

FishermenFirst Platform - Technical Documentation

Project Overview

This document outlines the technical architecture and implementation plan for the FishermenFirst analytics platform, which will support two distinct Alaska fishery management programs:

1. **Gulf of Alaska Trawl Electronic Monitoring (TEM) Incentive Plan Agreement**
2. **Central Gulf of Alaska Rockfish Program (CGOA RP) Intercooperative/Cooperative Management**

Key Decision: Platform Developer vs. Manager

The project owner will serve as the **technical platform developer only**, not the operational manager. This means:

- Building the data infrastructure and analytics platform
- Creating automated reporting systems
- Providing training and documentation
- Offering ongoing technical support
- **NOT responsible for:** Daily operations, stakeholder management, or regulatory compliance decisions

Program Overview

TEM Program

- **Participants:** ~56 vessels in Gulf of Alaska pollock fishery
- **Key Requirements:**
 - Track 4-trip averages (300,000 lb limit)
 - Monitor Maximum Retainable Amounts (MRA)
 - Calculate and invoice overages
- **Workload:** ~200 hours/year, concentrated in May and November

Rockfish Program

- **Participants:** 46 LLPs across 4 cooperatives (~20 active vessels)
- **Key Requirements:**
 - Real-time quota tracking across multiple species
 - Salmon bycatch monitoring (1,200 fish cap)

- Complex transfer and cap management
- **Workload:** ~866 hours/year, peak April-June

Critical Overlap

Many vessels participate in both programs, necessitating a shared user/vessel infrastructure while maintaining strict data separation.

User Types and Access Levels

Final User Types (Analytics Platform Only)

1. **Vessel Operator/Owner** - Primary users tracking compliance
2. **TEM Program Manager** - Manages violation calculations and billing
3. **Rockfish Program Manager** - Manages quota tracking and salmon monitoring
4. **Platform Administrator** - Technical system maintenance

Note: Co-op Managers (4 Rockfish cooperative presidents) have no initial platform access. This can be added as a paid add-on if requested.

User Task Map (Based on RFPs)

1. Vessel Operator/Owner

TEM Program Tasks:

- Calculate their 4-trip averages (assistance needed per RFP)
- Calculate MRA overages
- Receive overage invoices with documentation

Rockfish Program Tasks:

- Request vessel check-in/out status
- Track their quota and harvests
- Receive salmon hotspot reports

2. TEM Program Manager (Analytics Only)

- View parsed NMFS landing data
- See calculated overages (4-trip and MRA)
- Generate vessel-specific overage reports

- Track payment status
- Export data for Council report

3. Rockfish Program Manager (Analytics Only)

- Track quotas at vessel, cooperative, and intercooperative level
- Monitor for overages and negative balances
- Track harvest towards vessel and processor caps
- Monitor salmon counts for all deliveries
- Analyze bycatch location data
- Calculate overage and lease amounts
- Balance accounting with eFish
- Enter quota transfers via platform form
- Enter initial quota allocations via manual form

4. Platform Administrator

- Monitor data ingestion status
- View system performance metrics
- Review calculation logs
- Track user activity
- Generate platform usage reports

User Flows

Flow 1: Vessel Check TEM Compliance

Login → Dashboard → Select "TEM Program" →
View 4-trip average dashboard →
See current average and trips included →
Calculate remaining capacity →
Make fishing decision

Flow 2: Vessel Check Rockfish Quota

Login → Dashboard → Select "Rockfish Program" →
View quota balances by species →

Check processor cap status →
Make fishing decision

Flow 3: TEM Manager Process Violations

Login → Import NMFS Data →
System parses and calculates →
Review flagged violations →
Generate invoices → Mark as sent

Flow 4: Rockfish Manager Monitor Salmon

Login → Salmon monitoring shows spike →
Review vessel patterns →
Generate hotspot map →
Send alerts to vessels

Flow 5: Vessel Review MRA Violation

Login → Notification of overage →
View calculation breakdown →
See estimated fine →
Download documentation

Flow 6: Vessel Track Salmon Impact

Login → View salmon hotspot map →
Check vessel location vs hotspots →
View fleet-wide count →
Adjust fishing plans

Flow 7: TEM Manager Generate Report

Login → Select "Annual Council Report" →
System compiles statistics →
Export to PDF/PowerPoint

Flow 8: Rockfish Manager Balance Quotas

Login → Quota Reconciliation →

Compare platform vs eFish →
Identify discrepancies →
Flag for investigation

Flow 9: Platform Admin Monitor System

Admin Login → Check email processing →
Check active calculations →
Review error logs →
Re-run failed processes

Flow 10: Vessel Receive Invoice

Email notification → Login →
View invoice details →
Download PDF → Forward to bookkeeper

Flow 11: Rockfish Manager Enter Transfer

Login → Click "Enter Quota Transfer" →
Select vessels and species →
Enter pounds and type →
Preview quota changes →
Submit → Confirmation

Flow 12: Rockfish Manager Enter Initial Allocations

Login → Click "Enter Initial Allocations" →
Select season year →
Select cooperative →
Enter allocation by species →
Review totals →
Finalize allocations

Data Management Approach

Analytics vs. Management Separation

Platform Handles (Analytics):

- Calculations and monitoring

- Report generation
- Data visualization
- Alerts and notifications
- Historical tracking

Outside Platform (Management):

- Vessel check-in/out in eFish
- Transfer approvals
- Regulatory decisions
- Document signatures
- Appeals and exceptions

Quota Transfer Management

Since no API access to eFish exists, the platform will:

1. Provide a simple form for managers to enter transfers
2. Update all affected quota balances immediately
3. Maintain complete audit trail
4. Show real-time updates to vessels via Supabase

Audit Trail Requirements

All quota transfers must track:

- Who made the change (user ID)
- When (exact timestamp)
- What changed (all transfer details)
- Why (reason/notes)
- Before/after quota balances
- IP address for security

MVP Scope and Timeline

TEM MVP - January 2026 Launch

Development: October-December 2025

Features:

1. Data Ingestion

- Parse NMFS email attachments via n8n
- Store landing data in database

2. Calculations Engine

- Calculate 4-trip averages (vessels ≥ 60 ft)
- Calculate calendar day logic (vessels < 60 ft)
- Detect MRA violations against CFR Table 10
- Store all calculation results

3. Vessel Portal

- Login/authentication
- View current 4-trip average status
- View trips included in calculations
- View MRA compliance by species
- View any violations with amounts owed

4. TEM Manager Portal

- Login with manager permissions
- View all vessels' violation summary
- Generate PDF invoices with violation details
- Mark invoices as "sent"
- View parsed landing data

5. Infrastructure

- User authentication (Supabase Auth)
- Permission system (vessel vs manager)
- Audit trail for all calculations
- Basic error handling

Rockfish MVP - April 2026 Launch

Development: February-March 2026

Features:

1. Data Ingestion

- Parse processor landing emails via n8n

- Store landing data with species codes

2. Manual Data Entry

- Form to enter initial quota allocations
- Form to enter completed quota transfers

3. Quota Tracking

- Display current quota by species
- Calculate remaining quota
- Track vessel and cooperative levels
- Show cap utilization

4. Salmon Bycatch Monitoring

- Display Chinook count
- Show fleet total vs 1,200 cap
- Generate hotspot maps
- Track bycatch locations

5. Vessel Portal

- View quota balances
- View utilization percentages
- View transfer history
- View salmon status
- View check-in/out status (read-only)

6. Rockfish Manager Portal

- View all vessels' quota status
- Enter initial allocations
- Enter quota transfers
- View salmon patterns
- Generate hotspot reports
- Calculate payment amounts

Technical Architecture

Core Technology Stack

Selected Stack:

- **Database & Backend:** Supabase (PostgreSQL + Auth + Realtime + Auto-generated APIs)
- **Additional API Layer:** Node.js with Express (only for complex business logic)
- **Automation:** n8n (workflow automation)
- **Frontend:** Next.js
- **Deployment:** Vercel (frontend) + Railway/Render (if Node.js needed)

Rationale: Supabase provides both database and API functionality, reducing complexity. Node.js is only needed for calculations too complex for database functions or Supabase's auto-generated APIs.

Domain Structure

- `fishermenfirst.org` - Marketing site (Squarespace)
- `app.fishermenfirst.org/tem/` - TEM vessel portal (Vercel)
- `app.fishermenfirst.org/rockfish/` - Rockfish cooperative portal (Vercel)
- `app.fishermenfirst.org/admin/` - Internal management tools (Vercel)
- `api.fishermenfirst.org` - Node.js API if needed for complex calculations

Infrastructure Costs

- Squarespace: ~\$16/month
- Supabase: \$25/month
- n8n: \$20/month
- Node.js hosting (if needed): \$20/month

Database Design

Design Principles

1. **Shared Core Tables:** Users, vessels, and processors are shared between programs
2. **Program Isolation:** All operational data (landings, violations, etc.) is strictly separated
3. **Permission-Based Access:** Separate permission tables control program access
4. **Audit Trail:** Comprehensive tracking of all changes and calculations

Table Categories

Shared Core Tables

- `users`: All system users (captains, owners, bookkeepers, managers)
- `vessels`: Vessel registry with permanent identifiers

- `processors`: Processing plants and facilities

Permission/Access Tables

- `tem_participants`: Controls TEM program access
- `rp_participants`: Controls Rockfish program access
- `processor_programs`: Links processors to programs

TEM Program Tables

- `tem_pollock_landings`: Individual delivery records
- `tem_violations`: MRA and trip limit violations
- `tem_trip_calculations`: 4-trip average tracking
- `tem_invoices`: Billing records
- `tem_mra_species_lookup`: Reference table for MRA percentages

Rockfish Program Tables

- `rp_cooperatives`: Annual cooperative formations
- `rp_quota_allocations`: Initial and adjusted quotas
- `rp_rockfish_landings`: Delivery records
- `rp_quota_transfers`: Inter-vessel and inter-coop transfers
- `rp_salmon_bycatch`: Chinook tracking (extrapolated vs. actual)
- `rp_vessel_caps`: Vessel harvest limits
- `rp_processor_caps`: Processor receiving limits

System/Audit Tables

- `landing_edits`: Track all data corrections
- `system_notifications`: User alerts and messages
- `email_parse_logs`: Email processing history
- `calculation_runs`: Batch calculation audit

Critical Design Decisions

1. **User/Vessel Overlap**: Single login for users who participate in both programs
2. **Temporal Data**: All permissions and allocations include `season_year` for historical tracking
3. **Salmon Duality**: Track both extrapolated (official) and actual (plant count) salmon numbers

4. **Data Corrections:** Landing edits trigger recalculations with full audit trail

Data Flow Architecture

1.1 Email Ingestion Flow

Three Path Approach:

1. **Direct from NMFS** (if permissions allow)
 - NMFS → platform email → n8n processes automatically
2. **Manager Forward** (likely scenario)
 - NMFS → Manager → forwards to platform email → n8n processes
3. **Admin Manual Upload** (backup/corrections)
 - Admin Dashboard → Manual upload → Direct database insert

Email Processing Steps:

1. n8n monitors inbox every 5 minutes
2. Validates sender (whitelist of NMFS domains or registered managers)
3. Extracts attachments (Excel/CSV)
4. Parses vessel, dates, pounds
5. Validates data completeness
6. Success: Insert to database
7. Failure: Log error and notify admin

Admin Manual Upload Flow:

- Used for: Failed emails, corrections, testing, historical imports
- Provides parsed preview before confirmation
- Logs who uploaded, when, and why
- Historical imports marked as reference-only (no invoicing)

Historical Data Import:

- Admin selects season/year
- System validates date ranges
- Marked as "historical" - no new invoices generated
- Used for trending and comparison only

1.2 Calculation Trigger Sequences

TEM Calculations (Real-time on insert):

1. New Landing Trigger:

- Database trigger fires on insert
- For vessels ≥ 60 ft: Calculate 4-trip average
- For vessels < 60 ft: Calculate calendar day totals
- Check for violations ($> 300,000$ lb average)
- Check MRA compliance
- Create violation records if needed

2. Data Correction Trigger:

- Archive original record
- Recalculate affected trip averages
- Update violation records

Rockfish Calculations (Real-time):

1. Landing Trigger:

- Update vessel/cooperative quotas
- Check vessel and processor caps
- Update salmon bycatch counts
- Flag hotspot areas

2. Transfer Trigger:

- Validate quota availability
- Update both vessels atomically
- Recalculate utilization percentages
- Calculate payment amounts

3. Allocation Trigger:

- Initialize season quotas
- Calculate percentage-based caps
- Reset utilization to zero

1.3 User Interface Data Flow

Data Flow Architecture:

- Database → Supabase APIs/Custom endpoints → Frontend
- Most calculations done at database level for consistency
- Frontend displays pre-calculated results

Refresh Strategy (Cost-Optimized):

1. TEM Program - Manual Refresh

- Trip averages change infrequently (~1-2 times/day)
- All dashboards use standard HTTP requests
- "Refresh" button for updates
- Data cached for 1 hour

2. Rockfish Program - Hybrid Approach

- Vessel quota pages: Auto-refresh every 5 minutes (critical data)
- Manager tools: Manual refresh (forms and admin tasks)
- WebSocket connections only for quota tracking
- Keeps concurrent connections under 50 (free tier)

Caching Strategy:

- TEM trip averages: 1 hour cache
- Rockfish quotas: 5 minute cache
- Violation records: 24 hour cache
- Historical data: Permanent cache

Data Aggregation:

- All calculations performed in database
- Frontend receives pre-calculated summaries
- Ensures consistency across all users
- Reduces client-side processing load

1.4 Error Handling and Retry Logic

File Storage Strategy:

- ALL files stored immediately upon receipt
- Path: /[program]/[year]/[season]/originals/

- Files never deleted, only status updated
- Complete audit trail maintained

Error Categories and Responses:

1. Network/Temporary Errors

- Retry: 3 attempts with exponential backoff (1min, 5min, 15min)
- Then: Alert admin

2. Parsing Errors

- Retry: Once with alternate parser
- Then: Queue for manual review
- Admin can trigger reprocess

3. Data Validation Errors

- No automatic retry
- Queue for admin review
- Requires human decision

4. Business Logic Errors

- Log and skip
- Notify relevant manager
- No retry needed

File Status Tracking:

- received → processing → processed/error
- Error details stored separately
- Retry count tracked
- Can reprocess from original anytime

Benefits:

- No data loss even with 100% parsing failure
- Can update parser logic and reprocess historical files
- Complete audit trail for compliance
- Simpler error recovery

1.5 Real-time Update Mechanisms

Real-time Requirements:

- TEM Program: None (all manual refresh)
- Rockfish Program: Limited to critical features only

Rockfish Real-time Features:

1. Vessel Quota Updates

- Trigger: New landing processed
- Channel: Vessel-specific only
- Update: Remaining quota by species
- UI: Progress bars animate

2. Fleet-wide Salmon Count

- Trigger: Any salmon recorded
- Channel: All active Rockfish vessels
- Update: Total count and percentage
- UI: Counter with color warnings (80% yellow, 90% red)

3. Processor Cap Warnings

- Trigger: 90% of cap reached
- Channel: Affected vessels only
- Update: Warning banner
- UI: Non-dismissible alert

Implementation:

- Supabase Realtime for WebSocket connections
- Channel subscriptions by feature
- Automatic reconnection on network issues
- Unsubscribe on page navigation
- Maximum 50 concurrent connections (free tier limit)

Connection Lifecycle:

1. Page load → Subscribe to relevant channels
2. Receive current values immediately

3. Updates flow automatically while connected
4. Page leave → Unsubscribe all channels

Wireframes and UI Design

2.1 TEM Vessel Dashboard

The primary interface for vessel operators to monitor their compliance status.

Key Elements:

- Large 4-trip average display with visual progress bar
- List of trips included in current average
- MRA compliance grid for all species
- Clear violation status indicator
- Manual refresh button with timestamp

Design Priorities:

- Immediate visual understanding of compliance status
- Mobile-responsive layout
- Minimal cognitive load for at-sea use
- High contrast for outdoor visibility

See DESIGN.md for detailed wireframes and specifications.

Technical Specifications

3.1 4-Trip Average Calculation Logic

Core Algorithm for Vessels $\geq 60\text{ft}$:

- Group trips sequentially (1-4, 5-8, etc.)
- Exclude egregious trips ($> 335,000$ lbs) completely
- Calculate average for each group
- Violations triggered when average $> 300,000$ lbs

Edge Case Decisions:

1. **Less than 4 trips:** Calculate with available trips (minimum 2 for violation)
2. **Egregious trip handling:** Excluded from all groups, gets separate violation

3. **Season boundaries:** A and B seasons calculated separately, no carryover
4. **Data corrections:** Recalculate affected groups and issue amended invoices
5. **Vessel size changes:** Keep historical calculations, apply new rules going forward

Vessels <60ft Algorithm:

- Track calendar days with multiple deliveries
- Group consecutive multi-delivery days (2 days minimum)
- Calculate average across all trips in those days
- Same 300,000 lb threshold applies

Example Scenarios:

- Trips [295k, 340k(egregious), 285k, 290k, 280k]
- Groups: [295k,285k,290k,280k] = 287.5k average ✓
- Egregious trip: Separate violation + fine

3.2 MRA Violation Detection Algorithm

Core Algorithm:

- Compare each species percentage against Table 10 CFR limits
- Calculate based on total pollock retained (trip basis)
- Only violations >\$250 in value are recorded
- Different limits for CGOA vs WGOA regions

Implementation Details:

- Species aggregation (e.g., "rockfish group" combines multiple species)
- MRA percentages stored in `tem_mra_species_lookup` table
- Federal table imported via admin interface
- Effective dates tracked for regulatory changes

3.3 Database Triggers and Functions

Automatic Calculation Triggers:

1. **Landing Insert Trigger** (`process_tem_landing`)
 - Fires after new landing data inserted
 - Routes to appropriate calculation based on vessel size

- Checks MRA violations
- Logs all calculation runs

2. Rockfish Quota Update (`update_rockfish_quotas`)

- Updates remaining quota immediately
- Checks for negative balances
- Updates vessel and processor caps
- Sends notifications for overages

3. Data Correction Handler (`handle_landing_correction`)

- Archives original values
- Triggers recalculation cascade
- Maintains full audit trail

Key Database Functions:

- `calculate_four_trip_average()`: Groups trips and calculates violations
- `check_mra_violations()`: Compares species percentages to limits
- `update_vessel_caps()`: Tracks percentage-based harvest caps

Design Principles:

- All calculations in database for consistency
- Atomic transactions prevent partial updates
- Automatic notifications for critical issues
- Performance optimized with PostgreSQL functions

3.4 Caching Strategy for Performance

Cache Layers:

1. Database: Materialized views for complex calculations
2. Application: Redis for frequently accessed data (1-24 hour TTL)
3. Browser: Service worker + LocalStorage for offline capability

Cache Invalidation:

- Time-based: Different TTLs per data type
- Event-based: Clear on new landings/transfers
- Manual: Admin force refresh option

Performance Targets:

- Cached responses: <100ms
- Database queries: <500ms
- Full page load: <3s on 3G connection

3.5 Data Validation Rules

Input Validation:

- Required fields enforced at all entry points
- Type checking (integers, dates, UUIDs)
- Range validation (reasonable min/max values)
- Pattern matching for codes (e.g., species codes)

Business Rule Validation:

- Vessel must be active participant
- Landing dates within current season
- Sufficient quota for transfers
- No cap violations
- Inter-coop transfer rules

Database Constraints:

- Positive values only for pounds/quotas
- No future-dated landings
- Referential integrity for all foreign keys
- Check constraints for business rules

Validation Strategy:

1. Client-side for immediate feedback
2. Server-side as primary enforcement
3. Database constraints as final safety net
4. Clear error messages for users