# This project developed from 3 part:

1. Configuration
2. Test
3. Source-Code

* Configuration

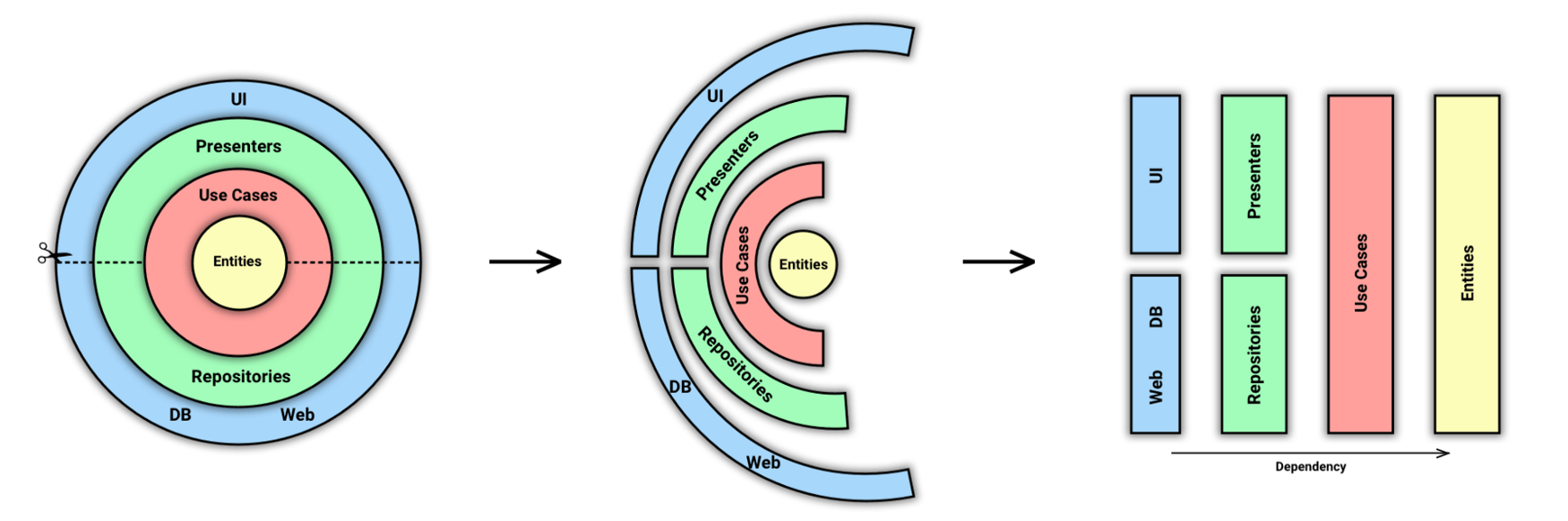
This section is for setup our key/value parameters as development mode (Dev, Test, Staging, and Production) as static files  
  
- How can do it in a better way؟  
Based on my experience, I suggest the environment variable as the best and safest way to adjust project private configuration parameters. Like Database connection info, Secret keys etc.  
Because we should develop and publish source-code without any private key as a developer.

* Test

This section is for setup project tests: Unit and Integration

I wrote many Unit tests as same as all files in the src folder and write Mock class for injection to the constructor of the class was a lovely challenge for me.

When I writing unit tests, I improve classes at the same time, change these to a unit class this meant class should not be dependent on a specific service or repository.  
  
I prefer Mocha between Jest and Mocha, because Mocha has a simple configuration to set up on your project to write your first test. However, I wanted to show my skills.

**Project Architecture: (Onion + Hexagonal)**   
*Testability*. All code must be isolated if you are following correctly this pattern, so everything can be mocked.  
*I know it did not implemented completely in the challenge project. Nevertheless, I tried doing best.*

* Source Codes

Please let me explain it folder by folder:

1. **common:**  
    1.1. *baseError.js:*  
    - this file have several classes extends from Error java script to customize log and API response message

1. **constants:**
   1. *APIVersions.js*  
      I planned API versioning as a static object. Because with this object I can manage version value in all API Endpoints easier and my mistake percentage will be near zero.
   2. *MachineStatusList.js*A response object should be readable by the user (such as Front-End Developers or Third-party Apps).

Because of that, I change all of the variables like Status to a string value when mapped DbModel to ViewModel

1. **controller:**

They have 3 main responsibilities:

Extract the parameters (query or body) from the request

Call the good Use Case (application/presentation layer)

Return an HTTP response (with status code and serialized data)

* 1. *defectController.js*  
     Should just control request and response in this class by several specific functions  
     in this layer should use try catch to catch error from nested layer and by next() function (ExpressJs features) process error by a Middleware to Log and send a friendly response as RestAPI

1. **presentation**
2. **domain**

5.1. data

I develop a class with name DbManager.js to manage all general postgresql (SQL) transaction (CRUD)

5.2. models (use Cases)  
 queryParams => some class defined from Services for send to Repositories as Insert/update operate  
 queryResults => some class defined from Repositories for return to Services

5.3. repositories

….Repository.js => defined all CRUD database transaction also complex query like Inner join

Improvement:  
Planned a layer (Repository Adapter) above Data (DbManager.js) to support several kind of Database (like: SQL or NoSQL) with different ORM (like: Mongoose 🡪 noSql: MongoDB, Sequelize 🡪 sql)  
- RepositoryAdapterMachineMongo.js  
- RepositoryAdapterMachineSequelize.js

My Questions:   
  
Do you have Project Manager? To manage Tasks and Project with milestone and work with Task Manager Tools like Jira, Clickup?  
  
What is a flow to develop a new feature in your technical team? Like: from Sale to Technical and ready to production  
  
Do you have Dev, Staging, Production?  
  
Do you follow Agile?  
  
What does your company offer Benefits to me when I relocate to Berlin?  
  
Do you have any problem if I want to work remotely?

What is your problem if I want to work remotely?  
  
Your company offered me 50 to 55K euros gross per year.

Can you tell me how much tax deducted from the salary?