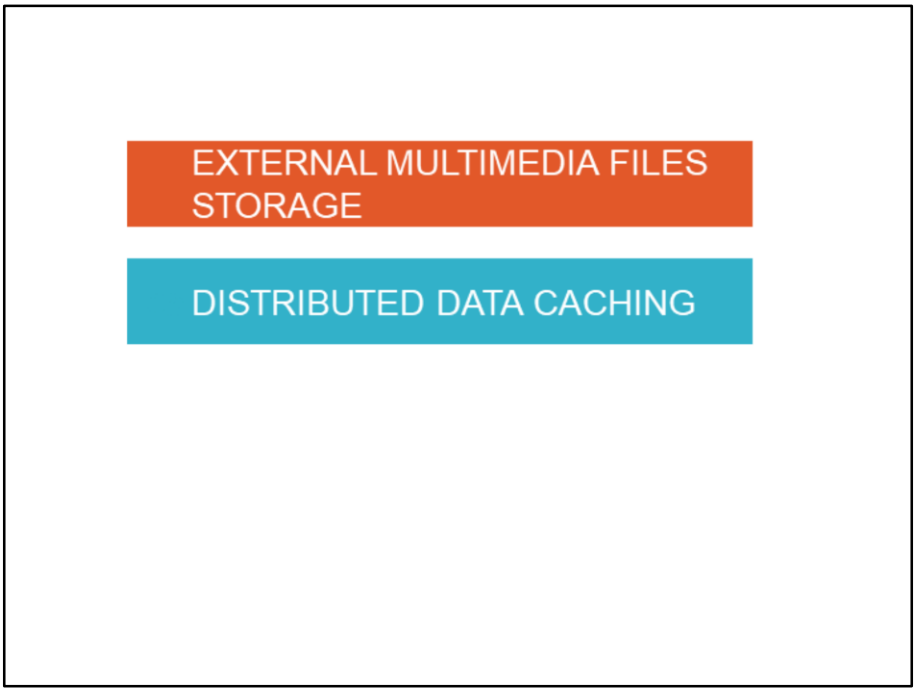
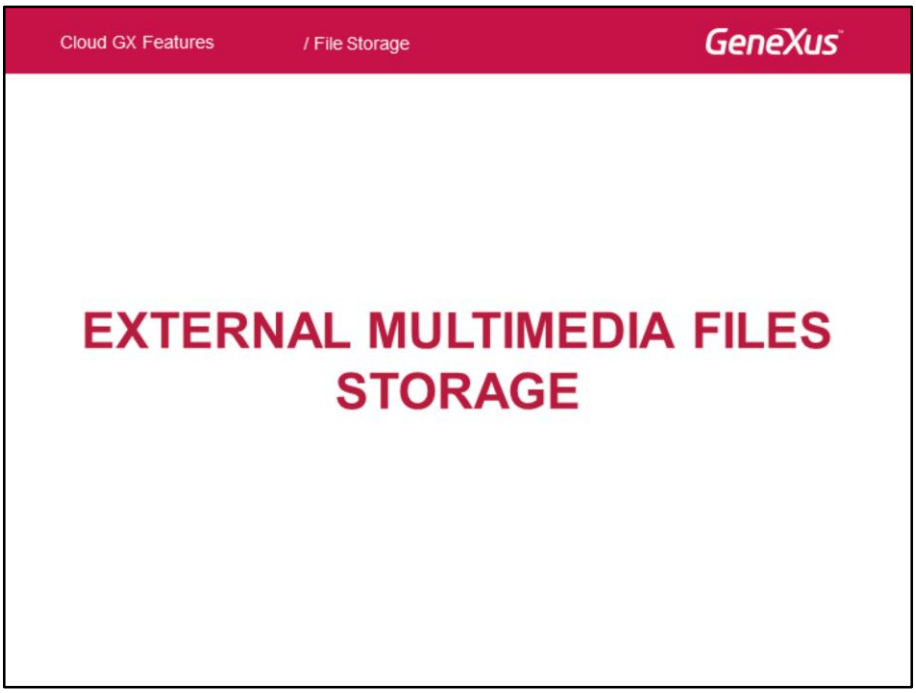
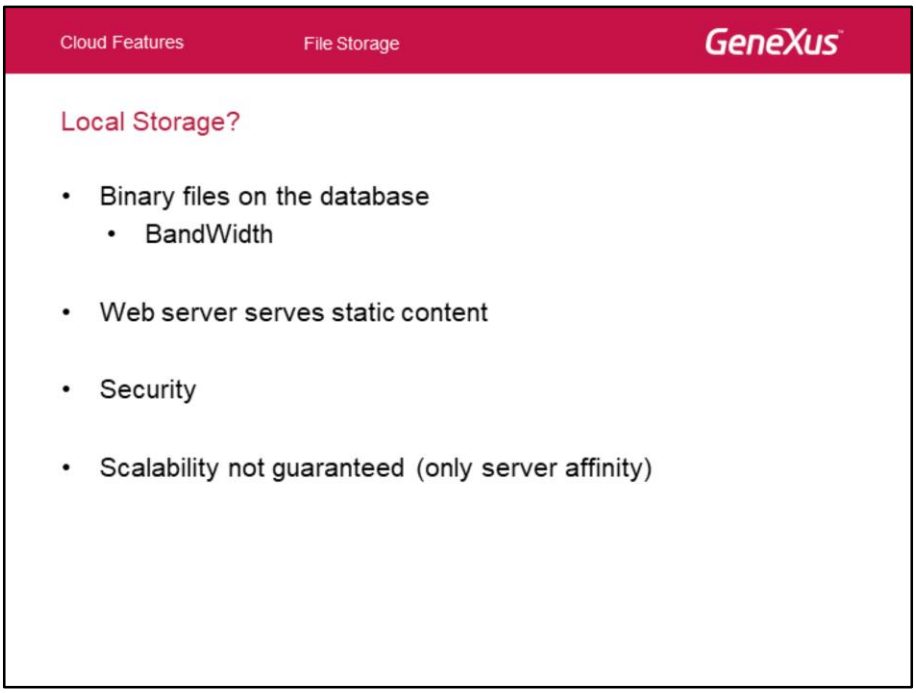
Cloud Features

GeneXus 15


EXTERNAL MULTIMEDIA FILES STORAGE
DISTRIBUTED DATA CACHING



Las empresas están poniendo más y más apps en la nube porque el modelo de la nube mantiene la agilidad del negocio durante el tiempo de vida de la app; entre los beneficios relevantes se incluye alta performance en el acceso a datos, seguridad de los datos y escalabilidad.



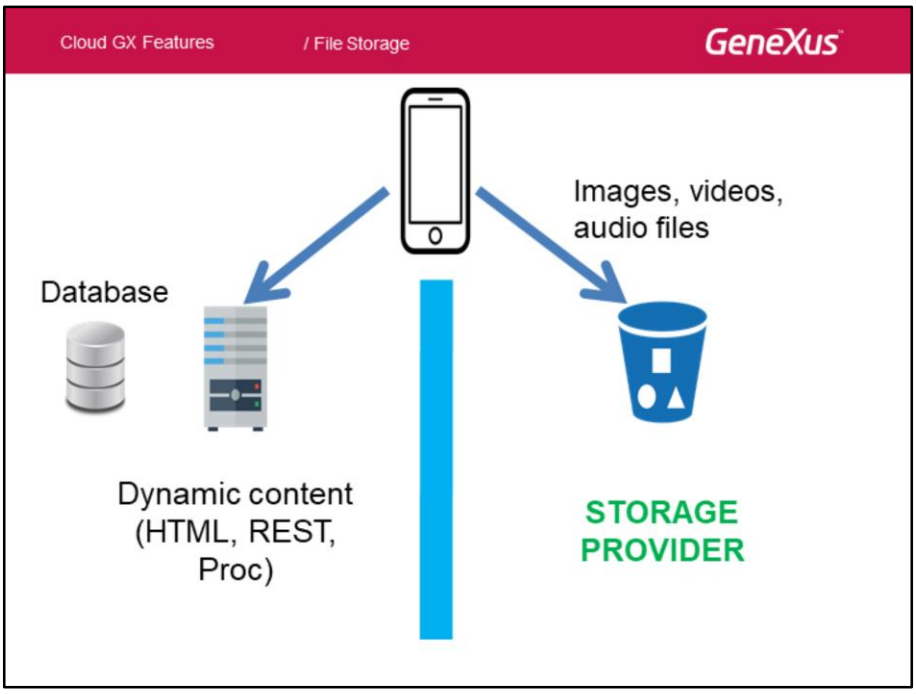
Con respecto al almacenamiento multimedia (para videos, audios o imágenes), existen varias razones convincentes para utilizar el almacenamiento externo en lugar de almacenar datos en la base de datos, que se detallan a continuación:

Escalabilidad
Para garantizar la escalabilidad de una aplicación en clúster, el sistema de archivos del servidor no se puede usar para almacenar el contenido multimedia, porque no se puede garantizar que el mismo servidor sirva al cliente en la siguiente solicitud HTTP.

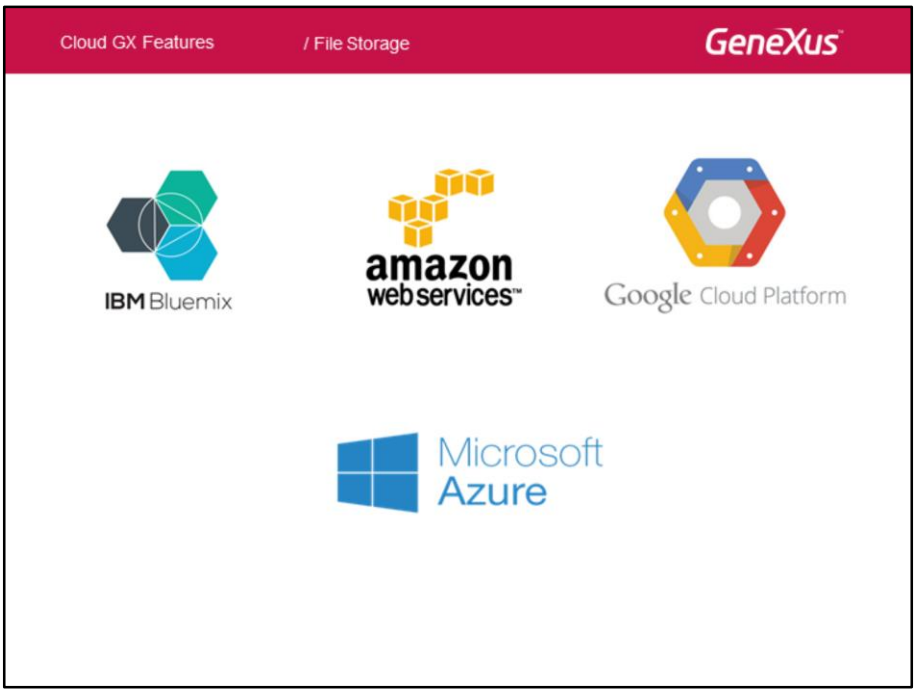
Seguridad

Los datos deben protegerse contra el acceso de usuarios no autorizados.

Performance
El almacenamiento temporal de datos en el sistema de archivos del servidor implica una penalización de E / S. Esto se puede evitar usando el mecanismo de almacenamiento externo.



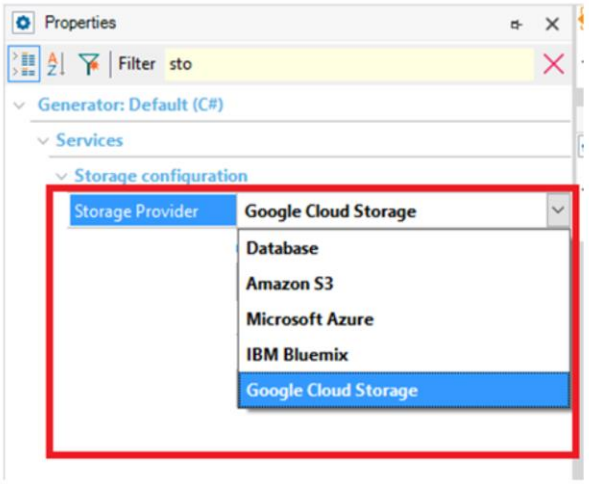
Cuando se utiliza un sistema de almacenamiento externo, los archivos multimedia no se almacenan en la base de datos y la URL de ubicación del archivo de almacenamiento se guarda en la columna "_GXI" de la tabla.



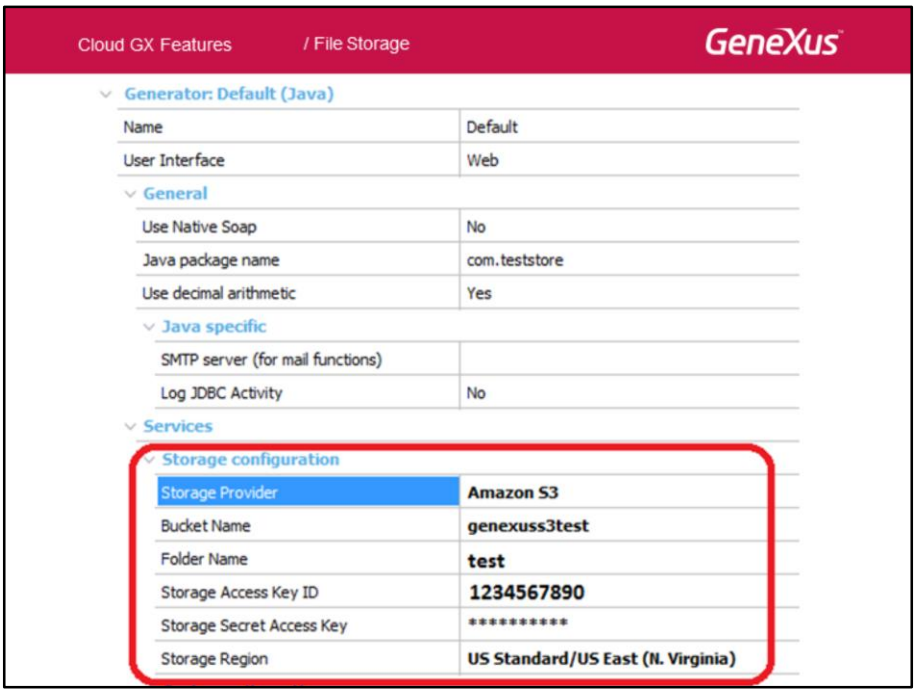
Los proveedores de almacenamiento soportados son:
Amazon S3
Bluemix
Windows Azure
Google Cloud Platform

Cada inserción o actualización desde cualquier fuente (pueden ser procedimientos, business components o transacciones) administra automáticamente los datos en el almacenamiento externo. Además, el almacenamiento externo se recupera automáticamente. El usuario no necesita programar nada o cambiar su lógica.

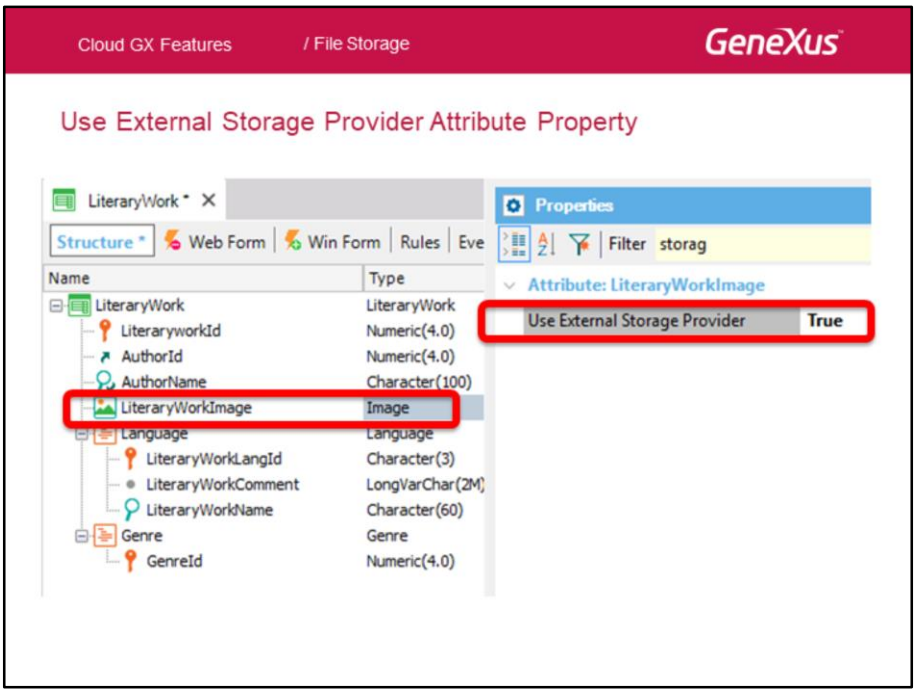
Storage Provider Property



La propiedad Storage Provider determina el proveedor de almacenamiento para los archivos multimedia utilizados en la base de conocimiento. A través de la propiedad puede seleccionar entre diferentes proveedores de almacenamiento externo.



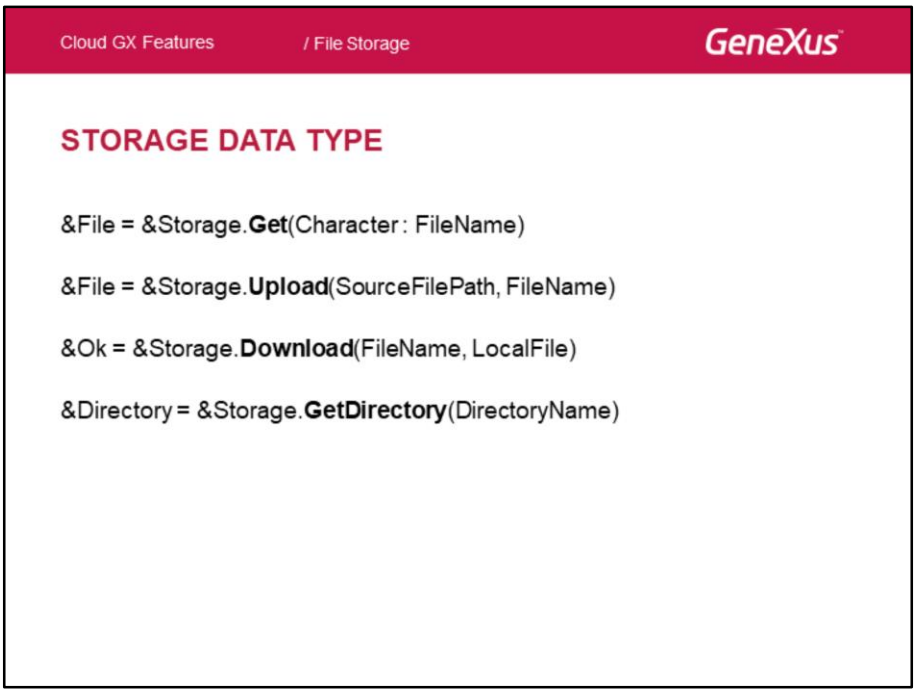
Dependiendo del proveedor de almacenamiento utilizado se disponibilizan diferentes propiedades adicionales para configurar la ubicación del almacenamiento. Estas propiedades se agrupan en la sección Services. Es propiedad tanto de generador y como de objeto.



A nivel de objeto podemos configurar la propiedad Use External Storage Provider (True, False) para cada atributo multimedia. Su propósito principal es determinar si un campo multimedia usará la misma propiedad Storage Provider configurada a nivel del generador. Si se selecciona False, será asumido el valor de Base de datos.

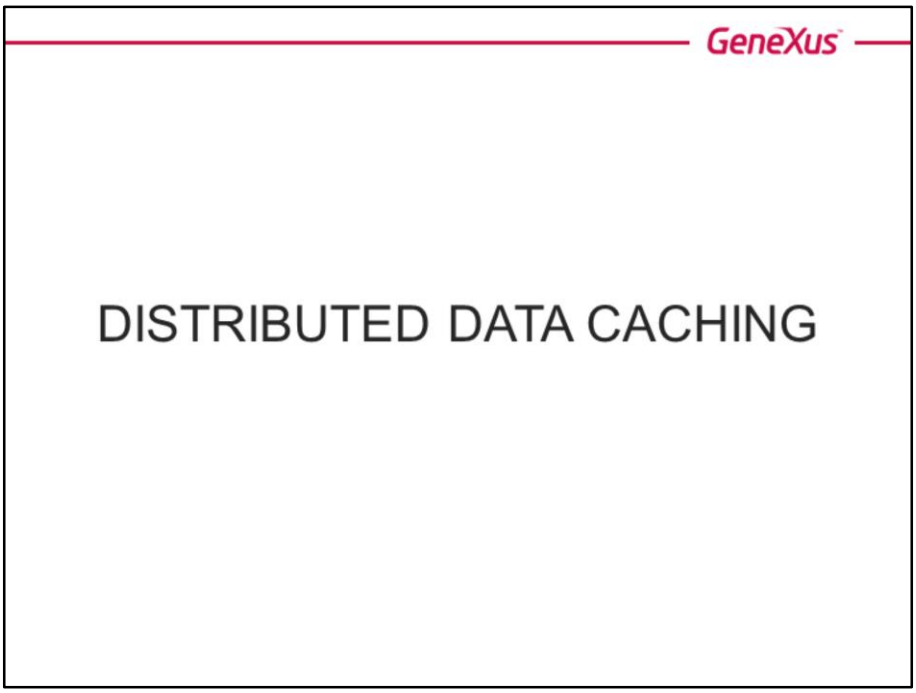
Más información:
http://wiki.genexus.com/commwiki/servlet/wiki?31121,Storage+Provider+Property,

EXTERNAL STORAGE API



La Storage Provider API nos permite administrar los datos globales de la aplicación (cualquier tipo de archivos) en un almacenamiento externo. El almacenamiento externo puede ser Amazon, Azure, Bluemix o Google. El almacenamiento externo utilizado es el mismo que el configurado en la propiedad Storage Provider.

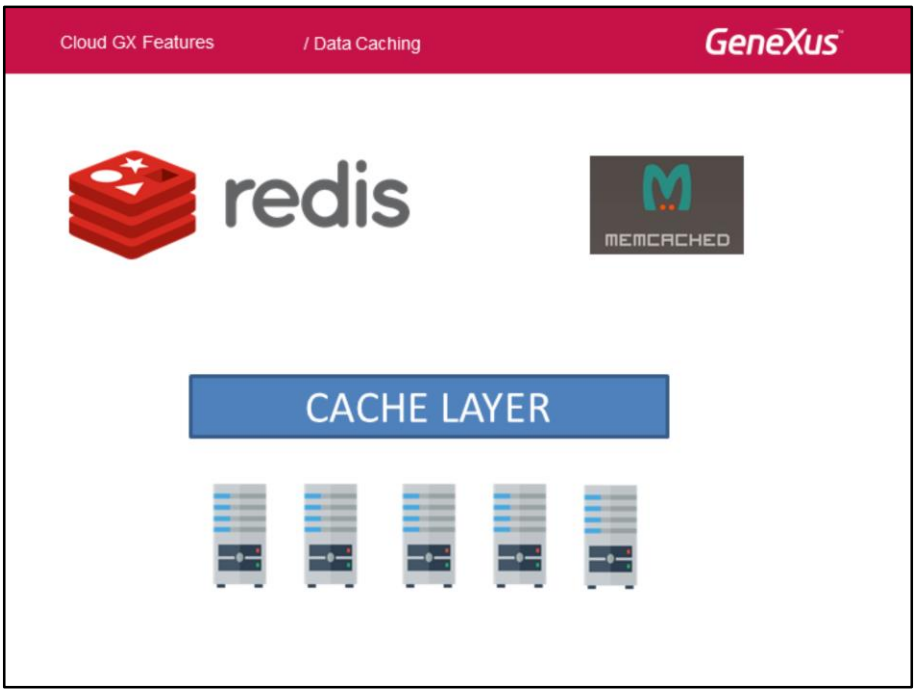
Vea:
http://wiki.genexus.com/commwiki/servlet/wiki?32087,Storage%20Provider%20API.



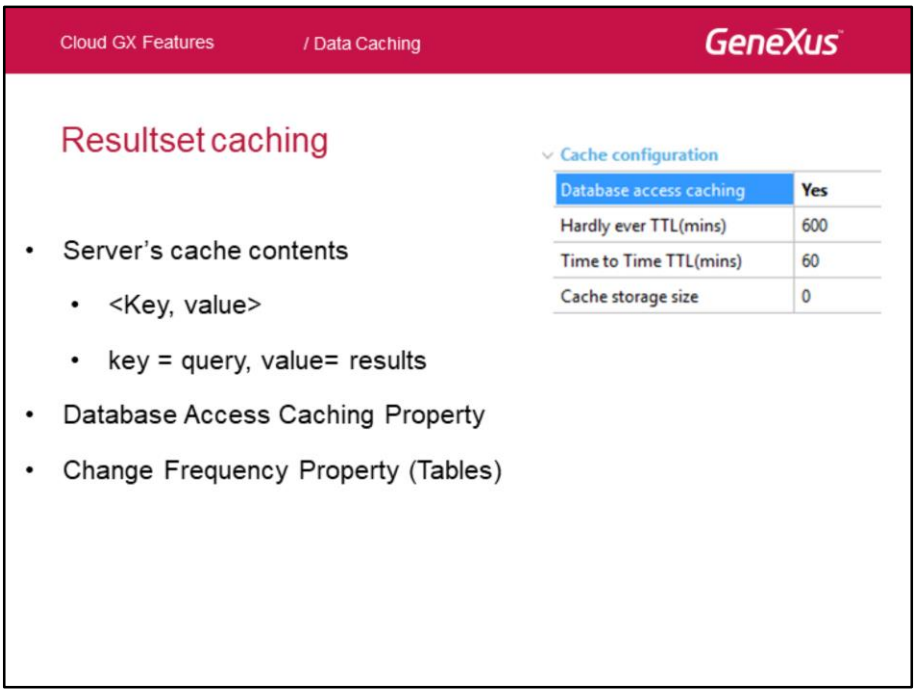
Los sistemas escalables de alto rendimiento a menudo necesitan alguna forma de acelerar las aplicaciones al reducir la carga de la base de datos. En la mayoría de los casos, para mejorar el rendimiento de las aplicaciones, la información debe recuperarse de los cachés rápidos en memoria, en lugar de leerlos desde la base de datos (que es un almacenamiento de disco persistente y más lento).

Las aplicaciones que son principalmente aplicaciones de lectura son un caso típico en el que este tipo de solución tiene un impacto positivo en el rendimiento y la escalabilidad de la aplicación. Otro caso típico es el de los objetos que hacen repetidamente la misma consulta.

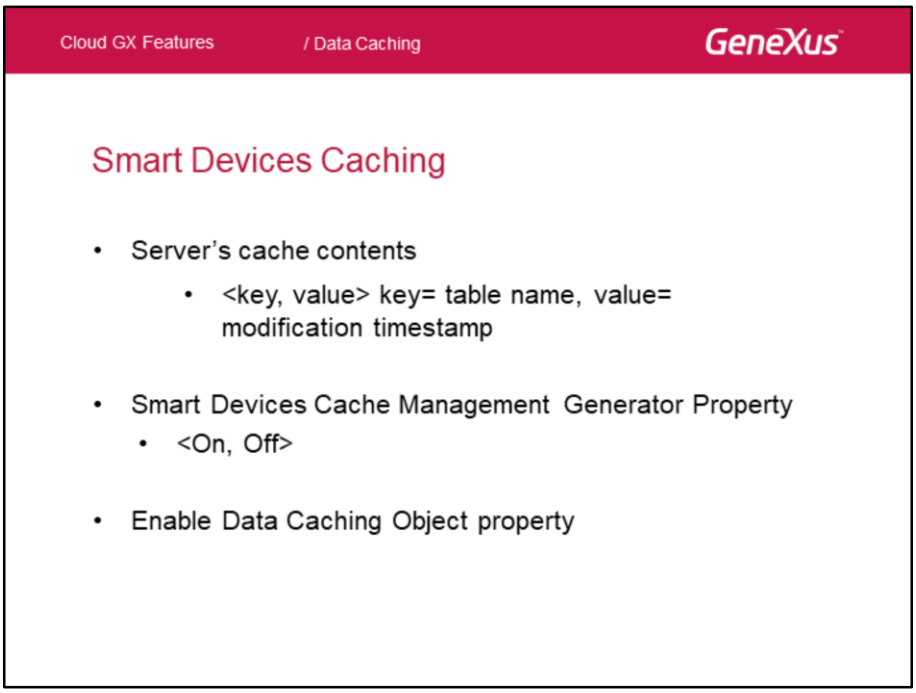
GeneXus permite recuperar datos del caché in-memory para aplicaciones web y de dispositivos inteligentes; pero, ¿qué ocurre si la aplicación se ejecuta en un entorno distribuido o en la nube? En un entorno distribuido, el caché local del servidor de aplicaciones es inútil. En este caso, necesitamos un sistema de caché de memoria distribuida para garantizar la validez de la información en caché. La solución en este escenario (principalmente aplicaciones que se ejecutan en la nube o en un entorno agrupado) es usar un caché distribuido.



Para entornos distribuidos, la aplicación GeneXus se puede configurar en tiempo de ejecución para usar Redis o Memcached



En el caché ResultSet, el conjunto de resultados de cada consulta de base de datos se almacena en un caché de memoria (llamado in-memory cache), y mientras el caché no caduca, se consulta en cada operación de lectura. Además, no es necesario acceder a la base de datos. Luego, en el in-memory caché, se almacena un par <clave, valor>, donde la clave es la consulta (incluidos los parámetros), y el valor es el conjunto de resultados de esta consulta.
La memoria caché se activa con la propiedad Database Accesss Caching y la validez de la memoria caché se gestiona a través de la propiedad Change Frequency de las tablas involucradas en la consulta. De hecho, en un join de tablas, la propiedad Change Frequency que es considerada es la más baja de todas las tablas involucradas.
Se aplica a las aplicaciones web y del lado del servidor (Restful services)) de las aplicaciones SD.



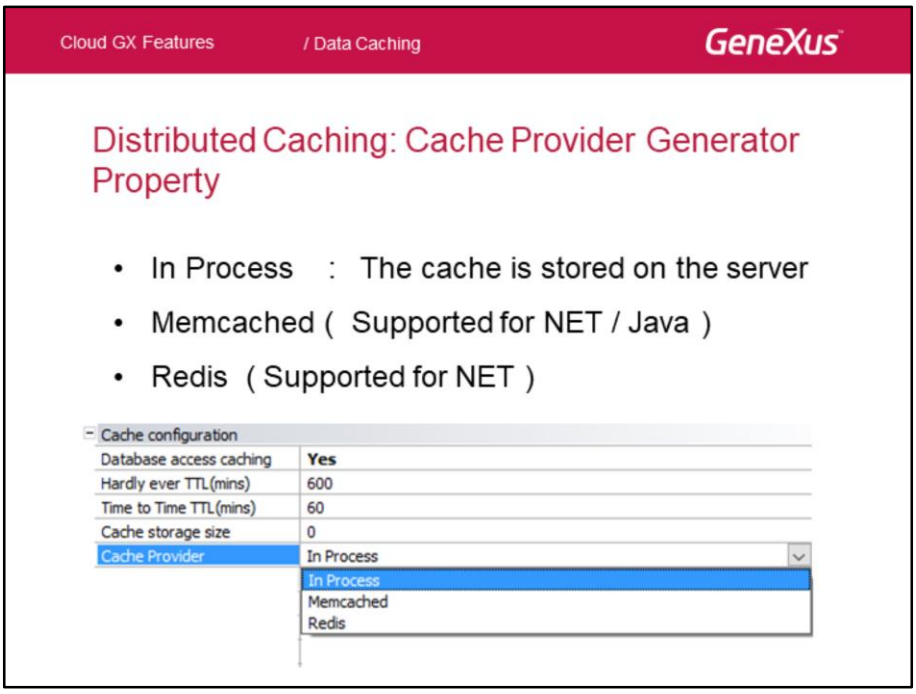
Evite el tráfico de datos entre el cliente y el servidor cuando los datos en el servidor permanecen sin cambios.

En el servidor, para determinar si los datos del cliente siguen siendo válidos, se necesitan dos elementos (estos son los elementos almacenados en el caché del servidor):

- Una lista de tablas
- La marca de tiempo (timestamp) de la última modificación de las tablas.

Cómo activar el almacenamiento en caché de Smart Devices
En primer lugar, debe activar la propiedad Cache Management y la propiedad Enable Data Chaching para cada objeto para el que se va a usar la memoria caché.

Cuando la propiedad de objeto Enable Data Caching es True, la propiedad Check for New Data se habilita, así como también la propiedad Check for New Data After Minutes Elapsed.



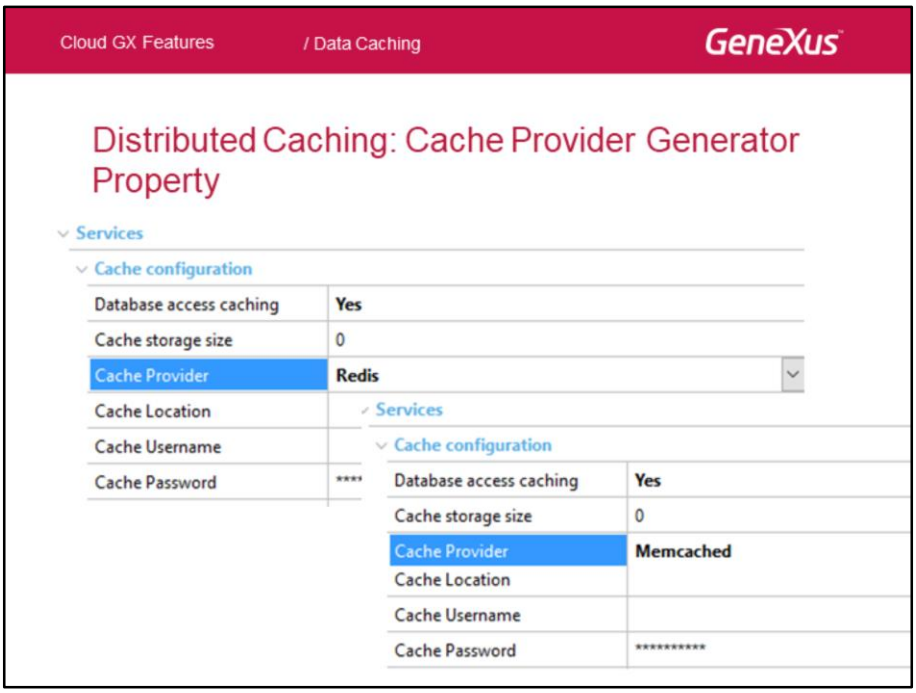
Para entornos distribuidos, la aplicación GeneXus se puede configurar en tiempo de ejecución para usar Redis o Memcached para ResultSet caching en GeneXus y Smart Devices también.

Cómo configurar el almacenamiento en caché distribuido en GeneXus

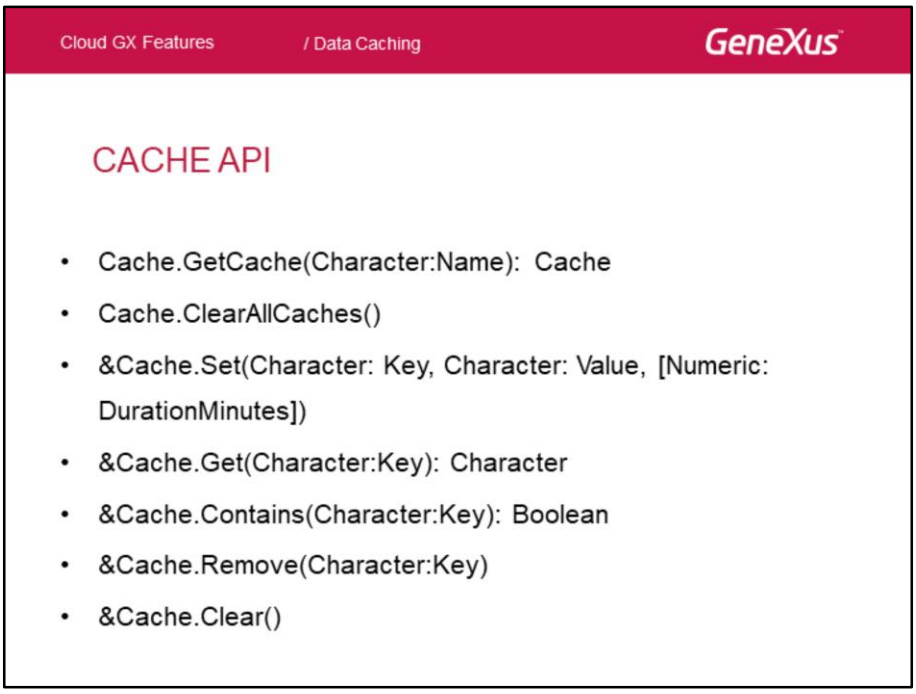
En primer lugar, configure el almacenamiento en caché en GeneXus. Es lo mismo que configurar el caché local del servidor. La forma de configurar el almacenamiento en caché distribuido es a través de la propiedad Cache Provider.

Nota: El caché es un almacenamiento in-memory de key-value, donde el par clave-valor depende de la plataforma de la aplicación. Para las aplicaciones SD, la clave corresponde a la tabla de la base de datos, y el valor es la marca de tiempo (timestamp) de modificación de la tabla. Para web y web services, la clave corresponde a la consulta con sus parámetros y el valor es el “resultset” de la consulta.

Más información:
http://wiki.genexus.com/commwiki/servlet/wiki?28136,Distributed+cache+in+GeneXus+applications,



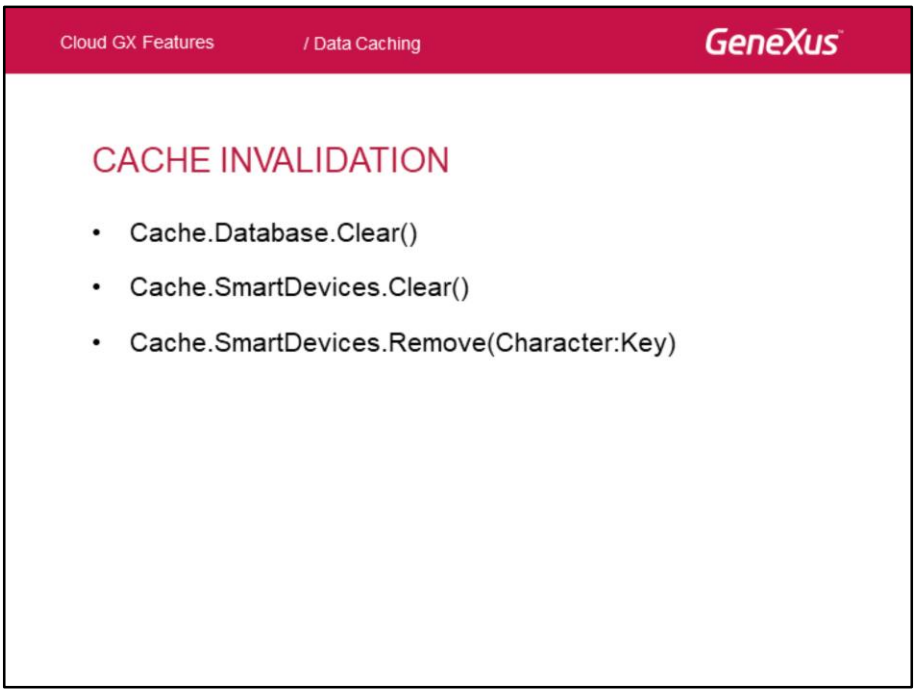
Cache Provider es una propiedad del generador que permite determinar el proveedor utilizado para el almacenamiento en caché distribuido.
Valores
In Process La memoria caché se almacena en el servidor y no se implementa ningún mecanismo de caché de datos.
Memcached La memoria caché se gestiona utilizando Memcached, que es un sistema de caché de objetos de memoria distribuida. Está soportado para NET y Java.
Redis El caché se gestiona utilizando Redis, que es un almacén de estructura de datos en memoria. Es soportado para NET.



Las aplicaciones a menudo tienen un conjunto de datos a los que se accede globalmente; es decir, datos que son iguales para cualquier usuario, que es el resultado de un proceso de inicialización. Una vez calculado, es útil tener estos datos accesibles y actualizados solo cuando la aplicación lo requiera.

Un posible escenario de uso es aquel en el que los datos no cambian con demasiada frecuencia (se pueden obtener de la base de datos o llamando a un servicio), y acceder a la base de datos o recuperar los datos llamando a un servicio web es demasiado costoso.

La Web Session podría ser una solución, pero está definida para cada usuario de la aplicación. Por el contrario, la API Caché está definida y es accesible para todos los usuarios de la aplicación.
La API Cache permite almacenar datos globales de una aplicación, acceder a los datos e invalidarlos cuando sea necesario. La caché puede ser local al servidor de aplicaciones, o puede ser una caché distribuida dependiendo de la propiedad Cache Provider.



Para el almacenamiento en caché de Smart Devices, así como el almacenamiento en caché ResultSet, también existen algunos métodos útiles para administrar la caducidad del caché.

En algunos casos, puede ser necesario borrar por completo el caché (por ejemplo, si necesita volver a ejecutar la consulta SQL y no recuperar los datos del caché de la base de datos). Esto es especialmente útil cuando se utiliza cualquier mecanismo de caché distribuido.

Para restablecer la memoria caché, puede usar el método *Cache.SmartDevices.Clear* y el método *Cache.Database.Clear* para SD y ResultSet, respectivamente.

http://wiki.genexus.com/commwiki/servlet/wiki?32105,Cache+API,

WHAT'S NEW IN GX15?

- **Data Storage**

  More database support on the Cloud

  File storage in the cloud

- **Caching**

  Distributed Caching in the Cloud (Redis, Memcached)
  Cache Data Type

  Invalidate cache