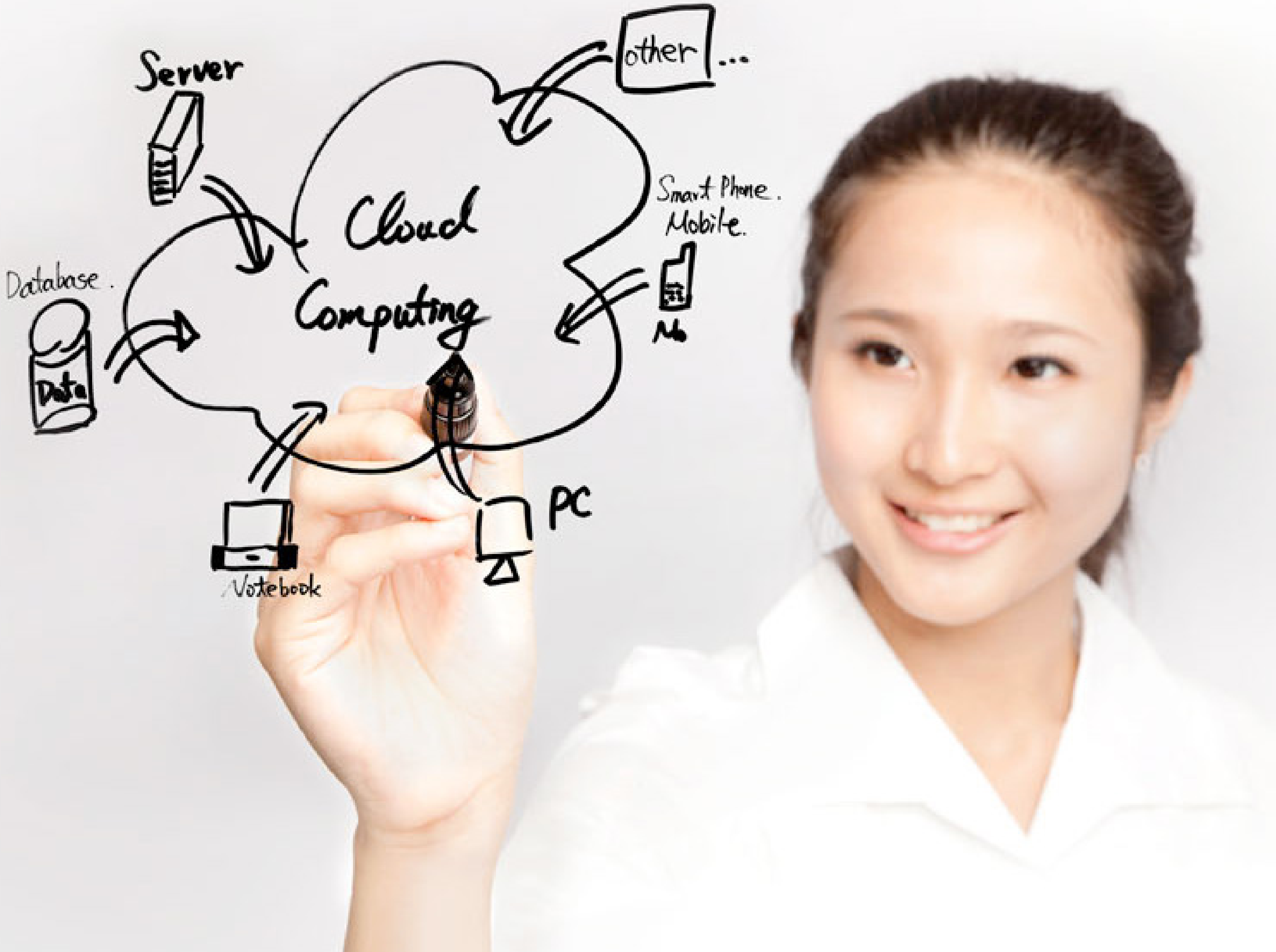
Mindtree

*Welcome to possible*

WHITE PAPER

# Migrating an existing on-premise application to Windows Azure Cloud



# Summary

This discusses how existing on-premise enterprise ASP.Net web application can be moved to Windows Azure Cloud, in PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) model. It is intended for software architects and developers who design and build services for the Cloud.

It covers a case study on moving Mindtree's enterprise application 'Gloria' to Windows Azure Cloud.

# Contents

## Overview – Windows Azure platform

Windows Azure is Microsoft's application platform for the public Cloud. Applications can be deployed on to Azure in various models. Windows Azure is used to:

- Build a web application that runs and stores its data in Microsoft datacenters.
- Store data while the applications that consume this data run on premise (outside the public Cloud).
- Create virtual machines to develop and test, or run SharePoint and other out-of-the-box applications.
- Develop massively scalable applications with many users.
- Offer a wide range of services.

With Windows Azure, the focus is on the development, not the infrastructure. Key benefits of hosting applications in Azure include:

- Minimal focus required on the infrastructure.
- No need to buy / maintain any infrastructure.
- Easy scale-up and scale-out available in Pay-As You-Go model.
- Developer(s) having .Net skill can develop and migrate applications on Azure by learning Azure SDK.
- Windows Azure provides SLA 99.95% for the hosted applications.

Typical implementation models available in Windows Azure are virtual machines, Cloud services, websites and mobile services.

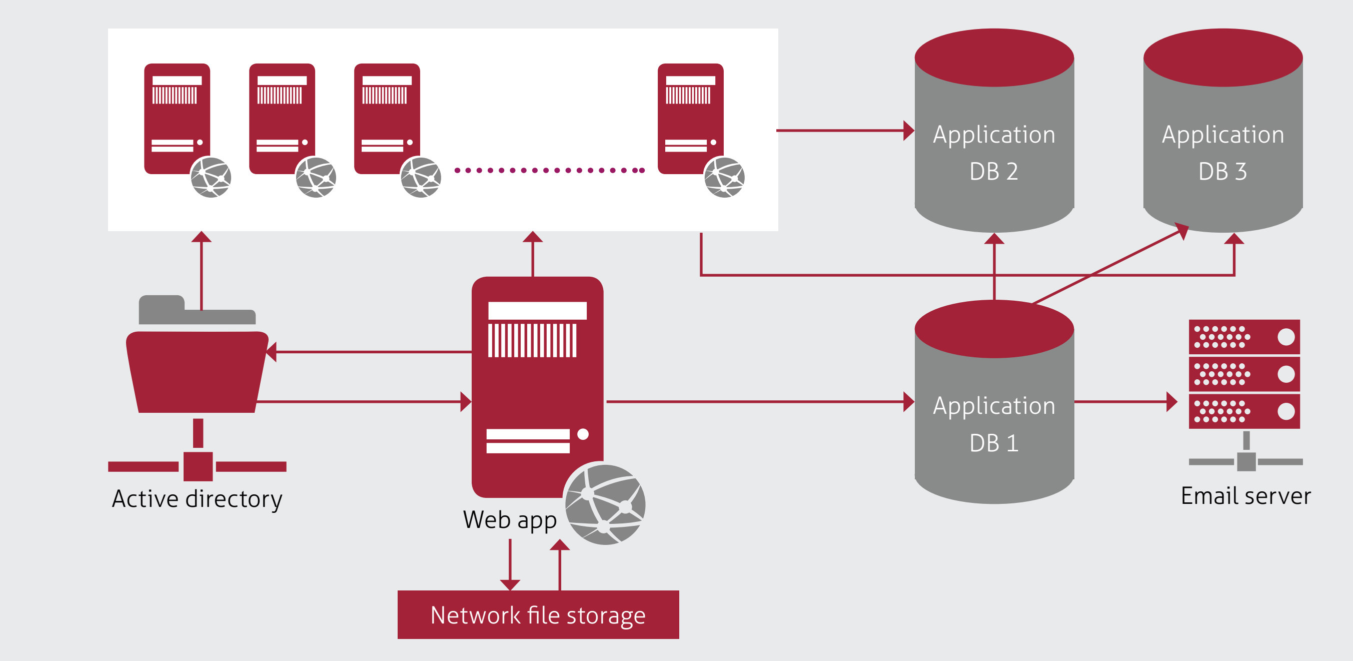
For more details on Windows Azure, visit http://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/

## Case study – migrating Gloria application to Windows Azure

Gloria is Mindtree's on-premise deployed, reward-recognition web-based application. Migration is discussed using this system as an example for both Windows Azure, through Cloud services (PaaS) and virtual machines (IaaS) implementation models. The current architecture and deployment setup of this application:

- Is built with ASP.Net 3.5 and hosted in an on-premise data center. It can be accessed from Mindtree intranet and internet. The application relies on Microsoft Active Directory services to authenticate employees.
- Uses a standard Microsoft SQL Server 2008 database for storing application data.
- Communicates with other Mindtree production databases to fetch employee and other related data using linked server.
- Connects to the production databases using service account.
- Stores the images uploaded to or created from the application using network file storage.
- Is hosted on single instance server.

GLORIA APPLICATION ARCHITECTURE (ON-PREMISE)



## Migration consideration and methodology

Several things should be considered while moving an application to Windows Azure. Major points that need to be addressed are:

- Application compatibility – analyze if the application is architecturally fit for Windows Azure before moving.
- External / internal dependencies – check if there are any external application dependencies and need to analyze and if those are accessible through Azure.
- Application class – verify how the application is classified in the business. Business critical and LOB applications demand high availability.
- Application integration – check whether candidate application is integrated with other on-premise applications and shared services.
- Database compatibility – analyze whether the existing database is best fit to migrate to Azure.
- Application maintenance / management – identify how the logs are maintained and where they are stored.
- Scalability / elasticity – identify whether the application design supports scalability as Azure supports it.
- Compliance requirements – check if there are enterprise compliance and regulations that govern whether the data can be moved / stored outside the enterprise's control.
- Cost - verify whether the moving application is cost-effective for the enterprise.
- Security – clarify whether the same level of security can be provided after migrating to Windows Azure in terms of:
  - Data security
  - Authentication
  - Authorizations

### METHODOLOGY

#### Analyze

- Is it benefitial?
- What can be done?
- Business analysis
- Impact analysis

#### Strategize

- Identifying alternate approaches
- How to implement
- Planning
- POCs

#### Implement

- Migrate
- Validate

## Migration strategy / approach

**UI analysis**

a. Analyze if UI interface can be migrated to Azure directly in the PaaS model, web applications and web services in non-cloud solutions can be mapped to Windows Azure web roles, while the non-web applications are kept on-premise. Re-engineering work is required to modify the existing web application code to use the Windows Azure SDK. This is to ensure it runs on Windows Azure as hosted Cloud services. In addition, if any third party framework / class library that is not supported by Azure is used, the library may need some modifications or, will have to be re-written.

In case of the IaaS model, no code changes are required. Instead, the entire server image will be migrated to the Windows Azure virtual machine.

The sections below focus on the PaaS approach as no major changes are required in the IaaS one.

b. Transactions / session handling

**Session state management:**

In Azure, each web role instance runs on its own VM server and are configured behind a load balancer. The ASP.Net session state is not automatically shared across instance in this load balanced environment. Various approaches to address this are as follows:

1. Session state management using Inproc: Inproc session state may be the best performing option and is the default state management, unless overridden. In load balanced environments such as Windows Azure, this will only function for single instance setup. If more than one instance is used, it might result in inconsistency.

2. Table storage session provider: It is a subset of the Windows Azure ASP.NET providers specifically designed and written for use in Windows Azure. The table storage session provider is a custom provider that is compiled into a class library (.dll file), enabling developers to store session state inside Windows Azure table storage. This

approach is relatively low cost, well tested and ready for consumption, with almost no re-engineering work.

3. Windows Azure SQL database session provider: Windows Azure SQL database is essentially a subset of SQL server. It can also be used as storage for session state. With just a few modifications, it can be derived from SQL server session provider. When used with the existing database, it is cost effective.

4. Windows Azure cache:
4.1) In-role cache allows caching. It uses a portion of its memory for the web or worker role instances that host an application, or with a dedicated one deployed to Windows Azure Cloud services. One does not have to pay any premium for cache.

4.2) Azure cache preview gives access to a secure, dedicated cache that is managed by Microsoft. A cache created using the service is accessible from applications within Windows Azure, running on Azure Web sites, web and worker roles and virtual machines.

## Business layer analysis

Windows Azure caching remains the recommended option, despite the debates. Developers and architects could always consider a different option, if it suits a given scenario.

**a. Authentication and authorization model**

Analyze and understand the current authentication mechanism in the application.

1.Check if the same authentication approach in the cloud application can be adopted (for instance, using forms-based authentication).

2.Explore Cloud based solutions for authentication.

- Utilize Windows Azure Active Directory Access Control (also known as Access Control Service or ACS) to authenticate users from identity providers, when they try to gain access to a web application. These include Microsoft, Google, Yahoo and Facebook.
- Deploy Azure virtual private network to communicate with enterprise active directory.
- Use Windows Azure Active Directory with integrated on-premise organization active directory.

**b. Interaction with other modules / applications**

**Web services:**

- They can be converted to Azure WCF services, hosted either as a web role or worker role.
- They can be left as on-premise services and can be exposed through Azure service bus or Azure VPN.

Windows Services:

- They can be hosted as worker roles.

Native code:

- A managed wrapper can be created and deployed as part of an Azure package. It can be consumed from any Azure hosted service.

Third party / non-Microsoft dependency:

- Need to confirm if they can be consumed directly from Azure application.

**c. Diagnostics support**

Windows Azure diagnostics provides non-intrusive capture of diagnostic data and its sub-sequent data to the Windows Azure storage service.
With WA diagnostics, one can:

- Implement custom logging and save the log information to Windows Azure storage tables.
- Push event logs to diagnostics store.
- Push failed request logs to diagnostic store.
- Push performance counter data to diagnostics store.

**d. Message queues**

Currently, MSMQ is not supported in Windows Azure. Azure storage queue and service bus queue features can be used to implement queues in Azure.

Azure service bus topics / subscriptions can be used for message publish and subscribe model.

**e. Configuration changes**

- If the application accesses any information from physical storage, it needs to be migrated to Azure storage / CDN / Windows Azure SQL database.
- Check for any hard coded physical disk paths in the applications. Azure applications should not have any hardcoded physical disk or network access values. Ex: Application saves any information to local system / access information.
- Check for any third-party library or content references. They need to be included as part of the Azure package.
- Provide session and identify provider (ADFS / ACS information) in the web configuration file.

**f. Miscellaneous**

- Replace static values and application states to handle scalability applications in Azure.

### Database analysis

Three ways to maintain application data while migrating an application to Windows Azure:

1. Reside on premise
2. Create SQL server in Azure VM
3. Windows Azure SQL Database (PaaS)

The section below describes the approaches and recommendations:

a. Analyze the DAL layer and see if any framework / class library is being used. Determine whether it is supported with Windows Azure SQL database.

- Frameworks such as Entity Framework (EF) are supported with Windows Azure SQL Database. In case any third party class library / framework that is not compatible with Windows Azure SQL Database is used, the DAL layer code needs to be re-written. This is to achieve compatibility with the database.
- Windows Azure SQL Database supports only SQL authentication. One should consider whether the changes are needed to the authentication scheme in the application.

b. Determine whether all the data required by the application is in a single database or if it is dependent on:

- If the application database uses linked server concept to interact with another database, it cannot be moved to Windows Azure SQL Database. This is because it does not support the linked server feature.
- The above problem can be solved by using Windows Azure Virtual Network feature. This is where the on-premise application database is exposed to Azure Cloud applications.

c. Determine the modifications required to be done in the database schema for migration to Windows Azure SQL Database.

- Run SQL Migration Wizard to have a complete list of all the modifications required for Azure porting.
- Windows Azure SQL Database demands some constraints on the database schema to be deployed, such as the primary key constraint on all the tables. One should make necessary modifications on the database schema to be ported to Windows Azure SQL Database.

### Data migration strategy

Application migration strategy should also include data migration strategy as most of the applications are typically data centric. Hence, while migrating application from on-premise to Cloud, we need to ensure that the users do not see any discrepancy in their data. Application can store data onto disk, into a database, network stores and more. Cloud provides the flexibility to persist data in the same way they are stored in the on-premise application. Azure hosted applications data can be saved in the following ways.

a. Data from Database to Windows Azure SQL Database / Windows Azure storage

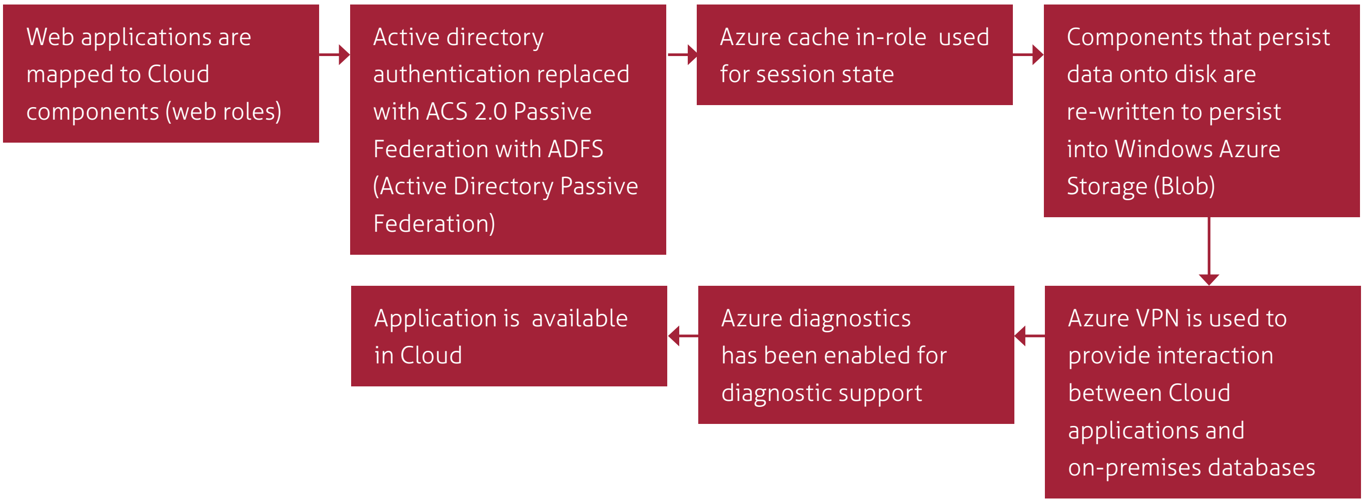
b. Static content to Windows Azure Storage (Blob) / XDrive.

c. Message queues to Windows Azure queue storage / service bus queue.

## Migration execution

In the candidate application migration (Gloria), components are migrated to Cloud in an incremental, independent fashion. At each phase, it was ensured that the application works with no internal (application) / external (network) issues. The section below explains the migration process for the PaaS and IaaS options.

## PaaS MIGRATION STEPS FOR GLORIA



Web applications are mapped to Cloud components (web roles)

Active directory authentication replaced with ACS 2.0 Passive Federation with ADFS (Active Directory Passive Federation)

Azure cache in-role  used for session state

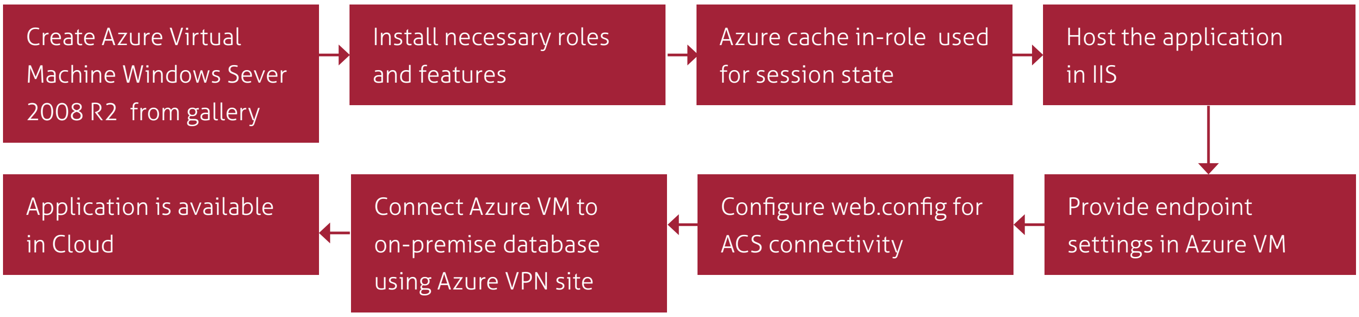
Components that persist data onto disk are re-written to persist into Windows Azure Storage (Blob)

Azure VPN is used to provide interaction between Cloud applications and on-premises databases

Azure diagnostics has been enabled for diagnostic support

Application is  available in Cloud

## IaaS MIGRATION STEPS FOR GLORIA



Create Azure Virtual Machine Windows Sever 2008 R2  from gallery

Install necessary roles and features

Azure cache in-role  used for session state

Host the application in IIS

Provide endpoint settings in Azure VM

Configure web.config for ACS connectivity

Connect Azure VM to on-premise database using Azure VPN site

Application is available in Cloud

## Conclusion

The strategy, consideration and methodology discussed in the case study of Gloria application, provide guidance to enterprises, looking to migrate their on-premise application to the Windows Azure Cloud.

| Acronym | Description |
|---|---|
| ACS | Access Control Service |
| ADFS | Active Directory Federation Service |
| IaaS | Infrastructure as a Service |
| PaaS | Platform as a Service |

**About the author:**

Kiran Kumar SVM is an Architect with Mindtree. He specializes in architecture and designing solutions using Microsoft Technologies with a particular focus on Azure technologies. He has over 12 years of experience in design, development, consulting and delivering world-class software products and solutions. Kiran provided solutions for several applications and services that run on Microsoft Windows Azure Cloud platform. He is passionate about technology and contributes to Microsoft forums.

## About Mindtree

Mindtree is a global information technology solutions company with revenues of over USD 435 million. Our 13,000 experts engineer meaningful technology solutions to help businesses and societies flourish. Mindtree's consulting-driven approach makes us a strategic partner to over 40 Fortune 500 enterprises.