

야구9단에 적용된 리얼타임웹 통신 소개 (RTCS)

이용현

NHN

2011년 11월 7일

목차

- 리얼타임웹을 위한 기술적 배경
 - ✓ HTTP
 - ✓ Servlet Container
- 야구9단에 적용된 리얼타임웹
 - ✓ 소개
 - ✓ 아키텍처
 - ✓ 동작 설명
- RTCS 소개
 - ✓ 기능
 - ✓ 컴포넌트
 - ✓ 적용 아키텍처
- RTCS 구현 경험
- 리얼타임웹의 방향

리얼타임웹을 위한 기술적 배경

리얼타임웹 이란?

리얼타임웹

- ❖ 다른 사용자에게 의해 갱신된 정보를 별도의 요청 없이 바로 받아 볼 수 있는 웹
(http://en.wikipedia.org/wiki/Real-time_web)

응용 서비스

- ❖ Facebook 채팅
- ❖ Google 파이낸스
- ❖ 웹 온라인 게임

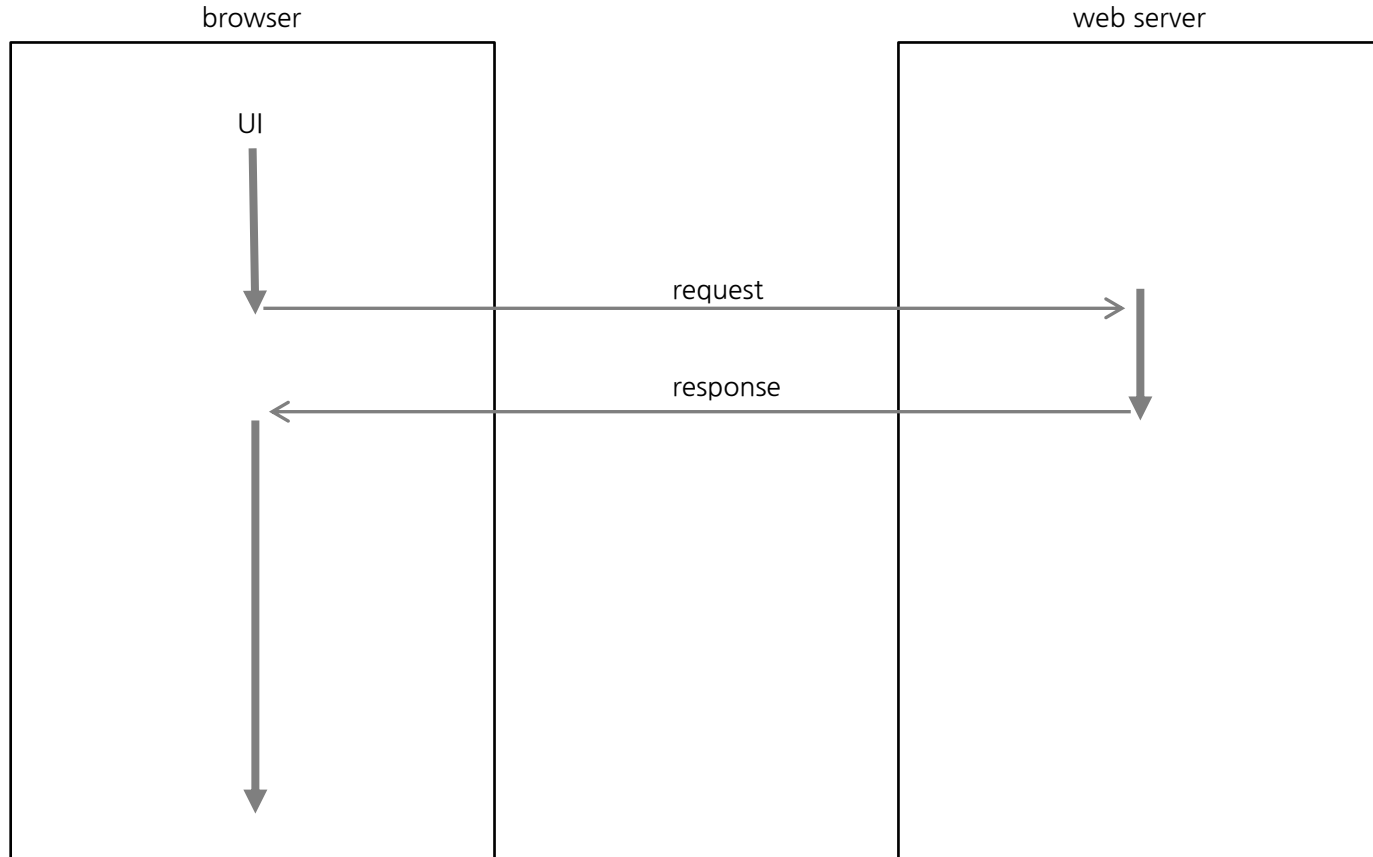


HTTP Transaction Model

Plain Old	Request/Response	정적인 HTML 페이지 요청 응답
	Ajax	백그라운드 동적 UI 요청 응답
	Polling	주기적인 데이터 요청 응답
Comet	Long Polling	요청 후 연결을 유지하다가, 서버에서 이벤트가 있으면 응답 (한번의 요청으로 한번의 이벤트 응답)
	Streaming	요청 후 연결을 유지하고, 서버에서 이벤트가 있을 때 마다 chunked response로 응답 (한번의 요청으로 복수개의 이벤트 응답, XHR/iframe/jsonp등의 방식)
HTML5	Server-Sent Events	서버에 연결 후 push notification을 DOM 이벤트 형식으로 받음
	WebSocket	양방향 통신

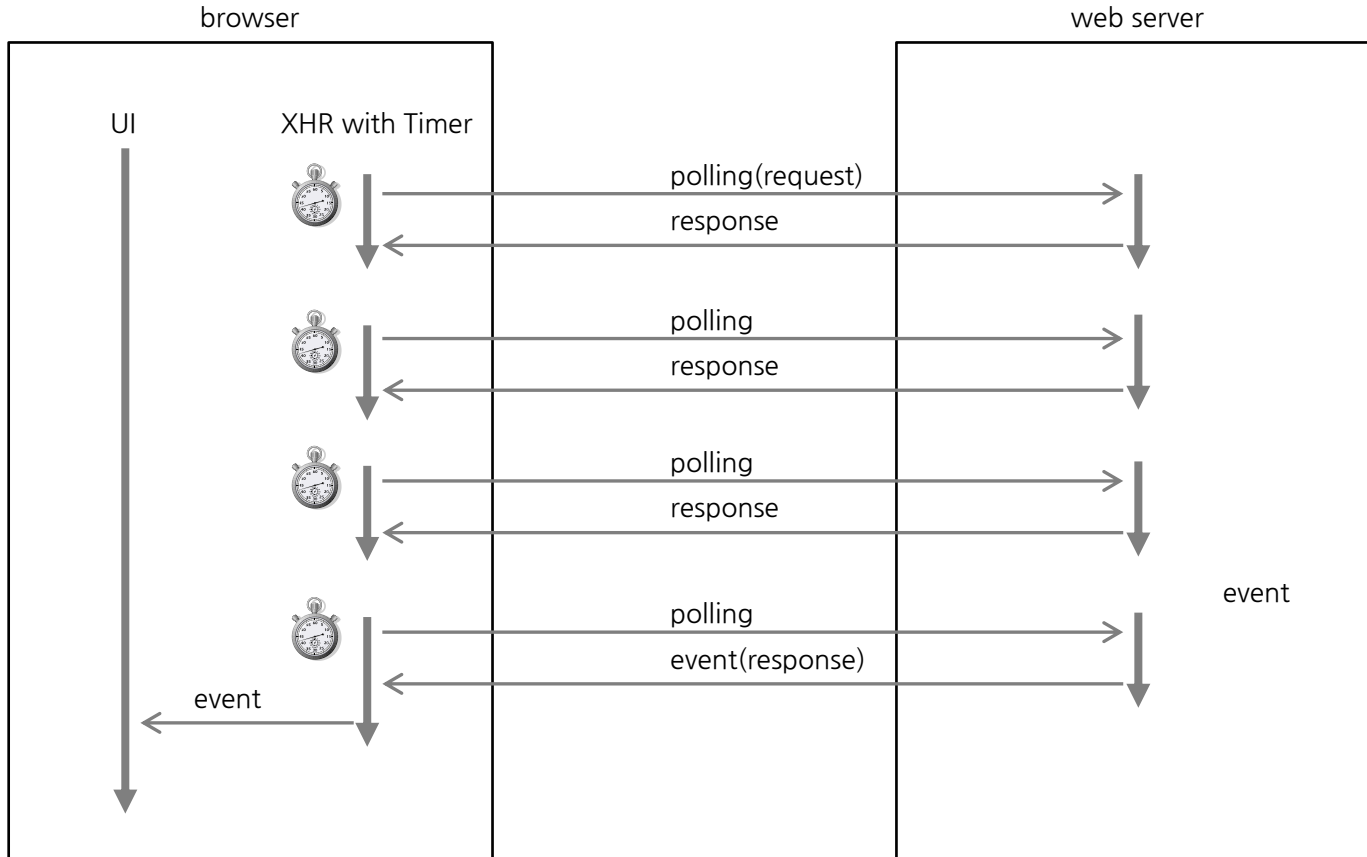
HTTP Transaction Model

Request/Response



HTTP Transaction Model

Polling



HTTP Transaction Model

Comet

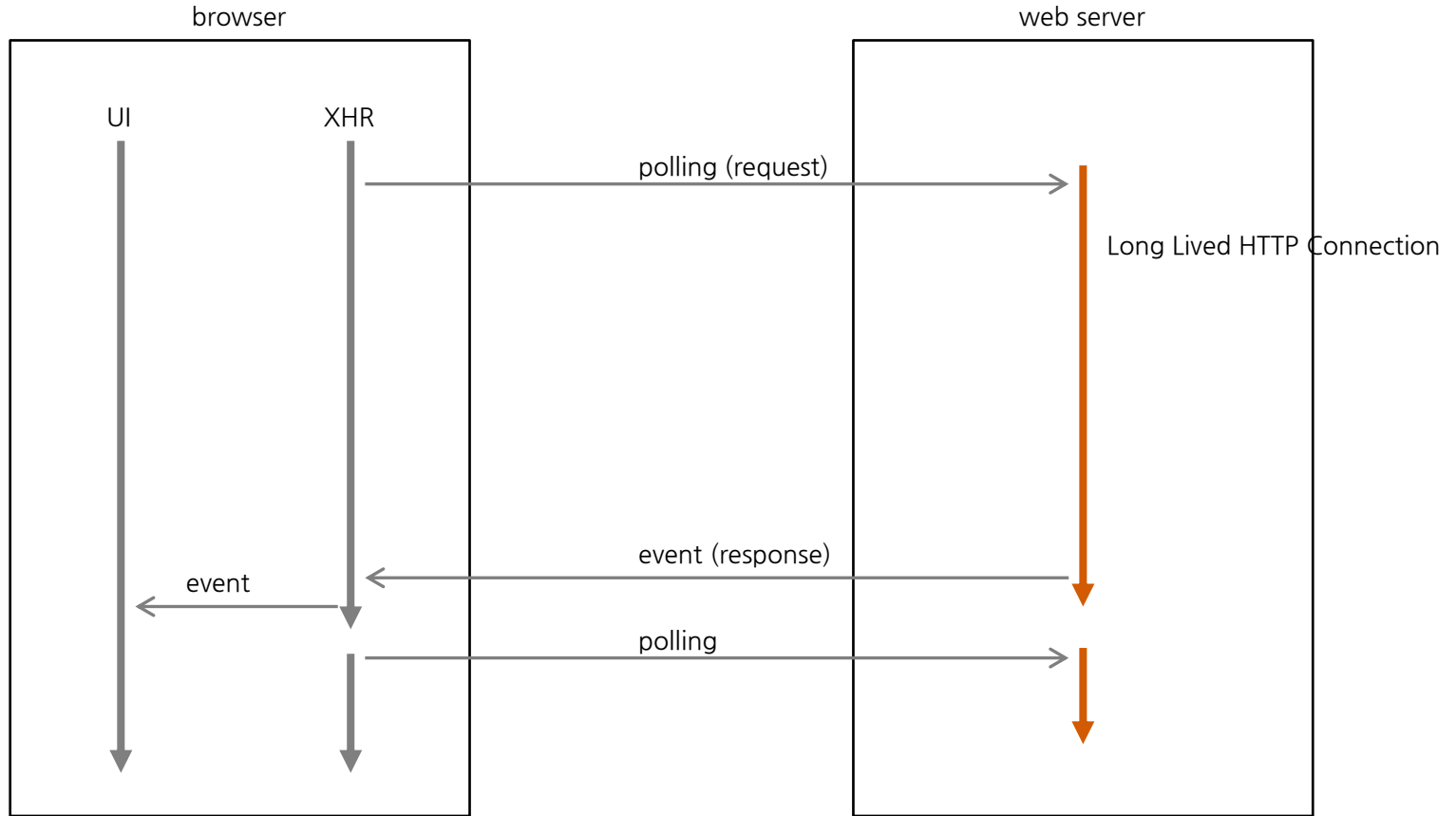


- ❖ Ajax는 1인을 위한 웹 UI 응답 속도를 개선
- ❖ Comet은 다중 사용자에게 동일한 데이터를 리얼타임에 보여 줄 수 있게 웹을 개선
- ❖ 기술적 어려움
 - 브라우저 별 다양한 동작의 차이가 존재
 - Long Lived HTTP Connection을 위한 서버 기술이 필요하다.

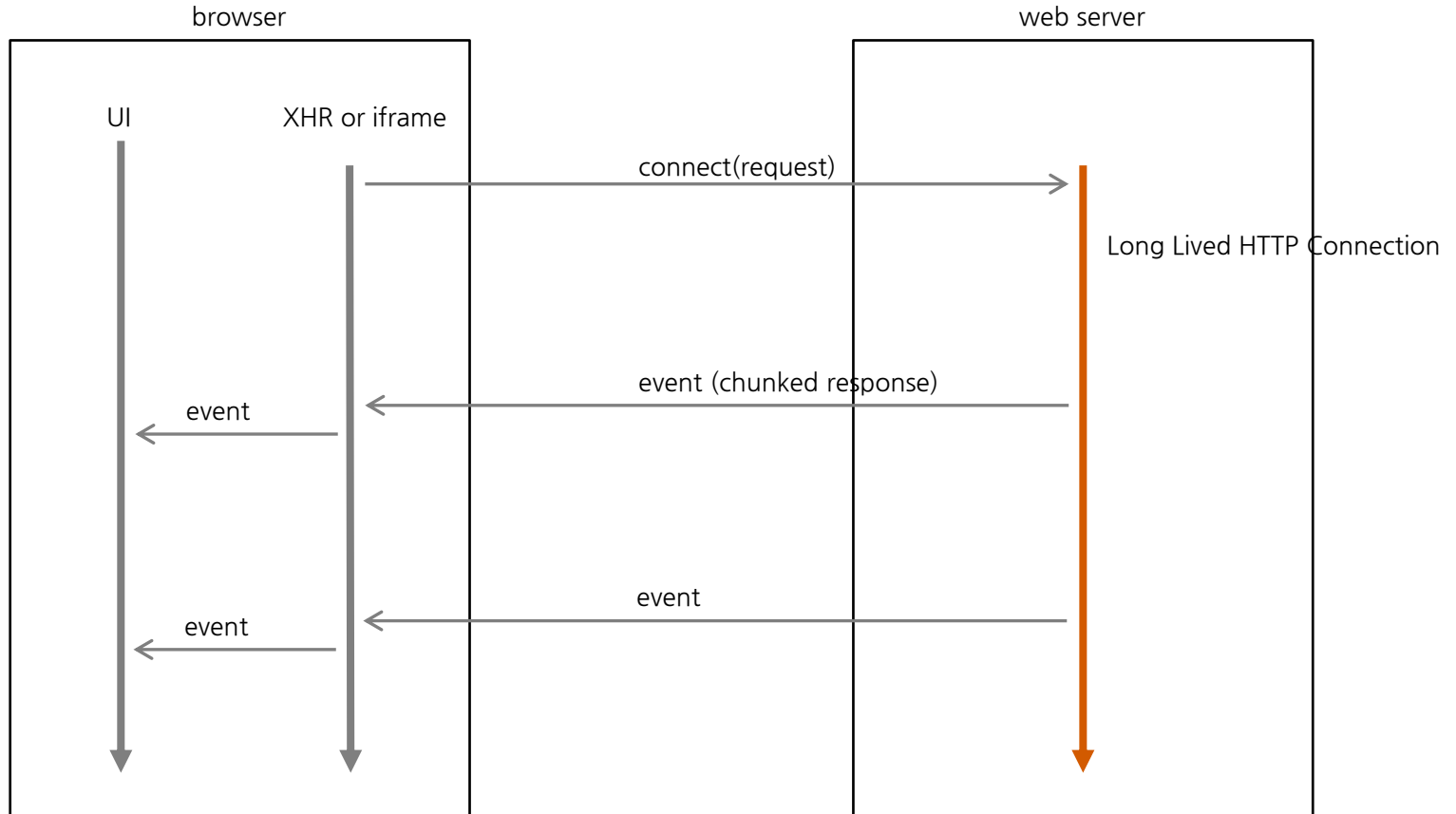
<http://infrequently.org/2006/03/comet-low-latency-data-for-the-browser/>

HTTP Transaction Model (Comet)

Long Polling



HTTP Transaction Model (Comet) Streaming



<http://blogs.msdn.com/b/ieinternals/archive/2010/04/06/comet-streaming-in-internet-explorer-with-xmlhttprequest-and-xdomainrequest.aspx>

HTTP Transaction Model

HTML5

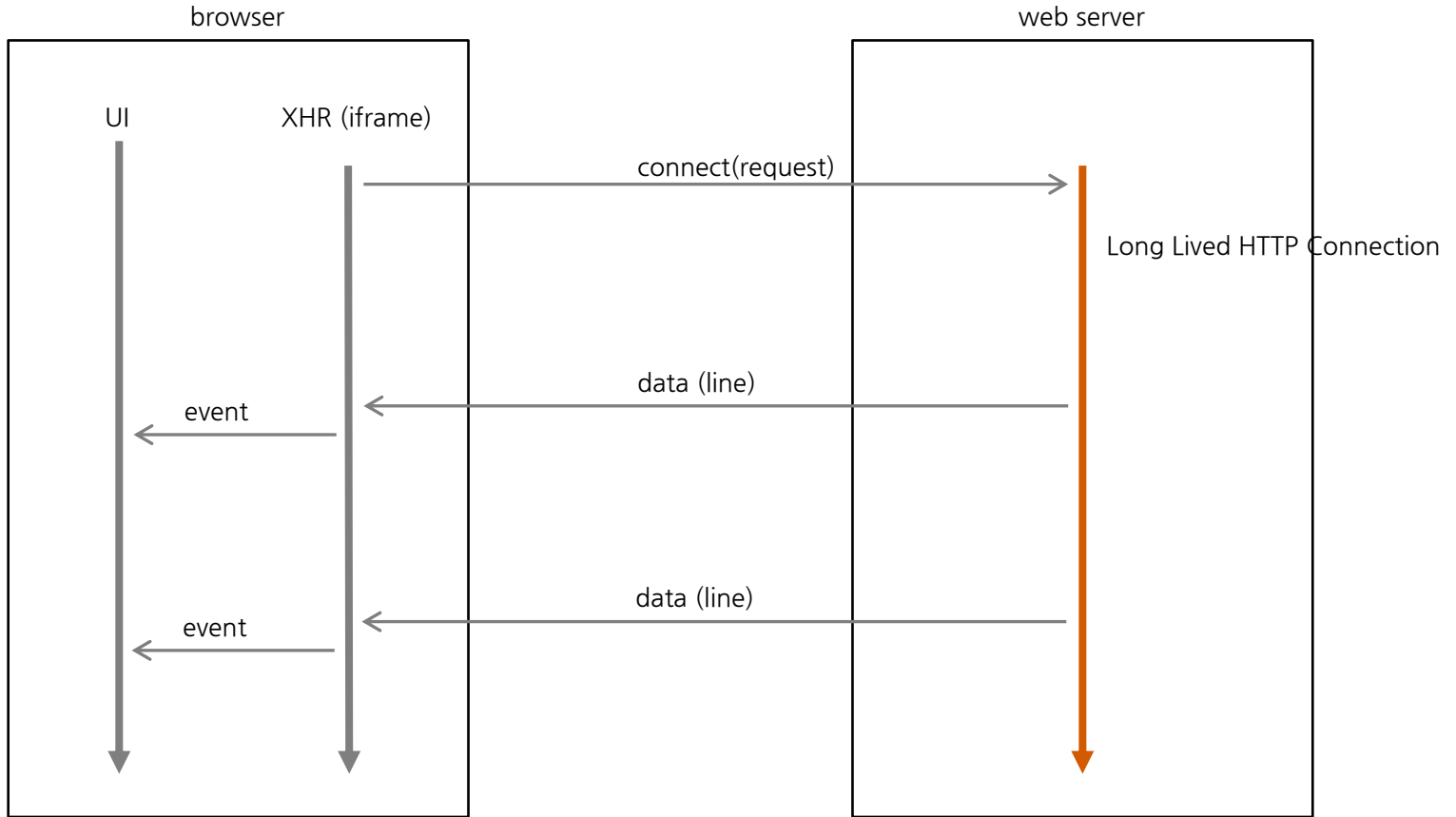


- ❖ Hyper Text Markup Language의 5번째 버전
<http://dev.w3.org/html5/spec/Overview.html>
- ❖ 광의로 Web Application을 위한 WebSocket등의 표준도 포함

HTTP Transaction Model (HTML5)

Server-Sent Events

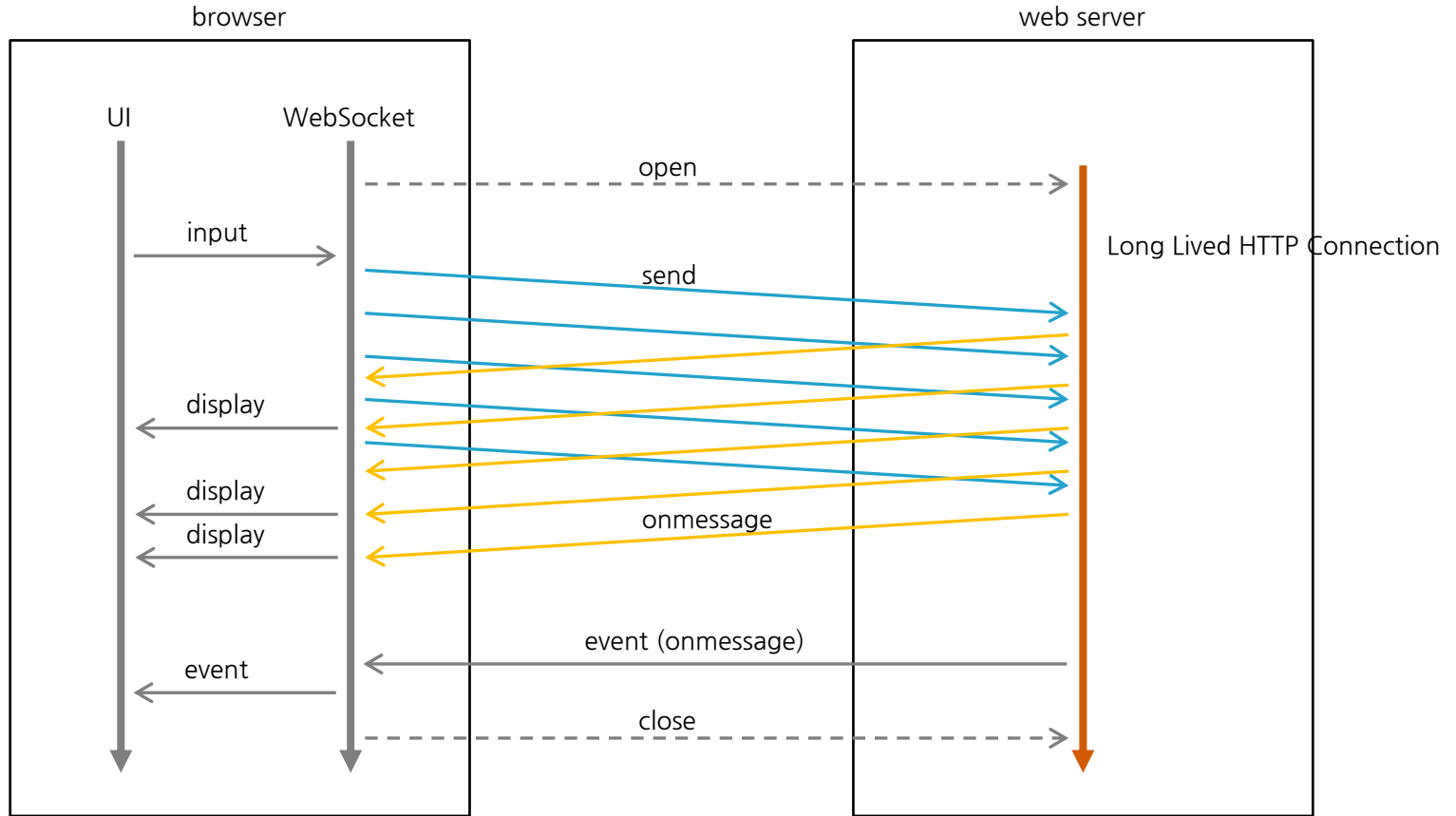
```
var source = new EventSource('updates.cgi');  
source.onmessage = function (event) { alert(event.data); };
```



<http://dev.w3.org/html5/eventsource/>

HTTP Transaction Model (HTML5)

WebSocket



API : <http://dev.w3.org/html5/websockets/>

Protocol : <http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-16> (Proposed Standard)

HTTP Transaction Model

브라우저 별 지원

참조 : <http://caniuse.com/>

브라우저	Long polling	Streaming	Server-Sent Event	WebSocket
IE	O	O	X	X
FireFox	O	O	O (6.0)	O (4.0)
Chrome	O	O	O (6.0)	O (4.0)
Safari	O	O	O (5.0)	O (5.0)
Opera	O	X	O (9.0)	O (11.0)
iOS Safari	O	O	O (4.0)	O (4.2)
Android	O	X	X	X

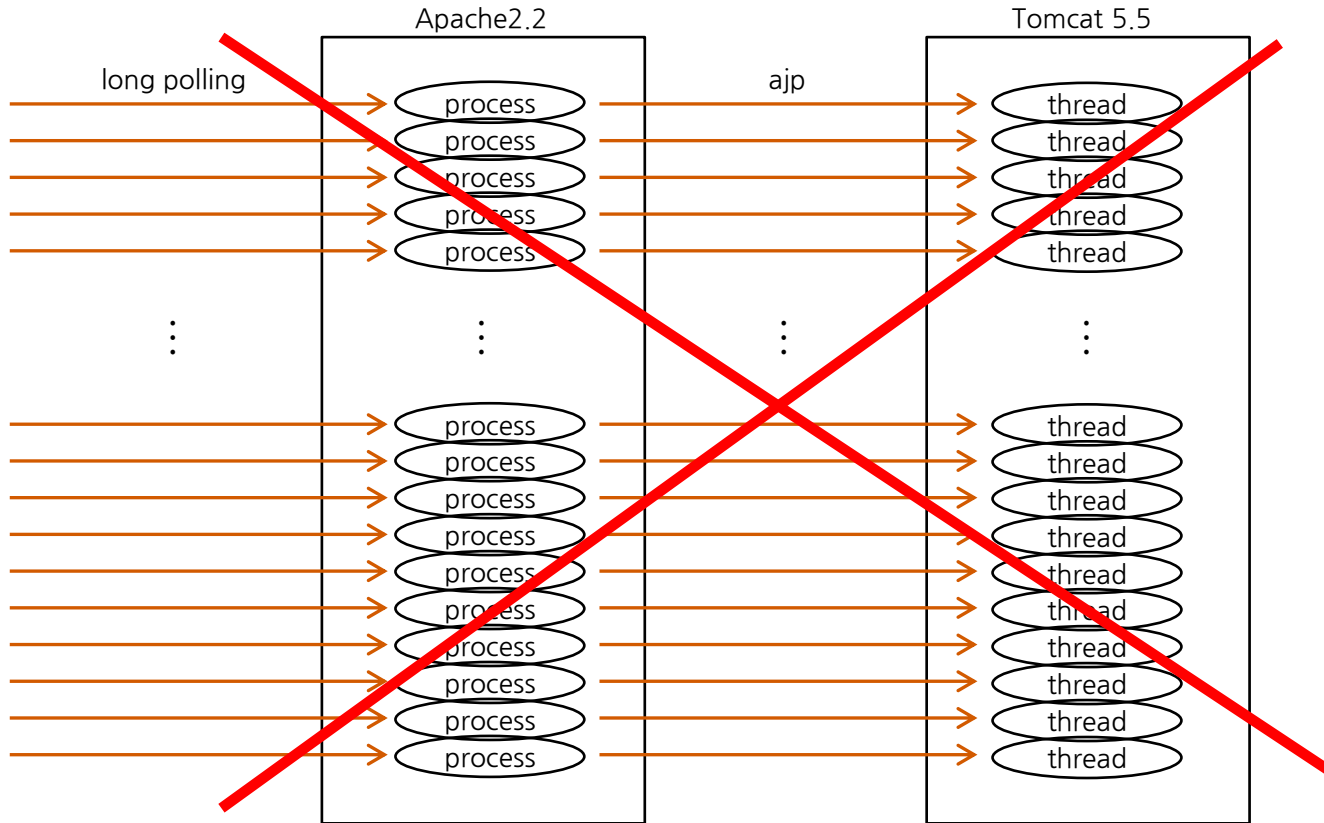
Servlet Container

- ❖ Long Polling, Streaming, Server-sent Events, WebSocket
- ❖ 모두 Long Lived HTTP Connection이 필요하다.

Servlet Container

Apache+Tomcat : thread per connection

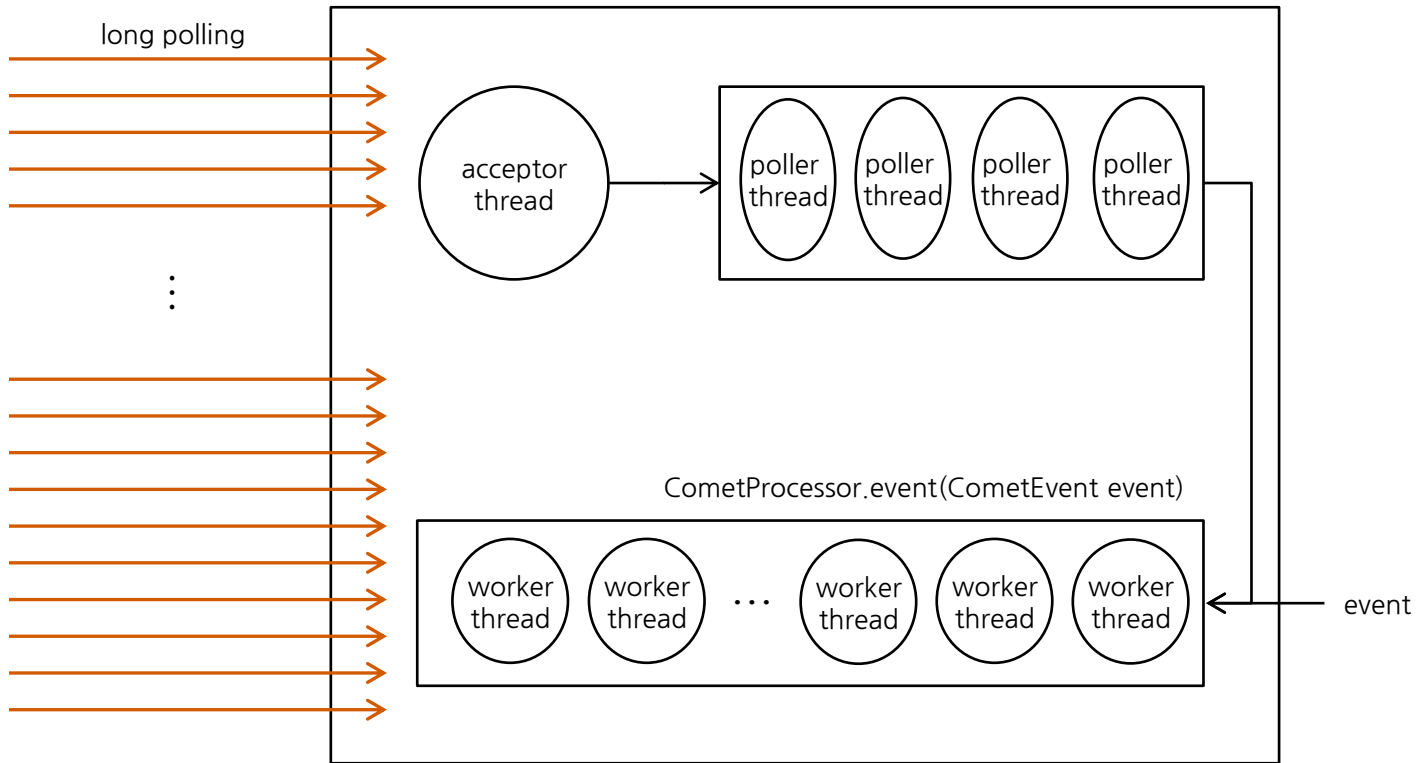
Apache2.2 + Tomcat 5.5



Servlet Container

Tomcat : NIO Connector

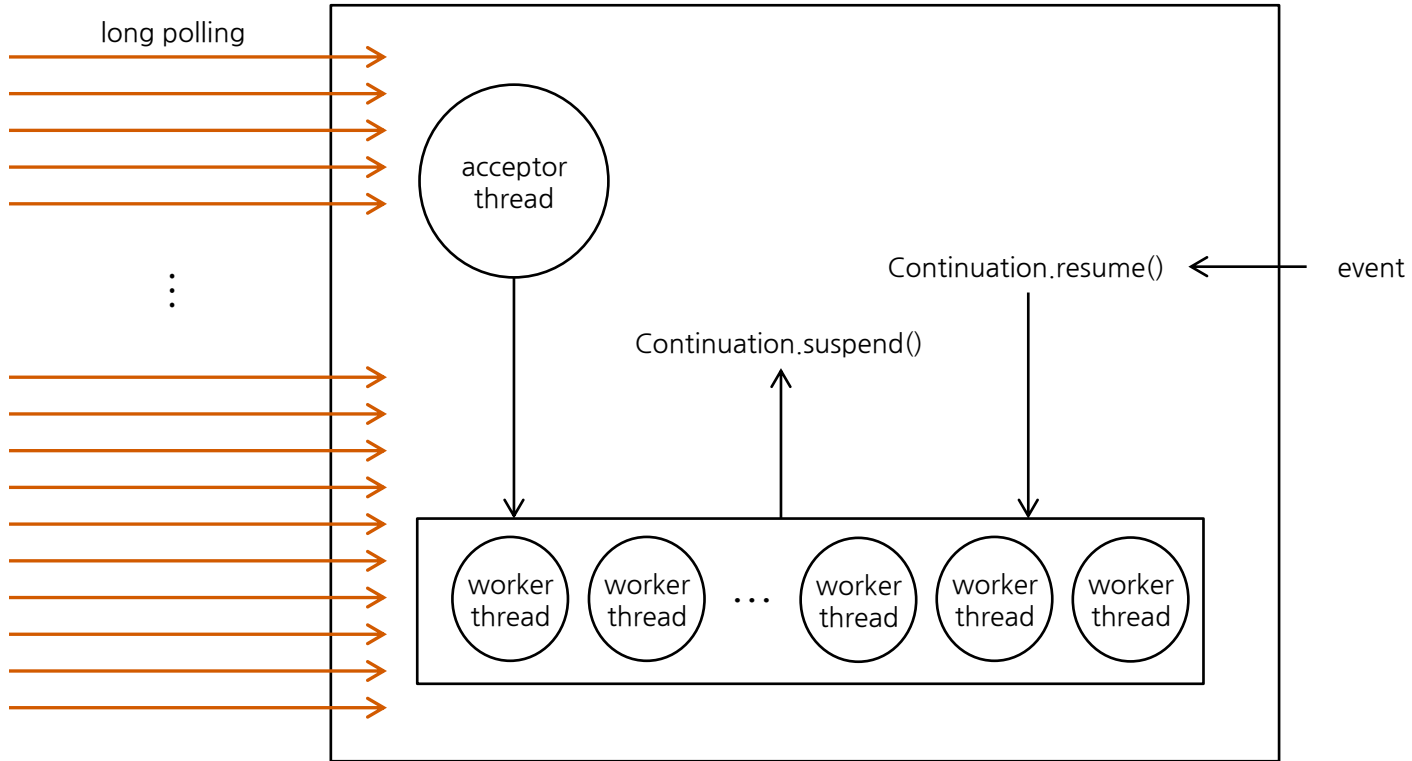
Tomcat 6 + NIO Connector (CometProcessor, CometEvent)



Servlet Container

Jetty : Continuation

Jetty7 + Continuation



야구9단에 적용된 리얼타임웹

야구9단 소개



야구9단 소개



야구9단 소개



야구9단 소개



100만 구단주 돌파!

- ❖ 멀티플랫폼을 위해 **순수 웹 기술**로만 제작
- ❖ 채팅, 실시간 개입을 위한 **리얼타임웹 기술** 필요
- ❖ 자세한 개발 히스토리는 DEVIEW 자료 참조
“야구9단” 어떻게 만들어졌을까요? - 웹 개발자들의 게임제작 도전기
http://www.deview.kr/2011/track/D_06.html

야구9단 소개

데모 : 채팅, 실시간개입

나만의 프로야구, 야구9단 :: 야구9단

NAVER 스포츠

월드라이브 [로그아웃] 인물검색

월드라이브 야구 Naver 스포츠

백태 2군 (AI) vs 골렘이 (GOM)

1 2 3 4 5 6 7 8 9 10 11 12 R H E B

AI 0 0 - - - - - 0 1 0 1

GOM 0 - - - - - 0 1 0 0

타순	이름	수비	상태	타율	출루	타점	도루	오늘
1	91 문승훈	중		0.069	1	1	0	0/0
2	96 박병호	우		0.217	0	0	0	0/1
3	00김광진	2		0.200	0	1	1	0/1
4	94김재덕	1		0.234	0	1	0	1/1
5	99 박연호	좌		0.240	0	4	0	0/0
6	86 장재근	포		0.190	2	3	0	0/0
7	82 최영호	3		0.156	0	1	1	0/0
8	89 장호익	지		0.147	0	2	0	0/0
9	99 김태룡	유		0.189	1	1	0	0/0

신발 90사동수

경기센터

백태 2군의 경기입니다.
5번 타자 박연호
심호흡을 갖게 하고 타석에 들어섭니다.
2-3 볼 카운트상황에서 박동수 위인드업.
던졌습니다.
쳤습니다!
좌중간으로 날아가는 볼.
좌중간 안타

생중계 영역

실시간 개입

90 박동수 (레벨3)

타율: 2.19 | 이닝: 49 1/3
승: 3 | 패: 2
세: 0 | 실점: 19

구속: 47 구위: 42 변화구: 47 제구력: 42
제책: 44 간담: 81 정신력: 48 수비: 55

오늘 이닝 1 투구수 17 실점 0 피안타 1

99 박연호 (레벨3)

타율: 0.240 | 도루: 0
출루: 0 | 출루: 0.321
타점: 4 | 장타: 0.240

정확성: 46 파워: 39 선구안: 42 주력: 35

오늘 이닝 5 투구수 17 실점 0 피안타 1

이름: 김태룡 | 코치: 김태룡

내야: 전진수비 | 포수: 박동수 | 투수: 장재근 | 좌중간: 박동수 | 우중간: 김태룡

외야: 전진수비 | 좌중간: 박동수 | 투수: 장재근 | 좌중간: 박동수 | 투수: 장재근

필수하루스런?

구단정보, 선수정보를 한번에 확인하고
홈런타자에 발도장 제어까지!

친구 관리하루스런? > | 미동

선수 정보 조회 >

★ ★ OBT 서비스오픈 기념 이벤트 ★ ★

야구9단에 올인하고
골든볼 받자!

구단: 포지션: 선수가단: 문학구장 (현관)

링크하기

이크블러드: http://me2day.net/dragon8325친추롬요
장관: 아오... 신비이나 삼수나..
소스예제: 음..
장관: 박병호 미록 유학 보내리며
장관: 혼이 있어야 되는데
장관: 그러려면 칭민이가 팔려야 하고 ;< ;<
고혁이: ghr1 dscn dEjrgpgkskdy?
고혁이: 혹시 친추 어떻게 하세요?
장관: 예?

채팅 영역

몇달일 아오시 채팅창 내 구단주영을 클릭하거나, /구단주영 메시지를 입력하세요.

입력

새소식 | 이벤트 | 단말기

- (원로) 04월 07일 임시정경 원로 ..
- (안내) 04월 07일 임시정경 안내 ..
- [안내] 04월 06일 주요 패치 및 행..
- [안내] 04월 05일 주요 패치 안내 ..

도움말 1:1문의

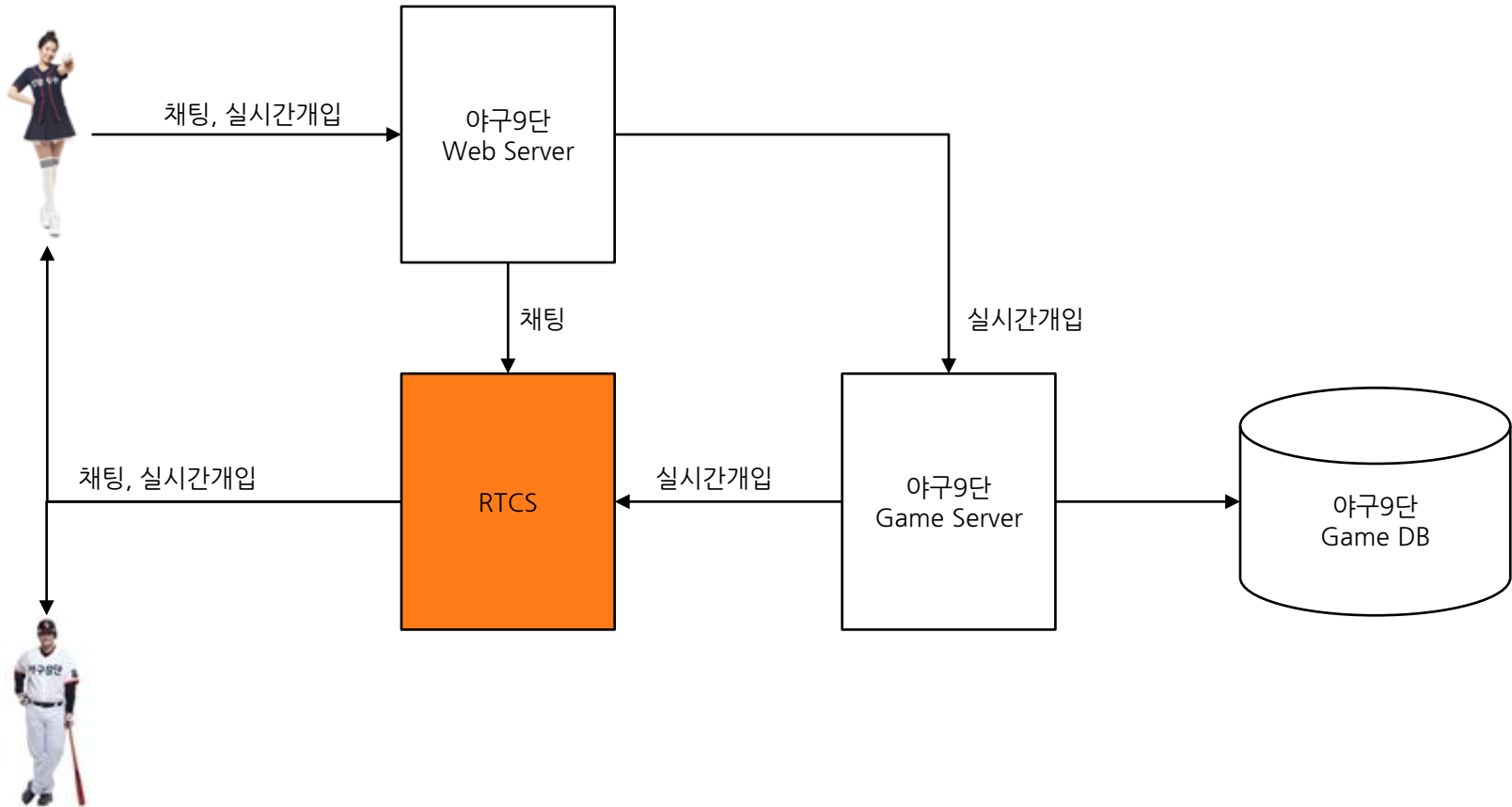
가이드를 이용하시고
게임 야니를 받으세요!

무엇이 > | 검색

게임 | 모바일 | SNS | FAQ

야구9단 소개

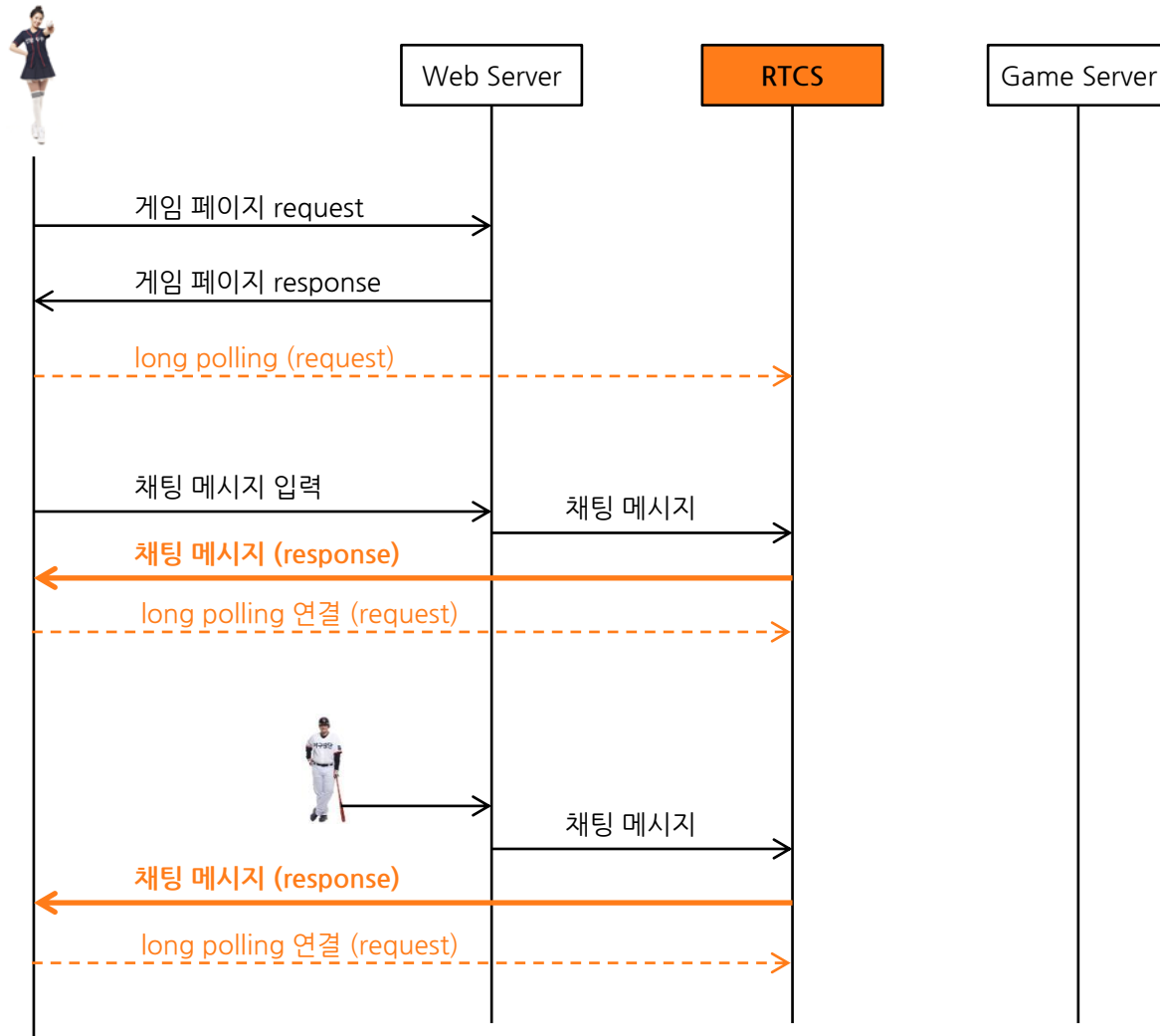
10KM 상공에서 본 아키텍처 (채팅, 실시간게임 중심)



리얼타임웹을 위해 **Long Polling** 사용

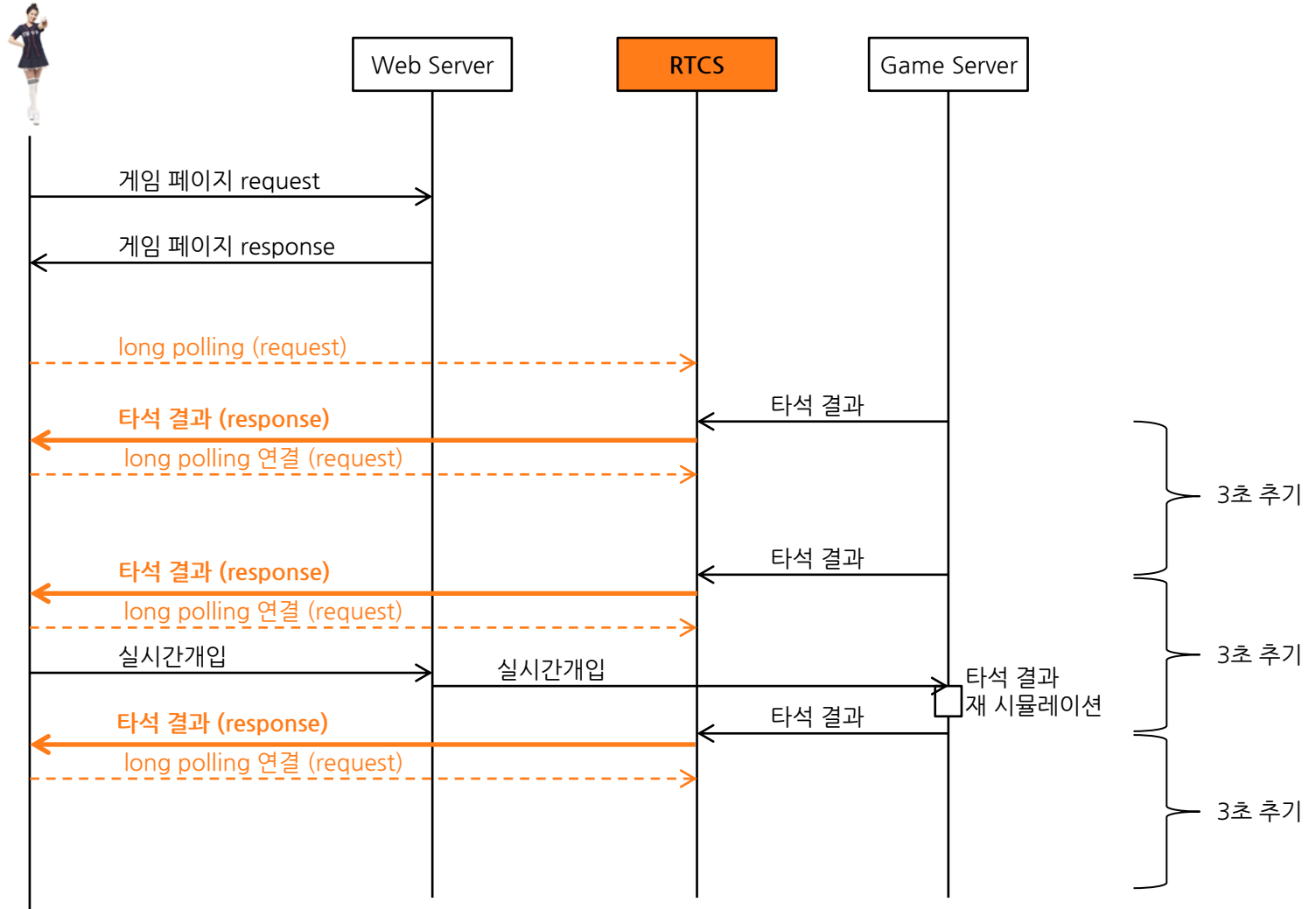
야구9단 소개

시퀀스 - 채팅



야구9단 소개

시퀀스 - 실시간개입



RTCS 소개

Real Time Communication System

(NHN에서 개발한 리얼타임웹 솔루션)

주요 기능

- ❖ 브라우저에서 양방향 통신 지원 (Long Polling, Streaming, Server-sent Events, Ajax)
- ❖ PC 5대 브라우저 및 iOS, Android 지원
- ❖ 사용자 세션 연결 보장
- ❖ 메시지 순서 및 전송 보장
- ❖ 채널을 통한 메시지 멀티캐스트 지원
- ❖ 대용량 서비스를 위한 수평적 확장 지원

온라인게임

채팅

리얼타임웹 서비스

실시간 증권 정보

야구 문자 중계

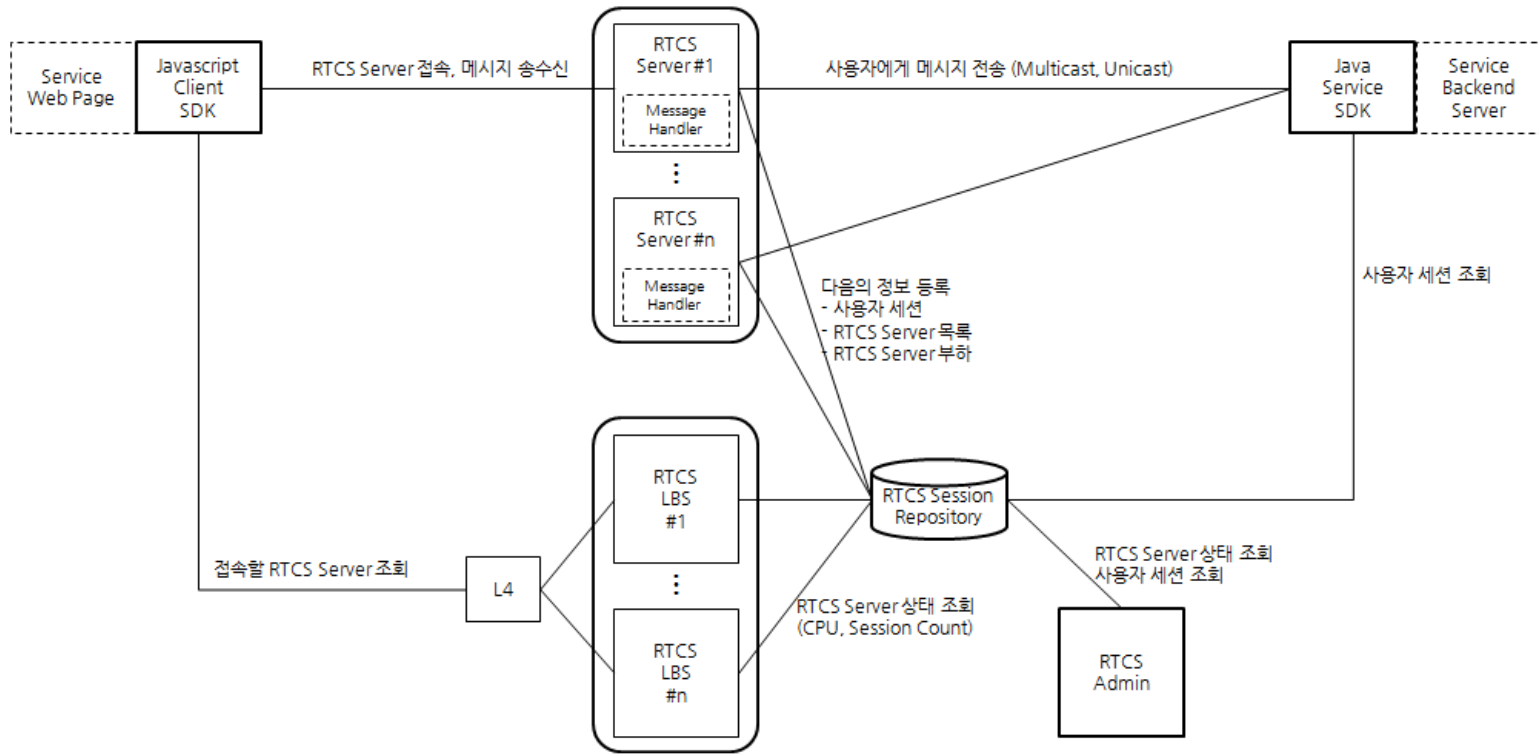
HTTP Transaction

- ❖ Client to Server : Ajax
- ❖ Server to Client : Long Polling, Streaming(XHR,iframe), Server-sent Event

Servlet Container

- ❖ Tomcat6 NIO Connector
- ❖ Jetty7 Continuation

RTCS Component



Javascript Client SDK	브라우저에서 RTCS Server와 메시지를 송수신 하기 위한 SDK
RTCS Server	세션 관리 및 메시지 송수신을 담당하는 서버
RTCS LBS	사용자들의 RTCS Server 접속을 분배하는 로드 밸런싱 서버
RTCS Session Repository	수평적 확장을 위한 글로벌 세션 저장소 (NHN에서 memcached를 이용해 개발한 플랫폼)
RTCS Admin	RTCS Server 및 LBS의 제어와 모니터링을 위한 서버
Java Service SDK	서비스의 백엔드 서버에서 RTCS에 메시지를 전송하기 위한 SDK

RTCS Component

Javascript Client SDK



```
// Option 객체를 생성한다.
var oOption = new nhn.rtcs.TransportOption({
  domain: "nhncorp.com",
  connectType: "Streaming",
  onEventMessage: function(data) {
    // 서버로부터 받은 메시지를 처리한다.
  }
});

// Client 객체를 생성한다.
var oClient = new nhn.rtcs.RTCSCClient(oOption);

// 서버에 연결한다.
oClient.connect();

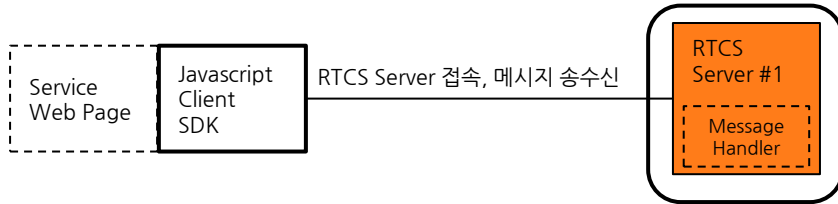
// 서버에 메시지를 보낸다.
oClient.send("message");

// 채널에 가입한다.
oClient.joinChannel("CHANNEL");

// 서버와의 연결을 종료한다.
oClient.disconnect();
```

RTCS Component

RTCS Server



세션 관리

Long Polling, Streaming 연결 처리 및 세션 유지

메시지 관리

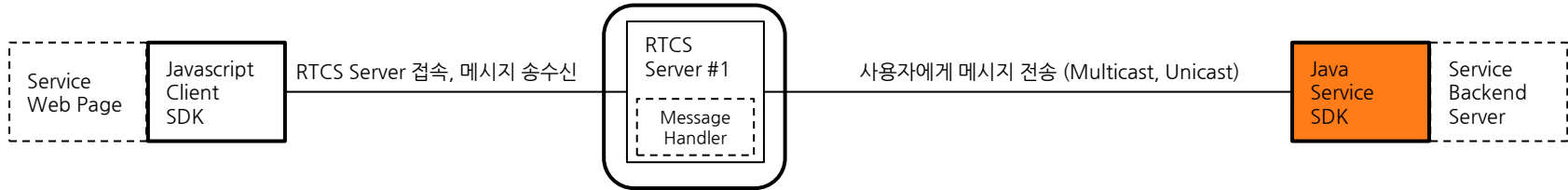
양방향 메시지 송수신 보장, 채널을 통한 멀티케스트 메시지 지원

Message Handler

Javascript Client SDK에서 send() 하는 메시지를 수신해서 서비스 별 로직 처리

RTCS Component

Java Service SDK



```
RTCSClient client = factory.createClient();
```

```
// 특정 사용자에게 메시지 전송 (Unitcast)  
client.sendMessage(userId, message);
```

```
// 특정 채널에 메시지 전송 (Multicast)  
client.sendMessageToChannel(channelId, message);
```

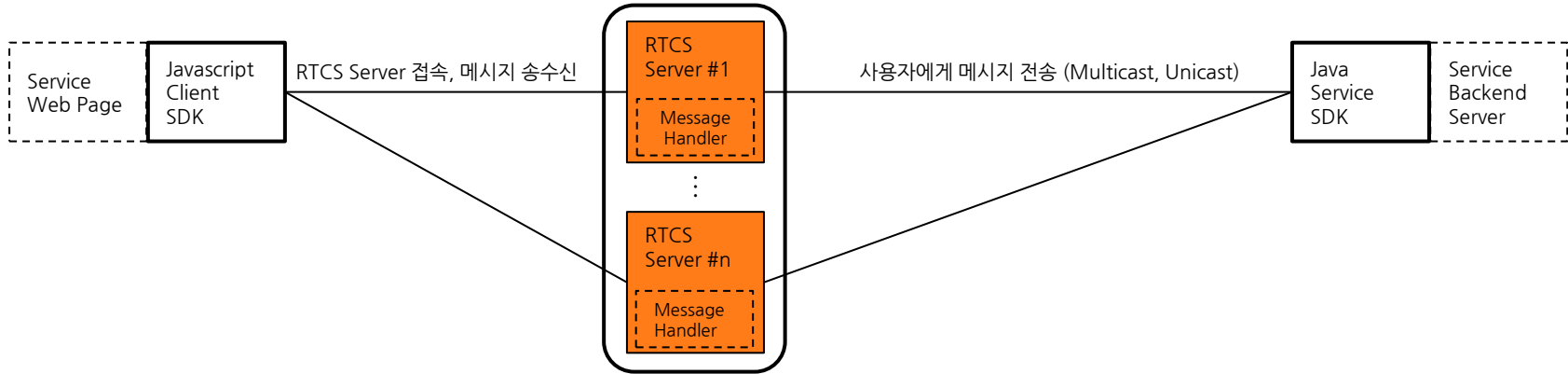
```
// 특정 서버에 메시지 전송  
client.sendMessageToServer(server, message);
```

```
// 모든 사용자에게 메시지 전송 (Broadcast)  
client.sendMessageToAll(message);
```

```
client.shutdown();
```

RTCS 컴포넌트

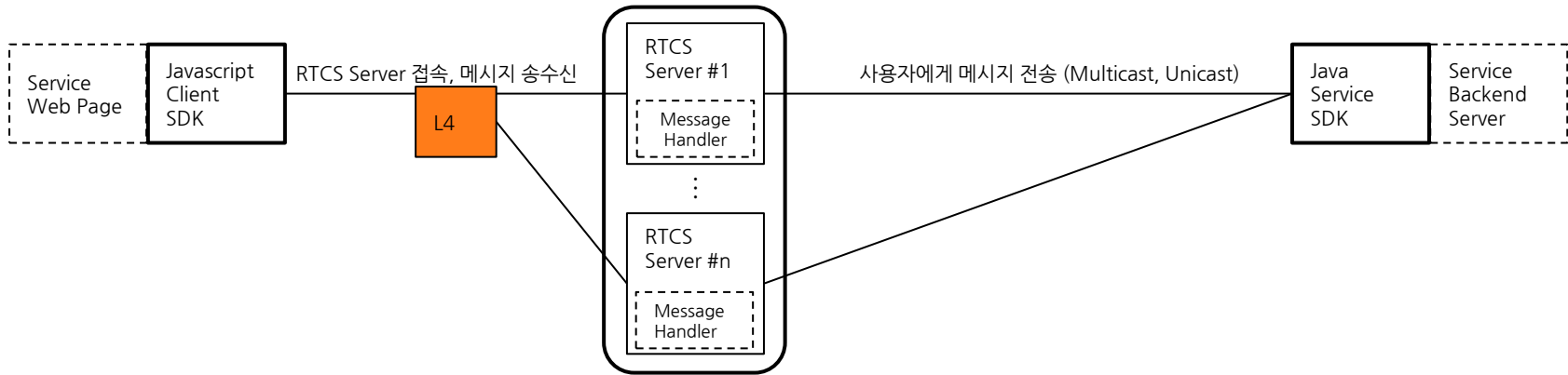
수평적 확장



어떤 RTCS Server에 연결하지?

RTCS 컴포넌트

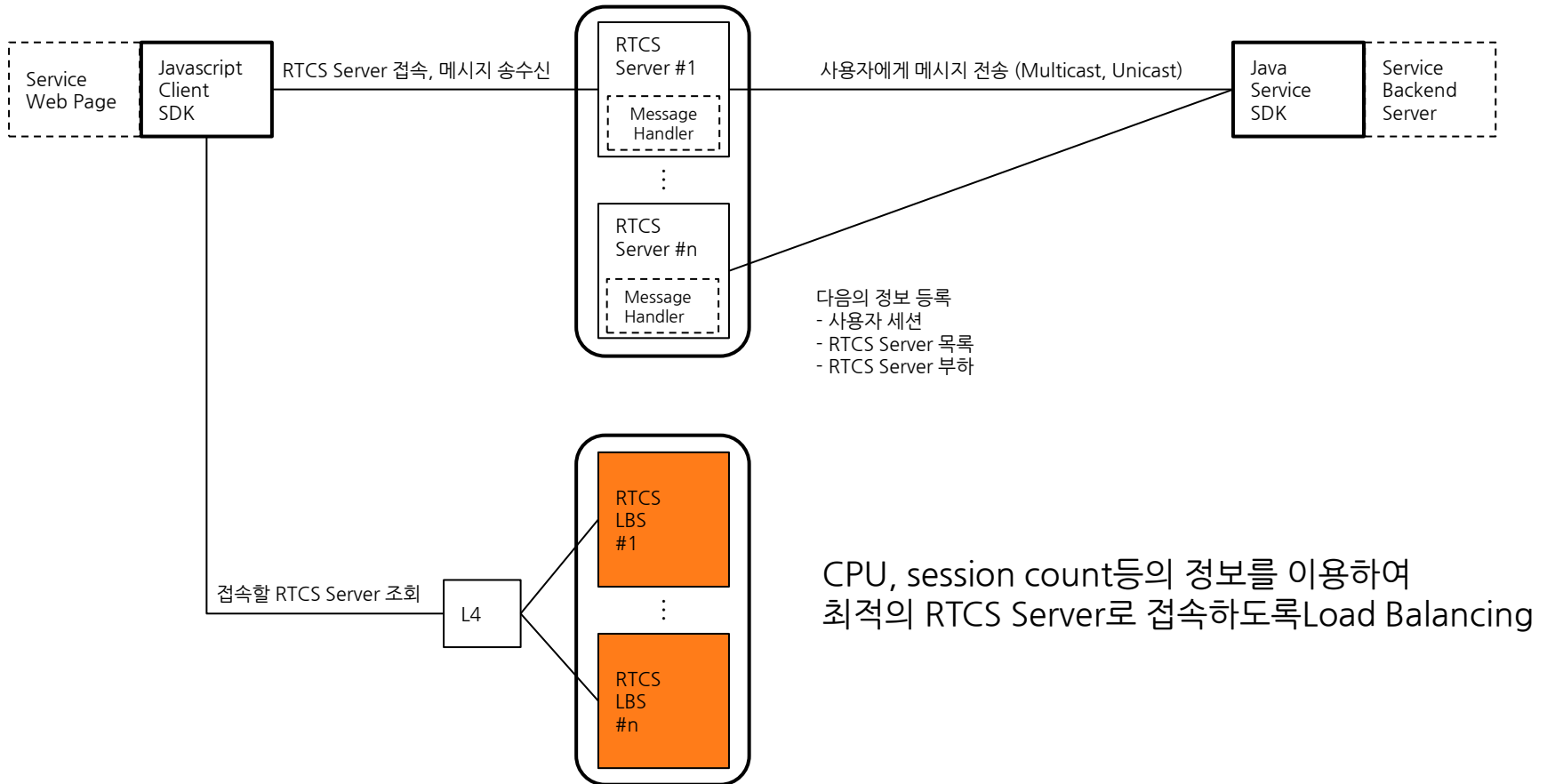
수평적 확장 - L4?



L4는 서비스에서 제어가 어렵다.

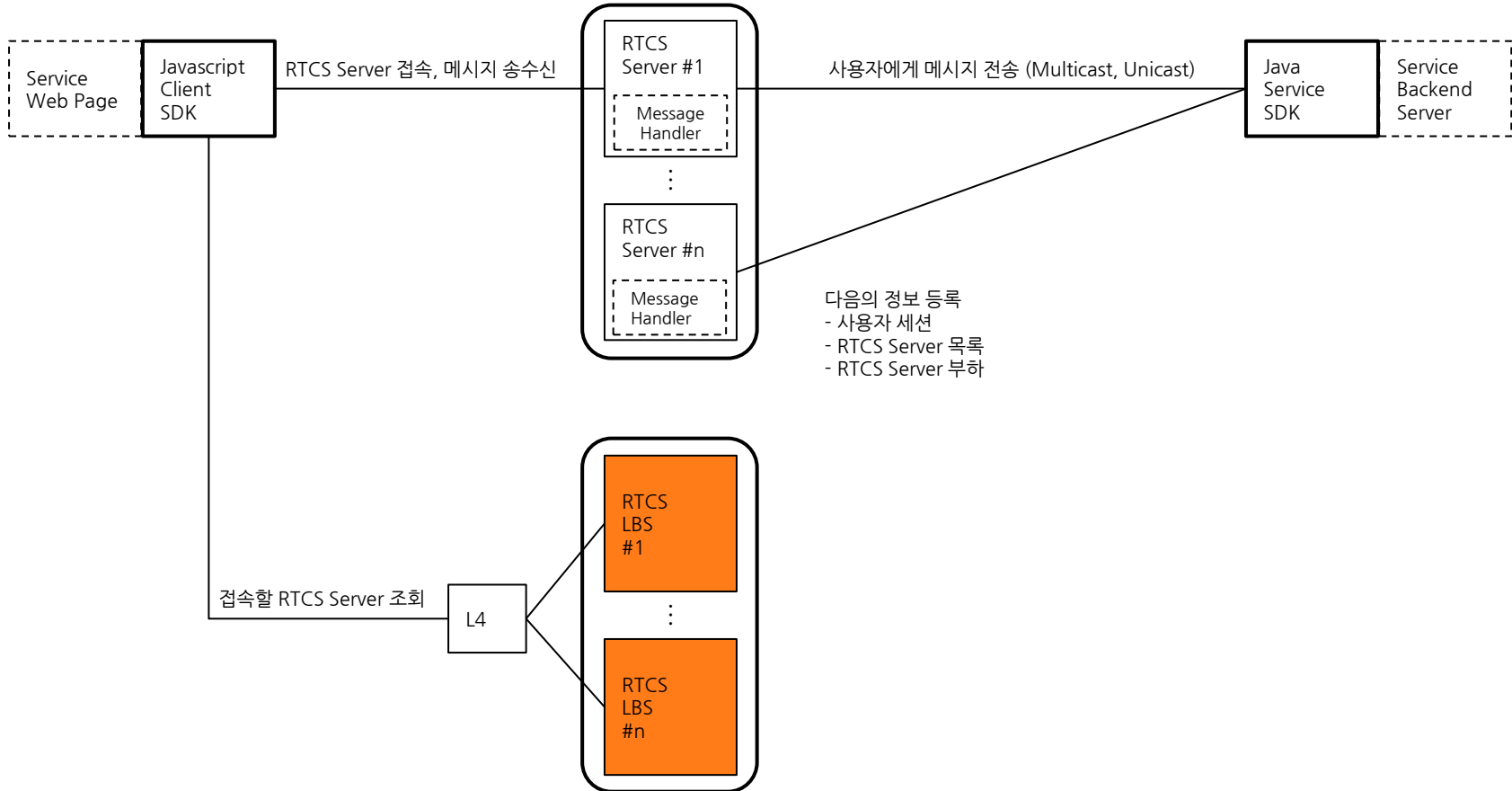
RTCS Component

RTCS LBS



RTCS Component

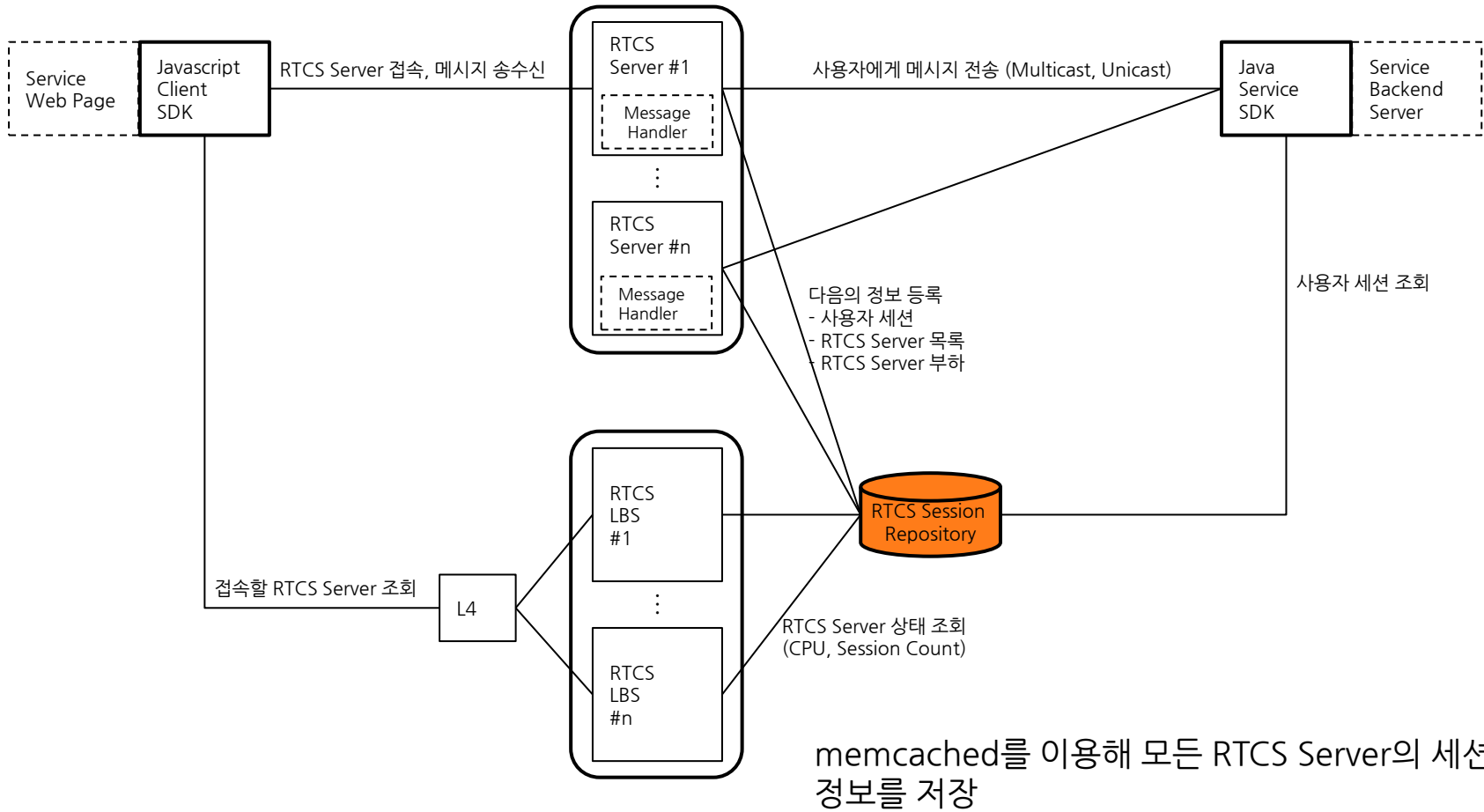
수평적 확장



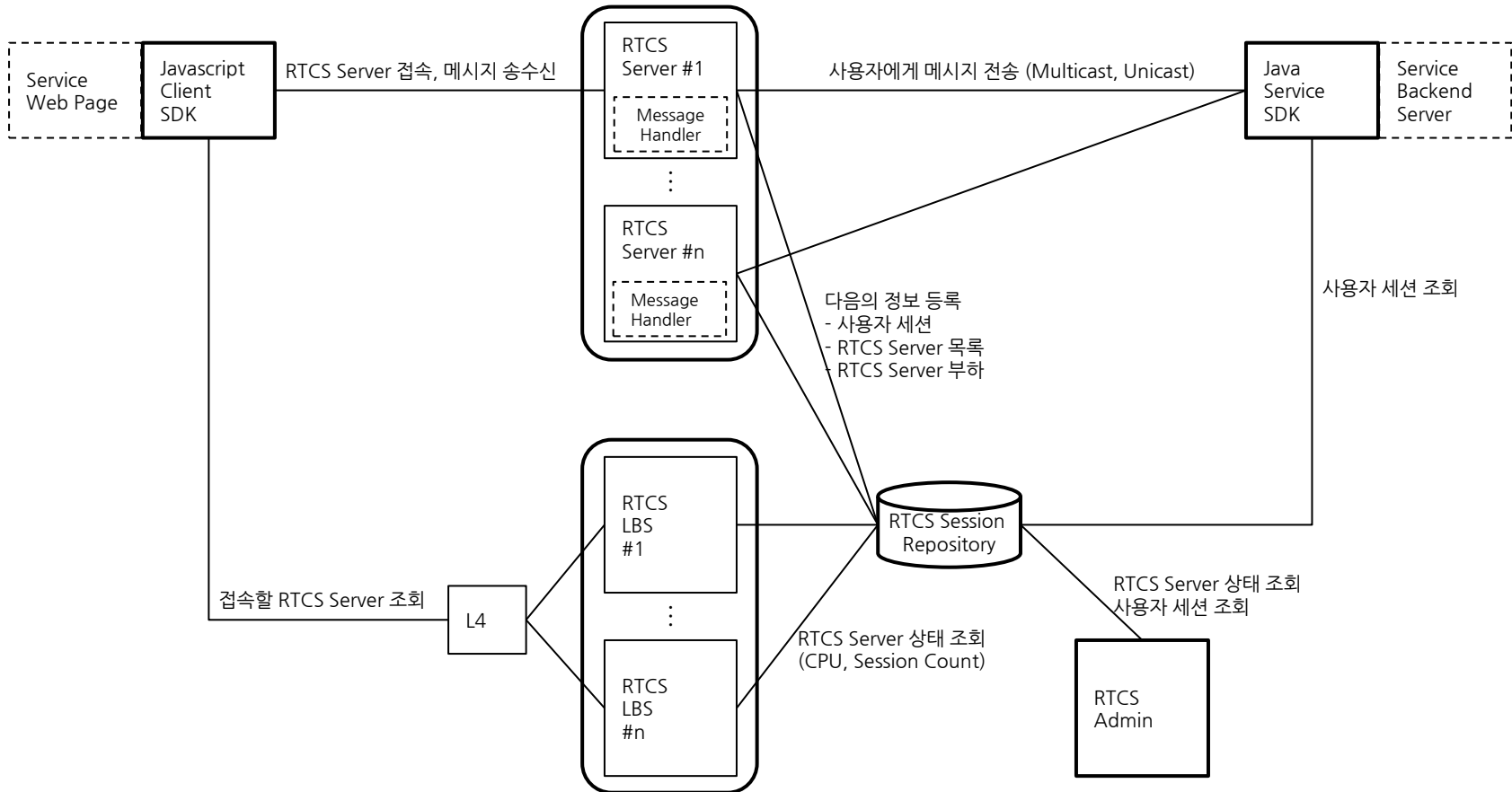
사용자가 어떤 RTCS Server에 연결되어 있지?

RTCS Component

RTCS Session Repository



RTCS Component Complete!

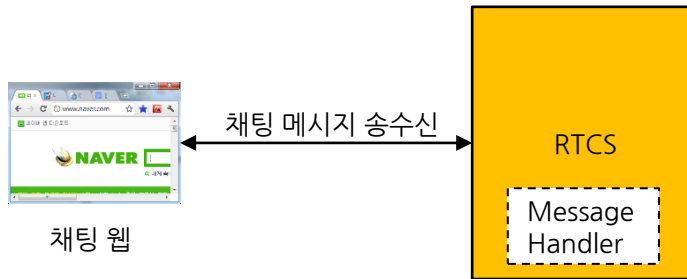


RTCS 적용 아키텍처

- ❖ 채팅
- ❖ 실시간 증권 정보
- ❖ 온라인게임

RTCS 적용 아키텍처

간단한 웹 채팅



Javascript Client SDK

- ❖ connect()로 접속
- ❖ joinChannel()로 특정 채팅 방 입장
- ❖ send()로 채팅 메시지 전송
- ❖ onEventMessage ()로 채팅 메시지 수신

RTCS Server

- ❖ 채팅 메시지를 받아서 채널로 전송하는 MessageHandler 구현

RTCS 적용 아키텍처

간단한 웹 채팅 - sample code

```
<script type="text/javascript">
    var oTransportOption = new nhn.rtcs.TransportOption({
        userId : "user_demo",
        urlForLBS : "http://rtcslbs.nhncorp.com/getServerInfo",
        onChangeReadyState : function(sState) {
            if (sState === "READYSTATE_OPEN") {
                oClient.joinChannel("chat");
            }
        },

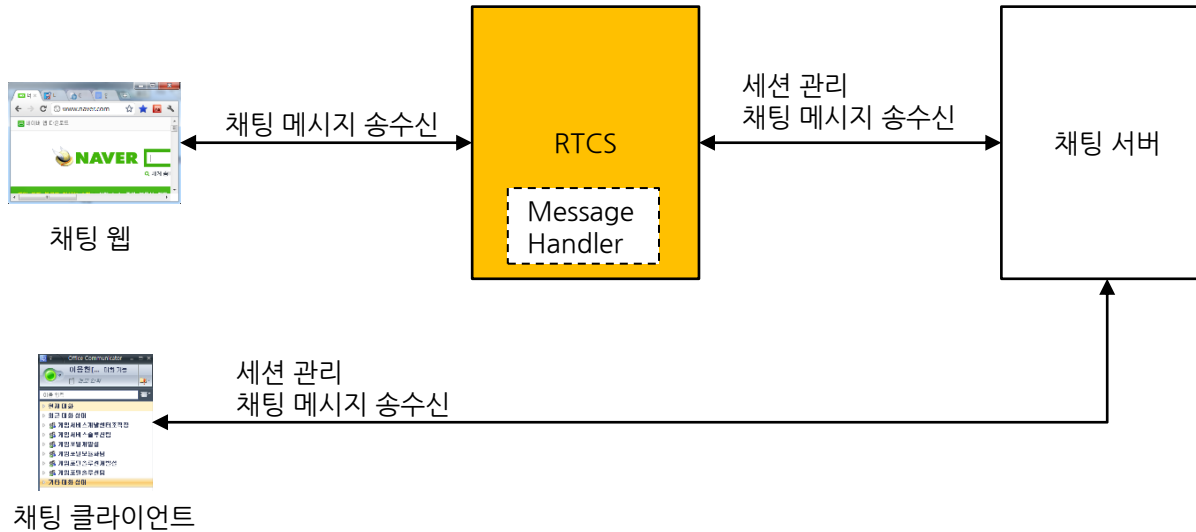
        onEventMessage : function(sMessage) {
            var chatHistory = document.getElementById("history");
            chatHistory.innerHTML += "<li>" + sMessage + "</li><br>";
        }
    });

    var oClient = new nhn.rtcs.RTCSCClient(oTransportOption);
    oClient.connect();

    function sendMessage(message) {
        oClient.send(message);
    }
</script>
```

RTCS 적용 아키텍처

멀티 플랫폼 채팅



Javascript Client SDK

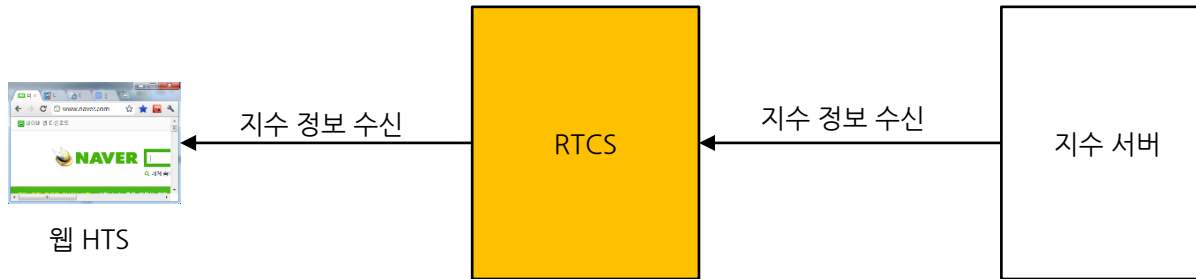
- ❖ connect()로 접속
- ❖ send()로 채팅 메시지 전송
- ❖ onEventMessage()로 채팅 메시지 수신

RTCS Server

- ❖ 세션 연결 및 종료 시 채팅 서버에 알려주는 EventHandler 구현
- ❖ 채팅 메시지를 받아서 채널 서버로 전송하는 MessageHandler 구현

RTCS 적용 아키텍처

주식 증권 정보



Javascript Client SDK

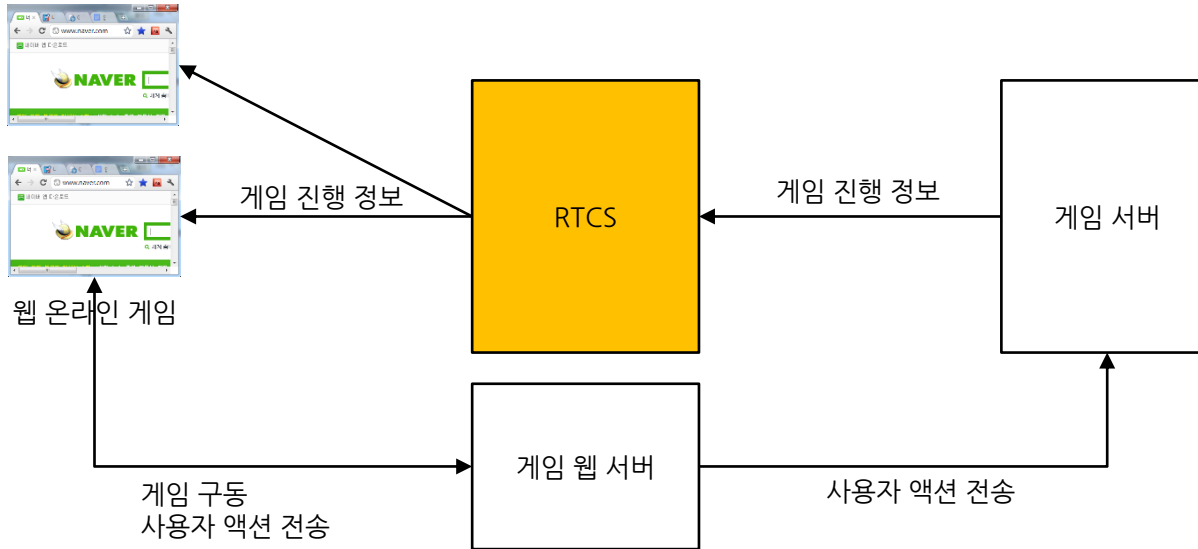
- ❖ connect()로 접속
- ❖ joinChannel()로 지수 정보 채널 가입
- ❖ onEventMessage()로 지수 정보 수신

Java Service SDK

- ❖ sendMessageToChannel로 지수 정보 송신

RTCS 적용 아키텍처

웹 온라인 게임



Javascript Client SDK

- ❖ connect()로 접속
- ❖ joinChannel()로 게임 방 입장
- ❖ onEventMessage()로 게임 진행 정보 수신

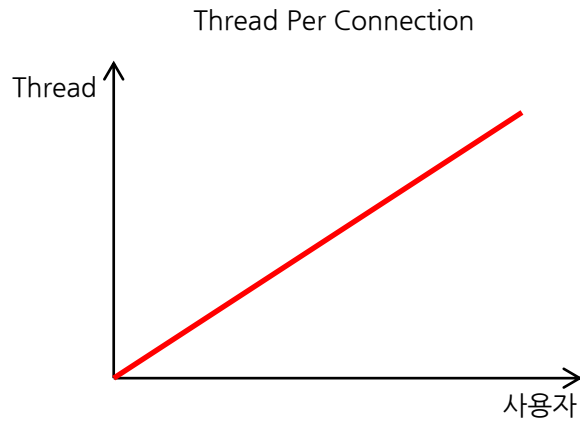
Java Service SDK

- ❖ 게임 서버에서 sendMessage로 게임 진행 정보 송신

RTCS 구현 경험

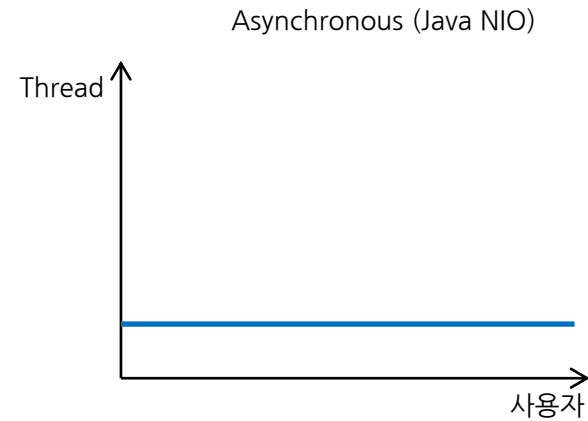
Servlet Container

Long Lived HTTP Connection



Thread Per Connection

- ❖ 사용자 수 만큼 Thread가 증가
- ❖ Context Switch 비용으로 성능 저하



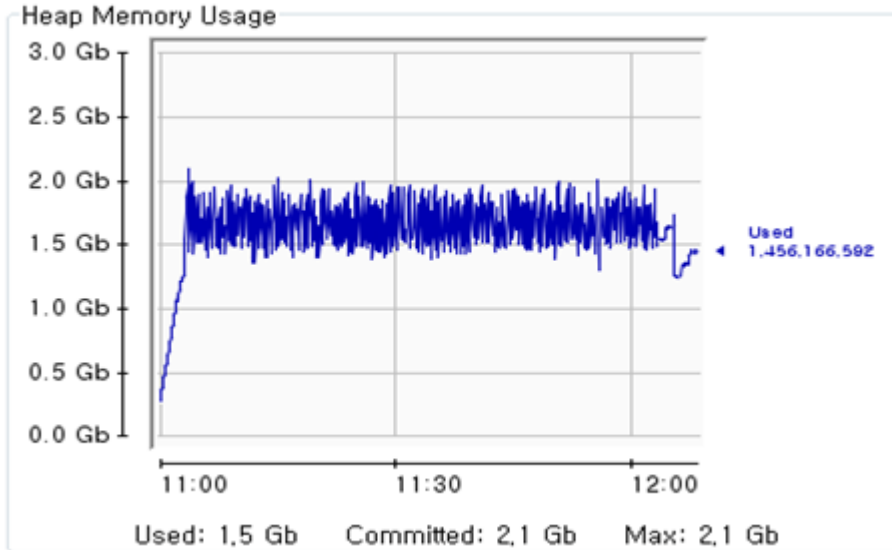
Asynchronous (Java NIO)

- ❖ 사용자 수가 늘어나도 Thread는 증가하지 않음
- ❖ Tomcat6 의 CometProcessor (event)
- ❖ Jetty7의 Continuation (Asynchronous Servlet)
- ❖ Netty

Servlet Container

최대 사용자 수 (Heap Memory)

Tomcat6에 10,000명 사용자 접속 시 Heap Memory Graph (-Xmx2048m)



Tomcat의 주요 Live Object

- org.apache.catalina.connector.Request
- org.apache.tomcat.util.net.NioChannel
- org.apache.catalina.connector.Response
- org.apache.coyote.RequestInfo

최대 사용자 수는 Heap Memory에 좌우

- ❖ Tomcat6는 http request당 약 100KB 사용
- ❖ Jetty7는 http request당 약 40KB 사용

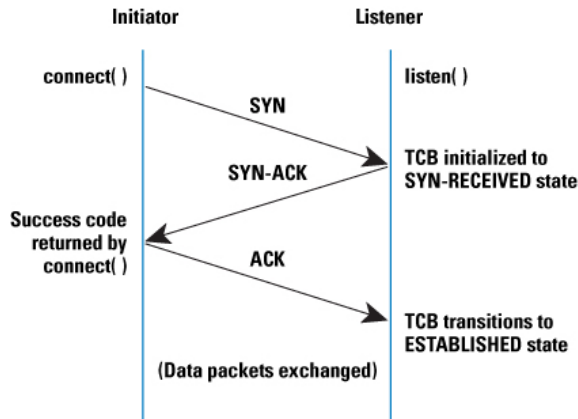
Max Heap Memory를 크게 하면?

- ❖ Full GC Time이 길어져 Stop-the-world가 발생 가능성 높음

Servlet Container

HTTP Keep-Alive

TCP 3-way handshake



TCP connection의 3-way handshake

- ❖ Long Polling의 경우는 빈번한 TCP connection이 이루어짐
- ❖ 빈번한 TCP connection은 서버 자원을 낭비한다.
- ❖ HTTP Keep-Alive 필수

Tomcat6의 설정

<Connector

```
protocol="org.apache.coyote.http11.Http11NioProtocol"  
useComet="true"  
maxKeepAliveRequests="1000"
```

...생략... >

maxKeepAliveRequests 옵션

- == 1 : http keep-alive off
- > 1 : 해당 request 까지 TCP connection 유지
- == -1 : 계속해서 TCP connection 유지

Servlet Container

Tomcat GC, http worker thread lock

processorCache 옵션 적용 전



processorCache 옵션 적용 후



Tomcat에서 GC 줄이는 옵션

다음 옵션을 동시에 처리할 사용자 수 만큼 설정해야 한다.

socket.bufferPool	NioChannel objects, default is 500.
socket.processorCache	SocketProcessor objects, default is 500.
socket.keyCache	KeyAttachment objects, default is 500.
socket.eventCache	PollerEvent objects, default is 500.

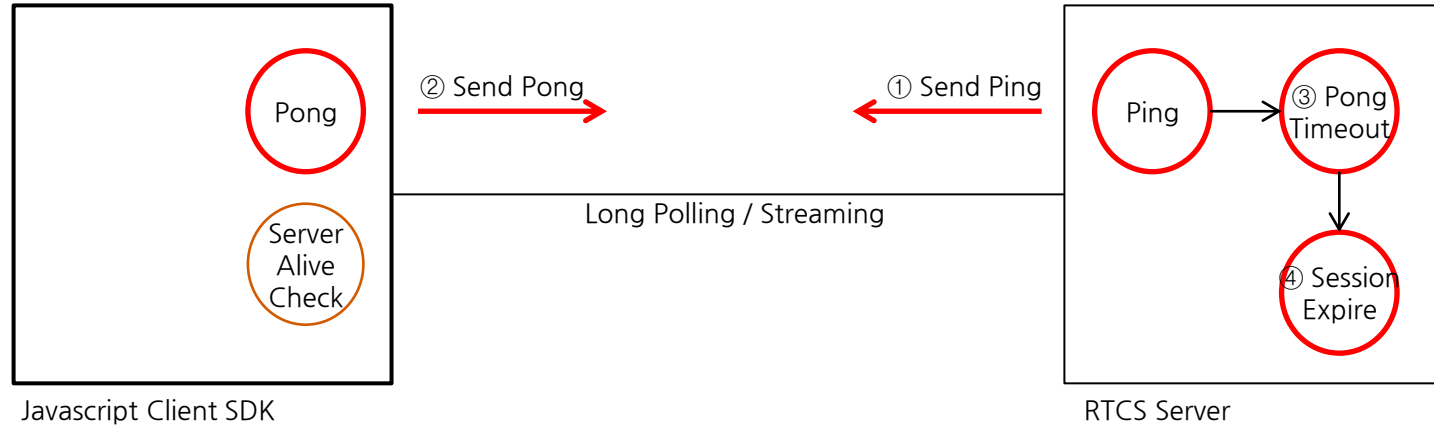
HTTP Worker Thread Lock을 줄이는 옵션

다음 옵션이 동시에 처리할 사용자 수 보다 적으면, work thread에 lock이 빈번히 발생한다.

processorCache	The protocol handler caches Http11NioProcessor objects to speed up performance. default is 200.
-----------------------	---

브라우저

연결 유지 (Ping/Pong, Alive Check)



Ping/Pong (Client Alive Check)

1. RTCS Server에서 Ping을 보내면 Client는 Pong응답을 보낸다.
2. 특정 시간 동안 Pong이 안 오면 RTCS Server는 연결 해제 후보로 등록한다. (Pong Timeout)
3. 연결 해제 후보 등록 후 특정 시간이 지나면 연결을 종료한다. (Session Expire)

Server Alive Check

Client SDK는 특정 시간 안에 RTCS Server에서 Ping이 안 오면 재 접속을 시도한다.

브라우저

기타 자질구레 한...

Same Origin Policy

- ❖ hidden iframe (base domain)
- ❖ [JSONP](#)
- ❖ [CORS](#)

Streaming 구현 시 브라우저 버퍼

- ❖ 서버에서 최초 응답 시 dummy bytes (4~5KB)를 보내야 한다.

Streaming 구현 시 브라우저 메모리 leak

- ❖ Streaming으로 응답을 계속 보내면 브라우저에서 시스템 메모리를 계속 사용함
- ❖ 주기적 reconnect 필요 (memory clear)

브라우저 별 Streaming 구현

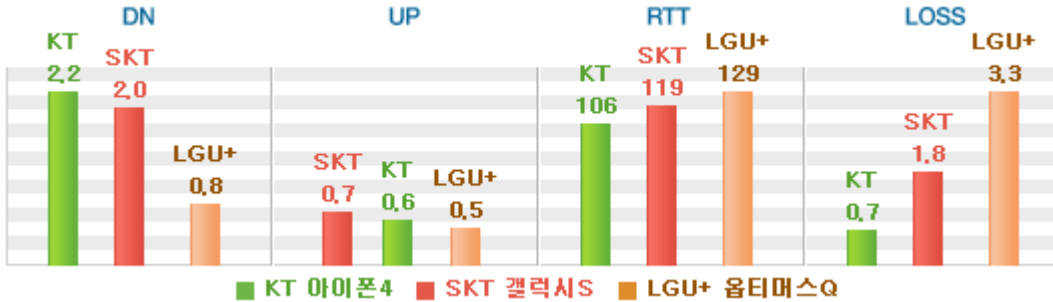
- ❖ IE는 XMLHttpRequest.readyState = 3 일 때 responseText 접근 안됨 (Hidden IFRAME로 구현)
- ❖ Opera는 XMLHttpRequest.readyState = 3을 첫 바이트 받았을 때만 발생시킴 (Server-sent Events로 구현)
- ❖ Android는 XMLHttpRequest.readyState = 3을 매번 약 4KB 받았을 때만 발생시킴
 - 방법 없음, 또는 flash socket (simulate WebSocket)

브라우저의 동일 도메인 커넥션 수 제한

Proxy서버 사용 시 영향?

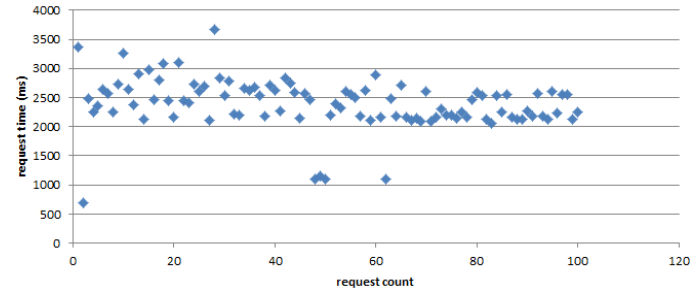
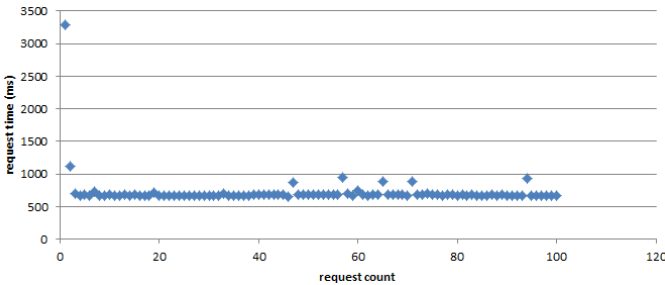
모바일 3G망의 특성

<http://www.benchbee.co.kr/04weather/speed/mobile.asp>



	Down (Mbps)	Up (Mbps)
KT+iPhone3GS	1.79	0.06
KT+iPhone4	2.17	0.59
KT+GalaxyS2	2.34	0.87
SKT+iPhone3GS	1.06	0.05
SKT+iPhone4	1.85	0.41
SKT+GalaxyS2	2.30	0.66

KT+iPhone3GS에서 요청 간격 별 요청 완료 시간 (좌 1초, 우 3초)



3G망에서 HTTP Request는 0.6~3초 걸림

- ❖ Upstream Bandwidth의 제약 및 긴 RTT
- ❖ 요청 간격이 길면 단말기의 idle → busy 전환 delay 발생
- ❖ 쿠키 등으로 인해 request packet 크기가 커지면 추가 지연 발생

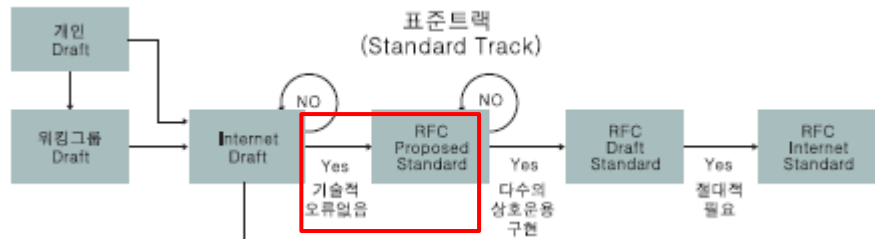
그래서

- ❖ 빈번한 request가 있는 Long Polling보다 Streaming이 유리
- ❖ request 크기를 최대한 줄여야 한다. (쿠키가 문제)

리얼타임웹의 방향

WebSocket is **Proposed Standard** now!

- Protocol <https://datatracker.ietf.org/doc/draft-ietf-hybi-thewebsocketprotocol/>
- API <http://dev.w3.org/html5/websockets/>



Android의 브라우저 Chrome으로 변경?

IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
6.0	4.0	11.0	3.2	10.6				
7.0	5.0	12.0	4.0	11.0	3.2		10.0	2.1
8.0	6.0	Moz 13.0	5.0	11.1	4.0-4.1		11.0	2.2
9.0	7.0	Moz 14.0	5.1	11.5	4.2-4.3	5.0-6.0	11.1	2.3 3.0
10.0	8.0	Moz 15.0		12.0	5.0			
	9.0	Moz 16.0	6.0	12.1				

Open Source

- ❖ <http://nodejs.org/> (<http://socket.io/>)
- ❖ <http://cometd.org/>
- ❖ <http://opencoweb.org/>
- ❖ <http://www.ape-project.org/>
- ❖ <http://meteorserver.org/>

Cloud

- ❖ <http://pusher.com/>
- ❖ <http://www.joyentcloud.com/> (nodejs hosting)
- ❖ <http://easywebsocket.org/> (?)

Commercial

- ❖ <http://www.lightstreamer.com/>
- ❖ <http://kaazing.com/>
- ❖ <http://www.electrotank.com/>
- ❖ <http://www.freeliberator.com/>
- ❖ <http://migratory.ro/>
- ❖ <http://www.stream-hub.com/>

You!

감사합니다.

