# EventLoop을 이용한 WebServer 구현

2018.04.27

19조 Tmax

경영학부 박성우

컴퓨터공학부 데이비드

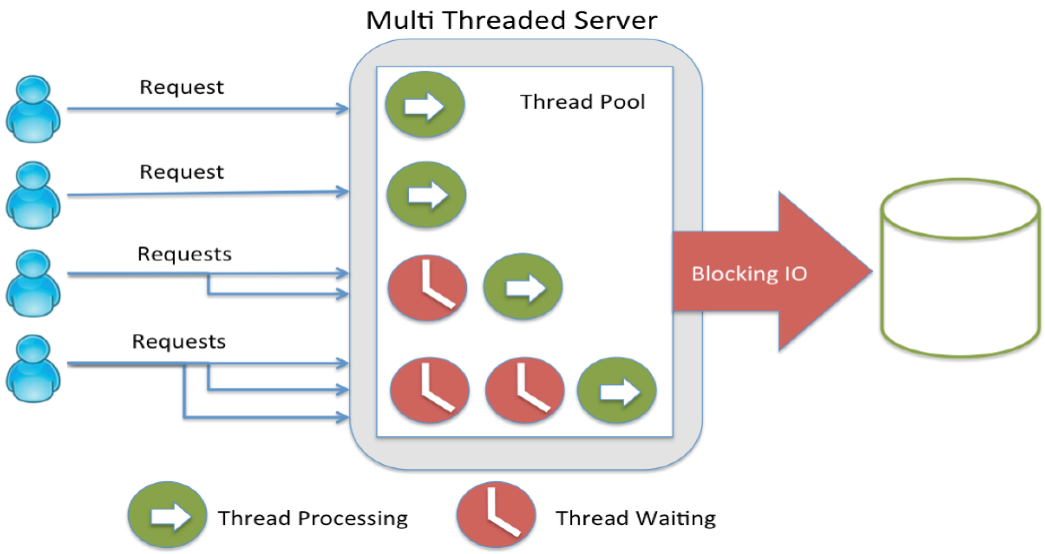
컴퓨터공학부 이승현

# Contents

# Problem Raised

✔ Vast concurrent connections to web server

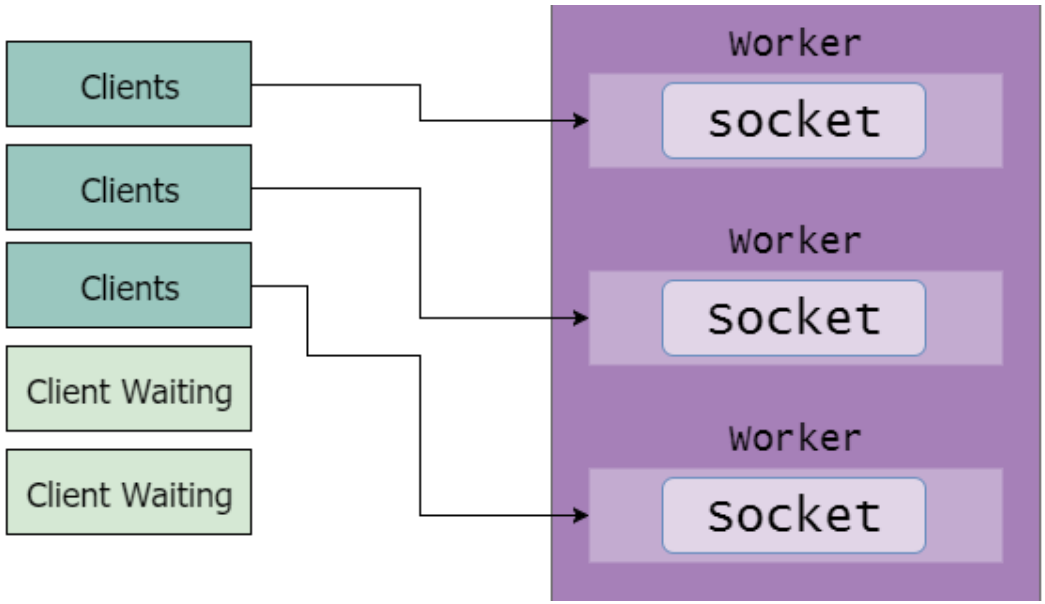
- 얼마나 많은 클라이언트를 동시에 처리할 수 있는가?
- 다량의 socket이 연결될 때 하드웨어가 충분하더라도 I/O 처리방식 때문에 서버가 제대로 동작하지 않을 수 있다.



# Existing Problems

## ✔ Limits of connection oriented model

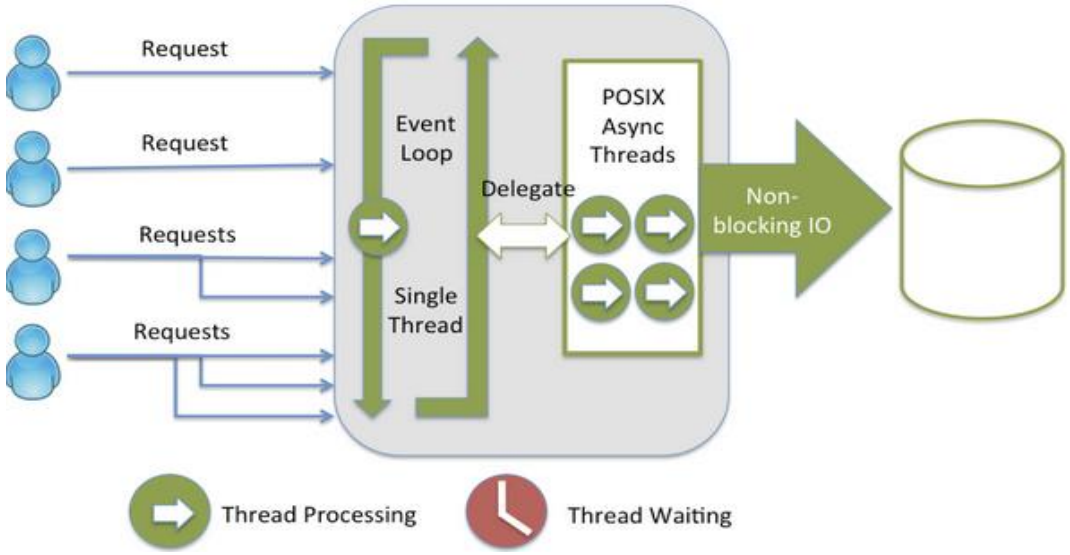
- Thread pool 크기만큼의 client만을 동시에 처리 가능
- Blocking I/O로 인한 저조한 CPU utilization
- Context switch overhead로 인해 performance 저하



# Approach

## ✔ Improvements with single threaded model

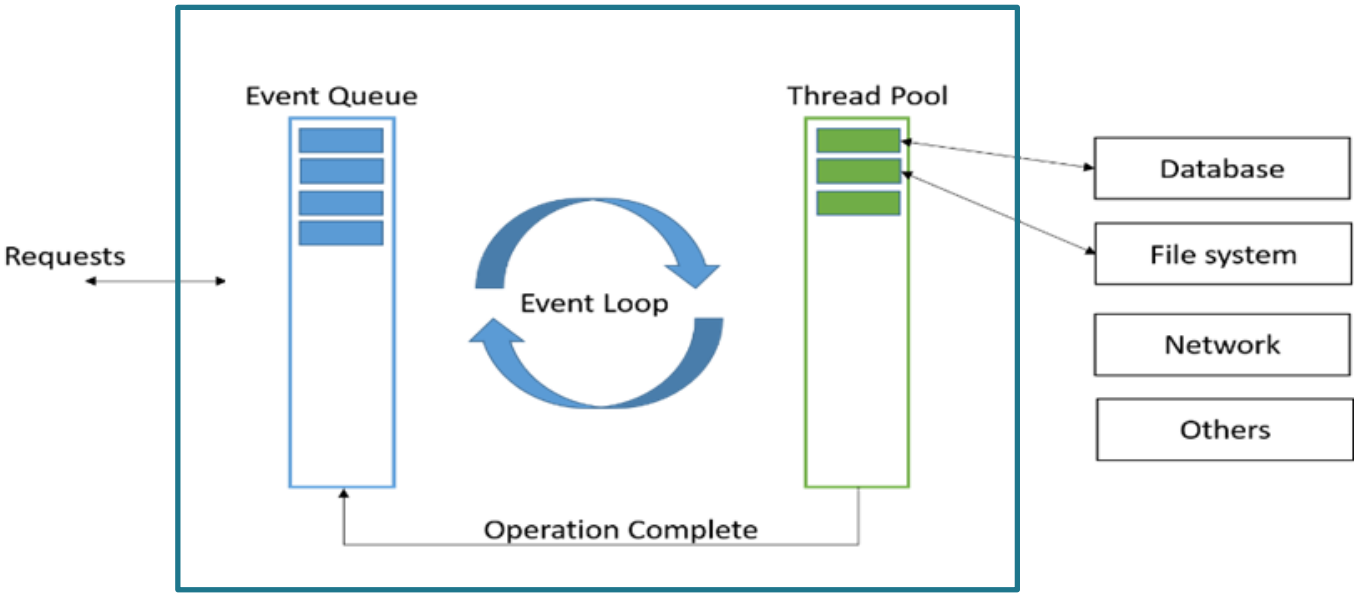
- 하나의 event loop thread가 infinite loop를 돌면서 request를 전담
- Block을 유발하는 I/O는 worker thread가 대신 처리
- Context switch 없이 infinite loop를 돌면서 callback을 받아서 처리



# Architecture(1)

## ✔ Components of single threaded model

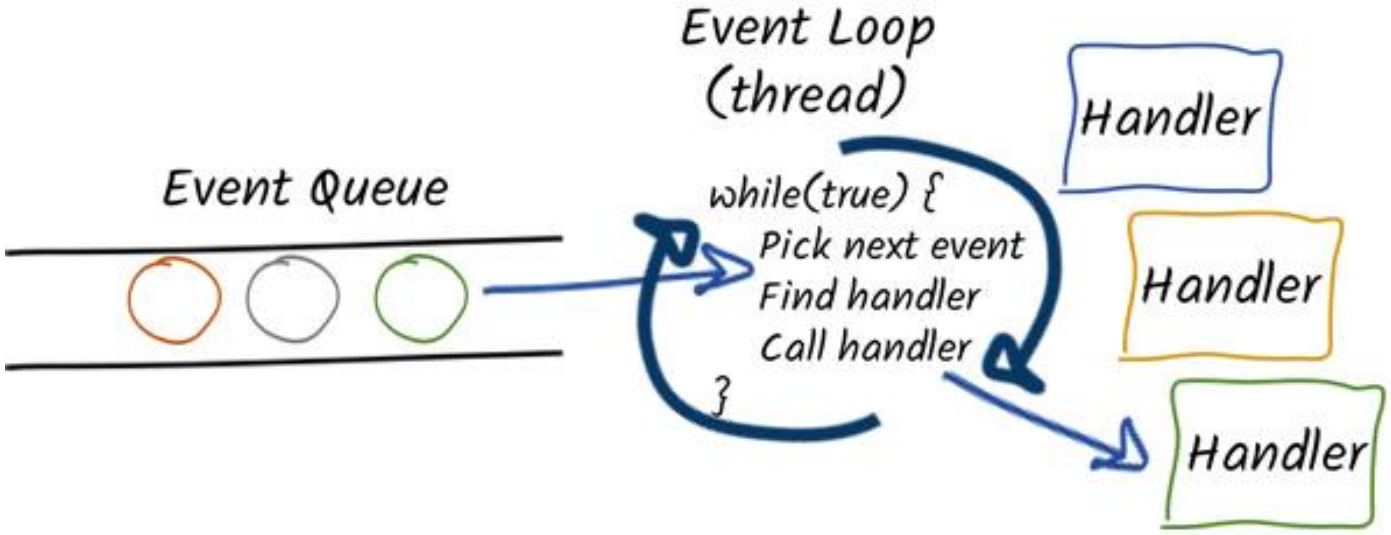
- Event queue ≒ to-do list
- Single event loop thread ≒ the exploited
- Internal multi thread pool ≒ helpers



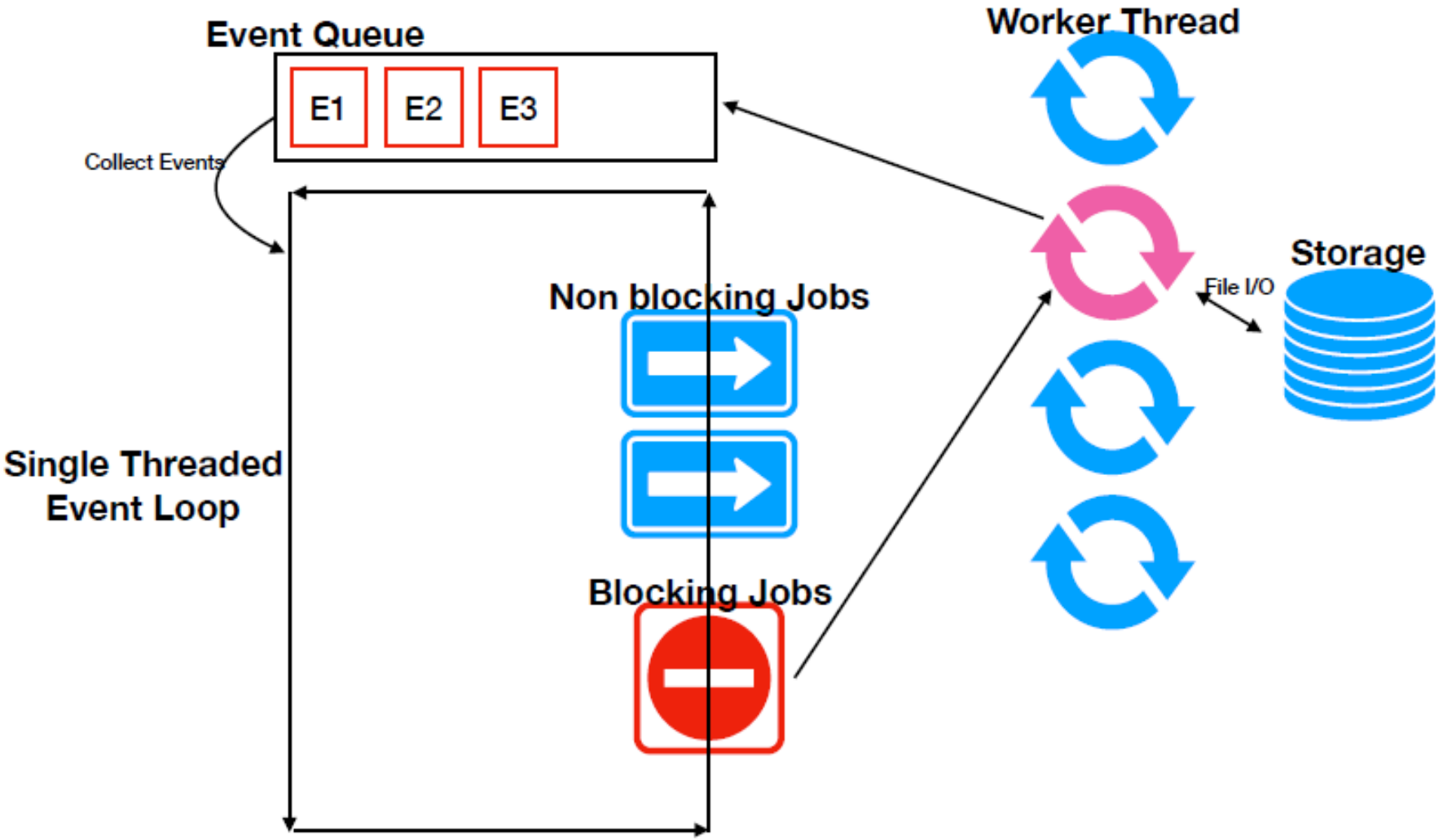
# Architecture(2)

## ✔ Event loop based architecture

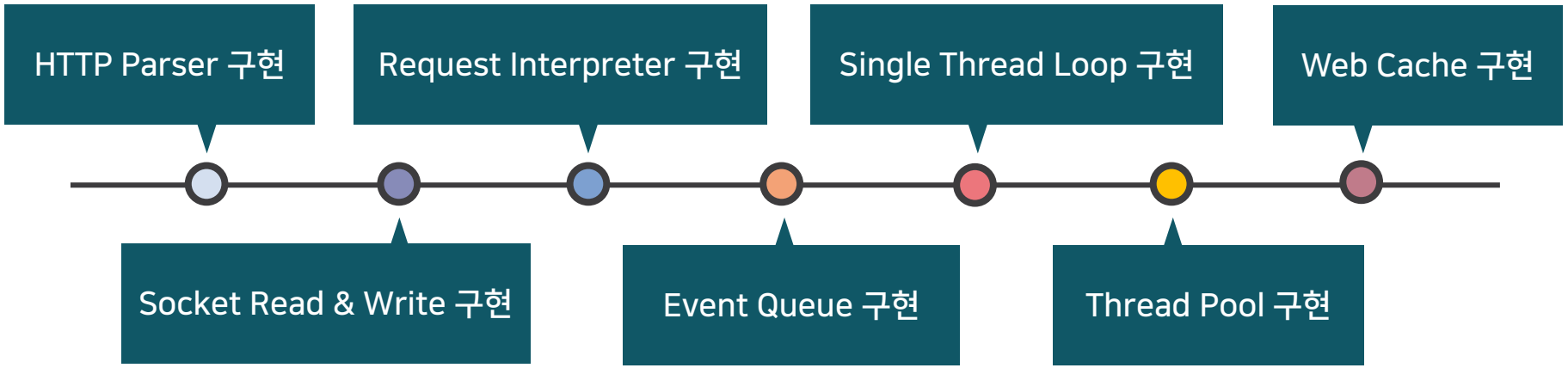
- Event loop는 loop를 돌 때마다 event queue에 쌓인 일을 하나씩 처리
- Blocking job은 worker thread에게 맡기고 곧바로 다른 job을 dequeue
- Worker thread가 I/O를 완료하면 다시 event queue에 enqueue



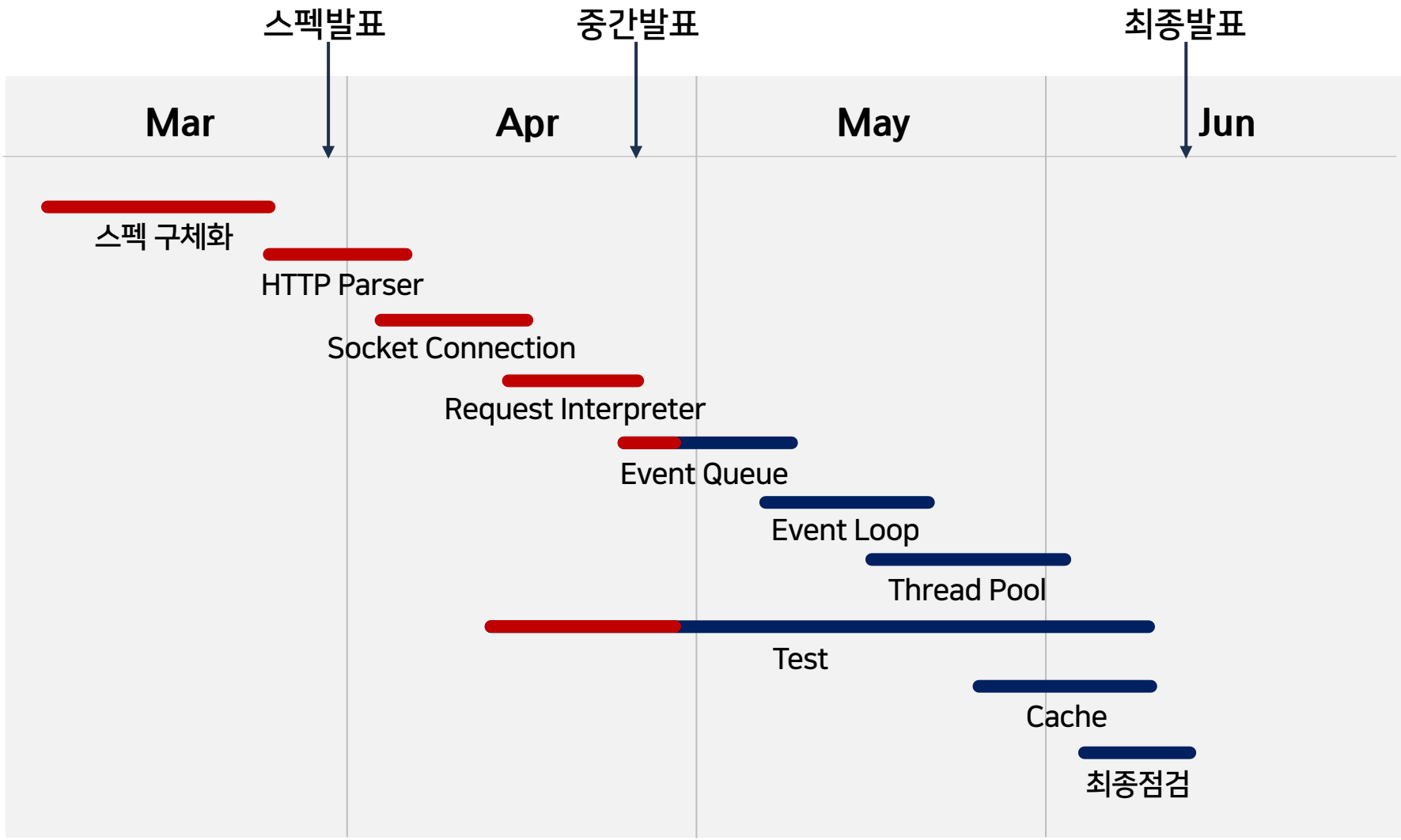
# Architecture(3)



# Implementation Flow



# Schedule

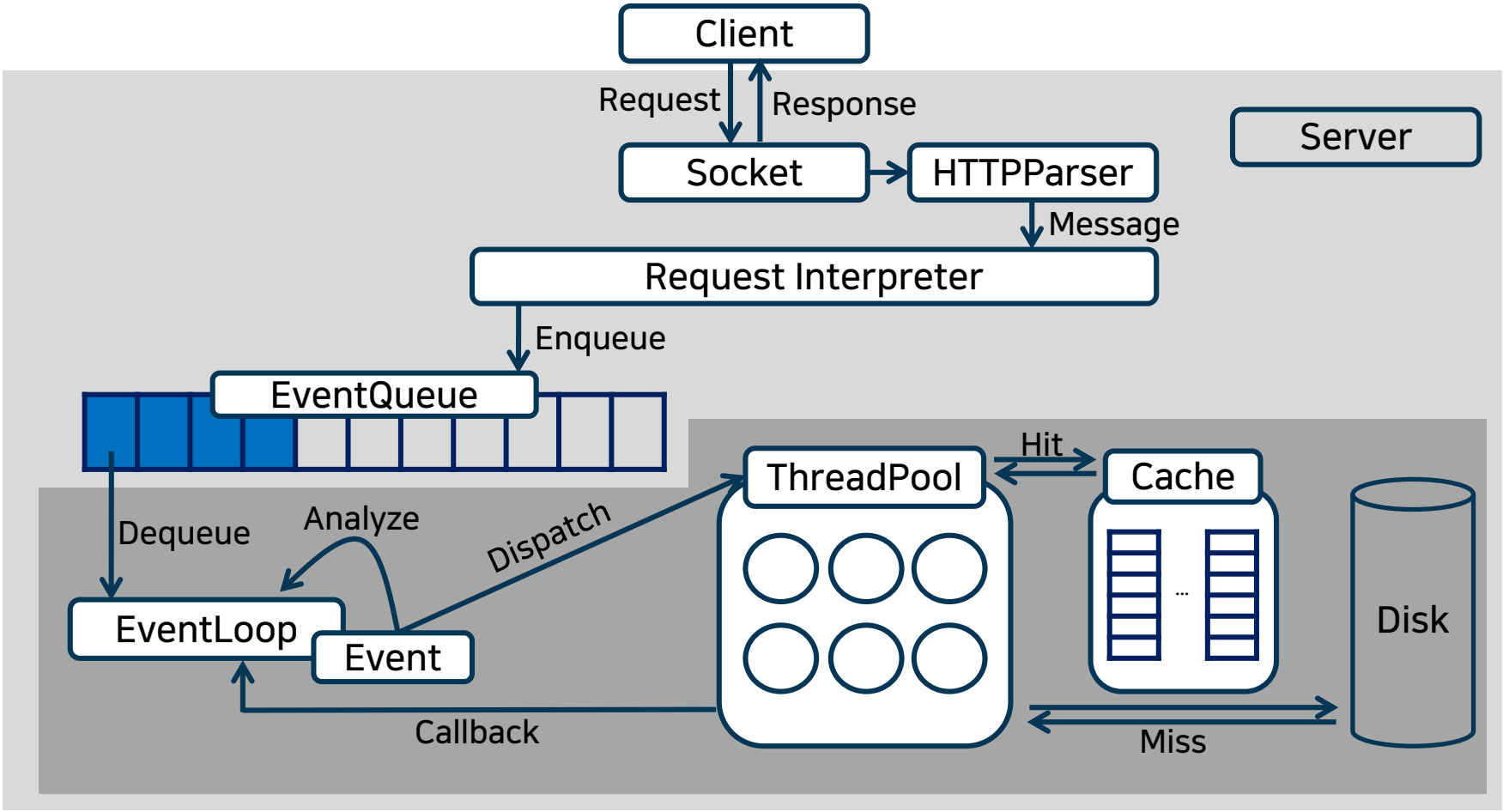

스펙발표
중간발표
최종발표
Mar
Apr
May
Jun
스펙 구체화
HTTP Parser
Socket Connection
Request Interpreter
Event Queue
Event Loop
Thread Pool
Test
Cache
최종점검

# Development Environment

| 운영체제 | 저장소 | 언어 | 테스트 툴 | 오픈소스 |
| --- | --- | --- | --- | --- |
| Windows 10 | GitHub | Java | JMeter | stackoverflow |
| ubuntu | | | Loadrunner Performance Testing tool | koha Open Source Library System |

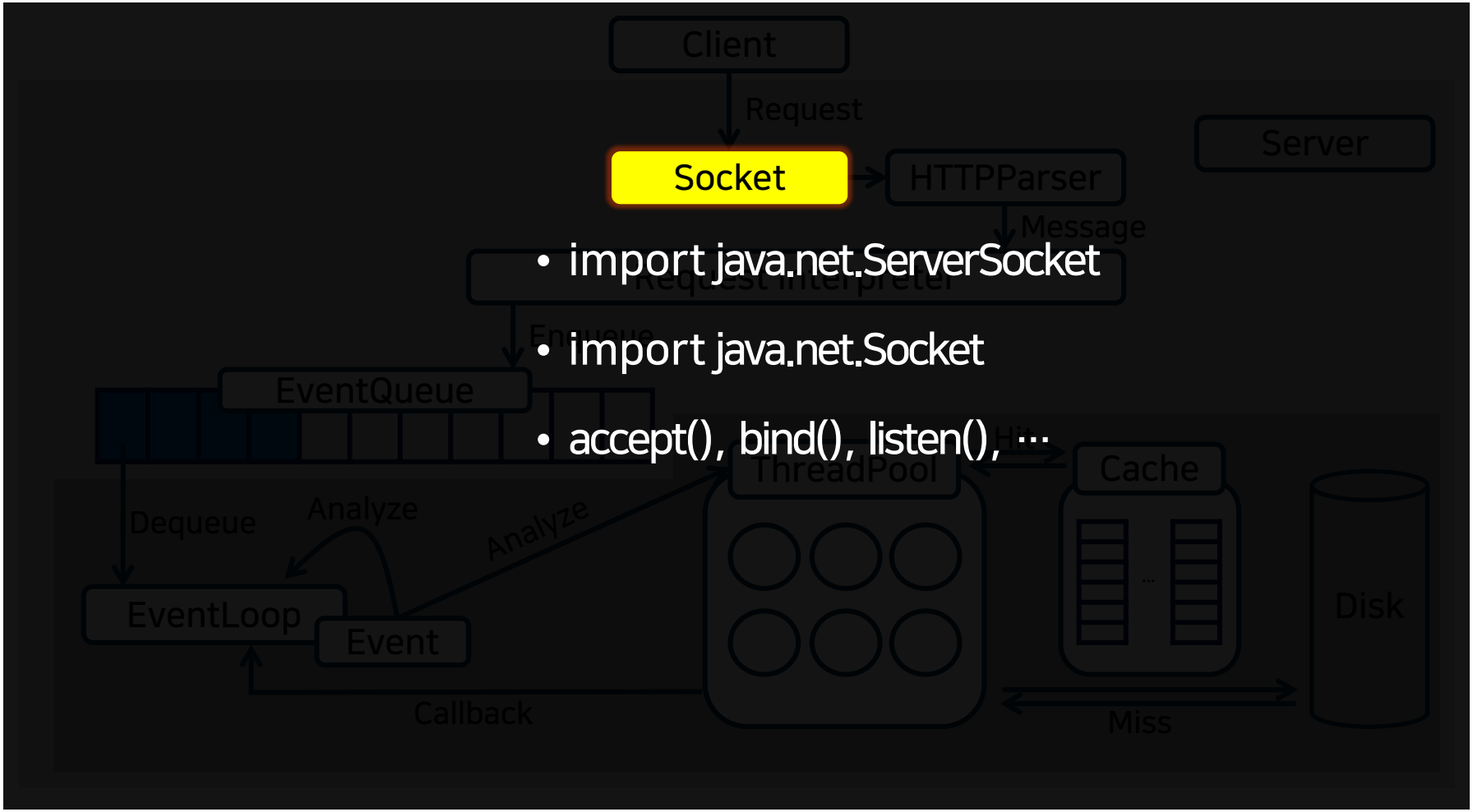
# Current Status(1)

## ✔ Big Picture



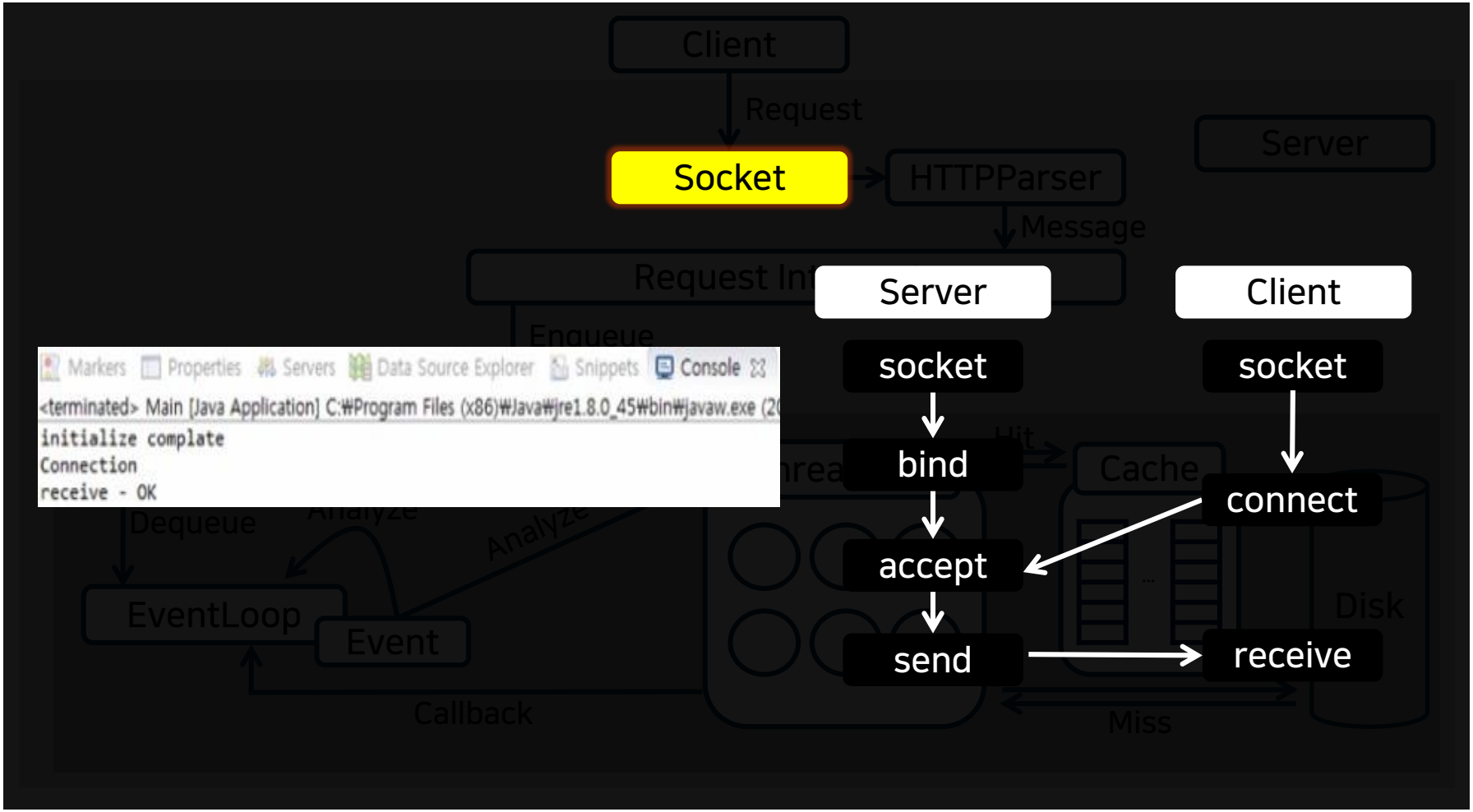
# Current Status(2)

## ✔ Socket Binding



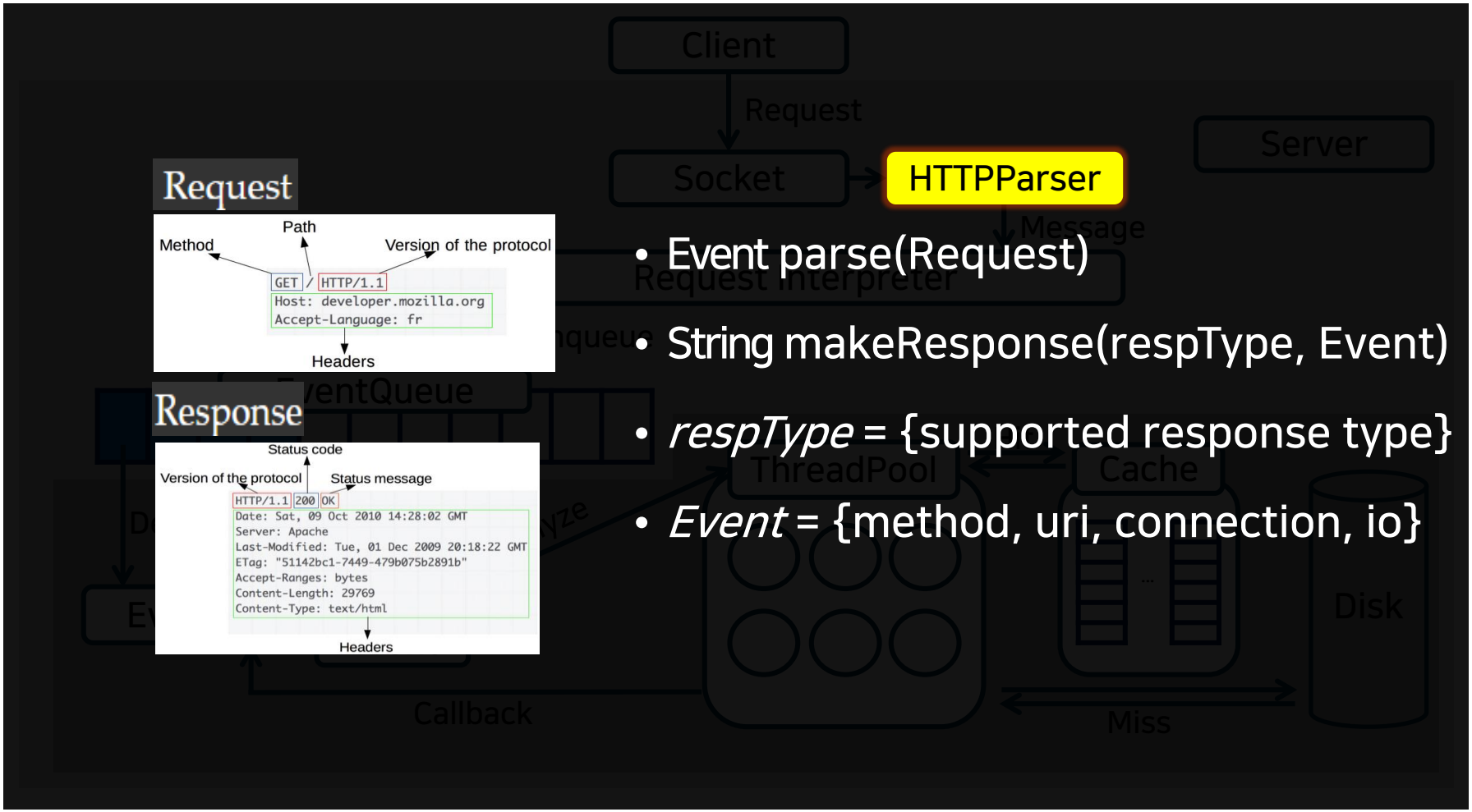
# Current Status(3)

## ✔ Socket Binding



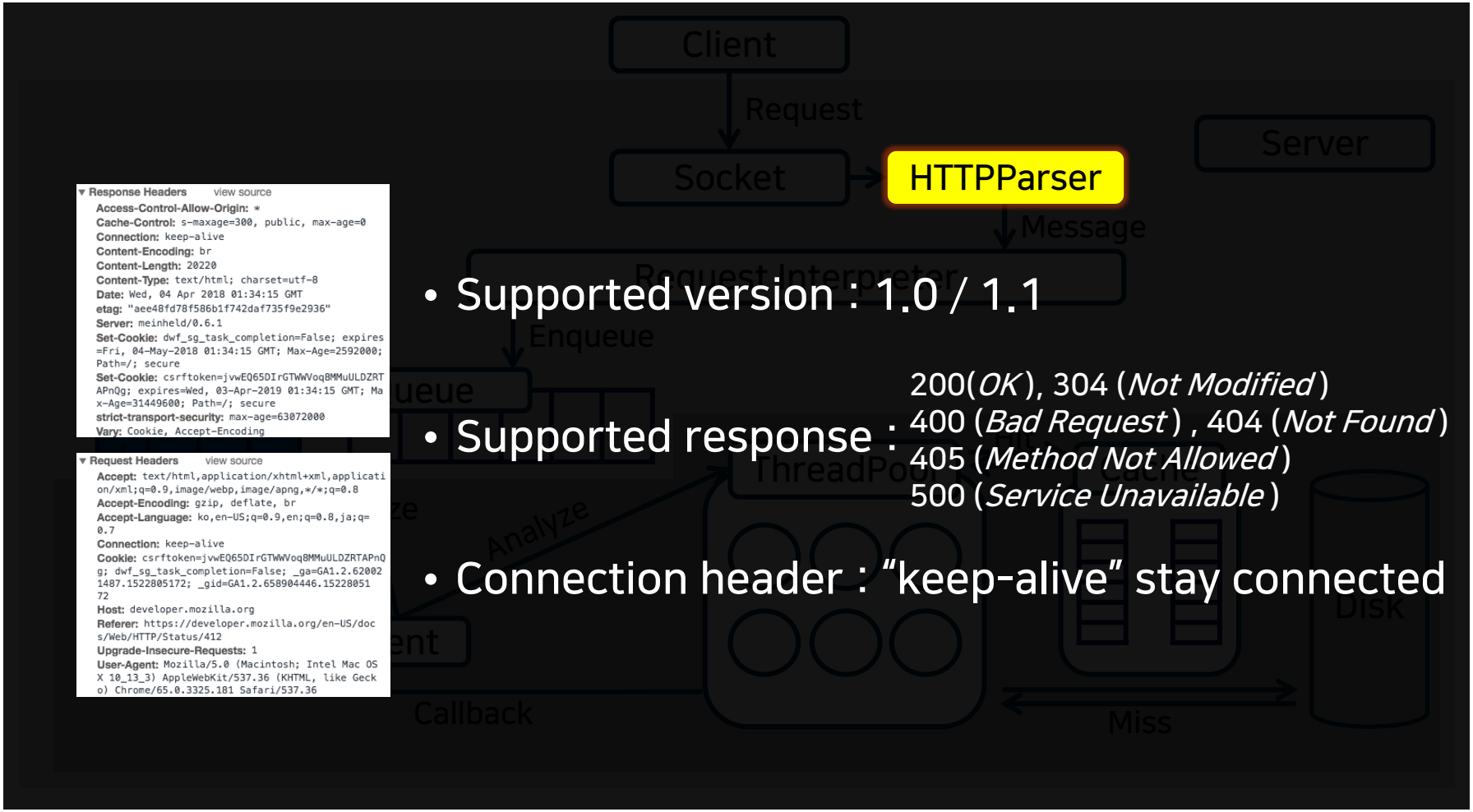
# Current Status(4)

✔ Parser



# Current Status(5)

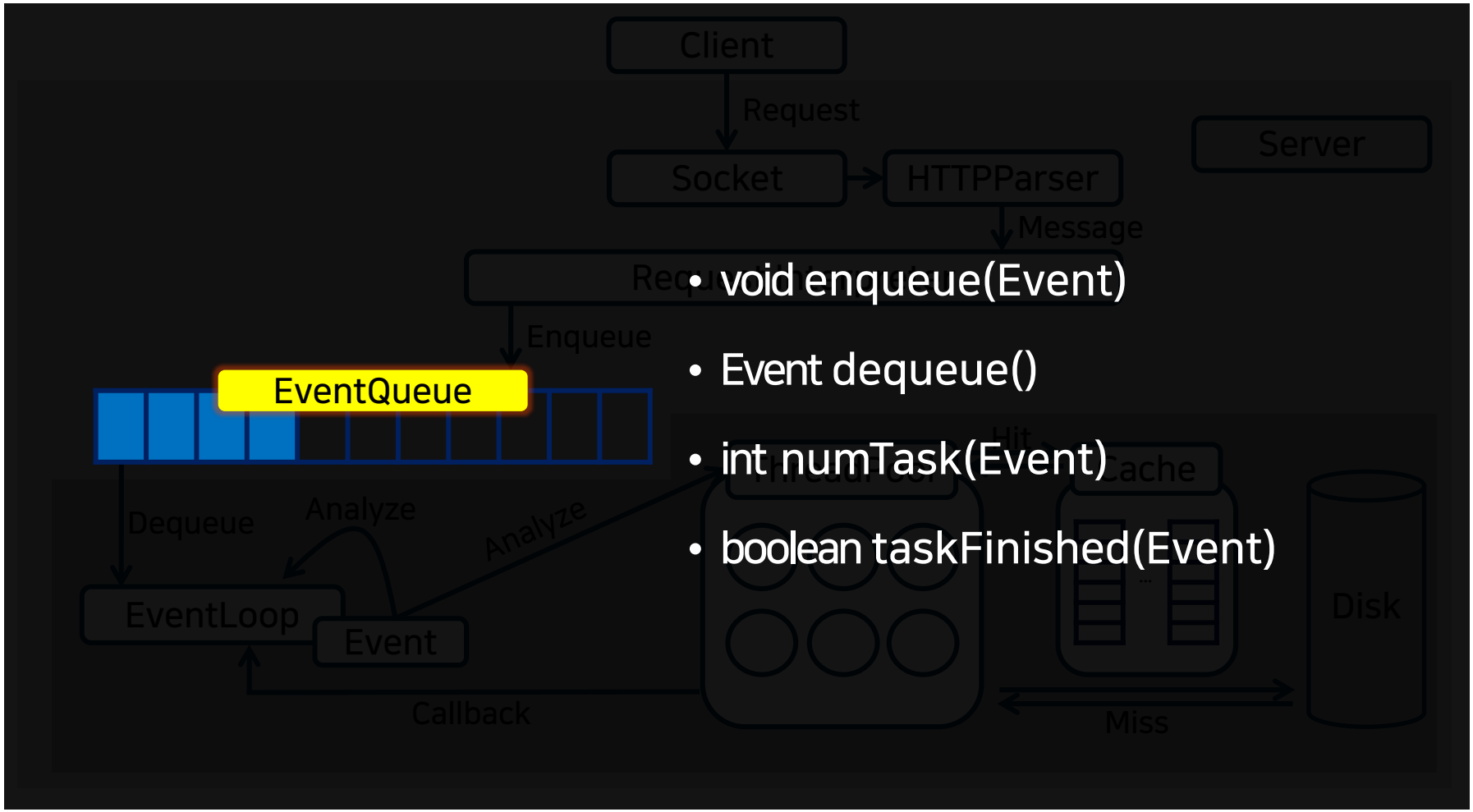
## ✔ Parser




- Supported version : 1.0 / 1.1
- Supported response : 200(*OK*), 304 (*Not Modified*)
  400 (*Bad Request*) , 404 (*Not Found*)
  405 (*Method Not Allowed*)
  500 (*Service Unavailable*)
- Connection header : “keep-alive” stay connected

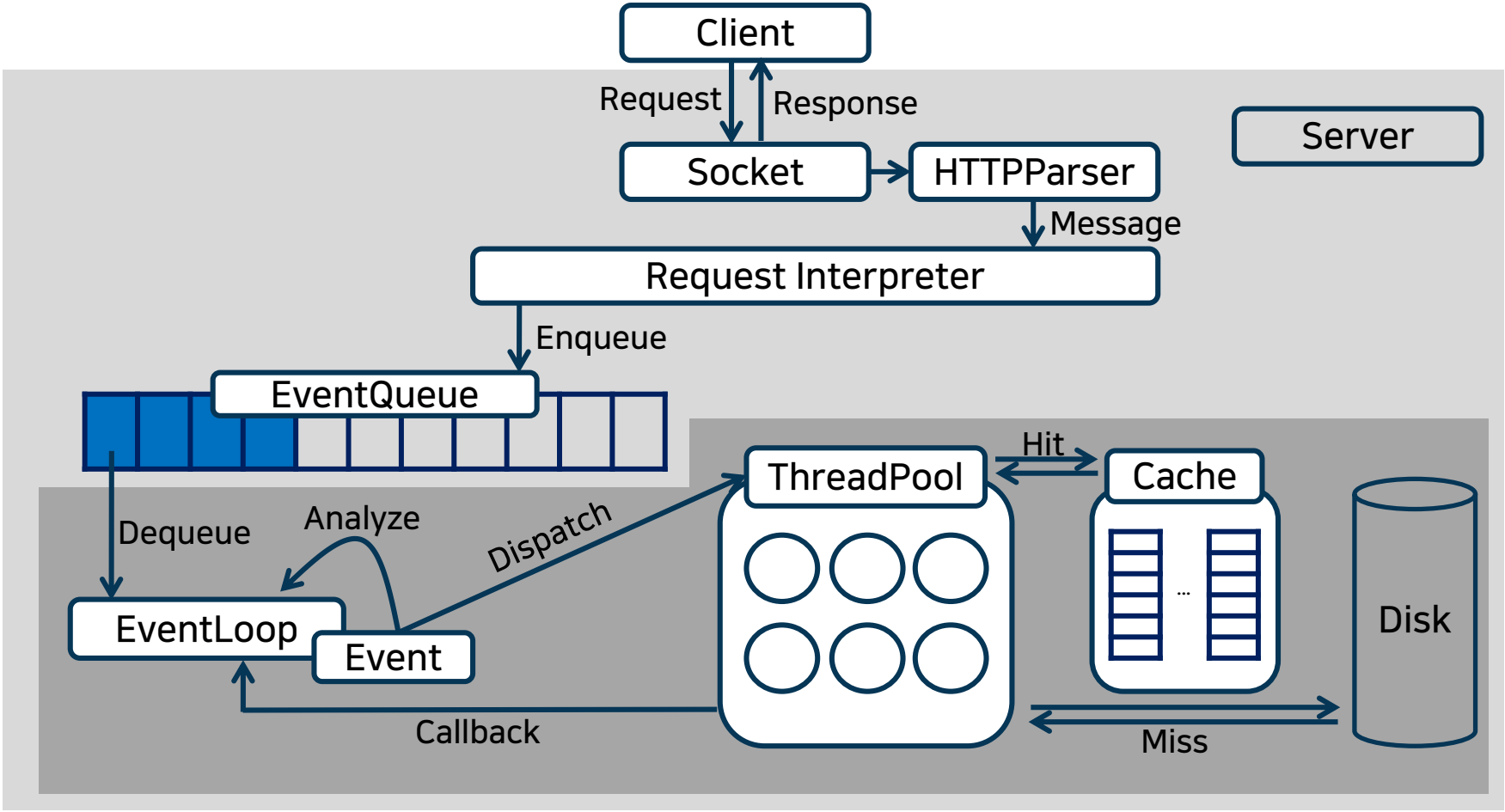
# Current Status(6)

✔ EventQueue



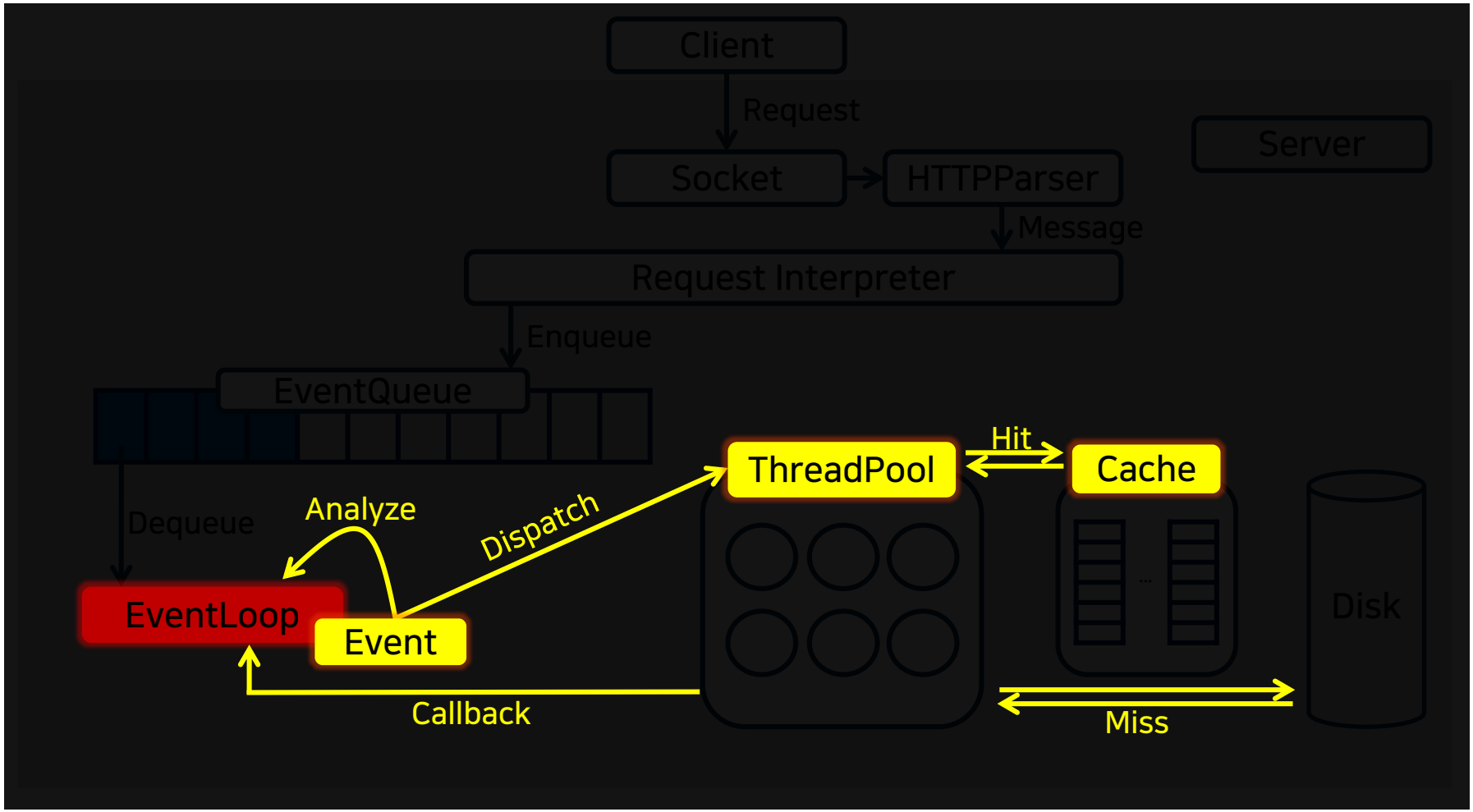
# Further Plan(1)

- ✔ Additional Implementation



# Further Plan(2)

- ✔ Additional Implementation



# Further Plan(3)

## ✔ Demo Plan(1)

- On browser test (Chrome) / using Apache Jmeter
- With famous asynchronous web serser node.js
- Static file system access
- Cache policy : LRU / FIFO

# Further Plan(4)

## ✔ Demo Plan(2)

- Above 1MB file access : using DMA (transfer directly from disk)
- Below 1MB file access : using cache
- Load generation : number of clients
- Client Numbers : 1K, 5K, 10K, 50K, 100K, 500K
- Comparative index : average response time
- Target compare value : at least 80%

# Division of Work

**이승현**

- 전반적인 프로젝트 관리와 일정 조율
- EventQueue 구현
- Connection header 이슈 처리

**박성우**

- 전반적인 프로젝트 관리와 일정 조율
- 소켓 프로그래밍 관련 이슈 담당
- HTTP Parser, Interpreter 구현

**David**

- 테스트 구성 및 오픈소스 코드 스크랩
- 소켓 프로토콜, 초기 개발환경 구축
- Job Classification 및 Type, 코드 리뷰

# 감사합니다