



HIPS

Topology and Configuration Guide

17 April 2020 7.3

Approved for external use

Acknowledgements

Council of Australian Governments

The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

HL7 International

This document includes excerpts of HL7™ International standards and other HL7 International material. HL7 International is the publisher and holder of copyright in the excerpts. The publication, reproduction and use of such excerpts is governed by the [HL7 IP Policy](#) and the HL7 International License Agreement. HL7 and CDA are trademarks of Health Level Seven International and are registered with the United States Patent and Trademark Office.

Disclaimer

The Australian Digital Health Agency (“the Agency”) makes the information and other material (“Information”) in this document available in good faith but without any representation or warranty as to its accuracy or completeness. The Agency cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document control

This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

Copyright © 2020 Australian Digital Health Agency

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the permission of the Australian Digital Health Agency. All copies of this document must include the copyright and other information contained on this page.

OFFICIAL

Document information

Key information

Owner	National Health Chief Information Officer, Infrastructure Operations
Contact for enquiries	Australian Digital Health Agency Help Centre
Phone	1300 901 001
Email	help@digitalhealth.gov.au

Table of contents

1	Introduction	5
1.1	Purpose	5
1.2	Scope.....	5
1.3	Assumptions.....	5
1.4	Definitions and acronyms	5
2	Server Roles	6
2.1	Database Server	6
2.1.1	Overview.....	6
2.1.2	Allocation of components	6
2.1.3	Software requirements.....	6
2.2	Application Server	6
2.2.1	Overview.....	6
2.2.2	Allocation of components	7
2.2.3	Software requirements.....	7
2.3	ESB Server (optional)	8
2.3.1	Overview.....	8
2.3.2	Allocation of components	8
2.3.3	Software requirements.....	8
2.4	Load Balancer (optional).....	8
2.4.1	Overview.....	8
3	Dependencies.....	9
3.1	Infrastructure dependencies.....	9
3.1.1	Active Directory	9
3.1.2	Certificates.....	9
3.1.3	Network	10
4	Topologies.....	11
4.1	Single server topology.....	11
4.2	Dual server topology.....	11
4.3	High availability topology.....	12
5	Server Role Specifications	15
6	Server Role Guidance.....	16
6.1	Database Server	16
6.2	Application Server	16
6.3	ESB Server (optional)	18
6.4	Load Balancer.....	18

1 Introduction

1.1 Purpose

To provide guidance for implementers on the best practice for HIPS topology and configuration to optimise performance.

1.2 Scope

HIPS Topology and Configuration Guide.

1.3 Assumptions

The following assumptions have been made in the development of this document:

- Nil

1.4 Definitions and acronyms

Item	Definition
WCF	Windows Communication Foundation https://docs.microsoft.com/en-au/dotnet/framework/wcf/

2 Server Roles

A typical HIPS deployment topology consists of a number of server roles that support the various HIPS components. In non-Production environments these roles may be fulfilled by a single physical or virtual server. In Production environments, it is recommended that each role is fulfilled by at least one separate physical or virtual server.

The following sub-sections provide an overview of each server role, describe how components may be allocated to roles, and the software requirements for the server role.

2.1 Database Server

2.1.1 Overview

The Database Server role hosts the databases utilised by HIPS for the HIPS Core, HIPS UI and (optionally) MirthConnect components. The databases must be hosted in a SQL Server instance, which may be either a single server or a SQL Server high availability cluster.

2.1.2 Allocation of components

Depending on the HIPS products to be deployed, the following components are typically allocated to the Database Server role:

- HIPS Core database
- HIPS UI database
- MirthConnect database

Databases may be allocated to either a single Database Server or across multiple Database Servers.

2.1.3 Software requirements

The Database Server role has the following software requirements:

Requirement	Software
Operating System	Windows Server 2008 R2
	Windows Server 2012 R2
	Windows Server 2016
Database Management System	SQL Server 2008 R2
	SQL Server 2012
	SQL Server 2014
	SQL Server 2016

2.2 Application Server

2.2.1 Overview

The Application Server role hosts the runtime components utilised by HIPS for the HIPS Core and HIPS UI components. The majority of components are hosted in the Internet Information Services (IIS) web server or installed as Windows Services managed by the operating system.

2.2.2 Allocation of components

Depending on the HIPS products to be deployed, the following components are typically allocated to the Application Server role and installed into IIS:

- HIPS Core web services
- HIPS UI web application

These components may be deployed in various configurations, such as:

- Single website with separate web applications on the same Application Server
- Separate websites on the same Application Server
- Single or separate websites distributed across multiple Application Servers

When multiple instances of the same component are distributed across multiple Application Servers it is typical to access them via the optional Load Balancer role (for example in a high availability topology).

The following components are typically allocated to the Application Server role and installed as Windows Services managed by the operating system:

- HIPS Core background process: queue consumer (HL7 Reports)
- HIPS Core background process: queue consumer (MHR Operations)
- HIPS Core background process: queue consumer (HL7 Acknowledgements)
- HIPS Core background process: alert monitoring

At least one instance of each listed queue consumer background process must be installed and running for the correct function of HIPS Core. It is however possible to deploy queue consumer instances in various configurations, such as:

- A single instance of each queue consumer on a single Application Server
- Multiple instances of the same queue consumer on a single Application Server
- Single or multiple instances of each queue consumer distributed across multiple Application Servers

The HIPS queue consumer infrastructure ensures that multiple instances of the same consumer safely interact with the HIPS message queue.

2.2.3 Software requirements

The Application Server role has the following software requirements:

Requirement	Software
Operating System	Windows Server 2008 R2 Windows Server 2012 R2 Windows Server 2016
Runtime	.NET Framework 4.5.2+
Scripting	Windows PowerShell 5.0+
Web Server	Internet Information Services

2.3 ESB Server (optional)

2.3.1 Overview

The optional ESB Server role hosts the MirthConnect ESB to facilitate integration between healthcare systems and the HIPS Core web services. The ESB Server role using MirthConnect is only required when the site does not have an alternative means for integration between their healthcare systems and the HIPS Core web services, such as their own ESB.

2.3.2 Allocation of components

If MirthConnect is to be deployed, the following components are typically allocated to the ESB Server role:

- MirthConnect runtime

2.3.3 Software requirements

Refer to the MirthConnect documentation at

<http://www.mirthcorp.com/community/wiki/display/mirth/System+Requirements>.

2.4 Load Balancer (optional)

2.4.1 Overview

The optional Load Balancer role provides load balancing of HTTP-based interactions with the web-based components deployed to the Application Server role. It is typically employed to distribute load when there are multiple Application Server instances, such as in a high availability topology.

The implementation of the load balancer is outside the scope of this document, except to note the applicable guidance provided in the *Server Role Guidance* section of this document.

3 Dependencies

3.1 Infrastructure dependencies

3.1.1 Active Directory

HIPS uses Active Directory to secure its internal connections.

It is recommended that an Active Directory service account is created and used for the following:

- As the identity for the IIS application pools associated with the HIPS Core web services and HIPS UI web application
- As the identity used by the HIPS Core background processes when installed as Windows Services

The service account should be configured with no expiry and no lockout. In a typical configuration it will also be used by the listed HIPS components to connect to the HIPS databases hosted in SQL Server.

3.1.2 Certificates

HIPS utilises certificates installed into the Certificate Store on each Application Server. The following types of certificates need to be installed on each Application Server:

- NASH PKI Certificate for Healthcare Provider Organisation (for packaging clinical documents and if connecting to the My Health Record system as a Healthcare Provider Organisation)
- NASH PKI Certificate for Supporting Organisation (if connecting to My Health Record as a Contracted Service Provider)
- DHS Site PKI Certificate for Healthcare Provider Organisation (if connecting to the HI Service as a Healthcare Provider Organisation)
- DHS Site PKI Certificate for Contracted Service Provider (if connecting to the HI Service as a Contracted Service Provider)

3.1.3 Network

HIPS has the following requirements for network communications between nodes:

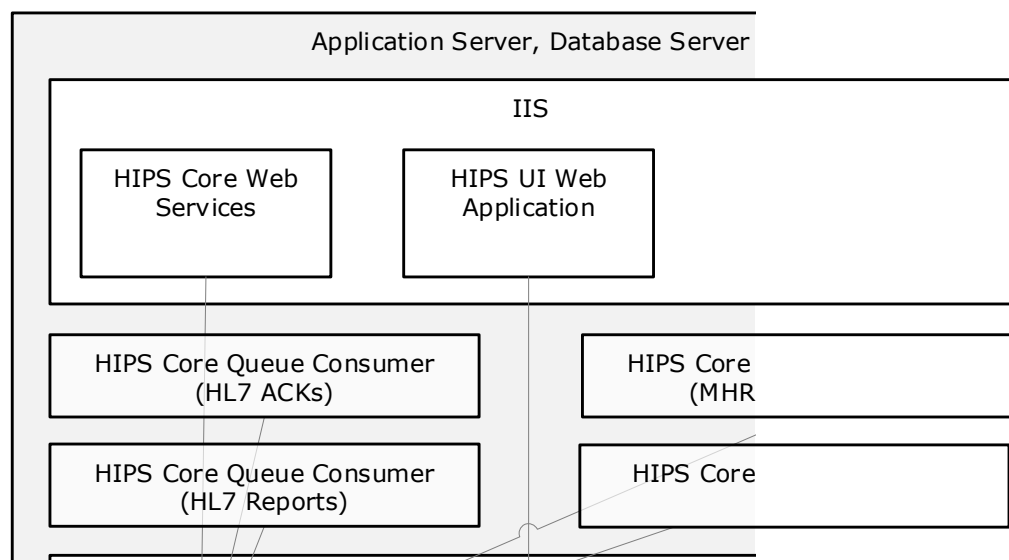
Name	Origin (server role)	Destination	Port	Protocol
HI Service	Application Server	Production https://www3.medicareaustralia.gov.au/pcert/soap/services/ Non-Production https://www5.medicareaustralia.gov.au/cert/soap/services/	443	HTTPS
MHR System	Application Server	Production https://services.ehealth.gov.au:443/ Non-Production https://b2b.ehealthvendortest.health.gov.au/	443	HTTPS
NASH CA CRL Distribution Point	Application Server	Production https://www.certificates-australia.com.au/cgi-bin/getcrl_health.pl?DN=cn=Medicare%20Australia%20Organisation%20Certification%20Authority,ou=Medicare%20Australia,o=GOV,c=AU Non-Production http://matest.certificates-australia.com.au/TestMAOCA/latest.crl https://matest.certificates-australia.com.au/MAOCA2/latest.crl	80	HTTP
SQL Server	Application Server	Database Server		TDS
Active Directory	Application Server	Domain Controller		LDAP/S
Load Balancer (HIPS-Core)	Load Balancer	Application Server		HTTP
Load Balancer (HIPS-UI)	Load Balancer	Application Server		HTTP

4 Topologies

4.1 Single server topology

The single server topology consists of a single physical or virtual server performing both the Database Server and Application Server roles. It should only be employed in low performance non-Production environments for the purposes of evaluation.

The diagram below illustrates the allocation of HIPS components in the single server topology:

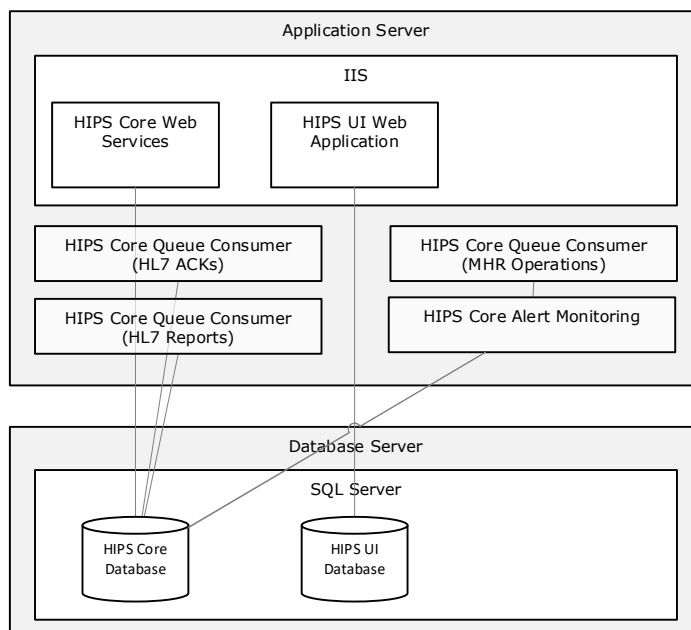


4.2 Dual server topology

The dual server topology consists of separate physical or virtual servers performing the Database Server and Application Server roles respectively. It is suitable for moderate performance Production environments.

It may also be employed in non-Production environments for purposes such as evaluation or integration development or testing.

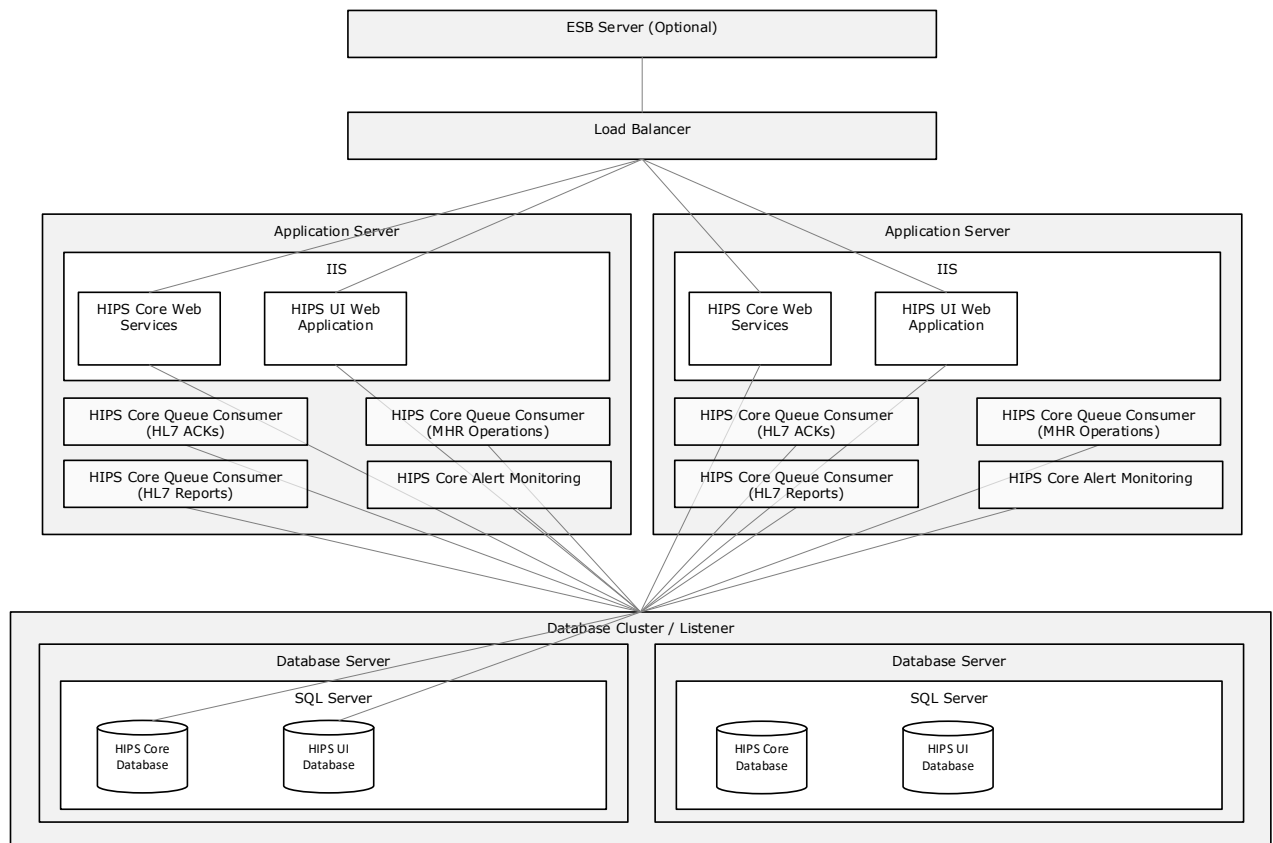
The diagram below illustrates the allocation of HIPS components in the dual server topology:



4.3 High availability topology

The high availability topology consists of multiple physical or virtual servers performing each of the Database Server and Application Server roles, typically also utilising the Load Balancer and optional ESB Server roles. It is suitable for highly available Production or non-Production environments.

The diagram below illustrates the allocation of HIPS components in the high availability topology:



In the high availability topology:

- The HIPS-Core Web Services component is deployed to multiple Application Servers, with HTTP/HTTPS traffic distributed between servers by the Load Balancer. This distributes load but also ensures that if one Application Server becomes inactive, the other Application Server can still continue to service requests via the Load Balancer.
- The HIPS-UI Web Application component is deployed to multiple Application Servers, again with HTTP/HTTPS traffic distributed between servers by the Load Balancer¹. This primarily serves to distribute load.
- The HIPS-Core Queue Consumer components are deployed to multiple Application Servers, with each component active on each Application Server. This distributes load but also ensures that if one Application Server becomes inactive, the other Application Server continues its background processing.
- The HIPS-Core Alert Monitoring service is deployed to multiple Application Servers. This distribution is crucial for several alert items, in particular the certificate and application server disk space checks.

NOTE: Having multiple alert monitoring components installed will result in duplicate alerts being generated from each server, however it is possible to use the custom host log4net property to configure a filter for these duplicates.

¹ As noted in section 6.4 *Load Balancer*, the HIPS-UI Web Application component does not support distributed sessions.

- The HIPS-Core and HIPS-UI database components are deployed to multiple Database Servers in a Database Cluster, with one Database Server being active at any time. While this does not distribute load, it does ensure that if the active Database Server node becomes inactive, the other Database Server node will become active and provide these services. The Database Cluster employed in the high availability topology is typically implemented as a SQL Server Always On Availability Group, with the HIPS components deployed to the Application Servers configured to connect to the Availability Group Listener name rather than a specific Database Server node name or instance.

Further, while the preceding diagram illustrates a SQL Server cluster performing the Database Server role and two servers performing the Application Server role, it is important to note that this can be further scaled as required, for instance:

- By distributing HIPS databases to separate SQL Server instances or separate SQL Server clusters
- By distributing HIPS web runtime components and HIPS background processing components to separate servers

5 Server Role Specifications

The following table shows the recommended specifications for a Production environment based on HIPS usage:

Usage Profile	Low Usage	High Usage
	HIPS is used by 3 or fewer facilities for viewing and uploading less than 1,000 documents per day	HIPS is used by multiple facilities for viewing and uploading up to 150,000 documents per day
Database Server²		
Processor	1 x 2 cores	2 x 8 cores
Memory	4GB	56GB
Storage		
System Volume	100GB	100GB
Data Volume	200GB + 200GB per year	800GB + 300GB per year
Log Volume	100GB	260GB
Index Volume	80GB	160GB
Backup Volume ³	1TB + 500GB per year	2TB + 1TB per year
TempDB Volume	40GB	80GB
Application Server		
Processor	1 x 2 cores	2 x 8 cores
Memory	4GB	24GB
Storage		
System Volume	50GB	50GB
Application Volume	30GB	30GB

² Specifications represent resources allocated to the SQL Server instance hosting the HIPS databases.

³ Retaining 4 daily backups of each database.

6 Server Role Guidance

6.1 Database Server

- **Database file sizing & growth**
 - It is recommended that the initial file size be pre-set to a large size (at least 100GB each) to reduce the need to auto-grow frequently.
 - Auto-grow settings for these files should be configured with a MB value rather than a % value, otherwise once the file is large the auto-growth can be a time-consuming process and can cause errors.

NOTE: For a managed Production system, you should consider auto-grow to be merely a contingency for unexpected growth. You should avoid managing your data and log growth on a day-to-day basis with auto-grow.
- **Configuring database backups & log truncation**
 - Periodic back-ups and log truncation should be performed on all databases.
- **Monitoring database file size & growth**
 - It is important to monitor the database file size regularly to ensure the files do not grow to completely consume available disk capacity.
- **Index maintenance**
 - HIPS Core database indexes should be rebuilt periodically to help improve performance and reduce fragmentation.

6.2 Application Server

- **Configure .NET Max Connections**
 - The `system.net/connectionManagement` element limits the number of concurrent connections from HIPS to remote endpoints including the HI Service and My Health Record B2B Gateways.
 - This element is configured to allow 50 connections to each remote host in the default HIPS configuration files.
 - For further information refer to [<connectionManagement> Element \(Network Settings\)](#).
- **Configure WCF Service Throttling Behaviour**
 - The `system.serviceModel/behaviors/serviceBehaviors/behavior/serviceThrottling` element limits the number of concurrent calls to HIPS Core web services. Implementers may choose to tune these settings to make the maximum efficient use of available server capacity.
 - All other services use the `metadataSupport` behaviour where the WCF defaults of `maxConcurrentCalls = 16 × processor count`, `maxConcurrentSessions = 100 ×`

processor count, and `maxConcurrentInstances = 116 × processor count` apply. This allows a more powerful server to automatically have higher limits.

- For further information refer to [<serviceThrottling> Element \(WCF Configuration Schema\)](#)
- **Configure the HIPS-Core Queue Consumer to maximise throughput**
 - The key settings that affect Queue Consumer throughput are:
 - `QueueConsumer.BatchSize`: Defaults to 30
 - `QueueConsumer.PollingInterval`: Defaults to 10
 - `QueueConsumer.BatchDegreeOfParallelism`: Defaults to empty (maximum)
 - The default configuration settings for the HIPS-Core Queue Consumer are designed to provide a reasonable default for most sites.
 - On an otherwise idle Application Server with both the HI Service and My Health Record System available each Queue Consumer will achieve throughput of approximately 3 messages per second, with the Queue Consumer idle 33%.
 - This allows some headroom for other work being performed by the Application Server, as well as HI Service and My Health Record System interactions taking longer.
 - Queue Consumer throughput can however be further adjusted as follows:
 - By reducing the *PollingInterval* while maintaining the *BatchSize*. For example, by configuring the *PollingInterval* to 5 and the *BatchSize* to 30, it may be possible to achieve throughput of up to 12 messages per second, depending on other work.
 - Similar throughput could possibly be achieved by increasing the *BatchSize* while maintaining the *PollingInterval*. For example, by configuring the *BatchSize* to 80 and the *PollingInterval* to 10, it may be possible to achieve throughput of up to 8 messages per second, depending on other work.
 - Both preceding configurations would leave the Queue Consumer idle 0%, but are dependent on the impact of other work to achieve the cited throughput.
 - It is also important to note the effect of the *BatchDegreeOfParallelism* setting on throughput. This setting affects the way that messages are processed once acquired as a batch at each polling interval. By setting *BatchDegreeOfParallelism* to empty, messages will be processed on as many separate threads as are available to be allocated from the .NET Thread Pool.
 - While in most cases this default is acceptable, it is dependent on the availability of threads from the .NET Thread Pool.
 - For more certainty on the number of threads allocated, the *BatchDegreeOfParallelism* setting can be set to a specific positive integer value. If the configured number of threads are available from the .NET Thread Pool, as many messages from the batch as there are threads allocated will be processed in parallel.
 - For instance, if *BatchDegreeOfParallelism* was set to 5 and *BatchSize* set to 20, and all 5 threads were available from the .NET Thread Pool, then 5 messages

from the batch would be processed in parallel, repeated 4 times in total to completely process all 20 messages acquired by the batch.

- Any adjustment to the default configuration settings should be verified in a suitable non-Production environment.
- **Configure HIPS-UI to use HTTPS & HTTP/2**
 - It is strongly recommended that HIPS-UI is configured such that end-user web browsers connect to HIPS-UI over HTTPS. Depending on the server topology, the IIS instance that hosts the HIPS-UI web site components may handle HTTPS itself, or a load balancer or reverse proxy in front of one or more IIS instances may handle HTTPS and proxy web requests back to the IIS instances over either HTTPS or an unencrypted HTTP channel.
 - Further to this, to maximise performance for implementing sites that are using modern web browsers, HIPS-UI should be accessed using HTTPS via HTTP/2. This is provided natively by IIS 10+ (Windows Server 2016+), and is also provided by modern load balancers. HTTP/2 allows web browsers to multiplex requests on the same connection, rather than being constrained to 1 request per connection. When using HTTP/1.x, web browsers are limited to sending 2-8 requests simultaneously (web browser dependent). When using HTTP/2 the web browser can send multiple requests multiplexed on the same connection. This can significantly improve the total response time for HIPS-UI pages. It should also be noted that the use of HTTP/2 may require some additional configuration of web browser settings, which could for instance be conducted via Windows Group Policy.

6.3 ESB Server (optional)

- **Queue buffer & memory heap size**
 - MirthConnect has a configurable queue buffer size that allows implementers to control how many messages can reside in memory at any given point. The default is 1000, but performance can be improved by increasing the size to allow more messages to be stored in memory. As the queue buffer size is increased, the max heap size must also be increased to ensure the memory can hold the number of messages.
 - For more information refer to [this forum post](#).
- **Channel properties / message storage**
 - Non-Production: "Development"
 - Production: "Production"

6.4 Load Balancer

- **Session management**
 - HIPS UI: Requires session persistence ("sticky sessions"), recommend "Source IP" for maintaining session
 - HIPS Core: Does not require session persistence
- **Endpoint monitoring**
 - Interval: No more frequently than 60 seconds

- Expected Response Time: < 30 seconds
- Expected Response: 200 OK
- HIPS Core web services (GET):
 - `http://servername:port/HIPS.Service.AckService.svc`
 - `http://servername:port/HIPS.Service.AssistedRegistrationService.svc`
 - `http://servername:port/HIPS.Service.CdaService.svc`
 - `http://servername:port/HIPS.Service.ConsentService.svc`
 - `http://servername:port/HIPS.Service.DatabaseLoaderService.svc`
 - `http://servername:port/HIPS.Service.HiReferenceService.svc`
 - `http://servername:port/HIPS.Service.HpiiService.svc`
 - `http://servername:port/HIPS.Service.IHIService.svc`
 - `http://servername:port/HIPS.Service.PatientService.svc`
 - `http://servername:port/HIPS.Service.PCEHRService.svc`
 - `http://servername:port/HIPS.Service.ReferenceService.svc`
- HIPS UI web application (GET)
 - Request to HIPS UI endpoint (root of site)