BOS GLOBAL

PROJECT KNOWLEDGE SYNTHESIS

Complete Understanding of All 17 Project Documents + Codebase

Version 1.0 | February 2026

"If an action cannot be explained as an event, corrected safely, and audited clearly — it does not belong in BOS."

Sources: 17 Official Documents + Full Codebase Analysis
Scope: Architecture, Engines, Modules, AI, Security, Governance
Classification: CONFIDENTIAL — Architect Eyes Only

1. BOS NI NINI — THE BIG PICTURE

BOS (Business Operating System) ni enterprise platform iliyojengwa juu ya Django, inayotumia event-sourcing architecture. Kila ukweli ndani ya BOS unaandikwa kama immutable event — hakuna direct database writes, hakuna updates, hakuna deletes. System inajumuisha engines 9 za biashara, zinazoongozwa na core kernel yenye hash-chain integrity, deterministic command processing, na engine isolation ya nguvu.

BOS inashughulikia biashara tofauti — retail (supermarket, pharmacy, hardware), restaurant (dine-in, fast food, bar), workshop (windows, doors, panels), na hospitality (rooms). Zote zinatumia primitives moja, events moja, na truth moja.

Falsafa Kuu:

- Event-first, append-only — kila kitendo ni event, hakuna direct DB mutation
- Engine isolation — kila engine inaandika events zake tu, cross-engine ni read-only
- Human-in-control — AI ni advisory tu, haiwezi commit state
- Corrections, not deletions — makosa yanasahihishwa na event mpya, siyo kufuta
- Full auditability — kila kitu kina actor, timestamp, hash, na correlation trail

Architecture ya Msingi:

- Framework: Django (container, siyo authority — BOS architecture ndio boss)
- Pattern: Event Sourcing + CQRS (Command Query Responsibility Segregation)
- Scaling: Golden Image replication — no containers, identical VMs
- Multi-tenant: Logical isolation per business_id, branch_id scope
- Offline-first: Events marked PROVISIONAL, sync preserves order, conflicts = human review

2. CORE KERNEL — MSINGI WA BOS (Phase 0)

Phase 0 ni foundation ambayo engines zote zinasimama juu yake. Imeandikwa, imekaguliwa, na imefungwa (FROZEN). Hakuna breaking changes zinazoruhusiwa — additive only.

2.1 Event Store — Vault ya Ukweli

Event Store ni moyo wa BOS. Kila ukweli unaandikwa hapa kama immutable event yenye fields 17 zilizofungwa (frozen). Events hazifutwi, hazibadilishwi, na hazirekebishwi — corrections ni events mpya zenye correction_of field.

Canonical Event Model (17 Frozen Fields):

Field	Aina	Maelezo
event_id	UUID (PK)	Unique, enforces idempotency
event_type	CharField	engine.domain.action format (e.g. inventory.stock.moved)
event_version	PositiveSmallInt	Schema version ya payload
business_id	UUID	Tenant boundary — always required
branch_id	UUID (nullable)	Branch scope, null = business-wide
source_engine	CharField	Engine iliyoandika event hii
actor_type	Enum	HUMAN | SYSTEM | DEVICE | AI
actor_id	CharField	Unique ID ya actor
correlation_id	UUID	Story/journey grouping — required
causation_id	UUID (nullable)	Event iliyosababisha hii, null = first in chain
payload	JSONField	Versioned event data
reference	JSONField	External refs (receipt ID, PO number)
created_at	DateTime	Wakati event ilitengenezwa
received_at	DateTime (auto)	Wakati Event Store ilipokea
status	Enum	FINAL | PROVISIONAL | REVIEW_REQUIRED
correction_of	UUID (nullable)	Points to event inayosahihishwa
previous_event_hash	CharField(64)	SHA-256 ya event ya awali, GENESIS kwa ya kwanza
event_hash	CharField(64)	SHA-256 ya content + previous hash

Hash-Chain Integrity:

- Formula: event_hash = SHA256(canonical_json(payload) + previous_event_hash)
- First event per business: previous_event_hash = 'GENESIS' (siyo empty string)
- Canonical JSON: sorted keys, no whitespace variability, ensure_ascii=True
- Tamper detection: kama hash moja imebadilika, chain nzima inagundulika

Immutability Guards (Code Level):

- Event.save() — INSERT only, update inaraise PermissionError
- Event.delete() — ALWAYS raises PermissionError, hakuna njia ya kufuta
- Bootstrap checks verify guards zinafanya kazi kila system inapoanza

2.2 ONE Lawful Write Path — persist_event()

Kuna njia MOJA tu ya kuandika event ndani ya BOS. Hakuna shortcut, hakuna backdoor. Flow ni rigid na deterministic:

Hatua	Component	Ikishindwa
1	Validate event structure (schema, actor, context, type, status, correction)	Rejection + explicit code
2	Idempotency check (event_id exists?)	DUPLICATE_EVENT_ID rejection
3	Hash-chain verification (previous hash + computed hash)	HASH_CHAIN_BROKEN rejection
4	Atomic DB save (re-check idempotency + hash inside transaction)	TRANSACTION_ABORTED rejection
5	Dispatch to subscribers AFTER commit (transaction.on_commit)	Logged, never rolls back event
6	Return ValidationResult (accepted/rejected)	N/A

■ **Replay isolation: persist_event() inaraise ReplayIsolationError wakati replay mode iko active**

2.3 Registries Mbili — Tofauti Muhimu

	EventTypeRegistry	SubscriberRegistry
Kazi	Persistence gate — inaruhusu au kukataa event types	Routing table — inaelekeza events kwa handlers
Mahali	core/event_store/validators/	core/events/
Uhusiano	Hakuna cross-calling kati yao	Hakuna cross-calling kati yao
Format	engine.domain.action	engine.domain.action
Thread-safe	Ndiyo (Lock)	Ndiyo (Lock)

2.4 Event Bus (Architect Score: 9.5/10)

- Dispatcher: routes persisted events to registered subscribers, sequential execution
- Subscriber failure NEVER rolls back event, NEVER breaks other subscribers
- Engine isolation: self-subscription blocked unless explicit allow_self_subscription=True
- Post-commit dispatch: events dispatched AFTER DB commit via transaction.on_commit()

2.5 Replay Engine

- Event Store = truth archive, Replay Engine = time machine
- READ only — never writes to Event Store during replay
- ReplayContext activates isolation: is_replay_active() = True, persist_event() BLOCKED
- Modes: full replay, business-scoped, time-scoped, checkpoint resume
- Hash verification: structural (fast) au full recomputation (tamper detection)
- Composite cursor (received_at + event_id) prevents skipping same-timestamp events
- Projection rebuilder: truncate → replay → checkpoint (disposable read models)

2.6 Bootstrap Self-Check

- Runs on startup via AppConfig.ready(), skips during migrate/makemigrations/test
- Check 1: bos_event_store table exists

- Check 2: Immutability guards active (save/delete blocked)
- Check 3: Hash-chain structural integrity (no empty hashes)
- Check 4: EventTypeRegistry importable and instantiable
- Check 5: persist_event importable and callable

■ **Any check fails → SystemBootstrapError → system refuses to start**

2.7 Command Layer & Policy Engine

- Commands: deterministic intent → binary ACCEPTED/REJECTED outcome → event(s)
- Policy Engine: pure rule evaluation, no side effects, version-locked
- Enforcement levels: BLOCK (reject) | WARN (allow + flag) | ESCALATE (→ REVIEW_REQUIRED)

2.8 Phase 0 Gaps (Bado Hazijakamilika)

Component	Hali	Maelezo
Multi-tenant core	Interface tu	BusinessContextProtocol ipo; concrete implementation bado
Identity & Actor	Partial	Actor types zipo kwenye Event Model; full identity mgmt pending
Role-Permission-Scope	Planned	core/security/ directory ipo; implementation pending
Feature flags	Planned	core/config/ directory ipo; implementation pending
Country config loader	Planned	Planned; implementation pending

3. ENGINES 9 ZA BIASHARA

BOS ina engines 9, kila moja ina ownership ya events zake. Cross-engine communication ni read-only kupitia events au queries. Hakuna engine inayoandika data ya engine nyingine.

Engine	Events	Kazi Kuu
Retail	retail.*	POS, basket, pricing, multi-branch, remote cart, tax regime
Restaurant	restaurant.*	Orders, kitchen flow, table sessions, QR menu, split billing
Workshop	workshop.*	Style-driven costing, cutting optimization, offcuts, production
Inventory	inventory.*	Movement-only state, reservations, transfers, count reconciliation
Cash	cash.*	Multi-drawer, shift control, petty cash, cash vs non-cash separation
Accounting	accounting.*	Financial events, management views, tax awareness (not statutory)
Procurement	procurement.*	Request → PO → GRN → Invoice → Payment, approval workflows
Promotion	promotion.*	Rule-based promotions, loyalty, customer-controlled offers
HR	hr.*	Staff, roles, biometric attendance, payroll reference (not statutory)

3.1 Retail Engine — Cash-Flow Machine

- POS hybrid: items + services, measured goods, bundles, variable-price
- Multi-price layers: retail, wholesale, workshop, negotiated — permission-based overrides
- Basket-based POS: counter au remote (shared via link/QR), inventory affected after payment only
- Inventory integration: event-driven, batch/expiry tracking, serial numbers, return handling
- Retail ↔ Workshop bridge: materials sold as retail with optional project reference

3.2 Restaurant Engine — Order-Centric Hospitality

- Menu items = recipes + experiences (not inventory items)
- Order lifecycle: Created → Confirmed → In Preparation → Ready → Served → Closed
- Kitchen Display: real-time queues, modifiers, elapsed time, station routing
- Table sessions: multiple orders + payments, split billing, pay-before/after
- Ingredient consumption: on confirmation au preparation start (configurable)

3.3 Workshop Engine — Style-Driven Production

- Style = single source of truth for design, costing, cutting, ordering, optimization

- Style sources: own creation au supplier-shared (if both have Retail + Workshop plans)
- Default Style: rate-based quotation tu (sqm × rate), cannot enter production
- Shape model: Cut (1D profiles), Fill-Area (glass/panels, 2D), Fill-Cut (mosquito nets)
- Formula engine: width/height formulas per shape, W/H fallback, X/Y/Z reference variables
- Project states: Quote → In Progress (only In Progress generates cut lists)
- Cutting optimization: fundi-first logic — large pieces first, minimize remaining bars
- Material quantities finalized AFTER cut list — reflects real workshop practice

3.4 Inventory Engine — Physical Reality

- Movement-only state: no direct stock mutation — every change is an explicit event
- Events: received, sold, reserved, consumed, returned, damaged, transferred, adjusted
- Location-aware: business + branch + location (store, warehouse, kitchen, workshop floor)
- Reservations reduce available stock without reducing physical stock
- Internal transfers: same business only, full lifecycle (initiate → dispatch → receive)
- Stock count reconciliation: differences = adjustment events, never overwrite

3.5 Cash Management — Operational Cash Truth

- Cash always has location: drawer, safe, mobile cashier, smart box
- Session/shift control: opening balance → movements → closing count → explain differences
- Strict cash vs non-cash separation: prevents leakage hidden in digital totals
- Petty cash: controlled pool + limits + approval + categorization
- All movements immutable — errors corrected through adjustment entries

3.6 Accounting Engine — Management, Not Statutory

- NOT a bookkeeping system — prepares, structures, explains financial truth
- Aggregates financial events from ALL other engines
- Append-only internal ledger, corrections via adjustment events
- Flexible chart of accounts mapping (IFRS or local frameworks)
- Tax-aware but does NOT perform filings
- Designed for integration: exports trial balances, ledger extracts via APIs

3.7 Procurement Engine — Controlled Spending

- Flow: Purchase Request → Approval → PO → Receiving → Payment reference
- No inventory increase until goods physically received
- Role-based approvals, threshold-driven, emergency overrides (fully logged)
- Multi-branch: initiate at branch, approve at HQ, deliver to any branch

3.8 Promotion Engine — Rule-Driven, Customer-Controlled

- Promotions = rule sets that modify pricing transparently
- Composable rules: percentage, fixed, first-N, time window, volume, bundles
- Safety controls: usage limits, budget ceilings, customer caps, stackability rules
- Loyalty: lightweight, customer-controlled — customers choose which offers to activate
- Promotions reduce net price BEFORE tax — VAT on adjusted amount
- No invisible discounts — every application logged with user, time, receipt, rule ref

3.9 HR & Payroll Engine — Light HR

- NOT statutory HR/payroll — links people, roles, attendance to operational truth
- Fingerprint attendance: hashed references only, no raw biometric data stored
- Consent mandatory with timestamp and jurisdiction context
- Devices bound to branches, staff authorized per assignment
- Offline sync supported, marked accordingly for audit integrity
- Payroll figures = reference only, reflected in Cash + Accounting as labour cost events

4. CROSS-CUTTING CONCERNS

4.1 AI & Decision Intelligence

- Advisory ONLY — AI cannot commit state, modify records, or execute operations
- Domain-specific advisors: Finance, Cash Flow, Tax, Operations, Sales, Workforce
- Decision Simulation: what-if projections using real BOS data without committing
- Modes: advisory (suggest), assisted execution (suggest + user approves), limited automation
- Decision Journal: logs assumptions, AI explanations, user consent, timestamps, outcomes
- AI actor_type = 'AI' flagged with advisory_actor=True in ValidationResult

■ **AI CANNOT: sign contracts, borrow funds, pay money, dismiss staff, delete data, alter history**

4.2 Reporting & BI Engine

- Operates on event streams, not static tables
- Semantic layer: raw events → business metrics (revenue, margin, stock turnover, etc.)
- Role-based dashboards, versioned reports and metrics
- Alerts for anomalies: cash leakage, shrinkage, abnormal discounts

4.3 Integration Engine

- Controlled gateway between BOS and external systems
- External systems NEVER write directly into BOS core
- Hardware: fingerprint scanners, POS devices, kiosks, QR scanners
- Software: accounting systems, ERP, CRM, payment gateways, mobile money
- Integration failures isolated — never block core operations

4.4 Global Administration & Governance

- Global: platform rules, pricing, tax, AI governance, security, compliance
- Local: businesses retain full control over operations, staff, customers, decisions
- Business ownership verification, identity checks, duplicate detection, trust scoring
- Global Reference Pricing with hidden region affordability weighting
- Country rules = versioned rule packs in DB, updated without code changes
- GDPR-aligned privacy baseline, data minimization, right-to-access/delete

4.5 Security & Architecture

- Scaling: Golden Image replication (identical VMs), no container orchestration
- State in shared data layers, nodes terminable without data loss
- Tenant isolation: logical boundaries + cryptographic controls
- Offline-first: local queues, auto-sync, conflicts = human review
- Role → Permission → Scope model (business, branch, project boundaries)

5. NON-NEGOTIABLE TECHNICAL CONTRACTS

Kutoka Core Technical Appendix — kanuni hizi zinatumika kwa developer wote, hazivunjiki, hazipunguzwi.

Event Store: Kila ukweli ni event. No deletes, no overwrites. Corrections = new events.

Engine Access: Kila engine inaandika events zake tu. Cross-engine = read-only via events/queries.

Actor Model: Kila event ina actor mmoja. AI = advisory, haiwezi execute operations.

Offline Rules: Events za offline = PROVISIONAL. Sync preserves order. Conflicts = human review.

Permissions: Role = responsibility. Permission = atomic action. Scope = data boundary.

Data Lifecycle: Events hazifutwi. Financial data = immutable. PII masked after retention.

Config vs Data: Country/tax rules = configuration. Operational facts = data. Engines = code.

Migrations: Schema changes additive only. Feature flags control rollout. Rollback by disabling.

Human Review: States: REVIEW_REQUIRED, DISPUTED, CONFIRMED, REJECTED.

6. HALI YA CODEBASE (Sasa Hivi)

Directories zilizo na code halisi:

Path	Status	Content
core/event_store/	COMPLETE	Models, validators, idempotency, hashing, persistence, migrations
core/events/	COMPLETE	Dispatcher, SubscriberRegistry, errors, service (persist_event + dispatch)
core/replay/	COMPLETE	Event replayer, checkpoints, projection rebuilder, context, models, migrations
core/bootstrap/	COMPLETE	Self-check, invariants, errors, app config
config/	COMPLETE	Django settings, URLs, WSGI, ASGI
engines/*/	SCAFFOLDED	9 engines with empty __init__.py, commands/, services/, policies/, events.py, subscriptions.py
projections/*/	SCAFFOLDED	7 projection directories (retail, restaurant, workshop, inventory, finance, bi, guards)
integration/	SCAFFOLDED	Empty adapters, inbound, outbound, permissions, audit_log
ai/	SCAFFOLDED	Empty advisors, decision_simulation, journal, guardrails
interfaces/	SCAFFOLDED	Empty api, admin, ui directories

Known Issue:

■ **Model help_text inasema 'empty string' kwa first event previous_event_hash, lakini GENESIS_HASH='GENESIS' ndio authoritative. Help_text inconsistency noted, bado haijarekebishwa.**

config/settings.py Note:

■ **core.replay haipo kwenye INSTALLED_APPS ya config/settings.py (ipo tu kwenye core/replay/settings.py). Hii inahitaji kuongezwa kabla replay migrations hazijafanya kazi.**

7. ROADMAP — PHASES 0-13

Phase	Name	Status	Scope
0	**Core Kernel**	LOCKED ✓	Event Store, Engine Registry, Commands, Policy, Bus, Replay, Bootstrap
1	**Governance & Policy Hardening**	PLANNED	Engine contract freezing, cross-engine policy, legal proof infrastructure
2	**Global Compliance Engine**	PLANNED	Zero code-per-country, compliance profiles, DSL, validation engine
3	**Document Engine**	PLANNED	Templates, blocks, HTML/PDF renderers, numbering, verification portal
4	**Business Primitive Layer**	PLANNED	Ledger, Item, Actor, Document, Inventory Movement, Obligation, Approval, Workflow
5	**Enterprise Engines**	PLANNED	Accounting, Inventory, Procurement, Cash, BI, HR completion
6	**Vertical Modules**	PLANNED	Retail POS, Restaurant, Workshop, Rooms/Hospitality
7	**AI & Decision Intelligence**	PLANNED	Advisory AI, decision simulation, guardrails
8	**Security & Isolation**	PLANNED	Encryption, permissions, sandbox, leak detection, audit explorer
9	**Integration Layer**	PLANNED	API gateway, webhooks, payment/bank adapters, EFD/tax adapters
10	**Performance & Scale**	PLANNED	Partitioning, snapshots, projections, horizontal scaling, caching
11	**Enterprise Admin**	PLANNED	Admin console, compliance management, policy simulation, disaster recovery
12	**SaaS Productization**	PLANNED	Tenant onboarding, subscriptions, feature flags, white-label
13	**Documentation & Certification**	PLANNED	Developer handbook, governance manual, compliance guide, certification

Execution Doctrine:

- **Build from truth → operations → insight → intelligence → scale**
- **No phase skipped, merged, or rushed**
- **No breaking changes — additive only**

8. UHUSIANO MUHIMU KATI YA ENGINES

Retail Sale Flow (Mfano wa Event Chain):

- Customer anachagua bidhaa → Retail basket created
- Payment completed → retail.sale.completed event emitted
- Inventory subscribes → inventory.stock.moved event (stock reduction)
- Cash subscribes → cash.received event (cash in drawer)
- Accounting subscribes → accounting.entry.recorded (revenue + tax)
- Promotion (if applied) → promotion.applied event (discount audit trail)
- BI reads all events → updates dashboards, metrics, alerts

✔ Kila engine inaandika events zake tu. Hakuna engine inayomodify data ya nyingine.

Workshop Production Flow:

- Style selected → dimensions entered → project created (Quote state)
- Customer approves → project moves to In Progress
- Formula engine evaluates shapes → generates cut list per item
- Cut list aggregated at project level by material/brand/profile
- Material quantities determined FROM cut list (not raw formulas)
- Inventory consumption events emitted based on cut list execution
- Procurement triggered for materials not in stock

Restaurant Order Flow:

- Order placed (counter, QR, kiosk) → items enter lifecycle
- Confirmed → routed to correct kitchen/bar station
- Ingredient consumption on confirmation or prep start (configurable)
- Ready → Served → table session payment → Closed
- Cash/card recorded, accounting updated, inventory decremented

Engine Isolation Rules:

- Each engine writes ONLY its own events (source_engine field)
- Cross-engine communication: read-only via event subscriptions
- BI and AI NEVER write operational data
- SubscriberRegistry blocks self-subscription unless explicitly allowed
- Projections are disposable — rebuildable from events anytime

SYNTHESIS COMPLETE

Hii document inajumuisha maarifa yote kutoka project documents 17 na codebase analysis. BOS ni event-sourced enterprise platform yenye frozen kernel, engines 9, na roadmap ya phases 14 kutoka foundation hadi SaaS productization.