

AOEDE 이글 아키텍처 해설서

에이전틱 AI, 신뢰의 구조를 짓다

온톨로지에서 하네스, 그리고 현업의 책상까지 — 독수리의 해부학



비즈니스팀·중간관리자를 위한 통합 내러티브

2026년 7월

비즈니스 모델 거버넌스(유)

문서 안내

목적

이 문서는 AOEDE(Agentic Ontology Envisioning and Deployment Environment) 프레임워크의 다섯 편 내러티브 — 전체 아키텍처(이글), 테오리아, 포이에시스, 프락시스, CLAW — 를 하나의 해설서로 통합하고, 비즈니스 조직의 눈높이에서 다듬은 것입니다. 기술 사양서가 아니라, 도입을 검토하고 이끌어야 할 사람들이 프레임워크의 설계 의도와 관리 요점을 이해하도록 돕는 이야기 형식의 안내서입니다.

대상 독자

일차 독자는 **비즈니스팀과 중간관리자**입니다. 에이전틱 AI 도입 과제를 받았거나 검토 중인 사업부서장, 현업 프로세스와 리스크·컴플라이언스를 책임지는 관리자, 그리고 현업과 기술 조직 사이에서 요구사항을 번역해야 하는 기획·분석 담당자를 염두에 두고 썼습니다. 코드나 시스템 운영 경험은 전제하지 않으며, 기술 용어는 처음 등장할 때 풀어서 설명하고 부록의 용어집에 다시 정리했습니다.

문서의 구성과 읽는 방법

문서는 여섯 개의 장과 두 개의 부록으로 구성됩니다. **제1장**은 독수리 비유로 표현된 전체 아키텍처의 조감도입니다. **제2장(테오리아)**, **제3장(포이에시스)**, **제4장(프락시스)**는 각각 머리와 두 날개를 안쪽에서 들여다보고, **제5장(CLAW)**는 그 모든 것이 현업의 책상에 닿는 장면을 보여 줍니다. **제6장**은 다섯 장에 흩어진 관리자 지침을 로드맵·역할·지표·회의체의 실행 체계로 통합한 가이드입니다.

시간이 없는 독자는 **경영 요약과 제1장, 제6장**만 읽어도 의사결정에 필요한 골격을 얻을 수 있습니다. 각 장의 시작에는 다루는 내용의 예고가, 끝에는 핵심 요약이 상자로 표시되어 있으므로, 상자만 따라 읽으며 전체를 훑은 뒤 관심 있는 장으로 돌아오는 방법도 좋습니다. 본문 중 **굵은 글씨**는 그 문단의 핵심 명제를 표시합니다.

일러두기

본문의 화면 구성, 수치, 예시(주택담보대출 심사, 대출 실사, 기업 대출 워크플레이스 등)는 원 내러티브가 기술한 시점의 예시 구현을 기준으로 하며, 제품의 세부는 버전에 따라 달라질 수 있습니다. 다만 이 문서가 강조하는 것은 화면이 아니라 그 뒤의 규율 — 아키텍처로 강제되는 원칙들 — 이며, 도입 의사결정의 근거가 되어야 하는 것도 그쪽입니다. 외래 용어는 가급적 우리말로 풀되, 프레임워크의 고유 명칭(테오리아, 포이에시스, 프락시스, 넥서스, 미메시스 등)은 원명을 그대로 썼습니다. 아울러 이번 판에서는 사내 논의에서 통용되기 시작한 표기를 따라 몇 가지 용어를 통일했습니다. 능력의 구현 단위는 '능력 캡슐'로, 온톨로지의 씨앗이자 운반 형식은 '제너시스'로, 시스템에 올리는 행위는 '등록'으로, 엔티티와

관계의 구조 모델은 '비즈니스 객체모델'로 표기하며, 온톨로지 서버는 별도의 코드명 없이 'AOEDE 온톨로지 서버'로 부릅니다. 이전 자료를 함께 보시는 분들은 이 대응만 기억해 두시면 혼선이 없을 것입니다.

목차

문서 안내.....	2
목적.....	2
대상 독자.....	2
문서의 구성과 읽는 방법.....	2
일러두기.....	2
요약.....	8
이 문서가 답하려는 질문.....	8
한 마리의 독수리, 다섯 개의 기관.....	8
핵심 메시지 다섯 가지.....	9
여정의 지도: 다섯 장이 이어지는 방식.....	9
관리자가 이 문서에서 얻을 것.....	10
이 문서를 읽는 방법.....	11
문서의 구성.....	11
독자별 추천 경로.....	11
표기에 관하여.....	12
어원에 관하여.....	12
제1장. AOEDE 이글(독수리): 전체 조감도.....	14
1.1 하나의 질문에서 시작하다.....	14
1.2 왜 독수리인가.....	14
1.3 머리: 테오리아 — 판단이 행동에 앞선다.....	15
1.4 두 날개: 포이에시스와 프락시스 — 만들기과 다스리기.....	15
1.5 가슴의 벨트: 넥서스 하네스.....	16
1.6 몸통의 엔진: 미메시스 엔진.....	16
1.7 발톱: AOEDE CLAWS — 실제 비즈니스를 움켜쥐다.....	17
1.8 능력캡슐: 비유가 엔지니어링이 되는 지점.....	17
1.9 MVO 접근: 움켜질 때마다 자라나는 온톨로지.....	18
1.9.1 왜 '전부 다 설계'는 실패하는가.....	18
1.9.2 최소는 어떻게 정해지는가.....	18
1.9.3 조각들이 지도가 되는 방식.....	19
1.10 조감도를 점검표로 바꾸어 읽기.....	19
1.11 제1장을 맺으며.....	20
제2장. 테오리아: 판단하는 머리.....	21
2.1 테오리아 온톨로지 아키텍처.....	21

2.1.1 출발점: MOF, 표준에서 물려받은 기초	22
2.1.2 그림자 온톨로지: 그늘에서 인도하는 메타 프레임워크	22
2.1.3 양지의 진실 원천: 회사 비즈니스 모델과 온톨로지	23
2.1.5 두 번째 렌즈: 데이터 스트림과 운영의 실체	24
2.2 청사진이 화면이 되는 순간	25
2.3 세 개의 기둥과 하나의 순환	26
2.3.1 두 단계 모델: 메타 온톨로지와 회사 온톨로지	26
2.3.2 개념 모델: 언어의 언어를 다듬는 곳	27
2.3.3 제너시스: 온톨로지의 씨앗이자 운반 형식	28
2.3.4 시맨틱 팩토리: 문서의 세계와 모델의 세계를 잇는 다리	29
2.4 첫 화면: 온톨로지가 곧 내비게이션이다	30
2.5 작업 공간: 더블클릭으로 열리는 업무해결 방안 마법사	30
2.6 설계에서 실행 가능한 모델로: 온톨로지 상세 편집기들	31
2.7 보강과 고객 가치 실현: 온톨로지에 살을 붙이는 일	32
2.7.1 보강: 에이전트라는 새 직원의 직무기술서	32
2.7.2 고객 가치 실현(CVR): 니즈에서 에이전트 포트폴리오까지	32
2.7.3 능력캡슐 워크벤치: 지식이 최종 포장되는 곳	33
2.8 두 개의 하네스: 시스템이 무너지지 않는 이유	33
2.8.1 온톨로지 모델 하네스: MVO를 난장판에서 구하는 규율	34
2.8.2 AI 에이전트 실행 하네스: 문제 생성기가 되지 않는 조건	34
2.8.3 두 하네스는 한 벌이다	36
2.9 제너시스에서 디지털 트윈으로: 단계별 건설과 내비게이션 맵	36
2.10 온톨로지에게 직접 묻다: 트윈과의 대화	37
2.11 하루의 흐름으로 본 테오리아	38
2.12 맺으며: 머리를 맡은 관리자에게	39
제3장. 포이에시스: 만드는 날개	41
3.1 들어가며: 포이에시스가 만드는 것의 정체	41
3.2 세 단계의 하네스: 활동(워크플로우), 태스크, 능력	41
3.2.1 활동(워크플로우) 레벨 하네스: 흐름 전체를 감싸는 가장 큰 마구	42
3.2.2 태스크 레벨 하네스: 한 역할의 대리인	42
3.2.3 능력 레벨 하네스: 가장 작지만 가장 정교한 마구	42
3.2.4 포개지는 구조, 그리고 도입의 사다리	42
3.3 첫 번째 능력: 하네스 구현의 파이프라인을 한눈에 쫓다	43
3.4 두 번째 능력: 시나리오에서 출발하는 하네스 설계	43
3.5 세 번째 능력: 하네스에 넣을 모든 지식에 근거의 신분증을 붙인다	44
3.6 네 번째 능력: 스킬에서 활동까지, 일하는 주체를 조직처럼 설계한다	45
3.7 다섯 번째 능력: 통제 지점을 수학으로 버린다	46
3.7.1 의사결정 테이블의 조용한 함정	46
3.7.2 함정의 크기를 먼저 측정하다	47
3.7.3 세 가지 처방: 장부, 순서, 금지	47

3.7.4 희망 사항에서 증명 가능한 성질로	47
3.8 여섯 번째 능력: 활동 하네스의 기초를 계기판으로 조절한다	47
3.9 일곱 번째 능력: 배포하기 전에 비행해 본다	48
3.10 여덟 번째 능력: 이력과 관문으로 하네스의 출고를 통제한다	49
3.11 한 벌의 하네스가 태어나기까지	49
3.12 맷으며: 만드는 날개의 관리자에게	50
제4장. 프락시스: 다스리고 조정하는 날개	52
4.1 들어가며: 독수리의 마지막 조각	52
4.2 넥서스: 하네스의 규격이자, 하나의 원본에서 피어나는 일곱 개의 관점	52
4.2.1 아흔일곱 종의 어휘로 쓰인 계약서	53
4.2.2 두 개의 문, 그리고 자동 정렬	53
4.2.3 일곱 개의 관점: 어긋날 수 없는 문서들	54
4.3 미메시스 엔진: 작업이라는 이름의 충실한 실연	54
4.3.1 작업의 탄생: 명시된 트레이드오프	54
4.3.2 실연의 무대: 3층 구조와 살아 있는 시각화	54
4.3.3 계보가 다운로드 버튼으로 존재하는 상태	55
4.4 약속으로 묶인 실행: 요청되고, 약속되고, 진술되고, 수락된다	55
4.5 살아 있는 명부: 에이전트, 역할, 그리고 여덟 갈래의 능력	55
4.5.1 에이전트 명부: 사람과 AI가 같은 문법으로	55
4.5.2 역할 등록부: 위원회 의결까지 어휘 안으로	56
4.5.3 에이전트로지: 배우의 내면을 해부하다	56
4.6 공유 장부의 반대편: 같은 스킬을 운영의 눈으로 읽다	57
4.7 다스림의 세 도구: 변경 통제, 시그널, 감독 규칙	57
4.8 신뢰의 통장: 인증과 숙련도	58
4.9 날개 너머로: 연합과 테넌트	58
4.10 어느 오후의 실사 작업: 계보를 거슬러 오르다	59
4.11 맷으며: 독수리가 완성되다	60
제5장. CLAW: 발톱이 지면에 닿다	62
5.1 들어가며: 첫 문서의 약속을 지키는 자리	62
5.2 일터의 지도: 기업 대출이라는 업무의 전 생애	63
5.3 세 사람의 아침: 역할이 다르면 같은 시스템도 다른 얼굴	64
5.4 곁에 앉은 동료: AOEDE-CLAW 에이전트	64
5.5 한 문장의 마법: "분석 [기업명]"	66
5.6 증강의 문법: 왜 대체가 아니라 증강인가	67
5.7 발톱의 패턴: 이 예시가 보여 주는 것	68
5.8 관리자의 눈: 증강의 성과를 무엇으로 쥌 것인가	69
5.9 맷으며: 독수리가 쥐는 법을 배우는 곳	69
제6장. 통합 실행 가이드: 관리자의 책상 위에서	71
6.1 이 장의 성격	71

6.2 도입 로드맵: 증강에서 자율로 가는 사다리	71
6.3 역할과 책임: 새로 생기는 관리 업무의 주인 정하기	72
6.4 성과 지표 체계: 무엇을 재고, 무엇을 보고할 것인가	73
6.5 회의체와 운영 수칙: 다섯 번째 장의 당부를 캘린더에 올리기	74
6.6 자주 나오는 질문에 대한 답	74
6.7 맺으며: 날마다의 비행	75
부록 A. 용어집	77
부록 B. 다섯 장을 관통하는 문장들	81

요약

이 문서가 답하려는 질문

에이전틱 AI(Agentic AI), 즉 스스로 상황을 판단하고 행동하는 소프트웨어를 기업 업무에 진지하게 도입하려는 조직은 결국 하나의 질문 앞에 서게 됩니다. **스스로 판단하고 행동하는 소프트웨어에게, 행동할 자유와 신뢰할 수 있는 규율을 어떻게 동시에 부여할 것인가?**

자유만 주면 통제되지 않은 판단이 리스크를 만들고, 규율만 강조하면 자동화의 효익이 사라집니다. 이 문서는 그 딜레마에 대한 하나의 체계적인 답으로서 **AOEDE(Agentic Ontology Envisioning and Deployment Environment) 프레임워크**를 소개합니다. AOEDE는 온톨로지(비즈니스의 개념 모델)를 출발점으로 삼아, 에이전트를 설계하고(포이에시스), 운영을 다스리며(프락시스), 실제 현업 업무에 적용하는(CLAWS) 전 과정을 하나의 일관된 구조로 묶어 냅니다.

한 마리의 독수리, 다섯 개의 기관

AOEDE는 전체 아키텍처를 날개를 활짝 펼친 독수리 한 마리로 형상화합니다. 이 비유는 장식이 아니라 설계 그 자체입니다. 비행 중인 독수리가 여러 기관의 정교한 균형으로 움직이듯, AOEDE의 각 구성 요소는 서로 맞물려 하나의 시스템을 이룹니다.

독수리의 기관	AOEDE 구성 요소	역할 한 줄 요약
머리	테오리아(THEORIA)	비즈니스를 온톨로지로 정의하는 판단의 중심
왼쪽 날개	포이에시스(POIESIS)	정의된 모델을 실행 가능한 에이전트 하네스로 조립
오른쪽 날개	프락시스(PRAXIS)	배포된 에이전트의 실행을 감시·통제·거버넌스
가슴의 벨트	넥서스 하네스(NEXUS HARNESS)	행동·통제·거버넌스를 묶는 핵심 제어 계층
몸통의 엔진	미메시스 엔진(MIMESIS ENGINE)	정의된 것만을 충실히 실행하는 결정론적 동력원
발톱	AOEDE CLAWS	프레임워크가 실제 비즈니스와 만나는 접점

이 문서는 다섯 편의 내러티브 — 전체 아키텍처, 테오리아, 포이에시스, 프락시스, 그리고 발톱의 예시 구현인 CLAW(기업 대출 증강 워크플레이스) — 를 하나의 보고서로 통합하고, 비즈니스 조직과 중간 관리자의 시선에서 다시 정돈한 것입니다.

핵심 메시지 다섯 가지

이 문서 전체를 관통하는 생각은 다섯 가지로 간추릴 수 있습니다. 가장 앞자리에 오는 것은 **자동화보다 언어화가 먼저**라는 원칙입니다. 온톨로지 없이 움직이는 에이전트는 자율적인 것이 아니라 책임을 물을 수 없는 존재일 뿐이므로, AOEDE에서는 에이전트를 만들기 전에 비즈니스 자체가 먼저 명확한 언어로 정의됩니다. 이 원칙에서 자연스럽게 따라 나오는 두 번째 생각은, 우리가 만들어야 하는 것이 **에이전트가 아니라 하네스**라는 것입니다. 똑똑하지만 고삐 없는 AI는 이제 어디서나 구할 수 있습니다. 기업이 스스로 지어야 하는 것은 그 지능에 채울 마구(馬具), 곧 무엇을 알고, 무엇을 할 수 있고, 어디서 반드시 멈추는지가 짜여 들어간 구조물입니다.

세 번째 생각은 도입의 속도에 관한 것입니다. **자율성은 선언하는 것이 아니라 적립하는 것**이라는 명제 아래, AOEDE는 스킬(사람의 능력 증강)에서 태스크(한 역할의 대행)를 거쳐 활동(흐름 전체의 자율 운영)으로 오르는 세 단계의 사다리를 제시합니다. 작은 단위에서 검증된 신뢰만이 더 큰 자율성을 정당화한다는 뜻입니다. 네 번째는 온톨로지가 자라는 방식에 관한 것입니다. **온톨로지는 대성당이 아니라 지도처럼 자랍니다**. 최소 실행 가능 온톨로지(MVO) 접근은 "전체를 다 설계한 뒤 시작"하는 방식과 "통제 없는 실험"이라는 잘못된 양자택일을 모두 거부하고, 지금 필요한 스킬 하나에 충분한 만큼만 정의한 뒤 배포 하나가 성공할 때마다 지도를 한 구역씩 넓혀 갑니다.

그리고 마지막으로, **모든 결론에는 계보가 있어야 합니다**. 숫자는 상상되는 것이 아니라 계산되고, 결정은 즉흥이 아니라 근거로부터 도출되며, 그 과정 전체가 추적 가능한 기록으로 남습니다. 감사인의 어떤 질문에든 화면 하나로 답할 수 있는 상태 — 그것이 규제 산업에서 에이전트를 운영할 수 있는 조건이며, 이 문서의 모든 장이 결국 이 상태를 향해 수렴합니다.

여정의 지도: 다섯 장이 이어지는 방식

본문의 다섯 장은 각각 독립된 해설이면서, 하나의 이야기로도 이어집니다. 제1장에서 우리는 독수리의 해부 구조 전체를 조감하며, 판단이 행동에 앞서고 만들기과 다스리기가 한 벌이며 모든 결론에 계보가 있어야 한다는 프레임워크의 문법을 익힙니다. 제2장은 그 머리 안으로 들어가, 온톨로지가 정의되고 보강되고 검증되는 작업대와, 그 작업대를 무너지지 않게 붙드는 두 겹의 하네스를 봅니다. 제3장은 왼쪽 날개로 건너가, 그렇게 언어화된 업무가 스킬과 태스크와 활동이라는 세 크기의 하네스로 조립되고, 판단의 빈틈과 금지선이 수학으로 다듬어진 뒤 시험 비행을 거쳐 출고되는 공정을 따라갑니다.

제4장에서는 오른쪽 날개의 무대가 열립니다. 앞선 장들이 아껴 두었던 두 이름 — 하네스의 규격인 벡서스와 그것을 충실히 실행하는 미메시스 엔진 — 의 정체가 밝혀지고, 배포된 하네스 안에서 에이전트가 일하고 감시되고 인증되는 운영의 일상이 펼쳐집니다. 그리고 제5장에서 이 모든 구조가 마침내 현업의 책상에 닿습니다. 기업 대출 증강 워크플레이스라는 실물을 통해, 프레임워크가 한 사람의 담당자와 심사역과 리스크 관리자의 아침을 실제로 어떻게 바꾸는지를 확인하는 것입니다. 제6장은 이 여정에서 관리자가 챙겨야 할 것들을 로드맵과 역할과 지표의 실행 체계로 다시 묶습니다.

관리자가 이 문서에서 얻을 것

이 문서는 독자가 읽고 난 뒤 다음 질문들에 스스로 답할 수 있게 되는 것을 목표로 합니다.

에이전틱 AI 도입 과제가 들어왔을 때, 이것이 스킬 증강인지, 태스크 대행인지, 활동(워크플로우)의 자율 운영인지를 어떻게 판별하는가. 온톨로지 관리, 캡슐 승인, 변경 통제, 인증 갱신 같은 새로운 관리 업무의 주인은 누구여야 하는가. 증강 단계의 성과는 무엇으로 측정하고, 어떤 기록이 쌓여야 다음 단계의 자율성을 승인할 수 있는가. 그리고 감독 당국과 감사인 앞에서 "그 결정은 어떻게 나왔습니까"라는 질문에 조직이 어떤 구조로 답하게 되는가.

각 장의 말미에는 해당 영역을 맡을 관리자가 챙겨야 할 실천 항목이 정리되어 있으며, 제6장에서 이를 하나의 통합 실천 가이드로 묶었습니다.

물론 한 권의 문서가 도입의 모든 질문에 답할 수는 없습니다. 다만 프레임워크의 언어 — 하네스, 계보, 사다리, 지도 — 를 조직이 공유하게 되는 것만으로도 논의의 질은 달라집니다. "AI를 도입하자"는 막연한 문장이 "어느 스킬부터, 어느 크기의 하네스로, 무엇을 기록하며 시작하자"는 구체적인 문장으로 바뀌는 것, 그것이 이 경영 요약이 바라는 첫 번째 변화입니다.

이 문서를 읽는 방법

문서의 구성

이 문서는 여섯 개의 장과 두 개의 부록으로 구성됩니다.

제1장 「AOEDE 이글: 전체 조감도」는 프레임워크 전체의 해부 구조를 다룹니다. 독수리의 각 기관이 무엇을 뜻하는지, 능력캡슐과 MVO 접근이 왜 이 아키텍처의 핵심 설계 결정인지를 설명합니다. 시간이 부족한 독자는 이 장만 읽어도 전체의 뼈대를 파악할 수 있습니다.

제2장 「테오리아: 판단하는 머리」는 온톨로지가 정의되고 관리되는 작업 환경을 다룹니다. 개념 모델·제너시스·시맨틱 팩토리라는 세 기둥, 팀이 매일 사용하는 화면들, 시스템을 무너지지 않게 붙드는 두 개의 하네스, 그리고 디지털 트윈으로 가는 단계별 여정을 설명합니다.

제3장 「포이에시스: 만드는 날개」는 비즈니스 시나리오가 배포 가능한 에이전트 하네스로 조립되는 과정을 다룹니다. 활동·태스크·스킬이라는 하네스의 세 가지 크기와, 포이에시스가 조직에 부여하는 여덟 가지 능력을 하나씩 살펴봅니다.

제4장 「프락시스: 다스리는 날개」는 배포된 하네스가 실제로 일하는 운영의 무대를 다룹니다. 그동안 아껴 두었던 두 이름 — 하네스의 규격인 넥서스와 그것을 실행하는 미메시스 엔진 — 이 여기서 온전히 밝혀집니다.

제5장 「CLAW: 발톱이 지면에 닿다」는 발톱의 예시 구현인 기업 대출 증강 워크플레이스를 다룹니다. 추상적인 아키텍처가 현업 담당자의 아침 화면에서 어떤 모습으로 나타나는지, 그리고 '대체가 아닌 증강'이라는 설계 철학이 왜 채택 전략으로서도 옳은지를 설명합니다.

제6장 「통합 실천 가이드」는 다섯 장의 관리자 실천 항목을 하나의 운영 모델로 묶고, 도입 로드맵과 성과 지표 체계, 자주 묻는 질문을 정리합니다.

부록 A 「용어집」은 본문에 등장하는 주요 용어를 관리자의 언어로 풀이하고, **부록 B 「다섯 장을 관통하는 문장들」**은 본문 곳곳의 핵심 문장을 한자리에 모아 그 맥락과 함의를 되짚었습니다.

독자별 추천 경로

임원 및 의사결정자에게는 경영 요약과 제1장, 그리고 제6장의 도입 로드맵을 권합니다. 프레임워크의 전체 논리와 조직이 준비해야 할 것을 파악하는 데 충분합니다.

도입을 검토하는 현업 부서장과 중간 관리자에게는 전체 통독을 권하되, 자신의 역할과 가까운 장을 중심으로 깊이 읽기를 권합니다. 온톨로지와 지식 관리를 맡는다면 제2장을, 에이전트 설계와 구현을 맡는

다면 제3장을, 운영과 리스크 통제를 맡는다면 제4장을, 현업 적용과 변화 관리를 맡는다면 제5장을 중심으로 두십시오.

실무 담당자에게는 담당 영역의 장을 정독한 뒤, 부록 A의 용어집을 곁에 두고 다른 장을 참조하는 방식을 권합니다.

표기에 관하여

본문에서 고유 명칭은 처음 등장할 때 원어를 병기하고 이후 한글 표기를 사용합니다(예: 테오리아 (THEORIA)). 온톨로지, 하네스, 에이전트처럼 이미 업계에서 통용되는 외래어는 그대로 사용하되, 처음 등장하는 자리에서 뜻을 풀어 설명합니다. 각 장의 서두에는 해당 장의 핵심을 요약한 안내문이, 말미에는 관리자를 위한 실천 항목이 놓여 있습니다.

어원에 관하여

테오리아, 포이에시스, 프락시스 세 단어 모두 고대 그리스 철학, 특히 아리스토텔레스가 인간의 활동을 구분할 때 쓴 핵심 개념에서 왔습니다. AOEDE의 명명이 왜 절묘한지도 이 어원을 알면 더 분명해집니다.

테오리아(θεωρία, theoria)는 본래 "바라봄", "관조(觀照)"를 뜻합니다. 어원상 '구경하다, 관찰하다'라는 동사(theorein)에서 왔고, 원래는 축제나 신탁을 보러 가는 사절의 '봄'을 가리키던 말이었습니다. 아리스토텔레스에게 테오리아는 무언가를 만들거나 바꾸기 위해서가 아니라 진리 그 자체를 알기 위해 세계를 바라보는 활동, 즉 순수한 사유와 인식의 삶이었습니다. 그는 이것을 인간 활동 중 가장 높은 것으로 보았습니다. 영어 theory(이론)가 여기서 나왔습니다. AOEDE에서 온톨로지를 정의하는 '머리'에 이 이름을 붙인 것은, 행동에 앞서 비즈니스를 있는 그대로 '보고 아는' 단계라는 뜻과 정확히 맞아떨어집니다.

포이에시스(ποίησις, poiesis)는 "만듦", "제작"을 뜻합니다. '만들다'라는 동사(poiein)에서 왔고, 영어 poetry(시)의 어원이기도 합니다 — 시인은 말로 무언가를 '지어내는' 사람이라는 발상이지요. 아리스토텔레스의 구분에서 포이에시스의 특징은 활동의 목적이 활동 바깥의 결과물에 있다는 점입니다. 집을 짓는 행위의 목적은 짓는 행위 자체가 아니라 완성된 집입니다. 그래서 만드는 날개, 즉 하네스라는 산출물을 조립해 내는 곳에 이 이름이 붙었습니다.

프락시스(πράξις, praxis)는 "행함", "실천"을 뜻합니다. 포이에시스와 대비되는 개념으로, 목적이 결과물이 아니라 행위 그 자체와 그 행위를 잘 해내는 데 있는 활동입니다. 정치, 윤리적 삶, 공동체 안에서의 올바른 행동이 전형적인 프락시스입니다. 좋은 통치의 산출물은 어떤 '물건'이 아니라 잘 다스려지는 상태 그 자체이지요. 그래서 실천에는 프로네시스(phronesis), 곧 상황 속에서 옳게 판단하는 실천적 지혜가 요구된다고 보았습니다. 운영과 통제, 거버넌스라는 — 끝나는 시점 없이 '잘 해내는 것' 자체가 목적인 — 다스리는 날개에 이 이름이 붙은 이유입니다.

세 개념을 한 줄로 겹쳐 보면, 아리스토텔레스는 인간의 활동을 아는 것(테오리아), 만드는 것(포이에시스), 행하는 것(프락시스) 으로 나눈 셈이고, AOEDE는 그 고전적 삼분법을 에이전틱 AI의 수명주기 — 정의하고, 조립하고, 운영하는 — 에 그대로 옮겨 놓은 것입니다. 참고로 넥서스(nexus)는 라틴어로 '묶음, 연결'을, 미메시스(mimesis)는 그리스어로 '모방, 충실한 재현'을 뜻하니, 다섯 이름이 모두 같은 고전 어휘의 계보 위에 있습니다.

제 1 장. AOEDE 이글(독수리): 전체 조감도

이 장에서 다루는 것: 에이전틱 AI 도입의 근본 질문과 그에 대한 AOEDE의 답, 독수리 비유로 표현된 전체 아키텍처의 여섯 구성 요소, 능력캡슐이라는 구현 단위, 그리고 최소 실행 가능 온톨로지(MVO)라는 방법론적 핵심.

1.1 하나의 질문에서 시작하다

에이전틱 AI를 기업에 진지하게 도입하려는 조직은, 파일럿의 흥분이 가라앉고 나면 결국 하나의 질문 앞에 서게 됩니다. 스스로 판단하고 행동하는 소프트웨어에게, 행동할 자유와 신뢰할 수 있는 규율을 어떻게 동시에 부여할 것인가?

이 질문이 어려운 이유는 두 요구가 서로 반대 방향을 가리키기 때문입니다. 자유가 없는 에이전트는 기존의 규칙 기반 자동화와 다를 것이 없어 도입의 이유가 사라집니다. 반대로 규율이 없는 에이전트는, 특히 금융산업에서처럼 의사결정 하나의 비용이 자본 규모로 측정되는 산업에서는, 조직이 감당할 수 없는 리스크의 원천이 됩니다. 많은 조직이 이 사이에서 흔들립니다. 어떤 조직은 통제를 이유로 아무것도 시작하지 못하고, 어떤 조직은 속도를 이유로 통제 없이 시작했다가 첫 사고에서 프로젝트 전체를 잃습니다.

AOEDE(Agentic Ontology Envisioning and Deployment Environment) 프레임워크는 이 질문에 하나의 이미지로 답합니다. 날개를 활짝 펼친 독수리입니다.



1.2 왜 독수리인가

이 비유는 단순한 장식이 아닙니다. 비행 중인 독수리는 여러 부분이 정교하게 균형을 이루어야만 움직이는 하나의 시스템입니다. 상황을 판단하는 머리가 있고, 반드시 함께 움직여야 하는 두 날개가 있으며, 의도를 실제 움직임으로 바꾸는 몸통이 있고, 마침내 목표물을 움켜쥐는 발톱이 있습니다. 어느 하나가 빠져도 독수리는 날지 못하고, 날더라도 아무것도 쥐지 못합니다.

AOEDE는 아키텍처의 핵심 요소 하나하나를 이 구조에 대응시킵니다. 그리고 그 대응은 들여다볼수록 분명한 설계 의도를 드러냅니다. 판단이 행동에 앞서야 한다는 것(머리), 만들기과 다스리기는 한 벌이라는 것(두 날개), 모든 행동은 통제를 통과해야 한다는 것(가슴의 벨트), 실행은 충실해야 한다는 것(엔진), 그리고 이 모든 것의 가치는 결국 실제 업무를 움켜쥐는가로 평가된다는 것(발톱)입니다.

이제 각 기관을 차례로 살펴보겠습니다.

1.3 머리: 테오리아 — 판단이 행동에 앞선다

독수리의 머리에 해당하는 것이 온톨로지의 자리, **테오리아(THEORIA)**입니다.

여기서 온톨로지(ontology)란 철학 용어가 아니라 실무 용어입니다. 우리 업무 영역에 어떤 대상(엔티티)이 존재하는가, 그것들은 서로 어떻게 연결되는가, 이 영역에서 '의사결정'이란 구체적으로 무엇을 뜻하며 '근거'는 어떤 형태여야 하는가 — 이런 질문에 대한 답을 컴퓨터가 이해할 수 있는 구조로 적어 놓은 것이 온톨로지입니다. 쉽게 말해, **우리 비즈니스의 공식 지도이자 공식 사전**입니다.

AOEDE의 원칙은 명확합니다. 에이전트를 만들거나 어떤 행동을 자동화하기 전에, 비즈니스 모델 자체가 먼저 이 명확한 언어로 정의되어야 합니다. 테오리아는 바로 이 정의가 이루어지는 곳이자, 프레임워크 전체의 사고와 판단이 시작되는 개념적 중심입니다.

독수리가 날기 전에 먼저 보는 것처럼, AOEDE에서는 온톨로지가 자동화에 앞섭니다. 이 순서를 뒤집으면 어떤 일이 벌어질까요. 온톨로지 없이 움직이는 에이전트는 자율적인 것이 아니라, **단지 책임을 물을 수 없는 존재**일 뿐입니다. 그 에이전트가 스스로 무엇을 하고 있다고 믿었는지, 어떤 개념 위에서 판단했는지를 아무도 설명할 수 없기 때문입니다. 사고가 났을 때 "왜 그런 판단을 했는가"에 답할 수 없는 시스템은, 규제 산업에서는 존재할 수 없는 시스템입니다.

1.4 두 날개: 포이에시스와 프락시스 — 만들기와 다스리기

두 날개는 머리가 본 것을 실제 움직임으로 옮기며, 의도적으로 서로 상반된 역할을 맡습니다.

왼쪽 날개 **포이에시스(POIESIS)**는 '만드는 날개'입니다. 이곳에서 테오리아의 이론적 모델이 실행 가능한 컴포넌트로 바뀝니다. 개념이 실제로 작동하는 에이전트가 되는, 말하자면 조립 라인의 영역입니다. 중요한 것은 이 조립이 장인의 즉흥 작업이 아니라 **표준화된 공정**이라는 점입니다. 계약이 도출되고, 엔진이 조립되며, 모든 컴포넌트는 머리에서 정의한 온톨로지에 비추어 검증됩니다. 누가 만들어도 같은 규격의 결과물이 나오도록 공정 자체가 설계되어 있습니다.

오른쪽 날개 **프락시스(Praxis)**는 '다스리는 날개'입니다. 배포된 에이전트의 실행을 모니터링하고 관리하며, 모든 행동이 정의된 범위와 컴플라이언스 기준 안에 머물도록 보장합니다. 만드는 쪽이 "이 에이전트는 이렇게 일하기로 되어 있다"를 정의한다면, 다스리는 쪽은 "실제로 그렇게 일하고 있는가"를 매 순간 확인합니다.

이 대칭은 낯선 발명이 아닙니다. 잘 굴러가는 조직이라면 이미 알고 있는 원리 — 상품을 만드는 부서와 그 상품의 리스크를 심사하는 부서가 분리되어 있으며 같은 기준 위에서 일해야 한다는 원리 — 를 에이전트의 세계에 옮겨 놓은 것입니다. 차이가 있다면 사람의 조직에서는 이 분리가 규정과 회의로 유지되는 반면, AOEDE에서는 아키텍처로 유지된다는 점입니다.

이 대칭이 프레임워크의 핵심입니다. **통제 없이 만들기만 하는 조직은 리스크를 대량생산하고, 만들지 않고 통제만 하는 조직은 아무것도 생산하지 못합니다.** 비행에는 두 날개가 함께 움직이는 것이 필요합니다. 뒤에서 보겠지만, AOEDE는 이 대칭을 구호로 두지 않고 공학적으로 구현합니다. 두 날개는 같은 서버, 같은 데이터베이스를 공유하는 두 개의 얼굴이어서, 만드는 쪽이 넘긴 것과 다스리는 쪽이 받은 것 사이에 번역도, 복사도, 어긋날 틈도 없습니다.

1.5 가슴의 벨트: 넥서스 하네스

독수리의 가슴을 가로질러, 방패를 X자 형태로 감싸는 것이 **넥서스 하네스(Nexus Harness)**입니다. 행동(Action), 통제(Control), 거버넌스(Governance)로 이루어진 황금 벨트이자, 에이전트가 안전하고 효과적으로 작동하도록 보장하는 핵심 제어 계층입니다.

여기서 하네스(harness)라는 단어를 짚고 가겠습니다. 하네스란 말이나 견인 동물에 채우는 마구(馬具)를 뜻합니다. 마구는 동물을 묶어 두는 사슬이 아닙니다. 동물의 힘이 마차를 끄는 방향으로만 전달되도록 하는, 힘과 방향의 결합 장치입니다. AOEDE가 이 단어를 고른 이유가 정확히 그것입니다. 에이전트의 지능을 억누르는 것이 아니라, 그 지능이 조직이 허용한 방향으로만 발휘되도록 짜는 구조물 — 그것이 하네스입니다.

X자 형태에도 의미가 있습니다. 행동과 통제는 순서대로 거치는 단계가 아니라, 시스템의 심장 위에서 서로 팽팽하게 당기는 교차된 띠입니다. 행동이 커지면 통제도 함께 조여지고, 통제가 행동을 잃으면 벨트 전체가 헐거워집니다. 그리고 거버넌스는 그 긴장을 붙들어 매는 버클입니다. 에이전트가 하는 모든 일은 이 하네스를 통과하며, 하네스가 설명할 수 없는 일은 에이전트도 하지 않습니다.

1.6 몸통의 엔진: 미메시스 엔진

독수리의 중앙, 하네스 아래에서는 **미메시스 엔진(Mimesis Engine)**이 작동합니다. 넥서스를 실행하기 위해 설계된 실행 드라이버이자, 프레임워크 운영의 동력원입니다.

이름은 신중하게 선택되었습니다. 미메시스(mimesis)란 '충실한 재현'을 뜻합니다. 이 엔진은 결과를 지

어내지 않습니다. 온톨로지가 정의한 내용을, 하네스가 허용하는 범위 안에서 그대로 수행할 뿐입니다. 숫자는 상상되는 것이 아니라 계산되고, 결정은 즉흥적으로 내려지는 것이 아니라 근거로부터 도출됩니다.

이 지점에서 생성형 AI에 대한 혼란 우려 — "그렇듯하게 지어낸 답을 내놓지 않는가" — 에 대한 AOEDE의 태도가 드러납니다. 창의성이 필요한 자리와 충실함이 필요한 자리를 구분하고, 실행의 심장부에는 충실함만을 두는 것입니다. 이 엔진이 강력한 이유는 역설적으로, **정확히 시키는 대로만 움직이기 때문**입니다.

1.7 발톱: AOEDE CLAWS — 실제 비즈니스를 움켜쥐다

독수리의 구조는 결국 발톱이 무엇을 움켜질 수 있는가로 평가됩니다. 아무리 정교한 머리와 날개도, 발톱이 허공만 쥔다면 그 독수리는 비즈니스에 아무것도 가져다주지 못합니다. **AOEDE CLAWS**는 프레임워크가 실제 비즈니스와 만나는 접점입니다. 프런트 오피스와 고객 관계, 즉 CRM과 영업 환경에 직접 적용되어 수익과 고객 참여를 만들어냅니다.

왼쪽 발톱은 **전문 금융 서비스**를 담당합니다. 기업 대출, 무역 금융, 자금 관리처럼 복잡성이 높고, 판단의 근거가 계약 문서에 있으며, 통제되지 않은 의사결정의 비용이 자본 규모로 측정되는 영역입니다. 오른쪽 발톱은 **고객 측면**을 담당합니다. 고객 관리, 고객자산 관리, 그리고 포괄적 리스크 관리처럼 그 실체가 관계와 포트폴리오, 매일 변동하는 익스포저인 영역입니다.

두 발톱의 성격이 다르다는 점도 눈여겨볼 만합니다. 왼쪽은 문서와 계약이 근거의 중심인 세계이고, 오른쪽은 관계와 시계열 데이터가 중심인 세계입니다. 하나의 프레임워크가 성격이 다른 두 세계를 같은 원리 — 온톨로지, 하네스, 충실한 실행 — 로 움켜질 수 있는가. CLAWS는 그 검증의 무대이기도 합니다.

접점의 자리를 프런트 오피스로 잡은 것에도 이유가 있습니다. 백오피스의 자동화는 비용을 줄이지만, 프런트 오피스의 증강은 수익과 고객 경험을 직접 움직입니다. 그리고 프런트 오피스야말로 판단의 밀도가 가장 높은 곳 — 매일 수십 건의 조회와 비교와 우선순위 결정이 일어나는 곳 — 이어서, 계보와 통제를 갖춘 에이전트의 가치가 가장 빠르게 체감되는 지형이기도 합니다. 발톱이 먼저 닿는 자리가 조직이 프레임워크의 값어치를 처음 실감하는 자리가 되는 셈입니다.

1.8 능력캡슐: 비유가 엔지니어링이 되는 지점

CLAW는 **능력캡슐(capability capsule)**을 기반으로 구현되며, 바로 이 지점에서 독수리의 비유가 엔지니어링 현실이 됩니다.

능력캡슐이란 그 자체로 완결되고 거버넌스가 내장된 능력 단위입니다. 각 캡슐에는 업무 도메인 지식 팩과 스킬 팩을 비롯해 다음의 일곱 가지 요소가 담깁니다.

첫째, **골드 스탠더드 리포트**입니다. 이 스킬이 만들어 내야 할 결과물의 합격 기준 역할을 하는 모범 산

출물로, 캡슐의 모든 구성 요소는 이 기준을 향해 정렬됩니다. 둘째, **엔티티 모델과 데이터 계약**입니다. 이 스킴이 사용하는 비즈니스 의미에 따라 데이터가 무엇이고 어떤 형태여야 하는지를 정확히 기술합니다. 셋째, **솔루션 트리**입니다. 결론이 어떤 과정을 거쳐 도출되는지를 기록하여, 답만이 아니라 답에 이르는 길 자체를 검토 가능하게 만듭니다. 넷째, **결정론적 계산 엔진**입니다. 모든 수치를 만들어내는 계산 장치로, 같은 입력에는 반드시 같은 출력이 나옵니다. 다섯째, **시그널과 잭 감지기**입니다. 캡슐이 스스로의 상태를 감시하여 이상과 빈틈을 알립니다. 여섯째, **회귀 테스트 픽스처**입니다. 변경이 있을 때마다 기존 품질이 유지되는지를 자동으로 판정하여 배포 여부를 가늠합니다.

그리고 일곱째가 특히 중요합니다. **사람의 판단과 학습이 검토 가능한 형태로 남도록 마련된 명시적 슬롯**입니다. 업무 규칙을 담는 의사결정 테이블, 승인자가 직접 값을 채우는 거버넌스 파라미터, 아직 없는 모델에 대한 머신러닝 요구사항 등이 그것입니다. 사람의 개입이 필요할 때 그 개입은 임기응변이 아니라, 처음부터 설계된 자리에서 이루어집니다.

이렇게 보면 **각 능력캡슐은 축소된 독수리 한 마리**입니다. 골드 리포트는 테오리아의 의도가 구체화된 것이고, 캡슐의 조립은 포이에시스의 작업이며, 감사와 승인은 프락시스의 실행이고, 런타임 효과 임계값은 팽팽히 조여진 넥서스 하네스이며, 엔진은 정해진 대로 정확히 수행하는 미메시스입니다. 가장 작은 단위 안에 전체의 원리가 완결적으로 들어 있기에, 그것들을 쌓아 올린 더 큰 구조도 무너지지 않습니다.

1.9 MVO 접근: 움켜질 때마다 자라나는 온톨로지

CLAW 구현에서 가장 중요한 설계 결정은 기술이 아니라 방법론입니다. AOEDE는 **최소 실행 가능 온톨로지(MVO, Minimum Viable Ontology)** 접근을 채택합니다. 그 이유는 에이전틱 온톨로지 접근 방법이 전행적·전사적 혁신을 지향하기보다는 작게 시작하여 시행착오와 단계적 이행을 권하기 때문입니다. 아직 AI 에이전트 기술이 발전하고 있고, LLM 역시 발전 과정에 있어 전행적으로 동시에 추진하기에는 위험 요소가 많기 때문입니다.

1.9.1 왜 '전부 다 설계'는 실패하는가

엔터프라이즈 AI의 실패 사례를 보면, 첫 번째 능력을 완성하기도 전에 기업 전체를 모델링하려 했던 온톨로지 프로젝트가 적지 않습니다. 수백 명의 인터뷰, 수천 페이지의 정의 문서, 그리고 완성되기도 전에 낡아버린 수 년짜리 노력들입니다. 비즈니스는 온톨로지 모델링이 끝나기를 기다려 주지 않고, 완성된 순간의 모델은 이미 어제의 비즈니스를 그리고 있습니다.

MVO는 그 순서를 뒤집습니다. "우리 비즈니스의 완전한 온톨로지는 무엇인가?"를 묻는 대신, "**이 하나의 핵심 능력, 스킴을 신뢰할 수 있게 만드는 데 필요한 최소한의 온톨로지는 무엇인가?**"를 묻습니다. 즉, 목적지가 온톨로지를 끌어당겨 존재하게 만드는 방식입니다.

1.9.2 최소는 어떻게 정해지는가

구체적으로, 각 능력캡슐은 자신만의 최소 온톨로지를 가집니다. 그 범위는 골드 리포트와 능력캡슐이 지향하는 목표가 실제로 요구하는 엔티티, 속성, 관계, 업무 규칙, 거버넌스 파라미터 — 딱 그만큼이며, 그 이상도 이하도 아닙니다.

예를 들어 그룹 전이 리스크 매핑 능력캡슐에는 그룹 구조, 계열사 간 연결, 익스포저 상한에 대한 엔티티만 필요합니다. 이탈 리스크 평가 능력에는 계좌 잔액, 상품 보유, 접촉·상호작용 이력, 수익 레코드만 필요합니다. 각 능력캡슐에서 스킬은 은행 전체의 지도가 아니라, 자신의 목적지까지 가는 길의 지도만 요구합니다.

그렇다면 목적지가 아직 데이터에 존재하지 않는 무언가를 요구할 때는 어떻게 할까요. 예컨대 게이지 차트에 필요한 허용 한도나 궤적 분석에 필요한 전이 모델이 아직 없다면? 이때 온톨로지는 결코 **날조하지 않고, 선언으로 그 자리를 지킵니다**. 승인자가 채울 파라미터 슬롯이 만들어지거나 팀이 구축할 모델 요구사항을 명시하고, 그 값이 준비되기 전까지는 결정론적 폴백(안전한 기본 동작)이 자리를 지킵니다. 없는 것을 있는 척하지 않되, 없다는 사실 자체를 관리 가능한 형태로 등록하는 것입니다.

1.9.3 조각들이 지도가 되는 방식

이렇게 전사 온톨로지는 독수리의 악력이 강해지는 방식 그대로 — 발톱 하나, 능력캡슐 하나, 배포 하나씩 — 점진적으로 만들어집니다. CLAW 안에 캡슐이 쌓이면서 각각의 최소 온톨로지들이 겹치기 시작합니다. 이탈 고객 파악 능력캡슐의 '고객' 엔티티가 대출 능력캡슐의 '차주(obligor)' 엔티티와 만나는 순간이 옵니다. 이 둘을 조율하는 일은 이제 탁상공론식 모델링 작업이 아니라, 실제 근거가 충분한 작은 정리 작업이 됩니다. 두 능력캡슐이 이미 현장에서 검증되었으므로, 조율의 기준도 현장에 있습니다.

테오리아는 미리 지어 놓은 대성당이 아니라, **실제로 비행해 본 영토들로부터 조립되는 살아 있는 지도**가 됩니다. 머리가 발톱으로부터 배우는 것입니다.

1.10 조감도를 점검표로 바꾸어 읽기

이 조감도는 감상용 그림이 아니라, 그대로 점검표로 쓸 수 있습니다. 어떤 에이전틱 AI 과제가 책상에 올라왔을 때, 독수리의 기관을 하나씩 짚으며 여섯 개의 질문을 던져 보는 것입니다.

머리부터 묻습니다. 이 과제가 다루는 업무 영역의 대상과 관계와 판단 기준이, 담당자의 머릿속이 아니라 검토 가능한 언어로 정의되어 있는가. 다음은 왼쪽 날개입니다. 에이전트가 만들어지는 과정이 개인의 즉흥이 아니라 표준화된 공정을 따르며, 그 산출물이 온톨로지에 비추어 검증되는가. 오른쪽 날개로 넘어가서, 만들어진 것이 배포된 뒤에도 감시되고, 변경이 통제되며, 정의된 범위를 벗어나지 않도록 다스려지는가. 가슴의 벨트에 대해서는, 에이전트의 모든 행동이 통제와 거버넌스의 층을 통과하는가, 아니면 우회로가 열려 있는가를 묻습니다. 몸통의 엔진에 대해서는, 숫자와 결론이 이어지는 것이 아니라 계산되고 도출되는가, 그 과정이 재현 가능한가를 묻고, 마지막으로 발톱에 대해서는, 이 모든 구조가 결국 현업의 어떤 실제 업무를 움켜쥐는가, 그 효과를 누구의 하루에서 확인할 수 있는가를 묻습니다.

여섯 질문 가운데 어느 하나라도 답이 비어 있다면, 그 빈자리가 곧 프로젝트의 리스크가 도사린 자리입니다. 온톨로지 없이 시작한 과제는 사고가 났을 때 설명할 언어가 없고, 통제 없이 배포된 에이전트는 성실하게 잘못된 길을 가며, 발톱 없는 구조는 아무리 정교해도 성과 보고서에 쓸 문장이 없습니다. 이어지는 장들은 이 여섯 질문 각각에 대해 AOEDE가 마련해 둔 답을, 화면과 규율의 수준까지 내려가 보여 줄 것입니다.

1.11 제 1 장을 맺으며

이것이 AOEDE 이글의 조용하지만 근본적인 차별점입니다. '거대한 사전 설계'와 '통제 없는 실험'이라는 잘못된 양자택일을 거부하는 것. 프레임워크의 처방을 다섯 문장으로 압축하면 이렇습니다.

볼 수 있을 만큼만 온톨로지를 정의하십시오. 행동할 수 있을 만큼만 에이전트를 만드십시오. 모든 행동을 통제와 거버넌스에 매어 두십시오. 엔진이 충실히 실행하게 하고, 발톱이 실제 업무를 움켜쥐게 하십시오. 그리고 독수리가 실제로 쥐는 법을 배운 만큼 온톨로지가 자라나게 하십시오.

이어지는 네 개의 장에서는 이 조감도의 각 기관 안으로 들어갑니다. 제2장에서는 머리(테오리아)가 실제로 어떤 화면과 규율로 이루어져 있는지를, 제3장에서는 왼쪽 날개(포이에시스)가 하네스를 조립하는 여덟 가지 능력을, 제4장에서는 오른쪽 날개(프락시스)의 운영 무대와 함께 넥서스와 미메시스의 정체를, 그리고 제5장에서는 발톱(CLaw)이 실제 기업 대출 업무의 지면에 닿은 모습을 살펴봅니다.

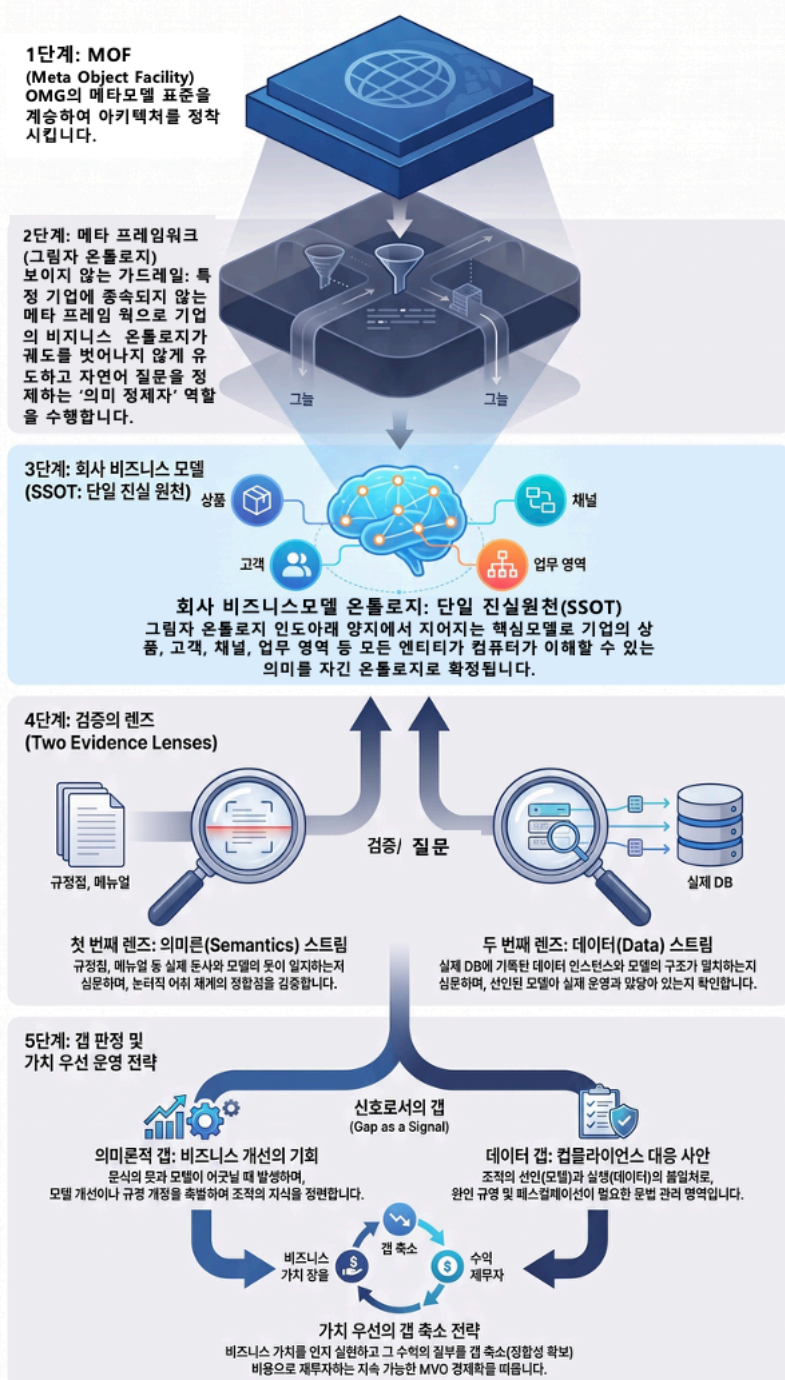
제1장 핵심 요약: AOEDE는 판단(테오리아), 제작(포이에시스), 통제(프락시스), 제어 계층(넥서스), 충실한 실행(미메시스), 현업 접점(CLawS)의 여섯 요소로 이루어진 하나의 시스템이다. 구현 단위는 거버넌스가 내장된 능력캡슐이며, 방법론의 핵심은 목적지가 요구하는 만큼만 온톨로지를 정의하고 배포 경험으로 그것을 키워 가는 MVO 접근이다.

제 2 장. 테오리아: 판단하는 머리

이 장에서 다루는 것: 온톨로지 프레임워크의 세 기둥(개념 모델, 제너시스, 시맨틱 팩토리), 팀이 매일 일하는 화면들, 시스템을 무너지지 않게 붙드는 두 개의 하네스(모델 하네스와 실행 하네스), 그리고 제너시스에서 출발하여 디지털 트윈을 단계별로 지어 올리는 여정.

2.1 테오리아 온톨로지 아키텍처

테오리아 온톨로지 아키텍처: 신뢰를 설계하는 5 단계 비즈니스 온톨로지 구조



온톨로지를 도입하려는 조직이 가장 먼저 받는 반문은 이것입니다. "그 모델은 무엇을 근거로 옳다고 말할 수 있는가?" 담당자 한 사람의 머릿속에서 나온 모델이라면 그 사람이 자리를 옮기는 순간 모델의 권위도 함께 떠나고, 컨설턴트가 그려 준 모델이라면 프로젝트가 끝나는 순간부터 낡기 시작합니다. 테오리아의 온톨로지 아키텍처는 이 반문에 구조로 답합니다. 위로는 국제 표준의 계보에 정박하고, 가운데에는 **회사 비즈니스 모델**이라는 단일 진실 원천(single source of truth)을 세우며, 아래로는 그 진실 원천을 서로 다른 관점에서 검증하는 **두 갈래의 증거 스트림**을 흘립니다. 위쪽의 정박이 "이 모델은 올바른 방법으로 적혔는가"를 담보하고, 아래쪽의 두 렌즈가 "이 모델은 뜻의 차원에서, 그리고 운영의 차원에서 현실과 맞는가"를 끊임없이 질문합니다. 그리고 이 질문에서 어긋남이 발견되는 순간이 이 아키텍처의 진짜 설계 의도가 드러나는 순간입니다 — 겹은 사고가 아니라 신호이며, 판정을 거쳐 어느 한쪽의 개선을 촉발하는 방아쇠입니다.

이 문서는 그 구조를 다섯 걸음으로 걷습니다. 표준의 기초(MOF), 그늘에서 인도하는 그림자 온톨로지(메타 프레임워크), 양지의 진실 원천(회사 비즈니스 모델과 온톨로지), 두 개의 증거 렌즈(의미론 스트림과 데이터 스트림), 그리고 겹의 판정과 가치 우선의 운영 전략입니다.

2.1.1 출발점: MOF, 표준에서 물려받은 기초

이야기의 맨 위에는 **MOF(Meta Object Facility, 메타 객체 기반)**가 있습니다. MOF는 국제 표준화 기구인 **OMG(Object Management Group)**가 제정한 메타모델링 표준으로, OMG의 **MDA(Model Driven Architecture, 모델 기반 아키텍처)** 체계의 최상위 기반을 이룹니다. 테오리아가 독자적으로 고안한 개념이 아니라, 소프트웨어 산업이 수십 년에 걸쳐 다듬어 온 표준을 계승한 것입니다.

MDA와 MOF의 핵심 통찰은 모델에도 층위가 있다는 것입니다. 맨 아래에 현실의 데이터가 있고, 그 데이터의 구조를 정의하는 모델이 있으며, 그 모델을 적는 방법을 정의하는 메타모델이 있고, 맨 위에 메타모델 자체를 기술하는 기반 층 — MOF — 이 있습니다. 층위를 섞지 않는 이 규율이, 뒤에서 볼 "정의적 질문과 인스턴스 질문의 구분"까지 이어지는 이 아키텍처 전체의 척추입니다.

관리자에게 이 출발점이 뜻하는 바는 분명합니다. **테오리아의 온톨로지는 어느 개인의 발명품 위가 아니라, 업계가 검증한 메타모델링 표준의 계보 위에 서 있습니다.** "그 방법론의 근거가 무엇입니까"라는 감사인의 질문에, 담당자의 경력이나 국제 표준의 이름으로 답할 수 있다는 뜻입니다. 물론 테오리아는 MOF의 층위 사고를 계승하되 비즈니스 온톨로지의 목적에 맞게 재해석합니다 — 무엇을 물려받고 무엇을 변형했는지를 명시하는 것이, 표준의 권위를 빌리는 올바른 방식입니다.

2.1.2 그림자 온톨로지: 그늘에서 인도하는 메타 프레임워크

MOF에 기초하여 세워지는 것이 **메타 비즈니스 모델 프레임워크**입니다. 업무대상·엔티티·속성·활동·태스크 같은 메타 객체들이 "어떤 회사 모델이든 담을 수 있는 문법"을 선언하는 층으로, 한 문장으로 요약하면 **회사에 대한 지식이 아니라, 회사에 대한 지식을 적는 방법에 대한 지식**입니다.

그런데 이 층의 전략적 정체를 정확히 짚을 필요가 있습니다. 메타 비즈니스 모델 프레임워크는 **특정 회사에 종속되지 않는(company-agnostic) 상위 온톨로지**입니다. 그 자체는 어느 회사의 실제

온톨로지도 아닙니다. 그러나 바로 그렇기 때문에, 모든 회사의 온톨로지에 대해 **그림자 온톨로지 (shadow ontology)**로 작동할 수 있습니다. 그림자 온톨로지의 역할은 두 가지입니다.

첫째는 **모델링의 가드레일**입니다. 회사의 비즈니스 온톨로지가 어떤 팀에 의해 어떤 순서로 지어지든, 그 작업은 그림자 온톨로지가 예비해 둔 완결된 상위 구조의 인도를 받아 궤도를 벗어나지 않습니다. 상향식으로 자라는 모델이 난장판이 되지 않는 첫 번째 이유가 여기에 있습니다.

둘째는 **의미의 정화기(semantic purifier)**입니다. 사용자가 자연어로 질문을 던질 때, 그 질문의 개념들은 먼저 그림자 온톨로지의 어휘에 비추어 해석되고 정련됩니다. 회사 온톨로지에 아직 정의되지 않은 개념이 질문에 등장하더라도, 상위 구조가 그것을 받아 낼 개념적 안전망을 제공하는 것입니다.

이 전략의 묘미는 그들과 양지의 관계에 있습니다. **그림자 온톨로지는 완결된 상위 구조로서 그들에 머물러 있고, 회사의 범위 온톨로지가 MVO 접근으로 정의되는 만큼만 그 개념들이 양지로 나옵니다.** 즉 완전성은 그들에 예비되어 있고, 실재성은 양지에서 한 구역씩 벌어들이는 구조입니다. 전체를 미리 다 짓지 않으면서도 전체의 인도를 받는 것 — "거대한 사전 설계"와 "통제 없는 실험"이라는 잘못된 양자택일을 거부하는 해설서의 원칙이, 이 그들과 양지의 분업으로 구현됩니다.

2.1.3 양지의 진실 원천: 회사 비즈니스 모델과 온톨로지

그림자의 인도 아래 양지에서 지어지는 것이 **회사 비즈니스 모델**입니다. 회사의 상품과 고객 세그먼트, 채널, 업무 영역과 업무대상이 메타 프레임워크가 정한 문법 안에서 정련되고 구체화되며, 그것이 컴퓨터가 이해하고 답할 수 있는 형태로 확정된 것이 **비즈니스 모델 온톨로지**입니다.

여기서 이 아키텍처의 첫 번째 원칙이 선언됩니다. **회사 비즈니스 모델이 단일 진실 원천**입니다. 아래에서 올라오는 두 증거 스트림은 진실 원천과 경합하는 또 다른 진실이 아니라, 진실 원천을 서로 다른 관점에서 검증하는 렌즈입니다. 의미론 스트림은 뜻의 관점(semantic perspective)에서, 데이터 스트림은 운영의 관점(operation perspective)에서 비즈니스 모델을 비추어 봅니다. 진실은 하나이 되 검증은 둘 — 이 비대칭이 뒤에서 볼 겹 판정 메커니즘의 전제가 됩니다.

첫 번째 렌즈: 의미론 스트림과 시맨틱 팩토리

첫 번째 증거 렌즈는 **의미론(semantic)**입니다. 이 렌즈의 표현 형식은 **개념 명료화(concept clarification)**입니다. 우리 도메인에서 '고객'과 '차주'는 같은 것의 다른 이름인가, '대출'은 '금융상품'의 하위 개념인가, 서로 배타적이라 선언한 두 개념을 동시에 상속하는 모순은 없는가 — 개념 편집기와 내장 추론기가 이런 질문에 논리적으로 검증된 답을 확정해 갑니다. 용어집을 만드는 일이 아니라, 논리적으로 검증되는 어휘 체계를 세우는 일입니다.

명료화된 개념의 근거는 **시맨틱 팩토리의 시맨틱 온톨로지**가 공급합니다. 시맨틱 팩토리는 규정집과 계약서와 업무 매뉴얼 같은 실제 문서를 읽어 엔티티와 사실과 관계를 추출하고, 모든 항목에 어느 문서의 어느 구절에서 나왔는지의 출처를 매달아 추적합니다. 원문에서 근거를 찾을 수 없는 항목은 충실성 게이트에서 기각됩니다. 이렇게 문서로부터 자라난 지식의 체계가 시맨틱 온톨로지이며, 회사 온톨로지와 별개로 관리되다가 주기적으로 대조됩니다. 문서에는 등장하는데 모델에는 없는 개

념, 문서가 말하는 관계와 모델이 선언한 관계가 어긋나는 지점이 모순 감지를 통해 수면 위로 올라옵니다. 요컨대 이 렌즈는 "우리 모델은 우리가 적어 둔 뜻과 일치하는가"를 질문합니다.

2.1.5 두 번째 렌즈: 데이터 스트림과 운영의 실체

두 번째 증거 렌즈는 **비즈니스 객체와 데이터 인스턴스**입니다. 개념이 뜻을 다룬다면 이 렌즈는 구조와 실체를 다룹니다. 각 업무대상이 소유한 비즈니스 객체모델이 그 영역의 엔티티와 속성과 관계를 구조로 선언하고, 그 구조의 실재를 실제 데이터베이스에 기록된 행들 — 데이터 인스턴스 — 이 보증합니다.

충위의 규율은 여기서도 지켜집니다. "분쟁처리에는 몇 개의 엔티티가 정의되어 있는가"는 비즈니스 객체모델이 답할 정의적 질문이고, "분쟁 사건이 지금 몇 건 쌓여 있는가"는 데이터 인스턴스가 답할 질문입니다. 두 평면을 엄격히 구분하되 위아래로 잇는 것이 정확한 답과 그럴듯한 오답을 가르는 차이입니다. 이 렌즈가 질문하는 것은 "우리 모델은 실제로 벌어지고 있는 운영과 일치하는가"입니다.

갭의 두 얼굴과 판정: 리스크가 수면 위로 올라오는 방식

이제 이 아키텍처의 심장입니다. 두 렌즈의 질문에서 어긋남 — **갭(gap)** — 이 발견되면 무슨 일이 일어나는가. 핵심은 갭의 종류에 따라 그 의미와 처리 경로가 다르다는 것입니다.

의미론 증거와 비즈니스 모델 사이의 갭은 개선의 기회입니다. 문서가 말하는 개념과 모델이 선언한 개념이 어긋났다면, 낡은 쪽이 어느 쪽인지는 열려 있습니다. 모델이 현실을 따라가지 못했을 수도 있고, 규정 문서가 개정을 놓쳤을 수도 있습니다. 그래서 이 갭의 판정은 "어느 쪽을 고칠 것인가"라는 양방향의 질문이며, 어느 쪽을 고치든 조직의 지식은 한 걸음 정련됩니다.

비즈니스 모델과 데이터 스트림 사이의 갭은 컴플라이언스 사안입니다. 모델은 조직이 "이렇게 하기로 했다"고 선언한 것이고, 데이터는 "실제로 이렇게 했다"의 기록입니다. 둘의 어긋남은 정의상 선언과 실행의 불일치이므로, 열린 조정이 아니라 준법의 언어 — 원인 규명, 시정, 필요시 에스컬레이션 — 로 다루어야 합니다.

어느 갭이든 식별되는 순간 **판정(verdict)**이 내려지고, 그 판정은 어느 한쪽의 개선을 촉발하는 방아쇠가 됩니다. 이 메커니즘의 진짜 가치는 개별 갭의 수정이 아니라 그 너머에 있습니다. 대부분의 조직에서 비즈니스는 상당 부분 암묵지 위에서 굴러가고, "규정과 실무가 다르다"는 사실은 사고가 난 뒤에야 발견됩니다. 이 아키텍처에서는 그것이 **상시 감지되는 신호**가 됩니다. 암묵 속에 잠겨 있던 잠재 리스크가 운영 중에 구조적으로 수면 위로 올라오는 것 — 갭 판정 메커니즘은 온톨로지를 문서화 도구에서 리스크 조기 노출 장치로 격상시킵니다. 마지막 조각으로, 판정의 주체와 에스컬레이션 경로 — 특히 컴플라이언스 갭을 누가 판정해 어디로 올리는지 — 를 조직의 역할표에 명시해 두면 이 메커니즘의 신뢰가 완성됩니다.

가치 우선의 갭 축소: 지속 가능한 운영 전략

갭의 발견이 축적되면 자연스러운 질문이 따라옵니다. 이 갭들을 줄이는 노력에 얼마를, 언제 쓸 것인가. 이 아키텍처가 택한 답은 정직합니다. **비즈니스 가치의 실현이 먼저이고, 실현된 가치의 일부를 갭 축소의 비용으로 배분합니다.**

이상적으로는 완전한 정합성을 먼저 확보하고 싶겠지만, 그 순서는 현실에서 지속되지 않습니다. 큰

예산을 들여 비즈니스 온톨로지를 구축했는데 비즈니스 실현이 뒤따르지 않으면, 조직은 겹을 줄이려는 관심 자체를 빠르게 잃습니다. 반대로 사용자가 온톨로지의 가치를 실제 비즈니스 케이스로 체감하면, 비정합성을 줄이려는 노력은 스스로 부지런해집니다. **겹 축소의 노력은 비즈니스 가치의 회수에 비례하여 커집니다.** 그러므로 가치를 먼저 실현하고 그 회수분으로 정합성을 사들이는 전략은, 최선(best)은 아닐지언정 지속 가능한 차선(better)입니다. 이것은 MVO의 경제학과 정확히 같은 리듬이기도 합니다 — 온톨로지가 목적지에 끌려 자라듯, 정합성 관리도 가치에 끌려 자랍니다. 조율의 예산이 아니라 조율의 동기를 설계하는 방식입니다.

그들의 인도, 양지의 진실, 두 렌즈의 질문

아키텍처 전체를 다시 접어 보겠습니다. 위에서 아래로는 정의의 계보가 흐릅니다. 국제 표준 MOF가 메타 프레임워크의 기초가 되고, 그 프레임워크는 그림자 온톨로지로서 그들에서 회사 온톨로지의 건설을 인도하며, 양지에서는 회사 비즈니스 모델이 단일 진실 원천으로 확정됩니다. 아래에서 위로 두 증거 렌즈의 질문이 올라옵니다. 의미론 스트림이 뜻의 일치, 데이터 스트림이 운영의 일치를 검증하고, 발견된 겹은 개선 기회 또는 컴플라이언스 사안으로 판정되어 어느 한쪽의 개선을 촉발합니다. 그리고 그 개선의 노력은 실현된 비즈니스 가치가 조달합니다.

관리자에게 이 구조가 주는 것은 점검의 언어입니다. 온톨로지의 품질이 의심될 때 던질 질문이 네 방향으로 정리됩니다. 문법에 맞게 적혔는가(표준의 정박), 그림자의 인도를 받고 있는가(상위 구조), 문서의 뜻과 어긋나지 않는가(의미론 렌즈), 운영의 실체와 맞닿아 있는가(데이터 렌즈). 네 질문 모두에 시스템이 답을 갖고 있다면 그 온톨로지는 믿어도 좋고, 어느 하나가 비어 있다면 그 빈자리가 곧 다음에 보강할 구역입니다. 온톨로지의 신뢰란 선언이 아니라, 그들의 인도와 양지의 진실과 두 렌즈의 질문이 매일 유지되고 있다는 사실 그 자체입니다.

2.2 청사진이 화면이 되는 순간

우리는 독수리의 해부 구조를 보았습니다. 판단하는 머리, 만들고 다스리는 두 날개, 가슴을 조이는 하네스, 충실히 실행하는 엔진, 그리고 실제 비즈니스를 움켜쥐는 발톱. 그 이야기에서 머리에 해당하는 것이 온톨로지의 자리, 테오리아였습니다. "자동화에 앞서 비즈니스가 먼저 언어화되어야 한다"는 원칙 말입니다.

이제 살펴볼 것은 그 원칙의 실물입니다. **AOEDE-테오리아 클라이언트**는 그 '머리'를 우리 팀의 브라우저 안으로 가져온 웹 애플리케이션이며, 그 뒤에서 온톨로지를 실제로 보관하고 검증하고 답변하는 것이 **AOEDE 온톨로지 서버**입니다.

이 장은 두 층을 함께 다룹니다. 화면에서 무엇을 하는가만이 아니라, 그 화면 뒤에서 어떤 규율이 작동하고 있는가까지 다룹니다. 관리자가 시스템을 신뢰하려면 후자를 알아야 하기 때문입니다. 화면은 바뀔 수 있지만 규율은 아키텍처이며, 도입 의사결정의 근거가 되어야 하는 것은 규율 쪽입니다.

이야기는 다섯 개의 큰 흐름으로 진행됩니다. 첫째, 온톨로지 프레임워크를 이루는 세 기둥 — 개념 모델,

제너시스, 시맨틱 팩토리 — 각각의 의미와, 이들이 회사의 비즈니스 온톨로지와 맺는 연결. 둘째, 그 프레임워크 위에서 팀이 매일 일하는 화면들 — 작업 공간, 다이어그램 편집기, 보강, 고객 가치 실현. 셋째, 이 전체를 무너지지 않게 붙드는 두 개의 하네스 — 온톨로지 모델 하네스와 AI 에이전트 실행 하네스. 넷째, 제너시스에서 출발하여 단계별로 디지털 트윈을 지어 올리는 여정과 그 내비게이션 맵. 마지막으로 관리자가 챙겨야 할 것들입니다.



2.3 세 개의 기둥과 하나의 순환

테오리아의 사이드바에는 개념 온톨로지, 비즈니스 온톨로지, 시맨틱 팩토리가 나란히 놓여 있습니다. 처음 보는 사람에게는 비슷비슷한 지식 메뉴로 보이지만, 이 셋은 서로 다른 층위에서 서로 다른 일을 하며, 그 관계를 이해하는 것이 테오리아 전체를 이해하는 지름길입니다.

결론부터 말하면 이렇습니다. 개념 모델은 문법이고, 제너시스는 씨앗이며, 비즈니스 온톨로지는 우리 회사라는 문장이고, 시맨틱 팩토리는 그 문장이 현실과 맞는지 대조하는 증거 보관소입니다. 이 한 문장을 기억해 두시면, 이하의 상세가 모두 이 문장의 주석이라는 것을 알게 되실 겁니다.

2.3.1 두 단계 모델: 메타 온톨로지와 회사 온톨로지

서버의 기초 설계 문서가 "이것을 잘못 이해하면 그 위의 모든 계층이 잘못된다"고 못 박아 둔 개념이 있습니다. 온톨로지는 한 층이 아니라 두 층이라는 것입니다.

위층은 메타 온톨로지입니다. '업무대상', '엔티티', '속성', '활동', '태스크' 같은, 이름 앞에 `#`가 붙는 메

타 객체들의 세계입니다. 아홉 개의 메타 비즈니스 객체모델(개체-관계 모델) 다이어그램이 하나로 연결되어, "어떤 회사 모델이든 담을 수 있는 것들의 문법"을 선언합니다. 예컨대 "하나의 업무대상은 여러 엔티티를 가진다", "사업부문은 고객 세그먼트, 상품군, 별도의 채널군을 가진다." 같은 관계와 카디널리티(연결의 개수 규칙)가 여기서 정해집니다. 이 층은 회사에 대한 지식이 아니라, **회사에 대한 지식을 적는 방법에 대한 지식**입니다.

아래층이 **회사 온톨로지**입니다. 예를 들어 '분쟁처리'라는 우리 회사의 업무대상은 두 가지 얼굴을 동시에 가집니다. 하나는 메타 객체 '#업무대상'의 인스턴스로서 데이터베이스에 기록된 한 행이라는 얼굴이고, 다른 하나는 자기 자신의 비즈니스 객체모델 다이어그램 — 분쟁 사건, 분쟁 조사 같은 자기 소속 엔티티들을 정의한 모델 — 을 소유한 하나의 모델이라는 얼굴입니다. 즉 '분쟁처리'는 메타 다이어그램과 구조적으로 같은 종류의 존재이되, 하나는 '#'가 붙은 프레임워크이고 다른 하나는 회사의 실물이라는 차이만 있습니다.

관리자에게 이 구분이 중요한 이유는 **질문의 종류가 갈리기 때문**입니다. "분쟁처리에는 몇 개의 엔티티가 정의되어 있는가?"는 모델에 대한 정의적 질문이어서 분쟁처리 자신의 비즈니스 객체모델에서 답해야 하고, "분쟁 사건이 지금 몇 건 쌓여 있는가?"는 인스턴스 질문이어서 데이터 행에서 답해야 합니다. 서버는 이 두 평면을 엄격히 구분합니다. 설계 문서의 표현을 빌리면, 이 층위 규율이 **"정확한 답과 그럴 듯해 보이는 오답의 차이"**를 만듭니다. 실제로 과거의 오답 사례들은 예외 없이 두 평면을 섞은 데서 나왔다고 기록되어 있습니다.

2.3.2 개념 모델: 언어의 언어를 다듬는 곳

개념 온톨로지(제너시스 개념 편집기)는 이 문법 층을 직접 편집하는 작업대입니다. 화면에서는 개념(concept)들과 그 사이의 상하위(IS-A) 관계, OWL 스타일의 속성, 분류 스킴, 그리고 개념과 동사의 연결을 다룹니다. 우리 도메인에서 '고객'과 '차주'는 같은 것의 다른 이름인가, '대출'은 '금융상품'의 하위 개념인가, '리스크'라는 말은 어떤 분류 체계 아래 놓이는가 — 이런 질문에 답을 확정하는 곳입니다.

이런 질문이 사소해 보인다면, 용어의 어긋남이 조직에서 일으키는 비용을 떠올려 보시기 바랍니다. 부서마다 '고객'의 정의가 달라 보고서 숫자가 맞지 않는 일, 같은 지표를 다른 이름으로 두 번 관리하는 일 — 사람 조직에서도 값비싼 이 혼선이, 에이전트의 세계에서는 곧바로 잘못된 자동 판단으로 이어집니다.

여기서 주목할 것은 편집기 안에 **추론기(reasoner)**가 내장되어 있다는 점입니다. 저장 버튼을 누르는 것으로 끝나지 않고, 서버가 개념 그래프 전체를 검사합니다. 상하위 관계에 순환이 생기지 않았는가(A가 B의 하위이고 B가 다시 A의 하위인 모순), 서로 배타적(disjoint)이라고 선언한 두 개념을 동시에 상속하는 개념은 없는가, 직접 선언하지는 않았지만 논리적으로 따라 나오는 상하위 관계는 무엇인가. 검사 결과는 일관성 여부, 추론된 관계 제안, 충족 불가능한 클래스 목록, 동치 그룹으로 정리되어 화면에 표시됩니다.

요컨대 개념 모델은 '용어집'이 아니라 **논리적으로 검증되는 어휘 체계**이며, 모든 수정은 리비전 이력으

이 양방향성이 뜻하는 바를 관리자의 언어로 옮기면 이렇습니다. **검증된 온톨로지는 복제 가능한 자산이 됩니다.** 한 사업부에서 다듬어진 여신 도메인의 모델을 패키지로 내보내 다른 법인의 저장소를 여는 씨앗으로 쓸 수 있고, 파일럿 저장소에서 실험한 구조를 본 저장소로 옮길 수 있습니다. 제1장에서 말한 MVO — 목적지가 요구하는 만큼의 최소 온톨로지로 시작한다 — 가 실무에서 가능한 것은, 그 '최소'를 담아 나르는 규격화된 그릇이 있기 때문입니다. 그렇기 때문에 MVO에 의한 AI 에이전트 적용은 즉시 효과를 보게 되고 체계적인 지속성을 갖게 됩니다.

2.3.4 시맨틱 팩토리: 문서의 세계와 모델의 세계를 잇는 다리

세 번째 기둥인 **시맨틱 팩토리**는 성격이 다릅니다. 개념 모델과 비즈니스 온톨로지가 사람이 의도적으로 설계하는 세계라면, 시맨틱 팩토리는 **문서 더미에서 지식을 캐내는 공장**입니다. 화면의 설명 문구가 그 성격을 요약합니다 — "문서 첨부에서 축적된 지식입니다."

내부 동작은 이렇습니다. 위키에 첨부된 문서(규정집, 계약서, 업무 매뉴얼 등)를 시맨틱 분석 대상으로 지정하면, 서버의 빌더가 백그라운드에서 문서를 읽고 지식을 추출합니다. 추출 단위는 세 가지입니다.

The screenshot displays the AOEDE-Theoria Semantic Factory interface. The top navigation bar includes 'Polesis' and 'Praxis' tabs, and the 'AOEDE-Theoria' logo. The main interface is divided into three main sections:

- Left Panel (시맨틱 팩토리):** Contains a search bar, a list of documents, and a table of document details. The table lists documents like '공사의 대금예', '공사 투자', '공사 건설 프로젝트', '공행', '교', '지불 시간', '거래 상대방은 서면으로 확정된 은행 이름 또는 ...', '거래 상대방', '거래의 합리성', '교통', '구', '구체적인 신용 업무', '구체적인 평가 수준', and '유효한 구속력 있는 부채 문서의 증명'.
- Center Panel (거래 상대방):** Displays the content of the selected document, '거래 상대방'. It includes a summary, a list of related documents, and a list of related terms. The summary states: "거래 상대방은 신용 대출 자금 과정에서, 대출인이 대출 자금의 최종 지급 대상입니다. 관련 규정에 따르면, 대출인은 자금 지급 거래 상대방에게서 직접 대출 자금을 지급받아서 안 되며, 어떠한 방식으로든 대출인 또는 감독 기관이 지정한 방식이 아닌 거래 상대방에게 지급해서는 안 됩니다. 대출 자금은 대출 인이 계약된 지급 방식으로 지급받아야 합니다." It also lists related terms like '대입', '자본 투자', '계약', etc.
- Right Panel (시맨틱 그래프: 거래 상대방):** Displays a semantic graph for the '거래 상대방' document. The graph shows a central node '거래 상대방' connected to various related concepts like '대입', '자본 투자', '계약', etc. The graph is a complex network of nodes and edges, representing the relationships between different concepts in the document.

첫째, 문서가 언급하는 **엔티티**입니다. 각 엔티티마다 페이지가 만들어지고, 새 문서가 들어올 때마다 그 페이지가 누적적으로 풍부해집니다. 둘째, 엔티티에 붙는 **사실**입니다. 모든 사실에는 어느 문서의 어느 구절에서 나왔는지 출처가 반드시 달립니다. 셋째, 엔티티들 사이의 **관계 트리플**입니다. 문서의 원문 표현을 보존하면서, 자체 유지되는 관계 어휘 사전에 정규화됩니다.

추출 과정에는 **충실성 게이트**가 있습니다. 원문에서 근거를 찾을 수 없는 — 즉 모델이 지어낸 — 항목은 저장되기 전에 기각되며, 기각 건수가 로그에 남습니다. AI가 문서를 읽되, AI의 상상이 지식 베이스에 스며들 수 없도록 문이 지켜지고 있는 것입니다.

그리고 결정적인 층이 하나 더 있습니다. **모순 감지**입니다. 시맨틱 팩토리가 문서에서 발견한 엔티티와 관계는 온톨로지의 엔티티와 별개로 관리되는데, 시스템은 이 둘을 주기적으로 대조합니다. 문서에는 등장하는데 온톨로지에는 없는 개념, 문서가 말하는 관계와 온톨로지가 선언한 관계가 어긋나는 지점 — 이것들이 사람이 검토할 발견 사항으로 정리됩니다.

여기서 세 기둥의 연결이 완성됩니다. 개념 모델이 문법을 정하고, 제너시스가 그 문법에 맞는 구조를 씌우므로 심어 비즈니스 온톨로지를 세우면, 시맨틱 팩토리는 **현실의 문서들이 그 온톨로지와 일치하는 지를 계속 감시**합니다. 온톨로지가 현실보다 앞서 나가면(선언했지만 증거 없는 개념) 그것이 보이고, 현실이 온톨로지보다 앞서 나가면(문서에는 있는데 모델에 없는 개념) 그것도 보입니다. 제1장에서 "머리가 발톱으로부터 배운다"고 했던 문장이, 여기서는 "모델이 문서로부터 배운다"는 구체적 메커니즘으로 구현되어 있는 것입니다.

2.4 첫 화면: 온톨로지가 곧 내비게이션이다

프레임워크를 이해했으니 이제 화면으로 들어가겠습니다. 테오리아에 로그인하면 왼쪽 아이콘 사이드바에 이 애플리케이션의 세계관이 나열됩니다. 개념 온톨로지, 비즈니스 온톨로지, IT 온톨로지, RAG 시스템, 온톨로지 프로젝트, 도메인 관리, 시맨틱 팩토리. 일반적인 업무 시스템이 '메뉴'를 나열한다면, 테오리아는 **'우리 비즈니스를 바라보는 렌즈'**를 나열합니다.

중심은 비즈니스 온톨로지 트리입니다. 상품 포트폴리오, 고객 세그먼트, 채널 분류, 시장, 업무 아키텍처의 능력 모델과 업무 영역, 업무 컴포넌트, 업무 대상 — 이 트리는 폴더 목록이 아니라 회사 구조 그 자체이며, 팀원이 어떤 작업을 하든 그 작업은 반드시 트리의 어딘가에 걸려 있습니다. "이 문서는 어느 업무의 산출물인가?"라는 질문에 파일 경로가 아니라 **비즈니스 구조상의 좌표**로 답하게 되는 것입니다.

화면 하단의 상태 바는 어느 서버의 어느 저장소에 연결되어 있는지, 지금 어떤 엔티티가 선택되어 있는지를 상시 표시하여, 여러 저장소를 오가는 팀의 실수를 줄여 줍니다.

트리가 처음 태어나는 방식은 앞서 본 대로 제너시스 부트스트랩이고, 자라는 방식은 이후 절에서 볼 보강과 편집입니다. 여기서 관리자가 기억할 것은 하나입니다. **트리는 모든 작업의 좌표계**이므로, 트리를 다듬는 권한과 책임이 명확해야 한다는 점입니다.

2.5 작업 공간: 더블클릭으로 열리는 업무해결 방안 마법사

트리에서 엔티티를 더블 클릭하면 오른쪽에 작업 공간이 열립니다. 브라우저 탭처럼 여러 개를 동시에 열 수 있어, 시나리오·프로세스 다이어그램·보강 편집기·보고서를 나란히 놓고 오갈 수 있습니다.

대표 격은 업무 해결방안 **시나리오 컴포저**입니다. 왼쪽 트리 다이어그램에서 문서의 구조를 노드로 짜고, 오른쪽 상세 패널에서 각 노드에 맥락과 질문을 적은 뒤 '마이닝'을 누르면 AI가 업무 해결방안 내용을 마이닝하여 생성합니다. 중요한 것은 AI가 백지에서 지어내지 않는다는 점입니다. 다른 노드의 내용, 첨부한 PDF, 웹 검색 결과, RAG(검색 증강 생성)로 검색한 사내 지식이 컨텍스트로 함께 들어가고, 생성된 내용은 완전성·명확성·일관성·실행 가능성의 네 축으로 품질 분석을 받을 수 있습니다.

AI 엔진 자체도 관리 대상입니다. 설정의 **브레인(Brain)** 탭에서 LLM 제공자, 온도와 토큰 한도, 시스템 페르소나를 정하며, 시나리오 단위·노드 단위로 설정을 달리 주고 '설정 전파'로 계층 전체에 내려보낼 수 있습니다. AI 활용이 개인기가 아니라 **팀 표준의 영역**이 된다는 뜻입니다. 같은 도구를 쓰더라도 어떤 팀은 표준 설정과 충실한 컨텍스트로 일하고 어떤 팀은 각자 임의로 일한다면, 산출물의 품질 격차는 도구가 아니라 관행에서 벌어집니다.

작성이 끝난 내용은 게시 미리보기로 확인하고, 노드 또는 계층 전체를 게시하며, 번역 연동으로 번역본을 만들고, 외부 게시 플랫폼으로 내보낼 수도 있습니다. 각 노드에는 초안·검토 중·승인됨·잠김의 상태가 붙어 "이 문서 어디까지 됐나?"가 상태 값 하나로 답해지고, 버전 이력에서 과거 스냅샷의 비교와 복원이 가능합니다.

2.6 설계에서 실행 가능한 모델로: 온톨로지 상세 편집기들

여기가 테오리아가 일반 문서 도구와 달라지는 지점입니다. 작업 공간에서는 글만 쓰는 것이 아니라 **실행의 재료**가 되는 모델을 그립니다.

결정 흐름

결정 모델

조건 인텔리전스

결정 감사

전체 게이트

G1: 서류 확인 (5)

G2: 신용 평가 (3)

G3: 담보 평가 (2)

G4: 대출 조건 검증 (2)

G5: 위험 평가 (5)

G6: 최종 승인 (2)

● 담보 평가 결정 테이블

게이트 계층

경고 3건

tbl_g3

결정 테이블

식별 가능성

테스트 비용

결정 트리

결정 다이어그램

역제 소개

역제 항목 17

인용 작업 목록

#	property_value	ltv_ratio	property_type	location_score	결과
1	Apartment	<=70%	High	Prime	승인
2	Officetel	<=75%	High	Good	승인
3	*	<=80%	Medium	Acceptable	조건부
4	*	>80%	*	*	거절
5	*	*	Low	*	보류
6	*	*	*	Poor	거절

● 보증 구조 결정 테이블

게이트 계층

경고 3건

tbl_guarantee

결정 테이블

식별 가능성

테스트 비용

결정 트리

결정 다이어그램

역제 소개

역제 항목 18

인용 작업 목록

#	property_value	ltv_ratio	property_type	location_score	결과
1	<=70%	Excellent	Apartment	Construction	HUG 표준
2	<=80%	Good	Officetel	Construction	HUG 표준
3	>80%	Fair	*	Construction	HUG 강화
4	*	Poor	*	*	추가 담보

워크플로우 설계는 BPMN 표준(업무 프로세스를 그리는 국제 표준 표기법) 기반의 태스크 플로우 편집기입니다. 업무의 시작과 끝, 작업과 하위 프로세스를 배치하고 각 태스크에 담당 비즈니스 컴포넌트를

지정합니다.

판단이 필요한 태스크는 **업무 결정 설계**로 연결됩니다. 이곳은 DMN 표준(의사결정 모델링의 국제 표준)의 의사결정 테이블 편집기로, "어떤 조건이면 어떤 결론"이라는 업무 규칙이 담당자의 머릿속이 아니라 검토 가능한 테이블로 문서화됩니다. 그 밖에 비즈니스 객체모델 뷰어와 보고서, 상품 설계, 전략 목표 설계, 비즈니스 컨텍스트 다이어그램 등이 엔티티 유형에 따라 동적 메뉴로 제공됩니다.

이 다이어그램들은 그림으로 끝나지 않습니다. BPMN 편집기에는 '**Poiesis에 제출**' 버튼이 있어 완성된 태스크 플로우를 만드는 날개로 넘기고, 설정에는 Poiesis URL과 Praxis URL이 나란히 있습니다. 우리 팀이 여기서 그리는 프로세스와 결정 테이블은 참고 자료가 아니라 **에이전트가 실제로 따르게 될 규칙**입니다. 그만큼 정확하게 그려야 하고, 그만큼 그럴 가치가 있습니다.

2.7 보강과 고객 가치 실현: 온톨로지에 살을 붙이는 일

2.7.1 보강: 에이전트라는 새 직원의 직무기술서

트리가 뼈대라면, 실무 시간이 가장 많이 들어가는 것은 뼈대에 살을 붙이는 **보강(Enrichment)**입니다. 엔티티를 우클릭하면 유형에 맞는 보강 편집기가 열립니다. 업무 대상 보강은 객체의 속성과 의미를, 도메인 지식 보강은 그 영역의 개념·절차·휴리스틱·인과 관계·규칙·규제를, 업무 결정 보강은 결정의 조건과 파라미터를 채웁니다.

그리고 역할 보강, 스킬 보강, 도구 보강, 거버넌스 정책 보강 — 이 넷을 나란히 읽어 보면 **사람 조직을 설계할 때 묻는 질문과 정확히 같다**는 것을 눈치채실 것입니다. 어떤 역할이 필요한가, 그 역할에는 어떤 스킬이 필요한가, 어떤 도구를 부여 줄 것인가, 어떤 정책 아래 일하게 할 것인가. 보강 작업은 곧 AI 에이전트라는 새 '직원'의 직무기술서를 쓰는 일입니다.

다만 결정적인 차이가 하나 있습니다. 그 직무기술서는 자연어 문서가 아니라 **온톨로지에 결합된 구조화 데이터**라는 점입니다. 그래서 "이 정책을 바꾸면 어떤 에이전트가 영향받는가"를 시스템이 추적할 수 있습니다. 사람의 직무기술서는 서랍에서 잠들지만, 에이전트의 직무기술서는 살아서 연결되어 있습니다.

2.7.2 고객 가치 실현(CVR): 니즈에서 에이전트 포트폴리오까지

고객을 향한 보강이 **고객 가치 실현(CVR)** 편집기입니다. 하나의 고객 활동을 일곱 단계로 다룹니다.

먼저 고객 여정과 동인을 매핑하는 프레임이 있고, 과업과 성과를 중요도·만족도로 순위화하는 과업 정의가 이어집니다. 미충족 성과를 솔루션과 에이전트 개념으로 바꾸는 아이디어션, 가치 흐름을 점점 채널 스위치라인으로 구체화하는 설계, 기회를 포트폴리오로 수렴하는 AI 에이전트 기회 단계가 그 뒤를 따릅니다. 고객 요구와 설계 대응의 정합성은 QFD(품질의 집) 기법으로 검증되고, 마지막으로 정의 단계의 성과 대비 실현 여부를 확인하는 실현 및 측정 단계가 순환을 닫습니다. 니즈에서 트레이드오프, 세부

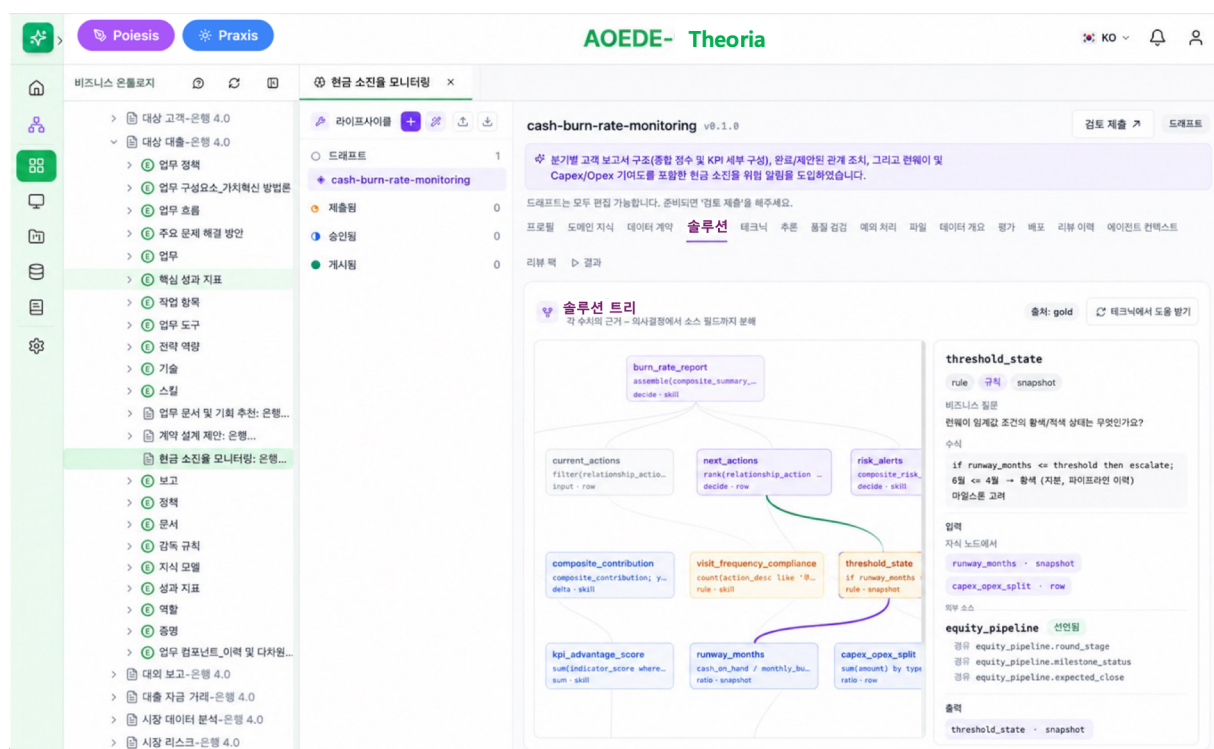
구현으로 내려가는 반복 퍼널이 함께 있어 한 시나리오를 라운드마다 충실도를 높여 갈 수 있습니다.

디자인 씽킹의 언어를 그대로 쓰되, 결과물이 워크숍 포스트잇으로 끝나지 않고 **온톨로지와 에이전트 포트폴리오로 직결되는** 도구라는 점이 차별점입니다.

2.7.3 능력캡슐 워크벤치: 지식이 최종 포장되는 곳

지식과 스킬이 최종 포장되는 곳이 **능력캡슐 워크벤치**입니다. 능력캡슐은 마치 감기약의 여러 약제가 하나의 캡슐에 담겨 효능을 발휘하듯이, 목표 능력의 효능을 전달하기 위한 업무 도메인 지식과 스킬 팩이 캡슐화된 단위입니다. 도메인 지식 캡슐과 능력캡슐은 버전 단위의 독립 문서로 저장되며, 모든 캡슐은 **초안 → 검토 제출 → 승인 → 게시**의 라이프사이클을 따릅니다. 제출된 캡슐은 반려·철회될 수 있고 승인된 캡슐도 게시 전에는 철회할 수 있지만, **한 번 게시된 버전은 최종본**입니다. 수정이 필요하다면 새 버전을 따라 합니다.

이 단순한 규칙의 무게를 짊어 두겠습니다. 에이전트가 참조하는 지식에는 언제나 "누가 언제 승인한 어느 버전인가"라는 꼬리표가 붙는다는 것 — 이것이 사후 감사와 사고 조사에서 조직을 지키는 가장 기본적인 방어선입니다.



2.8 두 개의 하네스: 시스템이 무너지지 않는 이유

제1장에서 독수리의 가슴을 X자로 가로지르는 넥서스 하네스를 이야기했습니다. 행동과 통제라는 두 개의 교차된 띠, 그리고 그 긴장을 붙드는 거버넌스라는 버클. 테오리아와 AOEDE 온톨로지 서버의 코드를 들여다보면, 그 은유가 실제로 **두 종류의 하네스**로 구현되어 있음을 발견하게 됩니다. 하나는 온톨

로지 모델 자체를 다스리는 **모델 하네스**이고, 다른 하나는 AI의 실행을 다스리는 **실행 하네스**입니다. 이 절이 제2장의 심장입니다.

2.8.1 온톨로지 모델 하네스: MVO 를 난장판에서 구하는 규율

MVO 접근의 매력은 분명합니다. 전체를 다 설계하지 않고, 지금 필요한 스킬 하나에 충분한 최소 온톨로지로 시작한다는 것. 그러나 이 접근에는 잘 알려진 그림자가 있습니다. **상향식으로 조각조각 자라는 모델은, 다스리는 손이 없으면 즉시 난장판이 됩니다.** 팀 A가 만든 '고객'과 팀 B가 만든 '고객'이 미묘하게 다른 뜻이 되고, 어제의 엔티티 이름이 오늘 바뀌어 있으며, 순환 참조와 모순된 분류가 아무도 모르게 쌓입니다. 사전 설계의 경직성을 피하려다 무정부 상태에 도착하는 것 — 이것이 상향식 접근이 실패하는 전형적 경로입니다.

온톨로지 모델 하네스는 이 경로를 막기 위해 네 겹으로 조여져 있습니다.

첫 번째 겹은 메타 온톨로지라는 문법입니다. 어떤 팀이 어떤 순서로 무엇을 만들든, 만들어지는 모든 것은 `#`로 시작하는 메타 객체들이 선언한 형식 — 업무대상은 엔티티를 가지고, 엔티티는 속성을 가진다는 문법 — 안에서만 존재할 수 있습니다. 상향식이 되 무형식이 아닌 것입니다. 서버는 저장소를 열 때 이 문법을 메모리에 올리고, 들어오는 데이터를 그 문법에 대조하여 검증합니다.

두 번째 겹은 개념 추론기입니다. 앞서 본 대로, 개념 모델의 모든 변경은 순환 검사, 배타 충돌 검사, 충족 불가능 클래스 검출을 통과합니다. 팀들이 각자 개념을 추가하더라도 논리적 모순은 저장 시점에 드러나며, 시스템이 추론해 낸 암묵적 상하위 관계는 제안으로 표시되어 사람이 확정합니다.

세 번째 겹은 경계와 이력입니다. 개념 모델과 온톨로지 데이터는 로그인 세션이 가리키는 저장소에만 귀속되어, 다른 저장소를 건드릴 수 있는 틈이 구조적으로 닫혀 있습니다. 편집 잠금은 두 사람이 같은 대상을 동시에 고치는 사고를 막고, 리비전 이력과 버전 관리는 모든 변경을 추적 가능하게 하며, 휴지통은 삭제조차 복구 가능한 행위로 만듭니다. 캡슐 라이프사이클의 승인 관문은 이 이력 위에 사람의 판단을 엮습니다.

네 번째 겹이 가장 흥미롭습니다. 시맨틱 팩토리의 모순 감지입니다. 앞의 세 겹이 모델 내부의 정합성을 지킨다면, 이 겹은 모델과 현실 문서 사이의 정합성을 감시합니다. MVO로 자라는 온톨로지가 "접지기 시작할 때" — 이탈 스킬의 '고객'과 대출 스킬의 '차주'가 만나는 그 순간 — 조율의 근거가 되는 것이 바로 이 증거 층입니다.

이 네 겹을 한 문장으로 줄이면 이렇습니다. **모델 하네스가 있기에, 발톱 하나씩 자라는 온톨로지가 대성당이 아니라 지도가 되면서도, 지도가 낙서장이 되지 않습니다.** 모델 하네스 없는 상향식은 빠르게 시작해서 빠르게 수렁에 빠지는 접근이고, 모델 하네스 있는 상향식이 비로소 MVO라는 이름값을 합니다.

2.8.2 AI 에이전트 실행 하네스: 문제 생성기가 되지 않는 조건

두 번째 하네스는 방향이 반대입니다. 모델 하네스가 "지도를 그리는 손"을 다스린다면, 실행 하네스는 "지도 위를 달리는 AI"를 다스립니다. 그리고 이쪽의 실패 양상은 더 시끄럽습니다. 통제되지 않은 AI 실행 시스템은 조용히 썩는 것이 아니라, **적극적으로 문제를 생산합니다**. 존재하지 않는 테이블을 조회하는 질의, 그럴듯하게 지어낸 숫자, 요청을 슬그머니 좁혀 놓고 전체를 답한 척하는 응답, 한계를 모르는 탐색이 시스템 전체를 끌어내리는 부하. 목표 지향적으로 움직이는 에이전트일수록, 고삐가 없으면 그 목표를 향해 성실하게 사고를 칩니다.

AOEDE 온톨로지 서버의 자연어 질의 하네스는 이 문제에 대한 교과서적인 답을 담고 있습니다. 설계 문서가 "협상 불가능한 두 가지"라고 부르는 원칙에서 출발합니다.

첫째, **스키마 환각 금지**입니다. AI는 실재하지 않는 엔티티·속성·관계를 참조하는 질의를 결코 실행시킬 수 없습니다. 둘째, **읽기 전용과 유한성**입니다. 모든 질의는 읽기만 하며, 결과 개수와 탐색 깊이가 구조적으로 상한에 묶입니다.

그리고 결정적인 조건이 붙습니다. 이 두 원칙은 "AI가 착하게 행동하는 것"이나 "검토자가 부지런한 것"에 의존해서는 안 되고, **구조적으로 강제**되어야 한다는 것입니다. 선의와 근면은 좋은 것이지만, 아키텍처의 안전을 그 위에 세울 수는 없습니다.

그래서 아키텍처는 세 단(tier)으로 나뉘고, 그 사이에 단 하나의 신뢰 경계가 그어집니다.

A단 — 언어 이해(신뢰하지 않음). LLM이 자연어 질문을 받아 형식화된 질의 계획 — 해석·선택·탐색·집계·영향분석 같은 타입 있는 연산들의 그래프 — 으로 번역합니다. 핵심은 이 단의 출력을 아무도 믿지 않는다는 것입니다. LLM은 데이터베이스 질의어를 직접 쓸 수 없고, 오직 중간 표현만 낼 수 있으며, 그 중간 표현은 틀려도 됩니다. 틀린 계획은 다음 단에서 걸러지기 때문입니다.

B단 — 검증과 컴파일(신뢰 경계). 결정론적 코드가 A단의 계획을 받아, 계획의 모든 끝단 — 대상 엔티티, 속성, 관계, 지표 — 를 **살아 있는 스키마 카탈로그에 대조하여 결속(bind)**합니다. 결속되지 않는 것은 실행되지 않고 탈락하며, 살아남은 연산만이 컴파일러를 통과합니다. 컴파일러는 시스템에서 질의어를 만들어 낼 수 있는 유일한 컴포넌트이고, 모든 산출 질의는 읽기 전용에 개수·깊이 상한이 걸린 채로만 만들어집니다. 계획이 "9999단계 탐색"을 요구해도 설정된 최대치로 잘립니다.

그리고 — 관리자 관점에서 백미인데 — 이 단은 **이해 보고서(Comprehension Report)**를 함께 냅니다. 질문 중 무엇이 그대로 수행되었고, 무엇이 좁혀졌으며, 무엇이 왜 탈락했는지의 명세입니다. 시스템은 못 하는 것을 조용히 빼고 답하는 대신, **못 했다고 말합니다**.

C단 — 실행과 검증(결정론). 검증된 계획이 실행되고, 비평기(critic)가 결과를 검사합니다. 결과가 비어 있지는 않은가, 형태가 질문과 맞는가, 조건이 실제로 적용되었는가. 문제가 있으면 유한한 횟수 안에서 계획을 고쳐 재시도하고, 최종 결과는 관련도로 정렬되어 근거·출처·능력 보고서·실행 추적과 함께 반환됩니다.

이 구조가 사용자에게 어떻게 보이는지는 이미 경험하셨을 수 있습니다. 온톨로지 Q&A에서 '확정된 영

향'과 '영향 가능 — 검토 필요'가 구분되고, 추론 항목에 미확정 표시가 붙고, 일치하는 엔티티가 없을 때 "영향이 없음을 확인한 것과 다르다"고 명시되는 것 — 그 모든 화면 문구의 뒤에 이 삼단 하네스가 있습니다.

2.8.3 두 하네스는 한 벌이다

정리하겠습니다. **모델 하네스가 없으면, 상향식으로 짓는 순간 온톨로지는 난장판이 됩니다.** 조각들이 서로 맞물리지 않고, 이름이 충돌하고, 모순이 침묵 속에 쌓여, MVO의 '최소'는 '영성함'의 다른 이름이 됩니다. **실행 하네스가 없으면, 시스템은 문제 생성기가 됩니다.** 그럴듯한 오답이 확신에 찬 어조로 배달되고, 통제되지 않은 질의가 자원을 태우며, 에이전트는 목표를 향해 성실하게 잘못된 길을 갑니다.

그리고 둘은 서로를 필요로 합니다. 실행 하네스의 B단이 끝단을 결속하는 대상인 '살아 있는 스키마 카탈로그'는, 모델 하네스가 정합성을 지켜 준 온톨로지 그 자체입니다. 모델이 어지러우면 아무리 훌륭한 실행 하네스도 어지러운 지도 위를 정확하게 달릴 뿐입니다. 거꾸로, 실행 하네스의 능력 보고서와 시맨틱 팩토리의 모순 발견은 모델의 빈 곳을 비추는 조명이 되어, 모델 하네스가 다음에 다스릴 대상을 알려 줍니다. 독수리 문장(紋章)의 X자 — 서로를 팽팽히 당기는 두 띠 — 는 장식이 아니라 이 상호 의존의 초상이었던 셈입니다.

2.9 제너시스에서 디지털 트윈으로: 단계별 건설과 내비게이션 맵

이제 마지막 큰 그림입니다. 지금까지 본 모든 조각 — 제너시스 씨앗, 두 층의 온톨로지, 보강, 시맨틱 팩토리, 두 하네스 — 는 사실 하나의 목적지를 향해 정렬되어 있습니다. **우리 비즈니스의 디지털 트윈**입니다.

여기서 디지털 트윈이란 공장 설비의 3D 복제 같은 것이 아닙니다. 우리 비즈니스의 구조와 지식과 상태를 컴퓨터가 이해하는 형태로 비추어, 질문하면 근거를 갖고 답하고 변경하면 파장을 짚어 주는 **살아 있는 대응물**을 뜻합니다.

중요한 것은 이 트윈이 한 번의 대공사로 지어지지 않는다는 점입니다. **제너시스는 그 단계별 건설의 주춧돌이며, 온톨로지는 매 단계에서 길을 잃지 않게 해 주는 내비게이션 맵입니다.** 건설의 여섯 단계를 따라가 보겠습니다.

1단계 — 씨앗 심기. 제너시스 패키지로 저장소를 부트스트랩합니다. 이 순간 태어나는 것은 뼈대뿐입니다. 업무대상들과 그들의 비즈니스 객체모델, 그리고 메타 온톨로지라는 문법. 그러나 이 뼈대가 곧 지도의 골격입니다. 이후의 모든 작업이 "온톨로지 지도의 어디에" 놓이는지가 이 순간 정해집니다.

2단계 — 정의 평면 세우기. 각 업무대상의 비즈니스 객체모델을 다듬고, 개념 모델에서 용어의 상하위와 배타 관계를 확정합니다. 추론기가 모순을 걸러 주는 가운데, "우리 도메인에서 무엇이 존재하고 어떻게 연결되는가"라는 정의적 지식의 평면이 완성되어 갑니다. 이 평면이 디지털 트윈의 좌표계입니다.

3단계 — 살 붙이기. 보강 편집기들로 속성과 의미, 도메인 지식, 결정 규칙, 역할·스킬·도구·정책을 채우고, BPMN과 DMN으로 프로세스와 판단을 모델링합니다. 트윈이 구조만이 아니라 행동 방식을 갖기 시작합니다.

4단계 — 증거 연결하기. 시맨틱 팩토리에 실제 문서들을 흘려 넣습니다. 규정집과 계약서와 매뉴얼에서 추출된 사실과 관계가 출처를 매단 채 쌓이고, 모순 감지가 모델과 현실의 어긋남을 비춥니다. 트윈이 설계도의 복제가 아니라 **현실의 반영**이 되는 단계입니다.

5단계 — 그래프로 깨어나기. 디지털 트윈 RAG가 온톨로지와 보강 내용 전체로부터 지식 그래프를 빌드합니다. 이때 앞서 본 층위 규율이 그대로 지켜집니다. 정의적 질문은 비즈니스 객체모델에서 유래한 모델 평면에서, 인스턴스 질문은 데이터 행 평면에서 — 두 평면이 섞이지 않기에 답이 정확합니다. 그리고 이 그래프 위에서 실행 하네스가 자연어 질문을 안전하게 받아냅니다.

6단계 — 번식하기. 충분히 여문 저장소는 다시 제너시스 패키지로 내보내져, 다음 도메인·다음 법인·다음 실험의 씨앗이 됩니다. 캡슐 가져오기와 컬렉션의 Excel 입출력이 부분 단위의 이동을 거듭니다. 트윈 하나의 완성이 다음 트윈의 출발점이 되는 것입니다.

이 여섯 단계가 MVO의 리듬과 정확히 포개진다는 점을 눈여겨보시기 바랍니다. 어느 단계도 "전체를 끝내야" 다음으로 갈 수 있는 관문이 아닙니다. 하나의 업무대상, 하나의 능력캡슐 스킬 범위 안에서 1단계부터 5단계까지를 작게 완주하고, 그 완주를 옆 영역으로 넓혀 갑니다. 그리고 매 단계에서 **온톨로지 트리가 내비게이션 맵의 역할**을 합니다. 지금 우리가 지도의 어느 구역을 짓고 있는지, 어느 구역이 정의만 있고 증거가 없는지, 어느 구역이 문서와 어긋나 있는지 — 트리와 상태 값과 모순 보고가 그것을 항상 보여 줍니다. 안개 속에서 짓는 것이 아니라, 지도를 보며 한 구역씩 등을 켜 가는 건설입니다.

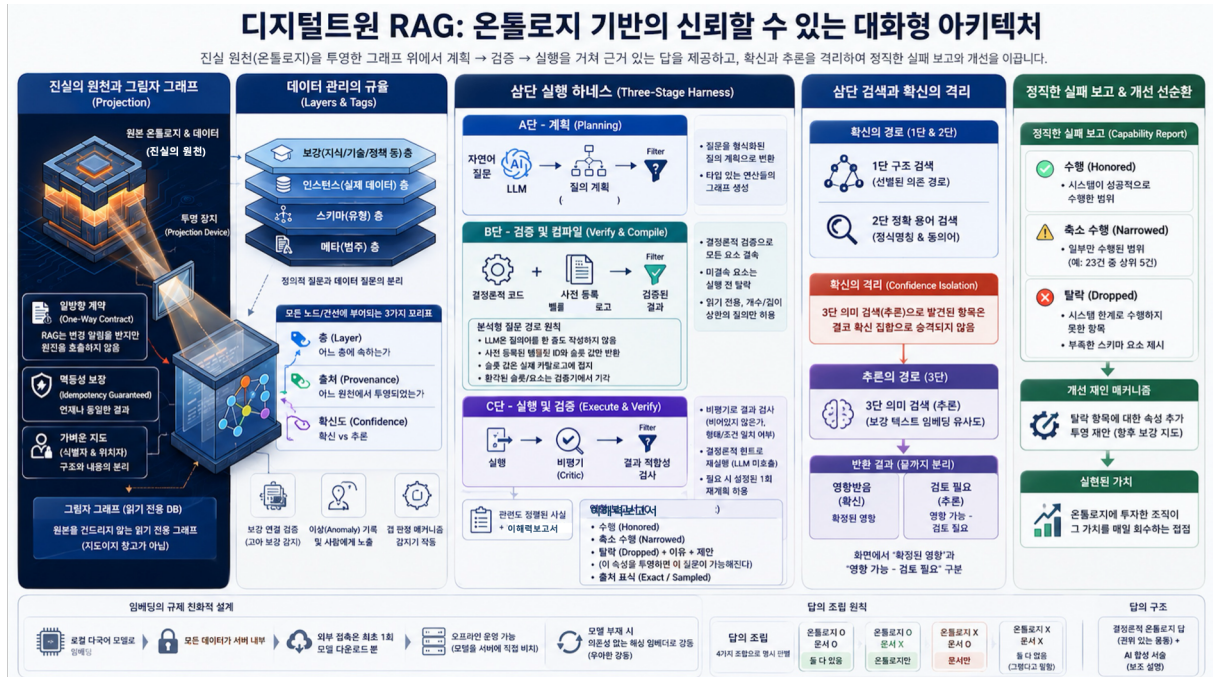
이 리듬이 조직에 주는 심리적 효과도 가볍지 않습니다. 수년짜리 대공사는 완공 전까지 아무도 성과를 손에 쥐지 못하므로, 중간에 예산이 흔들리고 후원자가 바뀌면 프로젝트 전체가 표류합니다. 반면 한 구역씩 등을 켜는 건설에서는 매 분기 실제로 켜진 등 — 배포된 스킬, 답할 수 있게 된 질문, 줄어든 놓침 — 을 보여 줄 수 있습니다. 온톨로지 프로젝트가 살아남는 조건은 완벽한 설계가 아니라 꾸준히 보여 줄 수 있는 불빛이라는 것을, 이 여섯 단계의 구조는 처음부터 계산에 넣고 있습니다.

2.10 온톨로지에게 직접 묻다: 트윈과의 대화

건설의 보상은 대화입니다. 온톨로지 Q&A 탭을 열면, 팀이 쌓아 온 트윈 전체를 상대로 자연어 질문을 던질 수 있습니다. "개인' 업무 대상의 세부 정보와 능력을 설명해 달라", "여기서 LTV는 무슨 뜻인가", "이 엔티티를 삭제하면 무엇이 영향을 받는가", 심지어 "어느 프로세스가 가장 복잡한가? 복잡도는 태스크 수로 계산하라"처럼 사용자가 지표를 즉석에서 정의하는 질문까지 받아냅니다.

답변의 성격은 이미 앞 절에서 본 실행 하네스가 결정합니다. 설명형 질문에는 서버가 온톨로지를 정확히 조회해 결정론적으로 구성한 블록 — 무엇이 일치했고, 어디에 속하며, 상하위 의존이 무엇인지 — 이 권위 있는 답으로 먼저 놓이고, AI가 합성한 서술은 그 아래에 놓입니다. 근거 노드들은 그래프로 시각화

되어 클릭하면 속성이 열립니다. 영향 분석에서는 확정과 추정이 구분되고, 보강 연결의 이상이 감지되면 해당 항목이 확정 결과에서 제외되었음이 명시됩니다.



관리 관점에서 이 기능은 두 가지를 바꿉니다. 첫째, "이거 바꾸면 뭐가 깨지나?"라는 변경 영향 질문이 담당자 수배와 회의 소집 없이 몇 초 만에 1차 답을 얻습니다. 둘째, 팀이 보강과 문서 연결을 성실히 할 수록 답의 품질이 눈에 띄게 올라가므로, 온톨로지 관리가 '나중을 위한 문서화'가 아니라 '지금 당장 되 돌아오는 투자'가 됩니다.

2.11 하루의 흐름으로 본 테오리아

지금까지의 이야기를 한 팀의 하루로 옮겨 보면, 이 작업대의 감촉이 좀 더 생생해집니다.

아침, 온톨로지 담당자가 테오리아에 로그인합니다. 화면 하단의 상태 바가 어느 서버의 어느 저장소에 연결되어 있는지를 확인시켜 주고, 왼쪽의 온톨로지 트리에는 어제까지 다듬어 온 회사의 구조가 그대로 펼쳐져 있습니다. 밤사이 시맨틱 팩토리가 새로 첨부된 여신 규정 개정본을 읽어 두었고, 모순 감지가 발견 사항 두 건을 올려 두었습니다. 하나는 개정 규정에는 등장하는데 온톨로지에는 아직 없는 개념이고, 다른 하나는 문서가 말하는 관계와 모델이 선언한 관계가 어긋나는 지점입니다. 담당자는 이 두 건을 오늘 오후 온톨로지 협의체의 안건으로 옮겨 둡니다. 모델과 현실의 어긋남이 방지된 부채가 아니라 아침 회의의 안건이 되는 것 — 이것이 이 작업대의 일상입니다.

오전에는 업무 설계자가 '분쟁처리' 업무대상을 더블클릭해 작업 공간을 엽니다. 시나리오 컴포저에서 새 업무 시나리오의 골격을 노드로 짜고, 각 노드에 맥락과 질문을 적은 뒤 마이닝을 누릅니다. AI가 초안을 채우지만, 그 재료는 백지가 아니라 다른 노드의 내용과 첨부된 규정 문서, RAG로 검색된 사내 지식입니다. 생성된 내용은 완전성과 명확성, 일관성, 실행 가능성의 네 축으로 품질 분석을 받고, 노드에

는 '검토 중' 상태가 붙습니다. 옆자리의 동료는 같은 시각 BPMN 편집기에서 태스크 플로를 다듬는 중이고, 판단이 필요한 태스크에는 DMN 의사결정 테이블이 연결되어 갑니다.

오후, 도메인 능력 캡슐 하나가 검토 제출됩니다. 승인자는 캡슐의 내용이 골드 스탠더드에 비추어 충분한지, 출처가 제대로 달려 있는지, 실행결과가 약속한 능력을 제공하는지를 확인한 뒤 승인 도장을 누릅니다. 이 버전은 이제 게시되면 최종본이 되고, 이후 어느 에이전트가 이 지식을 참조하든 "누가 언제 승인한 어느 버전인가"라는 꼬리표가 따라붙습니다. 퇴근 무렵, 팀장은 온톨로지 Q&A 탭을 열어 "이 엔티티를 삭제하면 무엇이 영향을 받는가"를 물어봅니다. 몇 초 뒤 확정된 영향과 검토가 필요한 추정이 구분되어 돌아오고, 능력 보고서는 어느 부분을 좁혀서 답했는지를 정직하게 밝힙니다. 팀장은 그 '좁힌 부분'을 다음 주에 보강할 구역으로 메모합니다.

이 하루에 극적인 장면은 없습니다. 그러나 눈여겨보면, 모든 손길에 같은 문법이 흐르고 있습니다. 발견은 시스템이 하고 조율은 사람이 하며, 생성은 AI가 하고 확정은 검토가 하며, 지식은 쌓이되 반드시 출처와 승인의 꼬리표를 달고 쌓입니다. 오늘도 새로운 능력이 회사 운영에 더해졌습니다. 디지털 트윈은 이런 하루하루가 포개져 자라는 것이지, 어느 날 완공식을 여는 건물이 아닙니다.

2.12 맺으며: 머리를 맡은 관리자에게

테오리아와 그 서버를 한 문장으로 요약하면 이렇습니다. **개념 모델이라는 문법 위에, 제너시스라는 씨앗으로 회사의 비즈니스 온톨로지를 심고, 시맨틱 팩토리로 현실의 증거를 잇대며, 두 개의 하네스로 모델과 실행을 함께 다스려, 단계별로 디지털 트윈을 지어 올리는 작업대.** 다국어 지원, 편집 잠금과 버전 이력, 휴지통과 캡슐 입출력 같은 실무 장치들이 그 작업을 받쳐 줍니다.

이 작업대를 맡게 될 관리자에게 몇 가지 당부를 남기고 싶습니다. 가장 먼저 정해 두어야 할 것은 **제너시스와 개념 모델의 관리 권한**입니다. 온톨로지 제너시스는 모든 작업의 좌표계이고 개념 모델은 그 좌표계의 문법이어서, 문법이 어지러우면 그 위의 모든 문장이 함께 어지러워집니다. 누가 트리의 구조를 바꿀 수 있고 누가 개념의 정의를 확정하는지는, 도입 첫 주에 이름으로 정해져 있어야 하는 문제입니다. 그 다음으로는 **브레인 설정과 보강 관행을 팀의 표준으로** 세우는 일입니다. 같은 도구라도 컨텍스트를 성실히 채우는 팀과 그렇지 않은 팀의 산출물 품질은 크게 벌어지며, 설정 전파 기능은 바로 그 표준을 계층 전체에 내려보내라고 마련된 것입니다.

캡슐 라이프사이클의 승인자 역할도 가볍게 볼 일이 아닙니다. 초안에서 게시로 넘어가는 그 관문이 우리 팀이 만들어 낼 에이전트의 신뢰 수준을 결정하므로, **승인은 결재가 아니라 검토**여야 합니다. 시맨틱 팩토리가 올리는 모순 보고 역시 그렇습니다. 모델과 문서의 어긋남은 방치하면 부채가 되지만, 정기 안건으로 다루면 온톨로지가 자라는 가장 값싼 원료가 됩니다. 하네스는 발견까지만 해 주고, 조율은 언제나 사람의 몫이기 때문입니다.

그리고 마지막으로, **이해 보고서를 읽는 문화**를 만들어 주시기 바랍니다. 시스템이 "이 부분은 좁혀서 답했고, 이 부분은 답하지 못했다"고 정직하게 말할 때, 그 고백을 흘려듣지 않는 팀만이 정직한 시스템의 값어치를 온전히 누립니다. 정직한 실패 보고는 벌점의 재료가 아니라, 다음에 지어야 할 구역을 알려

주는 지도입니다.

독수리의 머리는 이제 추상적인 개념이 아니라 매일 아침 열어 일하는 화면이고, 하네스는 은유가 아니라 코드로 조여진 실물이며, 디지털 트윈은 어느 날 완성되는 프로젝트가 아니라 한 구역씩 등을 켜 가는 지도입니다. 볼 수 있을 만큼만 정의하고, 정의한 만큼 보강하고, 보강한 것을 두 겹의 하네스로 다스려서 넘기는 것 — 테오리아에서의 하루하루가 곧 그 지도에 불을 켜는 일입니다.

제2장 핵심 요약: 테오리아는 개념 모델(문법), 제너시스(씨앗이자 운반 형식), 시맨틱 팩토리(증거 보관소)의 세 기둥 위에 세워진 온톨로지 작업대다. 모델 하네스(메타 문법, 추론기, 경계와 이력, 모순 감지)가 상향식 성장을 난장판에서 지키고, 실행 하네스(A단 언어 이해, B단 검증·컴파일, C단 실행·검증)가 AI의 답변을 구조적으로 강제한다. 이 위에서 디지털 트윈이 여섯 단계로, MVO의 리듬에 맞춰 자라난다.

제 3 장. 포이에시스: 만드는 날개

이 장에서 다루는 것: 포이에시스가 만드는 것의 정체(에이전트가 아니라 에이전트 하네스), 하네스의 세 가지 크기(활동·태스크·스킬)와 도입의 사다리, 그리고 포이에시스가 조직에 부여하는 여덟 가지 능력. 예시 도메인은 주택담보대출 심사를 사용합니다.

3.1 들어가며: 포이에시스가 만드는 것의 정체

제2장은 하나의 버튼으로 끝났습니다. 테오리아에서 완성된 업무 흐름을 만드는 날개로 넘기는 'Poiesis 에 제출' 버튼입니다. 이번 장은 그 버튼이 눌린 다음의 이야기입니다.

그런데 이야기를 시작하기 전에, **AOEDE-포이에시스**가 만드는 것의 정체를 정확히 해 둘 필요가 있습니다. 포이에시스는 흔히 말하는 'AI 에이전트'를 만드는 곳이 아닙니다. 더 정확히 말하면, **AI 에이전트 하네스를 조립하는 곳**입니다.

이 구분은 말장난이 아닙니다. 별거벗은 AI — 똑똑하지만 고삐 없는 지능 — 는 시중 어디서나 구할 수 있습니다. 기업이 만들어야 하는 것은 그 지능에 채울 마구(馬具)입니다. 무엇을 알아야 하고, 무엇을 할 수 있고, 어디까지 가도 되며, 어느 지점에서 반드시 멈추거나 사람에게 넘겨야 하는지가 짜여 들어간 구조물. 제2장에서 실행 하네스를 두고 "고삐 없는 목표 지향 에이전트는 목표를 향해 성실하게 사고를 친다"고 했던 바로 그 문제의식이, 포이에시스에서는 만들기의 대상 자체가 됩니다. **포이에시스에서 '만든다'는 것은 곧 '하네스를 짠다'는 것입니다.**

포이에시스가 겨냥하는 사용자가 개발자가 아니라 **비즈니스 분석가와 업무 설계자**라는 점도 이 정체와 맞닿아 있습니다. 하네스의 본질 — 어디까지 맡기고 어디서 멈추게 할 것인가 — 은 기술 판단이 아니라 업무 판단이기 때문입니다. 에이전트의 내부 명세는 전용 언어로 기술되지만, 포이에시스의 존재 이유는 현업 전문가가 그 코드를 직접 쓰지 않고도 자연어와 시각적 설계 도구로 완결된 하네스를 조립할 수 있게 하는 데 있습니다. **코드는 결과물이 아닌 입장권이 아닙니다.**

이 장은 화면 안내서가 아닙니다. 먼저 포이에시스가 짓는 하네스의 세 가지 크기 — 활동, 태스크, 능력 — 를 이해하고, 그 위에서 포이에시스가 조직에 부여하는 여덟 가지 능력을 하나씩 이야기하겠습니다. 예시는 시스템에 실제 탑재된 **주택담보대출 심사** 도메인을 사용합니다.

한편, 조립된 하네스의 내부 규격과 이를 실행하는 엔진의 이야기 — 넥서스와 미메시스 — 는 이 장에서 다루지 않습니다. 그것은 배포된 에이전트가 실제로 일하는 무대인 프락시스와 함께 제4장에서 온전히 펼쳐는 것이 옳기 때문입니다.

3.2 세 단계의 하네스: 활동(워크플로우), 태스크, 능력

포이에시스의 작업 대상 목록을 열면 구현 대상이 처음부터 두 층으로 나뉘어 도착해 있는 것을 보게 됩니다.

니다. **활동 레벨**과 **태스크 레벨**입니다. 그리고 그 아래에서 능력의 계층이 설계됩니다. 이 층 구분은 정리용 폴더가 아니라, AOEDE가 자동화를 바라보는 관점 그 자체입니다. 하네스에는 세 가지 크기가 있고, 크기마다 목적과 사람의 자리가 다릅니다.

3.2.1 활동(워크플로우) 레벨 하네스: 흐름 전체를 감싸는 가장 큰 마구

활동 레벨 하네스는 가장 큰 마구입니다. 하나의 업무 활동 — 예컨대 주택담보대출 심사라는 처음부터 끝까지의 흐름 전체 — 를 감싸며, 목표는 **자율 운영(autonomous operation)**입니다. 신청 접수에서 최종 의사결정까지 여러 태스크와 여러 관문을 통과하는 엔드투엔드 워크플로우가 에이전트들의 협업으로 굴러가고, 사람은 흐름 안의 정해진 개입 지점과 예외 에스컬레이션에서만 등장합니다.

이 크기의 하네스가 감당해야 하는 것은 개별 판단의 품질을 넘어 **흐름 전체의 건전성**입니다. 관문 사이의 정합성, 역할 간 인계, 전체 기초의 조절까지가 하네스의 일부입니다.

3.2.2 태스크 레벨 하네스: 한 역할의 대리인

태스크 레벨 하네스는 중간 크기입니다. 활동 전체가 아니라 **특정 역할이 수행하는 하나의 태스크** — 예컨대 심사역의 담보 평가, 준법 담당의 자금세탁 점검 — 를 감쌉니다. 이 크기에서 에이전트는 흐름의 주인이 아니라 **한 역할의 대리인 또는 동료**입니다. 태스크의 입력과 출력, 그 태스크에 필요한 스킬과 숙련도, 태스크가 지켜야 할 규칙과 권한이 하네스의 내용이 되고, 태스크의 앞뒤는 여전히 사람 또는 다른 하네스가 있습니다.

3.2.3 능력 레벨 하네스: 가장 작지만 가장 정교한 마구

능력 레벨 하네스는 가장 작지만, 어떤 의미에서 가장 정교한 마구입니다. 목표가 자동화가 아니라 **사람 일꾼의 능력 증강(augmentation)**이기 때문입니다. 담보 가치 산정, 소득 서류 검증, 규정 조회 같은 하나의 능력 단위에서, 에이전트는 사람을 대신하는 것이 아니라 사람의 손에 쥐어지는 정밀한 도구가 됩니다.

그리고 바로 그렇기 때문에 이 작은 하네스에는 두 가지가 응축되어 있어야 합니다. 첫째, 그 능력에 관한 **포괄적인 지식** — 관련 도메인 지식, 스킬, 판단의 재료, 참조해야 할 규정과 출처. 둘째, **통제 지점(control points)** — 어떤 값은 반드시 검증을 거쳐야 하고, 어떤 결론은 절대 내릴 수 없으며, 어떤 경우에는 반드시 사람의 확인을 받아야 한다는 지점들. 요컨대 능력 하네스는 **"일이 올바른 방식으로만 이루어지게 하는"** 지식과 **고삐의 최소 완결 단위**입니다.

3.2.4 포개지는 구조, 그리고 도입의 사다리

세 크기의 관계도 중요합니다. 이들은 서로 다른 세 제품이 아니라 **포개지는 구조**입니다. 활동 하네스는 태스크 하네스들의 짜임이고, 태스크 하네스는 능력 하네스들의 짜임입니다. 제1장에서 능력캡슐을 두고 "각 캡슐은 축소된 독수리 한 마리"라고 했던 문장이 여기서 구조적 의미를 얻습니다. 가장 작은 능력 하네스에도 지식과 통제가 완결적으로 들어 있기에, 그것들을 쌓아 올린 태스크와 활동도 무너지지 않

는 것입니다.

관리자에게 이 세 단계는 곧 **도입의 사다리**이기도 합니다. 조직은 대개 능력 하네스에서 출발하는 것이 안전합니다. 사람이 여전히 운전대를 쥔 채, 증강된 능력의 품질을 검증하는 단계입니다. 검증된 스킬들이 모이면 한 역할의 태스크를 하네스에 맡길 수 있고, 태스크들이 여물면 비로소 활동 전체의 자율 운영을 논할 자격이 생깁니다. 제1장의 MVO — 발톱 하나, 능력 하나씩 자라는 온톨로지 — 와 정확히 같은 리듬입니다. **자율성은 선언하는 것이 아니라, 작은 하네스에서부터 적립하는 것입니다.**

사람을 채용하는 일에 빗대면 이 사다리의 감각이 더 또렷해집니다. 신입 직원에게 첫날부터 부서 전체의 운영을 맡기는 조직은 없습니다. 처음에는 선배 곁에서 자료 조사와 초안 작성 같은 한 가지 일(능력)을 돕게 하고, 솜씨가 검증되면 하나의 업무(태스크)를 온전히 맡기며, 여러 업무에서 신뢰가 쌓인 뒤에야 한 흐름 전체(활동)를 책임지게 합니다. AOEDE의 세 하네스는 이 상식적인 승진 체계를 에이전트에게 그대로 적용한 것이고, 뒤에서 보게 될 인증과 숙련도의 장치들은 그 승진의 인사 기록에 해당합니다.

이제 이 세 단계의 렌즈를 끼고, 포이에시스가 조직에 부여하는 여덟 가지 능력을 걸어 보겠습니다.

3.3 첫 번째 능력: 하네스 구현의 파이프라인을 한눈에 진다

포이에시스가 관리자에게 주는 첫 번째 선물은 화려한 기능이 아니라 **가시성**입니다. 테오리아에서 넘어온 구현 대상들은 활동 레벨과 태스크 레벨로 층이 나뉜 채 도착하고, 도착하는 순간부터 하나의 생애 주기를 부여받습니다. **접수됨 → 작성 중 → 검증됨 → 배포됨**, 그리고 어느 단계에서든 갈 수 있는 **반려됨**.

이 다섯 상태는 단순한 꼬리표가 아닙니다. 포이에시스의 작업 흐름 자체가 **작성 → 검증 → 버전 → 배포**라는 네 단계의 순차 공정으로 짜여 있고, 각 공정은 다음 공정의 입장 조건이 됩니다. 만들지 않은 하네스는 검증할 수 없고, 검증하지 않은 하네스는 배포할 수 없습니다. 상태는 그 공정 위에서 각 하네스가 지금 어디에 있는지를 가리키는 좌표입니다.

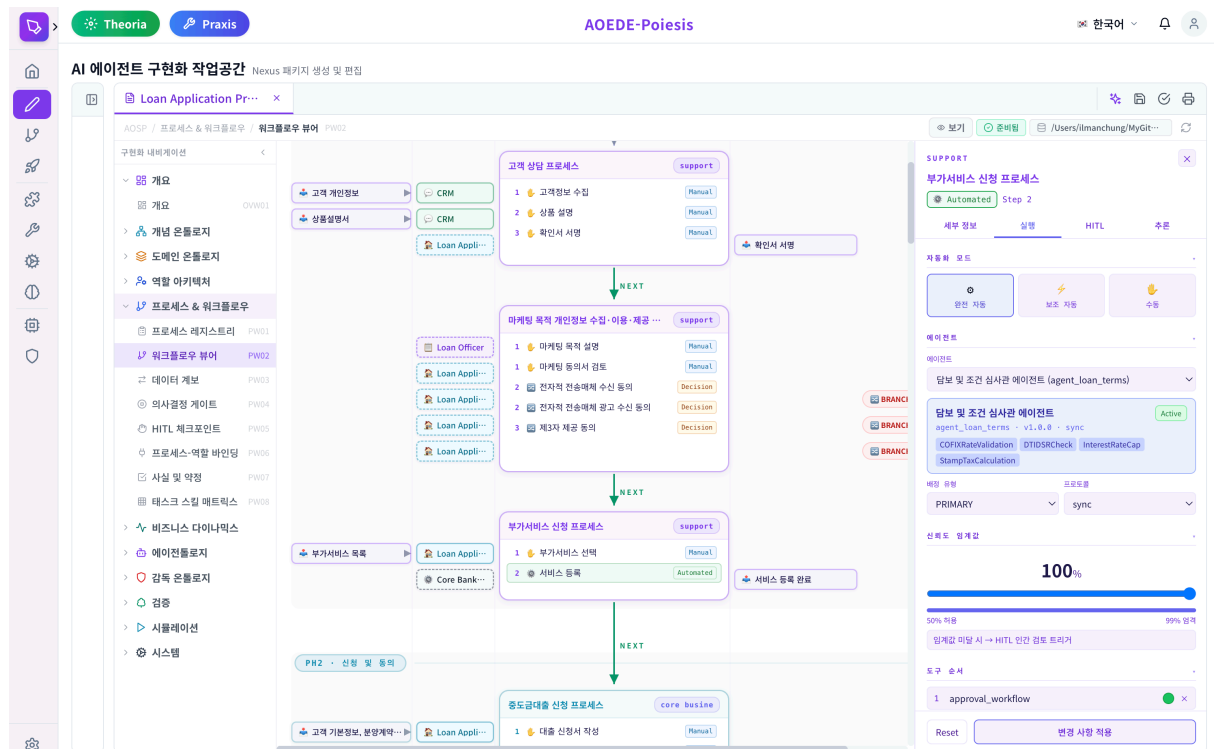
관리자 입장에서 이것이 바꾸는 것은 진척 회의의 언어입니다. "그 건 어떻게 되어 가나?"라는 질문에 담당자의 기억으로 답하는 대신, 파이프라인 위의 좌표로 답하게 됩니다. 어떤 활동 하네스가 몇 주째 '작성 중'에 머물러 있는지, 태스크 하네스들 가운데 '검증됨'과 '배포됨' 사이에 병목이 있는지가 **조회의 대상이 되지, 수소문의 대상이 되지 않습니다.**

작은 규율 하나가 이 능력의 성격을 잘 보여 줍니다. 근거가 될 도메인 데이터가 연결되지 않은 대상은 아예 작업을 시작할 수 없도록 잠겨 있습니다. 데이터 없이 하네스부터 짜기 시작하는 일 — 모든 구현 프로젝트가 한 번쯤 겪는 유혹 — 이 이 파이프라인에서는 구조적으로 일어나지 않습니다.

3.4 두 번째 능력: 시나리오에서 출발하는 하네스 설계

하네스 조립의 첫 삼은 코드가 아니라 **이야기**입니다. 담당자는 이 활동 또는 태스크가 하는 일을 자연어 시나리오로 서술하고, 이 업무가 달성해야 할 **목표**를 함께 정의합니다. 그리고 AI의 지원을 받아 그 시나

리오를 구조화된 설계 초안으로 발전시킵니다.



여기서 중요한 것은 AI 지원의 방식입니다. 포이에시스의 AI 생성은 자유 작문이 아니라 **규격이 주어진 작문**입니다. 하네스에 어떤 구성 요소가 존재할 수 있고 서로 어떻게 결합될 수 있는지가 사전에 정의된 규격으로 시스템에 등록되어 있고, 생성은 그 테두리 안에서만 이루어집니다. 제2장의 원칙 — "AI의 출력은 검증되기 전까지 신뢰되지 않는다" — 이 설계 단계에서는 "AI의 출력은 규격 안에서만 태어난다"는 형태로 관철되는 것입니다. **하네스를 짜는 조수에게도 하네스가 채워져 있는 셈입니다.**

AI 활용의 조건 자체도 팀 자산으로 관리됩니다. 어떤 모델을 어떤 온도와 페르소나로 쓸지, 번역은 어떻게 불일치가 설정으로 표준화되므로, 누가 작성하든 같은 조건에서 같은 품질의 초안이 나옵니다. 인터페이스는 한국어·영어·중국어·일본어를 오가며, 다국적 팀이 한 도메인을 함께 설계하는 상황을 처음부터 가정하고 있습니다.

관리자의 눈으로 이 능력을 요약하면 이렇습니다. **설계의 출발점이 현업의 언어가 되고, AI는 그 언어를 규격으로 번역하는 조수가 됩니다.** 현업이 요구사항을 적어 개발 조직에 던지고 몇 주 뒤 번역 오류를 발견하는 전통적 리듬이, 현업이 직접 하네스의 초안을 손에 쥐는 리듬으로 바뀌는 것입니다.

3.5 세 번째 능력: 하네스에 넣을 모든 지식에 근거의 신분증을 붙인다

능력 하네스의 정의를 다시 떠올려 보겠습니다 — "일이 올바른 방식으로만 이루어지게 하는 포괄적 지식과 통제 지점의 최소 완결 단위." 이 절은 그 앞부분, 지식의 이야기입니다. 하네스에 담기는 지식의 품질은 결국 "무엇에 근거했는가"로 결정되며, 포이에시스가 이 지점에서 제공하는 능력은 한 문장으로 요약됩니다. **에이전트가 참조할 세계의 사실 하나하나에 신분증이 붙어 있습니다.**

설계자는 작업 중에 언제든지 도메인의 전모를 여러 각도에서 확인할 수 있습니다. 어떤 업무 대상이 존재하고 무엇으로 같음을 판별하는지, 대상들이 어떤 관계로 얹혀 있는지, 어떤 사건이 어떤 상태 변화를 일으키는지. 그런데 포이에시스의 도메인 지식은 단순한 열람 자료를 넘어, 판본이 거듭될수록 세 방향으로 깊어져 왔습니다.

가장 먼저 깊어진 방향은 **규제와의 연결**입니다. 업무 규칙과 불변 조건 — 언제나 지켜져야 할 선언적 규칙들 — 에는 그 근거가 되는 규제 조항이 직접 연결되어 있고, 규정·조항·인용이 구조화된 그래프로 탐색됩니다. "이 규칙은 왜 있는가?"라는 질문이 담당자의 기억이 아니라 조항 링크로 답해집니다.

여기에 **권한과의 연결**이 더해집니다. 누가 어떤 행위를 어떤 범위에서 실행할 수 있으며 결재 체인은 무엇인지 — 위임의 규칙, 금액 한도, 유효 기간까지 — 가 매트릭스로 선언되어 있습니다. 하네스가 허용하는 행동의 경계가 관행이 아니라 명세인 것입니다.

그리고 마지막 방향이 **출처와의 연결**입니다. 각 데이터 값이 어디서 왔는지의 출처 귀속과, 그 값이 얼마나 확실한지의 불확실성 프로파일, 여러 출처가 충돌할 때의 조정 규칙까지가 값 단위로 기록됩니다.

이 세 겹이 합쳐져 만들어 내는 상태를 관리자의 언어로 옮기면 이렇습니다. **감사인의 어떤 질문에든 화면 하나로 답할 수 있는 하네스**. "그 숫자 어디서 났습니까?", "그 규칙의 법적 근거는 무엇입니까?", "그 행위를 그 에이전트가 할 권한이 있습니까?" — 규제 산업에서 에이전트를 운영하려는 조직이 반드시 통과해야 할 세 질문에 대한 답이, 설계 시점부터 하네스 안에 축적됩니다.

3.6 네 번째 능력: 능력에서 활동까지, 일하는 주체를 조직처럼 설계한다

하네스의 세 단계가 실제 설계물로 나타나는 곳이 바로 이 능력입니다. 에이전트를 만든다는 것은 결국 **새로운 일꾼을 조직에 들이는 일**이고, 포이에시스는 사람 조직을 설계할 때 쓰는 어휘를 그대로 이 세계에 적용합니다.

가장 아래에 **능력의 아키텍처**가 있습니다. 복합적인 능력이 원자적인 스킬로 분해되고, 말단의 원자 스킬이 실제 도구와 결합하는 계층 구조 — 능력 하네스들의 족보입니다. 각 스킬에는 그것을 뒷받침하는 도메인 지식, 수행의 기법, 판단의 골격이 되는 추론 템플릿이 연결됩니다. 앞 절에서 본 신분증 붙은 지식이 스킬 단위로 꾸러지고, 다음 절에서 볼 통제 지점이 여기에 결합하면, 사람 일꾼의 손에 쥐어질 증강의 단위 — 능력 하네스 — 가 완성됩니다.

그 위에 **역할과 태스크의 아키텍처**가 있습니다. 어떤 역할들이 존재하고, 역할 사이의 위계와 에스컬레이션 경로 — 문제가 커지면 누구에게 올라가는가 — 는 어떻게 되며, 각 역할은 어떤 태스크를 맡는가. 그리고 각 태스크가 어떤 스킬을 어느 숙련도로 요구하며 **자동화가 가능한 단계는 어디인지**를 밝히는 업무-스킬 대응표가 두 층을 잇습니다. 이 대응표가 바로 능력 하네스를 태스크 하네스로 승격시킬 수 있는지의 판정표입니다 — 태스크가 요구하는 스킬들이 모두 검증된 하네스로 존재하는가?

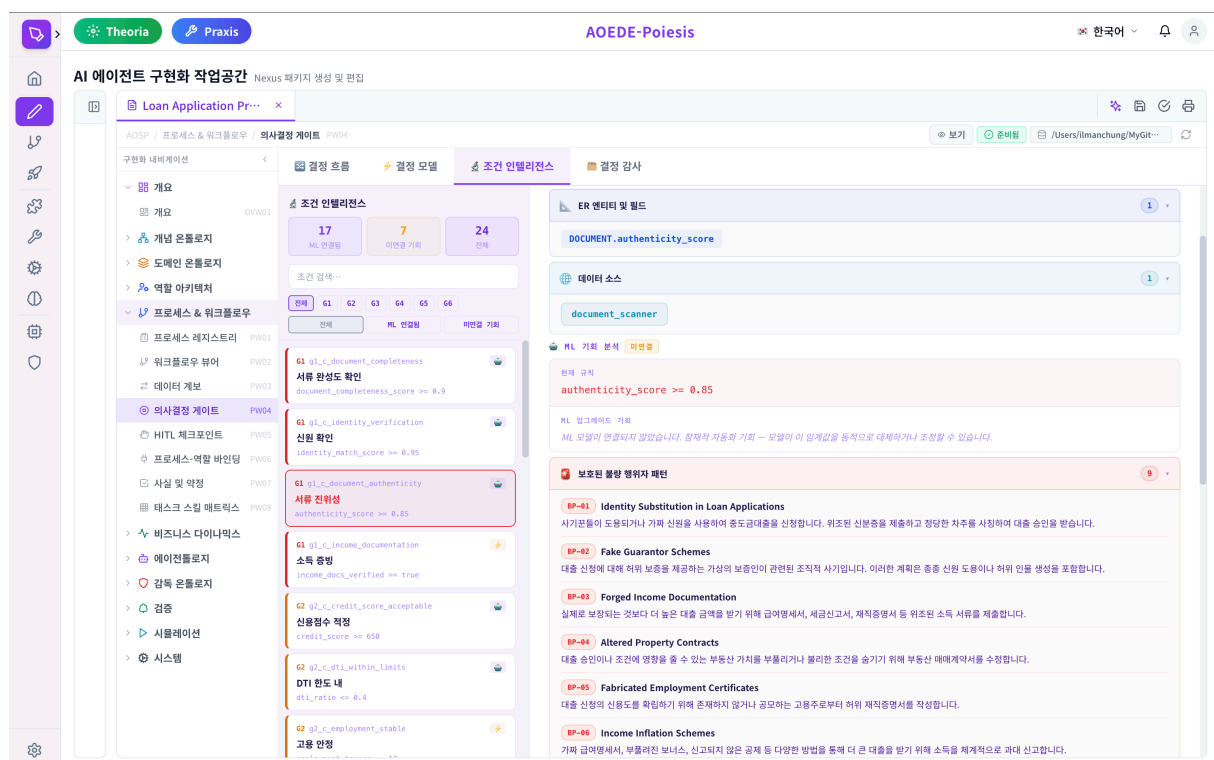
가장 위에 **활동의 아키텍처**가 있습니다. 태스크들이 프로세스로 이어지고, 관문들이 흐름을 통제하며,

결정적으로 **사람 개입 지점(HITL 체크포인트, Human-in-the-Loop)**이 일급 설계 요소로 선언됩니다. 활동 하네스의 자율성이란 사람이 사라진 상태가 아니라, **사람의 자리가 정확히 설계된 상태**입니다. 에이전트가 어디까지 가고 어디서 사람에게 넘기는지가 설계 산출물이지, 운영 중의 즉흥이 아닙니다.

흥미로운 원칙 하나가 이 관점의 깊이를 보여 줍니다. 에이전트의 모든 LLM 기반 작업조차 등록된 '인지 능력'이라는 형식을 통해 선언되어야 합니다. 이 시스템에서는 **AI의 지능조차 등록된 능력**이며, 등록되지 않은 능력은 하네스 안에 존재하지 않습니다.

3.7 다섯 번째 능력: 통제 지점을 수학으로 버린다

능력 하네스 정의의 뒷부분 — 통제 지점 — 의 이야기입니다. 하네스의 통제 지점 가운데 가장 정교하게
 그려진 것이 판단의 통제이며, 포이에시스의 최신 판본이 집중적으로 강화한 곳이 바로 여기입니다.



3.7.1 의사결정 테이블의 조용한 함정

출발점은 익숙한 **의사결정 테이블**입니다. "조건이 이러하면 결정은 이것"이라는 행들의 목록이지요. 주
택당보대출 도메인에는 심사 과문마다 이런 테이블이 연결되어 있습니다.

그런데 고전적인 운영 방식 — 위에서부터 읽어 처음 맞는 행을 채택하는 방식 — 에는 조용한 함정이 있습니다. 여러 행이 동시에 들어맞는 경우 어느 것이 옳은지 따져 보지 않고 위의 것을 집는다는 것, 그리고 어느 행에도 맞지 않는 경우를 소리 없이 흘려보낸다는 것입니다. 둘 다 시스템이 오류를 내지 않기 때문에, 잘못된 판단이 정상 처리의 얼굴을 하고 흘러갑니다.

3.7.2 함정의 크기를 먼저 측정하다

포이에시스는 이 함정의 크기를 먼저 측정했습니다. 이 도메인의 가능한 입력 조합 1,410가지를 전수 전 개해 보니, 97가지는 여러 규칙이 동시에 들어맞는 진짜 다치(多値) 사례였고, 897가지는 어떤 규칙에도 걸리지 않아 옛 방식이 조용히 떨어뜨리던 사례였습니다. 천 전에 가까운 어긋남이 그럴듯한 겉모습 아래 숨어 있었던 것입니다.

3.7.3 세 가지 처방: 장부, 순서, 금지

그리고 학계의 다치 의사결정 이론을 들여와 세 가지로 답했습니다.

먼저, 숨어 있던 다치 사례와 미포괄 사례를 **전부 장부에 올렸습니다**. 존재를 인정하는 것이 관리의 시작이기 때문입니다.

다음으로, 여러 결정이 경합할 때 **"가장 보수적인 결정부터"**라는 심각도 순서로 동물을 해소하는 판단 트리를 도입하고, 깊이 우선형과 간결형의 대안을 나란히 제시하여 도메인 전문가가 트레이드오프를 보고 선택하게 했습니다. 어떤 규칙에도 걸리지 않는 사례는 테이블마다 작성된 **기본 결정**이 받아내어, 판단의 커버리지는 1,410분의 1,410 — 빈틈 없음 — 이 되었습니다.

그리고 백미가 마지막에 있습니다. **금지 규칙** — "이 조건에서는 이 결정이 절대 나올 수 없다"는 '해서는 안 됨'의 문법 — 194개가 도출되었고, 각 규칙은 기존 규정 문구에서 근거를 물려받으며, 심각한 규칙들은 표준 검증 규격으로 컴파일되어 **설계 검증 시점과 운영 시점 양쪽에서 기계적으로 강제**됩니다. 이 강제는 관문마다 권고와 강제를 선택할 수 있고, 현재 여섯 개 관문 전부가 강제 모드입니다. 도입 방식 조차 신중했습니다. 관문 하나씩, 그림자 검증 기간을 거치는 단계적 확산이었고, 각 단계의 판단 근거가 보고서로 남아 있습니다. 이러한 다차원적인 통제 지점을 통하여 기존에 보지 못하던 위험들을 제거할 수 있게 되었습니다.

3.7.4 희망 사항에서 증명 가능한 성질로

"일이 올바른 방식으로만 이루어지게 한다"는 능력 하네스의 사명은, 여기서 희망 사항이 아니라 **수학적**으로 증명 가능한 성질이 됩니다. 모든 입력에 판단이 존재함(커버리지), 경합 시 보수적으로 기울(심각도 순서), 금지선을 기계가 지킴(금지 규칙의 강제).

에이전트에게 "무엇을 하라"만 가르치는 조직과 "무엇을 절대 하지 말라"까지 강제하는 조직 사이에는 리스크 등급의 차이가 있습니다. 포이에시스는 그 차이를 배포 이후의 사고 대응에서가 아니라, **만드는 단계에서** 만들어 냅니다.

3.8 여섯 번째 능력: 활동 하네스의 기조를 계기판으로 조절한다

앞의 능력이 개별 판단의 통제 지점을 버린다면, 이번 능력은 **활동 레벨** 하네스만이 가질 수 있는 통제 — 흐름 전체의 기조 — 를 다룹니다. 핵심 개념은 **스로틀(throttle)**입니다. 자동차의 가속 페달처럼, 자율

운영되는 활동 전체의 보수성과 공격성을 조절하는 손잡이입니다.

작동 원리는 하나의 회로입니다. 시장 지표, 연체 동향, 운영 데이터 같은 외부·내부 **신호**들이 각자의 가중치와 함께 등록되어 있고, 이 신호들이 합산되어 현재의 **스로틀 점수**가 됩니다. 그 점수는 활동을 이루는 각 심사 관문의 파라미터 — 승인 임계값, 판단의 보수성 편향 — 로 번역되어 일제히 반영됩니다. 그리고 그렇게 조절된 판단의 결과가 관문별 성과 추이와 부실채권 지표와의 상관으로 측정되어, 다시 신호가 되어 돌아옵니다. **신호 → 기초 → 판단 → 성과 → 신호**의 닫힌 고리입니다.

이 회로가 활동 하네스의 자율성에 어떤 의미인지 짚어 볼 가치가 있습니다. 자율 운영의 진짜 위험은 개별 판단의 오류보다, **세상이 변했는데 시스템의 태도가 그대로인 상태**입니다. 스로틀은 그 위험에 대한 답입니다. 시장이 나빠질 때 수백 개의 개별 규칙을 일일이 뜯어고치는 대신, 흐름 전체의 태도를 하나의 정책 레버로 조절할 수 있습니다.

그리고 관리자에게 이것은 사실 낯선 개념이 아닙니다. **여신 정책 위원회가 회의와 공문으로 해 오던 바로 그 일의 계기판 버전**입니다. 달라진 것은 투명성입니다. 이번 분기의 기초 조정이 어느 신호 때문에, 어느 관문에, 얼마만큼 작용했으며, 그 결과 부실률이 어떻게 움직였는지가 전부 기록으로 남습니다. 기초 조절이 감(感)의 영역에서 계기판의 영역으로 옮겨 오는 것입니다.

3.9 일곱 번째 능력: 배포하기 전에 비행해 본다

조립된 하네스가 실제 세계로 나가기 전, 포이에시스는 세 겹의 시험을 요구합니다. 구성이 규격에 맞는가를 보는 **형식 검증**, 내용이 업무 규칙에 맞는가를 보는 **의미 검증**, 그리고 실제 실행을 흉내 내 보는 **모의 실행**. 문장이 문법에 맞는가, 뜻이 맞는가, 굴러 보면 의도대로 움직이는가 — 세 질문이 차례로 던져 집니다.

시험의 도구도 풍부합니다. 관문과 범주의 격자를 채워 **어디가 비었는지를** 드러내는 커버리지 분석이 있고, 시스템이 스스로의 지식 기반을 비평하여 **자율 에이전트에게 부족한 정보가 무엇인지** 식별해 주는 '비평과 보완' 기능이 있습니다.

검증기의 최종 산출이 이 문화의 성격을 잘 보여 줍니다. 현재 도메인의 검증 결과는 오류 0건에 경고 18건인데, 그 경고들조차 "강제되고는 있으나 규제 조항 연결이 아직 안 붙은 금지 규칙" 같은 — 즉, **다음에 할 일의 목록**입니다. 검증은 합격/불합격의 도장이 아니라 백로그 생성기로 기능합니다.

모의 실행은 본격적인 시험 비행장이며, 하네스의 크기에 따라 시험의 성격도 달라집니다. 스킬과 태스크 수준에서는 개별 판단의 정확성이 관심사이고, 활동 수준에서는 에이전트들의 협업 — 단계별로 무슨 일이 일어났는지의 **축차 추적**과 여러 에이전트가 서로 무엇을 주고받았는지의 **상호작용 관계도** — 이 관심사가 됩니다. 미리 준비된 시나리오 묶음에서 사례를 고르거나 직접 지어 실행하고, 수십 수백 개의 시나리오를 한꺼번에 돌리는 배치 실행이 통계적 확신을 더하며, 집계된 성과 대시보드가 결과를 모읍니다.

시스템 곳곳의 문구 하나가 이 문화를 요약합니다. 모의 실행을 거치지 않은 분석 화면에는 '정적 모드'라는 배지가 붙습니다. **실행해 보지 않은 주장에는 꼬리표가 붙는 것입니다.**

관리자에게 이 능력의 의미는 승인 회의의 풍경 변화입니다. 배포 승인 안전에 올라오는 것은 담당자의 의견이 아니라 **비행 기록** — 어떤 시나리오를 몇 건 돌렸고, 추적에서 무엇이 보였으며, 커버리지의 빈칸은 무엇인가 — 이 됩니다.

3.10 여덟 번째 능력: 이력과 관문으로 하네스의 출고를 통제한다

시험을 통과한 하네스에는 두 개의 마지막 규율이 기다립니다. **이력**과 **관문**입니다.

이력의 규율은 소프트웨어 개발의 검증된 관행을 그대로 가져왔습니다. 모든 하네스는 갈래(브랜치)와 확정 기록(커밋)으로 관리되고, 두 판본 사이의 차이를 비교할 수 있으며, 과거 판본으로 되돌릴 수 있습니다. 판본마다 파이프라인 상태가 배지로 붙어 있어, "지금 실제로 운영에 나가 있는 판본은 무엇인가"라는 질문에 조희 한 번으로 답합니다. 테오리아의 캡슐 워크벤치에서 본 원칙이 여기서도 반복됩니다. **하네스의 어떤 판본이 언제 어떤 상태였는지는 기억이 아니라 기록의 문제**여야 한다는 원칙입니다.

관문의 규율은 배포를 하나의 의식(儀式)으로 만듭니다. 배포 전 점검 목록이 있고, 승인 절차가 있으며, 통과한 하네스만이 운영의 무대 — 다스리는 날개 프락시스 — 로 넘어갑니다. 그리고 되돌림(롤백)이 항상 준비되어 있습니다. **나가는 문이 좁고 돌아오는 문이 열려 있는 구조** — 배포 통제의 정석입니다.

여기서 프락시스와의 관계를 한 가지만 짚어 두겠습니다. 포이에시스와 프락시스는 **같은 서버와 같은 데이터베이스를 공유하는 두 개의 얼굴**입니다. 능력, 스킬, 기법, 추론, 에이전트 명부, 감독 체계 — 두 날개가 참조하는 이 핵심 업무 온톨로지들은 문자 그대로 같은 책의 같은 페이지입니다. 만드는 쪽이 넘긴 하네스와 다스리는 쪽이 받은 하네스 사이에 번역도, 복사도, 어긋날 틈도 없습니다. 제1장에서 "비행에는 두 날개가 함께 움직이는 것이 필요하다"고 했던 문장의 공학적 실현이 바로 이 공유 구조이며, 그 무대 위에서 벌어지는 일 — 배포된 하네스 안에서 에이전트가 실제로 일하고 감독받는 이야기 — 은 제4장의 몫입니다.

덧붙여, 조용하지만 인상적인 규율 하나가 이력의 세계를 완성합니다. 도메인 데이터를 직접 고치는 편집 기능조차 고삐가 채워져 있다는 점입니다. 수정이 허용되는 항목은 사전에 선언된 목록으로 제한되고, 저장은 바뀐 부분만 단일 통로로 전송되며, 되돌리기가 준비되어 있고, 변경은 감사의 대상으로 남습니다. **데이터를 고치는 손에도 장갑이 끼워져 있는 것** — 하네스를 만드는 공장 자신이 하네스를 차고 일하는 셈입니다.

3.11 한 벌의 하네스가 태어나기까지

여덟 가지 능력을 하나의 이야기로 이어 붙이면, 포이에시스에서 하네스 한 벌이 태어나는 과정은 대략

이렇게 흘러갑니다.

발단은 테오리아에서 넘어온 구현 대상입니다. 주택담보대출 심사 활동 아래, 심사역의 담보 평가라는 태스크가 접수된 상태로 도착해 있습니다. 근거가 될 도메인 데이터가 연결되어 있음을 시스템이 확인해 주고 나서야 작업이 열립니다. 업무 설계자는 이 태스크가 하는 일을 자연어 시나리오로 서술하고 달성해야 할 목표를 함께 적습니다. AI가 그 이야기를 구조화된 설계 초안으로 발전시키지만, 그 작문은 자유 작문이 아니라 하네스의 규격 안에서만 태어나는 작문입니다. 상태는 작성 중으로 바뀝니다.

설계가 두터워지는 동안, 담기는 지식 하나하나에 신분증이 붙습니다. 담보 평가에 적용되는 업무 규칙에는 근거 규제 조항이 링크로 연결되고, 이 태스크에서 허용되는 행위의 범위와 결재 체인이 권한 매트릭스로 선언되며, 참조하는 데이터 값에는 출처와 불확실성 프로파일이 기록됩니다. 태스크가 요구하는 능력들 — 담보 가치 산정, 소득 서류 검증, 규정 조회 — 은 각자의 도메인 지식과 추론 템플릿을 매단 능력 하네스로 꾸러지고, 판단의 관문에는 의사결정 테이블이 연결됩니다. 도메인 전문가는 규칙이 경합할 때의 보수성 순서를 확정하고, 어느 규칙에도 걸리지 않는 사례를 받아 낼 기본 결정을 적어 넣으며, "이 조건에서는 이 결정이 절대 나올 수 없다"는 금지 규칙들이 규정 문구에서 근거를 물려받아 강제 모드로 걸립니다.

이제 시험 비행입니다. 형식 검증이 규격과의 일치를, 의미 검증이 업무 규칙과의 일치를 확인하고, 커버리지 분석이 빈칸을 드러내며, 모의 실행이 준비된 시나리오 묶음과 배치 실행으로 통계적 확신을 쌓습니다. 검증기가 남긴 경고 열여덟 건은 불합격 판정이 아니라 다음에 할 일의 목록으로 백로그에 등록됩니다. 상태는 검증됨이 되고, 배포 승인 안건에는 담당자의 의견이 아니라 비행 기록 — 어떤 시나리오를 몇 건 돌렸고, 추적에서 무엇이 보였는지 — 이 첨부됩니다. 승인 관문을 지난 하네스는 커밋과 브랜치의 이력을 매단 채 배포된 상태로 프락시스의 무대에 오릅니다.

이 여정에서 눈여겨볼 것은, 어느 단계에서도 다스림이 만들기의 뒤로 미뤄지지 않는다는 점입니다. 규격은 설계의 시작부터 있었고, 근거는 지식이 담기는 순간에 붙었으며, 금지선은 배포 전에 이미 기계에 새겨졌습니다. 만드는 날개가 빠른 이유는 다스림을 생략해서가 아니라, 다스림을 공정 그 자체로 만들었기 때문입니다.

3.12 맺으며: 만드는 날개의 관리자에게

포이에시스가 조직에 부여하는 능력을 한 문단으로 모으면 이렇게 됩니다. **활동·태스크·능력**이라는 세 크기의 하네스를, 파이프라인 위에서 진행을 한눈에 쥐며 짓는다. 현업의 시나리오에서 규격 있는 설계를 얻고, 하네스에 담을 모든 지식에 규제·권한·출처의 신분증을 붙이며, 능력에서 활동까지 일하는 주체를 조직처럼 설계하고, 통제 지점을 수학으로 버리며, 자율 운영의 기초를 계기판으로 조절하고, 배포 전에 반드시 비행해 보며, **이력과 관문으로 출고를 통제한다**. 여덟 능력의 공통 문법은 하나입니다. 만드는 속도를 위해 다스림을 미루지 않고, 다스림을 하네스 그 자체로 만들어 내는 것.

이 날개를 맡을 관리자에게 남기고 싶은 당부도 이야기의 흐름 그대로입니다. 새 자동화 과제가 들어오면 "에이전트를 만들까"가 아니라 "이것은 능력 증강인가, 역할의 태스크인가, 활동의 자율 운영인가"

라는 크기의 질문부터 던지시기 바랍니다. 능력 하네스로 충분한 곳에 활동 하네스를 짓거나 활동이 필요한 곳에 능력만 쥐여 주는 크기의 오판이 이 영역에서 가장 비싼 실수이며, 능력 하네스에서 적립된 신뢰만이 태스크와 활동의 자율성을 정당화한다는 도입 사다리의 원칙이 그 질문의 나침반이 됩니다. 진척을 이야기할 때는 **파이프라인의 다섯 상태 — 접수됨, 작성 중, 검증됨, 배포됨, 반려됨 — 를 회의의 언어로** 삼으십시오. 진척 보고는 이 다섯 단어면 충분하고, 이 다섯 단어여야 정확합니다.

판단의 통제에 관한 일들은 기술 부서가 아니라 도메인 전문가의 책상에 놓여야 합니다. 규칙이 경합할 때 무엇이 더 보수적인 결정인지, 어느 규칙에도 걸리지 않는 사례를 어떻게 받아 내지는 리스크 정책의 문제이고, 시스템은 그 정책을 적을 칸을 마련해 두었을 뿐입니다. **보수성 원칙과 기본 결정의 작성을 전문가의 정식 업무로** 부여하시고, 검증이 남긴 경고의 대부분을 차지하는 **금지 규칙의 규제 근거 보강을 다음 분기의 과제로** 올려 두시기 바랍니다. 경고 목록이 곧 업무 목록입니다.

배포의 관문에서는 단호함이 미덕입니다. 시나리오 몇 건을 돌렸고, 배치를 몇 회 실행했으며, 추적에서 무엇이 보였는지 — 이 세 줄이 없는 승인 요청은 반려하는 것이 이 도구의 문화에 맞습니다. **모의 실행 없는 배포 승인을 관행에서 지우십시오.** 그리고 활동 하네스의 자율성이 커질수록 무거워지는 레버, **스로를 회로의 주인을** 정해 두십시오. 신호의 가중치와 관문 연결은 한번 정하고 잊는 값이 아니라, 성과 피드백을 보며 주기적으로 손보는 정책 레버입니다.

독수리의 왼쪽 날개는 이렇게 일합니다. 이야기가 들어와 세 크기의 하네스가 되어 나가고, 그 사이의 모든 손길에 규격과 근거와 기록이 따라붙는 곳. 다음 장에서는 오른쪽 날개 프락시스로 건너가, 배포된 하네스 안에서 에이전트들이 실제로 일하고 다스려지는 무대와 함께, 이번에 아껴 둔 두 이름 — 하네스의 규격인 넥서스와 그것을 충실히 실행하는 미메시스 엔진 — 의 이야기를 온전히 펼치겠습니다.

제3장 핵심 요약: 포이에시스에는 에이전트가 아니라 에이전트 하네스를 조립하는 곳이며, 하네스에는 능력(증강)·태스크(대행)·활동(자율 운영)의 세 크기가 있다. 여덟 가지 능력 — 파이프라인 가시성, 시나리오 기반 설계, 근거의 신분증, 조직형 설계, 수학으로 버린 통제 지점, 스로틀 기조 조절, 배포 전 모의 비행, 이력과 관문의 출고 통제 — 의 공통 문법은 다스림을 하네스 그 자체로 만들어 내는 것이다.

제 4 장. 프락시스: 다스리고 조정하는 날개

이 장에서 다루는 것: 그동안 아껴 두었던 두 이름의 정체 — 하네스의 규격인 넥서스와 그것을 충실히 실행하는 미메시스 엔진 — 그리고 배포된 하네스가 실제로 일하고 다스려지는 운영의 무대. 예시 도메인은 대출 실사(duel diligence) 업무를 사용합니다.

4.1 들어가며: 독수리의 마지막 조각

이 문서는 이제 네 번째 장에 이르렀습니다. 제1장에서 독수리의 해부 구조를 보았고, 제2장에서 판단하는 머리 테오리아를, 제3장에서 만드는 왼쪽 날개 포이에시스를 걸었습니다. 제3장은 하나의 약속으로 끝났습니다. 하네스의 규격인 **넥서스**와 그것을 충실히 실행하는 **미메시스 엔진**의 이야기는, 배포된 에이전트가 실제로 일하는 무대와 함께 온전히 펼쳐졌다는 약속이었습니다.

이번 장이 그 약속을 지키는 자리입니다. **AOEDE-프락시스**는 오른쪽 날개, 다스리는 날개입니다. 포이에시스가 세 크기의 하네스 — 능력, 태스크, 활동 — 를 조립해 관문을 통과시키면, 그 하네스는 프락시스의 무대에 오릅니다. 여기서 에이전트들은 실제 문서를 받아 실제 판단을 내리고, 사람은 그 실행을 실시간으로 지켜보며, 숙련도가 측정되고, 인증이 갱신되고, 변경이 통제되고, 이상이 알려집니다. 제1장의 문장을 빌리면 — "통제 없이 만들기만 하는 조직은 리스크를 대량생산하고, 만들지 않고 통제만 하는 조직은 아무것도 생산하지 못한다" — 프락시스는 그 문장의 뒷부분이 헛말이 되지 않게 하는 곳입니다.

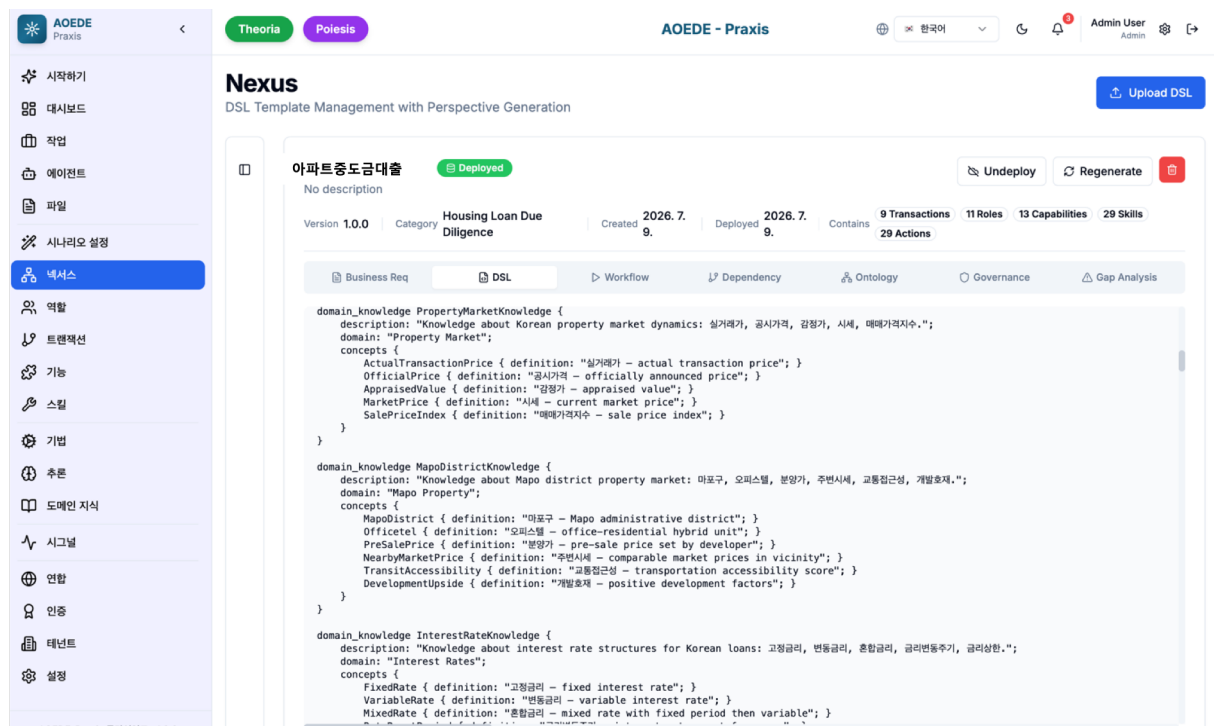
프락시스의 첫 화면이 이 문서 전체의 세계관을 한 장의 그림으로 요약하고 있다는 점도 언급해 둘 만합니다. 시작하기 화면에는 AOEDE — 에이전트 온톨로지 구상 및 배포 환경 — 의 전체 지도가 펼쳐지고, 테오리아·포이에시스·프락시스라는 세 워크플레이스와, 그 한가운데의 NEXUS, 그리고 에이전트를 이루는 능력의 수레바퀴 — 인지, 지식, 메모리, 추론, 상호작용, 학습, 페르소나, 프레젠테이션 — 가 그려져 있습니다. 지금까지 세 장에 걸쳐 이야기해 온 것들이 운영자의 첫 화면에 이미 한 장으로 걸려 있는 셈입니다.

이 장도 앞선 장들과 같은 방식을 따릅니다. 화면 안내가 아니라 **능력의 이야기**입니다. 다만 순서에는 이 유가 있습니다. 먼저 미뤄 두었던 두 이름 — 넥서스와 미메시스 — 을 밝히고, 그 위에서 프락시스가 조직에 부여하는 다스림의 능력들을 걸어가겠습니다. 예시 도메인은 포이에시스에서 조립된 그 하네스가 실제로 굴러가는 **대출 실사(duel diligence)** 업무입니다.

4.2 넥서스: 하네스의 규격이자, 하나의 원본에서 피어나는 일곱 개의 관점

먼저 첫 번째 이름입니다. 제1장에서 넥서스 하네스를 "행동, 통제, 거버넌스로 이루어진 황금 벨트"라 불렀고, 제3장에서는 그 조립을 다루면서도 이름의 정체는 아껴 두었습니다. 이제 말할 수 있습니다. **넥서스**는 하네스가 기록되는 규격이자, 만드는 날개와 다스리는 날개가 주고받는 계약 문서의 형식입니다.

다.



4.2.1 아흔일곱 종의 어휘로 쓰인 계약서

구체적으로 넥서스는 CC-DSL(Cognition Control-Domain Specific Language)이라는 전용 언어(도메인 특화 언어)로 기술된 하나의 패키지입니다. 그 안에는 지금까지 세 장에서 이야기한 모든 것 — 비즈니스 영역, 작업과 단계, 역할, 기능, 스킬, 기법, 지식 베이스, 추론 템플릿, 그리고 감독 규칙 — 이 한 벌의 정의로 담깁니다.

놀라운 것은 규모입니다. 이 규격은 **아흔일곱 종의 정의 요소와 스물여덟 종의 연결 관계**로 이루어져 있어, 에이전트 조직 하나를 기술하는 데 필요한 어휘가 사실상 빠짐없이 갖춰져 있습니다. 사람 조직으로 치면 조직도, 직무기술서, 업무 매뉴얼, 권한 규정, 감사 기준이 하나의 일관된 언어로 쓰여 있는 셈입니다. 여기서 사용하는 CC-DSL은 넥서스를 위한 전용 언어로, AOEDE 하네스를 표준화하기 위해 개발되었습니다.

4.2.2 두 개의 문, 그리고 자동 정렬

프락시스에서 넥서스는 두 개의 문으로 들어옵니다. 하나는 **시나리오 설정** 마법사입니다. DSL 파일들을 올리면 시스템이 정의가 참조보다 먼저 오도록 — 메인, 지식, 기법, 추론, 스킬, 핸들러의 순서로 — **종속성에 따라 자동 정렬**하여 하나로 병합하고, 검증을 거쳐 감지된 구성 요소의 목록 — 비즈니스 영역 몇 개, 역할 몇 개, 스킬 몇 개, 감독 규칙 몇 개 — 을 보여 준 뒤 넥서스를 생성합니다. 다른 하나는 넥서스 페이지의 직접 업로드와 배포입니다. 어느 문으로 들어오든, 배포된 넥서스에는 '배포됨' 표식이 붙고 운영의 리포지토리에 등록됩니다.

4.2.3 일곱 개의 관점: 어긋날 수 없는 문서들

그런데 넥서스 페이지의 진짜 가치는 등록이 아니라 **번역**에 있습니다. 하나의 넥서스를 선택하면 시스템은 같은 원본을 일곱 개의 관점으로 풀어 보여 줍니다. 경영진의 언어로 쓴 **비즈니스 요구사항**, 원문 그대로의 **DSL 소스**, 역할별 흐름을 그린 **워크플로우 스윘레인과 BPMN**, 액션들의 입력과 출력이 어떻게 맞물리는지의 **데이터 의존성 다이어그램**, 정의 요소들의 관계 전체를 조망하는 **온톨로지 맵**, 어떤 감독 규칙이 어디에 걸려 있는지의 **거버넌스 추적**, 그리고 정의의 빈틈을 짚는 **갭 분석**입니다.

관리자에게 이 일곱 관점이 뜻하는 바는 분명합니다. 지금까지 조직에서 요구사항 문서, 프로세스 도면, 데이터 흐름도, 통제 매트릭스는 서로 다른 사람이 서로 다른 도구로 그려 서로 어긋나기 마련인 네 장의 종이였습니다. 넥서스에서는 이들이 **하나의 원본에서 기계적으로 파생되는 관점들**이므로, 원리상 서로 어긋날 수가 없습니다. "문서와 실체가 다르다"는 오래된 병이, **문서가 곧 실제인 구조**로 치유되는 것입니다.

4.3 미메시스 엔진: 작업이라는 이름의 충실한 실연

두 번째 이름입니다. 제1장은 미메시스를 이렇게 소개했습니다 — "엔진은 결과를 지어내지 않는다. 온톨로지가 정의한 내용을, 하네스가 허용하는 범위 안에서 그대로 수행할 뿐이다. 엔진이 강력한 이유는 정확히 시키는 대로만 움직이기 때문이다." 프락시스에서 그 문장은 **작업(Job)**이라는 형태로 실물이 됩니다.

4.3.1 작업의 탄생: 명시된 트레이드오프

작업의 탄생부터 봅시다. 운영자는 새 작업을 만들면서 배포된 넥서스(DSL 템플릿)를 선택하고, 처리할 실제 문서들 — 계약서, 재무제표, 감정평가서 — 을 첨부하며, 우선순위와 실행 조건을 정합니다.

흥미로운 것은 처리 전략의 선택지입니다. 시스템이 알아서 정하는 **자동**, 한 번에 한두 페이지씩 정독하는 **품질 우선**, 중간 크기의 **균형**, 대량을 빠르게 처리하는 **처리량**, 그리고 직접 지정하는 **사용자 정의**. 대량 문서를 다뤄 본 관리자라면 이 다이얼의 의미를 즉시 알아볼 것입니다. **정확도와 속도의 트레이드오프**가 담당자의 감이 아니라 명시적 설정으로 올라와 있는 것입니다.

4.3.2 실연의 무대: 3 층 구조와 살아 있는 시각화

작업이 시작되면 미메시스의 실연이 펼쳐집니다. 실행의 단위는 넥서스가 정의한 그대로 — **트랜잭션이 단계를 거느리고 단계가 액션을 거느리는 3층 구조**입니다. 화면 상단의 이중 진행 막대는 트랜잭션 수준과 액션 수준의 진척을 동시에 보여 주고, 반복 처리 중이라면 몇 회차인지까지 표시합니다.

워크플로우 탭에서는 실행 흐름이 방향 그래프(DAG)로 살아 움직이고, **에이전트 탭**에서는 여러 에이전트가 시간축 위에서 일을 주고받는 스윘레인과 상호작용 관계가 그려지며, **이벤트 탭**에는 무슨 일이 언제 일어났는지의 흐름이 초 단위로 쌓입니다.

4.3.3 계보가 다운로드 버튼으로 존재하는 상태

그리고 실연의 끝에서 미메시스의 성격이 가장 선명하게 드러납니다. **산출물 탭**에는 최종 결과물만 있는 것이 아닙니다. 모든 **액션 하나하나의 입력과 출력이 쌍으로 보존되어, 각각 내려받을 수 있습니다.** 대출 실사 작업이라면 최종적으로 마크다운 보고서와 HTML 대시보드가 만들어지지만, 그 보고서의 어떤 숫자든 "어느 액션이, 어떤 입력을 받아, 어떤 출력을 냈는가"의 사슬로 거슬러 올라갈 수 있습니다.

결과를 지어내지 않는 엔진이란 바로 이런 것입니다. 모든 **결론에 계보가 있고, 그 계보가 다운로드 버튼으로 존재하는 상태.**

관리자의 언어로 이 절을 접했습니다. 전통적인 자동화에서 "그 결과 어떻게 나온 거예요?"는 개발팀에 문의하는 질문이었습니다. 미메시스의 무대에서 그것은 **탭을 하나 여는 일**입니다.

4.4 약속으로 묶인 실행: 요청되고, 약속되고, 진술되고, 수락된다

미메시스가 실행하는 트랜잭션에는 눈여겨볼 만한 구조가 하나 더 있습니다. 각 트랜잭션은 **요청됨 → 약속됨 → 진술됨 → 수락됨**이라는 네 위상을 거칩니다. 언어행위 이론(speech act theory)에 뿌리를 둔 이 구조는, 일이란 작업의 나열이 아니라 **행위자 사이의 약속의 고리**라는 관점을 담고 있습니다. 발신자가 일을 요청하고, 실행자가 하겠다고 약속하고, 했다고 진술하고, 발신자가 결과를 수락함으로써 고리가 닫힙니다.

그래서 프락시스의 실행 장부에는 **사실과 나란히 약속이** 일급 항목으로 존재하며, 약속에는 상태가 있습니다 — **활성, 이행됨, 위반됨, 취소됨**. 트랜잭션 화면에서는 각 트랜잭션의 발신자와 실행자, 트리거, 그리고 단계별 액션들이 흐름도와 트리트로 펼쳐집니다.

이것이 관리 관점에서 왜 중요할까요. 에이전트 시스템의 책임 소재는 흔히 안개 속에 있습니다. 여러 에이전트가 얹혀 일하다 무언가 잘못되면, 어디서부터 어긋났는지 짚기 어렵습니다. 약속의 고리는 그 안개를 걷어 냅니다. 모든 일에는 요청한 자와 약속한 자가 있고, 위반된 약속은 '위반됨'이라는 상태로 실행 장부에 남습니다. **책임의 추적이 사후 조사가 아니라 자료 구조인 것입니다.**

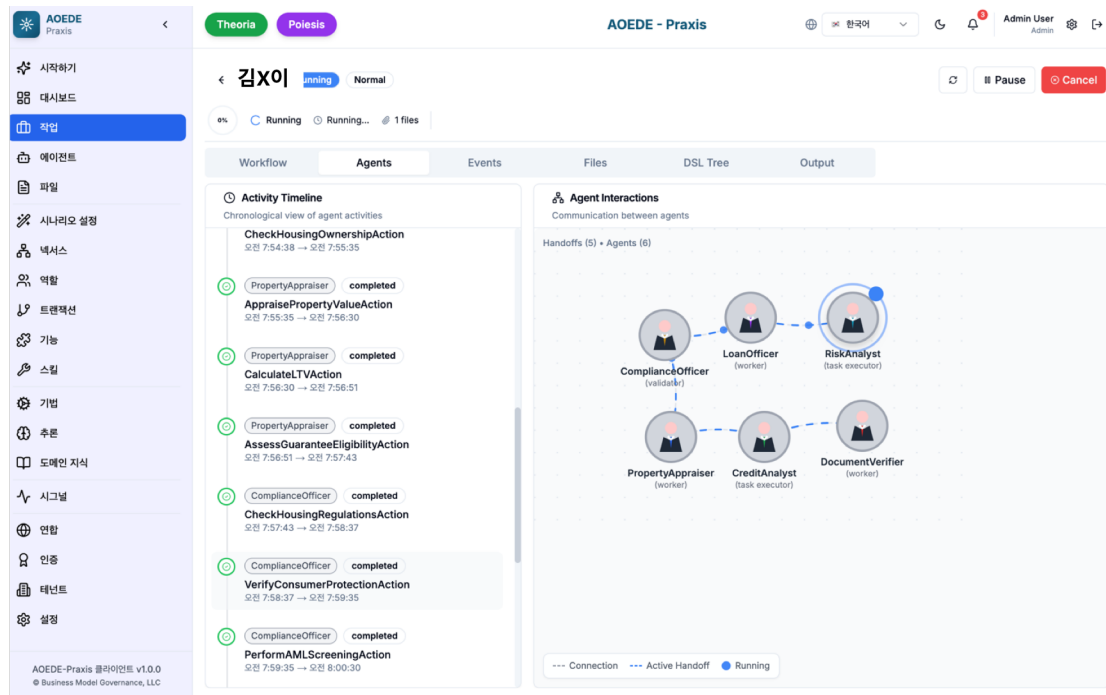
4.5 살아 있는 명부: 에이전트, 역할, 그리고 여덟 갈래의 능력

무대 위의 배우들을 볼 차례입니다.

4.5.1 에이전트 명부: 사람과 AI가 같은 문법으로

에이전트 화면은 등록된 에이전트들의 살아 있는 명부입니다. 각 에이전트에는 유형 — LLM 기반, 규칙 기반, 하이브리드, 그리고 인간 — 이 있고, 상태와 마지막 하트비트(생존 신호)가 있으며, 어떤 역할을 맡아 어떤 기능·스킬·기법을 보유하고 있는지가 트리트로 펼쳐집니다.

눈에 띄는 것은 '인간'이 에이전트 유형의 하나라는 점입니다. 이 명부에서 사람과 AI는 같은 문법으로 등록되는 동료입니다. 이는 수사가 아니라 실용입니다. 사람과 AI가 섞여 일하는 워크플로우를 하나의 실행장부로 관리하려면, 둘을 같은 어휘로 기술할 수 있어야 하기 때문입니다.



이 선택이 함의하는 바는 한 걸음 더 나아갑니다. 사람과 AI가 같은 문법으로 등록된다는 것은, 업무를 설계할 때 "이 태스크는 사람 것, 저 태스크는 AI 것"이라는 이분법 대신 "이 역할을 지금 누가 맡는 것이 최선인가"라는 배정의 질문을 던질 수 있다는 뜻입니다. 오늘 사람이 맡던 역할을 내일 검증된 에이전트가 이어받아도, 또는 그 반대여도, 흐름의 설계 자체는 바뀌지 않습니다. 하네스가 역할에 걸려 있지 개인에 걸려 있지 않기 때문입니다. 사람 조직에서 인수인계가 그토록 취약한 이유가 업무가 사람에게 붙어 있기 때문임을 떠올리면, 이 문법의 실용적 가치가 분명해집니다.

4.5.2 역할 등록부: 위원회 의결까지 어휘 안으로

역할 화면은 그 배우들이 맡는 배역의 등록부입니다. 역할에는 유형 — 외부, 인간, AI, 인간 그룹 — 이 있고, 그룹 역할에는 정족수와 투표 규칙(과반, 만장일치, 가중)까지 정의됩니다. 위원회 의결 구조가 하네스의 어휘 안에 들어와 있는 것입니다. 각 역할이 수행할 수 있는 허가된 작업의 목록과 위임 가능 대상이 명시되고, 지금 이 역할을 실제로 수행 중인 에이전트들이 나란히 조회됩니다.

4.5.3 에이전톨로지: 배우의 내면을 해부하다

그리고 에이전톨로지 화면이 배우의 내면을 해부합니다. 에이전트의 능력은 여덟 갈래로 분류됩니다. 핵심 AI 처리를 맡는 인지, 정보 접근과 검색의 지식, 맥락과 상태 관리의 메모리, 논리와 의사결정의 추론, 사람·에이전트와의 소통인 상호작용, 적응과 개선의 학습, 정체성과 행동 양식의 페르소나, 출력 형식의 프레젠테이션. 여기에 입력과 출력 양쪽에 걸린 일곱 겹의 안전 필터가 더해집니다. 제3장에서 "AI의 지능조차 등록된 능력"이라 했던 원칙이, 운영 측에서는 이 여덟 갈래의 분류 체계로 살아 있습니다.

한 가지 개념 구분이 화면에 명시되어 있어 옮겨 둡니다. **능력(capability)**은 **비즈니스 계층 — 시스템**이 하는 일이고, **역량(competency)**은 **에이전톨로지 계층 — AI 에이전트**의 인지 능력입니다. "우리 시스템은 담보 평가를 한다"(능력)와 "이 에이전트는 문서에서 개체를 추출할 줄 안다"(역량)를 구분하는 어휘가 조직에 생기는 것입니다.

4.6 공유 장부의 반대편: 같은 능력을 운영의 눈으로 읽다

제3장에서 두 날개가 **같은 서버, 같은 데이터베이스**를 공유하는 두 얼굴이라 했습니다. 프락시스에서 그 공유 장부의 반대편 — 능력, 스킬, 기법, 추론, 도메인 지식 — 을 열어 보면, 만드는 쪽에서 조립한 능력 하네스가 운영의 눈으로 어떻게 읽히는지 알 수 있습니다.

스킬 화면에서 각 스킬은 자신이 구현하는 기능, 사용하는 기법, 참조하는 도메인 지식과 함께 성숙도를 드러냅니다. **기법** 화면은 각 기법의 입력, 단계별 타임라인, 출력 요구사항 — 필수 필드와 선택 필드 — 그리고 추출 프롬프트까지 보여 줍니다.

추론 화면이 특히 인상적입니다. 각 추론 템플릿에는 사고 체인의 목표와 핵심 포인트, 퓨샷 예시가 정리되어 있고, 무엇보다 **LLM 제약조건**과 **드리프트 방지 규칙** — 모델이 시간이 지나며 원래 의도에서 미끄러지는 것을 막는 장치 — 이 명세로 붙어 있습니다. 추론 유형의 어휘도 풍부합니다. 사고 사슬, 사고 트리, ReAct, 성찰(Reflexion)부터 개체관계 추론, 인과 추론, 비교 분석까지 열세 가지 유형이 표준 분류로 존재합니다.

이 장부의 모든 항목에는 두 개의 배지가 따라다닙니다. 하나는 **상태** — 초안, 활성화, 배포됨, 폐기 예정, 보관됨 — 이고, 다른 하나는 **변경 통제 수준**입니다. 후자는 다음 절의 주인공이니, 여기서는 한 문장만 남기겠습니다. 능력 하네스의 정의 — "포괄적 지식과 통제 지점의 최소 완결 단위" — 는 배포되는 순간 사라지는 설계 의도가 아니라, **운영자가 매일 열어 보는 장부의 실물**입니다.

4.7 다스림의 세 도구: 변경 통제, 시그널, 감독 규칙

다스림은 날개의 핵심 임무는 결국 통제입니다. 프락시스는 세 개의 도구로 이를 수행합니다.

첫 번째 도구는 **변경 관리**입니다. 장부의 각 항목에는 변경 통제 수준이 선언되어 있습니다 — 자유 변경, **검토 필요**, 그리고 **MRM 필요**. MRM, 즉 모델 리스크 관리(Model Risk Management)는 금융권 관리자에게 익숙한 무계의 단어일 것입니다. 규제가 요구하는 모델 검증 절차가 필요한 항목은 그렇게 표시되고, 변경 관리 대시보드에는 검토 대기 중인 항목들이 모여 승인 또는 거부를 기다리며, 처리의 이력이 남습니다. 포이에시스에서 하네스를 만들 때 새겨진 다스림이, 운영 중의 **변경**에까지 이어지는 것입니다. 한번 검증된 하네스라도, 그 부품을 고치는 일은 다시 관문을 지나야 합니다.

두 번째 도구는 **시그널과 모니터링**입니다. 실시간 메트릭과 시그널 활동이 흐르고, 알림은 정보·경고·긴급의 심각도로 분류되어 대시보드 최상단 — 전체 작업, 활성화 에이전트, 트랜잭션과 나란히 — 에 활성

알림 수로 집계됩니다. 제3장의 스로틀 회로가 신호를 소비해 기조를 조절했다면, 프락시스는 그 신호가 태어나고 감시되는 현장입니다.

그리고 세 번째 도구가 **감독 규칙**입니다. 넥서스에 담겨 배포된 감독 온톨로지 — 거버넌스와 감독의 규칙들 — 가 운영 측에서 조회되고, 앞서 본 넥서스의 거버넌스 추적 관점이 어떤 규칙이 어떤 실행 지점에 걸려 있는지를 보여 줍니다. 실행 기록의 상세함조차 통제 대상입니다. 감사 수준은 없음부터 기본, 상세, 전체까지 단계로 선언됩니다.

세 도구를 관리자의 시선으로 묶으면 이렇습니다. 무엇이 바뀌는가(변경 통제), 무엇이 벌어지는가(시그널), 무엇이 지켜지는가(감독 규칙) — 다스림의 세 질문에 각각 전담 도구가 있고, 세 도구가 같은 장부 위에서 작동합니다.

4.8 신뢰의 통장: 인증과 숙련도

제3장은 "자율성은 선언하는 것이 아니라 작은 하네스에서부터 적립하는 것"이라 했습니다. 프락시스는 그 적립이 기록되는 통장이 있습니다. **인증과 숙련도**입니다.

에이전트의 스킬 숙련도는 네 단계로 측정됩니다 — **미훈련, 초보, 숙달, 전문가**. 그리고 숙련의 주장은 시험으로 증명됩니다. 인증 화면에서 운영자는 에이전트를 골라 **테스트 스위트** — 테스트 케이스들과 합격 기준으로 이루어진 시험지 — 에 응시시키고, 결과는 점수와 함께 인증 기록으로 남습니다.

인증에는 **만료일**이 있습니다. 한 번만 자격이 영원하지 않다는 것 — 사람의 전문 자격이 보수 교육과 갱신을 요구하듯, 에이전트의 자격도 갱신을 요구합니다. 모델이 바뀌고, 데이터가 바뀌고, 규정이 바뀌는 세계에서 이는 당연한 요구입니다.

이 장치가 하네스의 세 단계와 만나는 지점이 핵심입니다. 능력 하네스에서 태스크 하네스로, 태스크에서 활동의 자율 운영으로 올라가는 사다리의 각 단은 결국 "이 에이전트를 얼마나 믿을 수 있는가"라는 질문인데, 인증과 숙련도는 그 질문에 **점수와 유효기간이 있는 답**을 제공합니다. 자율성의 확대가 담당 임원의 확신이 아니라 인증 기록의 축적 위에서 결정되는 조직 — 그것이 이 통장이 가능하게 하는 상태입니다.

4.9 날개 너머로: 연합과 테넌트

프락시스의 시야는 한 조직의 담장 안에서 끝나지 않습니다.

연합(Federation) 화면은 A2A(에이전트 대 에이전트) 프로토콜로 연결된 바깥 세계의 명부입니다. 다른 조직의 에이전트가 엔드포인트와 함께 등록되고, 조직별로 분류되며, 온라인 여부가 추적됩니다. 그리고 운영자는 특정 능력을 지정해 — 대상 에이전트를 고르거나 "사용 가능한 아무 에이전트"에게 — **능력을 호출**할 수 있습니다. 우리 하네스가 갖지 못한 능력을 연합의 이웃에게 빌리는 것입니다.

안으로는 **테넌트** 관리가 있습니다. 하나의 프락시스 위에서 여러 사업부, 여러 법인이 각자의 담장을 갖고 운영되며, 대시보드는 테넌트별로 나눠 볼 수 있습니다.

이 두 방향이 함께 가리키는 그림이 있습니다. 에이전트 경제는 한 회사의 한 시스템으로 닫히지 않을 것입니다. 조직 안에는 여러 주체가, 조직 밖에는 여러 파트너가 각자의 에이전트를 갖게 될 것이고, 그때 필요한 것은 **신원과 능력과 호출의 표준적인 문법**입니다. 프락시스의 연합과 테넌트는 그 문법의 이른 실물입니다.

4.10 어느 오후의 실사 작업: 계보를 거슬러 오르다

프락시스의 하루를 한 장면으로 접어 보겠습니다. 오후, 운영자가 새 작업을 만듭니다. 배포된 넥서스가 운데 대출 실사 템플릿을 고르고, 처리할 실제 문서들 — 계약서, 재무제표, 감정평가서 — 을 첨부하고, 오늘은 정확도가 우선이므로 처리 전략 다이얼을 품질 우선에 맞춥니다. 실행 버튼을 누르는 순간부터 미메시스의 실행이 시작됩니다.

화면 상단의 이중 진행 막대가 트랜잭션 수준과 액션 수준의 진척을 함께 채워 가고, 워크플로우 탭에서는 실행 흐름이 방향 그래프로 살아 움직입니다. 에이전트 탭을 열면 여러 에이전트가 시간축 위에서 일을 주고받는 스위치라인이 그려지고, 이벤트 탭에는 무슨 일이 언제 일어났는지가 초 단위로 쌓입니다. 그 사이 각 트랜잭션은 요청되고, 약속되고, 진술되고, 수락되는 네 위상을 조용히 통과합니다. 어느 약속 하나가 위반된 상태로 남는다면, 그것은 안개가 아니라 장부의 한 줄이 됩니다.

작업이 끝나면 산출물 탭에 마크다운 보고서와 대시보드가 놓입니다. 운영자는 보고서의 담보 평가액 하나를 골라 계보를 거슬러 오릅니다. 그 숫자를 낸 액션이 무엇이었는지, 그 액션이 받은 입력이 무엇이었는지, 그 입력은 어느 앞 단계의 출력이었는지 — 사슬의 마디마다 내려받기 버튼이 붙어 있습니다. 예전 같으면 개발팀에 문의 티켓을 열고 며칠을 기다렸을 질문이, 탭 몇 번의 일이 된 것입니다. 이 장면 하나에 다스리는 날개의 존재 이유가 압축되어 있습니다. 실행은 빠르되, 어느 결론도 자기 계보를 잃지 않는다는 것.



4.11 맺으며: 독수리가 완성되다

네 장을 관통한 여정을 한 문단으로 접겠습니다. **테오리아**에서 비즈니스가 온톨로지로 언어화되고 두 겹의 하네스로 다스려졌습니다. **포이에시스**에서 그 언어가 활동·태스크·능력 세 크기의 하네스로 조립되고, 판단의 빈틈과 금지선이 수학으로 버려진 뒤, 시험 비행과 관문을 거쳤습니다. 그리고 **프락시스**에서 하네스는 넥서스라는 계약의 형식으로 배포되고, 미메시스 엔진이 그것을 계보가 남는 방식으로 실연하며, 약속의 고리와 살아 있는 명부와 세 개의 통제 도구와 신뢰의 통장이 그 실행을 다스립니다. 머리가 보고, 왼쪽 날개가 만들고, 오른쪽 날개가 다스리고, 엔진이 실연하고, 발톱이 실제 업무를 움켜잡니다. **독수리가 완성된 것입니다.**

다스리는 날개를 맡을 관리자에게 남길 당부도 이 무대의 감각 그대로입니다. 무엇보다 **작업 산출물의 계보를 감사의 표준으로** 삼으시기 바랍니다. 모든 액션의 입력과 출력이 내려받아지는 시스템에서 결과만 보고 넘어가는 검토는 도구를 낭비하는 일이니, 중요 작업은 표본을 뽑아 계보를 거슬러 확인하는 절차를 정례화하십시오. **변경 통제 수준의 지정은 리스크 부서와 함께** 정할 일입니다. 어떤 스킴과 추론 템플릿이 MRM 대상인지는 기술 판단이 아니라 규제 대응의 판단이며, 지정이 끝나는 순간 변경 관리 대시보드의 대기 항목이 곧 리스크 부서의 업무 목록이 됩니다.

운영의 리듬 속에서 잊히기 쉬운 것들도 있습니다. **인증의 만료를 방치하지 마십시오.** 만료된 인증으로 일하는 에이전트는 갱신 교육을 건너뛴 직원과 같으므로, 만료 임박 인증의 점검은 월례 운영 회의의 고정 안건이어야 합니다. 그리고 트랜잭션 장부에서 '**위반됨**' 상태의 빈도와 패턴을 지표로 읽으시기 바랍니다. 약속 위반은 어느 하네스의 어느 이음매가 약한지를 알려 주는 가장 정직한 신호입니다. 마지막으로, 담장 밖의 질문 하나를 미리 당겨 두십시오. **연합에 내놓을 능력과 빌릴 능력의 정책** — 우리 에이전트의 어떤 능력을 바깥에 등록하고, 바깥의 어떤 능력을 어떤 조건에서 호출할지는, 기술이 성숙한 뒤에

정하면 늦는 지금의 거버넌스 질문입니다.

제1장은 이렇게 끝났습니다 — "볼 수 있을 만큼만 온톨로지를 정의하라. 행동할 수 있을 만큼만 에이전트를 만들라. 모든 행동을 통제와 거버넌스에 매어 두라. 엔진이 충실히 실행하게 하고, 발톱이 실제 일을 움켜쥐게 하며, 독수리가 실제로 쥐는 법을 배운 것에 맞추어 온톨로지가 자라나게 하라." 네 장을 지나온 지금, 그 문장의 모든 구절에 실물이 생겼습니다. 이제 남은 것은 그 독수리가 실제 비즈니스의 지면에 발톱을 내리는 장면 — 제5장의 이야기입니다.

제4장 핵심 요약: 넥서스는 아혼일곱 중의 정의 요소로 이루어진 하네스의 규격이자 두 날개가 주고받는 계약 문서이며, 하나의 원본에서 일곱 개의 관점이 기계적으로 파생되므로 문서와 실체가 어긋날 수 없다. 미메시스 엔진은 모든 액션의 입출력을 쌍으로 보존하여 모든 결론에 다운로드 가능한 계보를 남긴다. 약속의 고리, 살아 있는 명부, 세 가지 통제 도구(변경·시그널·감독 규칙), 그리고 인증·숙련도라는 신뢰의 통장이 운영을 다스린다.

제 5 장. CLAW: 발톱이 지면에 닿다

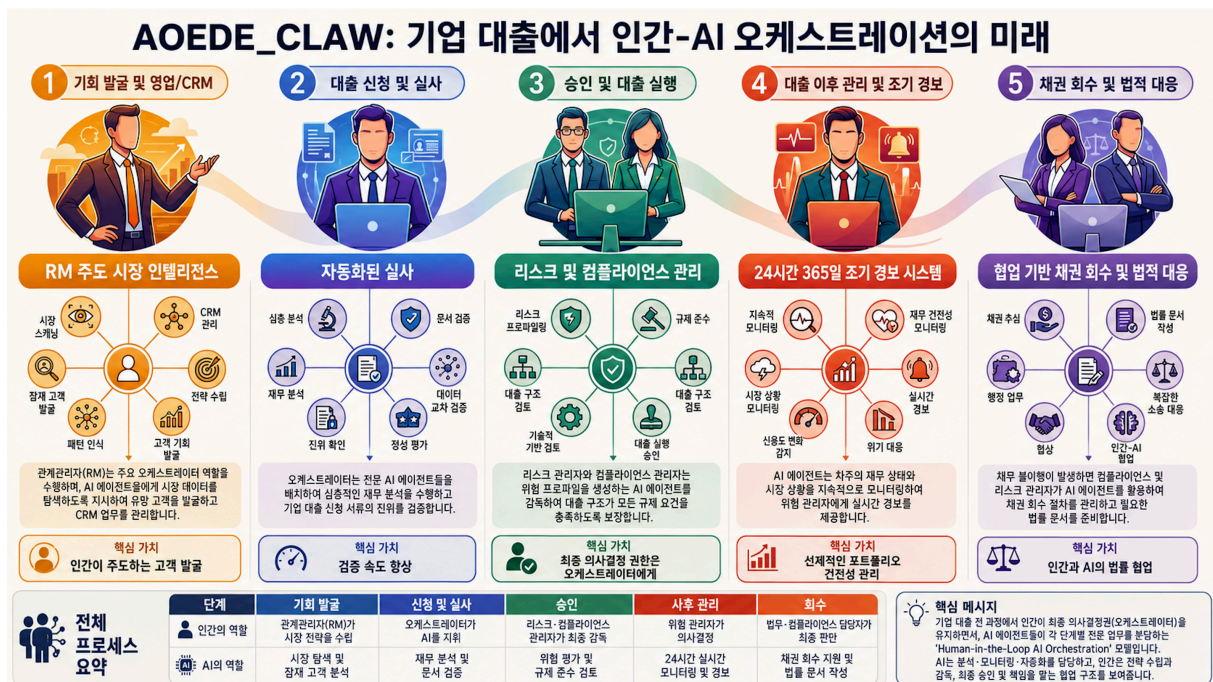
이 장에서 다루는 것: 발톱들(CLAWS)의 예시 구현 중 하나인 CLAW(기업 대출 증강 워크플레이스)의 전모. 기업 대출 업무의 전 생애를 담은 일터의 지도, 역할에 따라 달라지는 세 개의 대시보드, 곁에 앉은 동료로서의 에이전트, "분석 한 문장"의 위력, 그리고 대체가 아니라 증강을 택한 설계 철학과 그 성과 측정법.

5.1 들어가며: 첫 문서의 약속을 지키는 자리

제1장은 이렇게 썼습니다. "독수리의 구조는 결국 발톱이 무엇을 움켜질 수 있는가로 평가된다. AOEDE CLAWS는 프레임워크가 실제 비즈니스와 만나는 접점이다. 왼쪽 발톱은 전문 금융 서비스를 담당한다. 기업 대출, 무역 금융, 자금 관리처럼 복잡성이 높고, 근거가 계약 문서에 있으며, 통제되지 않은 의사결정의 비용이 자본 규모로 측정되는 영역이다."

제2장에서 제4장까지 우리는 머리(테오리아)와 두 날개(포이에시스, 프락시스)를 차례로 걸었습니다. 온톨로지가 정의되고, 하네스가 조립되고, 미메시스가 계보를 남기며 실연하는 것까지 보았습니다. 그러나 한 가지 질문이 남아 있었습니다. 그래서, 현업의 책상 위에서는 무엇이 달라지는가?

이번 장에서 그 질문에 답합니다. 주인공은 제1장에서 예고했던 바로 그 왼쪽 발톱, 기업 대출을 움켜쥐는 CLAWS의 예시 구현 — CLAW(Corporate Loan Augmenting Workplace, 기업 대출 증강 워크플레이스)입니다.



이름을 천천히 읽어 볼 가치가 있습니다. Corporate Loan은 도메인이고, Workplace는 형태입니다. 무게 중심은 가운데 단어, **Augmenting** — 증강에 있습니다. 제3장에서 하네스의 세 크기를 이야기하며 가장 작은 능력 하네스의 목표를 "자동화가 아니라 사람 일꾼의 능력 증강"이라 했고, "자율성은 선언하는

것이 아니라 작은 하네스에서부터 적립하는 것"이라며 도입의 사다리를 그렸습니다. CLAW는 그 사다리의 첫 단이 실물로 지어진 모습입니다. 이곳에서 AI는 심사역을 대체하지 않습니다. 심사역의 곁에 앉습니다.

한 가지를 미리 밝혀 둡니다. 이 애플리케이션은 CLAWS의 예시(레퍼런스) 구현입니다. 데이터는 시연용이고, 백여 개 화면 가운데 대표 화면들이 완전하게 구현되어 있고 나머지는 등록된 골격으로 존재합니다. 그러나 예시이기에 오히려 잘 보이는 것이 있습니다. AOEDE가 현업 시스템과 만날 때 취하는 형태, 즉 발톱의 패턴입니다. 이번 장은 그 패턴을 읽어 내는 데 집중하겠습니다.

5.2 일터의 지도: 기업 대출이라는 업무의 전 생애

CLAW를 열면 가장 먼저 만나는 것은 기업 대출 업무의 전 생애를 여덟 개 모듈로 펼친 지도입니다. 그 순서가 곧 여신(與信)의 일생입니다.

이야기는 **상담지원**에서 시작됩니다. 목표 고객을 검색하고, 기업의 종합 정보 — 기업 개요, 경영총과 주요 주주, 신용평가, 영업 현황, 관계회사, 기업 동향 — 를 조회하며, 재무 정보를 재무상태표·손익계산서·현금흐름표·재무비율 등 일곱 개 탭으로 들여다보고, 자행과 타행의 거래 이력을 확인하고, 업종 정보를 비교합니다. 그리고 마침내 **조건 시산과 신청** — 대출 조건을 시뮬레이션하고 신청서를 작성하는 화면 — 에 이릅니다.

이어서 **종합담보** 모듈이 담보의 평가와 등기와 해지를 맡고, **심사승인** 모듈이 여신 승인의 진행 상황과 기본 등록, 승인, 실행, 본부 관리를 담당합니다.

승인 이후는 관리의 시간입니다. **한도관리**는 Pre-NPL 관리, 신용여신 한도, 총익스포저(Total Exposure), 종합신용여신 한도, 대기업 영업 한도를 다룹니다. **여신자산건전성**은 자산 건전성의 등록과 분류, 충당금을 관리합니다. **사후관리**는 채권 재조정과 경매라는 여신의 가장 어려운 국면을 맡고, **여신심사** 모듈은 대출 점검 보고의 검증을 담당합니다. 마지막으로 **종합경영정보**가 각종 여신 실적 일보, 일연체 속보, 월별 연체율 현황, VINTAGE 분석, 금리 프리미엄 모니터링 같은 보고의 세계를 담습니다. 상담에서 경매까지 — 지도 한 장에 여신의 일생이 다 있습니다.

이 지도에서 관리자가 주목할 것은 규모와 등록의 방식입니다. 시스템에는 **백여든세 개의 화면이 하나의 레지스트리에 고유 ID와 함께 등록되어 있습니다**. 기업 대출이라는 업무의 관절 하나하나가 좌표를 갖고 있는 것입니다. 그리고 화면들은 열한 개의 **업무 도메인** — 기업고객관리, 상담지원, 신용평가, 재무분석, 심사승인, 종합담보, 한도관리, 조기경보, 여신자산건전성, 사후관리, 여신심사 — 으로도 묶여 있어, 사용자가 어느 화면으로 이동하든 시스템이 **지금 어느 업무 영역에 있는지를 자동으로 감지해 맥락을 맞춥니다**.

이 지도를 걷다 보면 한 가지 인상이 남습니다. 여덟 개 모듈의 순서가 곧 리스크의 일생이기도 하다는 점입니다. 상담과 시산의 단계에서 내려진 판단이 담보와 심사의 관문을 지나고, 승인 이후에는 한도와 건전성의 감시 아래 놓이며, 일이 어긋나면 사후관리의 어려운 국면으로, 잘 흘러가면 경영정보의 보고

서로 흘러듭니다. 증강 에이전트가 이 지도 위 어디에 앉든, 그 자리의 앞뒤에 무엇이 있는지를 시스템이 알고 있다는 것 — 그것이 맥락 감지의 실질적 의미입니다.

제2장의 문장을 기억하실 것입니다. "온톨로지 트리(제너시스)는 모든 작업의 좌표계다." CLAW의 화면 레지스트리와 도메인 지도는 그 좌표계가 현업의 일터에 내려앉은 모습입니다. **발톱은 아무 데나 움켜 쥐지 않습니다. 지형을 다 알고 움켜잡니다.**

5.3 세 사람의 아침: 역할이 다르면 같은 시스템도 다른 얼굴

CLAW는 세 사람의 하루를 가정하고 지어졌습니다. 오전 아홉 시, 같은 은행의 세 사람이 각자의 자리에서 같은 시스템에 로그인합니다. 지점의 고객 담당은 커피를 내려놓기도 전에 오늘 처리할 건들을 확인해야 하고, 본부의 심사역은 밤사이 쌓인 승인 대기 열의 길이가 궁금하며, 리스크 관리자는 어제의 연체 지표가 예정보선에 얼마나 다가섰는지부터 보고 싶습니다. 로그인하는 순간, 같은 시스템이 이 세 사람에게 세 개의 전혀 다른 얼굴로 나타납니다.

고객 담당(RM)의 아침은 오늘 해야 할 일의 풍경입니다. 대시보드의 중심은 **작업 큐**입니다. K전자의 B2B 한도 설정 신청, M기업의 신용평가 진행 건, L유통의 담보 평가 결과 도착, N건설의 현장 실사 준비가 줄지어 있습니다. 그 옆에는 **만기 긴급도** 패널이 초읽기에 들어간 일들을 세워 둡니다. A전자의 신용조사 만기 2일 전, B식품의 대출 만기 3일 전, C무역의 매입외환 만기 5일 전. 담당 고객의 목록, 건별 진행 상태, 은행 공지가 이를 둘러쌉니다. RM의 하루에서 가장 비싼 실수는 '놓침'입니다. 이 대시보드는 놓침의 후보들을 아침마다 눈앞에 도열시킵니다.

본부 심사역의 아침은 관문의 풍경입니다. 승인 대기 건들이 우선순위와 대기 일수를 달고 줄 서 있고 — M기업 신용평가는 벌써 5일째 대기 중입니다 — 심사·신용평가·담보평가·한도관리로 이어지는 본부 업무의 메뉴가 정렬되어 있습니다. 심사역의 하루에서 가장 비싼 실수는 '병목'이고, 이 화면은 병목을 숫자로 보여 줍니다.

리스크 관리자의 아침은 계기판의 풍경입니다. 대시보드의 첫 자리에 **NPL(부실채권)율 추이가 예정보선 2.0%와 함께** 그려져 있고, 자산 건전성 분류의 추이와 **VINTAGE 분석** — 대출 실행 시기별 묶음의 연체율 곡선 — 이 나란히 놓입니다. 실시간 보고, 포트폴리오 현황, 안정성, 수익성의 보고서 묶음이 그 아래를 받칩니다. 리스크 관리자의 하루에서 가장 비싼 실수는 '늦은 발견'이며, 이 화면은 발견의 시점을 앞당기기 위해 존재합니다.

세 대시보드가 공유하는 설계 원칙이 하나 있습니다. **역할이 곧 관점**이라는 것입니다. 이 시리즈 내내 역할은 하네스의 일급 요소였습니다. 테오리아의 역할 보강, 포이에시스의 역할 아키텍처, 프락시스의 역할 등록부가 그것입니다. CLAW에서 그 역할은 마침내 아침 화면의 생김새가 됩니다. 그리고 다음 절에서 보겠지만, 역할을 아는 것은 화면만이 아닙니다.

5.4 곁에 앉은 동료: AOEDE-Claw 에이전트

화면 오른쪽에는 접었다 펼 수 있는 패널이 하나 있습니다. **AOEDE-Claw AI 에이전트** — 이 워크플레이스에 상주하는 동료입니다. 시스템이 에이전트에게 부여한 자기소개가 설계 철학을 요약합니다.

"CLAW에 내장된(embedded) 지능형 AI **동료(co-worker)**." 도구도 아니고 챗봇도 아닌 '동료'라는 단어를 골랐고, 그 단어값을 하도록 두 가지가 설계되어 있습니다.

무엇보다 이 에이전트는 **사용자가 누구인지** 압니다. 지금 로그인한 사람의 역할과 작업 큐, 오늘 날짜가 에이전트의 맥락에 항상 들어가 있습니다. 그래서 RM에게 건네는 첫 인사와 리스크 관리자에게 건네는 첫 인사가 다릅니다. RM에게는 "오늘 만기 임박 건이 3건 있고 A전자의 신용조사가 가장 급합니다(2일 남음). 오늘 업무 계획을 우선순위로 잡아 드릴까요?"라고 말을 겁니다. 리스크 관리자에게는 "이번 달 연체율이 전월 대비 0.3%p 올라 예정보 임계값을 넘었습니다. 지금 바로 월간 리스크 분석 요약을 만들어 드릴까요?"라고 말합니다.

그리고 에이전트의 패널에는 **두 개의 탭**이 있습니다. 하나는 **협업 대화** — 묻고 답하는 익숙한 채팅입니다. 다른 하나가 이 설계의 핵심인데, 바로 **지능 통찰** 탭입니다. 묻지 않아도 에이전트가 먼저 꺼내 놓는 이야기들이 여기에 쌓입니다. RM에게는 만기 임박 경보와 5일째 지연 중인 평가 건, 그리고 "이번 달 데이터로 보아 B식품의 대출 현황 월간 보고서를 만들어 두면 좋겠다"는 제안이 놓입니다. 심사역에게는 대기 건의 우선순위 정렬과 함께 "BB+ 이상 등급의 표준 신청 3건이 감지되었으니 일괄 검토로 빠르게 처리할 수 있다"는 제안이 올라옵니다. 리스크 관리자에게는 연체율 임계값 초과 정보와 "2024년 3분기 실행 배치의 연체율이 역사적 평균을 웃돈다 — VINTAGE 이상 신호"라는 발견이 도착합니다.

이 목록을 다시 읽어 보면 공통점이 보입니다. 전부 **사람이 어차피 해야 했지만, 자주 늦거나 놓치던 종류의 일** — 훑어보기, 줄 세우기, 이상 검색 채기기 — 입니다. 그리고 하나 더 중요한 공통점이 있습니다. 에이전트의 모든 제안은 **확인 요청으로 끝납니다**. "생성을 시작할까요?", "일괄 처리를 확정할까요?", "

지금 분석할까요?" — 실행의 방아쇠는 언제나 사람의 손가락에 있습니다. 증강 워크플레이스의 통제 지점이란 거창한 것이 아닙니다. 바로 이 물음표들입니다.

5.5 한 문장의 마법: "분석 [기업명]"

CLAW의 증강이 가장 극적으로 드러나는 순간이 있습니다. 에이전트 대화창에 단 한 문장 — "분석 A전자" — 을 입력하는 순간입니다.

화면에는 네 단계의 진행이 차례로 표시됩니다. 기업 기본 정보를 가져오는 중, 재무 데이터를 분석하는 중, 시장 신호와 SNS 여론을 스캔하는 중, 종합 평가 보고서를 생성하는 중. 그리고 잠시 후, **기업 종합 분석 보고서**가 화면에 펼쳐집니다.



보고서의 첫 줄은 여섯 개의 핵심 지표입니다. 종합 점수, 신용등급과 전망, 예상 매출과 3년 누적 성장률, 영업이익률과 동업계 내 위치, 부채비율의 적정성, 그리고 SNS 긍정률과 동업계 평균 대비. 그 아래로 섹션들이 이어집니다. 3개년 재무 성과의 추이 차트. **경영층 평판 분석** — CEO, CFO, CTO 각각의 평판 점수와 시장의 평가 요지. 시장 동향과 **SNS 여론 분석** — 긍정·중립·부정의 여론 분포, 화제어 구름 (디지털 전환, ESG 경영, 원자재 비용, 규제 리스크 등), 최근 게시물의 감성과 주가 흐름. 그리고 마지막에 이 모든 것을 꿰는 **종합 분석 의견**이 놓입니다.

이 한 장면이 시리즈의 여러 문장을 동시에 실연하고 있다는 점을 짚어 두겠습니다. 제3장은 능력 하네스를 "포괄적 지식과 통제 지점의 최소 완결 단위"라 정의했습니다. '기업 종합 분석'은 정확히 그런 능력

중 하나입니다. 신용·재무·평판·시장·여론이라는 지식의 묶음이 정해진 절차에 따라 소집되고, 결과는 사람이 검토할 보고서의 형태로 멈춰 섭니다. 제4장은 미메시스를 "모든 결론에 계보가 있는 실연"이라 했습니다. 보고서의 단계별 수집 과정이 화면에 그대로 노출되는 것은, 그 계보 감각이 프런트 화면으로 번역된 모습입니다. 그리고 무엇보다, 이 능력을 호출하는 방법이 메뉴 일곱 번 클릭이 아니라 **자연어 한 문장**이라는 사실 — 상담 전화를 받으며, 품의서를 쓰다 말고, 회의에 들어가기 직전에도 부를 수 있는 동료의 형태라는 사실이 '증강'이라는 단어에 실감을 부여합니다.

RM의 시간 단위로 번역하면 이렇습니다. 예전 같으면 예닐곱 개 화면을 돌며 만나질 걸려 모으던 한 기업의 전모가, 고객과의 통화가 끝나기 전에 책상 위에 놓입니다. **판단은 여전히 RM의 몫입니다. 다만 판단의 재료가 준비되는 속도가 달라진 것입니다.**

5.6 증강의 문법: 왜 대체가 아니라 증강인가

여기서 한 걸음 물러나, CLAW가 취한 형태 자체를 읽어 보겠습니다. 이 시스템에는 눈에 띄는 특징이 하나 있습니다. **기존 워크플레이스의 문법을 거의 그대로 존중한다**는 점입니다. 조회 조건과 결과 테이블, 탭으로 나뉜 상세 화면, 입력 폼과 시산기 — 은행원이라면 눈 감고도 그릴 수 있는 화면 문법이 백여든 개 화면에 일관되게 흐르고, AI는 그 문법을 뒤엎는 대신 **오른쪽 패널 하나로** 조용히 들어와 앉았습니다.

이것은 기술의 한계가 아니라 의도된 설계이며, 그 이유는 세 겹으로 풀 수 있습니다.

가장 깊은 층에 놓인 것은 **신뢰의 경제학**입니다. 제3장의 도입 사다리를 떠올려 보십시오. 스킬에서 적립된 신뢰만이 태스크와 활동의 자율성을 정당화합니다. 기업 대출은 통제되지 않은 의사결정의 비용이 자본 규모로 측정되는 영역입니다. 이런 영역에서 첫날부터 판단을 기계에 넘기는 조직은 없고, 넘겨서도 안 됩니다. 증강은 그 신뢰가 적립되는 형태입니다. 에이전트의 요약이 정확했는지, 우선순위 제안이 옳았는지, 분석 보고서가 심사의 결론과 얼마나 일치했는지 — **매일의 업무가 곧 검증의 기록이 됩니다.**

그 위에 놓이는 것은 **사람의 자리에 대한 정직함**입니다. CLAW에서 승인의 도장, 일괄 처리의 확정, 보고서 생성의 개시는 전부 사람의 확인을 기다립니다. 이는 제3장과 제4장에서 본 HITL 체크포인트 원칙 — "자율성이란 사람이 사라진 상태가 아니라 사람의 자리가 정확히 설계된 상태" — 의 현업 버전이며, 규제 산업의 책임 구조와도 정합합니다. 감독 당국 앞에서 "그 여신은 누가 승인했습니까"라는 질문의 답은 언제나 사람의 이름이어야 합니다.

그리고 가장 바깥 층이 **변화 관리의 현실**입니다. 현업의 일터를 갈아엎는 도입은 교육 비용과 저항으로 무너지기 일쑤입니다. 익숙한 화면은 그대로 두고 곁에 동료 하나를 앉히는 도입은, 첫날부터 쓸 수 있고 첫 주부터 고마워집니다. 증강 워크플레이스는 기술 전략이기 이전에 **채택(adoption) 전략**입니다.

요컨대 CLAW의 문법은 이렇습니다. **화면은 그대로, 판단의 주인은 사람, AI는 요약과 우선순위와 초안과 경보를 맡는다.** 그리고 이 문법이 자리 잡은 뒤에야 — 어느 태스크에서 에이전트의 타율이 충분히 증명된 뒤에야 — 사다리의 다음 단, 태스크 하네스로의 승격이 논의됩니다.

5.7 발톱의 패턴: 이 예시가 보여 주는 것

CLAW는 예시 구현이지만, 예시가 보여 주는 패턴은 일반적입니다. AOEDE의 발톱이 어떤 현업 도메인을 움켜질 때 취하는 형태를, 이 애플리케이션에서 네 가지로 추출할 수 있습니다.

The screenshot displays the AOEDE-Claw AI platform interface. The main content area shows a report titled "내부 등급 모델 보조 보고서 · 한울정밀 주식회사" (Internal Rating Assist · Hanuljeongmil Co., Ltd.). The report includes a table of variables and their impact on the rating, with columns for variable name, value, and impact. The table shows that the rating is currently at BB+, and the impact of the variables is positive, leading to a higher rating of A.

변수	고액 값	업종 기준선 (반도체 장비 중소형)	이탈	영향
매출 (2025)	₩182억	₩95억	+91%	정방향 +0.18 등급
순이익률	11.8%	8.4%	+3.4pp	+0.10 등급
영업현금흐름 / 순이익	1.18x	0.92x	+28%	+0.08 등급
자산 회전을	1.42	1.15	+23%	+0.05 등급

The right sidebar shows a chat window with the text: "현재 업무 큐 분석 결과: A전자 신용 조사 만기까지 2일 남았습니다. 즉시 진행 권고드립니다. M기업 신용 평가가 5일 초과 상태입니다. 담당자에게 진행 상황 확인 부탁드립니다. K전자 B2B 한도 신청 처리 중, 내일 1차 심사 완료 예상. 오늘 A전자 신용 조사를 우선 처리하실 것을 권고드립니다. 조사 체크리스트 템플릿을 생성해 드릴까요?"

첫 번째 패턴은 **도메인의 전모를 먼저 등록하는 것**입니다. 백여든세 개 화면의 레지스트리와 열한 개 업무 도메인의 지도가 먼저 있고, 구현은 그 좌표 위에서 진행됩니다. 대표 화면 열두 개가 완전하게 지어지고 나머지가 골격으로 등록된 현재 상태는 미완성이 아니라 **MVO의 리듬 그 자체**입니다. 제1장의 표현으로, "목적지가 온톨로지를 끌어당겨 존재하게 만드는" 방식입니다. 다음에 지을 화면이 무엇인지는 지도가 이미 알고 있습니다.

두 번째 패턴은 **역할을 시스템의 일급 개념으로 삼는 것**입니다. 대시보드도, 에이전트의 인사도, 통찰의 내용도 역할을 따라 갈라집니다. 이는 시리즈 내내 쌓아 온 역할 아키텍처 — 테오리아의 역할 보강, 포이에시스의 역할 설계, 프락시스의 역할 등록 — 가 프런트 화면에서 결실을 맺는 지점입니다.

세 번째 패턴은 **에이전트를 화면이 아니라 맥락에 내장하는 것**입니다. AOEDE-Claw 에이전트는 특정 화면의 기능이 아니라 워크플레이스 전체에 상주하며, 사용자의 역할·작업 큐·현재 업무 도메인이라는 맥락을 상시 쥐고 있습니다. "분석 [기업명]" 같은 스킬 호출이 어느 화면에서든 가능한 이유입니다.

그리고 네 번째 패턴은 **뒤편의 독수리와 접속할 자리를 남기는 것**입니다. 이 예시의 에이전트는 실제 AI API로 동작하며(키가 없으면 시연 모드로 전환됩니다), 그 자리는 곧 시리즈가 걸어온 전체 스택 —

테오리아의 온톨로지가 근거를, 포이에시스의 하네스가 통제 지점을, 프락시스의 미메시스가 계보 있는 실행을 공급하는 — 이 꽃히는 소켓입니다. 오늘의 예시에서 통찰과 분석은 시연 데이터 위에서 움직이지만, 그 형태는 실전에서 넥서스로 배포된 능력 하네스가 채우게 될 형태와 같습니다. **발톱은 독수리의 끝이지, 독수리와 별개의 생물이 아닙니다.**

5.8 관리자의 눈: 증강의 성과를 무엇으로 잴 것인가

증강 워크플레이스의 도입을 검토하는 관리자에게, CLAW의 세 대시보드는 성과 지표의 힌트까지 함께 줍니다. 각 역할의 '가장 비싼 실수'를 뒤집으면 그것이 곧 측정 항목입니다.

RM 쪽에서는 **농침의 감소**입니다. 만기 누락 건수, 신용조사 기한 초과율, 그리고 고객 문의에서 종합 정보 회신까지의 소요 시간 — "분석 한 문장"이 단축시키는 바로 그 시간입니다.

심사 쪽에서는 **병목의 해소**입니다. 승인 대기 일수의 분포, 표준 건의 처리 리드타임, 그리고 일괄 검토 제안의 채택률과 그 정확도가 지표가 됩니다.

리스크 쪽에서는 **발견의 조기화**입니다. 임계값 초과에서 대응 착수까지의 시간, 그리고 VINTAGE 이상 신호가 얼마나 앞서 경보를 울렸는가(선행 일수)를 잹니다.

이 지표들이 실제로 쓰이는 장면을 상상해 보면 그 무게가 분명해집니다. 반년 뒤의 어느 승격 심사 자리, 안건은 "기업 종합 분석 능력을 태스크 하네스로 승격할 것인가"입니다. 테이블 위에 놓이는 것은 담당 임원의 소감이 아니라 기록입니다. 지난 반년간 이 능력의 제안이 몇 건 있었고 그중 몇 건이 채택되었는지, 채택된 분석이 이후 심사 결론과 얼마나 일치했는지, 인증 시험의 점수와 유효 기간은 어떠한지. 기록이 두껍다면 승격은 자연스럽고, 얇다면 심사는 미뤄집니다. 어느 쪽이든 결정의 근거가 사람의 확신이 아니라 축적된 데이터라는 점이 이 체계의 요체입니다.

그리고 하나가 더 있습니다. 앞 절의 논리를 따르면, 증강 단계의 가장 중요한 산출물은 효율이 아니라 **적립된 신뢰의 기록**입니다. 에이전트 제안의 채택률과 사후 정확도를 역할별·능력별로 쌓아 두십시오. 그 기록이 충분히 두꺼워진 능력이 태스크 하네스 승격의 후보이고, 제4장에서 본 인증·숙련도 체계가 그 승격을 공식화하는 절차가 됩니다. **증강의 대시보드는 그 자체로 자율화의 심사 서류인 셈입니다.**

5.9 맺으며: 독수리가 쥐는 법을 배우는 곳

다섯 장의 여정을 접겠습니다. **테오리아**에서 비즈니스가 언어가 되었고, **포이에시스**에서 언어가 세 크기의 하네스로 조립되었으며, **프락시스**에서 하네스가 넥서스로 배포되고 미메시스가 계보를 남기며 실행했습니다. 그리고 **CLAW**에서, 그 모든 것이 향하던 지면 — 한 사람의 RM이 아침에 여는 화면, 한 사람의 심사역이 마주하는 대기 열, 한 사람의 리스크 관리자가 지켜보는 예경보선 — 에 발톱이 닿았습니다.

이 발톱을 맡을 관리자에게 남길 당부는 결국 하나의 태도로 모입니다. **증강으로 시작하되, 증강에 머물**

계획은 세우지 마십시오. CLAW의 형태는 사다리의 첫 단이므로, 에이전트 제안의 채택률과 정확도를 도입 첫날부터 기록하여 어느 능력이 다음 단으로 올라갈 자격을 쌓고 있는지를 데이터로 알 수 있게 해야 합니다. 성과를 이야기할 때는 총론이 아니라 역할의 언어를 쓰시기 바랍니다. 놓침과 병목과 늦은 발견 — **역할별로 '가장 비싼 실수'를 정의하고 그것을 지표로 삼을 때 측정이 조직을 움직입니다.**

운영이 자리를 잡아 갈수록 지켜야 할 것은 **확인 요청의 문화**입니다. "확정할까요?"라는 물음표는 마찰이 아니라 통제 지점이며, 편의를 이유로 이 물음표를 걷어 내자는 요구는 반드시 나옵니다. 그 요구에는 해당 능력의 정식 승격 심사로 답하는 것이 옳지, 물음표의 삭제로 답할 일이 아닙니다. 아울러 **화면의 지도를 온톨로지와 이어 두시기 바랍니다.** 백여든세 개 화면의 레지스트리는 그 자체로 훌륭한 자산이지만, 이 화면이 어느 업무 대상, 어느 태스크의 표면인지가 테오리아의 좌표와 연결될 때 비로소 발톱이 몸통의 신경과 이어집니다.

그리고 시선을 조금 멀리 두십시오. 제1장은 왼쪽 발톱(전문 금융 서비스) 곁에 오른쪽 발톱 — 고객 관리, 고객자산 관리, 포괄적 리스크 관리 — 을 그려 두었습니다. CLAW에서 검증된 패턴, 곧 도메인 등록과 역할 일급과 맥락 내장 에이전트와 스택 소켓은 다음 도메인에서 그대로 재사용됩니다. **다음 발톱을 준비하십시오.** 발톱 하나의 성공은 발톱 하나의 성공으로 끝나지 않고, 움켜쥐는 법의 조직적 학습으로 남습니다.

제1장의 마지막 문장으로 이 여정을 닫는 것이 좋겠습니다. "엔진이 충실히 실행하게 하고, 발톱이 실제 업무를 움켜쥐게 하며, 독수리가 실제로 쥐는 법을 배운 만큼 온톨로지가 자라나게 하십시오." CLAW는 그 문장의 마지막 구절이 시작되는 곳입니다. 발톱이 쥐고, 전 경험이 기록되고, 그 기록이 머리로 올라가 온톨로지를 키우는 순환 — 독수리는 이제 나는 법만이 아니라, **쥐는 법을 배우는 중**입니다.

제5장 핵심 요약: CLAW는 기업 대출 도메인에 대한 CLAWS의 레퍼런스 구현으로, 183개 화면의 레지스트리와 11개 업무 도메인 지도 위에서 역할별(RM·심사역·리스크 관리자) 대시보드와 맥락 내장형 AI 동료를 제공한다. 설계의 무게 중심은 대체가 아니라 증강이며, 모든 에이전트 제안은 사람의 확인으로 실행된다. 증강 단계의 핵심 산출물은 효율이 아니라 적립된 신뢰의 기록이고, 그 기록이 태스크 하네스 승격의 심사 서류가 된다.

제 6 장. 통합 실행 가이드: 관리자의 책상 위에서

이 장에서 다루는 것: 다섯 편의 내러티브에 흩어져 있던 관리자 지침을 하나의 실행 체계로 통합한다. 도입 로드맵, 조직과 역할별 책임, 성과 지표 체계, 회의체 운영 수칙, 그리고 자주 나오는 질문에 대한 답변.

6.1 이 장의 성격

제1장에서 제5장까지는 각각 독수리의 한 기관을 걸으며 이야기를 풀었고, 각 장의 끝에는 그 기관을 맡을 관리자를 위한 당부가 다섯에서 여섯 가지씩 놓여 있었습니다. 이번 장에서는 그 당부들을 한 자리에 모아, **도입을 실제로 이끌어야 하는 관리자의 시간 순서**로 다시 배열한 것입니다. 새로운 내용을 더하기보다, 흩어진 것을 실행 가능한 형태로 묶는 것이 목적입니다.

6.2 도입 로드맵: 증강에서 자율로 가는 사다리

AOEDE의 도입은 기술 프로젝트의 일정표가 아니라 **신뢰가 적립되는 사다리**로 설계됩니다. 시리즈 전체를 관통한 이 원칙을 도입 단계로 옮기면 다음의 흐름이 됩니다.

준비 단계 — 첫 발톱과 첫 온톨로지를 고른다. 출발점은 "우리 회사의 전체 온톨로지"가 아니라 하나의 목적지, 즉 하나의 능력입니다. 복잡성이 높고, 근거가 문서에 있으며, 잘못된 판단의 비용이 큰 영역 — 그래서 계보와 통제가 절실한 영역 — 이 좋은 첫 후보입니다. 그 능력 하나가 신뢰할 수 있게 작동하는데 필요한 최소한의 엔티티, 속성, 관계, 거버넌스 파라미터를 정의합니다. 이것이 MVO(최소 실행 가능 온톨로지)의 출발입니다.

1단계 — 테오리아에서 언어화한다. 제너시스 패키지로 저장소를 부트스트랩하고, 해당 업무대상의 비즈니스 객체모델과 개념 모델을 다듬고, 보강 편집기로 지식·규칙·역할·정책을 채우고, BPMN·DMN으로 프로세스와 판단을 그립니다. 시맨틱 팩토리에 실제 규정과 계약 문서를 흘려 넣어 모델과 현실의 어긋남을 확인합니다. 이 단계의 완료 기준은 "Poiesis에 제출" 버튼을 누를 수 있는 상태"입니다.

2단계 — 포이에시스에서 하네스를 조립한다. 하네스의 크기(스킬·태스크·활동)를 먼저 합의하고, 시나리오에서 규격 있는 설계를 만들고, 의사결정 테이블의 커버리지·심각도 순서·금지 규칙을 버리고, 형식 검증·의미 검증·모의 실행의 세 겹 시험을 통과시킵니다. 이 단계의 완료 기준은 "비행 기록이 첨부된 배포 승인"입니다.

3단계 — 프락시스에서 다스리며 운영한다. 넥서스로 배포하고, 미메시스가 실연하는 작업의 계보를 표본 감사하고, 변경 통제·시그널·감독 규칙의 세 도구를 가동하며, 인증과 숙련도의 통장을 쌓습니다. 이 단계의 완료 기준은 따로 없습니다. 운영은 끝나지 않는 일이며, 대신 다음 단계로 가는 근거가 여기서 쌓입니다.

4단계 – CLAW형 증강으로 현업에 닿는다. 발톱의 네 가지 패턴 – 도메인 등록, 역할 일급, 맥락 내장 에이전트, 스택 소켓 – 을 따라 현업 워크플레이스에 에이전트를 동료로 앉힙니다. 화면은 그대로 두고, 판단의 주인은 사람으로 두고, 제안의 채택률과 정확도를 기록합니다.

5단계 – 사다리를 오른다. 채택률과 정확도의 기록이 두꺼워진 능력을 태스크 하네스 승격 후보로 올리고, 인증·숙련도 절차로 승격을 공식화합니다. 태스크들이 여물면 활동 수준의 자율 운영을 논의할 자격이 생깁니다. 동시에, 이 발톱에서 검증된 온톨로지와 패턴을 제너시스 패키지로 내보내 다음 도메인의 씨앗으로 씁니다.

이 로드맵을 달력 위에 올려 보면, 첫 90일의 그림은 대략 이렇게 그려집니다. 첫 달에는 첫 발톱이 될 능력 하나를 고르고, 그 능력의 골드 스탠더드 – 어떤 보고서가 나오면 성공인가 – 를 현업과 합의하며, 프로세스 역할에 페르소나 이름을 채웁니다. 둘째 달에는 테오리아에서 그 능력의 최소 온톨로지를 세우고 보강하며, 실제 규정과 계약 문서를 시맨틱 팩토리에 흘려 넣어 모델과 현실의 첫 대조를 받습니다. 셋째 달에는 포이에시스에서 하네스를 조립해 모의 실행의 비행 기록을 쌓고, 관문을 통과한 첫 배포를 프락시스의 무대에 올립니다. 석 달 뒤 경영진 보고에 올라가는 것은 거창한 전환 선언이 아니라, 실제로 켜진 등 하나 – 작동하는 능력, 그 계보, 그리고 다음 구역의 지도입니다.

이 로드맵에서 가장 흔한 유혹은 단계를 건너뛰는 것입니다. 특히 "이미 검증된 기술이니 처음부터 활동 수준의 자율 운영으로 가자"는 요구가 반드시 나옵니다. 시리즈의 답은 일관됩니다. **자율성은 선언하는 것이 아니라 적립하는 것이며, 크기의 오판 – 능력으로 충분한 곳에 활동 하네스를 짓거나, 활동이 필요한 곳에 능력만 쥐여 주는 것 – 이 이 영역에서 가장 비싼 실수입니다.**

6.3 역할과 책임: 새로 생기는 관리 업무의 주인 정하기

AOEDE의 도입은 새로운 종류의 관리 업무를 만들어 냅니다. 이 업무들의 주인이 정해지지 않으면, 도구는 있되 다스림은 없는 상태가 됩니다. 다섯 장의 당부에서 추출한 책임 목록은 다음과 같습니다.

온톨로지 소유자(테오리아). 온톨로지 트리와 개념 모델의 관리 권한을 갖습니다. 트리는 모든 작업의 좌표계이고 개념 모델은 그 좌표계의 문법이므로, 문법이 어지러우면 그 위의 모든 문장이 어지러워집니다. 시맨틱 팩토리의 모순 보고를 정기 안전으로 다루어, 모델과 문서의 어긋남을 부채가 아니라 온톨로지가 자라는 원료로 소화하는 것도 이 역할의 몫입니다.

캡슐·하네스 승인자(테오리아·포이에시스). 도메인 지식 캡슐과 능력캡슐의 라이프사이클(초안 → 검토 → 승인 → 게시), 그리고 하네스 파이프라인의 관문(접수됨 → 작성 중 → 검증됨 → 배포됨)을 지킵니다. 모의 실행 기록 없는 배포 승인 요청은 반려하는 것이 이 도구의 문화에 맞습니다.

리스크 정책 담당(포이에시스·프락시스). 의사결정 테이블의 보수성 원칙과 기본 결정의 작성, 금지 규칙의 규제 근거 보장을 도메인 전문가의 정식 업무로 부여합니다. 어떤 스킬과 추론 템플릿이 MRM(모델 리스크 관리) 대상인지를 리스크 부서와 함께 지정하며, 지정이 끝나면 변경 관리 대시보드의 대기 항목이 곧 리스크 부서의 업무 목록이 됩니다.

운영 감독자(프락시스). 작업 산출물의 계보를 표본 감사하고, 인증 만료를 월례 점검하며, 약속 위반('위반됨' 상태)의 빈도와 패턴을 하네스 이음매의 건강 지표로 추적합니다. 스로틀 회로 — 신호의 가중치와 관문 연결 — 의 주기적 조정도 이 역할이 주관하되, 정책적 판단은 여신 정책 협의체와 함께 합니다.

현업 채택 리더(CLAW). 역할별 '가장 비싼 실수'를 지표로 정의하고, 에이전트 제안의 채택률·정확도 기록을 관리하며, 확인 요청 문화를 지킵니다. 연합(Federation)에 내놓을 능력과 빌릴 능력의 정책을 미리 정하는 대외 거버넌스 질문도, 기술이 성숙한 뒤로 미루지 말고 지금 이 역할과 경영진이 함께 다뤄야 합니다.

한 사람이 여러 역할을 겸할 수는 있습니다. 그러나 **어느 역할도 공석이어서는 안 됩니다.** 특히 초기에 가장 자주 공석이 되는 자리가 온톨로지 소유자입니다. "모델은 컨설턴트가 만들어 주고 갔다"는 상태로 운영에 들어간 조직에서, 온톨로지는 반년 안에 낡은 문서가 됩니다.

6.4 성과 지표 체계: 무엇을 재고, 무엇을 보고할 것인가

다섯 장에서 언급된 측정 항목을 세 층으로 정리하면 다음과 같습니다.

첫째 층 — 현업 효과 지표. 역할별 '가장 비싼 실수'를 뒤집은 지표들입니다. 농침의 감소(만기 누락 건수, 기한 초과율, 종합 정보 회신 소요 시간), 병목의 해소(승인 대기 일수 분포, 표준 건 처리 리드타임), 발견의 조기화(임계값 초과에서 대응 착수까지의 시간, 이상 신호의 선행 일수). 이 층은 경영진 보고의 언어입니다.

둘째 층 — 신뢰 적립 지표. 자율성 승격의 심사 서류가 되는 지표들입니다. 에이전트 제안의 채택률과 사후 정확도(역할별·스킬별), 인증 시험의 점수와 유효 상태, 모의 실행의 커버리지와 배치 통계, 약속 위반의 빈도와 패턴. 이 층은 "다음 단계로 가도 되는가"라는 질문의 근거입니다.

셋째 층 — 거버넌스 건강 지표. 시스템 자체의 규율이 살아 있는지를 보는 지표들입니다. 시맨틱 팩토리 모순 보고의 발생과 처리 속도, 검증 경고 백로그(예: 규제 조항 연결이 미비한 금지 규칙)의 소진 추이, 변경 관리 대기 항목의 체류 시간, 인증 만료 방지 건수, 능력 보고서에서 "좁혀서 답했다·답하지 못했다"로 드러난 온톨로지의 빈 구역. 이 층은 운영 회의의 언어입니다.

세 층의 지표는 보고의 흐름으로도 이어집니다. 현업 효과 지표는 분기 경영진 보고의 첫 장에 오르고, 신뢰 적립 지표는 승격 심사의 서류가 되며, 거버넌스 건강 지표는 월간 운영 회의의 체크리스트가 됩니다. 같은 데이터가 세 개의 다른 회의에서 세 개의 다른 질문 — 효과가 있는가, 더 맡겨도 되는가, 규율이 살아 있는가 — 에 답하는 구조입니다. 지표를 새로 발명할 필요는 없습니다. 시스템이 운영 중에 자연스럽게 남기는 기록을 세 갈래로 읽는 습관이면 충분합니다.

지표 체계에서 시리즈가 특히 강조한 문화가 하나 있습니다. **이해 보고서를 읽는 문화**입니다. 시스템이 "이 부분은 좁혀서 답했고, 이 부분은 답하지 못했다"고 정직하게 말할 때, 그 말을 흘려듣지 않는 팀만이 정직한 시스템의 값어치를 온전히 누립니다. 정직한 실패 보고는 벌점의 재료가 아니라, 다음에 지을 온

톨로지 구역을 알려 주는 지도입니다.

6.5 회의체와 운영 수칙: 다섯 번째 장의 당부를 캘린더에 올리기

당부는 캘린더에 올라가야 실행됩니다. 다섯 장의 지침을 회의체 관점으로 재배열하면 다음의 정례 안건이 됩니다.

주간 — 파이프라인 리뷰(포이에시스). 진척 보고의 언어는 다섯 상태 — 접수됨, 작성 중, 검증됨, 배포됨, 반려됨 — 로 통일합니다. 몇 주째 '작성 중'에 머무는 하네스, '검증됨'과 '배포됨' 사이의 병목이 이 회의의 관심사입니다.

월간 — 운영·거버넌스 회의(프락시스). 만료 임박 인증의 점점을 고정 안건으로 두고, 변경 관리 대기 항목, 약속 위반 패턴, 활성 알림의 추이를 함께 봅니다. 중요 작업의 계보 표본 감사 결과도 이 자리에서 보고됩니다.

월간 또는 격월 — 온톨로지 협의체(테오리아). 시맨틱 팩토리의 모순 보고를 정기 안건으로 삼아, 문서와 모델의 어긋남을 조율합니다. MVO로 자라는 온톨로지들이 겹치기 시작하는 지점 — 예컨대 한 능력의 '고객'과 다른 능력의 '차주'가 만나는 순간 — 의 정리도 여기서 다룹니다. 하네스는 발견까지만 해 줍니다. 조율은 사람의 몫입니다.

분기 — 정책 레버 점검(포이에시스·프락시스). 스로틀 회로의 신호 가중치와 관문 연결을 성과 피드백과 함께 재검토하고, 금지 규칙의 규제 근거 보강 백로그의 소진을 확인합니다. 검증이 남긴 경고 목록이 곧 이 분기의 업무 목록입니다.

분기 또는 반기 — 승격 심사(전사). 신뢰 적립 지표가 충분히 쌓인 능력의 태스크 승격, 태스크의 활동 승격을 심사합니다. 심사의 근거는 담당 임원의 확신이 아니라 인증 기록과 채택률·정확도의 축적이어야 합니다. 확인 물음표를 걸어 내자는 현업의 요구도 이 심사를 통해서만 수용합니다.

6.6 자주 나오는 질문에 대한 답

도입 논의에서 반복적으로 등장하는 질문들에, 시리즈의 논리로 답을 정리해 둡니다.

"온톨로지 정비에만 몇 년이 걸리는 것 아닌가?" 그 걱정이 정확히 MVO가 겨냥한 회피 유형입니다. AOEDE는 전사 온톨로지를 먼저 완성하는 접근을 명시적으로 거부합니다. 첫 능력 하나에 필요한 최소한만 정의하고, 배포 하나가 성공할 때마다 온톨로지를 한 구역씩 넓힙니다. 전사 온톨로지는 목표가 아니라, 발톱이 쌓이면서 자연스럽게 조립되는 결과물입니다. 이러한 MVO 접근 방법의 문제를 보완하기 위해 그림자 온톨로지를 도입하였으며, 첫 번째 능력 온톨로지 적용에서부터 그림자 온톨로지가 양지로 나오도록 하였습니다.

"AI가 지어낸 숫자로 보고서가 나가면 어떻게 하나?" 구조적으로 차단되어 있습니다. 숫자를 만드는 것

은 LLM이 아니라 결정론적 컴퓨트 엔진이고, 자연어 질의는 삼단(A단 언어 이해 — B단 검증·컴파일 — C단 실행·검증) 하네스를 통과하며, 스키마에 결속되지 않는 요소는 실행 전에 탈락합니다. 시맨틱 팩토리의 추출에도 충실성 게이트가 있어 원문 근거가 없는 항목은 저장 전에 기각됩니다. 핵심은 이 차단이 "AI가 착하게 행동하는 것"에 의존하지 않고 아키텍처로 강제된다는 점입니다.

"결국 사람이 다 확인해야 한다면 효율이 나는가?" 증강 단계의 효율은 판단의 자동화가 아니라 판단 재료의 준비 속도에서 나옵니다. 반나절 걸리던 기업 전모의 수집이 통화가 끝나기 전에 놓이는 것, 농침·병목·늦은 발견이 지표로 줄어드는 것이 그 효율입니다. 그리고 확인의 기록 자체가 다음 단계 자율화의 근거가 되므로, 사람의 확인은 비용이 아니라 투자입니다.

"감사와 감독 당국 대응은 어떻게 되는가?" 세 가지 질문에 화면으로 답할 수 있는 상태가 목표이자 설계입니다. "그 숫자 어디서 났습니까"에는 액션 단위 입출력의 계보로, "그 규칙의 법적 근거는 무엇입니까"에는 규칙-규제 조항의 구조화된 연결로, "그 행위를 할 권한이 있었습니까"에는 권한 매트릭스와 감사 기록으로 답합니다. 그리고 최종 승인의 이름은 언제나 사람의 이름입니다.

"우리 데이터 품질로 가능한가?" 완벽한 데이터를 전제하지 않는 것이 이 프레임워크의 특징입니다. 시맨틱 팩토리는 모델과 현실 문서의 어긋남을 찾아내라고 있는 장치이고, 출처 귀속과 불확실성 프로파일은 값이 얼마나 믿을 만한지를 값 단위로 기록하며, 목적지가 요구하는 값이 아직 없을 때 온톨로지는 날조되는 대신 승인자가 채울 슬롯과 결정론적 폴백으로 자랍니다. 요컨대 데이터의 결함은 도입을 막는 벽이 아니라, 시스템이 드러내고 사람이 갯아 가는 관리 대상입니다. 오히려 첫 발톱의 범위를 정할 때 데이터가 비교적 온전한 영역을 고르는 것이 현실적인 요령입니다.

"배포된 하네스에 문제가 생기면 어떻게 되나?" 되돌아오는 문이 항상 열려 있습니다. 모든 하네스는 브랜치와 커밋으로 이력이 관리되어 과거 판본으로 롤백할 수 있고, 배포 후의 부품 수정은 변경 통제 수준에 따라 다시 관문을 지나야 하며, 시그널과 알림이 이상을 심각도별로 집계합니다. 약속의 고리 덕분에 어디서부터 어긋났는지는 사후 조사가 아니라 장부 조회의 문제이고, 계보 덕분에 문제가 된 결론의 원인 액션까지 거슬러 오를 수 있습니다. 나가는 문은 좁고 돌아오는 문은 열려 있는 구조 — 배포 통제 정석이 그대로 구현되어 있습니다.

"기존 시스템을 갈아엎어야 하나?" CLAW의 패턴이 답합니다. 익숙한 화면 문법은 그대로 두고, AI는 오른쪽 패널 하나로 들어와 앉습니다. 증강 워크플레이스는 기술 전략이기 이전에 채택 전략이며, 갈아엎는 도입이 교육 비용과 저항으로 무너지는 현실을 정면으로 반영한 설계입니다. CLAW는 기존 시스템을 그대로 두고 기업의 온톨로지를 활용해 기존 직원들의 능력을 증강하도록 설계되었습니다. 여기서 AI 에이전트는 능력 있는 조력자로서 화면의 한쪽에 앉아 인간 작업자의 능력을 증폭시킵니다.

6.7 맺으며: 날마다의 비행

이 문서의 처음에 놓였던 질문으로 돌아가겠습니다. 스스로 판단하고 행동하는 소프트웨어에게, 행동할 자유와 신뢰할 수 있는 규율을 어떻게 동시에 부여할 것인가.

다섯 장을 지나온 지금, 답은 하나의 구조로 요약됩니다. 볼 수 있을 만큼만 온톨로지를 정의하고(테오리아, MVO), 행동할 수 있을 만큼만 에이전트를 만들며(포이에시스, 세 크기의 하네스), 모든 행동을 통제와 거버넌스에 매어 두고(넥서스, 프락시스), 엔진이 충실히 실행하게 하며(미메시스, 계보), 발톱이 실제 업무를 옮겨주게 하고(CLAWS, 증강), 실제로 쥐는 법을 배운 만큼 온톨로지가 자라나게 한다(순환).

그리고 이 구조는 한 번의 결단으로 완성되는 것이 아니라, 날마다의 운영으로 유지됩니다. 테오리아에서 정의하고, 포이에시스에서 조립하고, 프락시스에서 다스리고, 발톱에서 기록하는 하루하루 — 그것이 독수리의 비행입니다.

마지막으로 한 가지만 덧붙이겠습니다. 이 문서가 길게 이야기해 온 것은 결국 기술이 아니라 조직의 습관입니다. 모순 보고를 안전으로 올리는 습관, 비행 기록 없는 승인을 반려하는 습관, 물음표를 지키는 습관, 정직한 실패 보고를 지도로 읽는 습관. 도구는 이 습관들이 자리 잡을 칸을 마련해 줄 뿐이고, 칸을 채우는 것은 언제나 사람입니다. 에이전틱 AI의 시대에 조직이 갖추어야 할 경쟁력이 있다면, 그것은 가장 똑똑한 모델을 고르는 안목이 아니라 이런 습관을 꾸준히 지켜 내는 근력일 것입니다. 이 문서가 그 근력을 기르는 데 쓸모 있는 교범이 되기를 바랍니다.

제6장 핵심 요약: 도입은 준비(첫 발톱 선정) → 언어화(테오리아) → 조립(포이에시스) → 운영(프락시스) → 증강(CLAWS) → 승격의 사다리를 따른다. 온톨로지 소유자, 승인자, 리스크 정책 담당, 운영 감독자, 채택 리더의 다섯 역할에 공석이 없어야 하며, 성과는 현업 효과·신뢰 적립·거버넌스 건강의 세 층으로 측정한다. 당부는 캘린더에 올라가야 실행되고, 자율성의 확대는 언제나 기록의 축적 위에서 심사된다.

부록 A. 용어집

이 부록은 본문에 등장하는 핵심 용어를 관리자의 언어로 풀이한 것입니다. 정식 정의가 아니라 이해를 돕기 위한 실무적 설명이며, 본문에서 처음 등장하는 맥락과 함께 읽으면 좋습니다.

AOEDE (Agentic Ontology Envisioning and Deployment Environment) 에이전틱 AI를 기업에 도입하기 위한 전체 프레임워크. 온톨로지 정의(테오리아), 하네스 조립(포이에시스), 운영 통제(프락시스), 현업 적용(CLAWS)을 하나의 일관된 구조로 묶으며, 전체 아키텍처를 날개를 펼친 독수리에 대응시킨다.

에이전틱 AI (Agentic AI) 정해진 절차만 반복하는 자동화와 달리, 목표를 받아 스스로 상황을 판단하고 행동을 선택하는 AI. 자유와 규율을 동시에 설계해야 하는 대상이며, 이 문서 전체의 주제다.

온톨로지 (Ontology) 우리 업무 영역에 어떤 대상이 존재하고, 서로 어떻게 연결되며, '의사결정'과 '근거'가 무엇을 뜻하는지를 컴퓨터가 이해할 수 있는 구조로 적어 놓은 것. 비즈니스의 공식 지도이자 공식 사전.

테오리아 (THEORIA) 독수리의 머리. 온톨로지가 정의·보강·검증되는 곳이며, 프레임워크 전체의 판단이 시작되는 개념적 중심. 웹 애플리케이션(테오리아 클라이언트)과 AOEDE 온톨로지 서버의 두 층으로 구현된다.

포이에시스 (POIESIS) 독수리의 왼쪽 날개, '만드는 날개'. 온톨로지로 정의된 업무를 실행 가능한 에이전트 하네스(활동·태스크·스킬의 세 크기)로 조립하는 곳. 대상 사용자는 개발자가 아니라 비즈니스 분석가와 업무 설계자다.

프락시스 (PRAXIS) 독수리의 오른쪽 날개, '다스리는 날개'. 배포된 하네스 안에서 에이전트가 실제로 일하는 무대이며, 실행의 모니터링·변경 통제·인증·감독 규칙을 담당한다.

넥서스 (NEXUS) 하네스가 기록되는 규격이자, 만드는 날개와 다스리는 날개가 주고받는 계약 문서의 형식. CC-DSL 전용 언어로 기술되며, 97종의 정의 요소와 28종의 연결 관계로 에이전트 조직 하나를 빠짐없이 기술한다. 하나의 원본에서 요구사항·프로세스·데이터 흐름·거버넌스 등 일곱 개의 관점이 기계적으로 파생된다.

미메시스 엔진 (MIMESIS ENGINE) 넥서스를 실행하는 결정론적 실행 드라이버. '충실한 재현'이라는 이름 그대로 결과를 지어내지 않으며, 모든 액션의 입력과 출력을 쌍으로 보존해 모든 결론에 내려받을 수 있는 계보를 남긴다.

CLAWS / CLAW CLAWS는 프레임워크가 실제 비즈니스와 만나는 접점(발톱)의 총칭.

CLAW(Corporate Loan Augmenting Workplace)는 그중 기업 대출 도메인의 예시 구현으로, '증강 워크플레이스'의 패턴을 보여 준다.

하네스 (Harness) AI라는 지능에 채우는 마구(馬具). 무엇을 알아야 하고, 무엇을 할 수 있고, 어디까지 가도 되며, 어디서 반드시 멈추거나 사람에게 넘기는지가 짜여 들어간 구조물. AOEDE에서 '에이전트'

를 만든다'는 것은 곧 '하네스를 짠다'는 뜻이다.

하네스의 세 크기 (활동·태스크·능력) 활동 하네스는 업무 흐름 전체의 자율 운영을, 태스크 하네스는 한 역할의 특정 태스크 대행을, 능력 하네스는 사람의 능력 증강을 목표로 한다. 셋은 포개지는 구조이며, 도입은 능력 → 태스크 → 활동의 사다리를 오른다.

MVO (Minimum Viable Ontology, 최소 실행 가능 온톨로지) "전체 온톨로지를 먼저 완성"하는 접근을 거부하고, 지금 필요한 능력 하나를 신뢰할 수 있게 만드는 데 필요한 최소한만 정의하는 방법론. 배포가 성공할 때마다 온톨로지가 한 구역씩 자란다.

능력캡슐 (Capability Capsule) 거버넌스가 내장된, 그 자체로 완결된 능력 단위. 골드 스탠더드 리포트, 스킬, 추론, 엔티티 모델과 데이터 계약, 솔루션 트리, 결정론적 컴퓨트 엔진, 시그널과 갭 감지기, 회귀 테스트 픽처, 그리고 사람의 판단을 위한 명시적 슬롯을 담는다.

메타 온톨로지 / 회사 온톨로지 온톨로지의 두 층. 메타 온톨로지는 '#'가 붙는 메타 객체들의 세계로, 회사에 대한 지식을 적는 방법에 대한 문법이다. 회사 온톨로지는 그 문법으로 적힌 우리 회사의 실물 모델이다. 두 평면을 섞는 것이 오답의 전형적 원인이다.

제너시스 (Genesis) 두 가지 뜻을 가진 이름. 하나는 개념 편집기의 이름, 다른 하나는 업무대상 정의와 비즈니스 객체모델 다이어그램을 담은 이식 가능한 패키지 형식. 검증된 온톨로지를 내보내고 새 저장소를 부트스트랩하는 씨앗이자 운반 그릇이다.

시맨틱 팩토리 (Semantic Factory) 위키에 첨부된 문서(규정집, 계약서, 매뉴얼)에서 엔티티·사실·관계를 추출해 축적하는 공장. 모든 사실에 출처가 달리고, 원문 근거가 없는 항목은 충실성 게이트에서 기각되며, 문서와 온톨로지의 모순을 감지해 사람의 검토 안전으로 만든다.

보강 (Enrichment) 온톨로지 트리라는 뼈대에 살을 붙이는 작업. 업무 대상의 속성과 의미, 도메인 지식, 결정 규칙, 역할·스킬·도구·거버넌스 정책을 구조화 데이터로 채운다. AI 에이전트라는 새 '직원'의 직무기술서를 쓰는 일에 해당한다.

CVR (고객 가치 실현) 편집기 고객 여정·과업 정의·아이데이션·가치 흐름 설계·에이전트 기획·QFD·실현 측정의 일곱 단계로 고객 활동을 다루는 편집기. 디자인 씽킹의 언어를 쓰되 결과물이 온톨로지와 에이전트 포트폴리오로 직결된다.

모델 하네스 / 실행 하네스 테오리아를 지탱하는 두 하네스. 모델 하네스는 온톨로지 모델 자체의 정합성(문법, 추론기, 경계와 이력, 모순 감지)을 지켜 상향식 성장이 난장판이 되는 것을 막는다. 실행 하네스는 AI의 자연어 질의 실행을 삼단 구조로 다스려 시스템이 문제 생성기가 되는 것을 막는다.

삼단 실행 하네스 (A단·B단·C단) A단(언어 이해)에서 LLM이 자연어를 형식화된 질의 계획으로 번역하되 그 출력은 신뢰되지 않는다. B단(검증과 컴파일)에서 결정론적 코드가 계획의 모든 요소를 살아있는 스키마에 결속하고, 결속되지 않는 요소는 탈락시키며, 읽기 전용·상한이 걸린 질의만 생성한다. C단(실행과 검증)에서 비평가가 결과를 검사하고 유한 횟수 안에서 재시도한다.

이해 보고서 (Comprehension Report) 질문 가운데 무엇이 그대로 수행되었고, 무엇이 좁혀졌으며,

무엇이 왜 탈락했는지를 명시하는 산출물. 시스템이 못 하는 것을 조용히 빼고 답하는 대신 못 했다고 말하게 하는 장치.

디지털 트윈 (Digital Twin) 비즈니스의 구조와 지식과 상태를 컴퓨터가 이해하는 형태로 비추어, 질문하면 근거를 갖고 답하고 변경하면 파장을 짚어 주는 살아 있는 대응물. 한 번의 대공사가 아니라 여섯 단계(씨앗 심기 → 정의 평면 → 살 붙이기 → 증거 연결 → 그래프로 깨어나기 → 번식)로 이어진다.

BPMN / DMN 업무 프로세스(BPMN)와 의사결정 테이블(DMN)의 국제 표준 표기법. 테오리아에서 그리는 이 다이어그램들은 참고 자료가 아니라 에이전트가 실제로 따르게 될 규격이다.

의사결정 테이블의 세 가지 규율 커버리지(모든 입력 조합에 판단이 존재함 — 기본 결정 포함), 심각도 순서(여러 규칙이 경합하면 가장 보수적인 결정부터), 금지 규칙("이 조건에서는 이 결정이 절대 나올 수 없다"는 규칙의 기계적 강제). 주택담보대출 예시에서 1,410개 입력 조합의 전수 분석으로 다치 사례 97건과 미포괄 사례 897건이 드러났고, 금지 규칙 194개가 도출되었다.

스로틀 (Throttle) 자율 운영되는 활동 전체의 보수성과 공격성을 조절하는 정책 레버. 시장·연체·운영 신호가 가중 합산되어 스로틀 점수가 되고, 각 관문의 승인 임계값과 보수성 편향으로 번역되며, 성과가 다시 신호로 돌아오는 닫힌 회로다.

HITL (Human-in-the-Loop) 체크포인트 사람 개입 지점. 활동 하네스의 자율성이란 사람이 사라진 상태가 아니라 사람의 자리가 정확히 설계된 상태이며, HITL 체크포인트는 그 자리를 일급 설계 요소로 선언한 것이다.

약속의 고리 (요청됨 → 약속됨 → 진술됨 → 수락됨) 언어행위 이론에 뿌리를 둔 트랜잭션의 네 위상. 모든 일에 요청한 자와 약속한 자가 있고, 약속에는 활성·이행됨·위반됨·취소됨의 상태가 있어, 책임의 추적이 사후 조사가 아니라 자료 구조가 된다.

에이전톨로지 (Agentology) 에이전트 능력의 여덟 갈래 분류 — 인지, 지식, 메모리, 추론, 상호작용, 학습, 페르소나, 프레젠테이션 — 와 입출력 양쪽의 일곱 겹 안전 필터. 등록되지 않은 능력은 하네스 안에 존재하지 않는다는 원칙의 운영 측 구현.

능력(Capability)과 역량(Competency) 능력은 비즈니스 계층 — 비즈니스 가치를 생성 하는 일("담보 평가를 한다"). 역량은 에이전톨로지 계층 — 에이전트의 인지 능력("문서에서 개체를 추출할 줄 안다"). 두 층을 구분하는 어휘.

MRM (Model Risk Management, 모델 리스크 관리) 규제가 요구하는 모델 검증 절차. 프락시스의 변경 통제 수준(자유 변경 — 검토 필요 — MRM 필요) 가운데 가장 무거운 단계이며, 지정 대상의 결정은 기술 판단이 아니라 규제 대응의 판단이다.

인증과 숙련도 에이전트 스킬 숙련도의 네 단계(미훈련·초보·숙달·전문가)와, 테스트 스위트 응시로 증명되고 만료일이 있는 인증 체계. 자율성 확대가 담당 임원의 확신이 아니라 인증 기록의 축적 위에서 결정되게 하는 신뢰의 통장.

연합 (Federation) / A2A 조직 담장 밖의 에이전트들과 표준 프로토콜(A2A, 에이전트 대 에이전트)로 연결되는 체계. 다른 조직의 에이전트가 엔드포인트와 함께 등록되고, 우리 하네스가 갖지 못한 능력을 조건에 따라 호출할 수 있다.

RM (Relationship Manager) 기업 고객을 담당하는 영업·관계관리 담당자. CLAW의 세 페르소나 (RM, 본부 심사역, 리스크 관리자) 중 하나.

NPL (Non-Performing Loan) / VINTAGE 분석 NPL은 부실채권. VINTAGE 분석은 대출을 실행 시 기별 묶음으로 나누어 각 묶음의 연체율 곡선을 추적하는 기법으로, 특정 시기 실행분의 이상 신호를 조기에 드러낸다.

증강 워크플레이스 (Augmenting Workplace) 기존 화면 문법을 존중한 채 AI를 맥락 내장형 동료로 앉히는 도입 형태. 화면은 그대로, 판단의 주인은 사람, AI는 요약·우선순위·초안·경보를 맡으며, 모든 제안은 사람의 확인으로 실행된다.

부록 B. 다섯 장을 관통하는 문장들

본문 곳곳에 놓였던 핵심 문장들을 한 자리에 모으고, 각 문장이 어느 맥락에서 나왔으며 관리자에게 무엇을 요구하는지를 짧게 되짚습니다. 회의 자료나 교육 자료에 인용할 때, 이 부록이 색인의 역할을 할 수 있을 것입니다.

"온톨로지 없이 움직이는 에이전트는 자율적인 것이 아니라, 단지 책임을 물을 수 없는 존재일 뿐이다." 제1장, 머리에 관한 문장입니다. 에이전트 AI의 도입 순서를 정하는 문장이기도 합니다. 이 문장이 요구하는 것은 완벽한 전사 모델이 아니라, 에이전트가 딛고 설 만큼의 언어 — 그 업무 영역의 대상과 관계와 판단 기준의 정의 — 가 자동화보다 먼저 존재해야 한다는 순서의 규율입니다.

"통제 없이 만들기만 하는 조직은 리스크를 대량생산하고, 만들지 않고 통제만 하는 조직은 아무것도 생산하지 못한다." 제1장, 두 날개에 관한 문장입니다. 도입 논의가 속도파와 신중파로 갈라질 때 양쪽 모두에게 건넬 수 있는 문장이며, AOEDE가 이 대칭을 구호가 아니라 같은 서버와 같은 데이터베이스를 공유하는 아키텍처로 구현했다는 사실이 문장의 담보가 됩니다.

"목적지가 온톨로지를 끌어당겨 존재하게 만든다." 제1장, MVO에 관한 문장입니다. "우리 비즈니스의 완전한 온톨로지는 무엇인가"라는 질문을 "이 하나의 능력을 신뢰할 수 있게 만드는 데 필요한 최소한의 온톨로지는 무엇인가"로 바꾸는 순간, 수년짜리 모델링 대공사는 분기 단위로 등을 켜 가는 건설로 바뀝니다.

"정확한 답과 그럴듯해 보이는 오답의 차이는 층위의 규율에서 나온다." 제2장, 두 층의 온톨로지에 관한 문장입니다. 모델에 대한 정의적 질문과 데이터에 대한 인스턴스 질문을 섞는 것이 오답의 전형적 원인이며, 시스템이 이 두 평면을 구조적으로 구분한다는 사실이 답변 신뢰의 바탕입니다.

"모델 하네스 없는 상향식은 빠르게 시작해서 빠르게 수렁에 빠지고, 모델 하네스 있는 상향식이 비로소 MVO라는 이름값을 한다." 제2장, 첫 번째 하네스에 관한 문장입니다. 작게 시작하자는 합의만으로는 부족하고, 작게 자라는 조각들을 문법과 추론기와 이력과 모순 감지로 다스리는 손이 함께 있어야 한다는 조건을 붙입니다.

"고삐 없는 목표 지향 에이전트는 목표를 향해 성실하게 사고를 친다." 제2장, 두 번째 하네스에 관한 문장입니다. 에이전트 리스크의 본질이 악의가 아니라 통제되지 않은 성실함이라는 통찰이며, 그래서 해법도 훈계가 아니라 구조 — 신뢰 경계를 사이에 둔 삼단의 실행 하네스 — 여야 한다는 결론으로 이어집니다.

"시스템은 못 하는 것을 조용히 빼고 답하는 대신, 못 했다고 말한다." 제2장, 이해 보고서에 관한 문장입니다. 정직한 시스템의 값어치는 그 정직을 읽어 주는 조직에서만 실현됩니다. "좀혀서 답했다"는 고백을 다음에 지을 구역의 지도로 읽는 문화가 이 문장의 나머지 절반입니다.

"포이에시스에서 '만든다'는 것은 곧 '하네스를 짠다'는 것이다." 제3장의 첫 문장이자, 이 프레임워크에

서 제작의 정의를 바꾸는 문장입니다. 똑똑한 지능은 시장에서 구하고, 조직이 짓는 것은 그 지능이 우리 업무 안에서 어디까지 가고 어디서 멈추는지의 구조물이라는 역할 분담이 여기서 확정됩니다.

"자율성은 선언하는 것이 아니라, 작은 하네스에서부터 적립하는 것이다." 제3장, 세 크기의 하네스에 관한 문장입니다. 능력에서 태스크로, 태스크에서 활동으로 오르는 사다리의 각 단은 결국 신뢰의 질문이며, 그 신뢰는 확신이 아니라 기록 — 채택률, 정확도, 인증 — 으로 적립됩니다.

"에이전트에게 '무엇을 하라'만 가르치는 조직과 '무엇을 절대 하지 말라'까지 강제하는 조직 사이에는 리스크 등급의 차이가 있다." 제3장, 금지 규칙에 관한 문장입니다. 커버리지와 보수성 순서와 금지 규칙이라는 세 규율이, "올바른 방식으로만 일이 이루어진다"는 희망을 수학적으로 점점 가능한 성질로 바꿉니다.

"실행해 보지 않은 주장에는 꼬리표가 붙는다." 제3장, 모의 실행에 관한 문장입니다. 배포 승인의 안전이 담당자의 의견에서 비행 기록으로 바뀌는 문화, 그리고 모의 실행 없는 승인 요청을 반려하는 관행이 이 문장의 실무 버전입니다.

"문서와 실체가 다르다는 오래된 병이, 문서가 곧 실체인 구조로 치유된다." 제4장, 넥서스에 관한 문장입니다. 요구사항 문서와 프로세스 도면과 데이터 흐름도와 통제 매트릭스가 하나의 원본에서 기계적으로 파생되는 일곱 관점일 때, 서로 어긋난 네 장의 종이라는 풍경은 원리적으로 사라집니다.

"모든 결론에 계보가 있고, 그 계보가 다운로드 버튼으로 존재한다." 제4장, 미메시스 엔진에 관한 문장입니다. "그 결과 어떻게 나온 거예요?"가 개발팀 문의가 아니라 탭 하나 여는 일이 되는 상태 — 감사 대응의 관점에서 이 프레임워크가 약속하는 종착점입니다.

"책임의 추적이 사후 조사가 아니라 자료 구조다." 제4장, 약속의 고리에 관한 문장입니다. 요청되고 약속되고 진술되고 수락되는 네 위상, 그리고 위반됨이라는 상태 값이 있기에, 여러 에이전트가 얽힌 사고에서도 어디서부터 어긋났는지가 장부 조회의 문제가 됩니다.

"이곳에서 AI는 심사역을 대체하지 않는다. 심사역의 곁에 앉는다." 제5장, CLAW의 설계 철학에 관한 문장입니다. 증강이라는 선택은 기술의 한계가 아니라 신뢰의 경제학과 책임 구조와 변화 관리를 함께 계산한 결과이며, 그래서 증강 워크플레이스는 기술 전략이기 이전에 채택 전략입니다.

"판단은 여전히 사람의 몫이다. 다만 판단의 재료가 준비되는 속도가 달라진 것이다." 제5장, "분석 한 문장"에 관한 문장입니다. 증강 단계의 효율이 어디서 나오는지 정확히 짚는 문장이며, 효율 지표를 설계할 때 무엇을 재야 하는지 — 판단의 대체율이 아니라 재료 준비의 속도와 농침의 감소 — 도 함께 알려줍니다.

"확인의 물음표는 마찰이 아니라 통제 지점이다." 제5장, 에이전트의 확인 요청 문화에 관한 문장입니다. 편의를 이유로 물음표를 걷어 내자는 요구는 반드시 나오며, 그 요구에는 물음표의 삭제가 아니라 해당 스킬의 정식 승격 심사로 답하는 것이 옳습니다.

"증강의 대시보드는 그 자체로 자율화의 심사 서류다." 제5장, 성과 측정에 관한 문장입니다. 오늘의 채택률과 정확도 기록이 내일의 승격 근거가 되는 순환 — 증강 단계에서 기록을 쌓지 않는 조직은 다음 단계로 오를 사다리를 스스로 치우는 셈입니다.

이 문장들을 다시 읽어 보면 하나의 공통된 어조가 들립니다. 어느 문장도 기술의 놀라움을 자랑하지 않습니다. 대신 순서와 규율과 기록을 이야기합니다. 에이전틱 AI의 시대에 조직이 새로 배워야 할 것이 있다면, 그것은 더 큰 모델의 이름이 아니라 이 어조 — 자유를 주되 계보를 남기고, 속도를 내되 관문을 지키는 태도 — 일 것입니다.