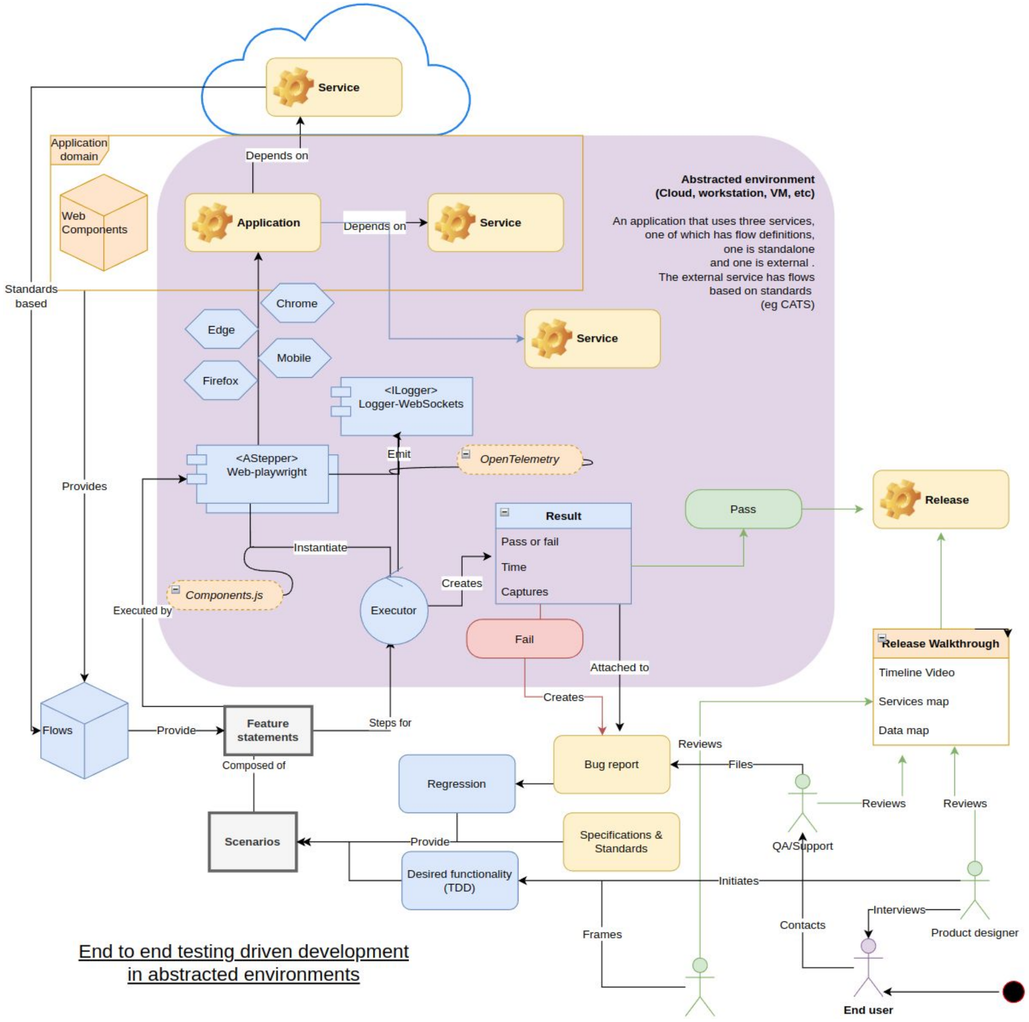
Haibun

End to end test driven development in abstracted environments



Abstracted environment (Cloud, workstation, VM, etc)

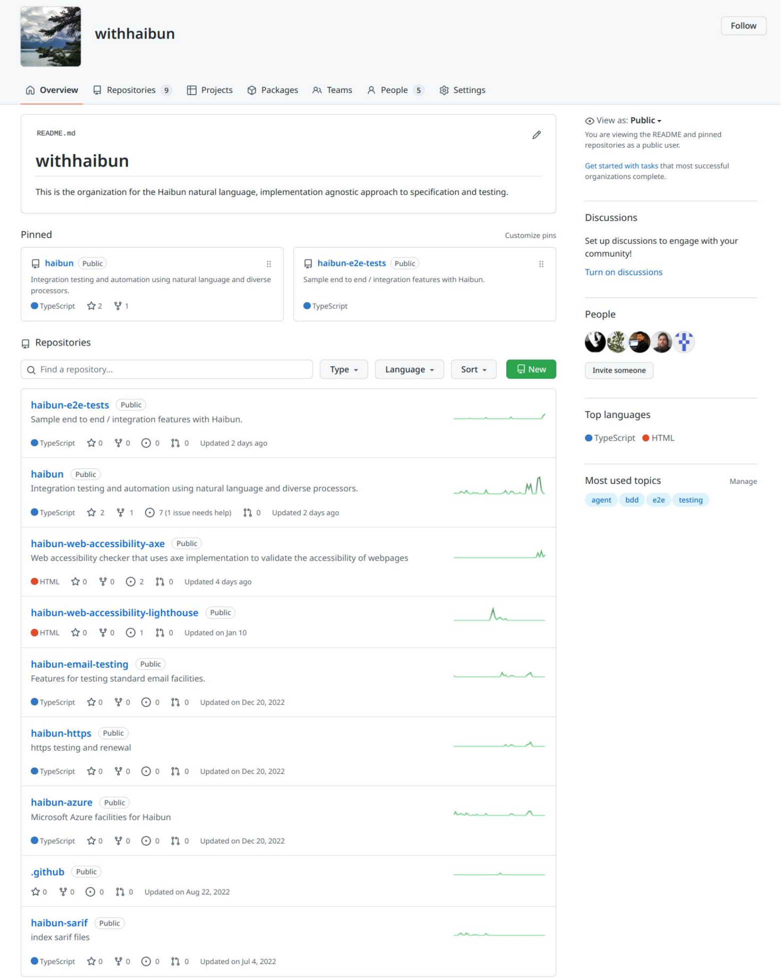
An application that uses three services,
one of which has flow definitions,
one is standalone
and one is external .
The external service has flows
based on standards
(eg CATS)

End to end testing driven development
in abstracted environments

Overview diagram

End to end testing

Test the functionality
and performance
of an application
under product-like circumstances
(services, data, user environments)
to replicate live settings.
The goal is to simulate
what a real user scenario looks like
from start to finish.



withhaibun

README.md

withhaibun

This is the organization for the Haibun natural language, implementation agnostic approach to specification and testing.

Pinned

haibun Public
Integration testing and automation using natural language and diverse processors.
● TypeScript ☆ 2 ⑂ 1

haibun-e2e-tests Public
Sample end to end / integration features with Haibun.
● TypeScript

Repositories

haibun-e2e-tests Public
Sample end to end / integration features with Haibun.
● TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated 2 days ago

haibun Public
Integration testing and automation using natural language and diverse processors.
● TypeScript ☆ 2 ⑂ 1 ⊙ 7 (1 issue needs help) ⇅ 0 Updated 2 days ago

haibun-web-accessibility-axe Public
Web accessibility checker that uses axe implementation to validate the accessibility of webpages
● HTML ☆ 0 ⑂ 0 ⊙ 2 ⇅ 0 Updated 4 days ago

haibun-web-accessibility-lighthouse Public
● HTML ☆ 0 ⑂ 0 ⊙ 1 ⇅ 0 Updated on Jan 10

haibun-email-testing Public
Features for testing standard email facilities.
● TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

haibun-https Public
https testing and renewal
● TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

haibun-azure Public
Microsoft Azure facilities for Haibun
● TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

.github Public
☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Aug 22, 2022

haibun-sarif Public
index sarif files
● TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Jul 4, 2022

View as: **Public**
You are viewing the README and pinned repositories as a public user.

Get started with tasks that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

Turn on discussions

People

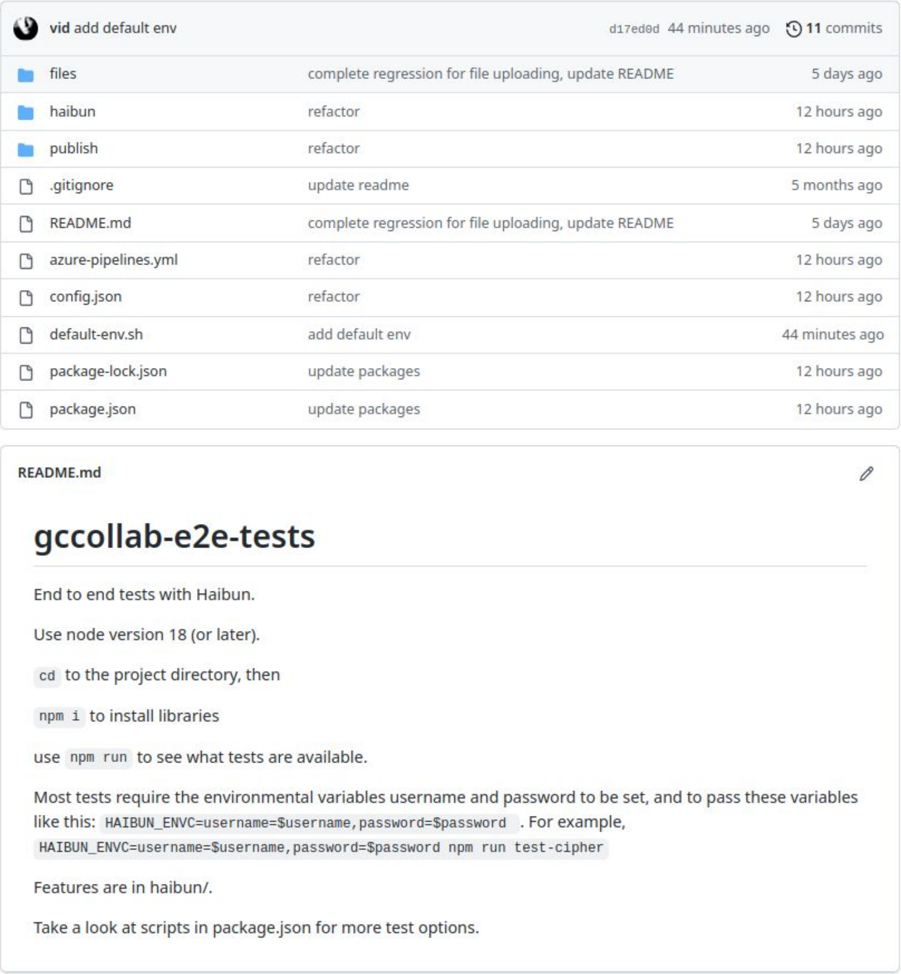
Top languages

● TypeScript ● HTML

Most used topics

agent bdd e2e testing

https://github.com/withhaibun



vid add default env d17ed0d 44 minutes ago 11 commits

files	complete regression for file uploading, update README	5 days ago
haibun	refactor	12 hours ago
publish	refactor	12 hours ago
.gitignore	update readme	5 months ago
README.md	complete regression for file uploading, update README	5 days ago
azure-pipelines.yml	refactor	12 hours ago
config.json	refactor	12 hours ago
default-env.sh	add default env	44 minutes ago
package-lock.json	update packages	12 hours ago
package.json	update packages	12 hours ago

README.md

gccollab-e2e-tests

End to end tests with Haibun.

Use node version 18 (or later).

`cd` to the project directory, then

`npm i` to install libraries

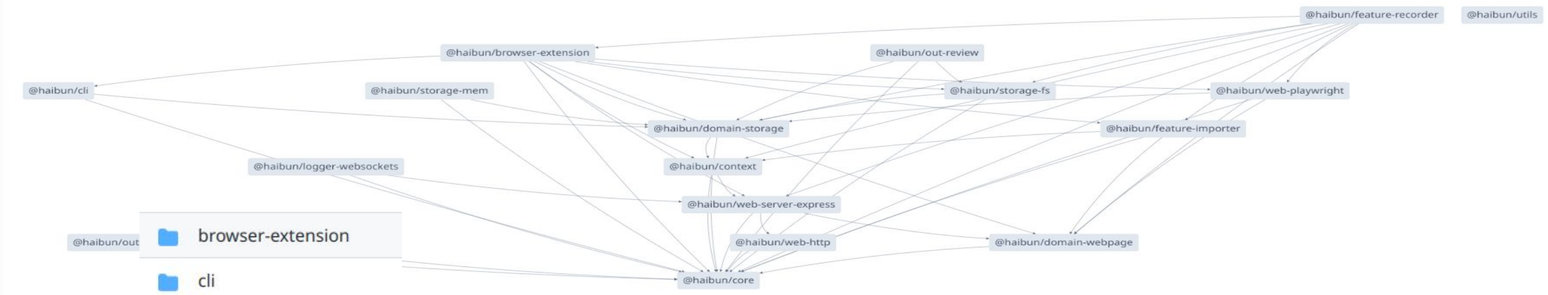
use `npm run` to see what tests are available.

Most tests require the environmental variables username and password to be set, and to pass these variables like this: `HAIBUN_ENVC=username=$username,password=$password` . For example, `HAIBUN_ENVC=username=$username,password=$password npm run test-cipher`

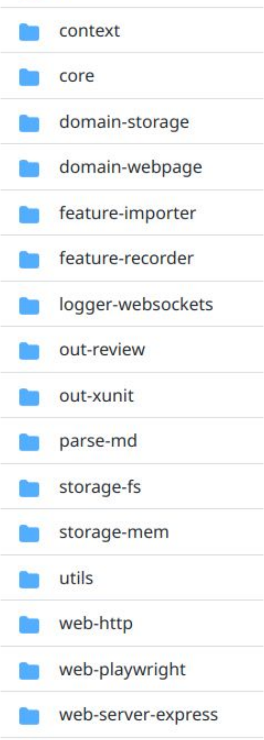
Features are in haibun/.

Take a look at scripts in package.json for more test options.

https://github.com/vid/gccollab-e2e-tests

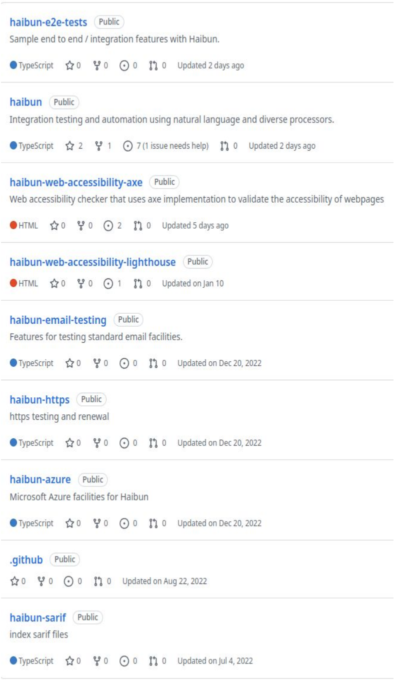


Module dependencies



- browser-extension
- cli
- context
- core
- domain-storage
- domain-webpage
- feature-importer
- feature-recorder
- logger-websockets
- out-review
- out-xunit
- parse-md
- storage-fs
- storage-mem
- utils
- web-http
- web-playwright
- web-server-express

Essential modules including abstractions; domains, storage implementations



haibun-e2e-tests Public
Sample end to end / integration features with Haibun.
TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated 2 days ago

haibun Public
Integration testing and automation using natural language and diverse processors.
TypeScript ☆ 2 ⑂ 1 ⊙ 7 (1 issue needs help) ⇅ 0 Updated 2 days ago

haibun-web-accessibility-axe Public
Web accessibility checker that uses axe implementation to validate the accessibility of webpages
HTML ☆ 0 ⑂ 0 ⊙ 2 ⇅ 0 Updated 5 days ago

haibun-web-accessibility-lighthouse Public
HTML ☆ 0 ⑂ 0 ⊙ 1 ⇅ 0 Updated on Jan 10

haibun-email-testing Public
Features for testing standard email facilities.
TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

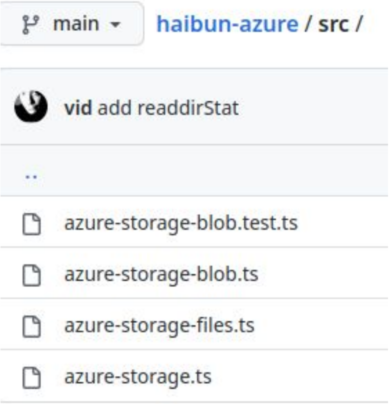
haibun-https Public
https testing and renewal
TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

haibun-azure Public
Microsoft Azure facilities for Haibun
TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Dec 20, 2022

.github Public
☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Aug 22, 2022

haibun-sarif Public
index sarif files
TypeScript ☆ 0 ⑂ 0 ⊙ 0 ⇅ 0 Updated on Jul 4, 2022

Module ecosystem



main

haibun-azure / src /

vid add readdirStat

..

azure-storage-blob.test.ts

azure-storage-blob.ts

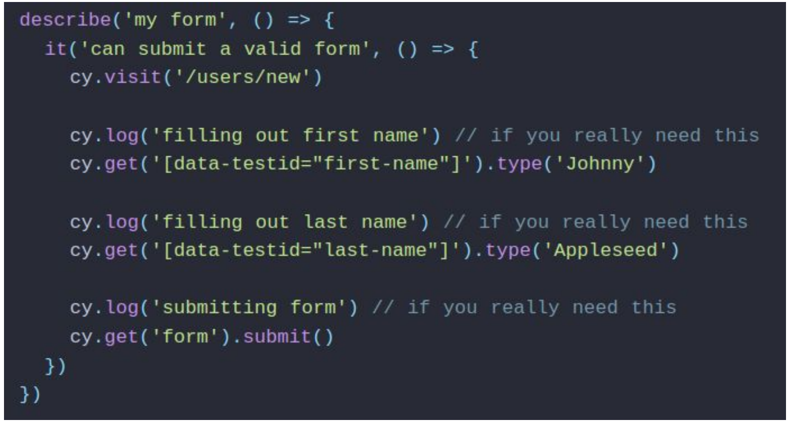
azure-storage-files.ts

azure-storage.ts

Storage implementation
and pipeline
for Azure
(files & blobs).
The pipeline uses the haibun-azure module,
so Azure volumes can store artifacts
with retention schemes
per test or deployment type.

```
trigger:
- test-e2e

jobs:
  - job: e2etests
    pool:
        vmImage: 'ubuntu-latest'
    container:
        image: mcr.microsoft.com/playwright:focal
        volumes: [e2e-reviews]
    steps:
        - task: Npm@1
          inputs:
            command: 'install'
          displayName: 'installing dependencies'
        - task: CmdLine@2
          displayName: 'Running login e2e tests'
          continueOnError: true
          inputs:
            script: |
              npm run
              HAIBUN_TITLE="Login e2e tests" HAIBUN_DEST=e2e HAIBUN_O_OUTREVIEWS_TRACE_STORAGE=StorageFS
HAIBUN_O_WEBPLAYWRIGHT_STORAGE=StorageFS HAIBUN_ENVC=${_ENVC} npm run test-pipeline-login

        - task: CmdLine@2
          displayName: 'Running regression e2e tests'
          continueOnError: true
          inputs:
            script: |
              HAIBUN_TITLE="Regression e2e tests" HAIBUN_DEST=regressions HAIBUN_O_OUTREVIEWS_TRACE_STORAGE=StorageFS
HAIBUN_O_WEBPLAYWRIGHT_STORAGE=StorageFS HAIBUN_ENVC=${_ENVC} npm run test-pipeline-regressions

        - task: CmdLine@2
          displayName: 'Generate sarif indexes'
          continueOnError: true
          inputs:
            script: |
              HAIBUN_DEST=sarif HAIBUN_O_AZURESTORAGEBLOB_DESTINATION='staticanalysis-sc'
HAIBUN_O_SARIF_TRACE_STORAGE=AzureStorageBlob HAIBUN_O_SARIF_INDEX_STORAGE=StorageFS
HAIBUN_O_AZURESTORAGEBLOB_ACCOUNT=${_O_AZURESTORAGEBLOB_ACCOUNT} HAIBUN_O_AZURESTORAGEBLOB_KEY=${_O_AZURESTORAGEBLOB_KEY}  npm
run test-pipeline-index-sarif

        - task: CmdLine@2
          displayName: 'Generate test result indexes'
          continueOnError: true
          inputs:
            script: |
              HAIBUN_O_OUTREVIEWS_TRACE_STORAGE=StorageFS HAIBUN_O_OUTREVIEWS_PUBLISH_STORAGE=AzureStorageBlob
HAIBUN_O_OUTREVIEWS_INDEX_STORAGE=StorageFS HAIBUN_O_OUTREVIEWS_REVIEWS_STORAGE=AzureStorageBlob
HAIBUN_O_AZURESTORAGEBLOB_DESTINATION=e2e-reviews-sc HAIBUN_O_AZURESTORAGEBLOB_ACCOUNT=${_O_AZURESTORAGEBLOB_ACCOUNT}
HAIBUN_O_AZURESTORAGEBLOB_KEY=${_O_AZURESTORAGEBLOB_KEY} HAIBUN_O_OUTREVIEWS_URI_ARGS=${_O_OUTREVIEWS_URI_ARGS} npm run
test-pipeline-indexes
```

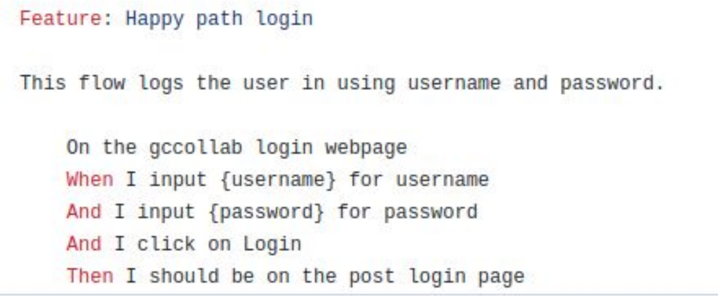
- Implemented with minimum of dependencies
- Highest fidelity for testing in each domain.
  Generate diagrams, videos, etc.
- Application tests only contain the minimum
  versioned high level language,
  that could be understood by anyone.
- Very high level of reusability between systems,
  reusable, versioned flows
  can be based on specifications,
  standards, components, applications
- Scenarios as descriptions of
  swappable systems
  (identity provider, cloud, services, interface)
- Very practical
  yet very wide potential scope
- Seamless integration
  in best practice system
- Based on very standard approaches
  (Playwright, Jest, Testing library),
  possible to “eject” basic standalone tests.



```
describe('my form', () => {
  it('can submit a valid form', () => {
    cy.visit('/users/new')

    cy.log('filling out first name') // if you really need this
    cy.get('[data-testid="first-name"]').type('Johnny')

    cy.log('filling out last name') // if you really need this
    cy.get('[data-testid="last-name"]').type('Appleseed')

    cy.log('submitting form') // if you really need this
    cy.get('form').submit()
  })
})
```

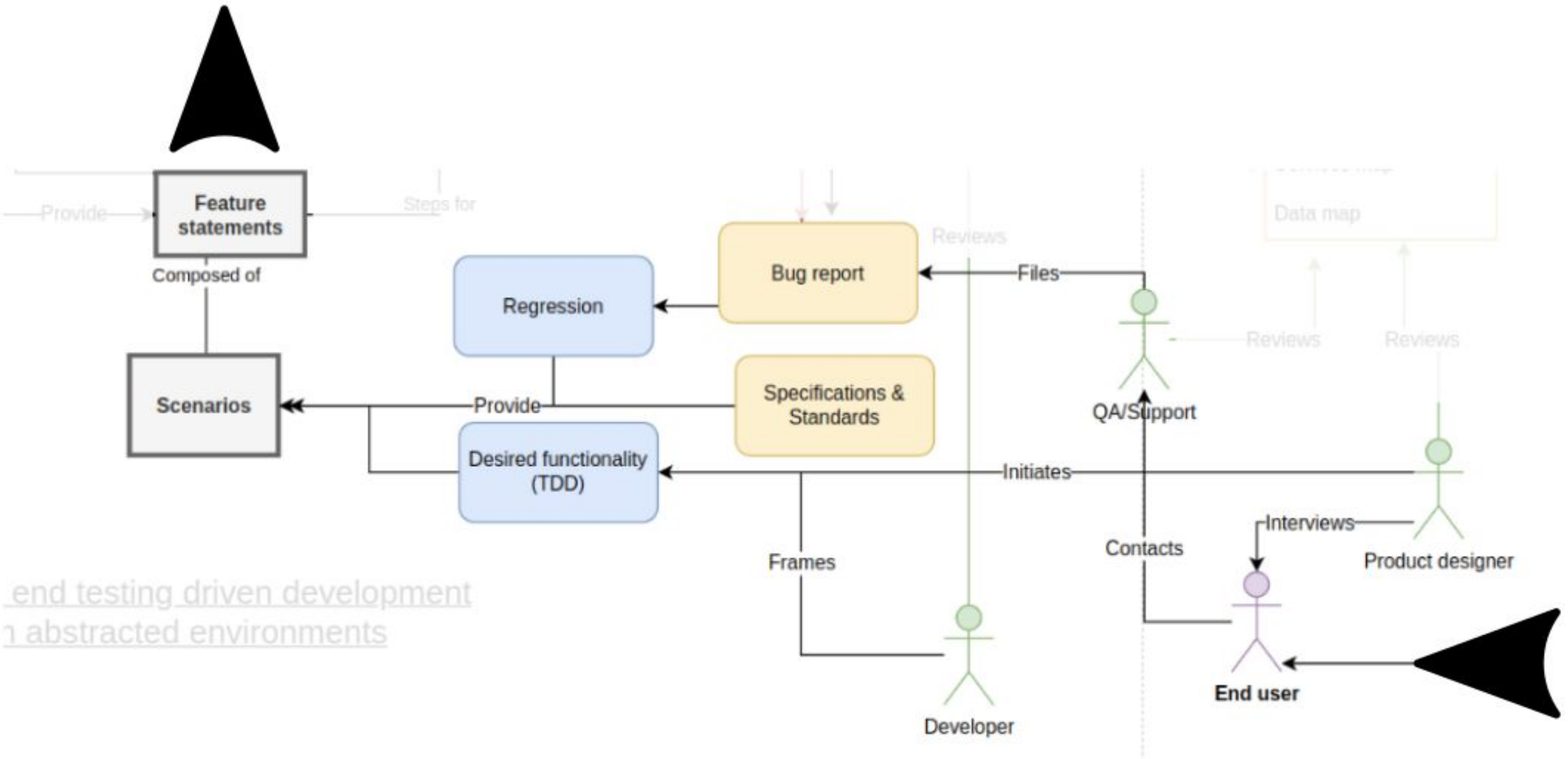
Cypress test

- Completely tied
  to specific implementation
  (not abstract)
- No consideration
  for connected services
  (not comprehensive)
- Gobbledygook

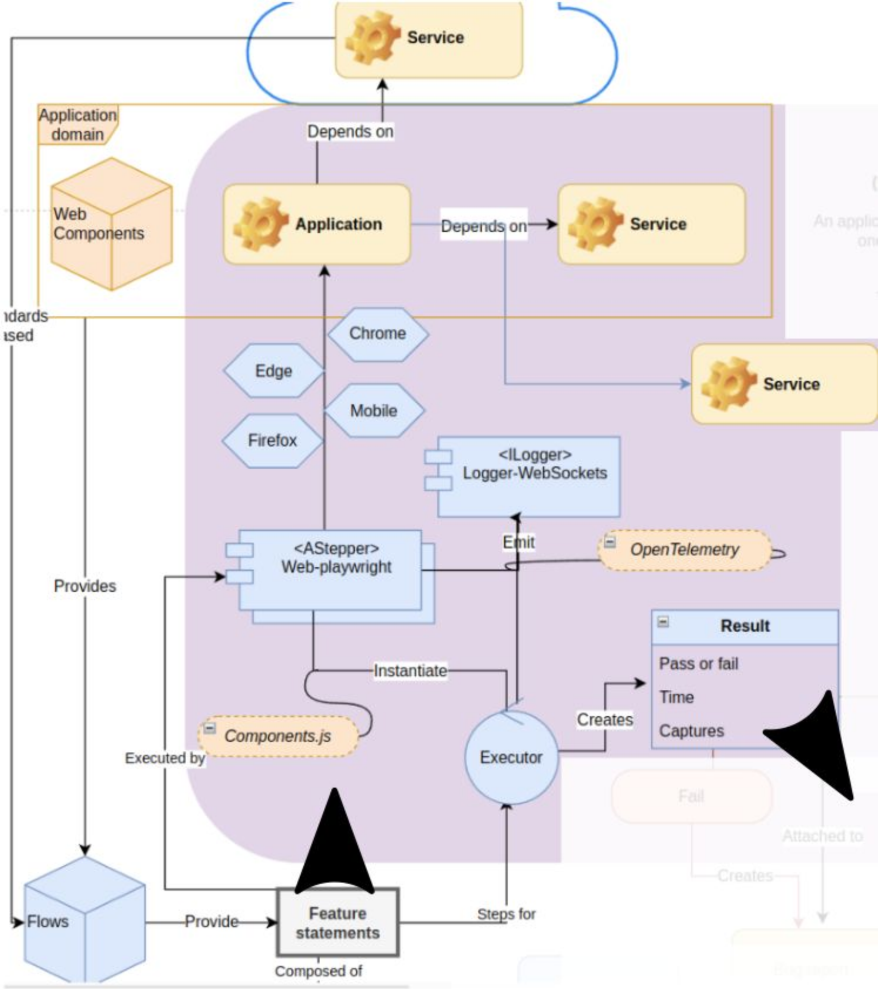


```
Feature: Happy path login

This flow logs the user in using username and password.

    On the gccollab login webpage
    When I input {username} for username
    And I input {password} for password
    And I click on Login
    Then I should be on the post login page
```

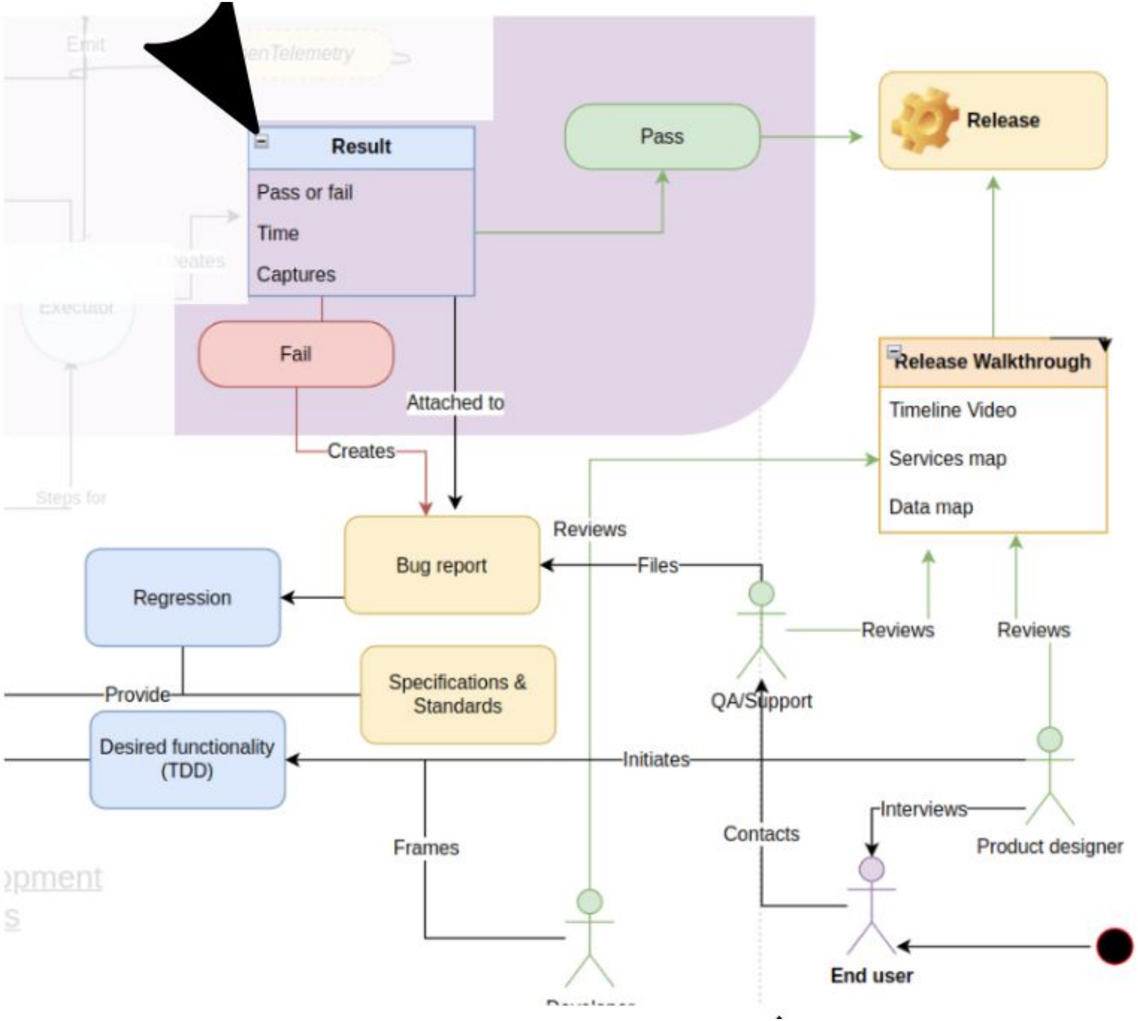
Haibun test



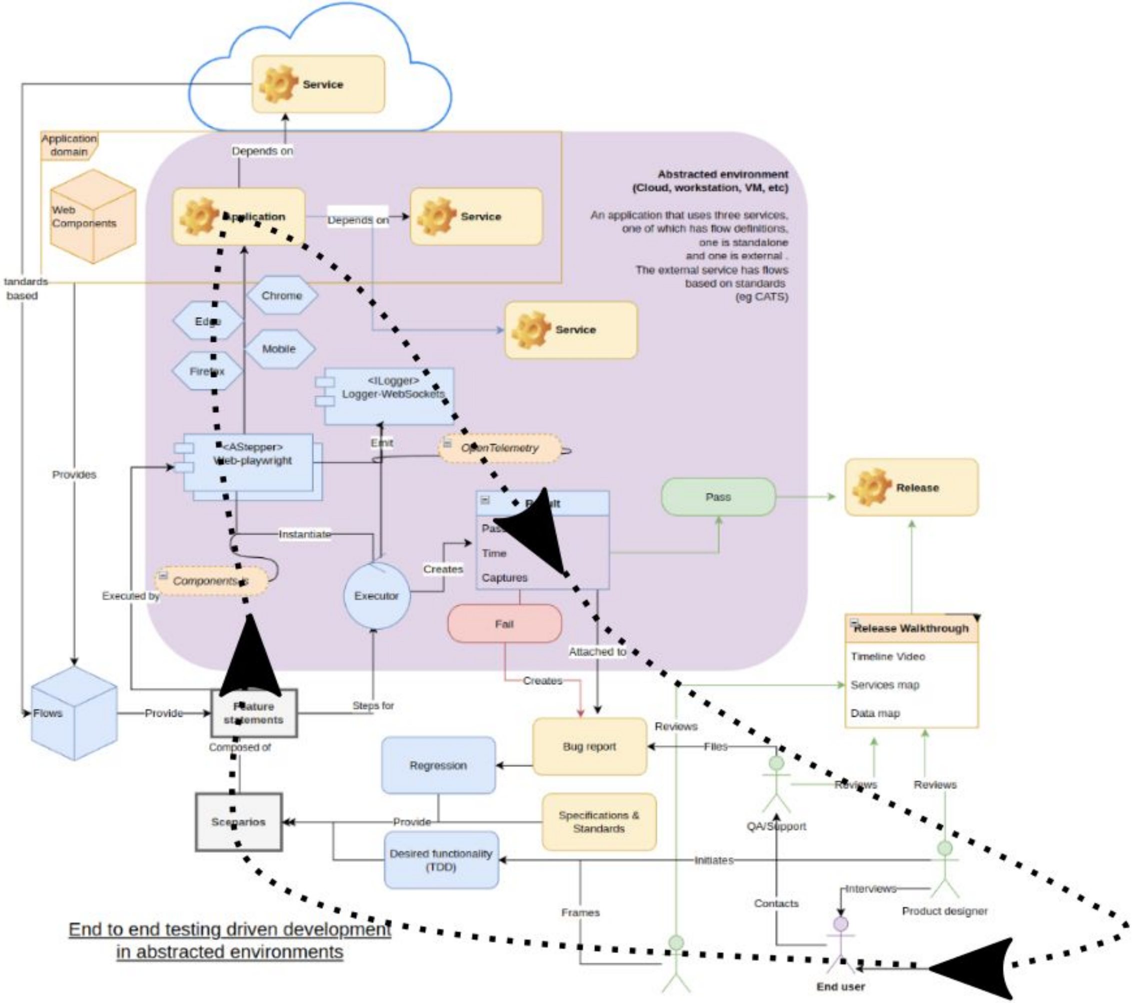
Workflow for specification   test development



Abstract evaluation environment
(cloud, workstation, VM, etc)
using features & flows



Results
feeding back into workflow
with deep,
interactive artifacts



End to end testing driven development
in abstracted environments

Agile
user centred
iterative
inclusive
built on best practices
heterogeneous
open source

From here

- ➡️Add mapping
- Cipher pipeline
- Use OpenTelemetry
  for messaging
- Improve reviews
- Alerts
  for unexpected service changes
- “View source” (with versioning)
  of any statement
- Adopt components.js
  for component model
- Abstractly automate
  connection with Azure DevOps
- Improve browser capture
- Development pipeline
- Developer
  and user ergonomics
  Editor helper
- High fidelity load testing
- Production pipeline
- Save, resume, branch sessions
- Automate security process
  and verification
- Specification as test
  as documentation
  as data map
  as application generator
  (OpenAPI for services & components)