RESTful API Infrastructure & Interoperability Services

Supporting Documentation

Last Modified August 3rd, 2021

Table of Contents

**Click the section title to jump to the desired page.*

Overview

The RESTful API Infrastructure allows MEDITECH and third party vendor software to securely access the MEDITECH EHR through APIs. Interoperability Services (or IOPS) — which is a component of the RESTful API Infrastructure and installed on the same machine(s) — adds a set of APIs to meet Meaningful Use Stage 3 (MU3) and Imaging Appropriate Use Criteria (AUC) requirements.

RESTful API is independent from any other web products or interoperability interfaces MEDITECH offers and requires dedicated hardware.

Notes

- "Proxy" and "Load Balancer" are used interchangeably throughout this document and always refer to the appliance or service which will proxy the traffic to the RestAPI Infrastructure server(s).

Requirements

Although some hardware and software requirements are listed below, the full and detailed list can be found here.

API Server(s)

An optimal RESTful API Infrastructure configuration consists of two or more servers running the RESTful API services as well as Interoperability Services. The cluster helps to ensure better performance and failover protection for the Infrastructure. However, as a minimum configuration, organizations can start out with one API server. More servers can be added based upon usage and performance needs.

Note: IIS is not used in this configuration and should not be installed on these servers. The RESTful API Infrastructure performs the role of a web server.

Purpose

These servers host the web services which clients connect to.

Support

MEDITECH provides support.

Hardware/Software

Please review the TECH task outlining your hardware configuration proposal (HCP).

Cache Server

The Redis service, which MEDITECH will install on the Cache Server, reduces latency and increases performance on requests by reducing the number of hits to the database.

The cache is memory-only, meaning it is never persisted to disk.

It is suggested that the Redis service run on its own server (as seen here). However, site may choose to have MEDITECH install the Redis service on one of the API servers (as seen here) if additional servers cannot be obtained. If combining the two servers, it is suggested you increase the RAM available to that API server by 4GB, bringing the total to 8GB for that one API server.

Please review the related security documentation to secure the Redis installation. It can be found here: RESTful API & IOPS Resources - Security.

Purpose

This server caches responses and also acts as a messaging service between API Servers.

Support

MEDITECH provides support.

Hardware/Software

Please review the TECH task outlining your hardware configuration proposal (HCP).

Database Server

A PostgreSQL database is required.

Purpose

The database stores the configuration and runtime details of the RESTful services. It does not store patient data nor does it store any other data that is stored in the MEDITECH database.

Support
MEDITECH does not provide support for deploying, maintaining, or configuring the database service(s).

TCP Proxy Service(s)
A TCP Proxy Service is connected to by an API Service when processing APIs whose data originates in the MEDITECH Expanse Platform. In the same manner that a cluster of API Servers is recommended to improve performance and fault tolerance, two or more TCP Proxy Services are also recommended.

Purpose
These services proxy connections from the API Server(s) to processes running MEDITECH Expanse software.

Support
MEDITECH provides support.

Hardware/Software
Please review the TECH task outlining your hardware configuration proposal (HCP).

Load Balancer/Proxy
A load balancer or proxy is required to sit in front of the API Server(s).

Purpose
This appliance is important for the stability and security of the system. Some of the reasons are:

- When in one of the two suggested configurations (more below):
    - It allows the customer to mitigate or prevent SSL vulnerabilities and to configure the SSL parameters according to their regulatory and corporate requirements
    - It allows the customer to install and maintain their SSL certificates in a single location instead of across multiple servers/services
- Using DNS round-robin for failover does not provide graceful failover to the client - the client software needs to be smart enough to retry the connection using another IP. Depending on the client technology, this can take seconds or minutes. Whereas failover with a load balancer or proxy is nearly instantaneous.
- When configured for SSL Termination, this reduces the CPU load on the RESTful API Infrastructure servers.

Support
Sites are expected to rely on their IT staff or vendor support for configuring and maintaining the load balancer/proxy. MEDITECH provides guidance but cannot furnish sites with step-by-step instructions on how to configure their load balancer/proxy.

SSL Certificate(s)
The site must purchase one or more SSL certificates from a trusted certificate authority to be installed on their proxy or load balancer. As there will be two hostnames per MRI or HIM database(See DNS Entries below), it is suggested that the site purchase a wildcard or SAN (Subject Alternative Name) certificate to keep costs down. Sites may purchase a certificate per hostname.

Self-signed certificates may ONLY be used to secure the connection between the load balancer/proxy and the RESTful services and only when configured for SSL Bridging.

Evaluation Task
Customers preparing for MU3 or AUC initiatives should have a Hardware Evaluation Task opened under TECH to provide a system review and detailed server specifications.

If you do not currently have a task open, contact your account manager/HCIS coordinator or technical account manager (TAM) to request an evaluation. Please allow at least 2 weeks for completion — upon receipt of your evaluation, an additional 6-8 weeks should be allotted to procure and set up new hardware.

DNS Entries

A DNS entry is needed for the API and Application end-point for each MRI or HIM database (both TEST and LIVE). These records should all point to VIPs on your Load Balancer - you may use one or multiple VIPs when configuring your Load Balancer.

Another DNS entry is needed for the TCP Proxy Service. This record should point to the IPs of all the TCP Proxy servers.

Example:
If you have 3 LIVE rings with 1 HIM database each and 3 TEST rings with 1 HIM database each, we would expect the following DNS entries:
mtrestapis-live01.CUSTOMER-DOMAIN
mtrestapis-live02.CUSTOMER-DOMAIN
mtrestapis-live03.CUSTOMER-DOMAIN
mtrestapis-test01.CUSTOMER-DOMAIN
mtrestapis-test02.CUSTOMER-DOMAIN
mtrestapis-test03.CUSTOMER-DOMAIN
mtrestapps-live01.CUSTOMER-DOMAIN
mtrestapps-live02.CUSTOMER-DOMAIN
mtrestapps-live03.CUSTOMER-DOMAIN
mtrestapps-test01.CUSTOMER-DOMAIN
mtrestapps-test02.CUSTOMER-DOMAIN
mtrestapps-test03.CUSTOMER-DOMAIN

If you are configuring the TCP Proxy Service, a single DNS entry should be configured that will point to all the TCP Proxy server IPs:
mtresttcpproxy.CUSTOMER-DOMAIN

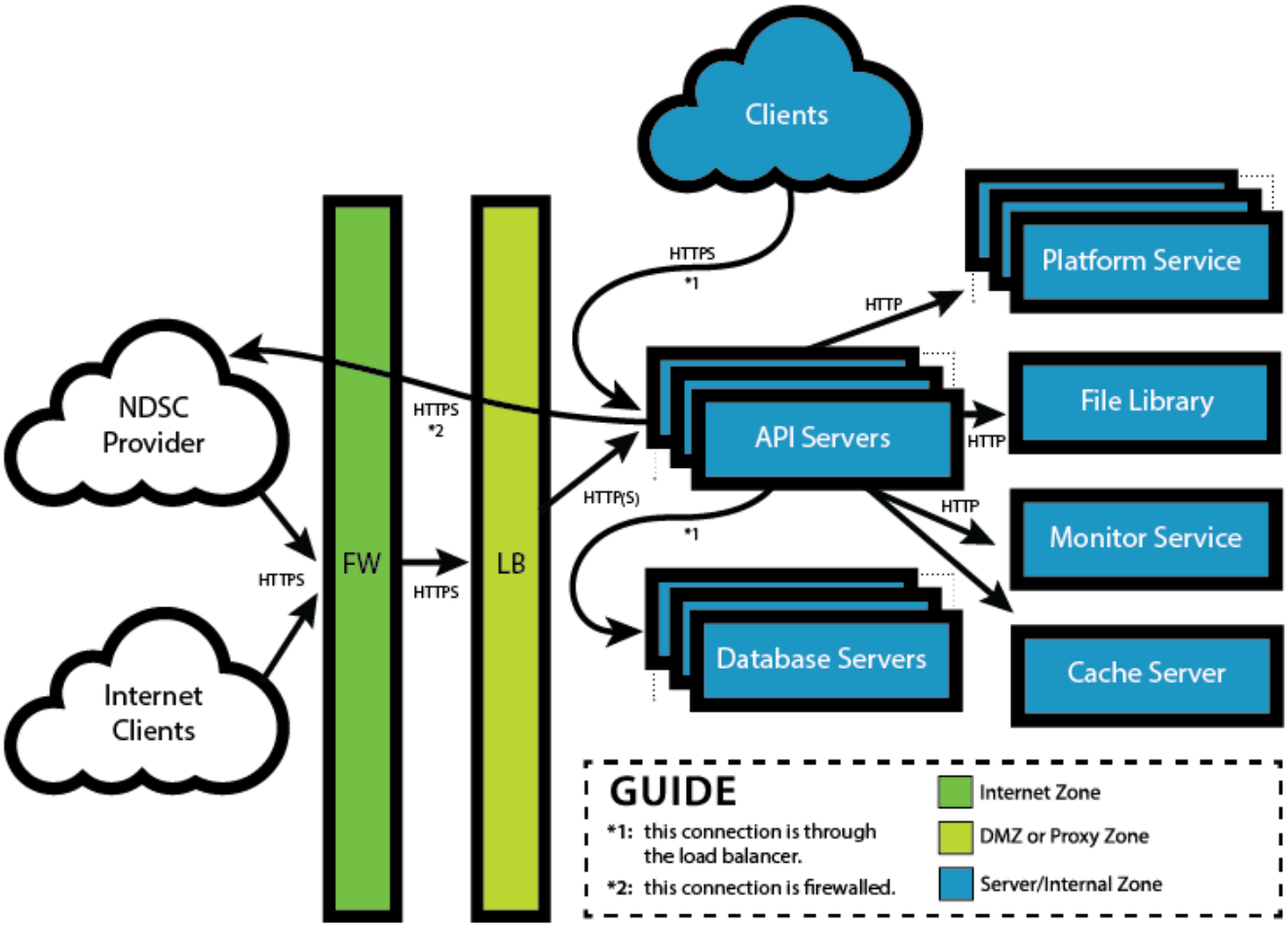
Purpose
The entries allow the infrastructure to differentiate requests as it is possible that an identifier may be reused in one or more databases. Additionally, the API and Application services run on different ports and within different processes because the workloads are significantly different

Support
Sites are expected to rely on their IT or vendor support for maintaining their DNS entries.

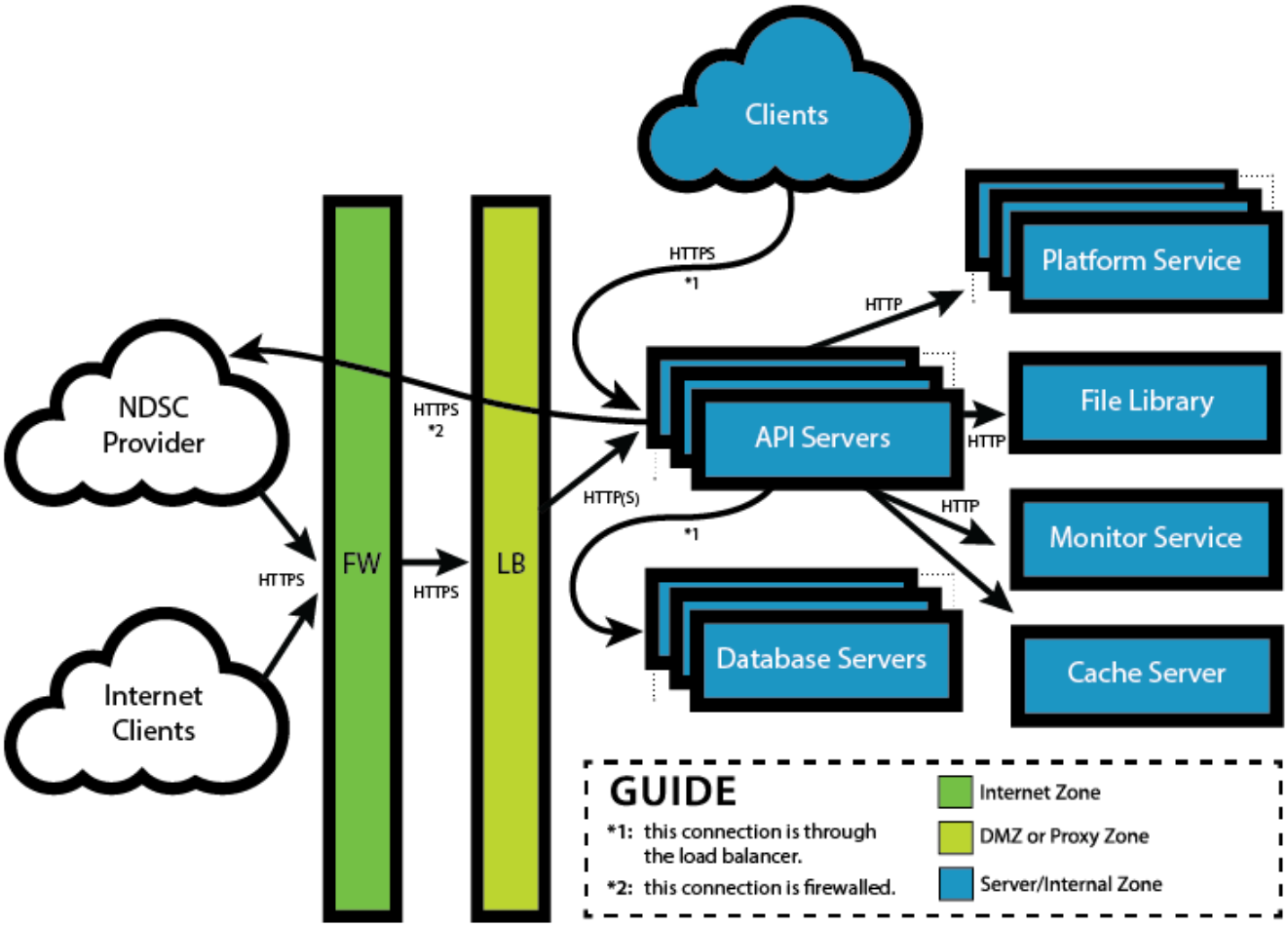
Configuration Diagrams

These diagrams show the servers and connections. A more detailed list of protocols and ports are listed in the Connectivity section.

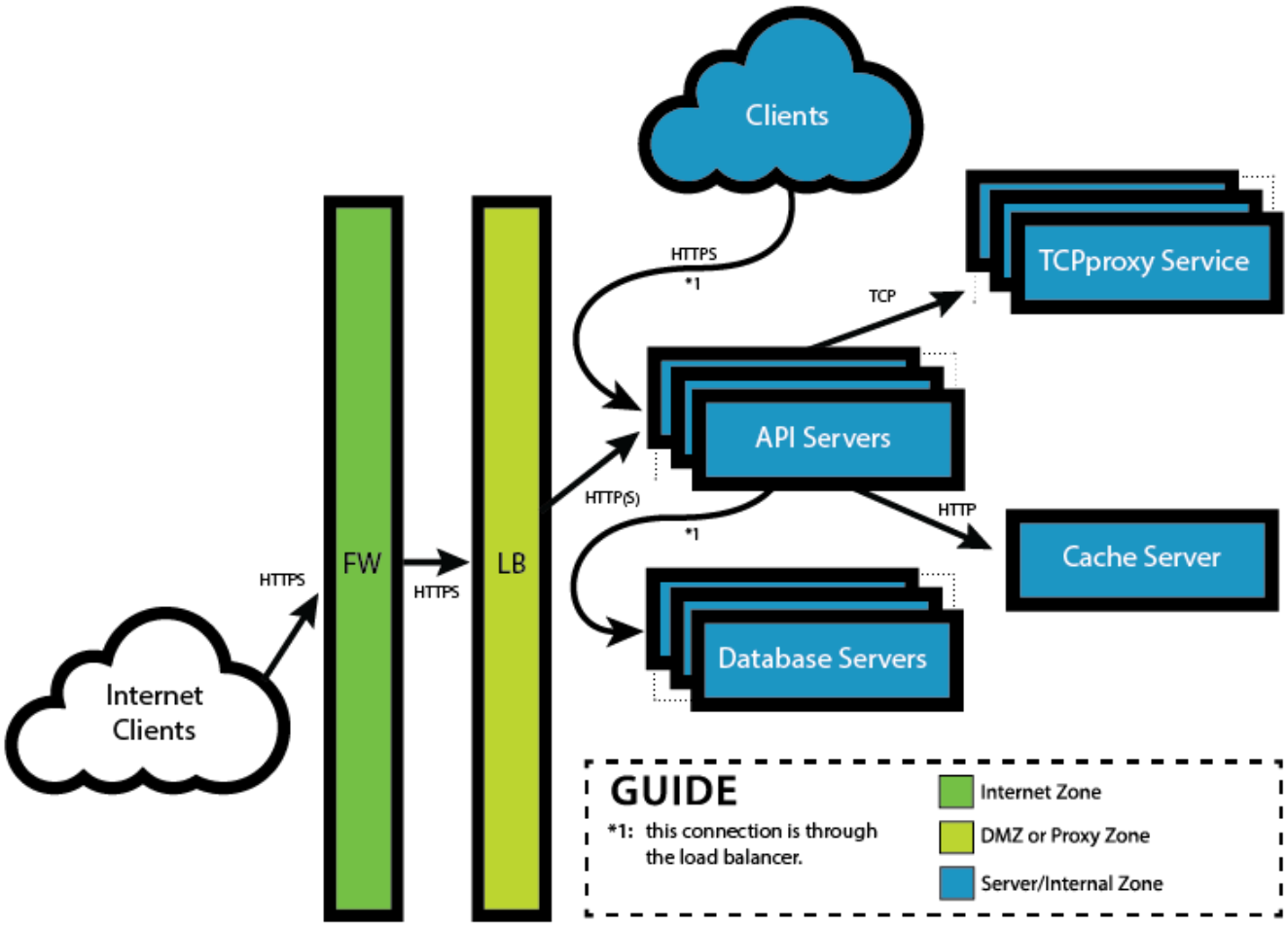
Clinical Decision Support Mechanism (CDSM) for AUC (Vendor such as NDSC) Optimum Configuration



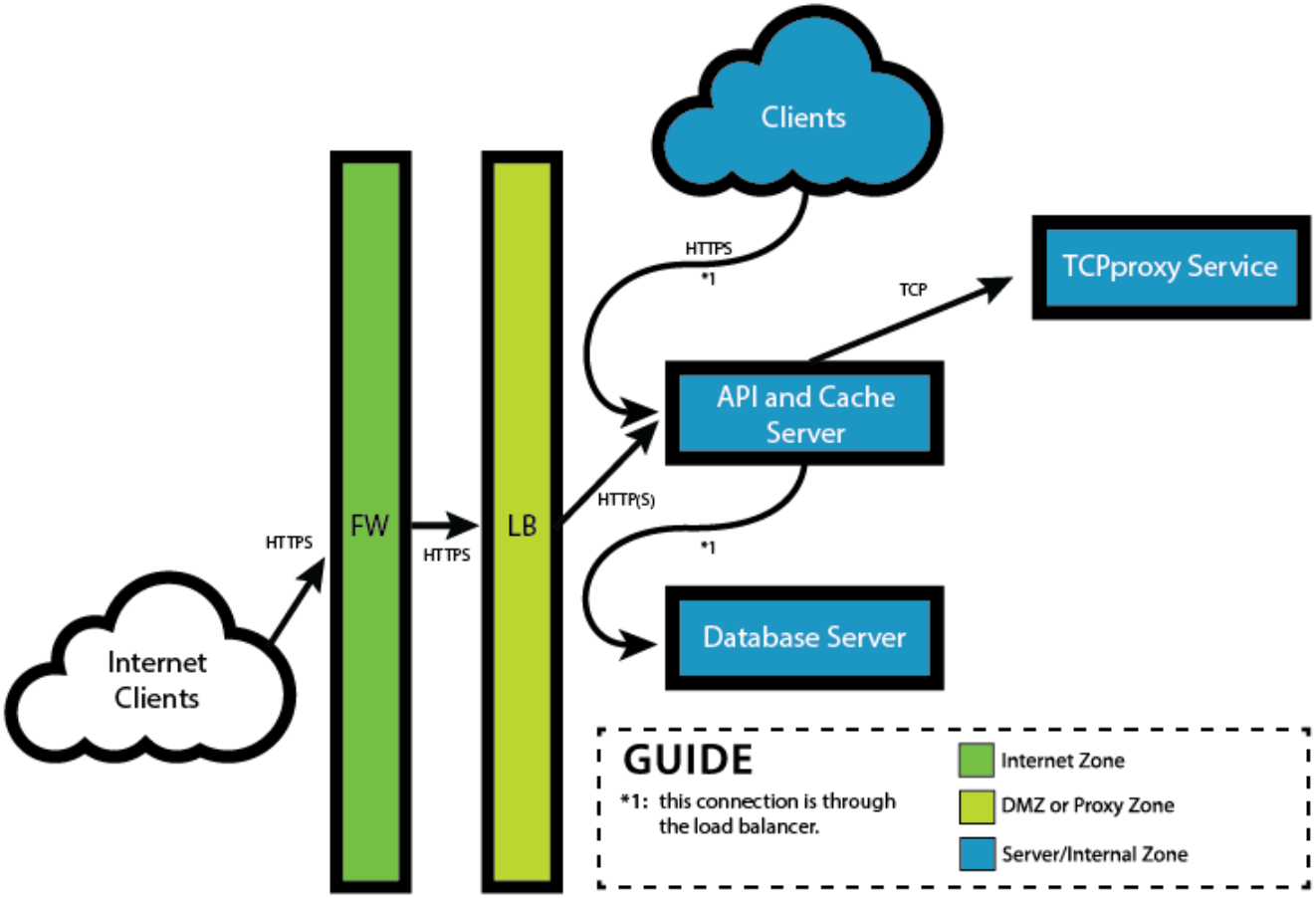
Minimum CDSM for AUC Suggested Configuration



TCP Proxy Optimum Configuration



Minimum TCP Proxy Suggested Configuration



Connectivity

Outbound Traffic

This is traffic originating from the RESTful API server(s).

Protocol	Destination	Port	Usage
TCP	Database Server(s)	Database Dependent	This is used to communicate with the database that stores the RESTful API Infrastructure configuration.
TCP	Cache Server	Configurable Default: 6379	This is used to cache resources, removing the need to talk to the back-end or database in some circumstances.
TCP	TCP Proxy Service(s)	Configurable	This is used to communicate with the TCP Proxy Service when handling API requests that require direct access to the MEDITECH Expanse database.
HTTP over TCP	One of the RESTful API Infrastructure Servers	Configurable Default: 24387	This is used to communicate with the IOPS Monitor service to obtain environmental settings. **Note:** this connection is no longer needed after IOPS is updated to v1.5.0 or above.
HTTP over TCP	MIS File Library Server(s)	Configurable Default: 24388	This is used to obtain clinical data.
HTTP over TCP	BKG Job Machines for Each HCIS	Configurable Default: 24400	This is used to speak with the MEDITECH database.
HTTPS over TCP	AUC Provider	443	This is used for the Appropriate Use Criteria for Advanced Diagnostic Imaging (AUC) MU3 requirement.[1]
HTTPS over TCP	Identity Provider	443	The Infrastructure must be able to communicate with the Identity Provider (e.g., Google) in order to validate assertions.

[1] Not all customers will use AUC. There are multiple certified decision support products such as NDSC's CareSelect.

Inbound Traffic (All Clients)

This is traffic destined for the RESTful API services from clients both on-network and off (i.e., internet traffic). This includes user desktops and other MEDITECH servers.

There are multiple possible configurations for traffic coming into the RESTful API services.

Proxy - SSL Termination

Please use this section when configuring your proxy or load balancer for SSL Termination. It is our suggestion that sites terminate SSL at their proxy/load balancer for the best performance.

Important: The VIP on the proxy should be accessible from the internet and all networks on site.

Protocol	Source	Destination	Port	Usage
HTTP over TCP	Proxy	Restful API Services	Configurable Default: 80	This will respond to API requests that originate from the proxy. Requests directly to the server or from an untrusted proxy will be issued a redirect to HTTPS.
HTTP over TCP	Proxy	Restful Application Services s)	Configurable Default: 8081	This will respond to Application requests that originate from the proxy. Requests directly to the server or from an untrusted proxy will be issued a redirect to HTTPS.
HTTPS over TCP	Anywhere	Proxy	Configurable Default: 443	This is the connection clients make to access the API and/or Application services proxied by the load balancer/proxy.

Proxy - SSL Bridging

This configuration is for organizations that do not have a trusted network between the proxy and the RESTful API servers.

Important: The VIP on the proxy should be accessible from the internet and all networks on site.

Protocol	Source	Destination	Port	Usage
HTTPS	Proxy	Restful API Services	Configurable Default: 443	This will respond to API requests.
HTTPS	Any Client	Proxy	443	This is the connection clients make to access the API services proxied by the load balancer/proxy.
HTTPS	Proxy	Restful Application Services server(s)	Configurable Default: 8888	This will respond to Application requests.
HTTPS	Any Client	Proxy	443	This is the connection clients make to access the Application services proxied by the load balancer/proxy.

Proxy - TCP Proxy

This configuration is for organizations that will be configuring real-time APIs that require direct access to the MEDITECH Expanse database. This should be configured using DNS (see DNS Entries).

Protocol	Source	Destination	Port	Usage
TCP	API Server(s)	TCP Proxy Service(s)	Configurable	This is used by the API Server(s) to handle API requests that require direct access to the MEDITECH Expanse database.

Inbound Traffic (Trusted Clients)

These are connections from MEDITECH applications running on your network. These connections MUST NOT be internet-accessible. They must share the same IP/VIP.

Protocol	Source	Destination	Port	Usage
HTTP over TCP	Proxy	Restful API server(s)	24431	Serves web pages to the MEDITECH platforms. **Note:** this connection is no longer needed after IOPS is updated to v1.4.0 or above.
HTTP over TCP	Trusted Client	Proxy	24431	This is the connection clients make to access the webpages proxied by the load balancer/proxy. **Note:** this connection is no longer needed after IOPS is updated to v1.4.0 or above.
HTTP over TCP	RESTful API Server(s)	RESTful API Server(s)	Configurable Default: 24387	This is used to communicate with the IOPS Monitor service to obtain environmental settings. **Note:** this connection is no longer needed after IOPS is updated to v1.5.0 or above.

Load Balancer/Proxy Configuration

MEDITECH cannot provide step-by-step instructions for configuring your load balancer or proxy. This is because every vendor is different and uses different names to reference the same thing.

There are two suggested configurations:

1. SSL Termination - this is when the connection to the appliance is encrypted with SSL but the connection from the appliance to the RESTful services is not encrypted.
2. SSL Bridging - this is when the connection to the appliance is encrypted with SSL and the connection from the appliance to the RESTful services is encrypted, sometimes with a different certificate and TLS configuration. SSL Bridging would indicate an End to End Encryption (E2EE).

SSL Passthrough (where the appliance operates at the TCP level instead of at the SSL level) is not a suggested configuration as it directly exposes the SSL implementation included in the RESTful services to the Internet. The danger, as outlined above, is that MEDITECH will be unable to patch and deliver fixes for SSL vulnerabilities quickly enough.

Header Insertion

In both suggested configurations, the appliance should be configured to add the "X-Forwarded-For" header to the request before proxying to the RESTful services. The header should be populated with the original client's IP. This allows the RESTful services to log the IP of the origin of the request. Without this header, all requests would appear to be from the appliance's IP.

In an SSL Termination configuration, an additional header is needed. "X-Forwarded-Proto" needs to be added to the request. It must have the static value of "https", without the quotes.

These headers allow the RESTful services to handle requests appropriately and helps in securing the connection.

You can read more about these headers on Mozilla's web-site:

- X-Forwarded-For
- X-Forwarded-Proto

IPs and Ports

A single IP, sometimes called a VIP, can be used for all environments. If using the same VIP for both API and Application environments, the load balancer will need to be configured to look at the Host header of the HTTP requests to route API requests to the API service port and Application requests to the Application service port. Our suggestion, however, is to use one VIP for all API environments and one for all Application environments.

Please note: all ports listed below may be different in your configuration due to differences in your environment and network. Please consult your TECH or Application specialist to determine if the default ports are being used.

In an SSL Bridging configuration, the appliance should proxy HTTPS/443 -> HTTPS/443 for the API VIP and HTTPS/443 to HTTPS/8888 for the Application VIP.

In an SSL Termination configuration, the appliance should proxy HTTPS/443 -> HTTP/80 for the API VIP and HTTPS/443 to HTTP/8081 for the Application VIP.

It is not suggested to configure the appliance to proxy HTTP/80 -> HTTP/80. HTTP/80 is never used by a functioning client and there is no need to allow them to connect on this port. The site may, optionally, configure their appliance to issue a redirect from HTTP/80 to HTTPS/443. If HTTP/80 is proxied, the services will issue a redirect to HTTPS/443 itself.

When configuring the TCP Proxy Service, the appliance should proxy TCP/6000 -> TCP/6000 for the TCP Proxy Service server IPs.

Important: ports used by the services in your installation may be different than the examples listed above. Please refer to the TECH task tracking the delivery of the RestAPI Infrastructure.

Health Check

The RESTful services support a standard HTTP health check:

```
HEAD / HTTP/1.1
```

If the service is functioning appropriately, it will return a "200 OK".

It is not required to configure a health check but doing so will allow the appliance to detect and react to outages better.

Trusted Proxies

The RESTful service's trusted proxies **MUST** be configured to include the load balancer's IP or IP range. Failure to do so will cause incorrect information to be recorded in the RESTful service audit logs and devices will be identified incorrectly when evaluating quotas and lockouts. This, in turn, can lead to the system becoming unusable.

Note: if there are multiple load balancers and/or proxies used, all of their IPs must be listed in Trusted Proxies.

Security

The RESTful API Infrastructure has multiple layers of security. This document aims to describe the security model.

The Flow

In order to put the layers into context, it is necessary to describe the flow of a typical application.

Authorization

1. The application redirects the user to our authorization endpoint.
2. The authorization endpoint does one of two things:
    a. Display the available authentication realms (if there is more than one), in which case it redirects the user to the identity provider after a realm is selected.
    b. Redirect the user to the only identity provider if there is a single realm.
3. The authentication completes[2] and the user is redirected back to a special endpoint on our service that processes the authentication information. This includes a step where the service (not the client) contacts the provider to validate the information, thereby mitigating a MITM attack.
4. The infrastructure evaluates whether or not the application and user combination has access to the requested scopes. It does this by confirming:
    a. All scopes in the request are valid.
    b. The user passes the ACLs for all scopes.
    c. The application passes the ACLS for all scopes.
5. An authorization code is generated and the user is redirected back to a registered endpoint belonging to the application.
6. The application makes a request to the RESTful API Infrastructure to trade the authorization code for an access token. The connection is authenticated by the application providing its OAuth2 client ID and secret.
7. The application can now make requests to APIs on the behalf of the user.

API Invocation

1. The application makes an HTTPS request to the RESTful API Infrastructure and includes the access token.
2. The infrastructure determines if the scopes granted to the token during authorization cover the API being invoked. If not, an error is returned.
3. The infrastructure then determines if there are any quota lockouts — and if there are, returns an error.

[2] What happens on the identity provider is outside the scope of this document.

4. A second level of authorization based on resource ownership is performed using the identity information tied to the token.
    a. For example, Patient A is considered to be the owner of their own patient resource. Only the patient (or others specifically granted access such as providers or healthcare proxies) can read the medical record for that patient. This does not mean the patient can do everything with their record, but only what a patient is allowed to do — in this example, obtain a copy of it.
5. The invocation — whether allowed, denied, or failed due to error — is logged in our audit table.

Layers

Firewall
The firewall protects against many threats such as DoS. In some environments, a firewall may be used to restrict access to the APIs over the internet to a set of well-defined sources.

Load Balancer
The load balancer performs multiple functions in this configuration. It returns an error if the request is malformed (i.e., not a valid HTTP request) and is used to offload SSL/TLS, allowing organizations to manage their certificates in a central location (instead of propagating certificates to multitudes of servers). It also makes setting SSL/TLS parameters, such as minimum protocol version and cipher suites, easier without the need to update or reconfigure the RESTful API Infrastructure.

One additional feature is the ability to scale out the RESTful API Infrastructure to handle more requests or to increase performance by balancing requests across the nodes in an installation.

TLS/SSL
All requests to the RESTful API Infrastructure must be over HTTPS and must be protected with a certificate from a trusted authority (not a self-signed certificate).

Application Registration
In order for an application to be able to obtain a token, they must first be added to the configuration by an administrator. This generates a client ID and client secret, which is then used by that application when obtaining tokens. The ID and secret do NOT grant access to APIs and are only used during authorization. Please consult the Interoperability group to onboard applications.

Important: The client secret should only be transmitted and stored securely. **DO NOT** send it via email if at all possible.

User Login
MEDITECH requires the use of an Identity Provider (IdP) to authenticate patients (or their representatives) and providers. The IdP your organization deploys must support the OpenID Connect or SAML 2.0 protocol. Alternatively, support for using your MEDITECH Patient Portal as an IdP has been added - please consult your Interoperability specialist to determine if you meet the minimum software requirements for this option. Before selecting an Other Vendor IdP for your organization, please work with MEDITECH so we can confirm its compatibility with RESTful API Infrastructure.

Note: The most immediate use case for the RESTful API Infrastructure and IOPS is to meet MU3 requirements, which will only use the IdP to authenticate patients. Future use cases will include allowing providers to use applications to access data. At that time, an IdP will need to be able to authenticate providers. This may mean a single IdP or multiple IdPs depending on your needs.

OAuth2
The OAuth2 standard is used to secure the APIs from unauthorized access.

Scope ACLs
Scopes have their own set of ACLs which prevent applications or users from obtaining a token with those scopes. Some scopes have a default ACL that denies token generation (i.e., whitelisting), while most other scopes have a default ACL that allows it (i.e., blacklisting). Which method MEDITECH chooses for a scope depends on the intended usage of the scope.

The evaluation of whether or not a token may be generated with the requested scopes is done during the Authorization stage after user authentication. This is because ACLs can also be used to deny/allow based on the authenticated user in addition to the authenticated application.

The Admin Panel can be used to view and configure these ACLs.

Quotas
Quotas allow a site to limit the number of requests made within a period of time. This helps alleviate strain on the MEDITECH database if an application is misbehaving. It also acts as another layer of security by temporarily locking out an application or user from accessing APIs.

Auditing
All requests to the RESTful API Infrastructure are audited. This enables identifying what data was accessed by bad actors, should a breach of a user's or application's credentials occur. It could also be used by analytics tools to report on suspicious activity. We are still developing the tools that can be used to review the audit entries.

Terms

Application Server/Service
The RestAPI Infrastructure listens on a second port which allows it to easily serve up integrated applications. These are web applications created by MEDITECH's development teams and includes the Admin Panel, which is used to configure the infrastructure.

API Server/Service
The primary purpose of the RestAPI Infrastructure is to expose APIs to trusted/authenticated clients.

This is also typically used to reference the physical server(s) or VM(s) on which the RestAPI Infrastructure is running.

ACL
Access Control List: A mechanism of defining who and what can access which resources.

Blacklisting
A way of granting access where access is allowed, unless an exception is made. To put it into an analogy: "Anyone may use the coffee pot except children under 12." This is the opposite of whitelisting.

Client ID
This is the unique identifier given to an OAuth Client, otherwise known as an application or service. It can be thought of as the username for an application.

Client Secret
This is a secret value given to an OAuth Client. When paired with the Client ID, it allows an application to authenticate as part of the authorization flow.

DoS
Denial of Service: An attack against a server that leads to clients being unable to connect or receive responses due to exhausting the resources of the server (CPU, IO throughput) or network.

End to End Encryption (E2EE)
Would indicate a secure connection between both endpoints of the communication, normally using an asymmetric encryption protocol. It is a way to prevent a man in the middle attack.

Environment
Environments come in two flavors:

1. API - this defines the hostname that links to a specific MRI or HIM database
2. Application - this defines the hostname that links an application to a specific API environment, thereby allowing that application to query the specific MRI or HIM database

Environments are one-for-one with the DNS entries defined by the site and allow the RestAPI Infrastructure to differentiate between API traffic (and which MRI/HIM database that API should resolve to) and Application traffic (and which API environment it uses).

MITM
A man-in-the-middle attack is one where a client's requests are sent to an insecure server that acts as a proxy to the real server.

Quota
A quota is a way of limiting the number of requests that can be made within a period of time. Quotas should be based on real-world application workflows.

Resource
A resource is any file or uniquely identifiable data, such as a patient record or an order.

Scope
A scope defines a set of related APIs.

TCP Proxy Service
A TCP Proxy Service is connected to by an API Service when processing API whose data originates in the MEDITECH Expanse Platform. This service is responsible for maintaining a pool of M-AT Magic Console processes and proxying TCP connections.

Token
A token is a string used to provide identity without exposing usernames and passwords to applications.

Whitelisting
A way of granting access where access is denied, unless an exception is made. To put it into an analogy: "No one may access the server room except network engineering." This is the opposite of blacklisting.

Installation Validation Guide

This validation guide is meant to assist both the IT staff at site and MEDITECH staff identify connectivity and/or configuration issues.

Important Note

"**HOSTNAME**" (referenced below) is the DNS record the site has configured to be used with the RESTful API and IOPS services. This is not the name of a server. Please refer to the configuration task for information on the hostnames — there may be many.

Check DNS

On Network

1. Log on to any device on network.
2. Launch the command prompt.
3. Execute "nslookup HOSTNAME." (See Important Note above.)
4. If an error is reported, internal DNS is not configured properly and must be resolved by site's network engineers.
5. Confirm the HOSTNAME resolves to the VIP on the load balancer. If it does not, DNS is not configured properly and must be resolved by site's network engineers.
6. Repeat these steps for all hostnames/environments.

Off Site's Network

1. Load this link in your browser: https://mxtoolbox.com/DNSLookup.aspx.
2. Enter in the HOSTNAME as you did for step #3 above, and click the orange "DNS Lookup" button.
3. If an error is reported, external DNS is not configured properly and must be resolved by site's network engineers.
4. Confirm the IP is NAT'd to the VIP on the load balancer. If the resolved address is incorrect, DNS is not configured properly and must be resolved by site's network engineers.
5. Repeat these steps for all hostnames/environments.

Check Services

1. Log on to the API server.
2. In the Start menu, select Control Panel.
3. From the Control Panel window, select Administrative Tools.
4. Launch the "services" and verify the following services are installed and in a "Running" status. *Note: The "(xxx)" below in each service name is a unique identifier for the installation.*
    a. MEDITECH IOPS Monitor (xxx)
        i. **Note:** this connection is no longer needed after IOPS is updated to v1.5.0 or above.
    b. MEDITECH IOPS MSF (xxx)
    c. MEDITECH Rest API Node Service (xxx)
    d. MEDITECH Rest API Redis Service (xxx)

    "MEDITECH IOPS Monitor (xxx)" will only be deployed on one of the API servers.
5. Open the browser.
6. Navigate to http://localhost.
7. If the browser reports "Connection Reset," "Connection Aborted," or "Connection Timed out," then the RESTful API services are not working correctly.
8. If IOPS services version is < 1.4.0 (ex. 1.3.0):
    a. Navigate to http://localhost:24431.
    b. If the browser reports "Connection Reset," "Connection Aborted," or "Connection Timed out," then the IOPS services are not working correctly.
9. If there are multiple API servers, repeat steps 1-9 for each one.
10. Log on to the server with the IOPS File-Explorer service installed.
11. In the Start menu, select Control Panel.
12. From the Control Panel window, select Administrative Tools.

13. Launch the "services" and verify the "MEDITECH IOPS Explorer" service is installed and in a "Running" status.

Check Connectivity

Many of the steps below use the browser to confirm connectivity. Due to differences in how browsers display the data returned by the RESTful API Infrastructure, you may not actually see the page indicated in the instructions. We suggest using Chrome or Postman to verify connectivity as both of these display the Infrastructure's error messages instead of displaying their own.

On Network

RESTful API Services

1. From any device on-network, open the browser.
2. Navigate to https://HOSTNAME.
3. If HOSTNAME is for an API environment, you should see this displayed in the page:
   `{"resource":"v1/resource/error/_version/1/","detail":"Not Found"}` or
   `{"resource":"v1/resource/error/_version/1/","detail":"could not route request"}`
   i. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall and load balancer logs.
   ii. IE will not display the response. Instead, it will display a generic error page that the resource was not found or there was a bad request. You can confirm the content of the response by opening the developer tools, navigating to the Network tab and reviewing the Response body for the request.
4. If HOSTNAME is for Application environments, you should either: be redirected to the https://HOSTNAME/home/ page and shown a list of available applications, or, if defined for the specific environment, defaulted into the application defined.
   i. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall and load balancer logs.

AUC Viewer & CCDA Validation Report (required for v1.3.x and less IOPS installations)

1. From any device on-network, open the browser.
2. Navigate to http://HOSTNAME:24431.
3. If the browser reports anything but a 404, then connectivity is blocked from that device, and the site will need to review their firewall and load balancer logs.

Multi-Service Framework (MSF) Service

1. From the ISWEB server, open the browser.
2. Navigate to http://{monitorHost:port} where "monitorHost" is the FQDN of the server and "port" is the port that the Monitor service is listening on.
   a. This is required for v1.4.x and less IOPS installations only. For v1.5.x and above skip to step 5.
3. You should see this displayed in the page:
   `{"connection": "success"}`
4. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall logs.
5. Navigate to http://{explorerHost:port} where "explorerHost" is the FQDN of the server and "port" is the port that the File-Explorer service is listening on.
6. You should see this displayed in the page:
   `{"connection": "success"}`
7. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall logs.
8. Repeat steps 5-7 to all File-Explorer services.
9. Navigate to http://{platformHost:port} where "platformHost" is the FQDN of the server and "port" is the port that the Platform service is listening on.

10. If the browser reports anything but a 403 or 404, then connectivity is blocked from that device, and the site will need to review their firewall logs.
11. Repeat steps 9-10 to all Platform services.

12. Repeat steps 1-11 on all ISWEB servers.

File-Explorer Service (required for v1.4.x and less IOPS installations)

1. From the server with File-Explorer Service installed, open the browser.
2. Navigate to http://{monitorHost:port} where "monitorHost" is the FQDN of the server and "port" is the port that the Monitor service is listening on.
3. You should see this displayed in the page:
   `{"connection": "success"}`
4. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall logs.
5. Repeat steps 1-4 on all servers with the File-Explorer service installed.

Off Network

RESTful API Services

1. From any device off-network, open the browser.
2. Navigate to https://HOSTNAME.
3. If HOSTNAME is for an API environment, you should see this displayed in the page:
   `{"resource":"v1/resource/error/_version/1/","detail":"Bad Request"}`
   i. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall and load balancer logs.
4. If HOSTNAME is for Application environments, you should either: be redirected to the https://HOSTNAME/home/ page and shown a list of available applications, or, if defined for the specific environment, defaulted into the application defined.
   i. If you do not, then connectivity is blocked from that device, and the site will need to review their firewall and load balancer logs.
   ii. If the Applications should only be available on network, you should not be able to connect (the browser should show "Connection Timed Out" or "Connection Refused").

AUC Viewer & CCDA Validation Report (required for v1.3.x and less IOPS installations)

1. From any device off-network, open the browser.
2. Navigate to http://HOSTNAME:24431.
3. The browser should report that the connection could not be made, failed, or was rejected. If you get a response from the server, like a 404, then the application has been exposed to the internet. The site must modify their firewall to disallow access on port 24431 from the Internet.

Change Log

August 3, 2021

- Updated Database section to note PostgreSQL as the only database that should be used to deploy new instances of the infrastructure. Explicit support for MariaDB, MySQL, and Microsoft SQL Server has been removed from this document, implicit support for those databases will continue until supported deployments have been migrated.
- Updated DNS Entries section to remove text that implies the application service should not be accessible over the internet.

March 10, 2021

- Updated DNS Entries section for TCP Proxy setup to replace mention of load balancer VIP with TCP Proxy IPs.
- Updated various other references to TCP Proxy setup with the updated DNS guidance.

July 10, 2020

- Added link to Security documentation for Redis configuration.
- Added more information to "SSL Bridging" definition.

May 26, 2020

- Added note on configuring trusted proxies in Load Balancer/Proxy Configuration.

January 10, 2020

- Replace NDSC vendor references with Clinical Decision Support Mechanism (CDSM) for appropriate use criteria (AUC).
- Updated and added configuration diagrams.
- Added documentation regarding TCP Proxy Service.

July 9, 2019

- Clarified documentation by changing "Database" to "Database Server".

May 8, 2019

- Added a note on updating max_allowed_packet for older MariaDB/MySQL versions.

January 31, 2019

- Updated Security Layers User Login section to indicate support for SAML 2.0 Idp and MEDITECH Patient Portal Idp.

December 10, 2018

- Updated Validation Guide section to indicate change in API server (≥1.5.0) response when navigating to the root URL from a browser.

December 7, 2018

- Updated Database section to indicate incompatibility with MySQL 8.0.4 or above.

October 19, 2018

- Updated ports under "Load Balancer/Proxy Configuration" to be in line with connectivity tables earlier in this document.

August 20, 2018

- Updated connectivity tables for v1.5.0 release of Interoperability Services.
- Removed "PostgreSQL" from documentation.