

# Developing Applications using Platform-as-a-Service

---

A Paradigm Shift in Application Development

## Contents

Abstract .....	3
Traditional Application Development .....	3
Client/Server Architecture.....	5
Service Oriented Architecture (SOA).....	5
Web Application development.....	6
Advantages of Web Based Applications.....	7
Why Platform-as-a-Service? .....	8
Difference between traditional application development and PaaS application development.....	8
Platform-as-a-Service Benefits .....	9
WOLF Platform-as-a-Service.....	9
WOLF Architectural Goals .....	10
Flexibility of WOLF Offering.....	10
What can you develop using WOLF?.....	11
Comparison of application development on WOLF and traditional development methods.....	11
Conclusion .....	14
References .....	14

## Abstract

Cloud computing is an emerging computing paradigm aimed at running services over the internet to provide scalability and flexibility. The advantages in using the cloud for start-up and small businesses that lack infrastructure have been shown to far outweigh the disadvantages. Cloud platform services, also known as Platform as a Service (PaaS), provide a computing platform or solution stack on which software can be developed for later deployment in a cloud. This whitepaper talks about developing applications using Platform-as-a-Service (PaaS) and how it is different from the traditional application development models. It proceeds to discuss the benefits of PaaS and how flexible WOLF Platform-as-a-Service is to build applications by giving a insight to its architecture. Finally we discuss about the various types of applications that can be developed using WOLF PaaS and give a detail comparison of WOLF application development to traditional methods.

## Traditional Application Development

Software has been part of modern society for more than 50 years. Software development started off as a messy activity often mentioned as “code and fix”. The software was written without much of a plan, and the design of the system was determined from many short term decisions. This worked well for small systems but as systems grew it became more difficult to add new features and bugs were harder to fix. This style of development was used for many years until an alternative was introduced: Methodology. Methodologies imposed a disciplined process upon software development with the aim of making software development more predictable and more efficient. Systems development life cycle (SDLC) provides overall framework for managing system development process. The popular software development models are:

- A. System Development Life Cycle (SDLC) Model
- B. Prototyping Model
- C. Rapid Application Development Model
- D. Component Assembly Model

### A. System Development Life Cycle (SDLC) Model

Traditional methodologies are plan driven in which work begins with the elicitation and documentation of a

complete set of requirements, followed by architectural and high level design development and inspection. The Traditional Systems Development Life Cycle (SDLC) process consists of the following steps:



Figure 1: Software Development Life Cycle

**Planning:** Develop a Project Management Plan and other planning documents. Provide the basis for acquiring the resources to achieve a solution.

**Requirement Analysis:** Analyze user needs and develop user requirements. Create a detailed functional requirement document.

**Design:** Transform detailed requirement document to a detailed system design document. Focus on how to deliver the required functionality.

**Develop:** Convert a design into a complete information system. This includes acquiring and installing systems environment; creating and testing databases, preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

**Test:** Demonstrate the development system and conform the requirements as mentioned in the Functional Requirement document. Conduct quality assurance and test analysis reports.

**Implement:** Implementation of the system into a production environment and resolution of problems identified in the testing phase.

Maintain: Operate and maintain information systems in a production environment and enhance to continue to provide intended benefits.

### B. Prototyping Model

This is a cyclic version of the linear model. In this model, once the requirement analysis is done and the design for a prototype is made, the development process gets started. Once the prototype is created, it is given to the customer for evaluation. The customer tests the package and gives his/her feed back to the developer who refines the product according to the customer's exact expectation. After a finite number of iterations, the final software package is given to the customer. In this methodology, the software is evolved as a result of periodic shuttling of information between the customer and developer. This is the most popular development model in the contemporary IT industry. Most of the successful software products have been developed using this model - as it is very difficult to comprehend all the requirements of a customer in one shot. There are many variations of this model skewed with respect to the project management styles of the companies. New versions of a software product evolve as a result of prototyping.

### C. Rapid Application Development (RAD) Model

The RAD model is a linear sequential software development process that emphasizes an extremely short development cycle. The RAD model is a "high speed" adaptation of the linear sequential model in which rapid development is achieved by using a component-based construction approach. Used primarily for information systems applications, the RAD approach encompasses the following phases:

1. Business modeling: The information flow among business functions is modeled in a way that answers the following questions:
  - a. What information drives the business process?
  - b. What information is generated?
  - c. Who generates it?
  - d. Where does the information go?
  - e. Who processes it?

2. Data modeling: The information flow defined as part of the business modeling phase is refined into a set of data objects that are needed to support the business. The characteristic (called attributes) of each object is identified and the relationships between these objects are defined.
3. Process modeling: The data objects defined in the data-modeling phase are transformed to achieve the information flow necessary to implement a business function. Processing the descriptions is created for adding, modifying, deleting, or retrieving a data object.
4. Application generation: The RAD model assumes the use of the RAD tools like VB, VC++, Delphi etc. rather than creating software using conventional third generation programming languages. The RAD model works to reuse existing program components (when possible) or create reusable components (when necessary). In all cases, automated tools are used to facilitate construction of the software.
5. Testing and turnover: Since the RAD process emphasizes reuse, many of the program components have already been tested. This minimizes the testing and development time.

### D. Component Assembly Model

Object technologies provide the technical framework for a component-based process model for software engineering. The object oriented paradigm emphasizes the creation of classes that encapsulate both data and the algorithm that are used to manipulate the data. If properly designed and implemented, object oriented classes are reusable across different applications and computer based system architectures. Component Assembly Model leads to software reusability. The integration/assembly of the already existing software components accelerates the development process. Nowadays many component libraries are available on the Internet. If the right components are chosen, the integration aspect is made much simpler.

All these different software development models have their own advantages and disadvantages. Nevertheless, in the contemporary commercial software development world, the fusion of all these methodologies is incorporated. Timing is very crucial in software

development. If a delay happens in the development phase, the market could be taken over by the competitor. Also if a 'bug' filled product is launched in a short period of time (quicker than the competitors), it may affect the reputation of the company. So, there should be a tradeoff between the development time and the quality of the product. Customers don't expect a bug free product but they expect a user-friendly product. That results in Customer delight!

## Client/Server Architecture

Businesses of various sizes have various computer needs. Larger businesses necessarily need to use more computers than smaller businesses do. Large businesses routinely have large computer setups, such as mainframes and networks. A network for a large business commonly has client-server architecture, also known as two-tier architecture.

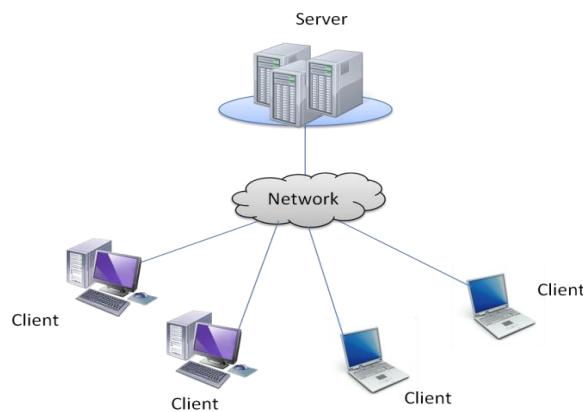


Figure 2: Client-Server Architecture

Under the structure of the client-server architecture, a business's computer network will have a server computer, which functions as the "brains" of the organization, and a group of client computers, which are commonly called workstations. The server part of the client-server architecture will be a large-capacity computer, perhaps even a mainframe, with a large amount of data and functionality stored on it. The client portions of the client-server architecture are smaller computers that employees use to perform their computer-based responsibilities.

Servers commonly contain data files and applications that can be accessed across the network, by workstations or employee computers. An employee who wants to access

company-wide data files, for instance, would use his or her client computer to access the data files on the server. Other employees may use a common-access application by accessing the server through their client computers.

This type of server is called an application server. It takes full advantage of the client-server architecture by using the server as a storage device for applications and requiring the clients to log in to the server in order to use those applications. Examples of this kind of application are numerous; among the most popular are word processors, spreadsheets, and graphic design programs. In each case, the use of the applications illustrates the client-server architecture.

The server is not just for storage, however. Many networks have a client-server architecture in which the server acts as a processing power source as well. In this scenario, the client computers are virtually "plugged in" to the server and gain their processing power from it. In this way, a client computer can simulate the greater processing power of a server without having the requisite processor stored within its framework. Here, the client-server architecture describes a virtual sort of power plant.

## Service Oriented Architecture (SOA)

SOA, or Service Oriented Architecture, is a technology that makes it possible to link a variety of resources on a network for use when and as needed. The use of SOA is sometimes more flexible in how resources are connected and utilized simultaneously than in other types of system architecture designs. Service Oriented Architecture can be especially helpful when larger numbers of users are connected with the system, especially if there are remote users that must have access to specific applications residing on the network.

The concept of SOA did not originate with the advent of computer systems. Because the underlying premise of the design has to do with the interaction of independent resources on a demand basis, the basic model has been employed for decades with defining the infrastructure of a business. The model may be applied to the connections existing between departments that function independent of one another, or even autonomous functions that take place within a single department. In all cases, the structure

established makes it possible to obtain data from any number of independent sources when and as needed.

With software applications, SOA provides the means of leveraging the components of various programs in a manner that makes communication between applications simple and easy. The flexibility of the SOA lends itself readily to systems where there is a need to continually upgrade applications or to expand the network in order to meet additional functions as well as users.

While SOA can be quick and provide easy connectivity to a wide range of program components, it is possible to control the flexibility of the architecture. This gives system engineers the ability to make a particular SOA as scalable and quick as desired, or implement any number of security measures that monitor access, the type of access allotted to each user in the network, and what modules within each program can be accessed simultaneously. In short, the Service Oriented Architecture can be configured to the exact specifications of the Information Technology team and function exactly the way the business needs it to function.

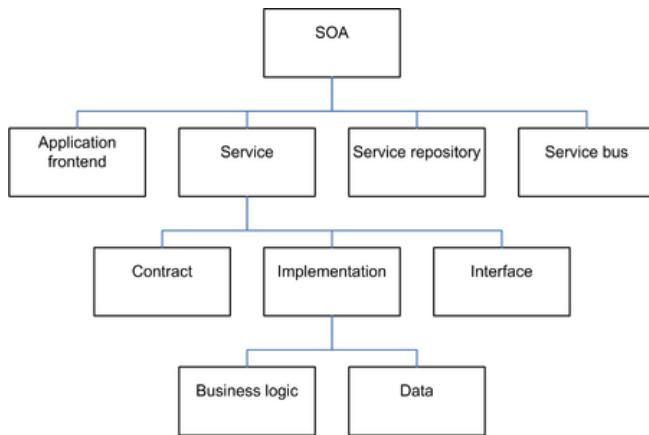


Figure 3: Elements of SOA

Implementing SOA does not necessarily mean that the entire existing network and resident program modules must be changed. There are a number of SOA tools that will easily graft into pre-existing networks without changing one element of the current functionality. What the Service Oriented Architecture will do is allow programs that previously were not able to draw on resources related to other programs to combine functions and make certain tasks easier to accomplish. For instance, a sales and

marketing database may be able to interact with word processing programs to create a mass mailing of promotional letters by combining information contained in a text template, the profiles established in the marketing database, and any independent address books residing on the network.

## Web Application development

Web application development is the process and practice of developing web applications.

The Web application development model, like many software development models, is constructed upon 3 tiers: User Services, Business Services and Data Services. This model breaks an application into a network of consumers and suppliers of services.

The User Service tier creates a visual gateway for the consumer to interact with the application. This can range from basic HTML and DHTML to complex COM components and Java applets.

The user services then grab business logic and procedures from the Business Services. This tier can range from Web scripting in ASP/PHP/JSP to server side programming such as TCL, CORBA and PERL, that allows the user to perform complex actions through a Web interface.

The final tier is the Data Service layer. Data services store, retrieve and update information at a high level. Databases, file systems, and writeable media are all examples of Data storage and retrieval devices. For Web applications, however, databases are most practical. Databases allow developers to store, retrieve, add to, and update categorical information in a systematic and organized fashion.

There are many entities that require applications for the Web-one example would be Business-to-Business interaction. Many companies in the world, today, demand to do business with each other over secure and private networks. This process is becoming increasingly popular with a lot of overseas companies who outsource projects to each other. From the simple process of transferring funds into a bank account, to deploying a large scale Web services network that updates pricing information globally,

the adoption of a Web applications infrastructure is vital for many businesses.

## Advantages of Web Based Applications

**No Installation Required:** Browser based software never requires installation processes or hard drive space. It lives in a virtual cloud in the Internet and this means that whenever you launch it, it always has the latest version. Ajax has made it possible to deliver Desktop-like look & feel, and functionality, with no loss of performance! Web applications do not require any complex "roll out" procedure to deploy in large organizations. A compatible web browser is all that is needed

**Seamless Updates:** Instead of having to patch each and every individual user, the patches/upgrades are applied to the server and each user received the updated version the next time they log in.

**No legacy:** This is a big issue for traditional software vendors. Users who purchase previous versions of a software almost always will result in legacy versions lying around which need support (which is costly). With web applications there is an end to legacy versions.

**No admin rights required:** Finally, a world where the network administrator in the company does not have to approve the installation of your software!

**Available anywhere, anytime:** Web applications can be accessed from anywhere and anytime from a web browser. Available 24 hours a day, 7 days a week.

**Platform independent:** This opens a wider market for software vendors & no longer do they have to build technology around a specific platform and limit their market (or incur additional costs to build for another platform). The browser is the platform here which makes it very simple to run web applications.

**Can be used from any PC:** Web based applications provide cross-platform compatibility in most cases (i.e., Windows, Mac, Linux, etc.) because they operate within a web browser window. No special configuration or changes are need on user's PCs.

**Collaboration:** There are many web based applications that provide chat facilities and the ability to share in real time. This removes the previous stand-alone functionality that used to exist with most installed desktop applications. The world is becoming more and more social – people want to collaborate and work online together – Web Apps allows this, painlessly.

**Lower cost of sale:** No boxes, printed manual, expensive shipping costs, CDs, distribution channels, middlemen, etc. Web apps are more economical to produce and result in a lower cost of sale!

**Piracy-proof:** Here is a big one. Imagine a world without software piracy. That world is here, and Web Applications are the solution to that problem.

**No bad debts:** Software companies often owe money from distributor, which invariably goes for a loss from time to time. With Web Apps, the cash is collected upfront and as long as the customer pays, the account is in good stead.

**Low-cost support and maintenance:** Given that the browser is now the platform, operational support costs and maintenance for Web Application providers will drop substantially. No need to have expensive operating system gurus on hand to help with installation problems. Also, using products like the Amazon EC2 cloud, will allow scalability, without a proportionate increase in costs.

**Data Safety:** User's data is kept safe in hosting environment. Although this is probably not going to be true for all Web App companies, but using providers like iWeb, Rackspace or Amazon's EC2 cloud will go a long way in reassuring your customers that their data is safer than on their desktop! Centralized data is secure and easy to backup.

**No Viruses:** No installation means, no viruses.

**Low cost global distribution:** No more channel reliance. Most software companies make it or break it, depending on their channel. Global distribution becomes much cheaper online.

**Access to the entire assets of the Web (APIs, widgets, messaging, and collaboration):** By being wired into the web, Web Apps are able to integrate seamless into APIs

etc. and are a lot more customizable, than traditional software applications.

**Mobile Support:** Compiled desktop applications are going to have a hard time being adapted for mobile devices. Web apps are ready made (in most cases).

**Widest potential audience:** For all the points above, this basically unlocks markets for software vendors that previously were inaccessible due to technical reasons.

### Why Platform-as-a-Service?

Whenever a business unit within an organization needs to automate a business process or have a centralized database to store information, they will work together with the IT team to develop a web based application.

Tools such as Microsoft Visual Studio can be utilized to develop web applications either using .NET programming language or open source ones such as Java and PHP.

Once an application is fully developed, it will be hosted in house on a production server for users to access. These days, organizations have more options when it comes to developing web applications.

Rather than investing on the following:

- Licenses for application development tools,
- Licenses for operating systems,
- Hardware,
- Network bandwidth;

Organizations can choose to develop their web applications online through Platform as a Service (PaaS). PaaS is quite similar to the concept of Software as a Service (SaaS) but instead of getting a pre-built application, you can develop your very own application via a PaaS provider. For example, let's say that your organization likes what a SaaS application has to offer but all you need is just a few of its features. On top of that, you need additional fields to be included in the system to meet your requirement. It is likely that the SaaS provider might or might not add the additional field for you since it involves changing the database schema which affects other users as well. A change usually takes place only if it benefits all users of the SaaS application.

Due to these SaaS limitations, PaaS providers allows businesses an ability to define and develop everything by themselves as if they are building an application from scratch minus the headaches of hardware maintenance, software licenses maintenance, network monitoring, etc.

There are already many PaaS providers out there in the market today, WOLF Frameworks being one of them.

### Difference between traditional application development and PaaS application development

The traditional model of building and running on-premise applications has always been complex, expensive and risky.

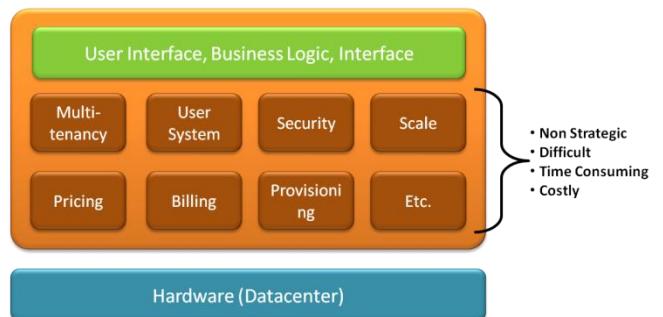


Figure 4: Traditional Application Development

Till recently there was no choice. In the traditional application development model all this had to be built or bought.

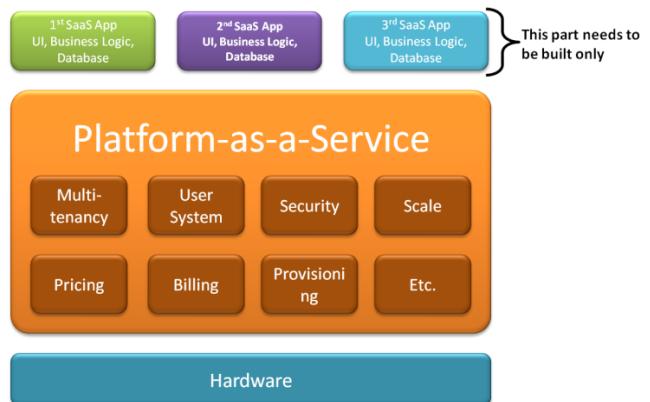


Figure 5: PaaS Application Development

But today with the advent of PaaS and Cloud Computing the scenario has changed. PaaS offers a faster, more cost-effective model for application development and delivery.

Platform-as-a-Service (PaaS) provides an independent platform or middleware on which developers can build and deploy customer application. Common solutions provided in this tier range from APIs and tools to database and business process management system, to security integration, allowing developers to build applications and run them on the infrastructure that cloud vendors own and maintains.

## Platform-as-a-Service Benefits

**For End Customers:** Following are the some benefits of PaaS as a Customer point of view:

1. **Low cost and lower total cost of ownership (TCO):** Generally when customer wants to have a software business solution (IT enabled business solution), there are many factors for which cost have to be considered. The various types of costs involved are datacenter costs, setup costs, licensing costs, training costs, maintenance costs and support costs. When a customer uses PaaS as a business solution, then they don't have to worry about this entire infrastructure and ultimately there is reduction in cost.
2. **Zero Infrastructure - Reduced Overheads:** Customer not has to consider setup and maintenance so there is less overhead associated with PaaS model.
3. **On demand service:** Customer can use a platform of their choice and requirement on a subscription basis and when there is no need for it they can discontinue the same.
4. **Lower risk - faster implementations and outsourced expertise dramatically lower risk**
5. **A more powerful and secure IT infrastructure**
6. **Cost-effective Infinite Scalability:** Customer can Scale the no of user licenses as per their requirements.
7. **Easy to implement:** As there is no infrastructure setup associated with PaaS, it is easy to implement
8. **Freedom of Choice:** As there are many PaaS providers, customer can select according to their choice and requirement.
9. **Centralized feature updating, which obviates the need for downloadable patches and upgrades**
10. **Per license cost has to give**

**For Business Users:** Platform as a Service (PaaS) helps business users to minimize operational costs and increase their productivity. Some of the advantages to business users are:

1. Time to Market
2. Requires no up-front investments
3. Minimize operational costs
4. Centralized information management
5. Enhanced productivity
6. Access to information anywhere, anytime
7. Easy collaboration
8. Secured and customized access

**For Developers:** Platform as a Service (PaaS) enables developers to focus only on innovation that provide real business value instead of infrastructure set-up. Some of the advantages are:

1. Zero Infrastructure
2. Lower Risk
3. Lower cost and improved profitability
4. Easy and quick development
5. Monetize quickly
6. Reusable code and business logics
7. Integration with other web services

## WOLF Platform-as-a-Service

WOLF is a Cloud Computing platform architected to help you design, deliver and use Software as a Service (SaaS) applications using only a web browser. Our aim is to allow professionals and businesses to focus on application innovation, rapid speed to market & continuous enhancement rather than writing routine technical code and managing the infrastructure behind it.

WOLF greatly reduces the barrier to convert an idea or business process into an online web application by offering:

- **Prebuilt SaaS Architecture** that handles multi-tenancy, scalability, provisioning, etc.
- **Code-Free Design Framework** that enables rapid delivery of applications.

Simply define your Entities (data tables/forms), configure Business Rules and drive complex analytics using Charts and Reports. You also have the complete flexibility to white label your solutions and design a customer centric application UX, while using WOLF behind every click.

## WOLF Architectural Goals

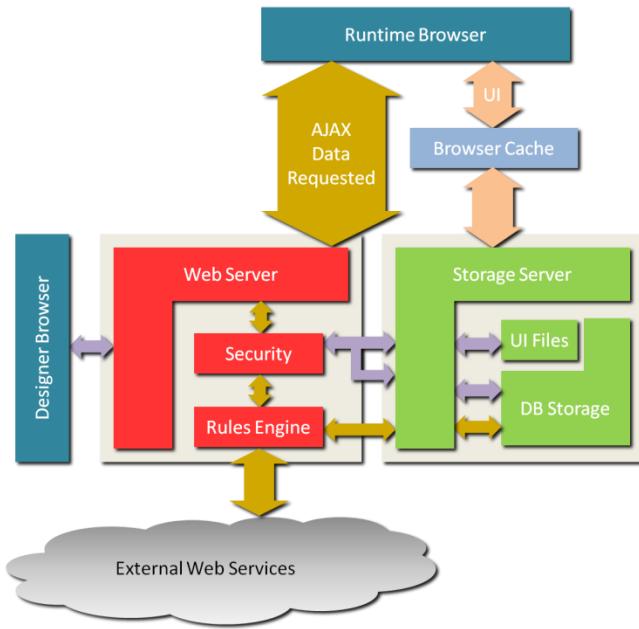


Figure 6: WOLF PaaS Architecture

- Data and UI rendering kept separate (flexibility to use any UI - Silver Light - Adobe Flex, Flash)
- Keep server requests to a minimum
- Data & transaction management using Business Rules actions
- Extend some of the processing to the client machines using the power of Web Browser Scripts (Java Script)
- High performance database (SQL Server and MySQL)
- Minimal Database Dependency
- Ability to extract complete Database and Application Design with a single click
- Ability to interact with multiple internet enabled devices
- Migrate custom applications from single-tenant to WOLF Platform
- Distributed Web Services Architecture

## Flexibility of WOLF Offering

WOLF is a 100% browser based Web Application Designing & Development Platform-as-a-Service (PaaS) for creating mashable and interoperable Software-as-a-Service (SaaS) Business Applications on the internet without writing a single line of technical code. It provides a quick, easy, flexible and cost effective way of building customized web applications, targeted towards Business Users.

All you need is an internet connection, a browser program and the knowledge of modeling your business to use WOLF Service. No software to install, no maintenance cost, no hassles of creating a database, middleware, writing technical scripts, simple point and click user interface with English like business rules to build cross platform SaaS applications. You save more than 70% of your time & cost in building customized applications that can even interact with your existing software system – Complete cloud computing SaaS development framework. The Flexibility of the WOLF Platform-as-a-Service offering are:

1. **Reduced time-to-market and faster ROI**
  - WOLF reduces the time & cost for developing custom web-based software by more than 70% as compared to any standard software development cycle.
  - Clicks instead of code – design all the way!
  - Business Rules engine to define custom workflow
  - Ready chart and report templates
2. **Customized SaaS solutions for every customer**
  - WOLF's unique technical architecture allows customers to develop and enhance custom features and processes within days with minimal investments.
  - You have freedom to white label your solutions – custom branding, personalized URL and custom UI. Easily embed WOLF functional widgets in your website, portal or blog.
3. **Intuitive Business Rules Engine**
  - Ready-to-use Business Rule Actions to fire & map complex business processes. Draw complex analysis from data and represent it in the form of reports, graphical charts and dashboard items within minutes.

#### 4. Interoperable & Mashable SaaS Applications

- Ready actions to interact with on-premise or other Cloud based software systems. No need to deal with APIs— use configurable business rules.

#### 5. Secure

- Built-in User/Role Management module, supports highly secured 128-bit encryption enabled URL, SSO authentication & AD integration.

#### 6. The End of IT Maintenance

- Focus on your business application functionality/core competency while WOLF maintains and upgrades the underlying technology & infrastructure and manages your business solution.

#### 7. Flexible Deployment Options

- Hybrid & flexible deployment models. Host the application in your preferred server: Private Cloud or just run it from the WOLF On Demand Public Cloud or a third party hosting provider

#### 8. Reduced Vendor Lock-in

- Own the Business IP via paper license, extract & save data in your private database & also extract application design in a portable XML format with a single click.

#### 9. Cost Effective, Scalable and pay as you earn

- Monthly subscription fee on per user / per month basis ranging from USD \$ 7 to \$ 10.
- Scale your application cost-effectively with your business.

#### 10. Leverage the WOLF Partner Ecosystem

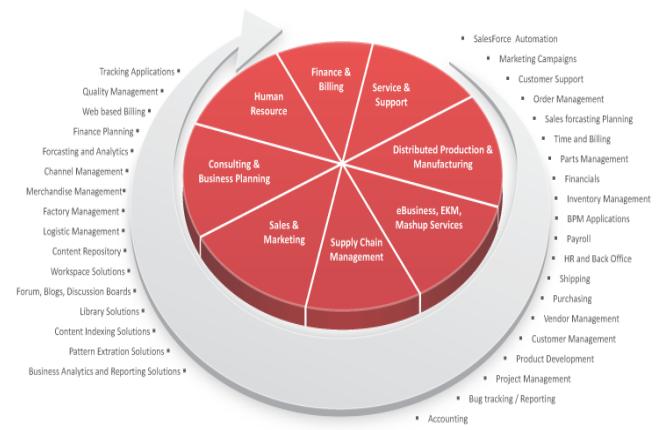
- The Partner Model is targeted to drive idea partners, SI's, ISV's, and Consulting firms to rapidly commoditize and commercialize their business processes and offer them as Software-

### What can you develop using WOLF?

Design and deliver comprehensive business applications for managing and collecting data, streamlining collaborative processes and providing actionable analysis. You can

leverage us to deliver variety of Web based Business Productivity SaaS applications, Utilities and embeddable widgets. Right from mini ERP and CRM to fully fledged HR and Accounting systems, WOLF is ideally suited for delivering complex business applications 70 percent faster at less than half the cost!

From mini ERP to Accounting system, from Fleet Management to School attendance system, from Hospital Patient Mgmt to Career & Library Mgmt System, from Field Force Effectiveness app to Project Mgmt & Broker Trade Mgmt, Compliance/Audit approval system, WOLF is an ideal platform for creating / delivering applications around your data, processes and transactions without managing the complexity of hardware infrastructure, upfront capital fees for software licensing, setting up a database, middleware, templates, application provisioning, authentication, security, etc. Complete OnDemand Cloud Computing framework.



Essential Factors	WOLF	Traditional Application Development	Delivery Time	Inbuilt Modules	200 + hours
<b>Design and Deliver Applications 70 percent faster at less than half the cost!</b>					
Design Time	<b>100 hours</b>	100 hours	Application Provisioning	<input checked="" type="checkbox"/>	
Development Time	<b>200 hours</b>	2000+ hours	Billing	<input checked="" type="checkbox"/>	
In built business logic and workflow engine	<input checked="" type="checkbox"/>		User Administration	<input checked="" type="checkbox"/>	
Ready templates for UI, Forms, Reports, etc.	<input checked="" type="checkbox"/>		Maintenance	<b>Not Needed</b>	15 to 35% of MRP
Custom Branding and easy application styling	<input checked="" type="checkbox"/>		Automatic Software Updates	<input checked="" type="checkbox"/>	
Drive complex analytics via reports and charts	<input checked="" type="checkbox"/>		Freedom from technology overheads	<input checked="" type="checkbox"/>	
Deployment Time	<b>Instant</b>	Minimum 2 weeks	<b>Ready SaaS Architecture</b>		
Scalable Hosted Infrastructure	<input checked="" type="checkbox"/>		Multi-Tenant Application Architecture	<input checked="" type="checkbox"/>	
Secured SSL encrypted URL Access	<input checked="" type="checkbox"/>		Ready Database	<input checked="" type="checkbox"/>	
Testing Time	<b>20 hours</b>	50 hours	Secure and Scalable	<input checked="" type="checkbox"/>	
Technical Testing	<b>Not Required</b>	30 hours	<b>Requires Design Skills only - no coding!</b>		
Functional Testing	<b>20 hours</b>	20 hours	Application Design Discipline	<input checked="" type="checkbox"/>	
			New Programming Language		<input checked="" type="checkbox"/>

No Scripting Language for Business Rules			Data portability across systems		
<b>Ready Actions for Integration</b>					
Less Extra Development Required			Not locked to specific infrastructure		
Work with MS Excel, Project Management, RSS Technology, etc.			No Programming language to learn and work		
Work with multiple databases			<b>Financial Benefits</b>		
<b>Standard Oriented Web Service Technology</b>					
Application Design Extraction (IP)			OpEx Vs. CapEx	<b>Subscription Model</b>	License Model
Replace WOLF UI with own templates			No annual maintenance and upgrade		
SSO Authentication - Role based security			Per user based monthly subscription fee		
Mobile Access			Cost Predictability		
<b>Minimized Lock-In</b>					
Save data in private server			High Investment Security		

## Conclusion

There has been a dramatic shift worldwide in the development tools arena over years. The 80's belonged to client server, '90s to three tiers and now the 00's and 10's to SaaS and PaaS. Each of these seismic shifts are primarily driven by ease of use, lowered cost, higher speed demands/improved networks, global standards, accommodating new user generation, new content & culture with increased efficiency. WOLF Frameworks Platform-as-a-Service has stood out to be one such development tool, designed to enable development of next generation custom SaaS business applications & affordable online services worldwide.

## References

- [1] <http://www.wolfframeworks.com/>
- [2] [http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service)
- [3] <http://jtonedm.com/2010/01/25/first-look-wolf-frameworks-paas/>
- [4] [http://www.newfangled.com/traditional\\_desktop\\_application\\_development](http://www.newfangled.com/traditional_desktop_application_development)
- [5] [http://en.wikipedia.org/wiki/Web\\_application\\_development](http://en.wikipedia.org/wiki/Web_application_development)
- [6] <http://www.saas-attack.com/SaaSCommunity/tbaid/255/aff/1/aft/10/afv/topic/Default.aspx>
- [7] <https://doubleclix.wordpress.com/2010/07/24/822/>
- [8] <http://www.computerweekly.com/Articles/2008/02/29/229632/Traditional-application-development-processes-doom-web-projects-to.htm>
- [9] [http://pds10.egloos.com/pds/200808/13/85/A\\_comparision\\_between\\_Agile\\_and\\_Traditional\\_SW\\_development\\_methodologies.pdf](http://pds10.egloos.com/pds/200808/13/85/A_comparision_between_Agile_and_Traditional_SW_development_methodologies.pdf)
- [10] <http://www.articlesbase.com/software-articles/web-based-applications-vs-traditional-applications-which-platform-1449844.html>
- [11] <http://www.techno-pulse.com/2010/04/cloud-platform-introduction-paas.html>
- [12] <http://www.wareprise.com/2010/05/17/how-to-develop-online-web-applications-using-wolf-paas/>
- [13] [http://java.dzone.com/articles/what-platform-service-paas?utm\\_source=feedburner&utm\\_medium=feed&utm\\_campaign=Feed%3aJavalobby%2ffrontpage%2fJavalobby%2fJava+Zone](http://java.dzone.com/articles/what-platform-service-paas?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3aJavalobby%2ffrontpage%2fJavalobby%2fJava+Zone)
- [14] [http://chucksblog.emc.com/chucks\\_blog/2009/08/private-clouds-vs-aas-models.html](http://chucksblog.emc.com/chucks_blog/2009/08/private-clouds-vs-aas-models.html)
- [15] <http://www.stylesinc.com/Common/Concerns/SoftwareDevPhilosophy.php>

## For More Information

For more information about WOLF products and services, call the WOLF Sales Information Center at +91 98865 46085. To access information using the World Wide Web, go to: <http://www.wolfframeworks.com>