



Poznańskie Centrum Superkomputerowo-Sieciowe
Poznan Supercomputing and Networking Center



Evolutionary Transition to Advanced Automation of Network and Service Management

A Blueprint Implementation in the PIONIER Network

TNC26, Helsinki, 8-12 June 2026



Tomasz Szewczyk (PCSS), Roman Łapacz (PCSS)

Agenda

1

PIONIER

2

The goal

3

The plan

4

Implementation

5

Production in PIONIER

6

Key takeaways

7

Future

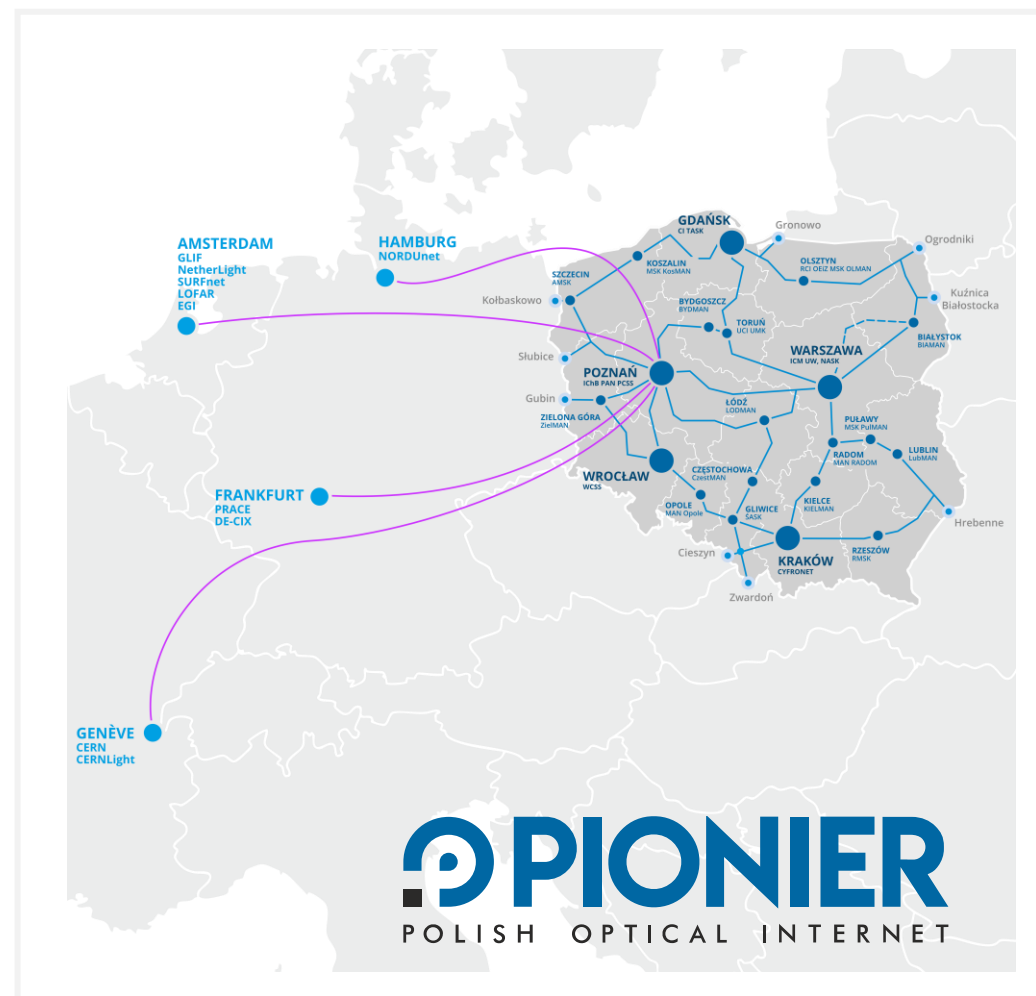
PIONIER – Polish National Research and Education Network

Consortium of 22 academic MANs

8,504 km of fibre-optic infrastructure connecting major scientific centres across Poland and beyond

International connectivity via GÉANT, linking Polish science to European NRENs and the global Internet

A platform for many national and international projects

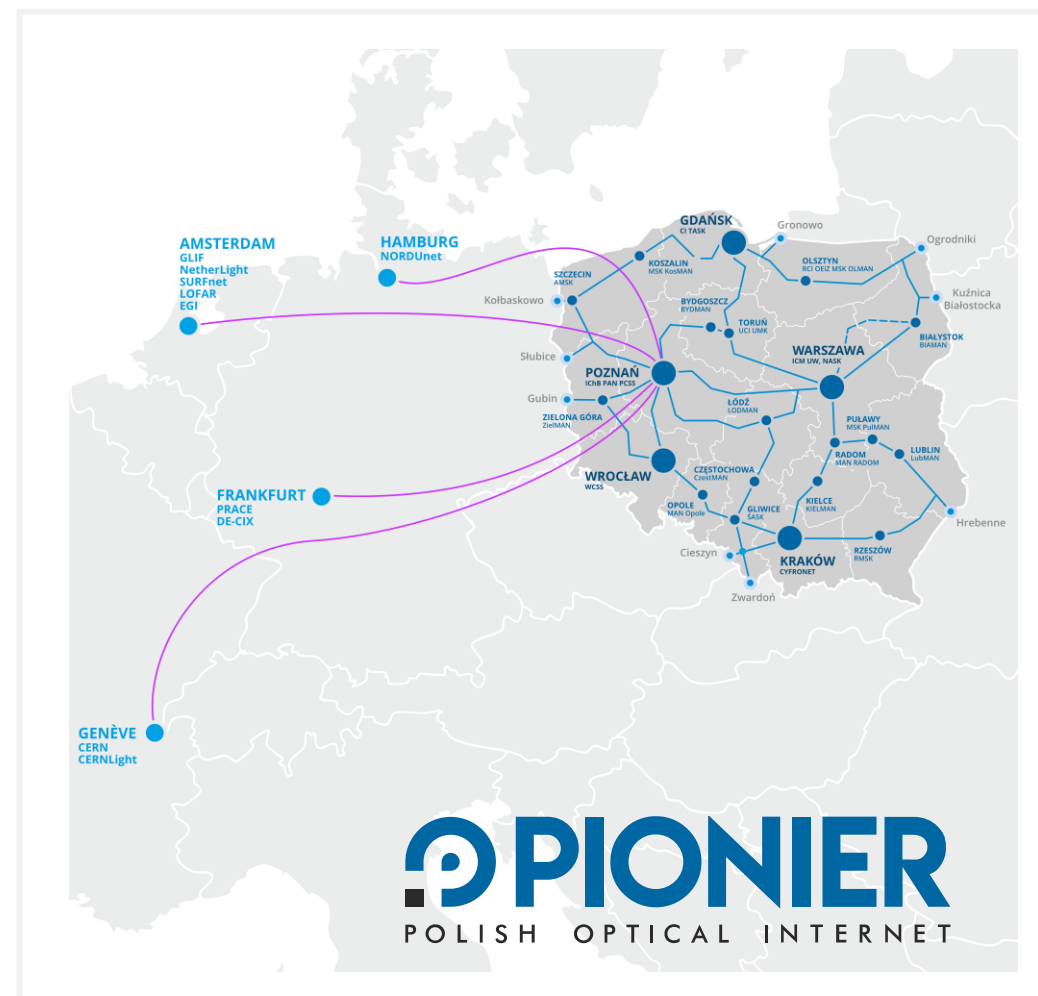


PIONIER – Polish National Research and Education Network

Optical transmission system: CDC nodes, ROADM architecture, alien wavelengths and spectrum as a service

Wide set of interfaces and transponders:
1200G/800G/400G/200G/100G

Packet network for IP and MPLS based services:
Internet access, virtual L2 and L3 networks



The goal



Implement a system for automated network service lifecycle management in PIONIER, gradually replacing manual operations while integrating with existing tools — ensuring zero negative impact on live services throughout the transition.

Key objectives

Integrate proven components

build upon existing, battle-tested management elements rather than replacing everything from scratch

Reduce operational time

significantly shorten the time required to provision, modify, and decommission network services

Eliminate human error

replace repetitive manual tasks with reliable, consistent automated workflows

Ensure extensibility

design for future growth, enabling new functionalities and adapting to evolving service market demands

Maintain full control

preserve operator oversight and trust despite increasing automation and autonomy of system components

Embrace open solutions

built on open standards and open-source technologies, supporting a heterogeneous, multi-vendor environment

Enable self-services

empower users and teams to independently manage selected operations without requiring manual operator intervention

The plan

1



The 1st phase

Data Model & Source of Truth

2



The 2nd phase

Workflows for network services and operations

3



The 3th phase

Scale up, extensions, autonomy

Data model & Source of Truth

The 1st phase of system development — consistent management of network information based on a flexible data model and a Source of Truth component

We need a way to quickly and easily define all possible resources and services (even those we don't know about yet) and the relationships between them

We want to avoid data duplication and synchronization issues

Source of Truth must integrate seamlessly with other components of the network management system

Maat – Source of Truth in PIONIER



TMF639 ResourceInventory
TMF638 ServiceInventory

Maat

Resources

Services

Maat service url:

https://maat.pionier.net.pl/resourceInventoryManagement/v4.0.0/resource

Create new Category

vendor

Juniper

QBC Photonics

model

PTX10001-30MR

MX304

ACX7024

ACX7100-48L

VRR

customer

site

device router

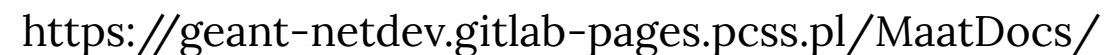
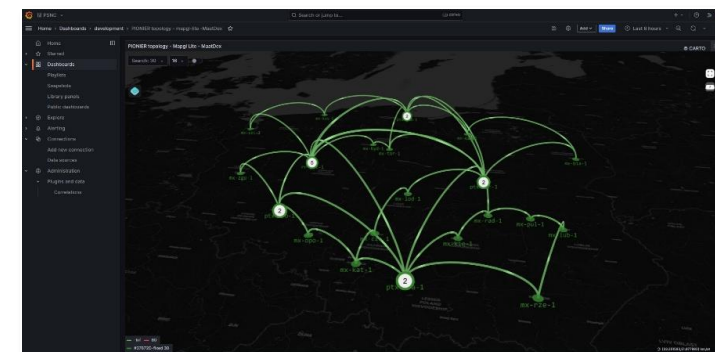
interface

Table

JSON

CREATE NEW ENTRY

Actions	Name	Description	Type	Category	StartOperatingDate	LastUpdateDate	ID
	Baldy_LWM	Baldy_LWM	PhysicalResource	site	2025-02-07T18:07:33Z	2025-02-07T18:10:00Z	b8dbd85-d99-4f6b-e62b-8d23cc0ca01
	Bialystok_PB	Bialystok_PB	PhysicalResource	site	2025-02-07T18:07:33Z	2025-02-07T18:09:07Z	5e610468-830f-4a7b-a96b-25c0418ba
	Bylgiszcz_PBS	Bylgiszcz_PBS	PhysicalResource	site	2025-02-07T18:07:33Z	2025-02-07T18:07:43Z	2ba0393b-3795-4a34-91af-e6eb55f51k
	Czestochowa_PCz	Czestochowa_PCz	PhysicalResource	site	2025-02-07T18:07:34Z	2025-02-07T18:09:11Z	bab73b6-15a0-d6e1-af53-6a0354a50c
	Gdansk_PG	Gdansk_PG	PhysicalResource	site	2025-02-07T18:07:34Z	2025-02-07T18:10:14Z	694b893a-95cf-42af-a673-aade9af64c
	Gniezno_PWSZ	Gniezno_PWSZ	PhysicalResource	site	2025-02-07T18:07:34Z	2025-02-07T18:08:15Z	7027899-67b1-48da-8057-3a0b819a
	Golezdzkie	Golezdzkie	PhysicalResource	site	2025-02-07T18:07:34Z	2025-02-07T18:08:15Z	03af1a3b-a5f0-4485-9d73-4e0ea8840
	Gorki	Gorki	PhysicalResource	site	2025-02-07T18:07:34Z	2025-02-07T18:08:06Z	80dc366-a77c-48ca-af65-1ea094849



Workflows

The 2nd phase - designing and implementing processes for managing the lifecycle of network services and automating network operations

- Workflows implemented in Python and managed by the orchestrator component
- Patterns and good practices of workflow designing from TMF 
- Close collaboration between network engineers, NOC staff and software developers
- Tests and continuous improvements
- Apache Airflow as a workflow orchestrator 

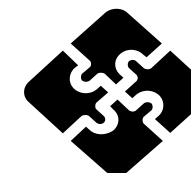
GP4L

<https://geant-netdev.gitlab-pages.pcss.pl/gp4ldocs/guides/playground/provision/idea/>

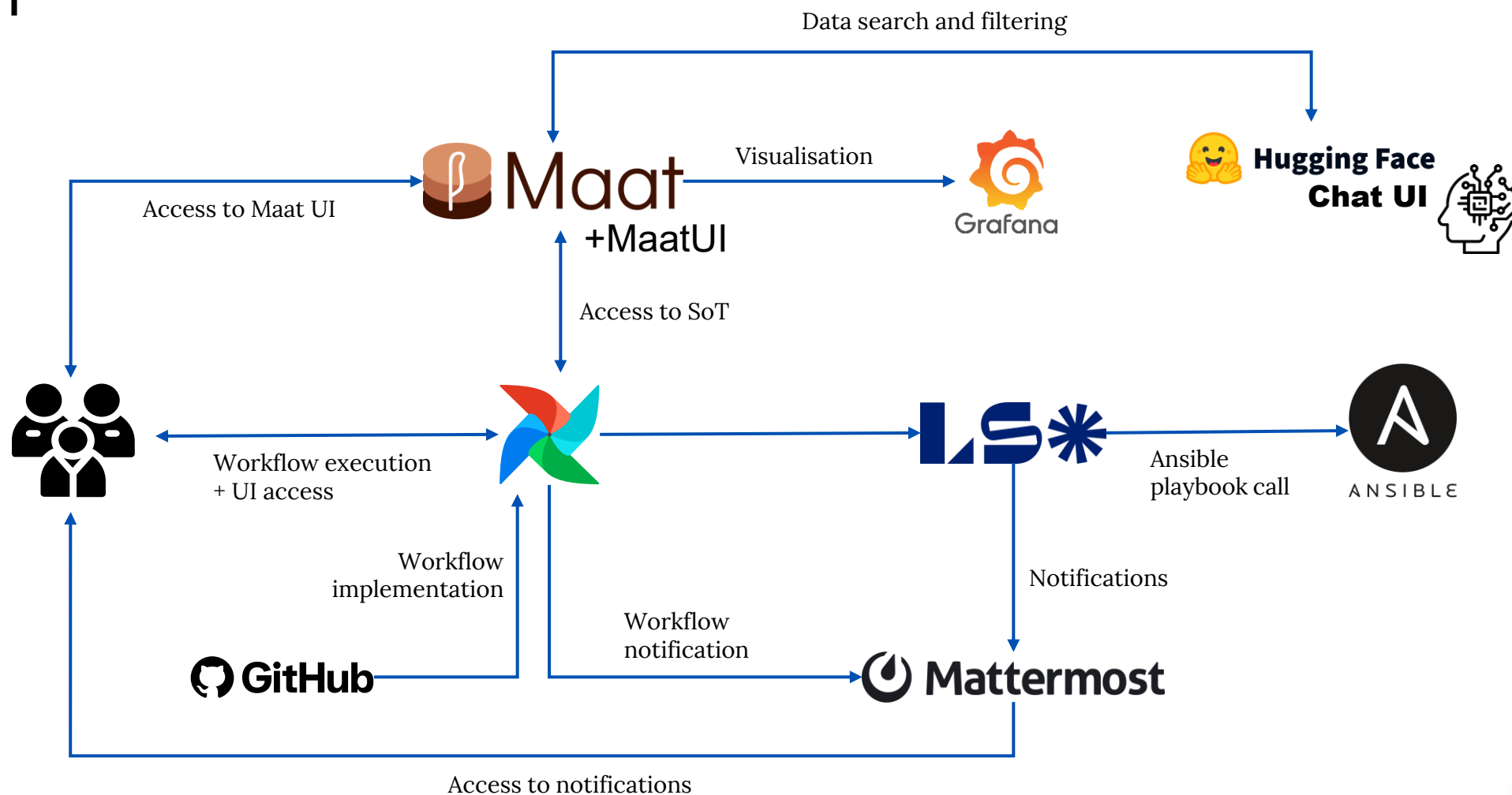
Scale and autonomy

The 3th phase – integration with other systems and implementation of new use cases

- New network services/workflows and data model extensions
- The use of AI for data analysis and controlled service management support
- Shift-left approach for the PIONIER stakeholders



Implementation



Production

Automated service delivery

Incremental deployment: It is impossible to automate all existing types of services at once and keep desired flexibility

First rollout for most popular service: Layer 2 point-to-point (E-LINE, l2circuit)

Keep SoT up to date: Extending the data model to support other services deployed in the network, but without full automation support

Validation-driven rollout: Define data model, deploy workflow, evaluate implementation, improve functionality

Devices	~100
L2 p2p	400+
L2 LAN	~10
L3 VRF	~5
L3 MVPN	1

Production

Current state

Automated services and operations: E-LINE (l2circuit), EVPN (vlan-based), interface configuration, E-LINE deployment validations, new device onboarding

Modeled and existing in SoT: L3 VRF (unicast)

Data modeling (work in progress): signaling protocols (BGP, ISIS, OSPF), routing policies, fiber cables and optical transport systems, transceivers

Information exchange: Mattermost chat bot (errbot) for metropolitan network (main customers) administrators

Production

Data modeling

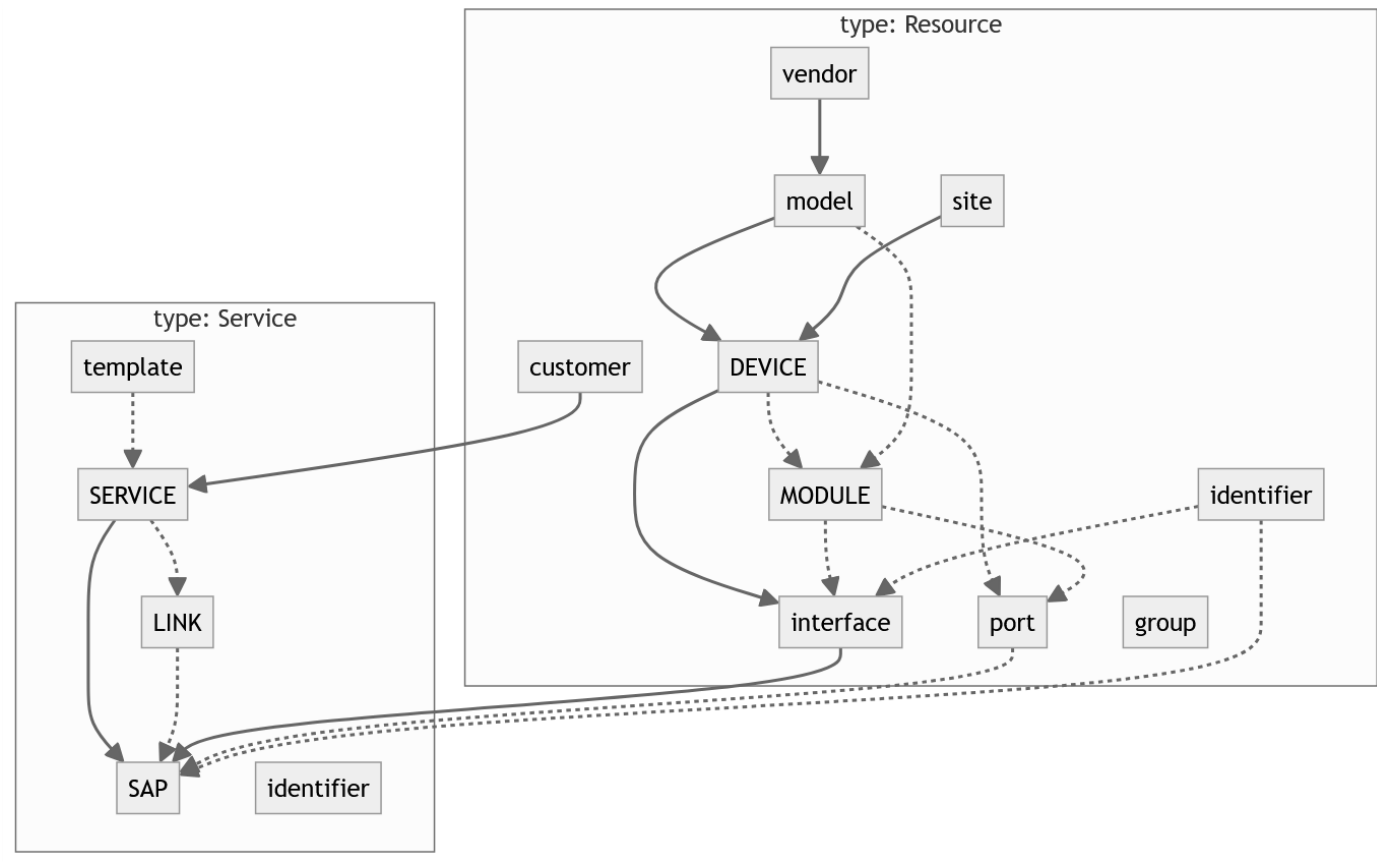
Relationship-driven referential integrity: dependencies between objects are implemented through resource/service relationships, which gives the structure of a dependency graph

Hierarchical characteristics: all model attributes are expressed as flat lists of name/value pairs with dot notation, which eliminates the need for the consuming application to know the schema - an iterator and a filter by key prefix are sufficient

Vendor-agnostic intent layer: the model describes what is to be configured (intent), not how - translation to vendor-specific configuration (JunOS, Nokia SR OS, etc.) occurs only at the Ansible/Jinja2 layer, which allows changing the device implementation without modifying the SoT or orchestration logic

```
category : "model"  
description : "Juniper MX304 Series"  
name : "MX304"  
  
@schemaLocation : "https://bitbucket.software.geant.org/projects/OSSBSS/repos/  
@type : "LogicalResource"  
  
▶ resourceRelationship : [...] 1 item  
▶ serviceRelationship : [] 0 items  
  
startOperatingDate : "2024-12-20T15:02:38Z"  
lastUpdateDate : "2026-04-24T12:38:25Z"  
href : "https://10.255.224.129:8082/resourceInventoryManagement/v4.0.0/resou  
  
▼ feature : [ 1 item  
    ▼ { 3 items  
        name : "interface_map"  
        id : 1  
        ▼ featureCharacteristic : [ 97 items  
            ▼ { 2 items  
                name : "port_modes.0.mode_type"  
                value : "native"  
            }  
            ▼ { 2 items  
                name : "port_modes.0.interface_name_template"
```

Data modeling



<https://gitlab.pcss.pl/geant-netdev/ssot-maat-datamodel-community>

Production

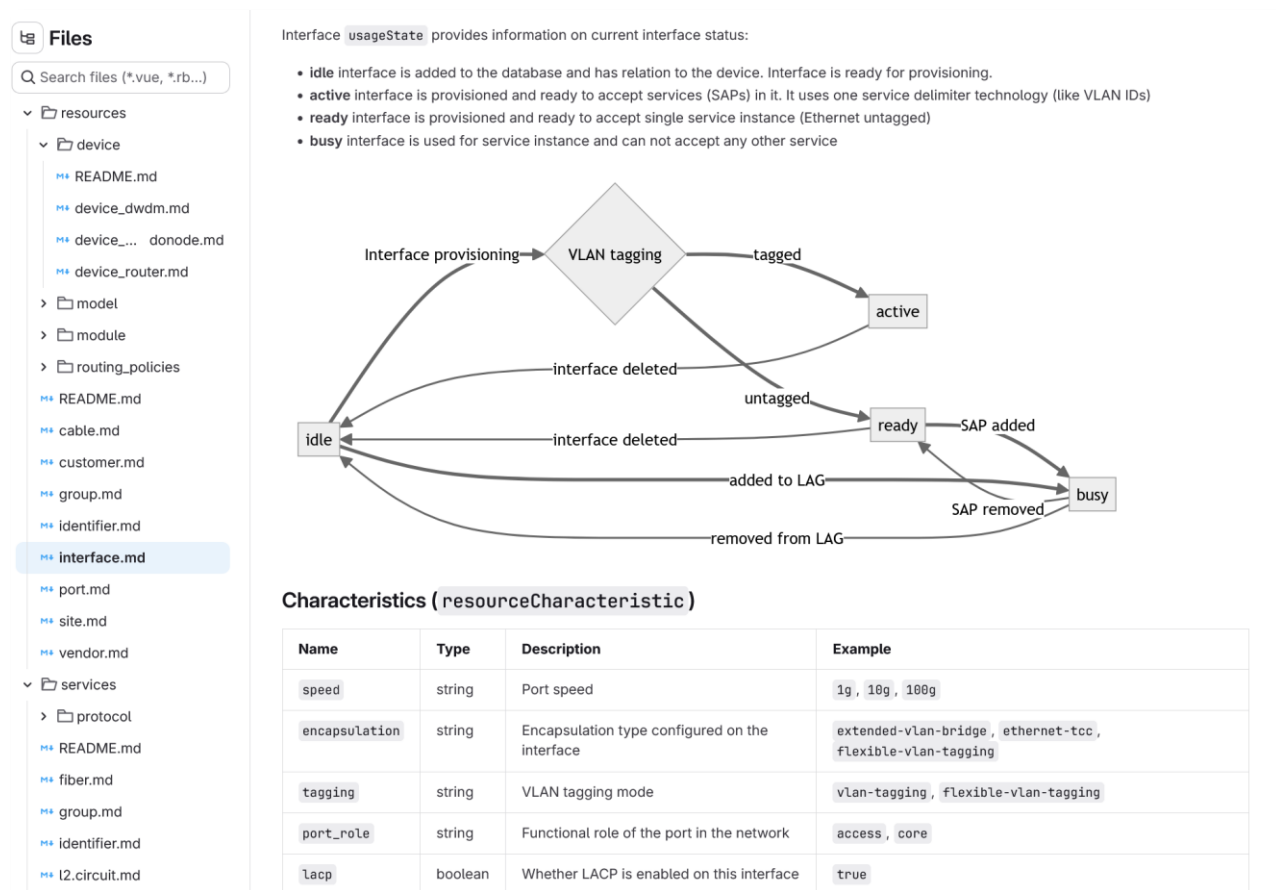
LLM-native readability: documentation can be directly injected into an LLM context

User-friendly format: you can view it in the editor, GitHub or Gitlab without knowing the schema format

Version control: allows you to track the evolution of the data model (e.g., model extension or update) with full context of the change

Modular decomposition: directory structure corresponding to the layers or categories of the model

Data modeling – documentation format



<https://gitlab.pcss.pl/geant-netdev/ssot-maat-datamodel-community>

Maat UI

Admin and developer tool

Manual changes

- model extensions
- validation
- data modification
- raw JSON editing

Automatic references

- maintaining data consistency
- **Keep SoT up to date**
- adding services and resources not (yet) supported by workflows

The screenshot displays the Maat UI interface. On the left, a sidebar shows a tree of resources under the 'Resources' tab. The selected resource is 'routing_policies.policy'. The main area shows the configuration for this resource. It includes fields for '@schemaLocation' (a URL to a JSON schema) and '@type' (LogicalResource). Below these, there is a section for 'resourceCharacteristic' which contains a table of characteristics. The table has columns for 'Name' and 'Value'. The table lists several characteristics, including 'term.1.then', 'term.1.from.1.community', 'term.1.from.0.prefix_list', 'term.1.name', 'term.0.then', 'term.0.from.0.community', and 'term.0.name'. Each row has a red trash icon in the 'Actions' column. At the bottom, there is a section for 'resourceRelationship' with a 'CREATE NEW RESOURCE RELATIONSHIP' button.

Actions	Name	Value
	term.2	Value
	term.1.then	accept
	term.1.from.1.community	c880d314-89dc-477b-9f23-196db3baf37
	term.1.from.0.prefix_list	c8dcc1fa-e43e-4bbe-b6ab-befa272a9b16
	term.1.name	term_2
	term.0.then	accept
	term.0.from.0.community	3d40ad4d-c1b6-4091-a71b-f705ee0e5075
	term.0.name	term_1

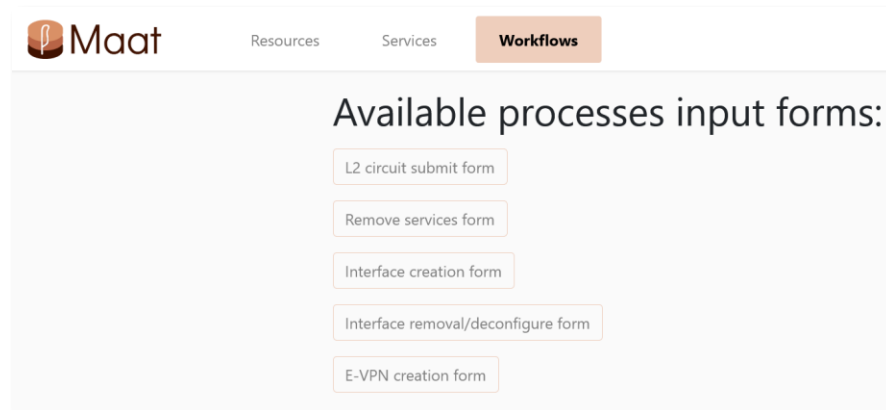
Maat UI

Collect information

- Retrieve existing inventory information
- Get user defined parameters

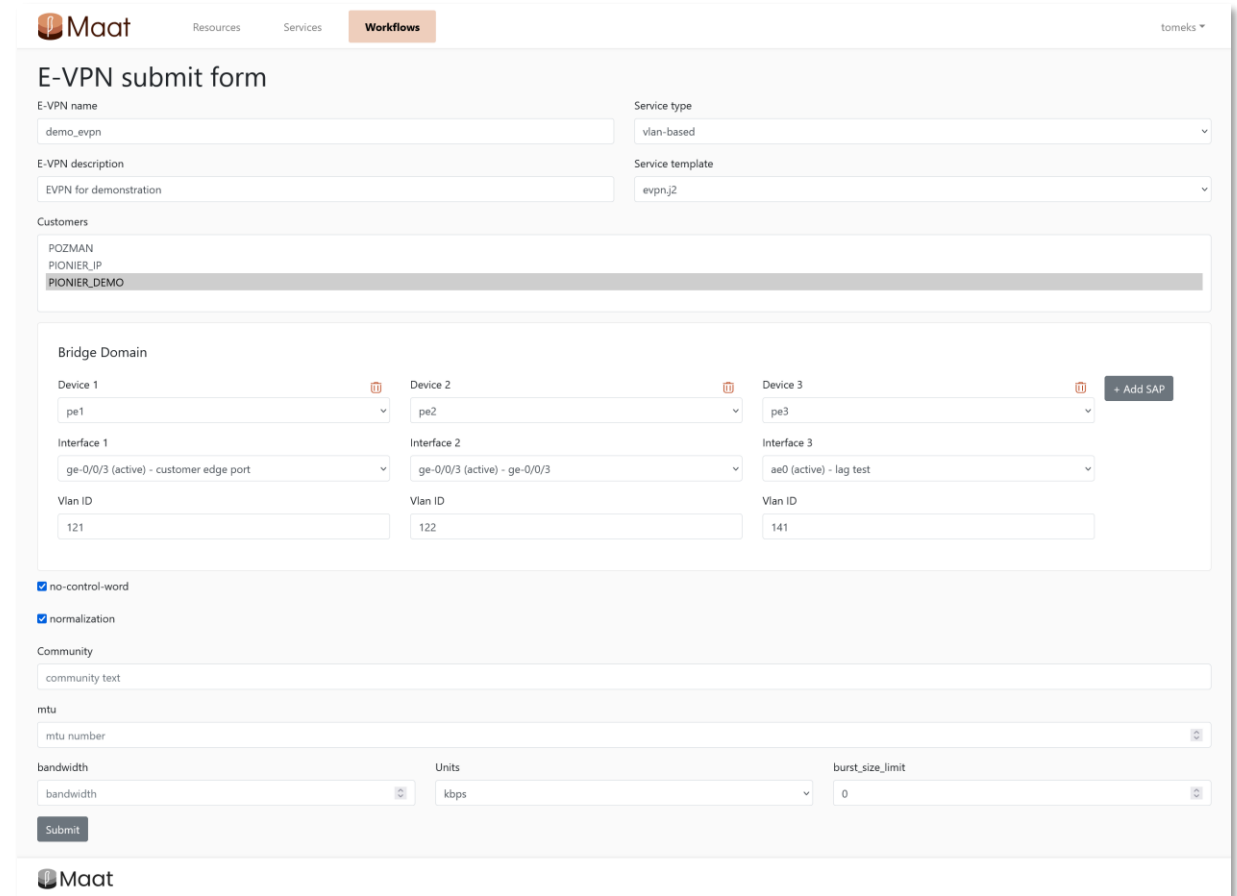
Service deployment

- Request remote Airflow DAG start



The screenshot shows the Maat UI interface with the 'Workflows' tab selected. The title 'Available processes input forms:' is displayed. Below the title, there are five buttons arranged vertically: 'L2 circuit submit form', 'Remove services form', 'Interface creation form', 'Interface removal/deconfigure form', and 'E-VPN creation form'.

Workflow input



The screenshot shows the Maat UI interface with the 'Workflows' tab selected. The title 'E-VPN submit form' is displayed. The form contains several input fields and dropdown menus:

- E-VPN name:** demo_evpn
- E-VPN description:** EVPN for demonstration
- Service type:** vlan-based
- Service template:** evpnj2
- Customers:** POZMAN, PIONIER_IP, PIONIER_DEMO
- Bridge Domain:**
 - Device 1:** pe1
 - Device 2:** pe2
 - Device 3:** pe3
 - Interface 1:** ge-0/0/3 (active) - customer edge port
 - Interface 2:** ge-0/0/3 (active) - ge-0/0/3
 - Interface 3:** ae0 (active) - lag test
 - Vlan ID:** 121, 122, 141
- no-control-word:** ☒
- normalization:** ☒
- Community:** community text
- mtu:** mtu number
- bandwidth:** bandwidth, Units: kbps, burst_size_limit: 0

The 'Submit' button is located at the bottom left of the form.

Interactions

Workflow output

- Task information
- LSO output
- Task result

Chatbot

- Interactive Maat query

Mattermost notifications

The image displays four overlapping screenshots of the Mattermost interface, illustrating workflow outputs and chatbot interactions.

Top Left Screenshot (Alarms): Shows a notification from AirFlow BOT at 02:00. It includes a table with workflow results:

Id	result
82be318f-947a-4bf0-8e71-5d08c26bc47e	Maat error
3662168d-6610-4691-8af6-09aa60a57baa	Verify

Top Right Screenshot (Notifications): Shows a notification from AirFlow BOT at 13:12 with the LSO output:

```
1 {
2   "job_id": "71adc096-c25c-4e82-8d52-7d2c39450cc1",
3   "output": "[
4     PLAY [Main workflow runner playbook] *****
5     TASK [Print the usage] *****
6     skipping: [mx-kra-1],
7     TASK [meta] *****
8     skipping: [mx-kra-1],
9     TASK [Including port role] *****
10    included: port for mx-kra-1,
11    TASK [port : Print debug info] *****
12    ok: [mx-kra-1] => {,
13      "msg": "Running port role with operation=add",
14    },
15    TASK [port : Including juniper/port_add.yaml] *****
16    included: /app/gap/ansible/roles/port/tasks/juniper/port_add.yaml for mx-kra-1,
17    TASK [Physical port setup] *****
18    included: build_config for mx-kra-1,
19    TASK [build_config : Set connection facts for Junos] *****
20    ok: [mx-kra-1],
21    TASK [build_config : Create temporary file] *****
22    changed: [mx-kra-1 -> localhost],
23    TASK [build_config : Render template juniper/port_add.j2 for mx-kra-1] *****
24    changed: [mx-kra-1 -> localhost],
25    TASK [build_config : Display rendered config file content] *****
26    changed: [mx-kra-1 -> localhost],
27    TASK [build_config : Show rendered config] *****
28    ok: [mx-kra-1] => {,
```

Bottom Left Screenshot (Alarms): Shows a notification from AirFlow BOT at 02:00, Part 2/2, with details for the workflow ID 3662168d-6610-4691-8af6-09aa60a57baa:

```
1 TASK [build_config : Verify juniper/l2circuit.j2 config on pe1] *****
2 changed: [pe1],
```

Bottom Right Screenshot (Chatbot): Shows a chat window with the Maat bot. It displays a table of network resources:

Resource	Id	Location
NORDUnet	l2.circuit	NORDUnet
Litnet	l2.circuit	Litnet
SURFnet	l2.circuit	SURFnet
Google_via_LIM	l2.circuit	Google via LIM
Microsoft	l2.circuit	Microsoft

Below the table, it shows a chat message from tomeks at 06:12: "lservice GEANT_AP1".

At the bottom, it shows a table of parameters for GEANT_AP1:

Parameter	Value
SAP ae6	ptx-poz-1
vlan_id	100
description	GEANT_AP1

Interactions

The screenshot shows the Maat AI Chat interface. The user prompt is "We need a short title 2-4 words descr...". The system prompt is "Using Custom System Prompt". The chat history shows a series of messages where the AI is refining its understanding of the user's request. The final message is a successful tool call to `maat_summarize`, which returns a summary of the Maat database content.

Summarize current content of maat database

The user asks: "Summarize current content of maat database". That's a request for an overview of what's in Maat: counts by category. According to instructions, first call `maat_summarize()` with default both. Then summarize in answer. So need...

Error calling tool `maat_summarize`

We need to call correct function name: `maat_summarize`. Use correct spelling.

Called tool `maat_summarize`

We need to summarize the current content of Maat database based on the summary data. Provide a textual summary: counts of resources and services per category, highlight key categories. Probably also mention that detailed queries can...

Maat Database – Current Inventory Summary (as of 2026-05-22)

Category	Count
Interface (all device ports, logical interfaces, LAGs, etc.)	1488
Customer (service customers)	56
Device – Router	49
Site (geographic locations/PoPs)	33
Device – Pseudonode (e.g., virtual routing nodes)	8
Model (hardware/platform models)	7
Vendor (equipment manufacturers)	3
Identifier (shared identifier pool)	2
Group (logical grouping of core routers)	1
Total resources	1647

AI experiments

The screenshot shows the Maat AI Chat interface. The user prompt is "We need a short title 2-4 words descr...". The system prompt is "Using Custom System Prompt". The chat history shows a series of messages where the AI is refining its understanding of the user's request. The final message is a failed tool call to `maat_get_services`, which returns an error message.

List in a table services for customer PIONIER_IP

We need to list services for customer named PIONIER_IP. Must first maybe summarize count? Steps: Use `maat_get_services` with filter `resourceRelationship.resource.name = "PIONIER_IP"`. That will retrieve services linked to that customer. Also...

Error calling tool `maat_get_services`

We called wrong function name, should be `maat_get_services`.

Error calling tool `maat_get_services`

The tool name is `maat_get_services` (per docs). Let's call correctly.

Error calling tool `maat_get_services`

The function name is `maat_get_services`, not `maat_`. Mistyped. Need to call correct function.

Error calling tool `maat_get_services`

The correct function name is `maat_get_services` (as per documentation). I attempted wrong ones. Need to call `maat_get_services`.

Called tool `maat_get_services`

We got the list of services that have `resourceRelationship.resource.name = "PIONIER_IP"`. That's presumably the customer name. The result shows 34 entries, all categories, mostly `I2.circuit` and a `I3.vrf` entry. Need to format a table with service...

Services attached to customer PIONIER_IP (34 total)

#	Service Name	Category
1	Pgw8-Lgw2	I2.circuit
2	SANET_bckp	I2.circuit
3	Google_LIM_bckp	I2.circuit
4	Microsoft_bckp	I2.circuit
5	PWSZ_Elblag_bckp	I2.circuit
6	RIPE_Atlas	I2.circuit
7	PWSZ_Elblag	I2.circuit
8	SANET	I2.circuit

Interactions

Maat AI Chat New Chat

Today

We need a short title 2-4 words descr...

① User wants details about service GEANT_AP1. That is one of the services listed, category I2.circuit. Need to retrieve its details via maat_get_services with filter name "GEANT_AP1". Then maybe also get related SAPs (service access points) via...

✓ Called tool maat_get_services

② We have service GEANT_AP1 details. We need to describe and provide details. Possibly also need to fetch SAP details. The service has two SAP relationships: SAP1.GEANT_AP1 and SAP2.GEANT_AP1. Might provide details about those services as...

✓ Called tool maat_get_services

③ Now need SAP2 details.

✓ Called tool maat_get_services

④ We have data for the service GEANT_AP1: It's an I2.circuit with description "GEANT AP1". It has two SAPs: SAP1.GEANT_AP1 and SAP2.GEANT_AP1. Each SAP has a VLAN ID characteristic: SAP1 VLAN100, SAP2 VLAN10. Resource...

Service Overview - GEANT_AP1

Property	Value
Name	GEANT_AP1
Description	"GEANT AP1" - a Layer-2 point-to-point circuit that connects the GEANT and PIONIER_IP customers.
Category	I2.circuit (Layer-2 circuit service)
State	active
Service-creation date	2025-03-26 12:58:50 UTC
Last update	2025-03-26 13:07:04 UTC
Service ID	9a82e06a-66d1-4f92-97ef-b522ac7b6e36
Related customers	• GEANT (resource ID 79a0aa7e-0c7f-4a80-baf5-cda4a67e8403) • PIONIER_IP (resource ID 9acb95cd-faee-4f37-bf52-f78913f19ddb)
Number of SAPs	2 (both are I2.circuit.sap objects)
SAP details	see below

Ask anything

+ MCP (1) x

Model: gpt-oss_120b

Describe and provide details about service
GEANT_AP1

AI experiments

Maat AI Chat New Chat

Today

We need a short title 2-4 words descr...

① We need to determine on which routers the service GEANT_AP1 is terminated. The service has SAPs linked to interfaces ae6 and ae4. Those interfaces belong to some device (router). Need to query interface resources with name "ae6" and...

✓ Called tool maat_get_resources

② We have retrieved three interface resources named ae6. They belong to routers: - pt x-poz-1 (router) for first ae6 (GEANT AP1A GA-01811) - pt x-poz-2 (router) for second ae6 (GEANT Hamburg AP1B) - mx-poz-1 (router) for third ae6...

✓ Called tool maat_get_resources

Routers that terminate the GEANT_AP1 service

The GEANT_AP1 Layer-2 circuit is realized by two Service Access Points (SAPs). Each SAP is bound to a physical interface, and each interface is attached to a router (device). From the interface records we can see the routers that host the endpoints:

SAP (service)	Interface (resource)	Router (device)
SAP1.GEANT_AP1	ae6 (ID 8d5802fa-ef9e-4c53-8a4e-37c8d3a13f54)	ptx-poz-1 (router ID 12a9744d-ca4e-448e-b6e2-e5b82ee3e945)
SAP2.GEANT_AP1	ae4 (ID 7214bf01-9f4c-473c-ab58-975c379990eb)	mx-poz-1 (router ID 13825043-fcad-4a62-9551-0949c9cd9cfe)

Quick summary

- Terminating routers: ptx-poz-1 and mx-poz-1
- Endpoints:
 - VLAN 100 on interface ae6 of router ptx-poz-1
 - VLAN 10 on interface ae4 of router mx-poz-1

These two routers constitute the two ends of the GEANT_AP1 point-to-point Layer-2 circuit.

Ask anything

+ MCP (1) x

Model: gpt-oss_120b

On which routers is this service
terminated?

Key Takeaways

Continuous improvement

- **Team interworking:** Network engineers & software developers
- **Understand the goal:** final result must be clear to all team members to achieve synergy
- **It's journey, not a destination:** Machine once started must be maintained continuously - otherwise it will break down
- **Engineering not magic:** Tools are just tools (even those magnificent AI based)

Future

- Extend workflows for new network services and operations
- Data model extensions: intent-based SLA enforcement, multi-domain & multi-operator federation, capacity & utilization tracking
- AI support – more MCP servers to access multiple data sources, use of AI agents to run the workflows
- Engage AI to verify data consistency and network configuration status
- Self-services for the PIONIER partners



Poznańskie Centrum Superkomputerowo-Sieciowe
Poznan Supercomputing and Networking Center

Thank you

tomasz.szewczyk@pcss.pl

roman.lapacz@pcss.pl



Co-funded by
the European Union



The scientific work is published for the realization of the international project co-financed by Polish Ministry of Science and Higher Education from financial resources of the programme entitled "PMW".