ZZedc: A Comprehensive Electronic Data Capture System

System Architecture, Core Features, and Regulatory Compliance Framework

ZZedc Development Team

2025-12-22

Executive Summary

ZZedc is an open-source Electronic Data Capture (EDC) system designed for clinical research across all therapeutic areas and study designs. Built on the R/Shiny framework, the system provides a modern, web-based platform for data collection, management, and regulatory compliance that addresses the needs of pharmaceutical trials, academic research institutions, and independent investigators.

This document serves two purposes. First, it provides comprehensive technical documentation of ZZedc's architecture, features, and implementation. Second, it offers a reference framework for what a modern, open-source, regulatory- compliant EDC system should include, using ZZedc as an exemplar implementation.

The system implements 49 distinct functional modules organized across six domains: core data capture, regulatory compliance, data management, system administration, integration services, and quality assurance. Over 3,000 automated tests validate system functionality, and comprehensive documentation supports both users and developers.

Key distinguishing characteristics of the ZZedc system include:

- **Open Architecture**: The entire codebase is available for inspection, modification, and extension, enabling organizations to verify compliance implementations and customize functionality for specific requirements.
- **Regulatory Framework**: Native implementation of GDPR, GCP, and FDA 21 CFR Part 11 requirements, with 32 dedicated compliance features addressing data protection, clinical trial integrity, and electronic records management.
- **Modern Technology Stack**: Built on R 4.4+ with Bootstrap 5 responsive design, providing a contemporary user experience while leveraging the statistical computing capabilities of the R ecosystem.
- **Deployment Flexibility**: Support for local installation, server deployment, containerized environments, and cloud platforms, enabling organizations to choose infrastructure appropriate to their security and scalability requirements.
- **Cost Efficiency**: As open-source software, ZZedc eliminates per-site licensing fees that characterize commercial EDC systems, enabling broader access to regulatory-compliant data capture for resource-constrained research programs.

The document is organized into seven parts. Part I describes the system architecture and technology foundation. Part II covers core EDC functionality including data capture, validation, and reporting. Part III addresses data management and export capabilities. Part IV details GDPR compliance features. Part V covers Good Clinical Practice (GCP) implementation. Part VI describes FDA 21 CFR Part 11 compliance. Part VII addresses integration, deployment, and quality assurance.

Part I: System Architecture

This section describes the technical architecture underlying ZZedc, including the technology stack, database design, modular structure, and security framework. Understanding these architectural foundations provides context for the feature implementations described in subsequent sections.

1. Technology Stack

Design Philosophy

ZZedc employs a technology stack optimized for clinical research computing. The selection criteria emphasized statistical computing integration, open-source licensing, cross-platform compatibility, and long-term maintainability. The resulting stack leverages mature, well-supported technologies with extensive community resources.

Application Framework

The system is built on Shiny, the web application framework for R developed by Posit (formerly RStudio). Shiny enables reactive programming paradigms where user interface elements automatically update in response to data changes, eliminating the complexity of manual state management that characterizes traditional web development.

The user interface employs bslib, which provides Bootstrap 5 components adapted for Shiny applications. Bootstrap 5 represents the current industry standard for responsive web design, ensuring that the interface functions appropriately across desktop computers, tablets, and mobile devices. The bsicons package provides a consistent icon library that enhances visual communication without requiring external dependencies.

```r
# Core application dependencies
library(shiny)      # Web application framework
library(bslib)      # Bootstrap 5 UI components
library(bsicons)    # Icon library
library(shinyjs)    # JavaScript integration
library(shinyalert) # User notifications
```

Database Layer

ZZedc uses SQLite as its primary database engine, with optional SQLCipher encryption for deployments requiring data-at-rest protection. SQLite provides a serverless, self-contained database that requires no separate installation or configuration, simplifying deployment while maintaining ACID (Atomicity, Consistency, Isolation, Durability) compliance.

The RSQLite package provides the R interface to SQLite, implementing the DBI (Database Interface) specification for consistent database operations. The pool package manages database connections, providing connection pooling that improves performance under concurrent user load while preventing connection exhaustion.

```r
# Database connectivity
library(RSQLite)    # SQLite database driver
library(DBI)        # Database interface abstraction
library(pool)       # Connection pooling
```

For deployments requiring encryption, SQLCipher extends SQLite with transparent AES-256 encryption. When configured, all data written to disk is encrypted, and decryption occurs transparently when the correct key is provided.

Data Processing

The system leverages the tidyverse ecosystem for data manipulation, providing consistent, readable syntax for data transformation operations. The native R pipe operator (|>) is used throughout for operation chaining, reflecting current R coding standards.

```r
# Data processing packages
library(dplyr)      # Data manipulation
library(tidyr)      # Data reshaping
library(stringr)    # String processing
library(lubridate)  # Date/time handling
```

Visualization and Reporting

Interactive data visualization employs ggplot2 for statistical graphics with plotly providing interactive features such as tooltips, zooming, and panning. The DT package wraps the DataTables JavaScript library, providing feature-rich data tables with sorting, filtering, and pagination.

```r
# Visualization packages
library(ggplot2)    # Statistical graphics
library(plotly)     # Interactive visualization
library(DT)         # Interactive data tables
```

Security and Cryptography

Security functions employ the digest package for cryptographic hashing and the openssl package for encryption operations. Password storage uses bcrypt- style hashing with configurable salt values, protecting credentials against both brute-force attacks and rainbow table lookups.

```r
# Security packages
library(digest)     # Cryptographic hashing
library(openssl)    # Encryption operations
```

2. Database Architecture

Schema Design Principles

The database schema follows normalization principles appropriate for transactional data, balancing query performance against data integrity requirements. The design emphasizes flexibility to accommodate diverse study types without schema modification, enabling deployment across therapeutic areas and research methodologies.

Core Tables

The database includes the following primary table categories:

Study Configuration Tables store metadata defining study structure, including protocol information, form definitions, field specifications, and validation rules. These tables enable configuration-driven behavior where study-specific requirements are defined through data rather than code modification.

Subject Data Tables contain participant information including demographics, enrollment status, and study-specific identifiers. The schema supports both screening and enrolled subjects, enabling tracking of the enrollment funnel.

Clinical Data Tables store form responses organized by subject, visit, and form. The flexible schema accommodates forms with varying field counts and data types, supporting the diversity of case report forms across studies.

Audit Tables maintain immutable records of all data modifications, user activities, and system events. The audit schema implements hash chaining for tamper detection, supporting regulatory compliance requirements.

System Tables manage users, roles, permissions, and configuration settings. These tables support multi-site deployments with site-specific access controls.

Referential Integrity

Foreign key constraints enforce referential integrity between related tables, preventing orphaned records and ensuring data consistency. Cascade rules are configured conservatively, typically preventing deletion of referenced records rather than cascading deletions, protecting against inadvertent data loss.

Indexing Strategy

Indexes are defined for frequently queried columns including subject identifiers, visit dates, form types, and user identifiers. The indexing strategy balances query performance against insert performance, recognizing that clinical data entry involves frequent insertions that would be slowed by excessive indexing.

3. Modular Architecture

Module Organization

ZZedc implements a modular architecture where each major feature is encapsulated in a self-contained module. Each module consists of a UI function defining the user interface elements and a server function implementing the business logic. This encapsulation enables independent development, testing, and maintenance of features.

```r
# Module structure example
example_module_ui <- function(id) {
  ns <- NS(id)
  tagList(
    # UI elements with namespaced IDs
  )
}

example_module_server <- function(id, shared_data) {
  moduleServer(id, function(input, output, session) {
    # Server logic
  })
}
```

Module Categories

The system organizes modules into functional categories:

Core Modules implement fundamental EDC operations including data entry (edc_module), data viewing (data_module), and navigation (home_module). These modules are loaded for all deployments.

Authentication Modules manage user identity and access control (auth_module, user_management_module). These modules implement session management, password policies, and role assignment.

Compliance Modules provide regulatory-specific functionality including GDPR data subject rights (privacy_module), FDA electronic signatures (cfr_compliance_module), and audit logging (audit_log_viewer_module).

Administration Modules support system management including backup and restore (backup_restore_module), configuration (setup_wizard_module), and quality monitoring (quality_dashboard_module).

Import Modules enable external data integration including instrument import (instrument_import_module) and form configuration from external sources.

Inter-Module Communication

Modules communicate through reactive values shared at the application level. The primary shared state includes the authenticated user information, current subject selection, and active study context. This shared state enables coordination between modules while maintaining encapsulation.

```r
# Shared reactive state
app_state <- reactiveValues(
  user = NULL,
  current_subject = NULL,
  current_visit = NULL,
  study_context = NULL
)
```

Module Loading

Modules load dynamically based on configuration and user permissions. The module loading system checks user role permissions before loading administrative modules, ensuring that unauthorized users cannot access restricted functionality even by manipulating URL parameters or form submissions.

4. Security Architecture

Authentication System

The authentication system implements secure credential verification with protection against common attack vectors. Upon login attempt, the system retrieves the stored password hash for the specified username, computes the hash of the provided password using the same algorithm and salt, and compares the results. This approach ensures that plaintext passwords are never stored or logged.

Password hashing employs PBKDF2 (Password-Based Key Derivation Function 2) with configurable iteration count, defaulting to 256,000 iterations. This computational cost deliberately slows authentication attempts, rendering brute-force attacks impractical while remaining imperceptible to legitimate users.

Session Management

Sessions use cryptographically random tokens generated at login and stored server-side. Session tokens are transmitted via secure cookies with appropriate flags (HttpOnly, Secure, SameSite) to prevent theft through cross-site scripting or request forgery attacks. Sessions expire after configurable inactivity periods, defaulting to 30 minutes.

Role-Based Access Control

The permission system implements role-based access control (RBAC) where users are assigned to roles, and roles are granted permissions. Standard roles include Administrator, Principal Investigator, Research Coordinator, Data Manager, and Monitor, each with permissions appropriate to their responsibilities.

Permissions operate at multiple levels: application features, data operations, and record scope. Feature permissions control which modules users can access. Operation permissions control which actions users can perform (create, read, update, delete). Scope permissions restrict which records users can access, enabling site-level or study-level data partitioning.

Input Validation

All user input undergoes validation before processing. Server-side validation prevents malicious input from affecting database operations or application behavior regardless of client-side circumvention attempts. Validation includes type checking, range verification, format validation, and sanitization of special characters.

Audit Logging

Every significant action generates an audit log entry capturing the user, timestamp, action type, affected data, and outcome. Audit logs are written atomically with the associated action, ensuring that no action can occur without corresponding audit documentation. The audit system operates at the database level, capturing events that bypass the application layer.

5. Configuration Management

Environment-Based Configuration

The system uses the config package for environment-aware configuration management. A single configuration file (config.yml) contains sections for development, testing, and production environments, with environment selection controlled by an environment variable. This approach enables consistent configuration management across deployment stages.

```yaml
default:
  database:
    path: "data/study.db"
    pool_size: 5
  security:
    session_timeout: 1800
    password_min_length: 8

production:
  database:
    encryption: true
    key_source: "aws_kms"
  security:
    session_timeout: 900
```

Study Configuration

Study-specific settings including form definitions, validation rules, and workflow configurations are stored in the database rather than configuration files. This database-driven approach enables modification of study parameters through the administrative interface without requiring file access or application restart.

Secret Management

Sensitive configuration values such as encryption keys and API credentials are stored outside the configuration file, typically in environment variables or external secret management services. The configuration system retrieves these values at runtime, preventing accidental exposure through configuration file access.

Part II: Core Electronic Data Capture

This section describes the fundamental data capture capabilities that form the heart of the EDC system. These features represent the essential functionality expected of any clinical data management platform.

6. Form Rendering System

Dynamic Form Generation

The form rendering system generates data entry interfaces dynamically from form specifications stored in the database. This configuration-driven approach eliminates the need for code modifications when adding or modifying forms, enabling study teams to define data collection requirements without developer involvement.

Each form specification includes the form identifier and metadata, the list of fields with their properties, the layout defining field arrangement, and the validation rules governing data quality. The rendering engine interprets these specifications to generate Shiny UI elements with appropriate input controls, labels, help text, and validation feedback.

Field Types

The system supports a comprehensive range of field types addressing clinical data collection requirements:

Text Fields capture free-text responses with configurable length limits, pattern validation, and case handling. Subtypes include single-line text for brief responses and multi-line text areas for extended narratives.

Numeric Fields collect quantitative data with configurable decimal precision, range limits, and unit specification. The system validates numeric input on both entry and submission, providing immediate feedback for invalid values.

Date Fields capture temporal data with configurable format display, range restrictions, and partial date handling for situations where complete dates are unavailable. Date validation includes logical checks such as preventing future dates for historical events.

Selection Fields present categorical choices through dropdown menus, radio buttons, or checkbox groups depending on the selection cardinality. Options can be defined inline or referenced from controlled terminology tables for reuse across forms.

Calculated Fields display derived values computed from other field values. Calculations update automatically as source values change, providing immediate feedback on computed endpoints. The calculation engine supports standard arithmetic, date arithmetic, and conditional logic.

Layout Management

Form layout employs a grid system enabling precise field positioning. Designers specify row and column positions for each field, with fields spanning multiple columns when appropriate for visual grouping. The grid system maintains consistent alignment while accommodating varying field widths.

Sections group related fields under common headers, providing visual organization for complex forms. Sections can be configured for conditional visibility, appearing only when relevant based on prior responses. This conditional display reduces form complexity for subjects not requiring certain sections.

Form Navigation

Multi-page forms present content across sequential pages with navigation controls for moving between pages. Page transitions validate data before proceeding, preventing navigation when required fields are incomplete or values fail validation. A progress indicator shows the current position within the form, helping users understand remaining effort.

7. Data Validation Framework

Validation Architecture

The validation framework implements multi-layer data quality enforcement operating at entry time, submission time, and batch processing. This layered approach catches errors as early as possible while enabling complex validations that require complete data.

Entry-Time Validation provides immediate feedback as users enter data. Format validation confirms that entered values match expected patterns. Type validation ensures numeric fields contain numbers and dates contain valid date values. Range validation checks that values fall within acceptable bounds. This immediate validation prevents accumulation of errors requiring later correction.

Submission-Time Validation performs comprehensive checks when users attempt to save form data. Required field validation confirms that mandatory fields contain values. Cross-field validation enforces relationships between fields, such as requiring end dates to follow start dates. Conditional validation applies rules that depend on other field values.

Batch Validation runs scheduled quality checks across the database, identifying issues that cannot be detected at entry time. Cross-form validation ensures consistency between related forms. Longitudinal validation checks value changes across visits for biological plausibility. Cross-subject validation identifies statistical outliers requiring review.

Plain English Validation Language (DSL)

ZZedc implements a domain-specific language (DSL) that enables clinical research staff to define validation rules using plain English syntax. This approach represents a significant advancement over traditional EDC systems that require programming knowledge for validation rule authoring. The DSL supports the full range of clinical trial validation requirements while remaining accessible to data managers, research coordinators, and clinical operations personnel.

Fundamental Design Principle: The validation language prioritizes readability and clinical accuracy over technical elegance. A rule written as `between 18 and 65` communicates its intent immediately to clinical staff reviewing the validation specification, whereas equivalent R code `input$age >= 18 && input$age <= 65` requires programming literacy.

DSL Syntax Categories **Range Validation** enforces numeric boundaries with intuitive syntax:

```
between 18 and 65
1..100
>= 0
<= 200
```

Value Matching validates categorical and text fields:

```
== 'Female'
!= 'Unknown'
in('Yes', 'No', 'N/A')
not in('Missing', 'Refused')
required
```

Conditional Logic applies rules based on other field values:

```
if sex == 'Female' then pregnant in('Yes', 'No', 'Unknown') endif
if age >= 18 then consent required endif
if pregnant == 'Yes' then gestational_age between 0 and 42 endif
```

Date Validation supports clinical timing requirements:

```
consent_date <= enrollment_date
visit_date within 7 days of scheduled_date
birth_date <= today
follow_up_date >= today + 14 days
```

Cross-Visit Validation enables longitudinal data quality checks:

```
weight within 10% of baseline_weight
weight within 10% of previous_weight
visit_date > previous_visit_date
```

DSL Processing Architecture The validation system processes DSL rules through a multi-stage pipeline:

1. **Tokenization**: The lexer converts rule text into tokens, recognizing keywords (between, and, if, then, within), operators (>=, <=, ==, !=), literals (numbers, quoted strings), and field references.

2. **Parsing**: A recursive descent parser constructs an Abstract Syntax Tree (AST) representing the rule structure. The parser validates syntax and reports errors with context (line, column, expected token).
3. **R Code Generation**: The AST is compiled to executable R closures for real-time validation during data entry. Generated validators achieve sub-5ms execution time.
4. **SQL Code Generation**: For batch validation, the AST compiles to SQL WHERE clauses enabling efficient database-level quality checks across thousands of records.

```r
# Example: DSL to R compilation
dsl_rule <- "between 18 and 65"
ast <- parse_dsl_rule(dsl_rule)
validator <- generate_r_validator(ast)

# The compiled validator can be applied to data
validator(25)  # Returns TRUE
validator(17)  # Returns FALSE with error message
```

Google Sheets Integration for Validation Rules

ZZedc provides seamless integration between the validation DSL and Google Sheets, enabling non-technical staff to author, review, and manage validation rules using familiar spreadsheet tools. This integration represents a novel approach to clinical trial validation management.

Validation Rules Sheet Schema The system expects a Google Sheet with the following structure:

Column	Required	Description
rule_id	Yes	Unique identifier (e.g., AGE_ELIGIBILITY)
field_code	Yes	Target field code (e.g., age)
rule_dsl	Yes	Validation rule in DSL syntax
form_code	No	Specific form (blank = all forms)
error_message	No	Custom error (auto-generated if blank)
severity	No	ERROR, WARNING, or INFO (default: ERROR)
rule_category	No	FIELD, CROSS_FIELD, ELIGIBILITY, SAFETY
requires_approval	No	TRUE if PI approval needed

Example validation rules in Google Sheets format:

rule_id	field_code	rule_dsl	error_message	severity
AGE_RANGE	age	between 18 and 65	Age must be 18-65 for eligibility	ERROR
BP_SYSTOLIC	bp_sys	between 80 and 200	Systolic BP outside valid range	ERROR
WEIGHT_CHANGE	weight	within 10% of previous_weight	Weight change >10%	WARNING
CONSENT_ORDER	consent_date	consent_date <= enrollment_date	Consent must precede enrollment	ERROR

```r

# Import validation rules from Google Sheets
result <- import_validation_rules_from_gsheets(
  sheet_id = "1ABC...xyz",
  sheet_name = "validation_rules",
  imported_by = "data_manager_01",
  validate_syntax = TRUE
)

# Result includes import statistics
# Imported: 47 rules, Skipped: 2 with syntax errors
```

Import and Synchronization The import process validates DSL syntax before storing rules, preventing malformed rules from entering the system. Syntax errors are reported with specific line and column information enabling quick correction in the source spreadsheet.

Template Generation For new studies, the system generates a pre-configured Google Sheet template with example rules and a syntax reference tab:

```
# Create a new validation rules template
create_validation_rules_template(
  sheet_name = "STUDY_2024_001_Validation_Rules",
  include_examples = TRUE
)
```

The template includes:

- Example validation rules covering common patterns
- Syntax reference documentation within the sheet
- Column headers with data validation for severity and category
- Conditional formatting to highlight required fields

Role-Based Validation Rule Management

The validation system integrates with ZZedc's authentication framework to provide granular access control for rule management. This integration ensures that validation rules undergo appropriate review before affecting data entry.

Permission Matrix

Role	View	Create	Edit	Delete	Approve	Activate
Admin	Yes	Yes	Yes	Yes	Yes	Yes
PI	Yes	No	No	No	Yes	Yes
Data Manager	Yes	Yes	Yes	No	No	Yes
Coordinator	Yes	No	No	No	No	No
Monitor	Yes	No	No	No	No	No

View: Access to view all validation rules and their status **Create**: Ability to import or create new validation rules **Edit**: Ability to modify existing rules (creates new version) **Delete**: Ability to remove rules from the system **Approve**: Authority to approve rules requiring PI sign-off **Activate**: Authority to enable/disable rules for production use

Approval Workflow Rules flagged as `requires_approval = TRUE` must complete an approval workflow before activation:

1. **Submission**: Data Manager imports or creates rule marked for approval
2. **Pending Review**: Rule stored but not active; visible in approval queue

3. **PI Review**: Principal Investigator or Admin reviews rule definition
4. **Decision**: Approve (rule activated) or Reject (with comments)
5. **Audit Trail**: All approval actions logged with timestamps and user IDs

```r
# Request approval for a rule
request_dsl_rule_approval(
  rule_id = "ELIGIBILITY_AGE",
  requested_by = "data_manager_01",
  comments = "New age criterion per protocol amendment 3"
)

# PI reviews and approves
review_dsl_rule(
  rule_id = "ELIGIBILITY_AGE",
  reviewer_id = "pi_johnson",
  reviewer_role = "pi",
  decision = "APPROVE",
  comments = "Approved per IRB amendment approval dated 2024-03-15"
)
```

Validation Rule Versioning and Audit

Every modification to validation rules creates a new version with complete audit trail. This versioning supports regulatory requirements for documenting validation changes throughout the study lifecycle.

The audit trail captures:

- Original rule definition and all modifications
- User performing each change with timestamp
- Reason for change (when provided)
- Approval workflow history
- Activation and deactivation events

```r
# Retrieve version history for a rule
history <- get_rule_version_history("ELIGIBILITY_AGE")

# Output shows complete change history:
# Version 1: Created 2024-01-15 by setup_wizard
# Version 2: Modified 2024-03-10 by data_manager_01
#            Change: "between 18 and 65" -> "between 18 and 70"
#            Reason: "Protocol amendment 3"
# Version 3: Approved 2024-03-12 by pi_johnson
```

Batch Quality Control Engine

Beyond real-time validation, the system includes a batch QC engine for scheduled data quality assessments. The QC engine compiles DSL rules to SQL for efficient database-level checking.

```r
# Run nightly batch validation
qc_results <- run_batch_validation(
  rule_categories = c("CROSS_VISIT", "CROSS_FORM"),
  since_date = Sys.Date() - 1
)

# Results identify records failing validation
# Record ID: 1042, Rule: WEIGHT_CHANGE
# Current weight: 95kg, Previous: 75kg
# Change: 26.7% (exceeds 10% threshold)
```

The batch engine supports complex cross-visit and cross-subject validations that cannot be evaluated during single-form data entry, enabling comprehensive longitudinal data quality monitoring.

Query Generation

When validation identifies discrepancies, the system generates data queries requiring site resolution. Each query documents the specific issue, affected record, relevant field values, and the validation rule that triggered the query. Queries enter a workflow that tracks resolution through site response, verification, and closure.

The query management system implements standard EDC query lifecycle: Open (issue identified), Answered (site response provided), Verified (response confirmed acceptable), and Closed (resolution complete). Query metrics track resolution times and identify sites or data patterns requiring attention.

8. Visit and Schedule Management

Protocol Schedule Definition

The scheduling system captures protocol-defined visit sequences with associated timing requirements. Each visit specification includes the visit identifier and display name, the target timing relative to a reference point (typically randomization or enrollment), the acceptable window defining permitted timing variation, the required and optional forms for the visit, and any predecessor requirements constraining visit ordering.

```
# Visit schedule definition
visits <- list(
  list(
    code = "SCREENING",
    name = "Screening Visit",
    day = -14,
    window_before = 7,
    window_after = 0,
    required_forms = c("ELIGIBILITY", "DEMOGRAPHICS", "MEDICAL_HISTORY")
  ),
  list(
    code = "BASELINE",
    name = "Baseline Visit",
    day = 1,
    window_before = 0,
    window_after = 3,
    required_forms = c("VITALS", "LABS", "EFFICACY_ASSESSMENT")
  )
)
```

Visit Tracking

The system tracks visit status for each subject, displaying scheduled dates, completion status, and any timing deviations. A visual schedule shows upcoming and overdue visits, enabling coordinators to manage subject calendars effectively. Color coding distinguishes on-time visits from those approaching or exceeding window limits.

Window Violation Detection

When visit data indicates timing outside protocol windows, the system flags the deviation and initiates documentation workflow. Window violations are categorized by severity (minor deviation within tolerance versus major violation affecting data interpretability) and type (early versus late). Deviation reports aggregate window violations for protocol compliance assessment.

9. Subject Management

Subject Registration

The subject management system handles participant identification and enrollment tracking. Registration captures screening identifiers before enrollment and assigns randomized subject identifiers upon enrollment. The system enforces identifier uniqueness within study and site scope, preventing duplicate registrations.

Enrollment Workflow

The enrollment process implements configurable workflow stages reflecting study requirements:

Screening captures initial subject contact and eligibility assessment. Screening forms collect demographic information and evaluate inclusion and exclusion criteria. The system calculates eligibility based on entered criteria values, flagging subjects who fail any criterion.

Enrollment confirms eligible subjects for study participation. The enrollment action requires completion of informed consent documentation, verification of eligibility, and assignment of randomized identifier. The system timestamps enrollment and initiates the visit schedule.

Randomization assigns treatment allocation for randomized studies. The system can integrate with external randomization systems or implement internal randomization algorithms. Randomization records are protected with additional access controls to maintain blinding.

Completion documents study conclusion for each subject including completion status (completed, withdrawn, lost to follow-up, death) and date. Completion triggers closeout workflows including form completion verification and query resolution requirements.

Status Tracking

The system maintains comprehensive status tracking at subject level, displaying current enrollment status, visit completion status, query status, and protocol deviation status. Status summaries enable study oversight through dashboards showing enrollment progress, retention rates, and data quality metrics.

10. Multi-Site Support

Site Configuration

The system supports deployment across multiple investigator sites with site-specific configuration and access controls. Each site record captures the site identifier, name, and contact information; the site principal investigator; the site activation date and status; and any site-specific configuration parameters such as timezone or form customizations.

Site-Level Permissions

Access controls operate at site level, restricting user data access to their assigned site(s). Site coordinators can view and modify data for their site while sponsors and monitors can access data across sites. Role-based permissions combine with site assignments to determine effective access rights.

Multi-Site Queries

Query management respects site partitioning, routing queries to appropriate site personnel based on the affected subject's site assignment. Query reports can be filtered by site, enabling site-specific performance assessment and targeted quality improvement efforts.

Site Performance Metrics

The system calculates and displays site-level performance metrics including enrollment rate relative to target, data entry timeliness, query resolution time, and protocol deviation frequency. These metrics support site selection decisions for future studies and identify sites requiring additional support or monitoring.

11. Reporting System

Three-Tier Architecture

The reporting system implements a three-tier architecture addressing different reporting needs:

Tier 1: Basic Reports provide operational summaries for day-to-day study management. These reports include enrollment status summaries, data entry completion by form and site, and visit completion tracking. Tier 1 reports are available to all authenticated users and update in near real-time.

Tier 2: Quality Reports support data management oversight with metrics on data quality and query status. Reports include missing data summaries, query status and aging, validation failure patterns, and data correction activity. Tier 2 reports are typically restricted to data management and quality assurance personnel.

Tier 3: Statistical Reports provide analytical summaries for scientific review. Reports include demographic summaries with baseline characteristics, endpoint distributions and trends, and safety data summaries. Tier 3 reports require appropriate authorization and may implement blinding rules to protect treatment allocation.

Report Generation

Reports are generated dynamically from current data, ensuring that displayed information reflects the latest database state. The generation process queries relevant tables, applies appropriate filtering and aggregation, formats results for display, and renders through the Shiny interface with interactive features.

```r
# Report generation example
enrollment_report <- function(study_id) {
  subjects <- get_subjects(study_id)

  subjects |>
    group_by(site_id, enrollment_status) |>
    summarize(
      count = n(),
      .groups = "drop"
    ) |>
    pivot_wider(
      names_from = enrollment_status,
      values_from = count,
      values_fill = 0
    )
}
```

Interactive Features

Reports incorporate interactive elements enabling user-directed exploration. Data tables support sorting, filtering, and column visibility toggling. Charts support zooming, panning, and data point selection. Drill-down links navigate from summary metrics to underlying detail records.

Export Capabilities

All reports support export to common formats including PDF for distribution, Excel for further analysis, and CSV for data interchange. Export includes metadata documenting generation timestamp, filter criteria, and user identity for audit purposes.

Part III: Data Management and Export

This section describes capabilities for managing, transforming, and exporting study data. These features support data cleaning, analysis preparation, and regulatory submission requirements.

12. Data Explorer

Interactive Data Viewing

The data explorer provides an interactive interface for viewing and analyzing study data. Users can select data sources (forms, visits, subject populations), apply filters to subset data, choose display columns, and sort results. The interface updates dynamically as users modify selection criteria.

Filtering Capabilities

The filtering system supports multiple filter types:

Categorical Filters select records matching specified category values. Users can select single values, multiple values, or exclude specific values. Filter options are populated from actual data values, ensuring that displayed options exist in the dataset.

Numeric Filters select records within specified ranges. Users can define minimum values, maximum values, or both. Open-ended ranges (greater than X, less than Y) support common filtering scenarios.

Date Filters select records within date ranges. Preset ranges (last 7 days, this month, last quarter) provide convenient shortcuts. Custom ranges enable precise date specification.

Text Filters search for matching strings within text fields. Search supports exact match, contains, starts with, and ends with operators. Case sensitivity is configurable.

Saved Views

Users can save filter configurations as named views for reuse. Saved views store all filter criteria, column selections, and sort orders. Shared views enable standardization of commonly needed data subsets across the study team.

13. Export Service

Export Formats

The export service generates data packages in multiple formats addressing different downstream requirements:

CSV (Comma-Separated Values) provides maximum compatibility with statistical software, spreadsheet applications, and data processing pipelines. CSV export includes configurable options for delimiter character, text quoting, date format, and missing value representation.

Excel (XLSX) generates formatted workbooks with appropriate column widths, headers, and data type formatting. Multi-sheet workbooks can include related data tables, data dictionaries, and metadata. Excel export supports formulas for calculated columns.

SAS Transport (XPT) produces files in SAS XPORT format suitable for regulatory submission. Variable names are truncated to SAS naming limits, and variable labels preserve descriptive information. This format is required for FDA electronic submissions.

JSON (JavaScript Object Notation) exports data in structured format for programmatic consumption. JSON export preserves data types and hierarchical relationships. This format supports integration with web services and modern applications.

R Data (RDS) preserves R objects with complete metadata including factor levels, attributes, and class information. This format provides lossless export for R-based analysis pipelines.

Export Configuration

Export operations are configured through an interface specifying data selection (forms, subjects, date ranges), format options, and output destination. Configurations can be saved for repeated exports with consistent parameters.

```r
# Export configuration example
export_config <- list(
  forms = c("DEMOGRAPHICS", "VITALS", "ADVERSE_EVENTS"),
  subjects = "enrolled",
  format = "csv",
  options = list(
    delimiter = ",",
    date_format = "%Y-%m-%d",
    na_string = ""
  )
)
```

Audit Trail for Exports

Every export operation generates an audit record documenting the export requestor, timestamp, data scope, and destination. For regulated studies, export audit enables demonstration that data access was controlled and traceable. Export logs support investigation if data appears in unauthorized locations.

14. Data Dictionary

Dictionary Generation

The system automatically generates data dictionaries from form specifications and database schema. Generated dictionaries include variable names and labels, data types and formats, valid value ranges and code lists, derivation formulas for calculated variables, and source form and field references.

Dictionary Formats

Data dictionaries export in multiple formats:

Human-Readable format produces formatted documents suitable for review by clinical personnel and regulatory authorities. This format emphasizes clarity and completeness over machine readability.

Machine-Readable formats (JSON, XML) enable automated processing for tasks such as validation rule generation, SDTM mapping, and database documentation. These formats follow standard schemas for interoperability.

Define-XML format produces CDISC-compliant metadata documentation for regulatory submission. Define-XML captures variable metadata in the structure required by FDA electronic submission standards.

Version Management

Dictionary versions track as form specifications change throughout the study. Each dictionary version documents the specification version it describes, enabling accurate interpretation of data collected under each version.

15. Query Management System

Query Lifecycle

The query management system implements a standard lifecycle for data discrepancy resolution:

Generation: Queries are created automatically by validation rules or manually by data managers identifying issues during review. Each query captures the affected record, the specific discrepancy, and the expected resolution.

Assignment: Queries route to appropriate personnel based on query type, affected site, and role assignments. Assignment rules ensure queries reach personnel able to investigate and resolve them.

Response: Site personnel provide responses explaining the discrepancy or proposing corrections. Responses are timestamped with user identity for accountability. Sites can request clarification if query intent is unclear.

Verification: Data managers review responses to confirm adequacy. Acceptable responses lead to closure. Inadequate responses return to the response stage with additional guidance.

Closure: Resolved queries are closed with documentation of the resolution. Closed queries remain in the audit trail for inspection purposes.

Query Metrics

The system tracks query metrics supporting process oversight:

Volume Metrics track query counts by type, site, form, and time period. Trend analysis identifies increasing error rates requiring intervention.

Timing Metrics track query age from generation through resolution. Aging reports highlight overdue queries requiring escalation.

Resolution Metrics track how queries are resolved (correction, confirmation of original value, unable to determine). Patterns in resolution types inform training and process improvement.

Auto-Query Configuration

Validation rules can be configured to generate queries automatically upon failure. Auto-query configuration specifies which rule failures generate queries, the query text template, and assignment routing. This automation ensures consistent query generation without manual intervention.

Part IV: GDPR Compliance Features

The General Data Protection Regulation (EU) 2016/679 establishes comprehensive data protection requirements for organizations processing personal data of EU residents. ZZedc implements the following GDPR-mandated features.

16. Data Encryption at Rest (Article 32)

Regulatory Requirement

Article 32 of the GDPR mandates that data controllers implement "appropriate technical and organisational measures to ensure a level of security appropriate to the risk" including "the pseudonymisation and encryption of personal data." This requirement reflects the regulation's risk-based approach to data protection, recognizing that encryption represents one of the most effective technical safeguards against unauthorized data access.

Technical Implementation

ZZedc implements transparent database encryption using SQLCipher, an open-source extension to SQLite that provides AES-256 encryption. The implementation addresses three critical aspects of encryption management:

Key Management

The system integrates with AWS Key Management Service (KMS) for enterprise deployments, providing hardware-backed key storage with comprehensive access logging. AWS KMS ensures that encryption keys never exist in plaintext outside of secure hardware modules, eliminating a common vulnerability in encryption implementations. For development and testing environments, the system supports environment variable-based key storage, enabling developers to work with encrypted databases without requiring cloud infrastructure. The key management subsystem automatically rotates keys according to configurable policies and maintains secure key escrow procedures for disaster recovery scenarios.

Encryption Algorithm

The implementation employs AES-256 encryption in CBC (Cipher Block Chaining) mode, representing the current industry standard for symmetric encryption. Each database page is encrypted independently, enabling efficient random access while maintaining security. HMAC-SHA512 authentication accompanies each encrypted block, providing tamper detection that alerts administrators to any unauthorized modification attempts. This authenticated encryption approach addresses both confidentiality and integrity requirements mandated by Article 32.

Key Derivation

For passphrase-based key generation, the system implements PBKDF2 (Password- Based Key Derivation Function 2) with 256,000 iterations. This computational cost deliberately slows key derivation, rendering brute-force attacks against weak passphrases impractical. The iteration count substantially exceeds NIST recommendations, reflecting the sensitive nature of clinical trial data and the extended timeframe over which such data must remain protected.

```
# Example: Initialize encrypted database
initialize_encrypted_database(
  db_path = "study_data.db",
  key_source = "aws_kms"
)
```

Compliance Verification

The system includes automated verification of encryption status through header inspection and key validation routines. When a database file is opened, the system verifies that the file header contains encrypted content rather than the standard SQLite signature, confirming that encryption is active. Audit logs record all key access events, including the identity of the requesting process, timestamp, and success or failure status. These logs support compliance documentation during regulatory inspections and provide forensic evidence in the event of security incidents.

17. Data Subject Access Request (Article 15)

Regulatory Requirement

Article 15 establishes the right of data subjects to obtain confirmation of whether personal data concerning them is being processed, and access to that data along with supplementary information about processing purposes, categories, and recipients. Controllers must respond to access requests without undue delay and within one month of receipt, with provisions for extension in complex cases.

Technical Implementation

The DSAR module provides a complete workflow for handling access requests, addressing each phase of the request lifecycle:

Request Intake

The system provides structured capture of request details through a dedicated intake interface that records the data subject's identity, contact information, and specific data categories requested. Upon submission, the system automatically calculates the response deadline based on the request receipt date, defaulting to 30 calendar days with provisions for extension to 90 days for complex requests involving large data volumes or multiple processing systems. The intake process generates a unique request identifier and initiates the tracking workflow, ensuring that no request is lost or overlooked.

Identity Verification

Before disclosing personal data, controllers must verify the identity of the requesting party to prevent unauthorized disclosure. The system implements a multi-factor verification workflow supporting various verification methods including government-issued identification documents, knowledge-based authentication using information only the data subject would know, and electronic identity verification services. The verification workflow documents all verification steps taken, the evidence reviewed, and the identity of the staff member who confirmed the subject's identity. This documentation protects the organization against claims of improper disclosure.

Data Collection

The system provides systematic identification and extraction of personal data across all database tables where subject information may reside. Administrators configure data source mappings that identify which tables contain personal data, which fields serve as subject identifiers, and which data categories each table represents. When processing a DSAR, the system queries each configured source, compiles the results, and associates them with the appropriate data category for the response. The collection process logs each data source accessed and the volume of data retrieved.

Response Generation

The system automates compilation of data packages in machine-readable formats as required by Article 15(3). Supported formats include JSON for maximum interoperability, CSV for spreadsheet compatibility, and structured PDF reports for human review. Each response package includes metadata describing the processing purposes, data categories, recipient categories, retention periods, and the data subject's rights. The system maintains a complete copy of each response for audit purposes, enabling the organization to demonstrate what information was provided and when.

```r
# Example: Create and process DSAR
request <- create_dsar_request(
  subject_email = "participant@example.com",
  subject_name = "John Doe",
  request_type = "ACCESS",
  requested_by = "dpo"
)

# Verify identity before processing
verify_subject_identity(
  request_id = request$request_id,
  verification_method = "DOCUMENT",
  verified_by = "admin"
)
```

Compliance Verification

All DSAR activities are logged with timestamps, creating an immutable audit trail demonstrating compliance with response deadlines. The system generates compliance reports showing request volumes, average response times, and any instances where extensions were required. These metrics support both internal quality management and regulatory reporting obligations.

18. Right to Rectification (Article 16)

Regulatory Requirement

Article 16 provides data subjects the right to obtain rectification of inaccurate personal data and completion of incomplete data. This right recognizes that data accuracy is fundamental to fair processing and that individuals should have recourse when organizations hold incorrect information about them.

Technical Implementation

The rectification system implements a controlled correction workflow that maintains data integrity while enabling corrections:

Request Processing

The system provides structured intake with specification of the data identified as incorrect and the proposed corrections. Each rectification request captures the specific field or fields requiring correction, the current value held by the system, the value the data subject believes to be correct, and any supporting documentation. The request workflow routes submissions to appropriate reviewers based on the data category affected, ensuring that clinical data corrections receive medical review while administrative corrections follow streamlined procedures.

Verification

The review process validates correction requests through comparison with source documents, consultation with the data subject for clarification, and verification against external authoritative sources where available. Reviewers document their verification activities, including what sources were consulted and what conclusions were reached. The system supports partial approvals where some requested corrections are validated while others require additional investigation or are determined to be unfounded.

Audit Trail

The system maintains complete history of original values, corrections, and authorizations. When a correction is applied, the system preserves the original value with timestamp and the identity of the person who entered it, the corrected value with timestamp and the identity of the person who authorized the correction, the reason for the correction, and any supporting documentation references. This audit history satisfies both GDPR accountability requirements and FDA 21 CFR Part 11 requirements for maintaining original data visibility.

Third-Party Notification

The system tracks data recipients requiring notification of corrections as mandated by Article 19. When personal data has been disclosed to third parties, the system generates notification requirements listing each recipient and the specific corrections to communicate. Staff record notification activities, including the date, method, and confirmation of receipt. The system can generate reports demonstrating that all required notifications were completed.

The system distinguishes between clinical data corrections (which follow FDA-mandated workflows with dual approval) and administrative data corrections (which follow streamlined GDPR-only workflows). This distinction reflects the different regulatory contexts while maintaining consistent audit capabilities.

Clinical Trial Considerations

In clinical research contexts, rectification must be balanced against data integrity requirements. The system implements a dual-approval workflow for clinical data corrections, requiring both the investigator and sponsor representative to authorize changes to efficacy or safety data. This approach ensures regulatory compliance while respecting data subject rights.

19. Right to Erasure (Article 17)

Regulatory Requirement

Article 17, commonly known as the "right to be forgotten," provides data subjects the right to erasure of personal data under specified circumstances, subject to exceptions for legal obligations, public health research, and archiving purposes in the public interest. The regulation recognizes that this right must be balanced against other legitimate interests.

Technical Implementation

The erasure module implements a sophisticated workflow accommodating both GDPR rights and regulatory retention requirements:

Legal Hold Management

The system provides capability to place data under legal hold when regulatory requirements supersede erasure rights. Clinical trial regulations typically require retention of subject data for periods ranging from 2 years (FDA post-approval) to 25 years (pediatric studies, some EU member states). When a legal hold applies, the system blocks erasure while documenting the legal basis for retention, the expected hold duration, and the authority responsible for the hold decision. Data subjects receive notification that their erasure request cannot be fulfilled immediately, along with explanation of the legal basis for continued retention. The system automatically reviews legal holds at expiration, enabling erasure to proceed once retention obligations conclude.

Selective Erasure

The system provides granular control over erasure scope, enabling compliance with Article 17(3) exceptions while maximizing respect for data subject rights. Administrators configure erasure rules that specify which data categories may be erased under which circumstances, which data must be retained regardless of erasure requests, and which data may be anonymized as an alternative to deletion. This configuration enables organizations to erase contact information and identifiers while retaining anonymized clinical observations necessary for study integrity.

Anonymization Alternative

When complete erasure conflicts with research integrity requirements, the system supports irreversible anonymization as an alternative that satisfies both data subject rights and scientific obligations. The anonymization process removes all direct identifiers, generalizes quasi-identifiers such as dates and geographic information to prevent re-identification, and applies statistical disclosure control techniques appropriate to the data type. The system verifies anonymization effectiveness by assessing re-identification risk and documents the techniques applied for regulatory review.

Third-Party Coordination

The system tracks and coordinates notification to downstream recipients who received the data subject's personal data. Following the Article 19 notification requirement, the system identifies all third parties to whom data was disclosed, generates notification requirements specifying the data to be erased, tracks notification delivery and acknowledgment, and documents any cases where recipients cannot be notified or refuse to comply. This coordination ensures that erasure requests propagate throughout the data processing ecosystem.

```r
# Example: Create legal hold before erasure
hold <- create_legal_hold(
  subject_id = "SUBJ-001",
  hold_type = "REGULATORY",
  hold_reason = "FDA retention requirement - 2 years post-study",
  held_by = "compliance_officer"
)
```

```r
# Erasure request blocked by legal hold
request <- create_erasure_request(
  subject_email = "participant@example.com",
  erasure_grounds = "WITHDRAWAL",
  requested_by = "dpo"
)
# Returns: status = "LEGAL_HOLD"
```

Regulatory Conflict Resolution

The system implements automatic detection of conflicts between GDPR erasure rights and FDA/ICH retention requirements, placing data under regulatory hold and documenting the legal basis for retention. When an erasure request arrives for a subject enrolled in an active or recently completed clinical trial, the system checks applicable retention requirements and, if a conflict exists, automatically applies a legal hold with appropriate documentation.

20. Right to Restriction of Processing (Article 18)

Regulatory Requirement

Article 18 provides data subjects the right to obtain restriction of processing in circumstances including accuracy disputes, unlawful processing, controller no longer needing the data, and pending objection reviews. During restriction, data may only be stored, and other processing requires subject consent.

Technical Implementation

The restriction module implements processing controls at the field level:

Scope Definition

The system allows restrictions to apply at varying levels of granularity, from specific data categories to processing purposes to entire subject records. When creating a restriction, administrators specify whether the restriction applies to all processing of the subject's data, to specific categories such as health data or financial data, or to specific processing purposes such as marketing or research. This flexibility enables precise compliance with restriction requests while minimizing disruption to legitimate processing activities not covered by the restriction.

Processing Blocks

The system implements technical controls preventing restricted data from being included in analyses, exports, or reports. When data is under restriction, the system flags those records in query results to alert users, excludes restricted records from aggregate analyses unless specifically authorized, blocks export of restricted data to external systems, and removes restricted subjects from mailing lists and communication campaigns. These technical controls operate automatically, reducing reliance on staff awareness and manual procedures.

Allowed Processing

The system supports configuration of permitted processing activities during restriction as specified in Article 18(2). Storage is always permitted, as is processing for establishment, exercise, or defense of legal claims. The system also allows processing with the data subject's explicit consent and processing for protection of another person's rights. Administrators configure which processing activities qualify under each exception, and the system logs all processing of restricted data with documentation of the applicable exception.

Lift Procedures

The system implements controlled workflow for removing restrictions with mandatory subject notification as required by Article 18(3). Before lifting a restriction, the system verifies that the grounds for restriction no longer apply, documents the basis for lifting, and generates notification to the data subject informing them that restriction will be lifted and processing will resume. The subject receives reasonable notice before processing resumes, enabling them to exercise other rights if desired.

Processing Attempt Logging

All attempts to process restricted data are logged, enabling demonstration of compliance and identification of system components requiring modification. These logs capture the user or process attempting access, the specific data requested, the timestamp, and the outcome (blocked or permitted with exception). Regular review of these logs helps identify gaps in technical controls and training needs.

21. Right to Data Portability (Article 20)

Regulatory Requirement

Article 20 provides data subjects the right to receive their personal data in a structured, commonly used, machine-readable format and to transmit that data to another controller. This right applies to data provided by the subject and processed by automated means on the basis of consent or contract performance.

Technical Implementation

The portability module supports multiple export formats and transfer mechanisms:

Export Formats

The system generates portable data packages in multiple standardized formats to maximize interoperability. JSON (JavaScript Object Notation) provides a widely-supported structured format readable by virtually any modern software system. XML offers an alternative structured format with schema validation capabilities. CSV enables import into spreadsheet applications for subjects who prefer tabular data representation. For clinical research data, CDISC ODM (Operational Data Model) format ensures compatibility with other clinical data management systems. Each format includes appropriate metadata describing the data structure, enabling receiving systems to interpret the data correctly.

Data Scope

The system provides configurable inclusion of provided data versus derived data. Article 20 specifically applies to data provided by the subject, but organizations may choose to include derived data as a matter of good practice. The configuration distinguishes between directly provided data such as questionnaire responses and uploaded documents, observed data such as vital signs recorded during visits, and derived data such as calculated scores and analysis results. Administrators configure which categories to include by default while preserving flexibility for specific requests.

Direct Transfer

The system supports controller-to-controller transfer as required by Article 20(2) where technically feasible. The transfer workflow verifies the receiving controller's identity and authorization, establishes a secure transfer channel using TLS encryption, transmits the data package with integrity verification, confirms successful receipt, and documents the transfer for audit purposes. The system supports both API-based transfers for automated interoperability and secure file transfer for organizations without API capabilities.

Security

All portability exports employ encryption and integrity verification to protect data during transmission and storage. Export packages are encrypted using the subject's chosen password or a secure key exchange mechanism. Each package includes cryptographic hash values enabling verification that the data was not modified during transfer. Audit logs record all export activities, including the data included, format selected, and delivery method used.

```r
# Example: Generate portable data package
export <- generate_portability_export(
  request_id = request$request_id,
  format = "JSON",
  include_metadata = TRUE
)

# Initiate controller transfer
transfer <- initiate_controller_transfer(
  request_id = request$request_id,
  recipient_controller = "New Research Institution",
  transfer_method = "SECURE_API"
)
```

22. Right to Object (Article 21)

Regulatory Requirement

Article 21 provides data subjects the right to object to processing based on legitimate interests or public interest, and an absolute right to object to direct marketing. Controllers must cease processing upon objection unless they demonstrate compelling legitimate grounds that override subject interests.

Technical Implementation

The objection module manages processing objections with configurable scope:

Objection Types

The system supports distinct objection categories reflecting the different legal standards that apply. General processing objections under Article 21(1) require the controller to demonstrate compelling legitimate grounds to continue processing. Profiling objections address automated decision-making that produces legal or similarly significant effects. Direct marketing objections are absolute and require immediate cessation without qualification. Research objections engage the Article 21(6) exception, requiring assessment of whether processing is necessary for task performance in the public interest. Each objection type follows appropriate workflow with documentation requirements matching the applicable legal standard.

Processing Cessation

The system implements immediate cessation of objected processing activities upon receipt of a valid objection. For direct marketing objections, cessation is automatic and unconditional. For other objection types, the system places processing on hold pending assessment of whether compelling grounds exist to continue. During this assessment period, no further processing occurs except as necessary to evaluate the objection itself. The cessation mechanism integrates with downstream systems through API notifications, ensuring that objections propagate to all processing components.

Override Capability

The system supports documented override for compelling legitimate grounds with appropriate escalation and notification. When a controller determines that compelling grounds exist to continue processing despite an objection, the system captures detailed documentation of the grounds relied upon, requires senior management authorization, generates notification to the data subject explaining the decision and their right to complain to supervisory authorities, and creates records for potential regulatory review. This override mechanism exists only for objections where the law permits override; it is unavailable for absolute rights such as direct marketing objections.

Marketing Preferences

The system maintains granular channel-level marketing preferences enabling subjects to opt out of specific communication channels while remaining subscribed to others. Subjects may object to email marketing while continuing to receive postal communications, or may object to telephone contact while permitting text messages. The preference system integrates with marketing automation platforms through standard APIs, ensuring that preferences are respected across all communication systems.

Research Exception Handling

For scientific research processing, the system implements the Article 21(6) exception, which provides that the right to object does not apply where processing is necessary for the performance of a task carried out for reasons of public interest. The system requires documentation of the public interest basis, assessment of whether the specific processing is necessary for that purpose, and maintenance of records supporting the exception determination.

23. Consent Management (Articles 6-7)

Regulatory Requirement

Articles 6 and 7 establish consent as a lawful basis for processing and specify requirements for valid consent including freely given, specific, informed, and unambiguous indication. Consent must be as easy to withdraw as to give, and controllers must be able to demonstrate that consent was obtained.

Technical Implementation

The consent management system provides comprehensive consent lifecycle management:

Purpose-Specific Consent

The system supports granular consent collection for distinct processing purposes, reflecting the GDPR requirement that consent be specific. Each processing purpose is defined separately with its own consent text, enabling subjects to consent to some purposes while declining others. For clinical research, typical purposes include primary research (use of data for the specified study objectives), secondary research (future use for related research questions), biobank storage (retention of biological samples), and commercial development (use in developing commercial products). Subjects receive clear information about each purpose and make independent decisions about each.

Consent Records

The system creates immutable records of consent including all elements necessary to demonstrate valid consent. Each consent record captures the identity of the consenting subject, the specific purpose consented to, the version of consent text presented, the timestamp of consent, the method by which consent was obtained (electronic signature, paper form, oral with witness), and any additional context relevant to demonstrating that consent was freely given. These records are stored with cryptographic integrity protection, ensuring they cannot be modified after creation.

Withdrawal Processing

The system implements immediate effect of withdrawal with downstream processing cessation as required by Article 7(3). When a subject withdraws consent, the system immediately flags the affected processing purposes, notifies dependent systems to cease processing, generates confirmation to the subject, and initiates any data deletion or anonymization required by the withdrawal. The withdrawal workflow respects the principle that withdrawal must be as easy as giving consent, providing a simple interface without requiring justification or imposing barriers.

Consent Refresh

The system provides automated identification of aging consents requiring refresh. Consent validity may be limited by the passage of time, changes in processing circumstances, or regulatory requirements for periodic reconfirmation. The system tracks consent age and generates alerts when refresh is needed, facilitating proactive outreach to subjects before consent expiration affects processing activities.

```r
# Example: Record granular consent
record_consent(
  subject_id = "SUBJ-001",
  purpose_id = purpose$purpose_id,
  consent_given = TRUE,
  consent_method = "ELECTRONIC",
  consent_text_version = "1.2",
  recorded_by = "coordinator"
)

# Check consent before processing
check <- check_consent(
  subject_id = "SUBJ-001",
  purpose_code = "RESEARCH_PRIMARY"
)
```

24. Data Retention Enforcement (Article 5)

Regulatory Requirement

Article 5(1)(e) establishes the storage limitation principle, requiring that personal data be kept no longer than necessary for processing purposes. Organizations must define retention periods, justify those periods, and implement processes to dispose of data when retention periods expire.

Technical Implementation

The retention module implements policy-based retention management:

Retention Policies

The system supports configurable policies by data category with comprehensive legal basis documentation. Each retention policy specifies the data category covered, the retention period, the legal basis for that period (regulatory requirement, contractual obligation, legitimate interest, or consent), the disposal action to take at expiration (deletion, anonymization, or archival), and review requirements. Policies can specify different retention periods for different purposes, ensuring that data is retained exactly as long as needed for each processing activity.

Automated Review

The system provides scheduled identification of data reaching retention limits. A configurable review process runs at specified intervals, identifying records approaching retention expiration and generating work queues for retention review. Reviewers assess each record to confirm that retention obligations have concluded, verify that no exceptions apply, and authorize disposal. The system supports bulk review for routine cases while ensuring individual attention for complex situations.

Disposal Actions

The system implements configurable disposal methods appropriate to the data category and regulatory context. Deletion permanently removes data from the database with cryptographic erasure of backup copies. Anonymization removes identifying elements while preserving data utility for aggregate analysis. Archival transfers data to long-term storage with restricted access for cases where data may be needed for future legal or regulatory purposes. Each disposal action is logged with timestamp, authorization, and method.

Legal Hold Integration

The system automatically suspends disposal for data under legal hold. When retention review identifies data subject to an active legal hold, the system bypasses normal disposal processing and flags the record for review after hold expiration. This integration ensures that retention automation does not inadvertently destroy data required for legal or regulatory purposes.

Retention Conflict Resolution

When GDPR retention limits conflict with regulatory requirements (e.g., FDA 15-year retention for clinical data), the system documents the legal basis and applies the longer retention period. The system maintains clear records of which requirement drives retention, enabling accurate response to data subject inquiries and demonstrating compliance with both frameworks.

25. Privacy Impact Assessment (Article 35)

Regulatory Requirement

Article 35 mandates Data Protection Impact Assessments for processing operations likely to result in high risk to data subjects' rights and freedoms, including systematic and extensive profiling, large-scale processing of special category data, and systematic monitoring of public areas.

Technical Implementation

The PIA module provides a structured assessment framework:

Risk Categories

The system enables systematic identification of processing risks across confidentiality, integrity, and availability dimensions. Confidentiality risks address unauthorized access, disclosure, or data breach. Integrity risks address unauthorized modification, data corruption, or loss of accuracy. Availability risks address system failures, data loss, or denial of service. Within each dimension, the system prompts assessors to consider specific risk scenarios relevant to the processing activity, ensuring comprehensive coverage.

Risk Scoring

The system implements quantitative risk assessment with likelihood and impact ratings. Each identified risk is scored on standardized scales for likelihood (rare, unlikely, possible, likely, almost certain) and impact (negligible, minor, moderate, major, severe). The system calculates risk scores and aggregates them to provide overall risk profiles. Visual representations help decision-makers understand risk distribution and prioritize mitigation efforts.

Mitigation Tracking

The system documents risk mitigation measures with effectiveness assessment. For each identified risk, assessors record proposed mitigation measures, the expected risk reduction, the implementation timeline, and the responsible party. The system tracks implementation progress and enables re-assessment of residual risk after mitigation. This tracking ensures that identified risks are actually addressed rather than merely documented.

DPO Consultation

The system implements workflow for mandatory DPO review as required by Article 35(2). Before a high-risk processing activity commences, the DPO receives the assessment for review, provides formal advice, and records any concerns or recommendations. The system documents whether DPO advice was followed and, if not, the justification for proceeding contrary to advice.

Supervisory Authority Consultation

The system tracks requirements for Article 36 prior consultation when residual risk remains high despite mitigation measures. If assessment determines that high risk cannot be mitigated sufficiently, the system generates documentation packages for supervisory authority consultation, tracks submission and response, and prevents processing from commencing until consultation concludes.

```r
# Example: Create and assess PIA
pia <- create_pia_assessment(
  assessment_name = "Phase III Trial Data Processing",
  processing_description = "Collection of sensitive health data",
  legal_basis = "EXPLICIT_CONSENT",
  created_by = "dpo"
)

# Add risk assessment
add_risk_assessment(
  pia_id = pia$pia_id,
  risk_category = "DATA_BREACH",
  risk_description = "Unauthorized access to trial data",
  likelihood = "LOW",
  impact = "HIGH",
  mitigation_measures = "Encryption, access controls, audit logging",
  residual_risk = "LOW",
  assessed_by = "security_officer"
)
```

26. Breach Notification (Articles 33-34)

Regulatory Requirement

Article 33 requires notification to supervisory authorities within 72 hours of becoming aware of a personal data breach likely to result in risk to data subject rights. Article 34 requires communication to affected data subjects when the breach is likely to result in high risk. These provisions ensure transparency and enable affected individuals to take protective measures.

Technical Implementation

The breach notification module implements a complete incident response workflow:

Incident Recording

The system provides structured capture of breach details through a dedicated incident recording interface. Each incident record captures the discovery datetime, the nature of the breach (confidentiality, integrity, availability), the scope including estimated number of affected subjects and data categories involved, the likely consequences for data subjects, and initial containment measures taken. The recording interface guides staff through required information collection, ensuring that essential details are captured during the often-chaotic initial response period.

Timeline Tracking

The system implements automatic monitoring of the 72-hour notification deadline with escalation alerts. Upon incident recording, the system calculates the deadline based on the awareness timestamp and begins countdown tracking. As the deadline approaches, the system generates escalating alerts to response team members and management. A deadline dashboard provides real-time visibility into notification status. This tracking ensures that regulatory deadlines are met despite the complexity and stress of incident response.

Risk Assessment

The system provides structured assessment of breach severity and subject impact using criteria from regulatory guidance. Assessors evaluate the type of breach, the nature and sensitivity of affected data, the ease of identification of affected individuals, the severity of consequences for subjects, the number of affected individuals, and any special characteristics of affected individuals (such as vulnerable populations). The assessment determines whether supervisory authority notification is required and whether subject notification is necessary.

Notification Generation

The system generates template-based notifications for both supervisory authorities and affected subjects. Authority notifications include all elements required by Article 33(3): the nature of the breach, DPO contact information, likely consequences, and measures taken. Subject notifications include the elements required by Article 34(2): clear language description of the breach, DPO contact information, likely consequences, and measures taken and recommended. Templates ensure consistency and completeness while enabling customization for specific circumstances.

Remediation Tracking

The system documents containment and remediation actions throughout the incident lifecycle. Each action is recorded with timestamp, responsible party, and outcome. The system supports action assignment and tracking, ensuring that remediation tasks are completed. Post-incident review documents lessons learned and process improvements, creating organizational learning from security events.

```r
# Example: Report and assess breach
incident <- report_breach_incident(
  breach_type = "UNAUTHORIZED_ACCESS",
  breach_description = "Unauthorized login detected",
  discovery_datetime = Sys.time(),
  estimated_subjects = 50,
  data_categories = "HEALTH,IDENTIFICATION",
  reported_by = "security_team"
)

# Check 72-hour deadline
deadline <- check_72_hour_deadline(incident$incident_id)
# Returns hours remaining and notification status
```

Part V: Good Clinical Practice (GCP) Features

Good Clinical Practice guidelines, established by the International Council for Harmonisation (ICH E6), define standards for clinical trial design, conduct, and reporting. ZZedc implements the following GCP-aligned features.

27. Protocol Compliance Monitoring

Regulatory Requirement

ICH E6(R2) Section 4.5 requires that clinical trials be conducted in accordance with the protocol. Section 5.18 mandates monitoring to verify protocol adherence. Protocol deviations may affect subject safety, data integrity, and study validity, making systematic monitoring essential.

Technical Implementation

The protocol compliance module provides systematic monitoring capabilities:

Protocol Definition

The system enables structured capture of protocol elements including visits, procedures, and eligibility criteria in a machine-readable format. Protocol definitions specify the study schedule with timing requirements, the assessments and procedures required at each visit, the inclusion and exclusion criteria for subject eligibility, and the endpoints and analysis populations. This structured capture enables automated monitoring that would be impossible with document-based protocol management.

Visit Scheduling

The system provides automated scheduling with configurable window calculations. When a subject enrolls or completes a reference visit, the system automatically calculates target dates for subsequent visits based on protocol-defined intervals. Visit windows specify acceptable variation from target dates, typically with narrower windows for critical assessments and wider windows for routine follow-up. The scheduling engine accounts for weekends and holidays when calculating permissible dates, reducing unnecessary deviations.

Deviation Detection

The system implements real-time identification of protocol deviations through continuous monitoring of data against protocol requirements. Deviations detected include visit timing outside protocol windows, missing required assessments, inclusion/exclusion criteria violations discovered after enrollment, and dosing errors or interruptions. Detection occurs as data is entered, enabling immediate corrective action rather than discovery during later monitoring visits.

Deviation Classification

The system categorizes deviations by severity and type according to sponsor- defined classification schemes. Severity categories typically distinguish major deviations affecting subject safety or data integrity from minor deviations with limited impact. Type categories identify the nature of the deviation such as visit window, informed consent, inclusion/exclusion, procedural, or study medication. This classification enables appropriate response and supports aggregate analysis of deviation patterns.

Corrective Action Tracking

The system documents responses to deviations and tracks completion of corrective actions. For each deviation, the system records the immediate response taken, root cause analysis, corrective actions to prevent recurrence, and verification that corrections were effective. This tracking demonstrates to regulators that deviations receive appropriate attention and that the organization learns from compliance failures.

```r
# Example: Define protocol and monitor compliance
protocol <- create_protocol(
  protocol_code = "STUDY-001",
  protocol_version = "2.0",
  study_phase = "PHASE_3",
  therapeutic_area = "ONCOLOGY",
  created_by = "medical_director"
)

# Add visit schedule
add_protocol_visit(
  protocol_id = protocol$protocol_id,
  visit_code = "SCREENING",
  visit_name = "Screening Visit",
  visit_day = -14,
  window_before = 7,
  window_after = 0,
  is_required = TRUE
)

# Record protocol deviation
create_protocol_deviation(
  protocol_id = protocol$protocol_id,
  subject_id = "SUBJ-001",
  deviation_type = "VISIT_WINDOW",
  deviation_description = "Screening visit outside window",
  deviation_date = Sys.Date(),
  reported_by = "coordinator"
)
```

28. Adverse Event Management

Regulatory Requirement

ICH E6(R2) Section 4.11 requires documentation and reporting of adverse events. Serious adverse events (SAEs) require expedited reporting to sponsors, institutional review boards, and regulatory authorities. Proper AE management is essential for subject safety and regulatory compliance.

Technical Implementation

The adverse event module provides comprehensive AE/SAE management:

Event Capture

The system provides structured recording of adverse events with support for standardized medical terminology coding. Each event record captures the verbatim term as reported, the coded term using MedDRA (Medical Dictionary for Regulatory Activities), onset and resolution dates, outcome, and relationship to study treatment. The interface guides investigators through required fields while remaining efficient for high-volume data entry during busy clinic days.

Severity Grading

The system implements CTCAE (Common Terminology Criteria for Adverse Events) aligned severity grading using the standard Grade 1-5 scale. Grade 1 indicates mild events requiring no intervention. Grade 2 indicates moderate events requiring minimal intervention. Grade 3 indicates severe events with significant medical consequence. Grade 4 indicates life-threatening events requiring urgent intervention. Grade 5 indicates death. The system provides grading guidance and validation to ensure consistent application across sites.

Causality Assessment

The system implements systematic causality evaluation workflow to assess the relationship between adverse events and study treatment. Investigators assess causality using standardized categories (not related, unlikely, possible, probable, definite) with guided criteria. The assessment considers temporal relationship, biological plausibility, effect of dechallenge and rechallenge, and alternative explanations. Documentation of causality reasoning supports regulatory review.

SAE Escalation

The system automatically identifies events meeting serious adverse event criteria based on regulatory definitions. SAE criteria include death, life- threatening event, hospitalization, persistent incapacity, congenital anomaly, and other medically important events. When an AE meets SAE criteria, the system immediately elevates the event, notifies responsible personnel, initiates expedited reporting workflows, and applies enhanced documentation requirements.

Expedited Reporting

The system tracks timelines for regulatory reporting requirements including 24-hour notification for fatal or life-threatening unexpected events, 7-day submission of initial reports for fatal or life-threatening events, and 15-day submission for other serious unexpected events. The system calculates deadlines based on awareness date, generates alerts as deadlines approach, produces regulatory-compliant report formats, and tracks submission and response.

Follow-up Tracking

The system documents event evolution and resolution through structured follow- up records. Each follow-up captures the current status, any changes in severity or causality assessment, new information about etiology or treatment, and resolution details when applicable. Follow-up tracking continues until the event resolves or stabilizes, ensuring complete documentation of the event course.

```r
# Example: Record adverse event with SAE upgrade
ae <- create_adverse_event(
  subject_id = "SUBJ-001",
  ae_term = "Headache",
  ae_description = "Moderate headache after dose",
  onset_date = Sys.Date(),
  severity_grade = "GRADE_2",
  reported_by = "investigator"
)

# Upgrade to SAE if hospitalization required
upgrade_to_sae(
  ae_id = ae$ae_id,
  sae_criteria = "HOSPITALIZATION",
  upgraded_by = "investigator"
)
```

29. CRF Completion Guidelines

Regulatory Requirement

ICH E6(R2) Section 4.9 requires that data be recorded accurately and in a manner that allows verification. CRF completion guidelines (CCGs) support consistent data collection by providing detailed instructions for each form field, reducing variability and errors.

Technical Implementation

The CCG module generates comprehensive field-level completion instructions:

Form Documentation

The system provides automatic generation of form completion guides from CRF metadata. Each guide includes the form purpose and context, completion timing and sequence, general instructions applicable to all fields, and special considerations for the form. Generation is automatic when CRF specifications change, ensuring that CCG documentation remains synchronized with actual forms.

Field Instructions

The system generates detailed guidance for each data field including the data type and format requirements, valid values and units of measure, source document requirements, collection procedures and timing, common errors to avoid, and examples of correct entries. These instructions draw from CRF specifications and supplementary guidance provided by study designers, creating comprehensive documentation without manual document creation.

Version Control

The system maintains CCG versioning aligned with CRF versions, ensuring that sites always have instructions matching the current form version. When CRFs are updated, the system generates updated CCGs and tracks which version is current at each site. Historical versions remain available for reference regarding data collected under prior versions.

Approval Workflow

The system implements review and approval process for CCG documents before distribution to sites. Medical monitors review clinical content, data managers review technical accuracy, and regulatory personnel review compliance considerations. Approved CCGs are distributed to sites with training acknowledgment tracking.

30. CRF Version Control

Regulatory Requirement

ICH E6(R2) Section 5.5.3 requires documentation of CRF modifications. FDA 21 CFR 312.62 requires maintenance of accurate case histories. Sponsors must maintain clear records of what data was collected using which form versions.

Technical Implementation

The version control module provides complete CRF lifecycle management:

Version History

The system maintains immutable records of all CRF versions throughout study conduct. Each version record includes the complete form specification at that version, the effective date range, the approvers and approval dates, and cross-references to related documents. This history enables reconstruction of exactly what form was in use at any point during the study, supporting data interpretation and regulatory inspection.

Change Documentation

The system captures detailed change logs with rationale for each modification. When a CRF is updated, the system records each field added, removed, or modified, the reason for the change, the assessment of impact on ongoing data collection, and any data migration requirements. This documentation demonstrates that changes were controlled and justified rather than arbitrary.

Comparison Tools

The system provides version-to-version comparison capabilities that highlight differences between CRF versions. Comparison reports identify added, removed, and modified fields, changes to validation rules, changes to field attributes such as labels or help text, and structural changes such as section reordering. These comparisons support impact assessment and training development.

Deployment Tracking

The system maintains records of version deployment to sites including the deployment date at each site, acknowledgment of receipt and training, and any site-specific implementation issues. This tracking ensures accountability for version currency and supports troubleshooting when sites report problems.

31. CRF Design Review

Regulatory Requirement

ICH E6(R2) Section 5.5.3 specifies sponsor responsibilities for CRF design. Industry best practices recommend formal design review processes involving medical, statistical, regulatory, and operational stakeholders to optimize data collection.

Technical Implementation

The review module implements structured CRF design review:

Review Cycles

The system supports multi-stage review with configurable reviewers at each stage. Typical stages include medical review for clinical content, statistical review for analysis requirements, regulatory review for compliance considerations, and operational review for site feasibility. Each stage has defined reviewers, completion criteria, and escalation procedures. The system tracks progress through review stages and alerts stakeholders to pending items.

Comment Tracking

The system provides structured capture and resolution of review comments. Comments are categorized by type (correction required, suggestion, clarification needed), severity, and affected form elements. Each comment tracks the commenter, the response, and resolution status. Comment discussion threads enable dialogue between reviewers and designers without losing context.

Approval Workflow

The system implements formal sign-off requirements by stage before CRF finalization. Approvers at each stage certify that they have reviewed the form and that it meets requirements within their domain. The system enforces approval sequence, preventing progression to later stages until earlier approvals are complete.

Design History

The system maintains complete record of review activities for regulatory inspection. The design history file includes all review comments and resolutions, all approval records, meeting notes and decision documentation, and change requests and their disposition. This history demonstrates appropriate design governance.

Part VI: FDA 21 CFR Part 11 Features

FDA 21 CFR Part 11 establishes requirements for electronic records and electronic signatures in FDA-regulated activities. ZZedc implements the following Part 11 features.

32. Electronic Signatures

Regulatory Requirement

21 CFR Part 11 Subpart C establishes requirements for electronic signatures including unique identification of the signer, signature manifestation showing the printed name, date/time, and meaning, and binding of the signature to the record such that it cannot be copied or transferred to falsify records.

Technical Implementation

The electronic signature module implements compliant e-signatures:

Signature Components

The system implements username plus password combination as the signature component in accordance with 11.200(a). Each signature requires entry of both credentials at the time of signing, ensuring that the signer is present and actively consenting to the signature. The system does not permit signature using cached credentials or single-factor authentication, maintaining the regulatory requirement for two-component signatures.

Signature Meaning

The system captures the meaning of each signature as required by 11.50. Upon signing, the signer selects from defined meanings including authorship (I created this content), review (I have reviewed and verified this content), approval (I authorize this action or content), and responsibility (I accept responsibility for this content). The meaning is recorded as part of the signature record and displayed with the signature manifestation.

Signature Manifestation

The system displays the printed name of the signer, the date and time of signature, and the signature meaning as required by 11.50. This manifestation appears wherever the signed record is displayed or printed, ensuring that viewers understand who signed, when, and for what purpose. The manifestation cannot be separated from the signed record.

Record Linking

The system implements cryptographic binding of signature to record content, ensuring that a signature cannot be transferred to a different record. Upon signing, the system computes a cryptographic hash of the record content and includes this hash in the signature record. Verification compares the current record hash to the stored hash, detecting any modification since signing. This binding prevents both intentional falsification and accidental association of signatures with wrong records.

Non-Repudiation

The system implements hash-based verification preventing signature forgery or repudiation. Because the signature includes a hash of the signer's credentials and the record content at signing time, no one can create a valid signature without possessing the signer's credentials and the exact record content. This provides non-repudiation: signers cannot plausibly deny their signatures, and others cannot forge signatures.

```r
# Example: Apply electronic signature
signature <- apply_electronic_signature(
  record_type = "CRF",
  record_id = "CRF-001-V1",
  signer_id = "dr_smith",
  signer_password = "****",
  signature_meaning = "APPROVAL",
  signature_reason = "Approving completed case report form"
)

# Verify signature integrity
verify_electronic_signature(
  signature_id = signature$signature_id
)
```

33. Audit Trail System

Regulatory Requirement

21 CFR 11.10(e) requires computer-generated, time-stamped audit trails that independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Audit trails must be available for agency review and copying.

Technical Implementation

The audit trail module provides comprehensive activity logging:

Automatic Capture

The system logs all record operations automatically without operator intervention or ability to bypass. The audit system operates at the database layer, capturing events before they can be intercepted or suppressed by application code. This automatic capture ensures complete coverage regardless of which interface or process modifies data. Operators cannot disable or circumvent audit logging.

Immutability

The system implements hash-chained audit records preventing undetected tampering. Each audit record includes a cryptographic hash of the previous record, creating a chain where modification of any record invalidates all subsequent hashes. The system periodically verifies chain integrity and alerts administrators to any detected anomalies. This immutability ensures that audit trails provide reliable evidence of actual system activity.

Timestamp Accuracy

The system generates timestamps independently of operator input, using server system time synchronized via NTP (Network Time Protocol). Operators cannot specify or modify timestamps, ensuring temporal accuracy.

The system logs any detected time synchronization failures, alerting administrators to conditions that might affect timestamp reliability.

Record Linkage

The system maintains direct association between audit entries and affected records through stable record identifiers. Each audit entry specifies the table, record identifier, and field affected. This linkage enables efficient retrieval of complete audit history for any record, supporting both routine review and regulatory inspection.

Retention

The system retains audit trails for the lifetime of the associated record plus the regulatory retention period. Retention configuration mirrors clinical data retention policies, ensuring that audit evidence remains available as long as the data it documents. The system prevents deletion of audit records while associated clinical records exist.

```
# Example: Audit trail verification
# All operations automatically logged
create_correction_request(
  table_name = "subjects",
  record_id = "SUBJ-001",
  field_name = "birth_date",
  current_value = "1990-01-01",
  proposed_value = "1990-01-02",
  correction_reason = "Transcription error",
  requested_by = "coordinator"
)

# Verify audit trail integrity
verify_correction_integrity(request_id)
```

Enhanced Security Event Monitoring

Beyond standard data operations, the audit system captures security-relevant events critical for compliance monitoring and threat detection.

Authentication Events

The system logs all authentication activity including successful logins, failed login attempts, session timeouts, and explicit logouts. Failed login attempts capture the username attempted, failure reason, IP address, and cumulative attempt count. This information supports detection of brute force attacks and credential stuffing attempts.

```
# Failed login attempts are automatically logged
log_failed_login(
  username = "john.doe",
  reason = "Invalid password",
  ip_address = "192.168.1.100",
  attempt_count = 3
)

# Account lockout after threshold exceeded
log_account_lockout(
  username = "john.doe",
  reason = "Too many failed attempts",
  duration_minutes = 30
)
```

Access Control Events

The system captures all changes to user permissions and roles. Each role change records the user affected, previous role, new role, administrator making the change, and justification. Password changes are logged regardless of whether self-initiated or administratively reset. These records support access review audits and privilege escalation detection.

```
# Role changes create audit records
log_role_change(
  user_id = "jane.smith",
  old_role = "coordinator",
  new_role = "data_manager",
  changed_by = "admin_user",
  reason = "Promotion per study staffing change"
)
```

Configuration Change Tracking

System configuration changes are logged with before and after values, enabling reconstruction of system state at any point in time. This tracking supports change control requirements and provides evidence for regulatory inspections demonstrating that configuration changes followed proper procedures.

```
# Configuration changes are audited
log_config_change(
  setting_name = "session_timeout_minutes",
  old_value = "30",
  new_value = "15",
  changed_by = "admin_user",
  reason = "Security policy update per IT audit findings"
)
```

System Operations Logging

Backup operations, maintenance activities, and system restarts are logged with operational details. Backup logs capture file paths, sizes, durations, and success status. This information supports disaster recovery verification and demonstrates that appropriate data protection measures are operational.

Anomaly Detection and Alerting

The audit system includes automated anomaly detection to identify suspicious patterns that may indicate security incidents or compliance violations.

Detection Categories

The system monitors for the following anomaly types:

Anomaly Type	Indicators	Risk Level
Brute Force Attack	Multiple failed logins per user	HIGH
Credential Stuffing	High volume of failed logins across users	HIGH
Data Exfiltration	Unusual export volume or timing	MEDIUM
Privilege Escalation	Multiple role changes in short period	MEDIUM
Off-Hours Activity	Significant activity outside business hours	LOW

Risk Scoring

Each detected anomaly contributes to a cumulative risk score. The system categorizes overall risk as LOW (score < 25), MEDIUM (25-49), or HIGH (50+). This scoring enables prioritization of security response efforts.

```r
# Run anomaly detection
anomalies <- detect_audit_anomalies(lookback_hours = 24)

# Example output:
# $risk_level: "MEDIUM"
# $risk_score: 35
# $alerts:
#   - type: "BRUTE_FORCE_SUSPECTED"
#     severity: "HIGH"
#     message: "Multiple failed logins for users: john.doe"
#     count: 7
#   - type: "OFF_HOURS_ACTIVITY"
#     severity: "LOW"
#     message: "15 events during off-hours"
```

Configurable Thresholds

Organizations can adjust detection thresholds to match their risk tolerance and operational patterns. Default thresholds are conservative but can be modified for environments with different usage patterns.

Advanced Audit Search and Analytics

The audit system provides comprehensive search capabilities for compliance review and incident investigation.

Multi-Criteria Search

Search filters include free text search across operations and details, event type filtering (INSERT, UPDATE, DELETE, LOGIN, etc.), user filtering for activity by specific individuals, date range filtering for time-bounded investigations, and severity filtering for high-priority events.

```r
# Search audit trail with multiple criteria
results <- search_audit_trail(
  search_term = "password",
  event_types = c("PASSWORD_CHANGE", "LOGIN_FAILED"),
  users = c("john.doe", "jane.smith"),
  date_from = as.Date("2024-01-01"),
  date_to = as.Date("2024-03-31"),
  limit = 500
)
```

Audit Statistics

The system generates summary statistics for operational monitoring and compliance reporting. Statistics include total events by period, events grouped by type, most active users, most accessed tables, and daily event trends.

```r
# Generate weekly audit statistics
stats <- get_audit_statistics(period = "week")

# Returns:
# - total_events: 1,247
# - events_per_day: 178.1
# - events_by_type: INSERT (523), UPDATE (412), SELECT (189), ...
# - events_by_user: coordinator_1 (312), data_mgr (287), ...
# - security_events: 23
```

These statistics support compliance reporting, capacity planning, and identification of unusual patterns requiring investigation.

34. Data Correction Workflow

Regulatory Requirement

21 CFR 11.10(e) requires that changes to records not obscure previously recorded information. Original entries must remain visible with the date/time and identity of the person making the change. This requirement preserves data integrity while permitting necessary corrections.

Technical Implementation

The data correction module implements controlled change management:

Correction Requests

The system implements a formal request process for data changes that separates the request from the authorization. Requestors identify the data requiring correction, document the correct value, and provide rationale. Requests enter a workflow queue for review, ensuring that corrections receive appropriate oversight. This separation of duties prevents unauthorized modifications and creates accountability.

Reason Documentation

The system requires mandatory capture of correction rationale meeting regulatory expectations. Every correction must include an explanation of why the correction is needed (such as transcription error, source document clarification, or protocol deviation). Vague or missing reasons are rejected by validation. This documentation supports regulatory review and demonstrates that corrections are legitimate rather than arbitrary data manipulation.

Dual Control

The system implements configurable approval requirements for corrections based on data sensitivity and correction type. Critical clinical endpoints may require approval from both the investigator and sponsor medical monitor. Administrative corrections may require only supervisor approval. The configuration reflects risk-based principles, applying greater control to higher-risk modifications while maintaining efficiency for routine corrections.

Original Preservation

The system retains original values with full history including who entered the original value and when, who requested the correction and why, who approved the correction and when, and the correction timestamp. This complete history ensures that original entries remain visible as required by regulation. Reports can display either current values or complete change history as needed.

Override Procedures

The system supports documented escalation for urgent corrections that cannot wait for normal approval cycles. Override authority is restricted to designated personnel. Each override is prominently flagged in audit records and reported to management. This procedure balances operational necessity with appropriate controls.

35. System Validation Framework

Regulatory Requirement

21 CFR 11.10(a) requires that systems be validated to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records. Validation provides documented evidence

that systems function as intended.

Technical Implementation

The validation framework implements a complete IQ/OQ/PQ approach:

Installation Qualification (IQ)

The system provides verification of correct installation and configuration through automated installation verification testing. IQ tests confirm that all required software components are present and at correct versions, configuration files contain expected values, database schemas match specifications, and system resources meet minimum requirements. IQ generates documentation suitable for inclusion in validation packages.

Operational Qualification (OQ)

The system implements testing of system functions under normal conditions to verify that all features work as designed. OQ tests exercise each system function with valid inputs, verify correct outputs and behaviors, test error handling with invalid inputs, and confirm user interface functionality. The comprehensive OQ test suite covers all regulatory-relevant functions.

Performance Qualification (PQ)

The system supports verification under actual use conditions to confirm ongoing proper operation. PQ tests use production-like data volumes and user loads, verify performance meets acceptance criteria, test system behavior under stress conditions, and confirm backup and recovery procedures. PQ may be repeated periodically to confirm continued proper operation.

Validation Documentation

The system automates generation of validation reports meeting regulatory expectations. Reports include test specifications with expected results, actual results with evidence, deviation documentation for any failures, overall pass/fail status, and approval signatures. This automation reduces documentation burden while ensuring completeness and consistency.

```r
# Example: Run validation suite
validation <- run_validation_suite(
  validation_type = "FULL",
  environment = "PRODUCTION"
)

# Generate validation report
generate_validation_report(
  validation_id = validation$validation_id,
  format = "PDF"
)
```

36. Change Control System

Regulatory Requirement

21 CFR 11.10(k)(2) requires controls over the revision and change of system documentation and operational systems. Changes must be evaluated for impact on validation status and subject to appropriate approval before implementation.

Technical Implementation

The change control module manages system modifications:

Change Requests

The system implements formal request process with categorization of change type, urgency, and scope. Requestors describe the proposed change, identify affected components, and propose implementation approach. Requests are assigned unique identifiers and enter the tracking workflow. This formality ensures that all changes are documented and visible to stakeholders.

Impact Assessment

The system provides systematic evaluation of change impacts on system validation, data integrity, regulatory compliance, and operational processes. Impact assessors consider whether the change affects validated functions, whether revalidation is required, whether training is needed, and whether documentation must be updated. Assessment findings inform approval decisions and implementation planning.

Approval Workflow

The system implements multi-level approval for changes based on impact severity. Minor changes with limited impact may require only technical approval. Significant changes affecting validated functions require quality assurance approval. Major changes affecting regulatory compliance require management approval. The system enforces approval requirements before implementation can proceed.

Implementation Tracking

The system documents change implementation with verification of completion. Implementation records capture who performed the change, when, and what specific actions were taken. Post-implementation verification confirms that the change achieved intended effects without adverse impacts. Implementation evidence is linked to the change request for complete traceability.

Validation Integration

The system triggers revalidation activities as appropriate based on change impact assessment. Changes affecting validated functions generate revalidation requirements specifying which tests must be repeated. The system tracks revalidation completion and prevents sign-off until validation is current. This integration ensures that validation status remains accurate after changes.

37. Access Controls

Regulatory Requirement

21 CFR 11.10(d) requires limiting system access to authorized individuals. 11.10(g) requires authority checks ensuring that only authorized individuals can use the system, access operations, or perform functions. These controls protect against unauthorized access and inappropriate actions.

Technical Implementation

The access control system implements role-based security:

Role Definitions

The system supports configurable roles with specific permission profiles reflecting organizational structure and job functions. Common roles include data entry (create and modify records), reviewer (read-only access with query capability), investigator (clinical oversight with signature authority), and administrator (system configuration and user management). Organizations define roles matching their specific needs and assign users to appropriate roles.

Function-Level Control

The system implements permissions at the operation level for granular access control. Beyond data access, the system controls who can perform specific functions such as running reports, exporting data, applying signatures, and modifying configurations. This granularity ensures that users can perform their job functions without access to unrelated capabilities that might create risk.

Session Management

The system implements secure session handling with configurable timeout to prevent unauthorized access through abandoned sessions. Sessions expire after configured inactivity periods, requiring reauthentication. Session tokens are generated with cryptographic randomness and transmitted only over encrypted connections. The system prevents session hijacking and fixation attacks.

Access Logging

The system maintains complete record of access attempts including both successful and failed authentications, session creation and termination, privilege changes, and access denial events. These logs support security monitoring, incident investigation, and compliance demonstration. Regular log review helps identify suspicious activity before security incidents occur.

Part VII: Integration, Deployment, and Quality Assurance

This section covers system integration capabilities, deployment options, and quality assurance infrastructure. These features enable ZZedc deployment in diverse organizational and technical environments.

38. Protocol-CRF Linkage

Regulatory Requirement

ICH E6(R2) requires traceability between protocol requirements and data collection elements. FDA expects clear documentation of data collection rationale. This traceability demonstrates that collected data serves specific scientific purposes.

Technical Implementation

The linkage module connects protocol elements to CRF fields:

Objective Mapping

The system maintains explicit links between protocol objectives and data collection endpoints. Each protocol objective connects to the specific variables used to assess that objective. This mapping enables verification that all objectives have corresponding data collection and that all collected data serves identified purposes. Traceability reports demonstrate the scientific rationale for data collection.

Visit-Form Mapping

The system associates forms with protocol visits to define when each form should be completed. This mapping drives schedule displays showing which forms are due at each visit, compliance checking to identify missing forms, and data entry workflows presenting appropriate forms for each visit. The association ensures that data collection follows the protocol schedule.

Traceability Matrix

The system generates automated requirement traceability matrices linking protocol requirements to implementing CRF elements. These matrices satisfy regulatory expectations for documentation of CRF development rationale and support change impact assessment by identifying which CRF elements depend on which protocol requirements.

39. Study Closeout Management

Regulatory Requirement

ICH E6(R2) Section 4.13 and FDA regulations require systematic study closeout with complete documentation, reconciliation of all data, and appropriate archiving. Closeout procedures ensure that study records are complete and suitable for long-term retention.

Technical Implementation

The closeout module provides structured study completion:

Closeout Checklist

The system provides comprehensive checklists of required activities customized by study type and regulatory requirements. Standard checklist items include query resolution, SAE reconciliation, signature completion, and documentation review. The system tracks checklist progress, assigns responsibility for each item, and prevents premature database lock until all critical items are complete.

Data Reconciliation

The system supports systematic data verification processes comparing clinical data against external sources such as safety databases, laboratory systems, and central facilities. Reconciliation identifies discrepancies requiring resolution before closeout. The system tracks discrepancy investigation and resolution, documenting the final reconciled state.

Database Lock

The system implements controlled database locking procedures with appropriate authorization requirements. Database lock prevents further data modification, preserving the dataset for analysis and regulatory submission. The lock process verifies that prerequisite conditions are met, captures authorization, and timestamps the lock event. Post-lock modifications require documented deviation procedures.

Archive Preparation

The system generates documentation of archive contents including data inventory, file manifests, and retention schedules. Archive packages include the database in preservation-appropriate format, audit trails, system configuration records, and metadata necessary for future interpretation. This documentation supports long-term retention and future data access.

40. Master Field Library

Regulatory Requirement

CDISC standards promote standardized data collection using common data elements. Regulatory submissions benefit from consistent variable definitions that align with submission standards. Standardization improves data quality and facilitates meta-analysis across studies.

Technical Implementation

The field library provides standardized field definitions:

CDISC Alignment

The system provides pre-defined fields aligned with SDTM (Study Data Tabulation Model) domains and controlled terminology. Standard fields include proper variable names, labels, data types, and value sets matching CDISC specifications. Using library fields ensures that collected data maps correctly to submission formats without transformation complexity.

Controlled Terminology

The system integrates CDISC controlled terminology codelists, ensuring that categorical data uses standardized values. The terminology database includes NCI Thesaurus codes enabling regulatory submission. Terminology versions are tracked, and the system alerts administrators when updates are available.

Reusability

The system enables centralized definitions for consistency across studies within an organization. Once defined, library fields can be incorporated into any CRF design. Changes to library fields propagate to all studies using those fields, ensuring organizational consistency. Usage tracking shows which studies employ each field, supporting impact assessment for terminology updates.

41. CRF Template Library

Purpose

Pre-built form templates accelerate study setup by providing starting points for common data collection requirements. Templates embody best practices for form design and regulatory compliance, reducing the expertise required for effective CRF development.

Technical Implementation

Standard Templates

The system includes templates for common clinical trial forms:

- Demographics (CDISC DM domain aligned)
- Medical History (MH domain aligned)
- Vital Signs (VS domain aligned)
- Physical Examination
- Concomitant Medications (CM domain aligned)
- Adverse Events (AE domain aligned)
- Laboratory Results (LB domain aligned)
- Electrocardiogram
- Informed Consent Tracking
- Study Completion/Discontinuation

Each template includes field definitions, validation rules, completion guidelines, and SDTM mapping specifications.

Therapeutic Area Extensions

The library includes therapeutic area-specific templates addressing common requirements in major therapeutic areas:

- Oncology (tumor assessment, RECIST criteria, survival endpoints)
- Neurology (cognitive assessments, disability scales)
- Cardiology (cardiac events, ECG interpretation)
- Immunology (autoimmune disease activity)
- Infectious Disease (pathogen detection, treatment response)

Customization

Templates serve as starting points for study-specific customization. The system enables copying templates into study-specific forms, modifying fields and validation as needed, and maintaining linkage to source templates for update propagation when desired.

42. Conditional Logic System

Purpose

Conditional logic enables dynamic form behavior where field visibility, editability, and validation depend on other field values. This capability reduces form complexity by showing only relevant fields and prevents collection of inapplicable data.

Technical Implementation

Show/Hide Rules

The system implements field visibility based on prior responses. Fields can be hidden when not applicable and shown when relevant. For example, pregnancy questions appear only for female subjects of childbearing potential, and detailed symptom questions appear only when the subject reports symptoms. This conditional visibility reduces form clutter and prevents collection of inapplicable data.

Enable/Disable Rules

The system controls field editability based on conditions separate from visibility. Disabled fields remain visible for reference but cannot be modified. This supports scenarios where data should be shown for context but not entered, such as calculated fields or data imported from external systems.

Validation Rules

The system implements conditional validation requirements that apply only when specific conditions are met. Required field validation can depend on other field values. Range checks can vary based on subject characteristics. Cross-field validation can enforce complex business rules. This conditional validation ensures appropriate data quality checks without inappropriate constraints.

Skip Logic

The system enables structured navigation based on responses, automatically advancing past inapplicable sections. Skip logic reduces data entry time and prevents errors from attempting to complete inapplicable sections. The navigation flow remains intuitive despite complexity of underlying rules.

43. Calculated Fields

Purpose

Calculated fields automatically derive values from other field entries, ensuring consistent computation of clinical measures and reducing transcription errors. Standard calculations implemented in the system reduce variability across sites and studies.

Technical Implementation

Standard Calculations

The system provides common clinical calculations:

- BMI (Body Mass Index) with WHO category assignment
- BSA (Body Surface Area) using configurable formulas (DuBois, Mosteller)
- eGFR (estimated Glomerular Filtration Rate) using CKD-EPI or MDRD
- Age calculation from birth date and reference date
- Duration calculations between dates
- Score summation for multi-item instruments

Custom Calculations

The system supports definition of study-specific calculations using a formula syntax supporting arithmetic operations, date arithmetic, conditional logic, and reference to other field values. Custom calculations undergo validation to ensure syntactic correctness and reference to existing fields.

Calculation Timing

Calculations update automatically when source values change, providing immediate feedback. Calculated values are stored in the database for query efficiency while maintaining derivation traceability. The system logs calculation formula versions to ensure correct interpretation of historical values.

44. CRF Designer

Purpose

The CRF designer provides visual form creation capabilities enabling study teams to develop data collection forms without programming knowledge. Visual design improves form usability by enabling designers to see forms as users will experience them.

Technical Implementation

Visual Layout

The system enables grid-based field positioning allowing precise control of form appearance. Designers drag fields to desired positions, seeing the layout as users will experience it. Grid-based positioning ensures consistent alignment and professional appearance. Layout can be adjusted for different screen sizes and printing formats.

Section Management

The system supports logical grouping of related fields into sections with headers and visual separation. Sections can expand and collapse to manage form complexity. Section-level permissions enable restricting access to sensitive portions of forms. Conditional section visibility hides entire groups of fields based on skip logic.

Field Properties

The system provides comprehensive field configuration including data type, label, help text, placeholder, validation rules, conditional logic, and database mapping. All field properties are accessible through an intuitive interface without requiring technical knowledge. Property changes take effect immediately in the preview.

Preview Generation

The system provides form preview before deployment, enabling validation of design decisions. Preview displays the form exactly as users will see it, including conditional logic behavior. Designers can test various scenarios by entering sample data and observing form response. This preview capability catches design issues before deployment to production.

45. Google Sheets Integration

Purpose

Google Sheets integration enables study configuration through familiar spreadsheet interfaces, reducing technical barriers to study setup. Teams can define forms, users, and study parameters in spreadsheets, then import configurations into ZZedc.

Technical Implementation

Configuration Import

The system imports study configuration from Google Sheets including:

- User accounts and role assignments
- Form definitions with field specifications
- Validation rules and conditional logic
- Visit schedules and form assignments
- Controlled terminology and code lists

Authentication

Integration uses the googlesheets4 package with OAuth 2.0 authentication. Service account credentials enable automated import without interactive login. Credentials are stored securely following Google security best practices.

Bidirectional Sync

For ongoing studies, the system supports synchronization between Google Sheets and the database. Updates in either location can propagate to the other, enabling collaborative configuration management. Conflict resolution rules handle concurrent modifications.

```r
# Example: Import configuration from Google Sheets
setup_zzedc_from_gsheets(
  auth_sheet_name = "Study_Users",
  dd_sheet_name = "Data_Dictionary",
  project_name = "My Clinical Trial"
)
```

46. Deployment Options

Local Deployment

Local deployment runs ZZedc on a single computer, suitable for individual researchers or small teams without server infrastructure. The application launches in a web browser while the R process runs locally. This deployment requires no network configuration and provides maximum simplicity.

```r
# Local deployment
launch_zzedc(
  host = "127.0.0.1",
  port = 3838,
  launch.browser = TRUE
)
```

Server Deployment

Server deployment runs ZZedc on a dedicated server accessible to multiple users over a network. Shiny Server (open source or professional) manages application lifecycle and user connections. This deployment supports concurrent users with centralized data storage.

Container Deployment

Docker containers provide reproducible deployment environments with all dependencies pre-configured. Container images include the R runtime, required packages, and application code. Orchestration platforms (Docker Compose, Kubernetes) manage multi-container deployments with database, application, and reverse proxy components.

```yaml
# docker-compose.yml example
services:
  zzedc:
    image: zzedc:latest
    ports:
      - "3838:3838"
    volumes:
      - ./data:/app/data
    environment:
      - ZZEDC_DB_ENCRYPTION_KEY_SOURCE=aws_kms
```

Cloud Deployment

Cloud platforms (AWS, Azure, Google Cloud) provide scalable infrastructure for enterprise deployments. Deployment patterns include:

- Virtual machine hosting with Shiny Server
- Container services (ECS, AKS, GKE)
- Managed R platforms (Posit Connect)
- Serverless options (Lambda with container images)

Cloud deployment enables auto-scaling based on demand, managed database services with automated backup, and integration with cloud security services.

47. Backup and Restore

Backup System

The backup module provides automated and manual database backup capabilities:

Scheduled Backups

The system supports scheduled backup creation at configurable intervals. Backup schedules specify frequency (hourly, daily, weekly), retention periods determining how long backups are kept, and storage destinations for backup files. Scheduled backups run automatically without operator intervention.

On-Demand Backups

Administrators can create backups at any time through the administrative interface. On-demand backups are appropriate before significant changes, at study milestones, or when prompted by unusual activity. Each backup receives a timestamp and optional description for identification.

Backup Verification

The system verifies backup integrity by computing checksums at creation and verifying them before restore. Periodic verification confirms that stored backups remain readable. Failed verification triggers alerts to administrators.

Restore System

Restore Procedures

The restore process replaces current database content with backup data. Before restore, the system creates a backup of current state to enable recovery if restore fails. Restore requires administrative authorization and logs all details for audit purposes.

Point-in-Time Recovery

When combined with transaction logging, the system supports point-in-time recovery to any moment between backups. This capability minimizes data loss in failure scenarios by enabling recovery to the last committed transaction before failure.

```r
# Example: Backup and restore
# Create backup
backup <- create_backup(
  description = "Pre-database-lock backup",
  created_by = "admin"
)

# Restore from backup
restore_backup(
  backup_id = backup$backup_id,
  authorized_by = "admin"
)
```

48. Performance Optimization

Database Optimization

The system implements several database performance optimizations:

Connection Pooling

The pool package manages a pool of database connections, eliminating connection establishment overhead for each request. Pool size is configurable based on expected concurrent users.

Query Optimization

Frequently executed queries use indexed columns. Complex queries employ query planning to identify efficient execution paths. Result caching reduces redundant database access for unchanged data.

Pagination

Large result sets are paginated to limit memory usage and response time. The data explorer and report modules display configurable page sizes with navigation controls for accessing additional pages.

Application Optimization

Lazy Loading

Modules and data load on demand rather than at application startup. This approach reduces initial load time and memory usage for features that users may not access during a session.

Caching

Computed values that change infrequently (such as reference data and aggregate statistics) are cached in memory. Cache invalidation occurs when underlying data changes. Configurable cache expiration ensures freshness for time-sensitive information.

Reactive Efficiency

Reactive expressions are structured to minimize unnecessary recalculation. Dependencies are scoped precisely to avoid triggering updates for unrelated changes. Observers are throttled where appropriate to prevent excessive processing during rapid data changes.

49. Testing Infrastructure

Test Framework

The system employs testthat for comprehensive automated testing:

Unit Tests

Unit tests verify individual functions in isolation. Each module includes tests for all exported functions, edge cases, and error conditions. Mock objects substitute for external dependencies (database, file system) to enable isolated testing.

Integration Tests

Integration tests verify interactions between modules. These tests exercise complete workflows such as subject enrollment, data entry, and report generation. Integration tests use test databases configured similarly to production.

End-to-End Tests

End-to-end tests verify complete user scenarios through the application interface. These tests confirm that the user experience matches expectations and that all components work together correctly.

Test Coverage

The system maintains high test coverage with over 3,000 individual tests. Coverage reports identify untested code paths. New features require accompanying tests before acceptance.

Continuous Integration

GitHub Actions workflows run the complete test suite on every code change. Tests execute across multiple R versions and operating systems (Ubuntu, Windows, macOS). Failed tests block code integration until resolved.

```r
# Run test suite
devtools::test()

# Run specific test file
testthat::test_file("tests/testthat/test-auth-module.R")

# Generate coverage report
covr::package_coverage()
```

Conclusion

ZZedc provides a comprehensive electronic data capture platform addressing the requirements of clinical research across regulatory frameworks. The system's modular architecture enables organizations to implement features appropriate to their specific regulatory context while maintaining flexibility for diverse clinical research applications.

Several characteristics distinguish ZZedc from alternative approaches:

Open Source Foundation: Complete source code availability enables verification of compliance implementations, customization for specific requirements, and freedom from vendor dependencies. Organizations can inspect exactly how regulatory features are implemented rather than relying on vendor attestations.

Integrated Compliance: Rather than treating regulatory compliance as an add-on, ZZedc incorporates compliance requirements throughout the architecture. Audit trails, access controls, and data protection are fundamental to the system design rather than retrofitted features.

Modern Technology: Built on current R and Shiny versions with Bootstrap 5 interface components, the system provides a contemporary user experience while leveraging the statistical computing capabilities of the R ecosystem. The technology stack is actively maintained with clear upgrade paths.

Deployment Flexibility: Support for local, server, container, and cloud deployment enables organizations to choose infrastructure appropriate to their security requirements, scalability needs, and technical capabilities. The same application code runs across all deployment models.

Documentation Depth: This document, along with accompanying vignettes and API documentation, provides comprehensive guidance for deployment, configuration, and operation. The documentation serves both as user guidance and as a reference implementation for regulatory-compliant EDC systems.

The implementation emphasizes automation of compliance activities, reducing manual burden while improving consistency and reliability. Comprehensive audit trails support regulatory inspections and demonstrate organizational commitment to data protection and research integrity.

Each feature described in this document has been designed with reference to underlying regulatory requirements, implemented following software engineering best practices, and validated through extensive automated testing. Organizations deploying ZZedc benefit from this rigorous development approach, gaining confidence that their EDC system supports rather than undermines their compliance obligations.

Appendix A: Feature Summary

The following table summarizes all features documented in this whitepaper:

#	Feature	Domain	Regulatory Basis
1	Technology Stack	Architecture	-
2	Database Architecture	Architecture	-
3	Modular Architecture	Architecture	-
4	Security Architecture	Architecture	21 CFR 11.10
5	Configuration Management	Architecture	-
6	Form Rendering	Core EDC	ICH E6(R2) 4.9
7	Data Validation	Core EDC	ICH E6(R2) 4.9
8	Visit Management	Core EDC	ICH E6(R2) 4.5
9	Subject Management	Core EDC	ICH E6(R2)
10	Multi-Site Support	Core EDC	ICH E6(R2) 5.18
11	Reporting System	Core EDC	-
12	Data Explorer	Data Management	-
13	Export Service	Data Management	GDPR Art. 20
14	Data Dictionary	Data Management	CDISC
15	Query Management	Data Management	ICH E6(R2)
16	Data Encryption	GDPR	Article 32
17	DSAR	GDPR	Article 15
18	Rectification	GDPR	Article 16
19	Erasure	GDPR	Article 17
20	Restriction	GDPR	Article 18
21	Portability	GDPR	Article 20
22	Objection	GDPR	Article 21
23	Consent Management	GDPR	Articles 6-7
24	Retention	GDPR	Article 5
25	Privacy Impact	GDPR	Article 35
26	Breach Notification	GDPR	Articles 33-34
27	Protocol Compliance	GCP	ICH E6(R2) 4.5
28	Adverse Events	GCP	ICH E6(R2) 4.11
29	CCG Generator	GCP	ICH E6(R2) 4.9
30	CRF Version Control	GCP	ICH E6(R2) 5.5.3
31	CRF Design Review	GCP	ICH E6(R2) 5.5.3
32	Electronic Signatures	FDA	21 CFR 11.50
33	Audit Trail	FDA	21 CFR 11.10(e)
34	Data Correction	FDA	21 CFR 11.10(e)
35	System Validation	FDA	21 CFR 11.10(a)
36	Change Control	FDA	21 CFR 11.10(k)
37	Access Controls	FDA	21 CFR 11.10(d,g)
38	Protocol-CRF Linkage	Integration	ICH E6(R2)
39	Study Closeout	Integration	ICH E6(R2) 4.13
40	Field Library	Integration	CDISC
41	Template Library	Integration	-
42	Conditional Logic	Integration	-
43	Calculated Fields	Integration	-
44	CRF Designer	Integration	-
45	Google Sheets	Integration	-

#	Feature	Domain	Regulatory Basis
46	Deployment Options	Operations	-
47	Backup and Restore	Operations	21 CFR 11.10
48	Performance	Operations	-
49	Testing	Quality	21 CFR 11.10(a)

Appendix B: Glossary

AE: Adverse Event. Any untoward medical occurrence in a subject administered a pharmaceutical product.

ALCOA+: Acronym for data quality attributes: Attributable, Legible, Contemporaneous, Original, Accurate, plus Complete, Consistent, Enduring, Available.

CCG: CRF Completion Guidelines. Detailed instructions for completing case report forms.

CDISC: Clinical Data Interchange Standards Consortium. Organization developing clinical data standards.

CFR: Code of Federal Regulations. United States federal agency regulations.

CRF: Case Report Form. Document designed to record protocol-required information.

CTCAE: Common Terminology Criteria for Adverse Events. Standard severity grading system.

DPO: Data Protection Officer. Role required by GDPR for certain organizations.

DSAR: Data Subject Access Request. Request from individual to access their personal data.

EDC: Electronic Data Capture. Computerized system for clinical trial data collection.

GCP: Good Clinical Practice. International standard for clinical trial conduct.

GDPR: General Data Protection Regulation. EU data protection law.

ICH: International Council for Harmonisation. Organization developing pharmaceutical guidelines.

IQ/OQ/PQ: Installation Qualification, Operational Qualification, Performance Qualification. Validation phases.

MedDRA: Medical Dictionary for Regulatory Activities. Standardized medical terminology.

PBKDF2: Password-Based Key Derivation Function 2. Cryptographic key derivation algorithm.

SAE: Serious Adverse Event. Adverse event meeting regulatory criteria for seriousness.

SDTM: Study Data Tabulation Model. CDISC standard for regulatory submission datasets.

References

Clinical Data Interchange Standards Consortium. (2021). CDISC Standards. https://www.cdisc.org/standards

European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Official Journal of the European Union, L119, 1-88.

International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. (2016). Integrated Addendum to ICH E6(R1): Guideline for Good Clinical Practice E6(R2).

U.S. Food and Drug Administration. (2003). 21 CFR Part 11: Electronic Records; Electronic Signatures. Code of Federal Regulations, Title 21.

U.S. Food and Drug Administration. (2017). Part 11, Electronic Records; Electronic Signatures - Scope and Application: Guidance for Industry.

Article 29 Data Protection Working Party. (2017). Guidelines on Data Protection Impact Assessment (DPIA) (wp248rev.01).

Article 29 Data Protection Working Party. (2018). Guidelines on Personal data breach notification under Regulation 2016/679 (wp250rev.01).

European Medicines Agency. (2010). Reflection paper on expectations for electronic source data and data transcribed to electronic data collection tools in clinical trials. EMA/INS/GCP/454280/2010.

National Institute of Standards and Technology. (2017). Digital Identity Guidelines: Authentication and Lifecycle Management. NIST Special Publication 800-63B.

Posit PBC. (2024). Shiny: Web Application Framework for R. https://shiny.posit.co/

R Core Team. (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.