

[Master Document]

차세대 공간 지능(Spatial Intelligence) 런타임 플랫폼 Spark.js 소프트웨어 아키텍처 및 산업 인프라 도입 백서

도입 의사결정 문서 (Adoption Decision Document)

1차 독자: 고객사·파트너사 CTO / 스마트팩토리·ADX 추진 총괄

Spark Spatial Intelligence OS - 차세대 공간 지능 디지털 트윈 런타임

Executive Summary (경영진 요약)

1. 문제

기존 3D 도구(Trellis, Tripo3D 등 정적 3D 생성기)는 공장을 "굳어버린 3D 조각상"으로만 재현합니다. 설비의 온도·진동·기계 관절을 이해하지 못하므로 고장 예측이나 자율 복구에 쓸 수 없고, 다운타임은 여전히 사람이 사후에 발견합니다.

2. 해결

Spark.js 기반 공간 지능 런타임은 공장 전체를 수백만 개의 가우시안 점으로 재구성하고, 여기에 **신경계(실시간 감각·정렬) / 두뇌(자연어-VLM 지각) / 물리 엔진(PINN·로봇 제어)**을 결합해 [센서 수집 → 인지 → 미래 고장 예측 → 자율 복구]의 4단계 폐쇄 루프를 구현합니다. 웹 표준(WebGL2) 기반이므로 관제실·태블릿·XR 기기 어디서나 동일한 화면으로 관제합니다.

3. 기대 효과 (KPI 목표치, 제6장에서 계약 가능한 측정 정의로 구체화)

지표	목표 [목표]	측정 정의 위치
비계획 다운타임	파일럿 라인 기준 감소율 목표 합의 후 확정	6.2
이상 징후 선행 감지	고장 발생 전 사전 경고 비율	6.2
제어 응답 지연	센서 감지 → 제어 명령 도달 < 20ms (Tier 1)	3.6 / 6.2
관제 렌더링	4K 60FPS 최소 보증, 120Hz 목표	4.3 / 6.2

4. 투자 규모 (원문 산정치, 제5장 산정 근거표 참조)

- 옵션 A (PoC / MVP, 1개 라인): 약 1,800만 ~ 2,500만 원 – 관제 겸용 엣지 워크스테이션 + 중량 AI 클라우드 렌탈
- 옵션 B (정식 상용화 엔터프라이즈): 약 1억 5,000만 ~ 2억 5,000만 원 – 완전 분리형 3-Tier(Edge RTX 6000 Ada + Core H100 클러스터 + Terminal RTX 4090 3대)
- 옵션 B 금액은 GPU 시세·환율에 민감하므로 5.3의 산정 근거표와 재산정 결과를 함께 검토 [?]

5. 도입 로드맵

1단계 PoC(3개월, 옵션 A) → 파일럿 성공 판정(6.4 게이트) → **2단계** 라인 확장 → **3단계** 전사 3-Tier 상용화 (옵션 B). 각 단계 전환은 정량 KPI 게이트로만 결정합니다.

6. 의사결정 요청 사항

- 옵션 A 파일럿 착수 승인 및 대상 공정(예: 전기차 모터 권선 조립) 지정
- 파일럿 예산 범위(1,800만 ~ 2,500만 원) 및 기간(3개월) 승인
- 고객사 측 협력 사항 승인: 현장 촬영·IoT 데이터 접근·PLC 연동 담당자 지정 (5.9)
- 보안 등급 결정: 파일럿 클라우드 허용 여부 vs 초기부터 온프레미스/에어갭 (5.10)

목차

문서 관리 정보	2
독자 안내 (3층 구조)	2
Executive Summary (경영진 요약)	3
1. 문제	3
2. 해결	3
3. 기대 효과 (KPI 목표치, 제6장에서 계약 가능한 측정 정의로 구체화)	3
4. 투자 규모 (원문 산정치, 제5장 산정 근거표 참조)	3
5. 도입 로드맵	3
6. 의사결정 요청 사항	3
약어표	5
제 1장. 서론: 공간 지능(Spatial Intelligence) 시대의 도래	7
1.1. 공간 지능의 정의 (페이페이 리 교수 및 World Labs의 비전)	7
1.2. 한눈에 보는 비교: 트렐리스(Trellis) vs 스파크(Spark.js)	7
1.3. 기술 계보: NeRF → 3DGS → 웹 3DGS 런타임	7
1.4. 산업 동향: 공간 지능-피지컬 AI 생태계에서의 위치	8
1.5. 문서 범위 · 비범위 · 전제	8
제 2장. 기존 3D 공간과 Spark.js 공간 지능의 4대 차이점	9
2.5. 5축 비교 매트릭스	9
2.6. 4대 차이점별 검증 방법	9
2.7. 산업 디지털 트윈 플랫폼 대비 포지셔닝	10
제 3장. Spark.js 핵심 내부 엔진 아키텍처 (소프트웨어 시스템)	11
3.0. 엔드투엔드 데이터 플로우	11
3.1. ♀ [신경계 - The Nervous System]	12
3.2. [두뇌 - The Brain]	13
3.3. ⚙ [물리 엔진 - The Physics Engine]	14
3.4. 4단계 자율 폐쇄 루프 (Closed-Loop) 완결 프로세스	16
3.5. 성능 벤치마크 방법론	16
3.6. 폐쇄 루프 시퀀스 및 지연 예산	16
3.7. 공간 취득·갱신 파이프라인	17
제 4장. 산업용 공간 지능 시스템 3계층(3-Tier) 아키텍처 및 인프라 설계	18
4.1. [Tier 1] 엣지 서버 (현장 물리 제거기)	18
4.2. [Tier 2] 중앙 AI 코어 (거대 월드 모델)	19
4.3. [Tier 3] 메인 통합 관제 워크스테이션 (런타임 클라이언트)	19
4.4. 네트워크 토폴로지 및 IoT 데이터 경로	19
4.5. 보안 · 가용성 · Fail-safe	20
4.6. 전력 · 냉각 · 랙 산정	20
4.7. 소프트웨어 스택	20
4.8. 하드웨어 세대 교체 대응 - 동급 대체 모델표	21
4.9. 런타임 전제 조건 (브라우저·XR)	21
제 5장. 예산 및 도입 규모별 전략 구성	22
5.1. 옵션 A: 공장 실증 및 파일럿 (PoC / MVP)	22
5.2. 옵션 B: 정식 상용화 엔터프라이즈 패키지 (Full Commercial)	22
5.3. 산정 근거표 및 3년 TCO	22
5.4. 단계별 로드맵 및 마일스톤 KPI	23
5.5. 리스크 매트릭스	23
5.6. 공급사 소개 (식별 정보 마스킹)	23
5.7. 책임 경계 (공급 범위 / 고객 범위 / 공동 범위)	24
5.8. 상업 조건 개요	24
5.9. PoC 실행 계획 (옵션 A, 3개월)	24
5.10. 보안 등급 단계화 (클라우드 → 온프레미스)	25

제 6장. KPI 및 검증 체계	26
6.1. 검증 원칙	26
6.2. KPI 정의표	26
6.3. 검수 절차	26
6.4. 파일럿 성공 기준 (게이트 1)	26
제 7장. 표준 · 규제 · 안전 케이스 · 데이터 거버넌스	28
7.1. 적용 표준 체크리스트 (고객사 심사팀용)	28
7.2. 안전 케이스 (Safety Case) – 자율 등급 단계	28
7.3. 데이터 주권 및 거버넌스	28
제 8장. 🏆 최종 결론 (Executive Summary)	29
8.1. 의사결정 항목 재확인	29
제 9장. 예상 반론 대응 (FAQ)	30
부록 A. 전체 코드 원문 및 발췌 범위 명세	31
A-4. 발췌 범위 명세	33
부록 B. 벤치마크 조건 명세 (기록 양식)	34
부록 C. 용어집	35
부록 D. 참고문헌	36
부록 E. 원문 ↔ 보고서 매핑표 (누락 0건 증명)	37

약어표

약어	전체 명칭	의미
SI	Spatial Intelligence	공간 지능 (본 문서 고정 의미)
Sler	System Integrator	시스템 통합사 – SI와 구분하기 위해 Sler로 표기
3DGS	3D Gaussian Splatting	가우시안 타원체 집합으로 공간을 표현·렌더링하는 기법
WASM	WebAssembly	브라우저에서 네이티브 속도로 실행되는 바이너리 포맷
SPZ	SPZ splat format	3DGS 압축 파일 포맷
LoD	Level of Detail	거리·화면 크기에 따라 세부 수준을 조절하는 기법
VLM	Vision-Language Model	시각-언어 통합 모델
LLM	Large Language Model	대규모 언어 모델
LAM	Large Action Model	행동 생성 모델
PAI	Physical AI	피지컬 AI(물리 세계에서 행동하는 AI)
PINN	Physics-Informed Neural Network	물리 미분방정식 제약을 학습에 포함한 신경망
XAI	Explainable AI	설명가능한 AI
SDF	Signed Distance Field	부호 거리장
AGV	Automated Guided Vehicle	무인 운반차
PLC	Programmable Logic Controller	산업 제어기
OT / IT	Operational / Information Technology	운영 기술 망 / 정보 기술 망
HA	High Availability	고가용성
TCO	Total Cost of Ownership	총 소유 비용
PoC	Proof of Concept	개념 검증

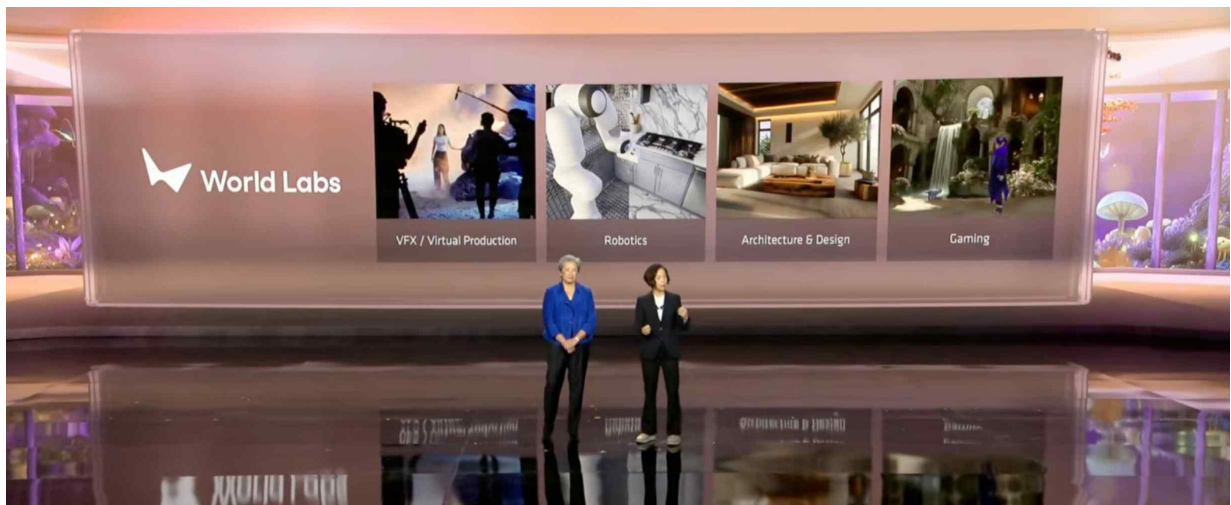
제 1장. 서론: 공간 지능(Spatial Intelligence) 시대의 도래



문서 개요: 본 문서는 글로벌 AI 및 3D 그래픽스 산업의 핵심 화두인 '**공간 지능(Spatial Intelligence)**'의 본질을 분석하고, 정적 3D 생성 AI(Trellis 등)와 동적 공간 지능 런타임(Spark.js)의 차이를 규명합니다. 나아가 Spark.js 내부의 3대 엔진(신경계, 두뇌, 물리 엔진) 동작 원리와 실제 코드를 공개하며, 이를 실제 스마트 팩토리 등 산업 현장에서 무중단 자율 폐쇄 루프(Closed-Loop)로 구현하기 위한 3계층(3-Tier) 하드웨어 인프라 설계 표준을 종합적으로 제시합니다.

1.1. 공간 지능의 정의 (페이페이 리 교수 및 World Labs의 비전)

스탠포드 대학교 페이페이 리(Fei-Fei Li) 교수가 주도하는 차세대 AI의 핵심 화두는 '**공간 지능(Spatial Intelligence)**'입니다. 기존의 AI가 2D 이미지나 텍스트를 인식(Seeing)하는 데 그쳤다면, 공간 지능은 AI가 3차원 물리 세계의 **구조(Geometry)**, **물리적 법칙(Physics)**, **사물의 쓰임새(Affordance)**를 완벽하게 이해하고(Understanding), 미래 상태를 예측하며(Predicting), 물리적 행동(Action)을 자율적으로 수행하는 능력을 의미합니다.



1.2. 한눈에 보는 비교: 트렐리스(Trellis) vs 스파크(Spark.js)

[마이크로소프트 Trellis / Tripo3D]	[Spark.js (공간 지능 런타임)]
• 역할: 3D 형상 생성기 (Static 3D Generator)	• 역할: 살아있는 공간 지능 실행 엔진 (Spatial Runtime)
• 결과물: 굳어있는 박제된 3D 조각상 (.glb/.fbx)	• 결과물: 수백만 개의 점이 물리 법칙과 프롬프트에 반응
• 상호작용: 마우스로 돌려보기만 가능 (정적)	• 상호작용: 프롬프트 입력으로 공간 변형, IoT 실시간 연동
• 물리/지능: 중력, 온도, 기계 관절을 이해 못함	• 물리/지능: PINN 물리 법칙, 로봇 관절 제어, XAI 인과 추론



비유하자면: Trellis가 사진 한 장으로 멋진 '정지된 밀랍 인형(Sculpture)'을 빚어내는 조각가라면, Spark.js는 그 밀랍 인형과 공간에 '신경계와 두뇌, 물리 엔진'을 부여하여 프롬프트를 알아듣고 스스로 움직이게 만드는 '생명체 런타임'입니다.

표 1-1. 정적 3D 생성기와 공간 지능 런타임의 역할 구분 (원문 비교 박스의 정식 표 재구성)

구분	Trellis / Tripo3D (정적 3D 생성기)	Spark.js (공간 지능 런타임)
핵심 역할	이미지·텍스트 → 3D 형상 1회 생성	생성된/촬영된 3DGS 공간을 실시간 실행·변형·제어
결과물 형태	고정 메시(.glb/.fbx)	수백만 가우시안 점 + 메시 하이브리드 씬
시간축	없음 (정지)	60~120Hz 갱신, IoT 스트림 동기화
입력 인터페이스	뷰어 조작	자연어 프롬프트, 센서, 제어 API
물리·지능	없음	PINN, 로봇 관절 제어, XAI 인과 로그
도입 위치	콘텐츠 제작 단계	운영(Run-time) 단계 - 관제·제어

1.3. 기술 계보: NeRF → 3DGS → 웹 3DGS 런타임

공간 지능 런타임의 기반 기술인 3D Gaussian Splatting은 신경 방사장(NeRF, 2020)이 제기한 "사진에서 실사 3D 공간을 복원한다"는 문제를, 수백만 개의 가우시안 타원체로 표현하고 GPU 래스터라이저로 그리는 방식으로 재정의한 기법입니다(SIGGRAPH 2023). NeRF가 픽셀당 신경망 추론을 요구해 실시간성이 부족했던 반면, 3DGS는 정렬(sort)과 알파 블렌딩만으로 실시간 렌더링이 가능해 웹 브라우저까지 내려올 수 있었습니다.

표 1-2. 기술 계보 요약 [공식]

세대	대표 기술	표현 방식	실시간성	산업 적용 한계
1세대	포토그래메트리 메시	폴리곤 + 텍스처	높음	복원 품질·반사·투명 재질 취약, 수작업 후처리
2세대	NeRF (2020)	암시적 신경 함수	낮음	렌더 지연으로 관제·제어 부적합
3세대	3DGS (2023)	명시적 가우시안 집합	높음	단일 장면·데스크톱 위주, 대규모 장면 메모리
4세대	웹 3DGS 런타임 (Spark.js 2.x, 2026)	3DGS + LoD 스트리밍 + 셰이더 그래프	높음 (WebGL2)	본 백서가 다루는 산업 적용 단계

Spark.js는 World Labs가 공개한 THREE.js용 오픈소스 3DGS 렌더러로, 스플랫과 메시 객체를 하나의 렌더링 파이프라인에서 융합하고 98% 이상의 WebGL2 지원 기기를 대상으로 합니다 [공식]. 2026년 4월 공개된 Spark 2.0은 연속 LoD 트리, 점진적 .RAD 스트리밍, GPU 가상 메모리 페이지징을 도입해 1억 개 이상의 스플랫 월드를 모바일·VR까지 스트리밍할 수 있게 했으며, LoD 트리 순회는 Rust로 작성되어 WebAssembly로 컴파일된 코드가 백그라운드 WebWorker에서 비동기로 수행합니다 [공식]. 원문 3.1절의 "Rust WASM 백그라운드 정렬" 구조는 이 공식 아키텍처와 부합합니다.

1.4. 산업 동향: 공간 지능·피지컬 AI 생태계에서의 위치

공간 지능은 학계 개념에서 산업 제품군으로 이동 중입니다. World Labs는 3DGS 기반 월드 생성(Marble)과 웹 런타임(Spark)을 공개했고, GPU 제조사는 Omniverse·Isaac 계열로 산업 디지털 트윈과 로봇 시뮬레이션을 묶는 전략을 취하고 있으며, 완성차·항만·전자 제조사는 "정적 디지털 트윈"에서 "센서 동기 + 예측 + 제어"로 요구 수준을 높이고 있습니다. 본 플랫폼은 이 생태계에서 **브라우저 네이티브 실사(3DGS) 런타임 + 오픈프리미스 AI 제어 계층**이라는 위치를 점합니다. 상용 플랫폼 대비 포지셔닝은 2.7절에서 다룹니다.

1.5. 문서 범위 · 비범위 · 전제

표 1-3. 범위 정의

구분	내용
범위	Spark.js 기반 공간 지능 런타임의 소프트웨어 아키텍처, 4단계 폐쇄 루프, 3-Tier 인프라 사양, 도입 옵션·예산, KPI·검증 체계, 표준·안전·데이터 거버넌스, 상업 조건 골격
비범위	특정 고객사 공정의 상세 PLC 프로그램, 로봇 제조사별 SDK 구현, 3DGS 학습 알고리즘 자체의 수학적 증명, 세부 견적서(별도 제출)
전제	고객사가 IoT 센서 데이터(온도·진동 등)에 접근 권한을 제공하고, 대상 라인의 촬영(3DGS 취득)이 허용되며, 로봇/AGV 제어 인터페이스(PLC/EtherCAT 또는 로봇 API)가 문서화되어 있음
독자 가정	1차 독자는 GPU·네트워크·로봇 제어에 대한 기본 이해가 있는 기술 의사결정권자

제 2장. 기존 3D 공간과 Spark.js 공간 지능의 4대 차이점

- ① 정적 폴리곤(Mesh) vs **살아있는 프로그래머블 가우시안 (Dyno)** 기존 3D는 삼각형 면(Polygon)이 고정되어 있어 실시간 파괴나 열역학 변형이 불가능합니다. 반면 **Spark.js의 Dyno 노드 엔진**은 수백만 개의 가우시안 타원체가 수학적 파라미터를 가져, 프롬프트 명령 즉시 유체역학 등 물리 법칙에 맞춰 실시간으로 재배열되고 춤을 춥니다.
- ② **자연어 프롬프트(Prompt)**로 3D 공간을 실시간 조작 (Spatial Copilot) Spark.js에 내장된 [AI 3D Copilot]에 자연어로 "모터 조립 속도를 15% 가속해줘"라고 명령하면, AI가 공간 내부의 의미를 이해하고 실제 3D 로봇 팔의 회전 속도와 AGV 궤적을 물리적으로 변경합니다.
- ③ 사물의 '**쓰임새(Affordance)**'와 0.01mm 정밀 계측 인식 (VLM 결합) 시각-언어 모델(VLM)과 결합하여 "이것은 전기를 넣으면 회전하는 모터 로터다", "이 틈새는 0.01mm 오차 이하로 로봇이 꽂아야 하는 곳이다"라는 사물의 쓰임새를 인지하고 피지컬 AI 로봇에게 정확한 좌표를 전달합니다.
- ④ **4단계 자율 폐쇄 루프 (Closed-Loop)** 디지털 트윈 완성 Trellis 모델과 달리 Spark.js는 현실 공장의 IoT 센서와 60FPS로 동기화되어 [센서 수집 → 인지 → 미래 고장 예측 → 자율 복구]의 과정을 인간 개입 없이 수행합니다.

2.5. 5축 비교 매트릭스

원문의 2자 비교(Trellis vs Spark.js)를, CTO가 실제로 검토 대상에 올리는 도구들까지 확장했습니다. 각 셀은 공개 문서·제품 소개 기준이며 세부 기능은 버전에 따라 달라질 수 있습니다.

표 2-1. 5축 비교 매트릭스

축	Trellis (MS Research)	Tripo3D	Luma / Polycam 계열	SuperSplat (PlayCanvas)	Spark.js 런타임
① 생성·취득	이미지→3D 생성	이미지/텍스트 →3D 생성	촬영→3DGS/NeRF 복원	3DGS 편집기(취득은 외부)	외부 3DGS 취득 + 절차적 생성(constructSplats)
② 렌더링	메시 출력	메시 출력	앱/웹 뷰어	웹 3DGS 편집·뷰어	웹 3DGS + 메시 융합, LoD 스트리밍
③ 상호작용	뷰어 회전	뷰어 회전	뷰어 탐색	편집(자르기·색)	자연어 프롬프트, 셰이더 그래프(Dyno), API 제어
④ 물리·지능	없음	없음	없음	없음	PINN 예측, 로봇 제어 신호, XAI 로그 (본 플랫폼 계층)
⑤ 산업 연동	없음	없음	제한적	없음	IoT 동기화, PLC/EtherCAT, OPC UA 브리지 (4장)

※ Trellis-Tripo3D는 "콘텐츠 제작 단계", Spark.js는 "운영 단계" 도구이므로 상호 대체재가 아니라 파이프라인의 앞 뒤 단계로도 결합 가능합니다.

2.6. 4대 차이점별 검증 방법

표 2-2. 차이점 → 검증 절차 (PoC 검수 항목으로 그대로 사용 가능)

차이점	주장	검증 방법	합격 기준 [목표]	근거 등급
① Dyno 가우시안	프롬프트 즉시 물리 법칙에 맞춰 재배열	셰이더 그래프 변형 명령 후 프레임 시간·시각 변화 기록	변형 반영 ≤ 2 프레임, FPS 하락 $\leq 10\%$	[목표]
② Spatial Copilot	자연어로 로봇 속도·AGV 궤적 변경	표준 명령 20종 입력 → 제어 신호 로그 대조	의도 일치율 $\geq 95\%$, 오작동 0건	[목표]
③ Affordance/0.01mm	VLM이 쓰임새·정밀 위치 인지	VLM 의미 인식 + 정밀 계측(비전/레이저) 결합 결과를 기준기와 비교	의미 인식 정확도 $\geq 90\%$, 위치 오차는 계측 장비 사양 이내	3.2-보강 참조
④ 폐쇄 루프	센서→인지→예측→복구 무개입 수행	주입 시나리오(과열·진동) 10회 재현, 단계별 타임스탬프	4단계 완주율 100%, 지연 예산(3.6) 준수	[목표]

2.7. 산업 디지털 트윈 플랫폼 대비 포지셔닝

기술 의사결정권자의 실제 비교 대상은 3D 생성기가 아니라 NVIDIA Omniverse, Siemens Xcelerator(Industrial Metaverse), Unity Industry 등 산업 디지털 트윈 플랫폼입니다. 본 플랫폼은 이들을 대체한다기보다, "실사 3DGS 기반 브라우저 관제 + 온프레미스 AI 제어 계층"으로 보완·병행이 가능합니다.

표 2-3. 산업 DT 플랫폼 포지셔닝 (공개 자료 기준, 세부는 계약 검토 시 재확인)

관점	Omniverse 계열	Siemens 계열	Unity Industry	본 플랫폼(Spark.js 기반)
공간 표현	USD 메시·시뮬레이션	CAD/PLM 연계 메시	메시·게임엔진	실사 3DGS + 메시 융합
클라이언트	전용 앱/RTX 스트리밍	전용 클라이언트	빌드 배포	웹 브라우저(WebGL2), 설치 불필요
데이터 취득	CAD·시뮬레이션 데이터	CAD/PLM	수작업 모델링	현장 촬영 → 3DGS (3.7)
AI 제어 계층	Isaac 등 별도 스택	PLM 연동	별도 구현	두뇌·물리 엔진 내장(3장), 온프레미스
라이선스 구조	상용 구독	상용	상용	런타임 MIT 오픈소스 + 자사 계층 라이선스 (5.8)
적합 상황	대규모 시뮬레이션·합성 데이터	CAD 중심 제조	커스텀 앱	기존 라인의 신속한 실사 관제·예측·제어

※ 위 표는 "왜 Omniverse가 아닌가"라는 질문에 대한 근거이며, 9장 FAQ에서 문답 형태로 다시 다룹니다.

제 3장. Spark.js 핵심 내부 엔진 아키텍처 (소프트웨어 시스템)

오픈소스 Spark.js와 우리가 직접 구축한 생성 AI 기반 공간 지능 디지털 트윈의 실제 코드를 통해, 단순 뷰어를 넘어 "신경계, 두뇌, 물리 엔진"으로 진화하는 과정을 상세히 공개합니다.

그림 3-1. 3대 엔진 아키텍처

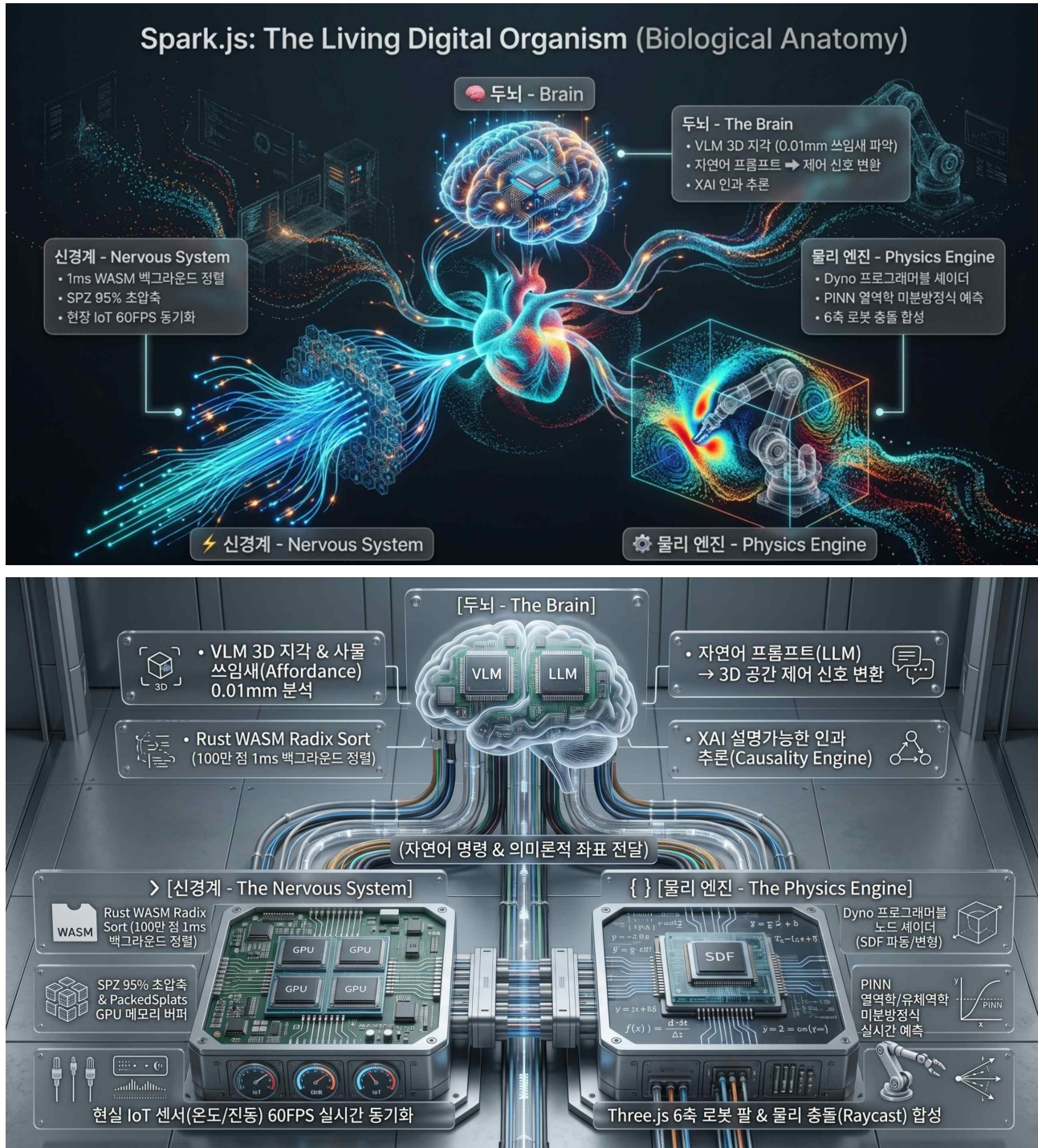


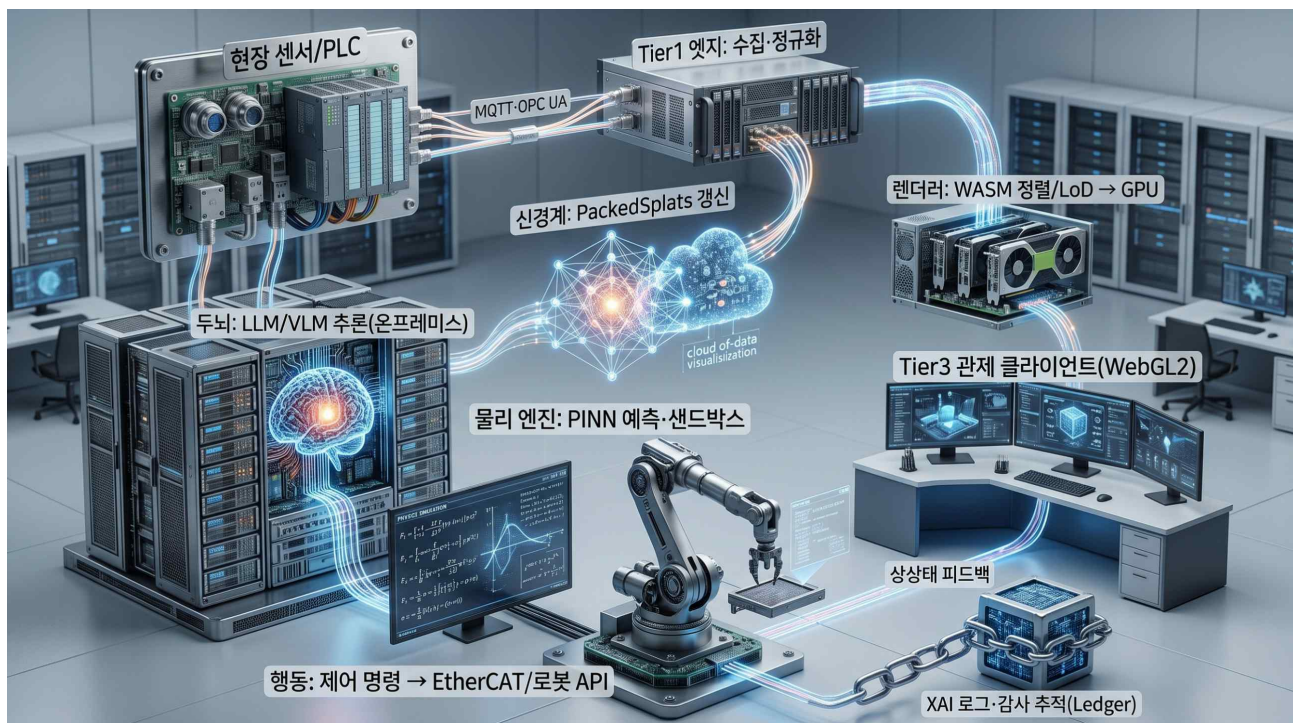
표 3-0. 3대 엔진 역할 요약 (그림 3-1의 정식 표 재구성)

엔진	핵심 구성 요소	입력	출력	실행 위치
신경계	Rust WASM 백그라운드 정렬/LoD, SPZ 압축, PackedSplats GPU 버퍼, IoT 60FPS 동기화	스플랫 파일, 센서 스트림	갱신된 GPU 스플랫 텐서	Tier 3 브라우저 (렌더) + Tier 1(수집)
두뇌	VLM 3D 시각-Affordance 분석, LLM 프롬프트→제어 신호 변환, XAI 인과 추론	자연어 명령, 씬 스냅샷, 센서 이상 이벤트	의미론적 좌표, 제어 의도, 설명 로그	Tier 1(온프레미스 추론)
물리 엔진	Dyno 노드 셰이더(SDF 파동/변형), PINN 미분방정식 예측, Three.js 로봇 팔·Raycast 합성	제어 의도, 경계 조건	변형된 씬, 미래 상태 예측, 토크 명령	Tier 1(PINN) + Tier 3(Dyno 시각화)

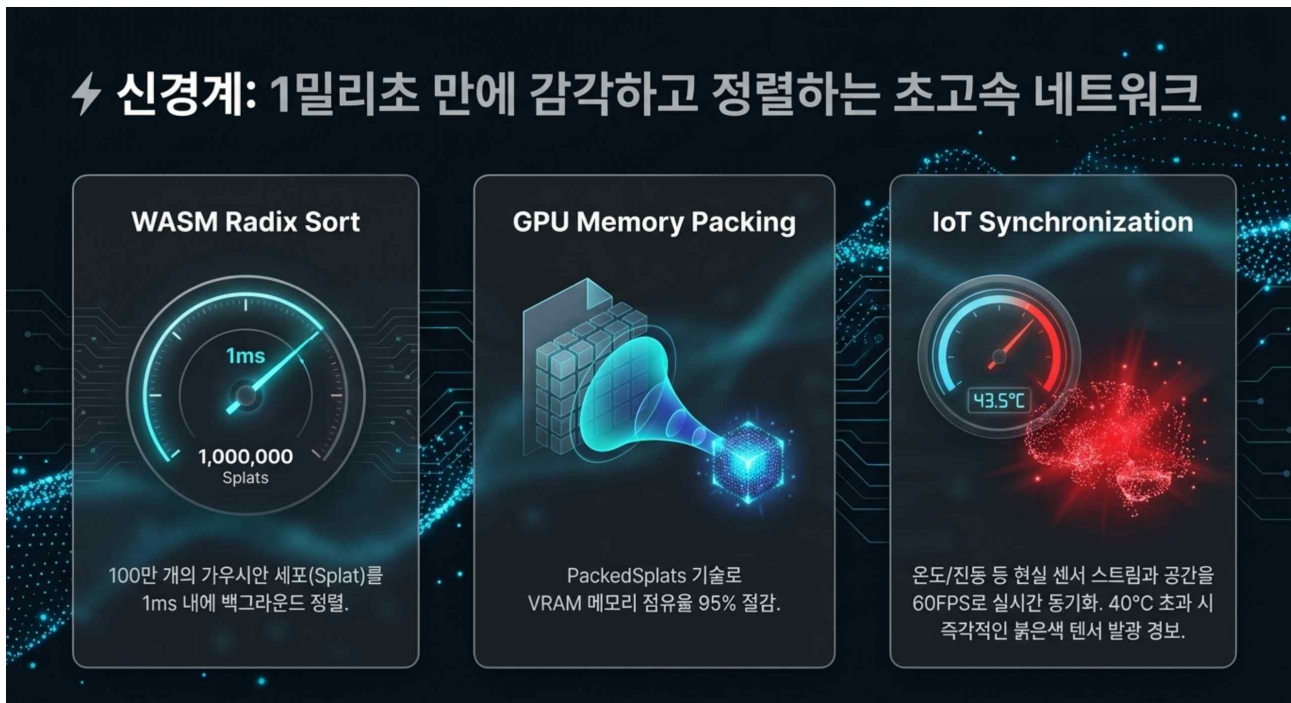
3.0. 엔드투엔드 데이터 플로우

원문 다이어그램이 3대 엔진의 역할을 보여준다면, 아래는 데이터가 실제로 흐르는 순서입니다. 각 화살표는 4.4의 네트워크 경로와 3.6의 지연 예산에 대응됩니다.

그림 3-2. 데이터 플로우



3.1. ⚡ [신경계 - The Nervous System]



"수백만 개의 세포(스플랫)를 1밀리초 만에 감각하고 정렬하는 초고속 신경망"

일반 3D 뷰어의 한계를 극복하기 위해, Rust로 작성된 WebAssembly(WASM) 병렬 정렬 엔진과 **GPU 메모리 패킹(PackedSplats)**을 통해 초당 60~120회 실시간 반응합니다.

코드 3-1. 신경계 (원문 그대로, 라인 번호만 부여)

```

1 import * as THREE from "three";
2 import { SparkRenderer, SplatMesh, PackedSplats } from "@sparkjsdev/spark";
3
4 // [1] 신경계 코어: 16/32비트 GPU 메모리 버퍼 생성 (메모리 95% 절감)
5 const splatMesh = new SplatMesh({
6   url: "https://sparkjs.dev/assets/splats/industrial_motor.spz",
7   lod: true,
8   extSplats: true,
9   onProgress: (ev) => console.log(`신경망 로딩: ${Math.round((ev.loaded/ev.total)*100)}%`)
10 });
11 scene.add(splatMesh);
12
13 // [2] 백그라운드 WebWorker (Rust WASM Radix Sort) 연동 구조
14 await splatMesh.initialized;
15 console.log(`신경계 활성화 완료: 총 ${splatMesh.numSplats.toLocaleString()}개 가우시안 세포 연결됨`);
16
17 // [3] IoT 실시간 센서 스트림 → 가우시안 텐서 신경 반사
18 function updateSensorNervousSystem(temperature, vibrationHz) {
19   if (!splatMesh.packedSplats) return;
20   if (temperature > 40.0) {
21     const alertTint = new THREE.Color(1.0, 0.2, 0.2); // 과열 시 붉은색 텐서 발광
22     splatMesh.splatRgba = new Float32Array([alertTint.r, alertTint.g, alertTint.b, 1.0]);
23   }
24 }

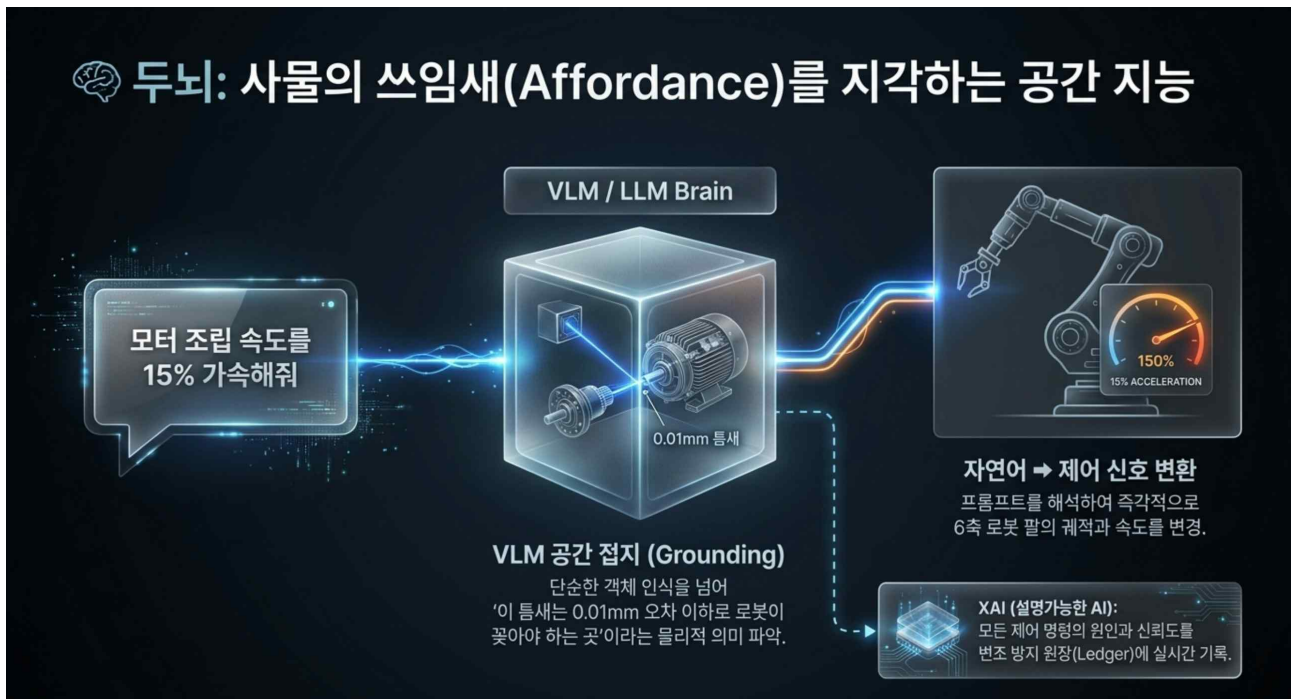
```

표 3-1. 코드 3-1 라인 해설 및 Spark.js 공식 API 대조

라인	동작	공식 API 대조 (Spark 2.x 문서 기준)	등급
2	SparkRenderer, SplatMesh, PackedSplats import	세 클래스 모두 공식 export. SparkRenderer는 씬에 add하여 사용	[공식]
5~10	URL로 .spz 로드, 진행률 콜백	SplatMesh는 url / fileBytes / stream / packedSplats로 초기화 가능, .spz.ply.sog.rad 등 지원. onProgress 콜백 공식	[공식]
7	lod: true	Spark 2.0의 LoD 스트리밍 기능과 대응되는 옵션. 정확한 옵션명·기본값은 사용 버전의 SplatMesh 문서로 재확인	-
8	extSplats: true	확장 스플랫(covariance 등) 관련 옵션으로 추정. 버전별 명칭 확인 필요	-
14	await splatMesh.initialized	초기화 완료 대기. onLoad 콜백과 동일 목적	[공식]
15	numSplats	공식 속성	[공식]
18~24	IoT 값에 따라 색 텐서 갱신	packedSplats 수정 후 needsUpdate = true가 공식 갱신 경로. splatRgba 직접 대입은 데모 편의 코드로, 상용 구현은 objectModifiers/worldModifiers(셰이더 그래프) 사용 권장	상용 구현 시 변경

※ 원문 다이어그램의 "100만 점 1ms 정렬", "SPZ 95% 초압축"은 자체 벤치마크 값으로, 측정 조건은 부록 B 양식에 따라 기록·첨부 예정 [실측 예정/?].

3.2. [두뇌 - The Brain]



"자연어 명령을 알아듣고 사물의 쓰임새(Affordance)를 지각하는 공간 지능"

VLM과 AI Copilot을 통해 3D 공간을 의미론적으로 파악하고 명령을 물리 제어 신호로 변환합니다.

코드 3-2. 두뇌 (원문 그대로)

```
1 // [1] 두뇌 코어: 자연어 프롬프트를 해석하여 3DGS 씬과 로봇 기계를 조작
2 async function executeSpatialBrainCommand(prompt) {
3   appendXaiLog('[두뇌/LLM] 자연어 명령 수신: "${prompt}");
4   const lower = prompt.toLowerCase();
5
6   if (lower.includes("모터") || lower.includes("가속") || lower.includes("조립")) {
7     showDigitalTwinMode("ev");
8     robotArmSpeedMultiplier = 1.5;
9     rotorRotationSpeed = 4.0;
10    appendXaiLog('[VLM/SI] 모터 조립 어포던스(0.01mm) 정렬 완료 → 로봇 궤적 150% 가속 신호 인가');
11  }
12  else if (lower.includes("항만") || lower.includes("agv")) {
13    showDigitalTwinMode("port");
14    triggerAgvDispatch("Bay-02");
15    appendXaiLog('[PAI/LAM] 스마트 항만 크레인 레이저 정렬 가이드 및 AGV 자율 주행 궤적 생성');
16  }
17  else if (lower.includes("사이버펑크") || lower.includes("네온")) {
18    setStudioLightingEnvironment("cyberpunk");
19    triggerSdfPulseWave();
20    appendXaiLog('[두뇌/환경] 사이버펑크 네온 스펙트럼 및 SDF 동적 조명 필드 전환 완료');
21  }
22 }
23
24 // [2] 설명가능한 AI(XAI) 인과 추론 엔진 실시간 스트리밍
25 function appendXaiLog(message) {
26   const logStream = document.getElementById("xai-log-stream");
27   const entry = document.createElement("div");
28   entry.className = "xai-log-entry";
29   entry.innerHTML = `<span class="xai-badge xai-badge-ai">LIVE</span> <span>${message}</span>`;
30   logStream.appendChild(entry);
31   logStream.scrollTop = logStream.scrollHeight;
32 }
```

3.2-보강 A. 코드 3-2의 성격과 발체 범위

코드 3-2는 **프로토타입 데모 발체본**입니다. lower.includes("모터") 계열의 조건문은 명령 라우팅의 골격을 보여주기 위한 것이며, 상용 두뇌에서는 아래 3.2-보강 B의 LLM/VLM 파이프라인이 이 자리를 대체합니다. showDigitalTwinMode, robotArmSpeedMultiplier, rotorRotationSpeed, triggerAgvDispatch, setStudioLightingEnvironment, triggerSdfPulseWave는 데모 애플리케이션의 다른 모듈에 정의된 함수·변수로, 본 발체본에는 포함되지 않습니다(부록 A 발체 범위 명세 참조).

3.2-보강 B. 상용 LLM/VLM 파이프라인

표 3-2. 상용 두뇌 파이프라인 구성

단계	역할	배치 위치	후보 구성	지연 예산 [목표]
① 의도 분류	자연어 명령 → 작업 유형(속도 조정/디스패치/환경) 분류	Tier 1	경량 LLM(온프레미스, 7~14B급 양자화) 또는 규칙+LLM 하이브리드	≤ 300ms
② 공간 접지 (Grounding)	명령 속 대상("모터", "Bay-02")을 3DGS 씬의 객체 ID·좌표로 연결	Tier 1	VLM(씬 스냅샷 + 객체 라벨 맵) + 공간 인덱스	≤ 500ms
③ 정밀 계측 결합	VLM이 지목한 위치를 계측 장비 값으로 보정	Tier 1	비전 계측 카메라 / 레이저 변위 센서 (0.01mm급은 계측 장비 사양이 결정)	장비 의존
④ 제어 계획	목표 → 안전 범위 내 제어 파라미터	Tier 1	규칙 기반 안전 필터 + 물리 엔진 사전 시뮬레이션	≤ 200ms
⑤ 실행·설명	제어 명령 송출 + XAI 로그(근거·대안·신뢰도)	Tier 1	EtherCAT/로봇 API + Ledger 기록	< 20ms(송출)
⑥ 대규모 학습·갱신	월드 모델 파인튜닝, 라벨 맵 갱신	Tier 2	H100 클러스터, 주기 배치	비실시간

핵심은 두 가지입니다. 첫째, 실시간 경로(①~⑤)는 전부 **온프레미스 Tier 1**에서 완결되어 클라우드 왕복이 없습니다. 둘째, 원문의 "0.01mm 정밀 계측 인식"은 VLM이 단독으로 달성하는 값이 아니라 **VLM(의미) + 계측 장비(수치)**의 결합으로 달성되며, 실제 정밀도는 결합된 계측 장비의 사양으로 보증됩니다. 이 정의를 6장 KPI에 그대로 반영합니다.

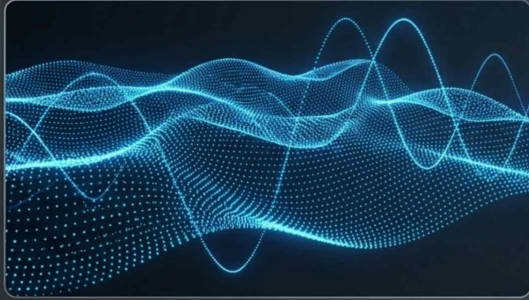
3.2-보강 C. XAI 로그의 상용 요건

- 모든 제어 명령은 원인(센서 값·규칙·모델 출력), 대안(검토된 다른 선택), 신뢰도, 안전 필터 통과 여부를 함께 기록
- 로그는 변조 방지 원장(append-only, 해시 체인)으로 저장하여 사후 감사·책임 소재 확인에 사용 (7장 데이터 거버넌스)
- 관제 화면(코드 3-2의 appendXaiLog UI)은 이 원장의 실시간 뷰로 동작

3.3. ⚙️ [물리 엔진 - The Physics Engine]

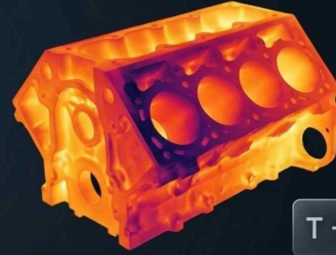
⚙️ 물리 엔진: 미분방정식과 셰이더로 공간을 실시간 변형

Dyno 프로그래머블 노드



수백만 개의 점군을 수학적 물리 필드로 실시간 재배열 (유체/파동 시뮬레이션).

PINN (Physics-Informed Neural Network)



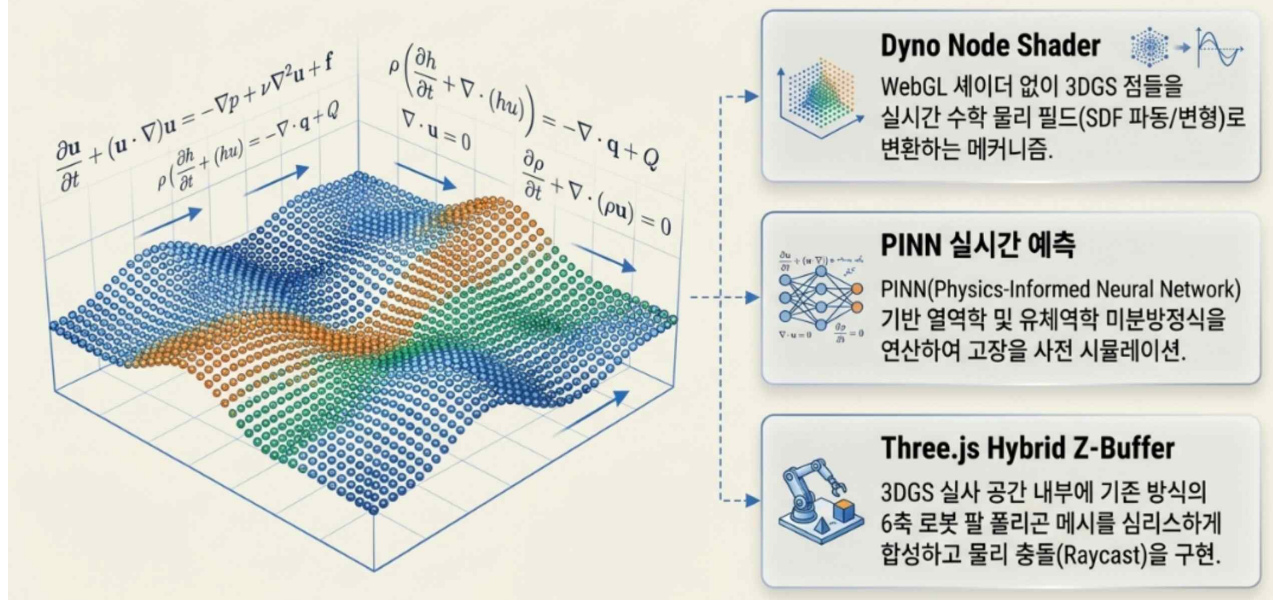
T + 2.4s

온도/진동 데이터를 기반으로 열전도 및 나비에-스토크스 방정식을 연산하여 단기 미래 상태(예: 2.4초 후 과열) 예측.

하이브리드 합성: 3DGS 실사 공간에 Three.js 기반의 고정밀 폴리곤 로봇 팔을 완벽한 Z-Buffer 깊이로 결합.

"Dyno 셰이더 그래프와 PINN 물리 미분방정식으로 공간을 변형하는 물리 엔진"

소프트웨어 해부도 3: ⚙️ 프로그래머블 물리 엔진 (The Physics Engine)



SISO(Single Input Single Output) → MIMO(Multiple Input Single Output) → MIMO(Multiple Input Multiple Output)

- 3DGS 생성 및 학습 단계 → MISO (Multiple Input, Single Output)
- Spark.js 실시간 구동 및 물리 제어 단계 → MIMO (Multiple Input, Multiple Output)

1. 3DGS 생성 및 학습 단계 → MISO (Multiple Input, Single Output)

3DGS 공간을 처음 구축하고 뼈대를 학습하는 초기 설계 단계는 전형적인 MISO(다중 입력, 단일 출력)구조를 따릅니다.

■ 다중 입력(Multiple Inputs):

다양한 각도와 위치에서 촬영된 수십~수백 장의 2D 실사 사진 및 비디오 프레임 데이터.

각 카메라의 정밀 기하학적 매개변수(내/외 매개변수, Intrinsic/Extrinsic) 데이터.

단일 출력(Single Output):

입력된 데이터들이 최적화 알고리즘을 거쳐 하나의 조밀한 3D 가우시안 공간 썸 데이터(수백만 개의 가우시안 세포 집합체인 단일 월드 모델 에셋)로 조율 및 수렴됩니다.

즉, 수많은 시각 정보를 융합해 하나의 정교한 가상 공간을 탄생시키는 단계는 MISO의 흐름입니다.

2. Spark.js 실시간 구동 및 물리 제어 단계 → MIMO (Multiple Input, Multiple Output)

MISO 파이프라인을 통해 구축된 가우시안 세포 공간이 Spark.js 공간 지능 운영체제(OS)에 이식되어 살아 움직이기 시작하면, 시스템은 매우 고도화된 **MIMO(다중 입력, 다중 출력)**시스템으로 완전히 탈바꿈합니다.

이 단계에서 Spark.js는 가상 가우시안 공간과 현실 물리 공장을 초저지연으로 연결하는 양방향 신경망 역할을 수행하게 됩니다.

① 다중 입력 (Multiple Inputs)

실시간 런타임 환경에서 시스템은 다양한 독립 변수 스트림을 실시간으로 동시 수용합니다.

1). 현실 IoT 센서 텔레메트리 (시계열 데이터) :

초당 60회 속도로 쏟아지는 모터와 기계의 실시간 온도, 진동(Hz), 유체 흐름 데이터.

2). 비전 및 액추에이터 피드백:

실시간 모니터링 카메라 피드와 다관절 로봇 팔의 현재 6축 토크 및 관절 위치 피드백 데이터.

3). 작업자의 실시간 자연어 프롬프트:

"모터 조립 속도를 15% 가속해라", "사이버펑크 조명으로 변환해라" 등 수시로 유입되는 자연어 명령어.

② 다중 출력 (Multiple Outputs)

수집된 실시간 입력을 분석한 시스템은 가상 공간과 실제 공장 현장에 동시에 다원화된 제어 신호를 출력합니다.

1). 가우시안 세포(Splat Tensors)의 즉각적인 수학적 변형 :

위험이나 과열 센서 신호 감지 시, 해당 위치 가우시안 세포들의 색상(RGB) 및 불투명도 파라미터를 붉은색 발광 텐서로 즉시 변경.

2). WebGL/WebGPU 실시간 디스플레이 출력: 관제실 모니터, 스마트 태블릿, 현장 순찰자의 Apple Vision Pro(WebXR) 등 서로 다른 해상도와 주사율을 가진 다양한 뷰포트 화면으로 실시간 4K 렌더링 화면을 동시 송출.

3). 물리 기기 제어 신호 송출 (PLC/로봇 액션): VLM 두뇌와 PINN 물리 엔진의 예측에 따라 다관절 로봇 팔의 관절 제어(Inverse Kinematics) 궤적을 수정하고, 공장 PLC 컨트롤러에 직접 '냉각수 주입 및 긴급 섯다운' 제어 전기 신호(EtherCAT)를 하달

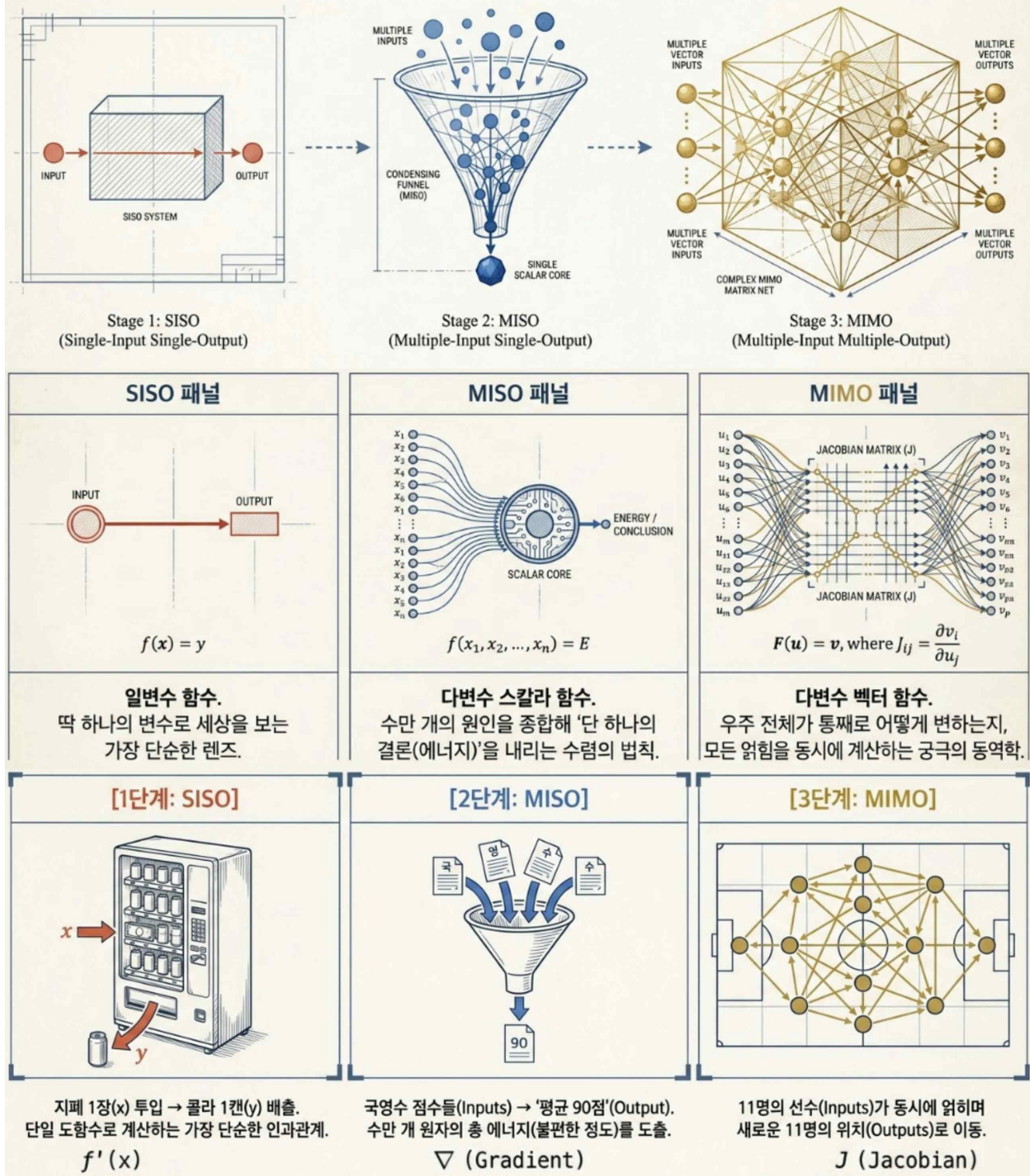
요약: 왜 MISO + MIMO의 결합인가?

■ Trellis 등 생성 AI가 이미지들을 모아(다중 입력) 정적인 3D 가우시안 공간(단일 출력)을 만드는 과정은 MISO입니다.

■ 하지만 이렇게 조각된 정적 3D 공간을 현업에 그대로 쓸 수는 없습니다. **Spark.js 공간 지능 OS**는 이 MISO 기반 3DGS 데이터에 실시간 감각 입력과 다양한 장비 제어 출력을 연결하여, 최종적으로 공장 전체를 유기적으로 움직이는 **완벽한 초저지연 MIMO 자율 제어 시스템**으로 승화시킵니다.

따라서, "MISO 방식으로 빌드된 3DGS 입체 세포 공간에 MIMO 형태의 실시간 양방향 제어 파이프라인을 결합하여 가동하는 시스템"으로 이해하시는 것이 가장 완벽한 기술적 해답입니다.

형태의 확장, 우주의 법칙을 담다



코드 3-3. 물리 엔진 (원문 그대로)

```

1 import { constructGrid, constructSpherePoints } from "@sparkjsdev/spark";
2
3 // [1] Dyno 물리 엔진: WebGL 셰이더 없이 3DGS 점들을 실시간 수학 물리 필드로 변형
4 function createProgrammablePhysicsField() {
5   const physicsSplatMesh = new SplatMesh({
6     constructSplats: (splats) => constructGrid({
7       splats,
8       extents: new THREE.Box3(new THREE.Vector3(-2.5, -1.5, -2.5), new THREE.Vector3(2.5, 1.5, 2.5)),
9       stepSize: 0.45,
10      pointRadius: 0.04
11    })
12  });
13  return physicsSplatMesh;
14 }
15
16 // [2] Three.js 일반 3D 폴리곤 로봇 팔과 3DGS 실사 공간의 Z-Buffer 하이브리드 결합
17 const robotArmGroup = new THREE.Group();
18 const joint1 = new THREE.Mesh(new THREE.SphereGeometry(0.11, 16, 16), new
THREE.MeshStandardMaterial({ color: 0xff4b63, metalness: 0.8 }));
19 const link1 = new THREE.Mesh(new THREE.CylinderGeometry(0.055, 0.055, 0.75, 16), new
THREE.MeshStandardMaterial({ color: 0x94a3b8, metalness: 0.8 }));
20 link1.position.set(0.28, 0.55, 0); link1.rotation.z = -Math.PI / 4;
21 robotArmGroup.add(joint1); robotArmGroup.add(link1);
22 scene.add(robotArmGroup);
23
24 // [3] PINN 기반 열역학 자율 복구 루프
25 function triggerAutonomousPhysicalRecovery() {
26   appendXaiLog('[PINN] 나비에-스토크스 유체역학 연산: 냉각 유체 순환 속도 20% 보정 계산');
27   appendXaiLog('[PAI/LAM] 다관절 로봇 6축 토크 피드백 제어: 모터 틱새 0.01mm 재정렬 완료');
28   document.getElementById("gauge-temp").textContent = "41.2 °C (정상)";
29   document.getElementById("gauge-vibe").textContent = "12.1 Hz (안정)";
30 }

```

표 3-3. 코드 3-3 해설 및 API 대조

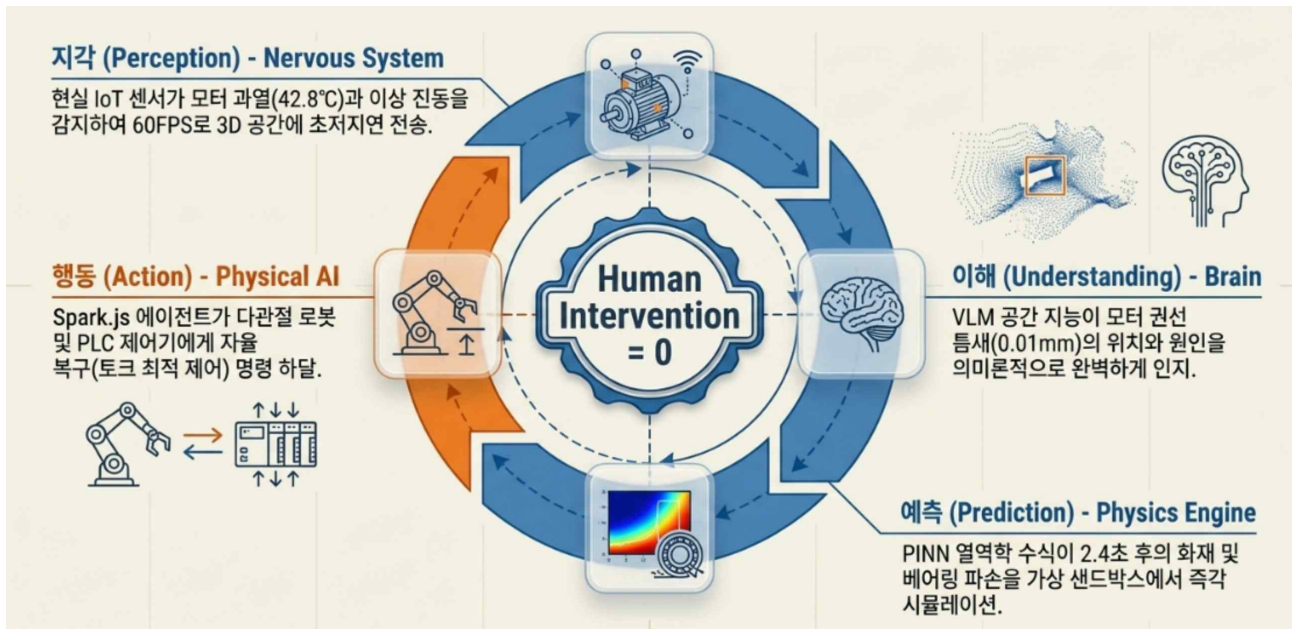
라인	동작	대조·비고	등급
1	constructGrid, constructSpherePoints import	constructSplats 콜백용 절차적 생성 도우미. 본 발췌본에서 constructSpherePoints는 미사용(다른 모듈에서 사용)	[공식]
4~14	격자 형태의 프로그래머블 스 플랫 필드 생성	SplatMesh의 constructSplats 옵션은 공식 절차적 생성 경 로. 이후 Dyno(셰이더 그래프)로 GPU에서 변형	[공식]
17~22	Three.js 메시(로봇 팔)와 3DGS 씬 합성	Spark.js는 스플랫과 메시지를 하나의 THREE.js 파이프라인에 서 깊이 정합하여 융합	[공식]
25~30	PINN 결과를 관제 UI에 반영	PINN 자체는 Tier 1 파이썬/CUDA 서비스로 동작, 결과만 WebSocket으로 수신하는 것이 상용 구조(4.7)	[목표]

3.3-보강. PINN 예측 엔진의 상용 구성

원문의 "2.4초 후 화재/단선 시뮬레이션"은 PINN이 열역학·유체역학 미분방정식(예: 열전도, 나비에-스토크스 근사)을 센서 경계조건으로 풀어 단기 미래 상태를 예측하는 절차입니다. 상용 구성에서는 (i) 설비별 PINN 모델을 Tier 2에서 사전 학습, (ii) Tier 1에서 센서 값으로 초기·경계 조건을 갱신하며 추론, (iii) 예측 지평(horizon)과 신뢰 구간을 함께 출력하여 XAI 로그에 남깁니다. "2.4초"는 특정 시나리오의 예측 지평 예시이며, 실제 값은 설비·샘플링 주기에 따라 달라지므로 [?]로 표기하고 파일럿에서 실측합니다.

3.4. 4단계 자율 폐쇄 루프 (Closed-Loop) 완결 프로세스

1. [신경계] IoT 센서가 모터 과열(43.5°C)과 이상 진동을 감지하여 3D 공간에 전송
2. [두 뇌] VLM 공간 지능이 모터 권선 틈새(0.01mm)의 위치와 원인을 의미론적으로 지각
3. [물리엔진] PINN 열역학 수식이 2.4초 후의 화재/단선을 가상 샌드박스에서 시뮬레이션
4. [행 동] 피지컬 AI(PAI)가 다관절 로봇에게 최적 토크 제어 명령을 내려 자율 복구



3.5. 성능 벤치마크 방법론

원문에 제시된 성능 수치를 [실측] 등급으로 승격하기 위한 측정 조건입니다. 부록 B에 기록 양식이 있습니다.

표 3-4. 벤치마크 항목·조건

항목	원문 수치	측정 조건 고정 항목	측정 도구	현재 등급
정렬/LoD 갱신 시간	100만 점 1ms	스플랫 수(100만/300만), GPU, 브라우저 버전, 카메라 이동 패턴	performance.now() + WebWorker 타임스탬프	[?]->[실측]
압축률	SPZ 95%	원본 .ply 크기 대비 .spz/.sog/.rad 크기, SH 차수	파일 크기 비교	[?]->[실측]
렌더 FPS	4K 60/120	해상도, 스플랫 예산(500K~2.5M), 메시 객체 수, 모니터 주사율	브라우저 프레임 타이밍	[?]->[실측]
IoT 동기화	60FPS	센서 샘플링 주기, 브리지 지연, 갱신 배치 크기	송수신 타임스탬프 차	[?]->[실측]
제어 지연	<20ms	센서 이벤트 → 제어 명령 송출까지 (Tier 1 내부)	PTP 동기 로그	[목표]->[실측]

3.6. 폐쇄 루프 시퀀스 및 지연 예산

그림 3-3. 폐쇄 루프 시퀀스

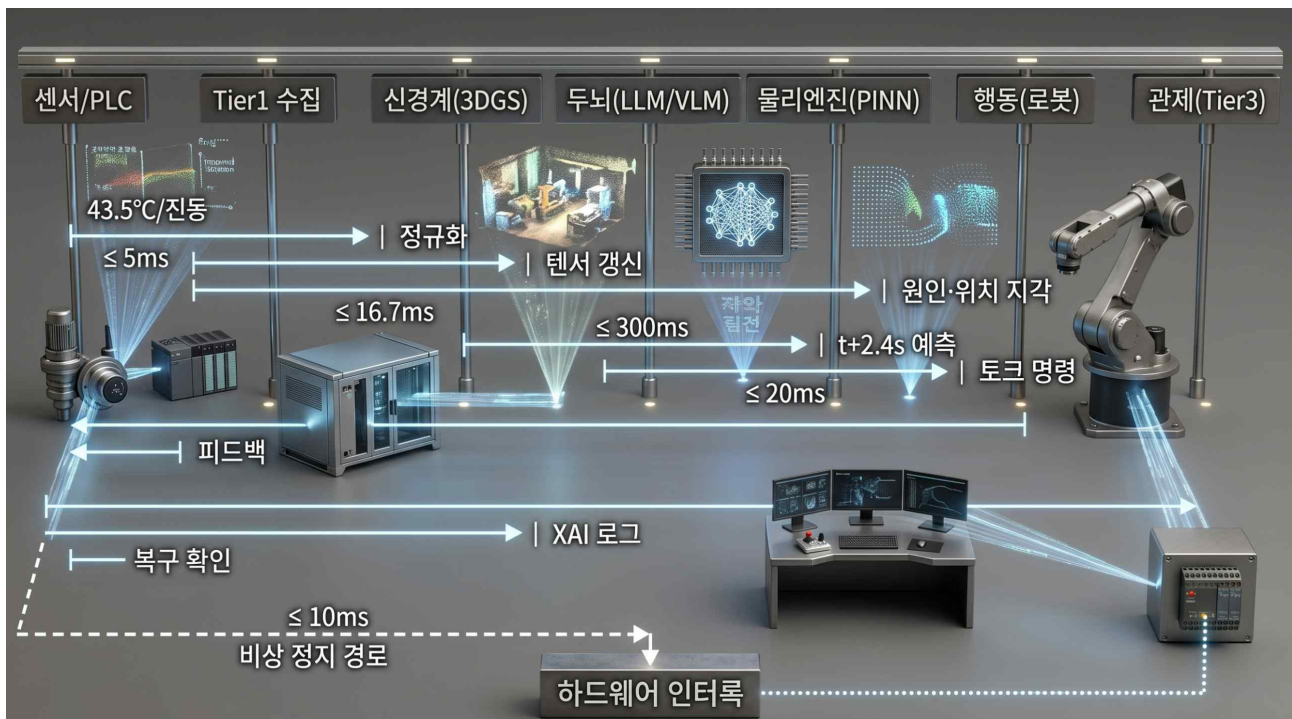


표 3-5. 단계별 지연 예산 [목표]

단계	구간	예산	비고
1	센서 → Tier 1 수집	≤ 5ms	10GbE/EtherCAT, PTP 시간 동기
2	수집 → 3DGS 텐서 갱신·렌더	≤ 16.7ms (60FPS)	관제 시각화 경로, 제어 경로와 분리
3	수집 → 두뇌 지각 (이상 이벤트 시)	≤ 500ms	VLM 추론, 비상 정지는 이 단계를 우회(7장)
4	두뇌 → PINN 예측	≤ 300ms	사전 학습 모델 추론
5	예측 → 제어 명령 송출	< 20ms	규칙 기반 안전 필터 통과 후
-	비상 정지 경로 (센서 임계 초과 → 인터록)	≤ 10ms	AI 경로와 무관하게 PLC 하드 인터록으로 보장

3.7. 공간 취득·갱신 파이프라인

원문은 공장이 이미 3DGS로 존재한다고 전제합니다. 실제 도입에서 가장 먼저 발생하는 작업은 현장을 3DGS로 만드는 것입니다.

표 3-6. 취득·갱신 파이프라인

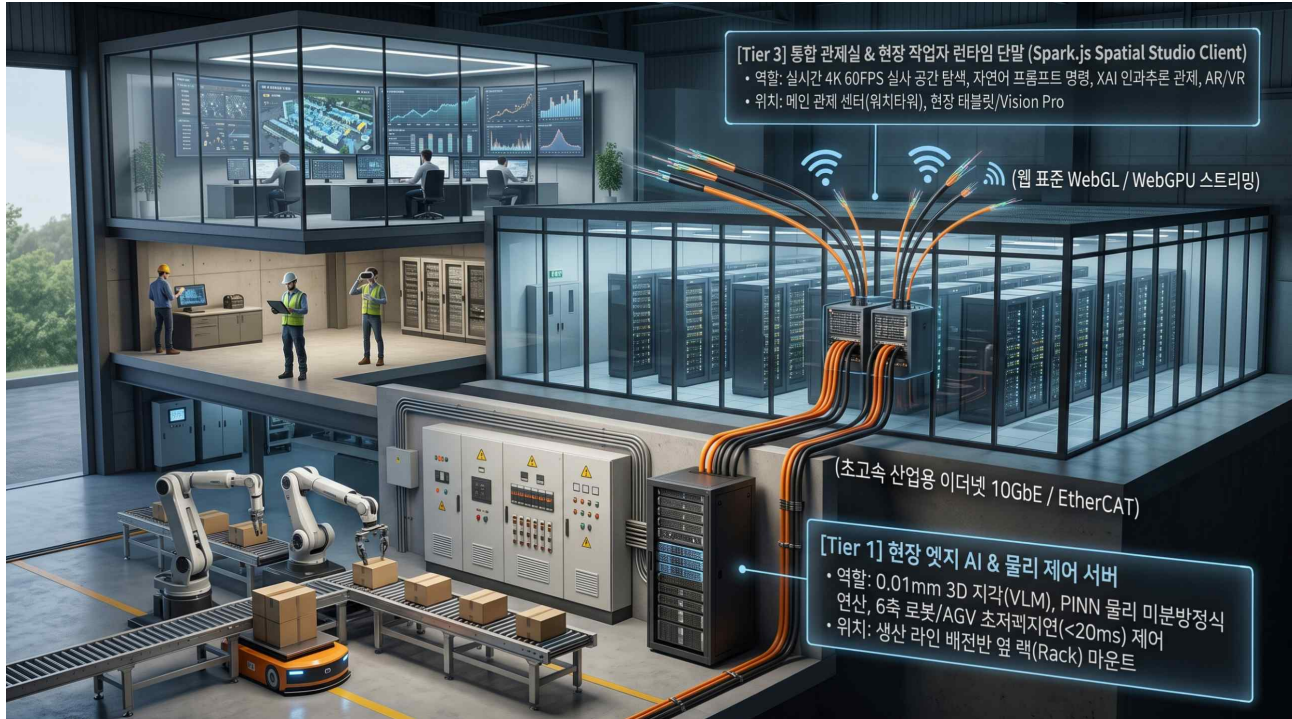
단계	작업	장비/도구	산출물	소요(1개 라인 기준) [목표]
1	촬영 계획·안전 승인	동선 설계, 작업 시간 협의	촬영 계획서	1~2일
2	촬영	고해상 카메라/짐벌, 드론(육외), 선택적 LiDAR(스케일 기준)	이미지 수천 장 + 스케일 기준점	0.5~1일
3	SfM·3DGS 학습	COLMAP 등 + 3DGS 트레이너 (Tier 2 GPU)	.ply → .spz/.sog/.rad	수 시간~1일
4	스케일·좌표 정합	기준점·LiDAR로 실측 단위 정합, 공장 좌표계 변환	좌표 변환 행렬	0.5일
5	LoD·스트리밍 패키징	Spark LoD 도구(Tiny-LoD/Bhatt-LoD)	스트리밍 가능 스플랫 트리	수 시간
6	객체 라벨링	설비·구역 ID 부여, VLM 접지용 라벨 맵	라벨 맵	1~2일
7	갱신	설비 변경·이설 시 부분 재촬영·재학습	버전별 씬	변경 시

- 데이터 용량: 라인 1개당 원본 이미지 수십 GB, 3DGS 결과 수백 MB~수 GB 수준 (파일럿에서 확정)
- 갱신 주기: 설비 배치 변경 시 즉시, 그 외 분기별 검증 촬영 권장
- 보안: 촬영 데이터는 고객사 자산이며 7장 데이터 주권 조항에 따라 취급

제 4장. 산업용 공간 지능 시스템 3계층(3-Tier) 아키텍처 및 인프라 설계

앞서 설명한 초고성능 Spark.js 소프트웨어 엔진을 전기차 공장, 스마트 항만 등 미션 크리티컬 현장에서 24시간 365일 초저지연(<20ms)으로 가동하기 위해서는 3-Tier 엔터프라이즈 하드웨어 인프라가 반드시 뒷받침되어야 합니다.

그림 4-1. 3-Tier 아키텍처



산업용 공간 지능 시스템 3 계층(3-Tier) 아키텍처 및 인프라

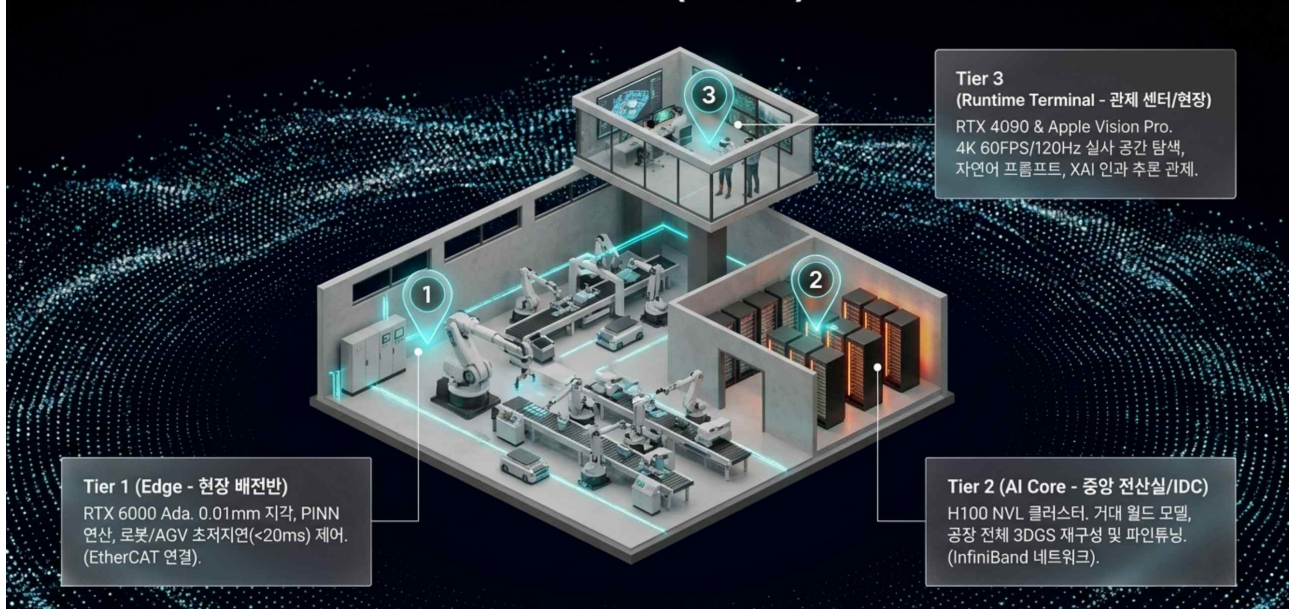


표 4-0. 3-Tier 역할 요약 (그림 4-1의 정식 표 재구성)

Tier	명칭	역할	위치	상위 연결
Tier 1	현장 엣지 AI & 물리 제어 서버	0.01mm 3D 지각(VLM), PINN 연산, 6축 로봇/AGV <20ms 제어	생산 라인 배전반 옆 랙	10GbE / EtherCAT
Tier 2	중앙 거대 월드 모델 & 3DGS 학습 서버	공장 전체 3DGS 재구성, 로봇 RL, 디지털트윈 VAE 동기화	중앙 전산실(IDC) 또는 프라이빗 AI 클라우드	WebGL/WebGPU 스트리밍
Tier 3	통합 관제실 & 현장 런타임 단말	4K 60FPS 실사 탐색, 자연어 명령, XAI 관제, AR/VR	관제 센터, 현장 태블릿 /Vision Pro	-

4.1. [Tier 1] 엣지 서버 (현장 물리 제어기)

- GPU: NVIDIA RTX 6000 Ada (48GB) 1~2장 (ECC VRAM 탑재로 무중단 가동, PINN 연산 동시 처리)
- CPU: AMD EPYC 9354 (32코어) (IoT 센서 데이터 파싱 및 저지연 IPC)
- RAM/Storage: 256 GB DDR5 ECC / PCIe 4.0 U.2 NVMe 3.84TB x 2 (RAID 1)
- Network: 10GbE SFP+ / EtherCAT PCIe (다관절 로봇 PLC 직결용)

4.2. [Tier 2] 중앙 AI 코어 (거대 월드 모델)

- GPU: NVIDIA H100 NVL (94GB) 2~4장 클러스터 (대규모 멀티모달 VLM, 월드 모델 파인튜닝)
- CPU: AMD EPYC 9654 (96코어 듀얼) (페타바이트급 산업 훈련 데이터셋 전처리)
- RAM/Storage: 512 GB ~ 1 TB DDR5 ECC / PCIe 5.0 NVMe 15.36TB x 4 (RAID 10)
- Network: NVIDIA Quantum-2 400Gb/s InfiniBand (분산 학습 병목 제로)



4.3. [Tier 3] 메인 통합 관제 워크스테이션 (런타임 클라이언트)

- GPU: NVIDIA RTX 4090 (24GB) (300만 3DGS 점군 및 로봇 메시 Z-Buffer 합성 4K 120Hz 렌더링)
- CPU: Intel Core i9-14900K (WebGL 브라우저 단일 코어 렌더링 스레드 극대화)
- RAM: 128 GB DDR5 (WASM 정렬 메모리 확보)
- Display/XR: 4K 144Hz 트리플 모니터 / Apple Vision Pro (WebXR 현장 관제용)



4.3-보강. 렌더링 수치의 정의 (60FPS / 120Hz / 144Hz)

원문에는 Tier 3에 대해 세 가지 수치가 등장합니다. 이는 서로 다른 층위의 값이며 아래와 같이 정의합니다.

수치	출처	정의	등급
4K 60FPS	그림 4-1 Tier 3 역할	최소 보증치 - 300만 점 + 메시 합성 시 관제 화면이 유지해야 하는 하한	[목표]
4K 120Hz	4.3 GPU 항목	설계 목표치 - RTX 4090급에서 스플랫 예산 조정 시 도달 목표	[목표]
4K 144Hz	4.3 Display 항목	패널 사양 - 모니터 주사율이며 렌더 성능 보증치가 아님	[공식]

4.4. 네트워크 토폴로지 및 IoT 데이터 경로

그림 4-2. OT/IT 분리 토폴로지

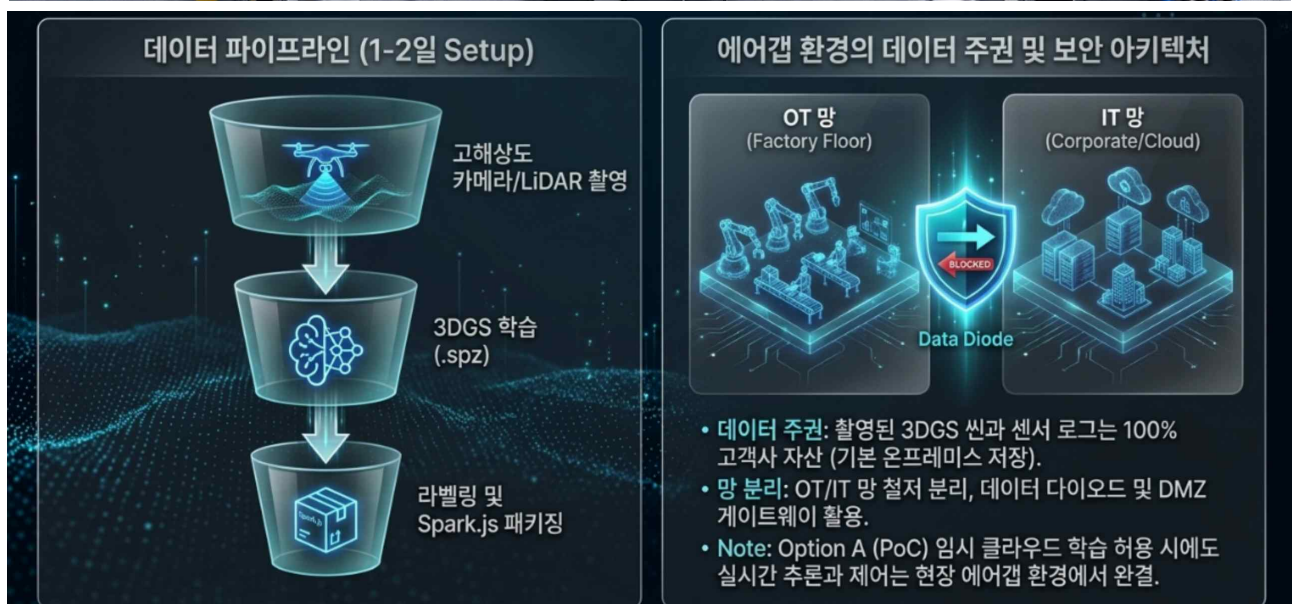
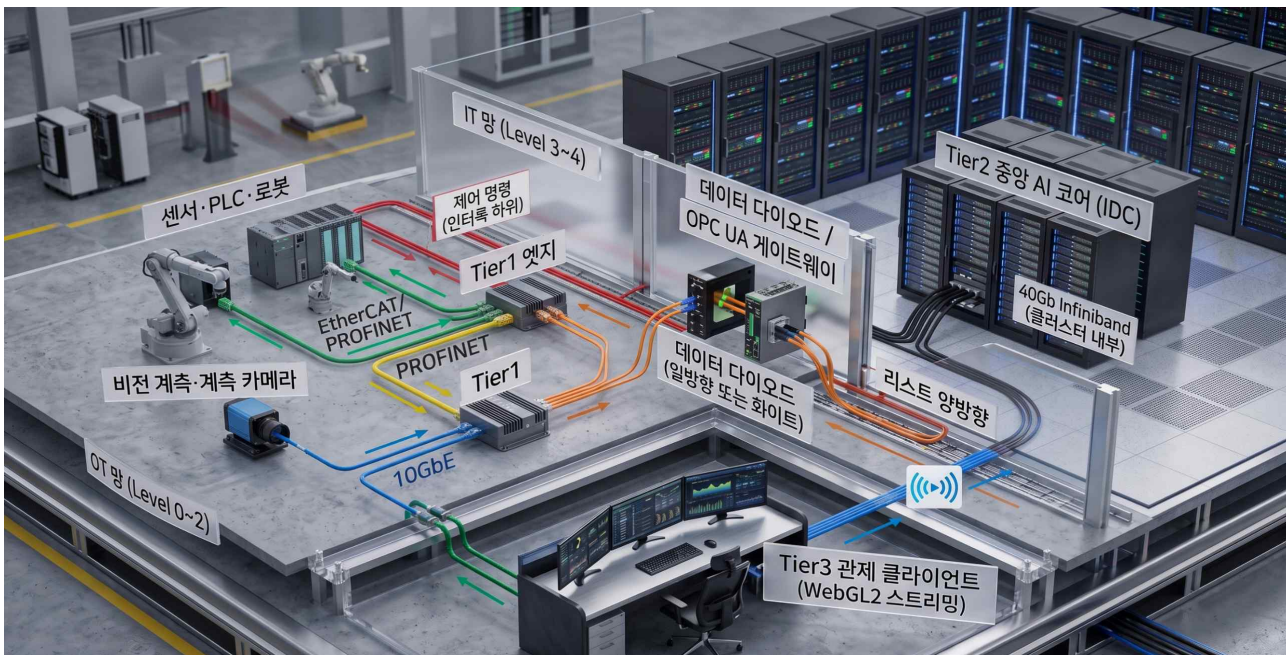


표 4-1. 데이터 경로·프로토콜·시간 동기

경로	프로토콜	주기/대역	시간 동기	비고
센서 → Tier 1	MQTT(경량 센서), OPC UA(설비), EtherCAT(로봇/PLC)	센서 10~1000Hz, 3DGS 갱신 60Hz 배치	PTP(IEEE 1588)로 Tier 1·PLC 동기	원문 "60FPS 동기화"는 갱신 배치 주기
Tier 1 → 로봇	EtherCAT 사이클 / 로봇 제어 API	1~4ms 사이클	PTP	비상 정지는 PLC 하드 인터록 우선
Tier 1 ↔ Tier 2	OPC UA 게이트웨이 경유, 파일 동기(모델·썬)	비실시간 배치	-	데이터 다이오드 적용 시 일방향
Tier 2 → Tier 3	HTTPS + WebSocket, 스플랫 스트리밍(.rad)	초기 로드 수백 MB, 이후 증분	-	관제망은 IT 망 내 분리 VLAN
Tier 3 → Tier 1 (명령)	WebSocket → 명령 게이트웨이 → OPC UA	이벤트 기반	-	명령은 화이트리스트·서명 검증 후 전달

4.5. 보안 · 가용성 · Fail-safe

표 4-2. 보안·가용성 설계

항목	설계	표준/근거
망 분리	OT/IT 분리, DMZ에 OPC UA 게이트웨이 또는 데이터 다이오드, Tier 1은 인터넷 비연결(에어갭 가능)	IEC 62443-3-3 구역·통로 개념
접근 제어	관제 명령은 역할 기반(RBAC) + 명령 서명, 감사 로그 원장	IEC 62443-4-2 기술 요구
가용성	Tier 1 이중화(액티브-스탠바이, ECC VRAM), Tier 2는 노드 장애 허용, 스토리지 RAID 1/10(원문)	HA 설계 관행
Fail-safe	AI 경로 장애·지연 초과 시 로봇은 마지막 안전 상태 유지 또는 PLC 인터록 정지; 폐쇄 루프는 "자율 등급"에 따라 승인 모드로 강등(7장)	ISO 13849 안전 관련 제어
백업·복구	썬·모델·원장 일 단위 스냅샷, 복구 목표 시간(RTO)·시점(RPO) 계약 명시	BCP
공급망	오픈소스 구성요소 SBOM 제공, 서명된 빌드 배포	SBOM 관행

4.6. 전력 · 냉각 · 랙 산정

표 4-3. 물리 설비 산정 (옵션 B 기준, 제조사 공개 TDP 근거 [공식/목표])

Tier	구성	예상 소비 전력	랙 공간	냉각
Tier 1	RTX 6000 Ada 2장(각 300W) + EPYC 9354 + 주변	약 1.2~1.5kW	2U~4U	현장 랙 공랭, 분진 필터·온도 관리 필수
Tier 2	H100 NVL 4장(각 최대 400W급) + 듀얼 EPYC 9654 + NVMe + IB	약 3.5~4.5kW/노드	4U~8U	IDC 열통로 격리, 필요 시 액체 냉각 검토

Tier	구성	예상 소비 전력	랙 공간	냉각
Tier 3	RTX 4090(450W) + i9-14900K 워크스테이션 3대	약 0.8~1.0kW/대	데스크사이드	일반 사무 환경
합계	-	약 8~11kW (UPS 용량 산정 기준)	-	-

4.7. 소프트웨어 스택

계층	Tier 1	Tier 2	Tier 3
OS	Ubuntu LTS (실시간 커널 옵션)	Ubuntu LTS	Windows/Linux + 최신 Chromium 계열 브라우저
GPU	CUDA + TensorRT (LLM/VLM/PINN 추론)	CUDA + NCCL (분산 학습), 3DGS 트레이너	WebGL2(WebGPU 선택) 드라이버
런타임	추론 서버, PINN 서비스, 명령 게이트웨이, OPC UA 클라이언트, EtherCAT 마스터	학습 파이프라인, 실행 빌드(LoD 패키징), 모델 레지스트리	Spark.js 2.x + THREE.js 앱 (Spatial Studio)
컨테이너/오케스트레이션	Docker(단일 노드) 또는 경량 k3s	Kubernetes/Slurm	-
모니터링	Prometheus/Grafana, 원장 뷰어	동일 + 학습 대시보드	브라우저 성능 오버레이

4.8. 하드웨어 세대 교체 대응 - 동급 대체 모델표

원문 사양은 유지합니다. 다만 구매 시점(2026년 하반기 이후)에는 후속 세대가 유통 중이므로, 동급 대체 모델을 병기하여 조달 유연성을 확보합니다.

표 4-4. 대체 모델표 [공식 제조사 자료 기준, 가격·재고는 견적 시 재확인 [?]]

Tier	원문 모델	동급/후속 대체 후보	대체 시 고려 사항
Tier 1	RTX 6000 Ada (48GB)	RTX PRO 6000 Blackwell (96GB GDDR7; Workstation 600W / Max-Q 300W / Server Edition)	96GB로 VLM-PINN 동시 상주 여유 증가. Max-Q(300W)가 원문 RTX 6000 Ada와 동일 전력·블로워 구조로 랙 교체 용이
Tier 2	H100 NVL (94GB)	H200 NVL / B200 계열, 또는 RTX PRO 6000 Blackwell Server Edition 다중 구성	메모리·대역폭 상향, 공급 리드타임·전력 요구 확인
Tier 3	RTX 4090 (24GB)	RTX 5090 (32GB) 등 후속 소비자 GPU	WebGL2 성능은 세대 상향으로 여유; 전력·케이스 규격 확인
Tier 3	i9-14900K	후속 데스크톱 CPU	브라우저 단일 스레드 성능 우선

4.9. 런타임 전제 조건 (브라우저·XR)

항목	전제	비고
브라우저	WebGL2 지원 최신 Chromium 계열(권장) / Firefox / Safari	Spark.js는 98%+ WebGL2 기기 대상 [공식]; WebGPU는 선택
메모리	탭당 GPU 메모리 상한을 고려한 스플랫 예산 (500K~2.5M) 설정	LoD 예산으로 GPU 부하 일정 유지 [공식]
XR	Apple Vision Pro 등 WebXR 지원 기기	WebXR 세션 제약(프레임 예산·입력)은 파일럿에서 확인
오프라인	관제 클라이언트는 사내망 전용, 외부 CDN 미사용(스플랫·라이브러리 자체 호스팅)	에어갭 환경 대응
보안 설정	관제 앱은 HTTPS·인증 필수, 명령 API는 서명·RBAC	4.5

제 5장. 예산 및 도입 규모별 전략 구성

5.1. 옵션 A: 공장 실증 및 파일럿 (PoC / MVP)

- 대상: 1개 조립 라인 (예: 전기차 모터 권선 공정 실증)
- 구성: 관제 겸용 고성능 엣지 워크스테이션 (i9-14900K + RTX 4090 2Way) + 중앙 AI 클라우드 렌탈
- **예산 인프라 비용: 약 1,800만 ~ 2,500만 원**

5.2. 옵션 B: 정식 상용화 엔터프라이즈 패키지 (Full Commercial)

- 대상: 완성차 공장, 항만 전체의 24/7 자율 폐쇄 루프 무인 관제
- 구성: 완벽한 분리형 3-Tier 아키텍처 (Edge RTX 6000 Ada + Core H100 클러스터 + Terminal RTX 4090 3대)
- **예산 인프라 비용: 약 1억 5,000만 ~ 2억 5,000만 원**

5.3. 산정 근거표 및 3년 TCO

원문 금액은 유지하고, 그 산정 근거를 부품 단위로 투명화합니다. 단가는 견적 시점·환율에 따라 변동하므로 모두 [?]이며, 계약 전 정식 견적으로 대체됩니다. 특히 옵션 B는 H100 NVL 4장 구성 시 GPU 비용만으로 원문 상한에 근접할 수 있어, **재산정 결과를 병기**합니다.

표 5-1. 옵션 B 부품별 산정 근거 (범위 표기, 단가)

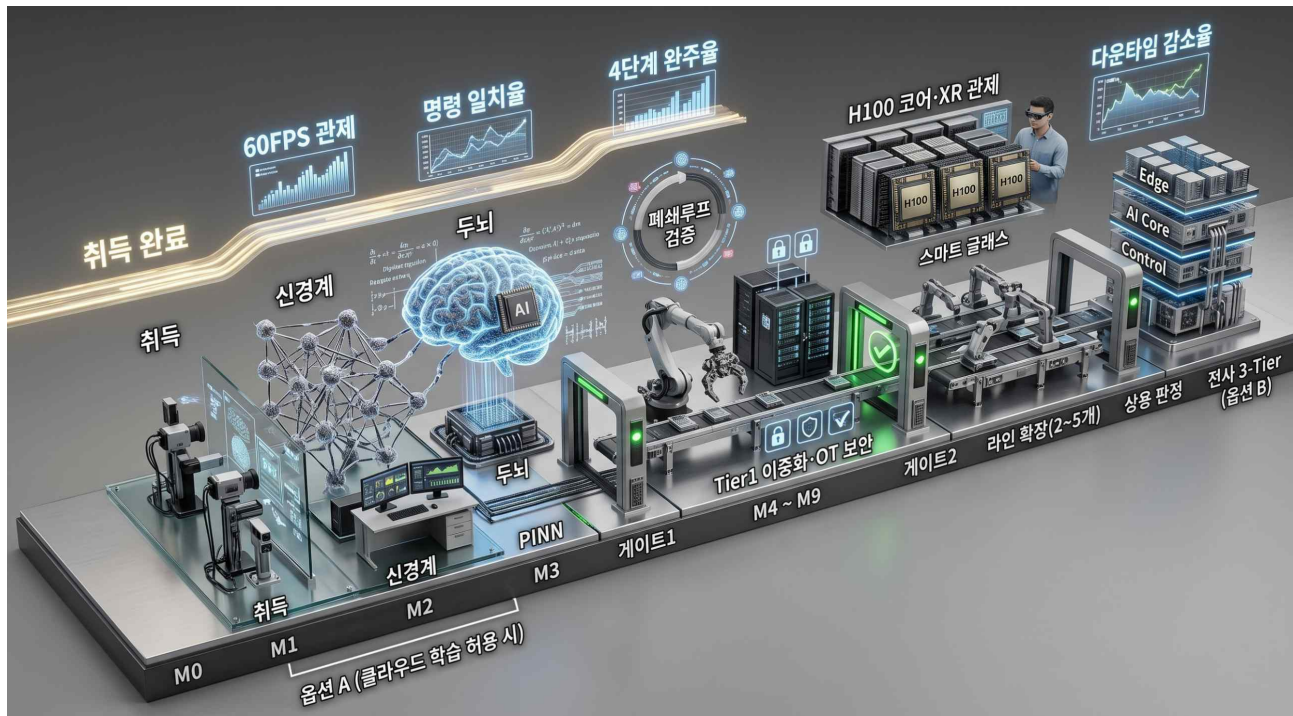
구분	항목	수량	비고
Tier 1	RTX 6000 Ada 또는 RTX PRO 6000 Blackwell	1~2	ECC, 현장 랙
Tier 1	EPYC 9354 서버 본체 + 256GB ECC + NVMe 3.84TB×2 + 10GbE/EtherCAT	1	이중화 시 ×2
Tier 2	H100 NVL 94GB	2~4	재산정 시 가장 큰 변동 요인
Tier 2	듀얼 EPYC 9654 서버 + 512GB~1TB ECC + NVMe 15.36TB×4 + IB 400Gb	1	IB 스위치 포함 여부 확인
Tier 3	RTX 4090 + i9-14900K + 128GB 워크스테이션	3	4K 144Hz 트리플 모니터 포함
Tier 3	Apple Vision Pro	1~2	WebXR 현장 관제
공통	랙·UPS·네트워크 스위치·케이블	1식	4.6 전력 산정 기준
원문 합계			약 1억 5,000만 ~ 2억 5,000만 원 (원문 유지)
재산정			H100 NVL 4장 구성 시 원문 상한 초과 가능성 있음 - 2장 구성 또는 대체 GPU(4.8) 적용 시 원문 범위 내 수렴 예상

표 5-2. 3년 TCO 항목 (금액은 전적 시 확정 [?])

항목	옵션 A	옵션 B	비고
하드웨어 구매	원문 1,800만~2,500만	원문 1.5억~2.5억	표 5-1
클라우드 렌탈	파일럿 기간 GPU 시간당 과금	해당 없음(온프레미스)	옵션 A만
전력	약 1kW급	약 8~11kW급 (4.6)	전기요금 단가 × 가동 시간
유지보수	제조사 보증	3년 지원 계약(부품·선지원)	SLA(5.8)
소프트웨어 라이선스	공급사 계층 라이선스(5.8)	동일 + 좌석/사이트 확장	Spark.js 런타임은 MIT
인력·통합(Sler)	공급사 파일럿 패키지	현장 통합·교육·검수	5.7 책임 경계
취득·갱신(3.7)	1개 라인 촬영·학습	전 구역 + 분기 갱신	

5.4. 단계별 로드맵 및 마일스톤 KPI

그림 5-1. 로드맵



단계	기간	핵심 산출물	게이트 KPI [목표]
1. PoC (옵션 A)	M0~M3	1개 라인 3DGS 씬, 관제 화면, 폐쇄 루프 시연 10회	6.4 파일럿 성공 기준 충족

단계	기간	핵심 산출물	게이트 KPI [목표]
2. 라인 확장	M4~M9	Tier 1 이중화, OT 보안 적용, 2~5개 라인	이상 선행 감지율, 오작동 0건 유지
3. 전사 상용화 (옵션 B)	M10~M18	3-Tier 완성, H100 코어, XR 관제, 24/7 운영	다운타임 감소율, SLA 달성

5.5. 리스크 매트릭스

#	리스크	구분	가능성	영향	대응
R1	GPU 공급 리드타임·가격 변동	공급망	중	중	대체 모델표(4.8), 2장 구성 우선
R2	3DGS 취득 품질(반사·이동 물체)	기술	중	중	촬영 가이드, 라인 정지 시간대 촬영, 부분 재촬영
R3	LLM 오해석으로 인한 잘못된 제어	안전	낮~중	높	규칙 기반 안전 필터, 승인 모드, 인터록 (7장)
R4	OT 망 보안 심사 지연	운영	중	중	IEC 62443 체크리스트 사전 제출(7장)
R5	PLC/로봇 인터페이스 미문서화	운영	중	높	전제 조건(1.5) 확인, 통합사 조기 참여
R6	브라우저·WebGL 드라이버 호환	기술	낮	중	관제 단말 표준 이미지 고정(4.9)
R7	오픈소스 런타임 버전 변화	기술	중	낮	버전 고정·자체 호스팅, 업그레이드 검증 절차
R8	KPI 정의 불일치로 인한 검수 분쟁	계약	중	중	6장 측정 정의를 계약 부속서로

5.6. 공급사 소개 (식별 정보 마스킹)

※ 회사명·특허번호·인명 등 구체 식별 정보는 계약 협의 단계 전까지 ★로 마스킹하며, 본 절은 일반적 소개로 한정합니다.

항목	내용
공급사	★★★ (국내 AI 소프트웨어 개발 기업)
주력 분야	온프레미스·폐쇄망 환경에서 동작하는 AI 시스템, 3D 공간 지능 런타임, 자율 소프트웨어 개발 체계
지식재산	자율·자기진화형 소프트웨어 개발 시스템 관련 등록 특허 제★★-★★★★★★호 보유 (본 플랫폼의 두뇌·XAI 계층 설계와 연관)
핵심 역량	3DGS 런타임 통합, 온프레미스 LLM/VLM 추론, 물리 기반 예측 모델, 산업 제어 인터페이스 연동
팀 구성	대표(★★★), 기술 책임자(★★★), 연구·개발 인력 ★명, 산업 설계·현장 경험 보유
참여 실적	★★★ (계약 협의 시 레퍼런스 제공)
지원 체계	파일럿 기간 상주 지원, 상용 단계 SLA 기반 원격·현장 지원(5.8)

5.7. 책임 경계 (공급 범위 / 고객 범위 / 공동 범위)

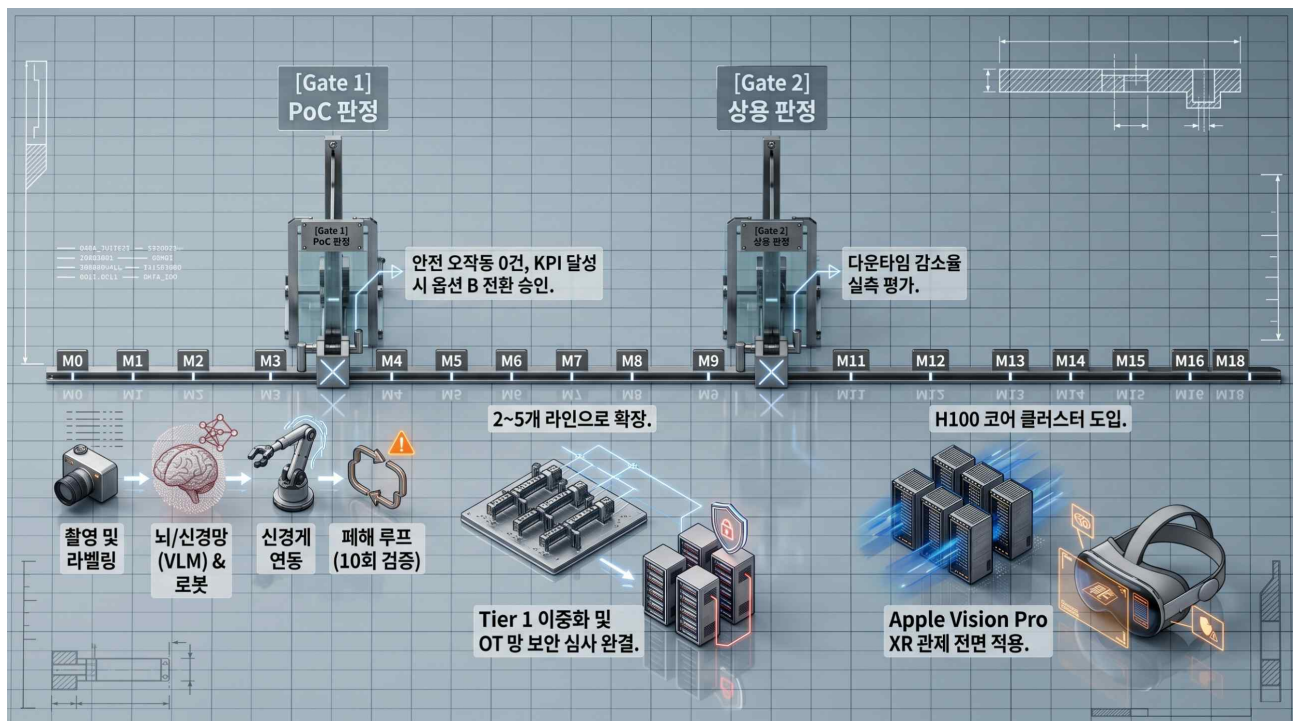
영역	공급사 ★★★	고객사	공동
소프트웨어	런타임 통합, 두뇌·물리 엔진, 관제 앱, XAI 원장	기존 MES/SCADA 연계 요구사항 제공	인터페이스 명세 합의
하드웨어	사양 제안, 설치 검수 지원	조달·설치·전력·랙	수용성 시험
네트워크·보안	보안 설계안, SBOM, 체크리스트	OT/IT 망 구성, 방화벽·DMZ 운영	보안 심사 대응
데이터	취득 계획·학습·라벨링 수행	촬영 허가, 센서·PLC 데이터 접근 권한	데이터 주권 계약(7장)
로봇·PLC 연동	제어 명령 게이트웨이, 안전 필터	로봇/PLC 인터페이스 문서·담당자	안전 검증(인터록 시험)
운영	교육, 유지보수(SLA)	일상 운영, 1차 대응	분기 성능 리뷰

5.8. 상업 조건 개요

항목	골격 (세부는 계약서)
라이선스 구조	Spark.js 런타임: MIT 오픈소스(상용 이용 가능) [공식] / 공급사 계층(두뇌·물리 엔진·관제 앱): 사이트 라이선스 + 연 구독(업데이트·지원 포함) 또는 영구 + 유지보수
확장 단가	라인 추가·사이트 추가 시 단계별 단가, 관제 좌석 수 기준 옵션
SLA	가용성 목표(예: 99.5% 이상), 장애 등급별 응답·복구 목표 시간, 정기 점검
지식재산	공급사 기존 IP는 공급사 귀속, 고객 데이터·센은 고객 귀속, 공동 개발 결과물은 별도 합의
검수	6장 KPI 측정 정의에 따른 단계별 검수, 파일럿 성공 시 옵션 B 전환 조건
탈출 조항	파일럿 실패 시 데이터·센 반환 및 파기 절차, 잔여 하드웨어는 고객 자산

5.9. PoC 실행 계획 (옵션 A, 3개월)

주차	활동	공급사 제공	고객사 제공	산출물
W1~2	착수, 대상 공정·센서 목록 확정, 촬영 계획	착수 문서, 촬영 가이드	현장 접근, 담당자, 안전 교육	실행 계획서
W3~4	촬영·3DGS 학습·라벨링	취득·학습 수행	라인 정지 시간대 협조	3DGS 씬 v1
W5~6	신경계·관제 구축, IoT 연동	워크스테이션 설치, 브리지	센서·PLC 데이터 접근	60FPS 관제 화면
W7~9	두뇌·PINN 통합, 명령 세트 정의	온프레미스 추론, 안전 필터	표준 명령 20종 검토	명령 일치율 리포트
W10~11	폐쇄 루프 시나리오 시험(승인 모드)	시나리오 주입·로그	안전 담당 입회	4단계 완주 로그 10회
W12	검수판정 회의, 옵션 B 전환 제안	결과 보고서, 재산정 건적	판정	게이트1 판정서



5.10. 보안 등급 단계화 (클라우드 → 온프레미스)

옵션 A의 "중앙 AI 클라우드 렌탈"과 4.5의 에어갭 설계는 모순이 아니라 단계입니다. 파일럿에서는 **학습 (3DGS·PINN 사전 학습)만** 클라우드 GPU를 임시 사용하고, 실시간 추론·제어 경로는 처음부터 현장 워크스테이션(온프레미스)에서 수행합니다. 클라우드로 보내는 데이터는 촬영 이미지·센서 로그로 한정하며, 고객사 보안 정책상 반출이 불가하면 파일럿부터 온프레미스 학습(RTX 4090 2Way)으로 대체합니다(기간 연장 가능). 상용(옵션 B)에서는 Tier 2가 IDC/프라이빗 클라우드로 들어오며 전 경로 온프레미스가 됩니다.

제 6장. KPI 및 검증 체계

본 장의 측정 정의는 계약 부속서로 그대로 사용할 수 있도록 작성되었습니다. 각 KPI는 정의·측정 방법·데이터 출처·판정 주기를 갖습니다.

6.1. 검증 원칙

- 모든 수치 판정은 로그·원장·측정 파일 등 **근거 포인터**를 동반해야 하며, 근거 없는 수치는 [?]로 표기하고 판정에서 제외
- 측정은 공급사·고객사 공동 입회 하에 수행하고, 원본 로그는 고객사가 보관
- 파일럿 KPI는 "기술 동작"을, 상용 KPI는 "사업 효과"를 측정

6.2. KPI 정의표

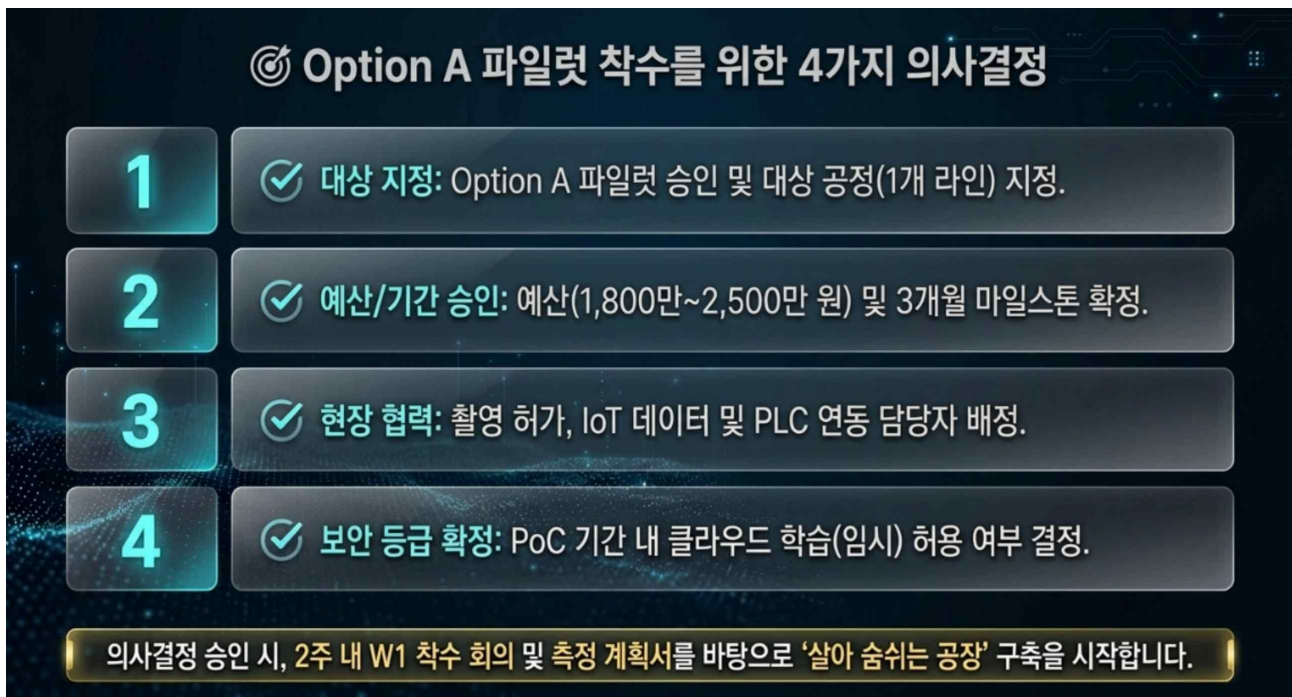
KPI	정의	측정 방법	데이터 출처	목표 [목표]	단계
K1 관제 렌더 FPS	4K 해상도에서 1분 평균 프레임율	브라우저 프레임 타이밍 로그	Tier 3 클라이언트	≥ 60 (목표 120)	PoC
K2 IoT 동기 지연	센서 타임스탬프 → 화면 반영까지	PTP 동기 로그 차	Tier 1/3	≤ 100ms	PoC
K3 명령 의도 일치율	표준 명령 세트 중 의도대로 제어 신호가 생성된 비율	20종×5회 입력, 로그 대조	XAI 원장	≥ 95%, 오작동 0	PoC
K4 접지 정확도	VLM이 지목한 객체가 실제 대상과 일치한 비율	라벨 맵 대조	XAI 원장	≥ 90%	PoC
K5 위치 정밀도	계측 결합 후 위치 오차	기준기(게이지) 대비	계측 장비 로그	장비 사양 이내	PoC
K6 폐쇄 루프 완주율	주입 시나리오에서 4단계 모두 완료된 비율	10회 시험	XAI 원장	100%	PoC
K7 제어 지연	예측 → 제어 명령 송출	PTP 로그	Tier 1	< 20ms	PoC
K8 이상 선행 감지율	실제 고장·정지 이전에 경보가 발생한 비율	고장 이력 대조	MES + 원장	합의 후 확정	확장
K9 비계획 다운타임 감소율	도입 전 기준 대비 감소율	MES 정지 기록	MES	합의 후 확정	상용
K10 가용성	관제·제어 서비스 가동률	모니터링	Prometheus	SLA 값	상용

6.3. 검수 절차

- 측정 계획서 합의(항목·조건·입회자·일정)
- 측정 수행 및 원본 로그 봉인(해시)
- 결과표 작성: 항목 / 측정값 / 근거 포인터 / 판정
- 불합격 항목은 원인·조치·재측정 일정 명시
- 판정 회의 및 게이트 서명

6.4. 파일럿 성공 기준 (게이트 1)

조건	기준
필수	K1, K3, K6, K7 모두 목표 충족, 안전 관련 오작동 0건
권장	K2, K4, K5 목표 충족 또는 원인 규명·개선 계획 합의
전환	위 충족 시 옵션 B 재산정 건적·확장 계획으로 게이트 2 준비



제 7장. 표준 · 규제 · 안전 케이스 · 데이터 거버넌스

7.1. 적용 표준 체크리스트 (고객사 심사팀용)

영역	표준/규정	본 플랫폼 대응	증빙
산업 보안	IEC 62443 (3-3 시스템, 4-2 구성요소)	OT/IT 분리, DMZ 게이트웨이, RBAC, 서명 명령, 감사 원장	보안 설계서, SBOM
로봇 안전	ISO 10218-1/2, ISO/TS 15066(협동 로봇)	AI 제어는 안전 PLC 인터록 하위에 위치, 속도·힘 한계 준수	안전 검증 시험 기록
안전 관련 제어	ISO 13849-1 / IEC 62061	비상 정지·인터록은 AI 경로와 독립	안전 회로 도면
통신	OPC UA(IEC 62541), EtherCAT(IEC 61158)	표준 브리지 사용	인터페이스 명세
기능 안전(설비)	고객사 설비별 규정	자율 등급 단계 적용(7.2)	운영 절차서
개인정보	개인정보보호법(촬영 시 작업자 영상)	촬영 시 비식별 처리, 보관 기간 제한	처리 방침

7.2. 안전 케이스 (Safety Case) – 자율 등급 단계

원문 3.4의 "인간 개입 없이" 수행되는 4단계 폐쇄 루프는 **최종 단계 목표**입니다. 도입은 아래 등급을 순서대로 거치며, 상위 등급 전환은 6장 KPI와 안전 검증 결과로만 결정합니다.

등급	명칭	동작	적용 단계
L0	관제·권고	AI는 이상 감지·예측·권고만, 제어는 사람이 수행	PoC 초기
L1	승인 후 실행	AI가 제어안을 제시, 관제자가 승인하면 실행(Human-in-the-loop)	PoC 후반~확장
L2	조건부 자율	사전 정의된 안전 범위 내 명령은 자율 실행, 범위 밖은 승인 요청	확장~상용
L3	완전 자율 폐쇄 루프	원문 3.4 구조, 인터록·감사 원장 하에서 무개입 수행	상용, 설비별 승인

- 비상 정지·인터록은 항상 AI 경로보다 우선하며 AI 장애 시에도 동작(4.5 Fail-safe)
- 모든 등급에서 제어 명령은 규칙 기반 안전 필터(속도·토크·구역 한계)를 통과해야 함
- 제조물책임: AI 권고·명령의 근거는 원장에 보존되어 사고 조사 시 책임 소재 확인에 사용; 등급별 책임 분담은 계약서에 명시

7.3. 데이터 주권 및 거버넌스

항목	원칙
소유권	촬영 데이터, 3DGS 씬, 센서 로그, 원장은 고객사 자산
저장 위치	기본 온프레미스; 파일럿 클라우드 사용 시 지역·계정·삭제 절차 명시(5.10)

항목	원칙
반출 금지	공장 3DGS 씬은 기밀 자산으로 취급, 공급사 외부 반출 금지, 데모 사용 시 별도 동의
접근	역할 기반, 접근 로그 보존
보관·파기	보관 기간 및 계약 종료 시 파기·반환 절차, 파기 확인서
모델	고객 데이터로 학습된 모델 가중치의 귀속·재사용 범위 합의

제 8장. 🎯 최종 결론 (Executive Summary)

기존의 3D 툴(Trellis 등)이 "과거의 굳어버린 사진"을 3D로 보여주는 데 그친다면, Spark.js는 그 공간을 산업을 움직이는 지능적 신경망으로 진화시킵니다.

Spark.js 기반의 공간 지능 플랫폼은:

1. Rust WASM 신경계로 수백만 개의 3D 가우시안 점을 실시간 감각하고,
2. AI Copilot 두뇌로 인간의 자연어 프롬프트와 물리 공간의 쓰임새(Affordance)를 인지하며,
3. Dyno & PINN 물리 엔진으로 실시간 물리 변형과 다관절 로봇 제어를 수행하는 **"살아 숨 쉬는 인더스트리얼 공간 지능 운영체제"**입니다.

이러한 혁신적인 생명체 런타임을 산업 현장에 안착시키기 위해서는 중앙 H100의 거대 연산력, 현장 RTX 6000의 초저지연 제어력, 관제실 RTX 4090의 시각화 능력이 유기적으로 결합된 3계층 인프라 도입이 필수적입니다. 이를 통해 전 세계 스마트 팩토리는 다운타임 제로에 수렴하는 진정한 **'자율 공간 지능 공장'**으로 도약할 것입니다.

8.1. 의사결정 항목 재확인

10. 옵션 A 파일럿 착수 및 대상 공정 지정
11. 파일럿 예산·기간 승인
12. 고객사 협력 사항(촬영·데이터·PLC 담당자) 승인
13. 보안 등급(파일럿 클라우드 학습 허용 여부) 결정
14. 자율 등급 L0→L1 전환 조건 합의(7.2)

다음 단계: 승인 시 5.9의 W1 착수 회의를 2주 내 개최하고, 정식 견적서(표 5-1 기준)와 측정 계획서(6.3)를 함께 제출합니다.

제 9장. 예상 반론 대응 (FAQ)

Q1. 웹 브라우저로 미션 크리티컬 관제·제어가 가능한가?

관제(시각화)는 브라우저가 담당하지만, 제어 경로는 Tier 1 서버와 PLC 인터록이 담당합니다. 브라우저가 멈춰도 제어·안전은 영향받지 않습니다(4.4, 4.5). 브라우저 기반의 이점은 설치 없이 관제실·태블릿·XR에서 동일 화면을 쓰는 것입니다.

Q2. 왜 Omniverse 같은 플랫폼이 아닌가?

대체가 아니라 병행 가능한 계층입니다. 본 플랫폼은 CAD 없이 촬영만으로 현재 상태의 실사 공간을 만들고, 온프레미스 AI 제어 계층을 결합하는 데 강점이 있습니다(2.7). 이미 Omniverse 자산이 있다면 USD 메시를 Three.js로 가져와 3DGS와 융합할 수 있습니다.

Q3. 3DGS는 기하 정밀도가 낮지 않은가?

3DGS는 시각 재현에 최적화되어 있으며 계측용 기하가 아닙니다. 그래서 0.01mm급 정밀도는 계측 장비(비전·레이저)가 담당하고, 3DGS+VLM은 "어디를, 왜"를 담당합니다(3.2-보강 B). 스케일·좌표 정합은 3.7에서 기준점·LiDAR로 수행합니다.

Q4. LLM이 명령을 오해하면 로봇이 위험한 행동을 하지 않는가?

모든 명령은 규칙 기반 안전 필터와 안전 PLC 인터록을 거치며, 도입 초기에는 승인 모드(L1)로 운영합니다(7.2). 완전 자율(L3)은 설비별 검증 후에만 허용됩니다.

Q5. 오픈소스 런타임에 의존해도 되는가?

Spark.js는 MIT 라이선스로 상용 이용이 가능하고 World Labs가 활발히 유지보수 중입니다 [공식]. 버전을 고정·자체 호스팅하며, 업그레이드는 검증 절차를 거칩니다(R7).

Q6. 원문 수치(1ms 정렬, 95% 압축, <20ms)는 믿을 수 있는가?

현재 근거 등급이 [?] 또는 [목표]인 항목은 부록 B 조건으로 파일럿에서 실측하여 [실측]으로 승격합니다. 근거 없는 수치는 검수 판정에 쓰지 않습니다(6.1).

Q7. 예산이 원문보다 커지면?

표 5-1에서 가장 큰 변동 요인은 H100 NVL 수량입니다. 2장 구성 또는 대체 GPU(4.8)로 원문 범위 내 수렴이 가능하며, 정식 견적서로 확정합니다.

Q8. 공장 스캔 데이터가 외부로 나가지 않는가?

데이터는 고객사 자산이며 기본 온프레미스 저장입니다. 파일럿에서 클라우드 학습을 쓰는 경우에도 범위·지역·삭제 절차를 계약에 명시하고, 불가 시 온프레미스 학습으로 대체합니다(5.10, 7.3).

Q9. 설비가 바뀌면 다시 다 찍어야 하는가?

변경된 구역만 부분 재촬영·재학습하며, LoD 스트리밍 구조상 씬을 구역 단위로 갱신할 수 있습니다(3.7).

Q10. 기존 MES/SCADA와 어떻게 연결되는가?

OPC UA 브리지로 설비 데이터를 받고, XAI 원장의 이벤트를 MES로 내보냅니다. 인터페이스 명세는 공동 범위로 합의합니다(5.7).

부록 A. 전체 코드 원문 및 발체 범위 명세

본 부록은 3장의 코드 3-1 ~ 3-3을 수정 없이 재수록합니다(라인 번호는 편의상 부여). 3장과 동일 내용이며, 엔지니어가 한 곳에서 대조할 수 있도록 모았습니다.

A-1. 신경계

```

1 import * as THREE from "three";
2 import { SparkRenderer, SplatMesh, PackedSplats } from "@sparkjsdev/spark";
3
4 // [1] 신경계 코어: 16/32비트 GPU 메모리 버퍼 생성 (메모리 95% 절감)
5 const splatMesh = new SplatMesh({
6   url: "https://sparkjs.dev/assets/splats/industrial_motor.spz",
7   lod: true,
8   extSplats: true,
9   onProgress: (ev) => console.log(`신경망 로딩: ${Math.round((ev.loaded/ev.total)*100)}%`
10 });
11 scene.add(splatMesh);
12
13 // [2] 백그라운드 WebWorker (Rust WASM Radix Sort) 연동 구조
14 await splatMesh.initialized;
15 console.log(`신경계 활성화 완료: 총 ${splatMesh.numSplats.toLocaleString()}개 가우시안 세포 연결됨`);
16
17 // [3] IoT 실시간 센서 스트림 → 가우시안 텐서 신경 반사
18 function updateSensorNervousSystem(temperature, vibrationHz) {
19   if (!splatMesh.packedSplats) return;
20   if (temperature > 40.0) {
21     const alertTint = new THREE.Color(1.0, 0.2, 0.2); // 과열 시 붉은색 텐서 발광
22     splatMesh.splatRgba = new Float32Array([alertTint.r, alertTint.g, alertTint.b, 1.0]);
23   }
24 }

```

A-2. 두뇌

```

1 // [1] 두뇌 코어: 자연어 프롬프트를 해석하여 3DGS 씬과 로봇 기계를 조작
2 async function executeSpatialBrainCommand(prompt) {
3   appendXaiLog(`[두뇌/LLM] 자연어 명령 수신: "${prompt}"`);
4   const lower = prompt.toLowerCase();
5
6   if (lower.includes("모터") || lower.includes("가속") || lower.includes("조립")) {
7     showDigitalTwinMode("ev");
8     robotArmSpeedMultiplier = 1.5;
9     rotorRotationSpeed = 4.0;
10    appendXaiLog(`[VLM/SI] 모터 조립 어포던스(0.01mm) 정렬 완료 → 로봇 궤적 150% 가속 신호 인가`);
11  }
12  else if (lower.includes("항만") || lower.includes("agv")) {
13    showDigitalTwinMode("port");
14    triggerAgvDispatch("Bay-02");
15    appendXaiLog(`[PAI/LAM] 스마트 항만 크레인 레이저 정렬 가이드 및 AGV 자율 주행 궤적 생성`);
16  }
17  else if (lower.includes("사이버펑크") || lower.includes("네온")) {
18    setStudioLightingEnvironment("cyberpunk");
19    triggerSdfPulseWave();
20    appendXaiLog(`[두뇌/환경] 사이버펑크 네온 스펙트럼 및 SDF 동적 조명 필드 전환 완료`);
21  }
22 }
23
24 // [2] 설명가능한 AI(XAI) 인과 추론 엔진 실시간 스트리밍
25 function appendXaiLog(message) {
26   const logStream = document.getElementById("xai-log-stream");
27   const entry = document.createElement("div");
28   entry.className = "xai-log-entry";
29   entry.innerHTML = `<span class="xai-badge xai-badge-ai">LIVE</span> <span>${message}</span>`;
30   logStream.appendChild(entry);
31   logStream.scrollTop = logStream.scrollHeight;
32 }

```

A-3. 물리 엔진

```

1 import { constructGrid, constructSpherePoints } from "@sparkjsdev/spark";
2
3 // [1] Dyno 물리 엔진: WebGL 셰이더 없이 3DGS 점들을 실시간 수학 물리 필드로 변형
4 function createProgrammablePhysicsField() {
5   const physicsSplatMesh = new SplatMesh({
6     constructSplats: (splats) => constructGrid({
7       splats,
8       extents: new THREE.Box3(new THREE.Vector3(-2.5, -1.5, -2.5), new THREE.Vector3(2.5, 1.5, 2.5)),
9       stepSize: 0.45,
10      pointRadius: 0.04
11    })
12  });
13  return physicsSplatMesh;
14 }
15
16 // [2] Three.js 일반 3D 폴리곤 로봇 팔과 3DGS 실사 공간의 Z-Buffer 하이브리드 결합
17 const robotArmGroup = new THREE.Group();
18 const joint1 = new THREE.Mesh(new THREE.SphereGeometry(0.11, 16, 16), new
19 THREE.MeshStandardMaterial({ color: 0xff4b63, metalness: 0.8 }));
20 const link1 = new THREE.Mesh(new THREE.CylinderGeometry(0.055, 0.055, 0.75, 16), new
21 THREE.MeshStandardMaterial({ color: 0x94a3b8, metalness: 0.8 }));
22 link1.position.set(0.28, 0.55, 0); link1.rotation.z = -Math.PI / 4;
23 robotArmGroup.add(joint1); robotArmGroup.add(link1);
24 scene.add(robotArmGroup);
25
26 // [3] PINN 기반 열역학 자율 복구 루프
27 function triggerAutonomousPhysicalRecovery() {
28   appendXaiLog('[PINN] 나비에-스토크스 유체역학 연산: 냉각 유체 순환 속도 20% 보정 계산');
29   appendXaiLog('[PAI/LAM] 다관절 로봇 6축 토크 피드백 제어: 모터 틱새 0.01mm 재정렬 완료');
30   document.getElementById("gauge-temp").textContent = "41.2 °C (정상)";
31   document.getElementById("gauge-vibe").textContent = "12.1 Hz (안정)";
32 }

```

A-4. 발췌 범위 명세

식별자	위치	상태	비고
scene, camera, renderer	전 코드	미수록 (데모 앱 초기화부)	THREE.Scene 등 표준 초기화 + SparkRenderer 등록
showDigitalTwinMode()	코드 3-2	미수록	데모 모드 전환(ev/port)
robotArmSpeedMultiplier, rotorRotationSpeed	코드 3-2	미수록 (전역 변수)	애니메이션 루프에서 참조
triggerAgvDispatch()	코드 3-2	미수록	AGV 궤적 생성
setStudioLightingEnvironment(), triggerSdfPulseWave()	코드 3-2	미수록	조명·SDF 효과
constructSpherePoints	코드 3-3	import 후 본 발췌본 내 미사용	다른 모듈에서 사용
#xai-log-stream, #gauge-temp, #gauge-vibe	코드 3-2/3-3	HTML 요소	관제 UI 마크업 미수록

부록 B. 벤치마크 조건 명세 (기록 양식)

필드	기록 내용
측정 항목	표 3-4의 항목명
Spark.js 버전 / THREE.js 버전	예: 2.1.0 / 0.180.0
브라우저 / 버전 / OS	
GPU / 드라이버 / VRAM	
스플랫 수 / 예산 / SH 차수 / 파일 포맷	
해상도 / 주사율 / 메시 객체 수	
카메라 이동 패턴	정지 / 회전 / 이동 스크립트 ID
측정 도구 / 반복 횟수 / 통계(평균 · p95)	
원본 로그 파일 / 해시	근거 포인터
판정 등급	[실측] 승격 여부

부록 C. 용어집

용어	설명
가우시안 스피랫(Splat)	위치·크기·회전·색·불투명도를 가진 3D 타원체. 수백만 개가 모여 실사 공간을 이룸
PackedSplats	Spark.js에서 스피랫 데이터를 GPU 텍스처로 패킹한 버퍼 구조 [공식]
Dyno(셰이더 그래프)	JavaScript 계산 그래프로 GLSL을 합성해 GPU에서 스피랫을 생성·변형하는 Spark.js 시스템 [공식]
LoD 트리	스�피랫을 계층적으로 병합해 거리·화면 크기에 따라 세부를 조절하는 구조(Spark 2.0) [공식]
.RAD	Spark 2.0의 점진적 스트리밍 포맷 [공식]
Affordance(쓰임새)	사물이 허용하는 행위(회전하는 로터, 삽입 가능한 틈새 등)
접지(Grounding)	언어 표현을 3D 씬의 객체·좌표에 연결하는 과정
PINN	물리 미분방정식 잔차를 손실에 포함하여 물리 법칙을 만족하는 예측을 하는 신경망
폐쇄 루프(Closed-Loop)	센서 → 인지 → 예측 → 행동이 피드백으로 연결된 제어 구조
인터록(Interlock)	조건 미충족 시 동작을 차단하는 안전 회로
자율 등급 L0~L3	7.2의 AI 개입 수준 단계
근거 등급	[실측]/[공식]/[목표]/[?]

부록 D. 참고문헌

- 15.Kerbl, B. et al., "3D Gaussian Splatting for Real-Time Radiance Field Rendering," SIGGRAPH 2023.
- 16.Mildenhall, B. et al., "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," ECCV 2020.
- 17.World Labs, Spark – An advanced 3D Gaussian Splatting renderer for THREE.js, GitHub [sparkjsdev/spark](https://github.com/sparkjsdev/spark) (MIT License), 문서 sparkjs.dev/docs (SplatMesh, PackedSplats, Level-of-Detail Deep Dive).
- 18.World Labs, "Streaming 3DGS worlds on the web" (Spark 2.0 기술 블로그), 2026-04-14.
- 19.Microsoft Research, TRELLIS: Structured 3D Latents for Scalable and Versatile 3D Generation.
- 20.NVIDIA, RTX PRO 6000 Blackwell Workstation Edition 제품 자료; RTX 6000 Ada Generation 제품 자료; H100 NVL 제품 자료.
- 21.IEC 62443 시리즈(산업 자동화 및 제어 시스템 보안); ISO 10218-1/2; ISO/TS 15066; ISO 13849-1; IEC 62541(OPC UA); IEC 61158(EtherCAT).
- 22.Raissi, M., Perdikaris, P., Karniadakis, G. E., "Physics-informed neural networks," J. Comput. Phys., 2019.

추가 사항: 원문 제목 "[Master Document] 차세대 공간 지능(Spatial Intelligence) 런타임 플랫폼: Spark.js 소프트웨어 아키텍처 및 산업 인프라 도입 백서"는 표지에 그대로 사용했습니다.