



econ-radar

뉴스레터를 매일 돌아가는 팀으로 만들었다

경제 뉴스를 매일 **3렌즈**로 정리해 **자동 발행**하고, 그 기록을 **리포트**, **블로그**, **책**으로 키우는 개인 지식 자산 시스템

매일 + 주간 자동 발행 · 블로그, 텔레그램, 이메일 3채널

한 장 요약

- **문제** : 매일 같은 품질로 경제 뉴스를 정리하기. 사람이 직접 하면 피로가 쌓이고 품질이 흔들린다.
- **해법** : 일을 **에이전트 팀과 작업 매뉴얼(스킬)** 로 나눠 매일 자동으로 돌린다.
- **차별점** : 한 번 쓰고 버리지 않는 **누적**. vault(옵시디언)에 쌓아 동향, 블로그, 책으로 키운다.
- **운영** : 매일 18:30 KST에 만들어 블로그, 텔레그램, 이메일로 완전 자동 발행한다. 승인 대신 **사실 검증(fact-check) 게이트**로 거른다.

프롬프트 한 번으로 끝나는 게 아니라, 역할과 매뉴얼과 검수와 누적을 갖춘 **작은 조직으로 설계했다.**

오늘 이야기할 네 가지

- **Part 1 무엇을, 왜** : 한 줄 정의, 실제 발행본, 3층 구조, 3렌즈
- **Part 2 어떻게 만들었나** : 8명의 에이전트 팀, 설계 4원칙, 파일로 둔 기준선, 자동화 3채널, OKF 지식 자산화
- **Part 3 진화** : 20일간 운영과 배포가 설계를 고친 기록, 프로덕션이 가르친 것
- **Part 4 원칙, 교훈, 토의** : 안전선, 배운 점, 로드맵, 데모

Part 1

무엇을, 왜 만들었나

실제 발행본, 매일 저녁에 나간다

kakyungkim.github.io/econ-radar/2026-06-24.html



 **econ-radar 데일리** 2026-06-24

제약·바이오·AI에 무게를 둔 경제 데일리 — 산업구조·투자·시장 3렌즈

TailwindCSS CDN 사용 — 오프라인 환경에서는 스타일이 제한될 수 있습니다.

 바로가기

At a Glance

Top 5

Today's Topic

Business English

Deep Dive

Demand

Investment

Companies

Threads

Sources

3층 구조, 오늘이 쌓여 책이 된다

층	무엇	산출물	자동화
1층 (매일)	수집→분석→편집→검수→검증→렌더	일일 뉴스레터, HTML, 푸시	매일 18:30 자동
2층 (자산)	옵시디언 vault에 누적	태그, 링크, 주제 MOC	큐레이션 자동
3층 (저술)	쌓인 기록에서 패턴 추출	동향 리포트, 블로그, 책	주간 동향 공개 발행

- 1층은 매일 돌아가고, 2층은 그날의 기록을 주제별로 잇고, 3층은 매주 월요일 아침에 한 주를 정리해 공개 발행한다.
- 위 층일수록 사람의 역할이 커진다. 기계는 초안까지 만들고, 최종 판단과 발행은 사람이 하게 한다.

3렌즈, 무엇을 보는가

뉴스를 늘 같은 세 각도로 나누어 본다.

1. **산업구조 (Industry)** : 판을 흔든 플레이어, 밸류체인 변화 지점
2. **투자 (Investment)** : 테마, 종목, 리스크, 기회 (코스피와 해외)
3. **시장 (Market)** : 유망 기업(파는 쪽), 수요(사는 쪽)

전 분야를 다루되 **제약·바이오·AI**에 무게를 둔다. 제약의 수요는 환자, 처방의, 지불자 셋으로 나눈다. 투자와 유망기업은 매수·매도 권유가 아님을 고지하고, 정보와 시나리오, 리스크로 다룬다.

Part 2

어떻게 만들었나: 하네스 구조

1층 파이프라인, 8명의 팀



매일 도는 8명은 가볍게 두고, 깊게 손볼 때만 전문가를 부른다

설계의 핵심 4가지

- **역할 분리** : 수집가는 해석하지 않고, 분석가는 사실만 다루고, 비평가는 표현만 고친다. 책임을 겹치지 않게 설계한다.
- **매뉴얼화(skill)** : 잘하는 법을 사람 머리에 두지 않고 `SKILL.md` 에 적어 둔다. 매일 같은 품질을 유지한다.
- **이중 게이트** : 발행 전에 `style-critic` 은 표현을, `fact-checker` 는 수치와 출처를 한 번씩 본다. 문체 오류와 사실 오류를 따로 거른다.
- **판단만 LLM에** : HTML 렌더처럼 판단이 필요 없는 단계는 **스크립트로 고정한다**. 토큰을 아끼고 양식이 흔들리지 않게 한다.

품질을 어떻게 지키나: 기준선은 전부 파일

- **문체 기준선** : `korean-style-samples.md` (실제 기사 표본)를 꼭 참조한다. "A가 아니라 B", "~인 셈" 같은 AI 상투구는 금지한다.
- **사실 기준선** : `fact-checker` 가 핵심 수치 최대 10개를 원문과 대조한다. 출처 없는 수치는 막고, 단일 소스는 표기하며, 1차 출처(IR·규제기관)를 명시한다.
- **깊이 기준선** : `benchmarks.md` 에 Why it matters 한 줄, 1차 자료 직접 링크해서 양면 시각을 다룬다.
- **수요 렌즈** : `demand-lens.md` 에서 투자자는 공급과 자본으로, Demand는 수요로 본다.
- **양식 기준선** : `NEWSLETTER-FORMAT.md` 계약과 템플릿 렌더로 양식이 흐트러지지 않는다.

기준선을 파일에 둔 게 핵심이다. 에이전트가 매번 같은 잣대를 본다.

자동화: 클라우드 루틴과 3채널 발행

루틴	언제	하는 일
데일리 완전 자동	매일 18:30 KST	수집→분석→편집→검수→검증→렌더→큐레이션→ 블로그·텔레그램·이메일
주간 동향	월요일 08:00 KST	한 주를 엮어 블로그·텔레그램·이메일로 공개 발행(3층)

- 왜 저녁 18:30 : 한국 장 마감 뒤라 그날 국내 시장을 온전히 담는다. 미국 장은 아침 수집이 유리하지만 국내 반영을 앞세웠다.
- 저녁에 동시 발송 : 같은 git push가 블로그, 텔레그램, 이메일(Buttondown) 워크플로를 한꺼번에 깨워 세 채널이 거의 동시에 도착한다.
- 승인 게이트 폐기 : 발행 전 사람 승인을 이틀 만에 접었다. "확인 전까지 미발송"에서 "발송하되 이상하면 취소"로 바꿨다.
- vault 전체 추적 : private git에 통째 담아, 클라우드가 과거를 읽고 여러 주의 흐름을 잇는다.

OKF: LLM이 읽는 지식 표준

마크다운과 링크로 지식을 쌓아 LLM이 읽게 하는 'LLM 위키' 패턴. Google Cloud가 2026년 6월 이걸 개방 규격 OKF(Open Knowledge Format, v0.1)로 정리했고, econ-radar vault가 그 규격을 따른다.

- **파일 경로가 곧 개념 ID.** `topics/AI.md` 는 개념 'AI'. 폴더만 열면 옵시디언 그래프와 백링크가 산다.
- 필수 항목은 **type** 한 줄만 두고, 진입점은 `index.md` 에, 변경 이력은 `log.md` 에 담는다.
- 링크는 **절대경로 마크다운** `[AI] (/topics/AI.md)` . 위키링크와 달리 어디서 열어도 안 깨진다.

SDK도 DB도 없고, 마크다운과 YAML만 쓴다.

한 폴더, 셋이 동시에 읽는다

OKF로 맞추면 vault 폴더 하나가 충돌 없이 세 가지로 동시에 쓰인다.

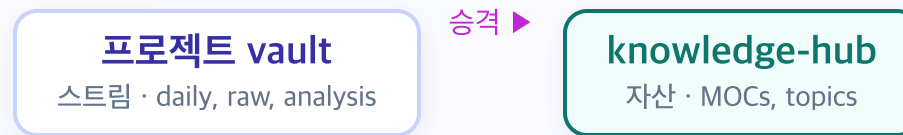
- **옵시디언 vault**: 그래프와 백링크로 탐색
- **LLM 위키**: 에이전트가 `cat` 으로 읽어 질의
- **git 저장소**: `git diff` 로 버전 관리, `clone` 으로 배포

바꾼 건 링크 문법 하나. econ-radar는 6/24에 위키링크 1,228개를 절대경로 마크다운으로 옮겨 한 번에 세 가지를 모두 만족시켰다.

`cat` 으로 읽히면 OKF, `git clone` 으로 받으면 배포.

2층 자산화: 스트림에서 포트폴리오로

- **프로젝트 vault**는 스트림이다. econ-radar의 daily, raw, analysis가 매일 쌓인다.
- 단순히 쌓이기만 하면 자산이 못 되고 노이즈로 남을 수 있다. 그래서 **큐레이터가 오래 쓸 주제만 골라 승격**시킨다.
- 승격한 주제가 모이는 곳이 **knowledge-hub**, 프로젝트를 가로지르는 포트폴리오 위키다. econ-radar와 paper-radar가 같은 OKF 표준으로 한곳에 모인다.



규약은 전역 한 곳, 자산은 허브, 변환은 자동. repo마다 규약을 복사하지 않는다.

Part 3

진화: 운영과 배포가 설계를 고민다

1주차: 5일간의 진화, 운영이 설계를 고친다

- **6/09** 하네스 구축, 첫 발행. 깊이가 부족해 보여 항목 수와 Why it matters를 규칙으로 박았다
- **6/10** 문체가 AI 같아 보여 실제 기사 표본을 들이고 style-critic 게이트를 달았다. 렌즈도 커리어에서 유망기업으로 교체했다
- **6/11** "투자자 관점에 치우쳤다"는 스터디 피드백에 **수요(Demand) 렌즈**를 더했다. 클라우드 루틴도 가동했다
- **6/12** 승인 게이트를 접고 **완전 자동 발행**으로 바꿨다. 텔레그램 채널도 추가했다
- **6/13** 구조를 다시 보고 **4건을 한꺼번에 개선**: 사실 게이트, vault 전체 추적, 렌더 스크립트화, 주간 3층 루틴

어색하다, 치우쳤다 같은 막연한 불만을 그때마다 파일로 된 규칙과 게이트로 바꿨다. 늘 같은 패턴이었다.

6/13 구조 검토, 비판이 곧 백로그

발행 4호 시점에 하네스를 스스로 비판해 보고, 우선순위 4건을 바로 적용했다.

발견한 문제	처방
게이트가 문체뿐, 잘못된 수치가 그대로 나간다	<code>fact-checker</code> 신설 (T4.7)
3층(저술)이 비어 있다, forcing function 부재	주간 동향 루틴 (일 21:00)
클라우드가 과거를 못 본다 (vault 로컬 전용)	레포 private, vault 전체 추적
렌더가 제일 비싼데(93K 토큰) 판단은 불필요	<code>render_html.py</code> 스크립트화

남은 건(MOC 요약 구조, 백로그 트리야지) TODO로 미뤘다. 검토 전문은 `vault/_meta/2026-06-13-harness-review.md`

2주차~3주차: 채널 확장과 3층 가동 (6/14 → 6/29)

- **6/14** 미뤄둔 백로그를 처리했다. MOC 갱신 규칙, 렌즈 범위 분리, 승격 기준을 문서로 정리했다
- **6/15** 텔레그램 직접 발송이 네트워크에 막혀 **GitHub Actions(push 감지)로 발송을 옮겼다**
- **6/16-18** Actions 다중커밋 감지 버그를 고치고 **뉴스 신선도와 중복 방지** 추가. 같은 약과 기업이 며칠씩 반복되던 에이전트 관성을 걷어냈다
- **6/24** 오피디언에서 **OKF 포맷으로 마이그레이션했다**. 위키링크 1,228개를 변환하고 frontmatter와 index/log를 정리했다
- **6/25 이메일 채널을 가동했다**(Buttondown). 텔레그램과 저녁 동시 발송. 첫 자동 발송이 API 헤더 누락으로 막혀 바로 고쳤다
- **6/27** 발표 피드백으로 한국어 문체를 전면 다듬고, 수집 레이어를 A/B/C/D로 실험해 **"보완이 대체보다 낫다"**를 데이터로 확인했다
- **6/28-29 3층을 공개 발행으로 가동했다**. 주간 동향을 3채널로 월요일 아침 자동 발송, 발송 전 CI로 한글 URL 깨짐을 미리 잡았다. on-demand 전문가 4명도 더했다

1주차 패턴이 그대로 이어진다. 운영과 배포가 새 실패 모드를 드러내면, 그게 규칙과 게이트, 채널로 바뀐다.

프로덕션이 가르친 것: 채널을 늘리며 부딪힌 벽

증상 (어디서)	진짜 원인	하네스 교훈
이메일 버튼·글씨가 깨짐	메일은 JS와 외부 CSS가 안 먹고, 클라이언트가 링크색을 덮어씀	로컬에서 된다고 받은편지함에서 되는 게 아니다. 인라인 CSS, table, <code>!important</code>
이메일 첫 자동 발송이 400 거부	Buttondown이 즉시 발송에 <code>X-Buttondown-Live-Dangerously</code> 헤더를 요구	외부 발송 API의 안전장치는 draft가 아니라 실전 발송에서만 드러난다
새 데일리가 옛 포맷으로 회귀	OKF 규약이 <code>~/.claude</code> 전역에만 있어 클라우드가 못 봄	규약과 기준선은 레포 안에 뒤야 유지된다
발음이 엉뚱한 단어	표제어 대신 괄호 속 영어 설명을 읽음	데이터에서 렌더로 가는 계약 을 명시(표제어 = 발음)
"고쳤는데 그대로"	배포 워크플로 지연에 브라우저·CDN 캐시까지	라이브는 curl로 검증하고 캐시를 의심한다

채널마다 런타임이 다르다. 같은 산출물도 **배포 환경에서 다시 검증해야 한다.**

운영 비용: 측정으로 줄이기

1회 발행에 약 43만 토큰, 벽시계로 25분쯤 든다 (`run-metrics.md` 에 단계별 누적)



제일 비싼 단계가 정작 판단은 가장 덜 필요했다. 스크립트로 갈아 이 몫(전체의 약 22%)을 덜어냈다. (애초 목표는 약 40%.)

돈으로 보면, 정액 구독으로 돌려 발행당 추가 비용은 사실상 0원. 토큰은 청구서가 아니라 시간·병목의 지표로 읽는다.

Part 4

원칙, 교훈, 토의

안전선: 하지 않는 것

- **자동 발행은 정기 발행물에만** : 데일리와 주간 동향은 블로그, 텔레그램, 이메일 3채널로 자동 나간다. 저술(블로그 글, 책)은 여전히 사람이 승인해야 발행된다.
- **매수·매도 권유 금지** : 투자와 유망기업은 정보, 시나리오, 리스크로만 다룬다.
- **출처 없는 수치 생성 금지** : fact-checker가 출처 없는 수치를 발행 전에 막는다. 추정은 "추정"이라 밝힌다.

자동화의 신뢰는 무엇을 자동으로 하느냐보다 무엇을 자동으로 안 하느냐에서 온다.

하네스로 배운 점 (스터디 토의용)

- **일을 조직으로 본다** : 프롬프트 한 방으로 끝내지 않고, 역할과 매뉴얼, 검수, 누적을 갖춘 작은 팀으로 운영한다.
- **지식은 파일에 둔다** : 기준선과 표본, 계약을 파일에 두면 품질이 사람에게 묶이지 않는다.
- **게이트는 실패 모드별로** : 문체 게이트는 어색함을, 사실 게이트는 오류를 막는다. 자동 발행일수록 사실 게이트가 먼저다.
- **판단과 기계 작업을 가른다** : 판단 없는 단계(렌더)는 스크립트로 고정하고, 판단하는 단계만 LLM에 맡긴다.
- **운영이 설계를 고친다** : 20일간 거의 매일 구조가 바뀌었다. 모두 운영하다 나온 불만에서 출발했다.
- **배달 환경마다 다시 검증한다** : 로컬 성공이 전부가 아니다. 이메일, 클라우드, 캐시는 저마다 다른 런타임이다.

→ 토의: 여러분의 반복 업무에서 "판단이 필요한 단계"와 "기계로 굳힐 단계"는 어디서 갈릴까요?

솔직한 미완: 다음 백로그

- **OKF 지속성** : 마이그레이션은 했지만 클라우드 에이전트가 아직 옛 포맷으로 만든다. 규약을 **레포 안으로** 옮겨야 자리잡는다(지금은 매일 클린업이 필요).
- **발송 실패 알림** : 한 채널이 조용히 실패해도(이번 이메일 400처럼) 자동으로 알 길이 없다. 워크플로 실패를 텔레그램이나 이슈로 알리는 일을 다음 과제로 둔다.
- **중복 규칙 검증** : "한 사건은 한 번만 서술" 규칙을 넣었으니, 실제 발행본에서 중복감이 줄었는지 며칠 더 본다.

하네스는 완성으로 끝나지 않고 **백로그가 계속 돌게 한다**. 스스로 비판한 점이 그대로 다음 할 일이 된다.

직접 보기: 아카이브와 설계 노트

kakyungkim.github.io/econ-radar/



econ-radar

아카이브

제약·바이오·AI에 무게를 둔 경제 데일리 — 산업구조·투자·유망 기업 3렌즈
매일 신뢰 경제지를 스캔해 정리합니다. 아직 실험 단계예요.

발행본

2026-06-12

mRNA 암백신 상업화 문턱 · K-바이오 L/O 2.9조 · 코스피 8,000 돌파

2026-06-11

미 CPI·ECB 동시 분수령 · 오라클이 드러낸 AI 인프라 청구서

.../econ-radar-agent-harness/ (설계 노트 블로그)

Ka-Kyung Kim

Bioinformatics · Data Science · Life Science

Topics About 한국어 EN

Ka-Kyung Kim

Bioinformatics specialist — NGS pipelines, clinical data, genomics.

Seoul, Korea

Email

GitHub

LinkedIn

경제 뉴스 읽기를 포기하고 AI 에이전트 팀을 만들었다 — econ-radar 설계 노트

June 10, 2026 · 3 minute read

Read this post in English

안녕하세요. 임상 데이터, 유전체학, 머신러닝을 넘나들며 일하는 바이오인포매틱스 연구자 김가경입니다.

하지만 오래가지 못했습니다. 며칠은 열심히 읽다가도 어느 순간 흐름이 끊기곤 했습니다. 의지가 부족해서라기보다 정보량이 너무 많았습니다. 읽어야 할 기사가 너무 많고, 하나씩 정리하다 보면 시간이 훌쩍 지나곤 했습니다.

제가 포기한 건 경제 뉴스를 읽는 일이 아니라, 매일 직접 수집하고 정리하는 방식이었습니다. 그래서 문득 이런 생각이 들었습니다. “시를 어떻게 쓸까”가 아니라 “이 반복 작업을 시스템으로 만들 수 있을까”로요. 뉴스를 모으고 정리하는 일은 역할을 나눈 에이전트 팀에 맡기고, 저는 판단과 검수에만 집중하기로 했습니다. 그렇게 해서 지금의 econ-radar가 만들어졌습니다.

econ-radar란?

econ-radar는 매일 주요 경제지를 스캔해 산업 구조, 투자 흐름, 시장이라는 세 가지 관점으로 정리하는 개인 경제 브리핑입니다.

On this page

econ-radar란?

하나의 AI가 아니라 적은 팀

만들면서 배운 점

1. “어색하다”는 피드백은 규칙이 되어야 한다

2. 사람이 할 일을 줄이는 게 목표다

3. 사실과 해석은 분리하는 편이 낫다

4. 읽고 끝나는 정보보다 쌓이는 정보

5. 누가 파는지는 봤지만, 누가 사는지는 보지 못했다

출발점

직접 보기

다음

▲ 클릭하면 각 페이지로 이동

econ-radar · 뉴스레터 하네스

26

감사합니다

econ-radar : 매일 돌아가는 3렌즈 뉴스 팀, 이중 게이트, 3채널 발행, 누적되는 지식 자산



QR은 econ-radar 아카이브. 발행 채널은 블로그, 텔레그램 @econradar, 이메일 셋. 데모는 오늘자 HTML 리포트.