MyEyes Dashboard - Business Logic Guide

For: Department Leads, Founders, Operations Team **Purpose:** Understand what business rules and calculations the dashboard implements

---

Table of Contents

Overview (Management) Tab

What It Shows

A single-page health dashboard for leadership. Combines live metrics from Operations, Customer Care, Sales, and the Device Fleet into one view.

Health Status Indicators

Each area gets a red/yellow/green health badge:

Area	Green	Yellow	Red
Operations	0 overdue	1-3 overdue	4+ overdue devices
Customer Care	≤10% negative sentiment	10-25% negative	>25% negative
Device Fleet	0-2 underutilized	3-5 underutilized	>5 underutilized
Prescribers	Computed from prescriber analytics	—	—

Key Metrics Displayed

Always Live (not date-filtered):

- Overdue device count

- Calls received today
- Active/total prescribers, conversion rate, prescriber drop-off count
- Fleet status: total devices, active devices, underutilized count

Period Activity (date-filtered):

- Orders processed
- Total call volume and transcripts available
- Negative sentiment percentage
- After-hours inbound call count

Beverly's Sales Section

Merges Beverly's attributed orders from two sources:

1. **WooCommerce source tag:** Orders with "AT Beverly" in the Source field
2. **ActiveCampaign deal matching:** Beverly's won deals matched to WooCommerce orders by customer phone number

Shows total orders attributed to Beverly, breakdown of purchases vs. rentals, and total revenue contribution for the selected period.

Prescriber Analytics

Data Source

Prescriber data is loaded from the `rx_txns_zap` Google Sheet (pre-aggregated as `rx_data.json`). This includes Rx submissions, conversion to orders, and prescriber account groupings.

Key Definitions

- **Active Prescriber:** Last Rx submitted within the past 90 days
- **Inactive Prescriber:** Previously prescribed but no Rx in the last 90 days
- **Sample-Only Prescriber:** Ordered samples but never prescribed rentals

Metrics

- **Sample Conversion Rate:** % of samples that lead to rental orders
- **Rx-to-Order Rate:** % of prescriptions that become paid orders
- **Revenue per Prescriber:** Average revenue from prescribing doctors
- **Unconverted Rxs:** Rx submissions per prescriber that did not result in an order

Account Grouping

Prescribers are aggregated by Account (independent practice or health system). Account-level metrics include prescriber count, total orders, total Rxs, and conversion rate.

Weekly Trends

A daily chart shows Rx submissions and order conversions across the selected date range.

Prescriber Alerts

The system generates **prescriber drop-off** alerts when a prescriber who previously had 3+ orders hasn't submitted an Rx in 14-90 days — targeting re-engagement by the sales team.

Sales Tab

What It Shows

Beverly's (primary sales agent, extension 102) call activity and sales performance.

Call Transfer Detection

When a front-desk agent transfers a customer call to Beverly, Nextiva creates multiple recordings tagged with a shared `callhalf-{base-number}` identifier. The system matches transfer-to-Beverly recordings (outbound calls to extension 102) back to the original inbound customer call via this shared ID. Direct inbound calls to Beverly are also captured.

Sales Attribution

Beverly's sales are attributed through two pathways:

1. **WooCommerce source tag:** Orders explicitly marked "AT Beverly" in the Source column
2. **ActiveCampaign deal matching:** Beverly's won deals in AC matched to WooCommerce orders by customer phone number (deduplicated against WooCommerce-sourced orders)

Lead Response Tracking

Measures sales responsiveness to inbound calls:

- Tracks first customer call transferred to Beverly (deduplicated per phone number)
- Finds earliest outbound sales call to that customer after their inbound
- Calculates response time in hours
- Flags as **"slow"** if no callback or callback exceeds 24 hours

CC Contribution Analysis

Identifies customer support team involvement before Beverly's closed deals:

- Shows shared customers who both Beverly and CC agents called before the sale
- Calculates CC vs. Beverly talk-time split per customer
- Aggregate stats: what % of Beverly's customers also received CC support

Weekly Hours

Tracks daily call hours against a 20-hour weekly target with a bar chart and progress indicator.

Ambassador Team (Customer Care)

Call Statistics

- Total calls (inbound + outbound)
- Inbound vs. outbound breakdown
- Average call duration
- Calls by agent and by time of day

Missed Call Detection (Three Sources)

The system identifies genuinely missed calls from three independent sources and merges them:

1. **Nextiva Recorder:** Inbound calls under 30 seconds — briefly answered or quick hang-ups

2. **XSI API (BroadWorks):** True ring-no-answer calls from individual agent voicemail boxes
3. **Gmail Voicemail:** Hunt group/queue voicemail notifications detected via email

Deduplication:

- Phone numbers normalized across all sources
- Calls from the same phone within 5 minutes are considered duplicates
- XSI missed calls take priority (most reliable "ring-no-answer" signal)
- Calls that were actually answered (30+ seconds) are filtered out

Callback Response Tracking

For each missed call:

- Searches for a matching outbound call to the same phone number after the miss
- Falls back to inbound answered call (customer called back themselves)
- Calculates response time in hours
- **Slow callback:** No response found OR response exceeds 24 hours

Call Topic Categorization

Keyword-based tagging from call transcripts:

Topic	Keywords
Hardware Issue	device, tonometer, probe, battery, charger
Software Issue	app, login, password, download, error
Shipping/Returns	tracking, delivery, UPS, return label
Billing	charge, payment, refund, invoice, pricing
How to Use	instructions, measure, reading, tutorial
Account/Orders	order, cancel, extend, rental, subscription
Voicemail/Missed	missed call, callback, no answer

Each unique call is tagged only once per topic to avoid duplicate counting.

Call Transcription (Automated)

- Calls are automatically transcribed using OpenAI Whisper API
- Runs on startup (60s delay) and daily at 2 AM ET
- Only calls longer than 10 seconds are transcribed
- Transcripts stored permanently in Google Sheets (`call_transcripts` tab) — survives Render deploys
- Cost: ~$0.006 per minute of audio
- Viewable in the Call Log by clicking on a call row

Staffing Intelligence (Erlang C)

Uses queueing theory to predict wait probability and staffing needs:

Erlang C Calculation:

- Converts call volume + average handle time into "offered traffic" (Erlang hours)

- Predicts % of inbound calls that will wait based on how many agents are working

Risk Levels:

- **GREEN:** <20% wait probability
- **YELLOW:** 20-40% wait probability
- **RED:** >40% wait probability

Features:

- **Hourly heatmap:** 9 AM - 6 PM ET breakdown showing traffic intensity and risk per hour
- **Real-time gauge:** Current hour's wait probability based on agents actually logged in
- **1-agent vs. 2-agent scenarios:** Shows risk under different staffing levels
- **Agent coverage timeline:** Tracks single vs. multi-agent coverage periods
- **Weekly KPI trends:** Week-over-week comparison of voicemail volume, peak wait %, single-coverage hours, missed calls

Voicemail Check Tracking

- Monitors calls to VM system (remote_party = 9999, direction = OUT)
- Tracks daily VM checks and minutes per agent
- Interpretation: HEAVY (>8 min), MODERATE (2-8 min), LIGHT (<2 min)

Business Hours

Definition: 9:00 AM - 7:00 PM Eastern Time, Monday-Friday

After-hours calls include weekends, weekday evenings (after 7 PM ET), and early mornings (before 9 AM ET).

Device Utilization

Overview

The Device Utilization tab tracks individual device lifecycle and usage patterns across the entire fleet. It shows how MyEyes devices move through the rental pipeline over time.

Data Sources

Three sources are merged (deduplicated by order number):

1. **OPS Google Sheet:** Current rental orders with device serial numbers and tracking stages
2. **WooCommerce Device History:** Historical rental orders from `woo_device_history.json`
3. **Shippo Tracking:** Exact delivery/return timestamps for precision date calculations

Fleet Summary Metrics

All summary metrics use **all-time data** (not affected by date filters):

Metric	Calculation
Total Devices	Count of unique serial numbers across all rental orders
Active Now	Devices currently out: OPS stage is "With Customer" or "Transit to Customer", OR Shippo shows delivered but not returned

Avg Rentals/Mo	Fleet-wide: total rental orders (excluding extensions) / total device-months in pool
Avg Utilization	Fleet average of each device's utilization rate
Underutilized	Devices with no rental in 90+ days AND not currently with a customer

Per-Device Metrics

For each device serial number:

- **Total Rentals:** Count of rental orders (excluding extensions)
- **Unique Patients:** How many different patients used this device
- **Total Days in Use:** Sum of rental days across all orders (using Shippo dates when available)
- **Avg Days/Rental:** Total days in use / number of rentals
- **Months in Pool:** Time since first rental (in months)
- **Rentals/Month:** Total rentals / months in pool
- **Utilization Rate:** (Total days in use / days since first rental) x 100, capped at 100%
- **Current Status:** With Customer, In Transit, or Available (based on latest OPS order stage)

Days-in-Use Calculation Priority

For each rental order, the system determines how long the device was out using the best available data:

1. **Shippo tracking (most accurate):** Uses `delivered_to_customer` → `returned_to_myeyes` timestamps
2. **WooCommerce dates (fallback):** Uses `received_date` → `completed_date`
3. **OPS stage estimate (last resort):** Uses order dates and stage-based inference

All calculations are capped at a reasonable maximum (rental period + 7-14 days buffer, or 60 days) to prevent data quality issues from skewing results.

Fleet Trend Charts

Fleet Size Over Time: Monthly bar chart showing cumulative device count (how many serial numbers had entered the pool by each month) with a line overlay showing monthly delivery count.

Monthly Utilization: Line chart showing fleet utilization % per month, calculated as: (total occupied days across all devices that month) / (fleet size x days in month) x 100.

Utilization Rate Distribution

Histogram showing how many devices fall into each 10% utilization bucket (0-10%, 10-20%, ... 90-100%). Color-coded: red (<30%), orange (30-60%), green (60-100%).

Device Lifecycle Chart

Bubble chart where each bubble is one device:

- **X-axis:** Months in pool
- **Y-axis:** Total rentals
- **Bubble size:** Proportional to total days in use
- **Bubble color:** Green (≥60% util), orange (30-60%), red (<30%)

Click any bubble to open the device detail modal with full rental history, travel map, and monthly activity chart.

Date Filter Behavior

The date filter only affects the **orders shown in the device detail modal** — all top-level metrics, charts, and the device table always show all-time data. This is intentional because utilization, rentals/month, and months-in-pool are inherently long-term metrics.

Operations

Stage Cards

Orders are grouped by tracking stage. Each stage shows a count and is clickable to filter the order table:

Stage	Meaning
1. Label Created	Shipping label printed, not yet shipped
2. Transit to Customer	Package shipped, in transit
3. With Customer	Delivered to customer
4. Return Transit	Customer shipped device back
5. Returned Complete	Device received at MyEyes

Stage is derived from the latest tracking event text in the OPS Google Sheet.

Overdue Device Tracking

Devices past their grace period with no return tracking event. Shows patient name, order number, days overdue, late fees owed vs. charged, and extension history.

Pipeline Board

Visual Kanban-style board showing orders by stage, with overdue indicators and rental timing details.

WooCommerce Analytics

Overview

The WooCommerce tab pulls data directly from the WooCommerce REST API and WC Analytics API to show ecommerce performance metrics. Data is cached for 10 minutes and pre-fetched on server startup for the default 30-day range.

Revenue Metrics

- **Total Sales:** Gross revenue from completed orders
- **Net Sales:** Revenue after refunds/discounts
- **Average Order Value (AOV):** Net sales / order count
- **Period Comparison:** Each metric shows % change vs. the previous period of equal length

Order Metrics

- **Order counts by status:** Completed, Processing, Pending, Refunded, Cancelled
- **Refund Rate:** Refunded orders / total orders (color-coded: green <2%, yellow 2-5%, red >5%)
- **Order Volume Trend:** Daily orders + revenue over time

- **Hourly Heatmap:** When orders are placed (converted to Eastern Time)

Product Analytics

- **Top Products:** Ranked by items sold with revenue
- **Category Breakdown:** Products categorized as:
  - **Rentals:** SKU contains "rental", "diagnose plan", or "weekly"
  - **Purchases:** SKU contains "tonometer"
  - **Accessories:** Everything else
- Revenue split shown as horizontal bar chart with percentages

Coupon Analytics

- **Coupon Usage Rate:** % of orders using a coupon
- **Total Discount Amount:** Sum of all coupon discounts
- **Average Discount %:** Total discounts / total sales
- **Top Coupons:** Most-used coupon codes
- **Discount Impact:** Average order value with vs. without coupons

Customer Analytics

- **New vs. Returning:** Based on unique customer IDs within the selected period
- A customer who ordered once = "new", ordered 2+ times = "returning"

WooCommerce Status Filtering

Revenue and order stats filter to **completed orders only** using the `status_is[]=completed` parameter on WC Analytics endpoints.

Date Presets

In addition to the global date presets (7/30/90 days), the WooCommerce tab has:

- **This Month:** 1st of current month to today
- **Last Month:** Full previous calendar month
- **This Quarter:** 1st of current quarter to today
- **YTD:** January 1st to today

IOP Reports

What It Does

The IOP Reports tab monitors whether intraocular pressure (IOP) reports are being sent to prescribers after patients complete equipment rentals. It answers: **"Did we send the IOP data to the doctor?"**

How It Works

Step 1 — Identify Completed Rentals: A rental appears in the tab if its completion date falls within the selected date range. Completion date uses Shippo's `returned_to_myeyes` timestamp (preferred) or falls back to the scheduled end date (ship date + rental duration from SKU).

Step 2 — Find the Prescriber: For each patient, the system performs a multi-step lookup:

1. Find the patient in ActiveCampaign by email (preferred) or fuzzy name match (75%+ similarity)
2. Get the patient's deals — prioritize deals with "Prescription" in the title
3. Extract the secondary contact (prescriber) from the deal, getting their email and name

4. Fallback: Use prescriber name from WooCommerce if ActiveCampaign has "#N/A"

Step 3 — Check Gmail for IOP Report: Searches the `support@myeyes.net` Gmail for emails matching:

- Subject contains "IOP", "Data", and "MyEyes"
- Sent to OR cc'd to the prescriber's email address
- Within the rental date range

Status Indicators

- **Report Sent:** Gmail search found a matching IOP email to the prescriber
- **Not Sent:** No matching email found — needs follow-up
- **No Prescriber Email:** Could not determine the prescriber's email address

Flagging System

Team members can manually flag patients as "checked" with optional notes. Flags are saved to Google Sheets for persistence and help coordinate follow-ups across the team.

Debug Endpoints

Three diagnostic endpoints are available for troubleshooting prescriber/email matching:

- `/api/iop-tracking/debug-prescriber` — Check prescriber lookup for a patient
- `/api/iop-tracking/debug-gmail` — Check Gmail search results for a prescriber email
- `/api/iop-tracking/flag` (POST) — Save or update patient flags

Rental Timing & Due Dates

How Rental Periods Work

The rental period **starts the day AFTER the customer receives the device**, not on delivery day.

Example Timeline (7-day rental):

```
Thursday, Jan 23  - Package delivered (Day 0) - NOT counted
Friday, Jan 24    - Rental period STARTS (Day 1)
...7 days later...
Thursday, Jan 30  - Rental period ENDS (Day 7)
Friday, Jan 31    - DUE BACK by end of day
Saturday, Feb 1   - GRACE PERIOD (1 extra day to drop off)
Sunday, Feb 2     - LATE FEES START if not returned
```

Grace Period

Customers get **1 extra day** after the due date to drop off the device at a carrier (UPS, USPS, FedEx).

Rental Period Lengths

The dashboard recognizes these rental periods from the product SKU:

- **1 week** = 7 days
- **2 weeks** = 14 days
- **1 month** = 30 days

If "rental" or "compare" is in the SKU but no specific period, defaults to **14 days**.

Late Fees

Rate: $25 per day

Late fees start accruing **after the grace period ends**.

Calculation:

```
Late Fees = (Days Overdue) x $25/day
```

When Late Fees Apply

1. **Simple Overdue:** Device not returned by grace period end — fees accrue daily
2. **Extension Gap:** Customer extends after the grace period — fees owed for gap between grace end and extension date
3. **Late Return:** Device returned after grace period — fees owed for gap between grace end and return date

Late Fee Tracking

The dashboard shows:

- **Owed:** How much the customer should be charged
- **Charged:** How much has been charged via WooCommerce subscriptions
- **Gap:** The difference (what still needs to be charged)

Display Colors:

- **Red:** Not charged at all
- **Orange:** Partially charged (gap exists)
- **Green:** Fully charged (no gap)

Important: The dashboard calculates what SHOULD be charged, but **Ericka manually creates the late fee subscriptions in WooCommerce**. The dashboard just shows if they match.

Extensions

How Extensions Work

When a customer extends their rental:

1. **Fresh Start:** The extension starts on the day they ordered it — it does NOT extend from the original due date
2. **Gap Period Fees:** Late fees apply for any gap between the grace period end and the extension order date
3. **New Grace Period:** Extension gets its own 1-day grace period

Extension Detection

The dashboard automatically detects extensions by looking for:

- Product SKU contains "extension"
- Same patient name (case-insensitive match) OR same device serial number
- Ordered AFTER the original rental

Multiple Extensions

Each extension starts fresh from its order date. Late fees calculated for each gap period.

Device Returns

Return Tracking

The dashboard checks Shippo API tracking data for return events:

- **Detection:** Shippo API `/tracks/{carrier}/{tracking_number}` endpoint
- **Direction:** Return labels identified by `address_to` matching MyEyes address
- **Refresh:** Every 60 minutes via Shippo API

Return Scenarios

1. **Returned On Time:** Return event date ≤ grace period end → no late fees
2. **Returned Late:** Return event date > grace period end → late fees for gap period only
3. **Not Returned:** No return event found → overdue, fees accruing daily

---

Shipping Transit Times

What It Tracks

The Operations tab includes two transit time charts:

1. **Outbound Transit:** Days from MyEyes shipping to customer delivery
2. **Return Transit:** Days from customer drop-off to MyEyes receiving the return

Data Source

Shippo tracking events are fetched directly from the Shippo API every 60 minutes. The key date fields per order:

Field	Meaning
`shipped_to_customer`	Date MyEyes shipped outbound package
`delivered_to_customer`	Date customer received outbound package
`return_shipped`	Date customer dropped off return at carrier
`returned_to_myeyes`	Date return package arrived at MyEyes

Calculation

- **Outbound Transit Days** = `delivered_to_customer` - `shipped_to_customer`
- **Return Transit Days** = `returned_to_myeyes` - `return_shipped`

Only orders with both dates for a given direction are included. Results capped at 30 days max.

Alert System

Types of Alerts

1. **Overdue Devices** — Critical (>7 days) / Warning (1-7 days): Device past grace period and not returned
2. **After-Hours Call Volume** — Info: >3 inbound calls outside business hours (9am-7pm ET)
3. **High Negative Sentiment** — Warning: >25% of calls in past week had negative sentiment
4. **Slow Callbacks** — Warning (>7 days) / Info (1-7 days): Inbound call with no return call in >24 hours
5. **Prescriber Drop-Off** — Warning: Prescriber with 3+ orders hasn't prescribed in 14-90 days
6. **Low Conversion Prescribers** — Info: Prescriber ordered samples but no follow-up rental orders

---

Data Sources & Integrations

Primary Data Sources

Source	What It Provides	Refresh
Google Sheets (OPS)	Orders, rentals, extensions, tracking events, device serial numbers	Every 15 minutes (push from Sheets)
WooCommerce REST API	Late fee subscriptions, order details, customer stats	10-minute cache
WooCommerce Analytics API	Revenue stats, order statistics, top products, coupon data	10-minute cache
Shippo API	Package tracking, delivery/return dates, carrier info	Every 60 minutes
Nextiva Recorder API	Call recordings, metadata, agent assignments, durations	Per-request
OpenAI Whisper API	Automated call transcription	2 AM ET daily + 60s after startup

Secondary Integrations

Source	What It Provides	Refresh
BroadWorks XSI API (via Nextiva)	Missed call logs with caller ID names (CNAM) for 4 agents	Every 15 minutes
Gmail API	IOP report email search, hunt group voicemail notifications	On-demand per request
ActiveCampaign API	Contact/deal lookup for prescriber emails, Beverly's deal attribution, overdue device enrichment	On-demand per request
Google Apps Script	Persistent transcript storage (doPost handler with duplicate checking)	Fire-and-forget per transcription

Data Refresh Summary

- **Manual refresh:** Click "Refresh" button anytime
- **OPS data push:** Every 15 minutes from Google Sheets
- **Shippo tracking sync:** Every 60 minutes, 50x concurrency, ~90s total

- **Transcription batch:** 25 calls max per run, 2-day lookback, calls >10 seconds
- **XSI missed calls:** Every 15 minutes for all 4 agents
- **WooCommerce cache:** 10 minutes per date-range key
- **First load:** May take 30-60 seconds (Render cold start)

ActiveCampaign Integration

ActiveCampaign (AC) is the CRM and serves as the system of record for patient contacts, prescriber associations, and sales pipeline tracking. The dashboard uses the AC API in four main ways:

1. Contact Name Resolution

On startup, the dashboard fetches **all AC contacts** and builds a phone-number-to-contact map (`contactMap`). This powers caller identification across every tab — when a phone number appears in a call log, missed call, after-hours alert, or sales timeline, the system looks up the AC contact by normalized phone number to display the customer's name. If no AC match is found, it falls back to a `customerMap` built from prior call recordings.

2. Beverly's Sales Attribution

The dashboard fetches all of Beverly's **won deals** from the AC pipeline and matches them to WooCommerce orders by customer phone number. This appears in two places:

- **Overview tab:** Beverly's attributed orders, purchase/rental split, and revenue
- **Sales tab:** Deal-attributed orders merged with WooCommerce source-tagged orders (deduplicated)

Deal resolution uses a `contactIdMap` (AC contact ID to contact info) built during the contact cache refresh, so deal contacts can be resolved without additional API calls.

3. Prescriber Lookup (IOP Reports)

For the IOP Reports tab, the system uses AC to find the prescriber associated with a patient's rental:

1. Search AC for the patient by email or fuzzy name match
2. Get the patient's deals — prioritize deals with "Prescription" in the title
3. Extract the **secondary contact** on the deal (the prescriber) to get their email and name
4. Fallback: check deal custom field #11 for a prescriber email (used in older deals)

The prescriber email is then used to search Gmail for IOP report delivery confirmation.

4. Overdue Device Enrichment

For overdue devices in the Operations tab, the system calls `getLastContactActivity()` to fetch the most recent deal stage and deal note for a patient. AC deal stages reflect actual patient communications (e.g., "Sent Rental Expiration Reminder"), providing context for whether follow-up has already occurred.

Refresh & Caching

- **Full contact sync:** On startup and every 60 minutes (hourly cache refresh)
- **Beverly deals:** Fetched alongside contact sync
- **Prescriber/deal lookups:** On-demand per IOP report request
- **Rate limiting:** 100ms delay between paginated API calls to avoid AC rate limits

---

Common Questions

Q: Why does a customer show overdue when they just extended?

A: Two possible reasons:

1. Extension order placed too long after due date — they owe late fees for the gap
2. Dashboard hasn't refreshed yet — wait up to 15 minutes or click "Refresh"

Q: Why don't I see late fee info for all overdue devices?

A: Late fees only show if the customer is actually overdue (past grace period) AND either has an extension with a gap period or is currently overdue.

Q: How do I know if a late fee was actually charged?

A: Look at the late fee display:

- **Green (Gap: $0)** = Fully charged
- **Orange (Gap: $X)** = Partially charged
- **Red (Not charged)** = Not charged at all

Verify in WooCommerce > Subscriptions for the actual subscription.

Q: Can the dashboard automatically charge late fees?

A: No. The dashboard only CALCULATES late fees. Ericka must manually create the late fee subscriptions in WooCommerce.

Q: Why does the dashboard show "0" data when I first load it?

A: Render free tier puts the server to sleep after inactivity. The first load takes 30-60 seconds to wake up.

Q: What's the difference between "due date" and "grace period end"?

A:

- **Due date:** Last day of rental period (when they SHOULD return it)
- **Grace period end:** 1 day later (when late fees START)

Q: How are extensions matched to original rentals?

A: By patient name (case-insensitive) OR device serial number, where the extension order date is after the original rental.

Q: Why does the Operations "With Customer" count differ from Device Utilization "Active Now"?

A: Operations counts **orders** — every rental order that currently has stage "3. With Customer." Device Utilization counts **unique devices** by serial number, checking only the latest order per device. Old orders with stale tracking data inflate the Operations count; the Device Utilization "Active Now" is more accurate for "how many physical devices are currently out."

Q: Why doesn't the IOP tab show a prescriber email for some patients?

A: The prescriber lookup requires the patient to have an ActiveCampaign deal with a secondary contact (the prescriber). If the deal doesn't exist, or the prescriber wasn't added as a contact, the lookup fails. Use the

debug endpoint to troubleshoot.

Business Rules Summary

Rule	Value	Notes
Rental start	Day after delivery	Delivery day not counted
Grace period	1 day	After due date
Late fee rate	$25/day	After grace period
Extension start	Day after extension order	Fresh start, not from due date
Extension gap	Charged late fees	Gap not forgiven
Return check	Shippo API	Address-based direction detection
Transit time cap	30 days max	Higher values excluded as data errors
Active Now	OPS stage + Shippo	"With Customer" or "Transit to Customer" or Shippo delivered without return
Underutilized threshold	90 days	No rental in 90+ days and not currently out
Utilization cap	100% max	Per-device, prevents data quality skew
Slow callback	>24 hours	Or no callback at all
Missed call dedup window	5 minutes	Same phone, any source
Missed call min duration	30 seconds	Shorter = missed, longer = answered
Active prescriber	90 days	Last Rx within 90 days
Prescriber drop-off	14-90 days	No Rx in 14-90 days, had 3+ orders
Erlang C risk: Green	<20% wait	Low staffing risk
Erlang C risk: Yellow	20-40% wait	Medium staffing risk
Erlang C risk: Red	>40% wait	High staffing risk
WooCommerce cache	10 minutes	Per date-range key
WooCommerce status	Completed only	Revenue/order stats filter to completed
Data refresh (OPS)	15 minutes	Push from Google Sheets
Data refresh (Shippo)	60 minutes	50x concurrency, ~90s total

Data refresh (XSI)	15 minutes	All 4 agents' missed calls
Transcription batch	25 calls max	Per run, manages costs
Transcription lookback	7 days	How far back to check
Min transcription duration	10 seconds	Shorter calls skipped
Transcript storage	Google Sheets	Persistent across deploys
Business hours	9am-7pm ET, Mon-Fri	For after-hours alerts
Refund rate threshold	Green <2%, Yellow 2-5%, Red >5%	Color-coded
VM check: Heavy	>8 min	Daily voicemail minutes per agent
VM check: Moderate	2-8 min	Daily voicemail minutes per agent
VM check: Light	<2 min	Daily voicemail minutes per agent

Support & Questions

Dashboard access: https://myeyes-ops.onrender.com **Technical issues:** Meghan (meghan@myeyes.net)
Business logic questions or requests: Meghan