

클라우드 엣지 기반 도시교통 브레인

2020.11

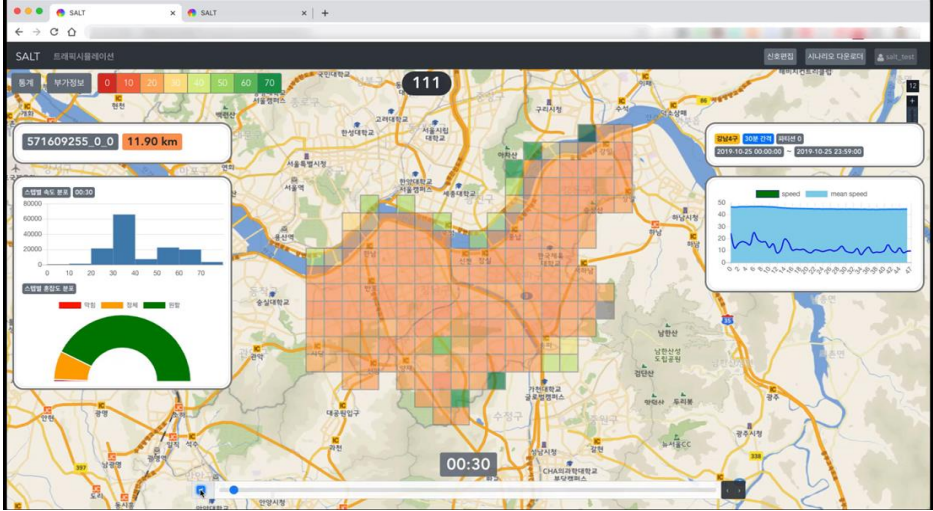
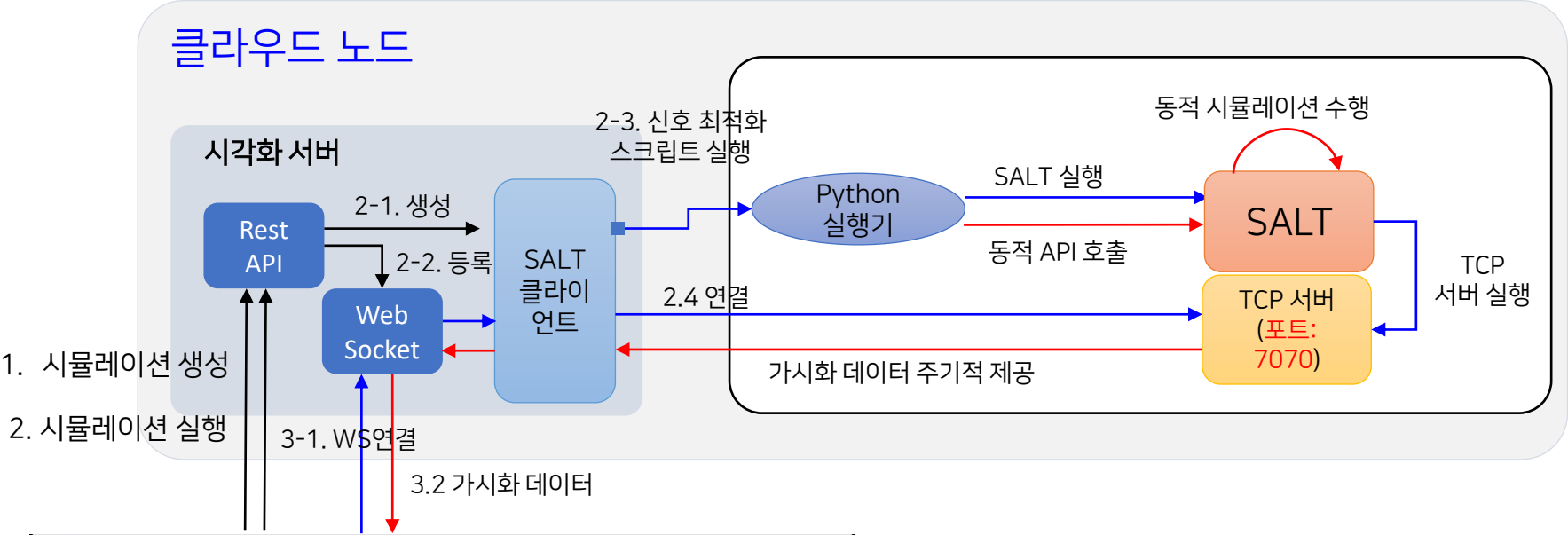
UNIQ



시뮬레이터 연동

COPYRIGHT © 2020. MODUTECH Inc. ALL RIGHT RESERVED.

GUI 기반 동적 시뮬레이션 SW 구조



시뮬레이션 가시화

브라우저 (클라이언트 노드)

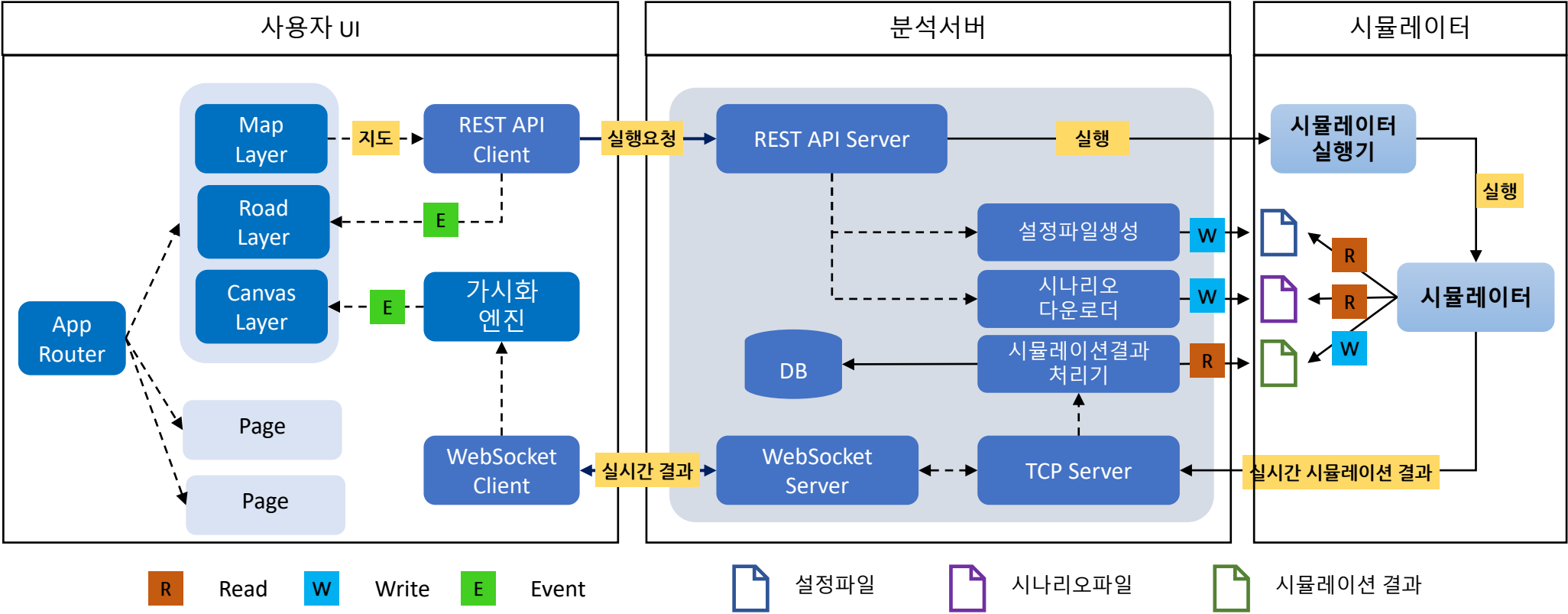
연동구조 구조

- 1차년도
 - 시뮬레이터와 시각화 서버가 같은 호스트에서 구동
 - 시뮬레이션을 위한 설정파일과 결과 파일은 파일 시스템을 통해서 공유
- 2차년도
 - 시뮬레이터가 별도의 VM 에서 구동 될 수 있음
 - 연동을 위해서 SSH 등의 원격접속 프로토콜 혹은 Agent 등의 기능이 필요함

설계원칙

- 시각화 서버가 서버 역할 수행
 - 즉, 항상 시뮬레이터가 접속 할 수 있도록 접속 대기 상태로 있음
- 둘 이상의 시뮬레이터가 동시 접속 가능해야 함
- 둘 이상의 사용자가 동시에 같은 시뮬레이션 결과를 볼 수 있어야 함 → 제공하지 X
 - 시뮬레이션을 등록한 사용자만 실시간 결과를 볼 수 있도록 조치
- 동시에 둘 이상의 시뮬레이션을 동시 실행할 수 있어야 함

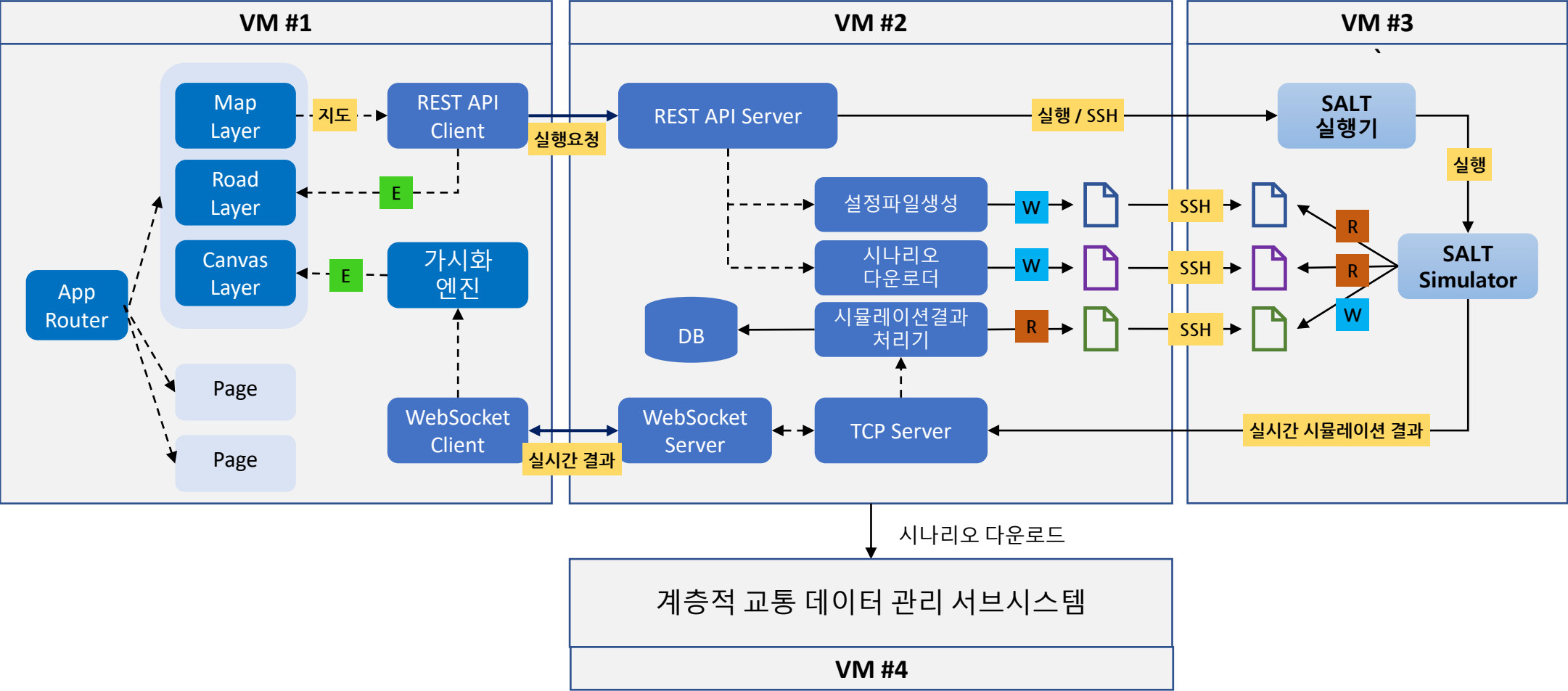
시뮬레이터 연동 구조 (1차년도)



1차년도 연동구조

- 시뮬레이터와 시각화 서버가 동일한 호스트에서 실행 됨
- 로컬 파일 시스템에 있는 시뮬레이션 실행 스크립트를 실행
- 시뮬레이션 결과 역시 로컬 파일 시스템에 저장

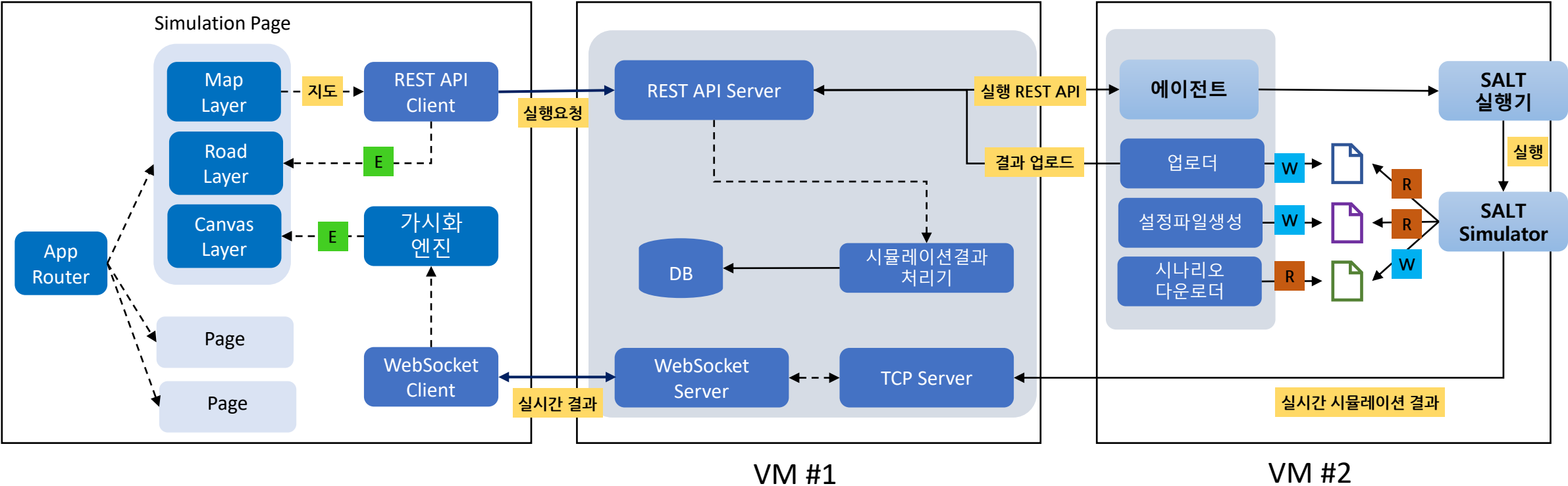
시뮬레이터 연동 구조 (2차년도) – SSH 기반



2차년도 구조

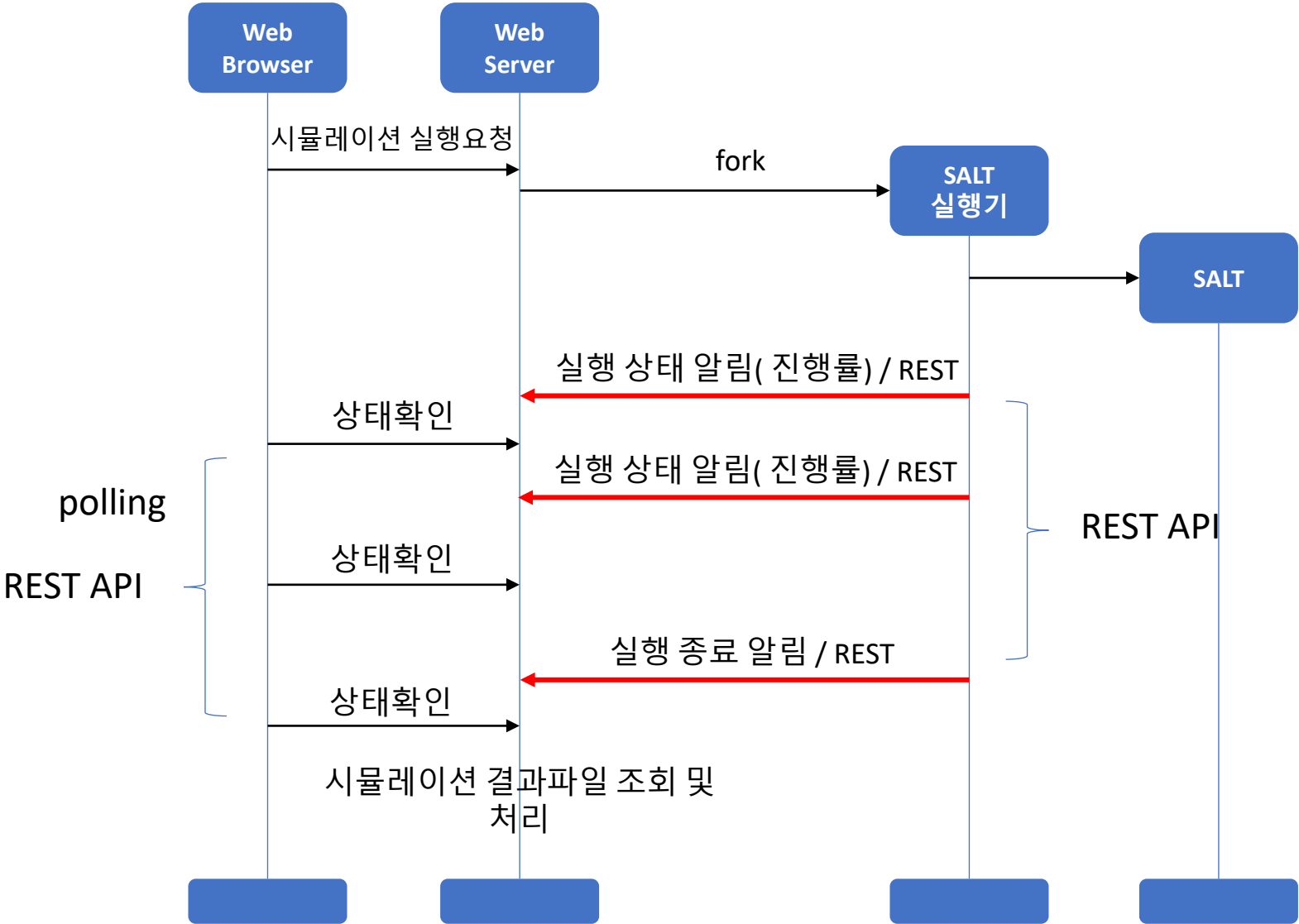
- 시뮬레이터가 분석서버와 다른 호스트에서 구동 될 수 있음
- SSH 또는 전용 에이전트를 이용할 수 있음

시뮬레이터 연동 구조 (2차년도) – Agent 기반

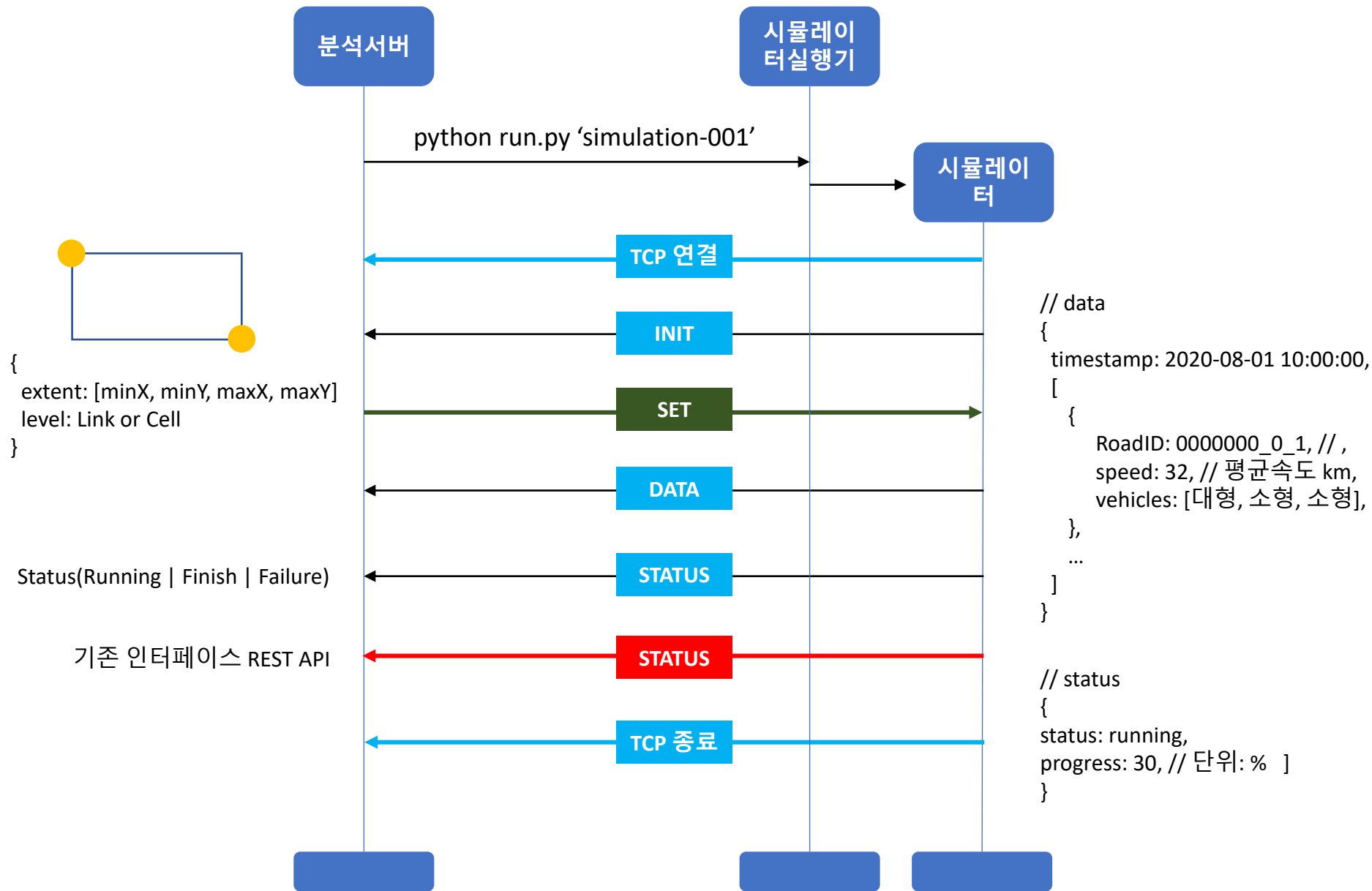


에이전트가 가시화 서버의 일부 기능을 대리해서 처리
시뮬레이션 종료는 프로세스 중지 시그널을 받았을 때
시뮬레이션이 종료되면 결과 파일을 가시화 서버로 업로드

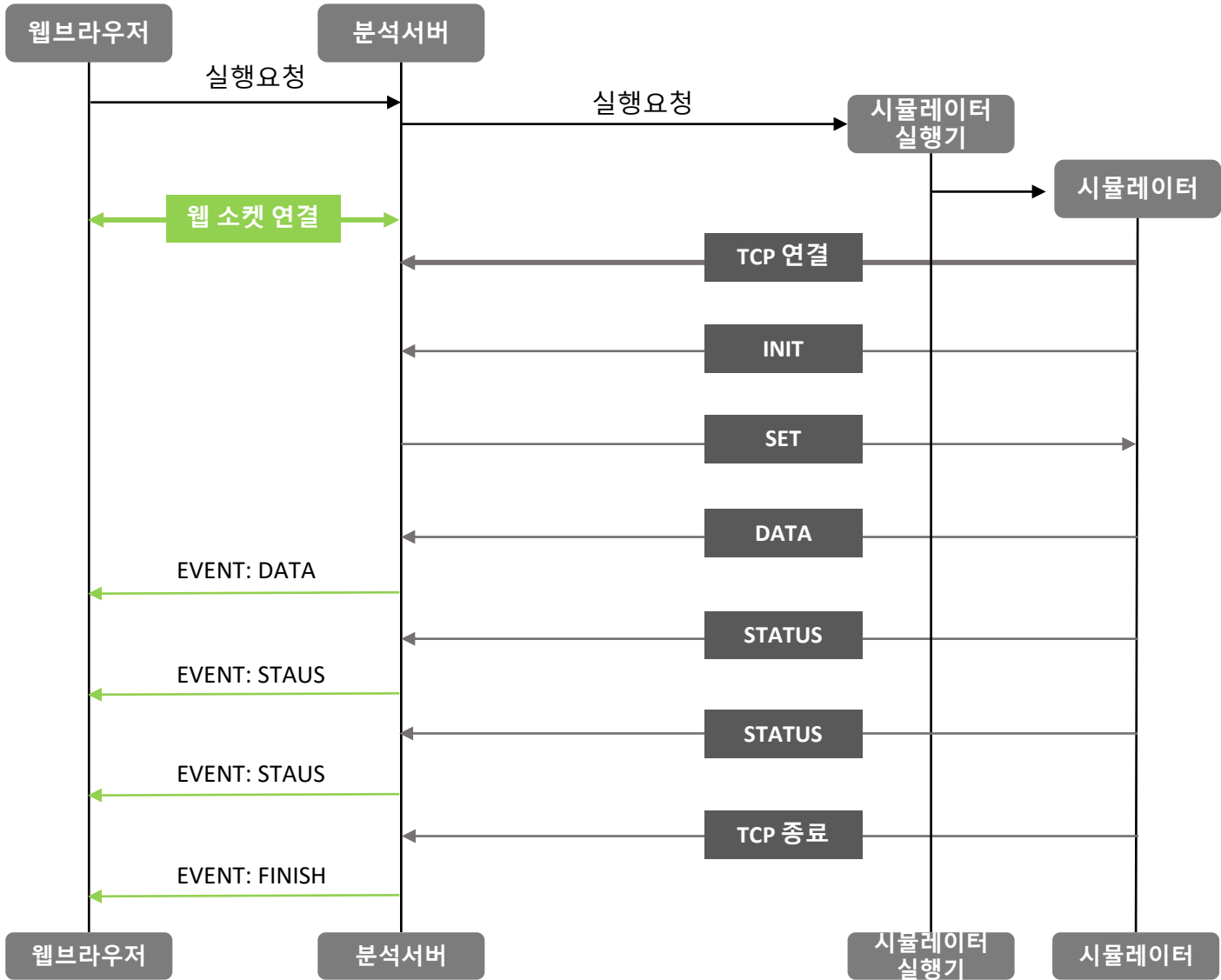
시각화서버 ↔ SALT 연동(기존방식)



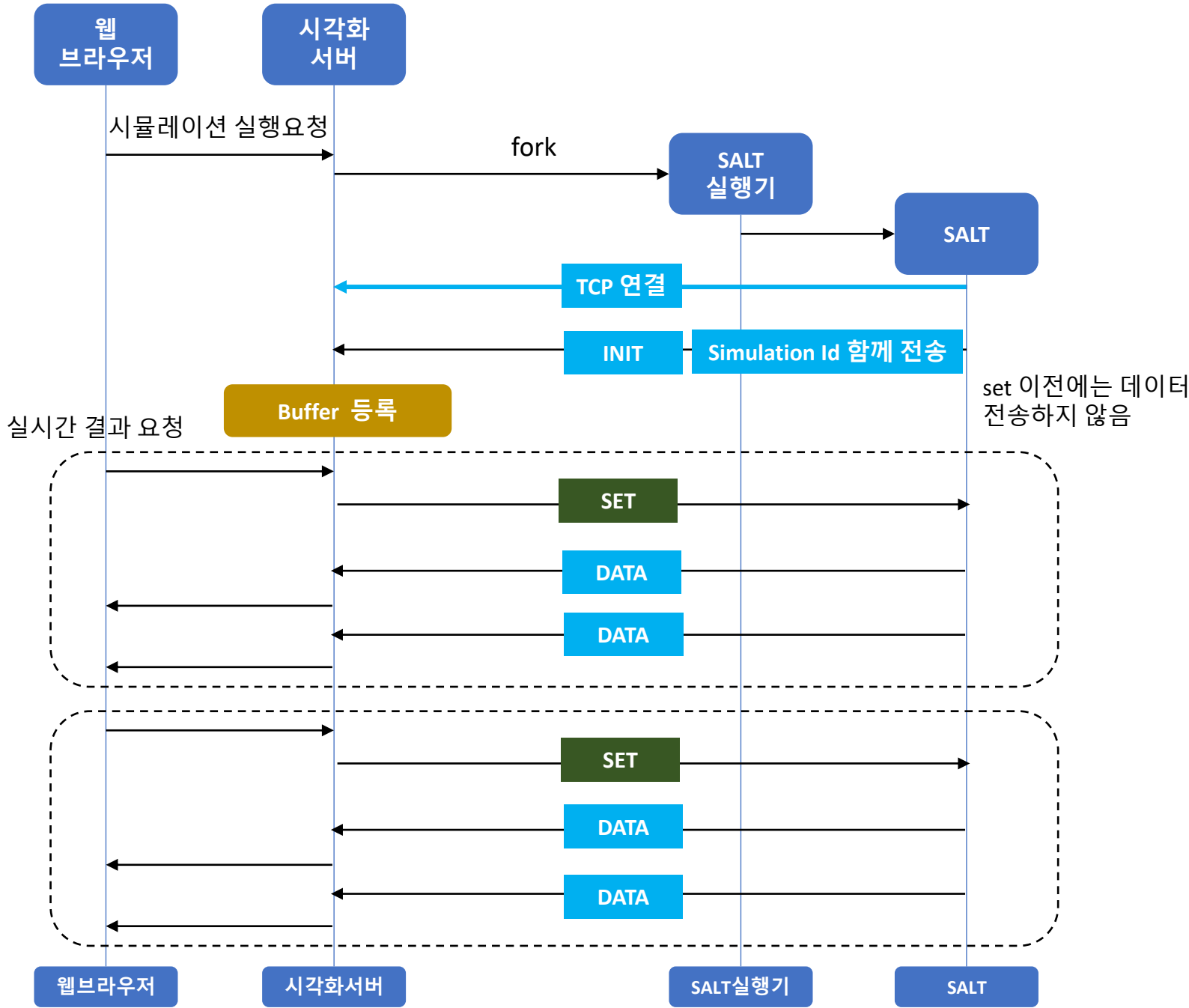
시각화서버 ↔ SALT 연동(신규방식)



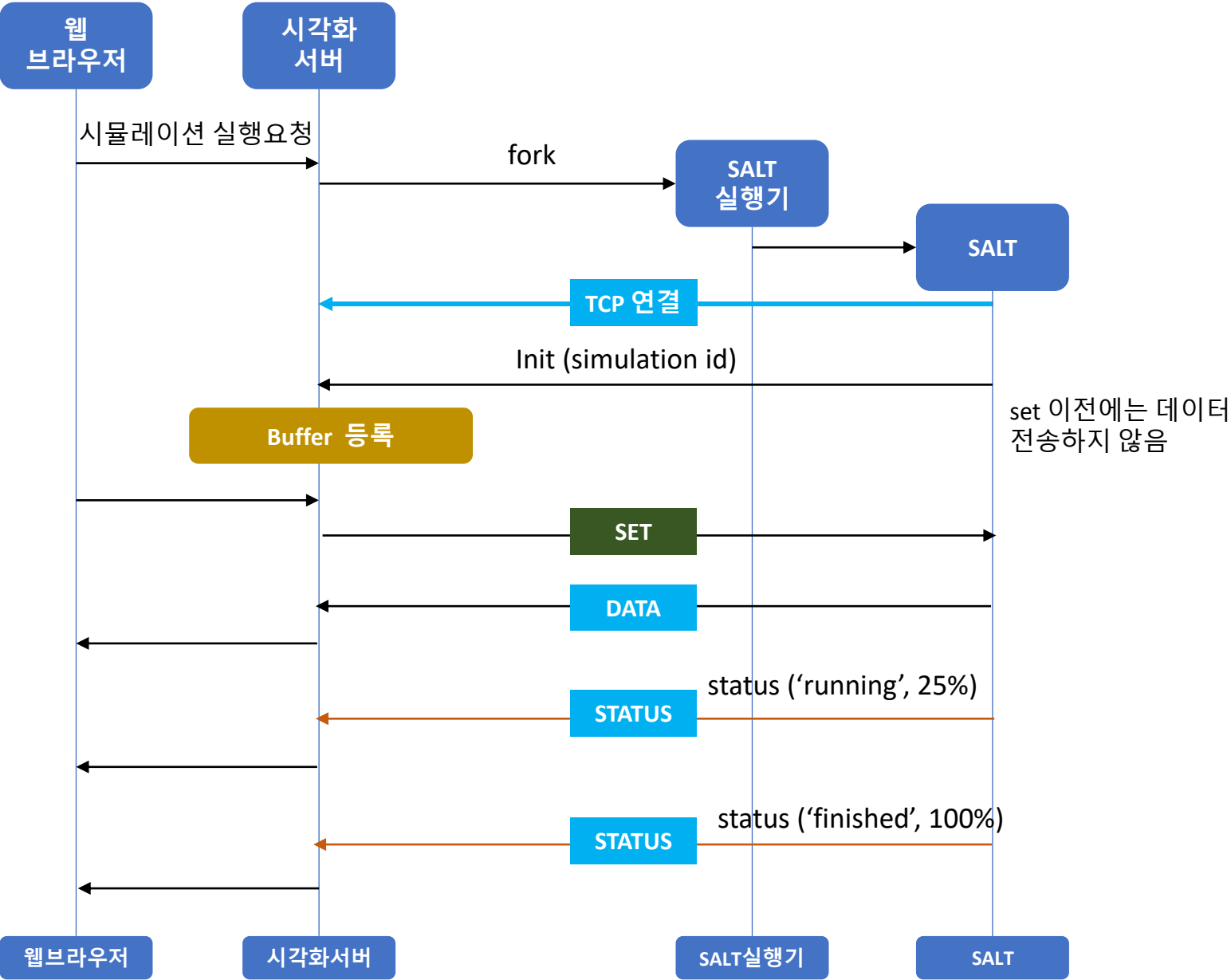
시각화서버 ↔ SALT 연동(신규방식)



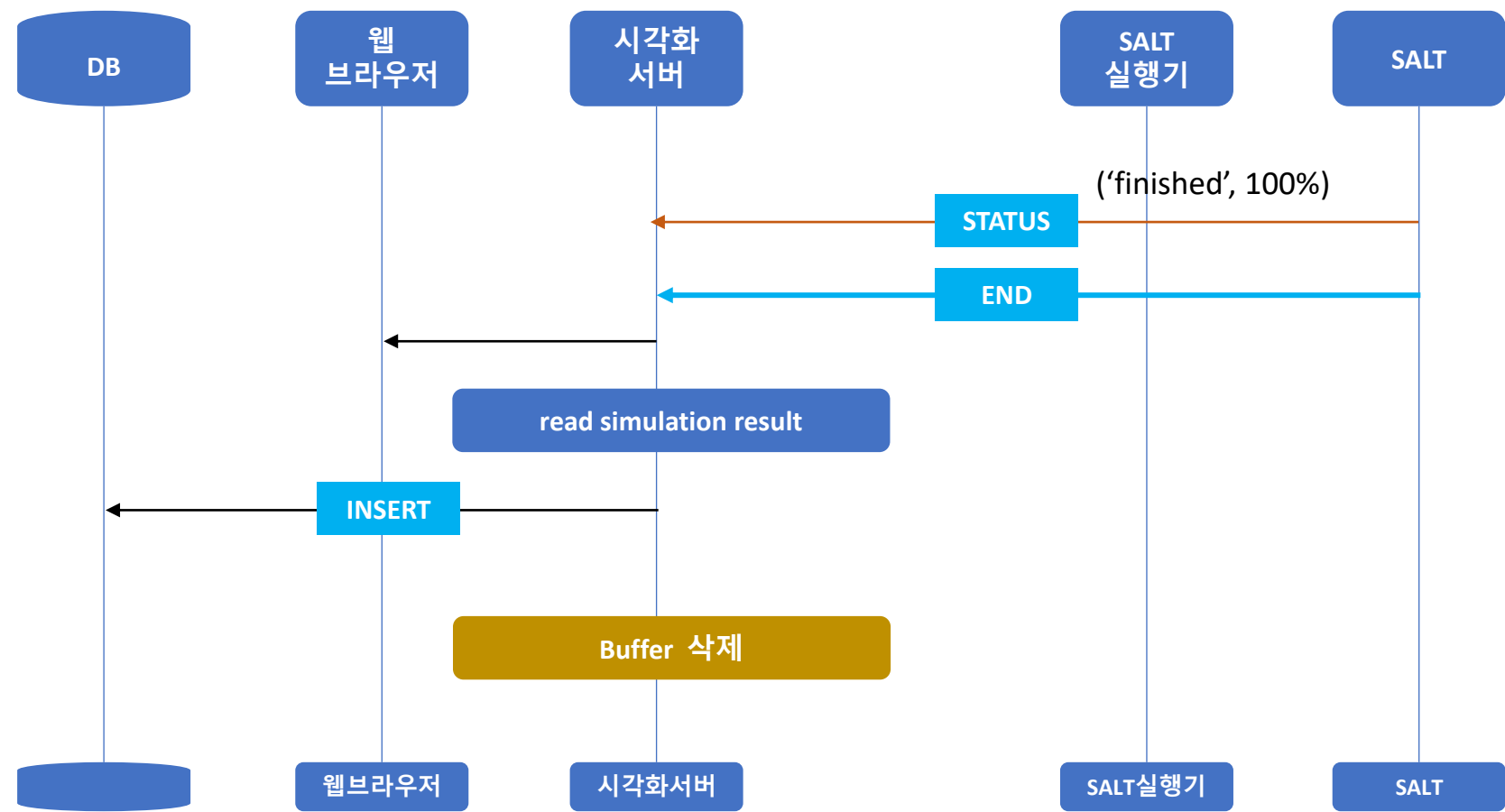
실시간 데이터 전송



실시간 데이터 전송



실시간 데이터 전송



실시간 시뮬레이션 상태 전송을 위한 메시지 구조

시각화 서버와 시뮬레이터는 TCP 를 이용하여 메시지를 통신 함
분석서버는 특정 포트에서 대기
동시에 둘 이상의 시뮬레이터가 접속할 수 있어야 함


```
// protocol message between SALT and SALT-VIS
struct MsgHeader {
    unsigned int type; // INIT(0), DATA(1), STATUS(2), SET(10), STOP(11)
    long timestamp; // milliseconds
}

// SALT-VIS -> SALT
struct SetMsg {
    MsgHeader header;
    float extend[4]; // minx, miny, maxx, maxy
    unsigned int roadType; // 0: link, 1: cell
}

// SALT-VIS --> SALT
struct StopMsg {
    MsgHeader header;
}
```

분석 서버에서 시뮬레이터로 전송하는 메시지

```
// SALT-VIS --> SALT
struct InitMsg {
    MsgHeader header;
    unsigned char[24] simulationId;
}

// SALT --> SALT-VIS
struct StatusMsg {
    MsgHeader header;
    unsigned int status; // RUNNING(0), FINISHED(1), FAILURE(2)
    unsigned int progress; // 0 ~ 100
}
```

시뮬레이터에서 분석 서버로 전송하는 메시지
제일 먼저 InitMsg 를 보내야 만 분석 서버에서 시뮬레이터를 구분할 수 있음
1차년도에는 인증 메커니즘은 사용하지 x

```
// SALT --> SALT-VIS
struct DataMsg {
    MsgHeader header;
    unsigned int numRoads;
    Road roads[numRoads]
}

struct Road {
    unsigned char roadId[16]; // cellId or linkId
    unsigned int speed; // road's mean speed
    unsigned int numVehicles;
    unsigned char vehicles[numVehicles]; // SMALL(0), LARGE(1)
    unsigned char currentSignal; // RED(0), GREEN(1)
}
```

Road 는 Link 와 Cell 로 구분됨

roadId: cellId 와 linkId 길이가 다름, 길이정보 포함해야 하나?

연동을 위한 이벤트

