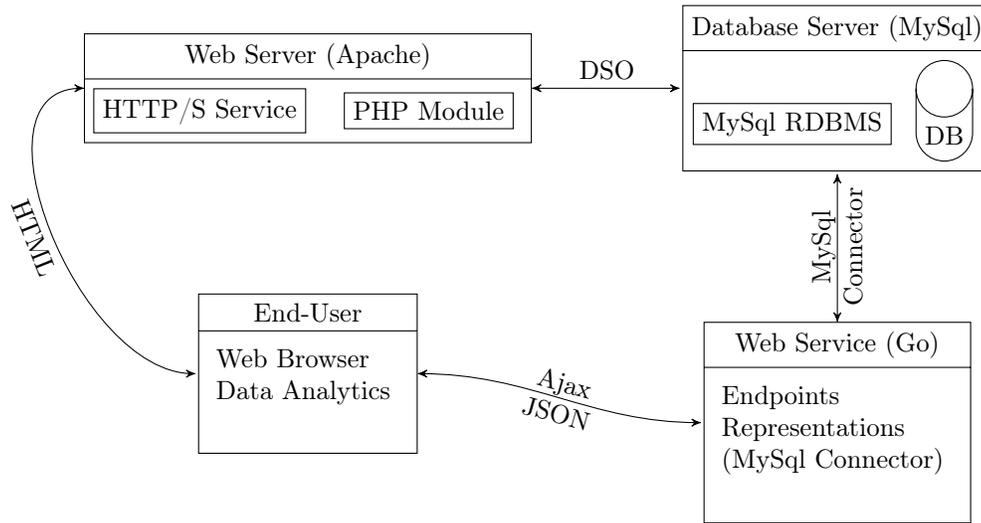


F19 ESOF 322 Software Engineering - Architecture Diagram
 Phillip J. Curtiss, Associate Professor, Department of Computer Science
 October 1, 2019



Description

The current functioning of AbOut relies on the PHP module that is executed on the server - server-side processing - with the results of that processing being passed to the requester. The result of the processing - most of the time - will be HTML, prepared on the server with any data retrieved from the MySQL DB and placed in HTML to be sent - and eventually rendered - in the browser. It is also possible for Ajax calls - originating in the browser - to make requests to PHP end-points which would likely send data - but could also send HTML - in response to the Ajax request. In such cases, the Ajax handler routine would then replace one or more elements in the currently rendered document object model (DOM) with the new information retrieved via the Ajax request.

As F19 ESOF 322 works to create a REST-based web service API to AbOut, it is important to keep in mind that calling these web service (WS) API end-points may not be possible from the PHP module inside the Apache web server. For one, it is old, and secondly, it is just not a common thing to do. Therefore, even if possible, http-client calls from the PHP module may be limited in the methods and formulation of the requests. This will be a limiter of the task of migrating from direct PHP database calls, to a web service approach. But there is hope! The front-end is not limited in making http-client requests, because, well, it is an http-client.

Therefore, as queries become available through the web service, it should likewise be the case that requests of these end-points should migrate to originate in the end-user space (browser) through Ajax calls, with responses being rendered by the end-user space. It should be noted that this is a shift in responsibility for creating views. Presently view logic is typically processed by the PHP module - running in the server - with the resultant HTML being shipped to the end-user space for rendering. As Ajax calls originate from the end-user space to the web service - where data is retrieved through JSON encoded representations - it will now be the end-user space that is responsible for computing the view and rendering the view.

Note that the end-user space may make requests to PHP end-points and render the resultant HTML, as well as make requests to the web service where it will need to render the computed view. As more and more of the web service API is completed, less and less corresponding PHP end-points should be requested by the end-user space.

Consequences Fundamentally, there should be a partitioning of responsibility of the application, with the end-user space responsible for computing the view(s) and rendering them; the web service responsible for the business logic - retrieving data models and computing representations from these, and shipping them to the requester; the database system implementing a data model for the application, filled with data, and providing an API through stored procedures and views - used by the web service layer - to interface with the data models and their content.

Later Stages Later stages should focus on transforming the 1:1 nature of the web service representations to decouple these from the data model, and move to an end-user framework to ease with developing views, managing end-points, and consuming representations to be used by the views.