**제안: 상세 마일스톤(티켓화)**

**M0 — 프로젝트 위생(코드 없이 준비 단계)**

* **M0.1 Repo/브랜치/디렉토리 규칙**: ftm2/ 트리, ftm2-dev 브랜치, # [ANCHOR:...] 주석 규칙.
  + DoD: 규칙 문서화, README 반영.
* **M0.2 ENV 체인/보안 규칙**: token.env 분리, .gitignore, load\_env\_chain() 기대 키 목록.
  + DoD: .env.sample/token.env.sample 스펙 문서화.
* **M0.3 patch.txt 운영 규칙**: **기능추가만 기록**, 버그픽스 제외, semver, 예시 포맷/훅.
  + DoD: patch.txt 템플릿 + patches 테이블 매핑 문서.
* **M0.4 Discord 채널/권한/명령 네이밍**: 슬래시 커맨드 목록·권한 표.
  + DoD: 명령 표/ID 매핑표.

**M1 — 골격 & 테스트넷 라운드트립(명세)**

* **M1.1 Orchestrator & State Bus**
  + 앵커: [ANCHOR:ORCH], 파일: app.py, core/state.py
  + 입/출력: tick 이벤트→버스→소비자(신호/리스크/실행)
  + DoD: 이벤트 타입·에러 코드·쿨다운 정책 명세서.
* **M1.2 Binance Connector 스펙(REST/WS, testnet↔live 토글)**
  + 앵커: [ANCHOR:BINANCE\_CLIENT], 파일: exchange/binance.py
  + 요구 채널: kline(1m,5m,15m,1h,4h), markPrice, user/account
  + DoD: 모든 엔드포인트 목록/쿼타/리트라이 정책 표.
* **M1.3 Market Streams & Snapshot 규격**
  + 앵커: [ANCHOR:STREAMS], 파일: data/streams.py
  + 스냅샷 스키마: price, mark, kline\_map, positions, balances
  + DoD: 스냅샷 필드/단위/타임스탬프 동기화 규칙.
* **M1.4 Persistence 스키마 & 저장 규칙**
  + 앵커: [ANCHOR:PERSISTENCE], 파일: core/persistence.py
  + 테이블: trades, positions, pnl\_daily, config, events, patches
  + DoD: DDL 명세/인덱스/무결성/마이그레이션 전략.
* **M1.5 Discord Agent(패널/대시보드/알림) 메시지 규격**
  + 앵커: [ANCHOR:DISCORD\_UI], 파일: discord\_bot/\*
  + 명령: /mode /auto /close /reverse /flat + 슬라이더·토글
  + DoD: 각 메시지 템플릿의 필드·편집(수정) 타이밍 정의.
* **M1.6 “더미 포캐스트→스카웃” 드라이런 시나리오**
  + 체결 없이 시뮬 주문 로그만 발생(주문 라우트는 문서화)
  + DoD: 시나리오 절차서 + 기대 로그 표본.

**M2 — 레짐·포캐스트·리스크(명세)**

* **M2.1 Feature Pipeline**
  + 앵커: [ANCHOR:FEATURES], 파일: data/features.py
  + 입력/출력 컬럼표, NaN 처리, 퍼센타일 캐시 규칙
  + DoD: 피처 목록·윈도·복잡도/지연 성능 목표.
* **M2.2 Regime Classifier**
  + 앵커: [ANCHOR:REGIME], 파일: signal/regime.py
  + 기준: EMA 스프레드, RV 백분위, 히스테리시스
  + DoD: 레짐 전이표·임계·테스트 케이스.
* **M2.3 Forecast Ensemble & Scoring**
  + 앵커: [ANCHOR:FORECAST][ANCHOR:SCORING]
  + P(up\_k)/MR\_edge/Cross\_edge 계산식, 온라인 가중치 업데이트 λ
  + DoD: 레짐별 θ 테이블/성능 로깅 포맷.
* **M2.4 Risk Engine(ATR-unit, 상관캡, 데일리컷)**
  + 앵커: [ANCHOR:RISK], 파일: trade/risk.py
  + 수식/단위/라운딩/최소수량·레버리지 캡 명세
  + DoD: 사이즈 계산 예제표 + 엣지 케이스.

**M3 — 실행/보호/하이드레이션(명세)**

* **M3.1 Gatekeeper**: AUTO, idem\_hit, day\_loss\_guard, cooldown
  + 앵커: [ANCHOR:GATEKEEPER], 파일: trade/gatekeeper.py
  + DoD: 게이트 통과/차단 사유 코드 매핑.
* **M3.2 Order Router**: link\_id, 재시도, 에러 맵, 슬리피지 모델
  + 앵커: [ANCHOR:ORDER\_ROUTER], 파일: trade/execution.py
  + DoD: 주문 상태 머신 다이어그램.
* **M3.3 스케일/부분청산/BE/트레일 규칙**
  + 앵커: [ANCHOR:PROTECTIONS]
  + 트리거 우선순위·상호배제·MFE/MAE 조건
  + DoD: 단계별 기대 로그/체결 후 상태표.
* **M3.4 Position Hydration & 미체결 복원**
  + 앵커: [ANCHOR:POSITIONS], 파일: trade/positions.py
  + DoD: 재부팅 시나리오 절차서 + 검증 체크리스트.

**M4 — 패널/대시보드 마감 & 리포팅**

* **M4.1 KPI 카드/편집 타이밍/스팸 억제** (편집만, 새 메시지 최소화)
* **M4.2 /patch log 노출 & DB-파일 동기화 규칙**

**M5 — 테스트 하네스 & 리플레이**

* **M5.1 24h 리플레이 러너 명세(E2E)**
* **M5.2 유닛 테스트 명세(피처/리스크/게이트)**

**M6 — 라이브 준비 점검표**

* 2단 ARM\_LIVE 플로우, 위험 시나리오 드릴, 롤백 절차.

**티켓 템플릿(이 서식으로 “코드 주세요” 요청하면 완성도↑)**

* **제목**: [M1.2] Binance Connector 명세 구현
* **앵커/파일**: [ANCHOR:BINANCE\_CLIENT] in exchange/binance.py
* **목표/스코프**: …
* **입력/출력 계약**: 함수 시그니처/딕트 스키마(필드/단위/예시 값)
* **ENV 토글**: MODE, BINANCE\_\*\_API\_KEY/SECRET
* **로그/에러 코드**: 예상 코드표
* **DoD**: 수용 기준(테스트/로그 표본/성능 수치)
* **patch.txt**: feat(connector): testnet/live 토글 가능한 REST/WS 스텁 추가

**patch.txt 운영 원칙(재확인)**

* **기능 추가만 기록(버그픽스 제외)**, semver, 날짜·버전·요약 3줄 규격.
* **DB patches 테이블**과 동기(뷰 명령: /patch log).
* 예시:
* 2025-09-05 v0.1.1
* - feat(state): 전역 스냅샷 버스 규격 확정(가격/마크/포지션)

- feat(discord): /mode /auto /close 명령 스펙 확정(메시지 포맷)