

MAXIDOGE 최종 구조 제안서 (User Flow + UIUX + Front/Back)

- 작성일: 2026-02-23
- 기준: 제공 파일 30개 분석 (중복 제거 후 25개 기준)

1) 최종 제품 구조 (Canonical)

MAXIDOGE는 아래 4개 축으로 통합한다.

1. Terminal: 전략 지휘, 상황판, 시그널 해석, 실행 진입점
2. Arena: 5단계 실전 시뮬레이션/리그, Entry Score 기반 의사결정 훈련
3. Signals: Scanner + Alert Ladder + Explainability
4. Passport: 유저 행동/성과/학습 이력, 개인화 추천, 재방문 루프

역할 정의는 다음으로 고정한다.

- GUARDIAN: 리스크/가드레일/정합성 검증 서비스
- COMMANDER: 전략 오케스트레이션 및 추천 엔진
- SCANNER: 시장 감시/이상징후/알림 생성 엔진

참고: 위 3개는 “캐릭터형 에이전트”가 아니라 “서비스 레이어 역할”로 설계한다.

2) 최적 유저 플로우

A. Onboarding (첫 진입)

1. 목표 선택: `빠른 수익` / `리스크 최소화` / `학습 중심`
2. 성향 진단: 공격형-중립-보수형 (3문항)
3. 튜토리얼: 90초 인터랙션 (Terminal -> Signal -> Arena 1회)
4. 초기 모드 결정: Recommended Track 자동 제안
5. 첫 성공 행동 유도: 관심자산 3개 등록 + 알림 1개 활성화

B. Activation Loop (매일 사용 루프)

1. Today Brief 확인 (핵심 3줄)
2. Scanner 신호 확인 (중요도 필터)
3. Explain 카드로 이유 확인
4. Action 선택: 관망 / 소액 진입 / Arena 연습
5. 결과 기록 -> Passport 점수 반영

C. Conversion Loop (학습 -> 실전)

1. Arena 5-Phase 완료
2. Entry Score 임계치 달성

3. 실전 전환 제안 (권장 사이즈/손절 기준 포함)
4. 실전 실행 후 회고 카드 자동 생성

D. Retention Loop (재방문)

1. 성과 요약 알림 (일/주)
2. 놓친 기회 복기 카드
3. 다음 목표 1개 자동 제안

3) Arena 구조 고정안

Arena는 5단계로 고정한다.

1. Scan
2. Validate
3. Plan
4. Execute
5. Review

각 단계에서 필수 입력/출력을 강제한다.

- 입력: 시장상태, 신호강도, 변동성, 리스크예산
- 출력: 행동결정, 근거, 손절/익절, 회고 태그

4) Alert/Anomaly 구조

알림 계층은 아래로 통일한다.

- LOW: 참고
- HIGH: 관찰 강화
- CRITICAL: 행동 후보
- P0: 즉시 확인 필요

모든 알림은 Explainability를 포함한다.

- 왜 발생했는지 (근거)
- 무엇을 해야 하는지 (권장 행동)
- 리스크는 무엇인지 (주의 포인트)

5) UI/UX 최종 설계

정보구조(IA)

1. Home (Today Brief)
2. Terminal (전략/신호/추천)
3. Arena (5-Phase 훈련)

4. Signals (탐지/알림/로그)
5. Passport (기록/성장/개인화)
6. Settings (알림/연동/위험설정)

핵심 UX 원칙

1. Action First: 읽는 화면보다 행동 버튼을 우선 배치
2. Explain Always: 모든 추천/알림에 이유 표시
3. Risk Visible: 손실 가능성을 항상 같은 위치에 노출
4. Progressive Disclosure: 초보는 단순, 고급은 확장
5. One-screen Decision: 핵심 의사결정은 1화면 내 완료

반응형 전략

- Desktop: 다중 패널 (Brief + Signal + Position)
- Mobile: 카드 스택 + Bottom Tab + Sticky Action CTA

6) 프론트엔드 아키텍처 (Target)

- App Shell + Feature Module 구조
- 공통 상태 계층: Session / Portfolio / Signals / Arena / UI
- UI 계층 분리: Presentational vs Container
- 서버 상태 캐시: 조회/변경 분리 (Query/Mutation)
- 에러 UX: Inline Error + Retry + Fallback Data

권장 폴더 예시:

- `src/features/terminal/*`
- `src/features/arena/*`
- `src/features/signals/*`
- `src/features/passport/*`
- `src/shared/ui/*`
- `src/shared/api/*`
- `src/shared/state/*`

7) 백엔드 아키텍처 (Target)

도메인 기반 모듈로 분리한다.

1. auth-service
2. user-profile-service
3. signal-service (scanner/anomaly)
4. strategy-service (commander)
5. risk-service (guardian)

- 6. arena-service
- 7. notification-service
- 8. analytics-service

이벤트 흐름 권장:

- `MarketEvent` -> `SignalDetected` -> `RiskChecked` -> `ActionSuggested` -> `UserActionLogged`

DB 원칙:

- OLTP: 사용자 상태/설정/트랜잭션
- OLAP: 성과/행동 로그/모델 평가
- 감사로그(Audit): 전략 변경, 리스크 룰 변경

8) API 계약 표준화

모든 API 응답 포맷 통일:

```
{
  "success": true,
  "data": {},
  "error": null,
  "meta": {
    "requestId": "...",
    "timestamp": "..."
  }
}
```

표준 에러 코드 세트:

- `AUTH_401`
- `AUTHZ_403`
- `RATE_429`
- `VALIDATION_400`
- `RISK_BLOCKED_460`
- `INTERNAL_500`

9) 데이터/학습 구조

학습 루프:

1. 유저 행동 수집
2. 결과 라벨링 (성공/실패/회복)
3. 전략 피드백 생성
4. 개인화 추천 반영
5. 다음 세션 실험(A/B)

RAG/지식 레이어:

- 문서 버전 관리
- 근거 출처 링크
- 오래된 규칙 자동 만료 정책

10) 실행 우선순위 (Roadmap)

1. IA 고정 + 라우팅 정리
2. Arena 5-Phase 단일 규칙 적용
3. Signal/Alert Ladder 통일
4. API 응답 스키마 전면 통일
5. Passport 지표 MVP 릴리즈
6. 개인화 추천/학습 루프 고도화

결론

최종 구조는 `Terminal(지휘) + Arena(훈련) + Signals(탐지) + Passport(성장)`의 4축 통합형으로 고정한다.
유저 플로우를 `오늘 브리프 -> 신호 해석 -> 행동 -> 회고`의 반복 루프를 중심으로 설계하며, 프론트/백엔드 모두 도메인 모듈화와 API 표준화를 우선 적용한다.