



SaaS & Cloud Application Development & Delivery

A Whitepaper by Ekartha, Inc.

by

Gurpreet Singh, Ekartha Inc.

Raj Sethi, Ekartha Inc.

Ekartha, Inc.

63 Cutter Mill Road
Great Neck, N.Y. 11021
Tel.: (516) 773-3533

Ekartha India Pvt. Ltd.

814/B Law College Road
Demech House, 4th Floor
Erandwane, Pune, India
Tel.: +91-20-6601-4103

Email: info@ekartha.com

Web: www.ekartha.com

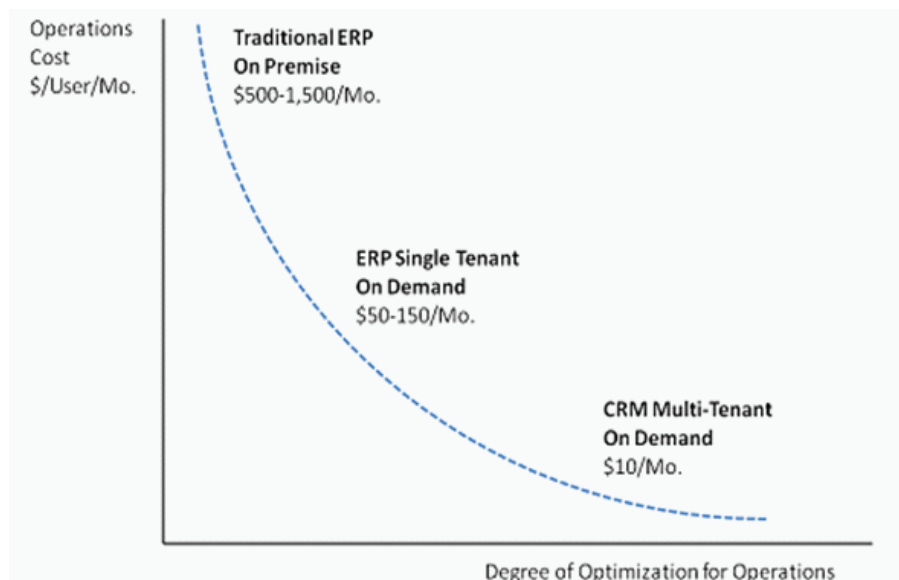
Introduction

The software industry has entered a period of enormous change. Software as a Service (SaaS) and Cloud Computing are fundamentally changing the software business model and the way software is delivered to customers. Compared to on-premise applications, SaaS applications are delivered through the web, billed on a subscription basis, and service providers themselves are responsible for delivering the application at acceptable service levels. As a consequence, the economics of delivering SaaS is different from traditional software applications. **Companies delivering SaaS/Cloud applications need to realize economies of scale, and keep the application delivery costs low.** These issues have a significant impact on how SaaS applications are architected, developed, and delivered.

Over the past 30 years, software vendors developed business applications meant to run on dedicated software and hardware infrastructures at the client's premise or facilities. Furthermore, internal IT departments were responsible for the day to day delivery of the application with software providers supporting bug fixes or the usual upgrades.

The increasing ubiquity of the Internet in the late 90's led many software companies to offer traditional on-premise applications as a service - also known as Application Service providers (ASP). In the ASP model, software companies hosted the software on their own infrastructure rather than at the client site. This business model eventually failed primarily due to the very high cost of managing and delivering traditional on-premise applications as a service. **This led to the realization among some companies that in order to deliver applications as a service, a fundamental change was needed.** Issues like application scalability, cost of delivery, and application availability were paramount, and a new set of architectural, development, and delivery principles were required for a successful business model - a model that is now known as the Software as a Service(SaaS).

The problem of developing scalable and robust SaaS applications with low application delivery costs is difficult. With the success of companies like Salesforce.com, Netsuite, and Oracle CRM the industry has come to some agreement on the critical elements needed to deliver a successful and profitable service. The chart below illustrates an example of how Oracle has dramatically reduced its operations cost compared to that of a traditional ERP On-Premise vendor. **This reduction in operations cost by an order of magnitude is a critical factor in the success of SaaS and Cloud Services.**



Source: Cloud – Seven Clear Business models by Timothy Chu

Achieving a level of application maturity similar to Oracle CRM as displayed in the graph above is a significant challenge for SaaS providers. Unfortunately, companies do not have choice in the matter. ***Over time subscription rates for most SaaS and Cloud services will face downward pressure, and not working towards a mature development and delivery model will make many SaaS business models unsustainable.***

In order to achieve acceptable levels of maturity, companies need to address issues in three core areas:

- They need to build applications that support a multi-tenant architecture that enables a single instance of the application to be shared among multiple customers. Multi-tenancy has a significant impact on all layers of the application stack, and is challenging to achieve. This architectural principle is a significant contributing factor in reducing application delivery costs.
- SaaS vendors need to address a significant number of non-functional application concerns that are essential for the success of the service. For example, traditional software vendors were not concerned with issues like meta-data management, tenant customization and configuration, scalability, fault tolerance to meet SLA's, metering, monitoring, robust security in distributed environments, and a host of other concerns.
- As applications grow and scale, companies need to address automation of operations and application management. Automation of operations and application management is among the primary contributing factors in reducing application delivery costs. Despite emerging automation in areas like the infrastructure cloud, 75-80% of the issues arising in operations are best solved at the application design and development level. Furthermore, it is difficult and expensive to achieve operational and administrative automation once the service is designed and developed. SaaS providers can achieve significant benefits if application architecture takes automation of operations into account early in the application life-cycle.

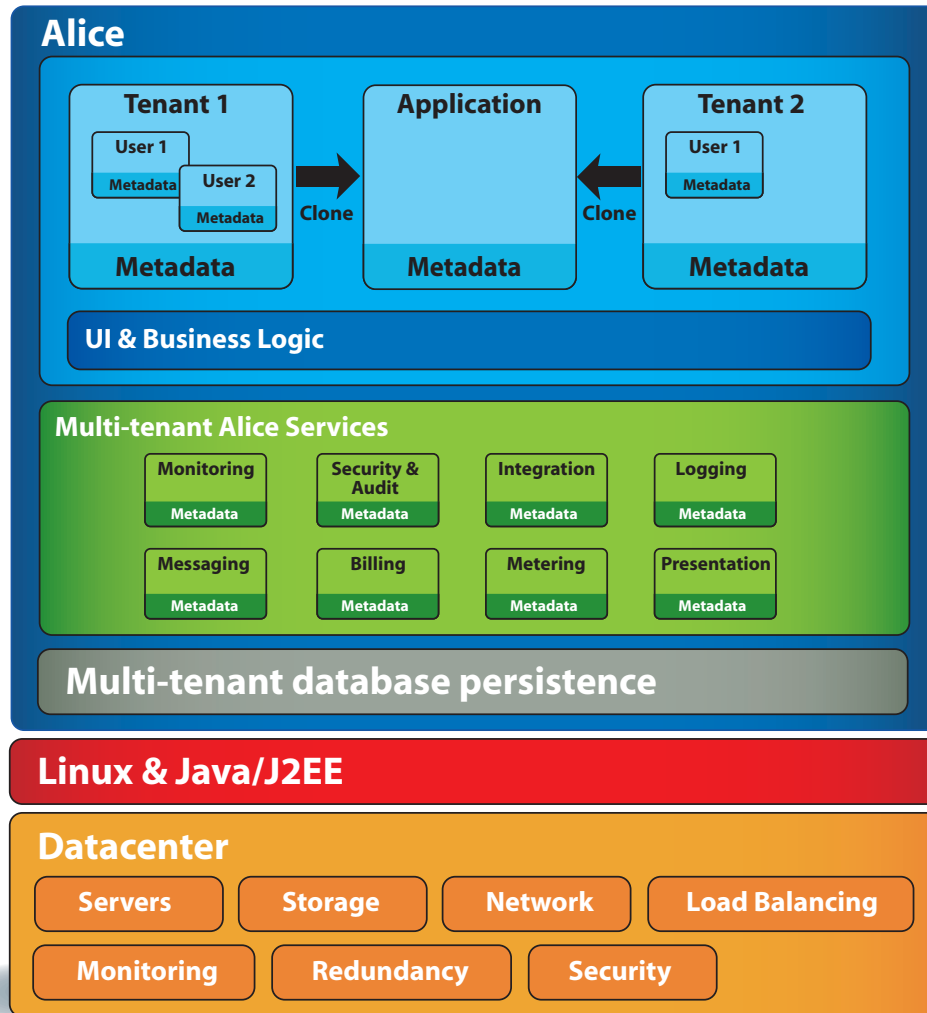
Development of the Alice Platform

In order to help companies with the challenges of building and delivering successful SaaS and Cloud services, we have developed the first open SaaS platform called Alice. The idea of developing Alice arose from the needs of our clients. As a company focused on developing SaaS and Cloud services, it became quite evident to us that traditional J2EE, .NET, Ruby on Rails platforms were not designed to address base level architectural concerns of large and scalable SaaS applications. While building applications for our clients, Ekartha engineers had to spend considerable time extending traditional base platforms and frameworks to address multi-tenancy, data management, security, scalability, caching, and many other features. Many of the most successful SaaS companies had themselves built their own platforms and frameworks to address their specific applications and cost needs. ***Companies' like Salesforce & Netsuite, first and foremost built platforms to meet their application needs and lower delivery costs, rather than building them to be sold as a Platform as a Service(PaaS).***

Release of SaaS application platforms by companies like Salesforce has not made a significant difference in the development and delivery of commercial SaaS applications. Currently, many PaaS/SaaS platforms on the market are suitable for development of only small situational applications, rather than commercial business

applications that are of interest to startups, independent software vendors (ISV's) and enterprises. These platforms use proprietary languages, are tied to a specific hardware/software infrastructures, and do not provide the right abstractions for developers. Alice was developed to address the above concerns, and provide a robust and open platform for the rapid development of scalable SaaS and Cloud applications.

Alice Architectural Diagram



Key Features - Alice Platform

1

Rapid Development and Delivery of complex SaaS and Cloud Applications

SaaS development is complex and applications need functionality such as multi-tenancy, meta-data management, metering, service provisioning etc., in addition to business functionality. Alice hides the technical complexities of SaaS development, and enables developers to focus on business logic. This leads to rapid development and accelerated time to market for SaaS applications.

2

Open Platform using standard technologies

Alice enables the development of applications using open and standard technologies. The open standards insure that you are building applications using technologies that can be developed, supported and maintained by a large community of engineers, architects, and developers worldwide.

3

Robust Multi-tenant and Meta Data Driven Architecture

SaaS applications are built as multi-tenant, meta-data driven applications that are shared across multiple clients. These applications have to be robust, flexible, and highly scalable across a shared infrastructure. With Alice, developers inherit a strong multi-tenant and meta-data foundation that significantly reduces development complexity. It is important to note that a multi-tenant architecture should not just support a single instance application shared among multiple clients. The architecture also needs to support application extensibility, configuration, and customization so that end customers can use SaaS solutions without any compromise.

4

Application Services for SaaS applications

Alice's provides standard application services like Logging, Validation, Presentation, Security, Workflow, Transactions, Messaging, right out of the box. These services can be used by developers during application development, and can be weaved in and out of the application without changing business logic. This significantly reduces development costs and time to market.

5

Integration

The Alice integration service provides a comprehensive set of technologies that makes it easy for companies to build a comprehensive integration infrastructure for their applications. All application's can have a full featured and straightforward Web services API that can provide programmatic access to most of the features and data of the application. Furthermore, all integration connectors built using Alice support full multi-tenancy including the associated security needs and exception handling mechanisms.

6

Operational Services for SaaS and Cloud applications

The operational overhead for SaaS and Cloud application delivery is significant. Issues like monetization, billing, monitoring, and application provisioning are complex issues and companies face enormous hurdles in cost and time in trying to address these challenges. Alice's Operational Services can be used by application developers to incorporate billing, pricing, monitoring, and provisioning services without writing any code.

7

Seamless Deployment of applications on the Cloud

Using Alice, your applications can be seamlessly deployed on any cloud or hosting infrastructure. This provides you complete independence to deploy your applications at an infrastructure of your choice, and not be tied to specific vendors.

Use Cases for Alice

SaaS Application for the Mortgage Industry

Alice, Ekartha's SaaS platform was used to rapidly develop a loan search service for the mortgage industry. The application had a scalable multi-tenanted architecture that allowed large and small customers to share the software and hardware infrastructure. A rule based search engine was built to process millions of rules in seconds. This was essential since fast search was an important qualitative requirement of the service. A robust security module that supported authorization, role based authentication, and encryption was developed in order to deal with sensitive data like credit reports. Tenants of different sizes were allowed to customize the application based on their specific subscription plan.

SaaS Application for Alternative Investment Industry

Alice was used for the development of an alternative investment management (Hedge Fund) Platform. Our client's goal was to develop a professional web platform that allows Investors, Investment Managers, and various Service Providers to securely and reliably exchange relevant information needed by all parties to make well informed investment decisions. Engineers used Alice for design and development of a multi-tenanted database for semi-structured and non-structured (documents, spreadsheets, presentations, images etc.) data. The system was designed to allow for high quality searching capabilities, traditional relational queries, information auditing, and complex event management. A robust security system is being designed to allow investors, and managers to share information in a highly granular manner. A highly interactive browser based user interface was also developed. The software is being currently deployed on Amazon EC2.

Nexus – SaaS based Billing & Subscription Management and System

Nexus is SaaS solution for billing, pricing and subscription management and for SaaS and Cloud companies. It is built on top of Alice and is a fully multi-tenanted solution. Companies using Nexus can easily manage simple or complex billing subscriptions along with comprehensive revenue management. Nexus supports a variety of pricing plans – whether they are based on simple models or complex multi-transaction pricing. Nexus offers advanced functionality such as: revenue management, partner revenue sharing, invoicing and bill presentment, revenue recognition, integration with multiple payment gateways, and automated payment processing. The service can be quickly and easily integrated with existing applications to deliver a comprehensive pricing, billing and revenue management solutions.

About Ekartha

Ekartha provides Rapid Design, Development and Delivery of SaaS and Internet applications. Ekartha's solutions provide companies significant cost reductions, and allow them to focus on their core business growth, rather than development and operations activities associated with SaaS and Internet applications.