

1. Game Server Architecture Overview & Dev Tools

Great Technology For Great Games



DK Moon

dkmoon@ifunfactory.com



Administrative Stuff

- ✓ 강사: 문대경 (dkmoon@ifunfactory.com)
- ✓ 날짜: 매주 수요일 오후 6:00-8:45pm
- ✓ 강의실: 산학관 421호
- ✓ 강의 웹 사이트: <https://github.com/dkmoon/ajou-spring-2017>
- ✓ Office Hours: 사전 약속이 있는 경우
- ✓ 평가: 절대 평가
출석 20% + 팀 프로젝트 60% + 과제 1회 20%



Syllabus

1st week	3/8	Development tools & Server architecture overview
2nd	3/15	Networking frameworks & message formats
3rd	3/22	Object Relational Mapping (ORM)
4th	3/29	Inter-server RPC & Scaling-out architecture
5th	4/5	Peer-to-Peer
6th	4/12	Game server management
7th	4/19	Project proposal presentation
8th	4/26	Case study #1: iFun Engine
9th	5/3	Game Backend-as-a-Service (GBaaS)
10th	5/10	Case study #2: Game world management
11st	5/17	Client engines: Unreal & Unity3d
12nd	5/24	TCP vs. UDP
13rd	5/31	System design principles
14th	6/7	Project final presentation



Term Project

✓ iFun Engine 을 이용한 네트워크 버전 게임 제작

- Unity/Unreal 게임에 서버를 붙여 게임 서비스 제작해보기
- 기 작업 중인 single play 게임 프로젝트가 있다면 이를 활용 가능
만일 없는 경우, 스켈레톤 클라이언트 제공 가능

✓ 발표

- 프로젝트 제안 발표: 4/19
- 최종 발표: 6/7

On System Design



Loose Definition: System

- ✓ 특정 목적을 위한 **물리적/논리적 컴포넌트**의 집합
- ✓ 핵심 디자인 요소: **Interface** 와 **Architecture**



System Interface

시스템이 **무엇**을 제공하는가?
(외부에서 바라보는 시스템의 모습)





System Architecture

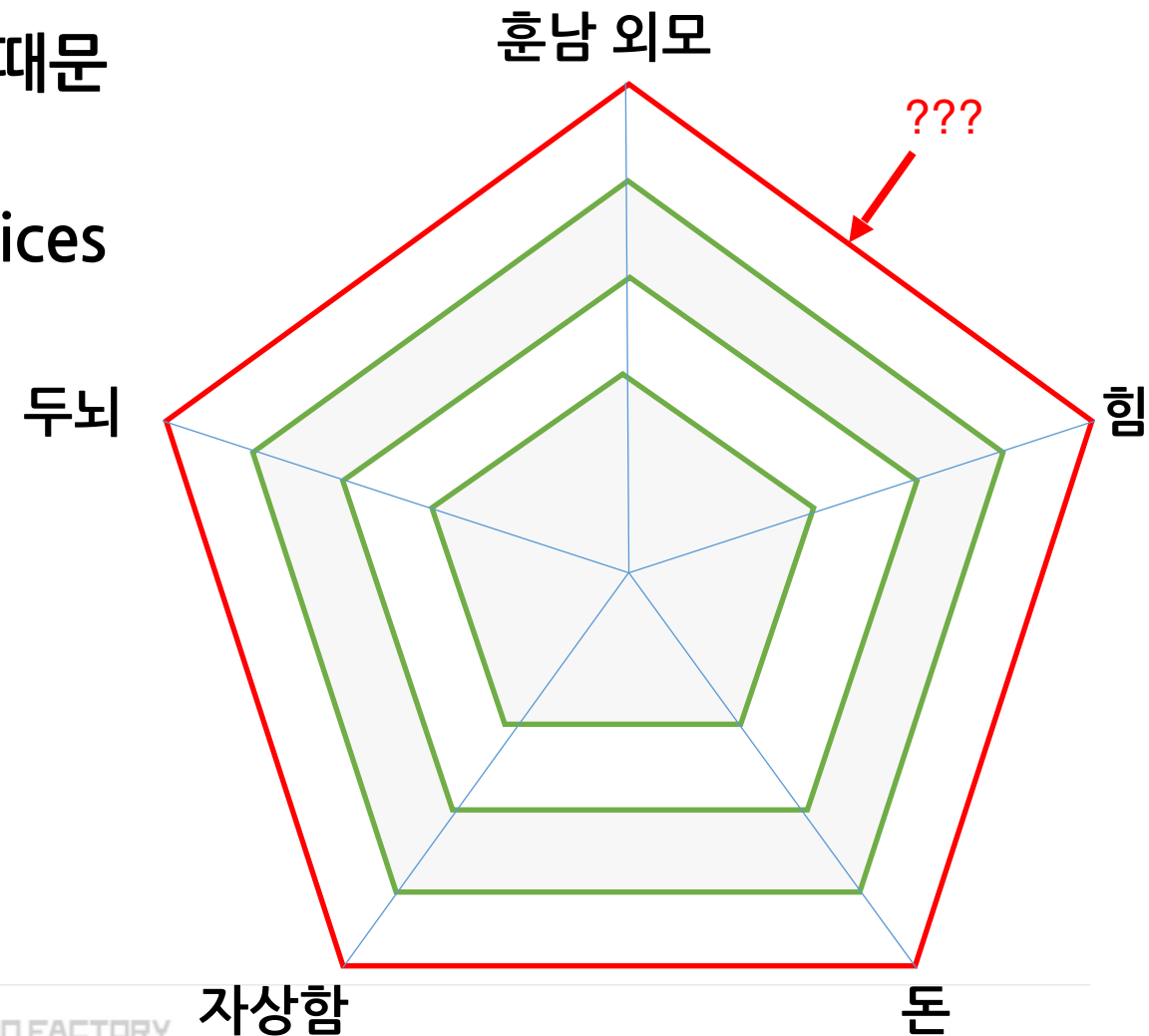
어떤 기능이 어디에 배치되는가?
(내부 구조에 대한 문제)





Trade-offs as System Design Principle

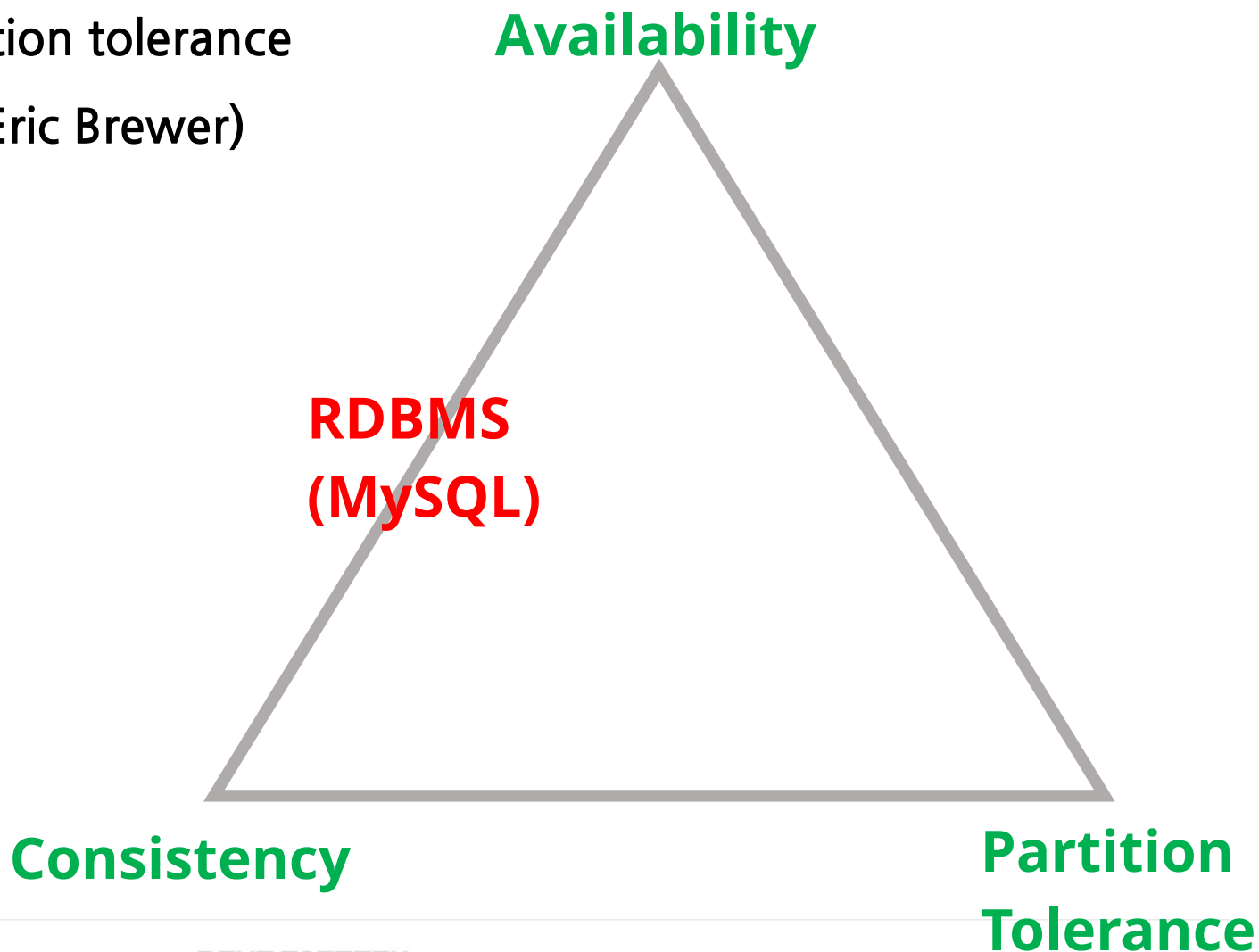
- ✓ 만렙 시스템은 없다
- ✓ 리소스 제약 외에도 선택지 간 충돌 때문
- ✓ Design decisions = Trade-off choices





Trade-off Example: CAP Theorem

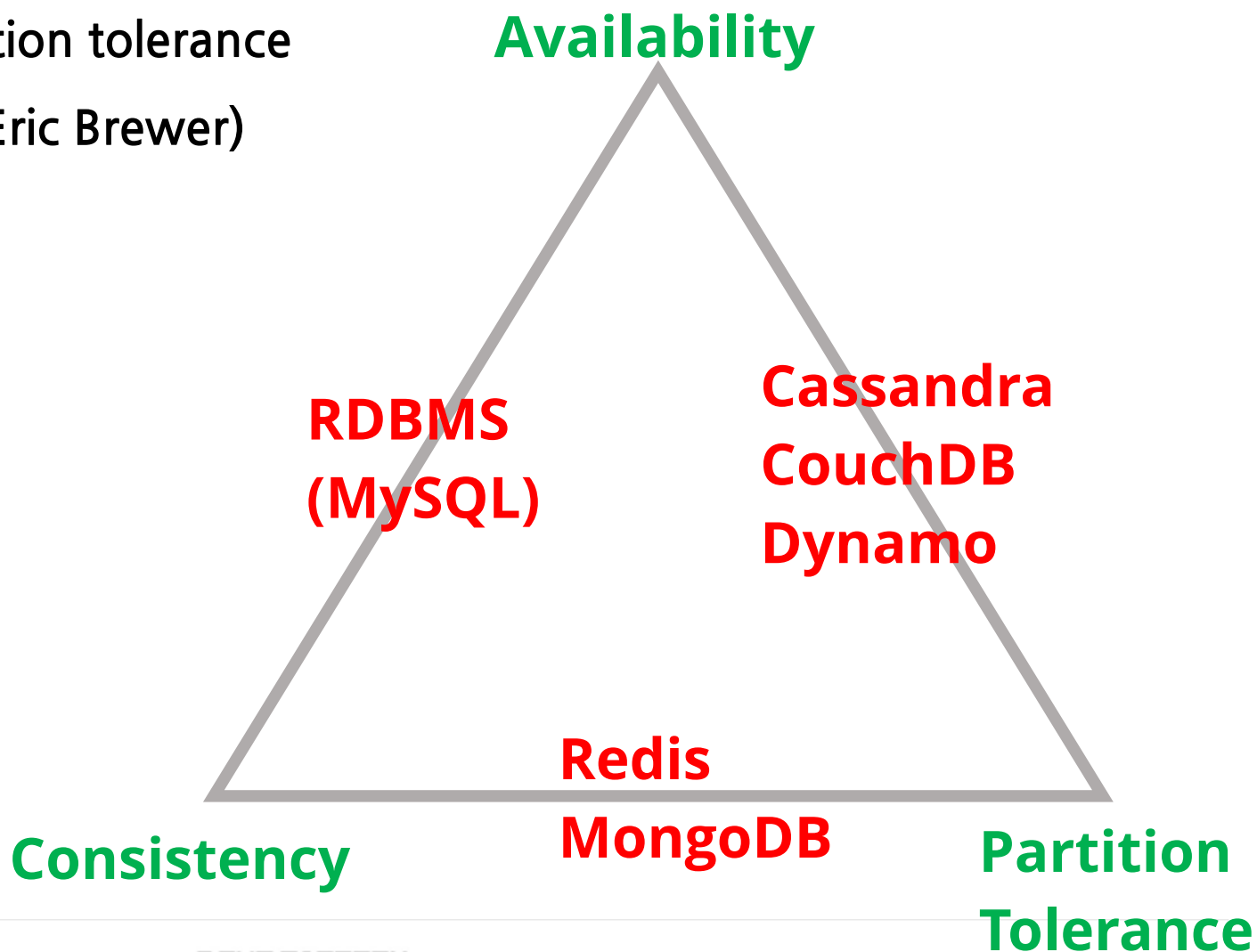
Consistency, Availability, Partition tolerance
이 3개 중 2개만 가능 (by Prof. Eric Brewer)





Trade-off Example: CAP Theorem

Consistency, Availability, Partition tolerance
이 3개 중 2개만 가능 (by Prof. Eric Brewer)



Experienced System Designer

“Know Your System”



(1) 목표의 정의

- 어떤 **가정** 하에 어떤 것들이 **필수**이고, 어떤 것들을 **포기**할 수 있는가?

(2) 목표를 반영하는 인터페이스/아키텍처 디자인

- 인터페이스와 아키텍처에 따라 시스템이 **제공하는 것, 할 수 있는 것, 할 수 없는 것이 결정된다.**

(3) 디자인을 현실화

- 어떤 **Technology**를 쓸 것인가?

System Component Granularity



- ✓ 코드 한 줄
- ✓ 함수
- ✓ 모듈
- ✓ 서비스
- ✓ 사람(?)



시스템 디자이너의 성장 방향

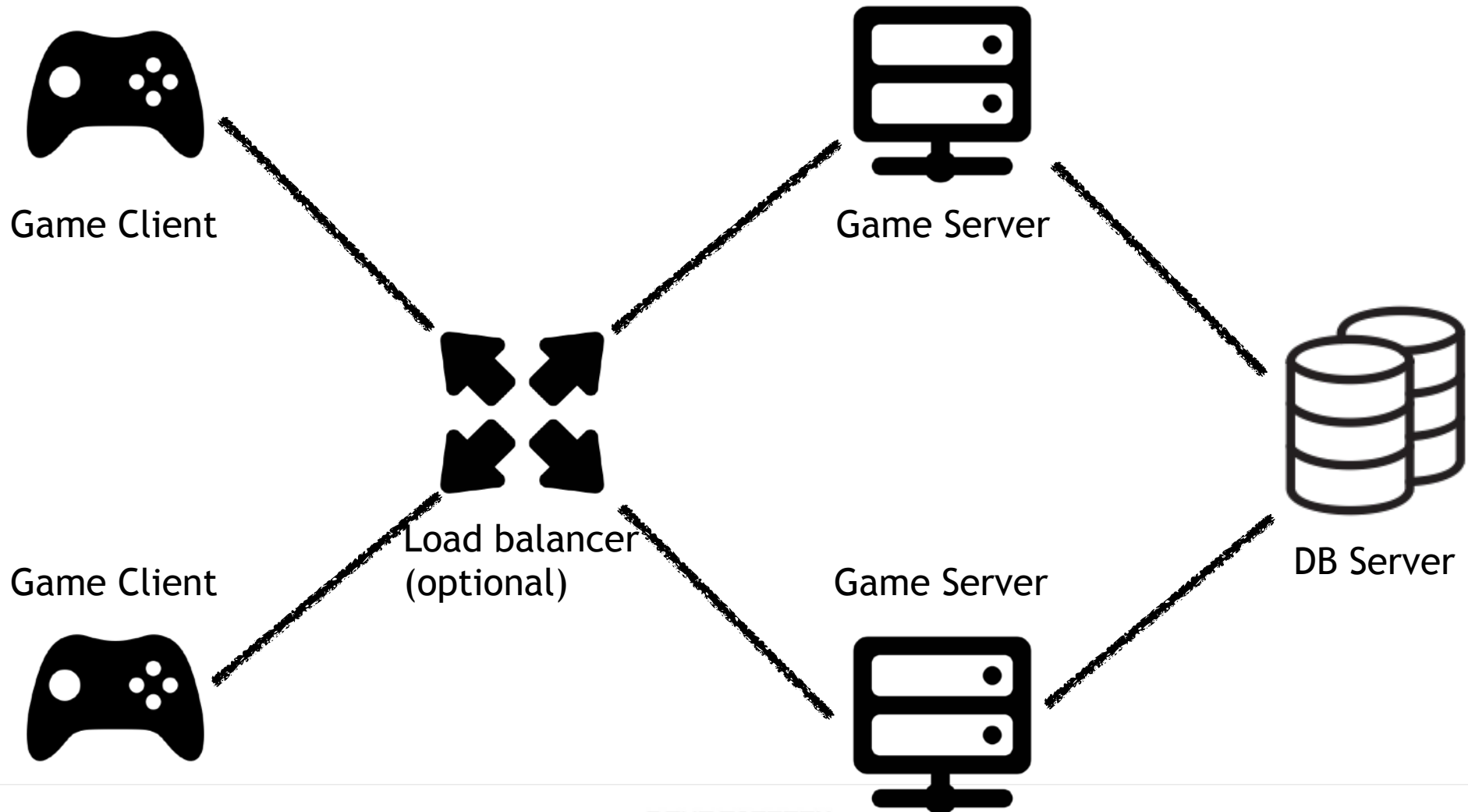
System Component Recap



1. 시스템 디자인에 **절대적으로 우월한 답은 없음**
2. 시스템 디자인의 핵심은 **어떤 가정하에서 무엇이 가능하고 불가능한지** 정하는 것
3. 이를 위한 트레이드 오프 선택과 각각의 의미를 파악
4. 디자인을 구현으로 옮기는 것은 또 다른 기술

Game Service Architecture & Game Server Roles

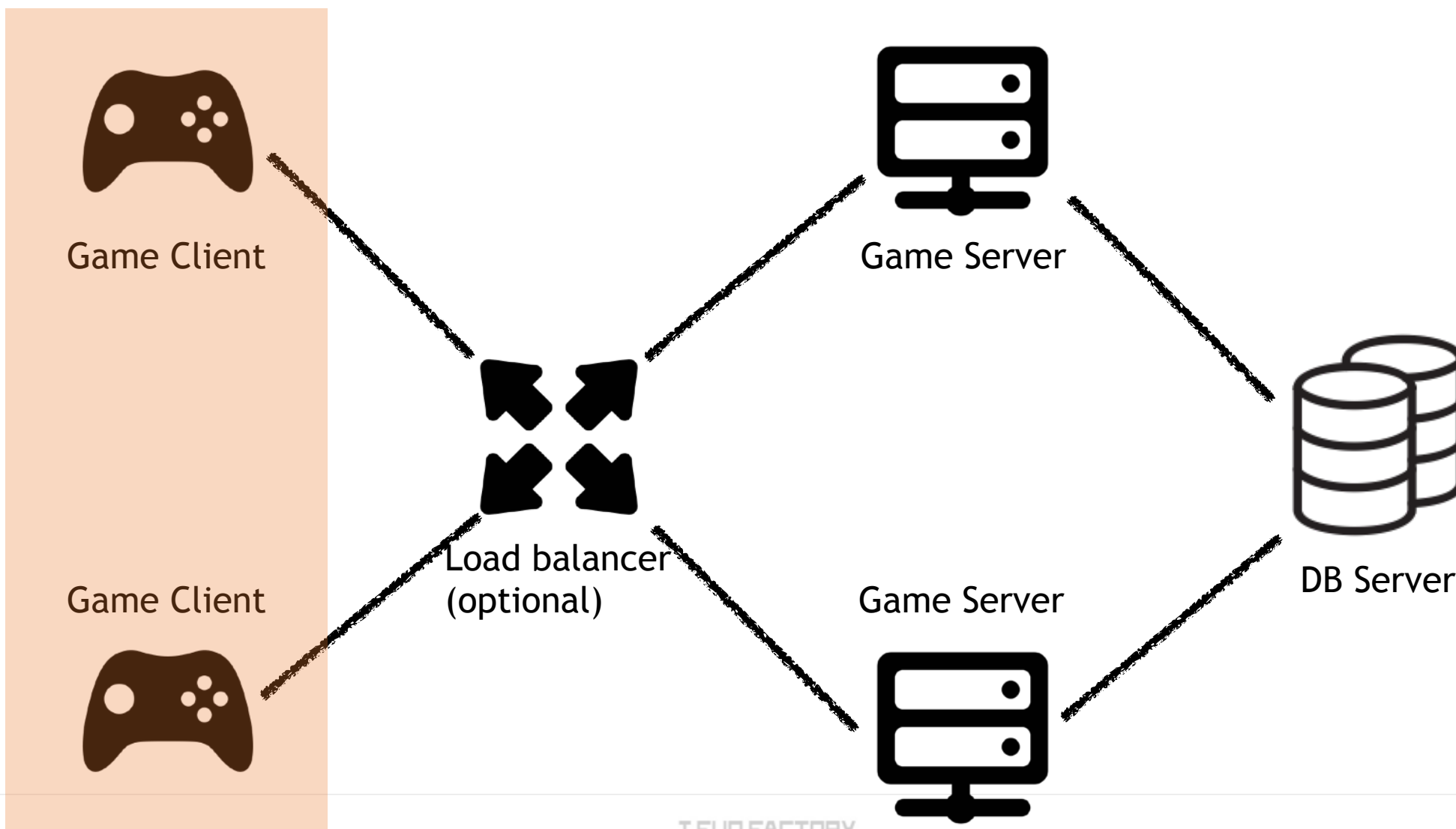
Game Service Architecture: Data Plane



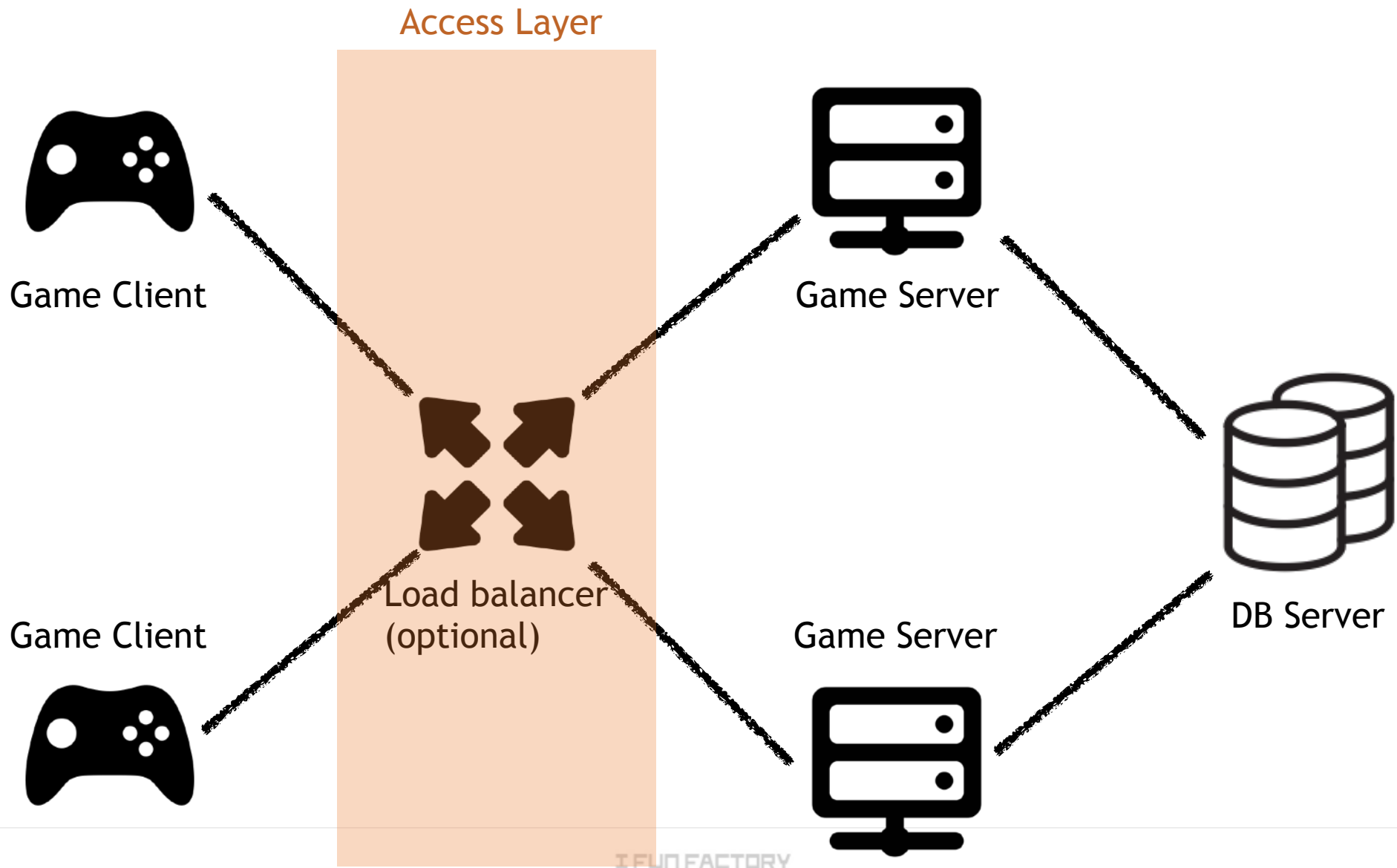
Game Service Architecture: Data Plane



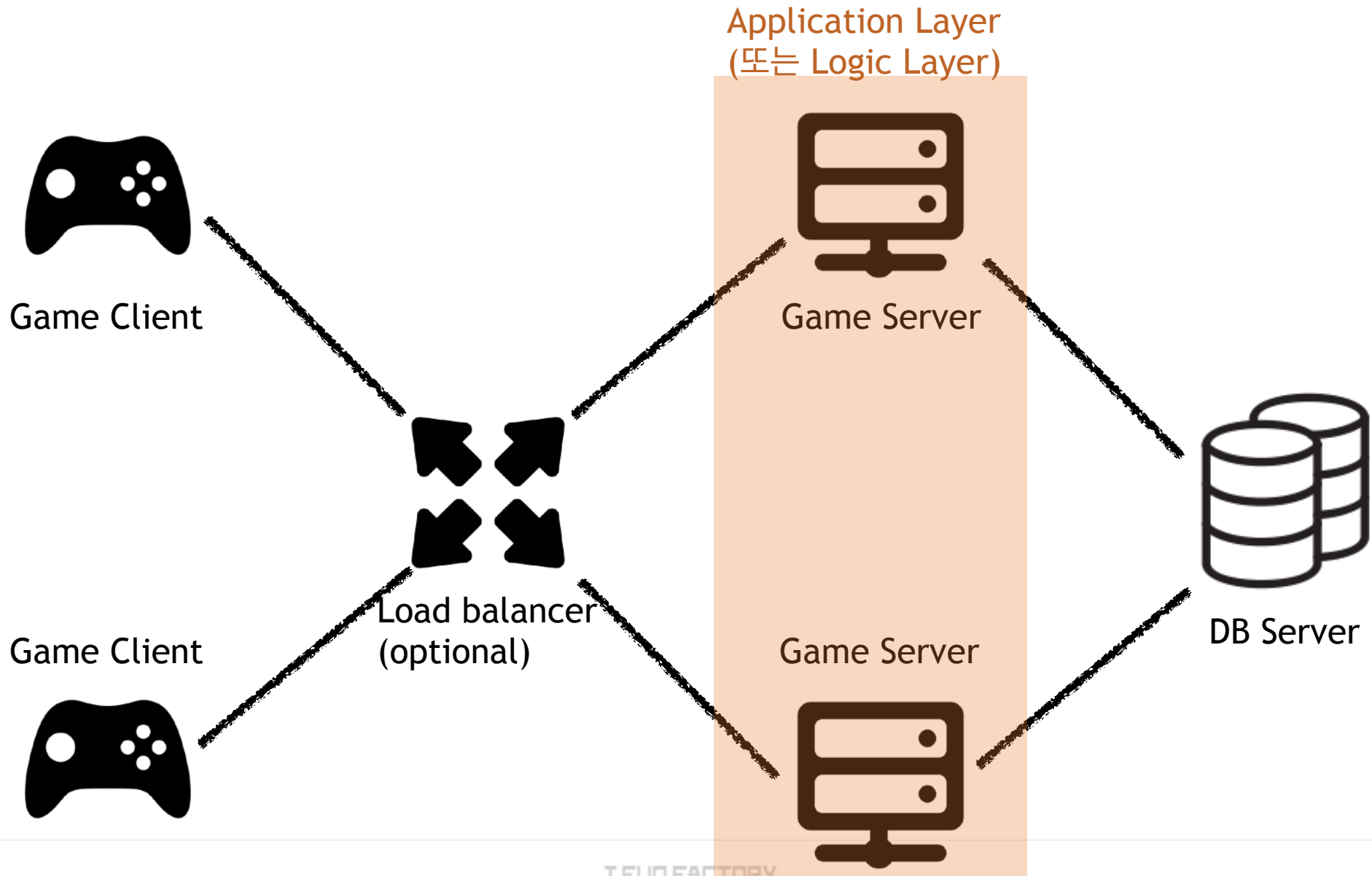
Presentation Layer



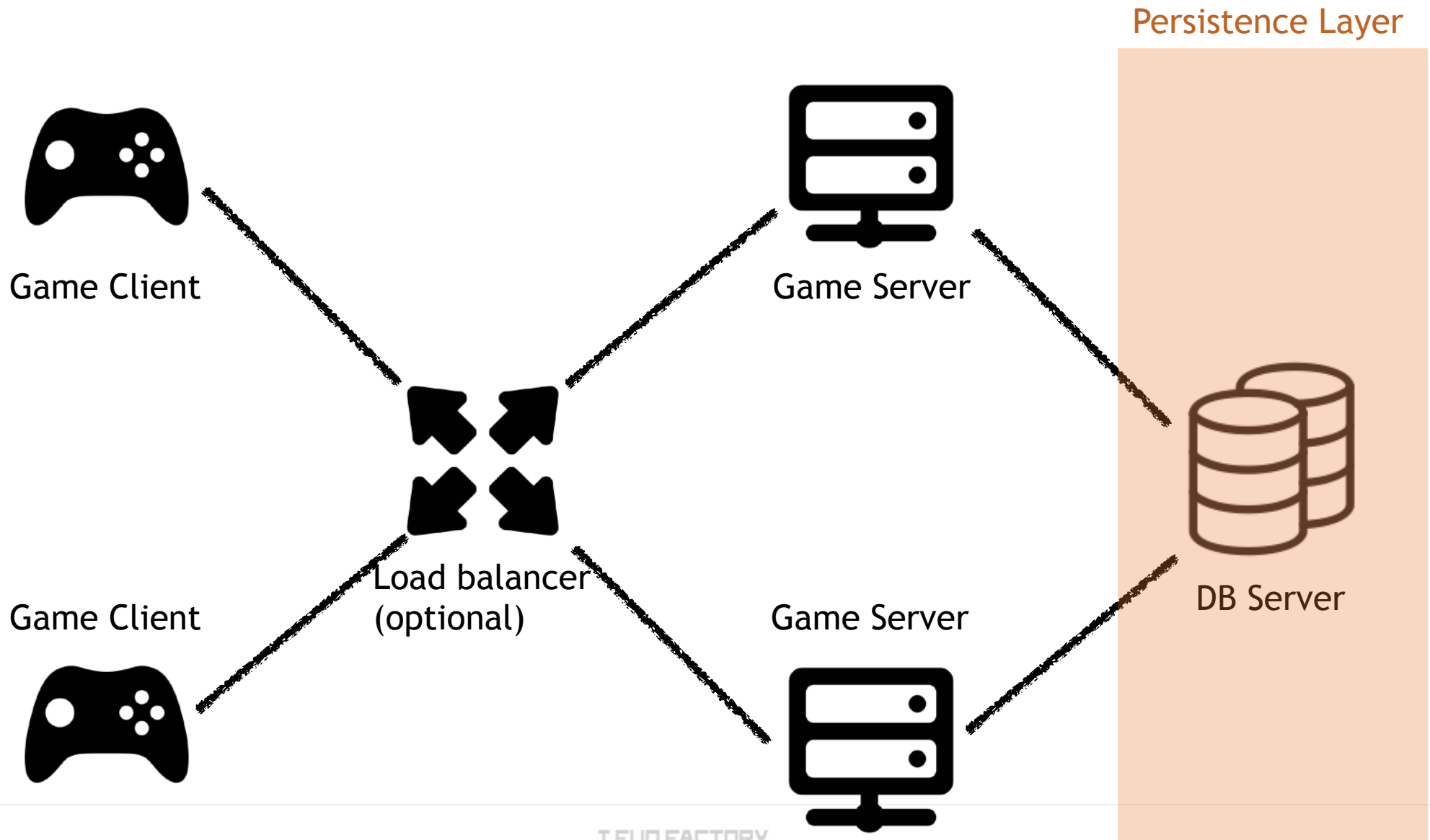
Game Service Architecture: Data Plane



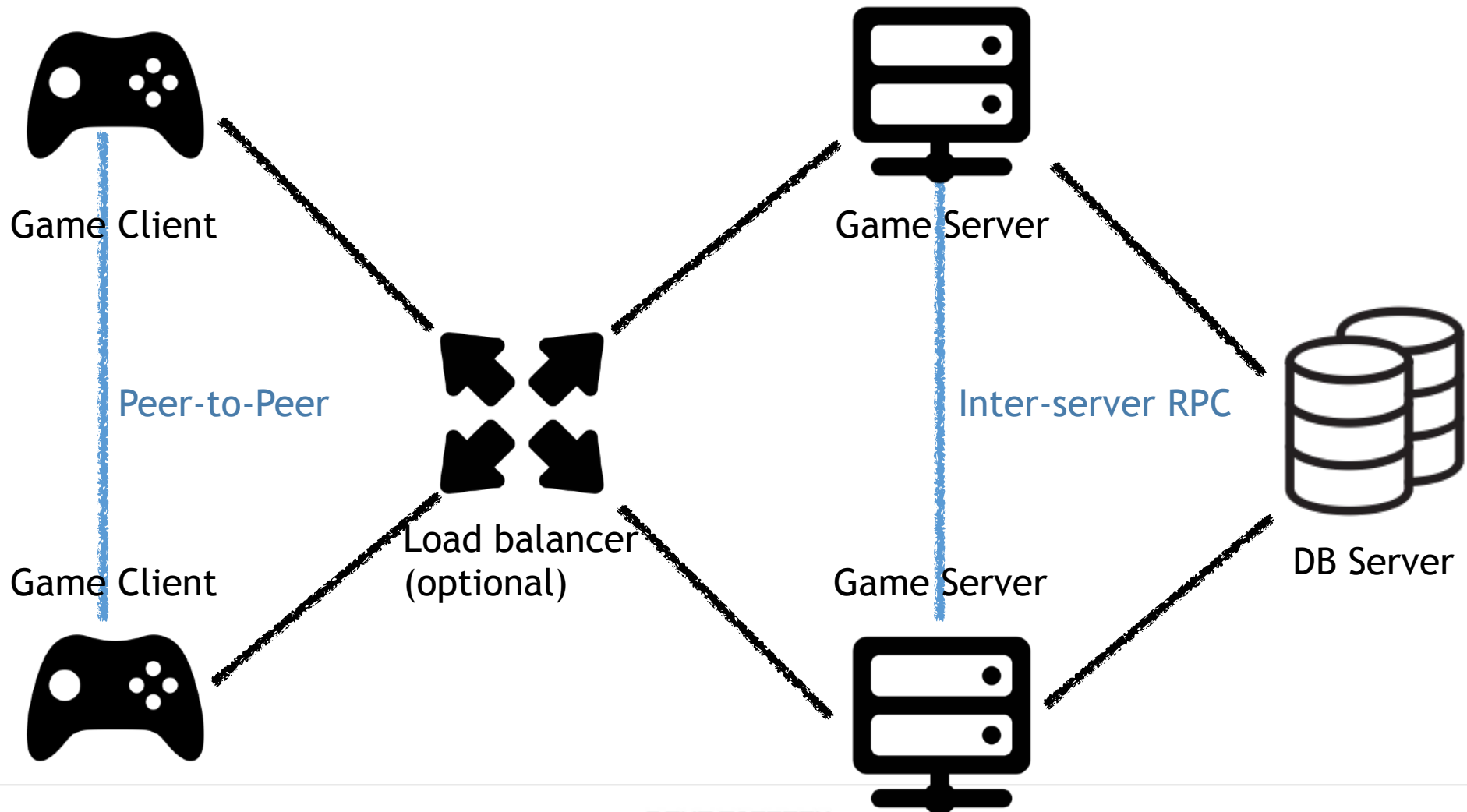
Game Service Architecture: Data Plane



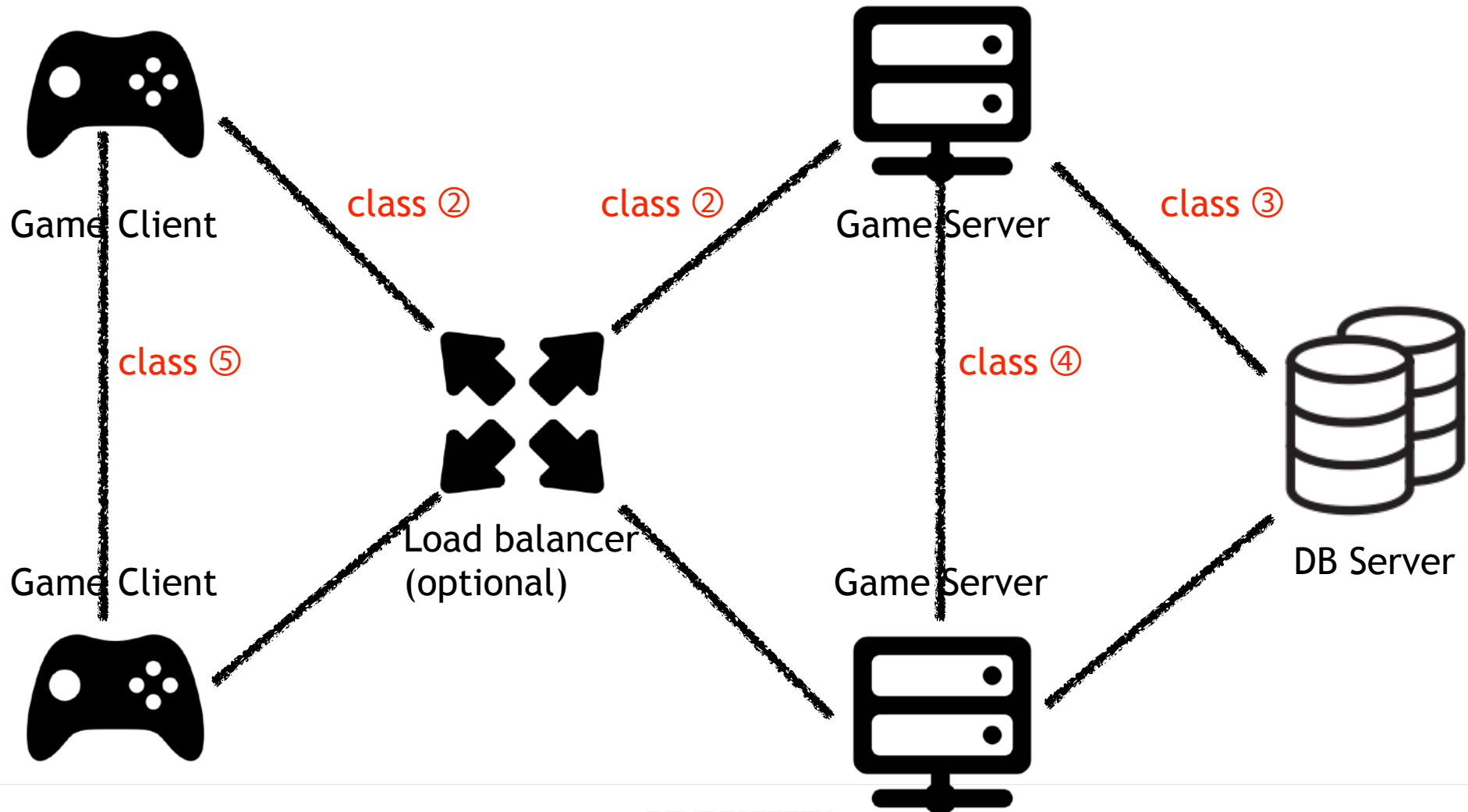
Game Service Architecture: Data Plane



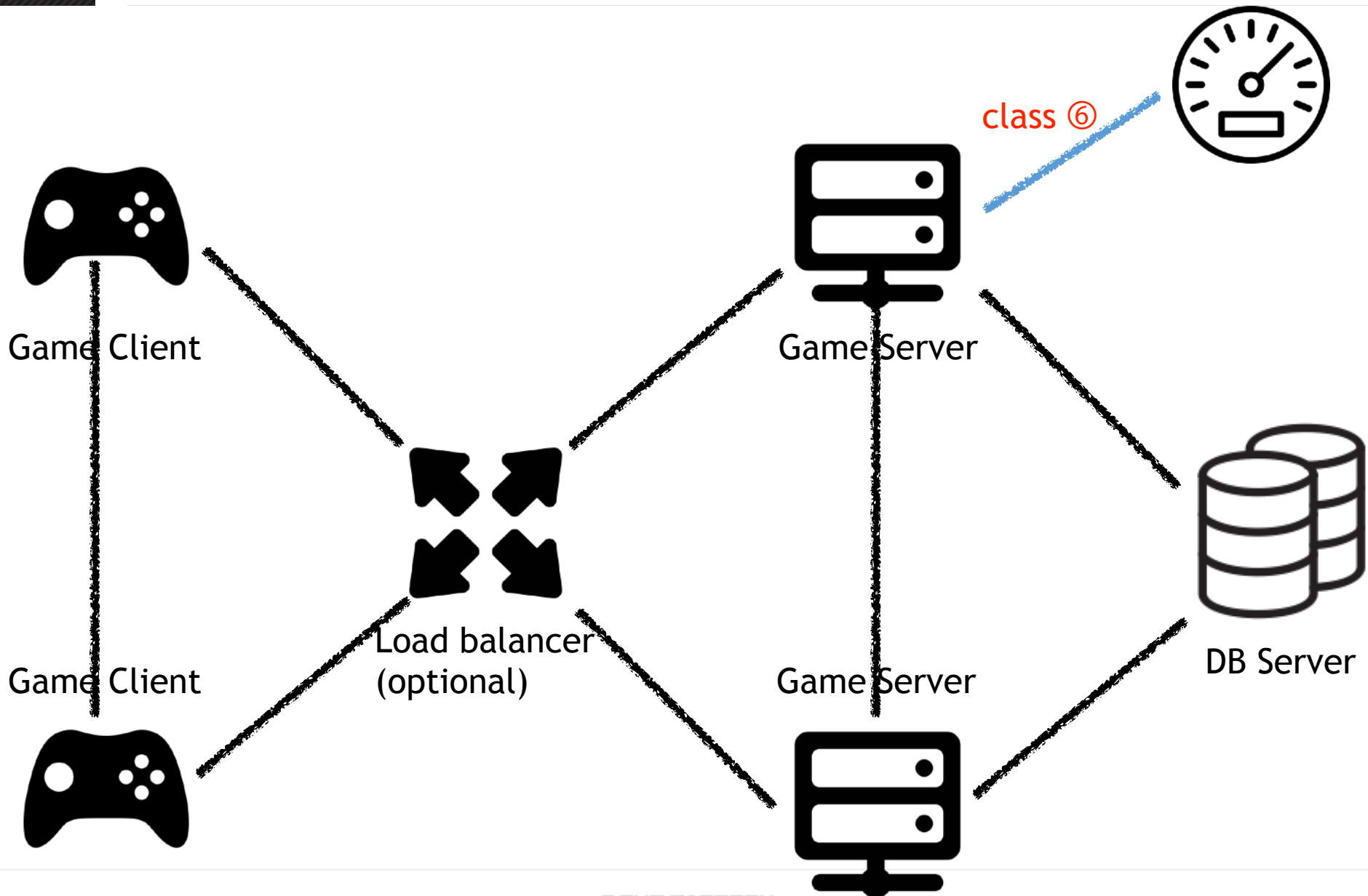
Game Service Architecture: Data Plane



Game Service Architecture: Data Plane

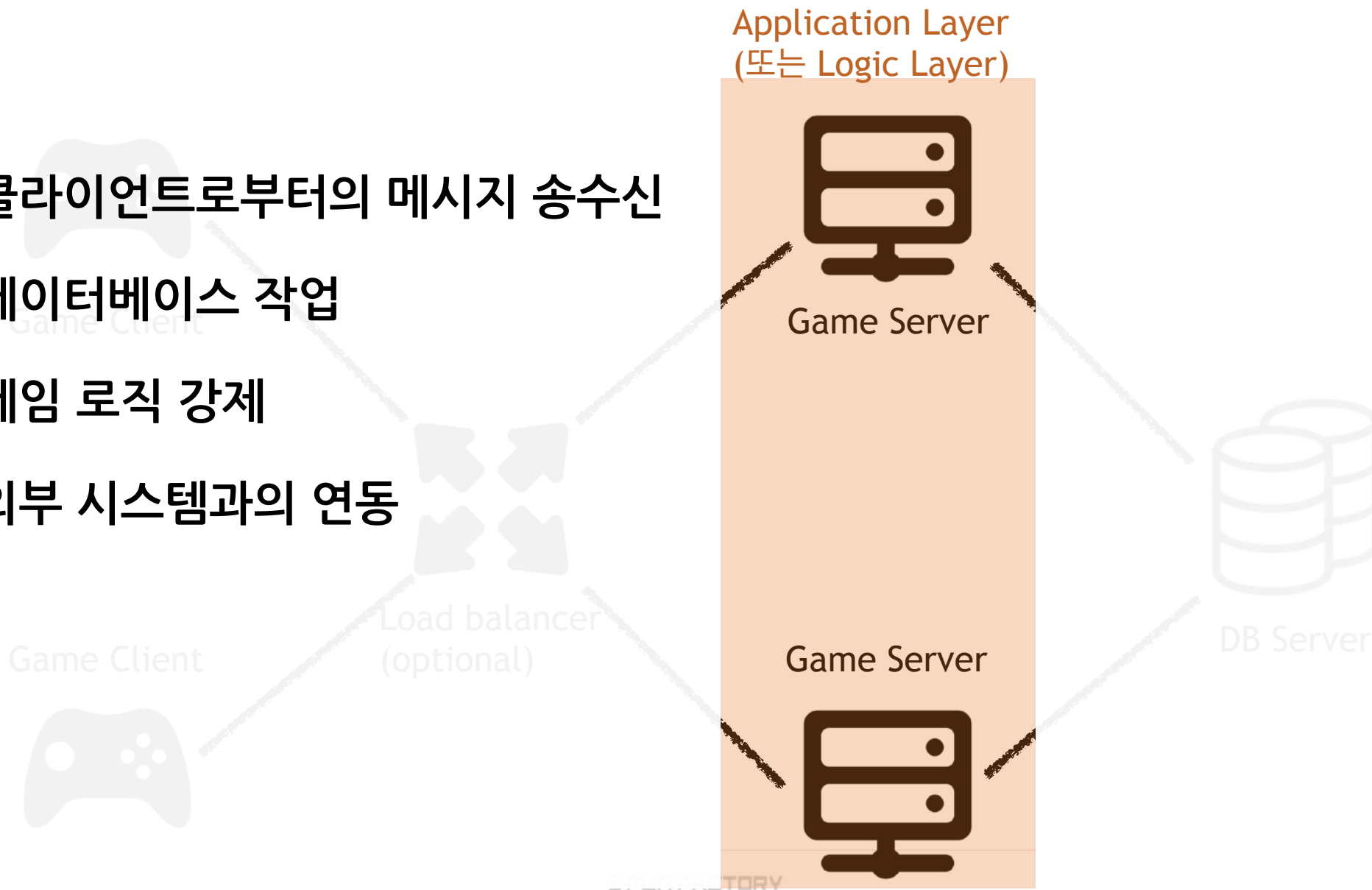


Game Service Architecture: Data Plane



Game Server Roles

- ✓ 클라이언트로부터의 메시지 송수신
- ✓ 데이터베이스 작업
- ✓ 게임 로직 강제
- ✓ 외부 시스템과의 연동



Game Development Tools



Dev Machine

✓ Local 가상화 솔루션들: Xen, VMware, Hyper-V

- Hyper-V 의 경우 Windows 8.1 이후 부터는 기능 활성화만으로 사용 가능



Dev Machine

✓ Local 가상화 솔루션들: Xen, VMware, Hyper-V

- Hyper-V 의 경우 Windows 8.1 이후 부터는 기능 활성화만으로 사용 가능

✓ IaaS 클라우드 서비스들: AWS, KT ucloud, RackSpace, GMO, ...

- IaaS 서비스들은 Local 가상화 솔루션이 깔린 서버들을 자동으로 관리해주는 서비스로 이해할 수 있음. 예: AWS는 Xen 을 활용함



Dev Machine

✓ Local 가상화 솔루션들: Xen, VMware, Hyper-V

- Hyper-V 의 경우 Windows 8.1 이후 부터는 기능 활성화만으로 사용 가능

✓ IaaS 클라우드 서비스들: AWS, KT ucloud, RackSpace, GMO, ...

- IaaS 서비스들은 Local 가상화 솔루션이 깔린 서버들을 자동으로 관리해주는 서비스로 이해할 수 있음. 예: AWS는 Xen 을 활용함

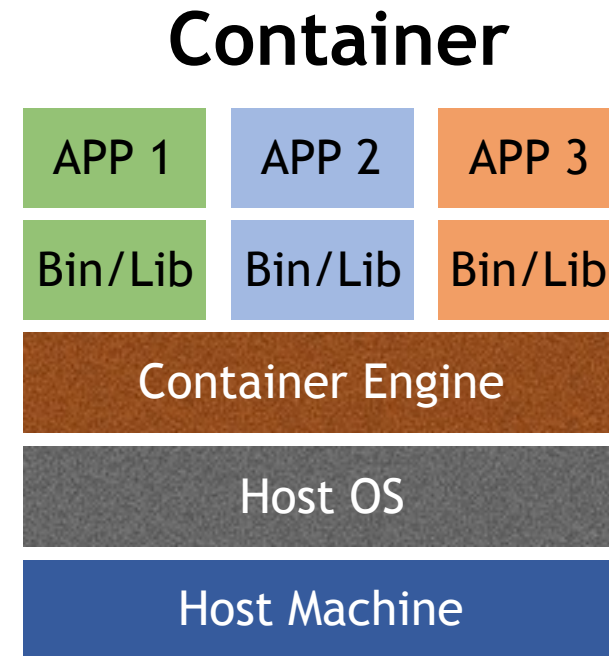
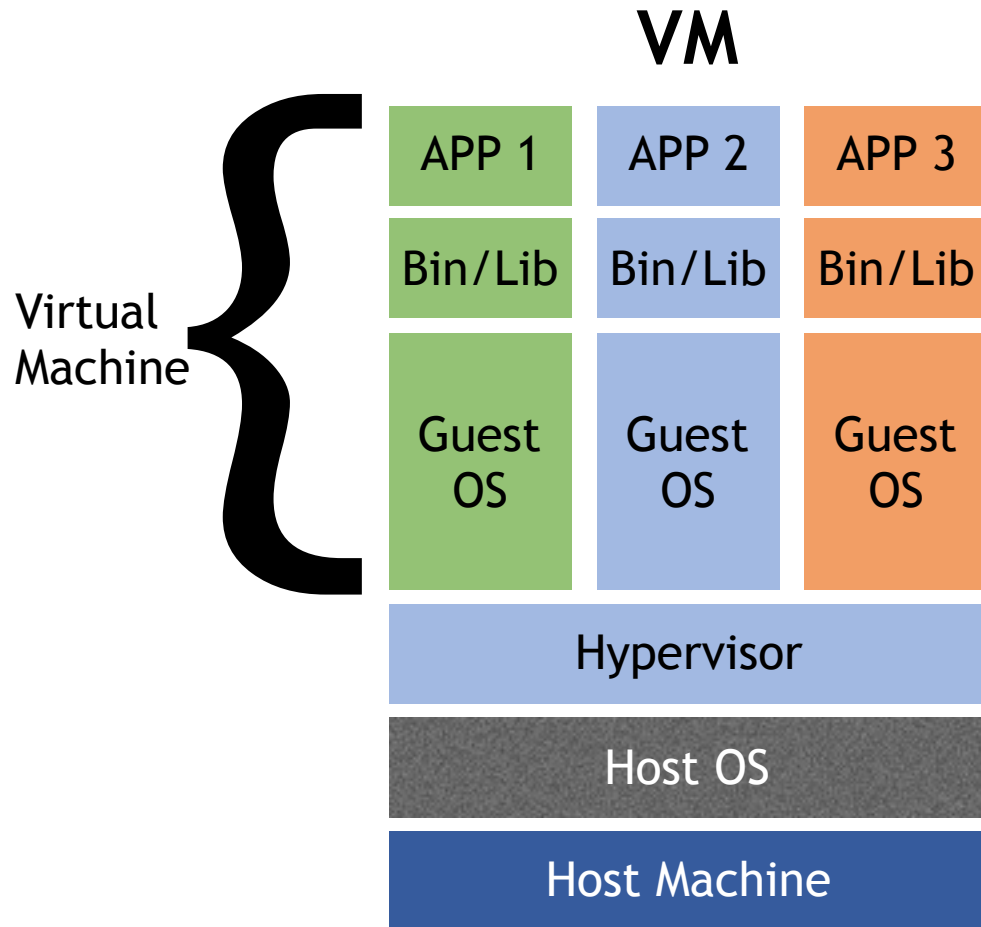
✓ Container 솔루션: Docker, Linux Container

- ✓ 가상화 솔루션 (VM) 은 OS 입장에서 “기계”로 인식될 수 있도록 가상 디바이스를 제공하고, OS 가 이 디바이스를 쓰기 위한 드라이버를 설치하는 등 전체 OS 를 모두 설치하므로 무거움

- ✓ Container 방식은 가상 디바이스 제공 대신, 원래 디바이스를 “공유” 하는 형태로 활용. 대신 같은 종류의 OS 만을 설치할 수 있음.



VM vs. Container





OS Installation: PXE

- ✓ PXE (Preboot Execution Environment)
- ✓ 원격으로 부팅 이미지 제어에 사용
- ✓ 부팅 이미지로 installation 이미지를 활용할 경우 원격 설치가 가능



OS Installation: PXE

- ✓ PXE (Preboot Execution Environment)
- ✓ 원격으로 부팅 이미지 제어에 사용
- ✓ 부팅 이미지로 installation 이미지를 활용할 경우 원격 설치가 가능
- ✓ STEP 1 (C -> S) DHCP request w/ PXE option
- ✓ STEP 2 (S -> C) DHCP reply w / PXE option
(iNitial Bootstrap Program 정보를 받을 수 있는 TFTP 주소 정보 포함)
- ✓ STEP 3 (C -> S) TFTP 를 통해 NBP 다운로드
(NBP 는 보통 HTTP, iSCSI 등을 지원하는 Kernel image)
- ✓ STEP 4 (C -> S) Kernel 로 부팅 후 OS 설치 script (PRESEED, KICKSTART)
다운로드 받아서 실행

Version Control Systems

SVN vs. GIT



✓ 중앙 집중식 저장소 vs. 분산 저장소

- 중앙 집중식에서는 각 개발자가 받아가는 저장소는 특정 snapshot 임.
- 분산 방식에서는 각 개발자의 저장소가 하나의 complete 한 저장소임.

✓ SVN: 특정 리비전의 소스 트리를 서버로부터 꺼내는 개념

✓ GIT: 전체 히스토리를 다른 repository 에서 clone 하는 개념

- 각 개발자가 complete한 저장소를 운영하므로, 공유 저장소에 publish 되기 전까지 그 안에서 commit 의 순서 및 내용을 자유자재로 변경 가능.
- Binary 파일의 경우 변경전 사본까지 모두 저장하게 됨
- 각 repository 가 개별 repository 이므로 local 에서 branch 작업이 용이

Version Control Systems

SVN vs. GIT



	SVN	GIT
출시	2000	2005
소스 체크 아웃	특정 subtree 를 체크아웃하는 형태	repo 전체를 복제하는 형태
속도 (diff, history, switching)		✓
repo 사이즈		✓
브랜치 핸들링 (비선형 개발)		✓
백업 용이성	✓	
사용 편이성	✓	
오프라인 작업 편이성		✓
명시적 GC 불필요성	✓	

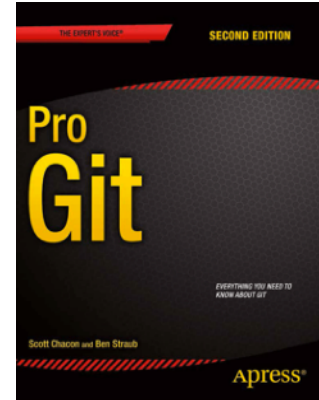
Version Control Systems

Learning/Using GIT



✓ 책: ProGIT (<https://git-scm.com/book/ko/v2>)

- 책을 open source 로 공개함
- 한국어 버전 번역 역시 open source project 임



Version Control Systems

Learning/Using GIT

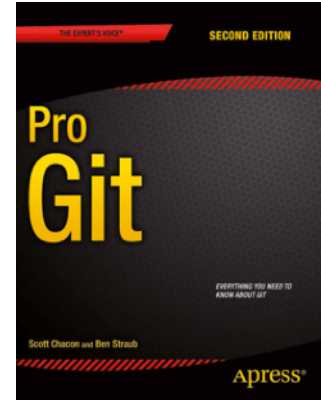


✓ 책: ProGIT (<https://git-scm.com/book/ko/v2>)

- 책을 open source 로 공개함
- 한국어 버전 번역 역시 open source project 임

✓ GitHub (git repository 서비스)

- 많은 open source 프로젝트들이 GitHub 상에 호스팅되고 있음
- 프로젝트 참여 권한 관리, Commit 내용 브라우징 등을 웹 상으로 제공
- 소스 repository 외에 Wiki 등의 부가적인 서비스 제공
- 클래스 홈페이지도 GitHub 에 호스팅되어있음



Version Control Systems

Learning/Using GIT



✓ 책: ProGIT (<https://git-scm.com/book/ko/v2>)

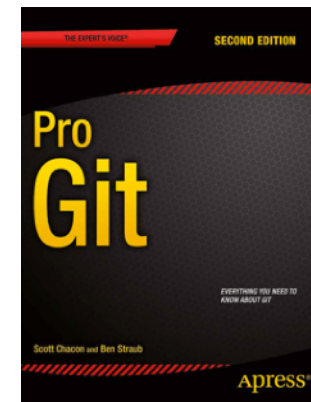
- 책을 open source 로 공개함
- 한국어 버전 번역 역시 open source project 임

✓ GitHub (git repository 서비스)

- 많은 open source 프로젝트들이 GitHub 상에 호스팅되고 있음
- 프로젝트 참여 권한 관리, Commit 내용 브라우징 등을 웹 상으로 제공
- 소스 repository 외에 Wiki 등의 부가적인 서비스 제공
- 클래스 홈페이지도 GitHub 에 호스팅되어있음

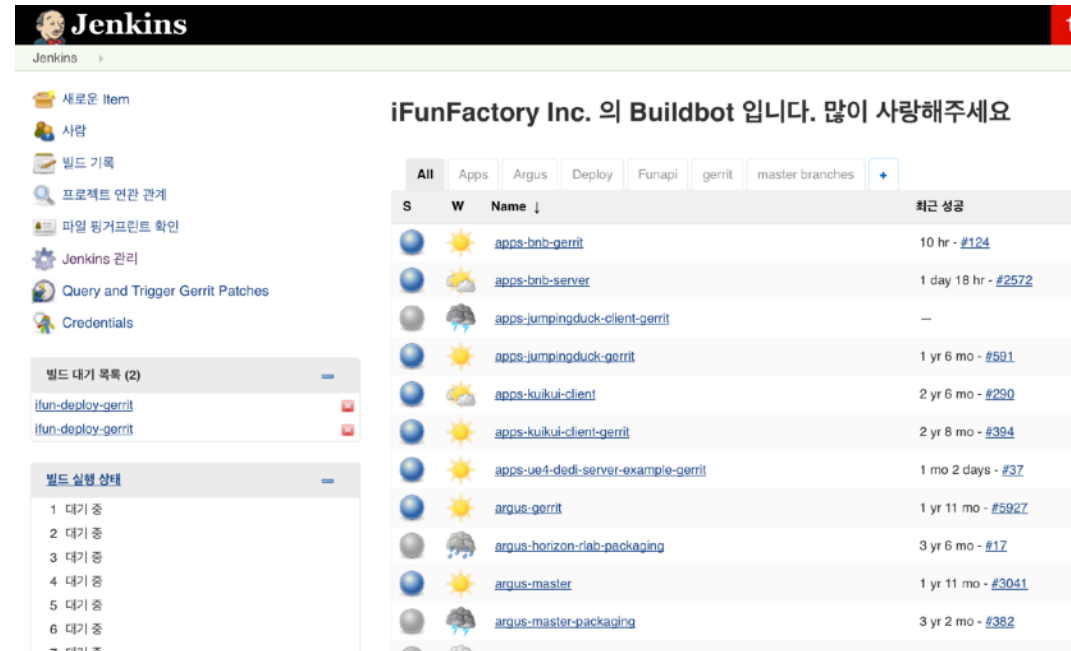
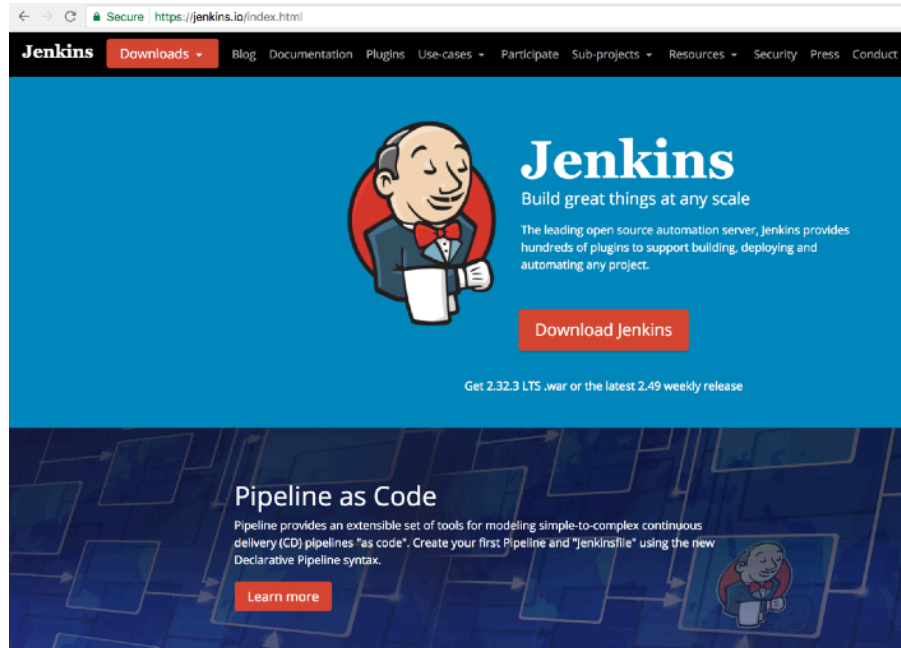
✓ Gerrit (설치형 git repository 및 코드 리뷰 시스템)

- Google 로부터 시작된 코드 리뷰 시스템
- 공유되는 git repository 로도 이용됨



Continuous Integration (CI)

Jenkins



✓ 저장소와 연동해서 commit 발생시 이를 자동으로 build 해주는 역할

✓ 직접 설치해야되지만, hosting 되는 형태로 travis 등의 솔루션도 있음



Project Mgmt Tools

✓ Redmine / Jira

- Issue tracker 의 성격을 가지면서 GANTT chart 나 activity tracking 등 project mgmt 를 위한 기능들을 제공
- Issue tracker 만으로는 Mantis 역시 활용됨

✓ Trello

- Scrum 툴로 활용

THANKS!

Great Technology For Great Games, **iFunFactory**



DK Moon



dkmoon@ifunfactory.com



www.ifunfactory.com

