The ultimate Citrix XenDesktop 7.x internals cheat sheet

Internal & external user authentication, the resource enumeration and launch process, VDA and XenDesktop internals and more.

Bas van Kaam

Let's take it one step at a time

I'll try and break it down phase by phase, starting with a brief FMA overview, followed by the user Authentication/enumeration process and take it from there. Here's what I want to cover during some of the next (mini) chapters:

1. Compact overview on the main components that make up the FMA.
2. Internal & external (NetScaler Gateway) user authentication steps.
3. The application, or resource, enumeration process.
4. What happens when a resource, desktop or application, is launched.
5. More specific, what happens inside the VDA during launch time.
6. ICA protocol stack breakdown, not really me taking but still.
7. A look at the main services that make up XenDesktop, or FMA if you will.
8. Some key takeaways!

1. FMA high-level overview

Just as a pre-appetizer, let's have a look at some of the main components that make up the Flexcast Management Architecture (FMA) today and, from a bird's eye view, how they interact and rely on each other before we get into some more detail on this.

1.1 Receiver

We'll start with Receiver; it's installed on the end user device and communicates with NetScaler, StoreFront and the virtual and/or physical machines in the data center over port 80 or 443. This can either be a VDI or HSD orientated architecture. It talks to StoreFront using the StoreFront Service API, again using ports 80 and/or 443. This API is primarily responsible for operations like, user authentication, enumeration, reconnecting, disconnecting, launching applications and desktops, power control and more. There is also a so-called StoreFront Web API a.k.a. Receiver for HTML 5, it provides access to applications and desktops using just a HTML 5 capable web browser, no locally installed Receiver is needed. Both API's act as a bridge between the StoreFront services, with the authentication service and the store service being the most important ones.

1.2 StoreFront

Within a XenDesktop Site you basically have two points of authentication, one of which is StoreFront and the other being the NetScaler Gateway. StoreFront communicates with Receiver, the Delivery Controllers and NetScaler (STA) with regards to external access. Optionally it can also be configured to communicate with AppController as part of a XenMobile deployment. It plays an important role in the application enumeration and resource launch (.ICA file) process and it functions as the main Store (there can be multiple) from which users subscribe to their desktops and applications.

1.3 NetScaler gateway

As mentioned above the NetScaler Gateway is primarily used (yes, it can do a lot, out of scope for now) to provide our users with secure external access to our XenDesktop sites. As we will see shortly there are some key differences when it comes to external and internal user authentication and the resource enumeration process. So in that respect you could say that is it part of the FMA.

1.4 Delivery Controller

Next up is the Delivery Controller; this is the real workhorse and centerpiece of the FMA. It brokers (VDA) sessions, verifies user credentials in AD when combined with StoreFront (note that I say verify and not authenticate, there is a difference between the two) and as such plays an important role during login and resource enumeration. It handles communications with StoreFront and/or Web Interface (XML), the underlying Host (Hypervisor) infrastructure, the central Site database and it also takes care of load balancing HSD connections, and as of version 7.6, it also includes Connection Leasing! You could say that the Delivery Controller is in fact the XenDesktop/XenApp installation. It houses all critical services, which in their turn control all major FMA components completely independent from each other. More details in a bit.

1.5 Virtual Desktop Agent

The Virtual Desktop Agent, or VDA in sort, is a piece of software that gets installed on all virtual and physical machines running a Windows Server and/or Desktop operating system in your XenDesktop Site (Citrix recently released a VDA for Linux as well) making their resources remotely available to users. The installed VDA software communicates with the Delivery Controller as well as with the Receiver software installed on the client endpoint.

When the connection is local (LAN) the VDA and Receiver will set up a direct (one to one) connection, if the connection is external, through NetScaler, traffic will flow from the internal network through NetScaler (SSL 443) on to the external client device. Or, if the client endpoint does not have Receiver installed it will, or can, use the StoreFront HTML5 based build-in Receiver instead. The earlier mentioned one to one connection will then run from the users browser (needs to be HTML 5 capable) to the virtual machine in the data center. It will of course traverse the NetScaler Gateway when the connection is external.

1.6 Host Infrastructure

The Host infrastructure is nothing more than you underlying Hypervisor hosting you virtual XenDesktop (VDI) and/or XenApp (HSD) machines. This can be XenServer, Hyper-V or VMware vSphere (ESXi). The host connection is configured in Studio and basically connects your Delivery Controllers to the Hypervisor, linking them together. From there on the Delivery Controller(s) will constantly communicate with the Host Infrastructure (through the XenDesktop Host Service) and tell it what to do, start up a VM for example, tell a VM that a new connection is coming, create new machines using MCS, log all configuration and/or machine connection state changes, monitor the infrastructure etc...

1.7 Site Database

The central (SQL) Site database. This is where all information, dynamic and static is stored. It holds all Site configuration information with regards to (HDX) Polices, Delivery Groups, Machine Catalogs etc. It's also responsible for storing all dynamic session information and as such it has a constant connection with the Deliver Controller(s). It needs to know everything that's going on.

If a Delivery Controller needs to broker a new session or reconnect an old one, this is where he (is it a he?) gets his information. As mentioned version, 7.6 introduces a new feature called Connection Leasing (see the link earlier), if for whatever reason the central Site database becomes unreachable, never mind if it's HA or not, users will still be able to login and use all their resources that they have accessed at least once during the past two weeks (default setting) as of that moment.

1.8 Citrix Studio

Citrix Studio, the main management console from where you configure and manage the biggest part of your XenDesktop Site deployment. Although there are several options and/or features that you can only configure using PowerShell, Studio has a build-in PowerShell prompt as well. However, if you use PVS for example, that's still a separate management console, the same applies to Director and StoreFront. The latter can be added to Studio as well but is only partly manageable. NetScaler is also separate, but then again, it isn't really part of the FMA, I guess it all depends on your point of view.

1.9 Citrix Director

I already briefly mentioned Director in the previous paragraph; it is Citrix's first line of defense monitoring and troubleshooting tool. It comes shipped with XenDesktop/XenApp. As of XenDesktop 7.x it has some of the former EdgeSight functionality build-in and if you own the proper licenses (XenDesktop and NetScaler) you can monitor your NetScalers (HDX insight) using the same console as well, again, based on EdgeSight technology. It offers some really nice overviews and statistics.

You can filter on failed machines and sessions, drill down deeper on a per user basis, see which applications are in use, shadow sessions, kill process and/or applications, log off sessions, reboot machines etc. On the main dashboard (in the administrative view that is) it also clearly displays the average login time and the time it takes per process during login, like GPO processing for example. It also monitors your Delivery Controllers on an ongoing basis using PowerShell under the hood.

1.1.1 ICA / HDX stack

There is a lot I could write about the ICA protocol, including HDX for that matter, but for now let's just say that without the ICA protocol stack it wouldn't be half as sufficient and smooth as it is this today. It is of course THE protocol used to communicate from client (Receiver) to virtual or physical machine (VDA), internally as well as externally.

Ok, that's it with regards to the FMA. Let's get to the fun stuff.

2. User authentication, internally and externally.

User authentication takes place internally, through StoreFront or Web-Interface, or externally through NetScaler (something that also needs to be configured on the StoreFront or Web-Interface server) which will than pass on the user credentials up to either StoreFront or Web-Interface.

2.1 StoreFront vs. Web-Interface

Before we go into the user authentication and application enumeration processes, a few notes on StoreFront and Web Interface first. Both products offer a different feature set, which I'm not going to fully discuss here, and handle certain functionality in different ways. One of which is how they take care of user authentication as part of the resource enumeration process, something that will automatically happen when someone starts a new session. Note that we are only referring to XenDesktop 7.x Sites here since StoreFront is also able to authenticate users to, and enumerate and aggregate resources from, XenApp Farms, XenMobile App Controllers or VIAB for example.

2.2 StoreFront

When using StoreFront users are authenticated by the authentication service, which is an integral part of StoreFront. Users can authenticate to StoreFront using different methods, using usernames and passwords for example or Domain pass-through, NetScaler pass-through, Smart card or unauthenticated user access. As soon as a user logs in by filing in his or her username and password (on the StoreFront web page using the so called Receiver for web configuration, or from within Citrix Receiver) the StoreFront authentication service will pick up the user credentials and authenticate them with a domain controller.

Once authenticated StoreFront will forward the user credentials, as part of a XML query, to one of the configured Delivery Controllers. Assuming you configured at least two of course. In between StoreFront will check its local data store for any existing users subscriptions and stores them in memory. Next, the Delivery Controller receiving the credentials will again contact a domain controller this time to validate the users credentials before responding back to the StoreFront server.

Note how I say user authentication and user validation! There is a distinct difference. Authentication is to make sure that somebody is who he or she claims to be. Verification is done to find out which resources are assigned (permissions) to the user, which will then be displayed in the users store, ready for subscription.

2.3 Web-Interface

If we look at Web-Interface, user authentication works a bit different, it has no internal authentication service. When a user logs in by filling in his or her username and password, Web-Interface will immediately forward these credentials, as part of a XML query, to a Delivery Controller where a domain controller will be contacted to authenticate the user before responding back to StoreFront. As like before, Web-Interface can also authenticate users to, and enumerate and aggregate resources from, multiple XenApp Farms and XenDesktop Sites, no App-Controller though.

So there you have it, similar solutions, same result, just another route. I'm aware that the Web-Interface XenDesktop 7.x combo probably won't be around much longer (Web-Interface will end of life in 2015) and even if it will, most newly build XD 7.x deployments are build using StoreFront anyway, so I doubt if there are that many to begin with. I just wanted to make clear the difference in authentication since this tends to confuse people from time to time.

3. Resource enumeration

When a user logs in for the first time, meaning that there are no active and/or disconnected sessions lingering around somewhere, right after the user is authenticated the resource enumeration process kicks in and will eventually show the user its assigned resources. That's why the user authentication process and resource enumeration basically go hand in hand. Let's put the two together and see what happens, I'll only use StoreFront in my example.

3.1 Applications subscription

Due note that when using StoreFront users will first have to select and subscribe to applications from the store before they show up on their main home screen and can be launched. When using so called Keywords, Administrators can pre-subscribe users to certain core applications so that their home screen won't be completely empty when logging in for the first time. This also works for assigned Desktops. Use Keywords:auto with the application and/or desktop of choice.

3.2 External user authentication through NetScaler

Let's take it step by step and see what happens when someone logs in externally. I'll assume that your NetScaler Gateway is set up and configured to integrate your StoreFront server(s), you have Receiver installed, SSL certificates are present, an STA / XML / Broker service address (Delivery Controller) and a domain controller for authentication purposes are also configured. Perhaps you want to load balance your StoreFront and/or Delivery Controllers by creating and configuring virtual load balance servers on the NetScaler, adjust the theme of the NetScalers Web Interface and last but not least, you'll probably have to configure your StoreFront deployment to accept pass-through authentication from NetScaler. Where port Nr. 443 (SSL) is used you can also use port 80, although 443 is recommended.

1. *A user opens up a web browser and connects to the external URL of the NetScaler Gateway (using SSL over port Nr. 443) here he or she will fill in his or her username and password. A locally installed Citrix Receiver can also be used to establish a direct connection to the NetScaler Gateway. Citrix Receiver uses so called Beacons to determine if a connection is internal or external and handles it accordingly. Check out the link.
2. The NetScaler will take the user credentials and authenticate them (session ticket) against Active Directory over TCP port Nr. 389. The NetScaler has its own authentication service just like StoreFront mentioned earlier.
3. Once authenticated the user session gets redirected to StoreFront where it will first perform a callback to the NetScaler that handled authentication to validate the user. The authentication details will then be send to the StoreFront.
4. From here the user credentials will be forwarded, as part of the earlier mentioned XML query, to the configured Broker (XML) service on one of the Delivery Controllers. Both these transactions will use port Nr. 443 / SSL).
5. In between, StoreFront will check its local data store for any existing recourse subscriptions and store these in memory.
6. The Broker (XML) service will again contact a domain controller (port Nr. 389) to validate the user credentials, note that this is different to the user authentication process, as we've established earlier. During this process it will find out to which security groups the user belongs.
7. With this information the Delivery Controller, or Broker (XML) service, will contact the central Site database to find out which resources have been assigned to the user. It does this over port Nr. 1433 / 1434
8. The Broker (XML) service will return an XML file to the StoreFront server including all assigned resources.
9. StoreFront will generate a web page containing all the assigned resources, which will be routed through the NetScaler Gateway and presented to the user. The users home screen will be populated with any pre-subscribed resources (Keywords). Depending on how you connected, your resources will be displayed either directly using a Receiver for web sites or you'll find them within the locally installed Citrix Receiver instead. The user will be able to browse its own personal app store for any assigned resources to which he or she can subscribe and then launch.

*If you don't enable authentication on the NetScalers login page the NetScaler will contact StoreFront and the user will be presented (through the NetScaler) with the StoreFront login page (Receiver for web sites). The user fills in his or her credentials and authentication will be handled by StoreFront.

3.3 Internal user authentication through StoreFront

So what happens when a user authenticates internally, directly to StoreFront? Let's have a look. Same rules apply here, use port Nr. 443 here you can.

1. A user opens up a web browser and connects to the internal StoreFront URL where he or she will fill in his or her username and password. This method is also referred to as Receiver for web sites as mentioned above (don't confuse this with the HTML 5 based Receiver for web, they're not the same). A locally installed Citrix Receiver can also be used to establish a direct connection to StoreFront, which is probably the preferred method whenever possible. The earlier mentioned (NetScaler) Beacon functionality applies here as well.
2. Next the StoreFront authentication service will pick up the user credentials and contact a domain controller to authenticate the user in Active Directory over TCP port Nr. 389. Here I'd like to note that if domain pass-through authentication is enabled on the StoreFront server, this step would automatically be skipped.
3. Once authenticated the user credentials, as part of the XML query, will be send to a Delivery Controller.
4. In between, StoreFront will check its local data store for any existing recourse subscriptions and stores these in memory.
5. During the next phase the Broker (XML) service will again contact a domain controller (port Nr. 389) to validate the user credentials, this is different to the user authentication process, as we've established earlier. During this process it will find out to which security groups the user belongs.
6. With this information the Delivery Controller, or Broker (XML) service, will contact the central Site Database to find out which resources have been assigned to the user. It does this over port Nr. 1433 / 1434.
7. The Broker (XML) service will return an XML file to the StoreFront server over port Nr. 443 / SSL.
8. StoreFront will generate a web page containing all the assigned resources, which will be presented to the user. The users home screen will be populated with any pre-subscribed resources (Keywords). Depending on how you connected, your resources will be displayed either directly using a Receiver for web sites or you'll find them within the locally installed Citrix Receiver instead. The user will be able to browse its own personal app store for any assigned resources to which he or she can subscribe and then launch.

4. The launch process

Here we basically pick it up were we left off at the end of the resource enumeration process as explained above. Just as with the authentication process, there are some differences between the internal and external resource launch process. Also, when launching a HSD or a published application there is an extra load balance step involved as well. Let's start with an external HSD launch trough NetScaler. Note that the below process is basically XenApp as we knew it earlier.

4.1 The Secure Ticket Authority

Before we continue... You might have heard about something called the STA, or the Secure Ticket Authority in full. It was first introduced with one of the earlier Secure Gateway editions over ten years ago. It (the STA) runs as a service and is part of the Broker service just like the XML service. During the resource launch sequence the StoreFront server as well as the NetScaler will both need to be able to communicate with the STA. As such you will need to configure the NetScaler as well as the StoreFront server(s) or Web-Interface server(s) to point to the same XML/STA service(s), or Deliver Controller(s).

Once a user launches a resource, externally through NetScaler Gateway, a secure ticket will be requested. As we will see shortly the STA ticket will eventually end up in the launch.ica file generated by StoreFront and/or Web-Interface. Once generated, the Delivery Controller hosting the STA service will hold the STA ticket information in memory for a configurable amount of time. As soon as a secure session is established the NetScaler Gateway responsible for handling the session only has to check the STA ticket (as part of the .ica launch file) with the STA service that originally generated the ticket. It (the STA service) does this from memory where the ticket was stored after it was created and send back to the StoreFront server as part of the XML file mentioned earlier. More (detail) on this in the overview below.

The STA is only used when traffic traverses a NetScaler, so you won't have to worry about the STA service and its tickets when authentication takes place internally.

1. Assuming that the login and enumeration process finished without any issues (see above) the user is free to subscribe to and launch any applications and/or desktops that have been assigned to him or her. As an example, let's say that the user tries launches a (XenApp) Hosted Shared Desktop session.
2. After the user clicks the icon the launch request is send to the NetScaler Gateway which from where it will be forwarded to the StoreFront server.
3. The StoreFront server will contact the Broker (XML/STA) service, or Delivery Controller, to find out if and where the resource is available and where it can be best started. This is where the well-known XenApp load balancing mechanism comes into play. Which as of FMA needs to be configured through policies.
4. During this time the StoreFront server will also request an STA ticket from the Broker (XML/STA) service. It will include the user, domain and resource

name it wants to start. It will also request a 'least loaded' server as part of the load balancing process.

5. The Broker (XML/STA) service will query the central Site database (ports Nr. 1433 and 1434) to find out which server is able to offer the requested resource. The Delivery Controller will use this information together with its load balance algorithm to decide which server to connect to.
6. At this time the Broker (XML/STA) service will create the STA ticket mentioned earlier. This will include information on the server and resource to connect to, as discovered in the previous steps mentioned.
7. Next the Broker (XML/STA) service will send this information back to the StoreFront server in the form of an XML file.
8. Based on this information the StoreFront server will then generate a launch.ica file (uses the default.ica file as a template) containing the STA ticket and a whole bunch of other connection properties that are, or might, be needed. This will also include the FQDN/DNS name of the NetScaler Gateway itself.
9. StoreFront passes on this information down through the NetScaler Gateway onto the locally installed Receiver, which initiated the connection to begin with.
10. *The locally installed Receiver will read and autolaunch the launch.ica file to set up a connection to the NetScaler Gateway (443 / SSL).
11. From here the NetScaler Gateway will first contact the Broker (XML/STA) service (this address is configured on the NetScaler as well) to verify if the earlier generated STA ticket, as part of the launch.ica file, is still valid.
12. The Broker (STA) service will validate the STA ticket from memory. Once verified it will send back the IP address, port Nr. Resource name etc. of the machine and the resource it needs to connect to and start. It gets deleted right after.
13. The NetScaler Gateway will set up a new ICA connection using port 1494 (ICA) or 2598 (CGP – Common Gateway Protocol) depending on config.
14. The installed VDA will verify its license file with the Delivery Controller.
15. The Delivery Controller checks with the Citrix License server to verify that the end user has a valid ticket. This will also be done for any Microsoft (CAL) licenses, with regards to HSD and published applications, that might be involved.
16. At this time any applicable session policies will be passed onto the VDA applying them to the session.
17. Finally the HSD is launched and the NetScaler Gateway acts as a proxy between the user and the XenDesktop resource in the data center.
18. Somewhere in between the session/connection information will passed on and registered in the central Site Database where it will be sued for future load balance purposes.

That's right, the STA ticket gets generated and send back after a user launches an application/desktop, not during the resource enumeration process.

*In my example I assume that you already have the Citrix Receiver installed locally, which is a pretty common scenario. But if you don't, you have a few options. First, when you connect to StoreFront, either directly or through the NetScaler Gateway as we've talked about, it automatically checks if you have a Receiver installed locally. If not, in most cases it will guide you to a download section or page (usually Citrix's) where you will be able to download the Citrix receiver. There are a few ways administrators can implement this.

4.1 HTML 5 to the rescue

If for whatever reason you are unable or not allowed to install a Citrix Receiver locally, Citrix offers the Receiver for HTML 5. You will still be able to connect to StoreFront / NetScaler and launch your resources without any loss of functionality. Although not enabled by default, StoreFront has a build-in HTML 5 based Receiver, which will kick in at launch time. It does this by fetching the HTML 5 engine from the StoreFront and making it part of the local browser. Note that you must use a HTML 5 supported browser for this to work. So basically your browser becomes your Receiver handling the launch.ica file. When you close the browser, you close the session. Even when your users will have Receiver installed you can enable it anyway as it will function as a fallback mechanism.

4.2 Broker, XML and STA.

Be aware that the STA (service) is also part of the Broker service, and has been as of Presentation Server 4.0. Before that it was written as an ISAPI extension for Microsoft Internet Information Services, or IIS. I also highlighted the so-called XML service multiple times. I put the XML and STA services between brackets because as of XenDesktop 4.x the XML service (ctxxmlss.exe) has been rewritten in .NET and became part of the Broker service. So the Broker service is actually build up out of three separate services, all handling different tasks, it brokers connections, it enumerates resources and it acts as the Secure Ticket Authority, generating and validating STA tickets.

Make sure that the Broker (XML/STA) service on the NetScaler and the Storefront server is configured identically. The same applies to the load balance/fail over order in which you configure them.

4.3 Launching an internal resource

Now that we've seen which steps are involved when launching a resource externally, a Hosted Shared Desktop in this case, let's have a look and see what happens when we launch a pooled VDI virtual machine internally. After this we will have looked at an external and internal resource launch, a HSD, which is comparable to a published application, and a VDI virtual machine. Again, user authentication and resource enumeration has successfully completed, here we go (again).

1. As mentioned we will launch a pooled VDI virtual machine this time. Let's assume that the VM is pre-subscribed and already present on the users

home screen, never mind which one: locally installed Receiver, Receiver for web sites or the HTML 5 based receiver as part of StoreFront.
2. After the user clicks the icon the StoreFront server will contact the Broker (XML/STA) service, or Delivery Controller, to check if any registered VDA's are available. It does this by communicating with underlying Hypervisor platform through the Host service on the Delivery Controller.
3. If needed it will start / boot the VM. It's not uncommon to pre-boot a few VM's, since, as you can probably imagine, this will positively influence the overall user experience.
4. Next the Delivery Controller, or Broker (XML/STA) service, will contact one of the VDA's and it will send a startlistening request. By default the VDA isn't listening for any new connections on port Nr. 1494 or 2595 until it gets notified that a user wants to connect.
5. As soon as the VDA is listening, the Broker (XML/STA) service will send this information back to the StoreFront server in the form of an XML file.
6. Based on this information the StoreFront server will then generate a launch.ica file (it uses the default.ica file as a template) containing the IP address of the VDA and a whole bunch of other connection properties that are, or might, be needed. This is send down to the user.
7. The locally installed Receiver will read and autolaunch the launch.ica file initiating a direct connection from the users end-point to the VDA.
8. The installed VDA will verify its license file with the Delivery Controller.
9. The Delivery Controller checks with the Citrix License server to verify that the end user has a valid ticket.
10. At this time any applicable session policies will be passed on to the VDA and the session is launched.

5. What happens inside the VDA during launch time?

Let's take it one step further and see what happens inside the VDA during launch time. The process below assumes that session reliability is enabled and that a desktop OS VDA gets launched as we've seen in the previous section, the below takes place somewhere around step 7. Remember that as of XenDesktop 7 there is also a server OS based VDA.

1. The CGP service will receive the connection and sends this information on to the tcpip.sys, which will forward it to the ICA stack.
2. The ICA stack will notify the ICA service a.k.a. the Portica service (Picasvc) that a connection has been made after which the Picasvc will accept the connection.
3. Then the ICA service will lock the workstation because the user needs to be authenticated to ensure that the user is allowed access to that particular machine.
4. As soon as the user logs on to the workstation the Portica service will communicate with the display manager to change the display mode to remote ICA, this request will be forwarded to the Thinwire driver.

5. In the meantime the Portica service will hand over the 'pre logon' ticket data, which it received from the ICA stack, up to the Desktop service and from there back to the Delivery Controller in exchange for 'real' credentials.
6. The Desktop service receives the users credentials, which are send back to the Portica service.
7. The Portica service contacts the authentication service to logon the user and this is sort of where the process ends.

It's kind of hard to find information on these kinds of traffic flows. I was fortunate to some across an older Citrix Synergy presentation, which handled the subject in great detail, helped me a lot. Do note that with the newer XenDesktop 7.x versions things might have slightly changed, but I still think this should give you a good idea on what going on internally. It was presented by Karen Sciberras by the way a.k.a. @XDTipster on Twitter, currently she's deep into XenMobile.

5.1 So what about the server OS VDA?

Well, there is not that much to tell to be honest, information is scares. There are a few (older) Citrix blogs which mention the new server OS VDA and how it's supposed to be kind of comparable to the former XenApp 6.5 worker only role (remember that one? It couldn't be elected as a Data Collector) only way more efficient. It has been built from the ground up and it should have a lot of similarities with its little brother, the desktop OS VDA. Or is it big brother? It is a lot older.
Anyway, here's what they wrote about it when they were on the verge of releasing XenDesktop 7, we have this:

- The Excalibur Delivery Agent for Windows Server Machines is designed from the ground up for dynamic provisioning with Machine Creation Services and Citrix Provisioning Services. Unlike XenApp, the Delivery Agent communicates only with the controllers in the site and does not need to access the site database or license server directly.
- Delivery Agents do not need to run the same version or OS as the controllers. This simplifies the process of upgrading sites, and allows Excalibur to support six different Windows operating systems all within a single farm: Server 2008 R2, Server 2012, XP, Vista, 7, and 8.
- While the XenApp session host-only mode disabled functionality and offered performance benefits over XenApp controllers, it consisted of the same installed services and same binaries as the controllers. The Excalibur Delivery Agent for Windows Server Machines is lighter weight, and only consists of the components needed for hosting sessions. It does not share any common installed components with the controllers.

And then we have:

The biggest difference between the two Delivery Agents is the ICA protocol stack. For desktop machines, Citrix ships a single-user ICA stack (internally known as Portica) which allows only one ICA session at a time. This version connects users to the machine's console session, similar GoToMyPC or other Remote Access products for a Desktop OS. It also includes additional HDX features such as USB and Aero redirection, which are only available on a single-user machine. For server machines, Citrix includes a multi-user ICA stack, which extends Windows Remote Desktop Services with the HDX protocol. This is the same ICA protocol stack developed for Citrix XenApp, just with a different management interface to make it compatible with Excalibur controllers.

Which also sounds familiar, although I'm not too sure on the server OS VDA ICA stack though. Like I mentioned before I couldn't find any real useful information on the internals with regards to the server OS VDA. But... I got a lot of help from Mick Glover a.k.a. @XDTipster on Twitter (thank you Mick) so I wouldn't be surprised if I can update this article with some new and interesting facts not to long from now. Stay tuned.

6. The ICA protocol internals.

I thought hard on what to do with this one, I mean, there is just so much to still learn and tell with regards to the ICA protocol and all that surrounds it, I wouldn't even know where to start. But then I came across this presentation given by Dennis Gundarev, a.k.a. @fdwl on Twitter, at BriForm this year. That's it, it will tell you all you need to know and more, make sure to give it a look.

Since a great deal of the communication flow within the ICA protocol, like graphics, printers, audio, smart cards and what not, takes place through virtual channels (inside ICA) I thought I'd throw this one in as well.

7. A (deeper) look into XenDesktop 7.x

Have you ever wondered what it is that keeps XenDesktop spinning once we've installed our first Delivery Controller and Studio along side it? Which service is responsible for interacting with our virtual machines, how new machines are created, how delegation of control is managed and a few more of those day-to-day tasks that we normally take for granted? During the next few sections I'd like to highlight a few of the lesser-known concepts around Citrix XenDesktop.

> I think we should just call it FMA from now on and use the technology stacks, XenDesktop (VD) and XenApp (HSD), to indicate what we offer to our users.

The Delivery Controller as part of XenDesktop is the hart of the FMA. It communicates with Studio, Director, the central Site database and the VDA's installed on the virtual and physical machines in our datacenter. Once installed it is build up out of ten primary services that take care of almost everything from there on. They communicate with the underlying Hypervisor and the virtual machines on there, the creation of new machines, delegated administration, configuration logging, service interaction and much more. Here they are:

1. Broker service – Probably the bet known one. It brokers new sessions, handles resource enumeration and the creation and verification of STA tickets.
2. Configuration service – Although most people probably don't know it, this is probably one of the most important services in there. It makes sure that all services are able to communicate with each other.
3. ID Identity service – It handles all Active Directory computer accounts related to XenDesktop virtual and physical machines.
4. Configuration Logging service – It monitors and logs all configuration changes made within a XenDesktop Site, including all Administrator activity.
5. Delegated Administration service – This manages the configuration of all delegated administrative permissions.
6. Machine Creation service – Handles the creation of new virtual (not physical) machines.
7. Host service – Manages all connections between the hosts and the underlying Hypervisor.
8. Environment Test service – Manages tests within your XenDesktop Site, can be initiated from Studio for example.
9. Monitor service – Monitors the overall FMA architecture and produces alerts and warnings when it finds something is potentially wrong.
10. StoreFront service – Manages your StoreFront deployment.

Now that you know the services that make up a XenDesktop Delivery Controller, you can probably imagine how all these services will, and need to, interact to get keep things going. This is one of the main reasons why the Configuration service is such an important one, it handles all inter service communication.

All other FMA/XenDesktop services need to register with the configuration service on startup so that it knows they are all good to go.

7.1 They are all independent

All FMA services run completely independent from each other; if one goes offline it won't directly affect any of the other services. Although each service points to the central Site database, they all have an independent location in the registry, meaning that their connection strings to the database are all stored separately from each other, eliminating any single point of failures.

All FMA services run under the NT AUTHORITY\Network service account. Also, when authenticating to the central Site database, all service use the computer account that they are running on.

7.2 XenDesktop (Site policies) using PowerShell

The states of your FMA services are best checked using PowerShell. Using some of the PowerShell Get- commendlets when checking up on your FMA services will show you exactly what is going on, when and if something is wrong. It's much more detailed then using the services.msc window for example.
If you have a central management server I suggest you create a personal PowerShell profile and include some of the basic Get- FMA service checks in it. This way, every time you open PowerShell these basic checks will be done automatically before you continue. If you look at Director for example, on the main dashboard, there you also see your Delivery Controllers listed at the bottom of the screen. If all is well, green checkmarks pop up next to them. This is also PowerShell issuing Get-commands in the background.

Here are a few examples to check some of the more important FMA services:

- Get-BrokerServiceStatus
- Get-ConfigServiceStatus
- Get-HypServiceStatus
- Get-AcctServiceStatus
- Get-ProvServiceStatus

Another thing to mention is that both Studio and Director run on top of the PowerShell SDK as well. Everything you can do within Studio can also be done through PowerShell, including a whole bunch of configuration options and tweaks that are not possible using 'just' Studio.

When you check your Delivery Controllers in Studio, you'll see a number in minutes next to each Controller that indicates when the Delivery Controller has last registered itself with the central Site database. This number should always be 0. By default the Controller checks in every 20 seconds, which will then be valid for another 40 seconds.

7.3 Site policies

When we publish resources, either Hosted Shared Desktops, VDI based virtual machines or just some applications on their own, we normally would use a combination of Catalogs and Delivery Groups to grant or allow access to these resources. Although this works fine in most cases, using PowerShell we can get a bit more granular.

Entitlement policies – These apply to pooled and shared desktops. With entitlement policies you can explicitly deny a certain user from a group of users access to a

pooled and/or shared desktop. So you have a group of let's 50 users and you want to exclude five users, using these policy rules you won't have to create a separate group of users to exclude, you can just exclude those five users without affecting any of the other users. There are two Entitlement polices:

1. BrokerEntitlementPolicyRule, this one issued for access to desktops.
2. BrokerAppEntitlementPolicyRule, this one is used to control access to applications.

Assignment policies – These basically do the same thing as the Entitlement policies described above only they apply to dedicated private desktops. Again, there are two policies:

1. BrokerAssignmentPolicyRule, this one issued for access to desktops.
2. BrokerAppAsignmentPolicyRule, this one is used to control access to applications.

Before we move on to the Site Access policy I'd like to point out another Entitlement policy gotcha:

Once we've configured a Delivery group with the Desktops and Applications delivery type we can use PowerShell to limit access to the HSD. Let me explain what I mean here. By default, when you create a delivery group with the delivery type set to Desktops and Applications, Studio creates one Entitlement Policy Rule and one App Entitlement Policy Rule for the group, meaning that each user is entitled to one desktop session and one app session. Studio doesn't expose the user filter on these objects, so both are available to all users of the delivery group.

Using the PowerShell command: Set-BrokerEntitlementPolicyRule we can change this behavior. It can set the *IncludeUserFilterEnabled* parameter to True instead of False, enabling the user filter, and it also lets you add in an AD security group, this way limiting access to just that group and that group alone as apposed to all users that are a member of the Delivery group.

7.4 Site Access policies

Site Access policies – This isn't directly about the users connecting, it is more about connections in general and the conditions that need to be met once a connection gets established, things like client IP addresses, the protocol used, Smart Access filters, host names etc. Based on this information connections can be excluded or denied access as well.

When a Delivery Group gets created two access rules are created and added by default, one for direct connections and one for connections through NetScaler. Using PowerShell we can look at and change these access rules, as we feel fit. To see what your Site Access policies currently look like, open PowerShell and type:

Get-BrokerAccessPolicyRule followed by the –DesktopGroupName command, so that you won't get overloaded with all Access policies currently enabled. To edit these policies you will use the Set-BrokerAccesssPolicyRule command just like you would with the earlier mentioned by the way.

8. Key takeaways

- Within a XenDesktop 7.x Site there are basically two points of authentication, the StoreFront server (internally) and the NetScaler Gateway (externally).
- The STA is only used when traffic traverses a NetScaler, so you won't have to worry about the STA service and its secure tickets when authentication takes place internally.
- StoreFront directly authenticates the user with a Domain Controller. Web Interface passes on the user credentials to the Delivery Controller where it will authenticate the user against Active Directory.
- User authentication and user validation! There is a distinct difference. Authentication is to make sure that somebody is who he or she claims to be. Verification is done to find out which resources are assigned (permissions) to the user, which will then be displayed in the users store, ready for subscription.
- Do not confuse the HTML 5 based Receiver with the Receiver for web sites configuration; they're not the same!
- Once authenticated and resource enumeration is completed the users home screen won't be automatically populated with resources unless Keywords (on the published resource properties) are used.
- If you don't enable authentication on the NetScalers login page the NetScaler will contact StoreFront and the user will be presented (through the NetScaler) with the StoreFront login page. The user fills in his or her credentials and authentication will be handled by StoreFront.
- The STA ticket gets generated and send back only after a user launches an application/desktop, not during the resource enumeration process.
- Both the STA and XML service are part of the Delivery Controllers Broker service. So the Broker service is actually build up out of three separate services, all handling different tasks, it brokers connections, it enumerates resources and it acts as the Secure Ticket Authority, generating and validating STA tickets.
- Make sure that the Broker (XML/STA) service on the NetScaler and the Storefront server is configured identically. The same applies to the load balance/fail over order in which you configure them.
- When connecting externally through a NetScaler Gateway, the launch.ica will include the FQDN / DNS name of the NetScaler.
- The Broker (STA) service will validate the STA ticket from memory. Once verified it will send back the IP address, port Nr. Resource name etc. of the machine and the resource it needs to connect to and start. It gets deleted right after.

- XenDesktop is made up out of ten primary services in total; the Configuration service is probably the most important service of them all.
- The Configuration service makes sure that all other services can communicate with each other. A service broker if you will.
- All other XenDesktop service need to register themselves with the Configuration service at startup.
- All FMA services run completely independent from each other, if one goes offline it won't directly interfere with one of the other services. Each service points to the central Site database, however, they all have an independent location in the registry, meaning that their connection strings to the database are all located separately from each other, eliminating any single point of failures.
- Make sure you know about the Site Entitlement and Assignment policies, including the Site Access policies and how to view and edit them using PowerShell.
- Check your FMA service using PowerShell; try to automate this process.

9. Round up

That about rounds it up for now, I hope you found this somewhat informative and if you have any additions or other remarks do let me know. As always I loved putting this together and it taught me a thing or two as well. Again a big thank you to Mr. @XDTipster for backing me up on some of the content discussed.