MO Game 'Battle Snake' 서버 전체 구조

BackEnd

Matchmaking DB (Replication)

- Matchmaking_status (master)
  - serverstatusTBL
- Replication
- Matchmaking_status x n (slave)
  - serverstatusTBL

Content DB (Sharding)

- shDB_data_1
  - accountTBL
  - contentsTBL
- shDB_data_2
  - accountTBL
  - contentsTBL
- • • •

Sharding Info DB (Replication)

- shDB_Index (master)
  - AllocateTBL
- Replication
- shDB_Index x n (slave)
  - AllocateTBL
- shDB_Info (master)
  - AvailableTBL
  - dbconnectTBL
- Replication
- shDB_Info x n (slave)
  - AvailableTBL
  - dbconnectTBL

PHP API

- 계정 생성 API
- 유저 정보 갱신 / 얻기 API
- 컨텐츠 정보 갱신 / 얻기 API
- 매치메이킹 서버 정보 얻기 API

특수한 경우를 제외한 DB 접속은 모두 PHP로 제작한 API를 거친다.
- Front와 BackEnd를 명확히 구분
- Front에서 BackEnd로 접근하기 위한 Gate 역할.
- Front에서는, BackEnd의 DB 구조를 전혀 몰라도, API만 호출하면 원하는 결과를 얻을 수 있다.

매치메이킹 서버 정보 갱신

Front

Server

Apache (웹 서버)

Apache를 향한 요청은 모두 POST.
- **Body :** JSON

HTTP (POST) JSON

Matchmaking Server x n
- 매치메이킹 Net 서버
- 마스터 Lan 클라이언트

Master Server
- 배틀 Lan 서버
- 매치메이킹 Lan 서버

2

Battle 서버 군 x n

Battle Server
- 배틀 MMO 서버
- 채팅 Lan 서버
- 마스터 Lan 클라이언트

Chat Server
- 채팅 Net 서버
- 배틀 Lan 클라이언트

3

HTTP (POST) JSON

HTTP (POST) JSON

Lobby Server

1

클라이언트 접속 절차

1. Lobby 서버에서, 접속할 매치메이킹 서버를 알아온다.
2. 매치메이킹 서버에서, 접속할 배틀서버,채팅서버와 입장할 방을 알아온다.
3. 배틀서버, 채팅서버 입장 후, 방 입장

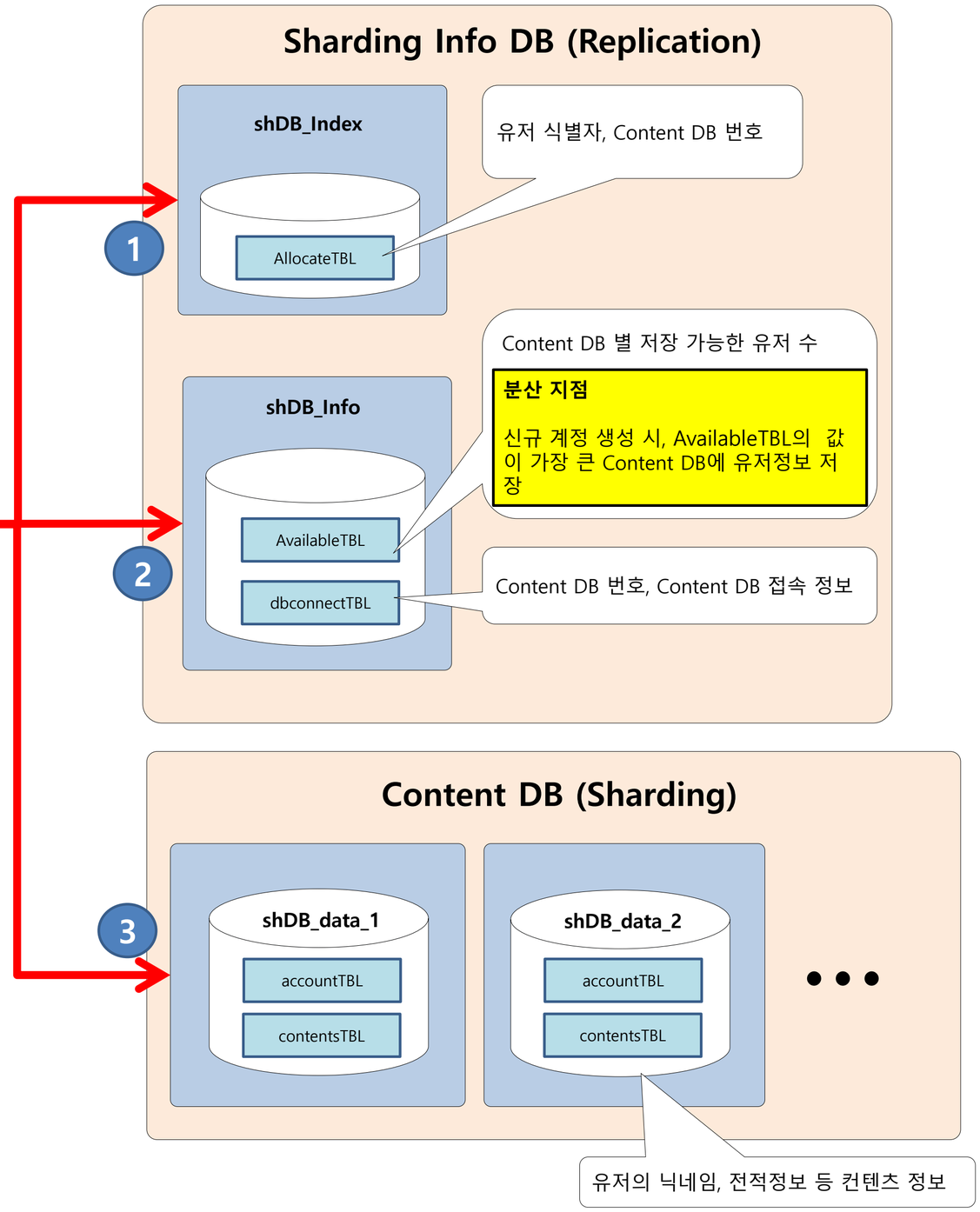
클라이언트 Connect

DB Sharding 구조

Content DB(Sharding)에 접근하기 위해서는, Sharding Info DB를 거쳐야 한다.

1. shDB_Index에서 ,해당 유저가 저장된 Content DB의 번호를 알아온다.
2. 알아온 Content DB의 번호를 이용해, shDB_Info에서 해당 DB로 접속하기 위한 정보를 알아온다.
3. 알아온 정보를 이용해 Content DB에 접속.

샤딩 DB 접근 요청



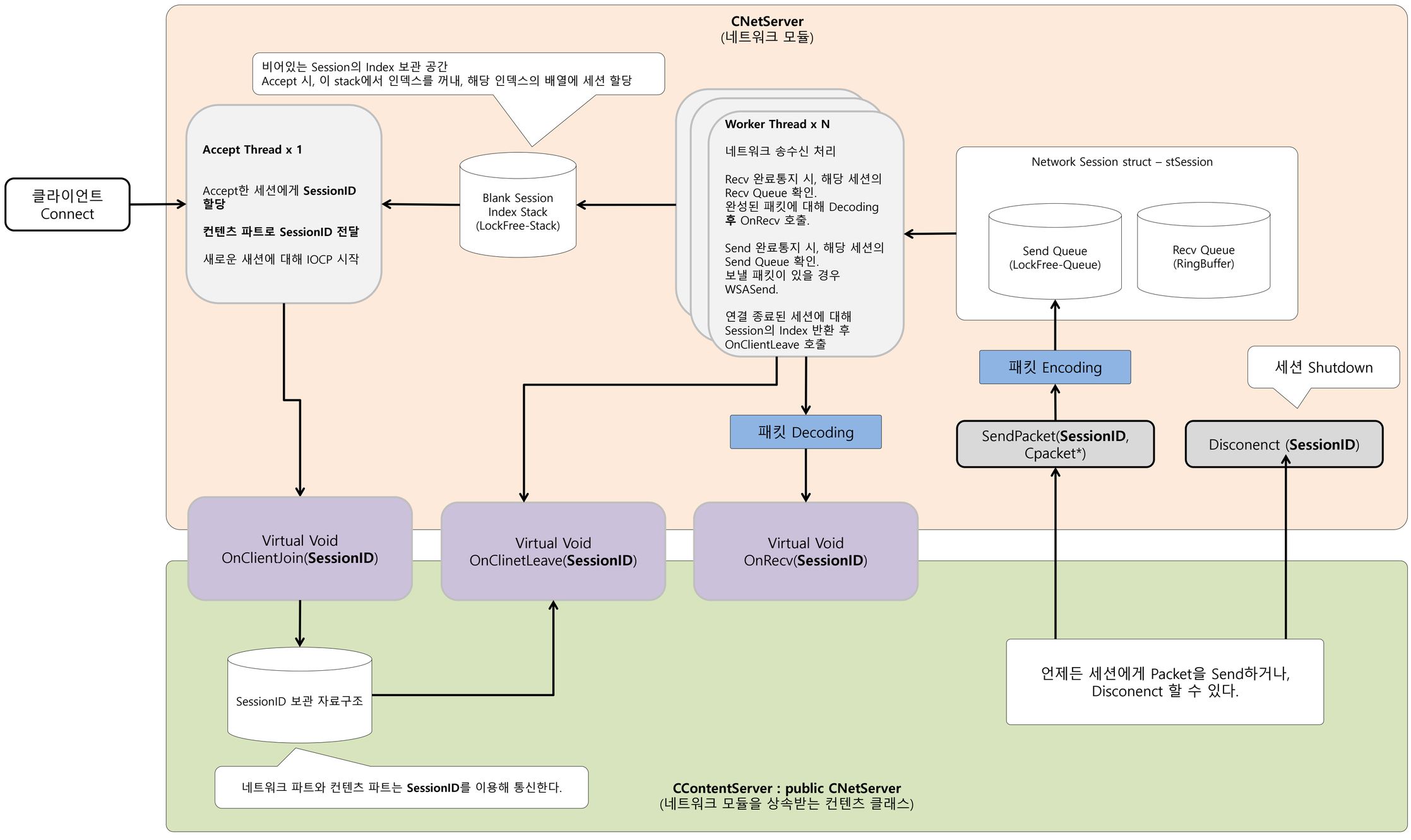
네트워크 모듈 – CNetServer / CLanServer

특징

- LockFree 구조의 네트워크 모듈
- SessionID를 이용한 네트워크 <-> 컨텐츠간 통신

CNetServer, CLanServer의 차이점

- CLanServer는 패킷 Encoding / Decoding 절차가 없다.
- 그 외 완전히 동일



네트워크 모듈 - MMOServer

특징

- LockFree구조의 네트워크 모듈
- 네트워크의 Session과 컨텐츠의 Player가 1개로 구성 (상속 구조). → 네트워크와 컨텐츠 간의 통신 프로토콜 불필요
- 세션의 현재 상태에 따라 세션의 모드 결정. (Auth 모드 세션, Game 모드 세션, Release 모드 세션)
- 세션의 모드에 따라 로직 처리 담당 스레드가 다름.
- 해당 모드의 세션에는 스레드 자신만 접근. (ex. Auth 모드의 세션에는 Auth 스레드만 접근) → 세션간 동기화 불필요.