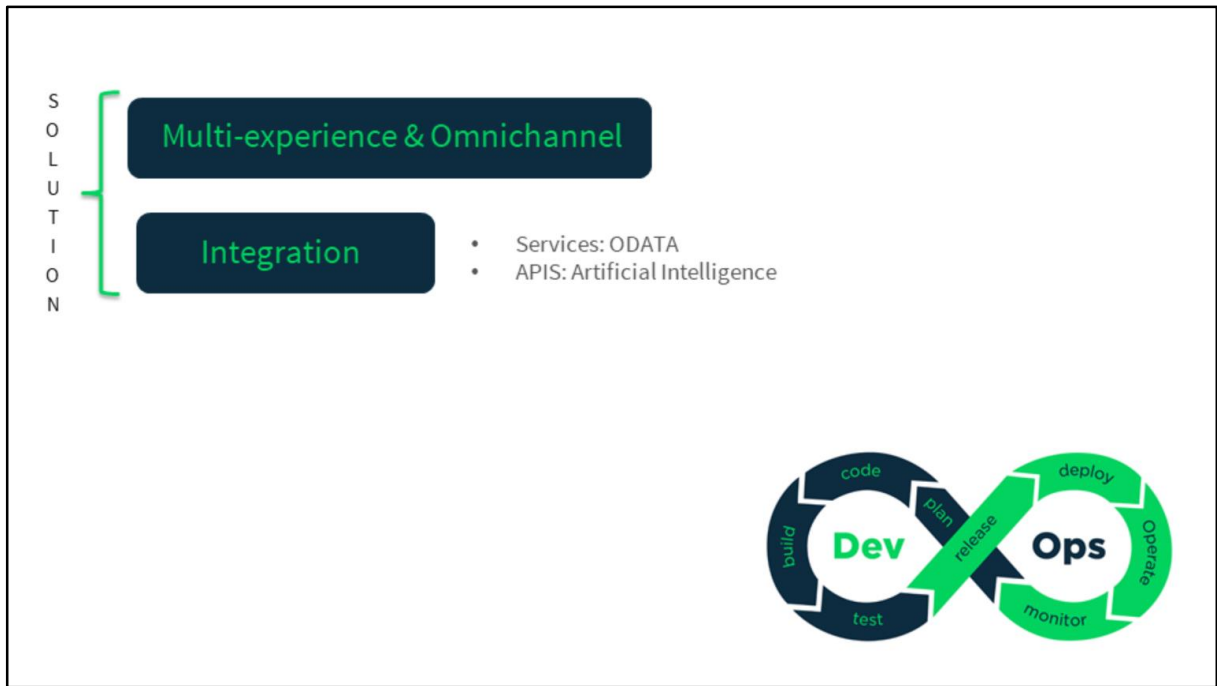


**GeneXus™**  
**The power of doing**

Integration

*GeneXus™ 16*



De todo lo que vimos en la introducción referido a integración, aquí solamente profundizaremos en dos aspectos: la integración con ODATA y la api de Inteligencia Artificial para integrarse con los servicios provistos por los proveedores más importantes de servicios cognitivos.



GeneXus

Algunos quizás ya sepan qué es, pero para los que no, el Open Data Protocol, mejor conocido como Odata, es un protocolo que utiliza REST, basado en estándares como HTTP, Atom/XML y JSON.

Se diferencia de los otros servicios REST en que provee uniformidad al momento de describir los datos y los modelos de datos.

Es una tecnología flexible que permite la interoperabilidad entre data sources, aplicaciones, servicios y clientes.

## Services

SOAP

OPEN API (Rest)

+

Services through Data Stores

ODATA

Data View → Transaction → Attributes → for each

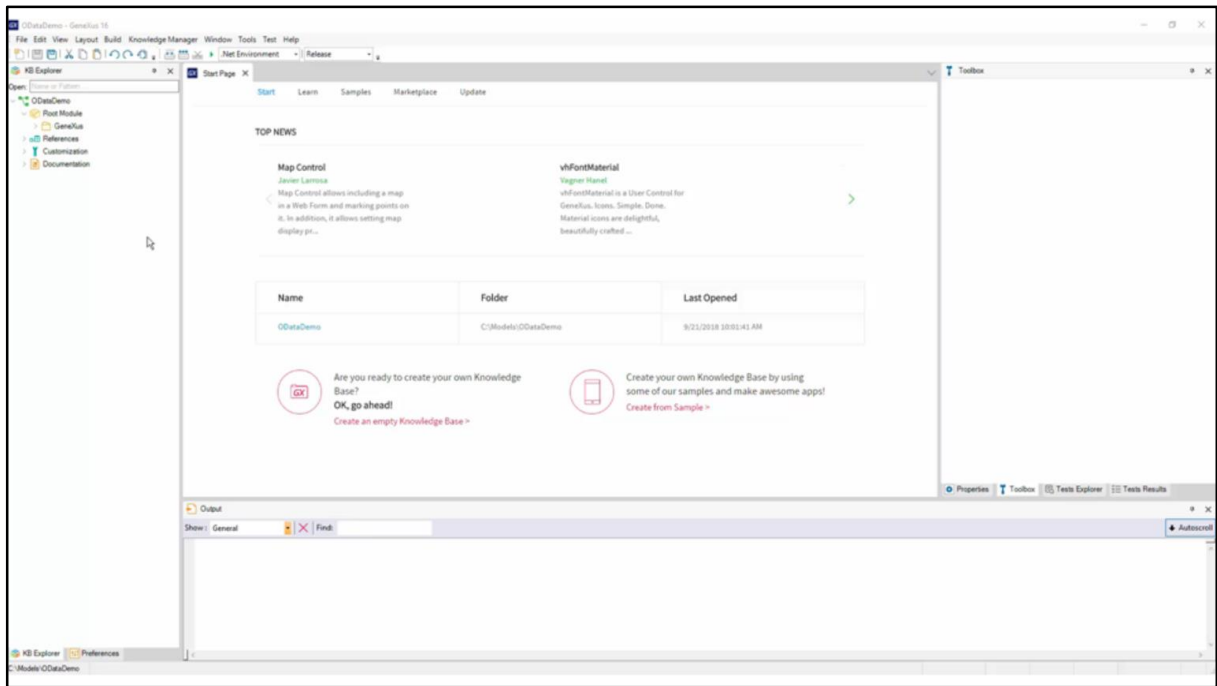
¿Qué beneficios trae incorporar OData a sus soluciones GeneXus?

Cuando se consume Odata desde una solución GeneXus, el servicio es visto como un data store más, y las entidades expuestas por el servicio son importadas como transacciones, lo que simplifica la forma de trabajar con los datos, permite usar for eachs para recorrer los datos, arrastrarlos a un web panel, aplicarle patterns o agregarle reglas de negocio a las transacciones.

Esto permite que los datos consumidos puedan ser utilizados para cualquier plataforma en la que se esté desarrollando Web o SD, ahorrando tiempo y esfuerzo a los desarrolladores, aumentando la productividad.



A continuación vemos una pequeña demo donde mostramos cómo consumir servicios Odata y cómo operar con ellos.



Aquí vemos una demo

# **Artificial Intelligence**



## Artificial Intelligence

IBM Watson

Microsoft Cognitive Services

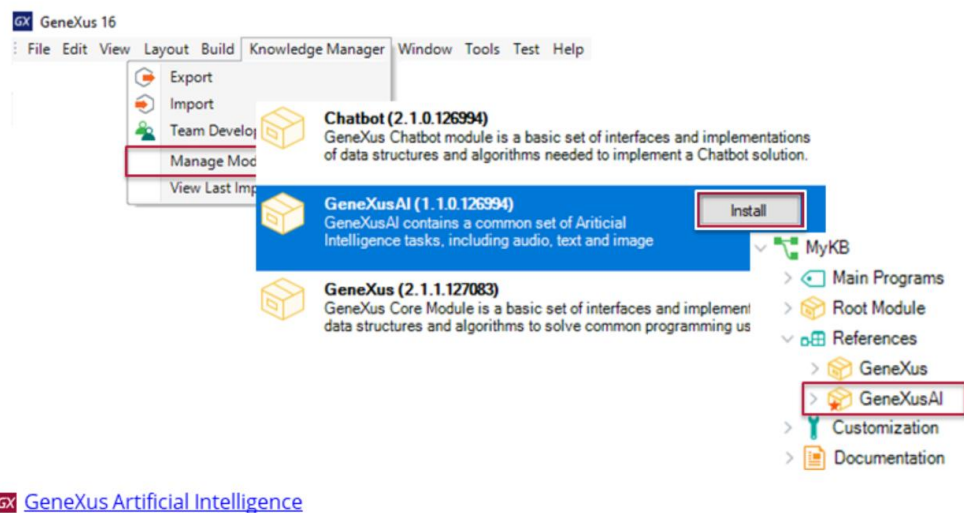
SAP Leonardo

Google Cloud Services

Amazon Web Services

Específicamente en lo que respecta a servicios cognitivos, estamos construyendo una API unificada para acceso a cualquiera de estas plataformas, logrando en GeneXus una definición multiplataforma.

# Artificial Intelligence

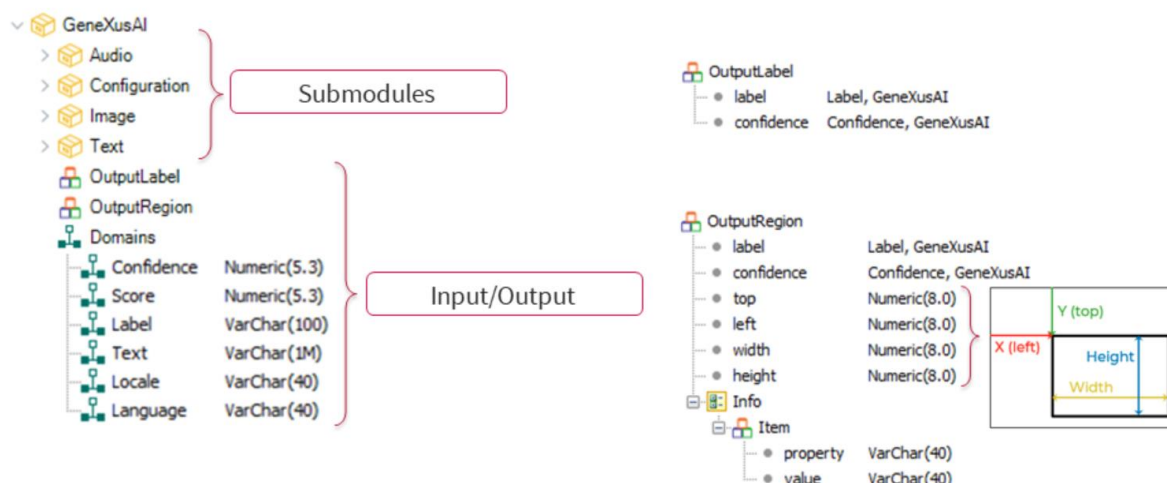


Para instalarlo simplemente nos dirigimos a Knowledge Manager > Manage Module References.

Allí vamos a ver un listado de módulos, entre ellos GeneXus AI, y lo instalamos haciendo clic en el botón "Install".

Una vez instalado, vamos a encontrar el módulo bajo el nodo "References" del KB Explorer, junto con el módulo con el core de GeneXus.

# Artificial Intelligence



Está constituido con cuatro sub-módulos que veremos en detalle a continuación.

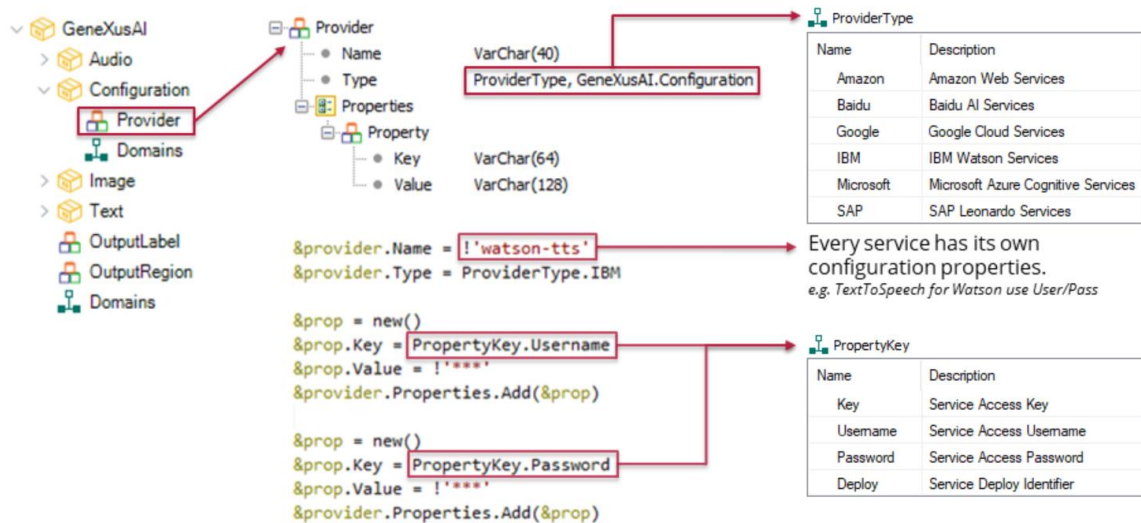
También disponemos de estructuras y dominios para manipular la entrada/salida de las tareas de AI, cada uno con una semántica definida.

- **Confidence** (confianza)
- **Score** (puntuación)
- **Label** (etiqueta)
- **Text** (texto)
- **Locale** (lenguaje de una región específica – por ejemplo, español de España)
- **Language** (lenguaje en el sentido genérico – por ejemplo, español, inglés)

Y por otro lado:

- **OutputLabel** nos brinda una etiqueta junto con su confianza
- **OutputRegion**, además de una etiqueta con su confianza, también nos indica una región rectangular.
  - Para visualizar cómo sería, veamos esta imagen (<X,Y> nos indica la esquina superior izquierda del rectángulo, y luego simplemente nos dice su altura y ancho).
  - Además nos brinda información adicional que nos podría devolver el proveedor.

# Artificial Intelligence



Comenzamos a estudiar el contenido del **módulo Configuration**.

- Cada Procedure de GeneXusAI tomará como entrada la configuración de un proveedor específico.
- Para ello debemos indicar un nombre, un tipo de proveedor en el dominio ProviderType, y una serie de propiedades
  - \* El **ProviderType** contiene los posibles proveedores que podemos seleccionar.

Veamos un ejemplo:

1. Primero completamos el primer nivel del SDT indicando un nombre y un proveedor (en este caso, IBM)
  - Notar que hacemos referencia al servicio cognitivo en el nombre porque cada servicio tiene sus propias propiedades de configuración.
2. Luego completamos el segundo nivel del SDT; es decir, sus properties.
  - Primero indicamos un Username
  - Luego indicamos un Password

Esto es porque TextToSpeech para IBM necesita Username/Password para funcionar, pero otros servicios podrían requerir una API key. Para saber qué propiedades podemos configurar nos apoyamos en el dominio enumerado PropertyKey.

# Artificial Intelligence

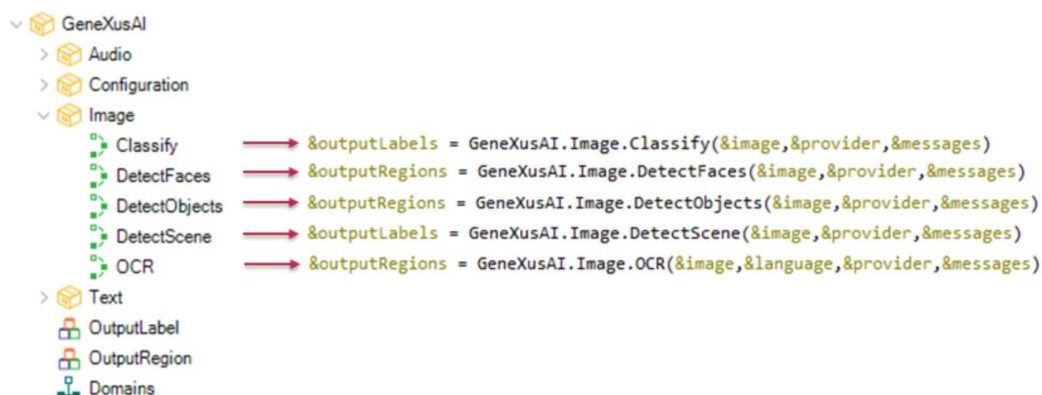


Ahora comencemos a ver el resto de los módulos.

En primer lugar, el **módulo Audio** nos permite manipular este tipo de contenido multimedia.

- **SpeechToText:** Nos permite transcribir audio a texto.
  - \* Notar cómo se recibe un audio, un lenguaje regional, la configuración para el proveedor y un messages. Con este último parámetro vamos a poder manejar errores que puedan ocurrir durante el prototipado o la ejecución – La especificación la pueden encontrar en la Wiki.
- **TextToSpeech:** Nos permite sintetizar texto en audio.
  - \* De forma análoga, recibe un texto, un tipo de voz, un lenguaje regional, etc. y devuelve un audio con el texto hablado.

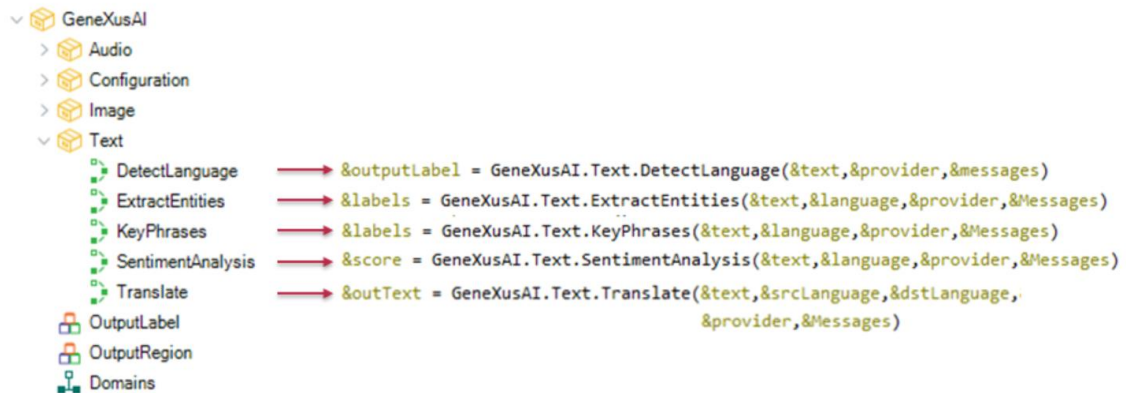
# Artificial Intelligence



En el **módulo Image** tenemos:

- **Classify:** Dada una imagen, nos devuelve una serie de categorías que describen la imagen (por eso su salida es OutputLabels – en plural).
- **DetectFaces:** Data una imagen, reconoce caras y puede darnos información genérica sobre cada una (por ejemplo su sexo, edad estimada, etc.) Su salida es un OutputRegions (también plural) ya que nos indica la ubicación de cada rostro en un rectángulo.
- **DetectObjects:** Similar al caso anterior pero para objetos.
- **DetectScene:** Nos describe el/los escenarios de una imagen (por ejemplo si se trata de una ciudad, playa, desierto, etc.)
- **OCR:** Es una sigla para Optical Character Recognition, y nos permite extraer texto desde una imagen.

# Artificial Intelligence



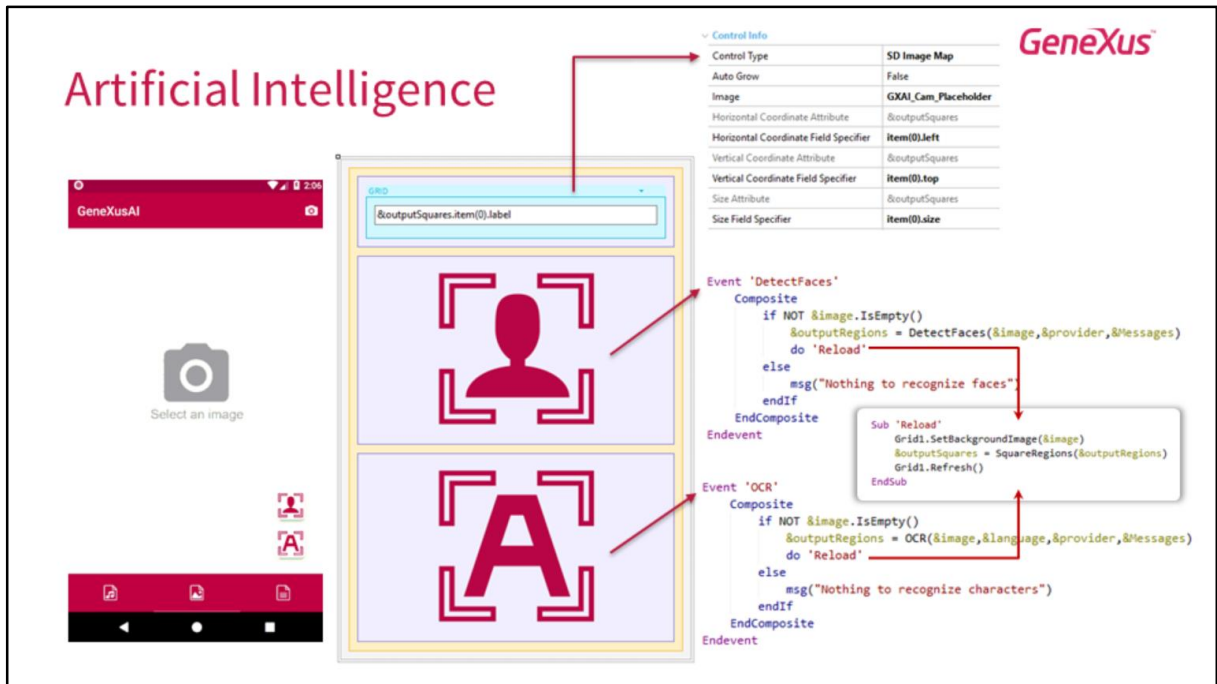
El **módulo Text** nos permite manejar aspectos del lenguaje natural, y está compuesto por:

- **DetectLanguage:** Permite detectar un idioma a partir de texto escrito. Su salida es un OutputLabel ya que no sólo nos dice el idioma, sino también un nivel de confianza de la detección.
- **ExtractEntities:** Extrae entidades a partir de texto, por ejemplo organizaciones, ciudades, nombres de personas, etc.
- **KeyPhrases:** Permite extraer las frases (o palabras) más relevantes del texto.
- **SentimentAnalysis:** Permite inferir el nivel de negatividad o positividad de un texto en 0 y 1, siendo 0.5 el sentimiento neutro.
- **Translate:** Nos permite traducir texto desde un lenguaje origen a uno destino.



Proof of Concept





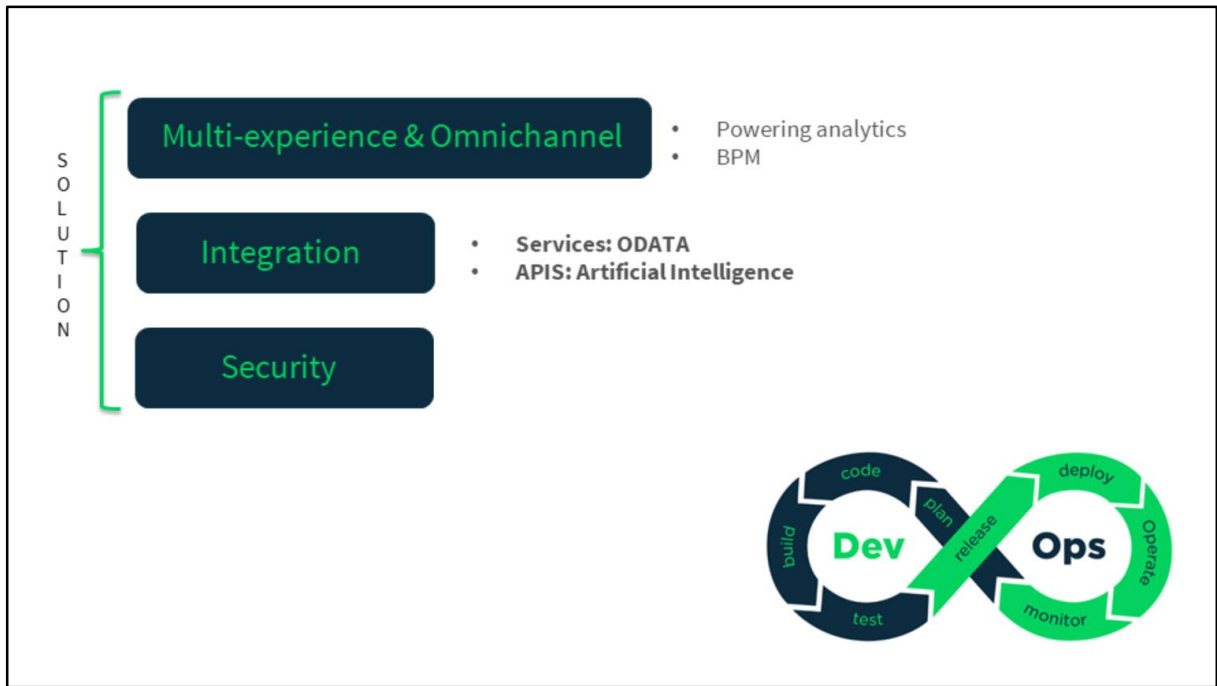
Veamos un ejemplo sencillo.

(\* Ver animación – Se muestra un caso de DetectFaces y otro de OCR \*)

1. Este es el layout asociado a esa pantalla. Tenemos un grid, y dos botones con las imágenes que vimos en la animación.
2. El Grid es de tipo SD Image Map para poder “dibujar los rectángulos” (en realidad, de momento, podemos dibujar cuadrados).
3. El primer botón tiene asociado un evento que verifica que la imagen no sea vacía e invoca al procedure DetectFaces de GeneXusAI.
4. Análogamente, el segundo botón invoca al procedure OCR.

La subrutina ‘Reload’ que vemos allí simplemente establece la imagen de fondo, obtiene los cuadrados a partir de los rectángulos del outputRegion, y finalmente realiza un “refresh” del grid.

¡Muy simple!



En esta sección vimos estos aspectos de integración.

En lo que sigue entraremos en el tercer gran punto en lo que hace a la solución desarrollada: los aspectos de seguridad, sumado a cómo mejorar el análisis de datos con reporting, y a los cambios en BPM.

**GeneXus™**  
**The power of doing**