



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

**CLOUDCONNECT EVENT – BENGALURU PRESENTS AN EXCLUSIVE PAPER ON:-**

# **Different approaches to build a multi-tenant SaaS application**

**BY Ramkumar, Director Product Management, Techcello**

From using PaaS platforms such as Force.com to complete ground-up development, there are many ways to build or migrate to a multi-tenant SaaS Architecture. Explore the various decision factors involved in choosing a suitable alternative.

Ramkumar will be speaking at Cloud Connect Bengaluru, and will present the same topic.

To attend his session, register now at Cloud Connect Bengaluru

---

Website: [WWW.CLOUDCONNECTEVENT.IN](http://WWW.CLOUDCONNECTEVENT.IN)

Conference agenda: [http://www.cloudconnectevent.in/conference\\_agenda.asp](http://www.cloudconnectevent.in/conference_agenda.asp)

Speaker List: [http://www.cloudconnectevent.in/speaker\\_list.asp](http://www.cloudconnectevent.in/speaker_list.asp)

Register now: [http://www.cloudconnectevent.in/conference\\_registration.asp](http://www.cloudconnectevent.in/conference_registration.asp)



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

## **Multiple Ways to Build a Multi-tenant SaaS Application**

### **1. Cloud Ready Multi-tenant Application Development Platform (CMAP)**

Cloud Ready Multi-tenant Application Development Platform (CMAP) refers to a suite of technology that can help enterprises and ISVs in designing, developing and deploying Multi-tenant SaaS applications on the Public or Private Cloud.

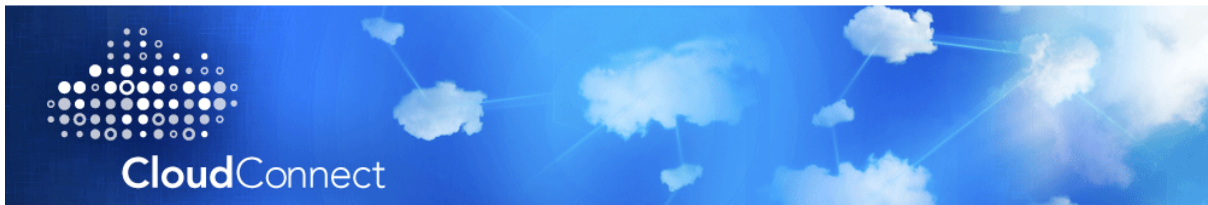
In the absence of CMAP , enterprises and ISVs end up using multiple technologies, architectures and engineering structures thus leading to issues in application maintenance and operational bottle necks. Lack of Multi-tenancy, Scalability and Configurability in the application results in huge inefficiency, cost over-run on application management and slower response to business needs.

Good choice of CMAP brings in the agility and ability to respond to fast changing business requirements in the IT world. Now, enterprises and ISVs have started realizing the need to migrate to a unified and strong CMAP which can act as their technology backbone and engineering foundation for all existing and future applications.

#### **1.1 Typical Requirements of an CMAP**

Following are some of the typical requirements/expectations of an enterprise or ISV from an CMAP ,

- Predefined architectural style for building applications better and faster.
- Global support for technologies used in CMAP
- Flexibility in designing technical solutions
- Availability of skill sets and resources to work on the CMAP
- Uniformity in the way applications are built and managed
- Ease of maintenance
- Support for web technologies
- Support for On-premise as well as Private and Public Cloud Deployment
- Support for building scalable, flexible and configurable applications easily
- Support for Multi-tenancy
- Support for Customizing and Configuring the application to suit the needs of different tenants (customers) while still maintaining a single code base.



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

- Tools that can facilitate or ease development effort

## 1.2 Technology Stacks

Many application development teams are using a combination of tools and technologies to overcome the complexity and inflexibility of applications. BPM is used to quickly define and change business processes, and collaboration suites like SharePoint and Lotus are used to respond to the increasing demands of long-tail apps. Progress Software's responsive process management (RPM) combines the best of BPM and business events to help businesses respond to real-time events and change business processes. This is just a small sampling of the next generation of business application development tools used by Enterprise IT. However conventional technology stacks such as .NET and J2EE, are still the dominant option used by most development teams, particularly in the SaaS products space.

## 2. Why CMAP ?

The estimated cost of software failures is estimated to be around 50 to 80 billion dollars per year. Reports also say that 3 out of 5 IT projects do not do what they were supposed to for the expected costs. While there are several factors that could contribute to this scenario, one of the major contributors is the lack of undivided focus on business functionalities on the one hand and application engineering quality on the other. The team's time is often diffused and spread across both these objectives. In many cases, development teams are tasked with building and maintaining the entire solution without proper support on architecture, design and engineering front. Following are some of the reasons why this will not work,

- **Developers don't know where to start when building an application** - Often a developer allows the user interface (UI) to drive the business requirements and begins to code. A typical mistake is that the UI developer goes too far when extending the presentation logic with actual business functionality. It then becomes difficult to de-couple the functionality and share it across the application, or the developer gets backed into a corner when faced with more difficult issues like data persistence.
- **Developers have difficulty with the steps needed to successfully build an application** - Even if a development team has been sufficiently trained in the basic technology, they're still left a blank slate when development begins. This opens up questions like "Where do I start?" "How does all of this technology – MVC, Silver Light, Cloud,.NET - help me with what I'm building?" "How do I glue it all together to get to my end application?" Technology often leaves the developer with more options than answers.



MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU

- **Developers are always under time pressure** – Developers are always under constant pressure to deliver the customer requests, and they don't get too much time to step back and think about the right way of doing things in the form of refactoring, redesign or re-architecting. This leads to a scenario where the developers only aim is to complete a given ticket/issue rather than doing a holistic thinking of the situation.
- **Many developers are solving the same technical problems again and again**- You would typically see the developers solving similar issues again and again. For example, every developer would write separate caching mechanisms, different loggers, etc. In the absence of a standard development architecture and environment, developers would be duplicating the efforts.
- **Project managers don't know what to expect from developers** - Take, for example, a developer who tells his or her manager that it will take two months to develop the "user flow experience." How does the project manager judge this answer when he or she doesn't have insight into the technical decisions to a working application?
- **Project managers can't easily assign work based on developers' skills, as the delineation of work is often nebulous** - How do you separate tasks such as business logic from the user interface or from integration? How do you make sure the user interface doesn't extend too far into the business code? This often forces a developer to tackle more issues than he or she needs to, such as addressing security and distribution.
- **The application is hard to maintain and extend** - Applications that are not following the right architecture, design and engineering are extremely difficult to update, extend, and modify for basic bug fixes and modifications, as well as for more robust overhauls, to meet changing business requirements.

The bottom line is that the architectural and engineering issues behind developing and maintaining applications are significant. The complexities and risks grow manifold with additional challenges brought by Cloud, SaaS, Multi-tenancy and Configurability. Without a proper platform in place, developers are left with undocumented procedures or a verbal design philosophy to guide their development. Therefore, CMAP plays a strategic role that can drastically improve success rates of products / projects and as well improve the productivity of developers.

### 3. Development Strategy – 3GL Versus 4GL

With the rapid advancement in programming languages, we have 2 groups of languages that are prevalent today – 3GL and 4 GL. Let's understand these in greater detail before we discuss about the development approaches.

#### 3.1 3 GL

3 GL refers to Third Generation Languages, which primarily follows a structure based programming. Some of the popular 3GLs are C, C++, C#, Java, PHP, etc. While 3GL was far



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

better than its predecessor, it still required the developers to be skilled enough to apply the language in the right way.

### **3.1.1 Advantages of 3 GL**

- Provides machine/infrastructure independency
- Improved productivity (compared to 2GL)
- Provides high level of flexibility for the programmers to create solutions as per their priority

### **3.1.2 Disadvantages of 3 GL**

- Requires more coding, which in turn creates more bugs.
- Time consuming – as the development process flows through architecture, design, coding and testing.
- Requires technical expertise. Cannot be used by business users.
- Individual expertise driven. Brings in inconsistency among the developers and resulting in different ways of architecture and development. Quality of the final application is purely individual experience driven.

## **3.2 4GL**

4GL languages are non-procedural; they concentrate on what you want to do rather than how to do it. Gartner defines a citizen developer as a user who creates new business applications for consumption by others using development and runtime environments sanctioned by corporate IT. In the past, end-user application development has typically been limited to single-user or workgroup solutions built with tools like Microsoft Excel and Access. However, today, end users can build departmental, enterprise and even public applications using shared services, fourth-generation language (4GL)-style development platforms and cloud computing services. This shift enables organizations to unlock the tacit knowledge of end users and releases IT resources to do what each does best; however, it requires a new style of IT support — helping end users be good citizen developers.



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

There are a wide variety of tools that offer varying support/scope of functionality. However, for this paper, we will discuss only about the tools that help you develop full-blown applications.

### **3.2.1 Advantages of 4GL**

- Considerably reduces programming effort
- Supports end to end development – right from designing to deployment – all without the need for any technical expertise.
- Allows business users to build applications on their own without support from technical team

### **3.2.2 Disadvantage of 4GL**

- Comes with predefined boundaries
- Not easy to customize or extend. Not suitable for applications with moderate and high complexity.
- Vendor lock-in and Platform Lock-in.

## **4. Development Approaches**

Following are the most commonly adopted development approaches,

- Ground-up Development
- Solution Accelerator
  - Framework
  - Code Generator
  - Components
- Point and Click Development (4GL)
  - On-premise model
  - Cloud model





**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

## **4.1 Traditional Development using languages such as C#, Java and PHP**

In this approach the IT team will develop everything using standard languages such as C#.NET, VB.NET, Java, PHP, etc.

### **Advantages**

- Provides complete control to the architects/programmers on the technology
- No enforced limitations as they are free to choose the technology
- No force fit. Since there are no external software components used.

### **4.1.1 Disadvantages**

- Extremely time consuming as everything has to be built from scratch.
- Reinventing the same wheel. Will not be able to leverage the best practices that already exist in the market.
- Non-availability or limited availability of architect level technical expertise to make the right decisions. Apart from huge cost overrun, architecture level and engineering level decisions can jeopardize the whole project.

## **4.2 Solution Accelerators**

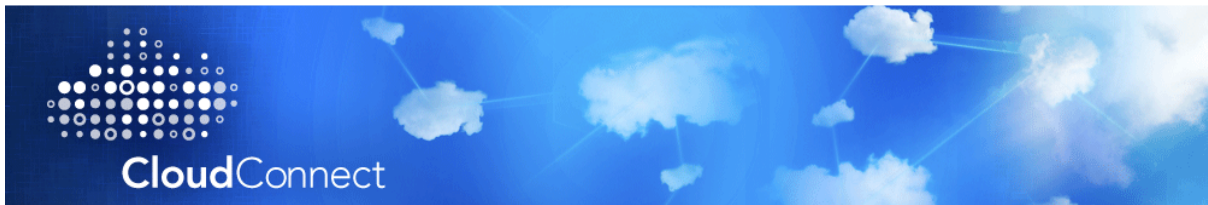
Solution accelerator as the name suggests helps in accelerating the development of applications through one or many of the following ways,

### **4.2.1 Frameworks**

Frameworks can address plumbing layers like security, authentication, rules, workflow, etc. This will significantly help in cutting down the time for developing new applications, as the IT team has to focus only on the functionalities rather than on the plumbing layer.

However, one needs to pay attention to the value that the framework brings to the applications. There are several frameworks in the market but many of them provide very basic elements, which may not be of major value for the applications.

Following are some of the Frameworks that are available in the market,



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

Name	URL
<b>Oracle Application Development Framework (ADF)</b>	<a href="http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html">http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html</a>
<b>Spring.net</b>	<a href="http://www.springframework.net/">http://www.springframework.net/</a>
<b>Ruby on Rails</b>	<a href="http://rubyonrails.org/">http://rubyonrails.org/</a>
<b>MS ASP.NET MVC</b>	<a href="http://www.asp.net/mvc">http://www.asp.net/mvc</a>
<b>CSLA.net</b>	<a href="http://www.lhotka.net/cslanet/">http://www.lhotka.net/cslanet/</a>
<b>Dot Net Nuke (DNN)</b>	<a href="http://www.dotnetnuke.com/">http://www.dotnetnuke.com/</a>
<b>AppFlower</b>	<a href="http://www.appflower.com/cms/home">http://www.appflower.com/cms/home</a>

*Note: Each framework may have certain goals/strengths that are unique to them. Therefore, it's very important for to list down the goals that are important/priority.*

#### 4.2.2 Code Generators

Code generators are tools that can help in generating code for a prefixed scenario. For example, if you have master tables and you want functionality to be developed that includes web pages for Creating, Viewing, Updating and Deleting records on the master tables. This could be achieved by using a code generator that can take up the necessary information and start producing code that will contain the desired functionality. Once again, scope of the code generators varies from tool to tool.

Following are some of the code generator tools available in the market,

Name	URL
<b>Iron Speed</b>	<a href="http://www.ironspeed.com/">http://www.ironspeed.com/</a>
<b>Code on Time</b>	<a href="http://codeontime.com/default.aspx">http://codeontime.com/default.aspx</a>
<b>Code Charge Studio</b>	<a href="http://www.yessoftware.com">http://www.yessoftware.com</a>
<b>Altova</b>	<a href="http://www.altova.com/solutions/code-generation-tools.html">http://www.altova.com/solutions/code-generation-tools.html</a>
<b>Code Smith</b>	<a href="http://www.codesmithtools.com/">http://www.codesmithtools.com/</a>
<b>Rad Software</b>	<a href="http://www.radsoftware.com.au/codegenerator/">http://www.radsoftware.com.au/codegenerator/</a>
<b>Code Generator Pro</b>	<a href="http://www.codegeneratorpro.com/">http://www.codegeneratorpro.com/</a>

#### 4.2.3 Components

Some of the vendors provide specific components that can help perform a particular feature/functionality. Unlike the framework, components are much lighter in nature due to the scope of functionality they address. Components are available in all architecture





**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

layers, starting from UI components (Ex:Grid) to middle-tier components (Ex: Rules Engine) to DB components (Ex:custom scheduling)

Following are some of the leading providers of components,

Name	URL
Telerik	<a href="http://www.telerik.com/">http://www.telerik.com/</a>
DevExpress	<a href="http://www.devexpress.com/">http://www.devexpress.com/</a>
Infragistics	<a href="http://www.infragistics.com/">http://www.infragistics.com/</a>
Component One	<a href="http://www.componentone.com/">http://www.componentone.com/</a>
Janus Software	<a href="http://www.janusys.com/controls/">http://www.janusys.com/controls/</a>
Nevron	<a href="http://nevron.com/">http://nevron.com/</a>

### 4.3 Point and Click Development (4GL)

With the increasing demand from business users for building more and more application, the IT teams of enterprises have a challenging time. 4GL helps resolve this challenge to some extent by allowing the business users (also referred as citizen users) to build and deploy applications by themselves. This is a big boon for the business user community as they now have the power and independence to conceptualize, design, develop and deploy applications that exactly meet their requirement.

#### 4.3.1 On-premise Model

This category of 4GLs are available as on-premise solutions that can be used to build client/server, intranet, internet (cloud) and SaaS applications. Below are examples of the same,

Name	URL
MS LightSwitch	<a href="http://www.microsoft.com/visualstudio/en-us/lightswitch/overview/build-quickly">http://www.microsoft.com/visualstudio/en-us/lightswitch/overview/build-quickly</a>
AppPoint	<a href="http://www.appoint.com/">http://www.appoint.com/</a>
Servoy	<a href="http://servoy.com">http://servoy.com</a>
Accelerator Development Solution	<a href="http://www.surroundtech.com/SoftwareSolutions/Accelerator_Development_Solutions.aspx">http://www.surroundtech.com/SoftwareSolutions/Accelerator_Development_Solutions.aspx</a>
IBM Informix	<a href="http://www01.ibm.com/software/data/informix/tools/4gl/">http://www01.ibm.com/software/data/informix/tools/4gl/</a>
Progress	<a href="http://www.progress.com/en/openedge/4gl-development.html">http://www.progress.com/en/openedge/4gl-development.html</a>



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

**Sculptor** <http://www.sculptor.co.uk>

### 4.3.2 Cloud Model

This category of 4GLs are much more sophisticated as they completely operate from the cloud. These are also referred as aPaaS (application Platform as a Service). Below are some of the leading aPaaS solutions,

Name	URL
Force.com	<a href="http://www.force.com/">http://www.force.com/</a>
Longjump	<a href="http://www.longjump.com/">http://www.longjump.com/</a>
Orangescape	<a href="http://www.orangescape.com/">http://www.orangescape.com/</a>
Magic Software uniPaaS	<a href="http://www.magicsoftware.com/en/products/?catID=70">http://www.magicsoftware.com/en/products/?catID=70</a>
Zoho Creator	<a href="http://www.zoho.com/creator/">http://www.zoho.com/creator/</a>
Mendix	<a href="http://www.mendix.com/">http://www.mendix.com/</a>
Bungee Labs	<a href="http://www.bungeelabs.com/">http://www.bungeelabs.com/</a>
Iceberg	<a href="http://www.geticeberg.com/">http://www.geticeberg.com/</a>
Intuit Quick Base	<a href="http://quickbase.intuit.com/">http://quickbase.intuit.com/</a>

### 4.3.3 Disadvantages of Point and Click Development Platforms

- Vendor lock-in to these platforms. If you would like to come out of these platforms for any reason in future, you will have to re-build everything from scratch.
- These platforms follow certain rules for development. And, generally these rules and approaches are inflexible. This becomes a major limitation.
- Quite Expensive

## 5. CMAP : The Third Alternative

As we saw above, every approach has its own advantage and disadvantage. A CMAP that combines the advantages from each of the above approaches and produce a best of breed solution can be of immense benefit. It can give complete freedom, flexibility and control of custom development, without vendor or platform lock-in and still save us from the complexities and cost overheads of ground up development.



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

## **5.1 Advantages of CMAP**

A CMAP can be used as the base for building and managing an application or multiple applications. The applications can then be deployed on any public or private cloud, configured differently for different tenants and offered to customers on a SaaS Model.

A CMAP brings together the engineering expertise, best practices, re-usable components and time saving tools in to a unified platform format that can be used by developers across an entire IT organization.

### **Platform & Architecture**

- Provide a reference architecture for building the application. This architectural framework should be simultaneously Multi-tenant, Cloud enabled and SaaS ready. It has to be a platform that is tested, proven and well supported.
- It should have a robust architecture that allows any type of application to be built. Unlike the 4GL scenario where only simple business applications can be developed, A CMAP can be used to build simple single user applications to complex multi-user multi-tenant applications.
- Maintain an excellent balance between platforms based approach and application level decisions. For example, if an application requires the front end to be developed on any new technology (Example Silverlight / Flex), the CMAP should provide complete control to the programmers to do the same without compromising the architecture.

### **Best Practices in Engineering Construction**

- Enforce the developers to follow good practices while building applications. Use an n-tier architecture and make sure that the developers follow the layered architecture.
- Provide the right templates and plumbing components such as distributed caching, tenant aware logging etc.
- Support for Customizability, Configurability, Scalability and Security, so that a single code base (or single instance) of the application can serve different groups of users with divergent needs.
- Save time and cost on design, engineering and architecting. Provide all the non-functional requirements in a ready to use form (APIs and Services). Developers can focus only on business functionality thereby improving their productivity levels.

### **Quality**

- Should be well tested for performance, security, scalability



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

- Developers use the best practices and components provided by the CMAP. So it completely eliminates the issues arising out of building the engineering and plumbing stack.

### Speed and Consistency

- Provide ready to use engineering and operational components that are likely to be common across applications – such as:
  - Schedulers,
  - Notification engine,
  - Rules and Workflow Engine,
  - Tenant level customization (of business rules, workflows, forms, grids, data models etc.),
  - Security,
  - Access Control and License / Subscription Management,
  - Role-Privilege mapping and Data Scope policy management,
  - Tenant / Admin management, User Management
- Should use a standard 3GL language such as C# or Java and standard technology stack such as .NET or J2EE. There should be no learning curve for IT team on any new language/technology.
- Combine the advantages of Frameworks, Code Generators and Re-usable Components in to a unified 3GL Application Development Platform, without losing the flexibility and control associated with 3GL development.
- When several applications existing within an enterprise or ISV, is migrated to a single reference architecture and engineering stack provided by a CMAP and developers are conversant with the underlying components, it increases the speed. Also, the consistency of development is much higher between different applications.

### Deploy anywhere

- Support for Cloud as well as on-premise:. With the increasing adoption of Private Cloud and Hybrid Cloud technologies we need the flexibility to operate today as well as in the future. Applications built on CMAP should be deployable anywhere from On-premise boxes, to Private Cloud or Public Cloud infrastructure.
- One of the key issues faced by a SaaS application is scalability. The architecture should allow both scale-up and scale-out, vertical and horizontal data partitioning and data connection abstraction, so that the scalability is configurable (Auto or Manual Scaling) during deployment depending on changing needs.

### No Vendor Lock-in, No Platform Lock-in

- An ideal CMAP should not lock the customer in to any proprietary development or run time platform. It should also be well supported with FREE upgrades etc.
- Source Code licensing options may also be explored, so that the customer has complete strategic and technical control over the entire application stack.



**MAY 24 - 25, 2012 | NIMHANS CONVENTION CENTRE, BENGALURU**

## **6. Conclusion**

We saw that High productivity platforms based on 4GL and Visual tools serve the non-technical business users for building simple, long tail applications. They also have disadvantages such as proprietary development environments and platform lock-in.

On the other hand, ground-up development the traditional way, using developers who write code, is not only time consuming but also risky as there is no unified platform, architecture and engineering structure that can be enforced across the organization. The design challenges of Cloud, SaaS and Multi-tenancy, increases this risk multi-fold.

Using a CMAP is a third alternative that gives developers the freedom, flexibility and control of custom development, without the costs and risks associated with ground-up development.

## **About the Author**

The author is the Director of Product Management at Techcello.