

iOS app

Contract and parity

🕒 3 min read · Updated 25 September 2026 · Verified against 417c92f on 2026-09-25

The Swift search, verify, reader, timing, bani and insight code is a port of the platform's Python, proven byte for byte against the vendored golden contract on the derived database, suite by suite.

<https://docs.gurbanisoul.com/ios/contract-and-parity/>

The app does not call the API: it ships its own Swift implementation of search, verification, the reader, timing, the banis and the insights, against a database it bundles. What keeps that implementation honest is the **golden contract** — recorded outputs of the platform's real functions — vendored at the release tag and replayed by parity tests on every pull request.

What is vendored

gurbani-soul-ios/vendor.lock.json · lines 1–32

```
{
  "sources": {
    "data": {
      "commit": "c36e3172bdf38a61070849bb604b38da89ca5d8b",
      "files": {
        "pipeline/build_ios_db.py":
        "b793cc73815601c50ced6287446967a4936cf0216a1572ce4ffff509e2f0f80a"
      },
      "ref": "c36e3172bdf38a61070849bb604b38da89ca5d8b",
      "repository": "Algorythmos-AI/sggs-data",
      "version": "1.2.0"
    },
    "platform": {
      "commit": "a7c43647dd8a3e6b3a49eac757f8b1aa68438ed0",
      "files": {
        "contract/_meta.json":
        "4d134ac1b1d599b3032d9e04a4eb85617656763b0e66a4120f5e01507abfc795",
        "contract/golden_analytics.ndjson":
        "a83e308a69414fbecebb0029a34f71487d4d55483d2e041925efc84d19270731",
        "contract/golden_banis.ndjson":
        "2aec62ccbd402e0324b8474256a12c858467a75f9a509f7908fa27a8506ea584",
        "contract/golden_constellation.ndjson":
        "1b21ddf67993ea883596b0258046e91710bbe829cb7deb9fdf2db263818c6e2c",
        "contract/golden_difflib.ndjson":
        "2abfeef0181ab2cae25c9c671336502a76245255c672d81f1bf73aa5d6bcacf56",
        "contract/golden_pahar.ndjson":
        "c1aa55c13ccfe90852265a095441fdc0253dc7ada56175919a4111fdf258c1ff",
        "contract/golden_reader.ndjson":
        "d31dddbd5d89e3e79655325cdce53668843bc0c83ecc9e5ed0dde01e94ff29b6",
        "contract/golden_roman_norm.ndjson":
        "31a715867d9cb2fa4be62467d163fec038efd79d99d9a101410160fd1e48689a",
        "contract/golden_search.ndjson":
        "f48a3027cc60fc1c0ff5806915f992a5b663107b7ea4c9279864a7e6f36c1055",
        "contract/golden_timing.ndjson":
        "6f9de023a40374e19b0ad146c8c21e6b3ec3b28f6201eb8f54c2d50dff487216",
        "contract/golden_verify.ndjson":
        "9383e014e3f50f224f6aff67955120b3a33d951034f9ca41343da98e854ef79d",
        "docs/brand/tokens.json":
        "483d2c35f04872fb381c114b41056c325f8e7c177b5af66ef9c902420c2434c0",
        "frontend/public/contributors.json":
        "cb8257ec01caddb60222aba22ec0147737735435319f5d5260be49f8cb246e28"
      },
      "ref": "a7c43647dd8a3e6b3a49eac757f8b1aa68438ed0",
      "repository": "Algorythmos-AI/sggs-platform",
      "version": "1.4.9"
    }
  }
}
```

Read from the source at build time — [view these lines on GitHub at a386422](#).

Source: **vendor.lock.json**² in gurbani-soul-ios.

Two sources, each with a commit and a sha256 per file: the platform (the contract files and the contributors roster, at a release tag whose APP_VERSION the lock records) and sggs-data (the iOS database builder, at the dataset's commit). scripts/vendor_sync.py

check --remote proves in CI that every file equals both its lock entry and the file at its source commit.

The parity suites

Package GurbaniSearchKit holds the ports and their tests
(ios/Packages/GurbaniSearchKit/Tests/):

Suite	Replays	Against
GoldenVectorTests	golden_roman_norm.ndjson (24,719 fold vectors) and golden_difflib.ndjson (the scorer, repr-exact)	pure functions, no database
SearchParityTests	golden_search.ndjson — queries through the Swift waterfall	the derived database
VerifyParityTests	golden_verify.ndjson — claims and full verdicts	the derived database
ReaderParityTests	golden_reader.ndjson — Ang, shabad and Hukam projections	the derived database
TimingParityTests, BaniParityTests, AnalyticsParityTests	the knowledge, Nitnem and insight suites	the derived database
DegradationParityTests	how each feature degrades on an older database without a layer (capability bits, never a fork)	a database with tables dropped

make parity runs them locally; the iOS fidelity gate · parity job runs them on every pull request after building both database profiles with the vendored builder.

The same fold, in Swift

The phonetic fold is the sharpest example: one function, three homes, one contract ([The Roman fold](#)). The Swift port is read here from the app repository at the pinned commit:

gurbani-soul-ios/ios/Packages/GurbaniSearchKit/Sources/GurbaniSearchKit/RomanNorm.swift · fold · lines 31-79

```

public static func fold(_ s: String) -> String {
    var out: [String] = []
    // Python str.split() with no args splits on runs of whitespace and drops empties.
    let words = s.lowercased().split(whereSeparator: { $0.isWhitespace })

    for wsub in words {
        var w = String(wsub)
        w = w.replacingOccurrences(of: "w", with: "v")
            .replacingOccurrences(of: "z", with: "j")
            .replacingOccurrences(of: "q", with: "k")
            .replacingOccurrences(of: "x", with: "k")

        for (from, to) in digraphs {
            w = w.replacingOccurrences(of: from, with: to)
        }

        w = w.replacingOccurrences(of: "b", with: "v")
            .replacingOccurrences(of: "k", with: "g")
            .replacingOccurrences(of: "t", with: "d")
            .replacingOccurrences(of: "p", with: "v")

        if w.hasPrefix("y") { w = "j" + String(w.dropFirst()) }
        w = w.replacingOccurrences(of: "y", with: "")

        let chars = Array(w)
        // head = first char iff it is a vowel
        let head: String
        if let first = chars.first, vowels.contains(first) {
            head = String(first)
        } else {
            head = ""
        }
        // body = strip all vowels, then collapse runs of identical consonants to one.
        // (Removing vowels first can make two same consonants adjacent – they then
collapse.)
        var body = ""
        var prev: Character? = nil
        for c in chars where !vowels.contains(c) {
            if c != prev {
                body.append(c)
                prev = c
            }
        }

        let combined = head + body
        out.append(combined.isEmpty ? w : combined)
    }

    return out.filter { !$0.isEmpty }.joined(separator: " ")
}

```

Read from the source at build time — [view these lines on GitHub at a386422](#).

Source: [RomanNorm.swift](#)⁴ in gurbani-soul-ios.

Compare it with the Python on **The Roman fold**: the rules are the same in the same order, and `GoldenVectorTests` proves the outputs equal for every one of the 24,719 recorded inputs. The verifier's scorer is held the same way: `SequenceMatcher.swift` reproduces `diffLib` ratio for ratio (`golden_diffLib.ndjson`), which is why the verdict ladder can use the platform's thresholds unchanged.

What a parity failure means

- **The contract changed upstream** (a deliberate engine change released on the platform): the app re-vendors at the new tag and ports the change; the failing suite names exactly which records differ.
- **The port drifted** (a Swift change without a platform change): fix the port; the contract is the truth.
- **The database differs** (a dataset bump): the suites run on the database the vendored builder derives from the pinned dataset, so a bump is a lock change plus a rebuilt database pair, never a test edit.

Nothing in the app repository is allowed to move the contract: vendored files are read-only by policy and by CI.

Links

1. <https://github.com/Algorythmos-AI/gurbani-soul-ios/blob/a386422e90c2c681f9ecc4f7167b4deb6a35fb92/vendor.lock.json#L1-L32>
2. <https://github.com/Algorythmos-AI/gurbani-soul-ios/blob/main/vendor.lock.json>
3. <https://github.com/Algorythmos-AI/gurbani-soul-ios/blob/a386422e90c2c681f9ecc4f7167b4deb6a35fb92/ios/Packages/GurbaniSearchKit/Sources/GurbaniSearchKit/RomanNorm.swift#L31-L79>
4. <https://github.com/Algorythmos-AI/gurbani-soul-ios/blob/main/ios/Packages/GurbaniSearchKit/Sources/GurbaniSearchKit/RomanNorm.swift>

Sri Guru Granth Sahib Ji — Knowledge Base wiki. Generated from commit c3d9360. Updated 25 September 2026. Verified against 417c92f on 2026-09-25. This copy may be out of date; the page is at <https://docs.gurbanisoul.com/ios/contract-and-parity/>