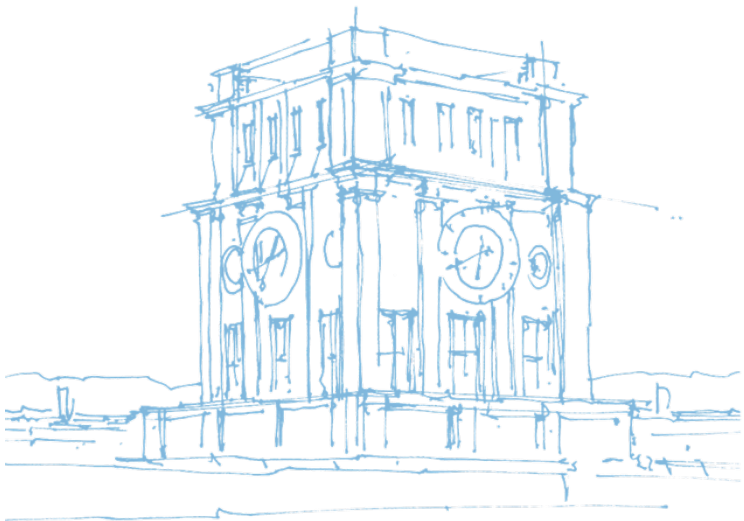
Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Achieving Reproducible Network Environments with INSALATA

Nadine Herold, Matthias Wachs, Marko Dorfhuber, Cristoph Rudolf, Stefan Liebald, Georg Carle

Tuesday 11th July, 2017

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich



Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Motivation

Why do we want reproducible network environments?

Motivation

Why do we want reproducible network environments?

- Requirement for reproducible experiments
  - e.g. from other testbeds

Motivation

Why do we want reproducible network environments?

- Requirement for reproducible experiments
  - e.g. from other testbeds
- Test changes before deployment in operational network
  - Routing
  - Software updates
  - Firewall rules
  - Configuration changes
  - ...

Our Solution: INSALATA

Manual replication:

- Error prone, time consuming
- Keeping it up-to-date is hard
- Multitude of Tools/Software used manually

Our Solution: INSALATA

Manual replication:

- Error prone, time consuming
- Keeping it up-to-date is hard
- Multitude of Tools/Software used manually

The goal:

- Automate complete replication process
- Scan an environment and deploy it on a testbed

Our Solution: INSALATA

Manual replication:

- Error prone, time consuming
- Keeping it up-to-date is hard
- Multitude of Tools/Software used manually

The goal:

- Automate complete replication process
- Scan an environment and deploy it on a testbed

Our Framework: IT NetworkS AnaLysis And deploymenT Application (INSALATA)

- Network information model
- Information collection component
- Infrastructure deployment component

Requirements	IF-MAP [1, 20, 21]	IDS Ontology [22]	INDL [18, 10, 8]	NML [24, 23]
R1: Network components	✓	✓	✓	✓
R2: Connections	✓	✓	✓	✓
R3: Addressing	✓	✗	✗	✗
R4: Reachability of components	✗	✗	✗	✗
R5: Network services	✓	✓	✓	✗
R6: Hardware information	✓	✗	✓	✗
R7: Extensibility of information elements	✓	✓	✓	✓
R8: History	✗	✗	✗	✓

✓ fulfilled, ✗ not fulfilled

Requirements and Related Work

Information Collection Component
Requirements and Related Work

Requirements	IO-Framework [4, 12]	cNIS [9]	MonALISA [6, 5]	PerfSONAR [19]	OpenVAS[16]	Nmap [13, 14]
R1: Configurability	✗	✓	✓	✓	✓	✓
R2: Collection of required topoloy information	✓	✗	✓	✗	✗	✗
R3: Extensible modules	✓	✓	✓	✓	✓	✓
R4: Periodical information collection	✗	✓	✓	✓	✓	✗
R5: Adaptable intervals	✗	✗	✗	✓	✓	✗
R6: Continuous monitoring	✗	✓	✗	✗	✗	✗
R7: Multiple environments	✗	✗	✓	✓	✓	✗
R8: Extensible export	✗	✓	✗	✗	✗	✗

✓ fulfilled, ✗ not fulfilled

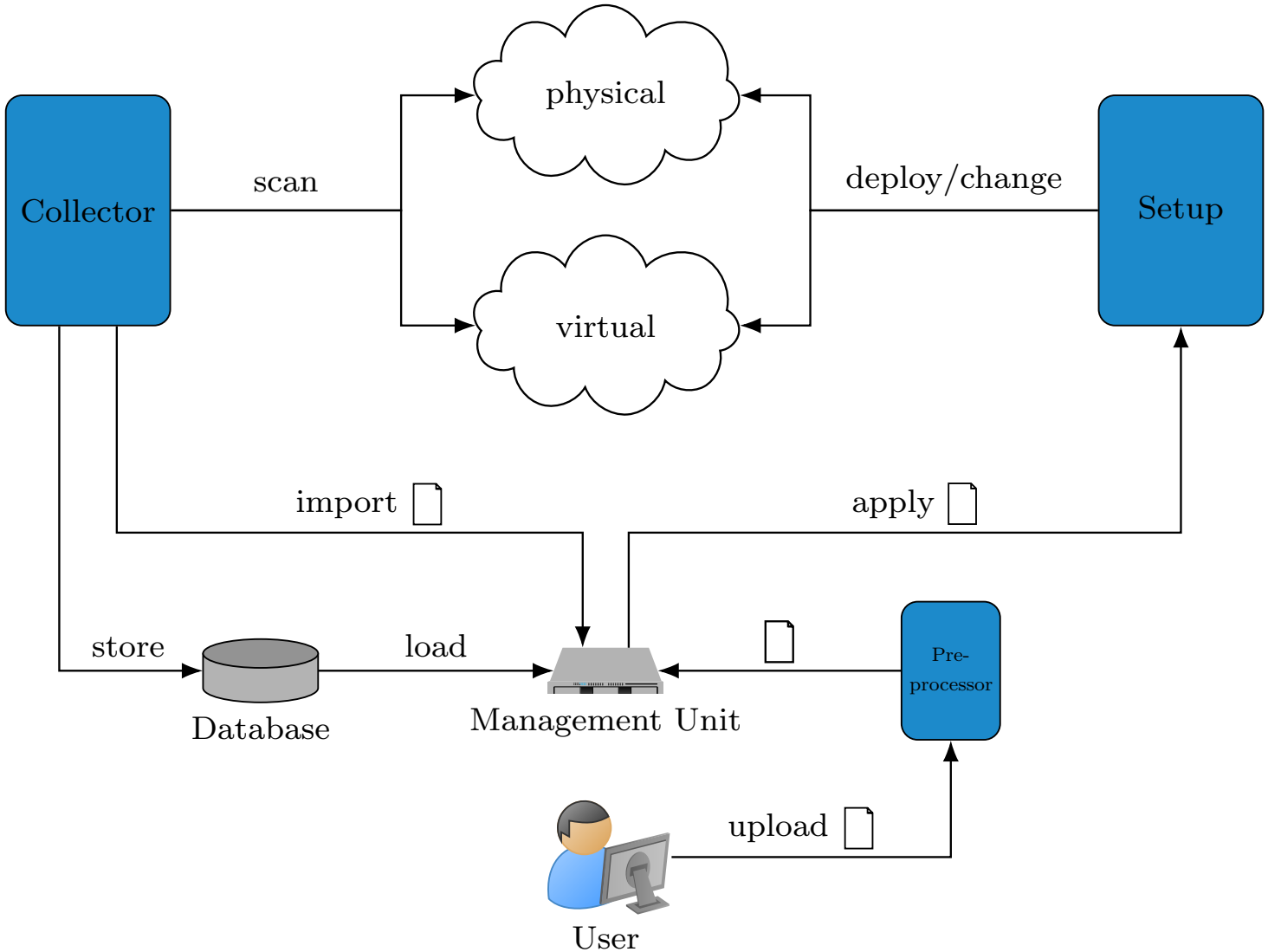
Requirements and Related Work

Infrastructure Deployment Component
Requirements and Related Work

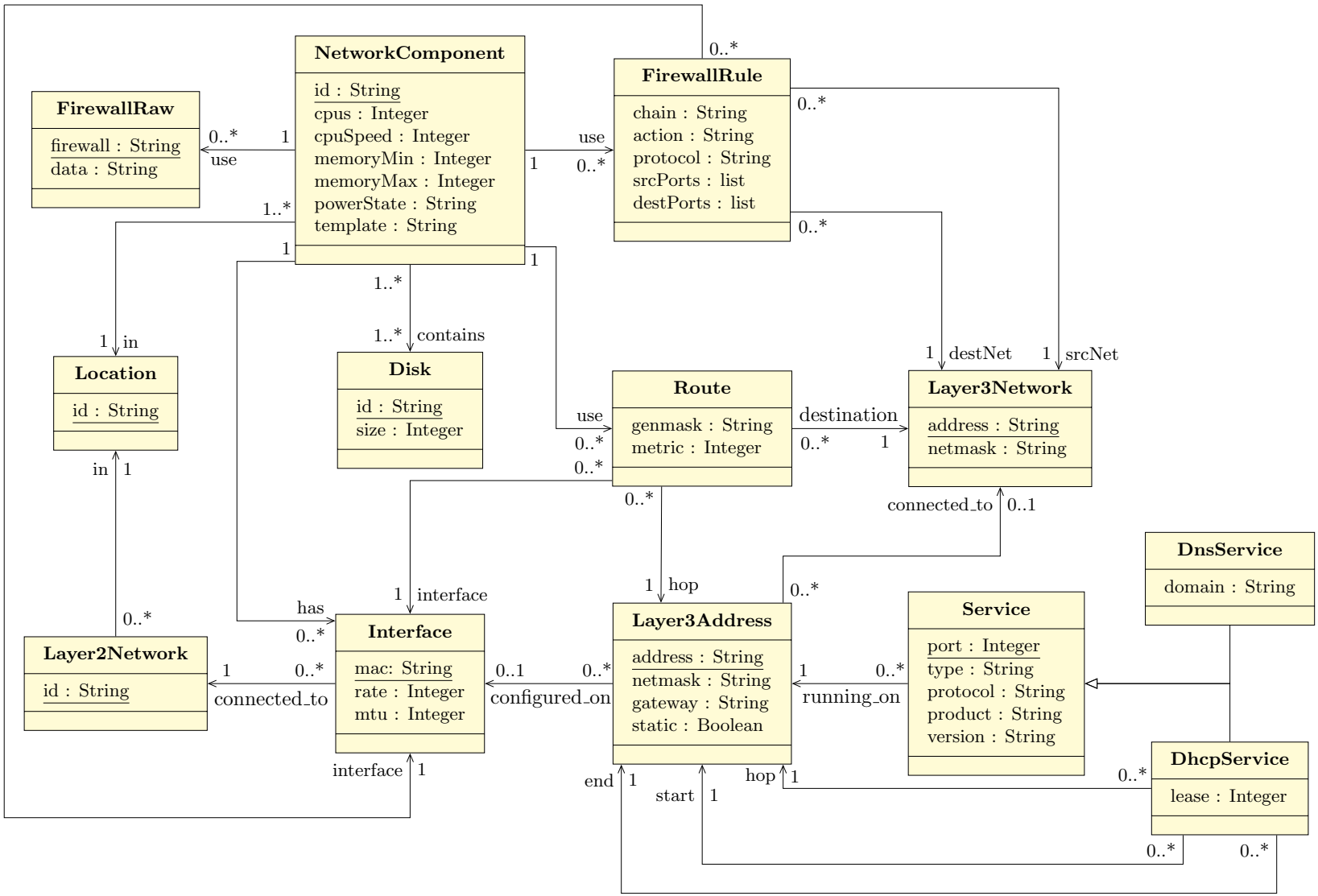
Requirements	LaasNetExp [17]	vBET [11]	Baltikum Testbed	NEPTUNE [3]	Algorizmi [2]	Emulab[25]
R1: Basic network components	✗	✓	✗	✗	✗	✓
R2: Description language for configuration	?	✓	✓	✓	✓	✓
R3: Virtual and physical components	✗	✗	✓	✗	✗	✓
R4: Software independent design	?	?	✗	✓	✗	✗
R5: Public availability	✗	✗	✓	✓	✓	✓
R6: Separated and persistent components	✓	✗	✓	✓	✓	✓
R7: Multiple users per testbed	✓	✗	✗	✗	✓	✓
R8: Common usage of central components	✓	✓	✗	✓	✗	✗
R9: Incremental Changes	✗	✗	✗	✗	✗	✗

✓ fulfilled, ✗ not fulfilled, **?** insufficient information

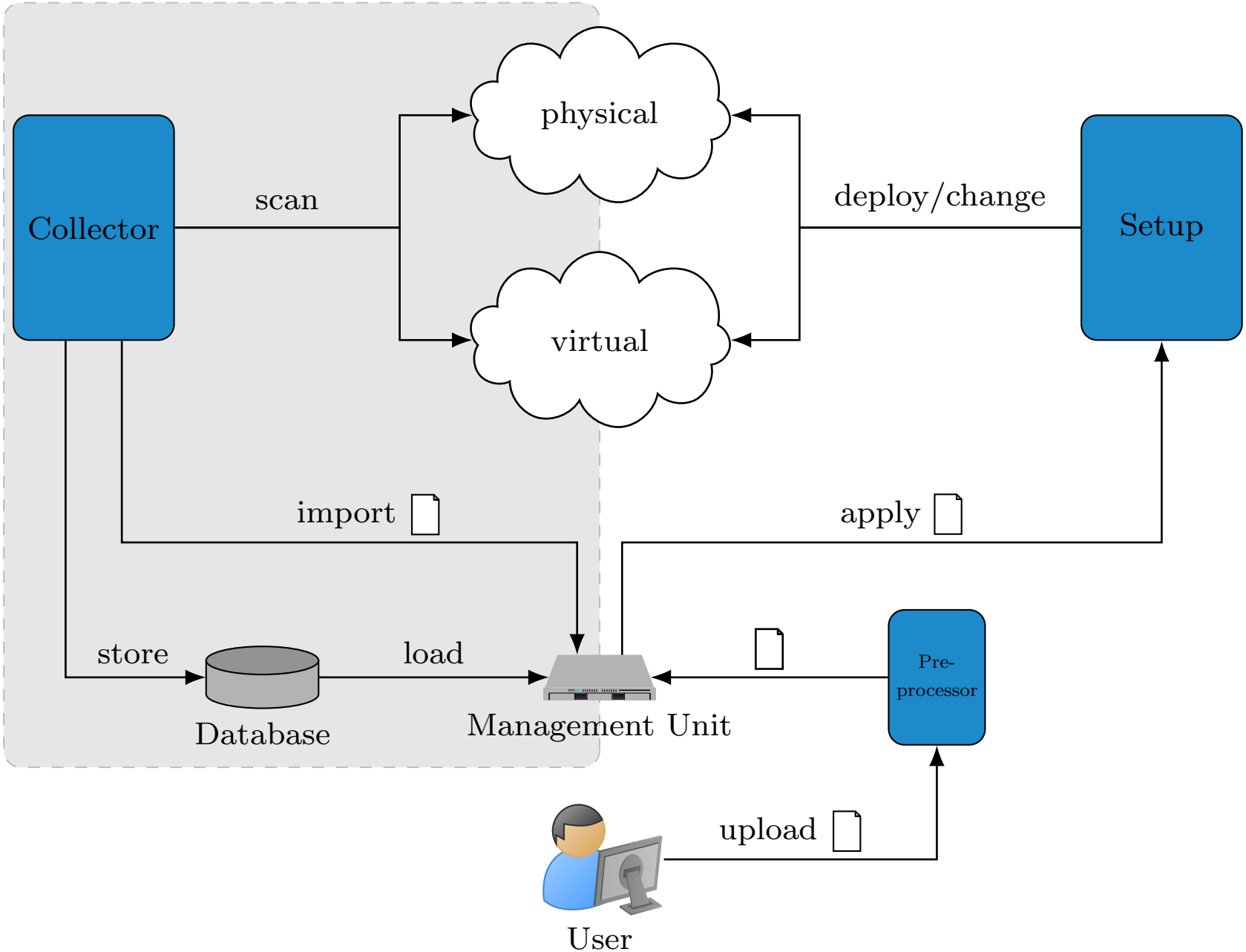
Architecture: Overview



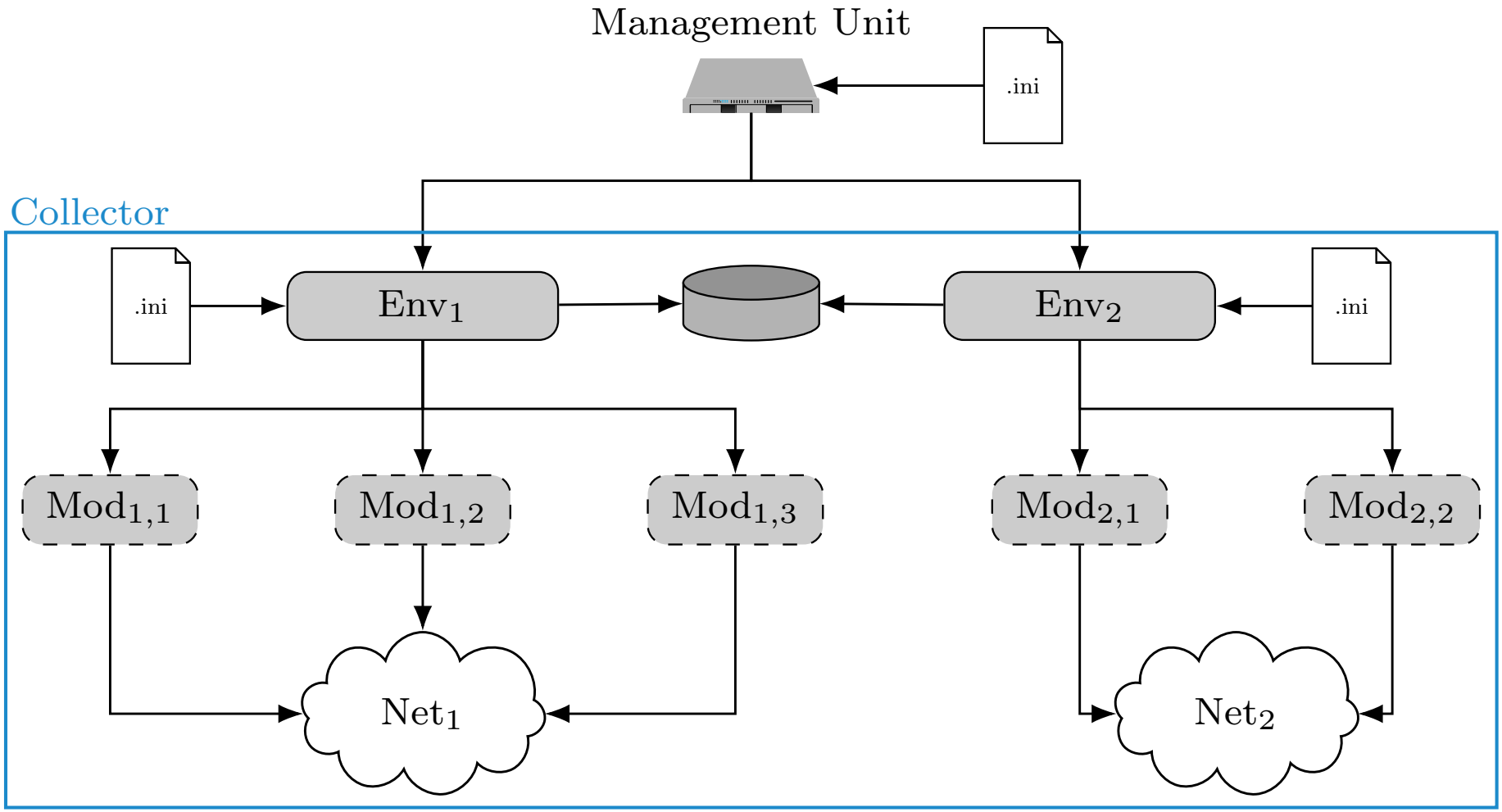
Architecture: Information Model



Architecture: Information Collection Component



Architecture: Information Collection Component

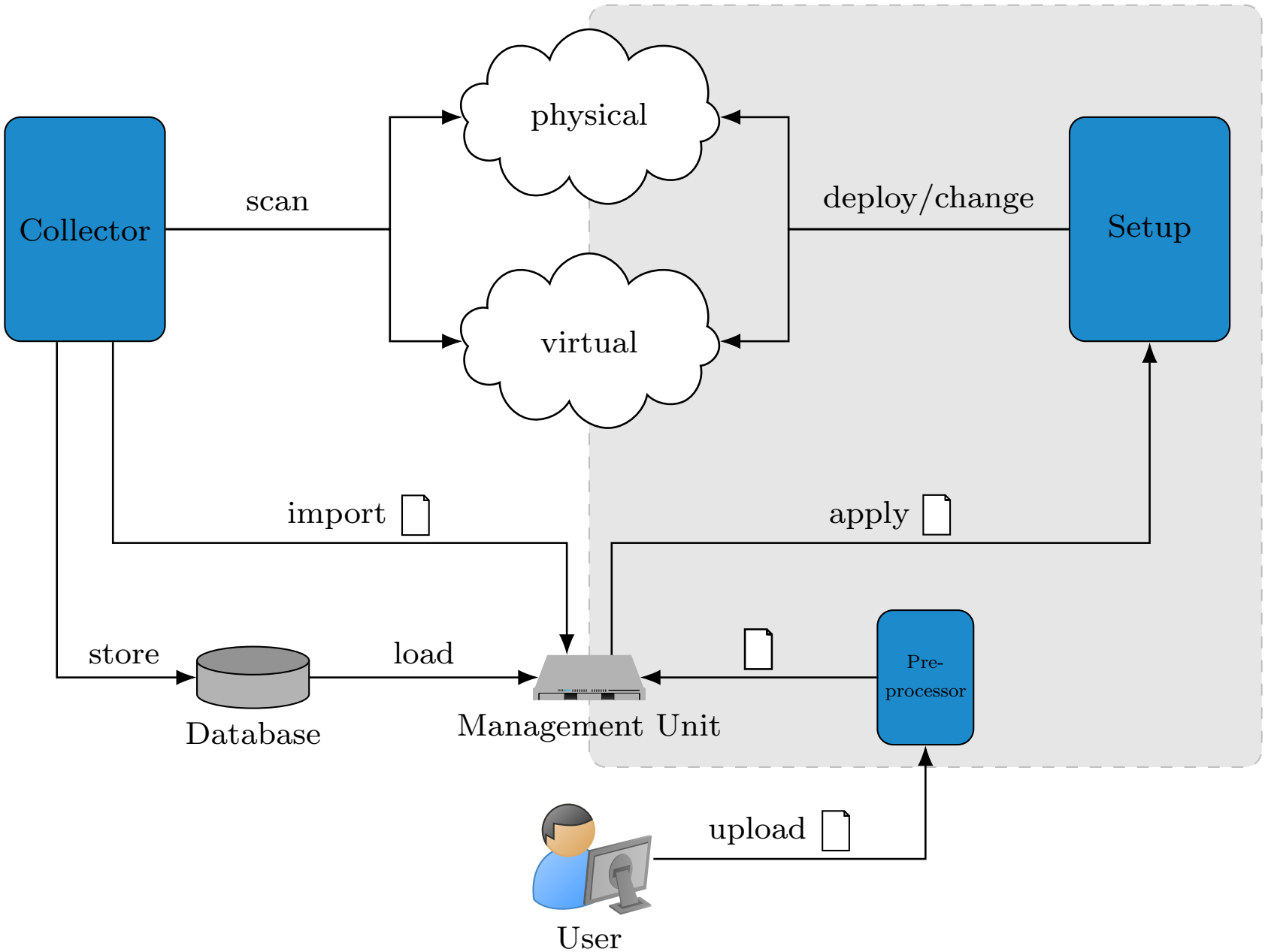


Architecture: Information Collection Component

Approaches for Information Collection

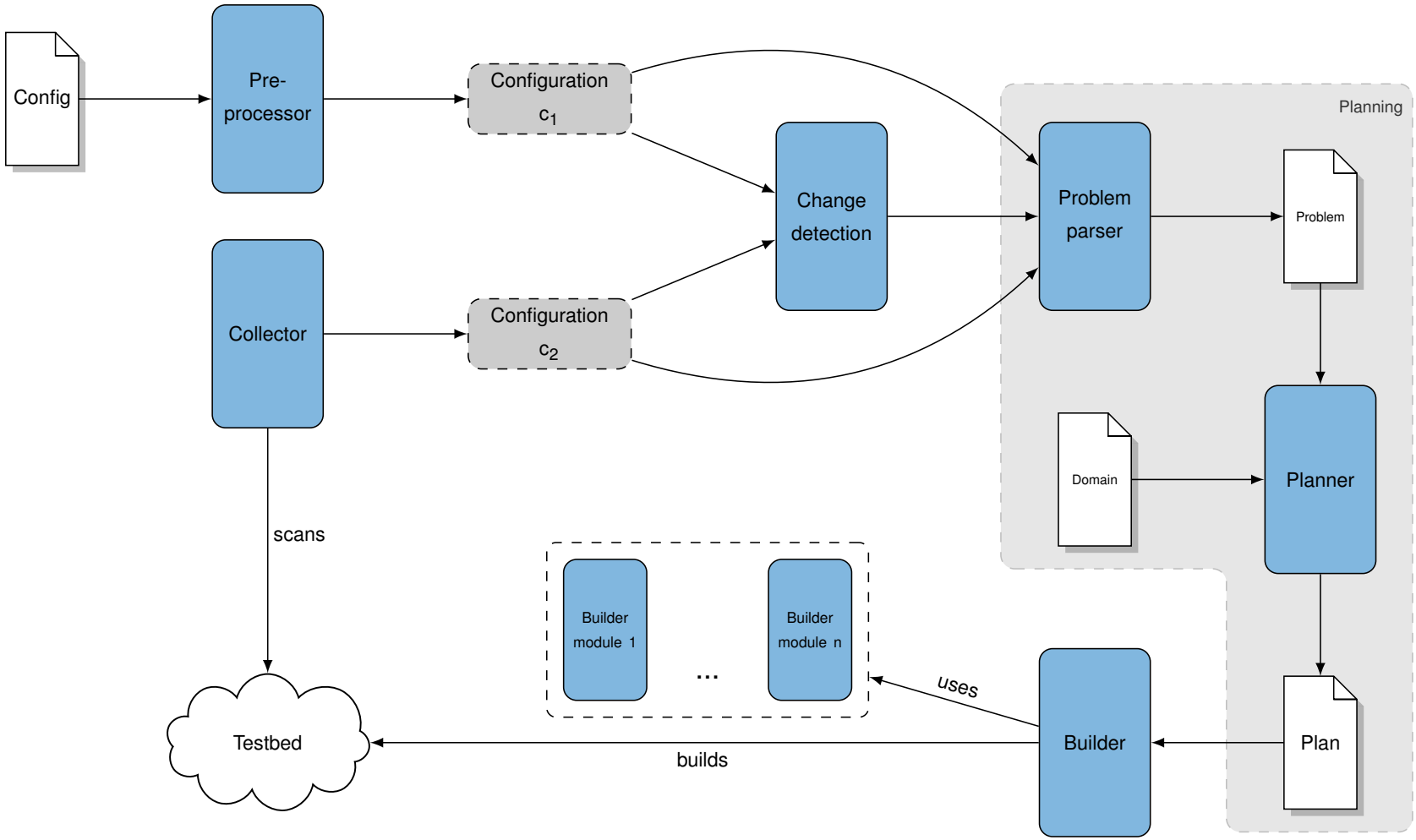
	Manual (xml)	Passive scans (tcpdump)	Active scans (NMAP)	protocol based (SNMP)	Direct access (ssh)	Client software (Zabbix)
Network traffic overhead	o	o	++	+	+	+
Client software	o	o	o	+	+	++
Direct access	o	+	o	o	++	+
Reliability	+	++	+	++	++	++
Topicality	o	o	+	+	+	++
Information variety	o	+	+	+	++	++
Information updatability	+	o	++	++	++	++

Architecture: Infrastructure Deployment Tool



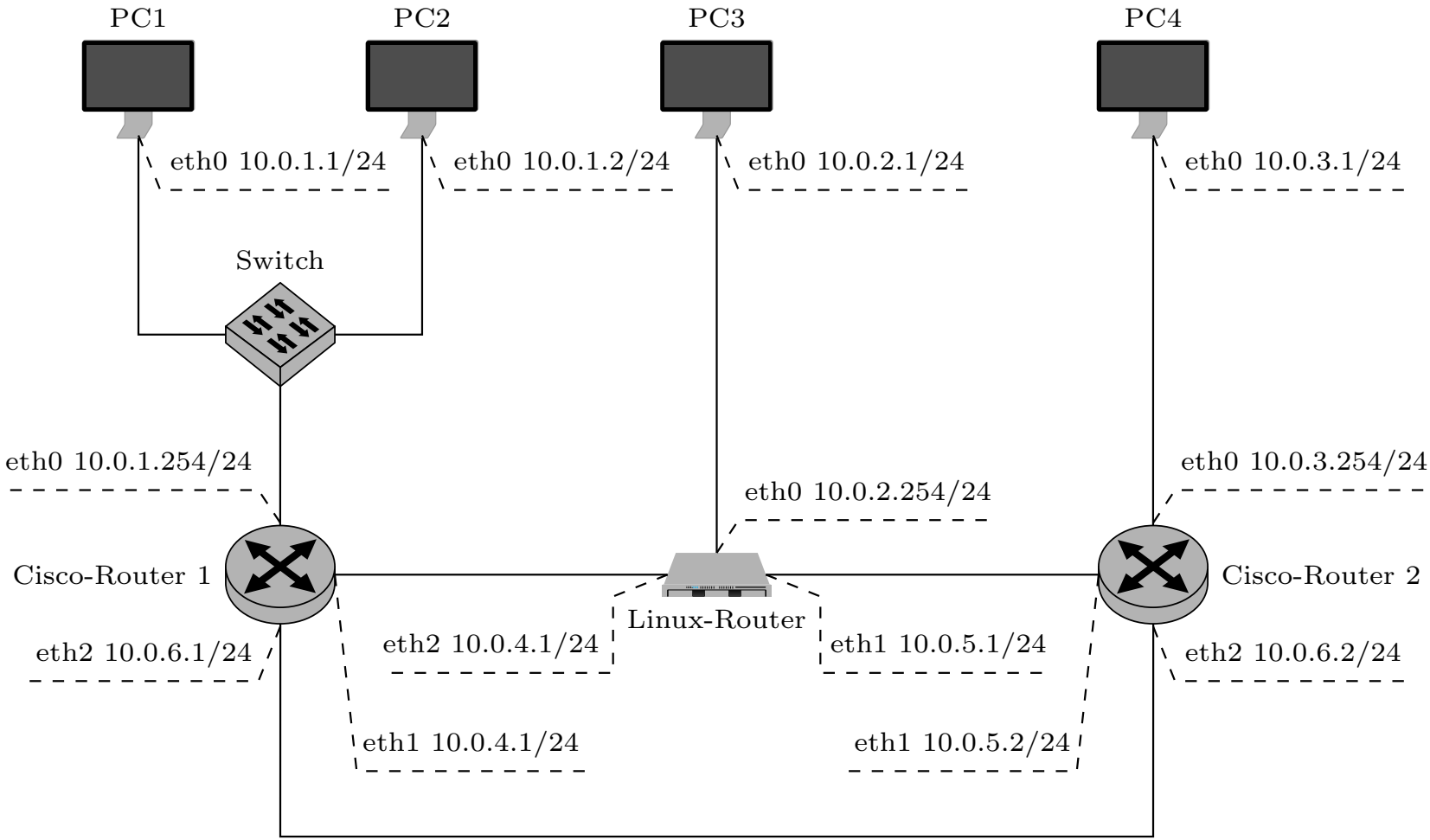
Architecture: Infrastructure Deployment Tool

Deployment Process



Case Study: iLab

Physical setup



Case Study: iLab

Deployment Phase

- Execution plan computed in <1 second
  - 92 steps
  - Contains: setup of virtual machines, networks, interfaces, routes,...
- Setup in our virtual testbed took ~42 minutes
  - Our builder modules utilize the XEN xapi toolstack
  - Most of the time required to clone the virtual machines and hard disk images
- Validation of setup in the testbed using the ping and traceroute tools

Conclusion and Future Work

- Conclusion:
  - Automated scanning and deployment of network environments can be done
  - Modularisation of such a tool is beneficial

Conclusion and Future Work

- Conclusion:
  - Automated scanning and deployment of network environments can be done
  - Modularisation of such a tool is beneficial
- Contribution:
  - Extensible framework for reproducible network setups
  - Applicable for virtual/physical/mixed environments
  - Incremental deployment process
  - Multiple collector/builder implementations
  - Case studie as prove of applicability

Conclusion and Future Work

- Conclusion:
  - Automated scanning and deployment of network environments can be done
  - Modularisation of such a tool is beneficial
- Contribution:
  - Extensible framework for reproducible network setups
  - Applicable for virtual/physical/mixed environments
  - Incremental deployment process
  - Multiple collector/builder implementations
  - Case studie as prove of applicability
- Future Work:
  - Implement additional collector/builder modules
  - Parallelize the deployment process
  - Include experiment execution

Literature

[1] V. Ahlers, F. Heine, B. Hellmann, C. Kleiner, L. Renners, T. Rossow, and R. Steuerwald.
Integrated Visualization of Network Security Metadata from Heterogeneous Data Sources.
In S. Mauw, B. Kordy, and S. Jajodia, editors, *Graphical Models for Security: Second International Workshop (GraMSec)*, pages 18–34, 2016.

[2] K. Ali.
Algorizmi: A configurable virtual testbed to generate datasets for offline evaluation of Intrusion Detection Systems.
Master's thesis, University of Waterloo, 2010.

[3] R. Bifulco, G. D. Stasi, and R. Canonico.
NEPTUNE for fast and easy deployment of OMF virtual network testbeds [Poster Abstract].
2010.

[4] H. Birkholz, I. Sieverdingbeck, K. Sohr, and C. Bormann.
IO: An Interconnected Asset Ontology in Support of Risk Management Processes.
In *Availability, Reliability and Security (ARES), 7th International Conference on*, pages 534–541, 2012.

[5] A. Carpen-Amarie, J. Cai, A. Costan, G. Antoniu, and L. Bougé.
Bringing Introspection Into the BlobSeer Data-Management System Using the MonALISA Distributed Monitoring Framework.
In *Complex, Intelligent and Software Intensive Systems (CISIS), International Conference on*, pages 508–513, 2010.

[6] C. Dobre, R. Voicu, and I. Legrand.
Monitoring large scale network topologies.
In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), IEEE 6th International Conference on*, volume 1, pages 218–222, 2011.

[7] M. Fox and D. Long.
PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains.
J. Artif. Int. Res., 20(1):61–124, 2003.

Literature

[8] M. Ghijsen, J. van der Ham, P. Grosso, C. Dumitru, H. Zhu, Z. Zhao, and C. de Laat.
A semantic-web approach for modeling computing infrastructures.
Computers & Electrical Engineering, 39(8):2553 – 2565, 2013.

[9] GÉANT.
GEANT2 common Network Information Service (cNIS) Schema Specification, `http://www.geant2.net`.

[10] Jeroen Johannes van der Ham.
A Semantic Model for Complex Computer Networks: The Network Description Language.
PhD thesis, University of Amsterdam, 2010.

[11] X. Jiang and D. Xu.
vBET: A VM-based Emulation Testbed.
In *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pages 95–104, 2003.

[12] L. Lorenzin and N. Cam-Winget.
Security Automation and Continuous Monitoring (SACM) Requirements.
Internet-Draft draft-ietf-sacm-requirements-15, Internet Engineering Task Force, 2016.

[13] G. Lyon.
The Offical Nmap Project Guide to Network Discovery and Securtiy Scanning.
2009.

[14] G. Lyon.
nmap(1) – Linux man page, 2015.

[15] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins.
PDDL – The Planning Domain Definition Language.
1998.

Literature

[16] OpenVAS.
Project Homepage `http://www.openvas.org`.

[17] P. Owezarski, P. Berthou, Y. Labit, and D. Gauchard.
LaasNetExp: A Generic Polymorphic Platform for Network Emulation and Experiments.
In *Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, number 24, pages 1–9, 2008.

[18] T. Taketa and Y. Hiranaka.
Network Design Assistant System based on Network Description Language.
In *Advanced Communication Technology (ICACT), 15th International Conference on*, pages 515–518, 2013.

[19] B. Tierney, J. Metzger, J. Boote, E. Boyd, A. Brown, R. Carlson, M. Zekauskas, J. Zurawski, M. Swany, and M. Grigoriev.
perfSONAR: Instantiating a Global Network Measurement Framework.
In *In SOSP Workshop on Real Overlays and Distributed Systems (ROADS09)*. ACM, 2009.

[20] Trusted Network Connect Work Group.
TNC IF-MAP Bindings for SOAP, Version 2.2, Revision 10, 2014.

[21] Trusted Network Connect Work Group.
TNC MAP Content Authorization, Version 1.0, Revision 36, 2014.

[22] J. Undercoffer, J. Pinkston, A. Joshi, and T. Finin.
A Target-Centric Ontology for Intrusion Detection.
In *Proceding of the 9th Workshop on Ontologies and Distributed Systems*, pages 47–58, 2004.

[23] J. van der Ham, F. Dijkstra, R. Lapacz, and A. Brown.
The Network Markup Language (NML): A Standardized Network Topology Abstraction for Inter-domain and Cross-layer Network Applications, 2013.

[24] J. van der Ham, F. Dijkstra, R. Łapacz, and J. Zurawski.
Network Markup Language Base Schema version 1, 2013.

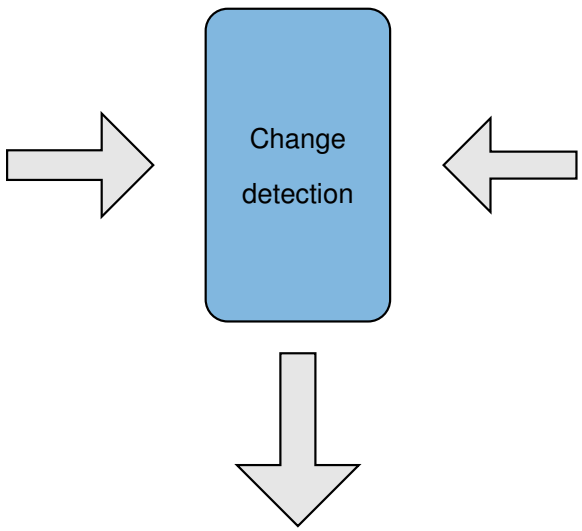
Literature

[25] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar.
An Integrated Experimental Environment for Distributed Systems and Networks.
pages 255–270, 2002.

Change Detection

networks:
- ▶ net-1
- ▶ net-2

hosts:
- ▶ router-1
  - • interfaces:
    - – eth0: 10.10.1.254/24
    - – eth1: 10.10.2.1/24
- ▶ host-1
  - • interfaces:
    - – eth0: 10.10.1.1/24



networks:
- ▶ net-1
- ▶ net-2

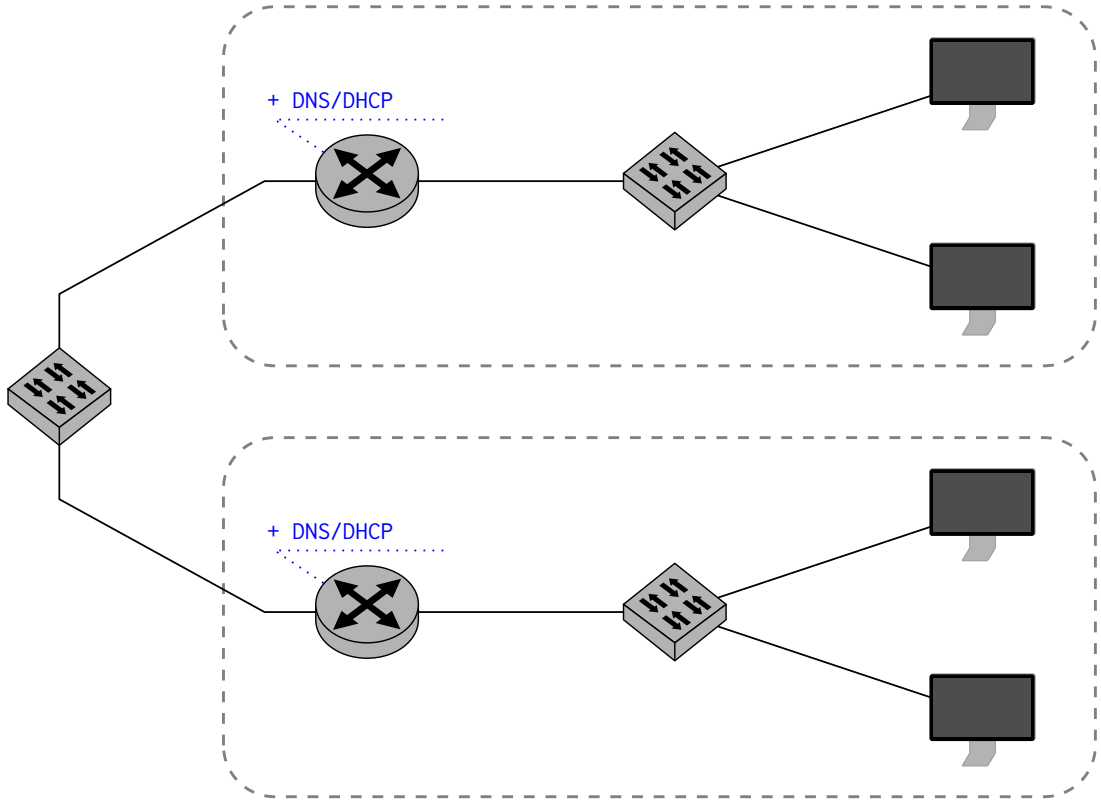
hosts:
- ▶ router-1
  - • interfaces:
    - – eth0: 10.10.1.254/24
    - – eth1: 10.10.2.254/24
- ▶ host-1
  - • interfaces:
    - – eth0: 10.10.1.1/24
- ▶ host-2
  - • interfaces:
    - – eth0: 10.10.2.1/24

The change detection marks `router-1.eth1.ip` as changed and `host-2` as new.

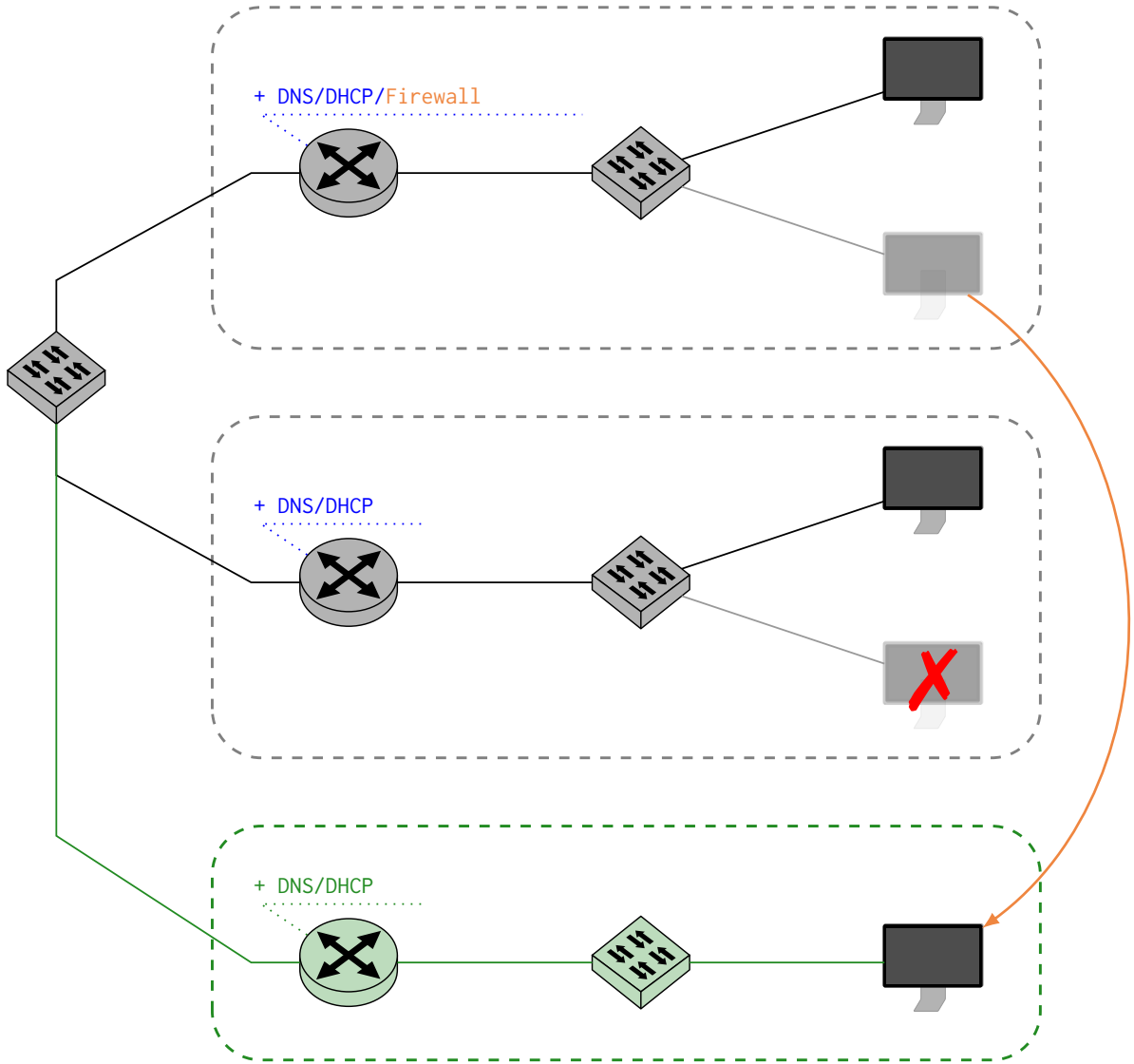
Planning

- Based on the input, the planner creates a deployment plan
- Currently the Planning Domain Definition Language (PDDL [15, 7]) is used
  - Domain description: describes the problem domain, static
    - Describes object types, predicates, actions
    - Actions apply to objects, have preconditions and have an effect
    - Provided with INSALATA for our information model
  - Problem description: describes the instance of the domain
    - Depends on current/desired state
- Output is given to the Builder which chooses fitting builder modules to realize the deployment

Planning Example



Planning Example



Planning example

new

- router-3

changed

- router-1.firewall
- router-1.routing
- router-2.routing
- host-3.eth0

removed

- host-4
- host-4-hdd

1. (createhost router-3)
2. (configurefirewall router-1)
3. (configurerouting router-1)
4. (configurerouting router-2)
5. (createnetwork net-3)
6. (shutdown host-4)
7. (removehost host-4)
8. (removedisk host-4-hdd host-4)
9. (createinterface router-3.eth0)
10. (createinterface router-3.eth1)
11. (bootunnamed router-3)
12. (configureinterface router-3.eth0)
13. (configureinterface router-3.eth1)
14. (configurerouting router-3)
15. (name router-3)
16. (rebootandnamed router-3)
17. (shutdown host-3)
18. (configurenetwork host-3.eth0)
19. (boot host-3)
20. (configuredhcp 172.16.246.1:67_udp_dhcp)
21. (configuredns 172.16.246.1:53_udp_dns)

Planning example

new

- router-3

changed

- router-1.firewall
- router-1.routing
- router-2.routing
- host-3.eth0

removed

- host-4
- host-4-hdd

1. (createhost router-3)
2. (configurefirewall router-1)
3. (configurerouting router-1)
4. (configurerouting router-2)
5. (createnetwork net-3)
6. (shutdown host-4)
7. (removehost host-4)
8. (removedisk host-4-hdd host-4)
9. (createinterface router-3.eth0)
10. (createinterface router-3.eth1)
11. (bootunnamed router-3)
12. (configureinterface router-3.eth0)
13. (configureinterface router-3.eth1)
14. (configurerouting router-3)
15. (name router-3)
16. (rebootandnamed router-3)
17. (shutdown host-3)
18. (configurenetwork host-3.eth0)
19. (boot host-3)
20. (configuredhcp 172.16.246.1:67_udp_dhcp)
21. (configuredns 172.16.246.1:53_udp_dns)

Planning example

new

- router-3

changed

- router-1.firewall
- router-1.routing
- router-2.routing
- host-3.eth0

removed

- host-4
- host-4-hdd

1. (createhost router-3)
2. (configurefirewall router-1)
3. (configurerouting router-1)
4. (configurerouting router-2)
5. (createnetwork net-3)
6. (shutdown host-4)
7. (removehost host-4)
8. (removedisk host-4-hdd host-4)
9. (createinterface router-3.eth0)
10. (createinterface router-3.eth1)
11. (bootunnamed router-3)
12. (configureinterface router-3.eth0)
13. (configureinterface router-3.eth1)
14. (configurerouting router-3)
15. (name router-3)
16. (rebootandnamed router-3)
17. (shutdown host-3)
18. (configurenetwork host-3.eth0)
19. (boot host-3)
20. (configuredhcp 172.16.246.1:67_udp_dhcp)
21. (configuredns 172.16.246.1:53_udp_dns)

Planning example

new

- router-3

changed

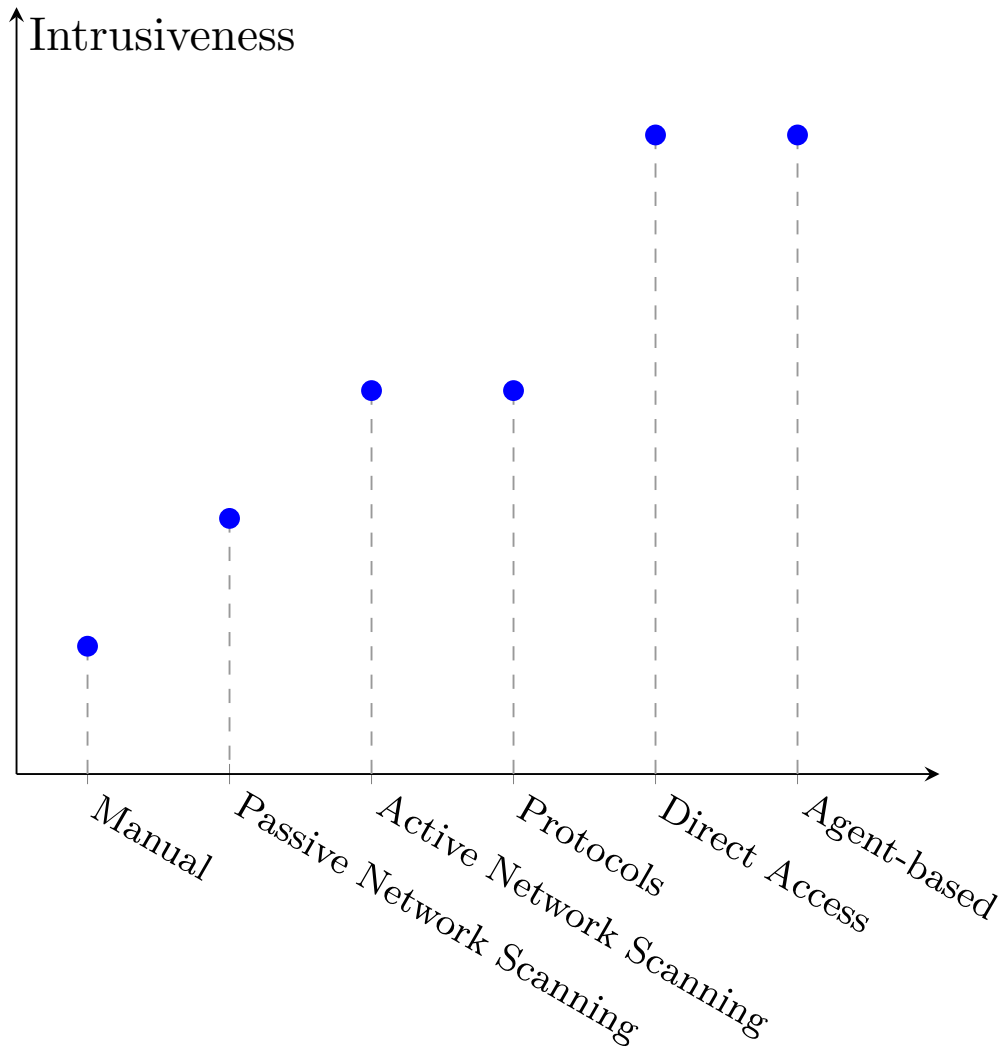
- router-1.firewall
- router-1.routing
- router-2.routing
- host-3.eth0

removed

- host-4
- host-4-hdd

1. (createhost router-3)
2. (configurefirewall router-1)
3. (configurerouting router-1)
4. (configurerouting router-2)
5. (createnetwork net-3)
6. (shutdown host-4)
7. (removehost host-4)
8. (removedisk host-4-hdd host-4)
9. (createinterface router-3.eth0)
10. (createinterface router-3.eth1)
11. (bootunnamed router-3)
12. (configureinterface router-3.eth0)
13. (configureinterface router-3.eth1)
14. (configurerouting router-3)
15. (name router-3)
16. (rebootandnamed router-3)
17. (shutdown host-3)
18. (configurenetwork host-3.eth0)
19. (boot host-3)
20. (configuredhcp 172.16.246.1:67_udp_dhcp)
21. (configuredns 172.16.246.1:53_udp_dns)

Intrusiveness of Information Collection



Quality of Information Depending on Collection Method