

Gizzard - A library for creating distributed datastores (서버 기술 분석)

개발실 / 데이터지능팀 / 성동찬

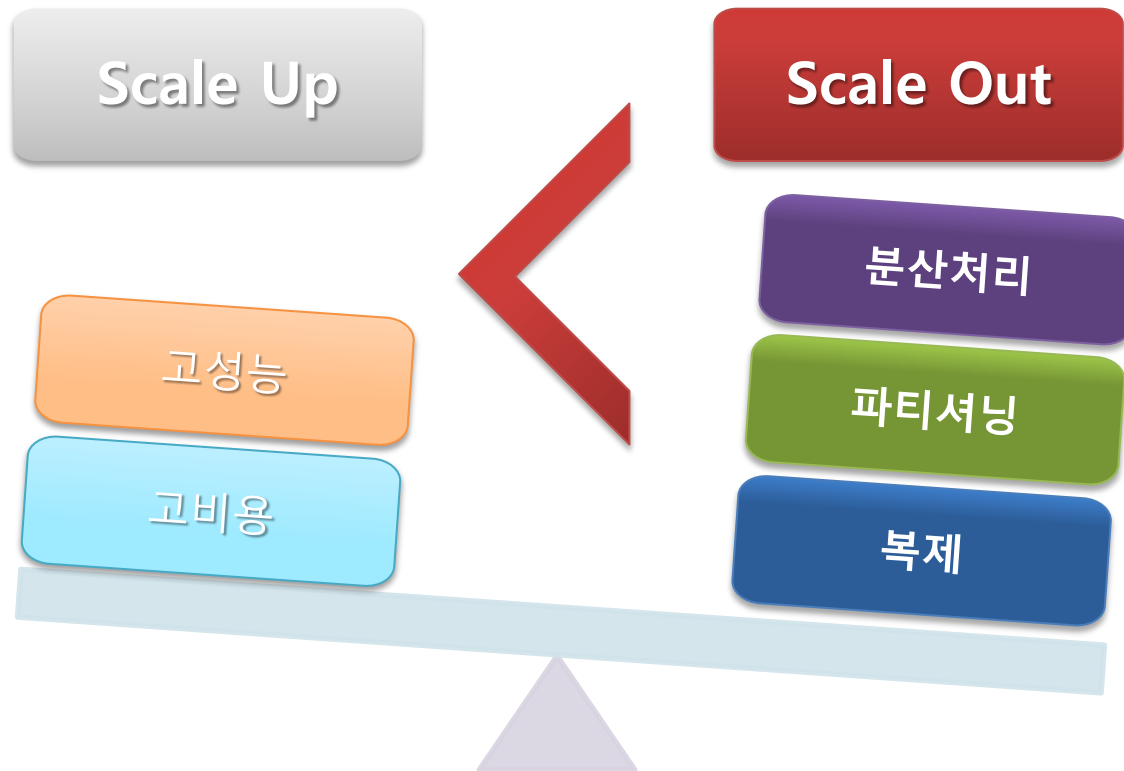


INTRODUCTION - SHARD



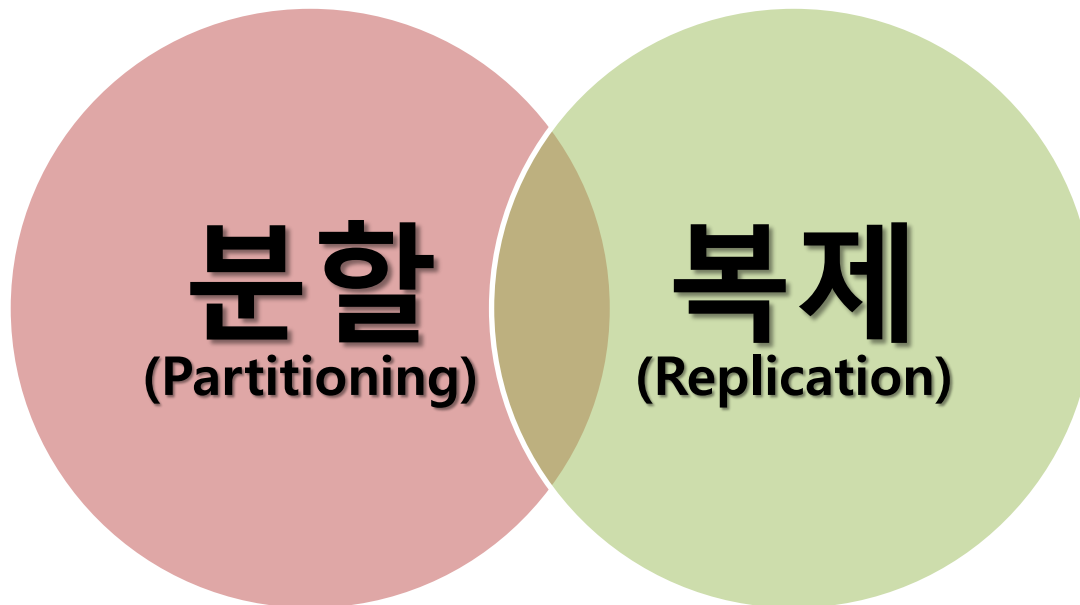
❖ 데이터 거대화에 따라 데이터 분산 처리 필요

- ✓ 고성능 단일 서버보다는 다수 서버에 분산 처리 유도하는 것이 효율적임
- ✓ 파티셔닝, 데이터 복제 등을 통해서 데이터 분산 유도



❖ 대용량 데이터 분할 및 복제로 분산 유도

- ✓ 분할(Partitioning)은 데이터를 여러 조각으로 나누어서 여러 서버에 저장
- ✓ 복제(Replication)은 동일한 데이터를 여러 서버에 저장

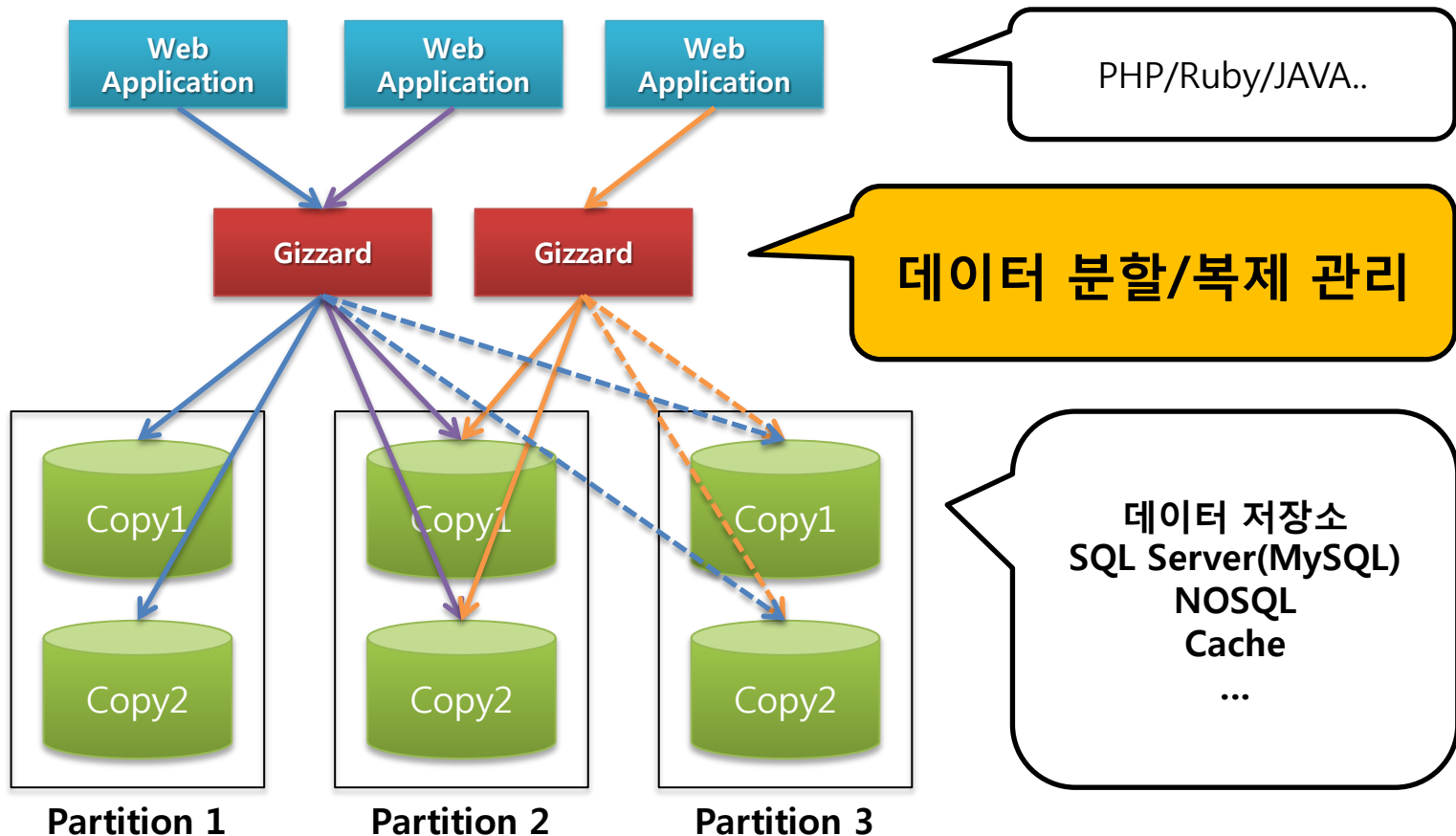


WHAT IS GIZZARD?



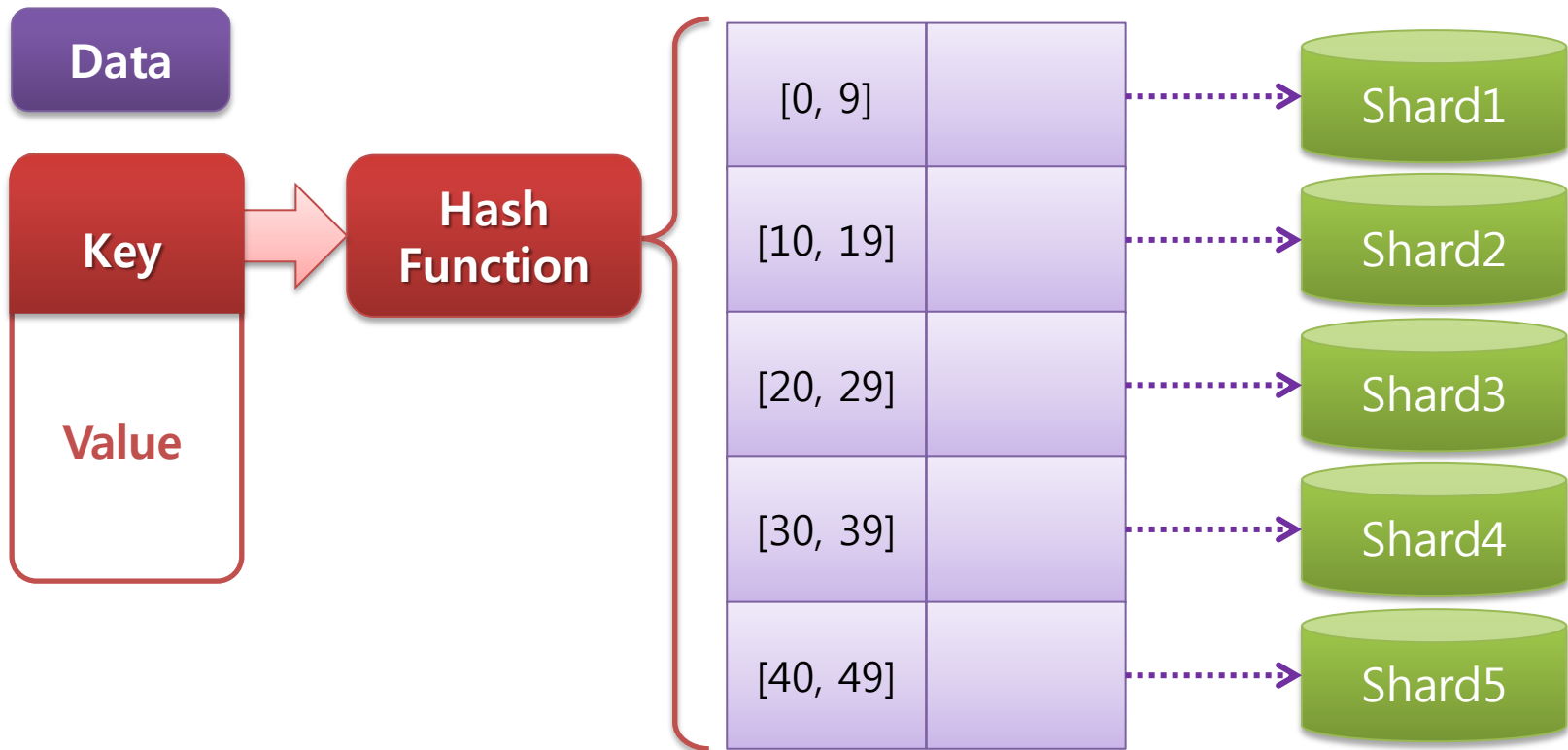
❖ 데이터 분산 저장을 위한 미들웨어

- ✓ 어플리케이션 서버와 데이터 저장소 중간에 위치하며 초당 10,000쿼리 처리
- ✓ 멥등성/가환성 원칙에 의거하여 데이터 일관성을 유지 → 데이터 변경 순서 보장 안 함



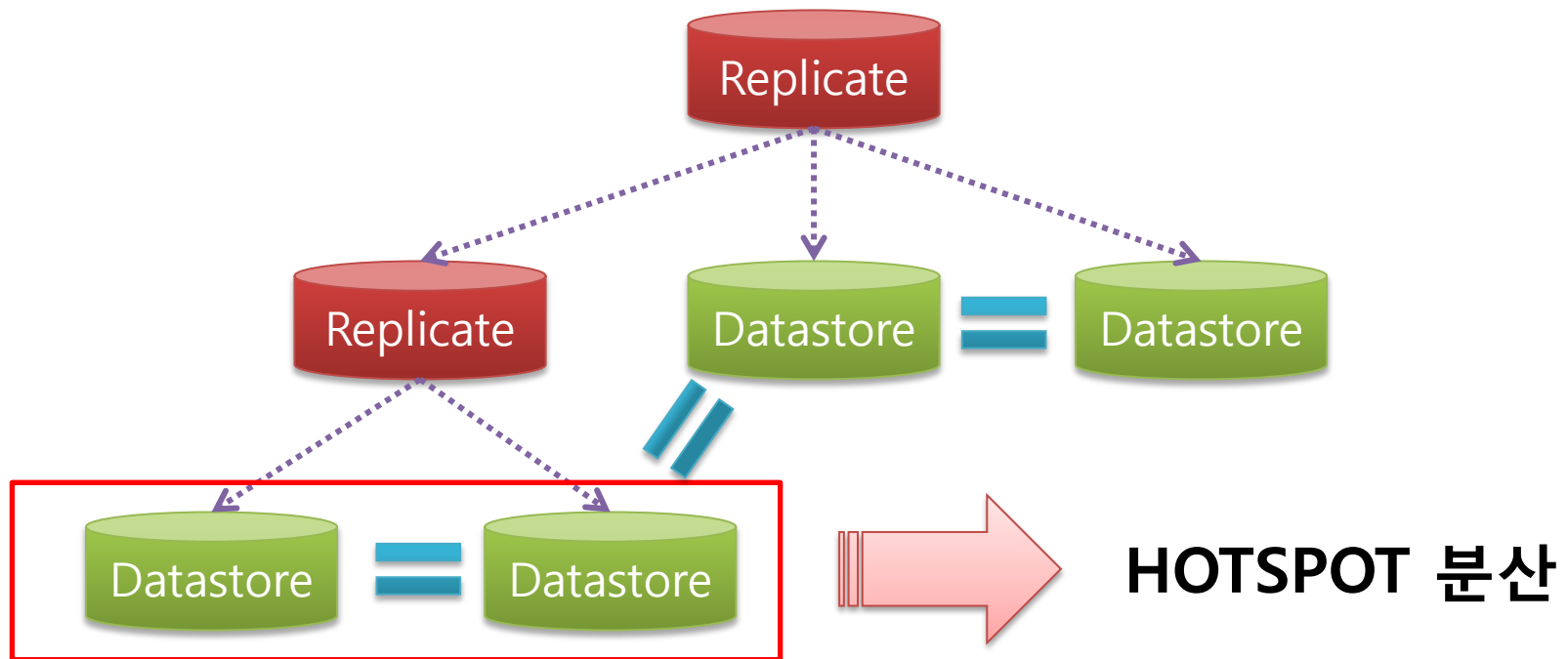
❖ 데이터 �핑 선행 테이블에서 “데이터 조각” 관리

- ✓ 키 값을 Hash 함수 호출 및 숫자로 변환 후 저장될 데이터 구역 결정
- ✓ 서비스 요구사항에 따라 변경하여 데이터 분산 재배치 가능



❖ “Replication Tree”로 “복제” 데이터 관리

- ✓ Replicate Branch는 모든 자식 노드에 쓰기 연산을 반복 수행
- ✓ 서비스 요구사항 데이터 복제 전략 재구성 가능



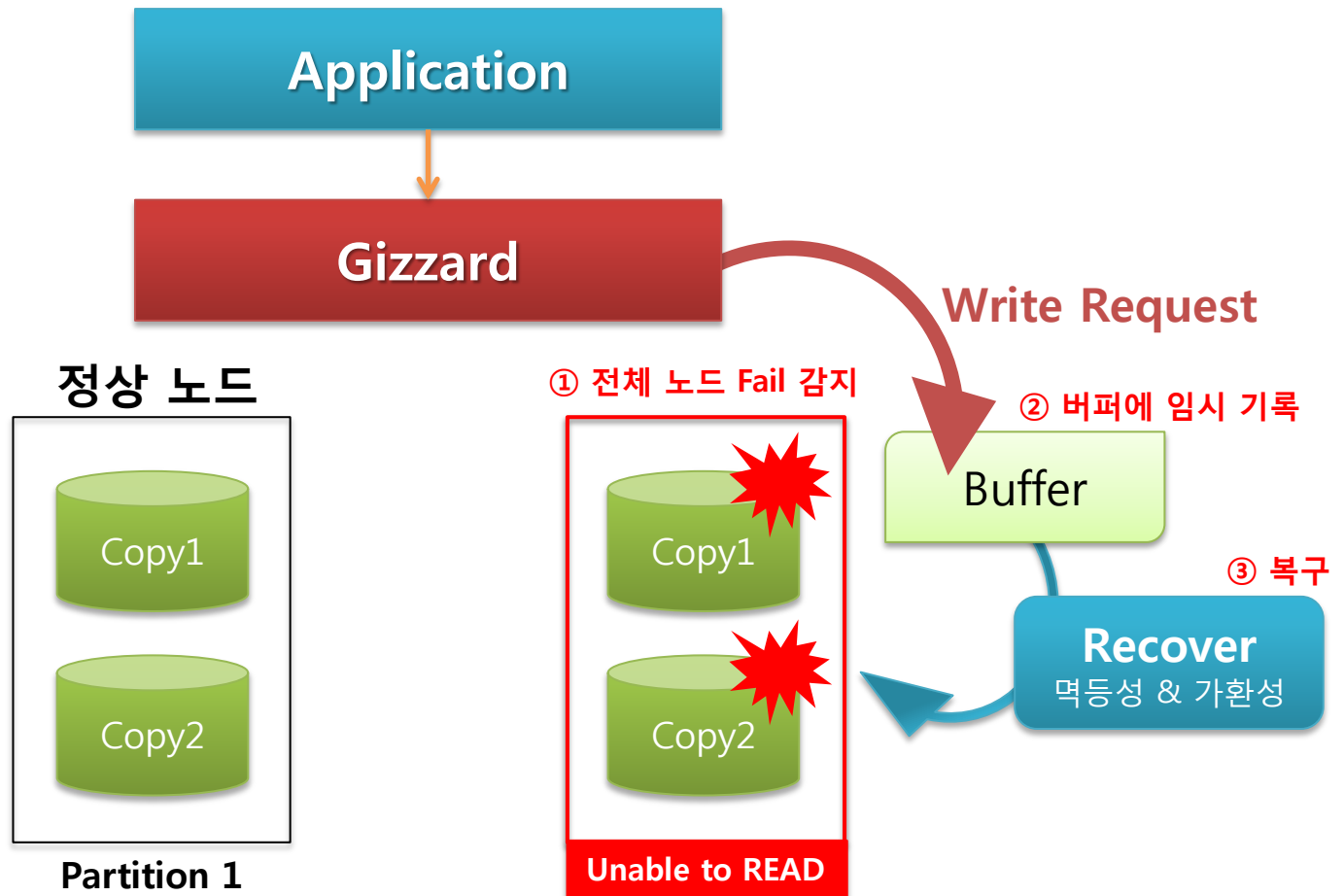
❖ 특정 노드 장애 시 정상 노드에 비 동기로 WRITE

- ✓ 데이터 일관성 및 장애 복구는 멍등성/가환성 원칙에 의거하여 수행
- ✓ 장애 발생 파티션 포함 전체 데이터 READ/WRITE 가능



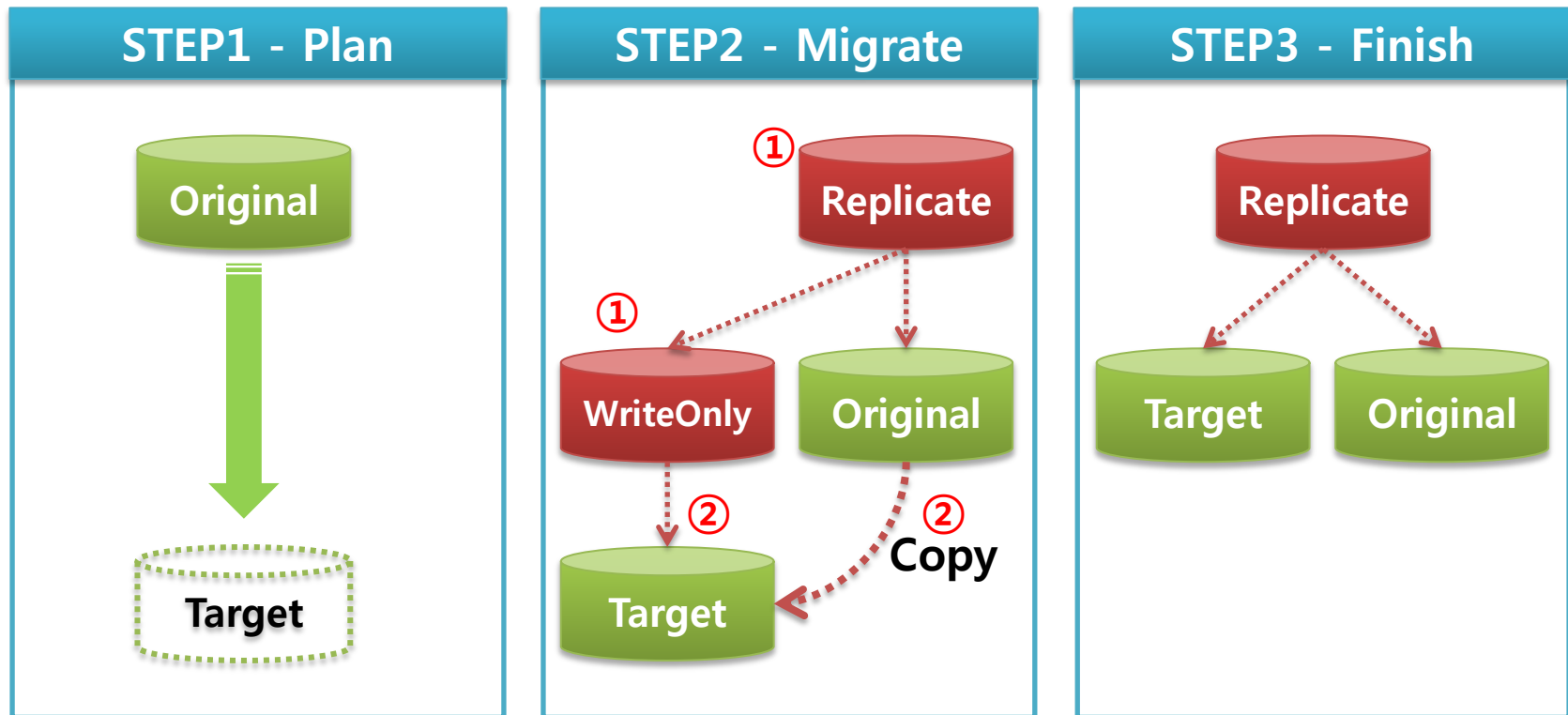
❖ 파티션 내 전체 노드 장애 발생 시 버퍼에 WRITE 수행

- ✓ 데이터 일관성 및 장애 복구는 멱등성/가환성 원칙에 의거하여 수행
- ✓ 장애 발생 파티션 외 다른 파티션은 정상적으로 READ/WRITE 가능



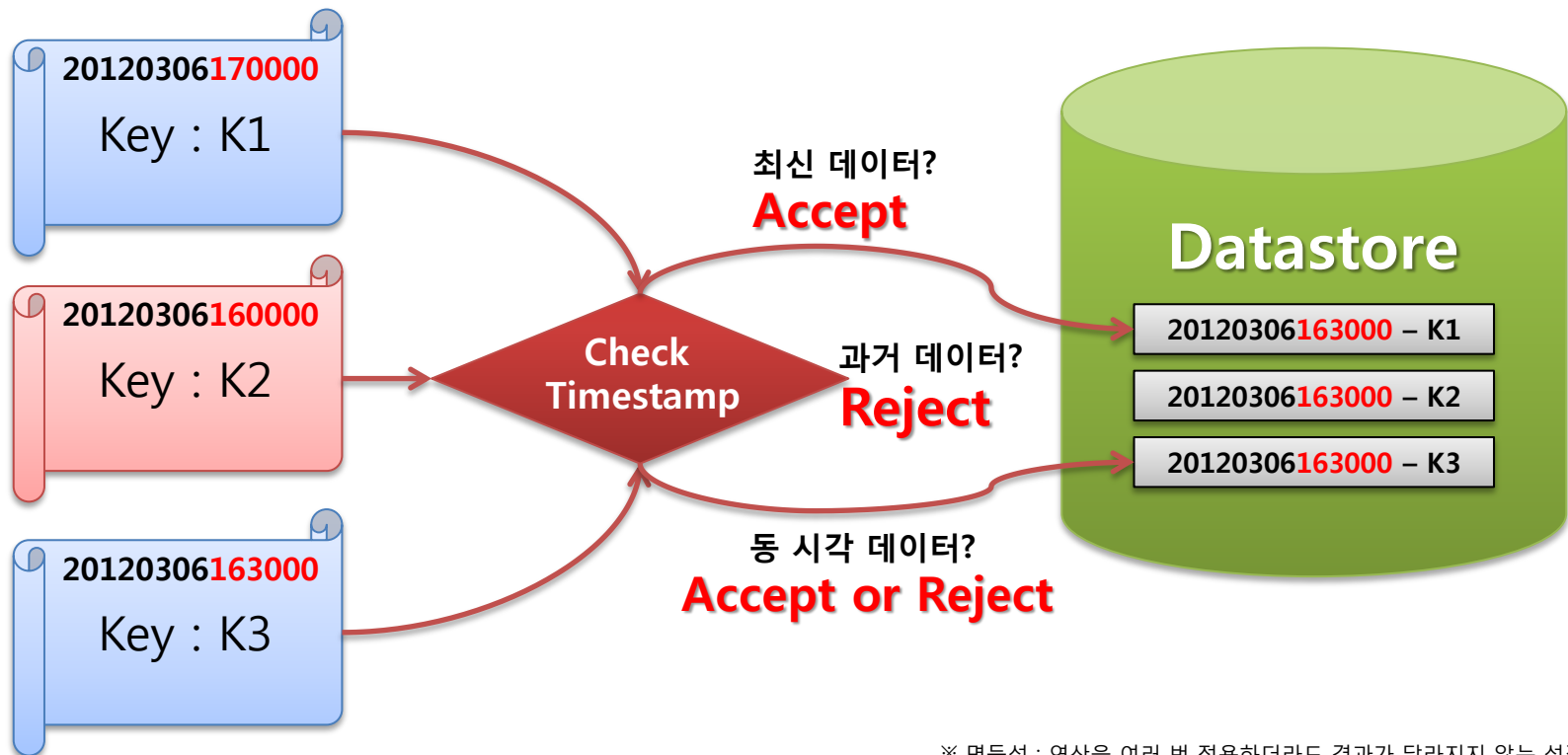
❖ “Replicate”와 “WriteOnly”를 임시 구성하여 데이터 이관

- ✓ 데이터 Copy 시 Write 연산은 멱등성/가환성 원칙에 따라 수행됨
- ✓ WriteOnly Shard는 새로운 Shard가 사용 가능한 시점까지 유지



❖ 데이터 충돌(Conflict)를 멍등성/가환성 원칙으로 해결

- ✓ 데이터가 순서에 따라 적용되는 것을 보장하지 않음
- ✓ Timestamp를 활용하여 신규 데이터 여부를 구분함



※ 멍등성 : 연산을 여러 번 적용하더라도 결과가 달라지지 않는 성질
※ 가환성 : 연산의 순서를 바꾸어도 그 결과가 변하지 않는 성질



COMPARE WITH..



❖ 데이터 적용 순서 보장 여부

- ✓ Gizzard는 데이터가 순서에 따라 적용되는 것을 보장하지 않음
- ✓ MySQL은 순차적으로 데이터가 적용되어야 함 (1 Thread에서 SQL 적용)

	Gizzard Replicate	MySQL Replication
읽기 분산	Data Shard 재구성	Slave Scale-out
쓰기 분산		불가
데이터 분산		Table Partitioning / Replacate_Do_DB
일관성 유지	역등성/가환성	데이터 순차 적용
이중화 구성	Replicate Table 활용	Master-Slave
데이터 이관	Replicate/WriteOnly Shard	Export/Import or Cold Backup
Fail-Over	자동 감지 및 복구	마스터/슬레이브 전환(수동)
데이터처리	Key/Value	SQL
복잡한 질의 처리	불가	가능

❖ 데이터 적용 순서 보장 여부

- ✓ Gizzard는 데이터 일관성을 멍등성/가환성에 의거하여 유지
- ✓ Cassandra는 데이터 일관성을 Gossip 통신과 Timestamp로 최신 데이터 결정

	Gizzard	Cassandra
읽기 분산	Data Shard 재구성	신규 노드 추가 및 Re-Balancing (Memtable/SSTable)
쓰기 분산		
데이터 분산		
일관성 유지	멍등성/가환성	Gossip with Timestamp 데이터 복제를 가져와서 Concurrency Level에 따라 최근 데이터 여부 결정
Fail-Over	자동 감지 및 복구	자동 감지 및 복구
데이터처리	Key/Value	Key/Value
복잡한 질의 처리	불가	불가

THANK YOU!

개발실 / 데이터지능팀 / 성동찬

