

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

 - generate Allure Report (Embedded web project)

 - generate UI difference Report

Client

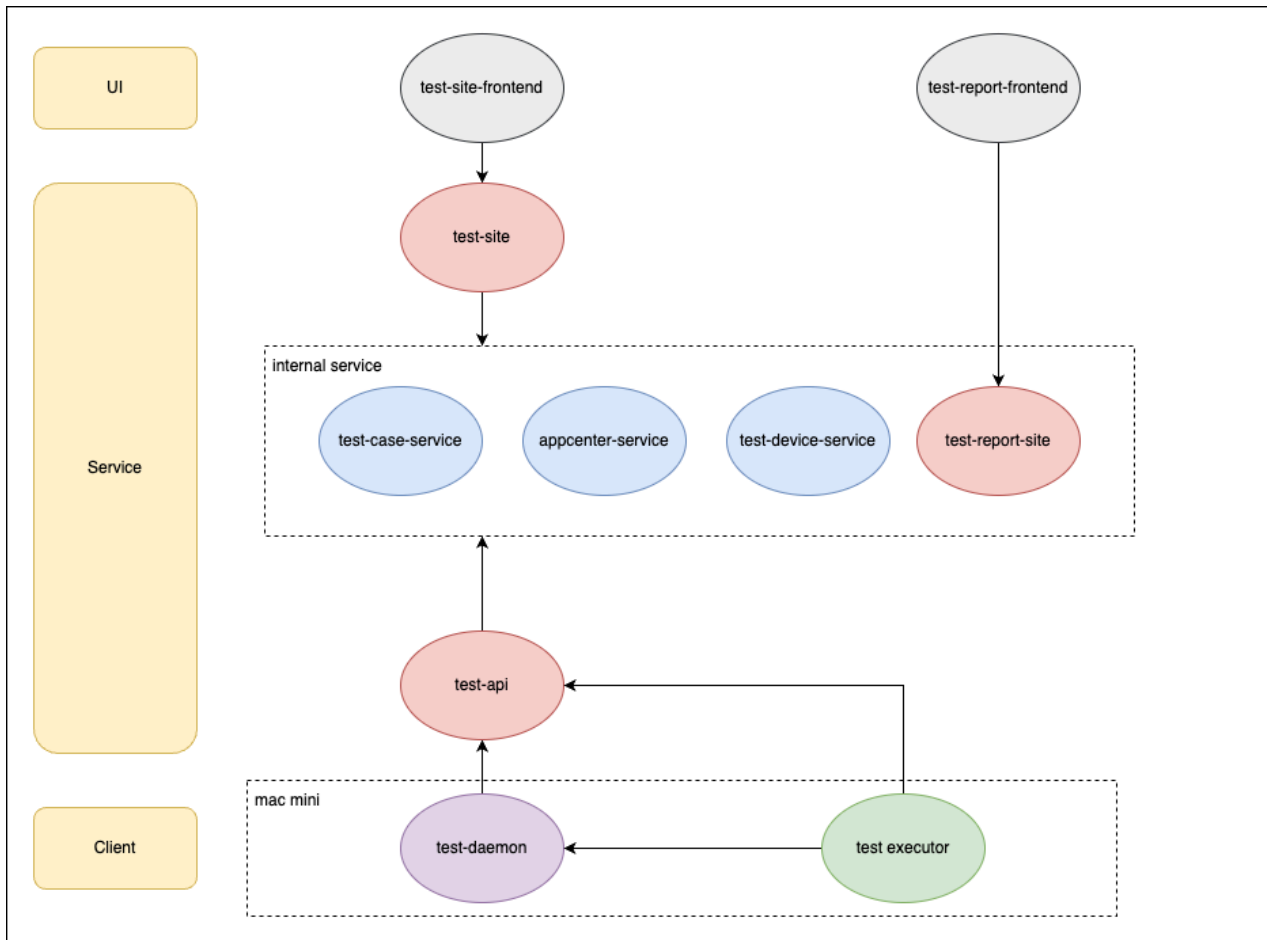
- test-daemon

 - job: upload device

 - job: run executor

How to use?

Summary



UI

i 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. $\text{plan.start_time} = \min(\text{device_task.start_time})$
- b. $\text{plan.end_time} = \max(\text{device_task.end_time})$
- c. $\text{device_task.start_time} = \min(\text{executor_case.start_time})$
- d. $\text{device_task.end_time} = \max(\text{case_test_result.end_time})$

2. Waiting minutes =

- a. $\text{Testing} = \text{plan.startTime} - \text{plan.createdTime}$
- b. $\text{Done} = \text{plan.startTime} - \text{plan.createdTime}$
- c. $\text{Pending} \rightarrow \text{Cancel} = \text{plan.cancelTime} - \text{plan.createdTime}$
- d. $\text{Testing} \rightarrow \text{Cancel} = \text{plan.startTime} - \text{plan.createdTime}$
- e. **$\text{Pending} = \text{now} - \text{plan.createTime}$**

3. Executing minutes =

- a. **$\text{Testing} = \text{now} - \text{plan.startTime}$**
- b. $\text{Done} = \text{plan.endTime} - \text{plan.startTime}$
- c. $\text{Pending} \rightarrow \text{Cancel} = 0$
- d. $\text{Testing} \rightarrow \text{Cancel} = \text{plan.cancelTime} - \text{plan.startTime}$
- e. $\text{Pending} : 0$

function: parse case

flow diagram

dependencies

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

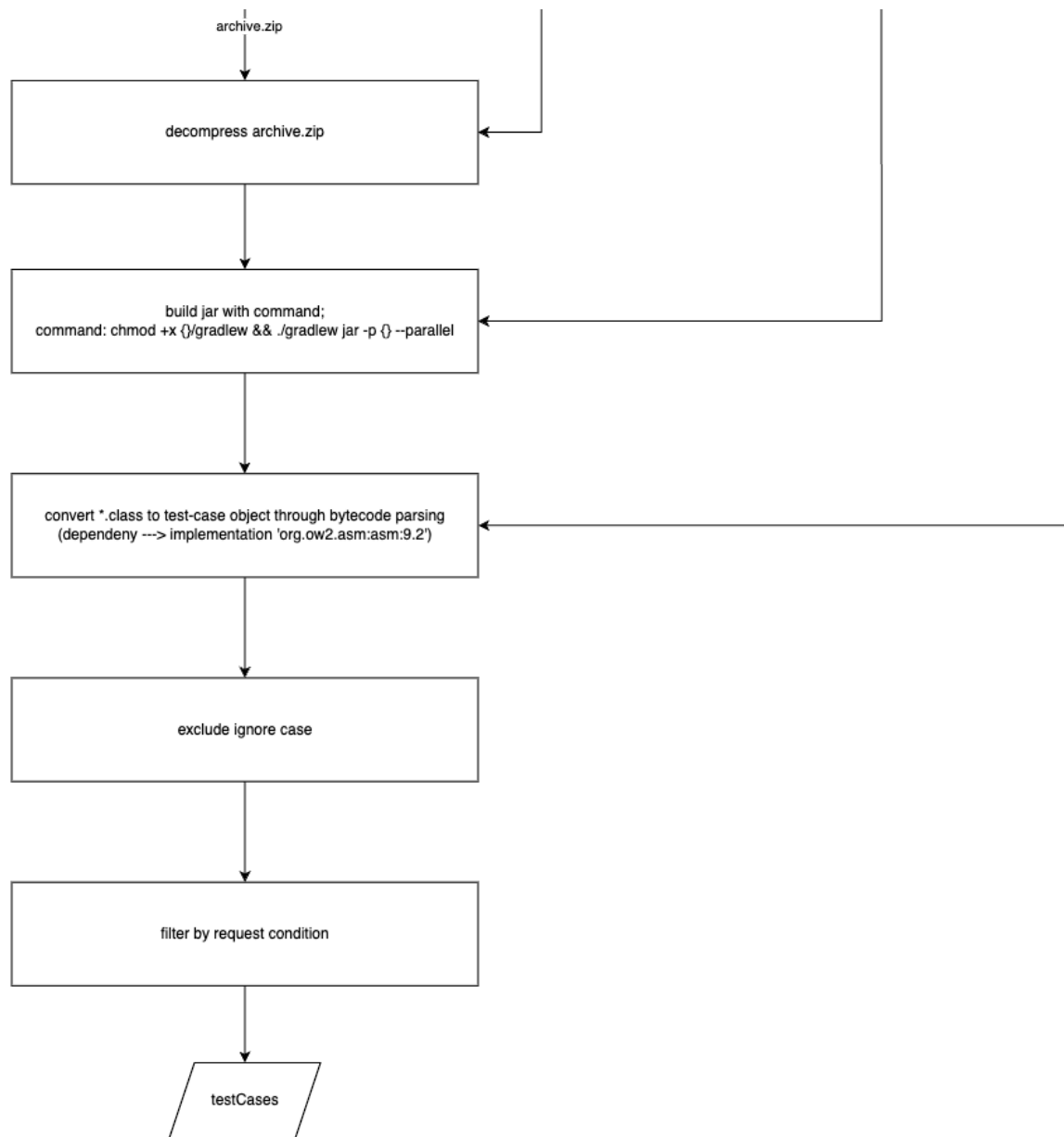
workspace**Directory Structure**

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

Demo

```
test-case-workspace
├── b22b03f7e20c8350cfc05a8c52de0dbdfa41f58
│   ├── archive.zip
│   ├── consumer-web-test-executor
│   └── build
│       ├── consumer-web-test-interface
│       └── libs
│           └── com.wonder.consumer-web-test-interface-1.0.0.jar
└──
```



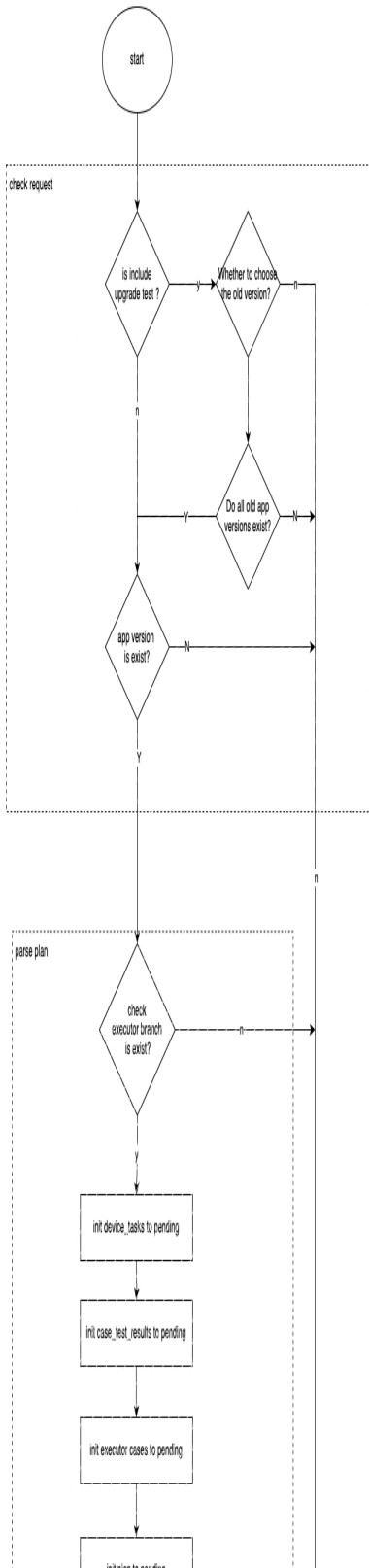


function: make plan

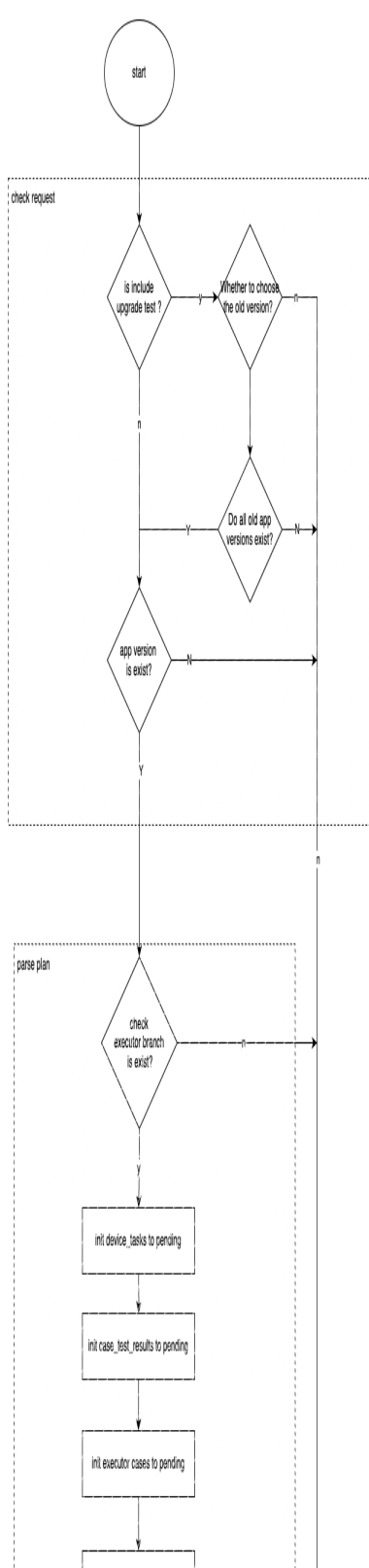
- make plan for app regression test (mac mini)
- make plan for app regression test (appcenter)
- make plan for mobile web test (mac mini)

flow diagram

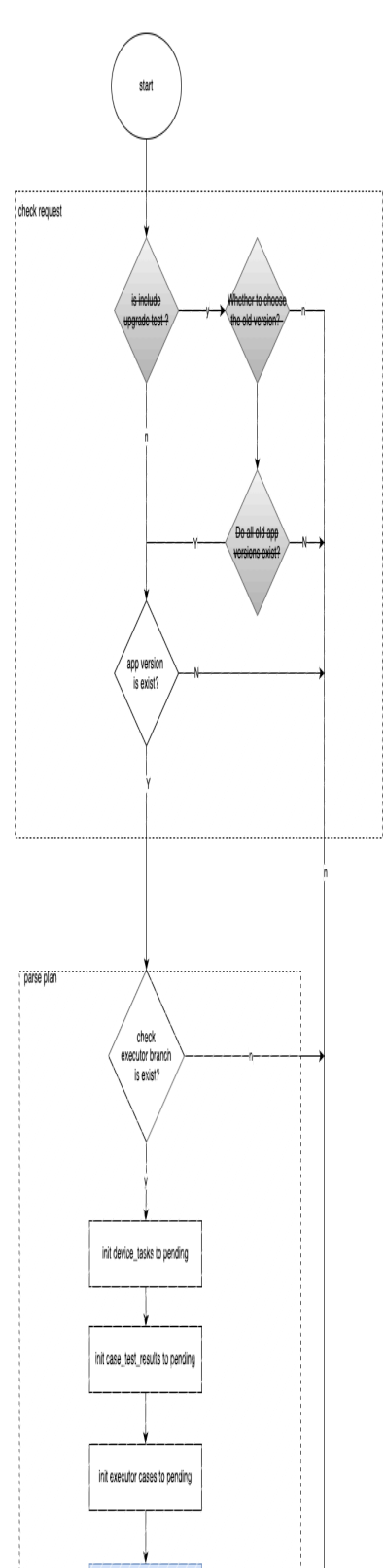
make plan for app regression test (mac mini)

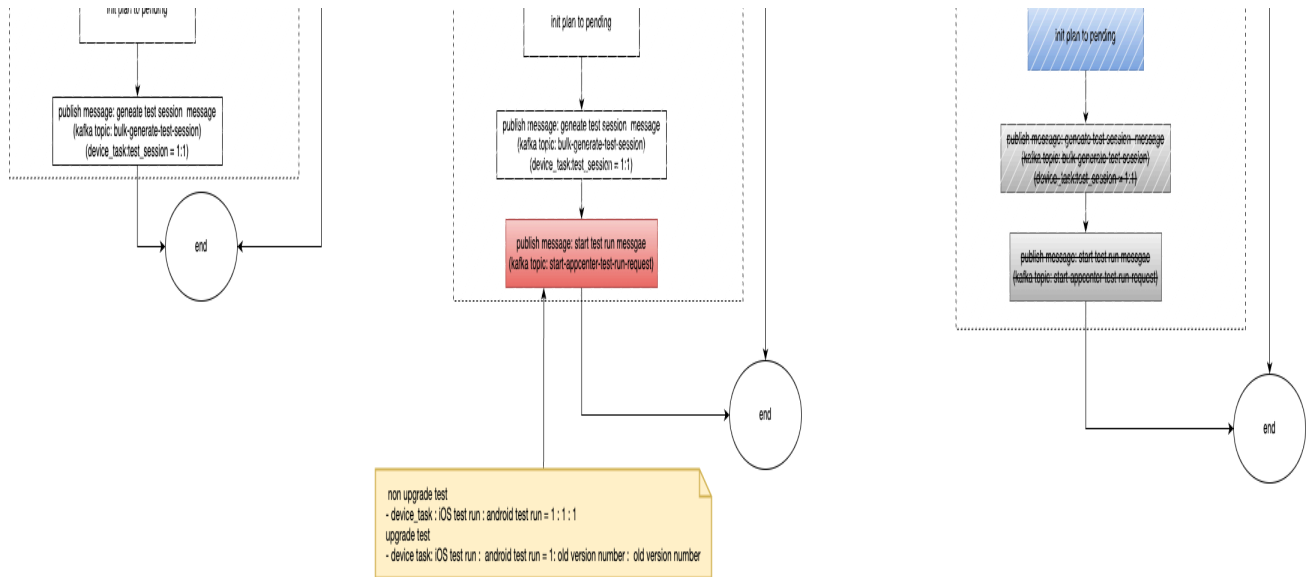


make plan for app regression test (appcenter)



make plan for mobile web test (mac mini)



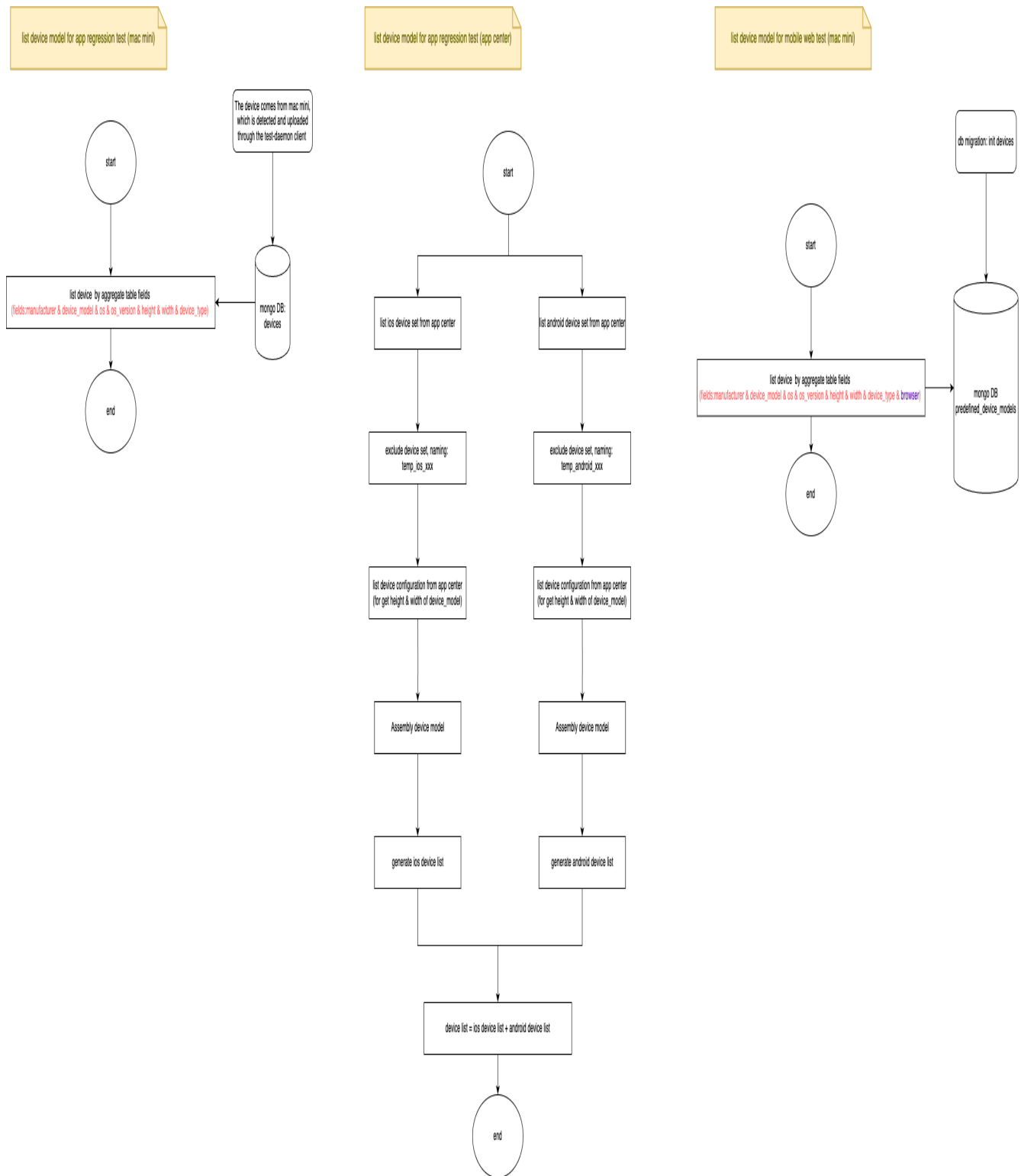


test-device-service

function: list device model

- list device model for app regresstion 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

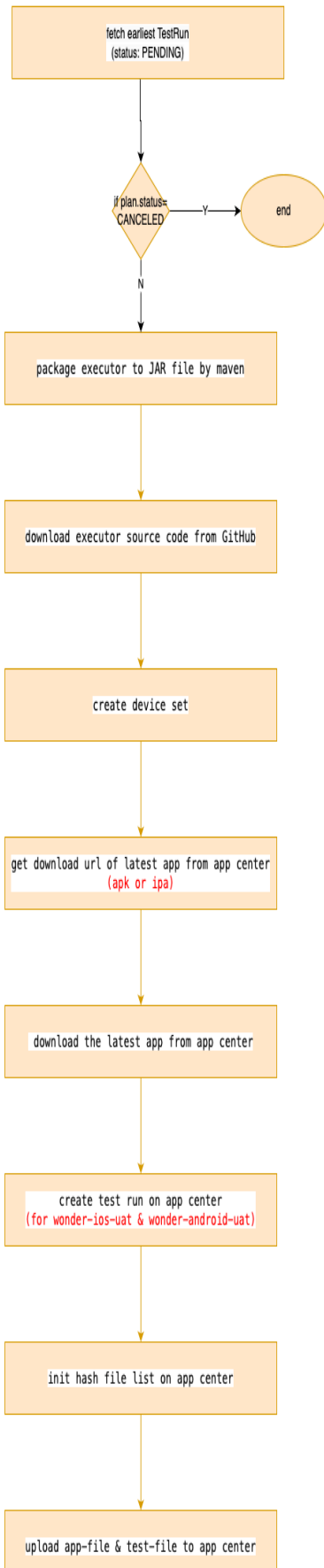
```
1 schedule().fixedRate("start-test-run-job", bind(StartTestRunJob.class), Duration.ofSeconds(10));
```


job: finish test run job

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

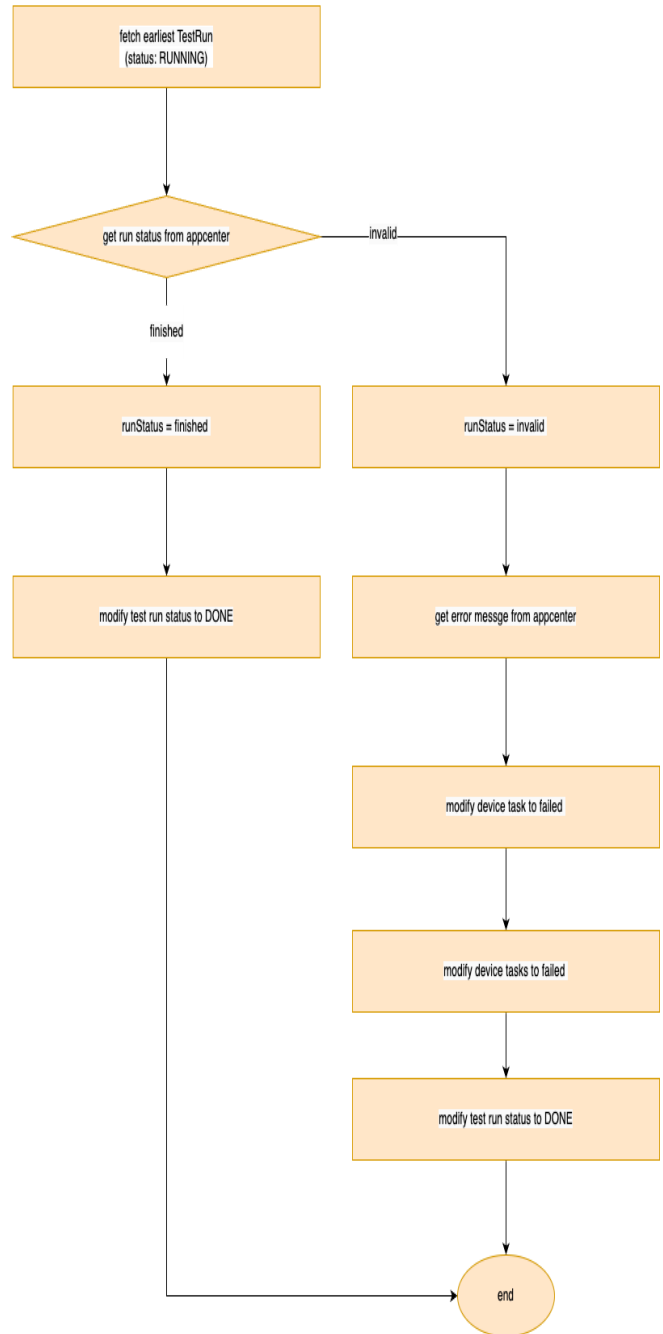
flow diagram

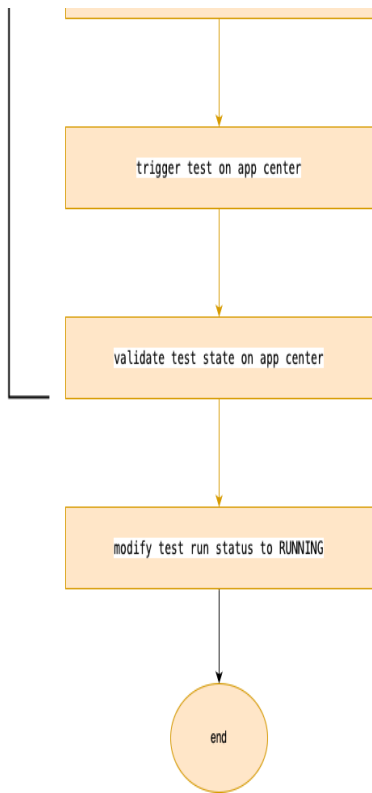
start test run



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

finish test run

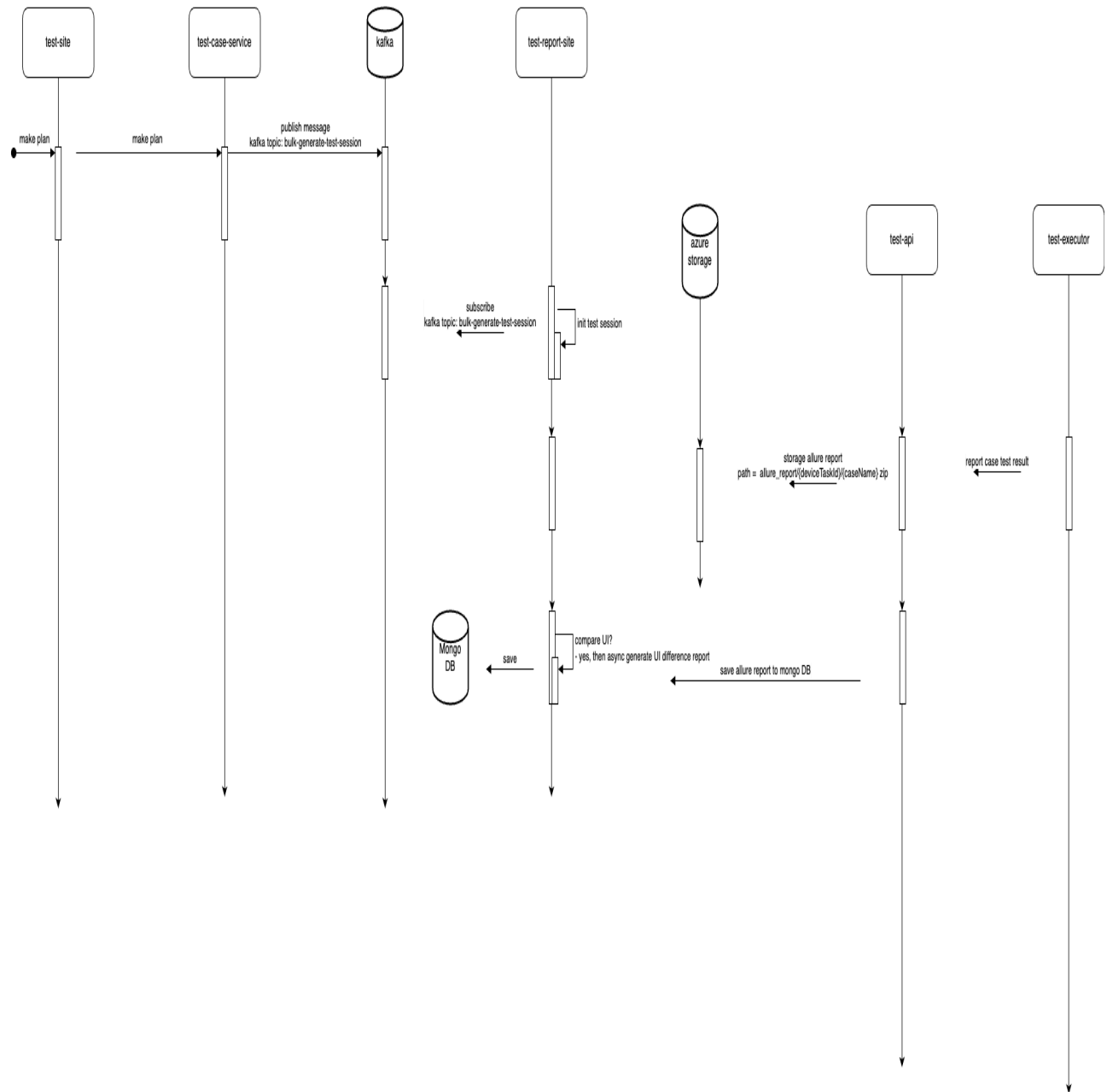




test-report-site

Sequence diagram

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



generate Allure Report (Embedded web project)

- frontend 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:

```

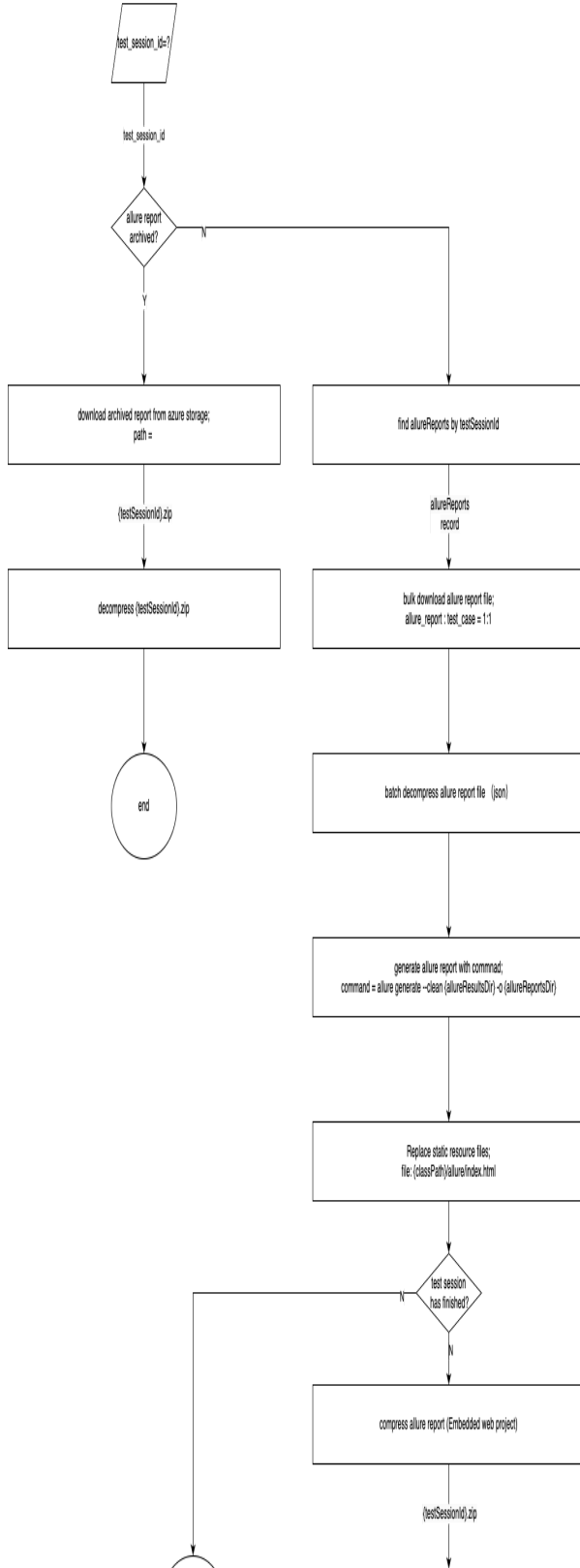
1 resources

```

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

- generate flow diagram

generate allure report web project



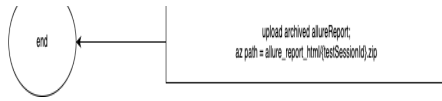
```
test-report-site-frontend:
allureReportPath: /allure-report/(testSessionId)/allure-reports/index.html
<iframe key=(temp) scrolling="yes" title="report" width="100%" height="1000" frameborder="0" src={allureReportPath} />
```

```
API: AllureReportAAWWebService.generate()
```

```
workspace
```

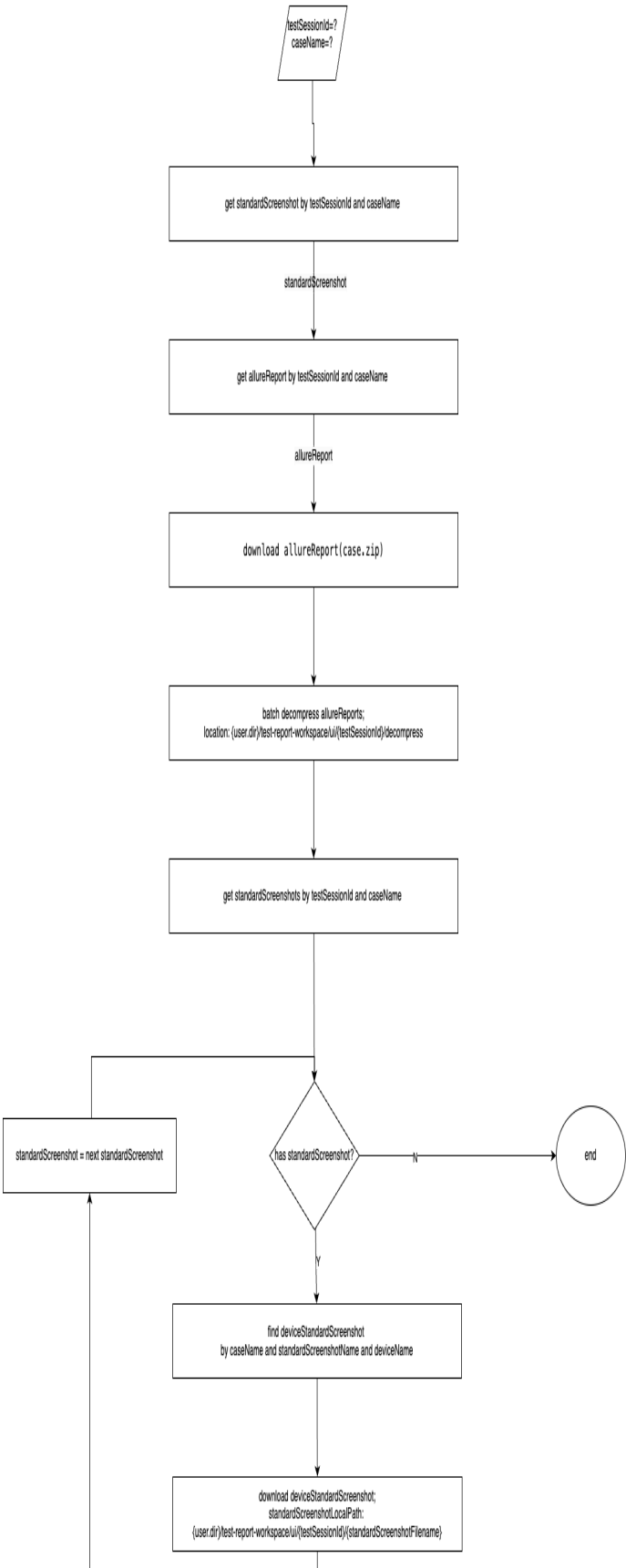
```
test-report-workspace
```

```
allure
├── a10fe2835e40c70cc2b625d7630a
│   ├── allure-reports
│   │   ├── app.js
│   │   ├── data
│   │   ├── export
│   │   ├── favicon.ico
│   │   ├── history
│   │   ├── category-trend.json
│   │   ├── duration-trend.json
│   │   ├── history-trend.json
│   │   ├── history.json
│   │   └── retry-trend.json
│   ├── index.html
│   ├── plugins
│   ├── styles.css
│   ├── widgets
│   ├── allure-results
│   └── compress
└── a10fe2835e40c70cc2b625d7630a.zip
```



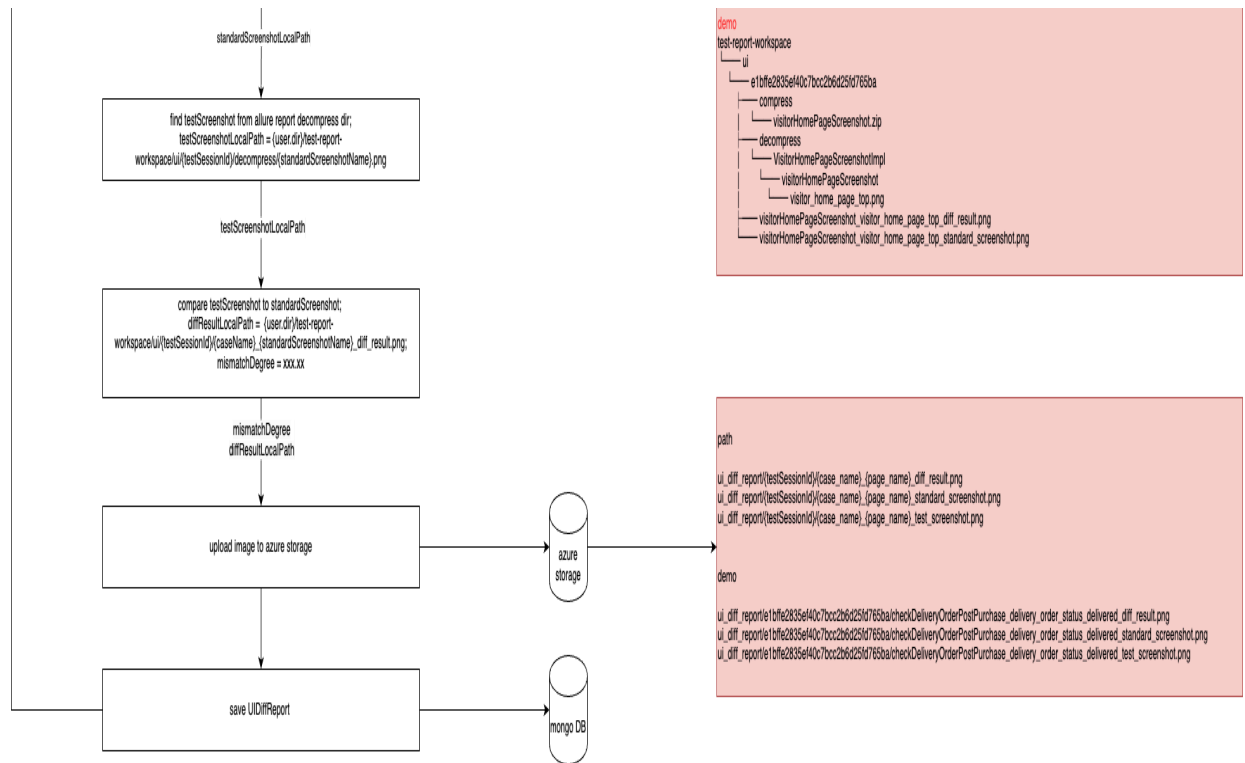
generate UI difference Report

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



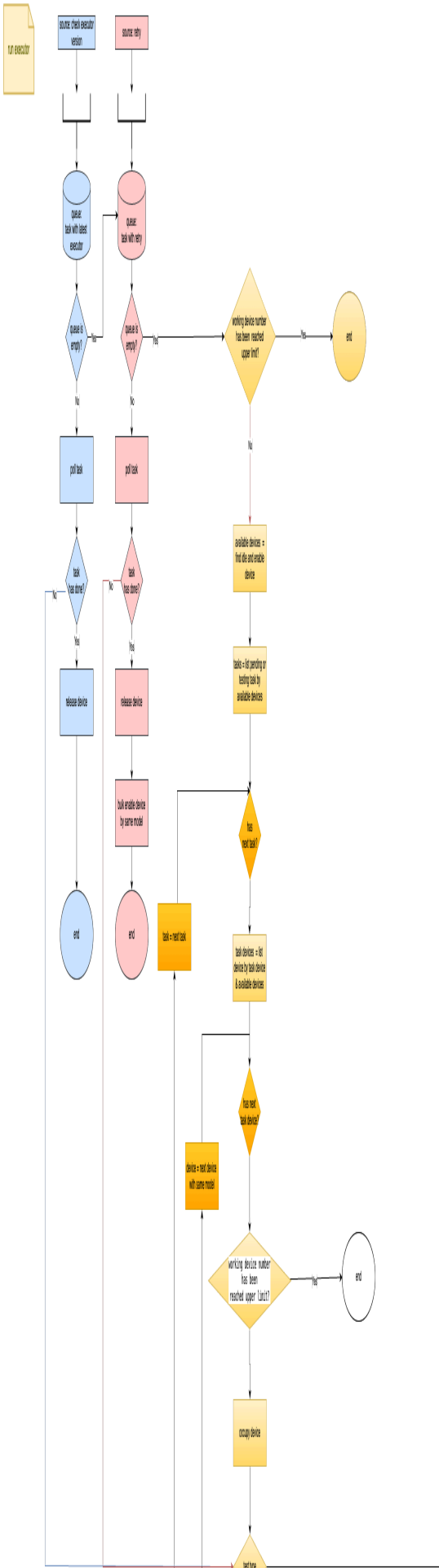
job: run executor

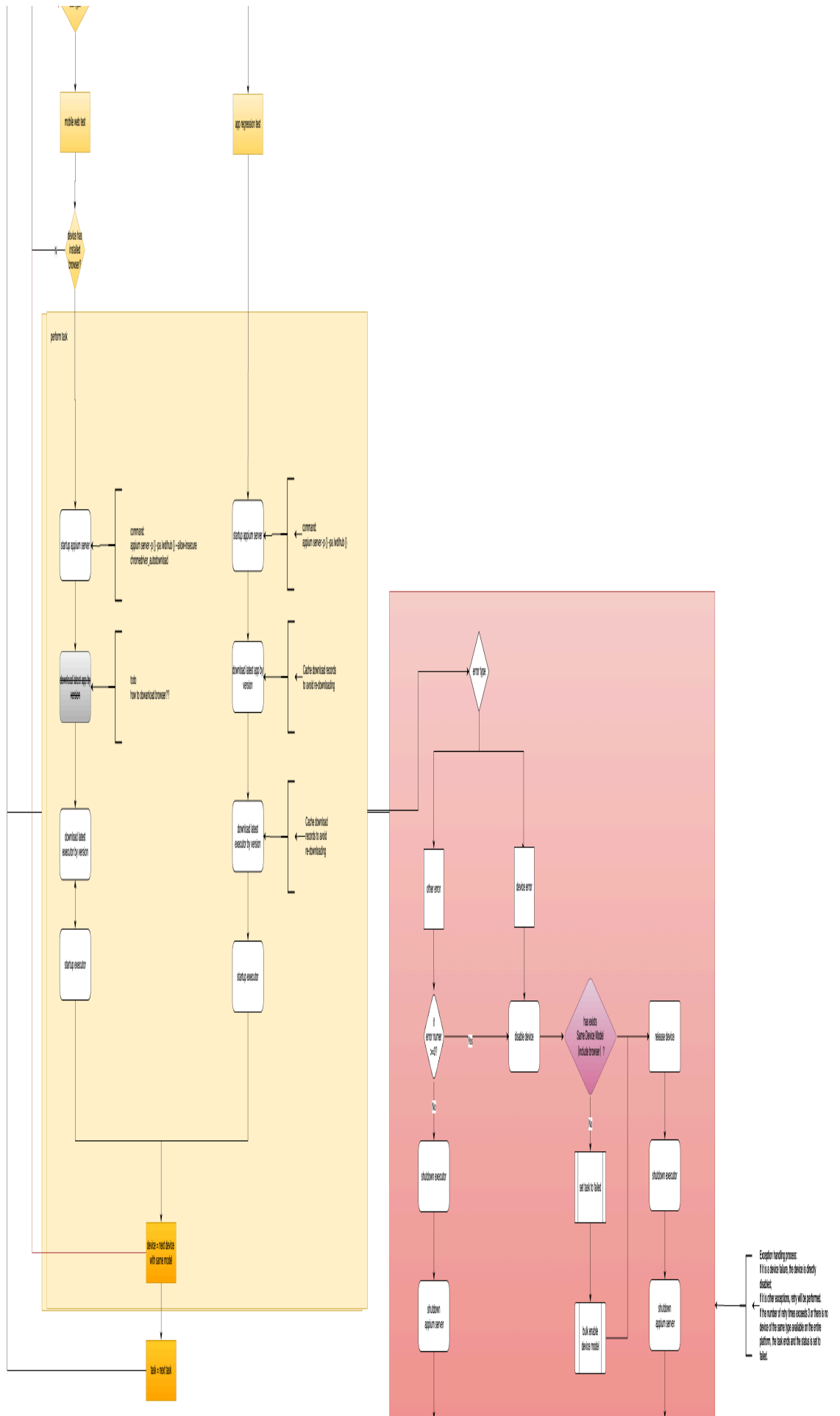
```
1 schedule().fixedRate("startup-test-executor-job", bind(StartupExecutorJob.class), Duration.ofSeconds(10));
```

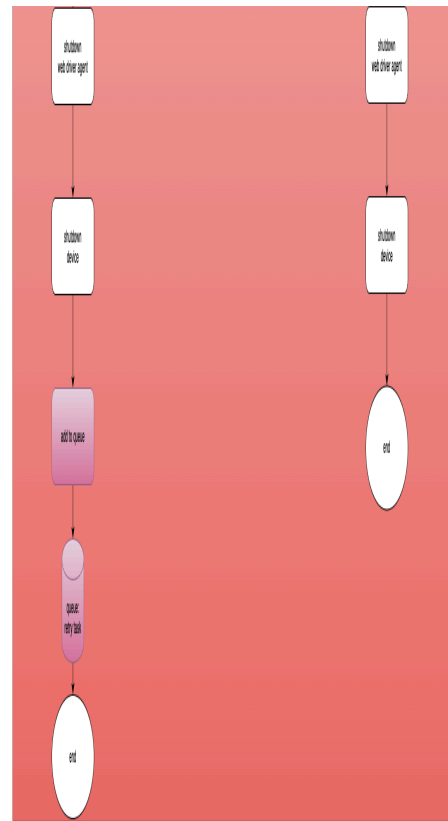
task source: check executor version, error retry, next pending task

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

flow diagram







How to use?

[User Guides](#)