유엔진 AX 컨설팅 서비스

Agent Storming

사람과 AI가 함께 일하는
업무 혁신 방법론

EventStorming에서 협업 구조와 실행까지

UEngine

목차

Agent Storming 방법론 설명서

01 문제 인식과 방법론의 출발점 03

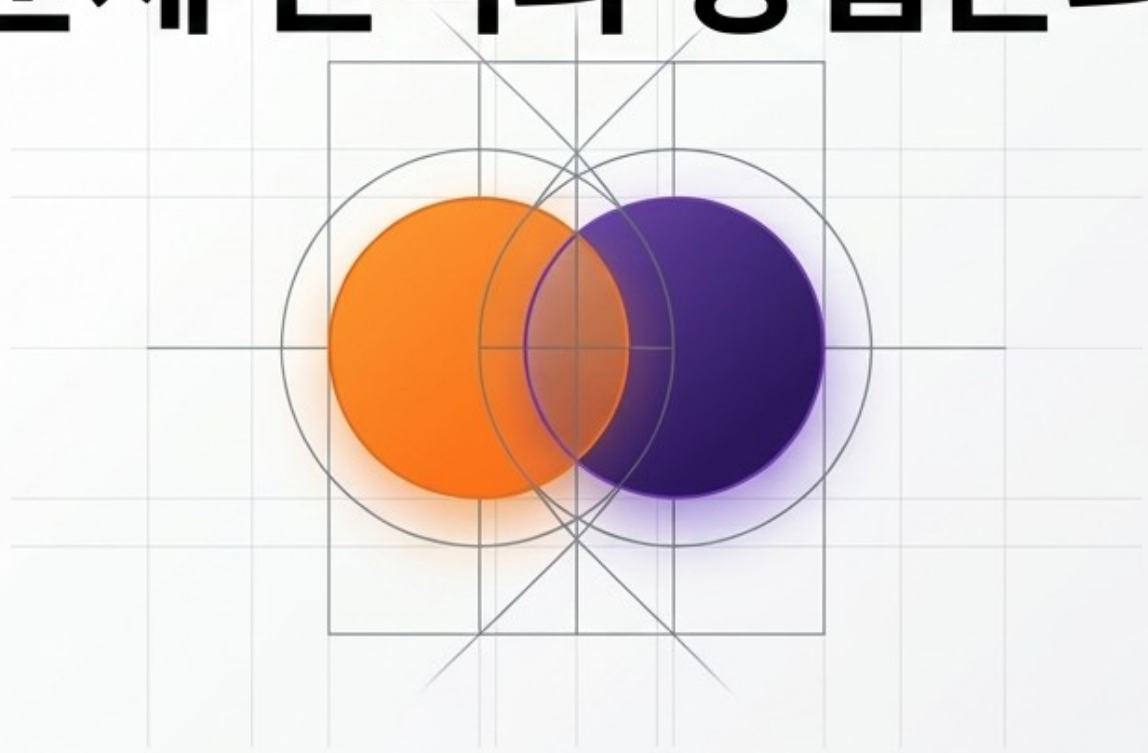
02 EventStorming과 기회 발견 13

03 에이전트의 책임과 경계 24

04 실행과 학습 36

UEngine

제1부: 문제 인식과 방법론의 출발점



AI 에이전트 도출을 위한 이벤트 스토밍 활용 방안

어디에 둘 것인가, 어떻게 나눌 것인가

UEngine

15%

Gartner 전망 (2028년):
일상 업무 의사결정의 15%를
에이전트가 자율적으로
내릴 것으로 예상.

실패는 다음 두 지점에서 동시에 발생합니다:

1. 어디에 쓸 것인가? (도입 대상 선정의 실패)
2. 어떻게 나눌 것인가? (구성 단위 설계의 실패)

40% 이상

Gartner 경고 (2027년 말):
에이전틱 AI 프로젝트의 40%
이상이 취소될 것으로 전망.

약 130곳

수천 곳의 에이전틱 AI 업체 중
실제 에이전트 역량을 갖춘 곳은
약 130곳에 불과 (Gartner 추정).

MEngine

41~86.7%

주요 오픈소스 멀티 에이전트 시스템 7종의 작업 실패율 범위 (출처: UC Berkeley MAST)

기반 데이터: 실행 기록 1,642건

분석된 실행 기록의 분류 (실패 원인 분포)

- **설계 (44.2%)**
  - 단계 반복: 15.7%
  - 종료 조건 인식 실패: 12.4%
- **협업 불일치 (32.3%)**
  - 추론, 행동 불일치: 13.2%
  - 작업 이탈: 7.4%
- **검증 (23.5%)**
  - 잘못된 검증: 9.1%
  - 검증 누락: 8.2%

결론: 전체 실패의 76.5%가 설계와 협업 구조에서 발생.

비용 문제: 멀티 에이전트의 토큰 사용량은 일반 채팅 대비 15배, 단일 에이전트는 4배 (출처: Anthropic 2025.06)

MEngine

어디에 쓸 것인가 (도입 대상 선정 실패)

기술 중심의 도입 대상 선정

- 비용 증가
- 불분명한 사업 가치
- 위험 통제 미흡

> 현실: 에이전틱으로 소개된 사용 사례 상당수는 에이전트 구현이 필요하지 않음 (Gartner)

어떻게 나눌 것인가 (구성 단위 설계 실패)

기술 중심의 에이전트 분할

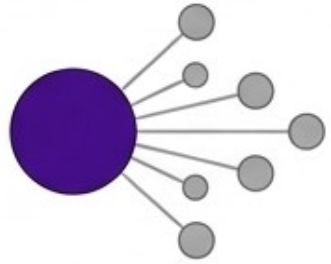
- 설계 및 협업 붕괴
- 토큰(비용) 폭발적 증가

> 현실: 기술이나 조직 형태에서 출발하여 업무의 '의미'를 놓침

Gartner의 권고: 기존 업무에 에이전트를 덧붙이기보다, 업무 흐름을 처음부터 다시 설계하는 것이 성공 경로' 입니다.

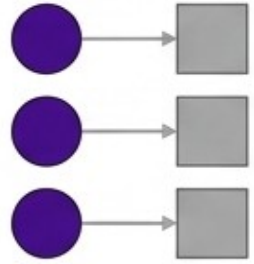
MEngine

1. 단일 만능 에이전트



나누는 기준: 나누지 않음 (모든 도구와 지시를 한 곳에)
대표 구현: 도구를 계속 더하는 단일 에이전트

2. 도구와 시스템 단위



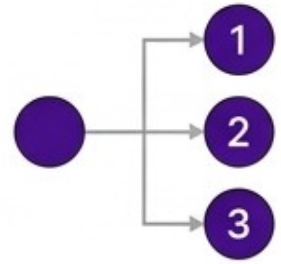
나누는 기준: 연결할 시스템이나 API마다 하나
대표 구현: 메일 에이전트, ERP 에이전트, 검색 에이전트 등

3. 직무와 페르소나 단위



나누는 기준: 사람 조직의 직무나 역할 이름
대표 구현: 조사 담당, 작성 담당, 검토 담당 (역할 기반 팀)

4. 작업 단계 단위



나누는 기준: 실행할 때 쪼갠 하위 작업
대표 구현: 오케스트레이터와 워커, 순차 파이프라인

현재의 범주들은 모두 '기술'이나 '조직 형태'에서 출발하며, 업무의 의미는 나누는 기준에 들어가지 않습니다.

MEngine

맥락 과부하 (하나로 통합)

1. 단일 만능 에이전트

발생 문제: 도구와 지시가 늘수록 지시를 잊고 도구를 잘못 고름.

- LangChain: 도메인과 도구가 늘수록 지시를 잊음.
- Chroma: 입력이 길수록 18개 모델의 성능이 불안정함.

맥락 단절 및 결정 충돌 (잘게 분할)

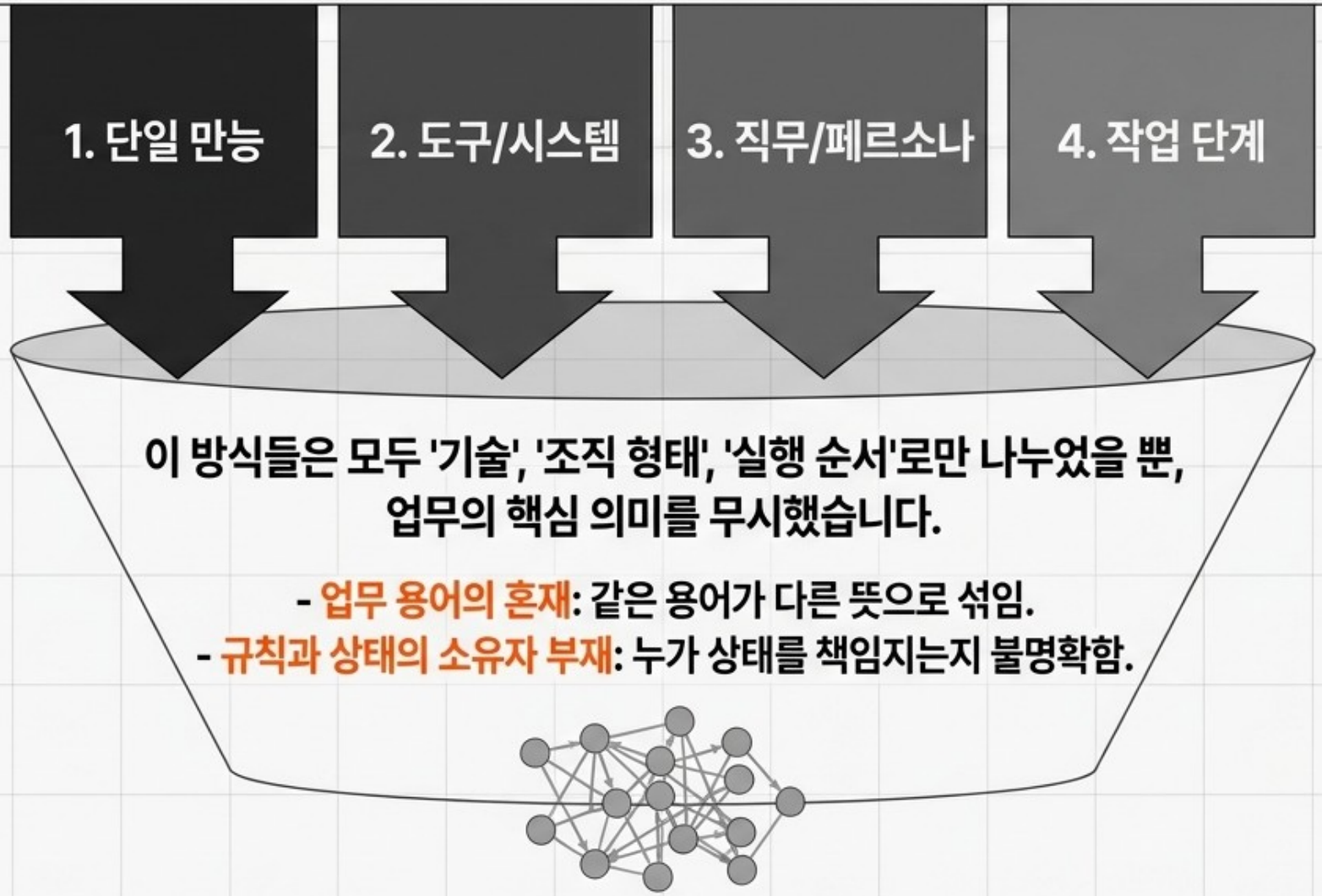
2. 도구/시스템, 3. 직무/페르소나, 4. 작업 단계 단위

발생 문제: 하위 에이전트가 서로의 결정을 몰라 충돌, 규칙을 지킬 주체 부재. 책임 혼선.

- Microsoft: 에이전트 간 연결마다 프로토콜, 상태 동기화 추가 필요. 역할 분리만으로는 나눌 이유 없음.
- MAST: 역할 명세 위반, 작업 이탈, 확인 누락 발생.
- Cognition: 병렬 에이전트의 암묵적 결정 충돌.
- Anthropic: 통신 비용으로 토큰 사용량 폭증 (일반 채팅의 15배).

에이전트 수를 늘리거나 줄이는 것만으로는 해결되지 않으며, '어디서 나눌 것인가'가 문제의 핵심입니다.

MEngine



결과: 에이전트 계층의 진흙탕 덩어리 (Big Ball of Mud)

에이전트 사이의 의존성이 거미줄처럼 얽힘.

MEngine



단일 모놀리스:
모든 기능이 한 덩어리.



기술 계층별 분할:
화면, 로직, 데이터로 나눔.



분산 모놀리스 (Big Ball of Mud):
나뉘었지만 함께 바뀌어야 함. 얽힘 발생.



바운디드 컨텍스트 단위 서비스:
DDD로 경계를 다시 그림. 질서 회복.



단일 만능 에이전트:
모든 도구와 지시가 한 곳에.



도구, 직무, 작업 단계별 분할:
기술과 조직 형태로 나눔.



에이전트 진흙탕:
나뉘었지만 서로 얽히고설킴.

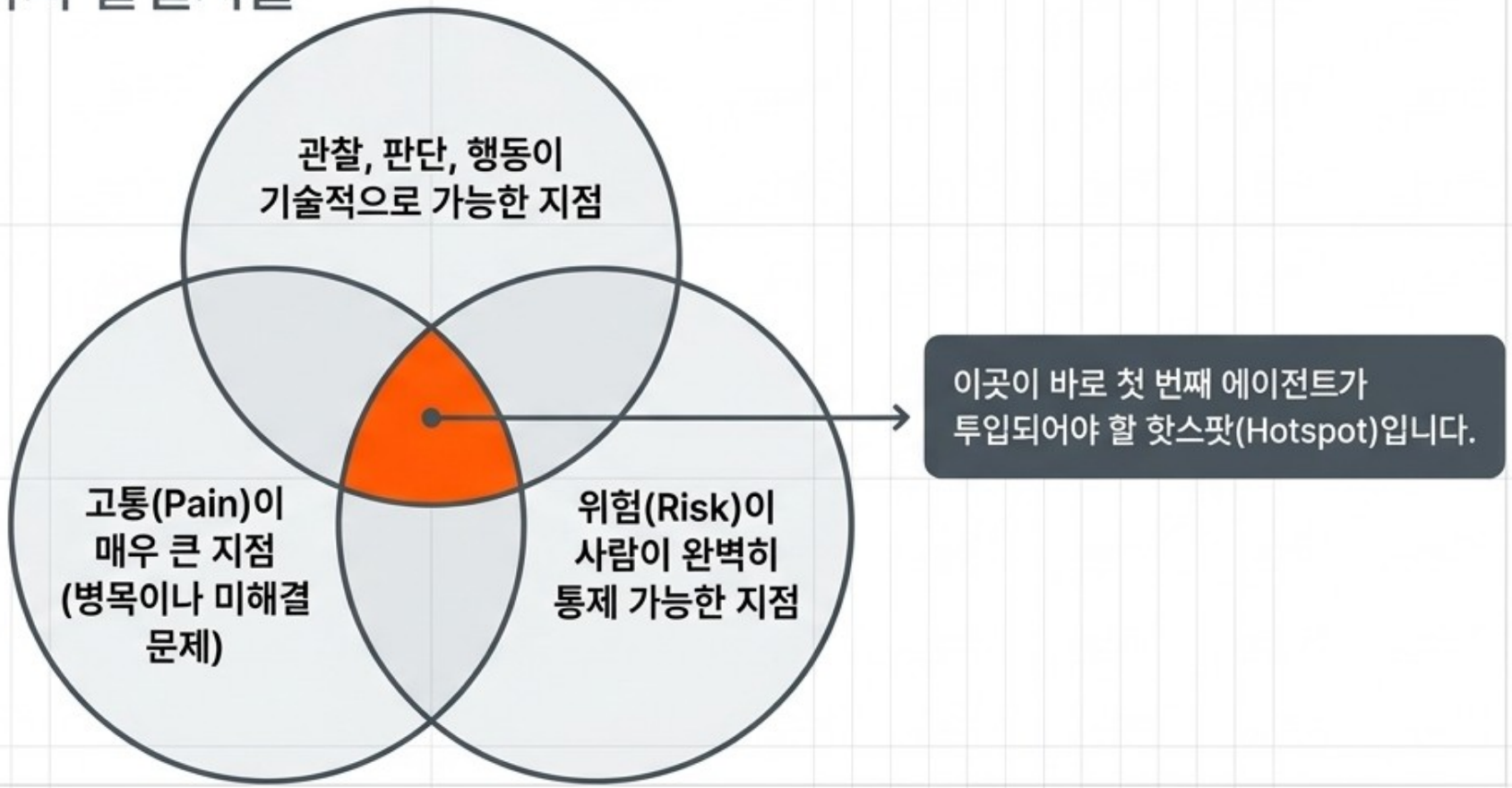


바운디드 컨텍스트 단위 에이전트:
(이 자료의 제안) 업무의 의미를 기준으로 경계를 다시 그림.

MEngine

에이전트 스토밍의 답변 1: 어디에 둘 것인가?

무엇을 자동화할지와 어디까지 맡길지를 동시에 정해야 합니다.



결론: Agent Storming 도입 권고 3단계

이벤트 스토밍 한 번으로 둘 곳, 경계, 설계 재료를 도출하고 검증합니다.



1. 현장 업무 시각화

현장 업무를 '과거형 사건(Event)' 중심의 이벤트 스토밍으로 먼저 펼칩니다.

2. 컨텍스트 기반 분할

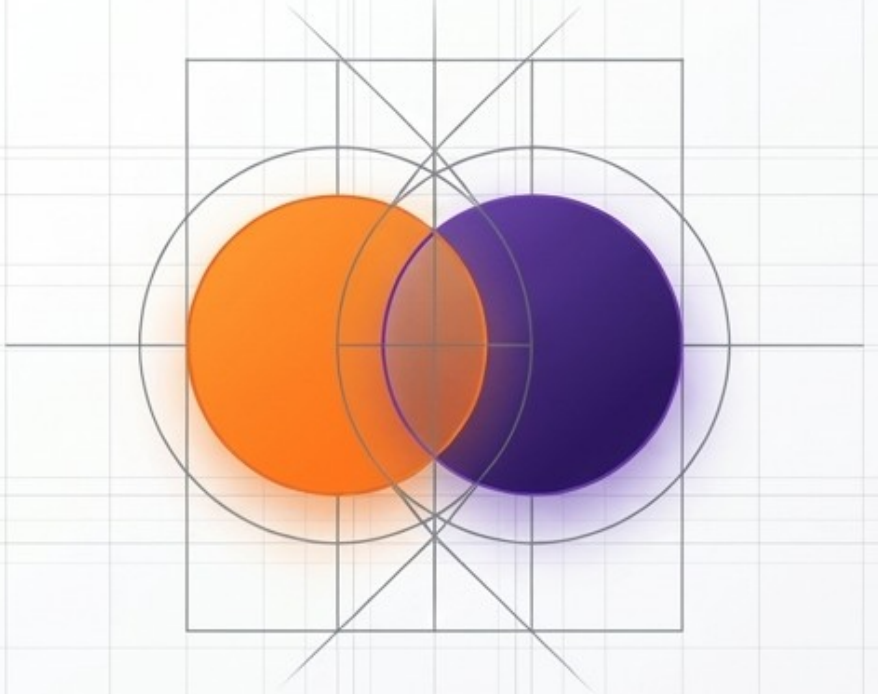
업무의 의미가 일치하는 '컨텍스트 경계' 안에서만 에이전트를 나누고, 이벤트로 연결합니다.

3. 단일 파일럿 검증

고통은 크고 위험은 통제 가능한 '핫스팟'부터 단일 파일럿으로 도입하고 검증 후 확장합니다.

기술 중심의 도입 실패를 피하고, 가치 흐름을 주도하는 사람 중심의 AI 협업 청사진을 지금 시작하십시오.

MEngine



제2부: EventStorming으로 기회와 설계 재료 발견

업무 현장에서 출발하는 가장 구체적인 지능형 에이전트 설계 매뉴얼

MEngine

기술 중심 접근의 한계를 극복하는 3가지 설계 결과물



1. 에이전트를 둘 곳 (위치)

어디에 쓸 것인가?
관찰, 판단, 행동이 가능한
핫스팟(Hotspot)을 에이전트
기회로 전환.

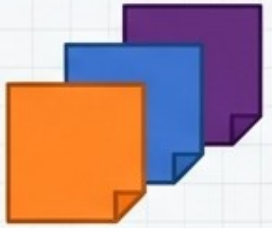
[해결하는 실패]
가치가 불분명한 도입 대상 및
유행을 따른 사용 사례



2. 에이전트의 경계 (책임)

어디까지 맡길 것인가?
용어와 규칙이 일관된 바운디드
컨텍스트(Bounded Context)를
책임 영역으로 지정.

[해결하는 실패]
맥락 과부하, 결정 충돌,
책임 혼선



3. 에이전트의 설계 재료

무엇으로 움직일 것인가?
워크숍의 9가지 아티팩트가 그대로
시스템 아키텍처 코드로 치환.

[해결하는 실패]
업무 규칙 위반, 규정 충돌에
따른 환각 현상

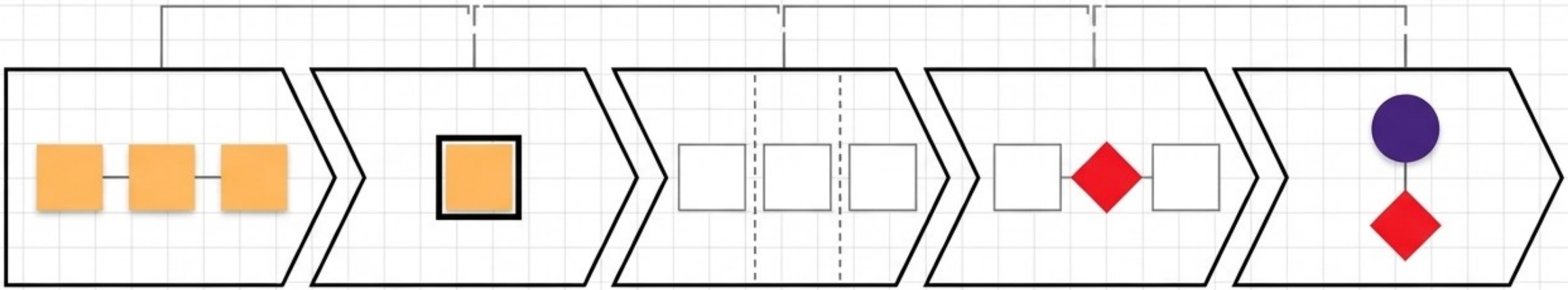
출처: 원문 2~6절 및 7쪽

[출처: 원문 2-6절 및 7쪽]

MEngine

혼돈의 현장을 재구성하는 5단계 워크숍 절차

업무를 사건의 시간 순서로 펼치고, 경계를 그어 에이전트 후보를 명세합니다.



1. 이벤트 펼치기

업무에서 일어난 과거형 사건을 시간 순서로 배열합니다.

2. 피벗 이벤트 표시

되돌릴 수 없는 전환점을 굵게 표시하여 기준을 세웁니다.

3. 경계 긋기

피벗 이벤트와 용어, 담당 변화를 기준으로 '컨텍스트'를 나눕니다.

4. 핫스팟 표시

병목과 불명확한 지점을 찾아 빨간 마름모로 표시합니다.

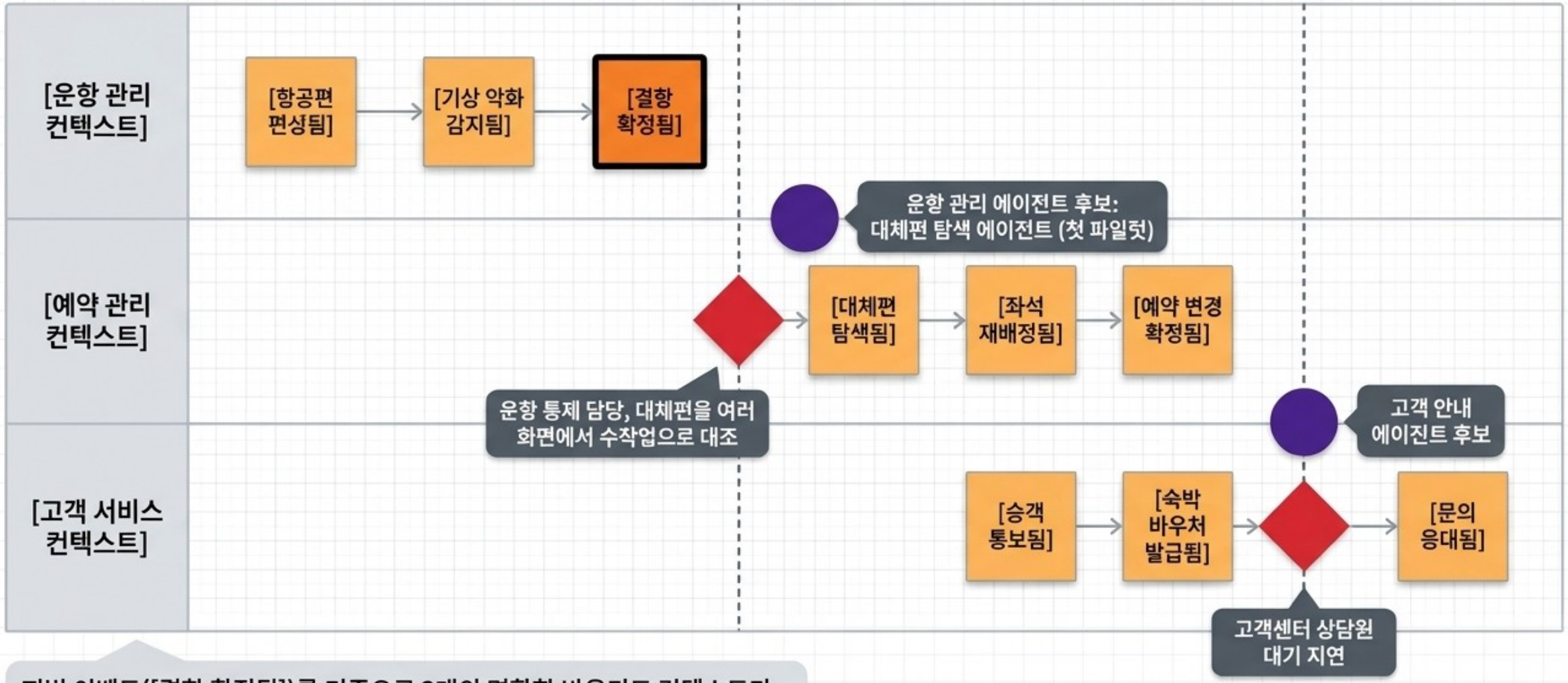
5. 기회와 명세

컨텍스트별 후보를 정하고 가장 성공률이 높은 첫 파일럿을 고릅니다.

[출처: 원문 4절, Brandolini] | ※ 권장 참여 규모: 업무 여정 하나를 함께 아는 8~12명

MEngine

사례 연구: 항공 결항 타임라인과 에이전트 후보 도출



피벗 이벤트([결항 확정됨])를 기준으로 3개의 명확한 바운디드 컨텍스트가 나뉘며, 각 영역의 핫스팟에서 에이전트 파일럿이 도출됩니다.

[출처: 원문 8쪽]

※ 본 보드는 원문의 결항 시나리오를 시각화

원본 그림 1의 결항 시나리오 예시

MEngine

설계 재료 치환 공식 [1/2]: 행동과 규칙의 매핑

이벤트 스토밍 워크숍에서 벽에 붙인 스티커는 그대로 AI 에이전트의 시스템 코드로 번역됩니다.

도메인 이벤트 (Domain Event)

도메인 의미: 이미 일어난 과거 사실

AI 매핑: 비동기 트리거 입력, 작업 완료 출력 메시지

커맨드 (Command)

도메인 의미: 상태 변경을 일으키는 명령과 의도

AI 매핑: 에이전트 작업 지시, 도구 호출(Function Calling) 신호와 행동

정책 (Policy)

도메인 의미: 조건에 반응하는 업무 규칙

AI 매핑: 시스템 프롬프트, 조건부 라우팅, 가드레일 (허용 범위 밖 실행을 막는 규칙)

읽기 모델 (Read Model)

도메인 의미: 의사결정용 정보 뷰

AI 매핑: RAG 컨텍스트, 인메모리 뷰, 데이터 조회 판단과 맥락

[출처:

출처: 원문 표 1

AIEngine

설계 재료 치환 공식 [2/2]: 상태 관리와 책임 경계

애그리게잇 (Aggregate)

도메인 의미: 트랜잭션 일관성의 최소 단위
AI 매핑: 영속 상태 저장소, 쓰기 권한 범위

행위자 (Actor)

도메인 의미: 작업을 수행하는 주체
AI 매핑: AI 에이전트 또는 사람 (Human-in-the-loop)

외부 시스템 (External System)

도메인 의미: 도메인 밖 서드파티와 레거시 서비스
AI 매핑: 외부 SaaS 구독형 서비스, 레거시 기간계 API

핫스팟 (Hotspot)

도메인 의미: 마찰, 지연, 병목, 미해결 위험 지점
AI 매핑: 자동화와 혁신의 핵심 후보 지점

바운디드 컨텍스트 (Bounded Context)

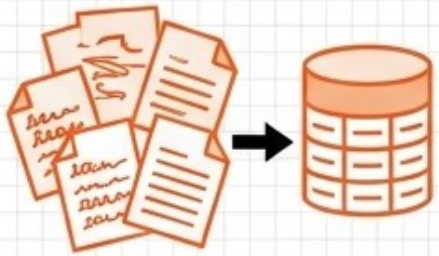
도메인 의미: 용어와 모델이 유효한 경계
AI 매핑: 단일 에이전트의 책임 영역 또는 멀티 에이전트 팀 우선순위와 경계

[출처: 원문 표 1]

MEngine

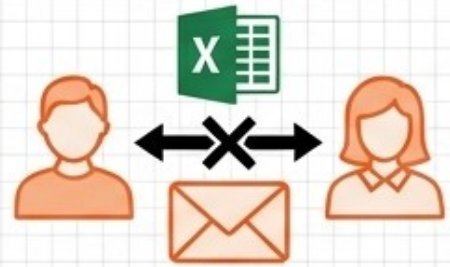
현장의 숨겨진 낭비: 핫스팟(Hotspot)이 발생하는 4가지 지점

매뉴얼과 다른 실제 업무(As-Is)의 우회로, 단절, 비공식 협업이 에이전트 도입의 가장 좋은 단서가 됩니다.



1. 비정형 데이터의 수작업 처리

형식이 정해지지 않은 자료를 사람이 직접 읽고 옮기며 처리가 지연되는 지점.



2. 엑셀과 이메일 기반의 수동 인계

부서와 시스템 사이의 단절된 업무를 파일과 메일로 넘기며 발생하는 마찰.



3. 규칙화되지 않은 암묵적 의사결정

판단 기준이 담당자의 경험에만 의존하여 문서나 시스템에 존재하지 않는 지점.



4. 여러 화면 대조에 따른 인지 과부하

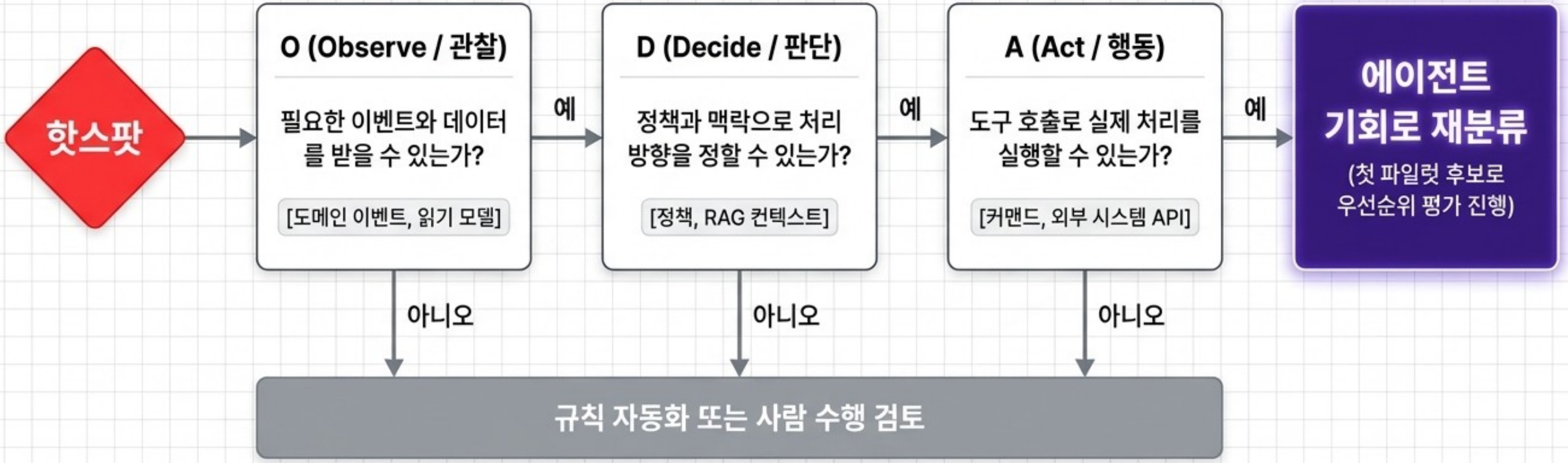
여러 모니터와 시스템의 정보를 사람이 직접 맞춰 보며 판단해야 하는 지점.

[출처: 원문 10쪽]

AIEngine

기회 검증: 에이전트 적합성을 판별하는 O.D.A 프레임워크

발견된 핫스팟마다 에이전트가 스스로 관찰, 판단, 행동할 수 있는지 검증합니다.



출처: 원문 3절, Orium | RAG: 검색한 자료를 근거로 답변 생성

[출처: 원문 3질, Orium] | ※ RAG: 검색한 자료를 근거로

역할 분담 매트릭스: 업무 성격에 따른 자동화 유형 구분

무조건적인 AI 도입을 지양하고, 의사결정의 성격과 데이터 형태에 따라 최적의 주체를 선정합니다.

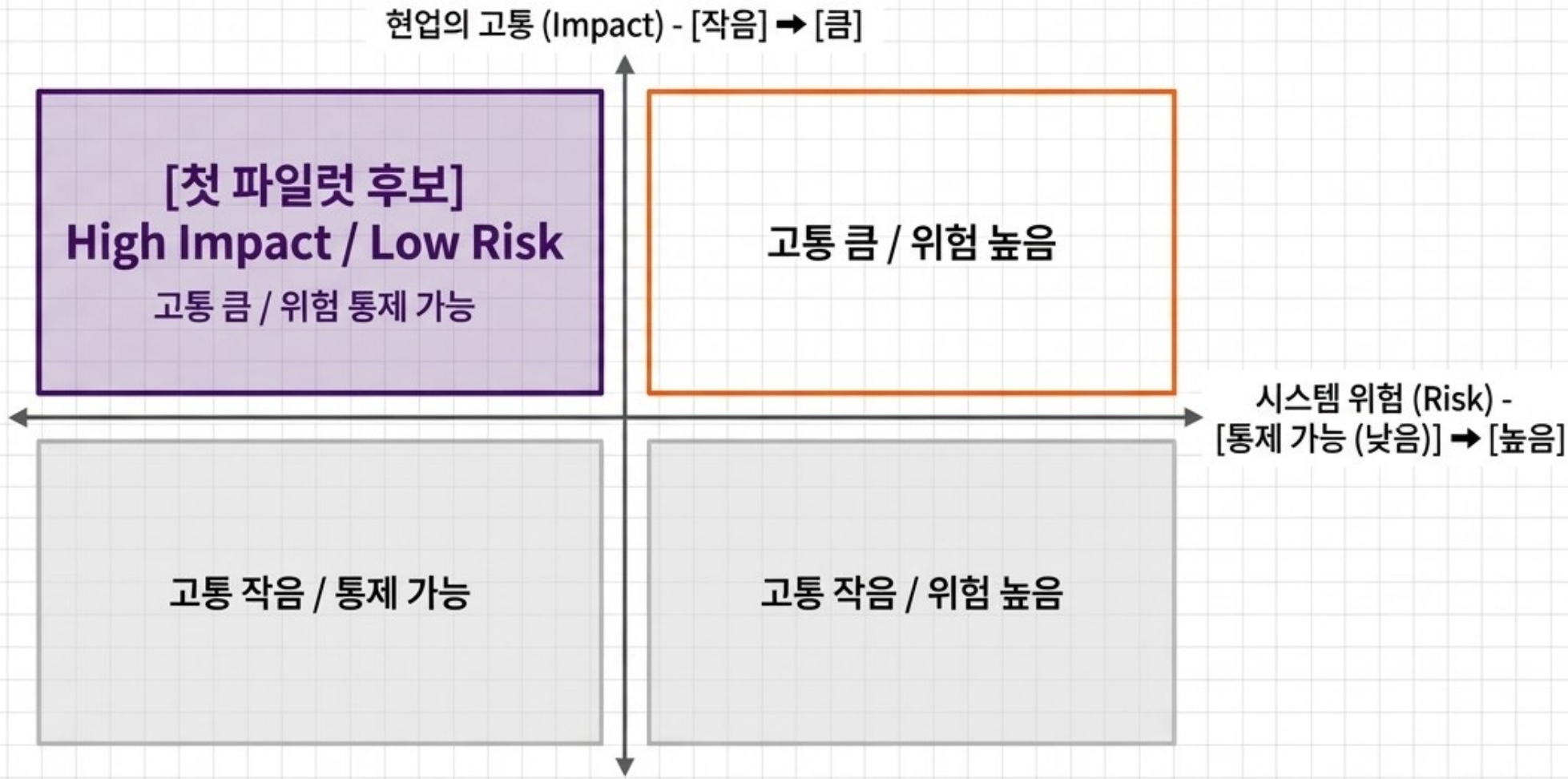
[구분]	[규칙 자동화와 RPA]	[지능형 AI 에이전트]	[사람 수행]
[의사결정 성격]	결정론적 (같은 입력에 같은 결과)	맥락적, 적응적	전략적, 정무적 판단
[데이터 형태]	정형 데이터	비정형 데이터	암묵지, 복잡한 이해관계
[실패 위험 허용도]	매우 낮음	중간 (가드레일과 사람 개입으로 보완)	매우 낮음 (법적, 윤리적 책임 집중)
[주요 기술]	워크플로 엔진, ETL 스크립트	LLM, LangGraph, 멀티 에이전트 프레임워크	전문 인력, 작업공간, 내부 결재선
[대표 적용 업무]	계정 프로비저닝, 단순 데이터 이관, 정형 고정 금액 승인	비정형 클레임 심사 보조, 공급망 병목 예측, 복합 여정 재조정	대규모 여신 심사 최종 승인, 조직 개편, 신규 도메인 전략 수립

[출처: 원문 표 2] | ※ RPA: 정해진 화면 조작을 반복 실행하는 소프트웨어 로봇. ETL: 데이터 추출/변환/적재.

MEngine

전략적 결정: 성공적인 첫 파일럿 과제 선정 기준

위험은 철저히 통제하면서도 현업의 체감 효과(Impact)가 가장 높은 과제에서 출발하십시오.



[출처: 원문 3절 우선순위 원칙 및 11쪽]

UEngine

결론: 엔터프라이즈 AI의 진정한 성패

엔터프라이즈 AI의 성패는 모델의 크기(Size)에 있지 않습니다. 업무의 경계와 책임을 에이전트에게 얼마나 명확히 전달하는가에 달려 있습니다.

[사실(Fact)]: 기술 중심 도입은 맥락 과부하와 책임 흔선을 유발함.

[저자의 해석(Insight)]: EventStorming을 통한 고도의 '경계 설계'와 '설계 재료의 명확화'만이 환각을 막고 현장의 낭비를 제거하는 유일한 해법입니다.

MEngine

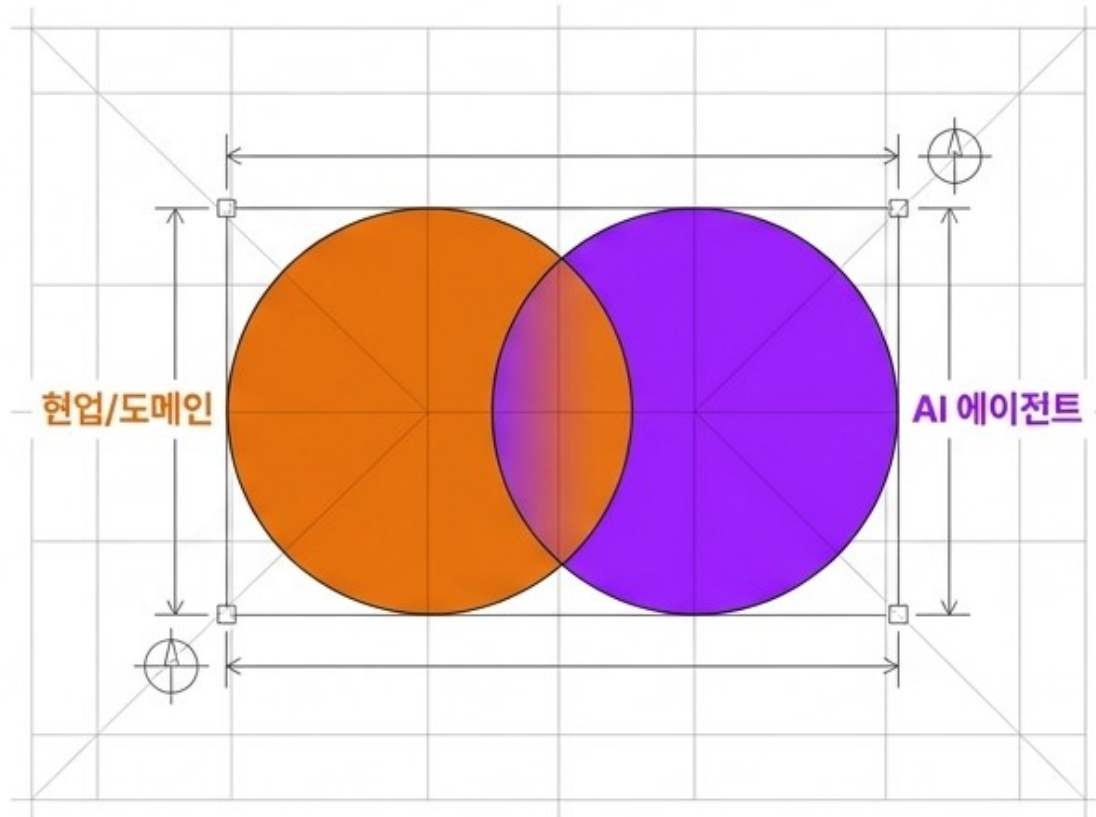
바운디드 컨텍스트(Bounded Context) 기반의 멀티 에이전트 오케스트레이션 방법론

도메인과 에이전트는 일대일로 고정되지 않습니다

본 방법론이 제안하는 컨텍스트 단위 구성은
절대적인 기술 표준이 아닌, 비즈니스 목적에 맞춘
‘설계 가설(Design Hypothesis)’입니다.

에이전트의 경계는 다음 5가지 기준에 따라
지속적으로 검증되고 유연하게 재조정되어야 합니다.

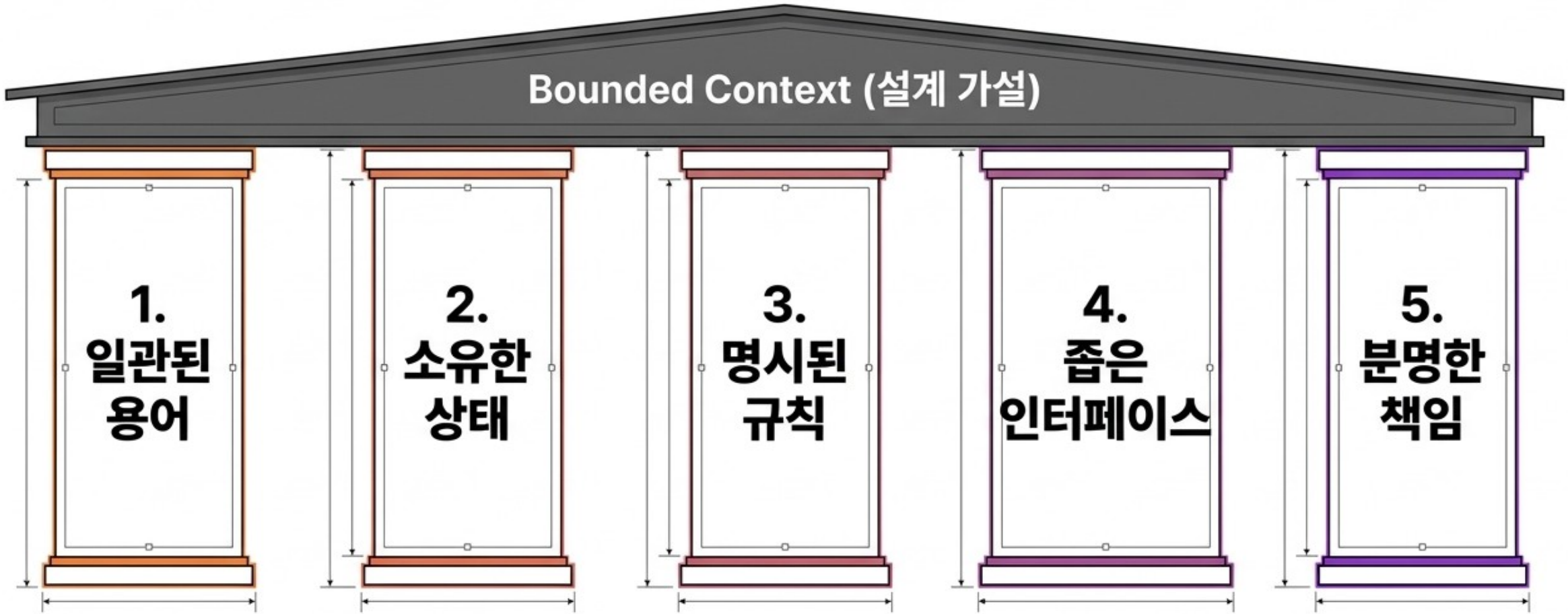
- 권한 (Authority)
- 상태 (State)
- 비용 (Cost)
- 위험 (Risk)
- 책임 (Responsibility)



MEngine

완벽한 통제와 자율성의 균형을 잡는 5가지 필수 조건

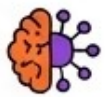
효과적인 에이전트 구성 단위를 결정하기 위해서는 다음 5가지 요소가 하나의 경계 안에서 완벽히 일치해야 합니다.
도메인 주도 설계(DDD)의 바운디드 컨텍스트는 이를 하나로 묶는 훌륭한 설계 가설을 제공합니다.



MEngine

조건 1 & 2: 일관된 언어와 소유권의 확립

1. 일관된 용어

	- **정의:** 프롬프트, 도구 설명, 데이터의 용어가 하나의 뜻으로 통함
	- **DDD 요소:** 유비쿼터스 언어 (Ubiquitous Language)
	- **막는 문제:** 맥락 과부하, 용어 혼선
	- **이벤트 스토밍 발견 지점:** 용어 변화 지점

2. 소유한 상태

	- **정의:** 에이전트가 수정할 수 있는 데이터 범위가 정해짐
	- **DDD 요소:** 애그리게잇 (Aggregate)
	- **막는 문제:** 결정 충돌
	- **이벤트 스토밍 발견 지점:** 애그리게잇

UEngine

조건 3, 4, 5: 규칙의 통제와 명확한 책임 소재

3. 명시된 규칙

- 정의: 지켜야 할 업무 규칙이 경계 안에 모여 있음
- DDD 요소: 정책, 불변식 (Policy)
- 막는 문제: 업무 규칙 위반
- 이벤트 스토밍 발견 지점: 정책

4. 좁은 인터페이스

- 정의: 다른 단위와는 정해진 사건으로만 주고받음
- DDD 요소: 도메인 이벤트 (Domain Event)
- 막는 문제: 맥락 단절, 의존 얽힘
- 이벤트 스토밍 발견 지점: 도메인 이벤트

5. 분명한 책임

- 정의: 담당 팀과 사람의 승인 지점이 정해짐
- DDD 요소: 컨텍스트 소유 팀
- 막는 문제: 책임 혼선
- 이벤트 스토밍 발견 지점: 행위자 (Actor)

MEngine

무엇을 기준으로 에이전트를 나눌 것인가?

에이전트의 '수'가 아닌 '경계의 위치'를 바꾸는 것이 성능, 비용, 책임 문제를 함께 해결하는 핵심입니다.

구성 방식	일관된 용어	소유한 상태	명시된 규칙	좁은 인터페이스	분명한 책임	에이전트 수를 정하는 근거
단일 만능 에이전트	○	○	○	○	○	근거: 항상 하나
도구와 시스템 단위	◑	◑	◕	◕	◕	근거: 연결할 시스템 수
직무와 페르소나 단위	◑	◑	◕	◕	◕	근거: 사람 조직의 직무 수
작업 단계 단위	◑	◑	◕	◕	◕	근거: 실행 때 쪼갠 작업의 수
바운디드 컨텍스트 단위 (제안)	●	●	●	●	●	근거: 업무 경계의 수

MEngine

[팩트 검증] 하위 작업이 완벽히 독립적일 때의 예외

기존 분할 방식 중 **'작업 단계 분할'**은 하위 작업이 서로의 결정에 영향을 주지 않는 **'독립성'**이 완벽히 보장될 때만 유효한 경계가 될 수 있습니다.

[Anthropic 사례 분석]

"Anthropic의 조사 시스템은 서로 독립적인 검색 작업을 하위 에이전트에 병렬로 나누어 수행하게 함으로써, 단일 에이전트 대비 90.2% 높은 성과를 기록했습니다."

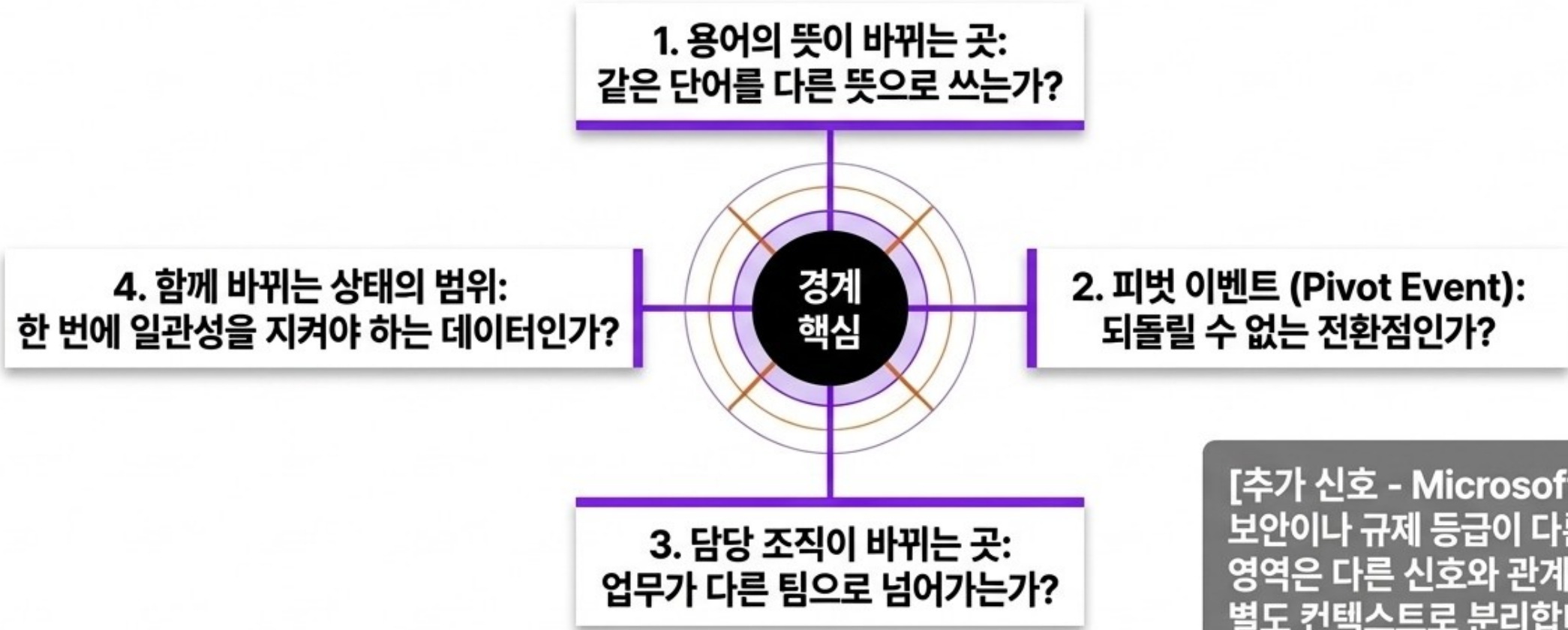
(출처: Anthropic, 2025)

이는 독립성이 곧 경계의 조건이 될 수 있음을 증명하는 특수 사례입니다.

MEngine

분할의 타이밍: 경계를 긋는 4가지 명확한 신호

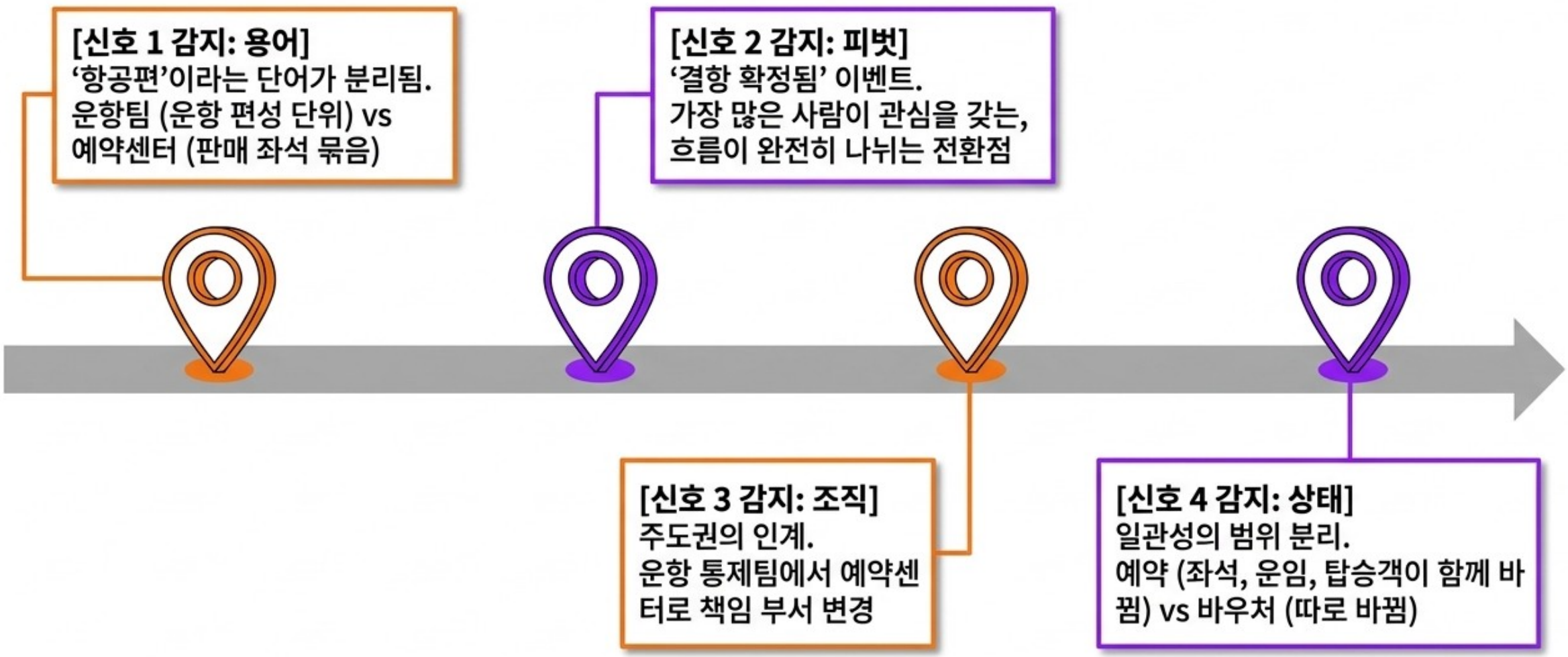
**여러 신호가 중첩되는 곳이 가장 확실한 경계 후보이며,
최종 경계는 반드시 '도메인 전문가'가 확정합니다.**



[추가 신호 - Microsoft 기준]
보안이나 규제 등급이 다른 데이터
영역은 다른 신호와 관계없이
별도 컨텍스트로 분리합니다.

MEngine

실전 예시: 항공 결항 시나리오에서의 신호 감지



UEngine

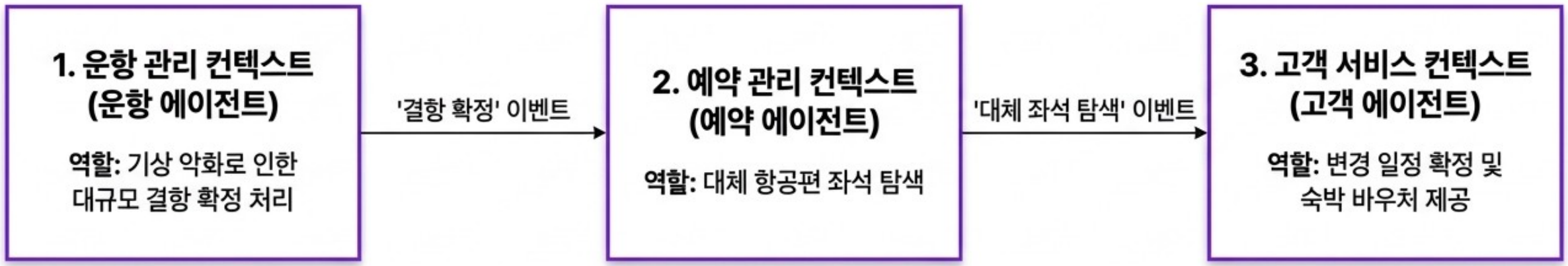
"단일 에이전트에서 시작하라"

Microsoft와 OpenAI의 벤더 가이드는 공통적으로
'단일 에이전트로 시작하여, 분명한 경계가 있을 때만 나눌 것'을 권고합니다.

Microsoft Cloud Adoption Framework	**보안/규제 경계**, **다른 담당 팀**, **계획된 확장**이 있을 때 분리합니다. 계획이나 실행 같은 **단순한 역할 분리**만으로는 나누지 않습니다.
OpenAI 에이전트 구축 가이드	**단일 에이전트**를 **최대한 활용**하십시오. **조건 분기가 너무 복잡해지거나**, **겹치는 도구**로 인해 **선택 오류**가 발생할 때가 분리할 신호입니다.

MEngine

✅ 최적화된 아키텍처: 컨텍스트 경계 기반 다중 에이전트

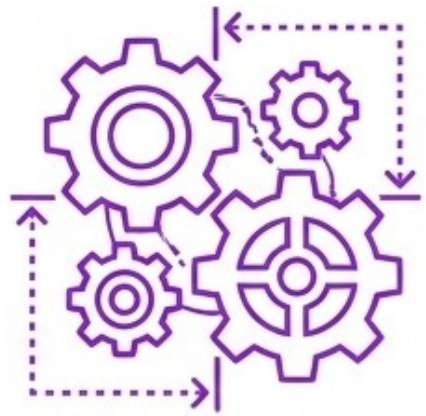


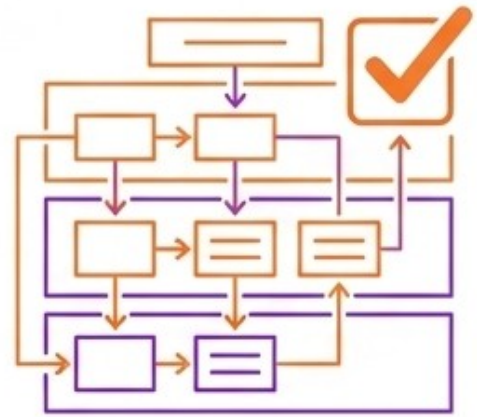
[유연한 통신 프로토콜] 각 컨텍스트 사이는 '결항 확정', '대체 좌석 탐색'과 같은 사전에 정의된 도메인 이벤트를 주요 통신 방식으로 사용하여 의존성을 최소화합니다. (※ 도메인 이벤트가 중심이 되지만, 시스템 특성과 설계 가설에 따라 다른 통신 방식도 보완적으로 활용될 수 있습니다.)

MEngine

성공적인 오케스트레이션 설계를 위한 3대 요건







1. 현장의 사건에서 출발하라

기술 부서의 상상력이 아닌,
실제 업무의 맥락과 피벗 이벤트를 기반으로
설계 가설을 세우십시오.

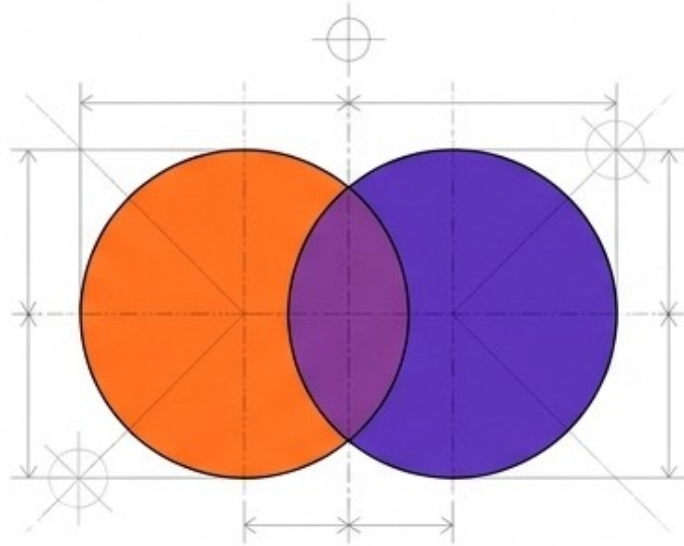
2. 경계를 유연하게 검증하라

도메인과 에이전트는 1:1이 아닙니다.
권한, 상태, 비용, 위험에 따라 가장 안전하고
효율적인 통제 경계를 확정하십시오.

3. 단일에서 다중으로 진화하라

막연한 다중 에이전트 도입을 지양하고,
명확한 신호(규정 충돌, 조직 변경)가 나타날 때
오케스트레이션을 분리하십시오.

MEngine



Agent Storming: 사람과 AI의 새로운 협업 청사진

기업 경영진과 현업 리더를 위한 가치 흐름 혁신 방법론 (제4부 도입, 실행과 학습)

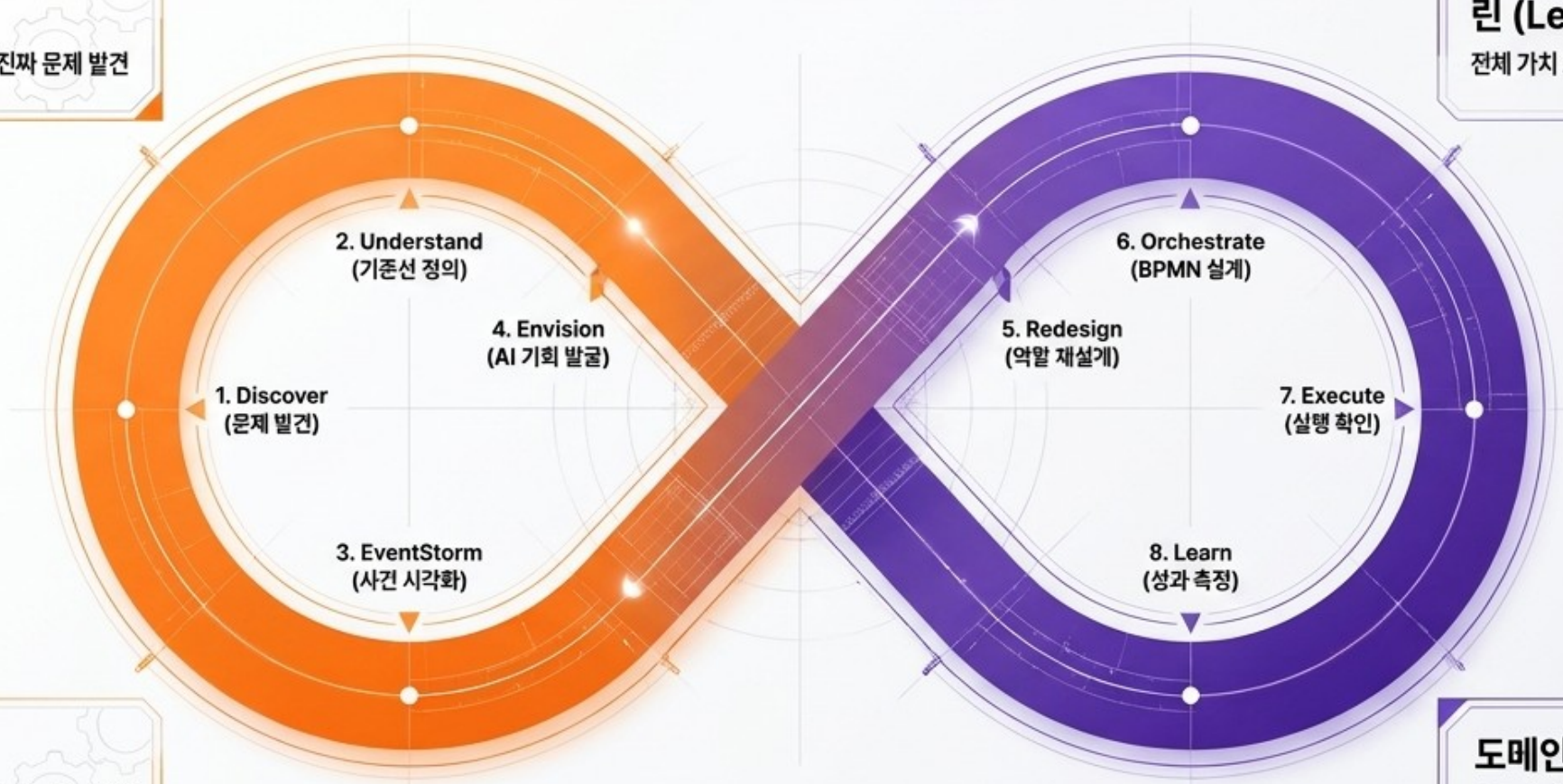
정의	현장 참여자들이 실제 업무 흐름을 발견하고, 사람과 AI 에이전트가 협업하는 프로세스를 설계·실행·개선하는 참여형 혁신 방법론.
주의	본 문서는 제안된 방법론 초안이며 검증된 표준이나 Process GPT의 확정된 현행 기능 명세가 아님을 밝힙니다.

UEngine

가치 흐름을 재구성하는 8단계 혁신 청사진

디자인 씽킹
현장 관찰과 공감을 통한 진짜 문제 발견

린 (Lean)
전체 가치 흐름 파악 및 낭비 산제적 제거



이벤트스토밍
참여자 간 공동 발견 및 사실 기만의 시각화

도메인 주도 설계 (DDD)
공통 언어 정립과 명확한 모델 경계

MEngine

퍼플 존 (Purple Zone): 기존 업무를 파괴하지 않는 AI 기회 발견



[주의] 도메인(DDD Bounded Context)과 에이전트는 1:1로 동일시되지 않습니다.
하나의 컨텍스트에 여러 에이전트가 존재하거나, 여러 컨텍스트를 넘나드는 유연한 매핑 구조를 가집니다.

MEngine

완벽한 통제: 자율성의 경계와 사람의 최종 책임

역할 분담 매트릭스 (Role Matrix)

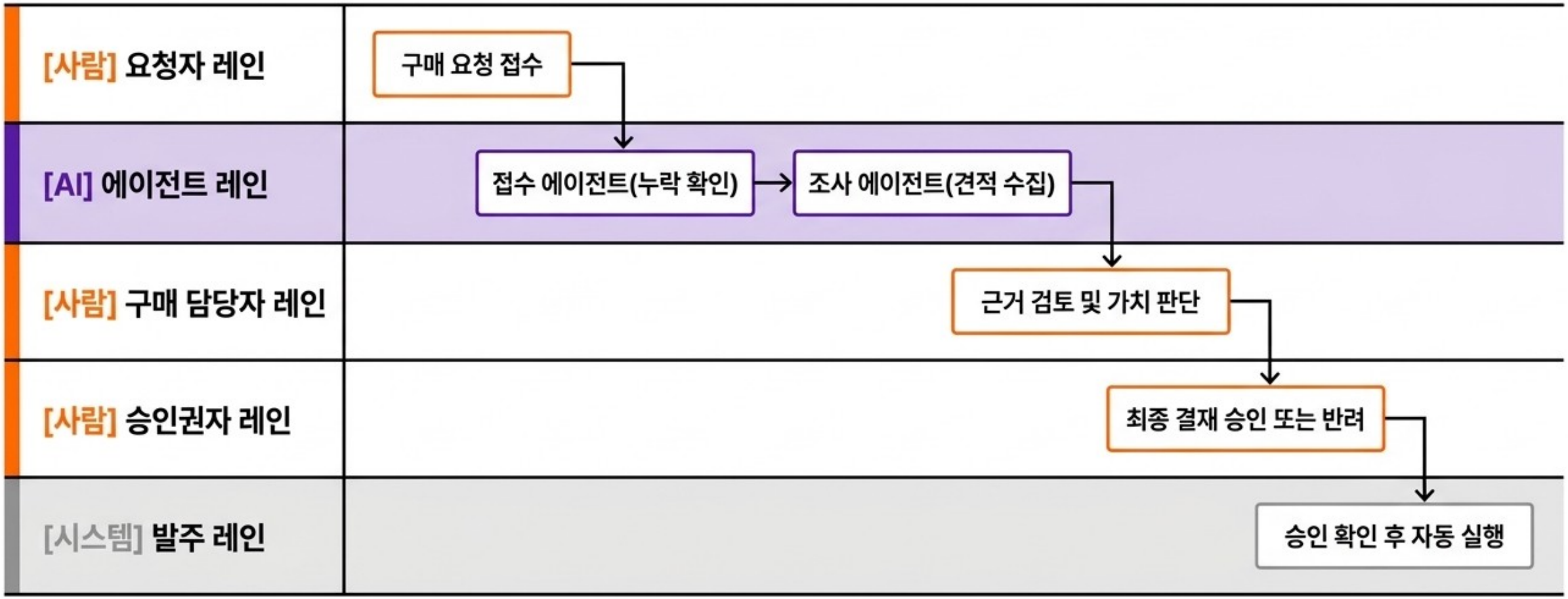
사람 (Human)	주요 임무: 가치 판단, 예외 처리 결정 경계: 최고 권한, 최종 승인 권한 보유
에이전트 (AI Agent)	주요 임무: 패턴 분석, 초안 작성, 반복 수행 결정 경계: 제한된 자율성, 사람의 승인 조건 하에 작동
외부 시스템 (System)	주요 임무: 기록 보존, 도구 실행 결정 경계: 사람의 최종 승인 후 자동 작동

에이전트 캔버스 핵심 명세

- 명확한 목표와 권한 범위
- 결정 규칙 및 지식 출처 (RAG)
- 사람 승인 조건 (Human-in-the-loop)
- 오류 시 에스컬레이션 경로 (Fallback)

MEngine

오케스트레이션: 5개의 레인(Swimlane)으로 구분된 투명한 업무 흐름



누가, 언제, 무엇을 하는지 완벽하게 분리하여 AI의 개입 시점과 사람의 판단 시점을 명확히 규정합니다.

MEngine

예외 처리와 승인: 견고한 안전망의 구축

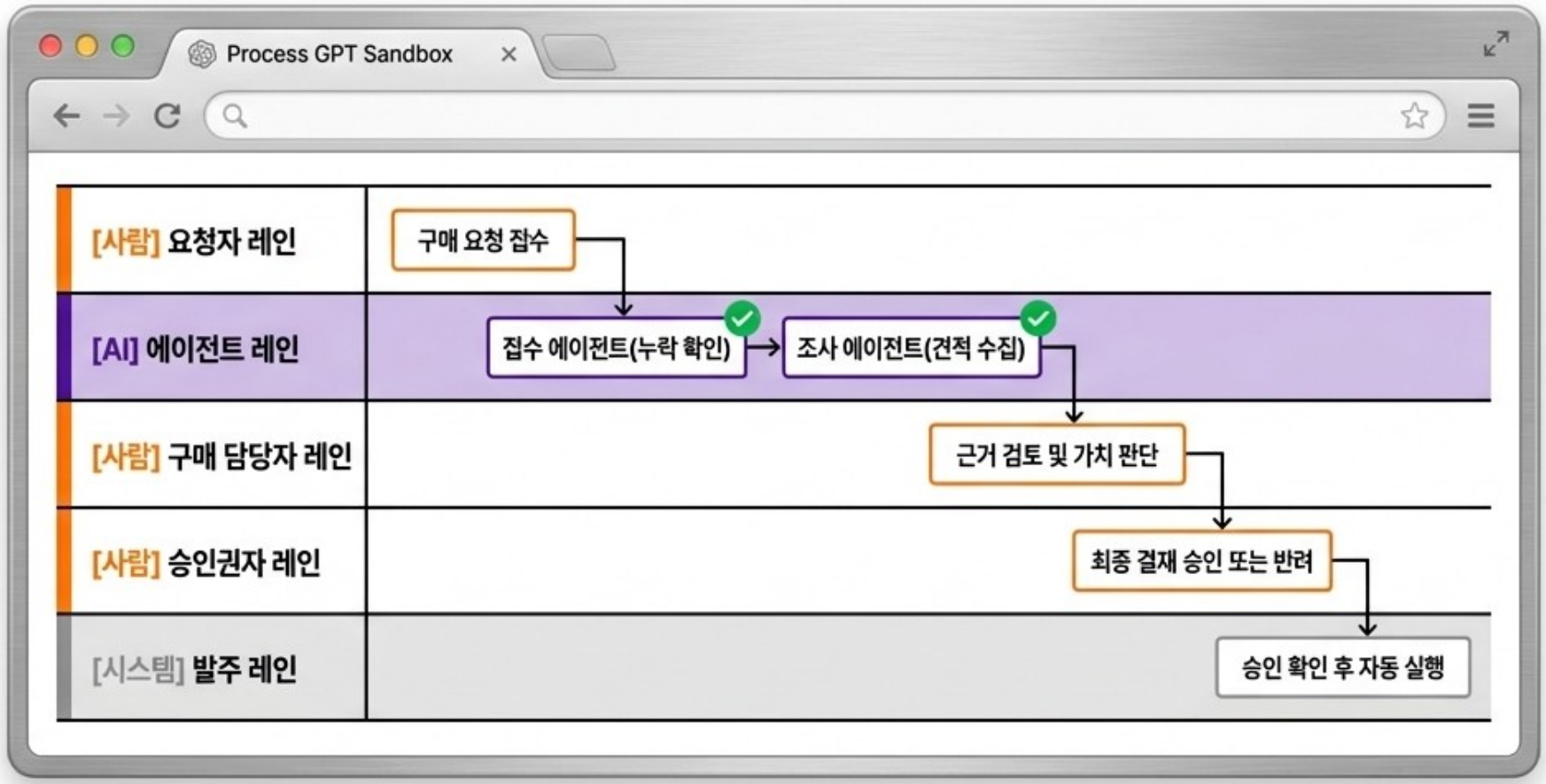


AI는 빠르고 효율적인 실행을 돕지만, 예외 상황(Exception)과 최종 결정은 항상 설계된 규칙에 따라 '사람'에게 회귀(Fallback)합니다.

MEngine

실행: Process GPT를 통한 가설 검증 샌드박스

거대한 '빅뱅' 도입 대신, 설계된 BPMN 모델을 플랫폼에서 가장 작은 단위로 시뮬레이션하여 실제 지원 여부를 검증합니다.



[제안 단계의 가설 명시] 본 자료에 명시된 Process GPT의 기능 및 실행 화면은 실제 검증 전 제안 단계의 가설(Proposed Hypothesis)이며, 확정된 현행 기능 명세가 아님을 밝힙니다.

WEngine

4

제4부 도입, 실행과 학습

엔터프라이즈 전사 확장을 위한 5단계 로드맵

완벽하게 설계된 청사진을 실제 비즈니스 환경에 어떻게 안전하게 이식할 것인가? 단일 파일럿에서 전사적 오케스트레이션으로 나아가는 구체적 경로를 제시합니다.

UEngine

전사 AI 에이전트 도입 5단계 로드맵



[안전한 확장의 법칙]

전체를 한 번에 전환하지 않습니다. 단일 파일럿을 제한된 그룹에 우선 배포한 뒤,
4주 단위 평가로 조정하며 인접 컨텍스트로 신중하게 넓혀갑니다.

MEngine

기반 구축: 업무와 언어의 일치 (단계 1~3)

단계 1: 정렬 (Align)

핵심 활동

- 빅 픽처 이벤트 스토밍
- 업무 이벤트 도출
- 컨텍스트 경계 식별

주요 산출물

- 통합 타임라인 보드
- 컨텍스트 후보안
- 유비쿼터스 언어 용어집

단계 2: 우선순위화 (Prioritize)

핵심 활동

- 핫스팟을 에이전트 기회로 전환
- 임팩트/위험 평가
- 파일럿 선정

주요 산출물

- 에이전트 기회 백로그
- 단일 파일럿 정의서
- KPI 핵심 성과 지표

단계 3: 아키텍처화 (Design)

핵심 활동

- 이벤트 스키마/상태 모델 정의
- 승인 가드레일 정의

주요 산출물

- Architecture on a Page
- OpenAPI 명세
- 가드레일 명세

실행과 검증: 데이터 기반의 확장 (단계 4~5)

단계 4: 구현 (Build)

핵심 활동

RAG 검색 기반 생성
오케스트레이션 구축
도구 호출 연동
제한된 사용자 배포

주요 산출물

워크플로우 코드
모니터링 대시보드
성능 검증 평가 슈트

단계 5: 반복과 확장 (Iterate & Scale)

핵심 활동

4주 단위 운영 데이터 평가
프롬프트/가드레일 조정
인접 컨텍스트 확장

주요 산출물

에이전트 성능 평가 리포트
컨텍스트 매핑 다이어그램
거버넌스 지침

단계별 결과를 통해 전사 공통 이해를 형성하고, 실운영 환경에서 완벽히 작동하는 초기 에이전트를 확정합니다.

MEngine

도입 한계와 전략적 대응 매트릭스

도입 한계 및 위험 (Limits/Risks)	전략적 방법론 대응 (Responses)
Human Reality	**AI Agent Intervention**
[콜드 스타트] 백지 시작 시 심리적 저항, AI 앵커링 편향으로 예외 처리 누락	AI 초안은 단지 '출발점'으로만 사용. 예외 처리는 반드시 현업 전문가와 함께 보완.
[빅뱅 시도] 전체 자동화 시도 시 예측 불가능성, 비용 및 디버깅 난도 급증	단일 핫스팟의 경량 파일럿으로 시작, 4주 단위 철저한 평가 후 확장.
[도메인 판단] 실제 소유권 경계와 표준 용어를 도구나 AI가 단독 결정 불가	도메인 전문가(사람)가 바운디드 컨텍스트 경계와 용어를 직접 확정.
[고위험 구간] 사람 통제 부재 시 규제 위반 및 법적 책임 발생	Human-in-the-loop (사람 승인) 가드레일을 컨텍스트 명세에 하드코딩.

AI 에이전트는 기술이 아닌,
비즈니스 사건(Event)에서 도출되어야 합니다.

출발 (Start)

막연한 기술 도입 대신,
현장의 업무 사건(Event)에서
출발하여 낭비를 찾습니다.

위치 (Place)

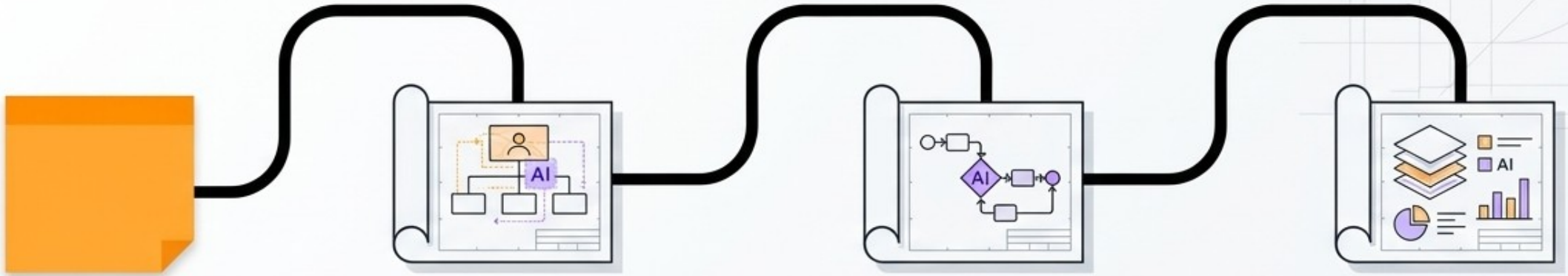
고통은 크고 위험은 통제 가능한
첫 번째 핫스팟(Hotspot)에
에이전트를 배치합니다.

경계 (Boundary)

바운디드 컨텍스트(Bounded
Context)를 경계로 삼아 단일
파일럿부터 점진적으로 넓혀갑니다.

MEngine

경영진을 위한 3대 최종 산출물 (Traceable Artifacts)



사건
(Event Fact)

1. 멀티 에이전트 협업 구조도

사람과 에이전트의 목표,
책임, 권한 명세

2. BPMN 실행 프로세스

승인 절차, 예외 처리 게이트웨이가
포함된 오케스트레이션

3. 성과 및 실험 지표 (KPIs)

도입 전 기준선(Baseline),
품질/비용 측정,
사람 개입률(Human Intervention Rate)

사건 발견부터 기회 발굴, 권한 설계, 최종 실행 지표까지 모든 산출물은 끊어짐 없이 투명하게 추적 가능합니다.

WEngine

참고 문헌 및 추가 조사 자료 (References)

[원문 바탕] uEngine 블로그: 이벤트 스토밍 기반 AI 에이전트 도출 (2026.11)

[방법론/로드맵] Orium: EventStorming for Agentic AI (2025.12) / Brandolini: Discovering Bounded Contexts with EventStorming / DDD Crew 용어집

[도메인 사례] AWS Builder Center: 항공과 여행 도메인 사례

[위험/성능 지표] Gartner 보도자료: 에이전틱 AI 프로젝트 40% 이상 취소 전망 / Cemri 외: Why Do Multi-Agent LLM Systems Fail? (NeurIPS) / LangChain: Benchmarking Single Agent Performance / Chroma: Context Rot

[멀티 에이전트 및 경계 연구]

- Kostyra: Agent as Bounded Context
- Anthropic: Building effective agents, How we built our multi-agent research system
- Cognition: Don't Build Multi Agents
- Microsoft: Single agent or multiple agents, Identify microservice boundaries
- OpenAI: A practical guide to building agents

Agent Storming 방법론 확장을 뒷받침하는 학술, 벤더 및 아키텍처 원문 자료입니다.