# BOS GLOBAL

## MASTER IMPLEMENTATION ROADMAP

*Phase-Based Execution Guide*

*Version 1.0.1 | February 2026*

*"BOS must be built from truth to operations, then insight, then intelligence, and finally scale. No phase should be skipped, merged, or rushed."*

— Final Execution Doctrine

Architecture: Event-Sourced, Append-Only, Engine-Isolated
Framework: Django | Phases: 0-13 | Engines: 9
Classification: CONFIDENTIAL — Architect Eyes Only

# TABLE OF CONTENTS

## PHASE 0 CORE KERNEL (LOCKED)

COMPLETE

The foundational layer upon which all BOS engines operate. Phase 0 establishes the immutable truth infrastructure, deterministic command processing, and engine isolation contracts. No breaking changes permitted — additive only.

### Completed Components

- Deterministic Command → Outcome → Event chain (binary ACCEPTED/REJECTED)
- Immutable Event Store with SHA-256 hash-chain integrity (GENESIS_HASH)
- Canonical Event Model — 17 frozen fields with first-class correlation_id / causation_id
- Engine Registry — registration, locking after bootstrap, thread-safe enforcement
- Engine Contracts — ownership rules, subscription isolation
- Event Bus — dispatcher, SubscriberRegistry, post-commit dispatch (Architect: 9.5/10)
- Policy Engine — pure deterministic rules, graduated enforcement (BLOCK/WARN/ESCALATE)
- Bootstrap Self-check — runs on startup, skip during migrate/makemigrations
- Replay Engine — implemented, awaiting architect scoring
- ONE lawful write path: validate → idempotency → hash → atomic save → on_commit dispatch

### Remaining Foundation (Phase 0 Gaps)

- Multi-tenant core — BusinessContextProtocol interface exists; concrete implementation pending
- Identity & Actor model — actor types in Event Model; full identity management pending
- Role-Permission-Scope engine — core/security/ directory planned; implementation pending
- Feature flags — core/config/ planned; implementation pending
- Country configuration loader — planned; implementation pending

**✔ Outcome: BOS kernel is frozen and replay-safe. Foundation gaps must be resolved before Phase 1.**

## PHASE 1 GOVERNANCE & POLICY HARDENING

PLANNED

### 1.1 Engine Contract Freezing

- Versioned engine contracts
- Engine capability manifest
- Engine compatibility checks

### 1.2 Cross-Engine Policy Layer

- Central Policy Engine
- Conflict detection
- Policy override (tenant level)
- Compliance binding

### 1.3 Legal Proof Infrastructure

- Event chain verification API
- Snapshot notarization option
- Admin action journaling
- Court-admissible export format

**✔ Outcome: BOS becomes legally defensible globally.**

# PHASE 2 GLOBAL COMPLIANCE ENGINE

PLANNED

**Zero code-per-country principle.** No `if country ==` logic in code.

## 2.1 Compliance Profile System (Admin-Configurable)

- Country, tax regime, required document fields
- Ledger requirements, reporting format
- QR requirements, fiscal numbering rules

## 2.2 Compliance DSL

- Required/forbidden block rules
- Tax computation rule definitions
- Numbering constraints

## 2.3 Compliance Validation Engine

- Template validation
- Transaction validation
- Report validation

✔ **Outcome: Global-ready without code forks.**

## PHASE 3 DOCUMENT ENGINE (STRICT BUILDER) PLANNED

### 3.1 Template Engine

- Structured JSON layout schema
- Versioning with hash integrity
- Locking mechanism

### 3.2 Block System

- Blocks: Header, Party, Meta, Item Table, Totals, Payment, Compliance, Notes, QR
- User can toggle, reorder (within constraints), activate per branch

### 3.3-3.4 Renderers

- HTML Preview — safe component-based, deterministic, escaped
- PDF Official — derived from same render source, hash stored, immutable artifact

### 3.5 Numbering Engine

- Prefix/suffix, fiscal reset, branch-level
- Compliance override, deterministic event-based numbering

### 3.6 Verification Portal

- Enter doc number, validate hash, show status (valid/tampered)

✔ **Outcome: Enterprise-grade document system.**

## PHASE 4 BUSINESS PRIMITIVE LAYER PLANNED

Universal, reusable event-driven primitives that all vertical modules share:

- Ledger Primitive
- Item/Service Primitive
- Actor Primitive
- Document Primitive
- Inventory Movement Primitive
- Obligation Primitive
- Approval Primitive
- Workflow Primitive

✔ **Outcome: Vertical modules reuse same primitives — no duplication.**

## PHASE 5 ENTERPRISE ENGINES COMPLETION PLANNED

### 5.1 Accounting Engine

- Double entry
- Auto journal from events
- Ledger replay verification
- Trial balance invariants

### 5.2 Inventory Engine

- Movement-only state (no direct stock mutation)

- FIFO/LIFO strategy plugin
- Offcut tracking (for workshop)

### 5.3 Procurement Engine

- Requisition → PO → GRN → Invoice → Payment

### 5.4 Cash Management Engine

- Multi-drawer, shift tracking
- Reconciliation engine

### 5.5 Reporting & BI Engine

- Event-driven projections
- Snapshot reporting
- KPI calculators

### 5.6 HR & Payroll Engine

- Role mapping
- Payroll ledger integration
- Permission binding

✔ **Outcome: Full enterprise core capability.**

## PHASE 6 VERTICAL MODULES

PLANNED

### 6.1 Retail Module

- POS hybrid (item + service)
- Remote item list
- Multi-branch stock
- Tax regime enforcement
- Promotion Engine integration

### 6.2 Restaurant Module

- Table mapping with QR per table
- Self-service ordering
- Kitchen workflow engine
- Split billing
- Order state machine

### 6.3 Workshop Module (Windows, Doors, Panels)

- Style-driven costing
- Supplier style inheritance (if subscribed)
- Cutting optimization engine (line + area)
- Offcut reuse logic
- Production lifecycle tracking
- Material consumption events

### 6.4 Rooms / Hospitality

- QR per room
- Service workflow
- Room billing integration

**✔ Outcome: Business-specific modules on shared primitives.**

## PHASE 7 AI & DECISION INTELLIGENCE

PLANNED

### 7.1 Advisory AI

- Cost anomaly detection
- Profit simulation
- Risk alerts

### 7.2 Decision Simulation Engine

- What-if projections
- Inventory stress simulation
- Cash flow forecast

### 7.3 Guardrails (Non-Negotiable)

- AI cannot commit state
- All actions through command pipeline
- Full journaling

**✔ Outcome: AI-enhanced but safe BOS. Human-in-control always.**

## PHASE 8 SECURITY & ISOLATION

PLANNED

- Tenant-level encryption
- Role-based permission matrix
- Engine sandbox enforcement
- Cross-tenant leak detection
- Audit explorer
- Suspicious pattern detection

## PHASE 9 INTEGRATION LAYER

PLANNED

- API Gateway Engine
- Webhook framework
- Payment adapters, bank adapters
- EFD / Tax authority adapters
- Data export formats
- Integration only through Integration Engine

## PHASE 10 PERFORMANCE & SCALE

PLANNED

- Event store partitioning
- Snapshot acceleration
- Read model projections
- Horizontal scaling
- Caching layer for preview
- Background PDF generation workers

## PHASE 11 ENTERPRISE ADMINISTRATION

PLANNED

- Global admin console
- Compliance profile management
- Policy simulation UI
- Engine state visualizer
- Disaster recovery tools
- Template library marketplace

## PHASE 12 SAAS PRODUCTIZATION

PLANNED

- Tenant onboarding automation
- Subscription plans & feature flags
- Plan-based engine activation
- White-label branding
- Regional configuration packs (admin only)

## PHASE 13 DOCUMENTATION & CERTIFICATION

PLANNED

- Developer handbook
- Engine development guide
- Governance manual
- Compliance configuration guide
- Disaster recovery manual
- Certification path

## FINAL EXECUTION DOCTRINE

■ **BOS must be built from truth to operations, then insight, then intelligence, and finally scale.**

■ **No phase should be skipped, merged, or rushed.**

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

■ **Rule: No breaking changes. Additive only.**