Fusion Knowledge

Automated cloud provisioning system

By Sachin Shinde

A quick introduction...

SACHIN SHINDE

Education:

Master of Science in Computer Engineering(2014 - 2016)
Santa Clara University, CA

Bachelor of Engineering in Electronics (2006 - 2010)
University of Mumbai, India

Past experience:

Deloitte Consulting - 6 months
Systems Engineer 3

Infosys Technologies - 3 years
Senior Systems Engineer

Why do we need this system?

- **Customer**

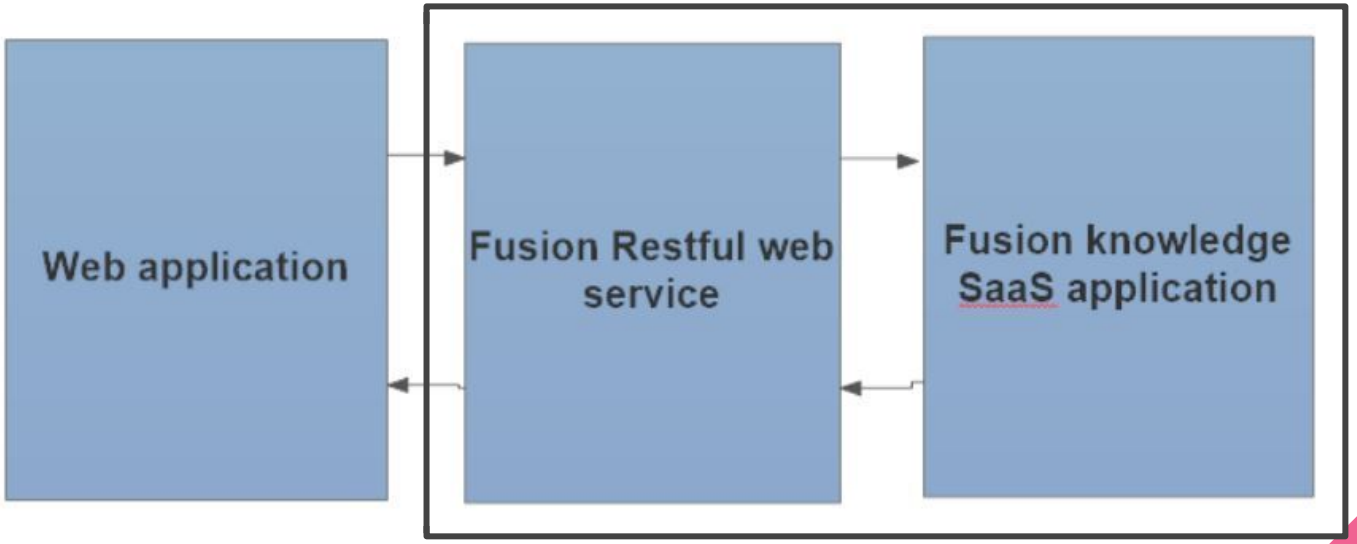
  For easy self provisioning

  For convenient subscription management

- **Service provider**

  No human intervention which avoids human error

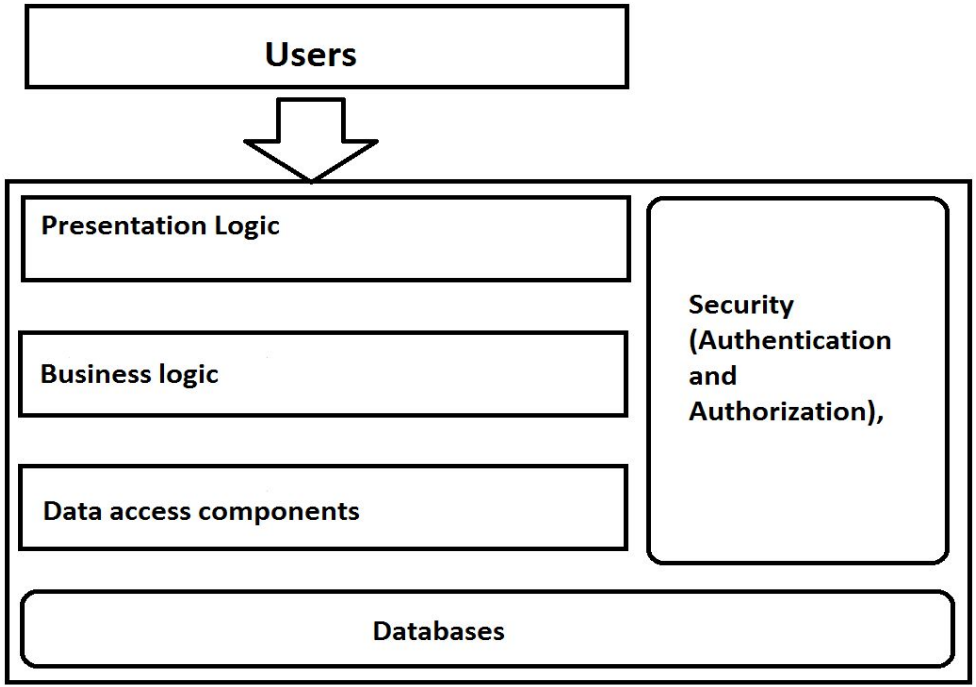
High level view of the system



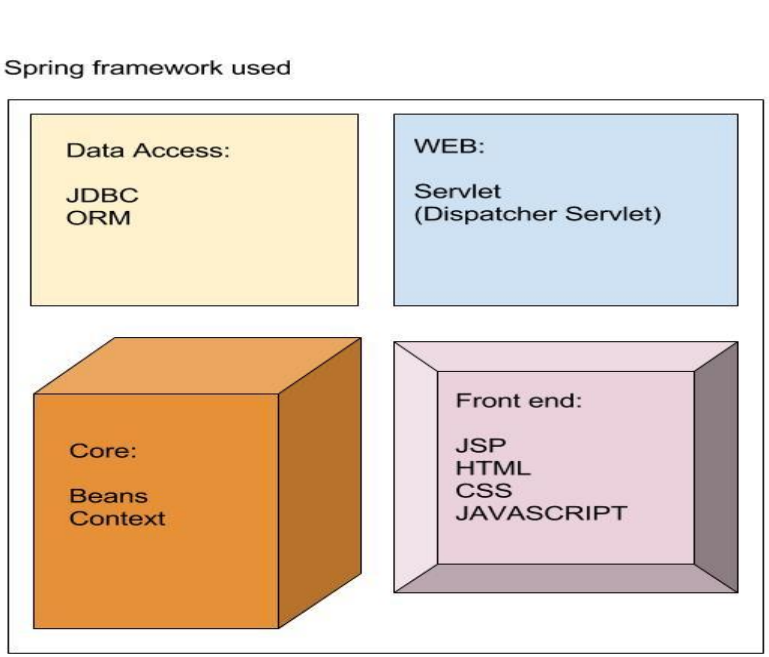
Let us focus on every component in detail...

The web application

High level view of the web application



Specifications



WebLogic application Server

Build tool:
MAVEN

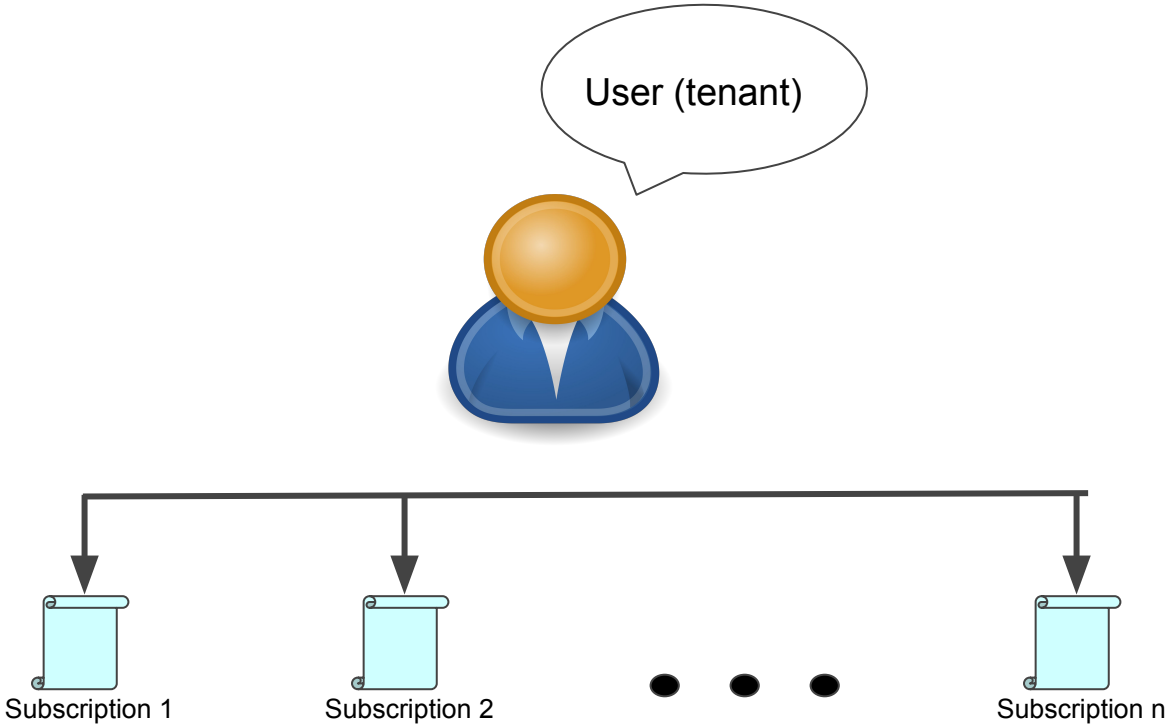
SMTP server

Why use spring framework?

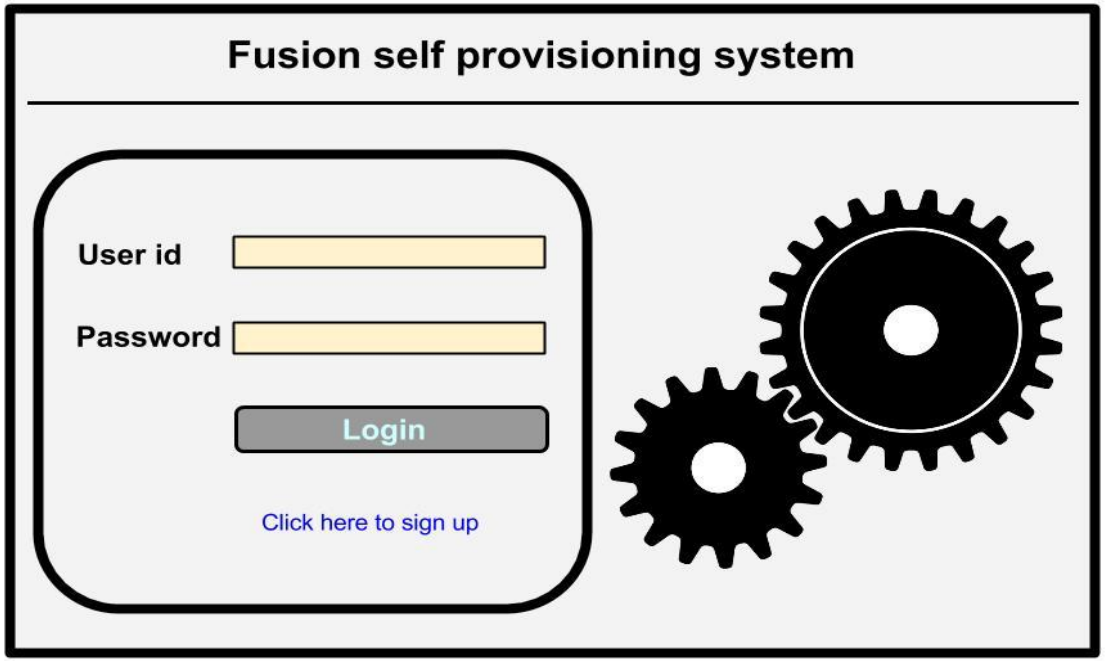
- Framework handles bean wiring and instantiation efficiently
- Better object oriented design
- Spring security
- Easy support for internationalization

Application flow for a user...

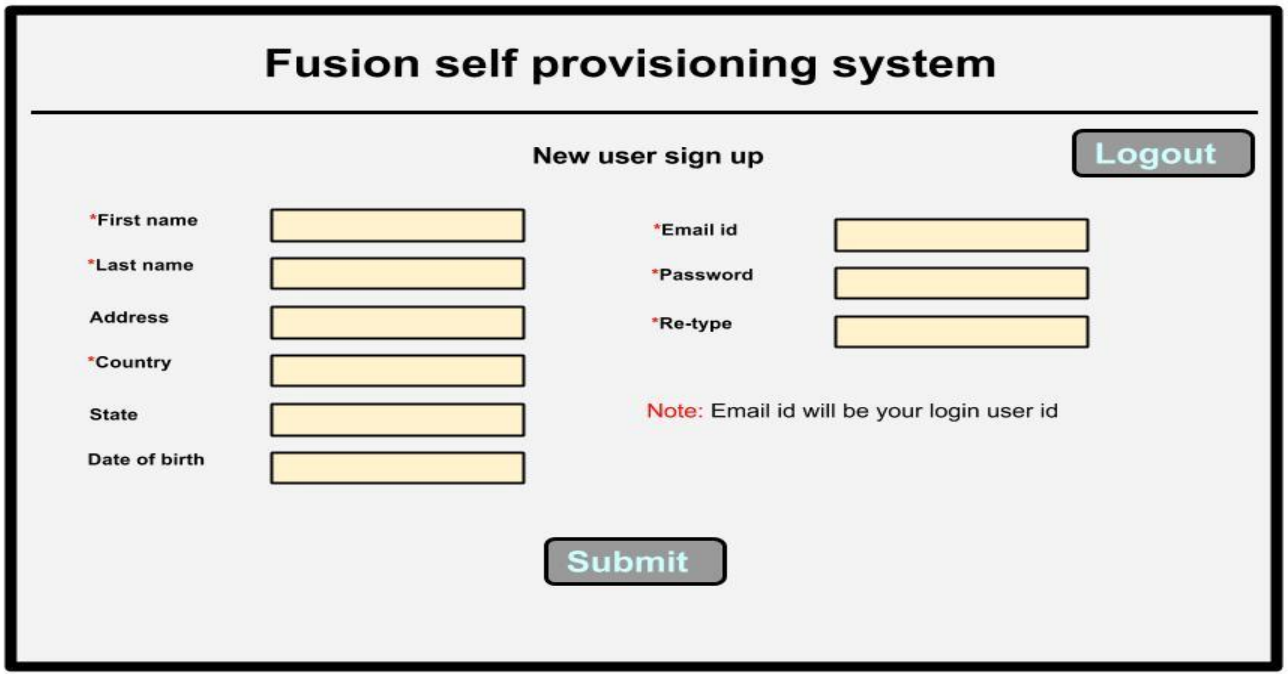
User and subscription hierarchy



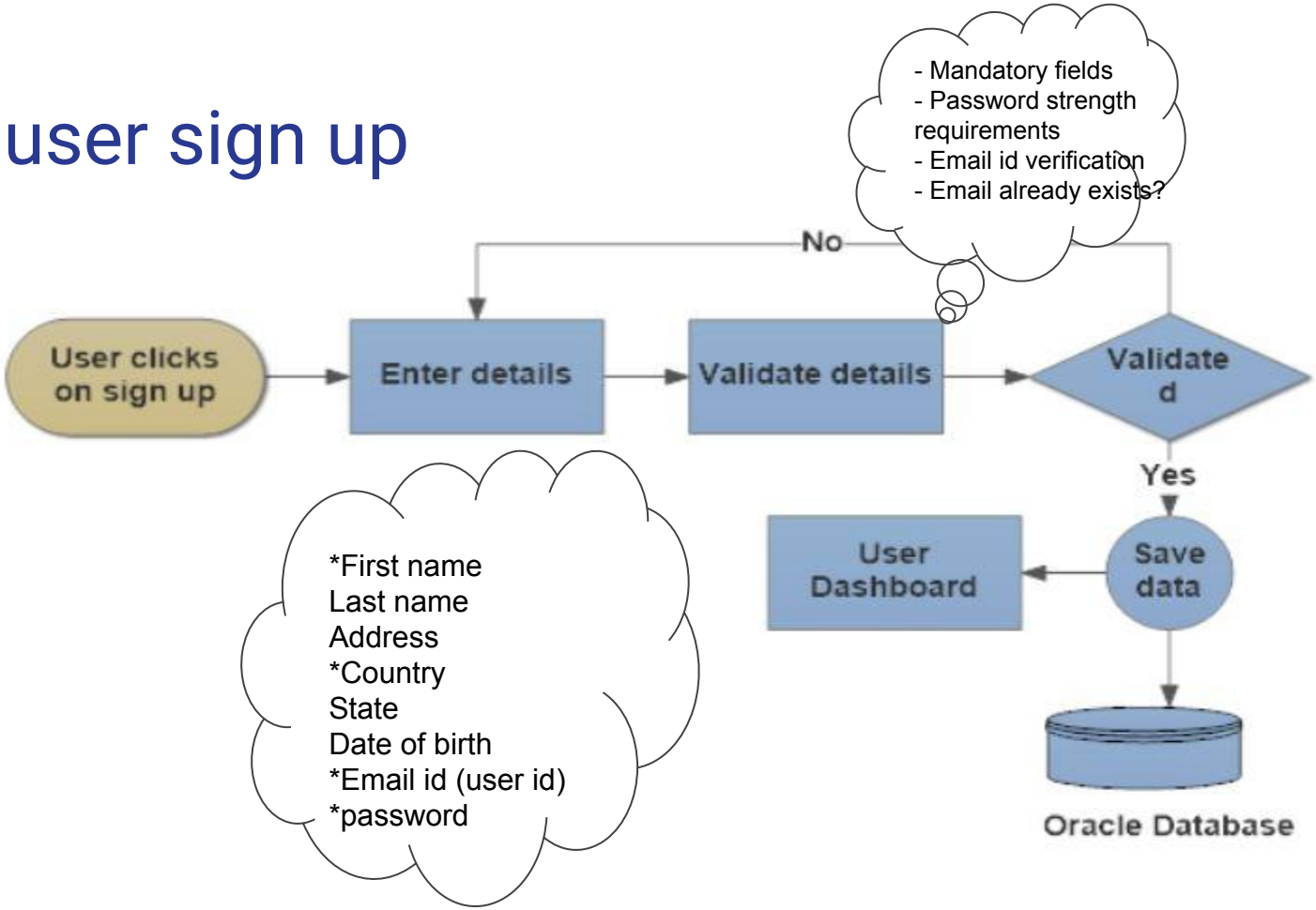
Sample login screen...



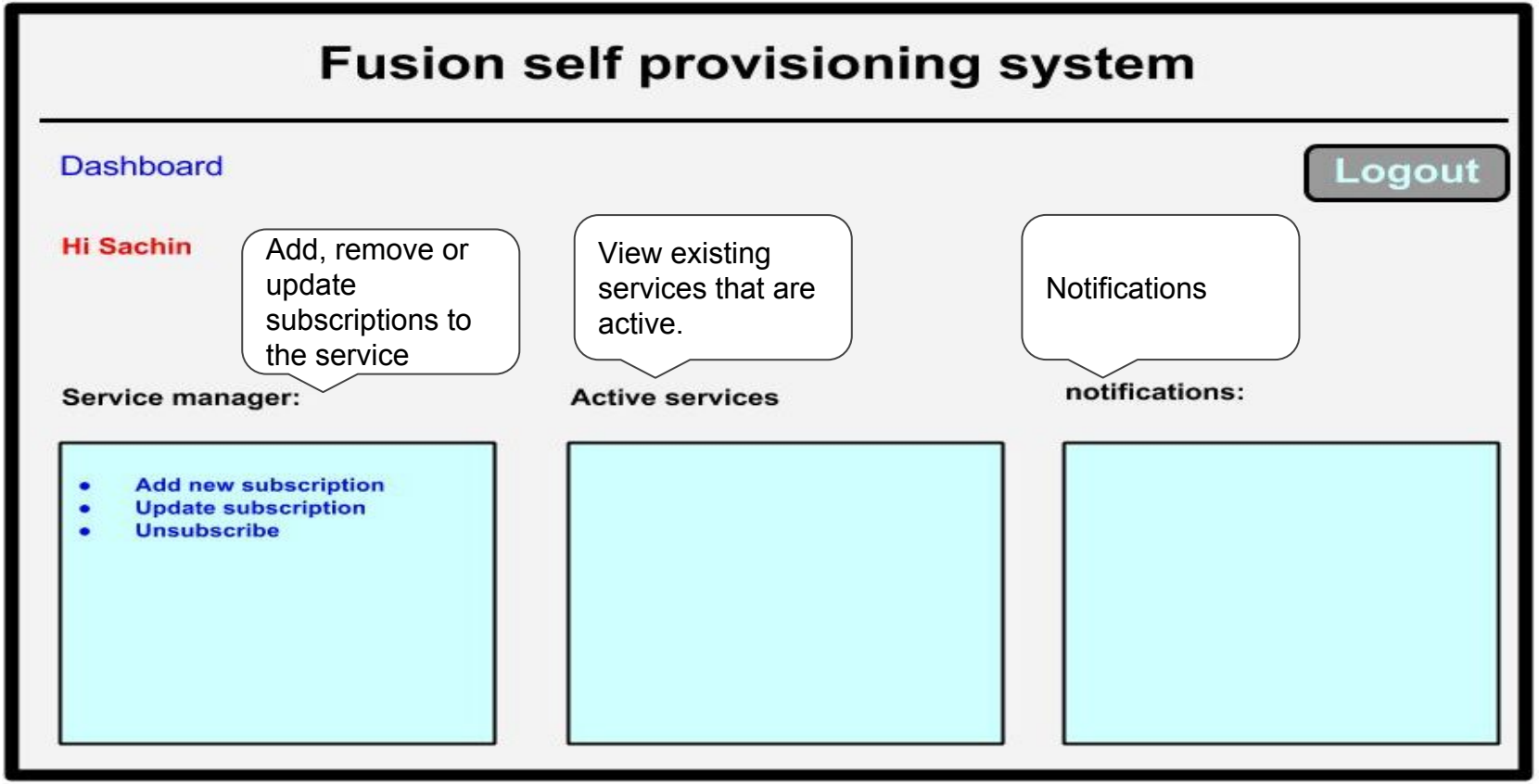
New user sign up screen



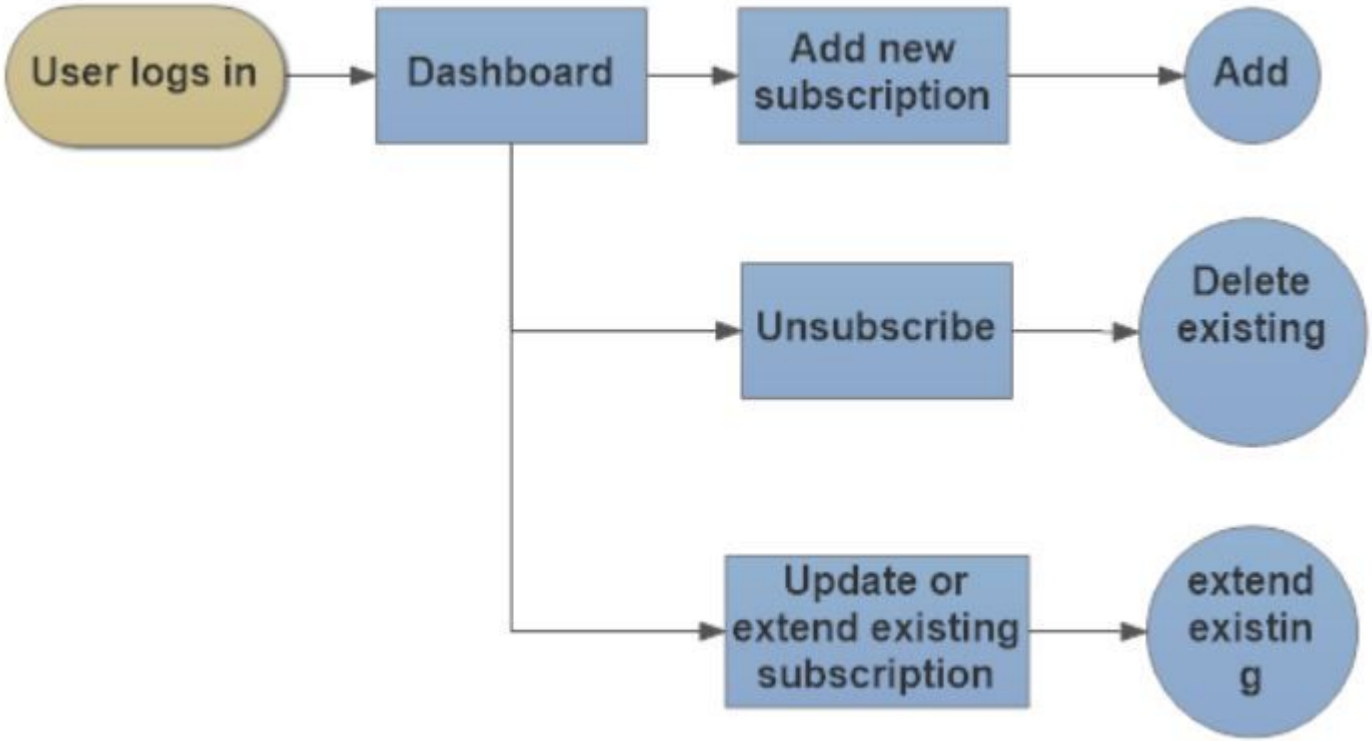
New user sign up



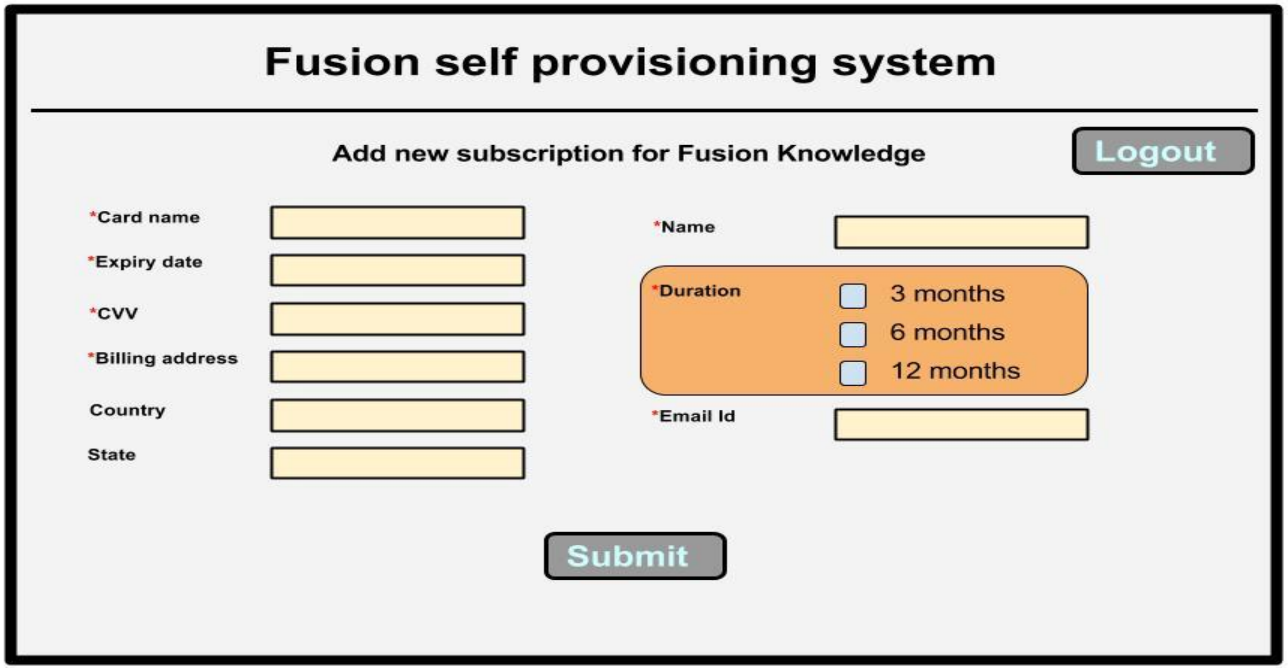
Dashboard


Fusion self provisioning system
Dashboard
Logout
Hi Sachin
Add, remove or update subscriptions to the service
View existing services that are active.
Notifications
Service manager:
Active services
notifications:
Add new subscription
Update subscription
Unsubscribe

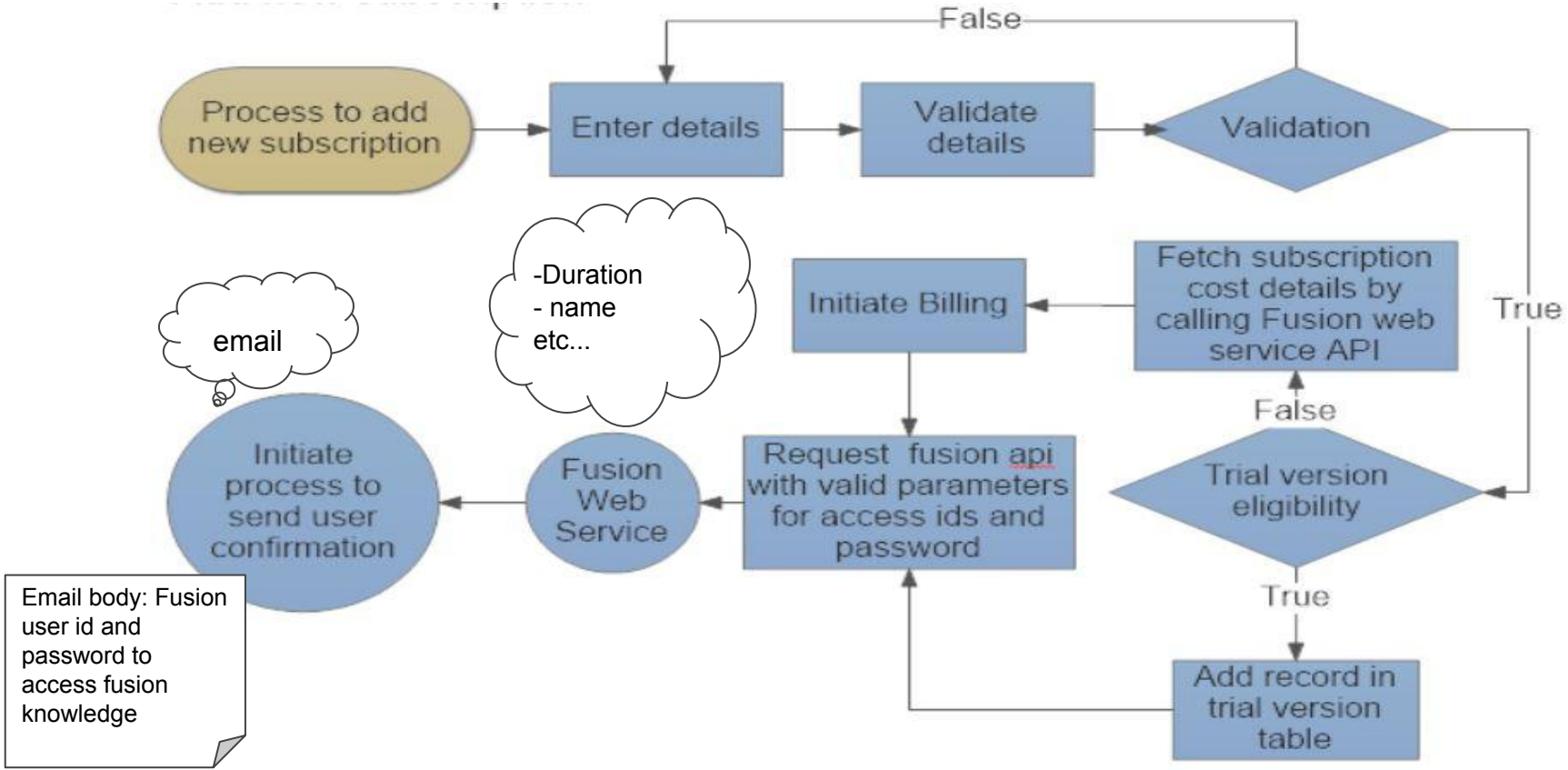
Service manager



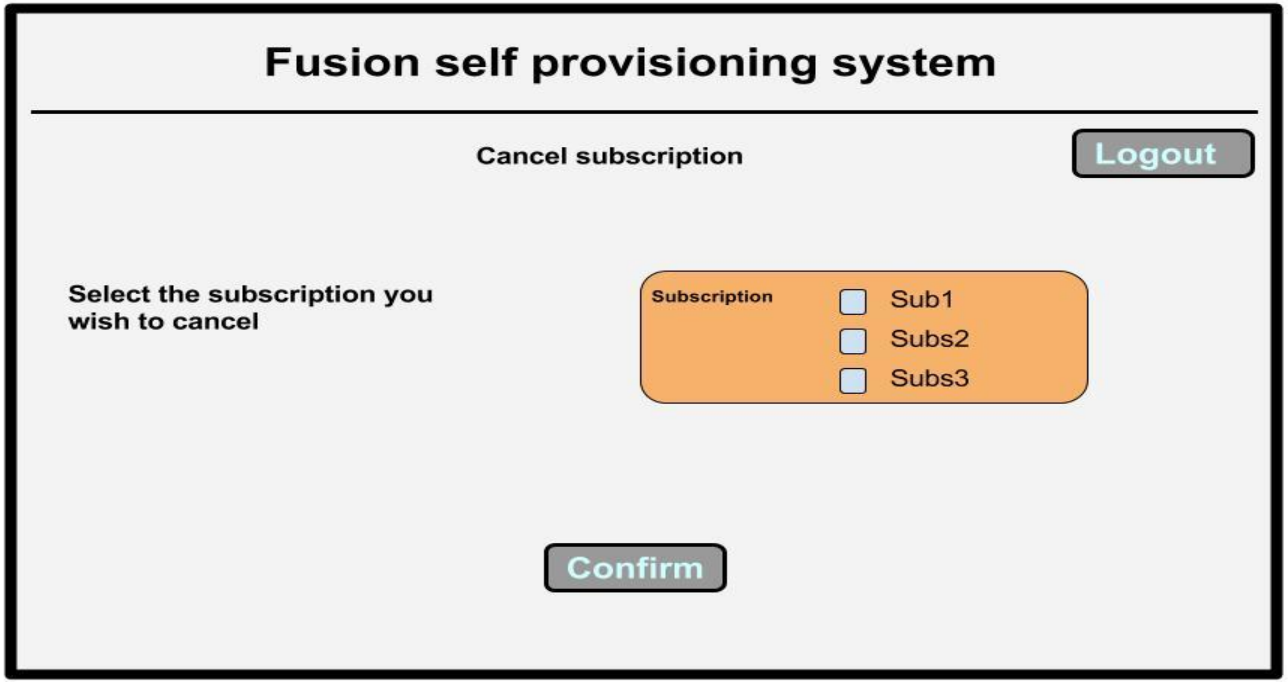
Add new subscription screen



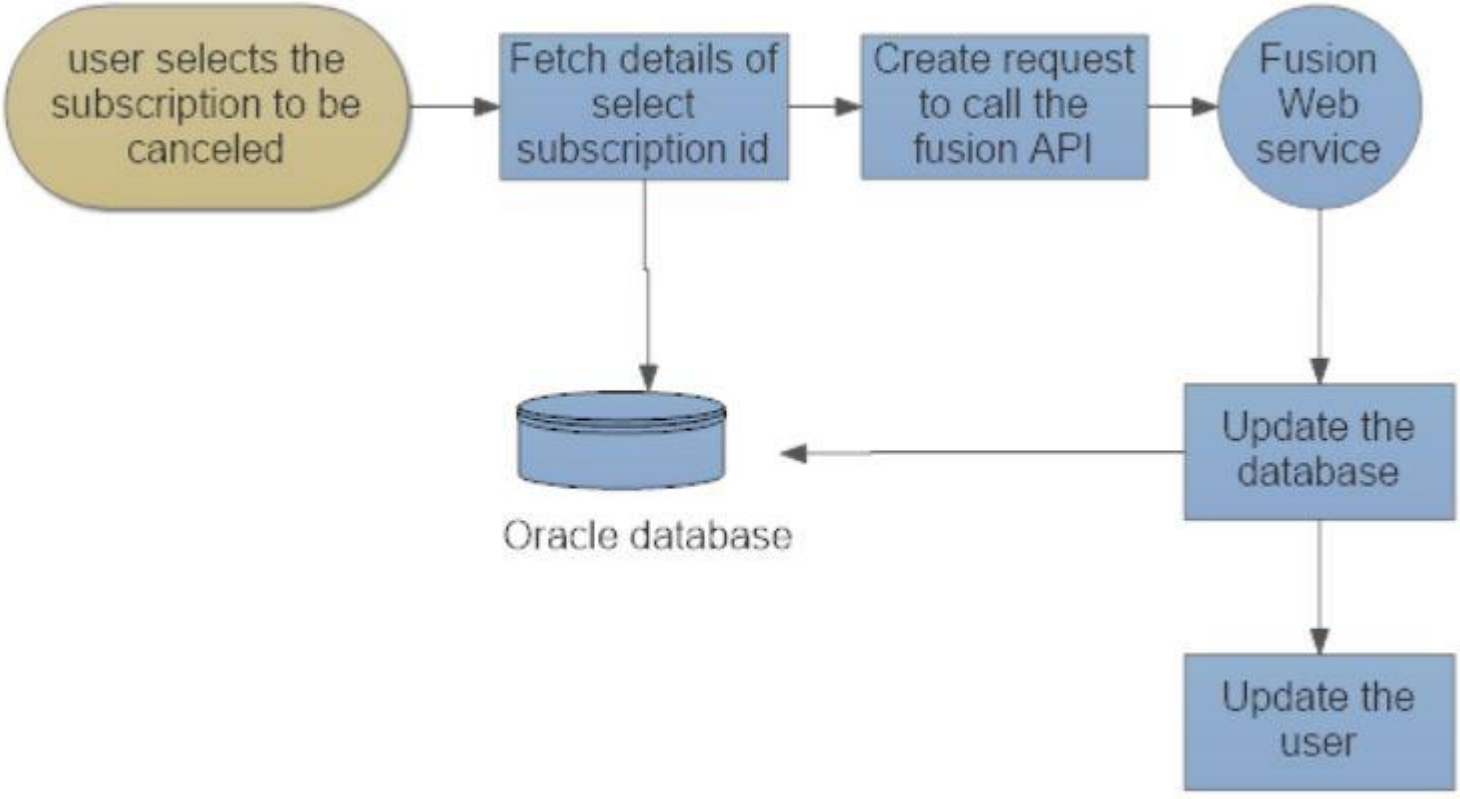
Add new subscription



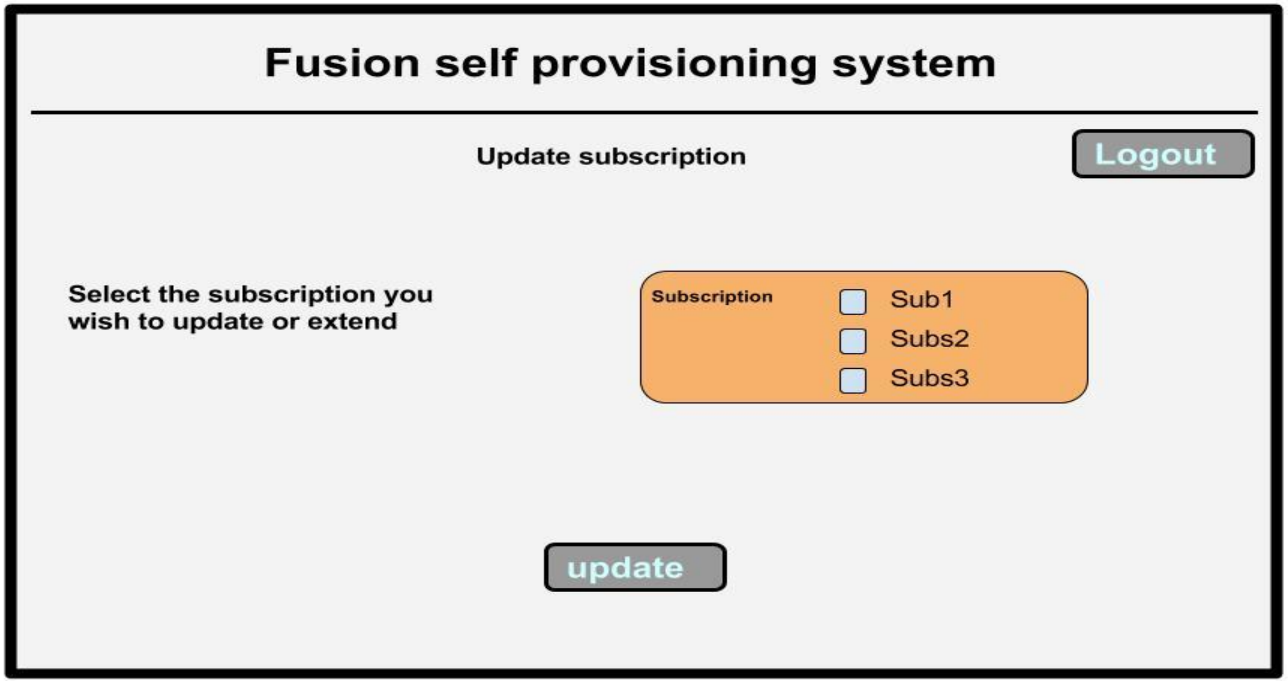
Cancel subscription screen



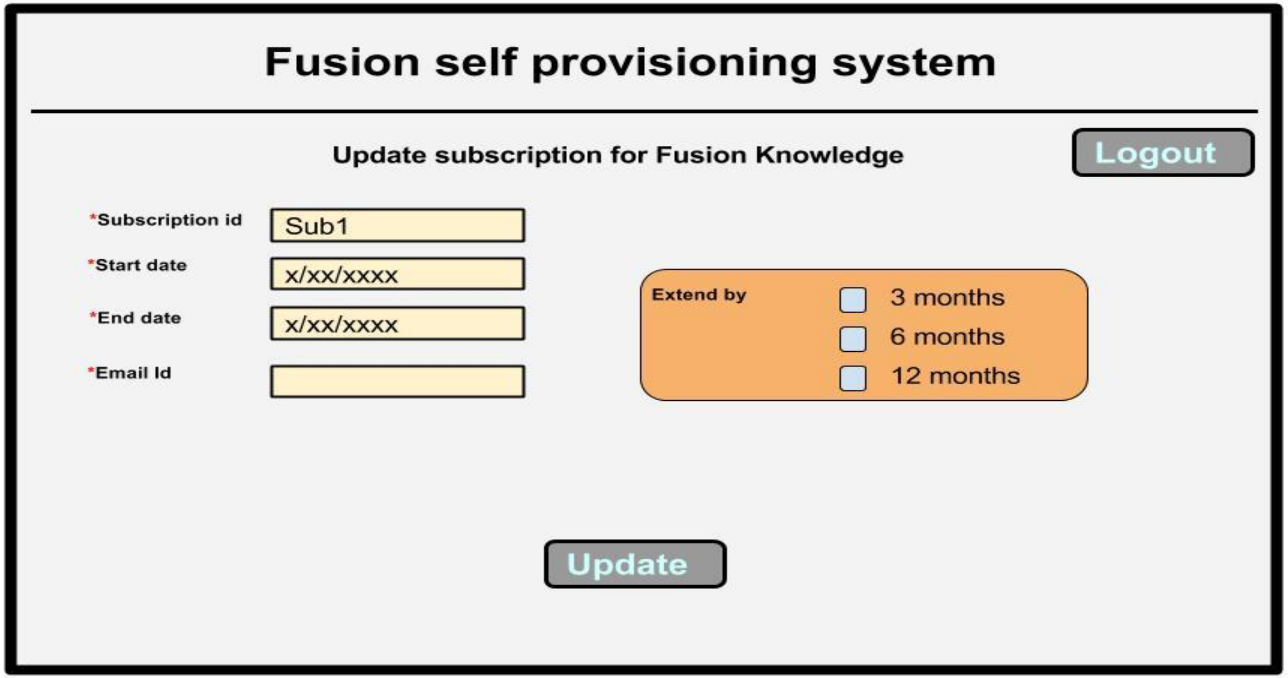
Cancel existing subscription



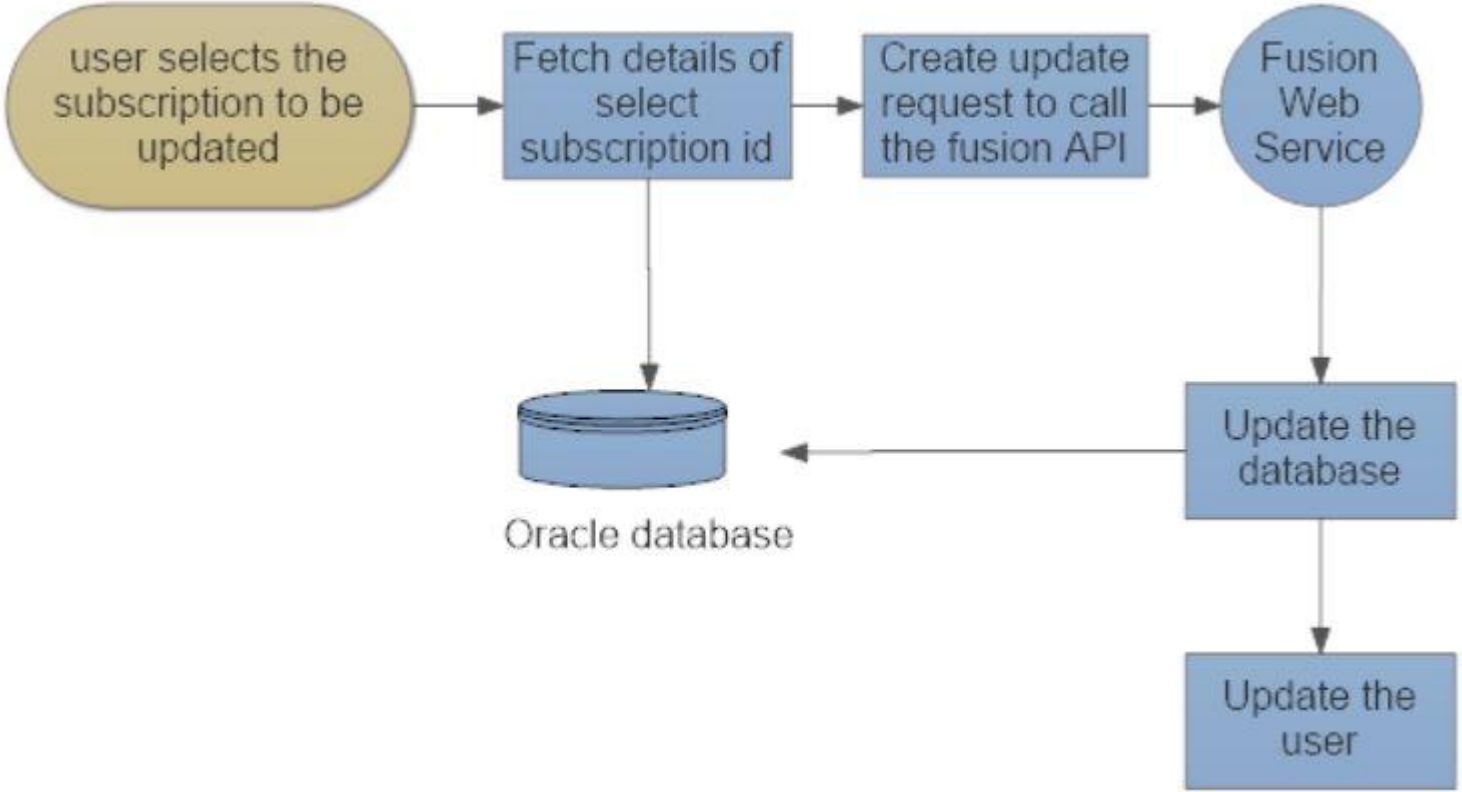
Select subscription to be updated screen



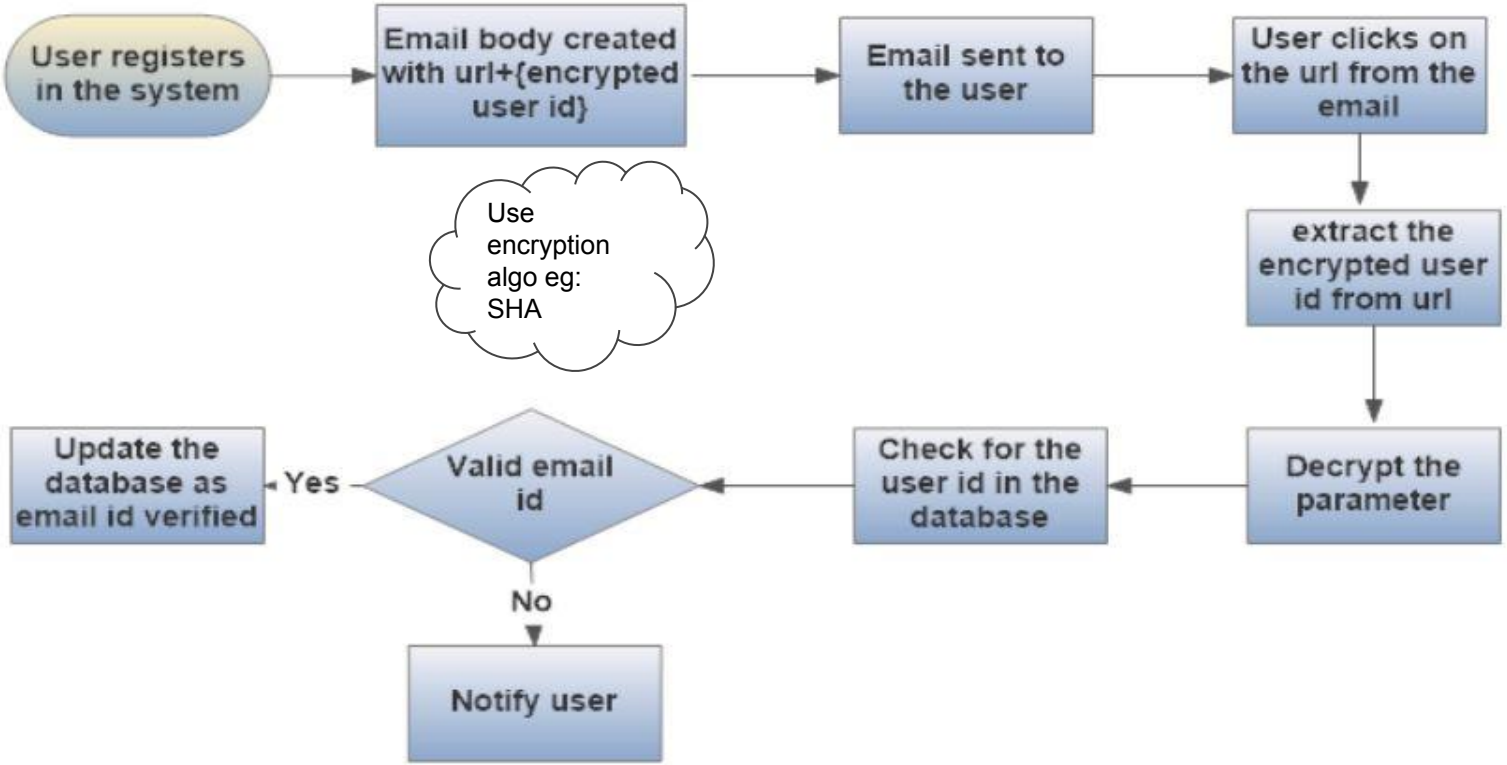
Update subscription screen



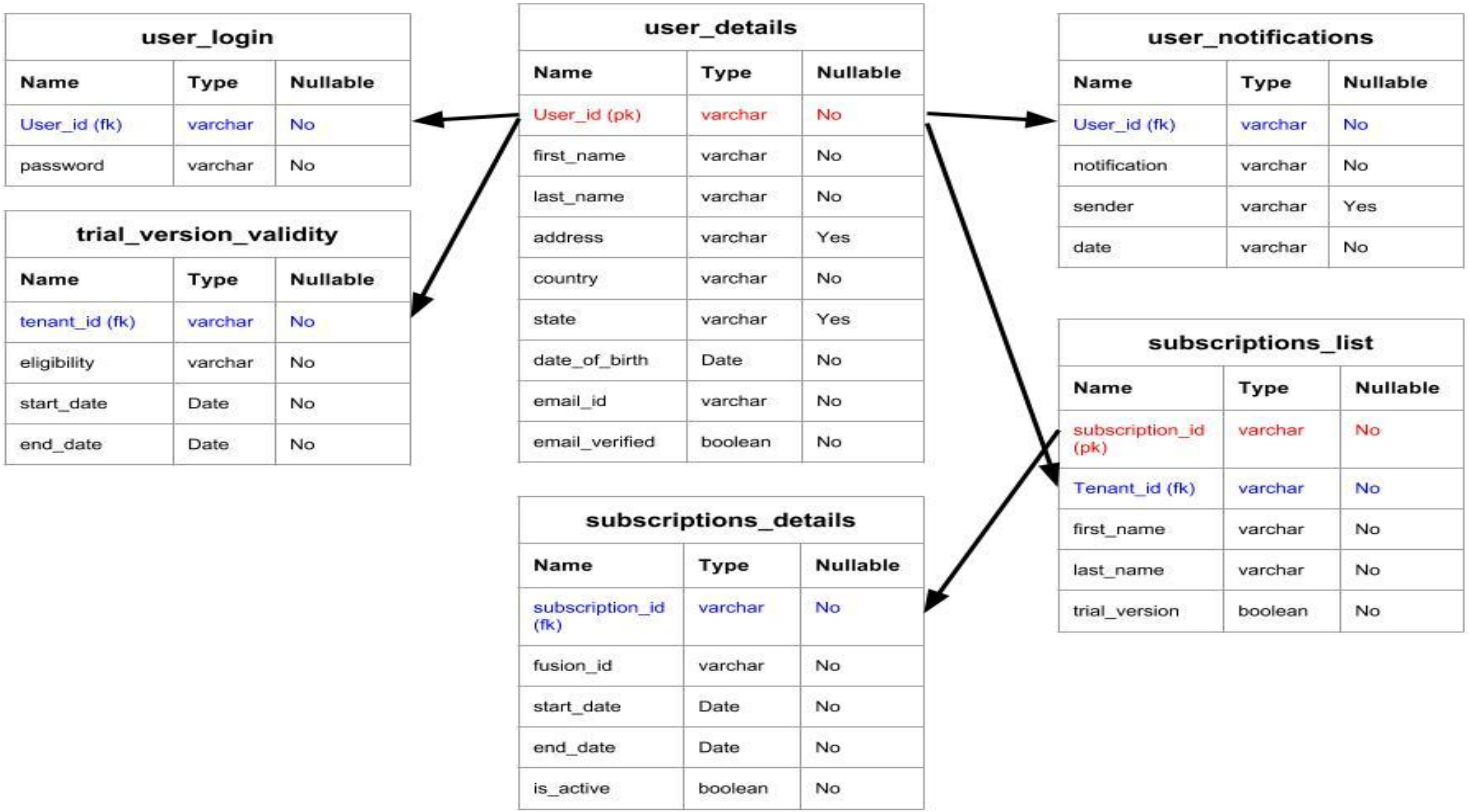
Update or extend existing subscription...



Email Id verification



Database schema



user_login

Name	Type	Nullable
User_id (fk)	varchar	No
password	varchar	No

trial_version_validity

Name	Type	Nullable
tenant_id (fk)	varchar	No
eligibility	varchar	No
start_date	Date	No
end_date	Date	No

user_details

Name	Type	Nullable
User_id (pk)	varchar	No
first_name	varchar	No
last_name	varchar	No
address	varchar	Yes
country	varchar	No
state	varchar	Yes
date_of_birth	Date	No
email_id	varchar	No
email_verified	boolean	No

subscriptions_details

Name	Type	Nullable
subscription_id (fk)	varchar	No
fusion_id	varchar	No
start_date	Date	No
end_date	Date	No
is_active	boolean	No

user_notifications

Name	Type	Nullable
User_id (fk)	varchar	No
notification	varchar	No
sender	varchar	Yes
date	varchar	No

subscriptions_list

Name	Type	Nullable
subscription_id (pk)	varchar	No
Tenant_id (fk)	varchar	No
first_name	varchar	No
last_name	varchar	No
trial_version	boolean	No

Support for Internationalization

- Support for multiple languages

  Sample code from jsp using <spring:message

```
<tr>
    <td><spring:message code="label.FirstName"/></td>
    <td><input type="text" name="firstName" /></td>
</tr>
<tr>
    <td><spring:message code="label.LastName"/></td>
    <td><input type="text" name="firstName" /></td>
</tr>
```

  Create properties for different languages

```
label.FirstName = "prénom"
label.LastName =  "nom de famille"
```

  Configure in Web.xml

Security

- **Authentication and Authorization**

  User id and Password method

  Encrypted password stored in database. Encryption algorithm used eg: SHA, MD5 etc.

- **Using third party SSL certificates**

  To validate provider identity.

  Encryption of data during transport

- **Session Tracking**

  HTTPSession

  Cookies

  Session timeout

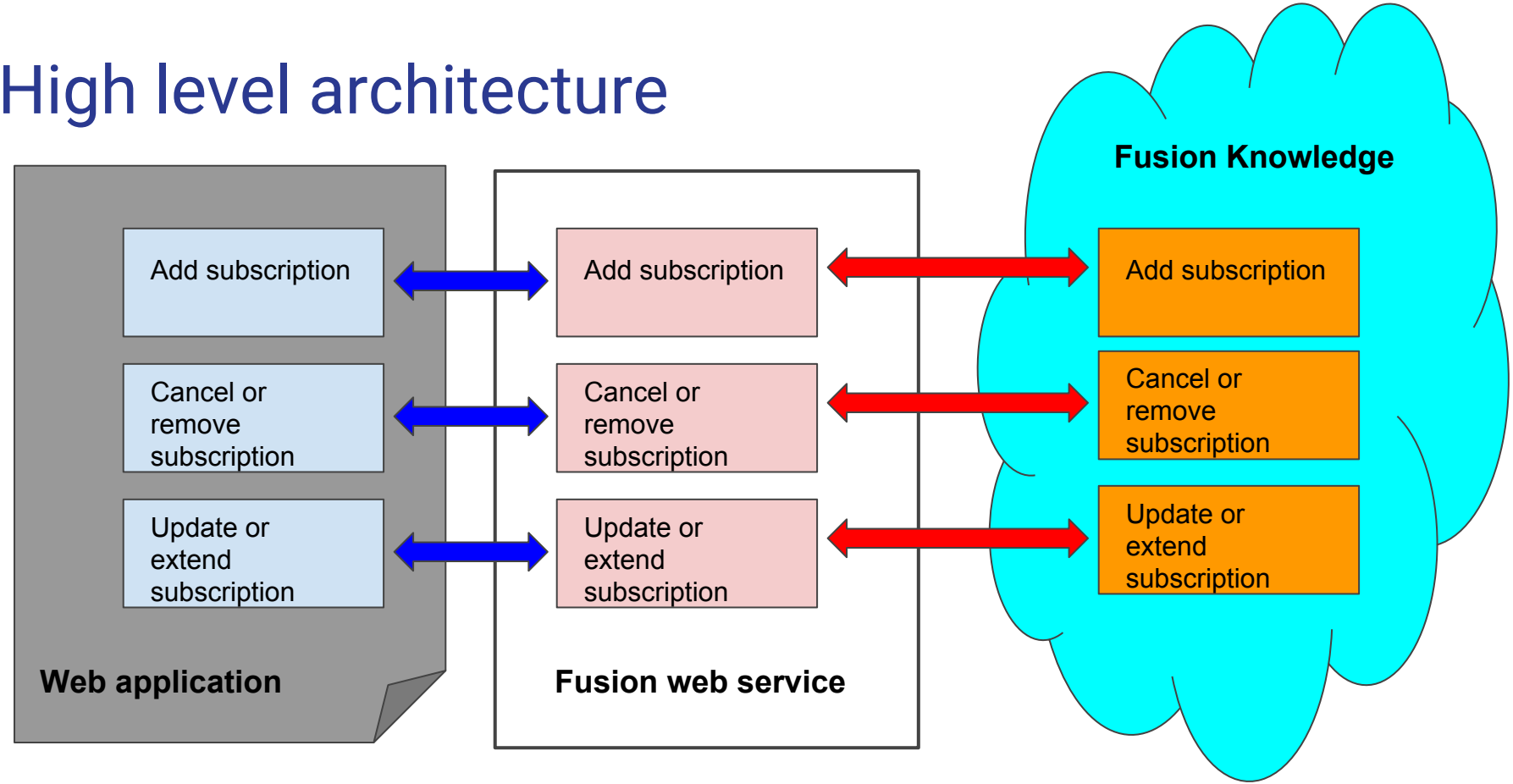
The web service

Purpose of the web service

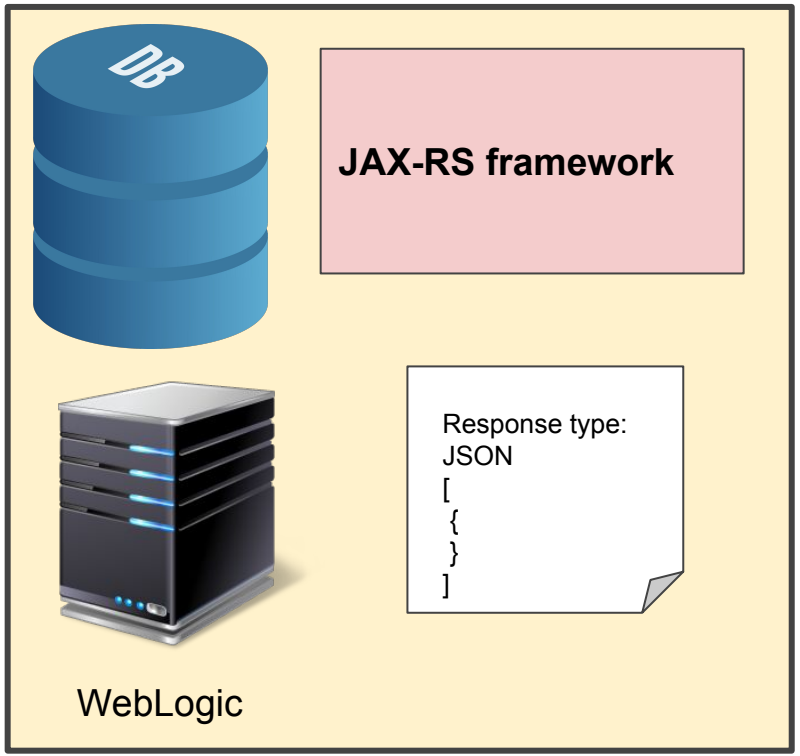
- Separation of concerns

- Future scope for web application admin

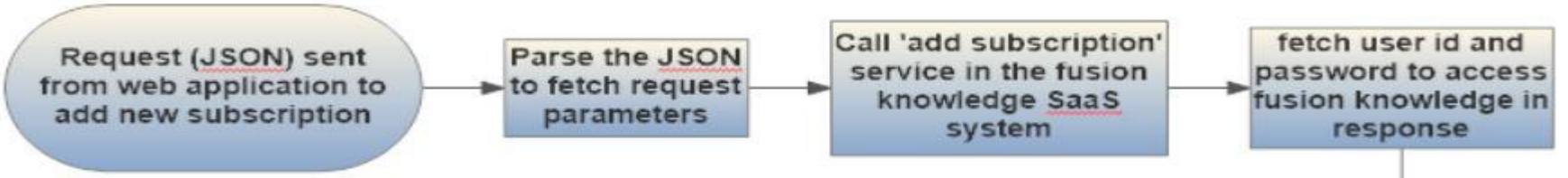
High level architecture



Components



Add new subscription



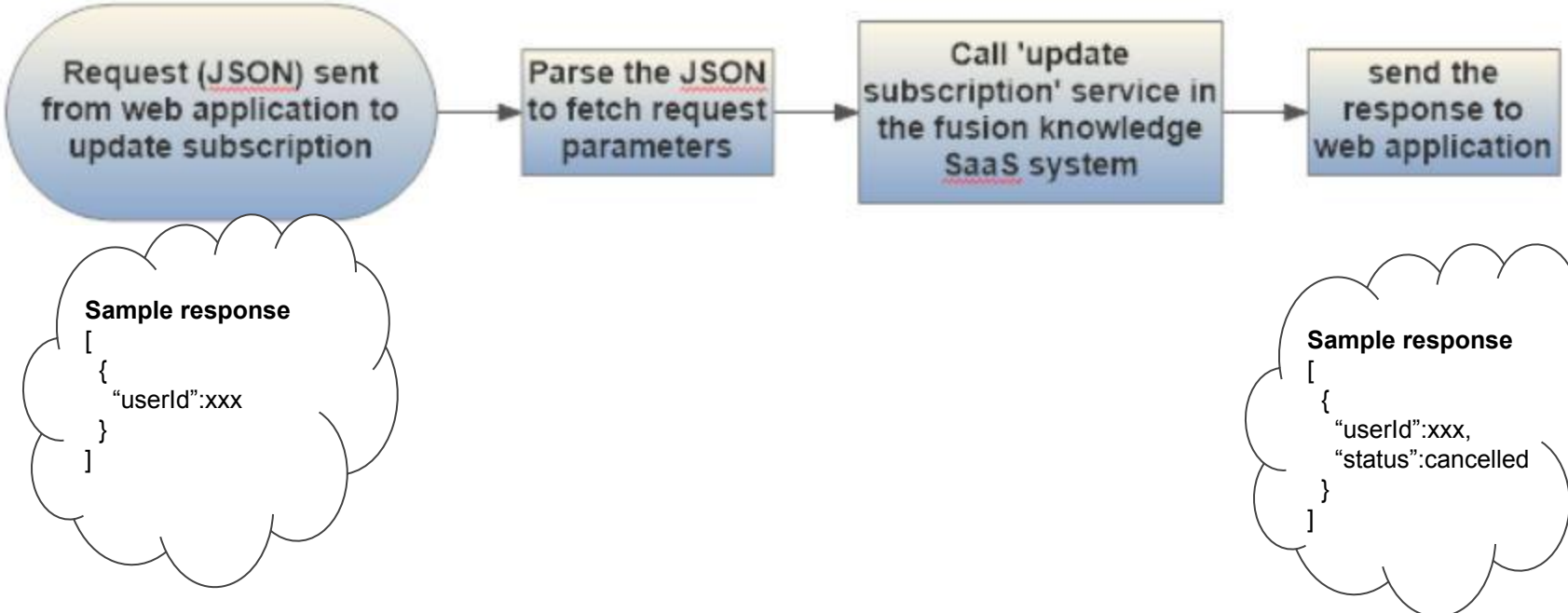
Sample request

```
[
  {
        “name”:xxx,
        “emailId”: xx@x.com,
        “start_date”:XX,
        “end_date”:XX,
        “trail_version”:true/false
  }
]
```

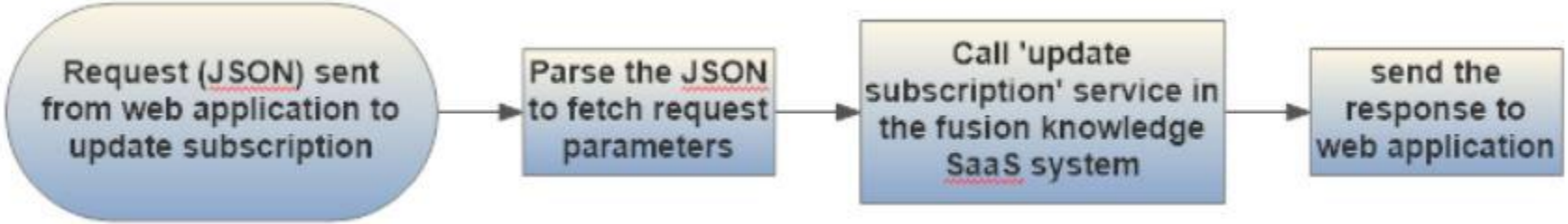
Sample response

```
[
  {
    “userId”:xxx,
    “Password”:xxxx,
    “cost”:xxx
  }
]
```

Cancel subscription



Update or extend subscription



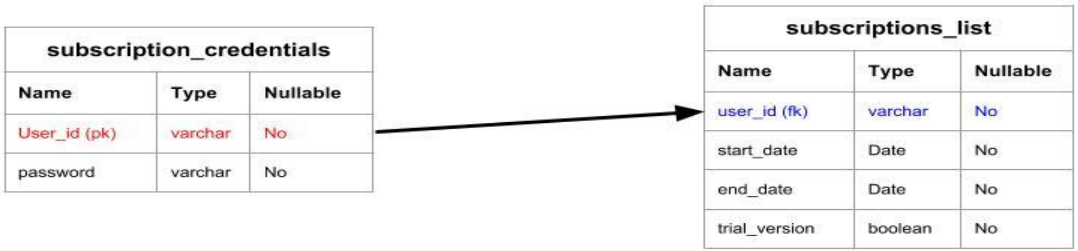
Sample request

```
[
  {
        "name":xxx,
        "emailId": xx@x.com,
        "extended_end_date":XX,
        "trail_version":true/false
  }
]
```

Sample response

```
[
  {
    "userId":xxx,
    "extension":accepted/rejected
  }
]
```

Database schema



subscription_credentials

Name	Type	Nullable
User_id (pk)	varchar	No
password	varchar	No

subscriptions_list

Name	Type	Nullable
user_id (fk)	varchar	No
start_date	Date	No
end_date	Date	No
trial_version	boolean	No

Security

- Private web service hence only be accessible to fusion cloud provisioning web application.
- Ip based authentication (accept requests from oracle owned application servers)

Maintenance

- For Weblogic server

  1 admin server to manage deployments and configurations

  1 or more managed server where the application is deployed

- Development and test environment

  A standalone domain with single single admin and managed server

Scalability and performance

- To ensure performance and high availability, more than 1 managed servers forming a cluster
- Admin server to manage the cluster
- Use of load balancer to evenly distribute jobs across servers

Challenges

Server failure

- **What if admin server fails?**

  Managed servers continue to work.

  Managed servers periodic attempt to reconnect admin server.

  Admin server restored without affecting managed servers and thus the application.

- **What if a managed server fails?**

  Clustered environment prevents performance deterioration due to server failures.

  Using **Node manager** process on every managed server.

  Features of using node manager:
  1. start and stop managed server from admin console.
  2. Kill failed server instances.
  3. Automatically restart failed managed servers.

System development plan

Date	Day	Hours	Task	Comment
5/27/2016	Friday	8	1. Finalising design. 2. Setup development environment with application servers and database. 3. Deploy Fusion knowledge SaaS application on development application server. 4. Develop web service code and deploy on server.	Weblogic application server setup with 1 admin and 1 managed server. Oracle database
5/30/2016	Monday	8	1. Setup oracle database and create required database tables. 2. Deploy in testing environment. 3. Test the web service by sending test requests. 4. fix bugs if any.	Web service development complete
5/31/2016	Tuesday	8	1. Create Login jsp. 2. Create New user registration jsp 3. Complete code for new user registration 4. Create jsp for dashboard, add, remove and update subscription.	
6/1/2016	Wednesday	8	1. Complete code for add, remove and update subscription 2. Perform integrated testing with the web service and fix bugs if any. 3. Deploy in test environment. 4. Deliver to QA or testing team	Web application development complete. Integerated testing begin.
6/2/2016	Thursday	8	1. Perform rigorous integrated testing. 2. Fix bugs if any.	
6/3/2016	Friday	8	1. Testing and bug fixing 2. Getting the code deployement ready 3. Generate necessary scripts 4. Start deployment at eod.	Code ready for deployment
6/4/2016	Saturday	Holiday		
6/5/2016	Sunday	Holiday		
6/6/2016	Monday	8	Buffer day	

Current Date	5/27/2016
Sprint start Date	5/23/2016
Sprint end date	6/62016
Team member name	Sachin Shinde
Project name	Fusion Knowledge
Version control	GIT
Defect and task logging	JIRA

Other challenges

Challenges:

- Understanding the system or currently implemented solution.
- Analyzing dependencies on system or others
- Unable to include all items in current release
- Integration with SaaS application
- QA rejection

Possible solutions:

- Read documentations
- Read docs, talk to people having related information to understand better and faster.
- Plan enhancement items for next release.
- Develop the interfacing module first (here web service). Use SaaS deployment in test env.
- To avoid this follow industry standards and best practices

Future enhancements

- Role based access with addition of admin role.
- Granting admin controls to revoke subscription.
- Enhancement for adding more services for self provisioning by the admin into the web application.

Thank you !