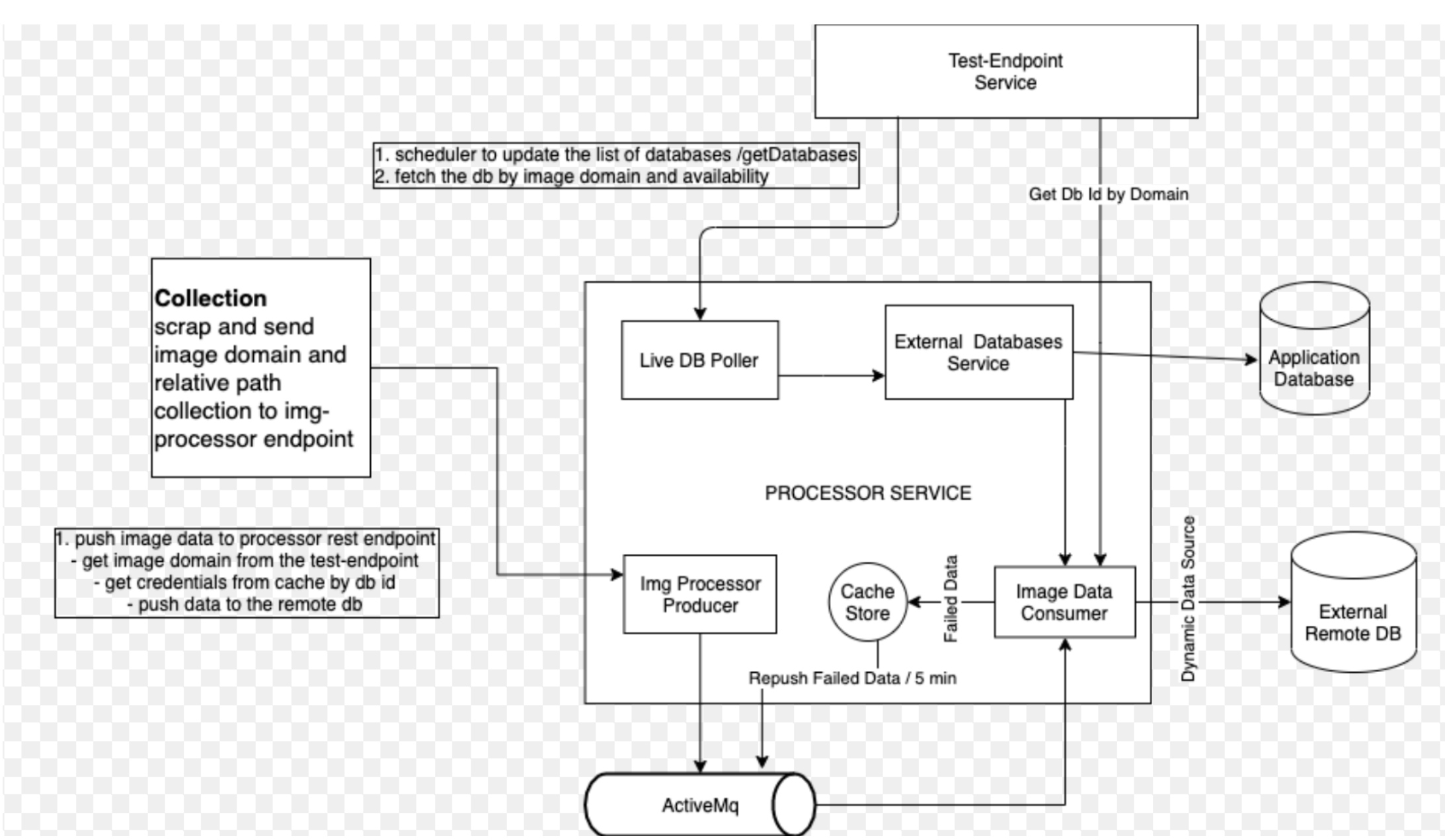
img-processor

Objectives

Image Processor application intended to poll the live available external databases configs and process the image data/file paths with domains.

1. This application uses local in memory h2 database for storing live databases polled from test-endpoints.jar application.
2. It exposes push api to consume the image data and store it to the activemq for async processing.
3. Processing image accomplished by checking live database and use dynamic data source routing on the fly to store data. if fails
   due to any reason store it to local store for repush to queue for retry.

High level design



Getting Started Guide

Prerequisites

- Java 17
- local data source (in this app we are using two data sources:
  1. local in memory h2 db for caching live available databases.
  2. alternate database for storing image data at runtime with dynamic data source router. (for now I've used local db with data source properties prefix `spring.alt-datasource`)
- Edit port and your alternate data source specific change to `application.properties.tmpl` file and run `schema-bkp.sql` on your alternate data source.
  (`schema-bkp.sql` is designed for mysql data source.)

> for now default and alternate data sources are same h2 database. (for this you don't have to run the `schema-bkp.sql`)

Running Application

- There will be a couple of files for running the solution in local
- Try to run following commands by opening terminal in same directory

```
# NOTE: use different tabs for each command

# to run test-endpoint application
sh ./runTestEndpoints.sh

# to build the img-processor app jar
sh ./build.sh

# to spin up the activemq and processor app
sh ./start.sh

# to stop the active mq
sh ./stop.sh
```

> These command take care to spin up the test-endpoints (8080) springboot app | active mq (61616) and img-processor (8082) springboot application on the localhost with ports mention with | *make sure these port are free for use in local*

Testing In Action

- `curls.md` this contains all curls related to processor app. use specific curls for pushing data of sample img data.

Processed Data UI

To access and see the processed data visit link

- This ui will show all processed image paths with there domain
- user can navigate through pages and search by domain name

To see the processed data see h2-console after running the application on local

- Use below values which require to access h2 console:

```
{
   "driverClass": "org.h2.Driver",
   "jdbcUrl": "jdbc:h2:mem:mdb",
   "username": "admin",
   "password": "admin"
}
```

Test Data

Random image domain data can be generated at - https://json-generator.com/#
using code:

```
[
  '{{repeat(20)}}',
```

```
  {
    domain: '{{lorem(1, "words")}}.{{random("com", "net", "in")}}',
    images: [
      '{{repeat(3, 10)}}',
      '{{random("app", "static")}}/{{random("images", "img")}}/{{lorem(1, "words")}}.
{{random("jpg", "png", "webp")}}',
      '{{repeat(1, 10)}}',
      '{{lorem(1, "words")}}.{{random("jpg", "png", "webp")}}'
    ]
  }
]
```