International Journal of Engineering and Techniques - Volume 2 Issue 2, Mar – Apr 2016

RESEARCH ARTICLE OPEN ACCESS

Efficient Multi Vendor services for Field Based Service

Madhushree M.Kubsad[1],Prof. Manu T.M[2]

1(Dept: Computer Engineer MTech, KLEIT Hubballi Karnataka, India)

Abstract:

Field service application (FSA) refers to a cloud-based system that combines the robust web application and dynamic mobile application to support field engineers. FSA most commonly caters to the customer who needs service or repairs of equipment. This application is targeted at the Service industry, intended for the field engineers. The various service industries register with this for cloud computing services for effective management of services such as painting, plumbing, Electrician, carpenter etc. This system compliments the software services, provided by the cloud, with a mobile-based client application, specially designing for the field engineers. This solution shares a single workflow among the registered tenants thereby ensuring efficient sharing of infrastructure and also ensuring the security and integrity tenant data.A Multi-tenant Application is an approach to share an application instance among different customers to reduce overhead the most.

Keywords **— Multi tenant architecture, Field service application, Cloud SaaS, PaaS**.

INTRODUCTION

This software as a service (SaaS) is targeted at the Service industry, intended for the field engineers. The service industries register with for cloud computing service for effective management of services such as painting, plumbing, etc. This application compliments the software services, provided by the cloud, with a mobile-based client application, specially designing for the field engineers. Multi tenant (vendor) application which allows the individual enterprise to use software as service. This software caters to automate workflow involved in any field base services.

I. PROPOSED METHOD

A. *Project Modules*

The modules are dividing by modules there is Presentation layer, Services and Cloud layer.

In figure 1 A presentation layer it helps to translate a data from or to a client. Here it consists two modules UI (User Interface) and REST clients.

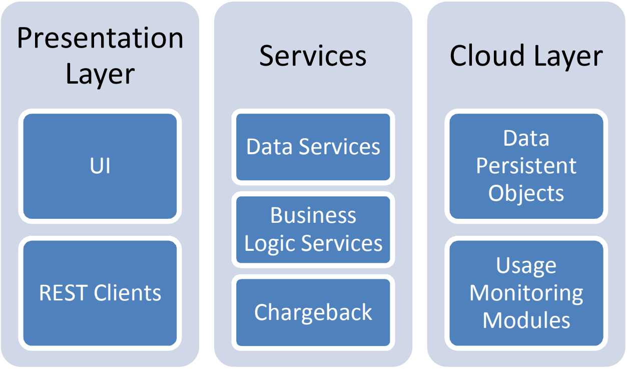


Fig 1: Three Architecture Layers

- **User Interface**: A User Interface (UI) is the place where human beings interact with the devices for Request and Response.
- **REST Clients:** REST Clients it defined as Representational State Transfer (REST) is the software architectural style of World Wide Web. It supports all HTTP request and HTTP responses like GET, POST, PUT, and DELETE.

In Services, it consists of three modules

- **Data Services**: Data Services delivers a single enterprise-class solution for data integration, data quality, data profiling, and text data processing that allows to integrate, transform, improve, and deliver trusted data to critical business processes. It provides one development UI, metadata repository, data connectivity layer and run-time environment.

- **Business Logic Services**: Business logic is also known as domain logic is the part of the program that encodes the real-world business rules that determine how data can be created, displayed, stored, and changed.
- **Chargeback**: Chargeback is the return of funds to a consumer, mainly used in the United States, forcibly initiated by the issuing bank of the instrument used by a consumer to settle a debt. Specifically, it is the reversal of a prior outbound transfer of funds from a consumer's bank account, a line of credit, or credit card.

In Cloud Layer, it consists of two modules

- **Data Persistent Objects**: Persistent objects are the fundamental logical units of data storage in Objectivity. Persistent objects are created by applications. Each persistent object has an object identifier (OID) and is an instance of a class. The class may be defined in an object-oriented programming language, or it may be a class defined by Object.
- **Usage Monitoring Modules**: The cloud usage monitor mechanism is a lightweight and autonomous software program responsible for collecting and processing IT resource usage data. Depending on the type of usage metrics they are designed to collect and the manner in which usage data needs to be collected, cloud usage monitors can exist in different formats.

B. Architecture Design

The architecture diagram shows the entire design of system by layers. The system architecture is divided into four layers as Presentation layer, Service layer, and Storage layer.

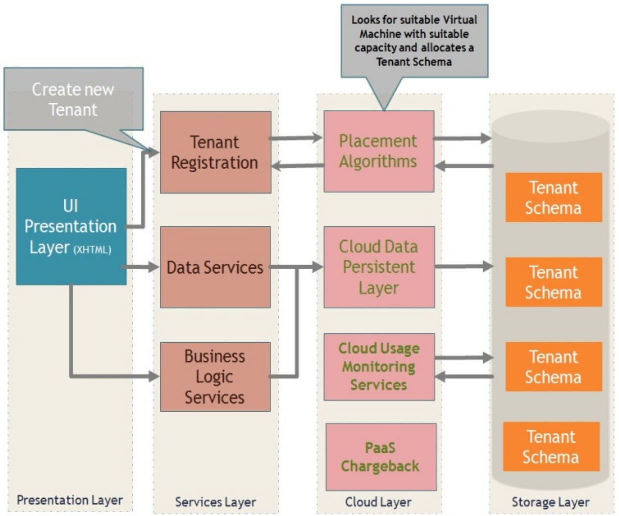
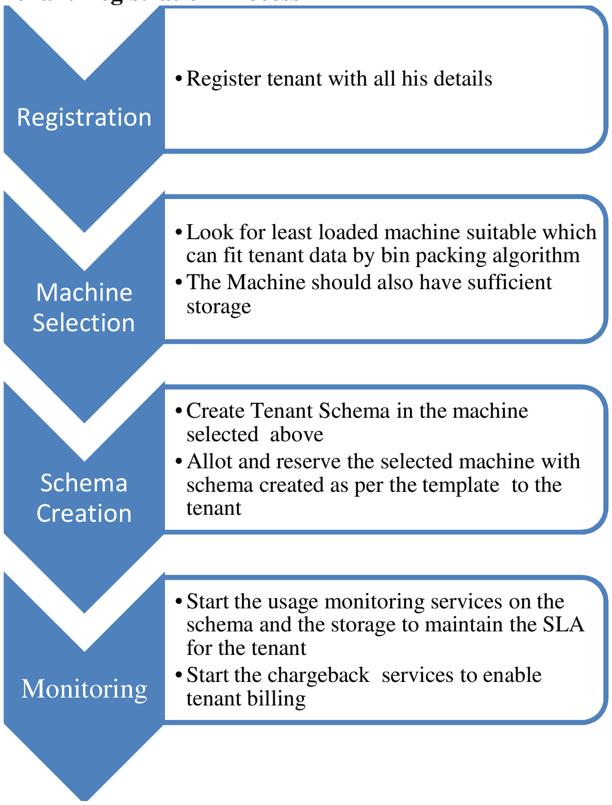


Fig 2: Architecture Design

In the presentation layer as explained earlier, it has two modules User Interface (UI) and REST Client. UI helps to interact with user and REST client are help to do for basic operation of users like GET, POST, PUT and DELETE. A presentation layer is connecting with the Service layer modules. A user interface helps to get a request from the user and to display results to the user.

In the service layer services are getting from the presentation layer and this layer also connect with the cloud layer modules. In this layer main thing is Tenant Registration. A multi-tenant can be register with the unique identification number and when a request come for tenant registration it use placement algorithm to place tenant information into the cloud.

Tenant Registration Process


Registration
• Register tenant with all his details
Machine Selection
• Look for least loaded machine suitable which can fit tenant data by bin packing algorithm
• The Machine should also have sufficient storage
Schema Creation
• Create Tenant Schema in the machine selected above
• Allot and reserve the selected machine with schema created as per the template to the tenant
Monitoring
• Start the usage monitoring services on the schema and the storage to maintain the SLA for the tenant
• Start the chargeback services to enable tenant billing

Data service is used to deliver a single enterprise-class solution for data integration, data quality, data profiling, and text data processing that allows to integrate, transform, improve, and deliver trusted data to critical business processes. Business logic is the part of the program that encodes the real-world business rules that determine how data can be created, displayed, stored, and changed.

In cloud layer, it consists of placement algorithm, Cloud Data Persistent Layer, Cloud Usage Monitoring Services and PaaS Chargeback. A placement algorithm Looks for suitable Virtual Machine with suitable capacity and allocates a Tenant Schema. In our project we use a Bin Packing algorithm as placement algorithm it helps to store data with a suitable capacity of the bin.

In storage layer, it stores Tenant schema in the cloud database with the suitable capacity of database size.

IV. FLOW DIAGRAM

A. *Tenant Registration:*

In this module, the tenant is allowed to register by creating his own ID. The tenant can register with the details of the tenant, like the information about the services offered by him (E.g. Plumber, carpenter, painter, electrician, etc) Company Addresses, Customers contacts and his account book keeping system details.

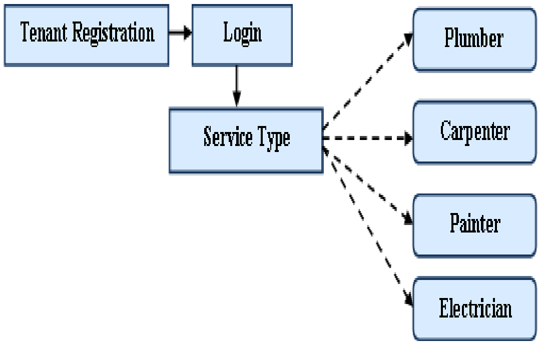


Fig 3: Tenant Registration

B. *Service Order:*

Figure 4 depicts the single workflow that is shared across all tenants. It starts with the service registration.

i) Service Registration: At the point of registration, a service order is created, and assigned to a service engineer. Once the SO is created with a unique SO number, and a mapping customer order number, the SO is created in the system.

ii) SO assignment: The privileged user (supervisor) can assign the SO to an engineer, with a unique engineer ID- the selection of engineer assisted by the algorithm based on the capacity of the engineer. The SO has to be pushed to the client application resident in the hand-held device of the engineer as a push notification.

iii) The application is capable of capturing the site location (co-ordinates long., and lat., on the map) so that the field engineer can locate the site of the service request.

iv) Once the user gets the notification, the field engineer visits the service location and does a field survey to record the issues, spare parts required to fix the issue and finally the labour charges to fix the issue.

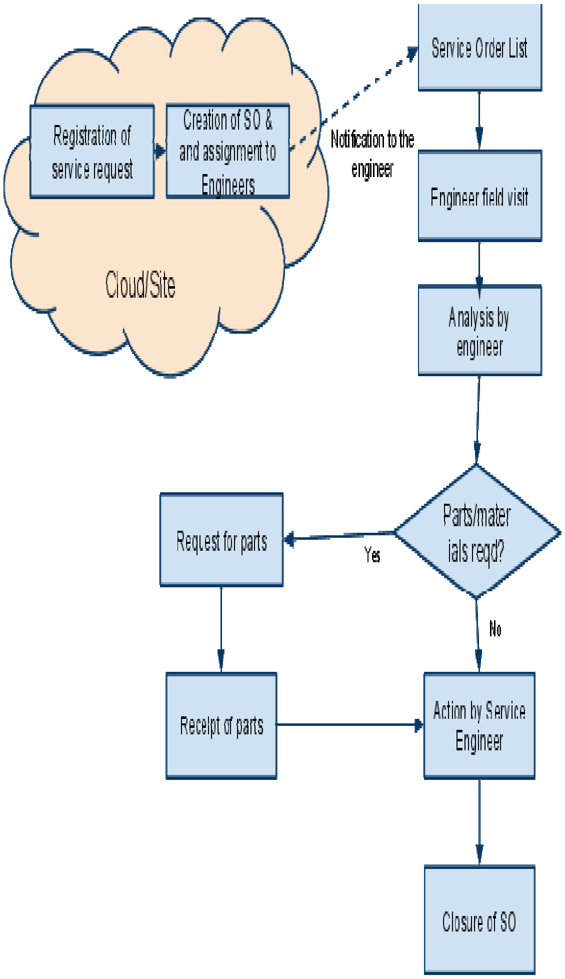


Fig 4: Service Order

v) Once the user replaces the spare parts and fixes the issue, he triggers an invoice generation shared module which calculates the tenant specific item-wise charges, sub-total and tenant-specific applicable taxes.

vi) Post-Invoice generation, the tenant can opt to integrate the invoice data with one of a pre-configured set of cloud-based account systems.

V. ALGORITHM

As explained in architecture design, placement algorithm for field service application uses a BIN PACKING algorithm.

The bin packing algorithm helps to find a schema (container/table space) for the tenant, there by fitting the content neatly and efficiently way in the cloud.

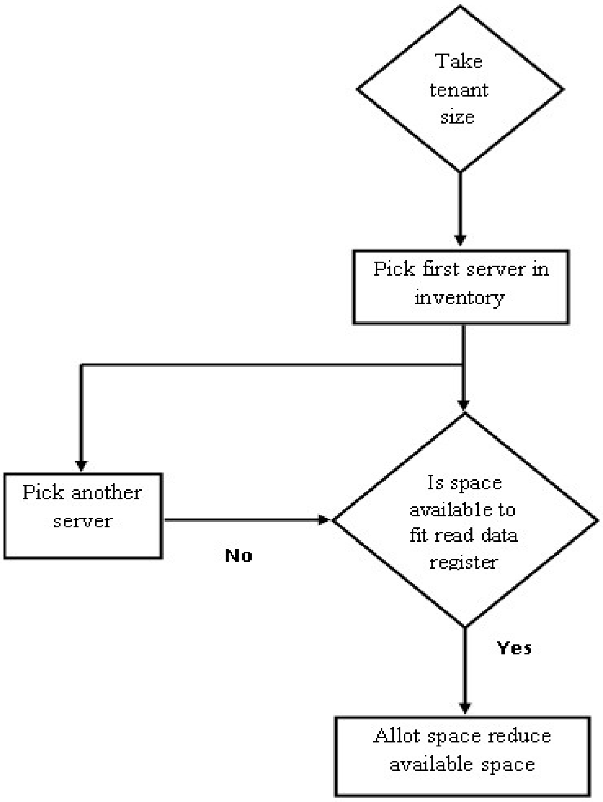

Take tenant size
Pick first server in inventory
Is space available to fit read data register
Pick another server
No
Yes
Allot space reduce available space

Fig 5: Flowchart of Algorithm

In this application, we use first fit Bin packing algorithm. As and when new tenants have registered the dataspace for the tenant's application space is calculated and allotted to the tenant based on the need determined at the time registration.

In particular, the dataspace usage needs of a tenant are characterized into one more categories. On the other hand, the cloud monitoring services would have capacity details in hand.

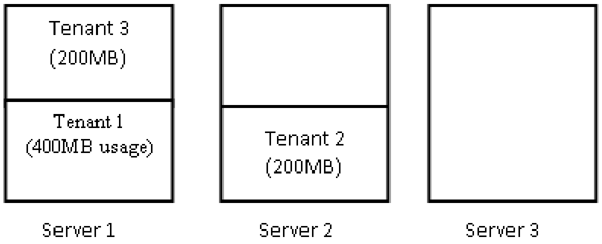


Fig 6: Example of Bin Algorithm

For example: Server (considered as a bin) Max Size is 500MB, randomly first tenant data come up with the requirement of 400MB and second tenant data comes with a data requirement of 200MB and third tenant data come with a requirement of 100MB. Then first tenant data is placed in first server 500MB-400MB=100MB and a 100MB place is left free. The next i.e. second tenant data comes up, it checks the first server to be having insufficient disk space for the second tenant. Hence, it would be placed on the third server. Next, when the third tenant comes up, the algorithm would again check the first server and finds that there is sufficient place for the third tenant so third tenant data is placed on the first server. Here first bin is full with 500MB and there is no wastage of disk space.

Figure 5 depicts the flowchart of the space finding and allocation. Another figure 6 shows the status of servers at the end of placement of all the three tenants above.

CONCLUSIONS

In this paper, we introduced Field service application using multi tenant architecture. It helps us too utilise cloud space efficiently. This application provides customer service like plumbing, painting, carpenter, electrician, etc.

ACKNOWLEDGMENT

This research paper is made possible through the great help and support from everyone, including my: parents, teachers, family, friends, and in essence, all sentient beings.

REFERENCES

1. *Rouven Krebs, Christof Momm and Samuel Kounev "Architectural Concerns in Multi-Tenant 1 SaaS Applications", SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany, 2013*

2. *Parul Kashyap and Rahul Singh, "Crypto Multi Tenant: An Environment Of Secure Computing Using Cloud SQL", Uttar Pradesh, India, IJDPS Vol.5 May 2014*

3. *Richard E. Korf, "A New Algorithm for Optimal Bin Packing", University of California, AAAI-02 Proceedings, Feb 2002*

4. *Piyush Aghera, Sanjay Chaudhary and Vikas Kumar, "An Approach to Build Multi-Tenant SaaS Application With Monitoring and SLA", IEEE, 2012*

5. *György Dósa and Jirí Sgall, "First Fit bin packing: A tight analysis", Leibniz International Proceedings in Informatics Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany, pp. 1–15.*

6. *Sunil Kumar Khatri, Himanshu Singhal and Khushboo Bahri, " Multi-Tenant Engineering Architecture in SaaS", International Journal of Computer Applications International Conference on Reliability 2013*

7. *Karabulut, Y. and Nassi, I. 2009. Secure Enterprise Services Consumption for SaaS Technology Platforms. In Proceedings of the 2009 IEEE international Conference on Data Engineering (March 29 April 02, 2009) . ICDE. IEEE Computer Society, Washington .*