THE MESOCRATIC PARTY | TECHNOLOGY WHITE PAPER

The Mesocratic Compliance Engine

A Modern, Open-Source Platform for FEC Compliance, Campaign Finance Reporting, and Political Technology

Build it once. Open it to everyone. Set the standard.

Published by the Mesocratic National Committee -- March 2026 (v2)

Table of Contents

Executive Summary

Every political committee in America faces the same problem: the technology available for campaign finance compliance is outdated, expensive, closed, and hostile to

integration. The FEC's own free software, FECFile, is a Windows-only desktop application last meaningfully updated in 2018. The leading third-party platforms -- ISPolitical, NGP VAN, Aristotle -- are closed-source, proprietary systems with dated APIs, opaque pricing, and no interoperability. The FEC's public API is read-only. There is no modern, open, programmable platform for political compliance.

The Mesocratic Compliance Engine (MCE) changes this. MCE is a comprehensive, API-first platform for FEC compliance, campaign finance reporting, contribution tracking, disbursement management, and .fec file generation. It is designed as three things simultaneously:

- An internal compliance platform for the Mesocratic National Committee, replacing the need for ISPolitical or any third-party compliance vendor
- A public, RESTful API hosted at developer.mesocrats.org, available to any political committee, campaign, or civic technology organization in America
- An open-source project, released under the MIT license, that any organization can self-host, fork, or contribute to

MCE is not a campaign management platform. It is not a CRM. It is not a fundraising tool. It is the compliance and reporting layer -- the single most critical and least innovated piece of political technology infrastructure in the country.

UPDATE (March 2026): The MCE public API is now live at developer.mesocrats.org/api/v1/ with 14 operations across 8 resource paths. External developers can create API keys, bind them to committees, and manage the full contribution-to-report lifecycle. Multi-tenant architecture with committee-scoped API keys, FEC limit enforcement, aggregate tracking, itemization detection, and immutable audit logging are all operational. End-to-end testing has been completed successfully.

Important architectural note: MCE operates as two independent layers. The internal MCE (src/lib/mce/, ~1,700 lines of TypeScript) handles the MNC's own compliance -- FEC Form 3X report generation (JSON, CSV, .fec, 8872 XML formats), Stripe fee auto-capture, disbursement tracking, contributor matching, and best-efforts follow-up emails. These are served at /api/mce/* and secured by MCE_REPORT_API_KEY. The public PartyStack API at /api/v1/* is a separate multi-tenant system for external developers with its own database, auth layer, and endpoint set. Both layers share the same compliance logic and FEC rules but serve different audiences.

1. The Problem: Political Technology Is Broken

1.1 The ISPolitical Reality

ISPolitical is the most commonly recommended compliance platform for small and mid-sized political committees. It offers FEC and state-level reporting, contribution tracking, and a basic REST API. It is also a closed, proprietary system with significant limitations:

- The API uses HTTP Basic Authentication -- a method deprecated by every modern API security standard in favor of OAuth 2.0 or API key + HMAC
- The API is write-only for most practical purposes: you can submit contributions, but querying, filtering, and bulk export are severely limited
- The platform dropped its direct Stripe integration several years ago, requiring manual CSV imports or batch uploads for organizations using modern payment processors
- Pricing is opaque and per-committee, creating cost uncertainty for organizations managing multiple entities
- There is no webhook system, no event-driven architecture, no real-time data synchronization
- The documentation consists of help articles with jQuery sample code -- not a versioned API reference with OpenAPI/Swagger specifications

ISPolitical serves a purpose. But it is not the platform that modern political technology demands.

1.2 The FECFile Dead End

The FEC provides FECFile, a free Windows desktop application for generating and submitting electronic filings. It has several critical problems:

- Windows-only (requires Windows 8, 10, or 11 and Java 8)
- No API, no programmatic access, no automation
- Manual data entry or bulk import via a fragile CSV format
- No cloud access, no collaboration, no audit trail
- The FEC is piloting a web-based replacement (FECFile Online), but it remains in early testing and cannot accept real filings

FECFile is a filing tool, not a compliance platform. It has no contribution limit enforcement, no aggregate tracking, no donor matching, and no integration capability.

1.3 What Doesn't Exist

What did not exist -- until MCE -- is a modern, open, API-first platform that:

- Accepts contribution and disbursement data via a documented REST API
- Enforces FEC contribution limits and prohibited source rules in real time
- Tracks donor aggregates across calendar years with fuzzy name/address matching
- Generates FEC Form 3X, Schedules A/B/E, and all supporting schedules
- Exports validated .fec files ready for electronic submission
- Generates IRS Forms 8872 and supports Form 1120-POL preparation
- Integrates with accounting platforms like QuickBooks for double-entry bookkeeping
- Provides webhooks for real-time event notification
- Is open-source, self-hostable, and free

That is what MCE is.

2. The Vision: developer.mesocrats.org

2.1 The Developer Portal (LIVE)

MCE is accessible through a public developer portal at developer.mesocrats.org. The portal is live and includes:

- Landing page with the PartyStack value proposition and product catalog
- Interactive API Reference with request/response examples for all 14 live endpoints
- OpenAPI 3.1.0 specification at /openapi.json (865 lines, importable to Postman/Swagger)
- API key management dashboard: create, view (prefix-only), and revoke keys
- Sandbox page with accurate example responses for every endpoint
- Prompt Library with 6 curated templates for building with AI assistants
- Products page showing live vs. coming-soon API products
- Authentication via GitHub and Google OAuth (Supabase Auth)

2.2 Open-Source Commitment

The MCE codebase will be released under the MIT license on GitHub. This means:

- Any political committee, campaign, or civic tech organization can use it for free
- Organizations can self-host the entire platform on their own infrastructure
- Contributors can submit improvements, bug fixes, and new features
- The compliance logic is auditable -- anyone can verify that the rules are implemented correctly

The Mesocratic Party maintains the hosted version at developer.mesocrats.org as a free public service. Organizations that want managed hosting, support, or custom integrations may opt into paid tiers in the future, but the core platform and API will always be free and open.

2.3 Who This Serves

MCE is designed for:

- New political parties and committees that need compliance infrastructure from day one
- Small campaigns that cannot afford ISPolitical, NGP VAN, or professional treasurers
- Civic technology organizations building voter engagement or donation platforms
- Political consultants and treasurers who manage compliance for multiple committees
- Journalists and researchers who want transparent, auditable compliance data
- Any organization that believes political technology should be open, not gatekept

3. System Architecture

3.1 Architecture Overview

MCE is a modular, API-first system built on four layers:

Layer	Description
Data Layer	PostgreSQL database (Supabase) storing all contributions, disbursements, contributors, and compliance state. Dedicated Supabase project for multi-tenant data isolation.
Compliance Engine	Business logic for validation, limit enforcement, aggregate tracking, prohibited source detection, and best-efforts follow-up
Report Generator	Modules for generating FEC Form 3X, all schedules, .fec file export, IRS Form 8872, and PDF summaries
API Layer	RESTful API at /api/v1/* with Bearer token authentication, bcrypt key hashing, committee-scoped access, and audit logging

3.2 Technology Stack

Component	Technology
Runtime	Node.js 20+ / TypeScript
Framework	Next.js 14 API Routes (unified: internal + public API in same codebase)
Database (Main)	PostgreSQL via Supabase -- auth, developer portal, API key storage
Database (API)	PostgreSQL via Supabase (separate project) -- multi-tenant committee data, 7 tables
Authentication	Bearer token (API keys, bcrypt-hashed at cost factor 12). One key = one committee.
File Generation	Custom .fec format generator (planned); PDF via @react-pdf/renderer
Accounting	QuickBooks Online API (planned)
Hosting	Vercel (web + API routes); Supabase (database); GitHub (source)
API Spec	OpenAPI 3.1.0 (865 lines at public/openapi.json)

3.3 Data Flow

The primary data flow for an online donation follows this path:

1. Donor submits contribution via mesocrats.org/donate
2. Stripe processes payment; webhook confirms charge
3. MCE Compliance Engine validates: limits, prohibited sources, certifications
4. Contribution record created in PostgreSQL with all FEC-required fields
5. Aggregate totals updated for the contributor (upsert on aggregates table)

6. If aggregate crosses $200 threshold, contributor flagged for itemization (itemized: true)

7. Audit log entry created (fire-and-forget) with action, user_id, ip_address, new_value

8. At reporting time: Report Generator queries database, builds Form 3X and schedules, exports .fec file

9. Treasurer reviews, validates, and submits .fec file to FEC

3.4 Security Model

- All API access over HTTPS (TLS 1.3)
- Bearer token authentication with bcrypt-hashed API keys (cost factor 12)
- Committee-scoped access: each API key is permanently bound to one committee. All queries filtered by committee_id.
- Row-level security in Supabase: defense-in-depth alongside application-level scoping
- Audit log for every data mutation (who, what, when, from where)
- Rate limiting: 100 requests/minute per API key (documented; enforcement planned)
- 3-year retention enforced per 11 CFR 102.9(c)

4. Database Schema

4.1 Core Tables (LIVE)

The MCE database consists of seven core tables, deployed in a dedicated Supabase project for multi-tenant isolation. All tables have RLS enabled. The schema is defined in src/app/(developer)/api/v1/schema.sql.

Table	Purpose	Key Fields
committees	Committee registration (FEC Form 1 data)	id, owner_user_id, name, legal_name, fec_id, ein, committee_type, treasurer_name, treasurer_address, mailing_address, filing_frequency
contributors	Unique contributor records with dedup keys	id, committee_id, entity_type, first_name, last_name, full_name, email, address fields, employer, occupation, match_key
contributions	Every incoming contribution with FEC metadata	id, committee_id, contributor_id, amount_cents, date_received, contribution_type, payment_method, attestation booleans (5), aggregate_ytd_cents, itemized, report_id
disbursements	Every outgoing expenditure	id, committee_id, payee_name, payee_address, amount_cents,

		date, purpose, category, check_number, receipt_url, report_id
aggregates	YTD totals per contributor/committee/year	contributor_id, committee_id, calendar_year, total_cents, contribution_count, last_contribution_date, itemization_required. UNIQUE constraint.
reports	Generated FEC reports and status	id, committee_id, report_type, coverage_start, coverage_end, filing_deadline, status (draft/review/submitted/accepted), fec_file_path
audit_log	Immutable mutation log	id, committee_id, user_id, action, table_name, record_id, old_value (JSONB), new_value (JSONB), ip_address, created_at

4.2 Contributor Matching

FEC reporting requires aggregating contributions from the same source. MCE implements a multi-signal matching algorithm:

- Primary match: normalized last_name + zip_code (the "match_key" field, auto-generated on contributor creation as lowercase last_name + "_" + zip_code)
- Secondary match: Levenshtein distance on full_name (threshold: 2 edits) + address similarity (planned)
- Tertiary match: email address exact match (planned)
- Manual override: treasurer can merge or split contributor records via the admin interface (planned)

When a potential match is detected, the system flags it for treasurer review rather than auto-merging, preserving data integrity.

4.3 Aggregate Tracking (LIVE)

The aggregates table is upserted transactionally with every new contribution via POST /api/v1/contributions. This ensures:

- Contribution limits can be checked against current aggregates before accepting a new contribution
- The $200 itemization threshold is enforced in real time (itemized flag set to true when aggregate exceeds 20,000 cents)
- Aggregate data is returned in the contribution creation response for immediate client-side feedback: { aggregate: { calendar_year, total_cents, contribution_count, itemization_required } }
- Calendar year boundaries are respected: aggregates reset each January 1

5. Compliance Engine

5.1 Contribution Validation (LIVE)

Every contribution submitted through the API is validated against:

- Required fields: contributor_id, amount_cents, date_received, contribution_type
- Valid date format (YYYY-MM-DD)
- Positive amount (amount_cents > 0)
- Valid contribution type: individual, committee, party, other
- Valid payment method: credit_card, check, wire, cash, other
- Five attestation booleans required for individual contributions: citizenship_attested, personal_funds_attested, non_contractor_attested, personal_card_attested, age_attested

5.2 Limit Enforcement (LIVE)

MCE enforces FEC contribution limits by committee type for the 2025-2026 cycle:

Committee Type	Individual Limit/Year	Cents
national_party	$41,300	4,130,000
state_party	$10,000	1,000,000
pac	$5,000	500,000
candidate	$3,300	330,000
super_pac	No limit	null (unlimited)
hybrid_pac	No limit	null (unlimited)

When a contribution would cause the contributor's aggregate to exceed the applicable limit, the API rejects it with a 422 status code and a clear error message specifying the limit, current aggregate, and excess amount.

Limits are returned via GET /api/v1/compliance/limits, which includes the committee type, cycle, per-source limits, and the $200 itemization threshold (20,000 cents).

5.3 Prohibited Source Detection

MCE validates the five FEC-required attestations on every individual contribution:

- U.S. citizenship or permanent residency (52 U.S.C. 30121)
- Personal funds, not corporate or union funds (52 U.S.C. 30118)
- Not a federal government contractor (52 U.S.C. 30119)
- Personal credit/debit card, not corporate card
- At least 18 years of age

Contributions missing any required attestation are rejected at the API level before the record is created.

5.4 Best Efforts Tracking

Federal election law (52 U.S.C. 30104(b)(3)(A)) requires committees to use "best efforts" to collect employer and occupation from contributors whose aggregate contributions exceed $200 in a calendar year. MCE supports this by:

- Automatically flagging contributors whose aggregate crosses the itemization threshold
- Exposing contributor records via GET /api/v1/contributors with employer/occupation fields visible
- The Prompt Library includes a template ("Set Up Best-Efforts Follow-Up") for building an automated email follow-up system using the API

Webhook-triggered follow-up is documented but not yet implemented. The current approach uses polling via the contributors endpoint.

5.5 Disbursement Tracking (LIVE)

MCE tracks all committee disbursements with FEC-required fields:

- Payee name and address
- Amount in cents
- Date
- Purpose description
- Category: operating, contribution_to_candidate, independent_expenditure, coordinated_expenditure, other
- Optional: check number, receipt URL

Categories are validated at the API level. Invalid categories are rejected with a 400 error.

6. FEC Report Generation

6.1 Form 3X Generation

MCE generates quarterly FEC Form 3X (Report of Receipts and Disbursements) reports. The current implementation creates draft report records via POST /api/v1/reports with coverage period and filing deadline. The report includes:

- report_type: quarterly (monthly and pre-election planned)
- coverage_start and coverage_end dates
- filing_deadline
- status: draft (review, submitted, accepted statuses to follow)

Full Form 3X generation with Detailed Summary Page, Schedule A, Schedule B, and .fec file export is planned for the next development phase.

6.2 Schedule A (Itemized Receipts)

Schedule A lists all contributions from individuals or entities whose aggregate exceeds $200. MCE identifies itemized contributions automatically via the aggregate tracking

system. Each contribution record includes an "itemized" boolean field set at creation time.

6.3 Schedule B (Itemized Disbursements)

Schedule B lists all disbursements to payees receiving aggregate payments exceeding $200. Disbursement records are queryable via GET /api/v1/disbursements with date range and category filters.

6.4 .fec File Format Export

The .fec file format is the FEC's proprietary electronic filing format. MCE will generate validated .fec files from report data. This capability is designed but not yet wired to the public API. Report records (POST /api/v1/reports) create drafts; .fec generation will be added in a subsequent release.

6.5 Validation and QA

MCE's validation suite (planned) will include:

- Field-level validation: required fields, valid ranges, correct formats
- Cross-reference validation: aggregate totals on Schedule A match calculated aggregates; summary page totals match schedule totals
- Limit validation: no reported contribution exceeds applicable limits
- Format validation: .fec file parsed and re-validated against FEC specification
- Comparison with FECFile: during initial deployment, every MCE-generated .fec file will be imported into FECFile for independent validation

7. IRS Reporting

7.1 Form 8872 Generation

Section 527 political organizations must file Form 8872 (Political Organization Report of Contributions and Expenditures) with the IRS. MCE will generate this form by:

- Querying contributions and expenditures for the applicable period
- Applying IRS-specific itemization thresholds (which differ from FEC thresholds)
- Formatting output for IRS electronic filing at irs.gov/polorgs
- Supporting quarterly, monthly, or semiannual schedules as elected by the committee

7.2 Form 1120-POL Support

MCE provides data exports to support preparation of Form 1120-POL (U.S. Income Tax Return for Certain Political Organizations), due annually on April 15. While the form itself is typically prepared by an accountant or tax professional, MCE provides:

- Total exempt function income (contributions) for the tax year
- Total taxable income (investment income, interest, etc.)
- Total expenditures by category
- All data exportable in CSV and PDF formats for accountant handoff

8. The MCE Public API (LIVE)

8.1 API Design Principles

Note: This section documents the public PartyStack API at /api/v1/. The internal MCE endpoints at /api/mce/* (report generation, disbursements) predate the public API and are documented separately. They are secured by MCE_REPORT_API_KEY and serve only the MNC's internal operations.*

The MCE API follows modern REST conventions:

- RESTful resource-oriented URLs: /api/v1/contributions, /api/v1/disbursements, /api/v1/reports
- JSON request and response bodies
- Standard HTTP methods: GET and POST (PUT, PATCH, DELETE planned)
- Consistent response format: { data } for success, { error } for failures
- Offset-based pagination with page, limit, total, and total_pages
- Comprehensive audit logging for all POST operations
- OpenAPI 3.1.0 specification at /openapi.json

8.2 Authentication (LIVE)

MCE uses Bearer token authentication with API keys:

- Developers sign in to developer.mesocrats.org via GitHub or Google OAuth
- API keys are created in the dashboard. Format: mce_live_ + 32 hex characters
- Keys are bcrypt-hashed (cost factor 12) on creation and stored hashed. The raw key is shown exactly once.
- On each request, the first 12 characters (prefix) are used to look up the key, then bcrypt.compare verifies the full key
- Each key is permanently bound to one committee via POST /api/v1/committees. After binding, all data operations are scoped to that committee.
- Authorization header format: Bearer mce_live_<32 hex chars>

There is no support for Basic Auth, query-string tokens, or other deprecated methods. All access is over HTTPS.

8.3 Live Endpoints (14 Operations)

Method	Endpoint	Auth	Description
GET	/api/v1/health	None	Status, version, timestamp, database connectivity
GET	/api/v1/committees	Key (any)	Returns committee bound to this API key
POST	/api/v1/committees	Key (unbound)	Creates committee, permanently binds key. Dual-database write.

GET	/api/v1/contributors	Key (bound)	Paginated list. ILIKE search on name/email via ?search= param.
POST	/api/v1/contributors	Key (bound)	Create contributor. Auto-generates match_key and full_name.
GET	/api/v1/contributions	Key (bound)	Paginated. Filters: date range, contributor_id, itemized, contribution_type.
POST	/api/v1/contributions	Key (bound)	Record contribution. FEC limit enforcement, aggregate upsert, itemization detection.
GET	/api/v1/contributions/:id	Key (bound)	Single contribution by ID. 404 if not found or wrong committee.
GET	/api/v1/disbursements	Key (bound)	Paginated. Filters: date range, category.
POST	/api/v1/disbursements	Key (bound)	Record disbursement. Category validation.
GET	/api/v1/compliance/limits	Key (bound)	FEC limits by committee type. Returns limits in cents + itemization threshold.
GET	/api/v1/reports	Key (bound)	List reports. Filters: status, report_type.
POST	/api/v1/reports	Key (bound)	Create draft FEC report with coverage period and filing deadline.

All endpoints return consistent JSON. Errors include machine-readable error messages. POST operations create audit log entries.

8.4 Webhooks (Planned)

MCE will dispatch webhooks for key events:

- contribution.created -- a new contribution has been recorded and validated
- contribution.flagged -- a contribution has been flagged for review
- disbursement.created -- a new disbursement has been recorded
- report.generated -- a report draft is ready for review

- aggregate.threshold_crossed -- a contributor has crossed the $200 itemization threshold

Webhooks will be delivered via HTTP POST with HMAC-SHA256 signatures for verification. This feature is documented in the OpenAPI spec but not yet implemented.

8.5 Rate Limiting and Versioning

Rate limiting: 100 requests per minute per API key. Rate limit headers (X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset) are documented but enforcement is not yet active.

Versioning: API versions are specified in the URL path (/v1/). Breaking changes only occur in new major versions. Deprecated versions will receive 12 months of support after a new version is released.

9. Accounting Integration

9.1 QuickBooks Online Integration (Planned)

MCE will integrate with QuickBooks Online (QBO) via the Intuit Developer API for double-entry bookkeeping. This ensures that every FEC-reportable transaction is also properly recorded in the committee's financial books. Integration points:

- Contributions: recorded as income (credit to Contributions Revenue, debit to Bank Account)
- Stripe fees: recorded as operating expenses (debit to Processing Fees, credit to Bank Account)
- Disbursements: recorded as expenses (debit to appropriate expense category, credit to Bank Account)
- Refunds: recorded as contra-revenue (debit to Contributions Revenue, credit to Bank Account)

9.2 Chart of Accounts

MCE will provision a recommended Chart of Accounts in QuickBooks tailored to political committee operations.

9.3 Reconciliation

MCE will provide a monthly reconciliation workflow comparing total contributions and disbursements in MCE to total deposits and payments in QBO, flagging discrepancies for treasurer review.

10. Developer Portal: developer.mesocrats.org (LIVE)

10.1 Portal Architecture

The developer portal is integrated into the mesocrats.org Next.js codebase under the (developer) route group. It is served at the developer.mesocrats.org subdomain via middleware rewrite. The portal includes:

- Landing page with PartyStack branding and product catalog
- Products page with live/coming-soon feature status
- Interactive API Reference with all 14 endpoints, request/response schemas, and code examples
- Sandbox page with accurate example responses for every endpoint and method
- Prompt Library with 6 curated AI prompt templates for common workflows
- SDKs page with TypeScript SDK documentation (generation from OpenAPI spec planned)
- Community page with discussion and support links
- Sign In via GitHub and Google OAuth (Supabase Auth, production mode)
- Dashboard with API key management: create (shown once), view prefix, revoke

10.2 Documentation

Documentation is structured around the developer journey:

- API Reference: every endpoint with request/response schemas and examples
- OpenAPI 3.1.0 spec at /openapi.json -- importable to Postman, Swagger UI, or any OpenAPI tool
- Prompt Library: 6 templates covering donation forms, quarterly reports, best-efforts follow-up, treasurer dashboards, limit tracking, and committee setup
- Sandbox: interactive endpoint explorer with real response format examples

10.3 SDKs and Client Libraries

MCE will publish official SDKs for:

- TypeScript/JavaScript (npm: @mesocrats/mce-sdk) -- planned, to be generated from OpenAPI spec
- Python (pip: mesocrats-mce) -- planned
- Ruby (gem: mesocrats-mce) -- planned

10.4 Sandbox

The sandbox page at developer.mesocrats.org/sandbox displays accurate example responses for all 7 endpoint groups (health, committees, contributors, contributions, disbursements, compliance/limits, reports). It supports GET and POST method selection where applicable and shows request body examples for POST endpoints. The sandbox displays static examples matching the live API response shapes -- it does not make live API calls.

11. Implementation Status

MCE has been implemented in phases. Current status as of March 1, 2026:

Phase	Status	Details
Database schema (7 tables)	LIVE	Deployed in dedicated Supabase project. All RLS enabled.
Compliance engine (validation, limits, aggregates)	LIVE	FEC limit enforcement, aggregate tracking, itemization detection.
Contributor matching (match_key)	LIVE	Auto-generated match_key (last_name + zip). Fuzzy matching planned.
Donation flow (Stripe integration)	LIVE	Custom Stripe on mesocrats.org/donate.
Disbursement tracking	LIVE	Full CRUD with category validation.
Audit log	LIVE	Immutable log on all POST operations.
Public API (14 operations)	LIVE	8 routes, Bearer auth, bcrypt keys, committee-scoped.
Developer portal	LIVE	9+ pages: Products, API Reference, Sandbox, Prompt Library, SDKs, Community, Dashboard.
OpenAPI 3.1.0 specification	LIVE	865 lines. 8 paths, 14 operations, 8 schemas.
API key security (bcrypt)	LIVE	Cost factor 12, prefix lookup, one key = one committee.
E2E testing	DONE	13/13 steps passed. Full flow: health to report creation.
Form 3X generation	PLANNED	Report records created as drafts. .fec generation to follow.
.fec file export	PLANNED	Module designed but not wired to public API.
IRS Form 8872	PLANNED	Data available via contributions endpoint.
Webhook delivery	PLANNED	Events documented in OpenAPI spec.
Rate limiting enforcement	PLANNED	Documented at 100 req/min. Not yet enforced.
TypeScript SDK	PLANNED	Will be generated from OpenAPI spec.

QuickBooks integration	PLANNED	Intuit Developer API.
Fuzzy contributor matching	PLANNED	Levenshtein + email matching.
Internal MCE (/api/mce/*)	LIVE	FEC 3X, IRS 8872 XML, Stripe fee capture, disbursements. ~1,700 lines. Secured by MCE_REPORT_API_KEY. Predates public API.

12. Conclusion

The state of political technology in America is an embarrassment. The tools available to political committees for the most fundamental task -- complying with federal election law -- are outdated, closed, and hostile to the kind of integration and automation that every other industry takes for granted.

The Mesocratic Party was founded on the principle that the middle class deserves better. That principle extends to the infrastructure of democracy itself. If we believe that new voices should be able to enter politics, we must ensure that the technology required to do so is not a barrier.

The Mesocratic Compliance Engine is not just a tool for the Mesocratic National Committee. It is a public good. It is an open platform that any committee, any campaign, any civic organization can use to navigate the complexities of FEC compliance without paying for proprietary software, without hand-entering data into a Windows desktop application from 2018, and without trusting their most sensitive data to closed systems they cannot audit.

As of March 2026, the MCE public API is live. Fourteen endpoints are operational. External developers can create API keys, bind them to committees, and manage the full compliance lifecycle. The developer portal is complete with documentation, examples, and AI-ready prompt templates. This is no longer a vision document -- it is a status report on a working system.

Build it once. Open it to everyone. Set the standard.

That is what the Mesocratic Party does.

This document is published by the Mesocratic National Committee and is available for public download at mesocrats.org.

Paid for by the Mesocratic National Committee. Not authorized by any candidate or candidate's committee.