

Architecture of Online Applications



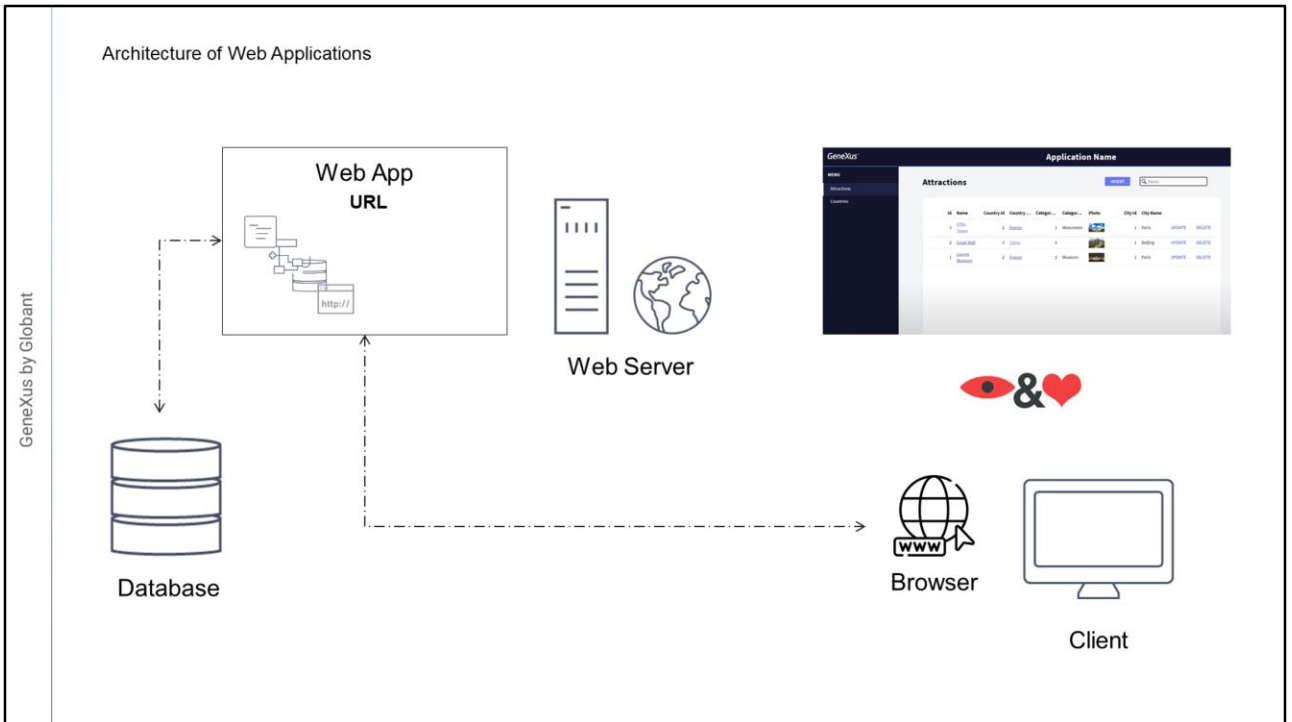
Diego Marranghello

Web Apps Architecture

Online Apps Architecture

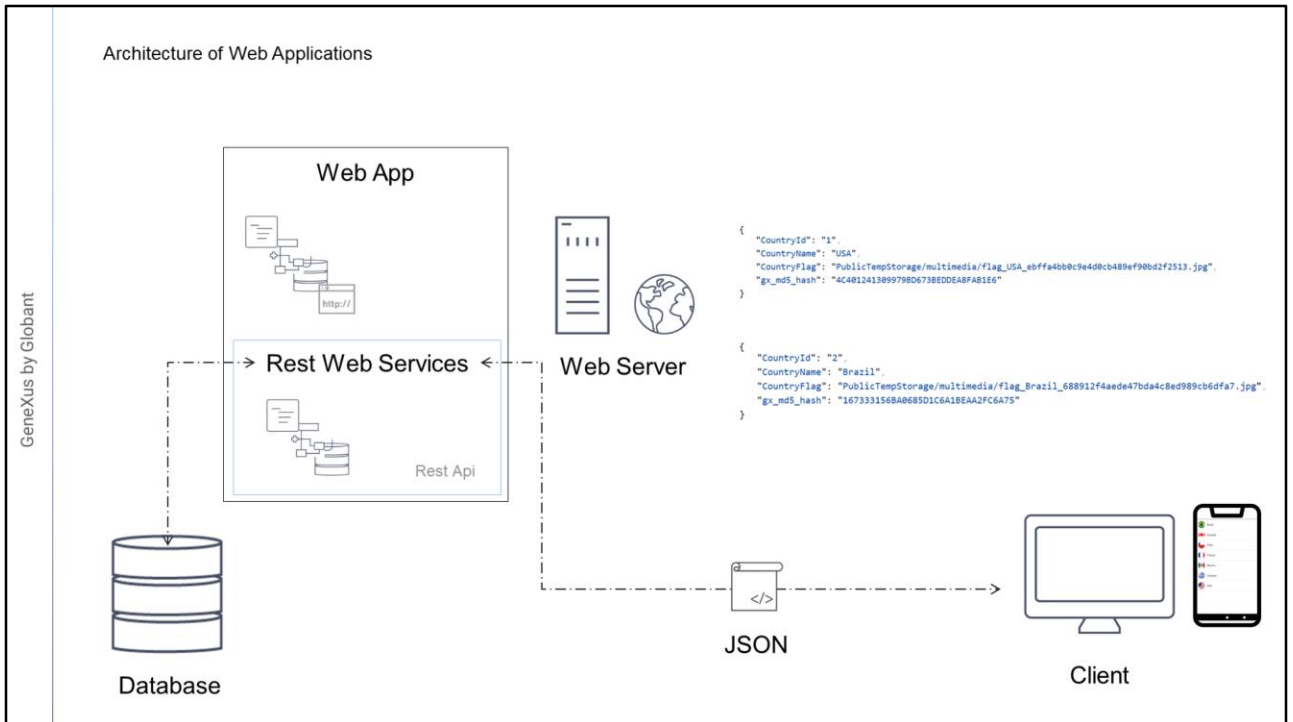
GeneXus Project Navigator (GPN)

We will start by reviewing the architecture of Web applications, then we will draw a parallel with online applications for mobile devices, and finally we will see a special case of these applications when using GeneXus Project Navigator.



In Web applications, the client requests access to certain content from a browser, such as the Web Panel of the countries Work With for Web. The server will query the database, get the information on the countries, and generate the corresponding HTML code that will be sent to the web browser to interpret it and display it to the user.

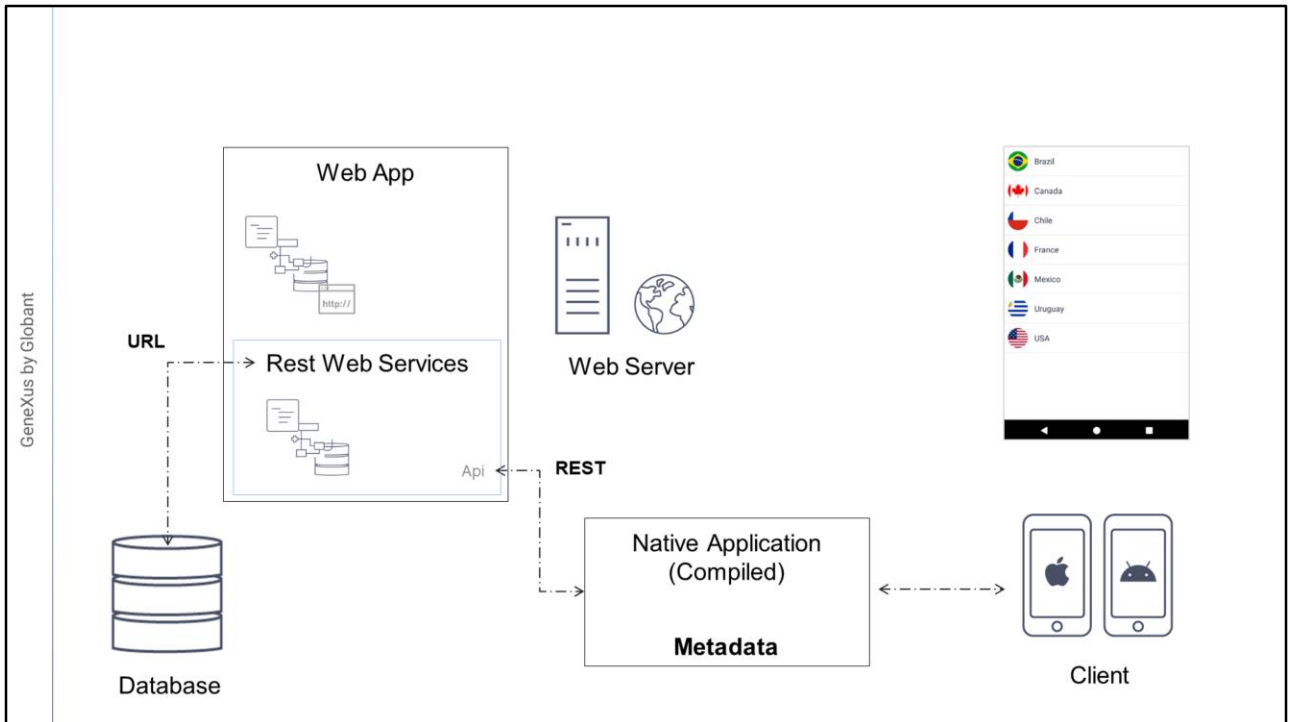
So why don't we use the smart device's browser to access the same web interface of the server? The problem is that we wouldn't have the same look & feel nor access to integration with the device hardware natively; that's why we need a different alternative.



We could also have a service layer in our application, especially a REST API made up of Business Components, Processes, or Data Providers. One of the advantages of REST APIs is that they not only allow us to retrieve information but also provide adequate mechanisms for updating it, in addition to using JSON as the output format so that the data is independent of its representation.

Using this REST layer, when the web server is requested to access a certain resource—for example, a Data Provider that returns a JSON with country information—the server will query the corresponding information in the database and generate a JSON output, which is finally returned to the client. Then the client will have to process that information and display it in a particular way in the user interface.

Considering the latter case, the client could be a device such as a phone and the interface could be a Panel object, solving data access in a natural way and leaving representation to the client side.

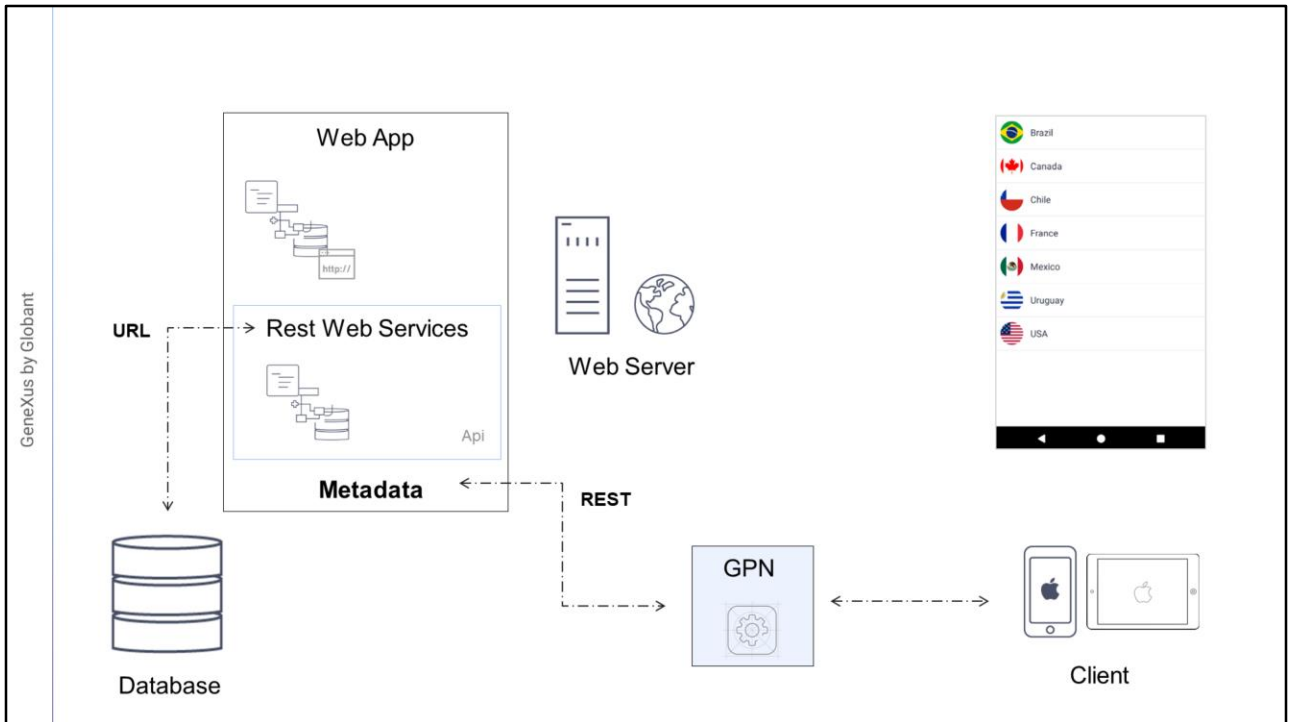


Now we can define the architecture of Online applications.

As we have already seen, each mobile device platform has its own language: Java in the case of Android and Swift in the case of Apple. When the application is compiled, a file is generated and installed on the device. This file will contain all the logic to handle the user interface, such as images and other resources, including the URLs needed for each resource of the REST API; this is called application metadata.

When the application is run, it accesses the Web server to run the REST Services, which will return the current data. This data will be processed by the application and displayed to the user.

An advantage is that the application will never access the Database. This will always be done through the service layer (the REST API) and this layer will be independent of the device we are using; that is, if we compile the application for Android and for iOS, both applications will use the same service layer.



Basically, it is a lightweight interpreter with the logic to read the metadata and application images from the Web Server. It will be able to decode all this information, invoking, if necessary, the REST services needed to obtain the responses with the data and build the corresponding interface on the device –that is, the one displayed to the user.

To learn more about this mechanism, you can visit the GeneXus wiki.

The image shows the App Store listing for "GeneXus Project Navigator" by GeneXus S.A. The app is designed for iPad, has a 5.0 rating, and is free. To the right, a developer menu is open, showing options like "Last Impact", "Last Navigation", "List Database", and "Show QR Codes". Below the app listing, there are three screenshots of the app's interface on iPad and iPhone. To the right of the screenshots, there is a QR code for iOS and a link to "Services URL for GPN". Below the QR code, there is a section for "GeneXus Project Navigator" with a "Download" link. At the bottom, a red banner states "Now also available on Android".

A particular case of online applications is a mechanism provided by GeneXus to run an application in interpreted mode. This mechanism is used when we want to run the application on an iOS device and we don't have a Mac computer to compile it.

For these cases, GeneXus provides an application called GPN (GeneXus Project Navigator) that can be downloaded from the Apple Store and installed on the device. In addition, in the Developer Menu there is a download link and a link to add the application to the GPN.

GPN is a native iOS application that allows you to navigate mobile device applications created with GeneXus as if you were using a Browser, by choosing a URL (corresponding to a main object of the application).

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com