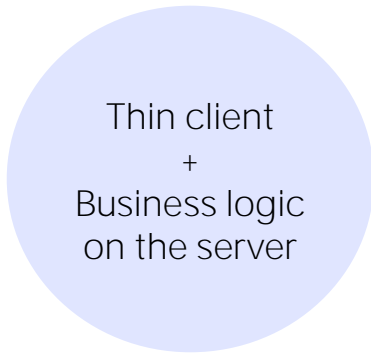


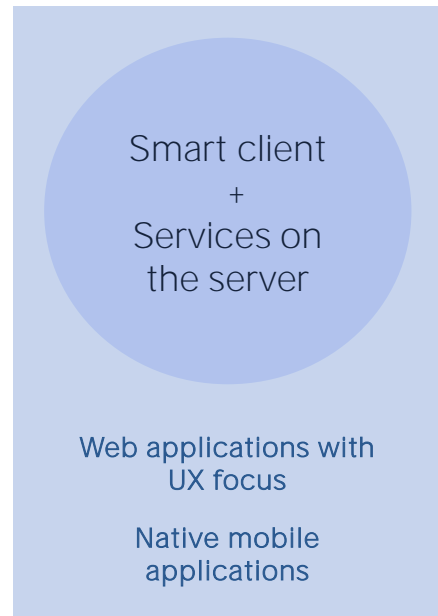
# Web Screens with Customer-facing Focus

Introduction

**GeneXus**<sup>™</sup>



Web applications with  
Back-office focus



**We've** said before that the applications' architecture conditions, among other things, the way they are programmed.

In this module we will focus on web applications that will run on a smart client, i.e. customer-facing applications that allow us to provide the best possible user experience.



Back-office

Web panels

### Travel Agency

Country Name: France

Attraction Name From:

Attraction Name To:

Attraction Name	Attraction Photo	Trips
<a href="#">Louvre Museum</a>		0 <a href="#">New trip</a>
<a href="#">Eiffel Tower</a>		0 <a href="#">New trip</a>
<a href="#">Notre Dame Cathedral</a>		0 <a href="#">New trip</a>

Total Trips: 0

### Travel Agency

Attraction Name: Eiffel Tower


Country Id: 2

Country Name: France

City Name: Paris

Category Name: Monument

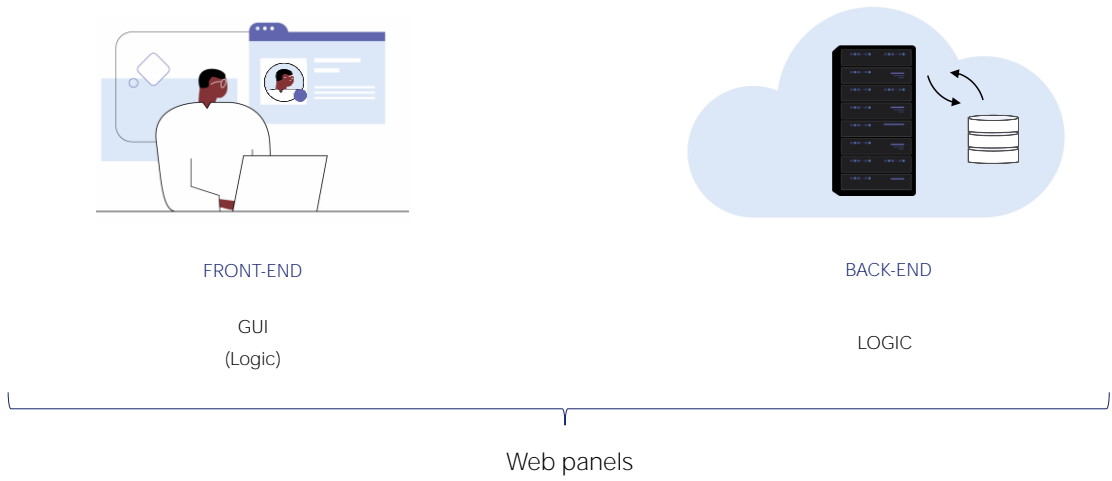
Trips: 2



City Id	City Name	Attractions
1	Paris	3

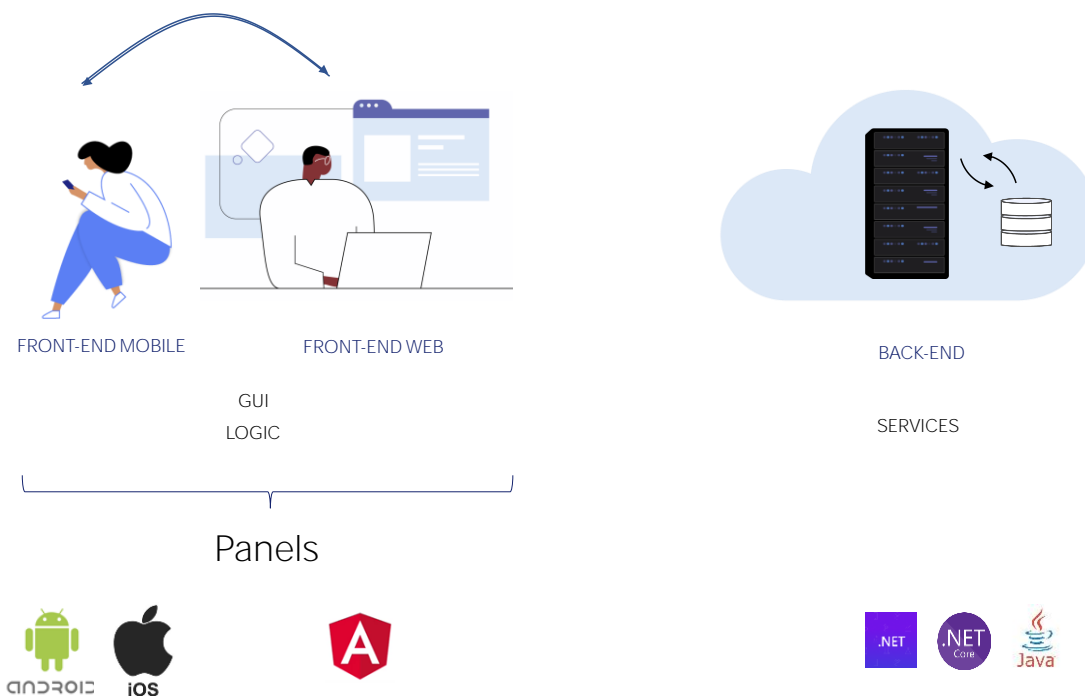
Total Attractions: 3

So far, we have seen how to use web panel objects for developing the travel agency's back-office web application.



We've also seen how web panels are designed and programmed, keeping in mind that most of their logic runs on the server.

As we are now interested in programming that is focused on the user experience, we should use an object that helps concentrate the logic on the client side of the application.



Every screen that the user interacts with (from both web and native applications) has a visual part that is displayed on the client and an associated logic. In the applications we are now interested in –customer-facing applications with the best possible user experience– this logic is executed on the client, and the server has services that provide the client with functions related to retrieving and maintaining data.

The GeneXus object that allows us to add code that runs on the client, as well as code that will run on the server through the invocation of back-end services, is called Panel object. This object will allow us to develop customer-facing applications, including web and native mobile applications.

In order to have these complex applications on the client side (front-end), we will use Angular to generate the web client and Android or iOS for a native application client. GeneXus generates the server part (back-end) in .Net, .Net Core or Java.

We will look at the Panel object starting with its logic, and then we will focus on screen design. **What's** interesting about this object is that the screens designed for the web application can then be used in the native application and vice versa, because the programming method is valid for both platforms.

This is just an introduction to the Panel object, particularly for web screen development. In other courses, such as web application

development with Angular or native mobile application development, you can learn more about the use of this object.

*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)