

Leveraging Open Source for Network Automation

A Practical Approach with MCP, GitLab, NetBox & Ansible

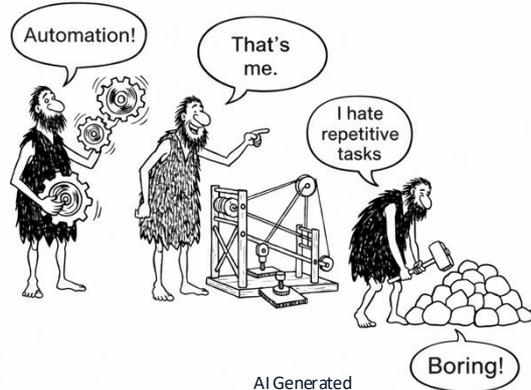
Patrick Mosimann
Solutions Engineer



About Patrick...



19+ years in Networking
3 years as customer
3 years as a partner
3 years as a competitor
10+ years at Cisco



Why this session?

From NetDevOps to AI-Assisted Operations

Automating Networks with a Source of Truth

Patrick Mosimann
Solutions Engineer

BRKOPS-2357

CISCO Live!



AI Generated

The slide features the OST logo (Eastern Switzerland University of Applied Sciences) and the title 'Network Observability Stack'. The subtitle is 'Redefined with a Modern Open-Source Pluggable Tech-'. The authors listed are Ramon Bister, Sascha Häring, and Jan Untersander, dated 5 December 2025. The footer identifies the organization as the 'Institute for Network and Security @ Eastern Switzerland University of Applied Sciences'.

Important project goal: “Deterministic”

- **Definition**

- Outcome is fully determined by initial conditions (no randomness).
- Same input → same output, every time.

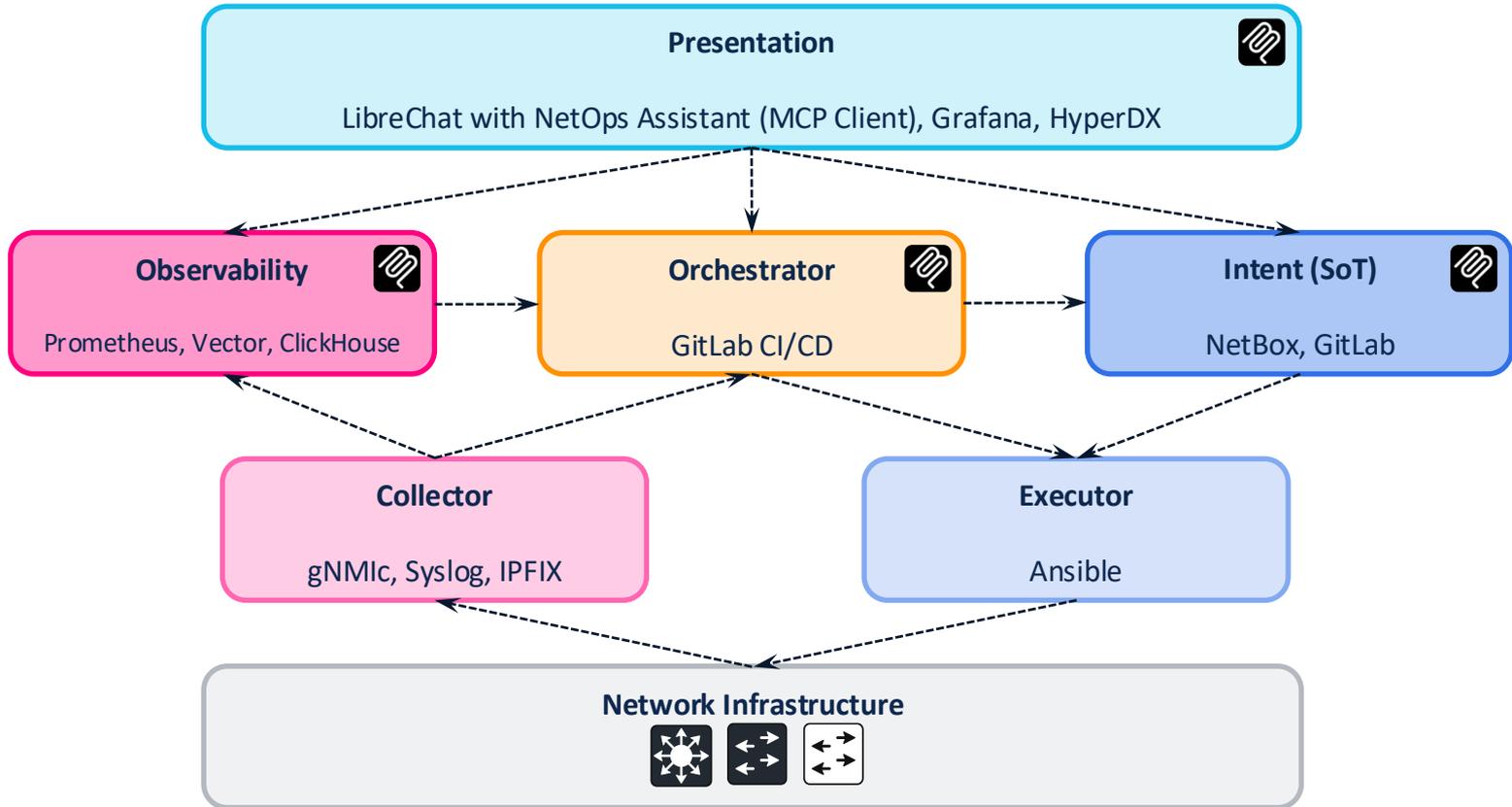
- **Contrast: LLMs**

- LLMs appear non-deterministic: they use randomness (e.g. sampling, temperature) for diverse, natural answers.
- You can reduce randomness (e.g. temperature=0) but they’re still not strictly deterministic like a program.

- **Possible solution**

- **GitLab CI/CD + Ansible** = deterministic automation.
- Same playbooks, inventory, and variables → same actions; the pipeline runs the same steps every time.

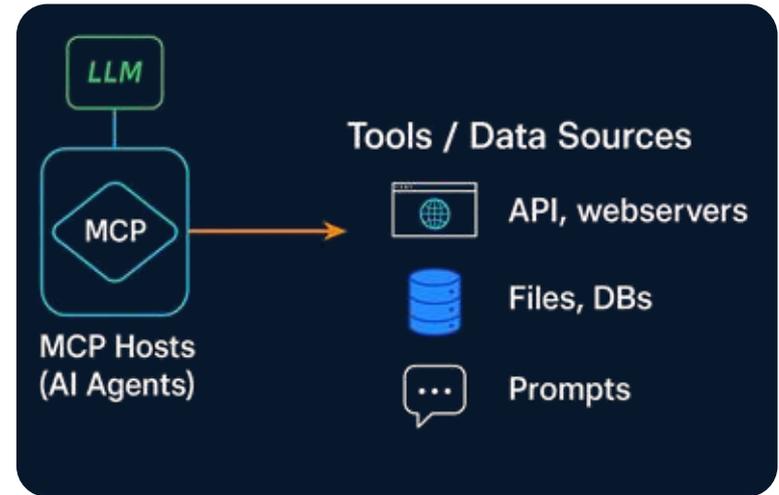
The Toolchain (NAF-aligned)



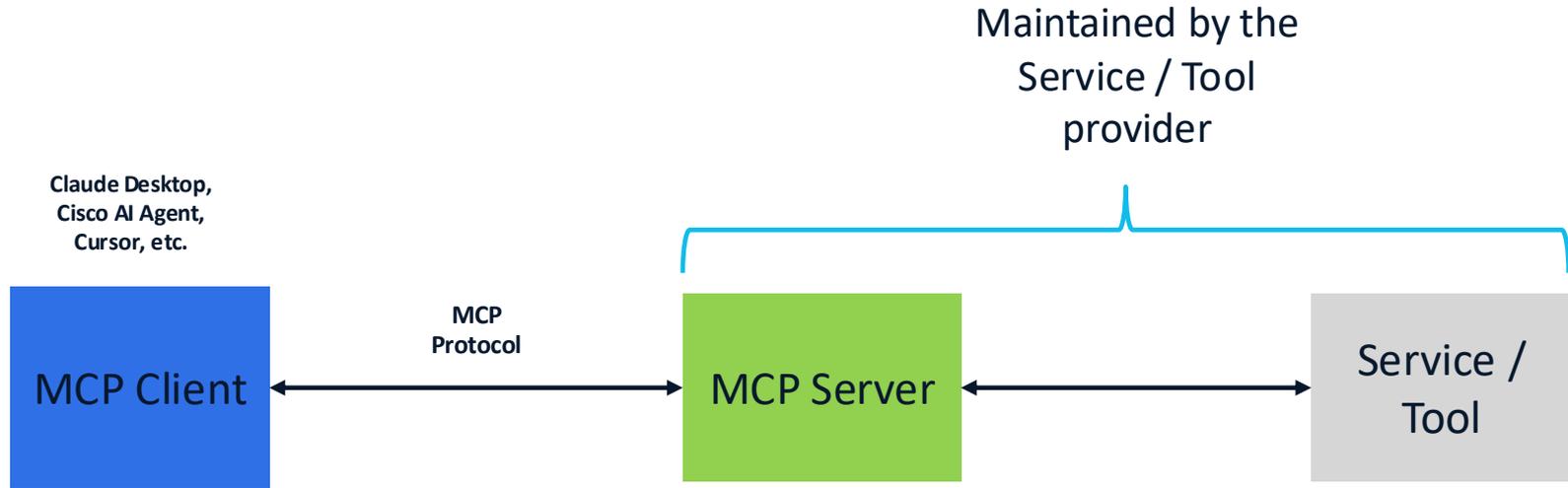
Reference: <https://reference.networkautomation.forum/Framework/Framework/>

MCP - The USB-C for AI Agents

- **Model Context Protocol (MCP):**
 - **Open standard** developed by **Anthropic**
 - Connects **AI Agents** with external data & tools
 - Defines a simple **Host-Protocol-Server architecture**
 - Exposes **primitives**: Tools (actions), Resources (data), Prompts (templates)



How MCP Works



MCP Client

- A software agent with integration to LLM model(s)
- Orchestrates conversation flow
- Initiates connections to MCP servers
- Handles user requests and responses

MCP Server

- Connects to tools and functions
- Acts like a “translator” between the tool and the client
- Can act as a security gateway

Code example of IOS XE MCP Server

```
mcp = FastMCP("ios-xe-mcp-server")
```

```
@mcp.tool()
```

```
def show_command(command: str, host: str) → str:
```

```
    """  
    Executes a 'show' command via SSH (Netmiko) on an IOS XE device.
```

```
    Args:
```

```
        command: Show command to execute (e.g., 'show ip interface brief')
```

```
        host: IP address or hostname of the IOS XE device
```

```
    Returns:
```

```
        Command output as string or error message
```

```
    """
```

```
    # Validate required parameters
```

```
    if not host:
```

```
        return "Error: host parameter is required"
```

```
    # Create device connection dictionary
```

```
    device = create_safe_device_dict(host, DEFAULT_USERNAME, DEFAULT_PASSWORD)
```

```
    try:
```

```
        # Log connection attempt with masked password
```

```
        ...
```

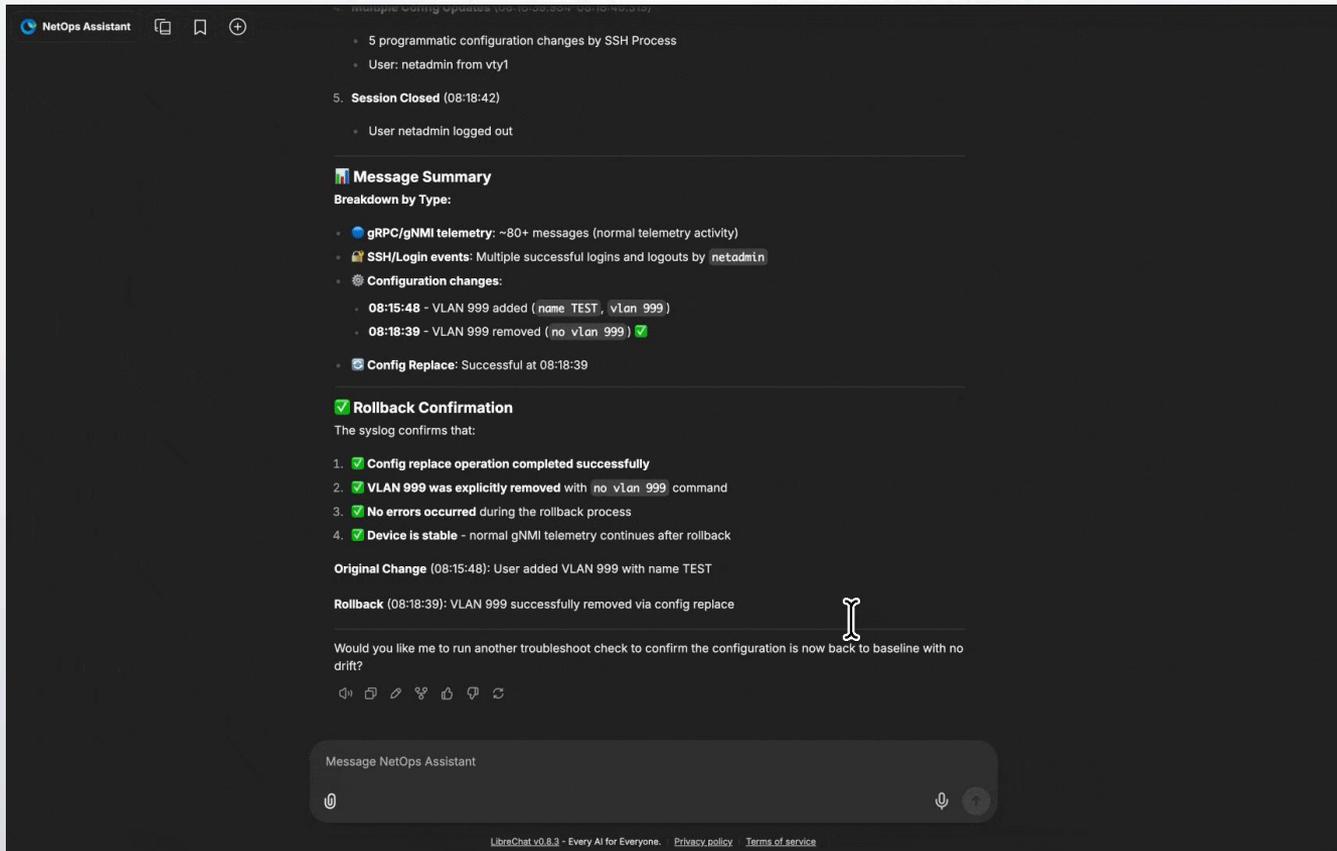
```
    except Exception as e:
```

```
        # Sanitize error message to remove any exposed passwords
```

Get the code: <https://github.com/pamosima/netops-stack/>

AI Flows

- **Flows = Resources + Tools.**
- **Runbook text = Resources** (application-controlled context).
 - The flow text lives in `flow_resources.py` and is exposed as MCP resources (e.g. `netops://flows/troubleshoot`, `netops://flows/apply_config`).
- **Executing steps = Tools** (model-controlled actions).
 - **Flow tools:** `flow_run_troubleshoot_flow`, `flow_run_apply_flow`, `flow_trigger_compare_pipeline`, etc.
 - **Domain tools:** `netbox_*`, `prometheus_*`, `clickhouse_*`, `gitlab_*`, `ios_xe_*`.



Get the code: <https://github.com/pamosima/netops-stack>

Key Takeaways

1

Single source of truth (SoT):

NetBox + GitLab for intent and config; Prometheus + ClickHouse for metrics and logs so “should be” and “is” are in one loop.

2

Deterministic toolchains:

Ansible + GitLab CI/CD for repeatable collect, compare, apply, and rollback.

3

MCP and AI:

MCP server with runbooks and tools so LLM drives NetBox, Prometheus, ClickHouse, and GitLab without skipping automation.

Additional Resources

- <https://github.com/pamosima/netops-stack>
- <https://github.com/Untersander/gnp-stack>
- <https://github.com/pamosima/network-mcp-docker-suite>
- <https://github.com/pamosima/BRKOPS-2357>
- <https://developer.cisco.com/site/ai/>
- <https://reference.networkautomation.forum/Framework/Framework/>

Thank you



