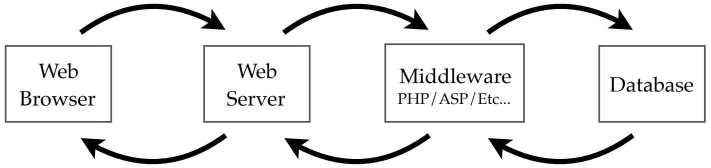
Comparing Web Application Platforms

Lets look at a simple example of a typical web application using some form of middleware such as PHP, ASP, JSP, Perl, etc...

An organization wants a web app to display a combo box, otherwise known as a drop down box. When the user clicks the drop down it should display a list of choices from a table in the database.

The Old Middleware Paradigm



```
<html>
<script>
// lots of code here
// ask server for data
// add data to 'combo' below
</script>
<body>
  <div id="combo"></div>
</body>
```

Old Paradigm HTML

To make this happen under the old paradigm, the developer has to write code in several places. Here we show the HTML page. We've shown where the code goes in the script. Script code is Javascript.

Note that the combo box is just a "div". In a page with a lot of items on display there will be a lot of DIVs to draw them all. As HTML documents grow it takes more time to alter them because the DIVs make it difficult to read the page.

The developer is far from done. Once the code in the HTML page is complete, only the request and display of the data has been accomplished.

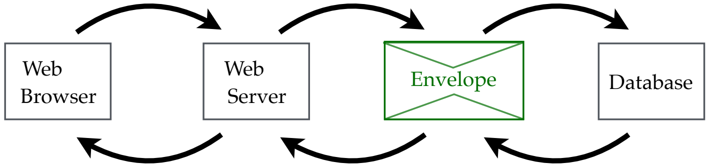
In the middleware we now need to write code to accept the request, talk to the database, make sure we have permission to see the data, fetch the data and then return it to the web page in a format that our HTML code can accept.

```
// accept request

// do we have permission?

// talk to database

// format and return data
```

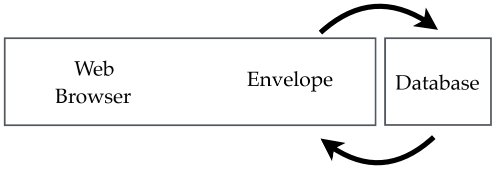
Old Paradigm Server

Introducing "Envelope"
The first truly new web platform since the beginning of the web.

Envelope is a completely new platform that prevents the developer from writing code in the middleware by replacing it with a fixed executable. This forces all code to be written for the web browser or the database. The Envelope architecture looks like the Middleware architecture...



However, in practice, a developer can build applications using the desktop architecture model. This greatly simplifies development resulting in reduced development time. The resulting reduction in code volume creates numerous security, maintenance and performance benefits.



```
<script>
//Link to Envelope Framework
</script>
<body>
  <gs-combo src="db.table"
     column="project_name">
     </gs-combo>
</body>

// No code anywhere else
```

Envelope Paradigm is HTML Only

In order to work like a desktop architecture, Envelope ships with a complete Javascript toolkit for building database applications. This Javascript toolkit is based on a technology called "Web Components".

The Envelope framework is very different from other frameworks in that it provides HTML tags that can be used to display rich interface objects on a web page and connect those objects to the database without the developer having to write Javascript or Middleware code.