

Data Interactive Flow 1.5



Intelligent FACTorys DiFlow

# I-FACTs DiFlow

SK(주) C&C | 제조Solution Digital그룹

# CONTENT .....

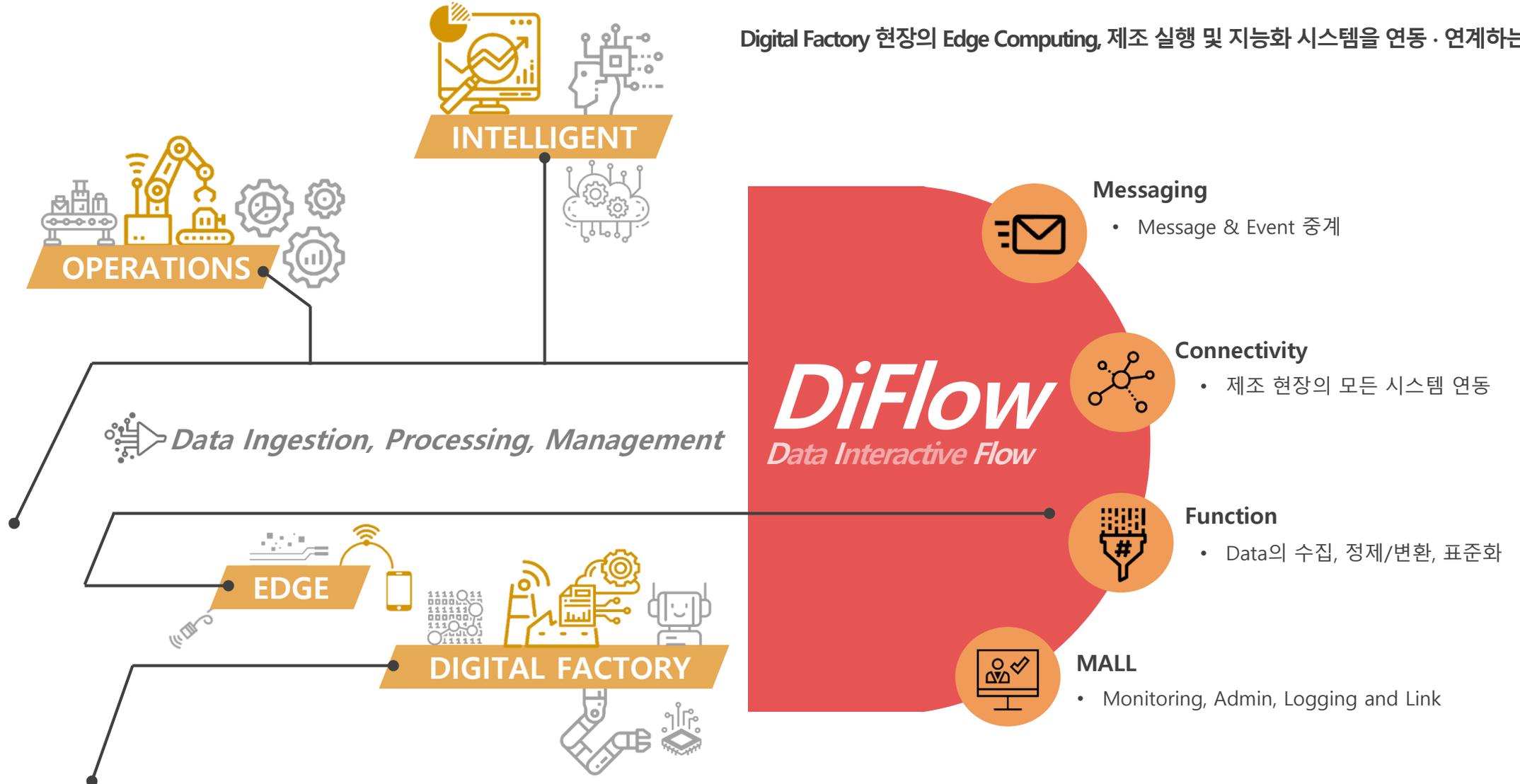
Data Interactive Flow 1.5



- **What is I-FACTs DiFlow ?**
- **Key Features**
- **Customer Values**
- **Business Offering**

# What is 'I-FACTs DiFlow'?

Digital Factory 현장의 Edge Computing, 제조 실행 및 지능화 시스템을 연동 · 연계하는 솔루션



# Key Features of DiFlow

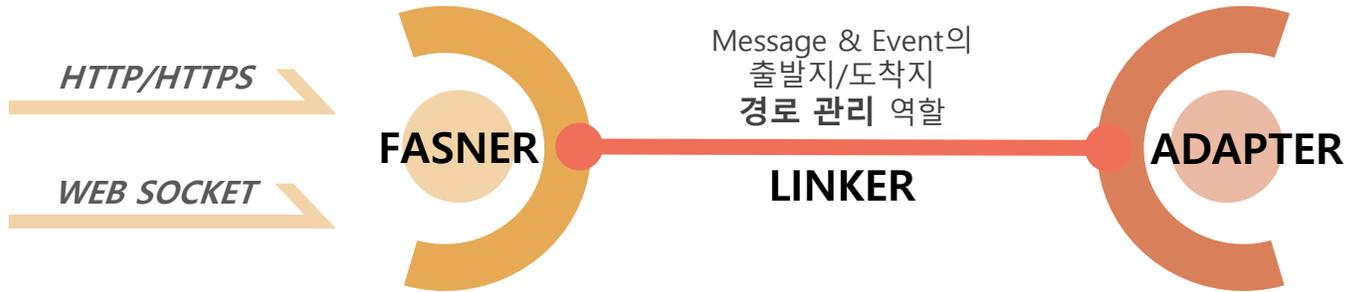


## Messaging Service

Web 표준 인터페이스를 제공하는 Real-Time Streaming 기반의 Message & Event 중계 서비스로 표준 API 를 관리하고 확장성, 추적성 기능을 제공

외부 시스템에서  
요청하는  
Message & Event를  
수신하는 역할

요청 받은  
Message & Event를  
외부 시스템에  
전송하는 역할



### Flexible Msg / Event Adapter (Broker) 연계

- HTTP Client
- REDIS
- KAFKA
- MongoDB
- FTP/FTP's
- IBM MQ

### API 표준 정의서

| API 상세 정의서   | 종류                              | 설명  | 목적     | 비고  |
|--------------|---------------------------------|---|--------|---|
| Interface ID | Interface ID                    | IF_COMMAND (외부명:인터페이스)  |        | IF_MES_OB_INFO_REV_001                              |
| Source ID    | Transaction ID                  | Transaction ID (외부명: System ID)   | 이러의 요청 | MES   |
| Command ID   | Command ID                      | {Source ID, COMMAND} Command에 의해 생성된 Unique 하게 필요하며 필수                                      |        | MES_OB_INFO_REV                                     |
| Request URL  | Transaction ID (외부명: System ID) | Source ID (T) MES, SPC 등 정보<br>FASNA (Source ID) Command ID<br>FASNA (Source ID) Command ID |        | FASNA/MES_OB_INFO_REV<br>FASNA/BA (SOP)             |
| Target ID    | Transaction ID (외부명: System ID) | 이러의 요청  |        | MES   |
| Target URL   | Transaction ID (외부명: System ID) | Target ID (T) MES, SPC 등 정보<br>FASNA (Target ID) Command ID                                 |        | MES, SPC<br>FASNA/MES_OB_INFO_REV<br>FASNA/BA (SOP) |

| Sample API List         | Command ID  | Source Type | Source ID                     | Request URL | Target Type | Target ID             | Target URL |
|-------------------------|-------------|-------------|-------------------------------|-------------|-------------|-----------------------|------------|
| MES_OB_INFO_REV         | reqSender   | MES         | FASNA/MES_OB_INFO_REV         | reqReceiver | SAP         | FASNA/BA (SOP)        |            |
| MES_VERIFY_SLOTMAP_SUCC | reqSender   | MES         | FASNA/MES_VERIFY_SLOTMAP_SUCC | reqReceiver | SAP         | FASNA/BA (SOP)        |            |
| SAP_MWV_RCD             | reqReceiver | SAP         | FASNA/SAP (SOP)               | reqSender   | MES         | FASNA/MES_OB_INFO_REV |            |
| SAP_PANA_NAME_DCOL      | reqReceiver | SAP         | FASNA/SAP (SOP)               | reqSender   | MES         | FASNA/MES_OB_INFO_REV |            |
| SPC_INTERLOCK_ALARM     | reqSender   | SPC         | FASNA/SPC_INTERLOCK_ALARM     | reqReceiver | MES         | FASNA/MES_OB_INFO_REV |            |
| ISL_INTERLOCK_ALARM     | reqSender   | ISL         | FASNA/ISL_INTERLOCK_ALARM     | reqReceiver | MES         | FASNA/MES_OB_INFO_REV |            |
| MES_OB_INFO             | reqSender   | MES         | FASNA/MES_OB_INFO             | reqReceiver | MES         | FASNA/MES_OB_INFO     |            |

| Message Format 정의서 | Depth | 필수여부 | 설명   | 비고   |
|--------------------|-------|------|--|--|
| 1                  | head  | 필수   | Header<br>Target, Command ID, Transaction ID 등 필수 정보 포함  | "head": {<br>"target": "FASNA/BA (SOP)",<br>"command": "IF_MES_OB_INFO_REV",<br>"transaction": "2017-08-14 14:51:13"}<br>} |
| 2                  | body  | 선택   | Body<br>Command ID                                       | "body": {<br>"command": "IF_MES_OB_INFO_REV",<br>"transaction": "2017-08-14 14:51:13"}<br>}                                |
| 1                  | body  | 선택   | Body<br>Message Set 정보 포함 (필수)<br>Transaction ID 포함 (선택) | "body": {<br>"message": "IF_MES_OB_INFO_REV",<br>"transaction": "2017-08-14 14:51:13"}<br>}                                |

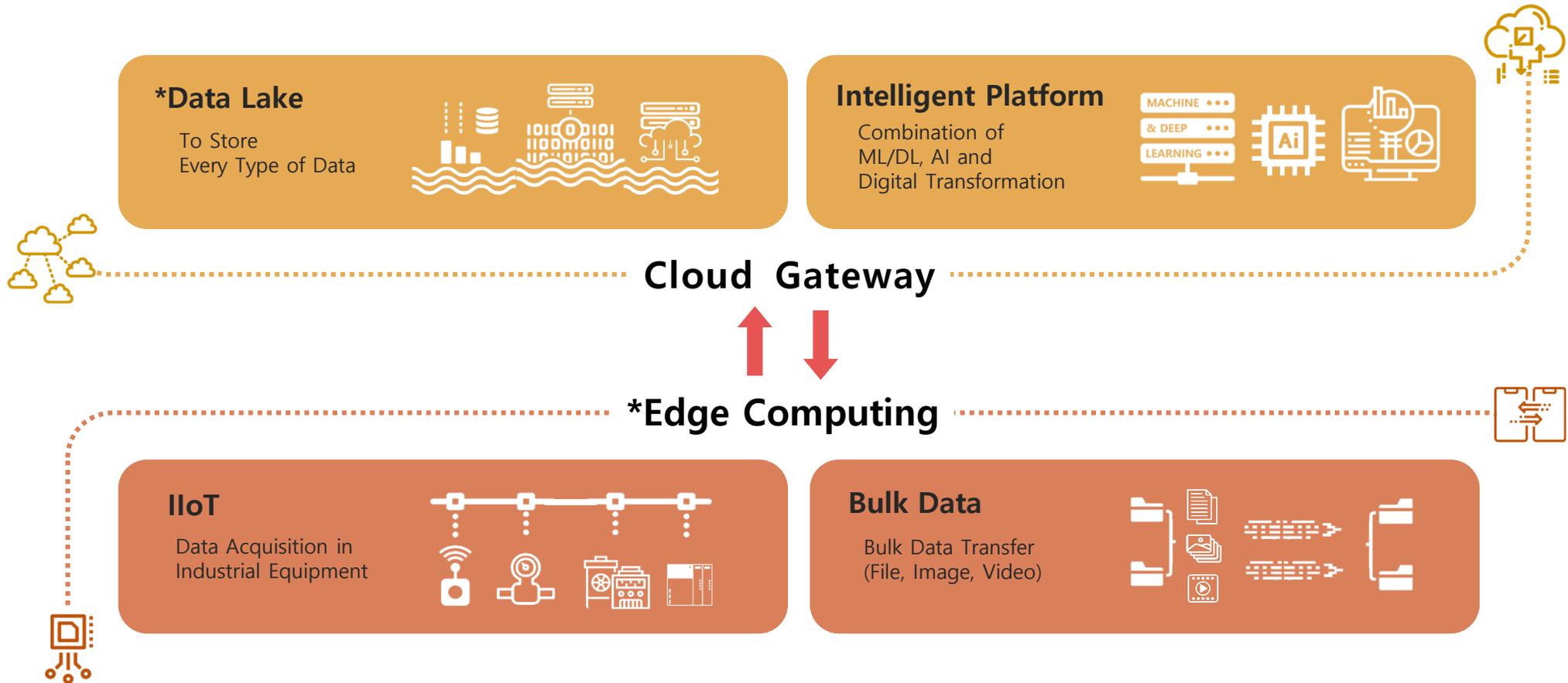
- API 를 Static 하게 등록 관리하여 Transaction의 이력 추적 (Traceability) 기능 제공
- Application ( MES, EES etc.)의 Vendor, Solution 무관하게 표준 API 기준의 공장 횡전개 등 확장성 제공

# Key Features of DiFlow



## Connectivity Service

제조 현장에서 발생하는 대규모 데이터를 처리하고, 이를 분석 인프라(On-Premise, Cloud)와 연계하는 서비스



\*Data Lake: 다양한 형태의 원형(Raw) 데이터들을 모은 저장소의 집합

\*Edge Computing: 산업용 기기 주변(edge)이나 자체에서 데이터를 처리하는 기술

# Key Features of DiFlow



## Function Service

제조 산업 군별 다양한 형식의 Raw Data 수집, 구조/단위/함수 변환 등을 통해 제조向 META 표준화 기능을 제공하는 서비스

### FUNCTION



#### Data Preparation

데이터 정제/변환을 위한 다양한 함수 제공



#### Metadata Management

표준 META 규약 정의 및 관리

#### Real-Time Transaction



#### Dynamic Data Movement

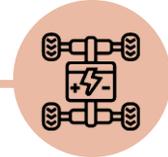
실시간으로 발생하는 소스데이터 수집

#### Standard Specification Non-High Tech Industries

Table 1 Message & Function Detail

| Message                              | Function                          | MF Id         | MF Name       |
|--------------------------------------|-----------------------------------|---------------|---------------|
| M1 : Equipment Status                | F01 : Are You There               | M1F01         | EQ_CONNECT    |
|                                      | F03 : Equipment Status Request    | M1F03         | EQ_STS_REQ    |
|                                      | F04 : Equipment Status Variable   | M1F04         | EQ_STS_VAL    |
|                                      | F09 : RESERVED                    | M1F09         | EQ_STS_REV    |
|                                      | F11 : Status Variable Name List   | M1F11         | EQ_STS_LIST   |
|                                      | F13 : Equipment Communication     | M1F13         | EQ_COM_REQ    |
| M2 : Equipment Control & Diagnostics | F01 : RESERVED                    | M2F01         | EQ_RMT_REV1   |
|                                      | F03 : RESERVED                    | M2F03         | EQ_RMT_REV2   |
|                                      | F13 : Equipment Constance Request | M2F13         | EQ_COS_REQ    |
|                                      | F17 : Date and Time Request       | M2F17         | EQ_DATE_REQ   |
|                                      | F21 : Remote Command Send         | M2F21         | EQ_REMT_CMD   |
|                                      | F23 : Trace Initialize Send       | M2F23         | EQ_TRACE_SND  |
|                                      | F33 : RESERVED                    | M2F33         | EQ_DEF_REPT   |
|                                      | F35 : RESERVED                    | M2F35         | EQ_LINK_EVENT |
|                                      | F37 : RESERVED                    | M2F37         | EQ_EVENT_REPT |
|                                      | F41 : Host Command Send           | M2F41         | EQ_HREMT_CMD  |
| F49 : Host Command Send              | M2F49                             | EQ_IBREMT_CMD |               |
| M3 : Material Status                 | F01 : Material Status Request     | M3F01         | MAT_STS_REQ   |
|                                      | F03 : RESERVED                    | M3F03         | MAT_REV1      |
|                                      | F13 : Material ID Send            | M3F13         | MAT_ID_SEN    |

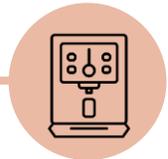
Battery



Semi. Materials



Home Appliance



표준 META 규약을 통해 다양한 제조 산업에서 일관된 Interface 적용 가능

# Key Features of DiFlow



## MALL Service

DiFlow에서 제공하는 Messaging, Connectivity, Function Services의 모니터링, 관리, 로깅, 연동 기능을 제공하는 서비스

### Monitoring

Transaction의 실시간 모니터링 및 리포트 기능

### Admin

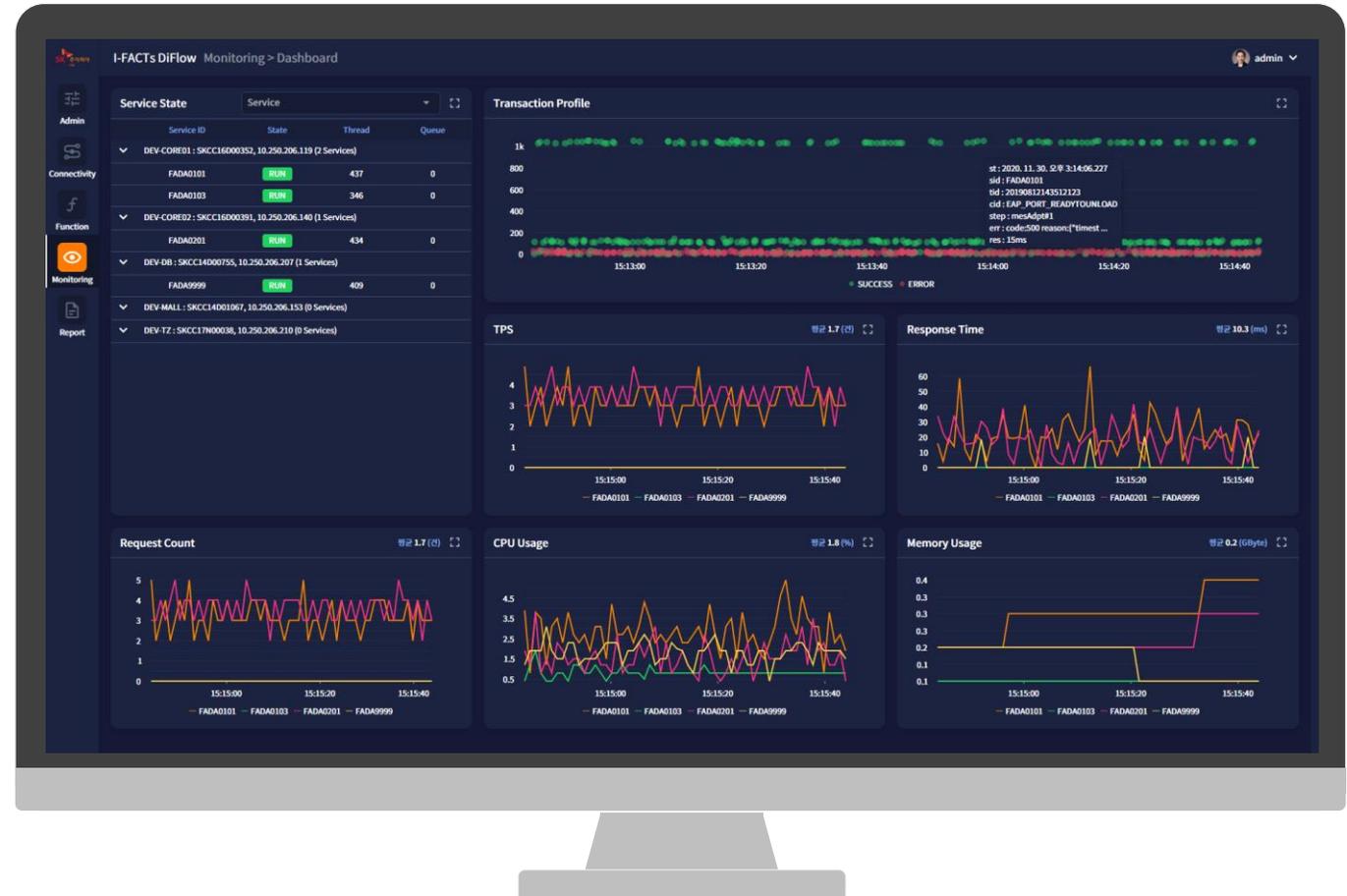
권한 및 기준 정보 관리, Service Control 기능

### Logging

Transaction History 관리 및 Recovery 기능

### Link

Transaction 경로 및 서비스 연동 관리 기능



# Customer Values

DiFlow에서 제공하는 최적의 Technical Service를 통해 개발자, 사용자, 관리자의 업무 효율성 증대 가능



- Standard API
- Open Framework
- Easy to Integration
- Testability



- Flexible I/F
  - Standard, Custom
- Cost Effectiveness
- Comprehensive
  - Sensor, IIoT, App, Cloud



- Stability
  - High-Availability, Recovery
- Audit-MALL
- Performance
  - 30k/s, 1000 EAs, 16Core

# Business Offering

고객사의 요구사항 및 Infra 환경을 고려하여 최적의 솔루션 구축 전략을 제안  
 솔루션 구축을 위한 서비스, 기술 교육, On-site 및 Remote 운영 서비스 등을 제공

|             |                |   |
|-------------|----------------|---|
| License     | 임시 라이선스        | 계약 후 정식 라이선스가 발급되기 전까지 제공되는 한시적 라이선스                        |
|             | 년간 라이선스        | Transaction Volume 기준의 정책에 따른 기간제 라이선스                      |
|             | 영구 라이선스        | Transaction Volume 기준의 정책에 따른 Factory/Site/Campus 등 영구 라이선스 |
| Service     | 구축             | 솔루션 도입에 따른 구축 서비스 제공  |
|             | Customizing    | 고객사 요구사항에 대한 솔루션 Customizing 서비스 제공                         |
| Maintenance | 설치             | 고객사의 Infra 환경에 따른 솔루션 초기 설치 지원                              |
|             | 교육             | 솔루션 사용자 교육, 운영자 교육  |
|             | Patch          | Bug & Upgrade/Rollback 기능 Patch 지원                          |
|             | 정기/비정기 점검      | 시스템 및 솔루션 성능 및 상태 모니터링                                      |
|             | 기술 문의 응대       | 고객사 시스템 Trouble, Q&A 대응                                     |
|             | 24h*365days 운영 | 고객사 현장 및 Cloud 기반 Remote 운영                                 |

# [별첨] 타 솔루션과의 비교

| 항목                   | SK C&C I-FACTs DiFlow   | 'A' Solution  | 'B' Solution                                    |
|----------------------|---|---|---|
| 메시지 전송방식             | Request-Response  | Publish-Subscribe   | Publish-Subscribe                               |
| 메시지 프로토콜             | UDP, TCP 기반의 Application Layer 까지 지원가능 ( HTTP/HTTPS, WebSocket 등)             | UDP, TCP  | TCP   |
| 이중화/고가용성             | 지원  | 지원  | 지원  |
| Transport            | Peer-to-peer, Multicast   | Peer-to-peer, Multicast, Broadcast                              | Peer-to-peer, Multicast, Broadcast              |
| 분산 환경 대응             | HAProxy 사용  | RVD(Rendezvous Daemon) 및 RVDQ(Rendezvous Distributed Queues) 사용 | Apache ZooKeeper 사용                             |
| 메시지 포맷               | 제조사 META 표준 제공<br>(데이터 포맷은 JSON을 표준으로 사용하며, XML, Binary, File 등 지원 가능)        | 고객사 별 사용하는 메시지 포맷이 다름   | 별도의 표준 메시지 포맷은 없으며, JSON, CSV XML 등의 데이터 포맷을 지원 |
| Admin                | Admin UI 기본 제공<br>(설정 관리, 상태 모니터링, Control 기능)                                | Admin console 제공<br>(Daemon 설정 및 상태 관리)                         | 미 지원<br>(별도의 관리 Tool 설치 필요)                     |
| Monitoring           | Transaction Monitoring, Service Dashboard, Topology, Bottleneck Report 등 기본제공 | 미 지원<br>(별도 구매 필요)  | 미 지원<br>(별도의 Monitoring Service 구축 필요)          |
| Transaction Recovery | 지원  | 미 지원  | 미 지원<br>(Data Backup/Recovery 지원)               |
| Data Preparation     | 지원  | 미 지원  | 미 지원  |
| Edge Computing       | 지원  | 미 지원  | 미 지원  |
| Cloud                | 지원  | 지원<br>(지원 기능이 제약적이고, 가격이 매우 비쌘)                                 | 지원  |

# [별첨] FAQ

| Questions |   | Answers  |
|-----------|---|--|
| Q1        | DiFlow의 Transaction 전송 성능은 어느정도 인가요?                            | 서버 사양 16Core 기준, 연동된 어플리케이션이 100개 이하일 경우 최대 <b>30,000TPS</b> (Transactions Per Second) 까지 지원 가능 합니다.   |
| Q2        | Interface 추가 등의 변경 작업 시 Service Down Time을 축소시킬 방법이 있을까요?       | Interface 항목의 경우에는 UI 메뉴에서 Service 재 시작 없이 추가가 가능하며, 내장된 Proxy기능을 통해 Service 이중화를 지원하여 변경 작업 시 Service Down Time은 발생하지 않습니다.                           |
| Q3        | 어플리케이션 간 전달되는 Interface 명세서를 효율적으로 관리할 수 있는 방법이 있을까요?           | API Management 메뉴를 통해 DiFlow에서 송·수신하는 Interface List를 관리하여 타 상용 솔루션 대비 Transaction에 대한 추적이 용이 합니다.   |
| Q4        | Transaction의 증가로 잦은 정체가 발생하는데, 효율을 개선할 수 있는 방안이 있을까요?           | Topology 기능과 Bottleneck Report을 통해 실시간으로 정체가 발생하는 구간을 모니터링 할 수 있으며, 각 서비스 내 Application Adapter 별 성능지표를 확인 하고 개별 Application 별 서버, 서비스 증설에 활용할 수 있습니다. |
| Q5        | 서비스 장애가 발생하여, Transaction 전송 중에 문제가 발생했을 경우, 처리할 수 있는 방안이 있을까요? | 장애가 발생한 Transaction을 재 전송 할 수 있는 Transaction Recovery 기능을 제공하여, UI에서 관리자가 Transaction을 복구할 수 있습니다.   |
|           |   |  |
|           |   |  |

# DiFlow 기능 구성 – Basic

| 대분류        | 중분류                   | 소분류         | 기능  |
|------------|-----------------------|-------------|---|
| Admin      | Common                | -           | 공통코드관리  |
|            | User & Authority      | -           | 시스템 사용자 계정 및 권한 관리  |
|            | Product License       | -           | 제품 라이선스 관리  |
|            | Service Configuration | -           | DiFlow Server 및 Service 정보 관리 (Server 및 Service 정보 등록/수정/삭제)  |
|            | Service Control       | -           | DiFlow Service 상태 Monitoring 및 제어 명령(Restart, Start, Stop, Upgrade) 실행  |
|            | API Management        | -           | DiFlow API 정보 관리 (Linker 별 Command 등록/수정/삭제)  |
|            | Proxy Configuration   | -           | DiFlow Proxy 정보 관리 (Proxy 등록/수정/삭제)   |
|            | Revision History      | -           | DiFlow Service 상태 및 설정 정보 변경 이력 관리  |
| Monitoring | Dashboard             | -           | 실시간 Monitoring Dashboard<br>(Service State, Transaction Profile, TPS, Response Time, Request Count, CPU/Memory 사용량) |
|            | Transaction View      | Profile     | Transaction Profile 확인  |
|            |                       | History     | Transaction History 조회  |
|            | Topology              | -           | DiFlow Service 별 Topology view ( 내/외부 Listener, Host, Adapter, Link 서비스 등 표현 )                                      |
|            | Recovery              | Transaction | Transaction의 Error list 와 해당 Transaction의 Recovery 실행 메뉴  |
|            |                       | History     | Recovery Transaction History 조회   |
| Report     | Summary               | Daily       | 일별 리포트 (피크타임 성능 요약, TPS, Response Time, Request Count, CPU/Memory 사용량)  |
|            |                       | Weekly      | 주간 리포트 (서비스 별 가동률, TPS, Response Time, Request Count, CPU/Memory 사용량)   |
|            |                       | Monthly     | 월간 리포트 (서버 별 가동률, TPS, Response Time, Request Count, CPU/Memory 사용량)  |
|            | Error Info            | -           | 하루 기준 시간/Command 별 오류 통계량   |
|            | Bottleneck            | -           | DiFlow Service 별 Adapter 기준으로 병목 구간 확인  |

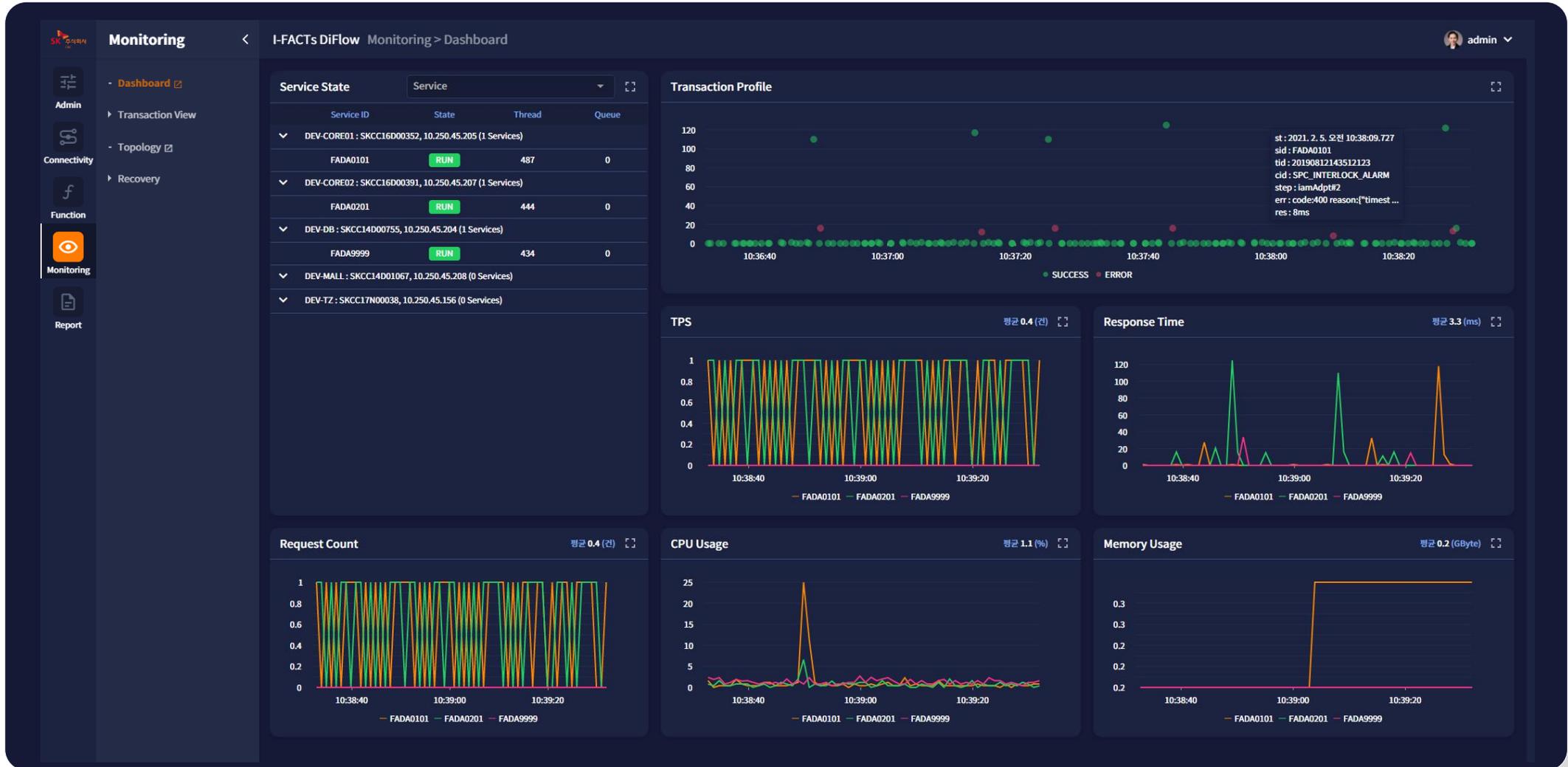
# DiFlow 기능 구성 – Advanced

| 대분류              | 중분류                             | 소분류               | 기능   |
|------------------|---------------------------------|-------------------|--|
| Connectivity     | ThingZ Configuration            | Add Service       | ThingZ App 등록/수정/삭제  |
|                  |                                 | Config Service    | ThingZ Type별 설정 값 수정 및 Tag List 정보와 DCOL Plan 의 표준 Meta의 Link 연결   |
|                  |                                 | DCOL Meta         | Message & Function 으로 표준 모델 정의 ( Meta 표준 정의 )  |
|                  | ThingZ Management               | Tag List          | 전체 정의된 Tag 등록/수정   |
|                  |                                 | Tag Monitoring    | 정의된 Tag 값 실시간 Monitoring   |
|                  |                                 | ThingZ Control    | ThingZ Type별 App 상태 확인 및 제어 명령(Restart, Start, Stop, Update) 실행  |
|                  |                                 | ThingZ Monitoring | ThingZ Type별 CPU/Memory 사용량 및 실시간 BulkData 전송 상태 Monitoring  |
| Revision History | ThingZ Service 상태 및 설정 정보 변경 관리 |                   |  |
| Function         | Function List                   | -                 | Application 간 Interface 연동 시 Raw Data를 가공하기 위한 Function 관리 <ul style="list-style-type: none"> <li>• 특정 Tag 조회 함수</li> <li>• If Else, Case 등 Conditional Branch</li> <li>• Tag별 Bit, Byte 단위의 변환 및 연산               <ul style="list-style-type: none"> <li>- getBitArray(byte 배열을 bit 배열로 변환), getCharFromInt(byte 배열을 문자열로 변환)</li> <li>concatArray(배열 합치기), conversionScale(데이터 scale 변환) 등</li> </ul> </li> <li>• Math 함수               <ul style="list-style-type: none"> <li>- abs(절대값), avg(평균), sum(합계), max/min(최대값/최소값), round(반올림), calc(사칙연산) 등</li> </ul> </li> </ul> |
|                  | Function Group                  | -                 | 여러 개의 Function을 순차적으로 처리하기 위한 Function Group관리   |

# DiFlow 대표 UI (1/4)

## Monitoring > Dashboard

실시간 Monitoring Dashboard (Service State, Transaction Profile, TPS, Response Time, Request Count, CPU/Memory 사용량 확인)



# DiFlow 대표 UI (2/4)

## Monitoring > Topology

DiFlow Service 별 Topology view ( 내/외부 Listener, Host, Adapter, Link 서비스 등 표현 )

The screenshot displays the DiFlow Monitoring interface in a dark theme. The left sidebar contains navigation options: Admin, Connectivity, Function, Monitoring (selected), and Report. The main area is titled 'I-FACTs DiFlow Monitoring > Topology' and shows a 'Topology View' with filters for 'Server ID' (5 items selected) and 'Service ID' (3 items selected). A 'View' button and a refresh rate of '3:38' are also present.

Three service panels are visible:

- DEV-CORE01 > FADA0101**: Shows a central node connected to various adapters like SPC, FDC, MES, MCS, IAM, Redis, and EAP. A transaction popup shows a latency of 115ms.
- DEV-CORE02 > FADA0201**: Shows a central node connected to adapters like EAP, SPC, MES, MCS, IAM, FDC, Redis, and EAP. A transaction popup shows a latency of 114ms.
- DEV-DB > FADA9999**: Shows a central node connected to adapters like MCS, FDC, MES, SPC, IAM, Redis, and EAP.

Each panel includes a 'reset all' button and zoom controls (+, -, square). A 'Transaction' popup in the DEV-CORE02 panel displays the following data:

|        |        |
|--------|--------|
| 평균응답시간 | 114 ms |
| 건수     | 16 건   |
| 에러     | 0 건    |
| TPS    | 0 건    |

# DiFlow 대표 UI (3/4)

## Admin > API Management

DiFlow API 정보 관리 (Linker 별 Command 등록/수정/삭제 기능 및 API 별 Source, Target 정보 확인)

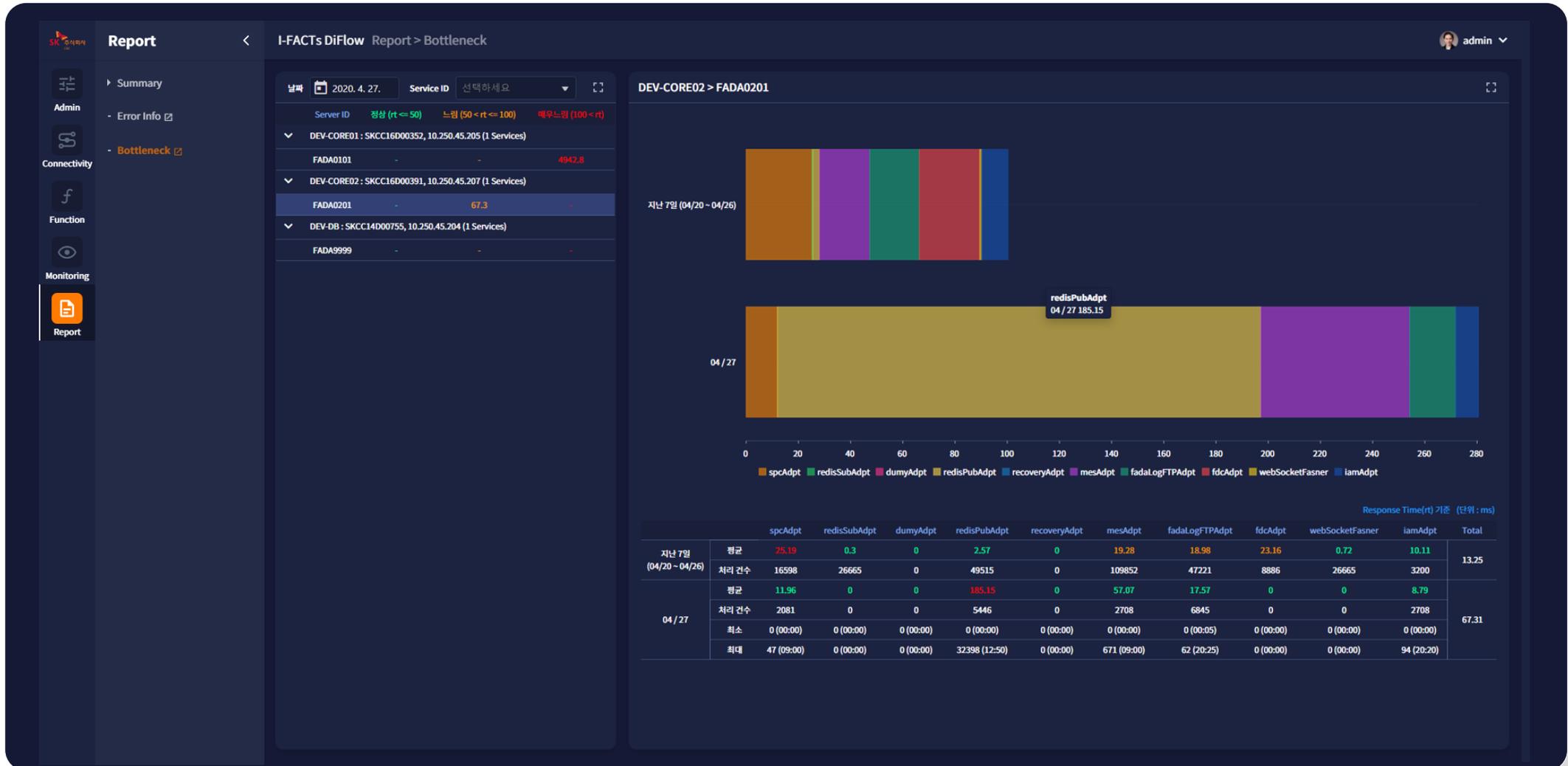
The screenshot displays the 'Admin > API Management' interface. On the left is a navigation sidebar with categories: Admin, Connectivity, Function, Monitoring, and Report. The main content area shows a 'Service List' on the left with a tree view including 'DEV-CORE01', 'DEV-CORE02', 'DEV-DB', 'DEV-MALL', and 'DEV-TZ'. The selected service is 'DEV-CORE01 > FADA0101'. The main table lists API configurations with columns: Link ID, Command ID, Source Type, Source ID, Request URL, Target Type, Target ID, and Target URL. The table contains 25 rows of data.

| Link ID     | Command ID              | Source Type | Source ID | Request URL                       | Target Type | Target ID | Target URL                       |
|-------------|-------------------------|-------------|-----------|-----------------------------------|-------------|-----------|----------------------------------|
| redisPubLnk | MES_JOB_INFO_RESV       | httpServer  | MES       | /FADA/MES/MES_JOB_INFO_RESV       | webSocket   | EAP       | /FADA/EAP/[EQPID]                |
| mestLnk     | EAP_VERIFY_PPID_SUCC    | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_VERIFY_PPID_SUCC   |
| mestLnk     | EAP_PORT_READYToload    | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_PORT_READYToload   |
| mestLnk     | EAP_PORT_ARRIVED        | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_PORT_ARRIVED       |
| mestLnk     | EAP_FOUP_LOAD           | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_FOUP_LOAD          |
| mestLnk     | EAP_VERIFY_SLOTMAP_REQ  | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_VERIFY_SLOTMAP_REQ |
| redisPubLnk | MES_VERIFY_SLOTMAP_SUCC | httpServer  | MES       | /FADA/MES/MES_VERIFY_SLOTMAP_SUCC | webSocket   | EAP       | /FADA/EAP/[EQPID]                |
| mestLnk     | EAP_START_CMD_REQ       | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_START_CMD_REQ      |
| redisPubLnk | MES_START_CMD           | httpServer  | MES       | /FADA/MES/MES_START_CMD           | webSocket   | EAP       | /FADA/EAP/[EQPID]                |
| mestLnk     | EAP_MVIN_REQ            | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_MVIN_REQ           |
| mestLnk     | EAP_PARA_NAME_DCOL      | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_PARA_NAME_DCOL     |
| mestLnk     | EAP_MVOU_REQ            | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_MVOU_REQ           |
| mestLnk     | EAP_PORT_READYTOUNLOAD  | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_PORT_READYTOUNLOAD |
| mestLnk     | EAP_FOUP_UNLOAD         | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_FOUP_UNLOAD        |
| mestLnk     | EAP_PORT_REMOVED        | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | MES       | /FADA/MES/EAP_PORT_REMOVED       |
| splLnk      | MES_DCOL                | httpServer  | MES       | /FADA/MES/MES_DCOL                | httpServer  | SPC       | /FADA/SPC/MES_DCOL               |
| almLnk      | SPC_INTERLOCK_ALARM     | httpServer  | SPC       | /FADA/SPC/SPC_INTERLOCK_ALARM     | httpServer  | MES       | /FADA/MES/SPC_INTERLOCK_ALARM    |
| almLnk      | SPC_INTERLOCK_ALARM     | httpServer  | SPC       | /FADA/SPC/SPC_INTERLOCK_ALARM     | httpServer  | IAM       | /FADA/IAM/SPC_INTERLOCK_ALARM    |
| almLnk      | FDC_INTERLOCK_ALARM     | httpServer  | FDC       | /FADA/FDC/FDC_INTERLOCK_ALARM     | httpServer  | MES       | /FADA/MES/FDC_INTERLOCK_ALARM    |
| almLnk      | FDC_INTERLOCK_ALARM     | httpServer  | FDC       | /FADA/FDC/FDC_INTERLOCK_ALARM     | httpServer  | IAM       | /FADA/IAM/FDC_INTERLOCK_ALARM    |
| fdcLnk      | EAP_TRACE_SENDING       | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | FDC       | /FADA/FDC/EAP_TRACE_SENDING      |
| fdcLnk      | EAP_EVENT_SENDING       | webSocket   | EAP       | /FADA/EAP/[EQPID]                 | httpServer  | FDC       | /FADA/FDC/EAP_EVENT_SENDING      |
| mestLnk     | MCS_JOB_REQ             | httpServer  | MCS       | /FADA/MCS/MCS_JOB_REQ             | httpServer  | MES       | /FADA/MES/MCS_JOB_REQ            |
| redisPubLnk | FDC_TRACE_REQ           | httpServer  | FDC       | /FADA/FDC/FDC_TRACE_REQ           | webSocket   | EAP       | /FADA/EAP/[EQPID]                |
| redisPubLnk | FDC_EVENT_REQ           | httpServer  | FDC       | /FADA/FDC/FDC_EVENT_REQ           | webSocket   | EAP       | /FADA/EAP/[EQPID]                |

# DiFlow 대표 UI (4/4)

## Report > Bottleneck

DiFlow Service 간 평균 응답시간 값 비교 및 Service 내 Adapter 기준으로 병목 구간 확인



Data Interactive Flow 1.5



# CONTACT

SK(주) C&C

제조Solution Digital그룹

박정영 매니저

[jungyoung02@sk.com](mailto:jungyoung02@sk.com)

+82-10-4626-7653

**Thank You!**