

2026 WTR/POP-BA/RNP



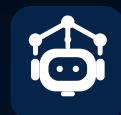
OpsSteward @ AmLight: Putting AI to Work in Network Operations

Jeronimo “Jab” Bezerra jbezerra@fiu.edu

IT Associate Director Florida International University

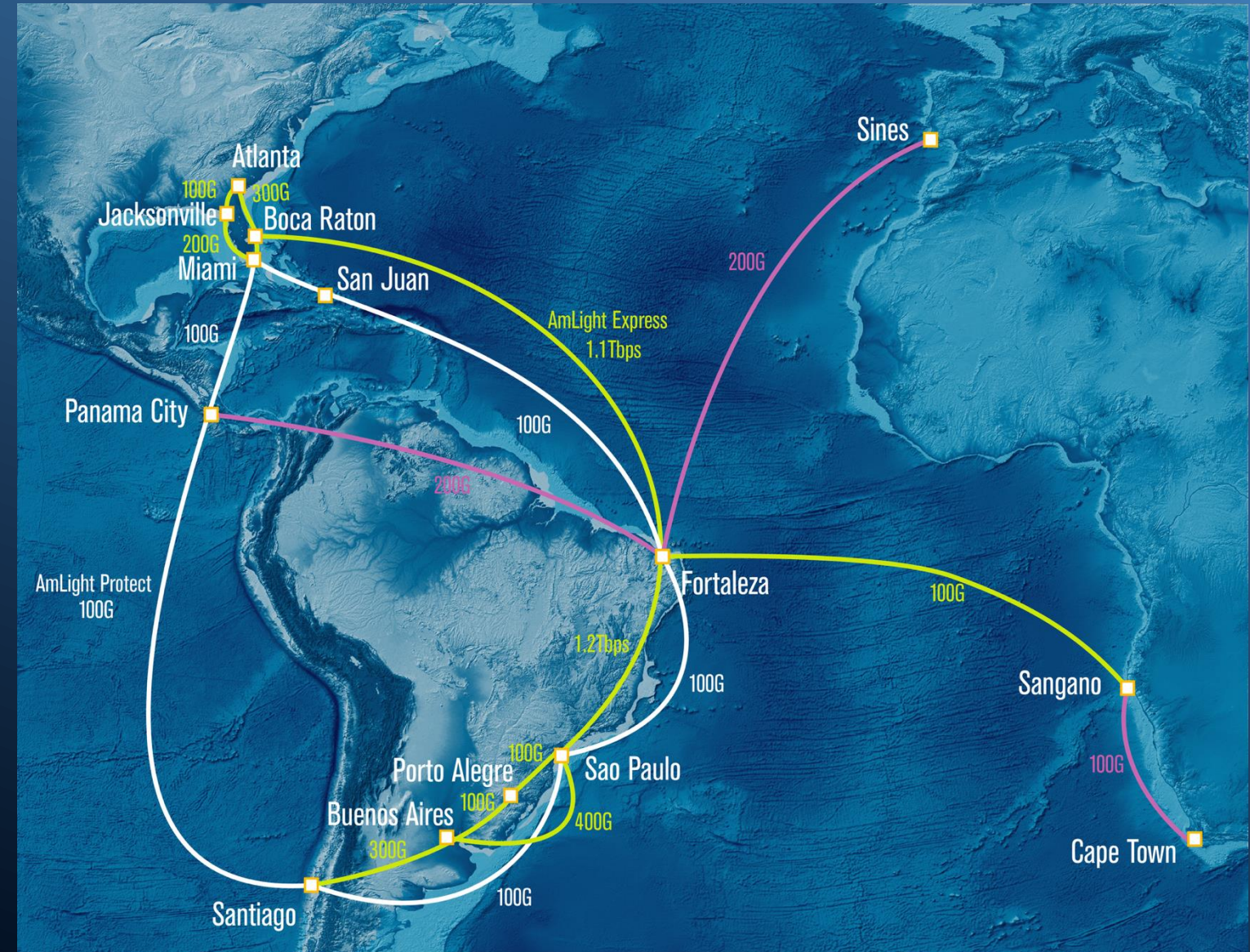
Outline

- Introducing AmLight
- AmLight in Numbers
- Our Science Driver
- Some AI concepts
- Introducing OpsSteward
- Future Work



Introducing AmLight Research and Education Network

- A distributed academic exchange point built to enable collaboration among Latin America, Africa, and the U.S.
 - Members: FIU, AURA, Vera Rubin Observatory, **RNP**, **Rednesp**, RedClara, REUNA, FLR, SANReN, TENET, and Internet2
- Supported by NSF and members since 2002
- 5Tbps of total capacity
- NAPs: Florida(3), Atlanta, Brazil(2), Chile, Puerto Rico, Argentina, Panama, and South Africa
- Infrastructure managed by a homemade SDN controller: Kytos-ng (github.com/kytos-ng)



Programmable Infrastructure and Telemetry in numbers

Network in numbers:

- Connectivity:
 - 1.2 Tbps between US and Latin America
 - 1.2 Tbps to U.S.'s IXPs
 - 41x 100G network links
 - ~38 P4-based programmable switches
 - 6x Routers
 - 7x Optical Data Center Interconnect Transponders
- Network measurement instrumentation
 - 11x PerfSONAR nodes
 - 1x Traffic Generator
 - 6x SmartNICs
 - Dedicated Programmable switches

Telemetry in numbers

- Per-packet network telemetry:
 - **~5 Mpps // 92 TB/day**
- Centralized Syslog
 - 187 devices // **~977,000 entries/day**
- + SNMP / sFlow / IPFIX / JTI / gNOI / gRPC
- **15+ Grafana dashboards**, several Zabbix dashboards, customized scripts
- XDP/EBPF, Netronome SmartNICs, Nvidia BlueField 3 DPUs collecting telemetry

Use Case: Vera Rubin Observatory Operation

- The Vera Rubin camera takes a picture of the southern sky every 30 seconds, and produces a 13GB dataset
 - Each dataset must be transferred **from Chile to California within 7 seconds**, inside the 30 second transfer window
- Constraints:
 - Distance from the Chile to California: 12,000 miles and RTT is 220+ ms
 - 0.00001% of packet loss will compromise the Rubin Observatory image transfer workflow
- Challenges:
 - How to operate the network to guarantee deliver of the 13GB dataset within the 7s window?
 - How to recover from events and continue to deliver the 13GB datasets on time?





AmLight: 20+ Years of Innovation

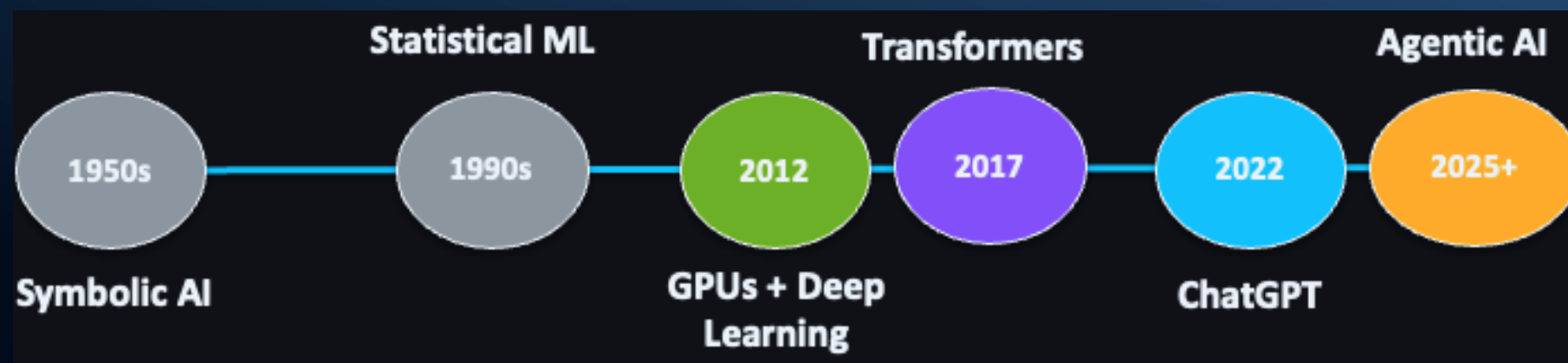
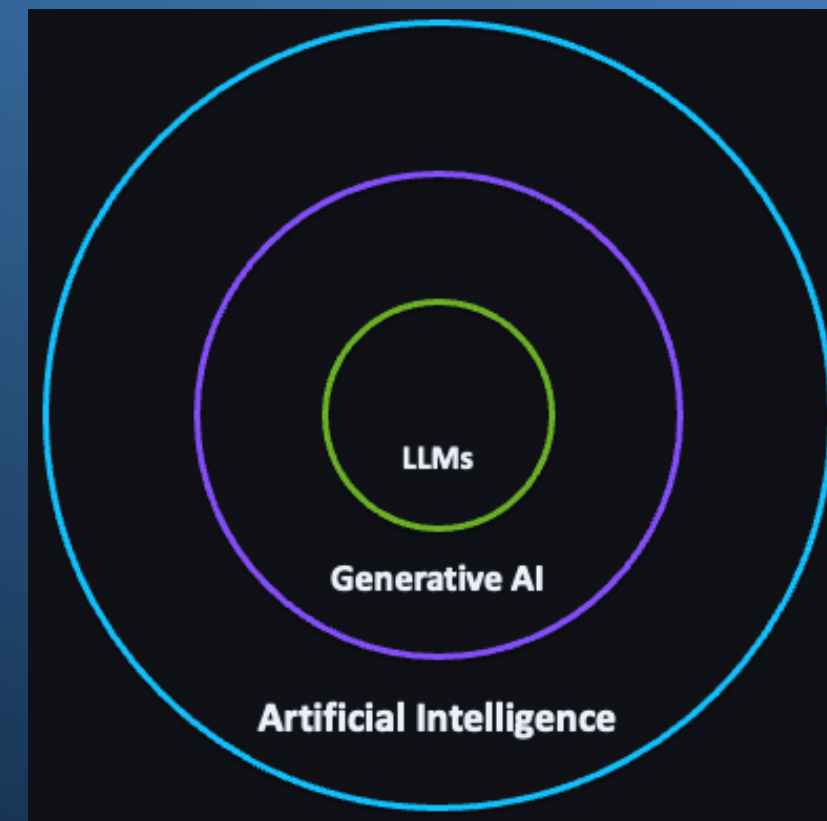
- 2005–2010: Building international R&E connectivity
- 2010: Pioneer using **coherent alien waves** on legacy submarine systems
- 2013: **Dynamic circuits** using OSCARS / NSIv2 in production
- 2014: Migration to Software-Defined Networking (**SDN**)
- 2018: Acquired **optical spectrum** on a submarine system (Monet)
- 2020: Programmable switches with P4 and **per-packet network telemetry** (INT)
- 2021: Deployment of a homemade network orchestrator: Kytos-ng SDN
- 2024: Use of traffic generators to continuously evaluate link quality
- 2026: Activated **a 1Tbps optical channel** between the U.S. and Latin America
- *What should we do next?*

Let's review some AI concepts.

Understanding AI vs GenAI vs LLM

- Artificial Intelligence

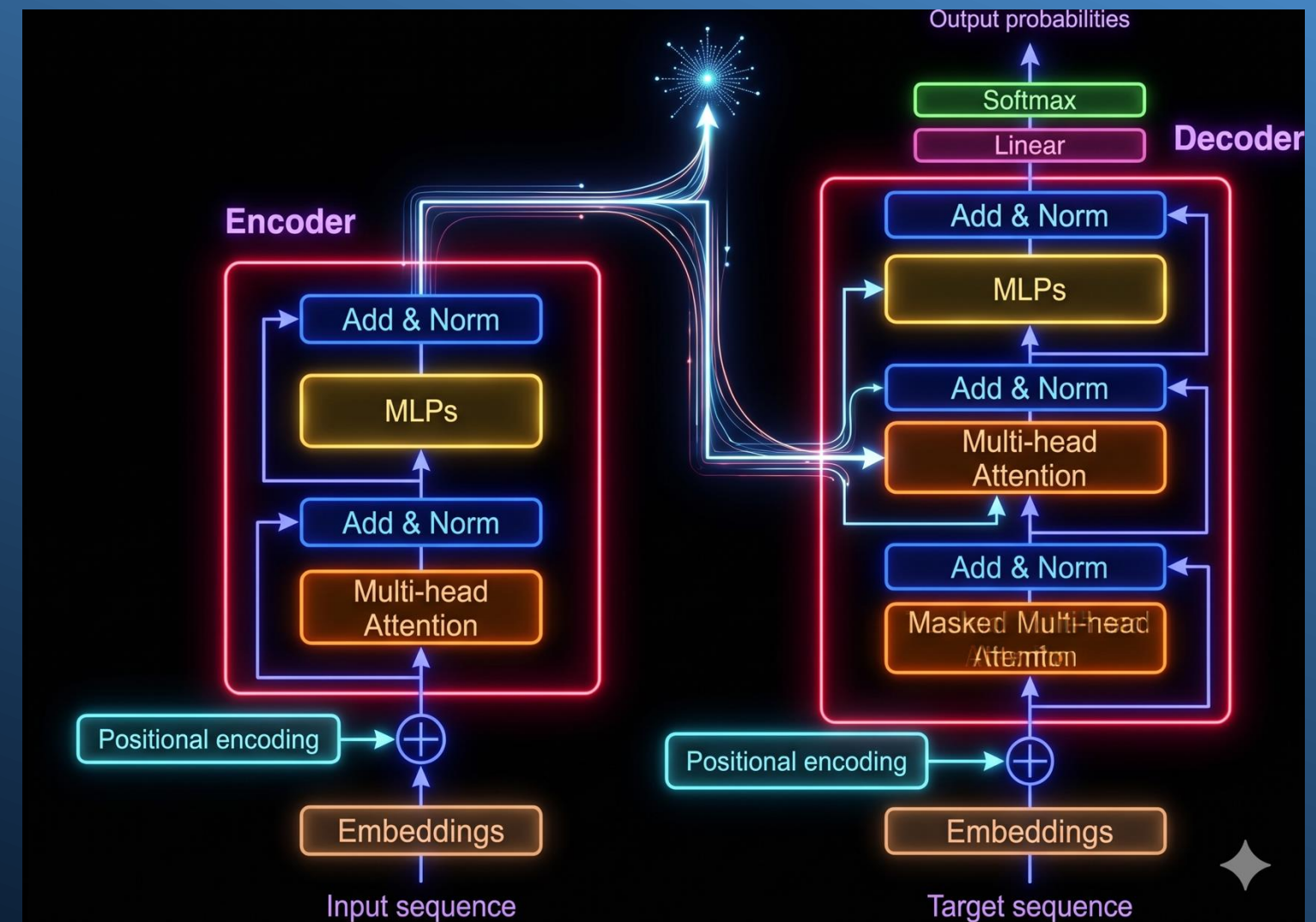
- Traditional ML
- Computer Vision
- Robotics
- Expert Systems
- Generative AI
 - Image Generation
 - Audio Generation
 - **LLMs**



The Transformer

The Evolution of Language Models

- From Sequential to Parallel:
 - Older models processed language sequentially and lost context over time
 - Transformers enable **parallelized training** and efficient scaling.
- **The Power of Attention:** Introduced in the 2017 paper '**Attention is All You Need**', this mechanism **allows words to "pay attention" to related words**, handling long-range context.
- Impact: Like intelligent path selection in a network, this architecture enabled the development of modern Large Language Models (LLMs).



Attention in Action

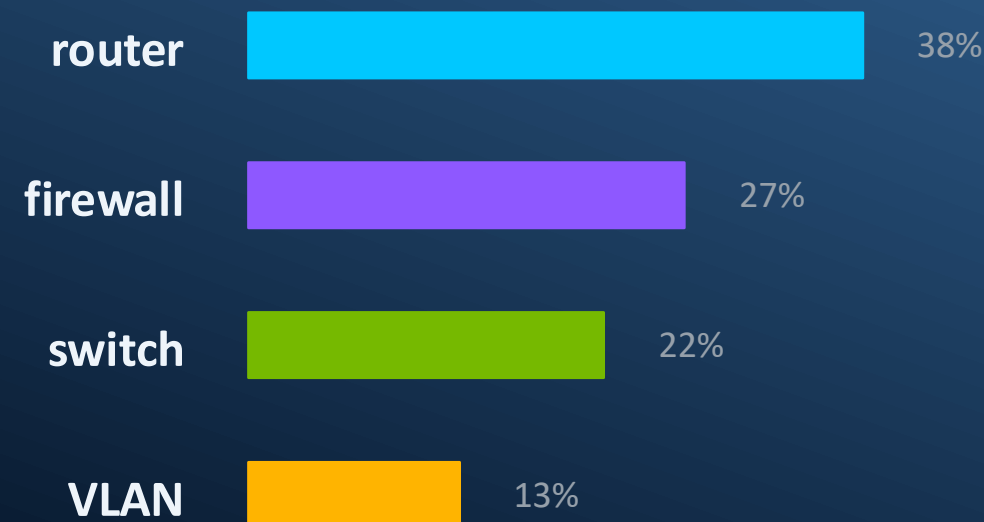
"The router forwarded traffic to Miami because IT had lower latency."

-> The model dynamically routes 'IT' to 'router', bypassing 'traffic' and 'Miami'.

How LLMs Work

- At the core, LLMs predict likely next tokens based on context.
- They are probabilistic systems, not deterministic databases.

“The network engineer configured the...”



LLMs are probabilistic completion engines.

Context Window

- LLMs have no memory
 - Context Window is the “memory” in “conversation”
- The model only “remembers” what fits in context.
 - Active working memory limit.
- Large context windows:
 - increase cost, reduce focus, and lead to hallucinations
- Hallucination = Buffer overflow/dilution of attention.
- Suggestion: Stay under 150-200k tokens

128k
Typical Token Limit

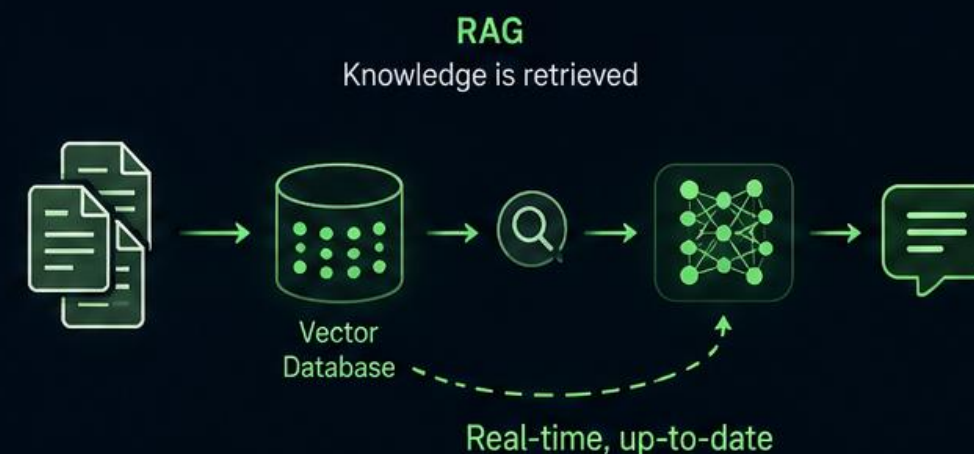
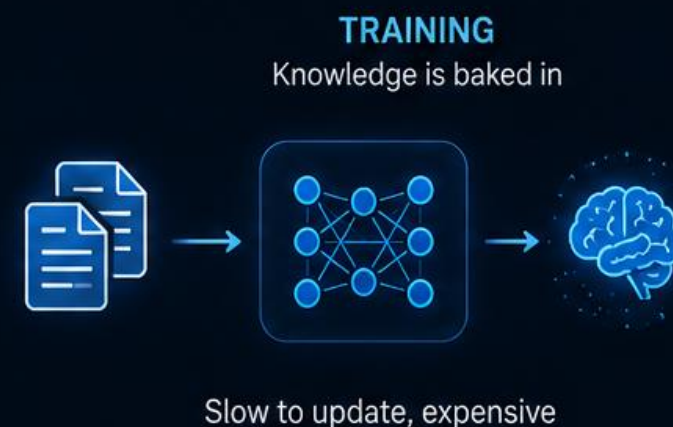
Extending the power of LLMs

1. Training vs RAG

Two paths to knowledge.

● Training = Internalize

● RAG = Retrieve



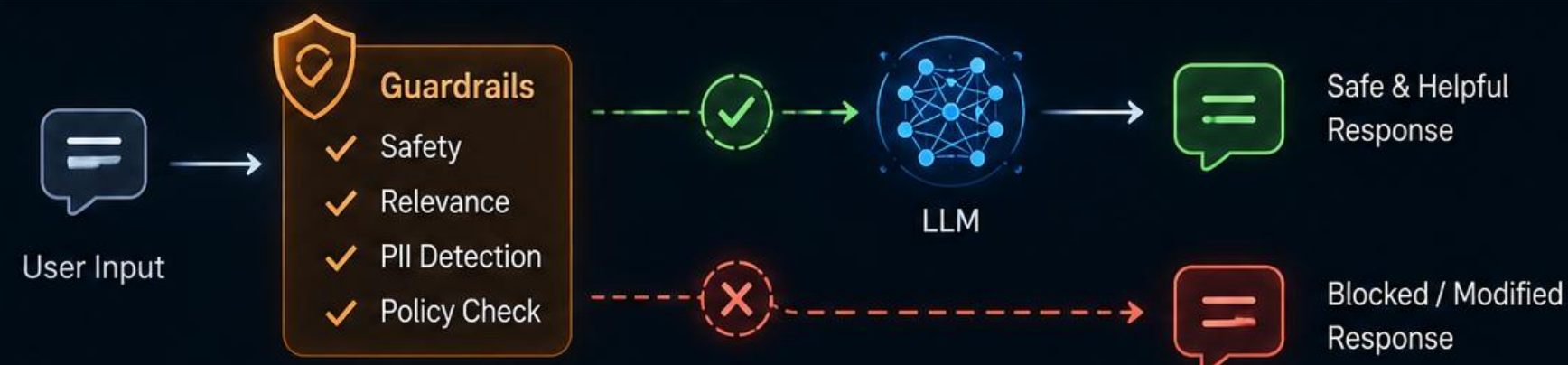
2. MCP (Model Context Protocol)

A standard way for models to connect to tools and data.



3. Guardrails

Keep AI outputs accurate, safe, and aligned.



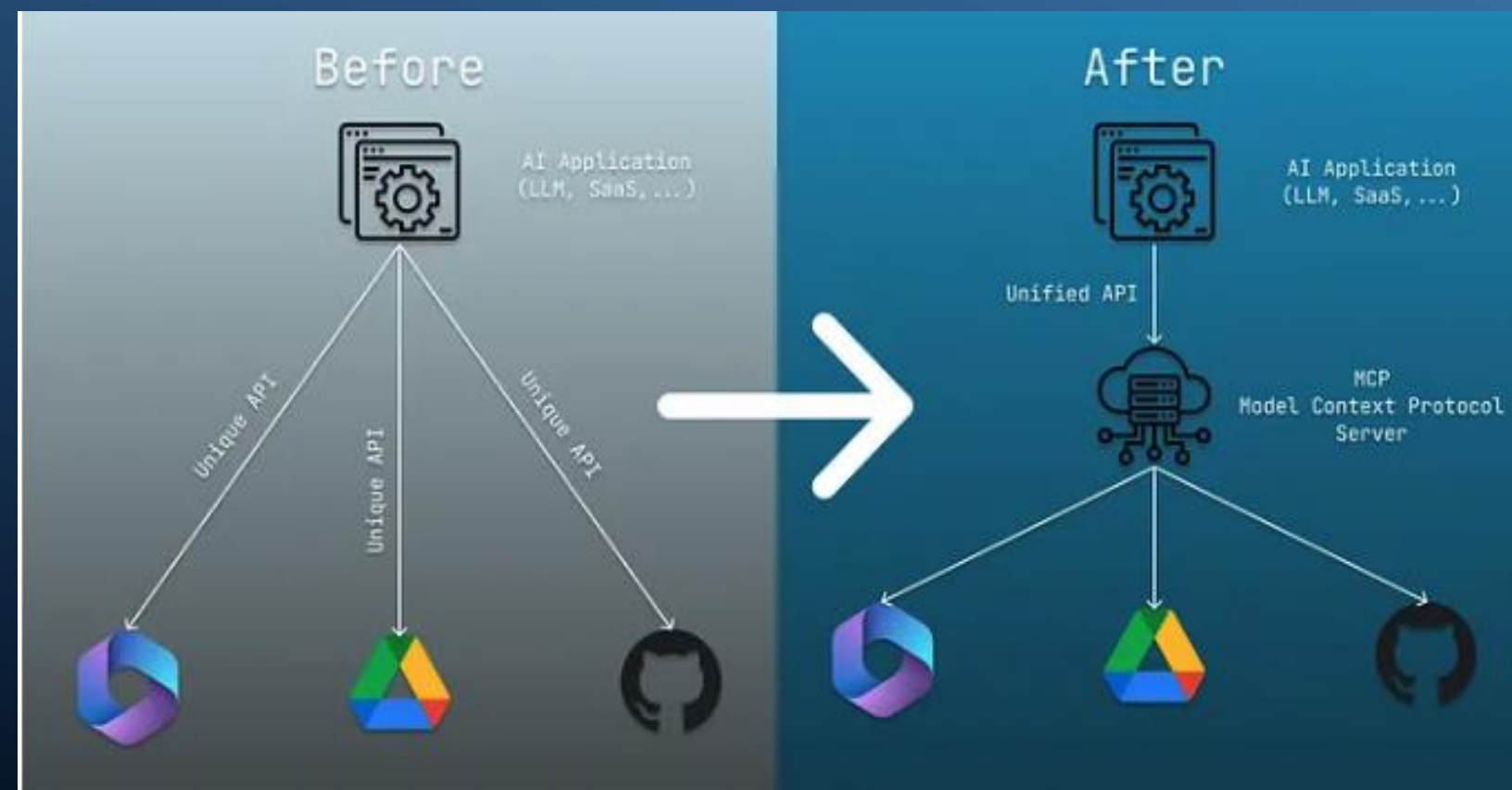
LLM Training vs RAG

Category	LLM Training	RAG
Knowledge Source	Internalized into weights	Retrieved dynamically
Update Speed	Slow	Fast
Training Cost	Very high	Low–moderate
Inference Overhead	Lower retrieval overhead	Additional retrieval steps
Context Window Usage	Minimal	Consumes context capacity
Best For	Stable large-scale knowledge	Dynamic/private knowledge
Hallucination Reduction	Limited	Better grounding
Infrastructure Needs	GPUs + distributed training	Vector DB + retrieval pipeline
Enterprise Data	Hard to keep current	Easy to update
Operational Complexity	Training-heavy	Retrieval-heavy

Training and RAG are complementary strategies.

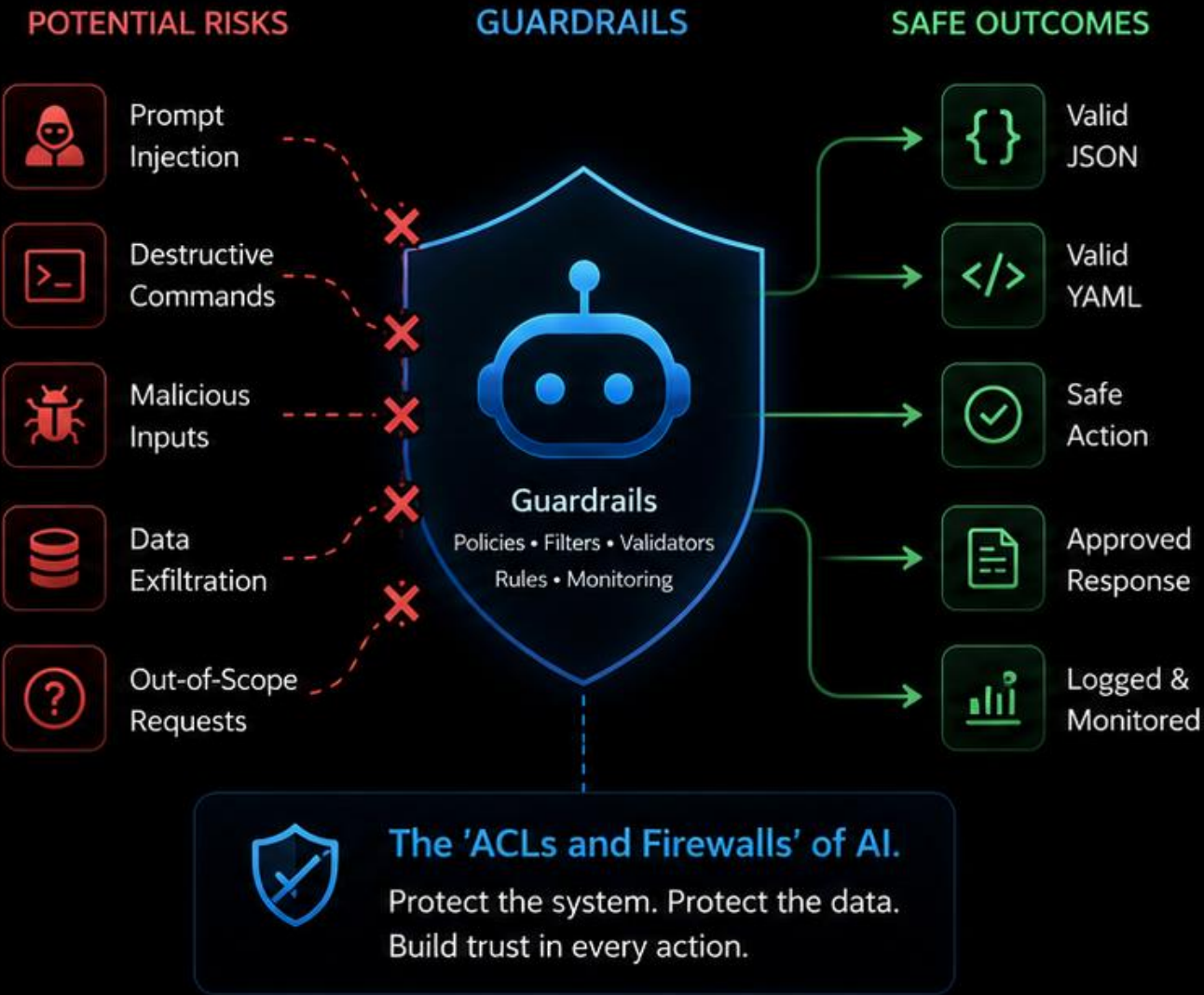
Model-Context Protocol - MCP

- An open standard to connect AI to data sources.
- The 'Control Plane Protocol' for AI
- Agents can finally interact with tools using a standard protocol.
- Replaces custom integrations with a unified API bus.
- Grafana, OpenSearch, Elastic, Zabbix, all of those have support for MCP.



Guardrails

	1. Define Boundaries Defining operational boundaries and output schemas.		Establish what the Agent can do and what it must produce.
	2. Input Filtering Stopping prompt injections and unsafe inputs.		Validate and sanitize all inputs before they reach the Agent.
	3. Execution Blocks Preventing destructive CLI commands and system actions.		Block or restrict high-risk operations (e.g., 'DROP TABLE').
	4. Schema Enforcement Ensuring the AI only outputs valid JSON/YAML.		Validate output structure to ensure correctness and consistency.
	5. Observability & Audit Logging, monitoring, and traceability of all actions.		Track behavior, detect anomalies, and enable accountability.



Guardrails don't limit AI — they enable safe, reliable, and trustworthy systems.



Local LLM Inference

Run powerful models locally via [Ollama](#), [vLLM](#), etc.



WHAT IS IT?

Running LLM inference on your own hardware instead of relying on cloud APIs.



WHEN TO USE



Data Sovereignty

Keep sensitive data and models within your control.



Keep IP Off the Cloud

Running inference locally (via Ollama or vLLM) keeps your intellectual property off the cloud.



Zero Network Latency

Eliminate network hops for faster, more reliable responses.



Air-Gapped Ready

Perfect for secure, isolated, or air-gapped environments.



THE TRADE-OFF



You must manage the **hardware infrastructure**, including **GPU**.

What this includes:

- ✓ GPU acquisition & capacity planning
- ✓ Drivers, CUDA, and dependencies
- ✓ Monitoring, scaling, and maintenance
- ✓ Power, cooling, and physical space



THE RIGHT GPU MAKES THE DIFFERENCE

GAMING GPU



Great for gaming and content creation.

Limited for large model inference.

From 100s of dollars

VS

DATACENTER GPU



Built for scale, reliability, and 24/7 workloads.

Higher throughput, stability, and VRAM.

PERFORMANCE & VALUE

VS

AI ACCELERATOR



Purpose-built for AI training and inference.

Maximum performance and efficiency.

To 10s of hundreds of dollars



Run where it makes sense. **Own your data.** **Control your stack.** Unlock the full potential of **AI on your terms.**

Graph Databases

Model reality as **connected data**. Discover **truth** in relationships.

- 

Stores Nodes (entities) and Edges (relationships).
Everything is connected.
- 

Relationships become first-class citizens.
Query and reason over connections naturally.
- 

Neo4j: Hard-wiring relationships to eliminate hallucination.
The database enforces what's actually connected.
- 

The Topology Map
Store relationships as Nodes and Edges.
Perfect for Dependency Mapping.



Hard-Wired Truth

If a Vector DB finds things that sound similar, a Graph DB finds things that are **actually connected**.

Vector DB (Similarity Search)



- ✗ Finds things that sound similar.
- ✗ No guaranteed relationship.
- ✗ Leads to hallucination.

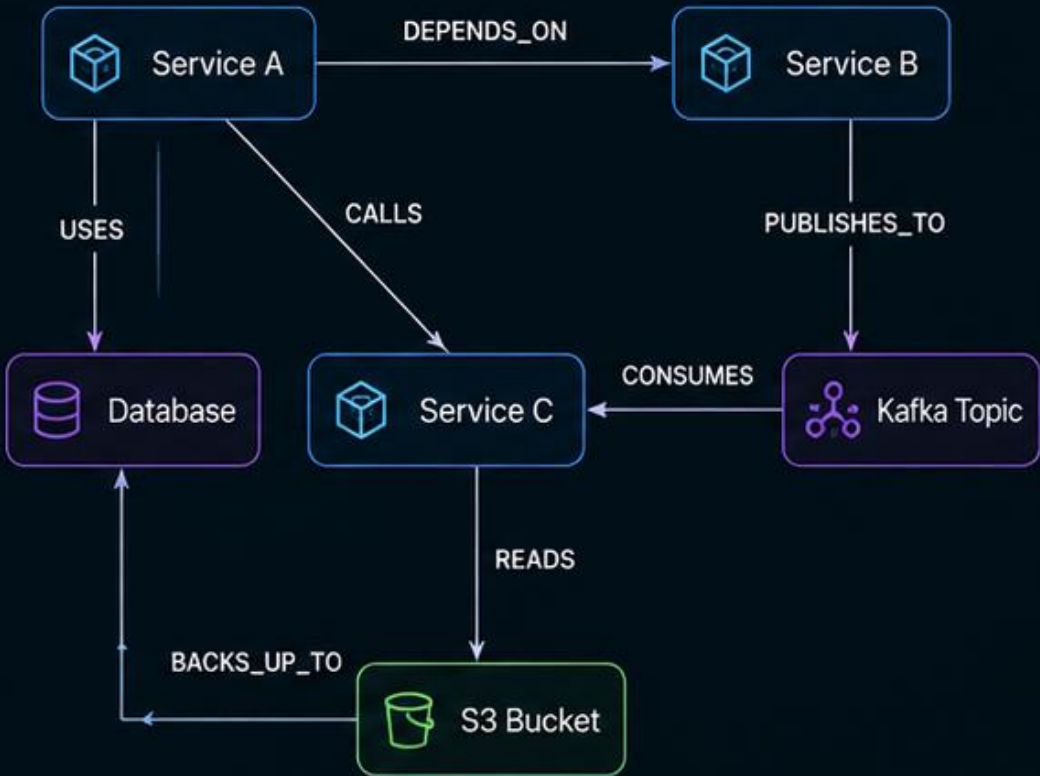
VS

Graph DB (Connected Truth)



- ✓ Finds things that are actually connected.
- ✓ Relationships are explicit and enforced.
- ✓ Eliminates hallucination.

Topology Map Example (Dependency Mapping)



From disconnected data to **connected intelligence**.



Better context



Accurate reasoning

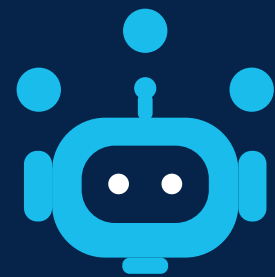


Confident decisions



AI Concepts vs Network

AI Concept	Networking Analogy
Tokens	Packets
Context Window	Buffer/queue memory
MCP	Standardized protocol/API
Embeddings	Routing coordinates
Agents	Autonomous controllers
RAG	External lookup/knowledge plane
Transformer Attention	Dynamic path selection
Hallucinations	Routing loops/bad telemetry
Vector Search	Similarity routing



OpsSteward.ai

AI-ASSISTED NETWORK OPERATIONS

OpsSteward.ai: AI-Native Network Operations Platform

- AI-driven operational intelligence platform to support SLA-based science drivers and R&E networks
 - Transforms operational data into intelligent network operations
- Designed for large-scale SDN, optical, and multi-domain infrastructures
 - *Integrates* topology, telemetry, logs, services, and operational context into a unified knowledge graph
 - *Combines* Data, Graph intelligence (Neo4j), Retrieval-Augmented Generation (RAG), and locally-hosted LLMs (vLLM)
 - To build AI-assisted operational workflows
- Built to support:
 - Faster troubleshooting, intelligent automation, root-cause analysis, SLA awareness, MOP writing, and autonomous operations
- Developed for environments where reliability, observability, and operational scale are critical

OpsSteward: How it works

- Reports network and policies inconsistencies, Correlates events, Notify operators, Create MOPs, Supports report-writing activities
- Operational answers cite the system of record, not model memory.
- Every decision is recorded for accountability and troubleshooting purposes.
- OpsSteward is not only chat. It is an evidence layer over network operations.
- **Operational Intelligence Through Context and Reasoning**



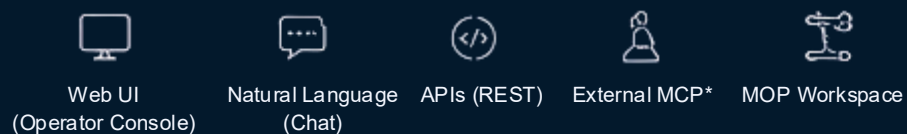
OpsSteward.ai v2

AI-Assisted Network Operations

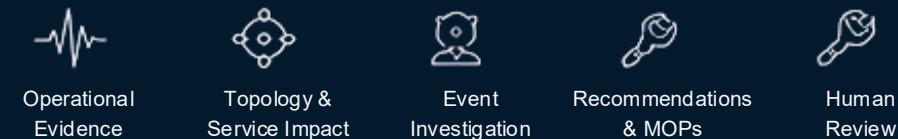
WHO USES IT



WAYS TO INTERACT



WHAT THEY GET

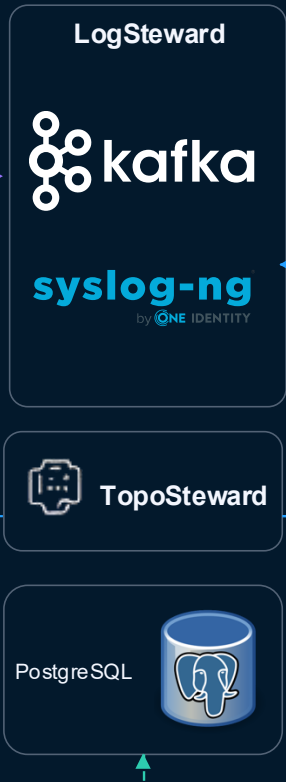


DATA & TELEMETRY SOURCES



Imports & Events

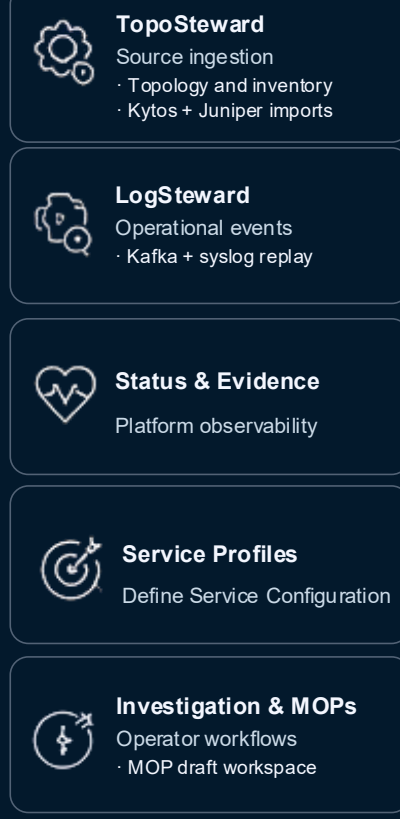
INGESTION & PERSISTENCE



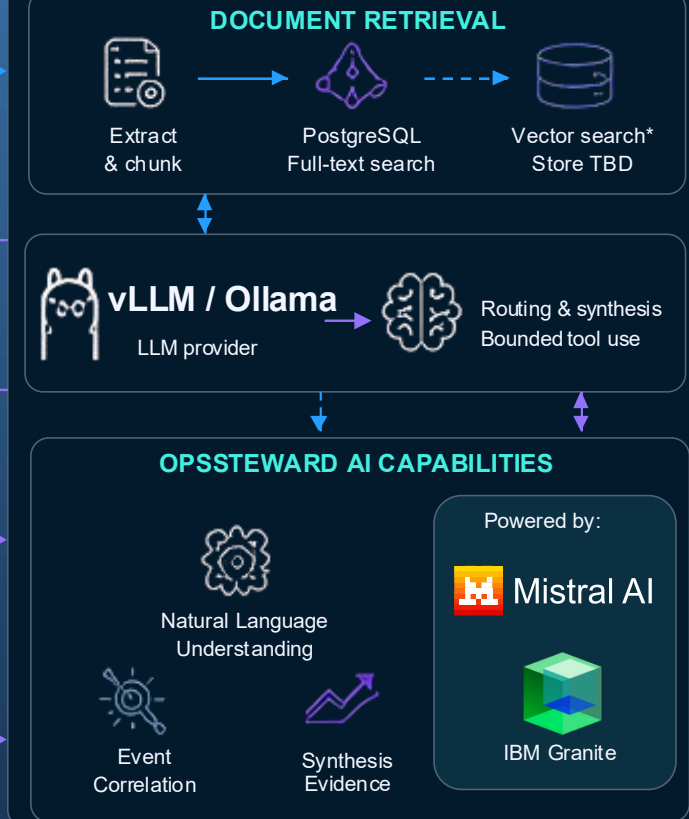
KNOWLEDGE GRAPH & OPERATIONAL INTELLIGENCE



OPSSTEWARD MODULES



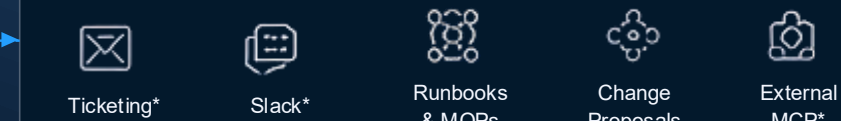
AI & INTELLIGENCE LAYER



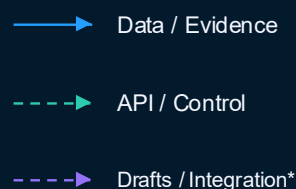
API & SERVICE LAYER



MOPs & EXTERNAL INTEGRATIONS



KEY



OBSERVABILITY, SECURITY & GOVERNANCE



AMLIGHT DEPLOYMENT



V2 DESIGN PRINCIPLES

- ✓ Evidence with source provenance
- ✓ Human control over decisions
- ✓ Topology and service context
- ✓ Explicit gaps and uncertainty
- ✓ Authenticated, scoped access
- ✓ Auditable tools and model use
- ✓ Adapters beyond AmLight

V2: read, analyze and draft · Human authority · No enabled production network-write path · * Planned / not yet established

FIU - Internal Data

Supported by NSF



Built with OpenAI Codex



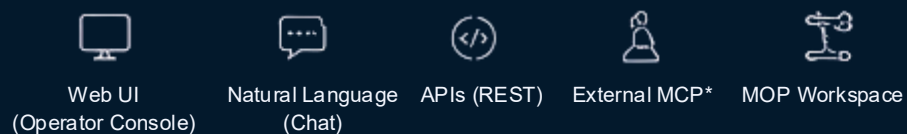
OpsSteward.ai v2

AI-Assisted Network Operations

WHO USES IT



WAYS TO INTERACT



WHAT THEY GET

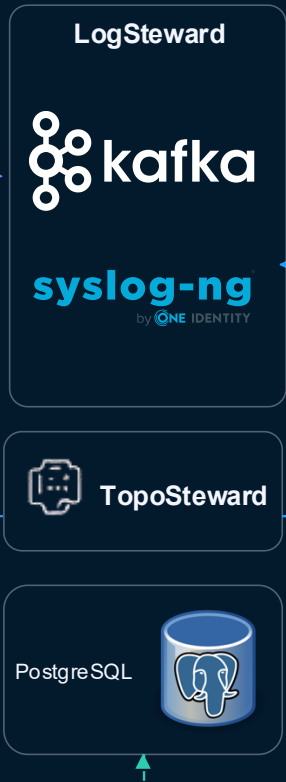


DATA & TELEMETRY SOURCES



Imports & Events

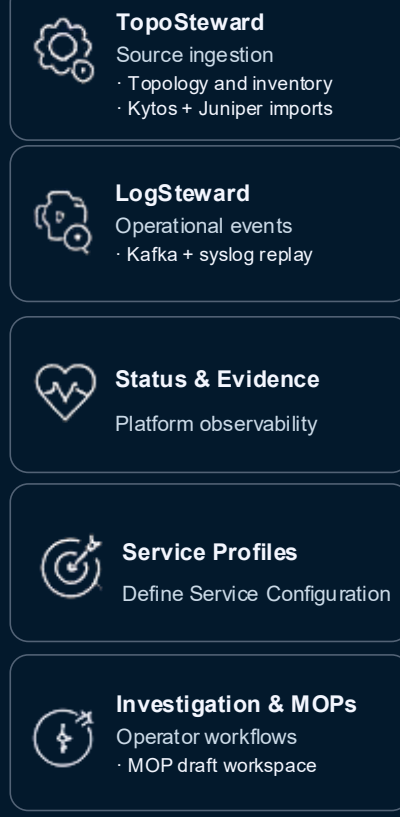
INGESTION & PERSISTENCE



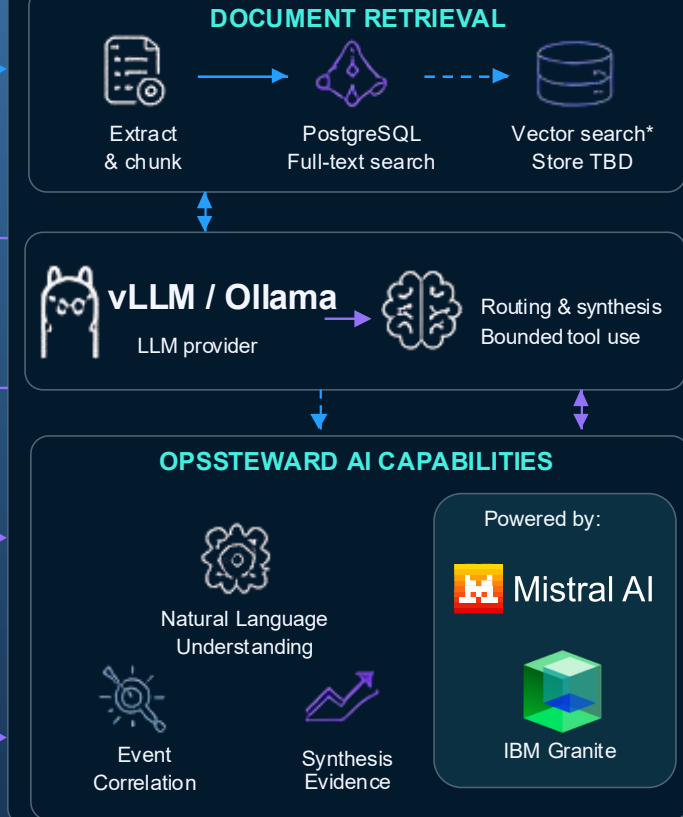
KNOWLEDGE GRAPH & OPERATIONAL INTELLIGENCE



OPSSTEWARD MODULES



AI & INTELLIGENCE LAYER



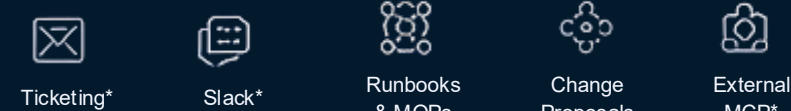
API & SERVICE LAYER



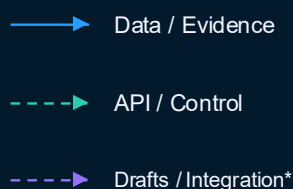
Authentication & RBAC
Explicit tool / scope checks
Interaction and routing audit
Versioned API contracts

MCP orchestration service
Graph + document evidence
Model-serving adapters
API access for operators

MOPs & EXTERNAL INTEGRATIONS



KEY



OBSERVABILITY, SECURITY & GOVERNANCE



AMLIGHT DEPLOYMENT



V2 DESIGN PRINCIPLES

- ✓ Evidence with source provenance
- ✓ Human control over decisions
- ✓ Topology and service context
- ✓ Explicit gaps and uncertainty
- ✓ Authenticated, scoped access
- ✓ Auditable tools and model use
- ✓ Adapters beyond AmLight

V2: read, analyze and draft · Human authority · No enabled production network-write path · * Planned / not yet established

FIU - Internal Data

Supported by NSF



Built with OpenAI Codex



The Knowledge Graph & Operational Intelligence

Neo4J/Graph Database builds all relationships

- Physical vs L1 vs L2 vs L3 vs Services vs Policies
- Historical State (up/down per item)

TopoSteward builds topology from Nautobot (network source of truth)

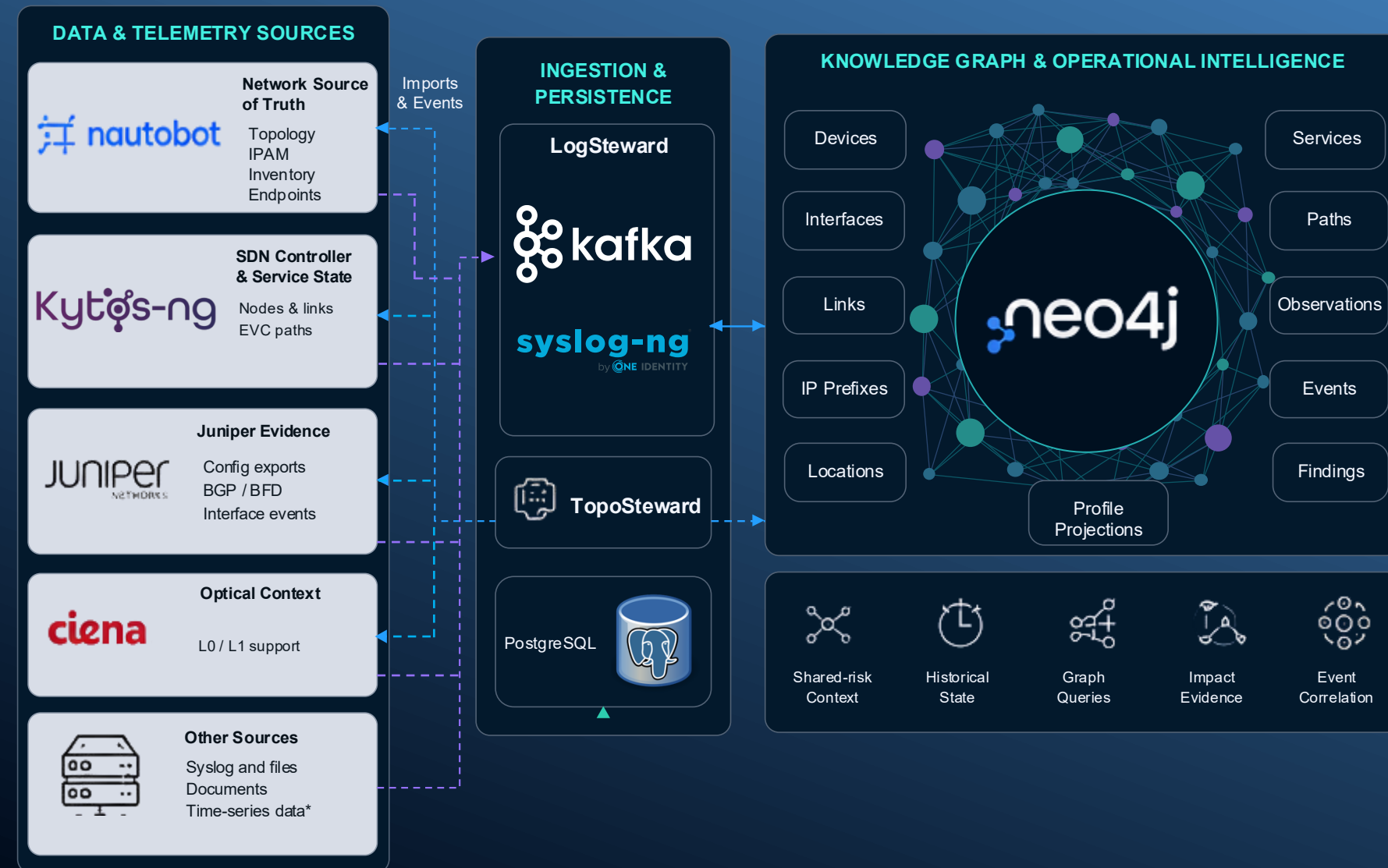
TopoSteward queries Kytos and Juniper devices for service information

LogSteward listens for events coming from Kafka

Syslog-ng server parses events and pushes to Kafka

With topology, service, and network state, OpsSteward can:

- Identify shared infrastructure and track risk
- Create SLA reports
- Correlate events



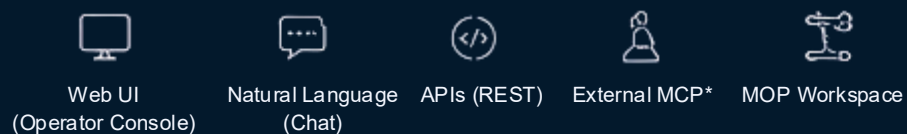
OpsSteward.ai v2

AI-Assisted Network Operations

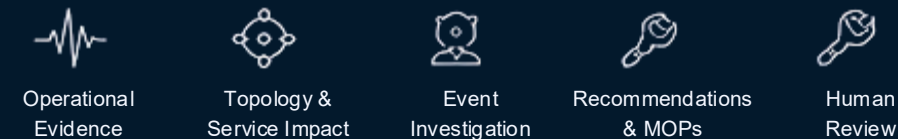
WHO USES IT



WAYS TO INTERACT



WHAT THEY GET

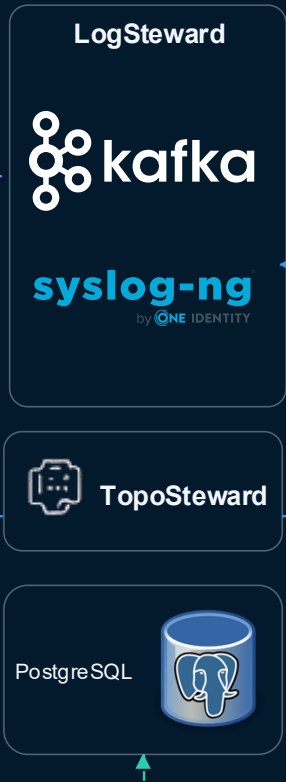


DATA & TELEMETRY SOURCES



Imports & Events

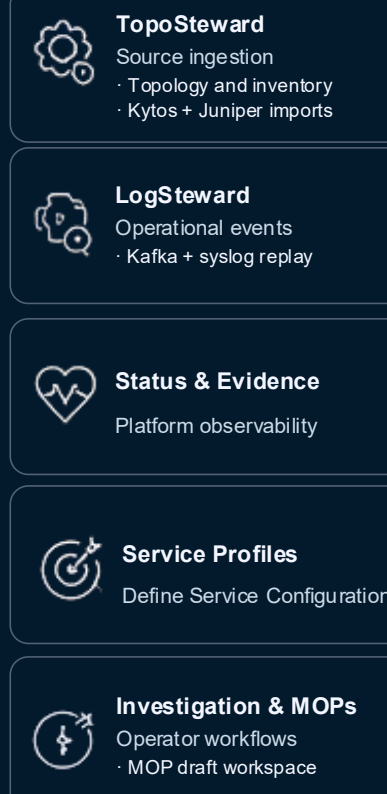
INGESTION & PERSISTENCE



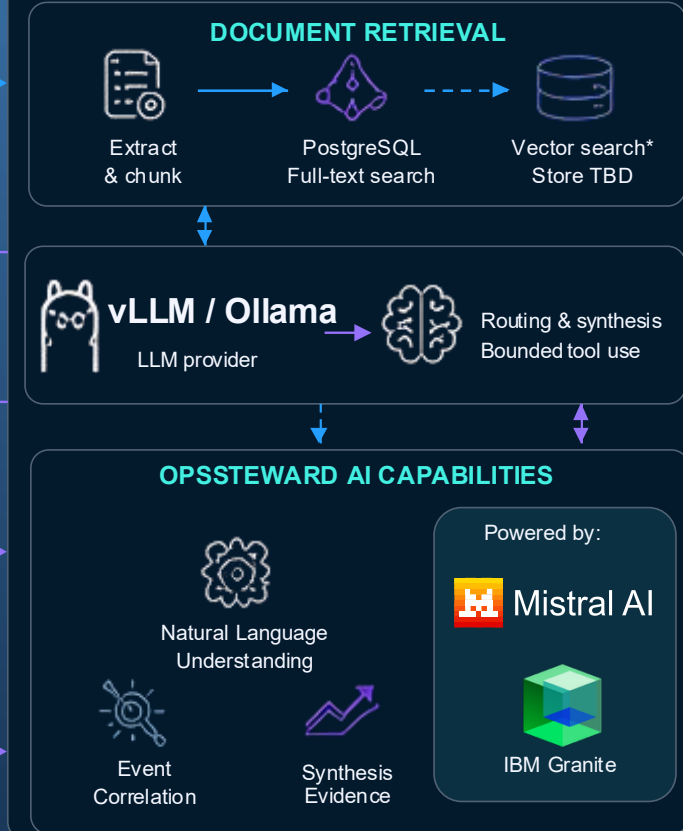
KNOWLEDGE GRAPH & OPERATIONAL INTELLIGENCE



OPSSTEWARD MODULES



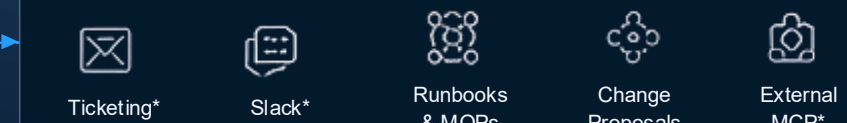
AI & INTELLIGENCE LAYER



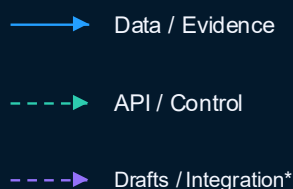
API & SERVICE LAYER



MOPs & EXTERNAL INTEGRATIONS



KEY



OBSERVABILITY, SECURITY & GOVERNANCE



AMLIGHT DEPLOYMENT



V2 DESIGN PRINCIPLES

- ✓ Evidence with source provenance
- ✓ Human control over decisions
- ✓ Topology and service context
- ✓ Explicit gaps and uncertainty
- ✓ Authenticated, scoped access
- ✓ Auditable tools and model use
- ✓ Adapters beyond AmLight

V2: read, analyze and draft · Human authority · No enabled production network-write path · * Planned / not yet established

FIU - Internal Data

Supported by NSF



Built with OpenAI Codex



Where and How AI supports OpsSteward

Knowledge Base via RAG

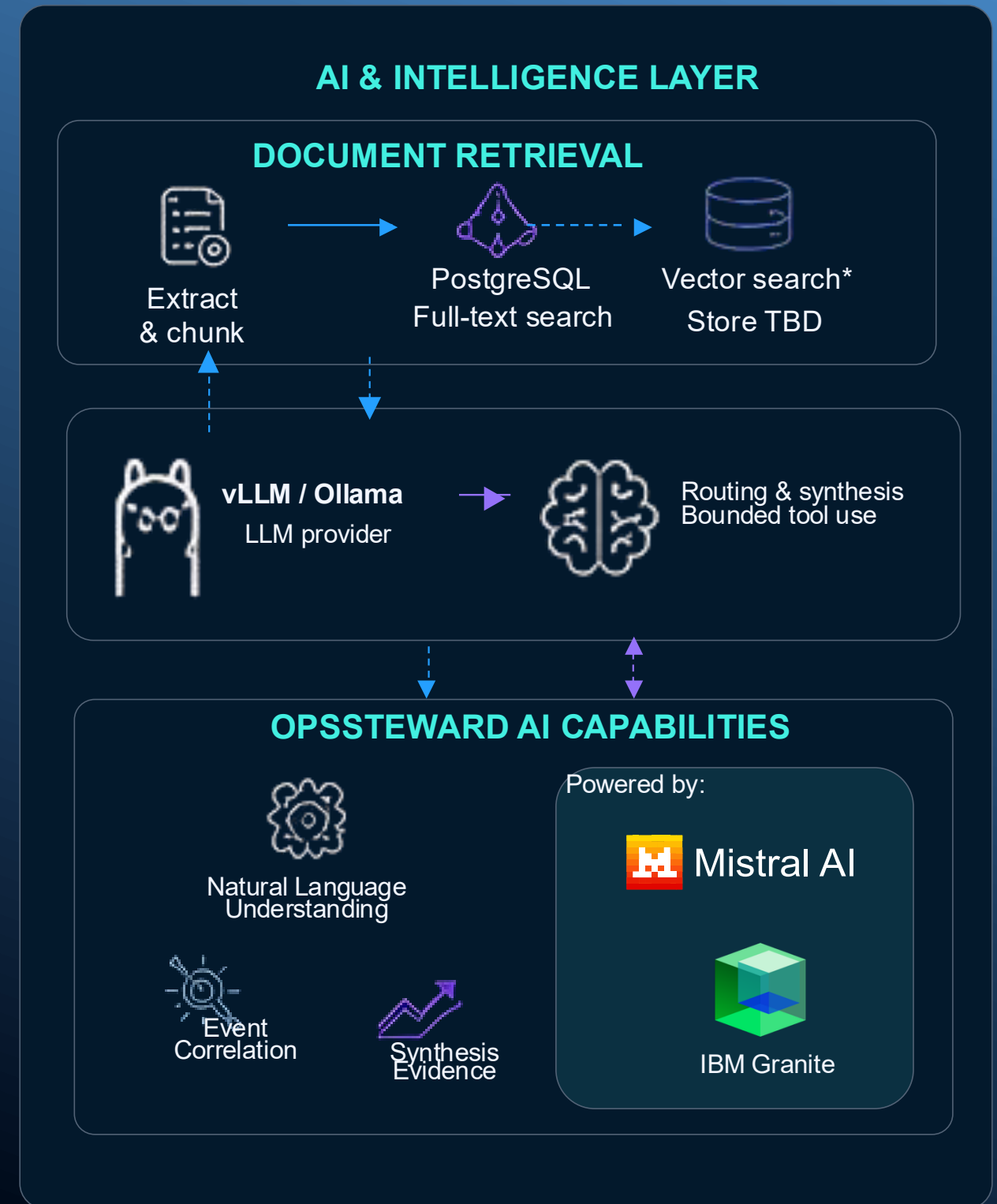
- Extract and chunk data into PostgreSQL / pgvector.
- Enables AmLight to import private repositories, reports, and internal documents.
- “Help me understand how the “*evc_manager*” tool works
- “*What is AmLight?*”, “*Who are AmLight Next Frontier project’s Pls?*”

LLM provider chosen was vLLM (vllm.ai)

- Parallelization
- Supports several models (depends on the GPU VRAM)

OpsSteward’s models:

- Mistral-7b
- IBM Granite
- Small models for faster processing



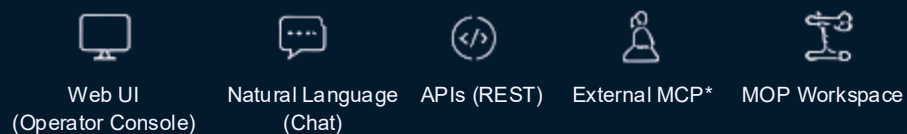
OpsSteward.ai v2

AI-Assisted Network Operations

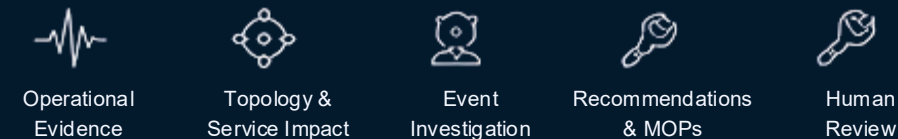
WHO USES IT



WAYS TO INTERACT



WHAT THEY GET

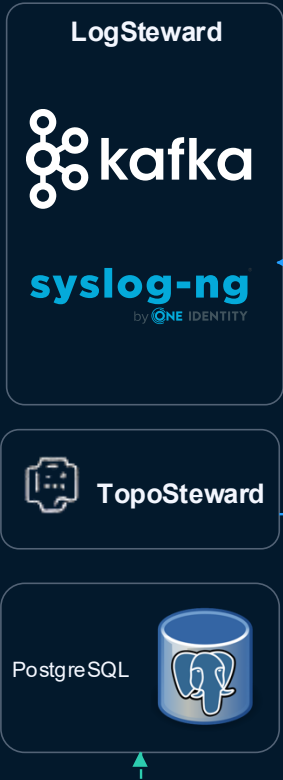


DATA & TELEMETRY SOURCES



Imports & Events

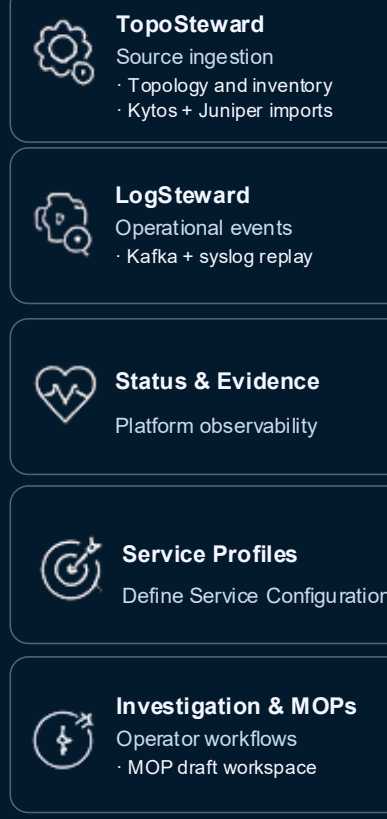
INGESTION & PERSISTENCE



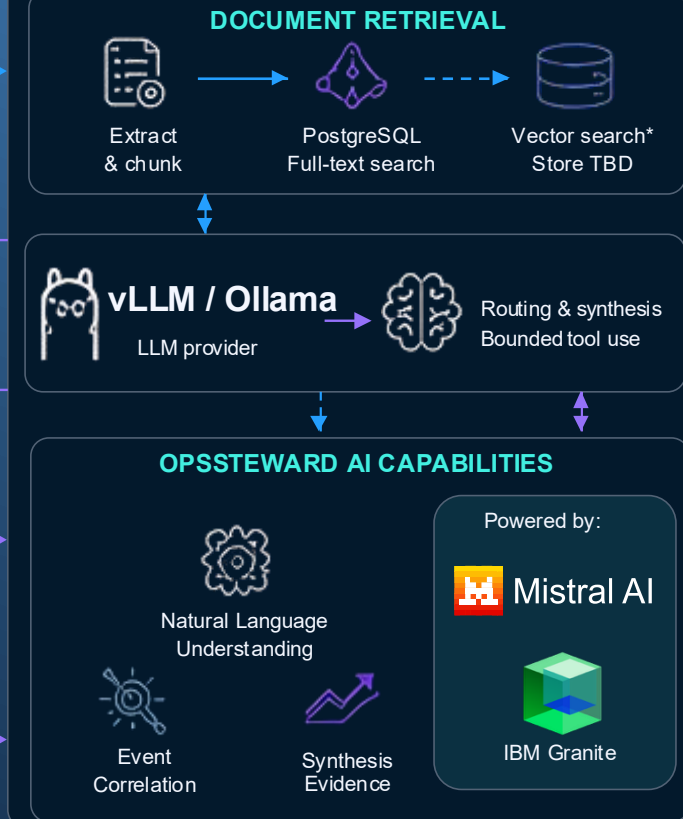
KNOWLEDGE GRAPH & OPERATIONAL INTELLIGENCE



OPSSTEWARD MODULES



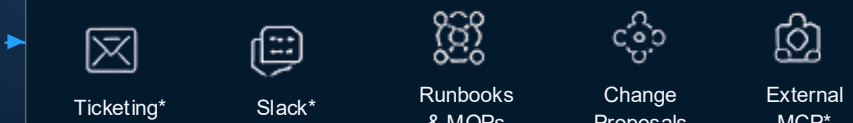
AI & INTELLIGENCE LAYER



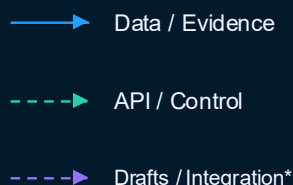
API & SERVICE LAYER



MOPs & EXTERNAL INTEGRATIONS



KEY



OBSERVABILITY, SECURITY & GOVERNANCE



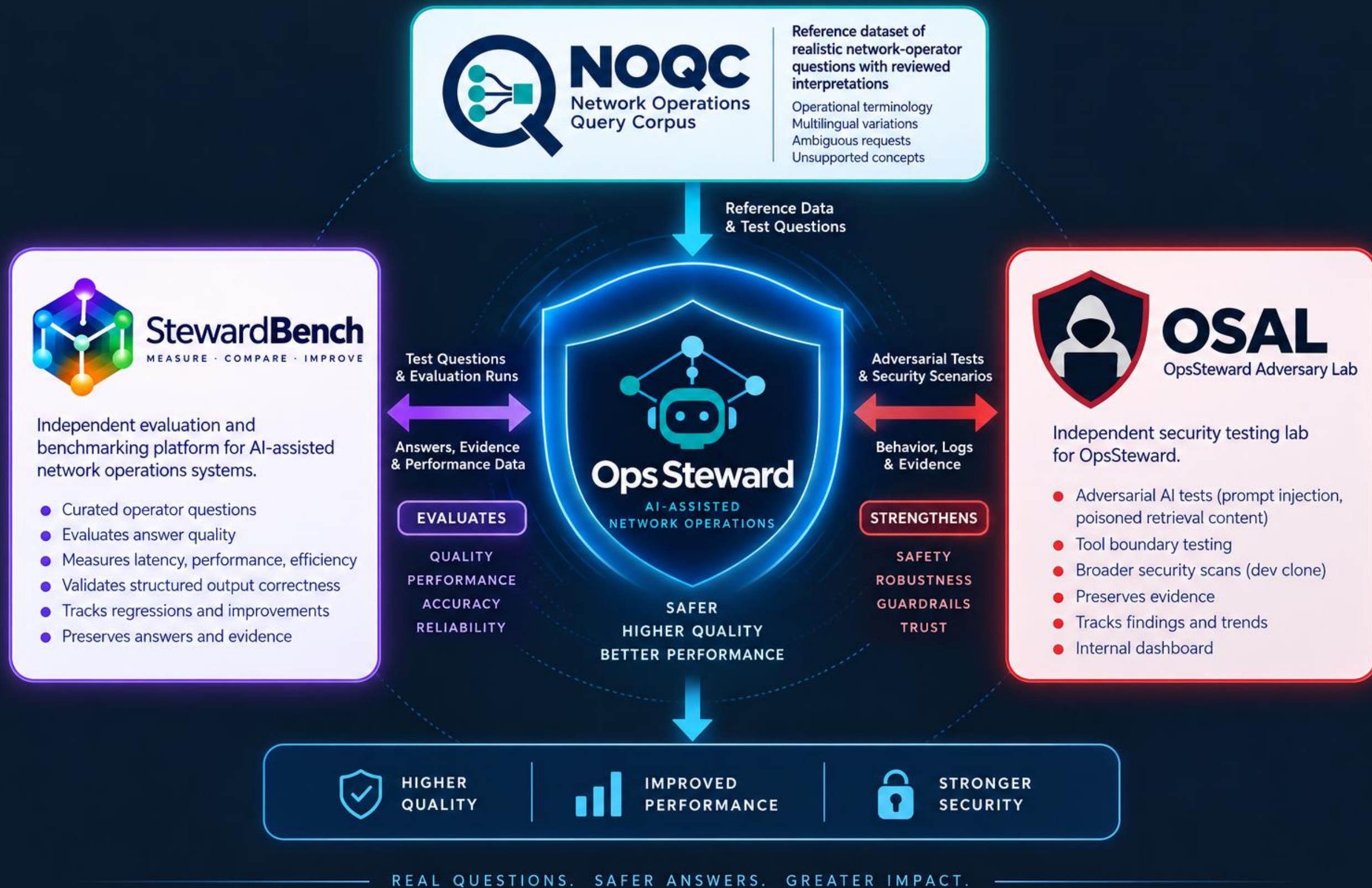
AMLIGHT DEPLOYMENT



V2 DESIGN PRINCIPLES

- ✓ Evidence with source provenance
- ✓ Human control over decisions
- ✓ Topology and service context
- ✓ Explicit gaps and uncertainty
- ✓ Authenticated, scoped access
- ✓ Auditable tools and model use
- ✓ Adapters beyond AmLight

How do we evaluate accuracy and safety?



Some Outputs

“What are the most unreliable links this month?”



OpsSteward
OPEN NETWORK OPERATIONS ASSISTANT

AI FOR NETWORK OPERATIONS
AMLIGHT REFERENCE DEPLOYMENT

Assistant

Docs

Status

Admin

Dark mode

What are the most unreliable links this month?

KNOWN QUESTIONS

What are the most unreliable links this month?

Production devices

Render topology

Interface timeline

Interface impact

PDU impact

MI1 feed impact

ANSWER SOURCE

☐ Auto

☐ Knowledge Base

☒ Graph DB

Send

Link ▾	L2Adjacency ▾	Flaps ▾	State Changes ▾	Total Downtime ▾	Ongoing Down ▾	Open Downtime ▾	Last Down ▾	Last Up ▾
BUE-BTW-SW02 Port 1 <-> SAO-SP4-SW03 Port 4	SAO-BUE-100G-05	118	555	5d 21h 30m 46s	no		2026-05-25 13:28:25+00:00	2026-05-25 13:28:27+00:00
MIA-MI1-SW18 Port 1 <-> JAX-LUM-SW01 Port 1	MIA-JAX-100G-02	27	123	14h 50m 40s	no		2026-05-20 04:06:12+00:00	2026-05-20 05:08:49+00:00
BUE-BTW-SW01 Port 32 <-> SAO-SP4-SW03 Port 23	SAO-BUE-100G-03	25	103	1d 0h 23m 42s	no		2026-05-18 18:39:05+00:00	2026-05-18 18:39:08+00:00
SCL-CIR-SW03 Port 1 <-> SAO-SP4-SW04 Port 1	SAO-SCL-100G-01	24	102	18h 19m 39s	no		2026-05-25 16:32:32+00:00	2026-05-25 16:34:31+00:00
SAO-SP4-SW03 Port 30 <-> BUE-BTW-SW01 Port 1	SAO-BUE-100G-01	24	102	4m 29s	no		2026-05-18 18:39:05+00:00	2026-05-18 18:39:08+00:00
BUE-BTW-SW02 Port 32 <-> SAO-SP4-SW03 Port 28	SAO-BUE-100G-04	23	103	1d 0h 23m 44s	no		2026-05-18 18:39:05+00:00	2026-05-18 18:39:08+00:00

👍

👎

↺

⚙️

</>

“show me AmLight topology”

RenderedInteractiveL0L1L2L3

Full

Blue

Contrast

Graphite

Role

Size

-

+

☒ Packet☒ Optical☒ Other

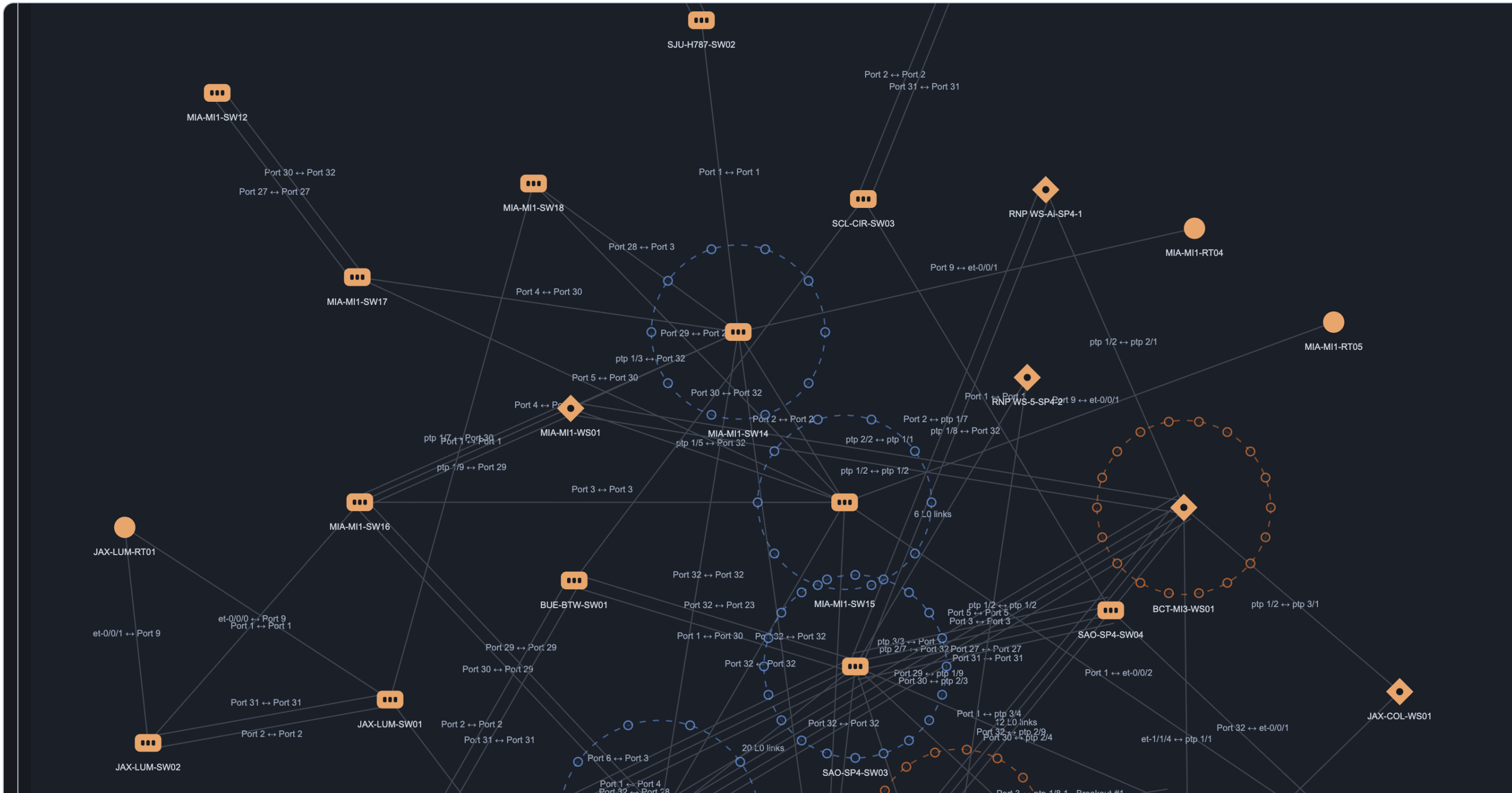
☒ Services☒ Devices☒ Interfaces☒ IP☒ Sessions

☒ Show AmLight network

Download SVG

Auto layout

Full screen



“Summarize what happened in the network last night...”



OpsSteward
OPEN NETWORK OPERATIONS ASSISTANT

AI FOR NETWORK OPERATIONS
AMLIGHT REFERENCE DEPLOYMENT

Assistant

Docs

Status

Login

Dark mode

Summarize what happened in the network last night and highlight Rubin LHN impact

KNOWN QUESTIONS

Summarize what happened in the network last night and highlight Rubin LHN impact

Production devices

Render topology

Interface timeline

Interface impact

PDU impact

MI1 feed impact

ANSWER SOURCE

☐ Auto

☐ Knowledge Base

☒ Graph DB

Send

Operational events in last night: 2 interface flaps, 4 BGP flaps, 4 BFD flaps, and 300 EVC state changes. Raw state-change records: 4 interface events, 8 BGP events, 8 BFD events, and 300 EVC events. A flap is a recovered down/up cycle; events are the individual state-change log records.

53 observed state events matched Rubin LHN using MOP Workbench desired_state.yaml. This is a v1.0 temporary service scope until Rubin LHN dependencies are modeled as graph service identity. Known routed-session downtime in the window is 38s.

👍

👎

↺

⚙️

deterministic route: inspect_operational_timeline · Processed in 1.45s · No LLM call



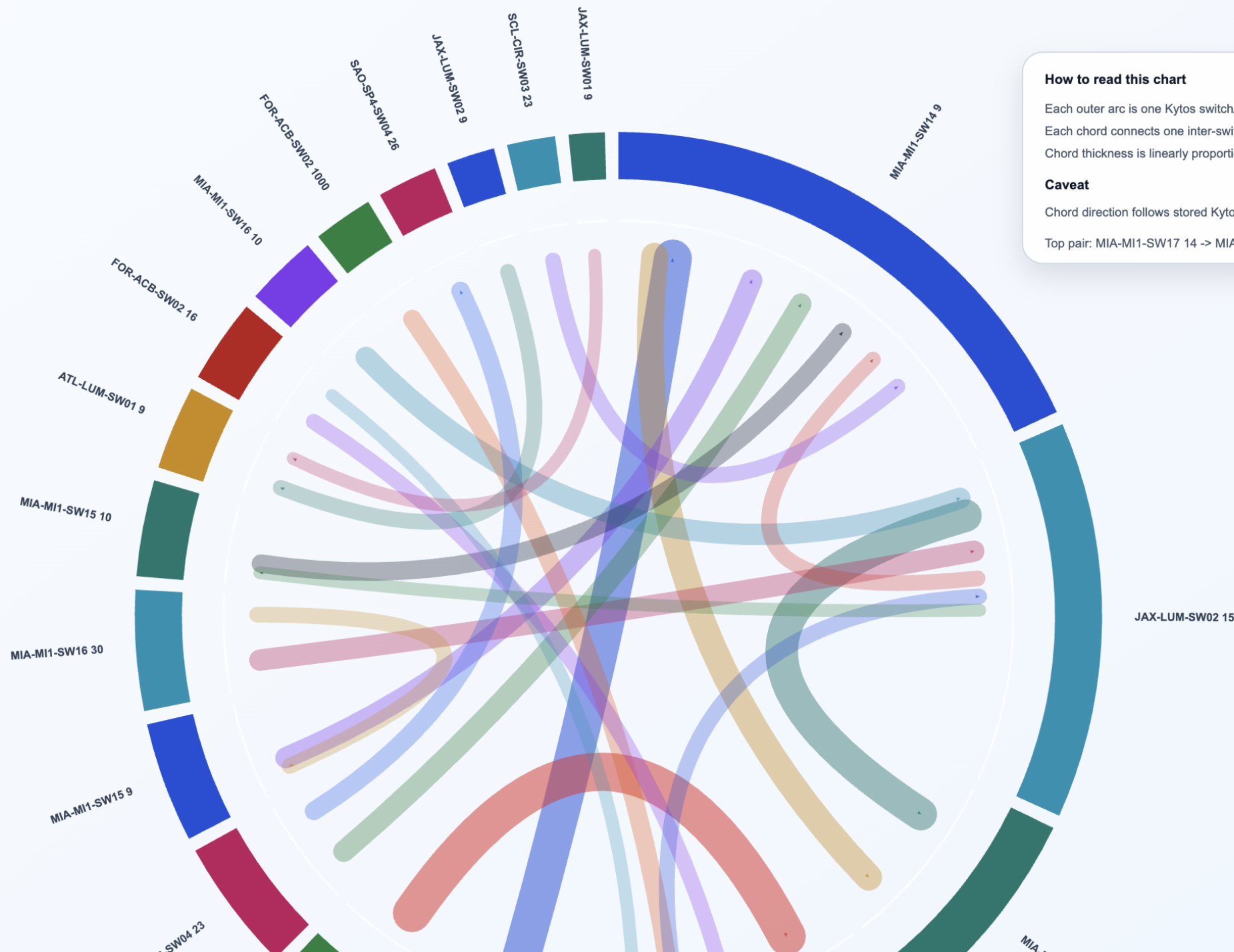
OpsSteward.ai
AI-Assisted Network Operations



“show me the top 20 used interfaces on Kytos”

Top 20 Kytos Interface Pairs by EVC Count

Top 20 active Kytos interface endpoint pairs ranked by the number of current EVCs.



How to read this chart

Each outer arc is one Kytos switch/interface endpoint.
Each chord connects one inter-switch stored A/Z endpoint pair.
Chord thickness is linearly proportional to active EVC count.

Caveat

Chord direction follows stored Kytos A/Z endpoint ordering, not measured traffic direction.
Top pair: MIA-MI1-SW17 14 -> MIA-MI1-SW14 9 (12 EVCs)

“List all spare transceivers in Chile per type”

List all spare transceivers in Chile per type

KNOWN QUESTIONS

List all spare transceivers in Chile per type

Production devices

Render topology

Interface timeline

Interface impact

PDU impact

MI1 feed impact

ANSWER SOURCE

☐ Auto

☐ Knowledge Base

☒ Graph DB

Send

Recent Questions

List all spare transceivers in Chile per type

9/28/2026, 4:07:19 PM

Are LAB-MI1-SW01 and LAB-MI1-SW02 using the same PDU?

9/28/2026, 4:07:12 PM

Show me the top 20 used interfaces on Kytos

9/28/2026, 4:06:04 PM

Type ▾	Serial ▾	Interface ▾	Device ▾	City ▾	Rack ▾	Country ▾
CW4	S0C290013	Port 10	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
LR4	DA190607216	Port 11	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
Unknown	FNSRMYU4LA4KR	Port 12	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
CW4	S19910051	Port 9	SCL-CIR-SW04	Santiago	DC11_044-005	Chile

deterministic route: inspect_transceiver_inventory · Processed in 124ms · No LLM call

👍

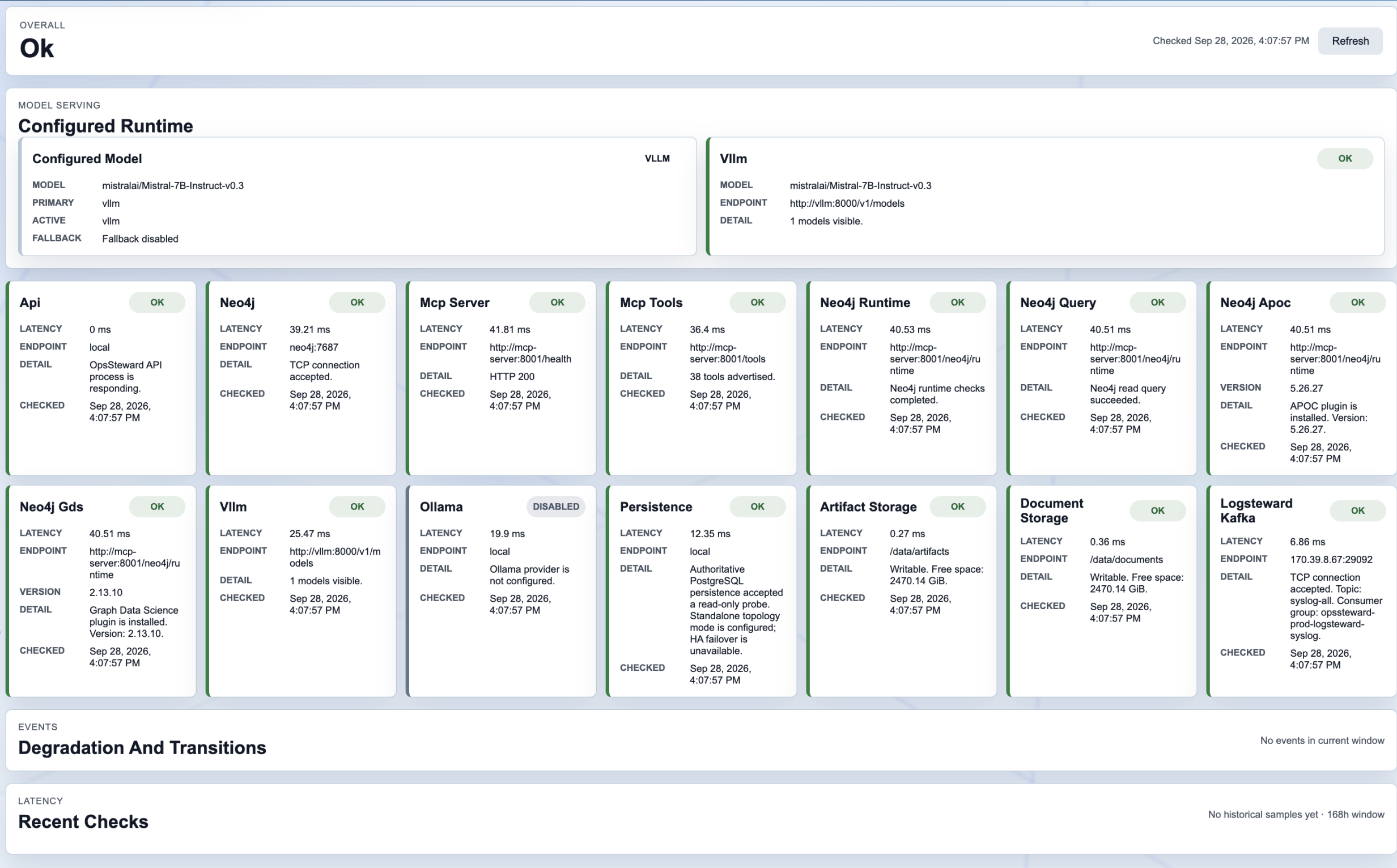
👎

↺

⚙️

</>

Status Page – Monitoring all components



MOP Workbench – AI-assisted maintenance windows

SIGNED IN AS
akadmin · admin

Learning · John & Wick

MOP Workbench

Accounts

Knowledge Base

Prompt Catalog

John triage is under Learning.
Stateful change planning is under MOP Workbench.

Log Out

MOP Workbench

Create persistent change-planning cases, capture missing data, draft MOP artifacts, and preserve evidence without executing network changes.

Refresh

NEW CASE

Show

moving-rubin-to-the-new-juniper-mx-router · active · needs_review

EVIDENCE COLLECTION

Moving Rubin to the new Juniper MX routers

Configure Rubin's EVCs to be physically disjoint and for the EVCs to terminate on the new routers JAX-LUM-RT01 and ATL-LUM-RT01.

Preplanning

Data Gathering

Desired State

Gap Analysis

Mop Drafting

Operator Review

Ready For Maintenance

Evidence Collection

Post Validation

Closed

AUTOMATED ACTIONS

Run read-only evidence checks through OpsSteward, validate the desired state, generate implementation plans, then draft human-approved MOPs.

Gather Data

Generate Gap Analysis

Validate Desired State

Generate Change Plan

Generate Juniper Config

Generate EVC Plan

Draft MOP

Generate Execution MOP

Evidence Collection

Post Validation

CASE METADATA

Title

Service or Scope

Owner

Stage

Status

Risk

Approval

Description

Operator Comment

Moving Rubin to the new Juniper MX routers

Rubin LHN

Jeronimo

Evidence Collection

Active

Critical

Needs Review

Configure Rubin's EVCs to be physically disjoint and for the EVCs to terminate on the new routers JAX-LUM-RT01 and ATL-LUM-RT01.

Optional audit note for this metadata update

Save Case Metadata

ARTIFACT

context.md

Editor

Preview

Save Artifact

38

OpsSteward.ai
AI-Assisted Network Operations

AmLight
The Next Frontier

NSF

Learning – John & Wick

SIGNED IN AS
akadmin · admin

Learning · John & Wick

MOP Workbench

Accounts

Knowledge Base

Prompt Catalog

John triage is under Learning.
Stateful change planning is under MOP Workbench.

Log Out

Search

Question, answer, or

Rating

All

Review

Unprocessed

Limit

100

Refresh

Export Regressions

Loaded 100 interactions.

Interactions

100

#113

OPEN

UNRATED

List all spare transceivers in Chile per type

Available transceivers in Chile: | Type | Serial | Interface | Device | City | Rack | Country | | --- | --- | --- | --- | --- | --- | auto · graph · 9/28/2026, 4:07:19 PM

#112

OPEN

UNRATED

Are LAB-MI1-SW01 and LAB-MI1-SW02 using the same PDU?

Power support comparison: SHARED. Shared target(s): MIA-MI1-PDU11, MIA-MI1-PDU12. auto · graph · 9/28/2026, 4:07:12 PM

#111

OPEN

UNRATED

Show me the top 20 used interfaces on Kytos

Rendered the top 20 active inter-switch Kytos interface endpoint pairs ranked by current EVC count. Direction follows stored Kytos A/Z endpo auto · graph · 9/28/2026, 4:06:04 PM

#110

OPEN

UNRATED

List production network devices

Device inventory for production network devices: auto · graph · 9/28/2026, 4:05:57 PM

#109

OPEN

UNRATED

How many transceivers in total at AmLight?

AmLight has 0 transceivers. auto · auto · 9/28/2026, 4:05:52 PM

#108

OPEN

UNRATED

How many Juniper routers are in use?

I could not map "Juniper are" to a graph entity that the local planner understands. Try one of: BFDSession, BFDSessionState, BGPSession, BGP auto · auto · 9/28/2026, 4:05:45 PM

#107

OPEN

UNRATED

#113 · auto · graph · 9/28/2026, 4:07:19 PM · text · summary

Not processed

✓

👍

👎

List all spare transceivers in Chile per type

Answer

Available transceivers in Chile:

Type	Serial	Interface	Device	City	Rack	Country
CW4	S0C290013	Port 10	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
LR4	DA190607216	Port 11	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
Unknown	FNSRMYU4LA4KR	Port 12	SCL-CIR-SW04	Santiago	DC11_044-005	Chile
CW4	S19910051	Port 9	SCL-CIR-SW04	Santiago	DC11_044-005	Chile

Replay

Replay Deeper

Planner Telemetry

{
 "trace": {
 "mode": "list",
 "cypher": "MATCH (d:Device)-[:HAS_INTERFACE|HAS_PORT]->(i:Interface)-[:HOSTS]->(t:Transceiver)\nWHERE (i.state = false OR toLower(toString(coalesce(i.state, \"\"))) IN [\"false\", \"disabled\", \"non-enabled\", \"down\", \"inactive\"]) AND (toLower(coalesce(d.country, \"\") CONTAINS toLower(\$scope_value) OR toLower(coalesce(d.city, \"\") CONTAINS toLower(\$scope_value) OR toLower(coalesce(d.facility, \"\") CONTAINS toLower(\$scope_value) OR toLower(coalesce(d.rack, \"\") CONTAINS toLower(\$scope_value)) AND (i.state = false OR toLower(toString(coalesce(i.state, \"\"))) IN [\"false\", \"disabled\", \"non-enabled\", \"down\", \"inactive\"]))\n WITH DISTINCT\n CASE WHEN toUpper(coalesce(i.part_id, t.part_id, \"\")) CONTAINS 'CWD4' THEN 'CWD4' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'CW4' THEN 'CW4' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'LR4' THEN 'LR4' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'FR4' THEN 'FR4' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'AOC' THEN 'AOC' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS '-DR' OR toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'GB-DR' THEN 'DR' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'SX' THEN 'SX' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'LX' THEN 'LX' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'LR' THEN 'LR' WHEN toUpper(coalesce(i.part_id, t.part_id, \"\") CONTAINS 'SR' THEN 'SR' ELSE 'Unknown' END AS type,\ncoalesce(t.serial, \"\") AS serial,\ncoalesce(d.name, \"\") AS device_name,\ncoalesce(d.country, \"\") AS country,\ncoalesce(d.rack, d.facility, \"\") AS rack\n RETURN type, serial, interface_name, device_name, city, country, facility, rack\n ORDER BY rack, city, device_name, interface_name, serial\nLIMIT \$limit",
 "params": {
 "limit": 500,
 "scope_value": "Chile"
 }
 }
}

Future

- Version 1.0 – Release-grade assistant (Q2/2026)
 - Builds graph database from Nautobot and Kytos SDN controller and Syslog events via Kafka
 - Supports IP/VRF, L2 Services, PDU, event correlation
 - Uses Ollama on a Nvidia DGX Spark
- Version 2.0 – Interoperable operations model (Q3/2026)
 - Adds support for vLLM over Ollama on a Nvidia DGX Spark
 - Adds support for Infrastructure Segments for Disjointness verification
 - Parallelism, High Availability, Public MCP Server, new visual identity, tests
- Version 3.0 - Governed closed loop (Q1/2027)
 - Agentic AI with Human **approval** in the loop for service orchestration
- Version 4.0 – Fully AI-based closed loop (Q3/2027)
 - Fully Agentic AI with human **notification** for service orchestration

Questions? Thank you!



The background of the slide features a map of the Americas and Africa. A network of glowing white lines connects several key cities: Los Angeles, New York, Miami, Panama, Fortaleza, Rio de Janeiro, Sao Paulo, Santiago, Luanda, and Capetown. The lines represent network paths across the Atlantic Ocean and through Central and South America.

OpsSteward @ AmLight: Putting AI to Work in Network Operations

Jeronimo “Jab” Bezerra jbezerra@fiu.edu

IT Associate Director Florida International University