
GameLift는 정말
알아서 움직였을까:
프리스타일 풋볼2 글로벌 게임 서버
재설계와 OBT 실전 기록
FREESTYLE
FOOTBALL2
유레

제가 만들어 온 게임

이충영

우레 · 서버 프로그래머

샷온라인 온네트

팡야 엔트리브소프트

트라이징 액토즈소프트

팡타지아 위메이드 크리에이티브

로스트사가 아이오엔터테인먼트

금빛 모자이크 메모리즈 kairos&co

윈조이 포커 Netmarble(zempot)

프리스타일 풋볼 2 우레

우레

목차

1 Amazon GameLift Servers를 선택한 이유

2 Amazon Aurora Global Database로 통합한 글로벌 백엔드

3 Amazon GameLift Servers 플릿 하나로
여러 지역에 경기 서버 배치

4 Amazon GameLift Servers와 AWS Graviton으로
줄인 비용과 운영 작업

5 Amazon Kinesis Data Streams와 Amazon DynamoDB로
연결한 기능별 서버

6 서비스 중에 바꾸는 서버 버전·매칭 룰·배치 기준

7 Amazon GameLift Servers와 Amazon Kinesis Data Streams
운영 중 겪은 문제와 해결

우테

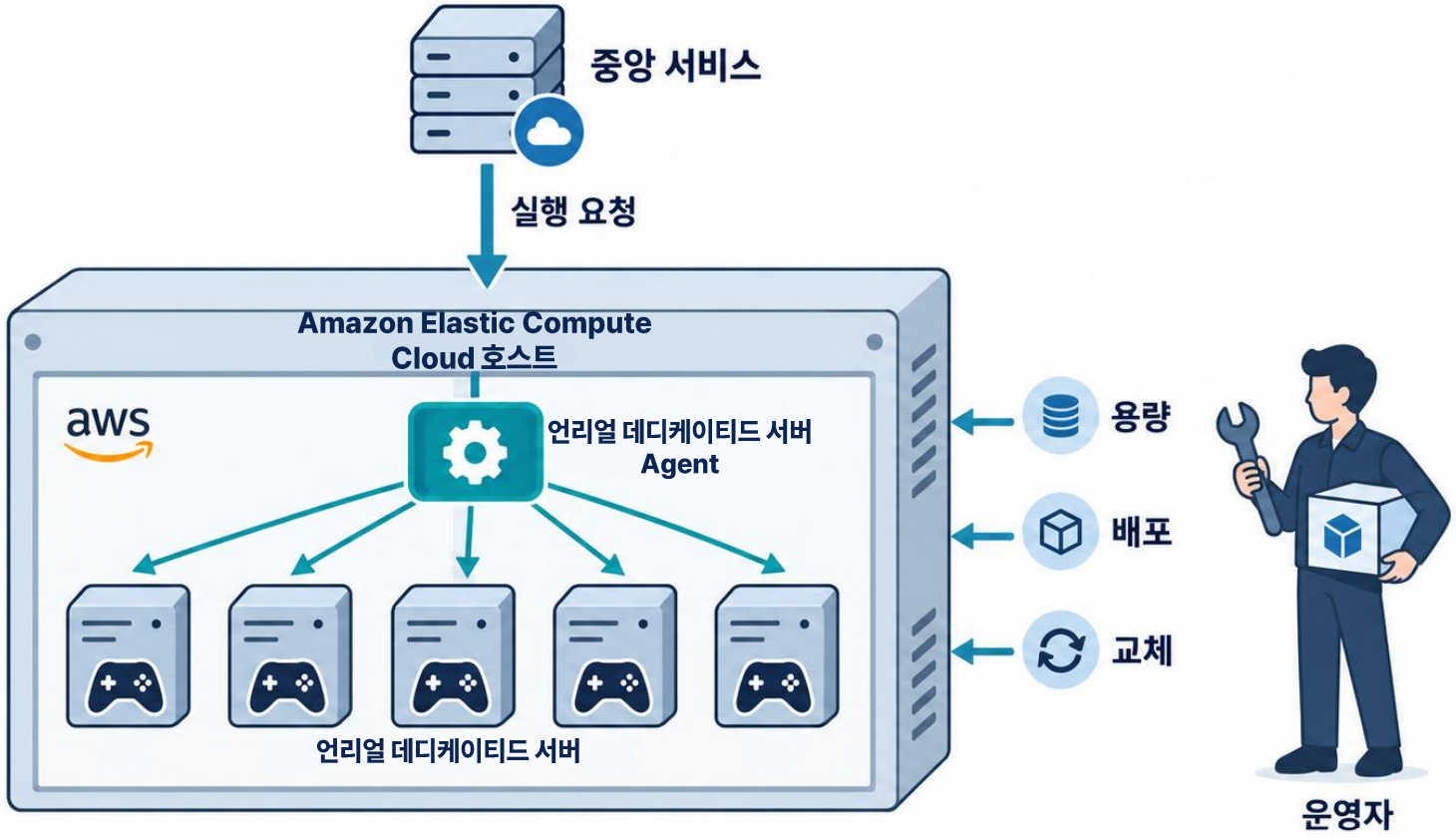


ⓒ JOYCITY Corp. & WOORE. All rights reserved.

01 Amazon GameLift Servers를 선택한 이유

우레

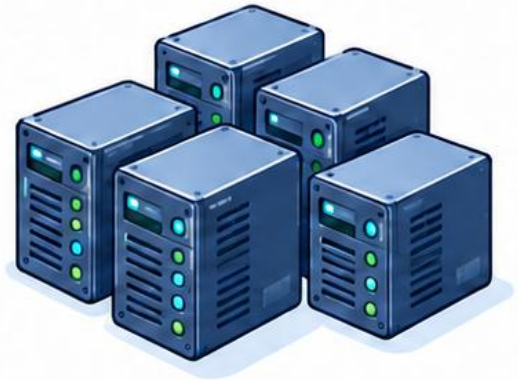
언리얼 데디케이티드 서버 Agent의 실행 관리와 사람의 호스트 운영



유저 수와 경기 수에 따른 언리얼 데디케이티드 서버 수의 차이

게임 서버

약 10대

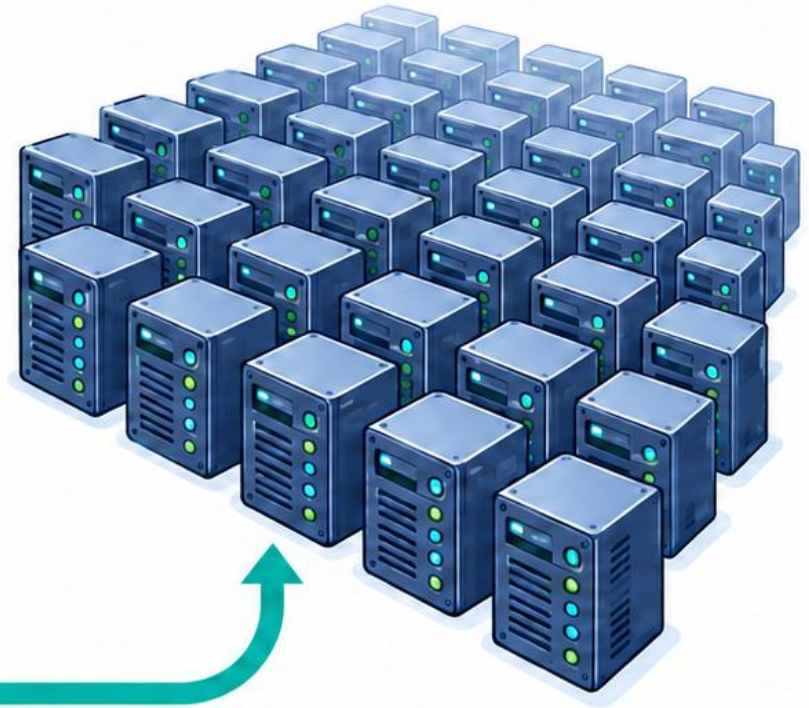


사람 1명 + AI 9명



언리얼 데디케이티드 서버

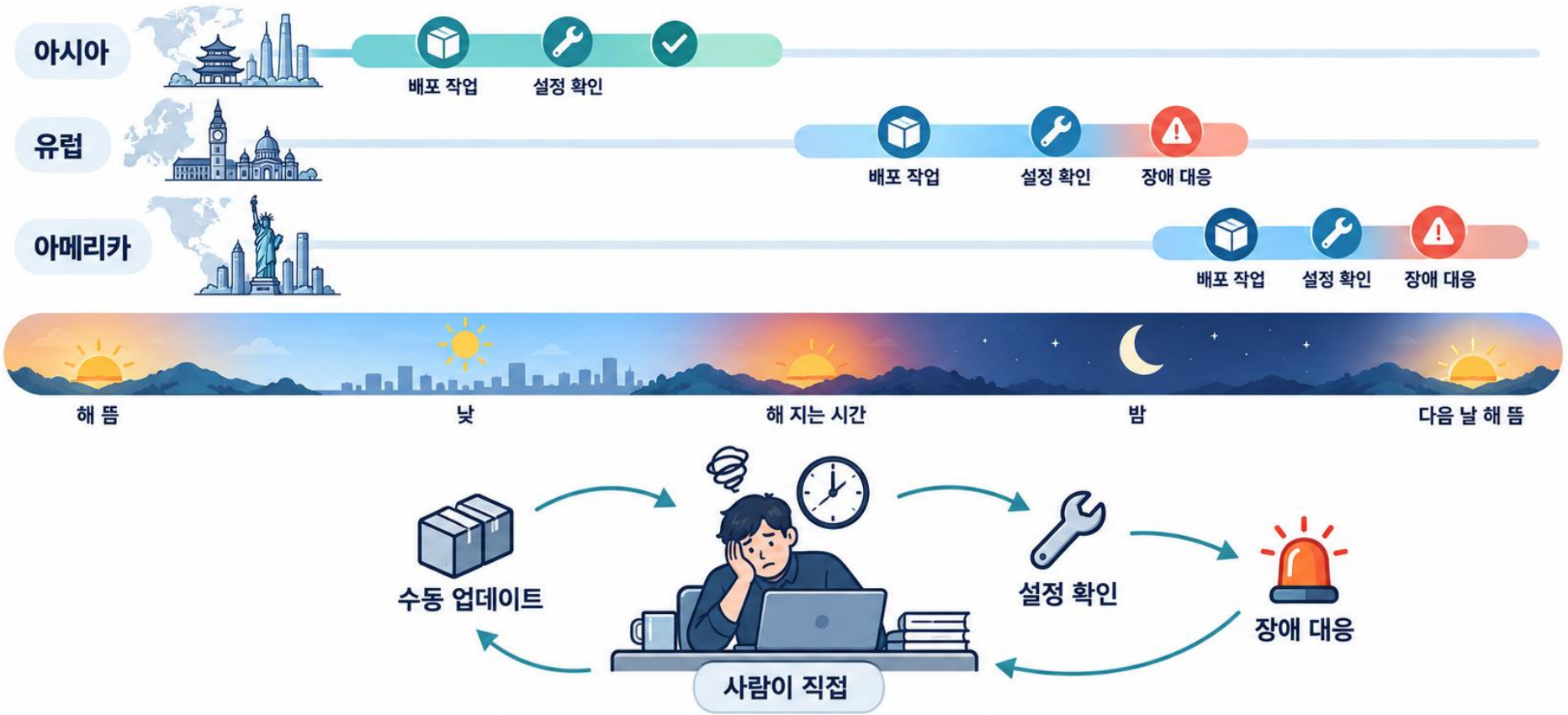
예상 1,000~3,000개





경기 1개 → 서버 프로세스 1개

지역 확대로 늘어난 수동 운영 작업



전 세계 서버의 실행과 용량 관리

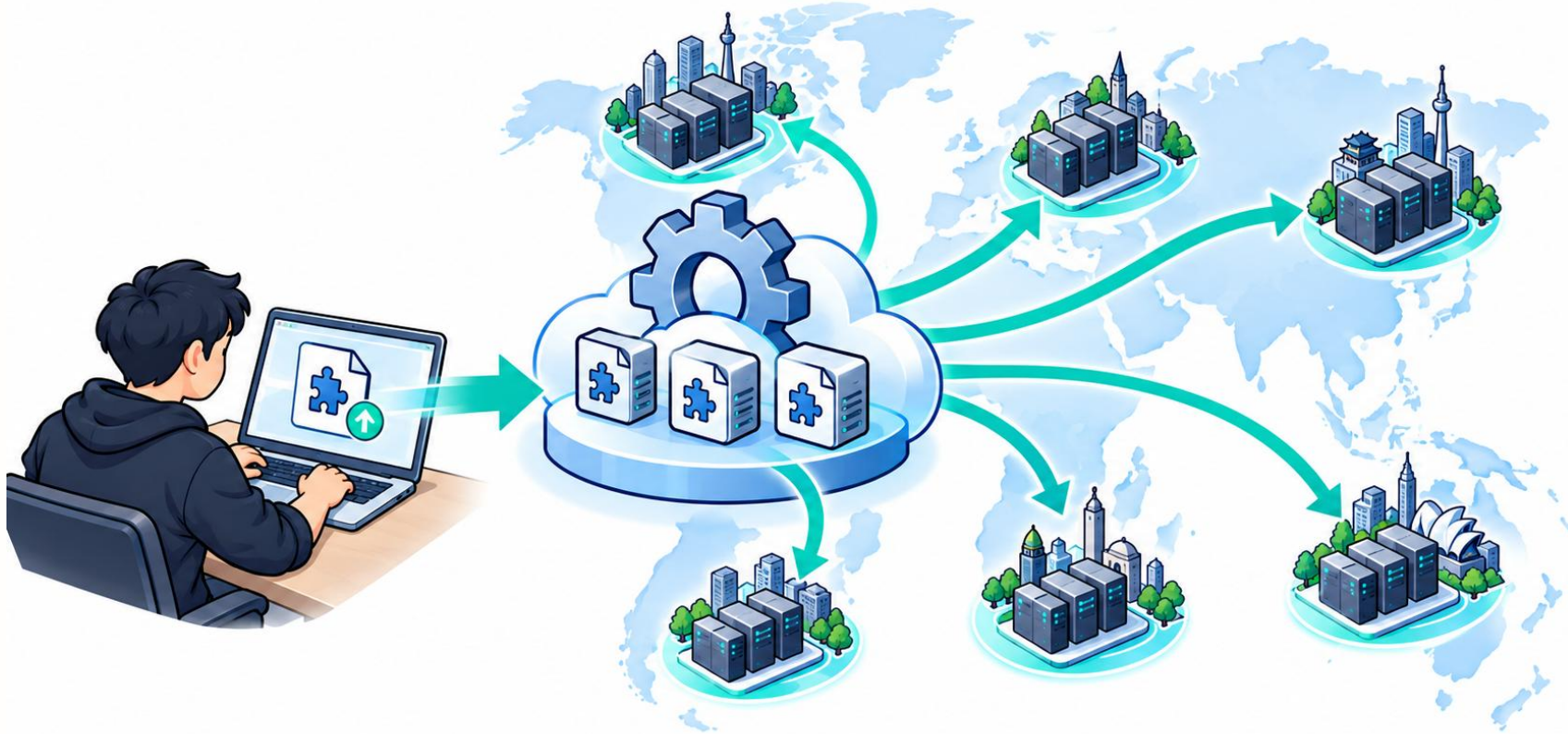
Amazon GameLift Servers 선택 이유 1 / 3

서버 실행

용량 관리

배포 버전 관리

여러 지역을
같은 방식으로 운영



네트워크 품질에 따른 경기 서버 배치

Amazon GameLift Servers 선택 이유 2 / 3

여러 국가의 유저

지역별 네트워크 품질

게임 세션 큐의 배치

함께 플레이할 유저에게
적합한 서버 선택



접속량에 따른 용량 조절과 운영 부담 감소

Amazon GameLift Servers 선택 이유 3 / 3

접속 증가 → 확장

접속 감소 → 축소

배포 도구 제작 용이

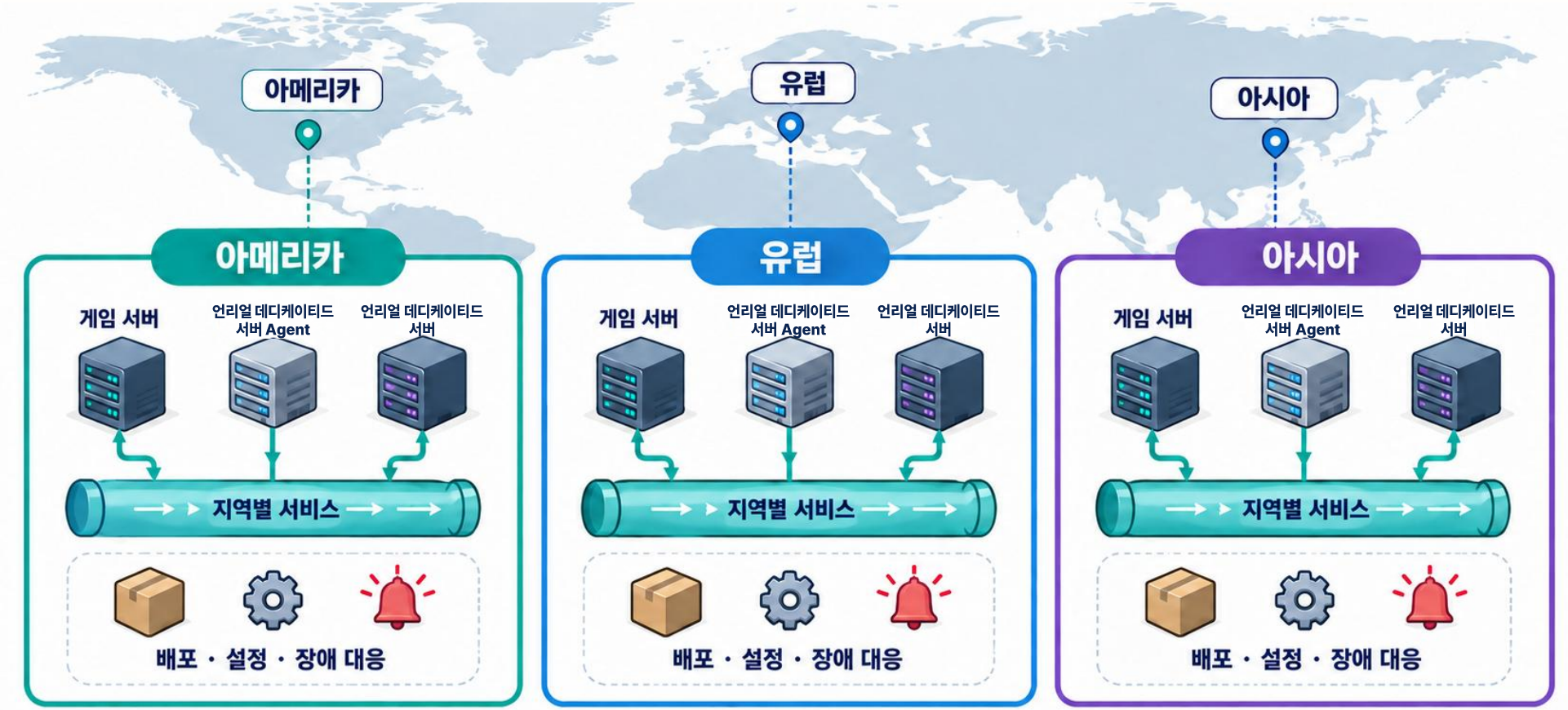


상시 가동 비용과
운영 부담 감소



02 Amazon Aurora Global Database로 통합한 글로벌 백엔드

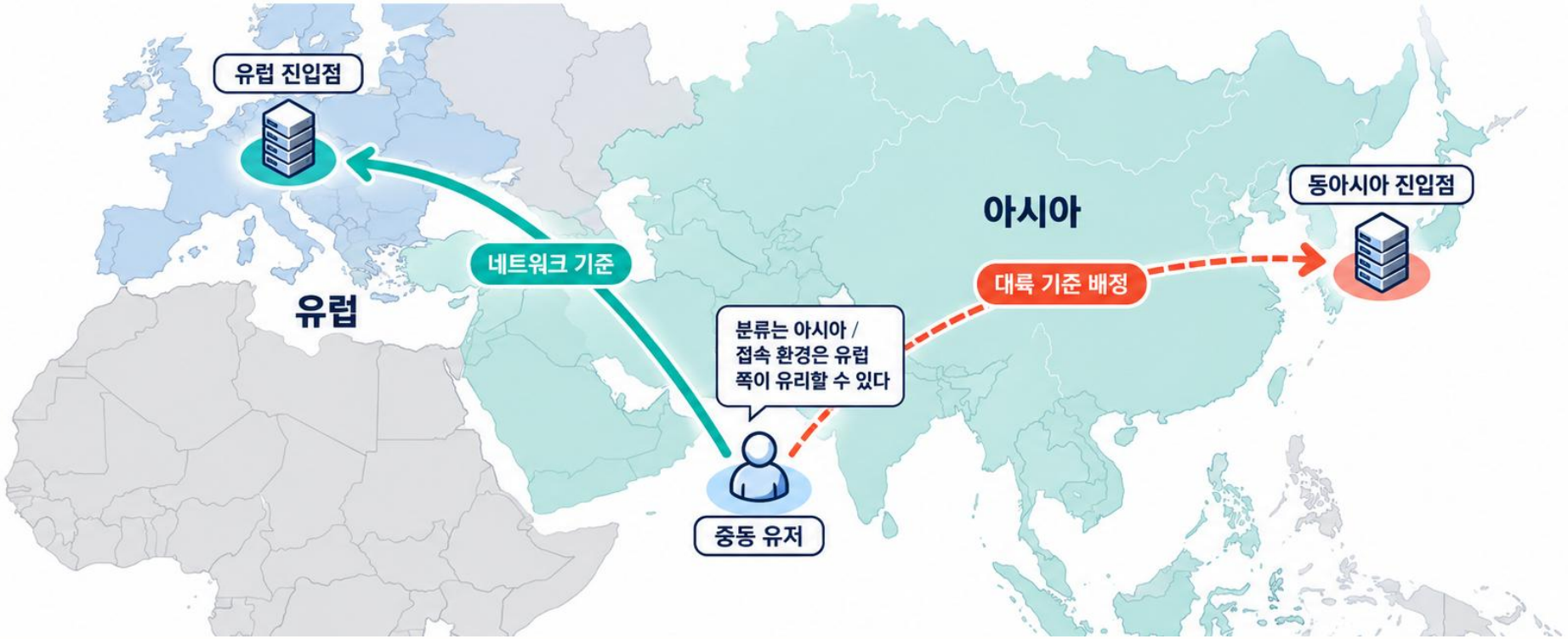
CBT의 아시아·유럽·아메리카 분리 운영



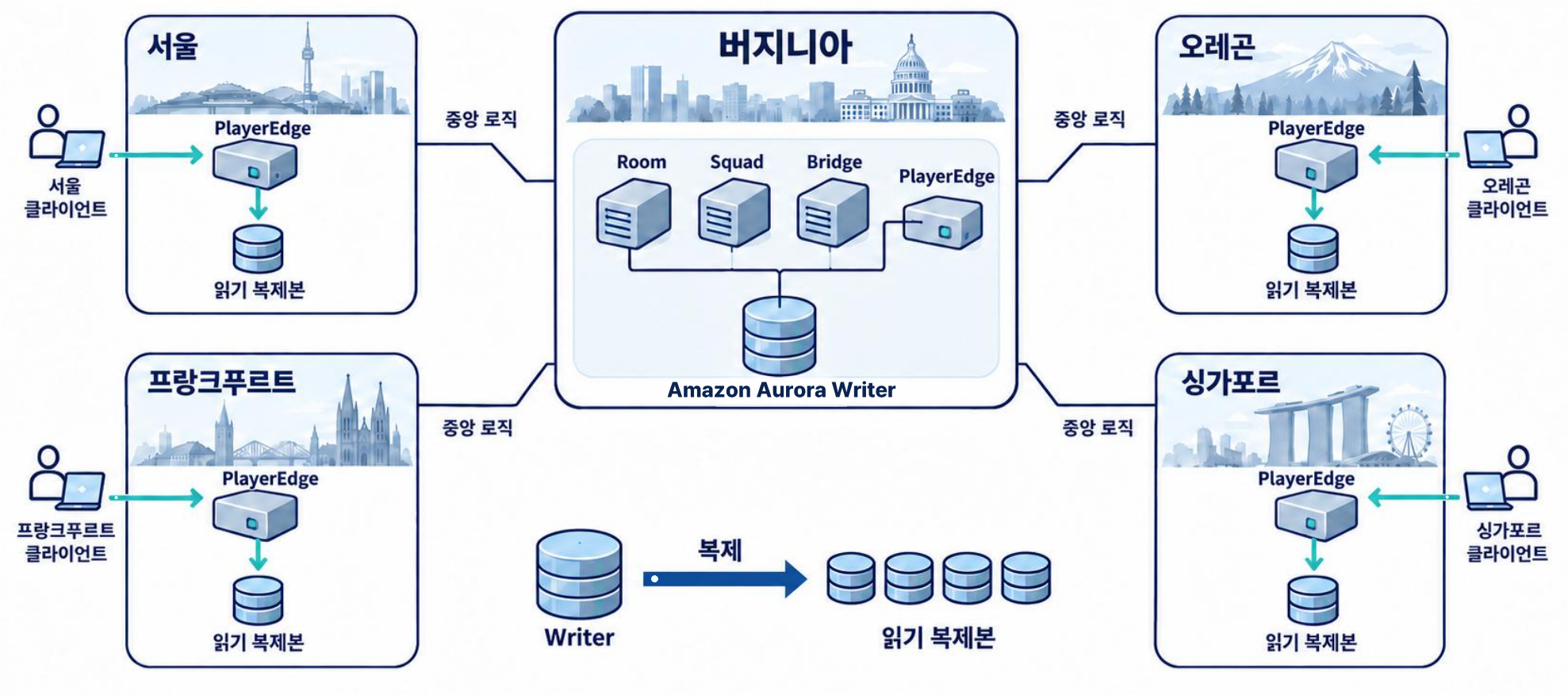
대륙 구분과 실제 접속 환경의 차이

대륙 구분 (정치적 기준)
국가와 지역을 행정상 대륙으로 분류

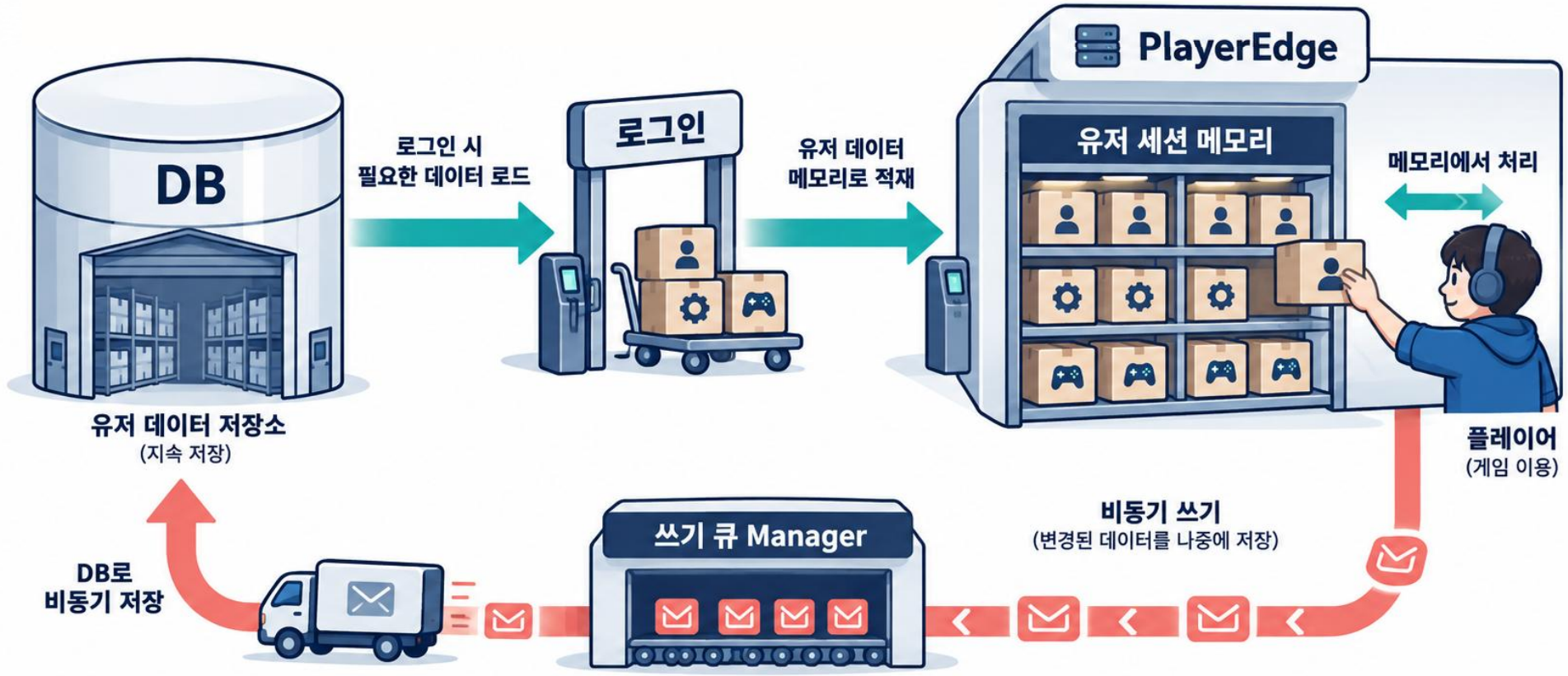
접속 환경 (네트워크 기준)
실제 접속 경로와 품질은 네트워크 환경에 따라 달라질 수 있음



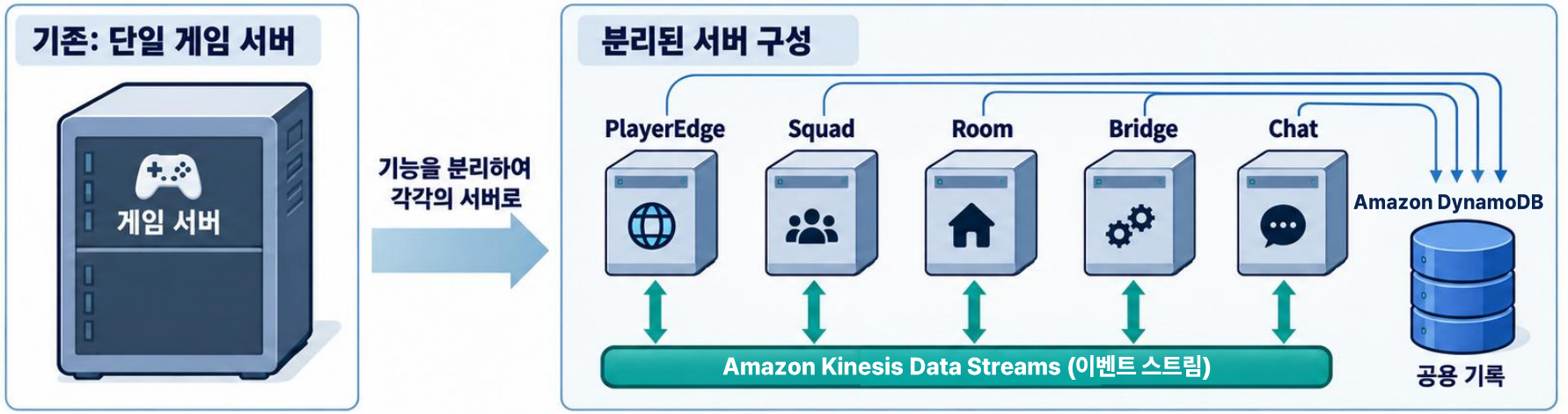
Amazon Aurora Global Database의 중앙 쓰기와 지역별 읽기



로그인 시 메모리 적재와 비동기 쓰기



기능별 서버 분리와 지역별 접속 분산



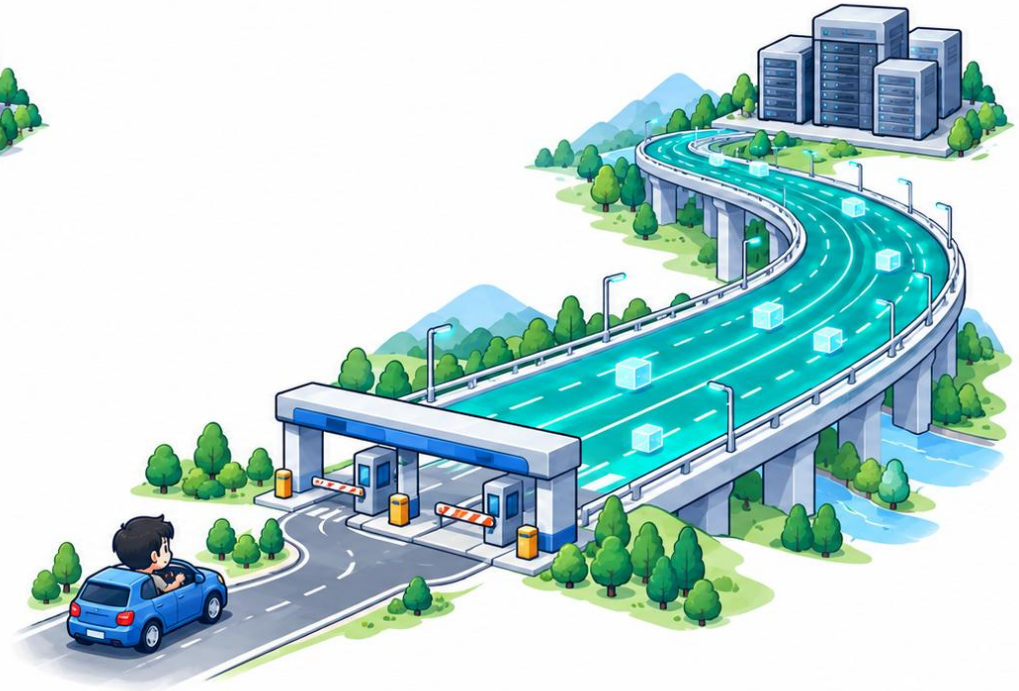
가까운 입구를 통한 리전 간 통신

국도 · 버지니아까지 직접 접속



고속도로 · 가까운 PlayerEdge를 거쳐 접속

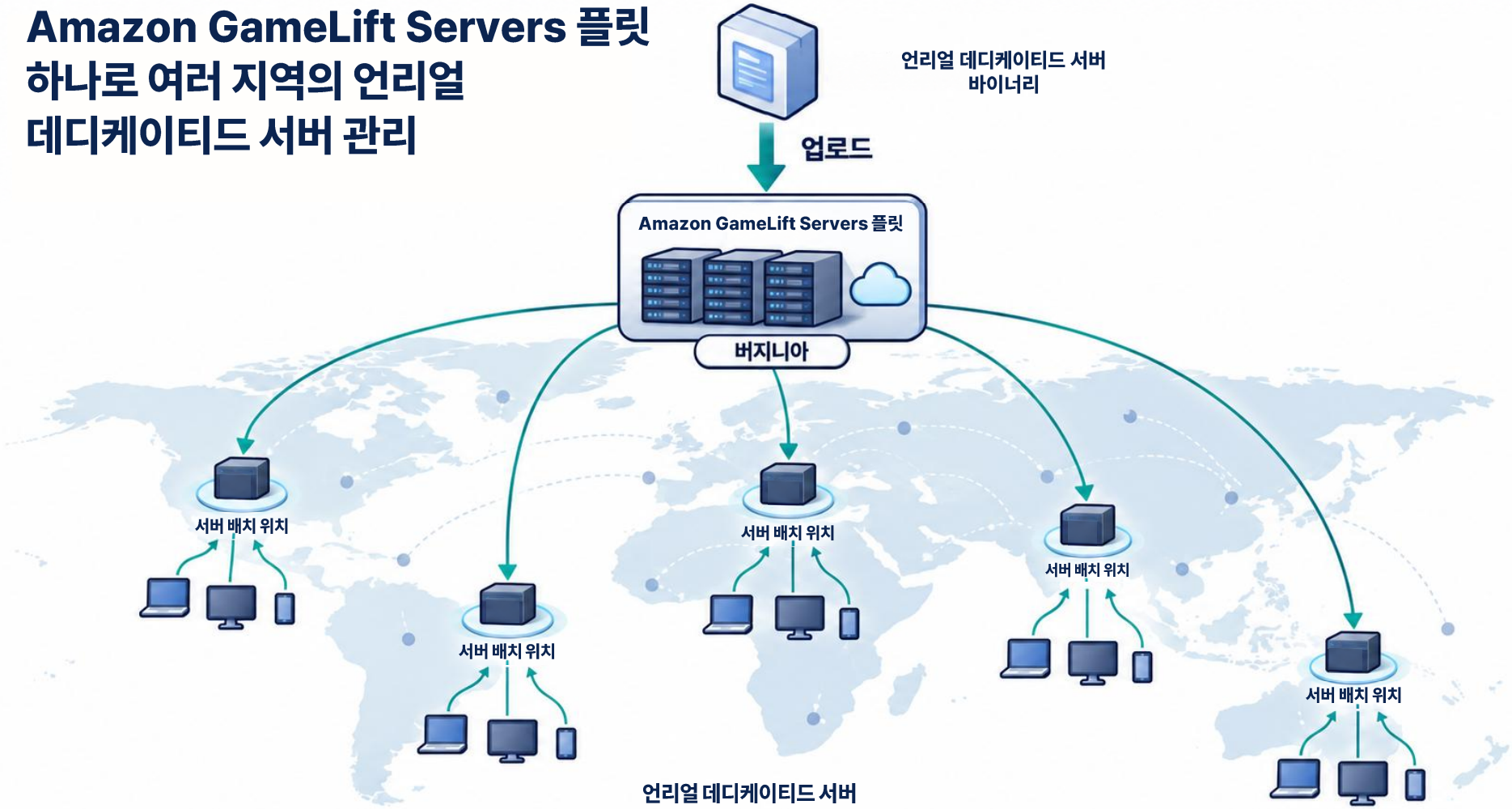
Amazon Web Services의 리전 간 통신망





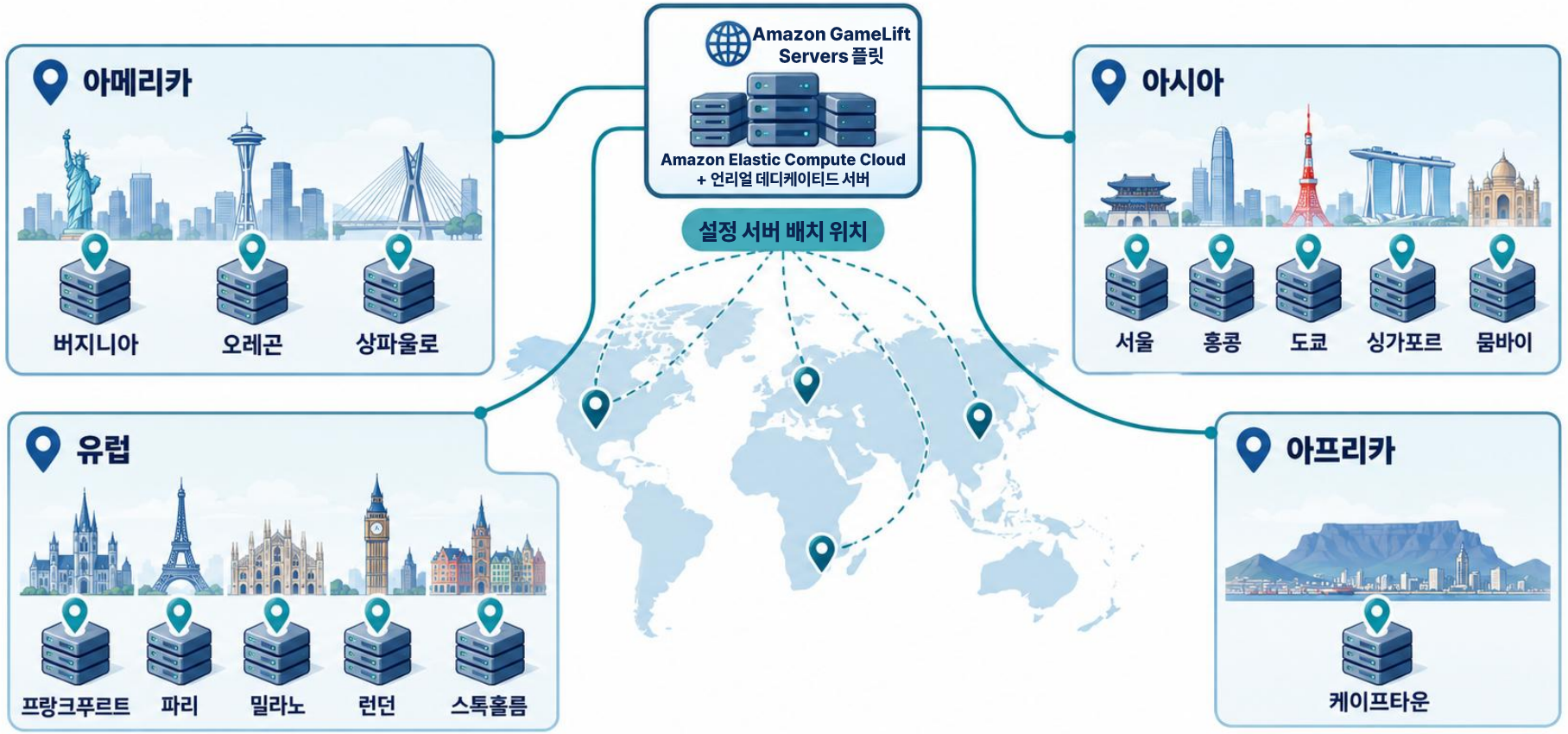
03 Amazon GameLift Servers 플릿 하나로 여러 지역에 경기 서버 배치

Amazon GameLift Servers 플릿
하나로 여러 지역의 언리얼 데디케이티드 서버 관리

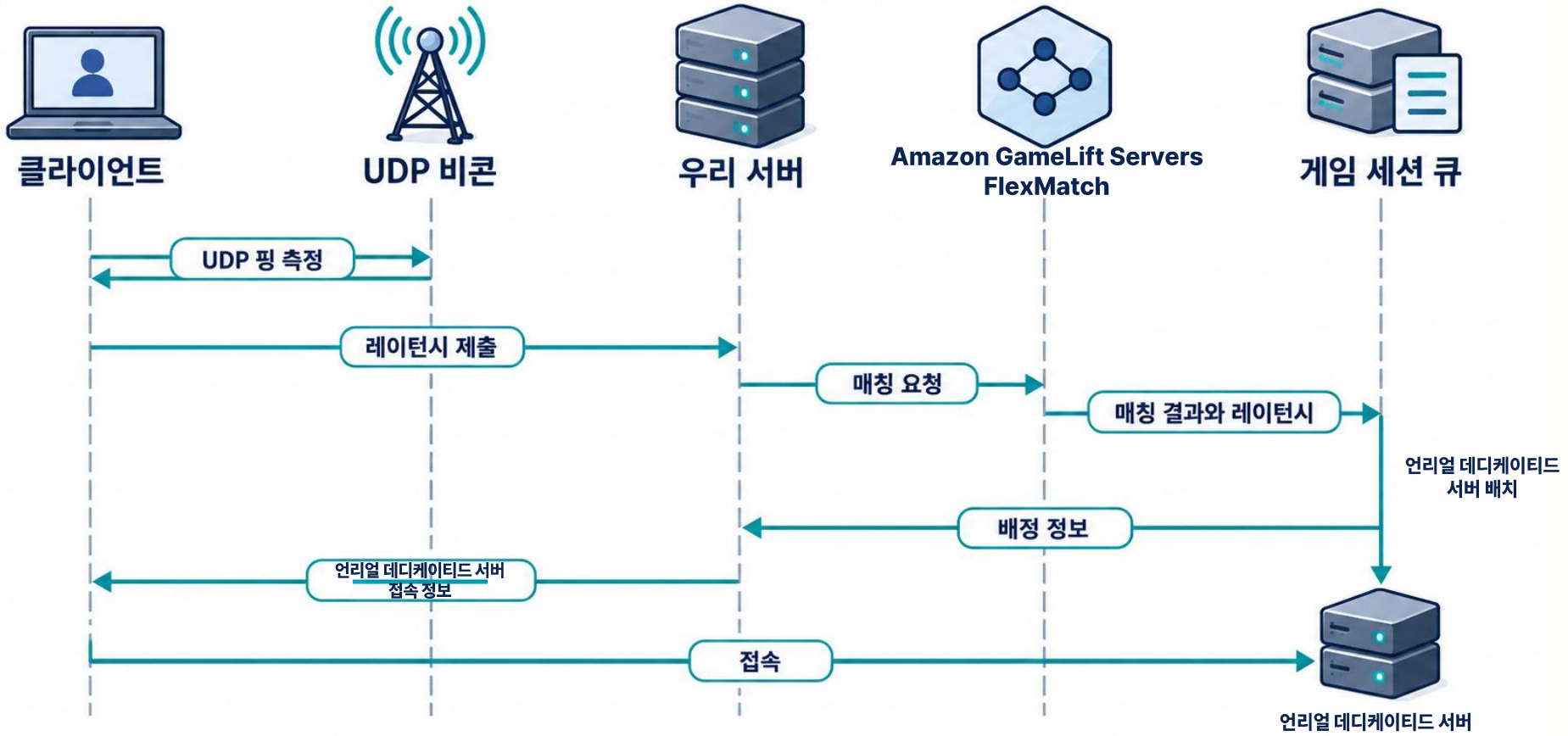


지역마다 Amazon GameLift Servers 플릿을 만들지 않았다 · 버전·슬롯별 플릿은 공존 가능

설정한 서버 배치 위치별 언리얼 데디케이티드 서버 배치

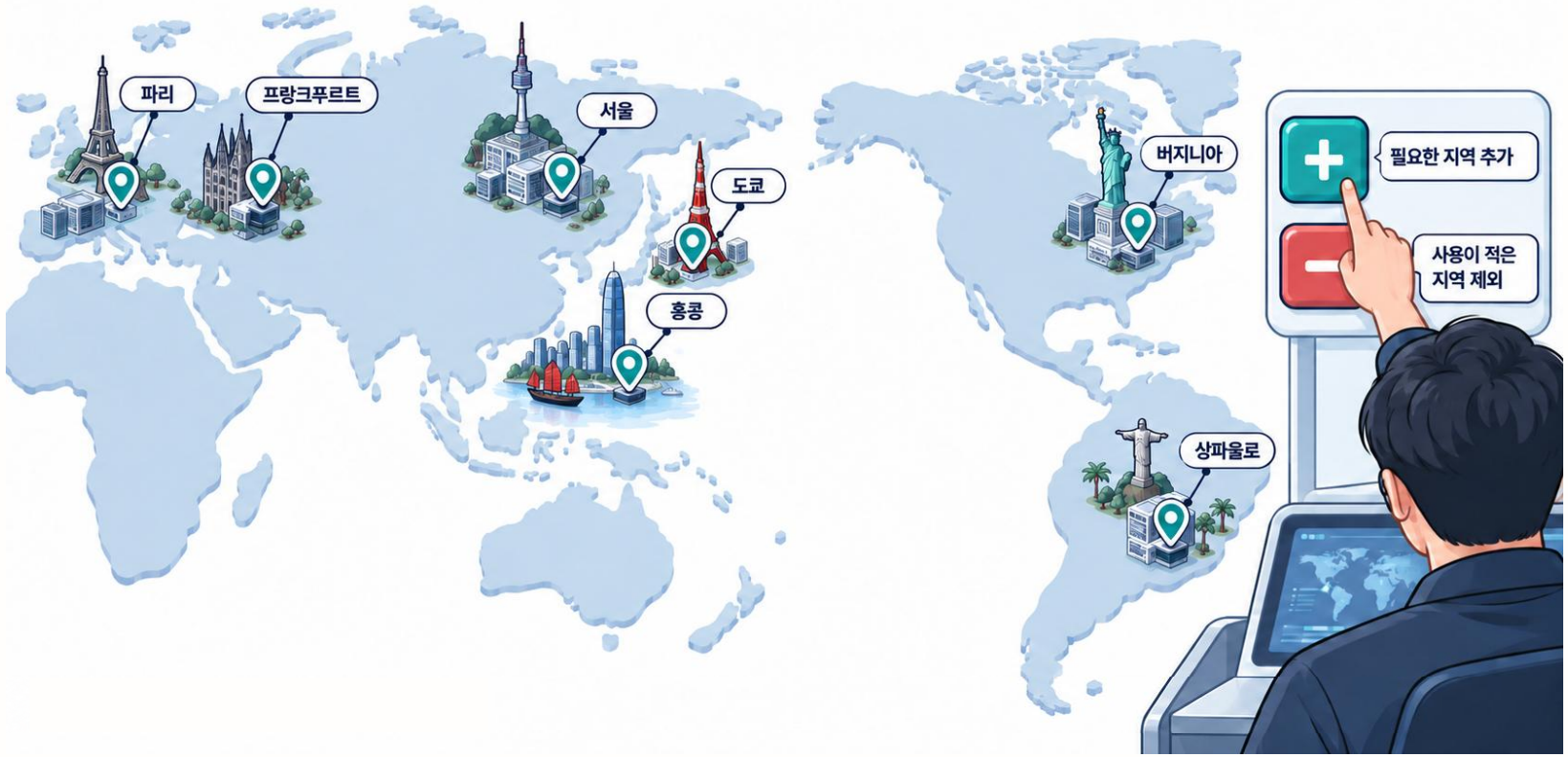


FlexMatch 매칭부터 게임 세션 큐 배치와 Amazon EventBridge 통지까지

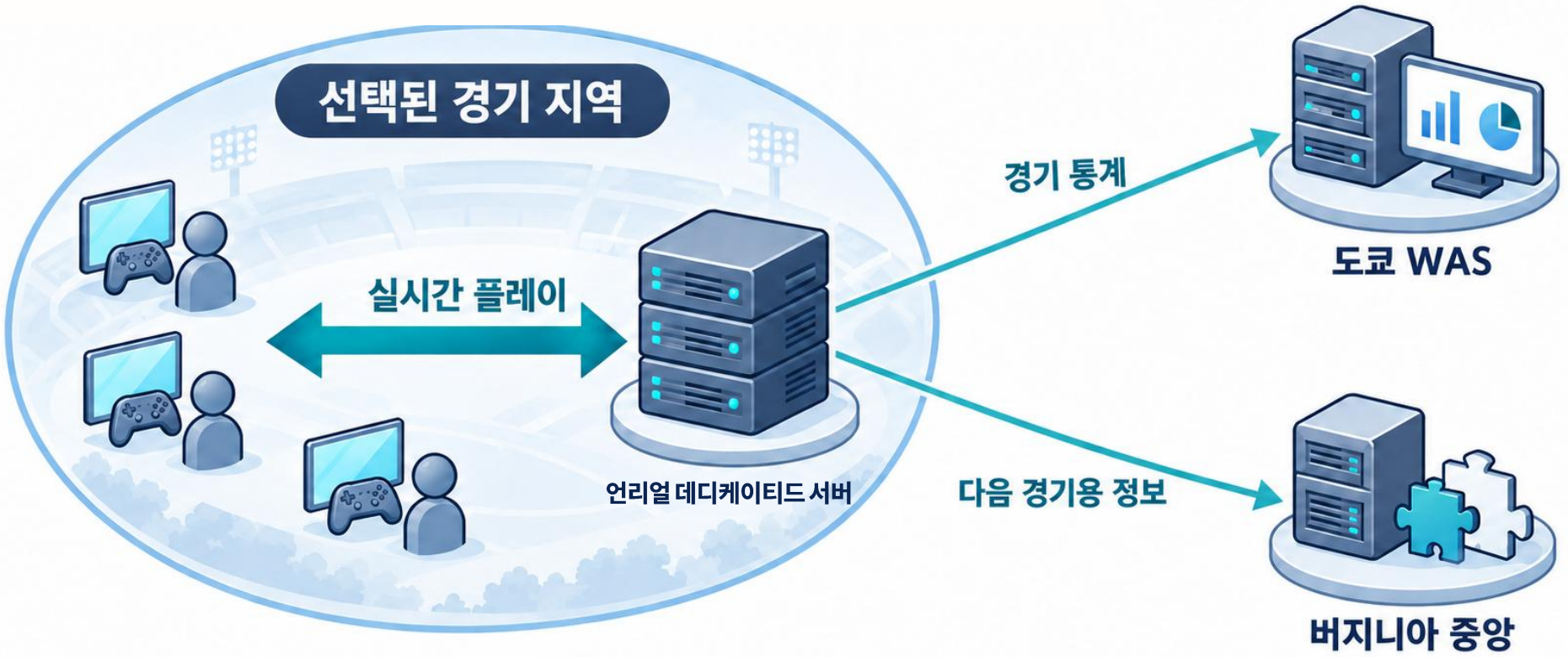


Steam Next Fest 사용량에 따른 운영 리전 조정

20개 지역 · Steam Next Fest 운영 회고



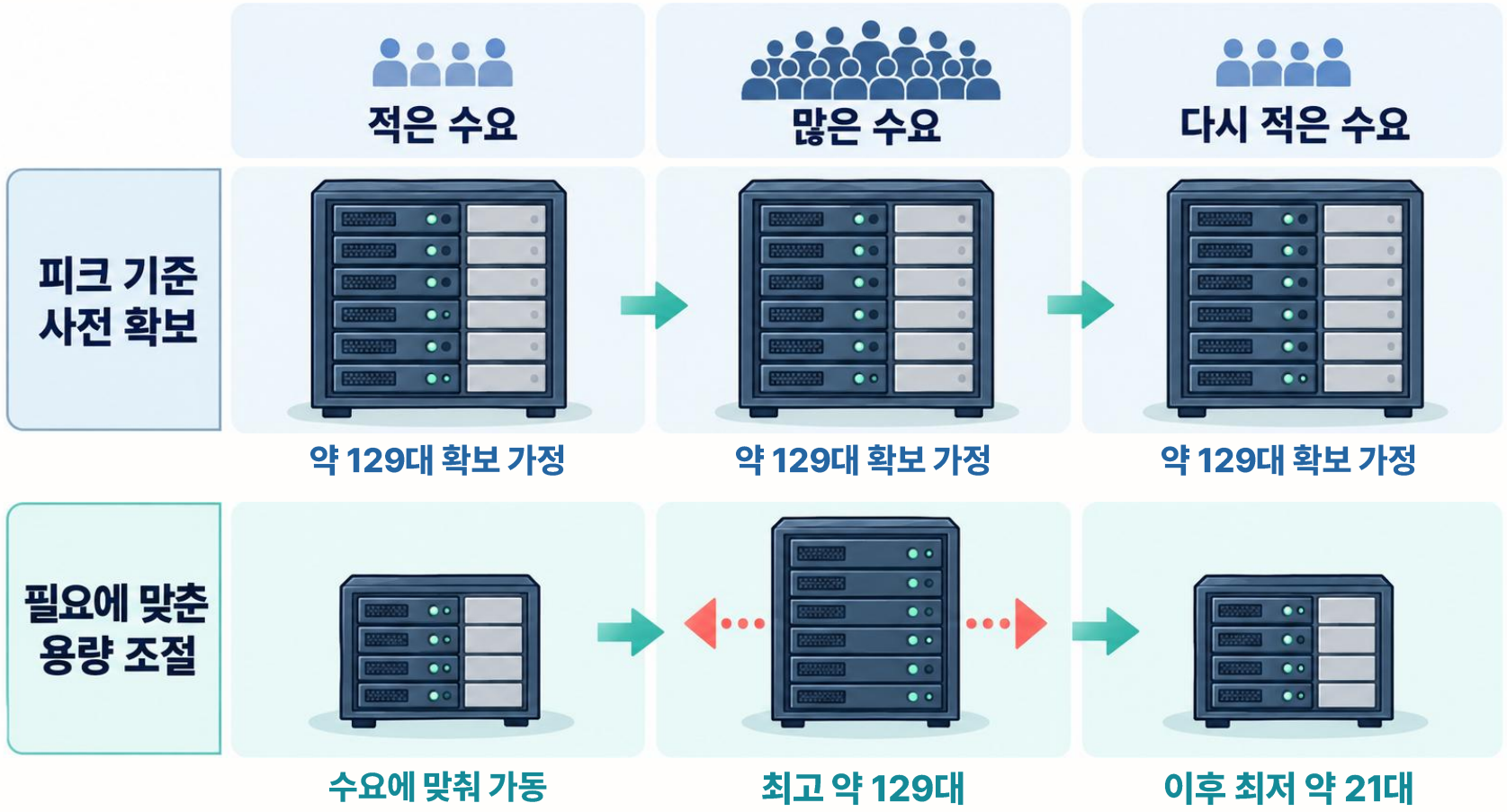
지역별 실시간 플레이와 중앙 결과 처리



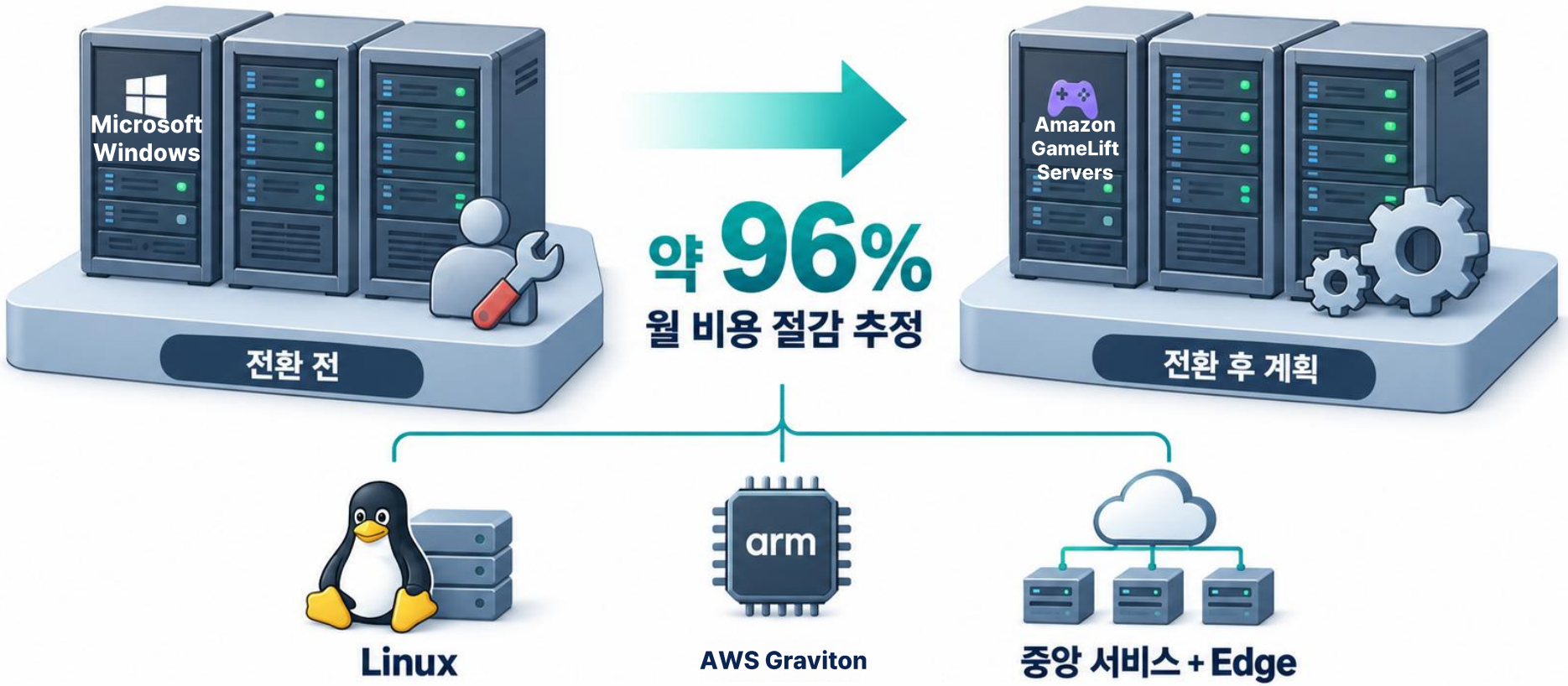


04 Amazon GameLift Servers와 AWS Graviton으로 줄인 비용과 운영 작업

피크 기준 사전 확보와 수요별 용량 조절



전환 전 구성과 전환 후 계획 비용 비교



전환 전 · 호스트마다 직접 수행한 운영 작업

이전 / 직접 작업

직접 설치·실행·확인



호스트마다 반복

- 호스트 확보
- 호스트별 에이전트 설치
- 프로세스 실행·종료
- 호스트별 업데이트 확인
- 상태 감시
- 지역별 용량 조절

매칭 서버와 경기 지역 연결도 직접 운영

전환 후 · 플릿 단위로 맡긴 운영 작업

이후 / 서비스에 위임

Amazon GameLift Servers



호스트별 직접 작업

같은 작업 · 달라진 담당



플릿 용량 확보



호스트 실행 환경 관리



프로세스 실행·종료



제공한 빌드 배포



상태 정보 제공



규칙에 따른 용량 조절

우리 몫: 빌드 제공 · 플릿 설정 · 버전 전환

지연 시간에 따른 유저 매칭과 경기 서버 배치

매칭은 유저 조합 · 배치는 경기 서버 선택

지역별 지연 시간 전달

Amazon GameLift Servers
FlexMatch

Amazon GameLift Servers
게임 세션 큐



클라이언트 측정값

규칙에 맞는 유저 조합

지연 시간·가용 용량에 따른 배치

함께 플레이할 유저 결정 → 경기할 데디케이티드 서버 배치

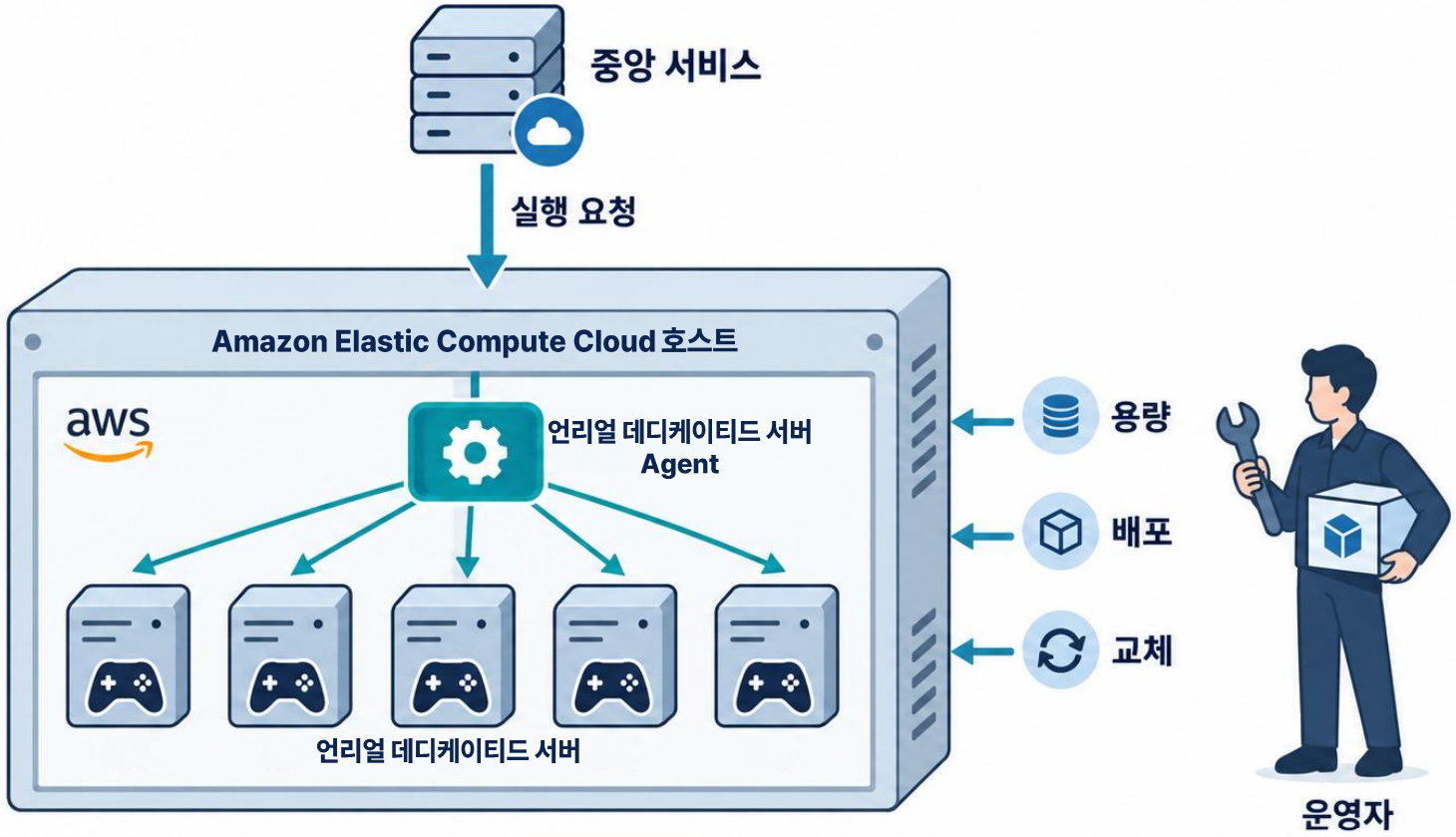
직접 만든 Amazon GameLift Servers의 플릿·게임 세션 큐·버전 관리 도구



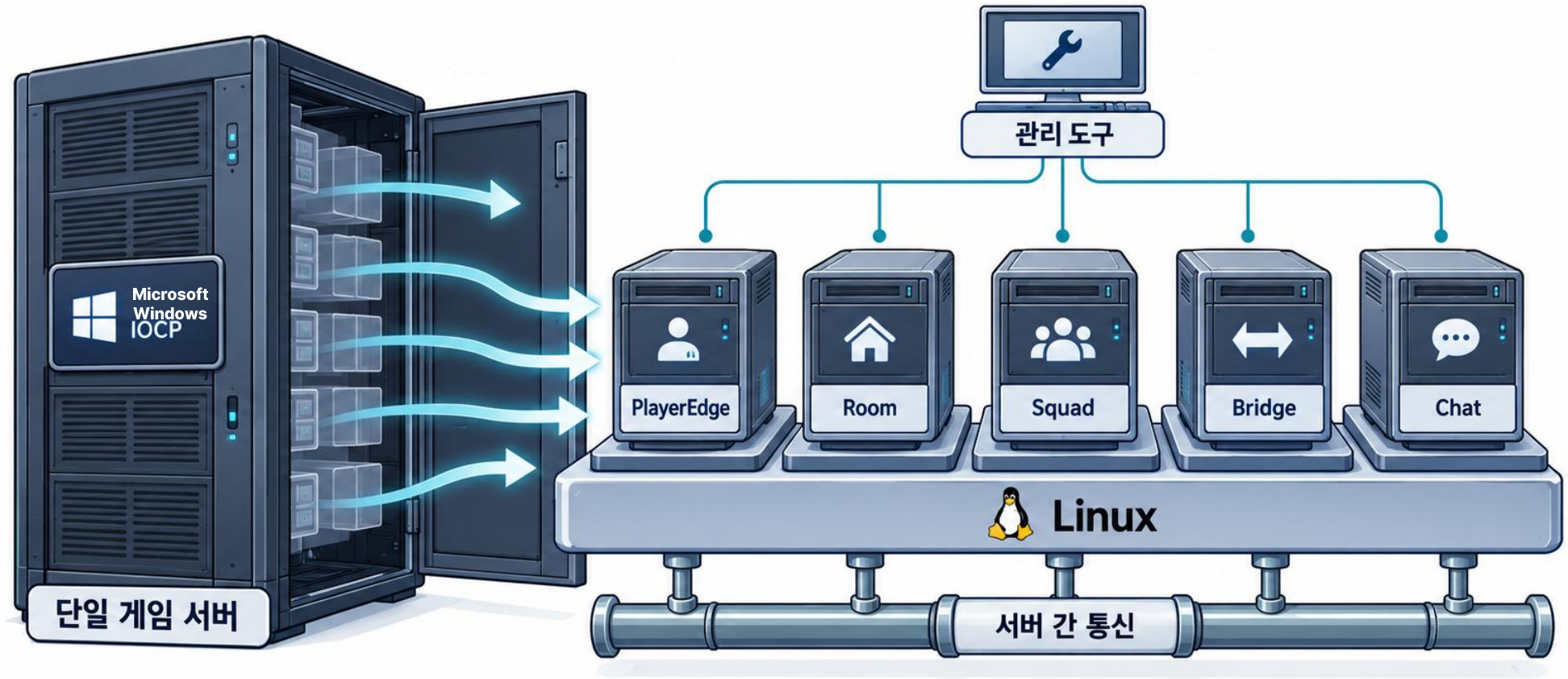


05 Amazon Kinesis Data Streams와 Amazon DynamoDB로 연결한 기능별 서버

기능 분리 전 Microsoft Windows IOCP 단일 서버

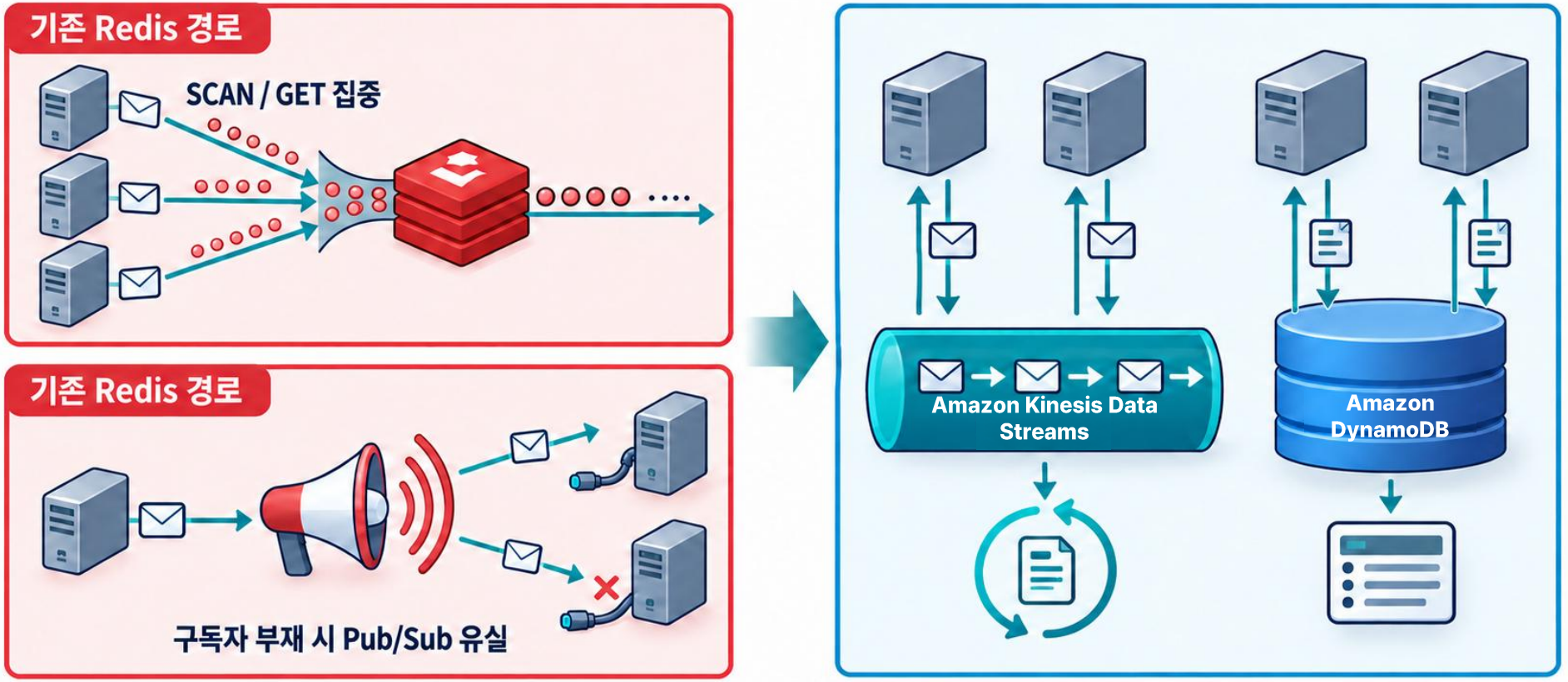


Linux 전환과 기능별 서버 분리

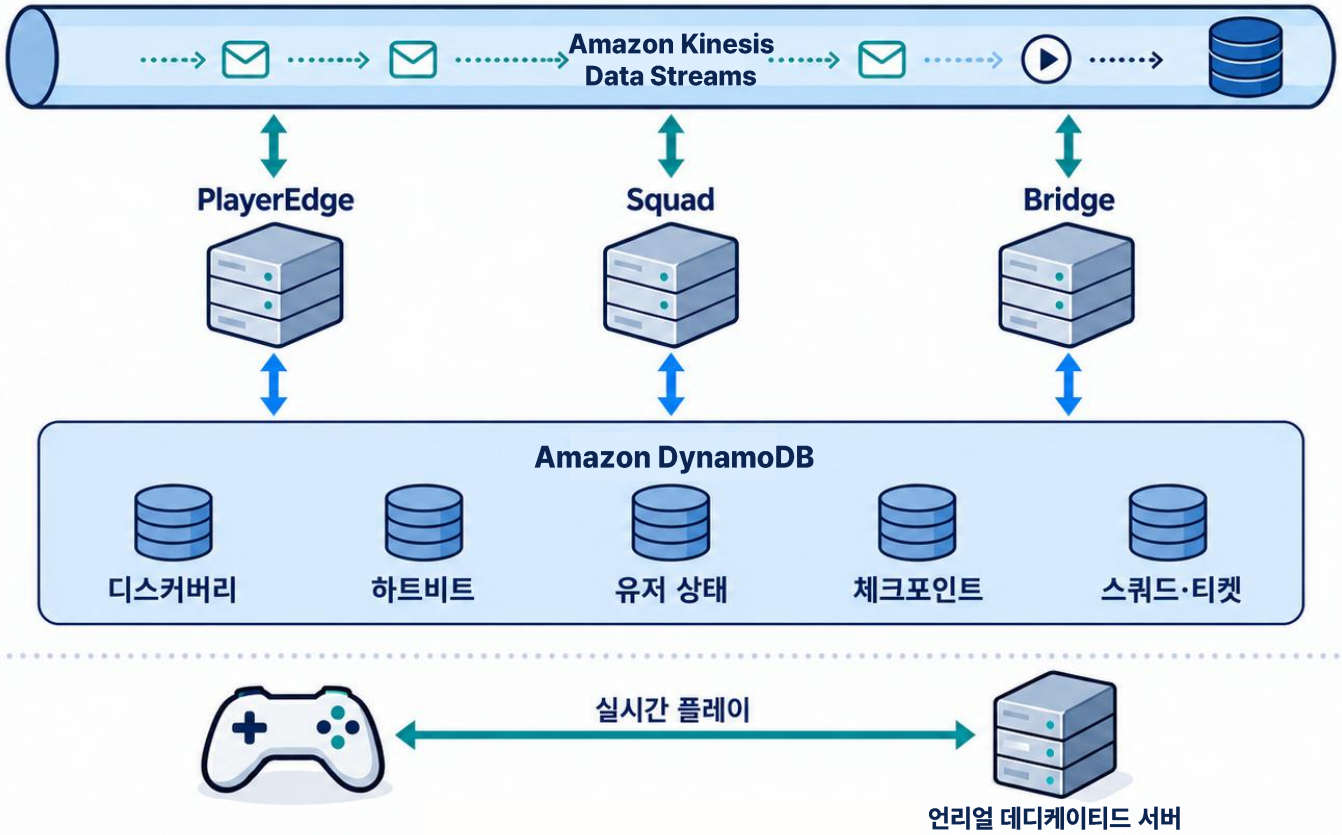


프로세스를 나누자 통신과 운영 도구가 필요해졌다

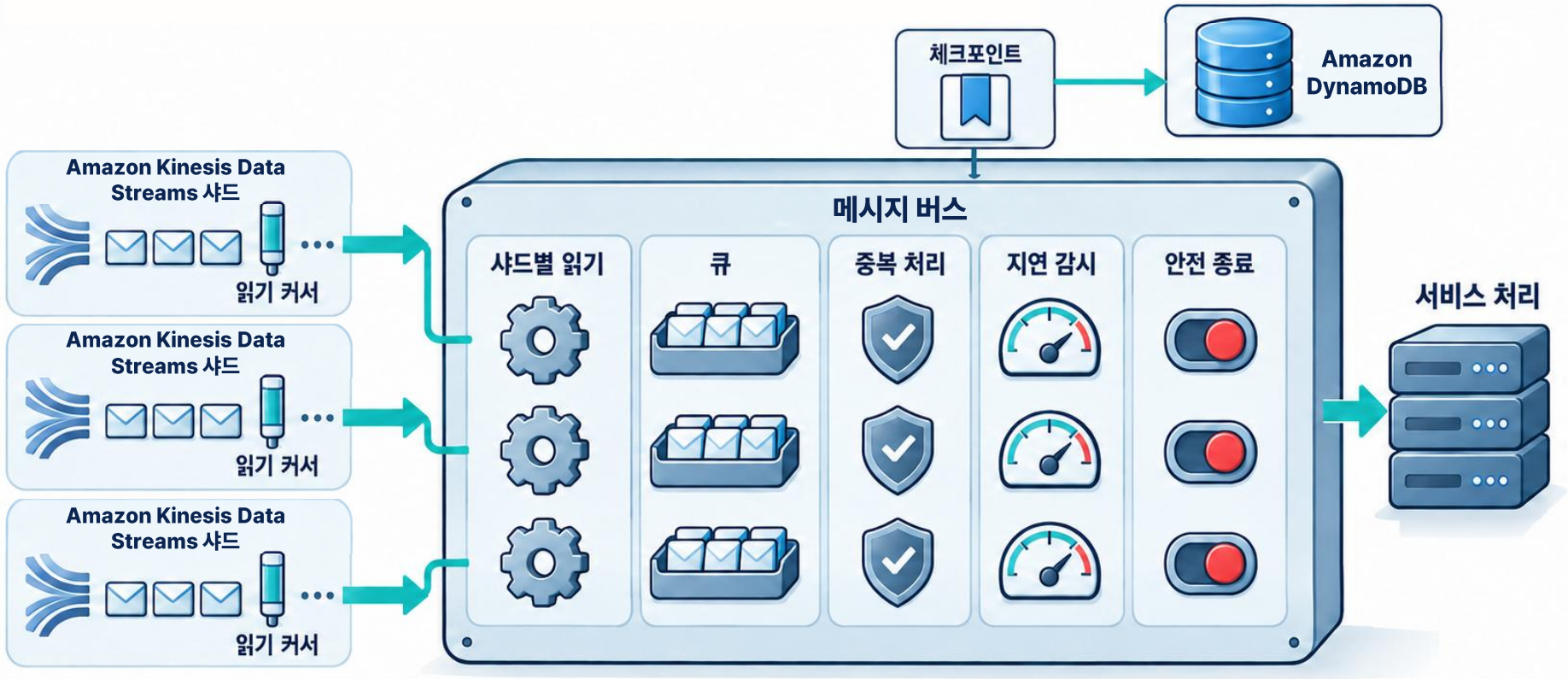
Redis의 문제와 Amazon Kinesis Data Streams 전환



Amazon Kinesis Data Streams의 메시지 전달과 Amazon DynamoDB의 상태 저장



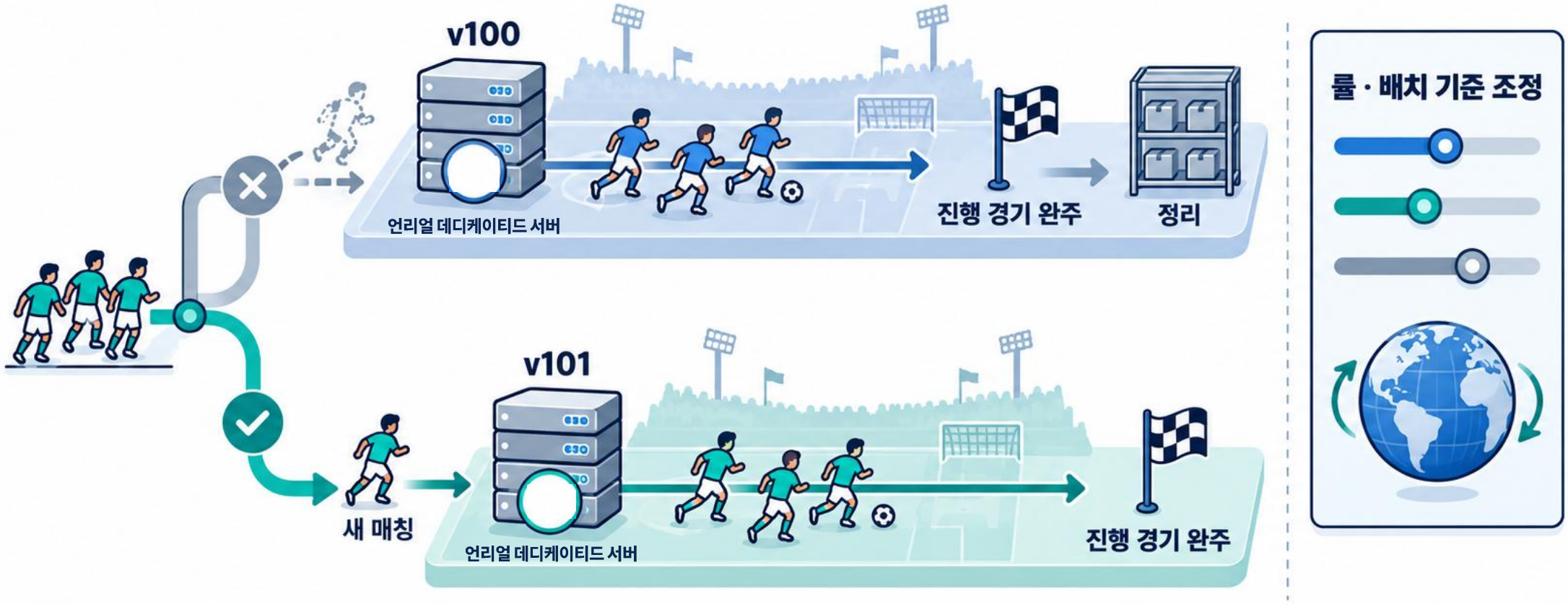
C++ 메시지 버스의 직접 구현





06 서비스 중에 바꾸는 서버 버전·매칭 룰·배치 기준

진행 중 경기 보존과 새 버전으로의 매칭 전환



진행 중인 경기와 새 매칭을 분리

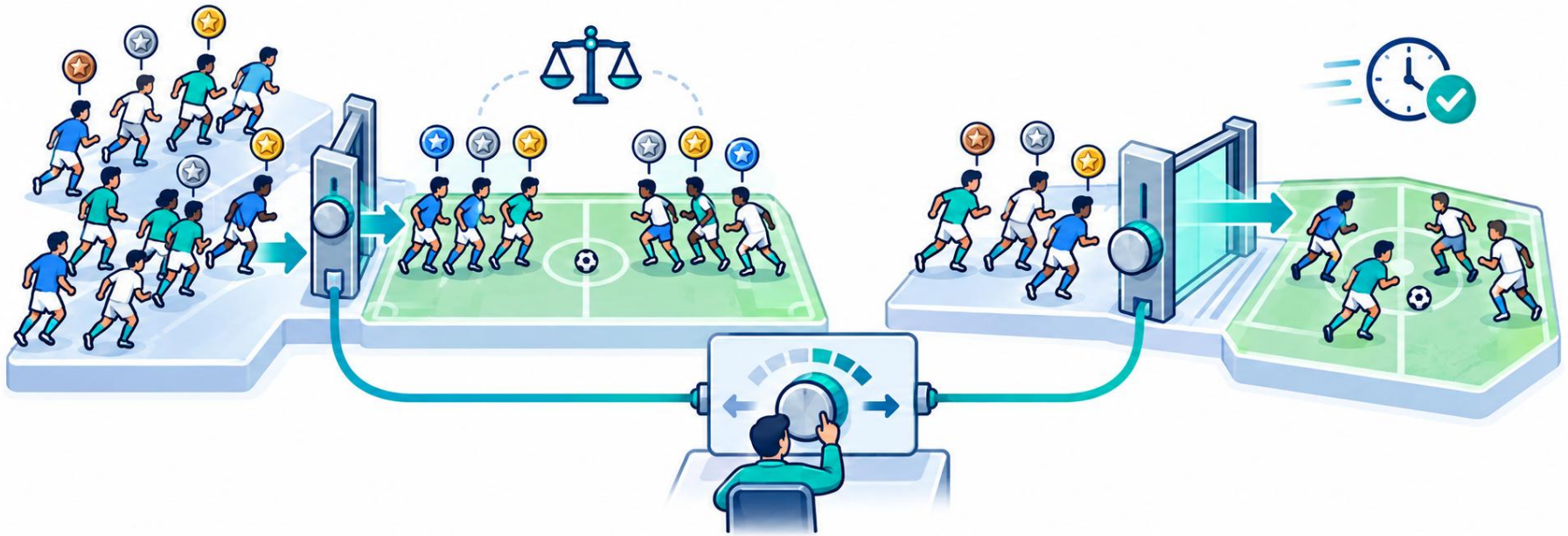
FlexMatch 룰 변경으로 조절하는 매칭 조건

유저가 많을 때

실력이 비슷한 유저끼리 매칭

유저가 적을 때

실력 차이를 허용해 빠르게 매칭



서비스 중 매칭 룰 변경

게임 세션 큐의 지연 시간 정책으로 조절하는 경기 지역

유저가 적을 때 · 댈러스로 연결

남아메리카 유저의 배치 예시



유저가 많을 때 · 상파울로에서 플레이

남아메리카 유저의 배치 예시

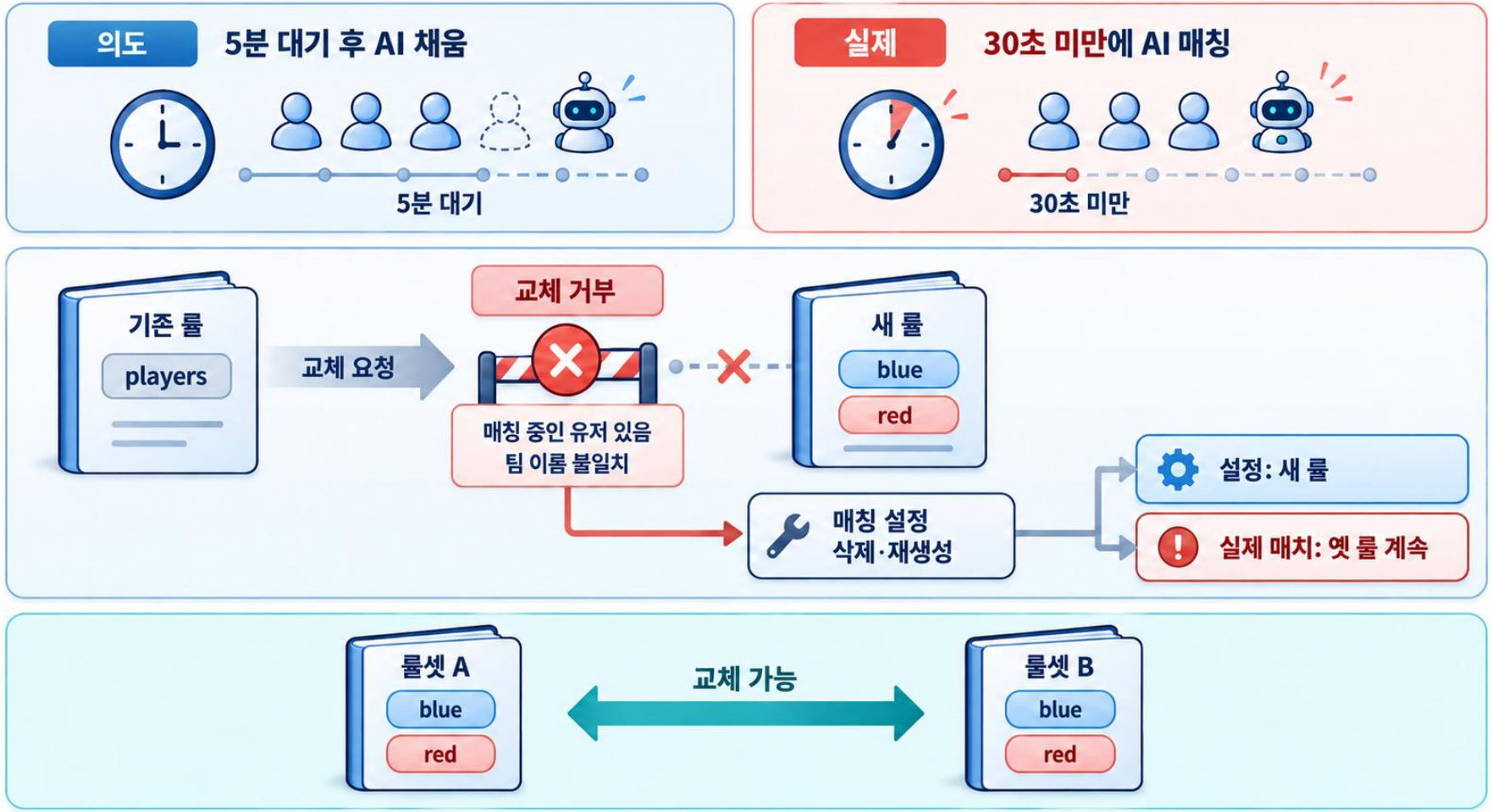


유저 수에 맞춰 배치 기준 변경



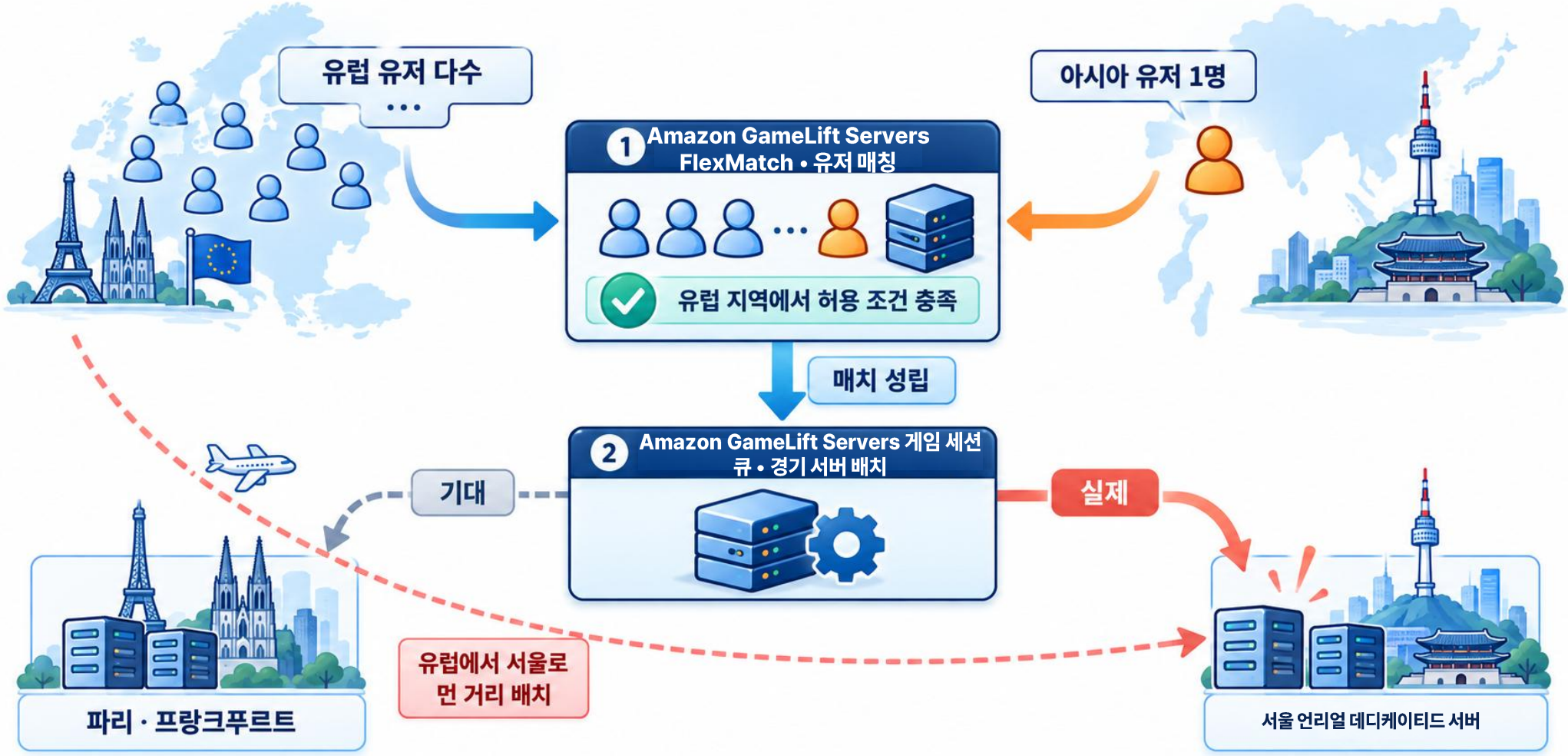
07 Amazon GameLift Servers와 Amazon Kinesis Data Streams 운영 중 겪은 문제와 해결

팀 이름 불일치로 거부된 룰셋 교체



모든 룰셋의 팀 이름을 blue와 red로 통일

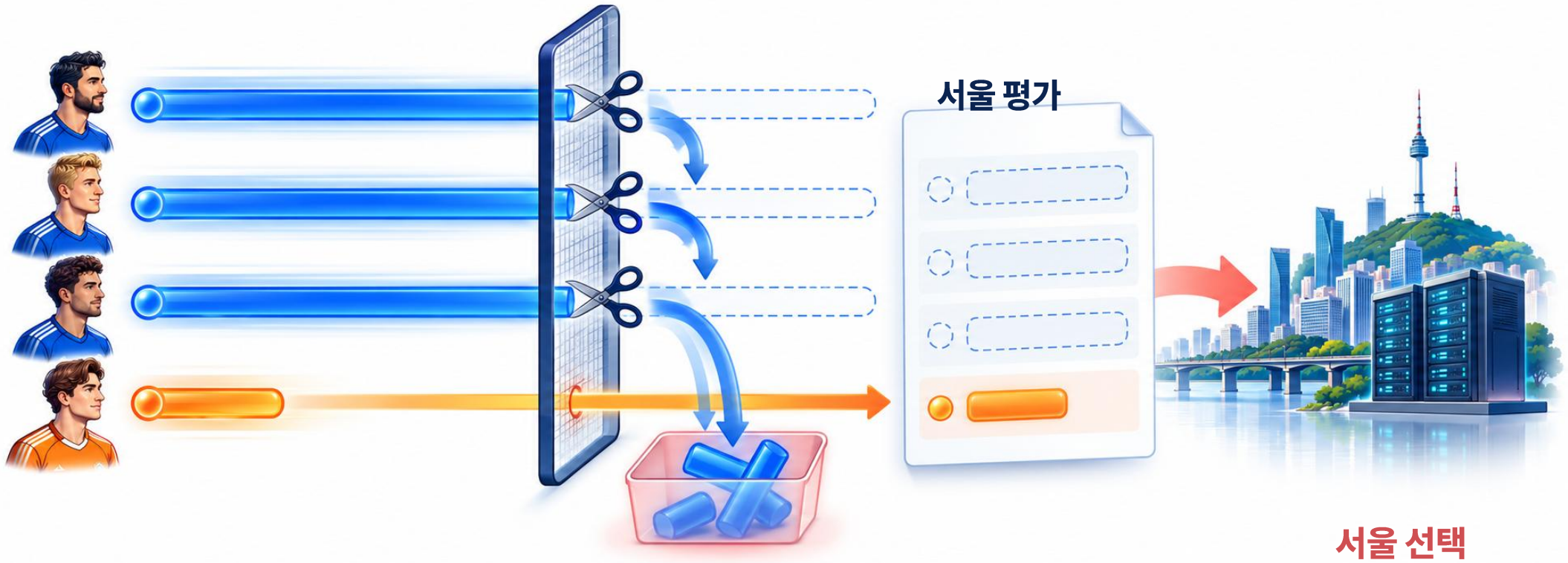
유럽 유저가 많은 매치의 서울 배치



유저를 묶는 매칭과 경기 서버를 고르는 배치는 별도 판단

높은 핑 누락으로 왜곡된 서버 배치

파랑: 유럽 유저 주황: 아시아 유저 | 막대 길이: 해당 지역까지의 핑

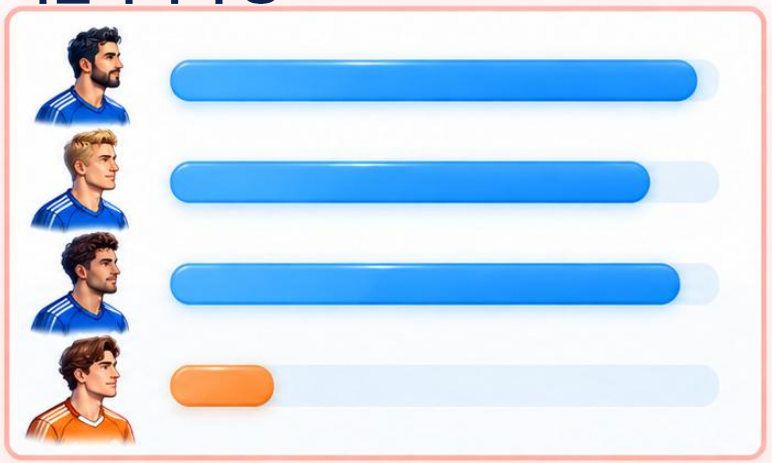


서울 핑을 보고한 유저만 평균에 반영

모든 지역의 핑을 반영한 서버 배치

파랑: 유럽 유저 주황: 아시아 유저 | 막대 길이: 해당 지역까지의 핑

서울까지의 핑





높은 핑도 평가에 포함

유럽 지역까지의 핑



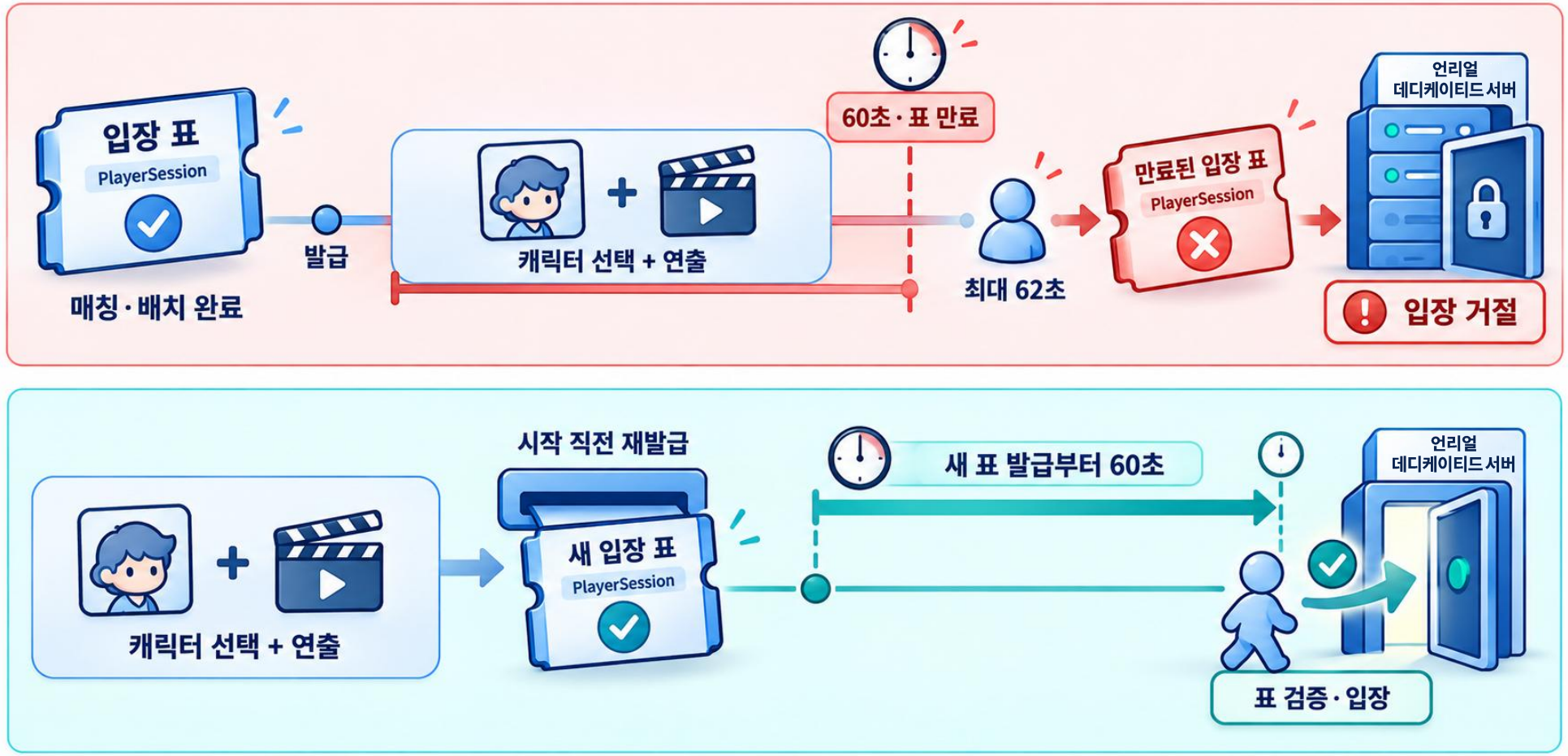


수정 후 유럽 지역 배치

열린 지역의 측정 핑 전체 전달 · 누락 없는 비교

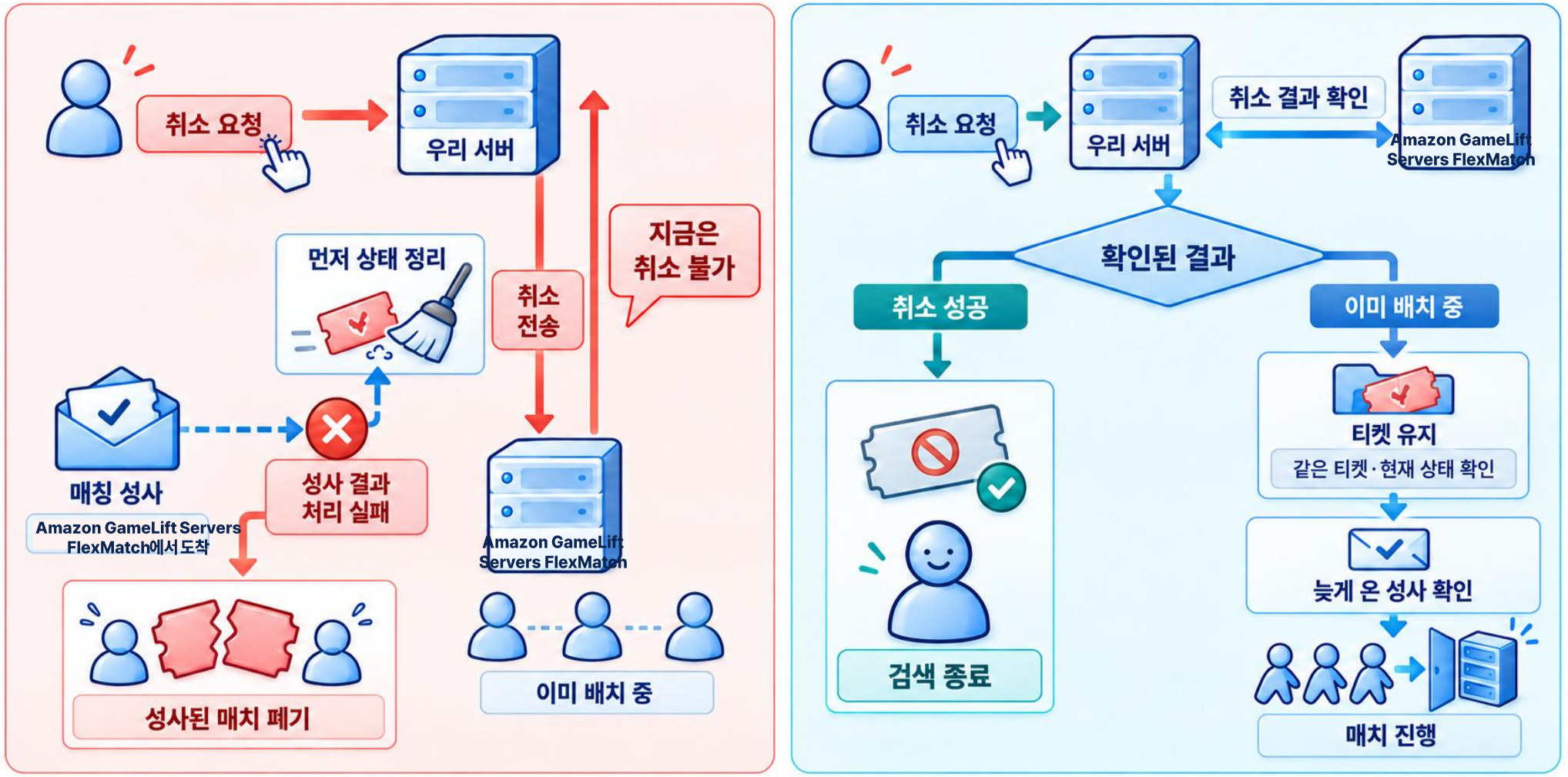
유테

60초 입장 제한을 넘긴 준비 시간



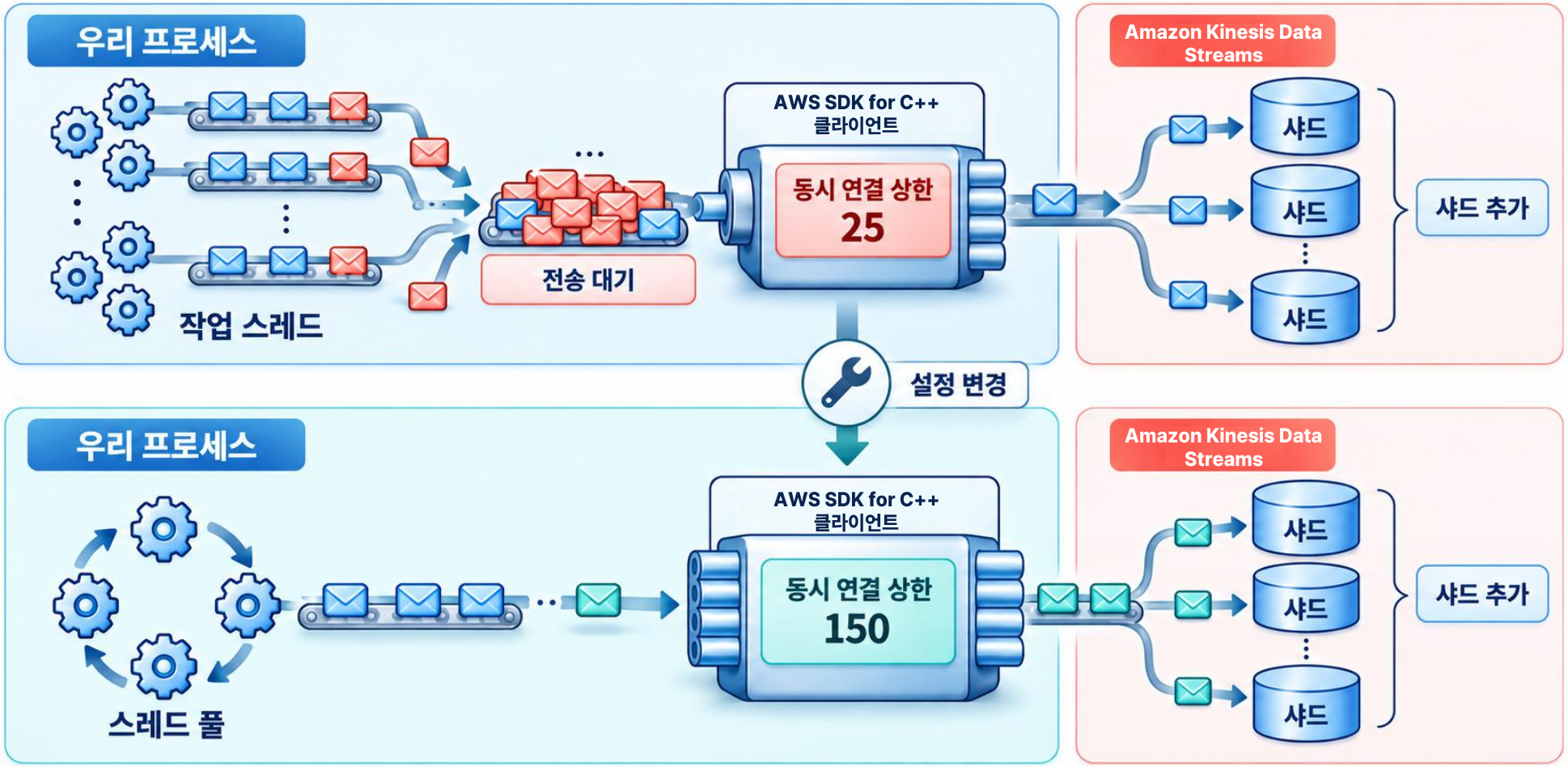
경기 시작 직전 재발급으로 입장 시간 확보

취소 요청과 취소 확정의 구분



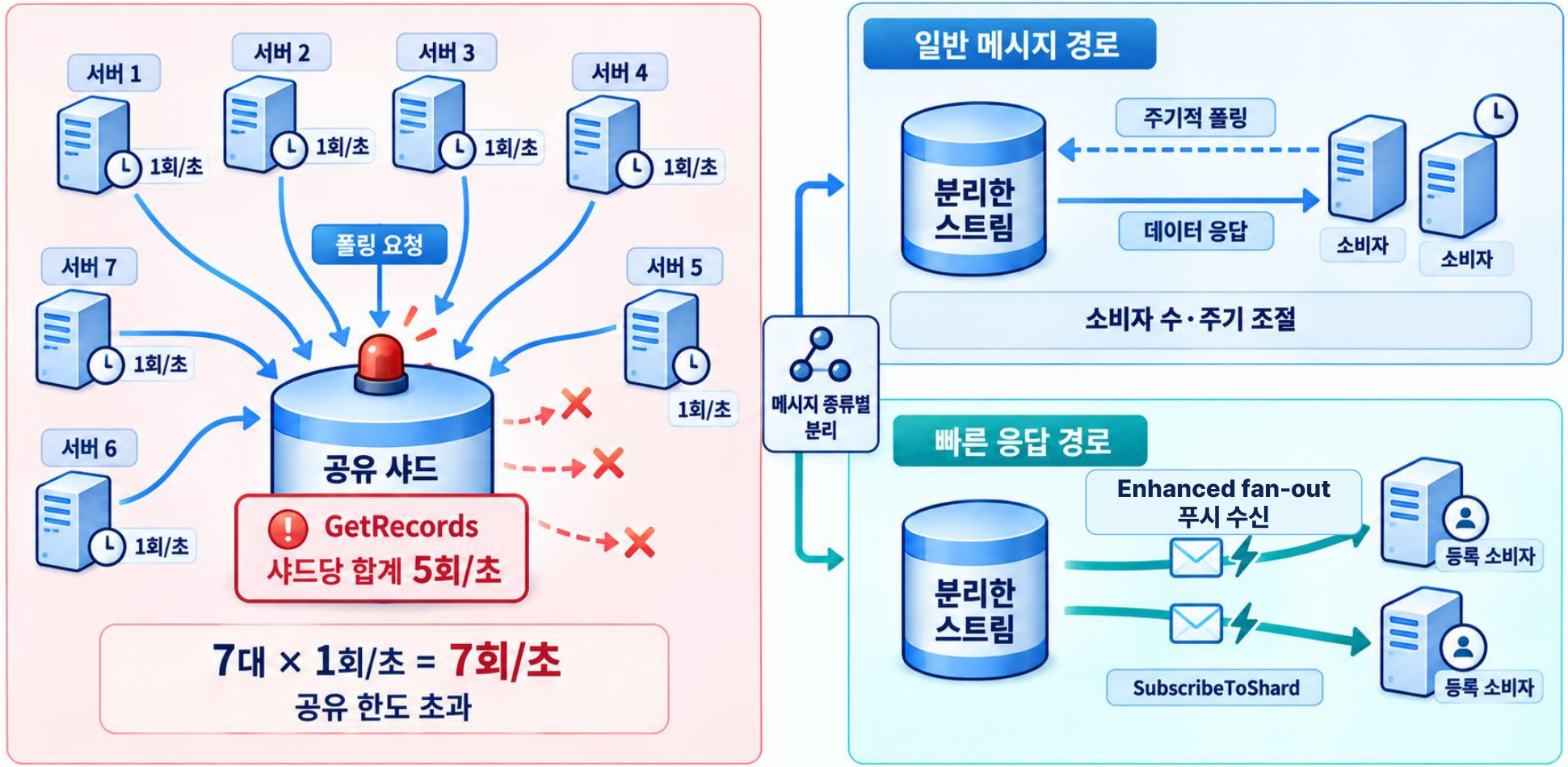
취소되면 검색 종료 · 이미 성사됐으면 매치 진행

샤드를 늘려도 남은 SDK 연결 병목



동시 연결 상한 25 → 150 · 스레드 풀 150개 함께 적용

여러 서버가 나눠 쓰는 샤드 폴링 한도



메시지 종류별 스트림 분리 · 필요한 경로에 Enhanced fan-out 적용



적은 인원으로 구현한 글로벌 게임 서비스

게임리프트와 AWS 서비스는 우리가 상상한 구조를 실제 서비스로 만들어 주었습니다. 서버 실행과 복구, 경기 서버 배치를 맡길 수 있었기에 적은 인원으로도 글로벌 서비스를 구현할 수 있었습니다.

우리 서비스에 맞는 방식을 찾고, 검증하고, 더 효율적으로 만드는 일은 우리의 몫이었습니다.

FREESTYLE FOOTBALL2

유레



감사합니다.

프리스타일 풋볼 2

STEAM XBOX SERIES X|S PS5

절찬 예약 판매 중

유테