Product Requirements Document (PRD)
Data Annotation Platform
Contract Addendum

Version: 2.0
Date: 2025-12-31
Status: Final

Confidential

1. Purpose

The Data Annotation Platform is an internal, end-to-end system designed to centralize queueing, task workflows, annotation interfaces, operational tooling, quality processes, and batch-level data export. Today, annotation workflows are distributed across multiple tools and manual processes; this platform consolidates them into a unified environment that is configurable, scalable, and optimized for operational consistency across projects and customers.

This PRD defines the functional scope, system expectations, and acceptance criteria governing vendor implementation. It serves as the authoritative requirements source for engineering vendors participating in the RFQ and forms the foundation for technical proposals, estimation, and alignment with the forthcoming Statement of Work (SOW).

2. Product Definition

2.1 What the Platform Is

The platform provides a structured, multi-layer annotation and review environment that includes:

- **Configurable pipelines and workflow stages**
- **Schema-driven task UIs** supporting multiple data types
- **FIFO queueing, assignment, and claim locking**
- **Rater and reviewer task interfaces** for completing and validating annotations
- **Operational tools** for task upload, export, tagging, and pipeline management
- **Benchmarks, consensus mechanisms, linters, and time-tracking**
- **Secure storage, access control, and audit logging**
- **Support for high-volume, multi-project operations** through the Workspace → Project → Batch → Task → Annotation hierarchy

Customer-facing capabilities include:

- **Organization-level account and subscription management:** Customer Admins can add or remove user seats and view usage summaries (e.g., annotation volume).
- **Dataset access and export:** Customers can securely upload data (for Ops Managers to manage) and download/export completed datasets associated with their projects.
- **API key viewability:** Customer Admins can view their API keys

These customer capabilities focus on **account and data access** — not project setup or operational configuration.

2.2 What the Platform Is Not

Unless explicitly included in the SOW, the platform does **not** provide:

- **Customer-facing project creation, workflow configuration, or operational tooling**, including the ability to create projects, define pipelines, manage batches, upload tasks directly, or operate queueing/assignment settings.
- **End-to-end analytics or advanced reporting dashboards**
- **Automated pre-/post-processing or ML-driven prediction workflows**
- **Deep integrations with internal systems not specified here**
- **Phase 3 optional capabilities**, such as drafts/autosave, advanced filters, custom grading, notification center, additional assignment strategies, and related enhancements

Customer-facing account controls (user seats, dataset export, API credential management) are explicitly **limited to access and data**, and do not grant visibility into or control over internal project configuration or operational workflows.

The table below clarifies the distinction between Customer-facing account and data-access capabilities and the internal - Data-only operational and project-management capabilities.

Customer vs. Internal Capabilities

Area	Customer Capabilities	Internal (- Data) Capabilities
Account & Access Management	• Manage organization-level subscription and user seats (add/remove users) • View their API keys	• Provision internal roles (Ops, QA, Admin) • Configure access boundaries and permission models
Data & Deliverables	• Download/export completed datasets	• Manage export configurations, batch structures, delivery schemas • Validate, audit, and re-export data
Usage Visibility	• View high-level consumption metrics (e.g., annotation volume)	• Full operational analytics (pipeline status, task progress, rater activity, quality signals)
Project & Workflow Control	**Not available.** Customers cannot create projects, configure pipelines,	• Create and configure Workspaces, Projects, Batches

	manage batches, upload tasks, or operate workflow tools.	• Define pipelines, schemas, assignment strategies • Upload task data and manage task lifecycle
Operational Tools	**Not available.** Customers do not have access to queueing, assignment, tagging, benchmarks, consensus, or Ops UI.	• Ops Portal: task upload, tagging, reset/archive, pipeline controls, export generation
Rater/Reviewer Operations	**Not available.** Customers do not interact with internal rater or QA tools.	• Full rater and reviewer interfaces, QA workflows, consensus/benchmark controls
System Integrations	• Use approved APIs for data submission and results retrieval	• Build and maintain integrations with Okta, Datadog, S3, CI/CD, monitoring, internal engineering systems
Security & Compliance	• Secure access to their own data and API credentials	• Enforce encryption standards, audit logs, retention policies, PII handling, compliance controls

3. Users & Core Workflows

3.1 Personas (In order of Priority)

- **Ops Manager / Project Lead / Internal User**: configures projects, uploads tasks, assigns raters, advances pipeline stages, exports results.
- **Annotator (Rater)**: claims or receives assignments, completes tasks in the task UI, views benchmark / feedback where enabled.
- **Reviewer / QA**: reviews annotations at higher layers, provides approvals or rejections, triggers workflow outcomes.
- **Product Admin**: manages internal vs external access boundaries, user accounts, and platform configuration.
- **Customer / Client Admin**: has visibility into the task statuses and can export results, optionally connect an API to import/export.
- **Engineering / DevOps**: maintains infrastructure, CI/CD, and monitoring.

3.2 Key User Journeys

User Journeys in this section reflect the end-to-end interactions of each platform persona. Journeys or individual steps that include an explicit phase label—(Phase 2) or (Phase 3)—represent capabilities

delivered beyond the Phase 1 MVP. All items without a phase label are delivered as part of Phase 1. This notation is intended to clarify scope boundaries and align user workflows with the phased requirements defined in Section 8.

Ops Manager / Project Lead:

1. **Project Setup & Launch:** The Ops Manager configures and launches a new labeling project based on customer requirements, enabling complex workflows to be setup and launched without engineering involvement.
    a. Journey:
        i. Create project
        ii. Define multi-stage pipeline (e.g., L1 → Review → Audit → Done → Delivered)
        iii. Configure stage-specific modular task UIs
        iv. Assign annotators and reviewers by role, tags, and/or level ***[Phase 3]***
        v. Configure quality gates (benchmarks, agreement % thresholds, review routing) ***[Phase 2]***
        vi. Select pre- and post-processing options ***[Phase 2]***
        vii. Upload batch of task data in bulk via UI
2. **Complex & Role Based Reviews:** The Ops Manager configures different task UIs and schemas per pipeline stage so that annotators, reviewers, and auditors see only role-appropriate information.
    b. Journey:
        i. Configure annotator-only fields (fields that will not be visible in the review stage)
        ii. Add review-only sections (fields that will only be visible to reviewers)
        iii. Define shared fields that persist across stages (visible to all)
3. **Task Assignment & Serving:** The Ops Manager assigns annotators and reviews to projects in order to serve tasks and to control workload distribution and respond to operational needs at scale.
    c. Option to preform action:
        i. Ops manager bulk assign raters on a project, batch, and/or pipeline-stage basis
        ii. Manual assignment or reassignment on individual tasks basis
4. **Dynamic Workflows & mid-project Changes:** The Ops Manager adapts workflows mid-project to accommodate changing customer requirements without disrupting production.
    d. Journey:
        i. Clone project configurations within a project-set (defined project grouping)
        ii. Modify UI schemas, pipelines, and/or quality rules in the new project
            1. UI/pipeline modifications
            2. Quality rules (benchmarks, consensus, linters) ***[Phase 2]***
        iii. Move selected tasks between projects ***[Phase 2]***

5. **Golden task/benchmark creation:** The Ops Manager creates and manages golden tasks and benchmarks to measure and enforce rater quality throughout the pipeline.
    e. Journey:
        i. Workflow A: Pre-Labeled Golden Tasks ***[Phase 2]***
            1. Tasks are uploaded with responses pre-populated ***[Phase 2]***
            2. Tasks are flagged as Golden at upload time ***[Phase 2]***
            3. Golden tasks are injected at a configurable rate ***[Phase 2]***
        ii. Workflow B: Post-Creation Golden Tasks
            1. Standard tasks are uploaded
            2. Project leads annotate selected tasks
            3. Responses are marked as Golden post hoc ***[Phase 2]***
6. **Visibility:** The Ops Manager monitors project progress and exports results to support delivery tracking, quality analysis, and customer reporting
    a. Journey:
        i. Track tasks and annotations statuses across project-sets within the platform ***[Phase 2]***
        ii. Filter tasks by status, tags, rater, and/or pipeline stage
            1. Standard filtering ***[Phase 2]***
            2. Advanced/custom filtering ***[Phase 3]***
        iii. Then export results as CSV or JSON, including associated assets ***[Phase 2]***
7. **Configurable Quality Linters:** The Ops Manager configures programmatic quality linters to automatically validate task responses and enforce quality standards during task submission and review.
    a. Journey: ***[Phase 2]***
        i. Select which linters apply per project ***[Phase 2]***
        ii. Configure linter behavior (blocking vs warning) ***[Phase 2]***
        iii. Tune linter for specific project (defined withing the linter build) ***[Phase 2]***
        iv. Optionally surface linter feedback to raters ***[Phase 2]***
        v. Store linter results linked to user and annotation ***[Phase 2]***

Other Internal Users (e.g., Ops/Customer Engineering):

1. Data processing:
    a. Journey
        i. Engineers create custom pre-processing scripts in AWS Lambda to convert customer data input into the [new platform] task upload format. ***[Phase 2]***
        ii. Engineers build post-processing scripts in AWS Lambda to convert project-set data back into the customer's custom format. ***[Phase 2]***

iii. Lambda URLs are added to the project-set configuration to automatically trigger these actions during project upload and download steps. ***[Phase 2]***

Annotators and Reviewers:

2. **Annotator Tasking:**
    a. **Journey:**
        i. Annotators log into their portal to access their pre-assigned task queue
        ii. They complete tasks according to provided instructions and UI capabilities.
        iii. Tasks are submitted upon completion.
        iv. Annotators are blocked or warned if automated checks fail, or required fields are left blank. ***[Phase 2]***
        v. Once a task is submitted, the next task in the queue is served, if available.
3. **Reviewer Tasking:**
    a. **Journey:**
        i. Reviewers log into their task queue to access tasks completed by annotators.
        ii. They can update responses they believe are incorrect or approve responses as-is.
        iii. Reviewers answer any reviewer-only questions.
        iv. They have the option to leave comments for annotators. ***[Phase 3]***
        v. Reviews are submitted upon completion.

Product Admin:

1. **Permission management:** Product admin user bulk actions to manage permissions at scale
    a. Examples:
        i. Create customer accounts and user accounts
        ii. Manage permissions (customer, annotator, reviewer, and internal)
        iii. Provide and revoke API permissions to internal and customer user
2. **Rater and Reviewer Impersonation**:
    a. Journey: ***[Phase 3]***
        i. Rater Reports an issue with their system.
        ii. Product admins/support admins log into the platform and go to that rater's user page to impersonate the user and view the platform from their account ***[Phase 3]***
        iii. They follow the actions reported by the user to reproduce the issue ***[Phase 3]***

Customer or Client Admin:

1. **Customer Experience:** Customer has access to a simple dashboard or account page where they can take a few specific actions:
    i. Manage their organization's subscription by adding or removing user seats ***[Phase 2]***
    ii. View API keys for programmatic access to submit new data and retrieve labeled results. ***[Phase 2]***
    iii. Review aggregated annotation volume consumption
        1. High-level consumption ***[Phase 2]***
        2. Detailed analytics ***[Phase 3]***
    iv. Import data for Ops Managers to manage for projects (see Project Set-up and Launch)
        1. API only
        2. Customer UI ***[Phase 3]***
    v. Export completed datasets ***[Phase 3]***

4. Scope Summary

4.1 Functional Scope

Category	In Scope	Out of Scope
Annotation Workflows	Multi-layer pipelines; configurable statuses; review flows; Ops controls	Automated ML-driven pre/post-processing; advanced analytics
Queueing & Assignment	FIFO queueing; manual/automatic assignment; deterministic claim locking	Random/LIFO/weighted assignment unless explicitly included
Task UI	Media rendering; core response components; level-specific UIs (annotator/reviewer)	Drafts/autosave; instructional pages; notifications (optional Phase 3)
Ops Tooling	Task upload; export; reset/archive/ignore; tagging; bulk assignment	Full operational dashboards; advanced filtering (optional)
Quality & Review	Benchmarks; consensus workflows; linters; time tracking	Deep QA analytics or ML-based quality scoring
Data Model	Workspace → Project → Batch → Task → Annotation; CRUD APIs	External data modeling frameworks; cross-system schema management

Rater Portal	Rater access, task view, submission	-Works routing unless explicitly selected
Customer Account & Data Access	Subscription & seat management; dataset upload and export; view their own API key	Customer project creation; workflow configuration; task uploads; queue/assignment controls
Project Configuration	Internal setup of Workspaces, Projects, Batches; pipeline & UI schema configuration; project cloning	Customer-facing project creation; unrestricted schema modification on in-progress tasks
Processing Hooks	Configuration of pre/post-processing triggers (e.g., external Lambda/webhook calls)	Internal ML prediction pipelines or automated data transformation

4.2 Technical Scope

Category	In Scope	Out of Scope
Architecture	Modular service + UI component model; required architecture diagrams; clear extensibility patterns	Integration with unrelated internal platforms; enterprise-wide architecture rework
Backend / Frontend	Node.js/Express backend; REST APIs; React/TypeScript/Tailwind UI	Alternate stacks/frameworks
Storage	PostgreSQL (transactional), MongoDB (schema-flexible), S3 object storage	Additional datastores or data lakes not listed
Infrastructure & CI/CD	AWS dev/staging/prod environments; GitHub Actions pipelines (build/test/deploy)	Custom CI/CD systems; provisioning outside platform scope
Security	Okta SSO (OAuth2); RBAC; encryption in transit/at rest; audit logging conventions	Pen-tests or security certifications not included in SOW
APIs	REST APIs for ingest and retrieval; token-based auth; OpenAPI documentation	GraphQL; custom integration frameworks
Observability	Datadog logs, metrics, dashboards, alerts; structured logging	SIEM integrations; analytics infrastructure beyond Datadog

Performance & Reliability	Deterministic claim locking; p95 latency targets for UI/API; defined SLO/SLA thresholds	Performance guarantees beyond SOW tiers; multi-region DR unless specified

5. System Expectations

5.1 Scale & Throughput

Vendors must design for - Data's expected operational scale, including:

- Daily task volume (assumptions provided in RFQ)
- Concurrent rater sessions
- Multiple active projects/batches
- Media file size and format handling

Assumptions affecting cost or timeline must be surfaced.

5.2 Performance Requirements

The platform must meet or exceed the following thresholds (final targets confirmed in SOW):

User-interaction-based

- UI load times
- Batch export performance
- API response time baselines
- Linter runtimes

State management

- Claim concurrency: deterministic locking (no double claim) (see QUEUE-01 in Appendix A for more detail)
- Moving tasks between different stages in a pipeline

5.3 Security Requirements

The platform must adhere to - Data security and compliance standards:

- Authentication via Okta SSO (OAuth2)
- Role-based access control with internal vs external separation
- Encrypted storage for all customer and uploaded data
- Signed URLs for asset access
- Audit logging for key events and state transitions
- Least-privilege access principles

5.4 Infrastructure Requirements

- Data provides standardized AWS environments (dev/staging/prod).
Vendor-delivered code must:

- Integrate into existing GitHub Actions pipelines
- Conform to - Data's stack (Node, React, Postgres, Mongo, S3)
- Support S3-structured storage and retrieval patterns
- Follow Datadog logging conventions where applicable

Detailed CI/CD and environment provisioning will be defined in the SOW.

6. High- Level Responsibilities

Area	Vendor Responsibilities	- Data Responsibilities
Platform Implementation	Deliver all in-scope backend, frontend, queueing, pipeline, and UI capabilities	Provide requirements, specifications, design feedback, and acceptance criteria
Infrastructure & Security	Implement Okta auth, RBAC, and storage logic using provided infrastructure	Provision AWS environments, S3 buckets, Okta config, networking, and secrets
Documentation	Provide API documentation, developer setup, user-facing guides	Provide documentation standards/templates and perform required reviews
Testing	Provide unit and integration tests for critical workflows	Provide test data, accounts, and SME support
Operational Readiness	Deploy to staging, support demos and walkthroughs	Perform acceptance, security validation, and production governance

Detailed execution responsibilities, processes, and deliverables will be defined in the SOW.

7. Definitions and Terminology

Term	Definition
Annotation	The structured response submitted for a task by a rater or reviewer.
Assignment / Claim	The mechanism by which a rater receives a task, either automatically or through manual claiming.
Batch	A group of tasks uploaded together within a project, sharing configuration such as pipeline and UI schema.
Benchmark (BM)	A special task used to evaluate rater quality, either as a prerequisite or interleaved within production tasks.
Consensus Workflow	A configuration in which multiple annotations (N) are required for the same task before progressing downstream.
Export	The batch-level output (CSV/JSON and media files or links) prepared for external delivery.
Ops (Operations)	Internal users who configure projects, manage raters, upload tasks, monitor pipelines, and export results.
Ops Portal	The internal administrative interface used to configure and manage projects, batches, pipelines, uploads, and exports.
Pipeline	The configurable sequence of workflow stages (e.g., L1, L2, Review, Hold, Archive) that tasks progress through.
Pipeline Stage	An individual step within the pipeline defining a task's status and required action.
Project	A container within a workspace that manages configuration such as pipelines, UI schemas, raters, and batch settings.
Queue	The ordered set of tasks eligible for assignment to raters, typically FIFO unless configured otherwise.
Rater	A user who completes tasks by submitting annotations.
Response Schema	The structured definition of fields and components required for annotation; drives task UI configuration.
Reviewer / QA	A user who reviews or approves annotations at higher pipeline levels.
Signed URL	A secure, time-limited link used to access stored media assets (e.g., images, audio, video).
Status	The system-defined state of a task or annotation (e.g., pending, assigned, claimed, review, done).

Task	The basic unit of work containing text/media/assets that require annotation.
Task UI	The interface presented to raters or reviewers to complete tasks, based on the response schema and pipeline stage.
Workspace	The top-level container that groups and isolates projects, data, permissions, and configurations.

8. Requirements Overview (Organized by Phased Acceptance)

The following requirements define the phased delivery expectations for the - Data Annotation Platform. Each phase builds upon the previous, with functional, technical, and operational capabilities expanding toward a complete, production-ready annotation environment. Full acceptance criteria and details can be found in Appendix A.

8.1 Phase Timeline

- Data is targeting an accelerated delivery for the initial MVP. The timeline below provides vendors with a high-level sequencing model to support estimation and resource planning. Final milestone dates will be defined in the SOW.

Phase	Duration (Target)	Description
Phase 1 — MVP Foundations	**4-6 Weeks**	**Goal:** Launch a production-ready internal labeling platform that replaces manual task distribution and eliminates vendor dependency for simple annotation projects. **Requirements:** Deliver a usable internal labeling platform with the foundational scaffolding (e.g., end-to-end task creation, serving, annotation, and basic pipeline movement). All P0 requirements across modules
Phase 2 — Operational Expansion	**4-6 weeks of Phase 1 Completion**	**Goal:** Scale to enterprise-grade operations supporting 10+ concurrent client projects with quality controls, workforce management, and compliance-ready audit trails within 4-6 weeks of Phase 1 completion. **Requirements:** Adds workflows required for scaled operations (e.g., consensus, benchmarks, Ops tooling, export

		capabilities, time tracking, full auditability, and multi-project parallelization). P1 requirements across modules.
Phase 3 — Optional Enhancements	**To be discussed (SOW Stage)**	**Goal:** Optimize operational efficiency and enable advanced AI training workflows through intelligent automation, real-time monitoring, and flexible task orchestration (e.g., autosave, advanced assignment strategies, instructional content, preprocessing, predictions, and custom logic). **Description:** SOW-dependent features aligned to workflow acceleration and rater experience improvements. Possible features are outlined in the P2/P3 requirements across modules.

Timeline Notes

- Phases may run **partially in parallel** depending on vendor approach.
- Phase boundaries represent **functional acceptance milestones**, not deployment cycles.
- Optional Phase 3 capabilities will be contracted based on SOW prioritization.
- Earlier delivery of key capabilities is encouraged if it reduces overall risk or complexity.

8.2 Phase 1 — MVP (Foundational Platform Capabilities)

Phase 1 delivers the essential scaffolding required for a functioning internal annotation platform. The goal of this phase is to enable end-to-end task creation, serving, annotation, and basic pipeline movement.

Core Platform Architecture

- Workspace → Project → Batch → Task → Annotation hierarchy
- CRUD APIs for all core objects
- Support for response schemas (including versioning considerations)

Authentication & Access

- Login and profile management for internal and external users
- Separation of internal vs. external user experiences
- Baseline RBAC enforcement (role-based permissions across Ops, Raters, Reviewers, Admins)

Pipeline & Status System

- Multi-layer, configurable pipeline structure (L1, L2, Review, Hold, Archive)
- Core status definitions aligned to pipeline stages
- Ops controls for advancing, holding, or archiving tasks

Queueing & Assignment

- FIFO queueing based on upload time
- Manual and automatic assignment
- Claiming logic with concurrency protection and deterministic locking
- Baseline reclaim/timeout behavior

Task UI (MVP)

- Rendering of supported media types (text, image, audio, video, HTML/markdown)
- Core response components (single-select, multi-select, free-text, etc.)
- UI differences per pipeline level (e.g., reviewer UI)

Storage & Infrastructure

- Structured S3 storage for uploads and artifacts
- Signed URL pattern for file access
- Adoption of - Data-standard stack (Node/Express, React/TS, Postgres, MongoDB, GitHub Actions)

Logging & Observability (Baseline)

- Logging of key platform activities (logins, task claims, submissions, status transitions)

Phase 1 establishes the minimal viable labeling platform needed for functional task processing.

8.3 Phase 2 — Workflow Expansion & Operationalization

Phase 2 builds on the MVP to support core - Data operational workflows at scale. These capabilities are required for real production usage across multiple teams and customers.

Advanced Assignment & Quality Mechanisms

- Rater levels and mass assignment
- Consensus workflows (N annotations per task)
- Benchmarking (as gate or interleaved)

Ops Tooling

- Task upload tooling (bulk upload with validation and progress visibility)
- Task export (CSV/JSON + file packaging or signed URLs)
- Reset/archive/ignore controls for Ops
- Tagging system for tasks and raters (bulk add/remove)

Quality Assurance Capabilities

- Linter framework (configurable rules, blocking/warning modes)
- Time tracking for payout (basic active-time measurement)
- Annotation metadata for operational insights (timestamps, durations)

Logging & Auditability (Full)

- Searchable audits for all key events
- Retention policy and alignment with Datadog logging conventions

Parallelization & Multi-Project Support

- Support for concurrent active projects and batches
- Ensured isolation of data, permissions, and configurations

Phase 2 brings the platform to the level required for - Data's current operational workflows and customer commitments.

8.4 Phase 3 — Optional Enhancements (SOW-Dependent)

Phase 3 includes optional features that provide workflow acceleration, improved rater experience, or operational efficiency but are not required for the MVP or baseline production workflows. These will be included depending on SOW negotiation.

Rater Experience & UI Enhancements

- Drafts/autosave for in-progress work
- Instructional pages embedded within the task UI
- Notification center for rater/reviewer comments (read-only)

Advanced Assignment & Workflows

- Additional assignment strategies (random, LIFO, weighted, etc.)
- Batch prioritization within a project
- Min/max task limits per batch/project (optional date constraints)

Preprocessing, Predictions, and Custom Logic

- Pre/post-processing pipelines for task upload and delivery
- Prelabels/predictions visible or editable in UI
- Custom grader logic with secure execution sandboxing

Phase 3 capabilities may be proposed independently or prioritized based on - Data's roadmap and the vendor's technical approach.

8.5 Requirement Mapping by Phase

Requirement ID	Module	Short Description	Phase
QUEUE-01	Queueing	FIFO queueing + claim locking	**Phase 1**
QUEUE-02	Queueing	Manual & automatic assignment	**Phase 1**
UI-01	Task UI	File/media rendering (text, image, audio, video, HTML/markdown)	**Phase 1**
UI-02	Task UI	Core response components (single/multi-select, free text)	**Phase 1**
UI-03	Task UI	Dynamic/multi-turn support	**Phase 1**
UI-04	Task UI	Different UI per pipeline level (review vs L1)	**Phase 1**

RATER-01	Rater Portal	Rater task interface (queue, submit, status)	**Phase 1**
INT-01	Infra/Modularity	Modular architecture (extensible stages/UI elements)	**Phase 1**
INT-02	Infra/Storage	Structured object storage (S3)	**Phase 1**
DATA-01	Data Model	Core object hierarchy (Workspace → Project → Batch → Task → Annotation)	**Phase 1**
DATA-02	Data Model	CRUD APIs for core objects	**Phase 1**
DATA-03	Data Model	Flexible response schemas; UI differs by pipeline level	**Phase 1**
PIPE-01	Pipeline	Configurable multi-layer pipeline (L1, L2, Review, Hold, Archive)	**Phase 1**
PIPE-02	Pipeline	Annotation statuses aligned to pipeline	**Phase 1**
PIPE-03	Pipeline	Ops controls for stage movement (advance/hold/archive)	**Phase 1**
AUTH-01	Authentication	Login + profile management	**Phase 1**
AUTH-03	Authentication	Internal vs external experience separation	**Phase 1**
CUST-01	Customer Portal	Customer Dashboard: subscription, seat usage, API key visibility	**Phase 2**
PIPE-04	Pipeline	Consensus workflows (N annotations per task)	**Phase 2**
PIPE-05	Pipeline	Benchmarks (gate or interleaved)	**Phase 2**
QUEUE-03	Queueing	Mass assignment by rater level	**Phase 2**
QUEUE-04	Queueing	Project-level assignment	**Phase 2**
UI-05	Task UI	Ops UI builder (GUI config of schemas)	**Phase 2**
OPS-01	Ops Tools	Task/batch upload tool (validation + progress)	**Phase 2**
OPS-02	Ops Tools	Batch export (CSV/JSON + packaging/signed URLs)	**Phase 2**
OPS-03	Ops Tools	Reset/archive/ignore controls	**Phase 2**

OPS-04	Ops Tools	Support Rater/Reviewer impersonation for operational troubleshooting.	**Phase 2**
OPS-06	Ops Tools	Tag management (bulk add/remove; queryable)	**Phase 2**
RATER-02	Rater Mgmt	Rater levels/qualifications mgmt	**Phase 2**
QUAL-03	Quality	Time tracking for payout (active time)	**Phase 2**
INT-03	Infra/Parallelization	Concurrent projects/batches (isolation)	**Phase 2**
LOG-01	Logging	Platform activity logging & audit	**Phase 2**
AUTH-02	Authentication	Okta SSO (OAuth2)	**Phase 2**
QUAL-01	Quality	Linter framework (blocking/warning; stored results)	**Phase 2**
CUST-02	Customer Portal	Customer Dashboard: usage and annotation volume metrics	**Phase 3**
CUST-03	Customer Portal	Customer upload and download of datasets for Ops-managed projects	**Phase 3**
QUEUE-05	Queueing	Advanced strategies (random/LIFO/weighted)	**Phase 3**
UI-06	Task UI	Drafts/autosave	**Phase 3**
OPS-05	Ops Tools	Ops dashboard (status counts + drill-down)	**Phase 3**
OPS-07	Ops Tools	Advanced task filtering (custom filters)	**Phase 3**
QUAL-02	Quality	Analytics/metadata (completion/throughput time)	**Phase 3**
PROC-01	Processing	Pre/post-processing pipelines	**Phase 3**
PROC-02	Processing	Prelabels/predictions	**Phase 3**
MISC-01	Misc	Min/max task limits	**Phase 3**
MISC-02	Misc	Batch prioritization	**Phase 3**
MISC-03	Misc	Instruction page in Task UI	**Phase 3**
MISC-04	Misc	Notification center (read-only comments)	**Phase 3**
MISC-05	Misc	Custom grader logic (secure sandbox)	**Phase 3**
RATER-03	Rater Mgmt	-Works routing/linkage	**Phase 3**

9. Definition of Done (Phase- Based Acceptance)

The Definition of Done outlines the minimum criteria the vendor must meet for - Data to accept **each phase** of delivery. Each phase is evaluated independently. Acceptance for a phase is based on:

- Demonstration of the phase's functional capabilities in the staging environment
- Completion and approval of phase-required documentation
- Passing test coverage and performance validations
- Completion of the phase-specific deliverable checklist

Full acceptance details for each requirement are provided in Appendix A.

9.1 Phase- Specific Functional Demonstration in Staging

For a phase to be accepted, the vendor must demonstrate that **all requirements assigned to that phase** (per PRD Section 8) operate end-to-end in - Data's staging environment.

Phase 1 – MVP Definition of Done

Phase 1 is considered complete when the vendor demonstrates a **usable internal labeling platform** that replaces the core functionality of - Data's current internal annotation and task distribution workflow solution, including:

- End-to-end task flow: workspace/project/batch creation → task upload → serving → annotation → review → export
- Pipeline and queue behaviors functioning according to specification (claim logic, locking, assignment rules, status transitions)
- Core MVP UI and role-based experiences (Ops, Rater, Reviewer, etc.)
- Storage, logging, and baseline observability in place
- All P0 requirements delivered and functioning without manual intervention

Phase 2 – Definition of Done

Phase 2 is considered complete when the vendor demonstrates **all Phase 2 (P1)** requirements, which represent highly requested operational and scaled-workflow capabilities including:

- Operational workflows needed for scaled, multi-project operations
- Consensus, benchmarks, mass assignment, tagging, auditability, time tracking
- Enhanced Ops tooling (bulk upload, export, reset/archive/ignore)
- Isolation of concurrent customer projects and data

- All P1 functions operating end-to-end in staging

Phase 3 – Definition of Done (Optional Enhancements)

Phase 3 is considered complete when the vendor demonstrates **only the optional features contracted via SOW**, including any P2/P3 enhancements such as:

- Autosave
- Advanced assignment logic
- Instructional content
- Pre/post-processing
- Predictions
- Custom grader logic
- Additional operational or UI enhancements

Phase 3 acceptance applies strictly to the subset of optional capabilities selected by - Data.

9.2 Minimum Documentation Delivered

Documentation must be delivered **for the capabilities completed in that phase**, including:

- API documentation (OpenAPI/Swagger)
- Architecture summaries for delivered components
- Developer setup instructions
- Ops and Rater guides relevant to delivered workflows
- CI/CD and environment configuration notes
- If a capability is delivered, documentation for that capability must be delivered

9.3 Test Coverage Requirements

For each phase, the vendor must provide test coverage appropriate to the capabilities delivered in that phase:

- Automated unit tests
- Integration tests for cross-component behaviors
- Workflow tests for queueing, assignment, claim concurrency, pipeline transitions, export
- All automated tests must pass in - Data's CI/CD pipelines

9.4 Performance Validations

For each phase, all capabilities delivered must meet the performance thresholds defined in PRD Section 3:

- UI load times
- Queue assignment latency
- Deterministic claim concurrency
- Export performance
- API response-time targets

Performance validation may be demonstrated through automated tests, logs, or live walkthrough.

9.5 Deliverable Checklist Completed

Each phase must satisfy the phase-specific deliverable checklist, including:

- Functionality deployed to staging
- All in-scope API endpoints implemented
- Required Ops and Rater/Reviewer UI functioning
- Logging and audit trails implemented for that phase
- Data model, schema handling, and pipelines implemented for that phase
- CI/CD integration functional
- Documentation delivered and approved
- Code handed off in GitHub with PR-based history

9.6 Verification & Acceptance

A phase is accepted when:

1. All required functionality for that phase is successfully demonstrated in staging
2. Documentation for that phase is approved
3. Test coverage and performance validations for that phase pass
4. All deliverables for that phase are approved

Any deviations must be managed through change control.

10. Risks

The following risks may affect delivery and estimation:

- Requirements may evolve during discovery.
- Changes to scale assumptions (tasks/day, concurrency, file size) may impact architecture or cost.
- Timely access to - Data SMEs, infrastructure, and accounts may influence delivery sequencing.
- Availability of rater workforce and benchmark tasks may affect validation timelines.
- Changes to security/legal compliance requirements may introduce additional effort.
- All risks will be reviewed during discovery and addressed via the change control process.

11 Change Control

Any change to scope, priorities, or acceptance criteria must be documented as a written change request and approved by both parties as an amendment to the SOW and/or this PRD addendum.

12. Sign-off

By signing below, the parties acknowledge that this PRD is incorporated into the applicable SOW/contract addendum and will be used for delivery and acceptance of the Platform.

- Data / -	**Consultant**
Name: Title: Signature: Date:	Name: Title: Signature: Date:
Approval indicates requirements and acceptance criteria are agreed.	Approval indicates commitment to deliver in accordance with this PRD and SOW.

Appendices

Appendix A: Functional Requirements (Detailed)

Priority Levels

Priorities are defined as follows:

Priority	**Meaning**
P0	Essential features for the product to be a labeling platform.
P1	Must-have features for - Data current workflows to exist.

P2	Important features.
P3	Nice-to-have features; not deal breakers.

Requirements are organized by module. Each requirement includes acceptance criteria intended for contractual verification.

Authentication & Access Control

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
AUTH-01	P0	Implement login and profile management for raters and internal users.	Users can authenticate and access role-appropriate areas; profiles can be created/updated; access to internal pages/tools can be restricted to internal users.	SSO integration required in AUTH-02.
AUTH-02	P1	Integrate Okta Single Sign-On (SSO) using OAuth2 for authentication.	Okta OAuth2 login works end-to-end in dev/staging/prod; users are provisioned and assigned roles/permissions; session management and logout behave consistently.	Referenced in technical requirements; also aligns with -Works SSO needs.
AUTH-03	P0	Support separation of internal vs. external (rater) platform experiences.	Platform provides a controlled mechanism to expose or hide pages/tools/fields based on user role; internal-only tools are not accessible by raters.	Must be enforceable server-side, not only via UI.

Core Data Model & APIs

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
DATA-01	P0	Implement core object hierarchy: Workspace > Project > Batch > Task > Annotation(s).	API supports create/read/update/delete (CRUD) of hierarchy objects and maintains referential integrity; tasks belong to a batch; batches belong to a project; projects belong to a workspace.	API all objects expanded in DATA-02.
DATA-02	P1	Provide APIs to create, update, delete, and fetch all core objects (task/project/user/workspace/batch/etc.).	Documented REST APIs exist for CRUD operations; OpenAPI/Swagger docs are published; basic authorization enforced per role.	Ensure consistency with stack guidelines.
DATA-03	P0	Support flexible annotation response schemas and complex responses, including differing UIs across pipeline levels.	A project/batch can define a response schema; annotations store responses that validate against schema; review layers can use different UI/schema than L1; schema changes are handled per defined behavior (e.g., apply to pending tasks).	Needs doc notes schema may be NoSQL-like while translating to SQL.

Pipeline & Status Management

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
PIPE-01	P0	Support a multi-layer, configurable pipeline per project or batch (e.g., L1, L2...Ln, R1...Rn, Hold, Archive).	Ops can define pipeline stages and ordering; tasks/annotations transition through stages; pipeline definition is stored	Must interoperate with queue/assignment and task statuses.

			and versioned per batch or project.	
PIPE-02	P0	Implement annotation statuses aligned to pipeline (e.g., pending, assigned, claimed, pending-review, done, hold, completed, delivered, archive).	Statuses are persisted, queryable, and enforced; transitions follow allowed rules; status changes are logged.	Status list may be extended per project.
PIPE-03	P0	Provide Ops controls to manually trigger movement of annotations/tasks through pipeline stages.	Authorized Ops users can advance, send to hold, or archive/ignore tasks; the resulting status is visible in Ops views and APIs.	Reset tasks capability expanded in PIPE-06.
PIPE-04	P1	Support consensus workflows: configure the number of annotations per task for consensus pipelines.	Ops can configure N annotations per task; the queue serves tasks until N annotations are collected; downstream review/aggregation stage receives the set.	Aggregation logic may be minimal in Phase 1; storage and serving must work.
PIPE-05	P1	Support benchmarks (BMs) as gate at project start and/or embedded in-queue.	Ops can configure benchmark tasks and whether they are served as a wall or interleaved; benchmarks can be enabled/disabled and served at a configured percentage.	Benchmark scoring rules may be simple initially.

Queue, Assignment, & Claiming

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
QUEUE-01	P0	Implement queueing and assignment/claim	Eligible raters receive tasks in FIFO order by default;	Requires careful 'task is claimed' logic.

		logic with FIFO based on task upload time.	tasks cannot be simultaneously claimed by multiple users; reclaim/timeout behavior is defined and tested.	
QUEUE-02	P0	Support manual and programmatic task assignments based on pipeline settings.	Ops can assign tasks to specific raters; system can auto-assign based on configuration; assignments are visible via API and UI.	Mass assignments and project-level assignments in QUEUE-03/04.
QUEUE-03	P1	Support mass-assignments based on rater 'level' for a project.	Ops can define rater levels per project and mass-assign tasks to raters meeting the level criteria; changes take effect without code changes.	Interacts with RATER-02.
QUEUE-04	P1	Support project-level assignments in addition to batch-level assignments.	Ops can assign raters to a project, with task serving limited to assigned raters for applicable batches; can override at batch level.	
QUEUE-05	P2	Support advanced assignment strategies (random, FIFO, LIFO, etc.).	Ops can select assignment strategy per batch/project; system serves tasks accordingly; strategy selection is auditable.	

Tasking UI

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
UI-01	P0	Support task UI elements including file upload (audio/image/video)	Tasks can include one or more files; uploaded files are stored and	Storage in S3; signed URLs recommended for delivery.

		and rendering of supported data types (audio, text, video, markdown, HTML).	retrievable; UI renders specified data types reliably; max file sizes and supported formats are documented.	
UI-02	P0	Support multiple response types (single-select, multi-select, free text) and multimodal UIs.	UI supports configuring response components; responses persist and validate; multiple components per task are supported.	
UI-03	P0	UI is dynamic based on upload and supports multi-turn tasks where needed.	Task UI can adapt to task metadata/schema; multi-turn interactions can be represented and stored without data loss.	
UI-04	P0	Support different UI configurations between pipeline levels (e.g., review UI differs from L1).	For the same task, the UI presented to a reviewer can differ from the annotator UI; level-based UI selection is configurable by Ops.	
UI-05	P1	Provide Ops-managed UI configuration tooling (GUI UI builder) to configure task UI elements.	Ops users can configure UI elements and schemas through a GUI; changes are versioned; behavior for applying changes to existing tasks is defined.	Need explicit policy for post-launch changes (all tasks vs pending tasks).

UI-06	P3	Support drafts/auto-save for in-progress work.	In-progress responses are auto-saved at a defined interval; users can resume after refresh; draft data is isolated per user and task.	

Ops Portal & Tools

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
OPS-01	P1	Provide an Ops tool to upload tasks into a batch.	Ops can upload via UI (and/or API) and validate inputs; upload progress and errors are visible; upload time captured for FIFO ordering.	May support bulk upload via file drop.
OPS-02	P1	Provide batch export of results as CSV and/or JSON.	Ops can export completed results for a batch; export includes required identifiers; if tasks include file uploads, export includes a mechanism to package links/files for customer delivery (e.g., signed S3 links).	
OPS-03	P1	Support reset tasks / archive / ignore status controls for Ops.	Ops can move annotations to archive/ignore; task becomes unavailable to raters; audit trail records action and actor.	
OPS-04	P1	Support Rater/Reviewer impersonation for operational troubleshooting.	Authorized Product Admins can impersonate a rater or reviewer to reproduce issues and view the platform exactly as that user would. Impersonation sessions must be explicitly logged to	Requires secure session switching, elevated-permission handling, and integration with audit logging. Must maintain RBAC boundaries and prevent unauthorized access to internal tools.

			indicate the impersonator's user account, time-bounded, and visible in audit trails. The impersonated user must not receive notifications or be impacted by the session.	
OPS-05	P2	Provide Ops visibility dashboard of project status (tasks per status; project/batch/task level visibility).	Dashboard displays counts by status and supports drill-down to batch/task lists; data is accurate and refreshes within agreed interval.	May be replaced by BI if needed, but platform must expose data.
OPS-06	P2	Support tag management: bulk assign/remove tags for raters and tasks.	Ops can bulk apply tags; tags are queryable; tag edits are logged; tags can be used by queue assignment rules.	Enables multi-domain expertise serving.
OPS-07	P3	Support filtering tasks in Ops tools by custom filters (# annotations, skipped tasks, rejected reviews, input data, labels, etc.).	Ops can apply filters to task lists; filtering works within reasonable performance bounds for expected dataset sizes.	

Rater Management & Portal

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
RATER-01	P0	Provide a rater portal to access assigned/claimable tasks and complete work.	Raters can see their queue/assignments, open tasks, submit responses, and view completion status; portal enforces permissions and shows only eligible tasks.	
RATER-02	P1	Provide Ops tooling to move	Ops can define rater	

		raters between levels on a project and/or across projects.	levels/qualifications; changes affect task serving; historical level changes are logged.	
RATER-03	P3	Link projects to -Works opportunities or provide URL to route raters to the appropriate project board/queue.	A project has a field for -Works opportunity linkage or a generated URL; raters can navigate via link to correct project context.	Depends on -Works integration approach.

Quality & Benchmarking

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
QUAL-01	P1	Implement configurable linters (programmatic quality checks) managed by Ops and built by engineering.	A linter template exists; Ops can configure which linters apply to a project; linters can set blocking vs warning; results are stored linked to user and annotation; optional display to rater is supported.	Linters may manipulate task/annotation/response data per configuration.
QUAL-02	P3	Provide annotation analytics/metadata such as completion time and throughput time.	Platform records timestamps needed to compute metrics; exposes data via API and/or Ops views; definitions are documented.	Time spent tracking in QUAL-03.
QUAL-03	P1	Track basic time spent per task for payout support.	Platform records active time in task UI	

			(with reasonable safeguards for idle time); data is queryable by Ops; methodology documented.	

Storage, Modularity, & Parallelization

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
INT-01	P0	Design modular platform architecture to easily add/remove pipeline stages and UI element types with minimal development work.	Codebase provides clear modules/interfaces for pipeline stage definitions and UI components; adding a new UI element type requires limited, localized changes; documented extension points exist.	Architectural diagrams required as deliverable.
INT-02	P0	Implement structured object storage (e.g., S3) for customer data, file uploads, delivery data, and other artifacts.	Storage structure is documented; access is controlled; files are retrievable by authorized users; delivery packaging approach is documented.	
INT-03	P1	Support batch/project parallelization so multiple projects/batches can run concurrently.	Platform can host concurrent active projects/batches without cross-contamination of data or configuration; performance meets defined SLOs.	

Logging & Auditability

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
LOG-01	P1	Implement platform activity logs.	Platform records key audit events (logins, task claims, submissions, status transitions, exports, admin actions); logs are searchable by Ops/Admins; retention policy supports ~15-day default retention unless otherwise configured.	Align with Datadog logging/monitoring; least privilege access.

Processing & Predictions (Optional)

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
PROC-01	P2	Support pre-processing on task upload and post-processing on delivery to match customer formats.	Platform provides interfaces to run pre/post processing transformations; Ops can select processing configuration per project/batch; processing results are logged and errors surfaced.	May be implemented via AWS Lambda or equivalent.
PROC-02	P2	Support preannotations/prelabels/predictions.	Tasks may include prefilled labels/predictions; UI can display	

			and/or allow editing; exports include original and final labels where configured.	

Customer Portal

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
CUST-01	P1	Provide a Customer Dashboard summarizing subscription, seat usage, and API credentials.	Customer Admins can view subscription tier, available vs. used seats, active users, and their API keys. Dashboard must enforce strict RBAC and must not expose any internal Ops tooling or project configuration.	Integrates with existing authentication and access controls (AUTH-01, AUTH-03).
CUST-02	P2	Support batch prioritization within a project.	Customer Admins can view aggregated counts of completed annotations, batches, and exported datasets for their organization. Metrics remain high-level and must not reveal rater activity, task-level details, or other internal operational data.	Depends on export metadata (OPS-02) and logging/audit infrastructure (LOG-01).
CUST-03	P2	Enable Customer Admins to upload input datasets and export completed datasets.	Customer Admins can upload supported dataset formats for Ops-managed projects and download final labeled datasets (CSV/JSON and asset links). Customers must not gain access to internal project configuration, task workflows, or Ops tooling.	Uses existing upload/export endpoints (OPS-01, OPS-02). RBAC separation must be maintained.

Other Enhancements (Optional)

ID	Priority	Requirement	Acceptance Criteria	Notes/Dependencies
MISC-01	P3	Allow min/max tasks that can be worked on per batch/project, configurable (optionally by date range).	Ops can set limits; platform enforces limits in assignment/claim; configuration can be updated without code changes.	
MISC-02	P3	Support batch prioritization within a project.	Ops can prioritize batches; queue serves higher priority batches first when multiple are eligible.	
MISC-03	P3	Provide instruction page embedded in task UI.	Task UI can display instructions content (text/markdown/HTML) associated with the project/batch; updates follow defined versioning policy.	
MISC-04	P3	Provide notification center for annotators/reviewers to see comments on tasks in read-only mode.	Users can view comments/notes attached to tasks; notifications indicate new comments; users cannot edit immutable comments.	
MISC-05	P3	Support custom grader logic: run custom code and save results per task.	A configurable grading hook exists; grading results are stored with task/annotation; security controls prevent unsafe execution.	Requires careful sandboxing and approvals.