

## Technologies of iPokr

In the light of ESPN's recent coverage of The World Series of Poker and other televised poker tournaments, online poker has literally exploded in popularity. This popularity has produced hundreds of poker sites and applications for playing poker over the internet. However, these poker sites utilize old, deprecated, or sub-optimal technologies that ultimately hinder the users' experience or even ability to play poker online. It is the goal of this project to develop a distributed online poker system that utilizes modern technologies for the benefit of both poker players and providers.

The iPokr system is to be a web-based, distributed poker system that maximizes the availability of the system to the target audience, while maintaining security and performance. First and foremost, to make the system available and usable to the widest possible audience, web standards will be employed to develop the client side interface. Whereas the majority, if not all, of the current poker systems require the download and installation of programs, and/or runtimes such as Java or .NET, iPokr will require nothing more than a standards compliant web browser. This allows for the flexibility of a player to connect via a desktop or laptop computer, PDA, cell-phone, or any number of internet enabled devices – perhaps even via an automotive dashboard touch screen console.

The web standards to be utilized include XHTML (with Strict Doctype), CSS, and unobtrusive DOM scripting via JavaScript. The usage of XHTML rather than standard HTML decreases code size and coding time as browser inconsistencies are greatly reduced. Additionally, as new browsers and user agents are released, iPokr will not need to be re-tooled as it will already be compliant with standards-based agents. JavaScript (specifically AJAX) will be utilized extensively to provide interaction resembling more a desktop application than a webpage. However, iPokr will be designed using the unobtrusive DOM scripting methodology thus providing a fully functional fall back if JavaScript is absent.

## Technologies of iPokr

With the distributed web-services model, it is possible to modularize the iPokr system. It is this modularization that will allow for flexible scaling and robustness as servers can be added at anytime to handle system growth and server failures can be hidden as traffic is routed to other operating servers. The iPokr model, therefore, consists of a central gateway webserver and any number of game servers and account servers. While, this central gateway server would clearly become the bottleneck of system traffic, it could easily be remodeled as a load balancing server that redirects users to other central processing servers which would then be the entry point to the system. However, due to resource availability and project scoping, the role of these central processing servers and the gateway server will be combined to one central server which handles the client communication and interaction to the game and account servers.

On the central server, various technologies exist to support a website interface to the iPokr system. These technologies include ASP, PHP, ColdFusion, JSP, CGI scripts, etc. Again, due to resource availability and cost concerns, PHP – as the freely available open-source language – will be used, running with the Apache web server on a Linux machine. More flexibility exists however, in the selection of tools for the game and account servers. The web-service model's single greatest strength is the ability to, in theory, run different programs in different environments and languages on distributed machines and still allows interactivity and communication. As a proof of concept, the game and account servers will be developed in Java utilizing J2EE technologies.

With differing environments, the various servers have three main options for communication, namely sockets, XML-RPC and SOAP. The performance costs for sockets almost immediately rule them out, in addition to lack of flexibility in communication with difference environments and inability to scale gracefully. SOAP and XML-RPC on the other hand, have many of the same strengths in terms of scalability, flexibility and performance.

## Technologies of iPokr

However, SOAP has a higher degree of robustness in terms of user-defined types and extensibility. This results in added complexity which highlights XML-RPC's greatest advantage – simplicity. The added robustness of SOAP outweighs its added complexity, and is therefore the chosen model of communication between servers.

Although the server to server communication will be accomplished with SOAP via HTTP, the client (browser) to central server communication will remain standard HTTP requests though modified with AJAX. Though it is possible to construct, send, and receive SOAP messages with JavaScript there are a number of significant drawbacks. The handling of these SOAP messages – namely creation and parsing – incurs a high performance cost on the client. Although this cost may or may not be noticeable on the vast majority of users' systems, the mission of iPokr is to reach the widest possible audience. Hence, the resource limitations of many handheld and portable devices preclude the utilization of JavaScript SOAP handling. The second, and more significant drawback, is the incurred requirement that the user have JavaScript, have it enabled, and require the proper version on their device. This again reduces the target audience and is therefore unacceptable.

Despite not *requiring* JavaScript, iPokr will utilize it if present on the user's device. To develop a more application-like feel, iPokr will make extensive use of the XMLHttpRequest object, otherwise called AJAX. This JavaScript object allows client to server communication, in the background so to speak. The user agent is able to send requests to the server for processing, which are then translated to SOAP requests to/from the various servers, finally resulting in a response to the client. The client is then able to alter the state of the webpage to appropriately reflect the new information. As all of this occurs behind the scenes without page reloads, the user interacts with the webpage in a manner similar to a normal desktop application. The webpage is to be constructed in such a manner that this AJAX interaction is value-added. That is

## Technologies of iPokr

to say, the website is first built to be fully functional without JavaScript and the AJAX features are developed as add-ons so that users without JavaScript will not be excluded from using iPokr.

As HTTP is a stateless protocol, there are a few other issues to be addressed. A new technology, termed Comet, is a new client-server model for the internet where the web is no longer client-get only but server-push as well. This model, however, requires special server hardware and additional resources so as to be infeasible and thus discarded as a viable option. Other techniques under consideration and research involve the central server and the issue of 'state'. One possible solution is to run a Java server on the same machine with the PHP installation such that PHP can talk to this Java server via sockets (same machine, thus low resource hit) and the Java server can then relay communication via SOAP to the other servers. The drawback to this option is the development of a protocol to define this socket communication. The creation of this specialized protocol seems redundant with the existence of SOAP and so this option is therefore discarded. An additional solution is a specialized installation of PHP and JSP where the two languages can be written in the same webpage and their two environments can communicate and share resources and objects. Again, the specialized installation requirement makes this alternative unviable. The selected solution is the utilization of database storage of PHP session data via an SQL database, most likely MySQL given its availability and cost.

Although there are many implementation details with distributed computing, the technologies selected for iPokr will be more than capable of solving these issues as they arise.