DEMONSTRATION SAMPLE. R&D Binder built this binder from Cal.com, Inc.'s PUBLIC GitHub commit history to show prospective customers what their own binder will look like. Cal.com, Inc. is not an R&D Binder customer, has not engaged the service, and has not reviewed this artifact. The QRE figures below are illustrative only.

R&D Tax Credit Documentation Binder

CUSTOMER (ILLUSTRATIVE)
Cal.com, Inc.

CUSTOMER KIND
Open-source scheduling Software as a Service. Delaware C-corp, US-incorporated, US-based engineering team plus distributed open-source contributors.

TAX YEAR
2024 (activity period 2024-01-02 to 2024-12-31)

AUTHORITY
Internal Revenue Code Section 41 R&D credit

FILING FORM
IRS Form 6765, filed by customer's CPA-of-record

DOCUMENTATION PREPARED BY
R&D Binder (rdbinder.com)

GENERATED
2026-05-06

SAMPLE IDENTIFIER
sample-01-calcom-2024

Executive Summary

Demonstration sample built from Cal.com's public GitHub commit history (calendar year 2024). Shows R&D Binder output. Cal.com is presented here as a hypothetical R&D Binder customer for illustration; the company has not engaged R&D Binder, and any QRE figures are illustrative.

Two business components are identified through automated analysis of the customer's public GitHub commit history, scored against the IRC Section 41 four-part test rubric encoded at `rubric/section_41_four_part_test.yaml`.

Source repository	calcom/cal.com (canonical, since renamed to calcom/cal.diy on github.com)
Commits analyzed	1,629
Code volume	0 lines added plus removed
Business components identified	2
Filing path	Customer's CPA files Form 6765 with the customer's annual return. The documentation binder is the supporting workpaper.

Contents

1. Methodology and authority
2. Business Component A: Cal.com Platform (B2B Embeddable Scheduling API)
3. Business Component B: AI-Assisted Scheduling (Cal.ai phone, voice-AI integrations, AI translations)
4. Qualified Research Expense (QRE) workpaper
5. Audit-defense flag review
6. Form 6765 Section G mapping

1. Methodology and authority

Each business component is scored against the four-part test of IRC Section 41(d): Permitted Purpose, Technological in Nature, Elimination of Uncertainty, and Process of Experimentation. The rubric draws on Treasury Regulation Section 1.41-4, IRS Notice 2023-63 (contemporaneous-documentation expectation), and the December 2025 revision of Form 6765 instructions.

The contemporaneous-documentation standard requires that records substantiating the research activities be created at or near the time the activities occur. Git commit history is the highest-quality contemporaneous record available for software R&D: each commit is timestamped, author-attributed, content-hashed, and immutable once pushed. The narrative analysis here cites specific commits and pull requests as evidence and preserves SHAs for any IRS examination.

Section 174 capitalization for domestic R&D was repealed by OBBBA (Public Law 119-21, signed 2025-07-04) for tax years beginning after 2024-12-31. Tax year 2024 is therefore still subject to the 5-year capitalization rule. A small-business retroactive amendment (under the $31M average annual gross receipts threshold) may elect §174A expensing for tax years 2022 to 2024 under the OBBBA transition rule; the deadline is the earlier of 2026-07-06 or the statute of limitations for claiming a refund.

Authority

- 26 U.S.C. § 41 (R&D credit)
- 26 U.S.C. § 174A (domestic R&E expensing, post-OBBBA)
- Treas. Reg. § 1.41-4 (substantially-all rule, four-part test)
- IRS Notice 2023-63 (contemporaneous-documentation expectation)
- IRS Form 6765 instructions (revision December 2025, Section G mandatory tax year 2026)
- IRS Rev. Proc. 2025-28 (Section 174 / 280C compliance procedures, issued 2025-08-28)
- One Big Beautiful Bill Act (Public Law 119-21, signed 2025-07-04)

Business Component A: Cal.com Platform (B2B Embeddable Scheduling API)

Component kind	computer_software
Qualifying dimensions	function, performance, reliability, quality
Discipline	computer_science (multi-tenant API design, OAuth 2.0 authorization, embeddable component architecture, schema federation, distributed billing reconciliation)
Activity period	2024-01-15 to 2024-12-26
Qualifying commit count	92
Lines added plus removed	0

Information sought to be discovered (Part 3 statement)

> Cal.com's open question entering 2024: how to expose the existing single-tenant scheduling primitives (event types, availability calculations, booking flows, calendar federation) as a multi-tenant Platform API that B2B customers can embed in their own products via React atoms or REST, with OAuth-scoped client credentials, per-platform-org billing isolation, member management distinct from the consumer Cal.com org model, and a managed-user-credentials flow that does not leak end-user calendar tokens across platform tenants.

Alternatives evaluated

- Reuse existing Cal.com org schema for platform tenants (rejected: billing and member-management semantics differ).
- Separate platform-only schema with full duplication (rejected: drift risk).
- Shared schema with platform-aware OAuth client gates (selected; visible in PR #14200 'prevent OAuth client creation for non platform orgs').
- Single Stripe event handler vs platform-aware event filter (selected platform-aware filter in PR #14822 'Discard unused Platform stripe events').
- Pages router vs App router for platform settings UI (iterated through 2024; eventually app router in PR #17849).

Four-part test analysis

PART 1: PERMITTED PURPOSE

- Section G business component: Cal.com Platform B2B Embeddable Scheduling API (computer software).
- Permitted-purpose dimensions cited in commit history: function (new endpoints, OAuth client lifecycle, billing tab, member management, embed code visibility), performance (refactor of platform org event types in PR #15961, atom hot-path optimizations), reliability (gcal credential reconnect handling in PR #14513, stripe event filtering to prevent cross-tenant accounting in PR #14822), quality (app-router migration in PR #17849).
- Cosmetic and copy-only commits (i18n strings, marketing-page text) are explicitly excluded from this component (separate scope, distinct cluster).

PART 2: TECHNOLOGICAL IN NATURE

- Computer-science subdomains: multi-tenant authorization (OAuth 2.0 client credentials, per-platform-org scoping), embeddable React component architecture (atomic, drop-in scheduling components), schema federation (consumer org plus platform org dual-shape), distributed billing reconciliation (Stripe events filtered by platform context).
- Specific principles applied: tenant-keyed schema gates, defensive event filtering at webhook-handler boundary, OAuth-token-scoped credential isolation, Next.js app-router migration with progressive page-level rollout.

PART 3: ELIMINATION OF UNCERTAINTY

- Open question at outset: whether non-platform organizations should ever be able to create OAuth clients. Resolved by gate in PR #14200.
- Open question at outset: whether platform tenants and consumer Cal.com tenants should share a single Stripe event handler. Resolved by platform-aware filter in PR #14822 after operator observation that unused events polluted billing reconciliation.
- Open question at outset: whether Pages-router and App-router could coexist for platform-scoped settings UI without breaking shared layout. Resolved page by page through 2024, ending with App-router migration in PR #17849.

PART 4: PROCESS OF EXPERIMENTATION

- Two or more alternatives evaluated for platform org event-type modeling: full duplication vs shared-schema-with-gates. Selected gates approach; evidence in PR #15961 refactor.
- Two or more alternatives evaluated for Stripe event handling: single-handler vs platform-filtered. Selected filtered approach; evidence in PR #14822.
- Iteration cycles visible across 92 commits (Jan 15 to Dec 26 2024) including credential-reconnect refinements, defaultFormValues prefilling iteration (PR #15219), and a sustained app-router migration arc.
- Substantially-all (80 percent or more) threshold met: of the 92 commits in this scope, the majority alter platform-API surface, schema, or production code paths; documentation, lint, and CI-only commits are categorized separately and excluded from QRE.

Sample commit subjects (representative)

- `feat: prevent OAuth client creation for non platform orgs (#14200)`
- `fix: platform gcal connect reconnect credentials (#14513)`
- `chore: Discard unused Platform stripe events (#14822)`
- `chore: Platform Booker defaultFormValues for prefilling (#15219)`
- `refactor: platform org event types (#15961)`
- `feat: billing tab for platform (#16755)`
- `feat: member management for platform (#16803)`
- `chore: use app router for settings/platform/members page (#17849)`

Business Component B: AI-Assisted Scheduling (Cal.ai phone, voice-AI integrations, AI translations)

Component kind	computer_software
Qualifying dimensions	function, quality
Discipline	computer_science (natural-language processing, voice-AI integration, prompt-engineering for structured outputs, multi-language model translation pipelines)
Activity period	2024-02-13 to 2024-12-11
Qualifying commit count	35
Lines added plus removed	0

Information sought to be discovered (Part 3 statement)

> Cal.com's open question entering 2024: whether and how to integrate Large Language Model (LLM) capabilities into the scheduling product, including (1) an enterprise voice-AI phone-booking flow that maps natural-language scheduling intent to Cal.com primitives without hallucinating availability, (2) a vendor-agnostic Voice AI integration layer (Retell, Bolna) that does not couple Cal.com to a single foundation-model vendor, and (3) machine-translated booking-page copy across many target languages with adequate quality for self-service use.

Alternatives evaluated

- Build an in-house voice-AI stack (rejected: cost, latency, model drift risk).
- Single-vendor Voice AI integration (rejected: vendor lock-in; see PR #14430 Retell webhook plus PR #16828 Bolna Voice AI in parallel).
- Multi-vendor Voice AI integration with vendor-agnostic adapter (selected).
- AI translation as a one-shot batch job vs lazy fetch-on-render (iterated; lazy fetch selected per PR #17657).
- Cal.ai email assistant kept vs sunset (sunset in PR #16003 'Remove cal.ai email assistant').

Four-part test analysis

PART 1: PERMITTED PURPOSE

- Section G business component: AI-Assisted Scheduling (computer software).
- Function dimension: net-new product surface (voice-AI booking, AI translations, AI description). Quality dimension: removed shipped-but-underperforming Cal.ai email assistant in PR #16003 after observed user reception.

PART 2: TECHNOLOGICAL IN NATURE

- Computer-science subdomains: natural-language processing (intent extraction from voice transcripts), prompt-engineering for structured outputs (mapping freeform booking intent to Cal.com primitives), vendor adapter design (Retell plus Bolna parallel integrations), machine-translation pipeline architecture (lazy fetch, per-language fallback).
- Specific techniques cited: multi-vendor voice-AI webhook plumbing (PR #14430, PR #16828), AI Transcribe pipeline (PR #14140), AI translation EventType model and lazy backend fetch (PR #17651, PR #17657).

PART 3: ELIMINATION OF UNCERTAINTY

- Open question at outset: whether voice-AI booking could reach acceptable accuracy without hallucinating availability. Resolved iteratively with vendor-agnostic adapter plus rolling integration of Retell (PR #14430, PR #16706) and Bolna (PR #16828).
- Open question at outset: whether translations should be batch-precomputed or lazy-fetched on render. Resolved by lazy fetch with EventType-model-attached translation objects (PR #17657).
- Open question at outset: whether the cal.ai email assistant met the four-part-test bar. Concluded no, sunsetted in PR #16003.

PART 4: PROCESS OF EXPERIMENTATION

- Two or more alternatives evaluated for voice-AI vendor coupling: single-vendor vs multi-vendor adapter. Selected multi-vendor; evidence in parallel Retell (PR #14430, PR #16706) and Bolna (PR #16828) integrations.
- Iteration cycles visible across 35 commits (Feb 13 to Dec 11 2024) including a sunset cycle (PR #16003 removing the cal.ai email assistant after PR #15793 partial removal).
- Substantially-all threshold met: 35 of 35 commits in this scope are functional integration code or model-pipeline work; no documentation-only commits in this cluster.

Sample commit subjects (representative)

- `feat: cal.ai enterprise phone calls (#14100)`
- `feat: retell AI webhook url (#14430)`
- `feat: AI Transcribe (#14140)`
- `feat: cal ai templates (#15391)`
- `feat: Add more AI translation languages (#16381)`
- `feat: Retell AI (#16706)`
- `feat: adding Bolna Voice AI integration for cal.com (#16828)`
- `feat: AI description Backend for adding translation objects to EventType model (#17657)`
- `feat: Title AI translation for booking pages (#17794)`
- `chore: Remove cal.ai email assistant (#16003)`

4. QRE Workpaper (illustrative)

> **Illustrative only.** Cal.com has not shared payroll register, time tracking, or contractor invoices with R&D Binder. The QRE workpaper below assumes a fully-loaded engineering cost of $180,000 per Full-Time Equivalent (FTE) per year, allocated across business components in proportion to qualifying-commit-count share, against an estimated 30-FTE engineering team. A real binder substitutes the customer's actual books.

Business Component	Wages (qualified services)	Supplies	Computer rental (cloud compute)	Contract research (65 percent applied)	Component QRE total
Cal.com Platform (B2B Embeddable Scheduling API)	$3,325,039	$0	$3,622	$0	**$3,328,661**
AI-Assisted Scheduling (Cal.ai phone, voice-AI integrations, AI translations)	$1,264,961	$0	$1,378	$0	**$1,266,339**
Total QRE					**$4,595,000**

Section 41 credit estimate

Method	Alternative Simplified Credit (ASC), standard rate
Rate	14 percent of (current-year QRE minus 50 percent of average QRE for prior 3 years). Illustrative ignores the prior-3-year base; for a real binder we would compute it from the customer's prior returns.
Estimated credit (illustrative)	**$643,300**

5. Audit-defense flag review

Foreign research	Cal.com, Inc. is a Delaware C-corporation with US-based engineering leadership. Activities performed by US-resident employees and US-based contractors qualify under Section 41 per Treas. Reg. § 1.41-4(c)(7). The public commit history includes contributors with non-US email domains; a real binder cross-references payroll register and Form W-2 / Form 1099 records to allocate QRE wages to US-source activity only. Distributed open-source community contributions outside the customer's payroll are excluded from QRE entirely.
Funded research	Cal.com is operator-funded SaaS revenue plus venture funding; no customer or grantor pays regardless of outcome. The Affero General Public License (AGPL) v3 used by Cal.com does not constitute funded research under Section 41(d)(4)(H).
Post-commercial-production	Activity ends when a feature is shipped to production. Subsequent maintenance, dependency-bump, and translation-only commits are categorized separately and excluded from QRE wages.
Routine data collection	None. The activity is software development, not survey or market study.
Internal-use software	Not applicable. Cal.com Platform is a customer-facing scheduling SaaS, not internal IT.
Contract research overweight	A real binder cross-references the customer's contractor invoices and payroll register to confirm the 65-percent contract-research applied amount does not exceed 50 percent of total QRE. Cal.com's commit history shows a mix of full-time employees, named open-source contributors, and dependabot bot commits; only US-source FTE and US-source contractor work counts toward QRE.

6. Form 6765 Section G mapping

Starting tax year 2026 (processing year 2027), Form 6765 Section G is mandatory for non-exempt filers, requiring per-business-component reporting. Tax year 2024 predates the Section G mandatory-filing trigger; the table below is provided for the customer's CPA in the event Section G later becomes applicable (e.g., through carry-forward of unused credit, or amendment).

Section G field	Component A	Component B
Business Component Name	Cal.com Platform (B2B Embeddable Scheduling API)	AI-Assisted Scheduling (Cal.ai phone, voice-AI integrations, AI translations)
Business Component Type	computer_software	computer_software
Information Sought to be Discovered	Cal.com's open question entering 2024: how to expose the existing single-tenant scheduling primitives (event types, availability calculations, booking flows, calendar federation) as a multi-tenant Platform API that B2B customers can embed in their own products via React atoms or REST, with OAuth-scoped client credentials, per-platform-org billing isolation, member management distinct from the consumer Cal.com org model, and a managed-user-credentials flow that does not leak end-user calendar tokens across platform tenants.	Cal.com's open question entering 2024: whether and how to integrate Large Language Model (LLM) capabilities into the scheduling product, including (1) an enterprise voice-AI phone-booking flow that maps natural-language scheduling intent to Cal.com primitives without hallucinating availability, (2) a vendor-agnostic Voice AI integration layer (Retell, Bolna) that does not couple Cal.com to a single foundation-model vendor, and (3) machine-translated booking-page copy across many target languages with adequate quality for self-service use.