



2015년 공개SW개발자센터 컨퍼런스



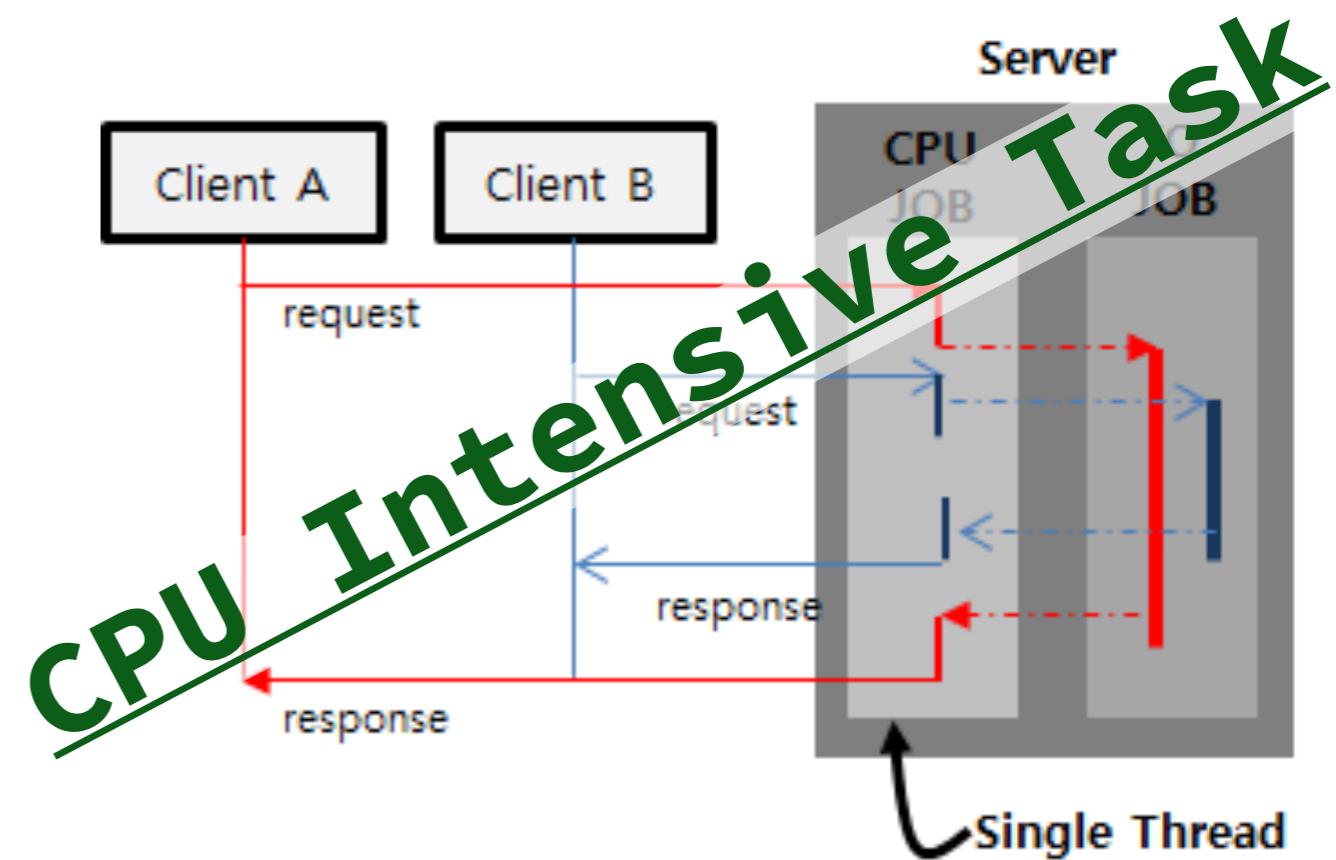
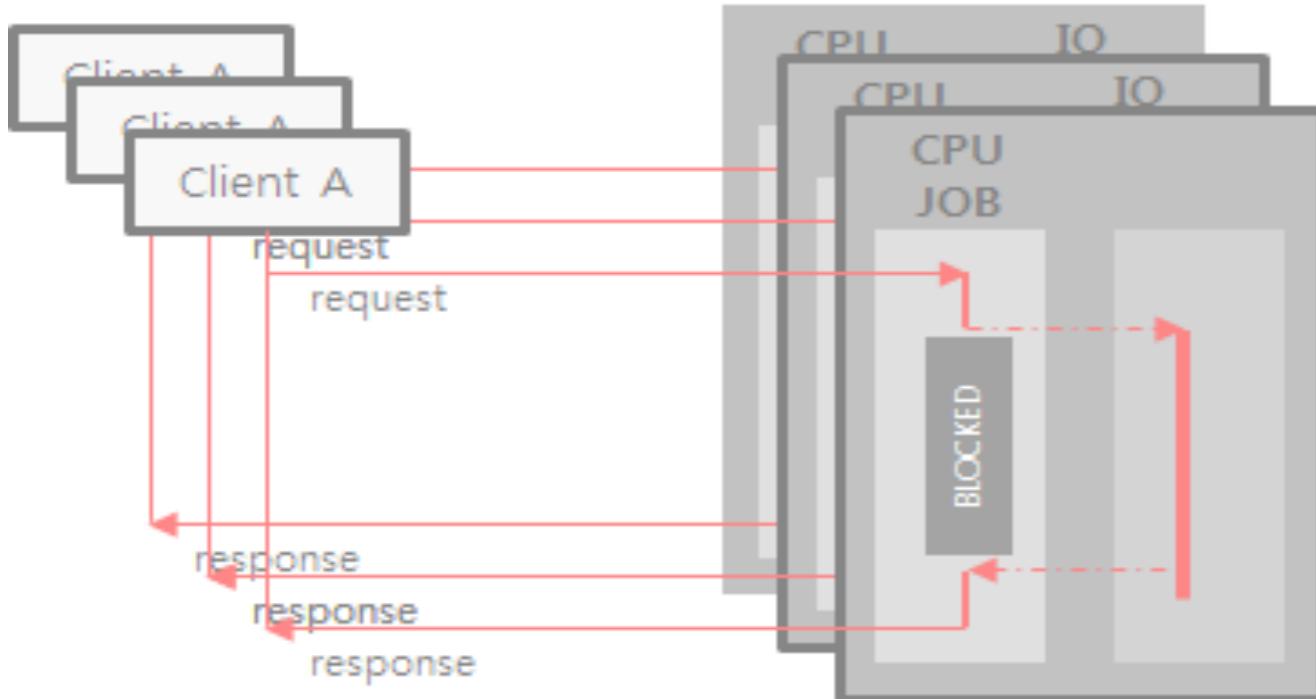
# 분산 환경을 고려한 실시간 서비스 플랫폼 개발 사례



<http://stalk.io>

김요한 (yohany@gmail.com)

n o d e J S ™ ?



# 코드 라인 수를 줄일 수 있다.

간단하다. 빨리 개발할 수 있다.

PayPal wrote the same app twice in Node and Java  
Node was

2x requests/sec,  
33% less code,  
35% less response time.

# 페라리 만큼 빠르다.

CPU 타입이나, OS 가 무엇인지, 그리고 어떤 걸 개발하느냐에 따라 다르다.  
개발 속도 만큼은 정말 빠르다.

난 미니밴도 좋은데 . . .

Ruby on Rails is like a minivan.  
It has all the features. Node is like a Ferrari.

# 서버 시작 / 중지 시간이 상당히 빠르다.

배포시간이 짧다. 하루에도 몇번씩 배포할 수 있다.

Fast deployment is a huge advantage for node.  
The 250-500 ms node process start / stop time  
has been a huge benefit.

Hard to deploy multiple times a day  
when it takes 2 hours each time.

# 죽으면 어때 다시 시작하면 되지. . .

Node restarts so quickly  
crashed aren't a big deal.

~~죽으면 어때 다시 시작하면 되지. . .~~

이건 좀 오버인것 같다!

Node restarts so quickly  
~~crashed aren't a big deal.~~

마성에 빠진 JAVA 개발자들은  
아직도 돌아오지 않고 있습니다. . . .

ㅠ, . ㅠ

# 실시간 서비스 플랫폼

# 실시간 서비스?

**Ossama shared an update**

You can save data, update data, and listen for data changes in realtime with only a few lines of code.

**Sara shared an update**

When data changes, apps built with Firebase update instantly across every device - web or mobile.

**Ali shared a picture**



**Sara is online**

누구세요?

Great movie last night!

How's your day?

Awesome, how about you?

**Alex is online**

Hi There?

누구세요?

Great movie last night!

How's your day?

Awesome, how about you?

type something...

12:35	Amsterdam	BD107	Flight closing	
12:40	Rome	AA6519	Boarding	26
12:50	Berlin	BA984	Flight closing	39
13:00	Reykjavik	FI451	Flight closing	36
13:05	Moscow	BA874	Flight closing	38
13:10	Madrid	BA464	Boarding	37
13:15	Hanover	LH6613	Please wait	
13:20	Warsaw	AA6649	Please wait	
13:25	Dusseldorf	BA940	Go to Gate	19
13:45	Tokyo	BA005	Please wait	
13:45	Stockholm	BA780	Gate opens 13:00	
13:50	San Francisco	BA287	Gate opens 12:45	
13:55	Munich	QF3563	Gate opens 13:10	
14:05	Budapest	BA868	Gate opens 13:20	
14:05	Madrid	IB7445	Gate opens 13:20	
14:15	Rome	BA556	Gate opens 13:30	
14:25	Kefallinia	BD7007	Gate opens 13:40	
14:30	Barcelona	BA480	Gate opens 13:45	
14:40	Palma Mallor	LH6561	Gate opens 13:55	
14:45	Milan-Malp	AA6520	Gate opens 14:00	
14:45	Larnaca	CY347	Gate opens 13:30	
14:55	Lisbon	BA502	Gate opens 14:10	
15:00	Prague	AA6635	Gate opens 14:15	
15:05	Nice	BA348	Gate opens 14:20	

15:15	Hamburg	15:15	Hamburg	
15:25	Munich	15:25	Munich	
15:35	Frankfurt	15:35	Frankfurt	
15:40	Marrakech	15:40	Marrakech via: Casablanca	
15:45	Tokyo	15:45	Tokyo	
15:45	Stuttgart	15:45	Stuttgart	
16:00	Malaga	16:00	Malaga	
16:05	Berlin	16:05	Berlin	
16:05	Los Angeles	16:05	Los Angeles	
16:10	Helsinki	16:10	Helsinki	
16:20	Brussels	16:20	Brussels	
16:25	Amsterdam	16:25	Amsterdam	
16:45	Dusseldorf	16:45	Dusseldorf	
16:45	Frankfurt	16:45	Frankfurt	
16:50	Barcelona	16:50	Barcelona	
17:10	Vancouver	17:10	Vancouver	
17:25	Munich	17:25	Munich	
17:25	Istanbul	17:25	Istanbul	
17:50	Warsaw	17:50	Warsaw	
17:50	Milan-Linate	17:50	Milan-Linate	
17:50	Stockholm	17:50	Stockholm	

12:46

Gates 2

Gates 3

Gates 4

You are at Heat

FIDS/T1-341/DEP/A/1

FIDS/T1-342/DEP/A/1

# 실시간 서비스 플랫폼

플랫폼?





# Volkswagen Group Platform



PORSCHE





Dashboards



Mobile Notifications

# REAL-TIME SERVICE PLATFORM



Messengers

## REAL-TIME SERVICE PLATFORM

is ideal for building dashboards, activity streams, notification and chat systems, **backend-less** apps and websites, real-time collaboration, live statistics, remote controls, multiplayer games, and more.

# An API that enables real-time communication across devices

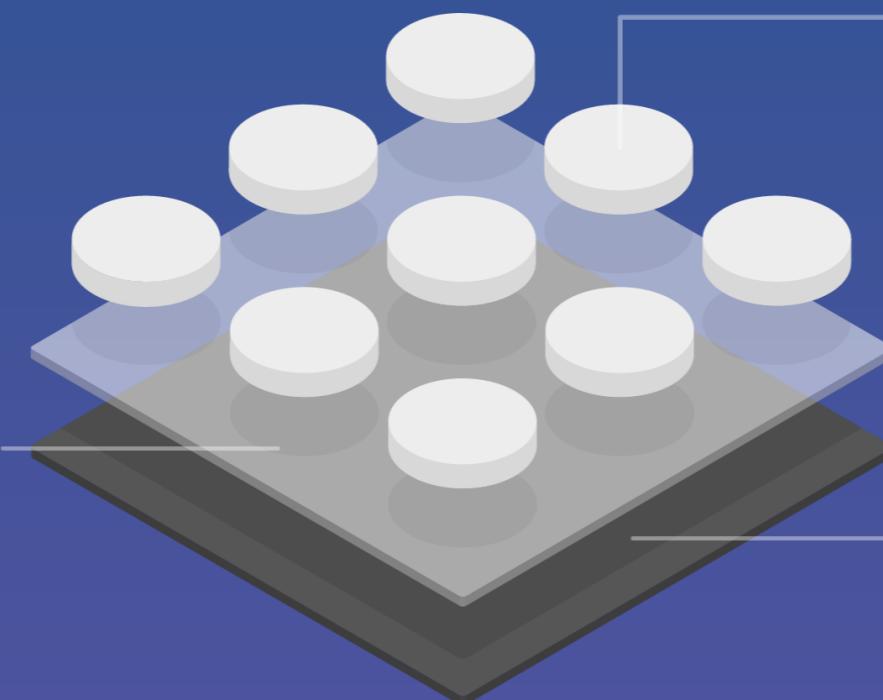
Connect devices to Hydna with just a few lines of code. Once connected, they can share data in real-time.

[Get started](#)

No credit card is required.

A little architectural overview, while Hydna is humming along in the "cloud", waiting for you to connect a device

**Routing** moves data to the intended devices  
Data is routed at lightning speed, whether it's a message to a single laptop or a broadcast to thousands of mobiles.



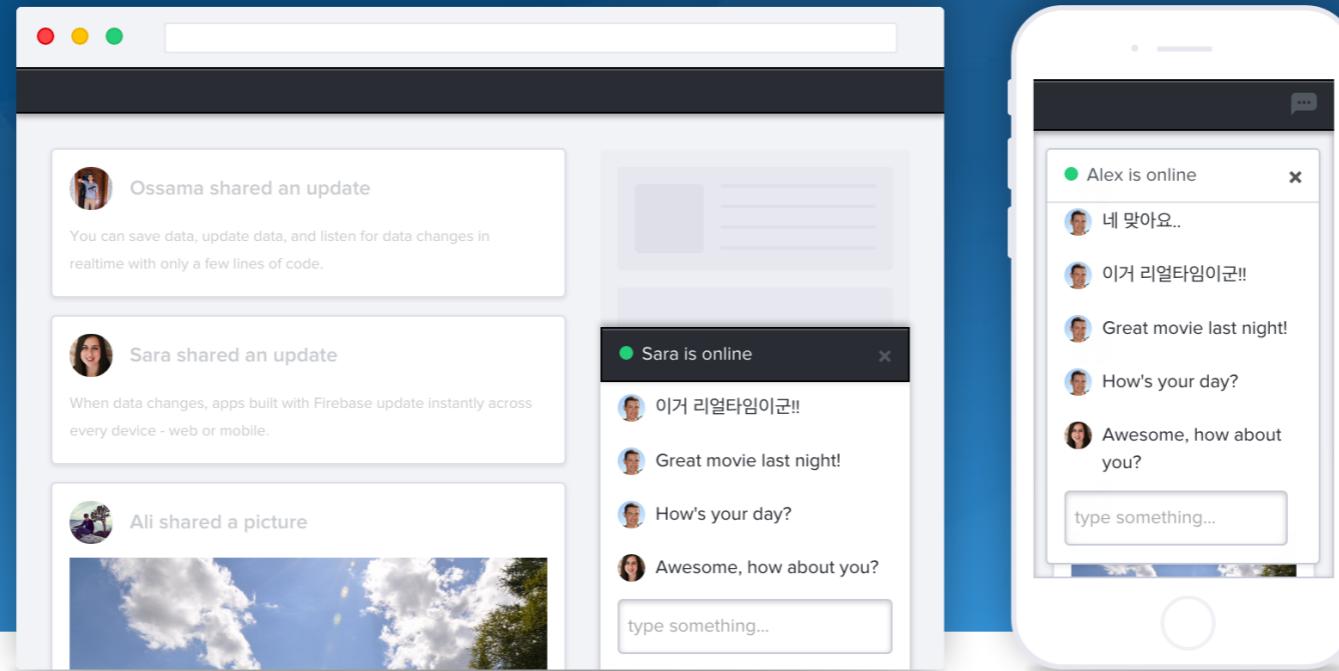
**Transports accept connections**  
Connect anything from iPhones and Android devices, to web browsers, servers and Arduinos.  
Currently support for WebSockets, Binary TCP, HTTP, Comet, and Flash.

**Behaviors act as a "brain", if you need one**  
You can enhance Hydna with small scripts that are deployed to our infrastructure.  
Authenticate devices and users, contact other APIs, store data, keep track of channels, and more.

<https://www.hydna.com/>

# Build Realtime Apps

A powerful API to store and sync data in realtime.



WE'RE EXCITED TO ANNOUNCE

Firebase has joined the **Google** team! Learn more...

THE REALTIME APP PLATFORM

## Store & Sync Data Instantly

Build realtime mobile and web apps in minutes using client-side code and our powerful API. Save time. Delight your users.

<https://www.firebaseio.com/>

**Start with a Platform,**  
**and Then Use it for Everything.**

# 분산 환경을 고려한 실시간 서비스 플랫폼 개발 사례

## “ socket.io 기반의 실시간 채팅 시스템 ”

- : zookeeper, redis 활용
- : 효율적인 Weighted Consistent Hashing 구현
- : 프로세스 모니터링 & 실시간 서버 모니터링

# Real-time Dashboard



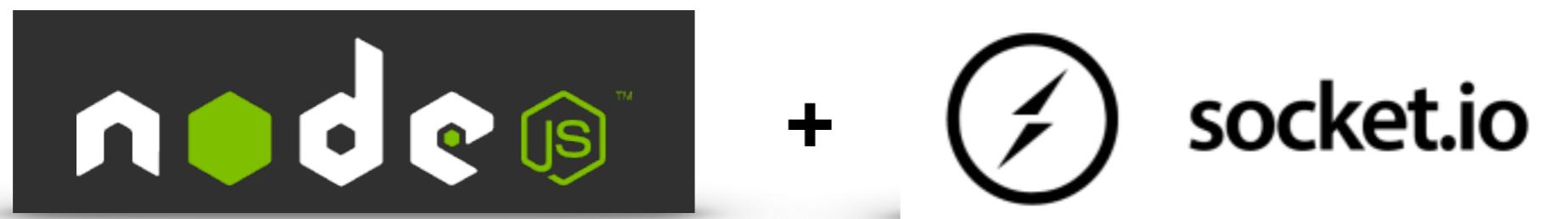
# Instant Messenger

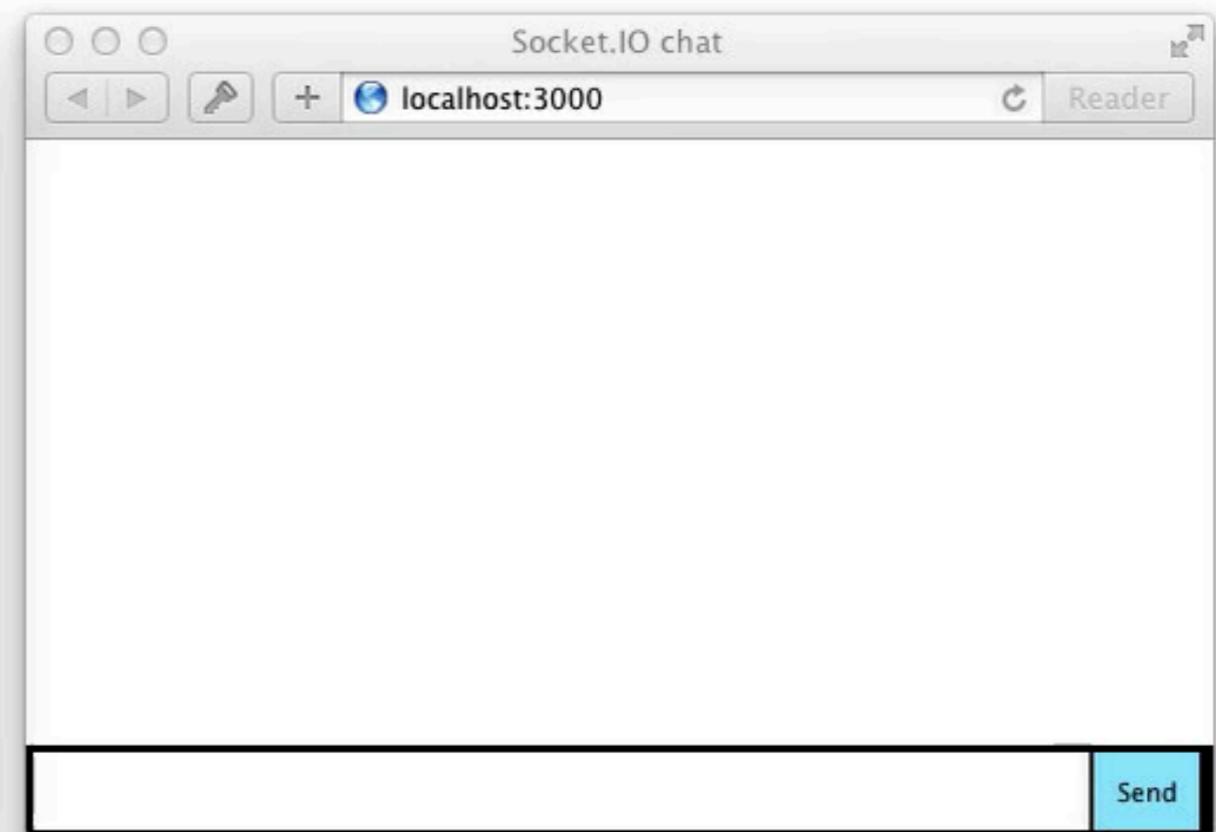


# Mobile Push Notification









```
1 var app = require('express')();
2 var http = require('http').Server(app);
3 var io = require('socket.io')(http);
4
5 app.get('/', function(req, res){
6   res.sendFile(__dirname + '/index.html');
7 });
8
9 io.on('connection', function(socket){
10   socket.on('chat message', function(msg){
11     io.emit('chat message', msg);
12   });
13 });
14
15 http.listen(3000, function(){
16   console.log('listening on *:3000');
17 });
```

**ACCESS** Authentication

**SESSION** Control

**DATA** Storage

**PUSH** Notification

**USER** Management

**DEVICE** Management



socket.io

**ACCESS** Authentication

**SESSION** Control

**DATA** Storage

**PUSH** Notification

**USER** Management

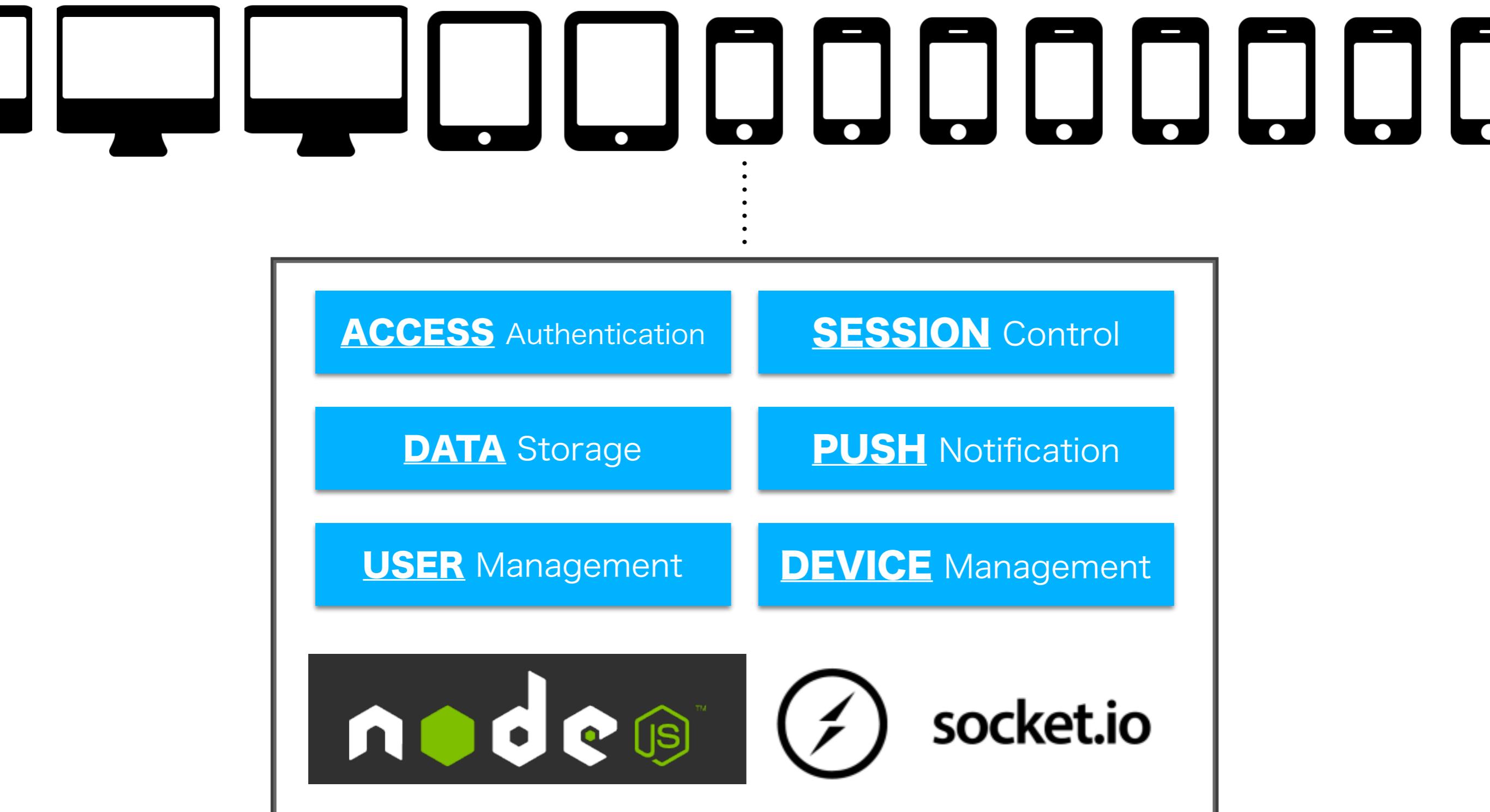
**DEVICE** Management

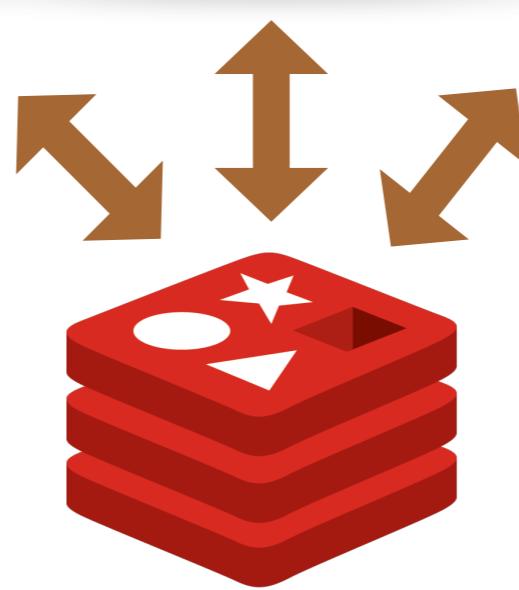


socket.io



mongoDB





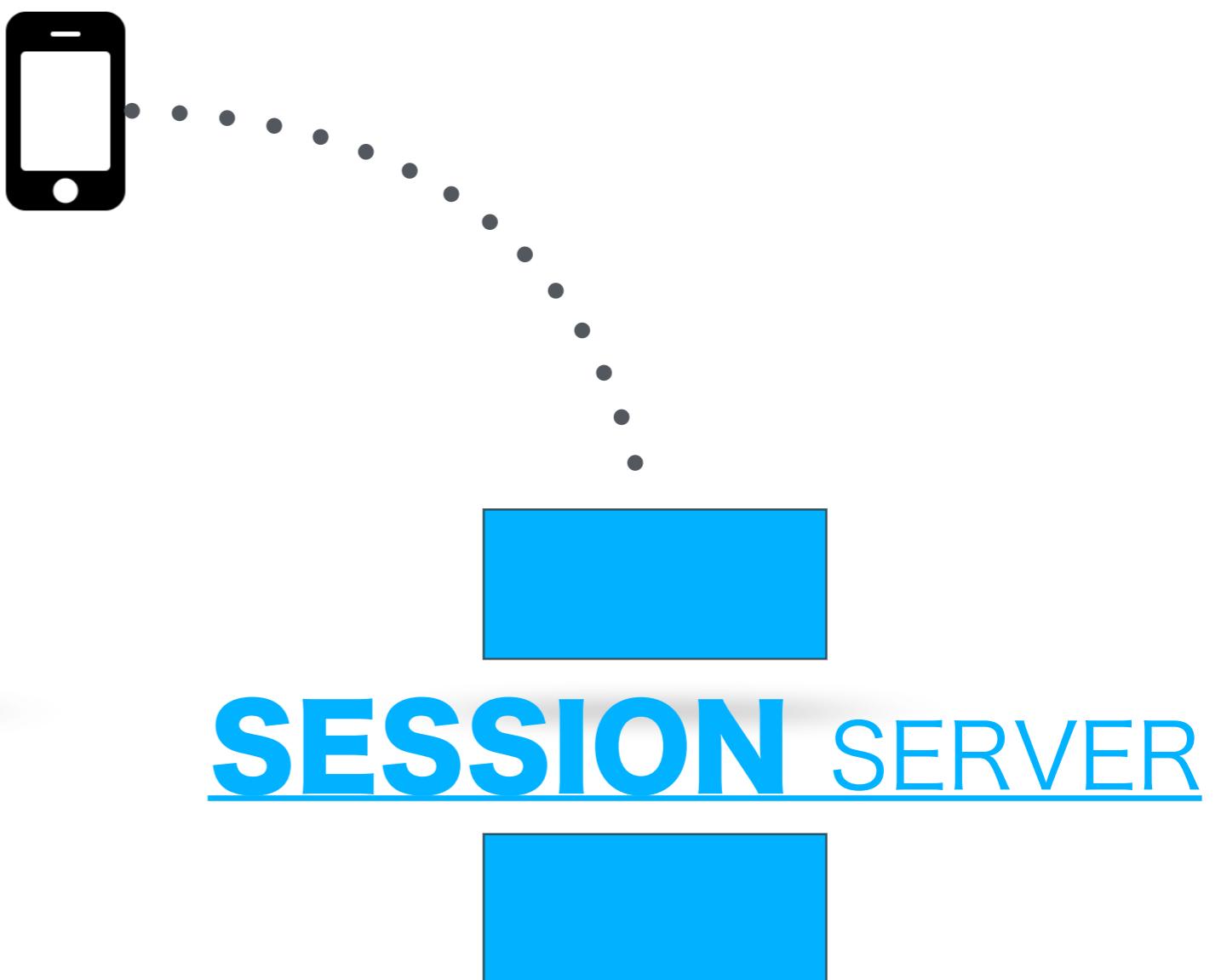
redis

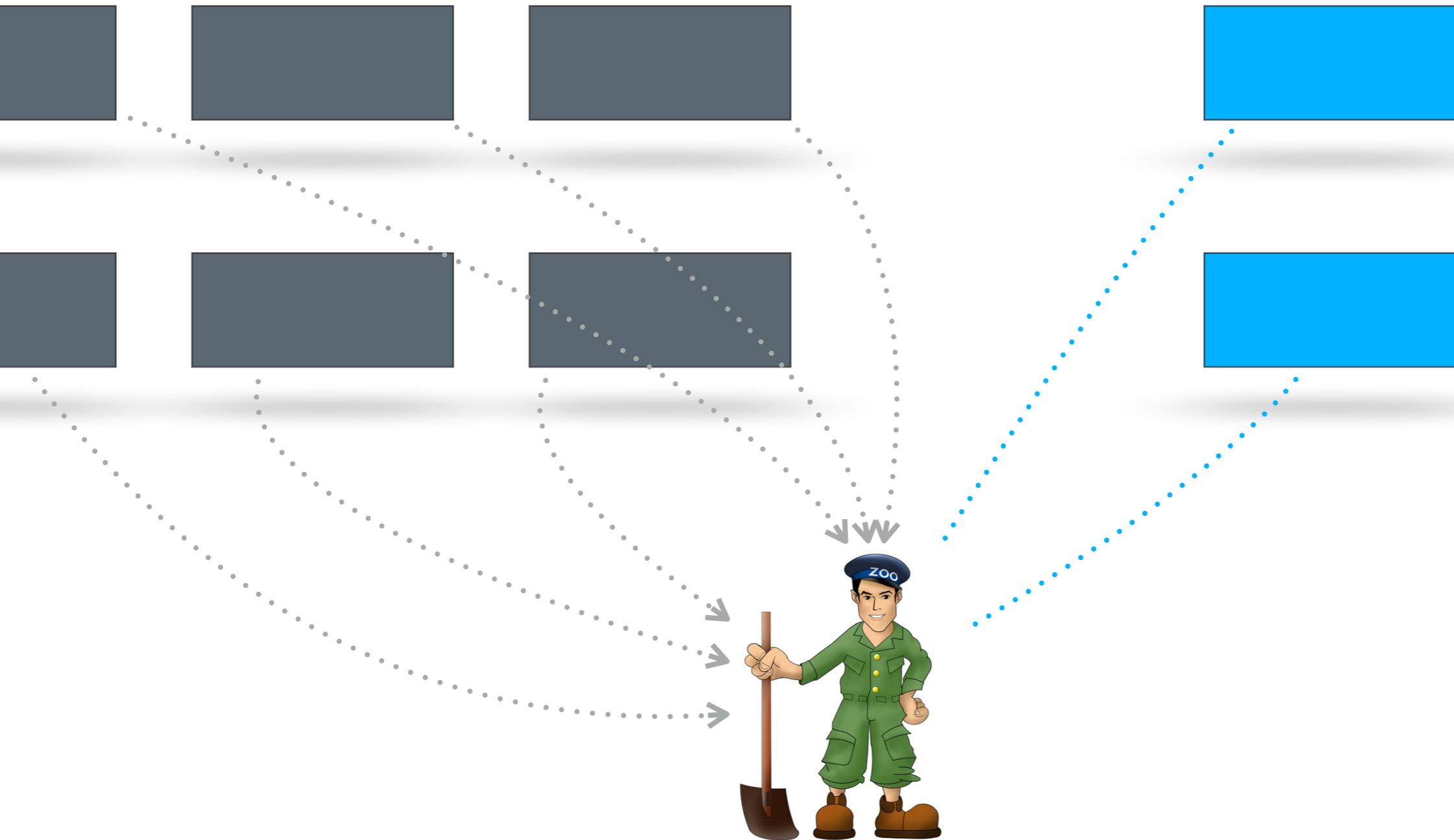


**CHANNEL SERVER**

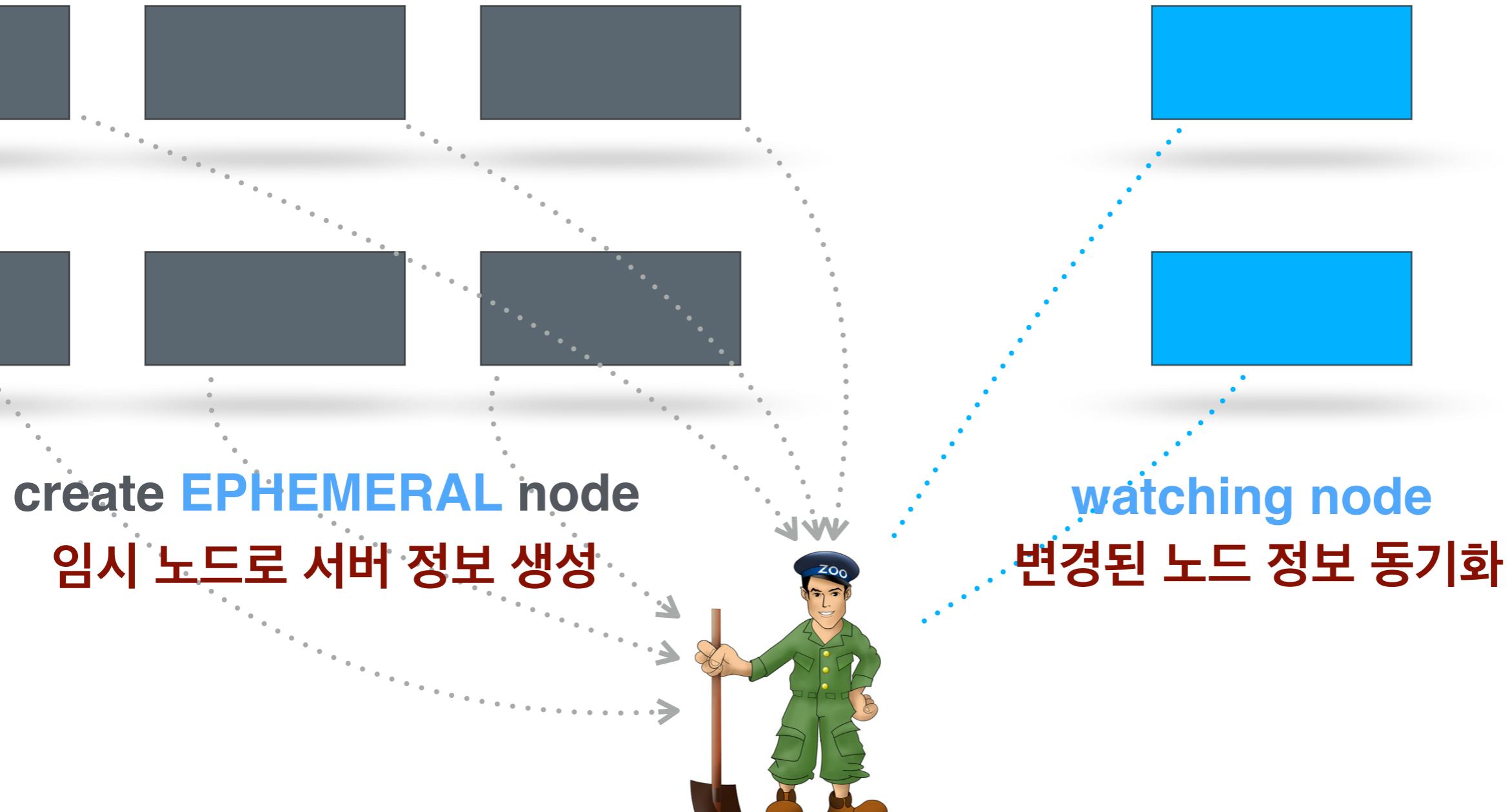


**SESSION SERVER**





**Apache ZooKeeper™**



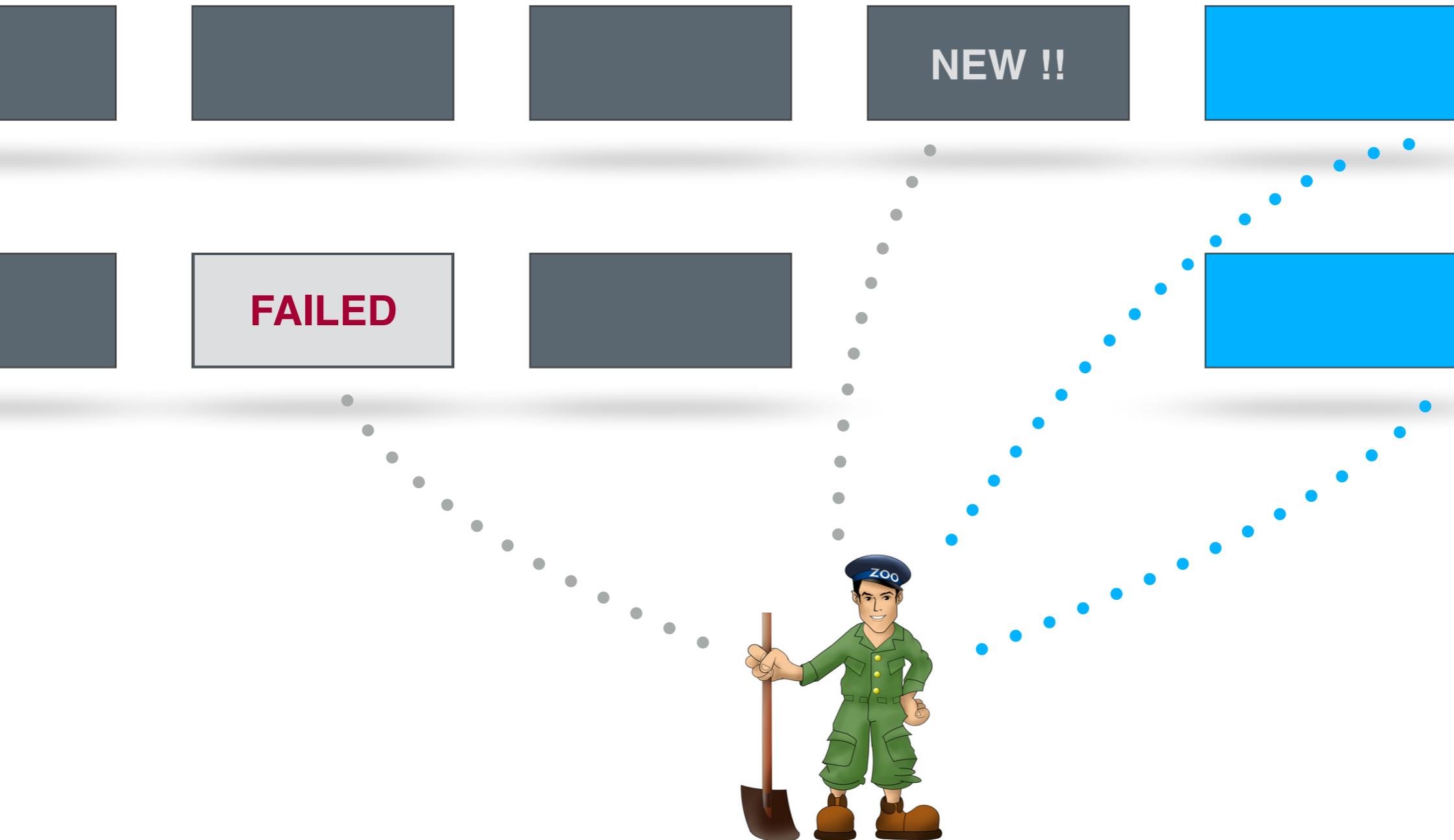
Apache ZooKeeper™

# The King of Coordination

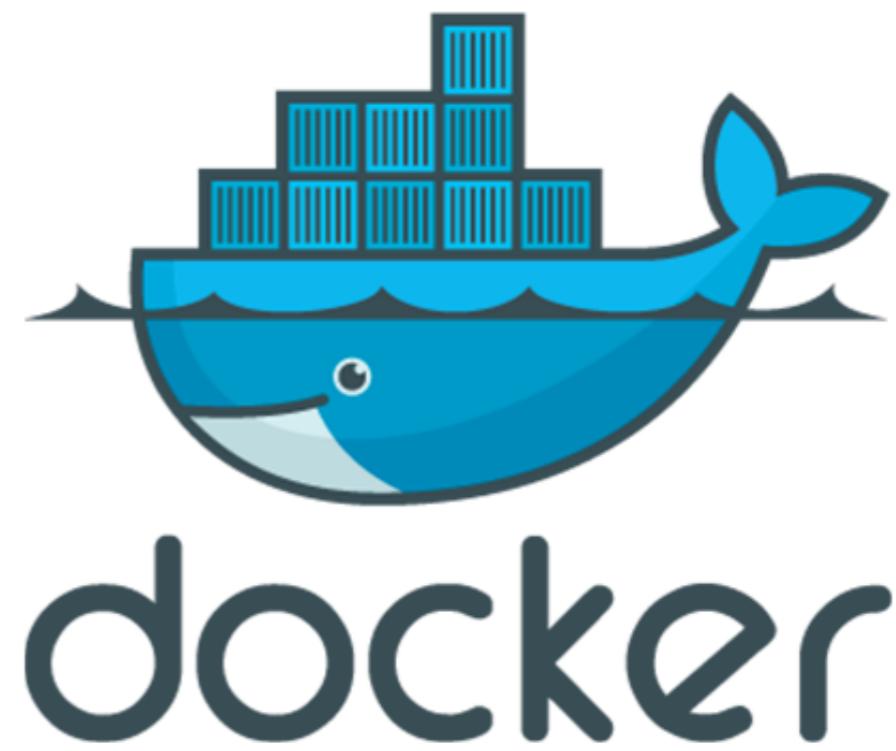
## apache zookeeper



- DISTRIBUTED LOCKING
- SERVICE MANAGEMENT  
(LOAD BALANCER, DNS ...)



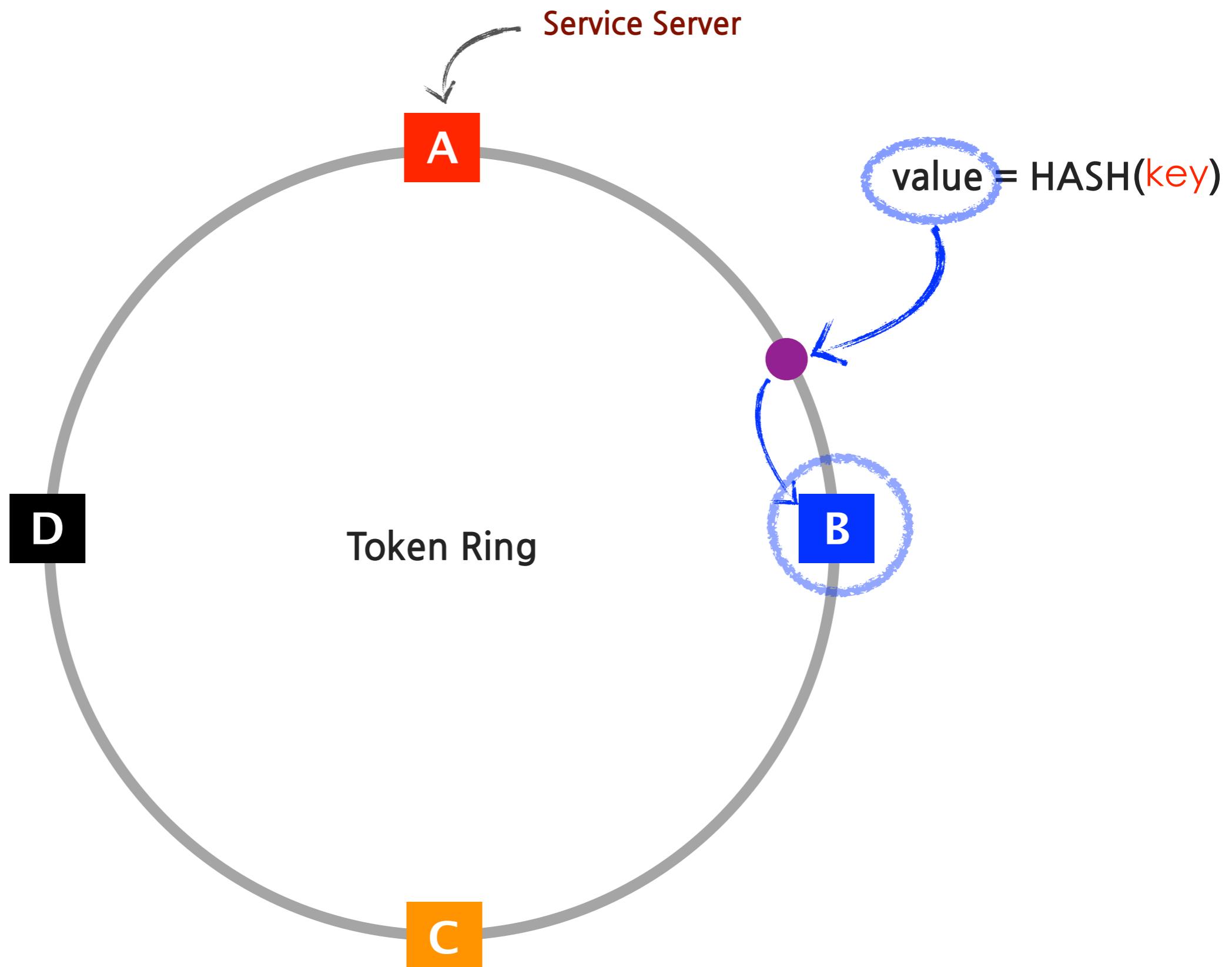
Apache ZooKeeper™



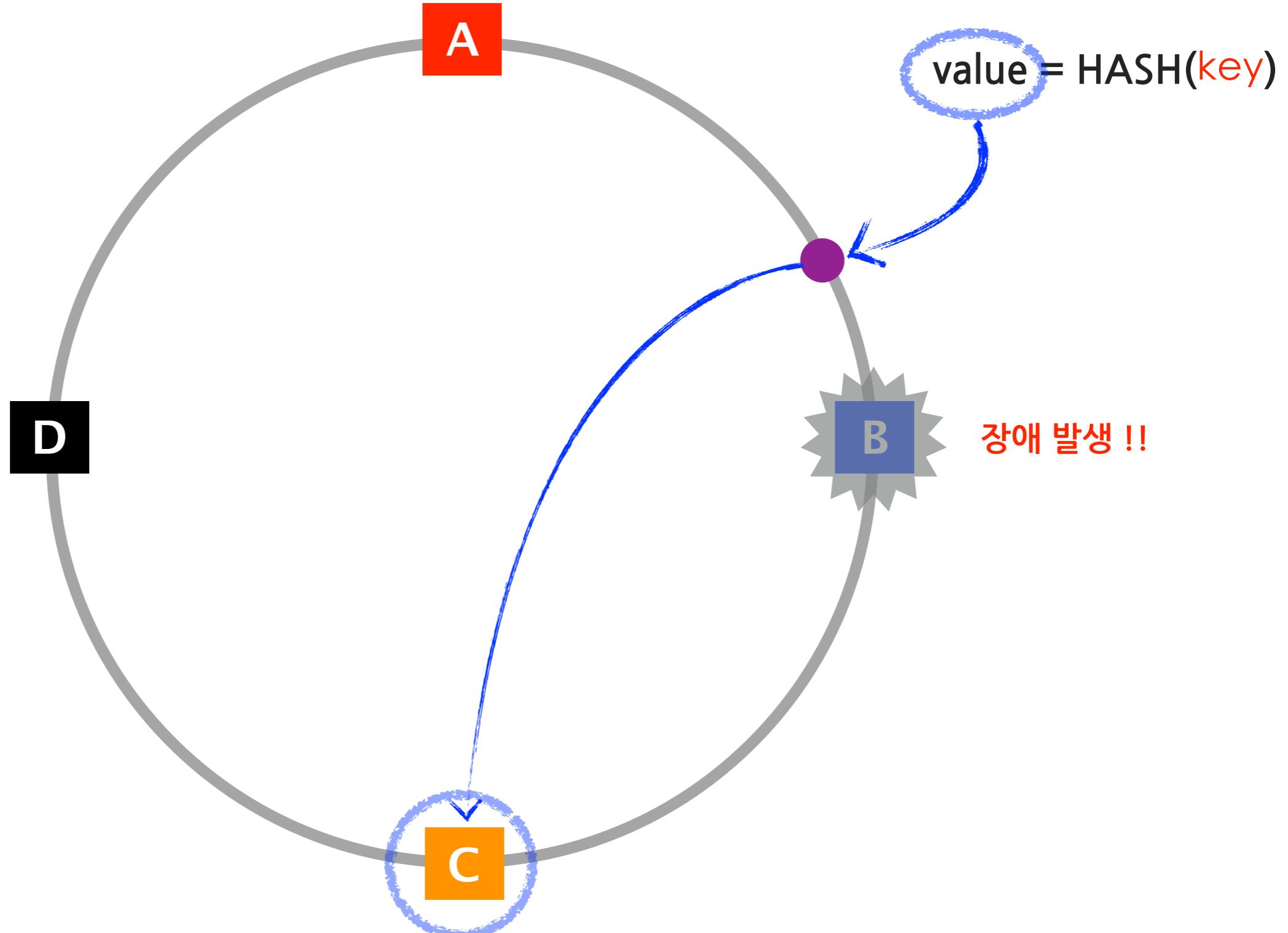
# 서버 할당은 어떻게 하는가 ?

SESSION 서버는 어떤 CHANNEL 서버를 할당해 줘야 하는지 ?

# CONSISTENT HASHING

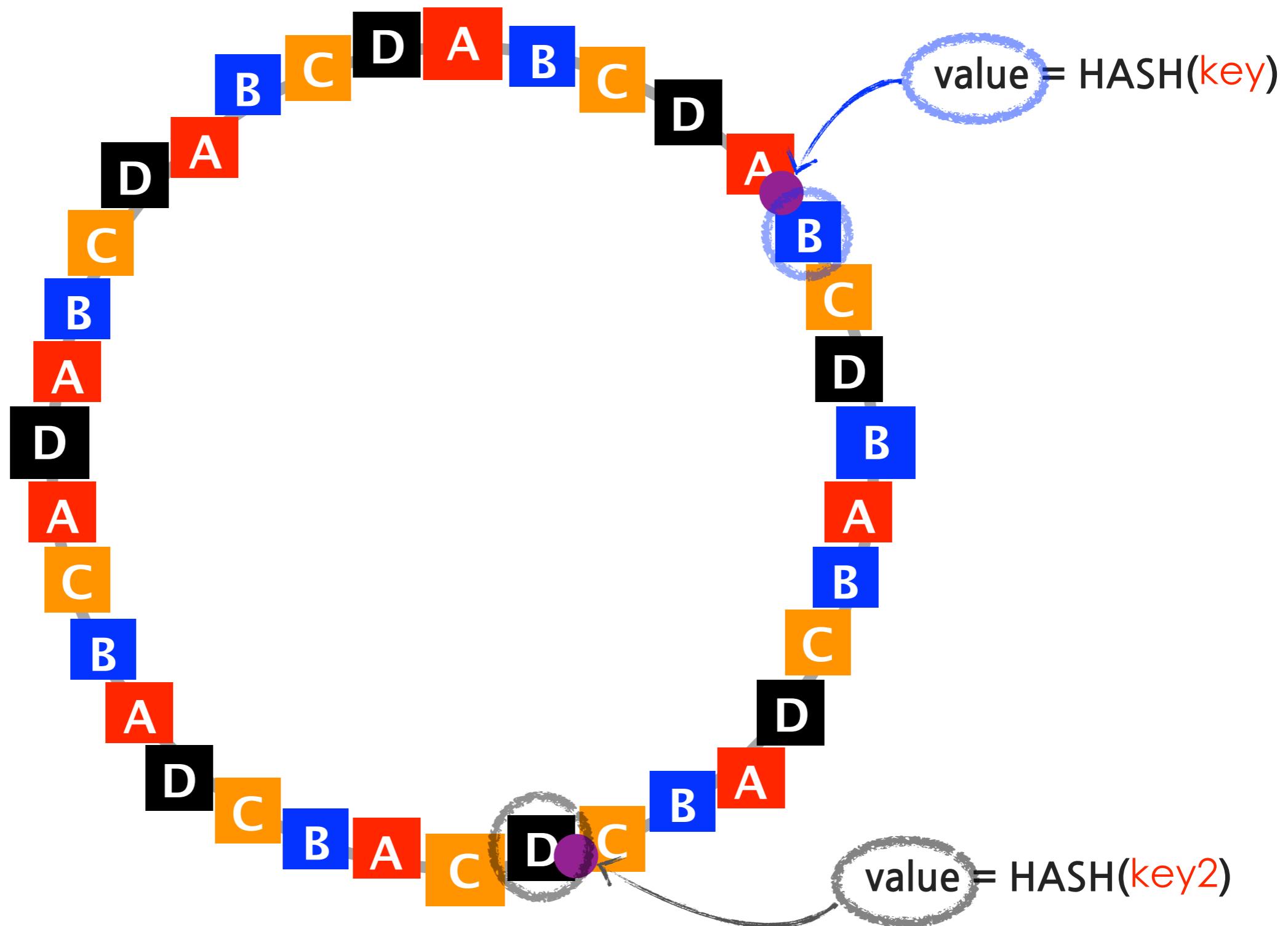


# CONSISTENT HASHING

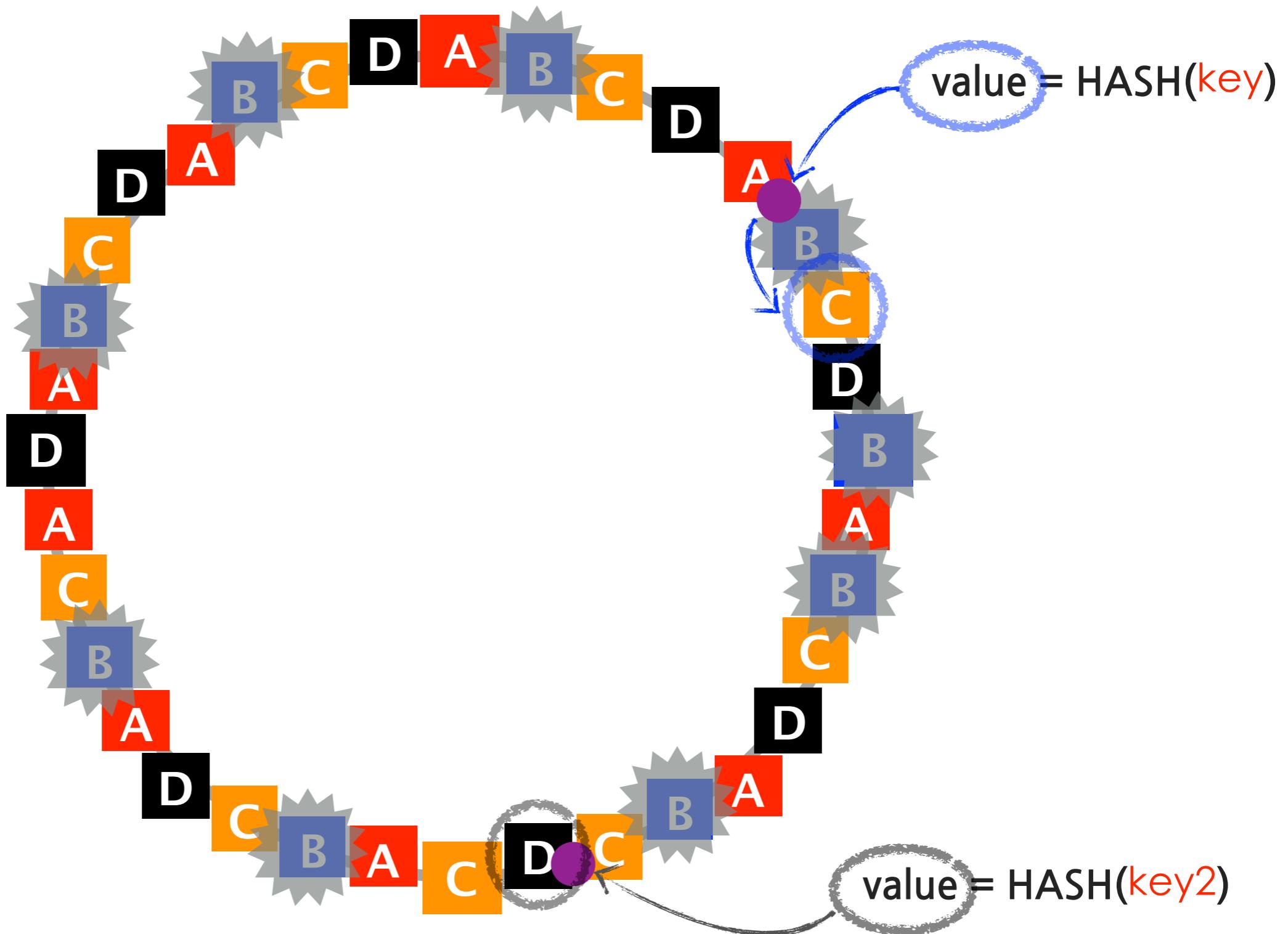


A - C 구간이 넓어서 C에 부하가 많을 수 있다 !

# CONSISTENT HASHING

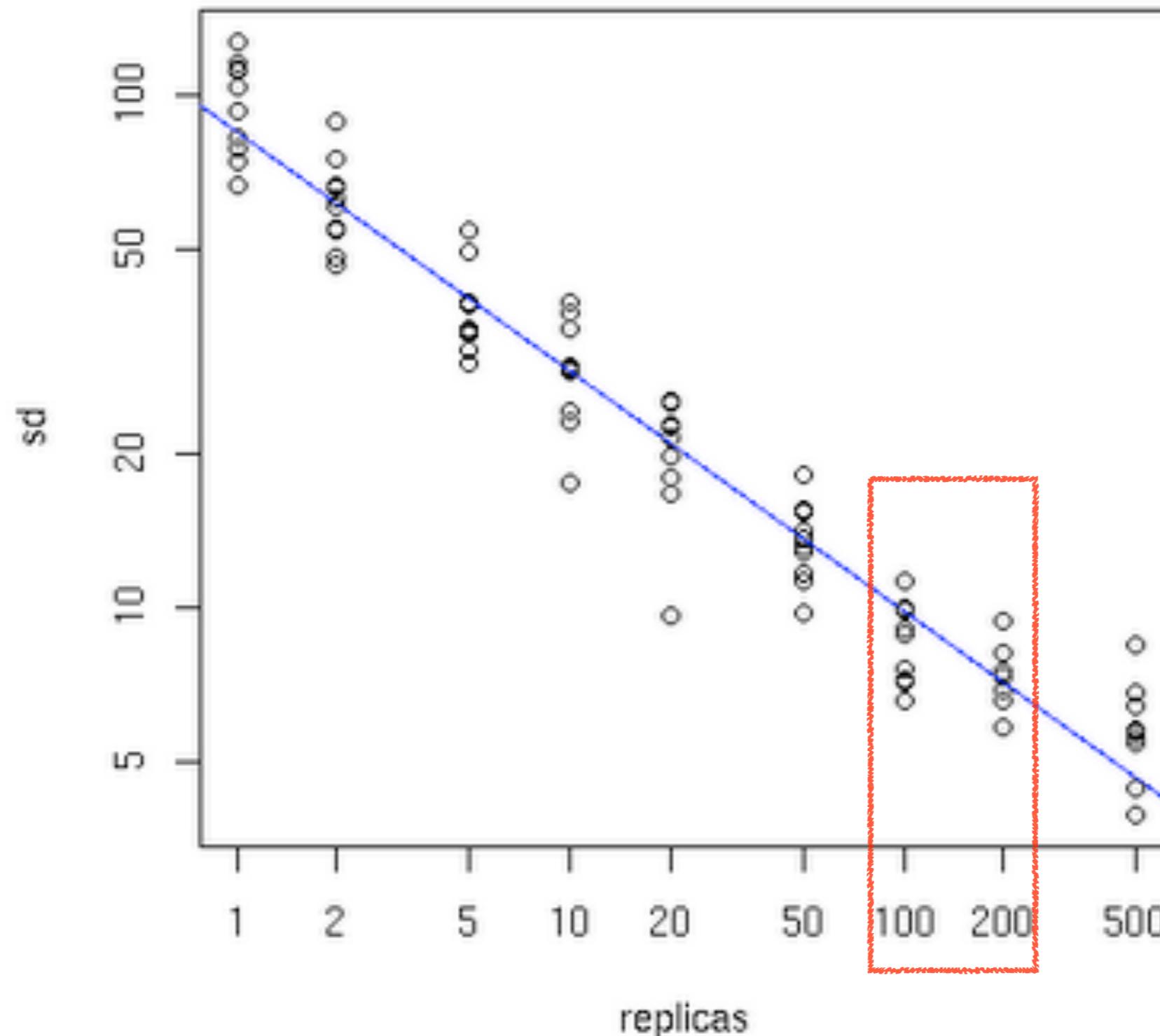


# CONSISTENT HASHING



# CONSISTENT HASHING

replicas 수에 따른 표준편차(standard deviation)



서버 노드 10 개를 1 부터 500 개 까지 replicas 로 배치 10,000 번의 consistent Hashing 결과

[https://weblogs.java.net/blog/tomwhite/archive/2007/11/consistent\\_hash.html](https://weblogs.java.net/blog/tomwhite/archive/2007/11/consistent_hash.html)

# CONSISTENT HASHING

<https://github.com/3rd-Eden/node-hashring>

```
var HashRing = require('hashring');

var ring = new HashRing([
  '127.0.0.1',
  '127.0.0.2',
  '127.0.0.3',
  '127.0.0.4'
]);

ring.get('채팅방키'); // 접속 서버 반환
```

# Weighted CONSISTENT HASHING

<https://github.com/3rd-Eden/node-hashring>

```
var HashRing = require('hashring');

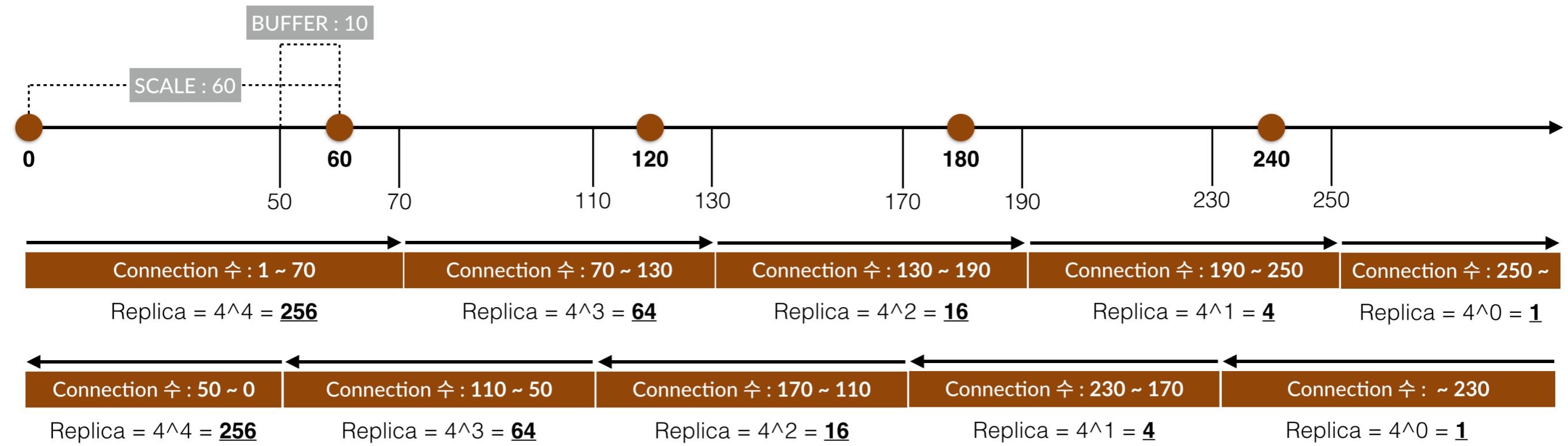
var ring = new HashRing({
  '127.0.0.1': { weight: 200 },
  '127.0.0.2': { weight: 200 },
  '127.0.0.3': { weight: 200 },
  '127.0.0.4': { weight: 1000 }
});

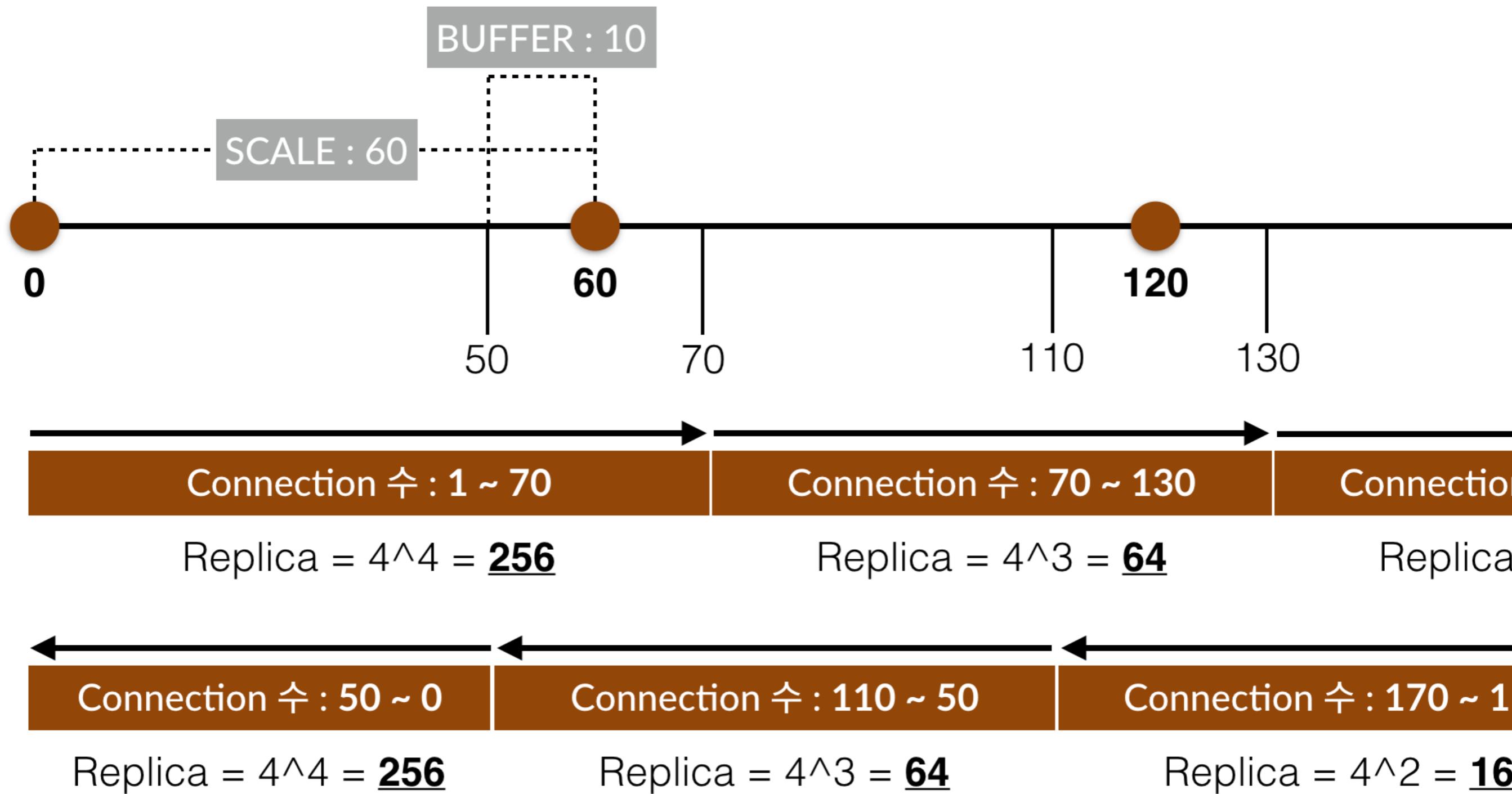
ring.get('채팅방키'); // 접속 서버 반환
```

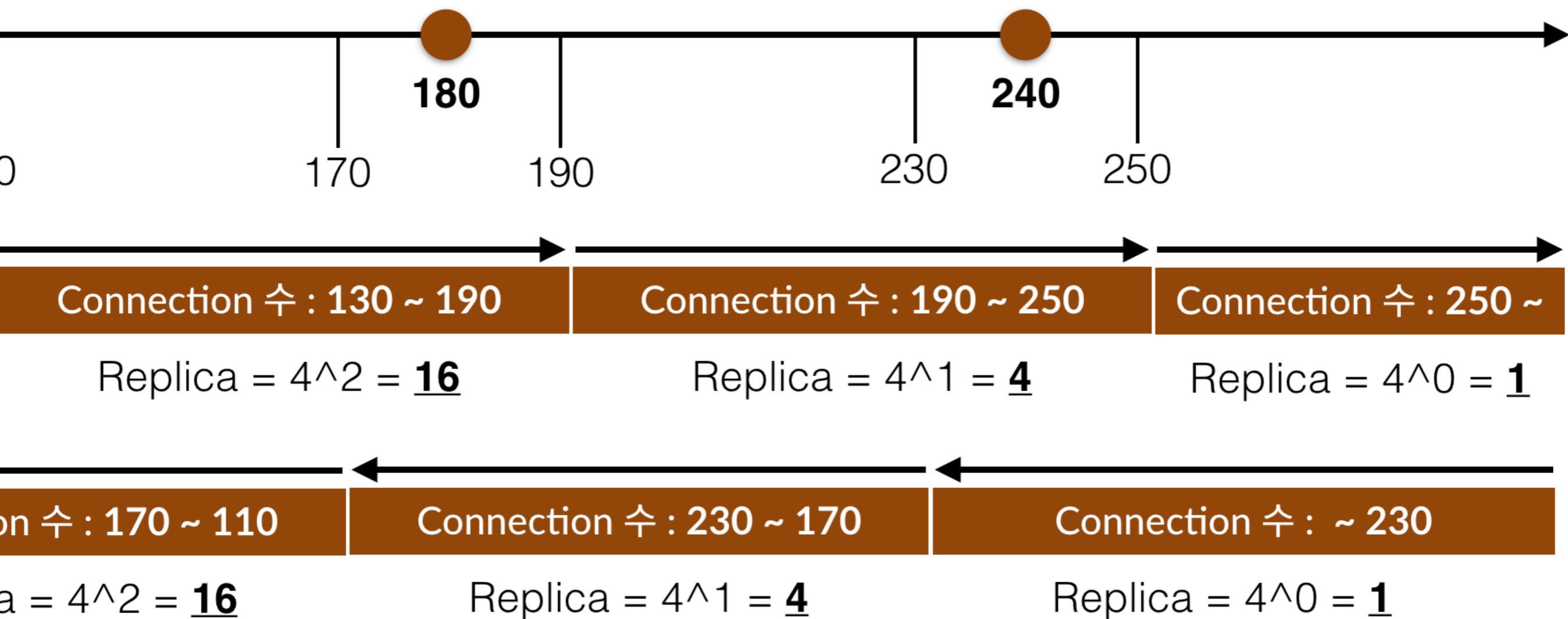
좀 더 잘하기 위해서 . . . . .

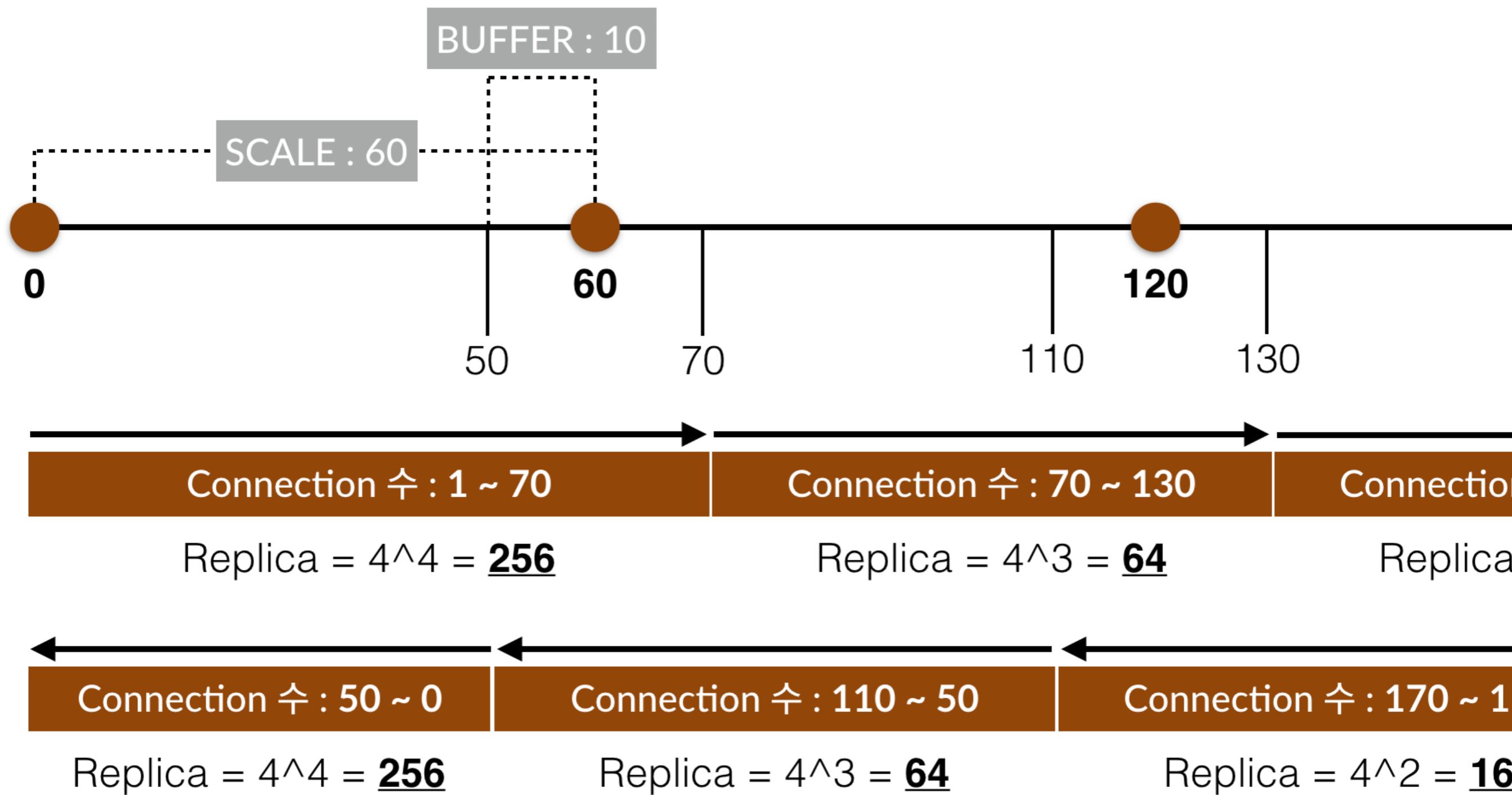
## 실시간 Weight 를 수정한다.

서버 접속자 수에 따라 실시간으로 replica 수를 조정한다.









# 죽으면 어때 다시 시작하면 되지. . .

모든게 말처럼 간단하면 얼마나 좋겠냐 ?

Node restarts so quickly  
crashed aren't a big deal.

# 프로세스 모니터링 데몬을 실행한다.

부모 프로세스와 자식 프로세스의 관계를 끊는다! 같이 죽으면 안되도록...

```
var spawn = require('child_process').spawn;
```

```
spawn(process.execPath, [monitorFilePath], {  
  stdio: ['ignore', out, err],  
  detached: true,  
  env: {  
    X_PID: process.pid  
  }  
}).unref();
```

# 프로세스 살아 있는지 체크한다.

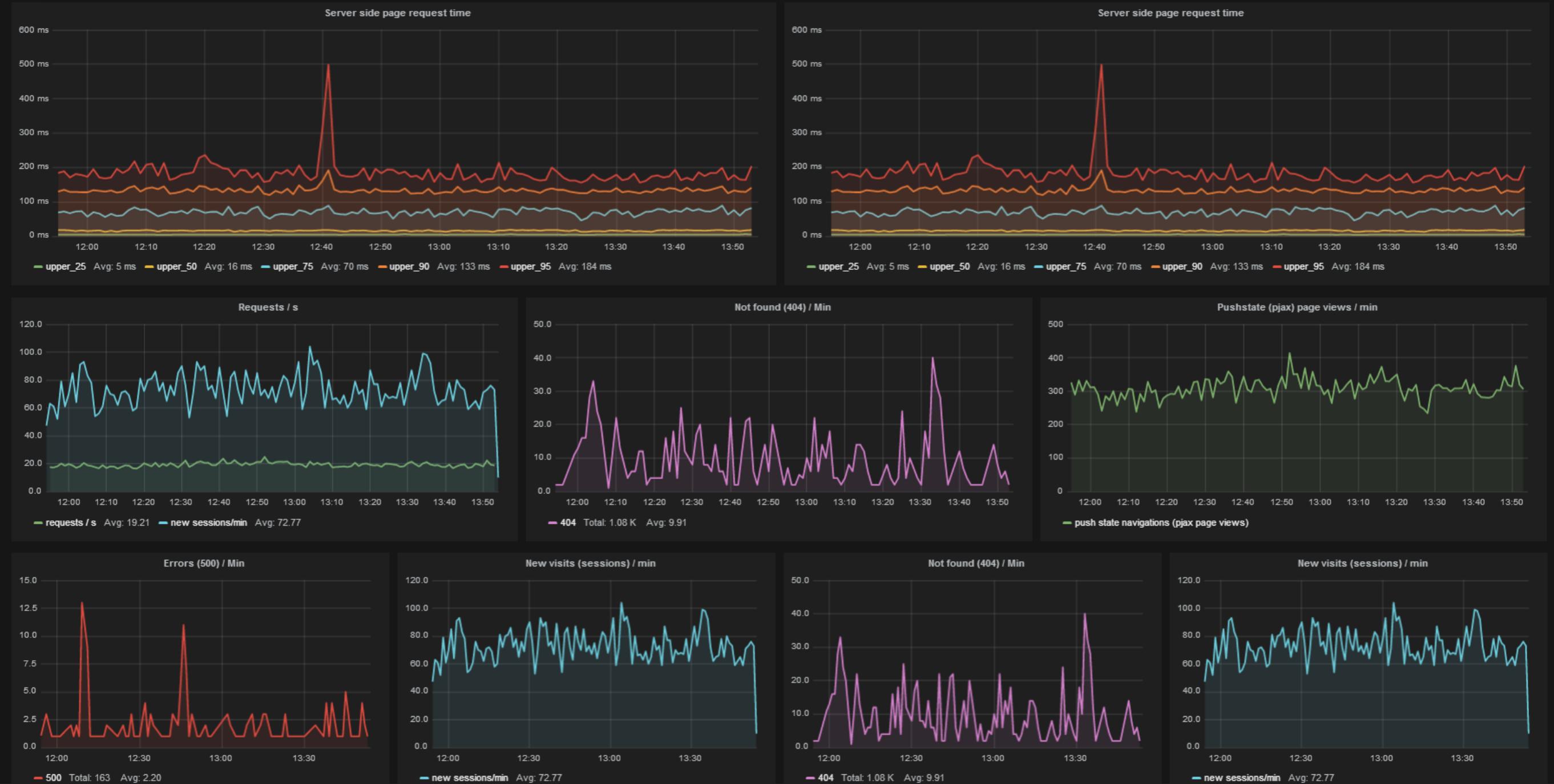
spawn 으로 실행한 데몬에서 PID 로 “0” 신호 보내기.

```
var isRunning = function (pid) {  
  try {  
    return process.kill(pid, 0);  
  }  
  catch (e) {  
    return e.code === 'EPERM'  
  }  
};
```

모니터링 해보자.



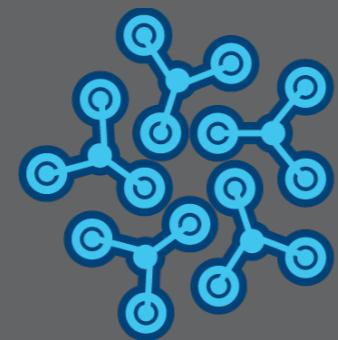
Grafana is a **frontend** for Graphite, InfluxDB and OpenTSDB with powerfull visualization features for time series data



Graph General Metrics Axes & Grid Display Styles

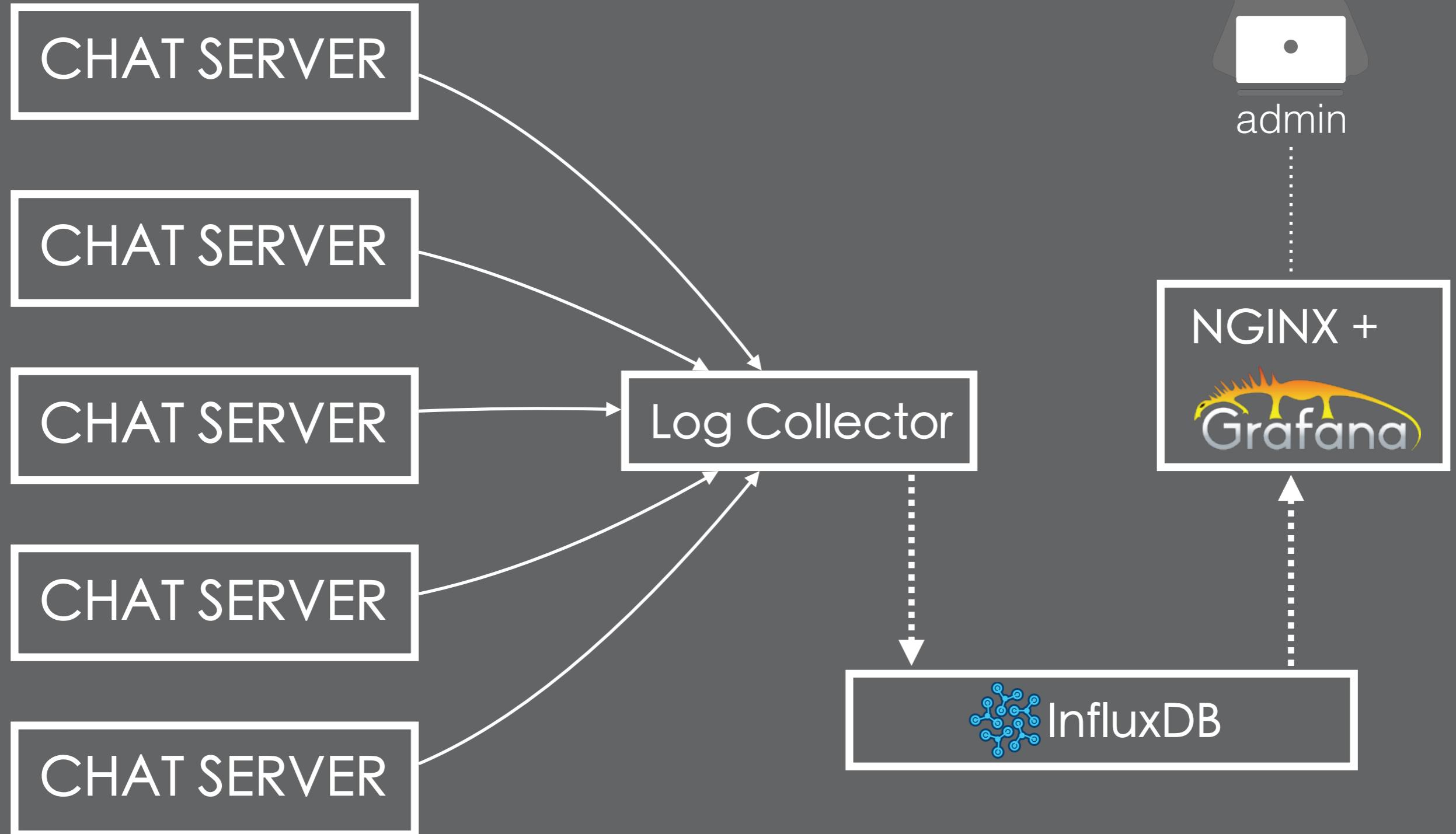
<span>eye</span>	series	/apps.\$app.\$server.counters.requests.count/	alias	\$1.\$2	<span>gear</span>	<span>cross</span>
select	mean(value)	where column ~= value	group by time	10m	<span>+</span>	column fill (0)

<span>key</span>	group by time	>10m	<span>?</span>				
<span>info</span>	alias patterns	stacking & and fill	group by time				



# InfluxDB

distributed, time series database  
**with no external dependencies**



CHAT SERVER

## Socket.IO

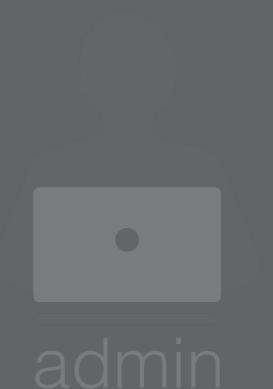
(separated namespace for monitoring)

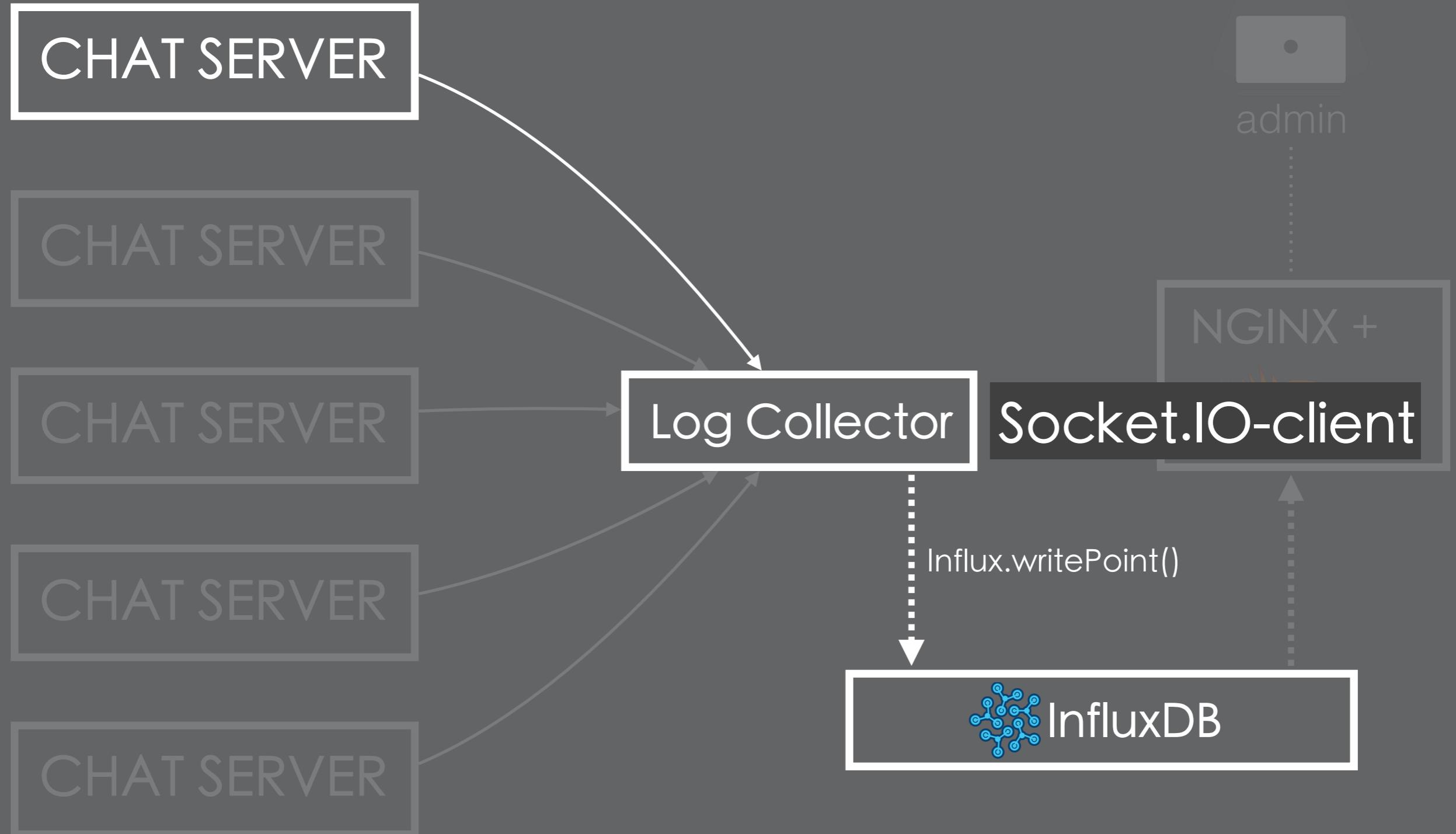
```
{  
  pid : process.pid,  
  arch : process.arch,  
  platform : process.platform,  
  uptime : process.uptime(),  
  memory : process.memoryUsage(),  
  
  // 그 외  
  // 서버에 연결된 Client 수  
  // 체팅방 수 등  
}
```

CHAT SERVER

CHAT SERVER

CHAT SERVER





## CHAT SERVER

```
// nginx 설정
```

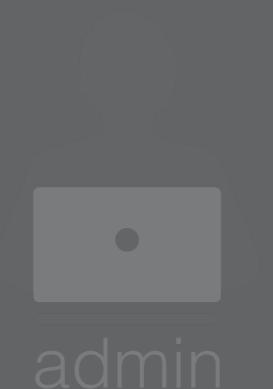
```
auth_basic "Restricted";  
auth_basic_user_file /path/to/my/htpasswd/file;
```

## CHAT SERVER

## Log Collector

## CHAT SERVER

## CHAT SERVER



# wrap up.

## 1. CPU Intensive Task 는 잘 생각해봐야 한다.

- CPU core 수 만큼 실행하자.
- Message 송수신하는 작업은 딱! 좋은 적용 대상이다.

## 2. 개발 시작부터 Scalability 를 고려해야 한다.

- 서버 정보는 zookeeper (분산 코디네이터) 에 등록해서 관리하자.
- 서버간 데이터 공유는 in-memory DB 또는 Data Grid 를 활용하자.
- Consistent Hashing 과 같은 알고리즘 정도는 알아두자.  
(Weighted 도 중요하다.)

## 3. 효율적인 모니터링 방법을 고민해야 한다.

서버의 uptime 이 중요하다.

프로세스를 모니터링하는 데몬도 고려해 보자.

**개발 중 오픈소스 프로젝트 . . .**



## SUPER EASY INSTALLATION

Very simple scripts that you can add to your Web pages in minutes, no knowing coding required. Copy and paste just tow line of code at the bottom of your body tag. That's it!

Put the tags and the javascript codes before the last </body> tag in your HTML page

```
<script src="http://stalk.io/stalk.js"></script>
<script>STALK.init();</script>
```

"STALK.IO" is the most beautiful and effective way to make visitors stay on your website. **Just add the code to your website's header or footer.** It's super easy to install this chat widget for your website.

You can install this right where you want it. For **Wordpress, Tumblr, Blogger and your A-Z website**.

ONLINE : 4 X

Welcome ! Try logging on for chatting.

야~ 호~ 이거 정말 쉽게 설치 가능한 체팅 위젯이군요!! ㅎㅎㅎㅎㅎ

John Kim  
으엥?? 뭉미? 이거 테그만 넣으면 되는 거임?

notdol P  
그런거 같은데용?

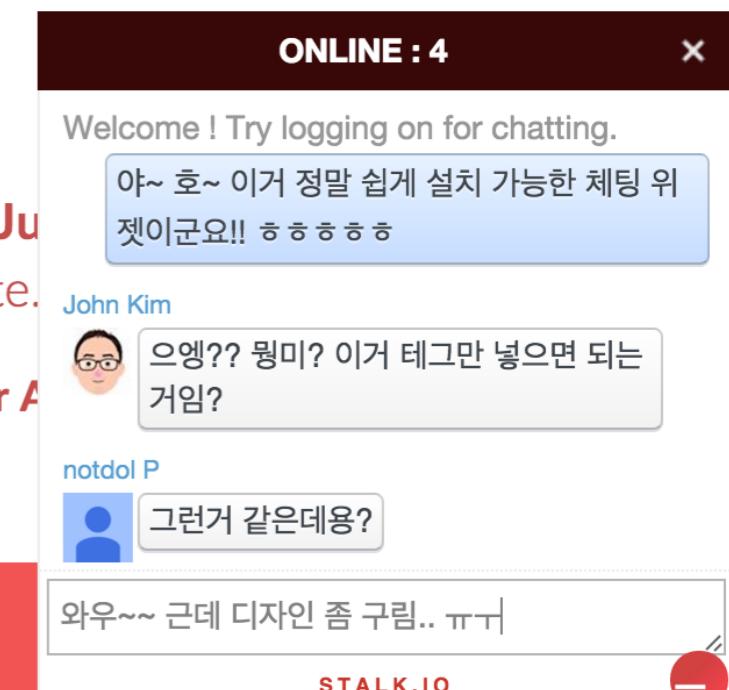
와우~~ 근데 디자인 좀 구림.. ㅠㅠ

page

```
<script src="http://stalk.io/stalk.js"></script>
<script>STALK.init();</script>
```

"STALK.IO" is the most beautiful and effective way to make visitors stay on your website. **Just page!** It's super easy to install this chat widget for your website.

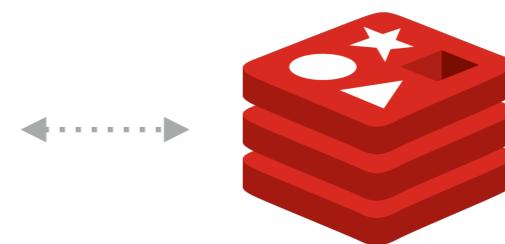
You can install this right where you want it. For **Wordpress, Tumblr, Blogger and your A**



<http://simple.stalk.io>



**XPUSH** - session server & channel server  
<https://github.com/xpush/node-xpush>



13:30

**Node.js 기반의 대용량 분산 서버 아키텍처 설계 사례**

김요한@GS SHOP

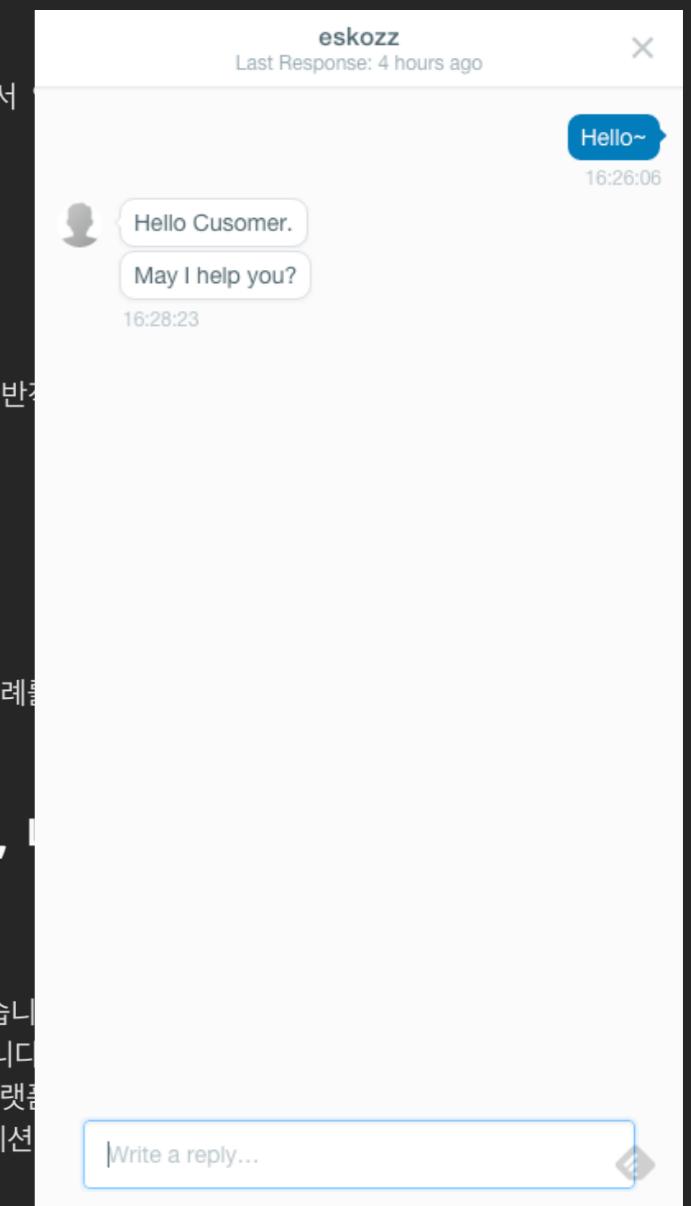
Node.js 기반의 주로 어떤 시스템이 적합할까? 무중단 Scalability 를 고려하려면 어떻게 해야할까? 실제 구축 사례를 기반으로 실무에 활용할 수 있는 방법들을 공유하고자 합니다. 실제 프로젝트나 서비스 구축시 Node.js 와 다양한 오픈소스의 조합을 통해 대용량 트래픽을 처리하는데 도움이 되고자 합니다.

14:20

**시간당 수백만 요청을 처리하는 node.js 서버 운영기**

김군우@Dable

Amazon Web Service에서 node.js를 이용하여 시간당 수백만 요청을 처리하는 서버를 운영하면서 경험한 문제, 해결법, 느낀점 등을 공유하려 합니다.



15:10

**Node.js in Flitto**

이승우@Flitto

집단지성 번역 플랫폼 Flitto(플리토)는 Node.js로 서버, 웹 개발을 하고 있습니다. Flitto 서버의 전반적인 구조와 웹 애플리케이션 개발하면서 경험했던 시행착오를 공유합니다.

16:00

**스프링과 Node.js의 공존**

장동수@카카오

자바 기반의 레거시 웹 서비스를 마이크로서비스 아키텍처와 node.js로 현대화 하는 과정을 실제 사례를 통해 살펴보겠습니다.

16:50

**Express, Angular.js를 이용해서 Big Data 세계를 헤엄친 이야기, 그리고 그 경험을 반복하지 마세요**

이병준@SK planet

자바스크립트는 TIOBE Index에서 2015년 10월 기준 8위를 차지할 정도로 많은 성장을 거듭하였습니다. Node.js는 특성상 Big Data 세계에서는 그 사용례가 그리 많지 않습니다. 이유로 인해서 Hue를 사용할 수 없게 되어 Galleon이라는 대체할 플랫폼을 개발하기 위해 주력 플랫폼으로 선택하여 겪은 경험담을 공유하려고 합니다. - Node.js에서 Hive를 다루어보고 싶은 분이라면 이번 세션을 꼭 놓치지 마세요.

Write a reply...



eskuzz

Dashboard



Live stalk

eskuzz

notdol

STALK.IO : real communi...

## stalk with customers

STALK.IO : real communication service

STALK.IO : real communication service

3 - x

16:22:20



HI

16:22:37

eskuzz

Hello. Customer



16:22:44

eskuzz

May I help you?



Type Message ...

Send

## Customer Info

Started : 오후 4:22:20

URL : http://static.stalk.io/#

Path :

OS : MacOS (Unknown)

Browser : chrome

IP : 192.168.100.103

## Customer Location



+

-

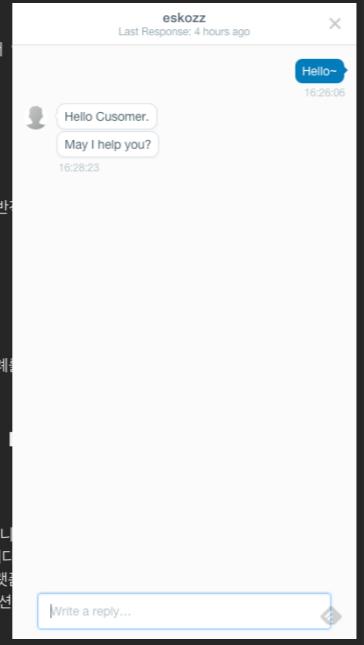
<https://github.com/xpush/io.stalk.static>

Node.js 기반의 주로 어떤 시스템이 적합할까? 무중단 Scalability 를 고려하려면 어떻게 해야할까? 실제 구축 사례를 기반으로 실무에 활용할 수 있는 방법들을 공유하고자 합니다. 실제 프로젝트나 서비스 구축시 Node.js 와 다양한 오픈소스의 조합을 통해 대용량 트래픽을 처리하는데 도움이 되고자 합니다.

#### 14:20 시간당 수백만 요청을 처리하는 node.js 서버 운영기

김군우@Dable

Amazon Web Service에서 node.js를 이용하여 시간당 수백만 요청을 처리하는 서버를 운영하면서 기, 느낀점 등을 공유하려 합니다.



#### 15:10 Node.js in Flitto

이승우@Flitto

집단지성 번역 플랫폼 Flitto(플리토)는 Node.js로 서버, 웹 개발을 하고 있습니다. Flitto 서버의 전반 개발하면서 경험했던 시행착오를 공유합니다.

#### 16:00 스프링과 Node.js의 공존

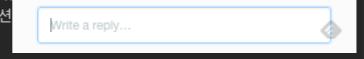
장동수@카카오

자바 기반의 레거시 웹 서비스를 마이크로서비스 아키텍처와 node.js로 현대화 하는 과정을 실제 사례로 공유합니다.

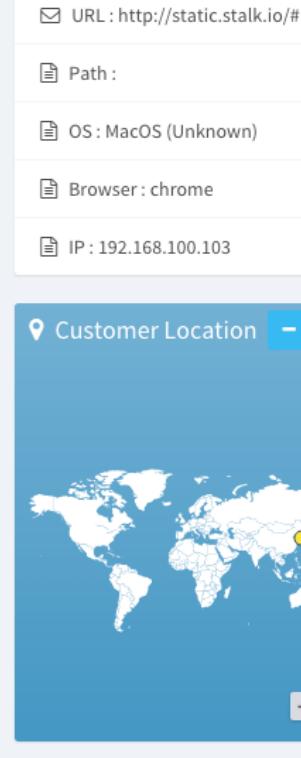
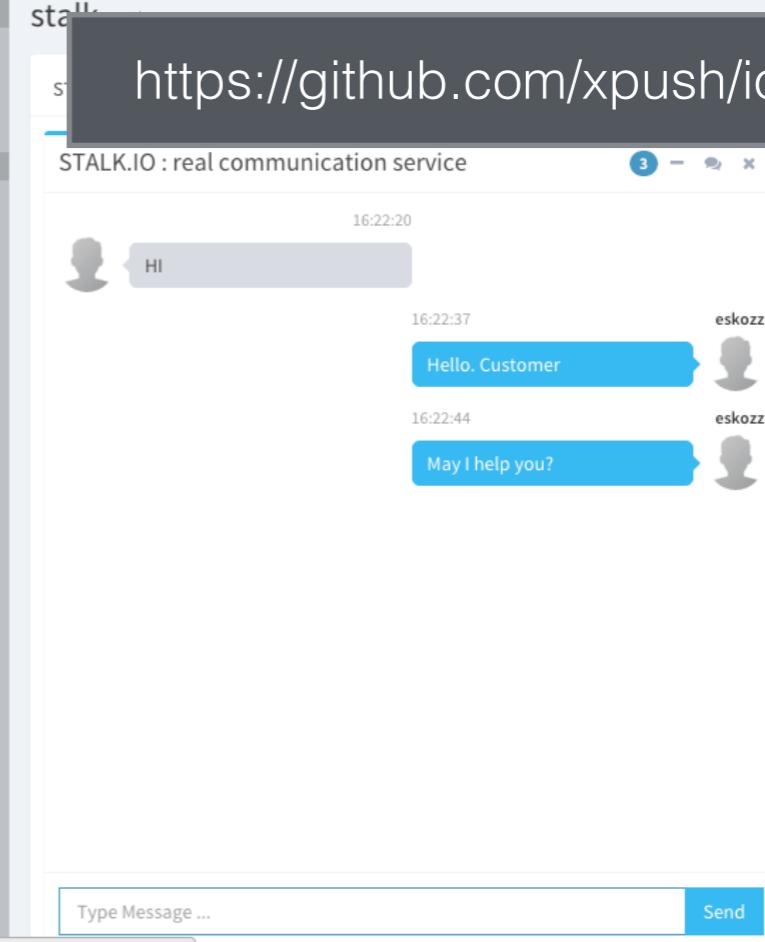
#### 16:50 Express, Angular.js를 이용해서 Big Data 세계를 헤엄친 이야기, Node.js를 반복하지 마세요

이병준@SK planet

자바스크립트는 TIOBE Index에서 2015년 10월 기준 8위를 차지할 정도로 많은 성장을 거듭하였습니다. 하지만 아무래도 Node.js는 특성상 Big Data 세계에서는 그 사용 레가 그리 많지 않습니다. 이유로 인해서 Hue를 사용할 수 있게 되어 Galleon이라는 대체할 플랫폼을 개발하기 위해 주력 플랫폼으로 탐색하여 겪은 경험담을 공유하려고 합니다. - Node.js에서 Hive를 다루어보고 싶은 분이라면 이번 세션에 참석해 주세요.



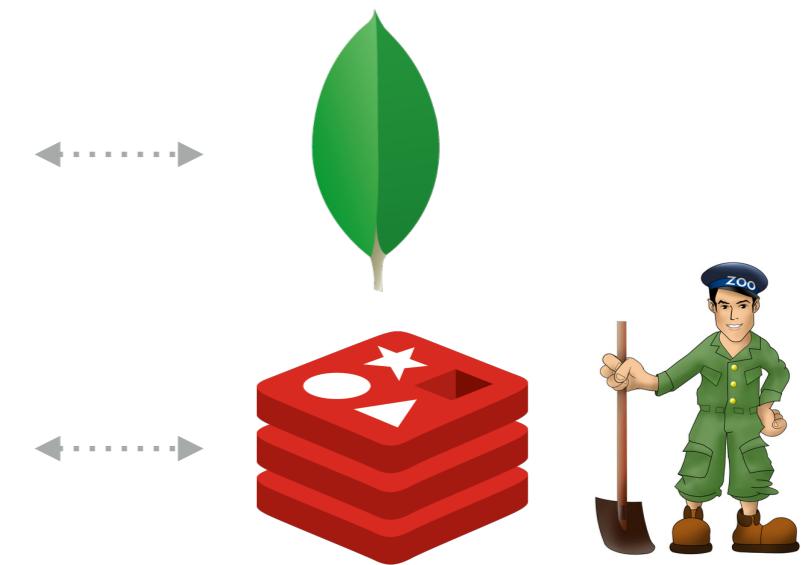
LIVE STALK  
eskuzz  
notdol  
STALK.IO : real communication service



require('xpush')  
**XPUSH-CHAT**

session server & channel server

<https://github.com/xpush/xpush-chat>





helps your internet business to communicate with customers on your website.

<http://stalk.io>

# AngularJS Full-Stack generator

<https://github.com/DaftMonk/generator-angular-fullstack>

**lets you quickly set up a project following best practices**

**Yeoman generator for creating MEAN stack applications,  
using MongoDB, Express, AngularJS, and Node.**

# AdminLTE

<https://github.com/almasaeed2010/AdminLTE>

The screenshot displays the AdminLTE dashboard interface. On the left, a dark sidebar contains a user profile for "Alexander Pierce" (online), a search bar, and a main navigation menu with items like Dashboard, Layout Options, Widgets, Charts, UI Elements, Forms, Tables, Calendar, Mailbox, Examples, and Multilevel.

The main content area features a header with the title "Dashboard Version 2.0" and a user icon for "Alexander Pierce". It includes four primary widgets: "CPU Traffic" (90%, blue icon), "Likes" (41,410, red icon), "Sales" (760, green icon), and "New Members" (2,000, orange icon).

Below these are two sections: "Monthly Recap Report" and "Goal Completion". The "Monthly Recap Report" section contains a line chart titled "Sales: 1 Jan, 2014 - 30 Jul, 2014" showing sales trends from January to July. It also displays three summary statistics: "TOTAL REVENUE" (\$35,210.43, up 17%), "TOTAL COST" (\$10,390.90, down 10%), and "TOTAL PROFIT" (\$24,813.53, up 20%). The "Goal Completion" section lists four goals with progress bars: "Add Products to Cart" (160/200), "Complete Purchase" (310/400), "Visit Premium Page" (480/800), and "Send Inquiries" (250/500).

At the bottom, there's a "Visitors Report" section showing a world map and a summary card with "Inventory 5,200" and "50% Increase in 30 Days".

# 개발자에게 개발환경을 강요하는건 안될 일이지만..

**NVM** (Node Version Manager)

<https://github.com/creationix/nvm>

cf. rbenv, pyenv



<https://www.jetbrains.com/editors/nodejs.jsp?ide=idea>

# .gitignore

<https://www.gitignore.io/>

## •gitignore.io

Create useful .gitignore files for your project

x Node

Generate

[Command Line Docs](#) | [Watch Video Tutorial](#)

# .editorconfig

<http://editorconfig.org>

```
root = true
```

```
[*]
```

```
indent_style = space
indent_size = 2
```

```
end_of_line = lf
charset = utf-8
trim_trailing_whitespace = true
insert_final_newline = true
```

```
[*.md]
trim_trailing_whitespace = false
```



2015년 공개SW개발자센터 컨퍼런스

감사합니다.