Homeless Individuals and Families Information System

HIFIS 4.0

Technical Architecture and Deployment Reference

HIFIS Development Team

May 16, 2014

Contents

**** THIS IS A PRE-RELEASE DOCUMENT AND SUBJECT TO CHANGE ****

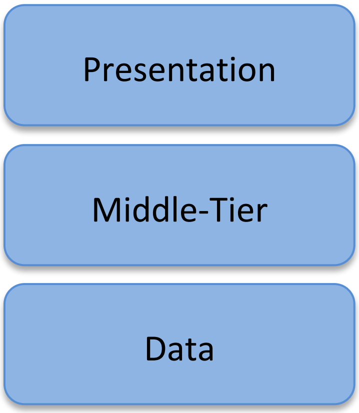
Introduction

Welcome to the HIFIS 4 Technical Architecture and Deployment Reference. This document explains the overall technical system architecture of HIFIS 4 and provides information that will help with planning a successful deployment.

HIFIS 4 System Design

HIFIS 4 is comprised of several distinct components with a design that can accommodate a variety of deployment configurations and implementation scenarios. The core components are representative of those common to a typical multi-tiered enterprise system and consist of a presentation layer, a business middle-tier and an underlying data store.

Figure 1 – HIFIS 4 System Design Layers



PRESENTATION INTERFACE

The Presentation Interface of HIFIS 4 consists of a web application that users see and interact with in a web browser. This allows users to access HIFIS from any web-enabled device with a compatible web browser and network access to the HIFIS web application server.

The web application uses a user interface framework called the Web Experience Toolkit[1] (WET) and a default theme that can be modified or replaced to customize the look and feel of HIFIS.

The web application also includes a web-based Crystal Reports runtime engine to generate and display reports created with Crystal Reports. HIFIS ships with a series of reports included, but with Crystal Reports you can also create new reports or edit existing reports and upload them directly into HIFIS for use directly in HIFIS.

The Presentation layer does not contain any application logic except that which is required for correctly displaying information for the user. It depends on the Middle-Tier component to handle application logic and information.

[1] http://wet-boew.github.io/

MIDDLE-TIER

The HIFIS 4 Middle-Tier is a service-oriented business component. It consists of a series of hosted web services that implement all of the HIFIS business logic and handles authentication. Because HIFIS 4 is entirely service-oriented with web services that fully encapsulate the rules and behaviours of the entire application, it is extremely adaptable to unique and complex operational environments. Also, access to the web services is highly configurable at the hosting level and within the HIFIS application itself to control and monitor precisely what systems are accessing services and what activities are taking place. Since the HIFIS web application is built on these same services you can be assured that the behaviour of the HIFIS web application and a third-party system using the same services will be the same.

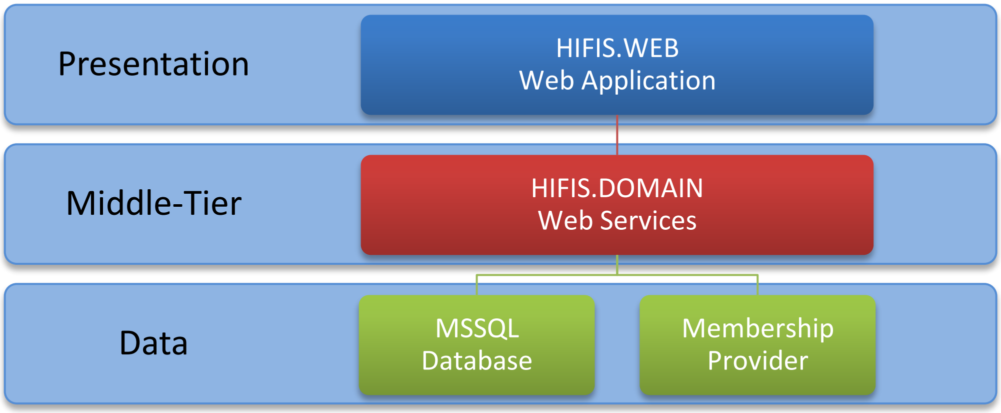
DATABASE

The Database layer of the system represents the lowest level component and contains that actual data for HIFIS. While the initial release of HIFIS uses a Microsoft SQL database, it can also be deployed with alternative database vendors. The HIFIS development team is working on initial support for Oracle and MySQL in addition to MSSQL.

Technical System Architecture

As the System Architecture has described, HIFIS is organized in a hierarchical structure ranging from the user interface to the underlying database. Each of these layers has one or more software components that must be deployed and configured. Here we will examine each layer's components in detail and the information necessary for deployment planning.

Figure 2 – HIFIS 4 System Design Layers With Technical Components



HIFIS.WEB

The front facing web application in the presentation layer is represented by the HIFIS.WEB component. HIFIS.WEB is a Microsoft .NET Framework 4.0 MVC 4 web application that must be hosted in Internet Information Services (IIS) 7 or above. As per the system's design, HIFIS.WEB does not include any of the formal HIFIS business logic, it is simply a user interface for interacting with the web services hosted in the middle-tier to perform all of the system operations and displays the results. Consequently, much of the configuration effort setting up the HIFIS.WEB web application in IIS is configuring the service references and bindings in the web.config configuration file.

Since it is a Microsoft MVC 4[2] web application, when deploying the HIFIS.WEB web application, you may choose to install MVC 4 in the Global Assembly Cache (GAC) of the IIS server, or deploy the MVC 4 libraries in the HIFIS.WEB Bin folder. All other libraries and dependencies are included in the application deployment folder except Crystal Reports.

Crystal Reports is used as the runtime report engine in HIFIS.WEB and can only be installed in the GAC. If Crystal Reports is not already installed on the server, the HIFIS.WEB installation package provides an installer for the required Crystal Reports runtime engine library files.

[2] http://www.asp.net/mvc/mvc4

Customizing the user interface can be achieved through several means. Since HIFIS.WEB is an MVC 4 application, each of the web views are represented by .cshtml files in the View folder of the deployment location. These files contain a combination of html, JavaScript and a special view syntax called Razor, and each can be modified to alter the way HIFIS.WEB displays information in the web browser. Be aware that modifying these files could break your HIFIS.WEB web application and should only be attempted by those with explicit knowledge of the .NET MVC framework and the Razor syntax. You can learn about MVC and Razor from the Microsoft website.

HIFIS.WEB also uses an interface toolkit called WET. It can be found in the content folder of the HIFIS.WEB web application deployment folder. WET consists of a series of CSS and JavaScript files to provide a suite of interface components that are used throughout the HIFIS.WEB. As with modifying the .cshtml files in the View folder, changes to most of the files in WET could break the HIFIS.WEB web application without a solid understanding of WET in general, CSS, and JavaScript. The best way to implement your own look and feel for HIFIS.WEB is to use one of the provided WET themes and modify it to meet your requirements then deploy it in the content folder of your HIFIS.WEB installation. You can learn about WET by visiting its documentation website[3].

HIFIS.DOMAIN

The middle-tier component, called HIFIS.DOMAIN, consist of a series of Microsoft .NET Windows Communication Foundation (WCF) web services that provide application logic functionality and data access to the HIFIS data for applications, like the HIFIS.WEB web application. HIFIS.DOMAIN deploys similar to how HIFIS.WEB is deployed, into an IIS application folder and is configured in much the same way through an app.config file. As with HIFIS.WEB, the bulk of the configuration is related to properly setting up the service endpoints and bindings. Because HIFIS.DOMAIN deploys as a separate component, it can be deployed to a different physical server instance than was used for HIFIS.WEB. HIFIS.DOMAIN also handles authentication which does require that you also configure a membership provider[4] in the app.config file.

HIFIS.DOMAIN must also include an installation of a Crystal Reports Runtime Engine in order to respond to report requests from HIFIS.WEB. A key strategy in the design of HIFIS 4 was to eliminate direct access to the HIFIS database by any components outside of the middle-tier web services. Crystal Reports, however, does require direct access to the underlying database in order to generate report results. To resolve this dependency on direct database access HIFIS.DOMAIN was designed with the ability to run Crystal Reports in the middle-tier environment with managed access to the database. The report results are then packaged and

[3] http://wet-boew.github.io/v3.1-ci/docs/index-en.html

[4] http://msdn.microsoft.com/en-us/library/tw292whz(v=vs.100).aspx

provided through the reporting web services to HIFIS.WEB to be displayed in a browser. Since the report is generated in the middle-tier but displayed in the presentation level, the runtime engine is required in both places.

An existing or alternate installation of Crystal Reports can be used in either HIFIS.WEB or HIFIS.DOMAIN as long as they both support Crystal Reports 2008 report format or better.

DATABASE

As it has been indicated already, HIFIS deploys with a Microsoft SQL Server database by default for both the HIFIS database and the membership provider. Like HIFIS.DOMAIN, because the database itself is a separate component it can also be deployed on a different physical server from HIFIS.WEB and HIFIS.DOMAIN.

By default, HIFIS uses Microsoft SQL Server 2008 R2 (with Advanced Services if using the Express edition). If a compatible SQL server already exists on the infrastructure being used for HIFIS, you could configure your deployment to utilize the existing instance and simply create a new database for the deployment. HIFIS includes the necessary scripts in the installer to create both the initial HIFIS database and a membership provider database in a Microsoft SQL environment. Installation tools for other platforms including Oracle and MySQL are under development and will be available as well.

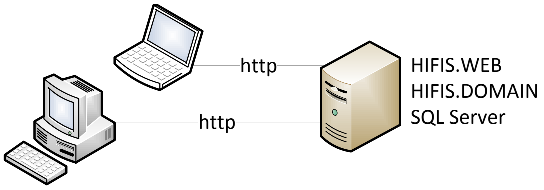
Deployment Scenarios

The HIFIS web application follows a service-oriented architecture (SOA) and therefore provides great freedom with respect to how the components are deployed.

SINGLE SERVER

The most common deployment scenario is to place everything on a single server in a private network. This means that HIFIS.WEB and HIFIS.DOMAIN are installed on a single instance of IIS on a Windows server, along with the database.

Figure 3 – Single Server Deployment

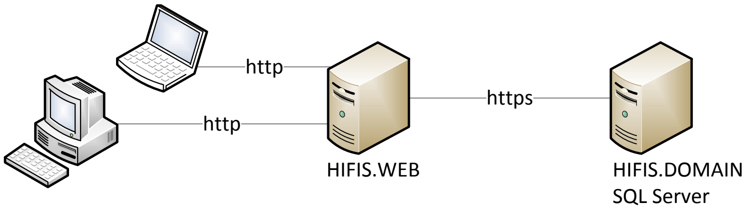


The interactions between the web application, services, and database are all occurring on the same physical server with no information being shared across a network which in turn mitigates any security risks with information being transmitted over a network. A user accessing the web application however would be transmitting information between the server and their device with a web browser. Your own circumstances and risk tolerance and/or security practices will determine if you need to take steps to ensure secure communications between clients and the server on a private, secure network.

DISTRIBUTED INSTANCES

In a distributed configuration, HIFIS.WEB and HIFIS.DOMAIN can be separated and installed on separate instances of IIS on different physical servers.

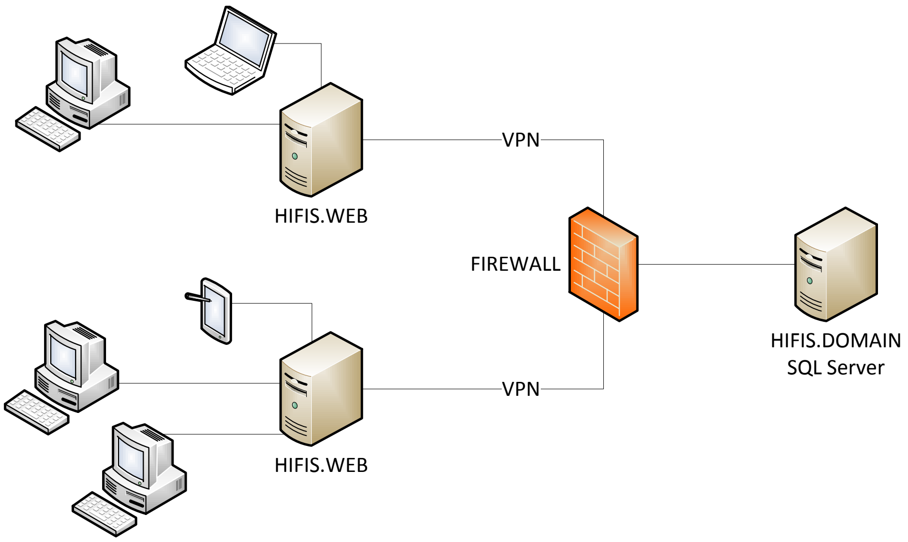
Figure 4 – Distributed Server Deployment



While separating the components on the network does introduce a potential performance penalty since data now has to travel across the network twice – once between HIFIS.DOMAIN and the HIFIS.WEB web application and again to the user's web browser, the components have been designed to minimize any delays by performing operations simultaneously where possible and ensuring that only data necessary to satisfy a user's request is travelling from the HIFIS.DOMAIN web services to the HIFIS.WEB web application.

This configuration can also be leveraged to minimize the exposure of the core HIFIS.DOMAIN web services and data to end users, while simultaneously enhancing the end user's experience. Consider the following scenario:

Figure 5 – Distributed Server Deployment with Multiple Instances



In this case a central authority is hosting the HIFIS.DOMAIN web services with the users of the HIFIS.WEB web application on external private networks. The central authority minimizes exposure of its infrastructure by providing access for HIFIS.DOMAIN to the external servers hosting HIFIS.WEB over a secure connection, like a virtual private network (VPN). In turn, the HIFIS.WEB servers provide local access to the HIFIS.WEB web application for clients on the private networks.

Also, the HIFIS.WEB web application component can be customized so that users on each of the networks are presented with a version of HIFIS that is themed and customized according to their needs and operating environment, all while remaining part of the same integrated system.

Authentication, Authorization and Security

The default installation and deployment of HIFIS 4 includes everything needed for authentication, but there are many ways this can be configured to meet your environment and operational requirements.

Before considering these concepts, here is a standard workflow for HIFIS 4:

- A user navigates to the login URL of the HIFIS.WEB web application.
- HIFIS.WEB returns a login web form requesting the username and password.
- The user enters a username and password and submits the web form.
- HIFIS.WEB passes the login credentials to the HIFIS.DOMAIN authentication service for authentication.
- HIFIS.DOMAIN uses the configured membership provider to authenticate the credentials. If they are valid, the service constructs an encrypted security token and passes it back to HIFIS.WEB.

  (If the credentials failed to authenticate, HIFIS.DOMAIN will respond to HIFIS.WEB with a validation error.)
- HIFIS.WEB receives the security token and can now use this to access other services in HIFIS.DOMAIN.
- The user navigates to a web page in HIFIS.WEB that requires information from HIFIS.DOMAIN.
- HIFIS.WEB passes the security token to HIFIS.DOMAIN through the desired service as a parameter in the service operation requesting data.
- HIFIS.DOMAIN uses the security token to determine if this is a valid request from an authenticated user. If it is, it proceeds to handle the service request and responds to HIFIS.WEB with the requested data. If the security token is not valid it will respond with a validation error.

AUTHENTICATION

Authentication refers to verifying that the logon credentials provided in order to access the system are valid. In HIFIS.DOMAIN, there is a membership provider configured to handle user authentication. The process for authentication in HIFIS is the same for both a user and a system – a username and a password are passed to an authentication web service that will access the configured membership provider to validate the credentials. If they are valid, a special security token is created by the authentication web service and returned to the caller. This security token is then passed by the caller in subsequent service calls to interact with the HIFIS services. This token is valid until it either times out or the application revokes its authorization.

The .NET framework provides two membership providers, one for SQL and one for Active Directory. HIFIS uses the SQL provider by default so that user account information is stored in an SQL database. The SQL membership provider is also specifically designed for use with a Microsoft SQL database, which is also the default database platform for HIFIS. By default, HIFIS is configured to use the same database for both the HIFIS data and the membership provider data. If you want your HIFIS users to login with their credentials from Active Directory you can edit the app.config file to use the Active Directory membership provider and configure its settings according to your information.

If you have other requirements not met by these configuration options, you can choose to create a custom membership provider. This requires writing .NET code to implement the .NET System.Web.Security.Membership interface for a custom provider, deploying your custom library to the HIFIS.DOMAIN bin folder in your IIS deployment, and editing the app.config file to use your custom implementation. In this way you can create an authentication component for your HIFIS deployment that uses whatever you require, like a database from another vendor, or some other mechanism altogether.

AUTHORIZATION

Aside from the credentials that users provide when logging in to access HIFIS, there are other features that can be configured to provide further authorization and to secure the information being exchanged between components of the system. Built into Windows and IIS are tools that you can enable that will provide greater control of who or what can access HIFIS. For example, you can use IP address or domain restrictions to prevent unauthorized devices from connecting to the web application or the web services. You can also edit the binding configuration for the WCF web services to define the access parameters for your clients, including requiring them to use a secure connection and provide other user credentials.

It is important to note that this is in addition to the user authentication that is done by HIFIS – consider the workflow described above where the HIFIS.WEB web application must already have access to the HIFIS.DOMAIN authentication web service before it can even pass in the HIFIS user credentials. This implies that there is a wraparound authorization and security context that can be configured and managed separately from the internal authorization features of HIFIS itself. These wraparound features are part of IIS and use a completely separate validation system like a domain account or a security certificate.

SECURITY

Thus far we have mostly been concerned with ways to authenticate and authorize access to HIFIS and its components. This simply ensures that who or whatever is interacting with HIFIS is allowed to do so. However, authenticated or authorized access does not mean it is also secure.

While the security token returned from HIFIS.DOMAIN is encrypted, the user credentials initially passed in for authentication are not. HIFIS does not encrypt data that is being transferred between the web services and the web application or another third-party application on its own. In order for that information to be encrypted and secured you must also configure IIS to use Secure Sockets Layer (SSL) encryption. If you enabled SSL for HIFIS.WEB, then users will access the HIFIS.WEB web application with the prefix https:// in their internet browser denoting a secure connection. You can also enable SSL for HIFIS.DOMAIN so that the exchange of information between HIFIS.WEB and the HIFIS.DOMAIN web services is also encrypted and secure.

Since the most common deployment configuration is to host both HIFIS.WEB and HIFIS.DOMAIN on the same IIS server instance, the HIFIS web application is setup to use non-secure HTTP bindings and message based security with Windows as the client credential type by default when communicating with the HIFIS.DOMAIN web services. If you are deploying HIFIS.WEB to a different server than HIFIS.DOMAIN, you will want to consider using SSL if the connection between the servers is insecure.

When configuring the use of SSL you have the option of installing a purchased SSL certificate from a trusted Certificate Authority or installing a self-signed certificate. If you elect to use a self-signed certificate be aware that modern web browsers may display visible warnings to the user alerting them that the web site they are visiting does not have a certificate issued by a trusted authority. This can be mitigated however by sharing the certificate with the client machines and installing it in the local certificate store.

Technical Summary

DEPLOYMENT REQUIREMENTS

The following is a comprehensive technical summary of the deployment requirements for each component of the HIFIS web application.

HIFIS.WEB

- Windows Server 2008 R2 (minimum)
- Internet Information Services (IIS) 7 (minimum)
- Microsoft .NET Framework 4.0

HIFIS.DOMAIN

- Windows Server 2008 R2 (minimum)
- IIS 7 (minimum)
- Microsoft .NET Framework 4.0

DATABASE

- Microsoft SQL Server 2008 R2 (minimum)
  **Express with Advanced Services*

Installation packages for each component are available for both 32-bit and 64-bit environments.

TECHNICAL SPECIFICATIONS

The following is a summary of the technical specifications of each of the HIFIS web application components.

HIFIS.WEB

- Microsoft .NET MVC 4 web application
- Razor template views
- Web Experience Toolkit (WET) 3.18 (jQuery 2.03)
- Crystal Reports Runtime Engine supporting RPT files created in Crystal Reports 2008 or higher.

The Razor template view files from the MVC 4 framework in HIFIS.WEB can be customized with Razor syntax, HTML, and JavaScript. HIFIS also uses the WET base theme which can be customized through CSS and the Razor templates. Customizations could include branding the application to provide a more integrated user experience and including additional organization content and links in views.

HIFIS.DOMAIN

- Microsoft .NET Framework 4.0 Windows Communications Foundation (WCF)
- Microsoft .NET Entity Framework 4.3
- Crystal Reports Runtime Engine[5] supporting RPT files created in Crystal Reports 2008 or higher.

[5] In order to allow end users to build their own custom Crystal Reports for HIFIS 4, the underlying data source must be the SQL database since end users cannot access the Entity Framework data model as a data source in Crystal Reports when designing a report. However, in the architecture of HIFIS, users do not have access to the SQL database. For this reason, Crystal Reports is required in both HIFIS.WEB and HIFIS.DOMAIN. When a user runs a report, the Crystal Reports Runtime Engine in HIFIS.DOMAIN will load and execute the report against the SQL database. It will then send the report object with the data embedded to HIFIS.WEB where another Crystal Reports Runtime Engine will render the report object for the web browser.