ProvTalk: Interpretable Multi-level Provenance Analysis in Network Functions Virtualization (NFV)

Azadeh Tabiban[1], Heyang Zhao[1], Yosr, Jarraya[2], Makan Pourzandi[2], Mengyuan Zhang[3] and Lingyu Wang[1]

[1]Concordia University, Canada
[2]Ericsson Security Research, Canada
[3]The Hong Kong Polytechnic University, China

NDSS Symposium 2022
San Diego, California

Outline

① Introduction

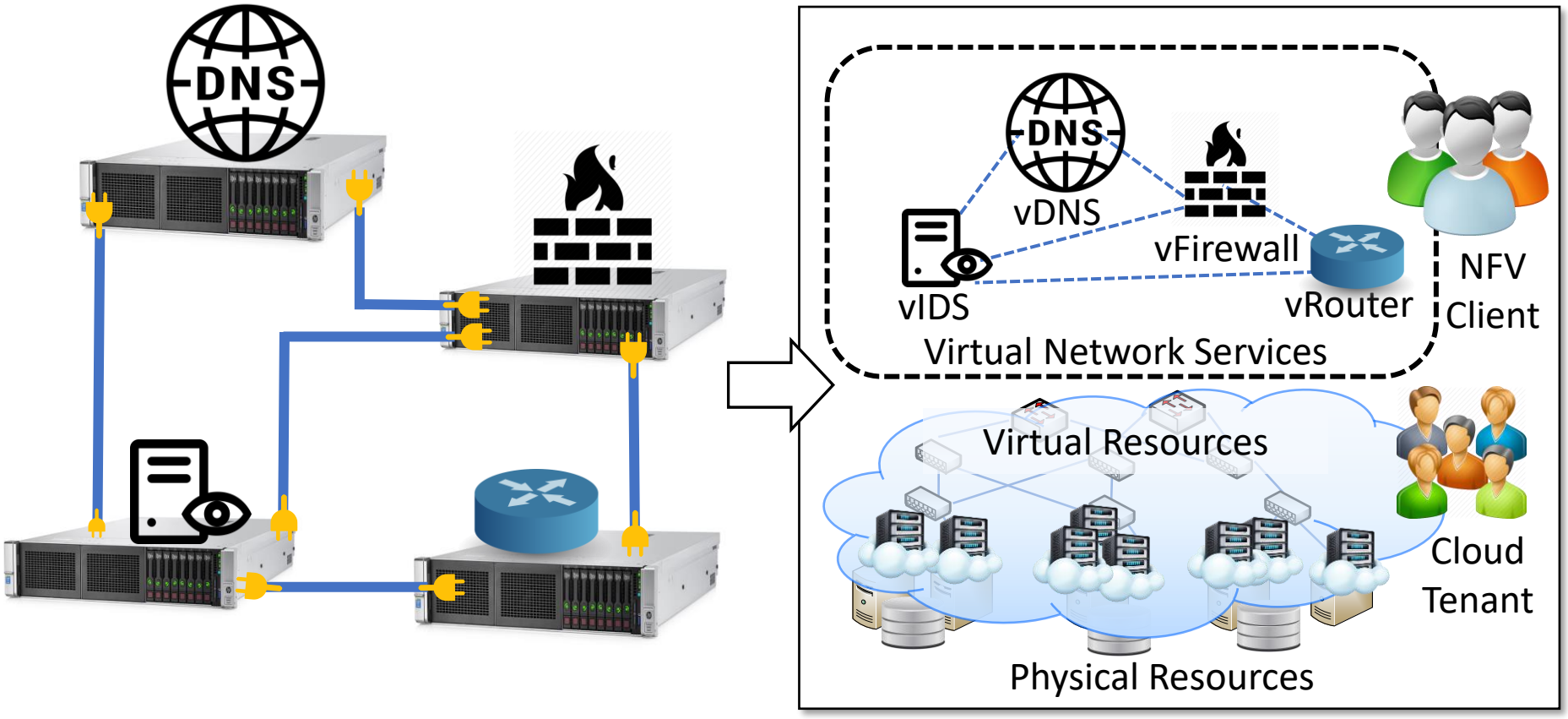
② Methodology

③ Implementation

④ Evaluation Results

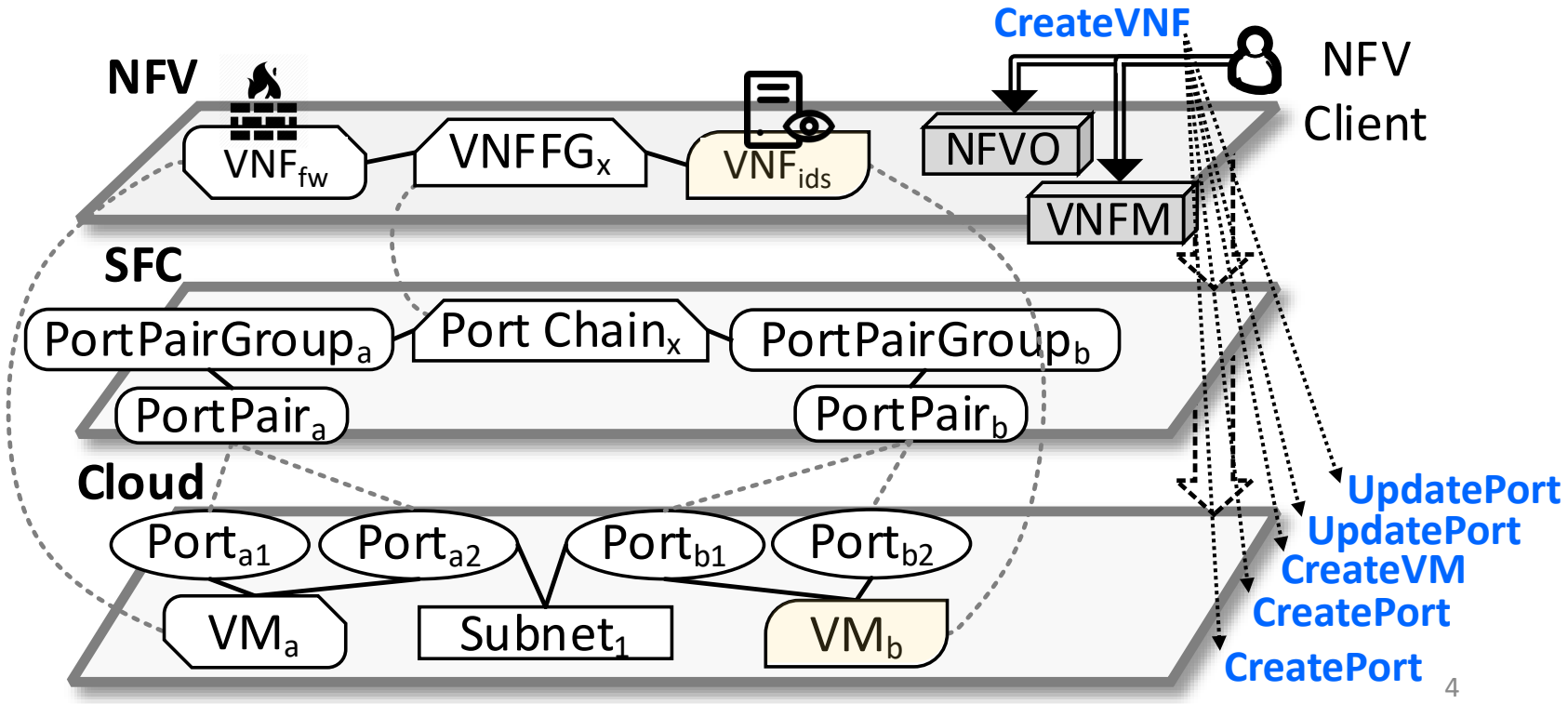
⑤ Concluding Remarks

Multi-level NFV Deployment Model



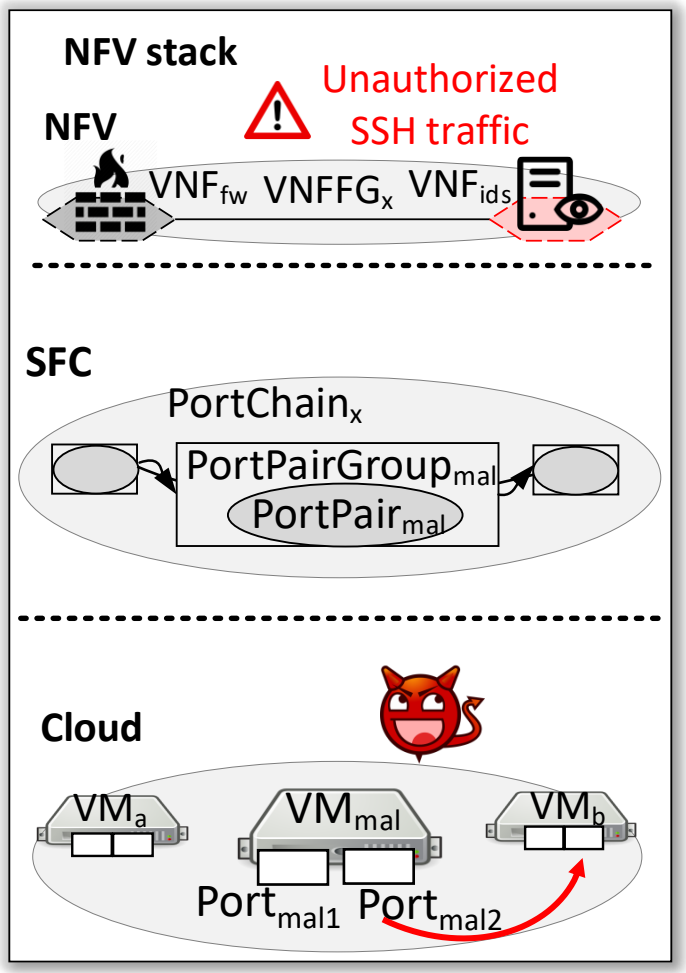
Multi-level NFV Deployment Model

- NFV enables dynamic and agile network services deployment
- However, the multi-level architecture also leads to a higher complexity and greater chances of misconfigurations

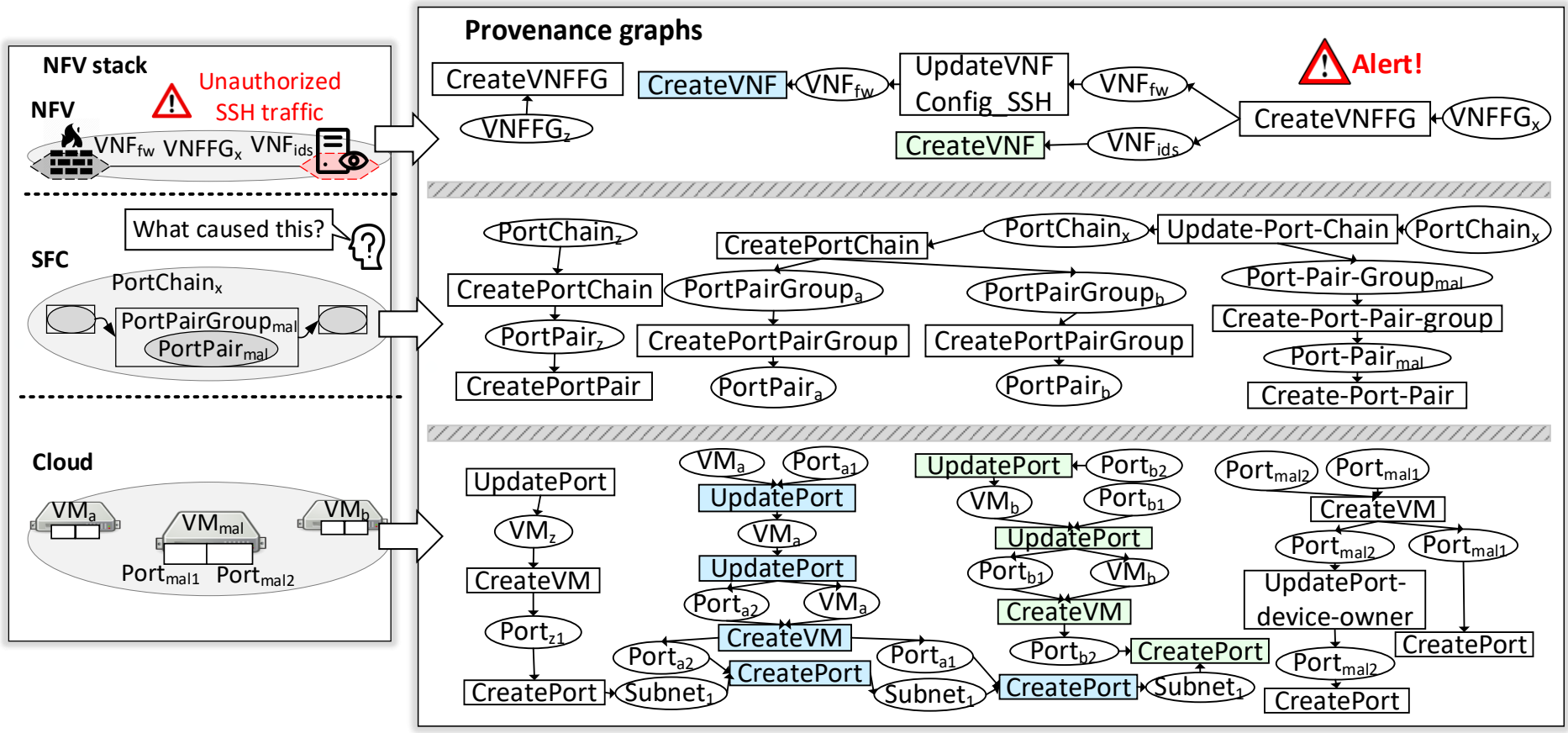


Challenges of Root Cause Analysis in NFV

- An analyst receives an alert from the virtual IDS (VNF_{ids}) about unauthorized SSH traffic
- **Root cause**:
  - The attacker creates $Port_{mal1}$ and $Port_{mal2}$
  - He/She updates the device owner field of Portmal2, and creates VM_{mal} attached to this port
  - The attacker inserts the port pair group of VM_{mal} into the port-chain corresponding to $VNFFG_x$

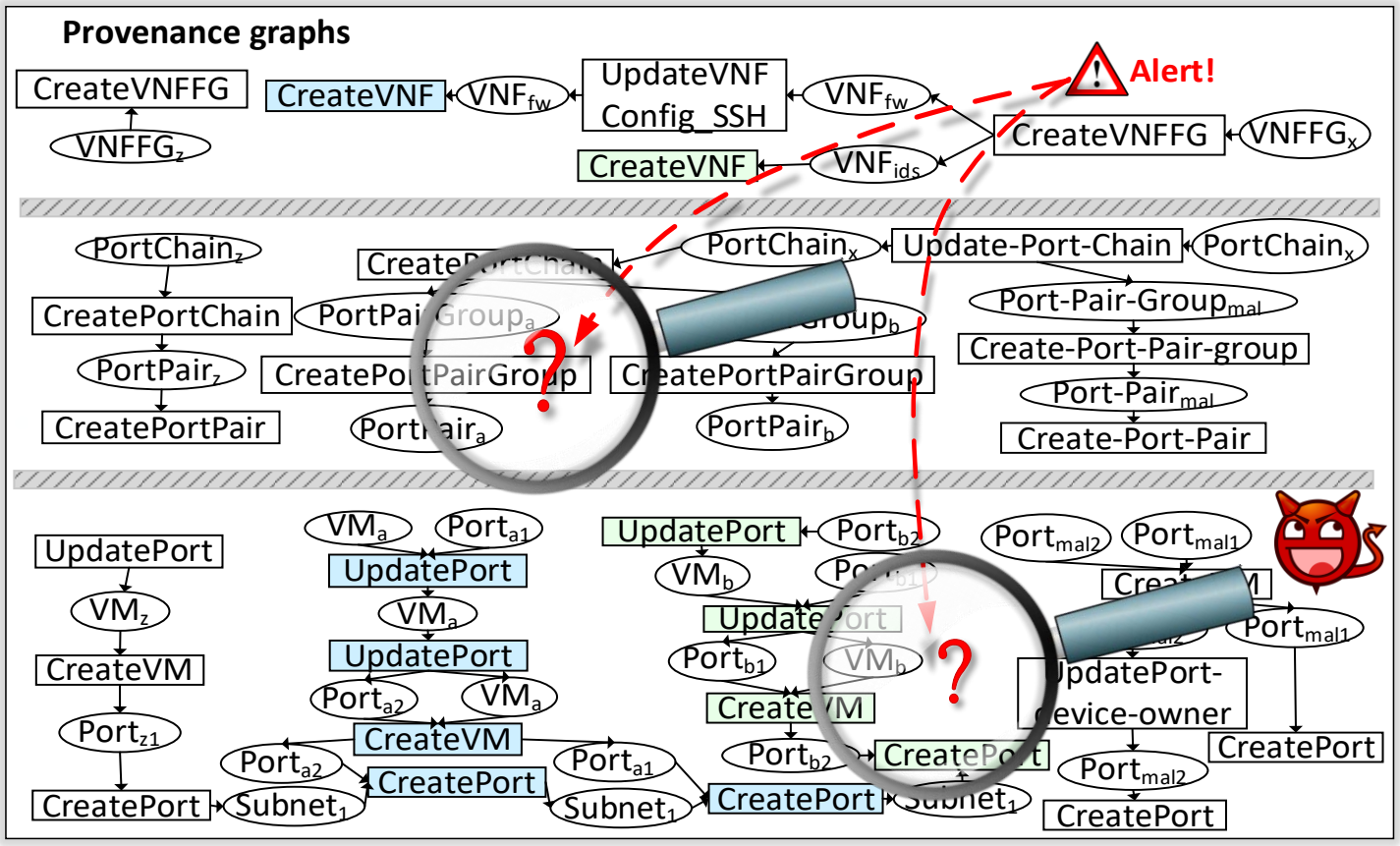


Challenges of Root Cause Analysis in NFV



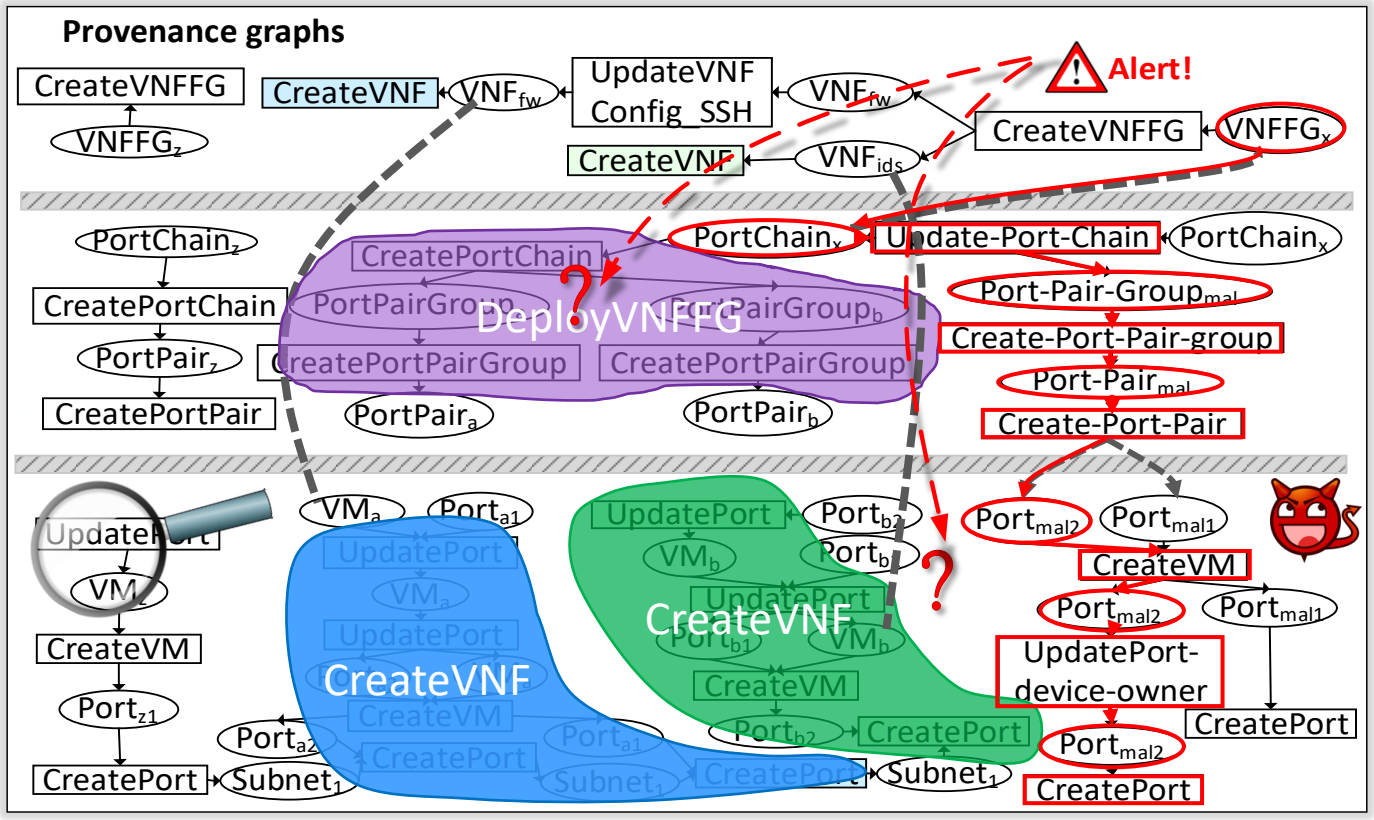
Challenges of Root Cause Analysis in NFV

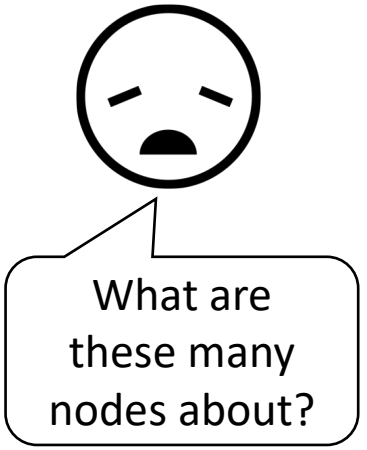
- No obvious link between the incident to its root cause at different levels of the NFV stack
- The impractically large and complex provenance graphs to interpret



ProvTalk: Goal

- Building a provenance model to handle the unique challenges brought by the multi-level aspect of NFV
- Taking advantage of the unique opportunities brought by such a multi-level aspect to increase the interpretability

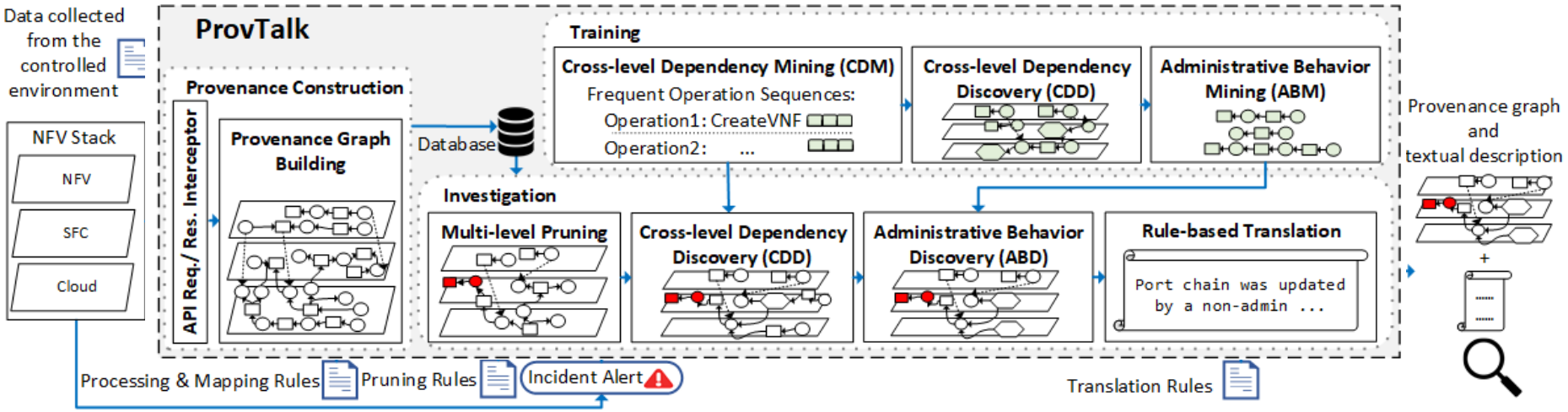




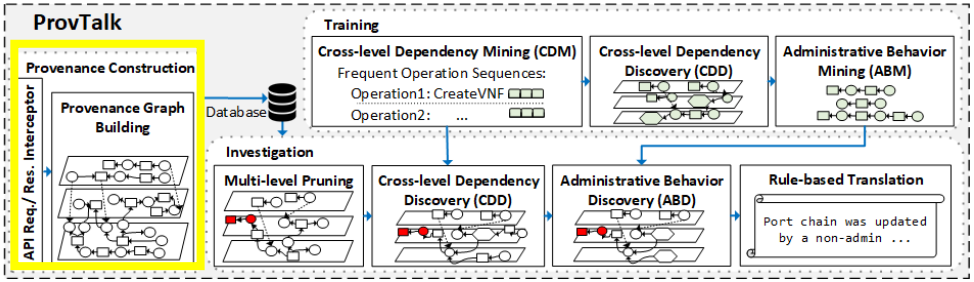
ProvTalk: Contributions

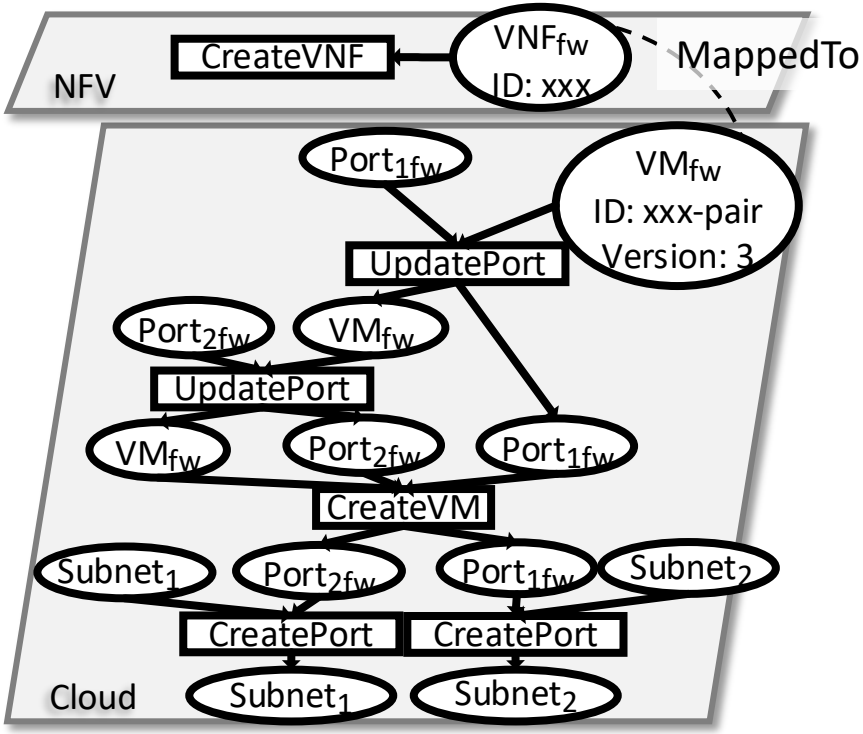
- The first multi-level provenance model for NFV capturing its multi-level nature with a mapping mechanism to pair the resources abstracted at different levels
- Increasing the interpretability of provenance analysis at NFV
  - A multi-level pruning mechanism
  - A mining-based multi-level aggregation technique
  - A rule-based translation mechanism
- Implementation and evaluation based on a real NFV testbed
  - Decreasing the size by 3.6 times via multi-level pruning
  - Decreasing the size by half via the aggregation

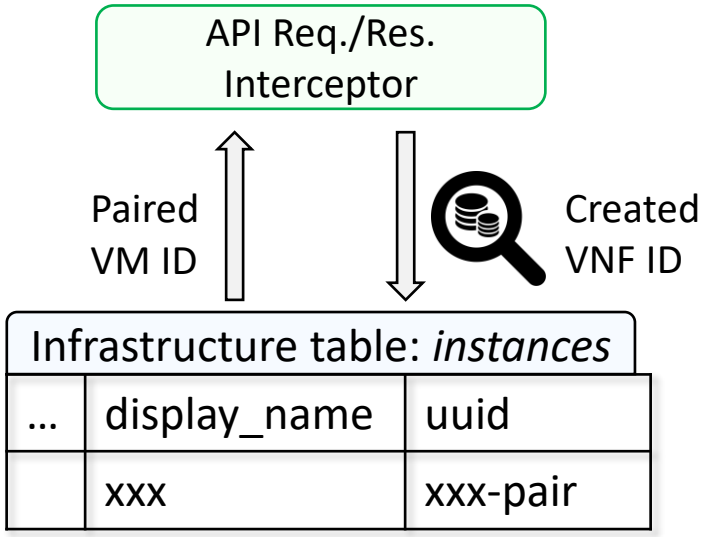
ProvTalk: Overview



1. Provenance Construction

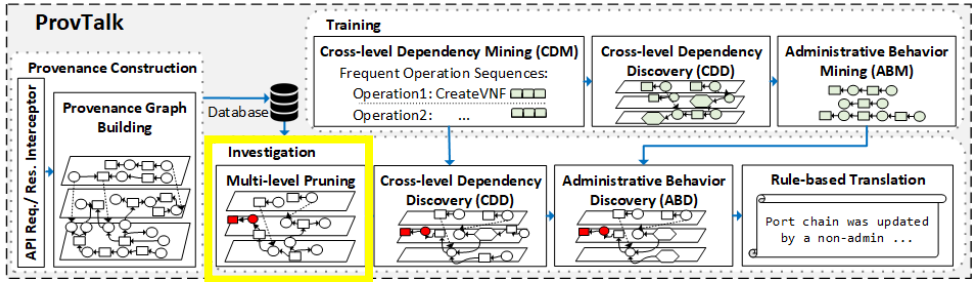


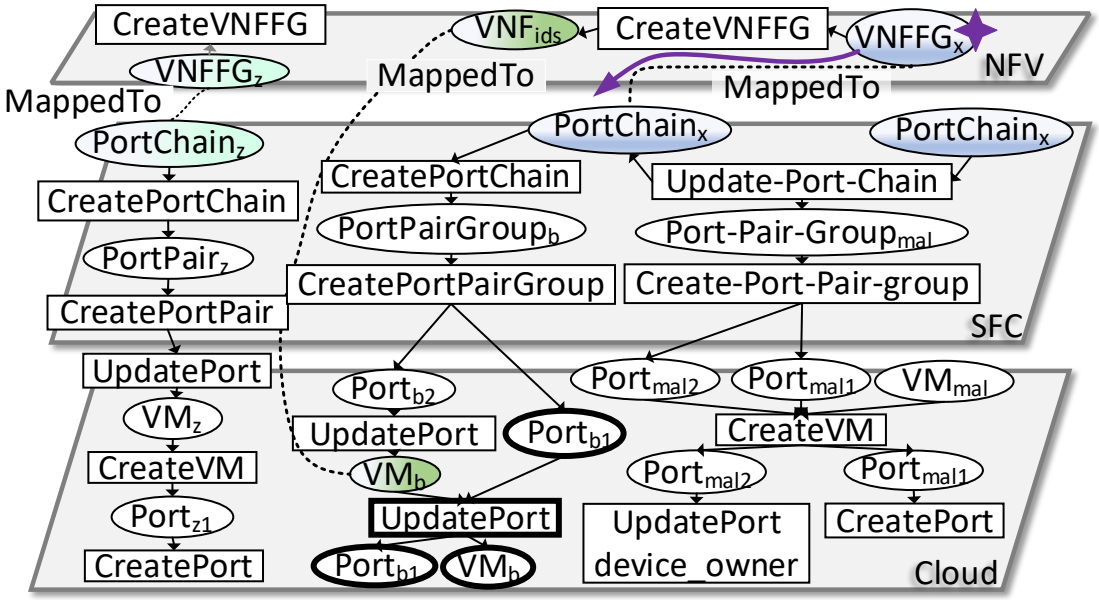




2. Multi-level Pruning

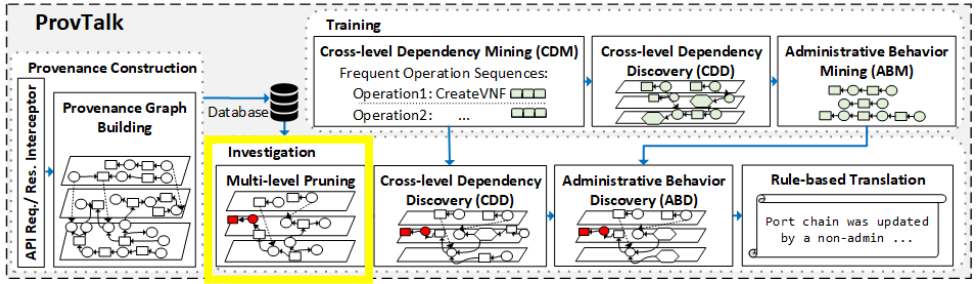
- **Objective**: Discarding irrelevant dependencies

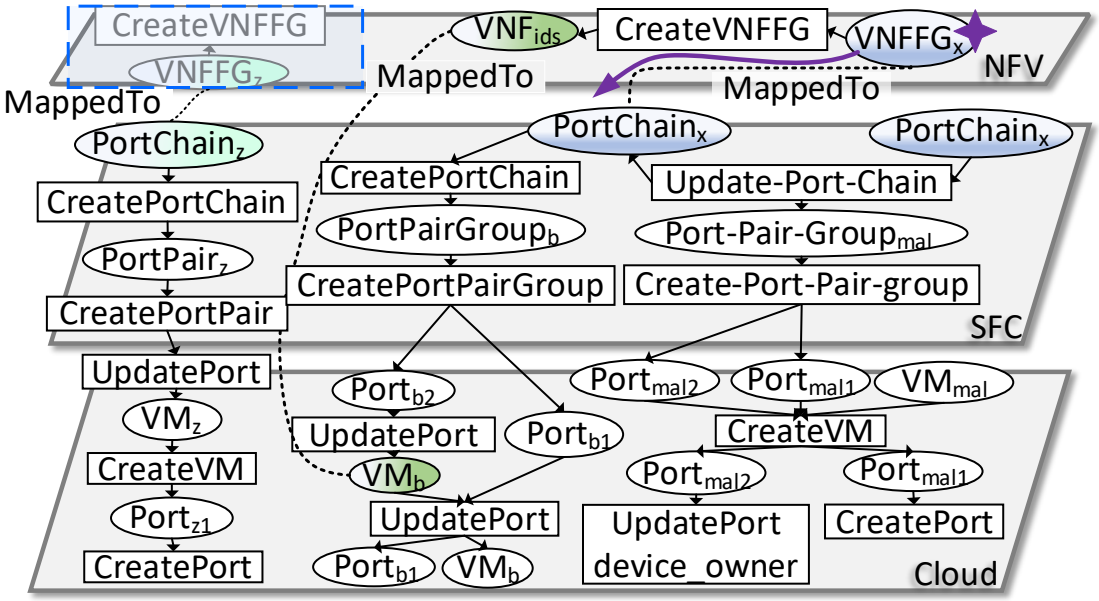




2. Multi-level Pruning

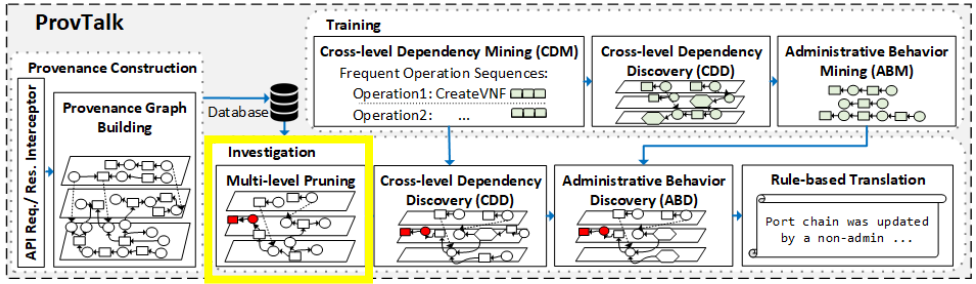
- **Objective**: Discarding irrelevant dependencies
  - Most existing techniques discard irrelevant nodes only at the same level as where the target incident is detected

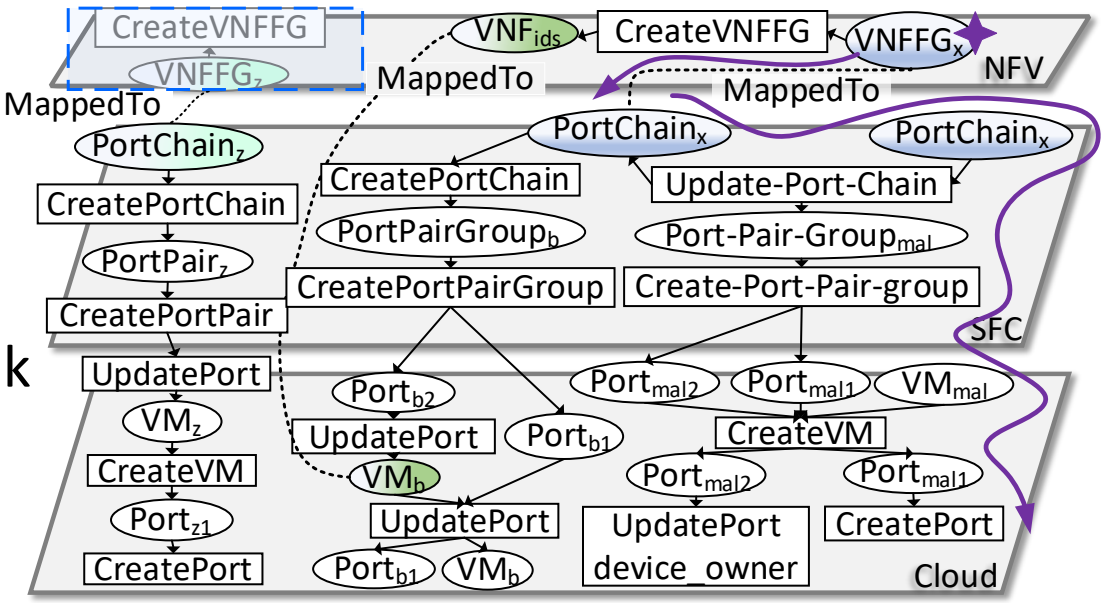




2. Multi-level Pruning

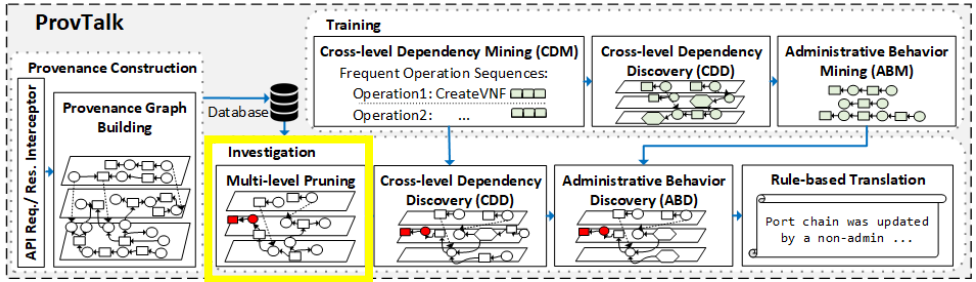
- **Objective**: Discarding irrelevant dependencies
  - Most existing techniques discard irrelevant nodes only at the same level as where the target incident is detected
  - Using cross-level dependencies, ProvTalk discards irrelevant nodes across different levels

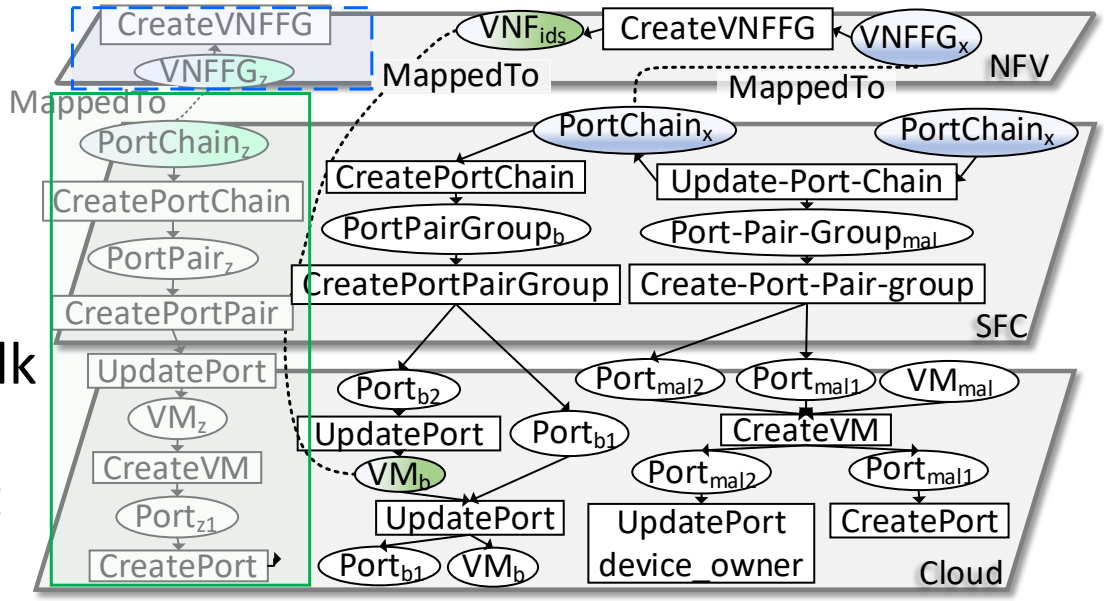




2. Multi-level Pruning

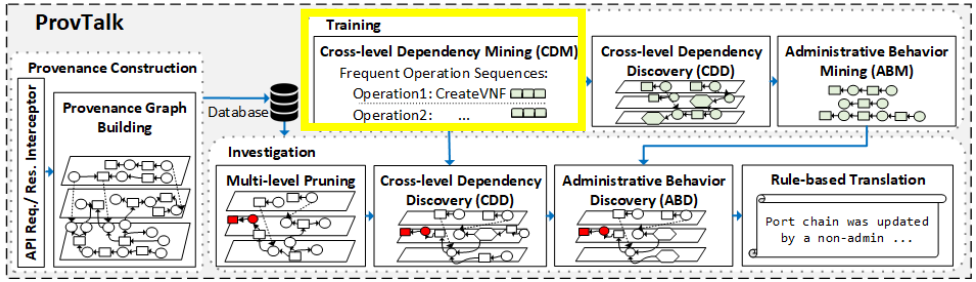
- **Objective**: Discarding irrelevant dependencies
  - Most existing techniques discard irrelevant nodes only at the same level as where the target incident is detected
  - Using cross-level dependencies, ProvTalk discards irrelevant nodes across different levels

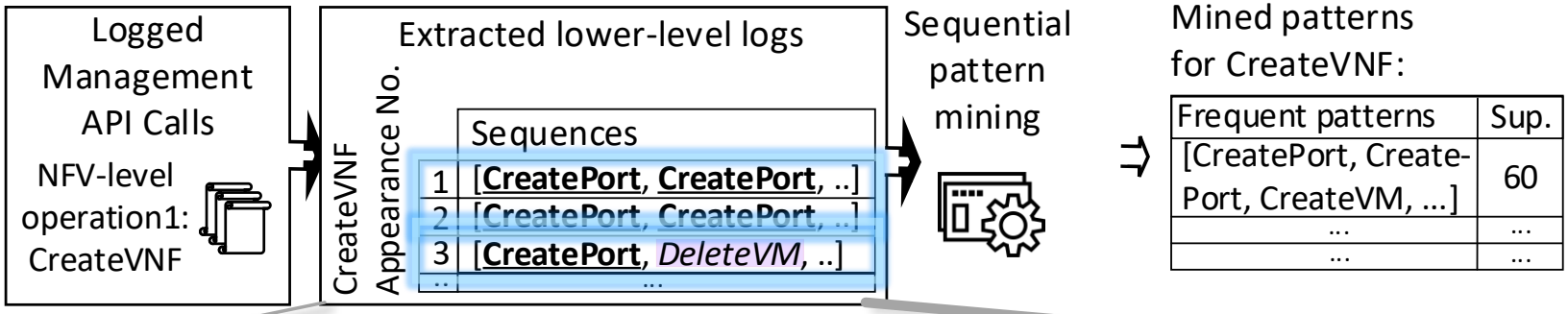


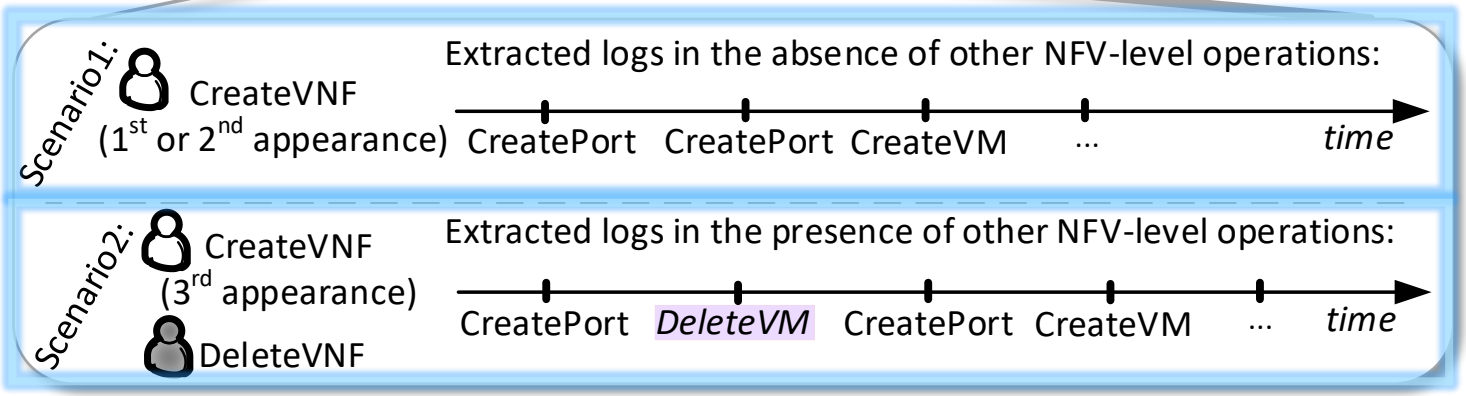


3(a). Cross-level Aggregation: Training – Sequential Pattern Mining

- Identifying the most frequent patterns of operations triggered after each NFV-level operation

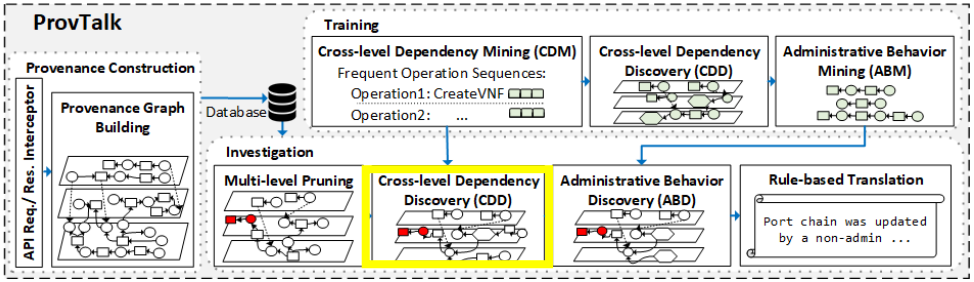


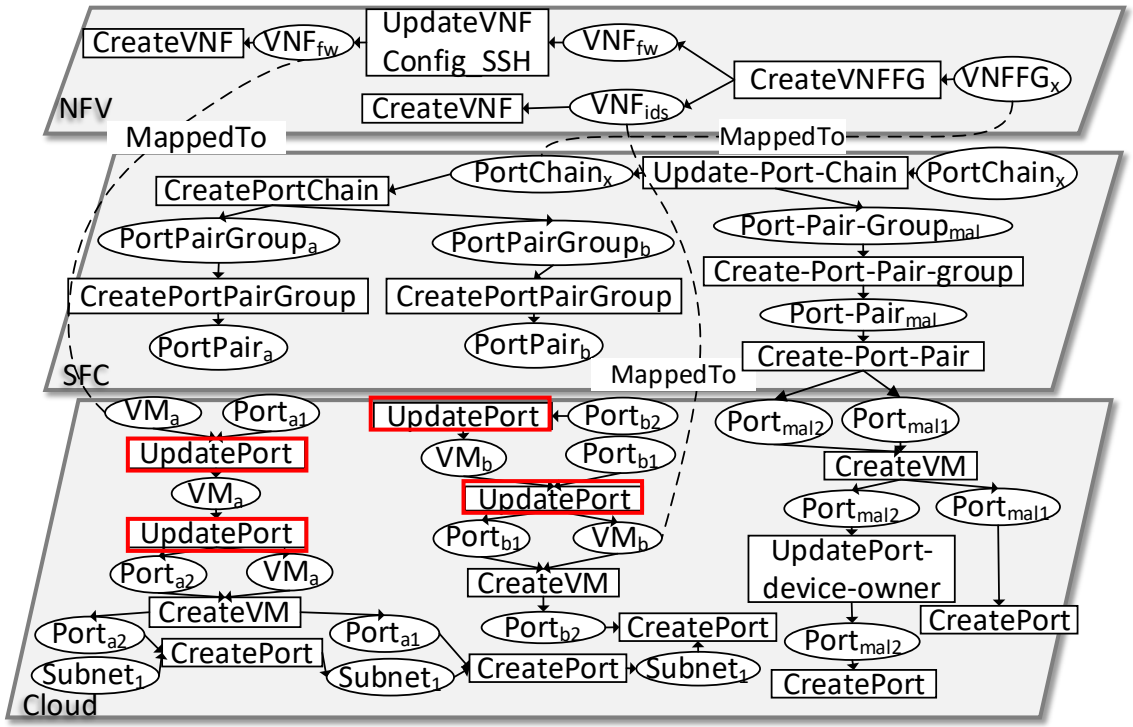




3(b). Cross-level Aggregation: Investigation

- Identifying and aggregating the nodes corresponding to the same NFV-level operation



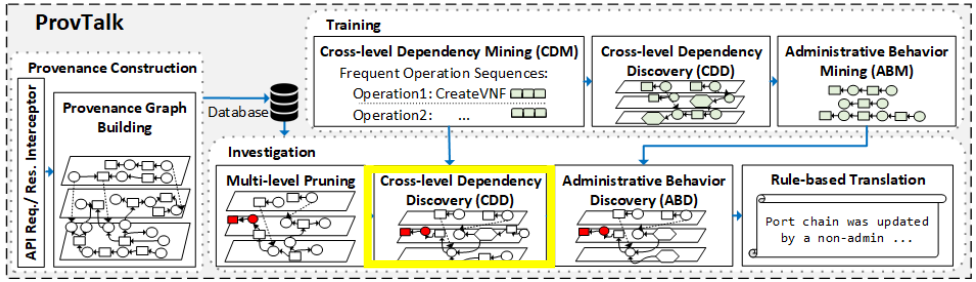


CreateVNF => “CreatePort, CreatePort, CreateVM, UpdatePort, UpdatePort”

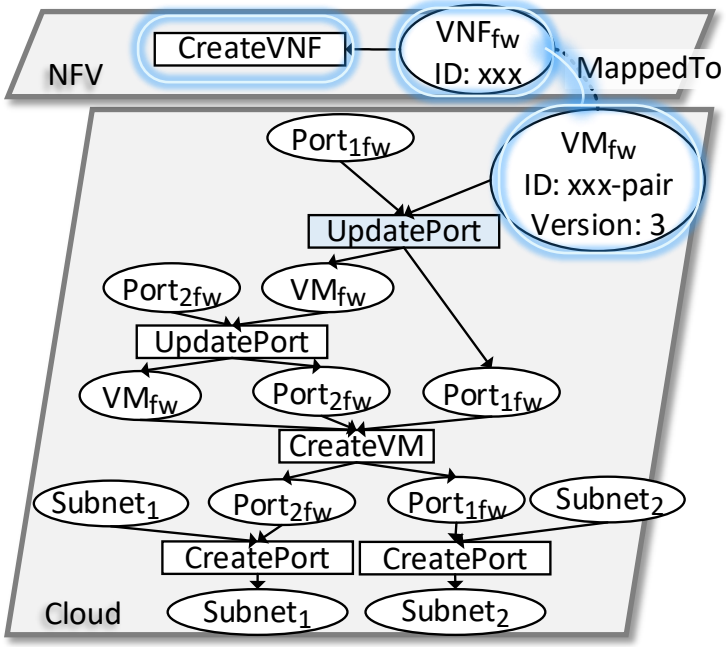
Which nodes correspond to the mined operations?

3(b). Cross-level Aggregation: Investigation

- Identifying and aggregating the nodes corresponding to the same NFV-level operation

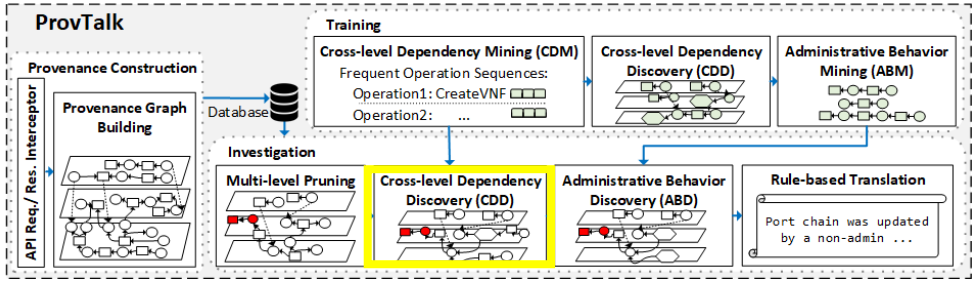


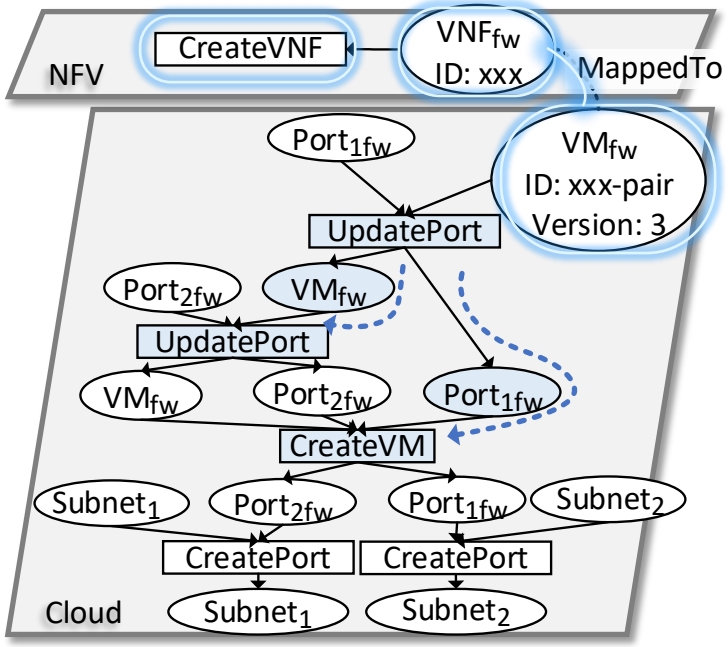
CreateVNF => "CreatePort, CreatePort, CreateVM, UpdatePort, UpdatePort"



3(b). Cross-level Aggregation: Investigation

- Identifying and aggregating the nodes corresponding to the same NFV-level operation

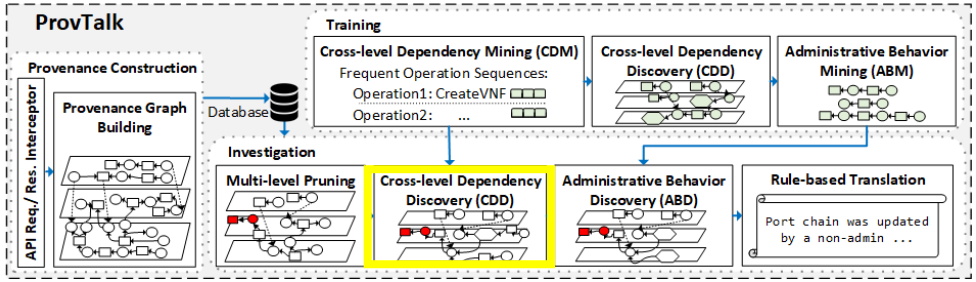




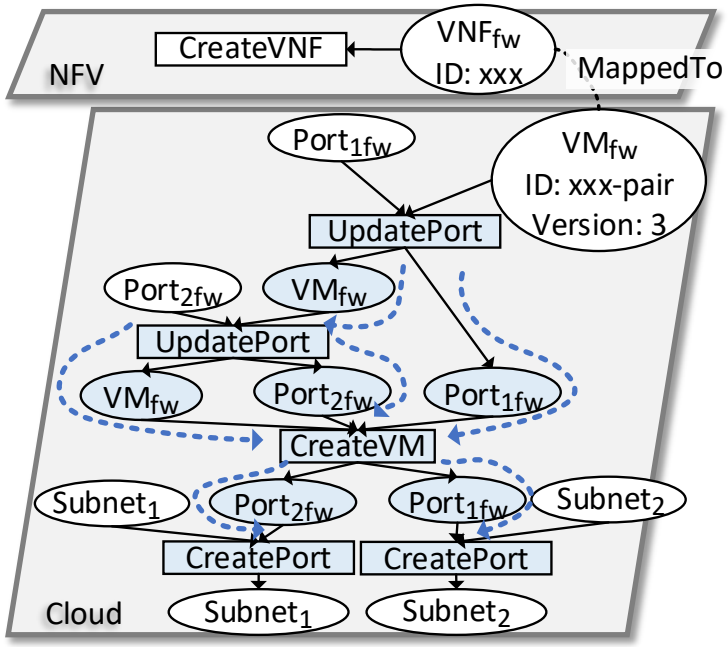
CreateVNF => “CreatePort, CreatePort, CreateVM, UpdatePort, UpdatePort”

3(b). Cross-level Aggregation: Investigation

- Identifying and aggregating the nodes corresponding to the same NFV-level operation

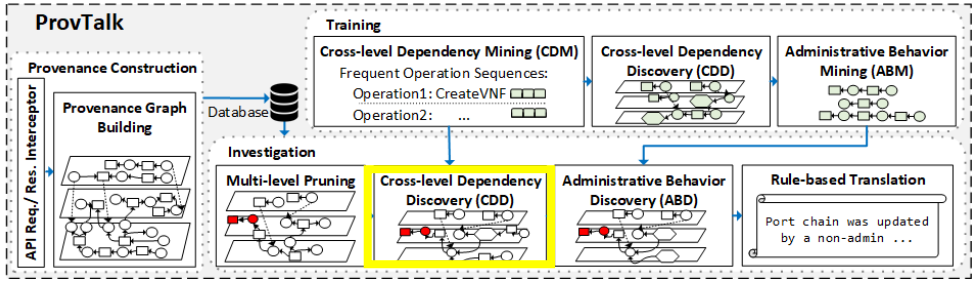


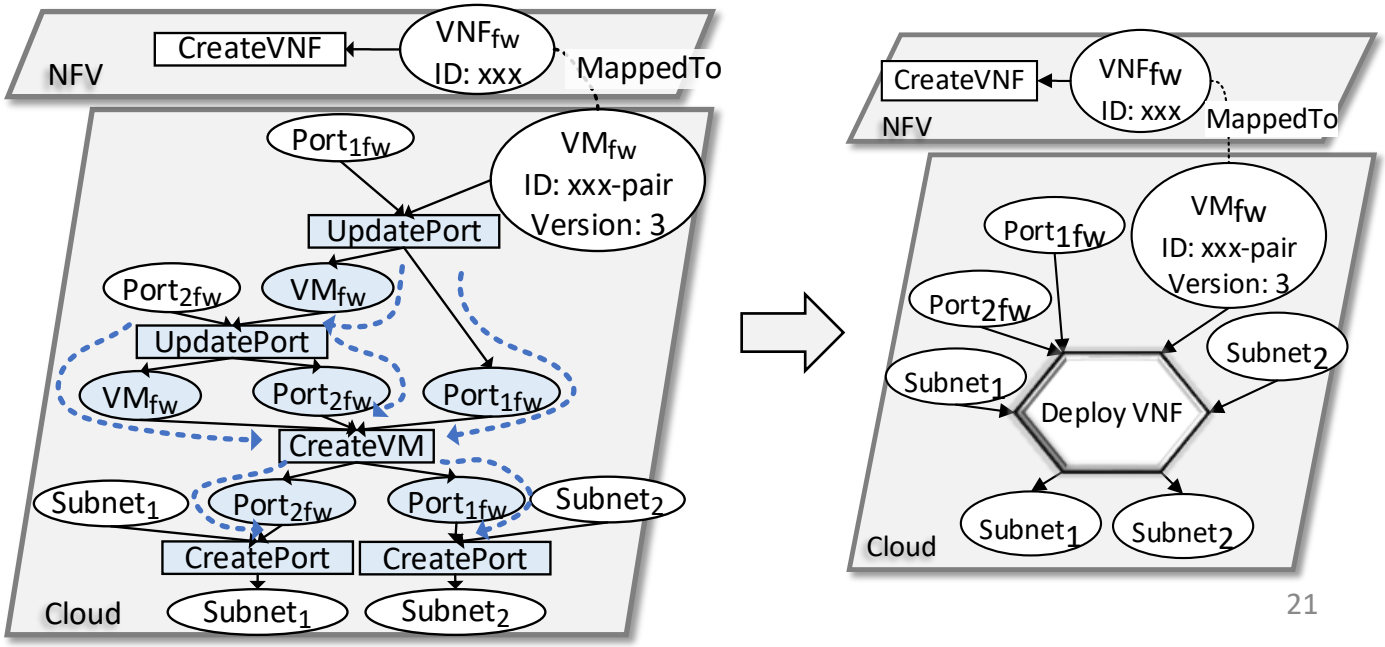
CreateVNF => "CreatePort, CreatePort, CreateVM, UpdatePort, UpdatePort"



3(b). Cross-level Aggregation: Investigation

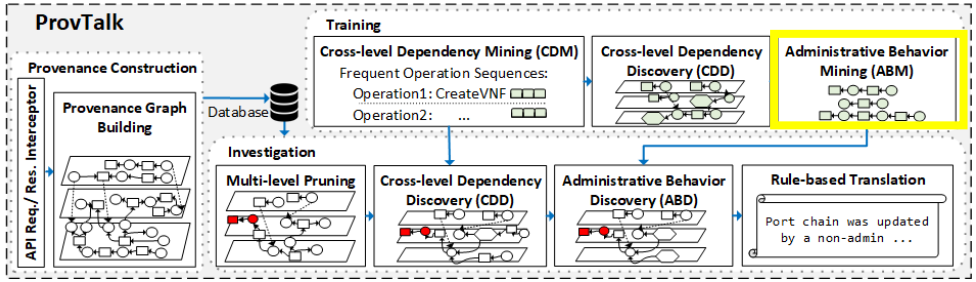
- Identifying and aggregating the nodes corresponding to the same NFV-level operation

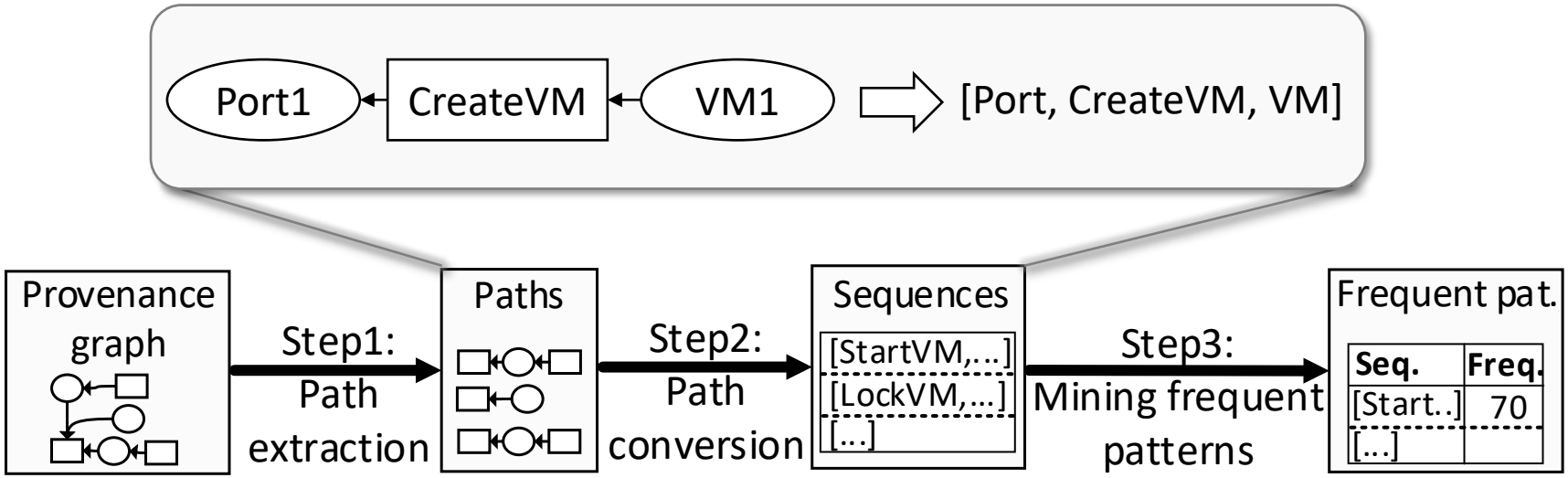




4(a). Administrative Behavior Aggregation: Training – Sequential Pattern Mining

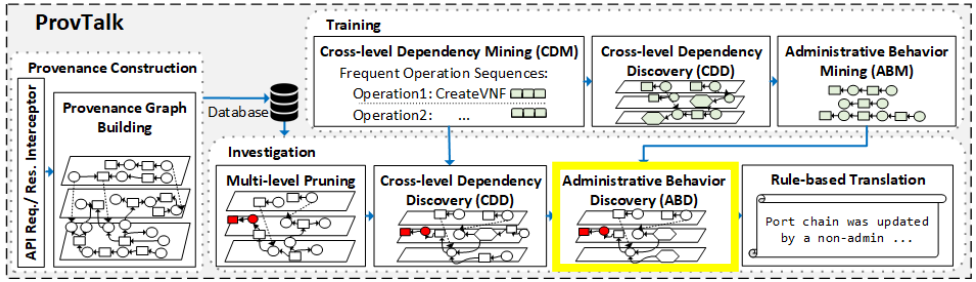
- Building a model of routine administrative behavior repetitively appearing in the provenance graph

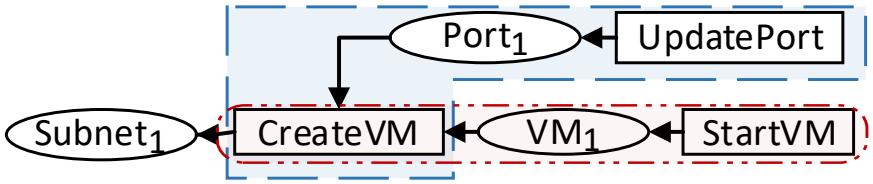




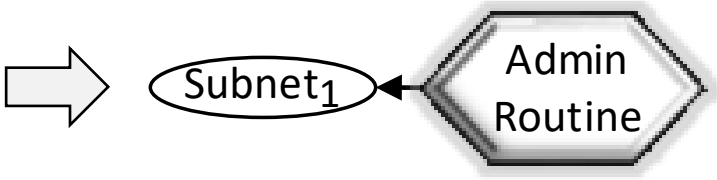
4(b). Administrative Behavior Aggregation: Investigation

- Aggregating the paths matching the mined sequences

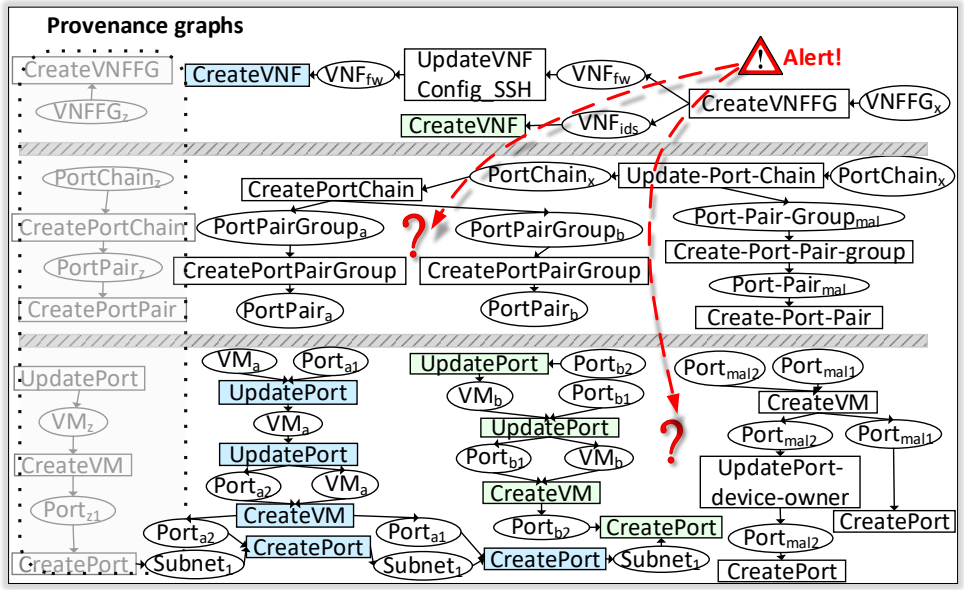


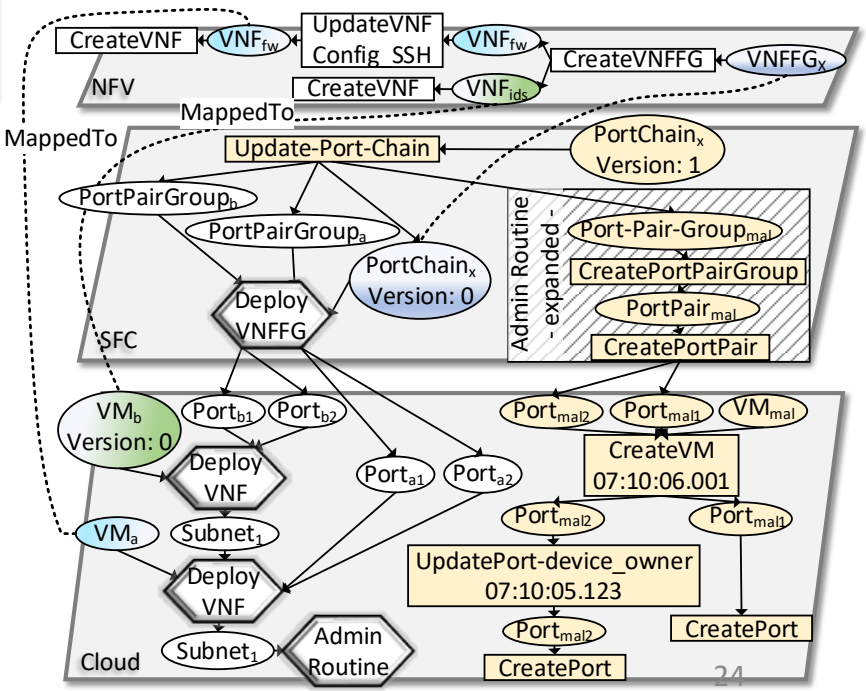


Sequence$_k$ = [CreateVM, Port, UpdatePort]
Sequence$_j$ = [CreateVM, VM, StartVM]



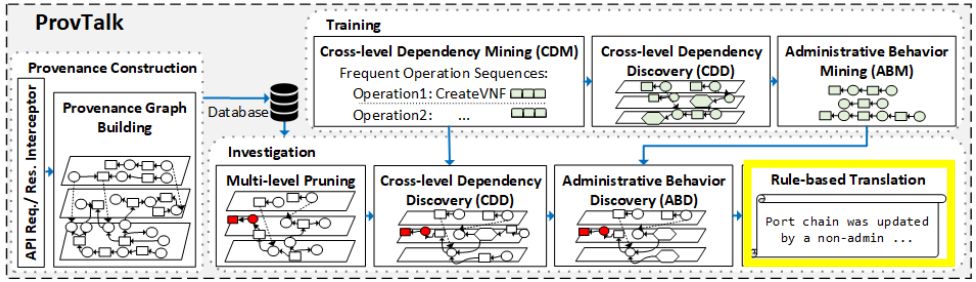
Result of ProvTalk





5. Rule-based Translation

- Translating the captured information into a summary by pre-specified rules

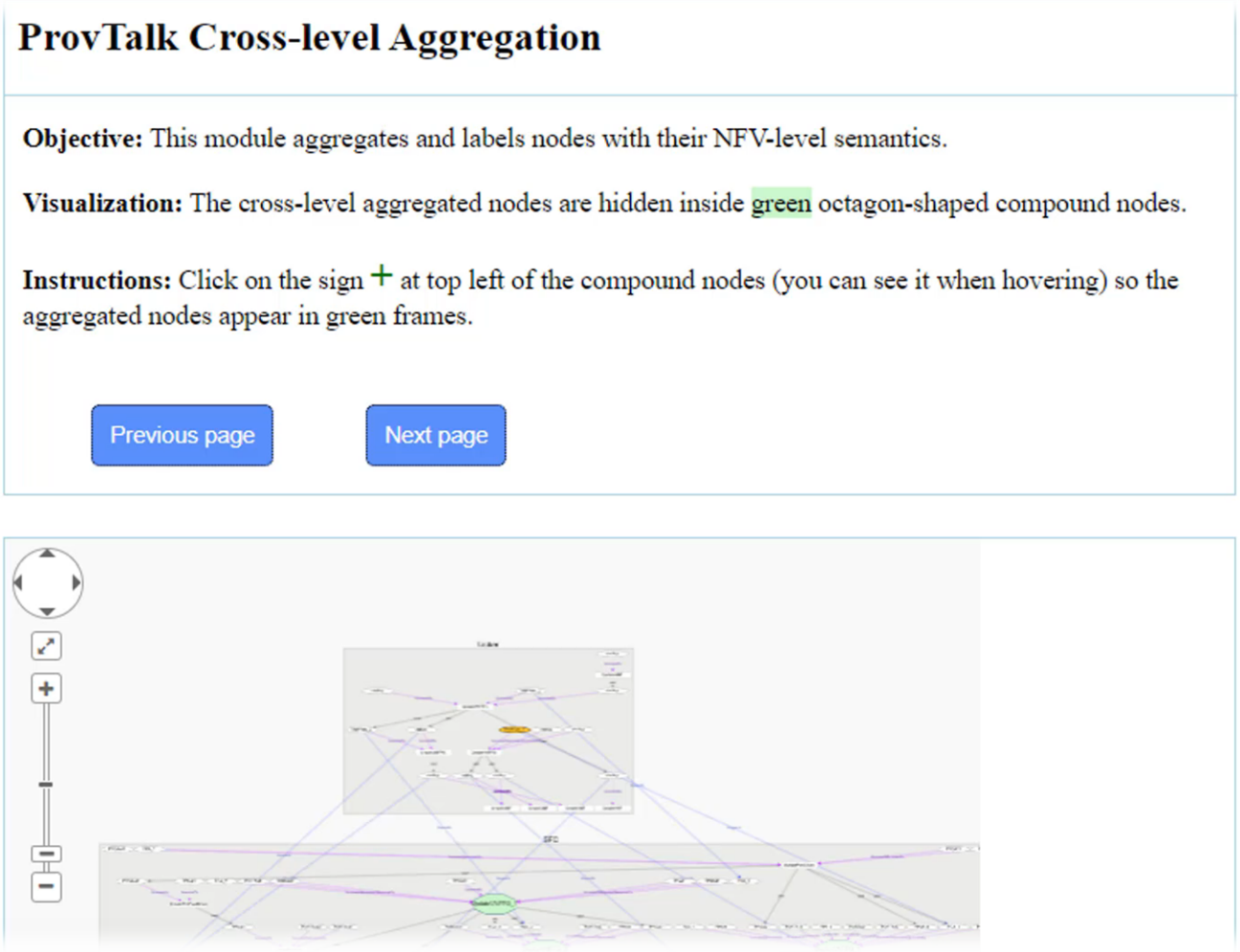


By the detection time 21-01-06 11:44:07.769, there are 6 operations performed in the specified time interval 0:00:15.454 hours corresponding to the target entity VNFFGx created by admin user using 2 VNFs.

User 12ddf created PORTmal2 at 07:10:03.403. He updated device_owner Portmal2 at 07:10:05.123. And created VMmal using that port(s) after 0:00:0.878 hours. Then, he created PortPairmal using that port(s) after 0:00:03.202 hours. And created PortPairGroupmal using that PortPair(s). He updated PortChainx, using that PortPairGroup(s) after 0:00:02.239 Hours.

More details can be found in the provenance graph following this node path [215 - 211 - 208 - 207 - 204 - 202]. Node(s) (UpdatePortChain-ID: 215) worth a closer look: nonAdmin user (ID: 12ddf) modifies admin (ID: 53atb) modified resource(s).

Glimpse of ProvTalk Dashboard



Implementation

- Implemented on OpenStack (*Rocky*) and *Tacker*
- *Neo4j* as the graph database and *Cypher* language to query
- *Py2neo* library to translate python queries into Cypher
- *BIDE* algorithm for sequential pattern mining
- *Cytoscape.js* graph theory (network) library for visualization and analysis of multi-level provenance graph
- Deployed as Python middlewares on Tacker and OpenStack services

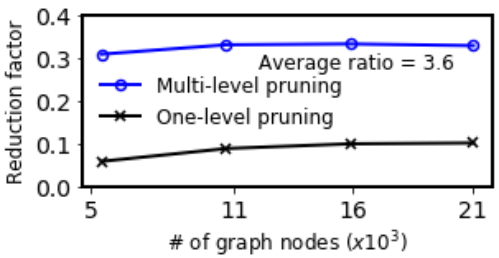
Evaluation – Effectiveness

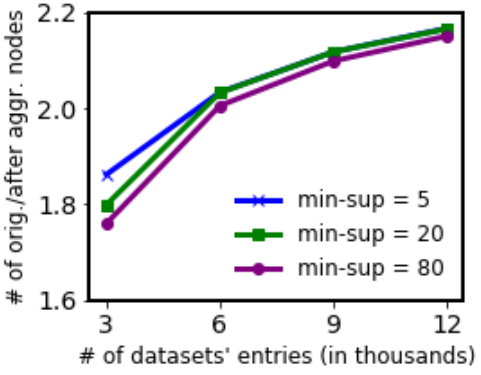
- For all the 10 attacks, we successfully trace back to the root cause of the reported incident using ProvTalk

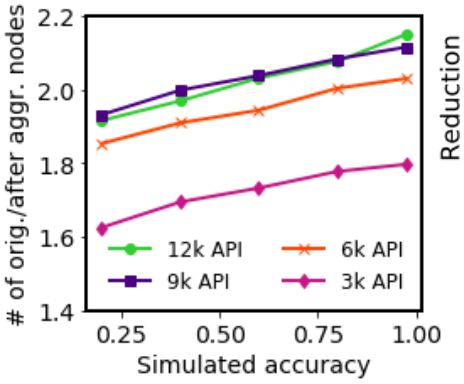
Root Cause	Detected Incident	Most Relevant Management Operation Types	Vulnerability
Stealthy node injection into a VNFFG	Unauthorized Access	CreatePortPair, CreatePortPairGroup, UpdatePortChain	CVE-2017-2673
Bypassing anti-spoofing rules in network	Unauthorized Access	CreateVNFFG, CreatePort, CreateVM, UpdatePort	CVE-2015-5240
Firewall VNF misconfiguration	CPU DoS	CreateVNFFG, UpdateVNFFG, UpdateVNF	CVE-2017-7400
Malformed security group rule addition	Host Unavailability	CreateSecurityGroup, CreateSecurityGroupRule, CreateVM	CVE-2019-9735
Overlapping security group rule addition	Host Unavailability	CreateSecurityGroup, CreateSecurityGroupRule, CreateVM	CVE-2019-10876
Update of security group is not applied	Data Leakage	AddSecurityGroup, StartVM, DeleteSecurityGroupRule	CVE-2015-7713
Neutron proper authorization failure	Port Scanning	CreateRouter, CreatePort, CreateVM	CVE-2014-0056
Wrong VLAN ID	Data Leakage	CreateNetwork, UpdateNetwork	Not specified
Failing to delete VMs in resize state	Disk DoS	CreateVM, ResizeVM, DeleteVM	CVE-2016-7498
Excessive VM creation on the same host	Disk DoS	Create-VM	Not specified

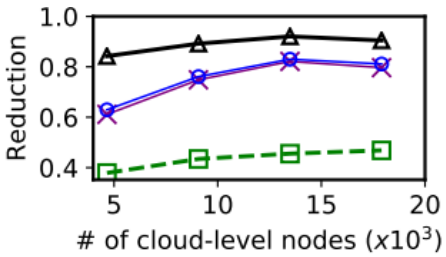
Evaluation – Size Reduction

- Multi-level pruning decreases the size by 3.6 times
- Multi-level aggregation decreases the size by half
- Effect of accuracy on the performance of aggregation:
  - 80% error rate causes only 0.4 less reduction power
- Twice the reduction factor of the existing work



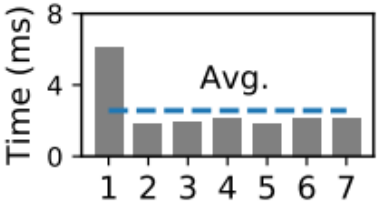




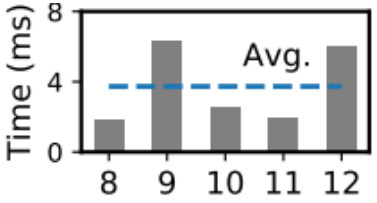


Evaluation – Efficiency

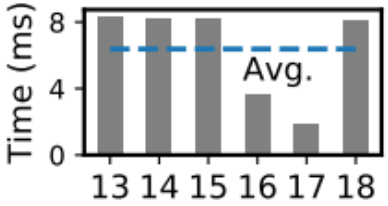
- Incurs a negligible overhead of around 0.04% additional delay to NFV management operations



Cloud-level operations

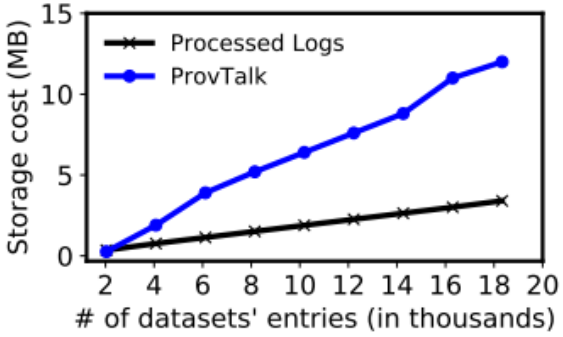


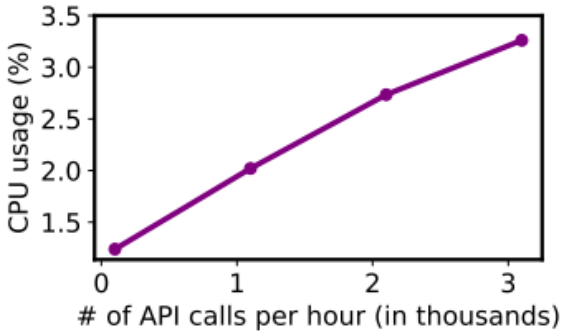
SFC-level operations



NFV-level operations

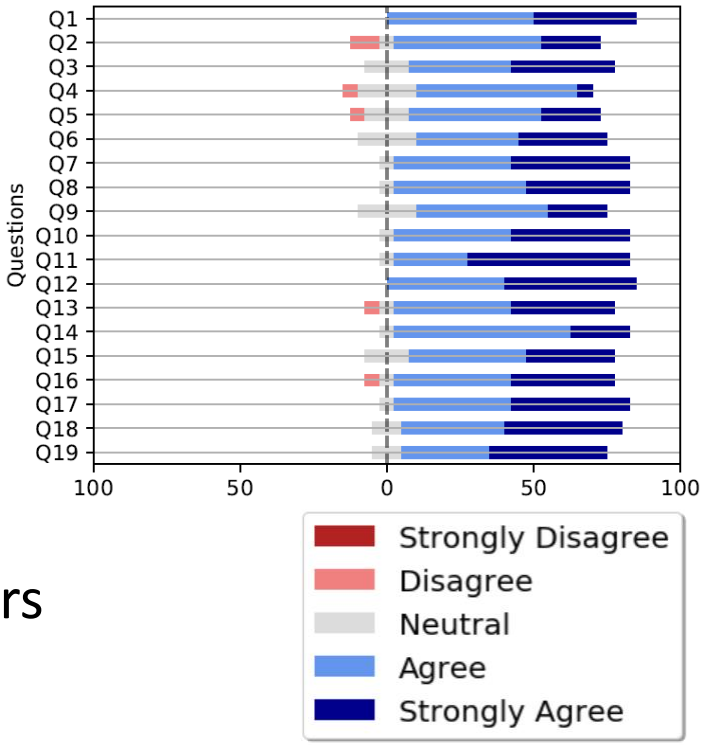
- Only 10MB storage is required for the dataset of 20,000 API calls
- The CPU usage is under 3.5% for the rate of 3,000 API calls/hour





Evaluation – User Study

- Participants
  - From a telecommunication industrial organization
  - Graduate students working in cybersecurity
- 19 questions
  - 3 multi-level provenance graph
  - 2 pruning
  - 5 aggregation
  - 3 rule-based translation
  - 6 interactive features
- Our user studies show that our approach significantly eases the analysis task of users



Concluding Remarks

Summary

- Proposed a multi-level provenance model for NFV with a mapping mechanism to pair the resources abstracted at different levels
- Designed an approach to increase the interpretability of provenance analysis at NFV
  - A multi-level pruning mechanism
  - A mining-based aggregation technique
  - A rule-based translation
- Implemented and evaluated based on real NFV lab setup

Future Directions

- Comparing with more complex learning techniques
- Integrating with OS-level provenance graphs

Thanks & Questions

Azadeh Tabiban: a_tabiba@encs.concordia.ca

NSERC/Ericsson Industrial Research Chair in SDN/NFV Security: https://arc.encs.concordia.ca/