Agile

JIRA

End to end process –SDLC

-Product Owner

-Product Backlog

-UserStory

1. Client / Business some requirement.
2. Explain to product owner
3. Product Owners (PO . Product) – understands the requirement from client. Write it as UserStories and puts it into JIRA.
4. PO has to explain these requirements to tech team : Gromming Meeting / refinement meeting / requirement gathering meeting .

Microsoft Teams – chatting and meeting purpose

Outlook – emails

Setup a meeting requests meeting in Microsoft teams

While writing requirements 🡪

1.Userstory :

As a <user>

I want <what>

So that < why>

2.Accemptance Criteria:

(you have to write in bullet points)

-Show link of forget password on login page.

-when user clinks link, open a popup and ask for email.

-validate whether email existsting in system.

-If email invalid , show error message to user, that email doesnot existing.

-rest password flow

3. UI “mockups”

UI/UX developer

CSS/bootstrap

4.

If 70%-80% clear, then development do but if it is not clear then donot start pogramming and the terminology is called DOR (Defination of ready).

5.Ask for estimation (how much time it will take to complete requirement)

2-3 days

Ive us estimates in the form of “Story Point”

1 2 3 5 8 13 …..

1.Time

i. How much UI guy will take time. ( eg. 2 days + 0.5days unitesting = 2.5 days )

ii. how much Server guy will take time. (eg. 3 days + 1 day unitesting = 4 days )

iii. How much testing guy will take time . (eg. 1.5 tescases + 2 for testing = 3.5days)

total 10 days/ 2 = 5

2.Complexity

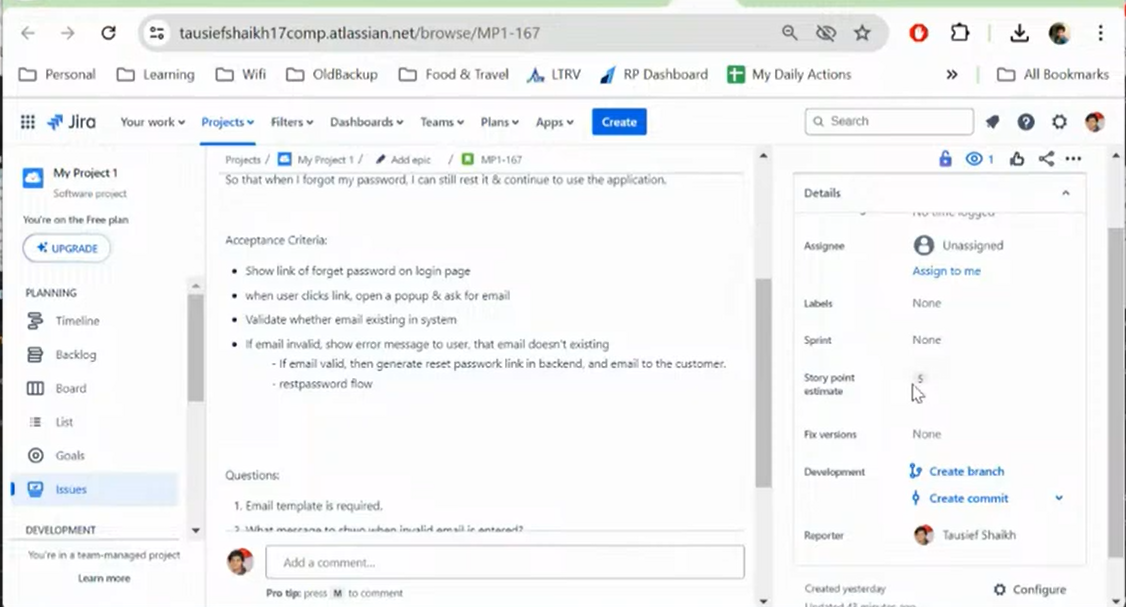
0, 0.5, 1, 1.5,2 = average is 1

3.Risk - 0

Total = 5 + 1 + 0 = 6

Now from fibonaci seriese closes to 6 is 5 so story point for the above estimation is 5.

Without analysis, do not give estimation.



1. Internal grooming meeting
   1. Discuss these requirements internally.
   2. High level, how dyou want to implement this?
      1. High level design
      2. Microservice – in which service , what code to be written
      3. Open sources
      4. DB design
      5. API design
      6. Performance
      7. Security
      8. Existing reusable logic

7. For all requiremetns, you would get 100% clarity of how to implement & storypoint

8. IN 2 weeks, how much work you can finish.

Sprint duration: 2 weeks.

Sprint Planning Meeting (Day1 of the sprint)

Sprint1:

Start: 8Jul

End: 19Jul

How much work we can finish?

Who is going to work on what features

UI - Java - test

2 ui - 3 java - 2 qa

Planned Leaves

Again ask for clarification of requirement

Mon-fri, 8/9 working

Subtasks

UI task, server task, qa testcase, qa execution

Task Breakout

About to implement feature.

Day1 of sprint:

Team is clear, what features needs to be developed in upcoming 2 weeks.

Who is going to work, on what features is clear.

ANyone on leave is clear.

Highlevel implementation approach is also clear.

Refect the list of features for current sprint, in Jira.

"Start Sprint"

ON daily basic, any progress happening in application development, should be reflected into Jira.

Update Jira status, to refelect the current state of work.

TODO | INPROGRESS | QA | DONE

9. About to work!!!!

Spring Boot & Microservice Project.

Core Java + JDBC + Basic of Web(Servlet/server)

-----------------------------

Core Java

Sprint Boot -

WebApplication (Something which can be accessed across Internet)

Browser you can access it.

https://roadmap.sh/java

Multiple frameworks for

Core java + Web framework = Web application

=====

Core Java

+

"Spring Boot"

=====

Where to find all these jar??

5-10 jars

Business

Software

Coding & deliver

1. Core Java + SPring Jars = Web application

2. Searching these jars are not possible manually..

Build Tool: "Maven"

3. Customer want software as quickly as possible.

Without even knowing spring, you can build your first application.

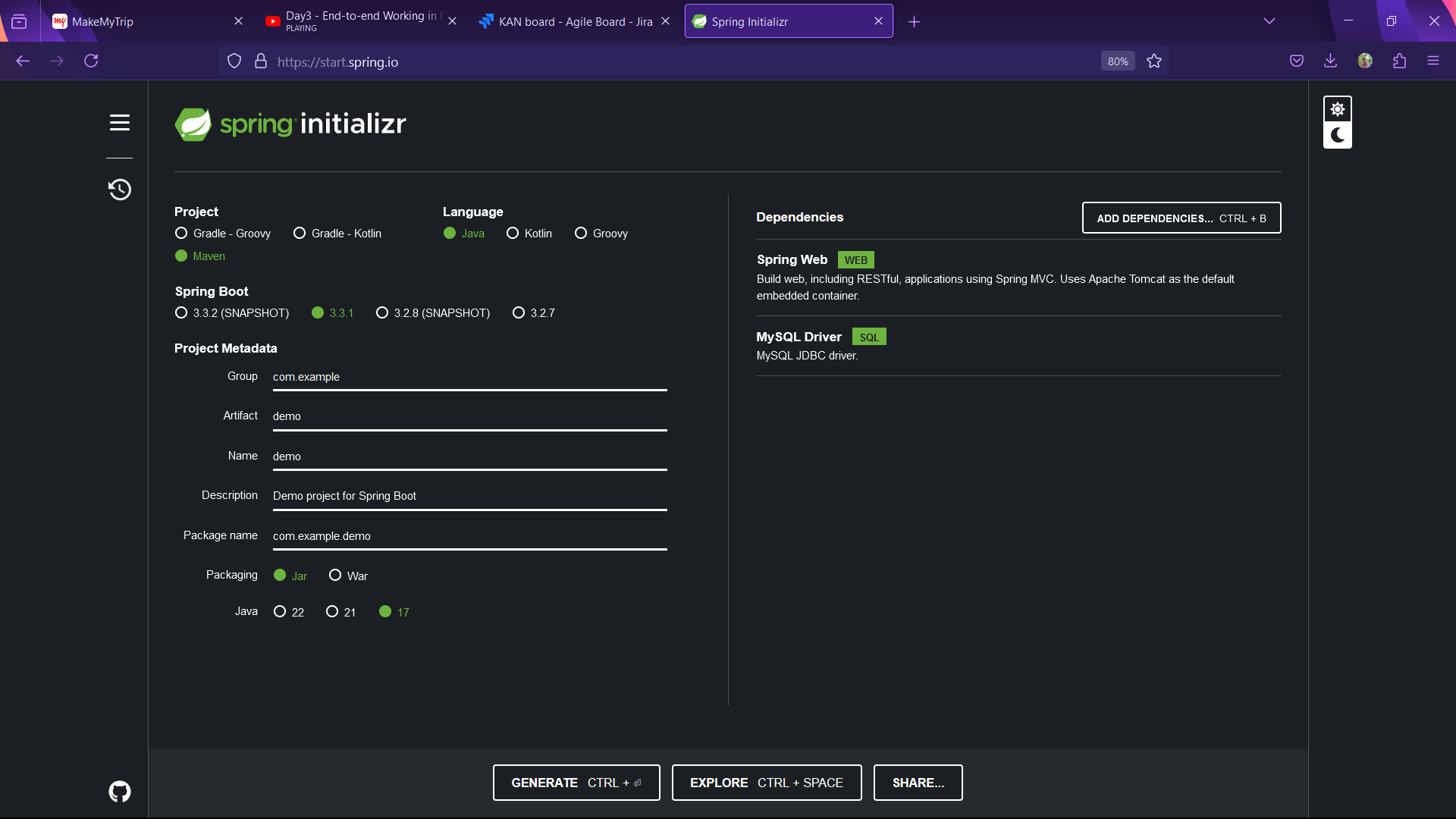
**https://start.spring.io/**

And fill maven specific details 🡪

Group: which company developed this code?

website: google.com

Group: com.google



Artifact: End output of your project.

A.java

javac

java

For web application - BuildTool

1. Develop

2. Compile

3. Execute (Server: Tomcat)

jar

war

Generate:

Create your 1st Spring Boot application with web dependency.

1. Got the project from start.spring.io

2. COding on this project

IDE: Eclipse,...., Spring STS

https://spring.io/tools

3. Get the project (which we generated from start.spring.io) & include in your IDE.

Import => Existing Maven Project

All Maven configurations: pom.xml

Maven:

Build tool

Dependencies for us

pom.xml

group, artifact

maven commands

-----

if you install maven manually in laptop

mvn clean

mvn install - gets the newly added dependencies in your projet

mvn package - builds archive file

PATH .. exe, maven/java

com

IDE,

java default installed

Maven default installed in IDE.

Version

run using IDE way:

clean

install - gets the newly added dependencies in your projet

package - builds archive file

javac A.java

java A

a.jar

mvn package

"Executable Jar files"

java -jar a.jar

Core Java into a SPring Boot.

Web application

You can build a standalone application using spring/springboot

Web application - additional dependency

**Date : 15/07/2024**

**In week2 we will do ->**

* Stripe integration
* Domain Knoladge
* Architecture knowledge ( application + infrastructures )
* Project development standsrds.
* Spring and spring boot
* restAPI

In weet3 we will do

* 3 sprints
* Functional daily requirement discussion and implementing.
* And there are total of 7 weeks.

**Stripe Integration :🡪**

* Payment team ( Integrate all payment into the system)
* There are different payment system like PayPal , CC, DB, Stripe, Neteller.
* Client wants to have payment capability.
* Payment integration would be developed by payment team(US).
* Client wants Stripe PSP Payment Integration. (Europe and US).

Tech Team wants to integrate with any PSP –

**API Integration Document:🡪**

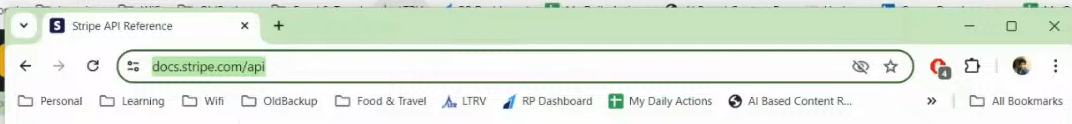
* API integration document. (Two pieces of software that want to communicate is called API ie Application Programming Interface). API document helps tech team, to build the software to do the coading. Api document is in the form of word, pdf etc. This will given by the client with different requirements.

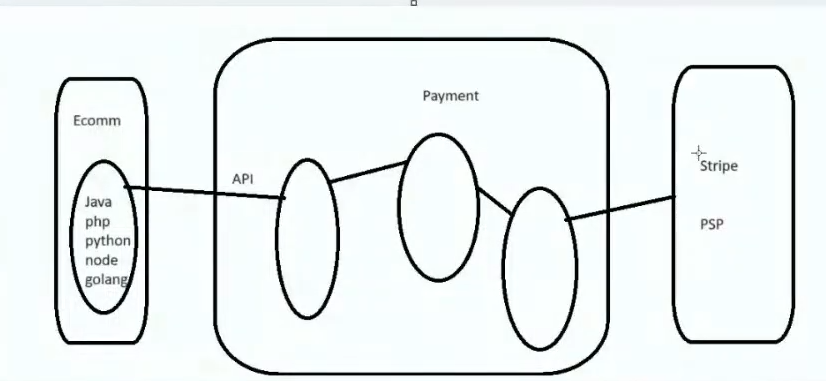
We are going to break down the whole project into small small projects by using microservice architecture.

Client will decide which payment provider to choose.

Product Owner starts doing some initial analysis.

Now since we are going to stipe so go the docs.stripes.com





Before giving the API documentation the peoduct requirements and for that requirements how much time required and price all these nigotiations are taken care by Product Owner and BDM(Business Development Manager)

Steps will be like 🡪

* Client will decide which payment provider to choose .
* BDM
* Product Owner start doing initial analysis
* Then atlast product owner will give the documentation to the technical lead.
* Now everyone will go through the documentation and analysis will done.

**a**

**Date 16/07/2024**

**Payment Domain and Architecture**

Highly responsive system should contain following points 🡪

1. You can not double process same payment 2 times.
2. Any kind of failure should not risk the state of the payment. System should always stable , network failure, exception.
3. Security

**Payment Method** : Card, APM(Alternate Payment Methods)

APM: Stripe, Trustly, Paypal, etc ( Payment Service Providers) these are third party softwares.

Even for card processing: Integrate with 3rd parties.

Why did you not guys not build rather integrating with others?

* Because of some government regulation, licensing, auding etc we have to manage So to not do these things we use alternate payment method (APM).

Any company which accepts and stores processes direct card numbers has to be PCI DSS (Payment Card Industry Data Security Standard) compliant.

Loads and performance are considered while building technical solutions.

PaymentType : **SALE**  - One time payment.

RECC - Subscription

REFUND –

In out project we are going to do **hosted page** payment implementation, SALE is our payment type, APM is our payment method , stripe is Provider.

Df

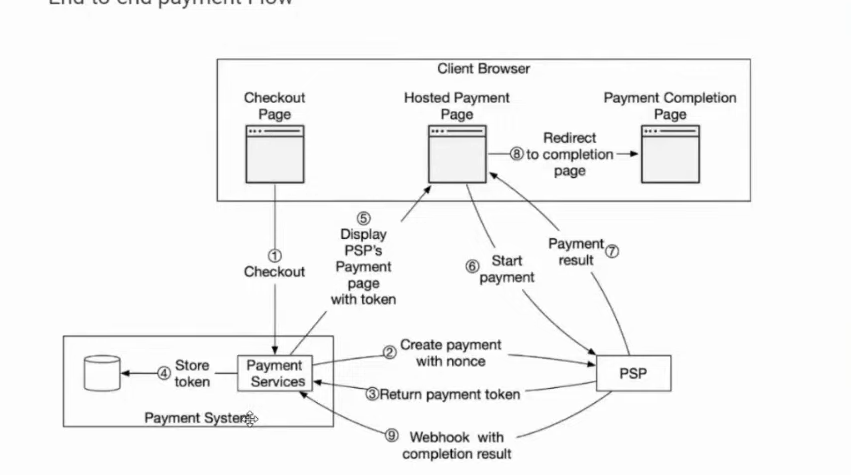
How basic money movements happends ->



There are two terminologies 🡪 1. Pay-in 2. Pay-out

**Functional Flow of the project**

1. End customer comes to the checkout page that he sees stripe as a option. Then he select the stripe as payment platform.
2. Then checkout will happen and he redirect to the payment system.



1. Now we have to write code in the payment system to talk to PSP ( Payment Service Provider). Yesterday we done call via postman API but now we will do it with java code .
2. In return PSP will get an unique id of that payment and a redirection url.
3. Now in payment system store the unique id in database and the url that we got send it to the hosted page in client.
4. Now there is direct connection between client and PSP where payment will start and payment result is shown and client will redirect to the payment successful page .
5. Now since payment is successful so PSP will send payment system a web hook with completion result.

**Tell me about your project 🡪**

* First tell the functional flow given above .
* Then come up to technical architecture ( how you did this).we are going to use microservice architecture.

Different architectures are –

* Monolitcs architecture.
* SOA architecture (Service Oriented Architecture)
* Microservice architecture

**Microservice**

There are 9 principles

1. **Componentization via services**

* Have 1 project in your ID “JulDemApp” and code entire logic inside same project .

Validate the payment..

Authentication and authorization.

Fraud and blacklisted.

Per day .. per person 1 lac amount..

Integrate with Stripe PSP

Integrate with PayPal PSP

Stable .. end – to – end stable payment processing logic web hooks ..

* We can not code everything in single project . Rather we will create small small projects.

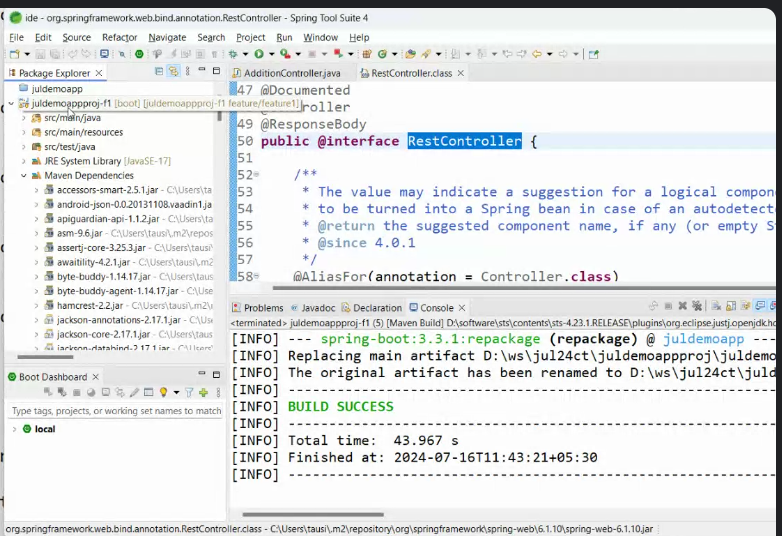
1. **Componentization via libraries:**

Every project are libraries and not executable individually. We have to process together.

* Write multiple smaller projects
* Project 1 : validation – project (All validation logic, validation.jar)
* Project 2 : Stripe Integration ( stripe-int.jar)
* Project 3 : Payment Integration logic (paypal-int.jar)
* Project 4 : Payment – processing project + web hooks.(processing.jar)
* Project 5 : wrapper project 5 . => Executes..
  + Validation.jar, stripe-int.jar, paypal-int.jar, processing.jar

1. **Componentization via services :**

Each project should execute independently .. without needing any other project… “service”

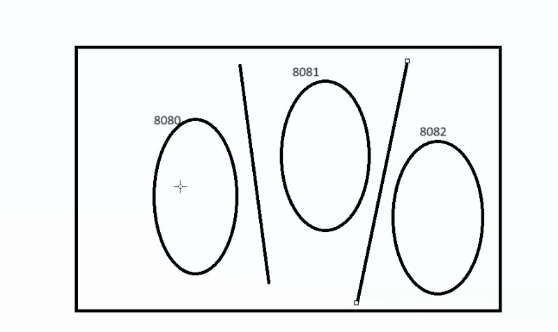


We will execute like

java –jar validation.jar

Java –jar processing.jar …… similarly others.

Here runs on different port 8080 , 8081, 8082

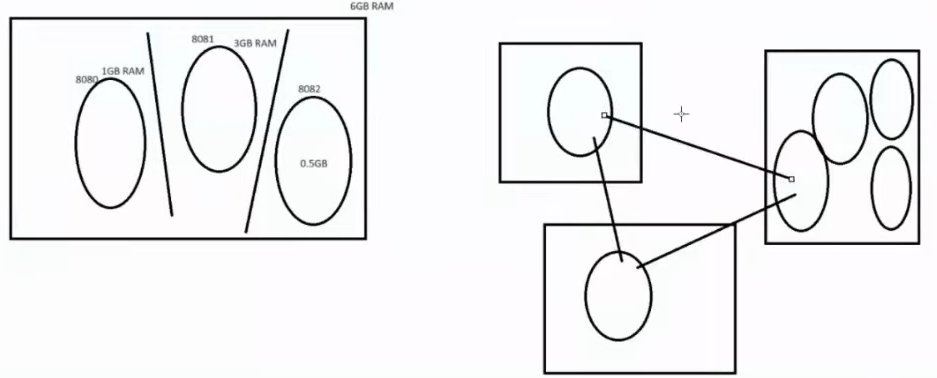


Different process, port, memory . But via libraries same port is taken and same process.

Every services can be coded independently .. compiled independently… started independently … can have their own memory size ... occupies separate port .. have its own process id …

* Can be independently restarted, without impacting, no downtime for other services…
* Can be deployed in any place as needed ….
* Auto scale.

Instead of running everything in single m/c you can run on different machines.



**Date : 17/07/2024**

In Our project we have three microservices :

* **Payment validateion service**

Payment validation rules.

Authentication and authorization

Field validation

Business validation

* **Payment processing service**

1. For core payment processing
2. Payment life cycle (system should be that precise state to tell what is right and what is wrong with customer payment . beginning to end tak. Eg. Customer choose stripe for making payment . request is validated and is being processed further. Then failed. Abour to call stripe. Go url from stripe. Webhook – success/ failed).
3. Payment ( there will be payment id and can be different status – *created, validated, initialized, pending, success, fail etc)*
4. Routing decition … stripe or paypal…
5. Notification / web hook processing logic

* **Stripe provider service**

Writing stripe integration code.

Decentralized governance

The team has the power to make descitions not the government.

Team can choose latest cutting edge technologies.

lkjnl