

# RAG 제조/금융 사례

- 모두가 MCP/Agent 할 때, Braincrew는 핵심을 본다

# CONTENTS

01 격동하는 AI 시대의 흐름

02 실제로 RAG를 어떻게 다루지?

03 실패 없는 AX 로드맵

## PART 01

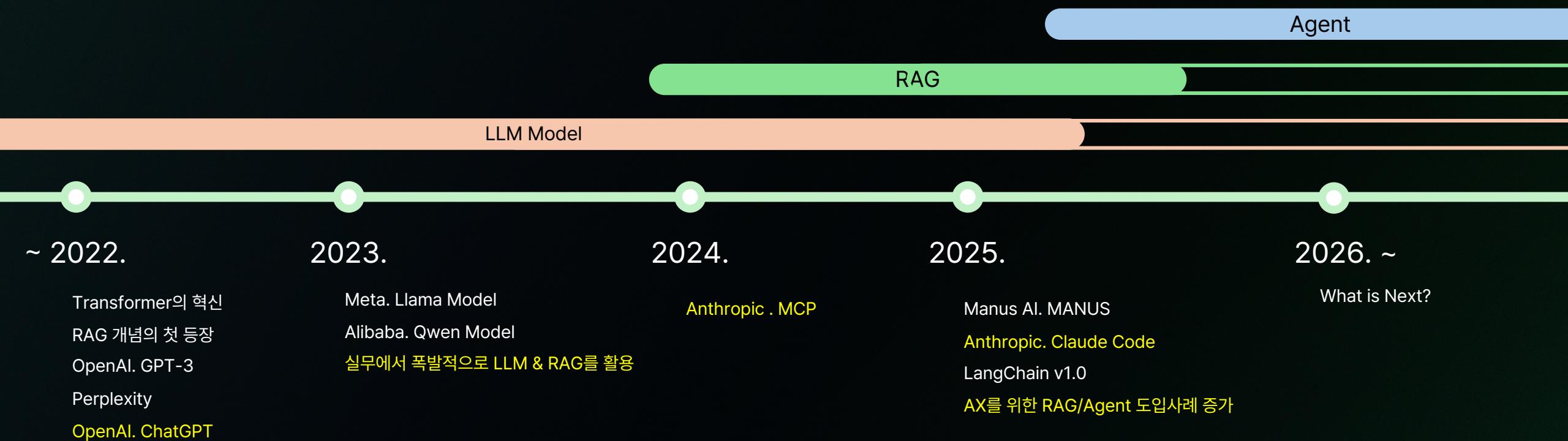
# 격동하는 AI 시대의 흐름

우리는 정말로 AI를 명확하게 알고 사용할까?

01

# 여러분께서는 AI를 어떻게 사용하고 계신가요?

우리가 겪어온 AI의 발전 패러다임



## PART 02

# 실제로 RAG를 어떻게 다루지?

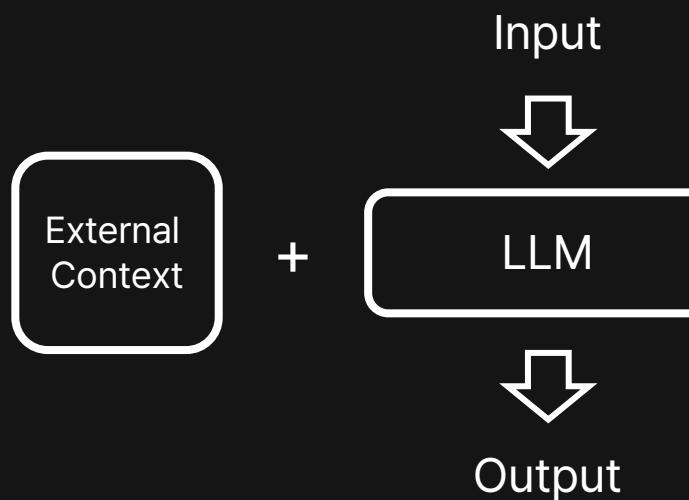
다양한 도메인에서의 성공적인 RAG 적용사례

02

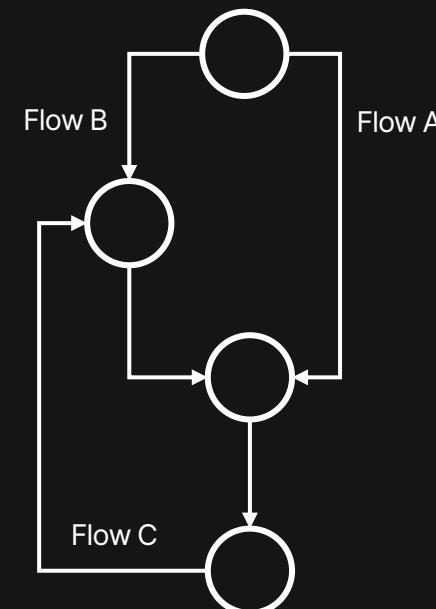
# RAG vs. Workflow vs. Agent

3가지 핵심 개념 비교

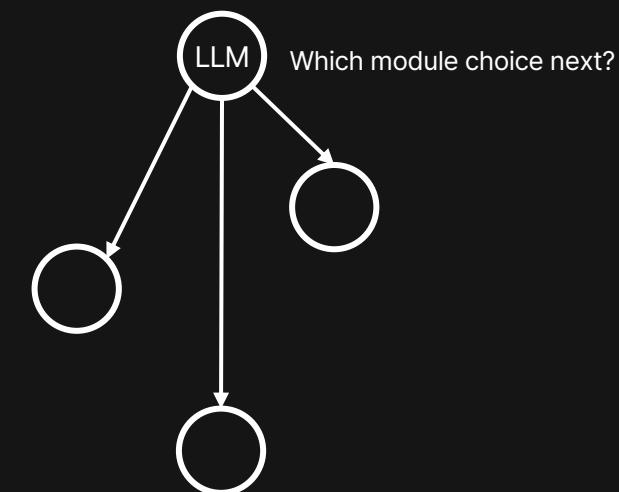
## [ RAG ]



## [ Workflow ]



## [ Agent ]



- \* LLM does not know domain-specific contents
- \* RAG can generate domain-specific answer from external context

- \* Static Flow
- \* High Consistency
- \* Hard Implementation

- \* Dynamic Flow
- \* Low Consistency
- \* Low Implementation

# 실제 적용 사례 : 제조 도메인

## 건설설계 기준 통합검색 플랫폼



[Figure 01] 제조 도메인 관련문서 예시

- \* 실무현장자들이 건설작업 진행 시 규제를 찾아보고 명확하게 파악하는데 과도하게 많은 시간과 인력이 소비되어 AI를 활용해 본 과정의 병목을 해소하고자 함.
- \* 시간을 단축하면서 동시에 정확한 규제와 관련된 정보를 찾아야 함.
- \* 다양한 형태와 타입의 기준 문서 (e.g. 건설규정 집, 산업규제 가이드)

높은 정확도

확실한 출처

데이터 가공

Multi-Type

# 실제 적용 사례 : 제조 도메인

건설설계 기준 통합검색 플랫폼

Domain Expert의  
원활한 소통

Multi-Type 데이터  
전처리 전략 수립

건설 규정에 대한  
검색 정확도 **85%**

\* General Performance 50%

# 실제 적용 사례 : 금융 도메인

## 기업대출 승인신청서 초안작성 서비스

A회사 재무상태표 (2015. 12. 31 현재)		(단위 : 억원)	
차 번	대 번		
유동자산	703	유동부채	533
당좌자산	535	매입채무	156
현금 및 현금성 자산	79	단기차입금	157
매출채권	260	기타유동부채 등	220
기타당좌자산 등	196	비유동부채	268
재고자산	168	장기차입금	133
비유동자산	960	회사채	69
투자자산	284	기타비유동부채	66
유형자산	604	자본	862
건물·구축물	151	자본금	113
기계	187	자본잉여금	165
기타유형자산 등	266	이익잉여금 등	584
무형자산	35		
기타비유동자산	37		
자산 총계	1,663	부채와 자본 총계	1,663

[Figure 02] 금융 도메인 관련문서 예시

- \* 기업대출을 받는 고객들이 대출을 받기 위해서는 은행의 인력들이 해당 기업의 **파편화된 정보**를 수집하여 **대출승인을 받을 수 있는지** 면밀히 검토하는 서류를 작성하여야 한다. 즉, 정보수집부터 서류작성에 기본적으로 3~4일 이상 소요되기에 해당 병목을 AI를 통해 해소하고자 함.
- \* **파편화된 정보**를 수집하고, 수집된 자료를 선별하여 검토 후 최종적으로 서류를 작성해야 함.
- \* **파편화된 정보**를 자동화 수집
- \* 수치의 정확도 보장 & 재무흐름에 대한 검토의견

수집 자동화

자료의 수치보존

데이터 가공

확실한 출처

# 실제 적용 사례 : 금융 도메인

기업대출 승인신청서 초안작성 서비스

Domain Expert의  
원활한 소통

수치 데이터의  
**Value** 보존

E2E 사용자의  
사용 만족도 **80%**

\* General Performance 60%

## PART 03

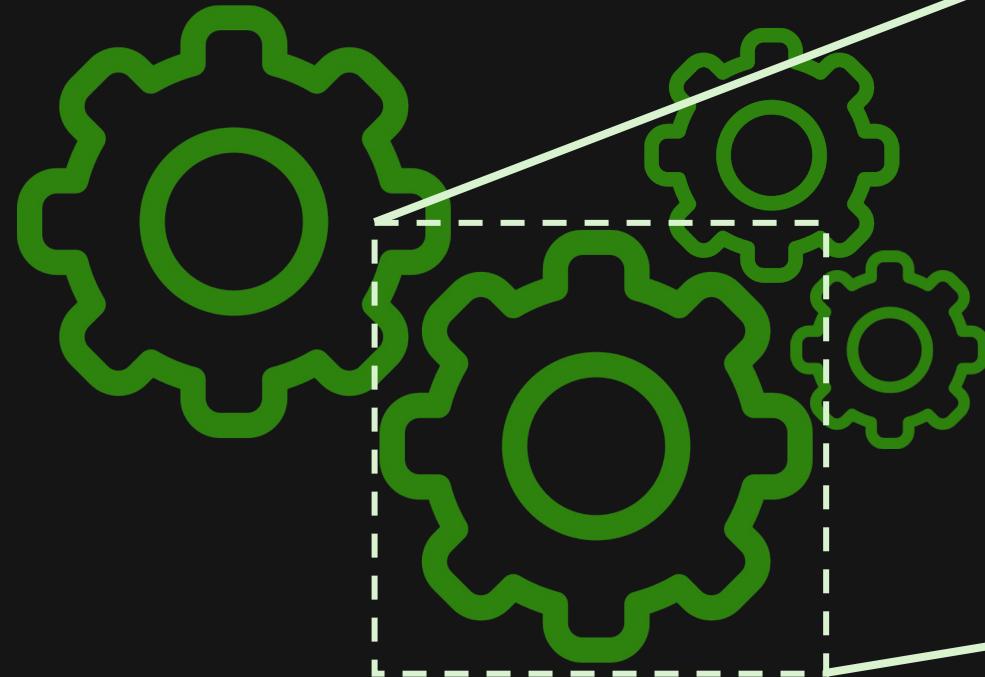
# 실패 없는 AX 비법

효과적으로 RAG/Agent를 적용하기 위한 Braincrew의 핵심

03

# 먼저, '기술'이 아닌 '문제'에 집중

AI는 기존과 다른 방식으로의 변화를 주는 기술적 '도구'이다



Modeling

RAG

MCP

Agent

Data Preprocessing

Infrastructure

Which is the **Best Choice** for you?

# 구체적인 '상황'을 더 면밀하게 파악

누가, 어디에, 어떻게, 왜 쓰는지를 파악하자

“사내 챗봇 한번 만들어보자”

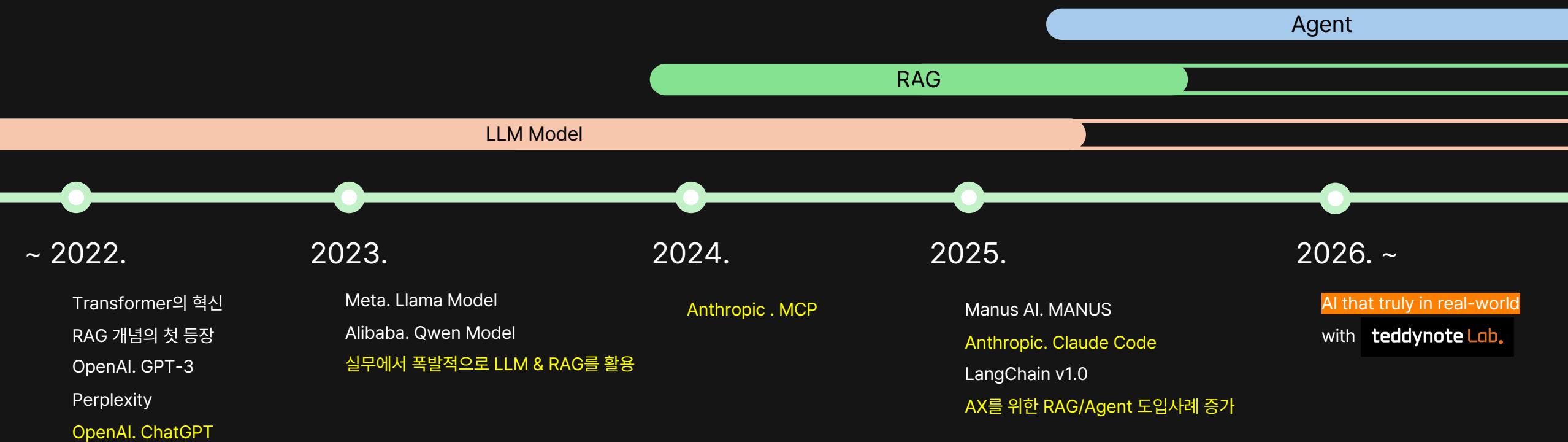
	<b>Who</b>	사내 임직원들이
	<b>Where</b>	사내 플랫폼위에서
	<b>How</b>	Chatbot 형태로
	<b>Why</b>	무수히 많은 사내 규정을 찾아보는 것이 어려워

“OK, 사용자의 개인별 상황에 맞는 사내 규정을 찾고, 그것에 답할 수 있는 Chatbot이 필요하겠구나.”

( 실제 플랫폼 위에서 사용하게 될 것이니 인프라도 고려하지만 임직원들이 사용해야 할 테니까 Chatbot의 UI는 확장성 또는 유연한 답변이 가능하게 해야겠어. 고정된 Workflow인 RAG 보다는 Agent 형태의 flow를 하는게 나을지도 모르겠는걸? )

# 남은 것은 '기술적 실현' 뿐

실패없이 실무에 사용할 수 있는 RAG/Agent가 만들어진다



# Thank You