

LG CNS DevOn

Java Enterprise Framework

INTRODUCTION

DevOn 주요특징

LG CNS의 DevOn Enterprise Framework는 다수의 금융권 차세대 Framework 이행경험을 통해 축적된 역량을 반영한 검증되고 준비된 Framework입니다.

— LG CNS DevOn Enterprise Framework —

검증된 아키텍처	준비된 기능	품질 및 기술력
▪ 은행 및 카드 차세대 최초/최다 적용 ▪ 최근 5년간 금융권 차세대 적용사례 다수 (은행, 카드, 보험, 캐피탈 등) ▪ 검증된 고성능의 온라인 및 대용량 처리 - PoC/BMT, 운영환경 처리기술력 검증 ▪ 유지보수 서비스 다수 수행	▪ 다양한 프로젝트 적용을 통한 고객의 요구사항 및 Know-how 반영 ▪ 개발/실행/운영환경을 구비한 통합 프레임워크 제품 ▪ 고성능 대량 처리를 지원하는 다중노드, 병렬처리 배치, Center Cut, 후행 아키텍처 ▪ 차세대 적용을 통해 엄선된 금융 특화 기능과 시스템 공통기능 탑재	▪ 핵심 엔진 전반을 자체 기술력으로 개발 ▪ 외부기관 - GS 인증 (2009) ▪ 다수의 대규모 금융 차세대 수행을 통해 축적된 Framework 이행 역량 ▪ 금융 기술지원(유지보수)전담 조직 보유로 안정적 서비스 지원

시스템 인프라에 독립적인 구조 제공

"Unix, Mainframe부터 Linux까지 적용 사례를 보유한 플랫폼 독립적인 Framework

DevOn Java
적용 플랫폼

HW (OS)

HP-UX, IBM-AIX,
IBM zOS, Sun-Solaris ..

Language

Java, UML

RDBMS

Oracle , DB2
Sybase ..

Middleware

WebLogic, Jeus,
WebSphere, Jboss ..



제품의 구성

DevOn Enterprise Framework 은 JavaEE를 지원하는 코어 아키텍처, OLTP 처리가 가능한 온라인 아키텍처, 배치 아키텍처, 센터컷 아키텍처, 후행처리 아키텍처를 가지고 있으며 Ecilpse 기반의 통합개발도구 및 웹 기반의 운영관리 도구와 함께 제공됩니다.



설명



명칭 및 버전

- 명칭 : DevOn Enterprise Framework
- 버전 : 5.1

1 실행 엔진

- 온라인 및 공통 기능 제공
- 금융 특화 기능 내재화(센터컷, 배치)

2 개발 지원 도구

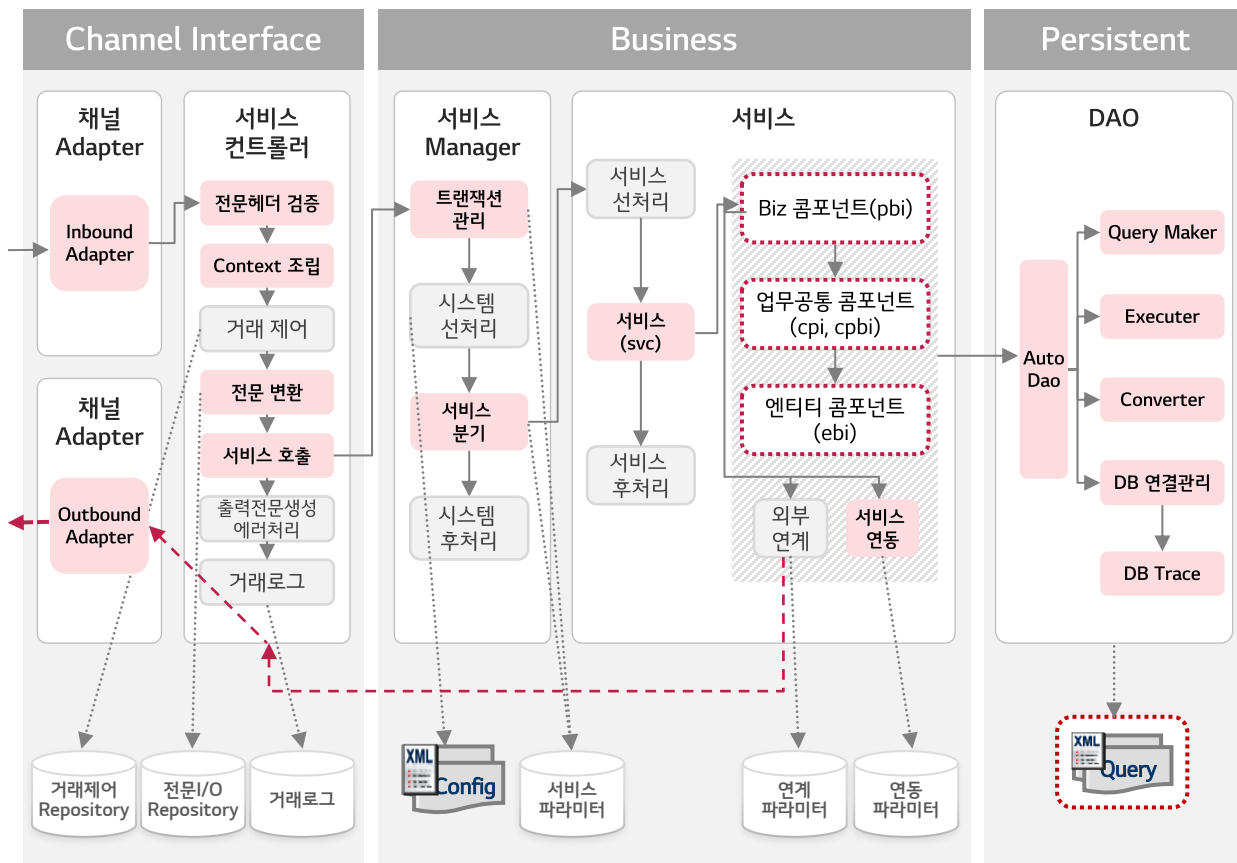
- Query 자동 생성 지원
- 서비스 기반 테스트 도구 지원
- 전문 설계 도구

3 운영 관리

- 서비스 관리
- 모니터링
- 온라인, 센터컷, 배치 통합 관리

온라인 실행 아키텍처

DevOn Framework의 온라인 아키텍처는 Java EE, 산업 표준 아키텍처 및 패턴을 준수하여 유연하고 계층화된 구조를 갖추고 있습니다.



설명

Channel Interface Layer

- 다양한 전문구조에 대한 전문 변환 처리 기능제공
- 전문구조와 독립적으로 Business Layer에서 사용하는 표준데이터 객체 생성
- 전문 Parsing/검증, 거래제어, 서비스 호출 등의 기능을 수행

Business Layer

- Business Layer에서는 업무로직만 처리, Service의 흐름과 제어를 FW이 통제/관리, 시스템 안정성, Tx, low level FW 수행, api 형태 제공

Persistent Layer

- DB Access처리를 위한 공통 컴포넌트 제공

공통

- 서비스를 처리하기위해 런타임 과정에서 참조되는 속성,설정정보,제어정보를 하드코딩 하지 않고 파라미터화하여 관리함, 시스템 안정성 기능 제공

프레임워크 파라미터 관리 (1/2)

운영 관리 기능(DevOn Management)을 통해 서비스 처리 과정에서 참조되는 정보들을 데이터베이스 테이블화 하여 파라미터로 관리합니다. Framework에서는 파라미터 정보를 통해 거래(서비스)를 처리합니다.

종류	설명	비고
설정 파라미터 (Config)	- 암복호화, 인코딩, File upload/download, datasource, paging 등 설정 - 기본값을 사용하지 않을 경우, 프로젝트에서 작성하는 각 설정파일(devon-core.xml, devon-framework.xml)에 해당값을 재정의하여 사용가능	
서비스 파라미터	- 서비스 단위의 기본정보, 거래정보, 로깅정보 등의 속성 관리 : 서비스가 사용하는 클래스 정보, 트랜잭션 유형, 로깅레벨 등 - FW에서 사용하는 필수 속성	
서비스확장 파라미터	- 서비스 단위의 거래처리용 정보, 제어정보 등의 속성 - 고객사 별로 정의해서 사용할 수 있도록 확장된 파라미터, 모든 거래의 공통파라미터 등록	
거래개별 파라미터	서비스(개별 거래단위) 단위의 제어 속성 정의	선택적 사용 가능
연동 파라미터	서비스 연동을 위한 연동 기본 정보 및 연동 처리 속성 관리	선택적 사용 가능
연계 파라미터	대외, EAI 등 외부 시스템과의 연계 시 필요한 속성 관리	선택적 사용 가능

설명



주요기능

- 서비스 처리 과정에서 참조되는 다양한 설정정보 및 서비스 통제를 위한 제어 정보 등을 파라미터화 하여 데이터베이스 테이블에 관리
- 서비스의 중요 속성 및 설정 정보통합 관리
- 서비스의 정보를 동적으로 변경할 수 있음
- 성능향상을 위해 기능 및 업무별로 공통된 속성의 전체 데이터를 메모리에 Caching
- 운영관리도구를 통하여 변경된 파라미터의 실시간 반영

프레임워크 파라미터 관리 (1/2)

서비스 파라미터

■ 서비스의 중요 속성 및 설정 정보 통합 관리

Home | 사이트 [prototype] | 언어 [ko] | 김주연(jooyekim) 님 환영합니다. | Logout

메뉴 화면 거래 메시지&코드 전문 연계 로그 온라인타임 엔터프라이즈 디퍼드&센터컷 배치 엑셀일괄처리 설정

< 거래 목록 >

> 카테고리 > 거래 목록

시작날짜 ~ 종료날짜 식별자 Q검색

○ 거래 목록

NO	S	구분	Spec	거래식별자	거래명칭	[요청클래스].[메서드]	수정자	수정일시	MD
1	+	IO		POPA0801	청약입력내역조회	[lgcns.insur.po.pbc.sucoMag.sucoInptPatInqrPbc].[POPA0801]	jooyekim	15/08/19	
2	+	IO		FWPOCTEST00	POC 테스트 거래	[devon.scp.proto.batch.ondemand.pbc.OnDemandPbc].[retrieveEmpPocInfo]	jyoon	15/08/10	
3	+	WEB	ajax	BHFWDONDEMO1	배치 온라인	[devon.scp.proto.batch.ondemand.pbc.OnDemandPbc].[callOnDemandBatch]	jyoon	15/05/10	
4	+	WEB	ajax	CALLONDEMANDBATCH	온디맨드배치CALL	[devon.scp.proto.batch.ondemand.pbc.OnDemandPbc].[retrieveCnt]	seonnam	15/04/20	
5	+	WEB	ajax	RETRIEVECNT	사원정보출건수	[devon.scp.proto.batch.ondemand.pbc.OnDemandPbc].[retrieveEmpInfoList]	jooyekim	15/06/30	
6	+	WEB	ajax	RETRIEVEEMPINFOLIST	사원정보목록조회	[devon.scp.proto.bank.withdraw.pbc.WithdrawPbc].[depositAccount]	jyoon	15/05/04	
7	+	WEB	form	DEPOSITACCT	입금거래	[devon.scp.proto.bank.withdraw.pbc.WithdrawPbc].[withdrawAccount]	jyoon	15/05/04	
8	+	WEB	form	WITHDRAWACCT	출금거래				

○ 서비스 반영 ○ 양식받기 ○ 엑셀업 ○ 엑셀다운 +추가 x삭제

기본정보 | 로그정보 | 처리정보 | 업정보 | 확장정보 | 요청검증 | 저장

요청클래스: lgcns.insur.po.pbc.sucoMag.sucoInptPatInqrPbc.SucoInptPatInqrPbc

요청 메서드: POPA0801

거래 종류 구분: Select Type

거래트랜잭션유형: No Transaction | 트랜잭션 타임아웃: -1 [sec]

거래 선후처리 클래스:

정보 수정

카테고리: 기능검증

거래유형: WEB ☐ 전문 ☒

거래식별자: EQPA0801

거래명칭: 청약입력내역조회

요청클래스: lgcns.insur.po.pbc.sucoMag.sucoInptPatInqrPbc

요청 메서드: POPA0801

요청전문: POPA0801_I

응답전문: POPA0801_O

소유자: - 2015.08.11 20:06:43 - jyoon

수정자: - 2015.08.19 17:31:55 - jooyekim

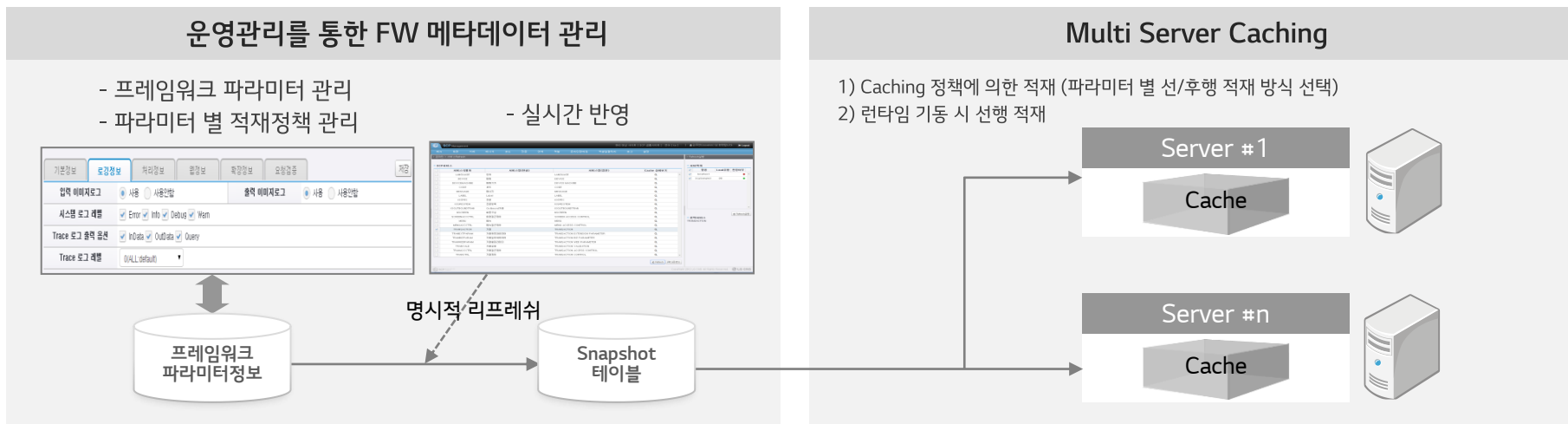
생성자: - 2015.08.11 20:06:43 - jyoon

MODIFY

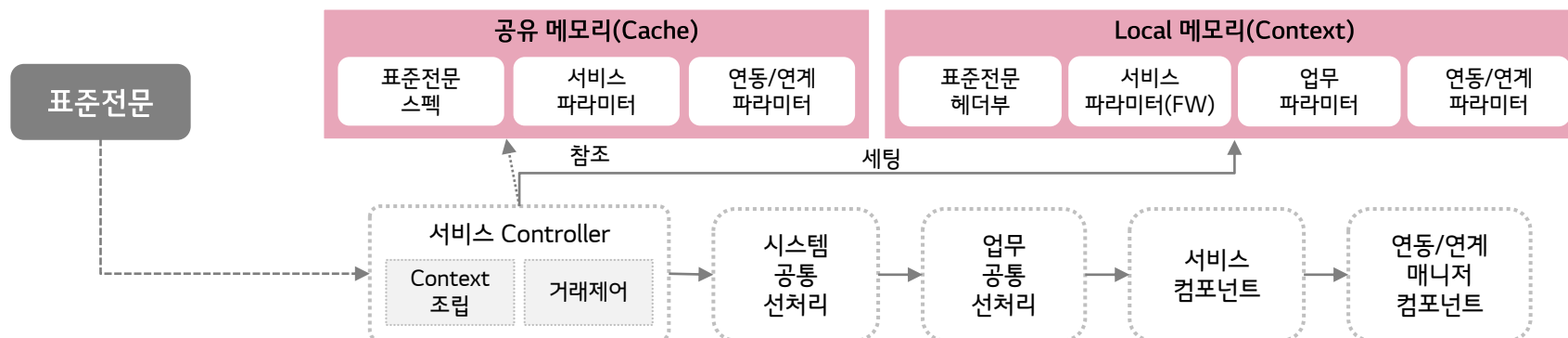
프레임워크 파라미터 관리 (2/2)

Framework 메타 데이터는 메모리 Caching하여 DB I/O 최소화 하고, 거래 단위 메모리 공유공간을 활용하여 중복 조회를 최소화 합니다.

어플리케이션 공유 영역

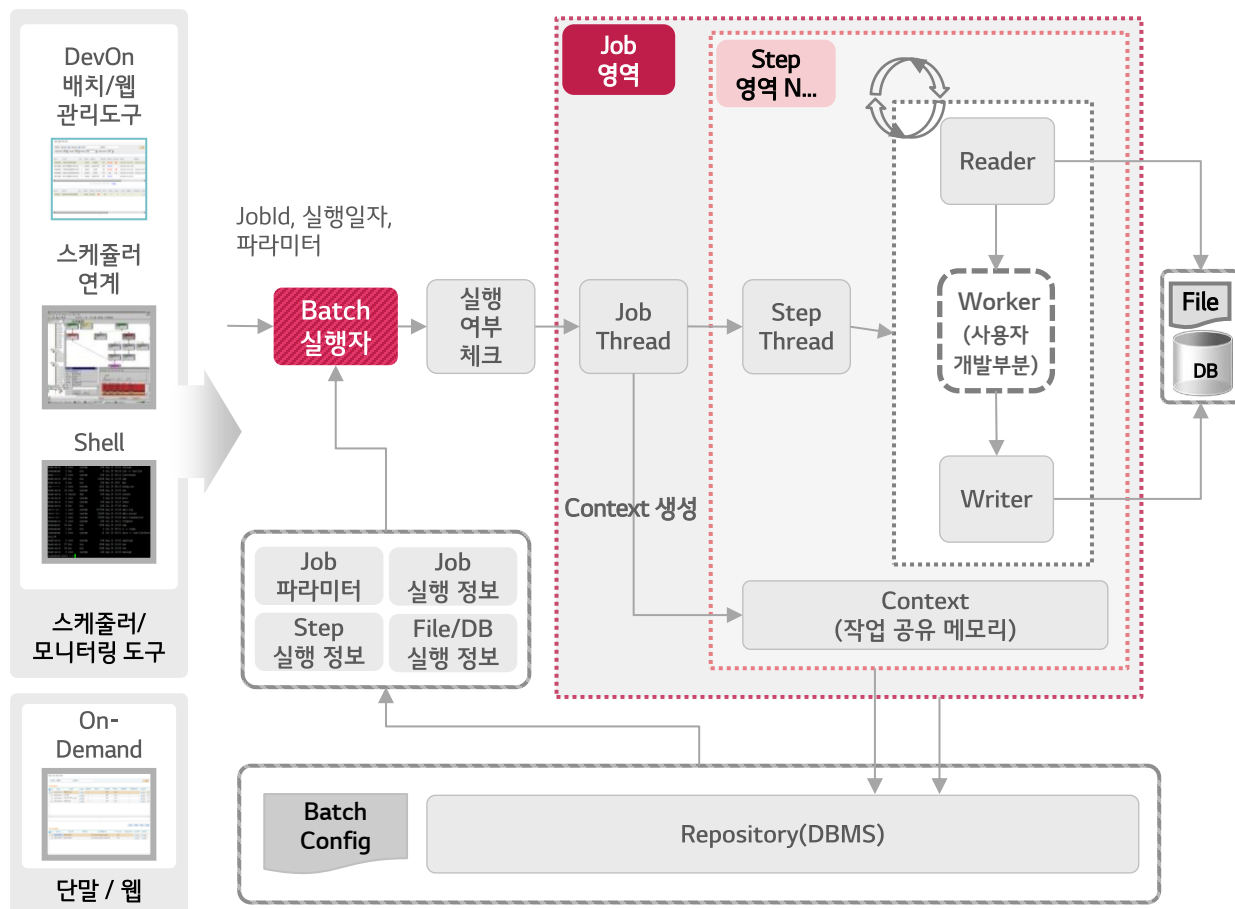


파라미터 처리 Process



배치 아키텍처

배치 아키텍처는 배치 업무 개발부터 작업 실행 및 모니터링까지 전반적인 기능을 제공하며 다양한 형식의 File I/O를 지원하는 Reader, Writer가 제공되며 도구를 이용하여 File I/O 구조를 설계할 수 있습니다.



설명

□ 다양한 유형의 금융특화 기능

- DB to DB 배치, File to File(SAM 파일, 일반 파일) 배치, On-demand 배치, 상주(데몬) 배치 기능 제공
- File 및 DB 유형에 맞게 다양한 Reader 및 Writer 제공에 따라 개발 생산성 향상

□ 개발도구 및 운영도구 제공

- FILE(SAM파일, 일반파일) 구조를 개발도구 (IO구조 정의)로 관리
- 별도의 코드변경 없이 설정을 통한 IO구조 정의 기능 제공
- 배치작업 처리 결과 및 이력을 운영관리 도구로 확인 기능 제공

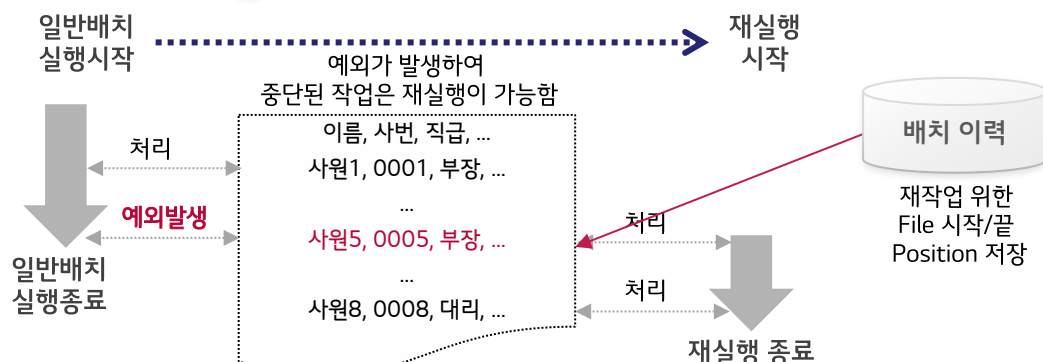
□ 스케줄러 연계

- 운영관리 도구를 통한 배치 컨트롤 및 관리 기능 제공
- 스케줄러 연동 기능 제공

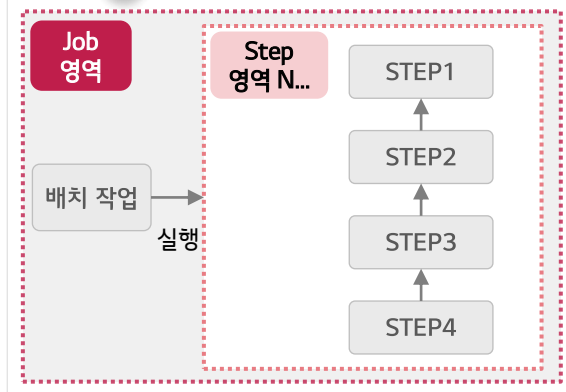
배치 작업 제어

반복적인 작업 형태를 가지는 FILE 에 대해 예외가 발생한 시점부터 재실행하여 FILE을 재작업합니다. 일반적인 배치실행, 중지, 재실행 외에도 의존 스텝 처리, 파티션 병렬처리 기능을 제공합니다.

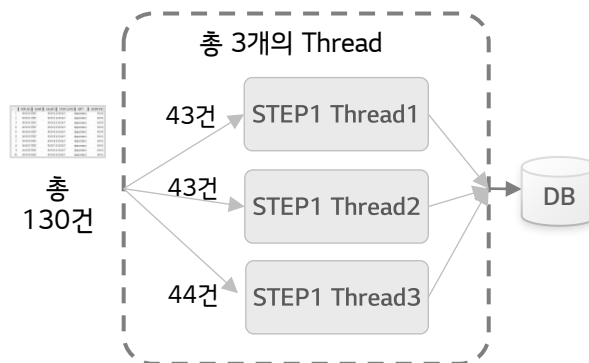
1 재실행을 위한 Restart Position 저장



2 스텝 의존관계 설정



3 병렬 처리



설명



1 배치 실행, 재실행

- 일반적인 실행, 중지, 재실행 기능
- 오류로 인한 배치의 종료시에는 운영관리에서 처음부터 수행할지 중지점에서 재작업을 할지 설정 가능
- 재작업시 예외발생 지점을 저장하여 그 부분부터 재실행 가능

2 스텝 의존관계

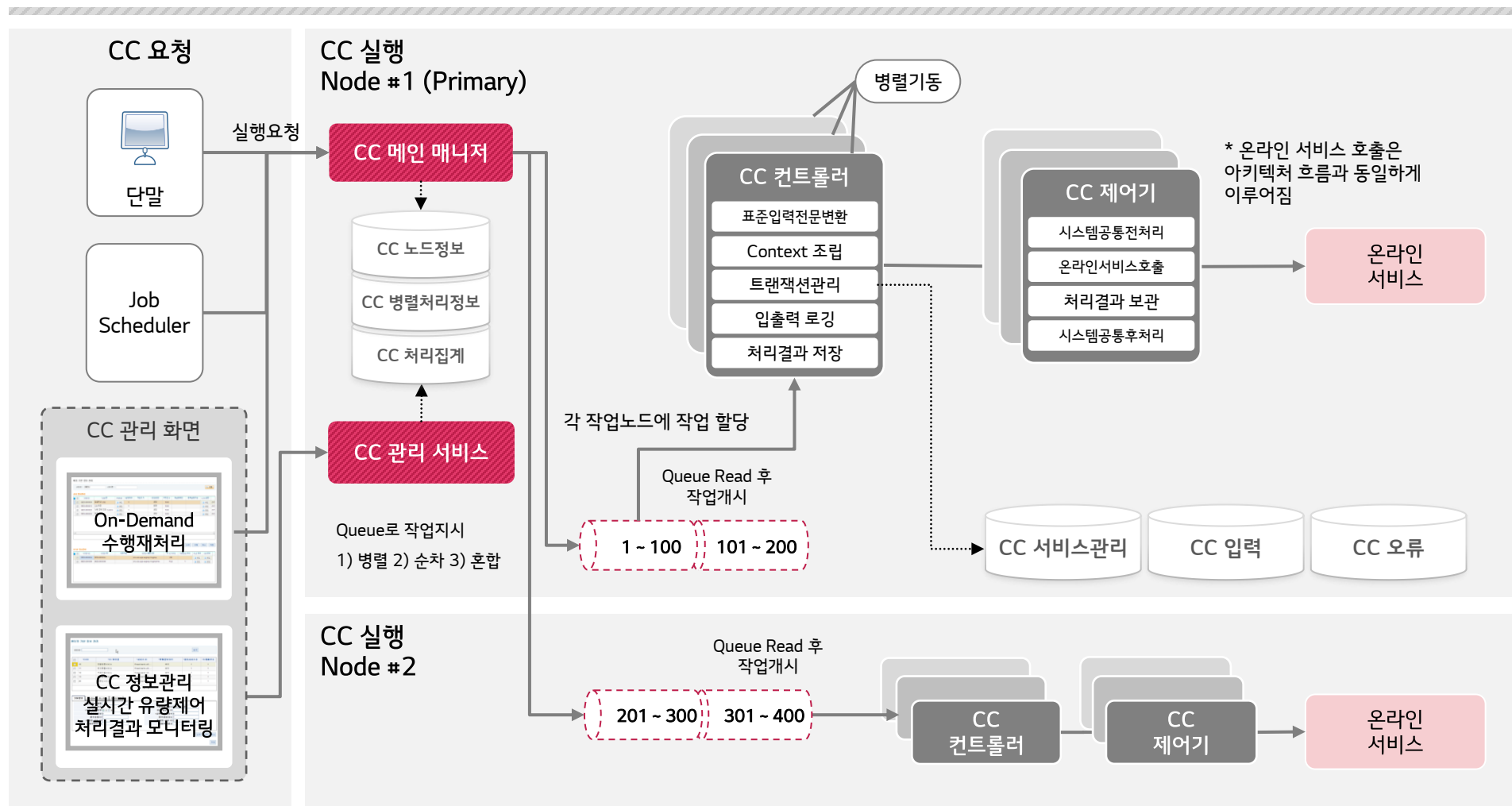
- Job 내의 여러 스텝에 관하여 의존관계를 설정가능
- 앞의 스텝이 실패할 경우 다음 스텝은 실행되지 않음

3 병렬 처리

- 설정된 Thread 개수대로 자동으로 건수 조정
- 같은 작업에 대하여 멀티 Thread로 동시 처리가능

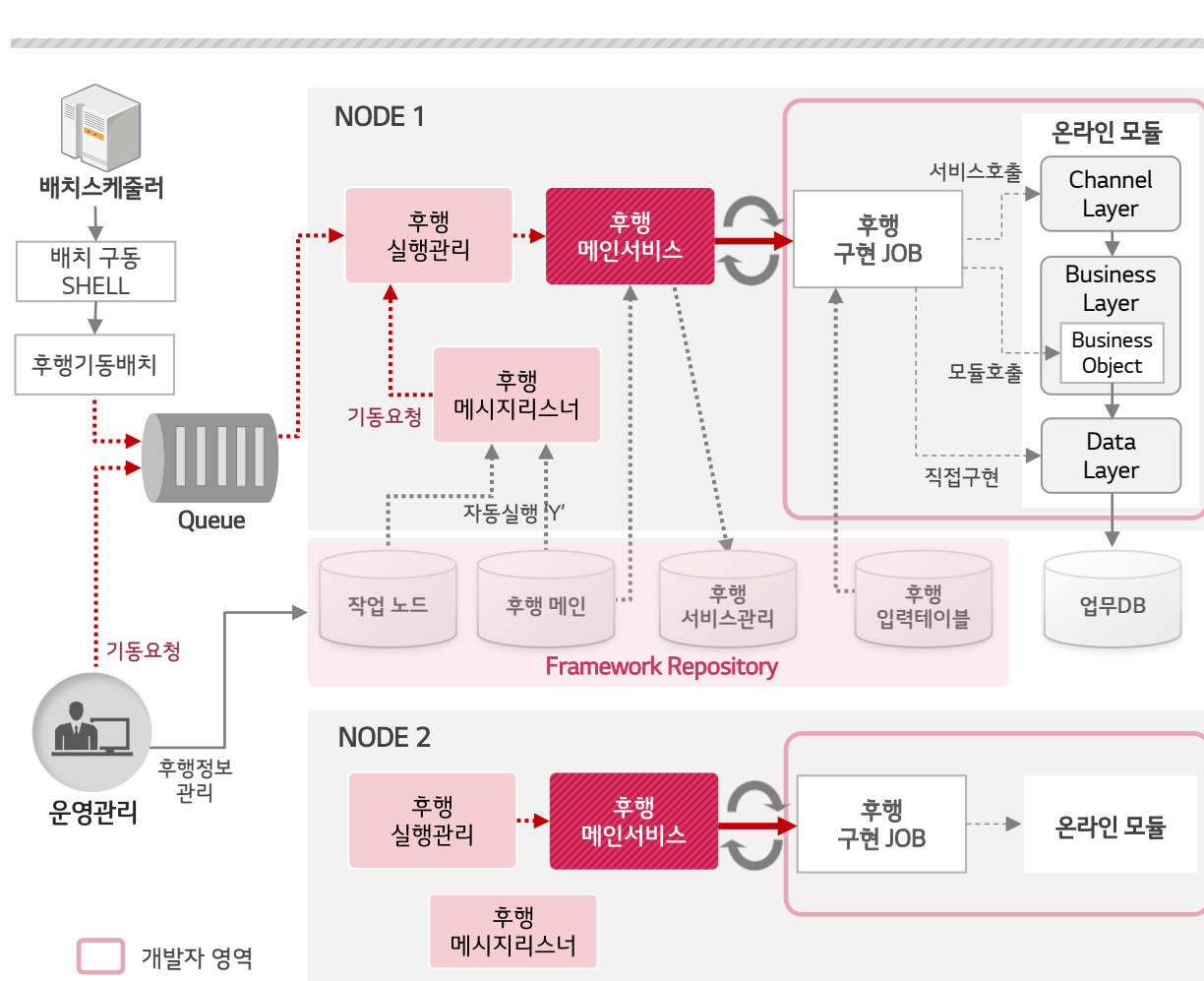
센터컷 아키텍처

센터컷 아키텍처는 신속한 대량처리를 위한 다양한 병렬처리 방식을 제공하며 시스템/작업상황에 따라 실시간으로 일시 정지, 강제종료, 오류재처리 기능을 제공하며, 센터컷 운영관리를 이용하여 센터컷 기본정보관리와 처리결과를 모니터링 합니다.



후행 아키텍처

후행 처리는 후행 메인 서비스가 Daemon형식으로 수행되며 후행기동배치, 운영관리, 자동실행 등의 기동방식을 제공합니다. 후행 구현 JOB이 주기적으로 특정 입력테이블을 모니터링하면서 데이터가 입력되면 반복작업(후행 서비스)를 실행합니다.



설명

온라인 재사용

- 온라인 서비스 재사용 기반 제공 (서비스 호출, 개발모듈 호출, 직접구현방식 제공)
- 온라인 서버와 후행 서버를 분리 구성하여 작업 영향도 최소화

준 실시간 처리

- 특정 입력테이블(업무로그 등)을 주기적으로 조회하여, 입력된 데이터를 이용해 준 실시간으로 후행 거래 처리
- 처리 프로세스 수, 모니터링 반복주기, 조회건수 등을 동적으로 제어

운영관리

- 후행 처리 현황 모니터링, 후행 처리 이력 제공
- 입력테이블 오류데이터 재처리 기능 제공 (단건, 다건 처리)

거래로그 관리

운영관리에서는 DB/File에 기록된 온라인 거래로그의 입출력 데이터를 조회할 수 있는 기능을 제공합니다. GUID, 거래식별자, 성공여부, 응답메시지, 채널구분, 실행서버 등의 다양한 조건 기반으로 특정 거래의 로그를 조회할 수 있습니다.

The screenshot shows the '거래로그' (Transaction Log) management interface. It includes search filters for '매제 [전제]', '성공여부 [전제]', 'GUID를 입력하세요', '거래식별자', '카테고리 식별자', and date/time ranges. A table lists transaction logs with columns: 성공여부, 로그일자, 추적 GUID, 카테고리, 거래식별자, 거래명, SYSTEM ID, WAS, IP주소, 매체, 상세 & 추적. A red dashed box highlights the search filters and the table header.

The screenshot shows the '거래로그 (입출력 상세화면)' (Transaction Log (Input/Output Detail View)). It displays transaction details including '추적식별자', '시스템 ID', 'WAS', '카테고리 식별자', '거래식별자', '응답코드', '로그일자', '매체', '사용자ID', and '시스템확대문자'. It also shows system information like 'TITR', 'S DO', 'CSCBS', 'sampleS', and '037'. A red dashed box highlights the transaction details section.

The screenshot shows the 'Trace Log' view, displaying detailed transaction data in a table format. It includes columns for 'SYSTEM ID', 'WAS', 'IP주소', '매체', and '상세 & 추적'. A red dashed box highlights the trace log table.

설명



- 1 다양한 검색조건**
 - 거래식별자, 성공여부, GUID, 실행서버, 채널구분 등의 다양한 조건으로 로그 조회
- 2 실시간 로그 On/Off**
 - 서비스 별로 입/출력 거래로그 실시간 조정 (켜기/끄기) 기능
- 3 통합 조회 항목 제공**
 - 거래추적 및 거래상태를 통합 조회할 수 있는 항목 제공 (로그일자, 거래정보, 성공여부, 시스템정보 등)
- 4 상세 로그정보 조회**
 - DB/FILE로 기록된 거래로그의 헤더정보를 포함한 상세 입출력 로그 조회 화면 제공
- 5 Trace 로그정보 조회**
 - 거래 추적을 위한 Trace 기능 제공

장애 추적, 시스템 진단, 거래 추적을 위한 선저 변경을 통해 On/Off 가 가능합니다.

각 Tracing 포인트에 대해 거래 별로

-
- 거래요청
- Log Agent
- 운영서버 #1
- Trace Log
- 거래요청
- Log Agent
- 운영서버 #2
- Trace Log
- 거래요청
- Log Agent
- 운영서버 #3
- Trace Log

(운영관리가 로그에이전에게 로그요청을 하면
로그에이전트는 해당 트레이스로그 파일을 GUID로
추출하여 응답데이터를 넘겨줌))

165.243.31.8:9088/management3/webdocs/common/module/TtLogView.jsp?srcId=TSHPOCUR00&buZCd=FW&guid=A5F31F08492014052812943251200001&trtDate=05

165.243.31.8:9088/management3/webdocs/common/module/TtLogView.jsp?srcId=TSHPOCUR00&buZCd=FW&guid=A5F31F08492014052812943251200001

```
(T) 19:24:32[513] AF5F31F08492014052812943251200001 ▶ [CHI] CHANNEL_CONTROLLER_START: Controller : domain.scpsample.framework.channel.controller.ScpsSampleInboundController  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 ▶ [CHI] SVC: Service-ID : TSHPOCUR00  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 ▶ [CHI] REQ_TMGS  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 00000190 000SS 20140524120000000  
000SelectEmpInfo kol.0 10000  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 ▶ [CHI] REQ_HDR  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 -----[LData Result]-----  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 KEY VALUE  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 std.tl_len | [[190]  
(T) 19:24:32[513] AF5F31F08492014052812943251200001 glob_id | [  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 pers_no | [[0]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 rast_rspn_cls_cd | [[S]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 sync_cls_cd | [[S]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 proc_rslt_cls_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 chnl_ttp_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 proc_tl_ttp_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 envr_info_cls_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 saltn_trsc_yn | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 fst_trms_sys_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 trms_sys_cd | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 trms_lstan_no | [[ 2]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 tl_rast_dttm | [[0140524120000000 ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 tl_rspn_dttm | [[ ]  
(T) 19:24:32[514] AF5F31F08492014052812943251200001 tnot_tm | [[0]
```

운영관리 거래로그 목록에서
로그 Agent를 통해 로그파일을 통합해서 조회
(운영관리가 로그에이전에게 로그요청을 하면
로그에이전트는 해당 트레이스로그 파일을 GUID로
추출하여 응답데이터를 넘겨줌))

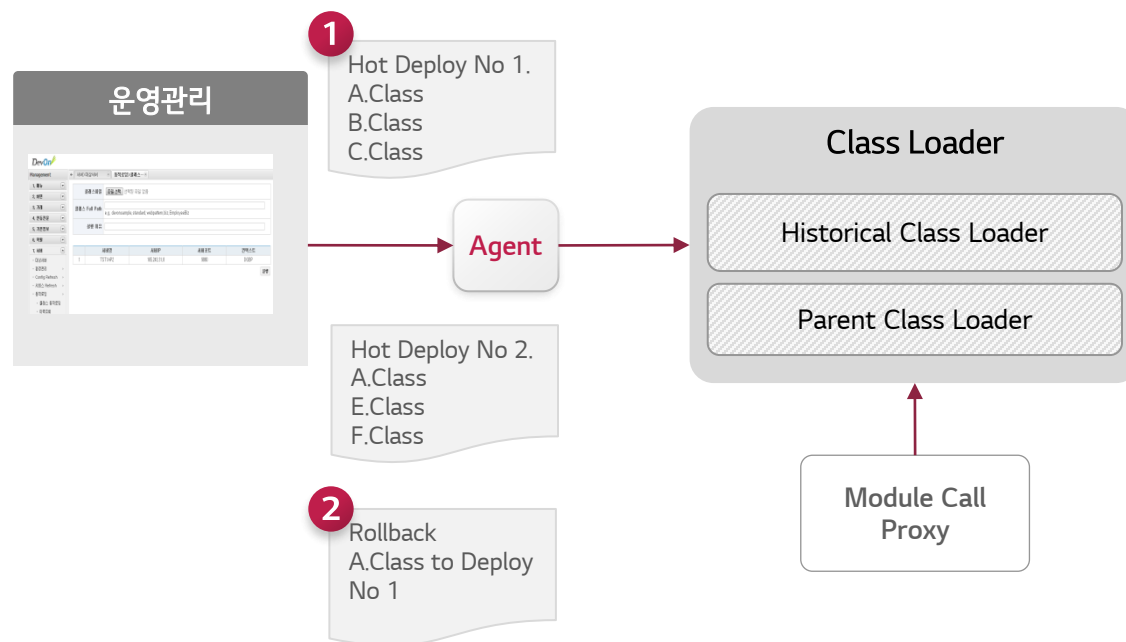
특장점

DevOn Enterprise Framework는 국내 점유율 1위 Framework로서 다양한 Reference를 보유하고 있으며 성능 및 안정성, 기능 완성도면에서 검증된 제품입니다.

지향점		특장점
금융권 최초/최대 경험을 반영한 검증된 프레임워크	검증된 기능 / 아키텍처	- 검증된 아키텍처 및 금융 특화 컴포넌트 보유 - 대내외 통신, 24*365, 센터컷/후행, 대용량처리, 통장전자표처리 등 - 다양한 채널과의 연계 가능 (통합단말, 시스템연계, 배치 연계)
Java 언어를 이용한 고성능/대량 처리 실현	고성능	- 다양한 유형의 캐시컴포넌트 제공으로 DB IO 최소화 - 거래 단위 메모리 공유공간을 활용하여 중복 조회 최소화 - 고성능 거래 로깅 기능 (비동기, 병렬, Bulk처리) 제공
금융업무 특성을 고려한 무중단 시스템	장애 대응	- 지속적으로 발생하는 장애 서비스에 대한 자동 거래 제어 - 실시간 반영 (Class Hot Deploy , 메타정보 refresh) - 거래 Timeout 기능 제공
최소 인원으로 최대 운영 효율 제공	모니터링	- 로그의 통합 관리 및 Viewer 제공 - 서비스별 거래 통계 - 장애 발생 서비스에 대한 알림 기능(SMS 등)
개발 생산성 극대화 테스트 편의성 극대화	개발 편의성	- 화면 없이 로컬/원격 거래테스트 제공 - 연계 시뮬레이션 기능 제공 - MDA 적용 시 산출물 및 소스 자동생성, 코드 표준체크 제공
다양한 업무경험을 기반으로 통합표준 프레임워크로 발전	멀티 채널 지원	- 다양한 모바일/인터넷 채널 통합 지원 - Plug & Play방식의 모듈 확장 제공

동적반영 - Custom Class Loader

Object단위의 Class Loader가 아닌 배포단위의 Class Loader를 사용하므로 적은 메모리를 사용하고 배포단위의 History가 관리되므로 동적으로 적용될 버전을 지정이 가능하여 쉽게 롤백 및 재적용이 가능합니다.



개념

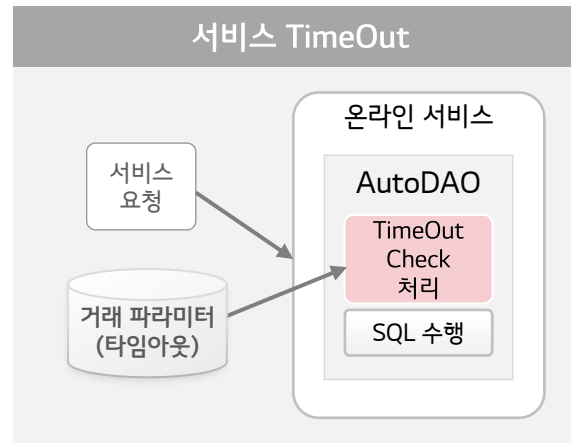
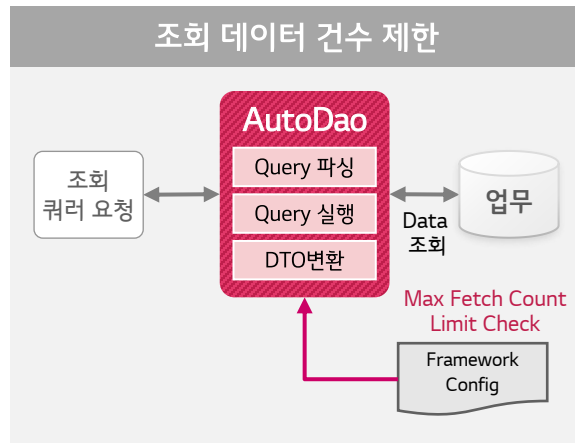
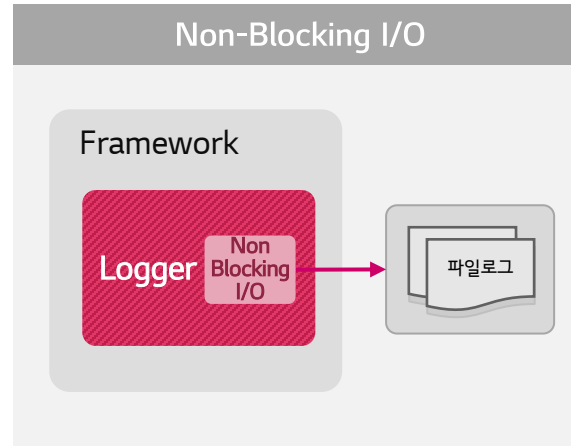
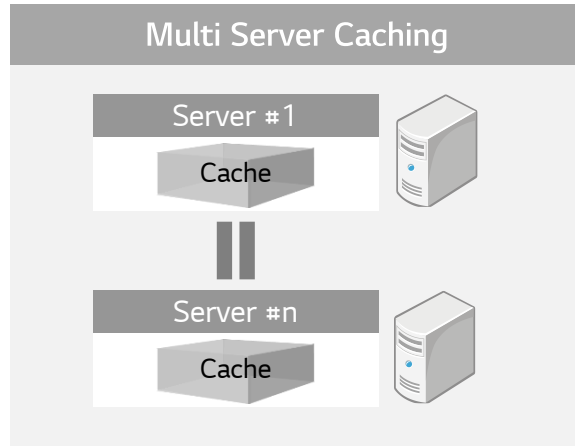
주요기능

- Historical Class Loader사용으로 적은 메모리 사용
 - Object단위의 class loader를 운영하지 않고 Hot deploy시점의 배포 단위 class loader를 사용하여 상대적으로 메모리 사용률이 적음
- Pre load 및 Lazy load 모드를 Hot Deploy 적용시점에 선택 가능
- Class Cache사용으로 빠른 class 로딩
- Historical Class Loader사용으로 운영서버 별 hot deploy로 적용된 대상 class 조회가 가능하고 필요 시 class단위로 이전 버전으로 처리가 가능

- Hot Deploy 발행 번호로 적용 대상 class 묶음을 전송하면 해당 묶음으로 Class Loader 가 생성
- A.Class 적용 후 문제가 발생해서 이전 버전으로 돌리기 위해서 운영관리에서 Deploy 대상 목록을 조회 후 롤백

대용량 TX처리를 고려한 성능 및 운영 안정성 - 온라인

온라인의 안정적인 성능확보를 위해 아래와 같은 주요 성능요소에 대해 최적의 방안으로 구성하고 목표성능을 조기에 확보합니다.



설명



Multi Server Caching

- 자주 사용하는 데이터를 Caching하여 DB I/O 최소화 하고, 거래 단위 메모리 공유공간을 활용하여 중복 조회를 최소화

Non-Blocking I/O

- I/O에 의한 Thread Block을 방지하는 Non-Blocking I/O를 사용하여 I/O성능을 확보

조회 데이터 건수 제한

- 프레임워크에 설정된 Max Fetch Count를 초과하여 데이터가 조회된 경우 ResultSet의 DTO변환을 중단으로 부하 방지

서비스 TimeOut

- 설정된 수행시간을 초과한 서비스에 대해 강제 Timeout Exception 발생시켜 자원(Thread) 점유로 인한 장애 예방

금융 차세대를 통해 검증된 프레임워크 및 인력 보유 – Reference(1/3)

국내 금융 차세대 프로젝트에서의 성공적인 구축으로 검증된 Framework 입니다.

사 업 명	발주처 명	공급일자	적용 업무
차세대시스템 구축	S보증사	2023	계정계
PA시스템 ICT인프라 고도화	K카드	2023	U2L
차세대시스템 구축	K손해보험	2022	계정계
One Payments platform 구축	D결제사	2022	계정계
프레임워크 기반 홈 리뉴얼	K카드	2021	채널계
태국법인 글로벌 IT시스템 구축	K카드	2021	국외 계정계
차세대시스템 구축	R캐피탈	2021	계정계
통합구축	S생명/O라이프	2020	처리계 및 단위시스템
IT시스템 구축	T뱅크	2020	계정계
차세대 시스템 구축	J은행	2020	계정계
BPR 차세대 시스템 구축	D은행	2020	BPR
업무계 클라우드 전환	R카드	2020	계정계
토스대행시스템 구축	H카드	2020	대행시스템
차세대 시스템 구축	H기업데이터	2020	계정계
차세대 시스템 구축	H저축은행	2019	계정계
차세대 시스템 구축	N캐피탈	2019	계정계
차세대 글로벌시스템 구축	H은행	2018	국외 계정계
차세대 시스템 구축	H국책은행	2018	계정계
영업점 전자문서 시스템 구축	W은행	2018	BPR
초개인화 시스템 구축	S카드	2018	마케팅허브
캄보디아법인 시스템 구축	K카드	2018	국외 계정계

금융 차세대를 통해 검증된 프레임워크 및 인력 보유 – Reference(2/3)

국내 금융 차세대 프로젝트에서의 성공적인 구축으로 검증된 Framework 입니다.

사 업 명	발주처 명	공급일자	적용 업무
차세대 시스템 구축	K 카드	2017	카드
차세대 시스템 구축	K 캐피탈	2017	캐피탈
캐피탈 시스템 구축	I 해외은행	2017	캐피탈
IFRS7	B 보험기관	2017	IFRS
BPR	W 카드	2017	BPR
계정계 구축	K 인터넷은행	2016	계정계
BPR	W 은행	2016	BPR
카드대행시스템 구축	K 카드	2016	대행시스템
차세대 구축	K 생명	2016	생명보험
차세대 구축	E 카드	2015.10	교통카드
차세대 구축	G 은행	2015.08	계정계
모바일/인터넷 구축	M 보험	2015.05	생명보험
통합이미지시스템 구축	S 중앙회	2015.01	이미지
차세대 구축	J 캐피탈	2014.08	캐피탈
차세대시스템 구축	S 공사	2014.07	보험
차세대 구축	P 생명	2014.03	생명보험
홈페이지 시스템 구축	I 생명	2013.11	홈페이지/모바일
BPR 시스템 구축	G 은행	2013.05	BPR
가맹점 영업지원 구축	H 카드	2013.03	SFA
차세대 시스템 구축	N 손보	2013.02	손해보험

금융 차세대를 통해 검증된 프레임워크 및 인력 보유 – Reference(3/3)

금융 산업 전반의 Core 계정계 시스템 뿐 아니라 BPR, 정보계, 채널계 시스템까지 금융 산업 전반의 폭넓은 Reference를 보유하고 있습니다.

사 업 명	발주처 명	공급일자	적용 업무
디지털생명 시스템 구축	K 생명	2012.10	생명보험
모바일 banking 구축	J 은행	2012.08	채널계
모바일 플랫폼 구축	K 생명	2012.04	채널계
차세대시스템 구축	J 은행	2012.02	계정계, 정보계
차세대시스템 구축	R 카드	2012.04	계정계
차세대시스템 구축	S 카드	2011.05	계정계
홈페이지/퇴직연금/다이렉트 구축	K 생명	2011.03	채널계
기간계 시스템 구축	J 생명	2011.03	생명보험
차세대시스템 구축	A 캐피탈	2011.01	캐피탈
카드 정보계 구축	N 카드	2011.01	카드 정보계
스마트지점 구축	K 은행	2011.12	단위시스템
CRM 시스템 구축	H 카드	2010.06	정보계
홈페이지 구축	S 카드	2010.03	홈페이지
차세대시스템 구축	B 카드	2010.03	카드