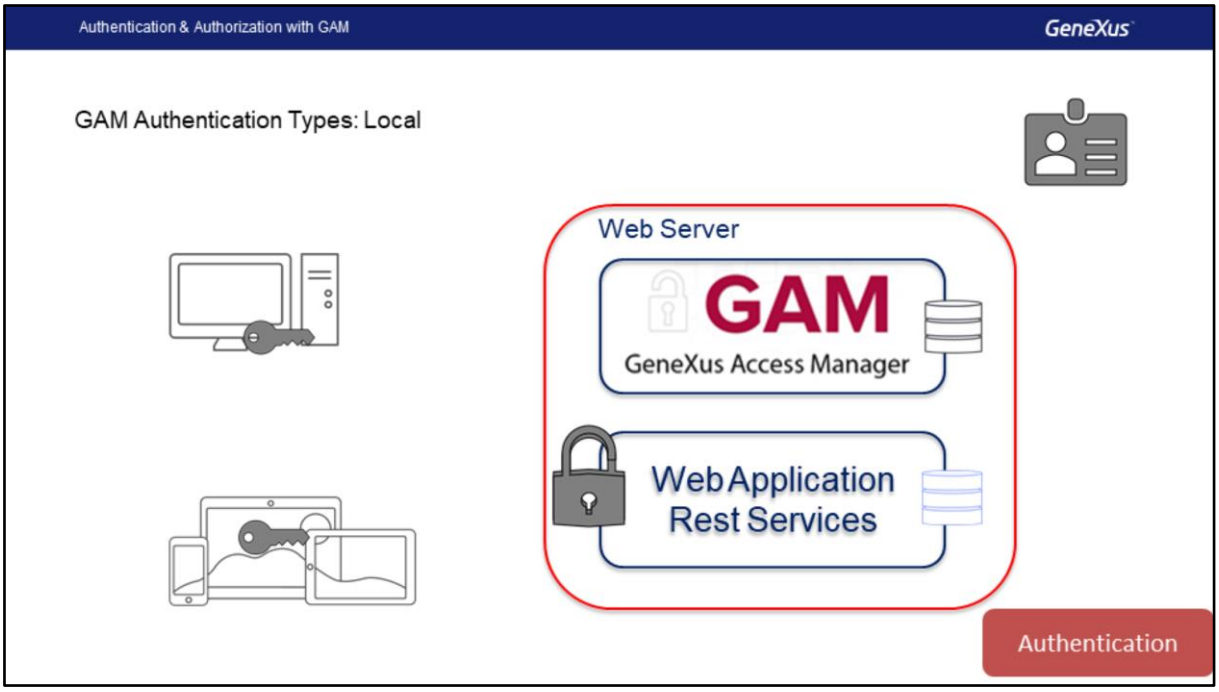

GeneXus™
The power of doing.

Authentication & Authorization with GAM

Security

GeneXus™ 16

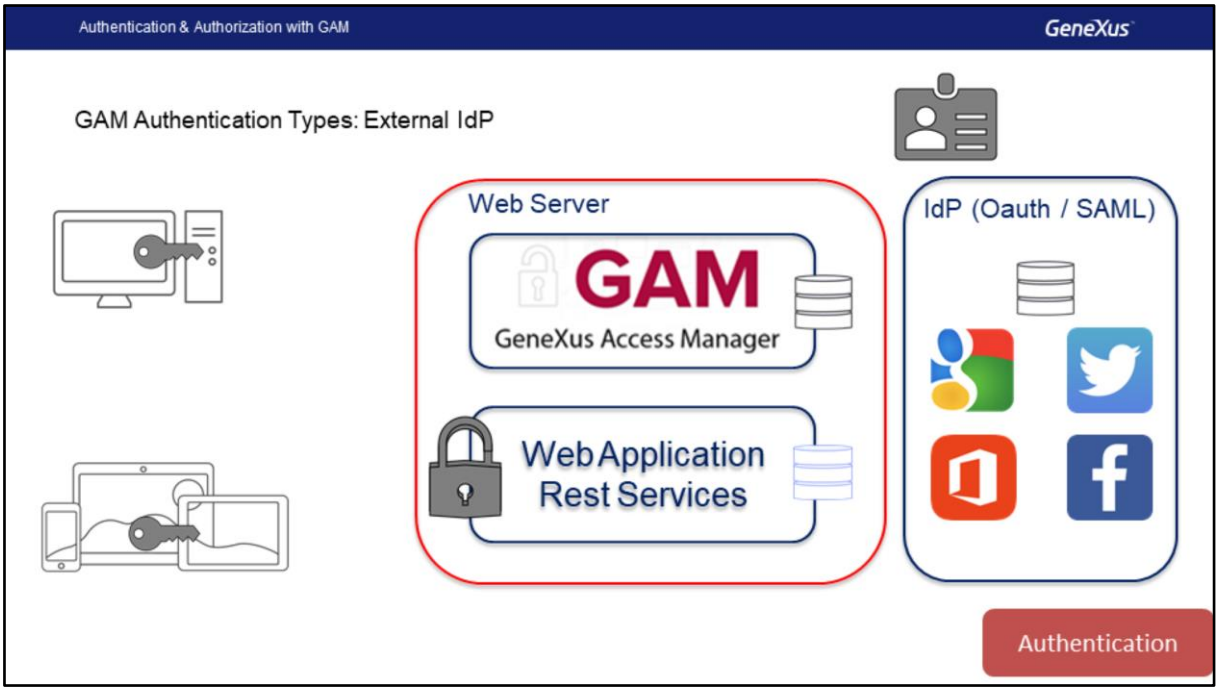
This video will show us further details on the features of GAM Authentication and Authorization, in addition to a demo where we will be using the web backend that this tool provides us with.



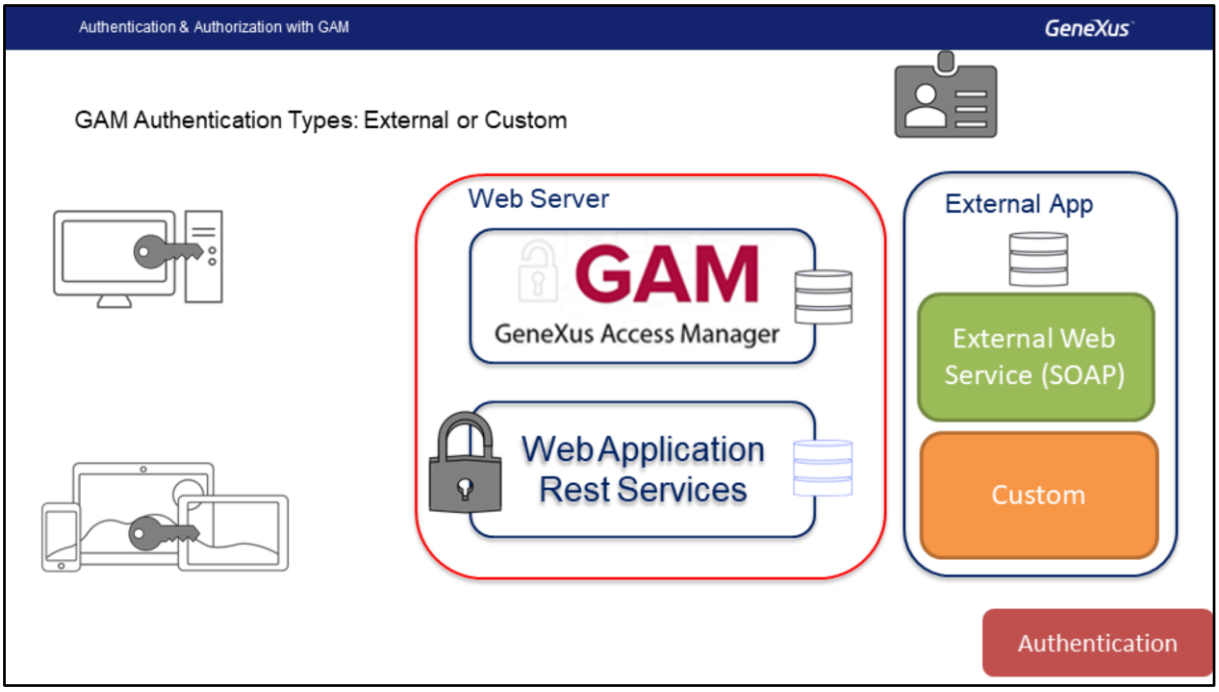
Authentication is the process to verify that a user is actually who she/he claims to be, done with the validation of credentials. In the case of Gam, credentials imply user and password.

We have different types of authentications for implementing, and it's also possible to enable different types simultaneously.

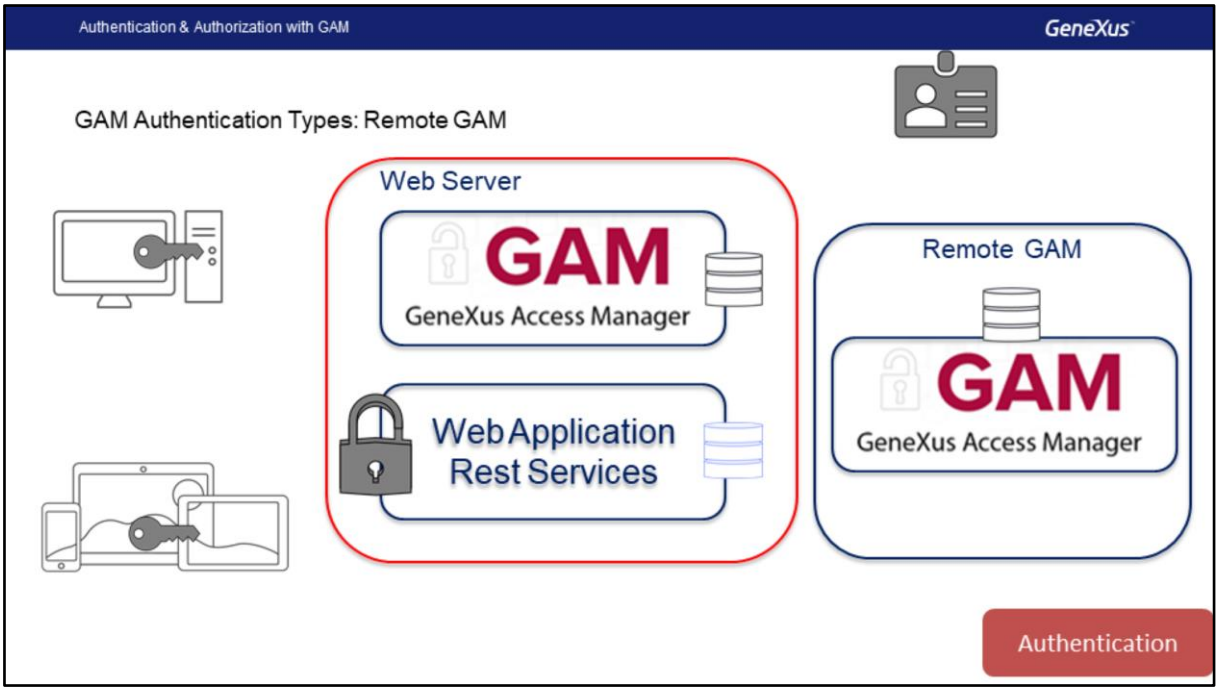
The types available are:
Local: in this case, the user's credentials will be stored in the GAM database in the users table. Due to security reasons, passwords are not stored. Instead, a Hash obtained in the application of an SHA-512 algorithm to the password entered by the user will be stored. Every time that credentials are validated that Hash is calculated for the password entered by the user, and it is compared against the Hash stored in the table. Among other things, this means that we have no way of recovering the user's password value, nor manipulate it in any way, because the process is non-reversible, so we will not be able to obtain the string that initially generated a hash.



Another option is to use an External Identity Provider supporting Oauth 2.0 or SAML 2.0: we may use several IdP (Identity Providers) such as Google, Facebook, Twitter, and Office 365, among others. In those cases, the GAM base will only store the user ID in the users tables. This is applied in, for example, assigning a ROLE to a user, where the user's credentials will be managed by the IdP.
Upon his/her authentication, the user will be redirected to the IdP, where he/she will enter his/her credentials. When they are satisfactory, the IdP will return to the site again.
The differences between Oauth and SAML deal with the technology used, and with the flow. However the options are similar in what concerns credentials, because these are just entered in the IdP, which, following the verification, returns the control to the system that required the validation. In all cases it is necessary to have public URLs to achieve the redirections necessary.



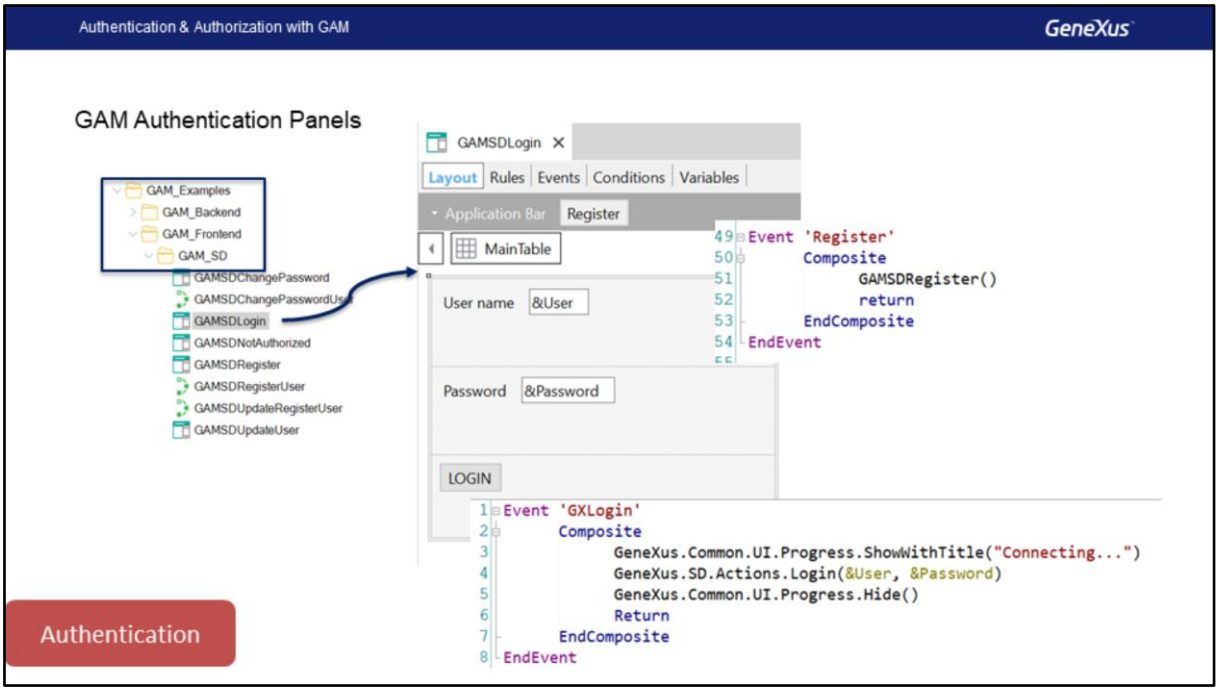
For the case of using external authentication, GAM is set up to interact with an external provider that uses web services or other customized mechanisms.
As in the previous case, only the user's minimum data is stored in GAM, because the validation of the access credentials takes place in a different system.
In these cases, GAM provides us with facilities to map the roles defined in GAm to external roles.



We may also use Remote GAM because GAM is actually an Identity Provider itself that will manage the user's credentials, so we may configure an app that uses GAM to validate the user's credentials in another GAM instance that will fulfill the IdP role.

For further information on authentication types you may access our wiki, where you will find plenty of information as well as use cases and detailed examples.

https://wiki.genexus.com/commwiki/servlet/wiki?15937

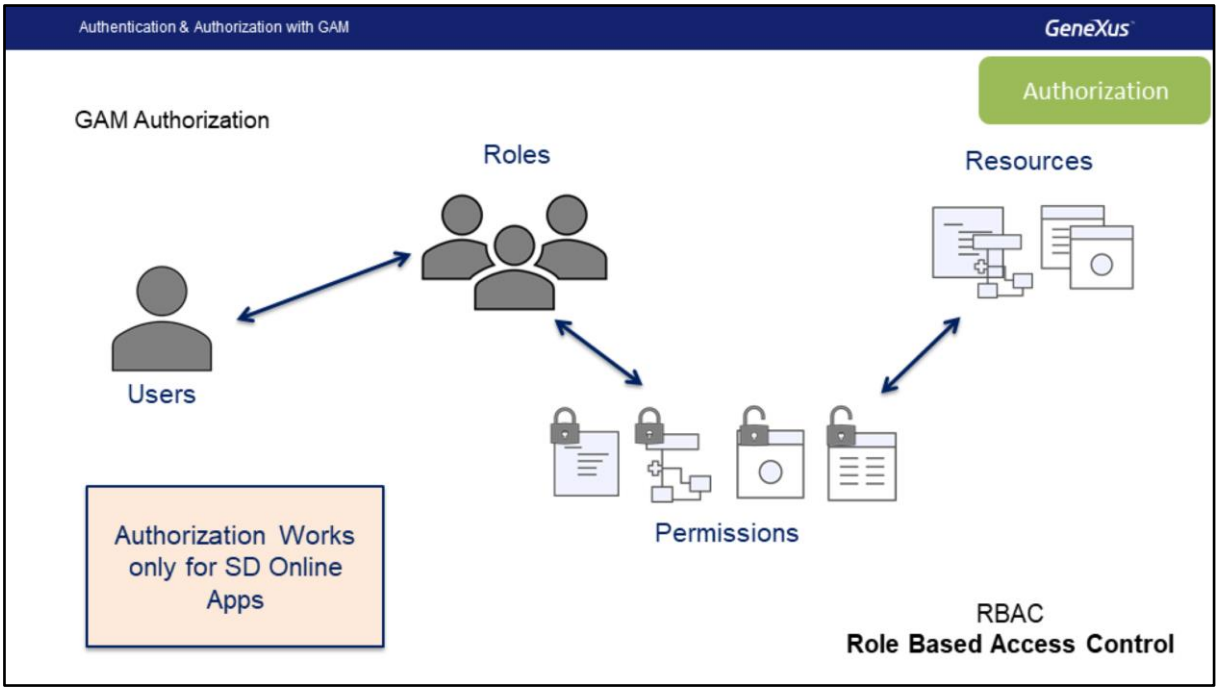


Since part of the GAM authentication mechanism will provide us with two objects –a Web Panel and a Panel for Smart Devices already programmed-, these panels may be customized if desired.

Specifically, inside the Gam Examples Folder, we have the Gam_FrontEnd Folder, and inside this other folder named GAM_SD there are objects that solve the Login, the change of password, the registration of new users and the update of user data.

For example, this is the Login panel, GAMSDLogin, which implements events to do the login and the registration of new users.

An important aspect in this panel is that in the case of an Offline SD app, the panel will have to execute Online, meaning that the user must have a connection to the server in order to have access.



With GAM, we can also solve the authorization, which implies the process of verifying that a user that was already authenticated has the permits required to perform an action within the system.

To this end, GAM has a schema based on User Roles; in GAM, each user has one or several associated roles. We will also have the Resources ensured, as well as the assignment of Permits on the Resources to the Roles.

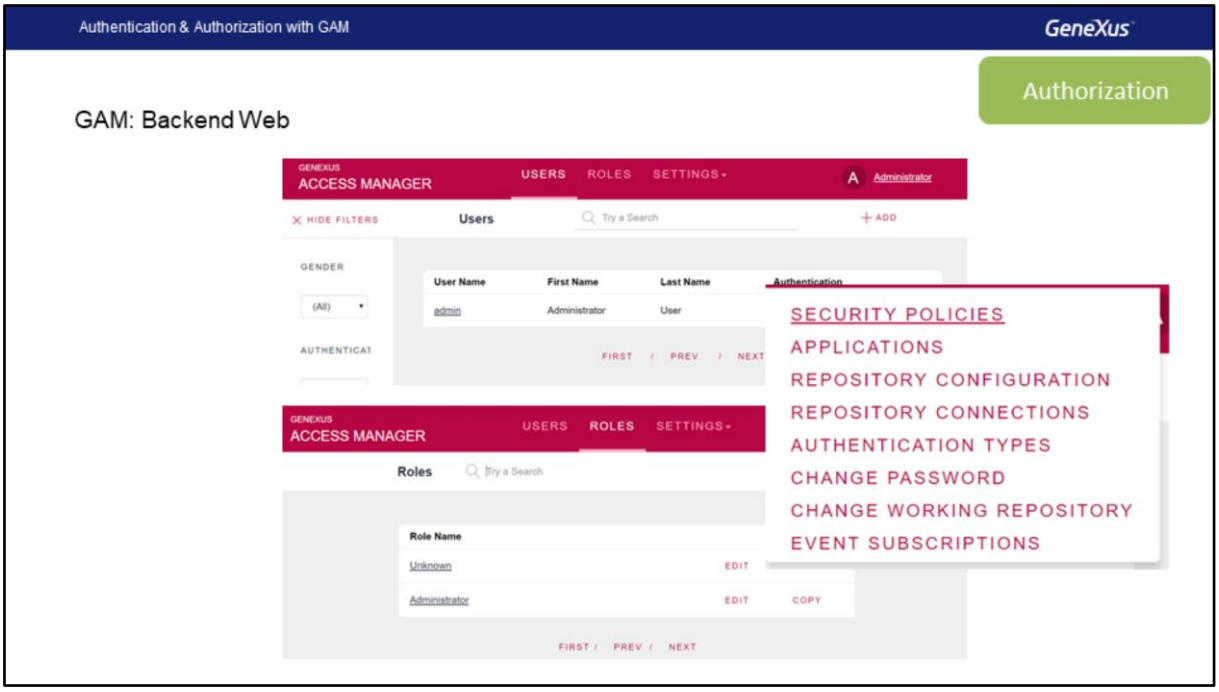
The resources that we may ensure are:

- Web Panels
- Web Components with Access through enabled URL
- HTTP Protocol processes such as reports with an output to PDF
- WEB transactions; additionally, in this case we may execute and also customize the Insert, Update and Delete modes or provide Full access to a transaction, that is: execution and all the modes.

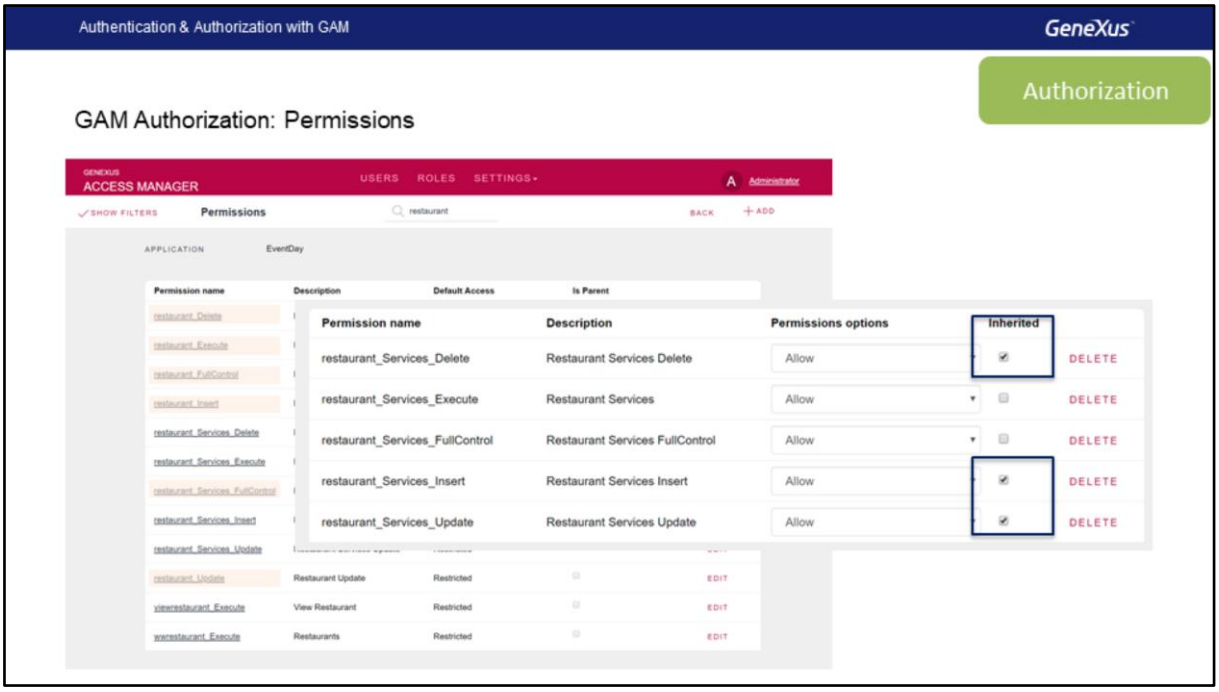
For the case of ONLY SD apps, the resources to ensure are:

- Panels for Smart Devices
- WorkWith for Smart Devices; in this case the permit relates to execution and actions on the transaction are handled on the transaction exposed as business component.
- Processes or Data Providers with Rest protocol.

For the case of Offline SD apps, we will only have the authentication, because when the app is offline we cannot keep our permits because, if we modify them, some devices may not be synchronized, so the schema is not viable. As mentioned, the Login panel, GAMSDLogin, must always execute OnLine; it must be connected to access the device so that the credentials are validated. After the user is authenticated, it may function disconnected until the expiry of the session initiated.

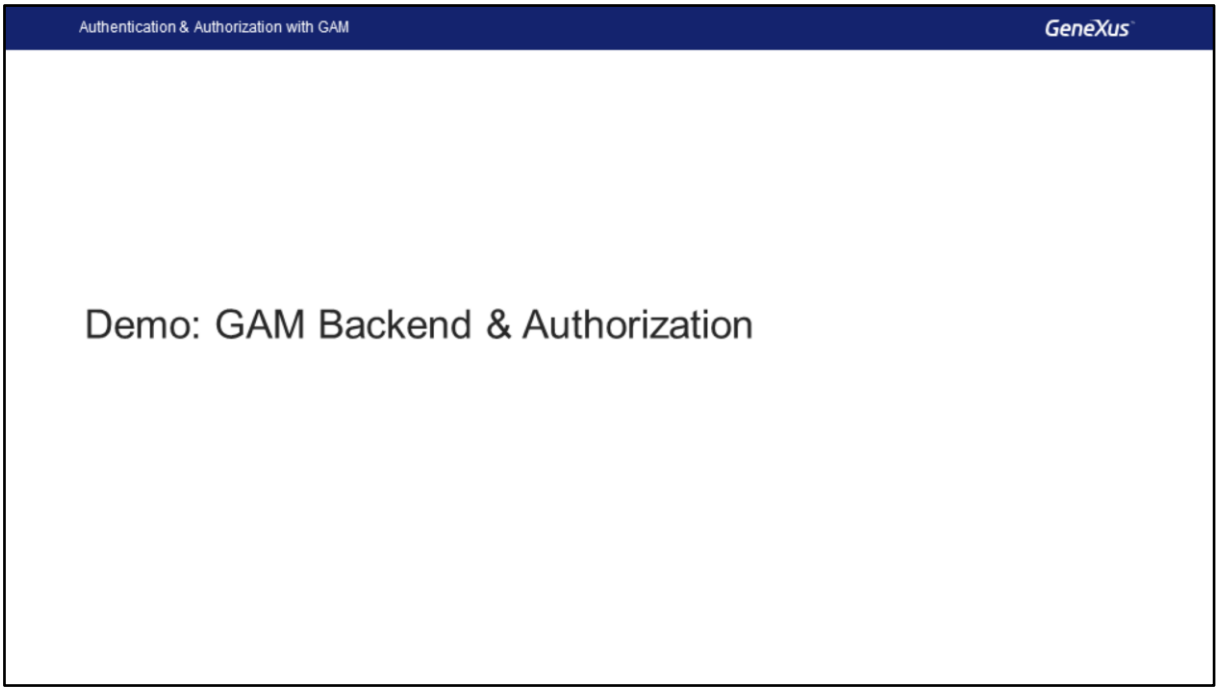


In order to manage all this data, GAM also offers us a web backend that will enable us, as security managers, to administer users, roles, permits and other configurations in the application, such as authentication types and other configuration parameters.



One of the facilities we have with GAM is that, for each application, it will generate the resources for which permits are to be granted later.

In the image we see those generated in relation to Restaurant.
There, we can see that on one side we have the permits of the Restaurant transaction, one for each mode (insert, update, and delete), in addition to execution, and another one indicating FullControl.
We also see that there are resources called Restaurant_Services that refer to the transition when it is used as BC and exposed as REST; and it is also the prefix used in the object WorkWith For Smart Devices.
When a role is authorized with FullControl, we grant, on that transaction, all permits, execution and each of the modes that will be shown as Inherited.



We will see al this in GeneXus.


GeneXus™
Videos
training.genexus.com
Documentation
wiki.genexus.com
Certificactions
training.genexus.com/certifications