

**[IxVDR] Introduction**



**Copyright Notice**

Copyright © 2009 Inzent Co., Ltd. All Rights Reserved.

대한민국 서울특별시 영등포구 대림3동 779-10 콤텍빌딩 2층

**Restricted Rights Legend**

All Inzent Software (iXVDR) and documents are protected by copyright laws and the Protection Act of Computer Programs, and international convention. Inzent software and documents are made available under the terms of the Inzent License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of Inzent Co., Ltd.

이 소프트웨어(iXVDR) 사용설명서의 내용과 프로그램은 저작권법, 컴퓨터프로그램보호법 및 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 Inzent Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 Inzent의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

**Trademarks**

iXVDR are trademark of Inzent Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

iXVDR 는 Inzent Co., Ltd.의 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

**Open Source Software Notice**

This product includes various open source software that has been developed and/or licensed by “Spring Framework”, “Hibernate”, “Terracotta, Inc”, “Apache Foundation”, “Bela Ban”, “QOS.ch”, ”XStream”. Inzent fully respects the aforementioned parties and the open source software used in this product. More details can be found within the directory “${INSTALL\_PATH}/license/oss\_licenses”

본 제품은 “Spring Framework”, “Hibernate”, “Terracotta, Inc”, “Apache Foundation”, “Bela Ban”, “QOS.ch”, ”XStream” 에 의해 개발 또는 라이선스된 오픈 소스 소프트웨어를 포함합니다. 오픈 소스 소프트웨어와 개발자에 대해 경의를 표합니다. 관련 상세 정보는 제품의 디렉터리 “${INSTALL\_PATH}/license/oss\_licenses”에 기재된 사항을 참고해 주십시오.

* 오픈소스 라이선스 넣어주세요

**사용설명서 정보**

제목: iXVDR Introduction

버전: v1.0.0

**목 차**

[1 iXVDR 소개 5](#_iXVDR_소개)

[1.1 개요 5](#_Toc35850694)

[1.1.1 iXVDR란 5](#_Toc35850695)

[1.1.2 iXVDR의 기대효과 5](#_Toc35850696)

[1.2 시스템 구성도 6](#_Toc35850697)

[2 실행 환경 7](#_Toc35850698)

[2.1 iXVDR Architecture 7](#_Toc35850699)

[2.2 구성 요소 7](#_Toc35850700)

[2.2.1 구성 요소1 7](#_Toc35850701)

[2.2.2 구성 요소2 7](#_Toc35850702)

[2.2.3 구성 요소3 9](#_Toc35850703)

[2.2.4 구성 요소4 9](#_Toc35850704)

[2.3 iXVDR 주요 기능 9](#_Toc35850705)

[2.3.1 룸 생성 9](#_Toc35850706)

[2.3.2 계약 관리 10](#_Toc35850707)

[2.3.3 권한 설정 10](#_Toc35850708)

[2.3.4 샌드박스 11](#_Toc35850709)

[3 사용 방법 13](#_Toc35850710)

[3.1 회원가입 및 로그인 13](#_Toc35850711)

[3.2 새로운 룸 생성하기 13](#_Toc35850712)

[3.3 사용자 관리 14](#_Toc35850713)

[3.3.1 사용자 초대 14](#_Toc35850714)

[3.3.2 사용자 권한 설정 14](#_Toc35850715)

[3.3.3 사용자 삭제 14](#_Toc35850716)

[3.3.4 사용자 그룹 관리 15](#_Toc35850717)

그림 목차

[**[그림 1‑１]** 시스템 구성도 14](#_Toc340603345)

[**[그림 2‑１]** iXVDR Runtime Architecture 15](#_Toc340603346)

[**[그림 2‑２] 로컬 서비스 흐름도** 19](#_Toc340603347)

[**[그림 2‑３]** 일반 서비스 흐름도 20](#_Toc340603348)

[**[그림 2‑４]** 복합서비스 흐름도 20](#_Toc340603349)

[[그림 2‑５] 계약관리 흐름도 21](#_Toc340603350)

[**[그림 3‑１]** iXVDR Architecture 27](#_Toc340603360)

[[그림 3‑２] Meta정보 실시간적용 흐름도 28](#_Toc340603361)

# iXVDR 소개

본 장에서는 iXVDR에 대한 기본적인 이해와 iXVDR의 개념과 아키텍처를 기술한다.

## 개요

예문

### iXVDR란

iXVDR는 솔루션이다.

* 내용
  + 내용1
  + 내용2

### iXVDR의 기대효과

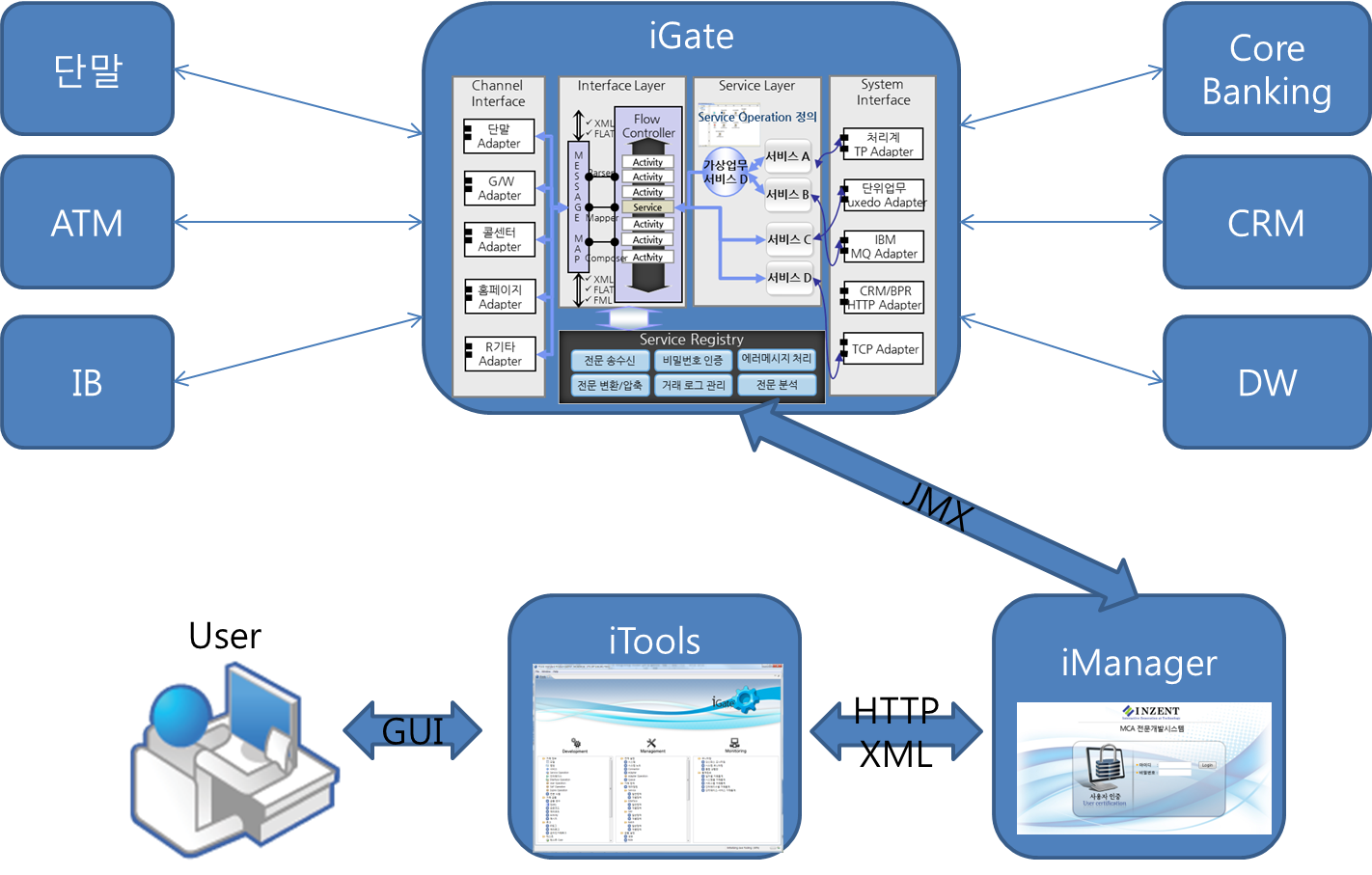
iXVDR는 내/외부 서비스을 실현한다.

다음과 같은 효과를 얻을 수 있다**.**

* 비즈니스 경쟁력 강화
  + 예문1
  + 예문2
  + 예문3
  + 예문4
* Middle layer Integration
  + 예문1
  + 예문2
* 운영 및 유지보수 용이성 확보
  + 예문1
  + 예문2
  + 예문3

## 시스템 구성도

솔루션은 구성되어있다.



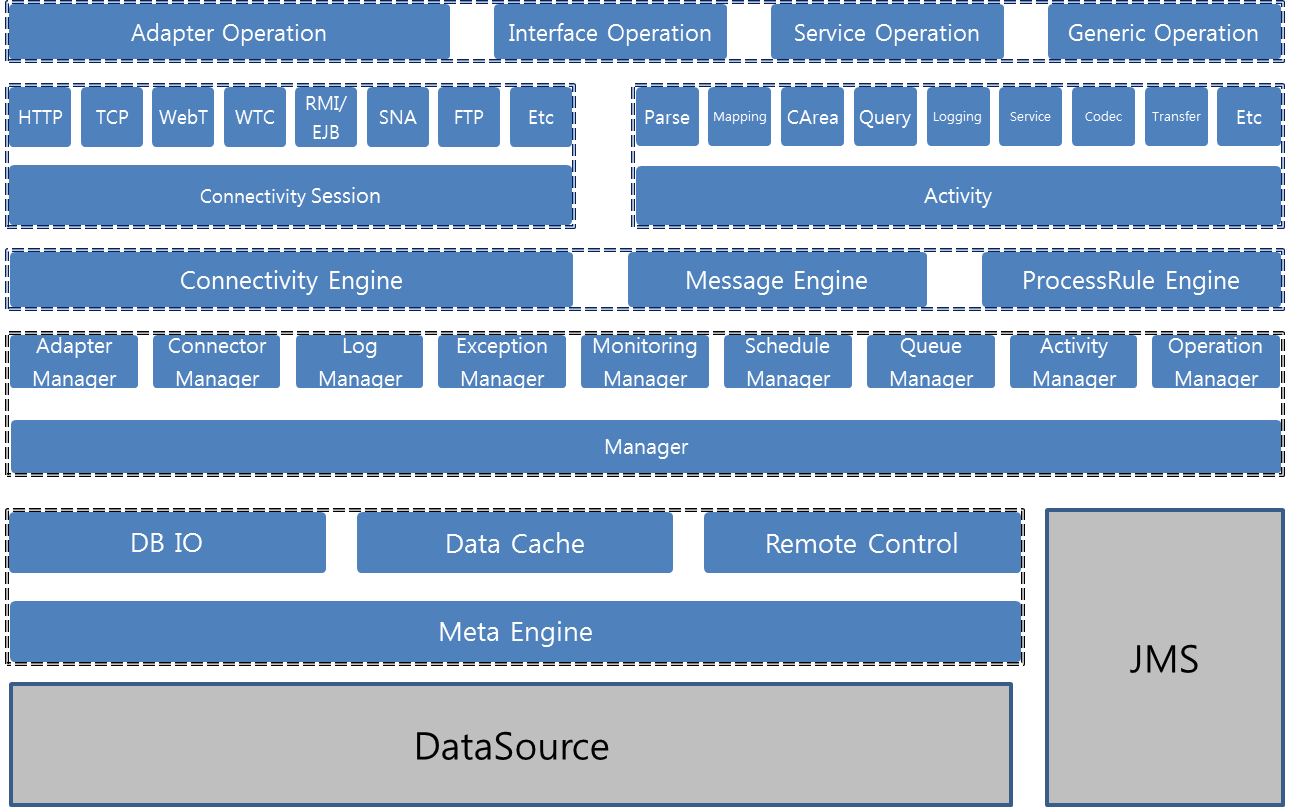
[그림 1‑１] 시스템 구성도

# 실행 환경

본 장에서는 iXVDR의 실행 환경 아키텍처와 그 기능들을 기술한다.

## iXVDR Architecture

예) iXVDR는 [그림 2.1] iXVDR Runtime Architecture와 같이 여러 구성요소들이 계층을 이루어 상호작용을 한다.



**[그림 2‑１]** iXVDR Runtime Architecture

## 구성 요소

### 구성 요소1

예) 구성 요소1은 역할을 한다.

|  |  |
| --- | --- |
| **주요 기능** | **비고** |
| **DBMS IO** | DB에 저장된 각종 Meta-data를 읽어 이를 Value-Object로 변환시킨다. |
| **Data Cache** | DB 장애 대응 및 DB IO 개선을 위해 각종 Meta-data를 JVM heap에 cache한다. |
| **Remote Control** | 거래처리 중에도 각종 Meta-data의 변경 내용이 거래에 적용된다. |

### 구성 요소2

예문) Connectivity Engine는 각종 연계 시스템들과 물리적인 통신을 담당하면서, 이들과의 데이터 송수신을 전담하여 처리한다.

예문) XVDR는 각종 상이한 통신 프로토콜을 보다 효율적으로 지원하기 위해서 각종 통신 프로토콜마다 이에 대응하는 Connection-Session이 제공된다.  
이를 통하여 여러 다양한 통신 프로토콜을 내부 표준으로 변환하여 일관되게 데이터 송수신을 수행한다

* In-House System

예문) iXVDR는 각종 In-House System들과 연계를 위해서 아래와 같은 Connection-Session을 제공한다.

|  |  |
| --- | --- |
| **프로토콜** | **비고** |
| **HTTP** | HTTP(HyperText Transfer Protocol RFC2616)를 사용하여 전문을 주고 받는 통신 방식을 지원한다. |
| **iXVDR@Link** | In-House System 연계 지원용 API가 iXVDR@Link이다. 이 API를 사용한 통신 방식을 지원한다. |
| **TCP/IP** | TCP/IP를 사용하여 일정한 길이정보를 전송데이터에 포함시키는 등의 방식으로 전문을 주고 받는 통신 방식을 지원한다. |
| **Java™ RMI** | Java™ Remote Method Invocation을 사용하여 연계시스템과 전문을 주고 받는 통신 방식을 지원한다. |
| **FTP** | FTP(File Transfer Protocol RFC959)를 사용하여 연계 시스템과 파일을 주고 받는 통신 방식을 지원한다. |

* Middle-Ware

예문) iXVDR는 다양한 Well-Known Middle-Ware들과의 연동을 지원한다.

|  |  |  |
| --- | --- | --- |
| **이름** | **비고** | |
| **Tmax** | Tmax사의 Java 라이브러리인 WebT, WebTAsync을 사용하여 Tmax Service로 전문을 주고 받는 통신 방식을 지원한다. | |
| **Tuxedo** | Oracle사의 WTC(WebLogic Tuxedo Connector), Jolt와 Tmax사의 JTC(JEUS-Tuxedo Connector)를 사용하여 Tuxedo Service와 전문을 주고 받는 통신 방식을 지원한다. | |
| **VTAM** | IBM사의 SNA LU0 프로토콜을 사용하여 Mainframe Service와 전문을 주고 받는 통신 방식을 지원한다. | |
| **Web Application Server** | HTTP | 보편적인 WAS(JEUS, Weblogic, Websphere)에서 기동되는 서비스들과 연계를 하기 위해 HTTP 통신 프로토콜을 지원한다. |
| EJB | EJB(Enterprise JavaBeans)을 사용하여 WAS에 설치된 서비스들과 전문을 주고 받는 통신 방식을 지원한다 |
| RMI | Java™ Remote Method Invocation을 사용하여 WAS에 설치된 서비스들과 전문을 주고 받는 통신 방식을 지원한다. |

* Application Framework

예문) iXVDR는 다양한 Application Framework들과의 연동을 지원한다.

|  |  |
| --- | --- |
| **이름** | **비고** |
| **ProFrame** | Tmax사의 Application Framework 제품인 ProFrame과 전문을 주고 받는 통신 방식을 지원한다. |
| **AnyFrame** | SDS사의 Java 기반 Application Framework 제품인 AnyFrame과 전문을 주고 받는 통신 방식을 지원한다. |
| **SYSTEMiER** | SDS사의 C 기반 Application Framework 제품인 SYSTEMiER와 전문을 주고 받는 통신 방식을 지원한다. |
| **DevOn-Java** | LG CNS사의 Java 기반 Application Framework 제품인 DevOn-Java와 전문을 주고 받는 통신 방식을 지원한다. |
| **DevOn-C** | LG CNS사의 C 기반 Application Framework 제품인 DevOn-C와 전문을 주고 받는 통신 방식을 지원한다. |
| **BANCS** | Financial Network Services사의 Application Framework 제품인 BANCS와 전문을 주고 받는 통신을 지원한다. |

### 구성 요소3

예문) Message Engine은 iXVDR에서 다루는 데이터 처리 기능이 집약된 Component이다.

* Parser

iXVDR의 Parser는 Connection-Session을 통해서 수신한 데이터를 해석하여 이를 Data-Object로 변환하는 역할을 한다.

### 구성 요소4

예문) iXVDR에서는 각종 비즈니스 로직을 Activity 혹은 Operation 형태로 작성한다. 이렇게 사용자가 작성한 각종 비즈니스 로직을 물리적으로 실행하는 Component가 ProcessRule Engine이다.

* Activity

Activity는 가장 기초 단위가 되는 비즈니스 로직이 담겨지는 Component인다. 이 Activity는 Java™ Language로 작성된다.

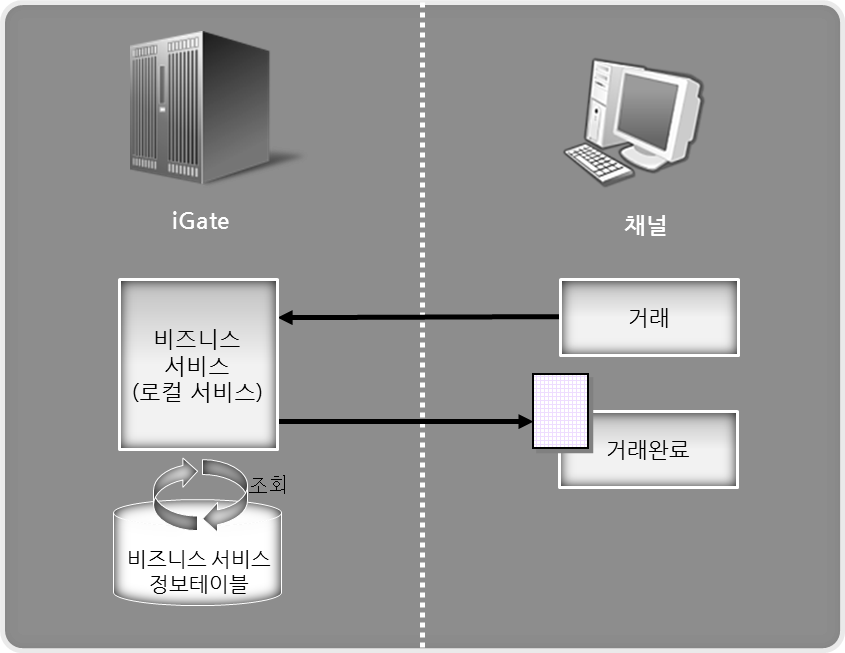
Java™로 작성하기 때문에 Java™로 작성된 각종 외부 Library를 참조할 수 있을 뿐만 아니라, Java™ Language에서 얻은 수 있는 각종 장점(예들 들어 Object-Oriented)들을 사용하여 비즈니스 로직을 구현할 수 있다.

## iXVDR 주요 기능

### 룸 생성

예문) 로컬 서비스는 iXVDR내 정의되어 있는 서비스이다.

외부 시스템과 이러한 로컬 서비스를 연계하는 거래 유형이다.

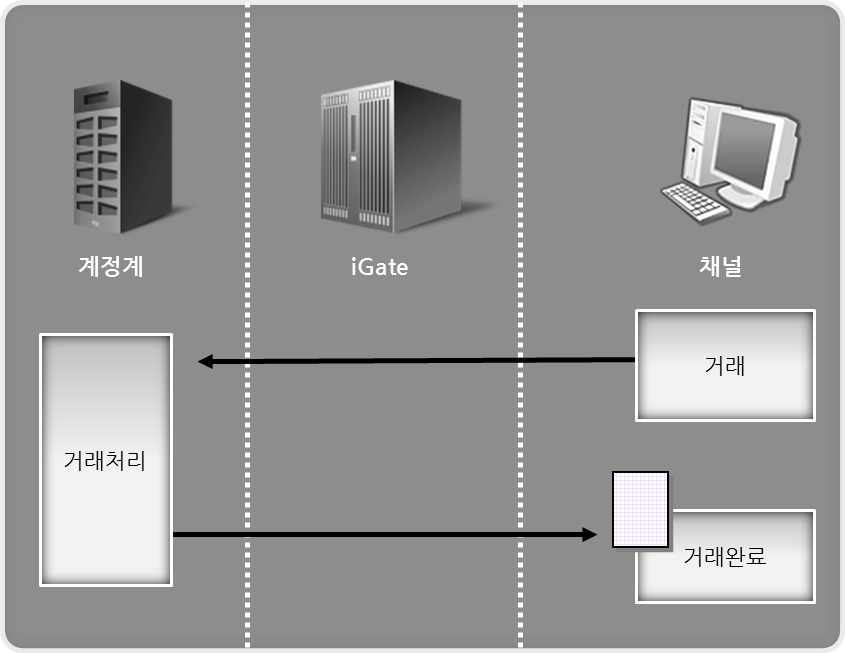


**[그림 2‑２] 로컬 서비스 흐름도**

### 계약 관리

예문) 일반 서비스는 외부 시스템에 존재하는 서비스이다.

일반 서비스 거래는 기타 외부 시스템에서 일반 서비스를 사용하여 거래하는 유형이다. 인터페이스 시스템으로부터 요청 메시지가 입력되면, 해당 서비스가 수행된 후 요청한 시스템으로 응답 메시지를 출력한다.



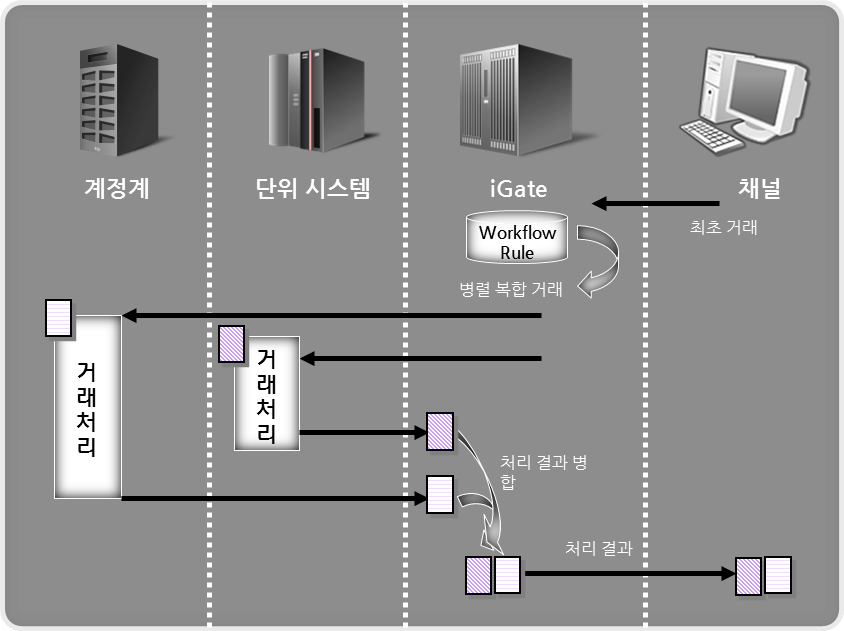
[그림 2‑３] 일반 서비스 흐름도

### 권한 설정

복합 서비스는 여러 서비스들을 조합하여 정의되는 가상의 서비스이다.

iXVDR는 SOA을 기반으로 작성된 서비스(재사용이 가능한 세분화된 서비스)들을 유기적으로 조합하여 다양한 복합 거래를 지원한다.  
업무 개발자는 Service Operation Designer를 이용하여 편리하게 복합 서비스를 작성할 수 있다.

복합 서비스는 비즈니스 변화를 신속하게 수용하고 적용할 수 있는 유연성을 제공한다.



[그림 2‑４] 복합서비스 흐름도

### 샌드박스

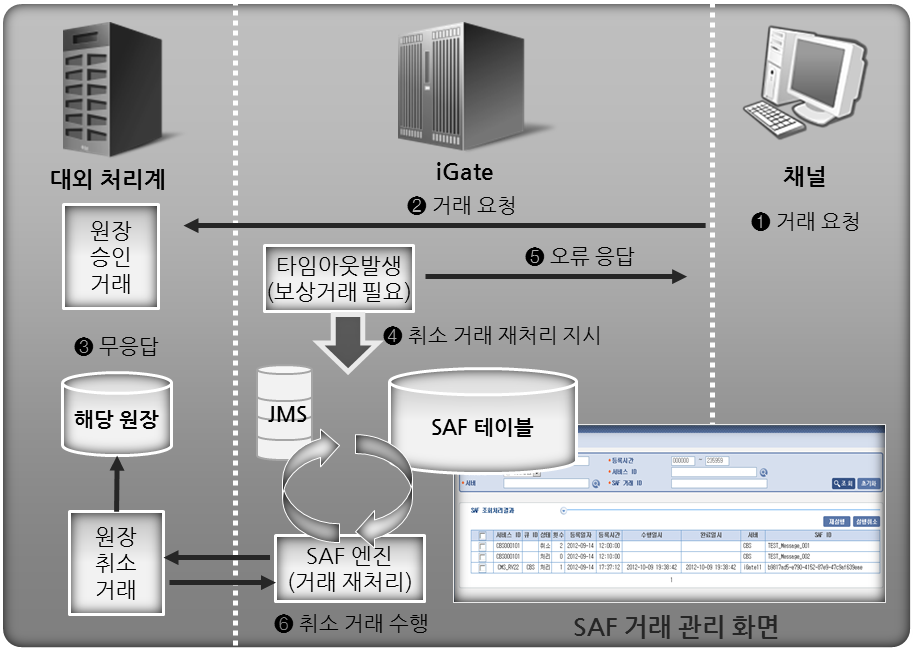
예문) 대외 거래는 대외기관과 수행하는 거래이다.

iXVDR는 거래보장 및 정합성을 제공하기 위한 SAF(Store And Forward)거래와 대외기관이 필요로 하는 파일을 처리하기 위한 파일 송수신거래를 지원 한다.

SAF(Store and Forward)거래

거래보장 및 거래정합성을 지원하기 위해서 SAF엔진 기반의 재처리기능을 제공한다.

iXVDR는 거래보장을 지원하기 위해 메시지를 SAF테이블에 저장하고, 거래처리 중 에러가 발생 하면 메시지를 재전송한다. 또한 거래정합성을 지원하기 위해 응답 처리 중 에러가 발생하면 SAF테이블에 취소전문을 저장하고 취소거래를 재전송한다.



[그림 2‑５] SAF거래처리 흐름도

파일송수신거래

예문) 대외기관과의 파일전송처리를 위하여 지원되는 거래로서 EDI방식과 표준FTP방식이 있다.

EDI(Electronic Data Interchange) 방식

EDI 전송 규약에 따라 파일을 정해진 크기만큼 나누어 송/수신하고 파일 송/수신 규약 처리는 업무 처리 Rule을 구성하여 처리한다. 동일한 규약의 신규 기관이 추가되면 기존의 업무처리 Rule을 재사용하여 구현한다.

표준FTP방식

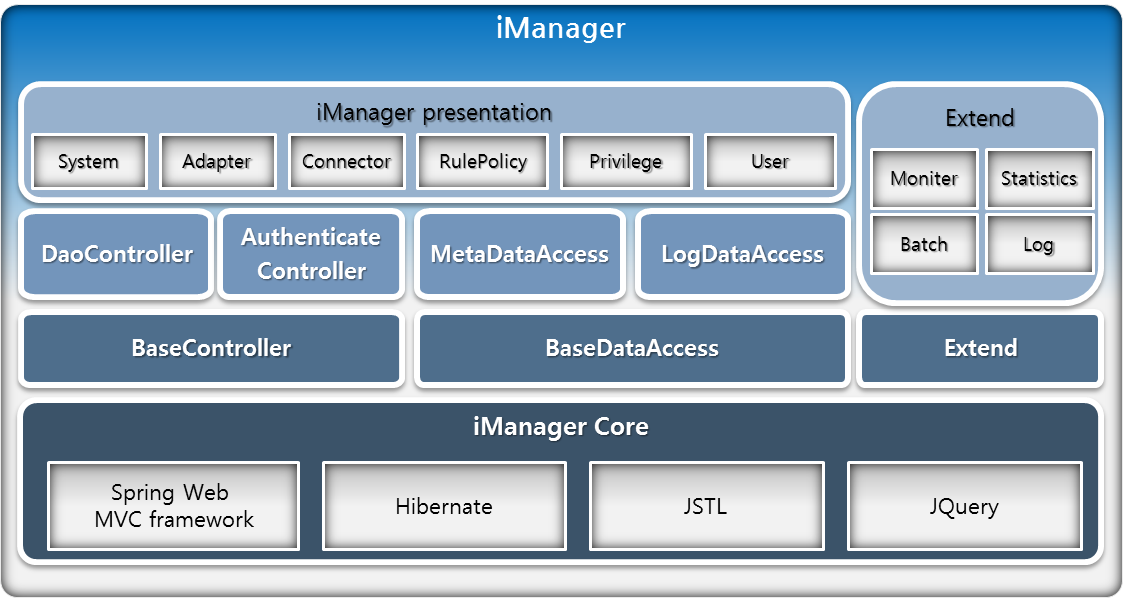
기관별 정의된 규약에 따라 표준 FTP 방식의 파일 송/수신 처리한다. 대내 시스템으로부터 수신한 파일을 파일 처리 Rule에 따라 송신하고, 대외 시스템으로부터 수신한 파일은 처리 Rule에 따라 대내 시스템으로 전송한다.

# 사용 방법

예문) 본 장에서는 iXVDR의 관리 환경 아키텍처와 그 기능들을 기술한다.

## 회원가입 및 로그인

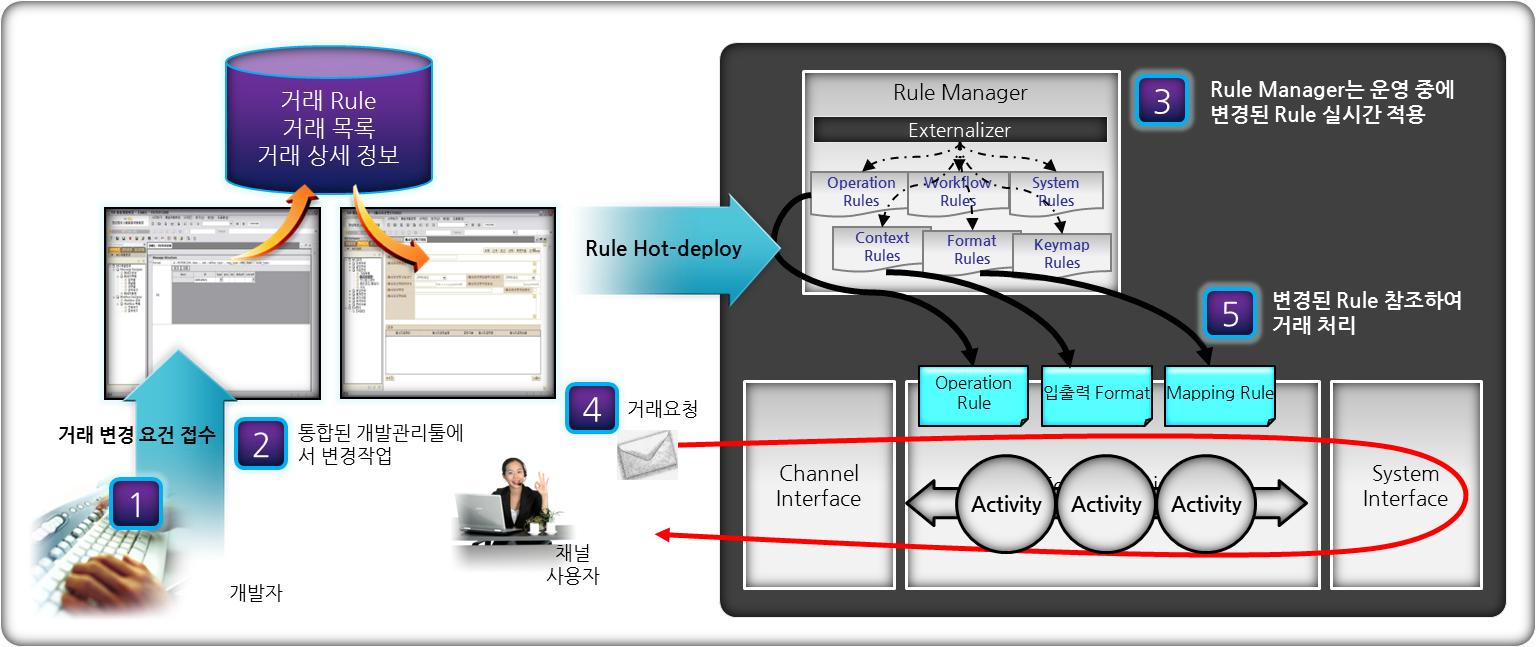
예문) iManager은 Spring Web MVC를 Base로 하여 다양한 데이터의 조회 및 관리를 유연하게 처리할 수 있도록 개발됨



**[그림 3‑１]** iManager Architecture

## 새로운 룸 생성하기

예문) iXVDR에서 사용되는 모든 Meta 데이터는 각각에 대응되는 Meta 관리 Component가 있으며, 이를 통해서 Meta 정보가 관리된다. Meta 관리 Component를 통하여 시스템 중단 없이 Meta 정보를 실시간으로 적용할 수 있다.



[그림 3‑２] Meta정보 실시간적용 흐름도

## 사용자 관리

예문) iManager는 iXVDR와 연계할 시스템의 정보를 정의하기 위해 System, System Node, Adapter 그리고 Connector 정보를 설정한다.

### 사용자 초대

예문) 연계할 시스템 및 전문 포맷에 대하여 정의하는 정보이다.

시스템정보에는 요청/응답 Data-Model, Charset, Parser, Composer등에 정보를 설정한다.

|  |  |
| --- | --- |
| 등록항목 | **설명** |
| **ID** | 연계 시스템의 식별자 |
| **Charset** | Parser 와 Composer에서 인코딩시 사용되는 문자셋을 입력한다. |
| **Endian** | Byte Order 방식을 지정한다. |
| **요청 Data-Model** | 요청 Data를 분석할 수 있는 Data-Model을 선택한다. |
| **응답 Data-Model** | 응답 Data를 분석할 수 있는 Data-Model을 선택한다. |
| **Parser** | 연계시스템에서 사용할 Parser를 선택한다. |
| **Composer** | 연계시스템에서 사용할 Composer를 선택한다. |
| **설명** | 연계시스템에 대한 설명을 입력한다. |

### 사용자 권한 설정

예문) 연계 시스템의 물리적인 정보를 정의한다.

System Node 정보에는 SystemID, SystemNode, NodeIP, IP로깅여부, 접속허용여부 정보를 설정한다.

|  |  |
| --- | --- |
| **등록항목** | **설명** |
| **SystemID** | System의 식별자. |
| **SystemNode** | 연계 시스템의 SystemNode의 식별자. |
| **Node IP** | **시스템노드의** IP주소를 입력한다. |
| **IP로깅여부** | IP**로그의** 출력 여부를 선택한다. |
| **접속허용여부** | Client의 접속 허용 여부를 선택한다. |

### 사용자 삭제

예문) 연계 시스템에서 발생하는 통신 Event에 따라 통신 처리 로직을 선택한다.

AdapterID, 시스템ID, 설명, 이벤트List를 설정한다.

|  |  |
| --- | --- |
| **등록항목** | **설명** |
| **AdapterID** | AdapterID를 입력한다. |
| **시스템ID** | 연계할 시스템ID를 선택한다. |
| **설명** | Adapter에 대한 설명을 입력한다. |
| **이벤트리스트** | Event에 따른 Operation을 선택한다. |

Event유형은 아래와 같다.

|  |  |
| --- | --- |
| **Event** | **설명** |
| **create** | Session이 생성될 때 처리하는 이벤트 |
| **read.response** | 응답 데이터를 읽었을 때 처리하는 이벤트 |
| **read.request** | 요청 데이터를 읽었을 때 처리하는 이벤트 |
| **write.response** | 응답 데이터를 생성할 때 처리하는 이벤트 |
| **write.request** | 요청 데이터를 생성할 때 처리하는 이벤트 |
| **extract** | 연계 시스템에서 데이터를 추출할 때 처리하는 이벤트 |
| **exception** | 예외상황이 발생했을 때 처리하는 이벤트 |
| **close** | Session이 종료될 때 처리하는 이벤트 |

### 사용자 그룹 관리

예문) 연계할 시스템에 대한 통신관련 정보를 설정한다.

Connector 설정에는 Connector 유형 및 Timeout, Port 등의 통신속성에 관하여 등록한다.

|  |  |
| --- | --- |
| **등록항목** | **설명** |
| **ConnectorID** | Connector 식별자. |
| **Class** | Connector의 유형을 선택한다. |
| **Log Level** | Connector에 적용될 로그레벨을 선택한다. |
| **RequestDirection** | Connector의 입/출력 방향을 선택한다. |
| **시스템ID** | 연계시스템ID를 선택한다. |
| **시스템노드** | 연계시스템노드를 선택한다. |
| **AdapterID** | AdapterID를 선택한다. |
| **자동기동여부** | iXVDR시스템이 기동할 때 Connector의 자동기동여부를 선택한다. |
| **설명** | Connector에 대한 설명을 입력한다. |
| **속성리스트** | Connector에 대한 통신속성을 입력한다. |
| **적용서버ID** | Connector을 적용할 서버ID를 선택한다. |



Technology delivers the happiness

서울시 여의도 유화증권빌딩9층  
**www.inzent.com** | Tel: 02) 6312-3333 | Fax: 02)787-3699

