

AI 다이어그램 자동화 백서: 한 문장으로 아키텍처 다이어그램을 그릴 수 있다면

"아키텍처 다이어그램 하나 그리는 데 도구 세팅부터 꾸미기까지, 왜 이렇게 오래 걸리는 거죠?"

팀마다 파워포인트, Visio, Lucidchart 를 따로 쓰면서 파일 호환은 안 되고, 연간 수천만 원의 라이선스 비용은 계속 불어나며, AI 시대에도 여전히 수동 드래그앤드롭에 의존하는 것이 대부분의 IT 조직이 직면한 현실입니다.

Excalidraw는 MIT 오픈소스 기반 무료 도구로, MCP 프로토콜을 통해 Claude와 ChatGPT에서 자연어 프롬프트만으로 아키텍처 다이어그램을 실시간 자동 생성합니다.



 hello@cncf.co.kr

 02-469-5426

 www.cncf.co.kr

Contents

1장: Excalidraw란 무엇이며 왜 주목하는가	4
1.1 Excalidraw의 탄생과 설계 철학	4
1.1.1 Meta 엔지니어가 시작한 오픈소스 화이트보드	4
1.1.2 손그림 스타일의 의도적 설계: 완벽주의의 함정을 깨다	5
1.1.3 MIT 라이선스와 기업 도입 장벽 최소화	6
1.2 기존 다이어그램 도구의 한계와 Excalidraw의 해결	6
1.2.1 Visio/PowerPoint/Lucidchart의 구조적 한계	7
1.2.2 Excalidraw가 해결하는 핵심 페인포인트	8
1.3 대상 사용자와 채택 현황	9
1.3.1 주요 사용자 그룹별 활용 가치	9
1.3.2 기술 채택 단계: 얼리 메인스트림 진입	10
2장: 경쟁 제품 비교와 Excalidraw의 포지셔닝	11
2.1 주요 경쟁 제품과의 기능 비교	11
2.1.1 draw.io: 정형화된 기술 다이어그램의 강자	11
2.1.2 Mermaid: 텍스트 기반 다이어그램과 토큰 효율성	13
2.1.3 tldraw과 Miro: 각각의 차별점	14
2.2 AI 통합 관점의 경쟁 우위	16
2.2.1 MCP 서버 기반 AI 연동 에코시스템 비교	16
2.2.2 3도구 병행 전략: Excalidraw + draw.io + Mermaid	18
2.3 Excalidraw의 알려진 한계	19
2.3.1 기업 도입 시 인지해야 할 기능 격차	19
3장: AI 시대의 Prompt Drawing — Excalidraw와 AI Agent 연동	21
3.1 MCP 프로토콜 기반 AI 다이어그램 생성	21
3.1.1 Excalidraw 공식 MCP 서버: 구조와 설치	21
3.1.2 Claude Code Skill과 커뮤니티 MCP 생태계	23

- 3.1.3 ChatGPT + Mermaid + Excalidraw 하이브리드 파이프라인 24
- 3.2 실전 Prompt Drawing 워크플로 25
 - 3.2.1 자연어로 아키텍처 다이어그램 생성하기 26
 - 3.2.2 Microsoft Architecture Review Agent 사례 28
 - 3.2.3 Headless 렌더링과 배치 내보내기 29
- 3.3 아키텍처 다이어그램 예제 갤러리 30
 - 3.3.1 MSA 설계 다이어그램 예제 31
 - 3.3.2 Kubernetes 구조와 LLM/RAG 파이프라인 예제 32
- 4장: 플랫폼 통합, 라이브러리, 내보내기 활용 33**
 - 4.1 개발 도구 통합 33
 - 4.1.1 VS Code: 코드베이스 내 다이어그램 관리 33
 - 4.1.2 Obsidian: 지식 관리와 시각적 사고의 결합 34
 - 4.1.3 Confluence와 Notion: 엔터프라이즈 위키 통합 36
 - 4.2 커뮤니티 라이브러리와 템플릿 생태계 37
 - 4.2.1 IT 인프라 아이콘 라이브러리 37
 - 4.2.2 시스템 설계 템플릿과 마켓플레이스 38
 - 4.3 내보내기와 산출물 활용 39
 - 4.3.1 PNG/SVG 내보내기와 썸 데이터 임베딩 39
 - 4.3.2 PowerPoint 연동 우회 전략과 NPM 임베딩 40
 - 4.4 핵심 기술 아키텍처 이해 42
 - 4.4.1 이중 캔버스 아키텍처와 성능 최적화 42
- 5장: 기업 도입 가이드 — 평가에서 운영까지 43**
 - 5.1 PoC에서 표준화까지: 4단계 도입 로드맵 43
 - 5.1.1 4단계 도입 경로와 조직 유형별 선택 44
 - 5.1.2 1주 평가 체크리스트와 의사결정 기준 45
 - 5.2 셀프호스팅 vs SaaS 의사결정 46
 - 5.2.1 비용/기능/보안 관점 비교 매트릭스 47
 - 5.2.2 셀프호스팅 인프라 구성 가이드 48

5.3 마이그레이션 전략과 운영 주의사항	49
5.3.1 기존 도구에서의 전환 경로	49
5.3.2 운영 시 성능 한계와 우회 방안	50
5.3.3 도입 시 흔한 실수 패턴과 방지 가이드	52
부록	53
11가지 카테고리 -> 5장 매핑 테이블	53
핵심 주제 -> 장 매핑 테이블	55
Appendix	57
References	57
Glossary	60

1장: Excalidraw란 무엇이며 왜 주목하는가

Excalidraw는 2020년대 들어 급부상한 오픈소스 화이트보드 다이어그램 도구로, 기업 IT 환경에서 표준 그리기 도구로 자리 잡을 수 있는지에 대한 논의가 활발히 이루어지고 있습니다. 본 장에서는 Excalidraw의 탄생 배경과 설계 철학, 기존 다이어그램 도구의 한계와 Excalidraw가 제공하는 해결책, 그리고 주요 사용자 그룹과 실제 채택 현황을 기술합니다. 이를 통해 Excalidraw가 왜 주목받는지, 그리고 기업 도입 관점에서 어떤 경쟁력을 갖추었는지 심층적으로 살펴봅니다.

1.1 Excalidraw의 탄생과 설계 철학

Excalidraw는 단순한 다이어그램 툴을 넘어, 오픈소스와 커뮤니티 중심의 혁신적인 접근, 그리고 손그림 스타일의 의도적 설계로 기존 시장의 패러다임을 변화시키고 있습니다. 이 섹션에서는 Excalidraw의 출발점, 디자인 원칙, 그리고 오픈소스 라이선스가 기업 도입에 미치는 영향까지 상세히 다룹니다. Excalidraw가 어떻게 시작되었으며, 어떤 설계 철학을 바탕으로 성장했는지, 그리고 기업 환경에서의 도입 장벽을 어떻게 낮추었는지 살펴봄으로써, 이 도구가 왜 IT 업계에서 빠르게 확산되고 있는지 그 근본적인 이유를 분석합니다.

1.1.1 Meta 엔지니어가 시작한 오픈소스 화이트보드

Excalidraw의 탄생은 Meta(구 Facebook) 엔지니어인 Christopher Chedeau(닉네임 vjeux)가 2020년 1월 1일 트위터를 통해 공개적으로 프로젝트를 시작하면서 이루어졌습니다. 그는 “Excali(bur)” (전설의 검)와 “Draw” (그리다)를 합쳐 Excalidraw라는 이름을 붙였으며, 단 3일 만에 첫 버전의 핵심 기능(도형 그리기, 이동, 삭제, 저장 등)을 완성했습니다. 이처럼 빠른 초기 개발 속도는 오픈소스 커뮤니티의 폭발적 관심을 불러일으켰고, GitHub 공개 후 불과 2년 만에 103,000개 이상의 Stars와 335명 이상의 기여자를 모으는 등 전 세계 개발자 생태계에서 독보적인 성장세를 보였습니다.

Excalidraw의 GitHub 저장소(<https://github.com/excalidraw/excalidraw>)는 2024년 기준 103K+ Stars, 335명 이상의 컨트리뷰터, 1,900개 이상의 포크, 그리고 3,000건이 넘는 이슈가 등록되어 있습니다. 이는 단순한 개인 프로젝트가 아니라, 글로벌 오픈소스 커뮤니티의 집단 지성에 의해 빠르게 진화하는 대표적인 사례로 평가받고 있습니다. Meta의 주요 엔지니어가 주도적으로

참여하면서 코드 품질과 프로젝트 관리 체계 역시 높은 수준을 유지하고 있습니다.

Excalidraw라는 이름은 전설의 검 엑스칼리버(Excalibur)에서 착안한 것으로, “누구나 강력한 무기를 사용할 수 있다”는 오픈소스의 철학을 상징합니다. 이처럼 프로젝트의 시작부터 “개방성”과 “접근성”을 강조한 점이, 이후 기업과 커뮤니티 모두에서 폭넓게 채택되는 배경이 되었습니다.

1.1.2 손그림 스타일의 의도적 설계: 완벽주의의 함정을 깨다

Excalidraw는 RoughJS 라이브러리를 활용하여, 일부러 비정형적이고 손으로 그린 듯한 “러프 스케치” 스타일을 구현합니다. 이는 사용자가 도형의 정렬, 색상, 테두리 등 세부적인 꾸미기에 집착하지 않고, 본질적인 아이디어와 구조를 빠르게 시각화하는 데 집중하도록 유도합니다. 실제로 많은 사용자가 “픽셀 단위의 완벽한 정렬”에 시간을 허비하는 기존 도구의 한계를 지적해왔으며, Excalidraw는 이러한 완벽주의의 함정을 의도적으로 피하도록 설계되었습니다.

손그림 스타일은 “이것은 완성된 산출물이 아니라, 논의를 위한 초안”이라는 신호를 줍니다. 이는 회의, 브레인스토밍, 아키텍처 설계 등에서 심리적 부담을 줄이고, 자유로운 아이디어 교환을 촉진하는 효과가 있습니다. 실제로 Excalidraw를 사용하는 팀에서는 “형식”이 아닌 “내용”에 집중하는 문화가 자연스럽게 형성되며, 이는 생산성과 창의성 모두에 긍정적인 영향을 미칩니다.

Excalidraw의 UI는 최소한의 도구(펜, 도형, 텍스트, 커넥터 등)와 직관적인 인터랙션에 집중되어 있습니다. 복잡한 메뉴나 옵션 없이, 누구나 웹 브라우저에서 즉시 그리기를 시작할 수 있습니다. 이러한 단순함은 “도구가 목적이 아니라, 아이디어 전달이 목적”임을 명확히 하며, 엔지니어, 디자이너, 기획자 등 다양한 역할의 사용자가 장벽 없이 활용할 수 있게 합니다.

이러한 손그림 스타일의 설계는 실제 업무 환경에서 여러 긍정적인 효과를 가져옵니다. 예를 들어, 개발팀이 시스템 아키텍처를 논의할 때, 완성된 다이어그램이 아닌 러프 스케치 형태로 의견을 공유함으로써, 각자의 아이디어가 자유롭게 제시되고 빠르게 수정될 수 있습니다. 또한, 디자이너가 UX 초기 설계를 진행할 때, 완성본이 아닌 스케치 형태로 피드백을 받아 반복적으로 개선할 수 있습니다. 이처럼 Excalidraw의 손그림 스타일은 완벽주의에 따른 병목을 제거하고, 실질적인 협업과 창의적 사고를 촉진하는 데 큰 역할을 합니다. 실제 사례로, 글로벌 스타트업에서는 Excalidraw를 활용해 투자자 프레젠테이션 초안을 빠르게 제작하고, 팀 내에서 실시간 피드백을 받아 최종안을 완성하는 과정을 단축한 바 있습니다. 이처럼 Excalidraw의 의도적 설계는 다양한 산업군에서 혁신적인 협업 문화를 만들어내고 있습니다.

1.1.3 MIT 라이선스와 기업 도입 장벽 최소화

Excalidraw는 MIT 라이선스를 채택하고 있습니다. MIT 라이선스는 소스 코드의 상업적 사용, 수정, 배포, 재라이선싱까지 모두 허용하는 가장 관대한 오픈소스 라이선스 중 하나로, 기업 입장에서는 법적·운영적 리스크가 거의 없습니다. 이는 Penpot(MPL-2.0), draw.io(아파치 2.0) 등 경쟁 제품 대비 명확한 차별점입니다.

MIT 라이선스 덕분에 기업은 Excalidraw를 자체 인프라에 배포(셀프호스팅)하거나, 필요에 따라 기능을 커스터마이징하여 사내 워크플로우에 통합할 수 있습니다. 특히 보안이나 데이터 주권이 중요한 조직에서는 오픈소스 버전을 직접 운영함으로써, SaaS의 데이터 외부 저장 문제를 우회할 수 있습니다.

Excalidraw는 무료 오픈소스 버전과 유료 SaaS 서비스인 Excalidraw+로 이원화되어 있습니다. 무료 버전은 모든 핵심 기능을 제공하며, 협업, 저장, 내보내기 등도 무제한으로 사용할 수 있습니다. Excalidraw+는 \$6~7/사용자/월의 가격으로, SSO, RBAC, 감사로그, SOC 2 인증, 음성/화면 공유, 프레젠테이션 등 엔터프라이즈 기능을 추가로 제공합니다. 그러나 기본적인 그리기, 협업, 내보내기 등은 무료 오픈소스 버전만으로도 충분히 활용할 수 있다는 점이 기업 도입의 진입장벽을 크게 낮춥니다.

기업 도입 관점에서 MIT 라이선스의 장점은 단순히 비용 절감에 그치지 않습니다. 예를 들어, 보안 규제가 엄격한 금융기관이나 공공기관에서는 외부 SaaS 사용이 제한되는 경우가 많습니다. 이때 Excalidraw의 오픈소스 버전을 자체 서버에 설치하여 내부망에서만 운영함으로써, 데이터 유출 위험을 최소화하고 규제 준수를 보장할 수 있습니다. 또한, 기업별 요구에 맞춰 기능을 추가하거나 UI를 변경하는 것도 자유롭게 가능합니다. 실제로 글로벌 IT 기업들은 Excalidraw를 사내 개발 포털, 아키텍처 리뷰 툴, 교육 플랫폼 등에 통합하여 자체 워크플로우에 맞게 커스터마이징하고 있습니다. 이러한 유연성은 경쟁 제품 대비 Excalidraw가 기업 환경에서 빠르게 확산되는 핵심 요인 중 하나입니다.

1.2 기존 다이어그램 도구의 한계와 Excalidraw의 해결

Excalidraw가 주목받는 배경에는 기존 다이어그램 도구들의 구조적 한계가 자리하고 있습니다. 이 섹션에서는 Visio, PowerPoint, Lucidchart 등 전통적 도구의 한계점을 구체적으로 짚고,

Excalidraw가 이를 어떻게 해결하는지 IT 의사결정자 관점에서 분석합니다. 기존 도구들이 어떤 문제를 야기하는지, 그리고 Excalidraw가 실제로 어떤 방식으로 이러한 문제를 해소하는지 구체적으로 비교함으로써, 기업 환경에서 Excalidraw의 도입 가치와 경쟁력을 명확히 설명합니다.

1.2.1 Visio/PowerPoint/Lucidchart의 구조적 한계

기업 내에서는 팀마다 Visio, PowerPoint, Lucidchart, Miro 등 다양한 다이어그램 도구를 사용합니다. 이로 인해 파일 호환성, 버전 관리, 협업 프로세스가 파편화되어, 조직 전체의 생산성과 정보 공유 효율이 저하됩니다. 특히, 하나의 프로젝트에서 여러 도구가 혼용될 경우, 다이어그램의 일관성 유지와 변경 관리가 매우 어렵습니다.

Visio는 \$13.99/월, Lucidchart는 \$7.95/월부터, Miro는 \$8~16/월 등 상당한 비용이 발생합니다. 대규모 조직에서는 수십~수백 명의 라이선스 비용이 누적되어, 연간 수천만 원 이상의 예산이 소모됩니다. 또한, 일부 도구는 사용자 수에 따라 가격이 급격히 증가하거나, 기능 제한이 있어 추가 비용이 발생합니다.

기존 도구들은 픽셀 단위의 정렬, 색상, 폰트, 테두리 등 시각적 완성도에 집착하게 만드는 UI를 제공합니다. 이로 인해 사용자는 다이어그램의 본질적 내용보다 “꾸미기”에 더 많은 시간을 소비하게 되며, 실제로 중요한 의사결정이나 아이디어 전달이 지연되는 병목이 발생합니다.

AI와 자동화가 강조되는 시대임에도, 기존 도구들은 여전히 수동 드래그앤드롭 방식에 의존합니다. 복잡한 아키텍처나 시스템 다이어그램을 반복적으로 업데이트할 때, 자동화·생성형 AI와의 연동이 미흡하여, 실질적인 생산성 향상에 한계가 있습니다.

이러한 한계는 실제 업무 환경에서 다양한 문제를 야기합니다. 예를 들어, 대규모 프로젝트를 진행하는 기업에서는 여러 팀이 각기 다른 도구를 사용함으로써, 다이어그램 파일의 호환성 문제가 발생하고, 버전 관리가 어려워집니다. 이로 인해 프로젝트 진행 상황을 공유하거나, 변경 사항을 반영하는 과정에서 혼란이 생기고, 의사결정이 지연되는 사례가 빈번하게 나타납니다. 또한, 라이선스 비용이 누적되어 예산 부담이 커지며, 일부 도구는 기능 제한 때문에 추가 비용을 지불해야 하는 상황이 발생합니다. 시각적 완성도에 집착하는 UI는 사용자가 본질적인 아이디어 전달보다 꾸미기에 시간을 소비하게 만들고, 실제로 중요한 논의가 늦어지는 병목 현상을 초래합니다. AI 시대에 맞는 자동화 기능이 부족하여, 반복적인 작업을 수동으로 처리해야 하는 비효율도 여전히 존재합니다. 이러한 구조적 한계는 기업이 새로운 다이어그램 도구를 도입할 필요성을 느끼게 하는

주요 원인입니다.

1.2.2 Excalidraw가 해결하는 핵심 페인포인트

Excalidraw는 MIT 오픈소스 라이선스 기반으로, 누구나 무료로 사용할 수 있습니다. 별도의 라이선스 비용이나 사용자 수 제한이 없으므로, 예산 부담 없이 조직 전체에 확산할 수 있습니다. 이는 기존 상용 도구 대비 가장 큰 경쟁력 중 하나입니다.

Excalidraw는 별도의 설치나 복잡한 가입 절차 없이, 웹 브라우저(<https://excalidraw.com>)에서 즉시 사용할 수 있습니다. 이는 IT 인프라 제약이 있는 환경, 원격 근무, BYOD(Bring Your Own Device) 환경에서도 손쉽게 도입할 수 있게 합니다.

Excalidraw는 URL 공유만으로 실시간 동시 편집이 가능하며, E2E(End-to-End) 암호화를 지원하여 보안성도 확보합니다. 별도의 계정 생성이나 복잡한 권한 설정 없이, 누구나 링크만 있으면 협업에 참여할 수 있습니다. 이는 Slack, Notion, Confluence 등 다양한 협업 툴과의 연동성도 뛰어납니다.

AWS, Kubernetes, Docker 등 주요 IT 인프라 아이콘 라이브러리와의 통합, 그리고 MCP(Model Context Protocol) 기반 Claude/ChatGPT 등 AI와의 네이티브 연동이 가능합니다. 이를 통해 AI 프롬프트 기반 자동 다이어그램 생성, 실시간 협업, 워크플로우 자동화 등 최신 트렌드에 부합하는 기능을 제공합니다.

Excalidraw는 비용, 접근성, 협업, 확장성, AI 연동 등 모든 핵심 페인포인트를 동시에 해결하며, 기업의 표준 다이어그램 도구로서의 경쟁력을 갖추고 있습니다. 특히, 오픈소스 기반의 커스터마이징 자유도와 셀프호스팅 옵션은 보안과 데이터 주권이 중요한 기업 환경에서 큰 강점으로 작용합니다.

Excalidraw가 실제로 해결하는 페인포인트를 구체적으로 살펴보면, 첫째로 비용 측면에서 MIT 라이선스 덕분에 조직 내 모든 사용자가 무료로 도구를 사용할 수 있습니다. 이는 기존 상용 도구 대비 연간 수천만 원의 예산을 절감할 수 있게 해줍니다. 둘째로 접근성 측면에서는 별도의 설치나 복잡한 가입 절차 없이, 웹 브라우저에서 즉시 사용이 가능하므로 IT 인프라가 제한된 환경에서도 손쉽게 도입할 수 있습니다. 셋째로 협업 측면에서는 URL 공유만으로 실시간 동시 편집이 가능하며, E2E 암호화로 보안성을 확보할 수 있습니다. Slack, Notion, Confluence 등 다양한 협업 툴과의 연동도 매우 뛰어나, 조직 내외부의 다양한 협업 시나리오에 대응할 수 있습니다. 넷째로

에코시스템 측면에서는 AWS, Kubernetes, Docker 등 주요 IT 인프라 아이콘 라이브러리와 통합, 그리고 Claude/ChatGPT 등 AI와의 네이티브 연동이 가능하며, 최신 트렌드에 부합하는 자동화 기능을 제공합니다. 마지막으로 IT 의사결정자 관점에서는 오픈소스 기반의 커스터마이징 자유도와 셀프호스팅 옵션이 보안과 데이터 주권이 중요한 기업 환경에서 큰 강점으로 작용합니다. 실제로 글로벌 IT 기업들은 Excalidraw를 사내 개발 포털, 아키텍처 리뷰 툴, 교육 플랫폼 등에 통합하여 자체 워크플로우에 맞게 커스터마이징하고 있습니다. 이러한 다각적인 해결책은 Excalidraw가 기존 도구의 한계를 극복하고, 기업의 표준 다이어그램 도구로 자리잡는 데 결정적인 역할을 하고 있습니다.

1.3 대상 사용자와 채택 현황

Excalidraw는 개발자, DevOps, 기획자, 디자이너, 교육자 등 다양한 그룹에서 폭넓게 활용되고 있습니다. 이 섹션에서는 각 사용자 그룹별 활용 가치와, 실제 시장에서의 채택 현황을 정량적 데이터와 함께 분석합니다. Excalidraw가 어떤 사용자들에게 실질적인 효용을 제공하는지, 그리고 실제로 시장에서 어떻게 채택되고 있는지 구체적인 사례와 데이터를 통해 설명함으로써, 도구의 확산 배경과 미래 전망을 명확히 제시합니다.

1.3.1 주요 사용자 그룹별 활용 가치

개발자들은 Excalidraw를 활용해 시스템 아키텍처, 마이크로서비스 구조, API 플로우, 데이터베이스 설계 등 복잡한 기술 구조를 빠르게 시각화할 수 있습니다. 손그림 스타일은 “완성된 문서”가 아닌 “토론용 초안”으로서, 팀 내 자유로운 아이디어 교환과 반복적 개선을 촉진합니다.

DevOps 엔지니어는 Kubernetes, 클라우드 인프라, 배포 파이프라인, 네트워크 구조 등을 Excalidraw로 시각화합니다. AWS, K8s, Docker 등 커뮤니티 라이브러리를 활용해 실제 인프라와 유사한 다이어그램을 손쉽게 제작할 수 있으며, 실시간 협업을 통해 장애 대응, 설계 리뷰 등에도 활용도가 높습니다.

제품 매니저와 기획자는 사용자 플로우, 기능 의존성, 릴리즈 플랜, 워크플로우 등을 Excalidraw로 표현합니다. 단순한 와이어프레임부터 복잡한 프로세스 맵까지, 다양한 유형의 다이어그램을 빠르게 그릴 수 있어, 비개발자와의 커뮤니케이션에도 효과적입니다.

디자이너들은 Excalidraw를 활용해 와이어프레임, UI 플로우, 사용자 여정 맵 등 초기 UX

설계에 집중합니다. 손그림 스타일은 “디자인 완성본”이 아니라 “아이디어 스케치”임을 명확히 하여, 빠른 반복과 피드백을 가능하게 합니다.

교육자들은 Excalidraw를 화이트보드 대체 도구로 활용해, 실시간 강의, 토론, 그룹 프로젝트 등에서 높은 참여와 이해도를 이끌어냅니다. 스타트업 팀은 투자자 프레젠테이션, 비즈니스 모델 캔버스, MVP 설계 등에서 빠른 아이디어 시각화와 공유에 Excalidraw를 적극 활용합니다.

이처럼 다양한 사용자 그룹이 Excalidraw를 활용하는 이유는 도구의 직관적인 UI, 손그림 스타일, 실시간 협업 기능, 그리고 오픈소스 기반의 자유로운 커스터마이징에 있습니다. 실제 사례를 살펴보면, 글로벌 IT 기업에서는 개발자들이 시스템 아키텍처를 논의할 때 Excalidraw를 활용하여, 빠르게 초안을 작성하고 팀 내 피드백을 받아 반복적으로 개선하는 과정을 단축하고 있습니다. DevOps 엔지니어는 클라우드 인프라 구조를 실시간으로 시각화하여 장애 대응 및 설계 리뷰에 활용하고 있으며, 제품 매니저와 기획자는 복잡한 릴리즈 플랜이나 기능 의존성을 시각적으로 표현함으로써, 비개발자와의 커뮤니케이션을 원활하게 하고 있습니다. 디자이너는 초기 UX 설계 단계에서 Excalidraw를 활용하여 빠른 아이디어 스케치와 피드백을 반복하고, 교육자는 실시간 강의와 토론에서 화이트보드 대체 도구로 활용하여 학생들의 참여와 이해도를 높이고 있습니다. 스타트업 팀은 투자자 프레젠테이션이나 비즈니스 모델 캔버스 작성에 Excalidraw를 사용하여, 빠른 아이디어 시각화와 공유를 실현하고 있습니다. 이처럼 Excalidraw는 다양한 산업군과 직군에서 실질적인 업무 효율과 협업 문화를 혁신하는 데 기여하고 있습니다.

1.3.2 기술 채택 단계: 얼리 메인스트림 진입

Excalidraw는 GitHub 103K+ Stars, 월간 오가닉 방문자 277,000+, ROSS Index 4~5위, TheirStack 기준 104개 기업 사용 등, 개발자 커뮤니티에서 사실상 표준 도구로 자리잡았습니다. Google Cloud, Meta, CodeSandbox, Replit 등 글로벌 빅테크 기업에서도 공식적으로 채택되어, 신뢰성과 확장성을 입증받고 있습니다.

초기에는 개발자와 오픈소스 커뮤니티 중심으로 확산되었으나, 최근에는 PM, 디자이너, 교육자 등 비기술 직군으로도 채택이 빠르게 확산되고 있습니다. 이는 “설치 불필요, 무료, 실시간 협업, 손쉬운 사용성” 등 Excalidraw만의 특징점이 다양한 조직에서 인정받고 있음을 의미합니다.

Excalidraw는 단순한 오픈소스 툴을 넘어, 기업 IT 환경에서 실질적인 표준 다이어그램 도구로 자리잡고 있습니다. 이는 커뮤니티 중심의 빠른 혁신, 광범위한 확장성, 그리고 AI 시대에 부합하는

자동화·연동성 덕분에 가능해졌으며, 앞으로도 다양한 산업군에서 채택이 더욱 확대될 것으로 예상됩니다.

최근에는 개발자뿐만 아니라 PM, 디자이너, 교육자 등 비기술 직군에서도 채택이 빠르게 확산되고 있습니다. 이는 Excalidraw의 설치 불필요, 무료, 실시간 협업, 손쉬운 사용성 등 특징점이 다양한 조직에서 인정받고 있기 때문입니다. 기업 IT 환경에서 Excalidraw가 표준 다이어그램 도구로 자리잡는 것은 커뮤니티 중심의 빠른 혁신, 광범위한 확장성, 그리고 AI 시대에 부합하는 자동화·연동성 덕분입니다. 앞으로도 다양한 산업군에서 Excalidraw의 채택이 더욱 확대될 것으로 예상되며, 이는 오픈소스 도구가 기업 환경에서 실질적인 경쟁력을 갖추는 대표적인 사례로 자리매김할 것입니다.

2장: 경쟁 제품 비교와 Excalidraw의 포지셔닝

2.1 주요 경쟁 제품과의 기능 비교

기업 및 조직에서 다이어그램 도구를 선택할 때는 단순한 그림 그리기 이상의 다양한 요인을 고려해야 합니다. Excalidraw는 손그림 스타일의 자유로운 스케치와 실시간 협업을 강점으로 내세우지만, 시장에는 draw.io, Mermaid, tldraw, Miro 등 각기 다른 특징점을 가진 경쟁 제품들이 존재합니다. 이 섹션에서는 각 도구의 기능적 차별점과 활용 시나리오를 심층적으로 분석하여, Excalidraw가 어떤 위치에 있는지 명확히 조망합니다.

2.1.1 draw.io: 정형화된 기술 다이어그램의 강자

draw.io는 다이어그램 작성에 있어 표준화와 공식성을 중시하는 환경에서 강점을 가진 도구입니다. 특히 소프트웨어 개발, 데이터베이스 설계, 비즈니스 프로세스 모델링 등에서 요구되는 UML, BPMN, ERD 등 다양한 표준 표기법을 지원하며, 엔터프라이즈 협업 툴과의 통합성이 뛰어납니다. 이 섹션에서는 draw.io의 주요 기능과 한계, 그리고 Excalidraw와의 차별점을 구체적으로 살펴봅니다.

UML/BPMN/ERD 표준 지원

draw.io는 UML(Unified Modeling Language), BPMN(Business Process Model and

Notation), ERD(Entity-Relationship Diagram) 등 다양한 표준 표기법을 공식적으로 지원합니다. 이로 인해 소프트웨어 아키텍처, 데이터베이스 설계, 비즈니스 프로세스 모델링 등에서 정확하고 일관된 다이어그램을 작성할 수 있습니다. 특히, 복잡한 시스템 설계 문서나 공식 보고서에 적합하며, 기호와 레이아웃이 표준화되어 있어 협업 시 오해의 소지가 줄어듭니다. 표준 도형 라이브러리와 템플릿이 풍부하게 제공되므로, 반복적인 설계 작업의 효율성도 높습니다. 실제로 대규모 프로젝트에서는 draw.io의 표준화된 다이어그램이 품질 관리와 검증에 중요한 역할을 하며, 규제 산업에서도 신뢰받는 도구로 자리잡고 있습니다.

Atlassian 네이티브 통합

draw.io는 Atlassian의 Confluence, Jira 등 엔터프라이즈 협업 툴과의 네이티브 통합을 지원합니다. 이를 통해 문서 내에 다이어그램을 직접 삽입하고, 변경 이력을 추적하며, 팀원들과 실시간으로 피드백을 주고받을 수 있습니다. Confluence 플러그인 형태로 배포되어 설치와 관리가 용이하며, 기업 내 위키 및 이슈 트래킹 시스템과의 연계성이 뛰어납니다. 이 점은 대규모 조직에서 표준화된 문서 관리와 협업을 중시할 때 큰 장점이 됩니다. 실제로 글로벌 IT 기업들은 draw.io와 Atlassian 제품군의 통합을 통해 문서화와 협업의 효율성을 극대화하고 있습니다.

자동 레이아웃 및 LLM 커스터마이징

draw.io는 복잡한 다이어그램을 자동으로 정렬해주는 레이아웃 엔진을 내장하고 있습니다. 노드 간의 관계를 분석하여 최적의 배치로 자동 정렬하거나, 계층형, 트리형, 원형 등 다양한 레이아웃 옵션을 제공합니다. 최근에는 LLM(대형 언어 모델) 백엔드와의 연동도 지원하여, 자연어로 다이어그램을 생성하거나, 자동화된 다이어그램 완성 기능을 커스터마이징할 수 있습니다. 이는 AI 기반 워크플로우와의 통합 가능성을 넓혀줍니다. 예를 들어, ChatGPT와 연동하여 복잡한 시스템 구조를 빠르게 시각화할 수 있으며, 조직 내 자동화된 설계 검증 프로세스에도 활용됩니다.

실시간 협업 및 학습 곡선

draw.io의 한계 중 하나는 실시간 동시 편집 기능이 제한적이라는 점입니다. 동시 편집이 가능하긴 하지만, Excalidraw나 Miro와 같은 실시간 WebSocket 기반 협업에 비해 반응성이 떨어집니다. 또한, 표준 표기법과 다양한 기능을 익혀야 하므로, 비개발자나 입문자에게는 진입 장벽이 높을 수 있습니다. 복잡한 UI와 세부 설정이 오히려 생산성을 저해하는 경우도 있습니다. 실제로 초보 사용자는 기능의 방대함에 압도되어, 단순한 스케치나 브레인스토밍에는 draw.io를 선호하지 않는 경향이 있습니다.

스케치 용도와 공식 문서 구분

draw.io는 공식 문서와 보고서, 표준화된 설계에 최적화된 반면, 빠른 아이디어 스케치나 브레인스토밍에는 다소 무거운 인상을 줍니다. 픽셀 단위 정렬, 세밀한 도형 배치 등 완성도 높은 결과물을 요구하는 환경에 적합하지만, 비공식적이고 자유로운 토론에는 Excalidraw가 더 적합합니다. 실제로 많은 팀이 “비공식 스케치 = Excalidraw, 공식 문서 = draw.io”라는 이원화 전략을 채택하고 있습니다. 이러한 구분은 조직 내 문서화 프로세스의 효율성을 높이며, 각 도구의 장점을 극대화하는 방식으로 활용되고 있습니다.

2.1.2 Mermaid: 텍스트 기반 다이어그램과 토큰 효율성

Mermaid는 텍스트 기반으로 다이어그램을 정의하는 독특한 접근 방식을 가진 도구입니다. 개발자 중심의 팀에서 코드와 다이어그램을 함께 관리할 수 있도록 설계되었으며, Git 기반 버전 관리와 LLM 연동에 최적화되어 있습니다. 이 섹션에서는 Mermaid의 기능적 장점과 한계, 그리고 Excalidraw와의 연계 가능성을 상세히 분석합니다.

Git 버전 관리와 협업

Mermaid는 다이어그램을 텍스트 코드로 정의하는 마크다운 스타일의 도구입니다. 이로 인해 Git을 통한 버전 관리가 용이하며, 코드 리뷰와 협업이 자연스럽게 이루어집니다. 다이어그램의 변경 이력이 명확하게 기록되고, 브랜치 간 충돌 해결도 수월합니다. GitHub, GitLab 등 주요 코드 호스팅 플랫폼에서 네이티브로 Mermaid 렌더링을 지원하여, 코드와 다이어그램을 한 곳에서 관리할 수 있습니다. 실제로 오픈소스 프로젝트에서는 README 파일이나 개발 문서 내에 Mermaid 다이어그램을 삽입하여, 코드와 설계 정보를 일관되게 관리하는 사례가 많습니다. 이러한 방식은 개발자 간 협업의 투명성을 높이고, 문서화 자동화에도 큰 도움이 됩니다.

LLM 토큰 효율성과 RAG 활용

Mermaid는 XML, SVG 등 그래픽 기반 다이어그램 포맷 대비 최대 24배의 토큰 효율성을 자랑합니다. LLM(대형 언어 모델) 기반 RAG(Retrieval-Augmented Generation) 시스템에서 다이어그램을 프롬프트로 활용할 때, Mermaid 코드는 토큰 소모를 80% 이상 절감할 수 있습니다. 이는 LLM API 비용을 크게 낮추고, 더 복잡한 다이어그램을 한 번에 처리할 수 있게 해줍니다. 또한, 텍스트 기반이므로 LLM이 직접 다이어그램을 생성하거나 수정하는 데 최적화되어 있습니다. 예를 들어, ChatGPT나 Claude와 같은 AI가 자연어 명령을 Mermaid 코드로 변환하여, 자동화된 설계 문서 생성에 활용할 수 있습니다. 이는 개발팀의 생산성을 크게 높이는 혁신적인 방식입니다.

시각적 커스터마이징 한계

Mermaid의 단점은 시각적 커스터마이징이 제한적이라는 점입니다. 색상, 폰트, 도형 스타일 등 세밀한 디자인 조정이 어렵고, 복잡한 레이아웃이나 인터랙티브 요소 구현에는 한계가 있습니다. 표준 플로차트, 시퀀스 다이어그램, 간트 차트 등은 쉽게 그릴 수 있지만, 복잡한 UI 와이어프레임이나 아키텍처 다이어그램에는 부적합할 수 있습니다. 실제로 디자인 팀이나 비주얼 중심의 프로젝트에서는 Mermaid의 단순함이 오히려 제약으로 작용하며, 시각적 완성도를 요구하는 환경에서는 다른 도구와 병행 사용이 필요합니다.

비개발자 접근성

Mermaid는 코드 기반이기 때문에, 비개발자나 시각적 틀에 익숙한 사용자에게는 진입장벽이 있습니다. 문법을 익혀야 하며, 실시간 드래그앤드롭 방식에 비해 직관성이 떨어질 수 있습니다. 하지만 개발자 중심의 팀이나 코드 중심 문서화에는 매우 강력한 도구입니다. 실제로 개발자들은 Mermaid의 간결한 문법과 Git 연동성을 높이 평가하며, 코드 리뷰 과정에서 다이어그램의 변경 이력을 쉽게 추적할 수 있다는 점에서 큰 만족도를 보입니다.

Mermaid-to-Excalidraw 변환 및 하이브리드 워크플로

최근에는 Mermaid 코드를 Excalidraw JSON으로 변환하는 mermaid-to-excalidraw 도구가 등장하면서, 두 도구의 장점을 결합한 하이브리드 워크플로가 가능해졌습니다. 예를 들어, ChatGPT에서 Mermaid 코드를 생성한 뒤, 이를 Excalidraw로 변환하여 시각적으로 보정하거나, 협업 환경에서 실시간 편집이 가능합니다. 이 방식은 코드 기반 자동화와 손그림 스타일의 자유로움을 모두 활용할 수 있게 해줍니다. 실제로 조직 내에서는 “자동화된 코드 → 시각적 보정 → 공식 문서화”의 프로세스가 정착되고 있으며, 개발자와 디자이너 간의 협업 효율성이 크게 향상되고 있습니다.

2.1.3 tldraw과 Miro: 각각의 차별점

tldraw와 Miro는 각각 프로토타이핑과 엔터프라이즈 협업에 특화된 도구로, Excalidraw와는 또 다른 차별성을 가지고 있습니다. 이 섹션에서는 두 도구의 주요 기능과 생태계, 그리고 AI 연동 측면에서의 격차를 구체적으로 비교합니다.

tldraw의 프로토타이핑 특화

tldraw는 “Make Real” 기능을 통해, 스케치한 UI를 실제 코드 컴포넌트로 변환하는 프로

토타이핑에 특화된 도구입니다. React, Figma 등 프론트엔드 생태계와의 연동성이 뛰어나며, 디자이너와 개발자 간의 핸드오프를 간소화합니다. 손그림 스타일과 정형화된 컴포넌트 변환을 동시에 지원하여, 빠른 UI 설계와 실제 구현 간의 간극을 줄여줍니다. 실제로 스타트업이나 웹 서비스 개발팀에서는 tldraw를 활용하여 초기 UI 스케치에서 코드 구현까지의 프로세스를 단축하고 있습니다. tldraw의 오픈소스 생태계는 빠르게 성장하고 있으며, 다양한 커스텀 플러그인과 API 연동이 활발하게 이루어지고 있습니다.

Miro의 엔터프라이즈 협업 워크숍

Miro는 대규모 워크숍, 디자인 스프린트, 전략 기획 등 엔터프라이즈 협업에 최적화된 화이트보드 플랫폼입니다. 수십 명이 동시에 접속하여 실시간으로 의견을 나누고, 포스트잇, 투표, 타임라인 등 다양한 협업 도구를 제공합니다. API 및 외부 플러그인 연동도 활발하며, 조직 내 다양한 워크플로우와 통합이 용이합니다. 실제로 글로벌 컨설팅 기업이나 대기업에서는 Miro를 활용한 워크숍과 프로젝트 관리가 일상적으로 이루어지고 있으며, 복잡한 협업 환경에서도 높은 생산성을 유지할 수 있습니다. Miro의 SaaS 기반 서비스는 보안과 접근성 측면에서도 강점을 가지고 있습니다.

커뮤니티 규모와 생태계

Excalidraw는 GitHub 기준 103K+ Stars, tldraw는 42K+ Stars로 커뮤니티 규모에서 차이가 큼니다. Miro는 상용 SaaS로 오픈소스 커뮤니티 규모는 공개되지 않으나, 엔터프라이즈 시장에서의 점유율이 높습니다. Excalidraw는 오픈소스 생태계의 확장성, 커스텀 라이브러리, AI 연동 등에서 우위를 보입니다. 실제로 Excalidraw는 다양한 커뮤니티 플러그인과 Skill이 활발히 개발되고 있으며, AI와의 연동성 측면에서도 빠르게 진화하고 있습니다. 반면 tldraw는 프로토타이핑 특화 생태계에 집중하고 있으며, Miro는 엔터프라이즈 협업에 최적화된 플랫폼으로 자리잡고 있습니다.

AI 연동 및 자동화 격차

tldraw는 아직 공식적인 AI 연동 기능이 없으며, Miro는 일부 AI 플러그인을 제공하지만, Excalidraw처럼 MCP 프로토콜 기반의 자연어 → 실시간 다이어그램 생성은 지원하지 않습니다. 자동화 및 AI 기반 워크플로우 통합에서는 Excalidraw가 한발 앞서 있습니다. 실제로 Excalidraw는 Claude, ChatGPT 등 다양한 LLM과의 실시간 연동을 통해 자동화된 다이어그램 생성과 피드백 루프를 지원하고 있으며, 조직 내 AI 기반 설계 자동화의 핵심 도구로 부상하고 있습니다. tldraw와 Miro는 이러한 AI 연동 측면에서 아직 제한적인 기능만을 제공하고 있습니다.

2.2 AI 통합 관점의 경쟁 우위

AI 시대의 다이어그램 도구는 단순한 그림 그리기를 넘어, 자연어 프롬프트와 LLM, 자동화 파이프라인과의 통합이 핵심 경쟁력으로 부상하고 있습니다. Excalidraw, draw.io, Mermaid, tldraw 등 주요 도구들은 각기 다른 방식으로 AI 연동을 지원하거나, 해당 영역에서 차별화된 기능을 제공합니다. 이 섹션에서는 MCP 프로토콜 등 AI 통합 관점에서의 경쟁 우위를 심층적으로 분석합니다.

2.2.1 MCP 서버 기반 AI 연동 에코시스템 비교

MCP(Model Context Protocol)는 AI와 다이어그램 도구 간의 연동을 위한 오픈 프로토콜로, 자연어 프롬프트를 실시간 다이어그램으로 변환하는 핵심 역할을 합니다. 각 도구별로 MCP 연동 방식과 AI 통합 수준이 다르며, 이로 인해 실무에서의 활용성과 자동화 가능성이 크게 달라집니다. 본 섹션에서는 Excalidraw, draw.io, Mermaid, tldraw의 MCP 연동 현황과 AI 에코시스템의 차별점을 구체적으로 비교합니다.

Excalidraw의 MCP 공식 연동

Excalidraw는 MCP(Model Context Protocol) 서버를 공식적으로 지원하여, Claude, ChatGPT, VS Code 등 다양한 AI 클라이언트와의 실시간 연동이 가능합니다. 자연어 프롬프트를 입력하면, MCP 서버가 이를 Excalidraw JSON 오브젝트로 변환하여 캔버스에 즉시 반영합니다. 커뮤니티에서 개발된 다양한 Skill(예: coleam00/excalidraw-diagram-skill, yctim-lin/mcp_excalidraw 등)이 존재하며, 스트리밍 기반의 다이어그램 생성, 뷰포트 제어, 반복적 피드백 루프 등 고급 기능을 제공합니다. 실제로 조직 내에서는 AI를 활용한 설계 자동화와 실시간 협업이 가능해져, 생산성과 혁신성이 크게 향상되고 있습니다. Excalidraw의 MCP 연동은 오픈소스 생태계에서 빠르게 확장되고 있으며, 다양한 AI 모델과의 호환성을 지속적으로 강화하고 있습니다.

draw.io의 MCP 경유 연동

draw.io 역시 MCP 프로토콜을 경유하여 LLM 기반 자연어 다이어그램 생성을 지원하지만, 공식적인 실시간 캔버스 렌더링은 Excalidraw에 비해 제한적입니다. 주로 텍스트 → 다이어그램 코드 변환 후, 수동으로 다이어그램을 삽입하는 방식이 일반적입니다. LLM 백엔드 커스터마이징이 가능하지만, 실시간성이나 자동화 수준에서는 Excalidraw에 미치지 못합니다. 실제로 draw.io

에서는 AI를 활용한 자동화 기능이 점진적으로 확대되고 있지만, 실시간 협업과 자동화된 피드백 루프 구현에는 한계가 있습니다. 조직 내에서는 공식 문서화와 표준 설계에 AI를 활용하는 사례가 늘고 있지만, 실시간성 측면에서는 Excalidraw가 우위를 보입니다.

Mermaid의 텍스트 생성 및 토큰 효율

Mermaid는 LLM이 직접 Mermaid 코드를 생성하여 다이어그램을 그릴 수 있도록 설계되었습니다. 텍스트 기반이기 때문에 토큰 효율이 매우 높으며, RAG 시스템이나 자동 문서화 파이프라인에서 널리 활용됩니다. 다만, 실시간 캔버스 렌더링이나 인터랙티브 편집은 지원하지 않으므로, Excalidraw와의 하이브리드 연동이 필요할 수 있습니다. 실제로 AI를 활용한 자동화된 문서화와 코드 리뷰 과정에서 Mermaid가 많이 사용되고 있으며, Excalidraw와의 변환 워크플로를 통해 시각적 보정과 협업이 가능해지고 있습니다.

tldraw의 AI 연동 부재

tldraw는 현재 공식적인 AI 연동 기능이 없습니다. 커뮤니티 플러그인이나 API를 통한 일부 자동화는 가능하지만, MCP 프로토콜 기반의 자연어 → 다이어그램 생성은 지원하지 않습니다. tldraw는 주로 프로토타이핑과 UI 설계에 집중하고 있으며, AI 기반 자동화 기능은 아직 초기 단계에 머물러 있습니다. 향후 AI 연동 기능이 추가될 가능성은 있지만, 현재로서는 Excalidraw와 draw.io에 비해 경쟁력이 떨어집니다.

비교표: AI 연동 에코시스템

도구	MCP 공식 지원	자연어 → 실시간 캔버스	커뮤니티 Skill	LLM 토큰 효율	하이브리드 워크플로
Excalidraw	예	예	풍부	중	예
draw.io	예(제한적)	제한적	일부	중	예
Mermaid	아니오	아니오	일부	매우 높음	예
tldraw	아니오	아니오	미지원	중	제한적

위 비교표는 각 도구의 AI 연동 수준과 MCP 프로토콜 지원 현황을 한눈에 보여줍니다. Excalidraw는 실시간 캔버스 연동과 커뮤니티 Skill의 풍부함, 하이브리드 워크플로 지원에서 가장 앞서 있으며, draw.io는 공식 문서화와 표준 설계에 강점을 보입니다. Mermaid는 LLM 토큰 효율성과 자동화된 문서화에서 우위를 보이고, tldraw는 프로토타이핑 특화 도구로 AI 연동은 아직 미흡한 상황입니다. 조직 내 AI 기반 설계 자동화와 협업 효율성을 극대화하려면 각 도구의 특성과

연동 방식을 충분히 이해하고, 목적에 맞는 병행 전략을 수립하는 것이 중요합니다.

2.2.2 3도구 병행 전략: Excalidraw + draw.io + Mermaid

다이어그램 도구를 단일 솔루션으로만 활용하는 것이 아니라, 각 도구의 장점을 병행하여 조직 내 워크플로를 최적화하는 전략이 점점 더 중요해지고 있습니다. Excalidraw, draw.io, Mermaid는 서로 다른 특성과 기능을 가지고 있으므로, 목적에 따라 병행 사용이 효과적입니다. 본 섹션에서는 3도구 병행 전략과 하이브리드 워크플로의 구체적인 적용 사례를 소개합니다.

비공식 스케치/협업: Excalidraw

Excalidraw는 빠른 아이디어 스케치, 브레인스토밍, 실시간 협업에 최적화되어 있습니다. 손그림 스타일의 자유로움과 웹 브라우저 기반의 즉시 접근성, 실시간 동시 편집 기능이 결합되어, 팀 내 비공식 논의나 초기 설계, 워크숍에서 탁월한 생산성을 제공합니다. URL 공유만으로 즉시 협업을 가능하며, E2E 암호화로 보안성도 확보할 수 있습니다. 실제로 스타트업이나 소규모 개발팀에서는 Excalidraw를 활용한 실시간 아이디어 공유와 빠른 설계 검토가 일상적으로 이루어지고 있습니다. Excalidraw의 직관적인 UI와 자유로운 표현 방식은 창의적 문제 해결과 팀 내 커뮤니케이션을 강화하는데 큰 역할을 합니다.

공식 문서/표준 설계: draw.io

draw.io는 UML, BPMN, ERD 등 표준 표기법을 요구하는 공식 문서, 보고서, 시스템 설계에 적합합니다. Atlassian Confluence, Jira 등 엔터프라이즈 협업 툴과의 통합이 강점이며, 복잡한 도형과 템플릿, 자동 레이아웃 기능이 생산성을 높입니다. 완성도 높은 결과물이 필요한 환경에서 표준 도구로 활용됩니다. 실제로 대기업이나 규제 산업에서는 draw.io를 활용한 표준화된 설계 문서와 보고서 작성이 필수적이며, 조직 내 품질 관리와 검증 프로세스에서도 중요한 역할을 합니다.

코드 내 문서화: Mermaid

Mermaid는 코드 주석, README, 개발 문서 등 코드와 밀접한 영역에서 다이어그램을 관리하는 데 최적화되어 있습니다. Git 기반 버전 관리, LLM 토큰 효율성, GitHub 네이티브 렌더링 등 개발자 친화적인 특성이 돋보입니다. 코드 리뷰, 자동화된 문서화 파이프라인에서 강력한 도구로 자리 잡고 있습니다. 실제로 오픈소스 프로젝트에서는 Mermaid를 활용한 자동화된 설계 문서와 코드 리뷰가 일상적으로 이루어지고 있으며, 개발자 간의 협업과 문서화 효율성이 크게 향상되고 있습니다.

연계 방식과 하이브리드 워크플로

세 도구는 상호 변환 및 연계가 가능합니다. 예를 들어, Mermaid 코드를 Excalidraw로 변환하여 시각적으로 보정하거나, Excalidraw 다이어그램을 draw.io로 내보내어 공식 문서에 삽입할 수 있습니다. mermaid-to-excalidraw, excalidraw-converter 등 변환 도구가 활발히 개발되고 있으며, 조직 내에서는 “스케치 → 보정 → 공식화”의 3단계 워크플로가 자연스럽게 정착되고 있습니다. 실제 사례로, 개발팀에서는 초기 아이디어를 Excalidraw로 스케치한 뒤, Mermaid 코드로 자동화하여 버전 관리하고, 최종 설계는 draw.io로 공식화하는 방식이 널리 활용되고 있습니다. 이러한 하이브리드 워크플로는 각 도구의 장점을 극대화하면서, 조직 내 문서화와 협업의 효율성을 높이는 데 매우 효과적입니다.

2.3 Excalidraw의 알려진 한계

Excalidraw는 혁신적인 오픈소스 다이어그램 도구로 빠르게 확산되고 있지만, 기업 도입 시 반드시 인지해야 할 기능적 한계와 제약이 존재합니다. 이 섹션에서는 실무 관점에서 발생할 수 있는 주요 격차와, 향후 보완 로드맵을 균형 잡힌 시각으로 정리합니다.

2.3.1 기업 도입 시 인지해야 할 기능 격차

Excalidraw는 자유로운 손그림 스타일과 실시간 협업 기능으로 많은 조직에서 빠르게 도입되고 있지만, 엔터프라이즈 환경에서는 몇 가지 중요한 기능적 한계가 존재합니다. 기업 도입 시에는 이러한 격차를 충분히 검토하고, 필요에 따라 대안 도구나 보완 전략을 마련해야 합니다. 본 섹션에서는 Excalidraw의 주요 한계와 실무에서 고려해야 할 사항을 상세히 설명합니다.

UML/BPMN 표준 표기법 미지원

Excalidraw는 손그림 스타일의 자유로운 스케치에 초점을 맞추고 있어, UML, BPMN 등 표준 표기법을 공식적으로 지원하지 않습니다. 복잡한 시스템 설계나 규제 산업에서 요구하는 표준화된 다이어그램 작성에는 draw.io, Lucidchart 등 대안 도구가 필요할 수 있습니다. 커뮤니티 라이브러리를 통해 일부 도형을 추가할 수 있으나, 표준 검증이나 자동화 수준에서는 한계가 존재합니다. 실제로 엔터프라이즈 환경에서는 표준화된 설계 문서와 규제 준수가 필수적이므로, Excalidraw 단독 사용은 제한적일 수 있습니다. 조직 내에서는 Excalidraw를 비공식 스케치와 협업에 활용하고, 공식 설계와 보고서는 draw.io 등 표준 도구로 작성하는 병행 전략이 일반적입니다.

SSO/RBAC/감사로그 부재(오픈소스 버전)

오픈소스 버전의 Excalidraw는 SSO(Single Sign-On), RBAC(Role-Based Access Control), 감사로그 등 엔터프라이즈급 보안 및 관리 기능이 미흡합니다. 사용자 인증과 접근 제어, 변경 이력 추적이 필요한 환경에서는 Excalidraw+ 유료 버전이나 별도의 커스텀 개발이 필요합니다. 특히, 규제 준수와 보안 요구사항이 높은 금융, 의료, 공공기관 등에서는 도입 전 신중한 검토가 필요합니다. 실제로 대규모 조직에서는 SSO와 RBAC, 감사로그 등 보안 기능이 필수적이며, Excalidraw 오픈소스 버전은 이러한 요구를 충족시키지 못하는 경우가 많습니다. Excalidraw+ 유료 버전은 SOC 2 인증을 완료하여 보안성과 신뢰성을 강화하고 있으며, 엔터프라이즈 기능의 지속적인 보완이 이루어지고 있습니다.

손그림 스타일의 규제 산업 부적합성

Excalidraw의 손그림 스타일은 창의적이고 자유로운 아이디어 표현에 적합하지만, 완성도와 일관성을 요구하는 규제 산업(예: 제약, 항공, 정부기관)에서는 부적합할 수 있습니다. 표준화된 도형, 픽셀 단위 정렬, 공식적 문서화가 필수인 경우, Excalidraw의 비정형적 스타일이 오히려 리스크가 될 수 있습니다. 실제로 규제 산업에서는 다이어그램의 정확성과 일관성이 중요하므로, Excalidraw의 자유로운 표현 방식은 공식 문서화에 적합하지 않은 경우가 많습니다. 이러한 환경에서는 draw.io, Lucidchart 등 표준화된 도구와 병행 사용이 필요합니다.

프레젠테이션 및 PowerPoint 내보내기 한계

Excalidraw는 현재 프레젠테이션 기능이 미완성 단계이며, PowerPoint로의 직접 내보내기를 지원하지 않습니다. PNG/SVG 내보내기 후 삽입, SVG → EMF 변환 등 우회 전략이 필요합니다. 공식 로드맵에는 프레젠테이션 및 다양한 내보내기 포맷 지원이 포함되어 있으나, 실무에서는 여전히 한계로 작용할 수 있습니다. 실제로 조직 내에서는 Excalidraw 다이어그램을 PowerPoint나 기타 프레젠테이션 도구에 삽입할 때 추가적인 변환 작업이 필요하며, 이로 인해 업무 효율성이 저하될 수 있습니다. 향후 프레젠테이션 기능과 다양한 내보내기 포맷 지원이 강화될 예정이지만, 현재로서는 제한적인 활용만 가능합니다.

엔터프라이즈 보안 로드맵

Excalidraw+ 유료 버전은 SOC 2 인증을 완료하였으며, SSO/SAML, RBAC, 감사로그, 프레젠테이션 등 엔터프라이즈 기능을 지속적으로 보완하고 있습니다. 오픈소스 버전에도 일부 기능이 순차적으로 반영될 예정이지만, 대규모 조직에서는 도입 전 기능 격차를 반드시 점검해야 합니다. 실제로 Excalidraw+는 엔터프라이즈 환경에서 요구되는 보안과 관리 기능을 지속적으로 강화하고

있으며, 조직 내 도입 시에는 기능별 차이와 보완 로드맵을 충분히 검토하는 것이 중요합니다. 향후 오픈소스 버전에도 엔터프라이즈 기능이 점진적으로 확대될 전망이지만, 현재로서는 유료 버전과의 기능 격차가 존재합니다.

3장: AI 시대의 Prompt Drawing — Excalidraw와 AI Agent 연동

3.1 MCP 프로토콜 기반 AI 다이어그램 생성

AI 시대의 다이어그램 작성은 더 이상 수동 드래그앤드롭에 머무르지 않습니다. Excalidraw는 MCP(Model Context Protocol) 기반의 AI 연동을 통해, 자연어 프롬프트만으로 실시간 다이어그램을 자동 생성하는 혁신적 워크플로를 제공합니다. 이 섹션에서는 Excalidraw 공식 MCP 서버의 구조와 설치 방법, Claude Code Skill을 비롯한 커뮤니티 MCP 생태계, 그리고 ChatGPT와 Mermaid, Excalidraw를 결합한 하이브리드 파이프라인까지, 최신 AI 다이어그램 생성 기술의 전모를 상세히 다룹니다. MCP 기반의 AI 통합은 개발자와 아키텍트, DevOps, 교육자 등 다양한 직군에서 생산성 혁신을 이끄는 핵심 요소로 부상하고 있습니다.

3.1.1 Excalidraw 공식 MCP 서버: 구조와 설치

Excalidraw 공식 MCP 서버는 AI와 Excalidraw 캔버스를 연결하는 중추적 역할을 합니다. 이 서버는 WebSocket 기반 실시간 통신을 지원하며, AI가 생성한 다이어그램 데이터를 스트리밍 방식으로 클라이언트에 전송합니다. 뷰포트 카메라 제어 기능을 통해 AI가 특정 영역을 확대하거나 이동할 수 있고, 인터랙티브 풀스크린 편집 모드를 제공하여 인간 사용자와 AI가 동시에 캔버스에서 작업할 수 있습니다. 이러한 구조는 복잡한 아키텍처 다이어그램이나 대규모 시스템 설계 등에서, 반복적 피드백과 실시간 협업을 가능하게 합니다.

MCP 서버의 핵심 기능은 실시간 데이터 교환과 협업 지원에 있습니다. 서버는 여러 클라이언트가 동시에 접속해 다이어그램을 수정할 수 있도록 WebSocket을 통한 양방향 통신을 제공하며, AI가 생성한 요소를 실시간으로 캔버스에 반영합니다. 예를 들어, 사용자가 프롬프트를 입력하면 AI가 다이어그램 요소를 생성하고, 서버가 이를 클라이언트에 스트리밍하여 즉시 시각화할 수

있습니다. 뷰포트 카메라 제어 기능은 AI가 특정 영역을 자동으로 확대하거나 이동시켜, 복잡한 구조를 효과적으로 탐색할 수 있게 합니다. 또한, 풀스크린 인터랙티브 모드는 인간 사용자와 AI가 동시에 캔버스에서 작업하며, 실시간 피드백과 반복적 개선이 가능합니다. 이러한 기능은 아키텍처 설계, 시스템 다이어그램, 교육용 워크플로 등 다양한 분야에서 활용되고 있습니다.

설치 및 배포 방법은 GitHub의 [excalidraw/excalidraw-mcp](https://github.com/excalidraw/excalidraw-mcp) 저장소에서 소스 코드를 제공하며, Vercel을 통한 원클릭 배포, Docker 컨테이너, 직접 Node.js 실행 등 다양한 방식으로 설치할 수 있습니다. 설치 과정은 다음과 같습니다.

1. [Release 페이지](#)에서 최신 .mcpb 바이너리 파일을 다운로드하거나, 소스 코드를 클론하여 `npm install && npm run build`로 빌드합니다.
2. 환경설정 파일(.env 또는 config.json)에 포트, 인증, WebSocket 옵션을 지정합니다.
3. Vercel 배포의 경우, GitHub 저장소를 연결하고 환경 변수를 설정한 뒤 배포 버튼을 클릭하면 자동으로 MCP 서버가 생성됩니다.

MCP 호환 클라이언트 연동은 Claude Desktop, ChatGPT, VS Code, Goose 등 다양한 MCP 호환 클라이언트에서 Excalidraw MCP 서버를 사용할 수 있습니다. 각 클라이언트는 설정 파일에 MCP 서버의 엔드포인트를 지정하거나, .mcpb 파일을 직접 불러와 연동합니다. 예를 들어, Claude Desktop에서는 Skill Store에서 Excalidraw Skill을 설치하고, MCP 서버 주소를 입력하면 즉시 다이어그램 생성이 가능합니다. VS Code의 경우, 확장 프로그램 설정에서 MCP 서버 URL을 추가하면 코드 내에서 다이어그램을 바로 생성할 수 있습니다.

운영 및 보안 고려사항으로는 MCP 서버가 실시간 협업과 AI 통합을 위해 외부 네트워크에 노출되는 경우가 많으므로, HTTPS 적용, 인증 토큰, CORS 정책 등 보안 설정이 필수적입니다. 대규모 사용 환경에서는 WebSocket 커넥션 풀, 로드밸런서, 장애 복구 전략을 함께 고려해야 합니다. 예를 들어, 여러 사용자가 동시에 접속하는 환경에서는 커넥션 풀을 통해 안정적인 통신을 유지하고, 장애 발생 시 자동 복구 기능을 적용하여 서비스 연속성을 보장할 수 있습니다. 또한, 인증 토큰과 CORS 정책을 통해 외부 접근을 제한하고, 데이터 유출을 방지할 수 있습니다. 이러한 보안과 운영 전략은 기업 환경에서 MCP 서버를 안정적으로 운영하는 데 필수적입니다.

3.1.2 Claude Code Skill과 커뮤니티 MCP 생태계

Excalidraw의 AI 연동은 공식 Skill뿐 아니라 커뮤니티에서 활발하게 개발되는 다양한 Skill을 통해 확장됩니다. 대표적으로 coleam00/excalidraw-diagram-skill, yctimlin/mcp_excalidraw, edwingao28/excalidraw-skill 등이 있습니다. 이들 Skill은 Playwright 기반 렌더 파이프라인, 풀 캔버스 생성/검사/반복 수정, 대형 다이어그램 분할 등 고유한 기능을 제공합니다.

대표 Skill 및 생태계 개요를 살펴보면, Excalidraw MCP 생태계는 공식 Skill 외에도 다양한 오픈소스 Skill이 존재하며, 각 Skill은 특정 워크플로와 기능에 특화되어 있습니다. coleam00/excalidraw-diagram-skill은 Playwright 기반 자동 렌더링과 피드백 루프를 제공하며, yctimlin/mcp_excalidraw는 대형 다이어그램 분할과 반복적 설계에 강점을 가지고 있습니다. edwingao28/excalidraw-skill은 경량화된 워크플로와 빠른 응답성으로 실무에서 활용도가 높습니다. 커뮤니티 Skill은 GitHub에서 활발하게 업데이트되고 있으며, 다양한 사용자 피드백을 반영해 기능이 지속적으로 개선되고 있습니다.

coleam00/excalidraw-diagram-skill의 특징은 Playwright를 활용해 AI가 다이어그램을 생성한 뒤, 실제 렌더링 결과를 스크린샷으로 확인하고, 필요시 레이아웃을 자동으로 수정하는 피드백 루프를 구현합니다. 덕분에 AI가 생성한 도형 위치나 크기가 어긋난 경우, 반복적으로 개선 명령을 내릴 수 있습니다. 이는 복잡한 시스템 아키텍처나 계층적 구조에서 특히 유용합니다. 예를 들어, 복잡한 네트워크 구조를 설계할 때, AI가 생성한 초기 다이어그램을 스크린샷으로 검토한 뒤, “서비스 간 연결을 더 명확하게 표시해줘”와 같은 추가 프롬프트를 입력하면, Skill이 자동으로 레이아웃을 조정하여 가독성을 높일 수 있습니다. 이러한 피드백 루프는 실무 설계에서 반복적 개선과 협업 효율성을 극대화합니다.

yctimlin/mcp_excalidraw의 풀 캔버스 툴킷은 생성, 검사, 반복 수정 등 캔버스 작업 전체를 자동화하는 툴킷을 제공합니다. 사용자는 자연어로 “API Gateway와 5개의 마이크로서비스를 그려줘”와 같이 요청하면, Skill이 전체 구조를 생성하고, 각 요소의 위치와 연결관계를 자동으로 최적화합니다. 반복 프롬프트를 통해 세부 요소를 추가하거나 수정할 수 있어, 실무에서 점진적 설계가 가능합니다. 예를 들어, 대규모 시스템 설계에서 “서비스별로 데이터베이스를 분리해줘”와 같은 추가 요청을 하면, Skill이 기존 다이어그램을 분석하여 각 서비스에 별도의 데이터베이스를 자동 배치합니다. 이러한 자동화와 반복적 개선 기능은 복잡한 시스템 설계와 변경 관리에 매우 효과적입니다.

대형 다이어그램 분할 전략은 AI Skill의 토큰 제한(~32,000 토큰)으로 인해, 대규모 다이어그램은 자동으로 여러 개의 청크로 분할되어 처리됩니다. 예를 들어, 50개 이상의 컴포넌트가 포함된 아키텍처에서는 “1단계: 인프라 계층”, “2단계: 서비스 계층” 등으로 분할 생성한 뒤, 최종적으로 하나의 Excalidraw 파일로 병합하는 전략이 권장됩니다. 이 방식은 복잡한 시스템 설계에서 토큰 제한을 극복하고, 각 계층별로 세부 설계를 반복적으로 개선할 수 있게 해줍니다. 분할 생성된 다이어그램은 최종적으로 병합하여 전체 구조를 시각화할 수 있으며, 팀 내 협업과 문서화에도 용이합니다.

최적의 Skill 선택 가이드는 실제 사용 경험과 커뮤니티 피드백을 종합하면, coleam00/excalidraw-diagram-skill이 가장 안정적이고, 복잡한 다이어그램에서도 반복적 개선이 뛰어나다는 평가를 받고 있습니다. yctimlin/mcp_excalidraw는 대규모 시스템이나 반복 설계에 강점을 보이며, edwingao28/excalidraw-skill은 경량화된 워크플로와 빠른 응답 속도가 특징입니다. 사용자는 프로젝트의 규모와 요구사항에 따라 적합한 Skill을 선택할 수 있으며, 각 Skill의 GitHub 저장소에서 설치 방법과 사용 예제를 참고하면 실무 적용이 더욱 용이합니다. 커뮤니티의 활발한 개발과 피드백 덕분에 Skill의 기능과 안정성이 지속적으로 향상되고 있습니다.

3.1.3 ChatGPT + Mermaid + Excalidraw 하이브리드 파이프라인

ChatGPT, Mermaid, Excalidraw를 결합한 하이브리드 파이프라인은 AI 기반 다이어그램 생성의 실전 효율을 극대화합니다. 전체 워크플로는 (1) ChatGPT에서 Mermaid 코드 생성, (2) @excalidraw/mermaid-to-excalidraw 라이브러리로 변환, (3) Excalidraw에서 시각적 보정의 3단계로 구성됩니다.

파이프라인 개요와 단계별 흐름을 살펴보면, 사용자는 ChatGPT에게 자연어 프롬프트를 입력하여 Mermaid 코드(플로차트, 시퀀스 다이어그램 등)를 생성하고, 이를 @excalidraw/mermaid-to-excalidraw 라이브러리나 온라인 플레이그라운드에서 Excalidraw JSON 포맷으로 변환합니다. 변환된 다이어그램은 Excalidraw에서 시각적 보정과 추가 편집을 통해 최종 결과물을 완성할 수 있습니다. 이 과정은 반복적 설계와 협업, 코드-다이어그램 동기화에 매우 효과적이며, 실무에서 빠른 피드백과 문서화 자동화를 지원합니다.

ChatGPT에서 Mermaid 코드 생성 단계에서는 사용자가 “Kubernetes 클러스터 구조를 Mermaid로 그려줘”와 같이 자연어로 요청하면, ChatGPT가 자동으로 Mermaid 플로차트, 시퀀

스 다이어그램, 클래스 다이어그램 등 텍스트 코드를 생성합니다. Mermaid는 텍스트 기반이므로 LLM 토큰 소모가 적고, 버전 관리에도 용이합니다. 예를 들어, 복잡한 시스템 구조를 텍스트 코드로 표현하면, GitHub에서 버전 관리와 협업이 가능하며, 변경 이력을 쉽게 추적할 수 있습니다.

Mermaid-to-Excalidraw 변환 단계에서는 생성된 Mermaid 코드를 [@excalidraw/mermaid-to-excalidraw](#) 라이브러리 또는 [Mermaid 변환 플레이그라운드](#)에서 Excalidraw JSON 포맷으로 변환합니다. 이 과정에서 Mermaid의 노드, 엣지, 레이블이 Excalidraw 요소로 자동 매핑되며, 복잡한 구조도 손쉽게 시각화할 수 있습니다. 예를 들어, 플로차트의 각 단계가 Excalidraw 도형으로 변환되고, 연결선과 레이블이 자동으로 배치되어 시각적 일관성을 확보할 수 있습니다.

Excalidraw에서 시각적 보정 단계에서는 변환된 다이어그램을 Excalidraw에서 추가 편집, 색상/레이어 조정, 커뮤니티 아이콘 라이브러리 적용 등 시각적 보정이 가능합니다. “Excalidraw Diagram Creator” GPT(ChatGPT Store)와 같은 GPT 기반 툴을 활용하면, 변환과정 전체를 자동화할 수도 있습니다. 사용자는 다이어그램의 레이아웃을 미세 조정하거나, 아이콘 라이브러리를 적용하여 전문성을 높일 수 있습니다. 또한, 팀 내 협업을 통해 실시간으로 다이어그램을 수정하고, 최종 결과물을 내보내기과 공유 기능을 활용할 수 있습니다.

활용상의 장점과 유의점으로는 이 하이브리드 파이프라인이 반복적 설계, 협업, 코드-다이어그램 동기화에 매우 효과적이라는 점이 있습니다. 단, Mermaid-to-Excalidraw 변환은 플로차트와 시퀀스 다이어그램에 최적화되어 있으므로, 복잡한 커스텀 도형이나 레이아웃은 수동 보정이 필요할 수 있습니다. 예를 들어, 복잡한 네트워크 구조나 특수한 도형이 포함된 경우, 변환 후 Excalidraw에서 직접 레이아웃을 조정하거나 아이콘을 추가해야 최적의 결과물을 얻을 수 있습니다. 이러한 유의점은 실무 적용 시 사전 테스트와 반복적 개선을 통해 극복할 수 있습니다.

3.2 실전 Prompt Drawing 워크플로

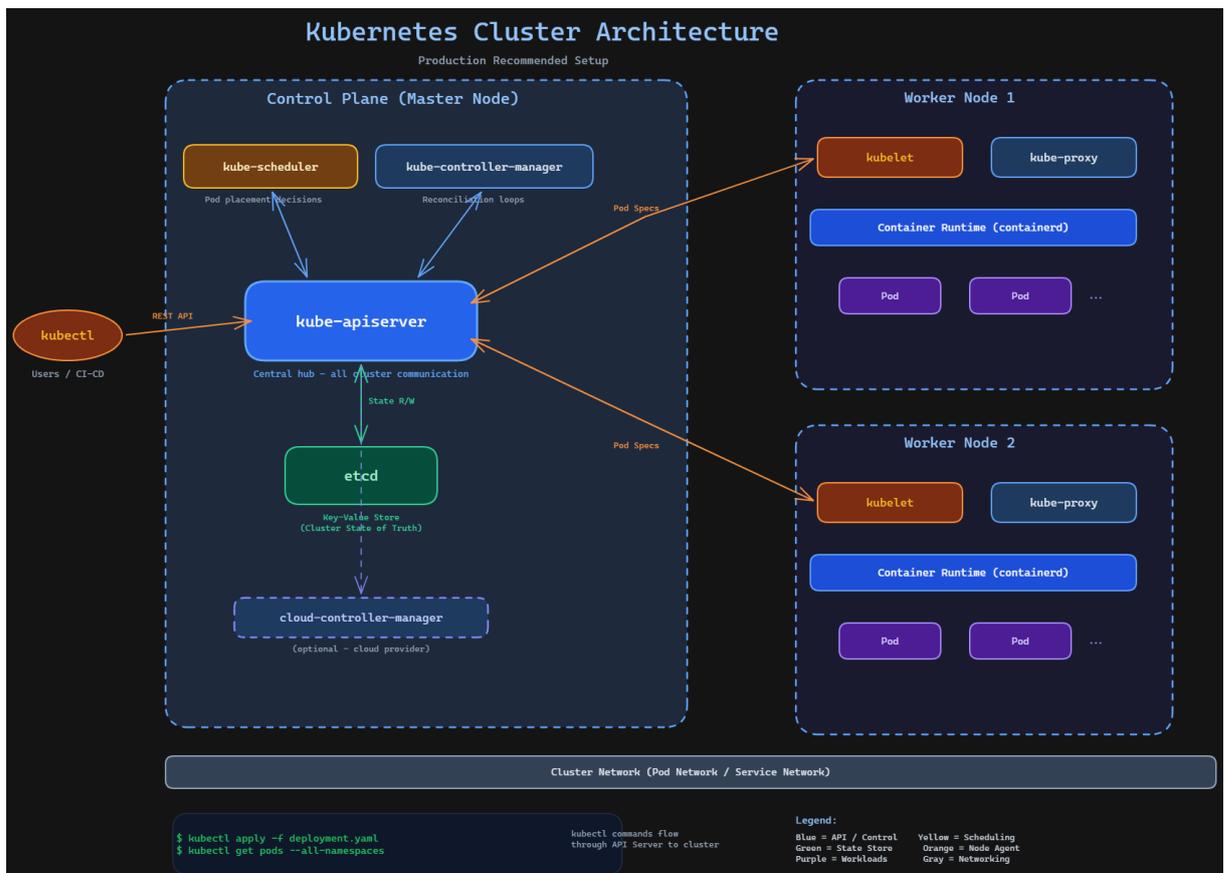
AI 기반 Prompt Drawing은 단순한 자동화가 아닌, 반복적 피드백과 점진적 개선을 통해 인간-기계 협업의 새로운 패러다임을 제시합니다. 이 섹션에서는 자연어 프롬프트를 활용한 아키텍처 다이어그램 생성, Microsoft의 실제 AI Agent 사례, 그리고 Headless 렌더링 및 배치 내보내기 까지, 실무에 즉시 적용 가능한 워크플로를 구체적으로 안내합니다. AI Agent와 Excalidraw의 결합은 IT 의사결정자와 실무자 모두에게 설득력 있는 생산성 혁신을 제공합니다.

3.2.1 자연어로 아키텍처 다이어그램 생성하기

Prompt Drawing을 활용하면 복잡한 아키텍처 설계도 자연어 프롬프트만으로 빠르게 시각화할 수 있습니다. 이 과정은 AI Skill 설치와 연동, 프롬프트 입력, 반복적 개선, 실제 예제와 결과물의 4단계로 구성되며, 실무에서 점진적 설계와 협업에 매우 효과적입니다.

Claude Code Skill 설치와 준비 단계에서는 Claude Desktop 또는 Skill Store에서 “Excalidraw Diagram Skill”을 검색해 설치합니다. MCP 서버 주소를 입력하거나 Skill 설정에서 연동을 마치면, 자연어 프롬프트를 통한 다이어그램 생성이 준비됩니다. VS Code, ChatGPT 등에서도 유사한 방식으로 Skill을 연동할 수 있습니다. 예를 들어, VS Code 확장 프로그램에서 MCP 서버 주소를 입력하면 코드 내에서 다이어그램을 바로 생성할 수 있습니다. Skill 설치와 연동은 GitHub 저장소의 안내 문서를 참고하면 쉽게 진행할 수 있습니다. Claude Code 에서 다음과 같이 실행 하면 다이어그램을 작성합니다.

/excalidraw-diagram Kubernetes Master 와 Worker 노드의 기본 구조 에 맞는 권장 아키텍처를 그려



프롬프트 입력과 다이어그램 생성 단계에서는 사용자가 “마이크로서비스 아키텍처(MSA) 구조를 그려줘”, “API Gateway와 3개의 서비스, 데이터베이스를 포함해줘”와 같이 자연어로

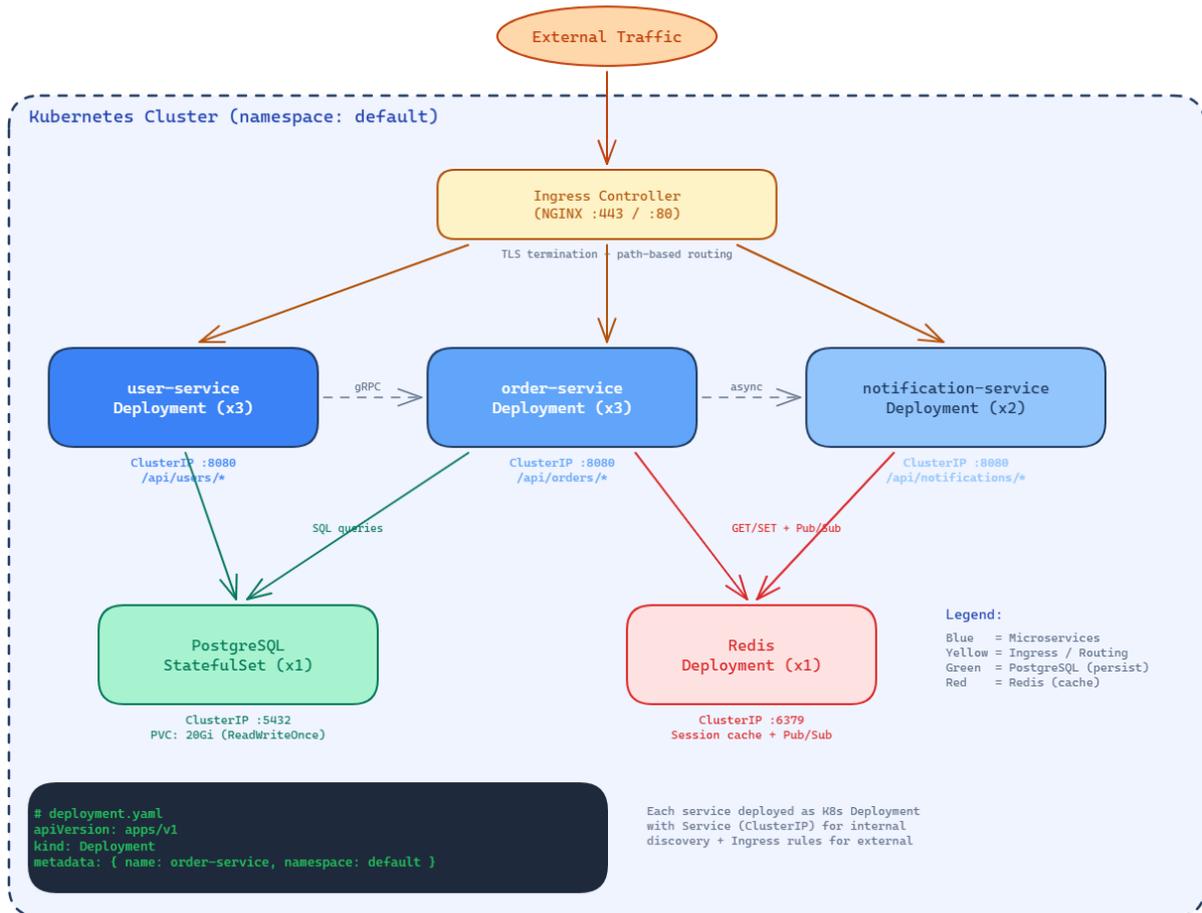
요청합니다. AI는 입력된 프롬프트를 분석하여, 각 컴포넌트의 위치, 연결관계, 레이블 등을 자동으로 배치한 Excalidraw 다이어그램을 실시간으로 생성합니다. 생성된 다이어그램은 즉시 캔버스에 표시되고, 사용자는 추가 설명이나 수정을 반복 요청할 수 있습니다. 예를 들어, “각 서비스에 고유 아이콘을 추가해줘”와 같은 추가 요청을 하면, AI가 자동으로 아이콘을 배치하여 시각적 전문성을 높일 수 있습니다.

반복적 프롬프트와 점진적 개선 단계에서는 실무에서는 한 번의 프롬프트로 완벽한 다이어그램이 나오기 어렵습니다. “API Gateway 아래에 인증 서버를 추가해줘”, “각 서비스에 Redis 캐시를 연결해줘” 등 반복적으로 프롬프트를 입력하면, AI가 기존 다이어그램을 분석해 변경사항만 반영합니다. 이 과정은 인간 설계자의 피드백과 AI의 자동화가 결합된 점진적 개선 패턴으로, 복잡한 시스템 설계에서 특히 유용합니다. 예를 들어, 팀 내에서 여러 명이 동시에 피드백을 주고받으며, 다이어그램을 반복적으로 개선할 수 있습니다.

실제 예제와 결과물 단계에서는 “Kubernetes 클러스터 내에 3개의 마이크로서비스와 Redis, PostgreSQL을 포함한 구조를 그려줘”라는 프롬프트를 입력하면, AI가 각 요소를 적절히 배치한 다이어그램을 생성합니다.

Kubernetes Microservices Architecture

3 Services + Redis + PostgreSQL on K8s



이후 “Ingress와 Service를 추가해줘” 라고 요청하면, 기존 구조에 새로운 컴포넌트가 자동으로 추가됩니다. 최종 결과물은 Excalidraw 캔버스에서 자유롭게 편집, 내보내기, 공유가 가능합니다. 예를 들어, 다이어그램을 PNG/SVG 파일로 내보내거나, 팀원과 실시간 협업을 통해 최적의 결과물을 도출할 수 있습니다. 이러한 워크플로는 실무 설계, 문서화, 프레젠테이션 등 다양한 용도에 즉시 활용할 수 있습니다.

3.2.2 Microsoft Architecture Review Agent 사례

Microsoft의 Architecture Review Agent는 실제 기업 환경에서 AI와 Excalidraw, Azure OpenAI, MCP 서버를 결합하여 자동화된 아키텍처 리뷰와 시각화 기능을 제공합니다. 이 시스템은 IT 의사결정자와 실무자 모두에게 빠른 피드백과 설계 자동화, 문서화 효율성을 제공하며, 대규모 조직에서 반복되는 아키텍처 리뷰와 변경 관리의 부담을 획기적으로 줄일 수 있습니다.

시스템 개요 및 구성 요소를 살펴보면, Architecture Review Agent는 Agent Framework,

Azure OpenAI, Excalidraw MCP 서버로 구성되어 있습니다. 사용자는 자연어로 시스템 요구사항을 입력하면, AI가 아키텍처를 분석하고, Excalidraw 다이어그램으로 시각화한 뒤, 리뷰 결과와 개선점을 자동으로 제안합니다. 예를 들어, “웹 API와 데이터베이스, 워커 서비스를 포함한 아키텍처를 설계해줘” 라고 요청하면, AI가 각 컴포넌트의 위치와 연결관계를 분석하여 다이어그램을 생성하고, 리뷰 결과를 자동으로 제시합니다.

ACR Tasks 기반 자동화 단계에서는 이 시스템은 Azure Container Registry(ACR) Tasks를 활용해, 다이어그램에 명시된 각 서비스별로 컨테이너 이미지를 자동 빌드합니다. 예를 들어, “Web API, Worker, Database” 로 구성된 다이어그램이 생성되면, 각 컴포넌트에 맞는 Dockerfile을 자동으로 생성하고, 빌드 및 배포까지 연동합니다. 이 과정은 YAML 또는 Terraform 코드로 자동 변환되어, 실제 클라우드 인프라에 적용할 수 있습니다. 사용자는 다이어그램을 기반으로 인프라 코드를 자동 생성하여, 배포와 운영을 효율적으로 관리할 수 있습니다.

자동 스케일링 및 운영 최적화 단계에서는 AI Agent가 다이어그램 내 레플리카 수, 오토스케일링 정책 등을 분석하여, 0~5개의 레플리카 자동 스케일링 구성을 제안합니다. 이 과정은 YAML 또는 Terraform 코드로 자동 변환되어, 실제 클라우드 인프라에 적용할 수 있습니다. 예를 들어, “각 서비스에 3개의 레플리카를 배치해줘” 와 같은 요청을 하면, AI가 자동으로 스케일링 정책을 생성하여 인프라 코드에 반영합니다. 이러한 자동화 기능은 운영 효율성과 확장성을 극대화합니다.

IT 의사결정자 관점의 가치에서는 이러한 AI Agent + Excalidraw 결합 사례가 설계-검증-배포의 전 과정을 자동화함으로써, IT 의사결정자에게 빠른 피드백, 오류 감소, 문서화 자동화, 협업 효율성 등 다각도의 가치를 제공합니다. 특히, 대규모 조직에서 반복되는 아키텍처 리뷰와 변경 관리의 부담을 획기적으로 줄일 수 있습니다. 예를 들어, 팀 내에서 여러 명이 동시에 리뷰를 진행하고, 변경 사항을 자동으로 반영하여 문서화와 배포를 효율적으로 관리할 수 있습니다. 이러한 사례는 실무 적용에서 생산성 혁신을 이끄는 핵심 요소로 평가받고 있습니다.

3.2.3 Headless 렌더링과 배치 내보내기

Headless 렌더링은 UI 없이 서버에서 Excalidraw JSON 요소를 받아, PNG/SVG 등 이미지 파일로 자동 변환하는 방식입니다. 이는 CI/CD 파이프라인, 문서 자동화, 대량 다이어그램 산출물 생성 등에서 필수적인 기술입니다.

Headless 렌더링 개념은 서버 환경에서 UI 없이 다이어그램을 이미지 파일로 자동 변환하는

기술로, 대량의 다이어그램을 반복적으로 생성/배포해야 하는 환경에서 생산성을 극대화합니다. 예를 들어, 개발팀이 여러 아키텍처 다이어그램을 자동으로 생성하여 문서화하거나, CI/CD 파이프라인에서 다이어그램을 자동 배포하는 데 활용할 수 있습니다. Headless 렌더링은 서버리스 환경, 배치 작업, API 연동 등 다양한 자동화 시나리오에 적용할 수 있습니다.

excalidraw-brute-export-cli 활용 단계에서는 [excalidraw-brute-export-cli](#)는 Playwright와 Firefox를 기반으로, Excalidraw JSON 파일을 PNG/SVG 이미지로 일괄 변환하는 CLI 도구입니다. 사용자는 “`excalidraw-brute-export -input ./diagrams -output ./images`”와 같이 명령어를 실행하면, 폴더 내 모든 다이어그램을 자동 변환할 수 있습니다. 이 도구는 대량의 다이어그램을 빠르게 이미지 파일로 변환하여, 문서화와 배포를 자동화할 수 있습니다. 예를 들어, 프로젝트 내 여러 다이어그램을 일괄 변환하여, 팀 내 공유와 프레젠테이션에 활용할 수 있습니다.

excalirender 등 기타 도구 단계에서는 [excalirender](#) 등 다른 CLI 도구는 Node.js 환경에서 SVG 내보내기, 커스텀 해상도, 배경 투명화 등 다양한 옵션을 지원합니다. 이들 도구는 서버리스 환경, 배치 작업, API 연동 등 다양한 자동화 시나리오에 활용됩니다. 예를 들어, 다이어그램의 해상도를 조정하거나, 배경을 투명하게 설정하여 프레젠테이션에 최적화된 이미지를 생성할 수 있습니다. 다양한 옵션을 활용하면 프로젝트 요구사항에 맞는 맞춤형 이미지 파일을 생성할 수 있습니다.

활용상의 장점과 유의점 단계에서는 Headless 렌더링은 대량의 다이어그램을 반복적으로 생성/배포해야 하는 환경에서 생산성을 극대화합니다. 단, 복잡한 다이어그램이나 커스텀 폰트, 외부 이미지가 포함된 경우 렌더링 결과가 다를 수 있으므로, 사전 테스트와 옵션 조정이 필요합니다. 예를 들어, 외부 아이콘이나 커스텀 폰트가 포함된 다이어그램은 렌더링 결과가 다를 수 있으므로, 사전에 테스트를 진행하고, 옵션을 조정하여 최적의 결과물을 얻을 수 있습니다. 이러한 유의점은 실무 적용에서 반복적 개선과 협업을 통해 극복할 수 있습니다.

3.3 아키텍처 다이어그램 예제 갤러리

AI 기반 Prompt Drawing의 실질적 가치를 확인하려면, 다양한 아키텍처 예제를 직접 보고 따라 하는 것이 중요합니다. 이 섹션에서는 마이크로서비스 아키텍처(MSA), Kubernetes 클러스터, LLM/RAG 파이프라인 등 실전에서 자주 활용되는 구조를 Excalidraw로 구현한 예제를 제시합니다. 각 예제는 AI 프롬프트 입력, 결과물, 수동 보정의 단계별 과정을 포함하여, 실무 적용에 즉시

활용할 수 있도록 안내합니다.

3.3.1 MSA 설계 다이어그램 예제

마이크로서비스 아키텍처(MSA)는 현대 소프트웨어 개발에서 가장 널리 사용되는 구조 중 하나입니다. 각 서비스가 독립적으로 배포되고, API Gateway를 통해 외부와 통신하며, 이벤트 버스와 데이터베이스가 서비스 간 데이터 흐름을 지원합니다. Excalidraw와 AI 프롬프트를 활용하면 복잡한 MSA 구조도 빠르게 시각화할 수 있습니다. 이 예제에서는 서비스 간 통신 구조, API Gateway와 이벤트 버스 구성, AI 프롬프트 입력과 수동 보정, 단계별 스크린샷과 실무 적용의 4단계로 MSA 설계 다이어그램을 완성합니다.

서비스 간 통신 구조 표현 단계에서는 마이크로서비스 아키텍처(MSA)에서 각 서비스가 독립적으로 배포되고, API Gateway를 통해 외부와 통신합니다. Excalidraw에서는 AI 프롬프트로 “API Gateway, 3개의 서비스, 이벤트 버스, 데이터베이스를 포함한 MSA 구조를 그려줘” 라고 입력하면, 각 요소가 네트워크로 연결된 다이어그램이 자동 생성됩니다. 예를 들어, User Service, Order Service, Payment Service가 각각 독립적으로 배치되고, API Gateway와 연결되어 외부 요청을 처리합니다.

API Gateway와 이벤트 버스 구성 단계에서는 API Gateway는 중앙에 배치되고, 각 마이크로서비스와 양방향 화살표로 연결됩니다. 이벤트 버스(RabbitMQ, Kafka 등)는 서비스들 사이에 위치하며, 비동기 메시지 전달 구조를 시각화합니다. AI가 기본 구조를 생성한 뒤, 사용자는 각 서비스에 고유 아이콘이나 색상을 추가해 가독성을 높일 수 있습니다. 예를 들어, RabbitMQ 아이콘을 이벤트 버스에 적용하고, 각 서비스에 색상을 지정하여 시각적 일관성을 확보할 수 있습니다.

AI 프롬프트 입력과 수동 보정 단계에서는 “User Service, Order Service, Payment Service를 구분해서 표시해줘” 라고 추가 프롬프트를 입력하면, AI가 각 서비스에 레이블을 붙이고, 도형을 그룹화합니다. 이후 수동으로 아이콘 라이브러리를 적용하거나, 레이아웃을 미세 조정해 최종 다이어그램을 완성합니다. 예를 들어, 각 서비스에 AWS, Docker, Kubernetes 아이콘을 적용하여 전문성을 높일 수 있습니다.

단계별 스크린샷과 실무 적용 단계에서는 각 단계(프롬프트 입력 → AI 생성 결과 → 수동 보정)는 스크린샷으로 기록하여, 팀 내 문서화, 리뷰, 프레젠테이션 등에 활용할 수 있습니다. 이 방식은 반복 설계, 협업, 변경 관리에 매우 효과적입니다. 예를 들어, 다이어그램의 변경 이력을

스크린샷으로 기록하여, 팀 내에서 피드백을 주고받으며 최적의 결과물을 도출할 수 있습니다.

3.3.2 Kubernetes 구조와 LLM/RAG 파이프라인 예제

Kubernetes 클러스터와 LLM/RAG 파이프라인은 AI 기반 Prompt Drawing에서 자주 활용되는 구조입니다. 이 예제에서는 Kubernetes 클러스터 구조, LLM/RAG 파이프라인 다이어그램, 커뮤니티 라이브러리 아이콘 활용, 실무 적용과 확장성의 4단계로 실전 예제를 완성합니다.

Kubernetes 클러스터 구조 예제 단계에서는 Kubernetes 클러스터 구조는 Pod, Service, Ingress, Namespace 등 핵심 컴포넌트로 구성됩니다. AI 프롬프트로 “Kubernetes 클러스터 구조를 그려줘. Pod, Service, Ingress, Namespace를 포함해줘” 라고 입력하면, 각 요소가 계층적으로 배치된 다이어그램이 생성됩니다. 커뮤니티 아이콘 라이브러리에서 Kubernetes, Docker, AWS 아이콘을 적용하면, 시각적 일관성과 전문성을 동시에 확보할 수 있습니다. 예를 들어, 클러스터 내에 AWS S3, RDS, EKS 아이콘을 배치하여 클라우드 네이티브 아키텍처를 직관적으로 표현할 수 있습니다.

LLM/RAG 파이프라인 다이어그램 단계에서는 LLM/RAG(Retrieval-Augmented Generation) 파이프라인은 Embedding, Vector Store, Retrieval, Generation의 4단계로 구성됩니다. “LLM RAG 파이프라인 구조를 그려줘. Embedding, Vector Store, Retrieval, Generation 단계를 포함해줘” 라는 프롬프트로, 각 컴포넌트가 순차적으로 연결된 다이어그램이 생성됩니다. AI가 자동으로 각 단계에 설명 레이블을 추가하고, 데이터 흐름을 화살표로 표시합니다. 예를 들어, Embedding 단계에서 데이터가 Vector Store로 저장되고, Retrieval 단계에서 검색된 데이터가 Generation 단계로 전달되는 구조를 시각적으로 표현할 수 있습니다.

커뮤니티 라이브러리 아이콘 활용 단계에서는 AWS, Kubernetes, Docker 등 주요 IT 인프라 아이콘은 libraries.excalidraw.com에서 원클릭으로 추가할 수 있습니다. 예를 들어, Kubernetes 클러스터 내에 AWS S3, RDS, EKS 아이콘을 배치하면, 클라우드 네이티브 아키텍처를 직관적으로 표현할 수 있습니다. 아이콘 라이브러리를 활용하면 다이어그램의 전문성과 가독성을 높일 수 있습니다.

실무 적용과 확장성 단계에서는 이러한 예제는 실무 설계, 교육, 문서화, 프레젠테이션 등 다양한 용도로 즉시 활용 가능합니다. 필요에 따라 다이어그램을 확장하거나, 팀원과 실시간 협업 편집을 통해 최적의 결과물을 도출할 수 있습니다. 예를 들어, 다이어그램을 팀 내에서 공유하고, 실시간으로

로 피드백을 주고받으며 반복적으로 개선할 수 있습니다. 이러한 워크플로는 실무 적용에서 생산성 혁신을 이끄는 핵심 요소로 평가받고 있습니다.

4장: 플랫폼 통합, 라이브러리, 내보내기 활용

Excalidraw는 단순한 웹 기반 다이어그램 도구를 넘어, 다양한 개발 환경과 협업 플랫폼에 통합되어 현대 IT 조직의 시각적 커뮤니케이션을 혁신하고 있습니다. 본 장에서는 VS Code, Obsidian, Confluence, Notion 등 주요 생산성 도구와의 통합 방안, 커뮤니티 기반 아이콘 라이브러리와 템플릿 생태계, PNG/SVG 내보내기 및 PowerPoint 연동 전략, 그리고 Excalidraw의 성능을 뒷받침하는 이중 캔버스 아키텍처와 최적화 기법을 심도 있게 다룹니다. 이를 통해 Excalidraw가 기업 표준 그리기 도구 및 AI Prompt Drawing 워크플로의 핵심 요소로 자리매김하는 이유를 구체적으로 설명합니다.

4.1 개발 도구 통합

Excalidraw는 현대 개발자와 지식 노동자의 다양한 워크플로에 자연스럽게 녹아들 수 있도록, 주요 개발 도구 및 엔터프라이즈 위키와의 통합을 적극 지원합니다. 이 섹션에서는 VS Code, Obsidian, Confluence, Notion 등과의 연동 방식과 각 환경에서의 활용 가치를 다각적으로 분석합니다. 이를 통해 Excalidraw가 단일 툴을 넘어 조직 내 시각적 협업과 지식 관리를 촉진하는 플랫폼임을 강조합니다.

4.1.1 VS Code: 코드베이스 내 다이어그램 관리

VS Code는 개발자들이 코드와 문서를 통합 관리하는 대표적인 IDE입니다. Excalidraw는 `excalidraw-vscode` 확장을 통해 VS Code 환경에서 다이어그램을 직접 생성하고, 코드베이스 내에서 관리할 수 있도록 지원합니다. 이로 인해 개발자는 시스템 설계, API 흐름, 데이터 모델링 등 복잡한 구조를 시각적으로 표현하며, 코드 리뷰와 협업 과정에서 다이어그램을 효과적으로 활용할 수 있습니다. Excalidraw의 파일 포맷과 Copilot 연동 기능은 VS Code 내에서의 시각적 협업을 한층 강화합니다.

지원 파일 형식과 확장성

excalidraw-vscode 확장은 개발자들이 코드베이스 내에서 다이어그램을 직접 생성, 편집, 관리할 수 있도록 설계되었습니다. 이 확장은 .excalidraw, .excalidraw.json, .excalidraw.svg, .excalidraw.png 등 다양한 파일 포맷을 지원합니다. .excalidraw와 .excalidraw.json 파일은 Excalidraw의 씬 데이터(scene data)를 JSON 구조로 저장하여, 다이어그램의 모든 객체 정보와 메타데이터를 보존할 수 있습니다. .excalidraw.svg와 .excalidraw.png 포맷은 시각적 결과물을 코드 리뷰나 문서화에 바로 활용할 수 있게 해줍니다.

소스 데이터 보존 메커니즘

Excalidraw의 파일 포맷은 다이어그램의 소스 데이터를 완전하게 보존하는 것이 특징입니다. 특히 .excalidraw.svg 파일은 SVG 벡터 이미지 내부에 base64로 인코딩된 씬 데이터를 포함하고 있어, 언제든지 Excalidraw에서 재편집이 가능합니다. 이는 단순 이미지 내보내기와 달리, 다이어그램의 버전 관리와 협업을 위한 강력한 기반을 제공합니다. Git을 통한 변경 이력 추적, PR 리뷰, 코드 내 문서화 등 개발자 워크플로에 최적화된 구조입니다.

Excalidraw Copilot과 자연어 다이어그램 생성

excalidraw-vscode는 GitHub Copilot과의 연동을 통해, @excalidraw 명령어를 활용한 자연어 다이어그램 생성 기능을 지원합니다. 사용자는 코드 내 주석 또는 명령어 입력으로 “Kubernetes 클러스터 구조를 그려줘”와 같은 프롬프트를 입력하면, Copilot이 LLM을 통해 자동으로 Excalidraw 다이어그램을 생성합니다. 이는 반복적인 시스템 설계, API 흐름, 데이터 모델링 등에서 개발자의 생산성을 획기적으로 높여주며, AI 기반 Prompt Drawing의 실질적 활용 사례로 주목받고 있습니다.

협업 및 버전 관리의 장점

VS Code 환경에서의 Excalidraw 통합은 다이어그램의 소스 데이터가 코드베이스와 동일한 버전 관리 체계에 편입된다는 점에서 큰 강점을 가집니다. 팀원 간 PR 리뷰, 코드 변경에 따른 다이어그램 업데이트, 문서와 코드의 동기화가 자연스럽게 이루어집니다. 이는 기존의 외부 다이어그램 도구와 비교할 때, 개발 생산성과 협업 효율성을 크게 개선하는 요소입니다.

4.1.2 Obsidian: 지식 관리와 시각적 사고의 결합

Obsidian은 마크다운 기반의 지식 관리 도구로, excalidraw-plugin(Zsolt Viczian 제작)을 통해 Excalidraw와의 강력한 통합을 제공합니다. 이 플러그인은 Obsidian Vault 내에 다이어그램

파일을 저장하고, 마크다운 문서 내에서 직접 편집 및 임베딩이 가능합니다. 사용자는 [[파일명.excalidraw]] 구문을 통해 다이어그램을 노트에 삽입할 수 있으며, 양방향 링크를 통해 도면과 노트 간의 네트워크를 구축할 수 있습니다.

Obsidian에서 Excalidraw를 활용하면 지식의 시각적 구조화가 가능해집니다. 예를 들어, 복잡한 시스템 아키텍처를 다이어그램으로 표현하고, 각 컴포넌트에 해당하는 노트와 연결함으로써 지식 그래프를 구축할 수 있습니다. 이는 단순 텍스트 기반 노트의 한계를 넘어, 시각적 사고와 정보 탐색을 동시에 지원하는 강력한 도구로 자리잡고 있습니다. 또한, 연구, 논문, 기술 문서 작성 등 다양한 분야에서 수식과 다이어그램을 결합하여 복합적인 정보를 효과적으로 전달할 수 있습니다.

Obsidian 플러그인 구조와 Vault 통합

Obsidian은 마크다운 기반 지식 관리 도구로, `excalidraw-plugin`(Zsolt Viczian 제작)을 통해 Excalidraw와의 강력한 통합을 제공합니다. 이 플러그인은 Obsidian Vault 내에 다이어그램 파일을 저장하고, 마크다운 문서 내에서 직접 편집 및 임베딩이 가능합니다. 사용자는 [[파일명.excalidraw]] 구문을 통해 다이어그램을 노트에 삽입할 수 있으며, 양방향 링크를 통해 도면과 노트 간의 네트워크를 구축할 수 있습니다.

마크다운 임베딩과 LaTeX 지원

`excalidraw-plugin`은 마크다운 문서 내에 다이어그램을 인라인 또는 블록 형태로 임베딩할 수 있습니다. 또한, LaTeX 수식 지원을 통해 수학적/과학적 다이어그램 작성이 가능하며, 복잡한 수식이나 알고리즘 시각화에 유용합니다. 이는 Obsidian을 활용한 연구, 논문, 기술 문서 작성 환경에서 Excalidraw의 활용도를 크게 확장합니다.

도면-노트 양방향 링크와 지식 그래프

Obsidian의 핵심 강점인 양방향 링크 기능은, `excalidraw-plugin`과 결합할 때 도면과 텍스트 노트 간의 유기적 연결을 실현합니다. 예를 들어, 시스템 아키텍처 다이어그램에서 각 컴포넌트 객체에 해당 노트로의 링크를 삽입할 수 있습니다. ExcaliBrain 플러그인은 이러한 노트 간 관계를 시각적 그래프로 표현하여, 지식 네트워크의 구조적 이해를 돕습니다.

스크립트 엔진과 자동화 기능

`excalidraw-plugin`은 스크립트 엔진을 내장하여, 반복적인 도면 생성, 자동화된 객체 배치, 데이터 기반 다이어그램 갱신 등 다양한 자동화 워크플로를 지원합니다. 예를 들어, 특정 마크다운 테이블을 파싱하여 ERD(Entity Relationship Diagram)를 자동 생성하거나, 일정 주기로 다이어그램을 업데이트하는 등 생산성 향상에 기여합니다.

Obsidian에서 Excalidraw를 활용하는 실제 사례로는 프로젝트 관리, 연구 노트, 기술 문서화 등이 있습니다. 예를 들어, 소프트웨어 개발 프로젝트에서 요구사항, 설계, 테스트 케이스를 각각의 노트로 관리하면서, 전체 시스템 구조를 Excalidraw 다이어그램으로 표현하고 각 노트와 연결할 수 있습니다. 또한, 복잡한 데이터 모델링이나 알고리즘 설명을 다이어그램과 수식으로 시각화하여 팀원들과 공유할 수 있습니다. Obsidian의 플러그인 생태계와 결합된 Excalidraw는 지식 관리와 시각적 사고를 동시에 실현하는 혁신적 도구로 평가받고 있습니다.

4.1.3 Confluence와 Notion: 엔터프라이즈 위키 통합

Confluence와 Notion은 엔터프라이즈 협업 및 문서화 플랫폼으로 널리 사용되고 있습니다. Excalidraw는 이 두 플랫폼과의 통합을 통해 조직 내 시각적 커뮤니케이션을 강화하며, 실시간 협업과 자산 관리 측면에서 다양한 기능을 제공합니다. Confluence에서는 공식 앱을 통한 완전한 통합이 가능하며, Notion에서는 임베딩 방식으로 간편하게 다이어그램을 활용할 수 있습니다. 각 플랫폼의 데이터 저장 구조와 협업 방식에 따라 Excalidraw의 활용 전략이 달라집니다.

Confluence용 Excalidraw Plus 앱의 기능

Confluence는 엔터프라이즈 위키의 대표적 솔루션으로, Excalidraw Plus 앱을 통해 Excalidraw 다이어그램을 네이티브로 통합할 수 있습니다. 이 앱은 220개 이상의 커뮤니티 아이콘 라이브러리, Mermaid 다이어그램 지원, PDF/Word 내보내기 시 다이어그램 자동 포함 등 엔터프라이즈 협업에 특화된 기능을 제공합니다. 실시간 협업, 버전 관리, 권한 설정 등 Confluence의 기본 기능과 자연스럽게 연동되어, 대규모 조직에서도 안정적으로 활용할 수 있습니다.

Notion 임베딩과 데이터 저장 구조

Notion에서는 /embed 명령을 통해 Excalidraw 다이어그램을 페이지에 삽입할 수 있습니다. 다만, 임베딩된 다이어그램의 데이터는 Notion 자체에 저장되지 않고, 외부 URL 또는 localStorage에 의존하는 구조입니다. 이로 인해 데이터 동기화, 백업, 버전 관리 측면에서 제한점이 존재합니다. 실시간 편집이나 협업에는 적합하지만, 장기적 자산 관리에는 별도의 백업 전략이 필요합니다.

엔터프라이즈 협업에서의 비교

Confluence와 Notion 모두 Excalidraw의 시각적 커뮤니케이션 역량을 엔터프라이즈 문서화 워크플로에 통합할 수 있는 장점을 가집니다. Confluence는 공식 앱을 통한 완전한 통합, Notion

은 임베딩 중심의 간편한 접근성을 제공합니다. 조직의 보안 정책, 데이터 주권, 협업 방식에 따라 적합한 플랫폼을 선택하는 것이 중요합니다.

실제 기업 환경에서는 Confluence를 활용하여 시스템 아키텍처, 서비스 플로우, 운영 매뉴얼 등을 시각적으로 문서화하고, 팀원 간 실시간 협업과 버전 관리가 이루어집니다. Notion에서는 프로젝트 관리, 회의록, 아이디어 브레인스토밍 등 다양한 문서에 Excalidraw 다이어그램을 삽입하여 시각적 정보를 공유할 수 있습니다. 두 플랫폼 모두 Excalidraw의 다이어그램을 통해 복잡한 정보를 쉽게 전달하고, 조직 내 지식의 시각적 자산화를 촉진하는 역할을 합니다.

4.2 커뮤니티 라이브러리와 템플릿 생태계

Excalidraw의 진정한 강점은 커뮤니티가 주도하는 방대한 아이콘 라이브러리와 시스템 설계 템플릿 생태계에 있습니다. 이 섹션에서는 주요 IT 인프라 아이콘 라이브러리의 활용법과, 다양한 도메인별 템플릿 및 마켓플레이스의 구조를 상세히 소개합니다. 이를 통해 사용자들은 복잡한 시스템 아키텍처, 클라우드 인프라, UX 와이어프레임 등 다양한 요구를 신속하게 충족할 수 있습니다.

4.2.1 IT 인프라 아이콘 라이브러리

Excalidraw는 IT 인프라 설계와 시스템 다이어그램 작성에 특화된 아이콘 라이브러리를 제공합니다. 공식 커뮤니티 포털인 libraries.excalidraw.com을 통해 AWS, Kubernetes, Docker 등 주요 벤더의 아이콘을 손쉽게 추가할 수 있으며, 개발자와 엔지니어는 복잡한 인프라 구조를 시각적으로 표현할 수 있습니다. 라이브러리 관리와 커스터마이징 기능은 최신 기술 트렌드에 신속히 대응할 수 있도록 지원합니다.

libraries.excalidraw.com의 원클릭 추가

Excalidraw는 공식 커뮤니티 라이브러리 포털인 libraries.excalidraw.com을 통해, 수백 종의 IT 인프라 아이콘을 원클릭으로 추가할 수 있습니다. 사용자는 원하는 라이브러리를 선택 후 “Add to Excalidraw” 버튼을 클릭하면, 즉시 자신의 캔버스에 해당 아이콘 세트가 추가됩니다. 이 방식은 별도의 파일 다운로드나 복잡한 설정 없이, 최신 아이콘을 실시간으로 활용할 수 있는 점이 큰 장점입니다.

주요 IT 라이브러리 소개

Excalidraw 커뮤니티에는 AWS Architecture Icons(5만+ 다운로드), Kubernetes,

Docker, Redis, PostgreSQL, VMware, Microsoft Fabric Shape Library 등 주요 IT 벤더와 오픈소스 생태계의 아이콘 팩이 다수 등록되어 있습니다. 각 라이브러리는 공식 벤더 가이드 라인을 준수하며, 실제 아키텍처 다이어그램 작성에 최적화된 벡터 그래픽으로 제공됩니다. 이를 통해 개발자와 DevOps 엔지니어는 복잡한 클라우드, 컨테이너, 데이터베이스 인프라를 신속하게 시각화할 수 있습니다.

라이브러리 관리와 커스터마이징

사용자는 라이브러리 내 개별 아이콘을 자유롭게 커스터마이징하거나, 자체 아이콘 세트를 만들어 공유할 수 있습니다. 커뮤니티에서 지속적으로 새로운 아이콘이 추가되고, 버전 관리가 이루어져 최신 기술 트렌드에 신속히 대응할 수 있습니다. 이는 Excalidraw가 단순한 그리기 도구를 넘어, 오픈소스 기반 IT 시각화 플랫폼으로 자리잡는 데 중요한 역할을 합니다.

실무에서 IT 인프라 아이콘 라이브러리를 활용하는 예로는 클라우드 아키텍처 설계, 네트워크 토폴로지, 데이터베이스 구조도 작성 등이 있습니다. 예를 들어, AWS 아이콘 라이브러리를 활용하면 EC2, S3, Lambda 등 다양한 컴포넌트를 표준 아이콘으로 표현할 수 있으며, Kubernetes 라이브러리는 Pod, Service, Deployment 등 컨테이너 오케스트레이션 구조를 시각화할 수 있습니다. 이러한 라이브러리는 조직 내 설계 문서의 일관성과 가독성을 높여주며, 기술적 커뮤니케이션의 효율성을 극대화합니다.

4.2.2 시스템 설계 템플릿과 마켓플레이스

시스템 설계 템플릿과 마켓플레이스는 Excalidraw 사용자가 복잡한 아키텍처, 서비스 플로우, UI/UX 디자인을 신속하게 시작하고, 검증된 패턴을 재사용할 수 있도록 지원하는 핵심 생태계입니다. marketplace.excalidraw.com에서는 다양한 도메인별 템플릿과 라이브러리를 검색하고 적용할 수 있으며, 커뮤니티 기반의 공유와 협업이 활발하게 이루어집니다. 이를 통해 조직 내외부에서 실무에 바로 활용 가능한 설계 자산을 구축할 수 있습니다.

marketplace.excalidraw.com의 구조와 자산

Excalidraw 마켓플레이스(marketplace.excalidraw.com)는 라이브러리, 시스템 설계 템플릿, UI 컴포넌트, 디자인 자산 등 다양한 리소스를 한 곳에서 제공하는 플랫폼입니다. 사용자는 원하는 템플릿이나 라이브러리를 검색 후, 즉시 자신의 작업 공간에 적용할 수 있습니다. 이는 복잡한 시스템 설계, 서비스 플로우, UX 와이어프레임, 데이터 모델링 등 다양한 요구에 빠르게

대응할 수 있게 해줍니다.

주요 시스템 설계 템플릿

system-design-templates-excalidraw, C4 모델 라이브러리, UX 와이어프레이밍 라이브러리, Wardley Maps 등 도메인별로 특화된 템플릿이 다수 제공됩니다. C4 모델 라이브러리는 소프트웨어 시스템의 컨텍스트, 컨테이너, 컴포넌트, 코드 레벨 다이어그램을 표준화된 방식으로 작성할 수 있게 해주며, Wardley Maps는 전략적 IT 의사결정에 활용됩니다.

커뮤니티 기반 템플릿 공유와 협업

Excalidraw 마켓플레이스는 누구나 템플릿을 업로드하고, 다른 사용자와 공유할 수 있는 오픈 생태계입니다. 이를 통해 조직 내/외부에서 검증된 설계 패턴, 아키텍처 샘플, UI/UX 디자인 자산을 손쉽게 재사용할 수 있습니다. 커뮤니티의 활발한 기여는 Excalidraw의 확장성과 실무 활용도를 지속적으로 높이고 있습니다.

실제 활용 사례로는 소프트웨어 개발팀이 C4 모델 템플릿을 활용하여 시스템 구조를 표준화하고, UX 디자이너가 와이어프레임 템플릿을 통해 빠르게 프로토타입을 제작하는 경우가 있습니다. 또한, Wardley Maps 템플릿을 활용하여 전략적 의사결정 과정을 시각화하고, 조직 내 다양한 이해관계자와 공유할 수 있습니다. 마켓플레이스의 오픈 생태계는 다양한 산업 분야에서 Excalidraw의 실무 활용성을 높이고, 혁신적인 설계 패턴의 확산을 촉진합니다.

4.3 내보내기과 산출물 활용

Excalidraw는 단순한 화면 내 작업에 그치지 않고, 다양한 포맷으로의 내보내기와 외부 시스템과의 연동을 지원합니다. 본 섹션에서는 PNG/SVG 내보내기와 썸 데이터 임베딩, PowerPoint 연동 우회 전략, NPM 패키지 임베딩 등 실무에서 요구되는 산출물 활용 방법을 심층적으로 다룹니다.

4.3.1 PNG/SVG 내보내기와 썸 데이터 임베딩

Excalidraw는 다이어그램을 PNG와 SVG 등 다양한 이미지 포맷으로 내보낼 수 있으며, 내보낸 이미지에 썸 데이터를 임베딩하여 언제든지 원본 복구와 재편집이 가능합니다. `exportToBlob()`와 `exportToSvg()` API를 활용하면 배경, 해상도, 다크모드 등 다양한 옵션을 조정할 수 있고, 임베딩된 썸 데이터는 장기적 자산 관리와 협업에 큰 가치를 제공합니다. 이러한 기능은 단순 이미지 내보내기를 넘어, 실무에서의 재사용성과 데이터 복구를 보장합니다.

exportToBlob()와 exportToSvg() API

Excalidraw는 exportToBlob() API를 통해 PNG 포맷, exportToSvg() API를 통해 SVG 포맷으로 다이어그램을 내보낼 수 있습니다. 각 API는 배경 포함/제외, 스케일(해상도), 다크모드 등 다양한 옵션을 지원하여, 프레젠테이션, 문서화, 웹 게시 등 다양한 활용 요구에 맞게 최적화된 결과물을 생성합니다. 예를 들어, exportToBlob({ exportBackground: false, scale: 2 })와 같이 옵션을 조합할 수 있습니다.

PNG/SVG 내 씬 데이터 임베딩 구조

Excalidraw의 PNG 내보내기는 tEXt 청크 영역에 씬 데이터를 base64로 인코딩하여 임베딩합니다. SVG 내보내기는 “desc” 태그 또는 문자열 내에 씬 데이터를 압축하여 포함합니다. 이 구조는 단순 이미지가 아니라, 언제든지 Excalidraw에 다시 불러와서 재편집 가능한 “리버서블 아트팩트”를 제공합니다. 이는 장기적 자산 관리, 협업, 버전 관리에 매우 유용합니다.

재편집 가능성과 데이터 복구

임베딩된 씬 데이터 덕분에, PNG/SVG 파일만 있으면 원본 다이어그램을 언제든지 복구할 수 있습니다. 사용자는 Excalidraw의 “Import Scene” 기능을 통해 파일을 업로드하면, 모든 객체와 메타데이터가 완벽하게 복원됩니다. 이는 기존의 비가역적 이미지 내보내기와 차별화되는 Excalidraw만의 강력한 기능입니다.

실무에서는 프로젝트 산출물, 기술 문서, 프레젠테이션 자료 등 다양한 곳에서 Excalidraw의 내보내기 기능이 활용됩니다. 예를 들어, 시스템 설계 다이어그램을 PNG로 내보내어 보고서에 삽입하거나, SVG로 내보내어 웹 페이지에 벡터 그래픽으로 활용할 수 있습니다. 임베딩된 씬 데이터 덕분에, 이미지 파일만 보관해도 언제든지 원본 다이어그램을 복구하고 재편집할 수 있어, 장기적 자산 관리와 협업에 매우 유리합니다. 또한, 버전 관리 시스템과 연동하여 다이어그램의 변경 이력을 추적할 수 있으며, 팀원 간 실시간 협업과 피드백이 용이해집니다.

4.3.2 PowerPoint 연동 우회 전략과 NPM 임베딩

PowerPoint와의 연동은 Excalidraw 사용자가 프레젠테이션 자료를 제작할 때 자주 요구되는 기능입니다. 현재 직접 내보내기는 지원되지 않지만, PNG/SVG 내보내기와 EMF 변환을 통해 다양한 방식으로 PowerPoint에 다이어그램을 삽입할 수 있습니다. 또한, React 기반 웹 애플리케이션에서는 @excalidraw/excalidraw NPM 패키지를 활용하여 화이트보드 기능을 직접 임베

당할 수 있으며, 커스텀 테마와 협업 기능 확장도 가능합니다. 이러한 내보내기 및 임베딩 전략은 조직 내 산출물의 실무 활용성을 크게 높여줍니다.

PowerPoint 연동의 한계와 우회 전략

Excalidraw는 현재 PowerPoint로의 직접 내보내기를 지원하지 않습니다. 이를 해결하기 위한 우회 전략으로는 (1) PNG 내보내기 후 PowerPoint에 이미지로 삽입, (2) SVG 내보내기 후 EMF(Enhanced Metafile)로 변환하여 PowerPoint에 벡터 객체로 삽입하는 방법이 있습니다. EMF 변환은 Inkscape, CloudConvert 등 도구를 활용할 수 있으며, 삽입 후 개별 객체 편집이 가능합니다.

@excalidraw/excalidraw NPM 패키지 임베딩

React 기반 웹 애플리케이션에서는 @excalidraw/excalidraw NPM 패키지를 활용해 화이트보드 기능을 직접 임베딩할 수 있습니다. 이 패키지는 ESM(ECMAScript Module) 전용이며, React 18+ 및 피어 의존성(react, react-dom, @excalidraw/excalidraw-next 등)이 필요합니다. 컨테이너 환경에서는 Node.js 18+, ESM 호환 번들러, 정적 파일 서빙 구성이 요구됩니다. 임베딩 시 커스텀 테마, 플러그인, 협업 기능 확장 등 다양한 커스터마이징이 가능합니다.

산출물의 실무 활용성

이러한 내보내기 및 임베딩 전략은 조직 내 프레젠테이션, 기술 문서화, 사내 포털, 교육 자료 등 다양한 산출물에 Excalidraw 다이어그램을 효과적으로 통합할 수 있게 해줍니다. 특히, PNG/SVG+썸 데이터 구조와 NPM 임베딩은 장기적 자산 관리와 재사용성 측면에서 큰 가치를 제공합니다.

실제 활용 사례로는 개발팀이 시스템 설계 다이어그램을 PNG로 내보내어 PowerPoint에 삽입하거나, SVG를 EMF로 변환하여 벡터 그래픽으로 활용하는 경우가 있습니다. React 기반 사내 포털에서는 @excalidraw/excalidraw 패키지를 임베딩하여 실시간 협업 화이트보드 기능을 제공하고, 교육 자료 제작 시 커스텀 테마와 플러그인을 활용하여 조직의 요구에 맞는 다이어그램을 제작할 수 있습니다. 이러한 전략은 Excalidraw의 산출물을 다양한 업무 환경에 효과적으로 통합하고, 장기적 자산 관리와 재사용성을 극대화하는 데 큰 도움이 됩니다.

4.4 핵심 기술 아키텍처 이해

Excalidraw의 뛰어난 반응성과 확장성은 독창적인 이중 캔버스 아키텍처와 다양한 성능 최적화 기법에 기반합니다. 이 섹션에서는 Static Canvas와 Interactive Canvas의 분리 구조, React+TypeScript+HTML5 Canvas 기술 스택, 그리고 Viewport Culling, Memoization, Render Throttling 등 핵심 최적화 전략을 심층적으로 해설합니다.

4.4.1 이중 캔버스 아키텍처와 성능 최적화

Excalidraw는 대형 다이어그램에서도 뛰어난 반응성과 확장성을 제공하기 위해 이중 캔버스 아키텍처를 채택하고 있습니다. Static Canvas와 Interactive Canvas의 분리 구조는 UI 반응성을 극대화하며, React+TypeScript+HTML5 Canvas 기술 스택은 대용량 객체 처리에 최적화된 성능을 보장합니다. Viewport Culling, Memoization, Render Throttling 등 다양한 최적화 기법은 실시간 협업과 대용량 다이어그램 편집에서도 일관된 성능을 제공합니다. 이러한 아키텍처는 미래 지향적 기능 확장에도 유연하게 대응할 수 있는 기반을 마련합니다.

Static Canvas와 Interactive Canvas 구조

Excalidraw는 Static Canvas와 Interactive Canvas의 이중 구조를 채택하고 있습니다. Static Canvas는 RoughJS 라이브러리를 활용하여 모든 드로잉 요소(선, 도형, 텍스트 등)를 빠르게 렌더링합니다. 반면, Interactive Canvas는 UI 오버레이(선택, 드래그, 핸들, 도구바 등)를 별도 레이어로 독립 렌더링합니다. 이 분리 구조는 대형 다이어그램에서도 UI 반응성을 유지하는 핵심 요소입니다.

React + TypeScript + HTML5 Canvas 스택

Excalidraw의 프론트엔드는 React와 TypeScript로 작성되어, 컴포넌트 기반의 유지보수성과 타입 안전성을 확보합니다. 렌더링은 HTML5 Canvas API를 직접 활용하여, SVG 기반 도구 대비 수천~수만 객체의 대용량 다이어그램도 부드럽게 처리할 수 있습니다. React의 상태 관리와 Canvas의 저수준 렌더링 성능이 결합된 구조입니다.

Viewport Culling(뷰포트 내 요소만 렌더링)

Excalidraw는 Viewport Culling 기법을 적용하여, 현재 뷰포트(화면 영역) 내에 위치한 객체만 선별적으로 렌더링합니다. 이를 통해 수천~수만 개의 객체가 있는 대형 다이어그램에서도

불필요한 연산을 최소화하고, 렌더링 속도를 극대화합니다. 이는 클라우드 인프라, MSA, 데이터 베이스 ERD 등 복잡한 구조의 실무 다이어그램 작성에 필수적인 최적화입니다.

Memoization과 Render Throttling

sceneNonce 캐시 무효화(Memoization) 기법을 통해, 동일한 씬 데이터에 대해 불필요한 재렌더링을 방지합니다. 또한, renderStaticSceneThrottled 함수는 렌더링 빈도를 제한(Throttling)하여, 빠른 연속 입력 시에도 UI 프레임 드랍을 최소화합니다. 이 두 가지 기법의 결합으로, 실시간 협업 및 대용량 다이어그램 편집에서도 일관된 성능을 제공합니다.

확장성과 미래 지향적 구조

이중 캔버스 아키텍처와 다양한 최적화 기법은, 향후 AI 기반 자동 레이아웃, 실시간 협업 확장, 3D/AR 다이어그램 등 미래 기능 추가에도 유연하게 대응할 수 있는 기반을 제공합니다. 오픈소스 커뮤니티의 활발한 기여와 함께, Excalidraw는 지속적으로 진화하는 시각적 협업 플랫폼으로 자리매김하고 있습니다.

실제 기술적 비교를 통해 Excalidraw의 아키텍처가 갖는 장점을 살펴보면, SVG 기반 다이어그램 도구는 객체 수가 많아질수록 렌더링 성능이 급격히 저하되는 반면, HTML5 Canvas는 GPU 가속과 저수준 API 활용으로 대용량 데이터 처리에 강점을 보입니다. 또한, React와 TypeScript의 결합은 코드의 유지보수성과 확장성을 높여주며, 커스텀 플러그인 및 협업 기능 개발에 유리합니다. Viewport Culling과 Memoization은 실시간 협업 환경에서 불필요한 연산을 줄여주고, Render Throttling은 사용자 입력에 따른 UI 프레임 드랍을 최소화하여 쾌적한 사용 경험을 제공합니다. 이러한 아키텍처와 최적화 전략은 Excalidraw가 미래 지향적 시각적 협업 플랫폼으로 성장하는데 핵심적인 역할을 하고 있습니다.

5장: 기업 도입 가이드 — 평가에서 운영까지

5.1 PoC에서 표준화까지: 4단계 도입 로드맵

Excalidraw를 조직 내 표준 그리기 도구로 도입하기 위해서는 체계적인 단계별 접근이 필수적입니다. 단순히 무료 도구를 시도하는 수준을 넘어, 조직의 규모와 보안 요구사항, 협업 방식에 따라 도입 전략을 차별화해야 합니다. 본 절에서는 PoC(개념 검증)에서 시작해 파일럿, 확산, 그리고 표준화에 이르는 4단계 도입 경로를 구체적으로 설명하며, 각 단계에서 적합한 배포 방식과 활용

시나리오를 제시합니다. 이를 통해 기업은 최소한의 리스크로 Excalidraw의 도입 효과를 극대화할 수 있습니다.

5.1.1 4단계 도입 경로와 조직 유형별 선택

Excalidraw의 도입은 단순한 도구 사용을 넘어 조직의 협업 문화와 IT 인프라에 깊이 연계되어야 성공적으로 정착할 수 있습니다. 각 단계별로 요구되는 기술적 조건과 조직의 특성에 따라 최적의 도입 경로가 달라지므로, 실무에서는 단계별로 세부적인 평가와 의사결정이 이루어집니다. 이 절에서는 PoC, 파일럿, 확산, 표준화의 4단계 도입 경로를 조직 유형별로 구체적으로 안내합니다.

PoC 단계: excalidraw.com 무료 버전 활용

PoC(Proof of Concept) 단계에서는 최소 2~3명의 핵심 인력이 excalidraw.com의 무료 버전을 사용해 실제 다이어그램 작성 및 협업 경험을 검증합니다. 이 단계에서는 별도의 인프라 구축이나 비용 부담 없이, 웹 브라우저만으로 즉시 사용이 가능하므로, 초기 도입 장벽이 매우 낮습니다. 주로 소규모 스타트업, 프로젝트 초기 단계, 혹은 신기술 도입에 신중한 조직에서 적합합니다. 이 과정에서 실시간 협업, 내보내기 기능, 기본 라이브러리 활용성을 집중적으로 평가합니다. 실제로 PoC 단계에서는 기존에 사용하던 다이어그램 도구와 비교하여 Excalidraw의 직관적인 UI와 빠른 협업 기능이 얼마나 업무 효율을 높이는지 확인하는 것이 중요합니다. 또한, 단기간에 도구의 장단점을 파악하고, 조직 내에서 확산 가능성을 검토하는 기초 자료를 수집하게 됩니다.

파일럿 단계: Docker 셀프호스팅 또는 Excalidraw+ 체험

파일럿 단계에서는 Docker 기반의 셀프호스팅 배포나 Excalidraw+의 14일 무료 체험을 통해, 보안 및 커스터마이징 요구가 높은 조직 환경에서의 적합성을 검증합니다. 셀프호스팅은 사내 네트워크 내에서 데이터 주권을 확보할 수 있으며, LDAP/SSO 연동 등 추가 보안 구성이 가능합니다. Excalidraw+는 SaaS 방식으로, SOC 2 인증, E2E 암호화, 음성/화면 공유 등 엔터프라이즈 기능을 단기간 내 경험할 수 있습니다. 중견기업, 보안 민감 조직, 혹은 사내 시스템 통합을 고려하는 팀에 적합합니다. 파일럿 단계에서는 실제 업무 환경에서 도구의 성능, 보안, 협업 품질을 집중적으로 테스트하며, 조직의 IT 정책과 연동 가능성, 커스터마이징 범위 등을 실무적으로 검토합니다. 또한, 파일럿을 통해 도입 후 예상되는 운영 이슈와 개선점을 미리 파악할 수 있습니다.

확산 단계: excalidraw-collaboration 또는 ExcaliDash 기반 협업

확산 단계에서는 excalidraw-collaboration 서버 또는 ExcaliDash(Docker Compose 기반

협업 환경)를 통해, 다수의 사용자가 동시에 협업할 수 있는 환경을 구축합니다. 이 단계에서는 WebSocket 기반 실시간 동기화, 중앙화된 스토리지, 사용자 그룹 관리 등 조직 내 대규모 사용을 위한 인프라가 요구됩니다. DevOps, 개발, 기획, 디자인 등 다양한 부서가 동시에 활용하며, 협업 품질과 데이터 관리 효율성을 중점적으로 평가합니다. 확산 단계에서는 실제로 여러 부서가 동시에 다이어그램을 작성하고 공유하는 과정에서 발생하는 협업 충돌, 데이터 동기화, 접근 권한 관리 등의 이슈를 해결해야 하며, 조직 내에서 Excalidraw가 표준 협업 도구로 자리잡을 수 있도록 운영 정책을 수립하는 것이 중요합니다.

표준화 단계: NPM 패키지 임베드 및 MCP 연동

표준화 단계에서는 @excalidraw/excalidraw NPM 패키지를 사내 시스템(예: 사내 포털, 위키, 개발자 도구)에 직접 임베드하고, MCP(Model Context Protocol) 연동을 통해 AI 기반 워크플로우(예: 자연어 프롬프트로 다이어그램 생성)를 통합합니다. 이 단계는 대기업, 글로벌 조직, 혹은 DevOps/AI 자동화가 중요한 조직에서 주로 적용되며, 완전한 커스터마이징, 사내 인증 체계 연동, 대용량 협업 환경 구축이 가능합니다. 표준화 단계에서는 조직의 핵심 시스템과 Excalidraw를 완전히 통합하여, 다이어그램 작성과 관리가 업무 프로세스의 일부로 자연스럽게 녹아들도록 설계합니다. 또한, AI와 연계한 자동화 기능, 대용량 데이터 처리, 글로벌 협업 환경 등 고도화된 운영 시나리오를 실현할 수 있습니다.

5.1.2 1주 평가 체크리스트와 의사결정 기준

Excalidraw 도입을 위한 1주 평가 체크리스트는 실제 현장에서 도구의 기능과 조직 적합성을 빠르게 검증할 수 있도록 설계되었습니다. 각 단계별로 구체적인 테스트 항목과 의사결정 기준을 마련함으로써, 도입 리스크를 최소화하고 최적의 도입 경로를 선택할 수 있습니다. 이 절에서는 일주일 단위로 진행되는 평가 절차와 주요 체크포인트를 상세히 안내합니다.

Day 1: 계정 설정 및 초기 환경 점검

도입 첫날에는 excalidraw.com 또는 셀프호스팅 인스턴스에 계정을 생성하고, 기본적인 다이어그램 작성 및 저장, 공유 기능을 점검합니다. 기존에 사용하던 다이어그램(Visio, draw.io, Lucidchart 등) 임포트 시도 및 호환성 확인이 중요합니다. 또한, 팀원 간의 실시간 협업이 원활하게 동작하는지, 파일 내보내기(PNG, SVG 등) 옵션이 충분한지 체크합니다. 실제로 초기 환경 점검에서는 브라우저 호환성, 네트워크 연결 상태, 데이터 저장 위치 등 기술적 요소를 꼼꼼히

확인해야 하며, 조직의 IT 정책과 맞는지 검토하는 과정이 필요합니다. 초기 사용 경험을 통해 도구의 직관성과 접근성을 평가할 수 있습니다.

Day 2~3: 협업 품질 및 워크플로우 테스트

이틀째와 셋째 날에는 2~3명의 팀원이 동시에 동일한 다이어그램을 편집하며, 실시간 동기화, 충돌 처리, 댓글/피드백 기능의 사용성을 집중적으로 평가합니다. 또한, 다이어그램을 다양한 포맷으로 내보내고, Notion, Confluence 등 사내 위키에 임베딩하는 과정을 테스트합니다. 이 과정에서 실제 업무 워크플로우에 Excalidraw가 자연스럽게 통합되는지, 기존 도구 대비 생산성에 변화가 있는지 확인합니다. 협업 품질 테스트에서는 실시간 편집 시 충돌 발생 빈도, 피드백 반영 속도, 다이어그램 공유와 임베딩의 편리성 등을 실무적으로 검증하며, 팀 내 커뮤니케이션 방식과 도구의 적합성을 평가합니다.

Day 4~5: 커스터마이징 및 보안 요구 검증

넷째, 다섯째 날에는 조직의 보안 정책(데이터 저장 위치, 인증 방식, 접근 제어 등)에 대한 적합성을 검토합니다. 셀프호스팅 환경에서는 LDAP/SSO 연동, 데이터 암호화, 감사 로그 기능을 점검하고, SaaS 환경에서는 SOC 2 인증, E2E 암호화, 데이터 주권 관련 정책을 확인합니다. 필요 시 커스텀 라이브러리 추가, 사내 아이콘 팩 적용, 워크플로우 자동화(MCP 연동 등)도 시도합니다. 이 단계에서는 조직의 보안 담당자와 협업하여 실제 데이터 흐름, 인증 체계, 접근 권한 설정이 조직 정책과 일치하는지 검증하며, 커스터마이징 옵션의 범위와 실무 적용성을 평가합니다.

Day 6~7: 최종 평가 및 팀 피드백 수집

마지막 이틀 동안에는 일주일간의 사용 경험을 바탕으로, 팀원별 피드백을 수집하고, 도입 전후 생산성 변화, 협업 품질, 내보내기 포맷 다양성, 기존 도구 대비 장단점 등을 종합적으로 평가합니다. 최종적으로 조직의 표준 도구로 채택할지, 파일럿을 연장할지, 혹은 도입을 중단할지에 대한 의사결정을 내립니다. 최종 평가에서는 팀원들의 실제 업무 경험을 바탕으로 도구의 장단점, 도입 후 예상되는 운영 이슈, 추가 개선 요구사항 등을 종합적으로 분석하여, 경영진이나 IT 담당자에게 객관적인 의사결정 자료를 제공합니다.

5.2 셀프호스팅 vs SaaS 의사결정

Excalidraw의 도입 방식은 크게 셀프호스팅과 SaaS(Excalidraw+)로 나뉩니다. 각 방식은 비용, 기능, 보안, 커스터마이징 측면에서 뚜렷한 차이를 보이므로, 조직의 IT 전략과 운영 요구에 따라

신중한 선택이 필요합니다. 본 절에서는 두 방식의 장단점을 비교 매트릭스로 정리하고, 실제 셀프호스팅 인프라 구성 방법을 단계별로 안내합니다.

5.2.1 비용/기능/보안 관점 비교 매트릭스

Excalidraw 도입 시 셀프호스팅과 SaaS 방식의 선택은 조직의 예산, 보안 정책, 기능 요구사항, 운영 인력의 역량에 따라 달라집니다. 이 절에서는 각 방식의 비용 구조, 기능 제공 범위, 보안 및 데이터 주권 측면에서 상세히 비교하여, 실무에서 의사결정에 도움이 되는 정보를 제공합니다.

비용 구조의 차이와 예산 계획

셀프호스팅은 오픈소스 무료 배포판을 기반으로 하며, 인프라 비용(서버, 스토리지 등)만 부담하면 됩니다. 별도의 라이선스 비용이 없고, 사용자 수에 따른 추가 비용도 발생하지 않습니다. 반면, SaaS인 Excalidraw+는 사용자당 월 \$6~7의 구독 비용이 발생하며, 엔터프라이즈 플랜은 별도 견적이 필요합니다. 중소기업 팀이나 예산이 제한된 조직은 셀프호스팅이 유리하지만, 대규모 조직은 SaaS의 확장성과 관리 편의성을 고려해야 합니다. 실제로 셀프호스팅은 초기 구축 비용이 저렴하지만, 서버 유지보수, 보안 패치, 장애 대응 등 추가 운영 비용이 발생할 수 있습니다. SaaS는 관리 부담이 적고, 자동 업데이트 및 기술 지원이 제공되어 운영 효율성이 높습니다.

기능 제공 범위와 협업 환경

SaaS Excalidraw+는 기본적으로 실시간 협업, E2E 암호화, 음성/화면 공유, 프레젠테이션 모드, PDF/Word 내보내기 등 엔터프라이즈 기능을 제공합니다. SOC 2 인증을 완료하여 보안 요구가 높은 조직에도 적합합니다. 셀프호스팅은 기본 배포만으로는 실시간 협업이 제한적이며, 별도의 WebSocket 서버 및 스토리지 구성이 필요합니다. 다만, 완전한 커스터마이징과 사내 시스템 통합이 가능하다는 장점이 있습니다. 실제로 셀프호스팅은 조직의 내부 시스템과 연동하여 맞춤형 기능을 추가하거나, 사내 인증 체계와 통합할 수 있으므로, 복잡한 워크플로우를 지원하는 데 유리합니다. SaaS는 표준화된 기능을 빠르게 사용할 수 있지만, 커스터마이징 범위가 제한적입니다.

보안 및 데이터 주권

셀프호스팅은 데이터가 전적으로 조직 내부에 저장되므로, 데이터 주권과 보안 통제가 극대화됩니다. LDAP, SSO, RBAC 등 조직의 인증 체계와 연동이 자유롭고, 커스텀 감사 로그 및 접근 제어도 가능합니다. SaaS는 클라우드 기반이므로, 데이터가 외부에 저장되지만, SOC 2 인증,

E2E 암호화, 정기적 보안 감사 등으로 보안 수준을 보장합니다. 규제 산업이나 민감 데이터 취급 조직은 셀프호스팅을, 빠른 도입과 유지보수가 중요한 조직은 SaaS를 선호합니다. 실제로 금융, 의료, 공공기관 등은 데이터 주권과 보안 정책을 중시하여 셀프호스팅을 선택하는 경우가 많으며, 스타트업이나 IT 서비스 기업은 SaaS의 빠른 도입과 확장성을 선호합니다.

5.2.2 셀프호스팅 인프라 구성 가이드

Excalidraw 셀프호스팅 환경을 구축하기 위해서는 기본 배포, 협업 서버 구성, 스토리지 백엔드, 네트워크 보안, 배포 자동화 등 다양한 기술적 요소를 고려해야 합니다. 이 절에서는 실제 인프라 구성 방법과 운영 시 주의사항을 단계별로 안내합니다.

기본 배포 환경과 요구 리소스

Excalidraw의 셀프호스팅은 Docker 컨테이너 한 대로 시작할 수 있습니다. 공식 이미지는 약 10MB로 매우 경량이며, 최소 1vCPU와 1GB RAM이면 소규모 파일럿 환경을 충분히 운영할 수 있습니다. 단일 서버로 시작해, 필요에 따라 스케일아웃 및고가용성 구성이 가능합니다. 실제로 초기 배포 시에는 테스트 환경에서 컨테이너를 실행하여 기능 검증을 진행하고, 이후 운영 환경에 맞춰 리소스 확장,고가용성 설계, 장애 대응 체계를 마련해야 합니다.

협업 서버 및 스토리지 백엔드 구성

실시간 협업을 위해서는 WebSocket 기반의 `excalidraw-room` 서버와 Redis가 필요합니다. 다수의 사용자가 동시에 접속하는 환경에서는 Redis를 세션 관리 및 메시지 브로커로 활용하며, MongoDB, Redis, S3 등 다양한 스토리지 백엔드를 선택할 수 있습니다. 데이터 백업, 장애 복구, 확장성 요구에 따라 스토리지 구성을 유연하게 조정할 수 있습니다. 실제로 협업 서버 구성 시에는 `excalidraw-room`의 URL 설정, Redis 클러스터링, 스토리지 백엔드의 데이터 동기화 및 백업 정책을 세부적으로 설계해야 하며, 운영 중 장애 발생 시 신속한 복구가 가능하도록 준비해야 합니다.

네트워크 보안 및 로드밸런서

HTTPS 인증서 적용, 사내 VPN 연동, 방화벽 설정 등 네트워크 보안이 필수입니다. 대규모 조직에서는 로드밸런서를 통해 트래픽을 분산하고, 장애 발생 시 자동 페일오버가 가능하도록 설계해야 합니다. 인증 체계는 LDAP, SSO, OAuth 등 조직 표준에 맞춰 연동합니다. 실제로 네트워크 보안 설계 시에는 SSL 인증서 관리, 내부망과 외부망 분리, 접근 제어 정책 수립, 로드밸런서 구성

등을 꼼꼼히 검토해야 하며, 보안 담당자와 협업하여 조직의 정책에 맞는 인프라를 구축해야 합니다.

ExcaliDash 및 배포 자동화

ExcaliDash는 Docker Compose 기반의 원클릭 배포 도구로, excalidraw, excalidraw-room, Redis, 스토리지 백엔드 등을 한 번에 배포할 수 있습니다. DevOps 담당자 1명이 반나절에서 2일 이내에 전체 환경을 구축할 수 있으며, Helm Chart, Ansible 등 IaC(Infrastructure as Code) 도구를 활용한 자동화도 지원됩니다. 운영 중에는 모니터링(Prometheus, Grafana), 백업, 장애 대응 체계를 함께 마련해야 합니다. 실제로 ExcaliDash를 활용하면 복잡한 인프라 구성 과정을 자동화할 수 있으며, 운영 효율성을 높이고 장애 대응 시간을 단축할 수 있습니다. 모니터링 시스템을 통해 실시간 상태를 점검하고, 정기적인 백업과 복구 테스트를 실시하는 것이 안정적인 운영의 핵심입니다.

5.3 마이그레이션 전략과 운영 주의사항

Excalidraw 도입 시 기존 다이어그램 자산의 이전, 대용량 운영 환경에서의 성능 한계, 그리고 도입 과정에서 발생하는 실수 패턴에 대한 이해가 필요합니다. 이 절에서는 주요 마이그레이션 경로와 실무 전략, 성능 한계와 우회 방안, 그리고 도입 시 흔히 발생하는 실수를 예방하는 가이드를 제공합니다.

5.3.1 기존 도구에서의 전환 경로

Excalidraw를 기존 다이어그램 도구에서 전환할 때는 파일 포맷 호환성, 변환 도구의 지원 범위, 그리고 실무적 재작성 전략을 고려해야 합니다. 공식적으로 직접 임포트 기능이 제한되어 있으므로, 주요 다이어그램을 새로 그리거나 변환 도구를 활용하는 하이브리드 전략이 필요합니다. 이 절에서는 실무에서 활용되는 주요 전환 경로와 사례를 안내합니다.

직접 임포트 미지원 현실과 대안

Excalidraw는 Visio, draw.io, Lucidchart 등 기존 다이어그램 도구의 파일을 직접 임포트하는 기능을 공식적으로 지원하지 않습니다. 이는 파일 포맷의 복잡성과, Excalidraw의 손그림 중심 구조 차이 때문입니다. 기존 자산을 1:1로 변환하는 것보다는, 주요 다이어그램을 새로 그리는 것이 오히려 효율적이라는 실무 전략이 널리 채택되고 있습니다. 실제로 Lucidchart에서 Excalidraw로 전환한 후, 생산성이 2배 이상 향상되었다는 사례도 보고되고 있습니다. 실무에서는 기존 다이어그

램의 핵심 구조만 추출하여 Excalidraw에서 재작성하고, 필요한 경우 SVG 등 중간 포맷을 활용해 일부 요소를 임포트하는 방식이 효과적입니다. 또한, 조직 내에서 전환 교육을 실시하여 새로운 도구의 사용법을 빠르게 익힐 수 있도록 지원하는 것이 중요합니다.

Mermaid 및 Gliffy 변환 경로

Mermaid 다이어그램은 mermaid-to-excalidraw.vercel.app(<https://mermaid-to-excalidraw.vercel.app>)와 같은 변환 도구를 통해 Excalidraw 포맷으로 일부 변환이 가능합니다. Gliffy 포맷의 경우, `excalidraw-converter`(CLI 도구)를 활용해 변환할 수 있습니다. 다만, 복잡한 다이어그램이나 표준 표기법(UML, BPMN 등)은 변환 과정에서 일부 정보 손실이 발생할 수 있으므로, 주요 구조만 변환 후 수동 보정이 필요합니다. 실제로 Mermaid 변환 도구는 플로차트 유형에 최적화되어 있으며, 복잡한 구조나 커스텀 스타일은 변환 후 추가 편집이 필요합니다. Gliffy 변환의 경우에도 표준 표기법을 완벽히 지원하지 않으므로, 변환 후 핵심 요소를 수동으로 보정하는 실무 전략이 필요합니다.

상호변환 및 하이브리드 워크플로

`draw.io`, Mermaid, Excalidraw 간에는 다양한 상호변환 도구가 존재합니다. 예를 들어, `draw.io`에서 SVG로 내보낸 후, Excalidraw에서 SVG를 임포트하거나, Mermaid를 Excalidraw로 변환해 시각적으로 보정하는 하이브리드 워크플로가 가능합니다. 이 과정에서 핵심 정보만 추출해 재구성하는 것이 실무적으로 가장 효율적입니다. 실제로 하이브리드 워크플로를 활용하면 기존 자산의 주요 구조를 빠르게 이전할 수 있으며, 조직 내에서 새로운 도구의 도입 장벽을 낮출 수 있습니다. 또한, 변환 과정에서 발생하는 정보 손실을 최소화하기 위해, 주요 다이어그램을 우선적으로 재작성하고, 필요 시 수동 보정 및 검증을 실시하는 것이 중요합니다.

5.3.2 운영 시 성능 한계와 우회 방안

Excalidraw 운영 환경에서는 대용량 다이어그램, 변환 도구의 지원 범위, 협업 서버의 인프라 한계 등 다양한 성능 이슈가 발생할 수 있습니다. 이 절에서는 주요 성능 한계와 실무적 우회 방안을 구체적으로 안내합니다.

대용량 다이어그램의 성능 한계

Excalidraw는 5,000~8,000개 이상의 객체를 포함하는 대형 다이어그램에서 성능 저하가 급격히 발생합니다. 10MB 이상의 파일은 저장 및 불러오기에서 제한이 있으며, 브라우저 메모리

한계로 인해 렌더링이 느려질 수 있습니다. 특히, 복잡한 시스템 아키텍처나 대규모 플로우차트 작성 시 주의가 필요합니다. 실제로 대용량 다이어그램 작성 시에는 객체 수를 관리하고, 논리적 단위별로 다이어그램을 분할하는 것이 성능 유지에 효과적입니다. 또한, 브라우저 메모리 사용량을 모니터링하여, 필요 시 다이어그램을 여러 파일로 분리하는 전략을 적용해야 합니다.

Mermaid 변환의 한계

Mermaid 다이어그램을 Excalidraw로 변환할 때, 플로차트 유형만 완전하게 지원되며, 기타 유형(시퀀스, 간트, 클래스 다이어그램 등)은 이미지로 렌더링되거나 변환이 제한적입니다. 따라서, 플로차트 외의 다이어그램은 변환 후 수동 편집이 필요합니다. 실제로 Mermaid 변환 도구를 사용할 때는 지원 범위를 명확히 파악하고, 복잡한 다이어그램은 변환 후 핵심 구조만 재작성하는 것이 효율적입니다. 또한, 변환 과정에서 발생하는 정보 손실을 최소화하기 위해, 주요 요소를 수동으로 보정하는 실무 전략이 필요합니다.

협업 URL 및 인프라 한계

셀프호스팅 환경에서는 협업 서버(WebSocket)의 URL이 빌드 타임에 하드코딩되어, 운영 중 URL 변경이 어렵습니다. 이로 인해, 운영 환경 변경 시 재빌드가 필요할 수 있으니, 초기 배포 시 협업 URL을 신중하게 설계해야 합니다. 실제로 운영 환경 변경이나 인프라 확장 시에는 협업 서버의 URL을 미리 계획하고, 필요 시 재빌드 및 배포 자동화 도구를 활용하여 운영 효율성을 높이는 것이 중요합니다.

우회 방안: 다이어그램 모듈화 및 워크플로 개선

대형 다이어그램은 8,000개 미만의 객체 단위로 분리해 관리하는 것이 성능 유지에 효과적입니다. PowerPoint 내보내기는 SVG로 변환한 후, EMF 포맷으로 재가공해 편집 가능한 객체로 삽입하는 우회 전략이 있습니다. 실시간 협업이 필요한 경우, excalidraw-room 서버를 별도로 구성해 확장성을 확보할 수 있습니다. 실제로 다이어그램 모듈화 전략을 적용하면 성능 저하를 예방할 수 있으며, 워크플로 개선을 통해 협업 품질과 데이터 관리 효율성을 높일 수 있습니다. 또한, PowerPoint 등 외부 도구와 연계하여 다이어그램을 다양한 포맷으로 활용하는 것이 실무적으로 유리합니다.

5.3.3 도입 시 흔한 실수 패턴과 방지 가이드

Excalidraw 도입 과정에서는 파일 집적, 협업 기능, 데이터 동기화, 변환 지원 범위 등에서 실무적 실수가 자주 발생합니다. 이 절에서는 주요 실수 패턴과 방지 가이드를 구체적으로 안내하여, 도입 리스크를 최소화할 수 있도록 지원합니다.

단일 파일 과도 집적의 위험

모든 다이어그램을 하나의 파일에 집적하는 경우, 8,000개 객체를 초과하면 성능이 급격히 저하됩니다. 따라서, 논리적 단위별로 다이어그램을 분할하고, 각 파일의 객체 수를 관리하는 것이 중요합니다. 실제로 대규모 프로젝트에서는 다이어그램을 기능별, 부서별, 업무별로 분할하여 관리하고, 정기적으로 파일 용량과 객체 수를 점검하는 것이 성능 유지에 효과적입니다.

협업 기능 과소평가

셀프호스팅 시 단순히 Docker 컨테이너만 배포하면 실시간 협업이 동작하지 않습니다. 별도의 `excalidraw-room`(WebSocket) 서버와 Redis 구성이 필수이며, 이를 간과하면 협업 품질이 크게 저하됩니다. 실제로 협업 기능을 제대로 활용하기 위해서는 `excalidraw-room` 서버와 Redis를 함께 구성하고, 네트워크 환경과 인증 체계를 조직 정책에 맞게 설계해야 합니다. 협업 품질을 높이기 위해 정기적으로 서버 상태를 점검하고, 장애 발생 시 신속한 대응 체계를 마련하는 것이 중요합니다.

Notion 임베딩과 데이터 동기화 오해

Excalidraw를 Notion에 임베딩할 때, 다이어그램 데이터가 Notion에 저장된다고 오해하는 경우가 많습니다. 실제로는 URL 인코딩 또는 `localStorage`에만 저장되므로, 데이터 동기화 및 백업에 주의해야 합니다. 실무에서는 다이어그램 데이터의 저장 위치와 동기화 방식, 백업 정책을 명확히 파악하고, 정기적으로 데이터 백업 및 복구 테스트를 실시하는 것이 중요합니다. 또한, 임베딩 시 데이터 손실을 예방하기 위해, 주요 다이어그램은 별도로 저장하고 관리하는 전략을 적용해야 합니다.

Mermaid 지원 범위 오해

Mermaid 전체 다이어그램 유형이 Excalidraw에서 완전히 지원된다고 오해하는 경우가 많습니다. 실제로는 플로차트만 완전 지원되며, 기타 유형은 이미지로만 렌더링됩니다. 도입 전 지원 범위를 명확히 파악하고, 필요 시 대체 워크플로를 마련해야 합니다. 실무에서는 Mermaid 변환 도구의 지원 범위와 한계를 사전에 검토하고, 복잡한 다이어그램은 수동으로 재작성하거나,

대체 도구를 활용하는 전략을 적용해야 합니다. 도입 과정에서 팀원 교육을 실시하여, 지원 범위와 워크플로를 명확히 안내하는 것이 리스크 예방에 효과적입니다.

부록

부록은 Excalidraw 백서의 전체 구조와 논리적 흐름을 체계적으로 정리하여, 독자가 필요한 정보를 신속하게 탐색할 수 있도록 돕는 역할을 합니다. 본 섹션에서는 11가지 카테고리과 핵심 주제별로 각 장이 어떻게 매핑되어 있는지 표와 해설을 통해 명확하게 제시합니다. 이를 통해 실무자, 도입 검토자, 기술 리더 등 다양한 독자들이 Excalidraw의 기술적 배경, 오픈소스 라이선스, 경쟁 제품 비교, 아키텍처, 실제 적용 사례 등 주요 논점을 한눈에 파악할 수 있습니다. 부록의 매핑 테이블은 각 장의 세부 내용을 빠르게 탐색할 수 있는 인덱스이자 가이드라인으로, 백서 전체를 효율적으로 활용하는 데 필수적인 도구입니다.

11가지 카테고리 -> 5장 매핑 테이블

카테고리별 장 매핑의 필요성

Excalidraw와 관련된 다양한 논점과 실무적 이슈를 명확하게 분류하고, 각 논점이 백서의 어느 장에서 다루어지는지 구조적으로 정리하는 것은 실무적 활용과 빠른 정보 접근에 매우 중요합니다. 아래 표는 Excalidraw 백서에서 다루는 11가지 핵심 카테고리가 각각 어떤 장에 배치되어 있는지를 일목요연하게 보여줍니다. 이 매핑 테이블은 각 카테고리별로 심도 있는 논의가 이루어지는 위치를 명확히 하여, 도입 검토자나 기술 리더가 특정 주제에 대한 세부 정보를 신속하게 찾을 수 있도록 지원합니다.

카테고리별 장 매핑은 Excalidraw 백서의 핵심 논점이 어디에서 집중적으로 다루어지는지 명확히 보여줍니다. 예를 들어, 탄생 배경은 1장에 집중되어 있고, 오픈소스 라이선스와 상용화는 1장과 5장에서 다루어집니다. 경쟁 제품 비교는 2장에, 해결하는 문제는 1장과 2장에 배치되어 있습니다. 핵심 기술 아키텍처는 4장에, 주요 쓰임새와 활용 시나리오는 3장과 4장에 집중되어 있습니다. 구축/적용 사례는 1장과 3장에, 사용 시 주의사항과 제약은 2장과 5장에 다루어집니다. 관련 기술 연동과 시너지는 3장과 4장에, 마이그레이션/도입 경로는 5장에, 대상 사용자와 관심 그룹은 1장에 배치되어 있습니다. 이러한 구조적 정리는 실무자와 기술 리더가 각 논점별로 필요한 정보를 빠르게 찾을 수 있도록 하며, 백서 전체의 논리적 흐름을 이해하는 데 큰 도움이 됩니다.

카테고리	배치된 장
카테고리 1: 탄생 배경	1장
카테고리 2: 오픈소스 라이선스와 상용화	1장, 5장
카테고리 3: 경쟁 제품 비교	2장
카테고리 4: 해결하는 문제	1장, 2장
카테고리 5: 핵심 기술 아키텍처	4장
카테고리 6: 주요 쓰임새와 활용 시나리오	3장, 4장
카테고리 7: 구축/적용 사례	1장, 3장
카테고리 8: 사용 시 주의사항과 제약	2장, 5장
카테고리 9: 관련 기술 연동과 시너지	3장, 4장
카테고리 10: 마이그레이션/도입 경로	5장
카테고리 11: 대상 사용자와 관심 그룹	1장

카테고리별 상세 해설

각 카테고리별로 Excalidraw 백서의 장 배치와 논의 내용을 구체적으로 해설하면 다음과 같습니다. 먼저, **탄생 배경**은 1장에서 Excalidraw의 출현, 설계 철학, 그리고 오픈소스 커뮤니티의 성장 과정을 다룹니다. 이 부분에서는 프로젝트의 시작 동기와 개발자들이 어떤 문제를 해결하고자 했는지, 그리고 커뮤니티가 어떻게 확장되었는지에 대한 상세한 설명이 포함됩니다. **오픈소스 라이선스와 상용화**는 1장과 5장에서 MIT 라이선스의 특징, 상업적 활용 가능성, 유료 플랜과의 차이점, 그리고 기업 도입 시의 라이선스 고려사항을 다루고 있습니다. 예를 들어, MIT 라이선스의 자유로운 사용과 수정, 배포 가능성, 그리고 상업적 활용 시의 법적 제약과 유료 플랜의 부가 기능을 비교 분석합니다.

경쟁 제품 비교는 2장에서 draw.io, Mermaid, tldraw, Miro 등과의 기능적, 생태계적 차별점을 심층적으로 분석합니다. 기능적 측면에서는 다이어그램 생성의 편의성, 협업 기능, 확장성 등을 비교하고, 생태계 측면에서는 커뮤니티 지원, 플러그인, 외부 연동 가능성 등을 논의합니다. **해결하는 문제**는 1장과 2장에서 기존 도구의 한계와 Excalidraw가 제공하는 혁신적 가치에 대해 구체적으로 서술합니다. 예를 들어, 기존 도구의 복잡한 UI, 제한된 협업 기능, 느린 성능 등의 문제를 Excalidraw가 어떻게 해결하는지 사례를 들어 설명합니다.

핵심 기술 아키텍처는 4장에서 이중 캔버스 구조, 성능 최적화, React/TypeScript 기반의 기술 스택을 집중적으로 다룹니다. 이중 캔버스 구조는 렌더링 성능을 높이고, React와 TypeScript의

조합은 유지보수성과 확장성을 강화합니다. **주요 쓰임새와 활용 시나리오**는 3장과 4장에서 AI 연동, Prompt Drawing, 실전 워크플로, 다양한 플랫폼 통합 사례를 다룹니다. 예를 들어, AI와의 연동을 통한 자동 다이어그램 생성, 실무 워크플로에서의 활용, Slack, Notion 등 다양한 플랫폼과의 통합 사례를 구체적으로 제시합니다.

구축/적용 사례는 1장과 3장에서 실제 도입 사례와 아키텍처 예제를 통해 실무적 적용 방법을 제시합니다. 예를 들어, 기업 내 협업 도구로서의 도입 사례, 교육 현장에서의 활용, 그리고 오픈소스 프로젝트에서의 적용 사례 등을 상세히 설명합니다. **사용 시 주의사항과 제약**은 2장과 5장에서 기능적 한계, 도입 시 흔한 실수, 성능 및 협업 이슈 등을 다룹니다. 예를 들어, 대규모 협업 시 발생할 수 있는 동기화 문제, 브라우저 호환성, 그리고 보안 이슈 등을 구체적으로 안내합니다.

관련 기술 연동과 시너지는 3장과 4장에서 AI Agent, MCP 프로토콜, Mermaid 등과의 연동 시나리오를 설명합니다. AI Agent와의 연동을 통해 자동화된 다이어그램 생성, MCP 프로토콜을 통한 외부 도구와의 통신, 그리고 Mermaid와의 통합을 통한 워크플로 확장 사례를 다룹니다. **마이그레이션/도입 경로**는 5장에서 Visio, draw.io, Lucidchart 등에서의 전환 전략과 실무적 고려사항을 안내합니다. 예를 들어, 기존 데이터의 변환 방법, 사용자 교육, 시스템 통합 전략 등을 구체적으로 설명합니다. **대상 사용자와 관심 그룹**은 1장에서 각 직군별 활용 가치와 채택 현황을 정리합니다. 개발자, 디자이너, 기획자 등 다양한 직군이 Excalidraw를 어떻게 활용하고 있는지, 그리고 각 그룹의 주요 관심사를 분석합니다.

이처럼 카테고리별 장 매핑과 상세 해설은 Excalidraw 백서의 전체 구조를 명확히 파악하고, 각 논점별로 심도 있는 정보를 빠르게 탐색하는 데 필수적인 역할을 합니다.

핵심 주제 -> 장 매핑 테이블

핵심 주제별 장 매핑의 중요성

Excalidraw 백서의 두 가지 핵심 주제—“기업 표준 그리기 도구로서의 가능성”과 “AI Agent 기반 Prompt Drawing 도구로의 활용 가능성”—가 각각 어떤 장에서 심층적으로 다루어지는지 명확히 하는 것은, 독자가 자신의 관심사에 따라 백서를 효율적으로 탐색하는 데 큰 도움이 됩니다. 아래 표는 각 핵심 주제가 어디에서 주요 논의 대상으로 다루어지는지를 정리합니다.

핵심 주제별 장 매핑은 독자가 자신의 목적에 맞는 정보를 신속하게 찾을 수 있도록 도와줍니다. 예를 들어, 기업 표준 그리기 도구로서의 가능성은 1장, 2장, 5장에서 주요 논의가 이루어지며,

4장에서는 보조적 논의가 포함됩니다. AI Agent 기반 Prompt Drawing 도구 활용 가능성은 3장에서 집중적으로 다루어지며, 2장과 4장에서는 보조적 논의가 이루어집니다. 이러한 구조적 매핑은 백서 전체를 효율적으로 탐색하고, 각 장의 세부 내용을 체계적으로 이해하는 데 중요한 역할을 합니다.

핵심 주제	주요 배치 장	보조 배치 장
Excalidraw가 기업 표준 그리기 도구가 될 수 있는가	1장, 2장, 5장	4장
AI Agent를 활용한 Prompt Drawing 도구 활용 가능성	3장	2장, 4장

핵심 주제별 상세 해설

Excalidraw의 기업 표준 그리기 도구 가능성은 여러 장에서 다양한 관점으로 심층적으로 다루어집니다. 1장에서는 Excalidraw의 탄생 배경, 설계 철학, 오픈소스 라이선스, 그리고 주요 사용자 그룹을 중심으로 서술됩니다. 이 부분에서는 Excalidraw가 왜 기업 표준 도구로서 적합한지, 오픈소스의 장점과 커뮤니티의 성장, 그리고 다양한 직군의 활용 사례를 구체적으로 설명합니다. 2장에서는 경쟁 제품과의 비교, 기능적 포지셔닝, 그리고 기업 도입 시의 장단점 분석이 이루어집니다. 예를 들어, draw.io, Miro 등과의 비교를 통해 Excalidraw의 협업 기능, 확장성, 비용 효율성 등을 분석하고, 기업 환경에서의 도입 시 고려해야 할 요소를 상세히 설명합니다. 5장에서는 실제 도입 로드맵, 셀프호스팅과 SaaS 비교, 마이그레이션 전략 등 실무적 관점에서의 표준화 방안이 다루어집니다. 이 장에서는 기업이 Excalidraw를 도입할 때의 단계별 전략, 인프라 구축 방법, 데이터 이전, 사용자 교육 등 실무적인 내용을 구체적으로 안내합니다. 4장에서는 플랫폼 통합, 라이브러리 활용, 내보내기 전략 등 보조적 논의가 이루어집니다. 예를 들어, Slack, Notion 등과의 통합, 외부 라이브러리 활용, 다양한 내보내기 옵션을 통해 기업 환경에서의 활용성을 높이는 방안을 설명합니다.

AI Agent 기반 Prompt Drawing 도구 활용 가능성은 3장에서 집중적으로 다루어집니다. 이 장에서는 MCP 프로토콜, Claude/ChatGPT 연동, 실전 Prompt Drawing 워크플로, 실제 아키텍처 예제 등 AI 연동의 핵심 기술과 실무 적용 방법을 상세히 설명합니다. 예를 들어, AI에게 자연어로 지시하여 다이어그램을 자동 생성하는 방식, MCP 프로토콜을 통한 AI와 Excalidraw의 통신 구조, 실제 워크플로 예제, 그리고 Claude/ChatGPT와의 연동 사례를 구체적으로 제시합니다. 2장과 4장에서는 AI 통합 관점의 경쟁 우위, 하이브리드 파이프라인, 플랫폼 연동 등 보조적

논의가 포함됩니다. 예를 들어, AI 통합을 통한 경쟁 제품 대비 우위, 하이브리드 파이프라인 구축 방법, 다양한 플랫폼과의 연동 전략 등을 설명합니다.

이와 같이 부록의 매핑 테이블과 상세 해설은 Excalidraw 백서의 전체 구조와 논리적 흐름을 명확히 제시함으로써, 독자가 자신의 목적에 맞는 정보를 신속하게 찾고, 각 장의 세부 내용을 체계적으로 이해하는 데 중요한 역할을 합니다. 이러한 인덱스와 해설은 실무자, 기술 리더, 도입 검토자 등 다양한 독자층이 백서를 효율적으로 활용하는 데 필수적인 가이드라인이 됩니다.

Appendix

References

1. Excalidraw Docs. “Architecture and Performance”. <https://github.com/excalidraw/excalidraw/blob/master/docs/architecture.md>
2. Excalidraw Docs. “Excalidraw Libraries”. <https://libraries.excalidraw.com/>
3. Excalidraw Docs. “Excalidraw Marketplace”. <https://marketplace.excalidraw.com/>
4. Excalidraw Docs. “Excalidraw Plus for Confluence”. <https://marketplace.atlassian.com/apps/1228171/excalidraw-plus>
5. Excalidraw Docs. “Exporting and embedding scenes”. <https://github.com/excalidraw/excalidraw/blob/master/docs/scene-embedding.md>
6. Excalidraw Docs. “NPM package for React”. <https://www.npmjs.com/package/@excalidraw/excalidraw>
7. Excalidraw Docs. “Obsidian Excalidraw Plugin”. <https://github.com/zsviczian/obsidian-excalidraw-plugin>
8. Excalidraw Docs. “excalidraw-vscode: Visualize and edit diagrams in VS Code”. <https://marketplace.visualstudio.com/items?itemName=pomdtr.excalidraw-editor>

9. Excalidraw 공식 사이트.<https://excalidraw.com>
10. Excalidraw 백서 목차(내부 문서)
11. Excalidraw+ 소개.<https://plus.excalidraw.com>
12. Excalidraw. (2024). “ExcaliDash: Collaboration Server”.<https://github.com/excalidraw/excalidash>
13. Excalidraw. (2024). “Excalidraw Documentation”.<https://excalidraw.com/>
14. Excalidraw. (2024). “Excalidraw+ SaaS”.<https://plus.excalidraw.com>
15. Excalidraw. (2024). “Self-hosting Excalidraw”.<https://github.com/excalidraw/excalidraw>
16. Excalidraw. (2024). “excalidraw-converter”.<https://github.com/edwingao28/excalidraw-converter>
17. Excalidraw. (2024). “excalidraw/excalidraw”.<https://github.com/excalidraw/excalidraw>
18. Excalidraw. (2024). “mermaid-to-excalidraw”.<https://mermaid-to-excalidraw.vercel.app>
19. GitHub. (2024). “colem00/excalidraw-diagram-skill”.<https://github.com/colem00/excalidraw-diagram-skill>
20. GitHub. (2024). “excalidraw/excalidraw”.<https://github.com/excalidraw/excalidraw>
21. GitHub. (2024). “excalidraw/mermaid-to-excalidraw”.<https://github.com/excalidraw/mermaid-to-excalidraw>
22. GitHub. (2024). “mermaid-js/mermaid”.<https://github.com/mermaid-js/mermaid>
23. GitHub. (2024). “tldraw/tldraw”.<https://github.com/tldraw/tldraw>
24. GitHub. (2024). “yctimlin/mcp_excalidraw”.https://github.com/yctimlin/mcp_excalidraw
25. Mermaid. (2024). “Mermaid Documentation”.<https://mermaid-js.github.io/mermaid/#/>
26. Meta, Christopher Chedeau 트윗.<https://twitter.com/vjeux/status/1212057>

795145932800

27. Miro. (2024). “Miro Help Center” .<https://help.miro.com/>
28. ROSS Index.<https://rossindex.com>
29. TheirStack.<https://www.theirstack.com/tools/excalidraw>
30. coleam00/excalidraw-diagram-skill. (2024). “Excalidraw Diagram Skill” .<https://github.com/coleam00/excalidraw-diagram-skill>
31. draw.io. (2024). “draw.io Documentation” .<https://www.diagrams.net/doc/>
32. edwingao28/excalidraw-skill. (2024). “Excalidraw Skill” .<https://github.com/edwingao28/excalidraw-skill>
33. excalidraw-brute-export-cli. (2024). “Excalidraw Brute Export CLI” .<https://github.com/edwingao28/excalidraw-brute-export-cli>
34. excalidraw/excalidraw-mcp. (2024). “Excalidraw MCP Server” .<https://github.com/excalidraw/excalidraw-mcp>
35. excalidraw/mermaid-to-excalidraw. (2024). “Mermaid to Excalidraw” .<https://github.com/excalidraw/mermaid-to-excalidraw>
36. excalirender. (2024). “Excalirender” .<https://github.com/edwingao28/excalirender>
37. <https://excalidraw.com/>
38. <https://github.com/excalidraw/excalidraw>
39. libraries.excalidraw.com. (2024). “Excalidraw Libraries” .<https://libraries.excalidraw.com>
40. mermaid-to-excalidraw.vercel.app. (2024). “Mermaid to Excalidraw Playground” .<https://mermaid-to-excalidraw.vercel.app>
41. tldraw. (2024). “tldraw Documentation” .<https://www.tldraw.com/docs>
42. yctimlin/mcp_excalidraw. (2024). “MCP Excalidraw Skill” .https://github.com/yctimlin/mcp_excalidraw

Glossary

용어	정의
@excalidraw/excalidraw	React 앱에 Excalidraw 화이트보드를 임베딩할 수 있는 공식 NPM 패키지
라이브러리(library)	Excalidraw에서 재사용 가능한 아이콘, 컴포넌트, 템플릿의 집합
셀프호스팅	오픈소스 소프트웨어를 자체 인프라에 직접 배포하여 운영하는 방식
씬 데이터(scene data)	Excalidraw 다이어그램의 모든 객체와 메타데이터를 포함하는 JSON 구조 데이터
카테고리	Excalidraw 백서에서 다루는 논점의 분류 단위
템플릿(template)	특정 목적의 다이어그램 구조를 미리 정의한 샘플 파일
핵심 주제	백서 전체를 관통하는 주요 논의 축
ACR Tasks	Azure Container Registry의 자동화된 컨테이너 빌드 및 배포 워크플로
BPMN	Business Process Model and Notation, 비즈니스 프로세스 표준 표기법
Confluence	Atlassian의 엔터프라이즈 위키 및 협업 플랫폼
E2E 암호화	End-to-End 암호화, 송수신자 간 데이터 보호 방식
EMF	Enhanced Metafile, Windows 벡터 그래픽 포맷
EMF(Enhanced Metafile)	Windows 환경에서 벡터 그래픽을 저장하는 포맷
ERD	Entity-Relationship Diagram, 데이터베이스 구조 설계 다이어그램
ExcaliDash	Excalidraw 협업 환경을 위한 Docker Compose 배포 도구
Excalidraw	오픈소스 손그림 스타일 다이어그램/화이트보드 도구
Excalidraw JSON	Excalidraw 다이어그램의 내부 데이터 포맷
excalidraw-plugin	Obsidian에서 Excalidraw를 통합·활용할 수 있게 해주는 플러그인
excalidraw-room	실시간 협업을 위한 WebSocket 서버 컴포넌트
excalidraw-vscode	VS Code에서 Excalidraw 다이어그램을 생성·편집할 수 있는 확장 프로그램
exportToBlob()	Excalidraw 다이어그램을 PNG 등 비트맵 이미지로 내보내는 API
exportToSvg()	Excalidraw 다이어그램을 SVG 벡터 이미지로 내보내는 API
Headless 렌더링	UI 없이 서버에서 이미지 파일로 자동 변환하는 렌더링 방식
Interactive Canvas	UI 오버레이(선택, 드래그 등)를 담당하는 별도 캔버스 레이어
LDAP	Lightweight Directory Access Protocol, 디렉터리 서비스 프로토콜
LLM	Large Language Model, 대형 언어 모델
MCP	Model Context Protocol, AI와 외부 도구 간 통신을 위한 프로토콜
Memoization	동일 입력에 대해 결과를 캐싱하여 불필요한 연산을 방지하는 최적화 전략

MIT 라이선스	상업적 사용, 수정, 배포가 자유로운 오픈소스 라이선스
Notion	노트, 데이터베이스, 위키 기능을 통합한 생산성 SaaS 플랫폼
Obsidian	마크다운 기반의 지식 관리 및 노트 툴
PoC	Proof of Concept, 개념 검증 단계
Prompt Drawing	AI에게 자연어로 지시하여 다이어그램을 자동 생성하는 방식
RAG	Retrieval-Augmented Generation, 외부 지식 검색과 LLM 생성을 결합하는 AI 파이프라인
RBAC	Role-Based Access Control, 역할 기반 접근 제어
Render Throttling	렌더링 빈도를 제한하여 UI 성능 저하를 방지하는 기법
ROSS Index	오픈소스 소프트웨어 인기 순위 지표
RoughJS	손그림 스타일 SVG/Canvas 렌더링 라이브러리
SaaS	Software as a Service, 클라우드 기반 소프트웨어 서비스
Skill	AI Agent 또는 MCP 클라이언트에서 특정 기능(다이어그램 생성 등)을 수행하는 플러그인 또는 모듈
SOC 2	Service Organization Control 2, 클라우드 서비스 보안 인증
SSO	Single Sign-On, 단일 계정으로 여러 시스템에 로그인하는 인증 방식
Static Canvas	Excalidraw에서 드로잉 요소를 렌더링하는 기본 캔버스 레이어
tEXt 칭크	PNG 파일 내에 메타데이터를 저장하는 영역
TheirStack	기업별 소프트웨어 도입 현황 데이터베이스
UML	Unified Modeling Language의 약자로, 소프트웨어 설계 표준 다이어그램 언어
Viewport Culling	화면에 보이는 요소만 렌더링하여 성능을 최적화하는 기법

Contact Us

 hello@cncf.co.kr

 02-469-5426

 www.cncf.co.kr

CNF Blog

다양한 콘텐츠와 전문 지식을 통해 더 나은 경험을 제공합니다.

CNF eBook

이제 나도 클라우드 네이티브 전문가
쿠버네티스 구축부터 운영 완전 정복

CNF Resource

Community Solution의 최신 정보와
유용한 자료를 만나보세요.

