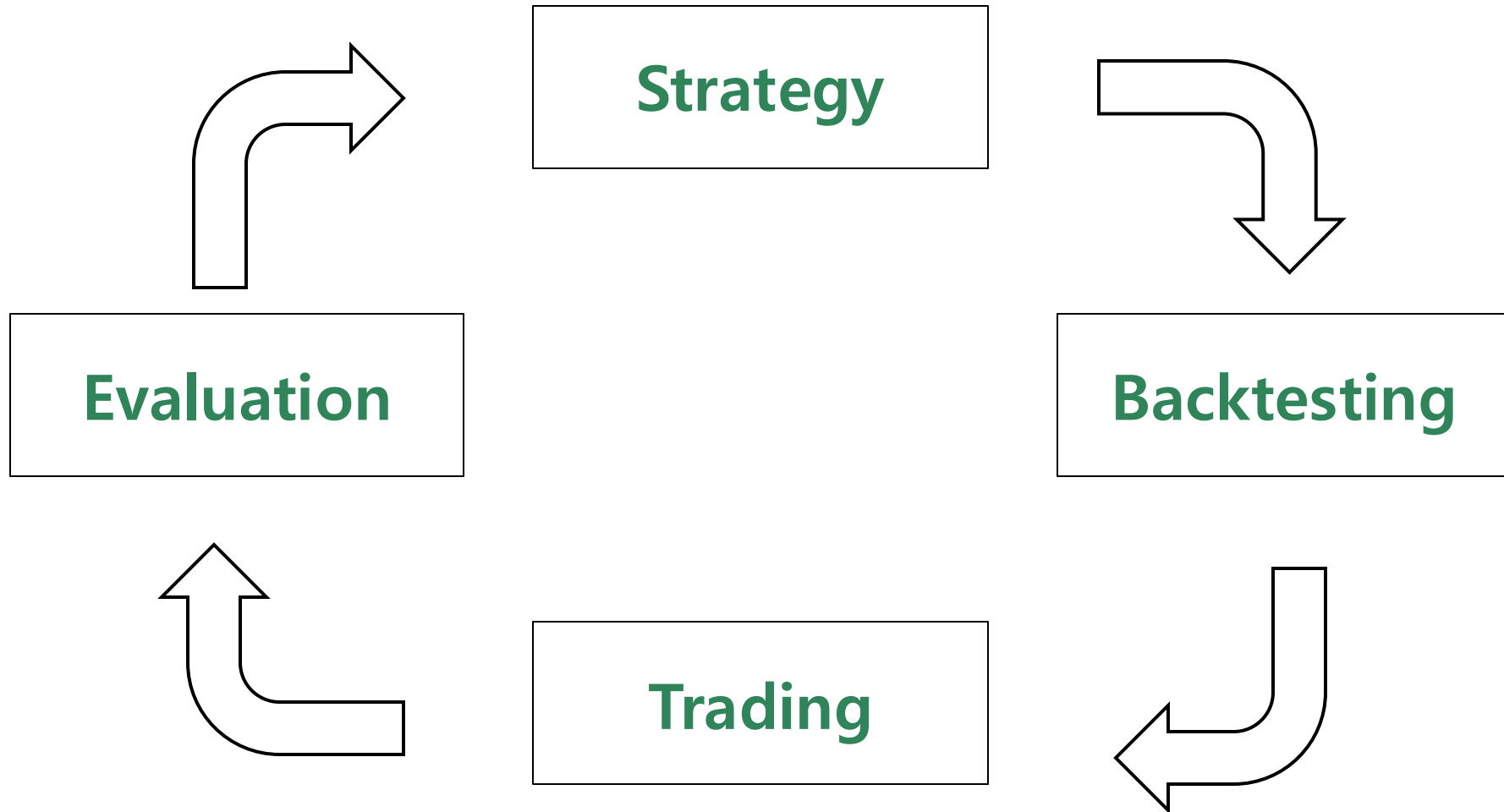


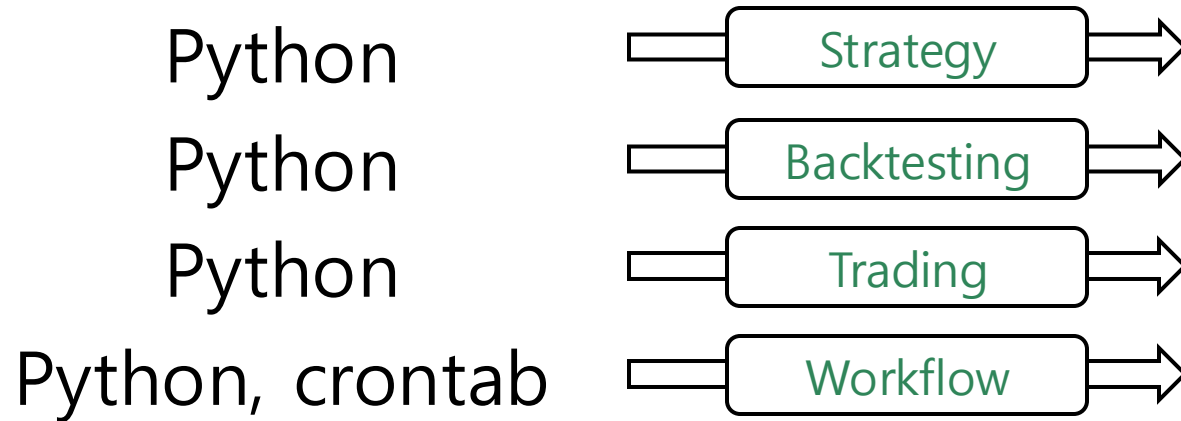
QuantNode.kr

퀀트 투자의 Workflow



Why n8n & MCP?

As is



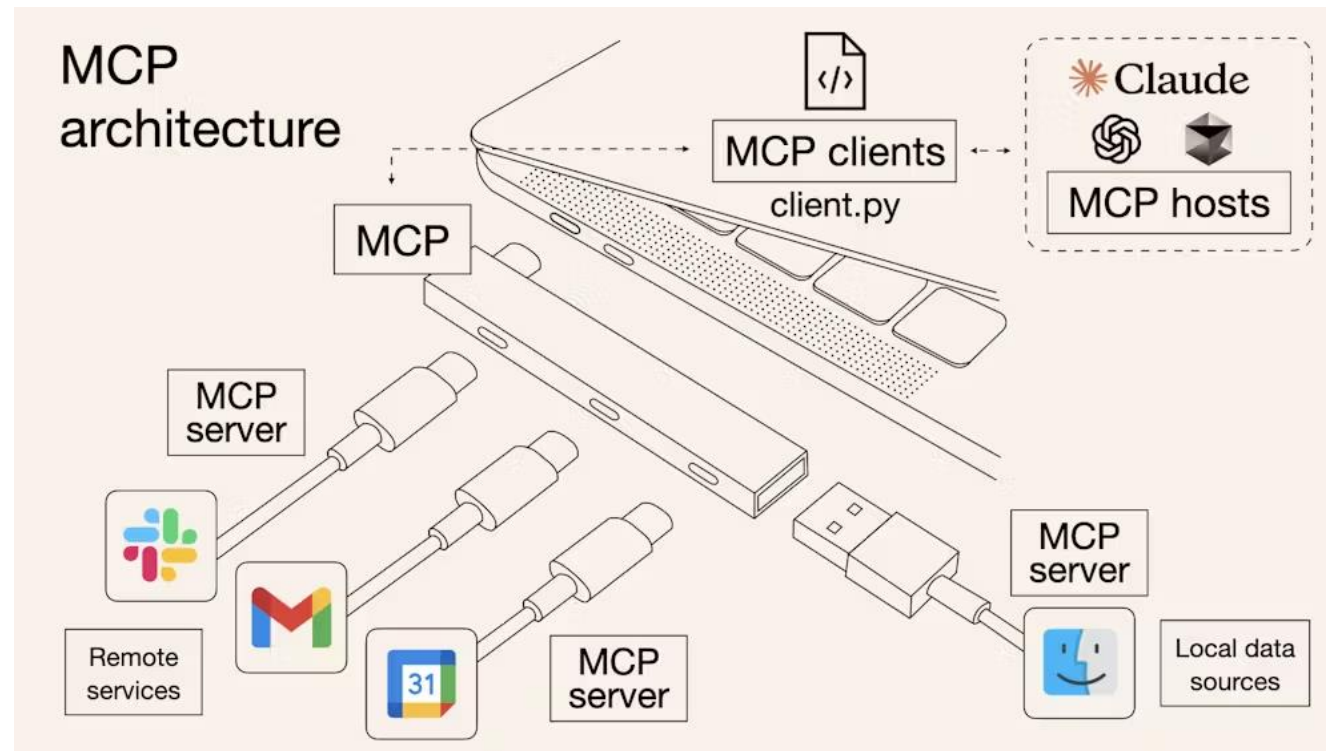
높은 구현 및 관리 복잡성

To be

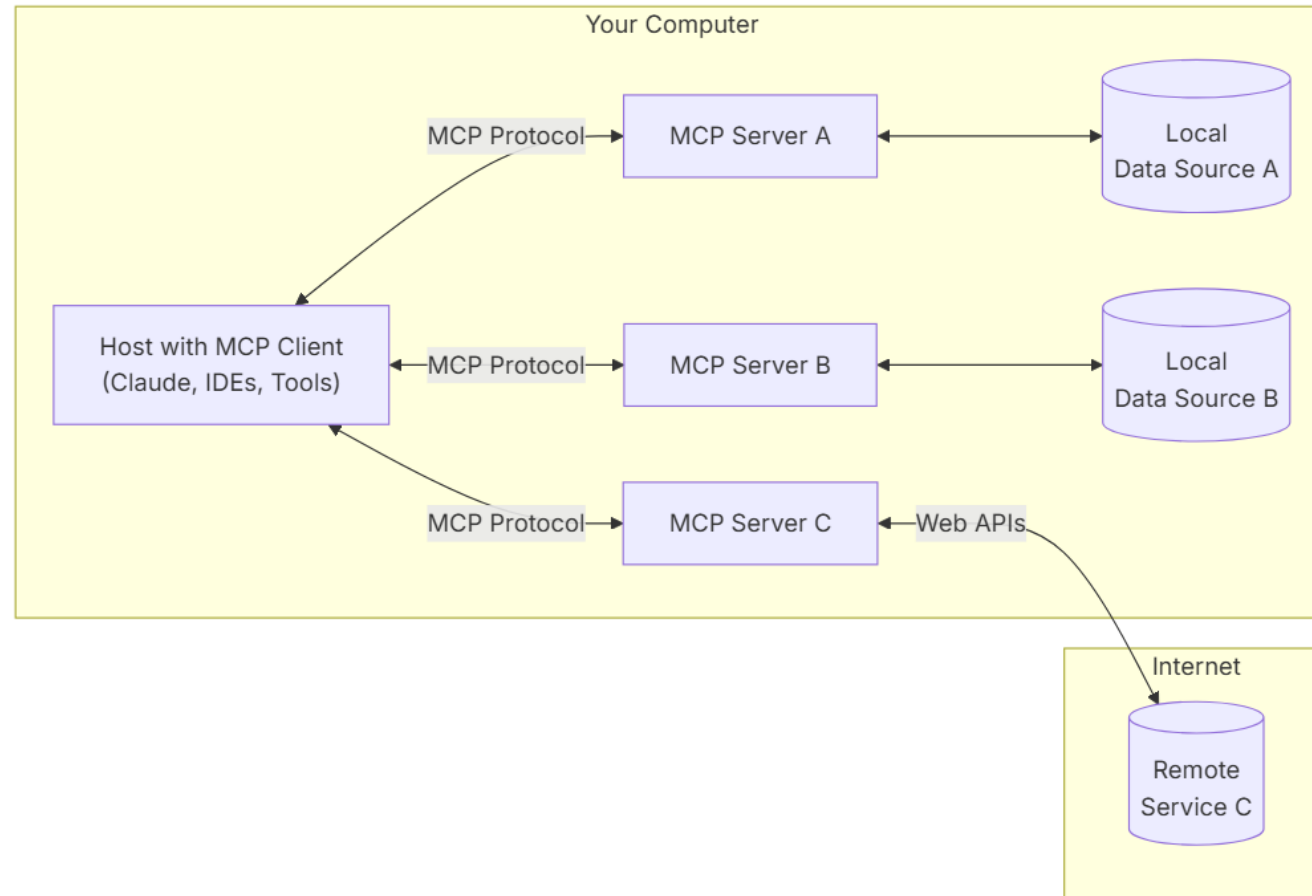
Prompt
TBD (MCP?)
MCP
n8n

간편한 구현 및 관리

MCP – AI의 USB-C



General architecture



MCP의 핵심 요소

Tools

Prompts

Resources

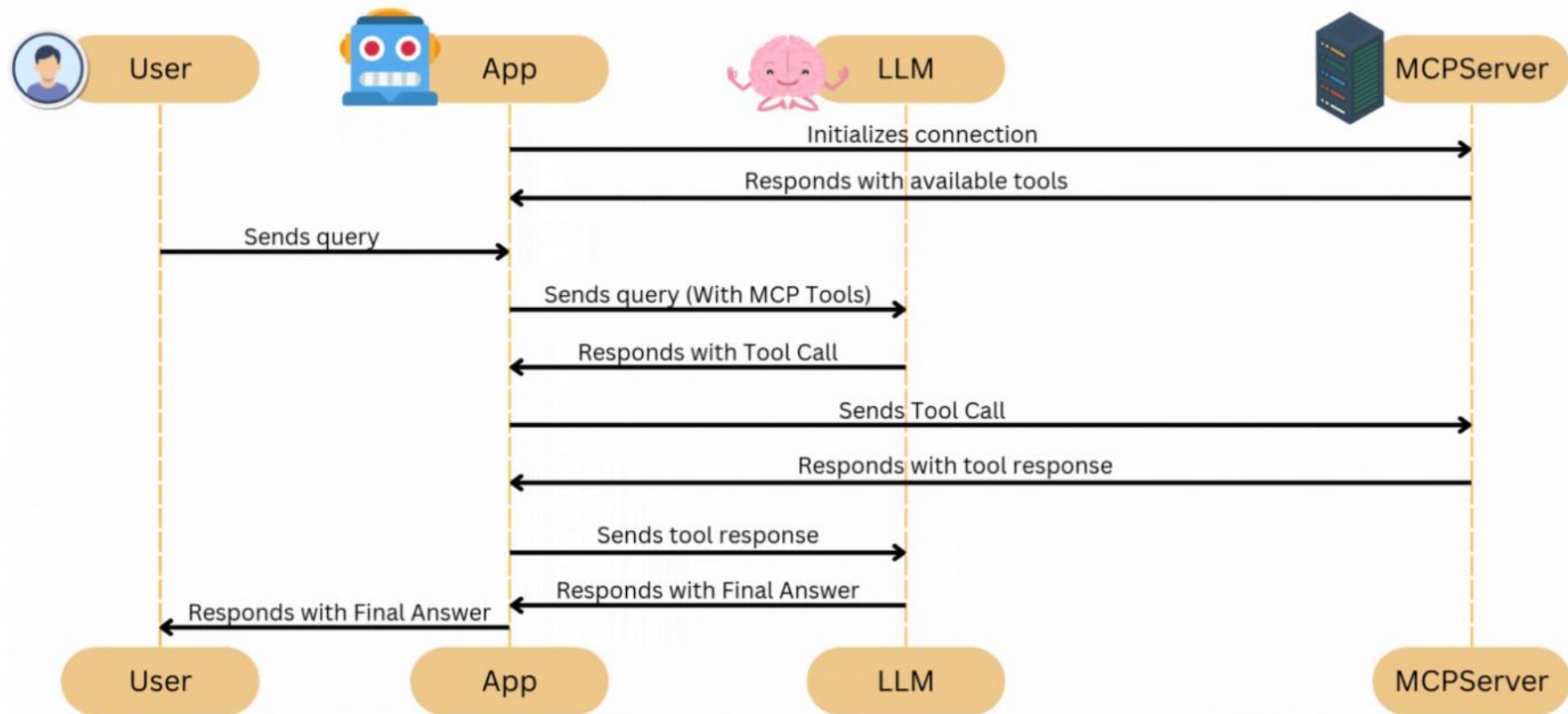
Sampling

Roots

Tools

- What
 - LLM이 실행 가능한 MCP 서버의 기능
 - 예: Send message, Update DB, Take action
- When
 - LLM이 판단해 자동으로 호출 (사용자 승인 포함 가능)

MCP의 Tool 사용 과정



Transports

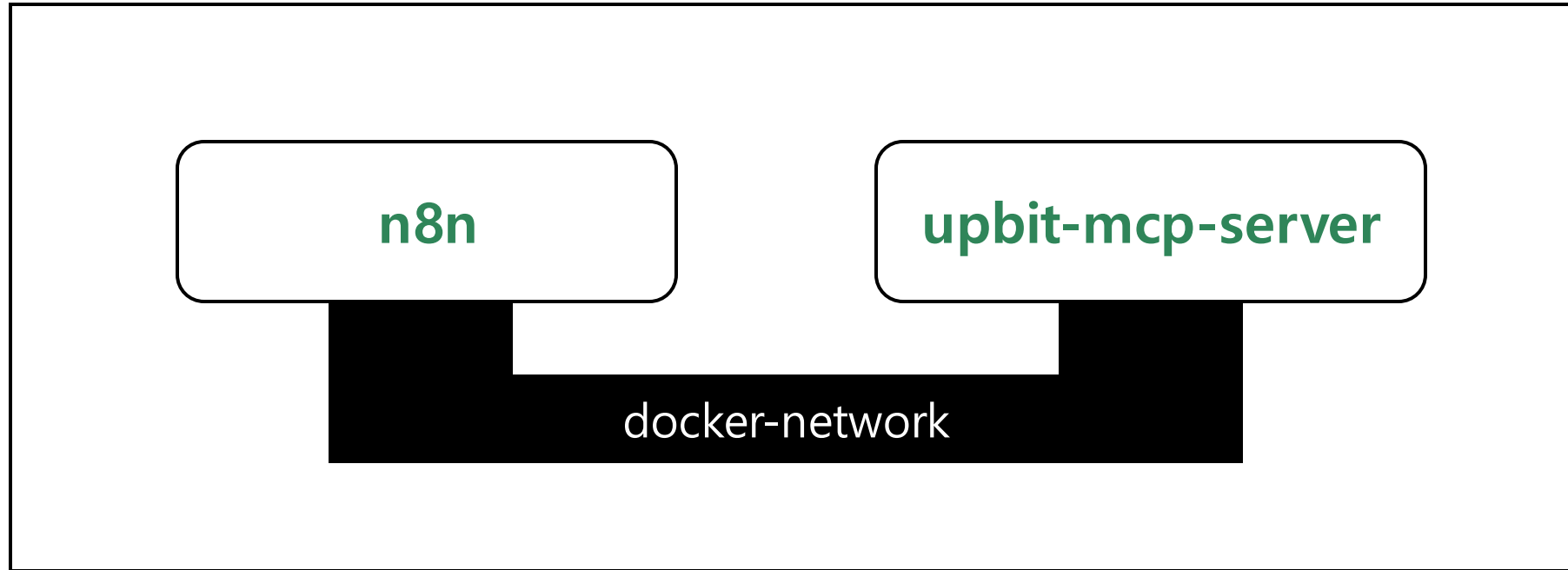
- What
 - MCP 클라이언트 ↔ 서버 간 메시지 전송 메커니즘
 - JSON-RPC 2.0 기반 메시지 포맷 사용
 - Request / Response / Notification

STDIO

SSE*

**Streamable
HTTP**

시스템 구성

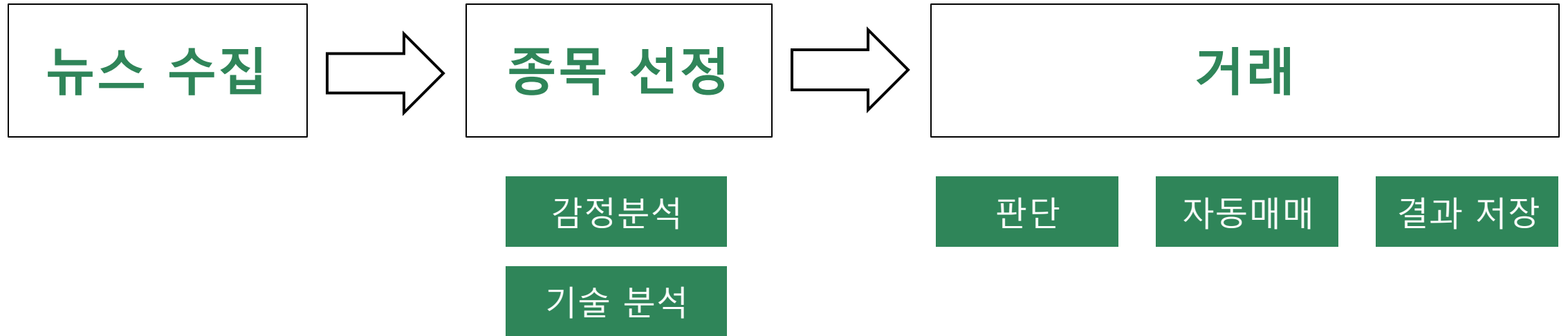


Upbit MCP Server

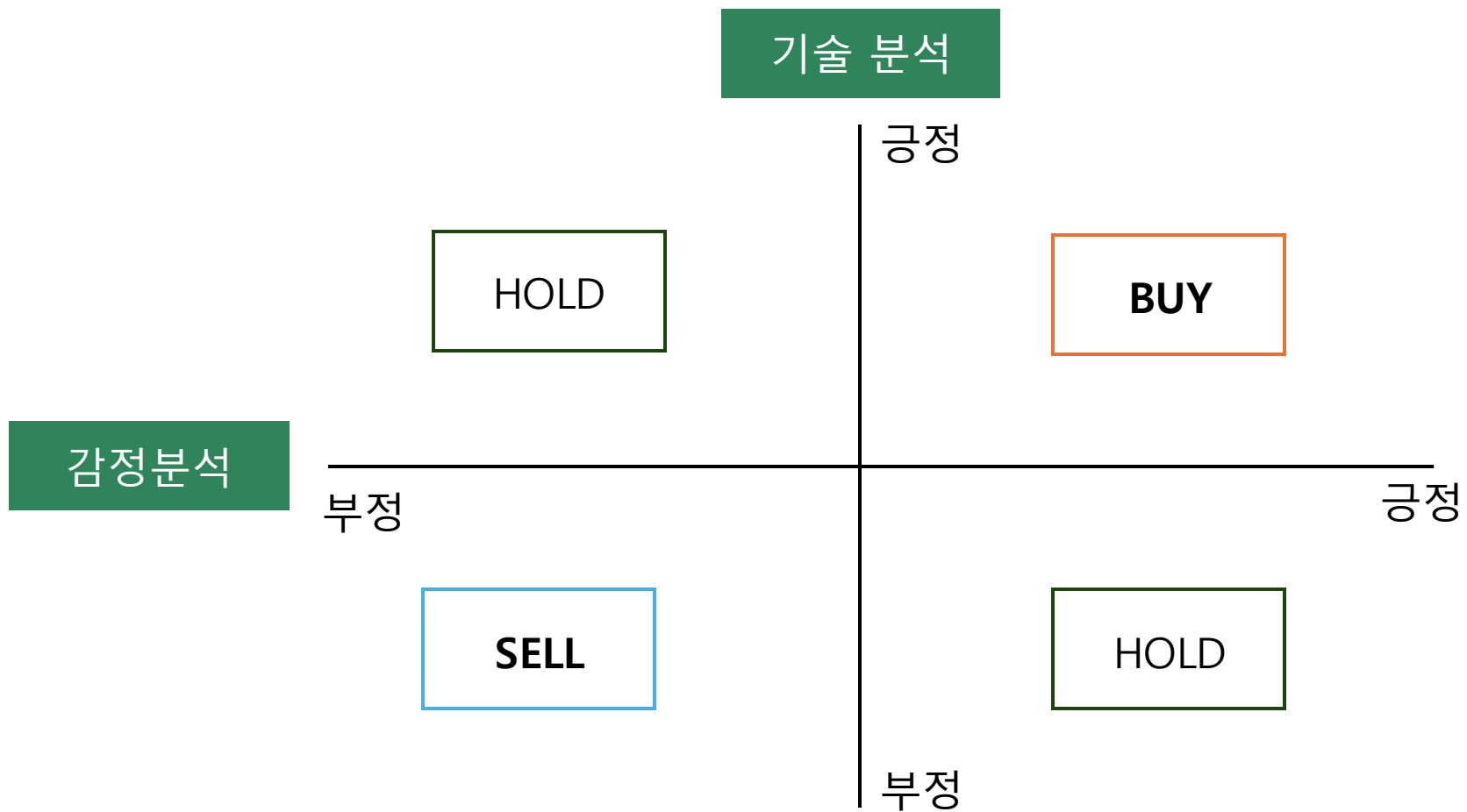
- 기존 **STDIO** 기반 MCP 코드*
 - FastMCP 0.1 기반: CLI 로 실행되는 STDIO 방식
- **SSE로 전환**
 - transport="sse"
 - FastMCP 버전을 0.1.x 에서 1.0.x 로
 - 오류 수정, doc string 추가

* <https://github.com/solangii/upbit-mcp-server>

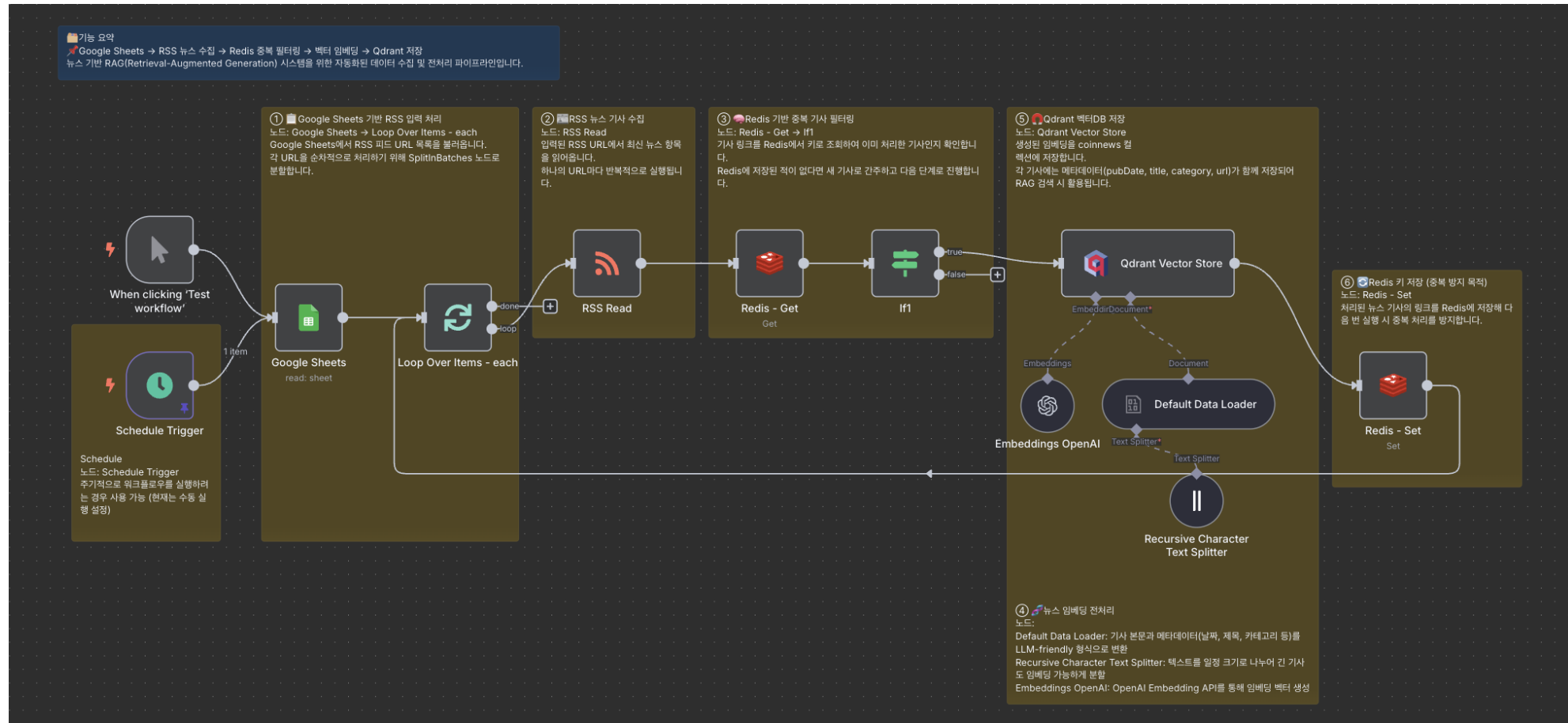
Workflow 예시



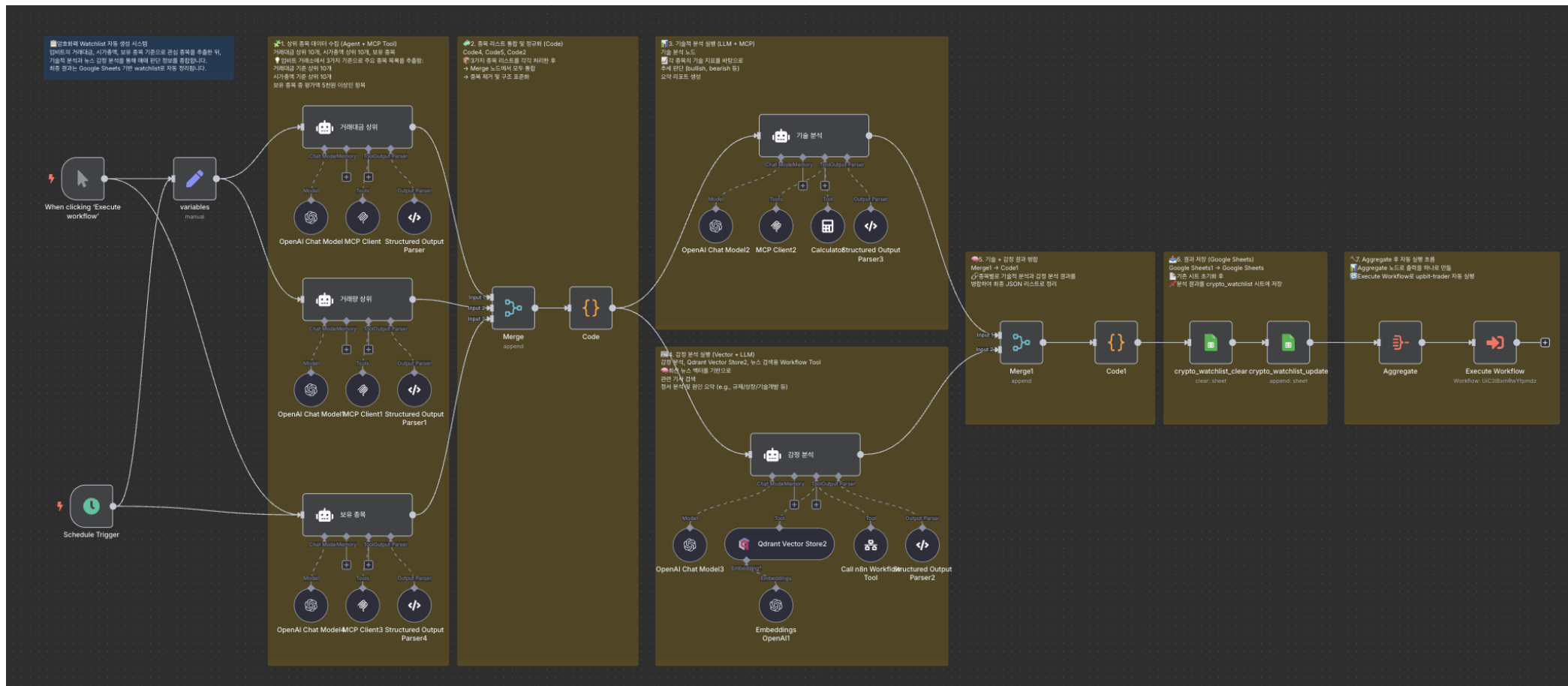
판단 전략



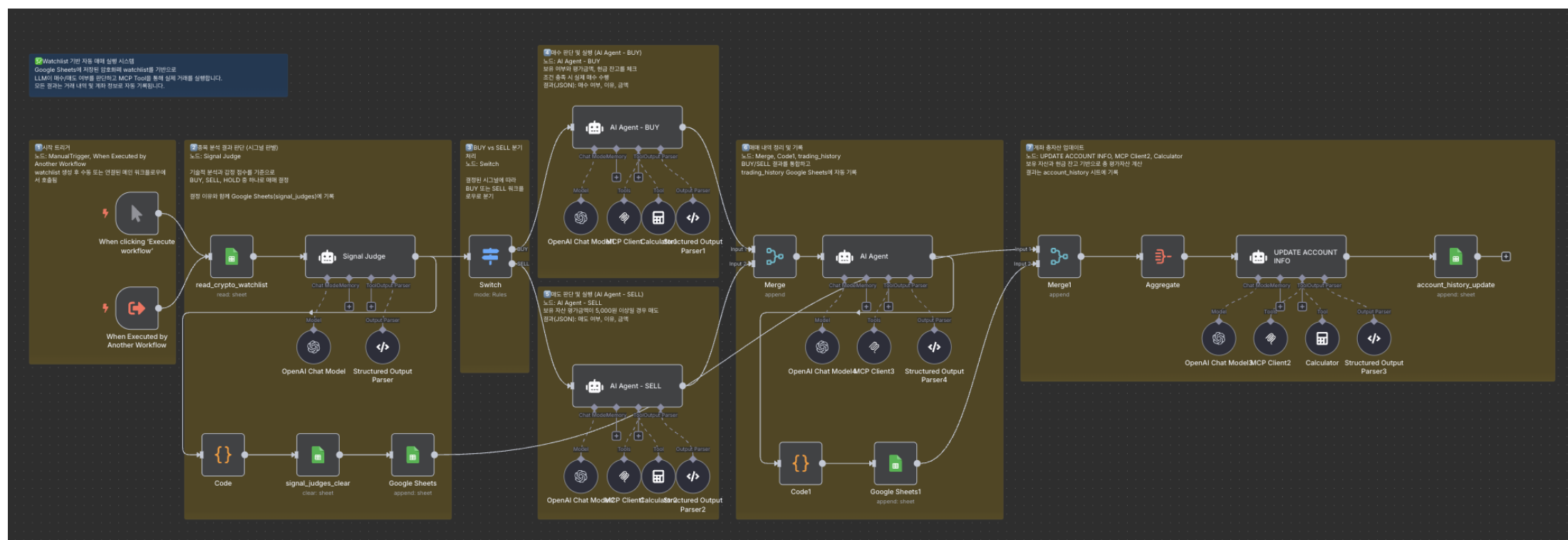
뉴스 수집



종목 선정



거래



마무리

- **n8n + MCP 조합은 거래 자동화를 위한 강력한 프레임워크**
 - Workflow 관리 / 거래 로직 / API 사용을 분리, 구현 및 관리 간편
- 단계별 유닛 테스트의 어려움
- 프롬프트+생성모델 기반 로직의 신뢰성 확인 필요

Quantnode.kr

감사합니다

유첨

Prompts

- What
 - Pre-defined templates for AI interactions
 - 서버가 정의한 재사용 가능한 LLM 프롬프트
 - 예: /gh-issue <id>, /usage-example
- When
 - User explicitly triggers to add context
 - 사용자가 명시적으로 선택하거나 실행하는 순간 사용됨
 - UI에서 슬래시 명령, 버튼 등으로 노출 가능

Resources

- What
 - Data or content exposed by the server
 - LLM이 활용할 수 있도록 서버가 노출하는 파일, DB 레코드, 메모리 등
 - URI 기반(file://, postgres://, screen://)으로 식별됨
- When
 - Application should control content
 - 클라이언트 또는 사용자에게 의해 명시적으로 선택됨
 - 일부 클라이언트는 자동 선택, 일부는 AI가 판단하도록 허용

Sampling

- What
 - LLM 응답 생성을 서버가 클라이언트를 통해 요청
 - 서버 → 클라이언트 → LLM → 결과 반환
 - 인간 승인(Human-in-the-loop) 기반
- When
 - 서버가 LLM 응답이 필요한 경우
 - 에이전트적 판단, 리소스 분석, 멀티스텝 대화 등에 활용

Roots

- What
 - 클라이언트가 서버에 작업 대상 리소스의 경계(URI)를 제안
 - 예시:
 - `file:///home/user/projects/myapp`
 - `https://api.example.com/v1`
- Why
 - Guidance: 서버에게 어떤 리소스를 다뤄야 할지 방향 제시
 - Clarity: 작업 대상 범위를 명확히 구분
 - Organization: 여러 root를 통해 다양한 리소스를 동시에 다룰 수 있음
 - Security: 허용된 범위 내에서만 자원 접근을 유도 → 보안성 향상

Transports

- What
 - MCP 클라이언트 ↔ 서버 간 메시지 전송 메커니즘
 - JSON-RPC 2.0 기반 메시지 포맷 사용
 - Request / Response / Notification
- Built-in Transports
 - Standard Input/Output (stdio)
 - 간단한 로컬 통신에 적합
 - 쉘 스크립트, CLI, 로컬 도구에 유용
 - stdin/stdout 기반의 빠르고 가벼운 통신
 - SSE (Server-Sent Events, **2025-03-26 이후 deprecated**)
 - 서버 → 클라이언트 단방향 스트리밍, HTTP GET 기반
 - 단순 알림용으로 적합하지만 세션/응답 미지원
 - Streamable HTTP
 - HTTP POST 요청+ SSE* 스트림 응답
 - 다중 클라이언트, 세션 유지, 중단 복구 지원
 - Mcp-Session-Id 헤더로 세션 식별 및 유지
 - 중단 시 Last-Event-ID로 재접속 및 메시지 재전송 가능