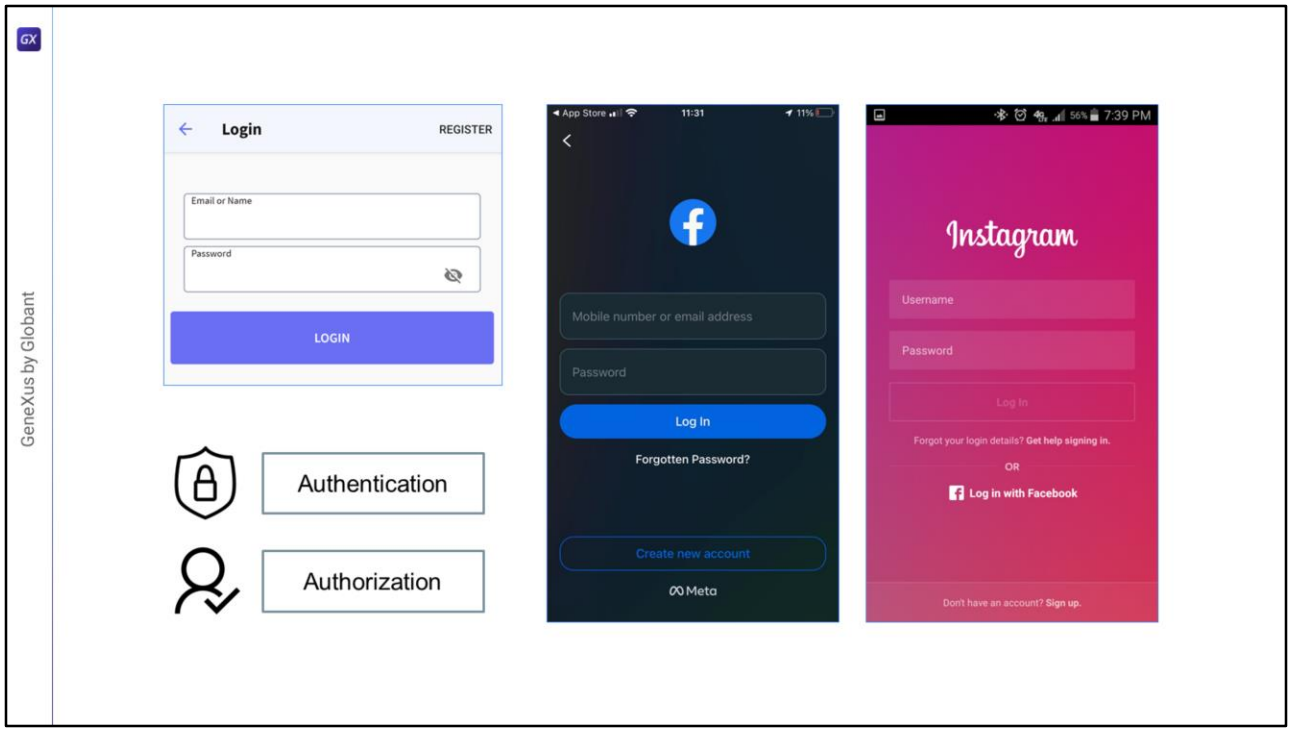
GeneXus by Globant

GeneXus Access Manager

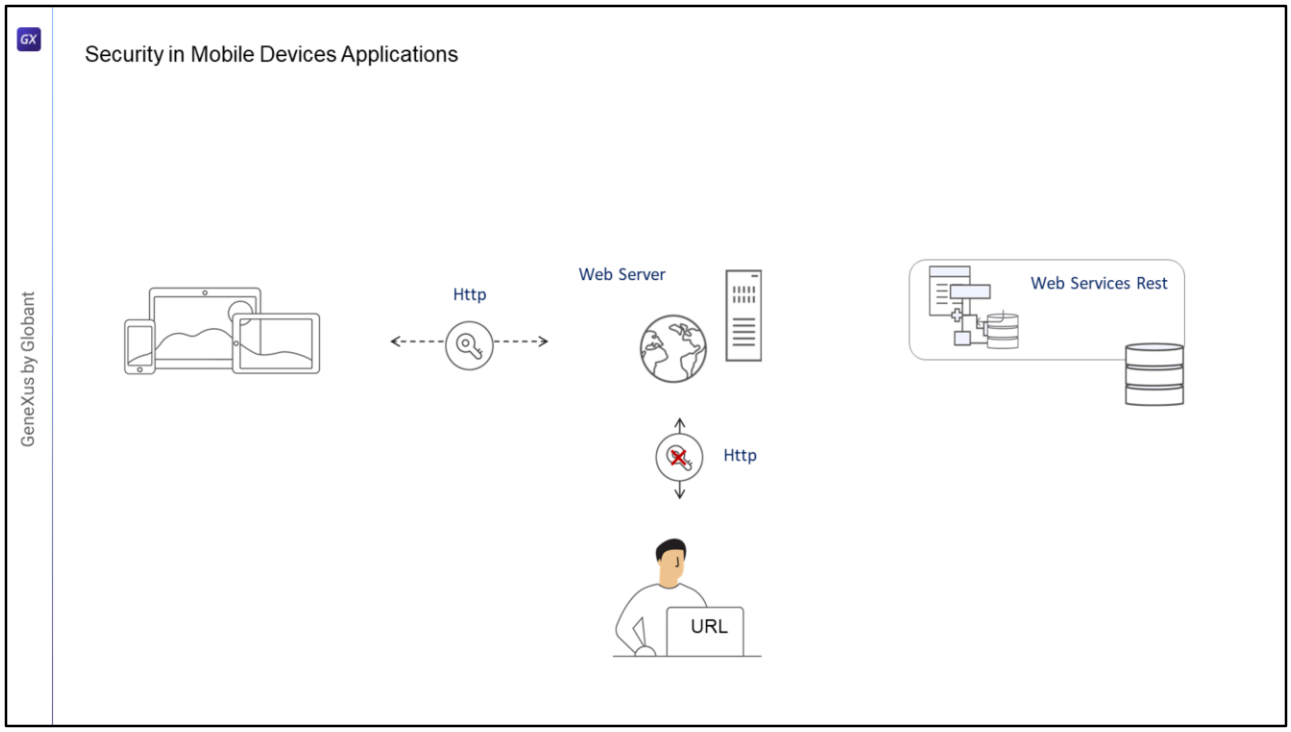
Introduction

Diego Marranghello



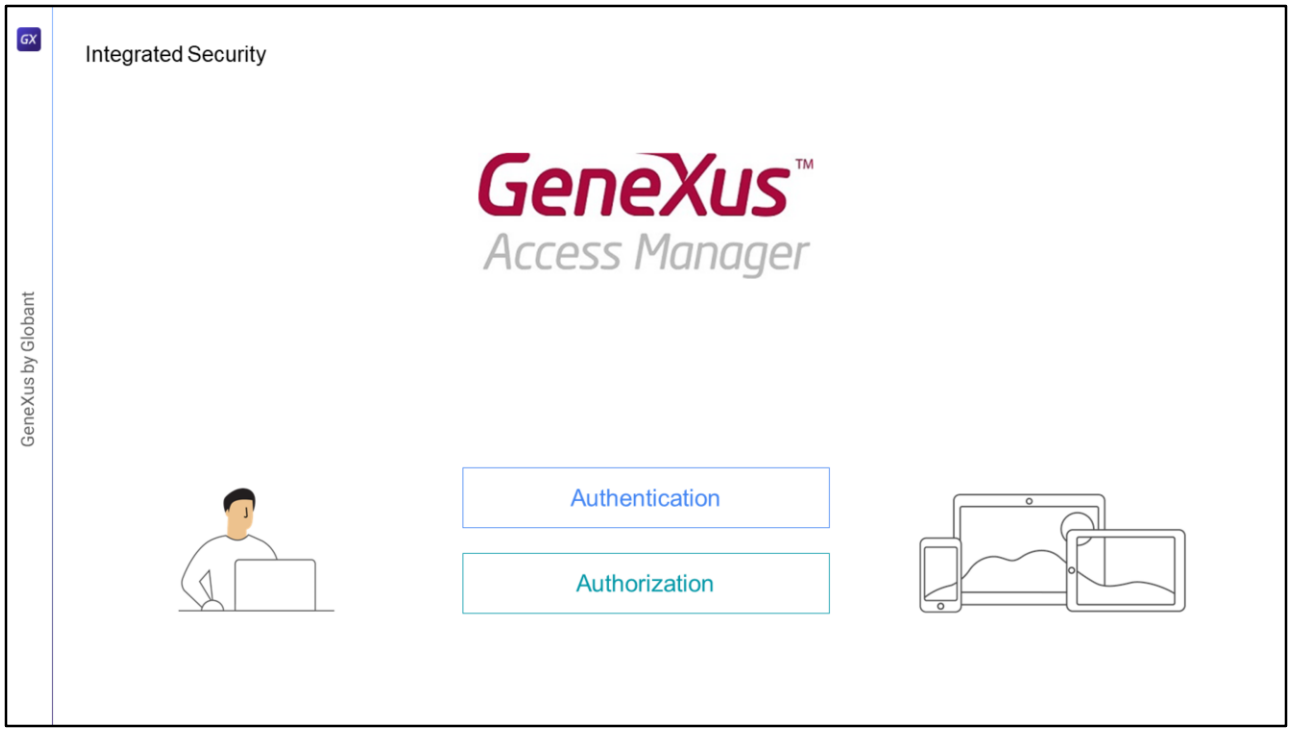
Como já sabemos, a grande maioria das aplicações modernas necessita de um esquema de segurança, para que apenas os usuários permitidos possam entrar e também autorizar ou restringir o acesso a partes da aplicação, dependendo das permissões atribuídas ao usuário.

Isso significa garantir que todos os usuários que entram estejam devidamente autorizados e autenticados.



No caso das aplicações para dispositivos móveis, por serem aplicações distribuídas, uma parte delas é executada no próprio dispositivo, e a camada de negócios da aplicação é resolvida através de serviços Rest que possuem uma URL de acesso, portanto que estão expostos a acessos indevidos.

Como para as aplicações web, o que se faz é confirmar que apenas usuários devidamente autenticados e autorizados possam acessar à aplicação, evitando a execução de usuários que não cumpram isso.

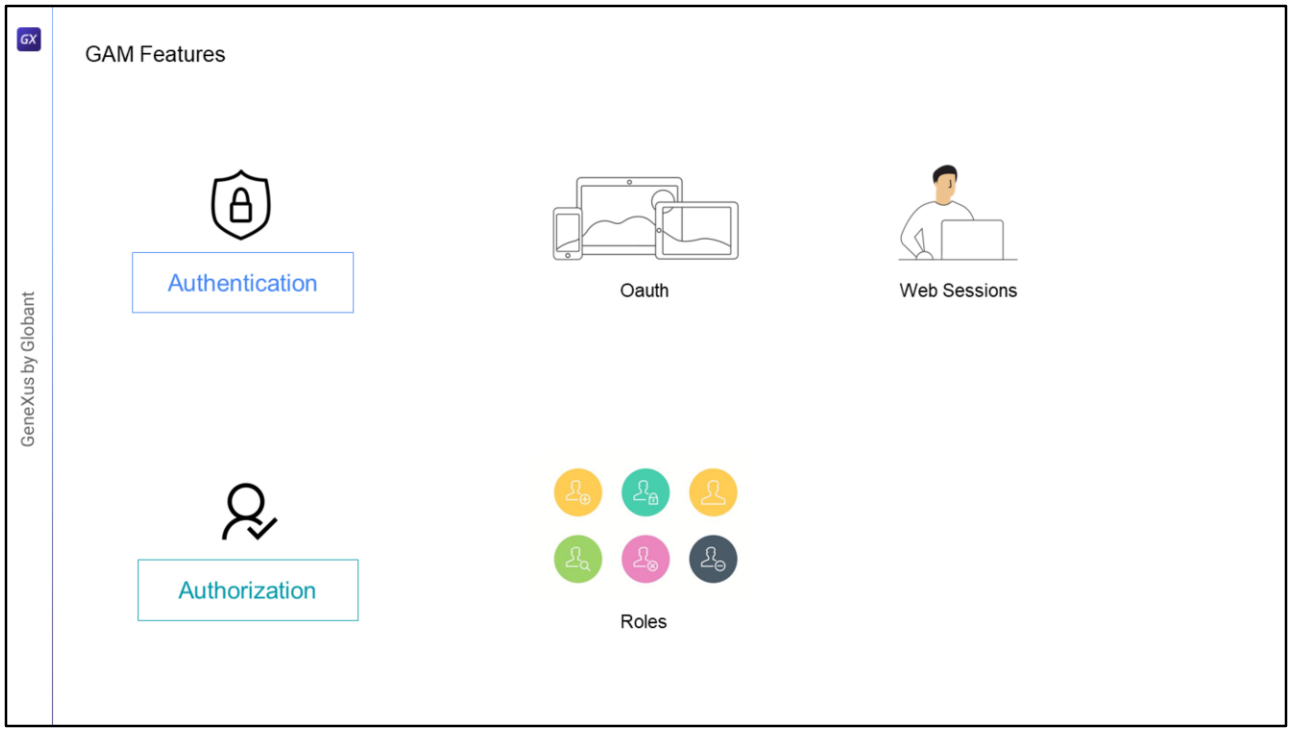


O módulo de segurança GeneXus Access Manager (GAM) resolve as funcionalidades de autenticação e autorização, tanto para aplicações Web como para aplicações para dispositivos móveis.

O GAM está desenvolvido em GeneXus pelo que se integra facilmente à KB da aplicação e permite resolver de maneira centralizada tudo relacionado à Segurança dela. O objetivo é que a solução de Segurança seja utilizada da forma mais declarativa possível dentro da aplicação, sem criar complexidade adicional.

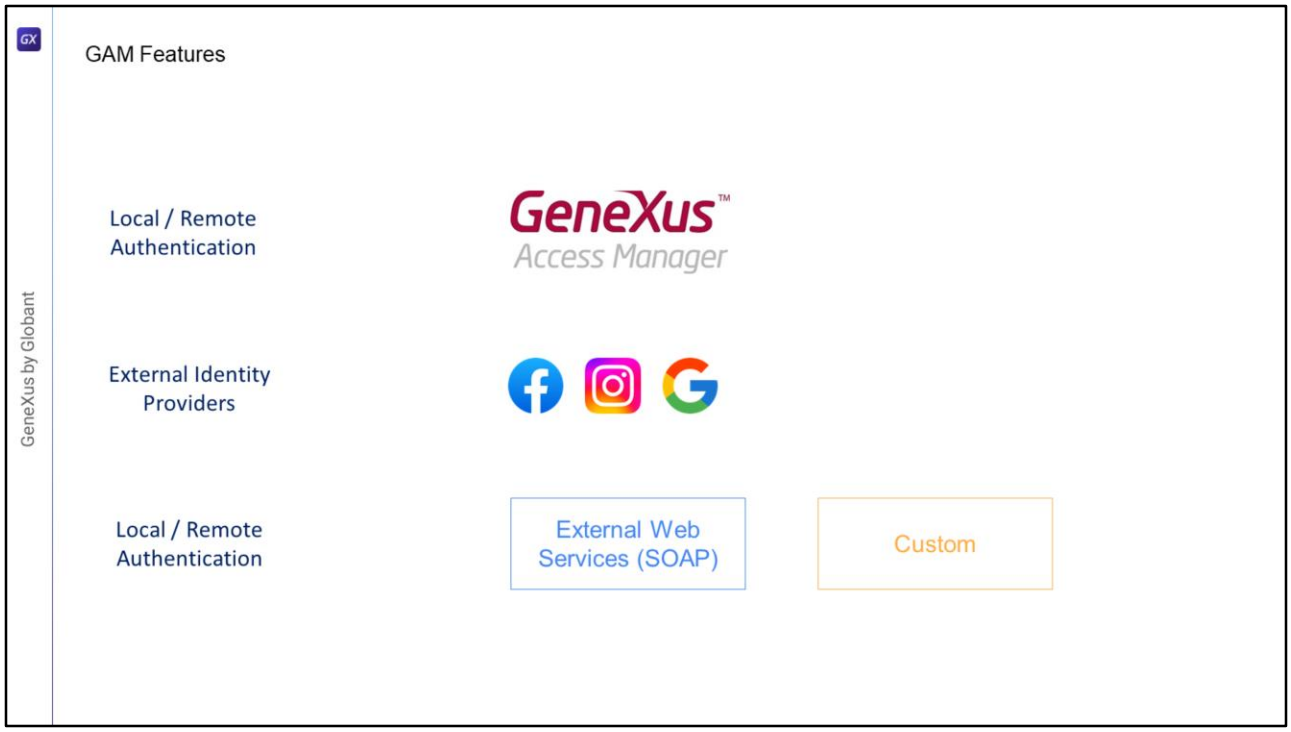
O GAM também disponibiliza um backoffice que permite definir usuários, permissões, políticas de segurança e acesso a objetos, entre outras coisas.

Também fornece uma API para ter acesso a muitas dessas funcionalidades de forma programática.



Para resolver a Autenticação em dispositivos móveis, internamente é usado Oauth, diferentemente das aplicações Web onde são utilizadas Web Sessions.

No caso da Autorização, sua implementação é baseada em Roles



O GAM disponibiliza diferentes Tipos de Autenticação, os tipos disponíveis são:

Autenticação local usando GAM onde os usuários e todas as suas credenciais são armazenados em uma base de dados da qual somos proprietários, ou também de forma Remota, já que uma aplicação que use GAM pode se tornar provedor de identidades e neste caso, outras aplicações com GAM podem se conectar remotamente a este server e obter a autenticação a partir dali.

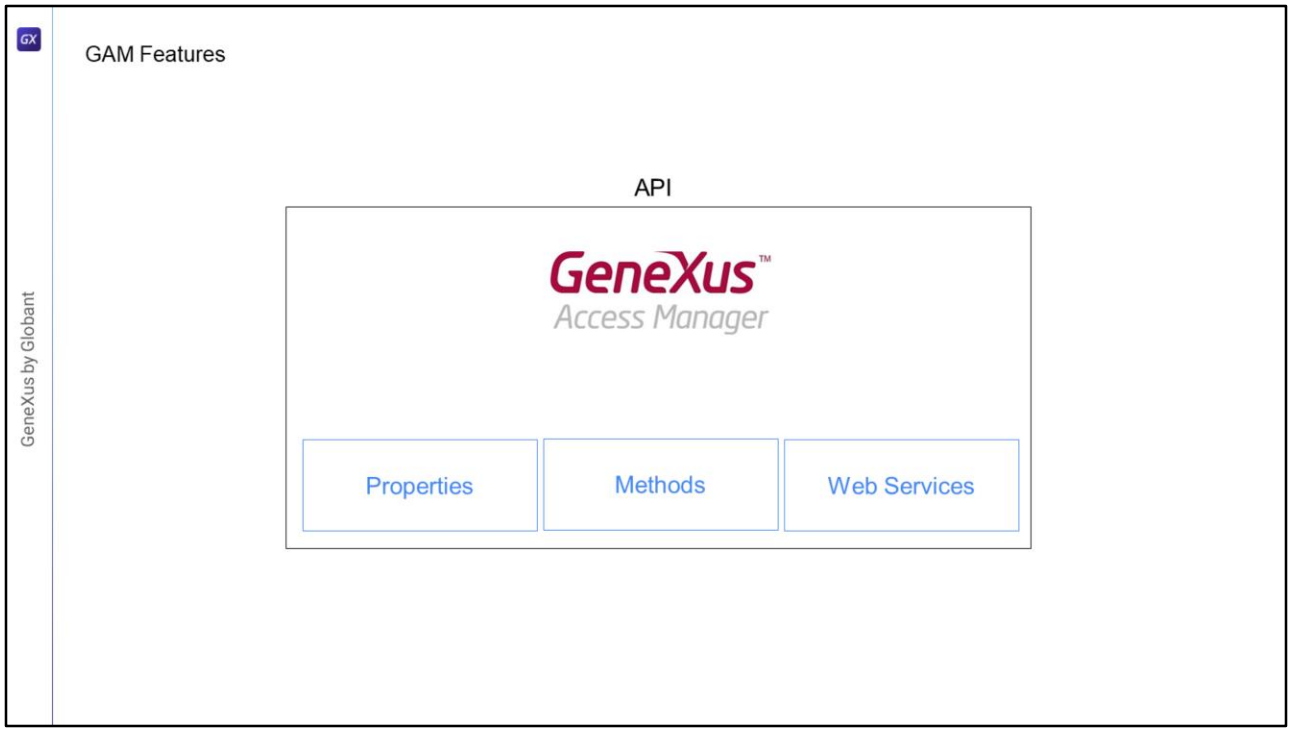
Podemos utilizar também outros provedores de identidade externos, estes fornecem uma autenticação baseada no protocolo Oauth 2.0, como Facebook, Instagram, Google, etc. Neste caso não há necessidade de definir usuários locais.

Às vezes é necessário integrar nossa aplicação com outras, através de uma autenticação externa à aplicação.
Uma forma de autenticação externa é utilizar um web service SOAP fornecido pela outra aplicação e configurar o GAM para que consuma desse web service.

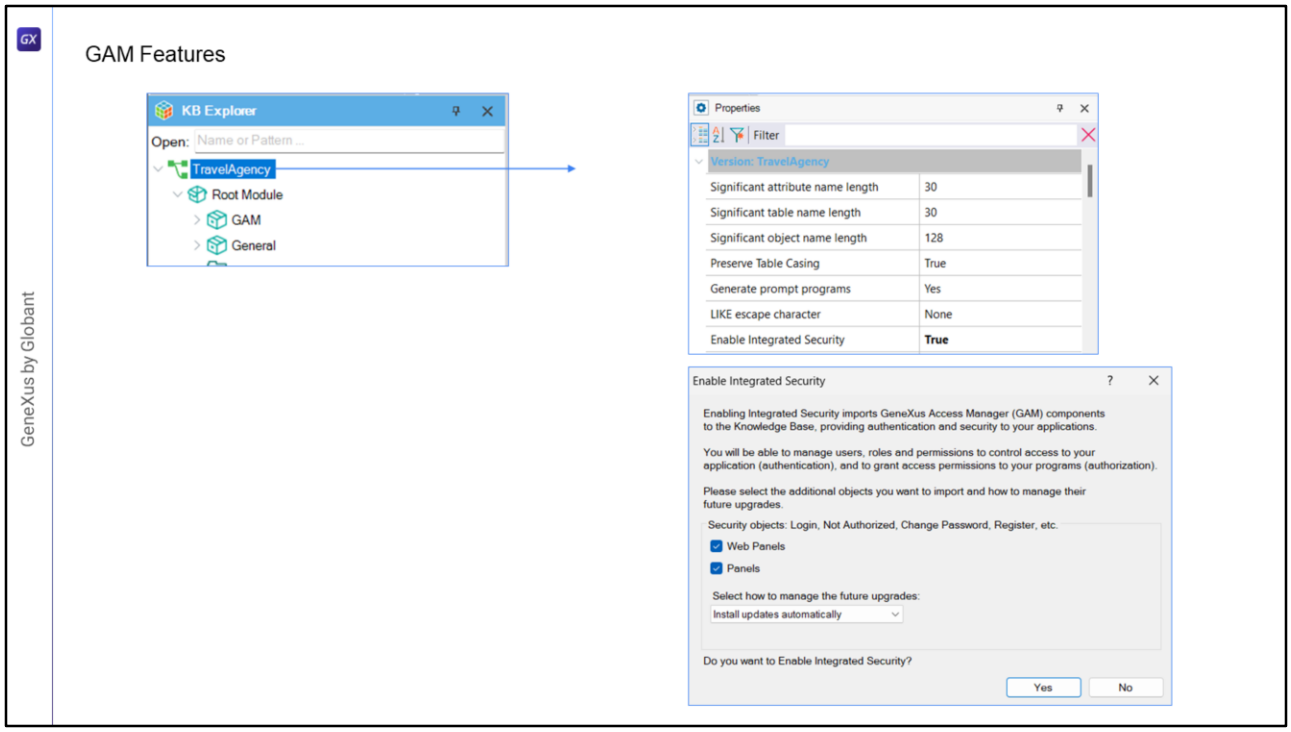
Pode ser que a outra aplicação forneça um programa externo para resolver a autenticação, mas que não seja necessariamente um web service. Neste caso é possível configurar o GAM para aceitar uma autenticação do tipo Custom.

Com a Autorização definimos as permissões e execução dos objetos e dos modos de operação das transações.

A definição é feita concedendo para cada objeto, permissões para cada role e dependendo de qual seja a role atribuída ao usuário, serão as permissões efetivas sobre o objeto como, por exemplo, em dispositivos móveis: os objetos WorkWith e Panel

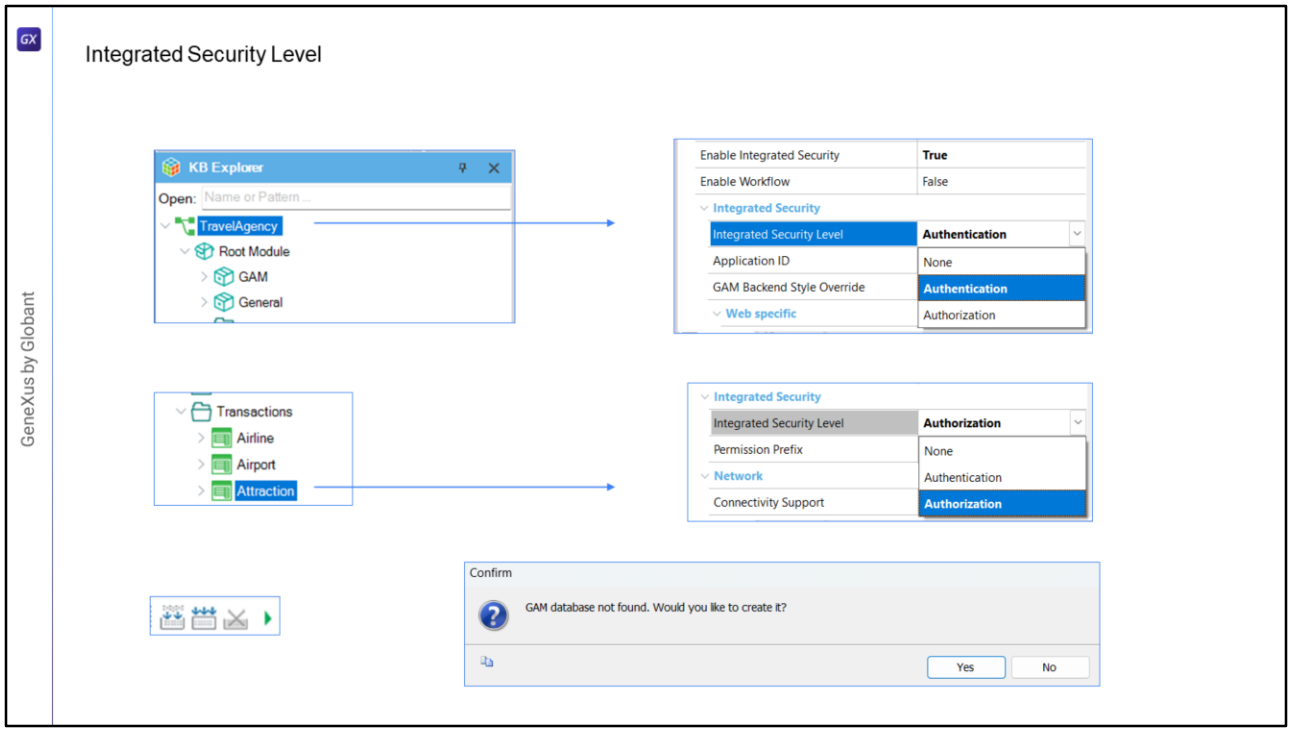


O GAM também expõe uma API para acessar suas propriedades e métodos caso seja necessário fazê-lo a partir de nossa aplicação e uma série de serviços Web que podem ser utilizados a partir de outras aplicações. Não entraremos em detalhes sobre essa funcionalidade neste vídeo.



Para habilitar o GAM deve-se ir até a versão ativa da KB e configurar a propriedade Enable Integrated Security com o valor True.

Ao fazer isso, será aberta uma caixa de diálogo que nos avisa que será instalado o módulo GAM em nossa KB, com a solução pronta tanto para web quanto para dispositivos móveis.



Uma vez habilitado GAM, veremos outra propriedade chamada Integrated Security Level que permite indicar o valor padrão de segurança dos objetos da KB.

Esta propriedade também se encontra no nível de cada objeto, pelo que será possível personalizar a segurança de cada objeto.

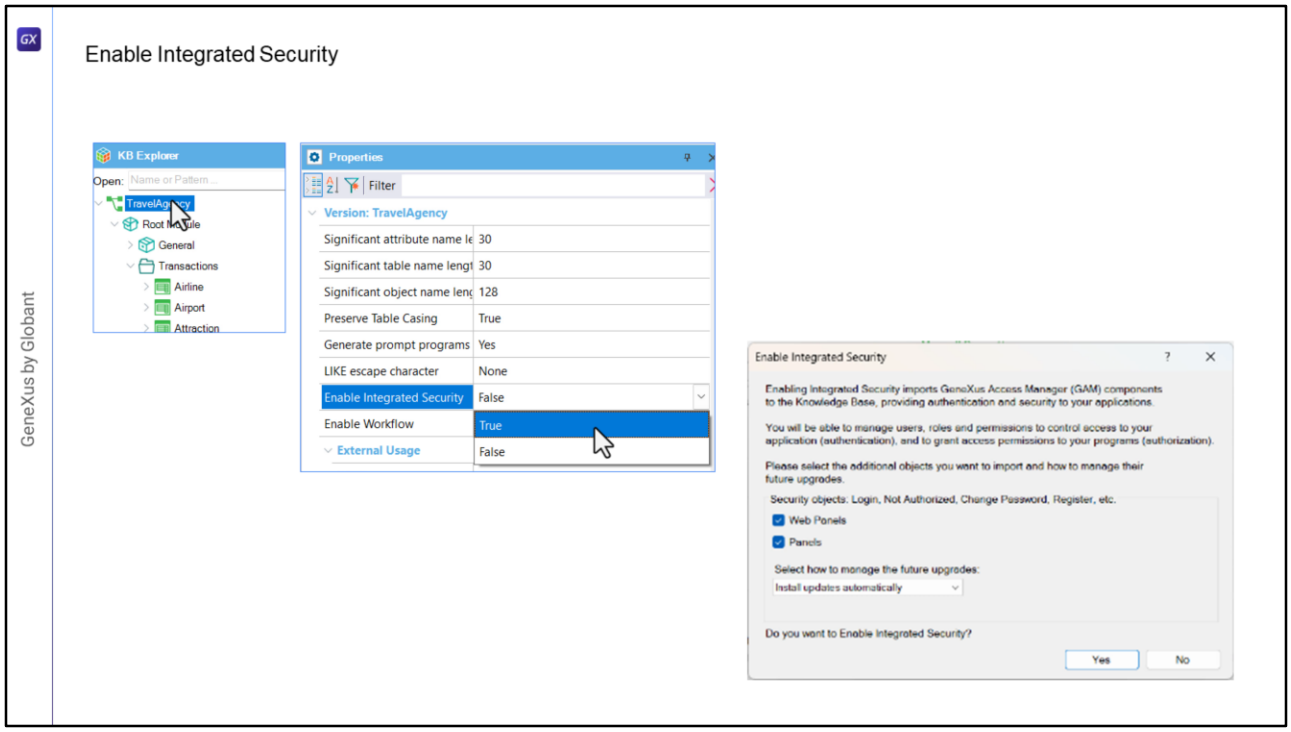
Existem três valores possíveis:

None: indica que o objeto será público, ou seja, não terá segurança.
Authentication: indica que apenas usuários autenticados poderão executá-lo.
Authorization: indica que o usuário, além de ter se autenticado, deverá estar autorizado a executar o referido objeto, ou seja, ter a role adequada para executá-lo.

Uma vez que tenhamos estas propriedades de segurança configuradas, serão importados de forma automática os objetos do GAM para a KB e então teremos que fazer um Rebuild All dela.

GAM solicitará a criação de uma base de dados independente da base de dados da aplicação que estará associada a um Data Store Independente na KB, com o qual toda a sua configuração é independente.

Vamos ver tudo isso em um exemplo.

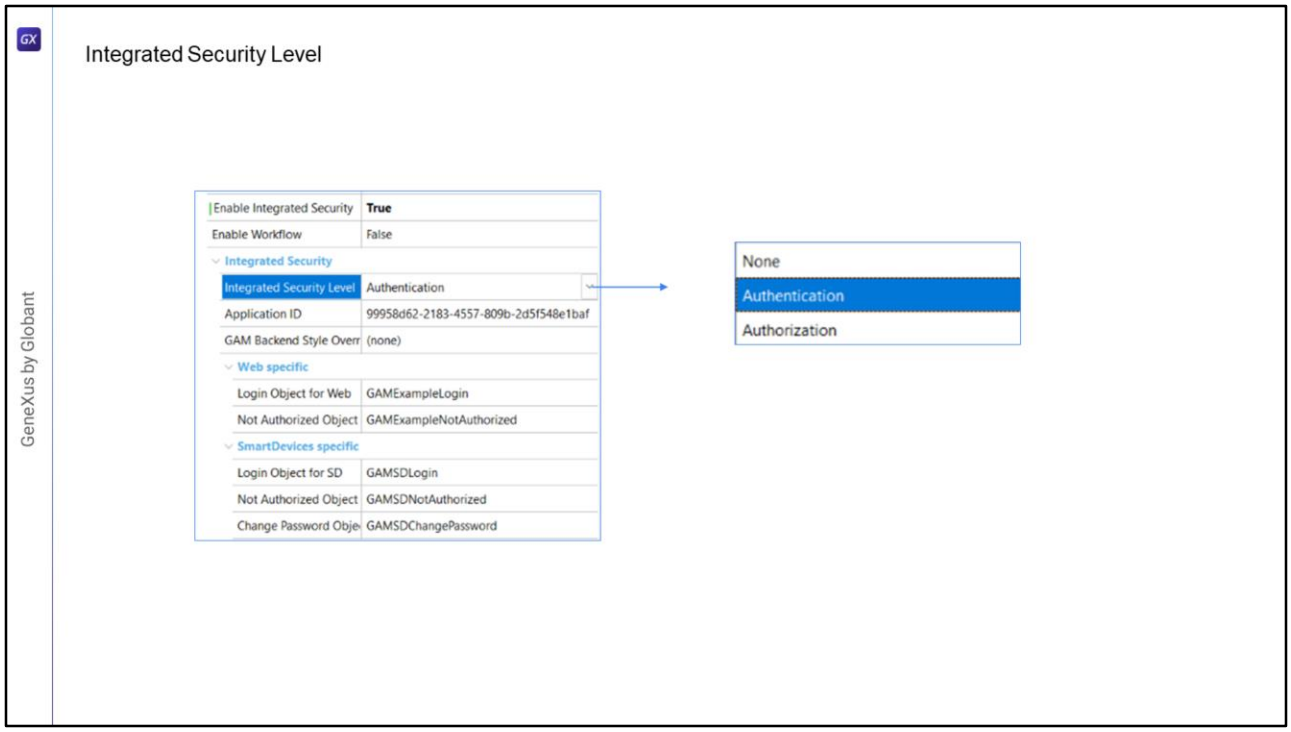


Temos criada parte de uma aplicação para uma agência de viagens. Vamos até as propriedades da Base de Conhecimento, clicamos sobre TravelAgency, o nome da KB, e vamos habilitar GAM colocando na propriedade Enable Integrated Security o valor True.

Aqui podemos indicar se queremos que seja integrado nos objetos utilizados para aplicações Web, e/ou no caso de termos algum objeto gerado para dispositivo móvel, como o caso dos Panels ou o WorkWith usado para mobile, como neste caso, podemos indicar que seja integrado a este tipo de aplicações, neste caso como temos interesse em aplicá-lo a uma aplicação mobile deverá estar selecionado.

Com este combo podemos escolher como queremos que seja atualizado este módulo, pode ser automático, podemos escolher que nos pergunte ou que nunca seja atualizado.
Confirmamos.

Aqui começam a ser importados os objetos do GAM.



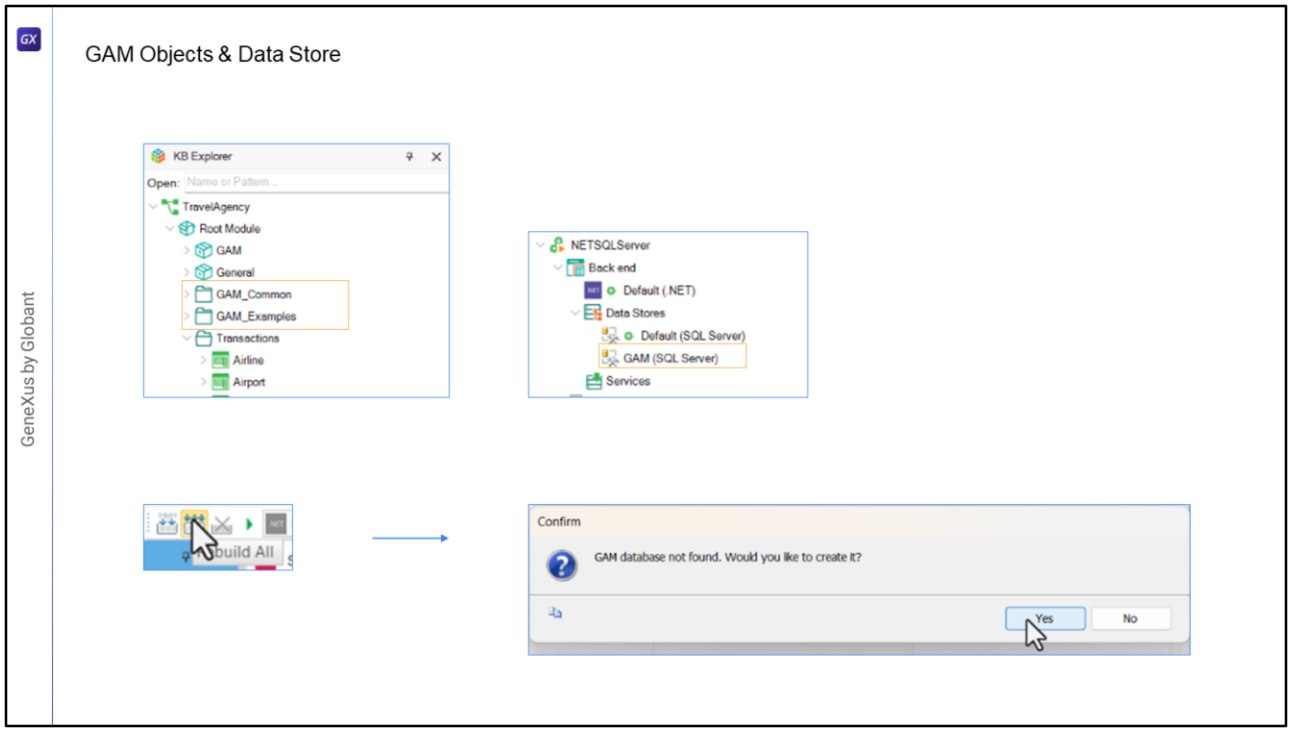
Agora temos disponível a propriedade Integrated Security Level.

Nesta propriedade podemos indicar se queremos habilitar apenas autenticação, que é o valor padrão, se queremos autorização, ou se não queremos ter segurança.

Deixaremos em Authentication, o que significa que para acessar qualquer objeto que permita segurança, nos solicitará credenciais de acesso.

Além disso, é atribuído um Application ID que será utilizado no repositório do GAM para identificar a aplicação.

Agora já temos as propriedades onde indicamos os objetos de login, um em caso de erro de autorização e outro para alterar a senha.

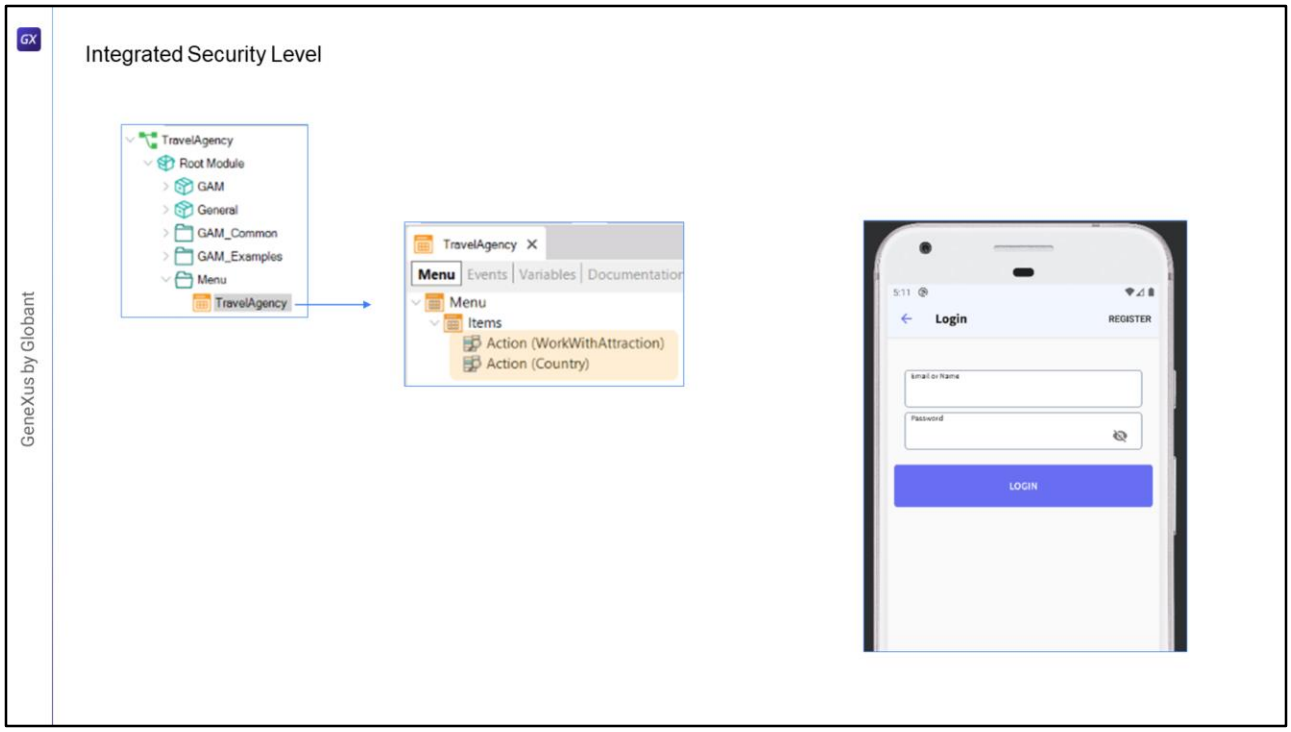


Na KB já podemos ver que foram criadas algumas pastas no root module.

Temos também um novo Data Store, GAM, com a informação dessa conexão.

Bem, ao terminar o processo de importação, vamos fazer um Rebuild All.

GeneXus nos indica que a base de dados do GAM não foi encontrada e se queremos criá-la. Vamos dizer que sim.
É criada a base de dados com todas as tabelas e depois é inicializada.



Em nossa aplicação temos um objeto Menu chamado TravelAgency, que declaramos como startup object, e que possui esses objetos como itens para que possamos acessar.

Ao executar, a primeira coisa que vemos é a tela de login, pois configuramos que para toda a aplicação tivesse como segurança Authenticaton.
Tudo isso nos fornece o GAM de forma automática.

Para entrar usaremos um usuário que é criado por padrão: “admin”, com a senha “admin123”.
E aí sim nos permite acessar nossa aplicação.

Com isto terminamos a introdução ao objeto GAM para dispositivos móveis.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com