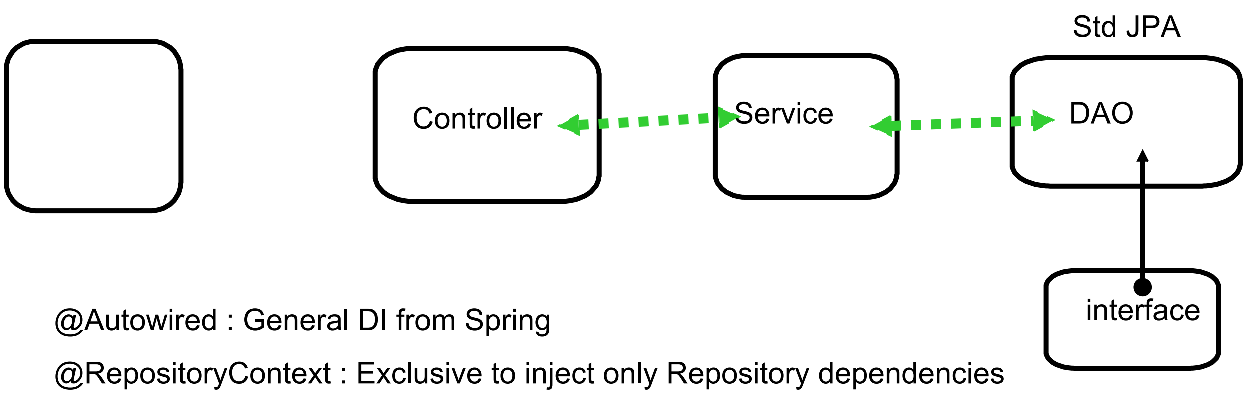
Boot Starter JPA

# USing entity manager (implementing native hibernate)

# USing entity manager (standard JPA methods) : no vendor locking

# Spring Boot Starter Data JPA (concrete implementation of CRUD functionalities)



@Autowired : General DI from Spring

@RepositoryContext : Exclusive to inject only Repository dependencies

TO add custom query requirements

1. add / declare new method in interface having correct names

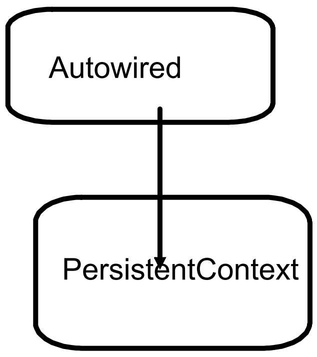
Sql

HQL

JPQL

interface : <RepositoryName>Custom

StudentDaoCustom



PersistantContext : Exception arising by EM will be represented by exception classes of JPA exception classes

Student (DAO->Service->Controller) [CRUD Functionality]
Employee (DAO->Service->Controller) [CRUD Functionality]
Book (DAO->Service->Controller) [CRUD Functionality]

Spring boot starter data rest
addition of data-jpa project + rest project

DaoRepository (interface inherits JPA Repository)
DAO impl
Service
Controller

Rule

Entity : Student ~ plural

/students GET

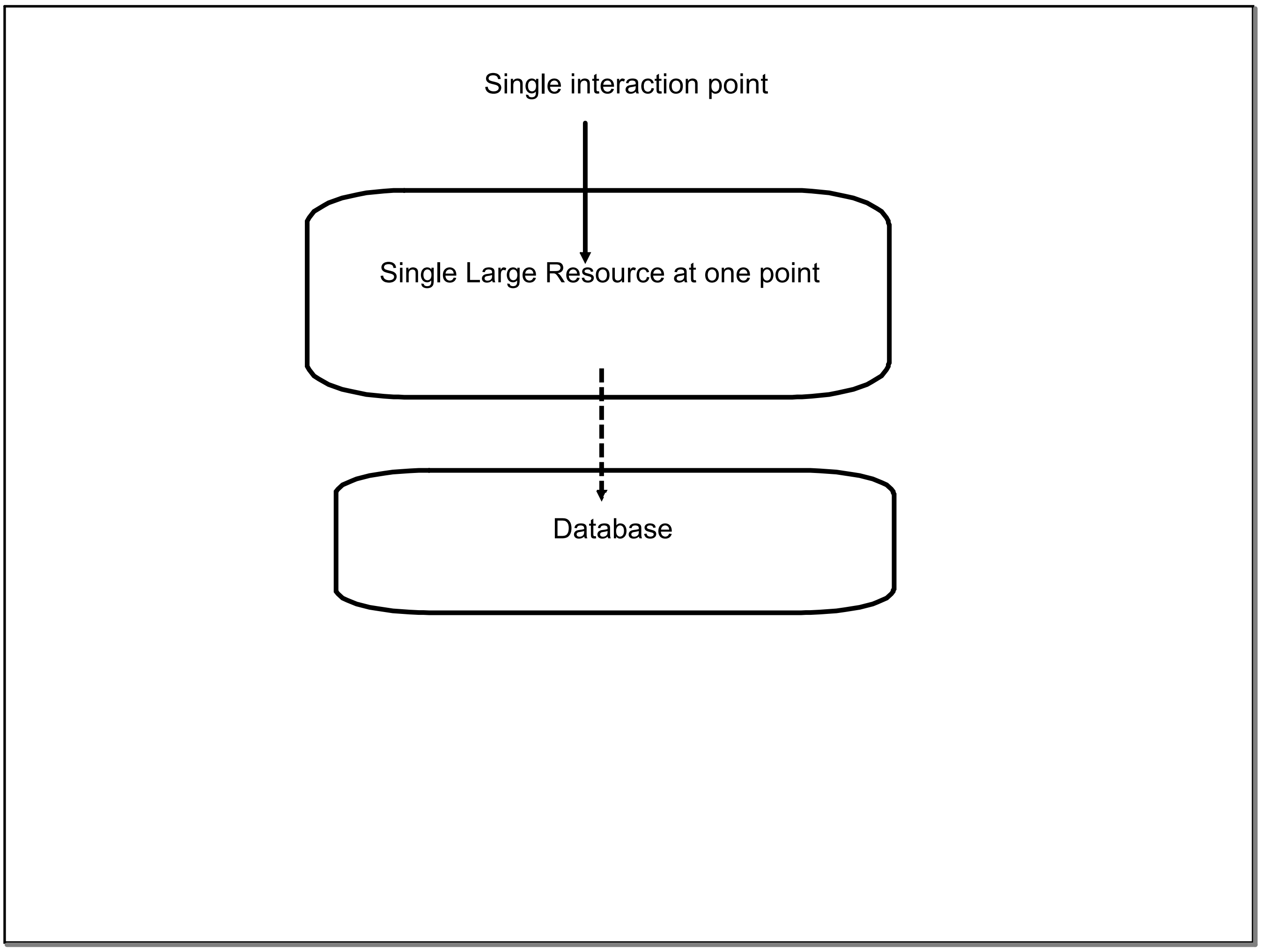
/students/{studentId} GET

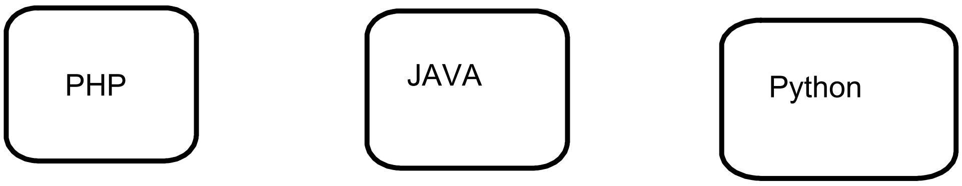
/student POST

/student PUT

/student/{studentId} DELETE

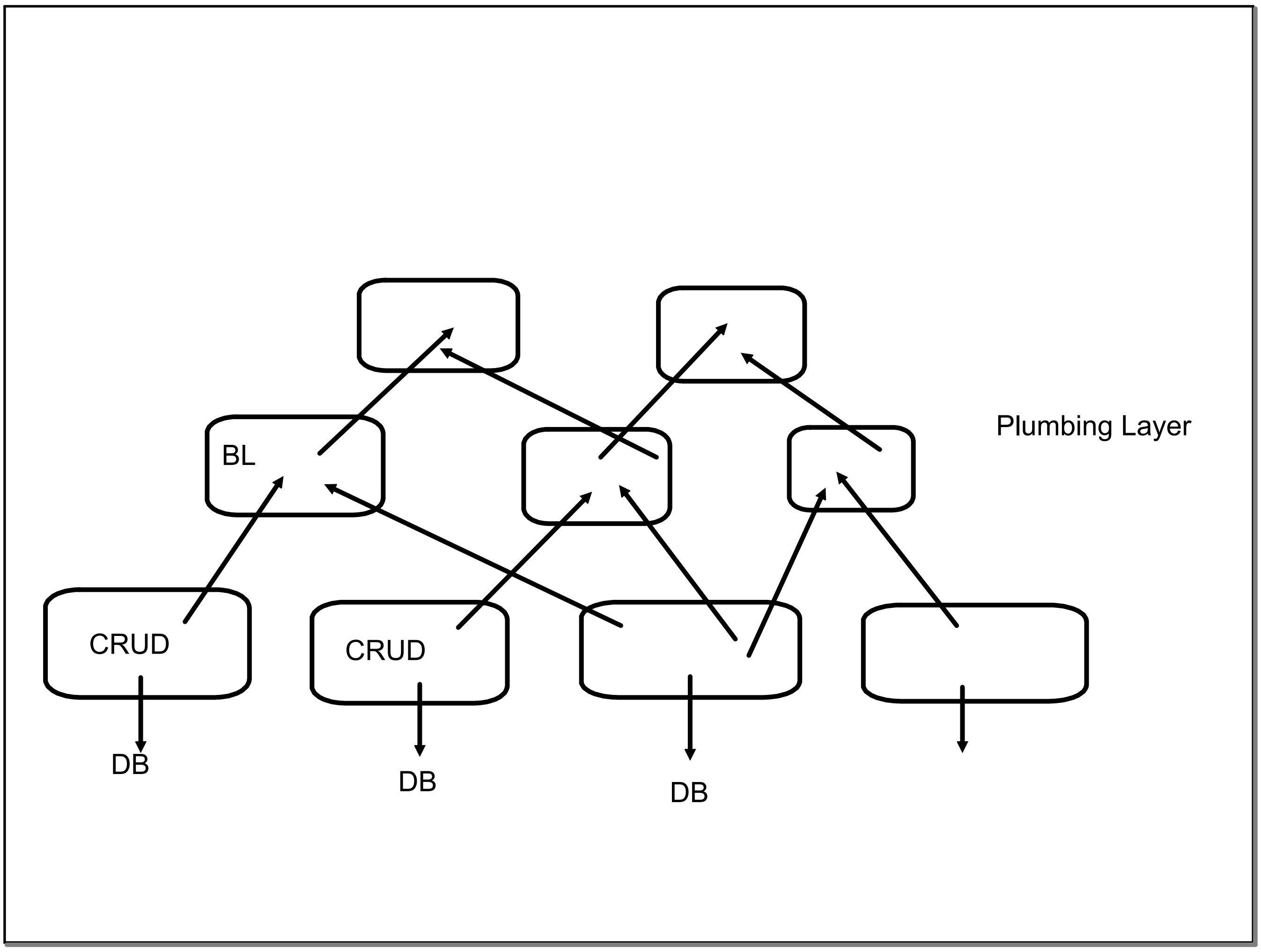
Spring Data REST : HATEOS Compliant

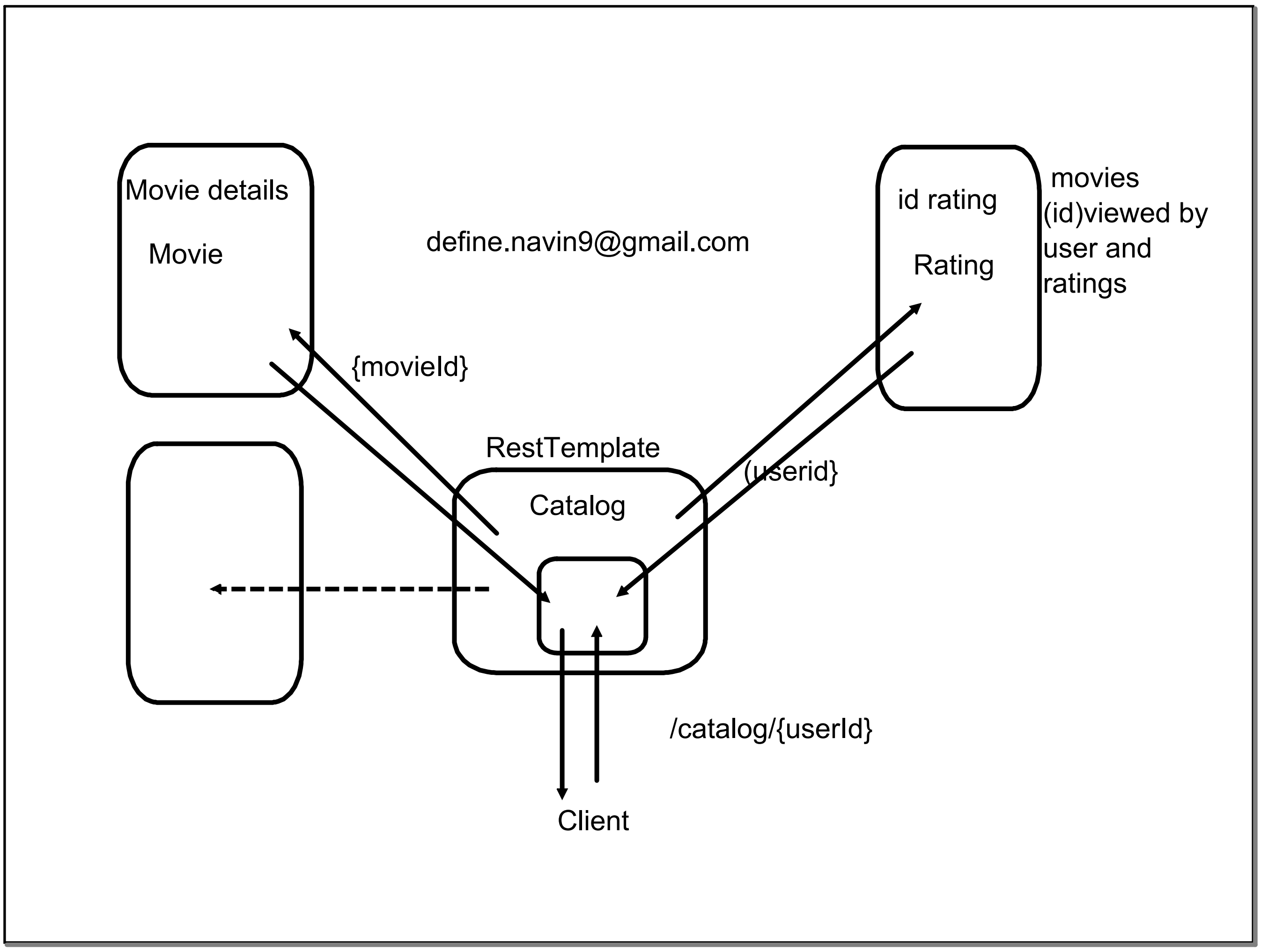

Single interaction point
Single Large Resource at one point
Database



1. Not bounded to technology
2. Easy Debugging
3. Distributed running
4. Single module will not effect complete app
5. Spin up new instance when required

Agile Methodology

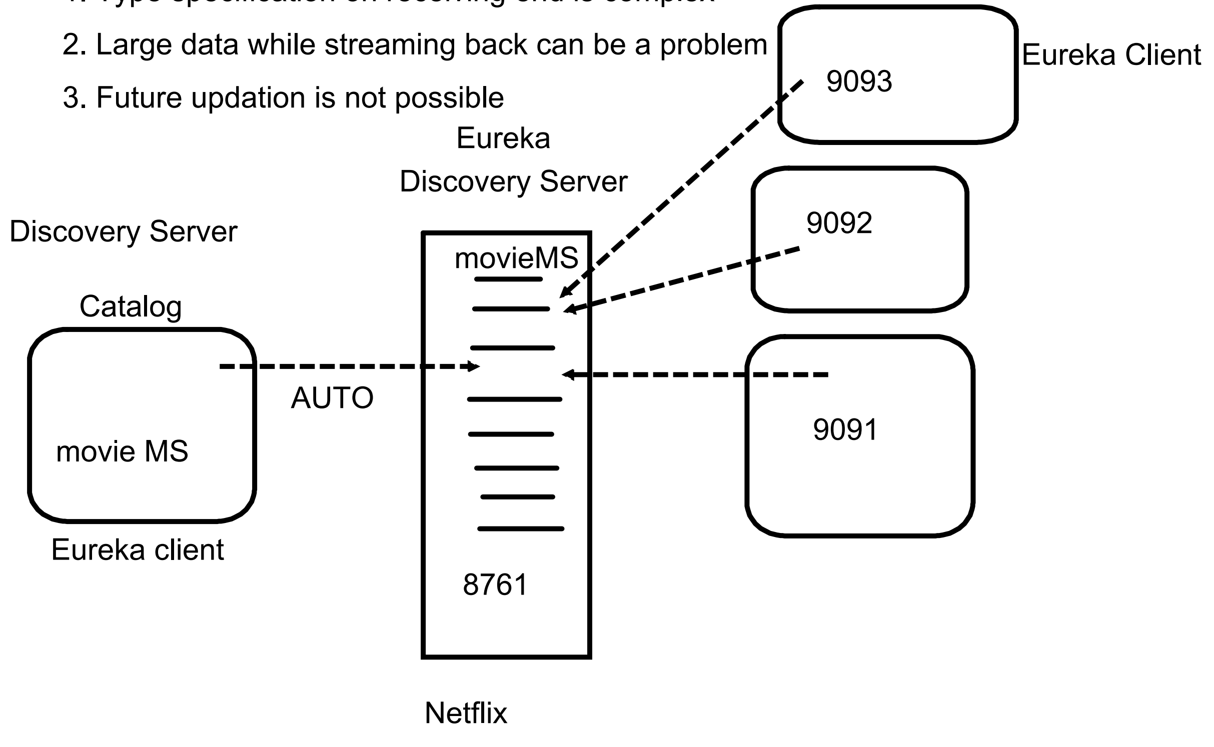

BL
Plumbing Layer
CRUD
CRUD
DB
DB
DB


Movie details
Movie
define.navin9@gmail.com
id rating
Rating
movies (id)viewed by user and ratings
{movieId}
RestTemplate
Catalog
{userid}
/catalog/{userId}
Client

Collection

Don't return a collection

1. Type specification on receiving end is complex
2. Large data while streaming back can be a problem
3. Future updation is not possible



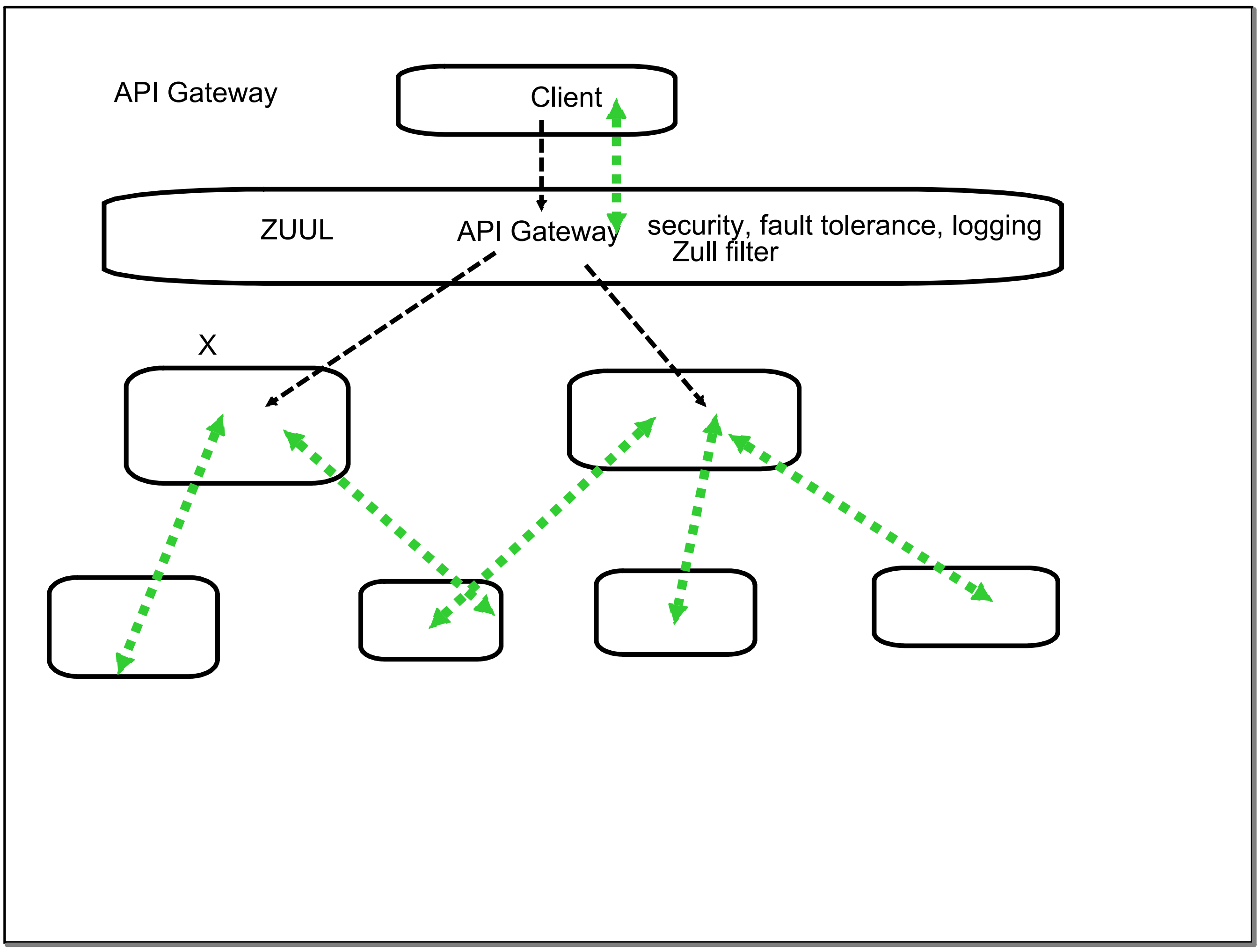
REST Template

Feign Client (Netflix)

1. Connect an exclusive Load balancing Tool with FeignClient : Ribbon (Configurable)
2. no more use of url

Fiegn Proxy interface

1 Interface for each ms that we want to talk with


API Gateway
Client
ZUUL
API Gateway
security, fault tolerance, logging
Zull filter
X

Docker

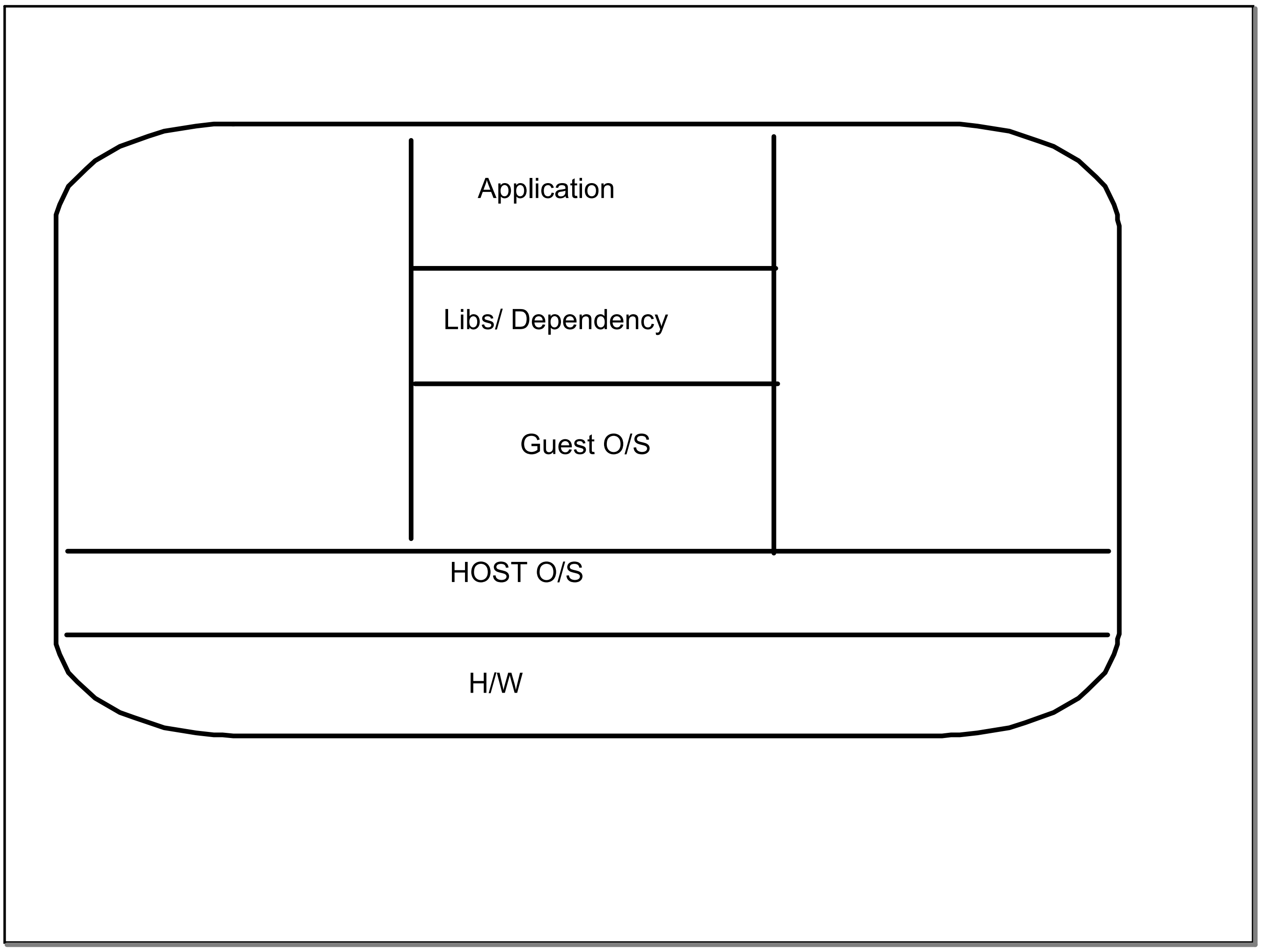
Problem

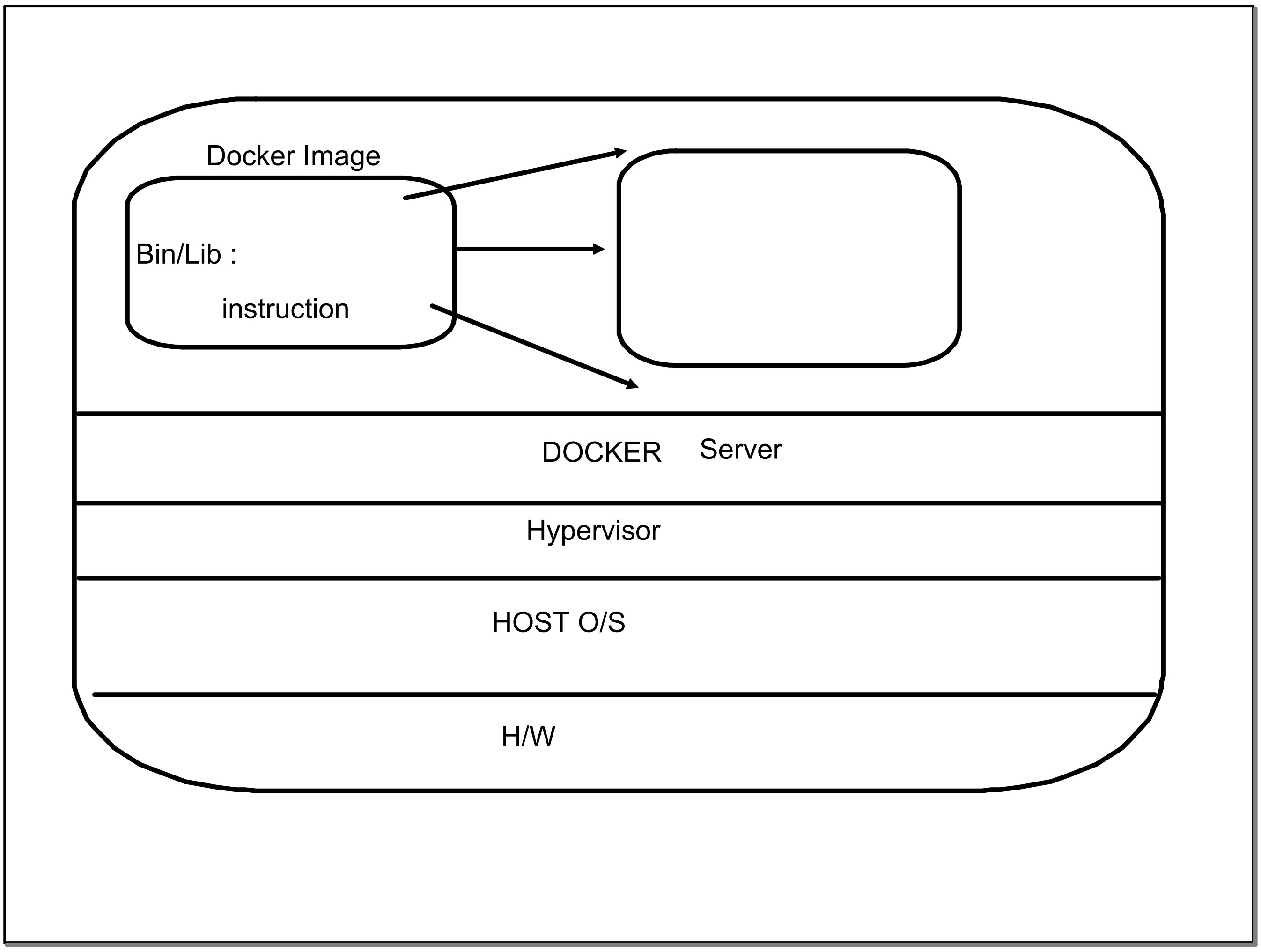
"Application runs on my machine"

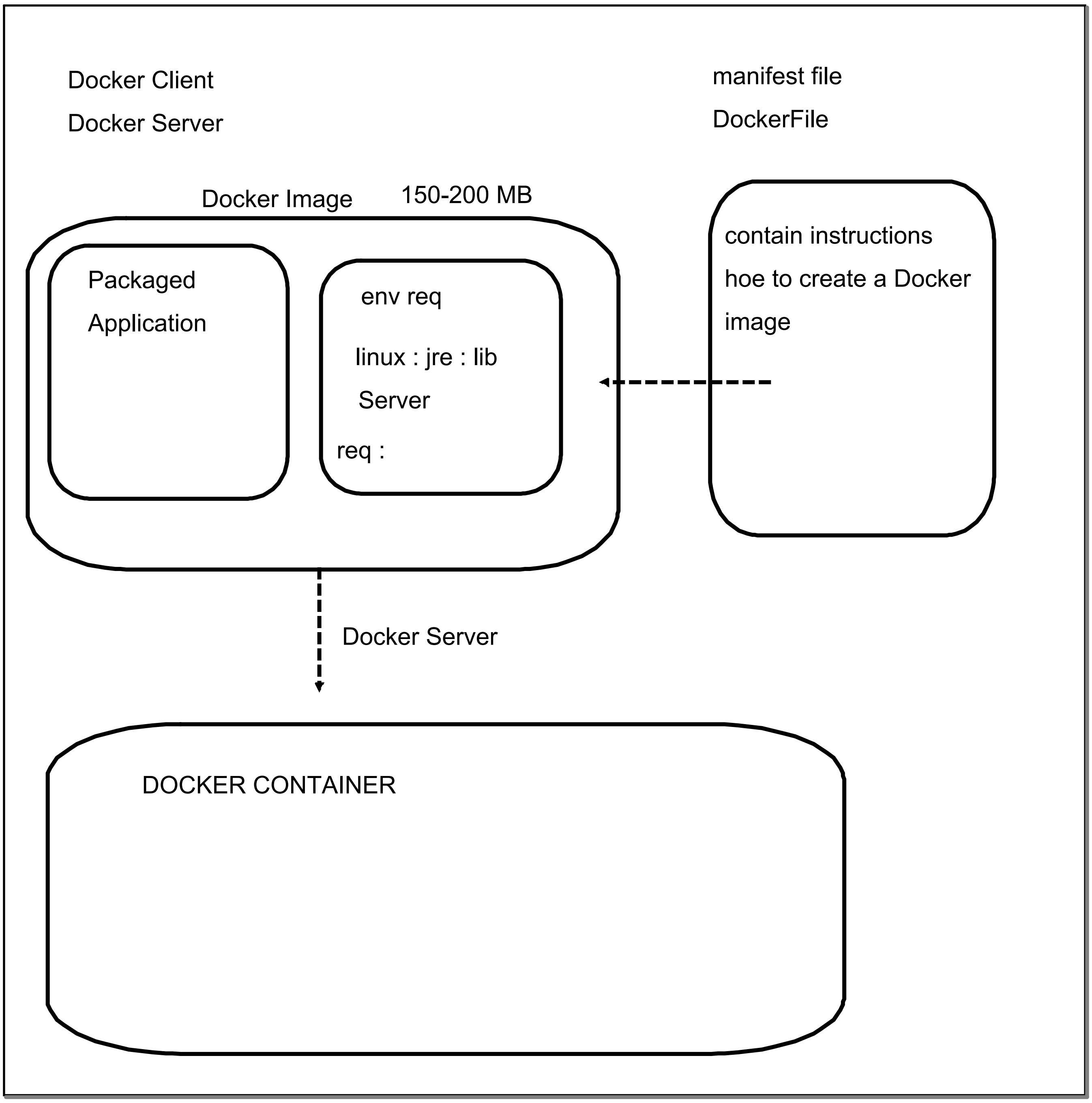
Agile style CI/CD

Dedicated

Virtual Machine Vs Container


Application
Libs/ Dependency
Guest O/S
HOST O/S
H/W


Docker Image
Bin/Lib :
instruction
DOCKER Server
Hypervisor
HOST O/S
H/W


Docker Client
Docker Server
manifest file
DockerFile
Docker Image
150-200 MB
Packaged Application
env req
linux : jre : lib
Server
req :
contain instructions hoe to create a Docker image
Docker Server
DOCKER CONTAINER

```
docker build -t <image name> <location of Dockerfile>
docker build -t naming-server .
```

```
docker container run -d  -p <new port> : <internal port number> <image name>
```

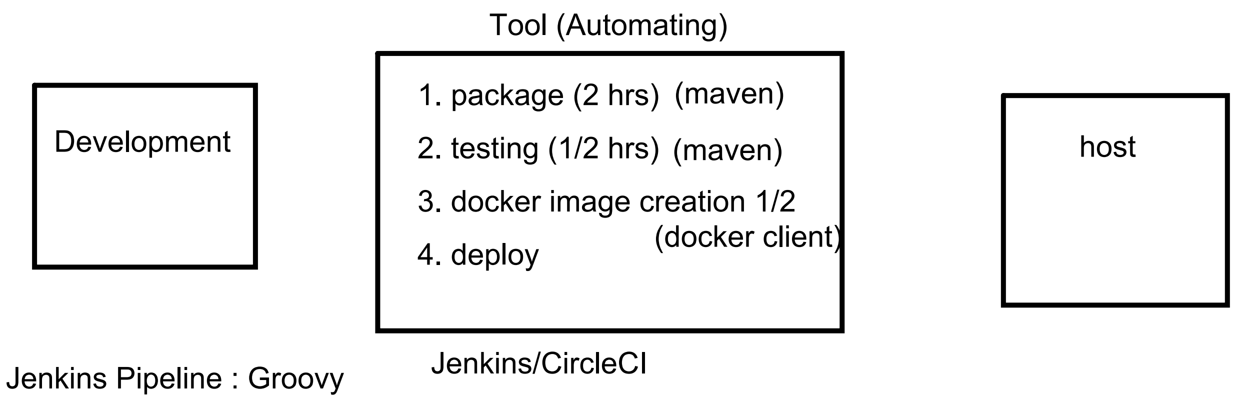
-d : detached mode | -p port mapping

docker images : lists all docker images

docker container ls : lists all the running containers

Agile process of development : CI/CD

Continous Integration & Deployment


Tool (Automating)
Development
1. package (2 hrs) (maven)
2. testing (1/2 hrs) (maven)
3. docker image creation 1/2
(docker client)
4. deploy
host
Jenkins Pipeline : Groovy
Jenkins/CircleCI

Jenkins

1. Create a new build job : FreeStyle project
2. prelims : Global Env (connect with local res or install auto)
    1. JDK
    2. Maven
    3. Git
3. configure the git location of project : Jenkins will fetch code auto...
4. to schedule the build
5. specify what to do in build job

https://
jenkins.io/doc/

Test :

Junit Test report generator

Code Coverage : JaCoCo