GO-GLOBAL®

Deployment Guide

6.4.2

COPYRIGHT AND TRADEMARK NOTICE

CONTACT INFORMATION

GraphOn Corporation
189 North Main Street, Suite 102
Concord, NH 03301 USA
sales@graphon.com
graphon.com

CONTENTS

Chapter 1: How GO-Global Works 1

A Typical GO-Global Farm 1

Running GO-Global 5

OpenID Connect Authentication 10

AppController, GO-Global's Native Client 8

Windows Authentication 11

Starting Sessions 13

Managing Sessions and Settings 15

Chapter 2: GO-Global Deployments 17

Perimeter Firewall 18

Web Application Firewall (WAF) 19

Reverse Proxy 20

Web Server 21

Network Load Balancer 22

Back-End Firewall 23

Domain Controller 23

Administration Firewall 23

Identity Provider 24

Auto-Scaling Infrastructure 25

Availability Zones and Regions 26

Monitoring and Observability Tools 26

Appendix 28

Definitions 28

RDS & GO-Global: A Simplified Translation Guide 30

GO-Global Architecture Type 1 (Recommended) 30

GO-Global Architecture Type 2 32

GO-Global Architecture Type 3 33

Port Requirements 34

Independent Host Deployment Checklist 36

CHAPTER I

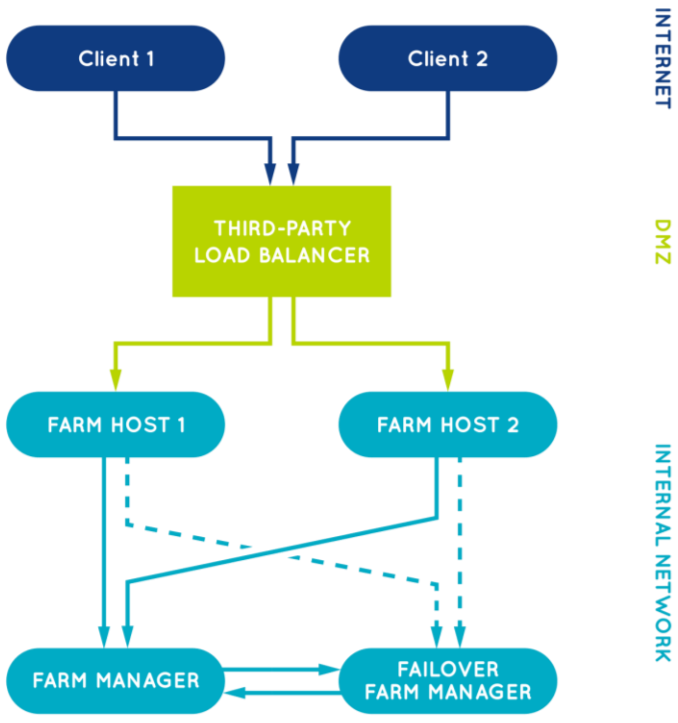
How GO-Global Works

1

The following section describes how GO-Global works and introduces the concepts and terminology that administrators need to understand to plan large-scale GO-Global deployments. It contains a mixture of high-level architectural information and low-level details. The goal of this section is to help administrators avoid many of the common stumbling blocks and pitfalls that occur when customers deploy GO-Global for the first time.

A Typical GO-Global Farm

GO-Global is a flexible product that supports a variety of configurations and server roles. Most large-scale deployments, however, use GO-Global's Farm Host and Farm Manager server roles. The following diagram illustrates the configuration of a typical GO-Global Farm.



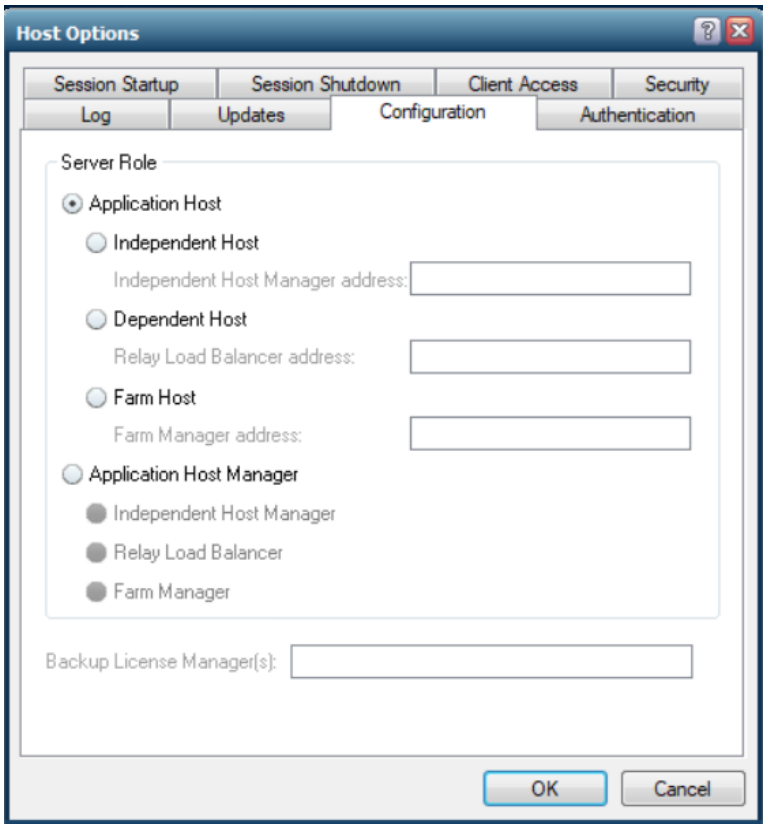
[Diagram 1: Typical GO-Global Farm Configuration]

A **Farm Host** is a GO-Global Host on which published applications run. The Farm Host server role enables GO-Global functionality similar to what the RDS Session Host, RDS Web Access, and RDS Web Client roles provide collectively in Remote Desktop Services (RDS).

On a Farm Host, GO-Global's **Application Publishing Service (APS)** listens for connections on a single port (491, by default). The APS can accept connections from browsers (HTTP/HTTPS), the GO-Global Web App (WS/WSS), and GO-Global's native clients and hosts (RXP/RXPS) on this one port.

When the APS starts on a Farm Host, it attempts to connect to its primary **Farm Manager**. If a Farm Host is unable to connect to its primary Farm Manager, it then attempts to connect to its failover **Farm Manager**. The primary Farm Manager and failover Farm Manager have an Active/Passive relationship.

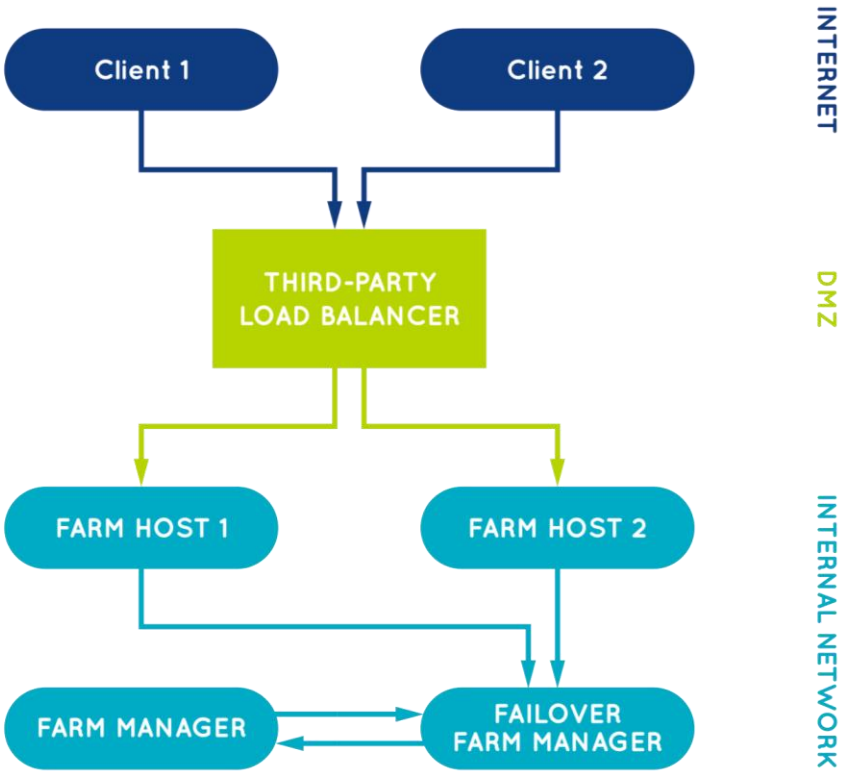
The addresses of the primary and failover Farm Managers are entered in a semi colon-delimited list in the **Farm Manager address** field of the Admin Console's Configuration tab, on each Farm Host. When the TLS protocol is enabled on the Farm Managers, the addresses entered must meet the requirements of the Farm Managers' TLS certificates.



[**Configuration** tab of the **Host Options** dialog]

Farm Hosts connect to Farm Managers using the same port on which they accept connections (port 491, by default).

The diagram below illustrates how clients connect to a GO-Global Farm.



[Diagram 2: Connecting to a GO-Global Farm]

The Farm Host server role enables GO-Global functionality similar to what the combined RDS Session Host, RDS Web Access, and RDS Web Client roles provide in Remote Desktop Services (RDS). The main exception is the GO-Global Farm Manager does not, by itself, function as a load balancer.

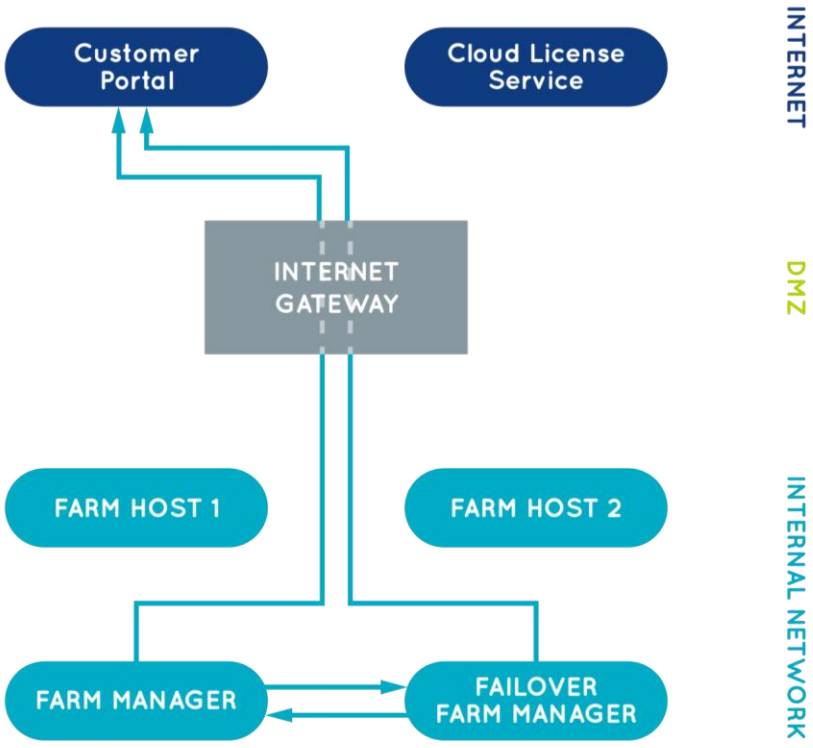
For large-scale deployments, GO-Global relies on third-party load balancers to balance client connections between Farm Hosts. At times, Farm Managers will direct hosts to relay connections between Farm Hosts, but clients do not connect to Farm Managers.

Optionally, Farm Managers can be separated from Farm Hosts by a firewall. In this case, the firewall must only allow connections from the Farm Hosts to the Farm Managers on the specified port. The firewall does not need to allow connections from the Farm Managers to Farm Hosts. Farm Managers *must*, however, be allowed to connect to **portal.graphon.com** and **cloud.graphon.com** on port 443, so they can manage GO-Global licenses.

To activate GO-Global, administrators run either the **GO-Global Activation Wizard** or the **Invoke-GGActivate** PowerShell command on Farm Managers. These utilities connect to the **GraphOn Customer Portal** (portal.graphon.com) on port 443 and configure the Farm Manager to use a specified license.

After a Farm Manager is activated, the APS on the Farm Manager connects to the **Cloud License Service** (cloud.graphon.com) on port 443 and sends a request to the service to reserve license seats. It requests the number of seats specified by the **Maximum sessions on this host** option. The Cloud License Service negotiates this request with any other Farm Managers that are using the license and returns a seat reservation that is less than or equal to the seats requested.

The following diagram shows the connections that the Farm Managers must make to activate a Cloud License.

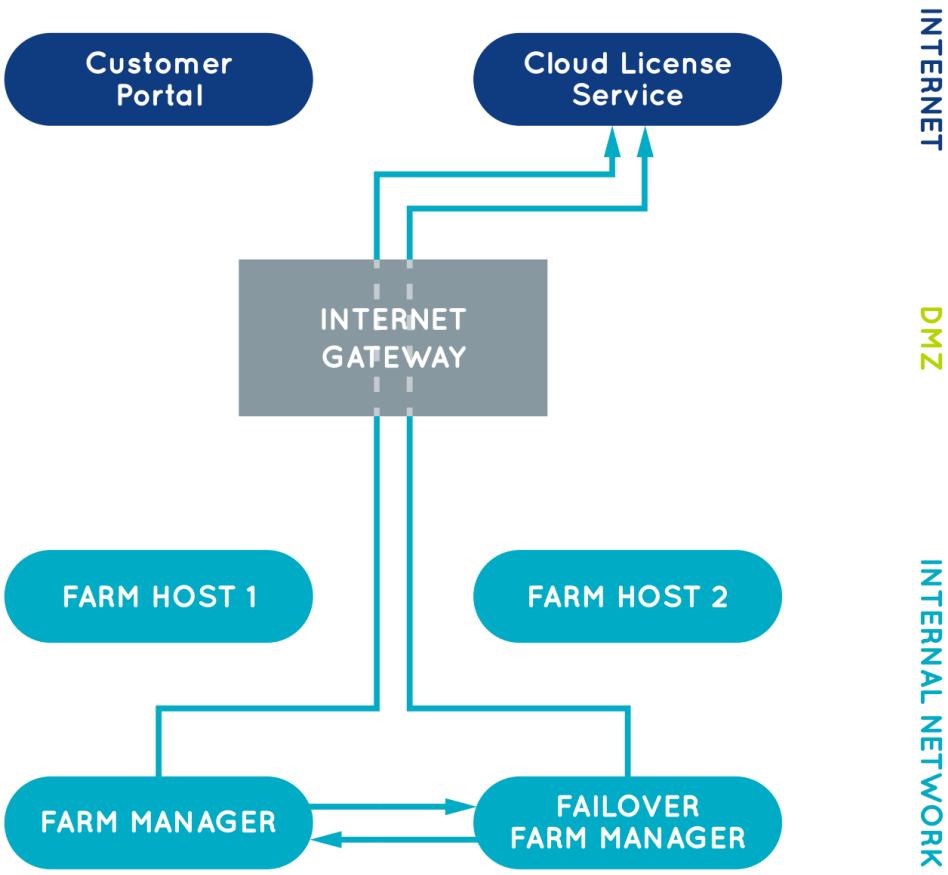


[Diagram 3: Activating a Cloud License]

Both the primary and failover Farm Managers are activated using the same license, and they are configured to be **Backup License Managers** of each other. This configuration enables the Farm Managers to operate offline and serve licenses to their Farm Hosts for up to 72 hours without being able to communicate with the Cloud License Service.

Farm Hosts do not need to be activated. Farm Hosts get their licenses from their Farm Managers. This enables Farm Hosts to be created from “gold” images and simplifies the process of adding and removing Farm Hosts from the farm as the load on the farm increases and decreases.

The diagram below shows the connections the Farm Managers make to the Cloud License Service after they have been activated.



[Diagram 4: Connecting to the Cloud License Service]

Running GO-Global

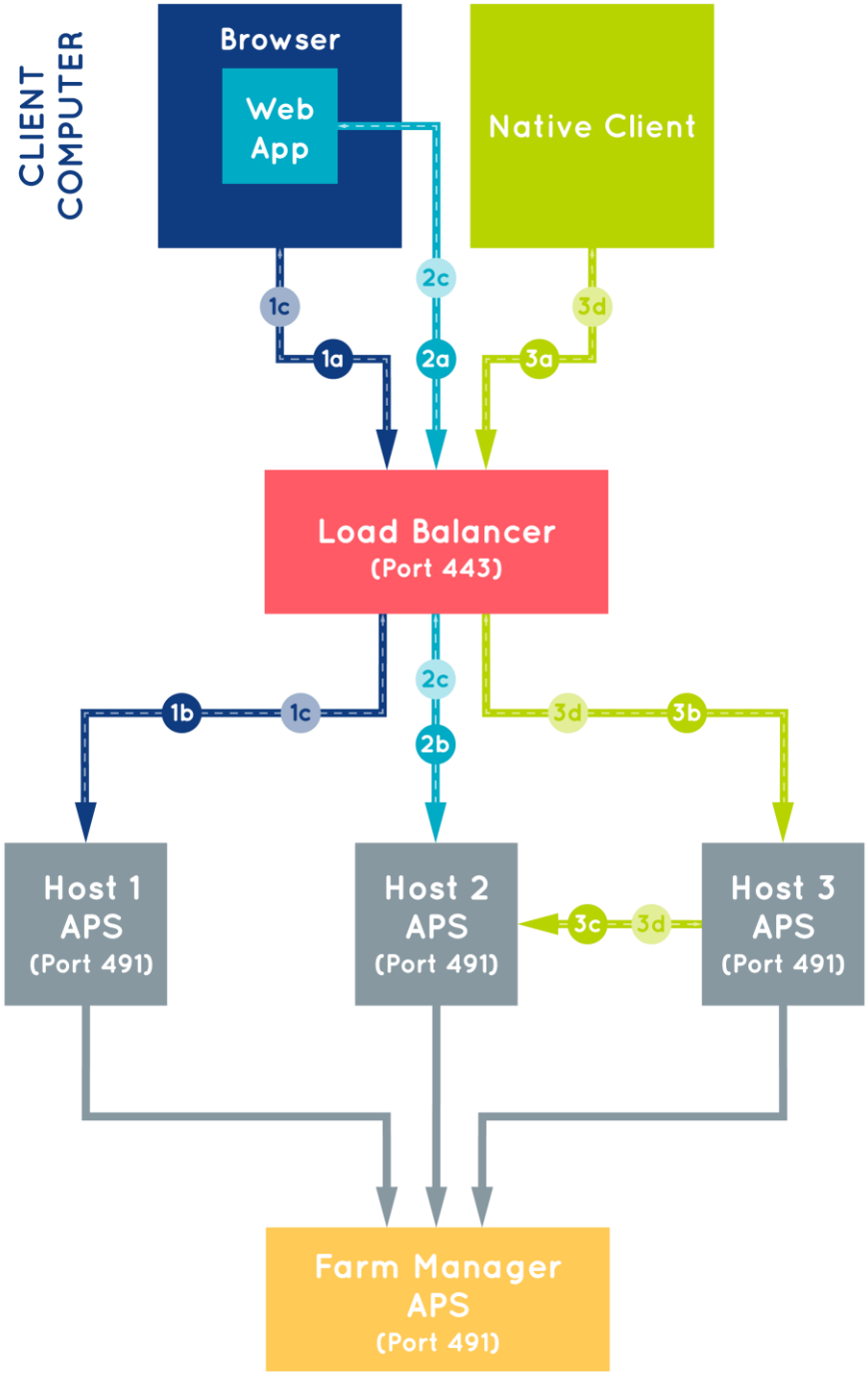
In most large-scale deployments, users start GO-Global from a browser. The reasons for this include:

1. GO-Global's browser interface helps users install and run GO-Global's native client, AppController
2. Administrators can configure GO-Global's client options in URLs and/or HTML pages.
3. GO-Global's browser interface supports OpenID Connect Authentication (SSO).

When users connect from a browser, there are generally three clients that must connect before a session can start:

1. The browser
2. The GO-Global Web App (GO-Global's HTML5 client)
3. AppController (GO-Global's native client)

The following diagram illustrates how these connections are established in a typical large-scale deployment.

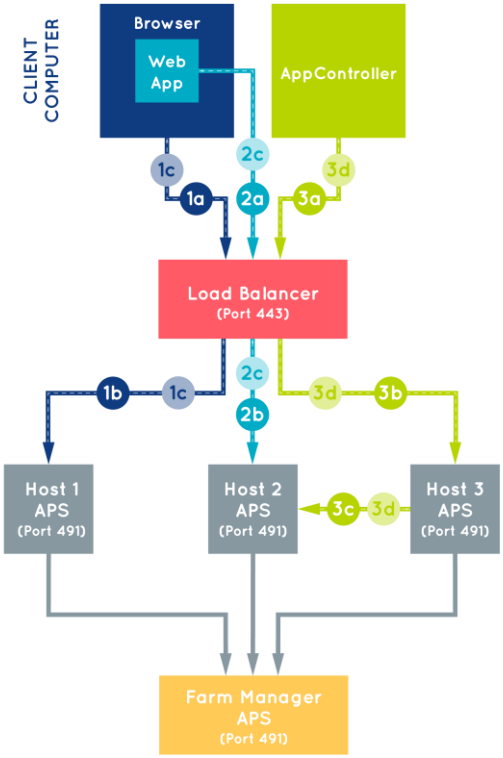


[Diagram 5: Connections in a Typical Large-Scale Deployment]

When a user browses to the endpoint of a service that is hosted using GO-Global, the browser's connection (**1a**) is usually accepted by a third-party load balancer. The load balancer may be a simple network load balancer, a (web) application load balancer, or a reverse proxy. In some cases, the connection will be accepted by an internet gateway (e.g., a web application firewall) and then forwarded on to a load balancer.

After accepting the connection (and optionally terminating the TLS), the load balancer opens a connection (**1b**) to a web server. The web server can be GO-Global's embedded web server (**Host 1**), or it can be a third-party web server (e.g., Apache or IIS). After the browser's connection is established through to the web server, the browser downloads the GO-Global Web App and its associated files (**1c**) and then starts the Web App.

When the Web App starts, it reads its configuration options from the URL and from parameters specified in the HTML page that loaded it (e.g., logon.html). The Web App then opens a WebSocket connection to the load balancer (**2a**). The load balancer then opens a connection (**2b**) to the Application Publishing Service (APS) on a GO-Global Host (**Host 2**).

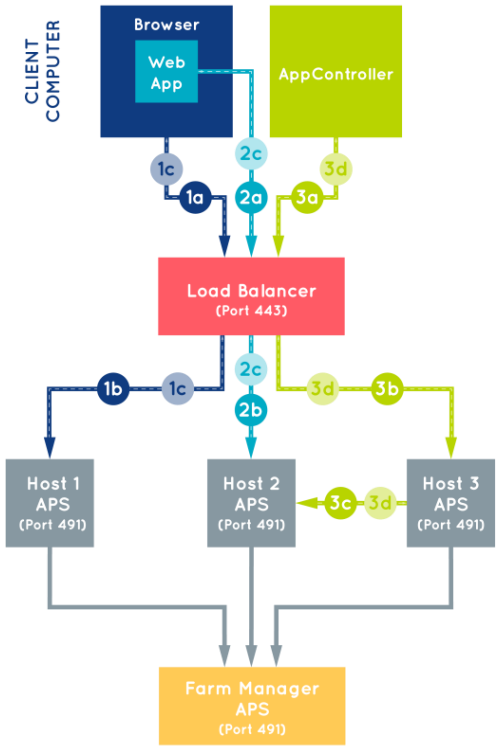


AppController, GO-Global's Native Client

After the user is authenticated, the APS may optionally start the user's session and display the user's application within the browser's window using the GO-Global Web App. When the session is run via the GO-Global Web App, however, users are unable to directly access client devices (e.g., printers and files) due to browser security restrictions. For this reason, GO-Global starts its native client, AppController, by default.

When configured to use AppController, the APS orchestrates the installation (if required) and launch of AppController via the Web App. When AppController starts, it reads its configuration options from the command line specified by the GO-Global Web App and opens a connection to the load balancer (**3a**). The load balancer then opens a connection to the APS on one of the hosts. (**3b**).

Connections **3a** and **3b** are typically established via the same service endpoint and load balancer that established the HTTP/HTTPS and WebSocket connections. In version 6.3 and earlier, however, AppController does not support WebSocket connections. Therefore, the load balancer that handles connections from AppController must be a TCP network load balancer. It cannot be a load balancer that inspects packet headers and routes traffic accordingly (e.g., a web application load balancer or reverse proxy).



In most cases, the load balancer routes connection **3b** to a different host (**Host 3**) than the host that initiated the launch of AppController (**Host 2**). When this occurs, the APS of the host that receives AppController's connection (**Host 3**) opens a connection (**3c**) to the host that initiated AppController's launch (**Host 2**). It then relays the session's data (**3d**) between the host and AppController. This is done so GO-Global can authenticate AppController's connection and properly display session startup status to the user in the browser's window.

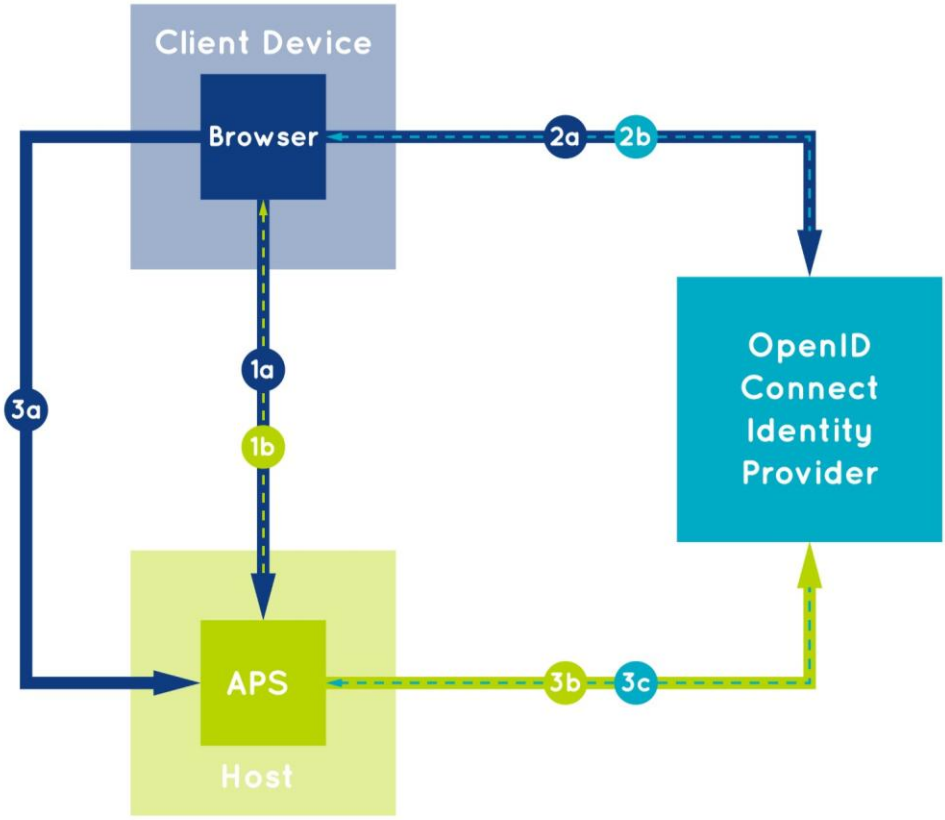
In the example, when **Host 3** opens connection **3c** to **Host 2**, it obtains the internal address of **Host 2** from data that **Host 2** sent to the Web App with its AppController launch request (**2c**). **Host 2** obtains its internal address from the value of the **RelayConnectionAddress** property in its HostProperties.xml file.

By default, the value of the RelayConnectionAddress property is set to the computer's NetBIOS address. When TLS connections are terminated at the hosts, however, administrators must change the value of the **RelayConnectionAddress** property to an FQDN address that matches the requirements of the host's TLS certificate. Otherwise, AppController will frequently fail to connect.

OpenID Connect Authentication

When GO-Global hosts are accessed over the internet, OpenID Connect (OIDC) Authentication is recommended because it helps prevent DDOS attacks by ensuring that only authenticated users are allowed to start sessions. In addition to this benefit, OIDC identity providers generally support a wide range of secure authentication options (e.g., two-factor authentication), and they enable users to reset their passwords, alleviating service-providers from this administrative burden.

When OIDC Authentication is enabled, the APS redirects the browser to an OpenID Connect identity provider after the WebSocket connection is established. Then, after the user is successfully authenticated, the identity provider redirects the browser back to the GO-Global service using the specified callback URL, passing it an authorization code.



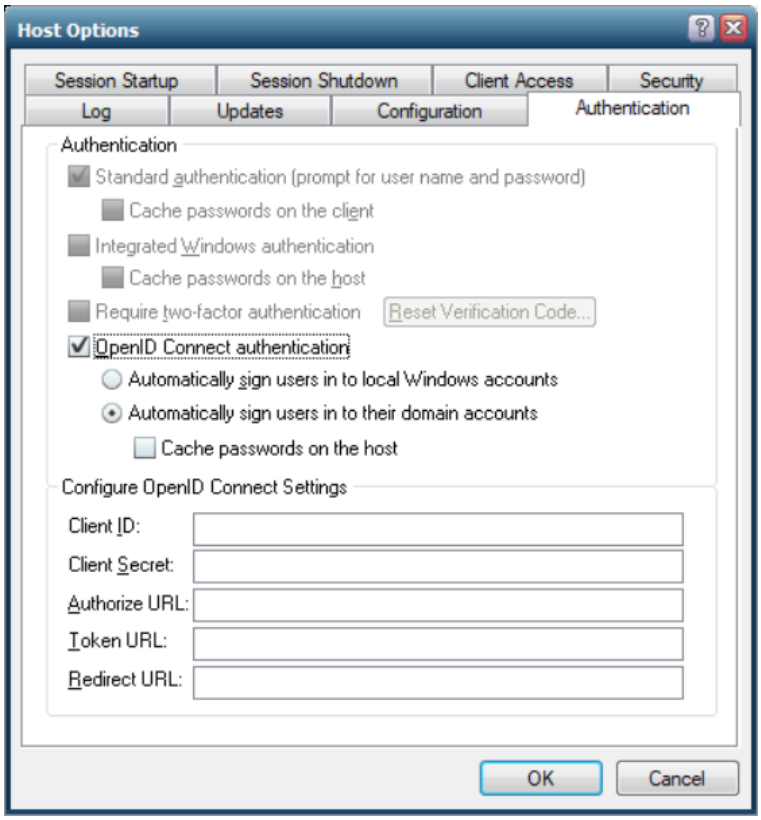
[Diagram 6: APS redirects the browser to an OpenID Connect identity provider]

When the browser receives the callback URL, it repeats steps 1 and 2 (reloads the Web App and opens a new WebSocket connection to the APS on a GO-Global Host.) After connecting, the Web App sends the authorization code, which it obtained from the OIDC identity provider, to the APS. The APS then sends a token request to the identity provider to validate the authorization code. If valid, the identity provider returns OIDC identity and access tokens to the APS.

Windows Authentication

To run Windows applications, GO-Global must authenticate the user on Windows. The tokens obtained during OIDC authentication cannot be used directly to authenticate the user on Windows. Therefore, when OIDC authentication is used, GO-Global provides two options for authenticating users on Windows:

- Automatically sign users in to local Windows accounts
- Automatically sign users in to their domain accounts



[**Authentication** tab of the **Host Options** dialog]

Administrators generally enable GO-Global's **Automatically sign users in to local Windows accounts** option when none of the published applications need to use Windows authentication to access resources on the host's network. When this option is enabled, GO-Global automatically creates a local user account on the host and signs the user in to that account. GO-Global derives the name of this account from the UPN obtained during OIDC authentication.

Alternatively, administrators generally enable GO-Global's **Automatically sign users in to their domain accounts** option for centralized user management, or when one or more published applications need to use Windows (i.e., Kerberos) authentication to access network resources (e.g., to run domain group policy, access file shares, authenticate to a database, etc.). When this option is enabled, GO-Global performs an S4U authentication using the UPN obtained during OIDC authentication.

GO-Global provides several options for deriving a user's UPN from claims in an OIDC ID token. By default, GO-Global searches for a valid UPN in the **email**, **upn**, **sub**, and **userid** fields of the ID token, in that order. Alternatively, administrators can specify the claim that contains the UPN via the **OpenIDConnectUserNameField** property in HostProperties.xml.

When the user's UPN is not available in the ID token (e.g., when the end customer's identity provider is used), administrators can configure GO-Global to look up the user's Active Directory account (and UPN) via the user's email address by setting the value of the **OpenIDConnectUserLookupByEmail** property HostProperties.xml to true. When this property is set to true, GO-Global searches the Active Directory of the Farm Host for an account with an email attribute that matches the OIDC ID token's email claim.

Unlike standard username and password authentication, S4U authentication allows privileged processes such as the APS to authenticate users without a password. Because of this, tokens produced by S4U authentications do not have the same rights as tokens produced from username and password authentications. Specifically, tokens from S4U authentications are not allowed to access network resources by default.

To enable applications running in OIDC/S4U-authenticated sessions to access network resources, administrators must grant GO-Global Hosts the right to access the specific network services that are required. This is done by configuring Delegation options in Windows. For example, to enable GO-Global to apply domain Group Policy, administrators must configure Delegation so computers running the GO-Global Host are allowed to access the LDAP and CIFS services on the domain controllers.

And when a published application needs to authenticate to network services such as SQL Server, administrators must:

a. configure the application to use Kerberos authentication
b. enable GO-Global's Kerberos extension for the application
c. configure Delegation to allow access to the desired service

Farm Managers do not require delegation to be configured.

Starting Sessions

After GO-Global authenticates the user on Windows, it sends a request to the host's Farm Manager to determine if the user has a disconnected session running on any of the farm's hosts or, if the **FollowMe** option is enabled, if the user has a connected session running on any of the farm's hosts. If such a session is found on the current host, GO-Global connects the client(s) to the session. Alternatively, if such a session is found running on a different host, GO-Global directs the client(s) to disconnect from the current host and reconnect to the host running the session. GO-Global does this to ensure that there is, at most, one "relay" connection (**3c**) between hosts for any given session.

If no such session is found, the APS creates a new session for the user. To do this, it:

a. sends a request to the Farm Manager to create the required session data structures
b. calls to GO-Global's System Extensions Driver to create the session's kernel-mode data structures and private address space
c. starts two core session processes: the Windows' Client Server Runtime Subsystem process (csrss.exe) and GO-Global's logon.exe process

Logon.exe is GO-Global's replacement for Windows' winlogon.exe process. When logon.exe starts, it helps complete the initialization of the session's kernel-mode components. Then, if the user has not already authenticated (e.g., via OpenID Connect), it prompts the user for credentials. Then, after the user is authenticated, it performs higher-level session-initialization tasks such as checking out a GO-Global license for the user, applying Group Policy, running logon scripts, and starting and configuring GO-Global features. This includes starting the remoteclip.exe process if client-side clipboard access is enabled, and configuring client printers. Finally, logon.exe starts the GO-Global Program Window (pw.exe).

When access to client printers is enabled, GO-Global starts configuring client printers after the GO-Global license is checked out. By default, printer configuration occurs in parallel while other session initialization tasks are executing, and printer configuration will often not complete until after the Program Window and other applications in the session are started. As a result, if a user immediately tries to print after signing in, not all the printers may be listed in the Print dialog.

In version 6.3.3 and later, administrators can configure GO-Global to wait to perform session initialization tasks and start applications until printers are configured. Administrators can do this by changing the value of the **PrinterConfigWaitTime** property in HostProperties.xml from 0 to the maximum number of seconds GO-Global should wait for printers to be configured. With this change, however, users will have to wait longer to be able to start applications in their sessions.

When the Program Windows starts, it optionally runs some processes to finish initializing the session and the user's environment. For example, if keyboard layout features are required, it starts the Windows' Collaborative Translation Framework Loader process, ctfmon.exe. And if GO-Global's **InitializeProfileWithExplorer** option in HostProperties.xml is enabled (the default) and the user has not previously started a session on the Farm Host, the Program Window starts Windows' explorer.exe in the background to initialize the user's User Profile.

When **explorer.exe** is run by a user for the first time on a computer, it performs initialization tasks and runs any "first run" programs that are registered. For example, it often runs Microsoft Edge so it can initialize the user's environment. It often takes explorer.exe and the programs it starts a long time to complete their initializations. This can significantly increase the time it takes GO-Global to start a published application. This delay only occurs the first time a user connects to a given Farm Host, but there is nothing in GO-Global to direct a given user's connection to the same Farm Host each time the user connects. Users will generally be connected to random Farm Hosts. As such, if there are a large number of Farm Hosts, users will experience these delays frequently.

If the initializations that explorer.exe and its child processes perform are not required by the published applications, administrators can disable this “first run” functionality and significantly reduce the start-up time of published applications by changing the value of the **InitializeProfileWithExplorer** property in HostProperites.xml to false.

After performing these tasks, the Program Window checks its command line to see if a specific application was specified to be launched via the **app** URL parameter or the **app** variable in the logon.html page. If an application was specified, the Program Window starts the specified application and then closes without displaying any of its UI. Alternatively, if no application was specified, the Program Window displays its UI and shows the icons of the applications that have been published to the user. The user can then start any of the published applications.

Managing Sessions and Settings

GO-Global’s primary tool for publishing applications and administering GO-Global sessions and settings is the **Admin Console**. The Admin Console is a Windows application that is installed on every GO-Global Host (e.g., on every Farm Host and Farm Manager). In addition to being a GUI tool, the Admin Console supports a PowerShell API that enables administrators to configure and monitor GO-Global farms programmatically.

When an administrator starts the Admin Console on a GO-Global Host, it attempts to connect to the APS that is running on the local computer. It connects to the port specified on the Security tab of the Admin Console’s Host Options dialog (port 491, by default). If the APS is running as a Farm Host, the Admin Console disconnects and attempts to connect to the APS on the Farm Host’s Farm Manager.

The Admin Console authenticates to the Application Publishing Service (APS) using Integrated Windows Authentication (IWA). It requires that the user running the Admin Console be a member of the computer's Administrators group. If an administrator runs the Admin Console on a Farm Host while signed in as a user who is not a member of the Administrators group on the Farm Manager, the IWA authentication will fail when the Admin Console attempts to connect to the Farm Manager. When this happens, GO-Global starts a remote session on the Farm Manager and prompts the administrator to sign in. After the administrator signs in to an account that is a member of the Administrators group on the Farm Manager, GO-Global runs the Admin Console in the remote session on the Farm Manager.

In addition to attempting to open a connection to the local APS or the APS of a Farm Host's Farm Manager, the Admin Console listens on port 491 for UDP broadcasts from other GO-Global Hosts and lists any hosts that are broadcasting on the local subnet. When administrators select a host in this list, the Admin Console attempts to connect to the APS on that host.

All GO-Global network traffic is, at the low layer, communicating over TCP/IP. Other protocols sit above this, such as WebSocket. The one exception is where the Admin Console listens on UDP port 491 for *broadcasts* from other GO-Global Hosts and lists any hosts that are broadcasting on the local subnet. When administrators select a host in this list, the Admin Console attempts to connect to the APS on that host.

GO-Global's UDP broadcast messages are sent over port 491 regardless of what port is specified on the **Security** tab of the Admin Console's Host Options dialog.

CHAPTER II

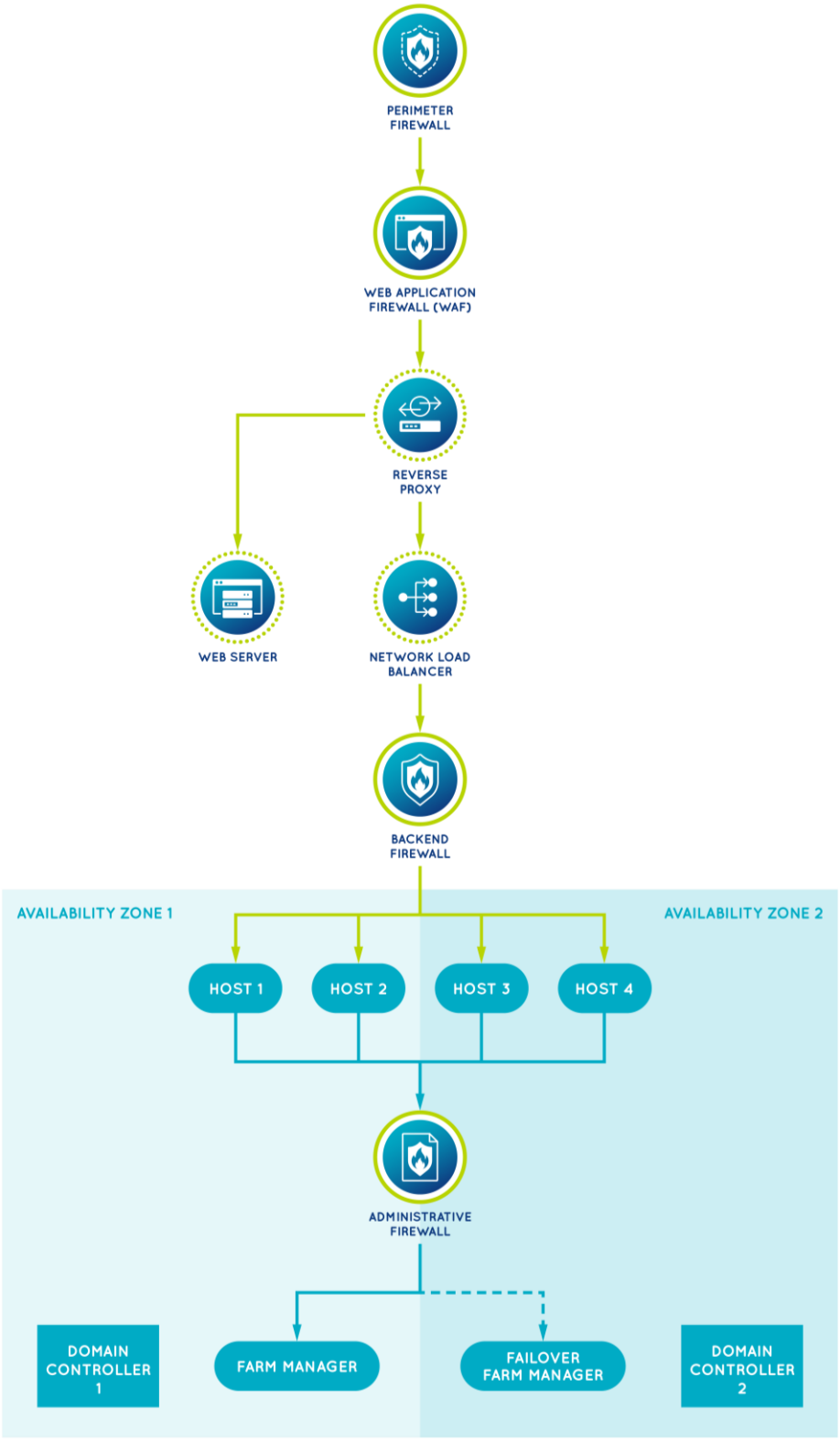
GO-Global Deployments

2

Large-scale GO-Global deployments usually include many supporting systems that provide traffic management, security, load balancing, scalability, high-availability, and other services. For example, in addition to GO-Global Farm Hosts and Farm Managers, a large-scale GO-Global deployment may include the following systems:

- Perimeter firewall
- Web application firewall
- Web server
- Reverse proxy
- Network load balancer
- Back-end firewall
- Domain controller
- Administration firewall
- Identity provider
- Auto-scaling infrastructure
- Availability zones and regions
- Monitoring tools

This chapter describes how to configure GO-Global to work in an environment that includes systems such as these. The following diagram illustrates how GO-Global fits into an environment that includes these systems.



[Diagram 7: Supporting Systems]

Perimeter Firewall

When a GO-Global deployment is accessible from the internet, a perimeter firewall is generally the first line of defense against intrusions. Perimeter firewalls are typically configured to allow incoming connections on TCP port 443, which is the standard port for secure HTTPS/TLS traffic.

Since port 443 is the standard port for secure traffic, and since it is the port that is open on most corporate (client) firewalls, GO-Global clients that connect over the internet are typically configured to connect to port 443 instead of GO-Global's standard port, 491. This is done by either adding the port in the URL (e.g., &port=443) or specifying the port in GO-Global's logon.html page.

Web Application Firewall (WAF)

A Web Application Firewall (WAF) analyzes incoming HTTP requests to identify and block malicious requests. WAFs can also route traffic to different backend servers based on information in HTTP requests, but this is not their primary function. Their primary function is to provide security and protect against attacks such as distributed denial of service (DDoS) attacks.

Examples of WAFs include Cloudflare WAF, AWS WAF (Amazon Web Services), Azure Web Application Firewall, F5 Advanced WAF, Imperva WAF, Akamai Kona Site Defender, Fortinet FortiWeb, Barracuda WAF, and Radware WAF.

To analyze HTTPS traffic, WAFs must terminate the TLS so they can decrypt the requests. When TLS is terminated by a WAF, GO-Global's **tls** parameter must be added to the client's URL or it must be specified in GO-Global's logon.html page.

Another consideration for GO-Global traffic is WAFs cannot analyze GO-Global's proprietary RXP protocol. Therefore, when communication between a GO-Global client and host passes through a WAF, it must either be wrapped in HTTP requests by configuring GO-Global to use a WebSocket, or the WAF's data analysis features must be disabled for the GO-Global traffic.

All version 6.3.3 and later GO-Global clients support WebSockets. In GO-Global versions 6.3.2 and earlier, however, only the GO-Global Web App supports WebSockets. AppController, GO-Global's native client, does not support WebSockets in version 6.3.2 and earlier.

If AppController version 6.3.2 or earlier is used with a WAF, the environment must be configured so the WAF does not inspect AppController's traffic. This can be done, for example, by providing another endpoint (public address) for AppController connections that do not use the WAF, or by configuring AppController to connect on a different port and configuring the WAF to pass traffic on this port without inspecting it.

Reverse Proxy

Like WAFs, reverse proxies analyze incoming HTTP requests. Reverse proxies have some security features, but unlike WAFs, the primary function of reverse proxies is traffic management, not security. For example, reverse proxies are often used when multiple web applications and services are provided from the same public endpoint. They route incoming requests for different applications to their respective application servers. They can also be used as a load balancer for GO-Global Farm Hosts.

Examples of reverse proxies include, Nginx, Apache HTTP Server, HAProxy, Traefik, and Cloudflare. The AWS Application Load Balancer and Azure Application Gateway have various functions, one of which includes serving as a reverse proxy.

Like WAFs, reverse proxies must terminate the TLS so they can analyze incoming HTTP requests. As with WAFs, when TLS is terminated by a reverse proxy, GO-Global's **tls** parameter must be added to the client's URL or it must be specified in GO-Global's logon.html page.

Like WAFs, reverse proxies cannot analyze GO-Global's proprietary RXP protocol. Therefore, as with a WAF, when communication between a GO-Global client and host passes through a reverse proxy, it must either be wrapped in HTTP requests by configuring GO-Global to use a WebSocket, or the reverse proxy's data analysis features must be disabled for the GO-Global traffic.

And as with a WAF, if AppController version 6.3.2 or earlier is used with a reverse proxy, the environment must be configured so the WAF does not inspect AppController's traffic. This can be done, for example, by providing another endpoint (public address) for AppController connections that do not use the reverse proxy, or by configuring AppController to connect on a different port and configuring the reverse proxy to pass traffic on this port without inspecting it.

Web Server

GO-Global Hosts have an integrated web server that can host the GO-Global Web App and its supporting HTML and JavaScript files. When GO-Global's integrated web server is used, HTTP requests for GO-Global's web files are forwarded to GO-Global Farm Hosts on the internal network.

In internet deployments, however, there are security benefits to hosting GO-Global's web files on a third-party web server located in the DMZ. Examples of third-party web servers include Apache HTTP Server, Nginx, and Microsoft IIS (Internet Information Services).

When a third-party web server is used, GO-Global's Web App and associated files must be installed on the web server, either via the Host installer or by copying the contents of GO-Global's Web directory to the web server.

While most third-party web servers have functionality to handle WebSocket connections, WebSocket connections are generally handled by a dedicated load balancer or reverse proxy. Administrators can configure WAFs and reverse proxies to route GO-Global's HTTP requests to a web server and to route WebSocket connections to GO-Global Farm Hosts or to a load balancer.

Alternatively, when a WAF or reverse proxy is not used, administrators can create a different endpoint (address or port) for GO-Global's WebSocket connections and specify this endpoint via the **host** and/or **port** parameters in the client's URL or GO-Global's logon.html page.

Network Load Balancer

When more than one Farm Host is needed, a load balancer is required to distribute user connections between the hosts. The load balancer can be a reverse proxy or a network load balancer.

In environments where multiple applications are accessed via a single internet endpoint, the perimeter firewall will typically forward connections to a reverse proxy (optionally via a WAF). The reverse proxy can then either forward the connections directly to Farm Hosts (balancing the load), or it can forward the connections indirectly to the hosts via a network load balancer.

Alternatively, in environments where GO-Global is the only application accessed via the internet endpoint, the perimeter firewall can forward connections on port 443 directly to a network load balancer. From there, the load balancer selects a Farm Host and forwards the connection to the Farm Host on the port specified in the Admin Console.

Unlike reverse proxies, network load balancers do not inspect traffic and route it based on the content of the data. They generally route traffic based on variables such as the destination port and source IP address. Examples of load balancers that can operate at layer 4 and route TCP connections (TLS is terminated at GO-Global hosts) are HAProxy, Azure load balancer, and AWS Network Load Balancer.

Some network load balancers can terminate TLS at the load balancer. If the load balancer is configured to terminate TLS, GO-Global's **tls** parameter must be added to the client's URL, or it must be specified in GO-Global's logon.html page.

When a load balancer terminates TLS, it can optionally re-encrypt the data that it transmits to Farm Hosts. If the Farm Hosts are configured to use the TLS protocol, the load balancer must be configured to re-encrypt the data that it sends to Farm Hosts. Conversely, if the load balancer does not re-encrypt the data, GO-Global's **Protocol** must be set to **TCP** and **Encryption** must be set to **None**.

When a network load balancer does not terminate TLS, it passes the data through to Farm Hosts without decrypting and re-encrypting it. In this configuration, the TLS protocol must be enabled on Farm Hosts, and the **tls** parameter must *not* be specified in the client URL or logon.html page.

Back-End Firewall

A back-end firewall may be used to limit access to Farm Hosts to specific front-end systems. For example, if reverse proxies are used to route connections to Farm Hosts, a back-end firewall may be configured to only allow the reverse proxies to connect to the Farm Hosts. In this example, the back-end firewall must be configured to allow the reverse proxies to connect to the Farm Hosts on the port specified in the Admin Console under Tools | Host Options | Security | Port (491, by default), and traffic must be allowed in both directions.

If the applications running on Farm Hosts must access the internet, then the back-end firewall must be configured to allow this access.

GO-Global Farm Hosts do not require access to the internet. If, however, GO-Global is activated using a cloud license, the APS on Farm Managers must be able to connect to portal.graphon.com and cloud.graphon.com on port 443.

Domain Controller

In the majority of deployments, Farm Hosts are members of a Microsoft Active Directory (AD) domain. In this environment, it is essential for Farm Hosts to maintain communication with their domain controllers. This connectivity enables them to authenticate accounts, retrieve user information and settings, load Roaming Profiles, and enforce Group Policies. In OpenID Connect deployments, customers are often required to configure constrained delegation on the AD computer objects of Farm Hosts.

Administration Firewall

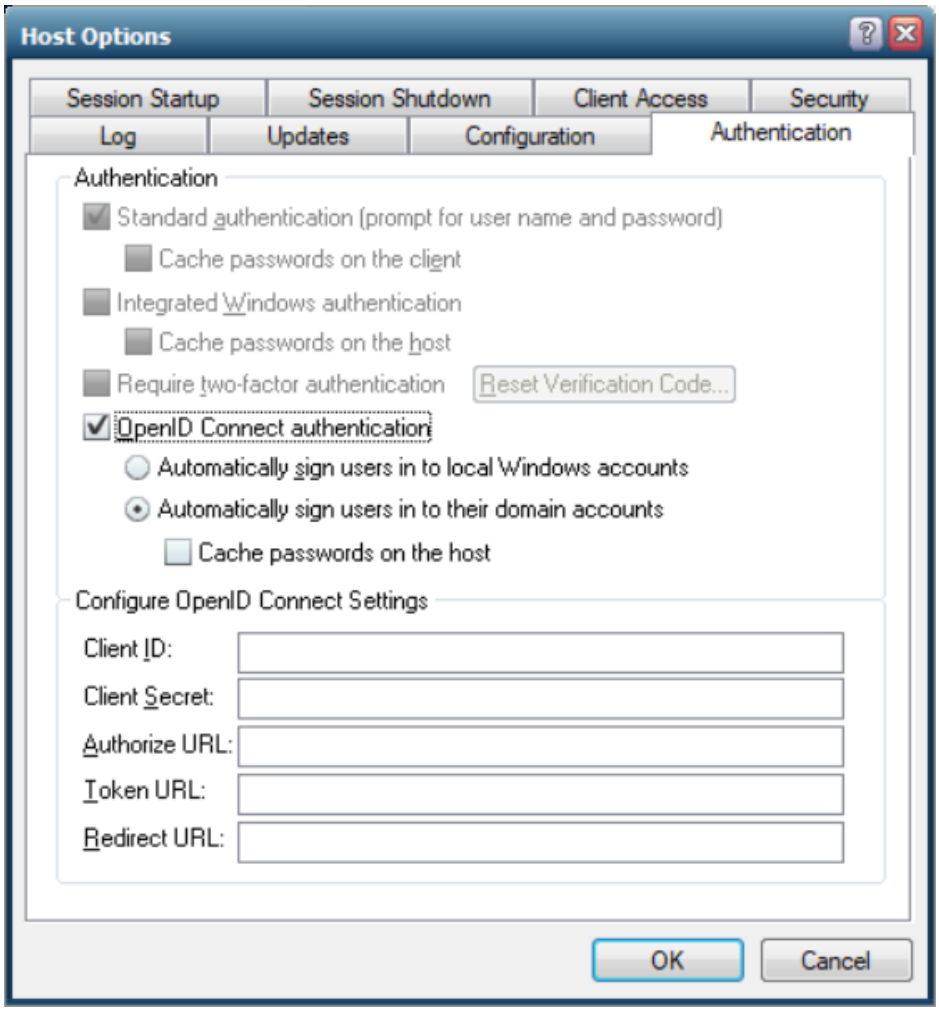
An administration firewall may be used to limit users' ability to access administration systems such as domain controllers and Farm Managers from Farm Hosts. For example, an administration firewall can be used to prevent users running applications on Farm Hosts from attempting to connect to domain controllers and Farm Managers using Remote Desktop.

If an administration firewall exists between Farm Hosts and Farm Managers, the firewall must allow Farm Hosts to connect to the Farm Managers on the port specified in the Admin Console under Tools | Host Options | Security | Port (491, by default), and traffic must be allowed in both directions.

If an administration firewall exists between Farm Hosts and domain controllers, the firewall must be configured to all communication on the ports that Windows uses to communicate between domain controllers and domain members. Refer to Microsoft's documentation for this information.

Identity Provider

When OpenID Connect Authentication is used, Farm Hosts must be able to open connections to the identity provider via the **Token URL** specified in the Admin Console under Tools | Host Options | Authentication | Configure OpenID Connect Settings | Token URL.



[**Authentication** tab of the **Host Options** dialog]

Auto-Scaling Infrastructure

Cloud Service Providers (CSPs) such as Amazon Web Services (AWS), Microsoft Azure, Oracle Cloud, and Google Cloud (GCP) provide features that can be used to automatically start and stop Farm Hosts as the load rises and falls. When using these features, administrators generally create a base ("gold") image of a Farm Host that can be replicated as needed to support the users connecting to the system.

Farm Host images have GO-Global installed and configured, as well as any applications that users will run on the Farm Hosts. Administrators typically automate the building of these images using a combination of PowerShell scripts and infrastructure code developed using an Infrastructure as Code (IaC) tool such as Terraform. See Automating the Configuration of GO-Global in the GO-Global Knowledge Base for information on how GO-Global can be installed and configured programmatically.

Farm Host images typically have the addresses of their Farm Manager and Failover Farm Manager stored in the image. This makes *scaling out* (adding hosts to a farm) straightforward. When new host instances are started, they automatically connect to the Farm Manager, and the CSP's auto-scaling feature generally takes care of adding the new instance into the load balancer's target group.

Farm Hosts obtain licenses from their Farm Managers, so no licensing configuration is required when a new Farm Host instance starts. When sessions start on a Farm Host, the host checks out license seats from the Farm Manager. Generally, one seat is consumed for each concurrent session.

Scaling in (removing hosts from a farm) is generally a little more complicated than scaling out. This is because it is usually desirable to wait to shut down a Farm Host instance until all sessions running on it have been closed. Taking a Farm Host Offline describes how to do this.

When a Farm Host stops, the host's connection to the Farm Manager is broken. When this happens, the Farm Manager automatically releases any seats that were in use on the host. As such, there is no risk that seats will be consumed by hosts that no longer exist. Nothing needs to be done to clear hosts that have been shut down; this is done automatically.

With GO-Global, the best metric to control auto-scaling is generally the number of running sessions. The number of sessions running on a host or a farm can be obtained, respectively from a Farm Host or Farm Manager via GO-Global's Performance Counters or via GO-Global's Get-GGSessions PowerShell function.

Availability Zones and Regions

To provide high availability, Cloud Service Providers (CSPs) support multi-datacenter deployments via availability zones and regions. Availability zones are one or more datacenters located within the same geographical region. CSP regions are geographical areas containing one or more availability zones. CSP Fault Domains, also known as availability groups, are fault tolerant resources located in physically separate parts of a single datacenter.

GO-Global supports deployments across multiple availability zones. The availability zones in which GO-Global is deployed may be located within the same CSP region or in different CSP regions.

Within a CSP region, GO-Global is generally deployed across at least two availability zones, with the primary Farm Manager running in one availability zone, and the failover Farm Manager running in a different availability zone. The Farm Hosts may run in any number of availability zones within the region, but they must be able to connect to both the primary Farm Manager and the failover Farm Manager.

When high availability across CSP regions is needed, or when users are located around the world, GO-Global may be deployed in multiple CSP regions with a DNS load-balancer such as AWS's Route 53 routing client connections across the regions. The DNS load balancer can then be configured to support active-active or active-passive failover, as needed.

Monitoring and Observability Tools

Monitoring and observability tools are used to monitor the health of GO-Global deployments and raise alerts when issues arise. GO-Global provides several features that these tools can use to monitor GO-Global.

Most observability tools have features for ingesting log files. GO-Global's log files are in HTML format, by default, which most observability tools do not support. Therefore, GO-Global must generally be configured to record its log files in text format when its logs will be ingested by an observability tool.

In addition to its log files, GO-Global records events in the Windows Application log. The events that GO-Global records in the Windows Application log include session starts and stops, user sign ins, and application starts and stops.

To support monitoring of Farm Hosts and Farm Managers, GO-Global provides a healthCheck request. The healthCheck request can be used to determine if the Application Publishing Service is running on a Farm Host or Farm Manager and, optionally, if the system is able to create a session. Load balancers, for example, can use the healthCheck request to check the health of Farm Hosts in their target group.

Systems that send healthCheck requests must be registered with each target Farm Host or Farm Manager. Specifically, the IP address of the system sending the request must be listed in CIDR format in the APIWhiteList property in the target system's HostProperties.xml file.

In addition to these tools, GO-Global's PowerShell API can be used to obtain detailed runtime information from Farm Hosts and Farm Managers. For example, the **Get-GGSessions** function returns information about the sessions running on a Farm Host or Farm Manager.

Appendix

Definitions

- **GO-Global Deployment:** One or more servers that use GraphOn's GO-Global product as the software to publish applications. This will include a supported combination of the roles below.

- **Farm Manager:** Centrally manages a collection of Farm Hosts. They manage information about joined Farm Hosts and client sessions. Unlike Relay Load Balancers, Farm Managers do not relay client connections to Farm Hosts.

- **Failover Farm Manager:** As above but on standby for use if the primary Farm Manager is unavailable. Also known as Backup Farm Manager.

- **Farm Host:** An application host that is *joined* to a Farm Manager. Users connect to Farm Hosts. Typical design is one Farm Manager, one Failover Farm Manager, and multiple *joined* Farm Hosts. Every GO-Global Farm Host runs a *light* web server spreading the load of this function across all Hosts.

- **Independent Host:** GO-Global Hosts that do not interact with other GO-Global Hosts running on the network. Users connect to Independent Hosts directly by specifying an IP address and TCP port, which can be public or private. Optionally a DNS address can be used.

- **Relay Load Balancer:** A Relay Load Balancer is a GO-Global Host that provides centralized control over one or more hosts. Relay Load Balancers maintain client connections and distribute sessions across a collection of load-balanced dependent hosts. Typical design is one or more Relay Load Balancers, and multiple *joined* Dependent Hosts. This role is never used with a Farm Manager and vice versa.

- **Failover Relay Load Balancer:** A Relay Load Balancer, as described above, but on standby for use if the primary Relay Load Balancer is unavailable. Also known as Backup Relay Load Balancer.

- **Dependent Host:** Hosts applications and is connected to a Relay Load Balancer. Users cannot connect directly to Dependent Hosts. Instead, they connect to the associated Relay Load Balancer, and the Relay Load Balancer selects one of the connected servers to host the session. The primary difference between a Farm Host and a Dependent host is that the former relies on a Farm Manager and the latter relies on a Relay Load Balancer. Every GO-Global Dependent Host runs a *light* web server spreading the load of this function across all Hosts.

- **Independent Host Manager:** An Independent Host Manager functions as a centralized license server, while enabling administrators to manage the sessions and settings of connected Managed Independent Hosts directly on each respective host.

- **Managed Independent Host:** An Independent Host can optionally be configured to connect to an Independent Host Manager for centralized license management. Apart from licensing, the Managed Independent Host continues to operate as a standalone server, maintaining its own published applications and configuration settings.

- **License Server:** An on-premises license server is a computer on which the GO-Global License Manager service is in use with license files, or alternatively, is hosting cloud licenses. The recommended option is to use GO-Global cloud licensing. This removes the requirement for customer-serviced license files. Administrators simply activate GO-Global Hosts with the Activation Wizard. In a farm or Independent Host Manager environment, GraphOn generally recommends only activating the Farm Managers/Independent Host Manager for cloud licensing. When high availability is required, redundant license servers or two activated Farm Managers can be configured.

- **GO-Global Farm:** A Farm Manager and its associated Farm Hosts. This may also include a failover Farm Manager. You can have multiple GO-Global Farms in a GO-Global deployment.

RDS & GO-Global: A Simplified Translation Guide

A typical Remote Desktop Services (RDS) deployment has the following components:

- One or two RDS Connection Broker servers (RDCB)
- One or more RDS Session host servers (RDSH)
- One or more RDS Web Access servers (RDWA)
- One or more RDS Gateway servers (RDGW)
- One or more RDS License servers (RDLS)
- One or more RDS 'collections'
- A Microsoft or third-party load balancer if scale or redundancy is required
- Microsoft Active Directory domain and DNS

GO-Global Architecture Type 1 (Recommended)

GO-Global Farm Manager

RDS Connection Broker — This RDS role balances and manages incoming connections. It maintains information about, and centrally manages, the RDS deployment. In contrast the GO-Global Farm Manager maintains information about, and centrally manages, the Farm Hosts, however, it does not maintain or balance connections to them. GO-Global can facilitate MFA via the inbuilt or OIDC options, whereas RDS does not offer MFA.

GO-Global Failover Farm Manager

Second RDS Connection Broker with High Availability *Configured* — This mode allows you to set up another connection broker to be available for redundancy. It requires setup of a shared database and DNS round robin or a load balancer to function. With GO-Global, the functions are the same as a Farm Manager, except the Failover Farm Manager is on standby should the primary become unavailable.

Third-Party Load Balancer (e.g., Azure load balancer)

RDS Connection Broker — This RDS role balances and manages incoming connections. It maintains information about, and centrally manages, the RDS deployment. With GO-Global, a third-party load balancer routes connections to the various farm hosts. It also performs health checks to make sure these hosts are responding, and if not, doesn't route connections to that host.

RDS Gateway — This server faces the end user and advertises access via TCP port 443. Incoming user connections first hit this server. With GO-Global a third-party load balancer performs a similar role to this on port 491 (customizable). With GO-Global a third-party load balancer performs the role of routing connections similar to the RDS gateway.

GO-Global Farm Host(s)

RDS Session Host — This server runs applications for user access, handling the primary compute tasks associated with user sessions. It is comparable to the GO-Global Application Host server roles.

RDS Web Access — This RDS server role allows users to log on via a web browser and download their RDP file to then access their session. This role requires Microsoft IIS. With GO-Global every Farm Host allows users to access their applications via a web browser, thus removing the need for having multiple RDS Web Access/RDS Gateway servers for redundancy and scale. Every GO-Global Farm Host runs a 'lite' web server spreading the load of this function across all Hosts.

RDS Web Client — This RDS optional component can be configured on the Web Access server's IIS. It allows users to run an RDS session from within a web browser. GO-Global Hosts allow access via a web browser by default.

RDP Client — The *Microsoft remote desktop connection client* is the end-user app that allows users to connect to an RDS collection manually or via an RDP file. The end-user app for connecting to GO-Global hosts is called AppController and is available for various operating systems.

GO-Global License Server

RDS License Server — This server role is installed to provide access to add licenses which must cover all unique/named (not concurrent) users or devices that have permission to sign in to RDS. GO-Global licenses are only required for concurrent user connections, significantly cutting costs over RDS. GO-Global licenses are usually loaded onto the same server as the Farm Manager. There is also the option of a GO-Global backup license instance (usually on the Failover Farm Manager).

GO-Global Architecture Type 2

GO-Global Relay Load Balancer

RDS Connection Broker — This RDS role balances and manages incoming connections. It maintains information about, and centrally manages, the deployment. The GO-Global Relay Load Balancer performs these same roles. GO-Global can also facilitate MFA via the inbuilt or OIDC options, whereas RDS does not offer MFA.

RDS Gateway — This server faces the end user and advertises access via TCP port 443. Incoming user connections first hit this server and are routed to a session host. The GO-Global Relay Load Balancer performs a similar role to this on port 491 (customizable).

GO-Global Dependent Host

RDS Session Host — This hosts applications for users to consume. It is where the direct user associated computer is utilized. This is similar to the GO-Global Host server role, with key distinctions below.

RDS Web Access — This RDS server role allows users to log on via a web browser and download their RDP file to then access their session. This role requires Microsoft IIS. With GO-Global every Dependent Host allows users to access their applications via a web browser, thus, removing the need for having multiple RDS Web Access/RDS Gateway servers for redundancy and scale. Every GO-Global Dependent Host run a ‘lite’ web server spreading the load of this function across all Hosts.

RDS Web Client — This RDS optional component can be configured on the Web Access server’s IIS. It allows users to run an RDS session from within a web browser. GO-Global Hosts allow access via a web browser by default.

RDP Client —The *Microsoft remote desktop connection client* is the end-user app that allows users to connect to an RDS collection manually or via an RDP file. The end-user app for connecting to GO-Global hosts is called AppController and is available for various operating systems.

GO-Global License Server

RDS License Server — This server role is installed to provide access to add licenses which must cover all unique/named (not concurrent) users or devices that have permission to sign in to RDS. GO-Global licenses are only required for concurrent user connections, significantly cutting costs over RDS. GO-Global licenses are usually loaded onto the same server as the Relay Load Balancer. There is also the option of a GO-Global backup license instance (usually on the failover Relay Load Balancer).

GO-Global Architecture Type 3

GO-Global Independent Host

RDS Session Host — This hosts applications for users to consume. It is where the direct user associated computer is utilized. This is similar to the GO-Global Host server role.

RDS Web Access — This RDS server role allows users to log on via a web browser and download their RDP file to then access their session. This role requires Microsoft IIS. With GO-Global every Independent Host allows users to access their applications via a web browser, thus removing the need for having multiple RDS Web Access servers for redundancy and scale. Every GO-Global Independent Host runs a *light* web server spreading the load of this function across all Hosts.

RDS Web Client — This RDS optional component can be configured on the Web Access server's IIS. It allows users to run an RDS session from within a web browser. GO-Global Hosts allow access via a web browser by default.

RDP Client — The *Microsoft remote desktop connection client* is the end-user app that allows users to connect to an RDS collection manually or via an RDP file. The end-user app for connecting to GO-Global hosts is called AppController and is available for various operating systems.

GO-Global License Server

RDS License Server – This server role is installed to provide access to add licenses which must cover all unique/named (not concurrent) users or devices that have permission to sign in to RDS. GO-Global licenses are only required for concurrent user connections, significantly cutting costs over RDS. GO-Global licenses are usually loaded onto the same server as the Independent Host Manager.

Port Requirements

GO-Global's port requirements are as follows:

- **Independent Hosts** must allow connections from clients on the port specified under Admin Console | Tools | Host Options | Security.
- **Dependent Hosts** must connect to the Relay Load Balancer on the port specified on the Relay Load Balancer under Admin Console | Tools | Host Options | Security.
- **Relay Load Balancers** must allow connections from clients and Dependent Hosts on the port specified under Admin Console | Tools | Host Options | Security.
- **Farm Hosts** must connect to the Farm Manager on the port specified on the Farm Manager under Admin Console | Tools | Host Options | Security. Farm Hosts must allow connections from the third-party load balancer on the port specified on the Farm Manager under Admin Console | Tools | Host Options | Security.
- **Farm Managers** must allow connections from Farm Hosts on the port specified under Admin Console | Tools | Host Options | Security.
- To use a **cloud license**, the Application Publishing Service must be able to connect to portal.graphon.com and cloud.graphon.com on port 443.
- To use a **Backup License Manager**, the Application Publishing Service must be able to connect to the Backup License Manager on the port specified under Admin Console | Tools | Host Options | Security, and the Backup License Manager must be configured to accept connections on the same port.
- To use an **on-premises license**, the Application Publishing Service must be able to connect to the GO-Global License Service (lmgrd.exe), and lmgrd.exe must be able to connect to the GO-Global license daemon (blm.exe). If there are firewall rules that restrict these connections, the GO-Global License Service should be configured to use port 27000 and blm.exe should be configured to use port 5678. In addition, if the Application Publishing Service has access to the internet, it must be able to connect to license.graphon.com on port 443.
- **Independent Host Managers** must allow connections from Independent Hosts on the port specified under Admin Console | Tools | Host Options | Security.
- **Managed Independent Hosts** must connect to the Independent Host Manager on the port specified on the Independent Host under Admin Console | Tools | Host Options | Security.

- GO-Global's **Universal Printer Driver** uses port 9010. This port cannot be changed. If any other software on a GO-Global Host uses port 9010, users will be unable to print with the Universal Printer Driver.

- The **Admin Console** can optionally allow broadcast messages over UDP port 491, which will allow it to populate a list of other hosts. This port number cannot be changed.

Independent Host Deployment Checklist

- ❑ Configure the hardware (physical, local or cloud virtual machine)
- ❑ Install and configure the Windows operating system
- ❑ Sign in with a user account that has administrator privileges
- ❑ Attach server to the network
- ❑ Perform Windows updates then implement ongoing Windows and app patching strategy
- ❑ Set a Windows computer name, and if applicable, join an AD domain
- ❑ Restart the computer
- ❑ Add international keyboard support (optional)
- ❑ For larger deployments with multiple application hosts, consider a Farm deployment instead of Independent Hosts
- ❑ Add DNS records for the host (if applicable)
- ❑ Install and configure application(s)
- ❑ Create/modify non-administrator users locally or in AD
- ❑ Configure server policies if required
- ❑ Install the GO-Global Host software which can be downloaded from the GraphOn Customer Portal
- ❑ Install a GO-Global license file, or use the Activation Wizard for cloud licensing
- ❑ Restart the computer
- ❑ Publish user apps using the Admin Console
- ❑ Set any server options via the Admin Console that you require
- ❑ Session startup
  - Group policy
  - Logon scripts
  - Limits

- ❑ Session shutdown
  - Idle timeouts
  - Disconnect action
- ❑ Client access
  - Clipboard
  - Printers
  - Client drivers
  - Open files on client
- ❑ Authentication
  - MFA
  - SSO/OIDC
- ❑ Security
  - Connection Protocol (TLS recommended)
- ❑ Secure the host with AV/EDR, network firewall, hardening as required
- ❑ Apply corporate branding using the Branding dialog (optional)
- ❑ Publish the host to the internet, VPN, or DMZ (optional)
- ❑ Configure server backups if applicable
- ❑ Configure monitoring and alerting (optional)