

Company Name: AakarDev Softwares
Contact Information: koradia.savan@gmail.com, +91-9428233422

SAMPLE REPORT — ALL DATA IS FICTIONAL, NOT A REAL CARTRIDGE OR CLIENT

Cartridge Engineering Health Report

int_party_cartridge · Loyalty Integration · PartyCo (Fictional Vendor)

Cartridge Version	4.1 (Current)
Audit Date	Sample — Illustrative
Framework	SFCC Tales Engineering Health Framework v1
Review Scope	Public source code & documentation review
Overall Health Score	82.0 / 100 — Low Risk

This is a sample report built entirely from fictional data, to illustrate the structure and depth of a full SFCC Tales Engineering Health Report. No real cartridge, vendor, or client is represented. It mirrors the format used in real engagements: public cartridge profiles on sfcc-tales.com show a condensed version of the findings below, while full detail, line-level evidence, and remediation guidance are typically available through a private assessment.

1. Executive Summary

Architect's Verdict

This fictional loyalty integration cartridge (int_party_cartridge) demonstrates solid SFRA architectural compliance and clean separation between storefront and backend logic. However, moderate platform-governance and performance risks are present: several background jobs run without chunked pagination, and a subset of write operations occur outside transaction wrapping. The cartridge is upgrade-friendly overall, but these governance and performance gaps should be resolved before high-volume production use.

Pillar	Score	Risk Level
Security	90.0 / 100	Low Risk
Platform Governance	62.5 / 100	Medium Risk
Upgradeability	75.0 / 100	Low Risk
Performance	70.0 / 100	Medium Risk
Maintainability	95.0 / 100	Low Risk
Documentation	88.0 / 100	Low Risk

Overall Health Score: 82.0 / 100 — Low Risk

Top Risks Identified

[HIGH] Top Risk #1 — Unchunked Background Job Processing

A nightly loyalty-points reconciliation job (sample: jobs/reconcilePoints.js) iterates all pending point-adjustment records in a single unbounded loop, without chunking or pagination. As the transaction volume grows, this job risks exceeding standard execution timeouts or exhausting available memory.

[MEDIUM] Top Risk #2 — Write Operations Outside Transaction Wrap

In the sample points-adjustment controller (sample: PointsAdjust.js), custom object updates are made directly without a Transaction.wrap block. Under concurrent requests, this creates a risk of partial or inconsistent writes to loyalty balance

e records.

Top Positive Finding

Clean Modular Structure — the integration separates backend logic, storefront templates, and Business Manager configuration into distinct cartridges, which keeps the storefront-facing code lightweight and isolates administrative configuration.

Estimated Fix Effort	8–12 hours (sample estimate)
Confidence Level	High — full sample codebase parsing, line-by-line validation of transactional code
Recommended Action	Use with caution — stable overall, but job chunking and transaction wrapping should be addressed before production deployment

2. Engineering Health Signals — Detailed Findings

Each check below reflects the underlying audit checklist (security, platform governance, upgradeability, performance, maintainability, documentation). PASS items are condensed; PARTIAL and FAIL items include full evidence and file references, as they would appear in a private assessment. All file names and code references in this sample are fictional.

PARTIAL SEC-003 — Missing input / webhook sanitization [MEDIUM]

Security · Reference: sample: notify.js, lines 10–15

Inbound loyalty webhook payloads are verified via a signature check. However, if the merchant does not configure the shared secret preference, signature verification defaults to a bypassed state.

FAIL GOV-004 — Transaction.wrap missing on write operations [MEDIUM]

Platform Governance · Reference: sample: PointsAdjust.js

The call to CustomObjectMgr.update() for loyalty balance records executes outside of a Transaction.wrap block, risking silent partial writes under concurrent access.

PARTIAL GOV-006 — Site Preferences used for secrets [MEDIUM]

Platform Governance · Reference: sample: service configuration

The outbound API key for the loyalty provider is stored in a custom Site Preference rather than a Service Credential, which is a platform governance best-practice violation.

FAIL PERF-004 — No job chunking / pagination on large datasets [HIGH]

Performance · Reference: sample: reconcilePoints.js

The nightly reconciliation job queries all pending adjustment records and iterates them in a single execution thread without chunk size limits, risking timeouts as data volume grows.

PARTIAL UPG-003 — Template full override [LOW]

Upgradeability · Reference: sample: loyaltyWidget.isml

One storefront template is fully shadowed rather than extended, which will require a manual merge review during future SFRA upgrades.

Full checklist (30+ checks across 6 pillars) with PASS-item evidence available in the complete private assessment.

3. Compatibility & Technology Profile

Capability	Support
SFRA	Yes
PWA Kit	Partial
OCAPI	Yes
SCAPI	No
Page Designer	Yes
Multi-Site	Yes
Multi-Currency	N/A
Multi-Locale	Yes
Min. SFCC Version	22.x (sample)

Installation Complexity

Metric	Value
Business Manager Config Steps	5 (sample)
Site Preferences to Configure	22 (sample)
Service Profiles	1 profile / 6 services (sample)
Metadata Import Required	Yes
Estimated Setup Time	1–2 hours (sample)
SI Complexity Rating	Low–Medium (sample)

4. Positive Findings

Every claim below is backed by evidence from the (fictional) source-code and documentation review.

✓ Clean Modular Structure

Backend logic, storefront templates, and Business Manager configuration are separated into distinct cartridges, keeping storefront code lightweight.

✓ Solid Documentation

An installation guide and configuration reference are present and reasonably current (sample evidence).

✓ Reasonable Security Baseline

State-changing endpoints include CSRF protection and basic input validation on the primary checkout-adjacent routes.

✓ Upgrade-Aware Design

Most controllers extend base SFRA routes via prepend/append rather than full overrides.

✓ Test Coverage Present

A meaningful set of unit tests exists alongside source modules (sample estimate: 40+ test files).

About This Report

Framework: SFCC Tales Engineering Health Framework

Method: Source-code and documentation review combined with Salesforce Commerce Cloud engineering best practices and hands-on implementation experience.

Disclaimer: This is a sample report built entirely from fictional data for illustrative purposes. `int_party_cartridge`, PartyCo, and all findings, file names, and figures in this document are invented and do not represent any real cartridge, vendor, or client engagement.

Want a report like this for a cartridge you're evaluating or a codebase you've inherited?

Request a Private Assessment — sfcc-ales.com/request-assessment