
TM
KAAZING

Kaazing Gateway: An Open Source HTML 5 Websocket Server

KAAZING

Speaker

- Jonas Jacobi
- Co-Founder: Kaazing
- Co-Author: Pro JSF and Ajax, Apress

KAAZING

Agenda

- Real-Time Web? Why Do I Care?
- Scalability and Performance Concerns
- Comet
- Is This It?

KAAZING

Defining Real-Time Web

Web Applications Typically Not Real-Time

KAAZING

Defining Real-Time Web

- Web Clients Receive Server Updates
  - Server-initiated communication
- End-Users Receive Updates Simultaneously
  - Collaboration

KAAZING

Defining Real-Time Web

Or, is it just nearly, nearly real-time?

Maybe we should rename it to something else – Event-Driven Web?

KAAZING

Ajax (XHR)

- Updates Limited To Preset Interval
  - Message buffering increases memory usage
  - Near real-time updates achieved with shorter intervals
- Shorter Updates Cause
  - Increased network traffic
  - Higher frequency connection setup & teardown

Push Technology History

- Push technology has been around for a while:
  - Pushlets (2002)
  - Bang Networks (early adopter)
- Previous attempts failed, because:
  - Scalability Limitations (Cost etc…)
  - Not general purpose
  - No standard

Push Technology

- Server-Initiated Message Delivery
  - Clients are listening
  - Clients behind firewalls
- Techniques such as Comet/Reverse Ajax
- Delays Completion of HTTP Response
- Generally Implemented in JS

KAAZING

Long Polling and Streaming

- Current Comet implementations center around two major areas:
  - Long Polling
  - Streaming

KAAZING

Long Polling

- Also known as asynchronous-polling
- Request open for a set period.
- HTTP headers often account for more than half of the network traffic.

KAAZING

Long Polling HTTP Request

From client (browser) to server:

```
GET /long-polling HTTP/1.1\r\n
Host: www.kaazing.com\r\n
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9)
   Gecko/2008061017 Firefox/3.0\r\n
Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
   \r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
\r\n
```

KAAZING

Long Polling HTTP Response

From server to client (browser):

```
Date: Tue, 16 Aug 2008 00:00:00 GMT\r\n
Server: Apache/2.2.9 (Unix)\r\n
Content-Type: text/plain\r\n
Content-Length: 12\r\n
\r\n
Hello, world
```

KAAZING

HTTP Streaming

- Persistent HTTP Connection
  - Pending POST
- Minimizes Latency
- Reduction in Network Traffic
- Optimizes Connection Setup & Tear-Down
  - Keep-alive
  - Security

KAAZING

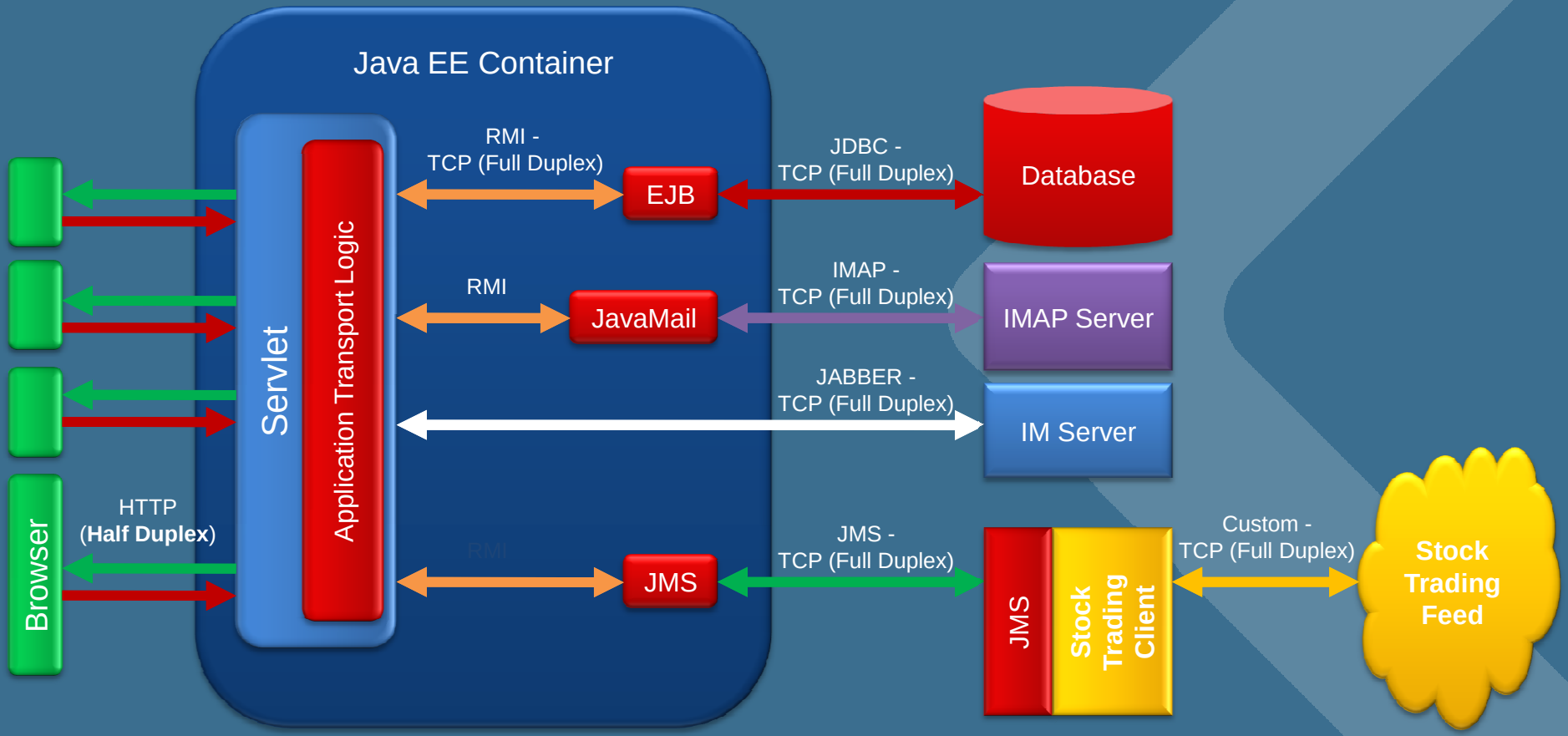
Concurrency

- Many Concurrent Synchronous Requests
- Standard Java EE Containers
  - Designed For Short-Lived Request/Response
  - One open socket per thread

Solutions: Vertical Scalability

- Asynchronous Request Processing (ARP)
  - Java NIO
  - Twisted (Python)
  - POE (Perl)
- Decouples Connections from Threads
  - Jetty Continuations
  - Grizzly CometEngine
  - Tomcat6 CometProcessor

Today's Architecture

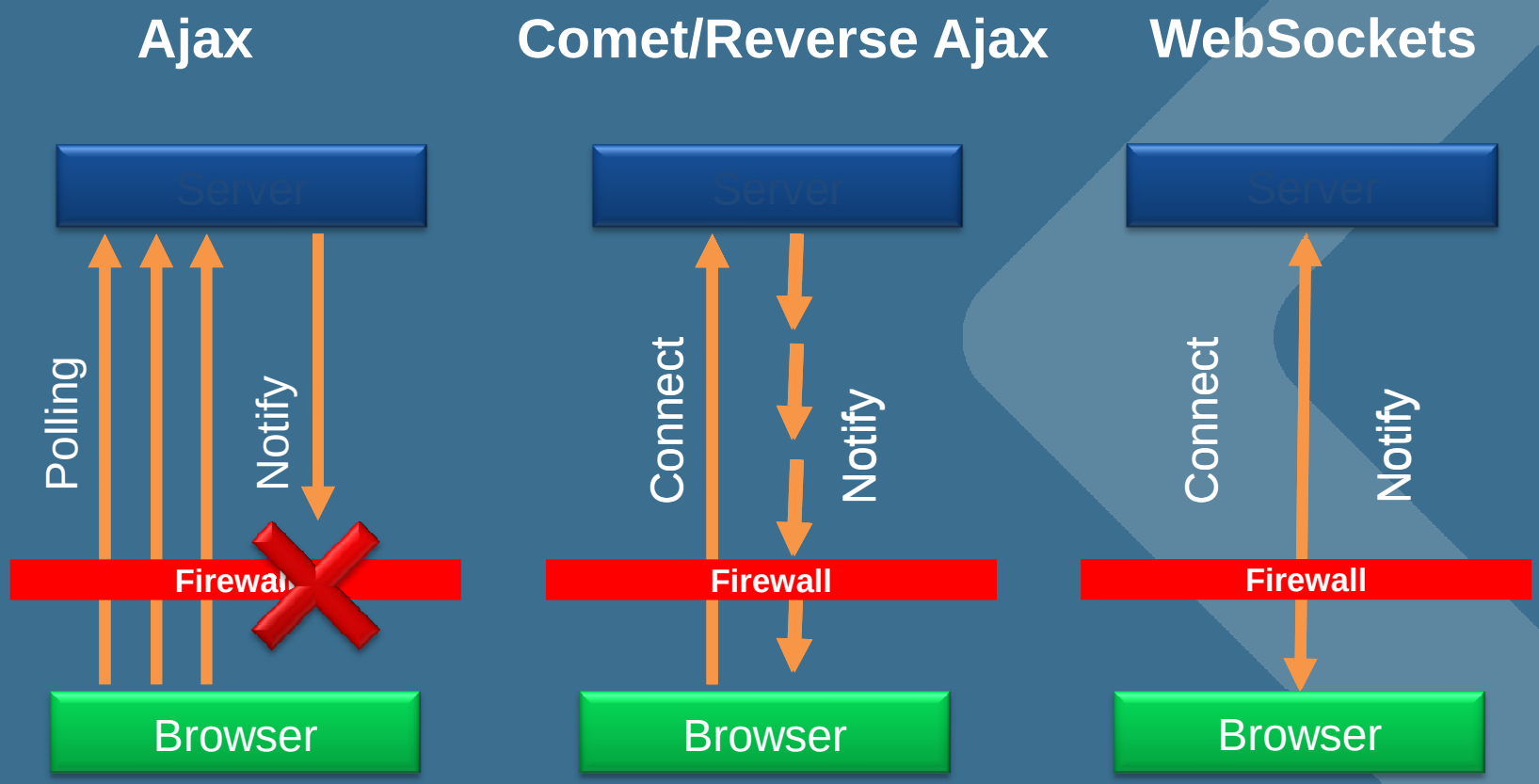


KAAZING

What's Missing?

- Not a standard
- No true bi-directional communication
- No guaranteed message delivery
- Complex middle-tier architecture
  - Adds unnecessary latency

Evolving the Web



KAAZING

HTML 5 WebSockets

- The Communication section:
  - WebSockets
  - Server-sent events
- Not New; TCPConnection API and protocol were initially drafted over two years ago
- HTML 5 – Final draft by 2022?!

KAAZING

Server-Sent Events

- Standardizes **and** formalizes how a continuous stream of data can be sent from a server to a browser
- Introduces eventsource—a new DOM element

Server-Sent Events

- Connects to a server URL to receive an event stream:

```
<eventsource src=
  "http://stocks.kaazing.com"
  onmessage="alert(event.data)">
```

KAAZING

Server Sent-Events

- Server can add the ID header so that clients add a Last-Event-ID header
- Used to guarantee message delivery
- Server specify an optional retry header as part of an event in the event stream

WebSockets

- Defines full-duplex communications
  - Operates over a single socket
- Traverses firewalls and routers seamlessly
- Allows authorized cross-domain communication
- Integrates with:
  - Cookie-based authentication
  - Existing HTTP load balancers

KAAZING

WebSockets

- Connection established by upgrading from the HTTP protocol to the WebSocket protocol
- WebSocket data frames can be sent back and forth between the client and the server in full-duplex mode

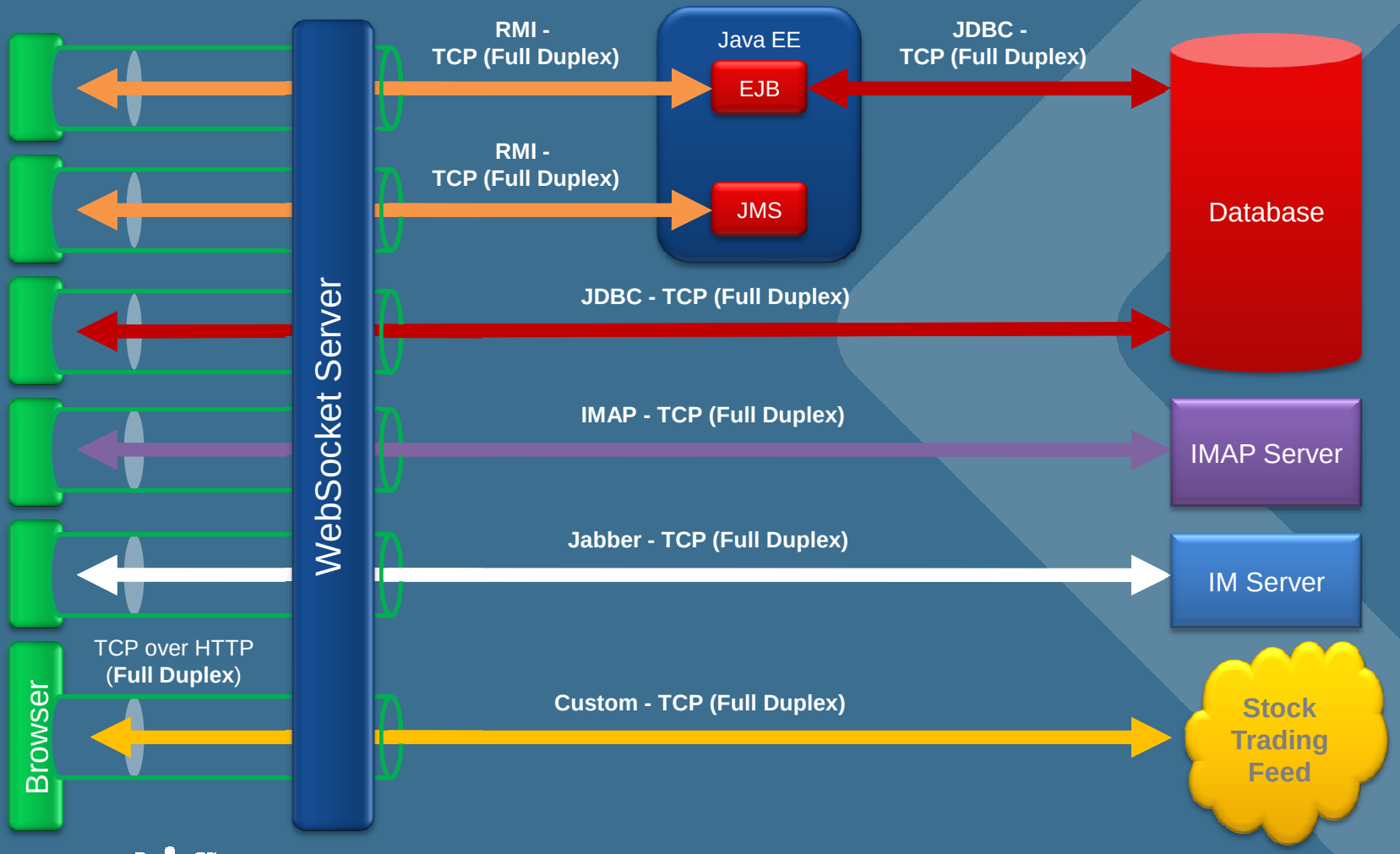
WebSockets

- Supports a diverse set of clients
- Cannot deliver raw binary data to JavaScript
  - Binary data is ignored if the client is JavaScript
- Enables direct communication with backend systems

WebSockets

- Detects presence of proxy servers
- A tunnel is established by issuing an HTTP CONNECT statement
- Secure Web sockets over SSL can leverage the same HTTP CONNECT technique

Simplified Architecture



KAAZING

WebSockets

- Creating a WebSocket instance:

```
var myWebSocket = new WebSocket
("ws://www.websocket.org");
```

KAAZING

WebSockets

- Associating listeners:

```
myWebSocket.onopen = function(evt) {
  alert(“Connection open ...”); };
myWebSocket.onmessage = function(evt) {
  alert( “Received Message:  ”  +
  evt.data); };
myWebSocket.onclose = function(evt) {
  alert(“Connection closed.”); };
```

KAAZING

WebSockets

- Sending messages:

```
myWebSocket.postMessage(“Hello Web
  Socket! Goodbye Comet!”);
myWebSocket.disconnect();
```

WebSocket Servers

- Kaazing Gateway
  - Open source
  - Standards compliant
  - Binary and text support
  - Production Release Sept 29th, 2008
- Orbited
  - Python open source project

KAAZING

Kaazing Gateway

- Enables full-duplex communication to any TCP-based back-end service:
- JMS
- Jabber
- Stomp
- etc…

KAAZING

Kaazing Gateway

- Based on SEDA (Staged Event-Driven Architecture)
  - Leverages Java New I/O (NIO)
- Simplifies architecture
  - Low Latency
- Client-side emulation of the standard if no browser support is available
- Full-duplex binary and text communication

KAAZING

Kaazing Gateway

- Scalability?
- ENOUGH 

KAAZING

Summary

- Event-Driven Solutions Are Required For a Multi-User Web
- WebSockets and SSE standardize Comet
- Available now!

KAAZING

Q&A

www.kaazing.com

KAAZING