

(붙임)

RTGS 프로토타입 시스템 PoC 테스트 3차 결과 보고

2025. 7

IT전략국 금융IT부 RTGS시스템팀

차 례

[요 약]

I. 개요	1
II. PoC 테스트 수행 내용	3
III. PoC 테스트 결과	12
IV. 주요 성과 및 향후 계획	15
<별첨1> PoC 테스트 용어정리	17
<별첨2> PoC 테스트를 위한 IT인프라 구성방안	21
<별첨3> PoC 테스트를 위한 프로세스	23

〈요 약〉

▶ RTGS시스템팀은 신속자금이체를 24×365 연중무휴 안정적으로 운영하기 위해 **다중 센터의 Active-Active 시스템** 아키텍처를 설계하여, 성능·안정·확장성 측면에서 실현 가능성을 검증하기 위한 **PoC 테스트***를 진행 후, **신속한 처리와 센터 장애시에도 정상 작동하는 것**을 확인

* **【PoC(Proof of Concept)】** 아이디어나 기술이 실제로 구현이 가능한지, 그리고 실제 상황에서도 작동할 수 있는지를 간단한 형태로 검증하는 테스트

<테스트 결과>

- **성능** : 초당 **15,000건** 이상의 접수처리, 건당 처리시간 평균 **0.014초** 이내 확인
- **안정** : 1개 센터 장애시 나머지 1개 센터의 서비스 지속 및 장애센터의 복원력 확인
- **확장** : ISO20022 전문처리를 **API 서비스**로 구현

⇒ PoC 테스트 결과를 바탕으로 **프로토타입 아키텍처**를 개선해서 **2025년 하반기 PoC 테스트와 2026~2027년 BMT 사업**을 진행할 예정

I

개요

□ IT전략국은 실시간총액결제 방식의 신속자금이체를 24×365 연중무휴로 제공하는 **RTGS시스템 구축을 계획**

○ 국내외 사례조사를 바탕으로 RTGS 핵심 아키텍처를 설계하고 프로토타입 시스템을 개발하여 2개 센터 간 **Active-Active 운영에 대한 기본 기능·성능에 대한 1~2차 검증을 완료**(2023.8월 ~ 2025.1월)

⇒ **프로토타입 시스템의 최고 성능 및 1개 센터의 장애시 복원력 확인**을 위한 **PoC 테스트 3차 사업 실시**(2025.2월 ~ 2025.6월)

II

PoC 테스트 수행 내용

□ (PoC 대상) 솔루션 조합별 3개 테스트 케이스로 구분하여 성능테스트와 복원력테스트를 진행하며 **각 기술 요소의 적정성을 평가**

테스트 케이스별 구성 내역

동기화 방식		인메모리DB	원장DB
①	이벤트 브로커 (Confluent)	Hazelcast	MongoDB
		Goldilocks	Goldilocks
②	이벤트 브로커 (Solace)	Hazelcast	MongoDB
		Goldilocks	Goldilocks
③	인메모리 클러스터링 (Hazelcast)	Hazelcast	MongoDB

- (수행 방법) PoC는 부하 테스트 도구를 활용해서 RTGS 핵심 기능에 대하여 「기능검증 → 성능검증 → 복원력검증」 단계별 순서로 수행
 - 기능검증은 신속자금이체 처리 테스트 후 프로토타입의 핵심 서비스들이 정상 작동되었는지 파악
 - 성능검증은 초당 최대 몇 건의 거래요청을 접수와 평균 End-to-End 처리시간(최초 접수부터 처리완료되는 시간)을 측정
 - 복원력검증은 한 센터의 모든 서버가 정지했을 때 다른 센터에서의 정상 작동과 함께 장애를 복원했을 때 양 센터의 정상 작동 및 데이터 정합성 여부를 확인

III

PoC 테스트 결과

- (기능테스트) RTGS 프로토타입 시스템의 핵심 서비스가 센터간 동시운영 (Active-Active)체제에서 이체처리기능을 정상적으로 수행
- (성능테스트) 모든 테스트 케이스가 초당 15,000건 이상의 요청 접수, 건당 처리시간 평균 0.014초 이내로 확인
- (복원력테스트) 1개 센터의 장애 시에도 나머지 센터에서 서비스를 유지하고 장애된 센터 복구 후 양 센터의 데이터 정합성을 확인*

* 단, Confluent는 센터 수 제약으로 인한 부분적인 성공

IV

주요 성과 및 향후 계획

- (주요 성과) PoC 테스트 3차 사업을 통해 RTGS 아키텍처 설계 개선, 실현가능성 검증, 향후 프로젝트에 대한 상세 요건 및 시사점 도출 등의 성과를 얻음
- (향후 계획) 현재 RTGS 프로토타입 시스템의 프로그램 및 구성 환경을 일부 개선하여 2025년 하반기 PoC 및 테스트를 추가 실시하고 동 결과를 바탕으로 2026 ~ 2027년 BMT* 사업을 추진할 예정

* BMT는 벤치마크 테스트(BenchMark Test)로, 하드웨어 인프라에 소요되는 성능이 어느정도인지 솔루션을 비교평가하기 위해 수행하는 테스트

I

개요

- IT전략국은 금융결제국과 함께 실시간충액결제 방식의 신속자금이체를 24×365 연중무휴로 제공하는 **RTGS시스템 구축을 계획***

* 「RTGS 방식 신속자금이체시스템 구축 종합계획」(결제정책팀-1169, 2023.12.28, 부총재보)
 「소액RTGS시스템 구축을 위한 IT부문 기본계획」(결제시스템팀-3, 2024.1.2, 부총재보)
 「소액 RTGS 사업 주요 추진 경과 및 향후 계획」(결제정책팀-479, 2024.6.10, 부총재보)
 「RTGS시스템팀 주요 추진 내용 및 향후 계획」(RTGS시스템팀-35, 2025.3.6, 부총재보)

- **국내외 조사 사례를 바탕으로** 향후 RTGS시스템 구축 시 필요한 **IT요소기술**과 **당행 환경**에 적합한 **아키텍처를 설계***

* 「소액RTGS시스템 구축을 위한 기본 아키텍처」(RTGS시스템팀-62, 2024.5.20, 부장)
 「소액RTGS시스템 구축 관련 기술검증 방안」(RTGS시스템팀-69, 2024.5.31, 부장)
 「소액RTGS시스템 기술검증을 위한 프로토타입시스템 구축계획」(RTGS시스템팀-78, 2024.7.1, 부장)

- 이를 토대로 RTGS의 핵심 프로세스에 대한 **프로토타입 시스템을 개발하여 센터 간 Active-Active 운영**에 대한 기본 기능·성능에 대한 **1~2차 검증**을 완료*(2023.8월~2025.1월)

* 「소액RTGS시스템 기술검증을 위한 프로토타입 핵심프로세스 개발보고」(RTGS시스템팀-99, 2024.8.23, 부장)
 「소액RTGS시스템 프로토타입 1차 테스트 완료 보고」(RTGS시스템팀-9, 2025.1.24, 부장)

- 다만, 내년 실시 예정인 BMT를 앞두고 **초당 최대 접수 처리건수 확인, 건당 평균 처리시간 및 센터 장애 시 복원력** 등에 대해서도 추가 검증 필요

⇒ **PoC 테스트 3차 사업***을 통해 **2개 센터 Active-Active 아키텍처 설계 개선**, 핵심 기능에 대한 프로토타입 시스템의 성능·안정 및 확장성 측면에서 **실현 가능성을 검증**, 향후 프로젝트에 대한 상세 요건 및 **시사점 도출**할 필요

* 「사업결의(RTGS 프로토타입 시스템에 대한 PoC테스트)」(RTGS시스템팀-85, 2025.4.30., 팀장)

「RTGS시스템 프로토타입」 PoC 테스트 추진

구분	2023												2024												2025											
	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6													
	분석/설계												개발												PoC 테스트											
IT요소기술 분석																																				
국내외 사례조사																																				
핵심 아키텍처 설계																																				
프로토타입 시스템 개발																																				
프로토타입 PoC 테스트(1~2차)																																				
프로토타입 PoC 테스트(3차)																																				

<참고 1>

RTGS시스템 구축 추진 경과

□ 당행은 전국민이 사용하는 실시간총액결제(RTGS, Real-Time Gross Settlement) 방식의 신속자금 이체를 24×365 연중무휴로 제공하는 RTGS시스템 구축을 추진중

○ 금융결제국은 RTGS 방식 신속자금이체시스템 구축 계획*에 따라 신속자금이체 업무에 대한 비즈니스 프로세스를 도출하고, 참가정책 및 시스템 운영모형을 수립

* 「RTGS 방식 신속자금이체시스템 구축 종합계획」(결제정책팀-1169, 2023. 12. 28), 「소액 RTGS 사업 주요 추진 경과 및 향후 계획」(결제정책팀-479, 2024. 6. 10)

○ IT전략국은 요소 기술에 대한 조사를 통해 시스템 아키텍처를 설계하고 구현 타당성 검토 등 IT 부문 구축계획*을 수립하고 시스템 구축사업을 추진

* 「소액RTGS시스템 구축을 위한 IT부문 기본계획」(결제시스템팀-3, 2024. 1. 2), 「RTGS시스템팀 주요 추진 내용 및 향후 계획」(RTGS시스템팀-35, 2025. 3. 6)

RTGS시스템 구축 관련 주요 사업 추진 경과

사업주체 추진시기	금융결제국 (지급결제혁신팀)	IT전략국 (RTGS시스템팀)
2022	- 구축방안 및 향후 추진계획 수립(1월) - 컨설팅 사업 추진(수행사 PWC社, 7~12월) - 참가기관 협의회(8~9월) - BOK-KFTC 실무T/F(11월~2023.3월)	한은금융망 현황 분석결과 검토
2023	- 컨설팅 사업 결과보고(3월) - 운영모델, 기대효과 및 해결과제 검토(7월) - 시스템 종합계획 수립(12월)	- 컨설팅 결과에 대한 IT부문 정리(6월) - 운영모델 및 IT거버넌스 방안 검토(7월) - 전담 반 신설(8월) - IT기술 조사 연구(8~12월) - 국내 사례조사 결과 보고(10개 기관, 12월)
2024	- 주요국 RTGS IT시스템 다중화 현황(2월) - 주요 추진 경과 및 향후 계획(6월)	- 시스템 구축을 위한 IT부문 기본계획(1월) - 주요 중앙은행 사례조사 결과(4월) - IT기술 조사 연구(4~12월) - RTGS시스템 기본 아키텍처(5월) - RTGS시스템 기술검증 방안(5월) - 프로토타입시스템 구축 계획(7월) - 프로토타입 핵심프로세스 개발 보고(8월) - RTGS 프로토타입 1차 테스트 완료(8월)
2025	미래전략 수립(2024.11월 ~ 2025.7월)	- RTGS 프로토타입 2차 테스트 완료(1월) - RTGS시스템팀 주요 추진내용 및 향후계획(3월) - IT기술 조사 연구(3~4월) - RTGS 프로토타입 3차 PoC 테스트(4~6월) - RTGS 프로토타입 4차 PoC 테스트(2025.하반기)

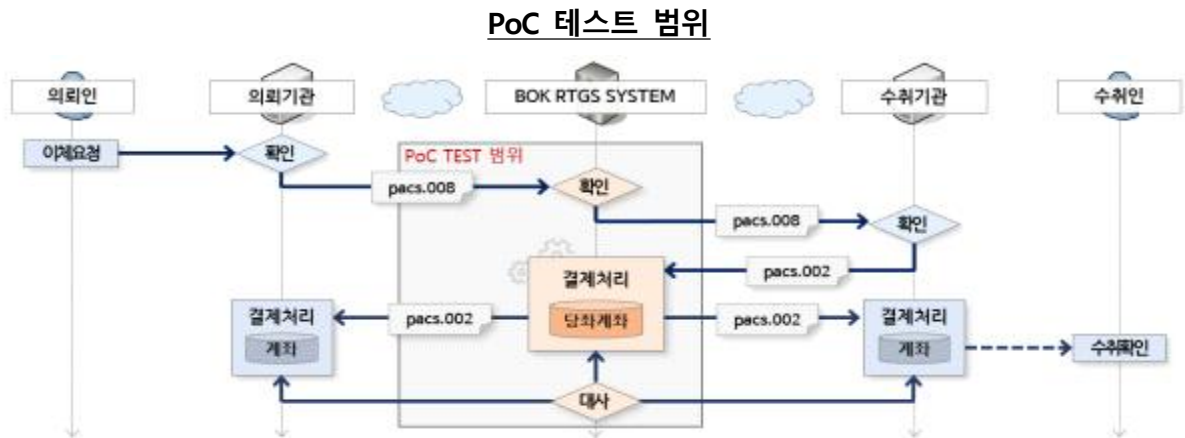
II

PoC 테스트 수행 내용

1. 테스트 대상

□ RTGS 프로토타입 시스템을 공공 클라우드의 2개 IT센터에 동시 운영 (Active-Active) 체제로 구성하여 기본 거래를 테스트함으로써 동 아키텍처의 실제 구현 가능성 및 기술 요소의 적정성을 평가

○ 지급결제 거래과정 중 핵심 영역인 전문송수신, 결제처리, 센터간 동기화를 대상으로 본 PoC 테스트는 ISO20022 전문을 생성하여 당좌계좌에서 결제 처리하는 과정을 검증



□ 2개의 IT센터(통신회선 거리 300km 거리의 수도권과 남부권)에서 동시 처리 성능을 측정하고, 양 센터 데이터의 정합성 확인 및 장애 발생 시 복원력 등을 테스트

프로토타입 시스템 PoC 테스트 목표 및 검증방법

검증내역	목표수준	검증방법
성능	초당 최대 접수 처리건수(TPS*) 10,000건 이상	10초마다 부하를 증가*하며 초당 최대 접수 처리건수 확인 * 초당 5000건에서 20000건까지 부하 증가
	평균 End-to-End(최초접수된 시간부터 최종완료된 시간) 처리시간 1*초 이내 * 전자금융공동망의 처리시간이 1~2초가 소요됨에 따라, 핵심기능에 대한 처리는 1초 이내 필요	평균 End-to-End 처리 성능* 확인 * 평균 end-to-end 시간 = (인메모리DB의 마지막 수정시간 - 테스트 시작시간)/총 처리건수
복원력	장애 발생 후에도 데이터 유실없이 2개 센터의 처리내용 일치	1개 센터에 장애발생*(약 5분) 이후 나머지 센터 정상작동 및 복구 후 데이터 확인 * 장애 발생은 서버 중지 및 프로세스 종료로 실행

* TPS는 1초당 처리건수로 상세한 내용은 (별첨1)의 PoC 테스트 용어자료의 6번 참조

□ 기술검증은 **솔루션 조합별 3개의 테스트 케이스**로 구분하여 센터간 동기화, 인메모리DB, 원장DB 등 **아키텍처 기술 요소별 적합성을 검토**

- **기술검증 결과**는 향후 RTGS시스템 구축 사업의 세부 요구사항에 반영하여 **상세 아키텍처 설계, BMT 등 시스템 구축 추진에 활용**할 예정

테스트 케이스별 구성 내역

동기화 방식		인메모리DB	원장DB	개요
①	이벤트 브로커 (Confluent)	Hazelcast	MongoDB	당행에서 실시한 1~2차 PoC 테스트 내용에서 이벤트 브로커의 센터 간 클러스터링 기능 추가
		Goldilocks	Goldilocks	2개 센터 동기화가 가능한 이벤트 브로커와 인메모리DB와 원장DB를 통합하여 구성
②	이벤트 브로커 (Solace)	Hazelcast	MongoDB	테스트 케이스 ①번에서 이벤트 브로커만
		Goldilocks	Goldilocks	Solace로 교체하여 2개 센터로 동기화 구성
③	인메모리 클러스터링 (Hazelcast)	Hazelcast	MongoDB	센터 간 양방향 동기화가 가능한 통합 솔루션으로 프로토타입 시스템을 일부 수정하여 구성

<참고 2>

IT기술 및 관련 솔루션 검토 내용

- 컨설팅 결과 검토 및 시사점 도출
 - 소액RTGS시스템 컨설팅결과에 대한 IT부문 정리(결제시스템팀-617, 2023.6.21, 부장)
 - 시스템 고가용성 보장을 위한 기본개념 정리(결제시스템팀-1053, 2023.10.12, 부장)
 - 자문실시 결과보고(IT기획팀-218, 2024.5.29)
- 기존의 DBMS 혁신을 위한 방안 검토
 - 소액RTGS 구축 관련 오라클과의 업무협의 결과(결제시스템팀-850, 2023.8.18)
 - NoSQL의 기본개념 정리(RTGS시스템팀-51, 2024.5.7, 부장)
 - 인메모리DG 솔루션(Hazelcast) 조사보고(결제시스템팀-1022, 2023.10.6)
 - 인메모리DG 솔루션(Hazelcast) 테스트 결과(RTGS시스템팀-52, 2024.5.7)
 - 인메모리DB 솔루션(Goldilocks) 조사보고(결제시스템팀-1274, 2023.12.1)
 - 인메모리DB 솔루션(Couchbase) 조사보고(결제시스템팀-1293, 2023.12.6)
 - 인메모리 컴퓨팅 기본개념 정리(RTGS시스템팀-90, 2024.7.24., 부장)
 - Saga 패턴과 CQRS 패턴의 개황 및 시사점(RTGS시스템팀-45, 2025.3.17)
 - NoSQL(MongoDB) 주요 내용 및 시사점(RTGS시스템팀-52, 2025.3.31., 부장)
- 센터 간 연계방식을 데이터 복제가 아닌 클러스터링을 위한 기술 검토
 - 이벤트 브로커 솔루션(Solace) 조사보고(결제시스템팀-1009, 2023.9.27)
 - 메시지 브로커 솔루션(Kafka) 조사보고(결제시스템팀-1205, 2023.11.17.)
 - 메시지 브로커와 이벤트 브로커의 개황 및 시사점(RTGS시스템팀-57, 2025.4.7)
 - 이벤트 브로커(컨플루언트) 주요 내용 및 시사점(RTGS시스템팀-60, 2025.4.9., 부장)
- 참가기관 간의 인터페이스 개선에 필요한 서비스(API, 인증) 검토
 - API에 대한 IT개념 정리(RTGS시스템팀-47, 2024.4.29, 부장)
 - RTGS시스템 프로토타입 개발을 위한 기술스택(JWT)(RTGS시스템팀-158, 2024.12.17, 부장)

2. 테스트 경과

- 개발 및 테스트 환경 설정을 포함하여 4개월간의 준비과정을 거쳐 여러 솔루션 업체들과 공공 클라우드에서 PoC 테스트 환경을 구성한 후, 1개월간의 테스트를 통해 계획대로 검증을 완료
 - (1월) 개발(고사양) PC 3대와 클라우드 환경을 활용하여 RTGS 프로토타입 시스템을 자체 개발하여 기본적인 기능을 검증
 - (2 ~ 3월) RTGS 프로토타입 시스템에 구성된 이벤트 브로커, 인메모리DB, 원장DB들을 테스트 케이스별로 개발
 - (4월) PoC 테스트 사업을 실시하기 위한 테스트 요구사항을 정의하고 사업 결의 및 계약 체결
 - (5월) 각 솔루션 별로 RTGS 프로토타입 시스템을 개선해 나가며 성능 측정 및 복원력테스트를 실시

프로토타입 시스템 PoC 테스트 추진일정

작업 내역	2025					
	1월	2월	3월	4월	5월	6월
프로토타입 시스템 PoC 테스트(2차)						
프로토타입 시스템 보완						
PoC 테스트(3차) 요구사항 정의						
PoC 테스트(3차) 사업결의 및 계약체결						
PoC 테스트 사업 실시*						
① Confluent+Hazelcast+MongoDB Confluent+Goldilocks						
② Solace+Hazelcast+MongoDB Solace+Goldilocks						
③ Hazelcast+MongoDB						
PoC 테스트 결과 정리						

* 5월 해당 사업기간 중 각 솔루션별로 비상주로 작업하여 개발한 설정파일을 당행 직원이 내부에서 직접 적용 후 테스트 실시(총 사업비(일반업무비) ,₩23,201,000)

3. 테스트 환경

- PoC 테스트 검증을 위한 서버는 클라우드 서비스 보안인증(CSAP)을 획득한 네이버 공공 클라우드의 수도권과 남부권 2개의 리전(region)*에 IT센터를 각각 동일하게 구성

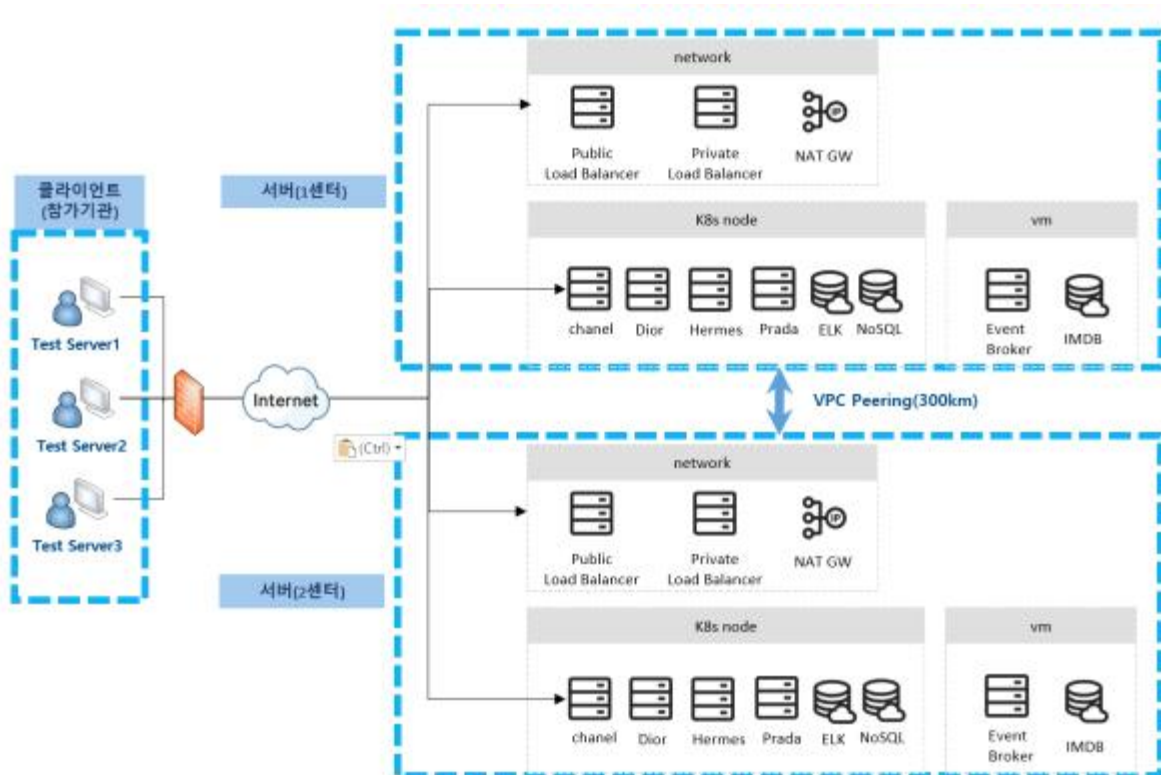
* 리전은 데이터 센터의 지리적 위치개념으로 상세한 내용은 (별첨1)의 PoC 테스트 용어자료의 8번 참조

- 서버내 시스템은 논리적으로 격리된 가상컴퓨팅(VPC) 환경에 비공개 서브넷으로 구성하며, 쿠버네티스* 기반 환경과 가상서버** 기반 환경을 분리하여 구성

* 쿠버네티스는 컨테이너 오케스트레이션 도구로, 상세 내용은 (별첨1)의 PoC 테스트 용어자료의 1번 참조

** 가상서버는 물리 서버 위에 가상머신을 설치하는 것으로, 상세 내용은 (별첨1)의 PoC 테스트 용어자료의 2번 참조

PoC 테스트 검증 시스템 구축 환경



- 참가기관의 역할을 수행하는 클라이언트 부분은 당행의 본부 사무실 내 개발PC에서 부하 발생기를 통해 ISO20022의 pacs.008 전문으로 자금 이체를 요청하고 응답으로 pacs.002 전문을 수신

4. 테스트 수행 방법

- (테스트 부하 발생) PoC의 모든 테스트는 **부하테스트 도구인 K6***를 활용하여 네이버 공공 클라우드의 2개(수도권, 남부권) 센터 중에 1개를 교대로 지정하며 **테스트를 실행****

* k6는 부하를 발생시키는 오픈소스 테스트 도구로 상세 내용은 (별첨1)의 PoC 테스트 용어자료의 3번 참조
** 1개의 센터에 전면 장애가 발생한 경우는 다른 센터로 우회하여 접속하고, 해당 센터는 장애 복구 이후 장애 중 미처리된 건부터 처리한 이후에 신규 접수된 건을 처리

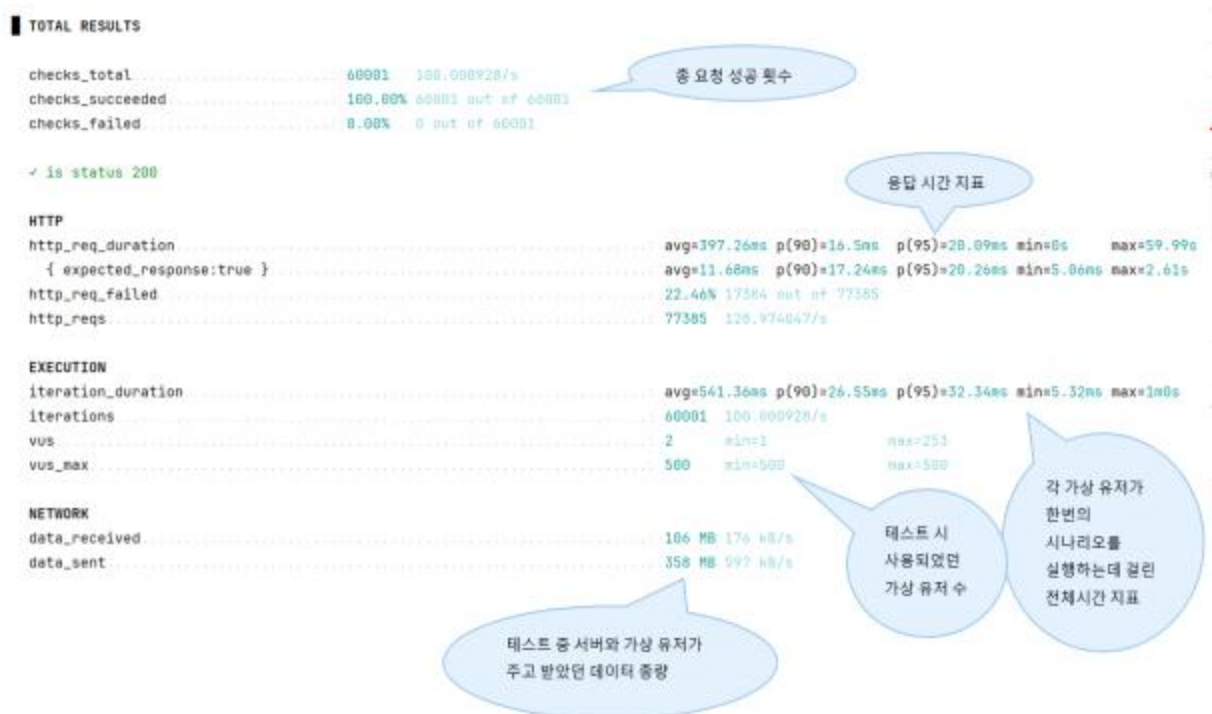
- 프로토타입 시스템에 테스트용으로 등록한 19개 참가기관 중 2개(송신, 수신) 기관을 무작위로 선정

- ISO20022 전문(pacs.008)을 실시간으로 생성(이체 금액은 1~200원 사이)

- 테스트 전문 접수처리를 실행*하여 참가기관 당좌계좌의 잔액을 수정하고 이체처리 결과를 응답전문으로 생성

* 테스트 실시는 본부 내 사무실의 개발PC에서 성능부하 테스트 오픈소스인 K6를 이용하여 네이버 공공 클라우드의 2개 센터내 서버로 전문을 송부(예시 : 성능테스트 초당 10,000~20,000개의 처리를 10~60초간 실시, 복원력 테스트는 10분간 실시를 반복)

부하테스트 결과 화면



□ (테스트 결과 확인) RTGS 프로토타입 시스템에 부하를 발생시킨 결과를
모니터링 도구를 활용해 확인

- o k6에서 부하를 발생시키며 생성된 데이터는 로컬 PC의 InfluxDB*에 저장
하였으며, 초당 최대 접수 처리건수(TPS)는 Grafana** 대시보드로 확인

* InfluxDB는 시계열 분석DB로 상세 내용은 (별첨1)의 PoC 테스트 용어자료의 5번 참조

** Grafana는 모니터링 도구로 상세 내용은 (별첨1)의 PoC 테스트 용어자료의 4번 참조

부하테스트 확인 대시보드*



- o 인메모리DB와 원장DB에 저장된 데이터는 각 DB의 모니터링 도구를
이용하여 최종 처리된 시간과 각 센터 DB의 계좌 데이터 일치 여부를
확인

5. 테스트 수행 절차

- PoC는 **부하테스트 도구를 활용**해 「**기능검증** → **성능검증** → **복원력검증**」 단계별 순서로 RTGS 핵심 기능을 점검

단계별 PoC 수행 방법

구분	1단계 「기능검증」	2단계 「성능검증」	3단계 「복원력검증」
검증 내용	- 서비스 기능 - 전문변환 기능	- 초당 최대 접수 처리건수 - 평균 end-to-end 처리시간	1개 센터 장애시 정상 작동 - 장애 복원 후 정상 작동
검증 대상	프로토타입 시스템	테스트 케이스 별로 성능 및 복원력 검증	

- (기능검증) 참가기관으로부터 **대량**의 **자금이체 전문**(ISO20022 형식)을 접수받아 당행이 보유하고 있는 참가기관의 당좌계좌에 **순차적**으로 **결제처리**를 수행하는 RTGS 시스템 **핵심 기능이 정상 작동하는지 파악**

- **전문송수신 서비스** : ① ISO20022 전문에서 핵심 전문을 추출·생성, ② 인메모리DB에 핵심 전문을 접수(RCVD)하여 유효성 체크 후 전문상태값을 승인상태(ACTC)로 수정, ③핵심 전문을 이벤트 브로커의 접수 토픽에 등록
- **접수동기화 서비스** : ④ 접수 토픽에 등록된 핵심 전문을 입수*, ⑤ 접수 토픽에서 입수한 전문은 모두 대기상태(PDNG)로 원장DB에 저장
 - * 1센터, 2센터에 각각 수신된 전문이 양 센터에 모두 공유되어 동기화됨
- **결제처리 서비스** : ⑥ 접수 토픽에 등록된 핵심 전문을 입수, ⑦ 입수된 핵심 전문을 대기상태(PDNG)로 저장하고 계좌 데이터의 잔액 정보를 수정한 후 동 전문의 상태값을 예약상태(ACSP)로 수정, ⑧핵심 전문을 이벤트 브로커의 결과 토픽에 등록
- **결과동기화 서비스** : ⑨ 결과 토픽에 등록된 핵심 전문을 입수, ⑩ 원장 DB의 계좌 데이터의 잔액 정보를 수정하고 처리상태값을 입금처리완료(ACCC)로 수정, ⑪ 인메모리DB의 핵심 전문의 처리상태값을 입금처리완료(ACCC)로 수정

테스트 확인 방법

테스트 대상	테스트 방법
핵심 전문 처리상태값	모든 데이터 처리상태값이 의뢰기관의 요청을 최종 수취기관의 응답완료인 ACCC(입금처리완료)로 양 센터의 인메모리DB, 원장DB에 저장여부 확인
양 센터의 계좌 데이터	기관별 당좌계좌 데이터가 양 센터의 인메모리DB, 원장DB에 모두 일치하는지 확인

<참고 3>

서비스별 세부 검증

□ 주요 서비스별로 연결 솔루션과 저장되는 데이터가 다르고, 분산 아키텍처인 만큼 모든 서비스가 정상 수행을 해야함

- 서비스의 진행 순서는「전문송수신 → 접수동기화, 결제처리 → 결과동기화」단계별 순서로 수행

주요 서비스별 연결 솔루션

구분	전문송수신	접수동기화	결제처리	결과동기화
서비스	Chanel	Dior	Hermes	Prada
연결 솔루션	확인 • K6 ↓	• 이벤트 브로커 ↓	• 이벤트 브로커 • 인메모리DB ↓	• 이벤트 브로커 • 원장DB ↓
	저장 • 인메모리DB • 이벤트 브로커	• 인메모리DB • 원장DB	• 이벤트 브로커 • 인메모리DB	• 이벤트 브로커 • 원장DB • 인메모리DB

- 서비스 진행 완료 후, 연결된 솔루션들의 데이터가 최종 데이터 상태값을 가지고 있으면 정상 수행

서비스별 연결 솔루션의 최종 데이터 상태

구분	전문송수신	접수동기화	결제처리	결과동기화
이벤트 브로커	접수 토픽 등록*	입수*	입수	-
	결과 토픽 -	-	등록	입수
인메모리DB	참가기관 계좌 -	-	결제 후 수정	-
	핵심 전문 상태값 승인(ACTC)	-	예약(ACSP)	입금처리완료 (ACCC)
원장DB	참가기관 계좌 -	-	-	결제 후 수정
	핵심 전문 상태값 -	대기(PDNG)	-	입금처리완료 (ACCC)

* 등록 : 이벤트 브로커의 토픽에 데이터를 저장

* 입수 : 이벤트 브로커의 토픽에 등록된 데이터를 순서대로 가져옴

- (성능검증) RTGS 프로토타입 시스템의 **초당 최대 접수 처리건수(TPS)**를 파악하고, 건당 처리되는 **평균 end-to-end 처리시간**(최초 접수부터 최종 처리 완료시간)을 파악

성능테스트 환경

출발지	목적지	초기 VU* 수	최대 VU 수	부하 건수(초당)	테스트 시간
RTGS시스템팀 개발PC	전문 송수신 서비스	500명	10,000명	5,000건 ~ 20,000건 (10초마다 5,000건씩 증가)	40초

* VU(Virtual User)는 부하테스트 도구인 k6에서 제공하는 가상 사용자 기능으로 Client와 Server간의 연결된 소켓을 의미함

RTGS 시스템 프로토타입 성능테스트 방법

테스트 대상	테스트 방법
최대 접수 처리건수 (TPS)	점진적으로 초당 거래요청을 늘려가며 최대 접수 처리건수 확인
평균 End-to-End 처리 시간	(마지막 거래가 처리 완료된 시간 - 테스트 시작 시간)/총 처리건수

- (복원력검증) 2개 센터를 Active-Active로 운영하는 환경에서 1개 센터에 장애가 발생해도 나머지 센터에서 정상 운영되고 이후 장애 복원 후 2개 센터의 운영 여부를 파악

- 장애가 발생 후 복원 시간을 고려하여 총 10분간의 테스트 진행 중간에 1개 센터 전체를 2분간 중지시켰다가 복원하는 테스트 수행(약 5분 소요)

복원력테스트 환경

출발지	목적지	부하 건수(초당)	장애 시나리오	테스트 시간
RTGS시스템팀 개발PC	전문 송수신 서비스	100건	① 장애 발생(1개 센터) ② 장애 유지(2분간) ③ 장애 복구 ④ 복구 확인	10분

RTGS 시스템 프로토타입 복원력테스트 방법

테스트 대상	테스트 방법	데이터 정합성 확인
센터 장애 발생에 대한 복원력	수도권/남부권 센터 중 한 곳의 서버 연결 단절	두 센터 간 처리건수 및 이체 금액 대사

Ⅲ

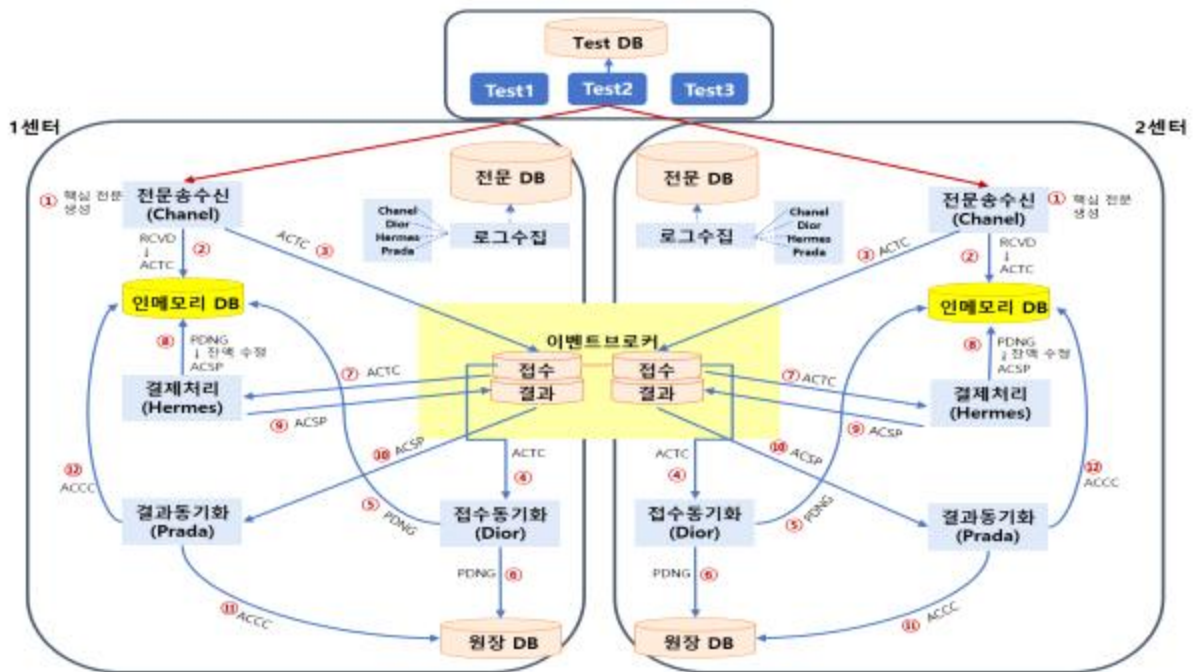
PoC 테스트 결과

1. 기능테스트 결과

□ 테스트 케이스별로 기능테스트를 실시한 결과, **모두 정상 수행되는 것을 확인**

○ 부하테스트 도구를 통해 접수한 건 모두 처리완료 후 상태값이 ACCC (입금처리완료)로 저장되고 처리 로그는 전문DB에 자동 저장

RTGS 프로토타입 시스템 업무 흐름도*



* 프로토타입 시스템의 세부 프로세스는 (별첨3) 참조

기능테스트 결과

동기화 방식	인메모리DB	원장DB	전문송수신	접수동기화	결제처리	결과동기화
① 이벤트 브로커 (Confluent)	Hazelcast	MongoDB	기능 확인	기능 확인	기능 확인	기능 확인
	Goldilocks	Goldilocks	기능 확인	기능 확인	기능 확인	기능 확인
② 이벤트 브로커 (Solace)	Hazelcast	MongoDB	기능 확인	기능 확인	기능 확인	기능 확인
	Goldilocks	Goldilocks	기능 확인	기능 확인	기능 확인	기능 확인
③ 인메모리 클러스터링 (Hazelcast)	Hazelcast	MongoDB	기능 확인	기능 확인		

2. 성능테스트 결과

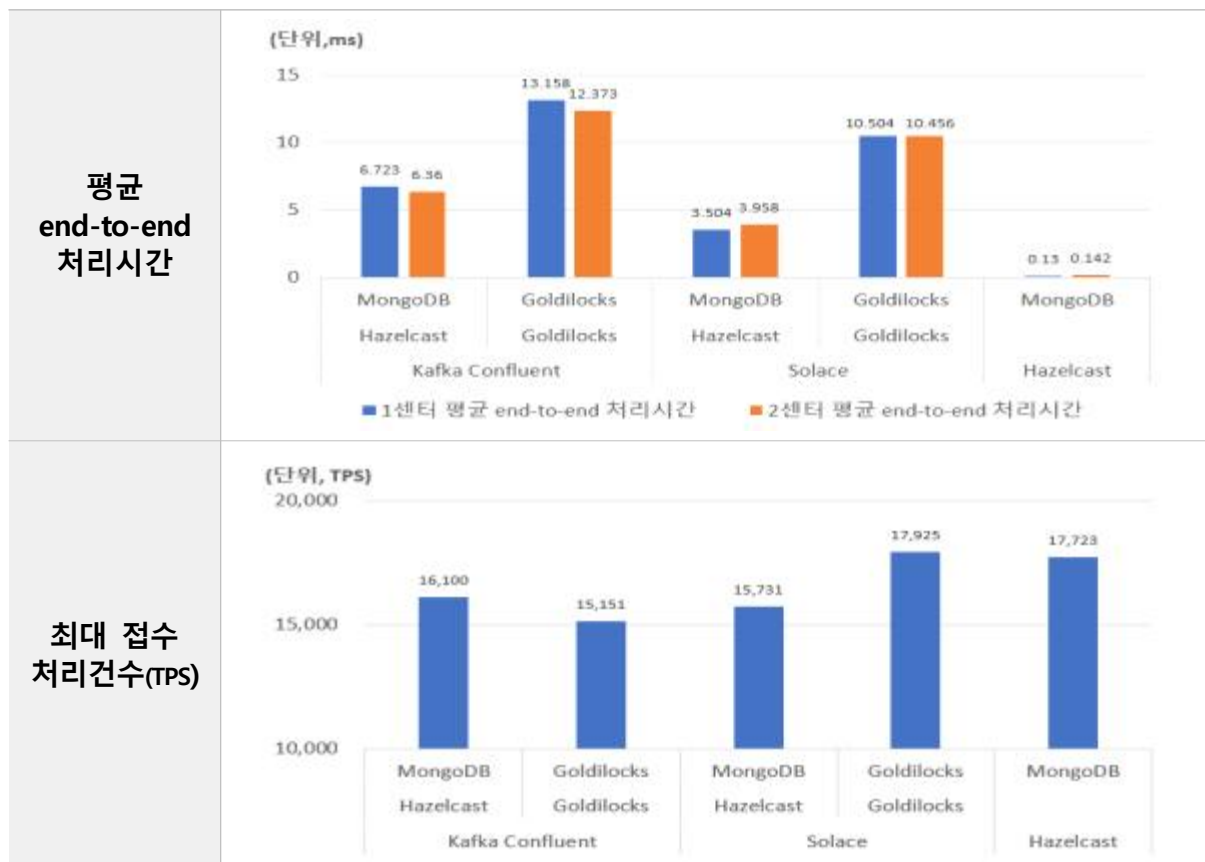
- 테스트 케이스 모두 최대 접수 처리건수는 15,000건 이상이였으며, 평균 end-to-end 처리시간은 ② Solace+Hazelcast+MongoDB 조합이 3.504ms ~ 3.958ms로 가장 빠르게 처리되는 것을 확인

성능테스트 결과

동기화 방식	인메모리DB	원장DB	1센터 평균 end-to-end 처리시간	2센터 평균 end-to-end 처리시간	최대 접수 처리건수(TPS)
① 이벤트 브로커 (Confluent)	Hazelcast	MongoDB	6.723ms	6.360ms	16,100건
	Goldilocks	Goldilocks	13.158ms	12.373ms	15,151건
② 이벤트 브로커 (Solace)	Hazelcast	MongoDB	3.504ms	3.958ms	15,731건
	Goldilocks	Goldilocks	10.504ms	10.456ms	17,925건
③ 인메모리 클러스터링 (Hazelcast)	Hazelcast	MongoDB	0.13ms*	0.142ms*	17,723건

* ③번 케이스는 Multi-Thread 방식으로 매우 빠른 성능으로 접수처리와 동시에 DB의 상태값 업데이트 처리까지 완료되고 양 센터의 데이터 정합성을 보장했으나, 결제처리 순서 보장은 안됨

성능테스트 결과 요약



3. 복원력테스트 결과

- 1개 센터 장애에 대한 복원력테스트 수행한 모든 테스트 케이스에서 양 센터 간의 데이터가 일치하는지 정합성을 확인

복원력테스트 결과

동기화 방식		인메모리DB	원장DB	1센터 장애 시 정합성 결과	2센터 장애 시 정합성 결과
①	이벤트 브로커 (Confluent)	Hazelcast	MongoDB	측정 불가능*	데이터 일치
		Goldilocks	Goldilocks	측정 불가능*	데이터 일치
②	이벤트 브로커 (Solace)	Hazelcast	MongoDB	데이터 일치	데이터 일치
		Goldilocks	Goldilocks	데이터 일치	데이터 일치
③	인메모리 클러스터링 (Hazelcast)	Hazelcast	MongoDB	데이터 일치	데이터 일치

* 다만, 과반 정족 수 합의 알고리즘에 의해 최소 3개 센터가 필요로 하는 이벤트 브로커(Confluent)는 1센터 장애 테스트는 불가능

<참고 4>

테스트 케이스별 복원력테스트 수행 상세절차

- 테스트*는 「장애 발생 → 장애 유지 → 장애 복구 → 복구 확인」 단계별 순서로 수행

* 테스트 케이스별 구체적인 수행 내역은 (별첨4)의 내용을 참조

①, ②번 테스트 케이스 복원력테스트 수행 방법

장애 발생	장애 유지	장애 복구	복구 확인
① Chanel 정지 ② Hermes, Dior, Prada 정지 ③ 인메모리DB, 원장DB 정지 ④ 이벤트 브로커 정지	① 장애 2분 동안 지속	① 이벤트 브로커 시작 ② 인메모리DB, 원장DB 시작 ③ Hermes, Dior, Prada 시작 ④ Chanel 시작	① 서비스별 로그 확인 ② 계좌 데이터 정합성 확인

③번 테스트 케이스 복원력테스트 수행 방법

장애 발생	장애 유지	장애 복구	복구 확인
① Hazelcast-Chanel 정지 ② Hazelcast 클러스터링 정지 ③ 원장DB 정지	① 장애 2분 동안 지속	① 원장DB 시작 ② Hazelcast 클러스터링 시작 ③ Hazelcast-Chanel 시작	① 서비스별 로그 확인 ② 계좌 데이터 정합성 확인

IV

주요 성과 및 향후 계획

□ (주요 성과) PoC 테스트 사업을 통해 RTGS 아키텍처 설계, 실현가능성 검증, 향후 프로젝트에 대한 상세 요건 및 시사점 도출하는 성과를 얻음

1) 당행에 적합한 RTGS 핵심 아키텍처를 설계하여 핵심 기능을 검증

- 트래픽 분산 처리와 장애가 발생해도 서비스 중단 없이 계속 동작할 수 있는 다중센터 Active-Active 아키텍처를 설계
- RTGS 지급결제 거래과정 중 핵심 영역인 전문송수신, 결제처리, 센터간 동기화를 대상으로 RTGS 프로토타입 시스템을 자체 개발

2) RTGS시스템 구축에 필요한 성능, 안정 및 확장에 대한 실현가능성을 검증

- ① 성능 : 초당 15,000건 이상의 접수처리, 1건 처리시간 0.014초 이내 확인
- ② 안정 : 1개 센터 장애 발생 중에도 서비스 지속 및 장애센터의 복원력 확인
- ③ 확장 : ISO20022 전문처리, API 서비스 개발

3) 향후 프로젝트 추진에 대한 상세 요건 및 시사점 도출

- 솔루션 장비 개선, 3개 센터 확장 및 인증·암복호화 추가, 1초 이내로 최종처리까지 가능하도록 end-to-end 처리성능 개선, 센터간 처리속도 차이에 따른 센터간 동기화 보장 등 프로그램 개선 필요
- 구성 요소별 장애 및 장애 지속시간 장기화 대처 등 다양한 장애 및 복구에 대한 테스트 확인 필요

□ 다만, 본 PoC 3차 테스트는 센터간 Active-Active 환경에서 성능과 데이터 정합성을 측정하기 위한 목적으로 원격지에서 2개의 공공 리전을 제공하는 클라우드 환경에서 진행

- 또한, RTGS 핵심 기능만을 프로토타입 시스템으로 개발하여 인증과 암호화 및 각종 예외 처리 등은 생략된 실험실 환경에서 테스트를 실시했기에 추가 실험 필요

- (향후 계획) 이러한 프로토타입의 아키텍처는 제약사항을 개선하여 2025년 하반기 PoC 추가 테스트를 검토하고 있으며, PoC 사업 결과를 바탕으로 2026 ~ 2027년 BMT* 사업을 추진할 계획

* BMT는 벤치마크 테스트(BenchMark Test)로 하드웨어 인프라에 소요되는 성능이 어느 정도인지 비교평가하기 위해 수행하는 테스트

① 2025년 하반기 PoC 테스트(4차) 계획

- 과반 정족 수 합의 알고리즘에 따라 3개 이상의 센터 필요한 경우와 센터 간 통신 장애 등에 대한 이슈를 해결할 수 있는 3개 센터를 이용한 복원력테스트를 실시

- 솔루션 업체의 소프트웨어 외에 어플라이언스 장비*를 활용한 최고 성능테스트를 실시

* 어플라이언스 장비는 특정 기능을 수행하도록 하드웨어와 소프트웨어가 통합된 일체형 장비로 PoC에서 사용했던 솔루션 소프트웨어보다 더 높은 성능을 가짐

- 각 센터의 결제처리 순서가 일치하는지 확인할 수 있는 조회 프로그램 개발 및 검증

② 2026 ~ 2027년 BMT* 추진 계획

- 높은 성능과 보안을 위한 프로세스 추가 및 개선하고 세부 서비스별 장애 등 다양한 장애 발생 및 복원력테스트를 실시하여 검증
- 각 솔루션별 CPU, 메모리, 스토리지 등 서버 시스템 자원의 사용량에 대한 자원 소모량 측정을 위한 성능테스트를 실시

향후 RTGS시스템 추진 계획

구분	2025						2026	2027	2028	2029	2030
	7	8	9	10	11	12					
		PoC 테스트						BMT	설계	시스템 구축	
프로토타입 시스템 개선											
프로토타입 PoC 테스트(4차)											
아키텍처 설계 개선											
1단계 사업(BMT)											
시스템 상세 설계											
2단계 사업(시스템 구축)											

PoC 테스트 용어정리

1) 쿠버네티스

- 쿠버네티스(Kubernetes, 줄여서 K8s)는 컨테이너화된 애플리케이션의 배포, 확장, 관리 등을 자동화하기 위한 오픈소스 플랫폼으로 **다수의 컨테이너*를 효율적으로 운영하고 관리할 수 있는 컨테이너 오케스트레이션 도구****

* 컨테이너(Container)는 애플리케이션과 그 실행 환경을 격리된 단위로 패키징하여 어디서든 동일하게 실행하는 가상화 기술

** 구글에서 시작한 프로젝트로, 현재는 클라우드 네이티브 컴퓨팅 재단(CNCF)에서 관리

- 애플리케이션 배포, 업그레이드, 복구를 자동화할 수 있음
- 컨테이너의 수를 사용량에 따라 자동으로 증가/감소할 수 있음
- 서비스 간 통신을 쉽게 할 수 있도록 자동으로 DNS 이름 또는 IP 주소를 할당하고 로드 밸런싱을 지원
- 문제가 발생한 컨테이너를 자동으로 재시작하거나 교체할 수 있음
- YAML 또는 JSON 파일로 애플리케이션과 클러스터의 상태를 정의할 수 있음

2) 가상서버

- 가상서버 기반 환경은 **물리 서버 위에 하이퍼바이저*를 설치하고, 그 위에 여러 개의 가상머신**을 구동하는 방식**

* 하이퍼바이저는 물리 서버 위에서 여러 개의 가상머신을 생성하고 실행할 수 있도록 해주는 소프트웨어

** 가상머신은 하이퍼바이저에 의해 생성된 논리적인 컴퓨터

- 각 가상서버는 자체 OS를 포함하여 독립된 컴퓨팅 환경을 제공
- 전통적인 애플리케이션과 서버 운영 방식에 익숙한 환경임
- 가상머신 간 격리 수준이 높아 보안성이 좋음

가상서버와 쿠버네티스 비교

항목	가상서버	쿠버네티스
기본 기술	하이퍼바이저+가상머신	컨테이너+오케스트레이션
격리 수준	높음(OS 단위)	낮음(프로세스 단위)
스케일링	수동 또는 제한적	자동화 기능
유지보수	수동, 복잡함	자동화 기능
운영 복잡도	상대적으로 낮음	초기에는 복잡하지만 자동화로 유리

3) k6

□ k6는 웹 서비스의 성능과 안정성을 검증하기 위해, 가상의 사용자가 실제처럼 접근하여 **부하를 발생시키는 오픈소스 테스트 도구**

- 리소스 최적화를 위한 용량 계획의 근거 데이터를 확보
- JavaScript 기반 스크립트를 작성하기에 개발자에게 친숙
- 클라우드에 연동이 가능하여 확장성이 좋음

부하테스트 도구 비교

항목	k6	JMeter	Locust	Gatling
언어	JavaScript	Java/XML	Python	Scala
사용자 인터페이스	코드 기반	GUI 중심	코드 기반	코드 기반
클라우드 적용	매우 쉬움	가능하나 복잡	가능	가능
학습 난이도	낮음	중간	중간	높음
성능	매우 높음	상대적으로 낮음	중간	높음
스크립트 가독성	높음	낮음	높음	중간
실시간 시각화	Grafana 연동(강력)	별도 설치 필요	제한적	별도 설정 필요

4) Grafana

□ Grafana는 **다양한 데이터 소스를 시각화하고 대시보드를 구성**할 수 있도록 도와주는 **오픈소스 모니터링 도구**

- 시계열 데이터를 기반으로 시스템 성능, 트래픽, 응답 시간, 장애 여부 등을 실시간으로 모니터링
- 사용자 정의 대시보드 구성이 가능하고 다양한 데이터 소스 지원

모니터링 도구 비교

항목	Grafana	Kibana	Prometheus	Datadog	Zabbix
주요 역할	시계열 데이터 시각화	Elasticsearch 시각화	메트릭 수집+ 일부 시각화	종합 모니터링 SaaS	SNMP/Agent 기반 모니터링
데이터 소스	다중 지원	Elasticsearch 전용	자체 시계열 데이터베이스	자체 수집	에이전트 수집 위주
시각화 수준	매우 높음	로그 위주 시각화	기본적인 그래프	높음	제한적
알림	내장+연동 가능	최근 지원	내장	내장(강력)	내장
설치/운영 난이도	쉬움	쉬움	중간	쉬움	복잡함
라이선스	AGPL	ELv2	Apache 2.0	유료	GPLv2

5) InfluxDB

☐ InfluxDB는 **시계열 데이터를 저장하고 분석하기 위한 데이터베이스**

- 시계열 데이터에 대해 고성능 데이터 처리 가능
- CPU 사용량, 메모리, 디스크 I/O 등 실시간 시스템 메트릭 수집과 주식 거래 및 금융 데이터의 변동 추적 및 저장 등에 용이

시계열 데이터베이스 비교

항목	InfluxDB	TimescaleDB	OpenTSDB	Graphite
개발 언어	Go	C	Java	Python
저장 방식	자체 시계열 엔진	PostreSql 확장	HBase 기반 (빅데이터용 저장소)	파일 포맷 저장
쿼리 언어	InfluxQL	SQL	복잡한 명령어 체계	Graphite 함수 기반
알림 기능	내장 or 외부연동	외부 도구 필요	없음	없음
실시간 처리	우수	보통	느림	느림
운영 편의성	쉬움	SQL 친숙	복잡	복잡
라이선스	MIT	Apache 2.0	LGPL	Apache 2.0s

6) TPS

☐ TPS(Transactions Per Second)는 시스템이 **1초 동안 처리할 수 있는 거래 또는 요청의 수**를 나타내는 성능 지표

- 빠르게 많은 요청을 처리할 수 있는지를 수치로 표현 가능
- 시스템이 견딜 수 있는 최대 TPS 테스트 필요

7) ms

☐ IT에서 **성능을 다룰 때 자주 등장하는 ms는 밀리초(milliseconds)**를 의미

- 컴퓨터, 네트워크, 서버 등 IT시스템은 빠르게 동작하기 때문에 초 단위로는 성능을 측정하기에는 큰 단위

☐ 적용사례

- TPS : 1초 동안 요청 및 처리한 건수
- 응답시간 : 서버가 요청을 받고 응답하기까지 걸리는 시간
- 처리시간 : 내부 로직을 처리하는 데 걸리는 시간

8) 리전(Region) & 레이턴시(Latency)

□ 리전(Region)은 클라우드 데이터 센터의 지리적 위치 단위를 의미

- NCP(Naver Cloud Platform)은 수도권, 남부권 2개의 공공 리전* 제공
- KT Cloud(KTC)는 수도권, 충청권 2개의 공공 리전 외에 2026년 3월부터 경북권에 공공 리전을 추가예정

* 공공 리전은 공공기관 전용으로 운영하는 물리적 데이터센터 지역 또는 구역

□ 레이턴시(Latency)는 데이터가 네트워크를 통해 이동하는데 걸리는 지연 시간을 의미

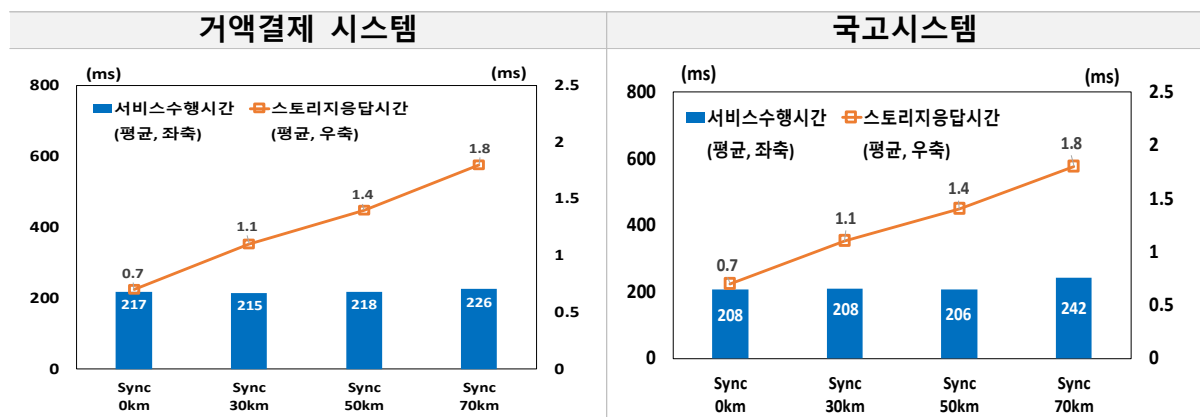
- 주로 요청을 보내고 서버에서 응답이 다시 돌아오기까지의 왕복시간인 RTT(Round Trip Time)로 레이턴시를 측정함

지연 요소

종류	설명	예시
네트워크 지연	거리, 라우터 수 등에 따라 증가	3200km 거리인 Fed Now의 미국 동부와 서부 리전 간에는 100ms의 지연 발생
서버 처리 지연	요청을 받은 서버가 처리하는 시간	DB 조회, 로직 실행 등
큐잉 지연	요청이 밀려서 대기하는 시간	트래픽 몰릴 때 증가
정보보호 등	정보보호를 위해 처리하는 시간	방화벽 1개 추가 시마다 약 1ms 지연

- 당행 결제시스템팀에서 수행한 테스트에서 거리별 레이턴시는 50km에 1.4ms 지연이 발생하는 것으로 파악

거리와 레이턴시의 관계*



* 「IT센터 격지간 회계결제시스템 디스크 동기화 테스트 결과 보고」(결제시스템팀-157, 2024.2.8.)의 결과값 참고

(별첨2)

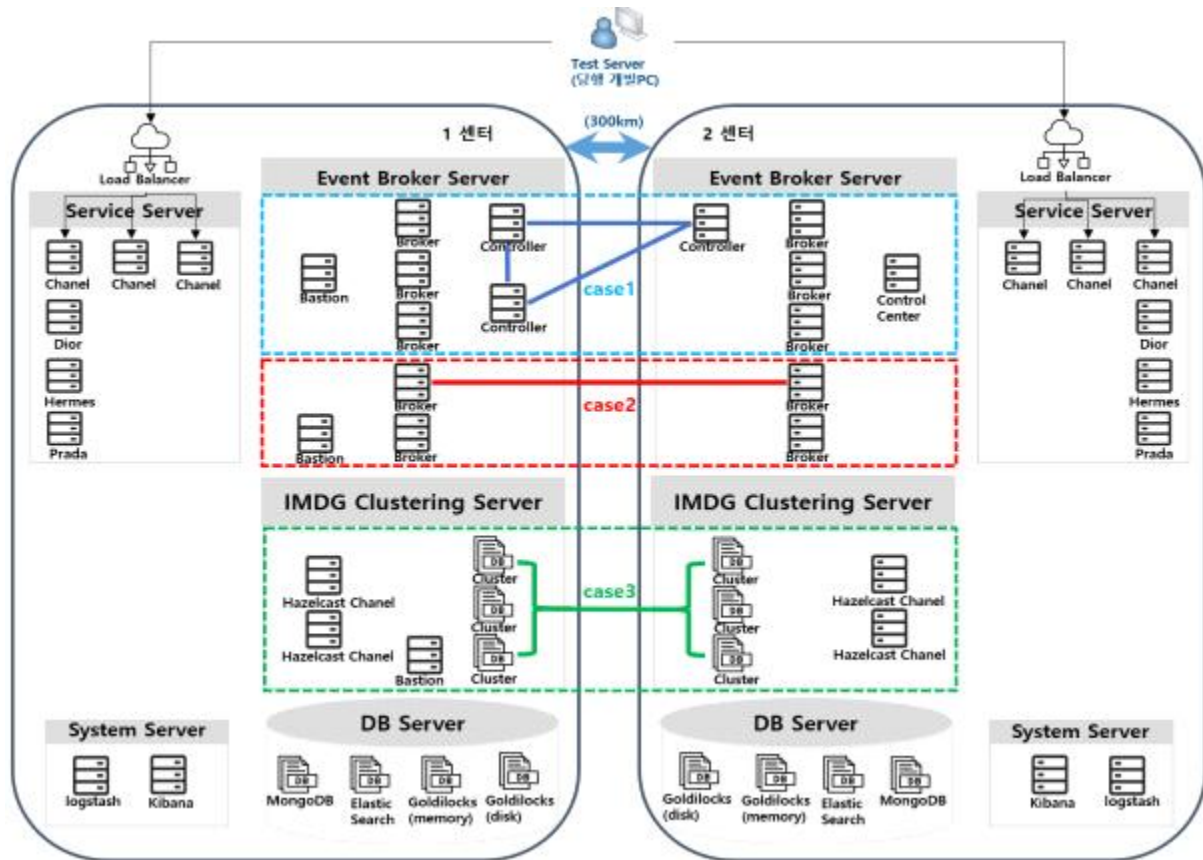
PoC 테스트를 위한 IT인프라 구성방안

- RTGS시스템 구축을 위한 프로토타입 시스템은 네이버 공공 클라우드*에서 제공하는 2개의 센터(수도권, 남부권)에 분산하여 구성

* 공공 기관에서 사용할 수 있는 클라우드 : 국가공공기관 민간클라우드 컴퓨팅서비스 도입 가능목록(국가 사이버안보센터, <https://www.ncsc.go.kr:4018>)

- 거래 접수를 최대로 처리하기 위해 Chanel 노드를 3개로 구성하여 로드 밸런스로 분산 부하

네이버 공공 클라우드(NCP) 서버 구성



테스트 케이스 구성

동기화 방식	테스트 케이스 번호	인메모리DB	원장DB
case 1	이벤트 브로커 (Confluent) ①-1	Hazelcast	MongoDB
	①-2	Goldilocks	Goldilocks
case 2	이벤트 브로커 (Solace) ②-1	Hazelcast	MongoDB
	②-2	Goldilocks	Goldilocks
case 3	인메모리 클러스터링 (Hazelcast) ③	Hazelcast	MongoDB

NCP 서버 상세내역

구분	리전		테스트 케이스*	서비스	IP주소	vCPU(core)	Memory(GB)	
IT센터	수도권	K8s	①-1,②-1	mongodb	(유동적)	2	8	
			①-1,①-2, ②-1,②-2	chanel		32	128	
				chanel		32	128	
				chanel		32	128	
				dior		2	4	
				hermes		2	4	
				prada		2	4	
			①-1,①-2, ②-1,②-2	elasticsearch		1	4	
		logstash		(유동적)				
		kibana		1		2		
		VM	①-1,①-2	tg-br-001	10.0.1.22	4	16	
				tg-br-002	10.0.1.21	4	16	
				tg-br-003	10.0.1.24	4	16	
				tg-kc-001	10.0.1.19	2	8	
				tg-kc-002	10.0.1.21	2	8	
				tg-bastion	10.0.1.18	2	4	
				②-1,②-2	rtgs-solace01	10.0.1.28	16	64
			rtgs-solace02		10.0.1.30	16	64	
			rtgs-solace03		10.0.1.31	4	16	
			solace-bestion		10.0.5.8	2	8	
			①-2,②-2	gd-bastion	10.0.5.7	2	8	
				goldilocks-01	10.0.1.27	8	16	
				goldilocks-02	10.0.1.8	8	16	
			①-1,②-1, ③	Hazelcast-bastion	10.0.5.9	2	4	
				rtgs-Hazelcast01	10.0.1.12	8	32	
				rtgs-Hazelcast02	10.0.1.33	8	32	
				rtgs-Hazelcast03	10.0.1.34	8	32	
				③	Hazelcast-chanel001	10.0.5.10	8	32
		Hazelcast-chanel002	10.0.5.11		8	32		
	남부권	K8s	①-1,②-1	mongodb	(유동적)	2	8	
			①-1,①-2, ②-1,②-2	chanel		32	128	
				chanel		32	128	
				chanel		32	128	
				dior		2	4	
				hermes		2	4	
				prada		2	4	
			①-1,①-2, ②-1,②-2	elasticsearch		1	4	
		logstash		(유동적)				
		kibana		1		2		
		VM	①-1,①-2	tg-br-004	10.1.1.18	4	16	
				tg-br-005	10.1.1.19	4	16	
				tg-br-006	10.1.1.20	4	16	
				tg-kc-003	10.1.1.17	2	8	
				tg-cc-001	10.1.1.23	4	16	
			②-1,②-2	rtgs-solace11	10.1.1.24	16	64	
				rtgs-solace12	10.1.1.25	16	64	
				rtgs-solace13	10.1.1.26	4	16	
			①-2,②-2	goldilocks-01	10.1.1.27	8	16	
				goldilocks-02	10.1.1.6	8	16	
				Hazelcast-bastion	10.1.5.6	2	4	
			①-1,②-1, ③	rtgs-Hazelcast11	10.1.1.22	8	32	
				rtgs-Hazelcast12	10.1.1.28	8	32	
				rtgs-Hazelcast13	10.1.1.29	8	32	
				③	Hazelcast-chanel011	10.1.5.7	8	32
			Hazelcast-chanel012		10.1.5.8	8	32	
			참가기관	당행 사무실	VM	모든 테스트 케이스	k6 컨테이너	10.16.16.101
10.16.16.102	24							128
10.16.16.72	24	125						

* 테스트 케이스 번호 ①-1은 Confluent+Hazelcast+MongoDB, ①-2은 Confluent+Goldilocks, ②-1은 Solace+Hazelcast+MongoDB, ②-2는 Solace+Goldilocks, ③은 Hazelcast+MongoDB를 나타냄

(별첨3)

PoC 테스트를 위한 프로세스

□ Chanel* 서비스 세부절차

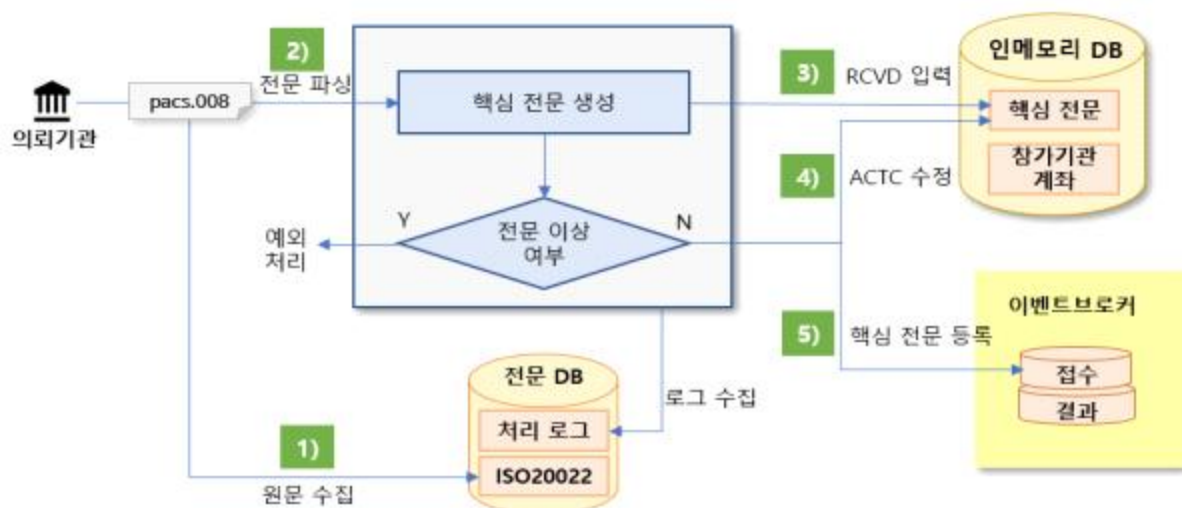
* Contact Hub for Advanced Networked Entry Liaison의 약자로 참가기관(의뢰기관)으로부터 거래 요청을 접수 받아 참가기관에게(수취기관)에게 처리결과를 통보하는 서비스

- 1) 요청의 전문 원본은 전문DB에 별도로 저장(Insert)
- 2) HTTP Request Body의 ISO20022 전문(pacs.008) 파일을 XML을 파싱(Parsing)하여 핵심 전문으로 생성(4,340Bytes → 75Bytes)
- 3) 핵심 전문의 처리상태값을 RCVD(접수)로 인메모리DB에 최초 저장(Insert)
- 4) 검증과정을 거친 후에 핵심 전문의 처리상태값을 ACTC(승인)로 인메모리DB에 수정(Update)
- 5) 센터 간 요청 동기화를 위해 이벤트 브로커의 접수 토픽으로 등록(Produce)

Chanel에서 DB별 작업 현황

DB 종류	작업 순서			
	순서	데이터	상태값	작업
전문DB	-	로그	-	Insert
	1)	XML 원문	-	Insert
	2)	XML 원문	-	Parsing
인메모리DB	3)	핵심 전문	RCVD	Insert
	4)	핵심 전문	ACTC	Update
이벤트 브로커 원장DB	5)	핵심 전문	ACTC	Produce
	-	-	-	-

Chanel 프로그램 흐름도



□ Dior* 서비스 세부절차

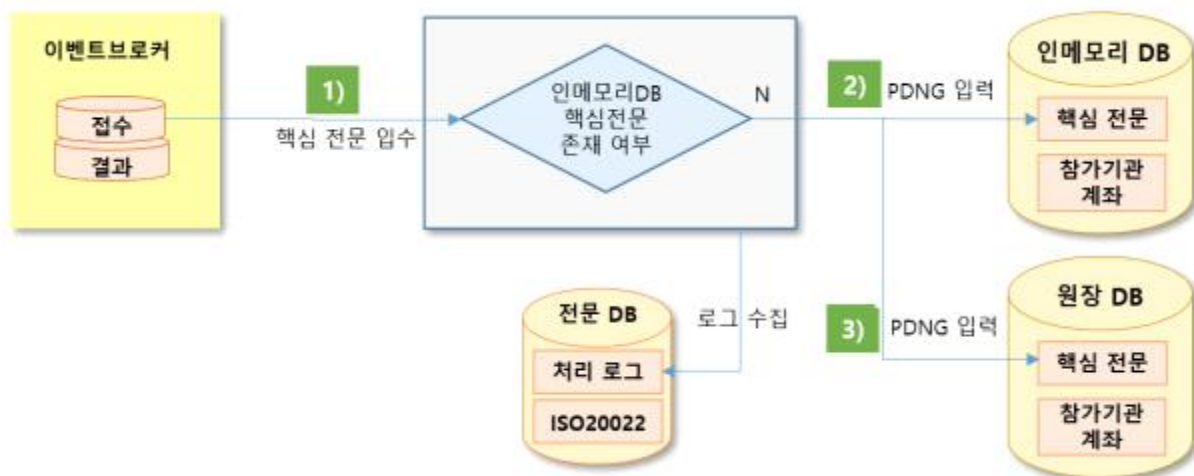
* Data Integration and Organization Relay의 약자로 참가기관(의료기관)이 접수한 거래를 다중센터에서 동일하게 처리하기 위한 접수 동기화 서비스

- 1) 이벤트 브로커의 접수 토픽(pacs.008)에서 처리상태값이 ACTC(승인)인 핵심 전문을 입수(Consume)해서 인메모리DB에 같은 BMI값이 있는지 체크
- 2) BMI값이 없으면 타 센터에서 온 핵심 전문을 의미하므로 처리상태값을 PDNG(대기)로 수정 후 인메모리DB에 최초 저장(Insert)
- 3) 핵심 전문의 처리상태값을 PDNG(대기)로 원장DB에 저장(Insert)(센터간 동기화 및 재해 복구 대비)

Dior에서 DB별 작업 현황

DB 종류	작업 순서			
	순서	데이터	상태값	작업
전문DB	-	로그	-	Insert
인메모리DB	2)	핵심 전문	PDNG	Insert
이벤트 브로커	1)	핵심 전문	ACTC	Consume
원장DB	3)	핵심 전문	PDNG	Insert

Dior 프로그램 흐름도



□ Hermes* 서비스 세부절차

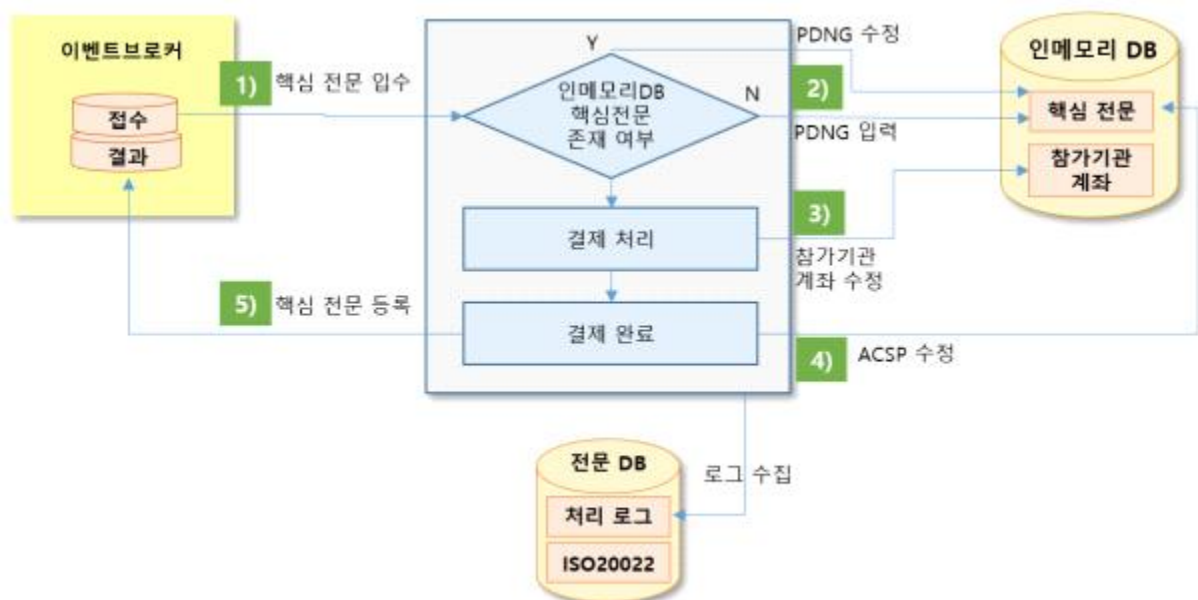
* High-Efficiency Rapid Memory Execution for Settlement의 약자로 각 센터에서 동일한 거래를 순서대로 처리하여 각 센터의 DB에 저장하는 결제처리 서비스

- 1) 이벤트 브로커의 접수 토픽(pacs.008)에서 처리상태값이 ACTC(승인)인 핵심 전문을 입수해서 인메모리DB에 같은 BMI값이 있는지 체크
- 2) BMI값이 없으면 타 센터에서 온 핵심 전문을 의미하므로 처리상태값을 PDNG(대기)로 인메모리DB 최초 저장(Insert)
- 3) 핵심 전문을 바탕으로 인메모리DB에 있는 계좌 정보를 수정(Update)
- 4) 인메모리DB에 있는 핵심 전문의 처리상태값을 ACSP(예약)로 수정(Update)
- 5) 센터간 동기화 및 영구 기록을 위해 이벤트 브로커의 결과 토픽(pacs.002)으로 처리상태값 ACSP(예약)인 핵심 전문을 등록(Produce)

Hermes에서 DB별 작업 현황

DB 종류	작업 순서			
	순서	데이터	상태값	작업
전문DB	-	로그	-	Insert
인메모리DB	2)	핵심 전문	PDNG	Insert
	3)	참가기관 계좌	-	Update
	4)	핵심 전문	ACSP	Update
이벤트 브로커	1)	핵심 전문	ACTC	Consume
	5)	핵심 전문	ACSP	Produce
원장DB	-	-	-	-

Hermes 프로그램 흐름도



□ Prada* 서비스 세부절차

* Persistent Repository Administrator for Data Aggregation의 약자로 각 센터에서 처리한 거래처리 결과에 대한 결과 동기화 서비스

- 1) 이벤트 브로커의 결과 토픽(pacs.002)에서 처리상태값이 ACSP(예약)인 핵심 전문을 입수(Consume)
- 2) 핵심 전문을 바탕으로 원장DB에 있는 계좌 데이터 수정(Update)(센터간 동기화 및 재해 복구 대비)
- 3) 원장DB에 있는 핵심 전문의 처리상태값을 ACCC(입금처리 완료)로 최종 수정(Update)
- 4) 인메모리DB에 있는 핵심 전문의 처리상태값을 ACCC(입금처리 완료)로 최종 수정(Update)

Prada에서 DB별 작업 현황

DB 종류	작업 순서			
	순서	데이터	상태값	작업
전문DB	-	로그	-	Insert
인메모리DB	4)	핵심 전문	ACCC	Update
이벤트 브로커	1)	핵심 전문	ACSP	Consume
원장DB	2)	참가기관 계좌	-	Update
	3)	핵심 전문	ACCC	Update

Prada 프로그램 흐름도

