PROCURE-TO-REPORT TRANSFORMATION

Comprehensive Approach Note

Invoice Processing • Imprest Elimination • Contract Automation

Duplicate Prevention • Governance Framework

Prepared For: Jubilant Foodworks Limited

Prepared By: SequelString

January 2026

Executive Summary

This document addresses the procure-to-report challenges identified during our discovery session. Based on the discussions, we have identified potential improvement areas and propose comprehensive solutions with implementation options.

~100K	~51K	~30%	15+
Monthly Invoices (Potential Volume)	Monthly Imprest (Discussed)	Non-PO Spend (Estimate)	Payment Systems (Indicated)

Key Assumptions

The following assumptions are based on our discovery discussions. These should be validated during the detailed assessment phase:

- **Invoice Volume:** Organization may be processing approximately 100,000 invoices monthly across all locations and expense categories
- **Imprest/Petty Cash:** There could be approximately 51,000 monthly imprest claims requiring reimbursement processing
- **Non-PO Spend:** Roughly 30% of total spend may be flowing through non-PO channels (rent, utilities, reimbursements)
- **System Landscape:** Multiple payment systems (potentially 15+) may exist without real-time integration
- **PO Creation:** Several functional areas (possibly 6+) may be involved in PO creation with limited standardization
- **SAP as Core:** SAP is assumed to be the primary ERP and source of truth for financial data

Challenge 1: High-Volume Invoice Processing

Potential Issues Identified

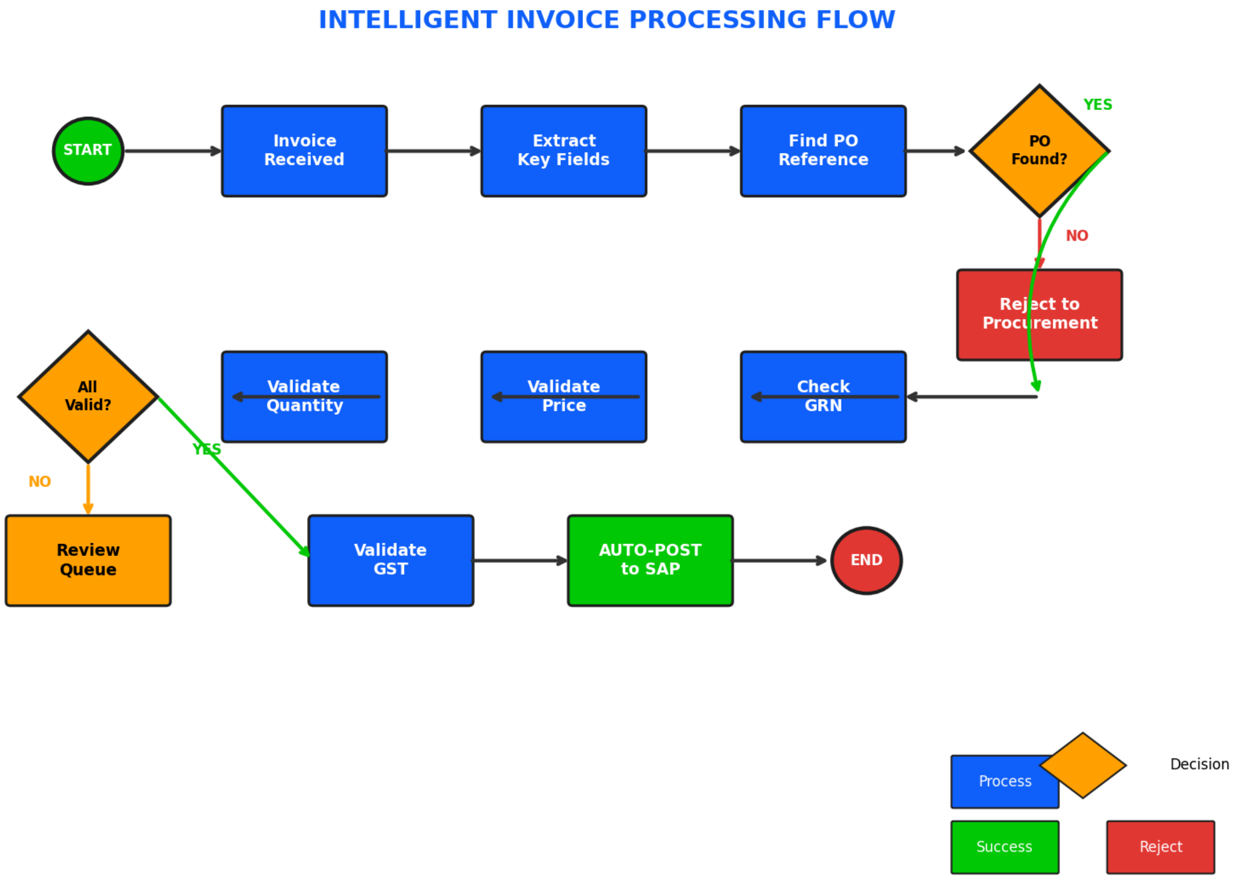
Current State (Possible)

- Manual matching of invoice to PO and GRN
- High processing time per invoice
- Delayed vendor payments
- AP team capacity constraints
- Low auto-posting rate (<20%)

Business Impact (Potential)

- Vendor dissatisfaction from delays
- Early payment discounts missed
- High processing cost per invoice
- Error-prone manual verification
- Scalability challenges

Proposed Solution: Intelligent 3-Way Match Engine



How It Works

The system automatically validates every invoice through a series of checks before posting to SAP:

Step	Validation	Pass Criteria	Fail Action
1	PO Lookup	Valid PO exists and is open	Reject to Procurement
2	Quantity Check	Invoice qty ≤ PO remaining qty	Route to Supply Chain
3	Price Validation	Within ±0.5% tolerance	Route based on variance
4	GRN Verification	Goods receipt exists	Hold until GRN created
5	Tax Validation	GST matches expected treatment	Route to Tax team

```
**Price Variance Decision Logic:** IF variance ≤ 0.5% → AUTO-APPROVE (minor rounding) IF variance 0.5% to
5% → ROUTE to Procurement Review IF variance > 5% → BLOCK payment, Alert Finance
```

Expected Outcomes

Metric	Current (Estimated)	Target
Auto-posting rate	<20%	60-70%
Processing time (auto)	3-5 days	Same day

Manual intervention	100%	30-40% (exceptions only)

Alternative Approach: Evaluated Receipt Settlement (ERS)

If PO-to-Invoice matching complexity is too high initially, consider ERS for high-volume, trusted vendors:

- Payment triggered automatically when goods are received (GRN-based)
- No invoice required from vendor
- Best suited for top 20-30 vendors with contract-locked pricing
- Prerequisites: Accurate GRN capture, vendor agreement to ERS terms

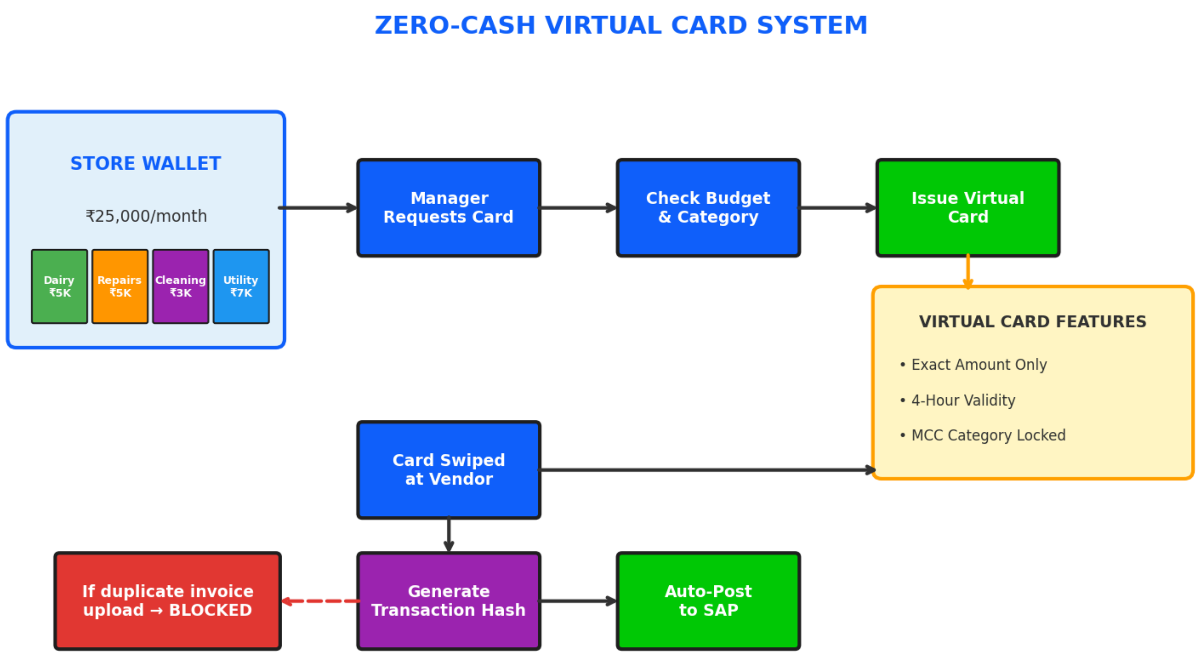
Challenge 2: Imprest/Petty Cash Claims

Potential Issues Identified

Store managers may be making local purchases using personal funds and filing reimbursement claims. This could result in:

- **High Volume:** Potentially 51,000 monthly claims across all locations
- **Delayed Reimbursements:** 2-4 weeks typical processing time
- **Limited Audit Coverage:** Less than 1% of claims may be verified
- **Fraud Exposure:** Fabricated receipts and duplicate claims possible
- **No Real-time Visibility:** Local spend not visible until claim processed

Proposed Solution: Zero-Cash Virtual Card System



System Design

Component	Description
Store Wallet	Each location gets monthly budget divided into category "pockets"
Virtual Card	Single-use card issued on-demand for exact purchase amount
MCC Locking	Card only works at merchant categories matching the pocket
Time Limit	Card expires in 4 hours if unused
Auto-Posting	Transaction posts to SAP immediately with correct GL/cost center
Hash Registry	Prevents duplicate invoice submission for same transaction

```
Example Transaction Flow: 1. Manager requests ₹450 virtual card for "Dairy" category 2. System checks:
Dairy pocket balance ≥ ₹450 → YES 3. Issues single-use card: Amount=₹450, Valid=4hrs, MCC=Grocery 4.
Manager purchases milk at local dairy 5. Transaction captured: Store_101 | Dairy | GL_5100101 | ₹450 6.
Hash generated: DAIRY_101_450_Jan9 7. Auto-posts to SAP within minutes
```

Fraud Prevention Controls

Control	How It Prevents Fraud
MCC Locking	"Dairy" card declined at electronics stores
Exact Amount	₹450 card cannot be used for ₹500
Time Expiry	Card unusable after 4 hours
Hash Registry	Same expense cannot be claimed again via invoice
Real-time Posting	Immediate visibility, no reconciliation gaps

Alternative Approach: Digital Petty Cash with Enhanced Controls

If virtual card infrastructure is not immediately feasible (bank partnerships, regulatory approvals):

- Keep physical imprest but digitize the claim process
- OCR extraction of receipt data via mobile app
- Automated duplicate detection using transaction hashing
- Anomaly detection for spending patterns
- Weekly auto-replenishment of imprest fund

Challenge 3: Non-PO Spend (Rent, Utilities, Services)

Potential Issues Identified

Approximately 30% of expenses may bypass the PO process entirely:

- **Rent Payments:** Could be managed through Excel spreadsheets with manual tracking
- **Utility Bills:** May arrive as physical documents requiring manual processing
- **Recurring Services:** Pest control, maintenance contracts tracked informally
- **No SES Creation:** Service Entry Sheets may not be created before payment

Potential Risks

Risk	Description
Payments to Closed Stores	Rent continues after store closure if not manually stopped
Missed Escalations	Annual rent increases not applied correctly
Accounting Gaps	Missing SES leads to improper expense recognition
Late Payment Penalties	Utility bills processed late incur penalties

Proposed Solution: Contract-Linked Auto-Settlement

For Rent Payments

```
Monthly Automation Logic: ON 1st of every month: FOR each active_contract: CHECK: Is lease still valid?
(end_date > today) IF NO → STOP payment, ALERT procurement CHECK: Is store still operating? IF
store.status = "Closed" → STOP payment, INITIATE deposit recovery CHECK: Is escalation due? IF today >=
next_escalation_date: new_rent = current_rent × (1 + escalation_pct) UPDATE contract CREATE Service
Entry Sheet (SES): - SAC Code: 997212 (Rental Services) - Amount: calculated rent - TDS: Applicable
rate - GL/Cost Center: From store master POST to SAP → TRIGGER bank payment
```

For Utility Bills

Step	Action
1	Connect to utility provider APIs (electricity, water, internet)
2	Auto-pull electronic bills directly from provider systems
3	Validate: Store active, amount within ±20% of historical
4	If valid: Auto-settle via bank API with correct accounting
5	If variance detected: Route to Facilities for review

Alternative Approach: Scheduled Payment with Validation Gates

If full API automation is not immediately possible:

- Weekly payment batch generated by system
- Validation checks run on each payment (store status, contract validity)
- Batch presented to Finance for single-click approval
- Exceptions flagged for individual review

Challenge 4: Cross-System Duplicate Payments

Potential Issues Identified

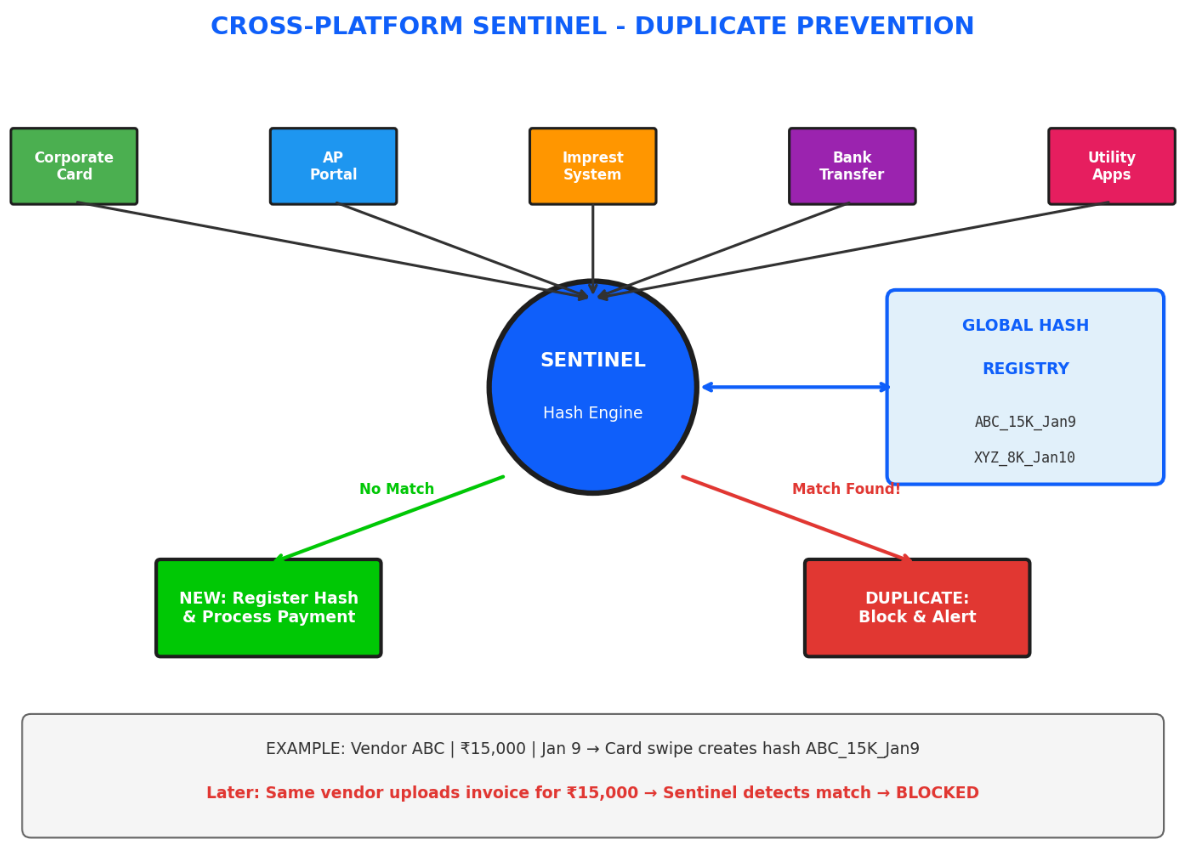
Multiple payment systems (potentially 15+) may exist without real-time integration:

- **SAP:** Primary vendor payments
- **Corporate Cards:** Immediate purchases
- **Imprest System:** Reimbursement claims

- **Utility Portals:** Direct payments
- **Departmental Systems:** Various specialized apps

Risk: Same expense could potentially be paid through multiple channels without detection.

Proposed Solution: Cross-Platform Sentinel


CROSS-PLATFORM SENTINEL - DUPLICATE PREVENTION
Corporate Card
AP Portal
Imprest System
Bank Transfer
Utility Apps
SENTINEL
Hash Engine
GLOBAL HASH
REGISTRY
ABC_15K_Jan9
XYZ_8K_Jan10
No Match
Match Found!
NEW: Register Hash & Process Payment
DUPLICATE: Block & Alert
EXAMPLE: Vendor ABC | ₹15,000 | Jan 9 → Card swipe creates hash ABC_15K_Jan9
Later: Same vendor uploads invoice for ₹15,000 → Sentinel detects match → BLOCKED

How Transaction Fingerprinting Works

```
Hash Generation Formula: transaction_hash = CREATE_HASH( vendor_tax_id, amount (rounded to nearest
100), date (±3 days window), description_keywords ) Matching Logic: Two transactions are "similar" if:
- Same vendor tax ID AND - Amount within ±5% AND - Date within ±3 days
```

Real-Time Protection Flow

Time	System	Event	Sentinel Action
10:00	Corporate Card	₹15,000 swipe at Vendor ABC	Hash Registered
14:00	AP Portal	Vendor uploads ₹15,000 invoice	BLOCKED - Match Found
16:00	Imprest	Manager claims ₹15,000	BLOCKED - Match Found

Alternative Approach: Periodic Reconciliation

If real-time cross-platform integration is technically complex initially:

- Nightly batch extraction from all 15 systems
- Hash generation and duplicate identification

- Exception report for Finance investigation
- Recovery process for confirmed duplicates

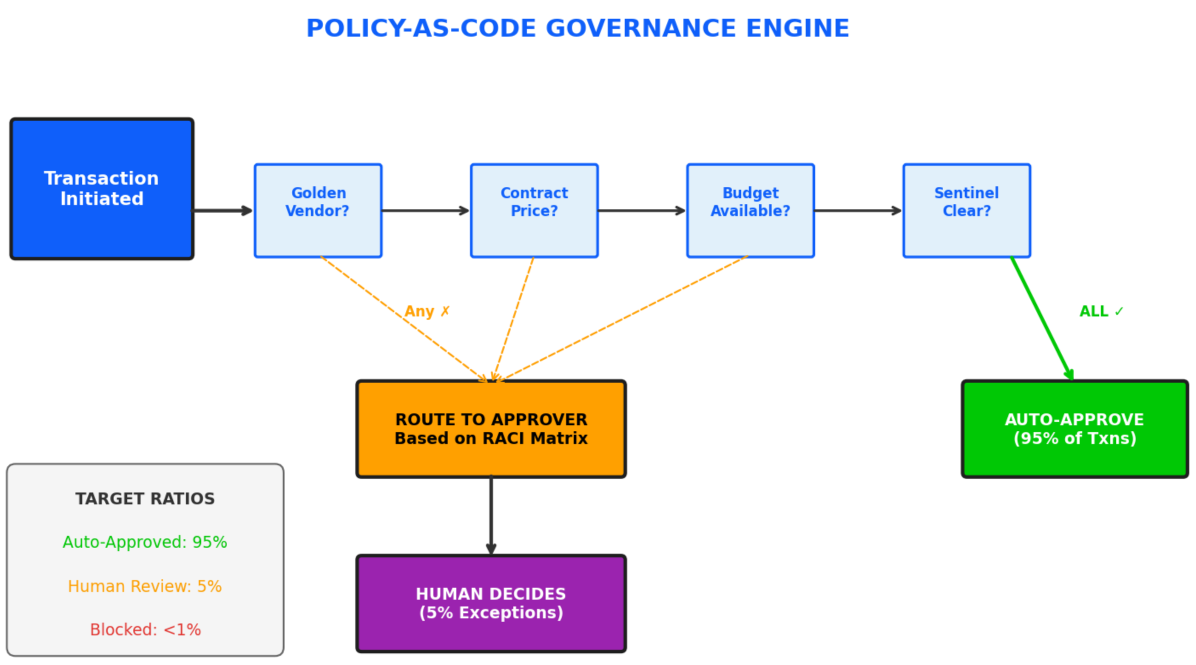
Challenge 5: PO Creation Governance

Potential Issues Identified

PO creation may be fragmented across the organization:

- **Multiple Functions:** Possibly 6+ departments creating POs
- **Many Users:** Could be 100+ people involved, often as secondary responsibility
- **Limited Standardization:** No fixed job descriptions or training requirements
- **Weak Oversight:** Manager-only approval without segregation of duties

Proposed Solution: Policy-as-Code Governance



Auto-Approval Criteria (Green Channel)

A transaction qualifies for automatic approval if ALL conditions are met:

Check	Criteria
Golden Vendor	Vendor exists in approved master with valid status
Contract Price	Price matches negotiated rate in CLM (±0.5%)
Budget Available	Transaction within department/store budget
Sentinel Clear	No duplicate transaction detected
Authority Limit	Amount within requester's approval authority

Escalation Routing Rules

```
Policy-as-Code Examples: IF category = "Kitchen Equipment" AND amount > ₹50,000: Route to: Regional
Maintenance Head + Finance Controller IF category = "Marketing" AND amount > ₹25,000: Route to: Brand
Manager + Marketing Head IF category = "Emergency Repair" AND amount < ₹10,000: Auto-approve IF
budget_available AND golden_vendor IF vendor NOT in Golden Record: Block transaction, require vendor
onboarding
```

Role-Based Access Control

Role	System Access	Capabilities
Store Manager	Guided Buying only	Add to cart, request services, view status
Area Manager	Approvals dashboard	Approve within limits, view regional spend
Category Manager	Contracts, vendors	Negotiate rates, manage vendor performance
Finance Controller	Full visibility	Override, investigate, configure rules

Alternative Approach: Centralized Procurement Desk

If policy-as-code is too complex initially:

- Centralize all PO creation to a trained procurement team (5-10 people)
- Requesters submit through self-service portal
- Procurement team creates POs following standard process
- Approvals still follow hierarchy, but creation is controlled

Implementation Approach

Phased Roadmap

Phase	Duration	Focus Areas	Key Deliverables
1. Foundation	Weeks 1-6	Data extraction, cleanup, infrastructure	Clean vendor master, contract database
2. Quick Wins	Weeks 7-12	Invoice automation, rent pilot	3-way matching, 100-store rent automation
3. Scale	Weeks 13-20	Virtual cards, full rollout	All stores automated, Sentinel active
4. Optimize	Months 6-12	Advanced features, analytics	Full governance, predictive insights

Solution Selection Guide

Challenge	Recommended Solution	Alternative If
100K Invoices	3-Way Match Engine	ERS for top vendors if simpler start needed
51K Imprest	Virtual Cards	Digital petty cash if bank partnership delayed
30% Non-PO	Contract Auto-Settlement	Scheduled batch if APIs unavailable
15 Systems	Cross-Platform Sentinel	Periodic reconciliation initially
Governance	Policy-as-Code	Centralized desk for quick governance fix

Next Steps

1. **Validation:** Confirm assumptions during detailed assessment
2. **Prioritization:** Agree on Phase 1 focus areas based on impact
3. **Data Access:** Obtain sample extracts from key systems
4. **POC Definition:** Define success criteria for pilot programs