

# KAAZING for GAMING

A BUSINESS WHITEPAPER

## Friends Don't Let Friends Long-Poll

In the world of online gaming, HTML5 promises to be a game-changer. HTML5 is a collection of technologies such as hardware-accelerated graphics, video and audio that can make a Web game feel like a native application. The most far-reaching innovation in HTML5 is its transport layer specification, known as WebSocket. HTML5 WebSocket enables full-duplex, low latency, high frequency, “live” communication between a browser and a back-end server, or between two servers. Until now, live communication over the Web could only be simulated using complex work-around techniques such as Comet, Reverse Ajax and long polling. That’s because *the legacy Web* was designed to handle request-and-response, not live full-duplex communication. HTML5 WebSocket has the potential to change all that, to transform the static legacy Web into the *living Web* by introducing native real-time communication between two endpoints for the first time ever. But there’s a lot more to the story than HTML5 WebSocket. So developers building game titles and platforms for the living Web need to fully understand both the promise and the limitations of this brave and exciting new world.

## A Pipe Dream

For all its promise, HTML WebSocket is merely a transport layer over which application-level protocols can be implemented; an “empty pipe,” in other words, not a solution. The real-world consequences of this distinction are enormous:

- **Support for all browsers on all devices** – HTML5 WebSocket is not designed to be backward compatible with legacy browsers, and some browser vendors are only now about to introduce WebSocket compatibility. Support may lag further in corporate environments and on legacy mobile devices, where users spend most of their time but have less control over the choice of browser vendor and version. Without a solution running on top of the pipe, many users will be unreachable using WebSocket.
- **Traversing proxies and firewalls** – Employers may not like it, but employees spend a lot of time playing non-work related online games on the job, and access to those players is critical to game companies locked in fiercely competitive battles for consumer mindshare. Unfortunately, HTML5 WebSocket can struggle with traversing proxies and firewalls and often does not make it through. Again, without a solution running on top of the pipe, many users will be unreachable using WebSocket.
- **Support for everything else that matters in building and growing a gaming business** – To form the basis of a sustainable business, a game or game platform needs performance and scalability, reliability and resiliency, security, and manageability. HTML5 WebSocket was not designed to address these business requirements. It provides no way to throttle, filter or conflate data flow based on bandwidth constraints or Internet congestion; no facility for load balancing, high

availability, auto-reconnect or guaranteed data delivery; no security framework; and no management or monitoring capabilities.

- **Support for gaming protocols** – Oftentimes the communication protocol used in a game or game platform is a source of competitive advantage, as is the choice of client-side technology. Developers need to be able to port their chosen protocol to the Web quickly and easily using client libraries that are specific to both the protocol and the client – all while maintaining backward compatibility, scalability, reliability, resiliency, security, and manageability.

## Worlds Collide

Everything discussed thus far applies to any TCP-based game with real-time elements where low latency and high performance are integral to the gaming experience. It would also apply to games that involve live in-game betting based on changing data, since maintaining a level playing field in those contests can literally depend on very fast, low latency transmission of game state information (odds, status of contestants) to all players.

As people become more accustomed to (and demanding of) instantaneous response in everything they do online, even a turn-based board game such as chess could benefit from the *living Web*. For instance, if an online chess game required players to move their pieces across the board by hand, rather than simply click on the destination square, and conveyed the action live, those watching might be able to glean clues about their opponents' strategies and mental state.

There is one game category that is unlikely to transition entirely to the browser anytime soon: complex massively multiplayer online games (MMOGs). But even here, living Web communication has an increasingly critical role to play. In response to the growth of the Web and mobile since the birth of MMOGs, publishers have begun to expose real-time elements of the in-game experience such as chat, virtual goods shopping and auctions, and game state updates to players who are away from the game client but accessible on a mobile device or Web browser. This collision of the in-game world and the out-of-game-but-online world can be expected to accelerate and expand as publishers expose more and more features to the Web in hot pursuit of new ways to generate revenue and capture gamers' mindshare.

As noted above, many employers don't want their employees playing non-work related games on the job. But not all games are designed primarily to entertain. Companies and other organizations are starting to use games or game logic internally, in their Intranet-based processes, to assist employees in various aspects of their work life. They are also applying game-play mechanics externally, via websites and mobile apps, to increase consumer engagement with their online brands, enhance employee recruiting, and even change core aspects of their customers' behavior. A healthcare company, for instance, might use a game-

like experience to persuade customers to exercise more, eat right, or take their medication on time. As enterprises and other organizations across industries employ games and game logic to fuel innovation and creativity, enhance internal processes, and strengthen bonds with prospects and customers, the ability to deliver a compelling, *living Web* gaming experience will literally take on bottom-line significance.

HTML5 WebSocket would be the ideal transport mechanism for all of these gaming applications – regardless of genre or whether they are played for fun, profit or corporate purposes – but for the limitations of the WebSocket standard already noted.

## Kaazing WebSocket Gateway to the Rescue!

HTML5 WebSocket is the answer to a longstanding prayer that has been on the lips of Web programmers since at least the mid-1990s, when the first rudimentary push technology appeared. But it is also a source of consternation, since it is generic standard rather than a workable business solution. As befits its name, Kaazing WebSocket Gateway (KWG), a high-performance Web communication platform, bridges that gap and delivers the full promise of HTML5 WebSocket and other new HTML5 standard communication options, bringing the *living Web* to game and game platform developers today.

KWG overcomes the limitations inherent in the HTML5 WebSocket specification by adding the structure, functionality, and suite of powerful tools needed to make it both complete and completely enterprise-ready.

- **Support for all browsers on all devices** – KWG makes the game accessible to almost everyone with a connected browser. If that browser does not support WebSocket or even HTML5, KWG emulates WebSocket – providing APIs identical to those in the WebSocket standard – *with near-native performance*. KWG automatically ensures the best possible connection environment regardless of whether Web clients or intermediate Web-enabled devices support the latest protocols – all without any plug-in requirements.
- **Traversing proxies and firewalls** – KWG can traverse any network, tunneling automatically through proxy servers and firewalls so the game reaches everyone who wants to play.
- **Support for everything else that matters in building and growing a gaming business** – KWG gives gaming *businesses* a complete, enterprise-grade solution:
  - **Performance and scalability** – KWG easily scales to enormous numbers of concurrent users.

- **Reliability and resiliency** – KWG keeps the game going by guaranteeing message delivery and re-establishing connections to disconnected clients.
- **Security** – KWG lets developers quickly build enterprise-strength security features into their Web-based applications to protect data, enable seamless user authentication, and help prevent malicious attacks on back-end servers. These features include single sign-on, credential caching, credentials injection, protocol validation, and full support for HTTP authorization and authentication.
- **Manageability** – KWG extensions allow administrators to manage and monitor the internal state of the gateway. KWG also supports industry standard management protocols and can plug into industry standard management solutions, allowing IT teams to use existing tools to oversee both LAN-based and browser-attached users.
- **Cost, complexity and productivity** – KWG allows developers to deploy a single code base across devices and browsers, eliminates the need for application server software, and reduces hardware requirements. Each of these factors contributes to a large reduction in overall cost and complexity and a large increase in developer productivity.
- **Support for gaming protocols** – KWG ships with pre-built libraries for many popular client technologies such as JavaScript, Java, Adobe Flash/Flex, and Microsoft Silverlight with client-side support for messaging and communication services, including JMS, XMPP, and AMQP. More importantly for many game developers, KWG can be extended **easily and quickly** to support other standard and custom protocols and protocol APIs. With KWG, developers can run their own custom protocols over the Web and still benefit from a full suite of robust enterprise features and services.

## With Kaazing WebSocket Gateway, Gaming Is Serious Business

KWG delivers on the promise of HTML5 WebSocket to permit online games that are better, faster, and cheaper. With KWG, developers of Web-based games and game platforms can achieve stunning competitive advantage by

- making their products truly ubiquitous and accessible
- slashing development and maintenance costs
- boosting developer productivity
- ensuring extreme yet cost-effective scalability
- fortifying their business with the most robust enterprise-class services obtainable
- renouncing the legacy Web in favor of the *living Web*.

Although they may not be moving entirely to the Web anytime soon, client-based MMOGs that wish to expose select features such as chat, shopping, auctions, or game state updates to the Web can also leverage Kaazing's high-performance Web communication platform. KWG permits messages to travel back and forth between server and client in their native, in-game format. So developers of these virtual worlds can use KWG to extend elements of the worlds' internal infrastructure out to mobile and Web clients. By obviating the need to build a separate external infrastructure or a bridge to interface and translate between the in-game world and the HTTP world, this *universal architecture* can save developers cost, complexity, latency, and network overhead, while giving them ultimate flexibility to expose additional game features at will in response to changes in user behavior or expectations, advances in client technology, or changes in business objectives.

## Easy Does It

Kaazing WebSocket Gateway is easy to evaluate and lightweight to deploy. You can download and use the full KWG server with no limitations on functionality and no trial period. And KWG client libraries are so lightweight that many of the images in your UI will be larger than the installed library.

Kaazing. It's how the game is played.

Revised August 2011