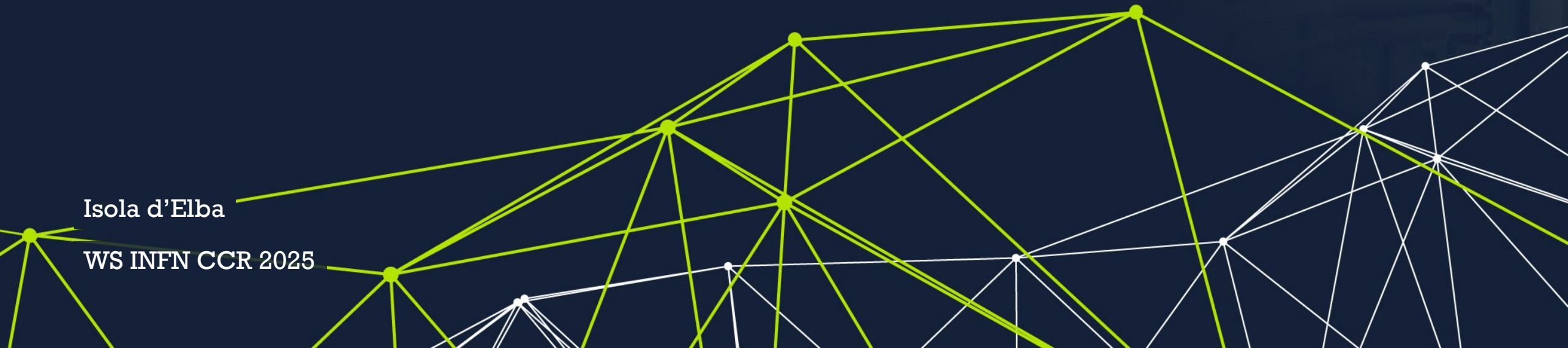
Consortium GARR | THE ITALIAN EDUCATION & RESEARCH NETWORK

Automazione della rete ottica in GARR-T: approcci, sfide e soluzioni

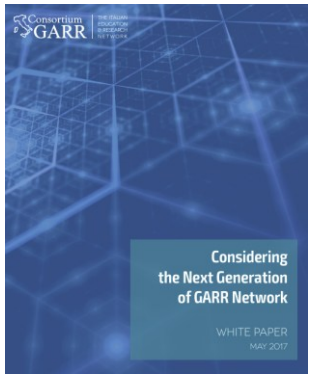
PAOLO BOLLETTA, MATTEO COLANTONIO



Isola d'Elba

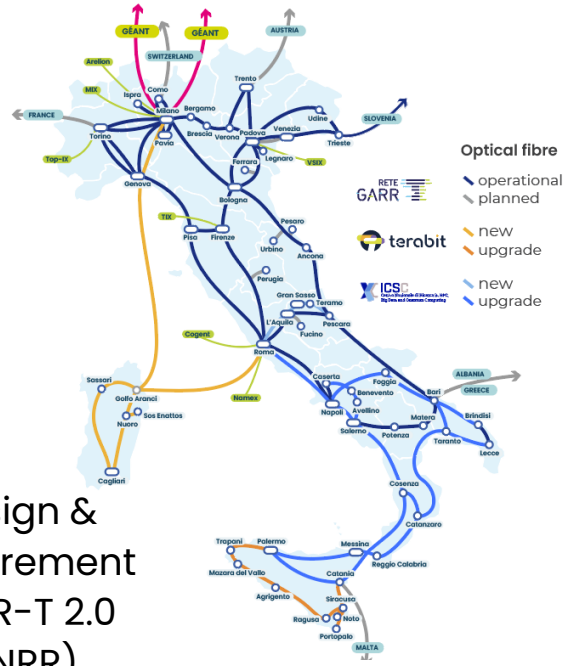
WS INFN CCR 2025

RETE GARR



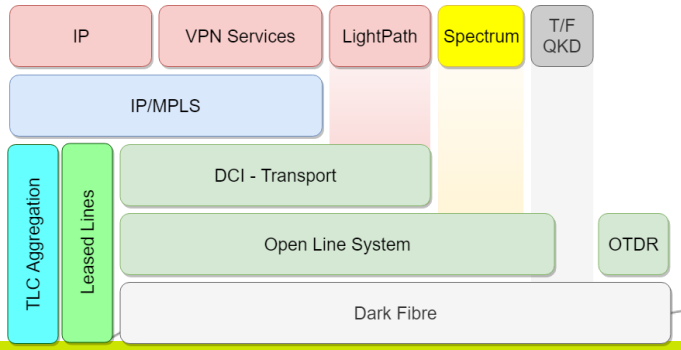
ELISA-WG
2017

GARR-T Procurement
2020

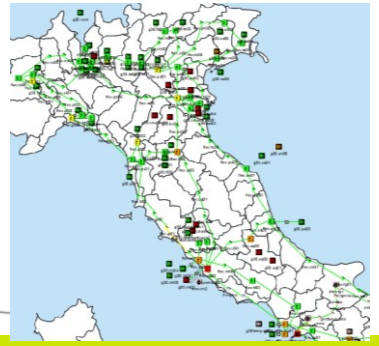


Design & Procurement GARR-T 2.0 (PNRR)
2023

2019
Design & Architecture GARR-T



2021-2023
Implementation and migration to GARR-T



2024-2025
Implementation and migration GARR-T 2.0 (PNRR)

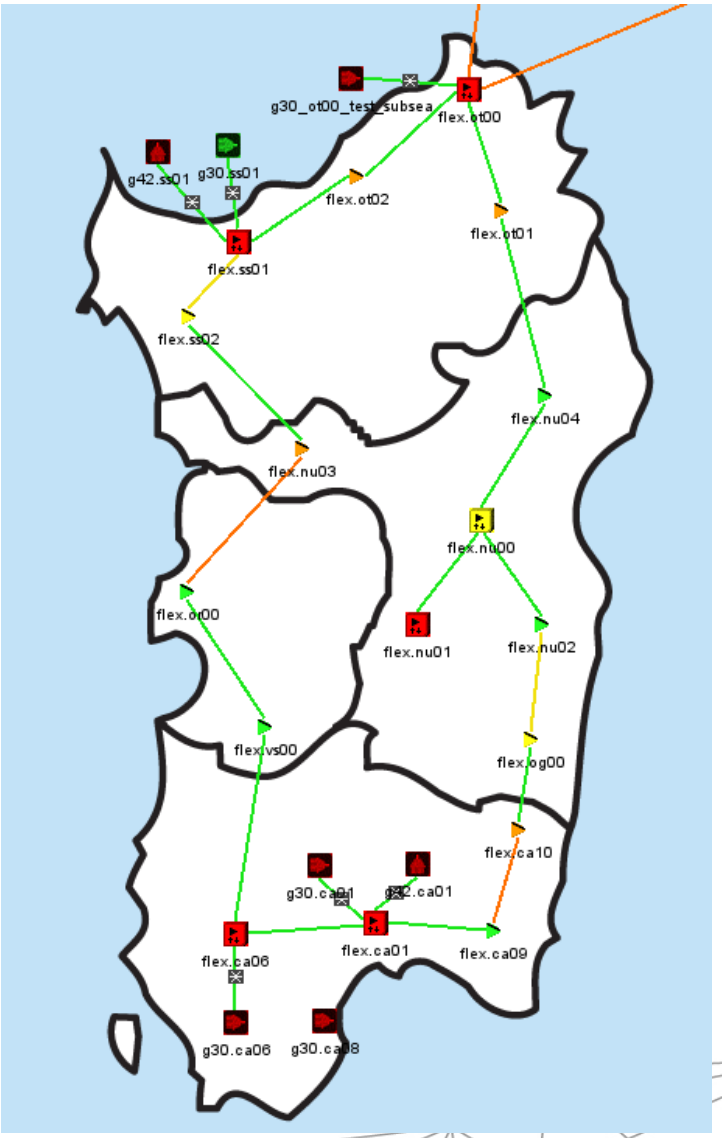
Greenfield Terrestrial Infrastructure

100% Hardware delivered and available for installation!

Terrestrial fiber spans ≈85% delivered

Implementation:

- **Sardinia**: infrastructure 100% deployed, capacity provisionin ongoing .
- **Abruzzo**: behind schedule we are confident to catch up!



Optical Line System Migration

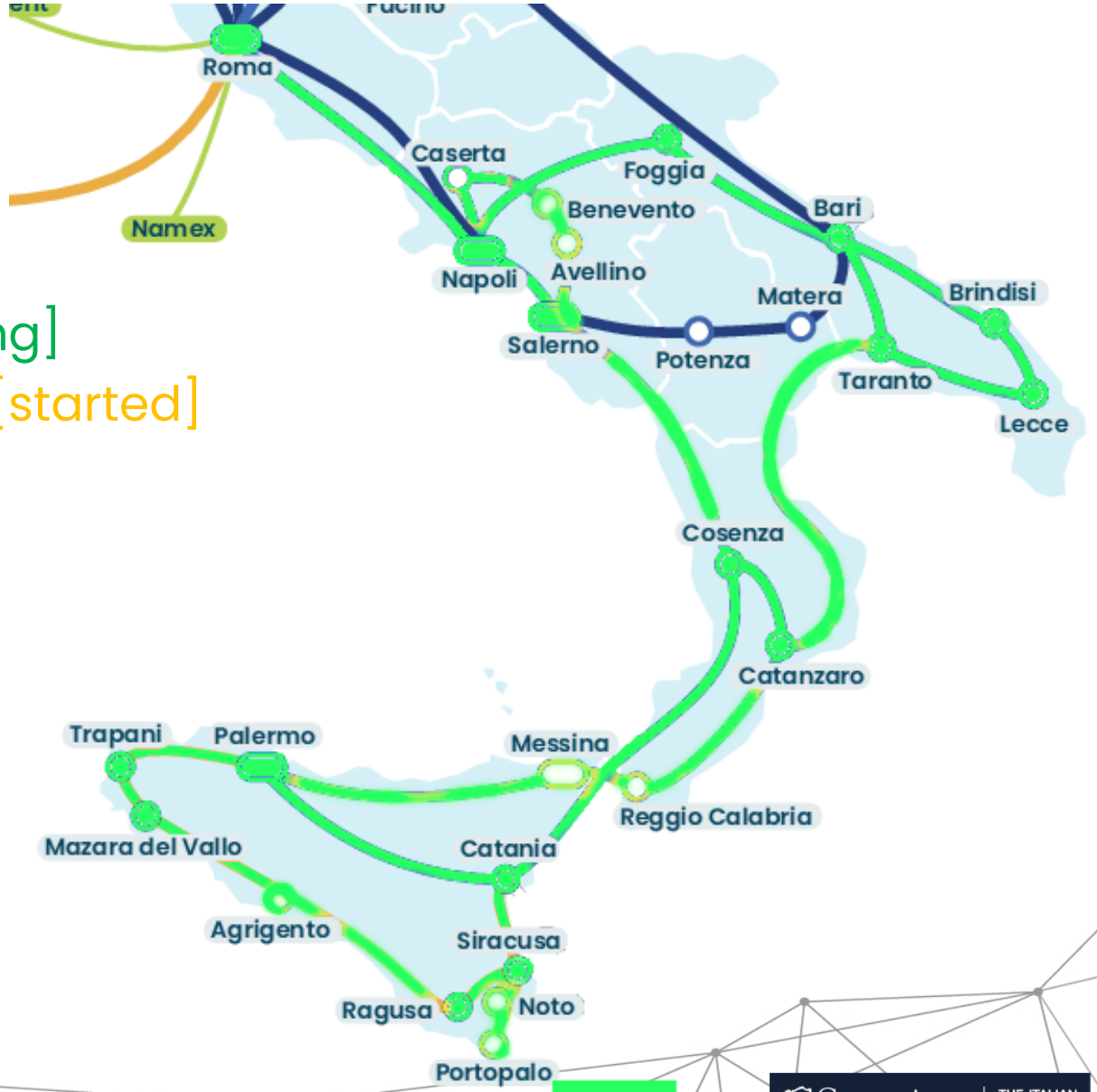
Disaggregated Approach:

- Hybrid Line System Platfom operations
- STEP1: Line system migration [DONE!]
- STEP2: Turn-up of the new capacity [ongoing]
- STEP3: decommissioning legacy OTN layer [started]

First node migrated May 2024

Current Status:

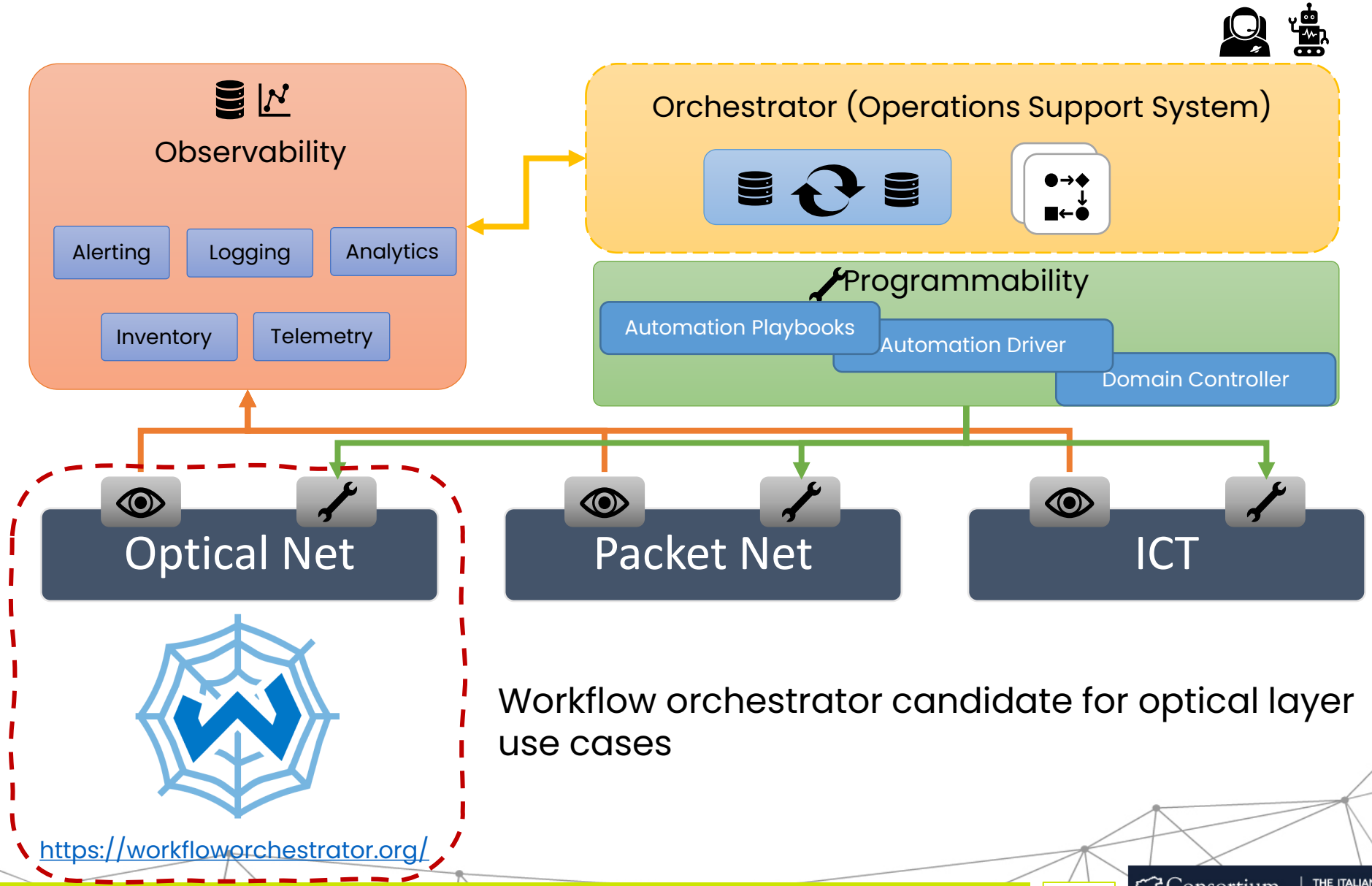
- A/D nodes migrated: 23/23
- ILA nodes migrated: 34/34
- 5 out to 6 Satellite nodes (H4) migrated



Infrastructure Complete – Overlay and Orchestration Kick In

Orchestration and Automation **Pilot** Use Cases:

1. Optical Circuits provision
2. Dark Spectrum provisioning



Workflow orchestrator candidate for optical layer use cases

Exp.#1: Ansible

We did upgrade 92 transponders on schedule with AWX but…

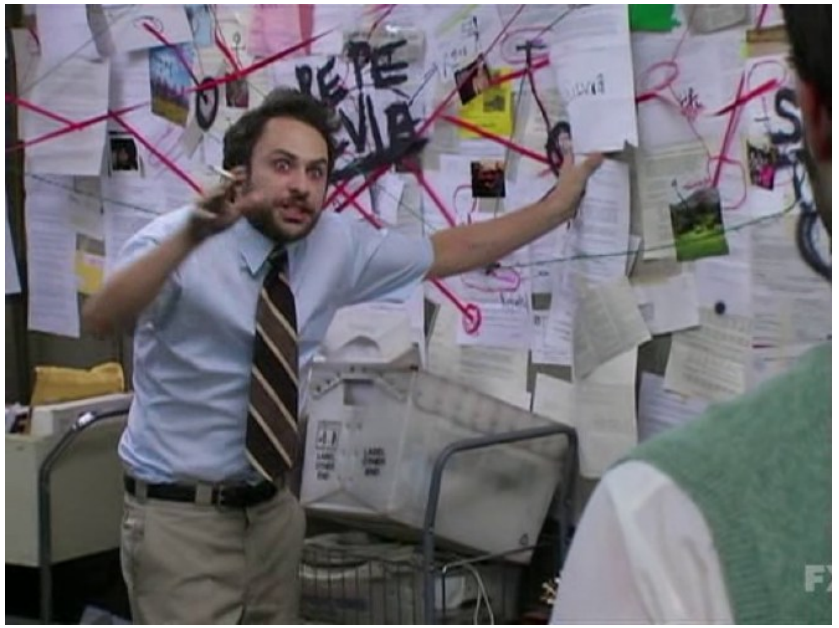
Job ID	3626	**Status**	Successful	**Started**	3/20/2024, 6:45:43 AM
Finished	3/20/2024, 8:12:20 AM	**Job Template**	G30 Upgrade Template	**Job Type**	Playbook Run
Launched By	G30 Upgrade Scheduler	**Inventory**	G30 Inventory	**Project**	G30 Upgrade Project
Revision	cf41b1baea7eef7ab420d04ce9b6f49dabc5d2f8	**Playbook**	playbook.yml	**Verbosity**	0 (Normal)
Execution Environment	GARR EE 3.0.1	**Controller Node**	awx-ba1-task-d7658bdcb-d72rn	**Container Group**	K8S-BA DCN nodes
Job Slice	0/1	**Forks**	0	**Timeout**	No timeout specified
Created	3/20/2024, 6:45:27 AM	**Last Modified**	3/20/2024, 6:45:42 AM		

Exp.#1: Ansible Shortcomings

Calling one function in a loop:

.

```
├── bax_check
│   ├── bax_card_checks.yml
│   └── bax_card_restart.yml
└── playbook.yml
```

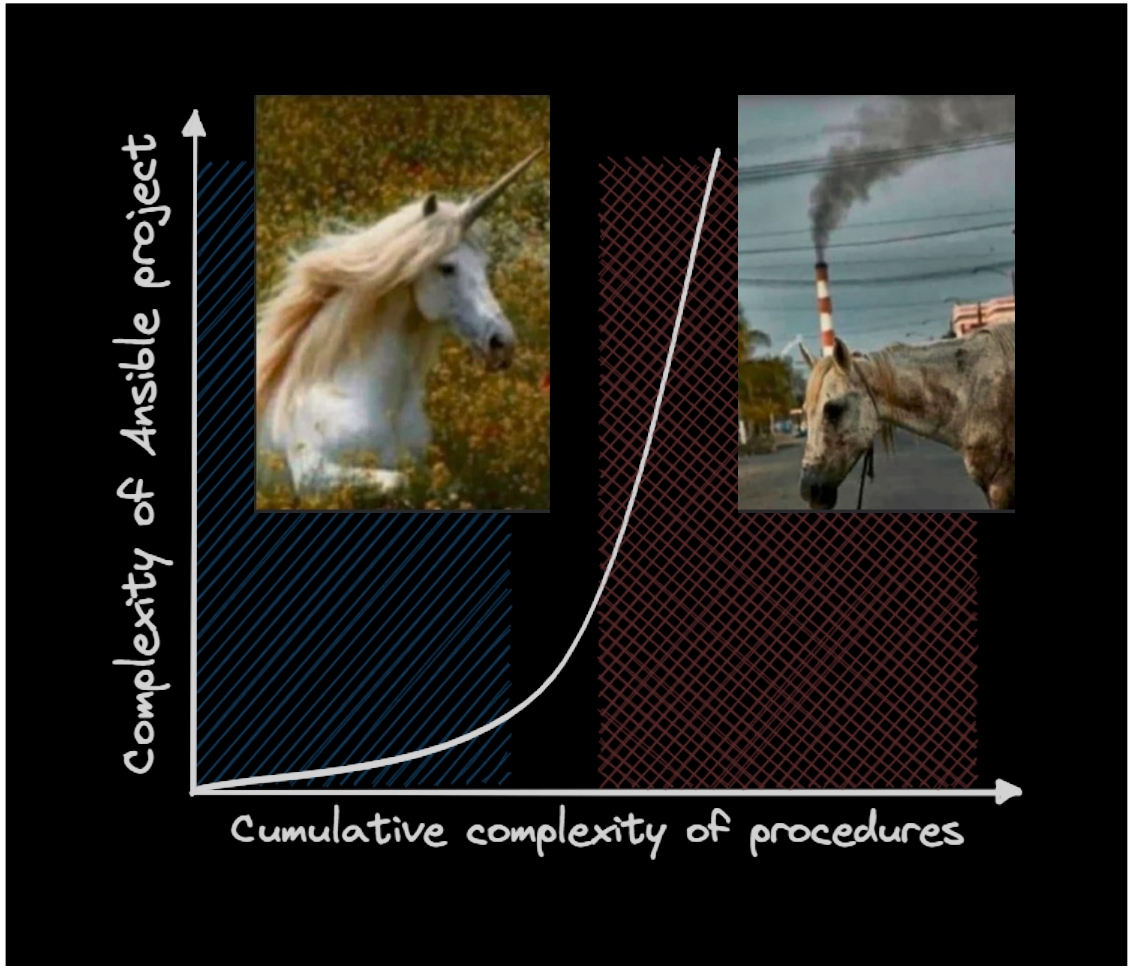


Devices with TL1 interface? DIY modules!

ENT-OEL::prova:trnbi:::LABEL=test,SRCNODENAME=ols-a,DSTNODENAME=ols-b,MODULATION=PM-NONE, RATE=NA,CARDTYPE=UNKNOWN,FREQSLOTPLANTYPE=FREQ-SLOT-PLAN-NONE,VALIDFRANGELIST=191325000&196125000,EXPLICITROUTE=ols-a&1-A-2-L1&ols-b&1-A-1-L1,OELSOURCE=MANUAL,GAURDBAND=0,NUMFECITRNS=4:IS;

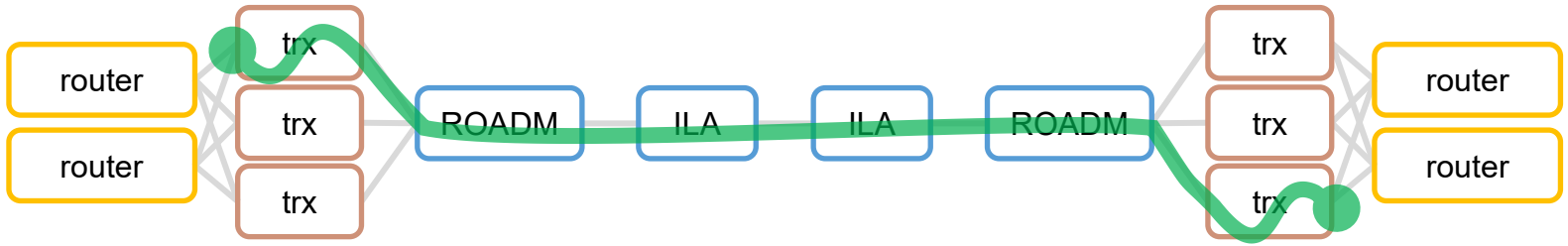
Exp.#1: Ansible - Lesson Learned

- Great to keep your Infrastructure Running and Clean!
- Great for very simple procedures
- Config-centric, not service-centric
- Use other programming languages for complex ones



Exp.#2: Vendor Controller's API/NBI

Goal: automate optical circuits provisioning

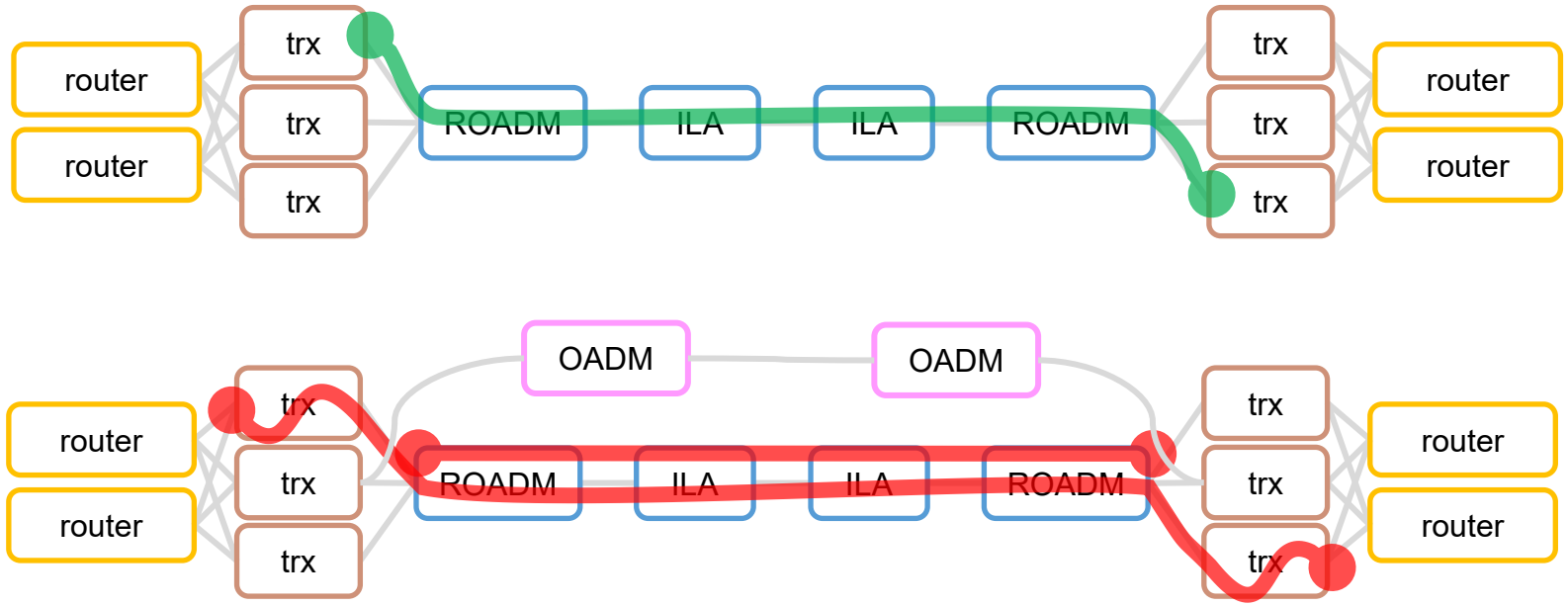


Manual procedure: ~40-50 clicks across 4 GUIs

Controller's API?

Exp.#2: Vendor Controller's API/NBI - Shortcomings

Can do, but only Line-to-Line



Not Client-to-Client

Not Spectrum Connections

Not Metro H4/Satellite sites

Exp.#2: Vendor Controller's API/NBI – Lesson Learned

- Can be a bottleneck
- You're not in the driving seat
- Use device interfaces (TL1, NETCONF, RESTCONF)

Here we are! - The Workflow Orchestrator

- Developed by SURF + ESnet ENERGY SCIENCES NETWORK
- open-sourced
  and registered in the Commons Conservancy
- workfloworchestrator.org
- Adopters and contributors:

GÉANT Networks • Services • People

HEAnet Ireland's National Education & Research Network

Consortium GARR

What you get out of the box

An open-source framework where you:

Define your network services /entities **(Products)**

Track individual customer instances of those Products **(Subscriptions)**

Define clear procedures **(Workflows)** for creating, modifying, validating, and terminating Subscriptions

- FastAPI core engine + NextJS UI
- Everything is stored and tracked in a database

Products: modeling your reality

Define and link your building blocks:

```
class OpticalFiberBlock(ProductBlockModel, product_block_name="OpticalFiber"):
    fiber_name: str | None = None
    oss_id: int
    terminations: ListOfPorts[OpticalDevicePortBlock]
    # … other fields
```

Turn them into a Product:

```
class OpticalFiber(SubscriptionModel):
    optical_fiber: OpticalFiberBlock
```

Workflows: making things happen

Define and link your building blocks:

```
>> reserve_resources_on_netbox                  # Regular step
>> confirm_patching                             # User input step
>> check_reachability                           # Retry step
>> callback_config_manager                      # Callback step
>> conditional(should_enable_monitoring)(       # Conditional step
       enable_monitoring                        # Group of steps
   )
```

From 50 clicks...

Start / Workflows / 95cc235f-076d-4a6a-a2f7-eb96ee3cdc3a

Create optical_digital_service

Retry Abort

optical_digital_service

Status	Current step	Customer	Started by	Started on	Last update	Related subscriptions
COMPLETED	Done	Default::Orchestrator-Core Customer	SYSTEM	6:12:39 PM	6:13:33 PM	f001c01 OCh042 ba01-bo01 (100Gbps E...

Workflow steps Expand all

Show subscription delta Options

Start
success - 4/23/2025, 6:12:41 PM
Duration 00:00:01

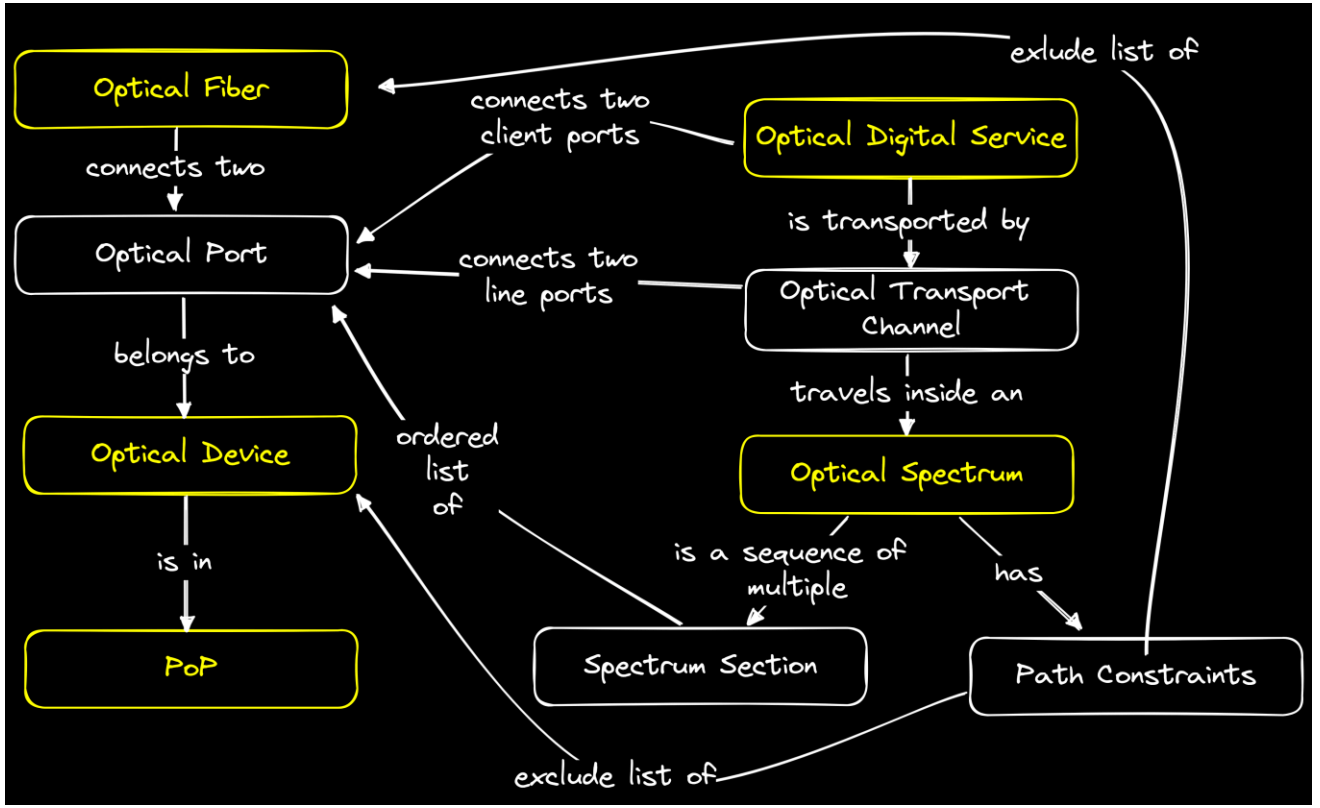
Saving input data into the optical digital service model
success - 4/23/2025, 6:12:41 PM
Duration 00:00:02

Create Process Subscription relation
success - 4/23/2025, 6:12:41 PM
Duration 00:00:00

Path computation
success - 4/23/2025, 6:12:42 PM
Duration 00:00:01

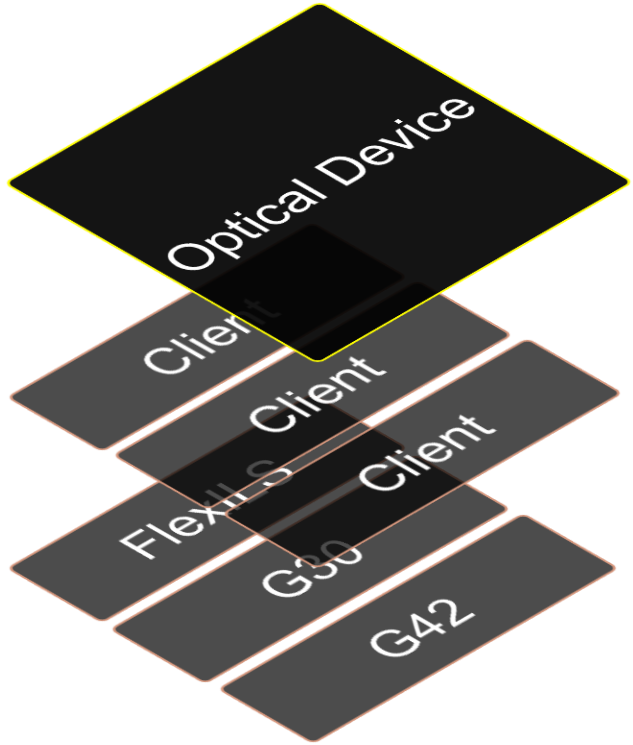
Abstract Composable Models and Device Clients

- A coherent architecture that aligns with our mental models



Abstract Composable Models and Device Clients

- A coherent architecture that aligns with our mental models
- Clients to decouple from hardware



One function to rule them all

- Problem: same task, different platform
- For example, in a workflow step:

```
for port in
subscription.optical_fiber.terminations:
    device = port.device
    port_name = port.port_name
    set_port_admin_state(device, port_name, "up")
```

```python
@set_port_admin_state.register(Platform.FlexILS)
def _(
    optical_device: OpticalDeviceBlock,
    port_name: str,
    admin_state=Literal["up", "down", "maintenance"],
) -> Dict[str, Any]:
    # FlexILS implementation

@set_port_admin_state.register(Platform.G30)
def _(…same args…) -> Dict[str, Any]:
    # G30 implementation
```

Benefits: keeps code organized, easy to add new platforms, simplifies workflow logic

Keep it simple, Talk Direct

Problem:
- NBI = loss of control/functionalities
- Ansible = just adds complexity

Idea: communicate directly with devices like any other API

Benefits: simplicity and maintainability

Easy? No. Worth it?

Yes! This approach is far more powerful than relying on playbooks or scripts.

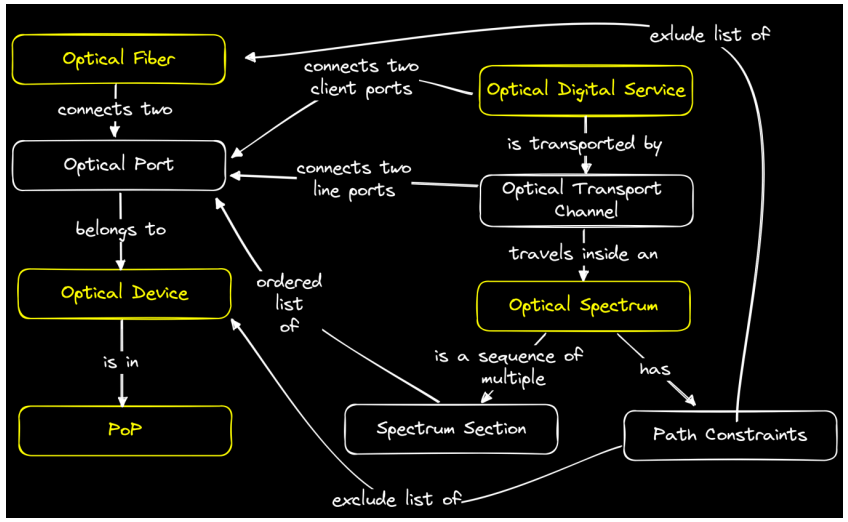
- Central definition of our services
- Consistent services' lifecycle management with less operational effort
- Visibility into the state and history of each service
- A reusable toolkit to automate tasks involving devices, systems, and people
- Hardware changes, services and procedures stay

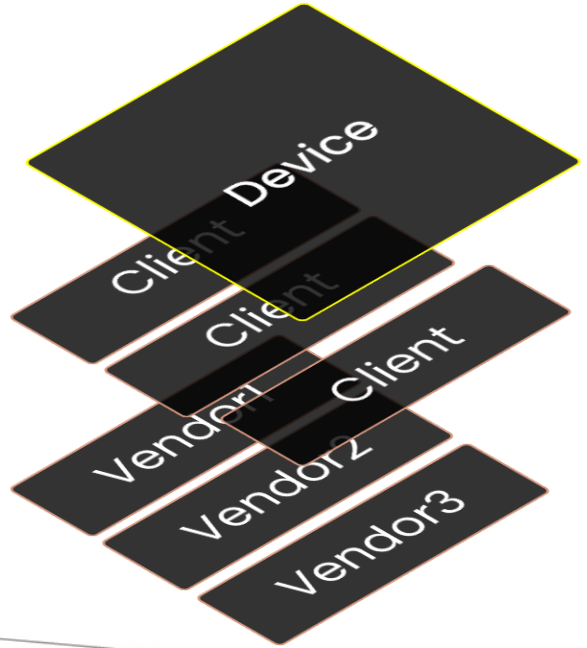
Key Take-Aways

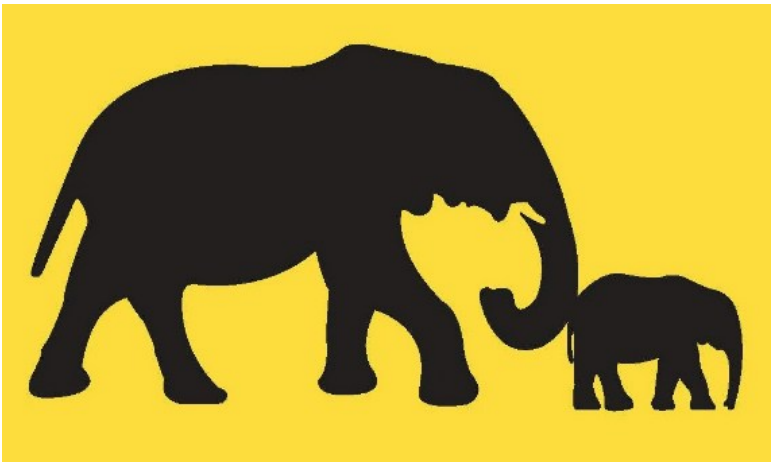
Scalable and maintainable? Use composable models and stateful instances and procedures

Complex logic? Use devices' programmable interfaces and high-level programming languages

Sustainable transformation? Automate one service at a time, lowering effort to nudge adoption







Conclusion

Achievements:

- GARR-T roll out and migration has benne completed!!
- Enhanced new services (e.g. multidomain 1.6Tbps DCI)
- Telemetry and Automation

Opportunities:

- Expansion in new areas
- Complete GARR-T as a unified network platform
- Spectrum over subsea cables

Next Steps:

- Network Data and Digital Twin
- Orchestrator and Self-Service Portal
- Beyond Data Services