An Approach to Build Multi-Tenant SaaS Application With Monitoring and SLA

By
Piyush Aghera
Sanjay Chaudhary

Outline

- Introduction to SaaS
- SaaS design parameters
- Possible configurability and way to develop SaaS
- Proposed architecture
- Database for SaaS
- Memcached flow
- Application logic flow
- References

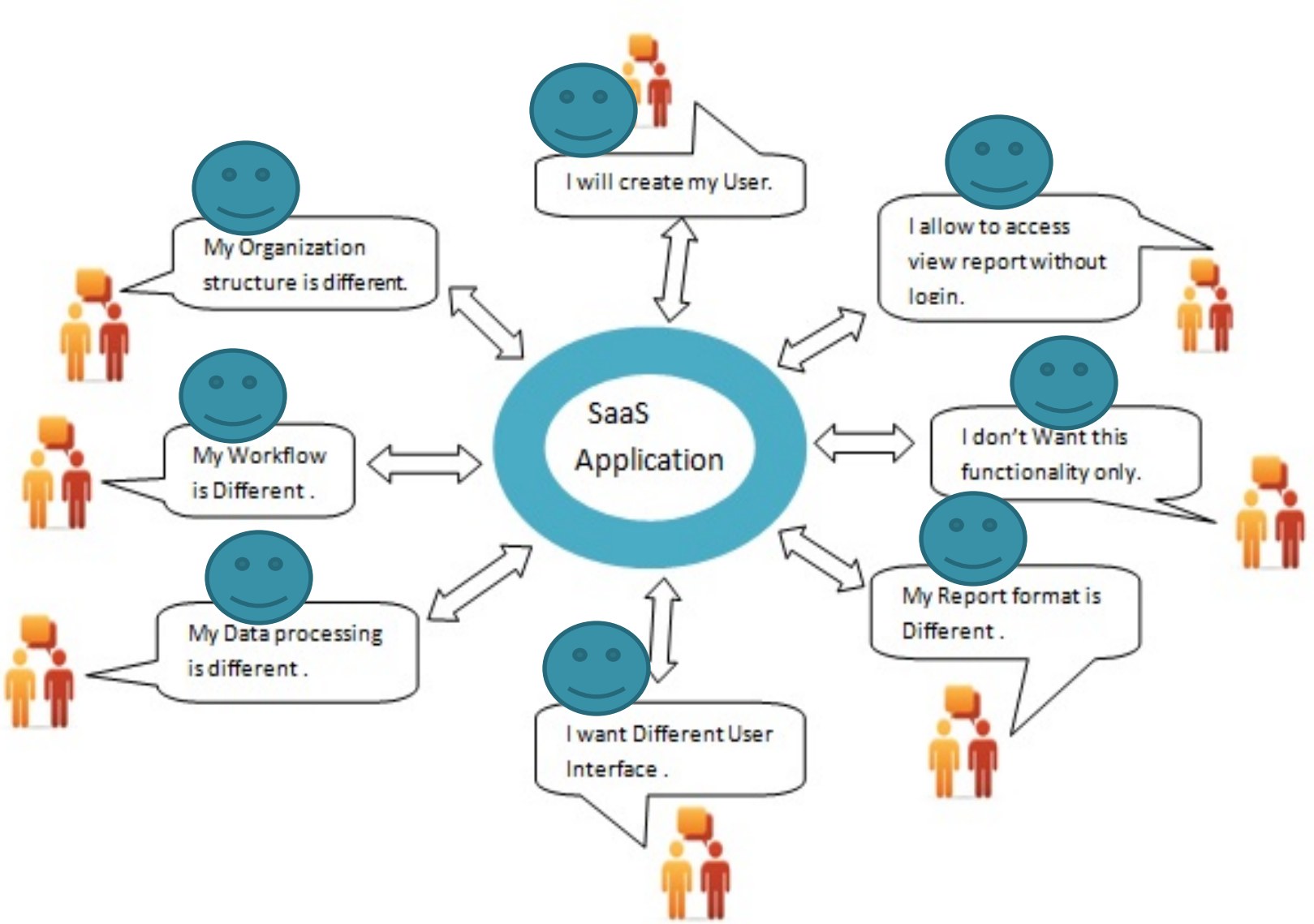
Software as a Service[1,2]

- Software as a Service (SaaS) model is both a software delivery model and business model .The SaaS delivery model involves delivering an application as a service, over the Internet.
- It provides service to multiple organizations with configuration functionality.
- SaaS product is developed as a web application and hosted on large scalable infrastructure cloud .
- SaaS users gets service without installing or maintaining any software on their premises .
- Users will pay based on the usage and functionality they have subscribed.

SaaS Design Parameters[3]

- Multitenancy
- Customizable GUI
- Customizable business logic
- Subscription
- Monitoring and billing
- Security and privacy
- Scalability, high availability and reliability
- Managing and administration for separate tenant
- Runtime per tenant customization

SaaS Challenges[9]



Possible Ways to Develop SaaS

- Current PaaS Provider for SaaS development.

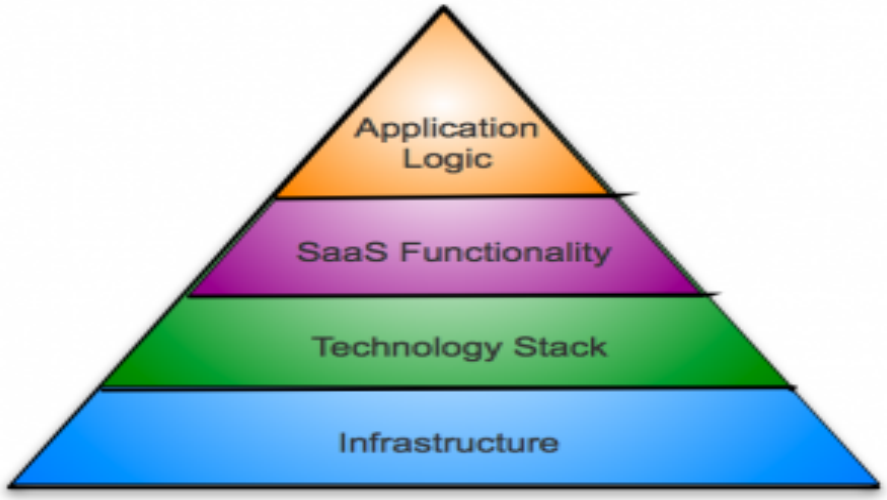
  [Apprenda's SaaSGrid {.NET}, Force.com{ApexCode}, Zoho Creator, MS Azure , Google App Engine etc..]

- Using core technology on own infrastructure.

  [like PHP, Java , .net with any scalable database.]

  { I am doing this }

What is PaaS Provider for SaaS[10]



SaaS Plumbing

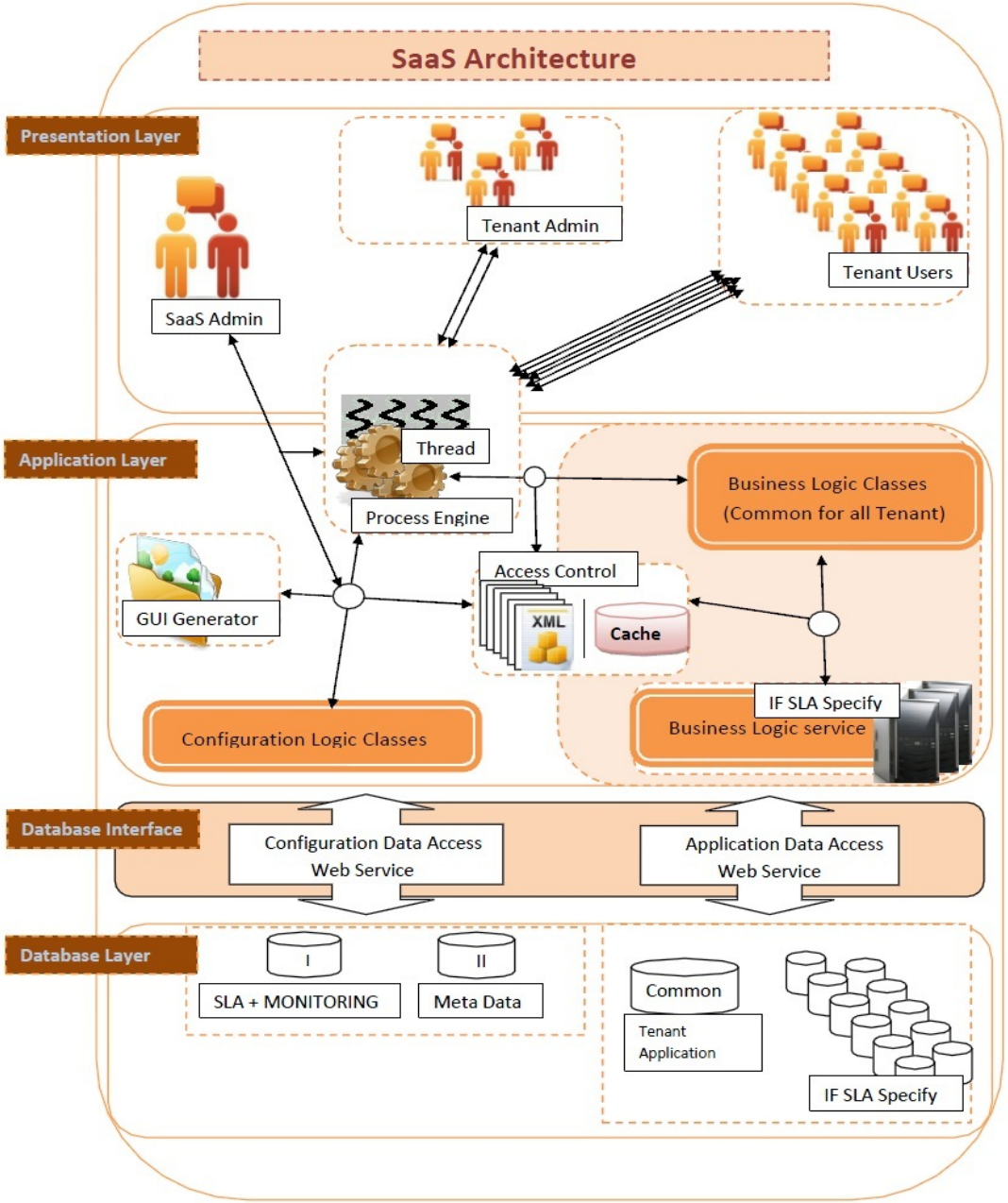
{Pricing , Billing & Payment Processing, Tenant & Subscription Management , Service Provisioning , Usage & Performance Monitoring , Subscriber Management & Self-Service}

- **Infrastructure** [Network, Servers, Storage, Load Balancing, etc.]
- **Technology Stack** [OS (Windows, Linux, etc), Database, Application Environment (.NET, Ruby, PHP, etc) and associated services.
- **SaaS Functionality** – The "SaaS Plumbing".
- **Application Logic** – The core business rules and functionality of the service to tenants.

Module Developed and Working

- Multitenancy — Developed
- Customizable GUI — Developed
- Customizable business logic — Working
- Subscription — Developed
- Monitoring and billing — Working
- Security and privacy — Partially DONE
- Scalability, high availability and reliability — Working
- Managing and administration for separate tenant — Developed
- Runtime per tenant customization — Developed

Proposed SaaS Architecture



Databases

- In metadata driven architecture, database plays dual role of user application data and configuration metadata.
- Every running moment of application has to retrieve metadata to understand user data and actions to be performed.
- We have divided database in the form of three instances based on its data and access frequency.

- Metadata database
- Application database
- SLA and monitoring database

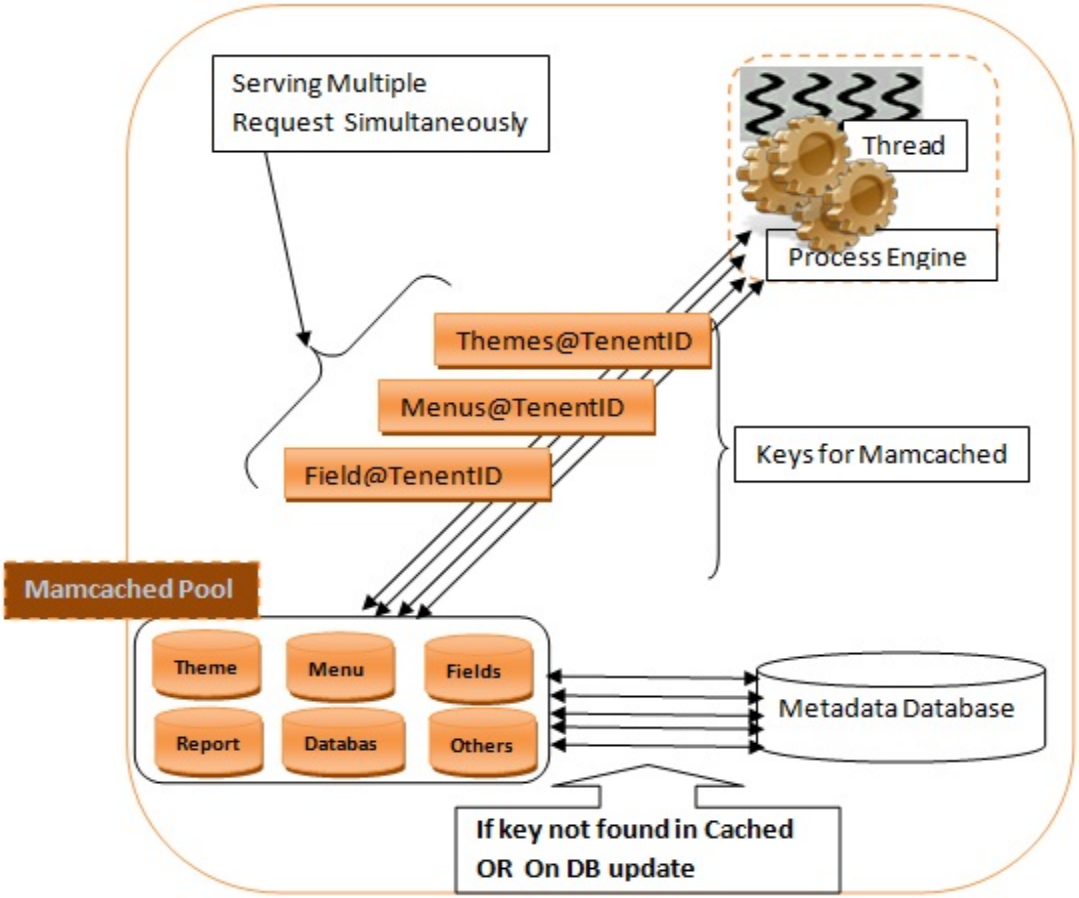
Metadata Database

- Metadata database is designed to store configuration information based on tenant specific requirements.
- It plays significant role to overall performance.
- Some of basic required metadata are the theme, menu, modified functionality, fields in forms, report.
- Configuration in XML file may suffer from performance point of view due to bottleneck of file read-write operation.
- Here , I used memcached for efficient configuration information retrieval simultaneously among tenants.

Memcached[11,12]

- Memcached is an in-memory key-value store for small chunks of arbitrary data (strings, objects) from results of database calls, API calls, page rendering.
- Memcached is likely to be faster than file operation.
- Memcached is not actual metadata storage for an application. It’s an interface to actual database.

Memcached Flow



Application Database

- The size of application database is very large as compared to configuration database.
- Application database can be stored either in the form of shared or dedicated separate database instance.
- Tenant database will remain in common instance by default.
- Extension in shared instance are preallocated fields and name-value pairs.

Preallocated Fields and Name-Value Pairs

Tenant ID	FirstName	Email	Ext1	Ext2	Ext3
1203	Abc	Abc@SaaS.com	Null	Gold	Null
1504	Pqr	Pqr@SaaS.com	Black	Null	Null
1560	Xyz	xyz@SaaS.com	Null	Null	Gold
...	...	....	....	...	...

Tenant ID	Fild Name	Extended field	Data type
1203	Service Status	Ext2	String
1504	Hair color	Ext1	String
1560	Ornament	Ext3	String
...	...		....

Tenant ID	FirstName	Email	RecordID
1203	Abc	Abc@SaaS.com	Null
1504	Pqr	Pqr@SaaS.com	566
1560	Xyz	xyz@SaaS.com	856
...	...	....	....

RecordID	ExtensionID	Value
856	1200	450
566	1201	Blue
566	1202	Gold
...	...	

Configuration Data

TenantID	ExtensionID	Lable	Datatype
856	1200	Quantity	Int
1504	1201	Eye Color	String
1504	1202	Ornament	String
...	...		

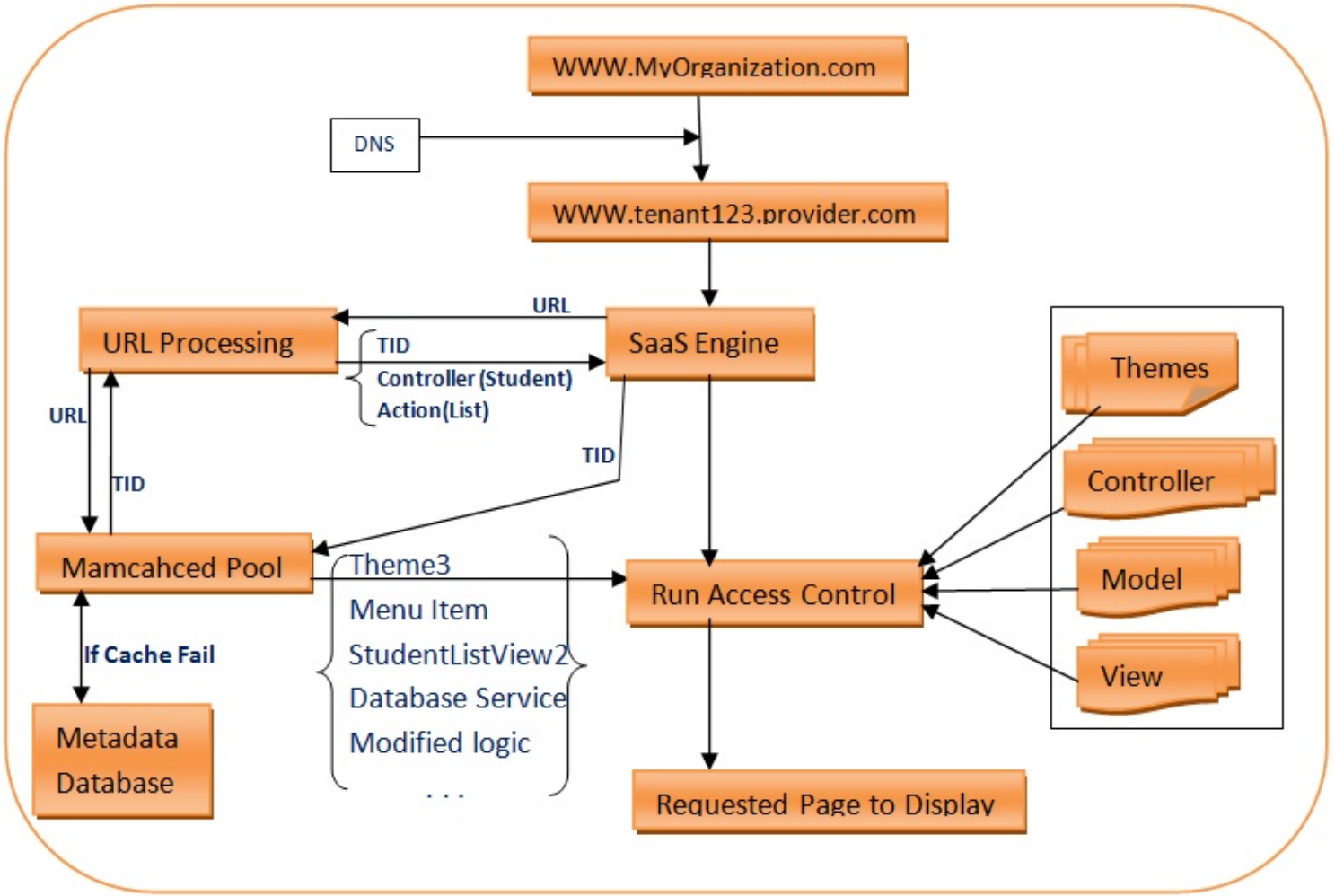
SLA and Monitoring Database

- SLA (Service Level Agreement) defines formal contract between a SaaS provider and a consumer.
- SLA database stores some of technical agreements like dedicated instance of a database or a shared instance of a database, accessible service package, unlimited number of users etc.
- Monitoring SaaS in one kind of logging system is to keep track of tenant usage
- Keep record for each tenant like number of active users, total service hours, login and logout, number of hits for an application.
- Also keeps record for database backup and restore.

Web Service Interface

- Web service is given as an interface for database access.
- Web service will be implemented in a such a away that it takes tenant id as a common parameter to access data.
- Web service contains all possible database transactions like insert, update and delete in the form of web service functions.
- Web service is selected as database interface to hide direct access to a database server.

Application logic flow



Application Pooling

- If large number of tenants access SaaS at the same time then it is required to implement isolation of processes to sustain performance of an application.
- Some of the tenants may require to have higher priority than other tenants.

Developed SaaS Module .

- Database design for metadata and application database.
- Web service interface that can be hosted anywhere independent of application.
- SaaS engine implementation for URL processing.
- Configuration control panel.
- SaaS functionality for university management system.

SaaS Demo for Multitenant

- Tenant-1 : http://saasprovider/Saas/
- Tenant-2 : http://daiict/Saas//
- Tenant-3 : http://nirma/Saas/
- Tenant-4 : http://daiict/Saas//
- Tenant-5 : http://iitb/Saas//
- Tenant-6 : http://Piyush/Saas/

Conclusion

- Architecture showed an easy way to develop SaaS application without using proprietary framework.
- Small enterprise SaaS developers need not be dependent on private framework for cloud.
- SaaS application will also be independent of underlying infrastructure and can be hosted at any private or public cloud .

REFERENCES

[1] "Software as a service," http://en.wikipedia.org/wiki/Softwareas service.

[2] "What is saas?" https://www.salesforce.com/saas/.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph,R. H. Katz, A.Konwinski, G. Lee, D. A. Patterson,A. Rabkin, I. Stoica, and et al., "Above the clouds A berkeley view of cloud computing," Computing, no. UCB/EECS-2009-28, 2009. [Online]. Available: http://www. Eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009- 28.html

[4] G. Liu, "Research on independent saas platform," in Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on, April 2010, pp. 110 –113.

[5] J. Ju, Y. Wang, J. Fu, J. Wu, and Z. Lin, "Research on key technology in saas," in Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on, june2010, pp. 384 – 387.

[6] J. Lee and S. J. Hur, “Level 2 saas platform and platform manage - ment framework,” in Advanced Communication Technology (ICACT), 2011 13th International Conference on, Feb 2011, pp. 1177 –1180.

[7] C. D. Weissman and S. Bobrowski, “The design of the force.com multitenant internet application development platform,” in Proce- edings of the 35th SIGMOD international conference on Manag- ement of data, ser. SIGMOD’09. New York, NY, USA: ACM,009, pp. 889–896. [Online]. Available: http://doi.acm.org/ 10.1145 /1559845.1559942

[8] W. Liu, B. Zhang, Y. Liu, D. Wang, and Y. Zhang, “New model of saas: Saas with tenancy agency,” in Advanced Computer Control (ICACC), 2010 2nd International Conference on, vol. 2, March 2010, pp. 463 –466.

[9] W. Sun, X. Zhang, C. J. Guo, P. Sun, and H. Su, “Software as a service: Configuration and customization perspectives,” in Congress on Services Part II, 2008. SERVICES-2. IEEE,Sept. 2008, pp. 18 – 25.

[10] "SaaS:All PaaS are Not Create Equal" http://blog.sciodev.com/ 2009/03/27/saas-all-paas-are-not-created-equal/

[11] "Memcached," http://memcached.org/.

[12] "Memcached," http://code.google.com/ appengine/articles/ scaling /memcache.html.

[13] "Application database,," http://msdn.microsoft.com/enus/ library/ aa479086.aspx.

[14] C. T. O. Daniil Fishteyn, "Deploying Software as a Servic " Web-Apps , Inc. a.k.a. SaaS.com, Tech. Rep., 2010.

[15] "Building a SaaS Business Top 10 Issues," SaaS-Attack, Tech. Rep., 01 2010.

[16] W. Li, Z. Zhang, S. Wu, and Z. Wu, "An implementation of the saas level-3 maturity model for an educational credit bank infor - mation system," in Service Sciences (ICSS), 2010 International Conference on, May 2010, pp. 283 –287.

Thank you …