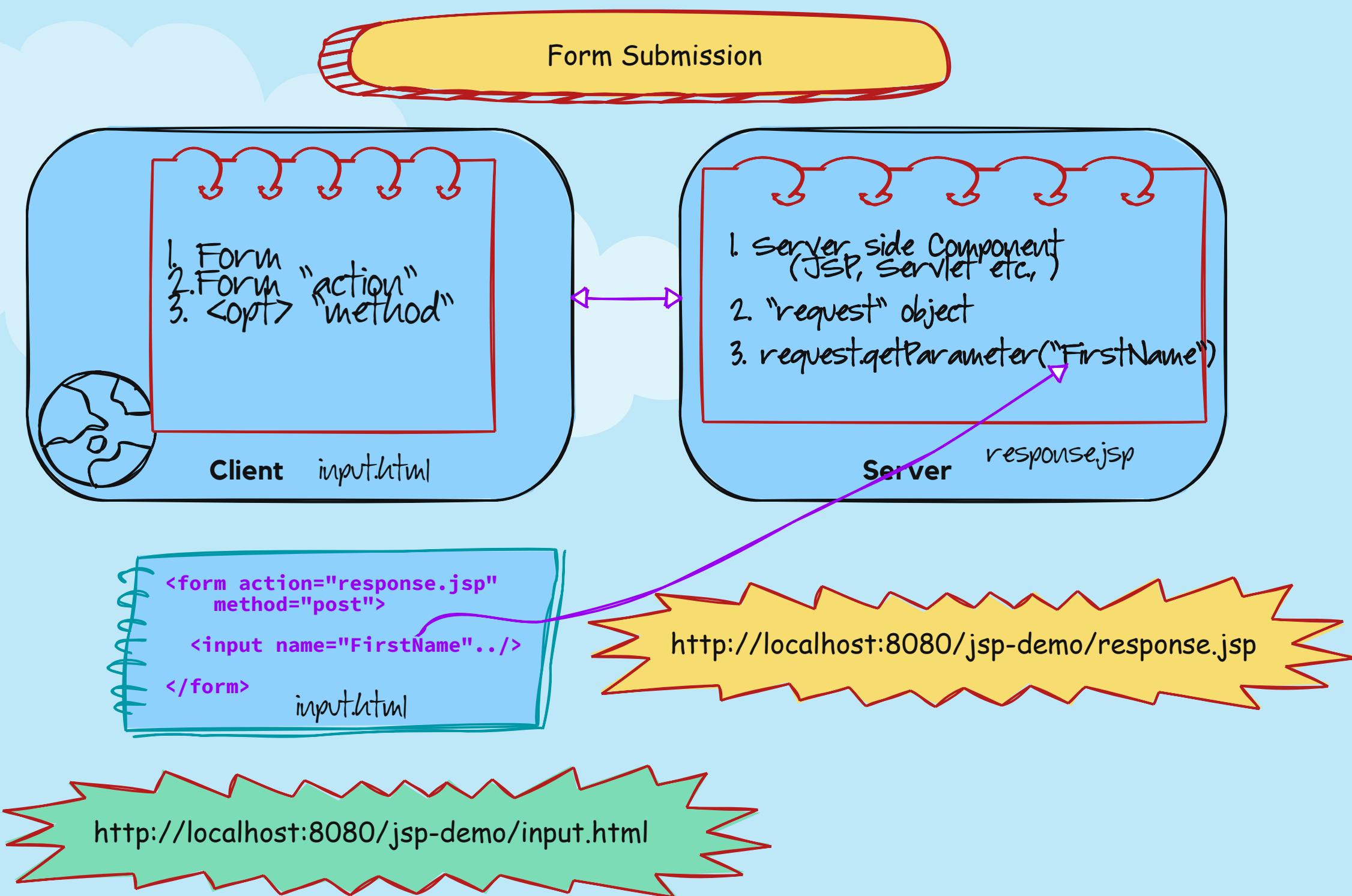
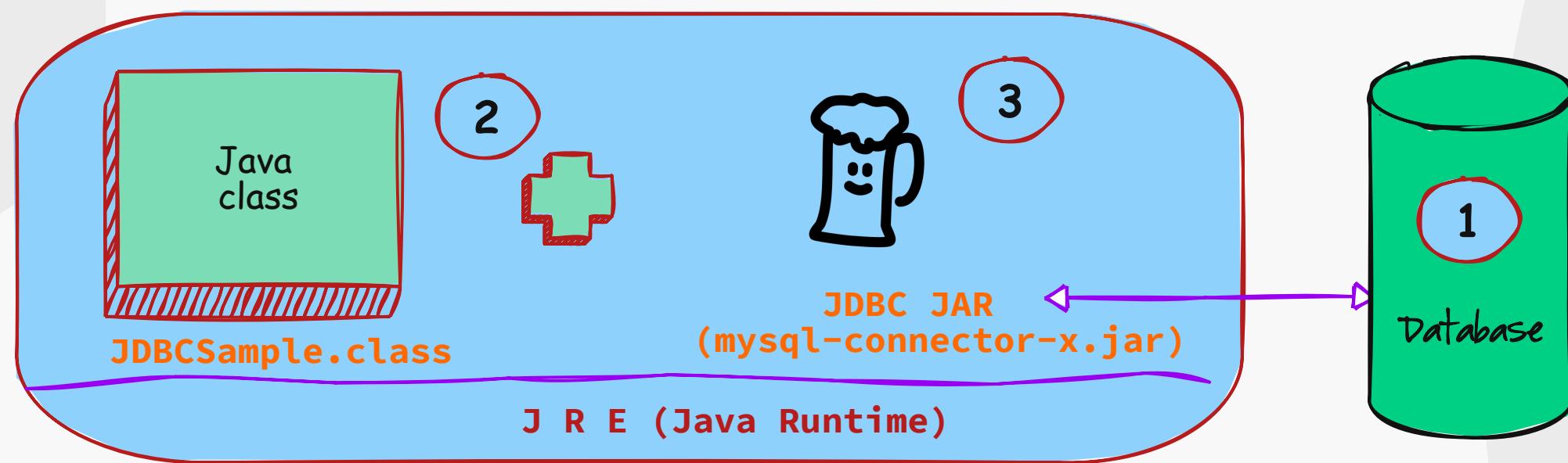


Two different applications "jsp-demo", and "jsp-tutorials" are hosted on the Server "localhost" and exposed at the port number "8080".

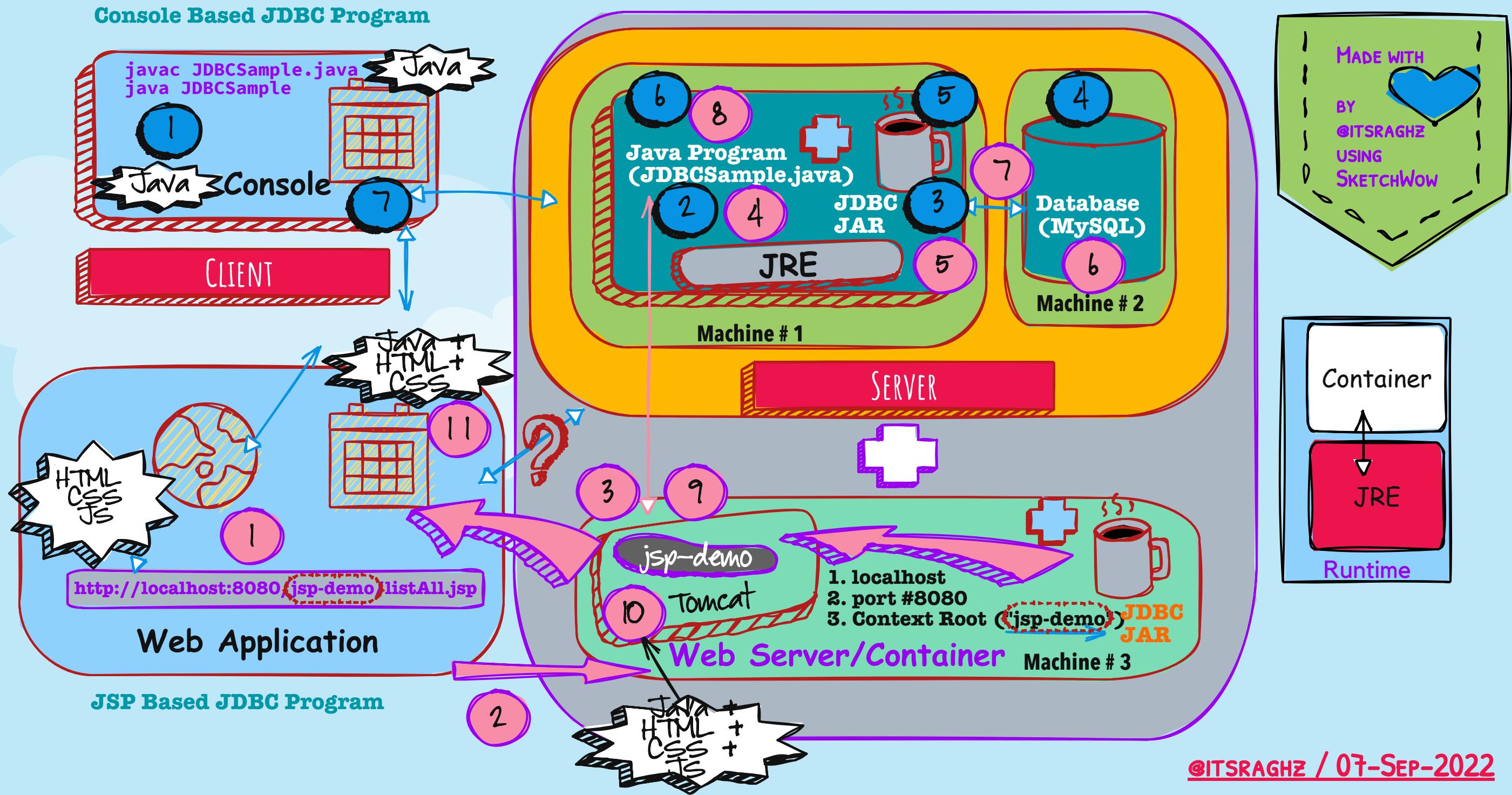
The resources being looked at each application differ. In "jsp-demo", it is "helloworld.jsp", and in "jsp-tutorials", it is "index.jsp".



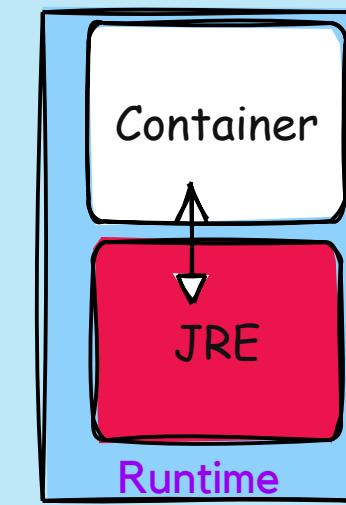


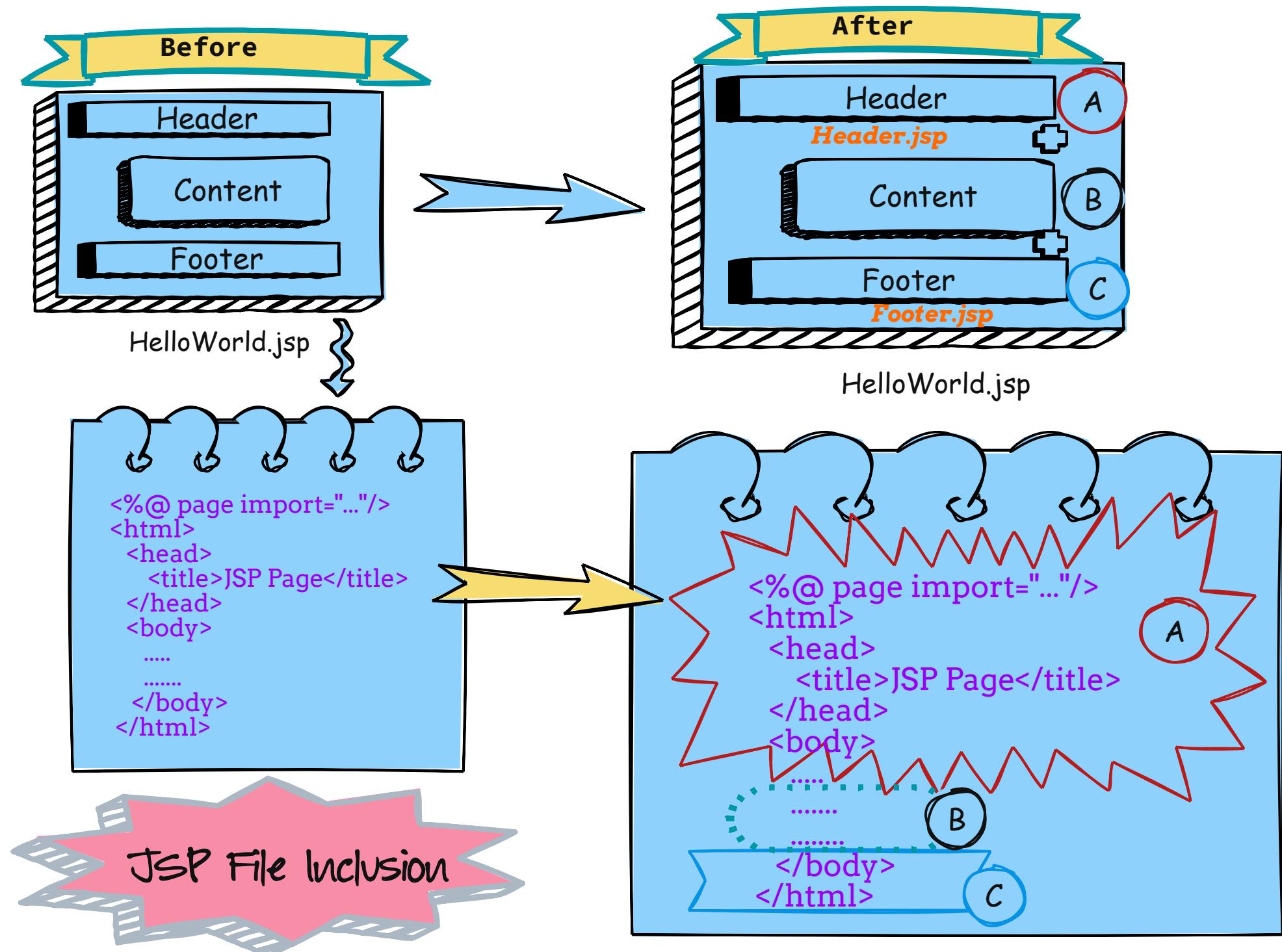
Adding the JDBC Jar into Eclipse Project

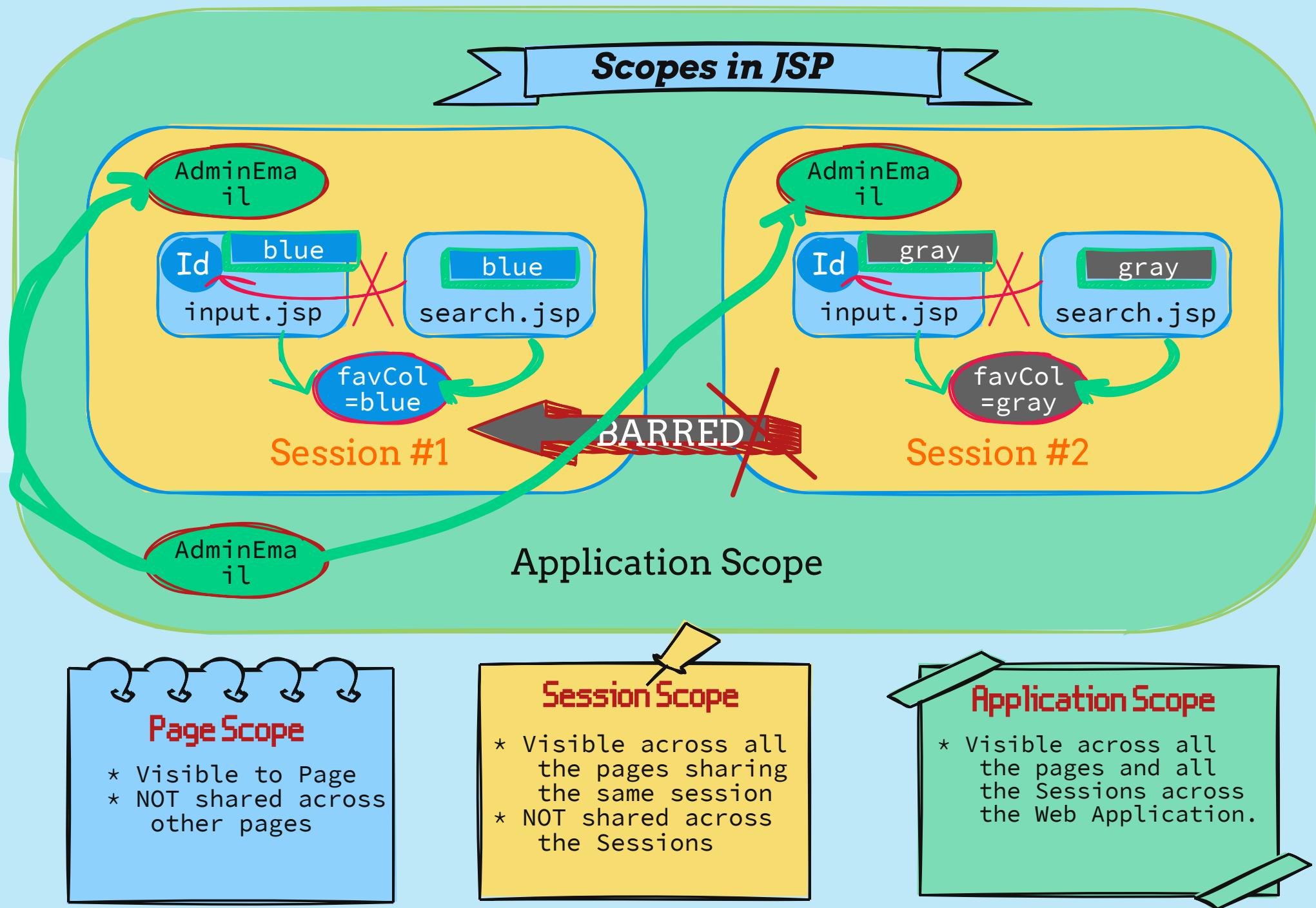
1. Right click project
2. Select Properties
3. Click the "Libraries" Tab
4. Select "Module Path"
5. Click "Add External JARs"
6. Choose the mysql-connector JAR file
7. Click Ok/Apply/Open
8. Click on "Order and Export" Tab
9. Ensure that this JAR is selected
10. Build the Project

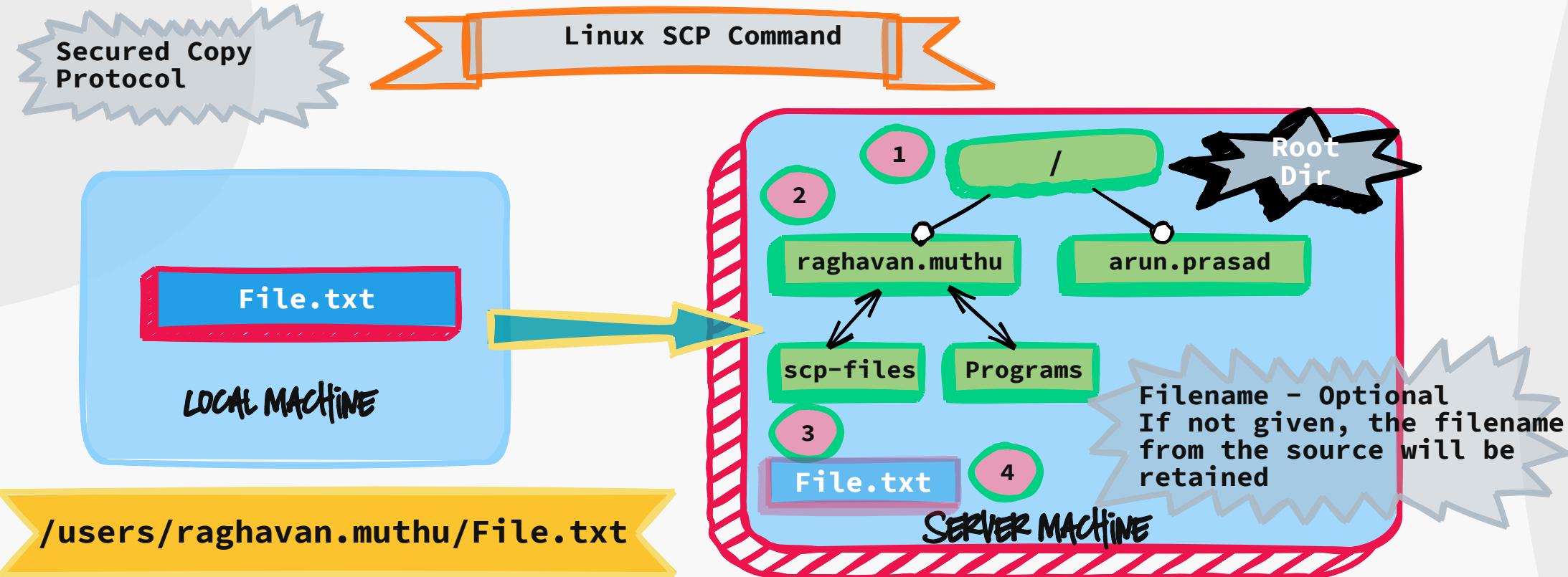


@ITSRAGHZ / 07-SEP-2022









Data Ingredients

1. User - raghavan.muthu
2. Server - 10.121.1.241
3. target Location - /home/raghavan.muthu/scp-files

cp /dir/..../source /dir/..../target

scp /local/path/to/file user@a.b.c.d:/targetPath

scp /local/path/to/file user1@a.b.c.d:/home/user1/path

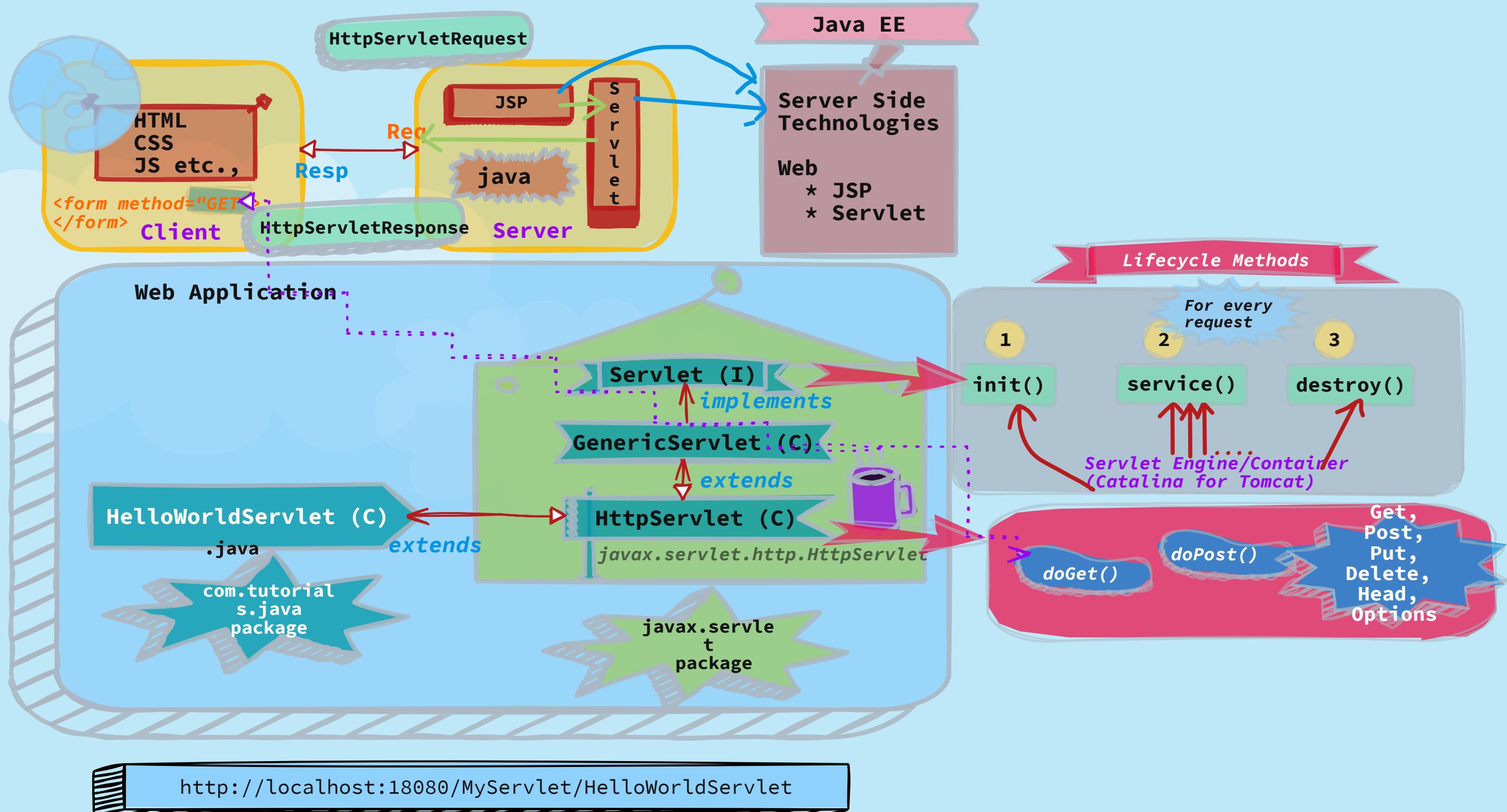
Path

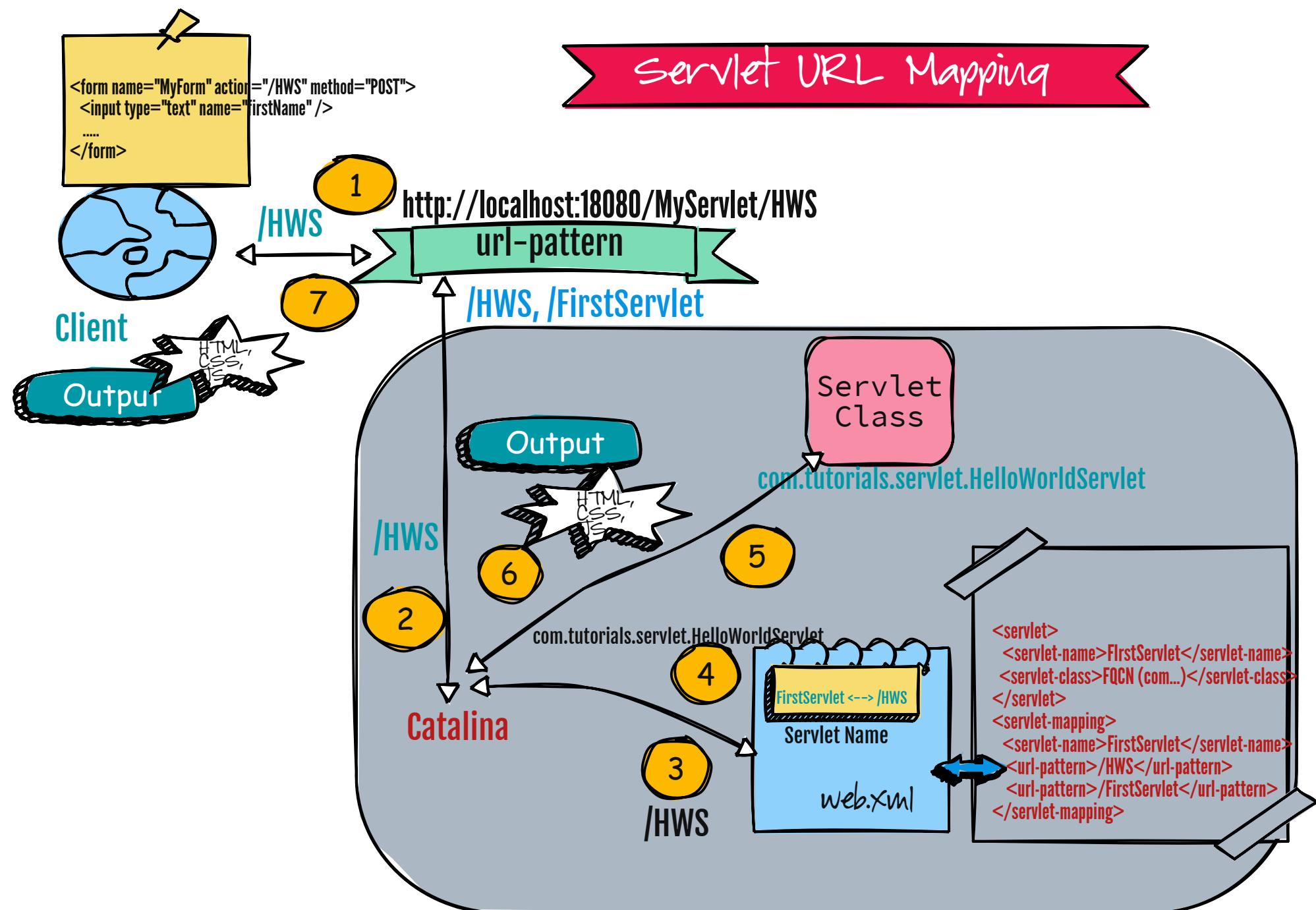
User Account
<userName>@<ServerIP>

scp /local/path/to/file user1@a.b.c.d:/home/user2/path

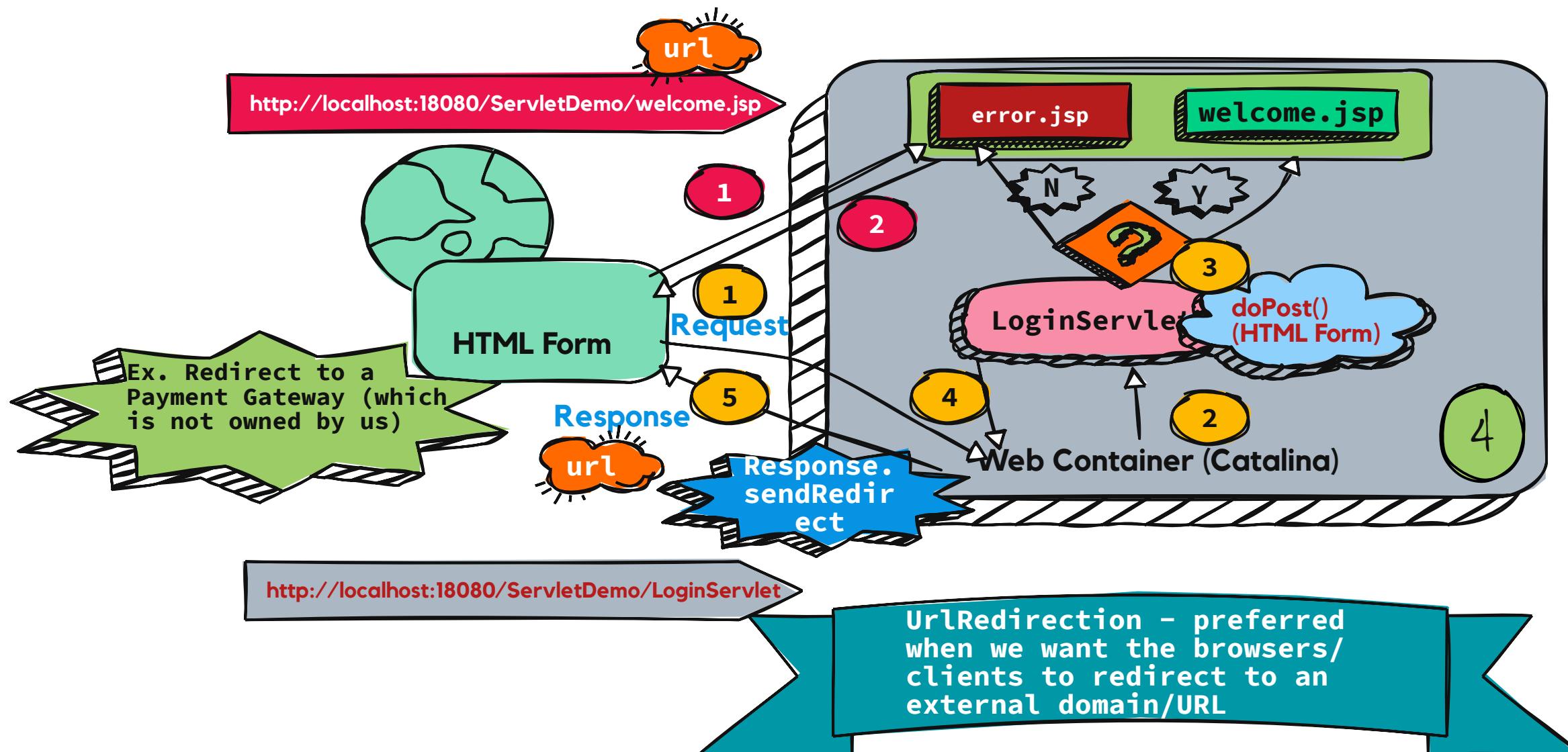
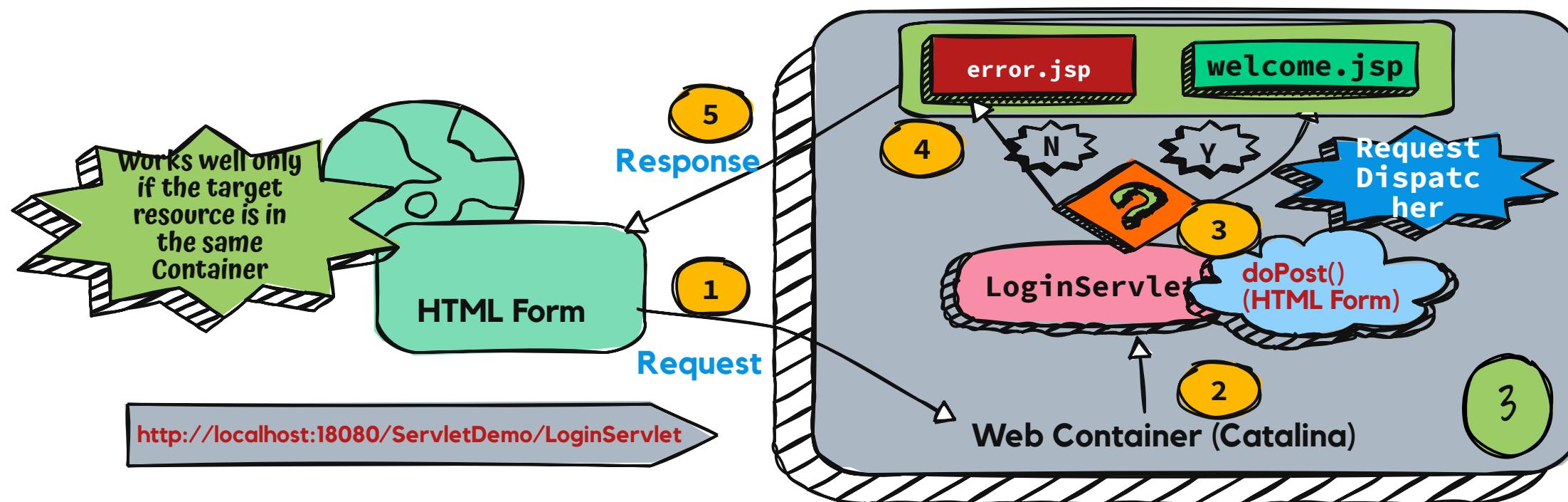
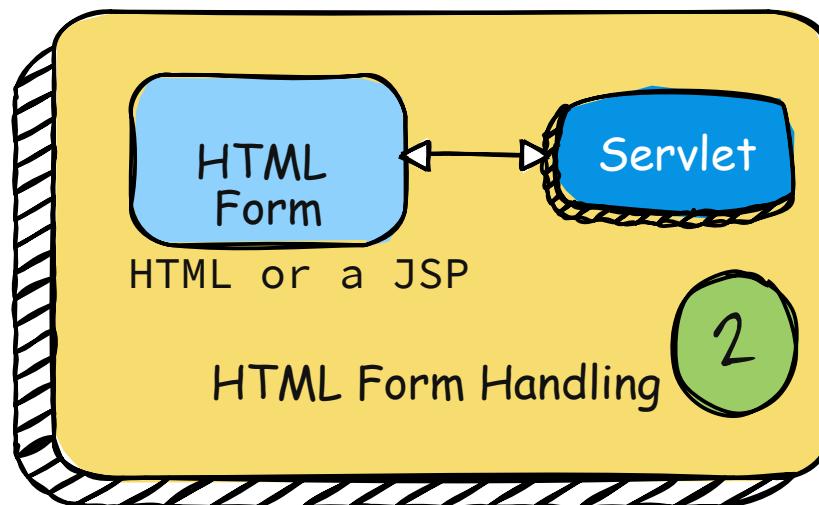
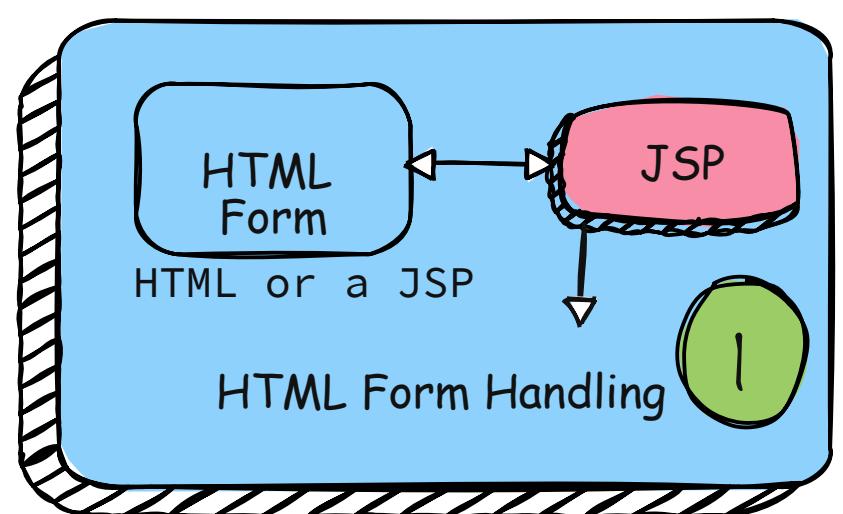
scp /local/path/to/file user2@a.b.c.d:/home/user1/path

scp /local/path/to/file user2@a.b.c.d:/home/user3/path

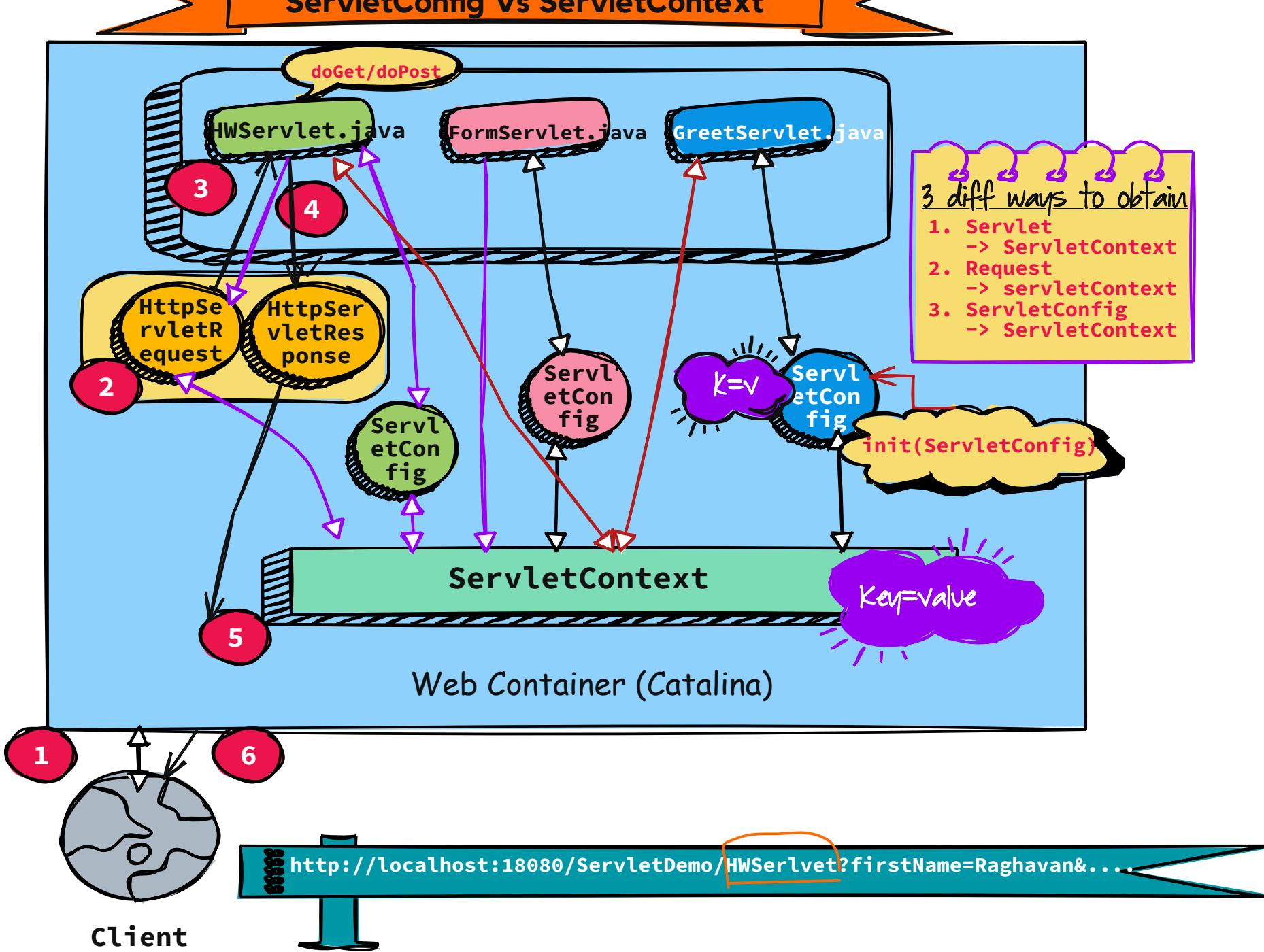


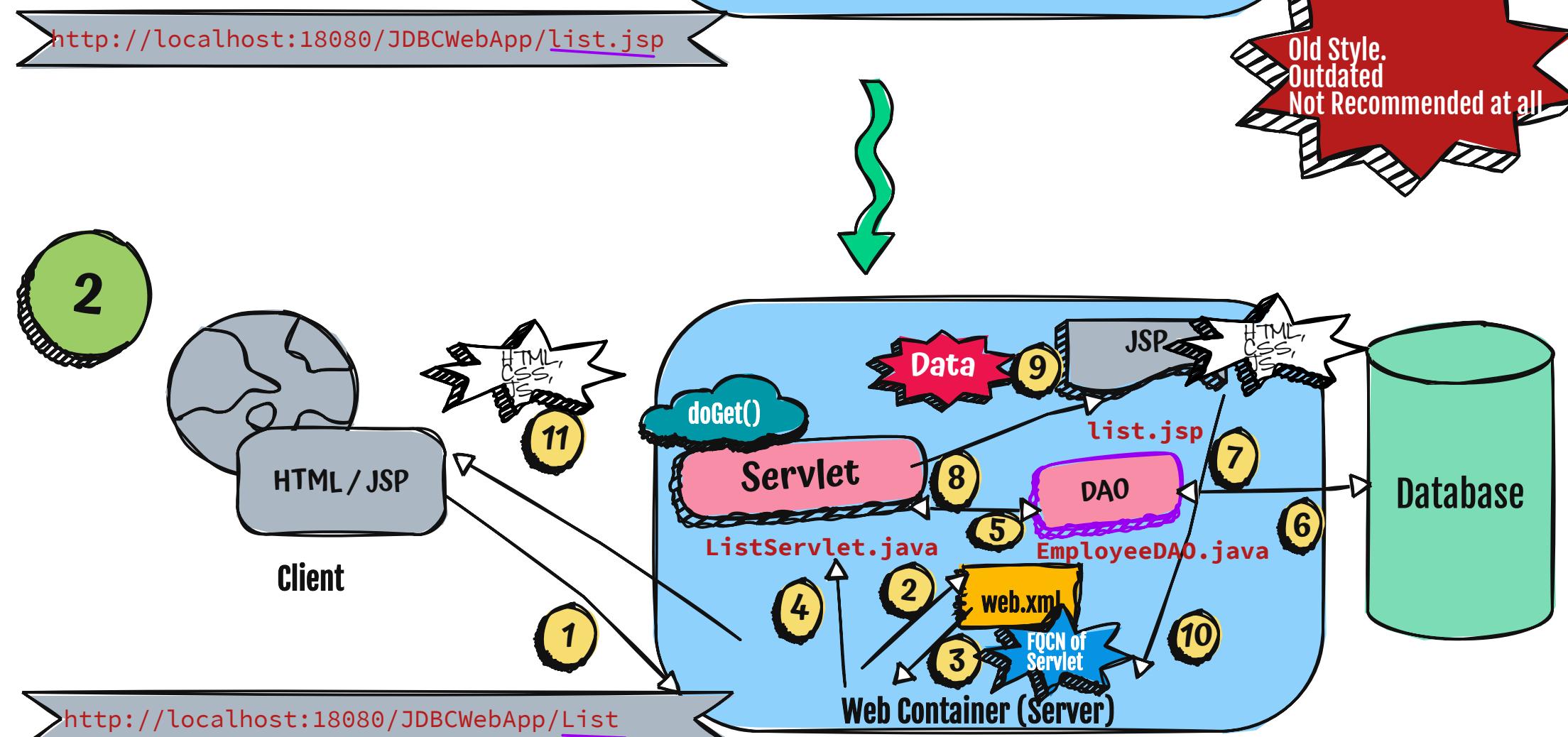
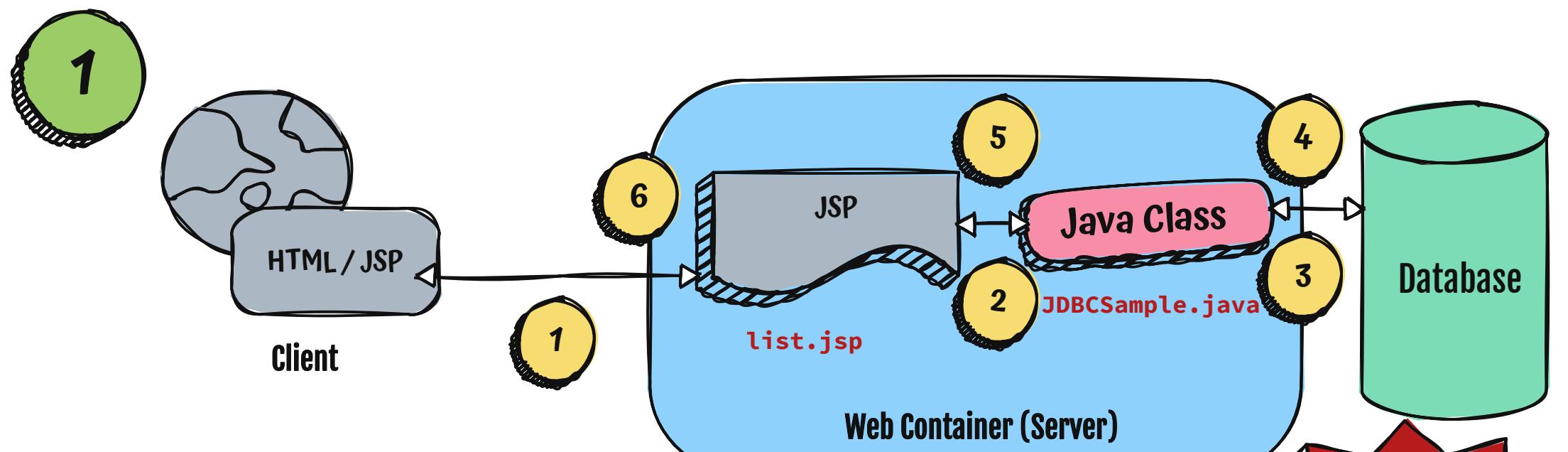


Different ways of Form Handling in Servlet/JSP



ServletConfig Vs ServletContext

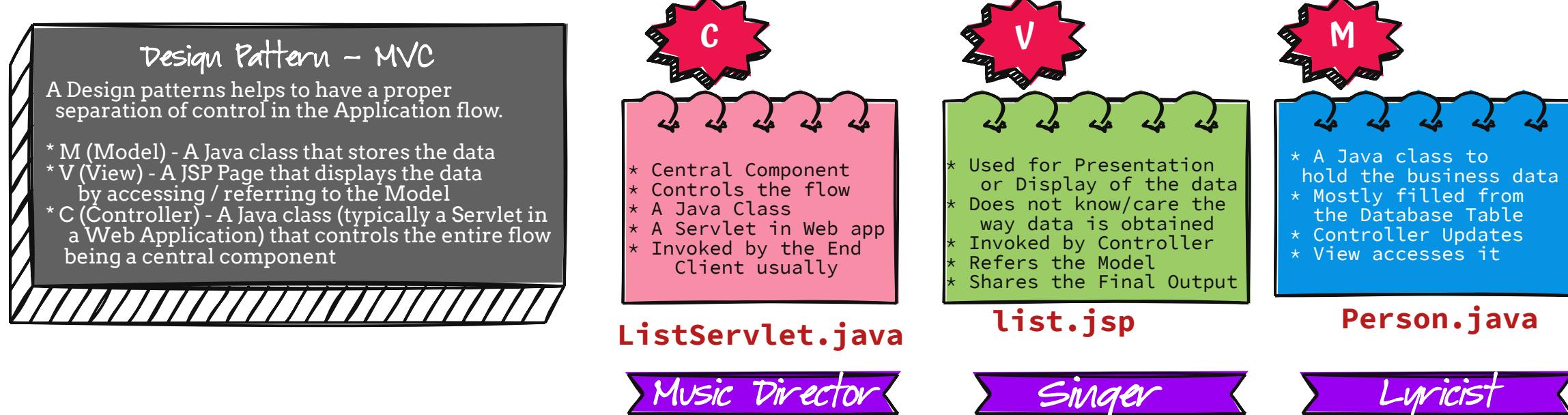
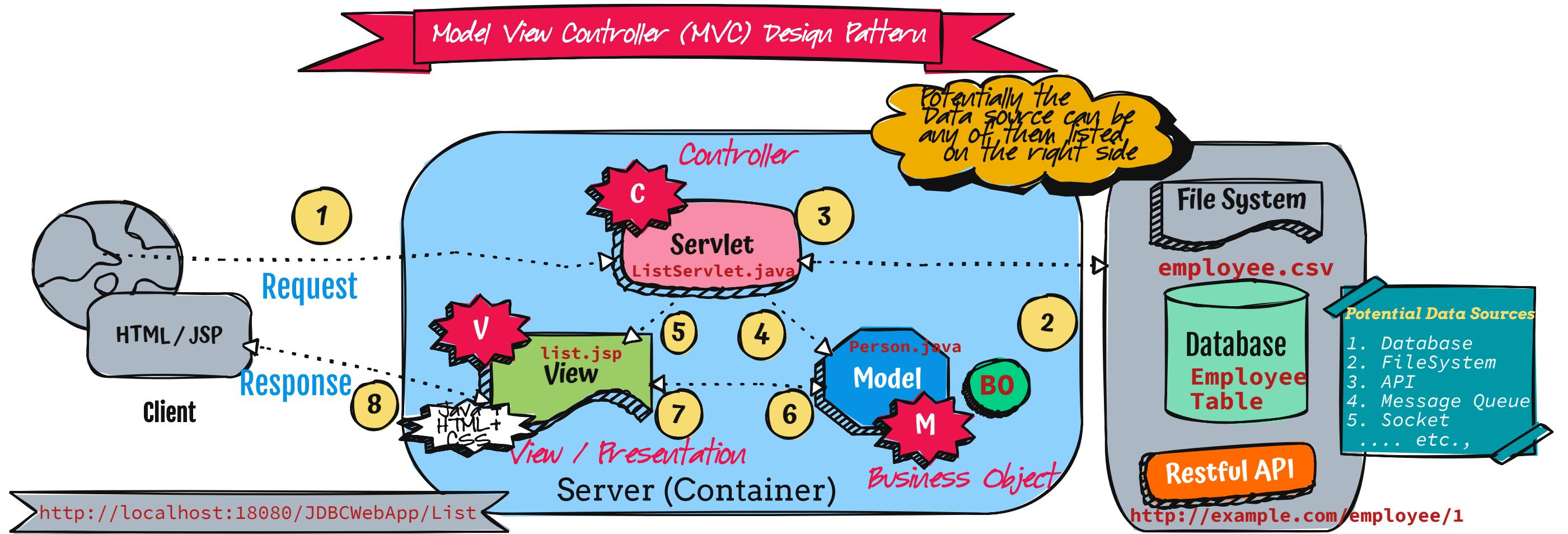


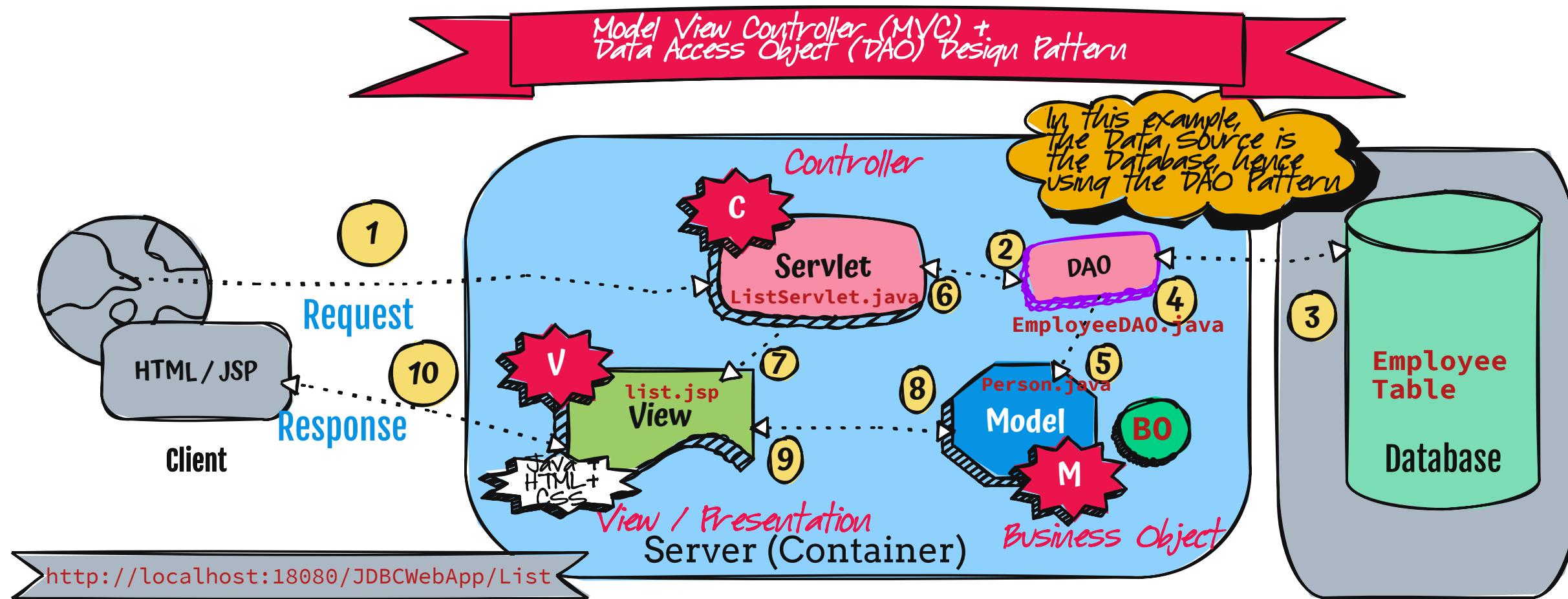


MVC Model

Design Principle for a Class
A Class should handle only one responsibility
Single Responsibility Principle

DAO - Data Access Object
* A design pattern that separates the Data Access Operations in a separate Java class
* EmployeeDAO - A DAO class deals with the Employee Table





<http://localhost:18080/JDBCWebApp>List>

Design Pattern - MVC

A Design patterns helps to have a proper separation of control in the Application flow.

- * M (Model) - A Java class that stores the data
- * V (View) - A JSP Page that displays the data by accessing / referring to the Model
- * C (Controller) - A Java class (typically a Servlet in a Web Application) that controls the entire flow being a central component

MUSIC Director



- * Central Component
- * Controls the flow
- * A Java Class
- * A Servlet in Web app
- * Invoked by the End Client usually

ListServlet.java

Singer



- * Used for Presentation or Display of the data
- * Does not know/care the way data is obtained
- * Invoked by Controller
- * Refers the Model
- * Shares the Final Output

list.jsp

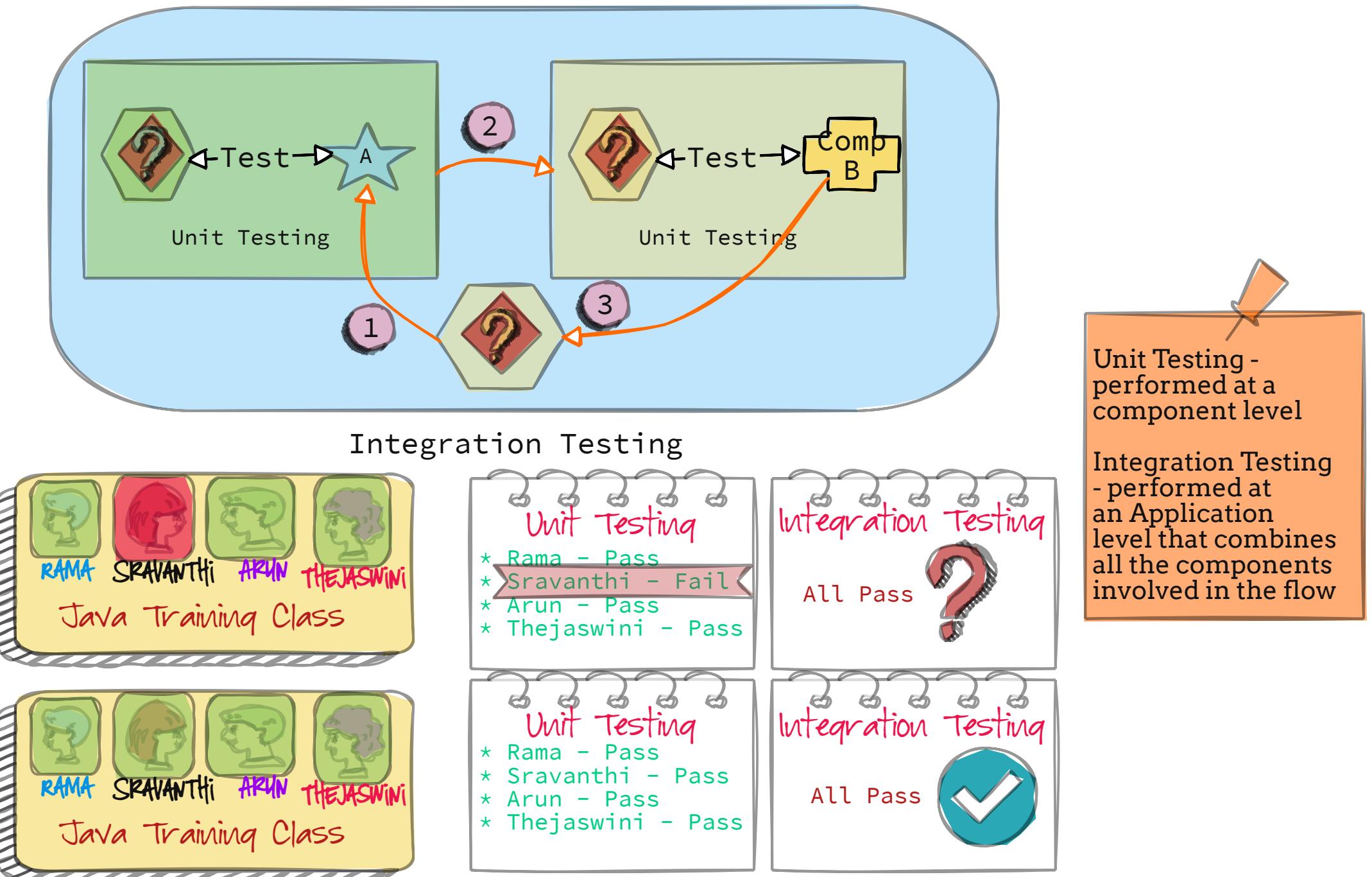
Lyricist



- * A Java class to hold the business data
- * Mostly filled from the Database Table
- * Controller Updates
- * View accesses it

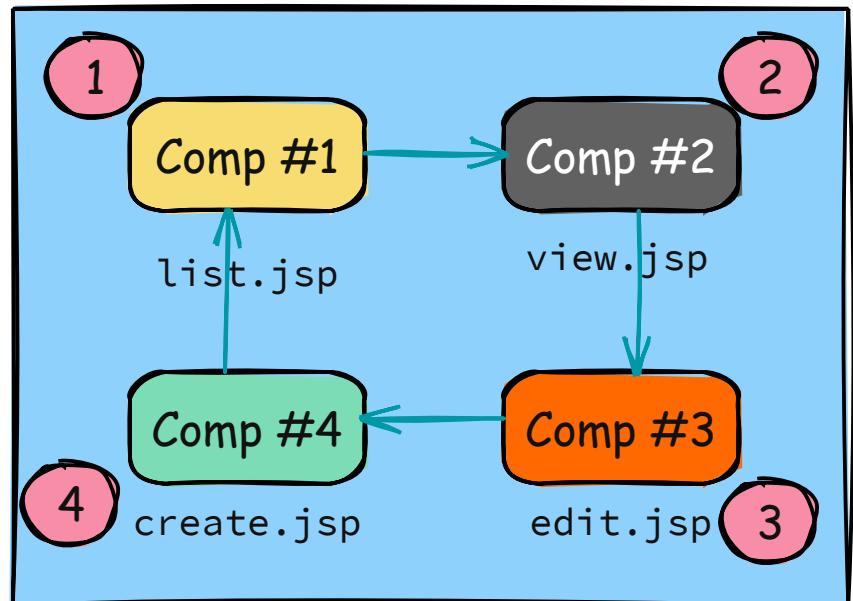
Person.java

Unit Vs Integration Testing

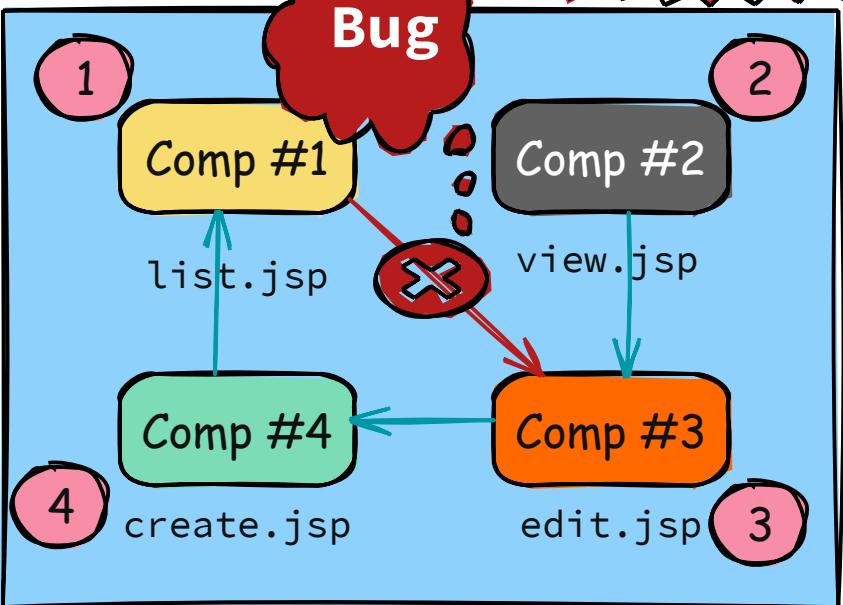


SANITY VS REGRESSION TESTING

Not guaranteed
but possible



Happy Path Scenario
Sanity Testing



Regression Testing
Unusual / Mixed Path

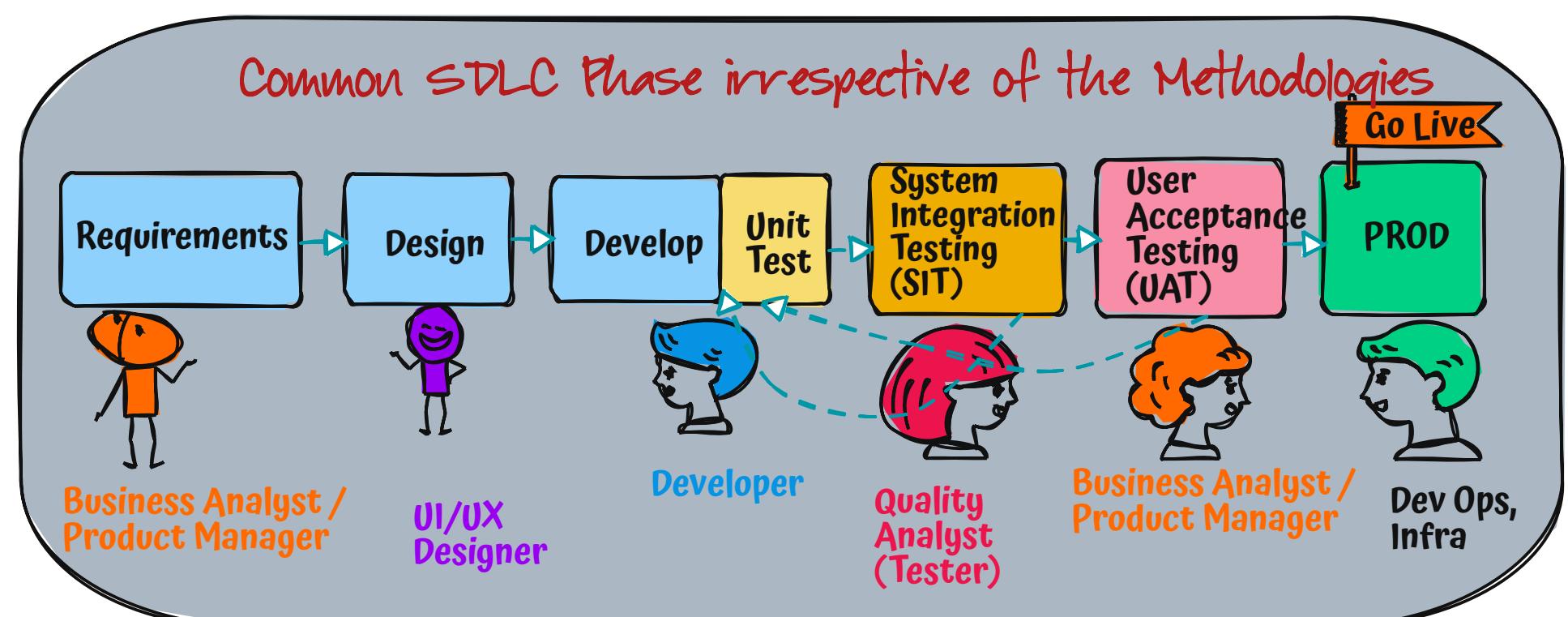
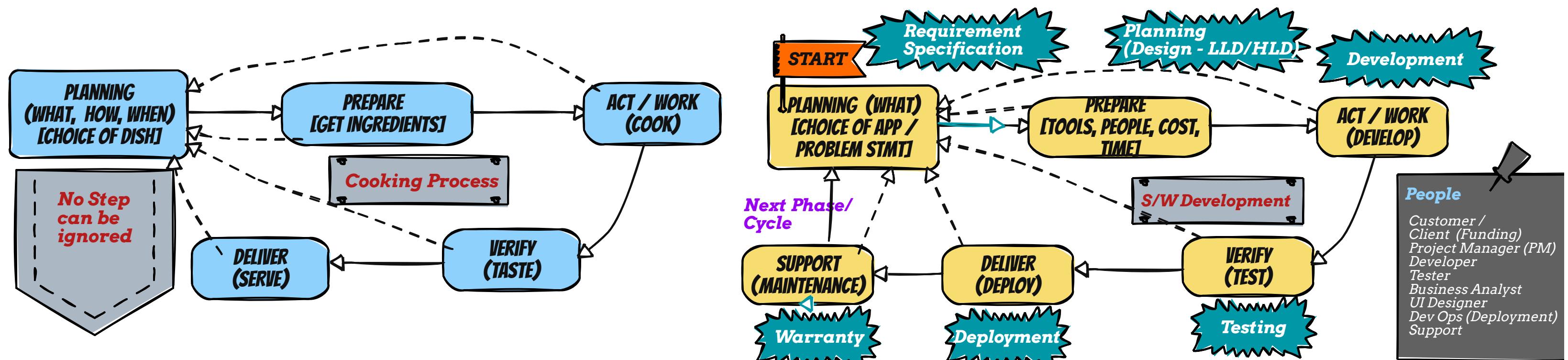
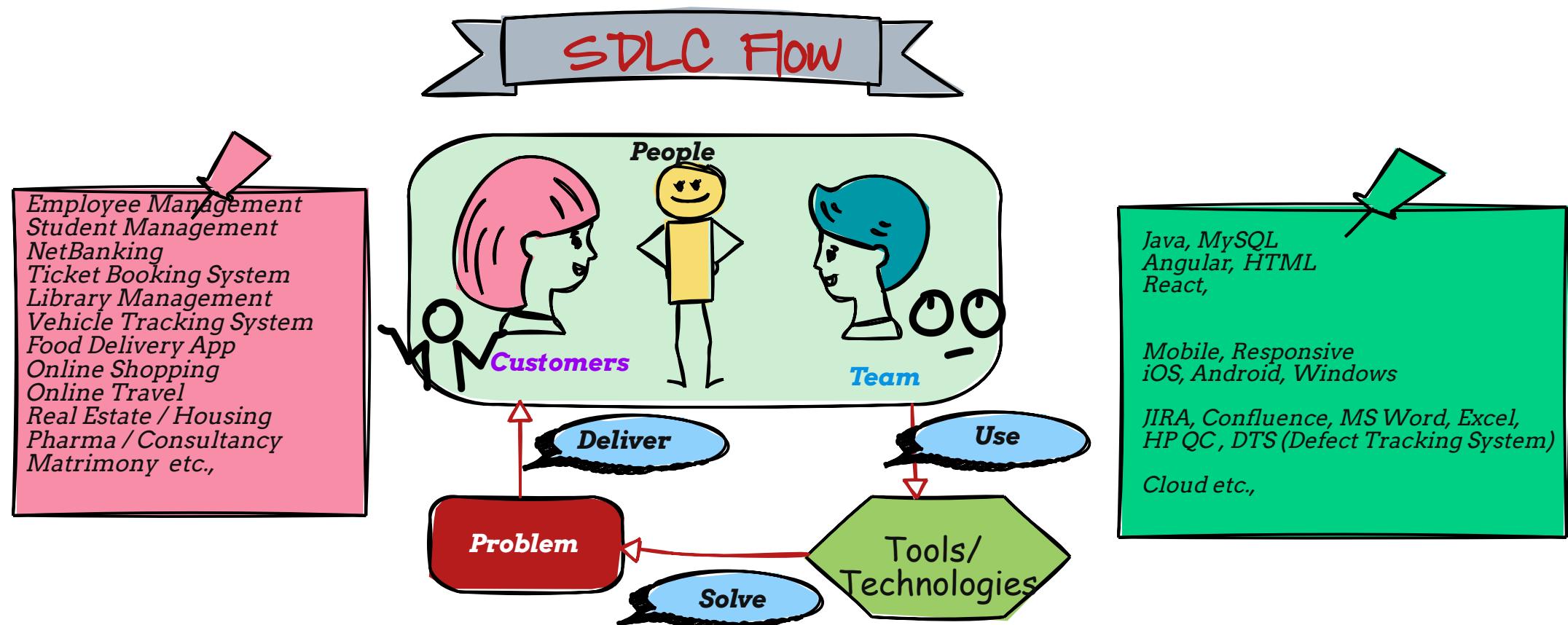
Sanity -
Order of testing

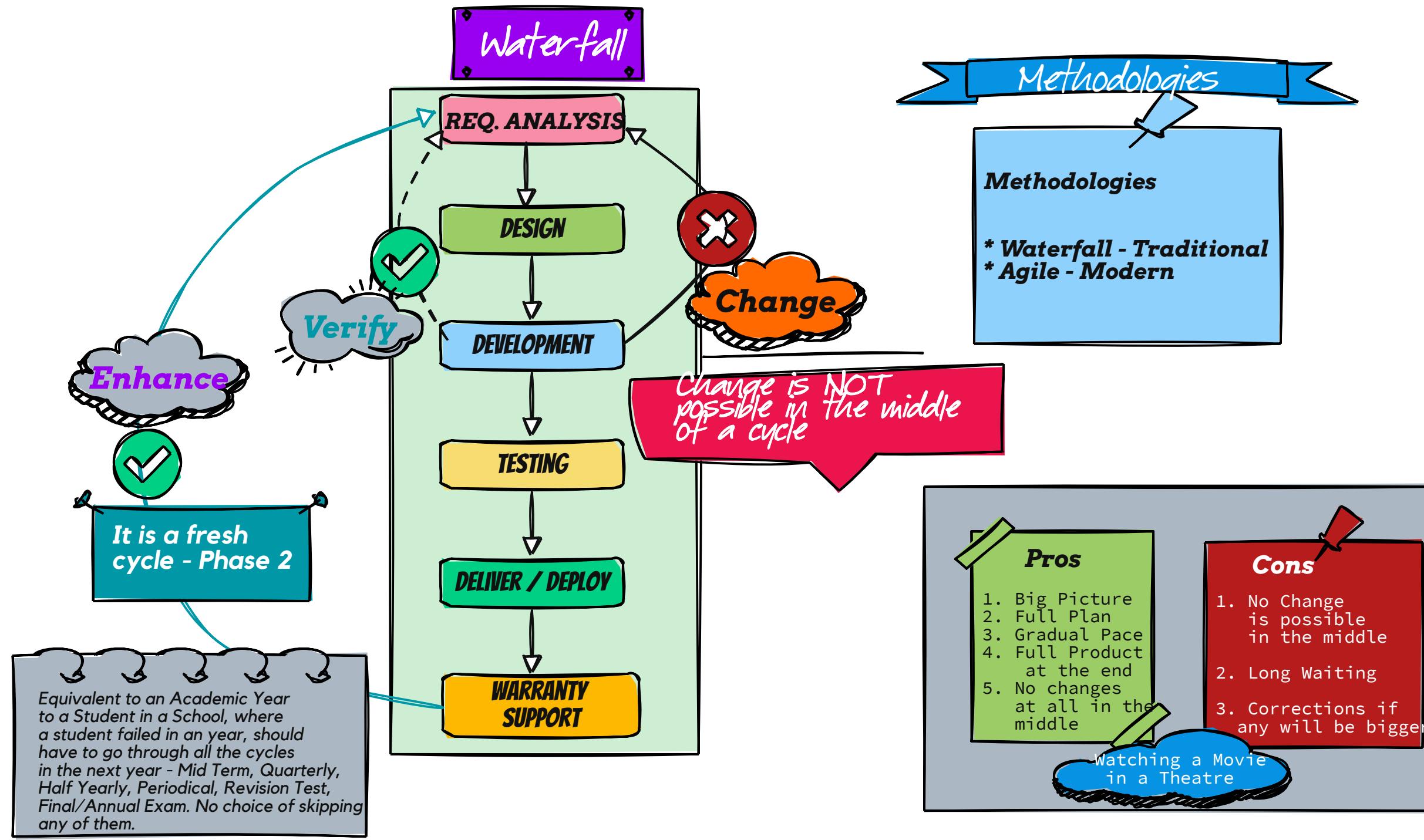
Sanity Vs Regression -
Usual or Natural Vs
Unusual or Mixed

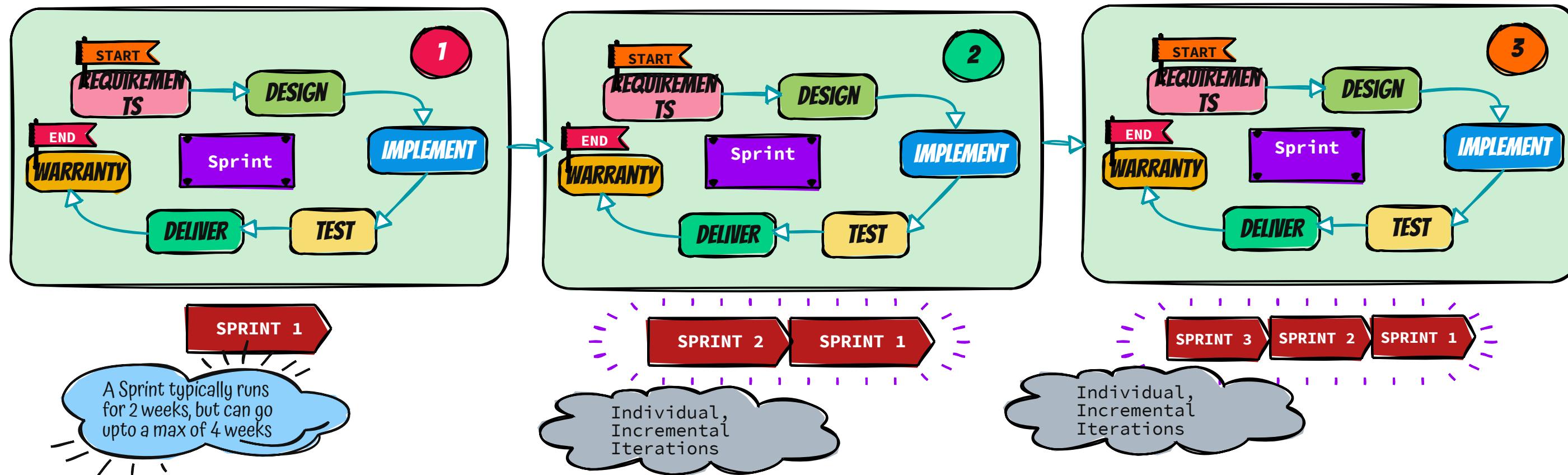
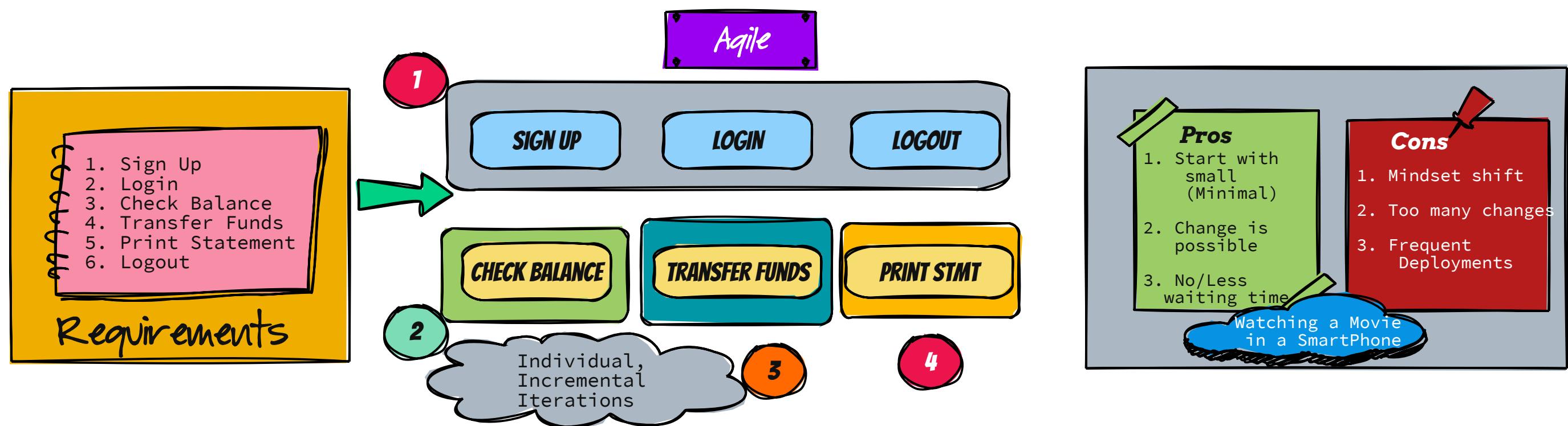
Unit -
Quantity of Components

Unit Vs Integration -
Single Component Vs
Multiple Components

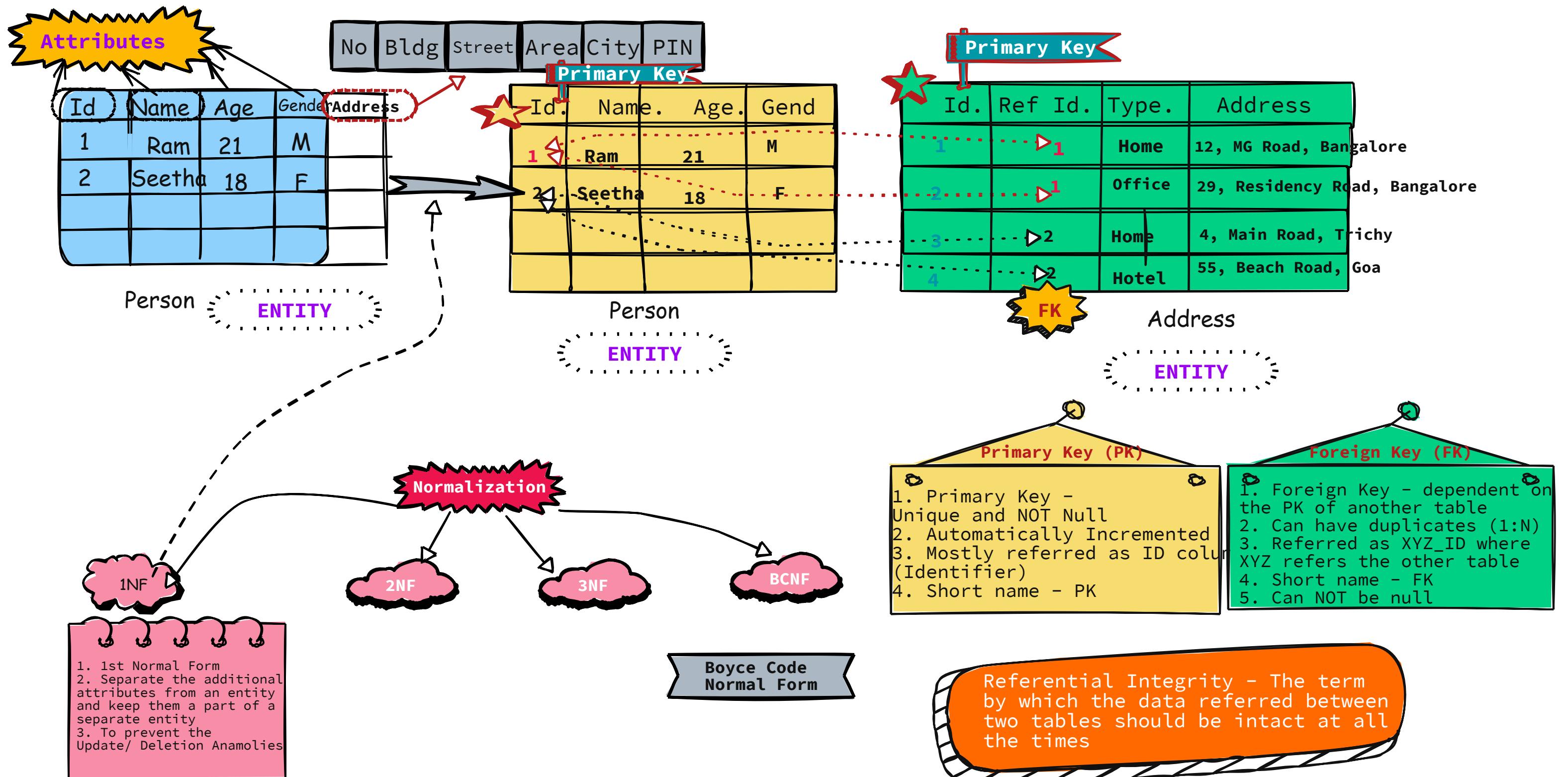
Software Development Life Cycle



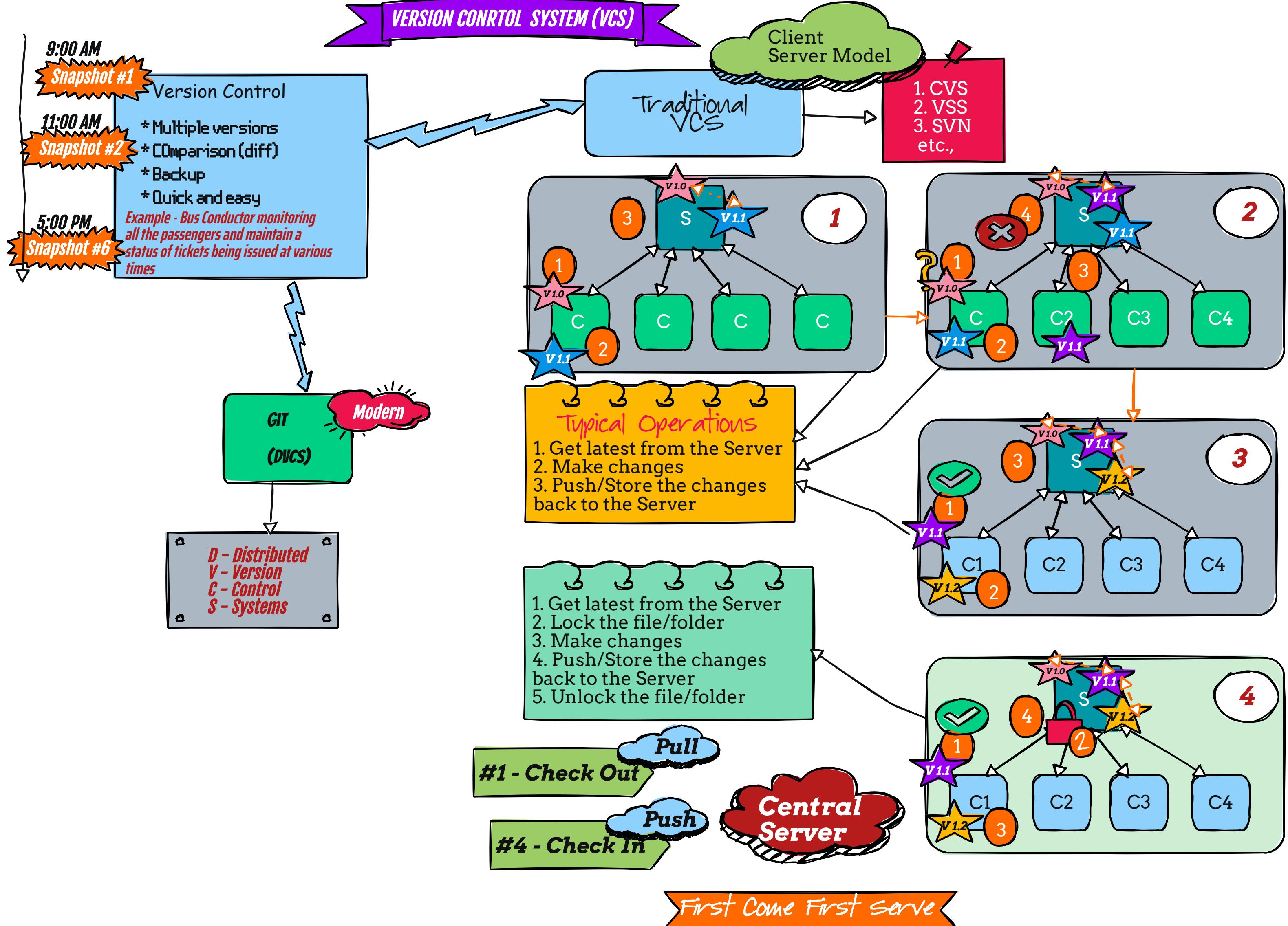




SQL - Primary, Foreign Keys



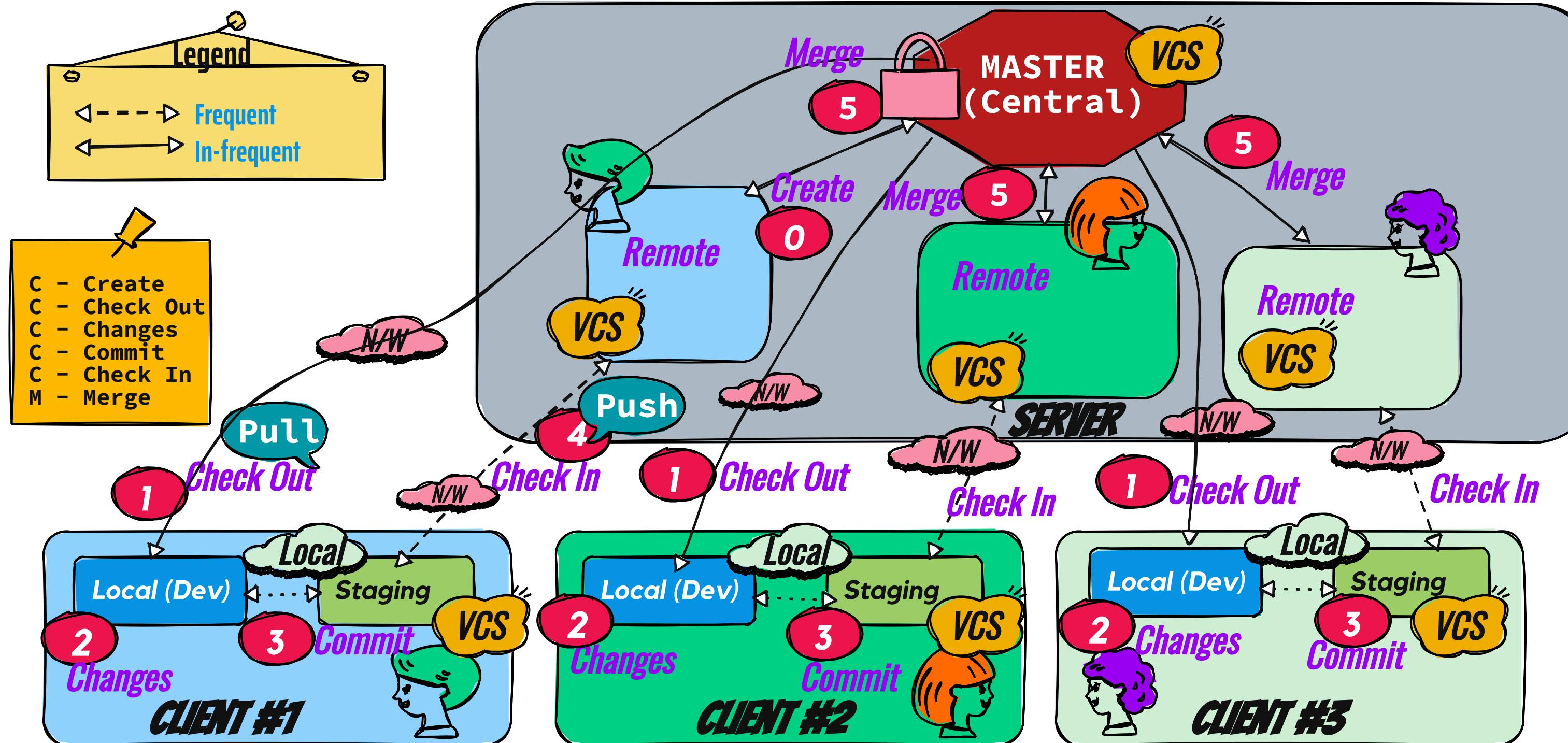
VERSION CONTROL SYSTEM (VCS)



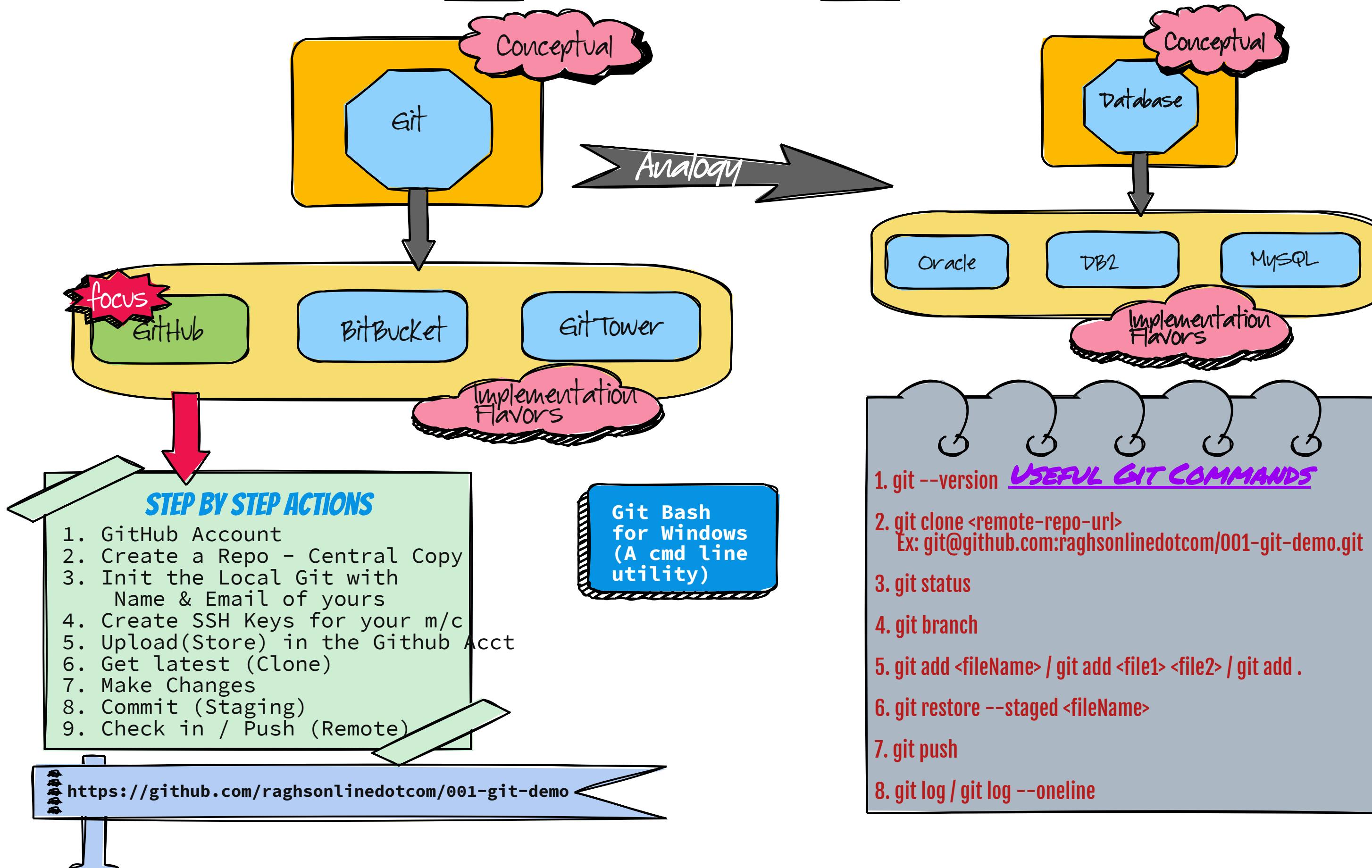
D - Distributed
V - Version
C - Control
S - Systems

git - DVCS

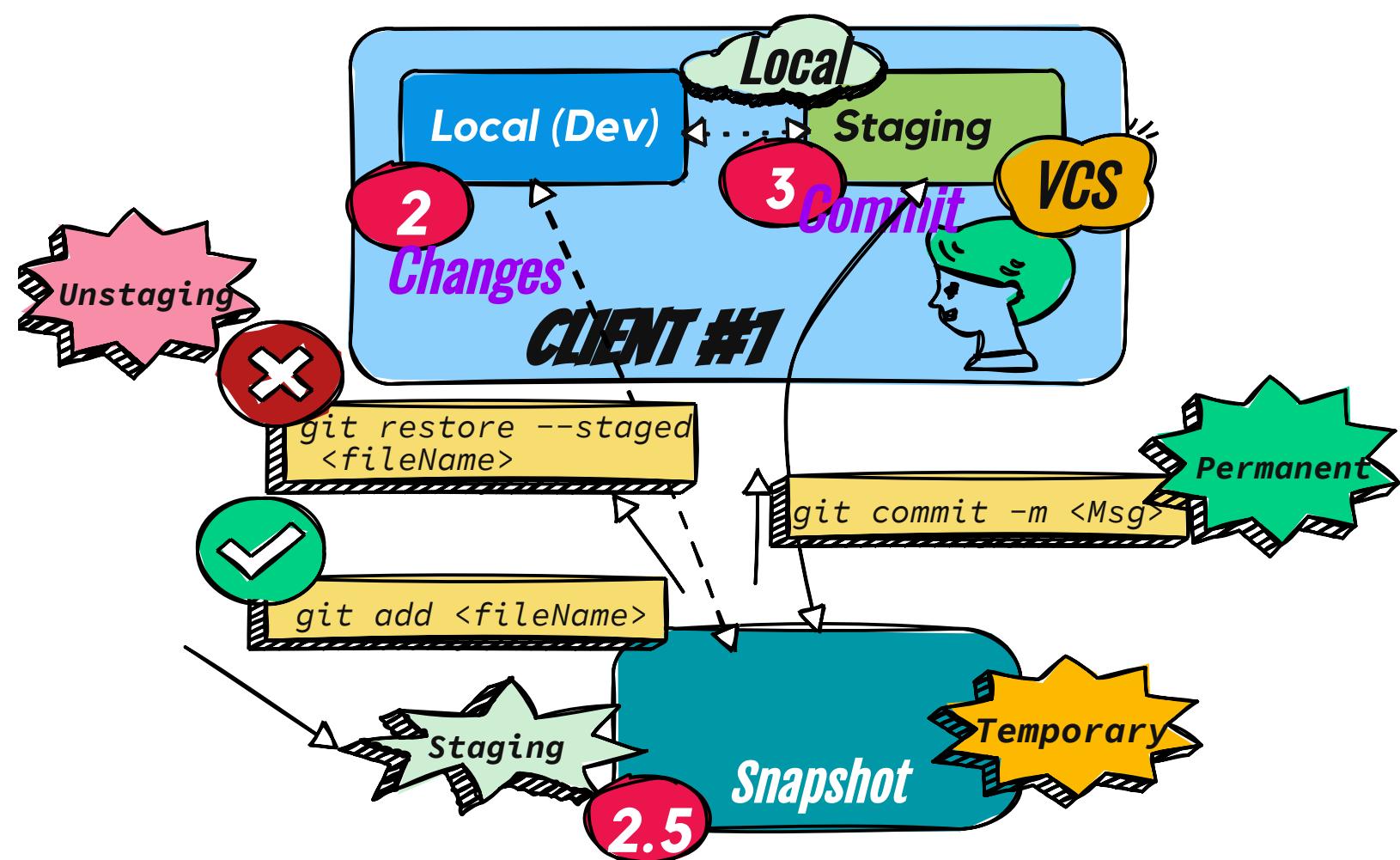
Made with ❤️
 Using SketchWow
 by itsraghz
 09.Nov.2022



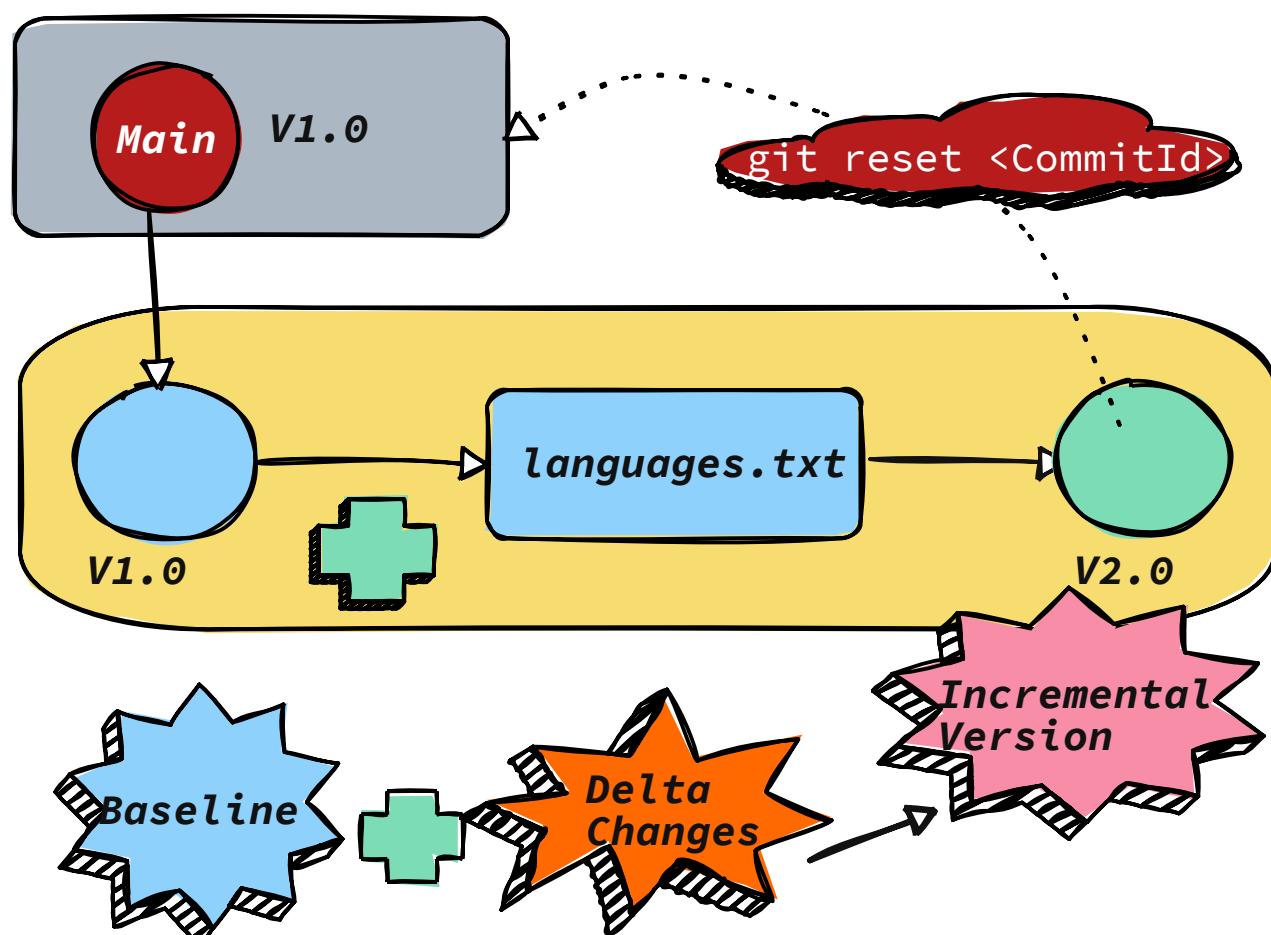
Git & Flavors, Git In Action



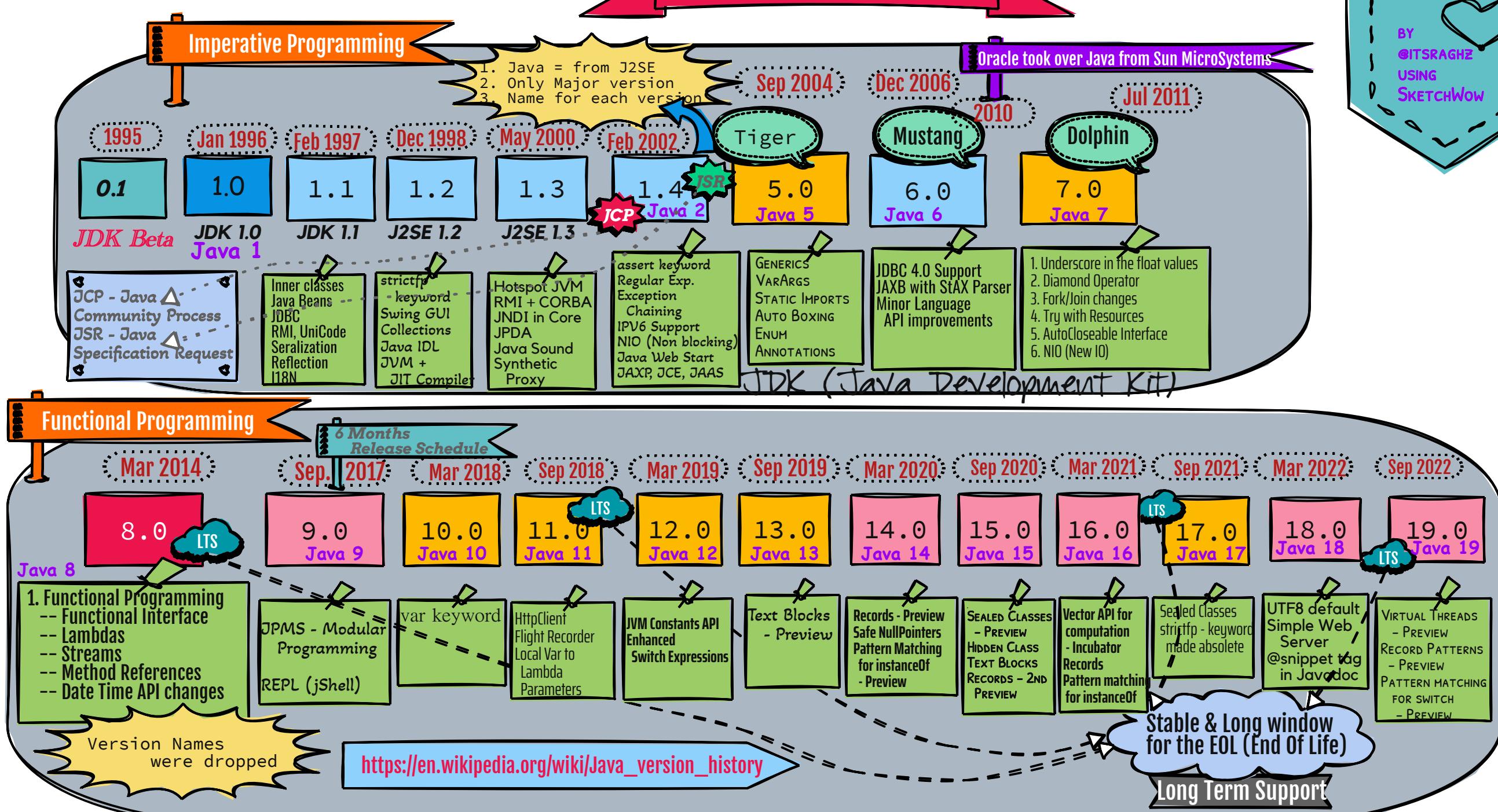
Git Local - Deep Dive



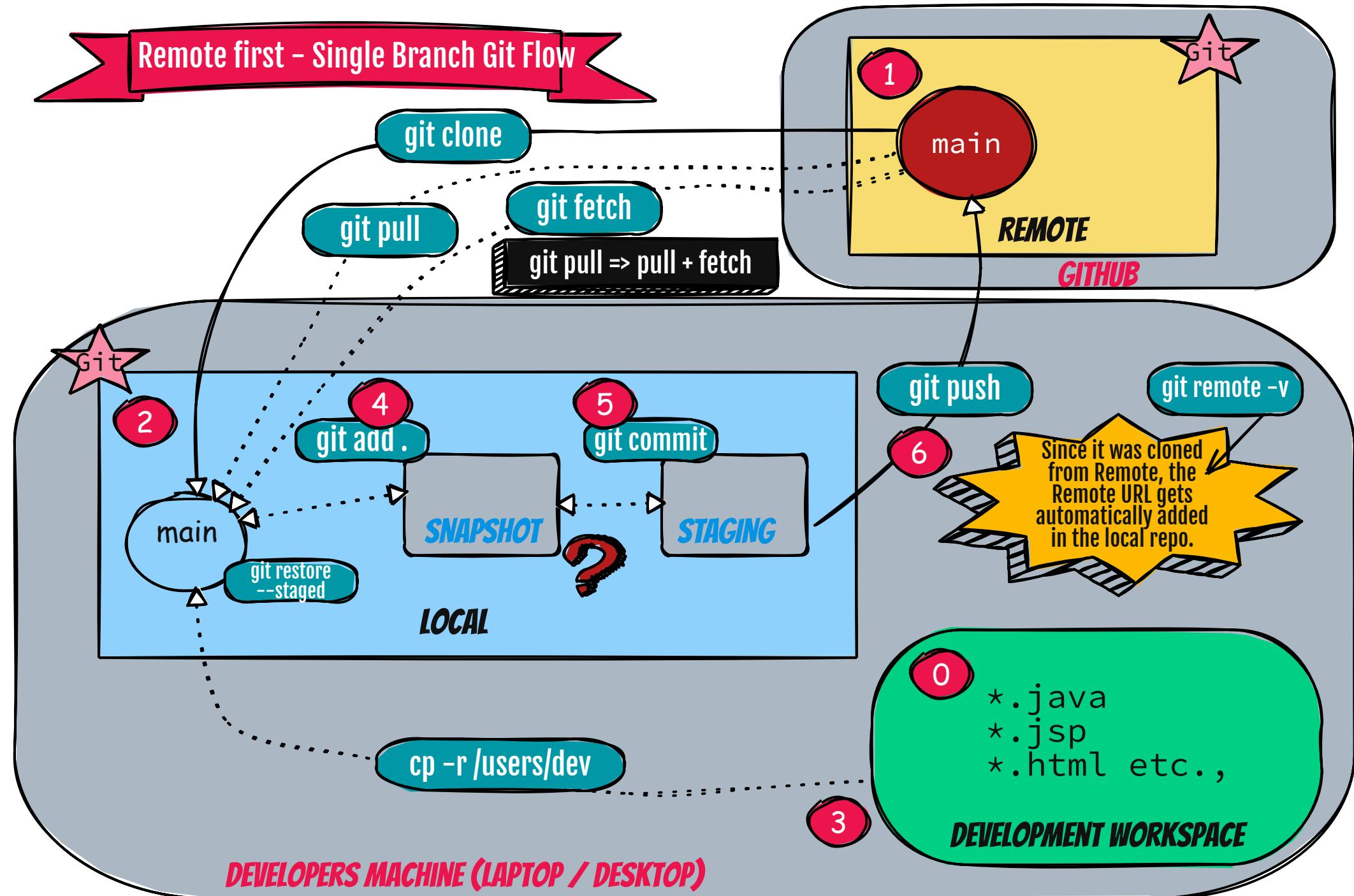
Git Incremental Delta

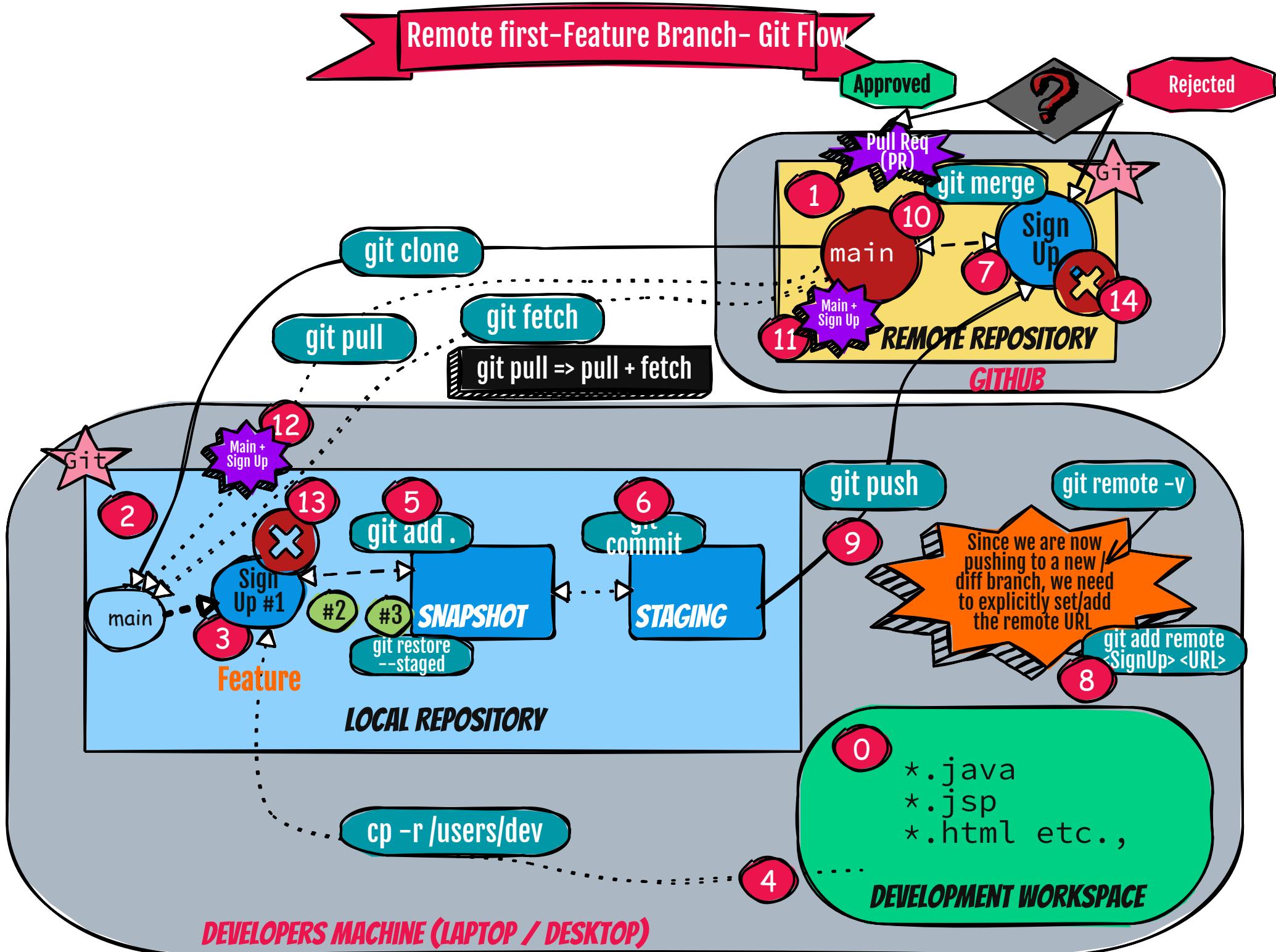


Java Version History



Remote first - Single Branch Git Flow





GITHUB - BUGFIX WORKFLOW

