

Complete NetOS Documentation

Downloadable reference generated from the NetOS Markdown documentation.

Documentation Overview

Xiber NetOS — Documentation

Network lifecycle management for wholesale circuits, facilities, and provider operations.

Guides

#	GUIDE	WHAT'S INSIDE
1	User Manual	Views, navigation, filters, bulk actions, circuit/infrastructure detail, customers, RF Links, Electrical, import workflow, dashboard KPIs
2	Admin & Operations	Dev stack, Docker services, config, migrations, seed data, RBAC, audit logging, PDF docs, troubleshooting
3	Data & Import Guide	Data model, system-of-record boundaries, CSV/XLSX import, templates, natural keys, validation rules
4	API & Integrations	REST endpoints, bulk APIs, customers, RF Links, Electrical, infrastructure attribution, MCP tools, external system integration map
5	Public Hostname & SSO	Cloudflare Tunnel setup, Microsoft Entra OIDC, OAuth2-Proxy, production deployment
6	Authorization & RBAC	Internal authorization, granular permissions, field-level controls, scopes, bootstrap admin, route protection
7	Infrastructure Attribution	Revenue and expense attribution model for towers, rooftops, data centers, POPs, electrical add-on costs, subtended links, diagrams, and waterfall reporting
8	Roadmap & Known Gaps	Feature inventory, gap analysis by area, recommended build phases

Downloadable PDFs

PDF versions are published under `/docs/*.pdf` for sharing with the team.

PDF	LINK
Complete documentation bundle	/docs/netos-documentation.pdf
User Manual	/docs/user-manual.pdf

PDF	LINK
Admin & Operations	/docs/admin-operations.pdf
Data & Import Guide	/docs/data-import-guide.pdf
API & Integrations	/docs/api-integrations.pdf
Public Hostname & SSO	/docs/public-hostname-ss0.pdf
Authorization & RBAC	/docs/authorization-rbac.pdf
Infrastructure Attribution	/docs/infrastructure-attribution.pdf
Roadmap & Known Gaps	/docs/roadmap-known-gaps.pdf

Current State

Implemented Views

VIEW	URL	STATUS
Circuits	/circuits	Complete, including bulk edit/delete
RF Links	/rf-links	Complete, including RF costs and FCC license upload
Electrical	/electrical-services	Complete, including site attachment and supporting documents
Customers	/customers	Complete, including commercial and MFC customer records
Circuit Detail	/circuits/{id}	Complete
Map	/map	Complete
Dashboard	/dashboard	Complete (modeled MRR)
Timeline	/timeline	Complete
Topology	/topology	Complete
Service Providers	/providers	Complete
Infrastructure	/infrastructure	Complete, including bulk edit/delete
CSV/XLSX Import	/import	Complete
Financial / P&L	/financial	Complete (modeled MRR)
Address Lookup	/address	Complete
Mobile Field View	/mobile	Complete
Admin Queue / Audit	/admin	Complete
Exports	/exports	Placeholder

Implemented Backend

- FastAPI with OpenAPI 3.1 auto-docs
- PostgreSQL 16 schema with PostGIS geometry and pgvector embeddings
- SQLAlchemy 2.0 async models with Alembic migrations
- Circuit CRUD, bulk edit/delete, site-backed circuit creation, map, dashboard summary, import preview/commit
- Customer CRUD for commercial and MFC endpoints
- RF Link CRUD with RF system presets, FCC/supporting documents, CAPEX, MRC/rent, depreciation inputs, and automatic customer endlink attribution
- Electrical service CRUD with infrastructure/customer site attachment, provider, meter/service details, average monthly cost, monitoring link, and supporting documents
- Infrastructure CRUD, bulk edit/delete, supporting documents, subtended links, upstream paths, and economics rollups
- Role-based access control (development shim)
- User activity audit logging and bug/feature feedback queue
- MCP server skeleton (2 working tools, 2 placeholders)

Key Gaps

- Authentication is a header-based dev shim (needs Entra OIDC)
- Financial data uses modeled MRR ($MRC * 1.35$) outside infrastructure attribution, not real Sonar revenue
- PDF contract ingestion schema exists but extraction is not implemented
- Renewal engine computes states on-the-fly, no persistent job or notifications
- Audit logging exists for key writes, bulk actions, feedback, and request activity; field-level read redaction remains incomplete
- Celery workers are wired but not active

See [Roadmap](#) for full gap analysis and build order.

Local URLs

SERVICE	URL
Web UI	<code>http://localhost:3000</code>
API	<code>http://localhost:8000</code>
API Docs (Swagger)	<code>http://localhost:8000/docs</code>
API Health	<code>http://localhost:8000/healthz</code>
PostgreSQL	<code>localhost:5432</code>
Redis	<code>localhost:6379</code>

User Manual

Xiber NetOS — User Manual

Daily usage guide for the NetOS web interface.

What NetOS Does

NetOS is Xiber's system of record for **wholesale network lifecycle management**. It tracks:

- Carrier circuits (waves, EPL, DIA, dark fiber, cross-connects, etc.)
- Infrastructure assets (data centers, towers, colos, rooftops, carrier hotels)
- Carrier/provider relationships and contacts
- Contract terms, renewal windows, and ETF exposure
- Lifecycle history (orders, installs, upgrades, outages, renewals)

What it does NOT do: end-customer billing (Sonar), device monitoring (LibreNMS), or network design (Wisdm).

Navigation

The left sidebar contains all views:

VIEW	URL	DESCRIPTION
Circuits	<code>/circuits</code>	Main circuit list with search, filters, bulk edit, and soft-delete
RF Links	<code>/rf-links</code>	Wireless link inventory with equipment, spectrum, cost, rent, endpoints, and FCC license documents
Electrical	<code>/electrical-services</code>	Utility service inventory attached to infrastructure or customer sites
Customers	<code>/customers</code>	Commercial and MFC customer records used by circuits, RF Links, and subtended endpoints
Map	<code>/map</code>	Geographic view of circuits and infrastructure
Dashboard	<code>/dashboard</code>	KPIs, spend analysis, renewal pipeline
Alerts	<code>/alerts</code>	Problem inventory records that need cleanup before maps, topology, attribution, or reporting can be trusted
Timeline	<code>/timeline</code>	Contract-term Gantt chart
Topology	<code>/topology</code>	Endpoint relationship graph with SPOF detection
Providers	<code>/providers</code>	Carrier database with contacts and spend
Infrastructure	<code>/infrastructure</code>	Non-circuit assets (towers, colos, DCs)

VIEW	URL	DESCRIPTION
Financial	/financial	Per-circuit P&L, modeled revenue, margin, ETF exposure, and carrier rollups
Address	/address	Address-based circuit lookup
Mobile	/mobile	Field-friendly read view with NOC call, copy ID, and map links
Import	/import	CSV/XLSX bulk import
Admin	/admin	Audit activity plus bug and feature request queue
Exports	/exports	Filtered data export *(planned)*

A **global search bar** appears across views. It searches across inventory, not just the current page, and returns recommended matches as you type.

Global search supports partial matches for:

- Circuits by label, carrier circuit ID, service type, status, BAN, billing account, notes, provider, and A/Z site names or geography
- RF Links by link label, status, equipment, spectrum, FCC call sign, A/Z sites, customer endpoint, and notes
- Infrastructure by site name, Xiber label, site type, account, address, provider, and notes
- Customers by customer name, property name, customer type, service/access technology, address, contacts, billing account, and notes
- Electrical services by service name, provider, account, meter, delivery location, address, status, and notes
- Providers by provider name, short name, contacts, support/billing details, and ASN

Use the recommended matches dropdown to jump directly to the matching record or inventory page. Page-level filters still apply only to the page currently being viewed.

Circuits

URL: </circuits>

The primary operational list. Columns include:

COLUMN	DESCRIPTION
Xiber Circuit Label	Internal Xiber identifier (click to open detail)
Carrier Circuit ID	The carrier's reference number
Carrier	Upstream provider name
Service Type	wave, epl, evpl, dia, broadband, ip_transit, dark_fiber, etc.
Bandwidth	Capacity in Mbps
Status	quoted, ordered, installing, active, suspended, decommissioned
A Endpoint	Near-end location
Z Endpoint	Far-end location

COLUMN	DESCRIPTION
MRC	Monthly recurring cost
NRC	Non-recurring cost
Term Start / End	Contract period

Click any Xiber circuit label to open the circuit detail page.

The circuit list is grouped by service type. Each service type section can be collapsed so large inventories remain navigable while preserving the existing filters and bulk-selection workflow.

Add or Edit Circuit

The circuit form now uses existing Infrastructure and Customer records as site selectors. Select the A site and, for multi-site services such as EVPL, EPL, wave, dark fiber, MPLS, or cross-connect, select the Z site. NetOS copies the saved site name, address, type, and coordinates into the circuit endpoint record so maps and topology stay consistent with the site inventory.

Circuit labels are generated from the standard pattern `Provider Short Name Service Type A Loc -> Z Loc` when a provider short name exists, otherwise `Carrier Service Type A Loc -> Z Loc`. The add/edit form starts with service type because that selection controls which site fields are needed. For single-site services such as DIA, broadband, and Starlink, the generated label omits the Z location. Users can edit the generated label before saving, but changing carrier, service type, or selected sites refreshes the generated label so circuit names stay consistent.

DIA, broadband, and Starlink are single-site services in NetOS, so the form only asks for one circuit site. If the selected site is a customer record, NetOS also associates that customer to the circuit. Circuit-specific install details such as meet-me room, rack/demarc position, demarc type, site contact, and circuit access notes can still be entered on the circuit because those details may differ from the base site record.

Create or update the Infrastructure or Customer site first when the location, address, or coordinates are missing or wrong. The circuit form should not be used to maintain master site addresses.

Circuit Bulk Actions

Use the checkboxes in the first column to select one or more circuits. The header checkbox selects all currently visible filtered rows.

ACTION	WHAT IT DOES
Apply	Bulk-updates selected circuits using only the fields filled into the bulk action row
Delete	Soft-deletes selected circuits after confirmation

Bulk update fields currently include status, provider, service type, MRC, NRC, term end, and notes. Bulk deletes are soft-deletes: the rows are hidden from normal views but remain in the database for audit and historical reporting.

Each affected circuit receives its own audit entry.

RF Links

URL: </rf-links>

Use RF Links for PTP, PtMP, Tarana, microwave, and other wireless paths that serve customer endpoints or downstream assets from a POP, tower, rooftop, or facility.

Tracked fields include equipment type, path role, diversity group, frequency, channel width, bandwidth, FCC call sign, license required status, CAPEX, straight-line depreciation inputs, MRC, tower or rooftop rent MRC, install date, A location, Z location, coordinates, notes, and FCC license uploads. Equipment presets can populate common equipment type, link type, default CAPEX, and useful life; users can still edit those values before saving.

RF Links use the same A/Z direction convention as circuits: A is the customer or downstream side, and Z is closer to the donor/core. For customer PtP service, select the customer endpoint and the Z donor facility. For facility-to-facility backhaul, select both A and Z infrastructure facilities.

RF Links can be attached from an infrastructure asset's **Subtended Network** section. When a customer endpoint is served over wireless transport, select the RF link in **RF link serving endpoint**. NetOS uses RF link MRC plus tower rent MRC as calculated monthly link cost unless a manual link cost override is entered.

When adding PtMP customer endlinks, select the PtMP source node first. NetOS shows only mapped customer sites within 10 miles of that sector and only subscriber radios compatible with the selected base/sector system. Select one or more customer endpoints to create multiple PtMP RF Links in one action.

The RF Link list is grouped by BTA market state, BTA, and then city, using the A infrastructure location first and falling back to the Z facility or customer endpoint geography when needed. Cross-state suburbs are consolidated into the parent metro market, such as Newport, KY under OH / Cincinnati BTA. State, BTA, and city sections can be collapsed.

Customers

URL: </customers>

Customers are first-class records that can be selected from endpoint-style circuits, RF Links, and infrastructure subtended links.

Commercial customer fields include customer name, status, address, coordinates, DIA or PTP service type, DIA/PTP capacity, contracted MRR, install NRC, revenue source, OTT service checkboxes for VoIP, IPTV, managed WiFi, and managed security, contact details, billing account, and notes.

MFC customer fields include property name, tenant count, address, coordinates, contracted MRR, install NRC, property CAPEX, maintenance reserve MRC, revenue source, access technology such as XGSPON, EoC, or ActiveE, property owner, property manager, access rules, contact details, billing account, and notes.

MFC records also include agreement and property-service detail fields. The **MFC Service Agreement** section tracks the agreement signed date, renewal terms, owner termination rights, system ownership terms, and force majeure language. The **MFC Service Scope** section tracks base and maximum in-unit speeds, whether video/IPTV is bulk,

retail, both, or not provided, common-area WiFi, mobile carrier offload, IoT coverage, property management office voice, and free-form notes for common-area or other property services.

Use customer selectors instead of free-form customer names when the customer already exists. Free-form customer names remain available for early discovery or incomplete records.

The customer table also shows associated circuits, RF Links, donor POPs/facilities, and protection status so users can see how the customer is served and click through to the serving asset where applicable. Serving paths are sorted with primary paths first and backup, secondary, redundant, or diverse paths underneath so the operating path is always visible before the protection path. MFC sites are flagged as **Protected** when NetOS sees either at least one primary serving path plus at least one backup path, or two or more primary serving paths that represent ring/diverse protection. Backup paths include Broadband, Starlink, cellular backup, circuits with purpose `backup`, or paths explicitly marked secondary, backup, redundant, or diverse. If DIA is the only service at an MFC property, NetOS treats that DIA as the primary path until a Starlink or other backup path is added. Unprotected MFCs are highlighted because the operating standard is to add backup or ring/diverse service where feasible.

The customer table is grouped by customer type, BTA market state, BTA, and city so commercial and MFC sites can be reviewed by both service model and metro market.

Customer add/edit forms warn when **Address line 1** appears to contain the city, state, or ZIP. Split the address into Address line 1, City, State, and Postal code fields so geocoding can create customer coordinates. Customer records are also flagged as unmappable when they have address data but no latitude/longitude after geocoding. Correct the address fields or enter coordinates manually; customers without coordinates cannot render as RF Link endpoints on the map.

When typing a customer or infrastructure site address, NetOS recommends normalized address matches. Choosing a match fills Address line 1, City, State, Postal code, Country, Latitude, and Longitude. Save the record after choosing the suggested address so maps, topology, and RF Link rendering update from the corrected location.

Attribution uses subtended link revenue when entered. If the subtended link has a selected customer and no explicit link revenue, NetOS uses the customer's contracted MRR.

Customer detail pages show attached electrical services alongside serving paths, related objects, a monthly revenue schedule, a collapsible ROI Analysis model, inherited donor cost detail, and a **What If** calculator. The serving path section combines explicit subtended links with directly attached customer circuits and RF Links, then labels each path as **Primary** or **Backup** so Broadband, DIA, Starlink, and secondary paths are visible in the same operating view as the main RF or fiber path. DIA is contextual for MFC sites: when it is the only serving path it is labeled **Primary**, and when another primary path exists it may remain a backup or secondary path if marked that way. Direct customer circuits are limited to customer-facing circuit endpoints or one-ended customer backup services; upstream donor transport that serves a POP, tower, or aggregation facility remains visible under inherited donor cost instead of being listed as a direct customer serving path. Inherited donor cost includes the customer's revenue pro rata share of shared facility MRC and upstream transport paths at each structure in the serving chain. The financial summary breaks out direct serving cost, inherited shared cost, maintenance reserve, commission, revenue escalator, cost escalator, and ramp assumptions. The monthly schedule shows MRR, escalated serving cost, maintenance reserve, commission, total cost, and margin by month. The calculator lets users compare the customer revenue schedule against a hypothetical Type 2 circuit, alternate RF build, or renewal case by entering monthly transport cost, build CAPEX, monthly OPEX/reserve, install NRC, and commission. The ROI model uses customer MRR, term years, term start/end dates, revenue escalator, ramp months, commission percentage or fixed monthly commission, attributed monthly serving cost, install NRC, RF Link CAPEX when present, customer property CAPEX, maintenance reserve, and 3% annual direct/inherited cost escalation starting in year 2 to project ARR, annual contribution margin,

CAPEX/ARR, margin payback, net cash, ROI percentage, IRR, NPV, margin multiple, and payback over 2-, 3-, 5-, and 10-year terms.

Customer detail pages also include **Customer Site Photos & Documents**. Users can upload field photos, screenshots, floorplans, permits, invoices, service agreements, and other supporting documents. Image uploads capture available metadata such as image dimensions, EXIF captured date/time, GPS coordinates, camera/software hints, and file size. When the AI provider is configured, NetOS analyzes images for visible equipment, site conditions, access or safety concerns, detected text, and recommended follow-up. AI analysis is advisory and should be reviewed by the user before operational decisions are made.

The customer list is grouped first by customer type, then by state, then by city. Each level can be collapsed to navigate large commercial and MFC inventories.

Alerts

URL: `/alerts`

Alerts collect records that need user cleanup. This is the first place to check when maps, topology, or attribution do not look correct.

Current alert categories include:

- Customers or infrastructure sites with address data but no coordinates
- Customers or infrastructure sites with no mappable address or coordinates
- Address line 1 values that appear to contain city, state, or ZIP data
- Circuits missing required endpoints
- RF Links missing an A site or Z site/customer endpoint
- MFC customers missing contracted MRR

Click an alert row to open the affected record, correct the data, and save it. The alert disappears once the record no longer matches the alert rule.

Circuit Detail

URL: `/circuits/{circuit_id}`

The single-circuit deep dive. Sections include:

Circuit Attributes

- Carrier circuit ID, carrier, service type, status
- Bandwidth, handoff type, purpose (backhaul, backup, DIA, customer dedicated, etc.)
- MRC, NRC, PO number, GL code

Contract & Renewal

- Contract term dates, renewal type (auto/manual/evergreen)

- Renewal notice window, decision deadline
- ETF formula and current estimated ETF value
- Price escalator details

Endpoints

- A and Z endpoint names, addresses, coordinates copied from selected infrastructure or customer sites
- Endpoint type derived from the selected site, such as tower, data center, POP, customer premise, or carrier hotel

Provider Info

- NOC phone number and escalation procedures
- Account manager contact
- Provider portal link (click to open carrier's management portal)
- Monitoring link editor for the exact LibreNMS, Grafana, or other monitoring URL
- Monitoring open button; circuits use a carrier circuit ID search link until an exact URL is saved

Documents & History

- Agreement upload/view panel
- Notes field
- Lifecycle event timeline (orders, installs, bandwidth changes, outages, renewals)

When to use: Validating a circuit before renewal, checking NOC escalation info during an outage, preparing a termination, or reviewing contract terms.

Import

URL: `/import`

Bulk-load circuits, facilities, towers, rooftops, and data centers from CSV or XLSX files.

Import Workflow

STEP	ACTION
1	Choose Circuits or Infrastructure
2	NetOS auto-detects column mappings
3	Review the first 5 rows and mapped fields
4	Click Commit to import valid rows
5	Existing circuits are updated by carrier + carrier circuit ID; infrastructure is updated by Xiber infrastructure label
6	Missing providers and circuit endpoints are auto-created as stubs

Circuit Minimum Required Fields

- `carrier_name` — the upstream provider
- `carrier_circuit_id` — the carrier's reference number

Circuit Recommended Fields

`xiber_circuit_label`, `service_type`, `bandwidth_mbps`, `status`, `install_date`, `mrc_usd`, `nrc_usd`, `ban_id`, `billing_account`, `protocol_handoff`, `monitoring_url`, `contract_effective_date`, `contract_term_end_date`, `contract_initial_term_months`, `contract_renewal_type`, `contract_msa_terms_text`, `contract_etf_formula_text`, `a_endpoint_name`, `a_latitude`, `a_longitude`, `z_endpoint_name`, `z_latitude`, `z_longitude`

For manual UI entry, prefer selecting existing Infrastructure or Customer sites instead of retyping endpoint addresses. The endpoint import fields remain supported for bulk onboarding and legacy records.

Infrastructure Required Fields

- `display_name` — facility, tower, rooftop, data center, POP, office, or aggregation site name
- `site_type` — use infrastructure site types such as `data_center`, `tower`, `rooftop`, `colo`, `pop`, `office`, or `other`

Templates are available directly on `/import`: **Circuit CSV** and **Infrastructure CSV**.

Electrical utility services are not infrastructure assets. Add them from **Electrical** after the infrastructure or customer site exists.

See [Data & Import Guide](#) for detailed field specifications, validation rules, and API examples.

AI Assistant

NetOS includes an AI assistant drawer for users with the `ai_assistant_use` permission. The assistant is a draft helper, not an autonomous writer. Paste carrier notes, quote details, install notes, or other source text into the drawer, then prepare a draft for the active workflow.

The assistant supports draft creation for customers, infrastructure assets, RF links, and circuits. Select the draft type in the drawer, paste source material or instructions, and prepare a draft. Click **Apply to active form** to populate the matching active add/edit form, then review and save normally. The normal API validation, RBAC checks, field-level permissions, and audit logging still apply.

Recommended intake order is customer or infrastructure first, RF links next, and circuits last. Customers and infrastructure assets become selectable sites/endpoints for RF links and circuits, so creating them first produces cleaner circuit records.

The Import page also supports AI free-form import for customers, circuits, and infrastructure. Paste multiple records into the AI import box, preferably separated by blank lines, and NetOS will create staged import rows. These rows still go through the normal import preview, validation, row selection, and commit workflow before any records are created or updated.

For customer imports, the Import page can also use the configured customer research connector. Enter a property name, address hint, website, or rough notes in **Customer site research**. NetOS calls the server-side Sonar/MCP connector, stages one customer import row, and includes research notes/citations in the preview data when available. This is an enrichment aid only; the user must review and commit the customer row before anything is saved.

The assistant also accepts PDF documents such as carrier quotes, service agreements, or order forms. Upload the PDF, add any instructions or questions, and prepare a draft. NetOS extracts selectable PDF text first and falls back to OCR when needed. The response includes page-level source snippets for extracted agreement fields and a short list of questions for missing or ambiguous values. PDF uploads are temporary for assistant drafting and do not create or attach documents to inventory records unless the user later saves through the normal forms.

Cost fields are omitted from assistant drafts when the current user cannot view circuit cost fields. The assistant reports missing required fields so the user can complete the form before saving.

When configured with an AI provider, the assistant uses durable NetOS memory entries for explicit user preferences and organization rules. Memory is stored in NetOS, not in the browser, and is only created from explicit instructions such as "remember..." or "always...". If no provider key is configured or the provider is unavailable, the assistant falls back to the deterministic parser.

Map

URL: `/map`

Geographic visualization of the network portfolio.

What's shown:

- **Circuit paths** — lines from A endpoint to Z endpoint
- **RF Link paths** — wireless links from infrastructure assets to infrastructure or customer sites when both ends have coordinates
- **Endpoint markers** — dots at circuit termination points
- **Infrastructure markers** — facility icons for data centers, towers, carrier hotels, etc.
- **Customer markers** — customer sites with saved or geocoded coordinates
- **Color coding** — lines colored by carrier/provider
- **Line thickness** — scaled by bandwidth

Interactions:

- Click a circuit line → popup with circuit summary and link to detail page
- Click an endpoint → popup with connected circuits
- Click an infrastructure marker → popup with facility details

Map data refreshes automatically after circuit, infrastructure, customer, RF Link, import, or subtended network changes made in the same browser session or another open NetOS tab. The map also refreshes when the tab regains focus. Customer and infrastructure addresses are geocoded when address fields are saved without manual latitude/longitude values; RF Links to customer endpoints render once the customer site has coordinates.

If an RF Link cannot be drawn, the map displays a coordinate warning above the map. The most common reason is a customer endpoint with no latitude/longitude after geocoding. Correct the customer address fields or enter coordinates manually, save the customer, and refresh the map.

Filters:

- Link source: Circuits, RF Links, or both
- Provider/carrier
- Link type: circuit service types and RF link types
- Facility type (data center, tower, colo, etc.)
- Status (active, ordered, decommissioned)
- Location / state
- Protected or unprotected customer sites
- Minimum and maximum link length in miles

Map filters support multi-select include and exclude behavior. Open a filter, choose one or more values, switch between **Include** and **Exclude**, or use **Only** beside a value to isolate it immediately.

Dashboard

URL: /dashboard

Portfolio-level analytics and KPIs.

Dashboard data refreshes automatically after circuit, infrastructure, customer, RF Link, electrical service, or subtended network changes made in the same browser session or another open NetOS tab. It also refreshes when the tab regains focus.

KPI Cards

CARD	DESCRIPTION
Total Circuits	All circuits in the system
Active Circuits	Circuits with <code>active</code> status
Infrastructure Assets	Non-circuit facility agreements
Circuit MRC	Total monthly circuit spend
Facility MRC	Total monthly infrastructure spend
Total Network MRC	Circuit + facility combined
Modeled MRR	Estimated monthly revenue*
Modeled Margin	Estimated margin percentage*
Renewal Window	Circuits within renewal decision window

**Modeled MRR currently uses `MRC * 1.35` as a demonstration multiplier. Production MRR will come from Sonar revenue attribution.*

Panels

PANEL	DESCRIPTION
Spend by Provider	Bar chart of MRC by carrier
Service Mix	Circuit count by service type
Margin Distribution	Histogram of per-circuit margins
Renewal Pipeline	Circuits grouped by renewal state
Endpoint State Mix	Geographic distribution
At-Risk Circuits	Circuits with critical renewal state or low margin
At-Risk Infrastructure	Facilities approaching renewal deadline
Top 10 Expensive	Highest MRC circuits
Top Facility Costs	Highest MRC infrastructure
Lowest Margin	Circuits with worst margin performance

Timeline

URL: </timeline>

A contract-term Gantt chart for both circuits and infrastructure agreements.

What's shown:

- Horizontal bars representing contract start → end dates
- Install date markers
- Renewal decision deadline markers
- Today line (vertical red line)
- Color-coded by renewal state:
- **Green** — stable (>180 days to deadline)
- **Yellow** — watch (90–180 days)
- **Orange** — active (30–90 days)
- **Red** — critical (<30 days)
- **Dark red** — overdue (past deadline)
- **Blue** — decided

Timeline rows can be grouped by vendor, type, or status. Each group header is collapsible and shows item count plus MRC rollup so large renewal timelines can be scanned by vendor exposure or asset/service type.

Filters:

- Group by: carrier, service type, or status
- Carrier
- Type
- Renewal state

Click a circuit or infrastructure row to open its detail page.

Topology

URL: `/topology`

Network relationship graph showing how sites connect through circuits and RF Links.

Visual encoding:

- **Nodes** = canonical sites (sized by connection count, colored by type)
- **Edges** = circuits (colored by carrier, weighted by bandwidth)
- **Dotted blue/teal edges** = RF Links, grouped by link type
- **Solid edges** = circuits. Circuit service or carrier differences are shown by color, not dash pattern
- **M / C markers** = MFC and commercial customer sites
- **Curved edges** = parallel circuits between the same sites
- **Dashed green node ring** = protected customer site, such as RF plus a wireline circuit, two primary ring/diverse paths, or multiple customer-serving circuits where one is Broadband, DIA, secondary, or backup
- **Infrastructure nodes** = facility assets that circuit endpoints are linked to or matched by normalized site name

Topology is site-centric. When circuit endpoints are linked to Infrastructure or Customer records, NetOS renders one node for the site and draws all related circuit edges from that node. Older imported endpoint records are also matched to infrastructure assets by normalized site name when possible, so repeated endpoint rows such as multiple BSI Huntington circuit endpoints collapse into one larger site node with multiple logical links. Customer A-side circuit endpoints are also matched to customer records by selected customer, customer name, or property name so RF and wireline paths terminate on one customer site icon. Single-ended customer backup circuits, such as Broadband or Starlink records with only an A endpoint, are attached to the customer node for redundancy detection even when no topology edge can be drawn. DIA-only MFC sites are treated as primary until an additional backup path is present. Circuits with purpose `backup` are also treated as backup serving paths for customer protection.

SPOF Detection: The side panel highlights **Single Point of Failure candidates** — sites with only one network connection and no detected redundancy. Automatic protection detection is intentionally customer-scoped: aggregation, POP, data center, tower, or rooftop sites are not considered ring-protected merely because they have one upstream path and many downstream RF/customer links. Customer sites with RF plus wireline, two primary ring/diverse paths, or multiple customer-serving circuits where one is Broadband, DIA, secondary, or backup, are treated as protected and are not counted as SPOF candidates.

Layout modes:

- **Logical** — layered hierarchy with upstream/core sites higher and subtended aggregation/customer sites lower. Circuit A/Z direction is used as the primary hierarchy signal: A-side sites are treated as downstream and Z-side sites as upstream/donor where the circuit data follows the NetOS naming standard.
- **Geographic** — nodes positioned by approximate endpoint coordinates

Logical view uses a wide canvas and parent-grouped spreading so subtended sites stay visually under their donor structure. For example, BSI Huntington Center customer endpoints are grouped under BSI Huntington Center, while BSI US Bank Cincinnati customer endpoints are grouped under BSI US Bank Cincinnati instead of being mixed into one flat customer row. When a site has more than one upstream connection, NetOS chooses a primary visual parent

for placement while still drawing the additional network links. Parents with larger downstream subtrees receive wider horizontal space, and each subtree receives a minimum lateral lane so adjacent donor groupings do not collapse into each other. Dense leaf groups wrap into multiple rows under the same parent instead of growing into one long horizontal row. If a downstream structure also has its own downstream customers, NetOS places that structure on the lower row beneath its donor before laying out its child endpoints, reducing crossed paths from intermediate aggregation sites such as Lamplighter. RF and circuit hierarchy direction is inferred from site rank and A/Z direction so core or donor sites remain above customer or subtended sites. After hierarchy placement, NetOS runs collision avoidance within each visual parent group and slightly staggers same-axis vertical chains so stacked elements do not appear to share a single straight-line path. Labels scale up modestly while zooming and use a white stroke behind the text, and parallel circuits between the same two sites are curved apart so redundant paths do not fully hide each other.

Interactions:

- Click a node → see attached circuits and RF Links
- Click an edge → see circuit or RF Link details and link to the related page

Topology filters use the same multi-select include/exclude pattern as the map. Use them to isolate or exclude link sources, providers, circuit/RF link types, statuses, facility types, protected or unprotected customer sites, and links within a minimum/maximum mile range.

Topology data refreshes automatically after network-shape changes to circuits, infrastructure assets, RF Links, customers, imports, or subtended links. Use the Refresh button when validating changes made by another user or process outside the current browser session.

Service Providers

URL: `/providers`

The carrier and provider database.

What's tracked per provider:

- Customer portal URL
- Label short name used for generated circuit labels
- NOC phone number
- Account manager (name, email, phone)
- Billing and legal contacts
- Escalation procedures
- Circuit count and total circuit spend
- Infrastructure count and total facility spend
- ETF exposure across all related contracts

Provider detail pages include tables of related circuits and related infrastructure assets.

Use **Add Service Provider** to create a provider record. Use **Edit** from a provider detail page to maintain portal URLs, label short names, NOC phone numbers, account manager contacts, legal/billing contacts, escalation paths, ASN, W-9 URL, and preferred payment method.

Infrastructure

URL: `/infrastructure`

Non-circuit network assets and facility agreements.

Asset types:

- Data centers
- Carrier hotels
- Colocation cabinets/cages
- Tower leases
- Rooftop rights
- Facility access agreements

What's tracked:

- Provider and facility name
- Monthly recurring cost (MRC) and non-recurring cost (NRC)
- Rack, cabinet, space, and power commitment details that are part of the facility agreement
- Contract term start/end dates
- Renewal notice window
- Agreement document links
- Monitoring URL for the exact facility, tower, rooftop, or data center monitoring page
- Access procedures and contact info

Electrical utility accounts are tracked separately under **Electrical** so utility service can attach to either an infrastructure site or a customer site without turning the utility account itself into a structure.

Infrastructure detail pages show an **Electrical Services** section when utility services are attached to that site. The section lists provider, service type, account number, meter number, electrical characteristics, delivery location, average monthly cost, monitoring link, and uploaded service documents.

Infrastructure detail pages also include **Site Documents** for photos, screenshots, floorplans, permits, invoices, service agreements, and other supporting documents. Image uploads capture available metadata such as dimensions, EXIF captured date/time, GPS coordinates, camera/software hints, and file size. When the AI provider is configured, NetOS analyzes images for visible equipment, site conditions, access or safety concerns, detected text, and recommended follow-up. If AI is not configured or unavailable, the photo still uploads and metadata is retained.

The infrastructure list is grouped by BTA market state, BTA, and then city. BTA groups place smaller cities into the larger metro market they operate around, such as Columbus, Cincinnati, Dayton, Lexington, or Indianapolis. Cross-state suburbs are consolidated into the parent metro market, such as Newport, KY under OH / Cincinnati BTA. Each state, BTA, or city group can be collapsed, and group headers show asset count and monthly recurring cost rollup.

Subtended links and economics:

- Customer endpoints served by the structure
- Child facilities subtended by the structure, such as towers fed by a data center or aggregation site
- Related circuits used as transport for the subtended relationship

- Direct revenue, direct link cost, facility cost, downstream revenue, downstream cost, total margin, and margin percentage
- Effective dates for relationship changes, so historical attribution can be preserved

The Subtended Network section has three add modes:

MODE	USE WHEN
Attach existing circuit	A circuit already exists in NetOS and should be associated with the current infrastructure asset. Select the circuit; NetOS creates the relationship to this site and calculates link cost from the circuit MRC when possible.
Add customer endpoint	The site directly serves a revenue-producing customer, subscriber, building, or endpoint. Enter customer/endpoint name and revenue MRR.
Attach downstream structure	Another tower, rooftop, POP, data center, or facility rolls up to this asset. Select the downstream structure and optionally link the transport circuit.

Existing subtended network rows can be edited inline. Use the edit action to change the customer/endpoint label, downstream structure, related circuit, relationship type, transport type, direct link allocation method, inherited cost allocation method, effective dates, revenue, cost override, or notes. The list shows inherited cost attribution and the calculated inherited cost applied to each row. The inherited cost balance shows the shared monthly cost pool, assigned inherited cost, fixed/manual assigned cost, calculated assigned cost, and any unallocated balance. Manual percentage and direct inherited allocations consume the pool first; calculated methods assign only the remaining pool. If multiple fixed rows request more than the available pool, the row most recently changed keeps precedence and older fixed rows are capped to the remaining balance. If a manual adjustment leaves a balance, use Assign Remainder Pro-Rata or Assign Remainder Equally to save explicit manual percentages on the other calculated rows. The form shows calculated previews for direct link cost and a live inherited cost calculator that updates after each form change. The inherited calculator compares revenue pro-rata, equal share, manual percent, and direct/full assignment before the user saves. It uses the parent site's shared inherited cost pool, including upstream donor-chain cost after applying upstream allocation rules. For downstream structures, the calculator uses the downstream rollup revenue as the revenue basis. Parallel transport rows to the same downstream structure or customer count that revenue target once for inherited allocation. If a future direct allocation method such as bandwidth share is selected before its driver fields are available, NetOS uses the full linked source MRC rather than dropping the path cost to zero. Inherited cost allocation defaults to revenue pro-rata. Use delete only when the relationship was created by mistake; for real historical relationship changes, prefer setting an effective end date and creating the new relationship.

For redundant downstream connectivity, add the same downstream structure more than once from the same parent, with a distinct related circuit or RF Link for each path. NetOS keeps the downstream structure's revenue and facility cost counted once while summing the cost of each active transport path. The same downstream structure still cannot be actively attributed under two different parent sites at the same time unless the existing relationship is ended.

Use the infrastructure detail page to model how revenue and expense roll through towers, rooftops, data centers, POPs, and aggregation facilities. See [Infrastructure Revenue & Expense Attribution](#) for the full attribution model, examples, and waterfall reporting approach.

Financial fields on circuit and infrastructure forms include contextual documentation links. Use those links when deciding whether a value belongs as circuit cost, direct facility cost, direct subtended-link revenue, calculated shared transport cost, or a manual override.

Infrastructure Bulk Actions

Use the checkboxes in the first column to select one or more infrastructure assets. The header checkbox selects all visible rows.

ACTION	WHAT IT DOES
Apply	Bulk-updates selected assets using only the filled fields
Delete	Soft-deletes selected assets after confirmation

Bulk update fields currently include status, provider, site type, MRC, escalator %, term end, and notes.

Electrical

URL: `/electrical-services`

Electrical services are utility cost items attached to an existing infrastructure asset or customer site. Use this section for commercial power, house power, dedicated meters, generator service, solar, battery backup, or other site electrical obligations.

Tracked fields include service provider, attached infrastructure or customer site, account number, meter number, service type, voltage, amperage, phase, delivery location, service address, average monthly cost, start date, monitoring URL, and notes.

Documents can be uploaded per service, including sample bills, invoices, service agreements, meter photos, and supporting documents. These documents help explain the recurring utility cost used in site and customer financial views.

Electrical service cost should be treated as an add-on cost to the site it serves. Do not create a separate infrastructure asset for an electrical utility account unless the physical asset is actually a facility, POP, tower, rooftop, office, or data center.

Attached electrical services appear on the related Infrastructure or Customer detail page so users can see utility services with other related circuits, RF Links, serving paths, and documents.

Financial / P&L

URL: `/financial`

The financial view provides portfolio spend and modeled P&L analysis.

Financial data refreshes automatically after circuit, infrastructure, customer, RF Link, electrical service, or subtended network changes made in the same browser session or another open NetOS tab. It also refreshes when the tab regains focus.

What's shown:

- Wholesale MRC
- Modeled MRR
- Monthly margin
- ETF exposure
- Carrier spend rollup
- Per-circuit P&L table
- Hierarchical attribution drilldown with row-level revenue and expense flow detail

In the hierarchical attribution drilldown, expand a row to see where the economics come from. Revenue items are labeled **Rolls Up** when customer or downstream structure revenue contributes upward to the serving structure. Expense items are labeled **Flows Down** when transport or path cost is assigned from a donor or parent relationship into a downstream structure or endpoint. Direct facility cost is labeled **Local** because it belongs to the structure itself.

Current limitation: Modeled MRR uses a sample `1.35x` multiplier until Sonar revenue attribution is connected.

Address Lookup

URL: `/address`

Search for circuits by address, endpoint name, city, state, carrier, or circuit ID.

What's shown:

- Matching circuit count
 - Carriers serving matched locations
 - States represented in the match set
 - Matched MRC
 - A/Z endpoint address summaries
-

Mobile / Field

URL: `/mobile`

A phone-friendly circuit lookup for field teams.

Actions:

- Search by circuit, carrier, city, or status
 - Tap **NOC** to open the phone dialer when a provider NOC number exists
 - Tap **Copy ID** to copy the carrier circuit ID
 - Tap **Map** to open directions from endpoint coordinates or address
-

Admin Queue

URL: `/admin`

The Admin page contains the user activity audit trail and the bug/feature queue.

Bug & Feature Queue

Users submit requests from the feedback form. Admins can manage each request with:

FIELD	VALUES
Priority	<code>low</code> , <code>normal</code> , <code>high</code> , <code>urgent</code>
Status	<code>new</code> , <code>triaged</code> , <code>planned</code> , <code>in_progress</code> , <code>done</code> , <code>declined</code>

Each request has a **Progress** thread. Add updates when work starts, when a fix ships, or when more information is needed. Public progress comments attempt to email the requester when SMTP is configured.

Audit Trail

The Recent Activity table shows user, role, event, entity, changed fields, IP address, and timestamp. Times are displayed in Eastern time with the timezone label. Until SSO is fully enabled, direct demo traffic falls back to `dev@xiber.com`.

Planned Views

These routes or capabilities are planned but not complete:

VIEW	PLANNED FUNCTIONALITY
<code>Exports</code>	Filtered CSV, XLSX, and PDF export of any view's data

Roles & Permissions

ROLE	CAN VIEW	CAN EDIT
<code>exec</code>	Everything	Everything
<code>finance</code>	Everything	Financial fields, contracts
<code>network_eng</code>	Everything	Technical fields, endpoints
<code>operations</code>	Everything	Operational fields, lifecycle events
<code>pm</code>	Everything	Project-related fields
<code>sales</code>	Everything	Sales-related fields

ROLE	CAN VIEW	CAN EDIT
read_only	Everything	Nothing
agent	Filtered by MCP tool scope	Via confirmed MCP write tools

Note: The current dev build uses a header-based auth shim. Production will enforce roles via Microsoft Entra group membership.

Admin & Operations

Xiber NetOS — Admin & Operations Manual

Development setup, services, configuration, migrations, roles, and troubleshooting.

Tech Stack

COMPONENT	TECHNOLOGY	NOTES
Database	PostgreSQL 16	PostGIS (geospatial), pgvector (embeddings)
Backend	FastAPI + SQLAlchemy 2.0	Async via asyncpg, Pydantic v2 validation
Migrations	Alembic	Baseline + infrastructure + agreements + audit/feedback versions
Frontend	Next.js 14 + React 18	TypeScript 5.5, TailwindCSS, App Router
Maps	MapLibre GL	Vector tile rendering
Queue	Celery + Redis	Wired but workers not yet active
MCP	Python FastMCP	Skeleton for XOS agent tools
Dev Deployment	Docker Compose	Hot-reload for both API and web

Prerequisites

- Docker Engine 24+ with Docker Compose v2
- 4 GB RAM minimum
- Ports 3000, 5432, 6379, 8000 available
- (Optional) Python 3.12+ and Node.js 20+ for local development outside Docker

Repository Layout

```
circuitos/
├── apps/
│   ├── api/ FastAPI backend
│   │   ├── app/
│   │   │   ├── main.py Entry point
│   │   │   ├── api/v1/ Route handlers
│   │   │   ├── core/ Auth, config
│   │   │   ├── db/ Session management
│   │   │   ├── models/ SQLAlchemy entities + enums
│   │   │   └── schemas/ Pydantic request/response models
└──
```

```

├── services/      Business logic (import, extraction)
├── workers/      Celery task definitions
├── alembic/      Migration versions
├── scripts/      Seed data, utilities
├── pyproject.toml Python dependencies
├── Dockerfile
├── web/          Next.js frontend
├── app/          App Router pages
├── components/   React components
├── lib/          API client, utilities
├── package.json
├── Dockerfile
├── mcp/          MCP server skeleton
├── app/server.py
├── pyproject.toml
├── packages/
├── shared/      TypeScript type contracts
├── infra/
├── docker/
├── docker-compose.yml      Local dev stack
├── docker-compose.public.yml Production profile (SSO + tunnel)
├── nginx/                  Public reverse proxy config
├── postgres/               Custom PostgreSQL Dockerfile
├── docs/                   Documentation (you are here)
├── examples/               Sample CSV data
├── .env.example            Environment variable template

```

Local Development

Start Everything

```
cd infra/docker
docker compose up --build
```

Run Migrations

```
docker compose exec api alembic upgrade head
```

Seed Sample Data

```
docker compose exec -T -e PYTHONPATH=/app api python scripts/seed_sample_data.py
```

Creates: 4 carriers (Lumen, Zayo, Cogent, Hurricane Electric), 10 endpoints with coordinates, 8 circuits with contracts, and lifecycle events. The script is idempotent — safe to run multiple times.

Restart Services

```
docker compose restart api web
```

Rebuild After Dependency Changes

```
docker compose up --build api web
```

View Logs

```
docker compose logs -f api      # API logs
docker compose logs -f web      # Frontend logs
docker compose logs -f postgres # Database logs
```

Services

SERVICE	URL	DOCKER NAME
Web UI	<code>http://localhost:3000</code>	<code>web</code>
API	<code>http://localhost:8000</code>	<code>api</code>
API Docs (Swagger)	<code>http://localhost:8000/docs</code>	<code>api</code>
API Health	<code>http://localhost:8000/healthz</code>	<code>api</code>
PostgreSQL	<code>localhost:5432</code>	<code>postgres</code>
Redis	<code>localhost:6379</code>	<code>redis</code>

Server Deployment

The current demo deployment runs on the Xiber NetOS server.

SERVICE	URL
Public UI	<code>https://circuitos.xiberian.net</code>
Direct UI	<code>http://172.98.130.238:3010</code>
Direct API Docs	<code>http://172.98.130.238:8010/docs</code>

Server path:

```
/home/stephen/netos
```

Server compose file:

```
infra/docker/docker-compose.server.yml
```

The server uses legacy Compose v1:

```
docker-compose -p netos -f docker-compose.server.yml ps
docker-compose -p netos -f docker-compose.server.yml logs -f api
docker-compose -p netos -f docker-compose.server.yml logs -f web
```

After code changes, rebuild and restart API/web:

```
cd ~/netos/infra/docker
docker-compose -p netos -f docker-compose.server.yml build api web
docker rm -f netos_api_1 netos_web_1 || true
docker-compose -p netos -f docker-compose.server.yml up -d api web
```

The server web service runs `next start` from a built image. Do not mount `apps/web` or `.next` over the production container; only `docs/` is mounted read-only at `/app/content/docs`.

For fast frontend iteration, run a separate local preview service instead of putting the public web container into development mode:

```
cd ~/netos
scripts/start-web-preview.sh
```

The preview service runs `next dev` with the web source bind-mounted. It is available locally at `http://127.0.0.1:3011/preview` and through the protected public proxy at `https://netos.xiberian.net/preview`. When a set of changes is ready for the public app, promote it with:

```
cd ~/netos
scripts/deploy-prod-web.sh
```

Run migrations on the server:

```
cd ~/netos/infra/docker
docker-compose -p netos -f docker-compose.server.yml run --rm api alembic upgrade head
```

Configuration

RF System Standards

RF system standards are maintained from the Admin page. They define the selectable defaults used when creating RF Links.

FIELD	PURPOSE
Name	System label used in generated RF Link names and equipment defaults
Description	Short operating note or intended use case
Frequency Used	Human-readable spectrum band, such as <code>CBRS / 5 GHz</code> or <code>11 GHz licensed</code>
Capacity Mbps	Default capacity applied to new RF Links
Cost	Default CAPEX applied to new RF Links
Type	PTP or PtMP
Role	Standalone, PtMP Base/Sector, or PtMP Subscriber
Compatible Bases	For PtMP subscriber radios, the base/sector systems that can serve that subscriber
Active	Controls whether the system appears as a selectable option for new RF Links

Use **PtMP Base/Sector** for equipment installed on an infrastructure asset as a PtMP node, such as Tarana BN, Siklu Terragraph DN, or Cambium ePMP AP. Use **PtMP Subscriber** for customer-side radios, such as Tarana RN/RNv, Siklu T280/T265, or Cambium Force 4600c. When a user creates a PtMP RF Link from a selected PtMP source node, NetOS shows only subscriber radios whose compatible base list includes that source node's base system.

RF Links no longer require a manually entered link name. NetOS generates the display name from the selected A facility, Z facility or customer endpoint, and RF system/equipment. Links between two infrastructure assets are classified as backhaul. Links from an infrastructure asset to a customer endpoint are classified as customer endlinks.

Environment Variables

VARIABLE	DEFAULT	DESCRIPTION
<code>DATABASE_URL</code>	<code>postgresql+asyncpg:// circuitos:circuitos@postgres:5432/ circuitos</code>	Async database connection string
<code>REDIS_URL</code>	<code>redis://redis:6379/0</code>	Redis connection for Celery broker
<code>ENTRA_TENANT_ID</code>	<code>common</code>	Microsoft Entra tenant (dev placeholder)
<code>ENTRA_CLIENT_ID</code>	<code>dev-client-id</code>	Entra app client ID (dev placeholder)
<code>JWT_AUDIENCE</code>	<code>api://circuitos</code>	Expected JWT audience claim
<code>API_BASE_URL</code>	<code>http://api:8000</code>	Server-side API URL (Docker internal)
<code>NEXT_PUBLIC_API_BASE_URL</code>	*(empty)*	Browser-side API URL (falls back to current host:8000)
<code>AI_PROVIDER</code>	<code>deterministic</code>	Bootstrap provider until Admin > AI settings are saved. Supported app providers are <code>openai</code> , <code>minimax</code> , and <code>deterministic</code>
<code>OPENAI_API_KEY</code>	*(empty)*	Bootstrap API key used by the backend assistant service; never expose this to the browser
<code>OPENAI_MODEL</code>	<code>gpt-5.2</code>	

VARIABLE	DEFAULT	DESCRIPTION
		Bootstrap model used for assistant extraction
<code>OPENAI_BASE_URL</code>	<code>https://api.openai.com/v1</code>	Bootstrap OpenAI-compatible base URL
<code>SMTP_HOST</code>	*(empty)*	Bootstrap SMTP host used until Admin notification settings are saved. If unset and no Admin setting exists, scheduled renewal reports are skipped
<code>SMTP_PORT</code>	<code>587</code>	SMTP port
<code>SMTP_USERNAME</code>	*(empty)*	Optional SMTP username
<code>SMTP_PASSWORD</code>	*(empty)*	Optional SMTP password
<code>SMTP_FROM</code>	<code>netos@xiber.com</code>	Bootstrap from address for system email
<code>NETOS_PUBLIC_BASE_URL</code>	<code>https://netos.xiberian.net</code>	Bootstrap public URL used to generate report links back into NetOS
<code>RENEWAL_REPORT_ENABLED</code>	<code>true</code>	Bootstrap toggle for the weekly renewal email scheduler
<code>RENEWAL_REPORT_WEEKDAY</code>	<code>0</code>	Bootstrap UTC weekday for the scheduled report, where Monday is <code>0</code>
<code>RENEWAL_REPORT_HOUR</code>	<code>8</code>	Bootstrap UTC hour after which the scheduled report can send
<code>RENEWAL_REPORT_LOOKAHEAD_DAYS</code>	<code>180</code>	Bootstrap renewal decision lookahead window included in the report

Copy `.env.example` to `.env` and adjust as needed. For production SSO, see [Public Hostname & SSO](#).

AI Assistant Provider

The NetOS Assistant uses `/api/v1/ai/extract`. With no provider configured it uses the deterministic parser. Admin users should normally configure the live provider from **Admin > AI**, where API keys are encrypted server-side and only the last four characters are shown after save.

Supported AI providers:

- **OpenAI**: default base URL `https://api.openai.com/v1`, using the Responses API with strict JSON output.
- **MiniMax**: default base URL `https://api.minimax.io/v1`, default model `MiniMax-M3`, using the OpenAI-compatible Chat Completions API for text/image tasks.
- **Fallback parser**: deterministic extraction with no external model.

Use **Test** in Admin > AI after entering a key. The test calls the provider's OpenAI-compatible `/models` endpoint and verifies that the selected model is advertised before users rely on the provider.

The assistant never writes inventory directly. It returns draft payloads that users apply to forms and save through the normal NetOS APIs. Durable assistant memory is stored in `ai_memory_entries` and is limited to explicit user or organization preferences/rules, such as lines beginning with "remember..." or "always...".

The Admin > AI page also includes a **Customer Research Connector** for Sonar/MCP-style enrichment. Configure the MCP URL, tool name, optional authorization token, and enable the connector. NetOS calls this connector only from the backend, then converts the research result into a normal customer import preview row. Users must still review, edit, select, and commit the staged row through the standard import workflow.

The connector supports either an MCP SSE URL such as `http://mcp.xiberian.net:8081/sse?agent_id=netos-customer-research` or a direct HTTP endpoint that accepts MCP JSON-RPC `tools/call` requests. For SSE, NetOS opens the stream, reads the server-provided message endpoint, initializes the MCP session, and then calls the configured tool. The configured tool should accept a `query` argument and return either structured JSON with `draft`, `citations`, `warnings`, and `summary`, or text containing that JSON. Browser clients never receive the connector credential. If the MCP server expects `Authorization: <token>` rather than `Authorization: Bearer <token>`, paste only the raw token in the API key field; NetOS sends the value exactly as entered.

Use the **Test** button after entering connector values. The test initializes the MCP session and runs `tools/list`; it does not perform customer research or create import rows. A passing test means NetOS can reach the MCP server and the configured tool name appears in the advertised tool list.

Weekly Renewal Email Report

Admin users with `settings_view` / `settings_edit` can manage delivery from **Admin > Notifications**. This page controls:

- SMTP host, port, username, password, and sender
- public NetOS base URL used in email links
- weekly report enablement, day, UTC hour, and lookahead window
- test email delivery before saving or sending reports

Saved Admin notification settings override environment variables. Environment variables remain useful as bootstrap defaults and for deployments that have not yet saved a database-backed notification setting. SMTP passwords are encrypted server-side and are never returned to the browser; the UI only shows whether a password exists and its last four characters.

The Timeline page includes a **Weekly Renewal Email Report** section. It previews and manually sends an HTML report for circuits and infrastructure agreements whose renewal decision deadline is overdue or within the configured lookahead window. Each report includes:

- urgency state (`watch`, `active`, `critical`, `overdue`)
- agreement/vendor/type/status details
- MRC, term end, renewal notice, and renewal term
- links back to the circuit or infrastructure detail
- links to available agreement documents
- an HTML timeline-style snapshot for the report items

The API process also runs a lightweight scheduler. When SMTP is configured and the renewal report is enabled, it checks every 30 minutes and sends once on the configured UTC weekday/hour to all active registered users. Sends are recorded in `user_activity_logs` with `event_type=report`, so a container restart does not send the same report date twice.

Docker Network

- Server-side web calls (Next.js SSR) use `API_BASE_URL=http://api:8000` (Docker internal DNS)
- Browser-side calls use the UI host origin on port 8000

Authentication & RBAC

Current State (Development)

A header-based shim accepts the `x-user-role` header on all API requests:

```
curl -H "x-user-role: exec" http://localhost:8000/api/v1/circuits
```

No real identity verification occurs in direct/dev mode. All requests are trusted. When no SSO identity header is present, audit logs fall back to `dev@xiber.com` instead of `anonymous`.

Accepted identity headers include:

- `x-user-email`
- `x-auth-request-email`
- `x-forwarded-email`
- `x-forwarded-user`
- `cf-access-authenticated-user-email`

Production Target

Microsoft Entra OIDC JWT validation. Roles derived from Entra group membership. See [Roadmap](#) for implementation details.

Roles

ROLE	DESCRIPTION	WRITE ACCESS
<code>exec</code>	Executive — full access	Yes
<code>finance</code>	Finance team — financial fields, contracts	Yes
<code>network_eng</code>	Network engineering — technical fields, endpoints	Yes
<code>operations</code>	Operations — operational fields, lifecycle events	Yes
<code>pm</code>	Project management — project-related fields	Yes
<code>sales</code>	Sales — sales-related fields	Yes
<code>read_only</code>	View only	No
<code>agent</code>	MCP/XOS agent — scoped by tool definitions	Limited

Database

Custom PostgreSQL Image

Built from `pgvector/pgvector:pg16` with PostGIS installed.

Dockerfile: `infra/docker/postgres/Dockerfile`

Core Tables

TABLE	PURPOSE
<code>users</code>	User identity and role
<code>carriers</code>	Carrier/provider companies
<code>service_type_catalog</code>	Editable circuit service type catalog
<code>rf_system_catalog</code>	Editable RF equipment/system presets
<code>customers</code>	Commercial and MFC customer sites used by circuits, RF Links, and attribution
<code>endpoints</code>	Physical locations with PostGIS geometry
<code>circuits</code>	Wholesale circuit records
<code>rf_links</code>	Wireless backhaul and customer endlink records
<code>rf_link_documents</code>	FCC licenses and supporting RF Link documents
<code>infrastructure_assets</code>	Towers, rooftops, data centers, POPs, carrier hotels, offices, and aggregation sites
<code>infrastructure_documents</code>	Site agreements, floorplans, invoices, and supporting infrastructure documents
<code>subtended_links</code>	Explicit infrastructure/customer/circuit/RF attribution hierarchy
<code>electrical_services</code>	Electrical utility service accounts attached to infrastructure or customer sites
<code>electrical_service_documents</code>	Bills, invoices, agreements, meter photos, and supporting electrical service documents
<code>contracts</code>	Contract terms, renewal windows, ETF
<code>customer_attributions</code>	Revenue attribution (future Sonar link)
<code>lifecycle_events</code>	Timeline events (order, install, outage, etc.)
<code>invoices</code>	Carrier invoices
<code>invoice_line_items</code>	Invoice detail lines
<code>renewal_decisions</code>	Renewal state machine records
<code>outage_events</code>	Outage tracking
<code>sla_credits</code>	SLA credit recovery records
<code>import_jobs</code>	CSV/XLSX import metadata
<code>import_staging_rows</code>	Staged import rows before commit
<code>user_activity_logs</code>	Request/write audit activity

TABLE	PURPOSE
<code>feedback_requests</code>	Bug and feature requests
<code>feedback_comments</code>	Progress comments on bug/feature requests

Materialized View

- `circuit_financial_mv` — pre-computed financial rollups for dashboard performance

Documentation PDFs

Markdown documentation is the source of truth. Downloadable PDFs are generated from the Markdown files and published from `apps/web/public/docs`.

Run the exporter after material documentation changes:

```
python3 scripts/build_doc_pdfs.py
```

Generated files include:

PDF	PUBLIC URL
Complete bundle	<code>/docs/netos-documentation.pdf</code>
User Manual	<code>/docs/user-manual.pdf</code>
Admin & Operations	<code>/docs/admin-operations.pdf</code>
Data & Import Guide	<code>/docs/data-import-guide.pdf</code>
API & Integrations	<code>/docs/api-integrations.pdf</code>
Public Hostname & SSO	<code>/docs/public-hostname-sso.pdf</code>
Infrastructure Attribution	<code>/docs/infrastructure-attribution.pdf</code>
Roadmap & Known Gaps	<code>/docs/roadmap-known-gaps.pdf</code>

The exporter embeds public documentation images such as the infrastructure attribution and waterfall diagrams.

Extensions

- **PostGIS** — geometry columns for endpoint coordinates, GeoJSON queries
- **pgvector** — vector embeddings (installed, not yet used)

Migrations

Migration Files

Located in `apps/api/alembic/versions/`:

MIGRATION	DESCRIPTION
<code>0001_foundation_schema.py</code>	Core tables, enums, constraints
<code>0002_infrastructure_assets.py</code>	Infrastructure asset tables
<code>0003_agreement_extractions.py</code>	Agreement extraction tables
<code>0004_audit_feedback.py</code>	User activity logs and feedback requests
<code>0005_interface_requested_fields.py</code>	BAN, agreement, escalator, MSA/ETF fields
<code>0006_feedback_comments.py</code>	Progress comments for feedback requests

Commands

Apply all migrations:

```
# Via Docker
docker compose exec api alembic upgrade head

# Or standalone
cd apps/api
alembic upgrade head
```

Check current version:

```
docker compose exec api alembic current
```

Create a new migration:

```
docker compose exec api alembic revision --autogenerate -m "description"
```

Review auto-generated migrations before applying — they may need manual adjustments for PostGIS columns or custom types.

Seed Data

Script: `apps/api/scripts/seed_sample_data.py`

What it creates:

ENTITY	COUNT	EXAMPLES
Carriers	4	Lumen, Zayo, Cogent, Hurricane Electric
Endpoints	10	With coordinates for map rendering
Circuits	8	Various service types and statuses
Contracts	8	With term dates and renewal windows
Lifecycle Events	~15	Orders, installs, bandwidth changes

The script uses upsert logic on natural keys — safe to run repeatedly without creating duplicates.

Audit Trail

Audit data is available in the Admin view at `/admin`.

What Is Logged

EVENT TYPE	EXAMPLES
Request activity	API path, method, user, role, IP address, response status
Record writes	Create/update circuit, infrastructure, provider
Bulk actions	Per-record entries for bulk circuit and infrastructure updates/deletes
Feedback	Bug/feature request creation, status/priority changes, progress comments

Timestamps are stored in UTC in PostgreSQL and displayed in the UI as Eastern time with the timezone label.

Soft Deletes

Circuit and infrastructure bulk deletes set `deleted_at`. They do not physically remove rows. Normal list/map/dashboard views filter out soft-deleted rows.

Bug & Feature Queue

The Admin view includes a feedback form and queue.

FIELD	VALUES
Priority	<code>low</code> , <code>normal</code> , <code>high</code> , <code>urgent</code>
Status	<code>new</code> , <code>triaged</code> , <code>planned</code> , <code>in_progress</code> , <code>done</code> , <code>declined</code>

Admins can add progress comments to each request. Public comments attempt to email the requester when SMTP is configured. SMTP failures are non-blocking, so feedback updates still save if email is unavailable.

SMTP configuration is normally managed from **Admin > Notifications**. The variables below are bootstrap fallbacks before Admin settings are saved:

VARIABLE	DESCRIPTION
SMTP_HOST	SMTP server hostname
SMTP_PORT	SMTP port, default 587
SMTP_USERNAME	Optional SMTP username
SMTP_PASSWORD	Optional SMTP password
SMTP_FROM	From address, default netos@xiber.com
ADMIN_EMAIL	Admin notification recipient

Verification Commands

Check API Health

```
curl -fsS http://localhost:8000/healthz  
# → {"status": "ok"}
```

Check Map Data

```
curl -fsS http://localhost:8000/api/v1/circuits/map | python3 -m json.tool | head -20
```

Check Dashboard

```
curl -fsS http://localhost:8000/api/v1/dashboard/summary | python3 -m json.tool | head -20
```

Compile Check (Python)

```
python3 -m compileall apps/api/app apps/api/scripts apps/mcp/app
```

Build Check (Next.js)

```
cd apps/web && npm run build
```

Troubleshooting

Map Doesn't Load

CHECK	SOLUTION
API reachable?	<code>curl http://localhost:8000/api/v1/circuits/map</code>
CORS?	Ensure <code>http://localhost:3000</code> is in allowed origins
Data exists?	Run seed script, verify circuits have endpoint coordinates
Browser cache?	Hard refresh (Cmd+Shift+R / Ctrl+Shift+R)
Containers running?	<code>docker compose ps</code> — restart <code>api</code> and <code>web</code> if needed

Frontend 500 on Server-Rendered Pages

- Verify `API_BASE_URL=http://api:8000` is set in Docker Compose
- Verify the `api` container is running: `docker compose logs api`
- Check for Python exceptions in API logs

No Sample Circuits Showing

- Run the seed script (see above)
- Verify: `curl http://localhost:8000/api/v1/circuits` returns data
- Check migrations: `docker compose exec api alembic current`

PostGIS Extension Errors

- Rebuild the custom PostgreSQL image: `docker compose build postgres`
- Verify `infra/docker/postgres/Dockerfile` is referenced in `docker-compose.yml`
- Check: `docker compose exec postgres psql -U circuitos -c "SELECT PostGIS_Version();"`

Container Won't Start

```
docker compose logs <service> # Check for errors
docker compose down -v # Nuclear option: remove volumes and rebuild
docker compose up --build
```

Warning: `down -v` destroys all data. Re-run migrations and seed script after.

Data & Import Guide

Xiber NetOS — Data & Import Guide

Data model, system-of-record boundaries, CSV/XLSX import workflow, and validation rules.

System of Record Boundaries

What NetOS Owns

DOMAIN	DETAILS
Carriers	Provider names, portal URLs, NOC contacts, account managers
Endpoints	Location names, addresses, coordinates, facility types
Circuits	Technical attributes, bandwidth, status, purpose, handoff
Financials	MRC, NRC, PO numbers, GL codes
Contracts	Terms, renewal windows, ETF formulas, escalators, governing law
Renewals	Renewal state machine, decision history
Lifecycle	Event timeline (orders, installs, upgrades, outages, decomms)
Invoices	Carrier invoices and variance detection (future)
Infrastructure	Towers, rooftops, data centers, POPs, carrier hotels, offices, aggregation sites, facility costs, rack/power commitments, and tower leases
Customers	Commercial and MFC customer sites used as customer endpoints for circuits, RF Links, and attribution
RF Links	Wireless backhaul and customer endlinks with RF system, capacity, CAPEX, MRC/rent, frequency, FCC license, and depreciation fields
Electrical Services	Utility accounts attached to infrastructure or customer sites, including provider, meter, service details, average monthly cost, monitoring link, and documents
Infrastructure Attribution	Explicit subtended relationships between parent sites, downstream structures, customer endpoints, circuits, and RF Links

What NetOS Does NOT Own

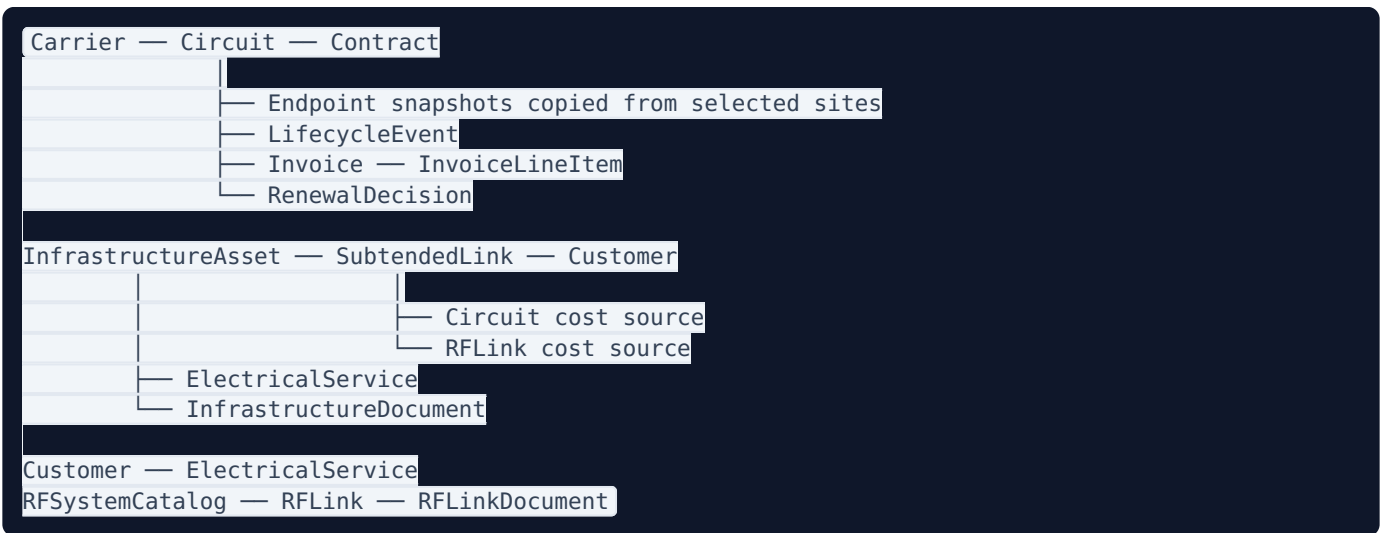
DOMAIN	SOURCE SYSTEM
End-customer billing	Sonar
Device telemetry & monitoring	LibreNMS / Prometheus / Loki
Physical network design	Wisdm
Document signatures	DocuSeal

DOMAIN	SOURCE SYSTEM
Project task management	Monday.com

NetOS will **pull from** these systems (Sonar revenue, LibreNMS monitoring links, Wisdm geometry) and **push to** them (Monday.com renewal tasks, DocuSeal termination notices) — but those systems remain the source of truth for their respective domains.

Data Model

Core Entities



Circuit Natural Key

```
(carrier_id, carrier_circuit_id)
```

This composite key prevents duplicate circuits when importing the same carrier inventory multiple times. If a row matches an existing natural key, the circuit is **updated** rather than duplicated.

Manual circuit entry should use Infrastructure and Customer site selectors. Imported endpoint fields remain supported for bulk onboarding, carrier inventory loads, and legacy records, but the preferred operating workflow is to create or correct the Infrastructure/Customer site first and then select that site on the circuit.

Key Enums

ENUM	VALUES
Service Type	wave, epl, evpl, dia, ip_transit, dark_fiber, mpls, cross_connect, cellular_backup, broadband
Status	quoted, ordered, installing, active, suspended, decommissioned

ENUM	VALUES
Purpose	backhaul, backup, customer_dedicated, internet, dr, management
Redundancy	primary, secondary, standalone
Endpoint Type	mdu, tower, data_center, colo, pop, customer_premise, carrier_hotel
Renewal State	stable, watch, active, critical, overdue, decided

CSV/XLSX Import

Import Templates

The import page now has separate downloadable templates:

TEMPLATE	USE FOR	IMPORT ENDPOINT
/templates/netos-circuit-import-template.csv	Wholesale circuits, A/Z endpoints, provider details, agreement terms	/api/v1/imports/circuits/*
/templates/netos-infrastructure-import-template.csv	Data centers, towers, rooftops, colocation, POPs, offices, and aggregation facilities	/api/v1/imports/infrastructure/*

Use separate files when the records have different lifecycles. Circuits are matched by provider plus carrier circuit ID; infrastructure is matched by Xiber infrastructure label.

Circuit Required Fields

FIELD	DESCRIPTION
carrier_name	Upstream provider (e.g., "Lumen", "Zayo")
carrier_circuit_id	The carrier's reference number

These two fields form the circuit natural key. All other fields are optional but recommended.

If `xiber_circuit_label` is omitted, NetOS generates it using `Provider Short Name Service Type A Loc -> Z Loc` when the provider has a short name, otherwise `Carrier Service Type A Loc -> Z Loc`. DIA and broadband use the same pattern without a Z location because they are single-site services. Generated labels are truncated to fit the circuit label field and receive a numeric suffix only when needed for uniqueness.

Recommended Fields

FIELD	TYPE	EXAMPLE
xiber_circuit_label	string	Lumen WAVE Site A -> Site Z
service_type	enum	wave, dia, dark_fiber, broadband

FIELD	TYPE	EXAMPLE
bandwidth_mbps	number	10000
status	enum	active
mrc_usd	decimal	2500.00
nrc_usd	decimal	5000.00
ban_id	string	BAN-100245
billing_account	string	BA-44129
protocol_handoff	string	10GBase-LR
monitoring_url	URL	https://librenms.xiber.local/device/123
install_date	date	2024-01-15
a_endpoint_name	string	Xiber HQ - 123 Main St
a_city	string	Columbus
a_state	string	OH
z_endpoint_name	string	Cologix COL1
z_city	string	Columbus
z_state	string	OH

Contract Fields (Optional)

FIELD	TYPE	EXAMPLE
contract_effective_date	date	2024-01-15
contract_term_end_date	date	2027-01-14
contract_initial_term_months	integer	36
contract_renewal_type	enum	auto_renew, manual, month_to_month, evergreen
contract_renewal_notice_days	integer	90
contract_msa_terms_text	text	36 month initial term...
contract_etf_formula_text	text	100% of remaining MRC
contract_price_escalator_pct	decimal	3.0
contract_pdf_url	URL	/api/v1/circuits/.../agreement/file/...

Infrastructure Fields

Infrastructure imports cover facilities and structures that have their own cost, agreement, and renewal lifecycle. Electrical utility accounts are tracked separately under Electrical Services after the infrastructure or customer site exists.

FIELD	TYPE	EXAMPLE
xiber_infrastructure_label	string	INF-IND-PWR-0012
provider_name	string	AES Indiana
site_type	string	data_center, tower, rooftop, colo, pop, office, other
display_name	string	Indy MDU 42 Electrical Service
facility_account_id	string	METER-100245
latitude, longitude	decimal	39.773350, -86.158050
service_description	text	Electrical utility service for network room
power_commit	string	400A 120/208V 3 phase
monitoring_url	URL	Exact monitoring, graph, or utility-status page
mrc_usd, nrc_usd	decimal	1450.00, 0.00
price_escalator_pct	decimal	3.0
effective_date, term_end_date	date	2024-01-01, 2024-12-31
msa_terms_text, etf_formula_text	text	Agreement language

Manual Bulk Maintenance

After import, users can make broad operational corrections from the UI.

Circuit Bulk Actions

URL: `/circuits`

Select rows with the checkboxes in the first column. The header checkbox selects all currently visible filtered rows.

BULK FIELD	CANONICAL FIELD
Status	status
Provider	carrier_name
Type	service_type
MRC	mrc_usd
NRC	nrc_usd
Term End	contract_term_end_date
Notes	notes

Infrastructure Bulk Actions

URL: `/infrastructure`

Select rows with the checkboxes in the first column. The header checkbox selects all visible rows.

BULK FIELD	CANONICAL FIELD
Status	status
Provider	provider_name
Type	site_type
MRC	mrc_usd
Escalator %	price_escalator_pct
Term End	term_end_date
Notes	notes

Bulk deletes are soft-deletes. They set `deleted_at` and preserve the record for audit/history.

Import Workflow

STEP	WHAT HAPPENS
1. Upload	User uploads a CSV or XLSX file via <code>/import</code> or the API
2. Signature	Raw file hash is calculated for deduplication
3. Mapping	Headers are mapped to canonical fields using heuristic matching
4. Staging	All rows are written to <code>import_staging_rows</code> with validation results
5. Validation	Each row is checked for required fields, type coercion, enum matching
6. Review	User sees first 5 rows with mapped columns and any validation errors
7. Commit	Valid rows are committed; invalid rows are skipped with error details
8. Carrier stubs	Missing carriers are auto-created with name only (fill in details later)
9. Endpoint stubs	Missing circuit endpoints are auto-created; coordinates are populated when latitude/longitude are present

What Happens on Re-Import

When a row matches an existing circuit by natural key `(carrier, carrier_circuit_id)`, or existing infrastructure by `xiber_infrastructure_label`:

- All provided fields are **updated** to the new values
 - Fields not present in the import are **left unchanged**
 - This is an **upsert**, not a replace
-

Validation Rules

FIELD	VALIDATION
carrier_name	Required, non-empty
carrier_circuit_id	Required, non-empty
service_type	Coerced to enum (case-insensitive, underscore-tolerant)
status	Coerced to enum
mrc_usd	Must be numeric if provided
nrc_usd	Must be numeric if provided
bandwidth_mbps	Must be numeric if provided
install_date, decom_date, contract dates, infrastructure term dates	Must be ISO date if provided

Current limitations:

- No inline editing of staged rows (must fix CSV and re-upload)
- No diff preview before updating existing circuits
- No explicit confirmation for financial field overwrites from CSV/XLSX
- Column mapping is heuristic-only (AI-assisted mapping planned)

API Import Examples

Preview (upload and detect mapping):

```
curl -H "x-user-role: exec" \
-F "file=@examples/sample_circuits.csv" \
http://localhost:8000/api/v1/imports/circuits/preview
```

Response:

```
{
  "import_job_id": "abc123",
  "detected_mapping": {
    "Carrier": "carrier_name",
    "Circuit ID": "carrier_circuit_id",
    "MRC": "mrc_usd"
  },
  "headers": ["Carrier", "Circuit ID", "Type", "BW", "MRC", ...],
  "sample_rows": [...],
  "total_rows": 45,
  "valid_rows": 42,
  "invalid_rows": 3
}
```

Commit (import valid rows):

```
curl -H "x-user-role: exec" \
-X POST \
http://localhost:8000/api/v1/imports/circuits/{import_job_id}/commit
```

Infrastructure imports use the same pattern:

```
curl -H "x-user-role: exec" \
-F "file=@netos-infrastructure-import-template.csv" \
http://localhost:8000/api/v1/imports/infrastructure/preview
curl -H "x-user-role: exec" \
-X POST \
http://localhost:8000/api/v1/imports/infrastructure/{import_job_id}/commit
```

Response:

```
{
  "inserted": 38,
  "updated": 4,
  "invalid": 3,
  "errors": [
    {"row": 12, "field": "mrc_usd", "error": "not a valid number"},
    {"row": 23, "field": "carrier_name", "error": "required field missing"},
    {"row": 31, "field": "service_type", "error": "unknown value: 'wavelength'"}
  ]
}
```

Sample CSV

Files:

- /templates/netos-circuit-import-template.csv
- /templates/netos-infrastructure-import-template.csv

```
carrier_name,carrier_circuit_id,xiber_circuit_label,service_type,bandwidth_mbps,status,mrc_usd,nr
c_usd,a_endpoint_name,a_city,a_state,z_endpoint_name,z_city,z_state
Lumen,DHEC.123456,XIB-LUM-0001,wave,10000,active,2500.00,5000.00,Xiber HQ,Columbus,OH,Cologix
COL1,Columbus,OH
Zayo,ZY-789012,XIB-ZAY-0001,dark_fiber,100000,active,4200.00,15000.00,56 Marietta St
NW,Atlanta,GA,Equinix AT1,Atlanta,GA
```

Common circuit purpose values include `backhaul`, `backup`, `customer_dedicated`, `internet`, `dr`, and `management`. Use `backup` for MFC backup circuits, such as broadband service backing up a primary fiber or RF path.

Endpoint Coordinates

Map rendering and geographic topology require endpoint coordinates.

FIELD	TYPE	FORMAT
<code>endpoints.geom</code>	PostGIS <code>POINT</code>	SRID 4326 (WGS84 lat/lng)

Coordinates are stored as PostGIS geometry. The seed data includes coordinates for all sample endpoints. When importing circuits, endpoints are auto-created as stubs without coordinates — add coordinates later via the API or UI to make them appear on the map.

Contract Data

Contracts are linked to circuits and/or infrastructure assets. Key fields:

FIELD	DESCRIPTION
<code>carrier</code>	Provider this contract is with
<code>contract_type</code>	Master, circuit-specific, facility
<code>parent_contract</code>	Link to master agreement (if applicable)
<code>effective_date</code>	When the contract began
<code>initial_term_months</code>	Original commitment length
<code>term_end_date</code>	Calculated end date
<code>renewal_type</code>	<code>auto</code> , <code>manual</code> , <code>evergreen</code>
<code>renewal_term_months</code>	Length of auto-renewal period
<code>renewal_notice_days</code>	Days before end to provide notice
<code>renewal_decision_deadline</code>	Calculated: <code>term_end - notice_days</code>
<code>price_escalator</code>	Annual price increase formula
<code>etf_formula</code>	Early termination fee calculation
<code>etf_current_value</code>	Current estimated ETF
<code>governing_law</code>	Jurisdiction
<code>pdf_url</code>	Link to signed contract PDF

Current state: Sample contracts are created by the seed script. PDF extraction and side-by-side approval are not yet implemented. See [Roadmap](#) for the PDF ingestion plan.

API & Integrations

Xiber NetOS — API & Integrations Guide

REST API endpoints, MCP tools, and external system integration map.

REST API

SETTING	VALUE
Base URL	<code>http://localhost:8000/api/v1</code>
Demo Base URL	<code>https://netos.xiberian.net/api/v1</code>
Docs (Swagger)	<code>http://localhost:8000/docs</code>
OpenAPI Schema	<code>http://localhost:8000/openapi.json</code>
Content Type	<code>application/json</code>

Authentication (Development)

Pass a role header on every request:

```
x-user-role: exec
```

Valid roles: `exec`, `finance`, `network_eng`, `operations`, `pm`, `sales`, `read_only`, `agent`

Production will use Microsoft Entra OIDC JWT tokens. See [Public Hostname & SSO](#).

Endpoints

Health Check

```
GET /healthz
```

```
{"status": "ok"}
```

No authentication required.

List Circuits

```
GET /api/v1/circuits
```

Query Parameters:

PARAM	TYPE	DESCRIPTION
<code>carrier</code>	string	Filter by carrier name
<code>status</code>	string	Filter by status enum
<code>search</code>	string	Full-text search across circuit label, carrier circuit ID, carrier name

Response: Array of circuit objects with nested carrier, contract, A endpoint, and Z endpoint data.

Example:

```
curl -H "x-user-role: exec" \
"http://localhost:8000/api/v1/circuits?status=active&carrier=Lumen"
```

Get Circuit Detail

```
GET /api/v1/circuits/{circuit_id}
```

Response includes:

- Circuit attributes (label, carrier circuit ID, service type, bandwidth, status, MRC, NRC)
- Carrier details (name, NOC phone, portal URL)
- Contract details (term dates, renewal type, ETF, escalator)
- A and Z endpoints (name, address, coordinates)
- Exact monitoring URL and generated monitoring search fallback
- Provider portal URL
- Lifecycle events (ordered by date)

Create Circuit

```
POST /api/v1/circuits
```

Requires: Write role (`exec`, `finance`, `network_eng`, `operations`, `pm`)

Request Body:

```
{
  "xiber_circuit_label": "XIB-LUM-0042",
  "carrier_circuit_id": "DHEC.123456",
  "carrier_name": "Lumen",
```

```
"service_type": "wave",
"bandwidth_mbps": 10000,
"status": "active",
"monitoring_url": "https://librenms.xiber.local/device/123",
"mrc_usd": 2500.00,
"nrc_usd": 5000.00,
"a_endpoint_name": "Xiber HQ",
"z_endpoint_name": "Cologix COL1"
}
```

The manual UI now uses Infrastructure and Customer site selectors for circuit A/Z locations. The REST API still accepts endpoint fields so imports and integrations can create endpoint snapshots. For attribution, creating a circuit is not enough by itself; create a `subtended_links` relationship to state which infrastructure asset is the upstream donor and which downstream structure or customer endpoint is served over that circuit.

For DIA and broadband services, NetOS treats the circuit as single-site and clears the Z endpoint.

Customer, RF Link, Electrical, and Attribution APIs

Customers

```
GET /api/v1/customers
POST /api/v1/customers
GET /api/v1/customers/{customer_id}
PATCH /api/v1/customers/{customer_id}
DELETE /api/v1/customers/{customer_id}
```

Customer records represent commercial and MFC customer sites. They can be selected by circuits, RF Links, and subtended links. Customer detail responses include associated circuits, RF Links, serving paths, inherited donor cost, customer revenue, attributed cost, and gross margin.

RF Links

```
GET /api/v1/rf-links
POST /api/v1/rf-links
PATCH /api/v1/rf-links/{rf_link_id}
DELETE /api/v1/rf-links/{rf_link_id}
POST /api/v1/rf-links/{rf_link_id}/documents
```

RF Links are built from existing infrastructure and customer sites. Links between two infrastructure assets are backhaul. Links from infrastructure to a customer site are customer endlinks. When a customer endlink is saved, NetOS synchronizes the matching subtended customer endpoint relationship so attribution can show RF transport.

RF System Catalog

```
GET /api/v1/rf-systems
POST /api/v1/rf-systems
```

```
PATCH /api/v1/rf-systems/{rf_system_id}
DELETE /api/v1/rf-systems/{rf_system_id}
```

RF systems define selectable defaults such as name, description, frequency used, capacity, cost, PTP/PtMP type, equipment role, and PtMP compatibility. PtMP source nodes must reference RF systems with role `ptmp_base`. PtMP customer RF Links must reference a subscriber/client RF system with role `ptmp_subscriber`; if the subscriber has compatible base system IDs, the selected PtMP source node's base system must be included.

Electrical Services

```
GET /api/v1/electrical-services
POST /api/v1/electrical-services
PATCH /api/v1/electrical-services/{electrical_service_id}
DELETE /api/v1/electrical-services/{electrical_service_id}
POST /api/v1/electrical-services/{electrical_service_id}/documents
```

Electrical services attach to exactly one Infrastructure or Customer site. They store provider, service type, account and meter numbers, voltage, amperage, phase, delivery location, service address, average monthly cost, start date, monitoring URL, notes, and supporting documents.

Infrastructure Subtended Links

```
POST /api/v1/infrastructure/{infrastructure_id}/subtended-links
PATCH /api/v1/infrastructure/{infrastructure_id}/subtended-links/{link_id}
DELETE /api/v1/infrastructure/{infrastructure_id}/subtended-links/{link_id}
```

Subtended links explicitly define the financial hierarchy. Use them to attach customer endpoints, downstream structures, existing circuits, or RF Links to a parent infrastructure asset. This is what drives the attribution tree, dashboard attribution cards, customer inherited-cost drilldowns, and infrastructure financial summaries.

Bulk Update Circuits

```
PATCH /api/v1/circuits/bulk
```

Requires: Write role (`exec`, `finance`, `network_eng`, `operations`, `pm`)

Request Body:

```
{
  "circuit_ids": ["8d8e4333-8182-4443-8e0d-f2307276db52"],
  "updates": {
    "status": "active",
    "carrier_name": "Lumen",
    "service_type": "dia",
    "mrc_usd": 1850.00,
    "nrc_usd": 0,
    "contract_term_end_date": "2027-12-31",
    "notes": "Updated through bulk action"
  }
}
```

```
}  
}
```

Only fields included in `updates` are changed. Each affected circuit receives an audit entry.

Response:

```
{"updated": 1, "requested": 1}
```

Bulk Delete Circuits

```
POST /api/v1/circuits/bulk-delete
```

Soft-deletes circuits by setting `deleted_at`. Records remain available for audit and historical reporting.

Request Body:

```
{  
  "circuit_ids": ["8d8e4333-8182-4443-8e0d-f2307276db52"]  
}
```

Response:

```
{"deleted": 1, "requested": 1}
```

Map Data

```
GET /api/v1/circuits/map
```

Query Parameters:

PARAM	TYPE	DESCRIPTION
<code>carrier</code>	string	Filter by carrier
<code>service_type</code>	string	Filter by service type enum
<code>status</code>	string	Filter by status enum
<code>state</code>	string	Filter by endpoint state (US state)

Response:

```
{  
  "circuits": [  
    {  
      "id": "uuid",  
      ...  
    }  
  ]  
}
```

```

    "label": "XIB-LUM-0042",
    "carrier": "Lumen",
    "service_type": "wave",
    "bandwidth_mbps": 10000,
    "status": "active",
    "a_endpoint": {"name": "...", "lat": 39.96, "lng": -82.99},
    "z_endpoint": {"name": "...", "lat": 39.95, "lng": -83.00},
    "geometry": {"type": "LineString", "coordinates": [[-82.99, 39.96], [-83.00, 39.95]]}
  },
],
"endpoints": [
  {
    "id": "uuid",
    "name": "Xiber HQ",
    "type": "pop",
    "geometry": {"type": "Point", "coordinates": [-82.99, 39.96]}
  }
],
"infrastructure": [
  {
    "id": "uuid",
    "name": "Cologix COL1",
    "type": "data_center",
    "geometry": {"type": "Point", "coordinates": [-83.00, 39.95]}
  }
],
"filters": {
  "carriers": ["Lumen", "Zayo", "Cogent"],
  "service_types": ["wave", "dia", "dark_fiber"],
  "statuses": ["active", "ordered"],
  "states": ["OH", "GA", "IL"]
}
}

```

Dashboard Summary

GET /api/v1/dashboard/summary

Response includes:

SECTION	FIELDS
kpis	total_circuits, active_circuits, infrastructure_count, circuit_mrc, facility_mrc, total_mrc, modeled_mrr, modeled_margin, renewal_window_count
spend_by_carrier	Array of {carrier, mrc, circuit_count}
service_mix	Array of {service_type, count}
state_mix	Array of {state, count}
renewal_pipeline	Array of {state, count}
margin_histogram	Array of {bucket, count}
at_risk_circuits	Array of circuit summaries

SECTION	FIELDS
top_expensive	Array of circuit summaries
lowest_margin	Array of circuit summaries

Import Preview

```
POST /api/v1/imports/circuits/preview
Content-Type: multipart/form-data
```

Upload a CSV or XLSX file. Returns import job ID, detected column mapping, headers, sample rows, and validation summary.

Example:

```
curl -H "x-user-role: exec" \
-F "file=@examples/sample_circuits.csv" \
http://localhost:8000/api/v1/imports/circuits/preview
```

Import Commit

```
POST /api/v1/imports/circuits/{import_job_id}/commit
```

Commits all valid staged rows. Returns insert/update/invalid counts.

Example:

```
curl -H "x-user-role: exec" \
-X POST \
http://localhost:8000/api/v1/imports/circuits/{import_job_id}/commit
```

List Service Providers

```
GET /api/v1/service-providers
```

Returns carrier records with circuit count, infrastructure count, and total spend.

Get Service Provider

```
GET /api/v1/service-providers/{carrier_id}
```

Returns provider detail with related circuits and infrastructure assets.

List Infrastructure Assets

```
GET /api/v1/infrastructure
```

Returns infrastructure asset records with facility type, provider, costs, terms, and `monitoring_url`.

Bulk Update Infrastructure Assets

```
PATCH /api/v1/infrastructure/bulk
```

Request Body:

```
{
  "infrastructure_ids": ["ae30e22f-565d-46f3-be94-9d0e001a59de"],
  "updates": {
    "status": "active",
    "provider_name": "American Tower",
    "site_type": "tower",
    "mrc_usd": 2200.00,
    "price_escalator_pct": 3.0,
    "term_end_date": "2028-12-31",
    "notes": "Updated through bulk action"
  }
}
```

Only fields included in `updates` are changed. Each affected asset receives an audit entry.

Response:

```
{"updated": 1, "requested": 1}
```

Bulk Delete Infrastructure Assets

```
POST /api/v1/infrastructure/bulk-delete
```

Soft-deletes infrastructure assets by setting `deleted_at`.

Request Body:

```
{
  "infrastructure_ids": ["ae30e22f-565d-46f3-be94-9d0e001a59de"]
}
```

Response:

```
{"deleted": 1, "requested": 1}
```

Audit Activity

```
GET /api/v1/audit/activity
```

Returns recent request/write activity including user, role, IP, event type, action, entity, changed fields, and response status.

Feedback Queue

```
GET /api/v1/feedback
POST /api/v1/feedback
PATCH /api/v1/feedback/{feedback_id}
POST /api/v1/feedback/{feedback_id}/comments
```

Feedback requests support `priority`, `status`, and progress comments. Public comments attempt to notify the requester by email when SMTP is configured.

MCP Server

Location: `apps/mcp/`

The MCP server exposes NetOS data to XOS agents (ExecOS, department agents, Claude Desktop users).

Current Tools

TOOL	STATUS	DESCRIPTION
<code>list_circuits</code>	Working	List circuits with optional filters
<code>get_circuit</code>	Working	Get single circuit detail
<code>get_renewals_due</code>	Placeholder	Circuits approaching renewal deadline
<code>get_etf_exposure</code>	Placeholder	ETF exposure by carrier or portfolio

Planned Tools

TOOL	DESCRIPTION
<code>get_carrier_summary</code>	Carrier spend, circuit count, ETF exposure

TOOL	DESCRIPTION
<code>get_circuit_pl</code>	Per-circuit P&L with revenue attribution
<code>find_circuits_at_address</code>	Find circuits near or at an address
<code>get_outage_history</code>	Outage events for a circuit
<code>get_renewal_pipeline</code>	All circuits by renewal state
Write tools	Create/update with explicit agent confirmation flow

Authentication

MCP tools use role-based header auth matching the REST API:

```
x-user-role: agent
```

In production, MCP requests will carry a service token validated against the gateway's agent registry.

Registration with XOS Gateway

The MCP server should be registered with the Xiber MCP Gateway (`gateway.xiberian.net`) so it's available to all XOS agents and Claude Desktop users via the SSE bridge. This is not yet configured.

External System Integration Map

SYSTEM	DIRECTION	PURPOSE	STATUS
Sonar	Pull	Customer accounts, MRR, revenue attribution per circuit	Planned
Monday.com	Bidirectional	Renewal tasks, variance tasks, circuit order workflow	Planned
Wisdm	Pull	Tower and network site geometry for map overlays	Planned
LibreNMS / Prometheus / Loki	Pull	Utilization data, outage detection, monitoring deep links	Planned
DocuSeal	Push	Generate termination notices and contract execution envelopes	Planned
Microsoft 365 / Entra ID	Pull	SSO, user identity, calendar/Teams/SharePoint references	Planned
XOS (MCP Gateway)	Expose	Agent-readable circuit tools for automated reporting and decisions	Skeleton
Lumen Control Center	Pull	Circuit status sync from carrier portal	Not started
Zayo Tranzact	Pull	Circuit status sync from carrier portal	Not started
Cogent Portal	Pull	Circuit status sync from carrier portal	Not started

Planned Webhooks (Outbound)

EVENT	TRIGGER	DESTINATION
<code>circuit.installed</code>	Circuit status → <code>active</code>	Monday.com, Slack
<code>circuit.decommissioned</code>	Circuit status → <code>decommissioned</code>	Monday.com, Slack
<code>renewal.state_changed</code>	Renewal state transition	Monday.com, Email, Slack
<code>invoice.variance_detected</code>	Invoiced MRC ≠ contracted MRC	Monday.com (task for finance)

Webhooks are not yet implemented.

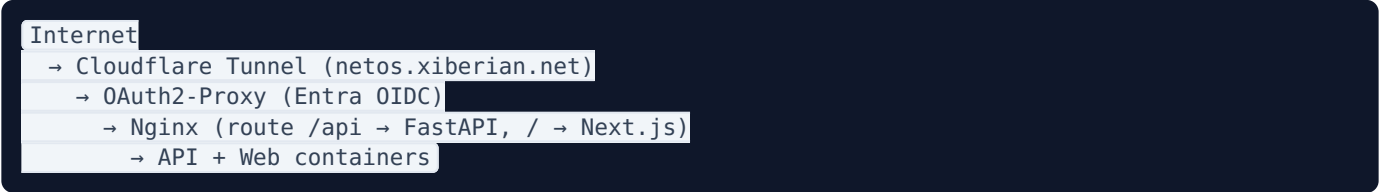
Public Hostname & SSO

Xiber NetOS — Public Hostname & SSO

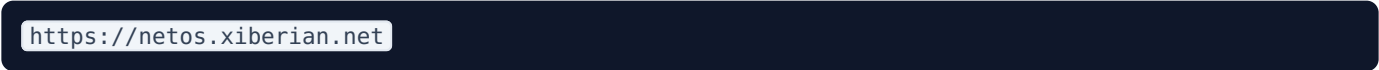
Cloudflare Tunnel deployment with Microsoft Entra ID single sign-on.

Overview

NetOS can be exposed publicly through a **Cloudflare Tunnel** and protected with **Microsoft Entra ID** SSO. The auth layer sits in front of the app — unauthenticated users never reach the API or web UI.



Target Hostname



The hostname routes to the `sso-proxy` container, not directly to web or API.

Runtime Components

CONTAINER	ROLE	PORT
<code>cloudflared</code>	Outbound tunnel to Cloudflare edge	—
<code>sso-proxy</code>	OAuth2-Proxy with Entra OIDC	4180
<code>public-proxy</code>	Nginx reverse proxy	8080
<code>web</code>	Next.js UI	3000
<code>api</code>	FastAPI backend	8000
<code>postgres</code>	PostgreSQL 16	5432
<code>redis</code>	Redis 7	6379

Routing Rules (Nginx)

PATH	DESTINATION
/	Web app (Next.js)
/api/	FastAPI backend
/docs-api/	FastAPI Swagger UI
/openapi.json	OpenAPI schema

Prerequisites

- Cloudflare account with Zero Trust enabled
- Microsoft Entra ID tenant (Xiber: 4944843d-e347-4fe7-a279-4006ce5efc33)
- Docker Engine 24+ with Docker Compose v2

Setup Steps

1. Create Entra App Registration

1. Go to **Azure Portal** → **App registrations** → **New registration**
2. Name: Xiber NetOS
3. Redirect URI (Web): https://netos.xiberian.net/oauth2/callback
4. Note the **Application (client) ID** and create a **client secret**

Recommended access policy:

- Assign only Xiber users or a dedicated NetOS-Users security group
- Require MFA and compliant device if available
- Map Entra groups to NetOS roles (exec, finance, network_eng, etc.)

2. Create Environment File

Copy the example and fill in real values:

```
cp infra/docker/.env.public.example infra/docker/.env.public
```

Required variables:

VARIABLE	DESCRIPTION	EXAMPLE
ENTRA_OIDC_ISSUER_URL	Entra OIDC issuer	https://login.microsoftonline.com/{tenant_id}/v2.0
ENTRA_CLIENT_ID	App registration client ID	29833a06-d27e-...

VARIABLE	DESCRIPTION	EXAMPLE
ENTRA_CLIENT_SECRET	App registration client secret	b-V8Q-b0_YRL...
OAuth2_PROXY_COOKIE_SECRET	Random 32-byte base64 string	(generate below)
CLOUDFLARE_TUNNEL_TOKEN	Tunnel token from Cloudflare dashboard	eyJhIjoi...

Generate cookie secret:

```
python3 -c "import base64, secrets;
print(base64.urlsafe_b64encode(secrets.token_bytes(32)).decode())"
```

3. Create Cloudflare Tunnel

1. Go to **Cloudflare Zero Trust** → **Tunnels** → **Create a tunnel**
2. Name: `netos`
3. Copy the tunnel token into `.env.public` as `CLOUDFLARE_TUNNEL_TOKEN`
4. Add a public hostname:

- **Hostname:** `netos.xiberian.net`
- **Service:** `http://sso-proxy:4180`

1. Optionally add a Cloudflare Access policy (Microsoft Entra ID, Xiber users only) as a second enforcement layer

4. Start the Public Stack

```
cd infra/docker
docker compose --env-file .env.public \
-f docker-compose.yml \
-f docker-compose.public.yml \
--profile public \
up -d
```

5. Verify

Check containers:

```
docker compose -f docker-compose.yml \
-f docker-compose.public.yml \
--profile public \
ps
```

Open in browser:

```
https://netos.xiberian.net
```

You should be redirected to Microsoft login. After authenticating, you'll see the NetOS UI.

Architecture Notes

Why OAuth2-Proxy + Cloudflare Access?

Both layers serve complementary purposes:

LAYER	PURPOSE
Cloudflare Access	Edge enforcement — blocks unauthenticated traffic before it reaches your server
OAuth2-Proxy	Origin enforcement — protects the app even if tunnel routing is misconfigured

Keep both. The overhead is negligible and the defense-in-depth is worth it.

JWT Validation (Future)

The current API uses a header-based dev auth shim. Production should:

1. Extract the JWT from the OAuth2-Proxy `Authorization` header or cookie
2. Validate signature against Entra JWKS endpoint
3. Extract user email and group claims
4. Map Entra groups to NetOS roles
5. Reject requests with invalid/expired tokens

This is tracked in [Roadmap](#) → [Authentication](#).

Security Checklist

ITEM	STATUS
HTTPS via Cloudflare	Automatic with tunnel
Entra OIDC authentication	Via OAuth2-Proxy
Cloudflare Access policy	Recommended additional layer
Cookie secret rotation	Manual — regenerate and restart periodically
Client secret rotation	Via Azure Portal — update <code>.env.public</code> after rotation
MFA enforcement	Configure in Entra Conditional Access
Role-based access	Dev shim now, JWT group mapping planned

Troubleshooting

ISSUE	SOLUTION
Redirect loop after login	Check <code>OAUTH2_PROXY_COOKIE_SECRET</code> is exactly 32 bytes base64-encoded
502 Bad Gateway	Verify <code>sso-proxy</code> and <code>public-proxy</code> containers are running
Tunnel not connecting	Check <code>CLOUDFLARE_TUNNEL_TOKEN</code> is correct; verify tunnel is active in CF dashboard
CORS errors in browser	Verify Nginx config allows the public hostname as an origin
"Access Denied" after login	User may not be in the assigned Entra group; check app assignment
API returns 401	JWT validation not yet implemented — ensure dev auth headers are passed through proxy

Authorization & RBAC

NetOS Authorization and RBAC

NetOS uses Microsoft SSO / Microsoft Entra ID for authentication only. Microsoft proves identity. NetOS decides authorization with internal users, roles, permissions, scopes, and field-level filtering.

Security Model

- Default deny: every `/api/v1` route requires an active internal user with `app_access` and the route permission.
- Backend enforcement is authoritative. Frontend hiding is only a usability layer.
- Financial and executive fields are filtered before API responses are returned.
- Permission changes are written to the activity audit log with `event_type="authorization"`.
- Sensitive financial access is covered by the normal API activity log and should be expanded for especially sensitive exports or investor reporting endpoints as those are added.

Identity Flow

1. Microsoft SSO authenticates the user.
2. The reverse proxy forwards trusted identity headers to the API.
3. NetOS normalizes the email and optional Entra object ID.
4. NetOS loads the internal `users` record.
5. If the user is inactive or has no record, access is denied unless they match the bootstrap Super Admin allowlist.
6. Roles and direct user permission overrides are resolved.
7. Route and field authorization is applied.

Authenticated does not mean authorized.

Seeing The Current User

The current API-resolved user is available at:

```
/api/v1/authz/me
```

The main navigation footer also shows the email that NetOS is authorizing and the source of that identity:

- `Microsoft SSO`: the API received and trusted proxy authentication headers.
- `Dev fallback`: the API did not use Microsoft proxy headers and fell back to development auth.

Production should show `Microsoft SSO`. If it shows `Dev fallback`, confirm `ALLOW_DEV_AUTH=false` and `TRUST_PROXY_AUTH_HEADERS=true` in the running API environment and verify the SSO proxy forwards identity headers such as `X-Auth-Request-Email`.

Bootstrap Super Admin

Set:

```
RBAC_BOOTSTRAP_SUPER_ADMIN_EMAILS=admin@example.com,second-admin@example.com
```

When one of those Microsoft-authenticated users first reaches the app, NetOS creates or activates the internal user and assigns the `Super Admin` role. If the variable is not set, NetOS falls back to `ADMIN_EMAIL`; in dev auth mode it also allows `DEV_AUTH_DEFAULT_EMAIL`.

Data Model

Core tables:

- `users`: internal user mapped to email and optional Entra object ID.
- `roles`: named permission bundles.
- `permissions`: granular permission catalog.
- `role_permissions`: permissions included in roles.
- `user_roles`: role assignments, including `scope_type` and `scope_ref_id`.
- `user_permissions`: optional direct allow/deny overrides.
- `permission_scopes`: reusable labels for global, project, site, and department scopes.
- `user_activity_logs`: audit history, including authorization changes.

Scopes currently support `global`, `project`, `site`, and `department`. Global enforcement is live; entity-scope query filters are intended to be applied module by module where project/site assignment data exists.

Default Roles

- `Super Admin`: all permissions.
- `Executive`: operational dashboards and financial summaries, including executive financial visibility.
- `Finance Admin`: full finance, invoices, reporting, and financial field visibility.
- `Project Manager`: project and assigned operational data with project/asset/circuit costs.
- `Network Engineer`: technical inventory edit access with relevant asset and circuit costs.
- `Field Technician`: assigned operational/site tooling, no financial fields by default.
- `Read-Only User`: non-sensitive operational read access only.

Roles are composed of permissions. Do not hard-code role names in new features except for bootstrap or emergency super-admin behavior.

Protecting Routes

Add a permission to `apps/api/app/core/permissions.py`, then add route mapping in `ROUTE_PERMISSION_PREFIXES`.

For explicit endpoint checks:

```
from app.core.auth import require_permission

@router.post("/example", dependencies=[Depends(require_permission("assets_edit"))])
async def create_example(...):
    ...
```

Every route under `/api/v1` also runs `require_route_permission`.

Protecting Fields

Financial fields must not be sent and hidden only in the browser. Use field permissions:

- `field.asset_cost.view`
- `field.circuit_cost.view`
- `field.customer_revenue.view`
- `field.margin.view`
- `field.capex.view`
- `field.executive_summary.view`

Example:

```
from app.core.auth import can_view_field

if not can_view_field(principal, "field.circuit_cost.view"):
    payload["mrc_usd"] = None
    payload["nrc_usd"] = None
```

Exports and search endpoints must use the same filtered payload strategy and must not include protected fields in labels, descriptions, or metadata.

Admin UI

The Admin page includes an Authorization section for:

- creating users
- activating and deactivating users
- assigning roles
- assigning global/project/site/department scopes
- editing role permission composition

The sidebar hides navigation items when the current user lacks the associated view permission.

Setup

Apply migrations:

```
docker exec netos_api_1 alembic upgrade head
```

Restart API/web after code changes:

```
docker restart netos_api_1 netos_web_1
```

Run tests:

```
docker exec netos_api_1 pytest apps/api/tests/test_rbac_authorization.py
```

Security Considerations

- Keep Microsoft Conditional Access and MFA in Entra; NetOS remains compatible because it only consumes the authenticated identity.
- Do not grant `app_access` automatically to every Microsoft user.
- Prefer scoped roles for project/site/department work.
- Use direct user permissions sparingly and audit them.
- Never add wildcard permissions.
- New financial, payroll, compensation, debt, investor reporting, export, or search features must define explicit route and field permissions before release.

Infrastructure Attribution

Xiber NetOS - Infrastructure Revenue & Expense Attribution

Operating model for measuring the financial performance of towers, rooftops, data centers, POPs, carrier hotels, and aggregation facilities.

Purpose

Infrastructure assets are not only places where equipment is installed. They are economic platforms that support revenue-producing services downstream. A rooftop, tower, data center, or aggregation site should be evaluated by the revenue it enables, the direct cost it carries, and the downstream costs required to deliver that revenue.

This guide defines how NetOS should attribute revenue and expenses through infrastructure relationships so the platform can eventually produce a full waterfall view of network economics.

The model separates three concepts:

CONCEPT	MEANING
Physical placement	Where a circuit, radio, or facility endpoint physically lands. Example: circuit A location is BSI 1America.
Subtended relationship	Which parent asset economically supports a customer endpoint, circuit, or child facility. Example: BSI 1America subtends Tower A over fiber.
Cost source	The object that contributes cost to the relationship. Example: the customer endpoint is served through the Lakota Lakes POP circuit, so the relationship is <code>customer endpoint</code> and the cost source is <code>Circuit: Lakota Lakes POP</code> .
Attribution period	The effective date range during which a relationship is financially true. Example: Tower A moved from Parent POP 1 to Parent POP 2 on 2026-05-01.

The same circuit endpoint can stay physically fixed while the business relationship changes over time.

Relationship Diagram

The diagram below shows the intent of the infrastructure attribution tool. Customer revenue is entered at the endpoint or downstream relationship that produces it. Facility costs stay with the asset that owns the obligation. Circuit costs stay on the circuit and are allocated through subtended links when that circuit supports a customer, tower, rooftop, or child facility. The waterfall output rolls those values up to the serving structure, parent facility, market, and portfolio.



Infrastructure attribution waterfall diagram

Core Objects

Infrastructure Asset

An infrastructure asset is a site, facility, or structure that can carry its own cost and may support downstream revenue. Electrical utility accounts are not infrastructure assets; they are add-on service cost items attached to infrastructure or customer sites.

Examples:

ASSET TYPE	EXAMPLES	TYPICAL DIRECT COSTS
Tower	owned tower, leased tower, water tower attachment	rent, ground lease, power, backhaul, insurance
Rooftop	building rooftop, hotel roof, MDU roof	roof lease, power, access fees
Data center	colocation facility, meet-me site	cabinet, cross-connects, power, remote hands
POP	aggregation point, hub cabinet, carrier hotel presence	rack, power, transport, cross-connects
Carrier hotel	interconnect facility	MMR fees, cross-connects, cabinet
Aggregation facility	cabinet, hut, shelter, central office	lease, utility, backhaul

Current NetOS fields that contribute to direct site cost:

FIELD	MEANING
mrc_usd	Monthly recurring site/facility cost
nrc_usd	One-time non-recurring site/facility cost
price_escalator_pct	Contractual recurring cost escalation
term_end_date	Renewal/expiration date
agreement_url and documents	Supporting contracts, invoices, floorplans, photos, MSA, service orders

Infrastructure assets also carry a `cost_classification` that determines how the facility MRC participates in financial reporting:

CLASSIFICATION	WATERFALL TREATMENT
operating	Default state. Facility MRC participates in normal structure economics and inherited donor cost attribution.
pre_revenue	Facility MRC is held as pre-revenue investment. It does not flow into downstream inherited cost. The asset carries target MRR, breakeven date, decision owner, notes, and optional review/promotion thresholds.
strategic_redundancy	Facility MRC is held as strategic redundancy investment. It does not flow into customer margin. The asset records the protected asset, SLA justification, and review date.
market_overhead	Facility MRC contributes to a market-level overhead pool and is excluded from direct structure/customer inherited cost.

This separates steady-state operating margin from deliberate strategic investment. The market waterfall now reports operating revenue, operating cost, and operating margin first. It then deducts pre-revenue investment, strategic redundancy, and market overhead to produce portfolio contribution. Non-operating assets can still appear in topology and inventory, but their facility rent does not distort a downstream tower, POP, or customer margin while the asset is still being justified.

Circuit

A circuit is transport, internet, fiber, cross-connect, or other carrier service. It has its own cost and may connect one infrastructure asset to another, or connect an infrastructure asset to a customer endpoint.

Current NetOS fields that contribute to circuit cost:

FIELD	MEANING
mrc_usd	Monthly recurring carrier cost
nrc_usd	One-time install or setup charge
service_type	DIA, broadband, EVPL, EPL, wave, dark fiber, etc.
A/Z endpoints	Physical locations where the circuit terminates
Agreement documents	Service orders, MSAs, amendments, invoices

RF Link

An RF Link is wireless transport between an A location and a Z location. It can represent PTP microwave, PtMP sectors, Tarana links, rooftop wireless links, or licensed backhaul. RF Links are separate from carrier circuits because their economics often include equipment CAPEX, tower or rooftop rent, spectrum/licensing data, and customer endpoint service delivered from a structure.

RF Links are built from existing site records and follow the same direction convention as circuits: the A side is the customer or downstream side, and the Z side is closer to the donor/core. For facility-to-facility backhaul, select an A infrastructure asset and a Z infrastructure asset. For customer PtP service, select the customer endpoint and the Z donor facility. NetOS classifies links with a customer endpoint as customer endlinks and links between two infrastructure assets as backhaul. The displayed link name is generated from the A side, Z side, and selected RF system or equipment type so users do not maintain a separate manual name.

When an RF Link is created from an infrastructure asset to a customer endpoint, NetOS should create or update the matching subtended customer endpoint relationship. The relationship remains `customer_endpoint`, but the transport becomes RF (`ptp_wireless`, `ptmp_wireless`, or `microwave`) and the cost source becomes the RF Link. This keeps the hierarchy drilldown clear: the endpoint is the revenue relationship, and the RF Link is the wireless transport/cost source serving it.

Current NetOS fields that contribute to RF Link attribution:

FIELD	MEANING
<code>rf_system_id</code> and <code>equipment_type</code>	RF system standard selected from Admin, plus the radio, antenna, sector, or platform used by the link
<code>network_role</code>	Generated role: customer endlink when one endpoint is a customer, or backhaul when both endpoints are infrastructure
<code>path_role</code> and <code>diversity_group</code>	Primary, secondary, backup, diverse, or grouped path metadata used to distinguish parallel connectivity
<code>map_color</code>	Optional display hint for differentiating multiple links that follow the same visual path
<code>frequency_mhz</code> and <code>channel_width_mhz</code>	Spectrum used by the path
<code>bandwidth_mbps</code>	Capacity available on the RF path
<code>capex_usd</code>	One-time equipment or construction cost, tracked separately from monthly margin
<code>salvage_value_usd</code> , <code>useful_life_months</code> , <code>depreciation_start_date</code>	Inputs for straight-line monthly depreciation
<code>mrc_usd</code>	Monthly recurring non-rent RF cost, if any
<code>tower_rent_mrc_usd</code>	Monthly rent incurred because this RF link uses a tower or rooftop
<code>fcc_call_sign</code> and documents	License reference and uploaded FCC license or coordination documents
A/Z locations	Existing infrastructure assets, with customer records acting as customer-site Z endpoints

For monthly waterfall reporting, NetOS treats RF Link MRC plus tower rent MRC as calculated link cost. CAPEX remains visible on the RF Link and NetOS calculates straight-line monthly depreciation when useful life data is present. Depreciation is an asset-cost lens, not a cash-cost replacement for MRC.

RF system standards live in Admin. Each RF system has a name, description, frequency used, capacity, cost, PTP/PtMP type, and equipment role. PtMP inventory is split into base/sector systems and subscriber/client systems. Infrastructure assets host PtMP nodes using base/sector systems, such as a Tarana BN, Siklu Terragraph DN, or Cambium ePMP AP. RF Links created from those PtMP nodes use compatible subscriber systems, such as Tarana RN/RNv, Siklu T280/T265, or Cambium Force 4600c. When a user selects a PtMP source node, NetOS limits the RF Link subscriber selector to radios compatible with that base system and preloads equipment, capacity, and CAPEX defaults. These are starting values; the actual saved CAPEX should be adjusted to match the bill of materials, labor, and site-specific installation cost.

For PtMP customer endlinks, NetOS filters the customer endpoint selector to mapped customer sites within 10 miles of the selected PtMP source node. Users can select multiple nearby customer endpoints and create one RF Link per selected customer in a single add action. Customers without valid coordinates are not shown in this PtMP proximity selector until their address or latitude/longitude is corrected.

When a site has multiple paths, create separate circuits and RF Links for each physical or logical path. Use `path_role` for operating intent, such as primary or backup, and `diversity_group` for paths that should be evaluated together. On maps, parallel lines that share an A/Z path should be visually separated by path role, color, and slight line offsets so primary, secondary, and backup paths remain distinguishable.

The same downstream structure can have multiple active subtended relationships to the same parent when each relationship has a distinct circuit or RF Link. This represents redundant transport such as two fibers, fiber plus RF, or primary and backup paths. Reporting should count the downstream structure's own revenue, facility cost, and child rollup once, then add the cost of every active transport path between the parent and downstream structure. A downstream structure should not be active under two different parent sites for the same period unless the old relationship is ended and a new effective-dated relationship is created.

Electrical Service

An electrical service is a utility account or site power service attached to an existing infrastructure asset or customer site. It is separate from infrastructure because the service is an operating cost item, not a structure in the network hierarchy.

Current NetOS fields that contribute to electrical service cost:

FIELD	MEANING
<code>provider_id</code>	Electrical service provider selected from existing service providers
<code>infrastructure_id</code> or <code>customer_id</code>	Exactly one site where the electrical service is delivered
<code>service_type</code>	Commercial power, house power, dedicated meter, generator, solar, battery backup, or other service
<code>account_number</code> and <code>meter_number</code>	Utility account and meter identifiers
<code>service_voltage</code> , <code>service_amperage</code> , <code>phase</code>	Electrical characteristics for the delivered service
<code>delivery_location</code> and service address	Where at the site the service is delivered

FIELD	MEANING
average_monthly_cost_usd	Monthly cost used for attribution until actual bill data is integrated
start_date	Service start date
monitoring_url	Utility portal, metering, generator, UPS, or power monitoring link
Documents	Sample bills, invoices, service agreements, meter photos, and supporting documents

For attribution, electrical service cost should roll into the site it is attached to. If the service is attached to an infrastructure asset, it increases the direct site cost of that asset. If the service is attached to a customer site, it increases the customer endpoint cost. This keeps the physical hierarchy clean while still capturing the recurring cost required to operate the site.

Infrastructure and customer detail pages should show attached electrical services next to the other related services for that site. Users should be able to see the provider, account, meter, service characteristics, average monthly cost, monitoring link, and uploaded bills or supporting documents without leaving the site detail context.

Customer

A customer is the revenue owner for an endpoint-style service. NetOS separates customers from circuits and RF Links so the same commercial account or MFC property can have multiple paths, multiple services, and multiple supporting assets.

Commercial customer records describe the sold service. DIA and PTP include capacity because they are transport products. OTT services such as VoIP, IPTV, managed WiFi, and managed security are tracked as service flags on the customer. Customer records also carry contracted MRR, install NRC, and revenue source. MFC customer records describe the property model, including address, tenant count, access technology, property owner, property manager, and access rules.

Circuits, RF Links, and subtended links can reference `customer_id`. That selector should be preferred when the endpoint customer already exists. Free-form customer names remain useful for discovery records that have not yet been normalized into a customer record.

Customers also carry address and coordinates because they can act as the customer-site Z location for circuits and RF Links. This lets the map show customer sites alongside infrastructure assets and lets RF Links terminate at either a facility, a free-form Z location, or a customer site.

Revenue precedence is explicit: subtended link `revenue_mrr_usd` is the override when entered. If the customer has no explicit subtended link revenue, NetOS uses the customer's contracted MRR once for the customer summary. When several protected or parallel paths serve the same customer and no explicit path revenue is entered, the serving path table displays the fallback MRR evenly across paths for readability, but the customer financial summary does not multiply the customer revenue by the path count. This allows quick customer-site setup while preserving precise per-link revenue when a customer has multiple services.

Customer detail pages are the endpoint drilldown for the waterfall model. When a user clicks a customer endpoint in the hierarchy, subtended network list, dashboard attribution, or financial view, NetOS opens that customer and shows:

CUSTOMER DETAIL SECTION	MEANING
Customer Revenue	Contracted MRR from the customer record, unless a subtended link revenue override is present
Direct Serving Cost	Cost of the circuit or RF Link directly serving the customer endpoint
Inherited Donor Cost	Allocated share of shared facility MRC and upstream donor paths that feed the POP, tower, rooftop, or facility serving the customer
Total Attributed Cost	Direct serving cost plus inherited donor cost, maintenance reserve, and commission when present
Gross Margin	Customer revenue minus total attributed cost
Serving Paths	The donor structure, transport type, circuit/RF source, revenue, and cost for each path serving the customer
ROI Analysis	Customer revenue and attributed cost projected against total build CAPEX and serving path costs over 2-, 3-, 5-, and 10-year terms

The inherited donor cost model defaults to revenue pro rata allocation at each structure in the serving chain. If a POP has \$500 of facility MRC and an upstream circuit costing \$1,000 MRC, and that POP serves \$10,000 of unique active customer MRR, a customer generating \$1,500 MRR inherits 15% of the POP facility cost and 15% of the upstream circuit cost. If the POP itself is fed by a core site, the customer also inherits a revenue pro rata share of the core site's facility MRC using total customer revenue generated under that core site. This is explicit in the UI as `revenue_pro_rata`. It applies to shared circuits, RF transport, tower or rooftop rent represented through RF Link monthly cost, and facility rent/MRC.

Users can override inherited cost allocation on the subtended relationship when revenue pro rata is not the right driver. The default should remain `revenue_pro_rata` unless there is a documented reason to use a different method. `manual_pct` applies the entered inherited allocation percentage to upstream facility and donor path cost. `equal_share` divides the inherited cost across the active revenue endpoints under that structure. `direct` assigns 100% of the inherited shared cost to the relationship. Customer inherited cost detail shows the method and percentage used for each row so manually adjusted attribution remains auditable.

ROI Analysis treats RF Link `capex_usd` when present plus customer property CAPEX as the upfront investment required to serve the customer. This keeps the model useful for RF-fed customers, circuit-fed customers with property build costs, and redundant serving paths with their own payoff requirements. Customer install NRC is treated as upfront revenue that offsets the payback requirement. The ROI table is a snapshot model: it adds an 8% annual RF CAPEX maintenance reserve when RF CAPEX exists plus any customer maintenance reserve MRC as monthly OPEX, generates monthly revenue from contracted MRR, term years, term start/end dates, revenue escalator, and ramp months, subtracts commission as either a percentage of monthly revenue or a fixed monthly dollar amount, and compounds direct serving cost plus inherited donor cost upward by 3% annually starting in year 2. The ROI detail separates serving cost, maintenance reserve, commission, total attributed cost, and margin before CAPEX so the assumptions can be audited. The monthly schedule uses the same variables by month. Net cash after CAPEX is the sum of monthly escalated contribution margin plus install NRC minus total build CAPEX; ROI percentage is net cash after CAPEX divided by total build CAPEX. IRR uses the monthly cash-flow stream of upfront NRC less total build CAPEX at month 0 followed by monthly contribution margin for the selected term. NPV uses the same cash flows with a 10% annual discount rate. Margin multiple is total contribution margin before CAPEX divided by total build CAPEX. If there is no build CAPEX, NetOS still shows net cash, but ROI percentage, CAPEX/ARR, margin multiple, and IRR are not meaningful.

The customer detail **What If** calculator is intentionally not stored. It reuses the customer revenue schedule and lets the user enter a hypothetical Type 2 circuit, alternate RF build, or renewal scenario. This supports conversion decisions such as comparing an owned RF build against a leased circuit by modeling alternate monthly transport cost, build CAPEX, monthly OPEX/reserve, install NRC, and commission assumptions.

Subtended Link

A subtended link is the economic relationship between a parent infrastructure asset and a downstream revenue source or downstream asset.

Current NetOS fields:

FIELD	MEANING
parent_infrastructure_id	The structure/facility receiving attribution credit
downstream_infrastructure_id	Optional child structure/facility that rolls up to the parent
circuit_id	Optional circuit used by or associated with the relationship
rf_link_id	Optional RF Link serving a customer endpoint or downstream asset
customer_id	Optional customer record receiving or producing the endpoint service
customer_name	End customer or account name when the endpoint is customer revenue
endpoint_name	Site, subscriber, building, or endpoint served
relationship_type	Customer endpoint, child structure, backhaul, aggregation, etc.
transport_type	Fiber, PTP wireless, PtMP wireless, DIA, broadband, cross-connect, etc.
revenue_mrr_usd	Direct monthly revenue attributed to this link; overrides customer contracted MRR
cost_mrc_usd	Manual monthly cost override for this link when calculated allocation is unavailable or intentionally overridden
effective_from	First date the relationship is financially active
effective_to	Last date the relationship is financially active
allocation_method	How shared revenue/cost should be allocated
allocation_pct	Percent applied when using percentage allocation
inherited_allocation_method	How upstream facility and donor path cost should be attributed; defaults to revenue_pro_rata
inherited_allocation_pct	Percent applied when inherited_allocation_method is manual_pct
reparented_from_link_id	Prior relationship when a child/customer moves to a different parent

On an infrastructure detail page, the same relationship is shown from two directions:

VIEW	MEANING
Subtended Network	What this site serves downstream: customers, child structures, RF links, and circuits.
Upstream Donor & Paths	What serves this site upstream: donor structure, transport path, and attributed path cost.

VIEW	MEANING
Structure Financial Summary	The site-level financial view: upstream path cost, direct facility cost, downstream path/customer revenue, downstream cost, total cost, and margin.

Example: if BSI US Bank Cincinnati subtends Lakota Lakes POP over an Altafiber EVPL circuit, the BSI page shows Lakota Lakes POP in its subtended network. The Lakota page shows BSI US Bank Cincinnati as the upstream donor and the Altafiber circuit as the path/cost source.

The structure detail summary includes the upstream path cost for the site being viewed. Parent portfolio rollups do not double count that same path: the parent sees the parent-to-child path as its direct link cost, while the child detail page shows the same path as its upstream cost.

Attribution Principles

1. Attribute Revenue to the Structure That Enables Service

Revenue should be assigned to the lowest infrastructure asset that directly enables the service, then roll up through parents.

Example:

RELATIONSHIP	DIRECT REVENUE
Tower A -> Customer 1	\$500 MRR
Tower A -> Customer 2	\$750 MRR
POP 1 -> Tower A	\$0 direct revenue, but receives Tower A rollup

Tower A gets \$1,250 direct revenue. POP 1 gets \$1,250 downstream revenue through Tower A.

2. Keep Facility Costs on the Asset That Owns the Obligation

Facility lease, cabinet, power, roof rent, or tower rent belongs to the asset that has the obligation.

Example:

ASSET	DIRECT FACILITY COST
Rooftop A	\$350 MRC roof lease
POP 1	\$1,800 MRC cabinet and power

The rooftop cost stays on Rooftop A. POP 1 only receives it as downstream cost if Rooftop A is subtended by POP 1.

3. Keep Circuit, RF Link, and Electrical Costs on the Source Record, Then Allocate Through the Relationship

Circuit MRC belongs to the circuit record. RF Link MRC, tower rent MRC, and CAPEX belong to the RF Link record. Electrical service average monthly cost belongs to the electrical service record attached to the site. A subtended link can reference the transport source to explain which circuit or RF Link supports the relationship, while site summaries include attached electrical services as add-on cost.

In the hierarchy drilldown, NetOS should distinguish the relationship from the cost source. A row may be a customer endpoint, downstream structure, or backhaul relationship, while the cost source may be a circuit, RF Link, electrical service, or manual override. This makes it clear what the node represents and which object is contributing cost to that segment of the network.

NetOS should calculate the attributed link cost automatically when it has the required driver data. Users should select the allocation method and provide the needed driver, such as allocation percentage, committed bandwidth, or subscriber count. Manual link cost should be treated as an override or fallback, not the normal workflow.

Attribution depends on how the circuit or RF Link is used:

USE CASE	RECOMMENDED ALLOCATION
Circuit serves one downstream endpoint	Direct allocation, 100% of linked circuit MRC
RF Link serves one customer endpoint	Direct allocation, 100% of RF Link MRC plus tower rent MRC
Circuit serves multiple customers on one tower	Calculate by revenue share or subscriber count
PtMP sector serves multiple customers	Calculate by revenue share, bandwidth share, or subscriber count
Backhaul circuit serves a child tower	Direct to child structure, then roll up
Aggregation circuit supports many facilities	Calculate by bandwidth, revenue, or configured percentage
Cross-connect supports one carrier handoff	Direct to the relevant circuit or facility

4. Never Rewrite History When Relationships Change

When a child facility or customer endpoint moves from one parent to another:

1. Set `effective_to` on the old subtended link.
2. Create a new subtended link under the new parent.
3. Set `reparented_from_link_id` on the new link.
4. Leave circuit endpoint history intact unless the physical circuit endpoint actually changed.

This preserves historical reporting and makes month-by-month waterfall views possible.

5. One Active Attribution Parent per Target

A circuit, RF Link, customer record, or downstream infrastructure asset should not be active under two parents for the same reporting period. NetOS blocks overlapping active assignments so revenue and cost are not counted twice in the hierarchy.

To move an attributed target:

1. Edit the existing subtended relationship and set `effective_to` to the last valid date.
2. Add the target under the new parent with a later `effective_from` date.
3. Use `reparented_from_link_id` when preserving the explicit move history is important.

NetOS also blocks circular hierarchy assignments, such as making a parent facility subtended by one of its own children.

Standard Rollup Formula

For each infrastructure asset:

```
direct_revenue = sum(active subtended link revenue_mrr_usd, or customer contracted_mrr_usd when
link revenue is blank)
direct_link_cost = sum(active subtended link cost_mrc_usd)
direct_facility_cost = infrastructure.mrc_usd

downstream_revenue = sum(child infrastructure total_revenue)
downstream_cost = sum(child infrastructure total_cost)

total_revenue = direct_revenue + downstream_revenue
total_cost = direct_facility_cost + direct_link_cost + downstream_cost
gross_margin = total_revenue - total_cost
gross_margin_pct = gross_margin / total_revenue
```

The current NetOS implementation uses this structure for infrastructure economics. Future versions should apply the same logic by accounting period and allocation method.

Allocation Methods

Allocation methods should produce calculated cost whenever NetOS has enough data.



Current implemented behavior:

METHOD	CALCULATION
direct	100% of linked circuit MRC unless a manual override is entered
manual_pct	linked circuit MRC multiplied by allocation_pct unless a manual override is entered
revenue_share	Uses full linked source MRC until revenue-share driver fields are modeled
bandwidth_share	Uses full linked source MRC until committed or measured bandwidth driver fields are modeled
equal	Uses full linked source MRC until sibling relationship group driver fields are modeled

The Subtended Network form shows a calculated direct link cost preview. When a manual override is entered, the override wins. When manual_pct is selected, the percentage controls the attributed cost. When a future method such as bandwidth_share, revenue_share, or equal is selected before its driver fields are modeled, NetOS uses the full linked source MRC rather than dropping the transport cost to zero.

Inherited shared cost allocation is separate from direct link-cost allocation. The subtended network list shows both the inherited attribution method and the calculated inherited cost applied to each relationship. The form includes a live inherited cost calculator that recalculates on each form change. It compares all inherited methods side by side using the current relationship revenue or downstream structure rollup revenue, sibling subtended relationships at the parent site, and the parent site's current shared monthly cost pool. The shared monthly cost pool includes the parent facility MRC plus the upstream donor-chain cost currently serving that parent, after applying the upstream relationship's own inherited allocation method. Parallel transport rows to the same downstream structure or customer count the revenue target once for inherited allocation. This lets the user compare revenue pro-rata, equal share, manual percent, and direct assignment before saving the relationship.

The infrastructure detail page also shows an inherited cost balance for the subtended network. Manual percentage and direct inherited allocations consume the shared cost pool first. Remaining cost is then assigned to calculated rows, such as revenue pro-rata or equal share rows. Calculated rows can never consume more than the remaining balance after manual/direct rows. If multiple manual or direct rows request more than the available pool, NetOS gives precedence to the row most recently changed and caps older fixed rows to the remaining balance. If manual adjustments leave cost unassigned, NetOS calls out the unallocated balance. Users can assign the remaining balance to the other calculated subtended links pro-rata or equally, which saves explicit manual percentages on those rows for auditability.

METHOD	CALCULATION
revenue_pro_rata	Default. Relationship share equals customer revenue or downstream rollup revenue divided by total revenue under the structure
manual_pct	Upstream facility and donor path cost multiplied by inherited_allocation_pct
equal_share	Upstream facility and donor path cost divided equally across active revenue endpoints under the structure
direct	100% of upstream facility and donor path cost assigned to the relationship

Direct Allocation

Use when a cost or revenue belongs completely to one relationship.

Example:

PARENT	LINK	REVENUE	COST	ALLOCATION
Tower A	Customer 1 via PTP radio	\$600	\$0	direct, 100%
Tower A	Dedicated DIA resale	\$1,200	\$650	direct, 100%

Percentage Allocation

Use when a known percentage of a shared cost belongs to a relationship.

Example:

SHARED CIRCUIT	TOTAL MRC	LINK	ALLOCATION %	ATTRIBUTED COST
POP 1 backhaul	\$2,000	Tower A	40%	\$800
POP 1 backhaul	Tower B	35%	\$700	
POP 1 backhaul	Tower C	25%	\$500	

Revenue Share Allocation

Use when a shared cost should follow revenue contribution.

Example:

LINK	REVENUE	SHARE OF REVENUE	ALLOCATED BACKHAUL COST
Customer A	\$500	25%	\$250
Customer B	\$1,000	50%	\$500
Customer C	\$500	25%	\$250

This is useful when capacity consumption is unknown but customer revenue is known.

Bandwidth Allocation

Use when a shared transport cost should follow committed or measured bandwidth.

Example:

LINK	COMMITTED MBPS	SHARE OF BANDWIDTH	ALLOCATED COST
Customer A	100 Mbps	10%	\$100
Customer B	400 Mbps	40%	\$400
Tower B uplink	500 Mbps	50%	\$500

This becomes more accurate after integration with monitoring or billing systems.

Subscriber Count Allocation

Use for PtMP systems where a shared AP/radio/backhaul supports many similar customers.

Example:

SECTOR	SUBSCRIBERS	SHARED SECTOR COST	COST PER SUBSCRIBER
Tarana Sector 1	20	\$1,000	\$50

Each customer link receives \$50 of shared sector cost.

Worked Examples

The worked examples below use the same basic pattern: revenue enters through customer or downstream links, direct facility costs stay on the owning asset, and transport costs are attributed through the relationship that uses the transport.



Example 1: Rooftop With PtMP Wireless Customers

Rooftop A has a Tarana base node serving five customer endpoints.

ITEM	AMOUNT
Rooftop lease	\$300 MRC
Power	\$75 MRC
Backhaul circuit	\$900 MRC
Customer revenue	\$2,750 MRR

Recommended modeling:

RECORD	MODEL
Rooftop A	Infrastructure asset with <code>mrc_usd = 375</code>
Backhaul	Circuit with <code>mrc_usd = 900</code>
Customer links	Five subtended links under Rooftop A
Backhaul allocation	Either direct to Rooftop A or allocated across customer links

Simple rooftop economics:

```

direct_revenue = 2750
direct_facility_cost = 375
direct_link_cost = 900
total_cost = 1275
gross_margin = 1475
gross_margin_pct = 53.64%

```

Example 2: Tower Subtended by a Data Center Over Fiber

Data Center POP 1 aggregates Tower A over an EVPL circuit. Tower A serves customers.

ITEM	AMOUNT
POP 1 cabinet/power	\$1,800 MRC
EVPL from POP 1 to Tower A	\$1,200 MRC
Tower A lease/power	\$600 MRC
Tower A customer revenue	\$4,500 MRR

Recommended modeling:

RECORD	MODEL
POP 1	Infrastructure asset with <code>mrc_usd = 1800</code>
Tower A	Infrastructure asset with <code>mrc_usd = 600</code>
EVPL	Circuit with A location POP 1 and Z location Tower A
POP 1 -> Tower A	Subtended link with <code>downstream_infrastructure_id = Tower A</code> , <code>circuit_id = EVPL</code> , <code>cost_mrc_usd = 1200</code>
Tower A -> customers	Customer subtended links totaling \$4,500

Tower A economics:

```

direct_revenue = 4500
direct_facility_cost = 600
total_revenue = 4500
total_cost = 600
gross_margin = 3900

```

POP 1 economics:

```

direct_revenue = 0
direct_link_cost = 1200
direct_facility_cost = 1800
downstream_revenue = 4500
downstream_cost = 600
total_revenue = 4500
total_cost = 3600
gross_margin = 900

```

Example 3: Aggregation Facility With Multiple Child Towers

Aggregation Facility A subtends three towers. Each tower has its own customers and costs.

CHILD	REVENUE	FACILITY COST	TRANSPORT COST FROM PARENT
Tower A	\$4,500	\$600	\$1,200
Tower B	\$3,200	\$500	\$900
Tower C	\$2,100	\$400	\$700

Parent facility direct cost is \$2,000 MRC.

Waterfall:

LAYER	REVENUE	COST	MARGIN
Tower A	\$4,500	\$600	\$3,900
Tower B	\$3,200	\$500	\$2,700
Tower C	\$2,100	\$400	\$1,700
Parent transport to children	\$0	\$2,800	-\$2,800
Parent facility	\$0	\$2,000	-\$2,000
Total parent rollup	\$9,800	\$6,200	\$3,600

This view answers whether the aggregation facility is economically successful after the child towers and required transport are included.

Example 4: Reparenting a Child Tower

Tower A originally rolls up to POP 1. On 2026-05-01, it is moved to POP 2.

LINK	PARENT	CHILD	EFFECTIVE FROM	EFFECTIVE TO
Link 1	POP 1	Tower A	2025-01-01	2026-04-30
Link 2	POP 2	Tower A	2026-05-01	null

Link 2 should store `reparented_from_link_id = Link 1`.

Reports for April 2026 attribute Tower A to POP 1. Reports for May 2026 attribute Tower A to POP 2. The physical EVPL circuit endpoint should only change if the actual circuit was moved.

Waterfall Reporting Model

A complete waterfall should show how revenue and expenses flow from customer endpoints to structures and then to parent aggregation points.

Use this example scenario:

1. Three customers are served from Tower A and generate \$4,500 MRR.
2. Tower A has \$600 MRC of direct facility cost.
3. POP 1 feeds Tower A with an EVPL backhaul circuit that costs \$1,200 MRC.
4. POP 1 has its own cabinet and power cost of \$1,800 MRC.

The waterfall answers two different questions. First, is Tower A locally profitable before upstream cost? Yes: \$4,500 revenue - \$600 facility cost = \$3,900 margin. Second, is POP 1 profitable after carrying Tower A, the EVPL backhaul, and its own facility cost? Yes, but only \$900 margin.



Recommended levels:

LEVEL	DESCRIPTION
Customer endpoint	End user or service endpoint revenue
Access link	PTP, PtMP, fiber drop, DIA resale, broadband, or cross-connect used by the endpoint
Serving structure	Tower, rooftop, MDU, POP, or data center directly serving the endpoint
Backhaul/transport	Circuits or wireless links connecting the structure upstream
Parent facility	Aggregation site, data center, POP, carrier hotel

LEVEL	DESCRIPTION
Regional/network rollup	Market, state, provider, service type, or entire infrastructure portfolio

Example waterfall output for the scenario above:

ROW	WHAT IT MEANS	REVENUE	COST	DOWNSTREAM	TOTAL COST	MARGIN
Customer links	The customer subtended links attached to Tower A	\$4,500	\$0	\$0	\$0	\$4,500
Tower A	The serving structure keeps the customer revenue and owns tower lease/power	\$4,500	\$600	\$0	\$600	\$3,900
POP 1 -> Tower A EVPL	The backhaul relationship required to support Tower A	\$0	\$1,200	\$0	\$1,200	-\$1,200
POP 1	The parent facility receives Tower A's economics and adds POP cabinet/power cost	\$0	\$1,800	\$4,500 revenue / \$1,800 cost	\$3,600	\$900
Market rollup	The portfolio layer sums POP 1 and other infrastructure assets in the market	\$0	\$0	\$4,500 revenue / \$3,600 cost	\$3,600	\$900

The key point is that the same revenue can look very profitable at the serving structure level, but the parent view is more honest about the full cost to deliver that revenue. Tower A's local margin is \$3,900. POP 1's rollup margin is \$900 after including the EVPL and POP costs.

Drilldown Flow Labels

The financial drilldown exposes row-level revenue and expense detail so users can understand both the amount and the direction of attribution.

LABEL	MEANING
Rolls Up	Revenue produced by a customer endpoint or child structure contributes upward to the serving structure, parent POP, market, and portfolio rollup.
Flows Down	Cost from a circuit, RF Link, or other transport path is assigned downward into the structure or endpoint that depends on that path.
Local	Cost belongs to the current structure itself, such as facility MRC, cabinet cost, rooftop rent, or other site-level obligation.

For example, Lakota Lakes POP can show customer endpoint revenue as **Rolls Up** because the customer revenue contributes to Lakota's total revenue. Its upstream EVPL path can show as **Flows Down** because that transport cost is assigned from the donor path into Lakota's cost stack.

Accounting Periods

Future reporting should evaluate active relationships by period:

```
relationship is active for period when:  
effective_from <= period_end  
and (effective_to is null or effective_to >= period_start)
```

Monthly reports should prorate revenue and cost when a relationship starts or ends mid-month. Until proration is implemented, NetOS should use whole-month attribution and clearly label reports as non-prorated.

Recommended period rules:

SCENARIO	RULE
Link active full month	Attribute full monthly value
Link starts mid-month	Prorate by active days, future state
Link ends mid-month	Prorate by active days, future state
Link reparented	Old parent receives value through old <code>effective_to</code> ; new parent receives value from new <code>effective_from</code>
Facility contract escalates	Apply escalated MRC beginning on escalation date
Circuit billing variance exists	Use invoice actuals for closed accounting periods once invoice ingestion exists

Recommended Data Entry Workflow

Adding a Customer Endpoint Under a Structure

1. Open the serving infrastructure asset.
2. Add a subtended link.
3. Enter customer name, endpoint name, transport type, revenue MRR, and direct link cost if known.
4. Link an existing circuit if the customer service depends on one specific circuit.
5. Set `effective_from`.
6. Save supporting documents if available.

Adding a Child Facility Under a Parent

1. Open the parent infrastructure asset.
2. Add a subtended link with `downstream_infrastructure_id`.
3. Link the transport circuit if one exists.
4. Put the transport cost either on the circuit record or in `cost_mrc_usd` for the subtended link, depending on whether it is already represented as a circuit.
5. Set the relationship effective date.

Changing a Parent Relationship

1. Open the current parent relationship.
2. Set `effective_to` to the last valid date.

3. Create a new subtended link under the new parent.
4. Set `effective_from` to the first valid date.
5. Link the new record back to the prior record with `reparented_from_link_id`.
6. Only update circuit endpoints if the physical circuit path changed.

Open Modeling Decisions

These should be resolved before production financial reporting:

DECISION	OPTIONS
Source of customer revenue	Manual entry first, then Sonar integration
Cost source of truth	Contract MRC first, invoice actuals after invoice ingestion
Shared transport allocation	Direct, percentage, revenue share, bandwidth, subscriber count
Wireless equipment costs	Expense as direct link cost, facility cost, or depreciated asset
NRC treatment	Exclude from MRR margin, amortize across term, or report separately
Mid-month proration	Whole-month initially, daily proration later
Tax/surcharge treatment	Include in actual invoice cost or separate pass-through category

Current Implementation Status

Implemented:

- Infrastructure assets have direct `mrc_usd` and `nrc_usd`.
- Circuits have direct `mrc_usd` and `nrc_usd`.
- Infrastructure detail supports multiple supporting documents.
- Infrastructure detail supports subtended links.
- Subtended links can reference child infrastructure, customer endpoints, circuits, and RF Links.
- Subtended links include effective dates, allocation method, allocation percent, and reparent lineage.
- RF Link calculated monthly cost uses RF MRC plus tower rent MRC. CAPEX is tracked on the link and is not included in monthly margin unless a future amortization model includes it.
- Infrastructure economics roll up direct revenue, direct costs, downstream revenue, downstream costs, total cost, total revenue, and margin.

Next recommended implementation steps:

1. Add a guided "Change Parent" action that closes the old link and creates the new one atomically.
2. Add edit support for existing subtended links in the UI.
3. Add accounting-period filters and historical waterfall reporting.
4. Add allocation calculators for revenue share, bandwidth, and subscriber count.
5. Integrate Sonar customer revenue as the revenue source of truth.
6. Integrate invoice actuals as the expense source of truth for closed periods.

7. Add exportable waterfall reports by asset, market, provider, and month.

Roadmap & Known Gaps

Xiber NetOS — Roadmap & Known Gaps

What's built, what's missing, and the recommended path forward.

Current Phase: Foundation Prototype

The current build delivers a working UI and API for circuit lifecycle visibility. Core CRUD, bulk edit/delete, mapping, dashboards, import, audit activity, and the admin feedback queue are functional. Authentication hardening, financial accuracy, renewal automation, and document ingestion remain the primary gaps before production use.

Feature Inventory

What's Built

AREA	DETAILS
Monorepo scaffold	apps/api , apps/web , apps/mcp , packages/shared , infra/docker
Database schema	13+ tables, PostGIS geometry, pgvector embeddings, materialized financial view
Migrations	Alembic baseline + infrastructure + agreements migrations
API	FastAPI with OpenAPI 3.1, circuit CRUD, bulk actions, map, dashboard, import, providers, infrastructure
Circuit table	Search, filter by carrier/status/type, bulk edit/delete, click-through to detail
Circuit detail	Full attributes, endpoints, carrier info, NOC contacts, contract terms, lifecycle history
Map	A → Z circuit lines, endpoint dots, infrastructure markers, colored by provider, thickness by bandwidth
Dashboard	17 KPI cards/panels including spend by carrier, service mix, renewal pipeline, at-risk circuits
Timeline	Contract-term Gantt with install markers, renewal deadlines, today line, state colors
Topology	Endpoint graph with logical/geographic layouts, SPOF detection, carrier-colored edges
Service providers	Portal links, NOC phones, account managers, circuit/infra counts, ETF exposure
Infrastructure	Data centers, towers, colos, rooftops, carrier hotels, POPs, offices, and aggregation facilities with costs, terms, and bulk edit/delete
Electrical	Utility service accounts attached to infrastructure or customer sites, with provider, meter, service details, average monthly cost, monitoring link, and documents
Financial / P&L	

AREA	DETAILS
	Per-circuit modeled revenue, margin, ETF exposure, carrier spend rollups, and infrastructure attribution foundation
Address lookup	Search circuits by address, endpoint, city, state, carrier, or circuit ID
Mobile field view	Phone-friendly circuit lookup with NOC call, copy ID, and map links
Admin / audit	Recent activity trail plus bug/feature queue with priority, status, and progress comments
CSV/XLSX import	Heuristic column mapping, staging, upsert by natural key, auto-stub carriers/endpoints
Sample data	4 carriers, 10 endpoints, 8 circuits with contracts and lifecycle events
MCP skeleton	<code>list_circuits</code> , <code>get_circuit</code> working; <code>get_renewals_due</code> , <code>get_etf_exposure</code> as placeholders
RBAC	8 roles defined, dev shim accepts <code>x-user-role</code> header
Docker dev stack	PostgreSQL 16, Redis 7, API, Web with hot reload

Gap Analysis

1. Authentication & Authorization

	DETAIL
-	-----
Current	Header-based dev shim (<code>x-user-role: exec</code>) plus support for common SSO email headers when present. Direct/ demo mode falls back to <code>dev@xiber.com</code> .
Impact	Cannot deploy to production. No audit trail of who did what.
Needed	Microsoft Entra OIDC JWT validation on API. Role assignment from Entra groups/claims. Session handling in Next.js (cookie-based, server-side). Token refresh flow.
Depends on	Entra app registration (see Public Hostname & SSO)

2. Audit Logging

	DETAIL
-	-----
Current	Activity logging exists for request/write activity, feedback, and bulk actions. <code>created_by</code> / <code>updated_by</code> are populated on newer write paths.
Impact	Basic accountability exists, but sensitive read tracking and field-level redaction are incomplete.
Needed	Complete field-level before/after coverage for all write paths. Field-level read audit for sensitive data (ETF formulas, banking, MD5 keys). Sensitive field redaction on read by role.

3. CSV/XLSX Import Enhancements

	DETAIL
-	-----
Current	Heuristic column mapping. Staging rows. Commit valid rows. Upsert by natural key.
Impact	Column mapping is fragile for non-standard headers. No way to fix staging errors in-app.
Needed	Claude Opus 4.7 mapping (send headers + 5 sample rows, get canonical mapping). Saved mappings by (user, source_signature). Inline staging row editor. Diff preview before updating existing circuits. Explicit confirmation dialog before overwriting financial fields.

4. PDF Contract Ingestion

	DETAIL
-	-----
Current	Contract and AgreementExtraction schemas exist. No upload or extraction pipeline.
Impact	Contract data must be entered manually or via CSV.
Needed	PDF upload endpoint with S3/MinIO object storage. pdfplumber text extraction with pytesseract OCR fallback. Claude Opus 4.7 structured extraction (carrier, term, MRC, ETF formula, renewal terms). Confidence scoring with source quotes and page numbers. Side-by-side approval UI (PDF viewer left, extracted fields right). Mandatory human approval for MRC, ETF, and term fields before commit.

5. Renewal Engine

	DETAIL
-	-----
Current	Renewal states computed on-the-fly in API/frontend. No persistent tracking or notifications.
Impact	No proactive alerts. Renewal deadlines can be missed.
Needed	Nightly Celery job to compute and persist renewal states (stable → watch → active → critical → overdue → decided). Materialized view or state table refresh. Email, Slack, and Monday.com notifications on state transitions. Decision workflow UI (renew / negotiate / terminate / port). DocuSeal termination notice generation. Test-mode fast-forward for QA.
State machine	stable (>180d) → watch (90–180d) → active (30–90d) → critical (<30d) → overdue (past deadline) → decided

6. Dashboard & Financial Accuracy

	DETAIL
-	-----
Current	Circuit-level modeled MRR uses MRC * 1.35 placeholder multiplier. Infrastructure detail now has subtended links, direct/downstream revenue and cost rollups, relationship effective dates, and circuit-linked attribution records.
Impact	

	DETAIL
	Circuit margin remains illustrative until Sonar revenue is connected. Infrastructure economics can be modeled manually now, but allocation methods, period reporting, and actual invoice/revenue integrations are still needed for production financial reporting.
Needed	Sonar integration to pull actual customer revenue attribution. Real margin calculation using customer MRR, carrier MRC, facility MRC, transport allocation, and invoice actuals. Full infrastructure waterfall view by asset, parent/child relationship, market, provider, service type, and accounting period. ETF exposure formulas and renewal projection forecasts using contract escalators.

7. Map Enhancements

	DETAIL
-	-----
Current	A → Z lines with provider coloring and bandwidth thickness. Provider/type/status filters.
Impact	Map is useful but isolated from table view. No external data overlays.
Needed	Polygon/lasso selection tool. Shared filter state with circuit table. Wisdm tower layer overlay. Sonar property/coverage layer overlay. Tile hosting decision (MapTiler vs. self-hosted). Cluster markers for dense endpoint areas.

8. Topology Enhancements

	DETAIL
-	-----
Current	Endpoint graph with logical/geographic layouts, SPOF detection by node degree.
Impact	Layout resets on reload. No true networking detail.
Needed	Drag/drop layout with save per user and shared default. True networking fields: VLANs, BGP sessions, router interfaces, handoff ports, IP blocks. Better redundancy modeling (path diversity, not just node degree).

9. Financial / P&L View

	DETAIL
-	-----
Current	Implemented with per-circuit modeled P&L, carrier spend, margin, ETF exposure, and the first infrastructure subtended-link economics model.
Needed	Sonar revenue attribution, invoice actuals, carrier/customer/market/service rollups, infrastructure waterfall reports, renewal projection forecast, allocation calculators, and export to XLSX.

10. Address Lookup

	DETAIL
-	-----
Current	Implemented text search across circuit, carrier, endpoint, city, state, and address fields.

	DETAIL
Needed	Integration with Xiber Service Address Availability Checker. Nearby endpoint/circuit search by radius. Quote-history cost-to-serve estimate.

11. Mobile Field View

	DETAIL
-	-----
Current	Implemented responsive field view with search, one-tap NOC call, copy circuit ID, and map app links.
Needed	Add infrastructure asset support, recent events, paired-circuit context, and offline-friendly behavior.

12. Invoice Ingestion

	DETAIL
-	-----
Current	Invoice and InvoiceLineItem schemas exist. No ingestion pipeline.
Needed	Mailbox polling (M365 Graph) or manual upload. PDF invoice parsing with line item extraction. Automatic matching to circuit by carrier circuit ID. Variance detection (invoiced MRC vs. contracted MRC). Auto-create Monday.com task for finance team on variance.

13. Vendor Portal Connectors

	DETAIL
-	-----
Current	Not implemented.
Needed	Lumen Control Center API connector. Zayo Tranzact API connector. Cogent portal connector. Vault/Doppler credential storage. Playwright scraping fallback where APIs are unavailable.

14. MCP Full Coverage

	DETAIL
-	-----
Current	2 working tools, 2 placeholders.
Needed	<code>get_carrier_summary</code> , <code>get_circuit_pl</code> , <code>find_circuits_at_address</code> , <code>get_outage_history</code> , <code>get_renewal_pipeline</code> . Write tools with explicit agent confirmation flow. Register with XOS MCP gateway for team-wide access.

Recommended Build Phases

PHASE	FOCUS	WHY FIRST
1	Real Entra authentication	Cannot deploy without identity. Blocks audit logging.
2	Audit logging	Required for compliance and accountability.
3	Renewal engine + notifications	Highest business value — prevents missed deadlines and auto-renew traps.
4	Contract CRUD + PDF ingestion	Enables self-service contract management, reduces manual data entry.
5	Financial/P&L + Sonar integration + infrastructure waterfall	Real margin visibility, accurate business intelligence across circuits, structures, child facilities, and customer endpoints.
6	Import enhancements (AI mapping, editor)	Reduces friction for bulk data onboarding.
7	Invoice ingestion + variance detection	Catches billing errors, automates finance workflows.
8	MCP full coverage	Enables XOS agents to query NetOS data for automated reporting and decisions.
9	Map + topology enhancements	Improved operational visualization.
10	Mobile, address lookup, exports	Quality-of-life features for field and sales teams.
11	Vendor portal connectors	Automated circuit status sync from carrier portals.

Technical Debt

ITEM	NOTES
Celery workers	Dependencies installed, workers not yet running. Needed for renewal engine and invoice polling.
Redis	Running but only used for Celery broker. Could also cache dashboard aggregations.
pgvector	Extension installed but no embeddings generated yet. Future use for semantic circuit search.
Error handling	API returns generic 500s in some edge cases. Need structured error responses.
Test coverage	No automated tests. Need unit tests for import logic, API endpoints, and renewal state machine.
CI/CD	No pipeline. Need GitHub Actions or similar for lint, test, build, deploy.