Kaiko

MIGRATION GUIDE

TradFi Level 1 Quotes & History

Amberdata → Kaiko

2026

www.kaiko.com

1 At a glance

What this data gives you. Level 1 quotes on the listed options of crypto-related US equities and ETFs, such as IBIT and MSTR, in 5-minute snapshots: for every contract, the best bid and ask, bid, ask and mark implied volatility, mark price, the five Greeks, and the underlying price and rate. This guide covers both ways to request it.

	Chain endpoint	History endpoint
Use it for	The option chain of one underlying, or a slice of it: the latest snapshot, or a window of history	The history of one contract, or of every contract in one expiry
Amberdata	`/level-1-quotes/tradfi`	`/level-1-quotes-instrument/tradfi`
Kaiko	`/tradfi-level-1-quotes`	`/tradfi-level-1-quote-history`

The two endpoints return the same records. Only the way you select them differs. In this guide, **Chain** and **History** mark what applies to one endpoint only.

What changes for you

- **The data is the same.** Both providers carry the same contracts, and where they carry the same quote the values agree (see §7).
- **Names change**: `currency=IBIT` becomes `base_asset=ibit` (lowercase), fields move to snake_case (`underlyingPrice` and `indexPrice` both become `index_price`), and instrument names drop the `.0` on whole-number strikes (see §3, §5, §6).
- **History: one contract is one parameter on Kaiko.** Amberdata identifies a contract by `expiration`, `strike` and `putCall`; Kaiko takes its `instrument` name (see §3).
- **Chain: Kaiko pages; Amberdata does not.** A full chain over an hour cannot be returned by Amberdata at all; on Kaiko it is one request followed through `meta.next_url` (see §7).
- **Two access types.** Each Kaiko endpoint has its own; if you use both, you need both (see §2).
- **History starts on 2026-10-14 on Kaiko.** Amberdata holds older data. If you need it, arrange it before you cut over (see §7).

Coverage & distribution

Data type: Level 1 option quotes with implied volatility and Greeks, per contract

Underlyings: crypto-related US equities and ETFs; the same 18 on both (§6)

Granularity: 5-minute snapshots; hourly and daily on request

Market hours: US trading sessions only, 14:30–21:00 UTC on weekdays (13:30–20:00 UTC while US daylight saving time is in effect), excluding US exchange holidays

Distribution: REST API (JSON, paginated on Kaiko)

2 Endpoint & authentication

	Amberdata (today)	Kaiko (new)
Method	GET	GET

	Amberdata (today)	Kaiko (new)
Host	`https://api.amberdata.com`	`https://api.kaiko.io`
Path: chain	`/markets/derivatives/analytics /volatility/level-1-quotes/tradfi`	`/analytics/volatility /tradfi-level-1-quotes`
Path: history	`/markets/derivatives/analytics /volatility/level-1-quotes-instrument/tradfi`	`/analytics/volatility /tradfi-level-1-quote-history`
Auth	API key header `x-api-key`	API key header `X-Api-Key`

Access. Each Kaiko endpoint needs its own access type on your account: `Equity_options_level_1_quotes` for the chain and `Equity_options_level_1_quote_history` for history. If you use both endpoints, you need both. Without the access type an endpoint returns HTTP 403 `not_entitled`; ask your Kaiko contact to enable it.

3 Request parameters

Applies to: Both endpoints, or the **Chain** or **History** endpoint only. Where the endpoints behave differently, the parameter has one row for each.

Amberdata param	Kaiko param	Applies to	Change	Notes
`currency` (required)	`base_asset` (required)	Both	RENAMED	**Lowercase**: `IBIT` → `ibit`. Kaiko rejects uppercase with HTTP 400. On history the instrument you send must belong to it.
—	`quote_asset`	Both	NEW	Optional; `usd` is the only value.
`instrument`	`instrument`	**Chain**	RENAMED	Same name, new format on whole-number strikes: `IBIT-20NOV26-45.0-C` → `IBIT-20NOV26-45-C` (§6).
`expiration` + `strike` + `putCall`	`instrument`	**History**	RENAMED	**Amberdata has no `instrument` parameter here** (HTTP 400 "not supported"). Three parameters become one name. See Identifying one contract below.
`expiration`	`expiry`	Both	RENAMED	ISO date, `2026-11-20`. On history it is **required on Amberdata**; Kaiko requires `instrument` or `expiry`.
`putCall`	`option_type`	Both	RENAMED	Kaiko takes `call` or `put`. Amberdata also accepted `C` / `P`.
`strike`, `minStrike`, `maxStrike`	`strike`, `min_strike`, `max_strike`	Both	RENAMED	Same meaning. On history they filter the contracts of one expiry.
`startDate`	`start_time`	Both	RENAMED	**Inclusive** on both. Kaiko accepts ISO 8601 with a time zone only (`2026-11-03T19:00:00Z`). **With no dates**: on the chain both return the latest snapshot; on history Amberdata returns the last 24 hours and Kaiko today's session so far.

Amberdata param	Kaiko param	Applies to	Change	Notes
`endDate`	`end_time`	Both	RENAMED	**Exclusive on Kaiko.** Amberdata documents it as exclusive, but in testing it was **inclusive** on both endpoints, so the same window returned one more snapshot from Amberdata (§8).
`timeInterval`	`interval`	**Chain**	RENAMED	`minute` / `hour` / `day` → `5m` / `1h` / `1d`. Amberdata's `minute` already returns 5-minute snapshots. Kaiko defaults to `5m`. Window caps per call in §7.
`timeInterval`	`interval`	**History**	RENAMED	Same values. **Amberdata's `day` returns an empty response here** (HTTP 200, no rows), although it works on its chain endpoint. Kaiko's `1d` works on both.
`isAtm`	`is_atm`	**Chain**	RENAMED	`true` returns the at-the-money contracts only. In testing both selected the same contracts.
—	`is_atm`	**History**	NEW	**Amberdata rejects `isAtm` here** (HTTP 400 "not supported"). Kaiko's works: with `expiry` it returns that expiry's at-the-money call and put.
`minDaysTo Expiration`, `maxDaysTo Expiration`	`start_days_ to_expiry`, `end_days_ to_expiry`	**Chain**	RENAMED	**Both exclude today's and tomorrow's expiries by default** (a lower bound of 1). Send `0` to include them. The upper bound is not counted identically: in testing a 7-day limit on 2026-11-03 included the 11-11 expiry on Amberdata and not on Kaiko.
—	`start_days_ to_expiry`, `end_days_ to_expiry`	**History**	NEW	Amberdata rejects `minDaysToExpiration` here (HTTP 400 "not supported"). Kaiko accepts them; only useful with `expiry`. The lowest `start_days_to_expiry` is **1** here: `0` returns HTTP 400.
—	`page_size`	Both	NEW	Rows per page, 1 to 10,000; default 1,000. Set `10000`. A full IBIT chain is about 2,800 rows (one page instead of three). One contract fits in a single page; a whole expiry may need more than one.
—	`sort`	**Chain**	NEW	`asc` or `desc`. Default `desc`, newest snapshot first, which is also Amberdata's order.
—	—	**History**	IDENTICAL	**No sort on either.** Kaiko rejects `sort` here with HTTP 400. Both return newest first.
`timeFormat`	—	Both	REMOVED	Kaiko timestamps are always ISO 8601 UTC strings.

Identifying one contract (history endpoint)

Amberdata's history endpoint works per expiry. `expiration` is required; sent alone, it returns every contract of that expiry (172 for MSTR on 2026-11-20). `strike` and `putCall` narrow it to one contract. Kaiko takes the contract's name instead:

Amberdata: `currency=MSTR&expiration=2026-11-20&strike=30&putCall=P`

Kaiko: `base_asset=mstr&instrument=MSTR-20NOV26-30-P`

Whole expiry, Kaiko: `base_asset=mstr&expiry=2026-11-20`, paged as in §4

To build the name: underlying, expiry as `DDMMMYY`, strike with no `.0` on whole numbers, then `C` or `P`, joined by hyphens (§6).

4 Response envelope & metadata

The fields around the data, the wrapper your parser reads before reaching the numbers. This envelope is **identical on every Amberdata endpoint**, and Kaiko's is the same on both endpoints in this guide.

Amberdata field	Kaiko field	Change	Notes
`status`	—	REMOVED	Read the HTTP status. A rejected Kaiko request returns 4xx with a JSON body: `code`, `detail` and, for parameter errors, `errors[]` naming each parameter.
`title`	—	REMOVED	"OK" label, no Kaiko equivalent.
`description`	—	REMOVED	Free-text status description.
`payload.data`	`data`	RENAMED	**Array** of quotes, at the top level: there is no `payload` wrapper (see §5).
`payload.metadata` `.api-version`	`meta.api_version`	RENAMED	`2023-09-30` on Amberdata; `2026-10-01` on Kaiko.
—	`meta.next_url`	NEW	URL of the next page, carrying its own cursor. On the last page it is `null`; loop until it is.
—	`meta.count`	NEW	Rows on this page.
—	`meta.url`	NEW	Echo of the request as Kaiko resolved it.

5 Response data fields

One record per contract per snapshot, with the same fields on both endpoints, except the one row marked below. Kaiko uses snake_case throughout.

Amberdata field	Kaiko field	Type	Change	Notes
`instrument`	`instrument`	string	RENAMED	Whole-number strikes lose the `.0`: `IBIT-05NOV26-33.0-P` → `IBIT-05NOV26-33-P`. Half strikes are unchanged (`44.5`).
`timestamp`	`value_time`	string	RENAMED	The 5-minute snapshot the row belongs to.

Amberdata field	Kaiko field	Type	Change	Notes
`exchange Timestamp`	`exchange_time`	string	RENAMED	When the quote was taken. Amberdata to the second; Kaiko to the **nanosecond** (`2026-11-03T19:00:59.715485000Z`). Compare these before comparing values (§7).
`expiration Timestamp`	`expiry`	string	RENAMED	ISO 8601, midnight UTC of the expiry date.
`strike`, `multiplier`	`strike`, `multiplier`	number	IDENTICAL	
`putCall`	`option_type`	string	RENAMED	`C` / `P` → `call` / `put`.
`bid`, `ask`	`bid`, `ask`	number	IDENTICAL	Null when there is no quote on that side.
`bidVolume`, `askVolume`	`bid_size`, `ask_size`	number	RENAMED	Null on every row on both, in testing.
`bidIv`, `askIv`, `markIv`	`bid_iv`, `ask_iv`, `mark_iv`	number	RENAMED	**In percent** on both (`37.139` = 37.139%). Kaiko gives three decimals, Amberdata two.
`markPrice`	`mark_price`	number	RENAMED	
`delta`, `gamma`, `theta`, `vega`, `rho`	same names	number	IDENTICAL	Exchange-provided on both (`is_exchange_provided_greeks`).
`underlying Price` (chain) `indexPrice` (history)	`index_price`	number	RENAMED	Price of the underlying equity or ETF. **Amberdata names it differently on its two endpoints**; Kaiko uses `index_price` on both.
`underlying Rate`	`underlying_rate`	number	RENAMED	
`isAtm`, `isCarry Forward`, `isExchange ProvidedGreeks`	`is_atm`, `is_carry_ forward`, `is_exchange_ provided_greeks`	boolean	RENAMED	
—	`open_interest`, `volume`	number	NEW	Kaiko only. Null on every row in testing.
`currency`, `exchange`, `instrument Normalized`	—	string	REMOVED	Not on Kaiko. The underlying is the one you requested; Amberdata's `exchange` is always `tradfi`; `instrumentNormalized` is `TRADFI-` + the instrument name.

6 Underlyings & instrument names

There is no exchange parameter on either side. Map the underlying you send and the instrument names you store:

Amberdata `currency`	Kaiko `base_asset`	Notes
`IBIT, MSTR, COIN, MARA, GME, CRCL`	`ibit, mstr, coin, mara, gme, crcl`	Lowercase on Kaiko
`BITB, BITO, BITQ, BITX, EETH, ETHA, ETHU`	`bitb, bito, bitq, bitx, eeth, etha, ethu`	
`MSTU, MSTY, SATO, SBET, BMNR`	`mstu, msty, sato, sbet, bmnr`	
`BTC, ETH`	—	Listed by Amberdata but returned no data in testing; Kaiko returns 404 `unknown_instrument`

In testing on 2026-11-04, each of the 18 underlyings returned the same number of at-the-money contracts from both providers.

Instrument names. Both use `UNDERLYING-DDMMMYY-STRIKE-C|P` in upper case, on both endpoints. Amberdata writes every strike with a decimal (`45.0`); Kaiko drops it on whole numbers (`45`) and keeps it otherwise (`44.5`). Strip a trailing `.0` from stored Amberdata names to match them.

History requests. If you called Amberdata's history endpoint with `expiration` , `strike` and `putCall` , build the Kaiko name from those three values (§3). Send it with the `base_asset` it belongs to.

7 Will your values change?

Short answer

The **quotes are the same** where both providers carry the same quote. What changes is the **history available** and, on the chain endpoint, **how you page through it**. Testing compared each provider's history wherever both carried the same quote. On the chain, over 15 sessions (2026-10-14 to 2026-11-03), bid, ask, the three IVs, mark price, the Greeks and the underlying price agreed on 99.97% of rows, allowing for Amberdata's rounding (§5). On history, for one contract over 16 sessions to 2026-11-04, all of those fields agreed on all 772 snapshots.

On both endpoints

- **Delayed data on both.** In testing, neither provider's newest snapshot was ever fresher than about 20 minutes. That is consistent with a standard 15-minute delayed options feed. Kaiko was ahead of Amberdata by about 5 minutes on the chain, and by 4–10 minutes (8.6 on average) on history. Snapshots are 5 minutes apart, so polling more often than every 5 minutes returns the same snapshot.
- **Compare quote times before values.** Each snapshot holds the quote for each contract at a moment inside its 5-minute bucket, and the two providers do not always take the same moment: in testing they held different quotes on about 4 in 10 shared rows on the chain and 3 in 10 on history. Match on `exchange_time` before treating a value difference as a discrepancy.
- **Amberdata rewrites history; Kaiko does not.** In testing, Amberdata replaced snapshots after first publishing them: a snapshot first served with quotes from the end of its 5-minute bucket was later served with quotes from its start, and 48 of 52 values changed. Kaiko's snapshots did not change. If you stored Amberdata data and re-fetch a window to reconcile, expect different values from Amberdata; reconcile Kaiko against Kaiko.

- **History starts on 2026-10-14.** Kaiko has no data before then on either endpoint. On its chain endpoint, Amberdata holds 5-minute data from 2024-11-01 and daily data from 2021-12-01. On its history endpoint it held the tested contract from 2026-10-05, and expiries as far back as June 2025. If you query older history, keep your Amberdata archive or arrange it with Kaiko support (support@kaiko.com) before you cut over.
- **Market hours.** Data exists only during US sessions, 14:30–21:00 UTC on weekdays (13:30–20:00 UTC under US daylight saving time). Both return HTTP 200 with an **empty** `data` array for a window outside a session, on both endpoints. On the chain, a request for the latest snapshot outside a session returns the last snapshot of the previous session. Treat an empty response outside market hours as normal, not as an outage.
- **Snapshots per session.** A session includes the snapshot at the close, and contracts still quoted after the close get later snapshots too. In testing, IBIT contracts ran to 10 minutes past it. The number of snapshots in a session is therefore not fixed: a full session of one MSTR contract was **79 snapshots**, ending at the close, and one IBIT contract was **81**. Across the whole IBIT chain on 2026-11-05, 2,525 of 2,846 contracts had a snapshot 5 minutes after the close, and 2,119 had one 10 minutes after. If you check a session for completeness, do not expect a fixed count.
- **IV is in percent on both these endpoints.** Kaiko's `variance-premium` endpoint on the same host returns volatility as a fraction (`0.393` = 39.3%). If you consume both, do not share one scaling function.

Chain endpoint

- **Window caps per request.** Kaiko: 1 hour at `5m`, 24 hours at `1h`, 31 days at `1d`, returned in pages of up to 10,000 rows. Amberdata has the same caps but no paging, and a **10 MB response limit** that bites first: a full IBIT chain over one hour at 5-minute granularity is about 25 MB, so Amberdata cannot return it at all. On Kaiko it is one request returned in pages: follow `meta.next_url` until it is null. A longer range is several requests on both.
- **Response time.** In testing from AWS us-east-1, 95% of Kaiko requests for the latest full IBIT chain (about 2,800 contracts) returned within 384.2 ms, and for one hour of one expiry within 266.0 ms. Amberdata's figures were 1,812.4 ms and 912.2 ms. The same latest chain was 1.6 MB from Kaiko and 2.1 MB from Amberdata. If your timeouts were sized for Amberdata's full-chain requests, Kaiko's do not need that headroom.

History endpoint

- **The caps do not bite on one contract.** In testing, Kaiko returned 7 days at `5m`, 32 days at `1h` and 2 years at `1d` in one page each; Amberdata returned 30 days at 5-minute granularity in one 1.2 MB response. Responses are small: one contract is about 5 KB for an hour and about 50 KB for a session. A whole expiry is larger: one day of a 172-contract expiry was 9.8 MB from Amberdata, just under its 10 MB limit, and runs to more than one page on Kaiko.
- **Daily history.** Amberdata's `timeInterval=day` returns no rows on this endpoint. Kaiko's `interval=1d` works, from 2026-10-14.
- **Order.** Both return newest first, and neither can change it (§3).
- **Response time.** In testing from AWS us-east-1, 95% of Kaiko requests for the last hour of one contract returned within 197.5 ms, and for one full session within 207.4 ms. Amberdata's figures were 404.5 ms and 455.5 ms. These responses are about 5 KB and 46–58 KB, against 0.8–2.1 MB on the chain, so the gap here is mostly per-request overhead, not transfer time: **do not compare these figures with the chain's**.

8 Before / after example

Chain endpoint: request (one IBIT contract, the 5 minutes from 19:00 UTC on 2026-11-03)

Amberdata

```
GET https://api.amberdata.com
    /markets/derivatives/analytics
    /volatility/level-1-quotes/tradfi
    ?currency=IBIT
    &instrument=IBIT-20NOV26-45.0-C
    &startDate=2026-11-03T19:00:00Z
    &endDate=2026-11-03T19:05:00Z
    &timeInterval=minute
x-api-key: <your-key>
```

Kaiko

```
GET https://api.kaiko.io
    /analytics/volatility
    /tradfi-level-1-quotes
    ?base_asset=ibit
    &instrument=IBIT-20NOV26-45-C
    &start_time=2026-11-03T19:00:00Z
    &end_time=2026-11-03T19:05:00Z
X-Api-Key: <your-key>

# for a full chain, add
# &page_size=10000 and follow
# meta.next_url until it is null
```

Chain endpoint: response (both real, captured 2026-11-04; keys as returned. Amberdata's inclusive `endDate` also returns the 19:05 snapshot, omitted here)

Amberdata


```
{
  "status": 200,
  "title": "OK",
  "description": "Successful request",
  "payload": {
    "data": [
      { ...19:05 snapshot... },
      { "ask": 1.0299999713897705,
        "askIv": 37.33,
        "askVolume": null,
        "bid": 1.0199999809265137,
        "bidIv": 37.07,
        "bidVolume": null,
        "currency": "IBIT",
        "delta": 0.4068300127983093,
        "exchange": "tradfi",
        "exchangeTimestamp":
          "2026-11-03T19:00:59.000Z",
        "expirationTimestamp":
          "2026-11-20T00:00:00.000Z",
        "gamma": 0.10570000112056732,
        "instrument":
          "IBIT-20NOV26-45.0-C",
        "instrumentNormalized":
          "TRADFI-IBIT-20NOV26-45.0-C",
        "isAtm": false,
        "isCarryForward": false,
        "isExchangeProvidedGreeks": true,
        "markIv": 37.14,
        "markPrice": 1.0225199460983276,
        "multiplier": 100,
        "putCall": "C",
        "rho": 0.008539999835193157,
        "strike": 45,
        "theta": -0.056540001183748245,
        "timestamp":
          "2026-11-03T19:00:00.000Z",
        "underlyingPrice": 43.8650016784668,
        "underlyingRate": 0.04462999850511551,
        "vega": 0.038350000977516174 } ],
    "metadata": {
      "api-version": "2023-09-30" } } }
```


Kaiko


```
{
  "data": [
    { "ask": 1.03,
      "ask_iv": 37.334,
      "ask_size": null,
      "bid": 1.02,
      "bid_iv": 37.073,
      "bid_size": null,
      "delta": 0.40683,
      "exchange_time":
        "2026-11-03T19:00:59.715485000Z",
      "expiry":
        "2026-11-20T00:00:00.000Z",
      "gamma": 0.1057,
      "index_price": 43.865,
      "instrument": "IBIT-20NOV26-45-C",
      "is_atm": false,
      "is_carry_forward": false,
      "is_exchange_provided_greeks": true,
      "mark_iv": 37.139,
      "mark_price": 1.02252,
      "multiplier": 100,
      "open_interest": null,
      "option_type": "call",
      "rho": 0.00854,
      "strike": 45,
      "theta": -0.05654,
      "underlying_rate": 0.04463,
      "value_time":
        "2026-11-03T19:00:00.000Z",
      "vega": 0.03835,
      "volume": null } ],
  "meta": {
    "api_version": "2026-10-01",
    "count": 1,
    "next_url": null,
    "url": "https://api.kaiko.io/analytics
      /volatility/tradfi-level-1-quotes
      ?base_asset=ibit&instrument=..." } }
```


Read across: the same contract and the same quote, taken at 19:00:59. `IBIT-20NOV26-45.0-C` is Kaiko's `IBIT-20NOV26-45-C`. Bid 1.02, ask 1.03, mark price 1.02252 and the Greeks are the same values, which Amberdata sends as float32 (`1.0199999809265137`). Mark IV 37.14 is Kaiko's 37.139 rounded to two decimals. `underlyingPrice` 43.865 is `index_price`. Kaiko adds `open_interest` and `volume`, and drops `currency`, `exchange` and `instrumentNormalized`.

History endpoint (one MSTR contract, the same 5 minutes; both real, captured 2026-11-05; excerpt of the identifying fields)

Amberdata

```
GET https://api.amberdata.com
    /markets/derivatives/analytics
    /volatility
    /level-1-quotes-instrument/tradfi
    ?currency=MSTR
    &expiration=2026-11-20
    &strike=30
    &putCall=P
    &startDate=2026-11-03T19:00:00Z
    &endDate=2026-11-03T19:05:00Z
    &timeInterval=minute

{ ...19:05 snapshot... },
{ "instrument":
    "MSTR-20NOV26-30.0-P",
  "strike": 30,
  "putCall": "P",
  "timestamp":
    "2026-11-03T19:00:00.000Z",
  "exchangeTimestamp":
    "2026-11-03T19:00:57.000Z",
  "bid": 0.009999999776482582,
  "ask": 0.029999999329447746,
  "indexPrice": 135.9550018310547,
  ... }
```

Kaiko

```
GET https://api.kaiko.io
    /analytics/volatility
    /tradfi-level-1-quote-history
    ?base_asset=mstr
    &instrument=MSTR-20NOV26-30-P
    &start_time=2026-11-03T19:00:00Z
    &end_time=2026-11-03T19:05:00Z

{ "instrument":
    "MSTR-20NOV26-30-P",
  "strike": 30,
  "option_type": "put",
  "value_time":
    "2026-11-03T19:00:00.000Z",
  "exchange_time":
    "2026-11-03T19:00:57.095679000Z",
  "bid": 0.01,
  "ask": 0.03,
  "index_price": 135.955,
  ... }
```

Read across: Amberdata's `expiration`, `strike` and `putCall` become the one name `MSTR-20NOV26-30-P`. The quote is the same one, taken at 19:00:57. Amberdata's `indexPrice` is `index_price`, as `underlyingPrice` is on the chain. The other fields map as in the chain example above.

Questions? Kaiko support · support@kaiko.com

docs.kaiko.com