

THE MESOCRATIC PARTY | TECHNOLOGY WHITE PAPER

developer.mesocrats.org

The Open Platform for American Political Technology

Political parties shouldn't be about competition.

They should be about co-existence.

We didn't build this to go from a duopoly to a triopoly.

We built it to democratize access to the main stage --

so every citizen with a great idea can build a party in the American politiverse.

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

mesocrats.org | developer.mesocrats.org

Table of Contents

Executive Summary

Part I: The Platform

1. The Problem: Why Political Technology Is Gatekept
2. The Vision: An On-Ramp, Not a Walled Garden
3. The Marketplace: developer.mesocrats.org
4. API Product Catalog
5. Developer Portal Architecture (LIVE)
6. The Prompt Engineering Layer (LIVE)
7. Open-Source Commitment and Governance

Part II: Product -- Compliance API (LIVE)

8. Product Overview
9. System Architecture
10. Database Schema (LIVE)
11. Compliance Engine (LIVE)
12. FEC Report Generation
13. IRS Reporting
14. API Specification (LIVE -- 14 Operations)

15. Webhooks (Planned)

Part III: Integrations and Operations

16. Accounting Integration (QuickBooks Online -- Planned)

17. Validation, QA, and the First Filing

18. Implementation Status

Executive Summary

The infrastructure of American democracy is closed. The two major parties have spent decades building proprietary technology platforms -- voter files, compliance tools, fundraising systems, data pipelines -- that give them an insurmountable structural advantage over any new entrant. A citizen who wants to start a political party in America today doesn't just face a battle for ideas. They face a technology gap.

To register a political committee, you need to navigate a maze of FEC forms. To track donations legally, you need compliance software that costs thousands of dollars a year. To file reports, you need either a Windows desktop application from 2018 or a proprietary vendor that locks you into their ecosystem. To access voter data, you need either the Democratic Party's VAN or the Republican Party's Data Trust -- neither of which will sell to you. Every step of forming and operating a political party in America is designed, whether intentionally or not, to favor incumbents and punish newcomers.

The Mesocratic Party was not founded to replace the duopoly with a triopoly. It was founded to build the infrastructure that makes political pluralism possible. developer.mesocrats.org is the embodiment of that mission.

developer.mesocrats.org is an open platform -- a marketplace of API products that remove the friction from building and operating a political party in the American politiverse. It is modeled on the best developer platforms in the private sector (Capital One, Stripe, GitHub) and designed to serve a constituency that has never had one: new political parties, independent campaigns, and civic technology builders.

The first product in the marketplace is the Compliance API -- the Mesocratic Compliance Engine (MCE) -- a comprehensive, open-source platform for FEC compliance, contribution tracking, disbursement management, report generation, and .fec file export. It replaces the need for ISPolitical, NGP VAN, or any proprietary compliance vendor.

UPDATE (March 2026): The Compliance API is live. The public API at developer.mesocrats.org/api/v1/ serves 14 operations across 8 resource paths. The developer portal is complete with authentication, API key management, interactive API Reference, Sandbox, Prompt Library with 6 curated templates, and Products page. External developers can sign up, create API keys, bind them to committees, and manage the full contribution-to-report lifecycle. Multi-tenant

architecture, FEC limit enforcement, aggregate tracking, and immutable audit logging are all operational.

But the most radical feature of the platform is not the APIs themselves -- it is how they are accessed. developer.mesocrats.org is the first political technology platform designed for the AI-native era. Every product page, every workflow, every integration guide includes curated prompt engineering templates -- copy-paste prompts that any citizen can take to an AI assistant like Claude or ChatGPT and produce working, compliant software without writing a single line of code themselves. The platform has two paths: the traditional path (SDKs, API references, code examples) for software engineers, and the prompt path for everyone else. This is what makes developer.mesocrats.org a platform from 2027, not 2017.

Everything on the platform is open-source, documented, and free. Because political parties shouldn't be about competition. They should be about co-existence.

PART I: The Platform

Building the infrastructure for American political pluralism

1. The Problem: Why Political Technology Is Gatekept

Every industry in America has been transformed by open platforms. Developers can build financial products on Stripe and Plaid. They can build communications products on Twilio. They can build e-commerce on Shopify's APIs. They can build AI applications on open-source models. The barrier to entry in virtually every sector of the economy has been demolished by open technology.

Except politics.

1.1 The Compliance Barrier

A political committee that raises or spends more than \$50,000 in any calendar year must file reports electronically with the FEC. The available tools:

- FECFile: The FEC's free software. Windows-only. Desktop application. No API. Manual data entry. Last meaningfully updated in 2018.
- ISPolitical: A proprietary SaaS platform. Basic HTTP authentication. jQuery-era API. Dropped its Stripe integration years ago. Opaque per-committee pricing. No webhooks.
- NGP VAN: The Democratic Party's platform. Enterprise pricing. Not designed for new parties.
- Aristotle: Expensive. Enterprise-focused. Not accessible to a two-person startup committee.

There was no modern, open, API-first compliance platform available to new political committees. Until now.

1.2 The Data Barrier

Voter data is controlled by the two major parties. The Democratic Party's voter file is accessed through VAN/VoteBuilder. The Republican Party's equivalent is Data Trust/i360. Neither sells access to third parties. A new party must build its own voter data operation from scratch, at enormous cost.

1.3 The Formation Barrier

Starting a political party requires navigating a different set of rules in every state: different incorporation requirements, different ballot access petition thresholds, different filing forms, different deadlines. There is no centralized, programmatic resource for party formation requirements.

1.4 The Structural Advantage

The cumulative effect of these barriers is a structural technology advantage for the two major parties. They have had decades and billions of dollars to build the platforms that power their operations. A new party starts with nothing -- and must either build everything from scratch or cobble together expensive, incompatible vendor tools. This is not a level playing field. And it will never be one -- unless someone builds the on-ramp.

2. The Vision: An On-Ramp, Not a Walled Garden

The Mesocratic Party exists to expand the American middle class and to expand the American politiverse. These two missions are inseparable.

developer.mesocrats.org is not a tool for the Mesocratic Party. It is infrastructure for the politiverse. The analogy is deliberate:

- Stripe didn't build payment processing for itself. It built it for everyone.
- GitHub didn't build version control for itself. It built it for every developer.
- developer.mesocrats.org is not building political technology for the Mesocratic Party. It is building it for every party.

The key principles:

- Open by default. Every API, every SDK, every line of source code is open-source under the MIT license.
- Free at the core. The hosted APIs are free for any political committee, campaign, or civic organization.
- Nonpartisan by design. The platform serves any party, any committee, any ideology. The APIs enforce law -- not political alignment.
- Modern by conviction. REST. OpenAPI specs. Webhooks. TypeScript SDKs. Bcrypt-hashed API keys. These are the minimum standard for infrastructure in 2026.
- AI-native by design. Every product includes curated prompt templates that let anyone build with the APIs using AI assistants. This is the first developer platform built for the era where the developer might not be a developer at all.

3. The Marketplace: developer.mesocrats.org

developer.mesocrats.org is a curated marketplace of API products, each designed to remove a specific barrier from the path of building and operating a political organization in America.

3.1 Marketplace Structure

Each product in the marketplace is:

- A standalone API with its own endpoints, authentication, and documentation
- Independently versioned and deployable
- Available as both a hosted service and a self-hostable open-source package
- Documented with interactive API references, guides, and prompt templates
- Backed by a sandbox with example responses for development and testing

3.2 The Portal Experience (LIVE)

A developer visiting developer.mesocrats.org sees:

- Products: A catalog of API products with live/coming-soon status badges and feature lists
- API Reference: Interactive documentation for all 14 live endpoints with request/response schemas and code examples
- Sandbox: Example responses for every endpoint, supporting GET and POST methods with request body examples
- Prompt Library: 6 curated, copy-paste prompt templates for building with AI assistants
- SDKs: TypeScript SDK documentation (generation from OpenAPI spec planned)
- Community: Discussion and support links
- Dashboard: Authenticated area for API key management (create, view prefix, revoke)
- Sign In: GitHub and Google OAuth via Supabase Auth (production mode)

4. API Product Catalog

Product	Description	Status
Compliance API	FEC compliance, contribution tracking, disbursement management, aggregate enforcement, report creation, FEC limits reference, immutable audit log	LIVE (14 endpoints)
Party Formation API	State-by-state guides for incorporating a political party: entity types, filing forms, bylaws templates, EIN registration, FEC Form 1 guidance	PLANNED (2026)

Ballot Access API	State-by-state petition requirements, signature thresholds, filing deadlines, form templates, status tracking across all 50 states	PLANNED (2026)
Election Calendar API	Primary dates, general election dates, filing deadlines, reporting periods, FEC calendar events, state-specific deadlines	PLANNED (2027)
Communications Compliance API	Disclaimer text generator, coordination rule checker, electioneering communication period tracker, communications log	PLANNED (2027)
Candidate Filing API	Federal and state candidate registration requirements: FEC Form 2, state filing forms, eligibility requirements, fee schedules	PLANNED (2027)
Voter Data API	Standardized access to public voter file data across all 50 states, normalized schema, registration status, district assignments	PLANNED (2027)

5. Developer Portal Architecture (LIVE)

The developer portal is integrated into the mesocrats.org Next.js 14 codebase under the (developer) route group, served at developer.mesocrats.org via middleware rewrite.

5.1 Portal Pages (Live)

Page	Path	Description
Landing	/	PartyStack branding, value proposition, product catalog, quick-start CTA
Products	/products	9 live features (green checks), 3 coming-soon (clock icons), 3 future API product cards
API Reference	/api-reference	14 endpoints across 7 categories: System, Committees, Contributors, Contributions, Disbursements, Compliance, Reports
Sandbox	/sandbox	Endpoint selector, GET/POST method toggle,

		request body examples, accurate response previews
Prompt Library	/prompt-library	6 templates: Donation Form, Quarterly Report, Best-Efforts Follow-Up, Treasurer Dashboard, Limit Tracking, Committee Setup
SDKs	/sdks	TypeScript SDK docs. Python/Ruby planned.
Community	/community	Discussion links, contribution guidelines
Sign In	/sign-in	GitHub + Google OAuth via Supabase Auth (production mode)
Dashboard	/dashboard	API key management: create (shown once), view prefix, revoke. Protected route.

5.2 Technology Stack

Component	Technology
Framework	Next.js 14 (App Router), TypeScript
Styling	Tailwind CSS, dark theme (bg-[#0B0F1A])
Fonts	DM Sans (body) + JetBrains Mono (code)
Authentication	Supabase Auth (GitHub + Google OAuth, production mode)
API Key Storage	Main Supabase: developer_api_keys table, bcrypt hashing (cost 12), RLS policies
API Data	PartyStack Supabase (separate project): 7 tables, RLS enabled, multi-tenant
API Spec	OpenAPI 3.1.0 at /openapi.json (865 lines)
Hosting	Vercel (auto-deploy on push to main)
Source	github.com/mesocrats/mesocrats-site (monorepo with main site)

6. The Prompt Engineering Layer (LIVE)

This is the feature that defines developer.mesocrats.org. It is the single most important design decision in the platform.

Traditional developer platforms assume their users are software engineers. They provide API references, SDKs, and code samples -- and they assume the reader can take it from there. This assumption excludes 99% of the people who might benefit from the platform.

developer.mesocrats.org makes a different assumption: the user has a great idea and access to an AI assistant. That's it.

6.1 Two Paths, One Platform

Traditional Path	Prompt Path
For software engineers	For everyone else
OpenAPI 3.1 specification	Curated prompt templates
TypeScript / Python / Ruby SDKs	Copy-paste prompts for Claude, ChatGPT, etc.
Code examples with request/response	Natural language workflow descriptions
Assumes coding proficiency	Assumes the ability to describe what you want

6.2 Live Prompt Templates (6 Templates)

The Prompt Library at developer.mesocrats.org/prompt-library contains 6 curated templates, each referencing real live API endpoints, field names, and response shapes:

#	Template	Difficulty	Key Endpoints Used
1	Build a Donation Form	Starter	POST /api/v1/contributors, POST /api/v1/contributions, GET /api/v1/compliance/limits
2	Generate a Quarterly Report	Intermediate	POST /api/v1/reports, GET /api/v1/contributions, GET /api/v1/disbursements
3	Set Up Best-Efforts Follow-Up	Intermediate	GET /api/v1/contributions, GET /api/v1/contributors, GET /api/v1/compliance/limits
4	Create a Treasurer Dashboard	Advanced	All endpoints: contributions, disbursements, compliance/limits, reports, committees, contributors
5	Track Contribution Limits	Starter	GET /api/v1/compliance/limits, GET /api/v1/contributions, GET /api/v1/contributors
6	Build a Committee Setup Wizard	Starter	POST /api/v1/committees, GET

			/api/v1/committees
--	--	--	--------------------

Each template includes the base URL (<https://developer.mesocrats.org/api/v1>), authorization format (Bearer token), exact request body fields, and real response shapes copied from E2E test output.

6.3 Why This Matters

The prompt engineering layer is not a nice-to-have. It is the core thesis of the platform.

If developer.mesocrats.org only works for software engineers, it reduces the barrier to entry from "you need a proprietary vendor" to "you need a software engineer." That's progress, but it's not transformation.

If developer.mesocrats.org works for anyone who can describe what they need in plain English, it reduces the barrier to entry to zero. That is transformation. That is what democratizing access to the main stage actually looks like.

7. Open-Source Commitment and Governance

Every product on developer.mesocrats.org is released under the MIT license. The source code for the entire platform -- APIs, documentation, portal -- is publicly available on GitHub under the mesocrats organization (github.com/mesocrats/mesocrats-site).

7.1 Contribution Model

- Issues and feature requests via GitHub Issues
- Pull requests reviewed by core maintainers (initially the MNC development team)
- RFC process for significant architectural changes
- Contributor License Agreement (CLA) for external contributors

7.2 Governance

The platform is governed by the Mesocratic National Committee. As the community grows, governance will expand to include an advisory board of contributors, users, and civic technology experts. The commitment: no product on the platform will ever be closed, paywalled at its core, or restricted by political affiliation.

PART II: Product -- Compliance API (LIVE)

The Mesocratic Compliance Engine (MCE)

FEC compliance, campaign finance reporting, and .fec file generation -- as a live, open API

8. Product Overview

The Compliance API is the first product on developer.mesocrats.org. It provides a complete, API-first platform for FEC compliance. As of March 2026, the following capabilities are live:

- Records and validates contributions with real-time FEC limit enforcement
- Tracks contributor aggregates across calendar years with match_key deduplication
- Validates five FEC-required attestations on every individual contribution
- Records and categorizes disbursements with category validation
- Creates draft FEC reports with coverage period and filing deadline
- Returns FEC contribution limits by committee type (2025-2026 cycle)
- Maintains a complete, immutable audit log for every data mutation
- Multi-tenant: each API key scoped to exactly one committee

8.2 What It Replaces

Capability	Legacy Tool	MCE Equivalent
Contribution tracking	ISPolitical, NGP VAN	POST /api/v1/contributions (LIVE)
Aggregate enforcement	Manual / ISPolitical	Real-time, automatic (LIVE)
Limit reference	Manual FEC lookup	GET /api/v1/compliance/limits (LIVE)
Report creation	FECFile (manual)	POST /api/v1/reports (LIVE, drafts)
Disbursement tracking	Spreadsheets	POST /api/v1/disbursements (LIVE)
Donor management	Manual / ISPolitical	POST /api/v1/contributors (LIVE)
Audit trail	None / spreadsheets	Immutable audit_log table (LIVE)
.fec file export	FECFile desktop app	Planned (report drafts live now)
IRS Form 8872	Manual preparation	Planned

9-13: Architecture, Schema, Compliance Engine, Reports, IRS

These sections are fully documented in

MP_The_Mesocratic_Compliance_Engine_v2.docx, the companion white paper. Key updates for v2:

- Architecture: Two Supabase projects (main site + PartyStack API) for multi-tenant data isolation
- Schema: 7 tables live with RLS. Schema defined in src/app/(developer)/api/v1/schema.sql
- Compliance Engine: FEC limit enforcement, aggregate tracking, itemization detection all live
- Reports: Draft creation live. .fec file generation planned.
- IRS: Form 8872 and 1120-POL support planned. Data available via contributions/disbursements endpoints.

14. API Specification (LIVE -- 14 Operations)

14.1 Authentication

MCE uses Bearer token authentication with bcrypt-hashed API keys:

- Developers sign in via GitHub or Google OAuth
- API keys created in dashboard: format mce_live_ + 32 hex characters
- Keys bcrypt-hashed (cost factor 12) on creation, shown once
- Prefix-based lookup + bcrypt.compare on each request
- Each key permanently bound to one committee via POST /api/v1/committees
- Authorization header: Bearer mce_live_<32 hex chars>

14.2 Live Endpoints

Method	Endpoint	Auth	Description
GET	/api/v1/health	None	Status, version, timestamp, DB connectivity
GET	/api/v1/committees	Key (any)	Returns committee bound to API key
POST	/api/v1/committees	Key (unbound)	Creates committee, binds key. Dual-DB write.
GET	/api/v1/contributors	Key (bound)	Paginated, ILIKE search on name/email
POST	/api/v1/contributors	Key (bound)	Create contributor. Auto match_key + full_name.
GET	/api/v1/contributions	Key (bound)	Paginated. Date range, contributor, itemized filters.
POST	/api/v1/contributions	Key (bound)	Record contribution. Limit enforcement + aggregate upsert.
GET	/api/v1/contributions/:id	Key (bound)	Single contribution. 404 if wrong committee.
GET	/api/v1/disbursements	Key (bound)	Paginated. Date range + category filters.
POST	/api/v1/disbursements	Key (bound)	Record disbursement. Category validation.
GET	/api/v1/compliance/limits	Key (bound)	FEC limits by committee type +

			itemization threshold.
GET	/api/v1/reports	Key (bound)	List reports. Status + type filters.
POST	/api/v1/reports	Key (bound)	Create draft FEC report with coverage period.

OpenAPI 3.1.0 specification: 865 lines at developer.mesocrats.org/openapi.json. 8 paths, 14 operations, 8 reusable schemas.

15. Webhooks (Planned)

MCE will dispatch webhooks for contribution.created, disbursement.created, aggregate.threshold_crossed, report.generated, and compliance.deadline_approaching. HMAC-SHA256 signed. Documented in OpenAPI spec but not yet implemented.

PART III: Integrations and Operations

16. Accounting Integration (QuickBooks Online -- Planned)

MCE will integrate with QuickBooks Online via the Intuit Developer API for double-entry bookkeeping. This is a convenience integration -- the compliance engine, report generation, and API all function independently of any accounting platform.

17. Validation, QA, and the First Filing

The Compliance API will be validated through a multi-layer QA process before any filing is submitted. Internal validation (.fec generation + FECFile comparison), external accounting firm review for initial filings, and continuous improvement after each filing cycle.

18. Implementation Status

Component	Status	Notes
Developer portal (9+ pages)	LIVE	Products, API Reference, Sandbox, Prompt Library, SDKs, Community, Sign In, Dashboard
OAuth authentication	LIVE	GitHub + Google via Supabase Auth, production mode
API key management	LIVE	Create, view prefix, revoke. Bcrypt hashing.
Public API (14 operations)	LIVE	8 routes. Bearer auth, committee-scoped, audit

		logged.
Multi-tenant database (7 tables)	LIVE	Separate Supabase project. RLS enabled.
FEC limit enforcement	LIVE	By committee type, 2025-2026 cycle.
Aggregate tracking + itemization	LIVE	Upsert on every contribution. \$200 threshold.
Contributor matching (match_key)	LIVE	Auto-generated. Fuzzy matching planned.
OpenAPI specification	LIVE	865 lines. 8 paths, 14 operations, 8 schemas.
Prompt Library (6 templates)	LIVE	Real endpoints, real field names, real response shapes.
E2E testing	DONE	13/13 steps passed.
.fec file export	PLANNED	Report drafts live. File generation to follow.
Webhook delivery	PLANNED	Documented in OpenAPI spec.
Rate limiting enforcement	PLANNED	Documented at 100 req/min.
TypeScript SDK	PLANNED	Generate from OpenAPI spec.
QuickBooks integration	PLANNED	Intuit Developer API.
Party Formation API	PLANNED	2026
Ballot Access API	PLANNED	2026
Election Calendar API	PLANNED	2027

Conclusion

The infrastructure of American democracy should not be proprietary. The tools required to participate in self-governance -- to form a party, to accept a donation, to file a report, to get on a ballot -- should not be locked behind enterprise pricing, Windows desktop applications, and jQuery-era APIs.

The Mesocratic Party was founded on a simple premise: the American middle class is larger, more capable, and more diverse than two political parties can represent. But that premise is meaningless if the barrier to entry for new parties is a technology gap that only decades of institutional funding can close.

developer.mesocrats.org is our answer. It is not a competitive advantage. It is shared infrastructure. It is the on-ramp that every citizen with a vision for this country deserves -- and that no one has built until now.

As of March 2026, the first product is live. The Compliance API serves 14 operations. The developer portal is complete. The Prompt Library lets anyone -- not just engineers -- build FEC-compliant political technology by pasting a prompt into an AI assistant. The on-ramp is open.

We are not building from a duopoly to a triopoly. We are building the infrastructure for co-existence. Because that is what democracy actually looks like when the technology gets out of the way.

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

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.

© 2026 Mesocratic National Committee. Source code released under MIT License.