



---

# 온라인 게임 서버 엔진의 설계

- 재사용성의 극대화를 위하여

---

2004. 3.

(주)락소프트 대표 조홍섭

## [1] 온라인 게임 서버 엔진 개발 개요

- ▶ 서버 엔진 개발의 필요성
- ▶ 서버 엔진 개발의 난관
- ▶ 서버 엔진 개발의 현실적 목표
- ▶ 엔진 설계의 지향 점

## [2] MMORPG 서버 구조 분석 및 평가

- ▶ Zone(지역 분할) 분산 서버
- ▶ Network-Processing 분산 서버
- ▶ Network-Processing-Zone 분산 서버
- ▶ Network-Processing-AI 분산 서버

## [3] 온라인 게임 서버 엔진 설계의 실제

- ▶ 엔진 계층 구조의 설계
- ▶ 각 계층별 구성 요소 및 핵심 기술
- ▶ 서버 엔진 개발 사례

### 서버 엔진 개발의 필요성

- ▶ 명시적 역할 분담
- ▶ 코드의 재활용성
- ▶ 콘텐츠 개발의 질적 향상 기반

 개발 공정의 합리화, 체계화

### 서버 엔진 개발의 난관

- ▶ 단일 타이틀에 매진하는 업계의 현실
- ▶ 경험 있는 개발자의 부족
- ▶ 기술 교류를 꺼리는 풍토

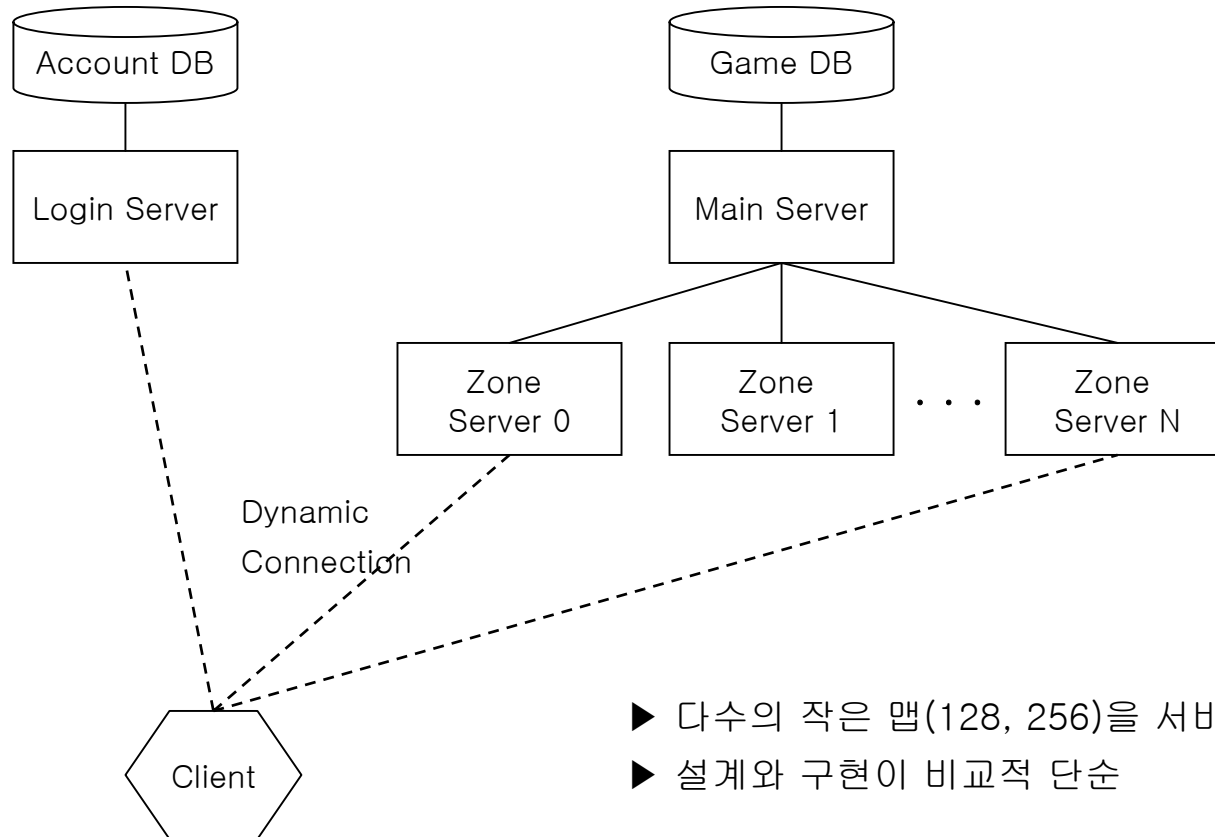
### 서버 엔진 개발의 현실적 목표

- ▶ Network, DB, 기반 Library 구축
- ▶ 콘텐츠에 독립적인 온라인 게임 서버 Framework 구축
- ▶ 동시접속자 5000명 이상 서비스 가능한 MMORPG 분산서버 구축

### 엔진 설계의 지향점

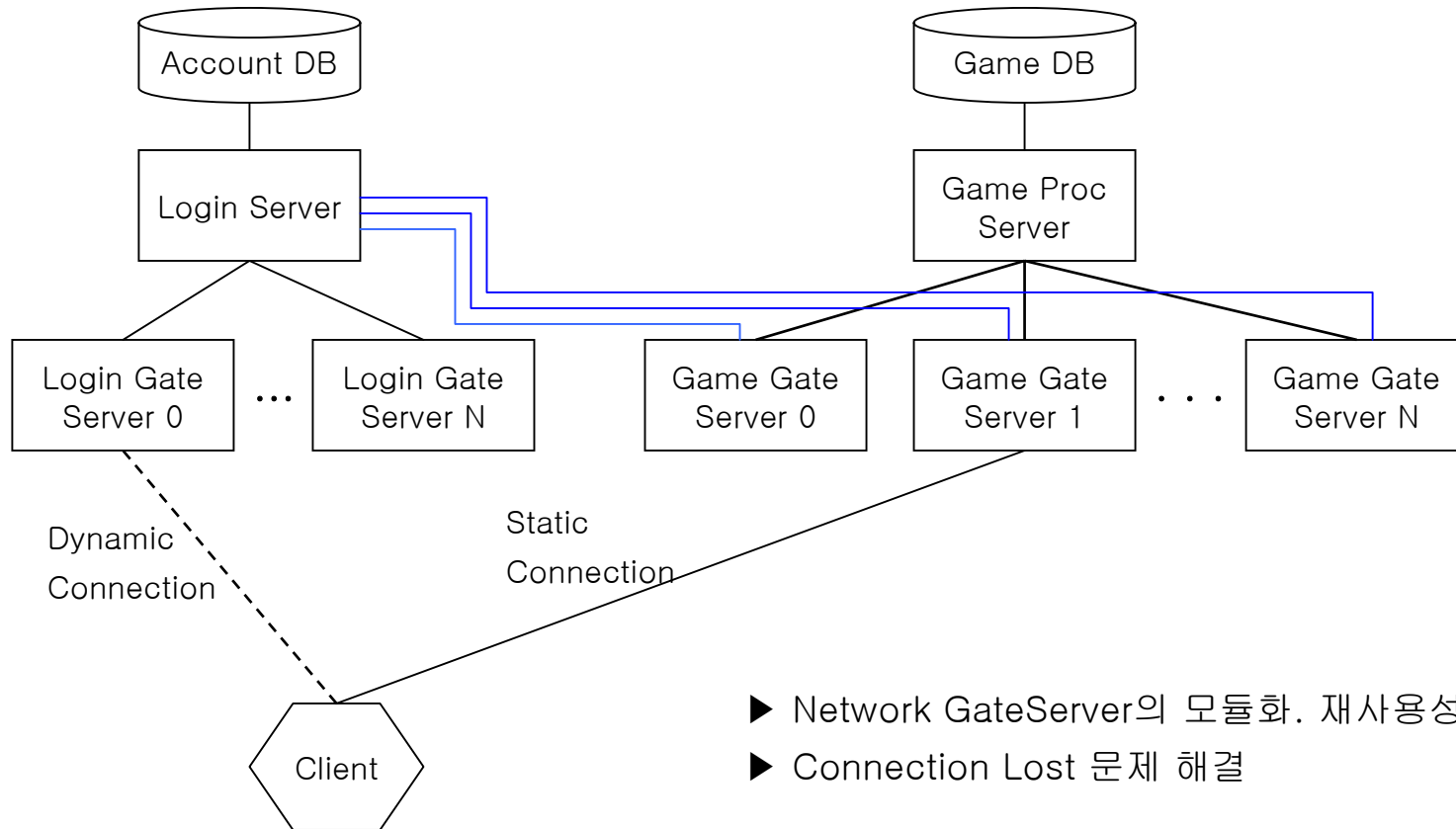
- ▶ 모듈화
- ▶ 분업화
- ▶ 재사용성 극대화

### Zone(지역) 분산 서버



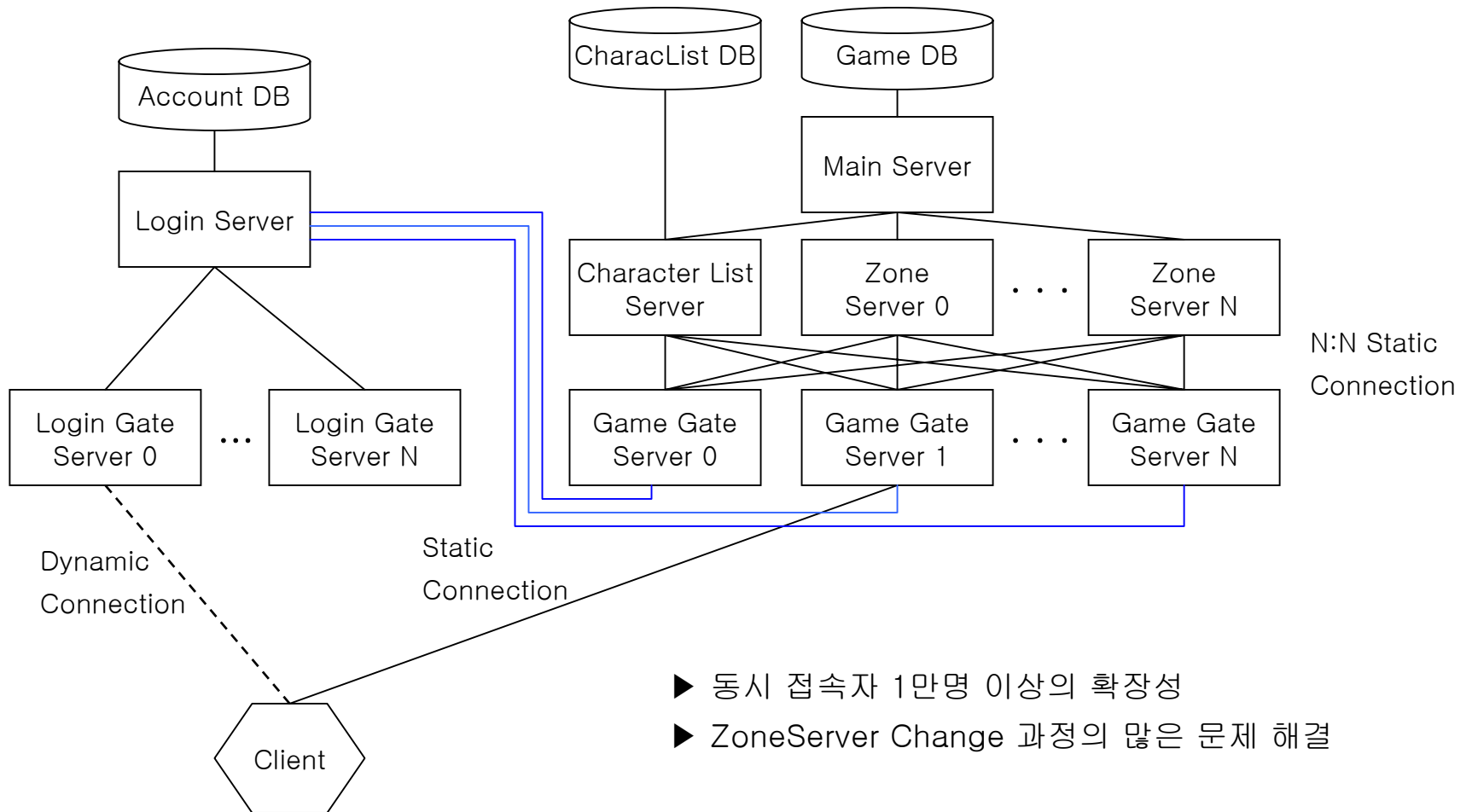
- ▶ 다수의 작은 맵(128, 256)을 서비스하기에 효율적
- ▶ 설계와 구현이 비교적 단순
- ▶ 콘텐츠와 엔진부의 모듈별 분리가 어려움
- ▶ 복잡해지는 코드 : 유지보수의 문제
- ▶ ZoneServer Change 과정에서 여러 문제 가능성

Network-Processing 분산 서버



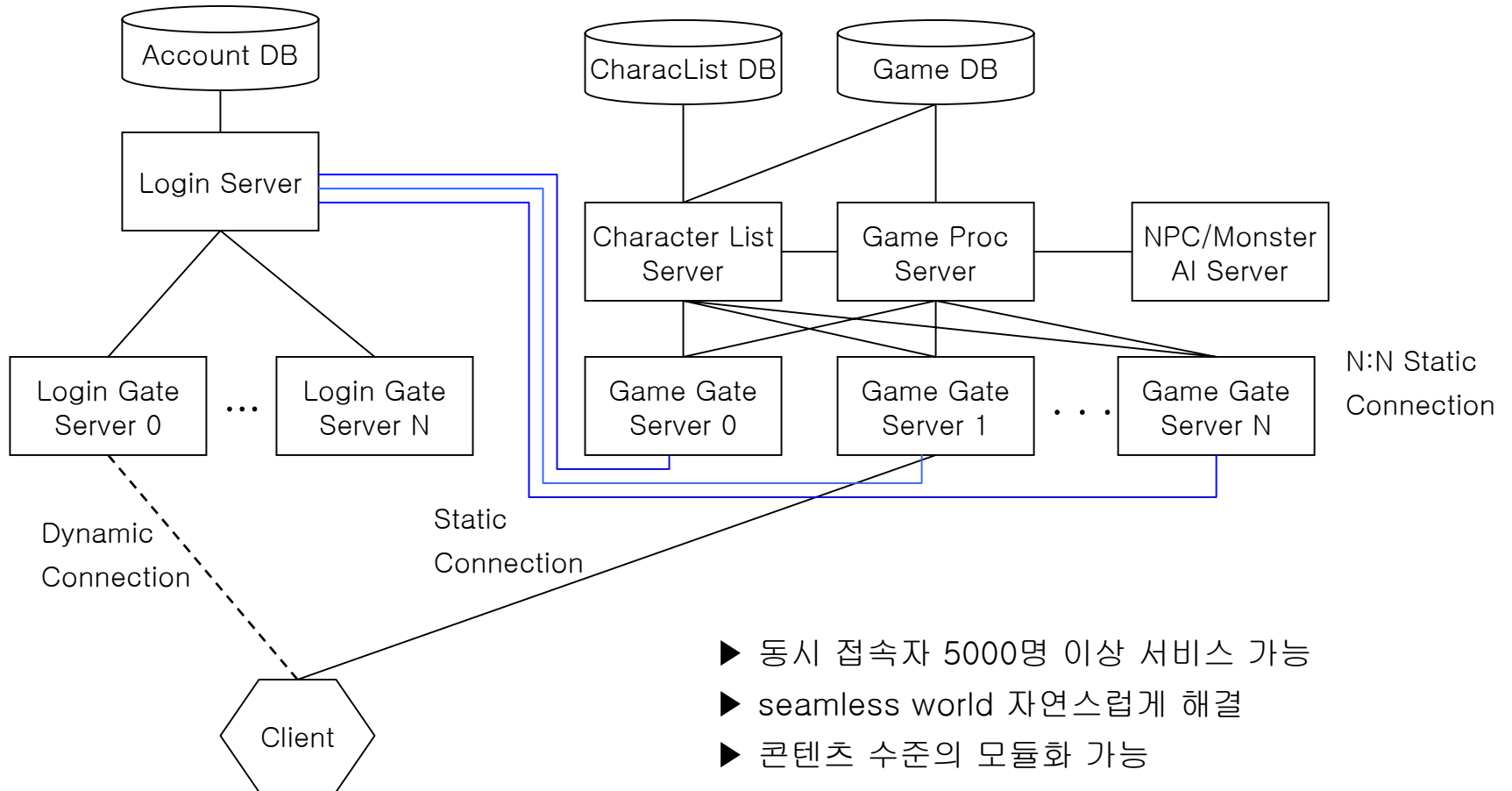
- ▶ Network GateServer의 모듈화. 재사용성 향상
- ▶ Connection Lost 문제 해결
- ▶ 단일한 GameProcess 서버에 Load가 집중
- ▶ 동시접속자 한계 : 2000 명

Network-Processing-Zone 분산 서버



- ▶ 동시 접속자 1만명 이상의 확장성
- ▶ ZoneServer Change 과정의 많은 문제 해결
- ▶ 복잡도 증가. 다수 프로세스 : 유지보수의 문제
- ▶ seamless world 문제

Network-Processing-AI 분산 서버

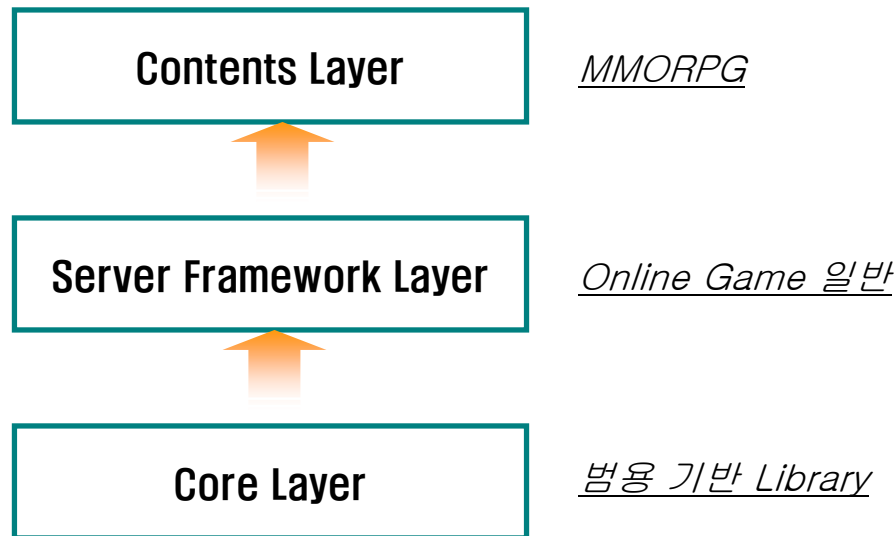


- ▶ 동시 접속자 5000명 이상 서비스 가능
- ▶ seamless world 자연스럽게 해결
- ▶ 콘텐츠 수준의 모듈화 가능
  
- ▶ 단일한 GameProcess 서버에 집중되는 Load  
- Hardware의 성능 향상에 의존



## 엔진 계층 구조의 설계

- ▶ Core Layer : Network Library, DB Library, Crypto Library
- ▶ Server Framework Layer : Login Server, GateServer, DB Middleware
- ▶ Contents Layer : Character List Server, Game Proc Server, Npc/Monster AI server



## Core Layer의 구성 요소 및 핵심 기술

### ▶ Network

Listen Socket, Connect Socket, None-Blocking Socket, Overlapped Socket, IOCP  
Network Buffer Manager

### ▶ Multi-Thread Relations

### ▶ DB

MDAC base OleDb Consumer, Command-Session-Connection Manager  
Table Bind, Command Prepare, Command Execute, BLOB Data Manage

### ▶ Crypto

RSA, DES, AES

AES : Advanced Encryption Standard – Rijndael Algorithm.

<http://csrc.nist.gov/CryptoToolkit/aes/rijndael>

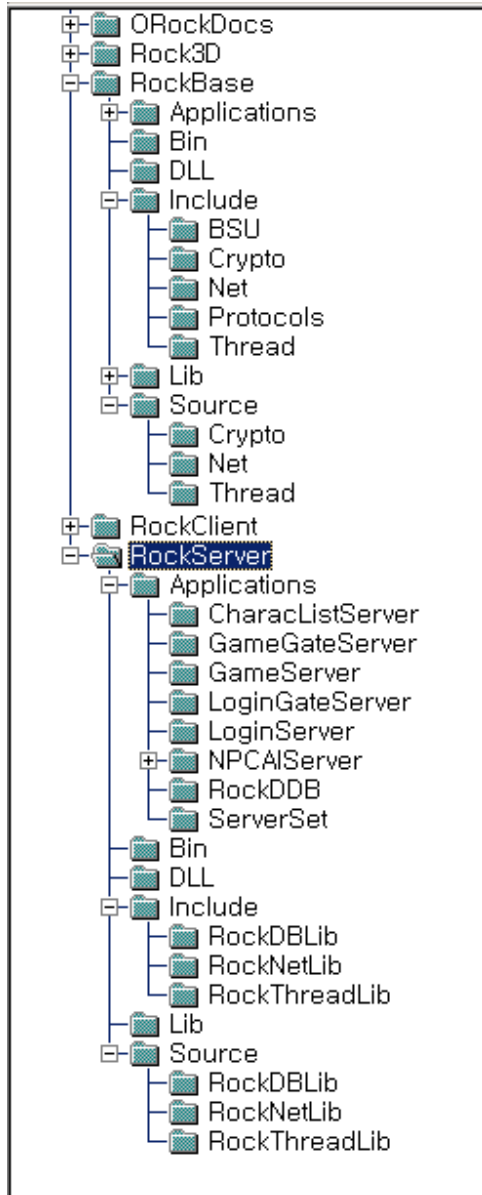
### Server Framework Layer의 구성 요소 및 핵심 기술

- ▶ Updater : FTP Server, MFC CInternetSession, CFtpConnection, CInternetFile
- ▶ Login Server : Account 인증, 과금, 중복 Login 검출, Server Down감지, GateServer 로드 분산
- ▶ GateServer : IOCP, 암호화, Network Defense(Packet Flood, Connection Flood, Speed Hack...)
- ▶ DB Middleware : 분산 DB 관리. Command Input & Output. Client와 TCP로 통신

### Contents Layer의 구성 요소 및 핵심 기술

- ▶ 시야처리 Logic / Quest Script / Path Finder

## 서버 엔진 개발 사례



### ▶ RockBase DLL

Server-Client 공통 요소.

Thread, Network( IOCP제외 ), Crypto...

### ▶ RockServer

DLL : Thread Library, Network Library, DB Library

Apps : LoginGateServer, GameGateServer, LoginServer

RockDDB

Apps : CharacListServer, GameServer, NPCAIServer

Contents 관련 Application은 별도의 Project로 정정 필요

감·사·합·니·다

