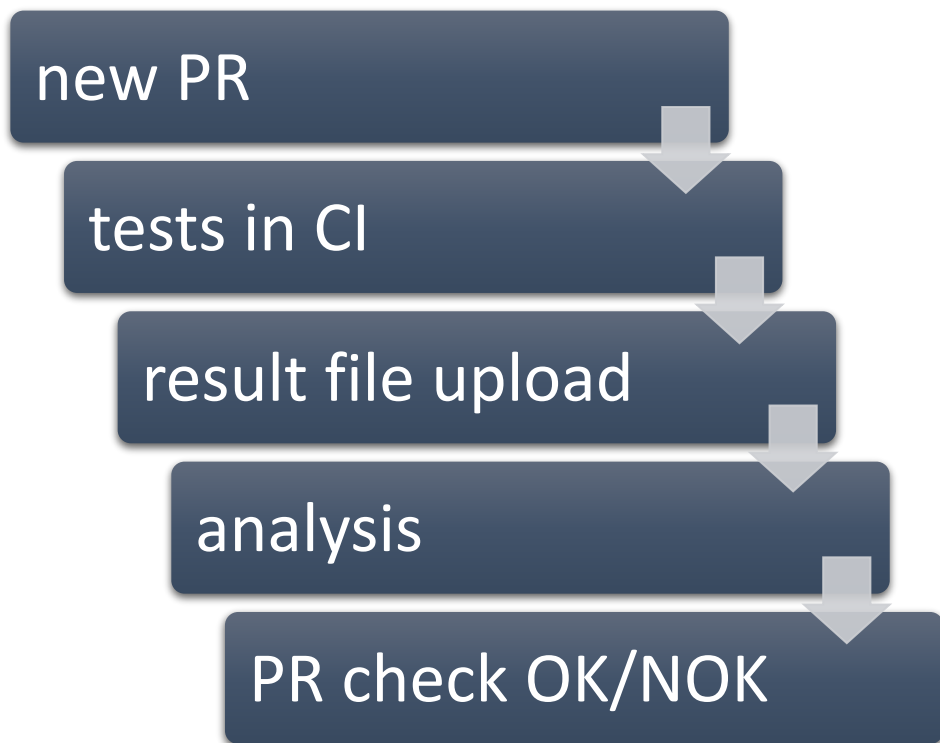
codecov (bot) commented 9 hours ago

Codecov Report

Patch and project coverage have no change.

Comparison is base (b91f7d3) 87.30% compared to head (5d4fe0c) 87.30%

▼ Additional details and impacted files

```
@@              Coverage Diff              @@
##              dev     #204   +/-   ##
===============================================
  Coverage   87.30%   87.30%
===============================================
  Files          57       57
  Lines        4466     4466
===============================================
  Hits         3899     3899
  Misses        486      486
  Partials       81       81
```

🦏 View full report in Codecov by Sentry.
📣 Do you have feedback about the report comment? Let us know in this issue.

☺

findy-network.github.io

ADD SOME FLAGS TO UNIT TESTS AND ENJOY THE RIDE

but what about the application tests?

SOLUTION: GO COVER

Cover

Go 1.20 supports collecting code coverage profiles for programs (applications and integration tests), as opposed to just unit tests.

HOW

```
# build with -cover flag
go build -cover -o ./program

# define folder for result files
export GOCOVERDIR="./coverage"

# run binary
./program

# convert and analyze result files
go tool covdata textfmt -i=$GOCOVERDIR -o coverage.txt
```

SERVICE DOCKERFILE

🐳 Dockerfile > …

```
 1    FROM golang:1.20-alpine3.17        ←  build stage
 2
 3    ARG GOBUILD_ARGS=""
 4
 5    WORKDIR /work
 6
 7    COPY go.* ./
 8    RUN go mod download
 9
10    COPY . ./
11
12    RUN VERSION=$(cat ./VERSION) && \
13      go build ${GOBUILD_ARGS} \
14      -ldflags "-X 'github.com/findy-network/findy-agent-vault/utils.Version=$VERSION'"\
15      -o /go/bin/findy-agent-vault
16
17    FROM alpine:3.17        ←  run stage
18
19    LABEL org.opencontainers.image.source https://github.com/findy-network/findy-agent-vault
20
21    EXPOSE 8085
22
23    # used when running instrumented binary
24    ENV GOCOVERDIR /coverage
25
26    COPY --from=0 /work/db/migrations /db/migrations
27    COPY --from=0 /go/bin/findy-agent-vault /findy-agent-vault
28
29    ENTRYPOINT ["/findy-agent-vault"]
```

APPLICATION TESTING STEPS

Build service image with instrumented binary → Launch test environment with docker-compose → Run test set → Convert and send coverage files

CONFIG FOR SERVICE UNDER TEST

docker-compose.yml

```yaml
vault:
  #image: ghcr.io/findy-network/findy-agent-vault:latest
  build:
    context: .
    dockerfile: Dockerfile
    args:
      GOBUILD_ARGS: -cover
  volumes:
    - ./cert:/grpc
    - .data/vault/coverage:/coverage
```

findy-network.github.io

AFTER TEST RUN: CONVERT AND STORE DATA

```yaml
e2e:
  runs-on: ubuntu-latest
  steps:
    - name: test e2e flow
      uses: findy-network/e2e-test-action@master
      with:
        service: "vault"
    - uses: actions/setup-go@v3
      with:
        go-version-file: "./go.mod"
    - name: convert coverage to txt
      run: go tool covdata textfmt -i=coverage -o coverage-e2e.txt
    - name: store coverage file
      uses: actions/upload-artifact@v3
      with:
        name: ${{ github.sha }}-coverage-e2e.txt
        path: ./coverage-e2e.txt
        retention-days: 1
```

SEND DATA FOR ANALYSIS

Triggered via push 6 hours ago

lauravuo-techlab pushed ◦— b375098 dev

Status
Success

Total duration
4m 35s

Artifacts
2

test.yml
on: push

✓ test   1m 20s   ✓ upload-coverage   6s

✓ e2e   2m 37s

Artifacts
Produced during runtime

Name	Size
b37509805fd020b1ca384c76b4893665d0aad616-coverage-e2e.txt	470 KB
b37509805fd020b1ca384c76b4893665d0aad616-coverage.txt	1020 KB

findy-network.github.io

GREAT SUCCESS

PROJECT OVERVIEW

findy-network.github.io

REMEMBER!
SHUTDOWN GRACEFULLY

COVERAGE DATA IS LOST IF PROGRAM DOES NOT EXIT "NORMALLY"

servers need to handle the SIGTERM signal
-> ensure that Docker container passes
the signal to the server process

TAKEAWAYS

USE KNIGHTS TO FIGHT REGRESSION

COMPLEMENT UNIT TESTS WITH APPLICATION TESTS

WRITE OPEN SOURCE

findy-network.github.io

THANK YOU!

QUESTIONS?

@lauravuo

@vuorenoja

@lauravuo@fosstodon.org

/lauravuorenoja

Laura Vuorenoja
Technology Strategist
OP Lab
findy-network.github.io