Agent

Joonho Park
2025-09-01

Introduction

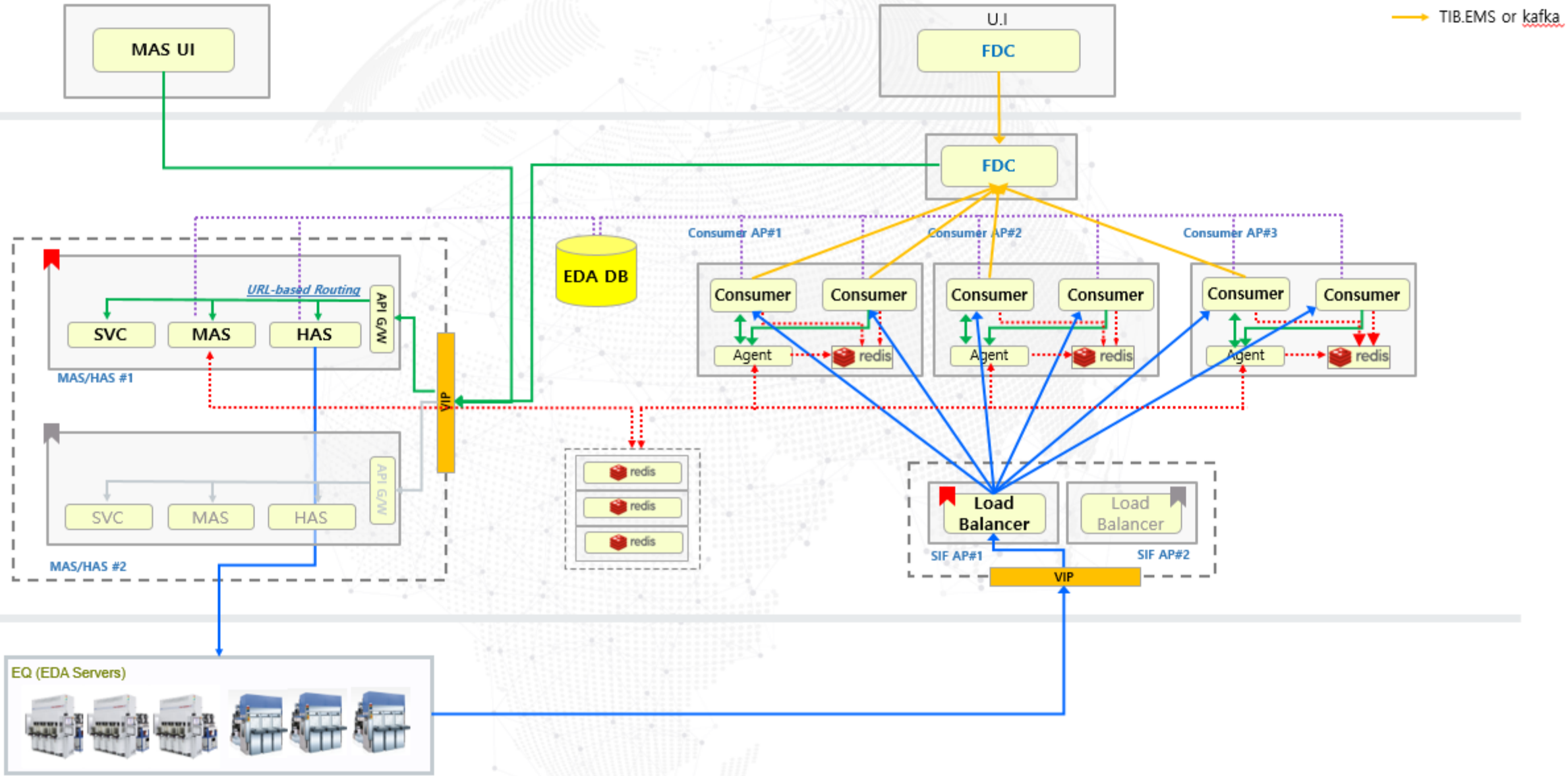
- 각 Application 이 있는 서버에서 각 Application의 상태, Resource 를 수집하고 Stop, Start, Restart 와 같은 Control을 수행한다.
- 각 서버에서 Application 상태, Resource 수집 및 제어
- AgentMode에 따른 역할:

  APP_SERVER → Subscribe/ App Control / Application 상태·리소스 체크 후 Redis 저장

  BK_CONSUMER → Subscribe/ App Control /Application 상태·리소스 체크 후 Redis 저장

  PROXY_SERVER → Log Pattern Match 후 Redis 저장

아키텍처 - Load balancer 기반 시스템 구성



Problem Setup

- - 각 서버별 기능 분리 필요
- - 각 서버의 Application 에 따른 Control 기능 필요
- - Resource 및 상태 수집 필요
- - Log 관리 및 전송 필요
- - Redis와의 통신 필요

Scope

- APP_SERVER
- BK_CONSUMER
- PROXY_SERVER
- 내부/외부 Redis 통신

Problem solution -API GateWay

- Agent Mode = APP_SERVER
- Redis 구독 최초 한번 하도록 실행. AppServer Control / Deploy / Patch
- Application 상태 1초 마다 체크 후 Redis Hash Set
- Resource 사용량 1초 마다 체크 후 Redis ListRightPush

Problem solution - BK_CONSUMER

- Agent Mode = BK_CONSUMER
- Redis 구독 최초 한번 하도록 실행. Consumer Control / Deploy / Patch
- Application 상태 1초 마다 체크 후 Redis Hash Set
- Resource 사용량 1초 마다 체크 후 Redis ListRightPush

Problem solution - PROXY_SERVER

- Agent Mode = PROXY_SERVER
- Redis 구독 최초 한번 하도록 실행. Proxy Log Send

Problem solution - 내부/외부 Redis 통신

- Agent Mode = APP_SERVER / BK_CONSUMER
- Redis 구독 최초 한번 하도록 실행.
- PROXY_SERVER 를 제외한 AgentMode 는 내부Redis 도 연결

Reference

- Serilog – Apache-2.0
- SSH.NET – MIT License
- StackExchange.Redis – MIT License
- Swashbuckle.AspNetCore – MIT
- PostgreSQL –PostgreSQL
- DotNurse.Injector – MIT
- Microsoft.Extensions – MIT
- Oracle.ManagedDataAccess.Core - Oracle

Result Notes

- 안정적인 Redis 통신 가능
- Data 흐름 파악하기 위한 Log 집약.
- 각 Application 제어 용이.
- 각 서버 상태,Resource 집약.