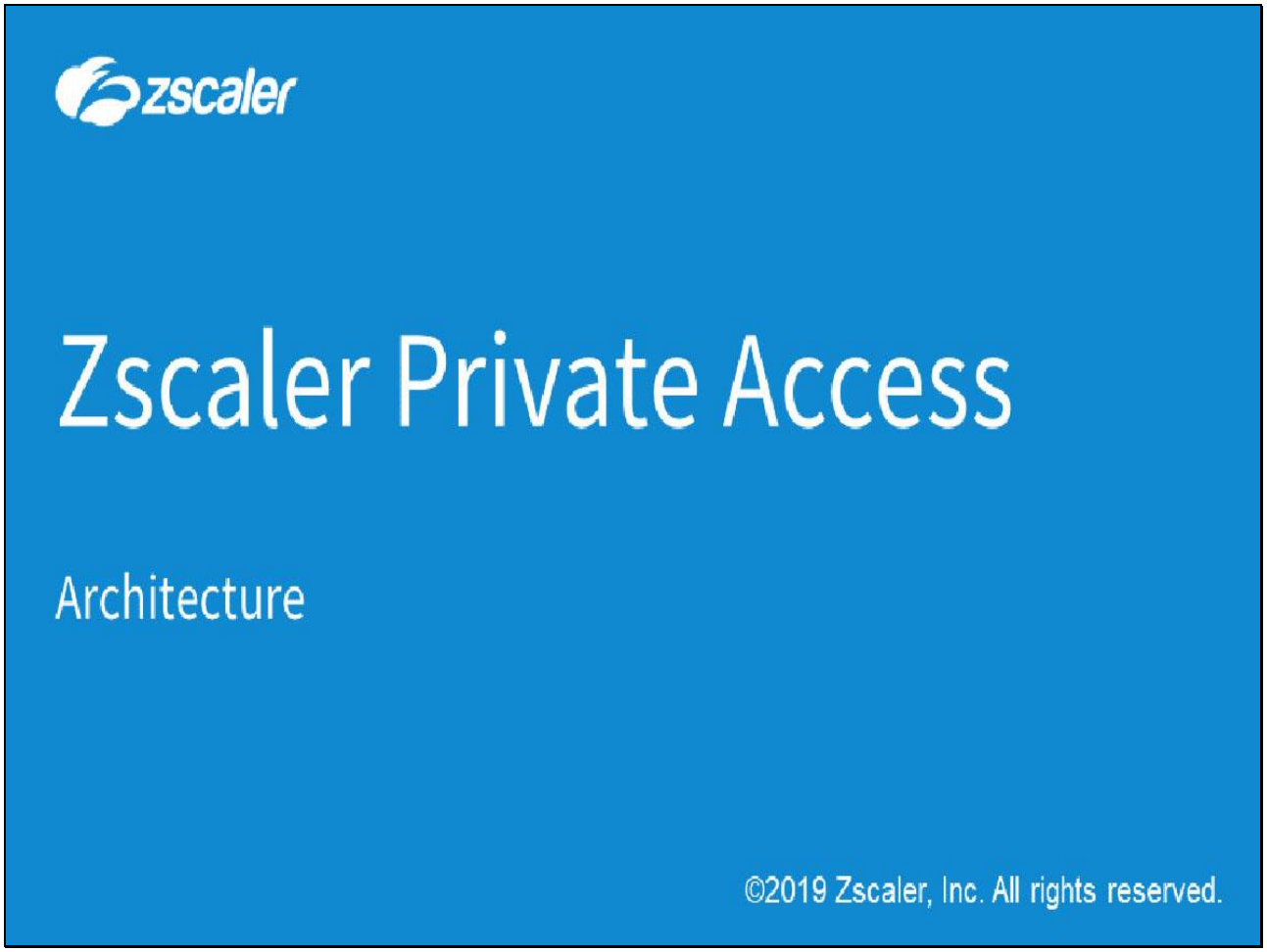
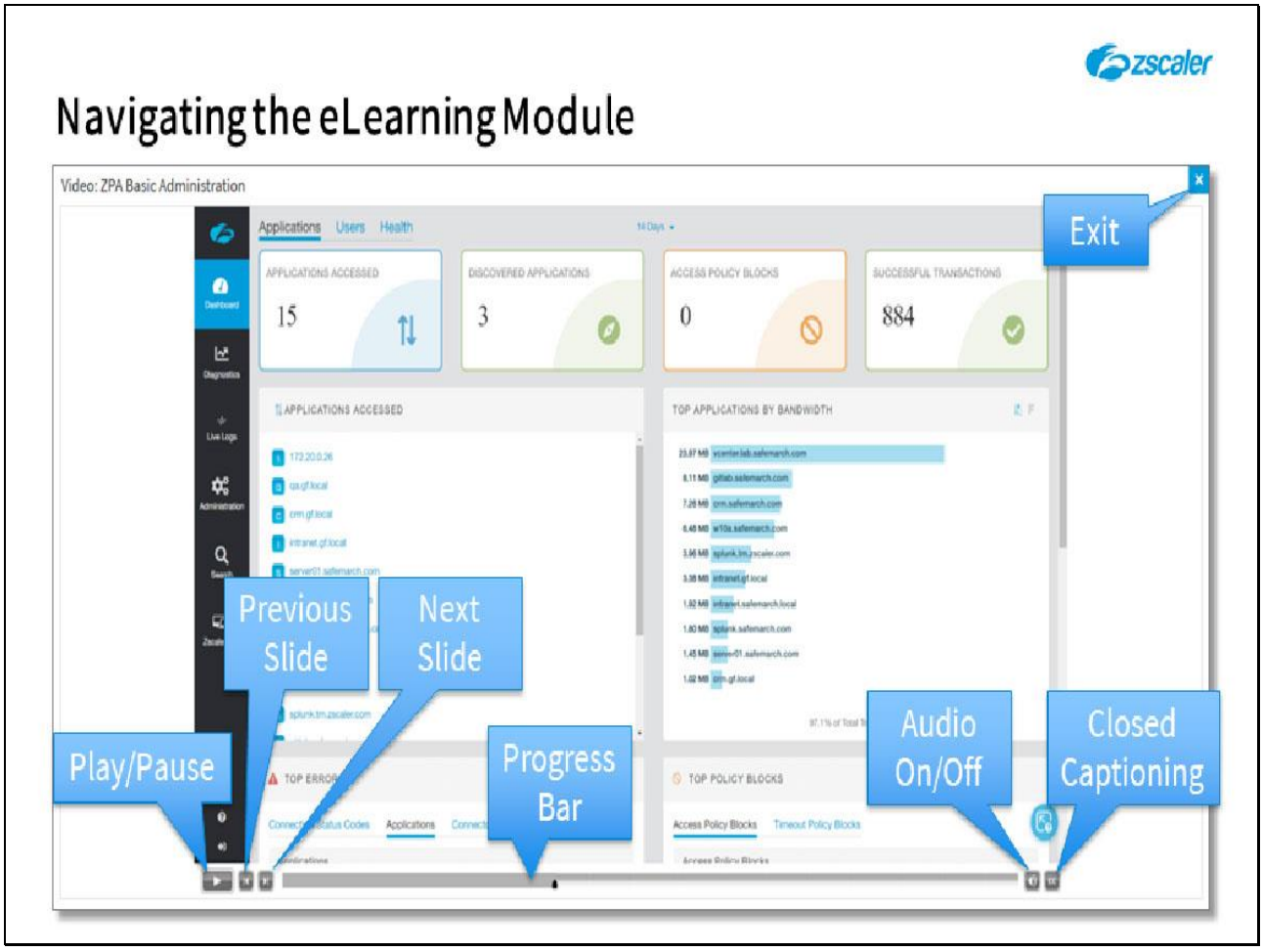
Slide 1 - Zscaler Private Access



Slide notes

Welcome to this training module on the Zscaler Private Access architecture.

Slide 2 - Navigating the eLearning Module



Slide notes

Here is a quick guide to navigating this module. There are various controls for playback including **Play** and **Pause**, **Previous**, and **Next** slide. You can also **Mute** the audio or enable **Closed Captioning** which will cause a transcript of the module to be displayed on the screen. Finally, you can click the **X** button at the top to exit.

Slide 3 - Agenda

Slide notes

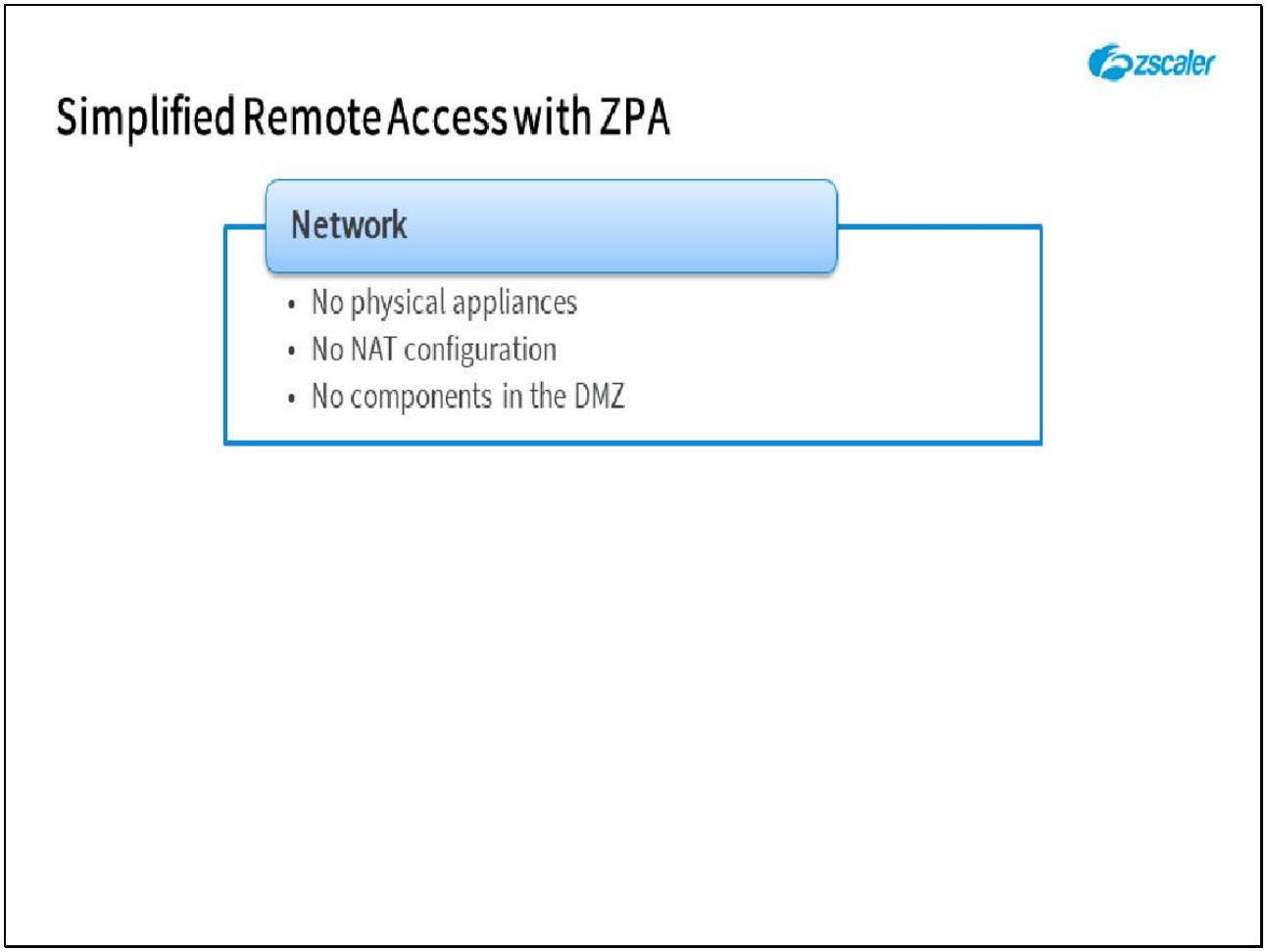
In this module we will we will look at: the ZPA architecture; the Browser Access (BA) architecture; ZPA security; the relationships between the various ZPA logical entities; the process of application discovery; and at the criteria used to select the best data path for a user.

Slide 4 - ZPA Architecture

Slide notes

The first topic we will cover is a look at the ZPA architecture.

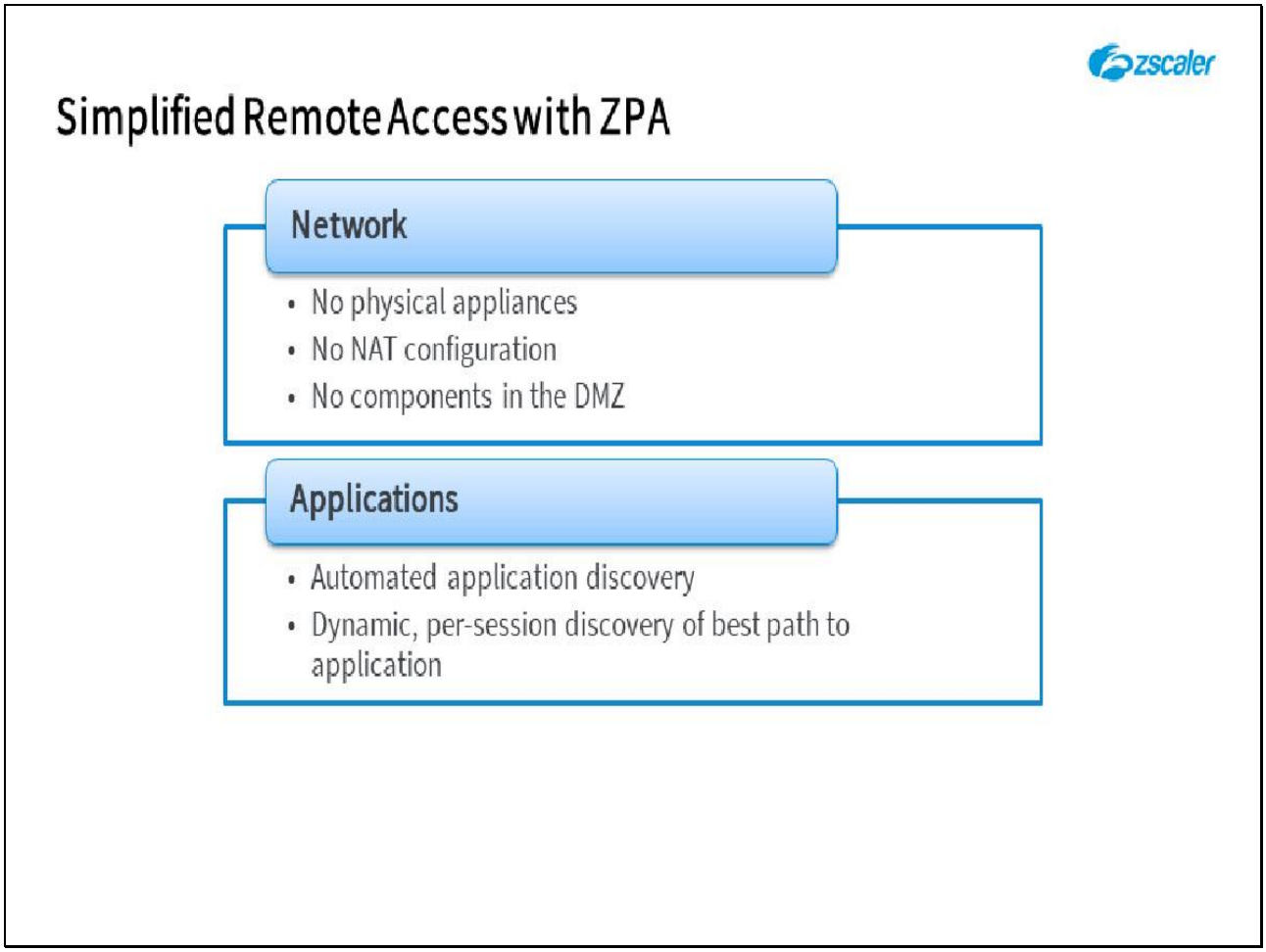
Slide 5 - Simplified Remote Access with ZPA



Slide notes

One of the best parts of the ZPA solution is its ease of deployment. Firstly, there are no physical appliances to deploy, no Network Address Translations to contemplate, nothing new to put in the DMZ.

Slide 6 - Simplified Remote Access with ZPA

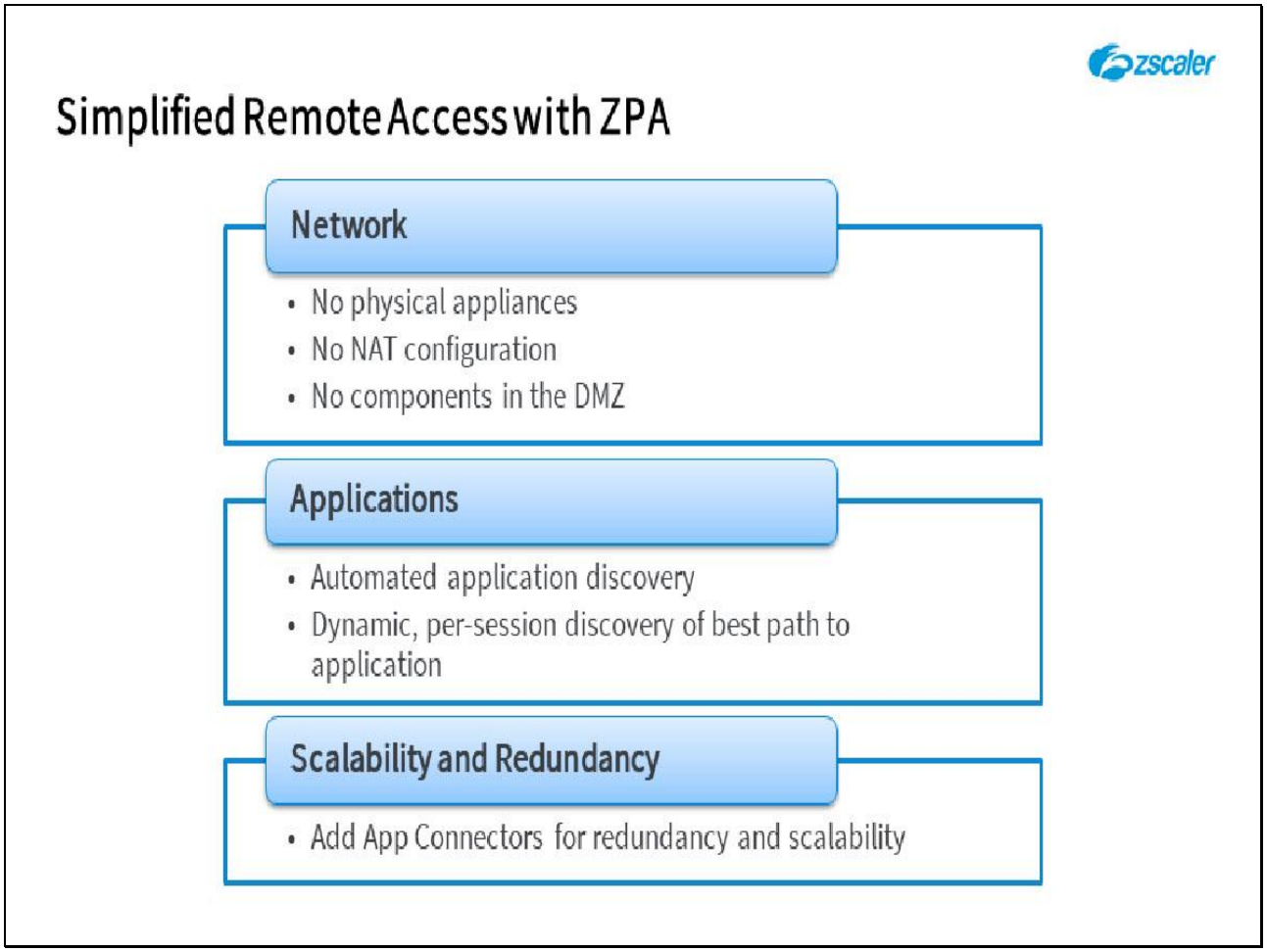


Slide notes

With the application discovery capability, the combination of global ZPA-ZENs and App Connectors in front of your servers can actually find applications in response to user requests. So IT does not have to map out IP addresses for each and every application - the ZPA solution will find them for you.

Also, if there is more than one path through the ZPA-ZENs and Connectors to an application, ZPA will apply a series of metrics to figure out which path will deliver the ideal experience for each individual user request. Every application connection is unique, although this won't be visible to the users - from their perspective, it 'just works'.

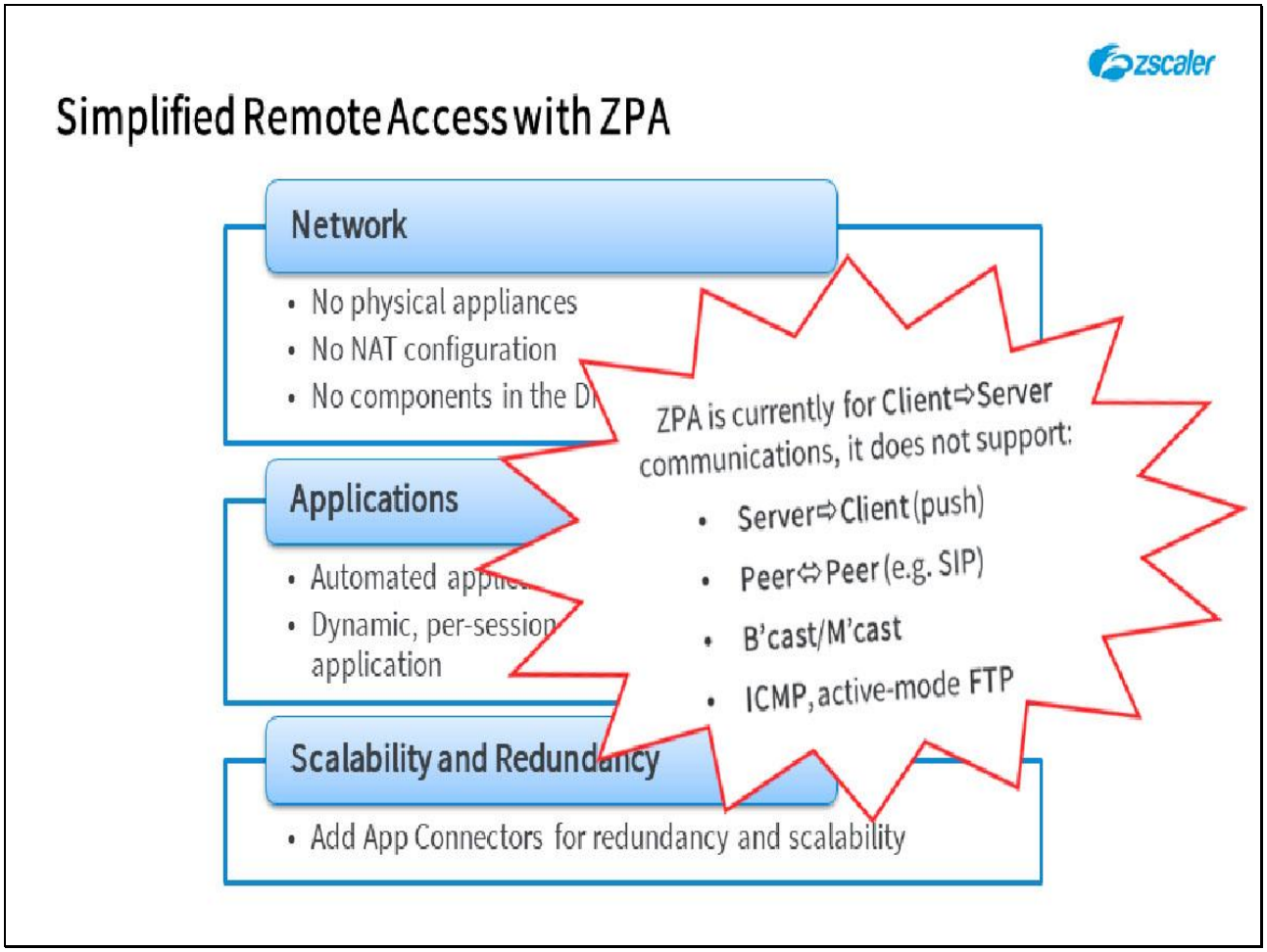
Slide 7 - Simplified Remote Access with ZPA



Slide notes

Each of the Connectors delivers about ½ Gbps of throughput but note that the Connectors were designed to scale horizontally and are completely independent of one another. If you need more capacity just add another Connector, which you can spin up in a matter of seconds - no clustering or load-balancing required.

Slide 8 - Simplified Remote Access with ZPA

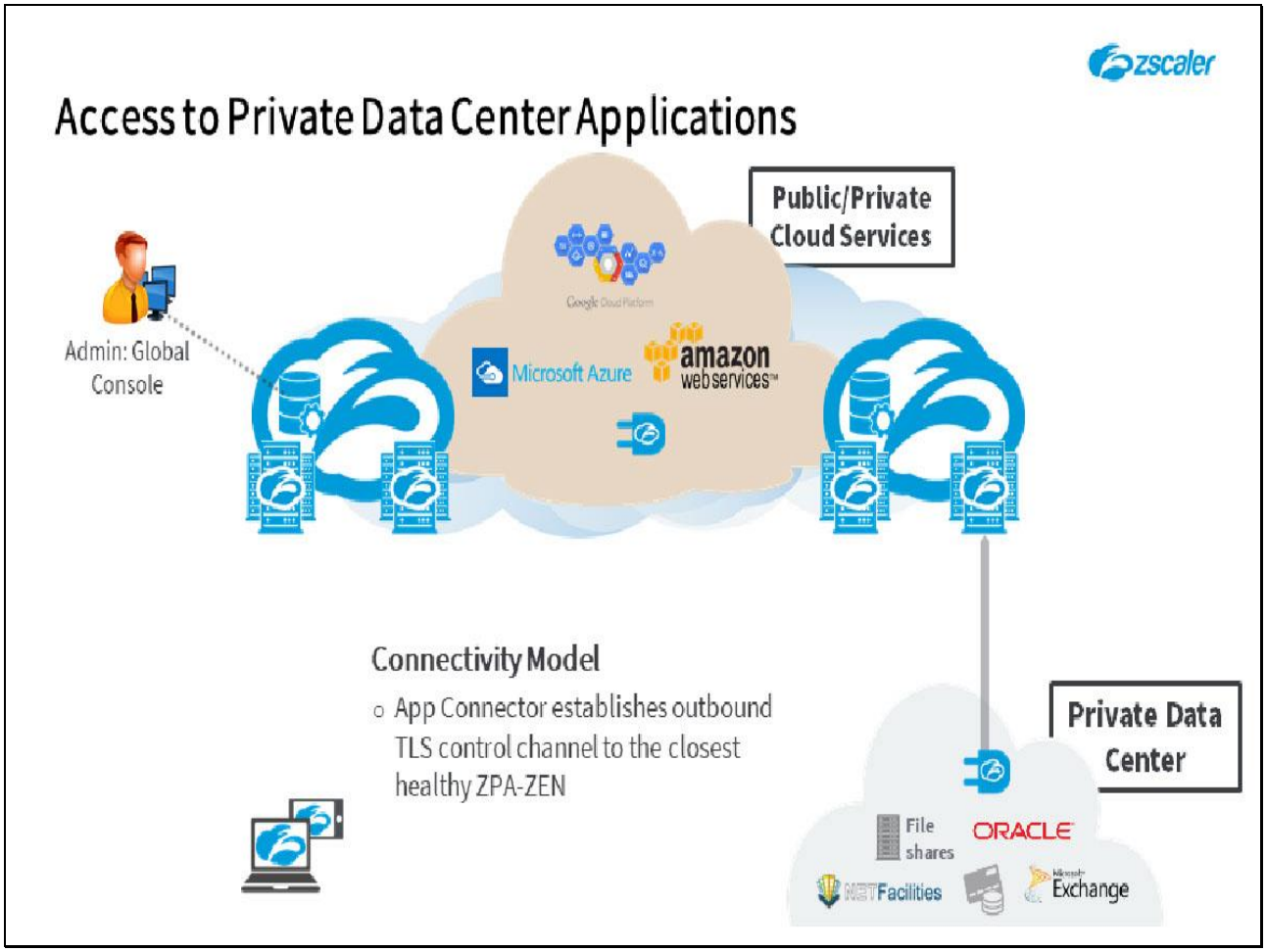


Slide notes

Note that ZPA is intended for client-to-server communications. Certain connectivity models and protocols are not currently supported, including:

- Server-to-client (push);
- Peer-to-peer, including applications that require an Application-Level Gateway (ALG) such as SIP, RTSP or BitTorrent;
- Broadcast and Multicast traffic;
- ICMP;
- Also, the FTP protocol in active-mode.

Slide 9 - Access to Private Data Center Applications



Slide notes

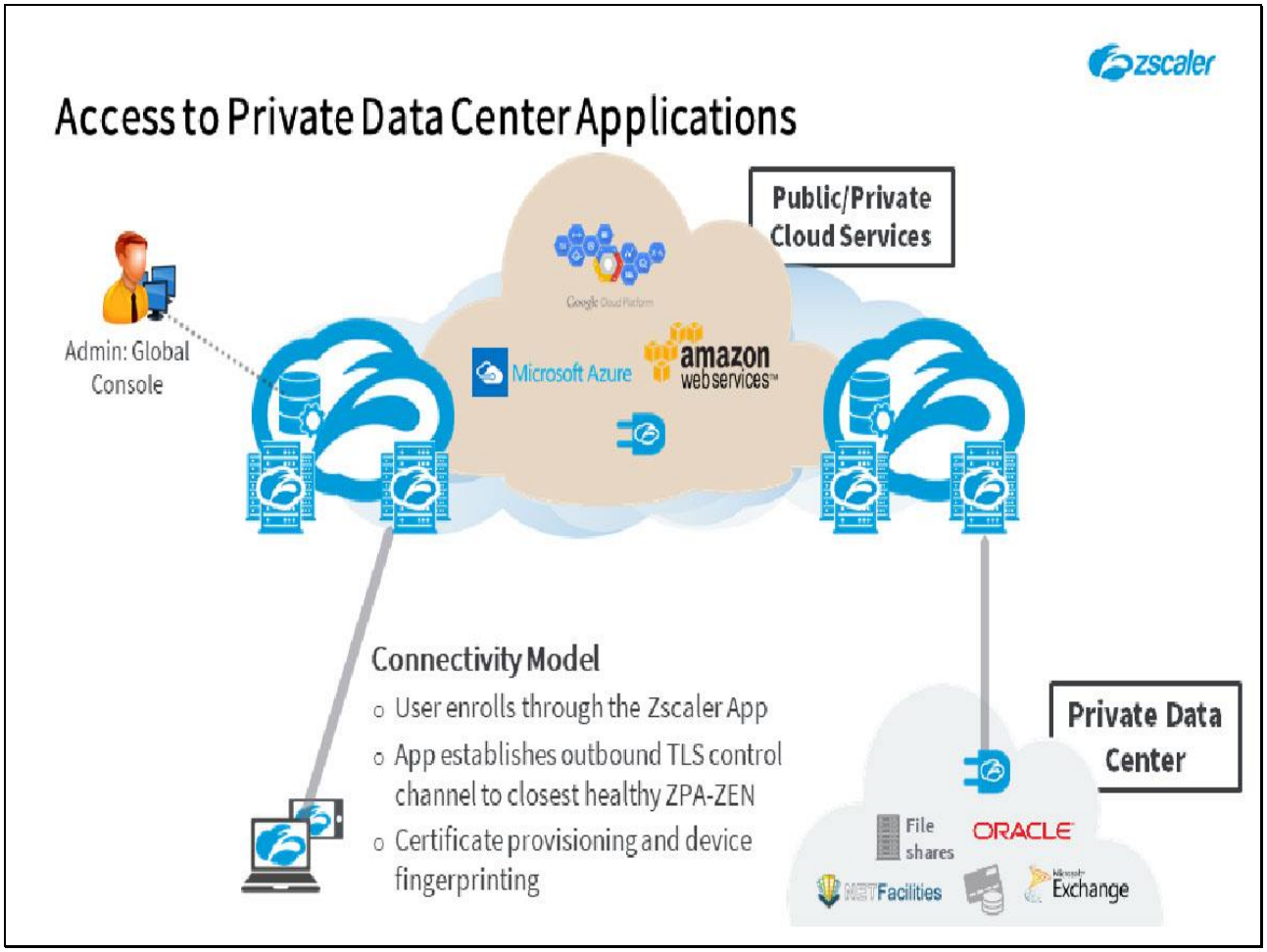
Let's dig a little deeper to see how this all works in practice. We will look here at how this works with the Zscaler App installed. With Z App, you can enable connectivity to just about any back-end application, regardless of protocol or port. The architecture for the Browser Access (BA) capability is dealt with in the next section of this module.

When you deploy an App Connector, as it boots onto the network, it 'calls home' and establishes an outbound encrypted TLS connection to the closest healthy ZPA-ZEN. This connection is used as a control channel to allow configuration of the Connector, and for it to notify the ZPA-CA of any discovered applications when required.

The enrollment of the Connector is secured through a **Provisioning Key** that is signed by the specified Subordinate CA for Connectors on the ZPA instance. On enrollment, the Connector requests and receives both an identity certificate (used to authenticate its Z Tunnels to the ZPA infrastructure), and a server certificate issued by the Sub-CA (which is used to establish encrypted Microtunnels when necessary).

A device fingerprint is captured at this point with unique data from the host device, to prevent any possibility of cloning the Connector and its certificates for unauthorized access.

Slide 10 - Access to Private Data Center Applications



Slide notes

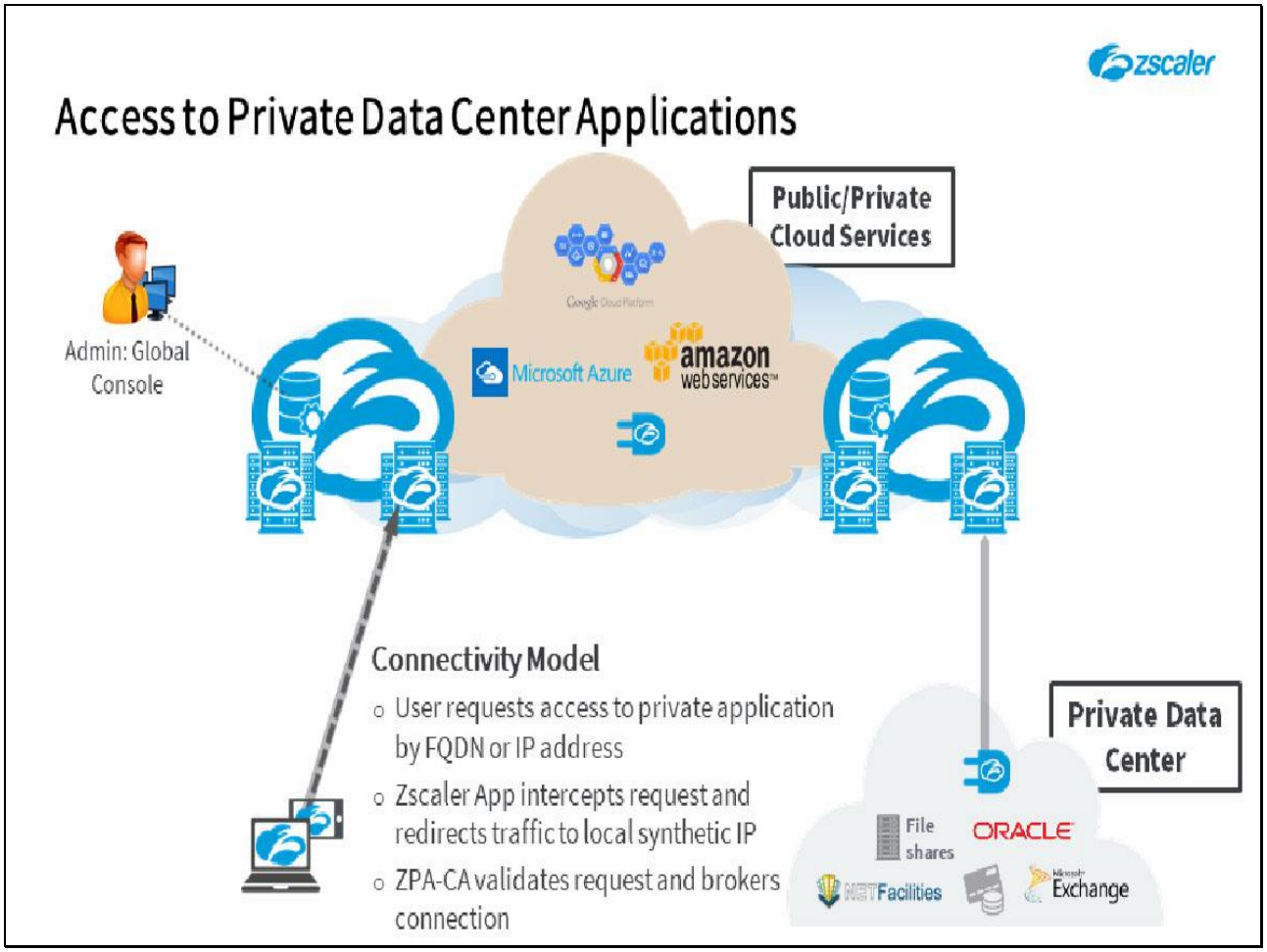
Simultaneously, the Zscaler App deployed to the end devices of your users will also connect to the closest healthy ZPA-ZEN, to establish an encrypted TLS control channel. Client enrollment is secured by a SAML authentication through the Zscaler App and on a successful enrollment an identity certificate (signed by the Subordinate CA for clients on the ZPA instance) is provisioned to the client.

The keys and certificates are stored securely within the Zscaler App itself, and they are renewed each time the user re-enrolls. If the administrator disables ZPA for that device through a **Force Remove** operation at the Zscaler App Portal, these certificates are automatically revoked.

A device fingerprint is captured at this point with unique data from the host device, to prevent any possibility of cloning the Zscaler App and its certificates for unauthorized access. The unique data captured includes among other items; the CPU ID, the HD serial number, and the battery unique ID. This fingerprint is validated at each check in of the Zscaler App.

The App is notified by the ZPA-CA of the private applications available, and of the applications that require double encryption.

Slide 11 - Access to Private Data Center Applications



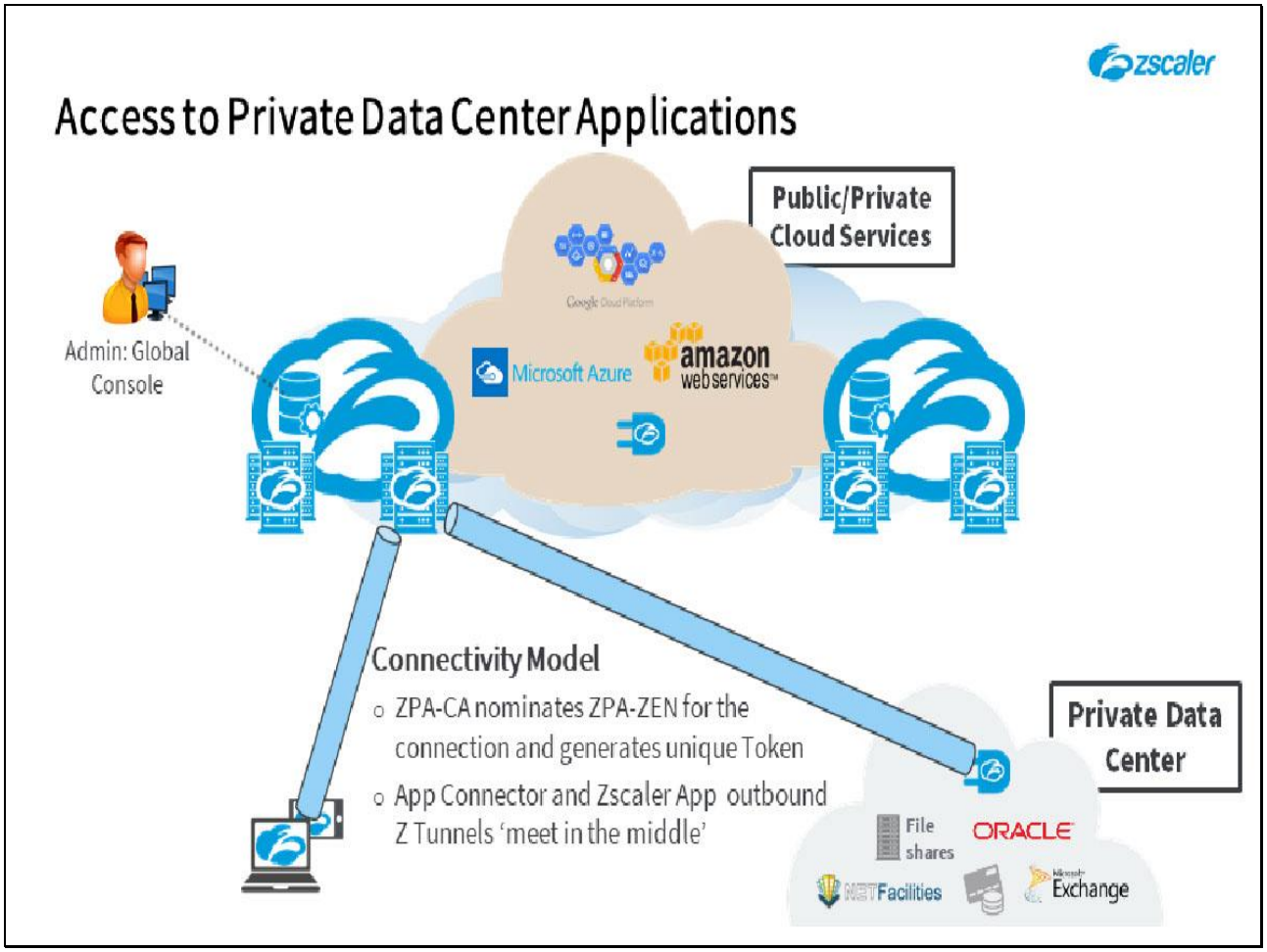
Slide notes

When the user requests access to a known private application (whether by FQDN or by IP address) the Zscaler App will intercept the request and, if access was requested by FQDN, establish a local synthetic IP address for the outbound traffic. The synthetic IP addresses are allocated from the RFC 6598 Carrier-grade range of **100.64.0.0/10**, although the Zscaler App uses a 16-bit mask. The Zscaler App essentially emulates the destination server for the client software, so that the client sends all requests to the allocated synthetic IP address.

The Zscaler App uses its connection to the ZPA-ZEN to validate the requested application and TCP/UDP port being requested and provides the current SAML assertion for the user. The ZPA-ZEN looks up policy against the ZPA-CA to verify that access is permitted according to the defined policies and that the application is available, then selects the best Connector to participate in the connection.

If necessary, during application discovery, the Connectors will be instructed to try to find the requested application using DNS requests on their local networks. The Connectors support A, AAAA, and SRV records for application discovery.

Slide 12 - Access to Private Data Center Applications



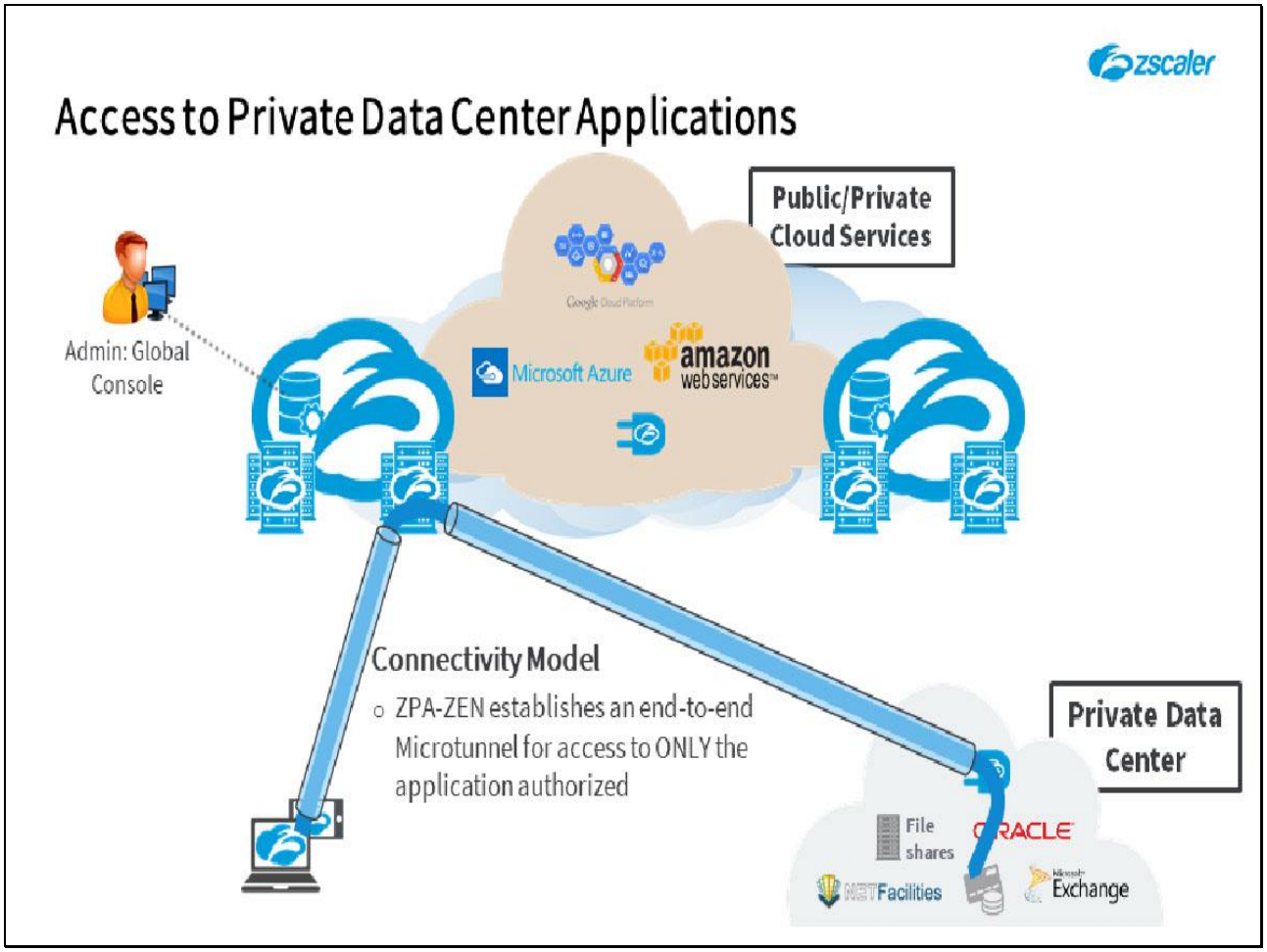
Slide notes

The ZPA-CA instructs the chosen Connector to meet at the optimum ZPA-ZEN, chosen to minimize end-to-end latency, and both the Zscaler App on the client device and the appropriate Connector establish outbound encrypted TLS tunnels (Z Tunnels) to the nominated ZEN.

The full path from ZEN to Connector to application is selected for ideal user experience, based on; geographical proximity to the user, RTT from the Connector to the application, on a round robin basis, and using a measure of 'stickiness' so that a user will generally return to a Connector used previously.

Note that as a Connector may need to serve many users simultaneously, it may establish Z Tunnel connections to multiple ZPA-ZENs simultaneously, depending on where the end users are connecting from. Also, depending on the applications requested by the user and their locations, the Zscaler App may establish multiple Microtunnel connections through its Z Tunnel.

Slide 13 - Access to Private Data Center Applications



Slide notes

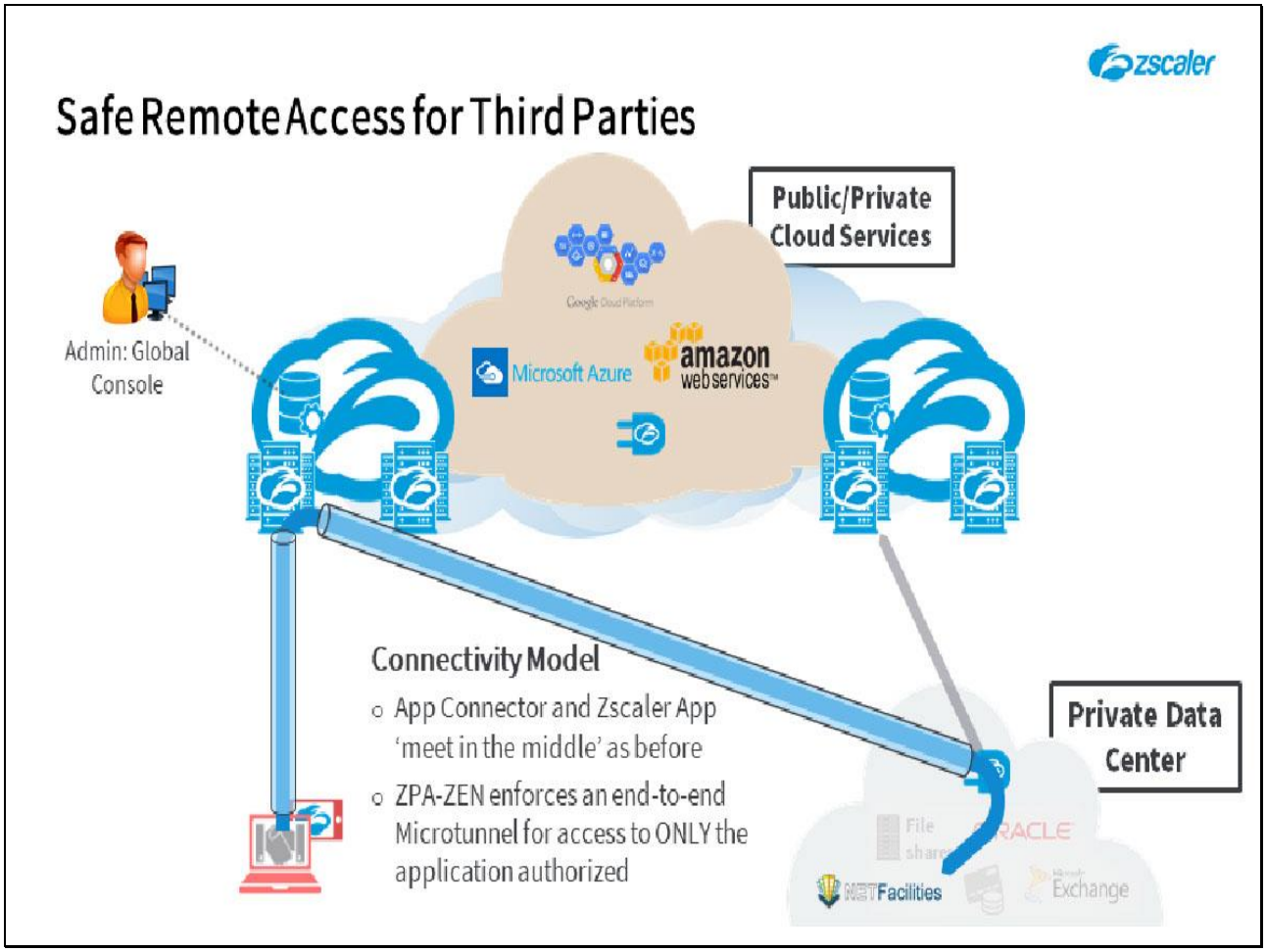
Once the Z Tunnels are in place the Connector is then instructed to create a ZPA Microtunnel with a unique Microtunnel ID, through the Z Tunnel to the relevant ZPA-ZEN. At the same time the Zscaler App on the end user's device is also instructed to establish a Microtunnel to the same ZPA-ZEN, through its Z Tunnel. The ZPA-ZEN then brokers these two Microtunnels to establish an end-to-end application traffic flow between the Zscaler App and the Connector.

A Microtunnel is a simple byte stream with source and destination uniquely identified by the tags generated for that purpose; at no point in the communication is an actual IP address used. The Microtunnel IDs can be likened to the labels for a label-switched path in an MPLS network.

All traffic destined for the requested private application is transported within this tunnel and delivered to the destination network by the Connector. The Connector acts on behalf of the client's browser or agent SW and the private application replies to it so that it can send the return traffic along the reciprocal Microtunnel path back to the end user. While the application actually talks directly to the Connector, it believes that it is talking to the requesting end user device.

As previously stated, the Microtunnels are only available for use by the requesting user and only for access to that specific application. Microtunnels cannot be shared with other users, nor can they be used to route to other applications that may exist on the destination network. Access to other private applications would require Microtunnels established specifically for that purpose, so if an end user needs access to multiple private applications they may have several simultaneously active Microtunnels, to the same or to different Connectors.

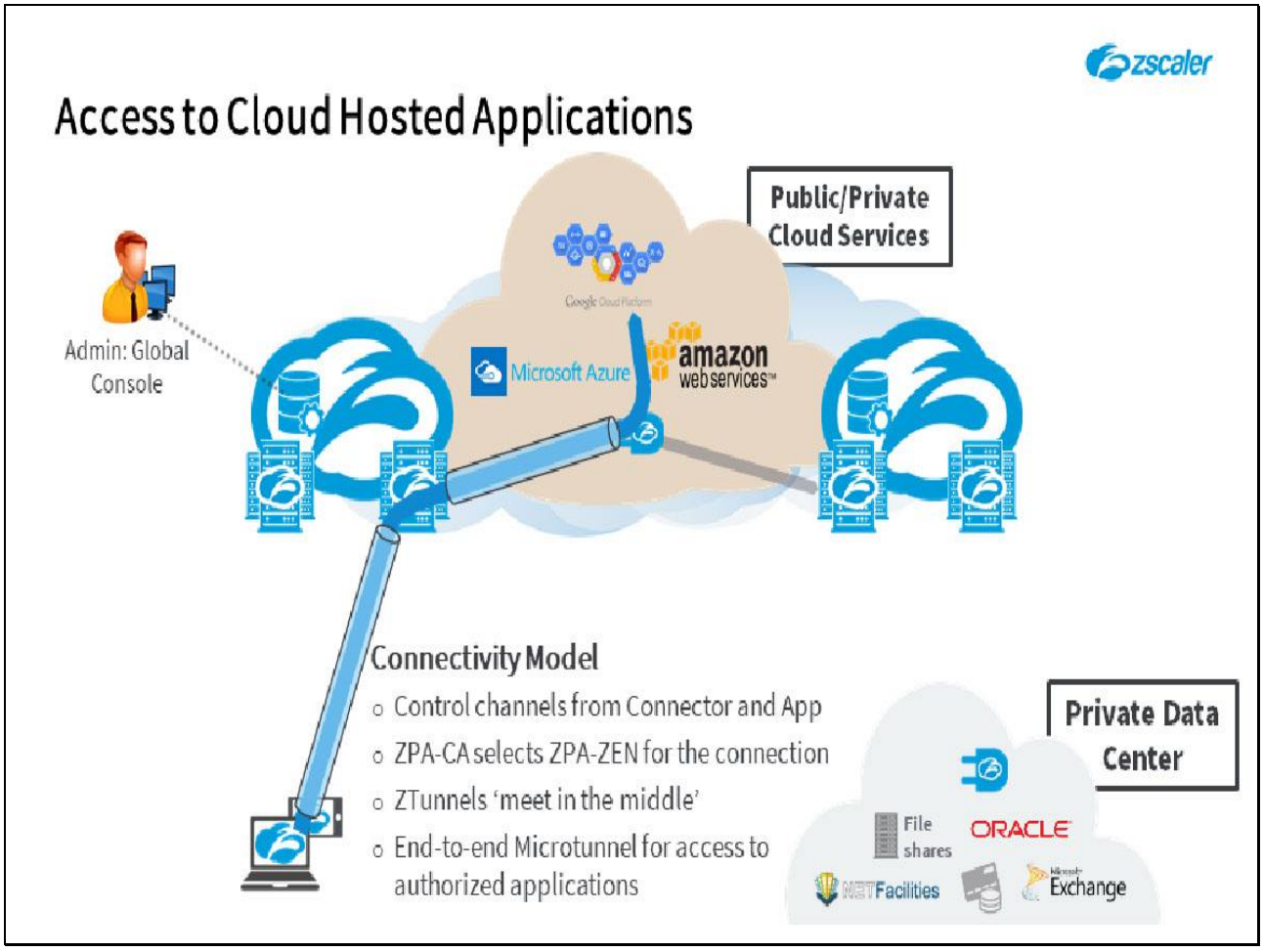
Slide 14 - Safe Remote Access for Third Parties



Slide notes

As a result of all this, ZPA is particularly useful for allowing access to 3rd parties, such as contractors, or vendors, who only require access to a single specific application. The Microtunnel model of ZPA prevents access to any other application, unless it is specifically enabled in the ZPA-CA console.

Slide 15 - Access to Cloud Hosted Applications



Slide notes

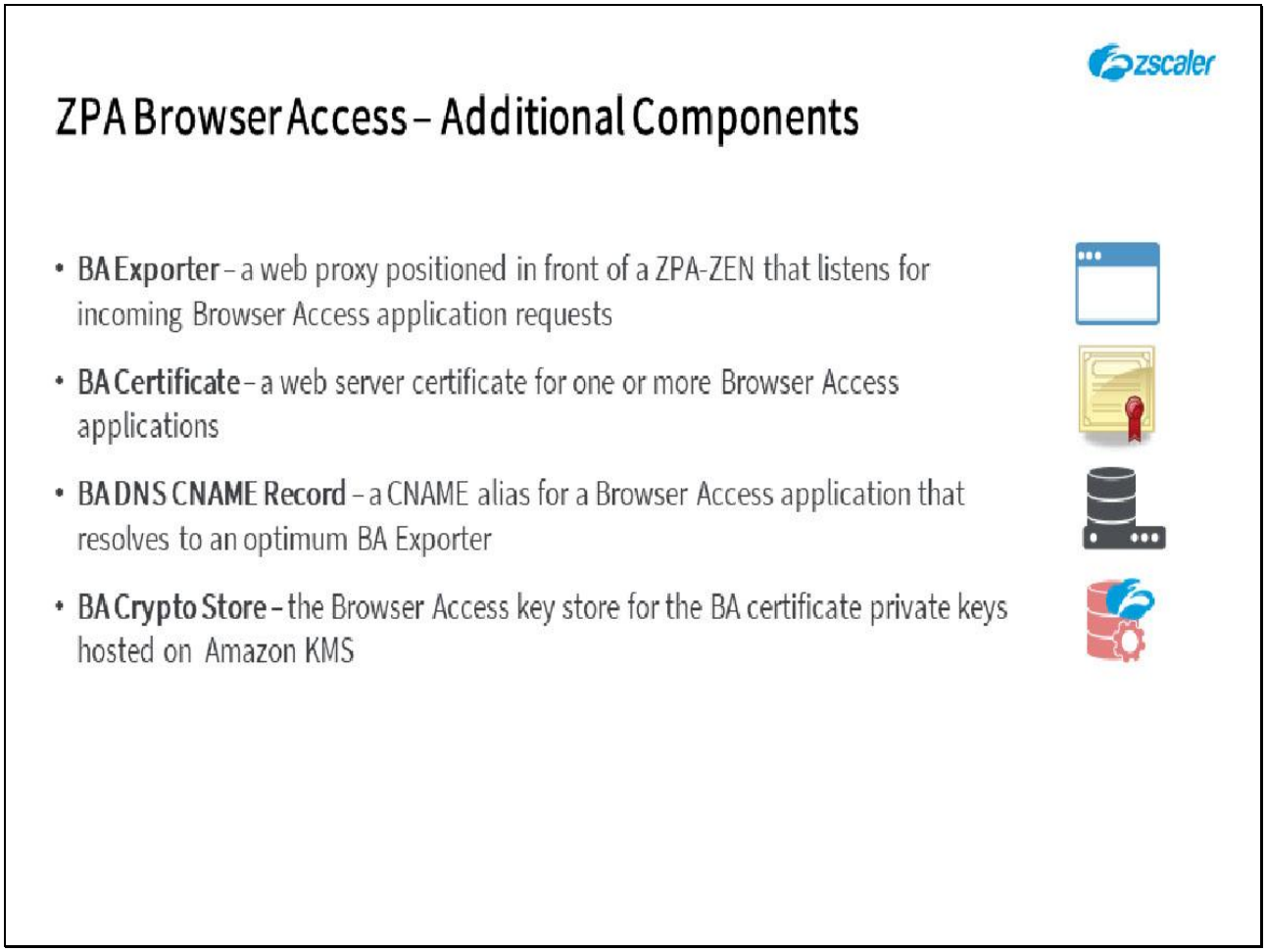
Access to private applications that are hosted in the Cloud can be managed in the exact same way. The only requirement is to install a Connector VM adjacent to the application, to allow it to be reached by the end users. Pre-built Connector images are available for both AWS and Azure, and Connectors can be installed via RPM in other cloud environments. The connection setup process is exactly the same as that described for on-premise applications.

Slide 16 - Browser Access Architecture

Slide notes

In the next section, we will provide an overview of the ZPA Browser Access (BA) architecture for access to web-based applications.

Slide 17 - ZPA Browser Access – Additional Components

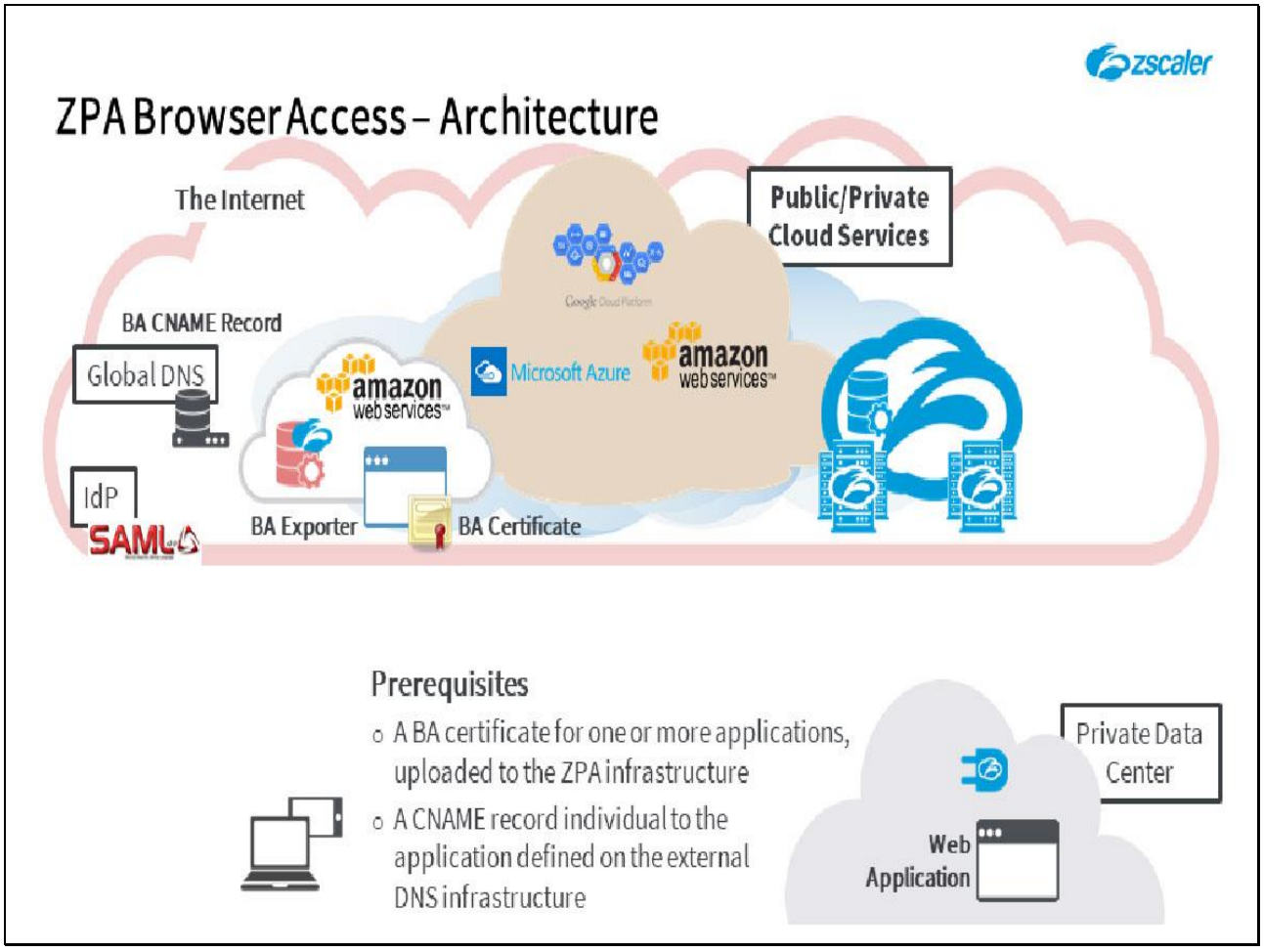


Slide notes

In addition to the regular ZPA architectural components, BA introduces some new components as listed and described here:

- **BA Exporter** - this is a secure web proxy sitting in front of a ZPA-ZEN that listens for incoming BA application requests;
- **BA Certificate** - a web server certificate for one or more Browser Access applications (for use for multiple applications, this may be a wildcard certificate);
- **BA DNS CNAME Record** - a CNAME alias for a Browser Access application that resolves to an optimum BA Exporter;
- **BA Crypto Store** - the Browser Access key store for the BA certificate private keys hosted on the Amazon Key Management Service (KMS).

Slide 18 - ZPA Browser Access – Architecture

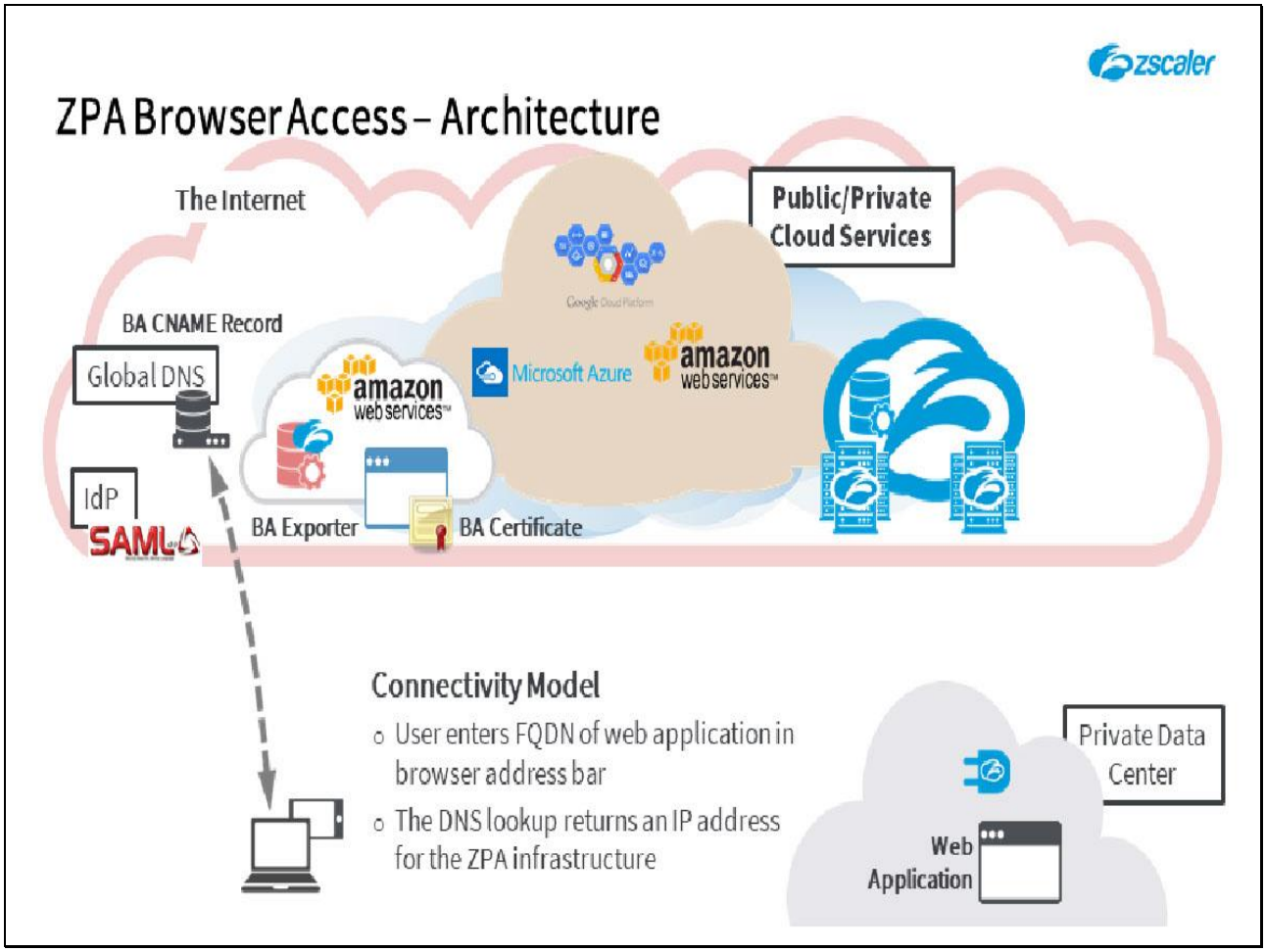


Slide notes

Prerequisite steps that are required to enable BA for an application are as follows:

1. A BA certificate for one (or more) applications must be uploaded to the ZPA infrastructure;
2. A CNAME record for each individual BA application must be defined on the DNS infrastructure (usually on the external public DNS), to allow the resolution of the FQDN for an application to the ZPA infrastructure.

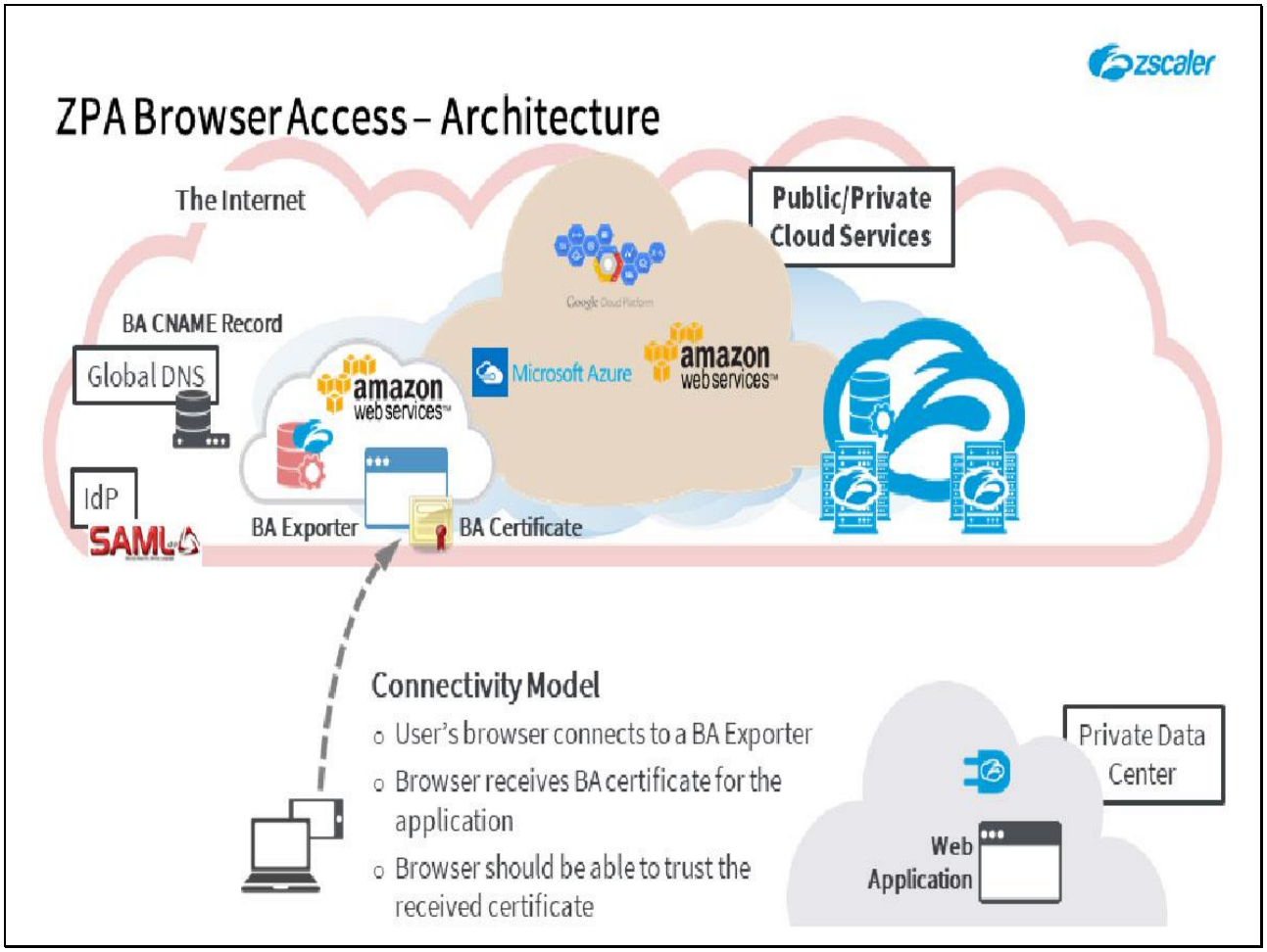
Slide 19 - ZPA Browser Access – Architecture



Slide notes

The user requests access to the private web application, by entering the FQDN for it in the browser address bar. The user's system will do a DNS lookup which will resolve to the CNAME, at which point the DNS server will look up the matching A record, then resolve that to an IP address. Ultimately, the user's system will receive an IP for the optimum BA Exporter to use to access the requested application.

Slide 20 - ZPA Browser Access – Architecture

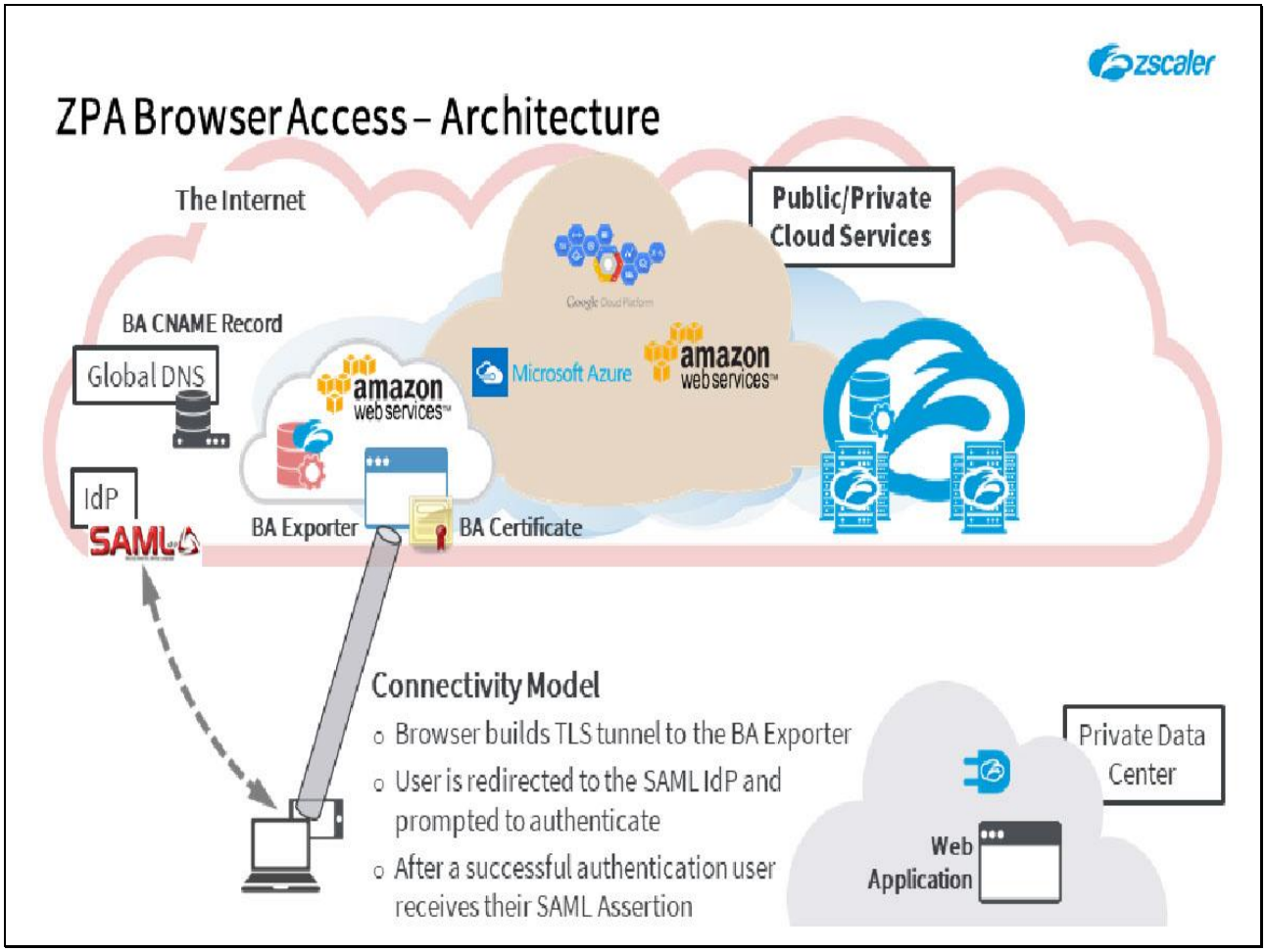


Slide notes

The user's browser connects to the BA Exporter that the DNS record resolves to and if necessary is redirected to HTTPS. It then receives the BA certificate for the requested application. The user's device should be in possession of the appropriate root CA certificate to allow it to trust the connection to the Exporter.

Note: Under most circumstances the root CA will be public, and the device will already have the required root CA certificate.

Slide 21 - ZPA Browser Access – Architecture

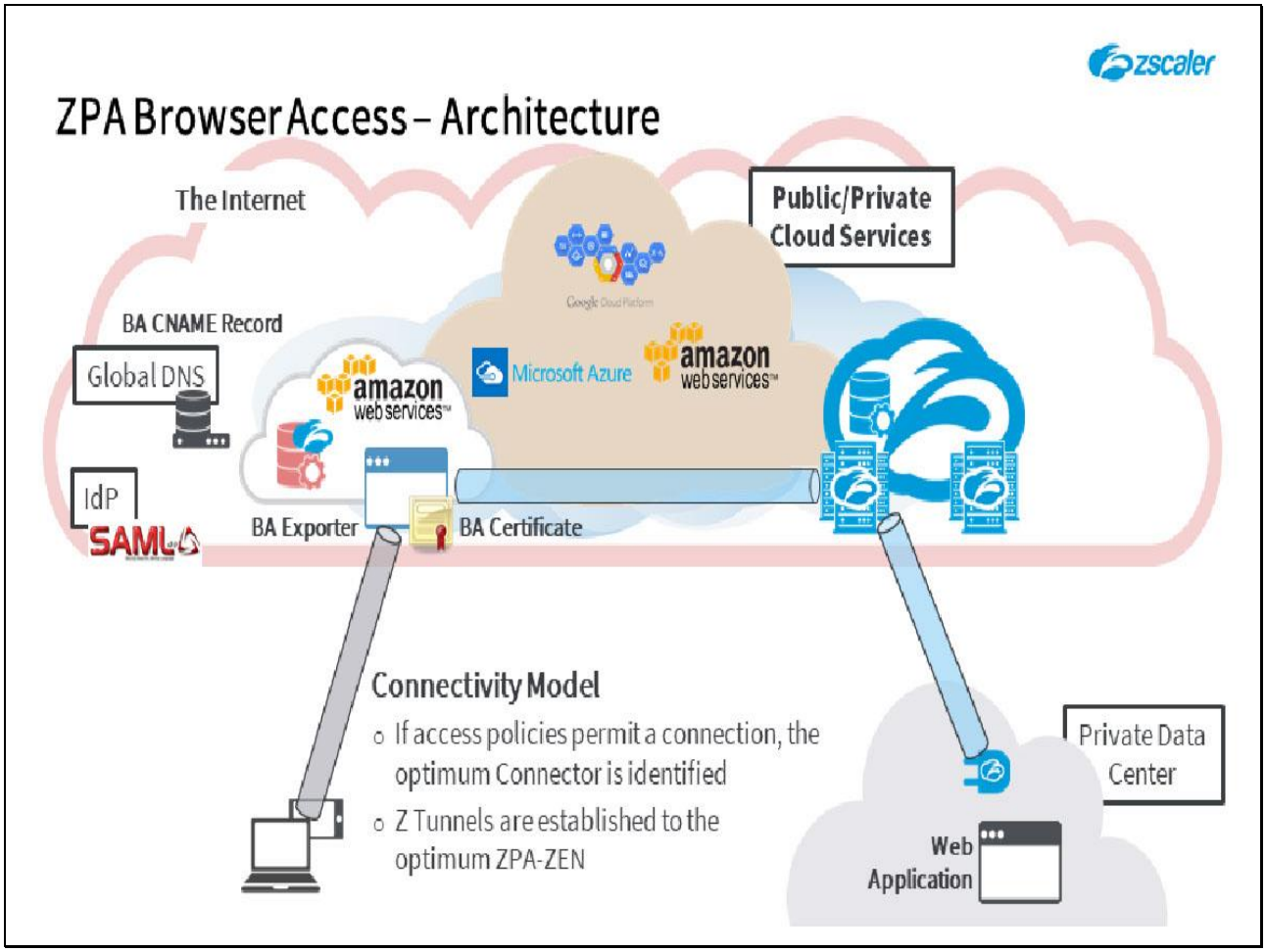


Slide notes

The browser establishes a TLS connection to the BA Exporter, which then triggers a user authentication using SAML and the configured IdP. The user will be prompted to login at this point and, after a successful authentication, will receive a SAML assertion.

Note: This is the exact same authentication process as for ZPA with the Zscaler App. It may use the exact same SAML settings, or a separate SAML IdP configuration.

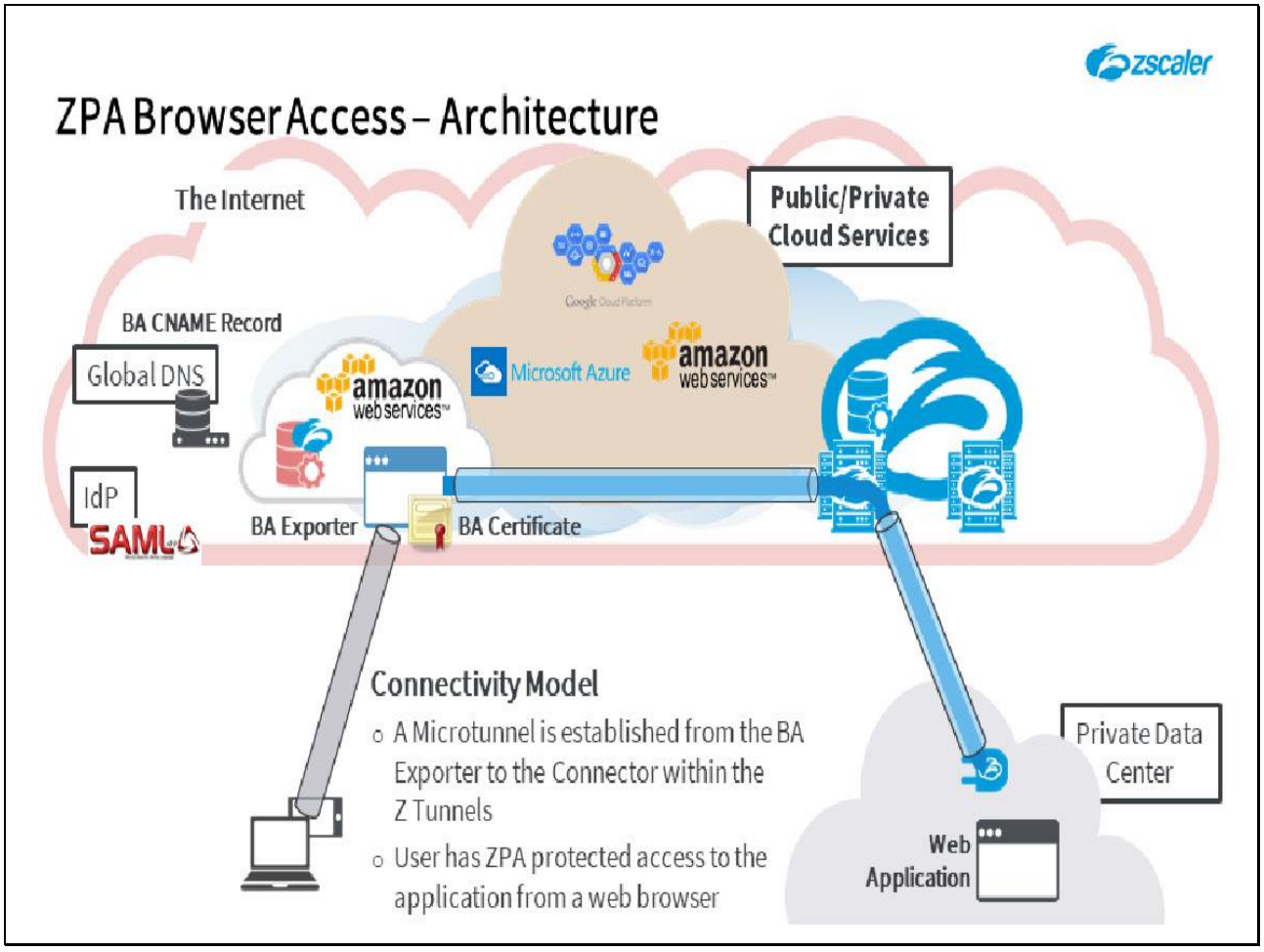
Slide 22 - ZPA Browser Access – Architecture



Slide notes

If the access policy configuration permits this user to connect to the requested application, ZPA identifies the optimum ZPA-ZEN and the best-path Connector to access it. Z Tunnels are established from the BA Exporter and the chosen Connector to the ZPA-ZEN. Note that the ZPA-ZEN will in most cases be immediately adjacent to the BA Exporter.

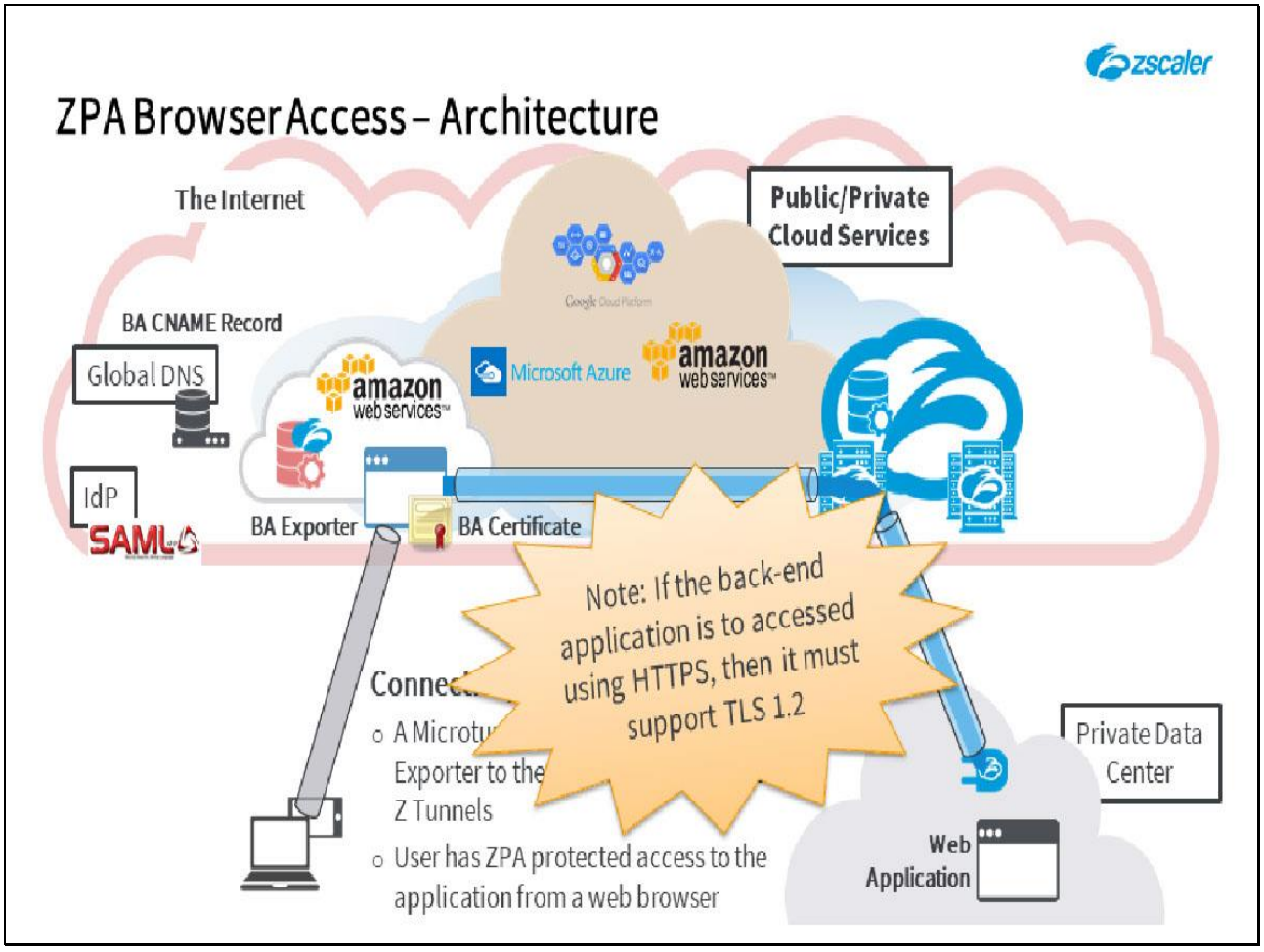
Slide 23 - ZPA Browser Access – Architecture



Slide notes

Subsequently a Microtunnel is established through the Z Tunnels to provide an encrypted end-to-end data path for the user to access the private web-based application. Data is encrypted between the user's browser and the BA Exporter using HTTPS, then it is again protected by TLS in the Z Tunnels within the ZPA infrastructure. As is always the case, the Microtunnel is established on a per-user and per-application basis; it cannot be used by another user, nor for accessing some other application.

Slide 24 - ZPA Browser Access – Architecture



Slide notes

Note that, when enabling BA for an application, you have the option to specify whether to use HTTP or HTTPS for the BA connection from the Connector to the application itself. If you elect to use HTTPS, then the back-end web applications must be implemented using TLS 1.2.

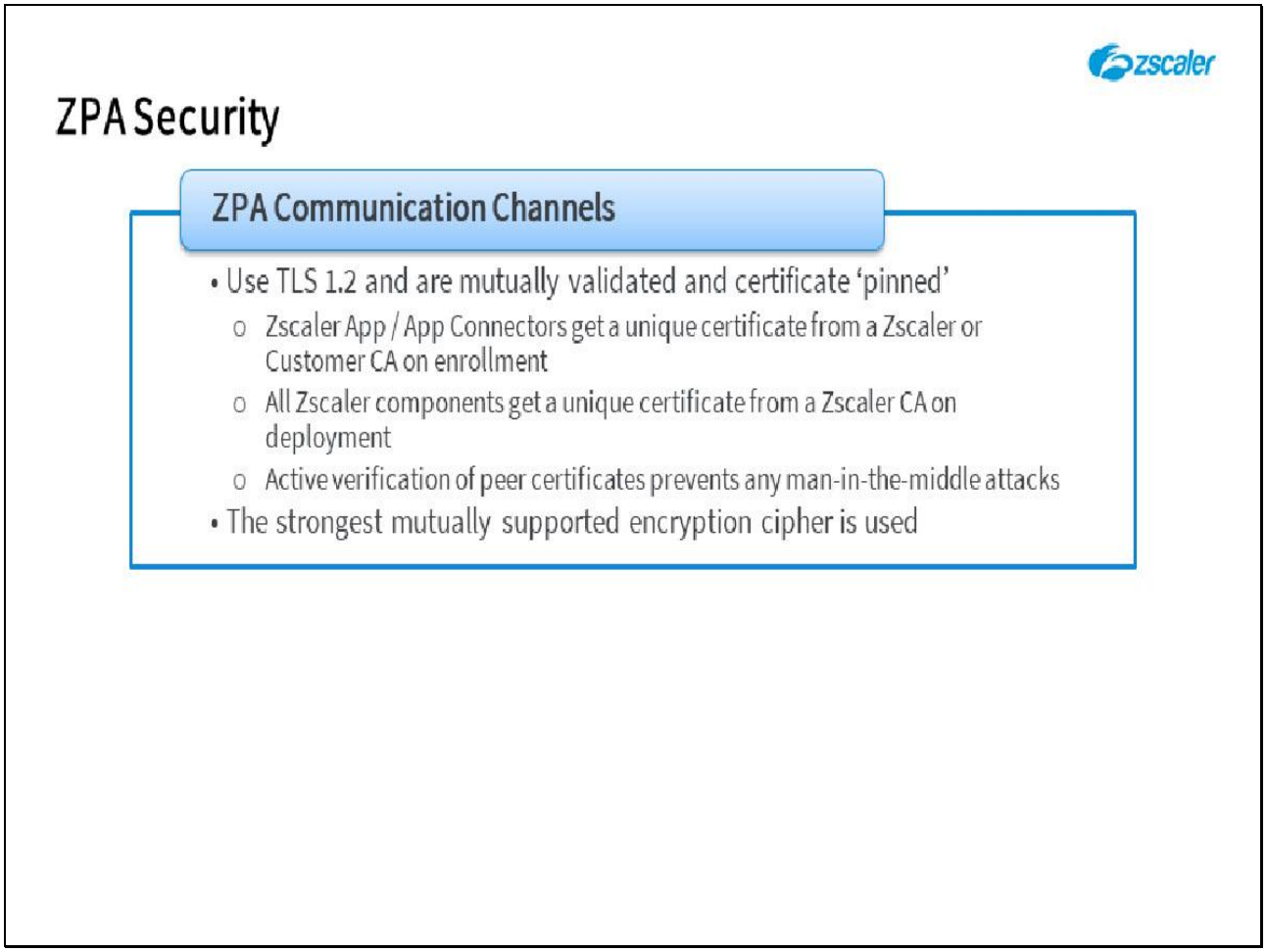
This is because the entity requesting the page on behalf of the end user is the BA Exporter, which acts as a full reverse proxy. The BA Exporter has been implemented to comply with the same level of TLS security as the rest of the system, meaning compliance with the TLS 1.2 standard using modern, strong cipher suites and crypto.

Slide 25 - ZPA Security

Slide notes

Let’s now have a look at how ZPA security works.

Slide 26 - ZPA Security



Slide notes

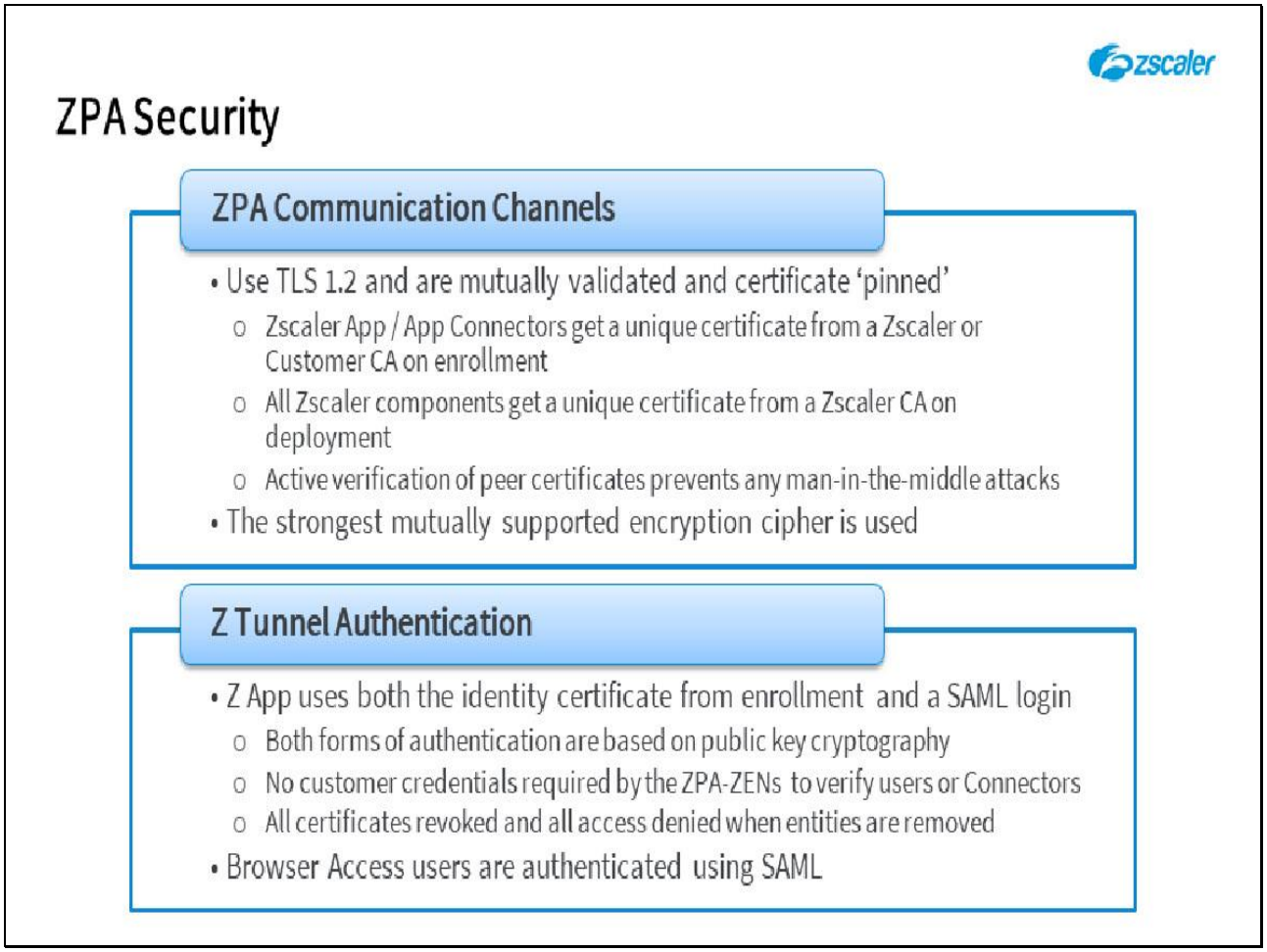
All ZPA communication channels use TLS 1.2 and are mutually validated and certificate pinned, meaning that mutual certificate validation is required before the Z Tunnels, or double encrypted Microtunnels can be established.

All customer-hosted components, which means the Zscaler App deployed to client devices and the Connectors, receive a unique certificate signed either by a Zscaler, or Customer CA when they first enroll with the infrastructure.

All Zscaler infrastructure components get a unique certificate from a Zscaler CA when they are deployed. The active verification of peer certificates prior to connection prevents any possibility of man-in-the-middle attacks against the Z Tunnels.

The encryption cipher used is the strongest that is mutually supported by the communicating components.

Slide 27 - ZPA Security



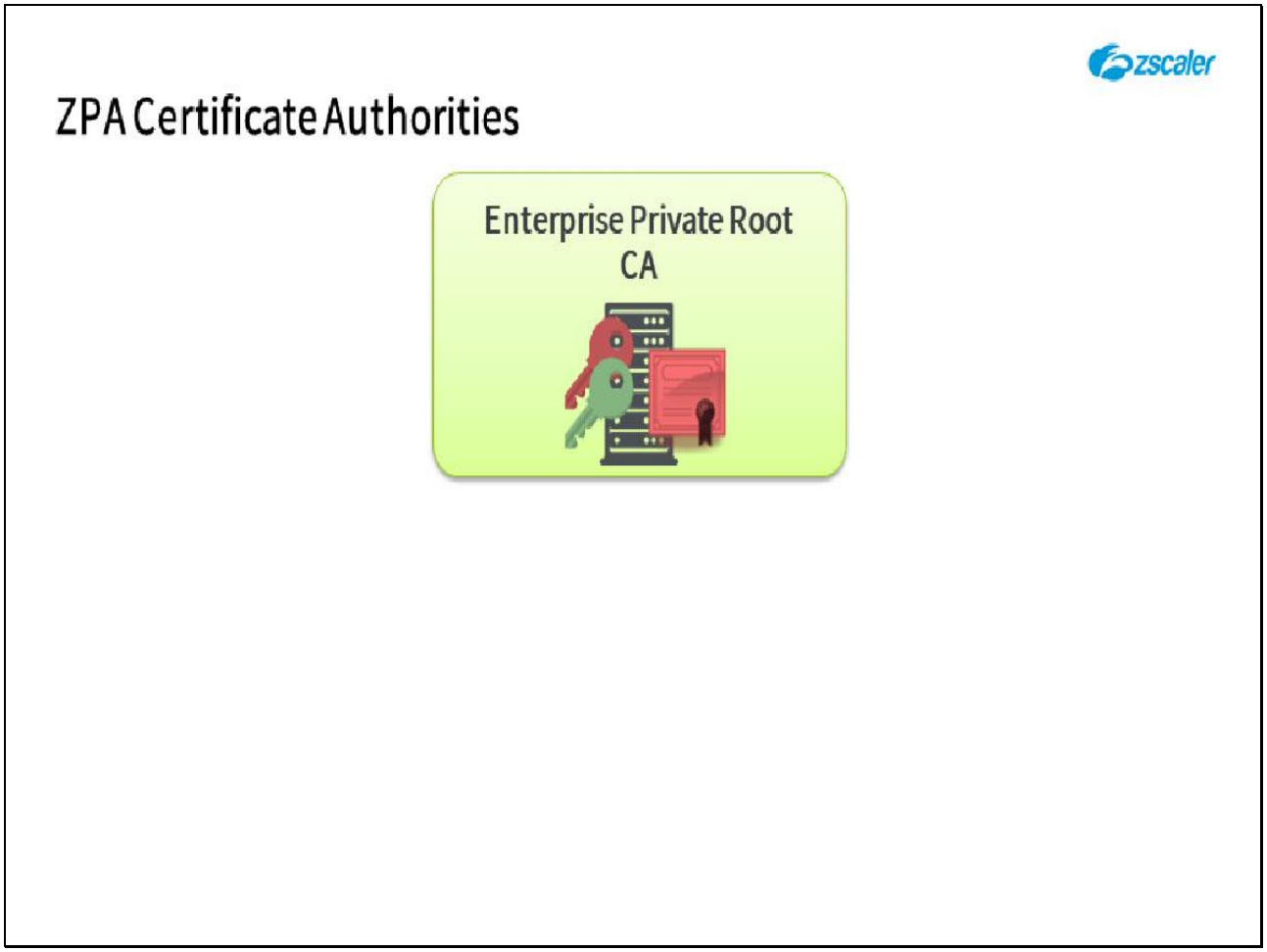
Slide notes

Z Tunnels from the Zscaler App use a combination of the identity certificate (from enrollment) and the user's SAML assertion for validation, both forms of authentication being based on public key cryptography. As a result, the ZPA-ZENs do not need to have any private customer credentials in order to verify authenticity of end users or Connectors.

The keys generated by, and the certificate received by, the Zscaler App are securely stored within the App. Should an administrator disable ZPA functionality for a device through a **Force Remove** operation at the Zscaler App Portal, these issued certificates are immediately revoked.

The ZPA-ZENs check all entities (Connectors and Zscaler App) for the presence of valid certificates every four minutes. If one is not found, connectivity to private applications using ZPA is disabled.

Slide 28 - ZPA Certificate Authorities



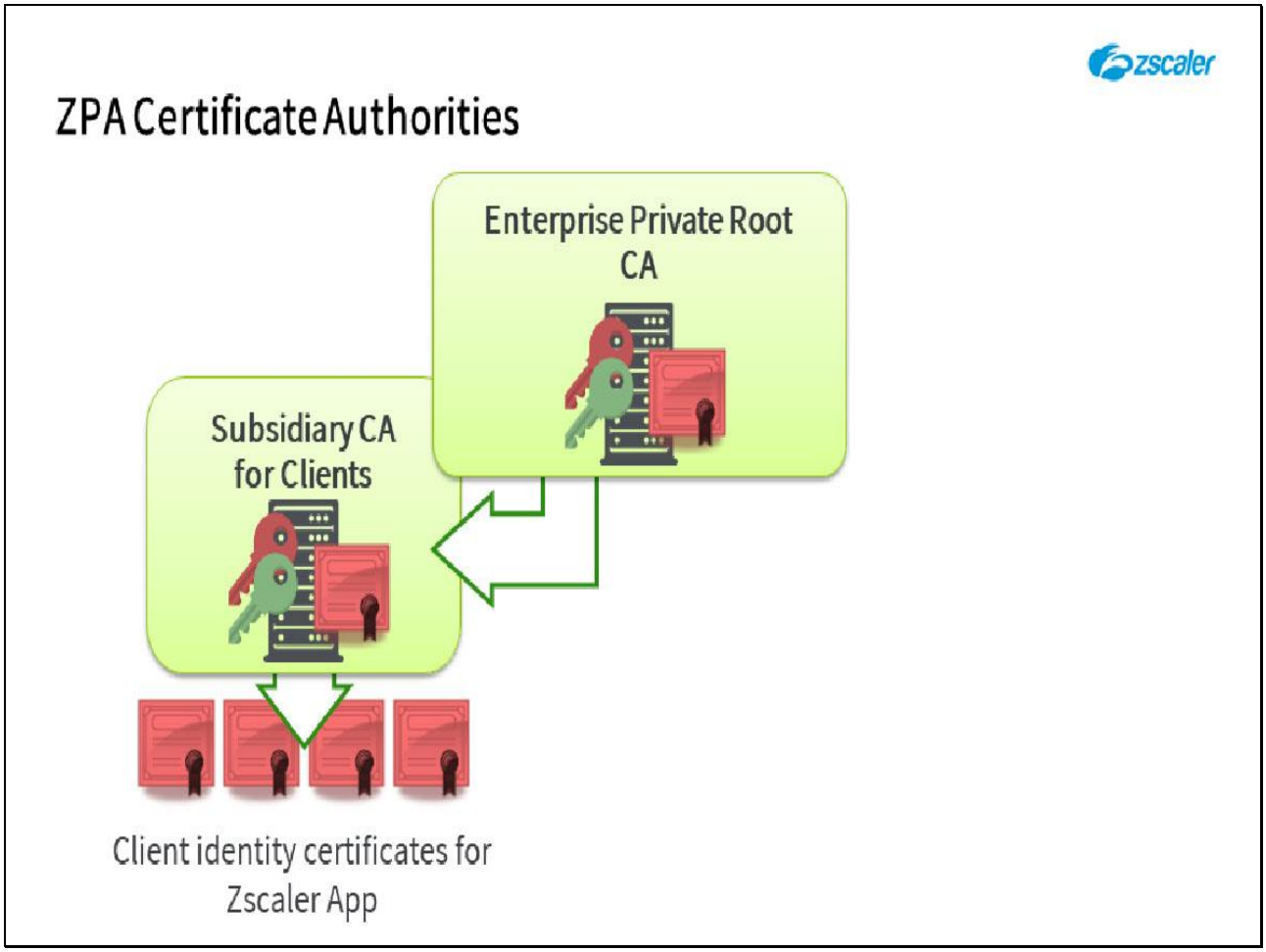
Slide notes

ZPA requires a set of Certificate Authorities to issue and sign the certificates required by the various entities (Zscaler App and Connectors). The CAs may be self-signed, i.e. a stand-alone Root CA with subordinates; or they may be an extension of the customer's internal PKI.

A default set of self-signed CAs are generated by Zscaler on the initialization of every new ZPA tenant, consisting of:

- A self-signed Root CA with its associated keys; ...

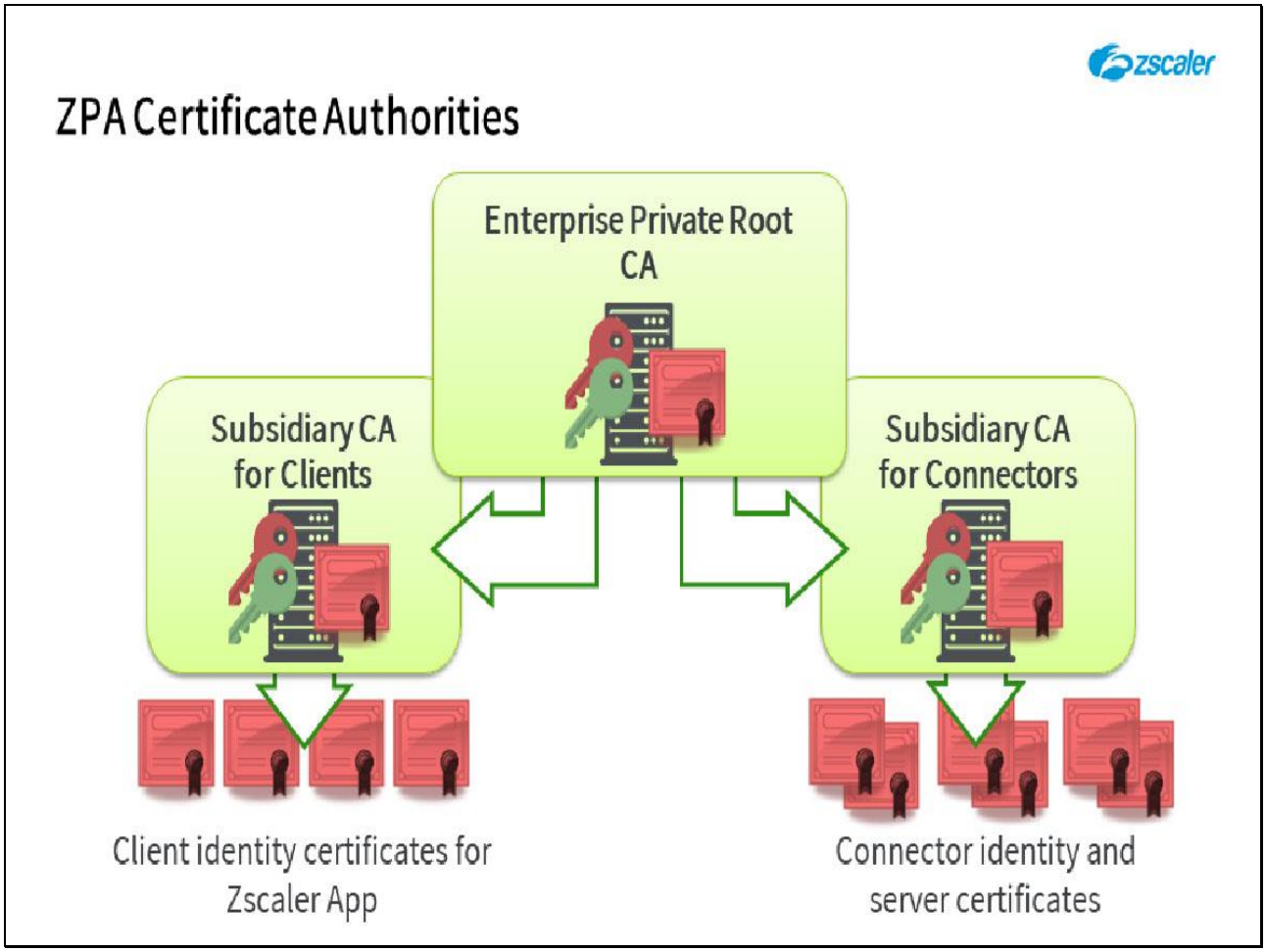
Slide 29 - ZPA Certificate Authorities



Slide notes

- ...a subordinate (aka subsidiary or intermediate) CA for issuing the identity certificates required by the Zscaler App users; ...

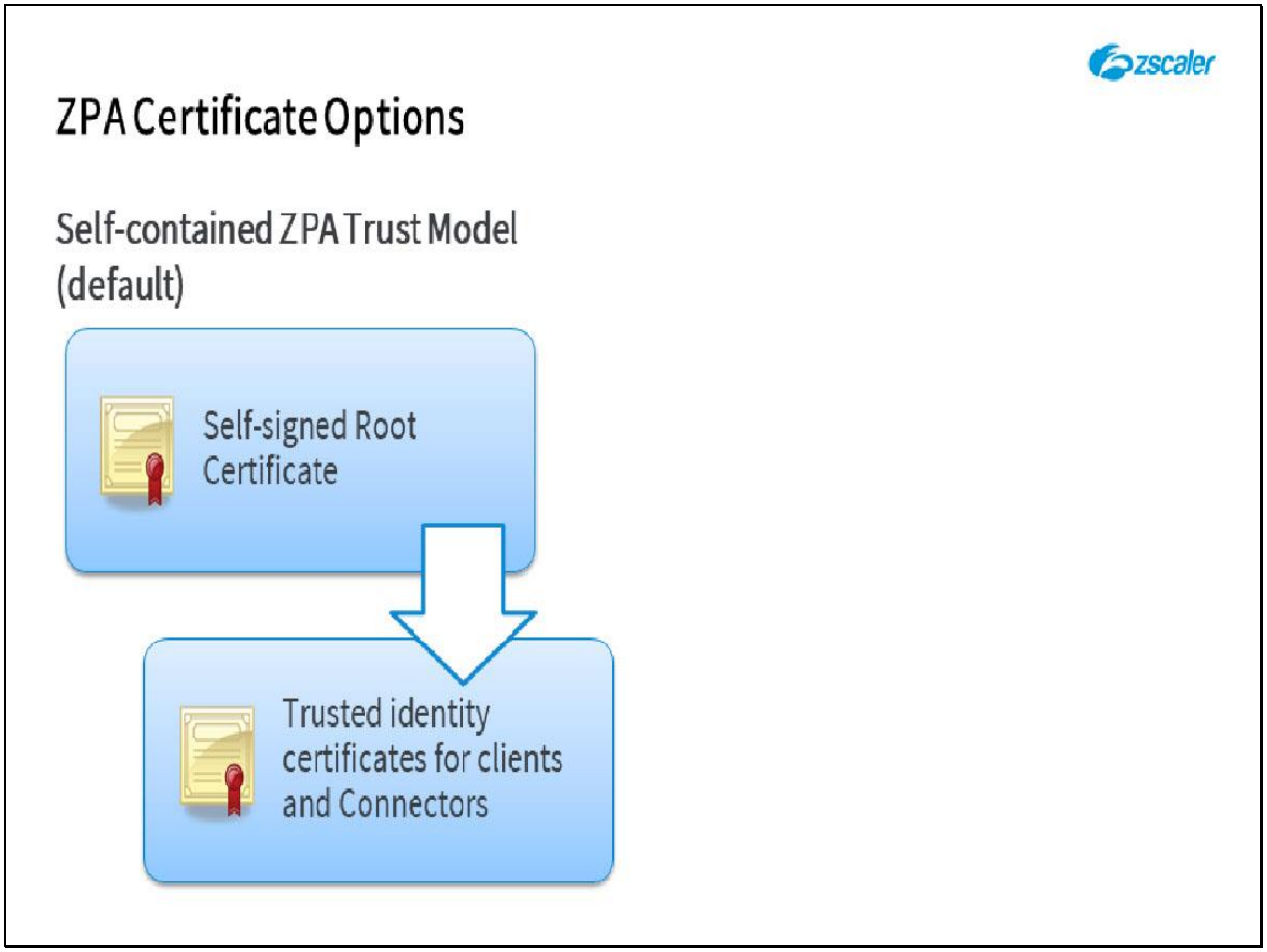
Slide 30 - ZPA Certificate Authorities



Slide notes

- ...and a sub-CA for issuing the identity and server certificates required by the Connectors.

Slide 31 - ZPA Certificate Options



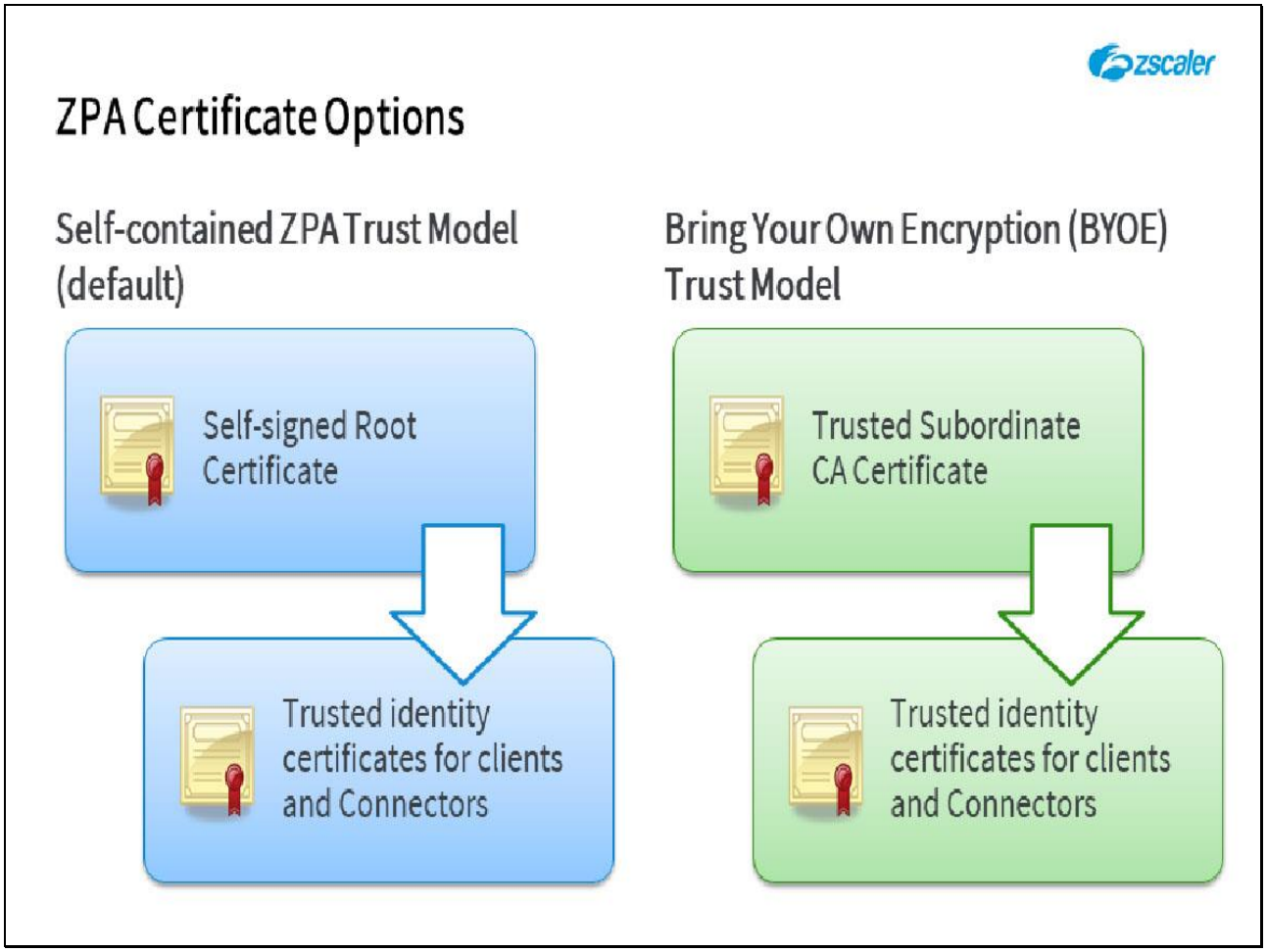
Slide notes

Let’s talk about the trust model options for ZPA, of which there are two: Firstly, you have the option within the ZPA Admin Portal to generate a new self-signed root CA for your ZPA instance, and subsequently generate the identity certificates required by the Connectors and Zscaler App clients for authentication based upon it.

Essentially you create a new stand-alone and self-contained certificate authority just to deploy the identity certificates to your ZPA infrastructure, that are then used for authenticating the Z Tunnel connections.

Note that a complete set of self-signed Root and subordinate CAs for Connectors and clients should already exist on any new ZPA tenant and can be used immediately.

Slide 32 - ZPA Certificate Options



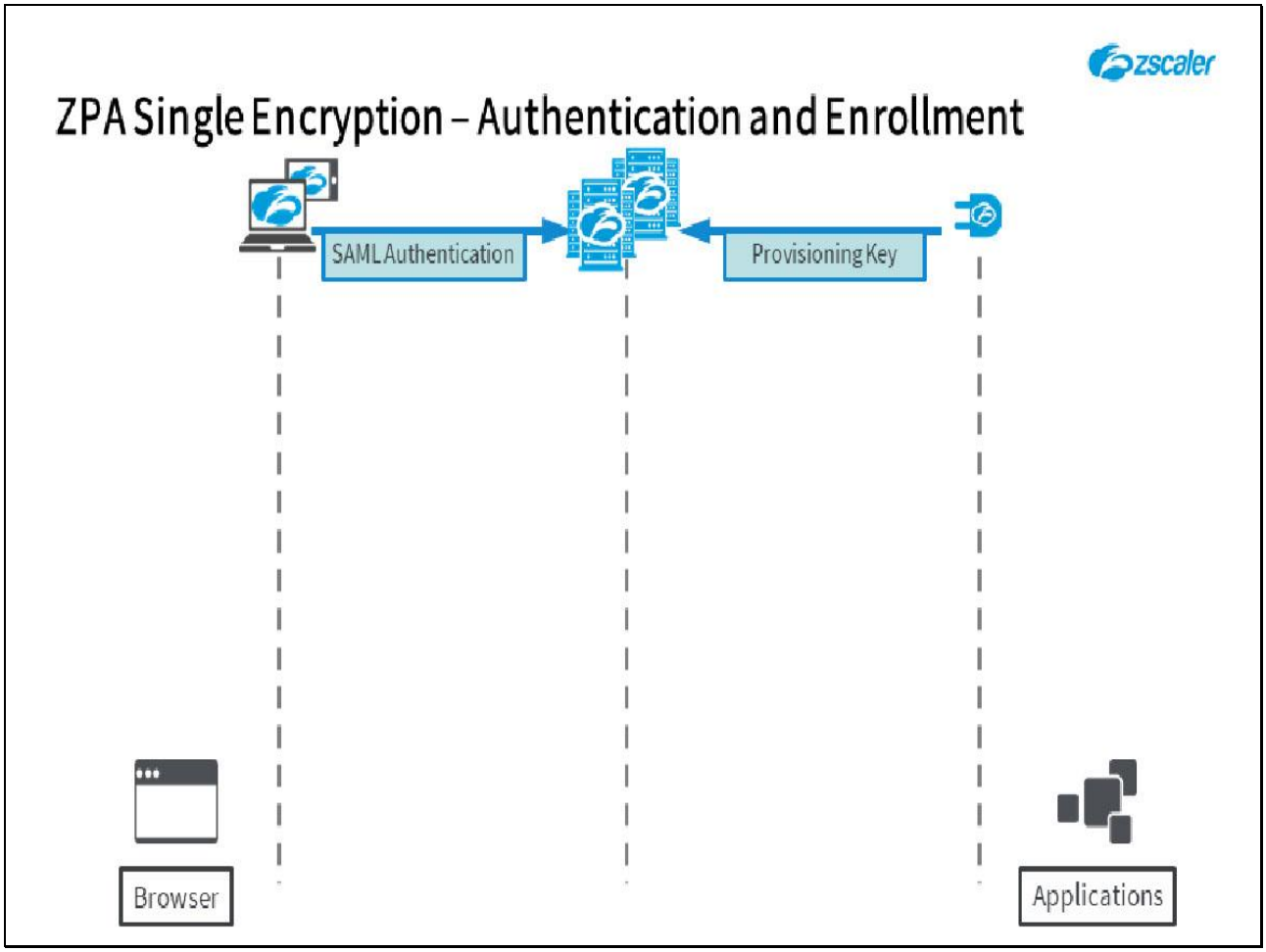
Slide notes

The 'Bring Your Own Encryption' (BYOE) option allows you to request a certificate for ZPA from your internal PKI, which then allows ZPA to act as a subordinate CA on its behalf. Note that public CAs are not suitable for this use case, as certificates purchased do not support the signing of subordinate certificates.

The issuing subordinate CAs for Connectors and clients can then be generated from this trusted subordinate CA. This provides a trust chain all the way back to the internal trusted root CA, although note that this option does require you to load all the intermediate certificates in the trust chain.

All certificates that are part of the ZPA system, including all customer-operated systems, have private keys that never leave the physical device in which they were generated.

Slide 33 - ZPA Single Encryption

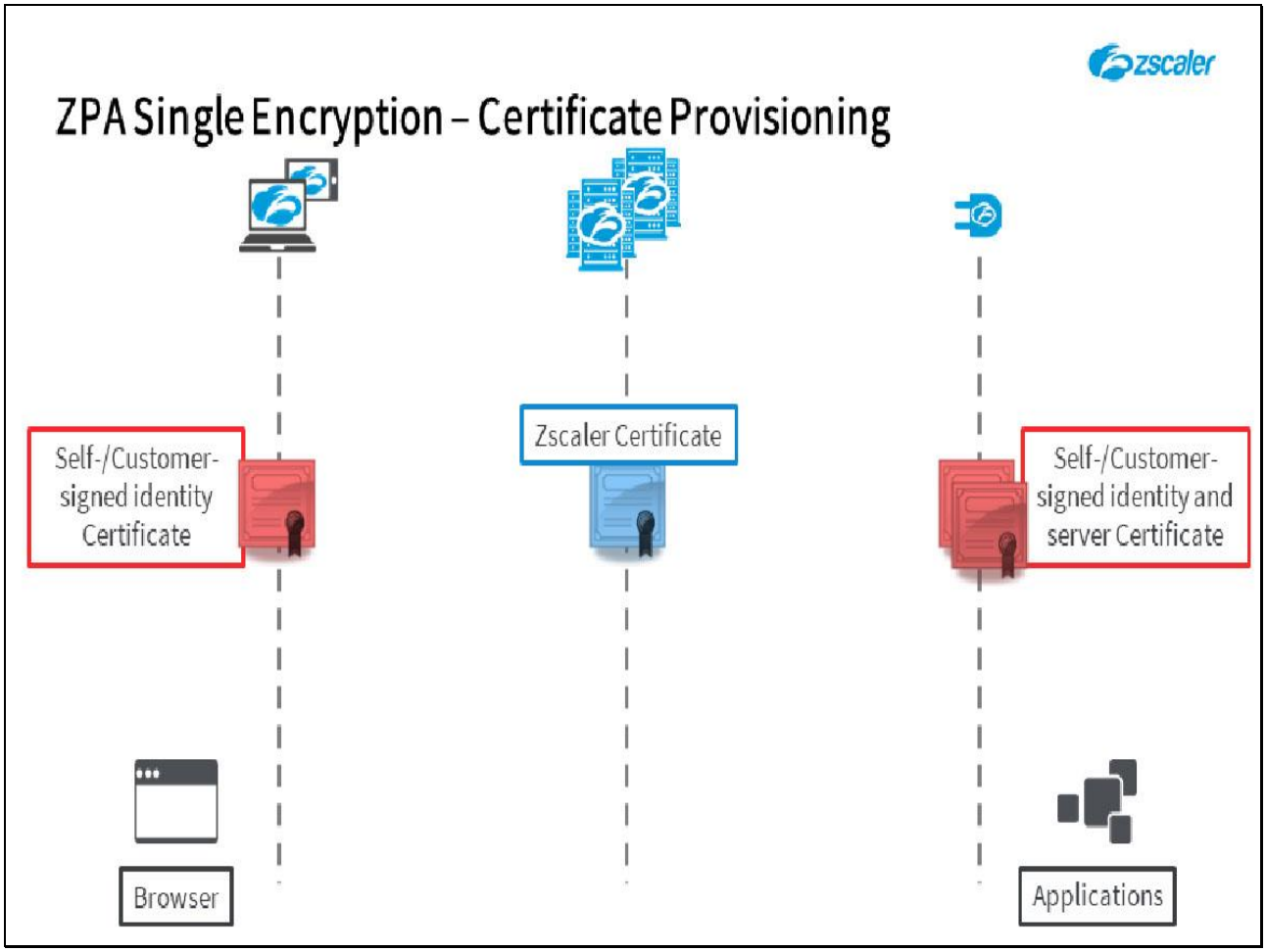


Slide notes

Let’s now step through the process for establishing the tunnels that ZPA requires to ensure secure access to your private applications.

The first step of the process is for the various entities to securely enroll into your ZPA tenant. This requires Z App and BA users to authenticate using SAML and Connectors to present a valid Provisioning Key.

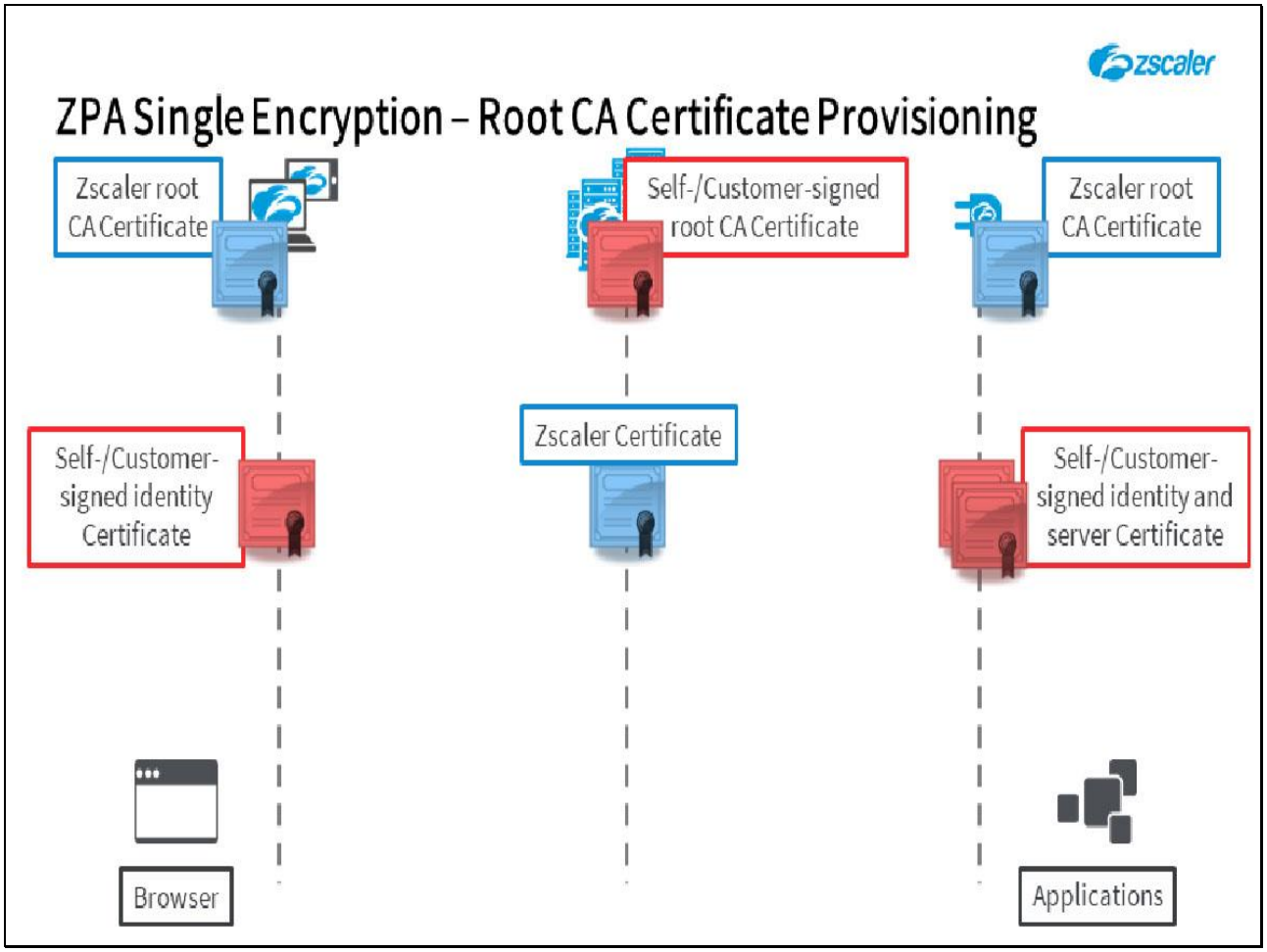
Slide 34 - ZPA Single Encryption



Slide notes

Z Tunnels are built using the set of certificates deployed to the Z App users and Connectors on enrollment. These end points may receive either a self-signed or customer-signed identity certificate, depending on customer preference and configuration, while the ZPA-ZENs are provisioned with a Zscaler-signed server certificate.

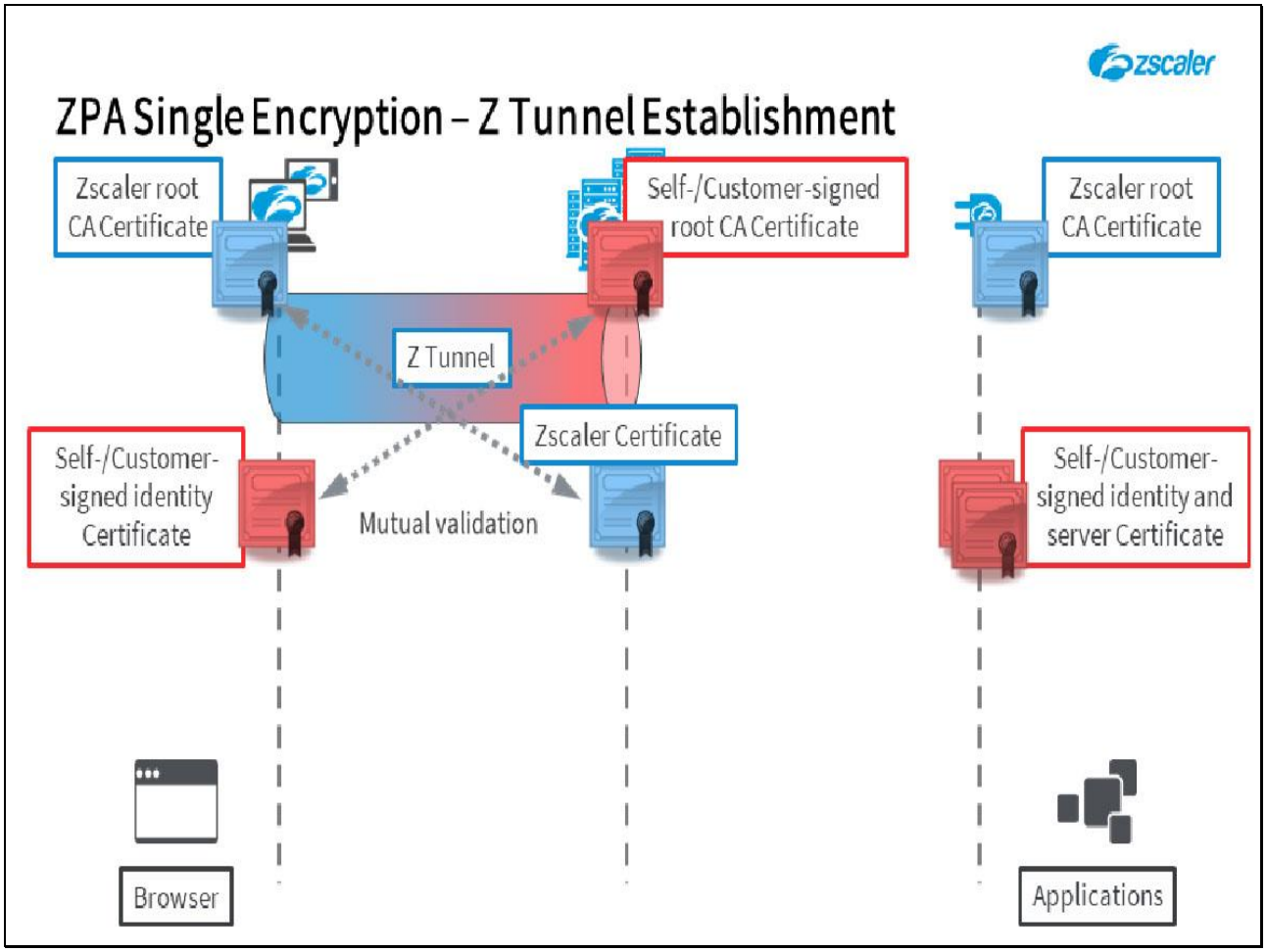
Slide 35 - ZPA Single Encryption



Slide notes

In addition, the end points and the ZPA-ZENs are simultaneously provisioned with the Root CA certificates that they will require in order to do a mutual certificate validation; the Zscaler Root CA certificate for Z App users and Connectors, while the ZPA-ZENs are provisioned with the appropriate customer Root CA certificate (whether self- or customer-signed).

Slide 36 - ZPA Single Encryption



Slide notes

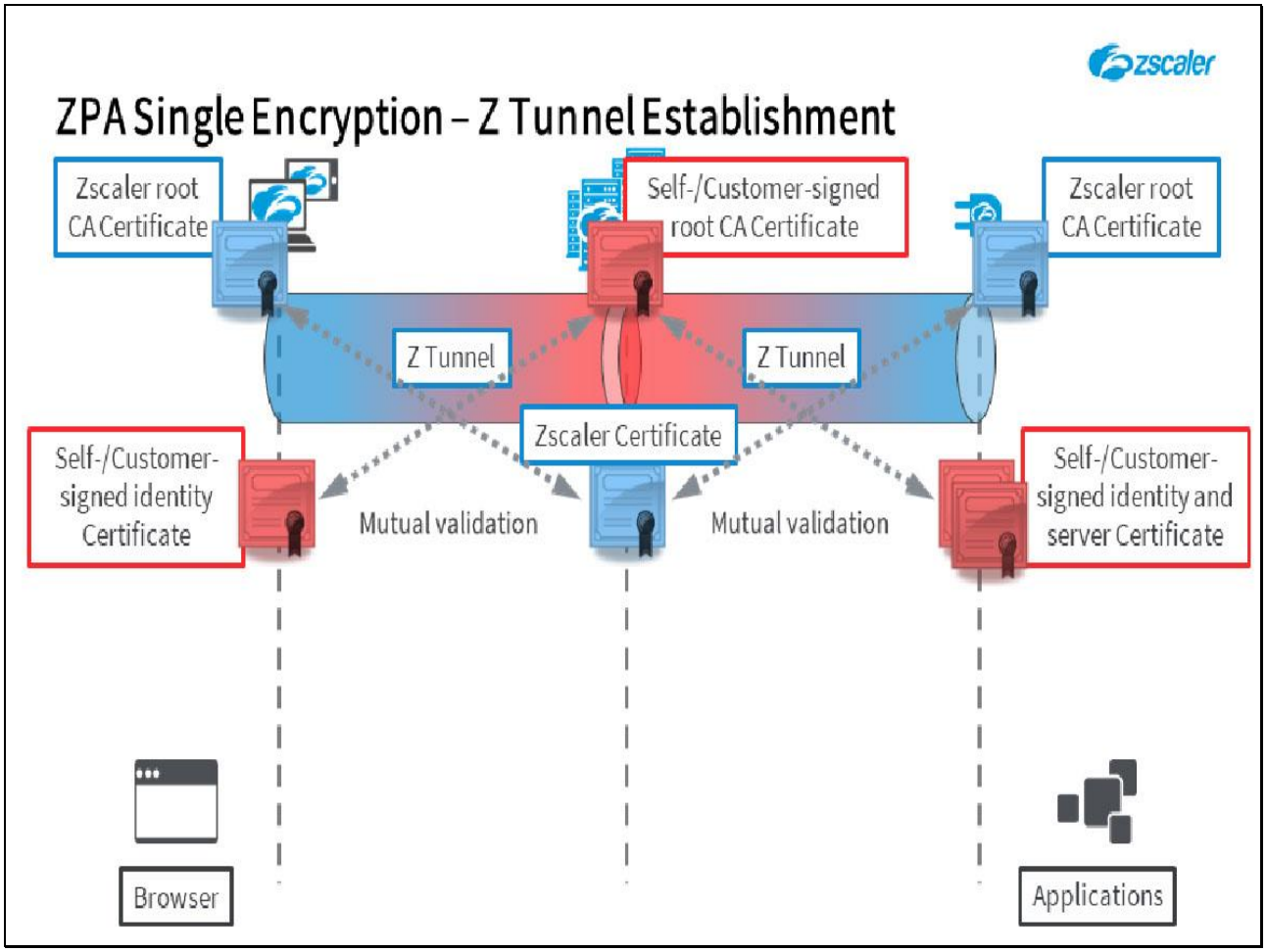
When an authenticated and authorized user requests access to a private application, this will trigger the establishment of the required tunnels. Z Tunnels are only ever established in the outbound direction and are mutually validated using the deployed certificates.

For the Zscaler App the tunnel is established in the outbound direction to the nominated ZPA-ZEN using the standard TLS 1.2 tunnel setup negotiation on port 443. As the App possesses the Zscaler root CA certificate, it can validate the server certificate it receives from the ZEN during tunnel establishment.

Similarly, as the customer has previously created or installed a root CA certificate through the ZPA-CA, the Zscaler infrastructure can validate the identity certificate presented by the App.

Z Tunnels use mutual validation with double-pinning; if either end of the conversation receives an invalid certificate, the tunnel will not be established. As a result, Z Tunnels are immune to Man-in-the-Middle attacks.

Slide 37 - ZPA Single Encryption

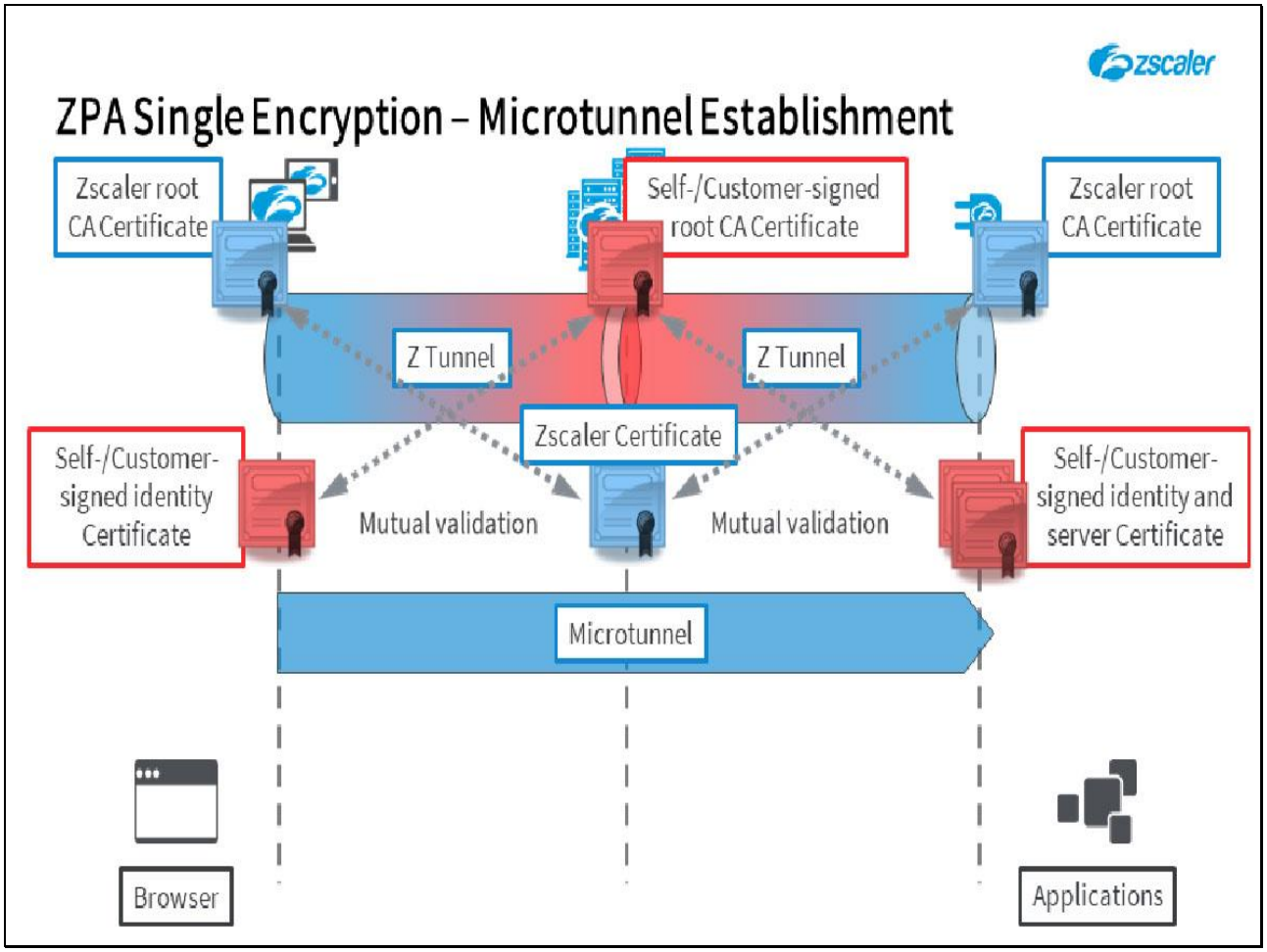


Slide notes

The Z Tunnel from the Connector is established in the exact same way, as an outbound connection from the Connector to the nominated ZPA-ZEN. Once again mutual certificate validation is done with double-pinning, with the Connector presenting the identity certificate it received during enrollment.

Z Tunnels are encrypted using the strongest cipher that is mutually supported by the Zscaler App and Connector hosts at one end, and the ZPA-ZENs at the other.

Slide 38 - ZPA Single Encryption

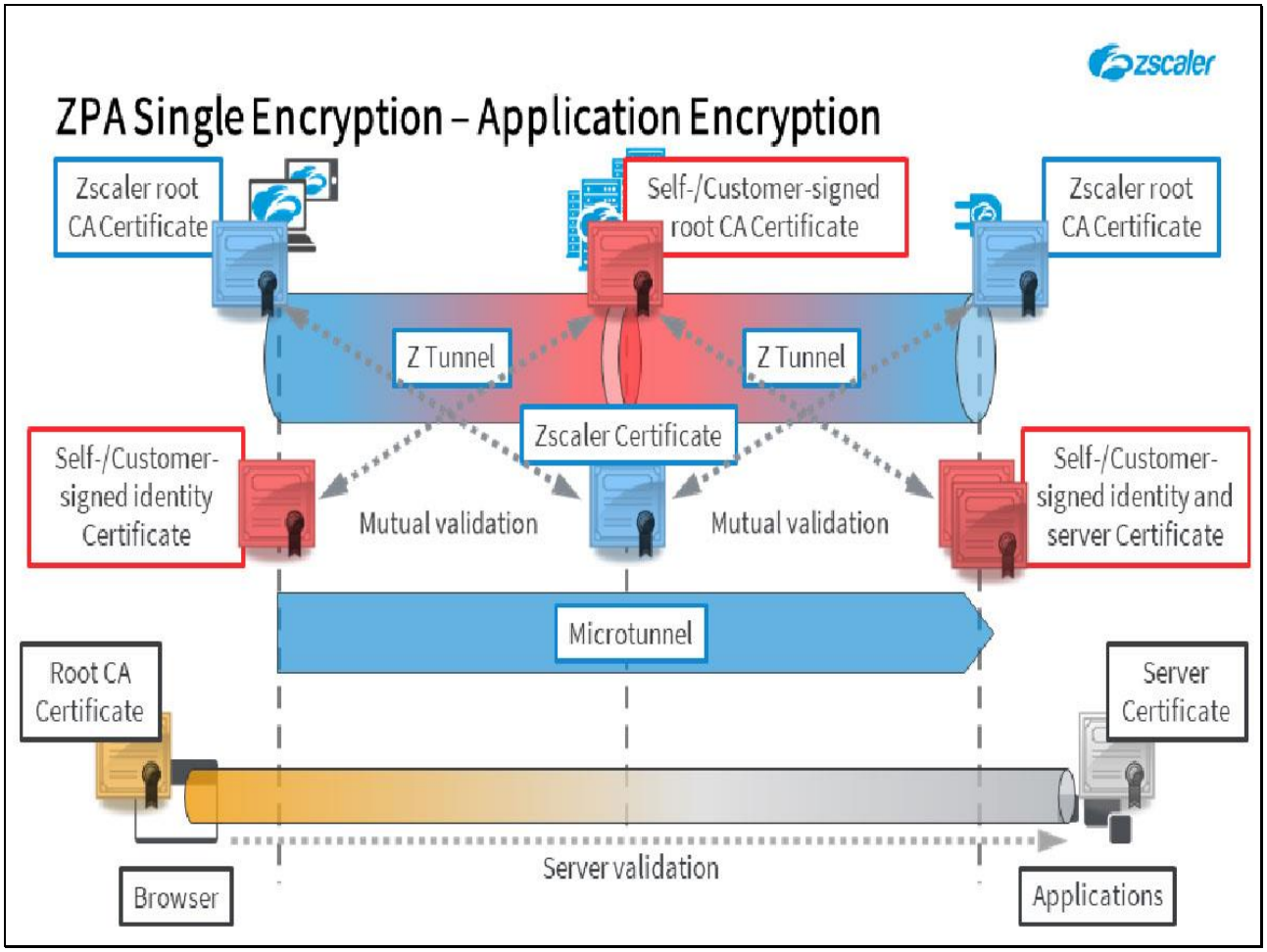


Slide notes

Once the Z Tunnels are in place, traffic can be securely transferred as a byte stream from the Zscaler App to the Connector, based on source and destination tags that are generated dynamically during the connection establishment process. This byte stream connection is referred to as the end-to-end Microtunnel.

Note that under normal circumstances, the Microtunnel is not separately encrypted, which means that in principal Zscaler could view and scan traffic as it transits the ZPA-ZEN (although this is not done currently).

Slide 39 - ZPA Single Encryption

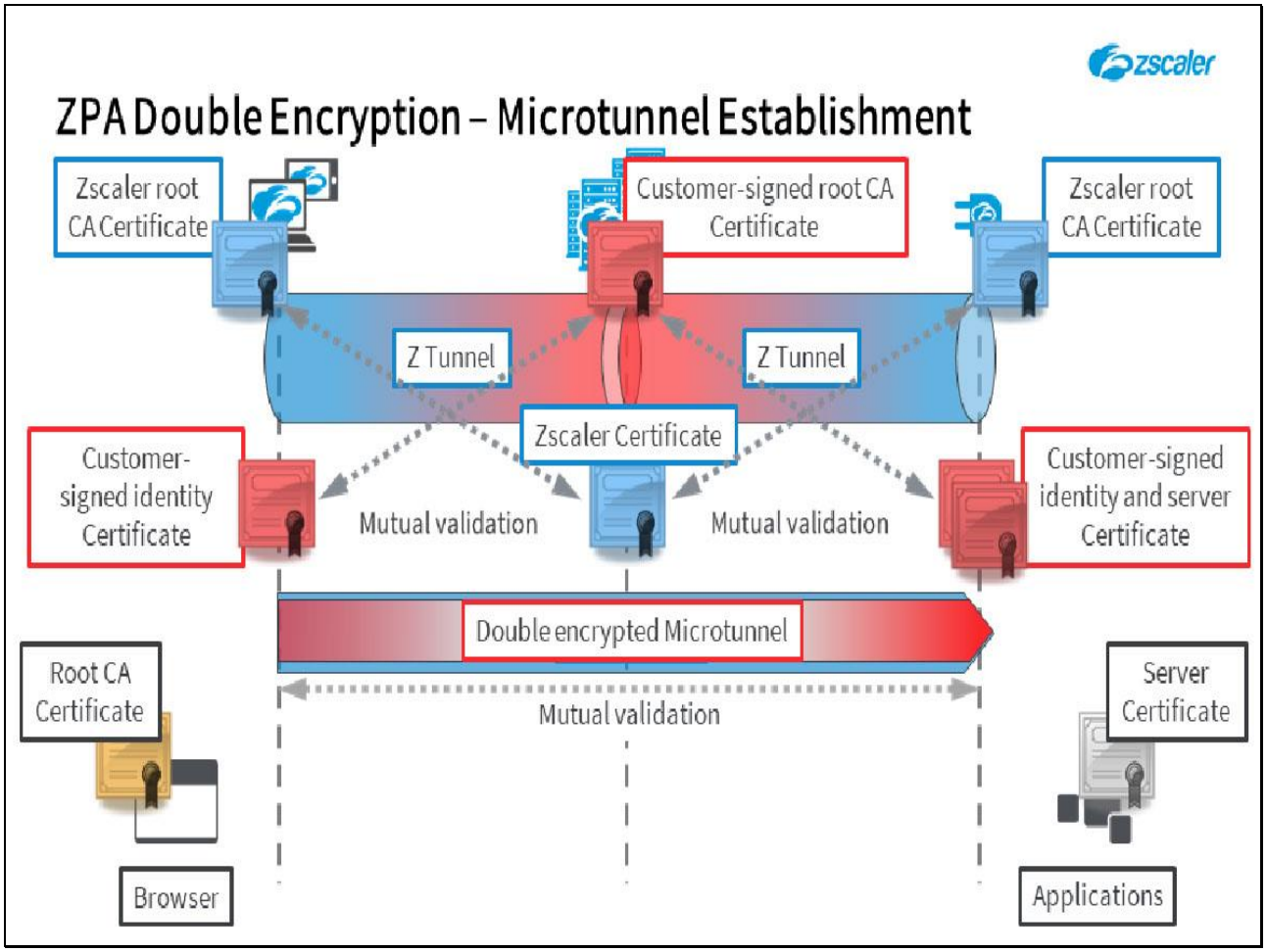


Slide notes

The end user now has a secure, encrypted, end-to-end data path that allows them to connect to the application. The browser or agent on the user's device believes it is talking to the Zscaler App on a synthetic IP address, while the application believes it is talking to the Connector. The ZPA service manages the secure transfer of data in between.

In addition to the layers of encryption provided by Zscaler, if the applications themselves use HTTPS then TLS encryption is established end-to-end within the Microtunnel based on the application's Server certificate, which may be validated at the client by the appropriate Root CA certificate.

Slide 40 - ZPA Double Encryption

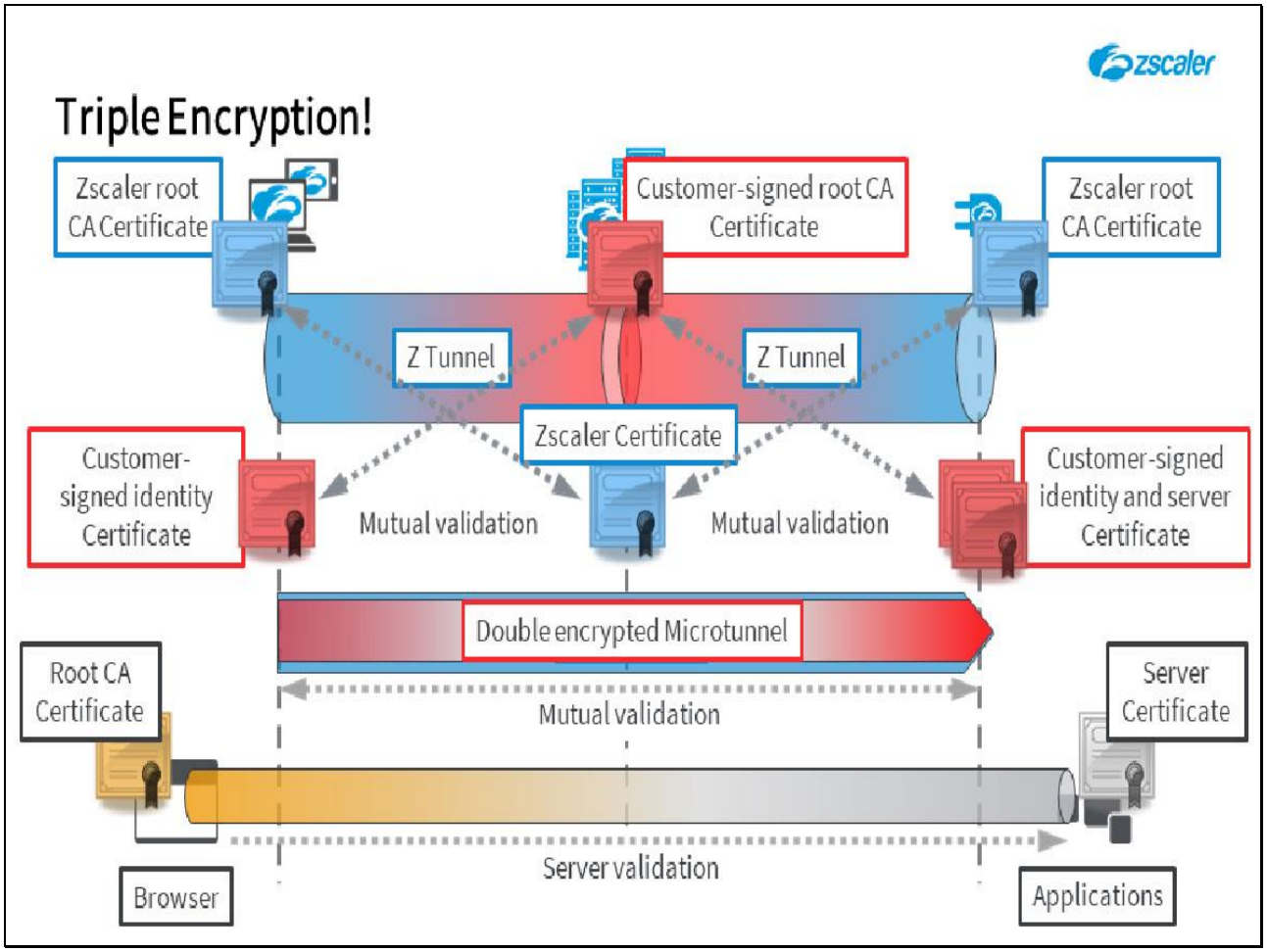


Slide notes

If the Double Encryption option is enabled for an application with the BYOE trust model, traffic is sent into the Microtunnel in an additional end-to-end TLS tunnel, which is established based on the customer-signed server certificate provisioned to the Connector.

As this is now an end point-to-end point tunnel, established with the customer's private keys, there is now no possibility whatsoever for the ZPA-ZEN to intercept or inspect the private traffic that it processes. As with the Z Tunnels, double encrypted Microtunnels are mutually validated, doubly-pinned and encrypted using the strongest cipher that is mutually supported by the Zscaler App and Connector hosts.

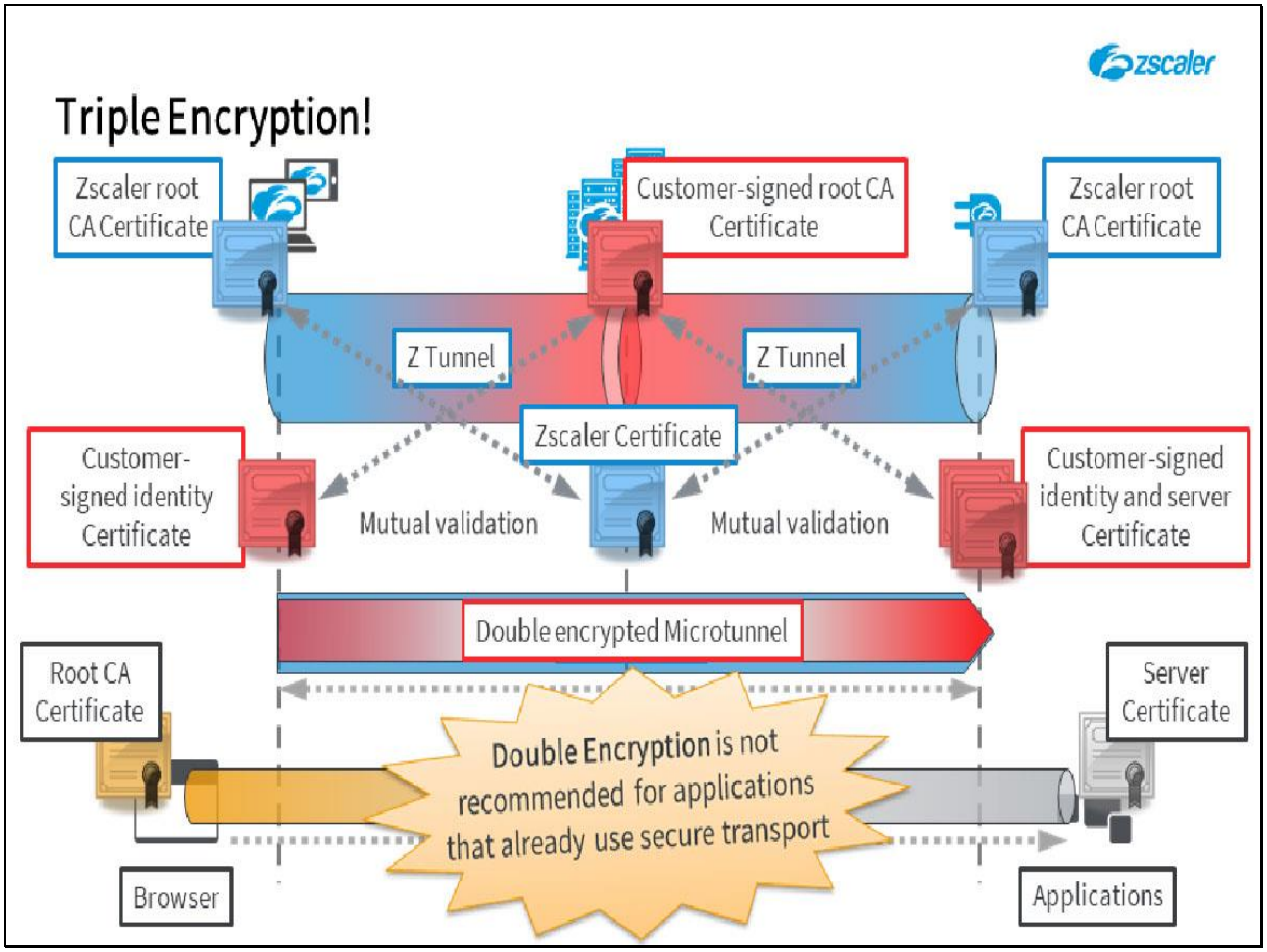
Slide 41 - Triple Encryption!



Slide notes

If the destination application is implemented using a transport protocol with some form of native encryption (e.g. HTTPS), this will mean that data will be triple encrypted as it passes across the ZPA infrastructure.

Slide 42 - Triple Encryption!



Slide notes

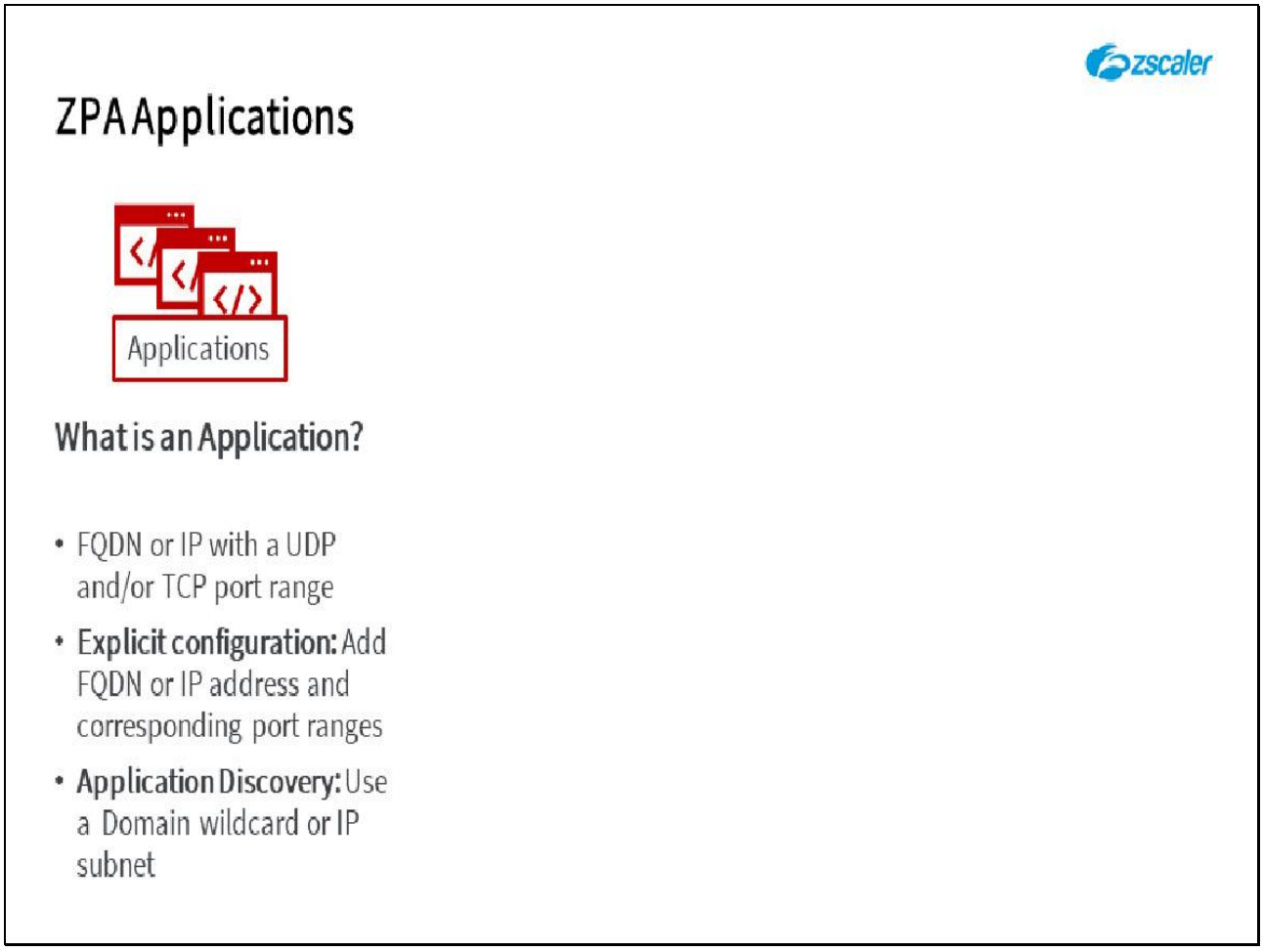
The double encryption option is not recommended for traffic that is encrypted anyway, such as secure web traffic over TLS between a web browser and an internal web application. For applications that already use encrypted transport between the client browser/agent and server, double encryption adds overhead without providing any additional real value and can cause an unnecessary reduction in capacity as well as potential MTU issues.

Slide 43 - Application Discovery

Slide notes

The next topic we will discuss is the relationship between the various logical ZPA entities.

Slide 44 - ZPA Applications



Slide notes

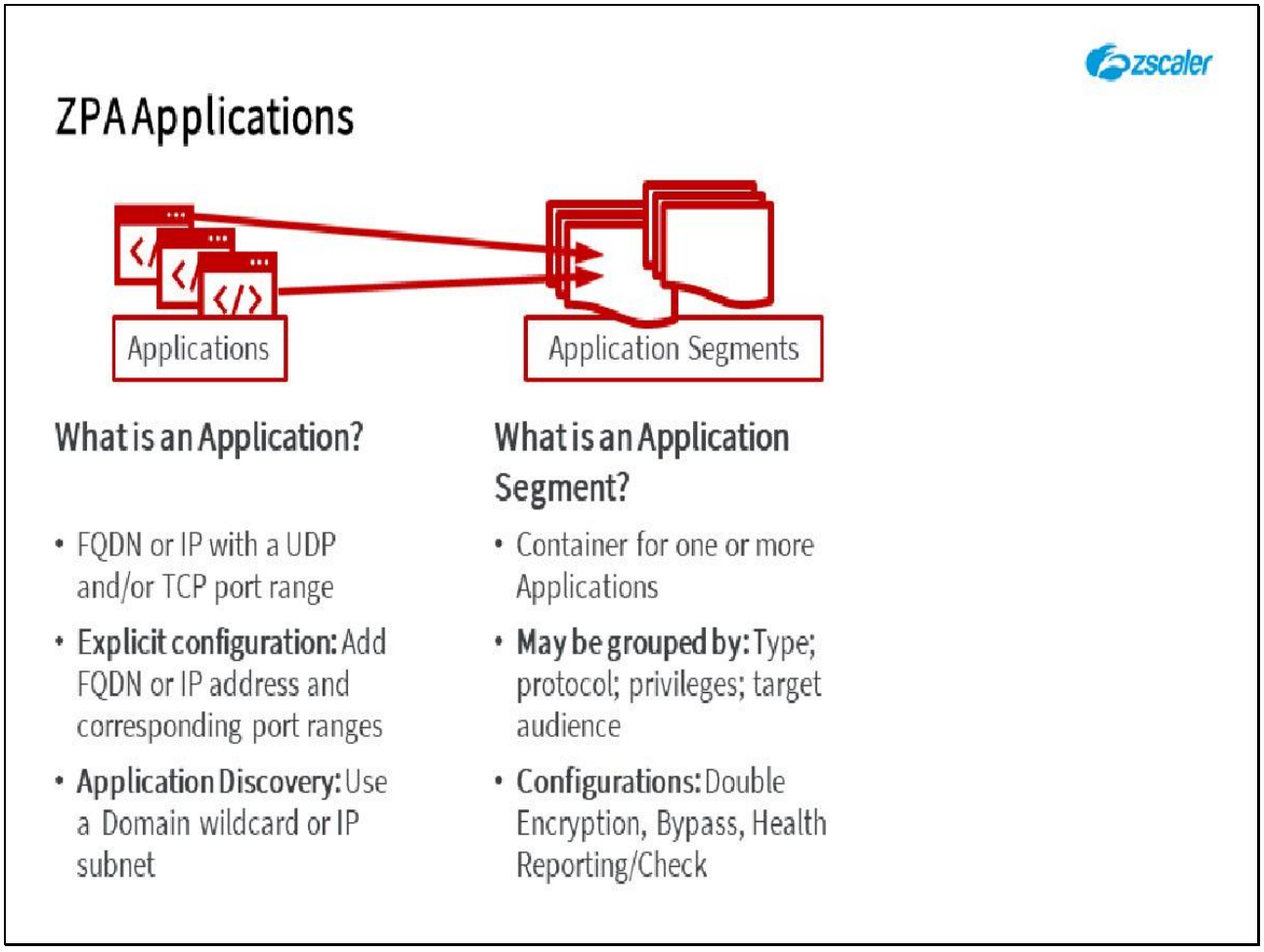
First up, let's look at the relationship between **Applications**, **Application Segments** and **Segment Groups**. What do we actually mean by an **Application** in the ZPA Admin Portal?

An **Application** is simply some combination of a FQDN or IP address, together with a TCP and/or UDP port range, e.g. intranet.patraining.safemarch.com on TCP ports 80 and 443, or 10.0.0.4 on TCP port 22.

Applications may be defined explicitly by adding a single FQDN or IP address and corresponding port ranges, as for the examples above; or, a wildcard application may be defined that can then be used for application discovery. In this case you would use a wildcard domain (e.g. *.patraining.safemarch.com), or an IP subnet. We'll talk more about application discovery in the next section.

However, you cannot create a stand-alone **Application** in the ZPA Admin Portal, an **Application** must always be a defined within an **Application Segment**. Even if you only want to add a single **Application**, it must be part of an **Application Segment**.

Slide 45 - ZPA Applications



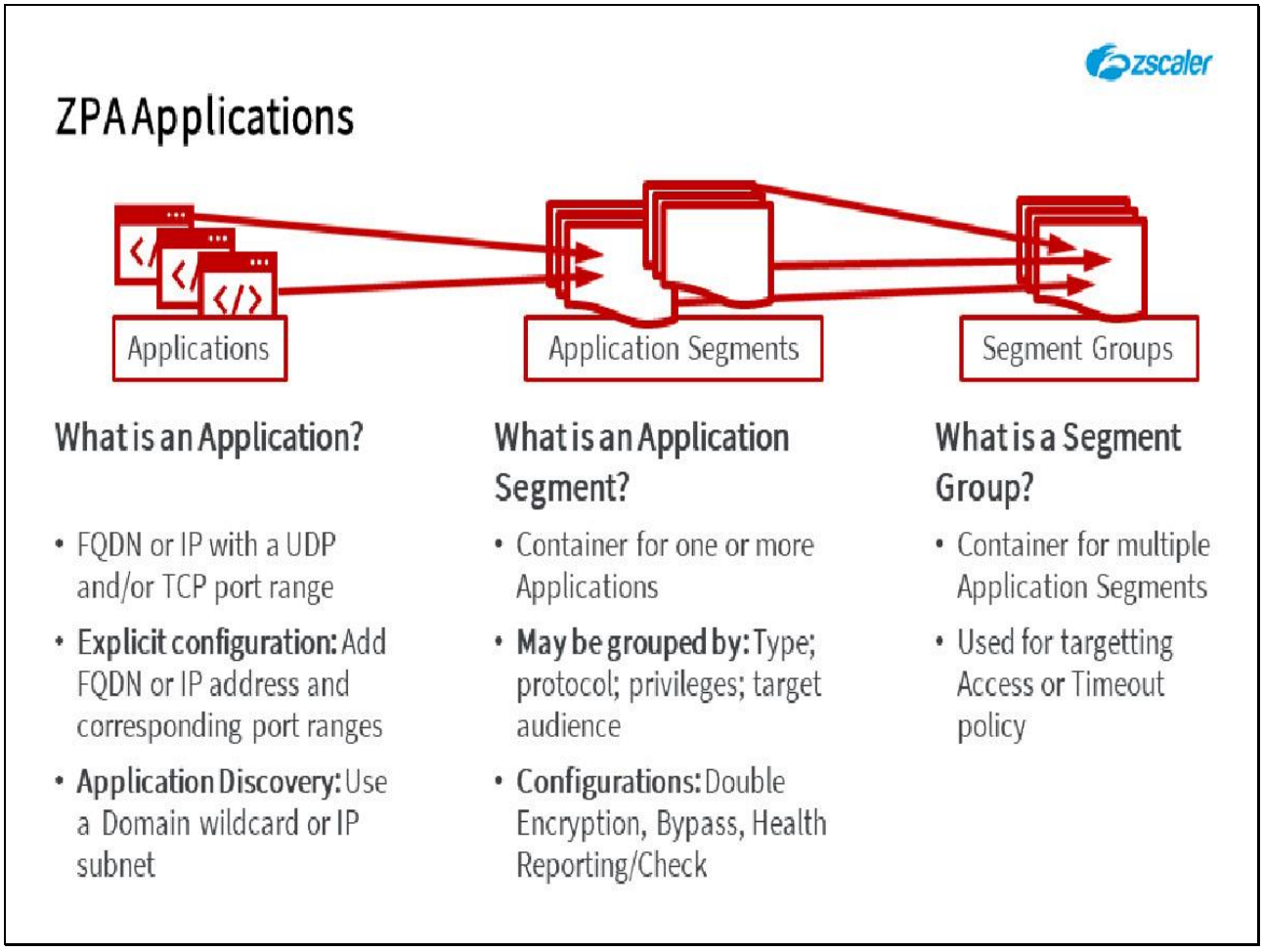
Slide notes

So what is an **Application Segment**? This is a container for one or more **Applications**, so it is a way to logically group applications within the ZPA service. How you group **Applications** onto an **Application Segment** is entirely up to you:

- You might group applications by protocol, e.g. all web applications on TCP ports 80/443, or all SSH connections;
- You might group applications based on their physical location, e.g. all applications hosted in a particular data center;
- You might choose to group all your applications that require **Double Encryption** onto the same **Application Segment**;
- Or, based on who the target user population is supposed to be.

There are some configuration settings within an **Application Segment** that would apply to all the **Applications** that you add to the segment, for example; the **Double Encryption** option, the **Bypass** configuration and the **Health Reporting/Check** configuration.

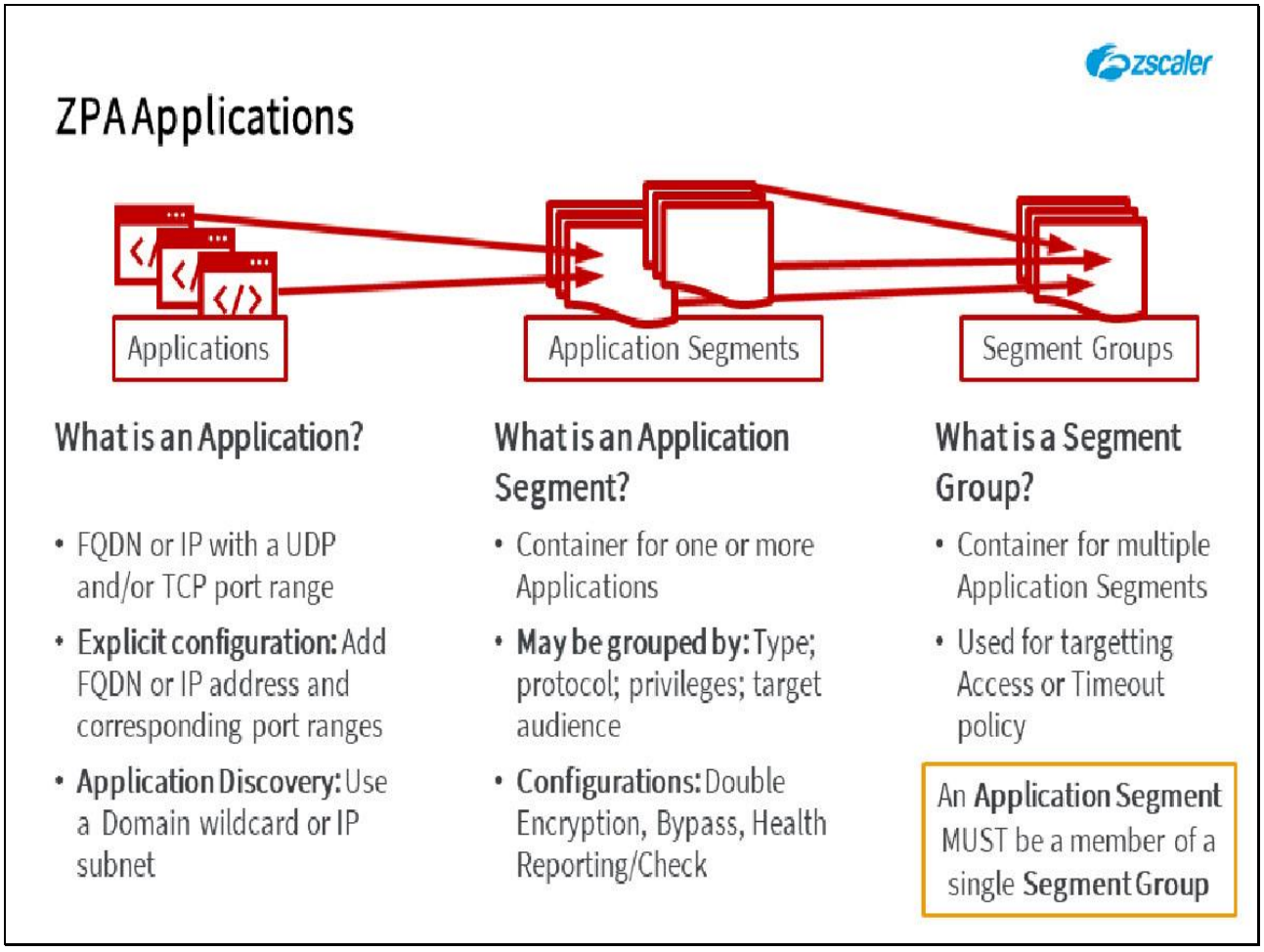
Slide 46 - ZPA Applications



Slide notes

Lastly, what is a **Segment Group**? This is simply a container for multiple **Application Segments**, which can then be used for targeting policy (Access or Timeout) against all the included **Application Segments** (and **Applications** defined within them).

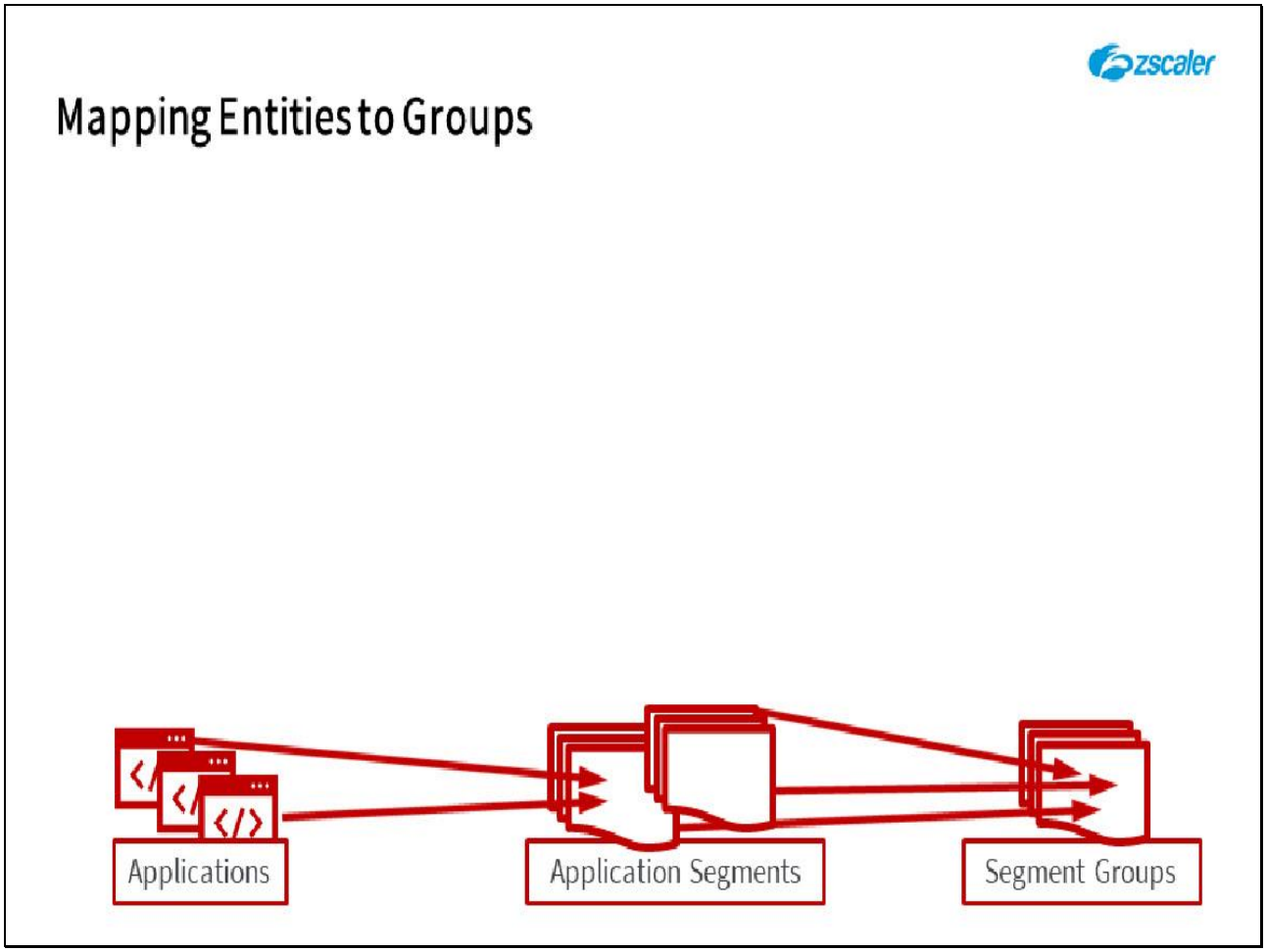
Slide 47 - ZPA Applications



Slide notes

Note that an **Application Segment** MUST be a member of one - and ONLY one - **Segment Group**. An **Application Segment** cannot belong to multiple **Segment Groups** simultaneously.

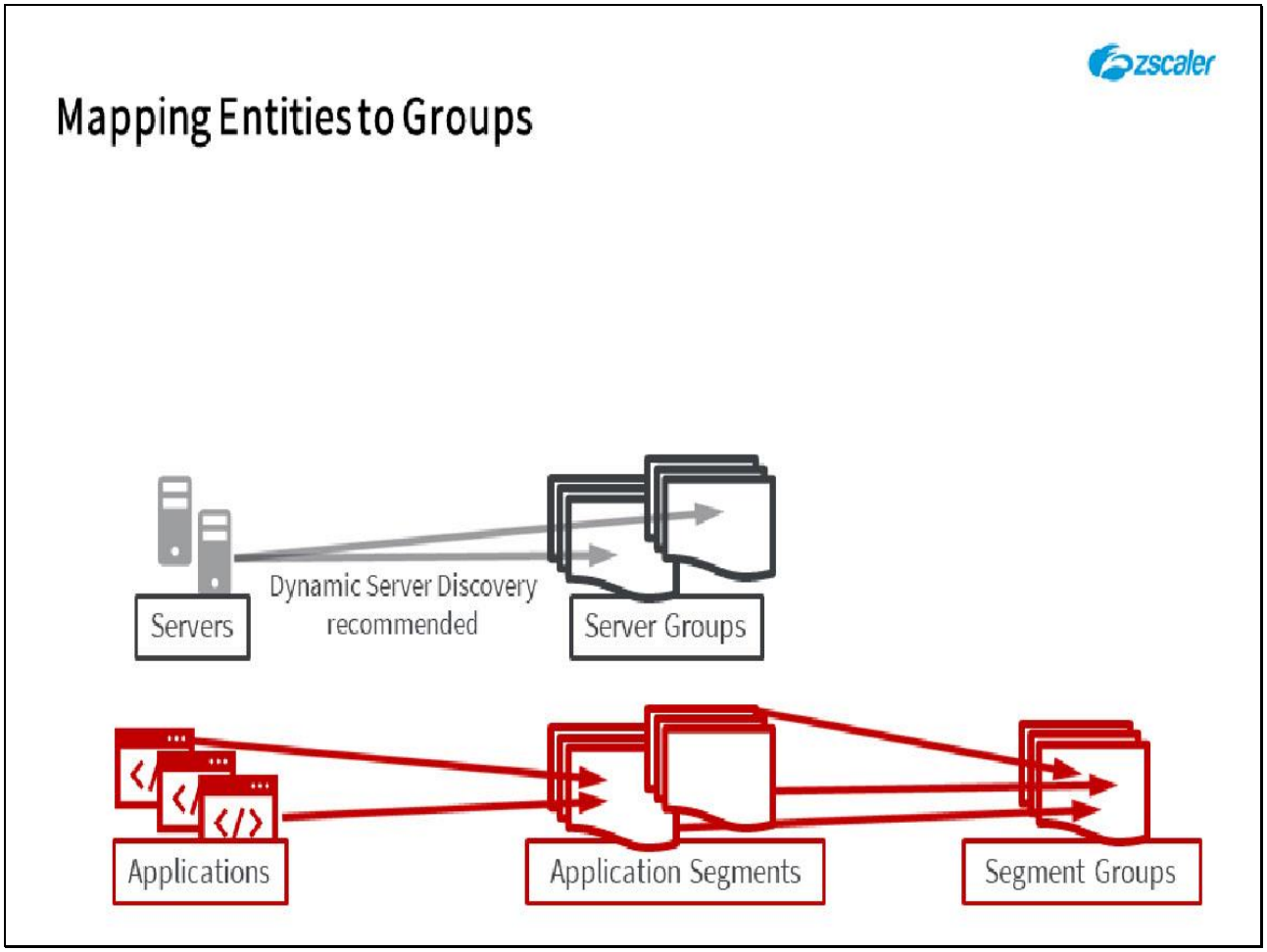
Slide 48 - Mapping Entities to Groups



Slide notes

Let’s now move on to talk about the mapping of the various logical ZPA entities to one another. We have just looked at the relationship of **Applications** to **Application Segments** and **Segment groups**.

Slide 49 - Mapping Entities to Groups

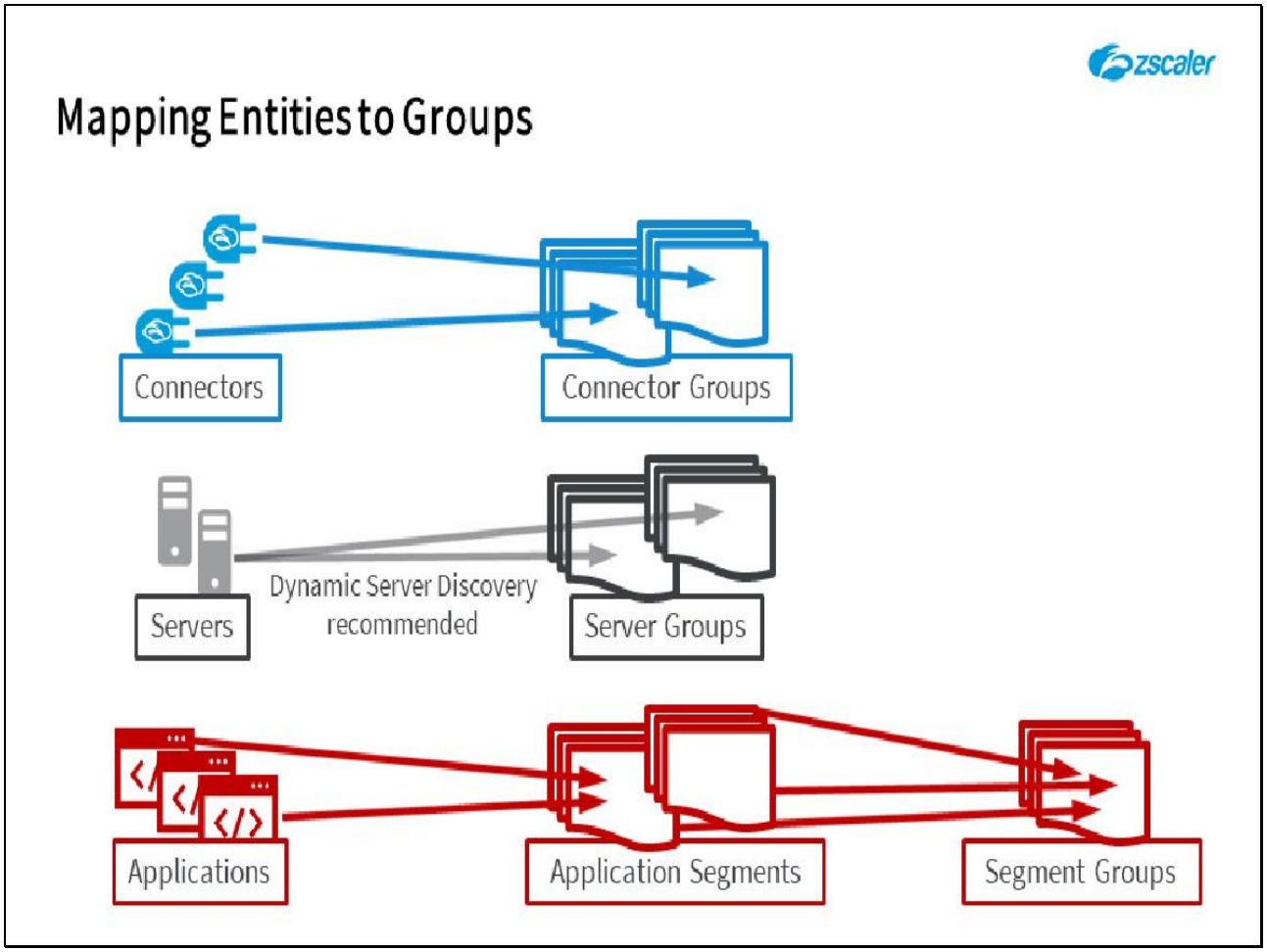


Slide notes

A **Server** (if statically defined) can be a member of one or more **Server Groups** if necessary.

Note, we recommend that you use the **Dynamic Server Discovery** option in the **Server Group** wherever possible. Statically defining servers should only be used under special circumstances (e.g. where you have 2 servers hosting the same application that need to be load-balanced).

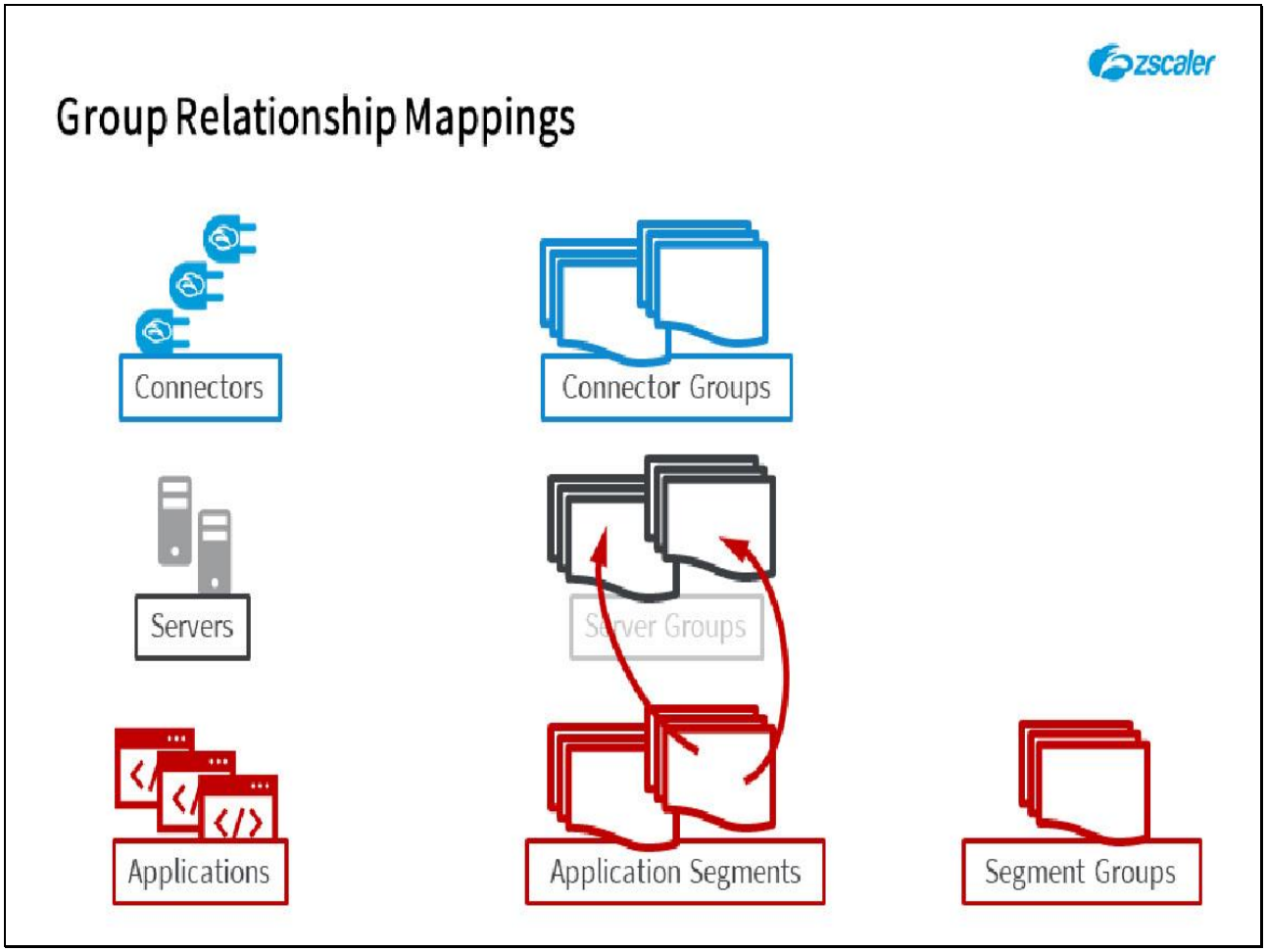
Slide 50 - Mapping Entities to Groups



Slide notes

A **Connector** must be a member of a **Connector Group** and can only be a member of a single group.

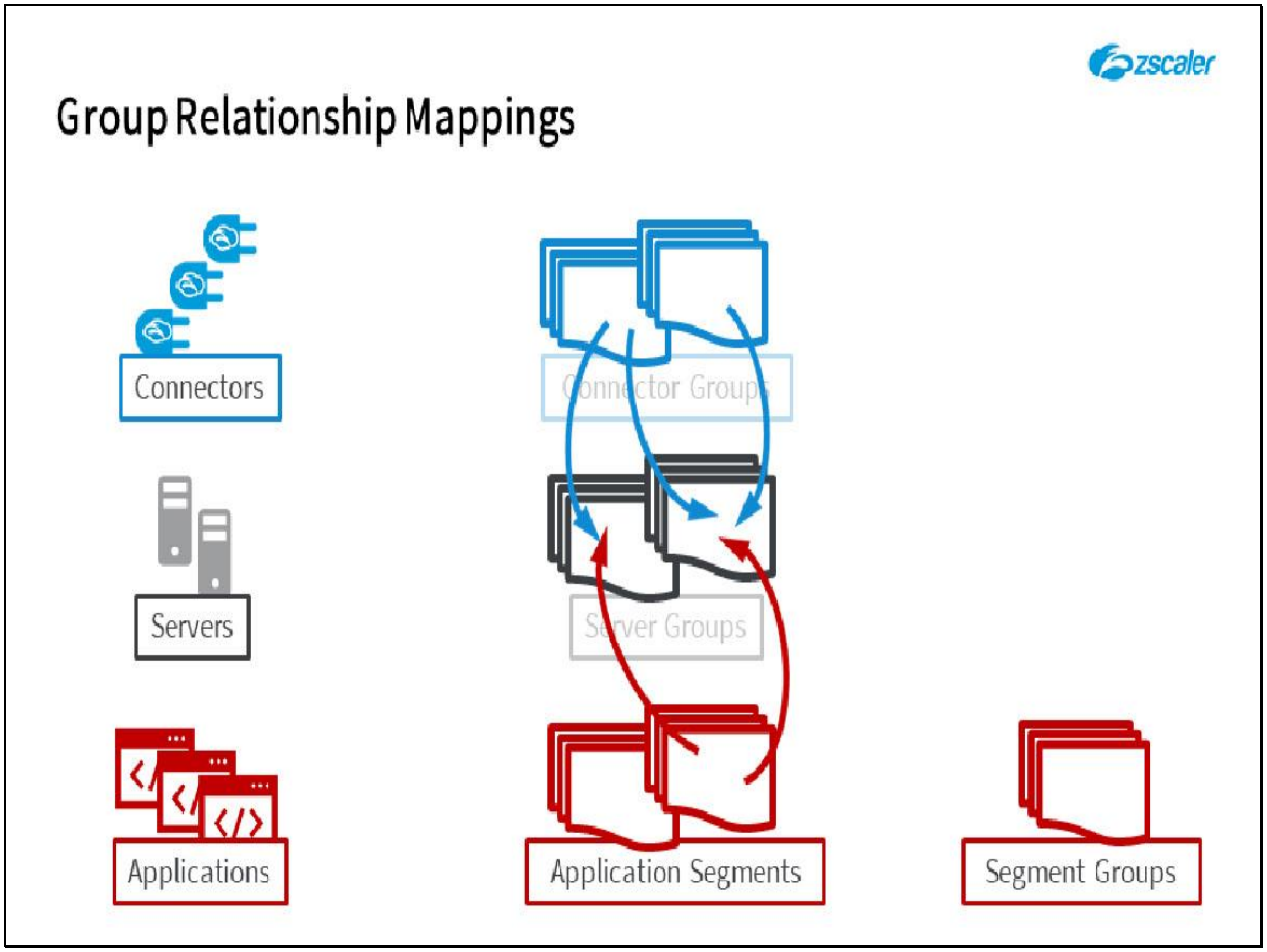
Slide 51 - Group Relationship Mappings



Slide notes

An **Application Segment** can be associated to multiple **Server Groups**. Which means that the associated **Applications** will be available through each **Server Group** that the **Application Segment** has been mapped to.

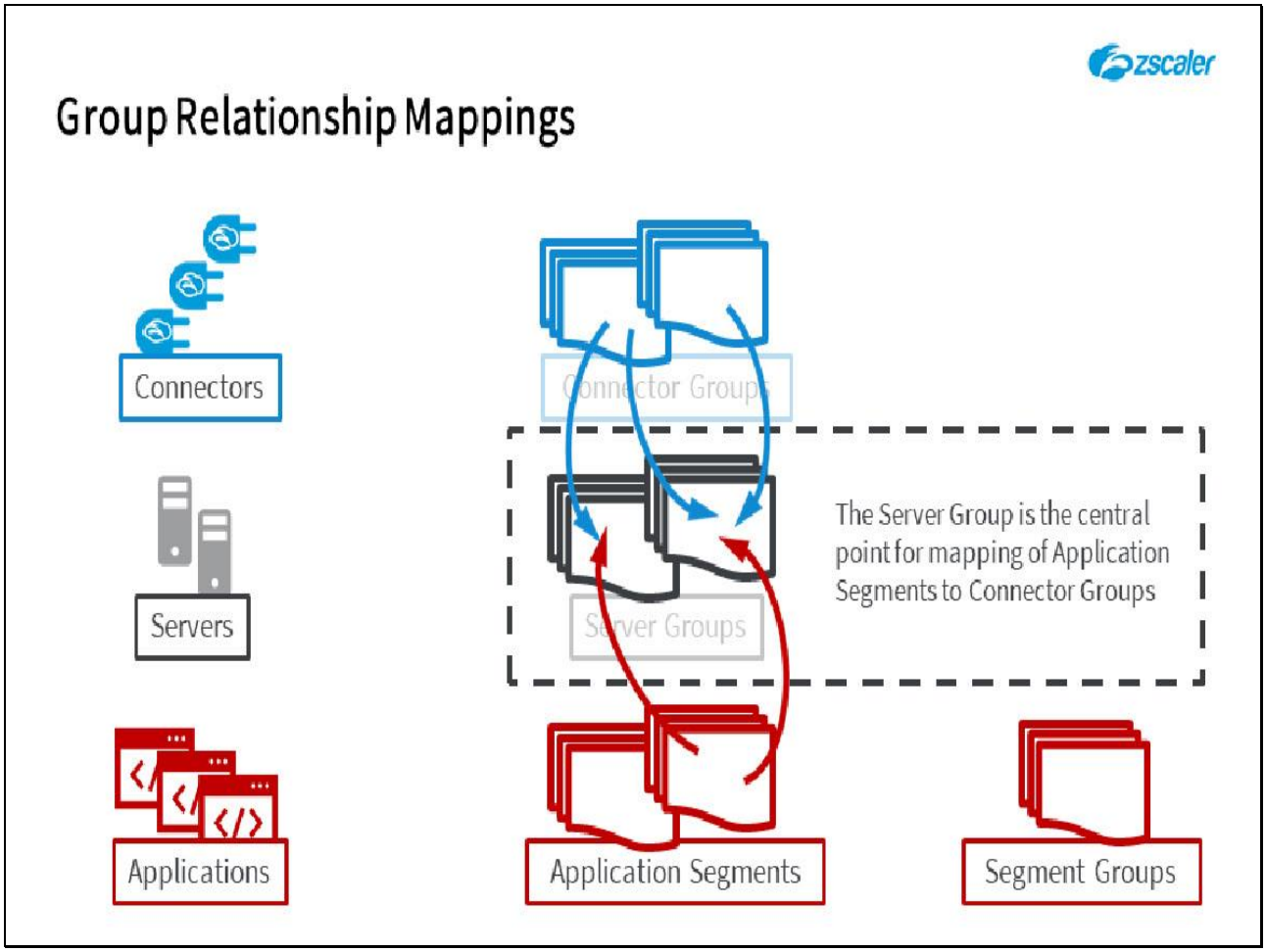
Slide 52 - Group Relationship Mappings



Slide notes

One or more **Connector Groups** can be associated with one or more **Server Groups**. Which means that the **Applications** defined on any **Application Segments** mapped to a **Server Group** will be visible and accessible from any Connector that is a member of an associated **Connector Group**.

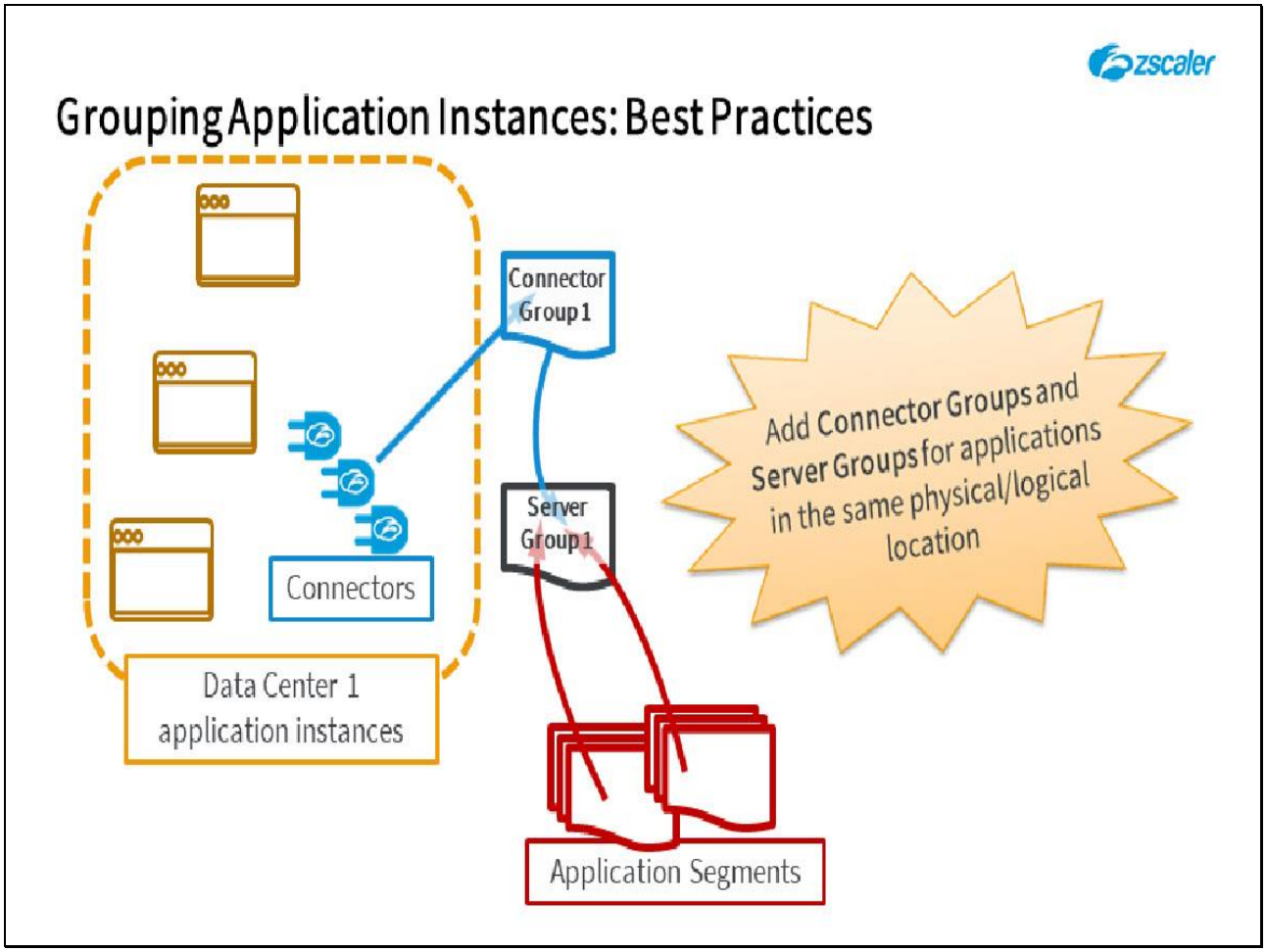
Slide 53 - Group Relationship Mappings



Slide notes

So the **Server Group** is actually the central point. The **Server Group** is where **Application Segments** are mapped; it is also where you associate the Connectors that are to advertise reachability to the included **Applications**.

Slide 54 - Grouping Application Instances: Best Practices

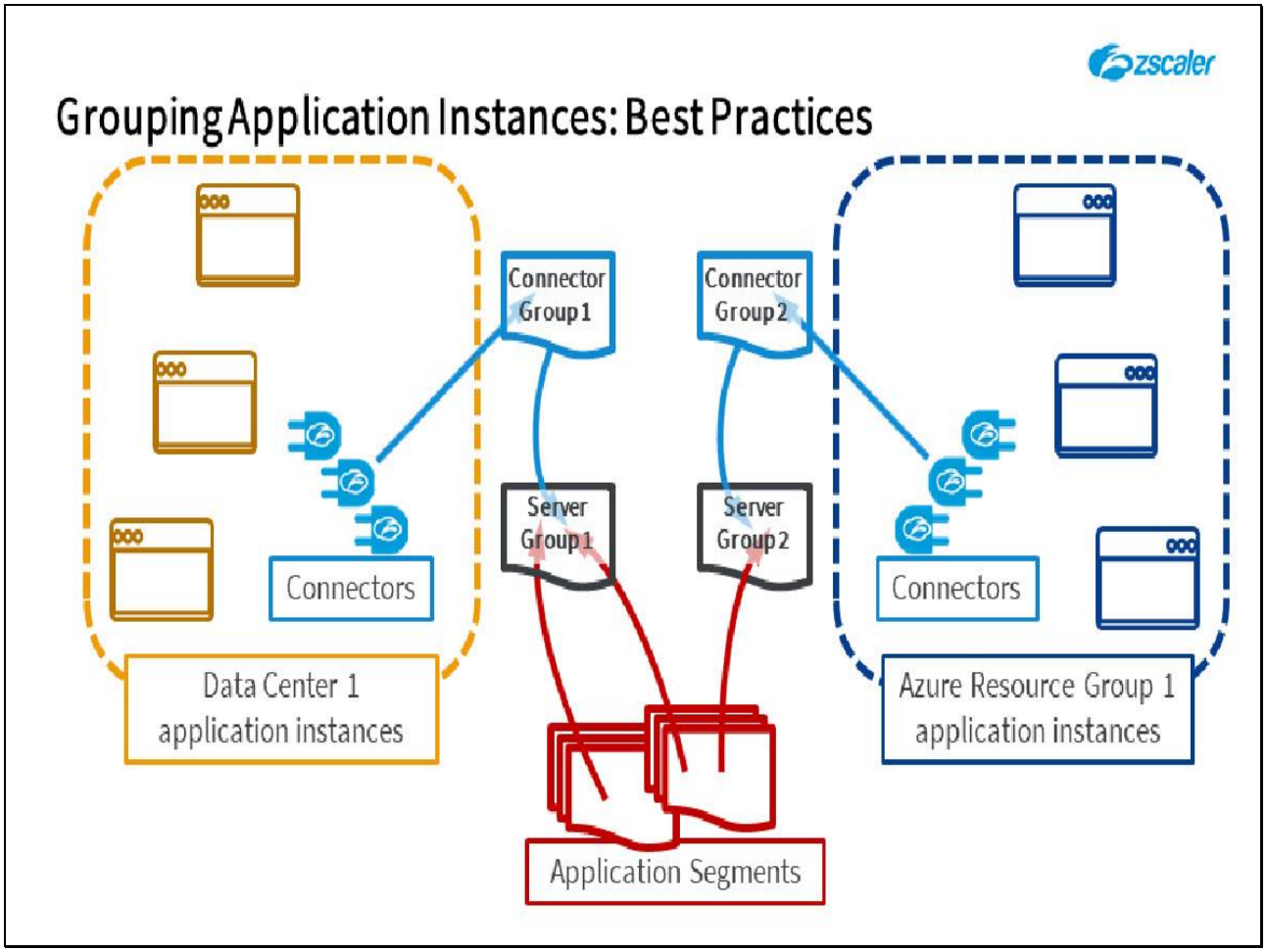


Slide notes

As far as best practices are concerned, we recommend that you create and map **Server** and **Connector Groups** based on the physical or logical location of the applications.

For example, you might create a **Server Group** for all the applications hosted on-premise in a particular data center. You can then map the applicable **Application Segments** to that **Server Group** and add the **Connector Group** for the on-premise Connectors. In this way, only the Connectors in a relevant group will be asked to find and connect to the target applications.

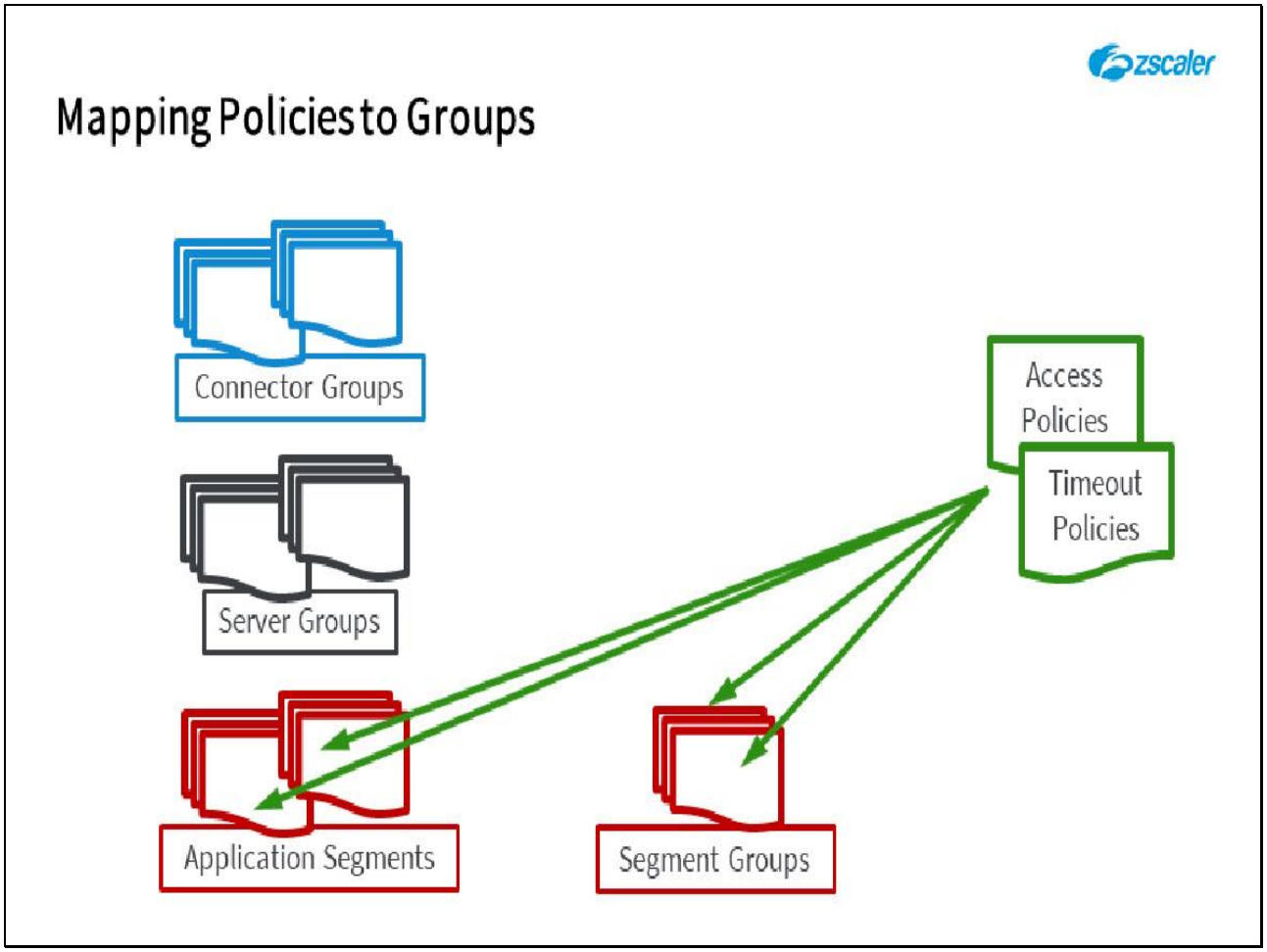
Slide 55 - Grouping Application Instances: Best Practices



Slide notes

Similarly, you might create a **Server Group** for the applications hosted on a particular Resource Group in Azure. You can then map the **Application Segments** and attach the relevant **Connector Group** for the Azure Connectors.

Slide 56 - Mapping Policies to Groups



Slide notes

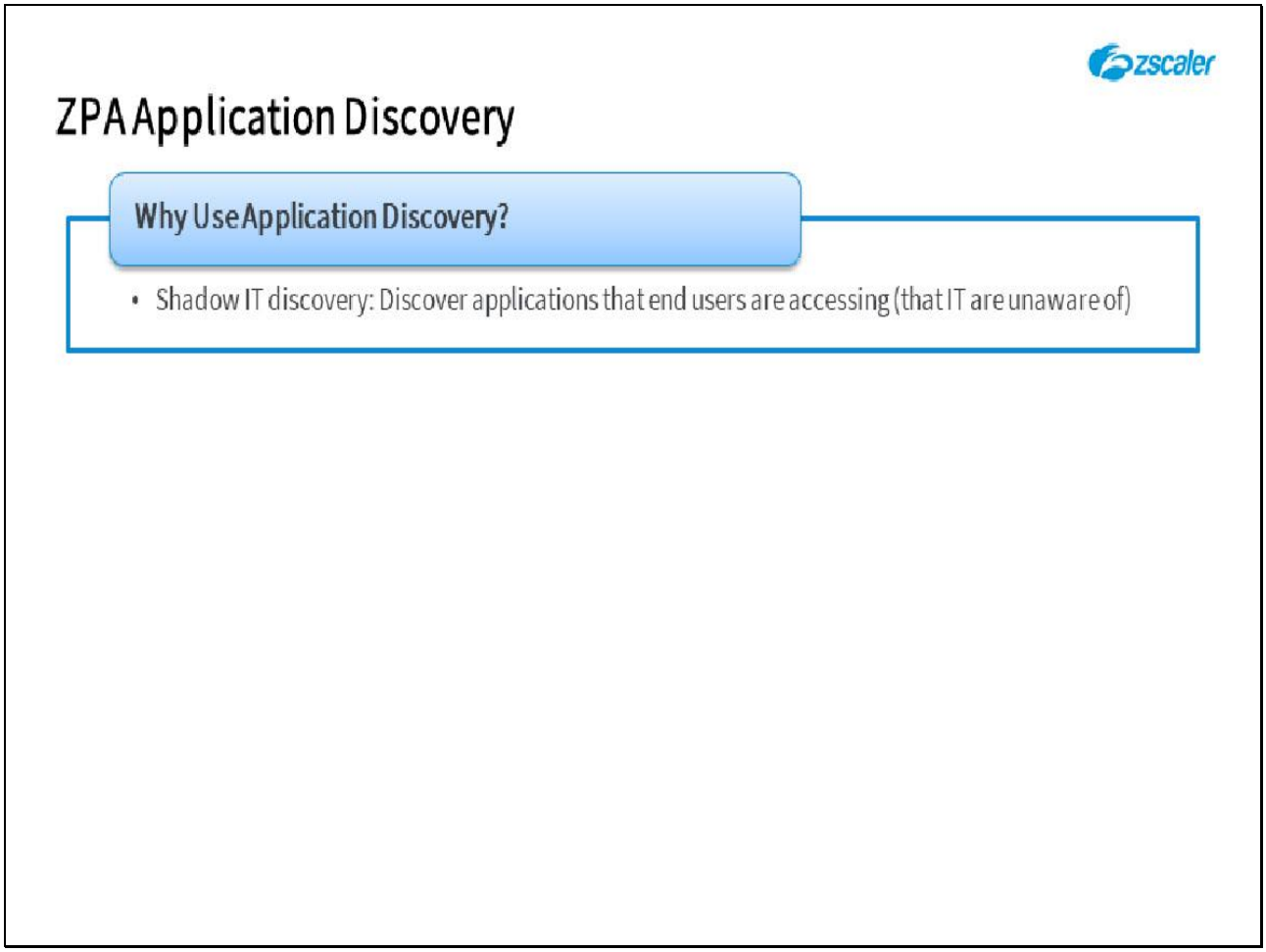
Just to complete the available ZPA mappings; an **Access Policy** or **Timeout Policy** rule may be applied to one or more **Application Segments**, or to one or more **Segment Groups**.

Slide 57 - Application Discovery

Slide notes

Next, we will look at the process of ZPA application discovery.

Slide 58 - ZPA Application Discovery

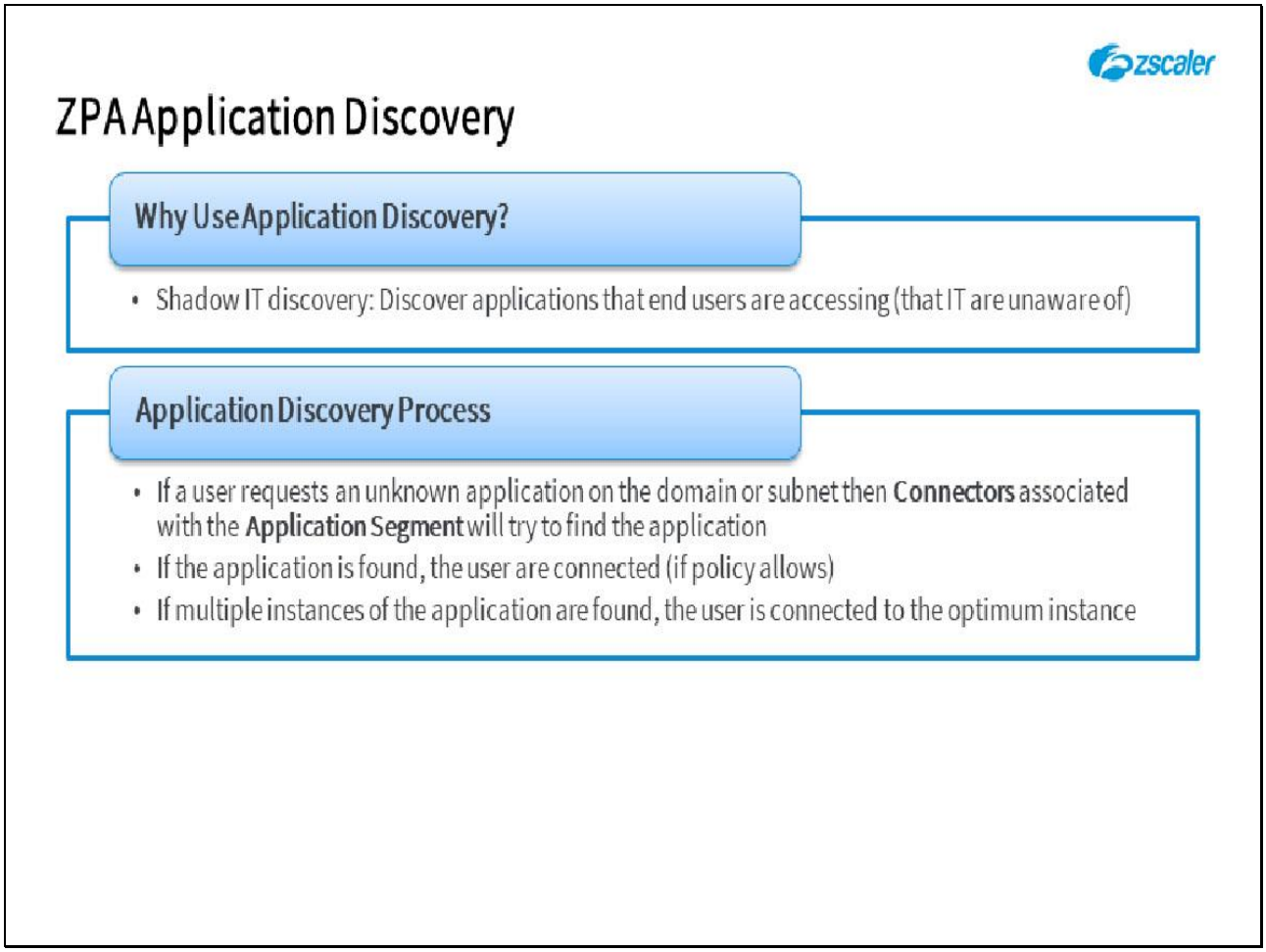


Slide notes

A major advantage of the ZPA solution is that it can be used to discover the applications that are actually in use by your end-user population. There is no need to know details for every application up-front, or to manually define them.

With ZPA application discovery, you have a tool to identify all the applications in use by your end-users, even those that have not necessarily been sanctioned or provided by IT. This capability, combined with ZPA's ability to make known applications invisible to unauthorized users, can be used to reduce your 'visible' attack surface dramatically.

Slide 59 - ZPA Application Discovery



Slide notes

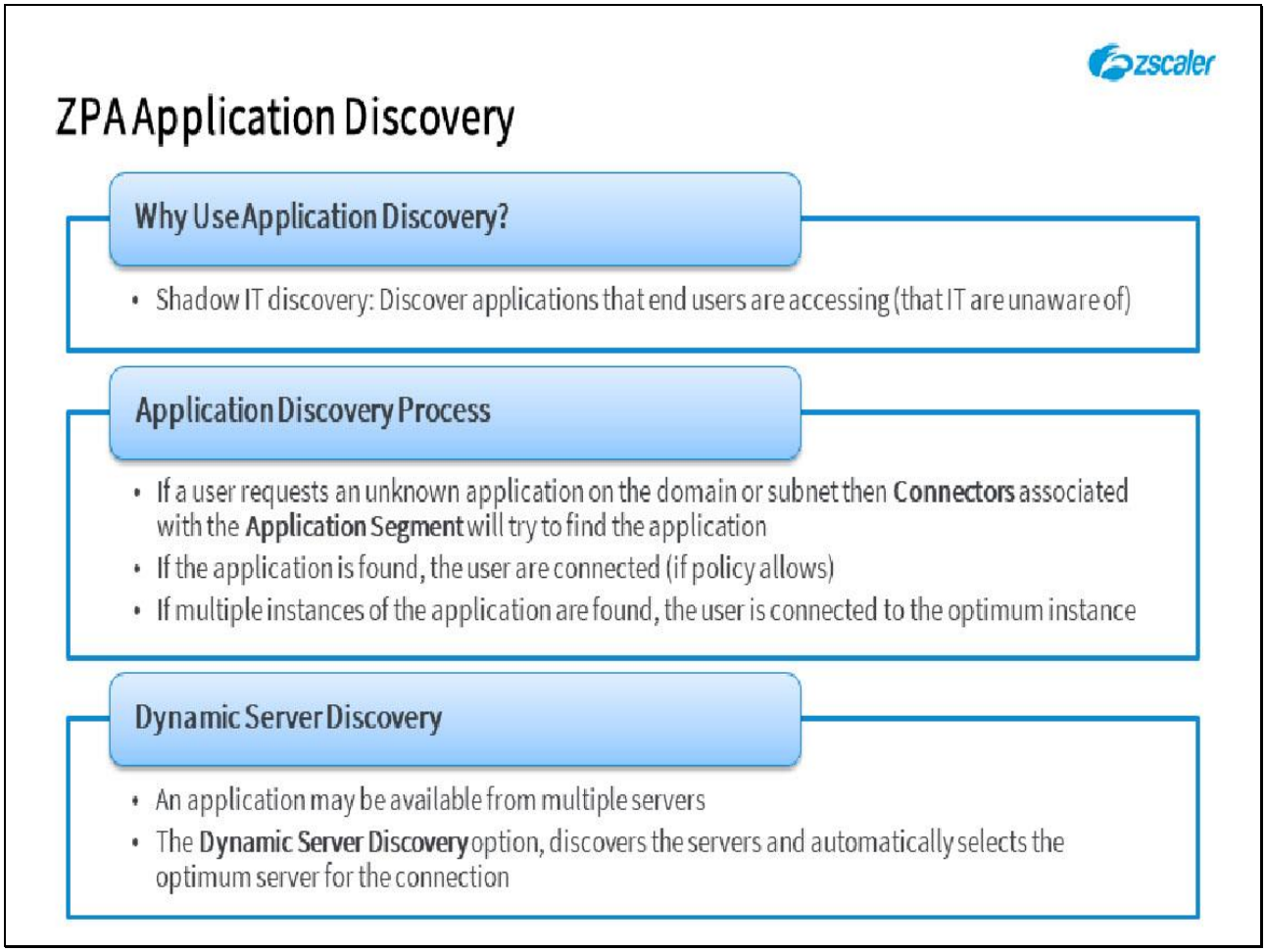
It is important to understand that ZPA application discovery is only ever an on-demand process; ZPA will only ever attempt to find an application when a user requests it. ZPA Connectors are not able to speculatively port-scan the network that they are connected to, just to see what they can find.

The whole purpose of the ZPA service is to conceal and protect your internal applications from unauthorized access. There is absolutely no need for Zscaler to know anything about applications or services on your network, unless they are to be made available over ZPA.

When application discovery is enabled (by adding an **Application Segment** with a wildcard domain), if an end user requests an application that we don't know about on that domain, the appropriate set of Connectors will be instructed to try to send traffic to it and report on the RTT.

If the application is found, the end-user will be connected to it (as long as that is allowed by an Access policy). If multiple instances are found, we will connect the user to the optimum instance that minimizes their latency to it.

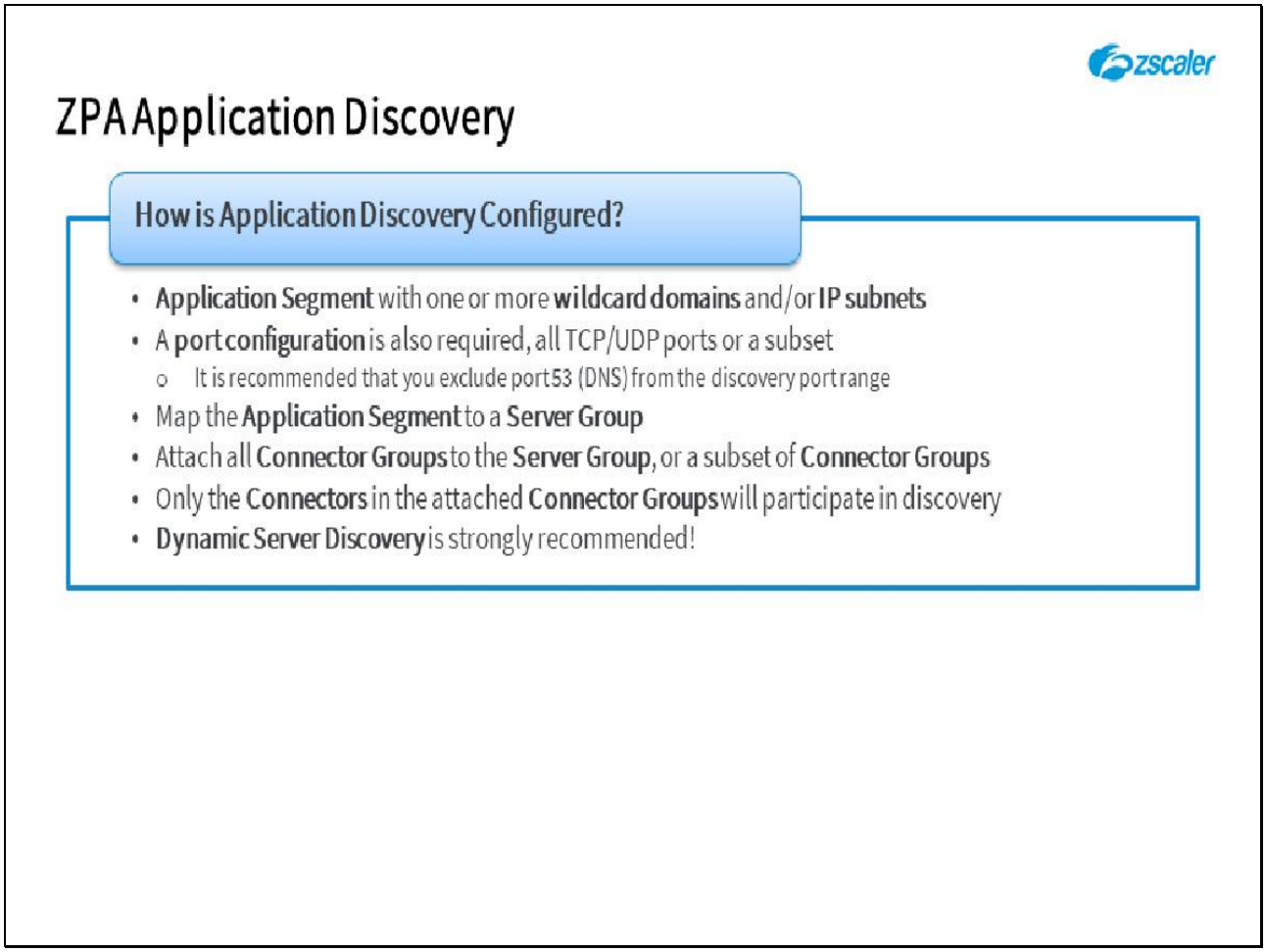
Slide 60 - ZPA Application Discovery



Slide notes

Part of the process of discovering applications is also the discovery of the servers that host them, through the **Dynamic Server Discovery** option. This option is recommended for all **Server Groups**; you only need to statically define individual servers under special circumstances.

Slide 61 - ZPA Application Discovery



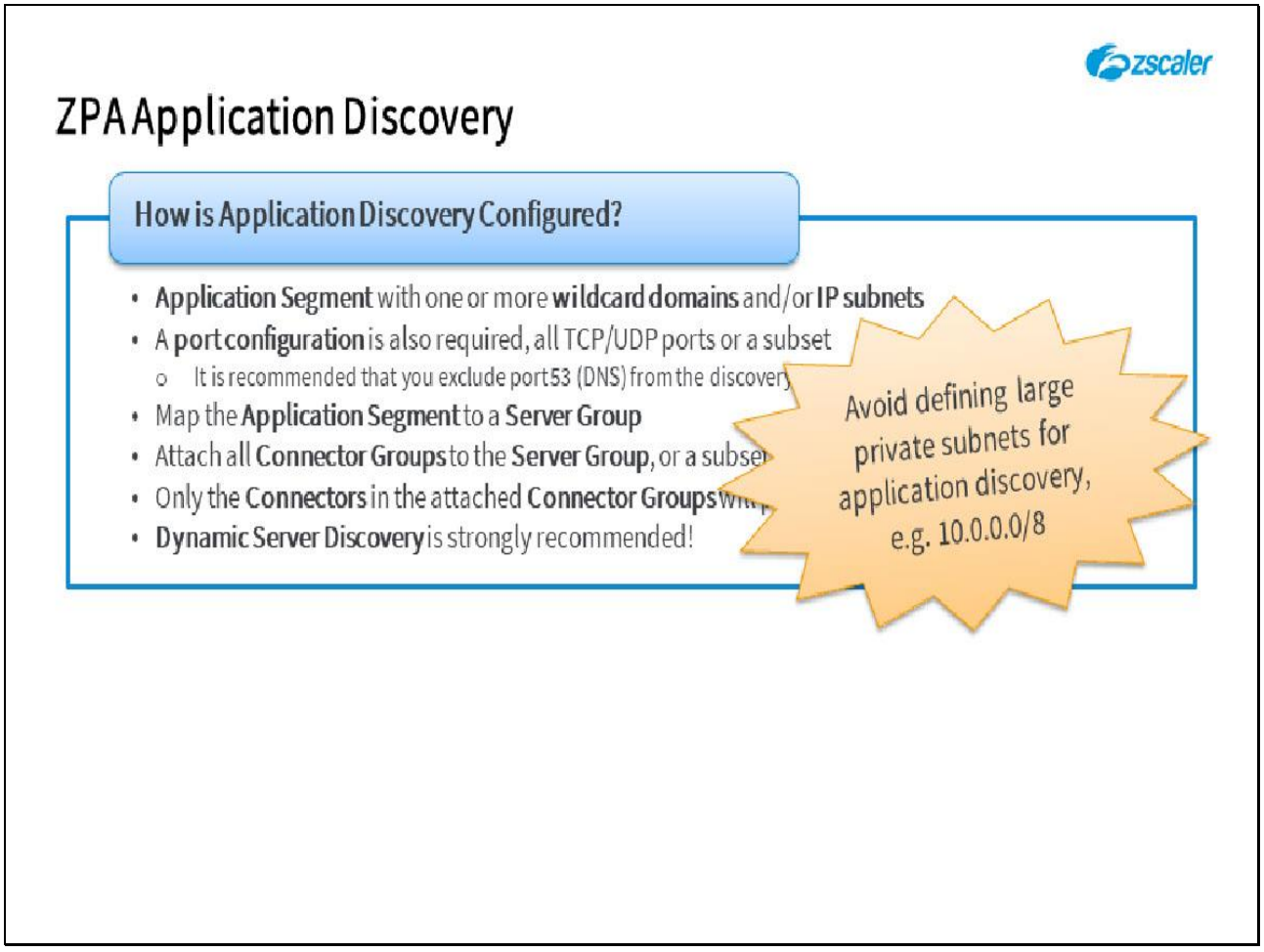
Slide notes

To enable application discovery, you will need to add at least one **Application Segment** with at least one wildcard domain specified as the **Application**, or you could add an IP subnet.

You will of course also need to define port ranges for the application discovery process. Typically, you will specify all TCP and UDP ports, although there are some ports that should be excluded, such as 53 (DNS).

The **Application** must be applied to a **Server Group** and at least one **Connector Group** associated to the **Server Group**. Once this configuration is saved, the Connectors of any associated **Connector Group** will be used to attempt to discover new applications (and servers) on the wildcard domain or subnet, when requested by an end-user.

Slide 62 - ZPA Application Discovery

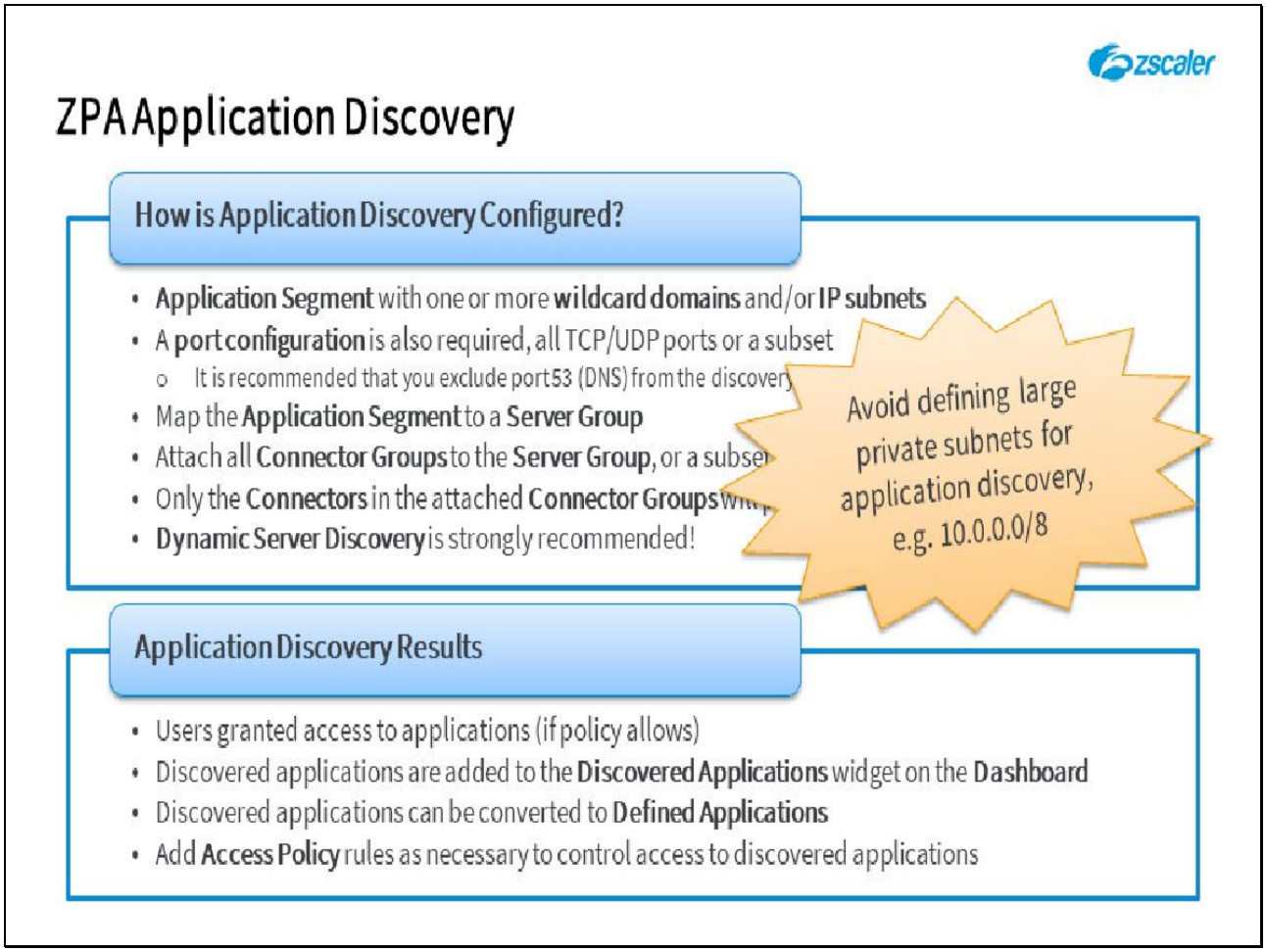


Slide notes

Note: we strongly recommend that you keep any IP subnets that you add for application discovery as small as possible, ideally no larger than a /24 mask. DO NOT add the 10.0.0.0 network with a /8 mask! Remember, EVERYBODY uses this address space for their internal networks!

Using this subnet will result in any and all applications on a 10.x.x.x address being considered a ZPA application, which will cause connectivity issues to partner-hosted or home-based applications.

Slide 63 - ZPA Application Discovery

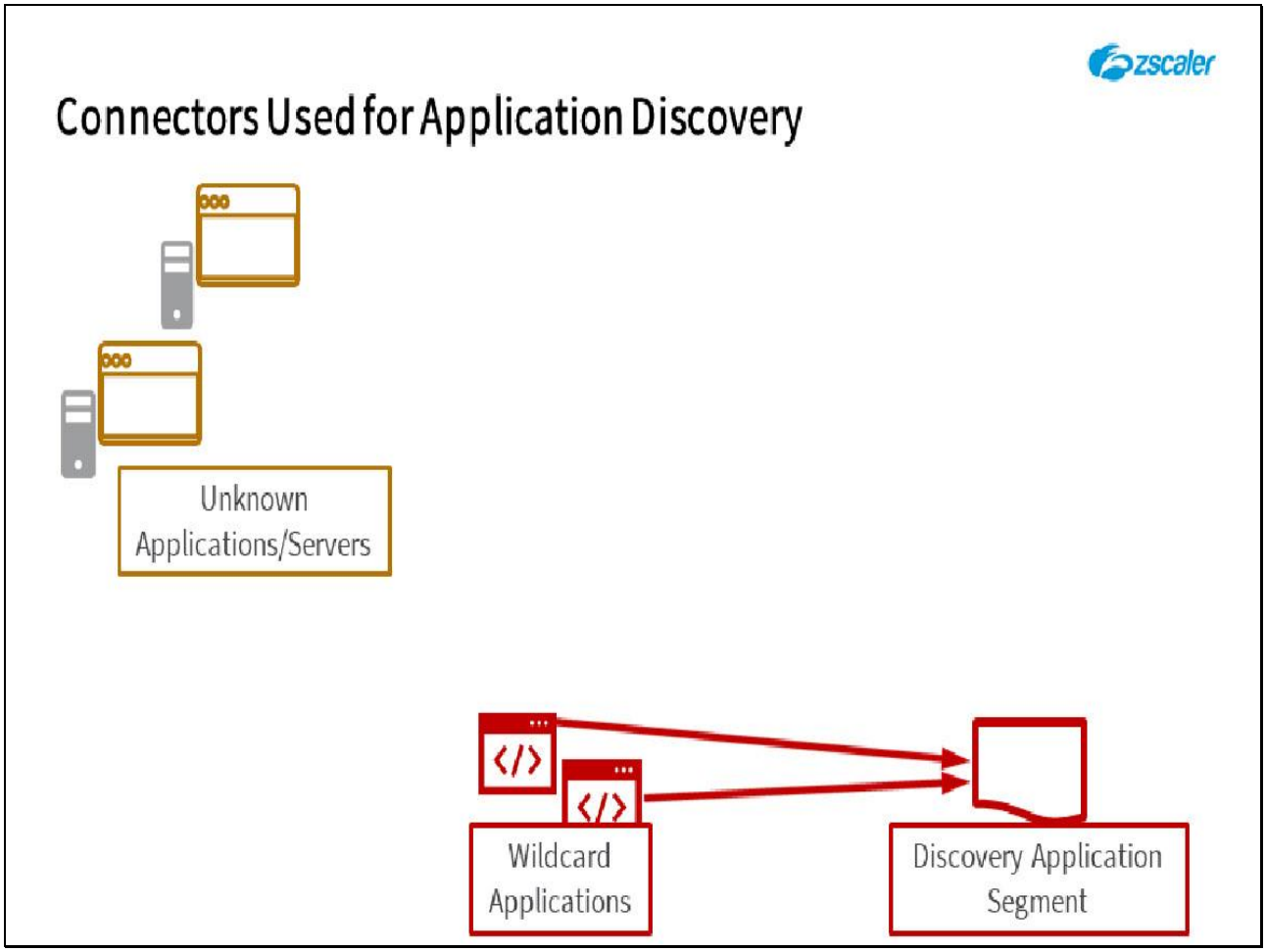


Slide notes

If an application and its server are discovered, the end-user can be given access to it, using an **Access Policy** rule. Discovered applications will appear on the Dashboards in the appropriate widgets, with full details of the end-users accessing them.

Once IT is aware of an application that is actively used by their end-users, they have the option to make it a defined application from the **Discovered Applications** widget on the **Applications Dashboard**, with an appropriate **Access Policy** configuration.

Slide 64 - Connectors Used for Application Discovery

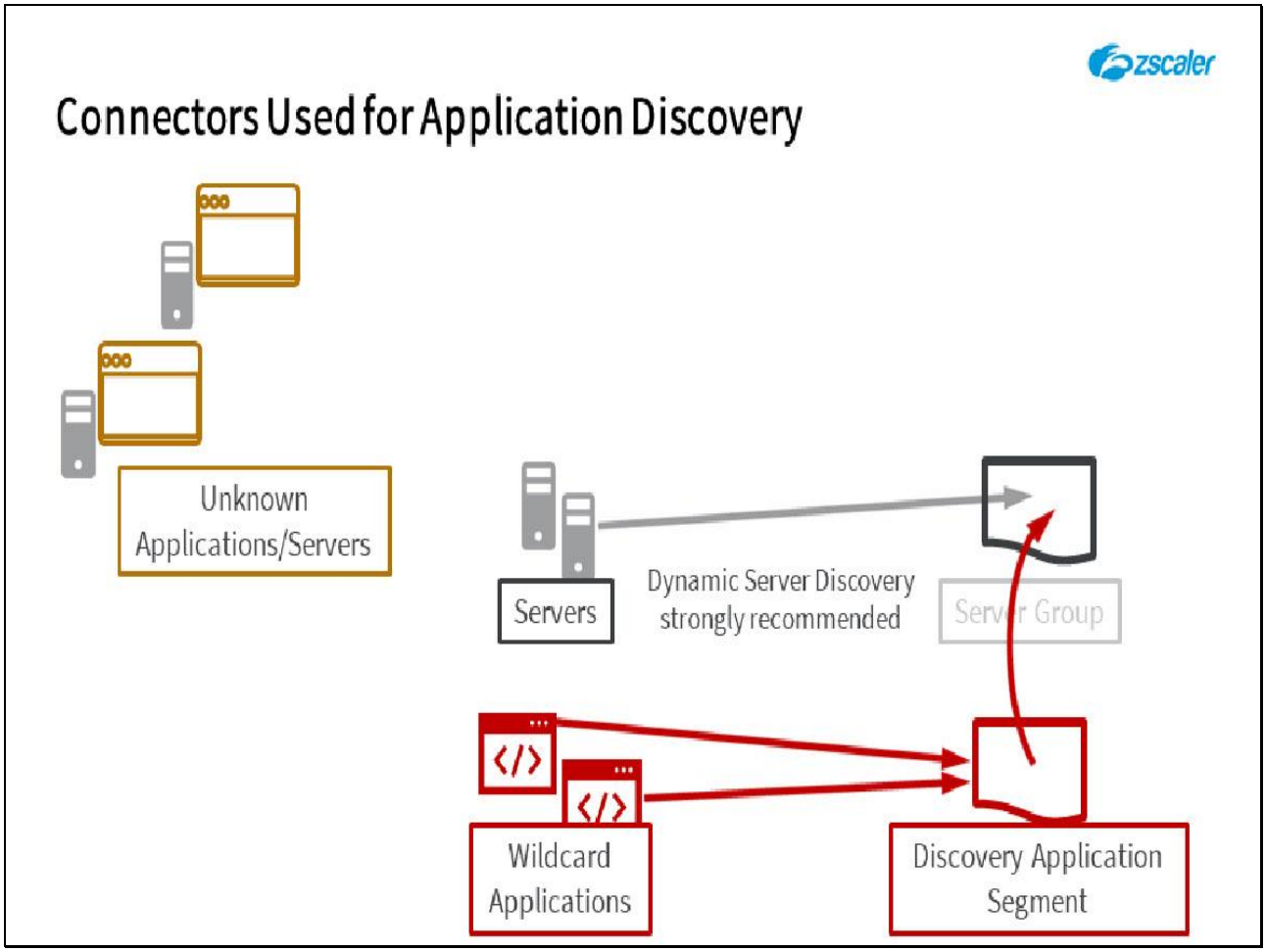


Slide notes

So the first thing that you need is a wildcard application domain or IP subnet, which must of course be configured on an **Application Segment**. For convenience in managing your discovery configurations, you may add multiple wildcards or subnets to the one **Application Segment** or spread them across their own individual **Application Segments**.

A suitable port range must also be added of course, which may include ALL TCP/UDP ports (with some exclusions) or be a more selective configuration. Only application requests that match the port range specified will be forwarded to the Connectors.

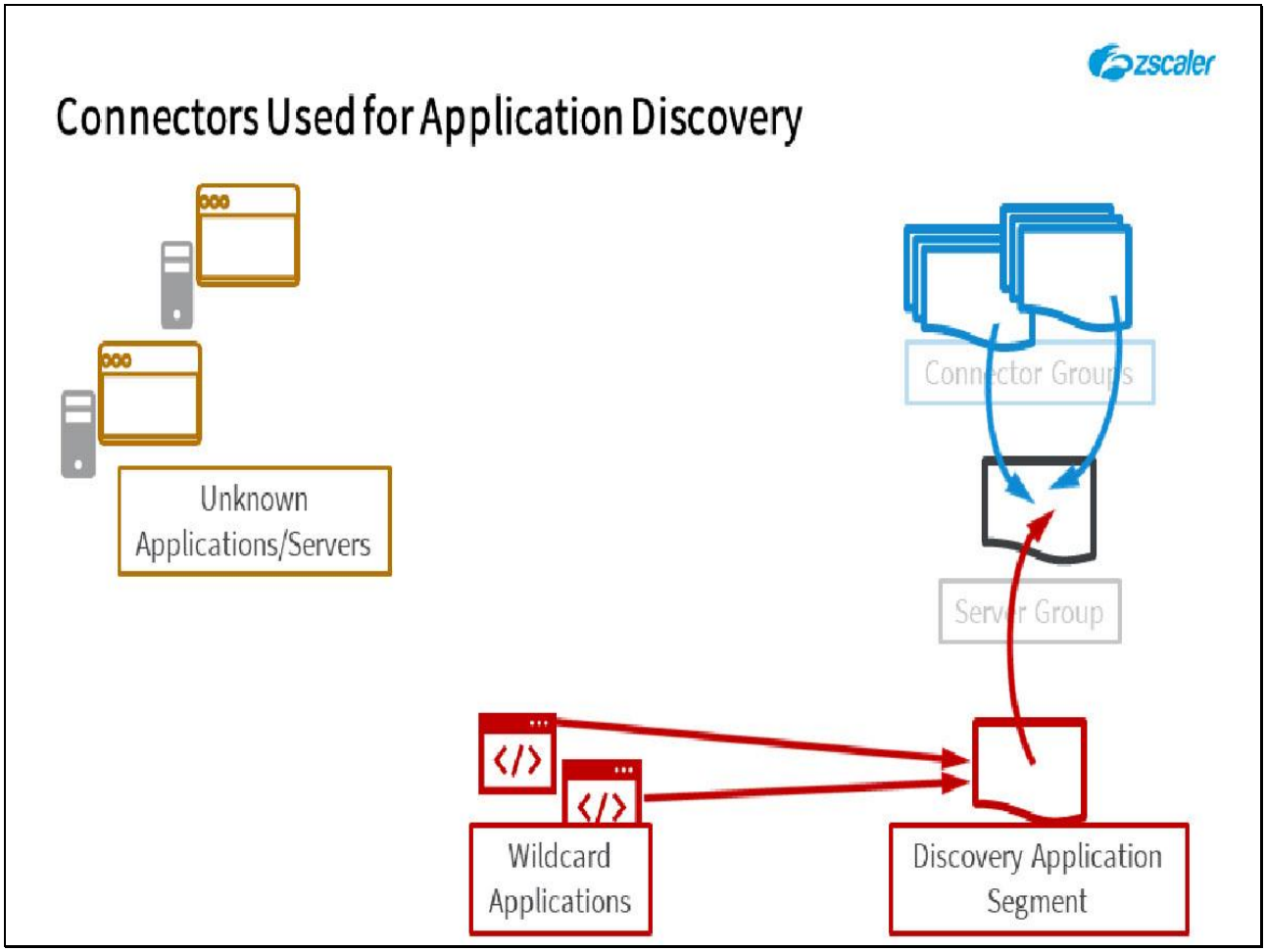
Slide 65 - Connectors Used for Application Discovery



Slide notes

The discovery **Application Segments** need to be mapped to an appropriate **Server Group** that has the **Dynamic Server Discovery** option enabled on it. There would be little point discovering applications, if you must then manually add and configure their servers!

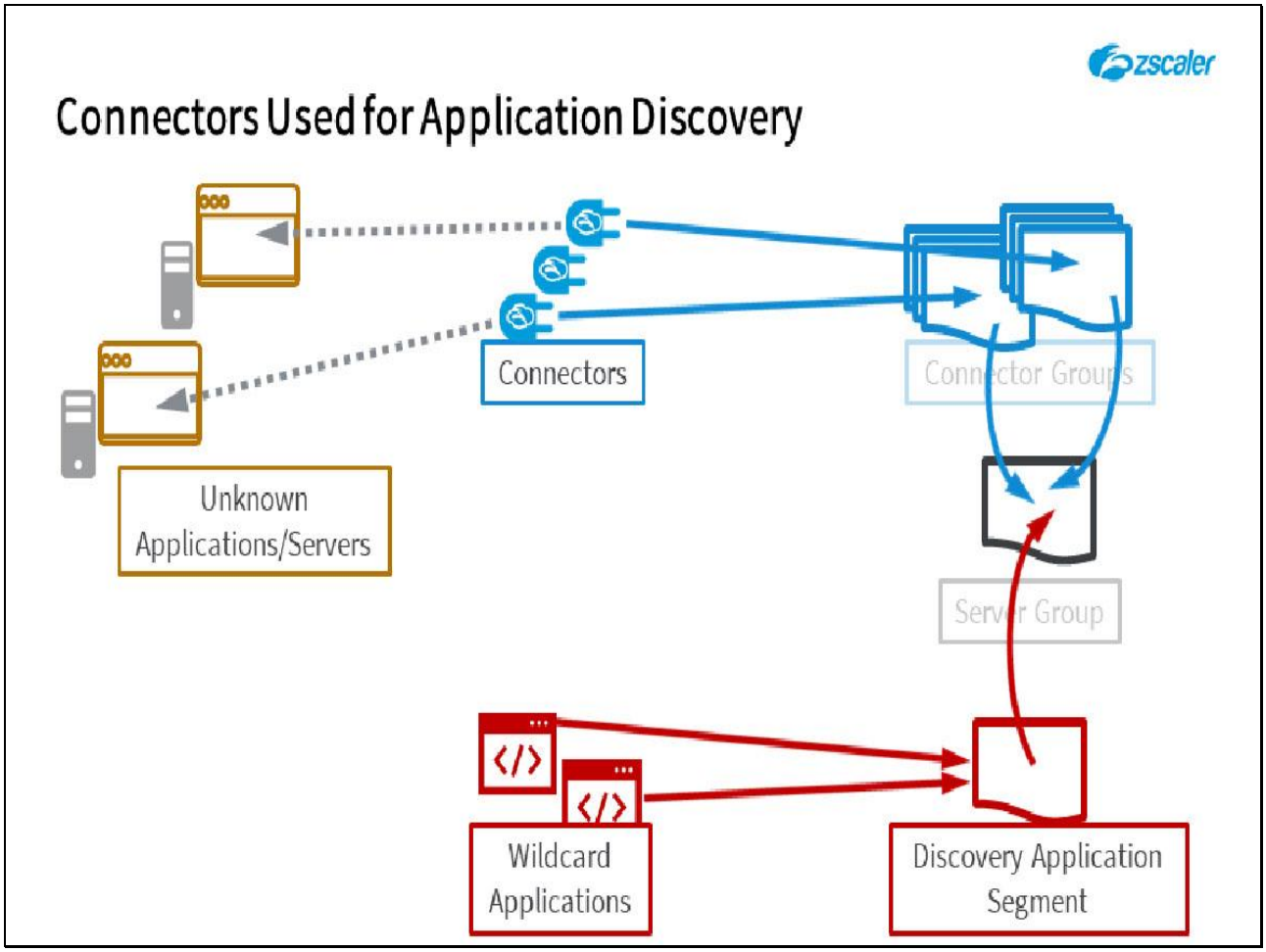
Slide 66 - Connectors Used for Application Discovery



Slide notes

At least one **Connector Group** needs to be added to the **Server Group**, ...

Slide 67 - Connectors Used for Application Discovery



Slide notes

...which means that, if a user requests an unknown application on any of the configured wildcard domains/subnets, any Connector from any of the mapped **Connector Groups** will be asked to find the application.

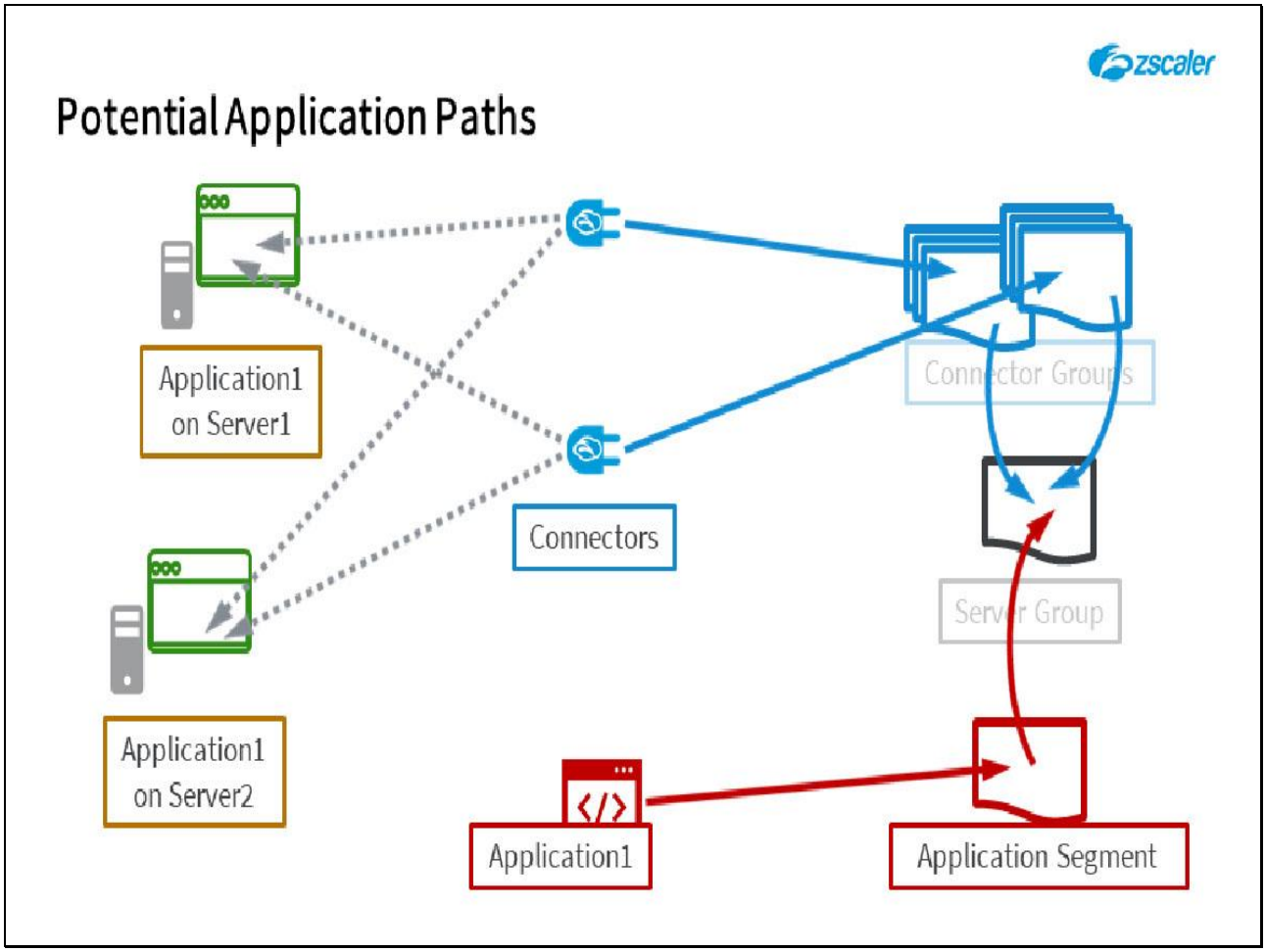
If the application is found, it will normally be found by multiple Connectors, so the ZPA service will identify the best one to use for this connection request.

Slide 68 - Path Selection

Slide notes

The final topic we will cover is a look at ZPA application Path Discovery.

Slide 69 - Potential Application Paths

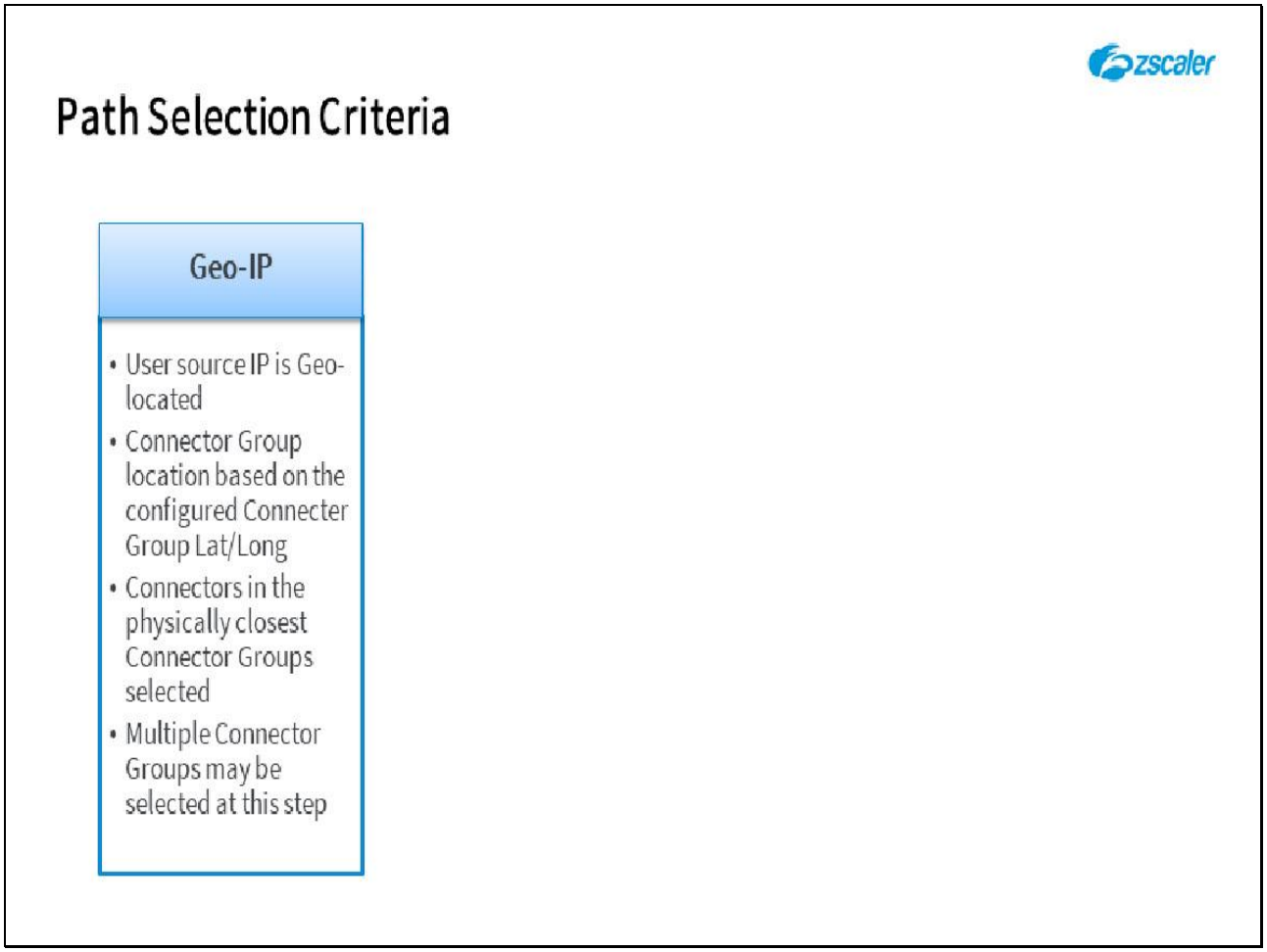


Slide notes

Under most circumstances with the ZPA service, there will always be multiple Connectors that are eligible to receive end-user traffic in order to reach any requested application. Remember, there should always be at least two Connectors per **Connector Group**, so at the very least, there will always be two Connector options.

If an application is deployed with a redundant configuration and is available in multiple locations (physical data centers, or virtual instances in a Cloud deployment), then it may well be seen by the Connectors from multiple **Connector Groups**. Which raises the question 'which is the best Connector to use for this user on this connection attempt?'.

Slide 70 - Path Selection Criteria



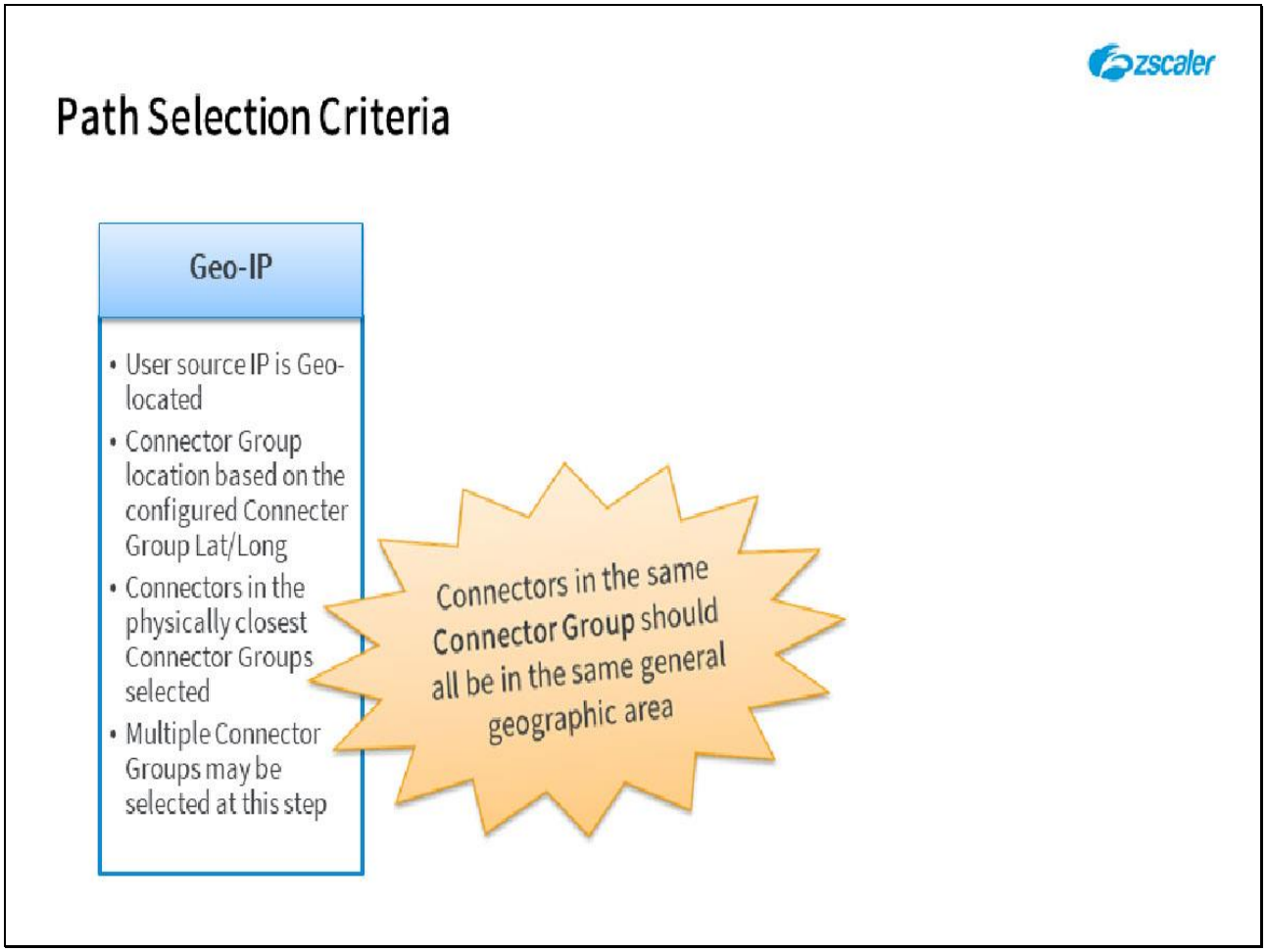
Slide notes

The ZPA service will always try to ensure that an end-user has an optimum connection to an application for every connection request. We use a number of factors when selecting which Connector to use.

The first criteria used is the Connector's physical distance from the end-user. We have the geo-location of the end-user from their egress IP address when connecting to the Internet. Plus, the location of the Connectors (Lat/Long) are configured on the **Connector Group**, so we have all the information we need to select the physically closest groups to the end-user.

This step actually selects any Connector Group within a certain distance from the end user, so it is possible - and actually quite common - that multiple Connector Groups are selected in this step.

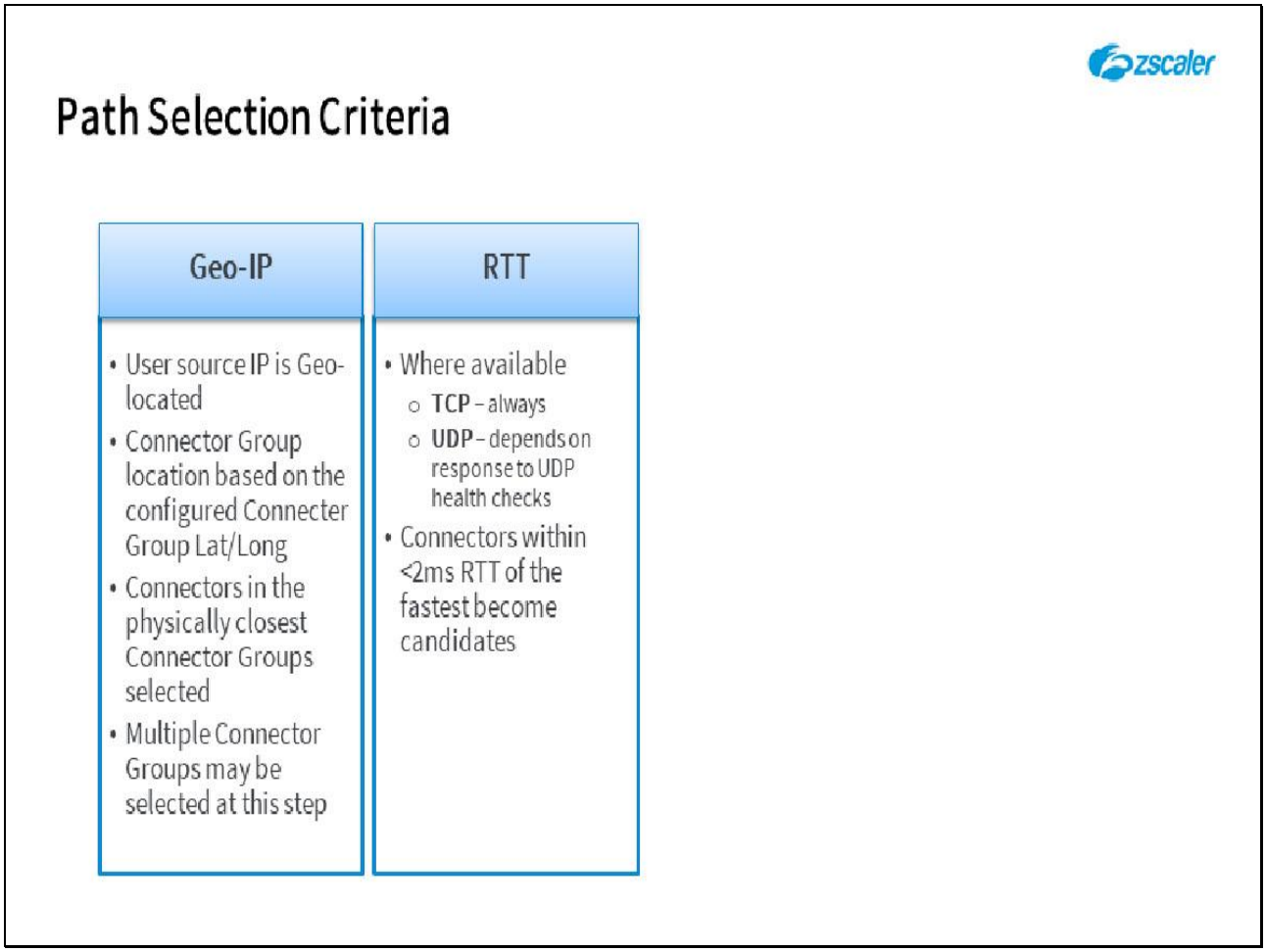
Slide 71 - ZPA Application Discovery



Slide notes

Which brings us to a best practice for Connector configuration: Connectors in the same **Connector Group** should be in the same general geographic area (ideally the same physical location) and the Lat/Long of the **Connector Group** should accurately reflect the center of gravity of the Connector physical locations.

Slide 72 - Path Selection Criteria

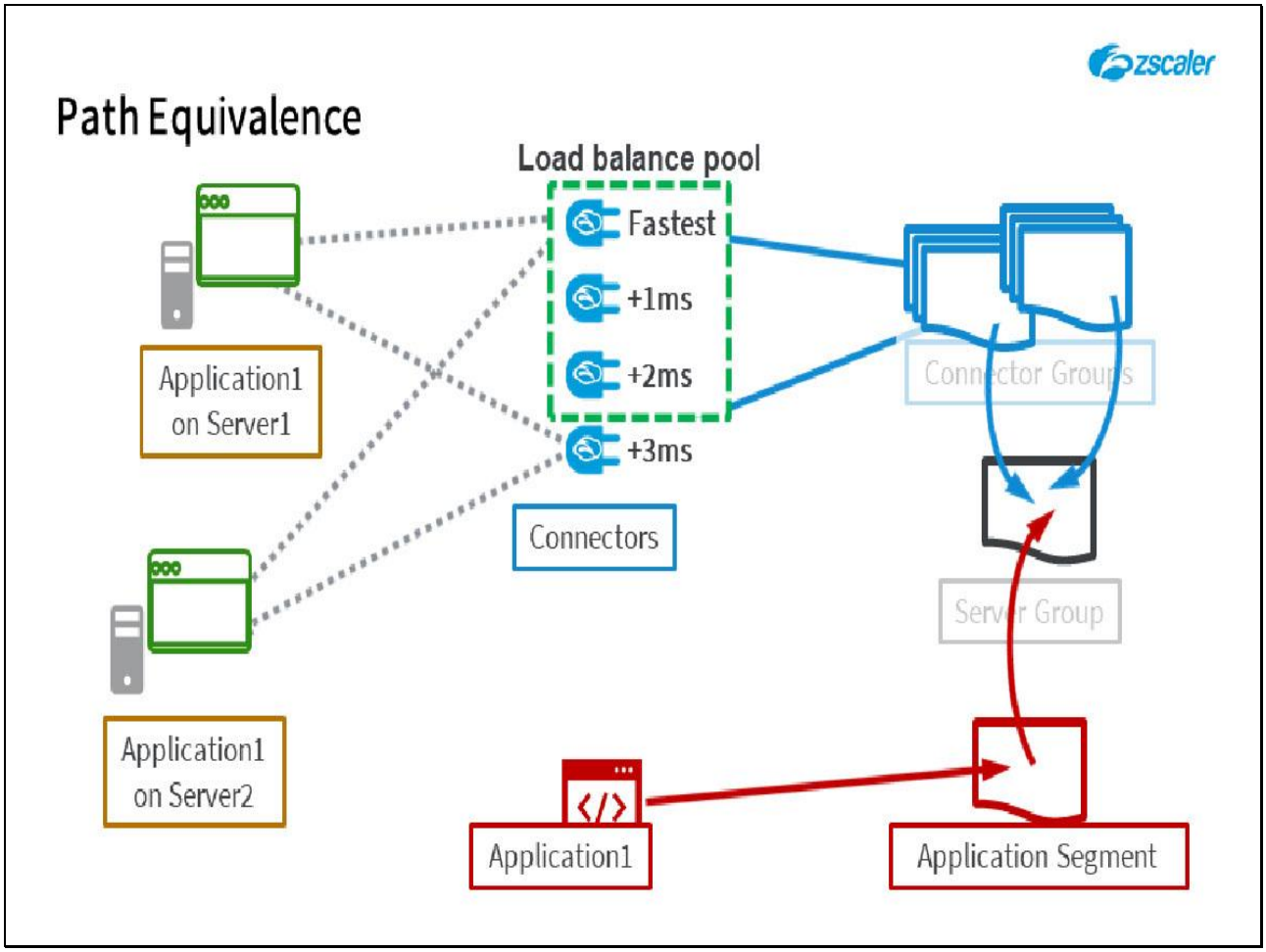


Slide notes

The next criteria we use is RTT from the Connector to the application, a measure that is always available to us for TCP connections and which may be available for UDP applications (as long as they respond to our ICMP health checks).

Connectors within an RTT of **2ms** of the fastest path become candidates and we will load balance end-user connections across this pool.

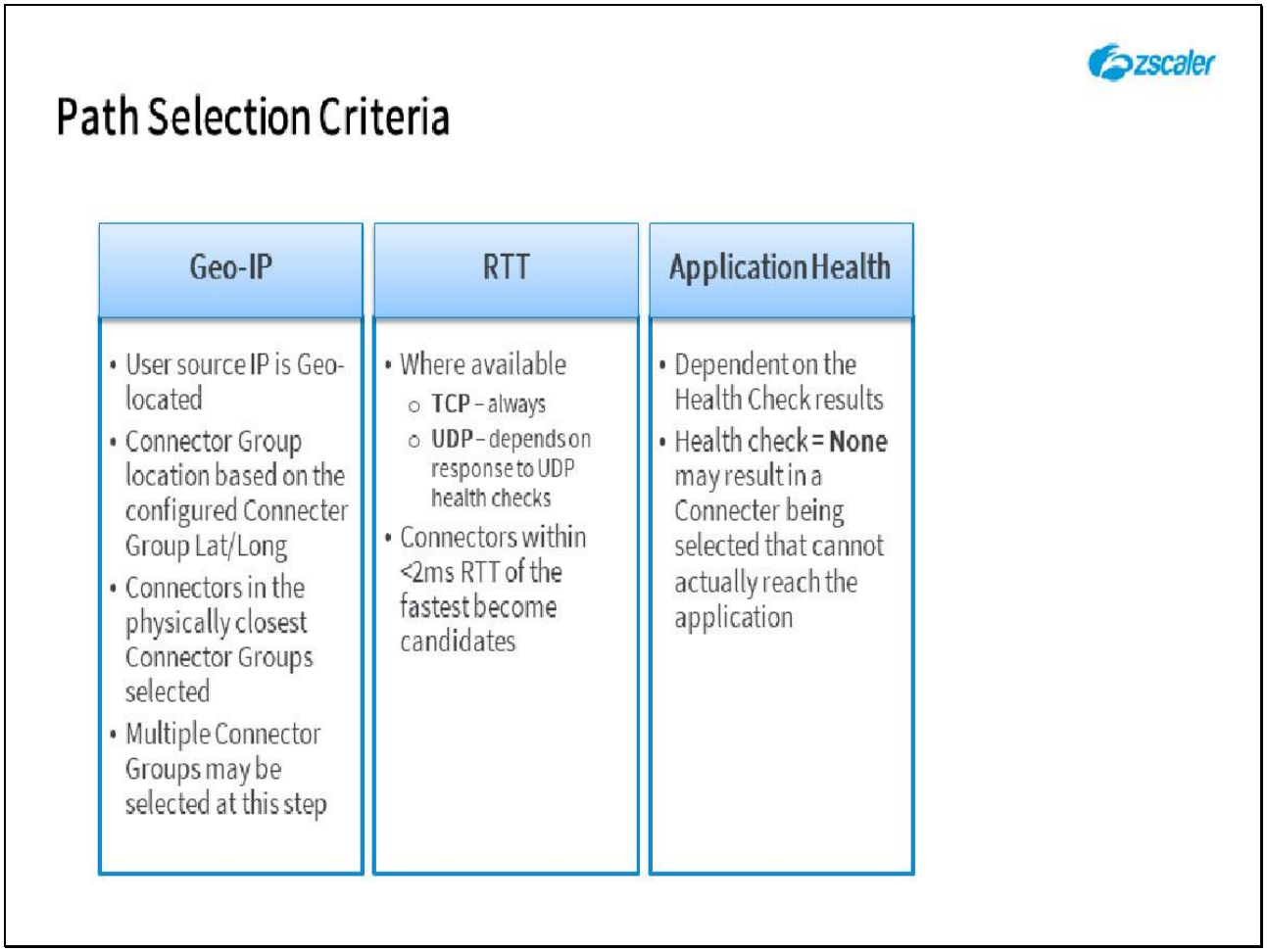
Slide 73 - Path Equivalence



Slide notes

Having selected the physically closest **Connector Group**, all Connectors with a RTT within **2ms** of the fastest Connector are considered to be equivalent. The ZPA system will round-robin user requests to the FQDN of the application across this pool of equivalent Connectors.

Slide 74 - Path Selection Criteria

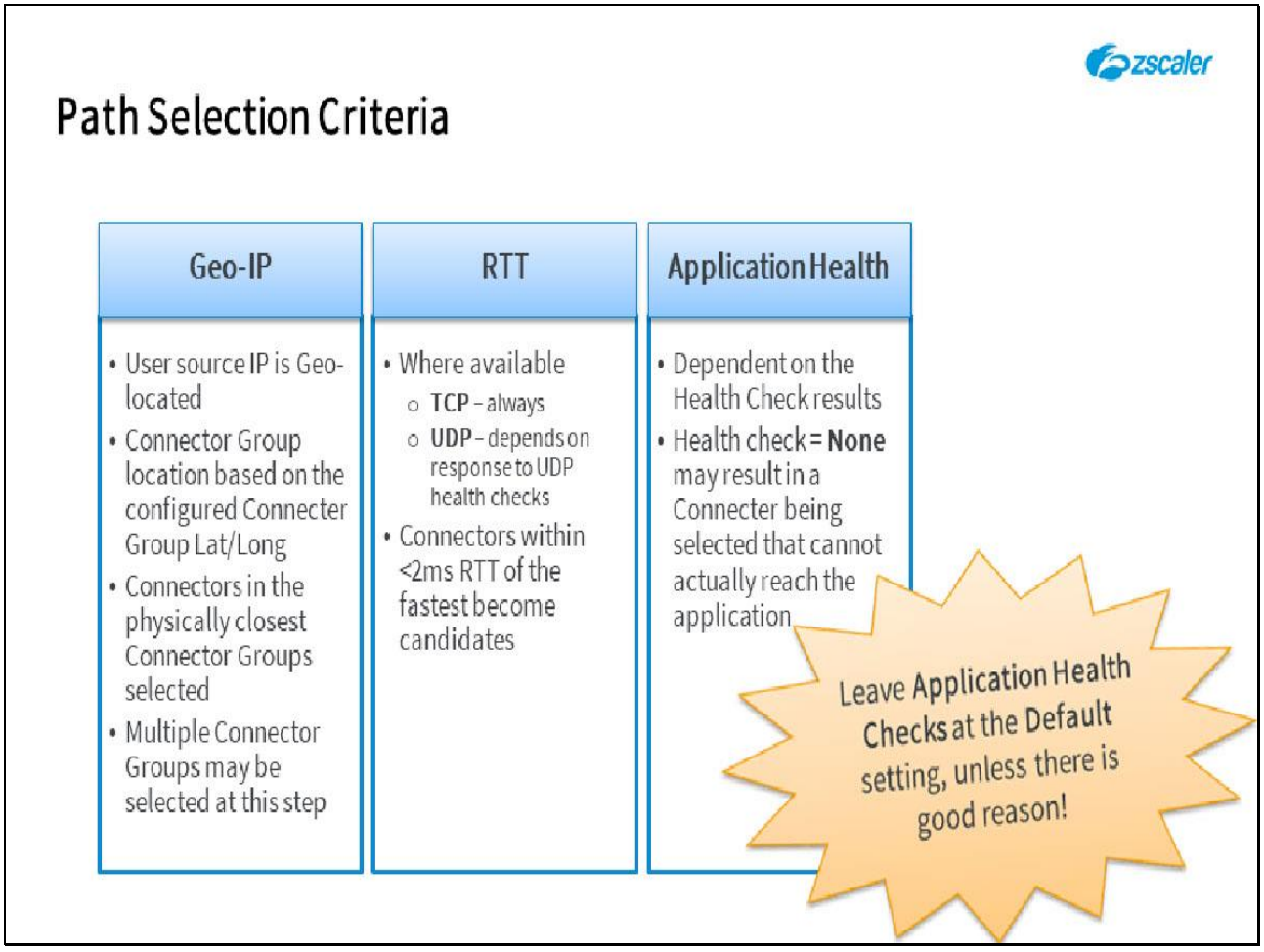


Slide notes

Next is **Application Health**, which is a measure that is dependent on the ZPA health checking configuration. Under normal circumstances, only Connectors that report that they can reach the application would even be considered.

Note that if you set **Health Check** to **None** in the **Application Segment** configuration, this means that the Connectors will not do any explicit health checks and will report the application as healthy with an RTT of **0**. This can lead to the situation where a Connector is selected that cannot actually reach the application.

Slide 75 - ZPA Application Discovery

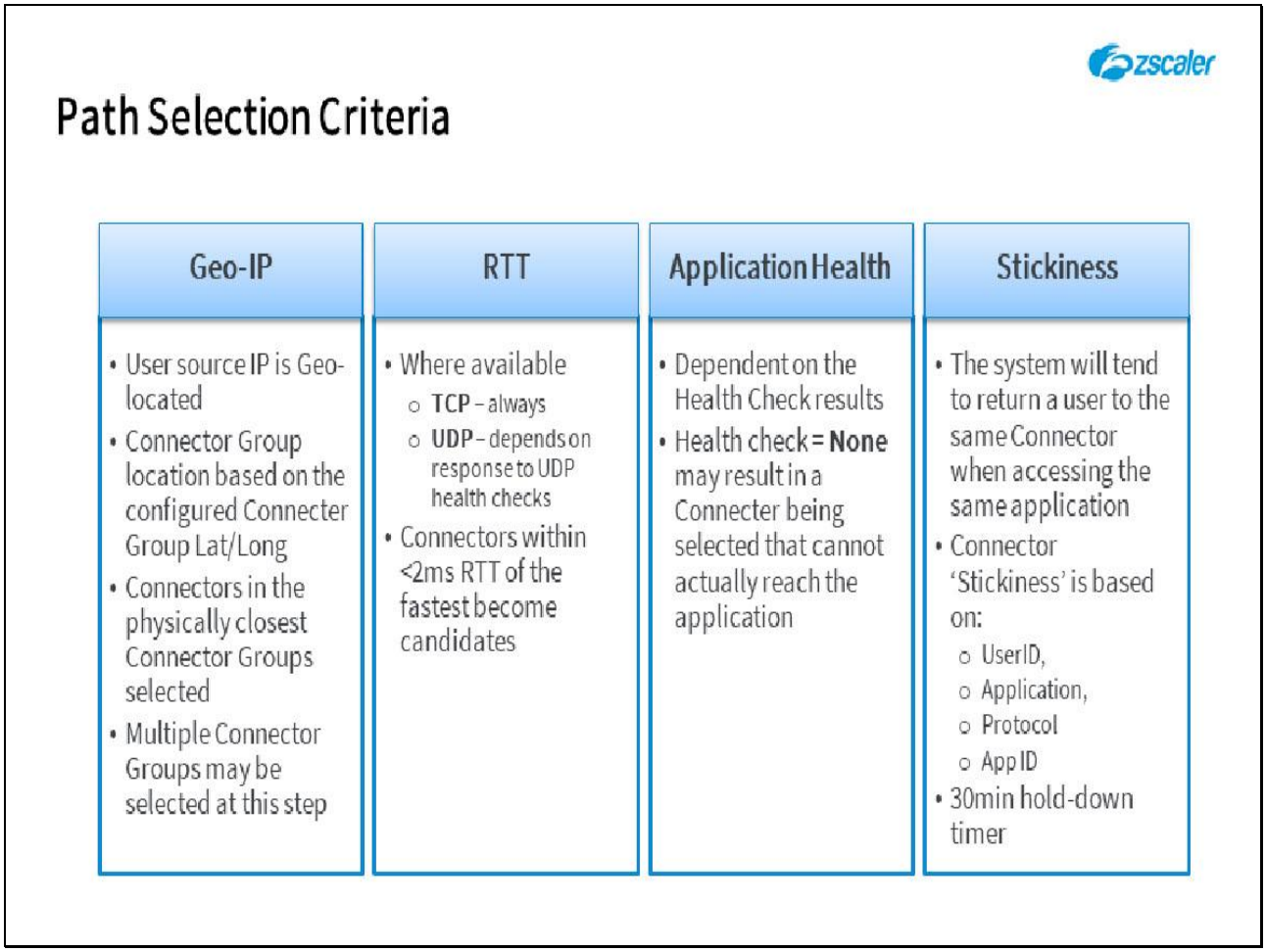


Slide notes

Which brings us to the next best practice for optimum user connectivity: Leave the **Health Check** setting in the **Application Segment** at the **Default** setting, unless there is good reason to change it.

A good reason to disable the health check is for some applications that open a port for a single connection and if this is used up by the health check, it is unavailable for actual application access.

Slide 76 - Path Selection Criteria

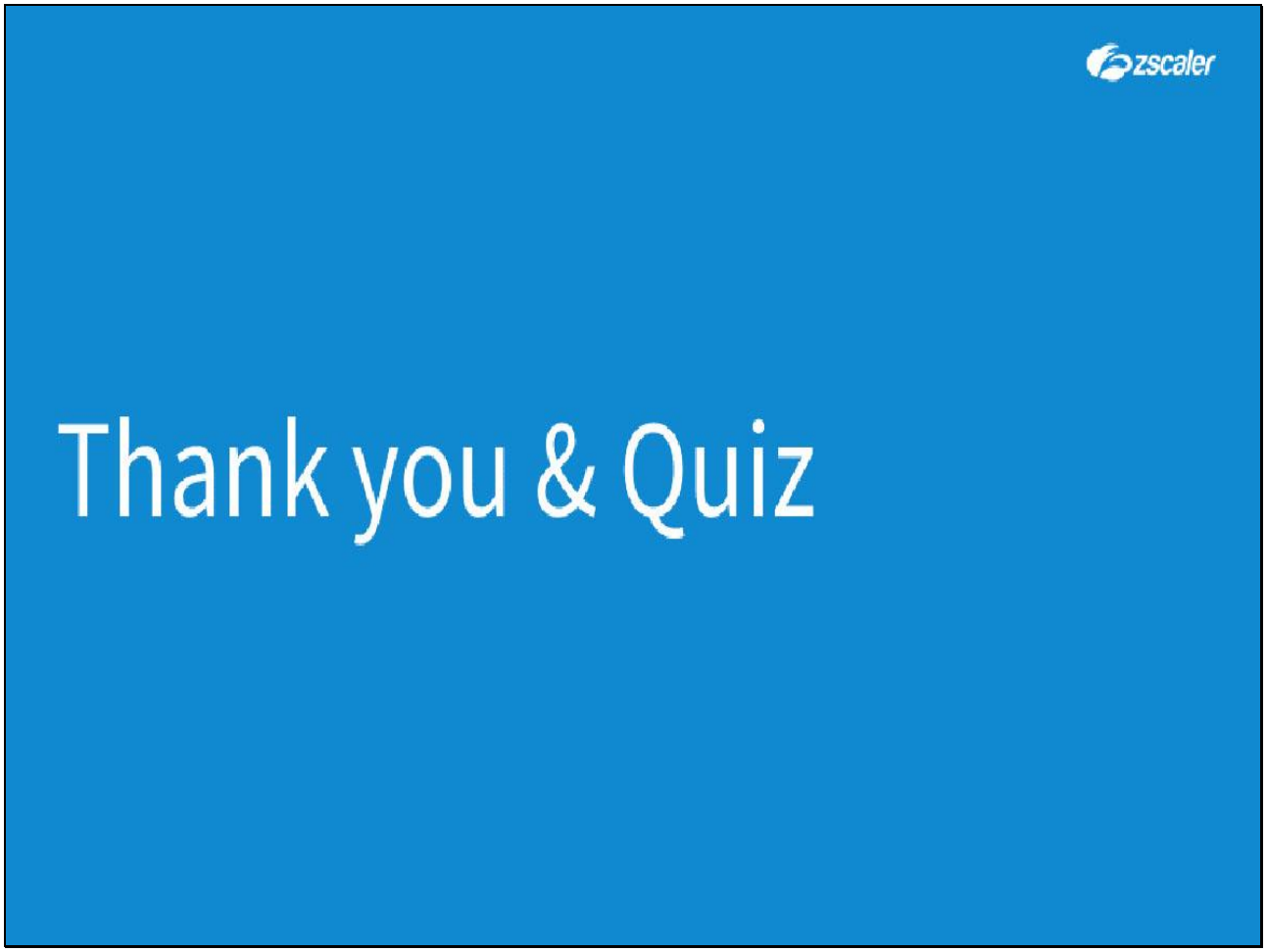


Slide notes

Finally, the system uses a 'stickiness' measure, so a user will generally return to the same Connector unless there is an appreciably better one available. This helps to prevent a user from bouncing between the available Connectors.

Stickiness is based on a number of connection attributes and has a hold-down timer of **30 minutes**. If the hold-down timer expires with no further user data, the next connection request will be evaluated as though it is a completely new request.

Slide 77 - Thank you & Quiz



Slide notes

Thank you for following this Zscaler training module, we hope this module has been useful to you and thank you for your time.

What follows is a short quiz to test your knowledge of the material presented during this module. You may retake the quiz as many times as necessary in order to pass.