



Ebix OneOffice

Software and Hardware Architecture



Introduction

This document describes the hardware & software architecture and deployment options for Ebix OneOffice.

OneOffice utilises an advanced and open, services based architecture that delivers a fully web-based system with advanced UI and a premium User Experience. As OneOffice uses a fully web-based thin client, no installations or downloads are required on the End User's computer. System architecture supports full Straight-Through Processing, paperless office, process automation and self-service portal.

OneOffice architecture is designed for an easy integration with a variety of third-party applications and technology landscape using a comprehensive series of APIs. For example OneOffice has been interfaced with Underwriting Rules Engines, and comparator systems through public facing APIs.

The solution is platform-agnostic, so is able to operate on all the industry's main environments and can leverage modern infrastructure needs.

Ebix OneOffice offers an industrialized and robust solution with proven ability to manage required business transactions in production environments.

Technical Architecture

Ebix OneOffice is a web browser based application developed on a Services Oriented Architecture (SOA) using a multi-tier architecture using Microsoft.NET framework to distribute the workload across different layers so as to make the application scalable.

The layers of the application are:

- User Interface (Presentation) Layer
- APIs Layer
- Business Services Layer
- Data Access Logic Layer

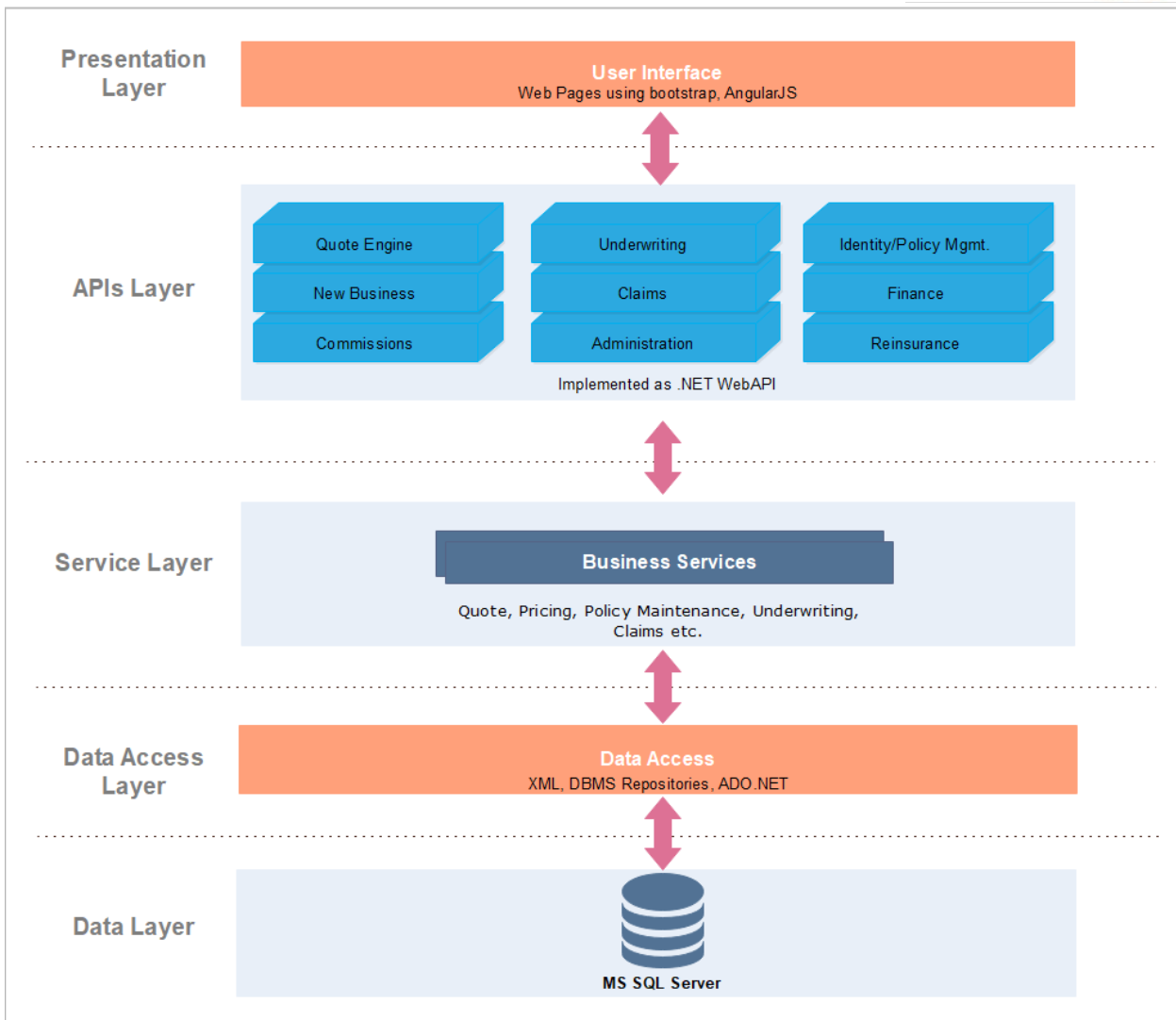


Figure 1: Technical Architecture

Presentation Layer - provides the interface between the user and the API layer. The presentation layer includes all pages that are used to either display or capture information. Data persistence / session management is controlled by ASP.NET at the application and session level and by AngularJS at the page level. When information is requested, the presentation layer captures the user request and calls the correct object in the API Layer. Similarly, when the API Service Layer returns data results, the presentation layer formats the data and displays the results on the page.

The contents of the presentation layer have been developed using Microsoft.Net MVC and the Bootstrap UI framework.

APIs Layer - acts as the interface between the presentation and business layers. It performs all the required security checking and data manipulation based on either the data received from the front-end or business logic



layer. The components making up this layer are implemented as services running on the Microsoft.Net Common Language Runtime (CLR) engine. This layer also acts as the interface between 3rd party applications (such as expert underwriting engines, document generation and storage platforms, commission payment platforms and financial adviser portals etc.) and the OneOffice business logic layer.

Business Logic Layer - all of the functionality in this layer is defined as components describing typical life insurance policy administration functions (premium quotes / projections, application processing, policy issue, additional business, benefit maintenance, underwriting, claims etc.). These components are implemented via Microsoft.Net C# classes running on Microsoft.Net Common Language Runtime engine.

Data Access Logic Layer – retrieves the data requested by the Business layer. This layer is implemented using an SQL Server database. The database is accessed using ADO.NET and Microsoft Entity Framework.

Code Overview

The Ebix OneOffice web UI and business logic layer code is governed by the Microsoft.Net framework utilizing features provided by the Microsoft.Net Common Language Runtime, including:

- Extensible types provided a class libraries
- Inheritance and polymorphism on object-oriented programming principles
- Support for explicit free threading that allows creation of multithreaded, scalable applications
- Support for structured exception handling, Logging and custom attributes
- Automatic Object Lifetime management so that reference counting is unnecessary
- Self-describing objects, which make using Interface Definition Language (IDL) unnecessary
- The ability to compile once and run on any CPU and operating system that supports the .NET runtime

Responsive Design

The Ebix OneOffice user interface has been built using the responsive design principals inherent in the Bootstrap framework. This allows the OneOffice user interface to render correctly in different screen resolutions on different devices (workstations, tablets and smart phones).

Security

Ebix OneOffice utilises role based security using ASP.NET Identity. The ASP.NET Identity system stores all the user information in a database. ASP.NET Identity uses Entity Framework Code First to implement all of its persistence mechanism.

This can be further extended to utilize third-party authentication providers using Owin middleware.

Hosted vs On Premise

Ebix OneOffice can be implemented as either an on premise installation or a hosted solution running on Ebix hosting infrastructure.



Application Components

The following diagram shows the business architecture of Ebix OneOffice which composed of loosely-coupled components. System can be deployed at client network or hosted in Ebix hosting environment. OneOffice application integrates with various third-party systems through different means which include manual file transfer and through real-time API connectivity.

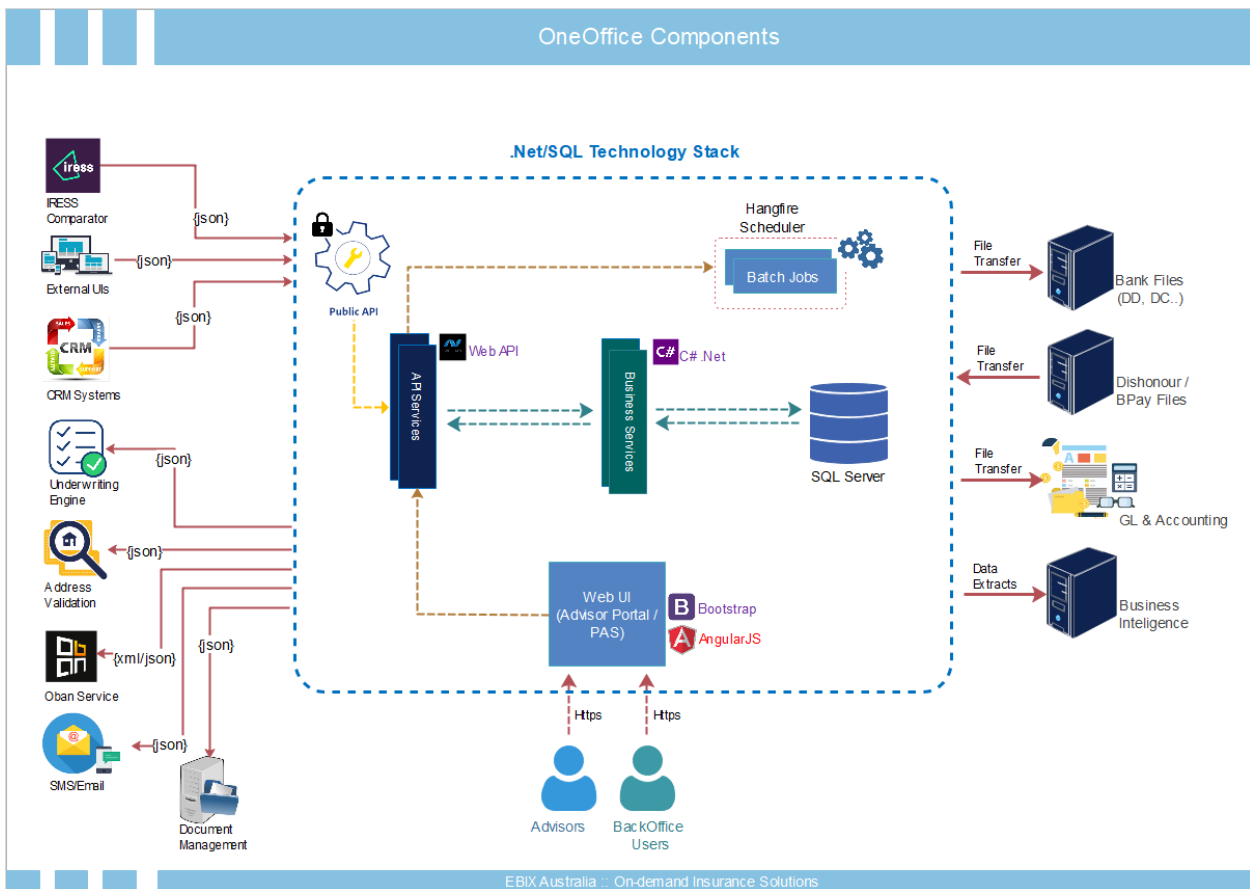


Figure 2: High Level Solution Components

IRESS Comparator: IRESS uses APIs provided by OneOffice for quote calculation for price comparison available for general public. OneOffice APIs are deployed in Ebix infrastructure on dedicated servers to be consumed by IRESS.

External UIs: Any external web/mobile application can access the public facing APIs exposed by OneOffice. As these APIs are based on industry standards, any third-party proprietary application can connect to OneOffice and exchange information.



CRM Systems: OneOffice is currently connected with internal/external CRM systems through public facing APIs. These include Ebix's SmartOffice and Salesforce.

Underwriting Engines: To perform underwriting on insurance quotes, various underwriting engines can be utilised by OneOffice. These include PacRe - UnderwriteMe, MunichRE - Allfinanz & SwissRE - Magnum underwriting engines. The integration with these systems is based on REST/Soap based APIs.

Address Validation: OneOffice connects with MasterSoft's Address validation services to validate the Australian Addresses and perform address lookups.

Superannuation Payments: In order to process the policies funded by superannuation, Oban services are integrated with OneOffice. This includes calling Oban APIs to pass on policy specific information in the required format.

SMS/Email: Customer correspondence and notifications are sent through SMS/Email. OneOffice currently utilises ClickSend as the SMS gateway and SendGrid for sending Emails. Both these systems are invoked through the external system APIs.

Document Management: Insurance policy related documents can be uploaded using OneOffice. These documents are stored in internal SQL database. However, OneOffice has the capability to call third-party document storage systems to pass these documents and store the external references for document viewing from OneOffice.

Payment – Direct Debit: OneOffice generates the bank files (direct debit) which are passed to bank for payment processing. Corresponding dishonour files & BPay files provided by the bank will be processed by OneOffice batch jobs to update the corresponding policies.

Payment - Credit Card: OneOffice can interface with 3rd party credit card tokenisation and processing providers. Currently OneOffice is interfaced with the IPSI solution which provides tokens and will process the direct credit files generated by OneOffice and return dishonour files to be processed similar to the bank Direct Debit files.

General Ledger & Accounting: OneOffice generates files for various GL and accounting systems which include Oracle Financial, TechOne, MYOB and SAP.

Business Intelligence: Data extracts are provided for input to third-party BI and Analytics solutions.



Components & Versioning Strategy

Following diagram depicts various deployable components of OneOffice package. A specific version number is specified with each component (Web-UI, APIs and Database) to uniquely identify these components.

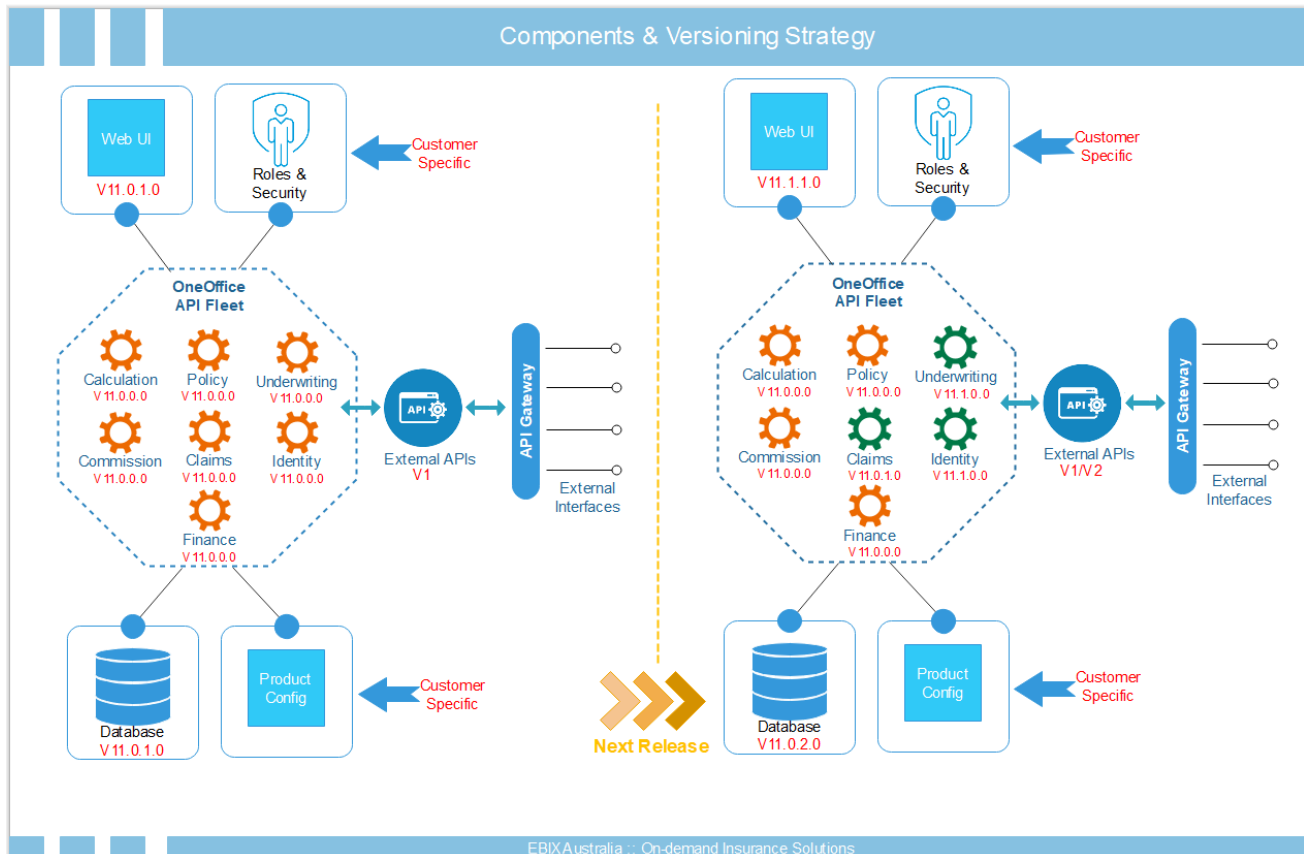


Figure 3: Components & Versioning Scheme

Version Number: The version number is based on following structure:

[ReleaseNo].[MajorVersion].[MinorVersion].[RevisionNo]

ReleaseNo: Reflects the current release of OneOffice (i.e. 11)

MajorVersion: Reflects the Major version of the specific component within the OneOffice package (i.e. 1)

MinorVersion: Reflects the Minor version within the mentioned major version of the specific component in provided package (i.e. 2)

RevisionNo: Reflects the RevisionNo of the specific component to identify any quick fixes performed. This number would reset with every new Release/MajorVersion/MinorVersion produced.



Web UI: The Web-UI is an individual component which would have a specific version number (e.g. 11.1.1.0). This version number will keep incrementing based on the future releases of OneOffice.

API Fleet: The APIs are a core component of all the business processing within OneOffice. These APIs are called by the Web UI layer to perform various business operations. APIs are categorised into logical business processes (Calculation, Policy, Identity, Underwriting, Claims etc.) and each group would have a separate version number associated. Future releases of OneOffice will allocate the new incremental version numbers to these APIs based on the changes.

There may be a situation where only a few of these APIs would undergo changes in any particular release, in such cases the APIs being changed would show a different version number than the rest of the APIs.

Note: As an API may have internal dependencies involved, Ebix will maintain a compatibility list to identify which APIs are compatible with rest of the APIs (and other components) to work together. Release upgradation recommendations will be based on that compatibility list.

Each deployment of OneOffice will contain the complete API fleet no matter which of these are required by that customer. OneOffice will disable the access of non-required APIs through Roles/Security configuration.

Database: OneOffice database will have a version number associated with it to ensure that its kept up to date and compatible with latest API versions.

External APIs: An external layer is added on top of the internal APIs to allow external integrations. In the case where OneOffice APIs are required to be consumed by third-party applications, this layer will be directly invoked. This layer may have multiple versions (to avoid introducing any breaking changes to previous consumers). This layer will be further exposed through an API gateway to control the life-cycle, security and quota management for all consumers.

Role & Security: These are customer specific settings within OneOffice to control the access level for UI and APIs. This will be achieved through user defined configuration.

Product Configuration: Customer specific products and business rules are configured in the OneOffice instance being deployed to fulfil the needs of the specific customer.



Deployment Architecture

Ebix OneOffice has been designed to be deployed either on a single server or in a highly scalable, load-balanced and clustered Windows server environment. Load balancing is utilised to distribute requests to different servers within the server cluster, which optimises system performance and increases availability and scalability of the web-based application.

As the load on OneOffice increases, the service layer and business logic layer services can be distributed across an array of machines that can handle the increased load. A typical OneOffice production deployment runs on a web-farm with two or more application/web servers and database servers in a cluster to provide redundancy.

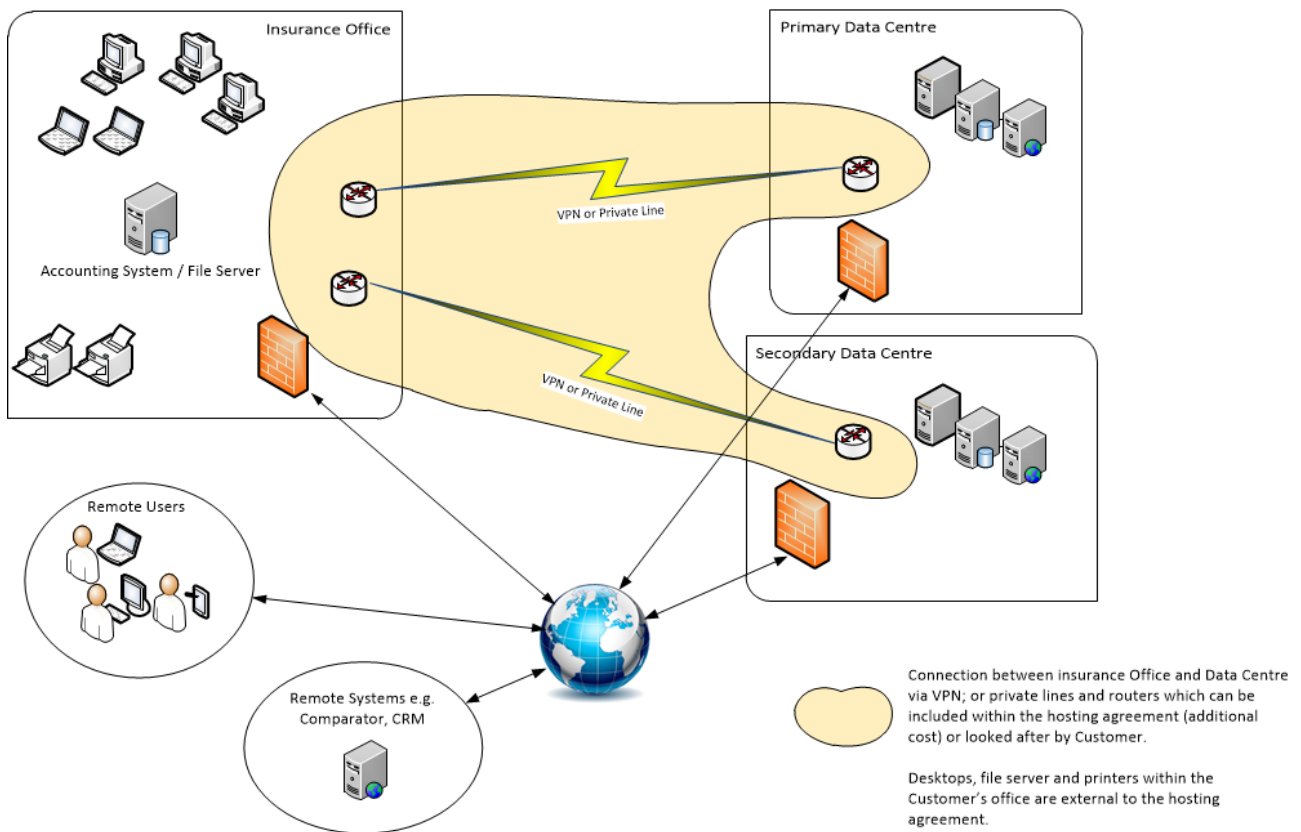


Figure 4: Deployment Architecture

Ebix recommends the following hardware and software for deploying the OneOffice production environment. Other environments such as SIT, UAT can be created on similar lines:

- Hardware Load Balancer
- Firewall and Switches



- Two (2) web servers – Windows 2016 Advanced server, IIS 7.0
- One (1) database server – Windows 2016 Advanced server - SQL Server database

Software and Tools used in developing OneOffice:

- C#.Net for the front-end interfaces, server side components and web services
- ASP.NET for creating pages for the web application
- SQL Server RDBMS 2016, Standard Edition
- Entity framework for accessing the database
- JSON over HTTP for all data transmission