

UML 다이어그램 설계 문서

개요

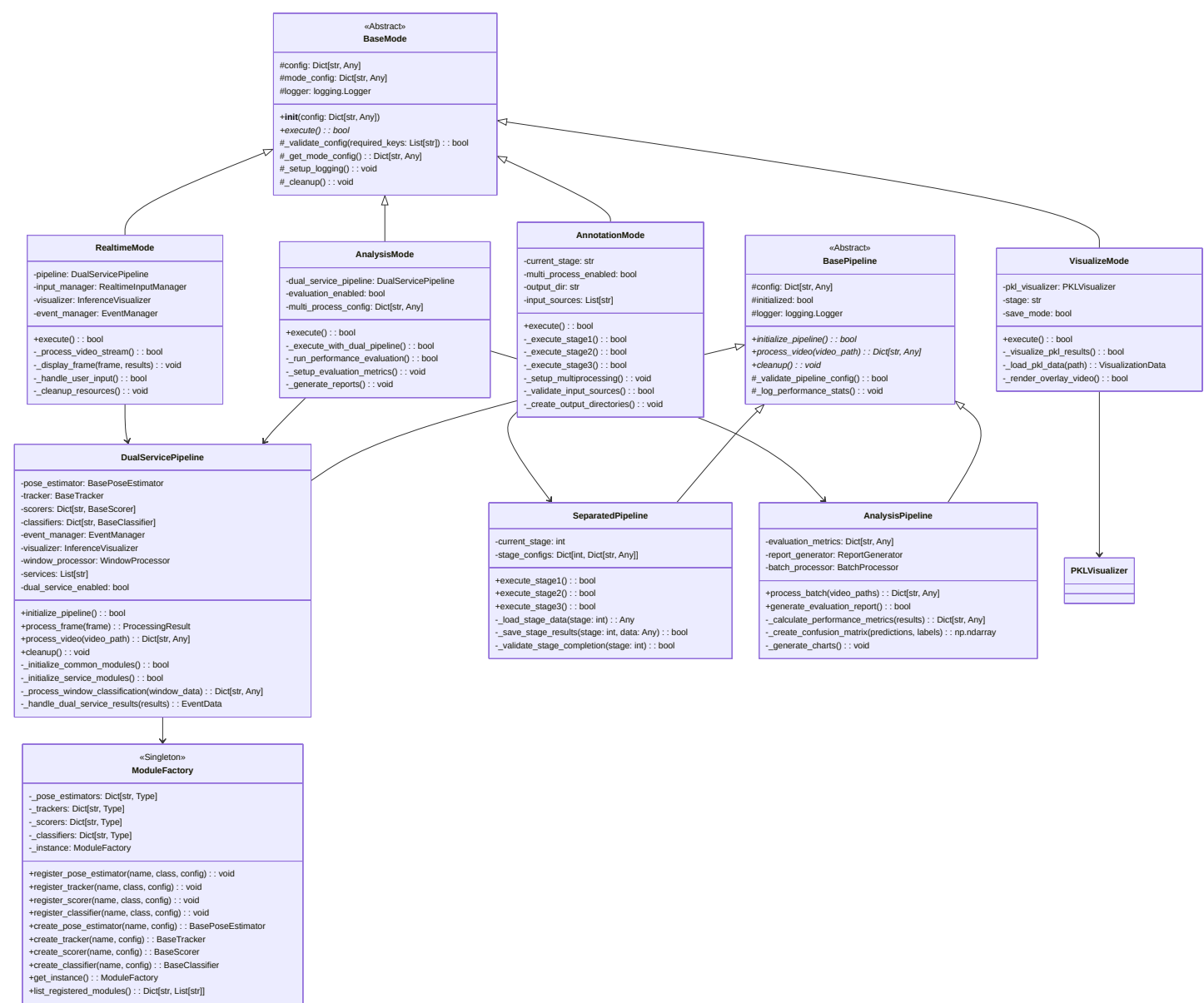
Recognizer 시스템의 포괄적인 UML 다이어그램 모음으로, 시스템의 구조적 관계, 행위적 상호작용, 데이터 플로우를 시각화합니다. 모든 다이어그램은 Mermaid 형식으로 작성되어 GitHub에서 바로 렌더링됩니다.

목차

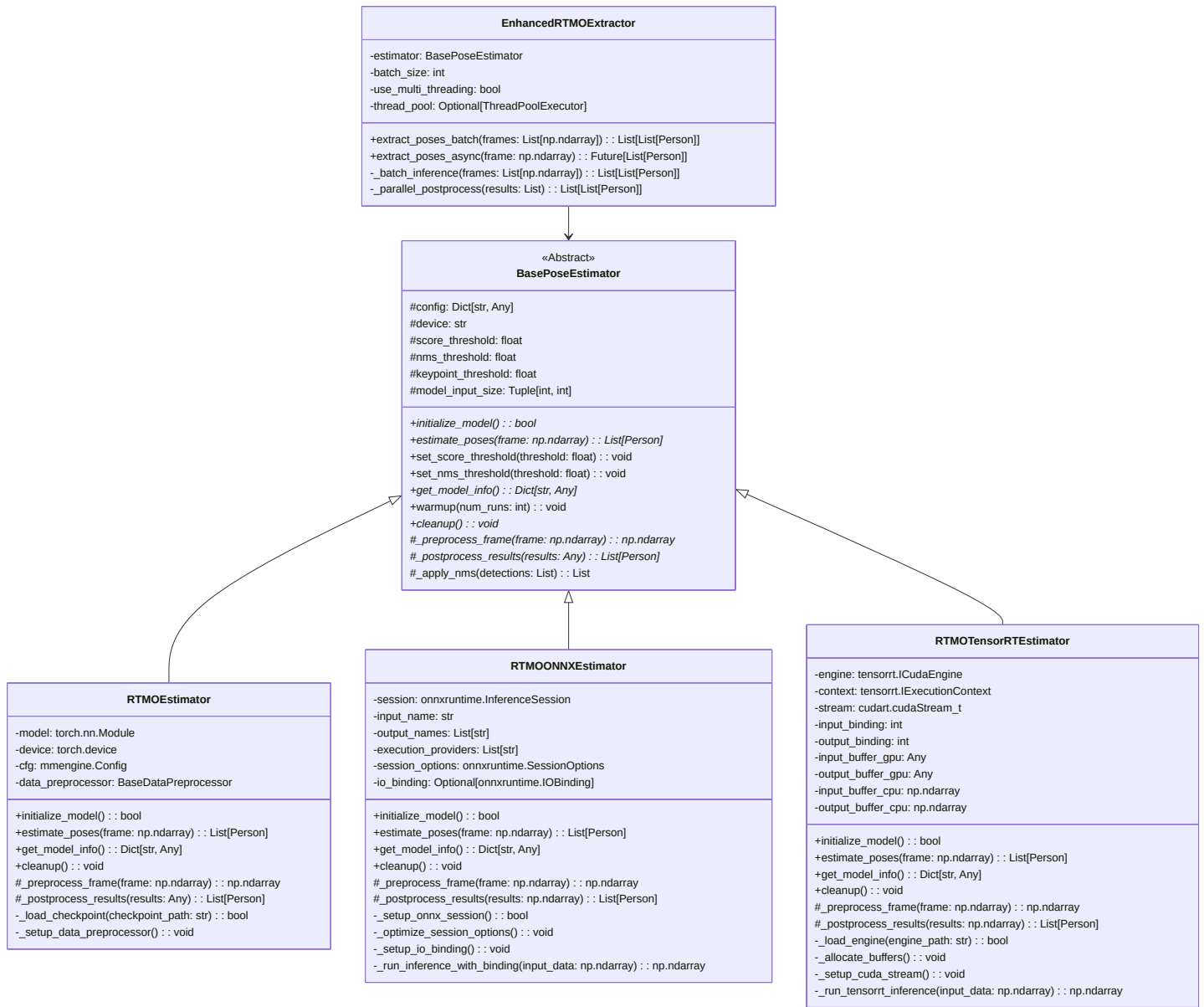
1. 클래스 다이어그램
2. 시퀀스 다이어그램
3. 상태 다이어그램
4. 컴포넌트 다이어그램
5. 배포 다이어그램
6. 액티비티 다이어그램
7. 유스케이스 다이어그램

클래스 다이어그램

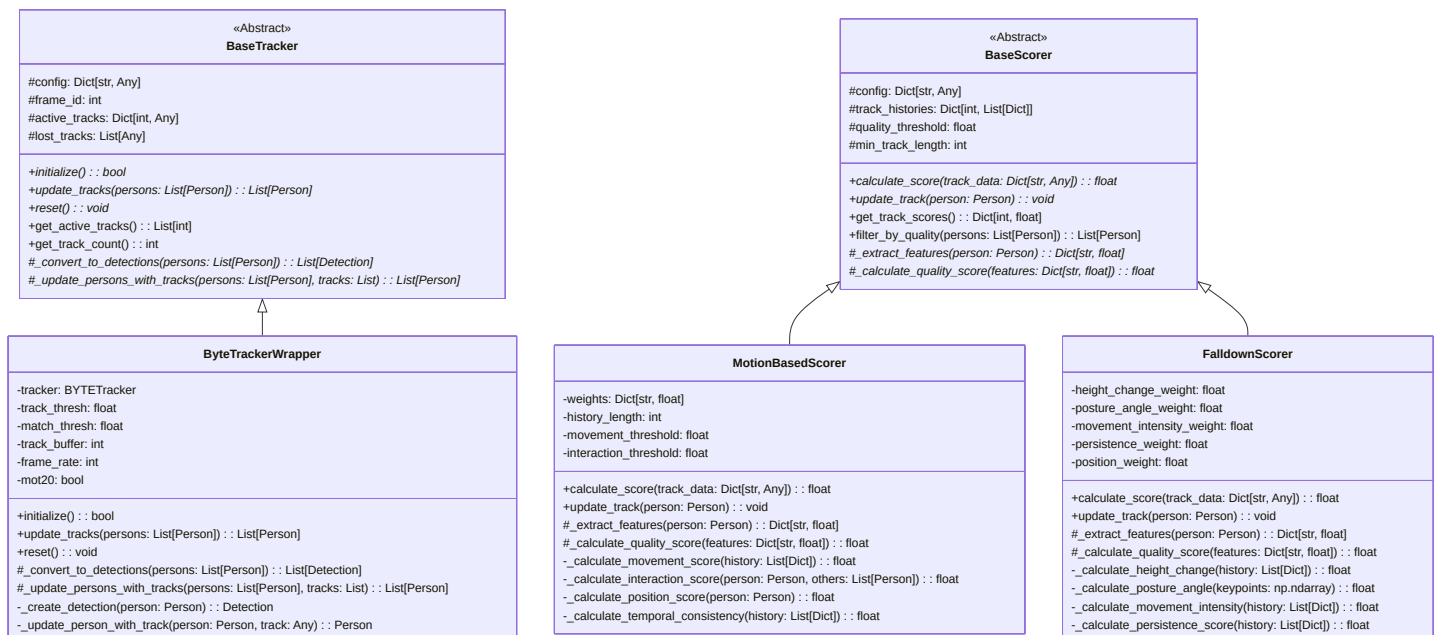
1. 전체 시스템 클래스 구조



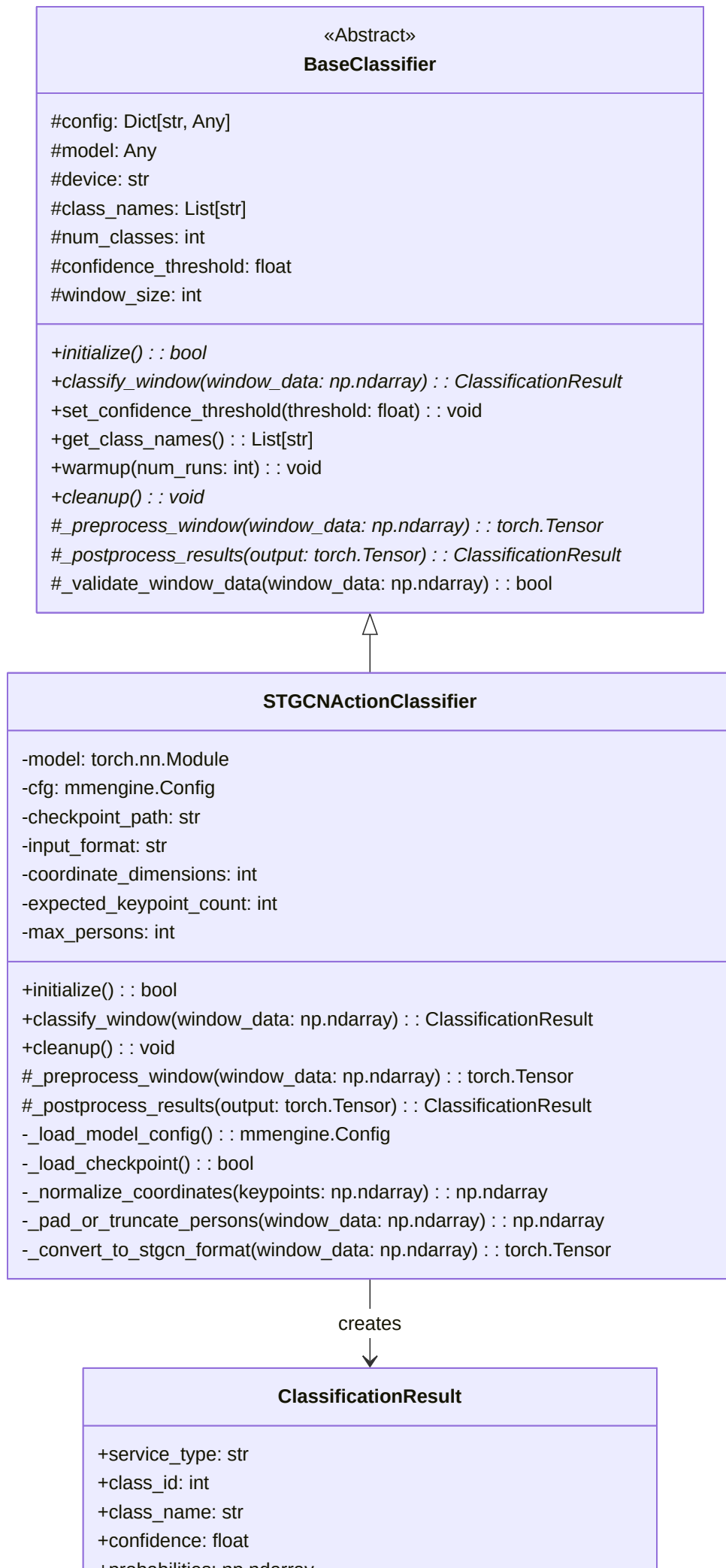
2. 포즈 추정 모듈 클래스 구조



3. 추적 및 스코어링 모듈 클래스 구조



4. 동작 분류 모듈 클래스 구조



+probabilities: np.ndarray

+timestamp: float

+frame_range: Tuple[int, int]

+window_id: int

+processing_time: float

+metadata: Dict[str, Any]

+init(...)

+is_valid() : : bool

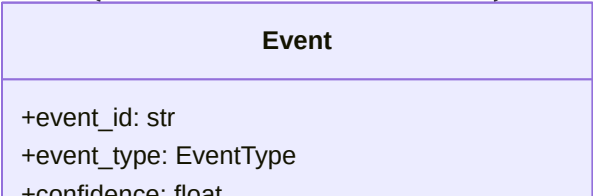
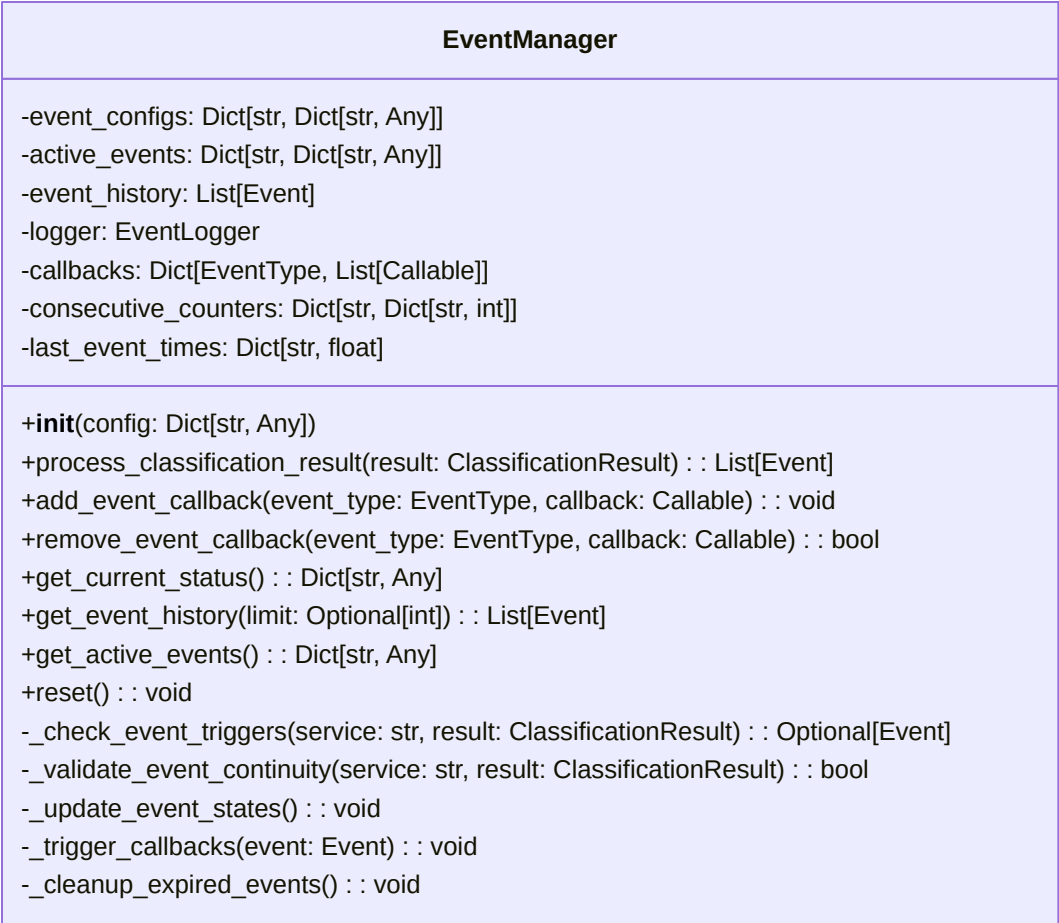
+get_max_confidence() : : float

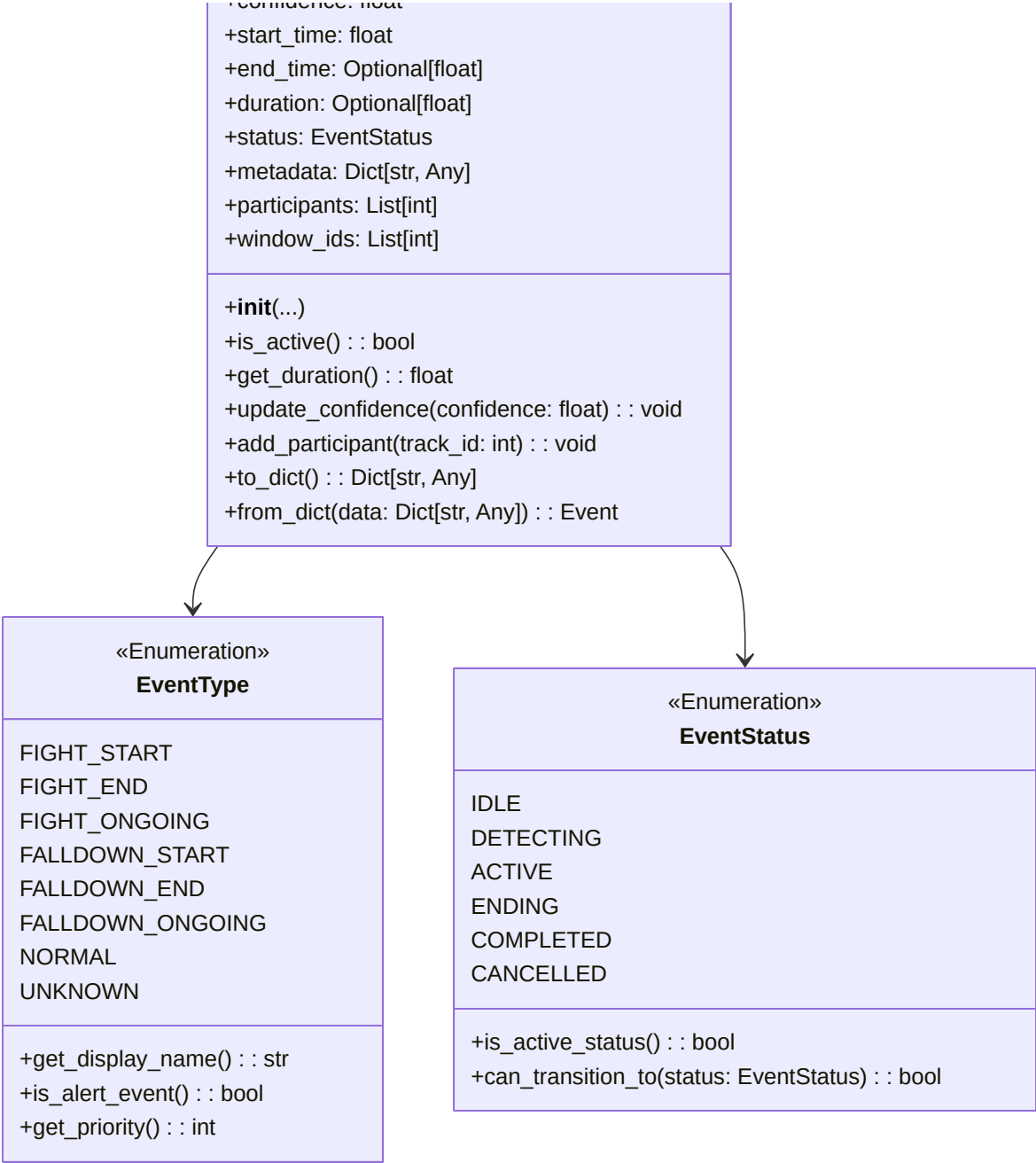
+get_predicted_class() : : str

+to_dict() : : Dict[str, Any]

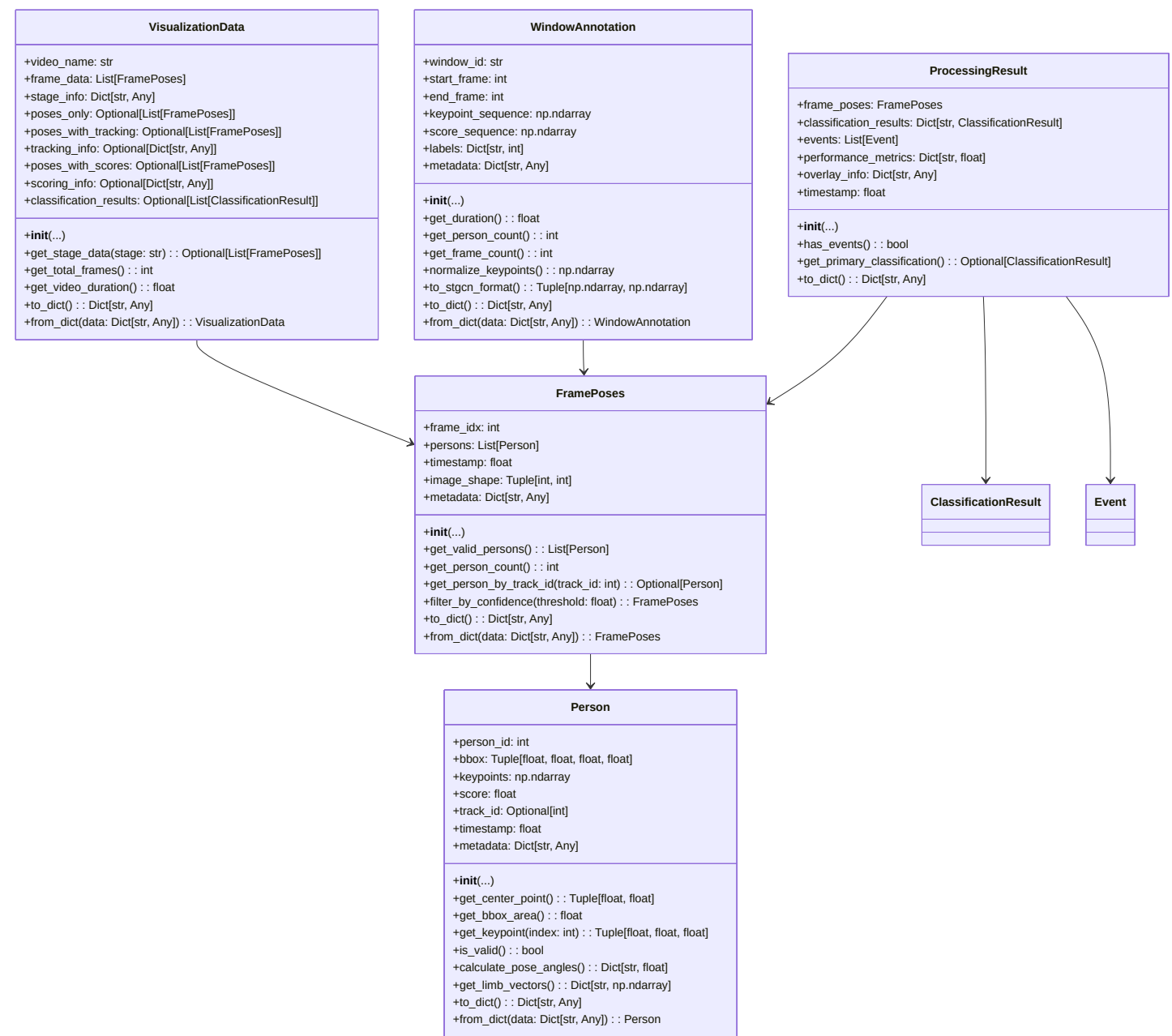
+from_dict(data: Dict[str, Any]) : : ClassificationResult

5. 이벤트 관리 시스템 클래스 구조

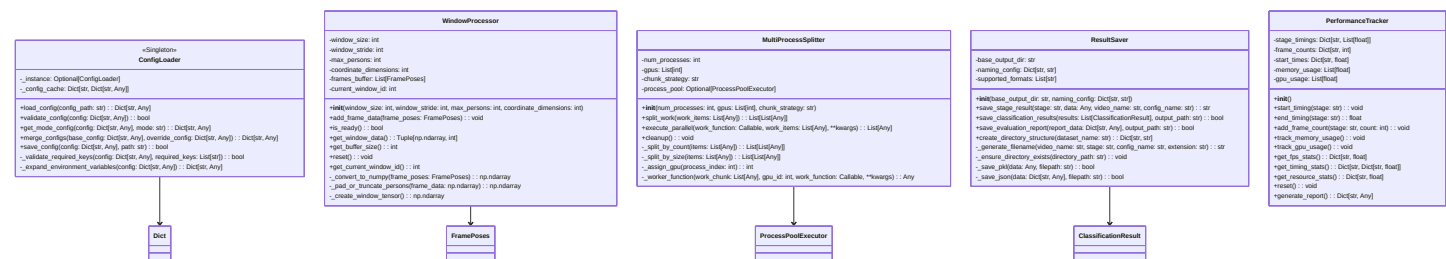




6. 데이터 구조 클래스 다이어그램

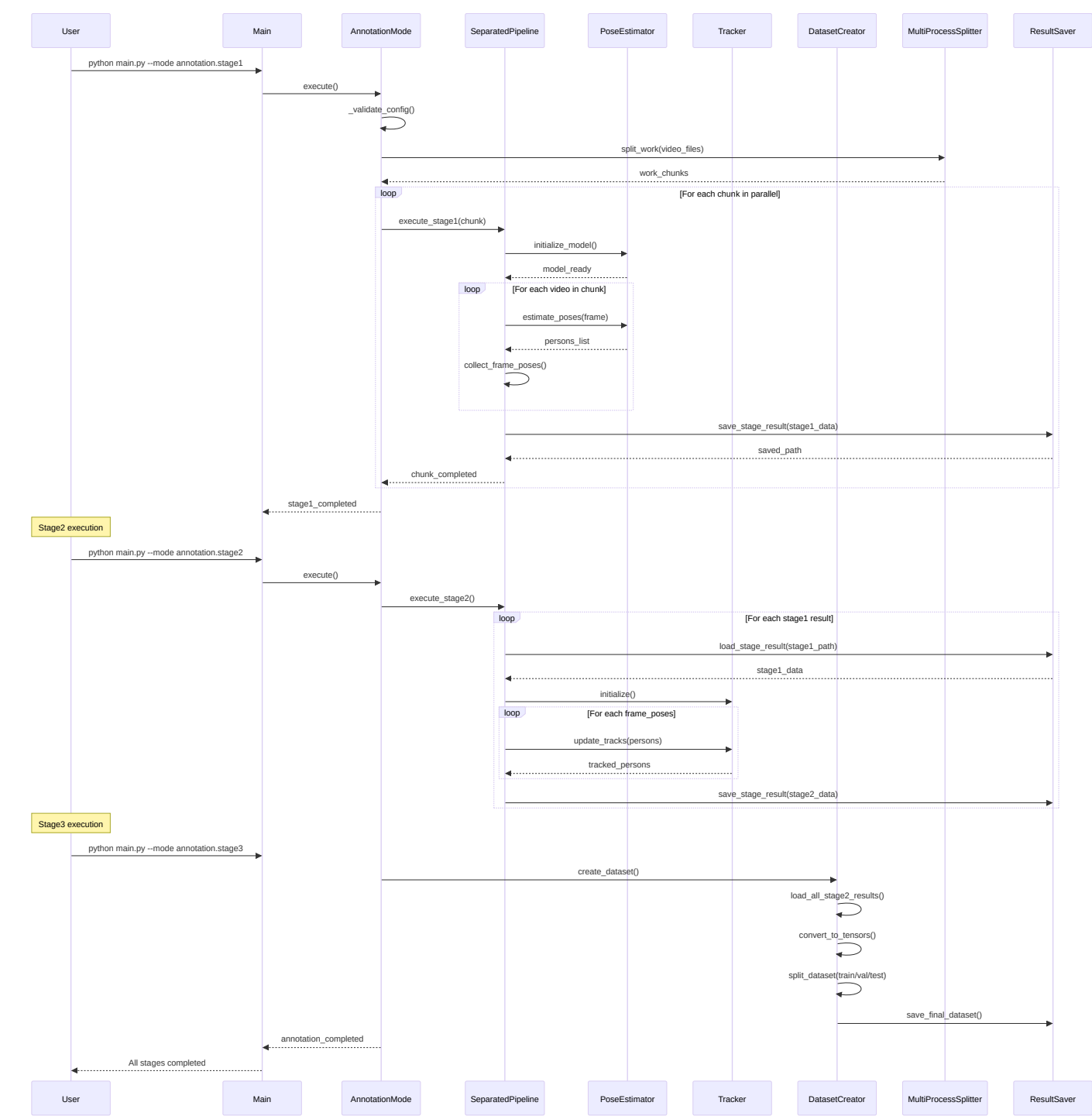


7. 유틸리티 클래스 다이어그램



시퀀스 다이어그램

1. Annotation 파이프라인 실행 시퀀스

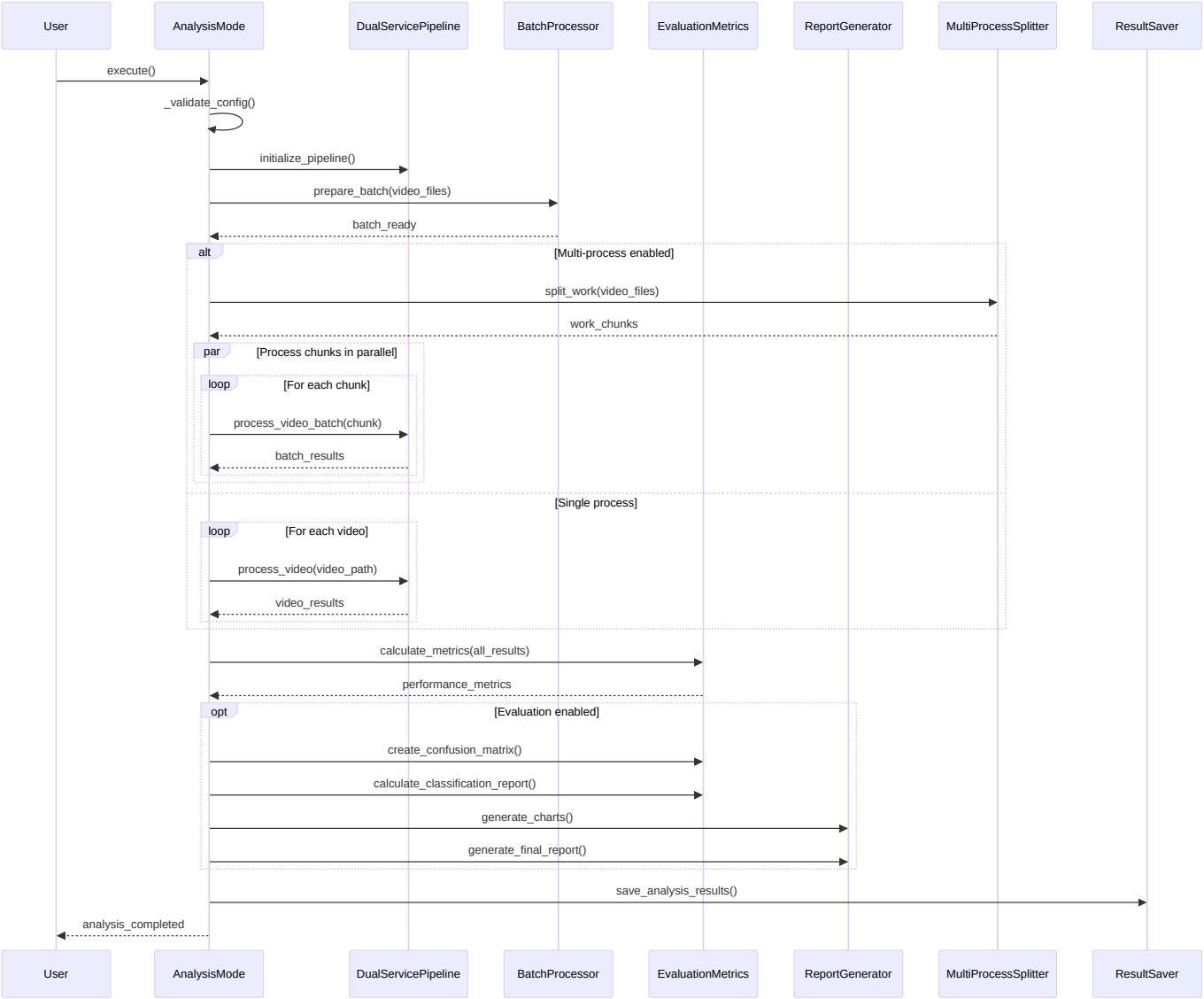


2. 실시간 추론 파이프라인 시퀀스

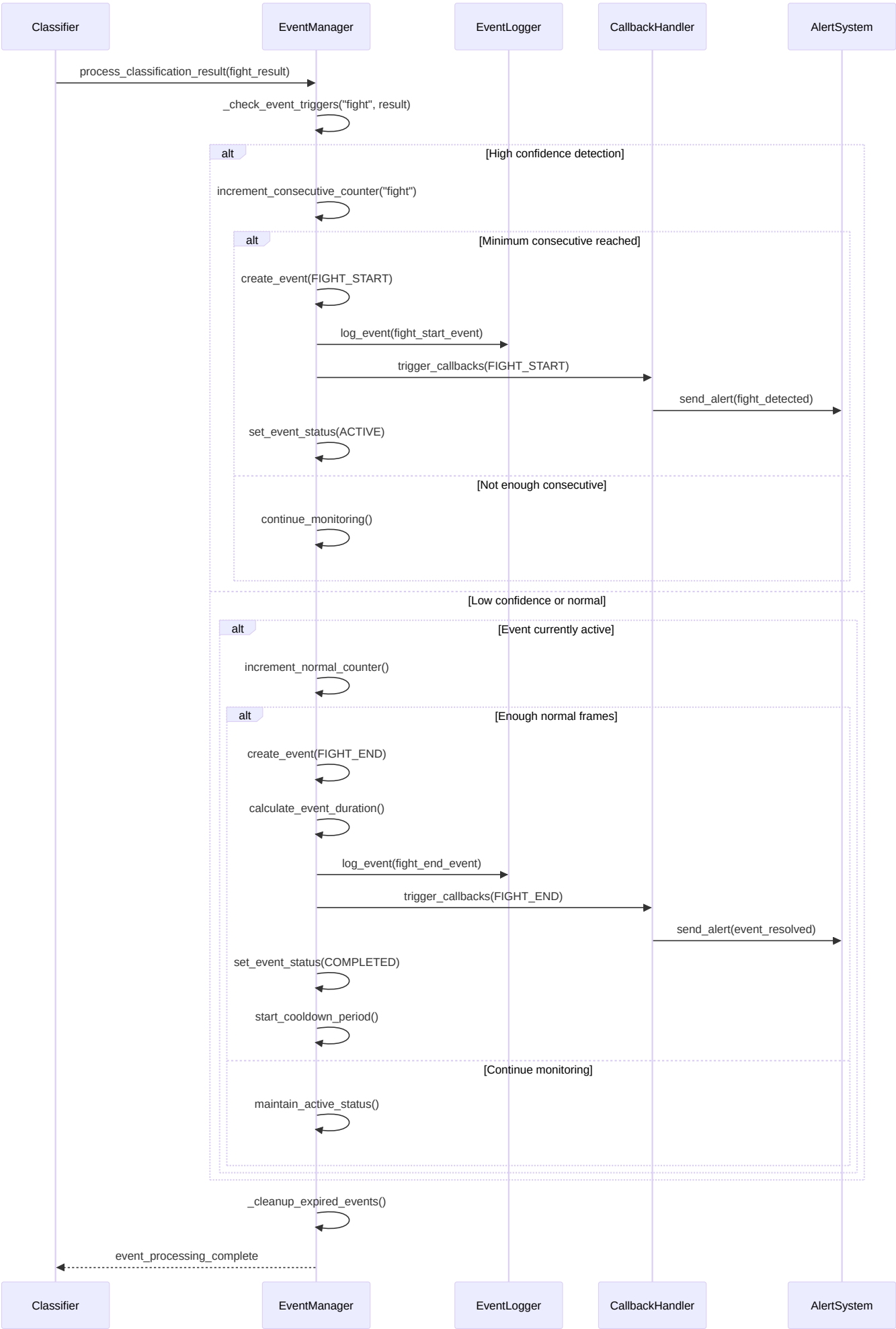


Syntax error in text
mermaid version 11.9.0

3. 분석 모드 배치 처리 시퀀스

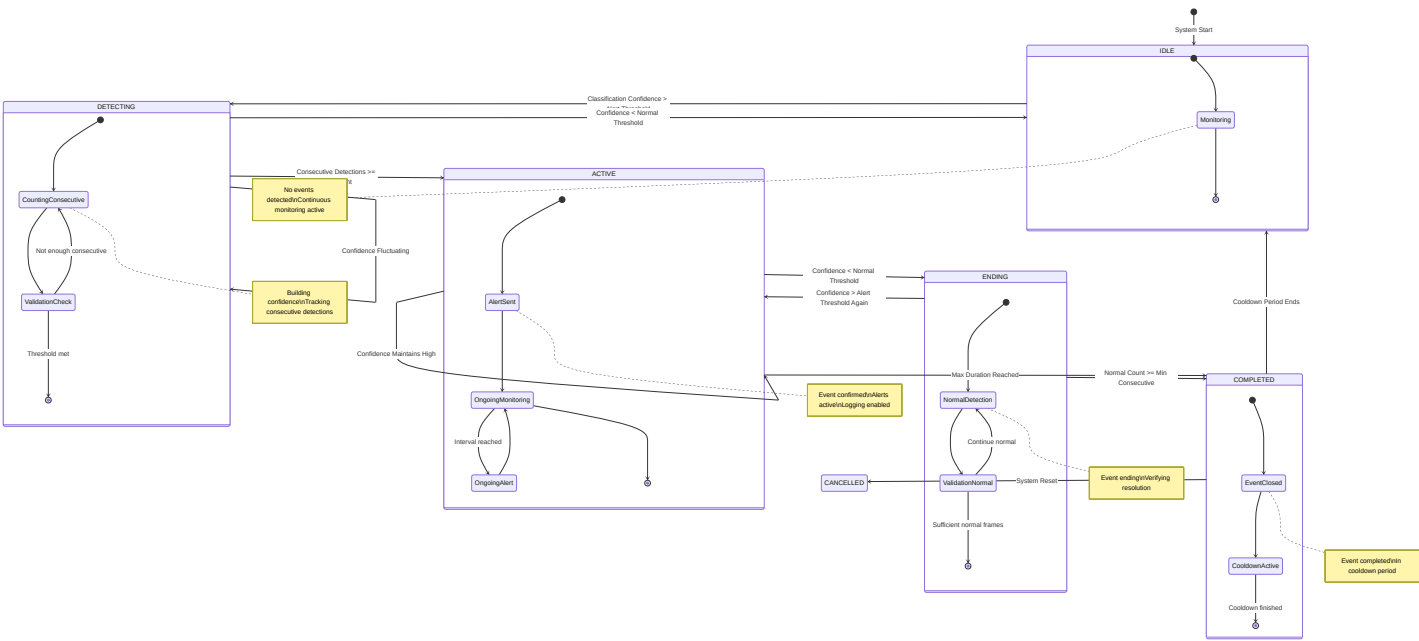


4. 이벤트 관리 상세 시퀀스

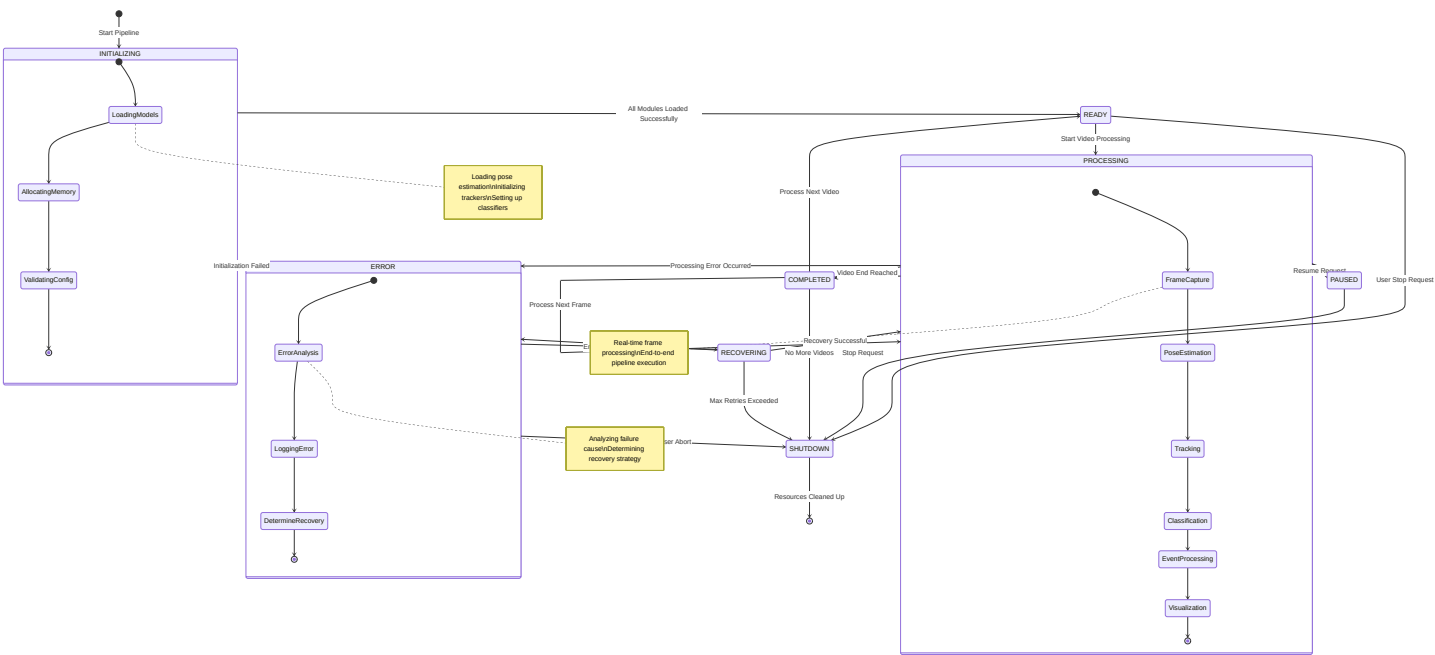


상태 다이어그램

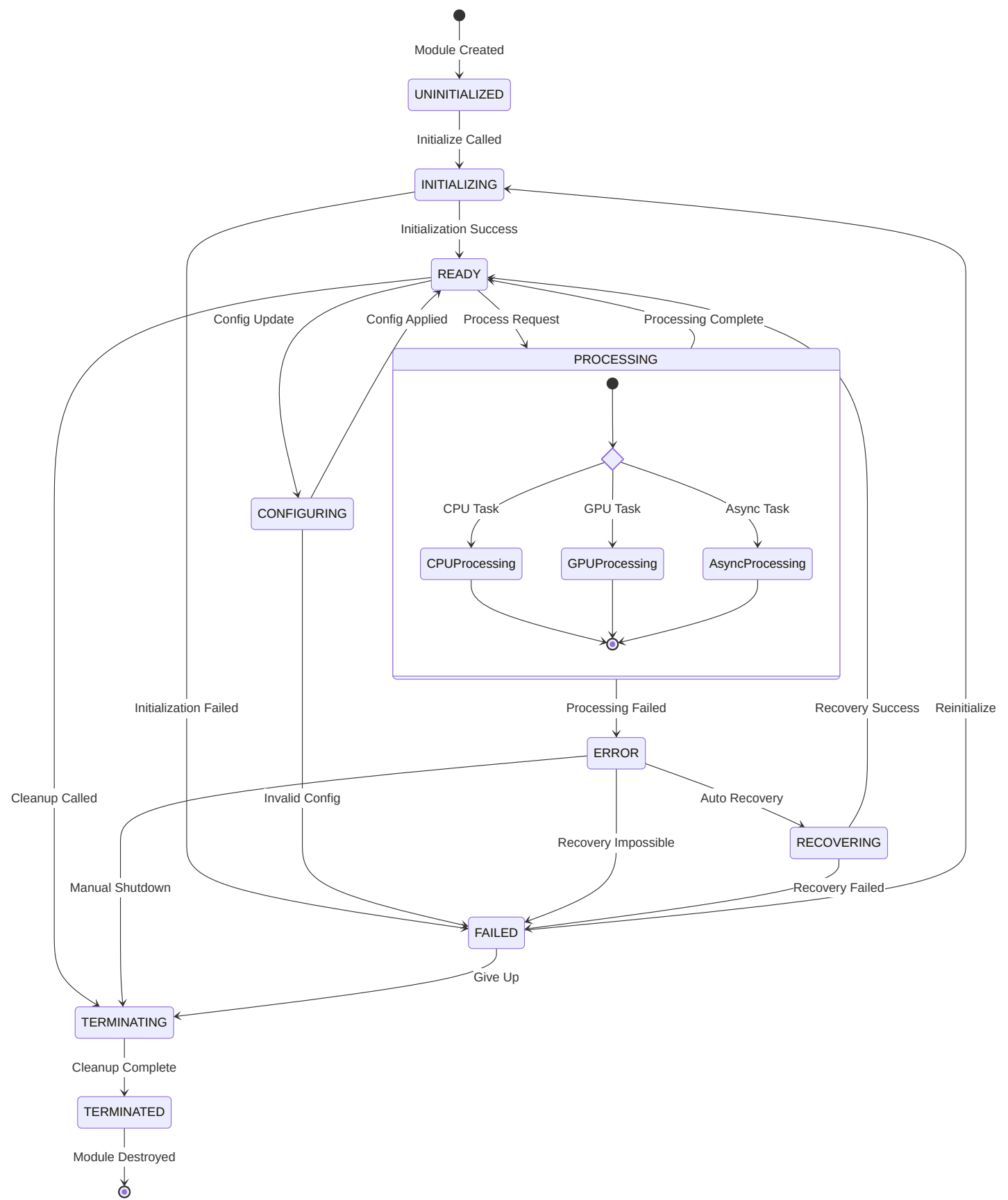
1. 이벤트 상태 전이도



2. 파이프라인 실행 상태

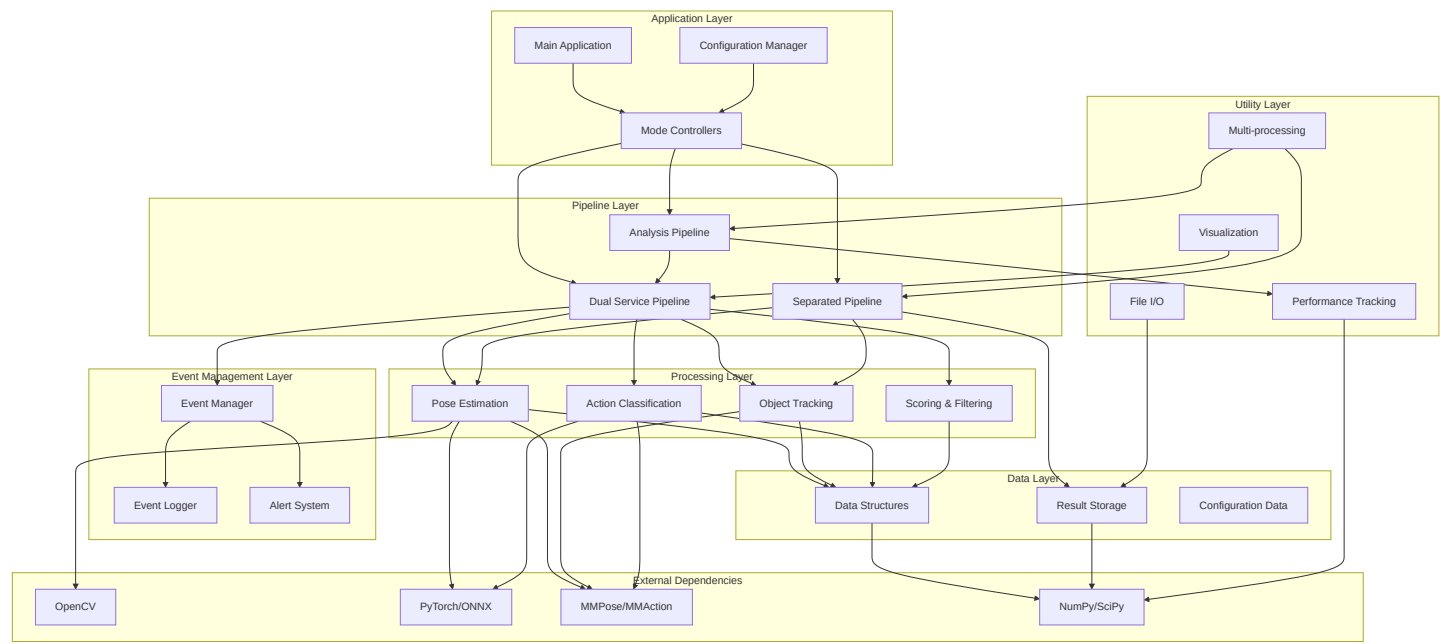


3. 모듈 생명주기 상태

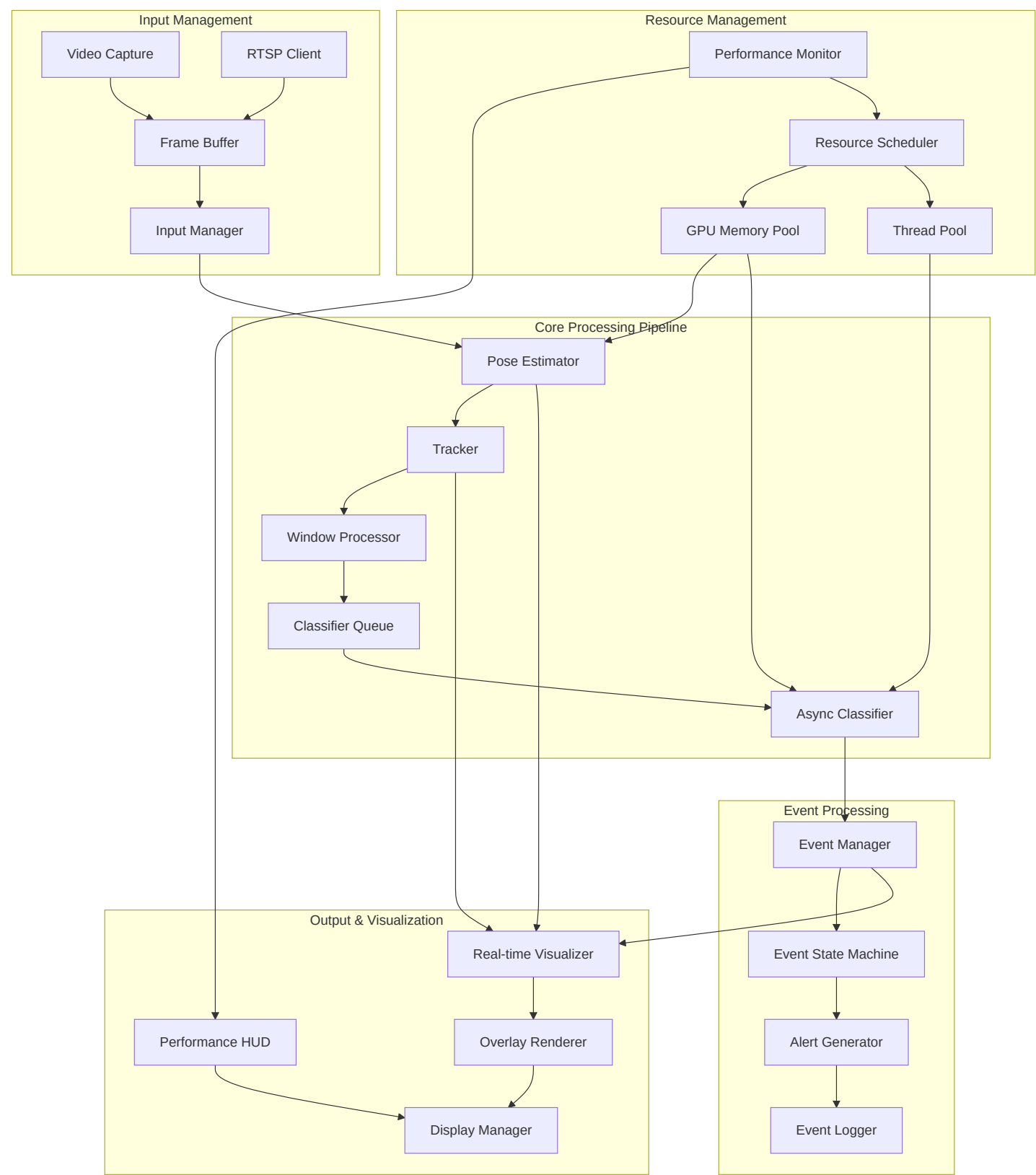


컴포넌트 다이어그램

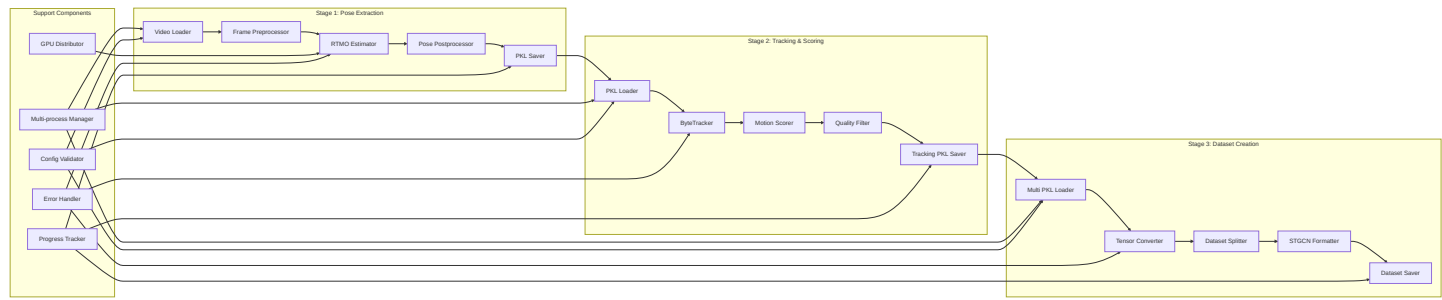
1. 전체 시스템 컴포넌트 구조



2. 실시간 처리 컴포넌트 상세



3. 데이터 처리 파이프라인 컴포넌트

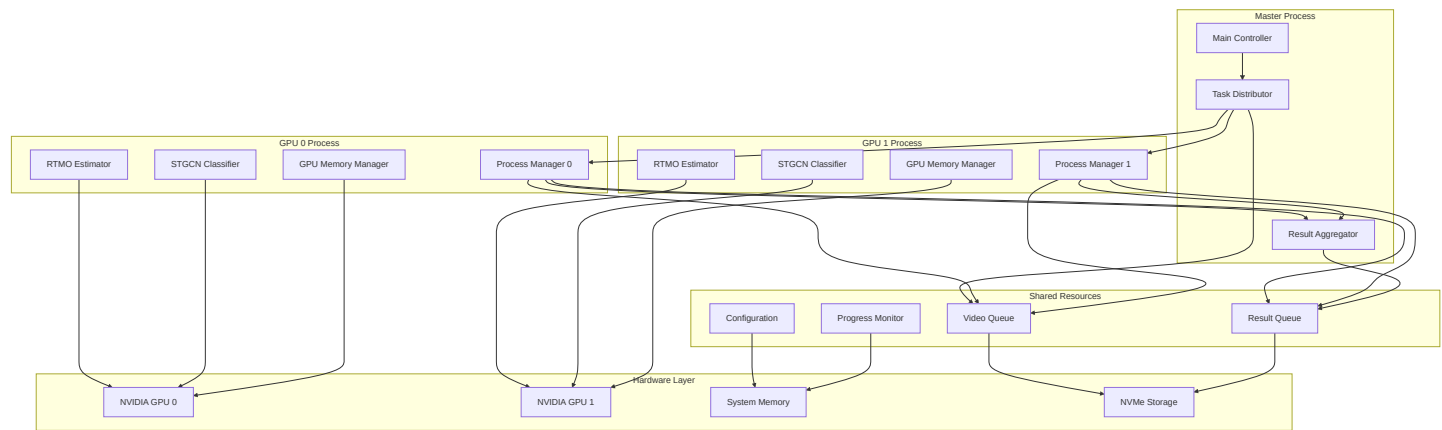


배포 다이어그램

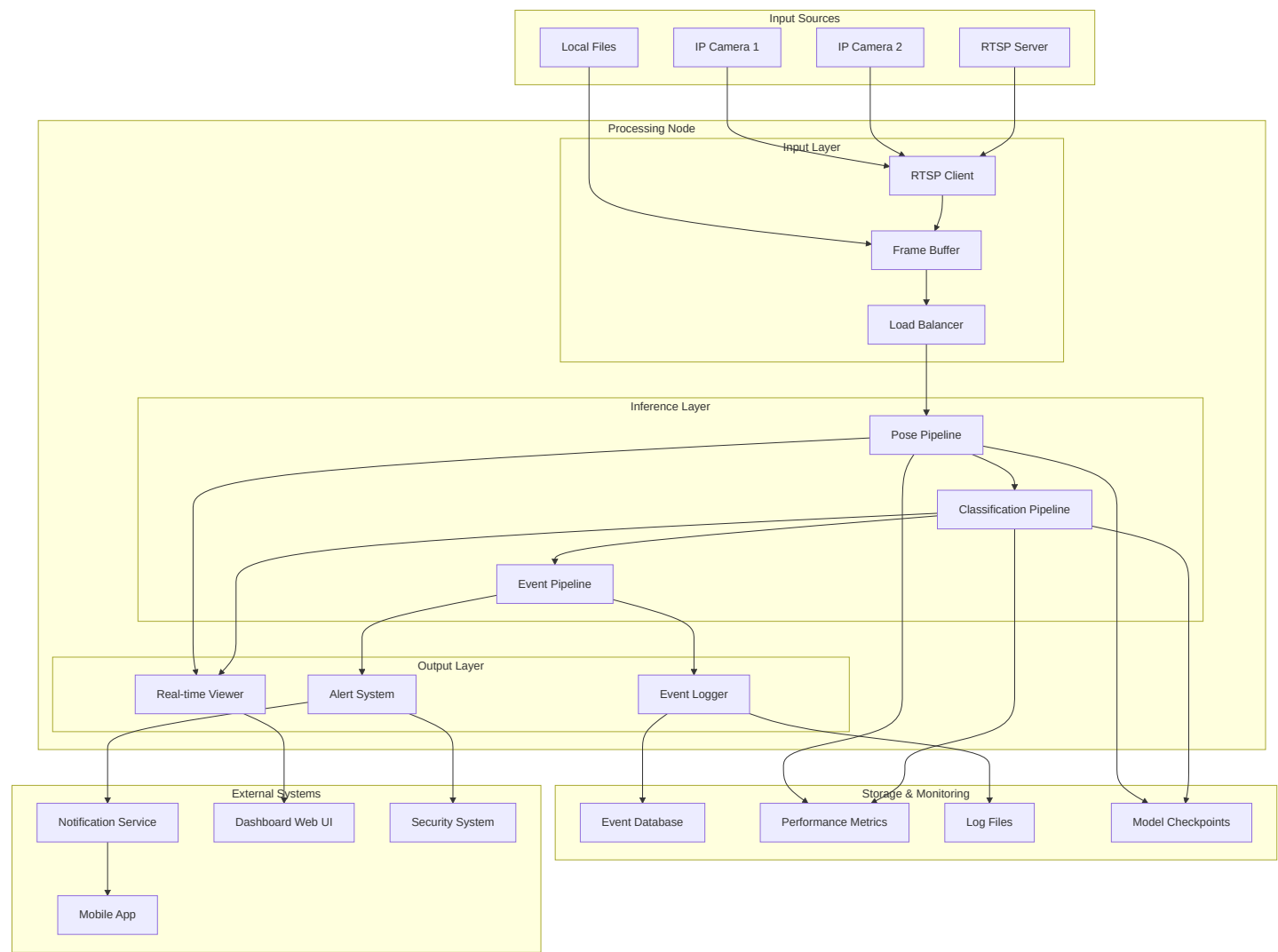
1. Docker 컨테이너 배포 구조



2. 멀티 GPU 분산 처리 구조

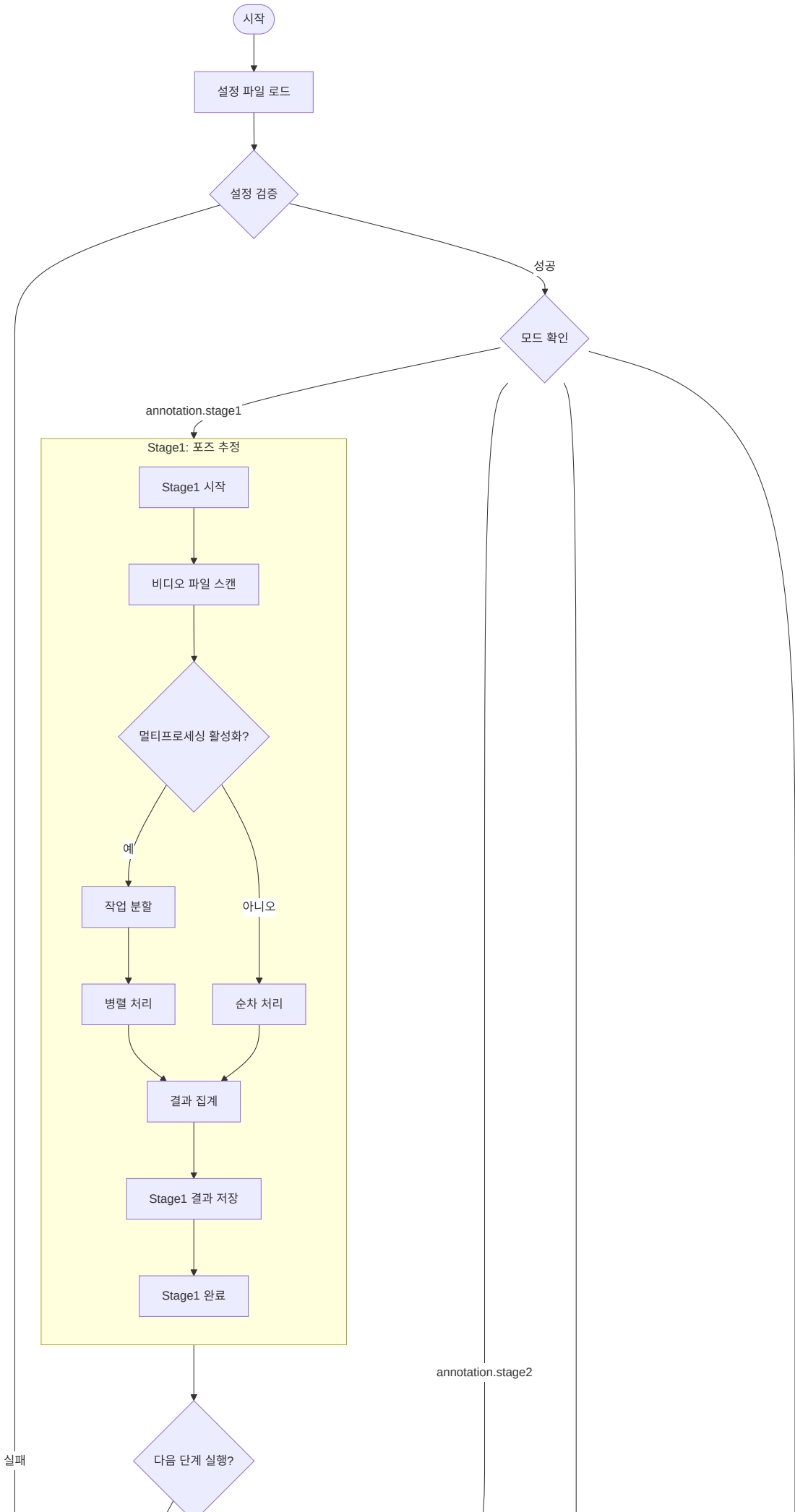


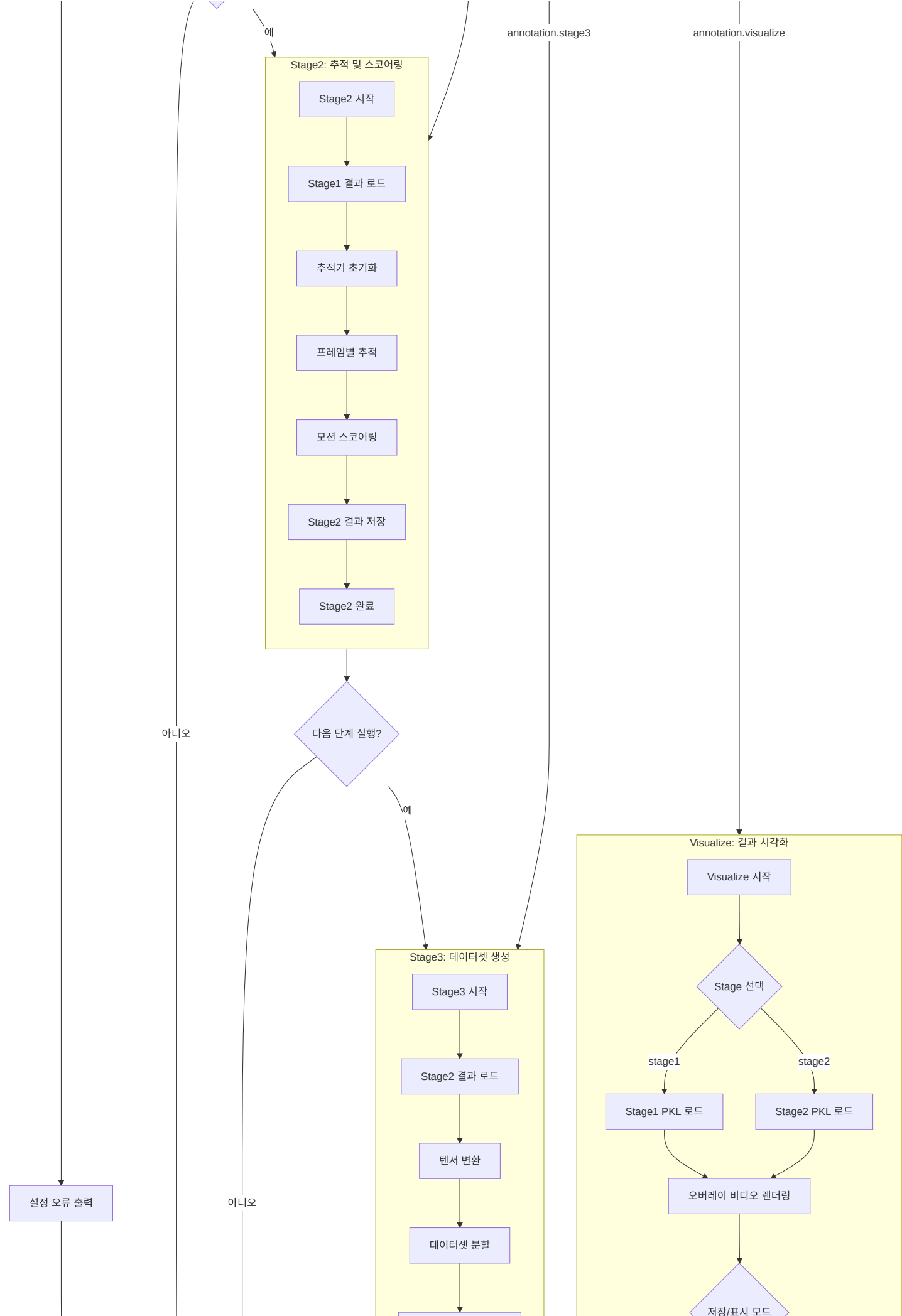
3. 실시간 스트리밍 배포 구조

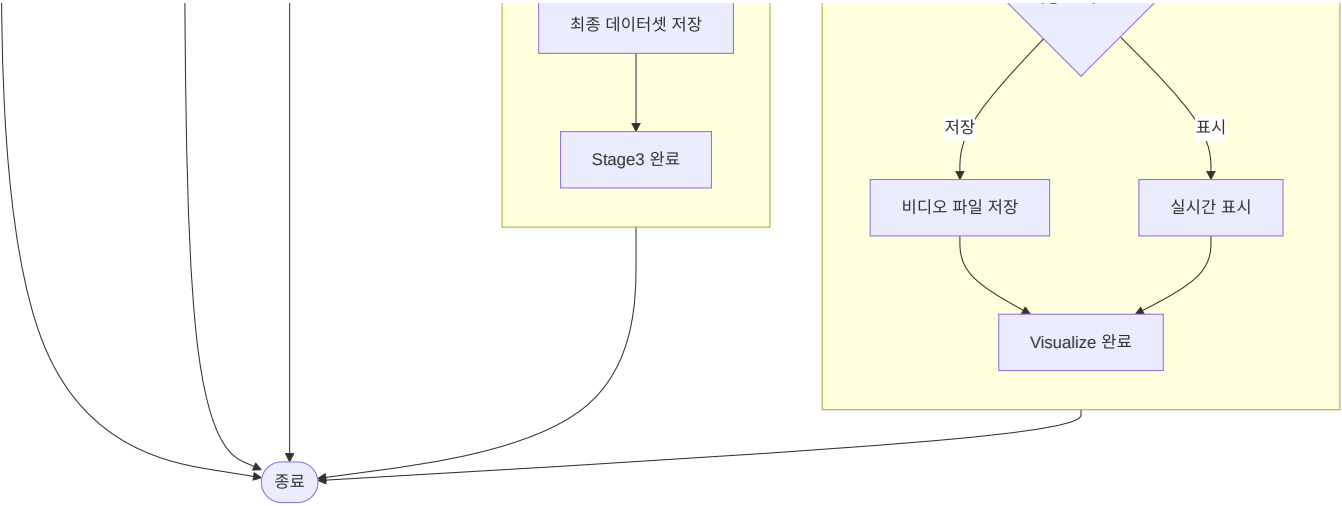


액티비티 다이어그램

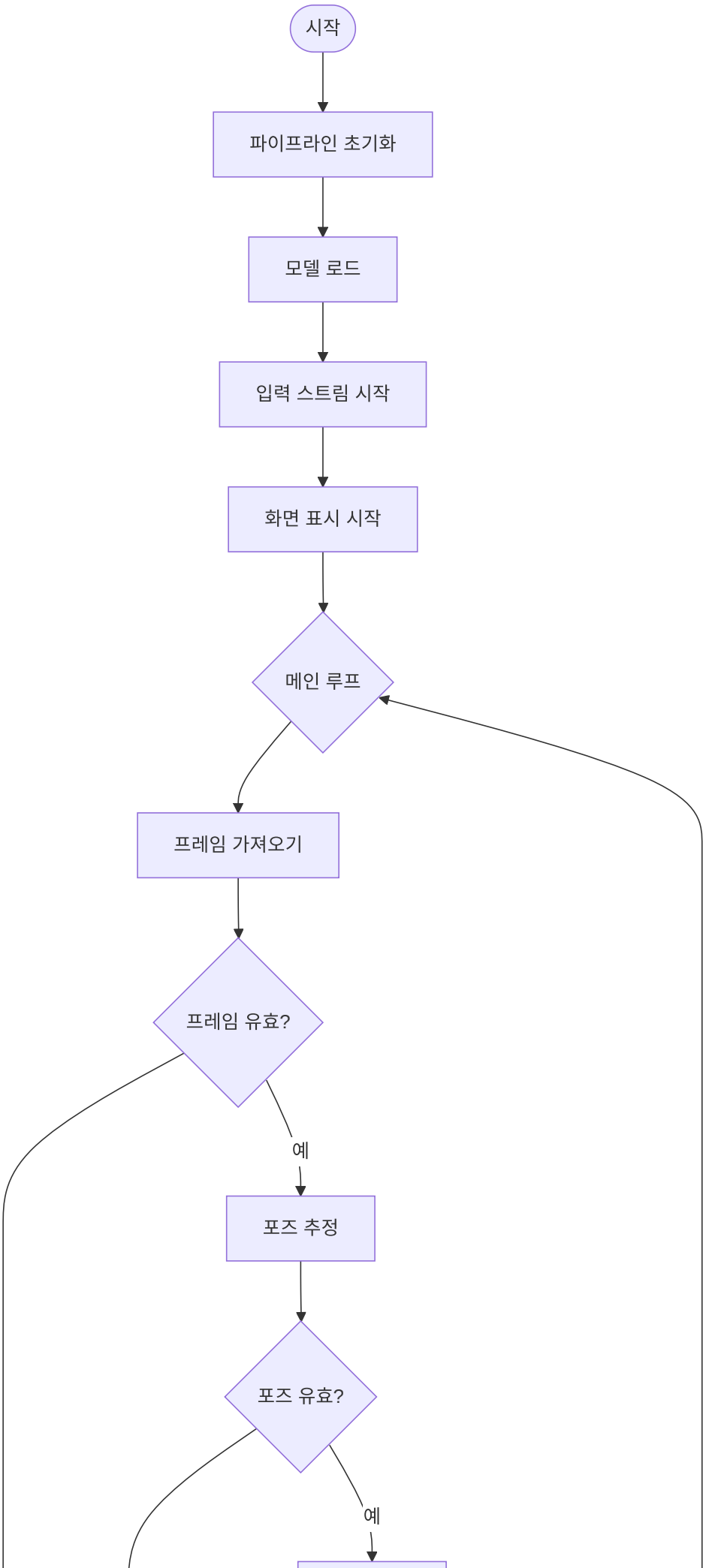
1. Annotation 모드 전체 워크플로우

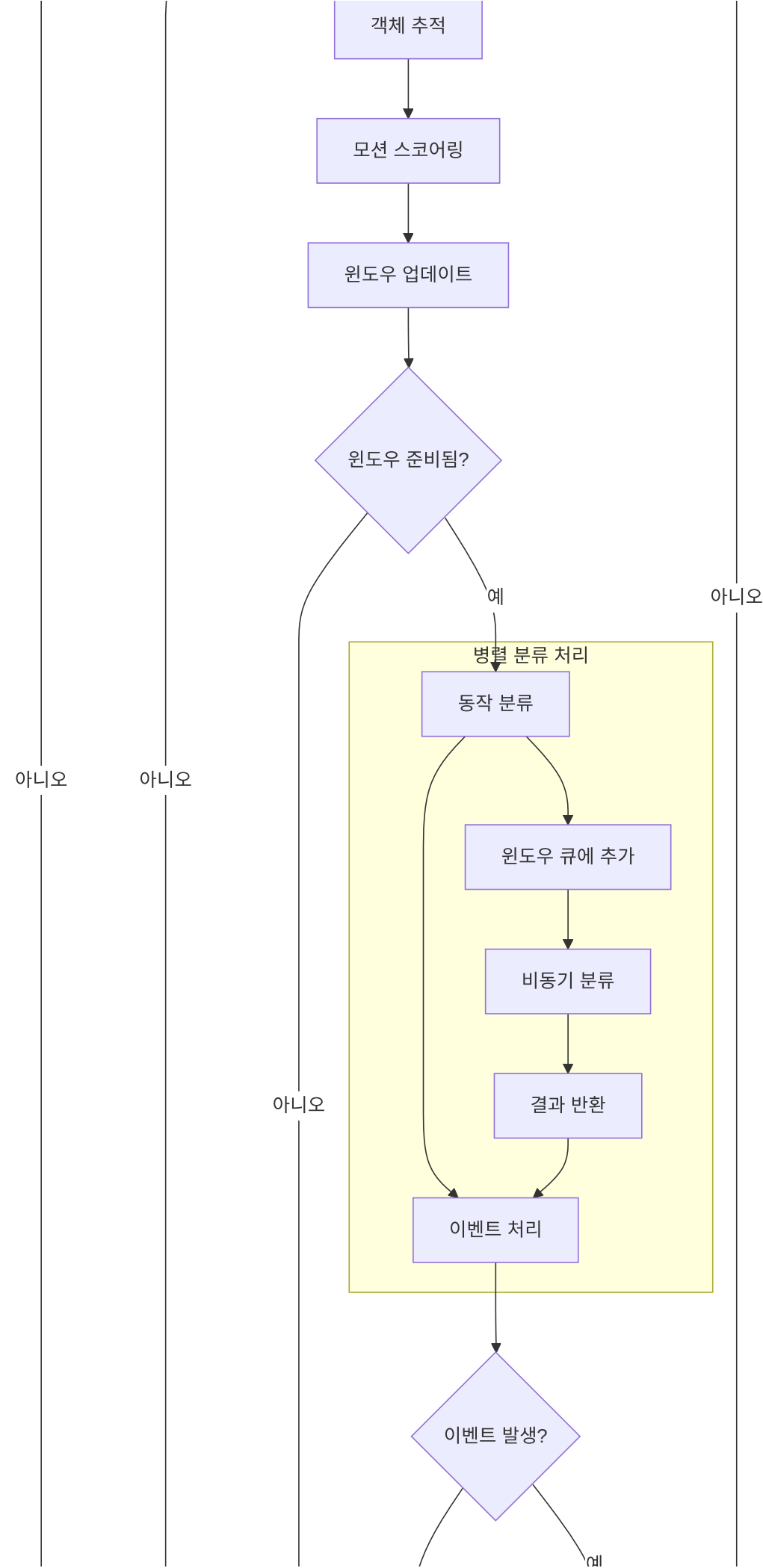


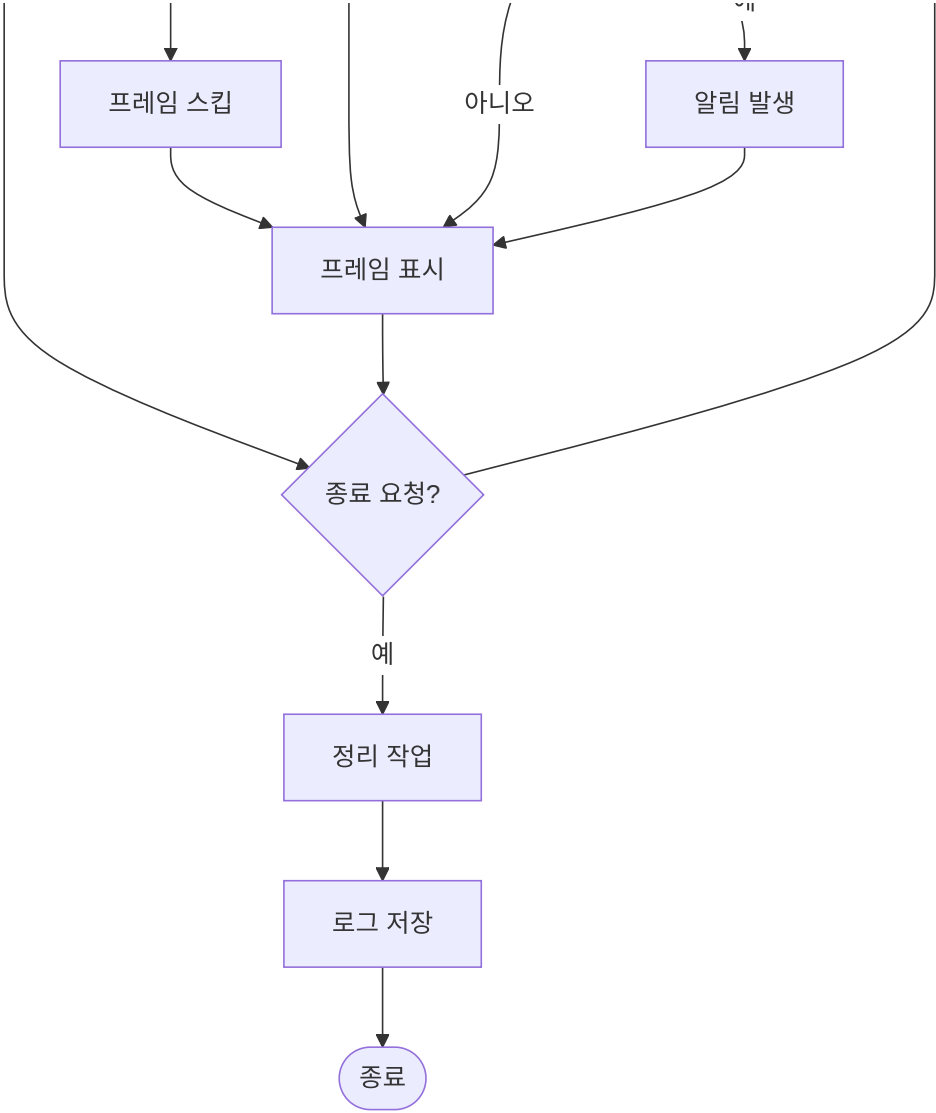




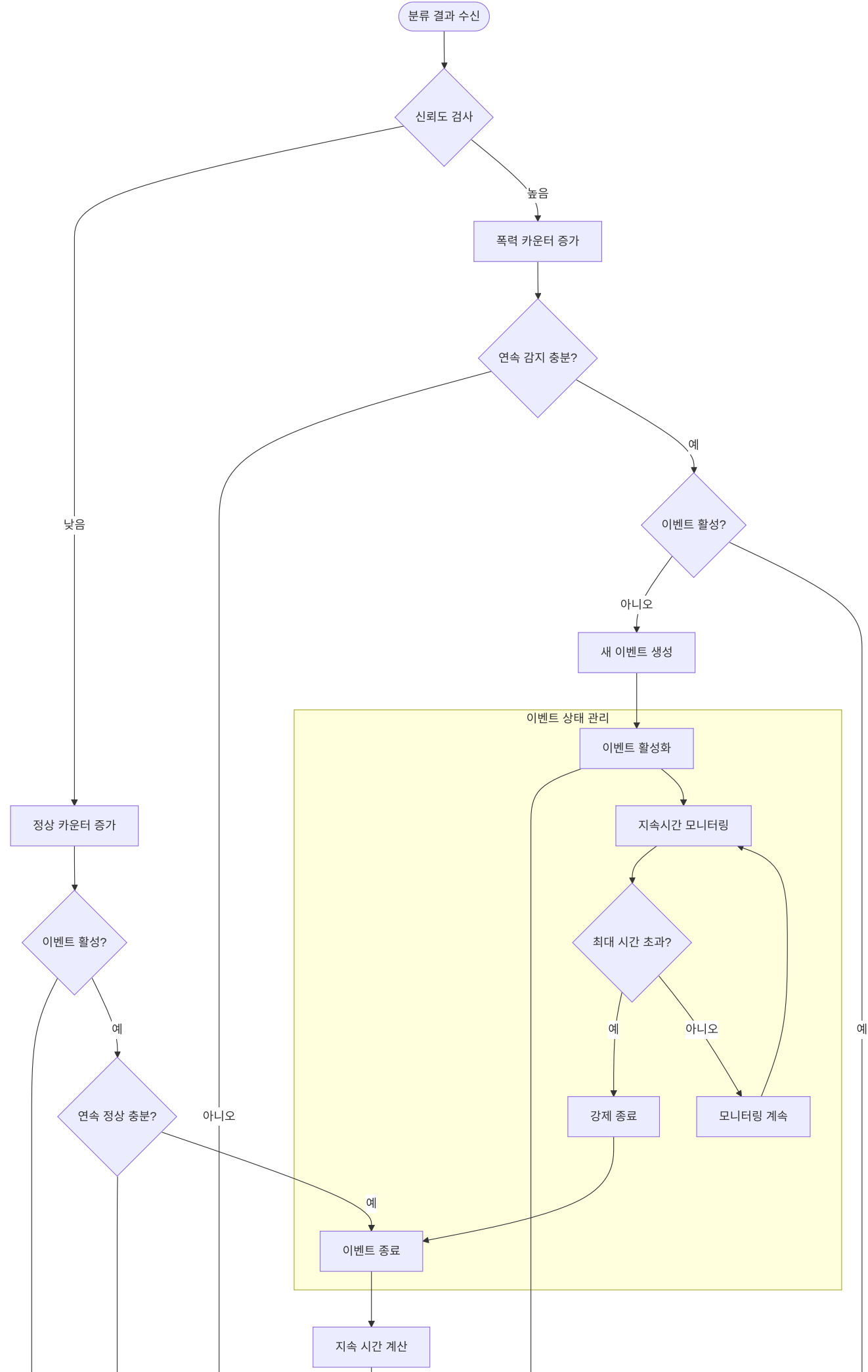
2. 실시간 처리 상세 워크플로우

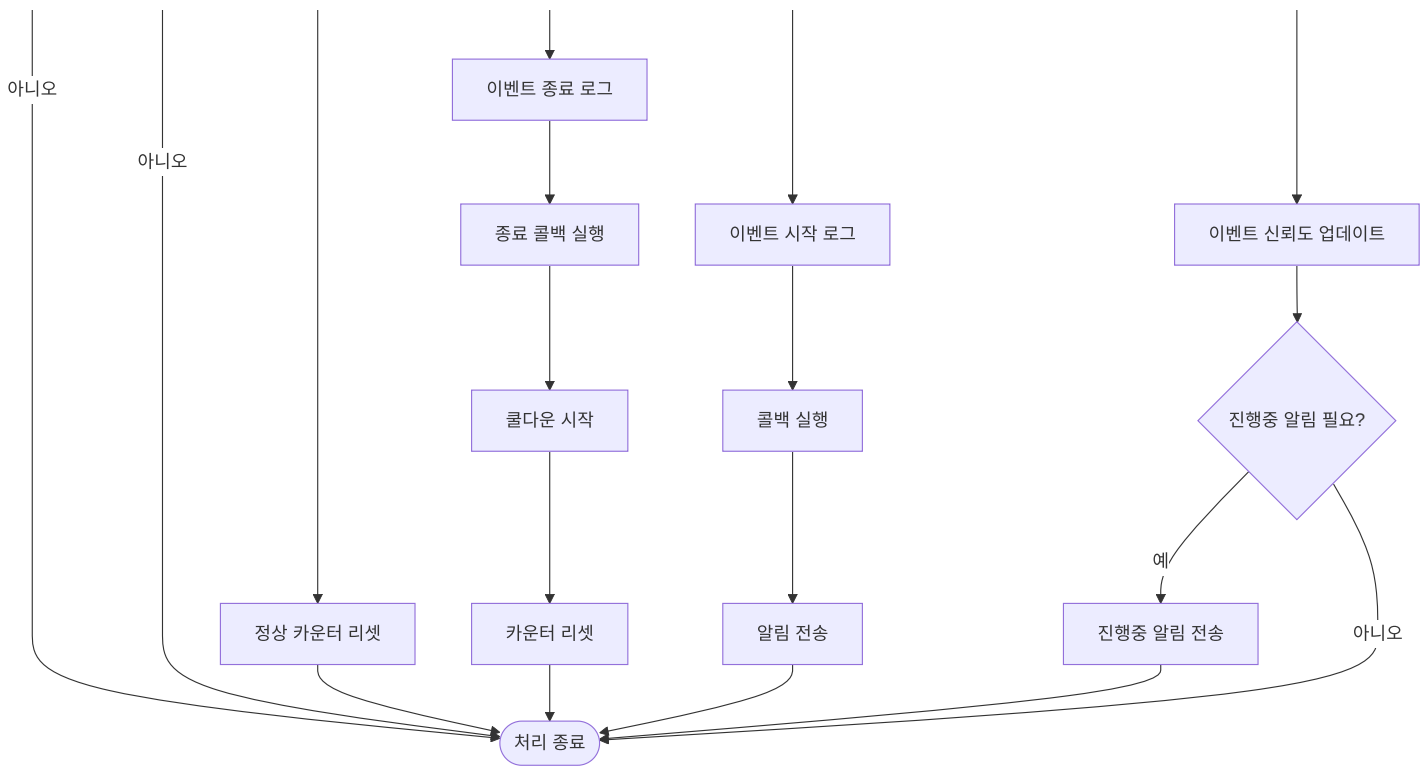






3. 이벤트 처리 상세 워크플로우





유스케이스 다이어그램

1. 전체 시스템 유스케이스



Syntax error in text
mermaid version 11.9.0

2. 실시간 처리 상세 유스케이스



Syntax error in text
mermaid version 11.9.0

3. 데이터 처리 파이프라인 유스케이스



Syntax error in text
mermaid version 11.9.0

이 포괄적인 UML 다이어그램 문서는 Recognizer 시스템의 모든 측면을 시각적으로 문서화합니다. 각 다이어그램은 시스템의 다른 관점을 제공하며, 개발자들이 시스템을 이해하고 확장하는 데 도움이 됩니다.

주요 특징

- 포괄적 커버리지:** 클래스 구조부터 배포까지 모든 아키텍처 측면 포함
- Mermaid 형식:** GitHub에서 바로 렌더링 가능한 다이어그램

3. **계층적 구조**: 전체 시스템부터 세부 모듈까지 점진적 상세화
4. **실제 코드 반영**: 실제 구현된 클래스와 메서드명 사용
5. **확장 가능성**: 새로운 모듈 추가 시 참조 가능한 패턴 제공