Project: Configurable Marketing Automation Platform (Event Campaign MVP)  
 Goal: Build, edit, and launch the end‑to‑end campaign shown in *Event Campaign Flow – ADTV* (page 1) using a flexible workflow engine, visual editor, and integrations for Email, SMS (Twilio), Voicemail Drop (ElevenLabs + drop provider), Calendly, internal e‑sign, and Podio CRM. The diagram enumerates the exact steps: Campaign 1 (Email/SMS/VM1 → Analytics → Check Data → Pos/Question/Neg/No‑Response), Campaign 2 (+3 days, Email/SMS/VM2), RSVP paths (Virtual Calendly / In‑Person BDR), Event‑Day comms (8am SMS, Big Email, +1hr SMS), Post‑event branches (No Shows, Cancellations, Selects #1/#2/#3, Attended/No Response), and e‑sign → Podio.

Event Campaign Flow - ADTV

0) Deliverables (what “done” means)

1. Monorepo with backend API, workflow engine workers, and React app (visual editor + dashboards).
2. Workflow JSON DSL & engine that executes the campaign, supports add/edit/delete nodes, new campaigns, and versioning.
3. Integrations: Email provider (provider‑agnostic adapter), Twilio SMS, ElevenLabs TTS + voicemail drop provider, Calendly webhooks, Internal e‑sign, Podio upsert.
4. Dashboards: campaign funnel, node performance, RSVP/attendance, e‑sign conversion, compliance/health.
5. Seeded campaign that mirrors the diagram’s nodes & timing (two pre‑event campaigns, nurture, RSVP, event‑day comms, post‑event branches, e‑sign).  
    Event Campaign Flow - ADTV
6. Launch‑ready: environment configs, CI checks, test suite, and a production rollout checklist.

1) Tech stack (opinionated for this build)

* Backend: Node.js + NestJS (typed modules, DI), PostgreSQL (Prisma ORM), Redis (BullMQ queues & timers), OpenAPI 3.1.
* Frontend: Next.js (App Router), React Flow for the diagram editor, Zustand for local state.
* Shared: TypeScript workspace with @types/shared for domain types.
* Storage: Render Storage (voicemail audio assets, exports).
* Observability: pino (structured logs), Prom‑style metrics endpoints.
* Auth & tenancy: Email/password or SSO (MVP: workspace-scoped JWT), RBAC (Owner/Admin/Marketer/Analyst/Support).

Temporal is a great future option; for MVP we use BullMQ timers and idempotent jobs to stay lightweight.

2) Monorepo layout

/apps

/api # NestJS HTTP + webhooks

/worker # BullMQ consumers, workflow engine

/web # Next.js UI (editor, dashboards)

/docs # OpenAPI, ADRs, runbooks

/packages

/shared # shared types, json schemas

/engine # pure workflow interpreter + validators

/adapters # email, twilio, elevenlabs, voicemail, calendly, esign, podio

/templating # safe templating & link tracking

/prisma

schema.prisma

/scripts

seed-campaign.ts

migrate-and-seed.sh

NPM scripts (root):

* pnpm dev:all (concurrently run api, worker, web)
* pnpm db:migrate, pnpm db:seed
* pnpm test, pnpm lint, pnpm typecheck

3) Data model (Prisma sketch)

model Workspace { id String @id @default(uuid()) name String createdAt DateTime @default(now()) users User[] }

model User {

id String @id @default(uuid())

email String @unique

role Role

workspaceId String

workspace Workspace @relation(fields: [workspaceId], references: [id])

createdAt DateTime @default(now())

}

enum Role { owner admin marketer analyst support }

model Contact {

id String @id @default(uuid())

workspaceId String

email String?

phoneE164 String?

firstName String?

lastName String?

timezone String?

locale String?

attributes Json @default("{}")

createdAt DateTime @default(now())

@@index([workspaceId])

}

model Consent {

id String @id @default(uuid())

workspaceId String

contactId String

channel Channel

purpose String

status ConsentStatus

source String?

metadata Json?

createdAt DateTime @default(now())

}

enum Channel { email sms voicemail all }

enum ConsentStatus { granted denied revoked }

model Campaign {

id String @id @default(uuid())

workspaceId String

name String

status CampaignStatus

version Int @default(1)

createdBy String

createdAt DateTime @default(now())

graphs CampaignGraph[]

}

enum CampaignStatus { draft published archived }

model CampaignGraph {

id String @id @default(uuid())

campaignId String

schemaVersion Int @default(1)

nodes Json // NodeBase[]

edges Json // Edge[]

startRules Json

settings Json

createdAt DateTime @default(now())

}

model Journey {

id String @id @default(uuid())

campaignId String

contactId String

state String

startedAt DateTime?

endedAt DateTime?

steps JourneyStep[]

}

model JourneyStep {

id String @id @default(uuid())

journeyId String

nodeId String

attempt Int @default(1)

status StepStatus

result Json?

startedAt DateTime?

endedAt DateTime?

idemKey String @unique

}

enum StepStatus { scheduled running succeeded failed skipped waiting }

model Event {

id BigInt @id @default(autoincrement())

workspaceId String

contactId String?

campaignId String?

type String

properties Json

occurredAt DateTime

@@index([workspaceId, occurredAt])

}

model Connection {

id String @id @default(uuid())

workspaceId String

kind String // email\_provider, twilio, elevenlabs, voicemail\_provider, calendly, podio

config Json // secrets in KMS

createdAt DateTime @default(now())

}

4) Workflow engine (package: @engine)

Node types: start | decision | ab\_split | wait | email\_send | sms\_send | voicemail\_drop | esign | web\_request | task | webhook\_wait | goal | exit | stage  
 Semantics: deterministic per‑Journey state machine; timers via BullMQ; idempotency key (journeyId,nodeId,attempt); retries with exponential backoff; per‑channel rate limits; quiet hours (contact TZ) with override for Event‑Day 8am SMS.

Execution loop:

1. Dequeue next step → run *entry guard* (json‑logic).
2. If send/action: check consent → dispatch via adapter → emit events.
3. If wait/webhook\_wait: schedule timer or subscribe to event.
4. On completion/failure/timeout: choose matching edge and enqueue next node.
5. Persist step result and metrics; continue until goal/exit.

5) Providers (package: @adapters)

* EmailAdapter (provider‑agnostic): sendEmail(), webhooks: delivered, bounce, complaint, open, click.
* TwilioSmsAdapter: sendSms(), delivery callbacks, inbound handlers with keyword router (STOP/START/HELP + custom “QUESTION”).
* ElevenLabsTtsAdapter: synthesize(text, voiceId) -> s3://... (cache by hash).
* VoicemailDropAdapter: createDrop(audioUrl, to), status callbacks (queued/delivered/failed).
* CalendlyAdapter: subscribe to invitee.created/invitee.canceled, map to events and hydrate contact fields.
* EsignAdapter (internal): create signer session, statuses (sent/viewed/signed/declined/expired), reminder schedule.
* PodioAdapter: upsert record on signed with mapped contact/event fields.

6) Visual editor (app: /apps/web)

* Canvas (React Flow): palette for node types above.
* Node inspector: per‑type schema forms (template pickers, wait durations, retry policy, fallback edges).
* Edges: draw & order by priority; conditions via JSON‑logic builder.
* Validation: dangling nodes, unreachable exits, missing templates/connections, consent checks, quiet‑hour violations.
* Versioning: Draft → Publish (immutable snapshot).
* Journey inspector: timeline of touches/events, current node & pending timers, consent snapshot.
* Dashboards: see §10.

7) API surface (apps/api)

* POST /contacts (bulk upsert), GET /contacts?...
* POST /campaigns → POST /campaigns/{id}/publish → GET /campaigns/{id}
* POST /campaigns/{id}/enroll (manual enrollment)
* POST /events/webhooks/:provider (email|twilio|voicemail|calendly|esign)
* GET /journeys?contact\_id=&campaign\_id= → GET /journeys/{id} → POST /journeys/{id}/cancel
* GET /metrics/\* (pre‑aggregated time series)
* All mutating endpoints accept Idempotency-Key.

8) Seed the exact campaign (diagram‑aligned)

The diagram shows Event Master List → Campaign 1 (Email 1, SMS 1 +10m, ElevenLabs VOICEMAIL DROP 1 +1 day) → Pull Analytics → Check Data → Pos/Question/Neg/No Response branches; Campaign 2 after +3 days with Email 2, SMS 2 +10m, VOICEMAIL DROP 2 +1 day; RSVP via Virtual Calendly or In‑Person BDR; Event Day triggers (8am Text, Big Email, +1hr Biggie‑Small SMS); Post‑event outcomes for no shows, cancellations, selects #1/#2/#3, attended/no response; e‑sign → Signs/No Signature (+1d follow‑up) → Podio Entry Created. Seed reflects that logic.

Event Campaign Flow - ADTV

Action: create /scripts/seed-campaign.ts that writes this JSON into CampaignGraph.  
 Notes on revisions:

* Added an explicit Event Type decision after RSVP Email to route to Virtual (Calendly) vs In‑Person (BDR) since both appear as options in the diagram; this is an implementation convenience for mutually exclusive paths.  
   Event Campaign Flow - ADTV
* Added Nurture (Sign Up Date Based) as a scheduled wait that can emit the RSVP Email on sign‑up date.  
   Event Campaign Flow - ADTV
* Added Virtual Event BDR TRIGGERS node to represent day‑of automations for virtual events.  
   Event Campaign Flow - ADTV

{

"schema\_version": 1,

"nodes": [

{"id":"N00","type":"start","name":"Event Master List"},

{"id":"N10","type":"stage","name":"Campaign 1"},

{"id":"N11","type":"email\_send","name":"Email 1","config":{"template\_id":"tmpl\_email\_1"}},

{"id":"N12","type":"sms\_send","name":"SMS 1","config":{"template\_id":"tmpl\_sms\_1"}},

{"id":"N13","type":"voicemail\_drop","name":"ElevenLabs VOICEMAIL DROP 1","config":{"tts":{"provider":"elevenlabs","voice\_id":"v\_default","script\_template\_id":"tmpl\_vm\_1"},"drop":{"provider":"vm\_provider"}}},

{"id":"N14","type":"web\_request","name":"Pull Analytics","config":{"endpoint":"internal://analytics/pull","method":"POST"}},

{"id":"N15","type":"decision","name":"Check Data","config":{"rules":[

{"label":"Pos Response","expr":{"==":[{"var":"contact.rsvp\_intent"},"positive"]},"to":"N20"},

{"label":"Question Response","expr":{"var":"contact.question\_flag"},"to":"N52"},

{"label":"Neg Response","expr":{"==":[{"var":"contact.rsvp\_intent"},"negative"]},"to":"N59"},

{"label":"No Response","expr":{"==":[{"var":"contact.rsvp\_intent"},null]},"to":"N30"}

]}},

{"id":"N20","type":"email\_send","name":"RSVP Email","config":{"template\_id":"tmpl\_rsvp"}},

{"id":"N21","type":"decision","name":"Event Type Decision","config":{"rules":[

{"label":"Virtual","expr":{"==":[{"var":"contact.event\_type"},"virtual"]},"to":"N50"},

{"label":"In Person","expr":{"==":[{"var":"contact.event\_type"},"in\_person"]},"to":"N51"}

]}},

{"id":"N30","type":"stage","name":"Campaign 2 (+3 days)"},

{"id":"N31","type":"email\_send","name":"Email 2","config":{"template\_id":"tmpl\_email\_2"}},

{"id":"N32","type":"sms\_send","name":"SMS 2","config":{"template\_id":"tmpl\_sms\_2"}},

{"id":"N33","type":"voicemail\_drop","name":"ElevenLabs VOICEMAIL DROP 2","config":{"tts":{"provider":"elevenlabs","voice\_id":"v\_default","script\_template\_id":"tmpl\_vm\_2"},"drop":{"provider":"vm\_provider"}}},

{"id":"N40","type":"wait","name":"Nurture (Sign Up Date Based)","config":{"until\_date\_field":"contact.signup\_date","offset":"PT0S"}},

{"id":"N50","type":"web\_request","name":"Virtual Event – Calendly Link","config":{"endpoint":"calendly://send\_link"}},

{"id":"N53","type":"webhook\_wait","name":"Confirmation via Calendly","config":{"provider":"calendly","event":"invitee.created","timeout":"P3D","on\_timeout":"N52"}},

{"id":"N51","type":"task","name":"In Person – Email Coordinated/BDR","config":{"assignee\_role":"BDR"}},

{"id":"N52","type":"task","name":"BDR Responds","config":{"assignee\_role":"BDR"}},

{"id":"N54","type":"task","name":"BDR Confirmation","config":{"assignee\_role":"BDR","mark":"confirmed"}},

{"id":"N60","type":"stage","name":"Event Day"},

{"id":"N61","type":"web\_request","name":"Virtual Event BDR TRIGGERS","config":{"endpoint":"internal://bdr/virtual-triggers"}},

{"id":"N63","type":"sms\_send","name":"Text Confirmation 8am event day","config":{"template\_id":"tmpl\_sms\_8am"}},

{"id":"N62","type":"email\_send","name":"Big Email During Event","config":{"template\_id":"tmpl\_email\_live"}},

{"id":"N64","type":"sms\_send","name":"Biggie-Small SMS (+1hr)","config":{"template\_id":"tmpl\_sms\_biggie"}},

{"id":"N70","type":"decision","name":"Post-Event Outcomes","config":{"rules":[

{"label":"no shows","expr":{"==":[{"var":"contact.attendance\_status"},"no\_show"]},"to":"N71"},

{"label":"Cancellations","expr":{"==":[{"var":"contact.attendance\_status"},"canceled"]},"to":"N73"},

{"label":"Attended / No Response","expr":{"==":[{"var":"contact.attendance\_status"},"attended"]},"to":"N80"}

]}},

{"id":"N71","type":"email\_send","name":"No Shows Email","config":{"template\_id":"tmpl\_email\_noshow"}},

{"id":"N73","type":"email\_send","name":"Cancellations Email","config":{"template\_id":"tmpl\_email\_cancel"}},

{"id":"N74","type":"decision","name":"Selects #1/#2/#3","config":{"rules":[

{"label":"Selects #1","expr":{"==":[{"var":"contact.selection"},"1"]},"to":"N77"},

{"label":"Selects #2","expr":{"==":[{"var":"contact.selection"},"2"]},"to":"N77"},

{"label":"Selects #3","expr":{"==":[{"var":"contact.selection"},"3"]},"to":"N78"}

]}},

{"id":"N77","type":"email\_send","name":"Click #2 – Questions/Calendly","config":{"template\_id":"tmpl\_email\_questions"}},

{"id":"N78","type":"email\_send","name":"Final Email","config":{"template\_id":"tmpl\_email\_final"}},

{"id":"N79","type":"decision","name":"Question Response","config":{"rules":[

{"label":"Has Questions","expr":{"var":"contact.question\_flag"},"to":"N81"},

{"label":"No Questions","expr":{"!":{"var":"contact.question\_flag"}},"to":"N82"}

]}},

{"id":"N80","type":"task","name":"Attended / No Response → BDR","config":{"assignee\_role":"BDR"}},

{"id":"N81","type":"task","name":"BDR Outreach","config":{"assignee\_role":"BDR"}},

{"id":"N82","type":"esign","name":"eSign Agreement","config":{"document\_template\_id":"doc\_agreement\_v1","reminders":{"enabled":true,"interval":"P1D","max":1}}},

{"id":"N83","type":"goal","name":"Signs"},

{"id":"N84","type":"web\_request","name":"Podio Entry Created","config":{"endpoint":"podio://upsert"}},

{"id":"N85","type":"decision","name":"No Signature?","config":{"rules":[

{"label":"No Signature","expr":{"==":[{"var":"contact.esign\_status"},"sent"]},"to":"N86"}

]}},

{"id":"N86","type":"wait","name":"eSign Follow Up (+1d)","config":{"duration":"P1D"}},

{"id":"N59","type":"exit","name":"Stop"},

{"id":"N89","type":"exit","name":"Stop"}

],

"edges": [

{"from":"N00","to":"N10"},

{"from":"N10","to":"N11"},

{"from":"N11","to":"N12","condition":{"after":"PT10M"}},

{"from":"N12","to":"N13","condition":{"after":"P1D"}},

{"from":"N13","to":"N14"},

{"from":"N14","to":"N15"},

{"from":"N15","to":"N20","condition":{"label":"Pos Response"}},

{"from":"N15","to":"N52","condition":{"label":"Question Response"}},

{"from":"N15","to":"N59","condition":{"label":"Neg Response"}},

{"from":"N15","to":"N30","condition":{"label":"No Response"}},

{"from":"N30","to":"N31","condition":{"after":"P3D"}},

{"from":"N31","to":"N32","condition":{"after":"PT10M"}},

{"from":"N32","to":"N33","condition":{"after":"P1D"}},

{"from":"N40","to":"N20"},

{"from":"N20","to":"N21"},

{"from":"N21","to":"N50","condition":{"label":"Virtual"}},

{"from":"N21","to":"N51","condition":{"label":"In Person"}},

{"from":"N50","to":"N53"},

{"from":"N53","to":"N54","condition":{"on":"success"}},

{"from":"N51","to":"N54"},

{"from":"N54","to":"N60"},

{"from":"N60","to":"N61"},

{"from":"N60","to":"N63","condition":{"at\_local":"08:00"}},

{"from":"N60","to":"N62"},

{"from":"N62","to":"N64","condition":{"after":"PT1H"}},

{"from":"N60","to":"N70","condition":{"after":"PT4H"}},

{"from":"N70","to":"N71","condition":{"label":"no shows"}},

{"from":"N70","to":"N73","condition":{"label":"Cancellations"}},

{"from":"N70","to":"N80","condition":{"label":"Attended / No Response"}},

{"from":"N71","to":"N82"},

{"from":"N73","to":"N82"},

{"from":"N80","to":"N81"},

{"from":"N82","to":"N83","condition":{"on":"signed"}},

{"from":"N82","to":"N85","condition":{"on":"sent"}},

{"from":"N85","to":"N86"},

{"from":"N86","to":"N82"},

{"from":"N83","to":"N84"},

{"from":"N83","to":"N89"}

],

"start\_rules":{

"enroll":[{"type":"segment","segment\_id":"event\_master\_list"}],

"reentry\_policy":"single\_entry",

"quiet\_hours":{"start":"21:00","end":"08:00","contact\_timezone":true},

"caps":{"email\_per\_day":2,"sms\_per\_day":1,"voicemail\_per\_day":1}

}

}

9) Templates & variables

* Email templates: tmpl\_email\_1, tmpl\_email\_2, tmpl\_rsvp, tmpl\_email\_live, tmpl\_email\_noshow, tmpl\_email\_cancel, tmpl\_email\_questions, tmpl\_email\_final.
* SMS templates: tmpl\_sms\_1, tmpl\_sms\_2, tmpl\_sms\_8am, tmpl\_sms\_biggie.
* Voicemail scripts: tmpl\_vm\_1, tmpl\_vm\_2.
* e‑sign: doc\_agreement\_v1.
* Variables: {{contact.first\_name}}, {{contact.event\_date}} (localized), UTM builder, conditionals (e.g., show/hide meeting link).
* Link tracking with signed redirects, UTM injection (source=automation, campaign={{campaign.name}}).

10) Dashboards (apps/web)

* Campaign Overview: enrolled → messaged (per channel) → responded (pos/question/neg) → RSVP confirmed → attended → e‑sign sent → signed → Podio created.
* Node Performance: send/deliver/fail; open/click/reply; VM delivered; e‑sign view/sign; path ratios at decision nodes.
* RSVP & Attendance: Calendly conversion, confirmation lag, no‑show/cancel rates.
* Event‑day Live: counts by status; SMS/Email throughput; inbound questions.
* Compliance/Health: STOP rate, hard bounces, drop failures, retry queues.

11) Compliance & safety

* Consent checks before every send; STOP/UNSUB keywords update consent & halt SMS branch.
* Quiet hours (21:00–08:00 contact TZ) except explicit Event‑Day 8am SMS.
* A2P/10DLC alignment for Twilio; store consent provenance.
* PII protection: encrypt sensitive fields, mask in logs.
* Audit logs on graph edits, template changes, consent updates.

12) Implementation steps (do these in Cursor)

12.1 Bootstrap & infra

1. Init PNPM workspace; add apps/packages folders.
2. Add NestJS API (/apps/api), Next.js web (/apps/web), worker (/apps/worker).
3. Configure Prisma → PostgreSQL; Redis in docker‑compose; S3 bucket (MinIO local OK).
4. Add BullMQ; set up queues: workflow:execute, workflow:timer, webhooks:\*.

12.2 Shared types & schemas

1. In @shared, export NodeType, NodeBase, Edge, CampaignGraph, event types.
2. Add Zod/JSON Schema for node configs; validator in @engine.

12.3 Engine & workers

1. Implement Engine.runStep(journey, node) with idempotency & retries.
2. Implement wait, webhook\_wait, and timer scheduling; persist timers in Redis keyed by journey.
3. Implement consent guard, rate limits, quiet hours deferral.
4. Emit Event rows for all provider callbacks; wire webhooks to wake waiting nodes.

12.4 Adapters

* Stub each adapter with sandbox/test creds; add signature verification for inbound webhooks.
* Store ElevenLabs audio in S3; reuse by hash (script + voiceId).
* Podio upsert maps: contact info, event metadata, esign doc id.

12.5 API & webhooks

* Routes from §7; expose OpenAPI docs; include Idempotency-Key.
* Webhook endpoints: /events/webhooks/email, /events/webhooks/twilio, /events/webhooks/voicemail, /events/webhooks/calendly, /events/webhooks/esign.

12.6 Visual editor

* React Flow canvas; node palette; JSON preview; validation panel.
* Versioning workflow (draft → publish).
* Journey inspector (search by contact).

12.7 Seed & simulate

* Implement /scripts/seed-campaign.ts to insert the JSON above as v1.
* Add a simulate endpoint to run a contact through the graph and produce an annotated timeline.

12.8 Dashboards

* Materialized views or nightly jobs for aggregates; charts in web.
* Live ops page for event day (polling/streaming metrics).

13) Testing & acceptance

Unit (Jest): node executors; consent guard; timers; idempotency; adapters (mocked).  
 Integration: webhook signature & de‑dupe; RSVP/Calendly round‑trip; e‑sign reminder path.  
 E2E (engine simulation):

* Happy path: N00 → N10 → … → N20 → N21(Virtual) → N50 → N53 → N54 → N60 → N63/N62/N64 → N70(attended) → N80 → N81 → N82 → N83 → N84 → N89.  
   Event Campaign Flow - ADTV
* No response to Campaign 1, then Campaign 2: N00 → N10 → … → N15(No Response) → N30 → N31 → N32 → N33 → RSVP path.  
   Event Campaign Flow - ADTV
* Negative response: N15(Neg Response) → Stop (N59).  
   Event Campaign Flow - ADTV
* No‑show: N70(no shows) → N71 → N82 → N83 → N84.  
   Event Campaign Flow - ADTV
* No signature: N82(sent) → N85 → N86(+1d) → N82 retry → either N83 or Stop (N89).  
   Event Campaign Flow - ADTV

Acceptance criteria (per node type):

* email\_send: idempotent; bounce/complaint suppression; tracked opens/clicks.
* sms\_send: STOP/HELP/START compliance; inbound “question” flags contact.
* voicemail\_drop: audio generated once; status tracked; retry w/backoff.
* webhook\_wait: resumes only on matching signed webhook or timeout.
* wait: accurate to ±1s.
* decision: deterministic evaluation (json‑logic).
* esign: signed → goal; sent but unsigned → reminder at +1d.

14) Security, ops, and SRE

* Secrets: .env for dev; KMS or Vault for prod.
* Logs: pino JSON; request IDs; redaction of PII.
* Metrics: queue depth, throughput, error rates, webhook latency, provider failures.
* Runbooks: provider outage, dead‑letter requeue, consent incident handling.
* Backups: nightly Postgres snapshots; S3 lifecycle for assets.

15) Launch plan (prod)

1. Create production Postgres, Redis, S3; wire secrets.
2. Configure Twilio A2P/10DLC, sending pool, webhooks; verify STOP handling.
3. Set ElevenLabs key; voice selection; generate sample audio.
4. Configure voicemail provider and callbacks.
5. Set Calendly webhook subscription; verify invitee.created/canceled.
6. Configure e‑sign document template; branding; test sign/decline/expire.
7. Configure Podio API credentials; test upsert.
8. Import contacts into Event Master List; publish v1; enroll a test cohort.
9. Dry‑run simulation; then enable real sends with capped throughput.
10. Event‑day ops: open live dashboard; verify 8am SMS and in‑event email/SMS sequencing.  
     Event Campaign Flow - ADTV

16) Developer ergonomics

* OpenAPI‑first: generate API clients for web/worker.
* Feature flags: quiet‑hour overrides, provider failover.
* CLI: pnpm simulate --contact <id> --campaign <id> to print a timeline.
* Docs: docs/adr/ for key architecture decisions.

17) Field mappings (Podio / contact attributes)

* rsvp\_status, event\_date, event\_type, meeting\_link, meeting\_time, attendance\_status, selection, question\_flag, esign\_status, podio\_record\_id.
* Events to log: email\_\* (sent/open/click/reply), sms\_\* (sent/delivered/reply/stop), voicemail\_*, rsvp\_*, calendly\_*, attendance\_marked, selection\_made, esign\_*, podio\_upserted.  
   These directly support the post‑event splits (no shows / cancellations / selects #1/#2/#3 / attended).  
   Event Campaign Flow - ADTV

18) What to double‑check against the diagram (content, not structure)

* Exact copy of “Click #2 – Send ‘What are your questions’ / Calendly link” (use the node key tmpl\_email\_questions now; adjust later if the copy changes).  
   Event Campaign Flow - ADTV
* The semantics of Selects #1/#2/#3 (we treat as a contact field selection driving the follow‑up).  
   Event Campaign Flow - ADTV

The structural flow, timings (e.g., SMS +10m, voicemail +1 day, Campaign 2 +3 days, 8am event‑day SMS, Biggie‑Small +1hr), and endpoints (Calendly, BDR, e‑sign, Podio) are implemented as shown on page 1.

Event Campaign Flow - ADTV

That’s the complete build/edit/launch instruction set.

If you paste this as INSTRUCTIONS.md in Cursor, you can scaffold the repo and start implementing module by module without guesswork. The seed JSON, node types, adapters, and dashboards all align to the Event Campaign Flow – ADTV diagram so your MVP produces the exact behavior you designed while remaining fully configurable for future campaigns.

1) Node catalog (from page 1)

IDs are assigned for clarity. “Δt” means delay relative to the node that starts the mini‑sequence (e.g., Campaign 1 or Event Day). Colors reflect the diagram’s visual cues (Purple = action step, Green = decision/outcome, Orange = BDR/manual, Red = stop, Brown = stage marker).

Event Campaign Flow - ADTV

| **ID** | **Label (exact/near‑exact)** | **Type** | **Channel/Function** | **Δt / Timing** |
| --- | --- | --- | --- | --- |
| N00 | **Event Master List** | Start/Segment | Enrollment source | — |
| N10 | **Campaign 1** | Stage | Container | — |
| N11 | **Email 1** | Send | Email | +0m |
| N12 | **SMS 1** | Send | Twilio SMS | +10m |
| N13 | **ElevenLabs VOICEMAIL DROP 1** | Send (2‑step) | ElevenLabs TTS → Voicemail drop provider | +1 day |
| N14 | **Pull Analytics** | Data task | Aggregate metrics/events | after N13 |
| N15 | **Check Data** | Decision | Branching over responses | after N14 |
| N16 | **Pos Response** | Outcome | Decision result | — |
| N17 | **Question Response** | Outcome | Decision result | — |
| N18 | **Neg Response** | Outcome | Decision result | — |
| N19 | **No Response** | Outcome | Decision result | — |
| N20 | **RSVP Email** | Send | Email (invite/RSVP) | after branching |
| N30 | **Campaign 2 (+3 days)** | Stage | Container | +3 days from Campaign 1 |
| N31 | **Email 2** | Send | Email | +0m |
| N32 | **SMS 2** | Send | Twilio SMS | +10m |
| N33 | **ElevenLabs VOICEMAIL DROP 2** | Send | ElevenLabs → Voicemail drop provider | +1 day |
| N40 | **Nurture (Sign Up Date Based)** | Journey | Date‑based nurture | schedule rule |
| N50 | **Virtual Event – Calendly Link** | Send/CTA | Email/SMS CTA to Calendly | pre‑event |
| N51 | **In Person – Email Coordinated/BDR** | Process | Coordinated via Email/BDR | pre‑event |
| N52 | **BDR Responds** | Manual | BDR task | after N50/N51 |
| N53 | **Confirmation via Calendly** | Webhook/Status | Calendly confirmation | after N50 |
| N54 | **BDR Confirmation** | Manual | BDR marks confirmed | after N52 |
| N59 | **Stop** | Exit | End branch | — |
| N60 | **Event Day** | Stage (Brown) | Day‑of marker | event date |
| N61 | **Virtual Event BDR TRIGGERS** | Trigger set | BDR automations | day‑of |
| N62 | **Big Email During Event** | Send | Email broadcast | during event |
| N63 | **Text Confirmation 8am event day** | Send | SMS | 8:00 (contact TZ) |
| N64 | **Biggie‑Small SMS** | Send | SMS | +1 hr (from N62 or event start) |
| N70 | **no shows** | Outcome | Decision result | post‑event |
| N71 | **No Shows Email** | Send | Email | after N70 |
| N72 | **Cancellations** | Outcome | Decision result | post‑event |
| N73 | **Cancellations Email** | Send | Email | after N72 |
| N74 | **Selects #1** | Outcome | Choice result | post‑event |
| N75 | **Selects #2** | Outcome | Choice result | post‑event |
| N76 | **Selects #3** | Outcome | Choice result | post‑event |
| N77 | **Click #2 – “What are your questions” / Calendly link** | Send/CTA | Email/SMS follow‑up with Calendly | post‑event |
| N78 | **Final Email** | Send | Email | after selections |
| N79 | **Question Response** | Outcome | Decision result | post‑event |
| N80 | **Attended / No Response** | Outcome | Decision result | post‑event |
| N81 | **BDR** | Manual | BDR outreach | after N80 or questions |
| N82 | **eSign Agreement** | Send/Task | Internal e‑sign | post‑event |
| N83 | **Signs** | Outcome | E‑sign status | after N82 |
| N84 | **Podio Entry Created** | Integration | Create/Update Podio record | after N83 |
| N85 | **No Signature** | Outcome | E‑sign status | after N82 |
| N86 | **eSign Follow Up** | Send/Task | Reminder | +1 day after N85 |
| N89 | **Stop** | Exit | End branch | — |

All of the above appear on **page 1** as labeled or strongly implied by adjacency and arrows.

Event Campaign Flow - ADTV

2) Edge map (who flows into what)

This condenses the arrows into ordered logic; conditions mirror the green outcome boxes.

Event Campaign Flow - ADTV

**Enrollment & Campaign 1**

1. N00 → N10 (enroll contacts from **Event Master List**).
2. N10 → N11 → N12 (Δ+10m) → N13 (Δ+1d).
3. N13 → N14 → N15 (analytics then decision).
4. N15:  
   * **Pos Response (N16)** → N20 (send **RSVP Email**).
   * **Question Response (N17)** → N52 (BDR Responds) → N54 (BDR Confirmation) or N53 (Calendly Confirmation).
   * **Neg Response (N18)** → N59 (**Stop**).
   * **No Response (N19)** → proceed to **Campaign 2**.

**Campaign 2 (+3 days)** 5. N10 → **after +3 days** → N30 → N31 → N32 (Δ+10m) → N33 (Δ+1d).  
 6. (Implied) Post‑N33 run analytics/decision similarly to N14/N15; unresolved contacts move to RSVP/Scheduling or Nurture.

**RSVP/Scheduling (pre‑event)** 7. RSVP path can reach:

* N50 (**Virtual Event – Calendly Link**) → N53 (**Confirmation via Calendly**), → N54 (**BDR Confirmation**) as needed.
* N51 (**In Person – Email Coordinated/BDR**) → N52 (**BDR Responds**) → N54 (**BDR Confirmation**) → N59 (**Stop**) when confirmed.  
   Event Campaign Flow - ADTV

**Event‑day comms** 8. N60 (**Event Day**) activates N61 (**Virtual Event BDR TRIGGERS**), N62 (**Big Email During Event**), N63 (**Text Confirmation 8am**), and N64 (**Biggie‑Small SMS +1hr**).

Event Campaign Flow - ADTV

**Post‑event branching** 9. Outcomes (green boxes): **no shows (N70)** → **No Shows Email (N71)**; **Cancellations (N72)** → **Cancellations Email (N73)**; **Selects #1/#2/#3 (N74–N76)** drive tailored follow‑ups including **Final Email (N78)** and **Click #2 – Questions + Calendly (N77)**; **Attended / No Response (N80)** → **BDR (N81)**. **Question Response (N79)** may also route to **BDR (N81)**.

Event Campaign Flow - ADTV

**Agreement & CRM** 10. **eSign Agreement (N82)** → **Signs (N83)** → **Podio Entry Created (N84)**; if **No Signature (N85)** → **eSign Follow Up +1d (N86)** then exit (**Stop**, N89) or loop once (policy).

Event Campaign Flow - ADTV

3) Start rules, re‑entry, and frequency caps (proposed)

* **Start/Enrollment:** Contacts in **Event Master List** (N00). Also allow webhook enrollment from RSVP forms or uploads.  
   Event Campaign Flow - ADTV
* **Re‑entry:** Single active journey per contact per campaign version; allow re‑entry after exit if event date changes (cool‑down 14 days).
* **Quiet hours:** 8pm–8am contact local time (except **Event Day 8am confirmation** which is explicitly allowed).
* **Frequency caps:** Max 2 emails/day, 1 SMS/day, 1 voicemail/day/contact.

4) Channel logic & compliance

* **Email:** link tracking, UTM injection; hard bounce/complaint auto‑suppression.
* **SMS (Twilio):** STOP/UNSUB/START/HELP keywords; inbound auto‑tag; reply‑driven branching (Question Response).
* **Voicemail Drop:** ElevenLabs TTS generates MP3 (version‑hashed); voicemail provider API for drop + delivery status; retries with backoff.
* **E‑sign:** internal signer link; statuses: sent/viewed/signed/declined/expired; reminders at +1d for “No Signature”.
* **Calendly:** webhook on scheduled/canceled; store meeting time and type.
* **Podio:** on signed, upsert Person/Deal (map fields below).

5) Data model (campaign‑specific fields)

**Contact attributes**

* rsvp\_status (none|invited|confirmed|declined|canceled)
* event\_date (date), event\_type (virtual|in\_person)
* meeting\_link (Calendly), meeting\_time
* attendance\_status (attended|no\_show|canceled|unknown)
* selection (select1|select2|select3|null)
* question\_flag (bool), question\_text (latest freeform)
* esign\_status (sent|viewed|signed|declined|expired)
* podio\_record\_id (nullable)

**Events to capture**

* email\_sent/open/click/reply
* sms\_sent/delivered/reply/stop
* voicemail\_queued/delivered/failed
* rsvp\_invited/confirmed/declined
* calendly\_scheduled/canceled
* attendance\_marked (attended|no\_show|canceled)
* selection\_made (1/2/3)
* esign\_sent/viewed/signed/declined/expired
* podio\_upserted

6) Integrations (endpoints you’ll implement)

* **Email provider:** /send, webhooks for delivered, bounce, complaint, open, click
* **Twilio SMS:** outbound, DLR + inbound webhook with keyword router
* **ElevenLabs TTS:** synthesize (voice\_id configurable), cache asset
* **Voicemail platform:** create drop, status callbacks
* **Calendly:** webhook subscription for invitee.created / invitee.canceled
* **Podio:** upsert on **Signs** with mapping (contact, event, esign doc id)

7) Dashboards (MVP)

**A. Funnel overview (campaign)**

* Enrolled → Messaged (E1/S1/VM1) → Responded (pos/neg/question) → RSVP confirmed → Attended → eSign sent → Signed → Podio created.

**B. Channel performance**

* Email 1/2, SMS 1/2, Voicemail 1/2 delivery & engagement; **Big Email During Event**; **Biggie‑Small SMS**; **Text Confirmation 8am**.  
   Event Campaign Flow - ADTV

**C. RSVP & scheduling**

* Calendly conversions, confirmation lag, no‑show predictions.

**D. Event‑day ops**

* Live counts: confirmed, attended, cancellations, no shows; response to “questions” CTA.

**E. Agreement/Revenue proxy**

* E‑sign conversion by segment; follow‑up effectiveness (+1d).

**F. Compliance & health**

* STOP rate, hard bounces, voicemail failures, retry queues.

8) Workflow JSON (seed for this exact campaign)

Condensed to core steps; you can expand with full retry/quiet hours in code. Node IDs match the catalog above.

{

"schema\_version": 1,

"nodes": [

{"id":"N00","type":"start","name":"Event Master List"},

{"id":"N11","type":"email\_send","name":"Email 1","config":{"template\_id":"tmpl\_email\_1"}},

{"id":"N12","type":"sms\_send","name":"SMS 1","config":{"template\_id":"tmpl\_sms\_1"}},

{"id":"N13","type":"voicemail\_drop","name":"VM Drop 1","config":{"tts":{"provider":"elevenlabs","voice\_id":"v\_default","script\_template\_id":"tmpl\_vm\_1"},"drop":{"provider":"vm\_provider"}}},

{"id":"N14","type":"web\_request","name":"Pull Analytics","config":{"endpoint":"internal://analytics/pull","method":"POST"}},

{"id":"N15","type":"decision","name":"Check Data","config":{"rules":[

{"label":"Pos Response","expr":{"==":[{"var":"contact.rsvp\_intent"},"positive"]},"to":"N20"},

{"label":"Question Response","expr":{"var":"contact.question\_flag"},"to":"N52"},

{"label":"Neg Response","expr":{"==":[{"var":"contact.rsvp\_intent"},"negative"]},"to":"N59"},

{"label":"No Response","expr":{"==":[{"var":"contact.rsvp\_intent"},null]},"to":"N31"}

]}},

{"id":"N20","type":"email\_send","name":"RSVP Email","config":{"template\_id":"tmpl\_rsvp"}},

{"id":"N31","type":"email\_send","name":"Email 2","config":{"template\_id":"tmpl\_email\_2"}},

{"id":"N32","type":"sms\_send","name":"SMS 2","config":{"template\_id":"tmpl\_sms\_2"}},

{"id":"N33","type":"voicemail\_drop","name":"VM Drop 2","config":{"tts":{"provider":"elevenlabs","voice\_id":"v\_default","script\_template\_id":"tmpl\_vm\_2"},"drop":{"provider":"vm\_provider"}}},

{"id":"N50","type":"web\_request","name":"Virtual Event – Calendly Link","config":{"endpoint":"calendly://send\_link"}},

{"id":"N51","type":"web\_request","name":"In Person – Email Coordinated/BDR","config":{"endpoint":"internal://bdr/coordinate"}},

{"id":"N53","type":"webhook\_wait","name":"Confirmation via Calendly","config":{"provider":"calendly","event":"invitee.created","timeout":"P3D","on\_timeout":"N52"}},

{"id":"N52","type":"task","name":"BDR Responds","config":{"assignee\_role":"BDR"}},

{"id":"N54","type":"task","name":"BDR Confirmation","config":{"assignee\_role":"BDR","mark":"confirmed"}},

{"id":"N60","type":"stage","name":"Event Day"},

{"id":"N63","type":"sms\_send","name":"Text Confirmation 8am","config":{"template\_id":"tmpl\_sms\_8am"}},

{"id":"N62","type":"email\_send","name":"Big Email During Event","config":{"template\_id":"tmpl\_email\_live"}},

{"id":"N64","type":"sms\_send","name":"Biggie-Small SMS","config":{"template\_id":"tmpl\_sms\_biggie"}},

{"id":"N70","type":"decision","name":"Post-Event Outcomes","config":{"rules":[

{"label":"no shows","expr":{"==":[{"var":"contact.attendance\_status"},"no\_show"]},"to":"N71"},

{"label":"cancellations","expr":{"==":[{"var":"contact.attendance\_status"},"canceled"]},"to":"N73"},

{"label":"attended\_no\_response","expr":{"==":[{"var":"contact.attendance\_status"},"attended"]},"to":"N81"}

]}},

{"id":"N71","type":"email\_send","name":"No Shows Email","config":{"template\_id":"tmpl\_email\_noshow"}},

{"id":"N73","type":"email\_send","name":"Cancellations Email","config":{"template\_id":"tmpl\_email\_cancel"}},

{"id":"N77","type":"email\_send","name":"Click #2 – Questions/Calendly","config":{"template\_id":"tmpl\_email\_questions"}},

{"id":"N78","type":"email\_send","name":"Final Email","config":{"template\_id":"tmpl\_email\_final"}},

{"id":"N81","type":"task","name":"BDR Outreach","config":{"assignee\_role":"BDR"}},

{"id":"N82","type":"esign","name":"eSign Agreement","config":{"document\_template\_id":"doc\_agreement\_v1","reminders":{"enabled":true,"interval":"P1D","max":1}}},

{"id":"N83","type":"goal","name":"Signs"},

{"id":"N84","type":"web\_request","name":"Podio Entry Created","config":{"endpoint":"podio://upsert"}},

{"id":"N86","type":"wait","name":"eSign Follow Up +1d","config":{"duration":"P1D"}},

{"id":"N59","type":"exit","name":"Stop"},

{"id":"N89","type":"exit","name":"Stop"}

],

"edges":[

{"from":"N00","to":"N11"},

{"from":"N11","to":"N12","condition":{"after":"PT10M"}},

{"from":"N12","to":"N13","condition":{"after":"P1D"}},

{"from":"N13","to":"N14"},

{"from":"N14","to":"N15"},

{"from":"N20","to":"N50"},

{"from":"N31","to":"N32","condition":{"after":"PT10M"}},

{"from":"N32","to":"N33","condition":{"after":"P1D"}},

{"from":"N50","to":"N53"},

{"from":"N53","to":"N54","condition":{"on":"success"}},

{"from":"N54","to":"N60"},

{"from":"N60","to":"N63","condition":{"at\_local":"08:00"}},

{"from":"N60","to":"N62"},

{"from":"N62","to":"N64","condition":{"after":"PT1H"}},

{"from":"N60","to":"N70","condition":{"after":"PT4H"}},

{"from":"N83","to":"N84"},

{"from":"N82","to":"N83","condition":{"on":"signed"}},

{"from":"N82","to":"N86","condition":{"on":"no\_signature"}},

{"from":"N86","to":"N82"},

{"from":"N71","to":"N82"},

{"from":"N73","to":"N82"},

{"from":"N81","to":"N82"},

{"from":"N83","to":"N89"}

],

"start\_rules":{

"enroll":[{"type":"segment","segment\_id":"event\_master\_list"}],

"reentry\_policy":"single\_entry",

"quiet\_hours":{"start":"21:00","end":"08:00","contact\_timezone":true},

"caps":{"email\_per\_day":2,"sms\_per\_day":1,"voicemail\_per\_day":1}

}

}

Notes: The JSON keeps the essence of your diagram (two pre‑event campaigns; analytics‑driven branching; RSVP/Calendly; event‑day comms; post‑event branching; e‑sign → Podio). You’ll likely add more edges between the selection branches (Selects #1/#2/#3) and follow‑ups once we confirm the exact copy/logic in those green boxes.

Event Campaign Flow - ADTV

9) Implementation checklist (Cursor-friendly)

* **Workflow engine** with timers, idempotent sends, consent checks, and provider adapters (Email, Twilio, ElevenLabs, Voicemail, Calendly webhooks, E‑sign, Podio).
* **Visual editor** to add/edit/delete nodes and edges; versioning and publish flow.
* **Template manager** (Email/SMS/Voicemail script/E‑sign).
* **Segmentation & enrollment** from **Event Master List** and optional webhooks.  
   Event Campaign Flow - ADTV
* **Dashboards** described above; journey inspector; audit log.
* **Compliance** (STOP/UNSUB, quiet hours, frequency caps).
* **Testing**: end‑to‑end paths for Positive/Negative/Question/No‑response; Calendly confirmations; Event‑day send timing; No‑show/cancellation branches; E‑sign signed vs reminder; Podio upsert.

10) Items I could not 100% read (please confirm text)

I can proceed without these, but a higher‑resolution export (PDF vector or 3000px PNG) would let me lock exact labels and any missing arrows.

Event Campaign Flow - ADTV

1. The full sentence on the node beginning **“Click #2 – Send ‘What are your questions’ / Calendly link. Have …”** (seems truncated; I captured as N77).
2. Exact intent of **“Selects #1 / #2 / #3”** (are these meeting types, offer bundles, or content paths?).
3. Whether **Campaign 2** also fans into **Pull Analytics / Check Data** or goes directly to RSVP paths.
4. Any implicit loopback from **Nurture (Sign Up Date Based)** to RSVP/Calendly.
5. Whether **BDR Confirmation** always precedes **Event Day**, or only for in‑person.
6. Any additional **Stop** nodes not labeled near the RSVP lane.

If you can re‑export at higher resolution, I’ll snap these labels exactly and update the JSON and node catalog accordingly—everything else is already captured above.

Event Campaign Flow - ADTV

Bottom line

* **Yes**: I can read and have enumerated the steps and nodes, with edges and timing, from your attached flow chart.
* You now have a **campaign‑specific scope + seed JSON** suitable for a Cursor build, covering functionality, workflow details, integrations, and dashboards.
* Send a higher‑res export if you want me to lock those few ambiguous labels, and I’ll finalize the exact strings in the spec.