

웨더마켓 V3 이관 갱신 2026-09

웨더마켓 프로토콜 백서

기술 백서 · v3 (2026-09)

날씨 데이터가 아니라 판정을 파는 합의 날씨 오라클입니다. 어떤 조건이 어느 시각에 성립했는지를 11개 소스에 고정된 규칙을 적용해 결정하고, 매시간 Flare 에 앵커해 어떤 계약이든 읽을 수 있는 판정으로 게시하며, XRP Ledger 의 XRP 로 결제하고, 사람과 자율 에이전트가 사람의 개입 없이 구매합니다.

초록

날씨는 시장이 직접 정산 기준으로 삼는 몇 안 되는 실물 지표입니다. 농작물 보험, 에너지 헤지, 행사 취소 보험, 해상운송·건설 지연 조항이 그렇습니다. 그런데 이런 계약이 참조하는 데이터는 거의 언제나 단일 공급자의 숫자 하나입니다. 그 공급자가 틀렸거나, 멈췄거나, 악의적이라면 정산도 틀리고 사후에 이를 증명할 방법도 없습니다.

Weather Data Market 은 가격 오라클이 자산 가격을 다루듯 날씨를 다룹니다. 매시간 11개 기상 소스 피드에서 같은 측정값을 수집하고, 신뢰가 아니라 고정된 규칙으로 이상치를 배제해 합의값을 만들고, 독립 관측과 대조한 뒤, 실행 전체의 해시를 Flare 메인넷의 SnapshotAnchor 컨트랙트에 씁니다. 실행마다 트랜잭션 하나이고 덮어쓰지 않으므로 첫 블록 시각이 곧 기록입니다. 2026년 9월 23일까지는 같은 해시의 1세대 사본을 BNB 체인에도 남겼고, 그 앵커들은 계속 검증할 수 있습니다. 파는 것은 그다음의 판정입니다. 명시된 조건이 그 시각에 성립했는가 - 충족, 불충족, 보류 - 를 이유와 함께 Flare 의 레지스트리에 게시하고, 그 레지스트리는 근거 실행이 앵커된 판정만 settlement-grade 로 표시합니다. 규칙의 매개변수는 비공개이지만, 어느 상대방이든 규칙이 바뀌지 않았고 실행이 분쟁 이전에 해시로 존재했음을 확인할 수 있습니다.

1. 문제

정산 오라클은 한 가지 질문에 설득력 있게 답해야 합니다. 계약에서 진 쪽이 이 숫자를 왜 받아들여야 하는가? 단일 소스 날씨 피드는 세 가지 지점에서 이 시험을 통과하지 못합니다.

검증 불가능성. API는 값을 돌려줄 뿐입니다. 그 응답이 정산 시점에 실제로 제공된 값이라는 것도, 이후에 수정되지 않았다는 것도 증명하지 못합니다.

단일 장애점. 기상 모델은 실제로 서로 다른 값을 냅니다. 관측소 장애, 수집 버그, 격자 보간 오류 하나로 특정 공급자만 몇 도씩 어긋날 수 있습니다. 그 공급자를 참조하는 계약은 그대로 잘못 정산됩니다.

사람 중심의 접근 구조. 날씨 API는 영업 상담과 월 청구서, 대시보드를 통해 팔립니다. 특정 도시의 내일 강수 판정이 필요한 자율 에이전트는 새벽 3시에 사람이 서명해주지 않으면 그 데이터를 살 수 없습니다.

2. 시스템 개요

```

11 source feeds (3 direct, 8 via Open-Meteo)
  | hourly collection (staged to respect rate limits)
  ▼
median + MAD outlier rejection          ← statistical, not reputational
  |
  ├── consensus index (T, H, W, P) + WDI / TDI
  ├── corroboration verdict vs the observation class
  |
  ▼
canonical JSON → SHA-256 → anchored every run
  |                                     ├── Flare: SnapshotAnchor.anchor(sha256, date, runId)
  |                                     |   anchor of record - block timestamp, never
  |
overwritten
  |                                     ├── BNB Chain: IndexAnchored event (first generation)
  |                                     └── (legacy, runs before 2026-09-23: XRP Ledger memo,
  |                                         provable on Flare via FDC XRPPayment)
  |
  ▼
verdicts → Flare: DeterminationRegistry (met / not met / withheld)
  |       └── settlement-grade when SnapshotAnchor.isAnchored(sha256) is true
  |
  ▼
settlement layer: XRP (XRP Ledger, priced by FTS0v2 XRP/USD) - USDT on BNB retired 2026-
09-23
  └── humans via wallet · agents via REST + prepaid 402 metering
  
```

3. 데이터 수집

167개 도시에 대해 매시간 11개 소스를 수집합니다. 소스 구성은 의도적으로 이질적입니다. 각자 독자적인 물리 모델을 운영하는 국가 기상청들, 직접 관측망, 그리고 벤더 합성값 하나입니다.

서로 상관된 소스만 모으면 의미가 없습니다. 이상치 제거는 소스들이 독립적으로 어긋날 수 있을 때만 작동하기 때문입니다.

그 밑에는 분명히 밝혀 둘 원칙이 있습니다. 도시의 대표값은 관측소 한 곳의 값이 아닙니다. 관측소마다 미기후·설치 환경·고도가 다르고, 수 킬로미터 밖 공항은 도심이 아닙니다. 그래서 방어 가능한 도시 지수는 여러 관측 계통 - 해당 도시의 공식 관측소, ICAO METAR, GTS 전문 - 에 여러 국가 기상청의 수치예보 모델을 결합하고, 공급자가 지원한다면 AI 기법과 토지 피복 같은 지형 정보까지 함께 고려해 만들어집니다. 케이웨더가 도시별 지수를 이 방식으로 생산하기 때문에, 그 값은 여기서 또 하나의 모델이 아니라 관측 계열 소스로 들어옵니다. 따라서 이 정확도는 고정된 상한이 아닙니다. 관측망이 조밀해질수록 같은 방법이 더 정확한 지수를 내놓습니다.

예보에도 같은 원칙을 적용합니다. 실황은 11개 소스로 합의하면서 예보만 단일 공급자에서 받는다면, 상품의 절반이 나머지 절반이 피하러 만든 바로 그 가정 위에 놓이게 됩니다. 그래서 예보도 독립된 두 공급자(Open-Meteo, 케이웨더)에서 받아 병합합니다. 겹치는 날짜는 평균하고 모든 행에 기여한 소스 수를 남겨, 단일 소스 날짜가 교차 확인된 날짜처럼 보이지 않게 합니다.

| 소스 | 종류 | 비고 |

| --- | --- | --- |

| METAR (NOAA AWC) | **observation** | 공항 지상 관측 - 모델이 아닌 실측값. ICAO 관측소가 카탈로그의 모든 도시에 매핑돼 있습니다. 합의값을 대조하는 관측 계열 두 구성원 중 하나이며 (다른 하나인 케이웨더 SYNOP 피드는 다른 관측망을 씁니다), 도시를 단독으로 정의하지 않습니다. |

| NOAA GFS | **model** | 미국 |

| DWD ICON | **model** | 독일 |

| JMA | **model** | 일본 |

| Météo-France ARPEGE/AROME | **model** | 프랑스 |

| ECMWF IFS | **model** | 유럽 (정부간 기구) |

| UK Met Office | **model** | 영국 |

| ECCC GEM | **model** | 캐나다 |

| MET Norway | **observation** | 노르웨이 기상청 - 전지구 수치예보 모델(locationforecast). 자체 관측망이 덮는 곳은 노르웨이·북유럽이며, 여기 실린 다른 도시가 아닙니다. |

| Open-Meteo best_match | **synthetic** | 벤더 자체 합성값 - 진실이 아니라 하나의 의견으로 취급합니다 |

| KWeather | **observation** | 한국. 공식 관측소·ICAO METAR·GTS 전문에 ECMWF/GFS/JMA/KIM 모델과 AI 기법·지형 정보를 종합해 도시별 지수를 생산하며, 자체 미세먼지 센서망 운영과 두 번째 예보 소스도 제공합니다. 해외 도시에서는 그 나라 WMO 종관관측소가 켜 GTS SYNOP 전문을 케이웨더가 재제공합니다 - 기상청이 관측하는 곳은 대한민국뿐입니다 |

3.1 설계 제약으로서의 호출 한도

167개 도시 × 11개 소스를 매시간 조회하면 상위 애그리게이터의 분당 한도를 넘습니다. 그렇다고 참여 모델을 돌려가며 쓰면(로테이션) 합의가 조용히 나빠집니다. 그 시간에 빠진 소스는 해당 시각 중앙값에 아무 기여도 하지 못하기 때문입니다.

그래서 미리 수집합니다. 5분 주기 크론이 가장 오래된 국가 모델 2종을 스테이징 테이블에 갱신하므로, 7종이 약 18분마다 한 바퀴를 돕니다. 정시 합의 런은 75분 이내의 스테이징 관측을 실시간 소스와 합칩니다. 모든 소스가 매 런에 참여하고, 어느 1분에도 배치가 2개를 넘지 않습니다. 고정된 시간 슬롯이 아니라 오래된 순으로 고르는 것이 핵심입니다. 서버리스 크론은 실행 시각이 흔들리므로, 슬롯을 고정하면 호출 한 번을 놓칠 때 그 모델이 한 시간 통째로 빠집니다.

4. 합의

각 지표는 독립적으로 축약됩니다. 추정량은 중앙값이며, 이상치는 중앙절대편차(MAD)로 제거합니다. 이는 표준적인 강건 추정 구성으로, 붕괴점이 보장된다는 이유로 선택했습니다. 소스의 절반까지 임의로 틀려도 결과는 움직이지 않습니다.

```
// per-metric consensus across the collected source feeds
center    = robust center of the values           // not a mean - one feed cannot drag it
spread    = robust dispersion around that center
threshold = f(spread, floor_metric)               // parameters: proprietary, fixed per
version
outliers  = { v : |v - center| > threshold }      // rejected, recorded by name
index     = robust center of the remaining values

// floor_metric keeps unanimous rounding differences from reading as outliers.
```

```
// Every run's output is hashed and anchored on-chain; the rule does not change
// between runs without a versioned notice.
```

제외된 소스는 조용히 버려지지 않습니다. 매 실행이 어느 소스가 얼마나 어느 방향으로 벗어났는지 기록하고, 구독자는 대시보드와 소스별 피드에서 그것을 봅니다. 상습적으로 제외되는 소스는 모든 유료 상대방에게 보이며, 이것이 단일 제공자 피드가 줄 수 없는 책임성입니다. 제외를 정하는 규칙은 판마다 고정되고 그 매개변수는 공개하지 않습니다.

4.1 실측 뒷받침 판정

모델끼리의 합의는 현실과의 일치가 아니므로, 각 런을 실측과 다시 대조합니다. 무엇과 대조하는지가 중요합니다. 관측소 한 곳은 도시가 아닙니다. 공항 METAR는 도심에서 수 킬로미터 떨어져 미기후도 다르므로, 그 한 지점을 진실로 놓으면 도시가 아니라 그 관측소를 재게 됩니다. 그래서 대조 대상은 관측 계열 전체입니다 - 공항 METAR, 그리고 도시 공식 관측소와 GTS 전문을 재제공하는 케이웨더 - 각각을 고정된 허용 오차 안에서(그 수치는 내부 규칙의 일부로 공개하지 않습니다) 합의값과 비교합니다.

부합하는 신선한 관측이 어긋나는 것보다 많고 부합이 2곳 이상이면 verified, 한 곳이 부합하고 반대가 없으면 single_source(뒷받침은 실재하나 모름), 찬반이 같으면 mixed(통과가 아니라 미결), 관측은 있으나 대체로 어긋나면 mismatch, 읽을 수 있는 신선한 관측 계열 소스가 없으면 unverified입니다. 부합·불일치·기권 관측 수가 판정과 함께 나가므로, 정산 측은 유리한 라벨만이 아니라 명시된 뒷받침의 깊이를 요구할 수 있습니다. 기준이 고정된 관측소가 아니라 계열이므로, 관측망을 추가하면 규칙을 바꾸지 않고도 판정이 두터워집니다 - 정확도는 관측소 밀도와 함께 자랍니다.

5. 지수

원지표와 함께 두 개의 지수를 발행합니다. 정산 계약은 재량 판단이 아니라 정의가 고정된 단일 스칼라를 원하기 때문입니다.

```
// deterministic indices over the consensus values
WDI = weighted composite of normalised T, H, W, P // weights and ranges: proprietary,
fixed per version (wdi_v1)
TDI = normalised temperature // fixed range, fixed per version
(tdi_v1)

// Same inputs, same number, every time. A change to any coefficient or range
// ships as a new version; contracts written against a version stay computable.
```

WDI(Weather Data Index)는 복합 지수이고, TDI(Temperature Data Index)는 더위만 보는 계약을 위해 기온 성분을 분리한 것입니다. 둘 다 합의값의 순수 함수이며 가중치와 범위는 고정돼 도시나 계절에 따라 바뀌지 않습니다. 계수는 비공개 규칙의 일부입니다. 각 지수는 버전을 지니며, 계수가 하나라도 바뀌면 새 버전으로 나가므로 옛 버전으로 쓴 계약은 계속 계산할 수 있습니다.

6. 검증 가능성

세 개의 독립적인 계층이 결과를 검증 가능하게 만듭니다. 각각은 서로 다른 질문에 답합니다.

6.1 정규 직렬화와 앵커링

매 실행은 정규 JSON 바이트열로 직렬화되어 해시됩니다. 다운로드 엔드포인트가 정확히 그 바이트를 돌려주므로 구매자가 로컬에서 계산한 해시가 서버가 게시한 해시와 같습니다. 정본 앵커는 Flare 메인넷에 있습니다. 퍼블리셔 키가 실행마다 SnapshotAnchor.anchor(sha256, date, runId) 를 한 번 부르고, 컨트랙트가 블록 번호와 시각을 저장하며, 이미 있는 해시는 덮어쓰기를 거부합니다 - 첫 시각이 증거입니다. DeterminationRegistry 가 같은 컨트랙트의 isAnchored(sha256) 를 읽으므로, 실행이 앵커되는 순간 판정이 settlement-grade 가 됩니다. 2026년 9월 23일까지는 1세대 사본을 BNB 체인에 IndexAnchored 이벤트로도 남겼고, 그 1,452건은 계속 검증할 수 있습니다. 2026년 9월 23일 이전의 실행은 XRP Ledger 1 drop 결제의 메모에 남긴 레거시 앵커도 갖고 있으며, XrplAnchorVerifier 와 FDC XRPPayment 증명으로 Flare 에서 증명할 수 있습니다. Flare 에 직접 쓰는 것이 증명 한 건 대신 트랜잭션 한 건이 되면서 그 경로는 신규 실행에 대해 폐기했습니다. 어느 앵커든 증명하는 것은 같습니다. 이 해시가 분쟁이 생기기 전 공개 시각에 우리 키로 발행됐다는 것입니다.

```
// Reproducing a purchased dataset byte-for-byte
1. GET /api/index/snapshot/{runId}      → canonical JSON (exact bytes hashed)
2. sha256(bytes)                        → compare with the value you were given
3. SnapshotAnchor.isAnchored(hash)      → Flare mainnet - the anchor of record;
   SnapshotAnchor.anchors(hash)         block number and timestamp it was fixed at
4. WeatherOracleAnchor (IndexAnchored) → the same hash, first-generation, on BNB Chain
5. Runs before 2026-09-23 only: the same hash in the memo of a 1-drop XRP Ledger
   payment, provable on Flare via XrplAnchorVerifier and an FDC XRPPayment attestation
6. The anchoring block predates any dispute, so the data could not have been
   altered after the fact. DeterminationRegistry reads step 3 to mark a verdict
   settlement-grade - a contract needs nothing from us to check it.
```

6.2 FCC - 연산에 대한 증명

앵커링은 실행이 존재했고 사후에 바뀌지 않았음을 증명하지만, 애초에 규칙을 정직하게 돌렸는지는 보여주지 않습니다. Flare 의 기밀 컴퓨팅은 합의를 신뢰 실행 환경 안에서 돌리고, 엔클레이브가 출력에 서명하며, WeatherFccConsumer 가 그 서명 체인을 온체인에서 검증한 뒤에야 저장합니다. 상대방이 확인하는 것은 엔클레이브의 코드 해시이며, 그 안의 규칙은 기밀로 남습니다.

```
resultHash = keccak256(abi.encodePacked(
    keccak256(resultData), actionId, keccak256(bytes(tag)), status));
digest      = keccak256(abi.encode("TEE_ACTION_RESULT", block.chainid, resultHash));
require(ecrecover(EIP191(digest), v, r, s) == teeAddress);
```

Coston2에서 전 구간 검증했습니다. 올바르게 서명된 결과는 승인·저장되고, 위조 서명은 InvalidTeeSignature로, 재전송은 AlreadyRelayed로 되돌려집니다.

기밀 정산: 투명하되 보이지 않게

위의 증명된 합의값은 여전히 증명한 값을 공개합니다. 정산에서는 그 공개가 곧 유출입니다 - 임계값과 지수가 체인에 있으면 계약의 포지션이 상대방에게 노출됩니다. 블라인드 경로는 증명을 남기고 숫자를 뺍니다. 요청자가 조건을 약속(해시)하고 엔클레이브에 봉인해 보내면, 엔클레이브는 자체 11개 소스 합의값에 대해 평가하고 결과 비트·조건 약속·지수 약속·요청자에게 봉인된 보고서에 서명합니다. 소스가 3개 미만이면 '불충족'이 아니라 '평가불능'입니다. 분쟁 시 salt 를 가진 쪽이 약속을 열 수 있습니다 - 공개를 가능하게 하되 자동으로 하지 않는 설계입니다.

```
requester  seal(condition) -> enclave pubkey      commitment = keccak(cityId, metric, op,
thresholdE2, salt)
enclave    open · collect 11 sources · median+MAD · evaluate
           sign(cityId, evaluatedAt, commitment, outcome, settleable, nSources,
               indexCommitment, sealedReport -> requester)
chain      stores bits + two commitments. No threshold, no index.
```

7. 정산

구매는 청구서가 아니라 컨트랙트 호출입니다. 견적이 상품·도시·기간을 quoteld에 묶고, 구매자는 온체인에서 결제하며, 서버는 발생한 이벤트를 견적과 대조한 뒤에 데이터를 제공합니다. 다중 도시 구매는 단일 트랜잭션으로 정산되며, citiesHash가 정확한 도시 집합을 결제에 바인딩합니다.

가격은 정액이 아니라 기간 단위입니다. 도시당 30일 25 또는 1년 200 USD이며, XRP Ledger 에서 네이티브 XRP 로 냅니다. XRP 금액은 견적 시점에 Flare 의 FTSOv2 XRP/USD 시세로 USD

정가를 환산해 건적 유효시간 10분 동안 고정됩니다. 구독은 이 두 기간으로만 판매합니다. 구독은 그 도시의 축적분 전부 - 이력, 소스별 원값, 5분 실황, 전체 예보, 매시간의 판정 - 를 제한 없이 열고 그 도시의 판정 상품을 포함합니다. 호출당 0.04 USD 의 종량 크레딧은 어느 도시든 같은 데이터를 호출당 1크레딧으로 열며, 자사 고객에게 대량으로 다시 공개하는 용도에 맞는 형태입니다. 청구는 호출 수를 따릅니다.

플레이어에서는 같은 카탈로그를 네이티브 FLR로 결제할 수 있습니다. USD 가격이 실행 시점의 FTSOv2 온체인 시세로 환산되고, 초과 송금분은 같은 트랜잭션에서 환불됩니다. 환율을 정하는 오프체인 주체는 없습니다. XRP Ledger 에는 가격을 다시 계산할 컨트랙트가 없으므로 결속 방식이 다릅니다. 건적과 함께 발급된 DestinationTag 가 그 결제를 해당 주문에 묶고, 통화와 발행자, 그리고 보낸 금액이 아니라 실제로 도착한 금액을 원장에 직접 대조해 검증합니다.

7.1 선도 계약

모든 구매는 선도 계약입니다. 기간은 오늘 이후에 시작해 최대 1년까지 지정할 수 있으며, 과거 데이터는 판매하지 않습니다. 미래 날짜는 null 행으로 제공되다가 데이터가 쌓이는 대로 채워지며, 다운로드 URL은 고정됩니다. 오늘 작성해 다음 시즌에 정산하는 계약의 기준으로 쓸 수 있게 하는 장치입니다.

7.2 접근 모델과 판정 상품

무료 등급은 현재입니다. 지정한 도시의 현재 합의값과 앞으로 사흘의 예보를 시간당 120회까지. 축적된 것은 전부 유료 상품입니다 - 합의값 이력, 소스별 원값, 5분 실황, 스냅샷 정보 바이트, 16일 전체 예보, 그리고 판정 자체. 경계는 값과 판정 사이가 아니라 현재와 축적 사이에 있습니다. 무료 응답을 쌓아 두면 누구나 판정을 재구성해 다시 배포할 수 있고, 그것이 곧 상품이기에 때문입니다.

판정이 우리가 파는 단위입니다. 사는 쪽이 조건 - 도시, 지표, 비교, 임계값, UTC 대상 시각 - 을 개설하면, 그 시각이 지난 뒤 다음 실행이 셋 중 하나로 답합니다. 맞다: 조건이 충족됐고 합의값이 정산에 쓸 만큼 뒷받침됐다. 틀리다: 같은 기준에서 충족되지 않았다. 보류: 독립 관측이 갈렸거나, 참여 소스가 모자랐거나, 그 시각의 실행이 없어 판정을 내리지 않는다. 보류는 실패가 아니라 결과입니다. 그 시각이 정산을 감당할 수 없다는 뜻이고, 그것이 바로 지는 쪽이 숫자를 받아들이기 전에 먼저 받아들여야 하는 정보입니다.

모든 결과에는 리포트가 붙습니다. 그 시각의 합의값, 조건, 합의값을 뒷받침한 독립 관측 수와 어긋난 수, 참여 소스 수와 제외된 소스 수, 그리고 평이한 문장의 이유. 리포트는 규칙을 공개하지 않으면서 결과를 설명합니다. 판정은 해당 도시를 덮는 구독에 포함되고, 그 외에는 종량

크레딧에서 건당 결제됩니다. 자격 없는 요청은 402 와 함께 가격과 충전 흐름을 받으며, 자율 에이전트가 스스로 판정을 개설하고 결제하는 경로가 이것입니다. 판정은 확정적입니다 - 같은 조건과 시각은 같은 앵커된 실행을 읽어 같은 결과를 냅니다.

8. 기계 친화적 거래

에이전트는 예외 사례가 아니라 1급 구매자입니다. 구매 경로 어디에도 가입 절차, API 키 발급 과정, 사람의 승인이 없습니다. 지갑이 신원이고 결제가 권한 부여입니다.

```
// Autonomous purchase - the entire flow
quote = GET /api/quote?cities=seoul,tokyo&from=...&to=...
        → { quoteIdHash, cityIds, amountWei, contract, usdt, chainId }
approve(usdt, contract, amountWei)
purchaseBundle(quoteIdHash, cityIds, fromDate, toDate) // one transaction
claim = POST /api/purchases/claim { txHash, quoteId }
        → dataset + sha256 + permanent downloadUrl

// Or pay in XRP: the same quote carries an "xrpl" block
quote.xrpl → { destination, destinationTag, amount: { drops }, rate: { usdPerXrp, source } }
send ≥ amount.drops of XRP with that destinationTag // any XRPL wallet, no approve
step
claim = POST /api/purchases/claim { txHash, quoteId, chain: "xrpl" }
```

호출당 이용은 같은 계정이 DestinationTag 402 를 넣어 결제 주소로 XRP 를 보내고 그 계정의 서명과 함께 트랜잭션 해시를 제출합니다. 서버는 검증된 원장에서 결제를 확인하고 도착한 XRP 를 그 시점의 FTSOv2 시세로 환산해 호출 크레딧을 발급합니다. 크레딧이 소진되면 종량 엔드포인트가 기계가 읽을 수 있는 결제 안내와 함께 402 Payment Required 를 돌려주므로, 에이전트가 사람 개입 없이 스스로 충전하고 계속할 수 있습니다.

같은 도구를 MCP 로도 /api/mcp 에서 쓸 수 있습니다. Cursor-Claude Desktop 등 Streamable HTTP 호스트에서 도는 에이전트가 HTTP 호출을 직접 짜지 않고도 현재 합의값을 읽고, HTTP 에서 쓰는 것과 같은 자격으로 그 뒤의 소스별 값과 스냅샷 검증 증거를 받습니다. 현재값에는 키가 필요 없고, 축적분과 모든 판정은 위의 지갑 흐름을 지닙니다.

9. 보안 모델

서버는 고객 키를 다루지 않습니다. 모든 서명은 구매자 지갑에서 이루어지고 백엔드는 트랜잭션 영수증이나 원장 기록을 검증할 뿐입니다. 서버가 보관하는 유일한 키는 스냅샷 앵커와 판정을 게시하는 일만 하며, 자금을 옮길 수 없습니다.

결제 무결성은 API에 대한 신뢰가 아니라 컨트랙트 이벤트에 근거합니다. 클레임 엔드포인트는 영수증에서 예상 건적 해시·도시 해시·기간·금액을 다시 유도해 하나라도 어긋나면 거부합니다. 재사용은 트랜잭션 해시의 유일성 제약으로 막습니다. 수집 엔드포인트는 시크릿으로 잠겨 있습니다. 인증 없는 호출자가 상위 소스의 호출 한도를 소진시켜 오라클을 멈출 수 있고, 이는 공격자에게 아무 비용도 들지 않는 데이터 거부 공격이기 때문입니다.

10. 배포 현황

정산은 두 메인넷에서 돕니다. Flare 메인넷이 정본 앵커(SnapshotAnchor), 판정 레지스트리(DeterminationRegistry), 그리고 2026년 9월 23일 이전 실행을 위한 레거시 앵커 검증기(XrplAnchorVerifier)를 맡고, XRP Ledger 가 XRP 결제를 받으며 그 금액은 건적 시점에 Flare 의 FTSOv2 XRP/USD 피드로 정해집니다. 2026년 9월 23일까지 USDT 정산과 1세대 앵커를 맡았던 BNB 스마트 체인은 퇴역했습니다. 1,452건의 앵커는 계속 검증할 수 있고 그때의 구매는 계속 내려받을 수 있지만, 새 트랜잭션은 만들지 않습니다. FTSOv2 마켓과 FCC 소비자만 Coston2 에 남아 있는데, FCC 가 아직 정식 출시 전이기 때문입니다. 소스는 블록 탐색기에서 검증되어 있고 주소는 빌드 상수가 아니라 설정이므로, 구성요소를 다른 네트워크로 옮기는 것은 재배포가 아니라 설정 변경입니다.

| 구성요소 | 네트워크 | 주소 |

| --- | --- | --- |

| Legacy anchor (until 2026-09-23) | BNB Chain (56) | weathermarket.io/whitepaper |

| XRP payment | XRP Ledger (mainnet) | weathermarket.io/whitepaper |

| Legacy memo anchor (until 2026-09-23) | XRP Ledger (mainnet) |

weathermarket.io/whitepaper |

| Snapshot anchor | Flare (14) | weathermarket.io/whitepaper |

| Anchor proof (FDC) | Flare (14) | weathermarket.io/whitepaper |

| Verdict registry | Flare (14) | weathermarket.io/whitepaper |

| FTSOv2 market | Flare (14) | weathermarket.io/whitepaper |

| FCC consumer | Flare (14) | weathermarket.io/whitepaper |

주소는 운영 설정에서 옴으로 이 허브에는 적지 않는다. 라이브 백서(위 링크) 10장에 현재 값이 있다.

11. 현황과 한계

증명된 것과 아직 아닌 것을 그대로 적습니다. 스스로를 과장하는 정산 오라클은 단지 미완인 오라클보다 나쁘기 때문입니다.

온체인에서 동작하고 검증된 것: 11개 소스 전체의 매시간 합의, 교차검증 판정, 매 실행의 정규 해시를 Flare 메인넷에 앵커(첫 실행까지 소급 백필)하는 것, XRPL 메인넷의 XRP 구독 정산(견적 시점 FTSOv2 시세로 금액 고정, DestinationTag 로 견적과 결속, 실제 도착한 drop 으로 검증), Flare 메인넷의 판정 레지스트리(실행이 앵커된 모든 판정이 settlement-grade), 실시간 FTSOv2 피드로 가격을 정하고 환불하는 FLR 정산, 위조 서명과 재사용을 거절하는 FCC 서명 체인, 402 결제 안내가 붙은 호출당 미터링. 구매·클레임·다운로드·미터링은 지갑 사용자와 자율 에이전트 양쪽이 전 구간을 실행해 봤습니다.

아직 완결되지 않은 것: FCC 경로가 현재 운영 엔클레이브가 아니라 테스트 키로 서명합니다. 연산 증명의 절차는 실증됐지만 아직 실제 증명된 하드웨어에 뿌리내리지 못했고, 이 경로와 FTSOv2 마켓은 FCC 가 Flare 에서 정식 출시될 때까지 Coston2 에 남습니다. 블라인드 정산은 작성했으나 배포하지 않았습니다. 결제는 네이티브 XRP 라 FDC 가 증명할 수 있지만, 결제 증명을 소비하는 컨트랙트는 아직 없습니다 - 결제는 서버가 원장에 대조해 검증합니다. RLUSD 는 아직 받지 않습니다. FDC 는 네이티브 XRP 결제만 증명하므로, 발행통화 증명 타입이 생기기 전에는 RLUSD 결제를 Flare 컨트랙트에 증명할 수 없기 때문입니다. 앵커는 모든 앵커가 그렇듯 우리 키가 씁니다. 그 가치는 작성자가 아니라 공개 시각에 있고, 우리에게 의존하지 않는 부분은 누구나 읽을 수 있는 판정 레지스트리와 누구나 다시 빌드할 수 있는 엔클레이브 코드 해시입니다. 합의는 소스가 독립적인 만큼만 독립적입니다. 모델 피드 7종이 하나의 애그리게이터를 거치므로, 기준 관측과 케이웨더가 독립적으로 남더라도 그쪽에 장애가 나면 다양성은 떨어집니다.

부록 - 인터페이스

전체 엔드포인트 레퍼런스와 실행 가능한 에이전트 예제는 [개발자 문서](#) 에 있습니다. 실시간 컨트랙트 레지스트리, 최신 앵커 증거, 재현 절차는 [기술 증명](#) 페이지에 있습니다. 공개 합의 피드는 인증 없이 열려 있습니다: `/api/index`

Weather Data Market · 합의 날씨 오라클 · 데이터가 아니라 판정을 팝니다 · 정본 앵커와 판정 레지스트리는 Flare · XRP Ledger 의 XRP · 매 실행 해시 앵커, 규칙은 고정되고 비공개

