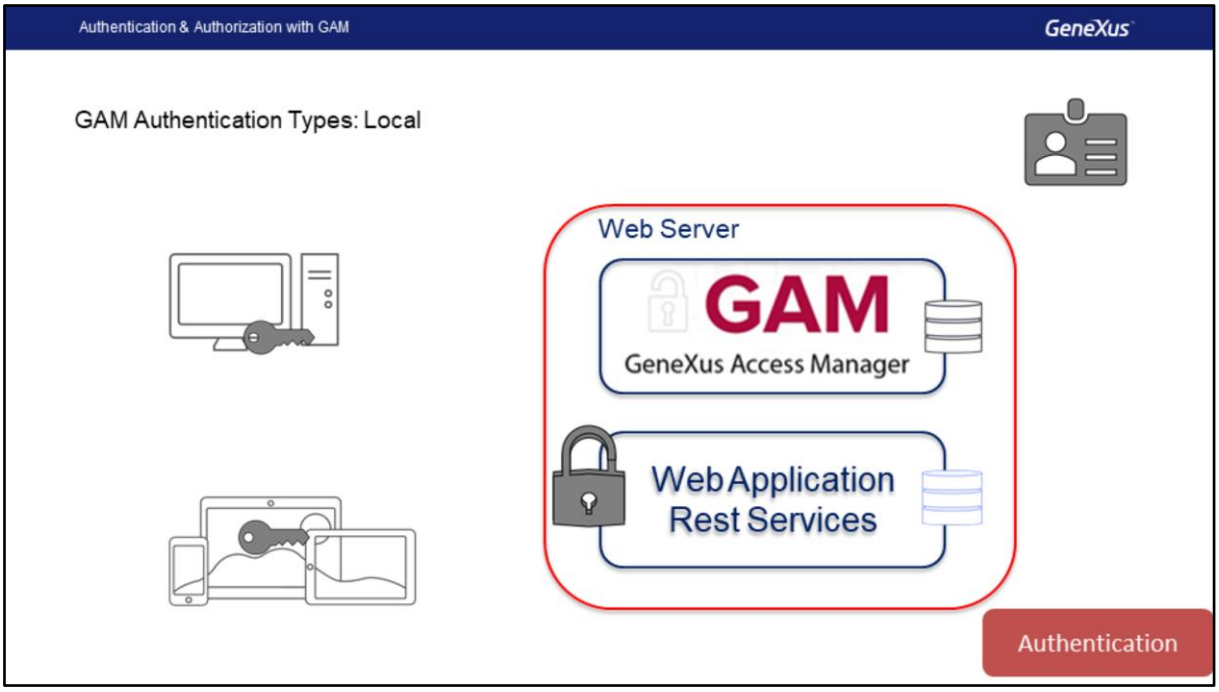

GeneXus™
The power of doing.

Authentication & Authorization with GAM

Security

GeneXus™ 16

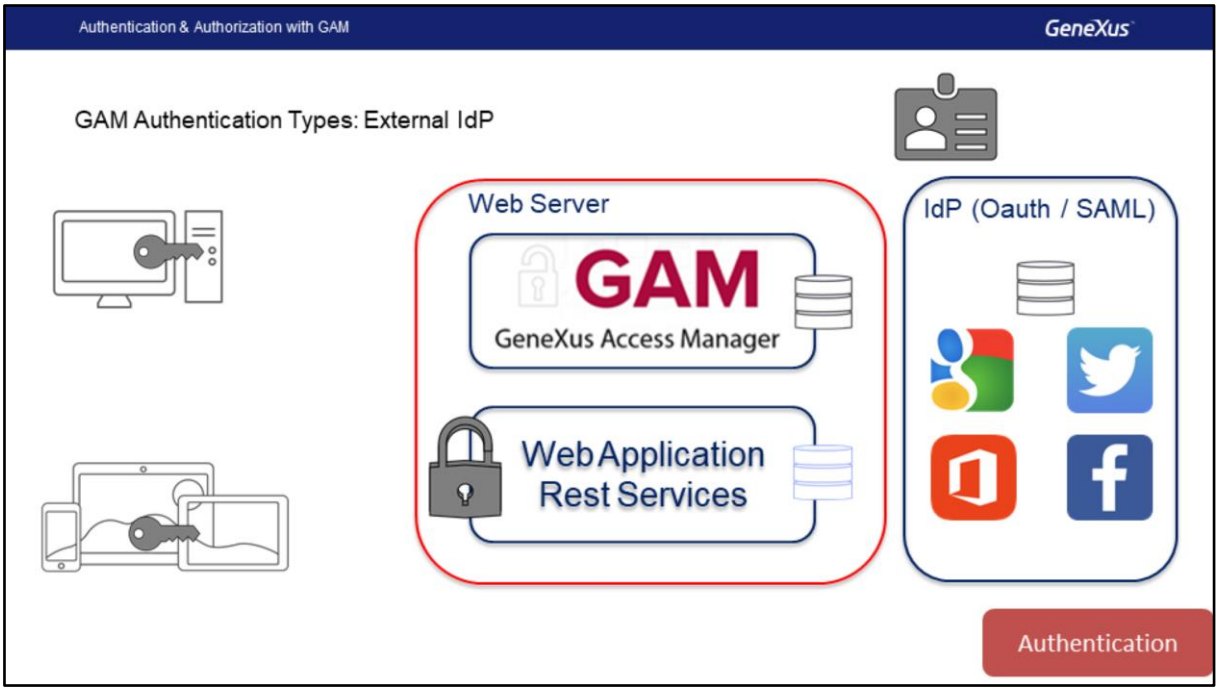
Neste vídeo, veremos um pouco mais sobre os recursos de autenticação e autorização usados no GAM e também veremos uma demonstração em que usaremos o back-end Web fornecido por essa ferramenta.



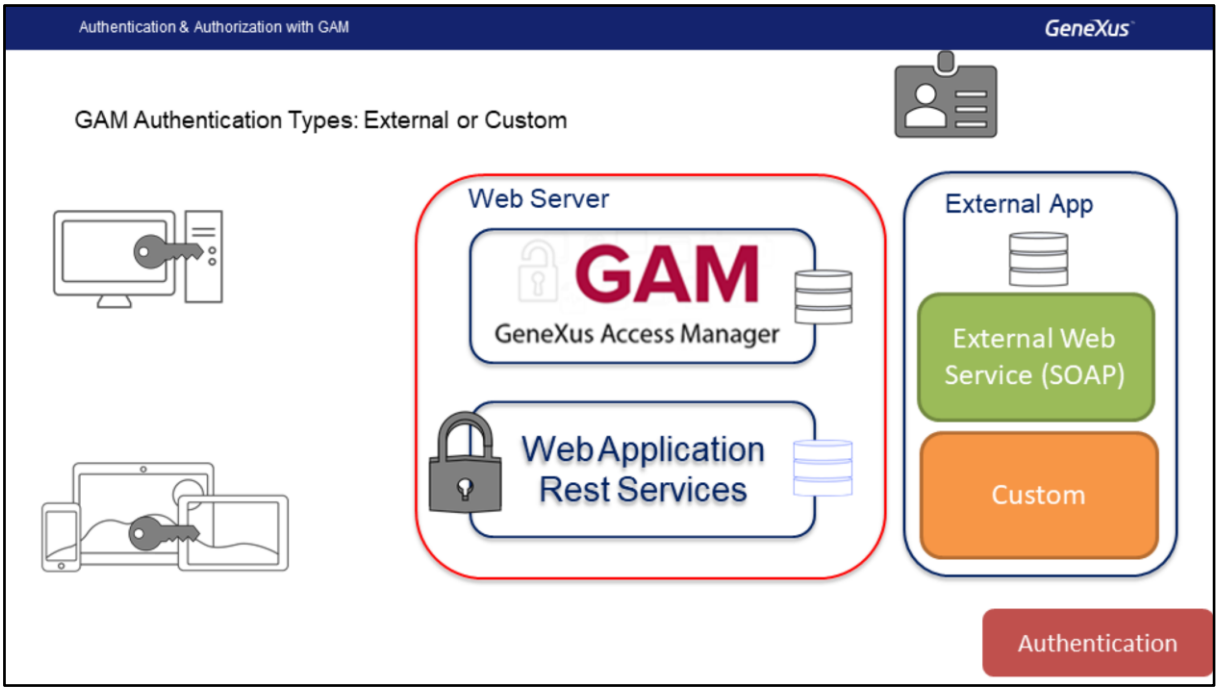
Autenticação é o processo de verificar se um usuário é quem ele diz ser, validando suas credenciais. No caso do GAM, as credenciais são nome de usuário e senha.

Temos diferentes tipos de autenticação para implementar, você pode até mesmo ativar vários ao mesmo tempo.

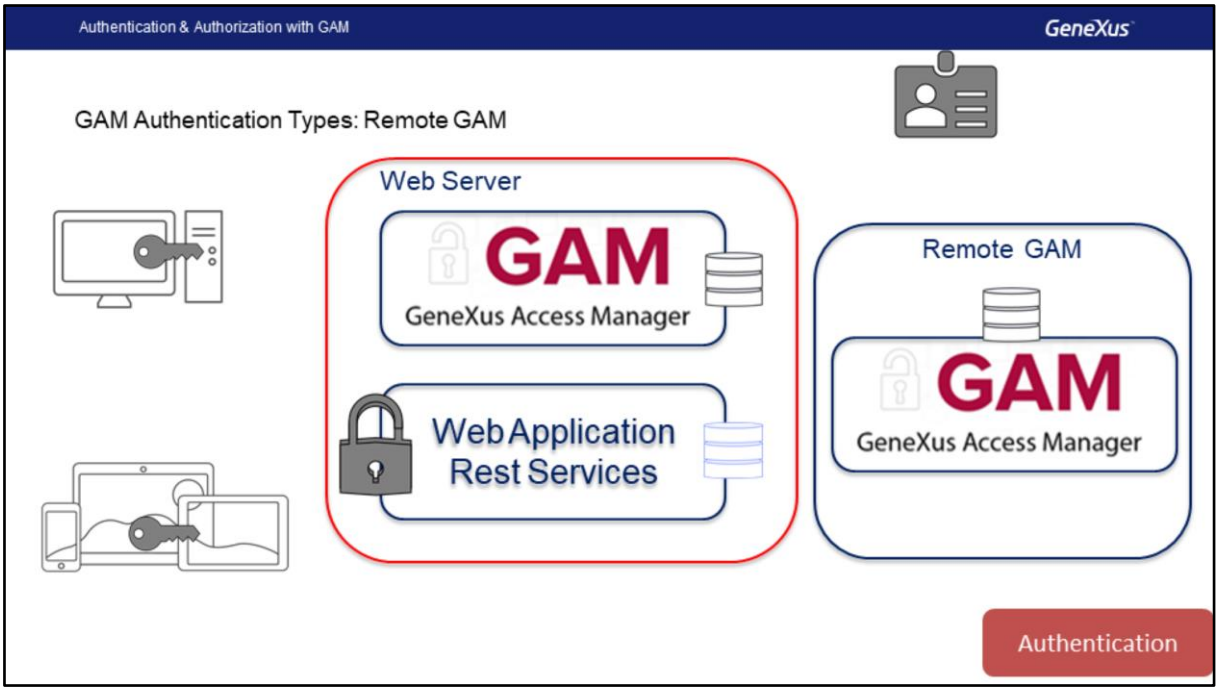
Os tipos são:
Local: neste caso, as credenciais do usuário serão armazenadas no banco de dados do GAM na tabela de usuários, por razões de segurança, a chave não é armazenada, mas sim um Hash obtido ao aplicar um algoritmo SHA-512 na chave inserida pelo usuário. Então, sempre que as credenciais precisarem ser validadas, esse Hash será calculado para a senha digitada pelo usuário e comparada ao Hash armazenado na tabela. Entre outras coisas, isso implica que não temos como recuperar o valor da senha do usuário ou manipulá-la de qualquer forma, já que o processo é irreversível, ou seja, dado um hash, você não pode obter a string que o gerou inicialmente.



Outra opção é usar um provedor de identidade externa que suporta OAuth 2.0 ou SAML 2.0: os IdP Provedores de Identidade que podemos usar são vários, por exemplo, Google, Facebook, Twitter, Office 365, etc., nestes casos, no banco do GAM somente o ID do usuário será armazenado na tabela de usuários, isso é usado, por exemplo, para atribuir um ROL a um usuário e as credenciais do usuário serão gerenciadas pelo IdP.
No momento da autenticação do usuário, o usuário será redirecionado para o IdP, onde o usuário irá inserir suas credenciais, caso estiver correto, o IdP retorna ao site novamente. Diferenças entre Oauth e SAML tem a ver com as tecnologias utilizadas e o fluxo, mas as alternativas são semelhantes no sentido de que as credenciais são inseridas no IdP e este, depois de verifica-las, retorna o controle para o sistema que solicitou a validação. Em todos esses casos, é necessário ter URLs públicas para obter os redirecionamentos necessários.



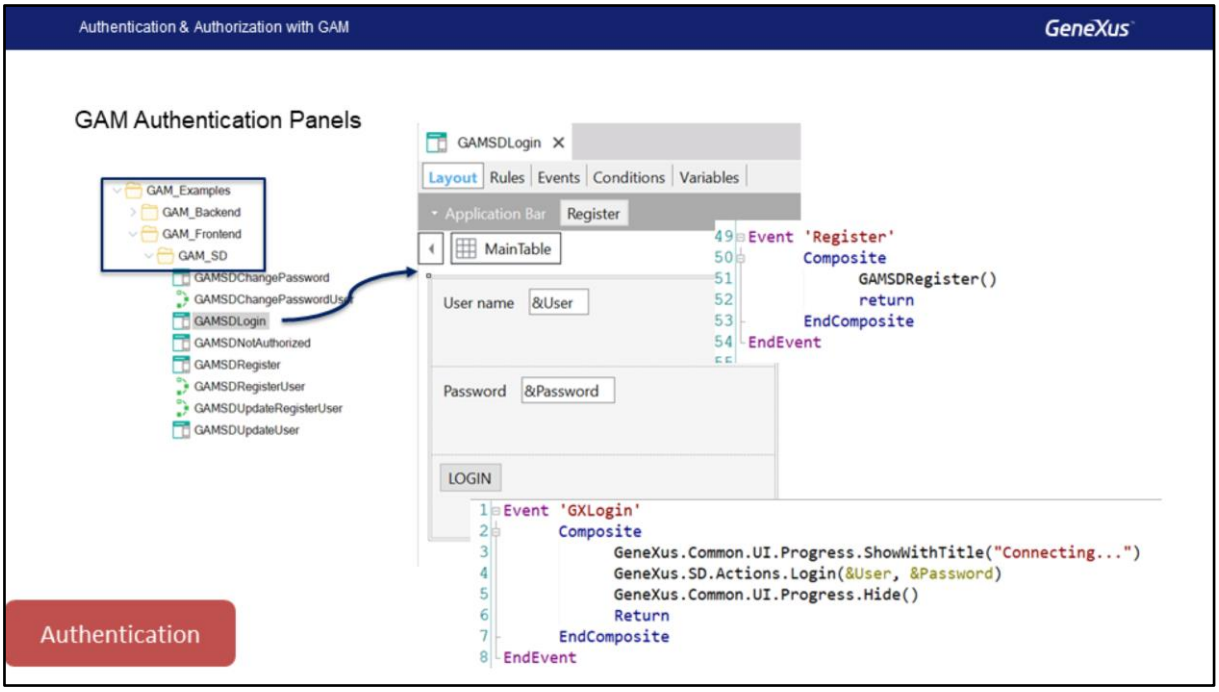
Se usarmos Autenticação Externa, nesse caso, o GAM será configurado para interagir com um provedor externo que possa usar o Web Services ou outro mecanismo personalizado. Neste caso, como no anterior, no GAM somente informações mínimas do usuário são armazenadas, uma vez que a validação das credenciais de acesso é realizada em outro sistema.
Nesses casos, o GAM nos fornece recursos para mapear as Roles definidas no GAM e enviar para Roles externas.



Também podemos usar o GAM Remoto, pois o próprio GAM é um provedor de identidade, que manipulará as credenciais do usuário, dessa forma podemos definir que um aplicativo que use o GAM valide as credenciais do usuário em outra instância do GAM que fará a função IdP

Para mais informações sobre os tipos de autenticação, você pode usar o Wiki, onde você encontrará muitas informações, casos de uso e exemplos detalhados.

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

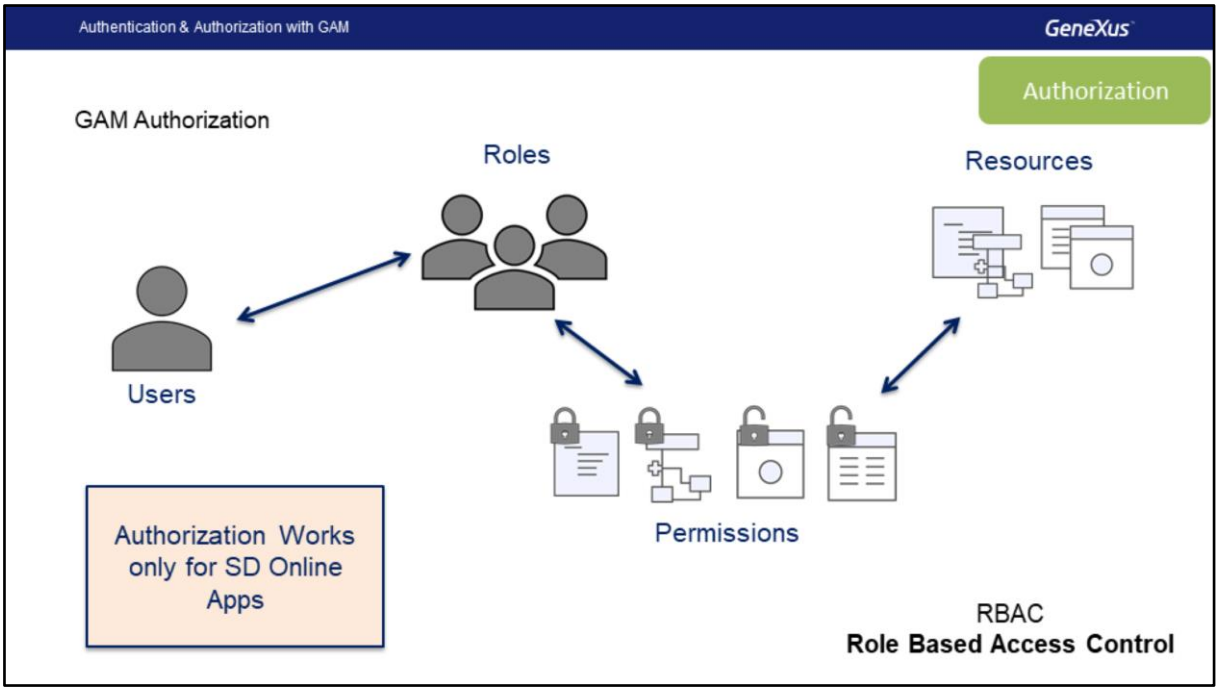


Como parte do mecanismo de autenticação do GAM, forneceremos dois objetos, um Web Panel e um Panel for Smart Devices já programados; esses painéis podem ser personalizados, se desejarmos.

Em particular, dentro da pasta Gam_Examples, vamos encontrar a pasta Gam_FrontEnd e dentro desta, uma pasta chamada GAM_SD com objetos para fazer o Login, a mudança de senha, o registro de novos usuários ou a atualização dos dados do usuário.

Por exemplo, este é o panel de Login, GAMSDLogin, que implementa eventos para realizar o login e registro de novos usuários.

Um aspecto importante deste panel é que, quando o aplicativo SD é off-line, esse painel deve ser executado on-line, ou seja, o usuário deve ter uma conexão com o servidor para acessá-lo.



Com o GAM, também podemos fazer a autorização, que é o processo de verificar se um usuário que já foi autenticado tem as permissões necessárias para executar alguma ação no sistema.

Para isso, GAM tem um esquema baseado em Roles de Usuário, cada usuário GAM tem associado uma ou mais Roles, temos também os Recursos assegurados e a atribuição de permissões sobre esses Recursos nas Roles.

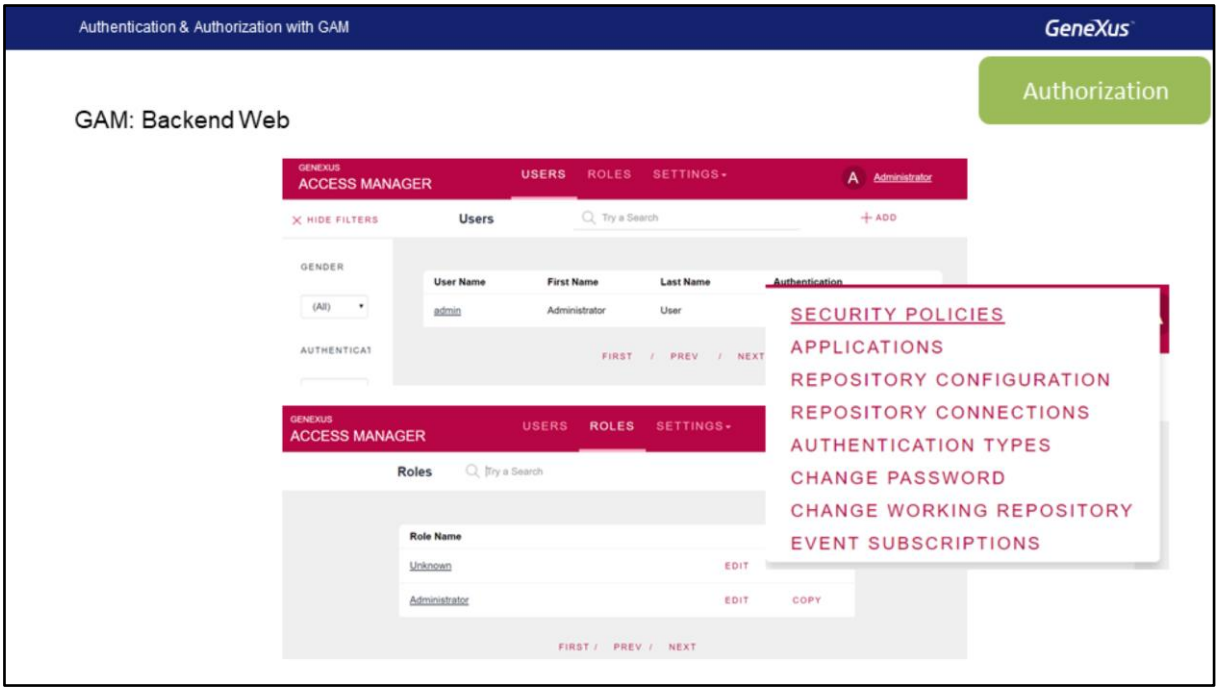
Os recursos que podemos assegurar são:

- Web Panels
- Web Components com o Acesso por URL habilitado
- Processos com protocolo HTTP, por exemplo, relatórios com saída PDF
- Transações WEB, neste caso, podemos também, além de executar, personalizar o modo Insert, Update, Delete ou dar acesso total a uma transação, isto é, a execução e todos os modos.

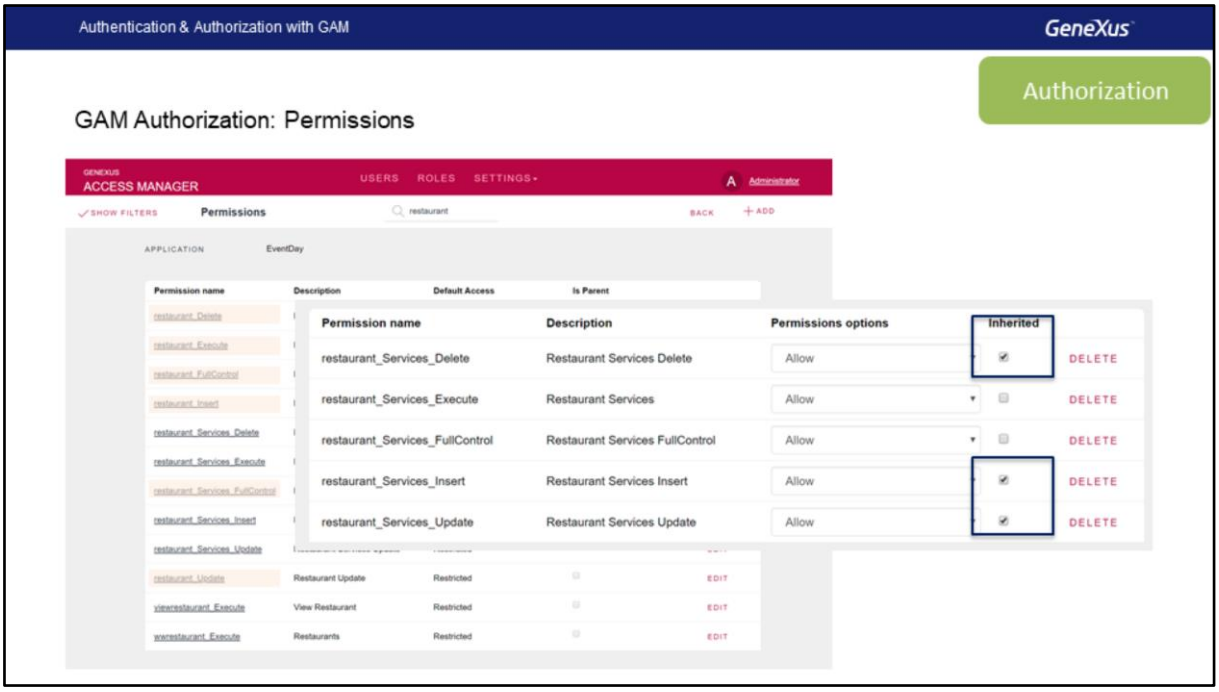
No caso de aplicativos ONLINE para Smart Devices, os recursos a serem protegidos são:

- Panels para Smart Devices
- WorkWith for Smart Devices, neste caso, a permissão é para execução e para as ações da transação são tratadas na transação exposta como business component.
- Processos ou Data Providers com Protocolo Rest

No caso de aplicações para Smart Devices terá apenas a autenticação off-line, uma vez que, por ser um aplicativo offline não pode manter permissões, porque se modificarmos, alguns dispositivos podem não ser sincronizados e dessa forma, o esquema é impraticável. Como mencionado, o painel de login, GAMSDLogin deve sempre online, ou seja, para acessar o dispositivo, deve estar conectado, para fazer a validação de credenciais, depois que o usuário foi autenticado, daí em diante pode trabalhar de forma desconectada, até a sessão expirar.

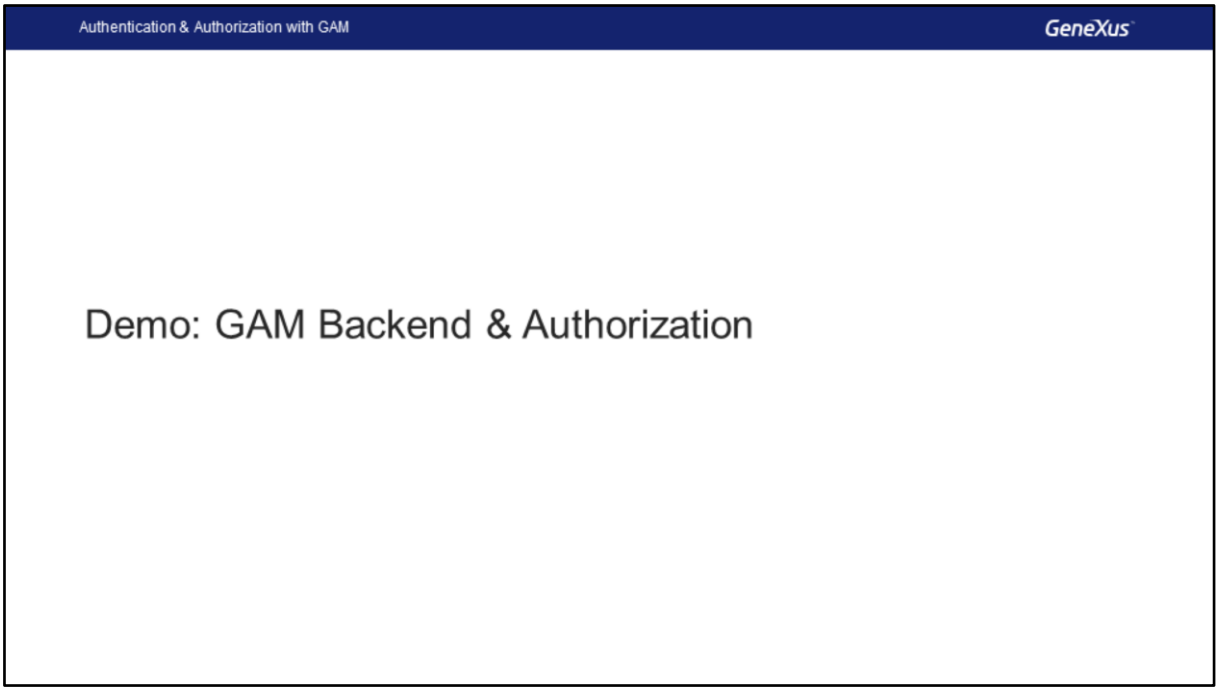


Para lidar com todas essas informações, o GAM também nos fornece um back-end web, que nos permitirá, como administradores de segurança, gerenciar usuários, roles, permissões e outras configurações do aplicativo, como tipos de autenticação e outros parâmetros de configuração.



Uma das facilidades que o GAM nos oferece é que, para cada aplicação, ele será responsável pela geração dos recursos sobre os quais precisaremos dar as permissões.

Na imagem vemos os que foram gerados relacionados a Restaurant.
Ali podemos ver que temos, de um lado as permissões da transação Restaurant, um para cada modo (insert, update e delete), também execução, e outra que indica FullControl.
Depois, também vemos recursos com o nome Restaurant_Services, que se referem à transição quando é usado como BC e exposta como REST, além do prefixo usado no objeto WorkWith para Smart Devices.
Ao autorizar uma função com o FullControl, estamos dando, nessa transação, todas as permissões, execução e cada um dos modos que serão mostrados como herdados.



Vamos ver tudo isso que falamos no GeneXus.


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