

Test Automation Platform Architecture

[Summary](#)

[UI](#)

[Service](#)

[test-case-service](#)

 function: search plan

 function: parse case

 function: make plan

[test-device-service](#)

 function: list device model

[appcenter-service](#)

 job: start test run job

 job: finish test run job

[test-report-site](#)

 Sequence diagram: UI Diff Report & Allure report

 function: generate Allure Report (Embedded web project)

 function: generate UI difference Report

[Client](#)

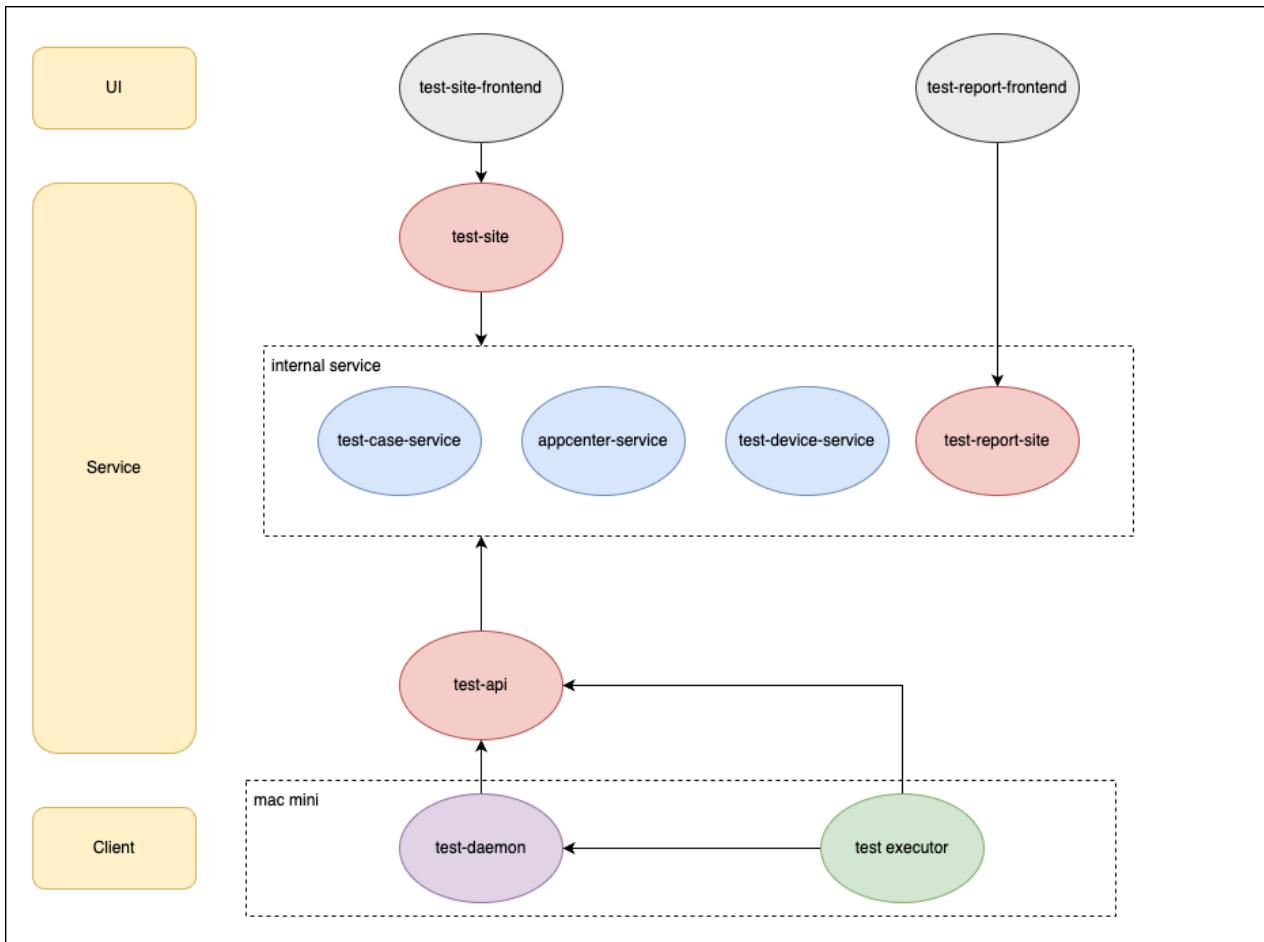
[test-daemon](#)

 job: upload device

 job: run executor

[How to use?](#)

Summary



UI

ⓘ Test Site
 QA: Wonder QA Automation Admin
 UAT: Wonder QA Automation Admin
 Test Report
 QA: <https://test-report.foodtruck-qa.com>
 UAT: <https://test-report.foodtruck-uat.com>
 identity V2: Identity Site V2 User Guide

Service

test-case-service

function: search plan

How to stat the duration?

1. plan.start_time & plan.end_time =

- a. plan.start_time = min (device_task.start_time)
- b. plan.end_time = max (device_task.end_time)
- c. device_task.start_time = min (executor_case.start_time)
- d. device_task.end_time = max (case_test_result.end_time)

2. Waiting minutes =

- a. Testing = plan.startTime - plan.createdTime
- b. Done = plan.startTime - plan.createdTime
- c. Pending -> Cancel = plan.cancelTime - plan.createdTime
- d. Testing -> Cancel = plan.startTime - plan.createdTime
- e. **Pending = now - plan.createTime**

3. Executing minutes =

- a. **Testing = now - plan.startTime**
- b. Done = plan.endTime- plan.startTime
- c. Pending -> Cancel = 0
- d. Testing -> Cancel = plan.cancelTime- plan.startTime
- e. Pending : 0

function: parse case

flow diagram

list case

```
dependencies
implementation("com.wonder:common-test-executor-framework:${commonTestExecutorFrameworkVersion}") {
    exclude group: "core.framework"
    exclude group: "org.slf4j"
}
implementation 'org.ow2.asm:asm:9.2'
```

workspace

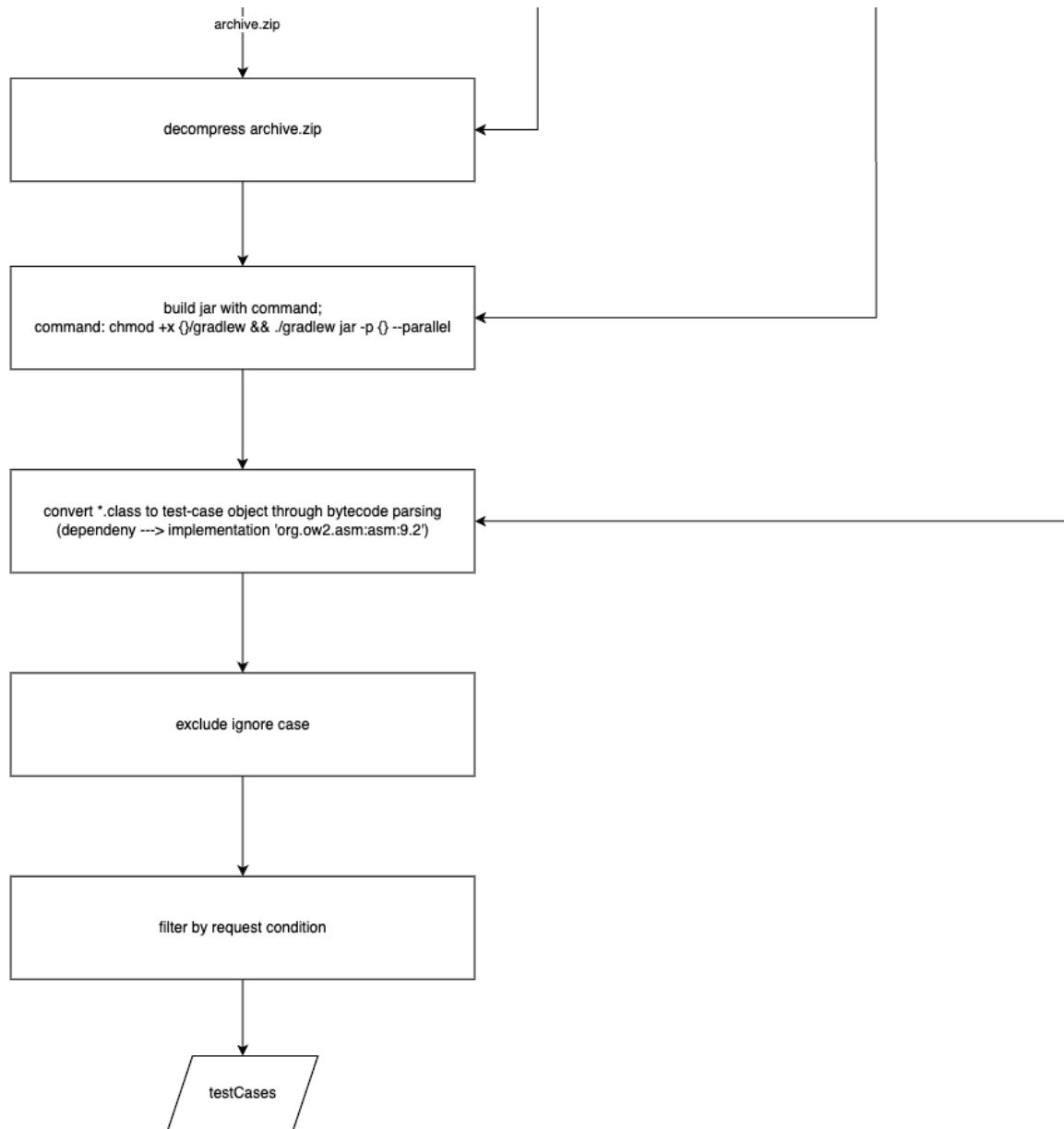
Directory Structure

```
test-case-workspace
└── {commitRevisionNumber}
    └── archive.zip
        └── {repo}
            └── build
                └── {case project name}
                    └── libs
                        └── {case project.jar}
```

Demo

```
test-case-workspace
└── b22b03f7e20c8350cf05a8c52de0dbdfa41f58
    └── archive.zip
        └── consumer-web-test-executor
            └── build
                └── consumer-web-test-interface
                    └── libs
                        └── com.wonder.consumer-web-test-interface-1.0.0.jar
```

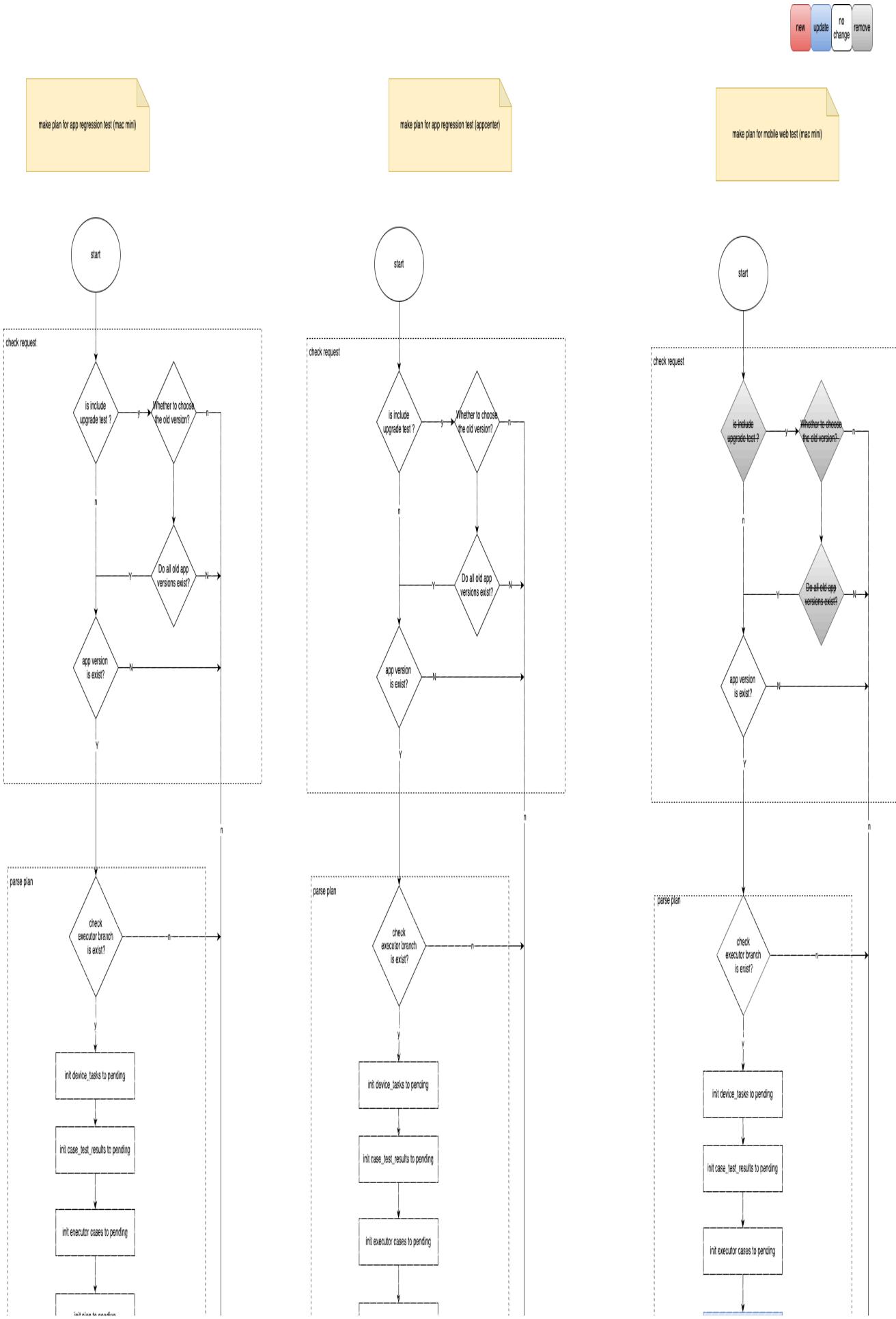


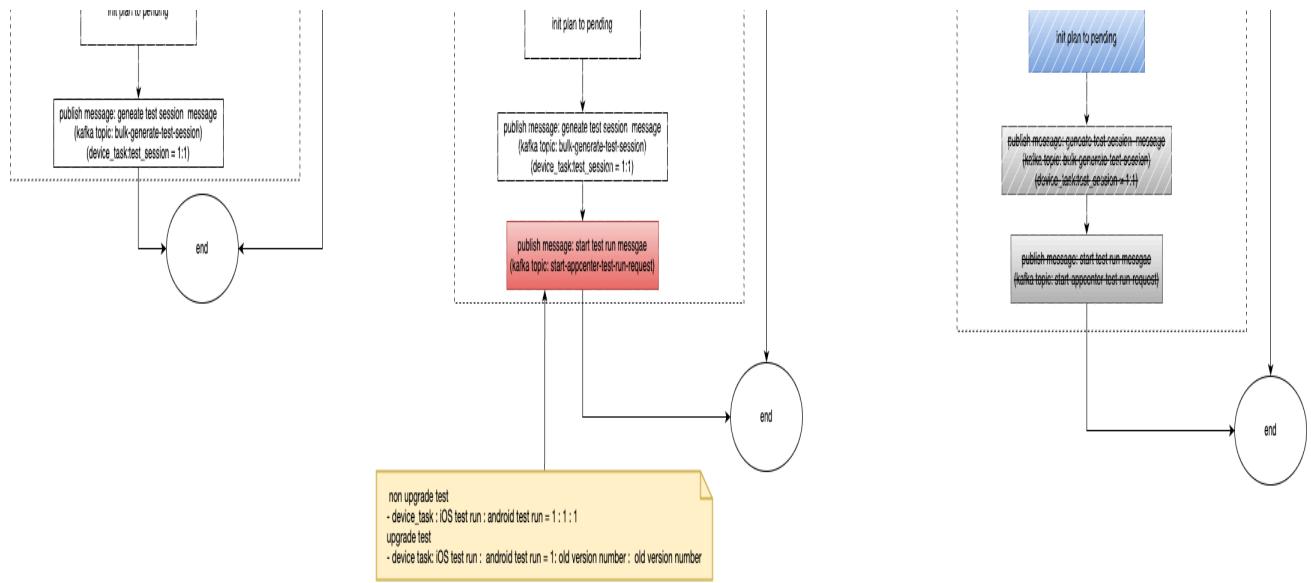


function: make plan

- make plan for app regression test (mac mini)
- make plan for app regression test (appcenter)
- make plan for mobile web test (mac mini)

flow diagram



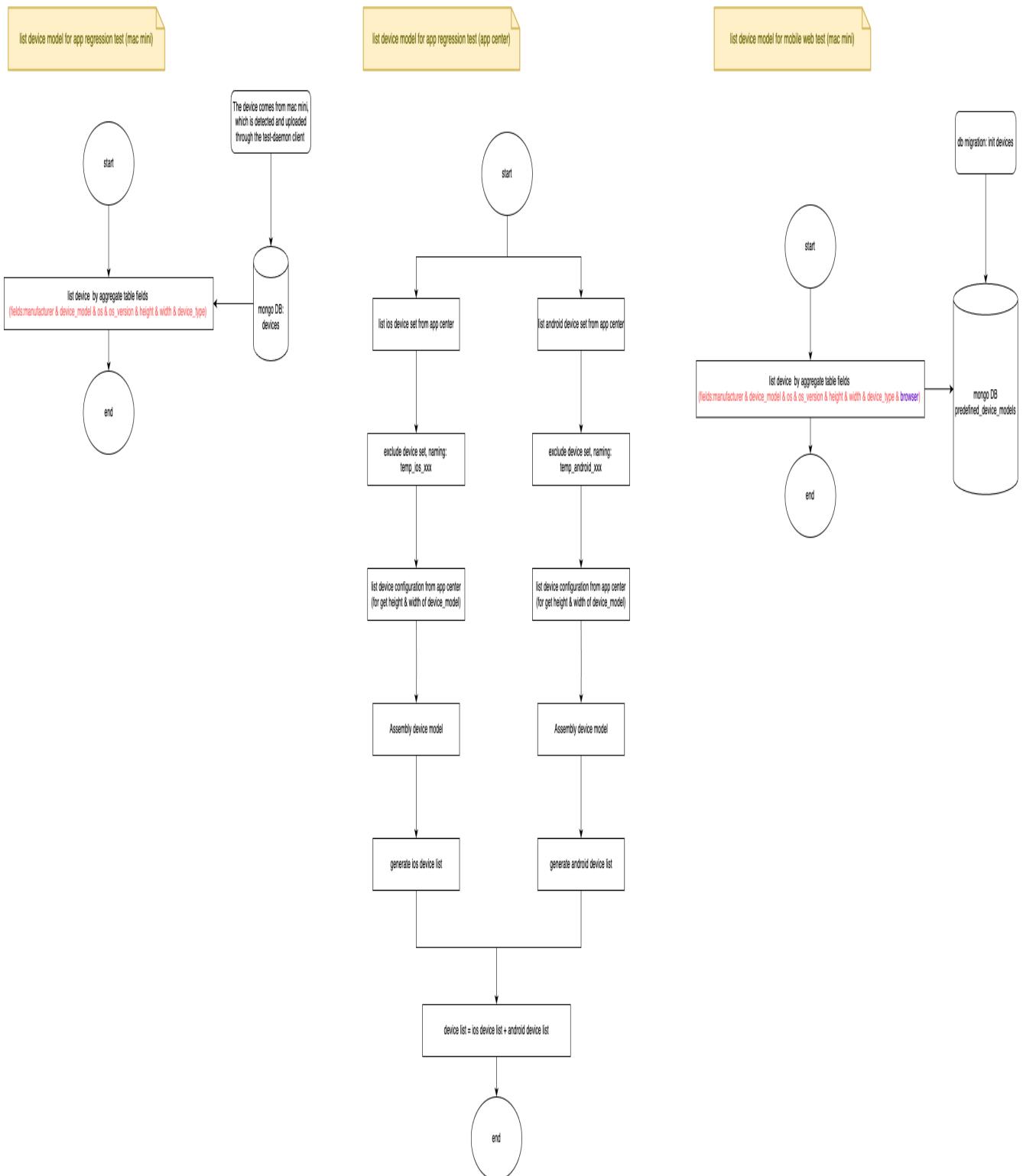


test-device-service

function: list device model

- list device model for app regression test (mac mini)
- list device model for app regression test (appcenter)
- list device model for mobile web test (mac mini)

flow diagram



appcenter-service

job: start test run job

```
1 schedule().fixedRate("start-test-run-job", bind(StartTestRunJob.class), Duration.ofSeconds(10));
```

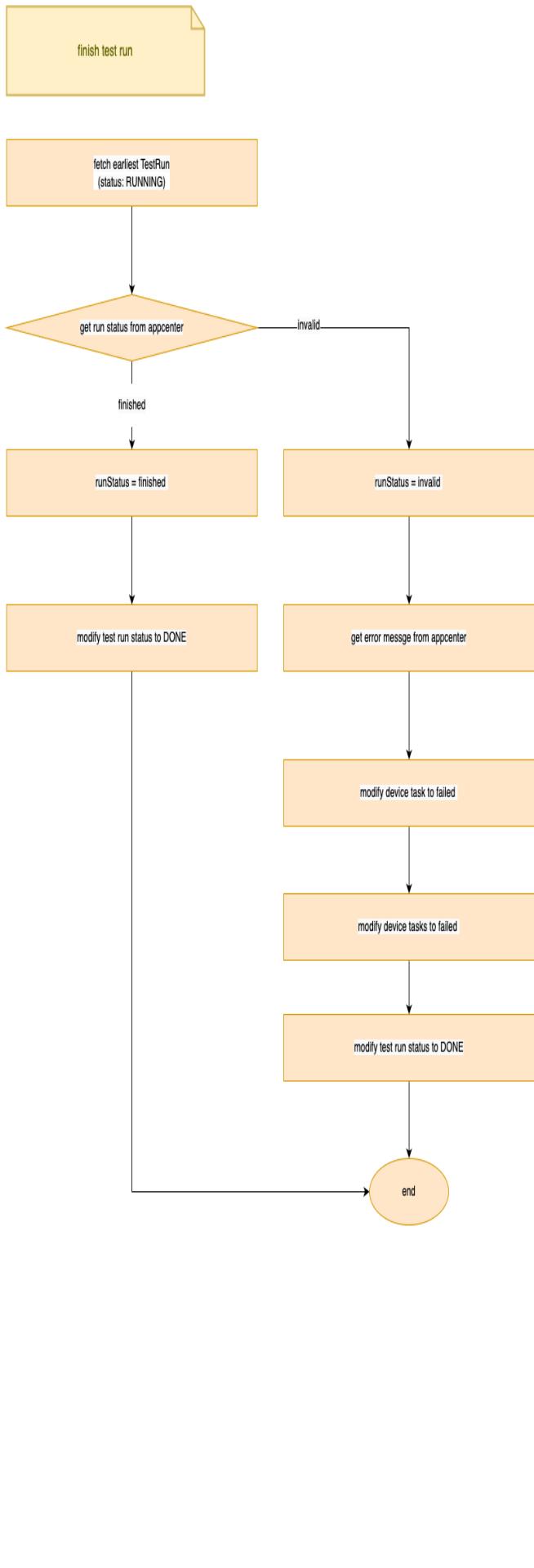
job: finish test run job

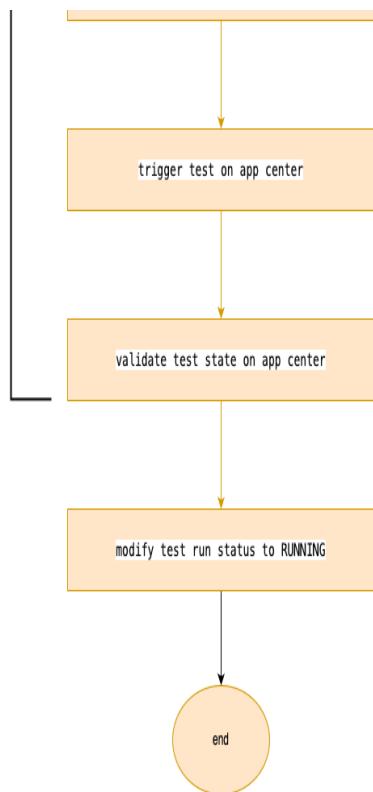
```
1 schedule().fixedRate("finish-test-run-job", bind(FinishTestRunJob.class), Duration.ofSeconds(10));
```

[flow diagram](#)



for ios: create a test run
for android: create a test run

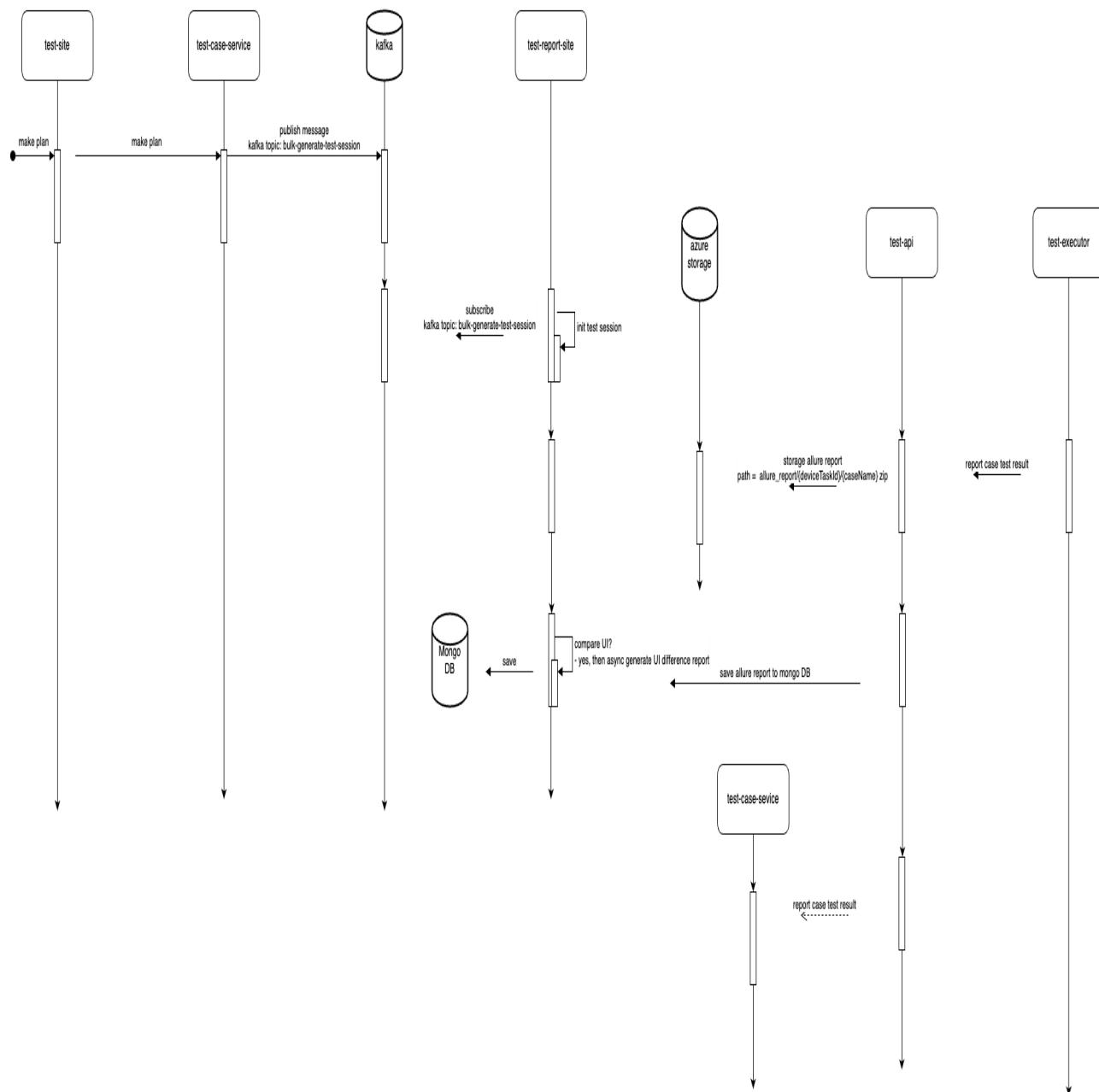




test-report-site

Sequence diagram: UI Diff Report & Allure report

init test session &
report case test result
async generate UI Difference Report by test_case



function: generate Allure Report (Embedded web project)

- forntend service:
 - pre-saved file: `app.js` & `styles.css`, location:

```

1 test-report-site-frontend
2   |- static
3     |- allure
4       |- app.js
5       |- styles.css
  
```

- backend service, generate allure report (Embedded web project)
 - pre-saved file: `index.html`, location:

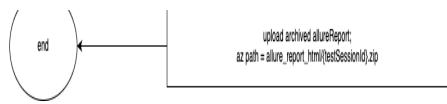
```

1 resources
  
```

```
2 └── allure  
3   └── index.html
```

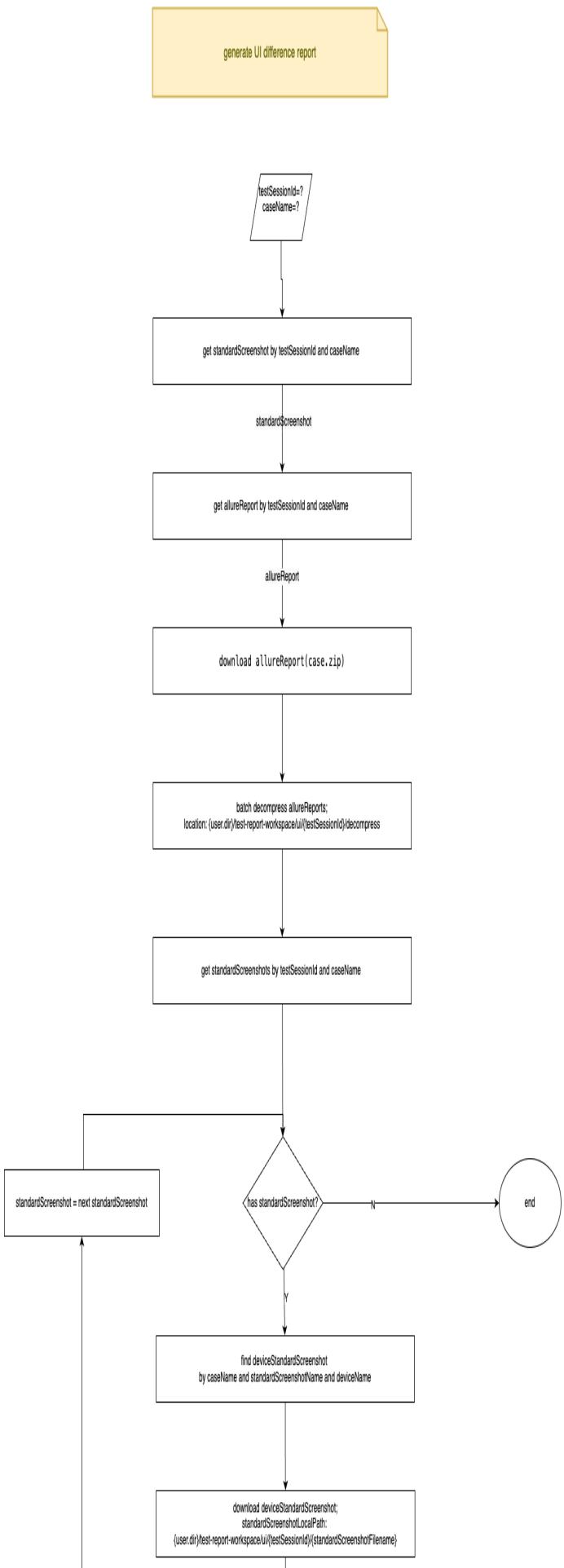
- generate flow diagram





```
upload archived allureReport;  
az path = allure_report_html/{testSessionId}.zip
```

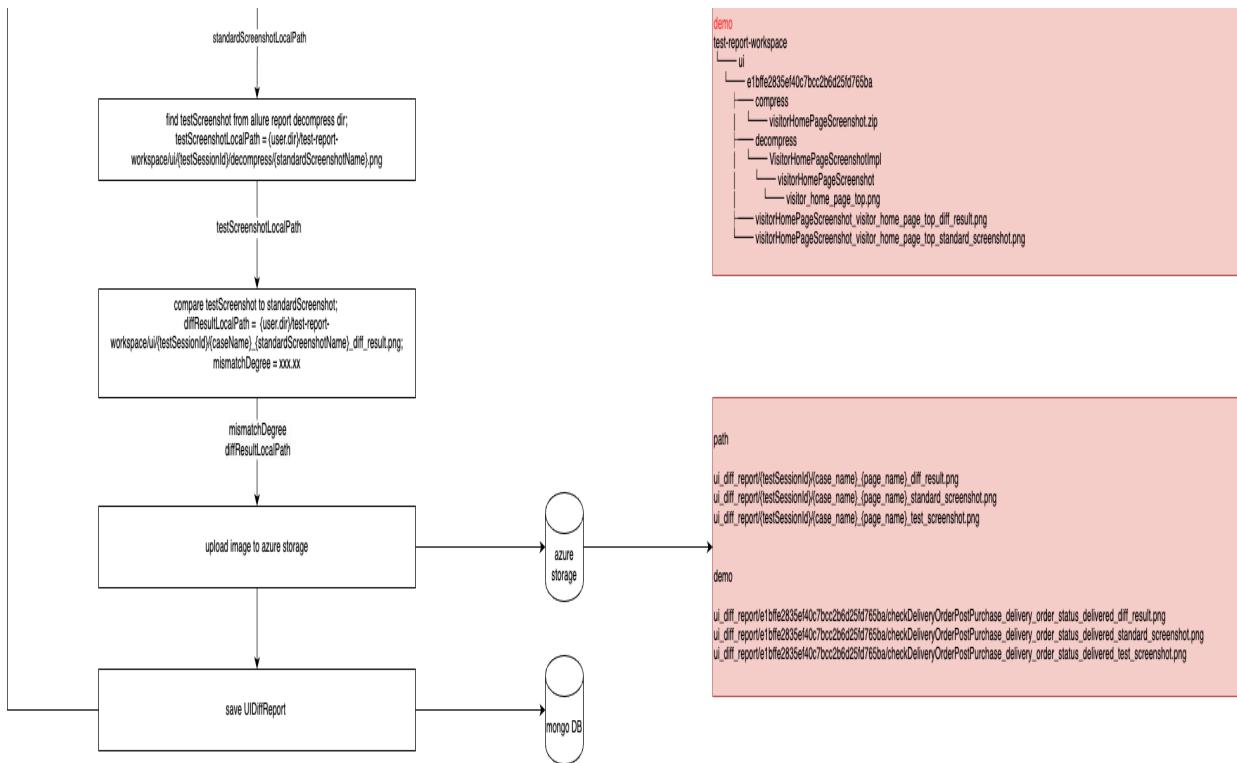
function: generate UI difference Report



workspace

```

test-report-workspace
└── ui
    └── (testSessionId)
        ├── compress
        │   └── (allureReport.zip)
        ├── decompress
        │   └── (groupname)
        │       └── (caseName)
        │           └── (screenshot.png)
        └── (caseName), (screenshotName), diff_result.png
            └── (caseName), (screenshotName), standard_screenshot.png
    
```



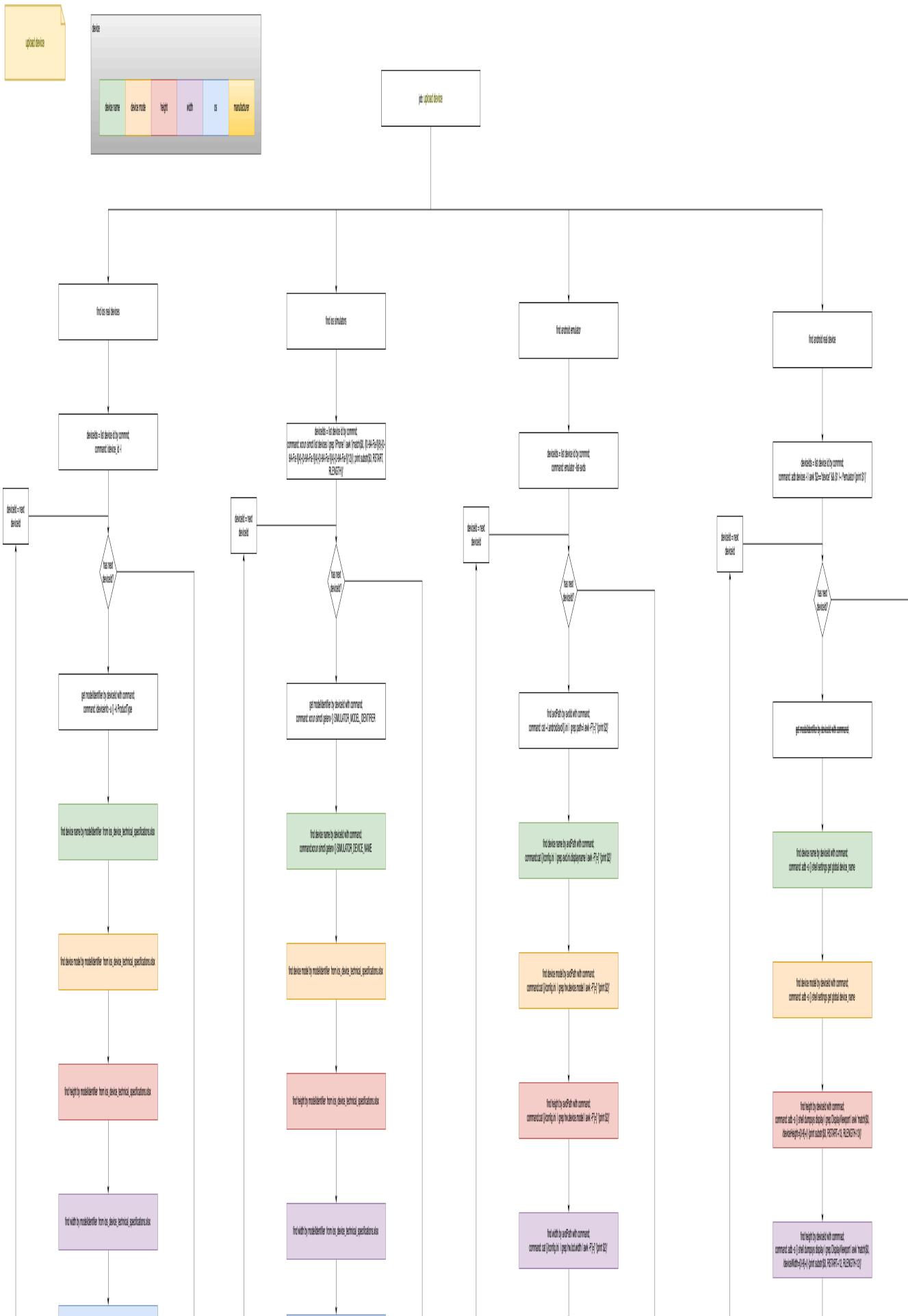
Client

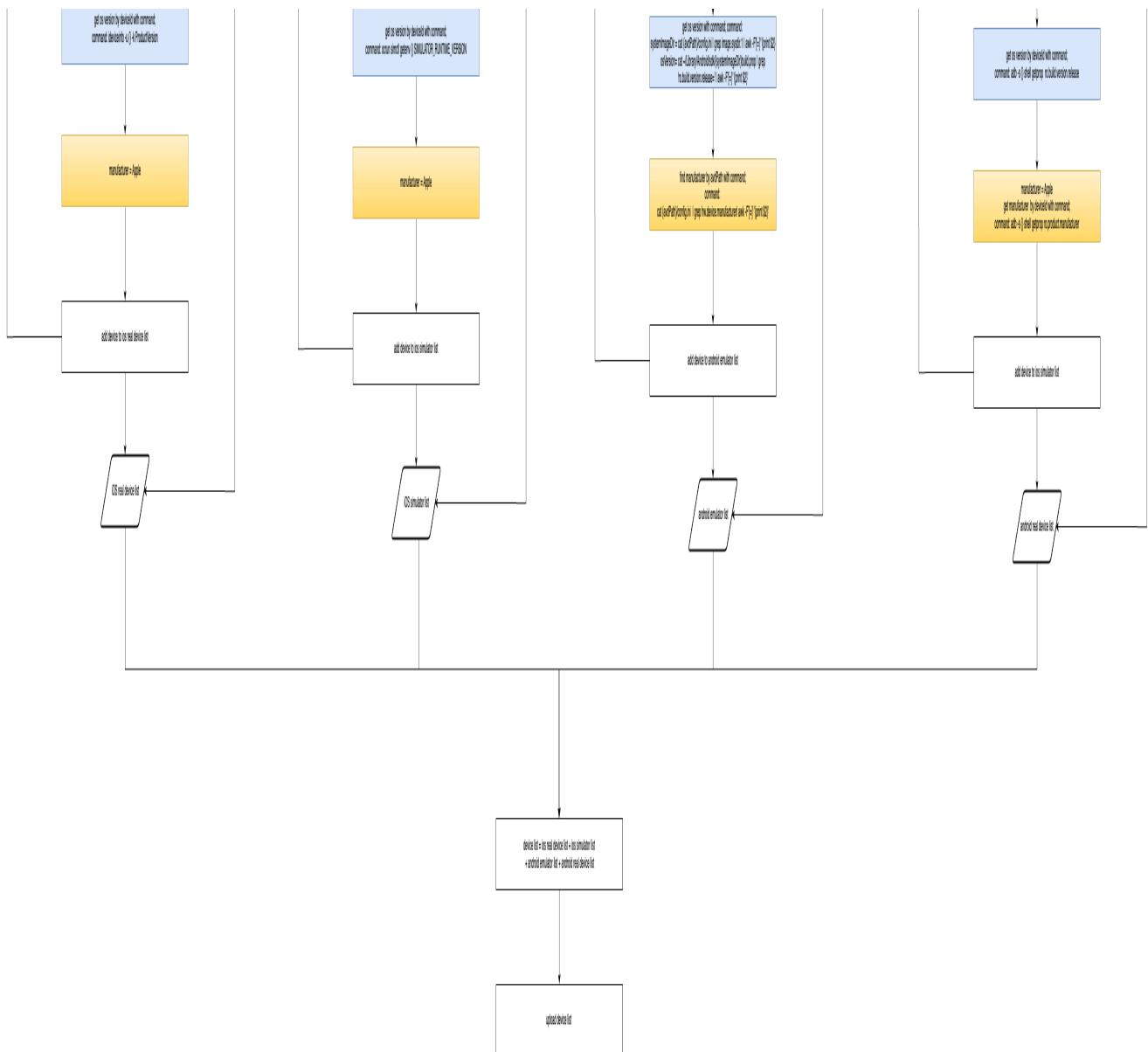
test-daemon

job: upload device

```
1 schedule().fixedRate("upload-device-job", bind(UploadDeviceJob.class), Duration.ofSeconds(10));
```

flow diagram



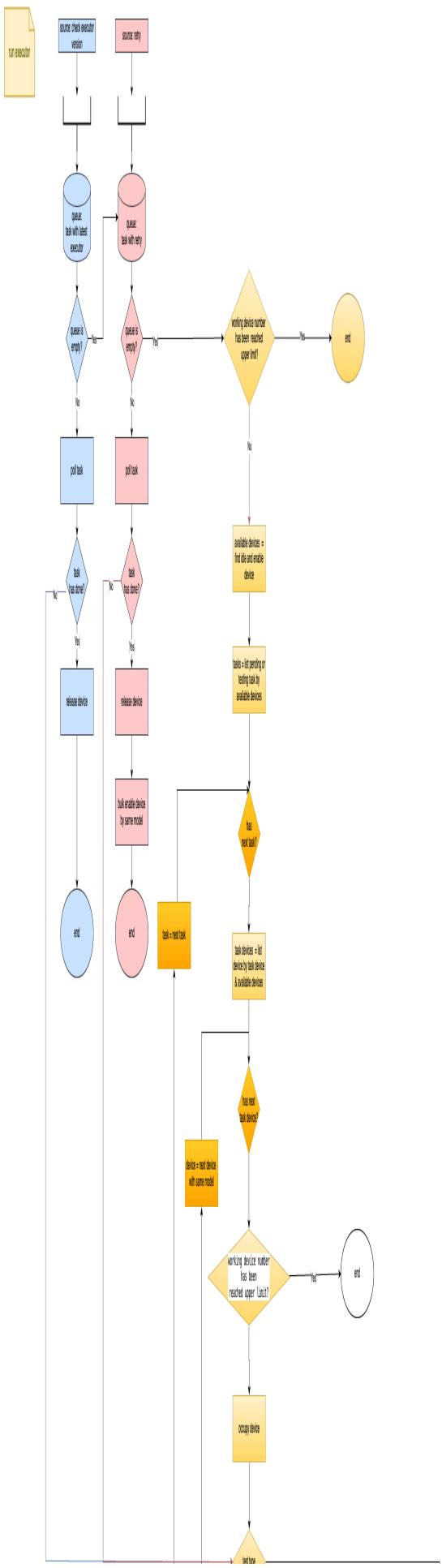


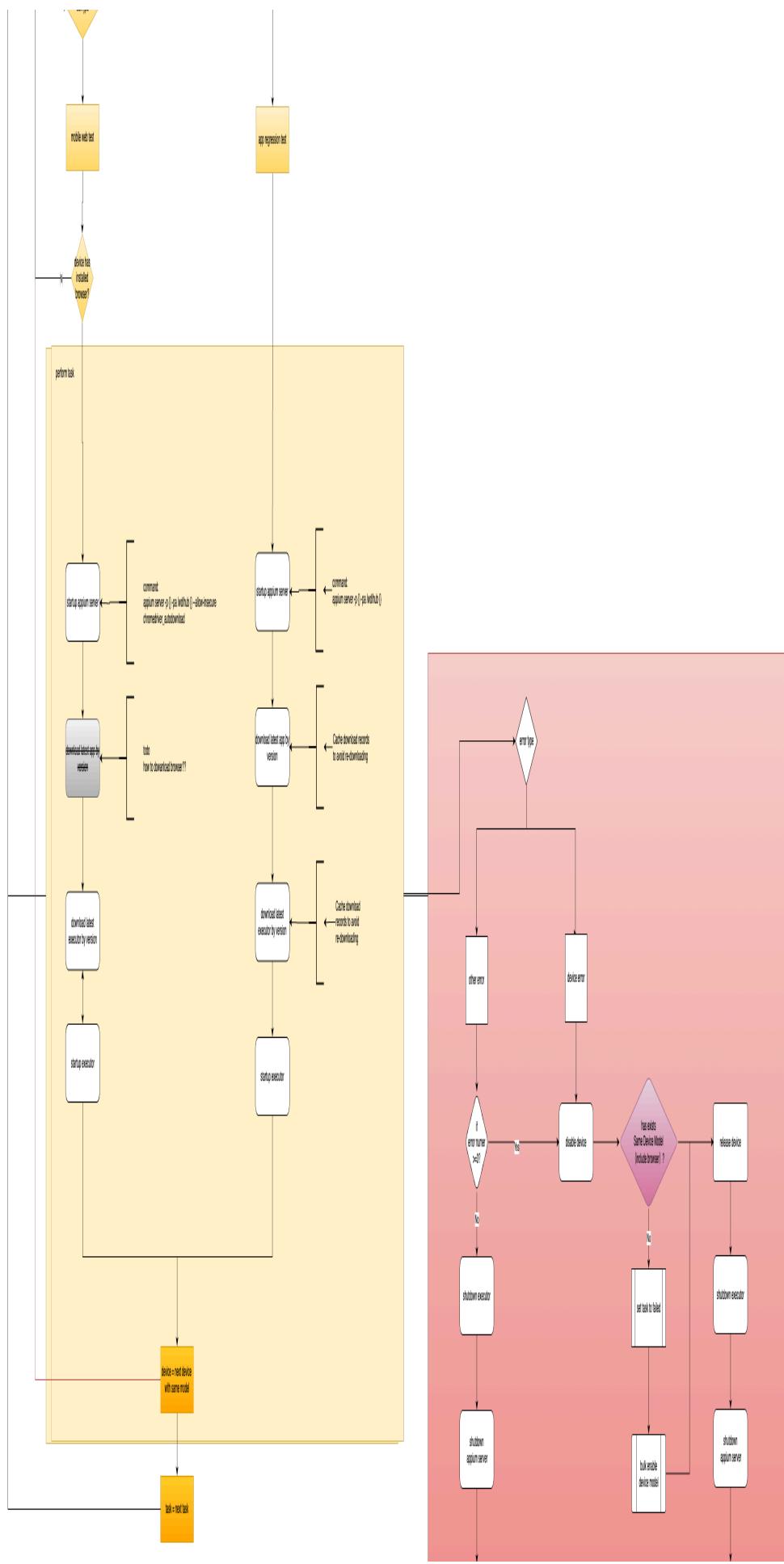
job: run executor

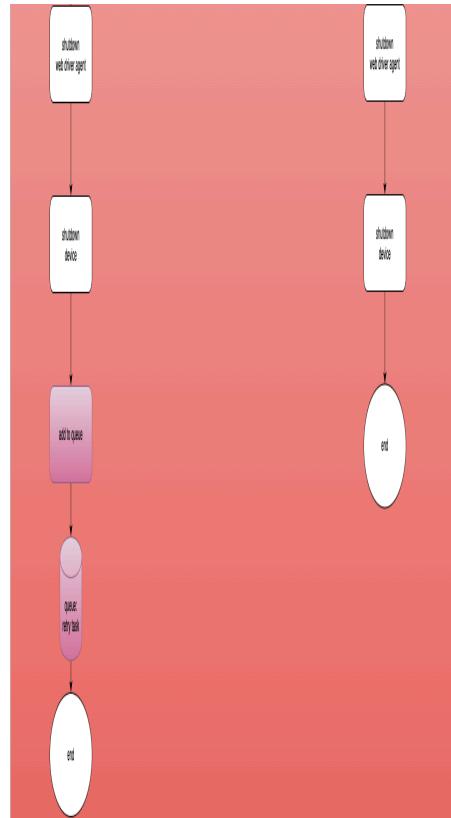
```
1 schedule().fixedRate("startup-test-executor-job", bind(StartupExecutorJob.class), Duration.ofSeconds(10));
```

task source: check executor version, error retry, next pending task
priority : check executor version > error retry > next pending task

flow diagram







How to use?

[Test Site User Guide](#)