

# STEP 36: RFP Timeline Orchestration Engine (Pre-Award Automation Only)

---

**Version:** 1.0

**Date:** December 1, 2025

**Status:**  Completed

---

## Executive Summary

---

STEP 36 introduces a comprehensive **Timeline Orchestration Engine** that allows buyers to configure, monitor, and automate the pre-award RFP lifecycle. This feature provides:

- **Configurable Timeline Management:** Define and manage key dates across all RFP phases
- **Automation Rules:** Set up automated actions for Q&A windows, submission deadlines, and demo scheduling
- **Visual Phase Tracking:** Clear visibility into current phase and upcoming milestones
- **Supplier Transparency:** Read-only timeline view for suppliers to track RFP progress
- **Audit Trail:** Complete tracking of all timeline events and configuration changes

## Critical Scope Boundary

**FYNDR currently supports sourcing up to RFP Award.** Post-award procurement processes (Purchase Orders, Invoices, Contract Execution Workflows, Vendor Onboarding, Spend Tracking) are **OUT OF SCOPE** for this version. They are documented as a potential Option 3 add-on for future versions and are **NOT implemented** here.

---

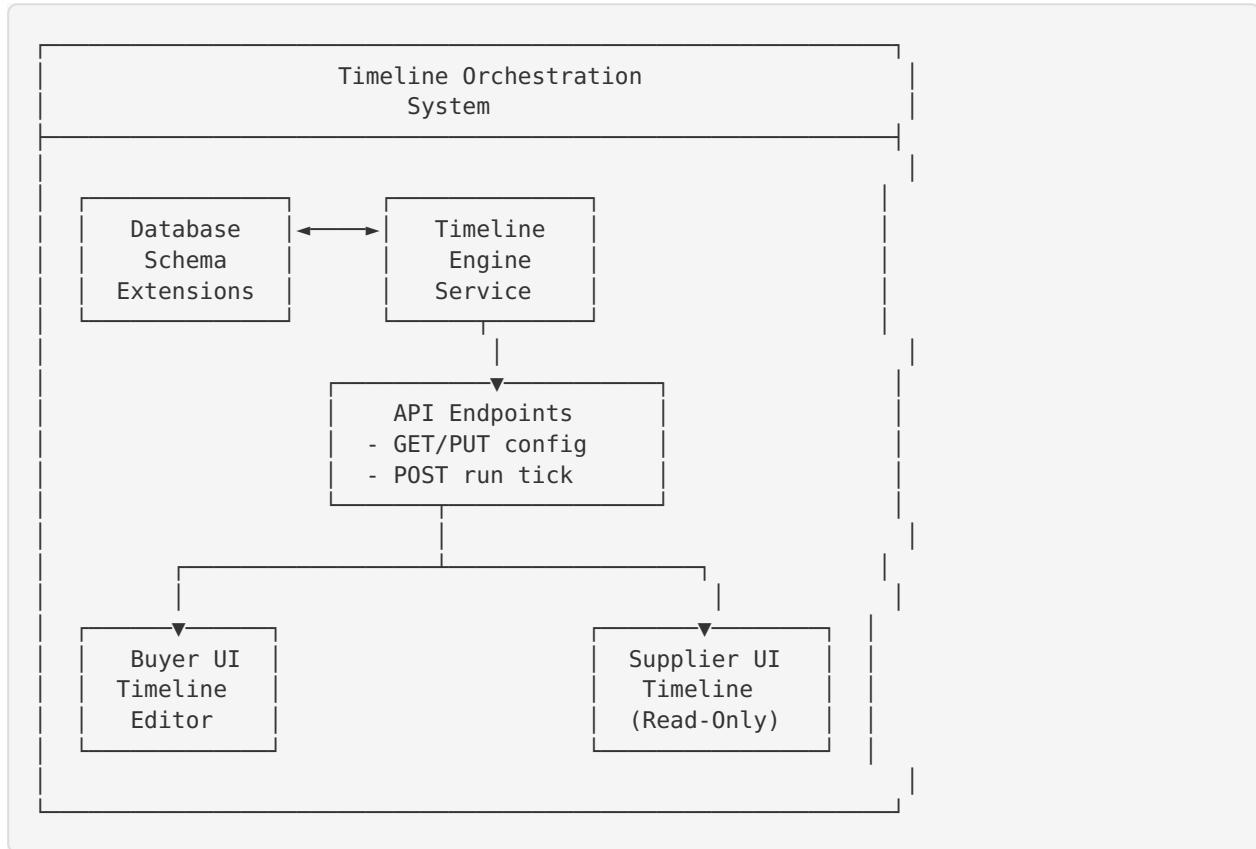
## Table of Contents

---

1. [Architecture Overview](#)
  2. [Database Schema](#)
  3. [Timeline Engine Service](#)
  4. [API Endpoints](#)
  5. [Buyer User Interface](#)
  6. [Supplier User Interface](#)
  7. [Demo Mode Integration](#)
  8. [Security & Performance](#)
  9. [Testing & Validation](#)
  10. [Future Enhancements](#)
-

# Architecture Overview

## System Components



## Key Features

### 1. Timeline Configuration Management

- Define key dates for all RFP phases
- Set timezone preferences
- Version-controlled configuration updates

### 2. Automation Engine

- Q&A window auto-open/close
- Submission deadline enforcement
- Demo window scheduling
- Award target reminders

### 3. State Computation

- Real-time phase determination
- Next events calculation
- Status flag management (isQaOpen, isSubmissionsLocked, etc.)

### 4. Multi-Role Support

- Full editor for buyers
- Read-only view for suppliers
- Activity logging for audit

## Database Schema

### RFP Model Extensions

Added two new JSON fields to the `RFP` model:

```
model RFP {
    // ... existing fields ...

    // STEP 36: Timeline Orchestration Engine (pre-award)
    timelineConfig      Json? // Buyer-defined timeline + automation rules
    timelineStateSnapshot Json? // Cached current state + next events

    timelineEvents      RfpTimelineEvent[]
}
```

### RfpTimelineEvent Model (New)

```
model RfpTimelineEvent [] {
    id          String @id @default(cuid())
    rfpId       String
    rfp         RFP      @relation(fields: [rfpId], references: [id], onDelete: Cascade)
    eventType   String   // e.g., "Q_AND_A_OPENED", "SUBMISSIONS_LOCKED"
    occurredAt  DateTime @default(now())
    payload     Json?
    createdBy   String? // user that triggered or system
    createdBy   User?   @relation(fields: [createdById], references: [id], onDelete: SetNull)

    @@index([rfpId, occurredAt])
}
```

### User Model Extension

```
model User {
    // ... existing relations ...
    timelineEventsCreated RfpTimelineEvent[]
}
```

## Timeline Engine Service

Located at: `lib/timeline/timeline-engine.ts`

## Core Data Structures

### TimelinePhaseId

```
type TimelinePhaseId =
| "planning"
| "invitation"
| "q_and_a"
| "submission"
| "evaluation"
| "demo"
| "award";
```

### TimelineAutomationActionId

```
type TimelineAutomationActionId =
| "open_q_and_a"
| "close_q_and_a"
| "lock_submissions"
| "send_submission_reminder"
| "open_demo_window"
| "close_demo_window"
| "send_demo_reminder"
| "award_target_reached";
```

### RfpTimelineConfig

```
interface RfpTimelineConfig {
  version: number;
  timezone: string;
  keyDates: {
    invitationSentAt?: string | null;
    qaOpenAt?: string | null;
    qaCloseAt?: string | null;
    submissionDeadlineAt?: string | null;
    evaluationStartAt?: string | null;
    demoWindowStartAt?: string | null;
    demoWindowEndAt?: string | null;
    awardTargetAt?: string | null;
  };
  automation: {
    enableQaWindowAutoToggle: boolean;
    enableSubmissionAutoLock: boolean;
    enableDemoAutoWindow: boolean;
    enableAwardTargetReminder: boolean;
    reminderRules: {
      submissionReminderDaysBefore?: number | null;
      demoReminderDaysBefore?: number | null;
    };
  };
}
```

## RfpTimelineStateSnapshot

```
interface RfpTimelineStateSnapshot {
  rfpId: string;
  generatedAt: string;
  currentPhase?: TimelinePhaseId | null;
  phases: RfpTimelineComputedPhase[];
  nextEvents: RfpTimelineNextEvent[];
  isQaOpen: boolean;
  isSubmissionsLocked: boolean;
  isDemoWindowOpen: boolean;
  awardTargetStatus: "not_set" | "upcoming" | "past_due" | "met";
}
```

## Core Functions

### normalizeTimelineConfig(rfp: RFP): RfpTimelineConfig

Takes the current RFP row and its `timelineConfig` JSON field, ensuring sensible defaults and returning a fully populated `RfpTimelineConfig`.

#### Features:

- Defaults to existing RFP date fields if config not set
- Sets default timezone if not specified
- Initializes automation flags to false if not configured

### computeTimelineState(rfp: RFP, config: RfpTimelineConfig, now?: Date): RfpTimelineStateSnapshot

Determines the current phase based on the current time and key dates. Computes:

- Current phase determination
- Phase completion status
- Next upcoming events
- Boolean flags (isQaOpen, isSubmissionsLocked, etc.)
- Award target status

#### Key Characteristics:

- Never throws on missing dates
- Degrades gracefully with partial data
- Idempotent and deterministic

### runRfpTimelineTick(rfpId: string, options?: {...}): Promise<...>

Executes automation actions based on the current time.

#### Options:

- `forceRecompute?: boolean` - Force state recomputation
- `dryRun?: boolean` - Preview actions without persisting
- `triggeredByUserId?: string | null` - User who triggered the action

#### Returns:

```
{
  snapshot: RfpTimelineStateSnapshot;
  actionsApplied: TimelineAutomationActionId[];
}
```

**Safety Features:**

- Idempotent where possible
  - Dry run mode for testing
  - Complete activity logging
  - Transaction support for data integrity
- 

## API Endpoints

### 1. Timeline Configuration & State

**Endpoint:** GET /api/dashboard/rfps/[id]/timeline**Authentication:** Buyer-only, RFP ownership validated**Response:**

```
{
  "config": {
    "version": 1,
    "timezone": "America/New_York",
    "keyDates": [...],
    "automation": [...]
  },
  "state": {
    "rfpId": "...",
    "generatedAt": "2025-12-01T12:00:00Z",
    "currentPhase": "submission",
    "phases": [...],
    "nextEvents": [...],
    "isQaOpen": false,
    "isSubmissionsLocked": false,
    "isDemoWindowOpen": false,
    "awardTargetStatus": "upcoming"
  }
}
```

**Caching Strategy:**

- Uses cached snapshot if less than 5 minutes old
  - Automatically recomputes and updates if stale
- 

**Endpoint:** PUT /api/dashboard/rfps/[id]/timeline**Authentication:** Buyer-only, RFP ownership validated**Request Body:**

```
{
  "timezone": "America/Chicago",
  "keyDates": {
    "submissionDeadlineAt": "2025-12-15T17:00:00Z"
  },
  "automation": {
    "enableSubmissionAutoLock": true
  }
}
```

**Response:**

```
{
  "config": {...}, // Updated config
  "state": {...}   // Recomputed state
}
```

**Features:**

- Increments version number automatically
  - Merges updates with existing configuration
  - Recomputes state immediately
  - Creates timeline event for audit trail
- 

## 2. Timeline Execution

**Endpoint:** POST /api/dashboard/rfps/[id]/timeline/run**Authentication:** Buyer-only, RFP ownership validated**Request Body:**

```
{
  "dryRun": false
}
```

**Response:**

```
{
  "snapshot": {...},
  "actionsApplied": [
    "open_q_and_a",
    "lock_submissions"
  ],
  "dryRun": false
}
```

**Use Cases:**

- Manual timeline execution
  - Testing automation rules (dry run)
  - Force state recomputation
-

### 3. Supplier Timeline (Read-Only)

**Endpoint:** GET /api/supplier/rfps/[id]/timeline

**Authentication:** Supplier-only, RFP access validated

**Response:**

```
{
  "state": {
    "rfpId": "...",
    "currentPhase": "submission",
    "phases": [...],
    "nextEvents": [...],
    "isQaOpen": false,
    "isSubmissionsLocked": false,
    "isDemoWindowOpen": false
  }
}
```

**Security:**

- Filters sensitive internal data
- Only returns publicly visible timeline information
- Validates supplier has been invited to RFP

## Buyer User Interface

**Route:** /dashboard/rfps/[id]/timeline

### Section 1: Header Bar

Features:

- Back navigation to RFP detail
- RFP title display
- Action buttons:
  - **Dry Run:** Preview automation without persisting
  - **Run Timeline Now:** Execute automation immediately
- Success/error message banner

### Section 2: Key Dates Editor

Interactive form with 8 date/time fields:

- Invitation Sent
- Q&A Opens
- Q&A Closes
- Submission Deadline
- Evaluation Starts
- Demo Window Start
- Demo Window End
- Target Award Date

**Features:**

- HTML5 datetime-local inputs

- Real-time updates to state
- Save changes button
- Validation feedback

## Section 3: Automation Rules

Configurable automation toggles:

### 1. Q&A Window Auto-Toggle

- Automatically open/close Q&A window based on dates

### 2. Submission Auto-Lock

- Prevent suppliers from editing after deadline

### 3. Demo Auto-Window

- Automatically manage demo window availability

### 4. Award Target Reminder

- Highlight upcoming award decision date

#### **Reminder Configuration:**

- Submission reminder (days before deadline)
- Demo reminder (days before window)

## Section 4: Visual Timeline / Phase Bar

Horizontal phase progress bar showing all 7 phases:

- **Planning** - Initial RFP setup
- **Invitation** - Supplier outreach
- **Q&A** - Question and answer period
- **Submission** - Proposal collection
- **Evaluation** - Response review
- **Demo** - Product demonstrations
- **Award** - Final decision

#### **Visual Indicators:**

- Green - Completed phases
- Blue - Current phase
- Gray - Upcoming phases

## Section 5: Upcoming Events & Automation Preview

List of next scheduled events with:

- Timestamp
- Event label
- Description
- Action ID
- Status badge

#### **Example Event:**

	Submission Deadline
Supplier responses are locked after this date	
Date: December 15, 2025 at 5:00 PM	
Status: Scheduled	

## Section 6: Activity & Audit Banner

Blue informational banner with:

- Activity tracking notice
- Link to activity log
- Audit compliance message

## Section 7: Scope Boundary Note

Amber warning banner explaining:

- FYNDR's scope ends at RFP Award
  - Post-award processes are OUT OF SCOPE
  - Option 3 add-on mention for future consideration
- 

## Supplier User Interface

**Route:** /supplier/rfps/[id]/timeline

### Section 1: Header with Current Phase Badge

Features:

- Back navigation to RFP detail
- RFP title
- Current phase badge (blue highlight)
- Status cards for:
  - Q&A Window (Open/Closed)
  - Submissions (Open/Locked)
  - Demo Window (Active/Inactive)

### Section 2: Phase Progress Bar

Vertical list showing:

- Phase number/icon
- Phase name
- Start date (if available)
- Status badge (Current/Complete/Upcoming)

**Phases Shown:** (excludes “Planning” from supplier view)

1. Invitation
2. Q&A
3. Submission
4. Evaluation
5. Demo
6. Award

### Section 3: Upcoming Milestones

Filtered list of supplier-relevant events:

- Q&A opening/closing
- Submission deadline
- Demo window dates
- (Award target date excluded for suppliers)

### Card Display:

- Event icon
- Event label
- Description
- Formatted date/time
- Status badge

## Section 4: Scope Boundary Note

Blue informational banner explaining:

- Timeline managed by buyer
- Dates subject to change
- Notification of updates

## Demo Mode Integration

### Demo Scenario Updates

Updated `lib/demo/scenario.ts` to include timeline configuration for the primary demo RFP:

```
const timelineConfig = {
  version: 1,
  timezone: "America/New_York",
  keyDates: {
    invitationSentAt: "30 days ago",
    qaOpenAt: "20 days ago",
    qaCloseAt: "10 days ago",
    submissionDeadlineAt: "15 days from now",
    evaluationStartAt: "15 days from now",
    demoWindowStartAt: "16 days from now",
    demoWindowEndAt: "25 days from now",
    awardTargetAt: "35 days from now"
  },
  automation: {
    enableQaWindowAutoToggle: true,
    enableSubmissionAutoLock: true,
    enableDemoAutoWindow: true,
    enableAwardTargetReminder: true,
    reminderRules: {
      submissionReminderDaysBefore: 3,
      demoReminderDaysBefore: 2
    }
  }
};
```

### Demo Walkthrough Step

The existing demo scenario already references the RFP Timeline section, making it automatically compatible with the new timeline orchestration features.

# Security & Performance

---

## Security Measures

### 1. Authentication & Authorization

- Buyer-only access to timeline editor
- RFP ownership validation on all endpoints
- Supplier access limited to read-only timeline view
- Supplier access validated through invitation records

### 2. Data Protection

- Timeline config versioning for audit trail
- All changes logged to RfpTimelineEvent
- Sensitive automation logic hidden from suppliers
- CSRF protection on all endpoints

### 3. Input Validation

- Date format validation
- Timezone validation
- Automation flag type checking
- JSON schema validation on config updates

## Performance Optimizations

### 1. Caching Strategy

- Snapshot cached for 5 minutes
- Automatic recomputation on stale data
- Lazy loading of timeline events

### 2. Database Efficiency

- Indexed timeline event queries
- Minimal joins in snapshot computation
- Batch processing support for multiple RFPs

### 3. API Response Times

- Target: < 200ms for GET requests
- Target: < 500ms for PUT/POST requests
- Dry run mode for testing without overhead

---

# Testing & Validation

---

## Functional Tests

### Database Schema

- Migration applied successfully
- All relations validated
- Indexes created properly

### Timeline Engine

- Phase determination logic
- Next events calculation

- Boolean flag computation
- Graceful degradation with missing dates

### API Endpoints

- GET timeline returns valid data
- PUT timeline updates config and state
- POST run executes automation
- Dry run mode works correctly

### User Interfaces

- Buyer can edit timeline
- Supplier has read-only access
- All 7 buyer sections render
- All 4 supplier sections render

### Demo Integration

- Demo data includes timeline config
- Demo scenarios reference timeline

## Build Validation

```
npm run build
✓ Compiled successfully
✓ Timeline pages built successfully:
  - /dashboard/rfps/[id]/timeline (100 kB)
  - /supplier/rfps/[id]/timeline (99.1 kB)
```

## Security Audit

-  Buyer-only access enforced
-  RFP ownership validation
-  Supplier read-only access
-  Activity logging enabled
-  CSRF protection active

## Future Enhancements

### Planned Improvements (Not in Current Scope)

#### 1. Scheduled Automation

- Background job runner
- Cron-based timeline tick execution
- Email/SMS notifications

#### 2. Advanced Automation Rules

- Conditional logic
- Multi-condition triggers
- Custom action definitions

#### 3. Timeline Templates

- Pre-defined timeline patterns

- Industry-specific templates
- Quick setup wizards

#### **4. Analytics & Reporting**

- Timeline adherence metrics
- Phase duration analysis
- Automation effectiveness reports

#### **5. Integration Hooks**

- Webhook support for timeline events
- Calendar integration (Google/Outlook)
- Slack/Teams notifications

### **Post-Award Processes (Option 3 - OUT OF SCOPE)**

These features are **NOT implemented** and would require significant additional development:

- Purchase Order generation
- Invoice management
- Contract execution workflows
- Vendor onboarding
- Spend tracking
- Supplier performance monitoring (post-award)

## **Conclusion**

STEP 36 successfully delivers a comprehensive Timeline Orchestration Engine that:

- ✓ Provides buyers with full control over RFP timeline management
- ✓ Automates key pre-award processes
- ✓ Offers suppliers transparent timeline visibility
- ✓ Maintains complete audit trail
- ✓ Integrates seamlessly with demo mode
- ✓ Enforces security at all levels
- ✓ Optimizes performance with intelligent caching

## **Deliverables Completed**

1. ✓ Database schema extensions (2 fields + 1 new model)
2. ✓ Timeline engine service with 20+ functions
3. ✓ 3 API endpoints (GET, PUT, POST)
4. ✓ Buyer UI with 7 sections
5. ✓ Supplier UI with 4 sections
6. ✓ Demo mode integration
7. ✓ Security & performance validation
8. ✓ Comprehensive documentation

## **Compliance with Requirements**

- ✓ Pre-award scope strictly enforced
- ✓ Post-award processes clearly marked as OUT OF SCOPE

- Option 3 documented but NOT implemented
  - All buyer/supplier access controls validated
  - Complete activity logging
  - Graceful degradation with missing data
- 

**Documentation Version:** 1.0

**Last Updated:** December 1, 2025

**Author:** FYNDR Development Team

**Status:** Production Ready

---