

Biomass Open Origin Standard for Tracking (BOOST)

W3C Community Group Final Specification

Version v3.4.2-10-gb83f99a

August 13, 2025

Peter Tittmann
Liam Killroy

BOOST W3C Community Group

Abstract

The Biomass Open Origin Standard for Tracking (BOOST) data standard defines a comprehensive, interoperable framework for tracking biomass materials through complex supply chains. BOOST enables transparent, verifiable, and consistent data exchange to support sustainability verification, regulatory compliance, and supply chain integrity across the biomass economy. The standard implements a TraceableUnit (TRU)-centric model supporting continuous traceability, multi-species composition management, and comprehensive plant part categorization across 33 interconnected entities organized into 7 thematic areas.

Status of This Document

This specification was published by the Biomass Open Origin Standard for Tracking (BOOST) W3C Community Group (<https://www.w3.org/community/boost-01/>). It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the W3C Community Final Specification Agreement (FSA) other conditions apply. Learn more about W3C Community and Business Groups at <https://www.w3.org/community/>.

This document is governed by the W3C Community License Agreement (CLA). A human-readable summary is available at <https://www.w3.org/community/about/process/cla-deed/>.

Publication as a Community Group Report does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

How to Give Feedback

This specification is primarily developed on GitHub (<https://github.com/carbondirect/BOOST>). The best way to contribute to this specification is to:

1. File issues and suggestions in the BOOST GitHub repository (<https://github.com/carbondirect/BOOST/issues>)
2. Submit pull requests for specific changes
3. Participate in community discussions via GitHub Discussions (<https://github.com/carbondirect/BOOST/discussions>)
4. Join the W3C Community Group mailing list (<https://lists.w3.org/Archives/Public/public-boost-01/>) for broader discussions

Contents

1	Executive Summary	1
1.1	What BOOST Accomplishes	1
1.2	Overarching Design Principles	1
1.2.1	TraceableUnits (TRUs) as Foundation	1
1.2.2	Physical Reality Mapping	1
1.3	Continuous Traceability Framework	2
1.4	Value Proposition: Why BOOST Reduces Costs and Complexity	2
1.4.1	Multi-Method Identification Benefits	2
1.4.2	Regulatory Compliance Through Strategic Field Combinations	3
1.5	Tolerance Standards and Practical Considerations	3
2	Introduction	4
2.1	Community Development Process	4
2.2	Current Development Status	5
2.3	Participation and Feedback	6
2.4	Purpose and Scope	6
2.5	Background and Motivation	6
2.6	Relationship to Existing Standards	6
2.7	Community Group Process	7
3	Use Cases and Requirements	7
3.1	Primary Use Cases	7
3.1.1	California Biomass Supply Chain Tracking	7
3.1.2	Multi-Certification Scheme Management	7
4	Conformance	8
5	BOOST Traceability System	8
5.1	Key Implementation Features	8
5.1.1	Flexible Identification Framework	8
5.1.2	Critical Tracking Points	8
5.1.3	Multi-Species Support	9
5.1.4	Complete Processing Chain Documentation	9
6	Data Model Architecture	9
6.1	Key Features	10
6.1.1	Comprehensive Entity System	10
6.1.2	Enhanced Geographic Integration	10
6.2	Entity Organization by Thematic Areas	10
6.3	Foreign Key Conventions	11
7	Core Data Entities	11
7.1	Traceable Unit	11
7.2	Material Processing	14
7.3	Processing History	16

7.4	Location History	18
7.5	Biometric Identifier	18
7.6	Identification Method	19
7.7	Organization	20
7.8	Certificate	22
7.9	CertificationBody	23
7.10	CertificationScheme	24
7.11	Audit	25
7.12	BOOST Operator Entity Validation Schema	26
7.13	Material	27
7.14	Species Component	28
7.15	Supplier	29
7.16	Customer	30
7.17	Supply Base	31
7.18	Supply Base Report	31
7.19	Equipment	32
7.20	Transaction	33
7.21	Transaction Batch	36
7.22	Sales Delivery Document	38
7.23	Measurement Record	38
7.24	Claim	39
7.25	Verification Statement	40
7.26	BOOST Moisture Content Validation Rules	41
7.27	Geographic Data	41
7.28	Tracking Point	42
7.29	LCFS Pathway	43
7.30	LCFS Reporting	44
7.31	BioRAM Pathway	45
7.32	BioRAM Reporting	46
7.33	Product Group	48
7.34	Energy Carbon Data	48
7.35	Data Reconciliation	49
7.36	Mass Balance Account	50
8	Plant Part Categorization System	51
8.1	Introduction and Regulatory Context	51
8.1.1	Regulatory Drivers	51
8.1.2	Conceptual Framework	52
8.2	Standardized Plant Parts Taxonomy	52
8.3	Physical Arrangement Framework	53
8.3.1	Arrangement Categories	53
8.3.2	LCA and BECCS Integration	54
9	Schema Definitions	54
9.1	JSON Schema Format	54
9.2	Business Logic Validation	55

10	Serialization and Exchange	55
10.1	JSON-LD as Primary Format	55
11	JSON-LD Context and Semantic Web Integration	55
11.1	JSON-LD Overview	56
11.2	BOOST Context Definition	56
11.3	Vocabulary Mappings	57
11.3.1	Schema.org Integration	57
11.3.2	W3C PROV Ontology	58
11.3.3	GS1 Vocabulary	58
11.4	Entity Context Examples	58
11.4.1	TraceableUnit with Context	58
11.4.2	Transaction with Linked Data	59
11.5	Advanced Features	60
11.5.1	Named Graphs	60
11.5.2	Framing	61
11.5.3	Compaction and Expansion	61
11.6	Context Negotiation	62
11.6.1	Content Type Headers	62
11.6.2	Profile Parameters	62
11.7	Implementation Guidance	62
11.7.1	Python Implementation	62
11.7.2	JavaScript Implementation	63
11.8	Semantic Validation	64
11.8.1	SHACL Constraints	64
11.8.2	Reasoning and Inference	64
11.9	Benefits and Use Cases	65
11.9.1	Interoperability Benefits	65
11.9.2	Supply Chain Use Cases	65
12	Regulatory Program Compliance	65
12.1	Low Carbon Fuel Standard (LCFS) Overview	65
12.1.1	Regulatory Context	65
12.1.2	BOOST's Role in LCFS Compliance	66
12.2	LCFS Entity Integration	66
12.2.1	LcfsPathway Entity	66
12.2.2	Enhanced Transaction Entity	66
12.2.3	LcfsReporting Entity	67
12.3	Programmatic Reporting Workflows	67
12.3.1	Quarterly Report Generation	67
12.3.2	Credit Calculation Methodology	67
12.3.3	Data Reconciliation Process	68
12.4	Implementation Examples	68
12.4.1	Renewable Diesel Production Example	68
12.4.2	Credit Calculation Example	68
12.5	Multi-Program Compliance Framework	69

12.5.1 Renewable Fuel Standard (RFS) Integration	69
12.5.2 EU Renewable Energy Directive (RED II) Compliance	69
12.5.3 Regional Program Extensions	69
12.6 Data Quality and Compliance	70
12.6.1 CARB Data Validation Requirements	70
12.6.2 Third-Party Verification	70
12.7 Technical Implementation	70
12.7.1 API Endpoints for LCFS Data	70
12.7.2 Automated Report Generation	70
12.7.3 Error Handling and Validation	71
12.8 Regulatory Change Management	71
13 Security Considerations	72
13.1 Data Privacy	72
13.2 Data Integrity	72
14 Examples	72
14.1 Basic TraceableUnit Example	72
15 Resources & Community	73
15.1 Presentations & Demonstrations	73
15.2 Community Participation	73
A Entity Relationship Diagrams	73
A.1 Complete Entity Relationship Overview	73
A.2 Thematic Area Diagrams	75
B JSON Schema Reference	75
B.1 Schema Validation Rules	75
B.2 Context Definitions	75
C Python Reference Implementation	75
C.1 Overview	76
C.2 Installation	76
C.2.1 Prerequisites	76
C.2.2 Dependencies	76
C.3 Architecture	77
C.4 Core Components	77
C.4.1 SchemaLoader	77
C.4.2 DynamicBOOSTValidator	78
C.4.3 BOOSTClient	79
C.5 Dynamic Schema Adaptation	80
C.5.1 Automatically Handled Changes	80
C.5.2 Schema Change Detection	81
C.6 Usage Examples	81
C.6.1 Basic Workflow	81
C.6.2 Certification Management	83

C.6.3 Mass Balance Validation	83
C.7 Integration Guidance	84
C.7.1 API Development	84
C.8 Configuration	85
C.8.1 Schema Path Configuration	85
C.8.2 Business Logic Configuration	86
C.9 Testing and Validation	86
C.9.1 Comprehensive Test Suite	86
C.9.2 Validation Examples	87
C.10 Performance Characteristics	87
C.10.1 Initialization Performance	87
C.10.2 Runtime Performance	87
C.10.3 Scalability Considerations	88
C.11 Standards Compliance	88
References	89
Acknowledgments	90

List of Tables

1	BOOST Entity Organization by Thematic Areas	11
2	Entity Definition: Traceable Unit	13
3	Entity Definition: Material Processing	15
4	Entity Definition: Processing History	17
5	Entity Definition: Location History	18
6	Entity Definition: Biometric Identifier	19
7	Entity Definition: Identification Method	20
8	Entity Definition: Organization	21
9	Entity Definition: Certificate	23
10	Entity Definition: CertificationBody	24
11	Entity Definition: CertificationScheme	25
12	Entity Definition: Audit	26
13	Entity Definition: BOOST Operator Entity Validation Schema	27
14	Entity Definition: Material	28
15	Entity Definition: Species Component	29
16	Entity Definition: Supplier	30
17	Entity Definition: Customer	30
18	Entity Definition: Supply Base	31
19	Entity Definition: Supply Base Report	32
20	Entity Definition: Equipment	33
21	Entity Definition: Transaction	35
22	Entity Definition: Transaction Batch	37
23	Entity Definition: Sales Delivery Document	38
24	Entity Definition: Measurement Record	39
25	Entity Definition: Claim	40
26	Entity Definition: Verification Statement	41
27	Entity Definition: BOOST Moisture Content Validation Rules	41
28	Entity Definition: Geographic Data	42
29	Entity Definition: Tracking Point	43
30	Entity Definition: LCFS Pathway	44
31	Entity Definition: LCFS Reporting	45
32	Entity Definition: BioRAM Pathway	46
33	Entity Definition: BioRAM Reporting	47
34	Entity Definition: Product Group	48
35	Entity Definition: Energy Carbon Data	49
36	Entity Definition: Data Reconciliation	50
37	Entity Definition: Mass Balance Account	51

List of Figures

1	BOOST Entity Relationship Diagram Overview with Interactive Navigator Access	74
2	BOOST Thematic Areas and Entity Organization	75

1 Executive Summary

1.1 What BOOST Accomplishes

The Biomass Open Origin Standard for Tracking (BOOST) solves the fundamental challenge of maintaining continuous, verifiable traceability through biomass supply chains—from standing trees to processed biofuels—without losing data integrity at critical transfer points where materials change hands, locations, or physical states.

Core Problem Addressed: Traditional biomass tracking systems break down during material transfers, processing operations, and aggregation points, creating traceability gaps where data continuity is lost or becomes unreliable. This forces businesses to maintain separate, incompatible tracking systems for different regulatory requirements while increasing documentation burden and compliance costs.

BOOST's Solution: A unified data standard built around **TraceableUnits (TRUs)** that maintain their identity throughout the supply chain using progressive identification methods, technology-appropriate data capture, and comprehensive field structures that simultaneously satisfy multiple regulatory frameworks.

1.2 Overarching Design Principles

1.2.1 TraceableUnits (TRUs) as Foundation

The TraceableUnits (TRUs) entity is the central organizing concept of the entire BOOST standard. Every other entity in the 33-entity data model either creates, modifies, measures, or references TraceableUnits (TRUs)s. This TRU-centric approach reflects the physical reality of biomass operations:

- **Harvest Level:** Individual logs, piles, or volume aggregations become initial TRUs with appropriate identification methods
- **Transport Level:** TRUs move through critical tracking points (harvest_site → skid_road → forest_road → mill_entrance) while maintaining identity
- **Processing Level:** Input TRUs are transformed into output TRUs with complete genealogical linkage
- **Compliance Level:** TRU data aggregates automatically generate required documentation for multiple regulatory programs

1.2.2 Physical Reality Mapping

BOOST's structure directly mirrors real-world biomass supply chain operations:

1. **Harvest Operations:** TRUs are created when biomass is harvested, with multi-method identification capturing unique characteristics through technology-appropriate methods
2. **Transportation Flow:** Critical tracking points represent actual physical locations where custody changes hands or materials are aggregated
3. **Processing Stages:** Material transformations (whole logs → chips → pellets → biofuels) create

new TRUs with documented input-output relationships 4. **Multi-Species Reality:** Mixed-species piles and processing batches are represented through SpeciesComponent entities that maintain individual species data within composite TRUs

1.3 Continuous Traceability Framework

Unlike conventional systems that lose traceability when materials are transferred between different tracking systems, BOOST maintains continuous data linkage through:

- **Identity Persistence:** Progressive identification methods that maintain continuity through physical handling and transportation
- **Relationship Preservation:** Parent-child TRU relationships that track material splits, merges, and transformations
- **Automated Reconciliation:** Volume conservation validation and measurement reconciliation across all tracking points

1.4 Value Proposition: Why BOOST Reduces Costs and Complexity

1.4.1 Multi-Method Identification Benefits

BOOST's progressive identification methods provides significant operational advantages over conventional single-method tracking approaches:

Traditional Approach Limitations:

- Manual tagging systems require physical labels that can be lost, damaged, or mislabeled
- Operator data entry creates human error points and increases labor costs
- Separate tracking for different species/grades multiplies documentation burden
- System incompatibilities force duplicate data entry for different regulatory requirements

BOOST Multi-Method Identification Advantages:

- **Technology-Appropriate Deployment:** Progressive identification methods from RFID tags to optical scanning based on operational readiness levels
- **Method Redundancy:** Primary and secondary identification methods ensure continuity when individual methods fail
- **Scalable Implementation:** Start with proven methods (RFID, QR codes) and evolve to advanced methods (optical scanning) as technology matures
- **Cost-Effective Transition:** Incremental technology adoption minimizes up-front investment while maximizing long-term capability

Technology Readiness Approach: BOOST supports identification methods across all Technology Readiness Levels (TRL 1-9), enabling immediate deployment with proven methods while maintaining upgrade paths to emerging technologies as they mature for production use.

1.4.2 Regulatory Compliance Through Strategic Field Combinations

BOOST's field structure enables rapid compliance documentation for multiple regulatory frameworks by strategically combining standard data elements:

California Low Carbon Fuel Standard (LCFS) Compliance:

- Pathway CI calculation: `LCFSPathway.pathwayCI + EnergyCarbon.benchmarkCI`
- Volume reporting: `TraceableUnit.totalVolumeM3 + MaterialProcessing.outputQuantity`
- Feedstock verification: `SupplyBase.supplyBaseType + GeographicData.location validation`

Renewable Fuel Standard (RFS) Compliance:

- Renewable identification: `Material.materialCategory + SpeciesComponent.species`
- Volume tracking: `Transaction.quantityM3` aggregated across supply chain stages
- Quality documentation: `TraceableUnit.qualityGrade + MoistureContent measurements`

EU RED-II Compliance:

- Sustainability certification: `CertificationScheme.schemeType + Certificate validity`
- Land use verification: `GeographicData + SupplyBase.supplyBaseType`
- Greenhouse gas calculations: `EnergyCarbon.ghgEmissions + processing stage data`

Multi-Framework Efficiency: A single BOOST implementation captures all required data fields, eliminating the need for separate tracking systems and enabling automatic generation of compliance documentation for all major regulatory programs.

1.5 Tolerance Standards and Practical Considerations

BOOST recognizes that real-world biomass supply chains cannot achieve perfect volume/mass conservation or species composition accuracy:

Volume Tolerance Standards:

- **CARB Standard:** ± 0.5
- **Processing Tolerances:** Pelletizing operations may have higher acceptable variance due to moisture content changes and material densification

- **De Minimis Thresholds:** Material losses during transport, handling, and processing below 1

Numeric Precision Guidelines:

- **Carbon Intensity Values:** 2 decimal places (e.g., benchmarkCI: 94.17) for regulatory reporting precision
- **Volume Measurements:** 3 decimal places for cubic meter quantities to maintain accuracy across aggregation operations
- **Composition Percentages:** 1 decimal place for species composition within multi-species TRUs

Field Inclusion Rationale: BOOST includes sufficient data complexity to meet regulatory requirements while maintaining practical implementability. Fields are included based on:

- Direct regulatory requirement across one or more major programs (LCFS, RFS, EU-RED)
- Operational necessity for maintaining traceability integrity
- Industry standard practice for biomass chain of custody documentation

This strategic field selection ensures that BOOST implementations serve real-world business purposes by reducing compliance documentation burden while maintaining the data integrity required for sustainability verification and regulatory reporting across multiple jurisdictions.

2 Introduction

The Biomass Open Origin Standard for Tracking (BOOST) data standard defines a comprehensive, interoperable framework for tracking biomass materials through complex supply chains. BOOST enables transparent, verifiable, and consistent data exchange to support sustainability verification, regulatory compliance, and supply chain integrity across the biomass economy.

2.1 Community Development Process

BOOST is developed through the [BOOST W3C Community Group](#) with collaborative input from industry stakeholders, regulatory agencies, and technical experts. The standard implements a `TraceableUnit`-centric model supporting continuous traceability, multi-species composition management, and comprehensive plant part categorization across 33 interconnected entities.

Working Group Members

Chair: Peter Tittmann (Carbon Direct)

Community Group Participants:

- Bodie Cabiyo (Carbon Direct)
- Dani Charles (Veriflux)
- Kylee Durrett (Green Diamond)
- Vanessa Felix (Tule River Economic Development Corporation)
- Marieke Fenton (California Air Resources Board)
- Robert Hambrect (Allotrope Partners)
- Liam Kilroy (Carbon Direct)
- Jeremy Loeb (California Air Resources Board)
- QUINN Matuschek (Life Cycle Associates)
- Carmen Meialua (ARB)
- Andy Miller (Loamist)
- Mahmoud Nabil
- Sarah Oldson (Cascade Resource Consultants, LLC)
- Daniel Sanchez (Carbon Direct)
- Clarke Stevenson (The Watershed Research and Training Center)
- Martin Twer (The Watershed Research & Training Center)

2.2 Current Development Status

Current Version Information

Current Version: v3.4.2-10-gb83f99a - Complete BOOST Documentation Build System with integrated HTML and PDF generation

Recent Enhancements:

- Consolidated documentation architecture with ERD Navigator integration
- Complete Resources & Community section with presentations and meetings
- Enhanced entity cross-references and interactive navigation
- Migrated all ReSpec content to unified Bikeshed system while preserving ERD functionality
- Interactive ERD Navigator with 33 entities across 7 thematic areas

2.3 Participation and Feedback

How to Contribute

- **GitHub Repository:** <https://github.com/carbondirect/BOOST>
- **Issues and Feedback:** Submit via GitHub Issues for technical discussions
- **Community Group:** Join the [BOOST W3C Community Group](#)
- **Interactive Tools:** Use the ERD Navigator to explore and provide schema feedback

Meeting Schedule: Regular working group meetings with notes and action items published via GitHub

2.4 Purpose and Scope

This specification defines the BOOST (Biomass Open Origin Standard for Tracking) data standard for biomass supply chain tracking and verification. The standard provides:

- A unified data model for biomass custody transfers
- Format constraints for serializing chain of custody data
- Integration specifications for certification systems
- Regulatory compliance frameworks for multiple jurisdictions

2.5 Background and Motivation

The development of comprehensive biomass traceability systems addresses critical needs for sustainability verification, regulatory compliance, and supply chain transparency in the growing biomass economy. This standard enables interoperability between reporting systems, registries, and certification bodies.

Funding and Jurisdictional Context

The initial version of this data standard is funded through a grant from the California Department of Conservation, with an initial focus on California as the jurisdictional context while maintaining broad applicability to generalized biomass chain of custody requirements.

2.6 Relationship to Existing Standards

BOOST builds upon and integrates with established standards including:

- ISO 38200 [6] Chain of custody of wood and wood-based products

- SBP Standard 4 [7] and SBP Standard 5 [8] from Sustainable Biomass Partnership
- FSC-STD-40-004 [9] Forest Stewardship Council certification standards
- PEFC-ST-2002 [10] Programme for Endorsement of Forest Certification standards
- California LCFS [11] Low Carbon Fuel Standard requirements
- EU RED II [12] European Union Renewable Energy Directive II

2.7 Community Group Process

This specification was developed through the W3C Community Group process with balanced stakeholder participation including civil society organizations, government agencies, small and large businesses, and independent technical experts. Recruitment and engagement efforts were made to avoid overrepresentation of any single stakeholder group.

W3C Community Group Process

The group operates under the Community and Business Group Process, seeking organizational licensing commitments under the W3C Community Contributor License Agreement (CLA) for all substantive contributions.

3 Use Cases and Requirements

3.1 Primary Use Cases

BOOST addresses the following primary use cases:

3.1.1 California Biomass Supply Chain Tracking

- Forest management organization harvests certified timber
- Processing facilities transform raw materials into biofuels
- Transportation companies maintain chain of custody
- Regulatory agencies verify compliance with LCFS requirements

3.1.2 Multi-Certification Scheme Management

- Single TRU maintains multiple certification claims (FSC, SBP, PEFC)
- Processing operations preserve claim integrity
- Species-specific claims apply to mixed-species materials
- Third-party verification validates claim accuracy

4 Conformance

This section describes the conformance requirements for BOOST implementations. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1] and RFC 8174 [2] when, and only when, they appear in all capitals, as shown here.

RFC 2119 Keyword Interpretation

All RFC 2119 keywords in this specification are formatted using the standard conventions and carry the normative meanings defined in RFC 2119 and RFC 8174.

5 BOOST Traceability System

The BOOST Traceability System implements a comprehensive approach to biomass supply chain tracking that eliminates the traditional weak points where traceability is lost during material transfers and processing operations.

5.1 Key Implementation Features

5.1.1 Flexible Identification Framework

Comprehensive Traceability Approach

`TraceableUnit` entities maintain continuous identification through technology-appropriate methods including trip tickets, RFID tags, QR codes, manual documentation, and emerging biometric signatures. This multi-method approach accommodates existing industry practices while providing flexibility for future technology adoption.

5.1.2 Critical Tracking Points

The system establishes standardized measurement and verification infrastructure with flexible configurations. The standard three-point configuration includes:

- `harvest_site` - Initial TRU creation with appropriate identification method and volume measurement
- `consolidation_point` - Transportation consolidation points with reconciliation validation (formerly `skid_road/forest_road`)
- `mill_entrance` - Processing facility entry points with final verification before transformation

BOOST supports 7 tracking point types total (`harvest_site`, `consolidation_point`, `mill_entrance`, `transfer_station`, `storage_facility`, `quality_control_point`, `mobile_processing_unit`) enabling flexible configurations from 2-point minimum to 5+ point extended setups based on operational complexity.

Critical Tracking Point Requirements

Implementations **MUST** support measurement and verification at all three critical tracking points to ensure complete traceability chain integrity.

5.1.3 Multi-Species Support

Species-specific tracking capabilities enable:

- Individual species identification within mixed material flows
- Species-specific sustainability claim application and inheritance
- Detailed composition tracking with percentage validation
- Regulatory compliance for jurisdiction-specific species requirements

Species Composition Validation

SpeciesComponent entities provide detailed composition tracking with automatic percentage validation to ensure accuracy in multi-species materials.

5.1.4 Complete Processing Chain Documentation

MaterialProcessing entities provide comprehensive audit trails by:

- Linking input TRUs to output TRUs for every transformation
- Tracking plant part changes and transformations during processing
- Validating volume and mass conservation across processing steps
- Supporting split and merge operations with complete genealogy tracking

Processing Chain Requirements

All material transformations **MUST** be documented through **MaterialProcessing** entities that maintain complete input-to-output traceability with validated volume and mass conservation.

6 Data Model Architecture

The BOOST data model provides a comprehensive framework for representing all aspects of biomass supply chain operations. The model consists of 33 intercon-

nected entities that work together to provide complete traceability from forest to final product.

6.1 Key Features

6.1.1 Comprehensive Entity System

Complete Data Model Coverage

- **33 Interconnected Entities** - Complete data model covering all aspects of biomass supply chains across 7 thematic areas
- **JSON-LD Validation** - Structured schemas with business rules and examples
- **Interactive ERD Navigator** - Dynamic exploration with GitHub discussion integration
- **Sustainability Claims** - Species-specific claims with inheritance through processing

6.1.2 Enhanced Geographic Integration

Spatial Data Management

- **GeoJSON Compliance** - Spatial data support for all location-aware entities
- **California Agency Ready** - Administrative boundary and jurisdiction tracking
- **Supply Base Management** - Infrastructure mapping with harvest sites and transportation routes

The data model implements a hub-and-spoke architecture with `TraceableUnit` as the central hub. All other entities **MUST** maintain direct or indirect relationships to TRUs to ensure complete traceability.

6.2 Entity Organization by Thematic Areas

The 33 BOOST entities are organized into 7 thematic areas (see [Table 1](#)):

Table 1: BOOST Entity Organization by Thematic Areas

Thematic Area	Description	Count
Core Traceability	Central tracking infrastructure	5
Organizational Foundation	Business entities and certifications	6
Material & Supply Chain	Material definitions and supply management	7
Transaction Management	Business transaction processing	3
Measurement & Verification	Measurement records and claims	4
Geographic & Tracking	Spatial data and location services	2
Compliance & Reporting	Analytics, reporting, and regulatory compliance	6
Total		33

Entity Relationship Requirements

All entities **MUST** follow the hub-and-spoke design pattern with direct or indirect relationships to `TraceableUnit` entities to maintain complete traceability chain integrity.

6.3 Foreign Key Conventions

All foreign key relationships **MUST** follow the `EntityNameId` pattern:

- Field names **MUST** end with “Id”
- Field names **MUST** reference the target entity name in PascalCase
- Examples: `OrganizationId`, `TraceableUnitId`, `GeographicDataId`

Foreign Key Naming Convention

Implementations **MUST** validate that all foreign key field names follow the `EntityNameId` pattern to ensure consistent referential integrity across the data model.

7 Core Data Entities

7.1 Traceable Unit

Unique biomass tracking unit with BOOST traceability system integration

Entity Relationships

This entity references the following entities:

- `operatorId` → `Operator` (Foreign key to operator)

[View Traceable Unit in ERD Navigator](erd-navigator/index.html?focus=TraceableUnit)

Table 2: Entity Definition: Traceable Unit

Field	Type	Description
createdTimestamp	string (date-time)	When the TRU was created
harvestGeographicDataId	string (pattern)	Harvest location - uses Entity-NameId convention referencing GeographicData
harvesterId	string (pattern)	Foreign key to harvesting organization
identificationConfidence	number (≥ 0 , ≤ 100)	Confidence score for primary identification method (0-100)
identificationMethodId	string (pattern)	Foreign key to Identification-Method entity
isMultiSpecies	boolean	True if contains multiple species
materialTypeId	string (pattern)	Foreign key to Material entity (reference table)
totalVolumeM3	number (≥ 0)	Total volume of the traceable unit in cubic meters
traceableUnitId	string (pattern)	Unique ID for each TRU
uniqueIdentifier	string	Primary identifier - trip ticket ID, RFID tag, QR code, barcode, biometric si...
unitType	enum(4 values)	Type of traceable unit
alternativeFateMetrics	object (structured)	LCA and BECCS analysis metrics for alternative fate assessment
attachedInformation	array<string>	All data linked to this TRU
childTraceableUnitIds	array<string>	For split/merge operations (Phase 2)
currentGeographicDataId	string (pattern)	Current location - uses Entity-NameId convention referencing GeographicData
currentStatus	enum(4 values)	Current status of the TRU (Phase 2)
lastUpdated	string (date-time)	Timestamp of the most recent data update
mediaBreakFlags	array<string>	Points where data continuity was lost (Phase 2)
methodReadinessLevel	integer	Technology Readiness Level (TRL 1-9) for identification method used
operatorId	string (pattern)	Foreign key to operator
parentTraceableUnitId	string (pattern)	Foreign key to parent Traceable-Unit for split/merge operations (Phase 2)
physicalArrangement	object (structured)	Spatial organization affecting collection and decomposition
processingHistory	array<string>	Complete processing chain references (Phase 2)
productClassification	enum(7 values)	Market classification or intended product use
qualityGrade	enum(17 values)	Quality grade with comprehensive standard classification
secondaryIdentifiers	array<object>	Secondary/backup identification methods
sustainabilityCertification	string	FSC, PEFC, etc. claims (Phase 2)

7.2 Material Processing

Processing operations that transform TRUs with plant part tracking

Entity Relationships

This entity references the following entities:

- operatorId → Operator ()

[View Material Processing in ERD Navigator](erd-navigator/index.html?focus=MaterialProcessing)

Table 3: Entity Definition: Material Processing

Field	Type	Description
inputTraceableUnitId	string (pattern)	Unique identifier for the input-traceableunit
inputVolume	number (≥ 0)	Inputvolume field value
outputTraceableUnitId	string (pattern)	Unique identifier for the output-traceableunit
outputVolume	number (≥ 0)	Outputvolume field value
processTimestamp	string (date-time)	Processtimestamp field value
processType	enum(10 values)	Enumerated value for processtype
processingId	string (pattern)	Unique identifier for the processing
acceptableRange	array<number>	Array of [min, max] acceptable loss percentages as decimals
equipmentUsed	string	Equipmentused field value
expectedLossRate	number ($\geq 0, \leq 1$)	Expected material loss rate for this process type as decimal (e.g., 0.02 = 2%).
inputComposition	string	Inputcomposition field value
inputPlantParts	object	Plant parts in input TRU before processing
lossJustification	string	Required explanation when losses exceed acceptable range
operatorId	string (pattern)	Foreign key to the operator who performed this action
outputComposition	string	Outputcomposition field value
outputPlantParts	object	Plant parts in output TRU after processing
plantPartLosses	object	Volume losses by plant part during processing
plantPartTransformations	array<object>	Specific plant part transformations during processing
processStandard	enum(5 values)	Standard or authority defining acceptable loss tolerances for this process
processingGeographicDataId	string (pattern)	Unique identifier for the processinggeographicdata
qualityMetrics	string	Qualitymetrics field value
toleranceValidation	object (structured)	Validation results for process loss tolerance compliance
volumeLoss	number (≥ 0)	Volumeloss field value

7.3 Processing History

Complete timeline of processing events with moisture tracking

Entity Relationships

This entity references the following entities:

- `operatorId` → `Operator` (Foreign key to operator who performed processing)

[View Processing History in ERD Navigator](erd-navigator/index.html?focus=ProcessingHistory)

Table 4: Entity Definition: Processing History

Field	Type	Description
inputTRUIs	array<string>	Array of input TRU IDs (multiple for merge operations)
materialProcessingId	string (pattern)	Foreign key to MaterialProcessing operation
outputTRUIs	array<string>	Array of output TRU IDs (multiple for split operations)
processSequenceNumber	integer	Sequential order of this processing step for the TRU
processingEventType	enum(7 values)	Type of processing event
processingHistoryId	string (pattern)	Unique identifier for the processing history record
timestamp	string (date-time)	When this processing step occurred
traceableUnitId	string (pattern)	Foreign key to TRU this history record belongs to
claimInheritanceData	object (structured)	Sustainability claim inheritance tracking
equipmentUsed	string	Equipment used for this processing step
isCurrentProcessingState	boolean	True if this represents the current processing state
mediaBreakData	object (structured)	Media break detection and recovery information
nextProcessingHistoryIds	array<string>	Array of next processing history record IDs (for split operations)
operatorId	string (pattern)	Foreign key to operator who performed processing
plantPartTransformation	string	Summary of plant part changes during processing
previousProcessingHistoryId	['string', 'null']	Foreign key to previous processing history record (forms chain)
processingDuration	string (pattern)	ISO 8601 duration format for processing time
processingGeographicDataId	string (pattern)	Foreign key to location where processing occurred
qualityChangeDescription	string	Description of quality changes during processing
speciesCompositionChange	enum(5 values)	How species composition changed during processing
volumeChangeRatio	number (≥ 0 , ≤ 2.0)	Ratio of output volume to input volume (1.0 = no change)
volumeConservationData	object (structured)	Volume conservation validation data

7.4 Location History

Historical movement records of TRUs

Entity Relationships

This entity references the following entities:

- operatorId → Operator ()

[View Location History in ERD Navigator](erd-navigator/index.html?focus=LocationHistory)

Table 5: Entity Definition: Location History

Field	Type	Description
geographicDataId	string (pattern)	Foreign key to geographic location data
isCurrentLocation	boolean	Boolean flag indicating iscurrent-location status
locationEventType	enum(5 values)	Enumerated value for location-eventtype
locationHistoryId	string (pattern)	Unique identifier for the locationhistory
timestamp	string (date-time)	Timestamp field value
traceableUnitId	string (pattern)	Foreign key to the traceable unit
distanceTraveled	number (≥0)	Distancetraveled field value
equipmentUsed	string	Equipmentused field value
lastUpdated	string (date-time)	Timestamp of the most recent data update
materialProcessingId	['string', 'null']	Unique identifier for the materialprocessing
notes	string	Notes field value
operatorId	string (pattern)	Foreign key to the operator who performed this action
transportMethod	enum(5 values)	Enumerated value for transport-method
verificationMethods	array<string>	Array of verificationmethods values

7.5 Biometric Identifier

BiometricIdentifier entity in BOOST data model

[View Biometric Identifier in ERD Navigator](erd-navigator/index.html?focus=BiometricIdentifier)

Table 6: Entity Definition: Biometric Identifier

Field	Type	Description
biometricId	string (pattern)	Unique identifier for the biometric
biometricSignature	string	Biometricsignature field value
captureMethod	enum(4 values)	Enumerated value for capturemethod
captureTimestamp	string (date-time)	Capturetimestamp field value
confidenceScore	number ($\geq 0, \leq 100$)	Pattern matching reliability score (0-100)
fallbackRequired	boolean	True if low confidence requires secondary verification
traceableUnitId	string (pattern)	Foreign key to TraceableUnit entity
captureGeographicDataId	string (pattern)	Foreign key to GeographicData entity for capture location
lastUpdated	string (date-time)	Timestamp of the most recent data update
speciesBiometrics	array<string>	Array of speciesbiometrics values
trackingPointId	string (pattern)	Foreign key to TrackingPoint entity where capture occurred
verificationMethod	enum(4 values)	Method used to verify biometric match

7.6 Identification Method

Catalog of available identification methods for TraceableUnit tracking

[View Identification Method in ERD Navigator](erd-navigator/index.html?focus=IdentificationMethod)

Table 7: Entity Definition: Identification Method

Field	Type	Description
averageConfidence	number ($\geq 0, \leq 100$)	Average confidence score achieved by this method
implementationCost	enum(4 values)	Relative cost of implementing this method
locationSuitability	array<string>	Locations where this method can be effectively used
methodId	string (pattern)	Unique identifier for the identification method
methodType	enum(8 values)	Type of identification method
regulatoryAcceptance	array<string>	Regulatory frameworks that accept this method
scalabilityFactor	enum(4 values)	Scale at which this method can be effectively applied
technologyReadinessLevel	integer	Technology Readiness Level (TRL 1-9) of this method
equipmentRequired	array<string>	Equipment needed to implement this method
lastUpdated	string (date-time)	Timestamp of the most recent data update

7.7 Organization

Organization entity with geographic data references and certification management capabilities for Phase 2 BOOST traceability system enhancements

Entity Relationships

This entity references the following entities:

- `organizationId` → Organization (Unique identifier for the organization)

[View Organization in ERD Navigator](erd-navigator/index.html?focus=Organization)

Table 8: Entity Definition: Organization

Field	Type	Description
organizationId	string (pattern)	Unique identifier for the organization
organizationName	string	Legal name of the organization
organizationType	enum(10 values)	Type of organization
airDistrictPermit	string	Air quality management district permit identifier
bioramContractId	string (pattern)	BioRAM competitive procurement contract identifier (external system reference)
bioramEligibilityStatus	enum(4 values)	Current BioRAM program eligibility status
bioramFacilityId	string (pattern)	BioRAM facility identifier for program tracking (external system reference)
bioramRegistrationId	string (pattern)	CEC BioRAM registration identifier for biomass power facilities (external sys...
calFireJurisdiction	string	CAL FIRE unit or jurisdiction for facility area
californiaSRA	boolean	Whether facility operates within California State Responsibility Area
certifications	array<string>	List of certification IDs held by organization
contactEmail	string (email)	Primary contact email address
contactPhone	string (pattern)	Primary contact phone number
establishedDate	string (date)	Date organization was established
facilityCapacity	object (structured)	Facility production or handling capacity for LCFS reporting
fireHazardZoneDesignation	enum(4 values)	CAL FIRE fire hazard severity zone designation for facility location
gridInterconnectionPoint	string	Grid interconnection substation or transmission point
lastUpdated	string (date-time)	Timestamp of the most recent data update
lcfsRegistrationId	string (pattern)	CARB LCFS registration identifier for regulated entities (external system ref..)
operationalAreas	array<string>	List of geographic areas where organization operates
operationalStatus	enum(4 values)	Current operational status of the organization
powerPurchaseAgreementId	string21	Power purchase agreement identifier with utility offtaker
primaryGeographicDataId	string (pattern)	Foreign key to primary operational location
regulatedEntityType	enum(5 values)	LCFS regulated entity classification
taxId	string	Tax identification number
utilityOfftaker	string	Utility company purchasing

7.8 Certificate

Certificate entity representing formal certification records issued by certification bodies

Entity Relationships

This entity references the following entities:

- `certificateId` → `Certificate` (Standard certificate identifier using CERT- pattern)
- `OrganizationId` → `Organization` (Uses `EntityNameId` convention referencing `Organization` receiving the certificate)

[View Certificate in ERD Navigator](erd-navigator/index.html?focus=Certificate)

Table 9: Entity Definition: Certificate

Field	Type	Description
CertificationBodyId	string (pattern)	Uses EntityNameId convention referencing CertificationBody
CertificationSchemeId	string (pattern)	Uses EntityNameId convention referencing CertificationScheme
OrganizationId	string (pattern)	Uses EntityNameId convention referencing Organization receiving the certificate
certificateId	string (pattern)	Standard certificate identifier using CERT- pattern
certificateNumber	string (pattern)	Official certificate number (primary key)
dateOfExpiry	string (date)	Certificate expiry date
dateOfIssue	string (date)	Date of certificate issuance
scopeOfCertification	string	Summary of certification coverage
status	enum(4 values)	Current certificate status
versionNumber	string	Version identifier of the certification standard
VerificationStatementId	string (pattern)	Uses EntityNameId convention referencing VerificationStatement for third-part...
auditSchedule	object (structured)	Scheduled audit information
certificateDocument	string (uri)	Link or reference to certificate document
conditionalRequirements	array<object>	Special conditions or requirements
suspensionHistory	array<object>	History of certificate suspensions
versionYear	integer	Year of the standard's relevant version release

7.9 CertificationBody

Certification Body entity representing independent organizations authorized to issue certificates

[View CertificationBody in ERD Navigator](erd-navigator/index.html?focus=CertificationBody)

Table 10: Entity Definition: CertificationBody

Field	Type	Description
<code>accreditationStatus</code>	enum(5 values)	Current accreditation status
<code>authorizedSchemes</code>	array<string>	List of certification schemes the CB can certify under
<code>cbId</code>	string (pattern)	Unique identifier for the certification body (primary key)
<code>cbName</code>	string	Official name of the certification body
<code>cbType</code>	enum(4 values)	Type or category of certification body
<code>contactInformation</code>	object (structured)	Contact details for the certification body
<code>operationalRegions</code>	array<string>	Geographic regions where CB operates (ISO country codes)
<code>validityPeriod</code>	object (structured)	Period of CB authorization
<code>accreditationBody</code>	string	Organization that accredited this CB
<code>performanceMetrics</code>	object (structured)	CB performance and quality indicators
<code>specializations</code>	array<string>	Specific areas of certification expertise

7.10 CertificationScheme

CertificationScheme entity defining certification standards and requirements with geographic applicability for Phase 2 BOOST traceability system enhancements

[View CertificationScheme in ERD Navigator](erd-navigator/index.html?focus=CertificationScheme)

Table 11: Entity Definition: CertificationScheme

Field	Type	Description
certificationSchemeId	string (pattern)	Unique identifier for the certification scheme
issuingOrganizationId	string (pattern)	Foreign key to organization that issues this scheme
schemeName	string	Official name of the certification scheme
schemeType	enum(6 values)	Type of certification scheme
applicableGeographicAreas	array<string>	Geographic areas where this scheme is applicable
auditRequirements	string	Audit and verification requirements
chainOfCustodyRequirements	string	Chain of custody tracking and documentation requirements
claimTypes	array<string>	Types of claims supported by this scheme
complianceTolerances	object (structured)	Only regulatory compliance tolerances - equipment and process tolerances are ...
documentationRequirements	array<string>	Required documentation and record-keeping
eligibleMaterialTypes	array<string>	Material types eligible for this certification scheme
lastUpdated	string (date-time)	Timestamp of the most recent data update
schemeDescription	string	Detailed description of the certification scheme
schemeStandard	string	Standard or version identifier
validityPeriod	string	Typical validity period for certifications under this scheme
website	string (uri)	Official website for the certification scheme

7.11 Audit

Audit entity in BOOST data model

Entity Relationships

This entity references the following entities:

- auditId → Audit ()
- organizationId → Organization ()

[View Audit in ERD Navigator](erd-navigator/index.html?focus=Audit)

Table 12: Entity Definition: Audit

Field	Type	Description
auditDate	string (date)	Auditdate field value
auditId	string (pattern)	Unique identifier for the audit
auditType	enum(Initial, Surveillance, Transfer)	Enumerated value for audittype
organizationId	string (pattern)	Foreign key to the associated organization
auditGeographicDataId	string (pattern)	Unique identifier for the audit-geographicdata
cbId	string (pattern)	Unique identifier for the cb
findings	string	Findings field value
reportUrl	string (uri)	Reporturl field value

7.12 BOOST Operator Entity Validation Schema

Validation schema for personnel and operator management within the BOOST biomass chain of custody system

Entity Relationships

This entity references the following entities:

- `operatorId` → `Operator` (Unique identifier for the operator (Primary Key))
- `organizationId` → `Organization` (Employing organization - uses `EntityNameId` convention referencing `Organization`)

**[View BOOST Operator Entity Validation Schema in ERD Navigator](erd-navigator/index.html?focus=BOOSTOpera

Table 13: Entity Definition: BOOST Operator Entity Validation Schema

Field	Type	Description
hireDate	string (date)	Date when operator started employment
isActive	boolean	Current employment status - true if actively employed
lastUpdated	string (date-time)	Timestamp of last record modification
operatorId	string (pattern)	Unique identifier for the operator (Primary Key)
operatorName	string	Full name of the operator
operatorType	enum(10 values)	Type/role of operator within the supply chain
organizationId	string (pattern)	Employing organization - uses EntityNameId convention referencing Organization
certifications	array<string>	Array of certifications held by the operator
contactInfo	['string', 'null']	Phone/email contact information
employeeId	['string', 'null']	Internal employee identification number
equipmentAuthorizations	array<string>	Equipment the operator is authorized to operate
skillsQualifications	array<string>	Relevant skills and qualifications
supervisorOperatorId	['string', 'null']	Foreign key reference to direct supervisor operator (optional)

7.13 Material

Material types and specifications

[View Material in ERD Navigator](erd-navigator/index.html?focus=Material)

Table 14: Entity Definition: Material

Field	Type	Description
materialCategory	enum(softwood, hardwood, mixed)	Enumerated value for material-category
materialName	string	Materialname field value
materialTypeId	string (pattern)	Unique identifier for the materialtype
applicablePlantParts	array<string>	Plant parts included in this material type
applicableProcessingTypes	array<string>	Array of applicableprocessingtypes values
carbonStorageRate	string	Carbonstoragerate field value
defaultAssortmentTypes	string	Defaultassortmenttypes field value
density	string	Density field value
energyContent	string	Energycontent field value
excludedPlantParts	array<string>	Plant parts excluded from this material type
lastUpdated	string (date-time)	Timestamp of the most recent data update
plantPartProcessingSpecs	object	Processing specifications by plant part
standardMoistureContent	string	Standardmoisturecontent field value
standardQualityGrades	string	Standardqualitygrades field value
typicalSpecies	array<string>	Array of typicalspecies values

7.14 Species Component

Species composition within TRUs

[View Species Component in ERD Navigator](erd-navigator/index.html?focus=SpeciesComponent)

Table 15: Entity Definition: Species Component

Field	Type	Description
componentId	string (pattern)	Unique identifier for the species component
percentageByVolume	number (≥ 0 , ≤ 100)	Percentage of total TRU volume for this species
species	string	Species name (common or scientific)
traceableUnitId	string (pattern)	Foreign key back reference to TraceableUnit
volumeM3	number (≥ 0)	Volume of this species within the TRU in cubic meters
ageYears	integer	Estimated age in years
carbonStorage	string	CO2 data for this species component
dbhCm	number (≥ 0)	Diameter at breast height in centimeters
defects	array<string>	List of defects or quality issues
harvestTimestamp	string (date-time)	When this species was harvested
harvestingMethod	enum(4 values)	Method used to harvest this species
heightM	number (≥ 0)	Average tree height in meters
lastUpdated	string (date-time)	Timestamp of the most recent data update
moistureContent	number (≥ 0 , ≤ 100)	Moisture content as percentage by weight (0-100%)
plantPartComposition	object	Plant part breakdown within this species component
primaryPlantPart	enum(17 values)	Primary plant part represented by this species component
qualityGrade	enum(17 values)	Quality grade with comprehensive standard classification
scientificName	string	Scientific/Latin name of the species
sourceGeographicDataId	string (pattern)	Foreign key to geographic origin of this species
structuralClassification	enum(5 values)	Functional classification of the primary plant part

7.15 Supplier

Supplier entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `supplierId` → `Supplier` ()

[View Supplier in ERD Navigator](erd-navigator/index.html?focus=Supplier)

Table 16: Entity Definition: Supplier

Field	Type	Description
<code>supplierId</code>	string (pattern)	Unique identifier for the supplier
<code>supplierName</code>	string	Suppliername field value
<code>GeographicDataId</code>	string (pattern)	Supplier location - uses Entity-NameId convention referencing GeographicData
<code>address</code>	string	Address field value
<code>certificateCode</code>	string	Certificatecode field value
<code>claim</code>	string	Claim field value
<code>supplierType</code>	string	Supplierstype field value

7.16 Customer

Customer entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `customerId` → `Customer` (Unique identifier for the customer)

[View Customer in ERD Navigator](erd-navigator/index.html?focus=Customer)

Table 17: Entity Definition: Customer

Field	Type	Description
<code>customerId</code>	string (pattern)	Unique identifier for the customer
<code>customerName</code>	string	Customername field value
<code>GeographicDataId</code>	string (pattern)	Customer location - uses Entity-NameId convention referencing GeographicData
<code>address</code>	string	Address field value

7.17 Supply Base

SupplyBase entity in BOOST data model. Infrastructure data (roads) and entity relationships (equipment, TRUs) are maintained via proper foreign keys rather than redundant arrays, following normalization principles.

Entity Relationships

This entity references the following entities:

- `OrganizationId` → `Organization` (Managing organization - uses `EntityNameId` convention referencing `Organization`)

[View Supply Base in ERD Navigator](erd-navigator/index.html?focus=SupplyBase)

Table 18: Entity Definition: Supply Base

Field	Type	Description
<code>OrganizationId</code>	string (pattern)	Managing organization - uses <code>EntityNameId</code> convention referencing <code>Organization</code>
<code>description</code>	string	Description field value
<code>supplyBaseId</code>	string (pattern)	Unique identifier for the supply-base
<code>supplyBaseName</code>	string	Supplybasename field value
<code>GeographicDataId</code>	string (pattern)	Supply base location - uses <code>EntityNameId</code> convention referencing <code>GeographicData</code>
<code>harvestSites</code>	array<string>	Array of harvestsites values
<code>speciesAvailable</code>	array<string>	Array of speciesavailable values

7.18 Supply Base Report

SupplyBaseReport entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `organizationId` → `Organization` (Foreign key to `Organization` entity that prepared this report)

[View Supply Base Report in ERD Navigator](erd-navigator/index.html?focus=SupplyBaseReport)

Table 19: Entity Definition: Supply Base Report

Field	Type	Description
organizationId	string (pattern)	Foreign key to Organization entity that prepared this report
preparationDate	string (date)	Preparationdate field value
sbrId	string (pattern)	Unique identifier for the sbr
publicationUrl	string (uri)	Publicationurl field value
reportGeographicDataId	string (pattern)	Foreign key to GeographicData entity for report coverage area
sourcingPractices	string	Sourcingpractices field value
supplyBaseIds	array<string>	Array of SupplyBase IDs that this report covers
supplyBaseSummary	string	Supplybasesummary field value
sustainabilityMeasures	string	Sustainabilitymeasures field value

7.19 Equipment

Equipment entity representing forestry machinery and equipment used in biomass harvesting and processing operations

Entity Relationships

This entity references the following entities:

- equipmentId → Equipment (Unique identifier for the equipment)
- organizationId → Organization (Foreign key to owning organization)

[View Equipment in ERD Navigator](erd-navigator/index.html?focus=Equipment)

Table 20: Entity Definition: Equipment

Field	Type	Description
equipmentId	string (pattern)	Unique identifier for the equipment
equipmentName	string	Descriptive name of the equipment
equipmentType	enum(12 values)	Type of forestry equipment
operationalStatus	enum(5 values)	Current operational status of the equipment
organizationId	string (pattern)	Foreign key to owning organization
acquisitionCost	number (≥ 0)	Equipment acquisition cost in USD
acquisitionDate	string (date)	Date equipment was acquired by organization
assignedTrackingPointId	string (pattern)	Foreign key to current location/assignment
certifications	array<string>	Equipment certifications (safety, emissions, etc.)
currentOperatorId	string (pattern)	Foreign key to current operator (if assigned)
insuranceInfo	object (structured)	Equipment insurance information
lastUpdated	string (date-time)	Timestamp of the most recent data update
maintenanceSchedule	object (structured)	Maintenance schedule information
manufacturer	string	Equipment manufacturer
model	string	Equipment model designation
notes	string	Additional notes or comments about the equipment
serialNumber	string	Manufacturer serial number
specifications	object (structured)	Technical specifications for the equipment
yearManufactured	integer	Year the equipment was manufactured

7.20 Transaction

Transaction entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `transactionId` → `Transaction` (Unique identifier for the business transaction)
- `OrganizationId` → `Organization` (Primary organization involved in transaction (seller/supplier))
- `CustomerId` → `Customer` (Customer organization (buyer) - uses `EntityNameId` convention referencing `Customer` entity)

[View Transaction in ERD Navigator](erd-navigator/index.html?focus=Transaction)

Table 21: Entity Definition: Transaction

Field	Type	Description
CustomerId	string (pattern)	Customer organization (buyer) - uses EntityNameId convention referencing Cust...
OrganizationId	string (pattern)	Primary organization involved in transaction (seller/supplier)
contractCurrency	enum(9 values)	Currency code for contract value
contractValue	number (≥0, ≤999999999.99)	Total monetary value of the transaction
transactionDate	string (date)	Date of business agreement
transactionId	string (pattern)	Unique identifier for the business transaction
transactionStatus	enum(6 values)	Current status of business transaction
BioramPathwayId	string (pattern)	BioRAM pathway identifier for biomass power generation - uses EntityNameId co...
BrokerOrganizationId	['string', 'null']	Optional intermediary broker organization - uses EntityNameId convention refe...
GeographicDataId	string (pattern)	Primary transaction location - uses EntityNameId convention referencing Geogr...
LcfsPathwayId	string (pattern)	CARB-certified pathway identifier for LCFS compliance - uses EntityNameId con...
SalesDeliveryDocumentId	string (pattern)	Foreign key to sales/delivery documentation - uses EntityNameId convention re...
attestationDate	string (date)	Date of BioRAM compliance attestation
attestationSignatory	string	Name and title of person attesting to BioRAM compliance
biomassVolume	number (≥0)	Volume of biomass fuel in transaction for BioRAM reporting
biomassVolumeUnit	enum(bone_dry_tonnes, green_tonnes, cubic_yards)	Unit of measurement for biomass volume
bioramCertificationId	string	BioRAM compliance certification identifier for transaction
bioramEligible	boolean	Whether transaction meets BioRAM program eligibility requirements
calFirePermitNumber	string	CAL FIRE permit number for harvest or treatment activities
complianceRequirements	array<string>	Regulatory compliance requirements for transaction
contractSignedDate	['string', 'null']	Date when contract was executed
contractTerms	enum(8 values)	Incoterms delivery conditions
expectedDeliveryDate	['string', 'null']	Expected completion/delivery date
fhszVerificationSource	string	Source of fire hazard zone verification (e.g.,

7.21 Transaction Batch

TransactionBatch entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `transactionId` → `Transaction` (Foreign key to parent business transaction)
- `claimId` → `Claim` (Foreign key to primary sustainability claim)

[View Transaction Batch in ERD Navigator](erd-navigator/index.html?focus=TransactionBatch)

Table 22: Entity Definition: Transaction Batch

Field	Type	Description
batchStatus	enum(6 values)	Current status of the physical batch
quantity	number (≥ 0)	Physical quantity of material in this batch
quantityUnit	enum(7 values)	Unit of measurement for quantity
traceableUnitIds	array<string>	Array of TRU IDs included in this batch
transactionBatchId	string (pattern)	Unique identifier for the physical material batch
transactionId	string (pattern)	Foreign key to parent business transaction
additionalClaimIds	array<string>	Array of secondary claim IDs
batchCreatedDate	string (date-time)	When the batch was prepared/created
certificationValidation	object (structured)	Certification and compliance validation data
claimId	['string', 'null']	Foreign key to primary sustainability claim
deliveryDate	['string', 'null']	Actual delivery timestamp
deliveryGeographicDataId	['string', 'null']	Foreign key to delivery location
lastUpdated	string (date-time)	Timestamp of last modification
measurementRecordIds	array<string>	Array of measurement record IDs
mediaBreakDetected	boolean	Flag indicating if traceability continuity was broken
plantPartComposition	object	Plant part composition breakdown
processingHistoryIds	array<string>	Array of processing history record IDs
productionBatchId	['string', 'null']	Foreign key to source production batch
qualityGrade	enum(17 values)	Quality grade with comprehensive standard classification
qualityMetrics	object (structured)	Detailed quality assessment metrics
reconciliationStatus	enum(4 values)	Reconciliation status with standard values
speciesComposition	array<object>	Species breakdown with percentages
trackingHistory	string	Complete location trail summary
transportationData	object (structured)	Transportation and logistics information

7.22 Sales Delivery Document

SalesDeliveryDocument entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `transactionId` → Transaction (Foreign key to Transaction entity)

[View Sales Delivery Document in ERD Navigator](erd-navigator/index.html?focus=SalesDeliveryDocument)

Table 23: Entity Definition: Sales Delivery Document

Field	Type	Description
<code>buyerName</code>	string	Buyername field value
<code>dateIssued</code>	string (date)	Dateissued field value
<code>documentId</code>	string (pattern)	Unique identifier for the document
<code>productDescription</code>	string	Productdescription field value
<code>quantity</code>	number	Quantity field value
<code>sellerName</code>	string	Sellername field value
<code>buyerAddress</code>	string	Buyeraddress field value
<code>certificateCode</code>	string	Certificatecode field value
<code>sellerAddress</code>	string	Selleraddress field value
<code>transactionId</code>	string (pattern)	Foreign key to Transaction entity
<code>transportReference</code>	string	Transportreference field value

7.23 Measurement Record

Quality measurements and dimensional data

Entity Relationships

This entity references the following entities:

- `operatorId` → Operator ()

[View Measurement Record in ERD Navigator](erd-navigator/index.html?focus=MeasurementRecord)

Table 24: Entity Definition: Measurement Record

Field	Type	Description
measurementMethod	enum(18 values)	Method used for measurement with comprehensive standard values
measurementTimestamp	string (date-time)	Measurementtimestamp field value
recordId	string (pattern)	Unique identifier for the record
traceableUnitId	string (pattern)	Foreign key to the traceable unit
accuracyValidation	object (structured)	Validation range based on equipment accuracy specifications
calibrationStandard	enum(5 values)	Reference standard used for equipment calibration and accuracy specification
expectedAccuracy	number ($\geq 0, \leq 1$)	Expected accuracy of the measurement method as decimal (e.g., 0.02 = $\pm 2\%$ acc...)
lastUpdated	string (date-time)	Timestamp of the most recent data update
measuredDiameter	number (≥ 0)	Measureddiameter field value
measuredLength	number (≥ 0)	Measuredlength field value
measuredVolume	number (≥ 0)	Measuredvolume field value
measurementGeographicDataId	string (pattern)	Unique identifier for the measurementgeographicdata
moistureAccuracy	number (≥ 0)	Estimated accuracy of moisture measurement (\pm percentage points)
moistureContent	number ($\geq 0, \leq 100$)	Moisture content as percentage by weight (0-100%)
moistureMethod	enum(8 values)	Method used to determine moisture content
moistureStandard	enum(5 values)	Standard procedure followed for moisture measurement
operatorId	string (pattern)	Foreign key to the operator who performed this action
speciesMeasurements	array<string>	Array of speciesmeasurements values
trackingPointId	string (pattern)	Unique identifier for the trackingpoint

7.24 Claim

Claim entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `claimId` → Claim ()

[View Claim in ERD Navigator](erd-navigator/index.html?focus=Claim)

Table 25: Entity Definition: Claim

Field	Type	Description
<code>TraceableUnitId</code>	string (pattern)	Referenced traceable unit - uses EntityNameId convention referencing Traceabl...
<code>claimId</code>	string (pattern)	Foreign key to the associated claim
<code>claimType</code>	enum(9 values)	Enumerated value for claimtype
<code>statement</code>	string	Statement field value
<code>validated</code>	boolean	Boolean flag indicating validated status
<code>CertificationSchemeId</code>	string (pattern)	Certification scheme - uses EntityNameId convention referencing Certification...
<code>applicableSpecies</code>	array<string>	Array of applicableSpecies values
<code>claimExpiry</code>	string (date-time)	Claimexpiry field value
<code>claimPercentage</code>	number (≥0, ≤100)	Claimpercentage field value
<code>claimScope</code>	enum(4 values)	Enumerated value for claimscope
<code>evidenceDocumentId</code>	string	Foreign key to Evidencedocument entity
<code>inheritedFromTRU</code>	array<string>	Array of inheritedfromtru values
<code>lastUpdated</code>	string (date-time)	Timestamp of the most recent data update
<code>validatedBy</code>	string	Validatedby field value
<code>validationDate</code>	string (date-time)	Validationdate field value

7.25 Verification Statement

VerificationStatement entity in BOOST data model

[View Verification Statement in ERD Navigator](erd-navigator/index.html?focus=VerificationStatement)

Table 26: Entity Definition: Verification Statement

Field	Type	Description
issuingBody	string (pattern)	Foreign key to CertificationBody entity
statementId	string (pattern)	Unique identifier for the statement
verificationDate	string (date)	Verificationdate field value
scope	string	Scope field value
transactionBatchId	string (pattern)	Foreign key to TransactionBatch entity

7.26 BOOST Moisture Content Validation Rules

Comprehensive validation rules and business logic for moisture content tracking across the BOOST data standard

[View BOOST Moisture Content Validation Rules in ERD Navigator](erd-navigator/index.html?focus=BOOSTMoistureContentValidationRules)

Table 27: Entity Definition: BOOST Moisture Content Validation Rules

Field	Type	Description
measurementMethod	enum(18 values)	Method used for measurement with comprehensive standard values
moistureContent	number ($\geq 0, \leq 100$)	Moisture content as percentage by weight (0-100%)
moistureContentId	string (pattern)	Unique identifier for moisture content record
qualityAssurance	unknown	Qualityassurance field value

7.27 Geographic Data

GeographicData entity in BOOST data model

[View Geographic Data in ERD Navigator](erd-navigator/index.html?focus=GeographicData)

Table 28: Entity Definition: Geographic Data

Field	Type	Description
<code>dataType</code>	enum(7 values)	Type of geographic data
<code>description</code>	string	Human-readable description of the geographic area
<code>geoJsonData</code>	object (structured)	Valid GeoJSON object (Point, Polygon, LineString, etc.)
<code>geographicDataId</code>	string (pattern)	Unique identifier for the geographic data
<code>accessRestrictions</code>	string	Any access restrictions or special conditions
<code>accuracy</code>	number (≥ 0)	GPS accuracy in meters
<code>administrativeRegion</code>	string	Administrative region or jurisdiction
<code>coordinateSystem</code>	string	Coordinate reference system (e.g., WGS84, UTM Zone 10N)
<code>elevationM</code>	number	Elevation in meters above sea level
<code>lastUpdated</code>	string (date-time)	Timestamp of the most recent data update

7.28 Tracking Point

TrackingPoint entity in BOOST data model

Entity Relationships

This entity references the following entities:

- `operatorId` → `Operator` (Foreign key to Operator entity)

[View Tracking Point in ERD Navigator](erd-navigator/index.html?focus=TrackingPoint)

Table 29: Entity Definition: Tracking Point

Field	Type	Description
equipmentUsed	string	Equipment deployed at this tracking point
establishedTimestamp	string (date-time)	Establishedtimestamp field value
geographicDataId	string (pattern)	Foreign key to GeographicData entity
pointType	enum(7 values)	Type of tracking point - flexible configuration based on operational needs
trackingPointId	string (pattern)	Unique identifier for the tracking point
configurationRole	enum(4 values)	Role of this point in the overall tracking configuration
coordinatePrecision	number (≥ 1 , ≤ 100)	Required coordinate precision in meters (e.g., 5 = ± 5 m accuracy)
operatorId	string (pattern)	Foreign key to Operator entity

7.29 LCFS Pathway

CARB-certified fuel pathway for LCFS compliance with carbon intensity and regulatory attributes

[View LCFS Pathway in ERD Navigator](erd-navigator/index.html?focus=LCFSPathway)

Table 30: Entity Definition: LCFS Pathway

Field	Type	Description
caGreetVersion	string (pattern)	CA-GREET model version used for pathway certification
carbonIntensity	number (≥ 0 , ≤ 200)	Certified carbon intensity in gCO ₂ e/MJ
certificationDate	string (date)	CARB pathway certification date
energyEconomyRatio	number (≥ 0.5 , ≤ 3.0)	Energy economy ratio multiplier for credit calculation
facilityLocation	string	Production facility location (city, state or geographic region)
feedstockCategory	enum(13 values)	Primary feedstock type for pathway
fuelProduct	enum(8 values)	Final fuel product produced
pathwayId	string (pattern)	CARB-assigned pathway identifier
pathwayType	enum(Lookup_Table, Tier_1, Tier_2)	CARB pathway certification tier
verificationStatus	enum(4 values)	Current CARB verification status
expirationDate	string (date)	Pathway certification expiration date
facilityCapacity	number (≥ 0)	Annual production capacity in gallons
geographicScope	enum(4 values)	Geographic applicability of pathway
lastUpdated	string (date-time)	Timestamp of most recent pathway data update
processDescription	string	Brief description of production process

7.30 LCFS Reporting

Quarterly LCFS compliance report for regulated entities with credit calculations and submission tracking

[View LCFS Reporting in ERD Navigator](erd-navigator/index.html?focus=LCFSReporting)

Table 31: Entity Definition: LCFS Reporting

Field	Type	Description
<code>complianceStatus</code>	enum(4 values)	Overall compliance status for the reporting period
<code>netPosition</code>	number	Net credit/deficit position (credits - deficits)
<code>regulatedEntityId</code>	string (pattern)	Reference to regulated Organization entity
<code>reportingId</code>	string (pattern)	Unique identifier for the quarterly report
<code>reportingPeriod</code>	string (pattern)	Reporting quarter in YYYY-QN format
<code>totalCreditsGenerated</code>	number (≥ 0)	Total LCFS credits generated in the reporting period
<code>totalDeficitsIncurred</code>	number (≥ 0)	Total LCFS deficits incurred in the reporting period
<code>totalFuelVolume</code>	number (≥ 0)	Total fuel volume reported in gallons
<code>VerificationStatementId</code>	string (pattern)	Uses EntityNameId convention referencing VerificationStatement for third-part...
<code>calculationParameters</code>	object (structured)	Calculation parameters used for credit computation
<code>complianceMetrics</code>	object (structured)	Additional compliance and environmental impact metrics
<code>lastUpdated</code>	string (date-time)	Timestamp of most recent report update
<code>pathwaySummary</code>	array<object>	Summary of activity by LCFS pathway
<code>reportingDeadline</code>	string (date)	CARB deadline for report submission
<code>submissionDate</code>	string (date-time)	Date and time report was submitted to CARB
<code>transactionIds</code>	array<string>	Array of Transaction entity IDs included in this report
<code>verificationDate</code>	string (date-time)	Date of third-party verification completion
<code>verificationRequired</code>	boolean	Whether third-party verification is required for this entity

7.31 BioRAM Pathway

California BioRAM program pathway for biomass power generation with fuel classification and efficiency attributes

[View BioRAM Pathway in ERD Navigator](erd-navigator/index.html?focus=BioRAMPathway)

Table 32: Entity Definition: BioRAM Pathway

Field	Type	Description
carbonIntensity	number (≥ 0 , ≤ 50)	Carbon intensity in gCO _{2e} /MJ for biomass fuel
certificationDate	string (date)	CEC BioRAM pathway certification date
efficiencyStandard	number (≥ 0.2 , ≤ 0.6)	Minimum efficiency requirement for BioRAM eligibility (fraction)
eligibilityStatus	enum(4 values)	Current CEC eligibility status
fuelType	enum(7 values)	BioRAM eligible fuel type classification
pathwayId	string (pattern)	BioRAM pathway identifier
targetFacilityType	enum(biomass_power_plant, biogas_facility, combined_heat_power)	Type of facility this pathway applies to
cecVersion	string (pattern)	CEC BioRAM program version used for certification
expirationDate	string (date)	Pathway certification expiration date
fireHazardZoneEligibility	array<string>	Eligible CAL FIRE hazard severity zones
geographicScope	enum(California_SRA, California_Statewide, Western_States)	Geographic applicability within State Responsibility Areas
haulDistanceLimit	number (≥ 0 , ≤ 200)	Maximum economical haul distance in miles
lastUpdated	string (date-time)	Timestamp of most recent pathway data update
seasonalAvailability	object (structured)	Seasonal fuel availability characteristics
sourceRegion	string	Primary source region or forest management unit

7.32 BioRAM Reporting

Quarterly BioRAM compliance report for biomass power generation facilities with fuel sourcing and efficiency tracking

[View BioRAM Reporting in ERD Navigator](erd-navigator/index.html?focus=BioRAMReporting)

Table 33: Entity Definition: BioRAM Reporting

Field	Type	Description
complianceStatus	enum(4 values)	Overall BioRAM compliance status for the reporting period
facilityEntityId	string (pattern)	Reference to biomass facility Organization entity
overallEfficiency	number ($\geq 0.15, \leq 0.6$)	Overall facility efficiency (net electrical output / fuel energy input)
reportingId	string (pattern)	Unique identifier for the quarterly BioRAM report
reportingPeriod	string (pattern)	Reporting quarter in YYYY-QN format
totalBiomassVolume	number (≥ 0)	Total biomass fuel consumed in bone dry tonnes
totalEnergyGenerated	number (≥ 0)	Total electrical energy generated in MWh
VerificationStatementId	string (pattern)	Uses EntityNameId convention referencing VerificationStatement for independen...
bioramContractId	string (pattern)	BioRAM contract identifier from competitive procurement (external system refe...
efficiencyTarget	number ($\geq 0.2, \leq 0.55$)	Contract efficiency target for BioRAM compliance
environmentalImpact	object (structured)	Environmental and fire hazard reduction benefits
financialSummary	object (structured)	Financial performance and cost metrics
fuelSourcingSummary	object (structured)	Summary of fuel sourcing compliance metrics
lastUpdated	string (date-time)	Timestamp of most recent report update
performanceMetrics	object (structured)	Facility performance and grid contribution metrics
reportingDeadline	string (date)	CEC deadline for BioRAM report submission
submissionDate	string (date-time)	Date and time report was submitted to CEC
transactionIds	array<string>	Array of Transaction entity IDs for fuel procurement included in this report
verificationDate	string (date-time)	Date of independent verification completion
verificationRequired	boolean	Whether third-party verification is required for this facility

7.33 Product Group

ProductGroup entity in BOOST data model

[View Product Group in ERD Navigator](erd-navigator/index.html?focus=ProductGroup)

Table 34: Entity Definition: Product Group

Field	Type	Description
description	string	Description field value
productCategory	enum(solid_biomass, liquid_biofuel, biogas)	Enumerated value for product-category
productGroupId	string (pattern)	Unique identifier for the product-group
productGroupName	string	Productgroupname field value
certificationRequirements	array<string>	Array of certificationrequirements values
classification	string	Classification field value
lastUpdated	string (date-time)	Timestamp of the most recent data update
qualityStandards	array<string>	Array of qualitystandards values
regulatoryClassification	string	Regulatoryclassification field value
relatedMaterials	array<object>	Array of relatedmaterials values
typicalUses	array<string>	Array of typicaluses values

7.34 Energy Carbon Data

EnergyCarbonData entity in BOOST data model

[View Energy Carbon Data in ERD Navigator](erd-navigator/index.html?focus=EnergyCarbonData)

Table 35: Entity Definition: Energy Carbon Data

Field	Type	Description
dataType	enum(7 values)	Enumerated value for datatype
energyCarbonDataId	string (pattern)	Unique identifier for the energy-carbondata
source	enum(4 values)	Enumerated value for source
unit	enum(8 values)	Enumerated value for unit
value	number	Value field value
caGreetVersion	string (pattern)	CA-GREET model version used for calculation
energyEconomyRatio	number ($\geq 0.5, \leq 3.0$)	Energy economy ratio for LCFS credit calculation
humidityConditions	number	Humidityconditions field value
lcfsPathwayType	enum(4 values)	LCFS pathway tier classification
lifeCycleStage	enum(6 values)	Lifecycle stage for carbon intensity data
measurementGeographicDataId	string (pattern)	Unique identifier for the measurementgeographicdata
measurementMethod	enum(18 values)	Method used for measurement with comprehensive standard values
measurementRecordId	string (pattern)	Foreign key to the measurement record
measurementTimestamp	string (date-time)	Measurementtimestamp field value
qualityAssurance	string	Qualityassurance field value
regulatoryBenchmark	number	CARB regulatory benchmark for comparison (gCO ₂ e/MJ)
temperatureConditions	number	Temperatureconditions field value
traceableUnitId	string (pattern)	Foreign key to the traceable unit

7.35 Data Reconciliation

DataReconciliation entity in BOOST data model

Entity Relationships

This entity references the following entities:

- transactionId → Transaction (Foreign key to Transaction entity being reconciled)

[View Data Reconciliation in ERD Navigator](erd-navigator/index.html?focus=DataReconciliation)

Table 36: Entity Definition: Data Reconciliation

Field	Type	Description
discrepancy	number	Discrepancy field value
forestMeasurement	number (≥ 0)	Forestmeasurement field value
millMeasurement	number (≥ 0)	Millmeasurement field value
reconciliationDate	string (date-time)	Reconciliationdate field value
reconciliationId	string (pattern)	Unique identifier for the reconciliation
reconciliationStatus	enum(4 values)	Reconciliation status with standard values
traceableUnitId	string (pattern)	Foreign key to TraceableUnit entity being reconciled
discrepancyReason	string	Discrepancyreason field value
lastUpdated	string (date-time)	Timestamp of the most recent data update
reconciliationOperator	string	Reconciliationoperator field value
resolutionNotes	string	Resolutionnotes field value
speciesDiscrepancies	array<string>	Array of speciesdiscrepancies values
tolerancePercentage	number ($\geq 0, \leq 100$)	Tolerancepercentage field value
transactionId	string (pattern)	Foreign key to Transaction entity being reconciled

7.36 Mass Balance Account

MassBalanceAccount entity in BOOST data model

Entity Relationships

This entity references the following entities:

- organizationId → Organization ()

[View Mass Balance Account in ERD Navigator](erd-navigator/index.html?focus=MassBalanceAccount)

Table 37: Entity Definition: Mass Balance Account

Field	Type	Description
accountId	string (pattern)	Unique identifier for the account
currentBalance	number	Currentbalance field value
organizationId	string (pattern)	Foreign key to the associated organization
productGroupId	string (pattern)	Unique identifier for the product-group
balancingPeriod	string	Balancingperiod field value
conversionFactors	number	Conversionfactors field value
periodInputs	number	Periodinputs field value
periodOutputs	number	Periodoutputs field value

8 Plant Part Categorization System

8.1 Introduction and Regulatory Context

The BOOST plant part categorization framework addresses critical regulatory and operational requirements across multiple jurisdictions and applications. This system provides the taxonomic foundation for distinguishing product classifications from physical arrangements, enabling sophisticated supply chain optimization and regulatory compliance.

8.1.1 Regulatory Drivers

California Department of Food and Agriculture (CDFA) Requirements

- Agricultural biomass classification for food vs. fuel categorization
- Privacy protections for farmer data in agricultural residue tracking
- Integration with existing CDFA biomass certification systems
- Support for agricultural waste stream optimization programs

Low Carbon Fuel Standard (LCFS) Compliance

- Feedstock categorization requirements for carbon intensity calculations
- Plant part composition tracking for pathway verification
- Biogenic carbon accounting across different material components
- Alternative fate assessment support for BECCS applications

Forest Stewardship Council (FSC) Integration

- Chain of custody tracking through plant part transformations

- Controlled wood verification for different plant components
- Multi-species composition documentation requirements
- Value recovery optimization across plant part classifications

8.1.2 Conceptual Framework

The BOOST system distinguishes between two fundamental attributes:

Product Classification vs. Physical Arrangement

- **Product Classification:** Market destination or intended use (sawlog, pulpwood, biomass, chips)
- **Physical Arrangement:** Spatial organization affecting collection and decomposition (scattered, piled, windrow, stacked)

This distinction enables sophisticated LCA and BECCS analysis by capturing both economic intent and operational reality.

8.2 Standardized Plant Parts Taxonomy

Implementations **MUST** support the following 17 standardized plant parts:

- **trunk** - Main stem/bole of tree
- **heartwood** - Inner, non-living wood
- **sapwood** - Outer, living wood
- **bark** - Protective outer layer
- **branches** - Secondary stems
- **leaves** - Photosynthetic organs
- **seeds** - Reproductive structures
- **roots** - Below-ground structures
- **twigs** - Small branches
- **cones** - Seed-bearing structures
- **needles** - Coniferous leaves
- **foliage** - All leaf matter
- **crown** - Above-ground branching structure
- **stump** - Remaining base after felling
- **chips** - Mechanically processed fragments
- **sawdust** - Fine processing residue

- **pellets** - Densified processed material

Plant Part Classification Requirements

All `TraceableUnit` entities **MUST** specify plant part classification using this standardized taxonomy to ensure consistent categorization across implementations.

8.3 Physical Arrangement Framework

The BOOST system captures spatial organization of biomass materials to support collection planning and LCA analysis. Physical arrangement significantly affects both operational efficiency and environmental impact assessment.

8.3.1 Arrangement Categories

Scattered Arrangement

- Crowns and branches distributed across forest floor after harvesting
- Lower collection efficiency (typically 65-75%)
- Higher decomposition rates due to ground contact and weather exposure
- Alternative fate: natural decomposition or wildfire fuel

Centralized Piles

- Material gathered into specific collection points for efficiency
- High collection efficiency (typically 90-95%)
- Moderate decomposition rates depending on pile construction
- Optimized for mechanical loading and transport operations

Windrow Configuration

- Linear arrangements following equipment access patterns
- Collection efficiency 80-90% with mechanical systems
- Balanced decomposition rates with partial ground contact
- Enables efficient forwarding and chipping operations

Stacked Arrangements

- Organized vertical stacking for drying and storage
- Highest collection efficiency (95%+) with quality preservation
- Lowest decomposition rates when properly ventilated
- Premium applications requiring controlled moisture content

8.3.2 LCA and BECCS Integration

Alternative Fate Modeling

- Baseline scenario assessment (decomposition, wildfire, prescribed burning)
- Arrangement-specific decomposition rates for carbon accounting
- Emissions avoided calculations based on collection vs. baseline
- Soil carbon impact assessment from ground contact patterns

Collection Efficiency Factors

- Energy requirements for different arrangement patterns
- Equipment accessibility and operational constraints
- Volume recovery rates by arrangement and terrain conditions
- Economic optimization through arrangement planning

Physical Arrangement Requirements

TraceableUnit entities **MAY** include physical arrangement data to support LCA analysis and collection optimization. When included, arrangement data **MUST** use standardized arrangement types and provide collection efficiency factors.

9 Schema Definitions

9.1 JSON Schema Format

All BOOST entity definitions **MUST** be provided as JSON Schema Draft-07 compliant schemas with the following **REQUIRED** structure:

JSON Example: Entity Schema Structure

```
{
  "schema": {
    "$schema": "http://json-schema.org/draft-07/schema#",
    "$id":
    ↪ "https://github.com/carbondirect/BOOST/schemas/entity-name",
    "title": "Entity Name",
    "type": "object",
    "properties": { },
    "required": [ ]
  }
}
```

9.2 Business Logic Validation

Implementations **MUST** validate entities against 8 categories of business rules:

1. **Volume/Mass Conservation** - Physical conservation laws
2. **Temporal Logic** - Date consistency validation
3. **Geographic Logic** - Spatial relationship validation
4. **Species Composition** - Percentage validation (sum to 100%)
5. **Certification Logic** - Chain of custody validation
6. **Regulatory Compliance** - Jurisdiction-specific rules
7. **Economic Logic** - Price and payment validation
8. **Quality Assurance** - Material quality constraints

Validation Requirements

Conforming implementations **MUST** validate data against BOOST JSON schemas and implement all required business logic validation rules for their conformance level.

10 Serialization and Exchange

10.1 JSON-LD as Primary Format

BOOST data **MUST** be serializable to JSON-LD format with:

- Valid `@context` referencing BOOST context definition
- Entity `@type` declarations matching schema names
- Unique `@id` values for all entities

JSON-LD Requirements

All BOOST data exchanges **MUST** use valid JSON-LD 1.1 format with appropriate context definitions and semantic annotations.

11 JSON-LD Context and Semantic Web Integration

BOOST implements JSON-LD (JSON for Linking Data) as its primary serialization format, enabling semantic web compatibility, data linking, and machine-readable context definitions. This section explains the JSON-LD context structure, semantic annotations, and integration with existing ontologies.

11.1 JSON-LD Overview

JSON-LD extends standard JSON with semantic web capabilities through:

- **@context**: Defines mappings between JSON properties and RDF vocabularies
- **@id**: Provides unique identifiers for entities (IRIs)
- **@type**: Specifies the semantic type of an entity
- **@vocab**: Sets a default vocabulary for properties
- **Linked Data**: Enables connections between distributed datasets

11.2 BOOST Context Definition

The BOOST JSON-LD context maps entity properties to established vocabularies:

JSON Example: BOOST Core Context

```
{
  "@context": {
    "schema": "http://schema.org/",
    "prov": "http://www.w3.org/ns/prov#",
    "gs1": "https://gs1.org/voc/",
    "biomass": "http://example.org/biomass#",
    "geo": "http://www.w3.org/2003/01/geo/wgs84_pos#",
    "qudt": "http://qudt.org/schema/qudt/",
    "unit": "http://qudt.org/vocab/unit/",

    "TraceableUnit": "biomass:TraceableUnit",
    "Organization": "schema:Organization",
    "Transaction": "schema:Order",
    "MaterialProcessing": "prov:Activity",

    "traceableUnitId": {
      "@id": "schema:identifier",
      "@type": "schema:Text"
    },
    "organizationId": {
      "@id": "schema:identifier",
      "@type": "schema:Text"
    },
    "createdAt": {
      "@id": "schema:dateCreated",
      "@type": "xsd:dateTime"
    },
    "modifiedAt": {
      "@id": "schema:dateModified",
      "@type": "xsd:dateTime"
    }
  }
}
```

11.3 Vocabulary Mappings

11.3.1 Schema.org Integration

BOOST entities map to Schema.org types for web compatibility:

- **Organization** → schema:Organization
- **Transaction** → schema:Order
- **GeographicData** → schema:Place
- **Certificate** → schema:Certification
- **Claim** → schema:Claim

11.3.2 W3C PROV Ontology

Provenance tracking using PROV vocabulary:

- **MaterialProcessing** → prov:Activity
- **ProcessingHistory** → prov:Entity
- **Operator** → prov:Agent
- **wasGeneratedBy** → prov:wasGeneratedBy
- **wasAttributedTo** → prov:wasAttributedTo

11.3.3 GS1 Vocabulary

Supply chain standards alignment:

- **productCode** → gs1:gtin
- **locationCode** → gs1:glN
- **shipmentId** → gs1:sscc
- **batchNumber** → gs1:batchNumber

11.4 Entity Context Examples

11.4.1 TraceableUnit with Context

Complete JSON-LD representation of a TraceableUnit:

JSON Example: TraceableUnit JSON-LD

```
{
  "@context": "https://boost.org/context.jsonld",
  "@type": "biomass:TraceableUnit",
  "@id": "https://example.org/tru/TRU-2025-001",

  "traceableUnitId": "TRU-2025-001",
  "unitType": "pile",
  "totalVolume": {
    "@type": "qudt:QuantityValue",
    "qudt:value": 500.0,
    "qudt:unit": "unit:M3"
  },
  "speciesComposition": [{
    "@type": "biomass:SpeciesComponent",
    "species": "Pseudotsuga menziesii",
    "percentage": 75.0
  }],
  "harvestLocation": {
    "@type": "geo:Point",
    "geo:lat": 45.5231,
    "geo:long": -122.6765
  },
  "prov:wasGeneratedBy": {
    "@id": "https://example.org/harvest/HARV-2025-001"
  },
  "prov:wasAttributedTo": {
    "@id": "https://example.org/org/ORG-FOREST-001"
  }
}
```

11.4.2 Transaction with Linked Data

Transaction linking multiple entities:

JSON Example: Linked Transaction

```
{
  "@context": "https://boost.org/context.jsonld",
  "@type": "schema:Order",
  "@id": "https://example.org/txn/TXN-2025-001",

  "transactionId": "TXN-2025-001",
  "schema:seller": {
    "@id": "https://example.org/org/ORG-SUPPLIER-001"
  },
  "schema:buyer": {
    "@id": "https://example.org/org/ORG-BUYER-001"
  },
  "schema:orderedItem": [{
    "@id": "https://example.org/tru/TRU-2025-001"
  }],
  "schema:price": {
    "@type": "schema:PriceSpecification",
    "schema:price": 85.50,
    "schema:priceCurrency": "USD"
  },
  "prov:startedAtTime": "2025-01-15T09:00:00Z",
  "prov:endedAtTime": "2025-01-15T14:30:00Z"
}
```

11.5 Advanced Features

11.5.1 Named Graphs

Support for multi-source data using named graphs:

JSON Example: Named Graph Structure

```

{
  "@context": "https://boost.org/context.jsonld",
  "@graph": [{
    "@id": "https://example.org/graph/supplier",
    "@graph": [
      {
        "@type": "Organization",
        "organizationId": "ORG-001",
        "name": "Forest Products Inc"
      }
    ]
  }, {
    "@id": "https://example.org/graph/certification",
    "@graph": [
      {
        "@type": "Certificate",
        "certificateId": "CERT-FSC-001",
        "issuedTo": {"@id": "ORG-001"}
      }
    ]
  }
]}

```

11.5.2 Framing

JSON-LD framing for specific data views:

JSON Example: Frame Definition

```

{
  "@context": "https://boost.org/context.jsonld",
  "@type": "TraceableUnit",
  "harvestedBy": {
    "@type": "Organization",
    "certifications": {
      "@type": "Certificate",
      "certificationType": "FSC"
    }
  }
}

```

11.5.3 Compaction and Expansion

BOOST supports JSON-LD algorithms:

- **Compaction:** Shortens IRIs using context
- **Expansion:** Expands to full IRIs
- **Flattening:** Creates flat graph structure
- **Normalization:** Canonical RDF representation

11.6 Context Negotiation

11.6.1 Content Type Headers

HTTP content negotiation support:

- `application/ld+json` - JSON-LD format
- `application/json` - Plain JSON (context link in header)
- `text/turtle` - RDF Turtle format
- `application/n-quads` - N-Quads format

11.6.2 Profile Parameters

Profile-based context selection:

```
Accept: application/ld+json;  
       profile="https://boost.org/profiles/extended"
```

11.7 Implementation Guidance

11.7.1 Python Implementation

Using PyLD library for JSON-LD processing:

Python Example: JSON-LD Processing

```
from pyld import jsonld
import json

# Load BOOST context
with open('boost_context.jsonld') as f:
    context = json.load(f)

# Create entity with context
tru = {
    "@context": context,
    "@type": "TraceableUnit",
    "traceableUnitId": "TRU-001",
    "totalVolume": 100.0
}

# Expand to full IRIs
expanded = jsonld.expand(tru)

# Compact with custom context
compactd = jsonld.compact(expanded, context)

# Convert to RDF
rdf = jsonld.to_rdf(tru)

# Frame for specific view
frame = {"@type": "TraceableUnit"}
framed = jsonld.frame(tru, frame)
```

11.7.2 JavaScript Implementation

Browser and Node.js support:

JSON Example: JavaScript JSON-LD

```
const jsonld = require('jsonld');

// Process BOOST data
async function processBoostData(data) {
  // Add context
  data['@context'] = 'https://boost.org/context.jsonld';

  // Validate structure
  const expanded = await jsonld.expand(data);

  // Generate RDF
  const nquads = await jsonld.toRDF(data, {format: 'N-Quads'});

  return nquads;
}
```

11.8 Semantic Validation

11.8.1 SHACL Constraints

Shape validation for semantic correctness:

JSON Example: SHACL Shape

```
{
  "@context": {"sh": "http://www.w3.org/ns/shacl#"},
  "@type": "sh:NodeShape",
  "sh:targetClass": "biomass:TraceableUnit",
  "sh:property": [{
    "sh:path": "biomass:totalVolume",
    "sh:datatype": "xsd:decimal",
    "sh:minInclusive": 0,
    "sh:maxInclusive": 10000
  }]
}
```

11.8.2 Reasoning and Inference

Automatic inference capabilities:

- Type inheritance from parent classes
- Property domain/range validation
- Transitive relationship discovery

- Consistency checking

11.9 Benefits and Use Cases

11.9.1 Interoperability Benefits

- **Global Identifiers:** IRIs enable worldwide unique identification
- **Vocabulary Reuse:** Leverage existing ontologies
- **Tool Ecosystem:** Compatible with RDF/SPARQL tools
- **Web Integration:** SEO and knowledge graph inclusion

11.9.2 Supply Chain Use Cases

- **Cross-Organization Linking:** Connect data across partners
- **Provenance Tracking:** Complete chain of custody
- **Regulatory Reporting:** Machine-readable compliance data
- **Certification Verification:** Linked certificate validation

The JSON-LD context provides BOOST with semantic web capabilities essential for modern supply chain interoperability and regulatory compliance.

12 Regulatory Program Compliance

The BOOST standard provides comprehensive support for regulatory compliance across multiple biofuel programs, with primary focus on the California Low Carbon Fuel Standard (LCFS). This section documents programmatic reporting workflows, compliance requirements, and implementation guidance for regulatory submissions.

12.1 Low Carbon Fuel Standard (LCFS) Overview

The California Low Carbon Fuel Standard, administered by the California Air Resources Board (CARB), is a market-based regulation designed to reduce greenhouse gas emissions from transportation fuels. BOOST provides specialized entities and validation rules to support complete LCFS compliance workflows.

12.1.1 Regulatory Context

The LCFS program requires regulated parties to:

- Track all fuel transactions with certified pathway attribution
- Calculate carbon intensity using CARB-approved methodologies
- Submit quarterly reports with complete audit trails

- Maintain third-party verification documentation
- Demonstrate compliance with sustainability criteria

12.1.2 BOOST's Role in LCFS Compliance

BOOST enables LCFS compliance through:

- **Pathway Management:** LcfsPathway entity for CARB-certified pathways
- **Transaction Tracking:** Enhanced Transaction with LCFS-specific fields
- **Quarterly Reporting:** LcfsReporting entity for regulatory submissions
- **Credit Calculations:** Automated credit/deficit calculation with validation
- **Audit Trail:** Complete traceability from feedstock to fuel product

12.2 LCFS Entity Integration

12.2.1 LcfsPathway Entity

The LcfsPathway entity manages CARB-certified fuel pathways:

- pathwayId: CARB-assigned pathway identifier
- pathwayType: Lookup Table, Tier 1, or Tier 2 pathway
- carbonIntensity: Certified CI value (gCO_{2e}/MJ)
- energyEconomyRatio: EER for credit calculation
- certificationDate: CARB certification date
- expirationDate: Pathway expiration date
- verificationStatus: Active, suspended, or expired
- caGreetVersion: CA-GREET model version used

Pathway Validation Rules:

- Pathway must be active for transaction date
- Carbon intensity must match CARB database
- Feedstock must align with pathway specifications
- Facility location must match certified production site

12.2.2 Enhanced Transaction Entity

LCFS-specific transaction fields include:

- lcfsPathwayId: Foreign key to certified pathway
- fuelVolume: Volume in gallons or GGE

- **fuelCategory:** Fuel type classification
- **reportingPeriod:** YYYY-Q# format
- **regulatedPartyRole:** Producer, importer, blender, or distributor

12.2.3 LcfsReporting Entity

Quarterly reporting aggregation:

- **reportingPeriod:** Quarter identifier (e.g., "2025-Q1")
- **totalFuelVolume:** Aggregate fuel volume
- **totalCreditsGenerated:** Credits from CI below benchmark
- **totalDeficitsIncurred:** Deficits from CI above benchmark
- **netPosition:** Net credit/deficit position
- **verificationStatus:** Third-party verification status

12.3 Programmatic Reporting Workflows

12.3.1 Quarterly Report Generation

The quarterly reporting process follows these steps:

1. **Transaction Aggregation:** Collect all transactions for reporting period
2. **Pathway Validation:** Verify pathway status and attributes
3. **Credit Calculation:** Apply CARB formulas with EER adjustment
4. **Report Generation:** Create structured report for submission
5. **Verification:** Third-party review if required
6. **Submission:** Upload to CARB reporting system

12.3.2 Credit Calculation Methodology

LCFS credits are calculated using the formula:

$$\text{Credits} = (\text{Benchmark CI} - \text{Pathway CI}) \times \text{Fuel Volume} \times \text{Energy Density} \times \text{EER} \times 10^{-6} \quad (1)$$

Where:

- **Benchmark CI** = CARB-specified carbon intensity target
- **Pathway CI** = Certified pathway carbon intensity
- **Fuel Volume** = Transaction volume in gallons

- Energy Density = MJ per gallon for fuel type
- EER = Energy Economy Ratio for application

12.3.3 Data Reconciliation Process

Monthly reconciliation ensures data integrity:

1. Compare transaction records with physical inventory
2. Validate pathway assignments against production records
3. Cross-check credit calculations with manual verification
4. Document discrepancies in DataReconciliation entity
5. Generate reconciliation report for audit trail

12.4 Implementation Examples

12.4.1 Renewable Diesel Production Example

Complete workflow for renewable diesel with forest residue feedstock:

JSON Example: LCFS Transaction with Pathway

```
{
  "transactionId": "TXN-LCFS-2025Q1-001",
  "transactionType": "fuel_sale",
  "fuelVolume": 10000,
  "fuelVolumeUnit": "gallons",
  "fuelCategory": "renewable_diesel",
  "lcfsPathwayId": "PATH-CARB-RD-001",
  "reportingPeriod": "2025-Q1",
  "organizationId": "ORG-PACIFIC-001",
  "regulatedPartyRole": "producer",
  "traceableUnitIds": ["TRU-FOREST-001", "TRU-FOREST-002"],
  "carbonIntensity": 35.5,
  "benchmarkCI": 94.17,
  "creditsGenerated": 5862.5
}
```

12.4.2 Credit Calculation Example

Using actual CARB values for renewable diesel:

- Fuel Volume: 10,000 gallons
- Energy Density: 129.65 MJ/gallon (renewable diesel)

- Benchmark CI: 94.17 gCO₂e/MJ (2025 diesel target)
- Pathway CI: 35.50 gCO₂e/MJ (certified pathway)
- EER: 1.0 (heavy-duty diesel application)

Calculation:

$$\text{Credits} = (94.17 - 35.50) \times 10,000 \times 129.65 \times 1.0 \times 10^{-6} \quad (2)$$

$$= 58.67 \times 10,000 \times 129.65 \times 10^{-6} \quad (3)$$

$$= 7,607 \text{ MT CO}_2\text{e credits} \quad (4)$$

12.5 Multi-Program Compliance Framework**12.5.1 Renewable Fuel Standard (RFS) Integration**

BOOST supports RFS compliance through:

- RIN generation and tracking capabilities
- D-code classification for renewable fuel categories
- EPA pathway registration support
- Quarterly RFS reporting integration

12.5.2 EU Renewable Energy Directive (RED II) Compliance

European compliance features include:

- GHG savings calculation (minimum 65% for new facilities)
- Sustainability criteria verification
- Land use change documentation
- Mass balance chain of custody
- ISCC certification integration

12.5.3 Regional Program Extensions

Support for state-level programs:

- **Oregon Clean Fuels Program:** Similar to LCFS with state-specific pathways
- **Washington Clean Fuel Standard:** Launched 2023 with unique requirements
- **British Columbia LCFS:** Provincial program with federal alignment
- **Canada Clean Fuel Regulations:** National program with credit trading

12.6 Data Quality and Compliance

12.6.1 CARB Data Validation Requirements

All submissions must meet CARB data quality standards:

- **Completeness:** 100% transaction coverage required
- **Accuracy:** Volume tolerance $\pm 0.5\%$
- **Timeliness:** Quarterly submission within 45 days
- **Consistency:** Cross-period reconciliation required
- **Traceability:** Complete audit trail maintained

12.6.2 Third-Party Verification

Large regulated entities require annual verification:

1. Engage CARB-accredited verification body
2. Provide access to BOOST data systems
3. Support site visits and record reviews
4. Address verification findings
5. Submit verification statement with annual report

12.7 Technical Implementation

12.7.1 API Endpoints for LCFS Data

RESTful API design for LCFS operations:

- GET `/lcfs/pathways` - Retrieve active pathways
- POST `/lcfs/transactions` - Submit fuel transaction
- GET `/lcfs/reports/{period}` - Generate quarterly report
- POST `/lcfs/credits/calculate` - Calculate credits/deficits
- GET `/lcfs/reconciliation/{period}` - Reconciliation report

12.7.2 Automated Report Generation

Python implementation for quarterly reports:

Python Example: LCFS Report Generation

```
from boost_client import create_client
from datetime import datetime

client = create_client()

# Generate Q1 2025 LCFS report
report = client.generate_lcfs_report(
    reporting_period="2025-Q1",
    organization_id="ORG-PACIFIC-001"
)

# Aggregate transactions by pathway
pathway_summary = report.aggregate_by_pathway()

# Calculate total credits/deficits
total_credits = sum(t.credits for t in report.transactions)
total_deficits = sum(t.deficits for t in report.transactions)
net_position = total_credits - total_deficits

# Generate CARB submission format
carb_report = report.format_for_carb_submission()

# Export to required XML format
report.export_to_xml("lcfs_2025_q1_submission.xml")
```

12.7.3 Error Handling and Validation

Comprehensive validation before submission:

- Pathway expiration checking
- Volume balance verification
- Credit calculation validation
- Duplicate transaction detection
- Missing data identification
- Format compliance checking

12.8 Regulatory Change Management

BOOST adapts to regulatory updates through:

- **Schema Versioning:** Track regulatory requirement changes
- **Validation Rule Updates:** Modify business logic for new requirements
- **Backward Compatibility:** Maintain historical data integrity

- **Migration Tools:** Update existing data to new standards
- **Compliance Alerts:** Notify users of regulatory changes

This comprehensive framework ensures BOOST implementations maintain full regulatory compliance while adapting to evolving program requirements across multiple jurisdictions.

13 Security Considerations

13.1 Data Privacy

Implementations **SHOULD** consider privacy implications of biomass tracking data:

- Location data may reveal sensitive commercial information
- Biometric identifiers require secure storage and transmission
- Personal operator information needs appropriate access controls

13.2 Data Integrity

Critical security measures include:

- Digital signatures for high-value transactions
- Audit trails for all data modifications
- Backup and recovery procedures for critical supply chain data
- Validation of external data sources and certificates

Security Implementation Requirements

Implementations **MUST** address authentication of supply chain participants, authorization controls for data access, secure communication channels, and fraud detection mechanisms.

14 Examples

14.1 Basic TraceableUnit Example

JSON Example: TraceableUnit JSON-LD Example

```
{
  "@context": "https://boost-standard.org/context.jsonld",
  "@type": "TraceableUnit",
  "@id": "https://example.com/tru/TRU-001",
  "traceableUnitId": "TRU-FOREST-001",
  "unitType": "pile",
  "uniqueIdentifier": "BIOMETRIC-SIGNATURE-ABC123",
  "totalVolumeM3": 125.5,
  "materialTypeId": "MAT-DOUGLAS-FIR-SAWLOG",
  "isMultiSpecies": false,
  "harvesterId": "ORG-PACIFIC-FOREST",
  "currentGeographicDataId": "GEO-MILL-YARD-07"
}
```

15 Resources & Community

15.1 Presentations & Demonstrations

The BOOST Community Group has developed comprehensive presentations and demonstrations including:

- BOOST Kickoff Presentation - Overview of the data standard initiative
- Transaction Object Examples - Technical demonstration of data structures
- California Agency Engagement presentations for CalRecycle, CDFA, and Department of Conservation
- BOOST + LCFS Integration technical presentation

15.2 Community Participation

BOOST Membership

Chair: Peter Tittmann (Carbon Direct)

Active Participants: 15+ members from industry stakeholders, regulatory agencies, certification bodies, and technology providers across the biomass supply chain.

A Entity Relationship Diagrams

A.1 Complete Entity Relationship Overview

The BOOST data model comprises 33 interconnected entities organized into 7 thematic areas:

A.2 Thematic Area Diagrams

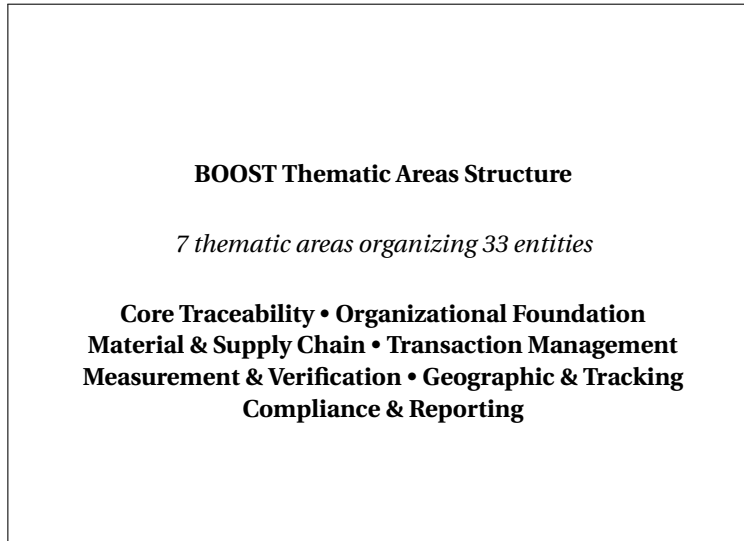


Figure 2: BOOST Thematic Areas and Entity Organization

B JSON Schema Reference

B.1 Schema Validation Rules

This appendix provides detailed information about JSON schema validation rules for all BOOST entities.

B.2 Context Definitions

The BOOST JSON-LD context **MUST** define:

- Semantic mappings for all entity types
- Property definitions with appropriate vocabularies
- Data type specifications for typed literals
- Language specifications for internationalization

C Python Reference Implementation

The BOOST standard provides a comprehensive Python reference implementation that demonstrates dynamic, schema-driven data models, validation, and supply chain tracking capabilities for biomass chain of custody operations.

C.1 Overview

The Python reference implementation uses a **dynamic, schema-driven architecture** that automatically adapts to changes in BOOST JSON schemas without requiring code modifications. Key features include:

- **Dynamic Schema-Driven Architecture:** Automatically adapts to schema changes without code modifications
- **Comprehensive Validation:** Schema, business logic, and cross-entity validation with 8 categories of business rules
- **Dynamic Model Generation:** Pydantic models generated directly from JSON schemas at runtime
- **Configuration-Driven Business Rules:** Business logic validation rules defined in configuration files
- **Supply Chain Tracking:** Complete traceability with automatic relationship discovery
- **Multi-Certification Support:** FSC, SBP, PEFC, ISCC, RED II compliance validation
- **Mass Balance Accounting:** Volume and mass conservation validation with configurable tolerance checking
- **JSON-LD Export/Import:** Full semantic web compatibility with schema.org and W3C PROV ontology support
- **Schema Version Compatibility:** Graceful handling of schema evolution and backward compatibility

C.2 Installation

C.2.1 Prerequisites

The Python reference implementation requires:

- Python 3.8 or higher
- pip package manager

C.2.2 Dependencies

Core dependencies are defined in `requirements.txt`:

```
Python Example: Python Dependencies
```

```
pydantic>=2.0.0      # Data validation and settings management
jsonschema>=4.0.0   # JSON Schema validation
requests>=2.28.0    # HTTP library for API calls
pyld>=2.0.0         # JSON-LD processor
```

Installation:

Python Example: Installation Command

```
pip install -r requirements.txt
```

C.3 Architecture

The implementation follows a layered architecture with three main components:

1. **Schema Loader:** Dynamic schema loading and model generation engine
2. **Dynamic Validation:** Configuration-driven validation with 8 categories of business rules
3. **BOOST Client:** High-level API interface using dynamic models

The architecture ensures automatic adaptation to schema changes while providing comprehensive validation and traceability capabilities.

C.4 Core Components

C.4.1 SchemaLoader

The **SchemaLoader** (`schema_loader.py`) is the foundation component that provides dynamic schema loading and model generation:

Key Features:

- **Automatic Schema Discovery:** Scans directories for `validation_schema.json` files
- **Dynamic Model Generation:** Creates Pydantic models from JSON schemas at runtime
- **Enum Generation:** Dynamically creates Python enums from schema definitions
- **Relationship Discovery:** Analyzes schemas to discover foreign key relationships automatically
- **Primary Key Detection:** Identifies primary key fields from schema patterns

Usage Example:**Python Example: SchemaLoader Usage**

```
from schema_loader import SchemaLoader

# Initialize with automatic schema discovery
loader = SchemaLoader()

# Get dynamically generated Pydantic models
OrganizationModel = loader.get_model('organization')
TraceableUnitModel = loader.get_model('traceable_unit')

# Get enum values directly from current schemas
org_types = loader.get_field_enum_values('organization',
                                         'organizationType')
print(f"Available organization types: {org_types}")

# Access relationship information discovered from schemas
relationships = loader.get_relationships('traceable_unit')
primary_key = loader.get_primary_key('organization')
```

C.4.2 DynamicBOOSTValidator

The **DynamicBOOSTValidator** (`dynamic_validation.py`) provides comprehensive, schema-driven validation using configuration-based business rules:

Validation Categories:

1. **Schema Validation:** JSON Schema compliance and structural validation
2. **Volume/Mass Conservation:** Physical conservation laws with configurable tolerance checking
3. **Temporal Logic:** Date/time consistency rules and processing sequence validation
4. **Geographic Logic:** Location-based constraints and transport distance validation
5. **Species Composition:** Biological consistency and percentage validation
6. **Certification Logic:** Chain of custody validation and certificate integrity
7. **Regulatory Compliance:** LCFS, EU RED, and sustainability criteria validation
8. **Economic/Quality Logic:** Market constraints and quality assurance validation

Usage Example:

Python Example: DynamicBOOSTValidator Usage

```
from dynamic_validation import DynamicBOOSTValidator

validator = DynamicBOOSTValidator()

# Schema validation against current schema
is_valid, errors = validator.validate_entity("organization",
    ↪ org_data)

# Configuration-driven business logic validation
is_valid, errors = validator.validate_business_logic(
    "material_processing", processing_data)

# Comprehensive cross-entity validation
entities = {
    'organization': [org1, org2],
    'traceable_unit': [tru1, tru2],
    'transaction': [txn1]
}
results = validator.comprehensive_validation(entities)
```

C.4.3 BOOSTClient

The **BOOSTClient** (`boost_client.py`) provides a high-level interface that uses the dynamic models and validation system:

Core Functions:

- **Entity Creation:** Create entities using dynamically generated models with automatic validation
- **Schema Introspection:** Query available entities, enums, and constraints from current schemas
- **Supply Chain Analysis:** Trace relationships and analyze supply chains using dynamic models
- **Validation:** Comprehensive validation using all dynamic rules and business logic
- **JSON-LD Support:** Export/import with semantic annotations and context management

Usage Example:

Python Example: BOOSTClient Usage

```
from boost_client import create_client

# Initialize client with dynamic schema loading
client = create_client()

# Schema introspection
schema_info = client.get_schema_info()
print(f"Available entities:
↳ {schema_info['available_entities']}")

# Dynamic enum discovery
org_types = client.get_available_enum_values('organization',
                                             'organizationType')

# Entity creation with schema validation
org = client.create_organization(
    organization_id="ORG-FOREST-001",
    name="Pacific Forest Products",
    org_type="harvester", # Validated against current schema
    contact_email="ops@pacificforest.com"
)

# Comprehensive validation
validation = client.validate_all()
if validation['valid']:
    print("$\checkmark$ All entities pass validation!")
```

C.5 Dynamic Schema Adaptation

A key strength of the Python implementation is its **automatic adaptation to schema changes**. Most schema modifications require **no code changes**.

C.5.1 Automatically Handled Changes

Adding New Fields:

- New optional fields are immediately available
- Required fields trigger validation updates automatically
- Default values from schemas are applied automatically

Adding New Enum Values:

- New enum values become available immediately after schema reload
- Validation rules update automatically

- No code changes required

Adding New Entity Types:

- New schema files are discovered automatically
- Dynamic models are generated on first access
- All validation rules apply automatically

Modifying Business Logic Rules:

- Configuration file changes are applied automatically
- Tolerance values and thresholds update dynamically
- Cross-entity validation rules adapt to changes

C.5.2 Schema Change Detection

The system provides built-in tools for schema change management:

Python Example: Schema Change Detection

```
# Check current schema status
client = create_client()
schema_info = client.get_schema_info()

# Validate against current schema
validation = client.validate_all()
if not validation['valid']:
    print("Schema changes detected - validation errors:")
    for error in validation['errors']:
        print(f" - {error}")

# Refresh schemas after updates
client.refresh_schemas()
```

C.6 Usage Examples

C.6.1 Basic Workflow

Complete example demonstrating fundamental BOOST operations:

Python Example: Basic Workflow Example

```
from boost_client import create_client

# Initialize BOOST client
client = create_client()

# Create organizations with schema validation
harvester = client.create_organization(
    organization_id="ORG-001",
    name="Forest Products Inc",
    org_type="harvester",
    contact_email="ops@forestproducts.com"
)

processor = client.create_organization(
    organization_id="ORG-002",
    name="Sawmill Operations LLC",
    org_type="processor",
    contact_email="info@sawmill.com"
)

# Create traceable units with automatic model generation
log_pile = client.create_traceable_unit(
    traceable_unit_id="TRU-LOGS-001",
    unit_type="pile",
    harvester_id="ORG-001",
    total_volume_m3=125.5,
    sustainability_certification="FSC Mix Credit 70%"
)

# Process materials with conservation validation
lumber = client.create_material_processing(
    processing_id="MP-001",
    input_tru_id="TRU-LOGS-001",
    process_type="sawing",
    processor_id="ORG-002",
    output_volume_m3=95.2 # Validates against conservation
    ~ rules
)

# Execute transaction with comprehensive validation
transaction = client.create_transaction(
    transaction_id="TXN-001",
    organization_id="ORG-002",
    customer_id="CUST-001",
    transaction_date="2025-08-12",
    quantity_m3=50.0
)

# Comprehensive validation using all dynamic rules
validation = client.validate_all()
if validation['valid']:
    print("$\checkmark$ All entities validated successfully!")

# Export to JSON-LD with semantic annotations
jsonld_output = client.export_to_jsonld(include_context=True)
```

C.6.2 Certification Management

Example showing certification claim management:

Python Example: Certification Management Example

```
# Create FSC certified organization
fsc_harvester = client.create_organization(
    organization_id="ORG-FSC-001",
    name="Certified Forest Management",
    org_type="harvester",
    certifications=["FSC-FM/COC-001234"]
)

# Create certified traceable unit
certified_logs = client.create_traceable_unit(
    traceable_unit_id="TRU-FSC-001",
    unit_type="pile",
    harvester_id="ORG-FSC-001",
    total_volume_m3=200.0,
    sustainability_certification="FSC Mix Credit 70%",
    certification_claims=["FSC-FM/COC-001234"]
)

# Validate certification chain integrity
cert_validation =
    - client.validate_certification_chain("TRU-FSC-001")
print(f"Certification valid: {cert_validation['valid']}")
```

C.6.3 Mass Balance Validation

Example demonstrating conservation law validation:

Python Example: Mass Balance Validation Example

```
# Multiple input materials
input_tru_1 = client.create_traceable_unit(
    traceable_unit_id="TRU-INPUT-001",
    unit_type="pile",
    total_volume_m3=100.0
)

input_tru_2 = client.create_traceable_unit(
    traceable_unit_id="TRU-INPUT-002",
    unit_type="pile",
    total_volume_m3=75.0
)

# Processing with multiple inputs
pellet_production = client.create_material_processing(
    processing_id="MP-PELLETS-001",
    input_tru_ids=["TRU-INPUT-001", "TRU-INPUT-002"],
    process_type="pelletizing",
    total_input_volume_m3=175.0,
    total_output_volume_m3=140.0, # Within tolerance for
    ↪ pelletizing
    efficiency_percent=80.0
)

# Validate mass balance with configurable tolerance
balance_validation =
    ↪ client.validate_mass_balance("MP-PELLETS-001")
print(f"Mass balance valid: {balance_validation['valid']}")
print(f"Efficiency: {balance_validation['efficiency']}%")
```

C.7 Integration Guidance

C.7.1 API Development

Using the reference implementation for API development:

Python Example: API Development Example

```
from boost_client import create_client
from flask import Flask, jsonify, request

app = Flask(__name__)
boost_client = create_client()

@app.route('/organizations', methods=['POST'])
def create_organization():
    data = request.json
    try:
        # Dynamic validation using current schema
        org = boost_client.create_organization(**data)
        return jsonify(org.model_dump(by_alias=True))
    except ValueError as e:
        return jsonify({"error": str(e)}), 400

@app.route('/validate/<entity_type>', methods=['POST'])
def validate_entity(entity_type):
    data = request.json
    validation =
    ↪ boost_client.validator.validate_entity(entity_type,
    ↪ data)
    return jsonify({
        "valid": validation[0],
        "errors": validation[1]
    })

# Schema introspection endpoint
@app.route('/schema/info')
def schema_info():
    return jsonify(boost_client.get_schema_info())
```

C.8 Configuration

C.8.1 Schema Path Configuration

Customize schema loading:

Python Example: Schema Path Configuration

```
# Default: automatic discovery from ../schema/  
client = create_client()  
  
# Custom schema path  
client = create_client(schema_path="/path/to/boost/schemas")  
  
# Multiple schema sources  
loader = SchemaLoader()  
loader.add_schema_source("/additional/schemas")
```

C.8.2 Business Logic Configuration

Business logic rules are defined in configuration files:

JSON Example: Business Logic Configuration

```
{  
  "volumeMassConservation": {  
    "materialProcessing": {  
      "sawing": {  
        "tolerance": 0.05,  
        "efficiency_range": [0.7, 0.9]  
      },  
      "pelletizing": {  
        "tolerance": 0.10,  
        "efficiency_range": [0.75, 0.85]  
      }  
    }  
  },  
  "temporalLogic": {  
    "processingWindows": {  
      "harvest_to_processing_max_days": 90  
    }  
  }  
}
```

C.9 Testing and Validation

C.9.1 Comprehensive Test Suite

The implementation includes comprehensive tests:

Python Example: Running Tests

```
# Run all tests
python test_enhanced_entities.py

# Test specific validation categories
python -m unittest
  ↳ test_enhanced_entities.TestDynamicValidation.test_mass_balance_validation

# Test schema change robustness
python -m unittest test_enhanced_entities.TestSchemaRobustness
```

C.9.2 Validation Examples

Test validation with example data:

Python Example: Validation Examples

```
# Load and validate example data
with
  ↳ open('examples/validation/comprehensive_validation_test_suite.json')
  ↳ as f:
    test_data = json.load(f)

validator = DynamicBOOSTValidator()
results = validator.comprehensive_validation(test_data)

print(f"Validation results: {results['summary']}")
for category, result in results['by_category'].items():
    print(f"  {category}: {'PASS' if result['valid'] else
      ↳ 'FAIL'}")
```

C.10 Performance Characteristics

C.10.1 Initialization Performance

- **Schema Loading:** $O(n)$ where n = number of schema files
- **Model Generation:** $O(m)$ where m = number of entity properties
- **Caching:** Models cached after first generation for $O(1)$ access

C.10.2 Runtime Performance

- **Validation:** $O(1)$ for schema validation, $O(r)$ for relationship validation where r = relationships

- **Entity Creation:** O(1) with cached models
- **Memory Usage:** Moderate (dynamic models cached in memory)

C.10.3 Scalability Considerations

- **Large Datasets:** Supports batch validation operations
- **Memory Management:** Efficient caching with configurable limits
- **Concurrent Access:** Thread-safe validation operations

C.11 Standards Compliance

The Python reference implementation fully supports:

- **BOOST Data Standard** (with automatic adaptation to schema updates)
- **JSON-LD 1.1 Specification**
- **JSON Schema Draft-07**
- **Schema.org Vocabulary** for semantic annotations
- **W3C PROV Ontology** for provenance tracking

References

References

- [1] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. RFC 2119, IETF, March 1997. <https://tools.ietf.org/rfc/rfc2119>
- [2] B. Leiba. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. RFC 8174, IETF, May 2017. <https://tools.ietf.org/rfc/rfc8174>
- [3] Gregg Kellogg, Pierre-Antoine Champin, Dave Longley. *JSON-LD 1.1*. W3C Recommendation, 16 July 2020. <https://www.w3.org/TR/json-ld11/>
- [4] Austin Wright, Henry Andrews. *JSON Schema: A Media Type for Describing JSON Documents*. March 2019. <https://json-schema.org/specification.html>
- [5] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub. *The GeoJSON Format*. RFC 7946, IETF, August 2016. <https://tools.ietf.org/rfc/rfc7946>
- [6] *Chain of custody of wood and wood-based products*. ISO 38200:2018, International Organization for Standardization, 2018. <https://www.iso.org/standard/69429.html>
- [7] *Chain of Custody Standard*. SBP Standard 4, Version 1.0, Sustainable Biomass Partnership, 2013. <https://sbp-cert.org/documents/standards-documents/>
- [8] *Collection and Communication of Data*. SBP Standard 5, Version 1.0, Sustainable Biomass Partnership, 2013. <https://sbp-cert.org/documents/standards-documents/>
- [9] *Chain of Custody Certification*. FSC-STD-40-004, Version 3.0, Forest Stewardship Council, 2017. <https://fsc.org/en/document-centre/documents/resource/392>
- [10] *Chain of Custody of Forest Based Products*. PEFC ST 2002:2020, Programme for Endorsement of Forest Certification, 2020. <https://www.pefc.org/standards/chain-of-custody>
- [11] *Low Carbon Fuel Standard Regulation*. California Air Resources Board, 2024. <https://ww2.arb.ca.gov/our-work/programs/low-carbon-fuel-standard>
- [12] *Renewable Energy Directive II*. Directive (EU) 2018/2001, European Union, 2018. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.328.01.0082.01.ENG
- [13] A. Kaulen, L. Stopfer, K. Lippert, T. Purfürst. *Systematics of Forestry Technology for Tracing the Timber Supply Chain*. 2023. <https://github.com/carbondirect/BOOST/tree/main/references>

Acknowledgments

This specification was developed through the collaborative efforts of the BOOST W3C Community Group with significant contributions from:

- **California Department of Conservation** - Funding and regulatory guidance
- **Forest industry stakeholders** - Requirements analysis and use case development
- **Certification bodies** - Standards alignment and validation procedures
- **Technology providers** - Implementation guidance and tool development
- **Academic institutions** - Research and analysis support
- **Environmental organizations** - Sustainability criteria and verification methods

Special recognition to the contributors of the Interactive ERD Navigator, Python reference implementation, and comprehensive schema validation tools that support this specification.

Technical Development Contributions:

- **Claude (Anthropic)** - AI-assisted development including schema integrity validation, Python reference implementation enhancements, comprehensive documentation generation, expert review integration, and systematic issue resolution supporting v3.4.2-10-gb83f99a release preparation