

합성에 대한 설명

좋습니다. 이전 설계를 하나의 통합 접근으로 보고, '얇은'이라는 표현은 제외했습니다. 슬라이드 수도 축약해 핵심 메시지를 더 밀도 있게 담았습니다. 아래는 텍스트 기반 슬라이드입니다(복붙용).

형식적 추론에서 실행 가능한 시나리오까지: AAS·SWRL·SHACL·시나리오 생성의 통합 아키텍처

- 논지: "추론 결과를 실행 가능한 인공물로 담아 재현성·추적성을 기본값으로 확보한다."
- 구성요소: QueryGoal(AAS) → SWRL(모델 선택) → SHACL(검증) → 시나리오 생성(바인딩·그룹·결합·렌더·검증) → 컨테이너 실행 → AAS 업로드
- 핵심 효과: 페루프(Closed Loop) 완결, 측정 가능한 운영·연구 품질지표(TTR/오류율/해시 일치율)

2. 전체 구조 · 데이터 흐름

입력

- QueryGoal.json (매크로 @현재시간 포함), Model MetaData (필수 입력의 진실원천), bindings.yaml (바인딩/그룹/결합 규칙), scenario.yaml.j2 (렌더 템플릿)

처리(순서 보장)

1. 매크로 확정(Asia/Seoul, ISO-8601)
2. 메타 로드(requiredInputs)
3. 바인딩 확장(URI 치환, aas:// / file:// 해석, 그룹/정렬/윈도우)
4. 결합 정책 적용(overlay/concat/union/latest; 충돌 규칙 고정)
5. 템플릿 렌더 → scenario.yaml
6. 2차 검증(JSON Schema/SHACL) → manifest.json (모든 입력·출력 sha256 기록)

출력·후속

- 컨테이너 실행 → 결과 키-값 → outputSpec 매핑 → AAS ExecutionStatus 업로드

3. 형식층(1): QueryGoal·Provenance

QueryGoal 핵심 필드

- `goalId`, `goalType`, `parameters`(필수+사용자옵션), `outputSpec`, `termination(condition/timeout)`
- `selectedModelRef` (참조) + `selectedModel` (스냅샷: `modelId`, `MetaData`, `outputs`, `preconditions`, `container{image,digest}`, `catalogVersion`, `frozenAt`)
- `selectionProvenance` (`ruleName`, `ruleVersion`, `engine`, `evidence.matched[]`, `inputsHash`, `timestamp`, `notes`)

원칙

- 시간 매크로 `@현재시간` 은 전처리 단계에서 확정
- `selectedModel.MetaData` 에서 `requiredInputs` 를 역참조 → 시나리오 입력의 기준
- 참조+스냅샷 병행으로 카탈로그 변경에도 재현성 보장

4. 형식층(2): SWRL·SHACL 분업

SWRL(추론/상태 밀기)

- 모델 선택(`GoalType`→`ModelPurpose`), 입력 바인딩 후보 생성(`Param`→`Input`), 준비 완료 신호
- 다중 후보 시 **한 개** 선택은 SPARQL SELECT(ORDER BY rank, version DESC LIMIT 1)로 결정

SHACL(검증)

- `requiredInputs` \subseteq `providedInputs`
- `outputSpec` 타입·완결성, 컨테이너 `digest` 패턴, 시간 형식(ISO-8601)
- 상태 단일성: `status` 는 FunctionalProperty + `maxCount=1`

결론: SWRL=무엇을, SHACL=제대로인가. 실행·업로드는 외부 오케스트레이터가 담당

5. 시나리오 생성: 결정적 의미론 · 실행 예시



시나리오 생성은 시나리오 생성물(특정 생성물 이름 기반) 혹은 시나리오 생성기를 통해서 생성함

결정적 규칙(재현성의 토대)

- 정렬: `mtime(desc) → name(asc) → size(asc)`
- 윈도우: `{count:N}` 또는 `{since: ISO, until: ISO, inclusive: both|left|right|none}`
- 결합:
 - **overlay**: 충돌 시 `last-wins` 또는 `error` 명시
 - **concat**: 안정 정렬 후 순차 연결
 - **latest**: 동일 키는 최신 타임스탬프만 유지
- 타임존: **Asia/Seoul** 고정

실행 예(요지)

1. `@현재시간` → `"2025-09-12T15:13:31+09:00"` 확정
2. Meta → `requiredInputs=["JobRoute","MachineState","Calendar","SetupMatrix","WIP","Backlog"]`
3. 그룹/정렬/윈도우 → WIP 최신 5개 선택
4. 결합: WIP=**overlay(last-wins)**, Backlog=**concat**
5. 렌더 → `scenario.yaml` / 검증 → OK / `manifest.json` 에 모든 경로·정책·해시 기록

6. 품질·위험·지표 (논문·운영 공통)

핵심 지표

- **TTR(Time-to-Ready)**: Goal → scenario 생성까지
- **오류율**: SHACL 위반/경로 누락/결합 충돌
- **재현성 점수**: `scenario.digest` 일치율
- **스루풋/동시성**: 동시 N개 Goal 처리량·실패율
- **E2E 지연**: Goal → AAS 업로드까지

주요 위험 & 처방

- 결정성 미흡 → 정렬/윈도우/결합 정의 고정
- 다중 모델 후보 → **랭킹·버전 기반 단일 선택**(SPARQL)
- 상태 충돌 → **원자적 상태 전이**(SPARQL UPDATE: 기존 제거 → 신규 삽입)
- 보안/무결성 → 파일 루트 화이트리스트, 템플릿 Sandbox, 전 입력·출력 **sha256** 기록
- 메타 일관성 → Meta 로더에 `metaLoadedAt/metaDigest` 기록(SSOT)

7. 결론 · 실행 체크리스트

메시지

- 하나의 통합 접근으로 *형식(의미 보장)*과 *엔지니어링(운영 실효)*을 접목해 **페루프**를 완결한다.
- 참조+스냅샷, 매크로 확정, 결정적 생성 규칙, 매니페스트 해시가 **재현성·감사**의 4축.

체크리스트(바로 적용)

- ☐ 매크로 전처리(Asia/Seoul)
- ☐ Meta 로더(`requiredInputs` SSOT)
- ☐ 바인딩 확장(URI/그룹/정렬/윈도우)
- ☐ 결합 정책(overlay/concat/union/latest) 문서화·테스트
- ☐ 템플릿 렌더 + 2차 검증(JSON Schema/SHACL)
- ☐ `manifest.json` 에 모든 입력·정책·해시 기록
- ☐ 실행→결과 키-값→ `outputSpec` 매핑→**AAS 업로드** 자동화

필요 시, 위 슬라이드에 맞춰 bindings 스키마/시나리오 스키마/상태 전이 SPARQL/참조 구현 CLI를 한 묶음으로 정리해 드리겠습니다.