Software-defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim

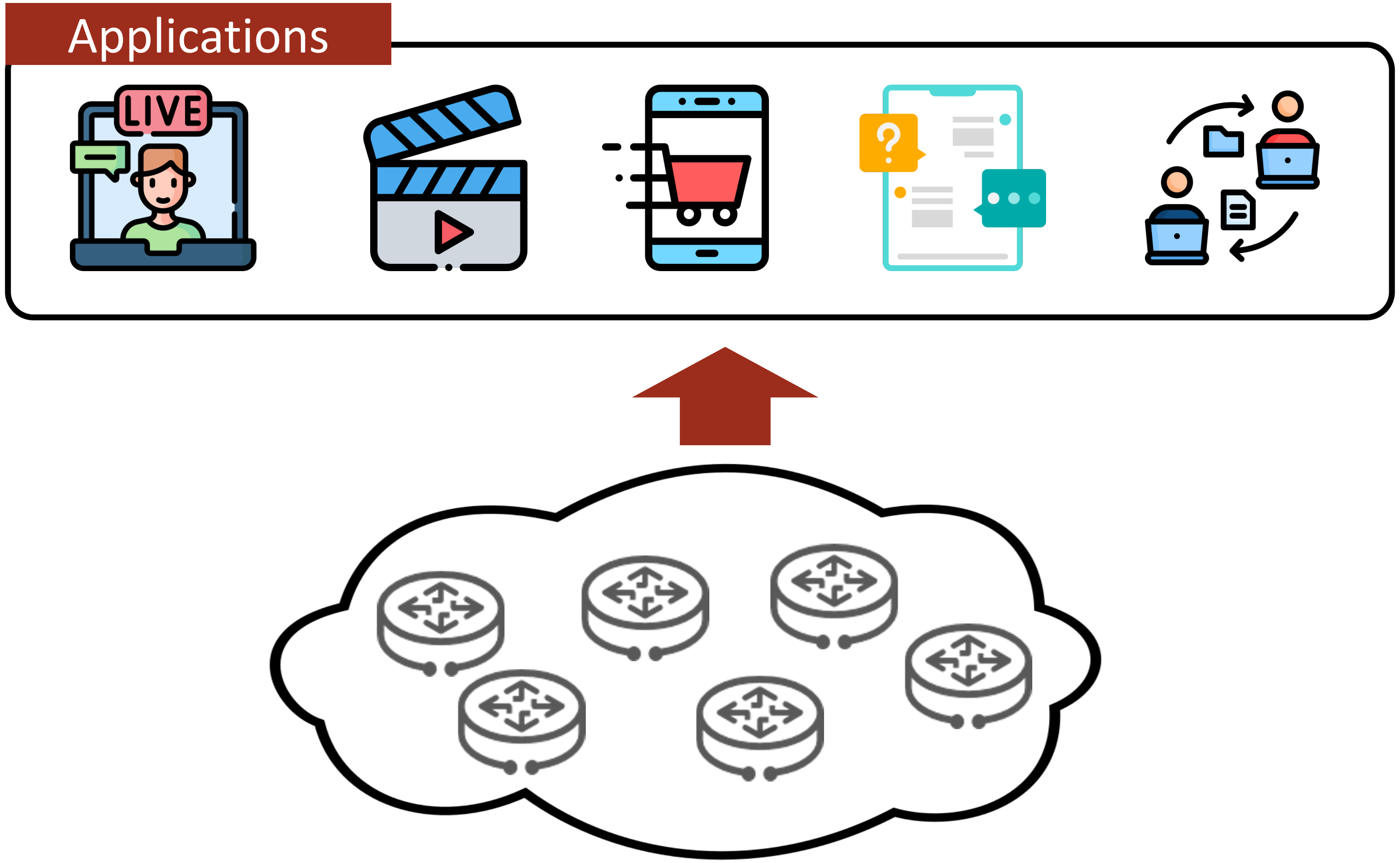
Huangxun Chen[1], Yukai Miao[1,2], Li Chen[3], Haifeng Sun[4], Hong Xu[5], Libin Liu[6], Gong Zhang[1], Wei Wang[7]

[1]Huawei Theory Lab, [2]University of New South Wales, [3]Zhongguancun Laboratory

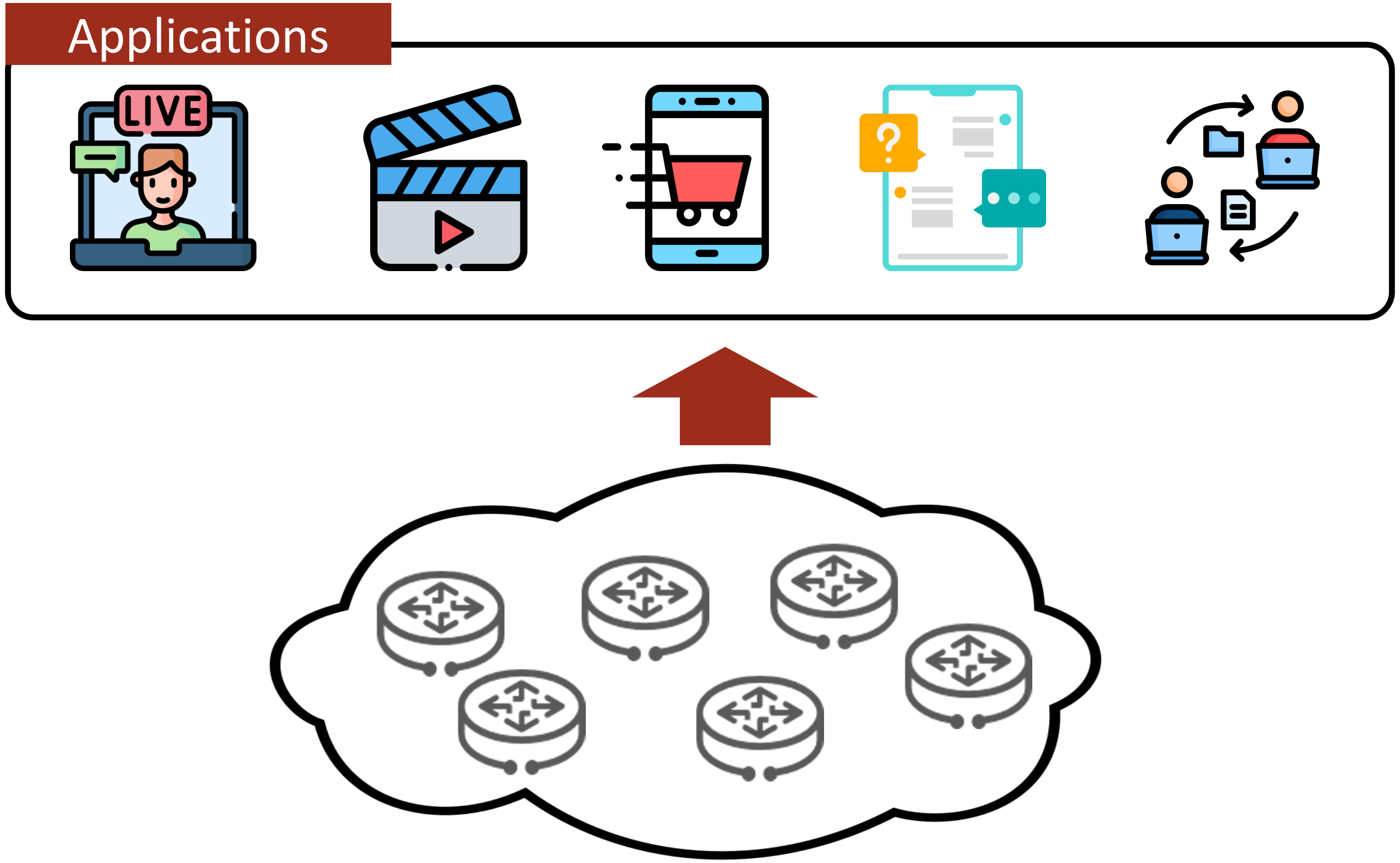
[4]Beijing University of Posts and Telecommunications, [5]The Chinese University of Hong Kong

[6]Shandong Computer Science Center (National Supercomputer Center in Jinan)

[7]Hong Kong University of Science and Technology (Guangzhou & Hong Kong)

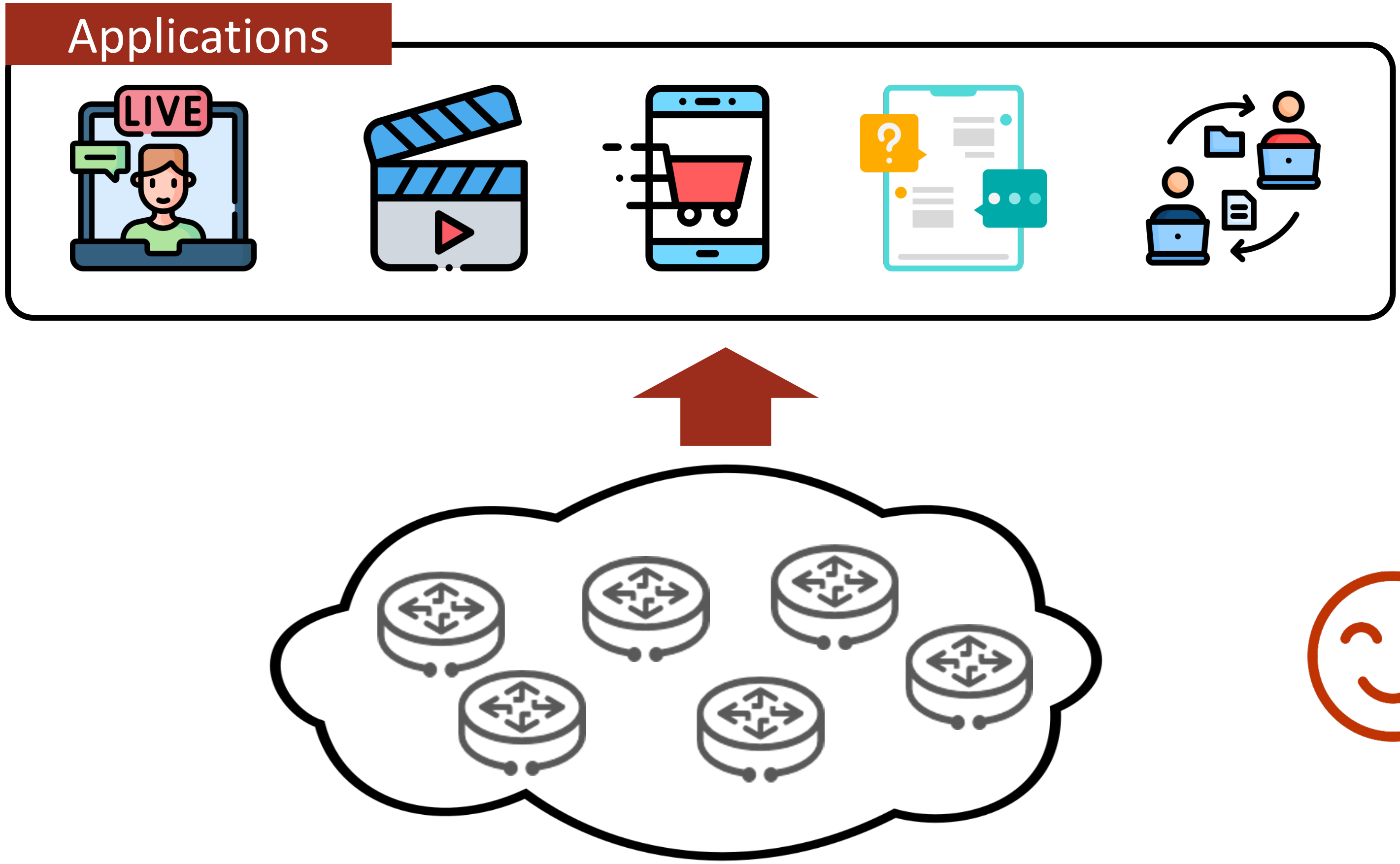

Applications
LIVE

The ideal network for NetOps consists of homogeneous devices

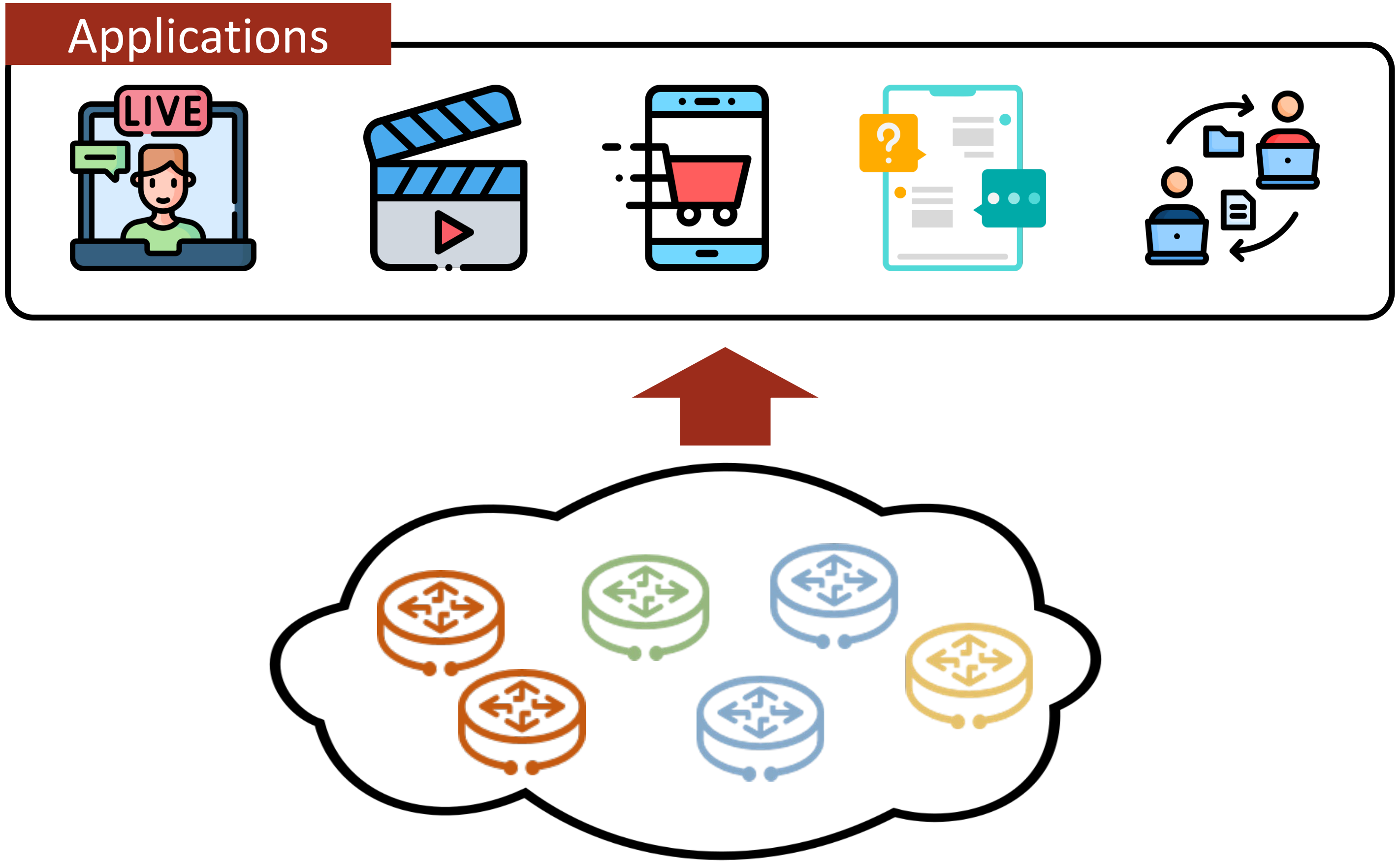


The ideal network (Homogeneous Device Model)

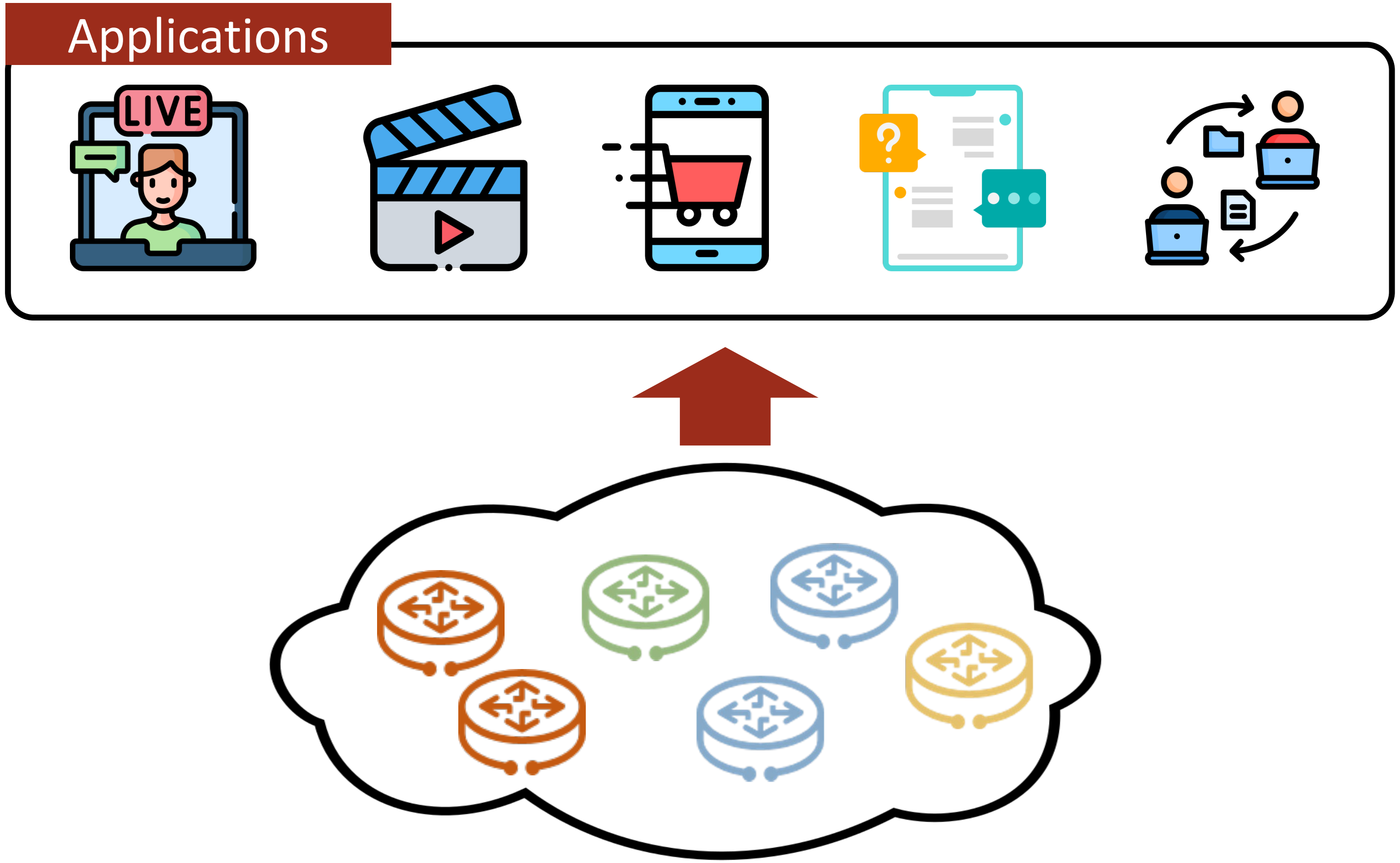
The ideal network for NetOps consists of homogeneous devices



The ideal network (Homogeneous Device Model)

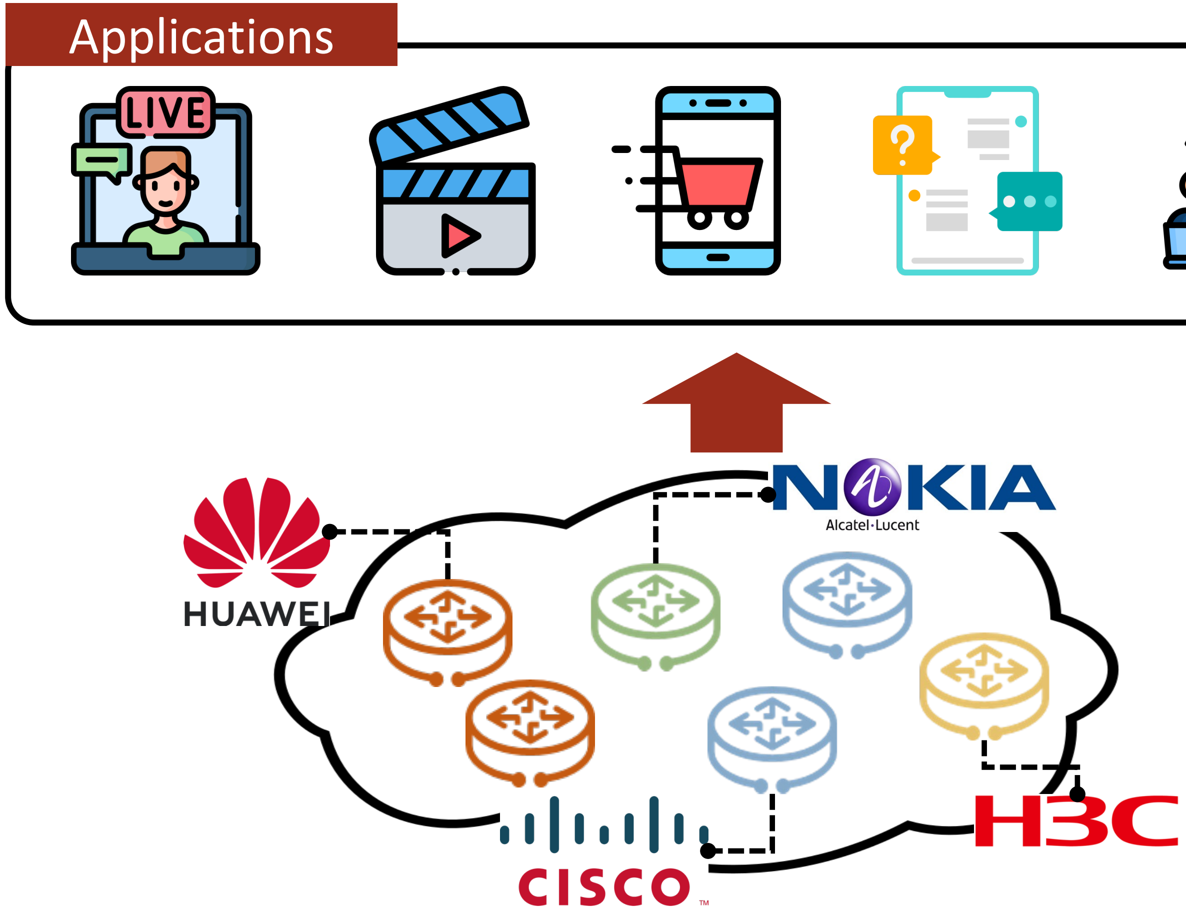

Applications
LIVE

The real network consists of multi-vendor heterogeneous devices



The real network (Heterogeneous Device Model)

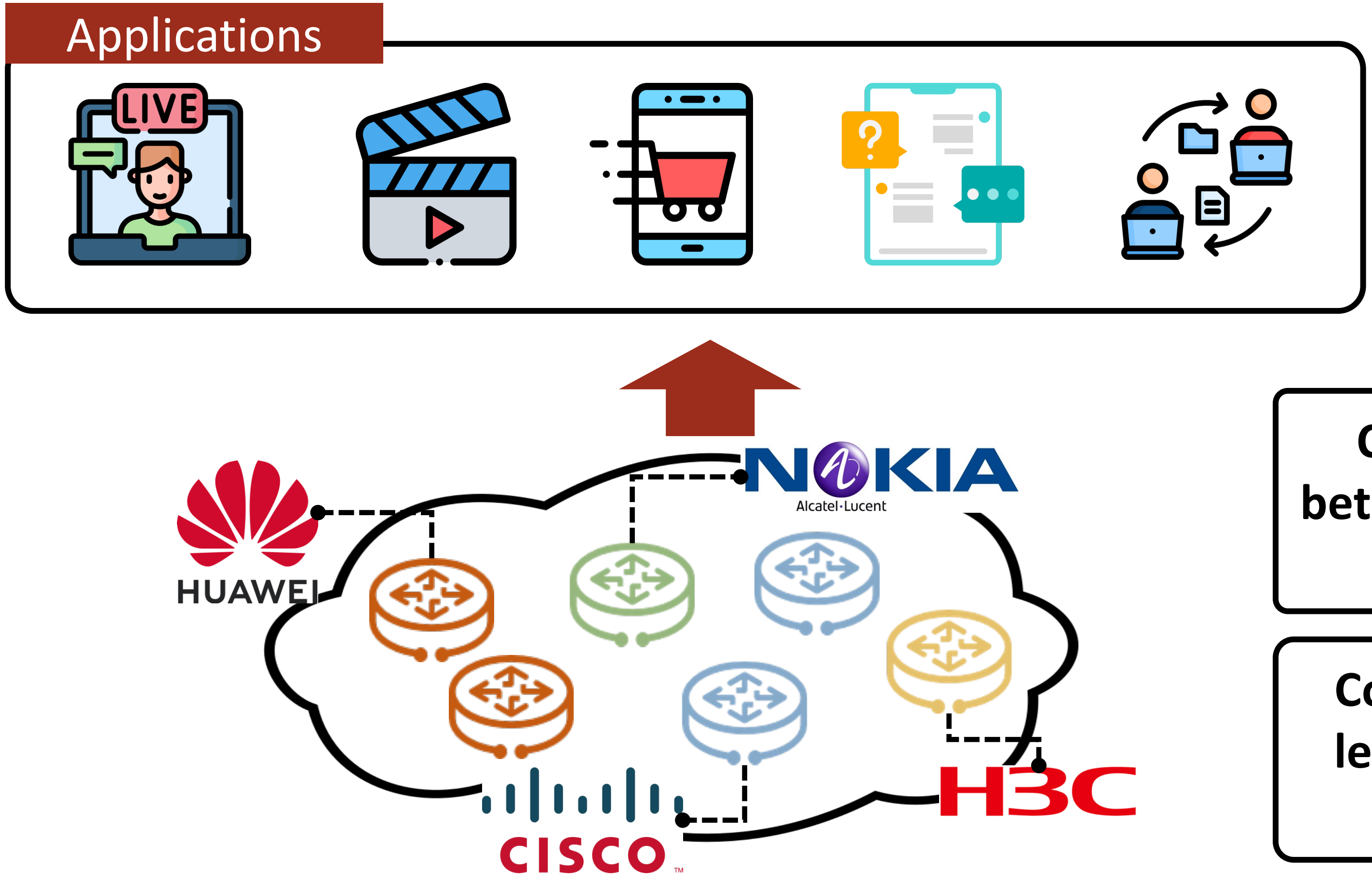
The real network consists of multi-vendor heterogeneous devices



Competition between multiple vendors

The real network (Heterogeneous Device Model)

The real network consists of multi-vendor heterogeneous devices

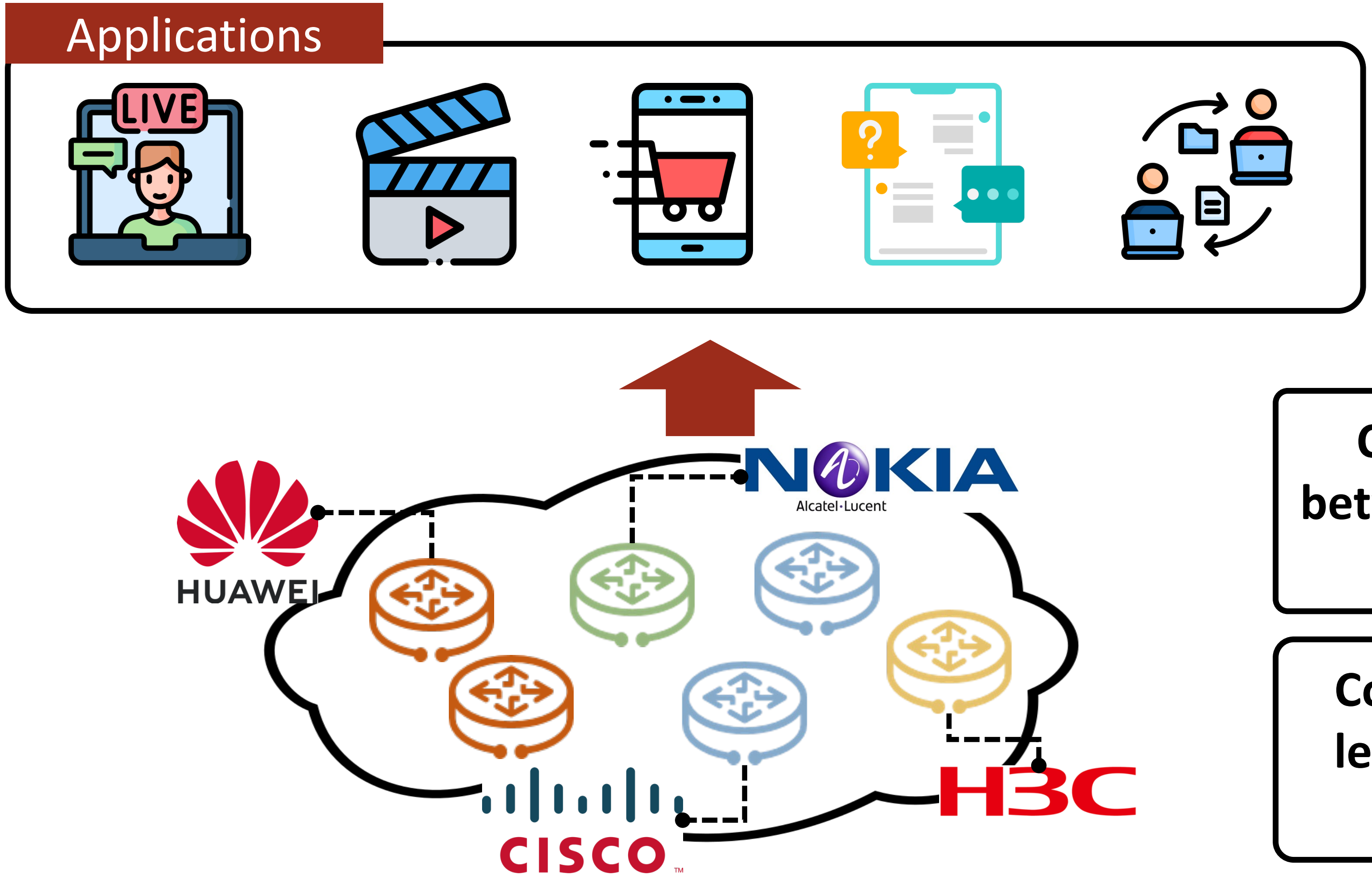


Competition between multiple vendors

Coexistence of legacy & latest devices

The real network (Heterogeneous Device Model)

The real network consists of multi-vendor heterogeneous devices

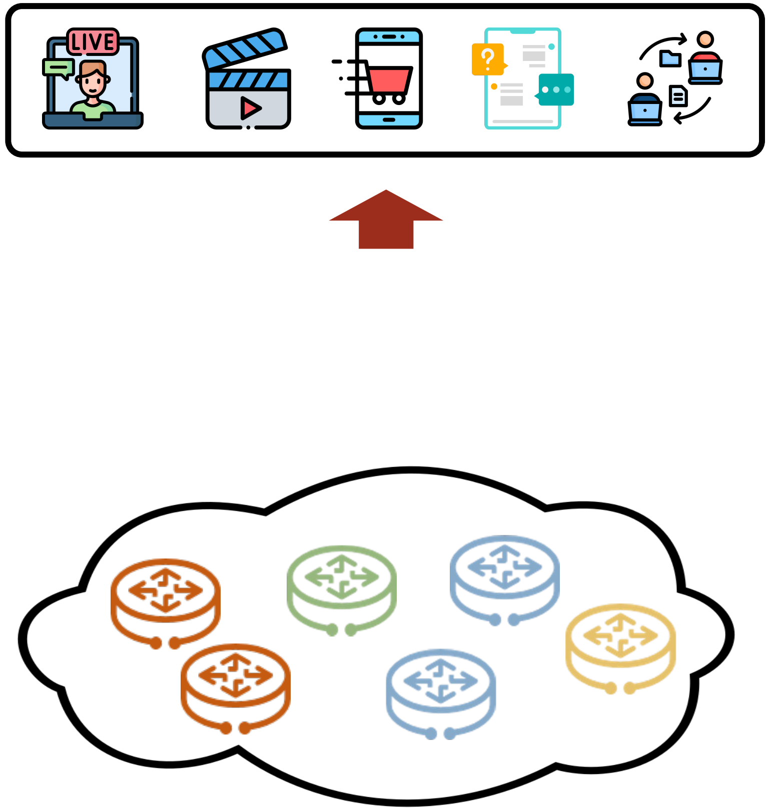


Competition between multiple vendors

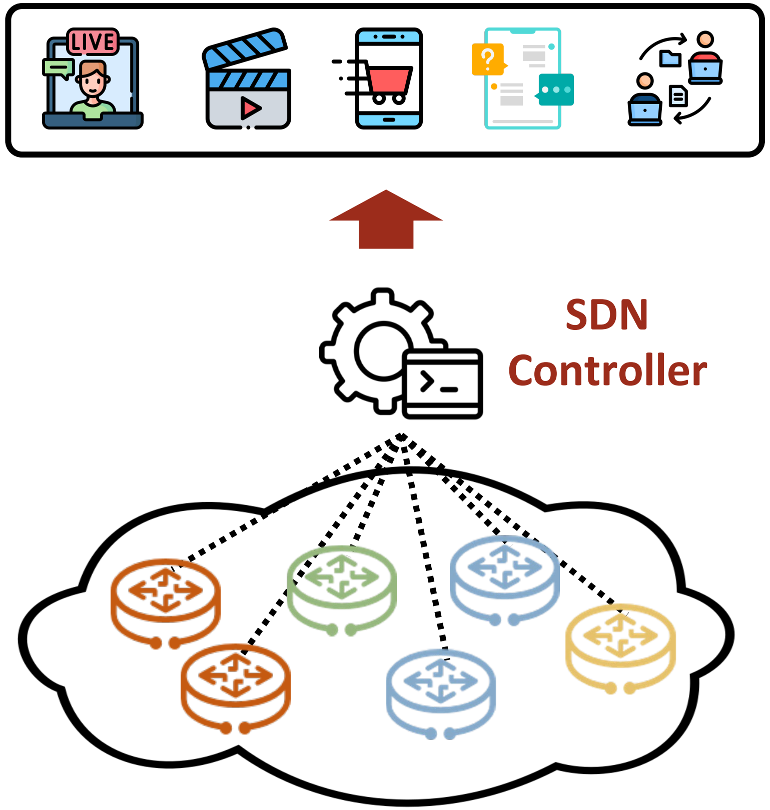
Coexistence of legacy & latest devices

The real network (Heterogeneous Device Model)

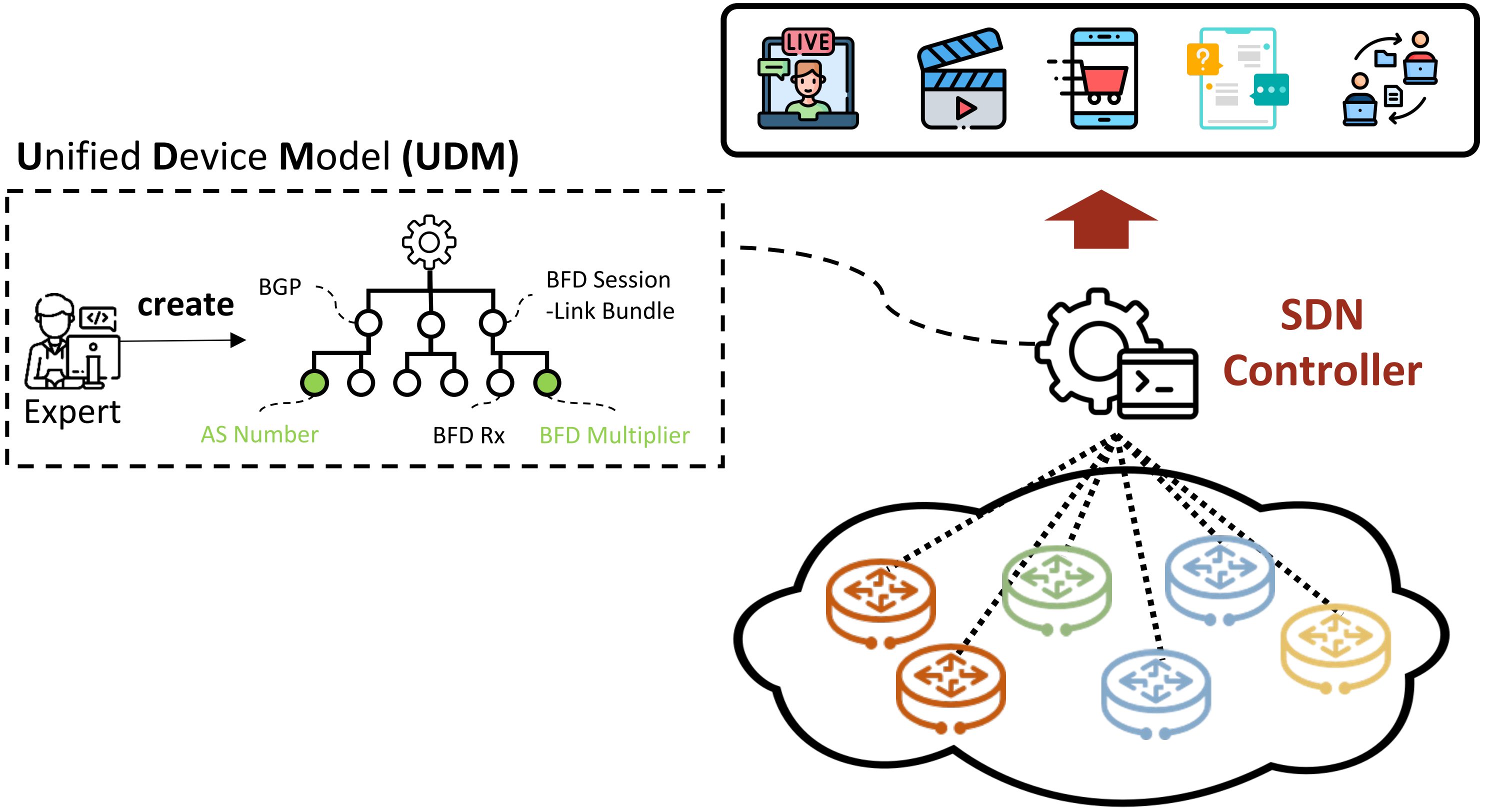
Managing a multi-vendor network is a pain...



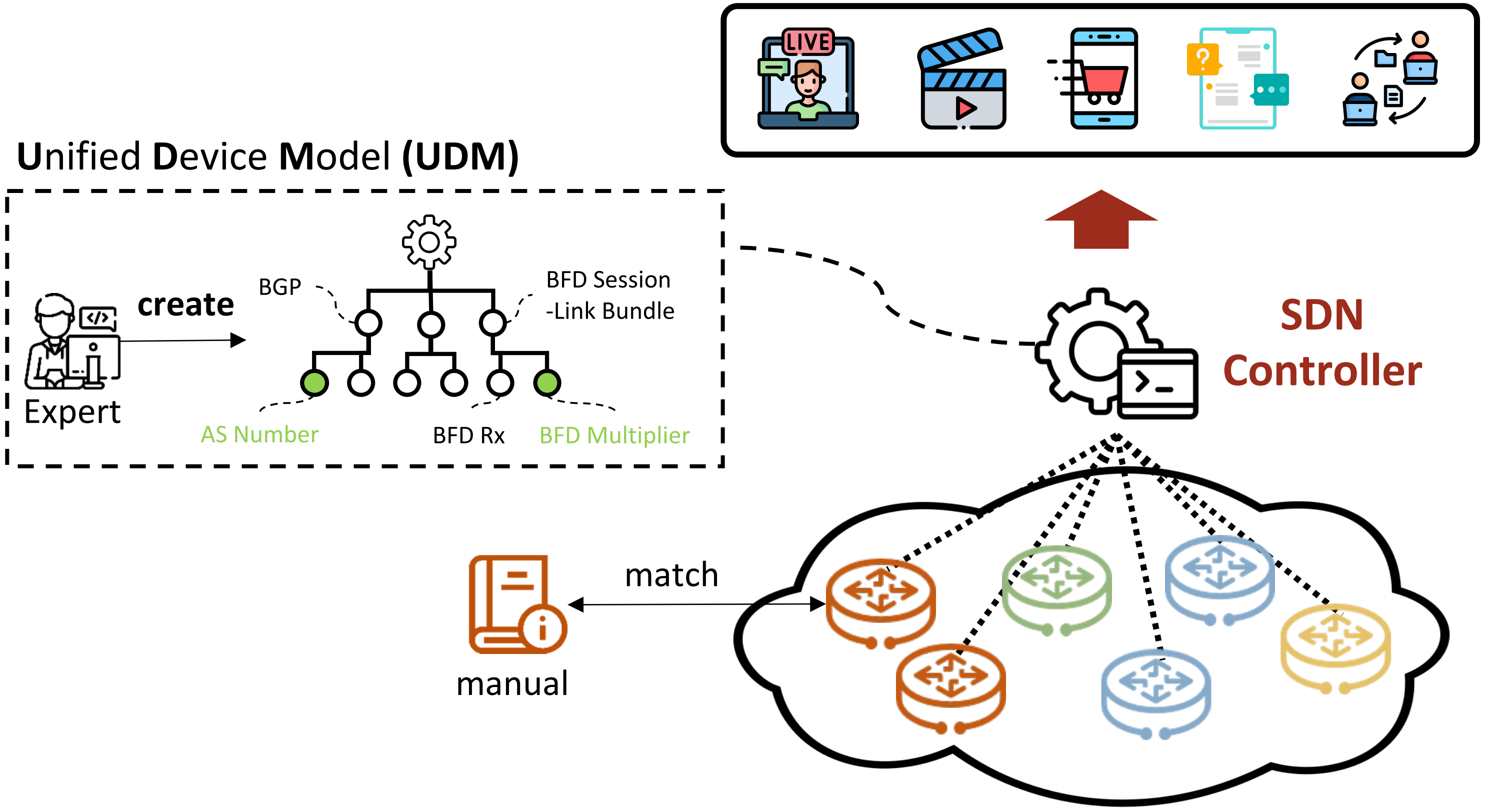
Managing a multi-vendor network is a pain…



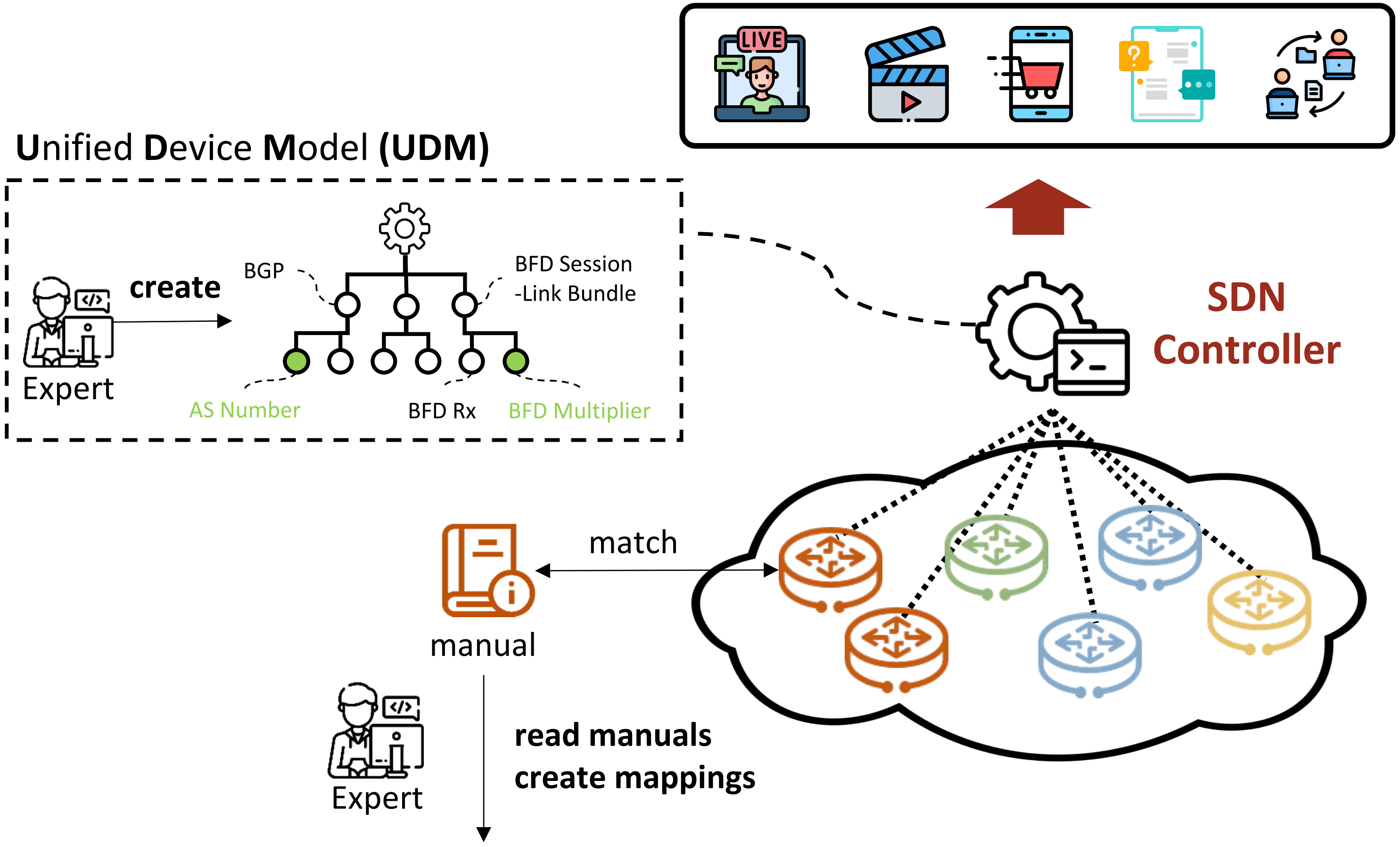
Managing a multi-vendor network is a pain…



Managing a multi-vendor network is a pain…



Managing a multi-vendor network is a pain…



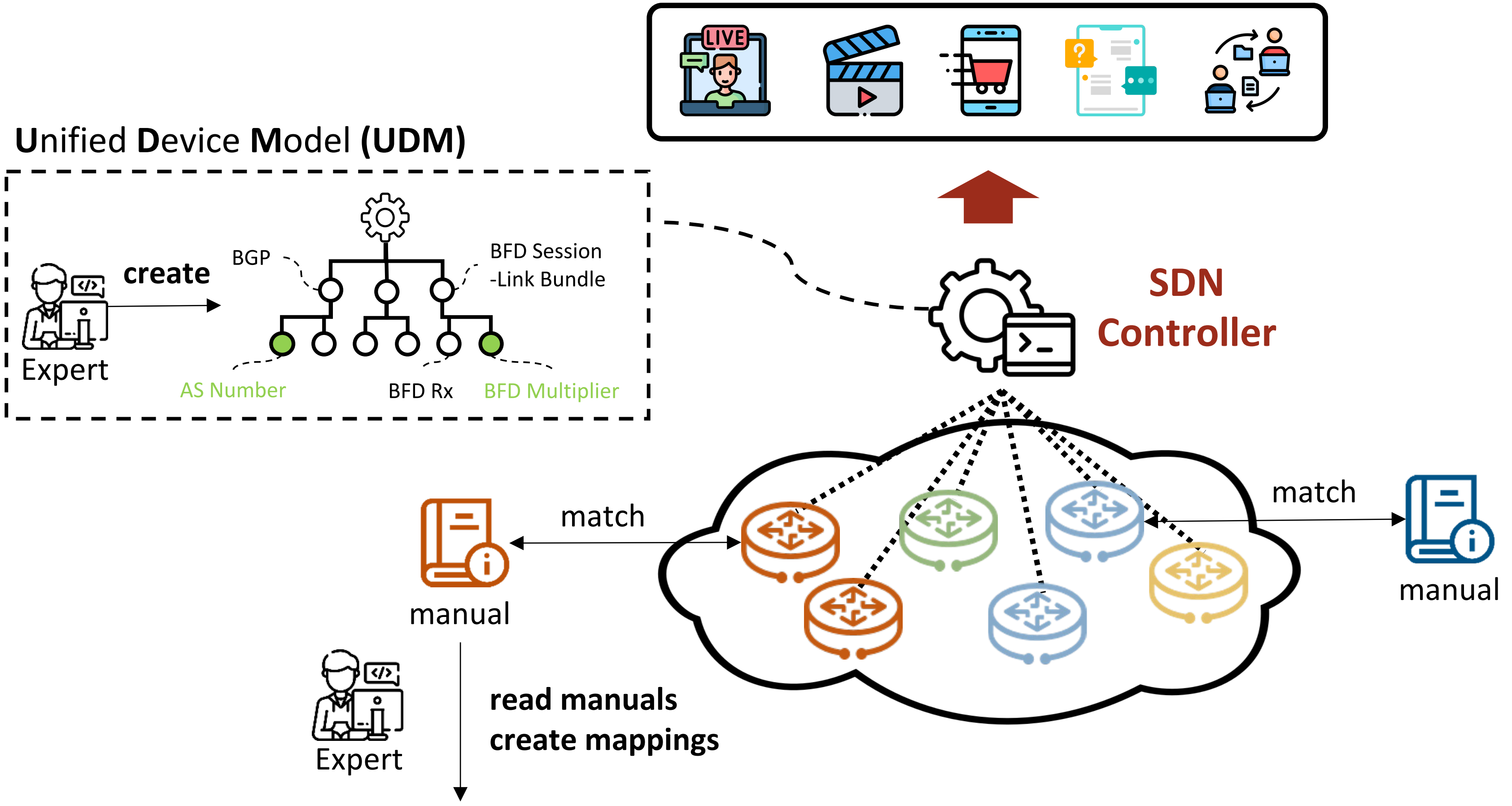
CLI command

```
system-view
  bgp <as-number>
```

CLI command

```
system-view
  bfd <session-name> bind peer-ip <peer-ip>
    detect-multiplier <multiplier>
```

Managing a multi-vendor network is a pain…



Unified Device Model (UDM)

Expert

create

BGP

BFD Session
-Link Bundle

AS Number

BFD Rx

BFD Multiplier

SDN
Controller

match

manual

match

manual

Expert

read manuals
create mappings

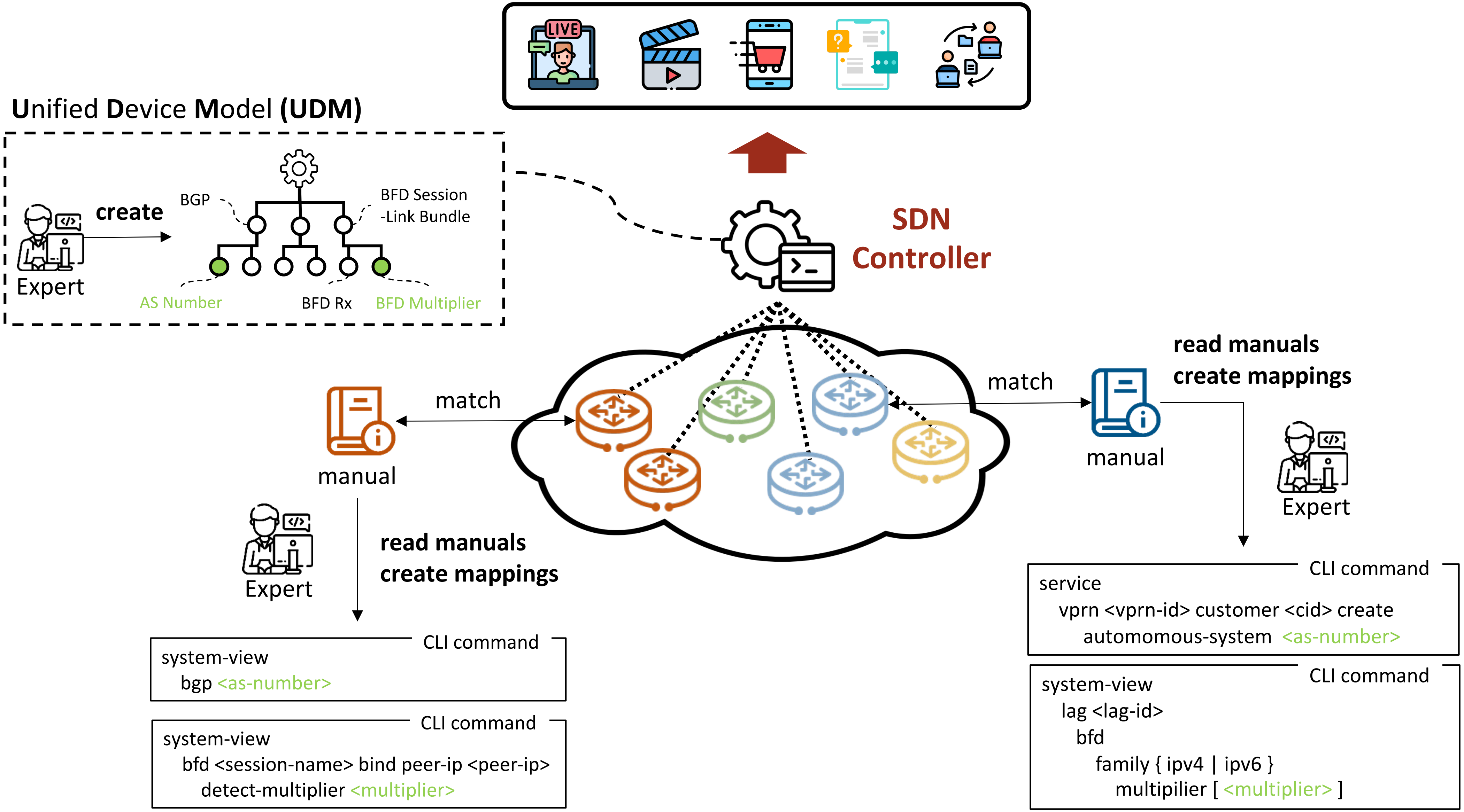
CLI command

```
system-view
  bgp <as-number>
```

CLI command

```
system-view
  bfd <session-name> bind peer-ip <peer-ip>
    detect-multiplier <multiplier>
```

Managing a multi-vendor network is a pain…



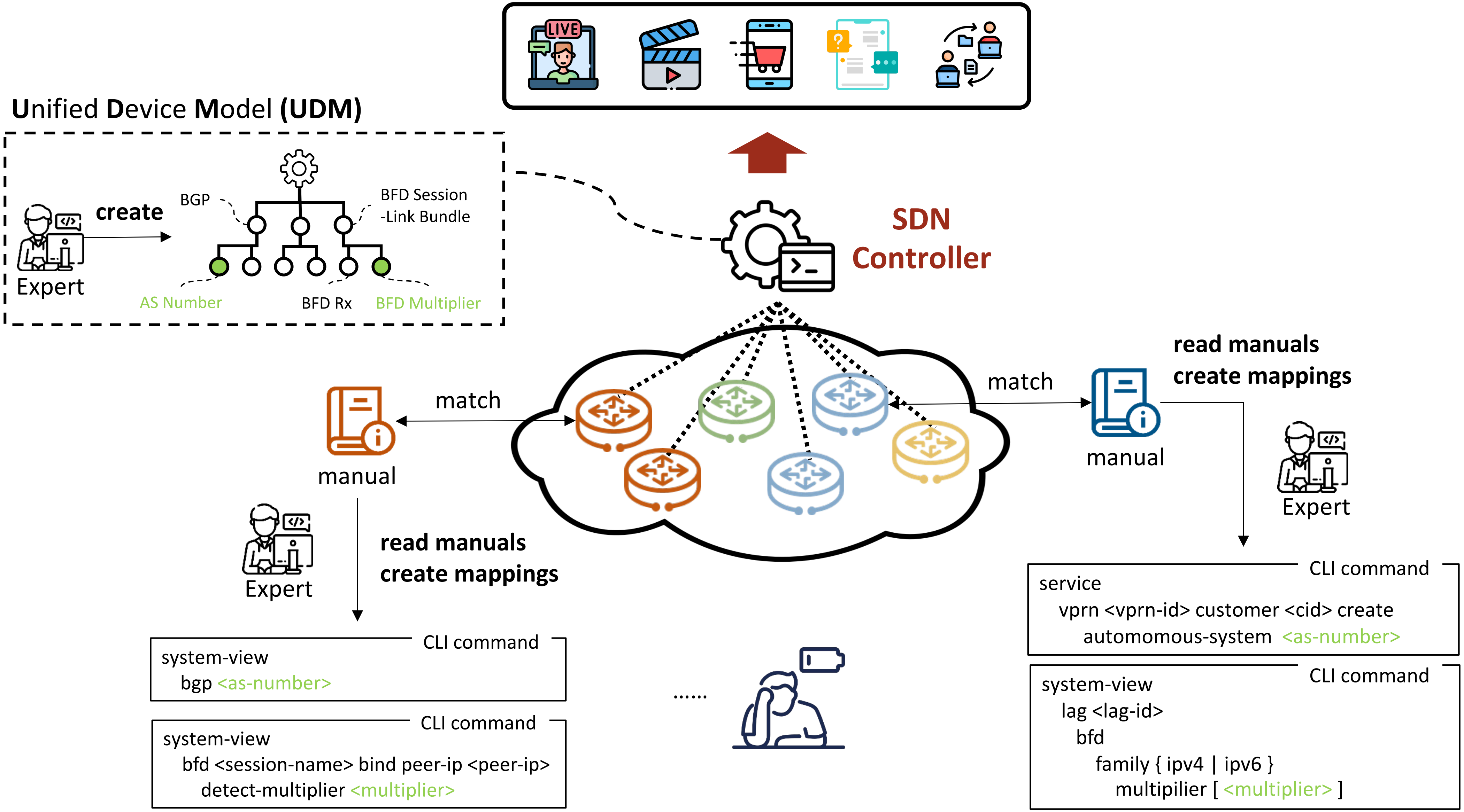
```
system-view
  bgp <as-number>
```

```
system-view
  bfd <session-name> bind peer-ip <peer-ip>
    detect-multiplier <multiplier>
```

```
service
  vprn <vprn-id> customer <cid> create
    automomous-system  <as-number>
```

```
system-view
  lag <lag-id>
    bfd
      family { ipv4 | ipv6 }
        multipilier [ <multiplier> ]
```

Managing a multi-vendor network is a pain…



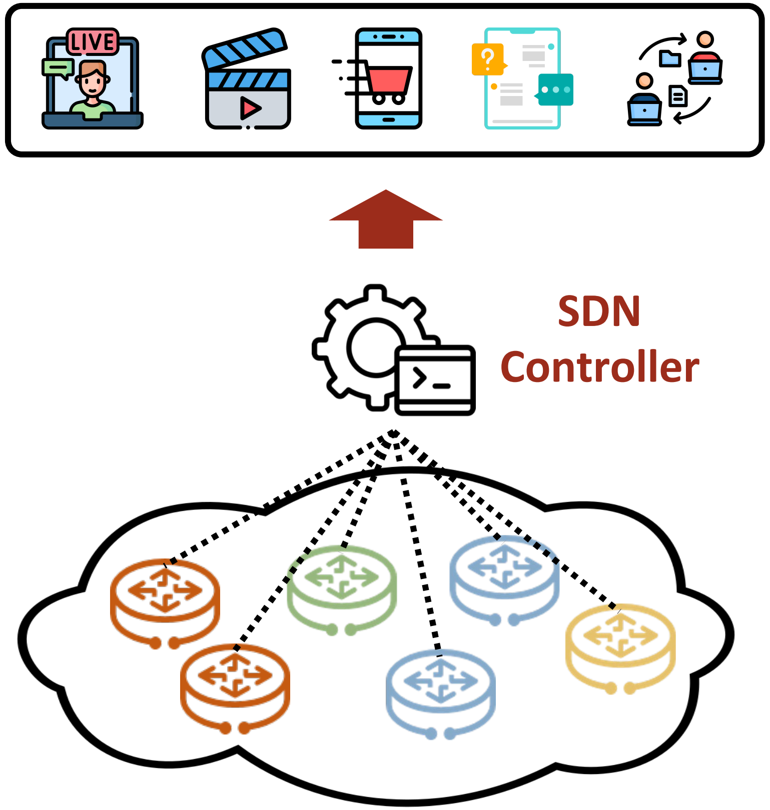
```
system-view
  bgp <as-number>
```

```
system-view
  bfd <session-name> bind peer-ip <peer-ip>
    detect-multiplier <multiplier>
```

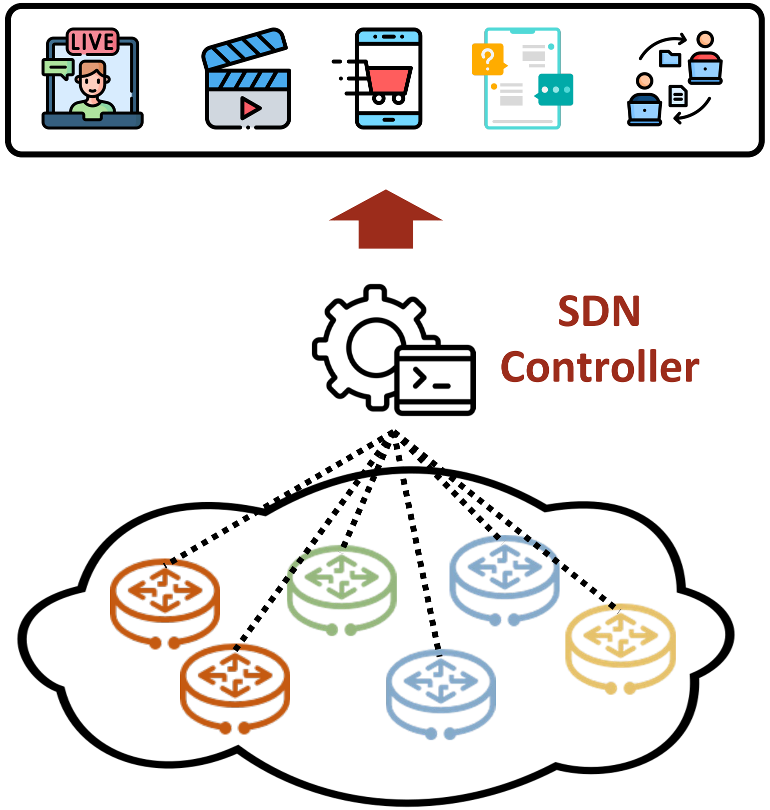
```
service
  vprn <vprn-id> customer <cid> create
     automomous-system  <as-number>
```

```
system-view
  lag <lag-id>
    bfd
      family { ipv4 | ipv6 }
        multipilier [ <multiplier> ]
```

Current SNA approaches require significant human efforts

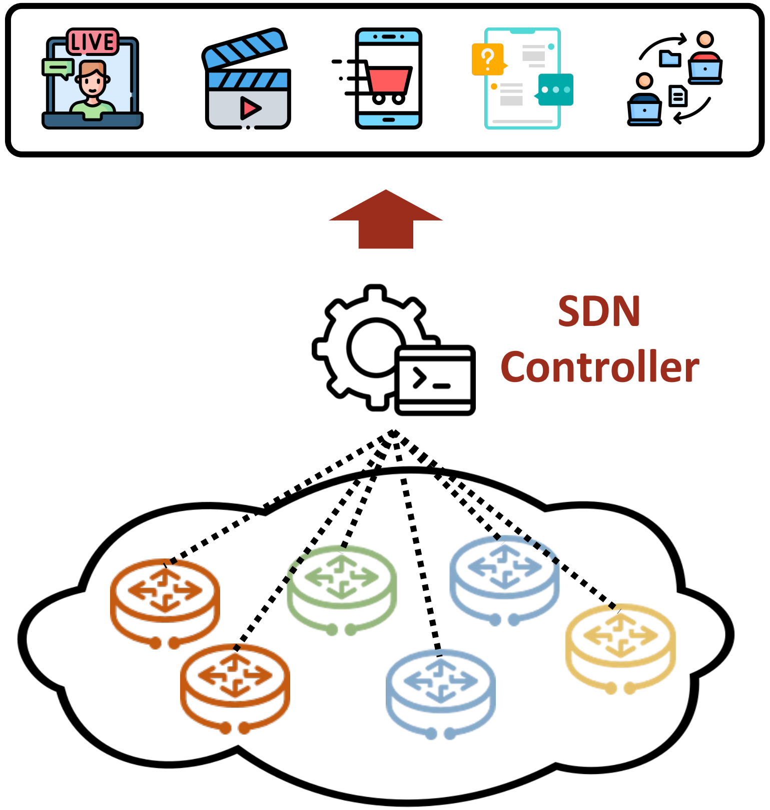


Current SNA approaches require significant human efforts

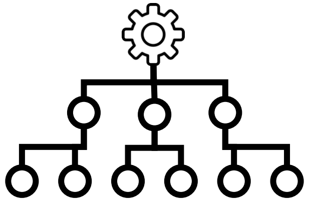


Software-defined Network Assimilation (SNA)	The process of introducing heterogeneous network devices (e.g., legacy devices & devices from a new vendor) into a centrally controlled, existing SDN network.

Current SNA approaches require significant human efforts



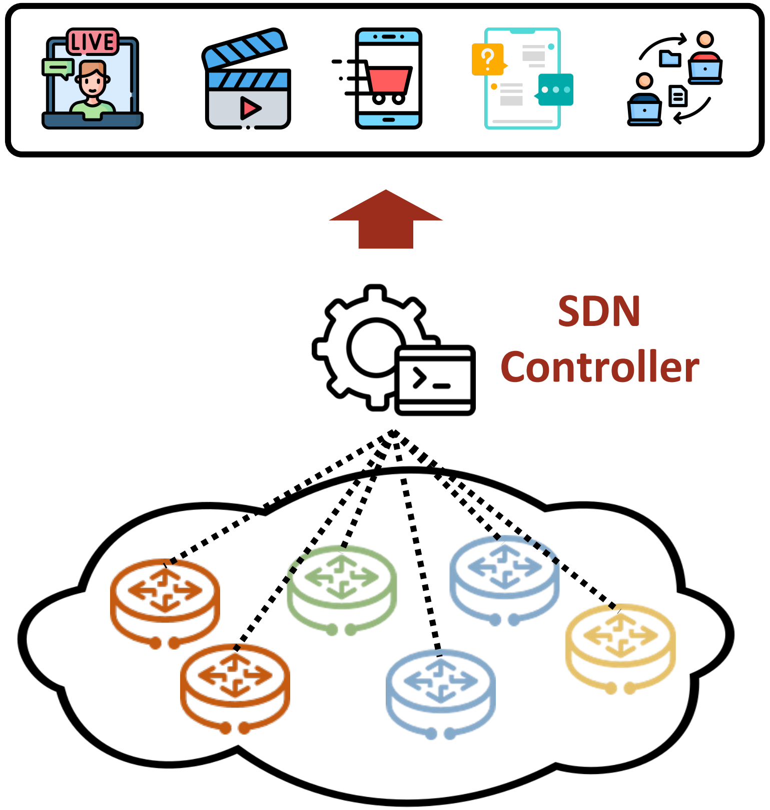
Unified Device Model (UDM)

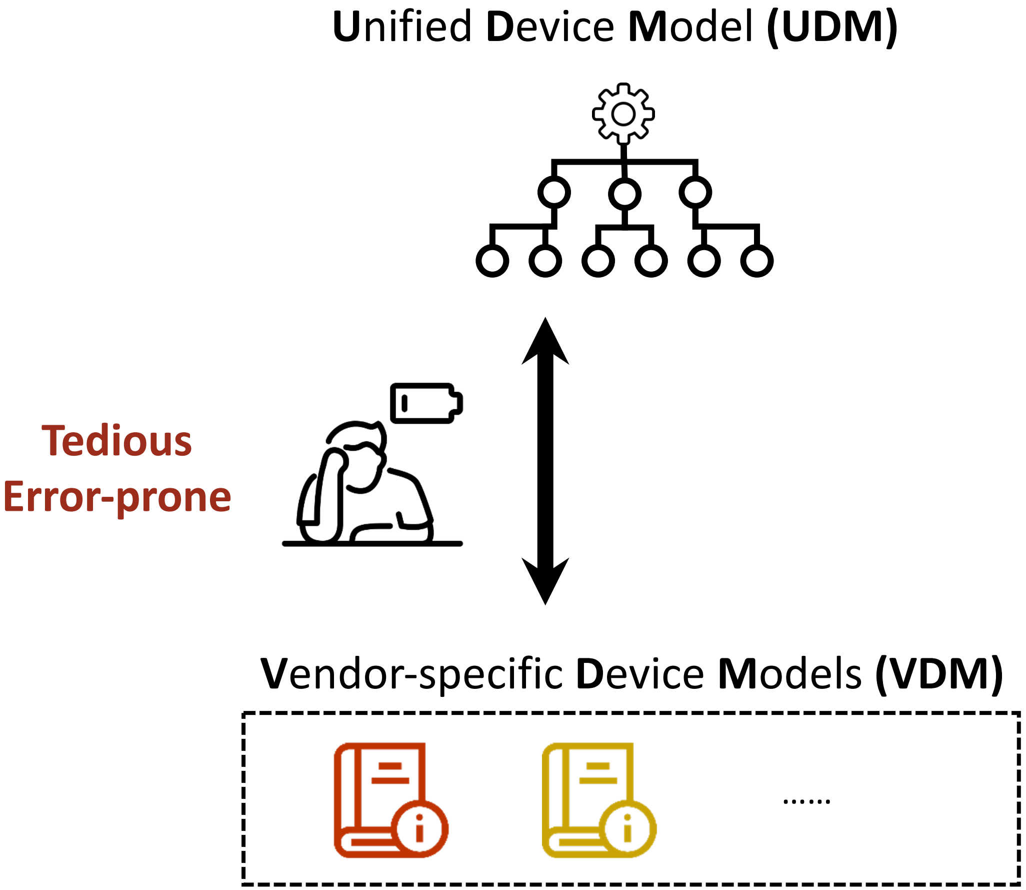


Vendor-specific Device Models (VDM)

Software-defined Network Assimilation (SNA)	The process of introducing heterogeneous network devices (e.g., legacy devices & devices from a new vendor) into a centrally controlled, existing SDN network.

Current SNA approaches require significant human efforts

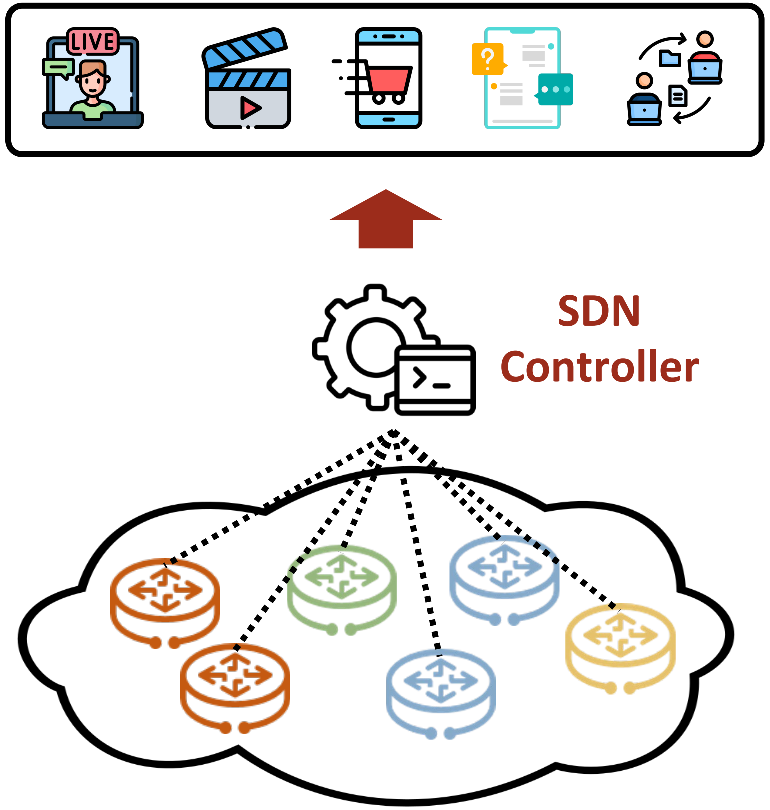



Unified **D**evice **M**odel **(UDM)**
Tedious
Error-prone
Vendor-specific **D**evice **M**odels **(VDM)**
……

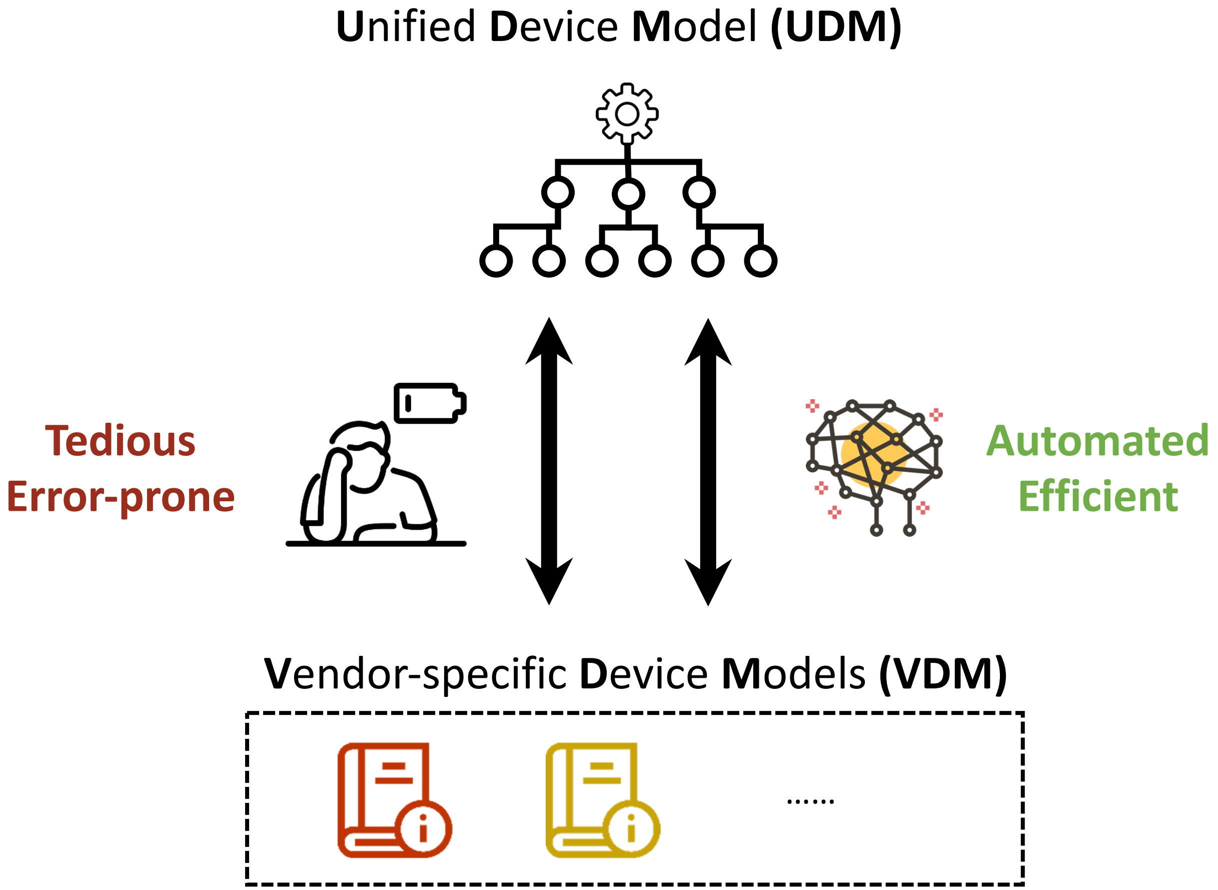
Software-defined Network Assimilation (SNA)

The process of introducing heterogeneous network devices (e.g., legacy devices & devices from a new vendor) into a centrally controlled, existing SDN network.

Current SNA approaches require significant human efforts



SDN Controller



Unified Device Model (UDM)

Tedious
Error-prone

Automated
Efficient

Vendor-specific Device Models (VDM)

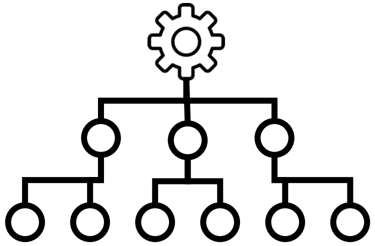
Software-defined Network Assimilation (SNA)

The process of introducing heterogeneous network devices (e.g., legacy devices & devices from a new vendor) into a centrally controlled, existing SDN network.

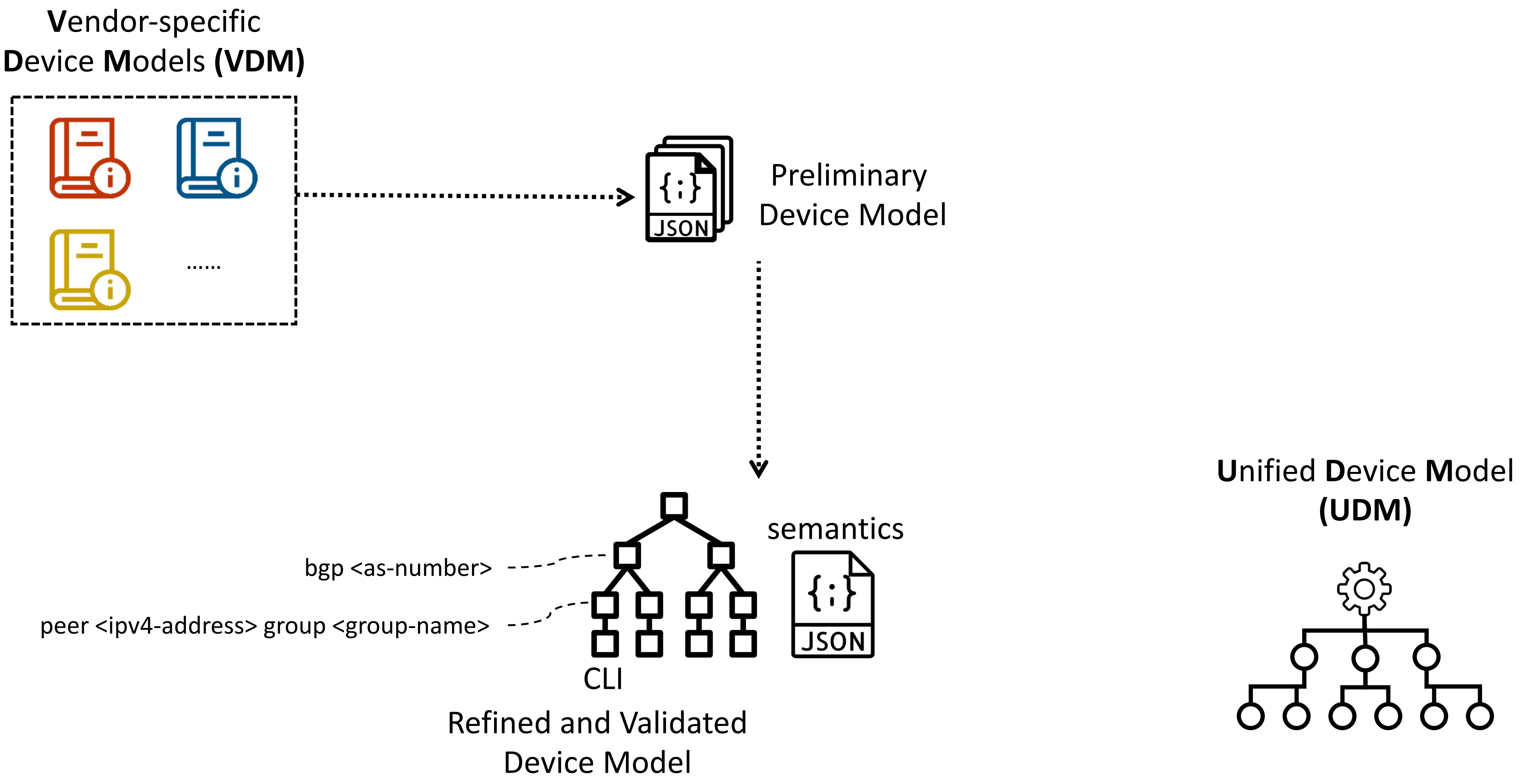
Basic Insight: Imitating the Practices of NetOps

Vendor-specific
Device **M**odels **(VDM)**

Unified **D**evice **M**odel
(UDM)



Basic Insight: Imitating the Practices of NetOps



Vendor-specific **D**evice **M**odels **(VDM)**

……

Preliminary Device Model

bgp <as-number>

peer <ipv4-address> group <group-name>

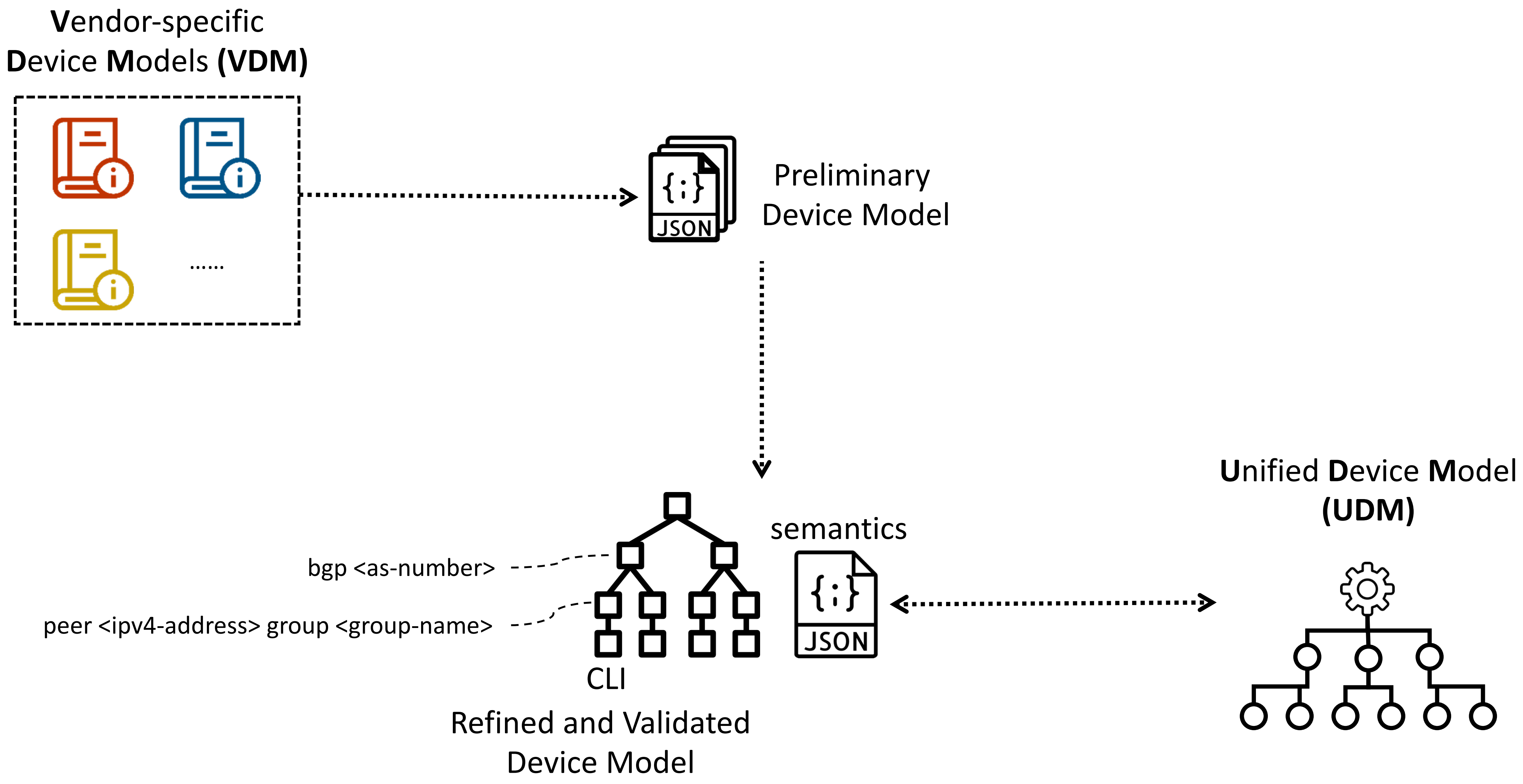
CLI

semantics

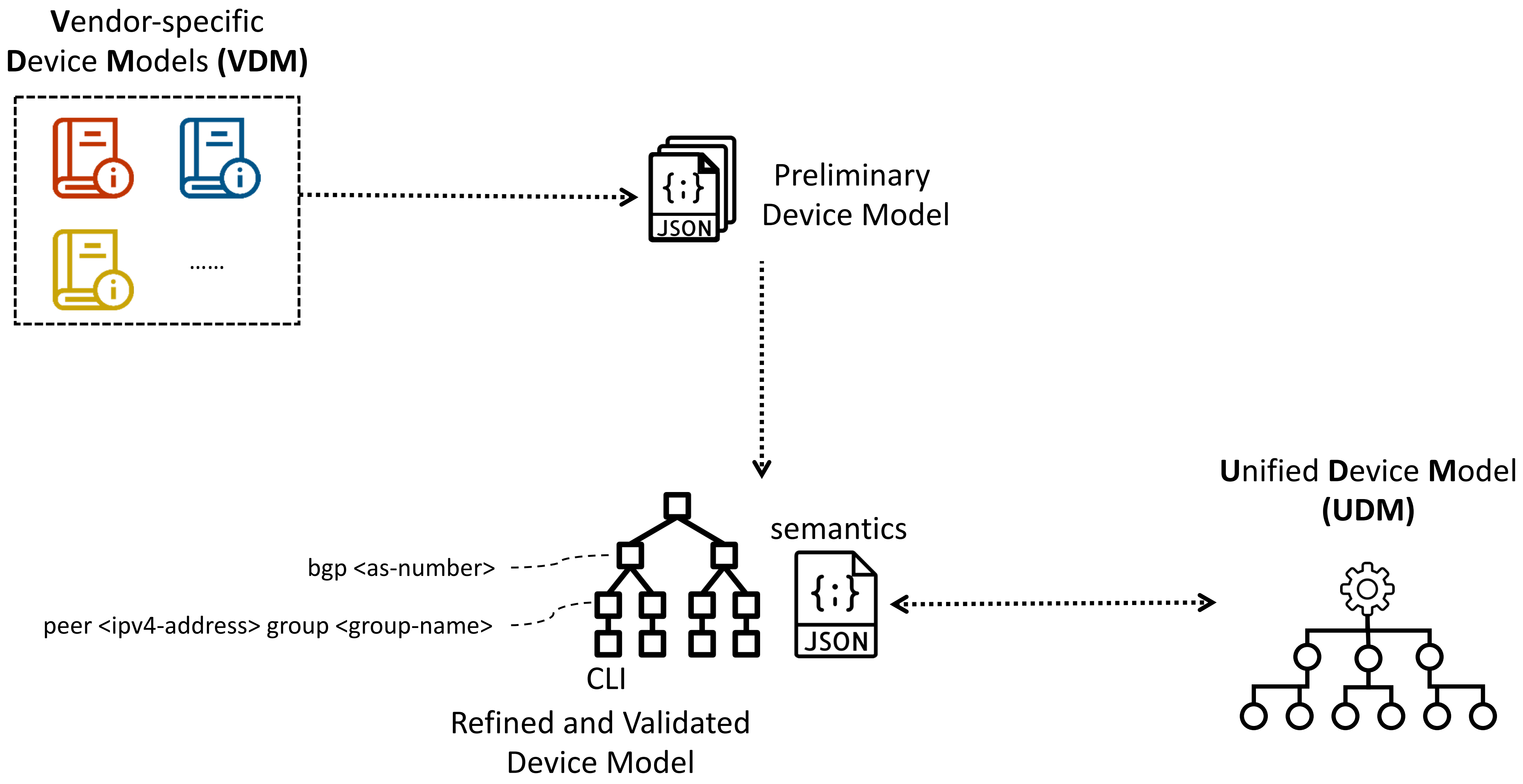
Refined and Validated Device Model

Unified **D**evice **M**odel **(UDM)**

Basic Insight: Imitating the Practices of NetOps



Challenges for SDN Network Assimilation (SNA)



Vendor-specific
Device **M**odels **(VDM)**

Preliminary
Device Model

bgp <as-number>

peer <ipv4-address> group <group-name>

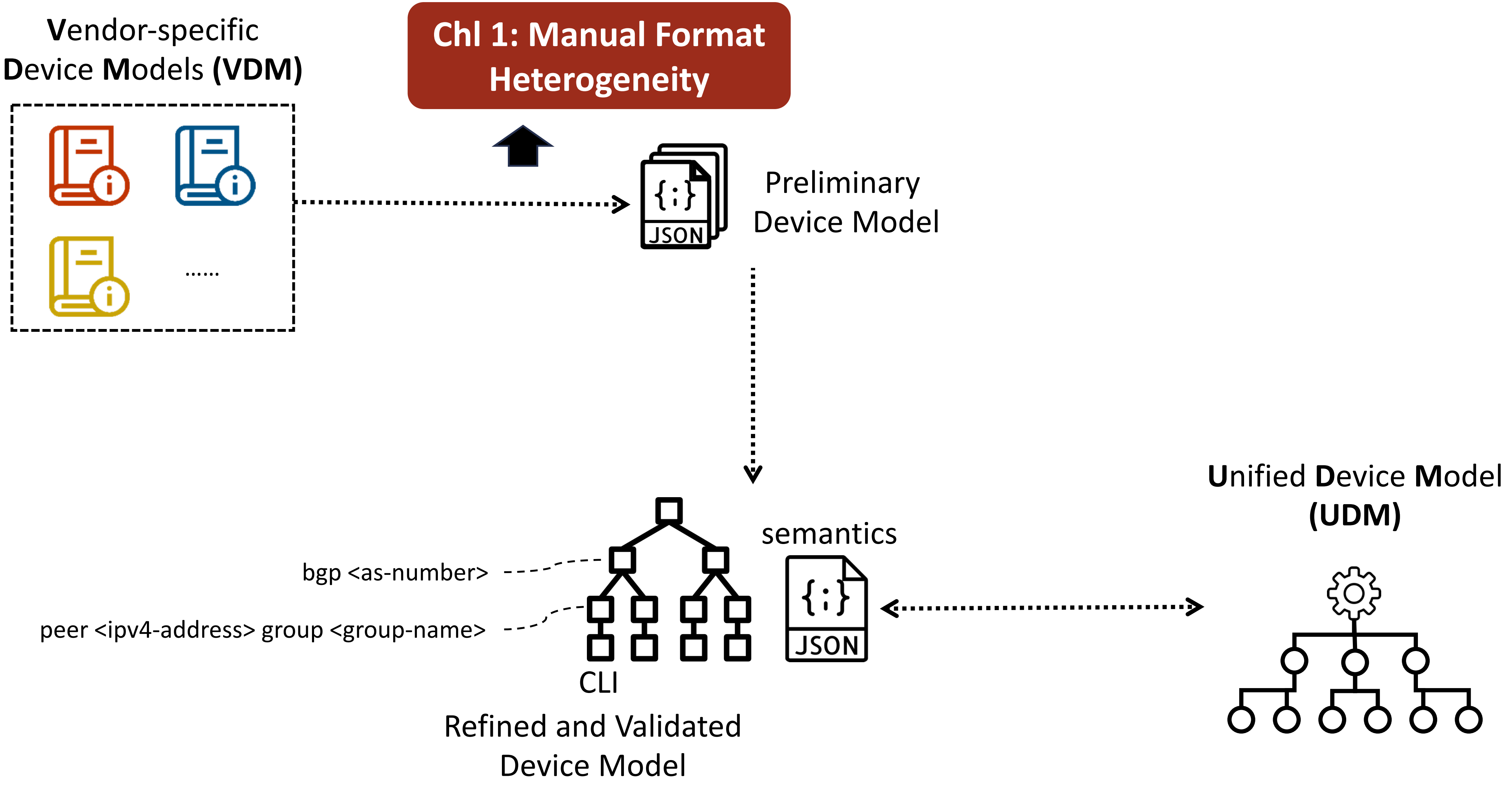
CLI

semantics

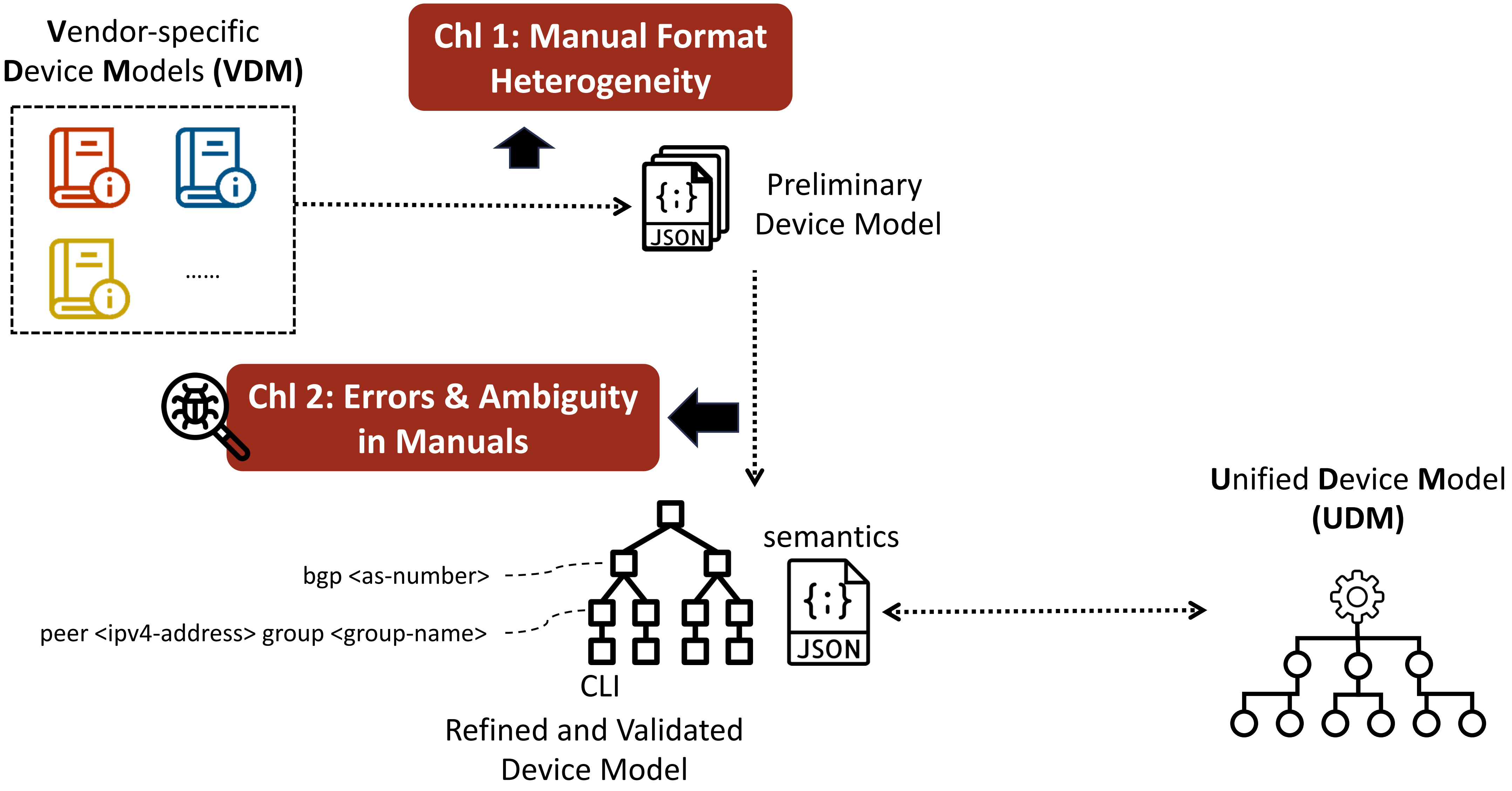
Refined and Validated
Device Model

Unified **D**evice **M**odel
(UDM)

Challenges for SDN Network Assimilation (SNA)


Vendor-specific
Device Models (VDM)
……
Chl 1: Manual Format Heterogeneity
{;}
JSON
Preliminary Device Model
bgp <as-number>
peer <ipv4-address> group <group-name>
CLI
semantics
{;}
JSON
Refined and Validated Device Model
Unified Device Model (UDM)

Challenges for SDN Network Assimilation (SNA)



Vendor-specific Device Models (VDM)

Chl 1: Manual Format Heterogeneity

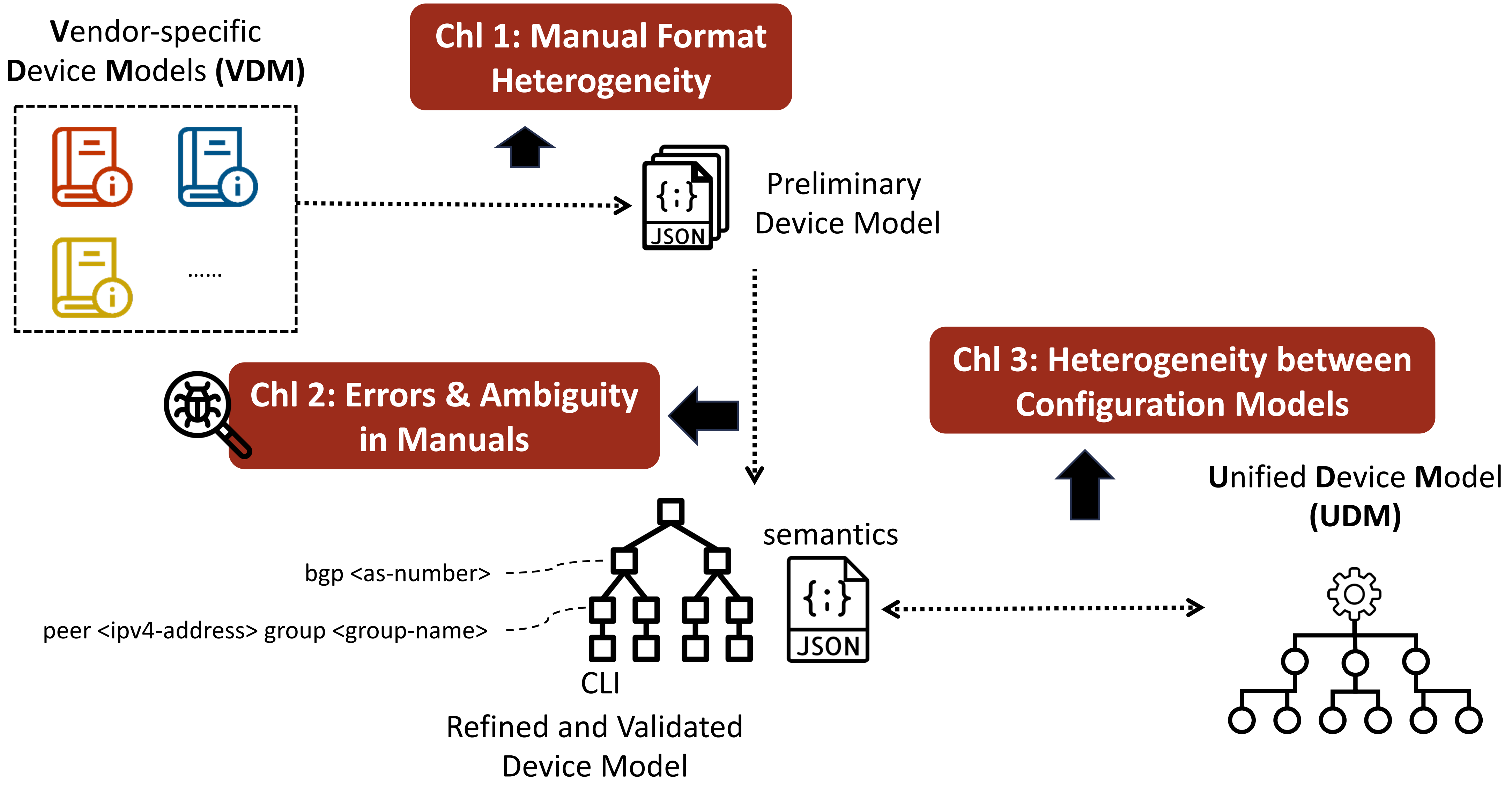
Preliminary Device Model

Chl 2: Errors & Ambiguity in Manuals

Refined and Validated Device Model

Unified Device Model (UDM)

Challenges for SDN Network Assimilation (SNA)



Vendor-specific Device Models (VDM)

Chl 1: Manual Format Heterogeneity

Preliminary Device Model

Chl 2: Errors & Ambiguity in Manuals

Chl 3: Heterogeneity between Configuration Models

Unified Device Model (UDM)

Refined and Validated Device Model

Our Key Contributions: NAssim

Our Key Contributions: NAssim

- NAssim Parser Framework (for Manual Format Heterogeneity)
  - Vendor-independent device model corpus format
  - Test-driven development procedure for reliable parsing

Our Key Contributions: NAssim

- NAssim Parser Framework (for Manual Format Heterogeneity)
  - Vendor-independent device model corpus format
  - Test-driven development procedure for reliable parsing

- NAssim Validator (for Errors & Ambiguity in Manuals)
  - Validation on command-level, inter-command-level, and snippet-level
  - Identify 184 syntactic errors and 59 ambiguities in four mainstream vendors’ manuals

Our Key Contributions: NAssim

- NAssim Parser Framework (for Manual Format Heterogeneity)
  - Vendor-independent device model corpus format
  - Test-driven development procedure for reliable parsing

- NAssim Validator (for Errors & Ambiguity in Manuals)
  - Validation on command-level, inter-command-level, and snippet-level
  - Identify 184 syntactic errors and 59 ambiguities in four mainstream vendors' manuals

- NAssim Mapper (for Heterogeneity between Configuration Models)
  - NetBERT: semantic similarity inference model for network configuration domain
  - NetBERT achieves 89% and 70% top 10 recall for mapping device models of Huawei and Nokia to a given UDM respectively

Our Key Contributions: NAssim

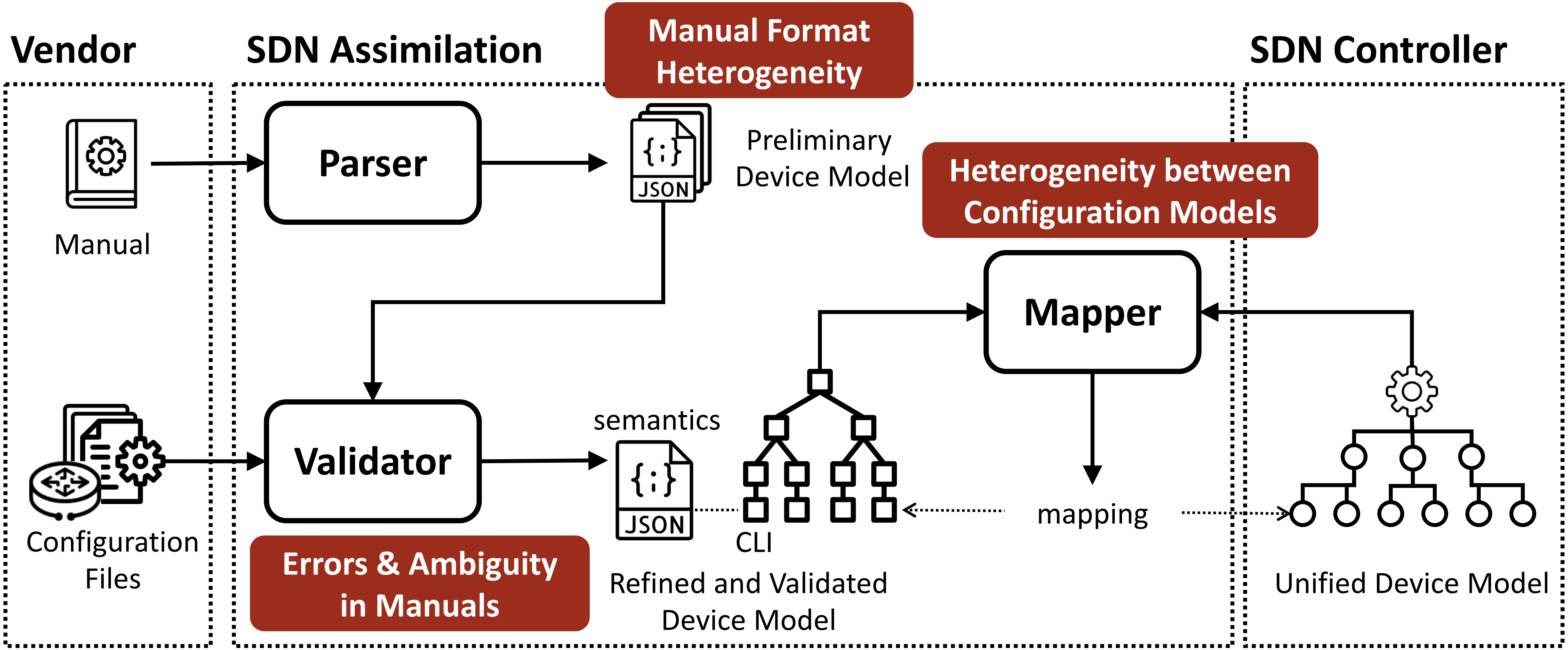
- NAssim Parser Framework (for Manual Format Heterogeneity)
    - Vendor-independent device model corpus format
    - Test-driven development procedure for reliable parsing

- NAssim Validator (for Errors & Ambiguity in Manuals)
    - Validation on command-level, inter-command-level, and snippet-level
    - Identify 184 syntactic errors and 59 ambiguities in four mainstream vendors' manuals

- NAssim Mapper (for Heterogeneity between Configuration Models)
    - NetBERT: semantic similarity inference model for network configuration domain
    - NetBERT achieves 89% and 70% top 10 recall for mapping device models of Huawei and Nokia to a given UDM respectively

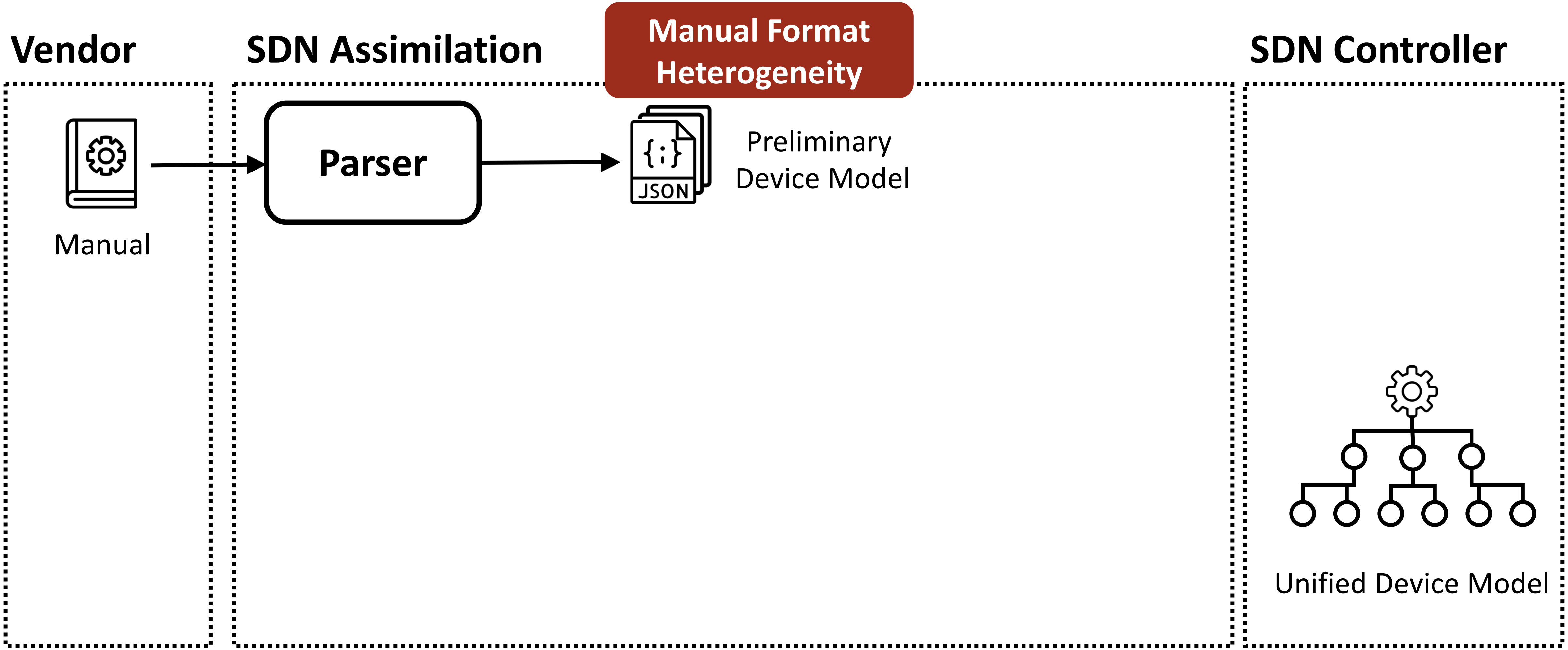
- Dataset: a validated and expert-curated dataset of parsed manual corpus for future research. (https://github.com/AmyWorkspace/nassim)

SDN Network Assimilation (NAssim) in a Nutshell



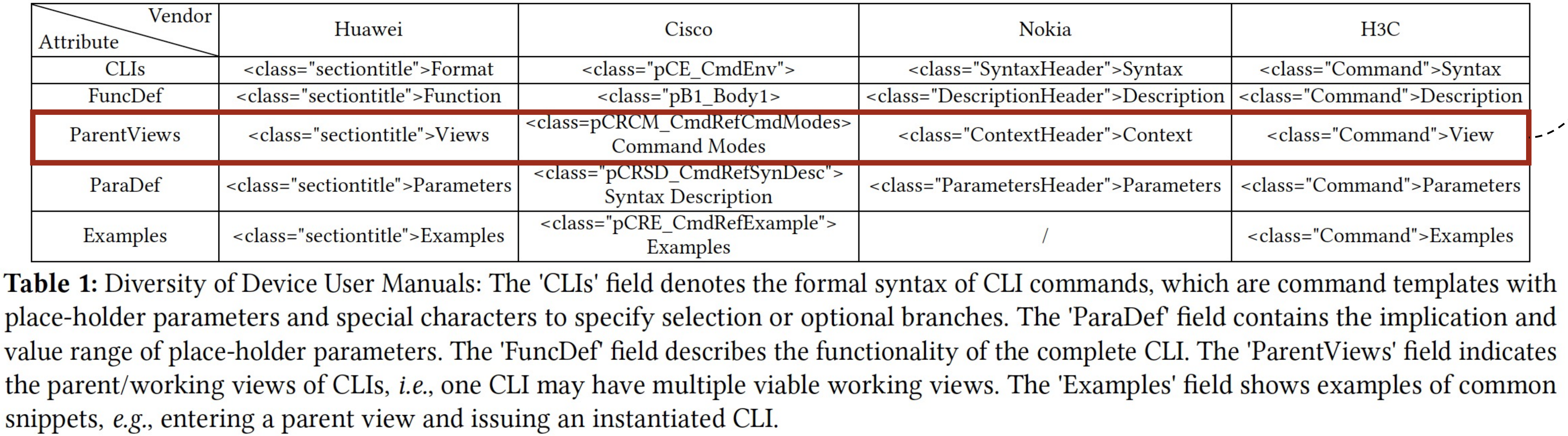
An Assistant Framework for Bridging the Last Mile Towards Centralized Network Configuration Management

SDN Network Assimilation (NAssim) in a Nutshell



NAssim Parser Framework: Key Insights

the same concept may have different names

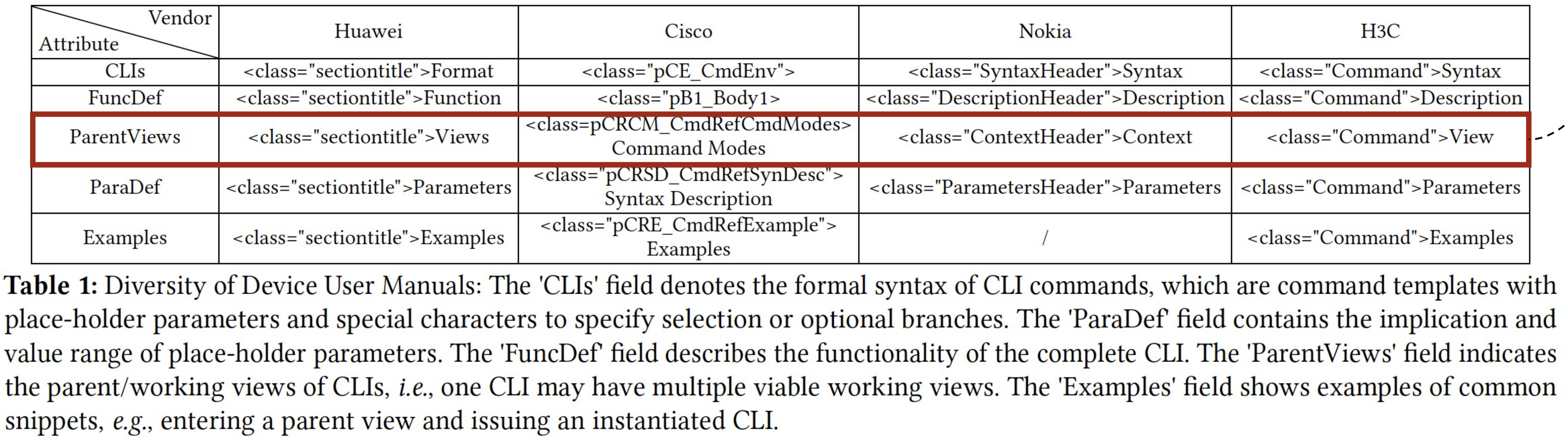


Vendor / Attribute	Huawei	Cisco	Nokia	H3C
CLIs	<class="sectiontitle">Format	<class="pCE_CmdEnv">	<class="SyntaxHeader">Syntax	<class="Command">Syntax
FuncDef	<class="sectiontitle">Function	<class="pB1_Body1>	<class="DescriptionHeader">Description	<class="Command">Description
ParentViews	<class="sectiontitle">Views	<class=pCRCM_CmdRefCmdModes> Command Modes	<class="ContextHeader">Context	<class="Command">View
ParaDef	<class="sectiontitle">Parameters	<class="pCRSD_CmdRefSynDesc"> Syntax Description	<class="ParametersHeader">Parameters	<class="Command">Parameters
Examples	<class="sectiontitle">Examples	<class="pCRE_CmdRefExample"> Examples	/	<class="Command">Examples

Table 1: Diversity of Device User Manuals: The 'CLIs' field denotes the formal syntax of CLI commands, which are command templates with place-holder parameters and special characters to specify selection or optional branches. The 'ParaDef' field contains the implication and value range of place-holder parameters. The 'FuncDef' field describes the functionality of the complete CLI. The 'ParentViews' field indicates the parent/working views of CLIs, *i.e.*, one CLI may have multiple viable working views. The 'Examples' field shows examples of common snippets, *e.g.*, entering a parent view and issuing an instantiated CLI.

NAssim Parser Framework: Key Insights

the same concept may have different names



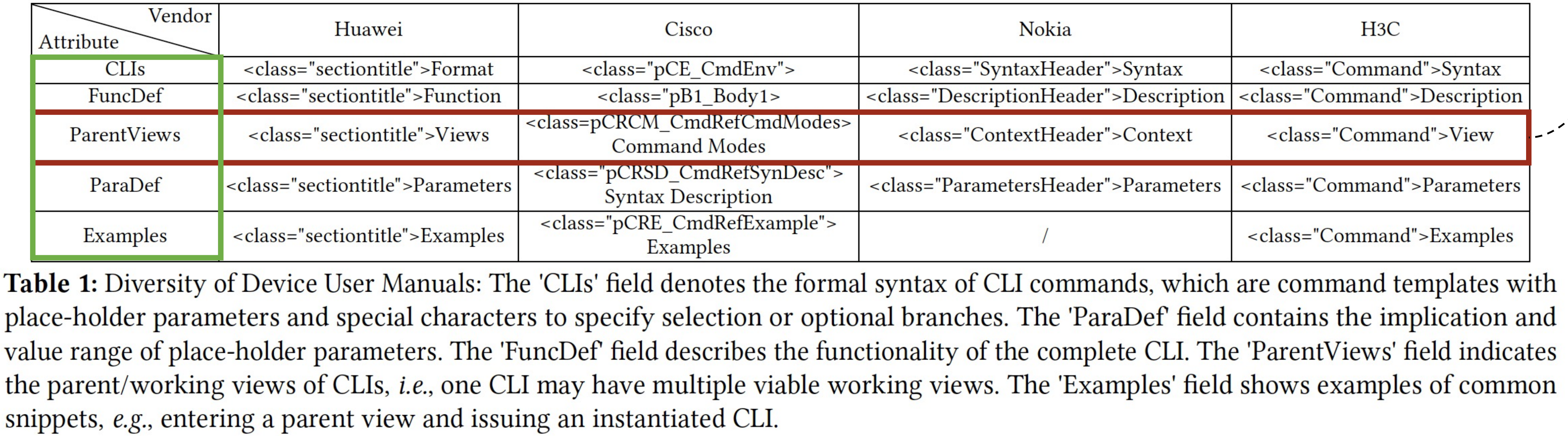
Attribute \ Vendor	Huawei	Cisco	Nokia	H3C
CLIs	<class="sectiontitle">Format	<class="pCE_CmdEnv">	<class="SyntaxHeader">Syntax	<class="Command">Syntax
FuncDef	<class="sectiontitle">Function	<class="pB1_Body1>	<class="DescriptionHeader">Description	<class="Command">Description
ParentViews	<class="sectiontitle">Views	<class=pCRCM_CmdRefCmdModes> Command Modes	<class="ContextHeader">Context	<class="Command">View
ParaDef	<class="sectiontitle">Parameters	<class="pCRSD_CmdRefSynDesc"> Syntax Description	<class="ParametersHeader">Parameters	<class="Command">Parameters
Examples	<class="sectiontitle">Examples	<class="pCRE_CmdRefExample"> Examples	/	<class="Command">Examples

Table 1: Diversity of Device User Manuals: The 'CLIs' field denotes the formal syntax of CLI commands, which are command templates with place-holder parameters and special characters to specify selection or optional branches. The 'ParaDef' field contains the implication and value range of place-holder parameters. The 'FuncDef' field describes the functionality of the complete CLI. The 'ParentViews' field indicates the parent/working views of CLIs, *i.e.*, one CLI may have multiple viable working views. The 'Examples' field shows examples of common snippets, *e.g.*, entering a parent view and issuing an instantiated CLI.

Despite diverse styles, all manuals serve the same purpose: show how to configure the devices via CLI.

NAssim Parser Framework: Key Insights

the same concept may have different names

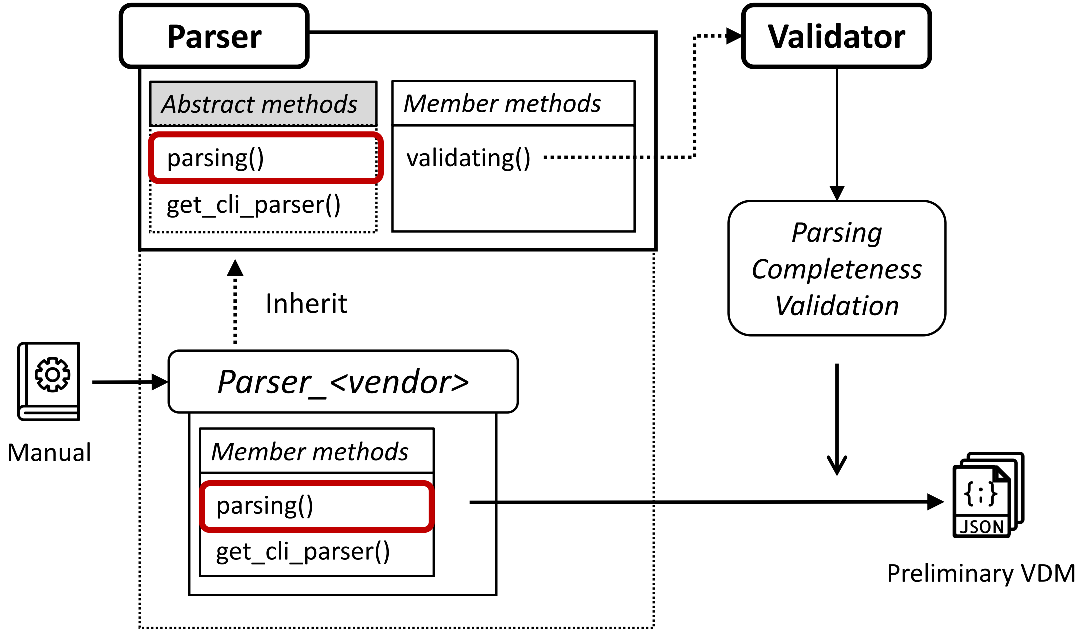


Vendor Attribute	Huawei	Cisco	Nokia	H3C
CLIs	<class="sectiontitle">Format	<class="pCE_CmdEnv">	<class="SyntaxHeader">Syntax	<class="Command">Syntax
FuncDef	<class="sectiontitle">Function	<class="pB1_Body1>	<class="DescriptionHeader">Description	<class="Command">Description
ParentViews	<class="sectiontitle">Views	<class=pCRCM_CmdRefCmdModes> Command Modes	<class="ContextHeader">Context	<class="Command">View
ParaDef	<class="sectiontitle">Parameters	<class="pCRSD_CmdRefSynDesc"> Syntax Description	<class="ParametersHeader">Parameters	<class="Command">Parameters
Examples	<class="sectiontitle">Examples	<class="pCRE_CmdRefExample"> Examples	/	<class="Command">Examples

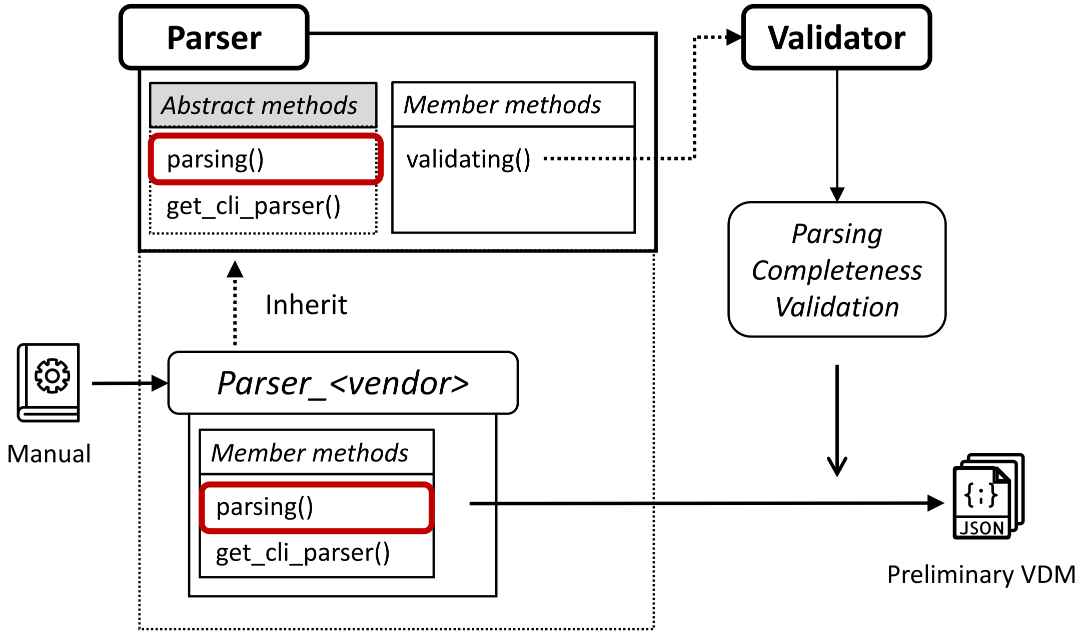
Table 1: Diversity of Device User Manuals: The 'CLIs' field denotes the formal syntax of CLI commands, which are command templates with place-holder parameters and special characters to specify selection or optional branches. The 'ParaDef' field contains the implication and value range of place-holder parameters. The 'FuncDef' field describes the functionality of the complete CLI. The 'ParentViews' field indicates the parent/working views of CLIs, *i.e.*, one CLI may have multiple viable working views. The 'Examples' field shows examples of common snippets, *e.g.*, entering a parent view and issuing an instantiated CLI.

Despite diverse styles, all manuals serve the same purpose: show how to configure the devices via CLI.

NAssim Parser Framework: Design



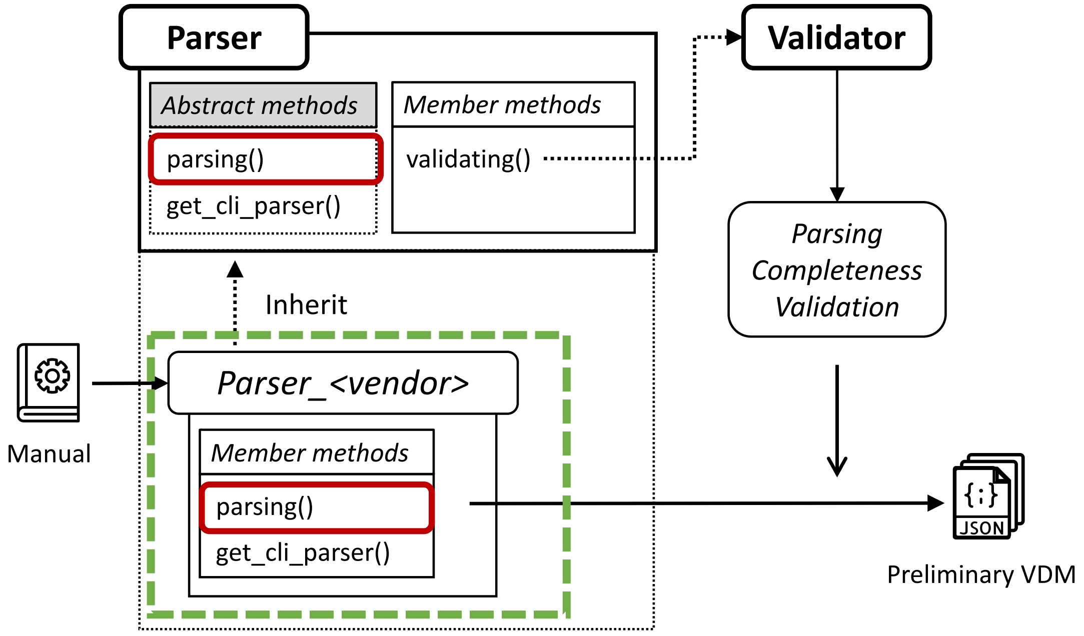
NAssim Parser Framework: Design

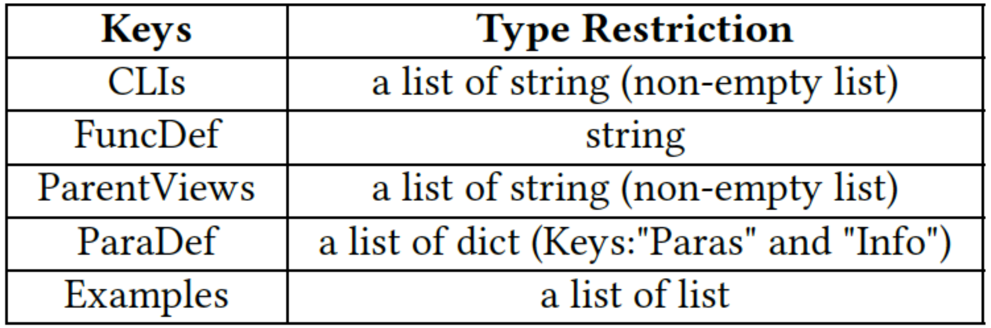


Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys:"Paras" and "Info")
Examples	a list of list

Format Definition of Vendor-Independent Corpus (JSON)

NAssim Parser Framework: Design

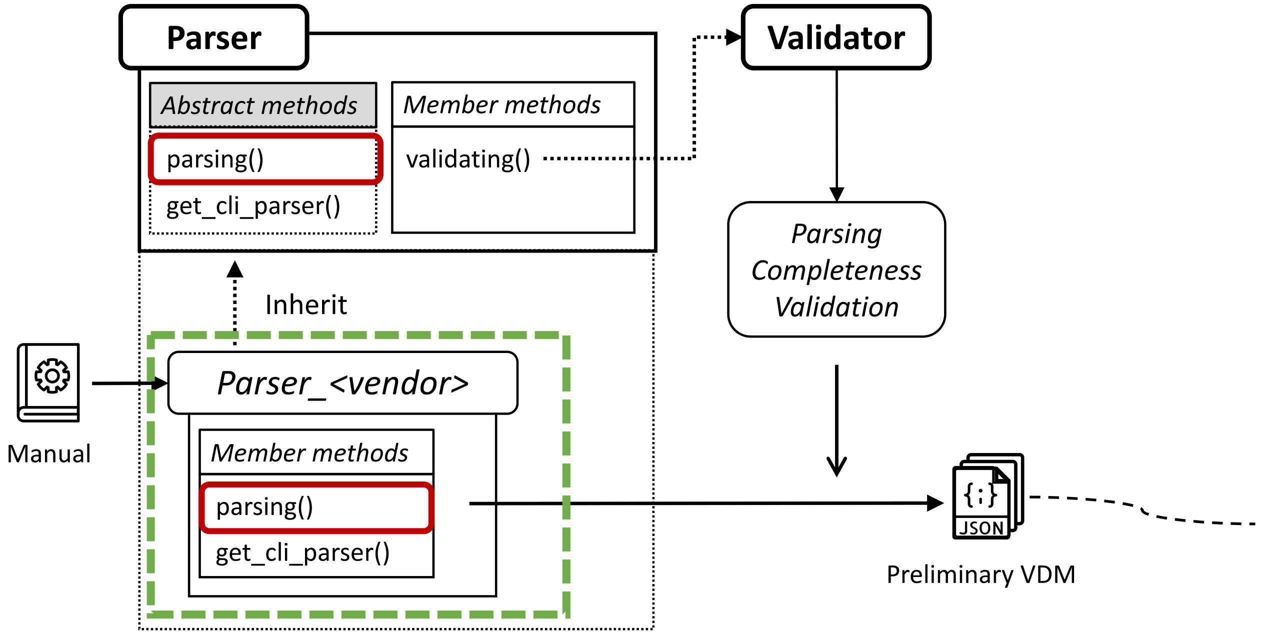




Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys:"Paras" and "Info")
Examples	a list of list

Format Definition of Vendor-Independent Corpus (JSON)

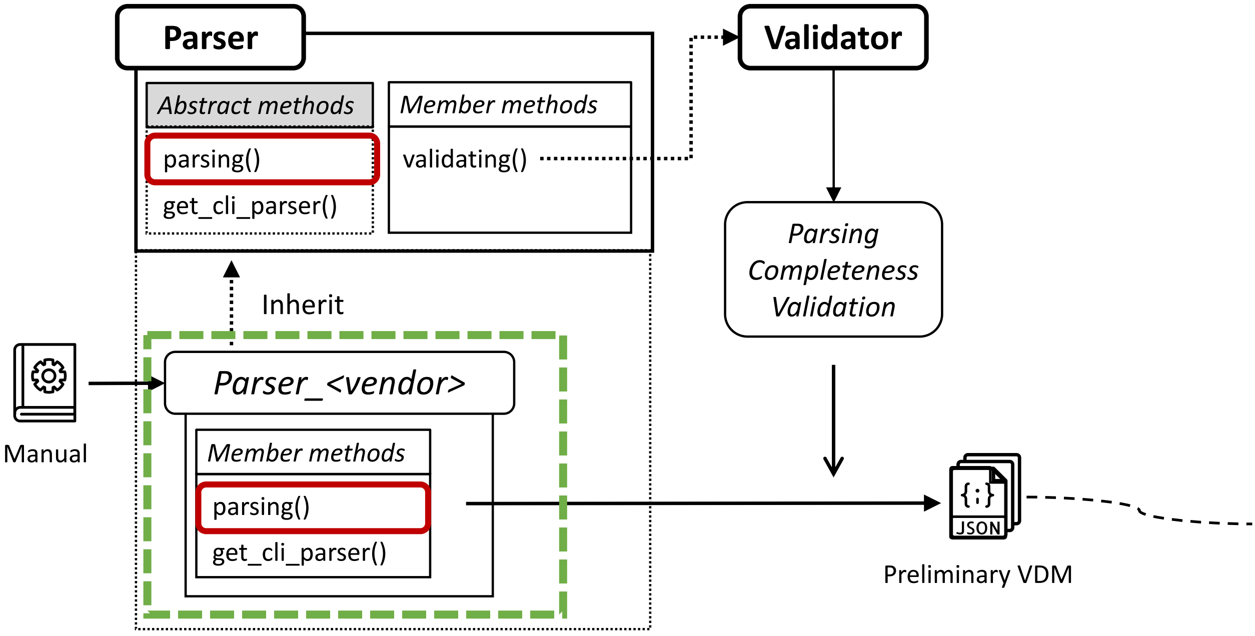
NAssim Parser Framework: Design

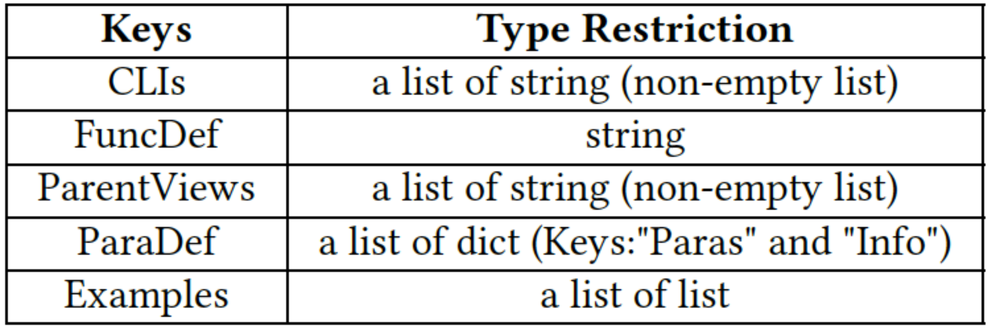


Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys:"Paras" and "Info")
Examples	a list of list

Format Definition of Vendor-Independent Corpus (JSON)

NAssim Parser Framework: Design





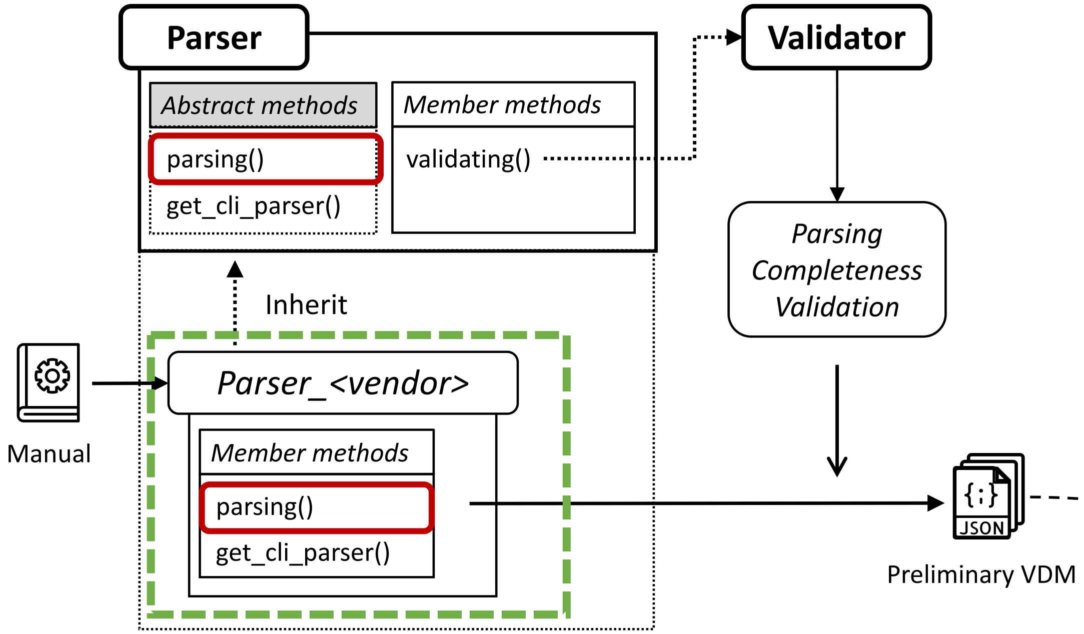
Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys:"Paras" and "Info")
Examples	a list of list

Format Definition of Vendor-Independent Corpus (JSON)

```
{
 "CLIs": [
  "peer <ipv4-address> group <group-name>",
  "undo peer <ipv4-address> group <group-name>"
 ],
 "FuncDef": "The peer group command adds a peer to a peer. The undo peer group
command deletes a peer from a peer group and all configurations of the peer. By default,
no peer group is created",
 "ParentView": ["BGP view"],
 "ParaDef": [
  {"Parameters": "ipv4-address",
   "Info": "Specifies the IPv4 address of a peer. It is in dotted decimal notation."},
  {"Parameters": "group-name",
   "Info": "Specifies the name of a peer group. The name is a string of 1 to 47 case-
sensitive characters, with spaces not supported."}
 ],
 "Examples": [
  ["<HUAWEI> system-view",
   "[~HUAWEI] bgp 100",
   "[*HUAWEI-bgp] group test internal",
   "[*HUAWEI-bgp] peer 10.1.1.1 group test"]
 ]
}

// Extended corpus with distilled ParaType info
{
 "CLIs": [
  "peer <ipv4-address> group <group-name>",
  "undo peer <ipv4-address> group <group-name>"
 ],
 "…": "… ",
 "ParaType": {
    "ipv4-address": [ipv4],
    "group-name": [string, 1, 47]}
}
```

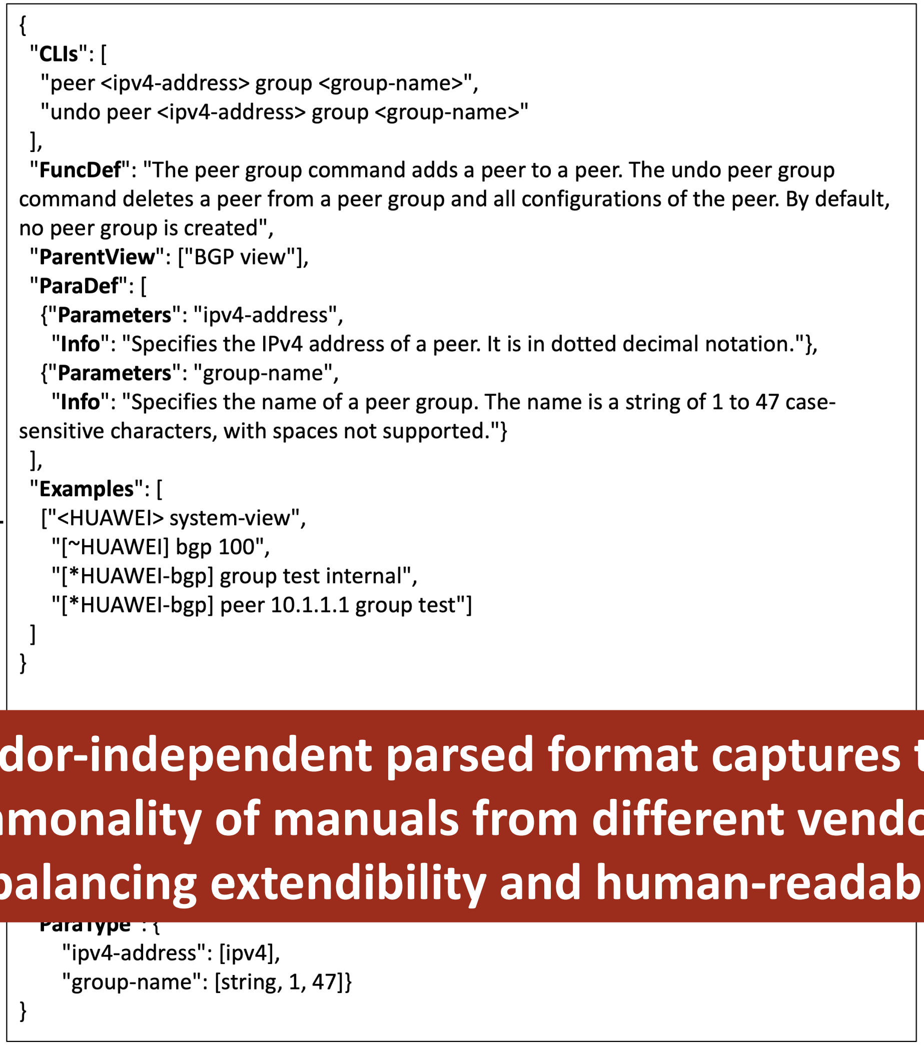
A sample of parsed VDM corpus.

NAssim Parser Framework: Design



Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys:"Paras" and "Info")
Examples	a list of list

Format Definition of Vendor-Independent Corpus (JSON)



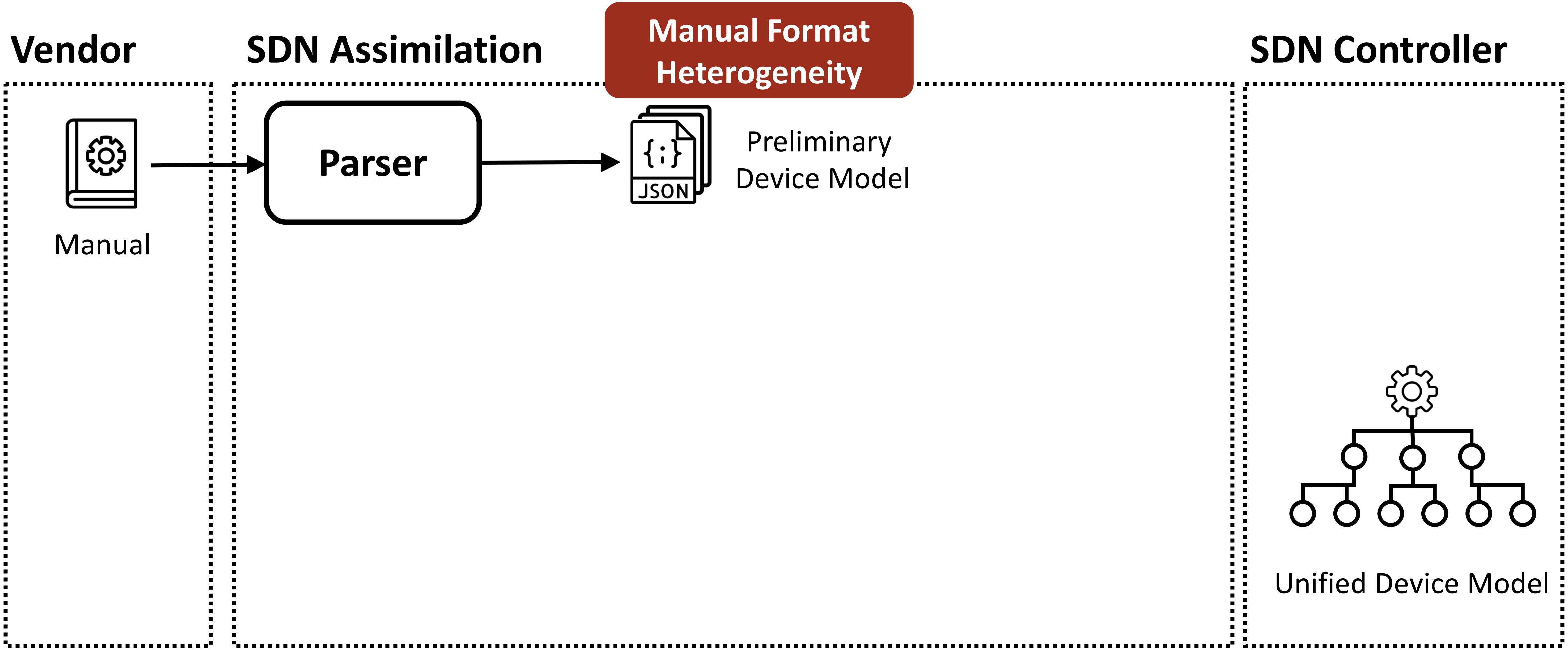
```
{
 "CLIs": [
   "peer <ipv4-address> group <group-name>",
   "undo peer <ipv4-address> group <group-name>"
 ],
 "FuncDef": "The peer group command adds a peer to a peer. The undo peer group
command deletes a peer from a peer group and all configurations of the peer. By default,
no peer group is created",
 "ParentView": ["BGP view"],
 "ParaDef": [
   {"Parameters": "ipv4-address",
     "Info": "Specifies the IPv4 address of a peer. It is in dotted decimal notation."},
   {"Parameters": "group-name",
     "Info": "Specifies the name of a peer group. The name is a string of 1 to 47 case-
sensitive characters, with spaces not supported."}
 ],
 "Examples": [
   ["<HUAWEI> system-view",
     "[~HUAWEI] bgp 100",
     "[*HUAWEI-bgp] group test internal",
     "[*HUAWEI-bgp] peer 10.1.1.1 group test"]
 ]
}
```

```
 "ParaType": {
     "ipv4-address": [ipv4],
     "group-name": [string, 1, 47]}
}
```

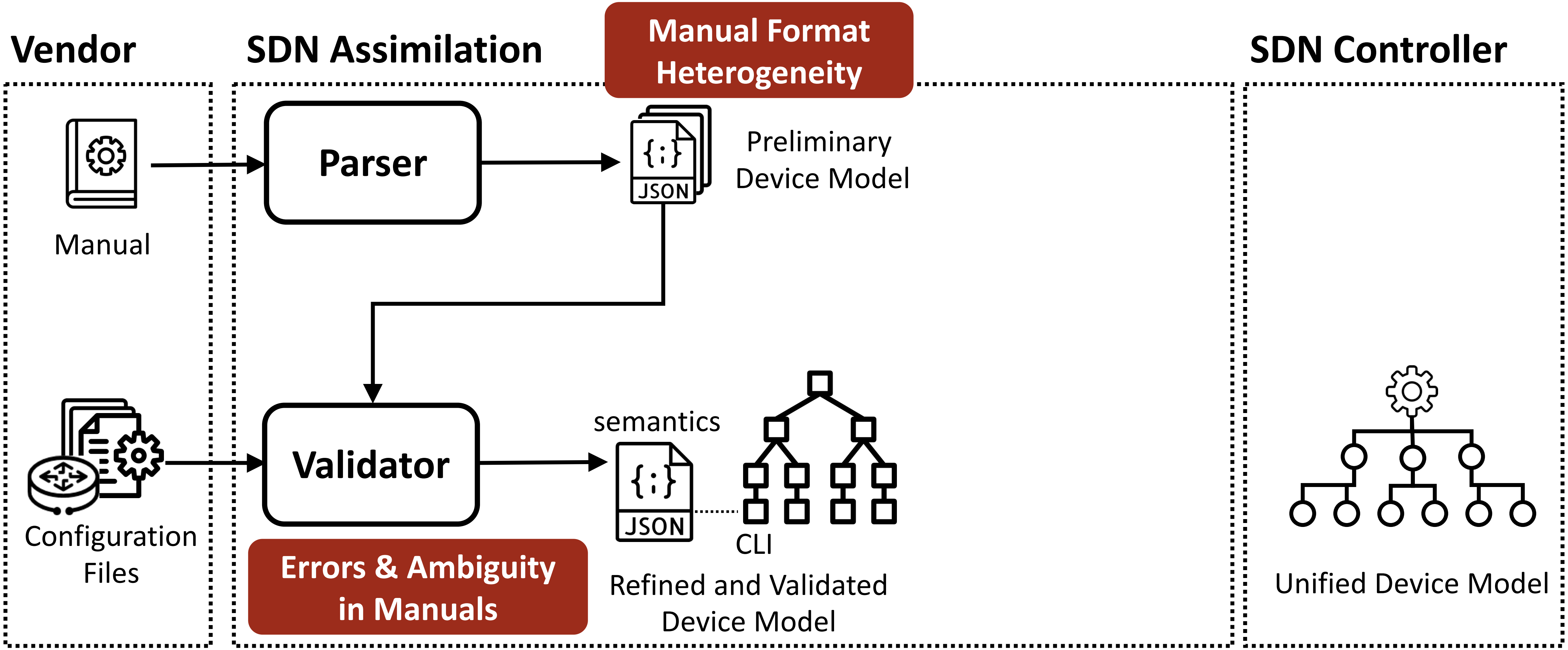
A sample of parsed VDM corpus.

Vendor-independent parsed format captures the commonality of manuals from different vendors, also balancing extendibility and human-readability.

SDN Network Assimilation (NAssim) in a Nutshell



SDN Network Assimilation (NAssim) in a Nutshell


Vendor

SDN Assimilation

SDN Controller

Manual Format Heterogeneity

Errors & Ambiguity in Manuals

NAssim Validator: Key Insights

NAssim Validator: Key Insights

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

NAssim Validator: Key Insights

Syntactic Ambiguities

An example of ambiguous CLI command template*

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

*https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html

NAssim Validator: Key Insights

Syntactic Ambiguities

```
neighbor { <ip-addr> | <ip-prefix/length> } [ remote-as { <as-num>
[ <.as-num> ] | route-map <name> }
```

unpaired left bracket

An example of ambiguous CLI command template*

Manals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

*https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html

NAssim Validator: Key Insights

Syntactic Ambiguities

```
neighbor { <ip-addr> | <ip-prefix/length> } [ remote-as { <as-num> [ <.as-num> ] | route-map <name> }
```

unpaired left bracket

Correction Option 1: removing the left bracket

```
neighbor { <ip-addr> | <ip-prefix/length> } remote-as { <as-num> [ <.as-num> ] | route-map <name> }
```

Correction Option 2: adding a right bracket after remote-as symbol

```
neighbor { <ip-addr> | <ip-prefix/length> } [ remote-as ] { <as-num> [ <.as-num> ] | route-map <name> }
```

Correction Option 3: adding a right bracket at the end of the CLI

```
neighbor { <ip-addr> | <ip-prefix/length> } [ remote-as { <as-num> [ <.as-num> ] | route-map <name> } ]
```

An example of ambiguous CLI command template*

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

*https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html

NAssim Validator: Key Insights

Syntactic Ambiguities

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }

unpaired left bracket

Correction Option 1: removing the left bracket

neighbor { <ip-addr> | <ip-prefix/length> } remote-as { <as-num> [<.as-num>] | route-map <name> }

Correction Option 2: adding a right bracket after remote-as symbol

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as] { <as-num> [<.as-num>] | route-map <name> }

Correction Option 3: adding a right bracket at the end of the CLI

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }]

An example of ambiguous CLI command template*

Hierarchy Ambiguities

An example of ambiguous view

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

**https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html*

NAssim Validator: Key Insights

Syntactic Ambiguities

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }

unpaired left bracket

Correction Option 1: removing the left bracket

neighbor { <ip-addr> | <ip-prefix/length> } remote-as { <as-num> [<.as-num>] | route-map <name> }

Correction Option 2: adding a right bracket after remote-as symbol

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as] { <as-num> [<.as-num>] | route-map <name> }

Correction Option 3: adding a right bracket at the end of the CLI

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }]

An example of ambiguous CLI command template*

Hierarchy Ambiguities

```
{
  "CLIs": [
    "import-source acl { acl-number | acl-name }",
    "undo import-source"
  ],
   "…": "… ",
  "ParentView": ["VPN instance MSDP view", "MSDP view of a public network instance"],
    "…": "… ",
  "Examples": [
    ["<HUAWEI> system-view",
      "[~HUAWEI] acl number 3101",
      "…",
      "[*HUAWEI-acl4-advance-3101] quit",
      "[*HUAWEI] multicast routing-enable",
      "[*HUAWEI] msdp",
      "[*HUAWEI-msdp] import-source acl 3101"]
  ]
}
```

An example of ambiguous view

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

*https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html

NAssim Validator: Key Insights

Syntactic Ambiguities

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }

unpaired left bracket

Correction Option 1: removing the left bracket

neighbor { <ip-addr> | <ip-prefix/length> } remote-as { <as-num> [<.as-num>] | route-map <name> }

Correction Option 2: adding a right bracket after remote-as symbol

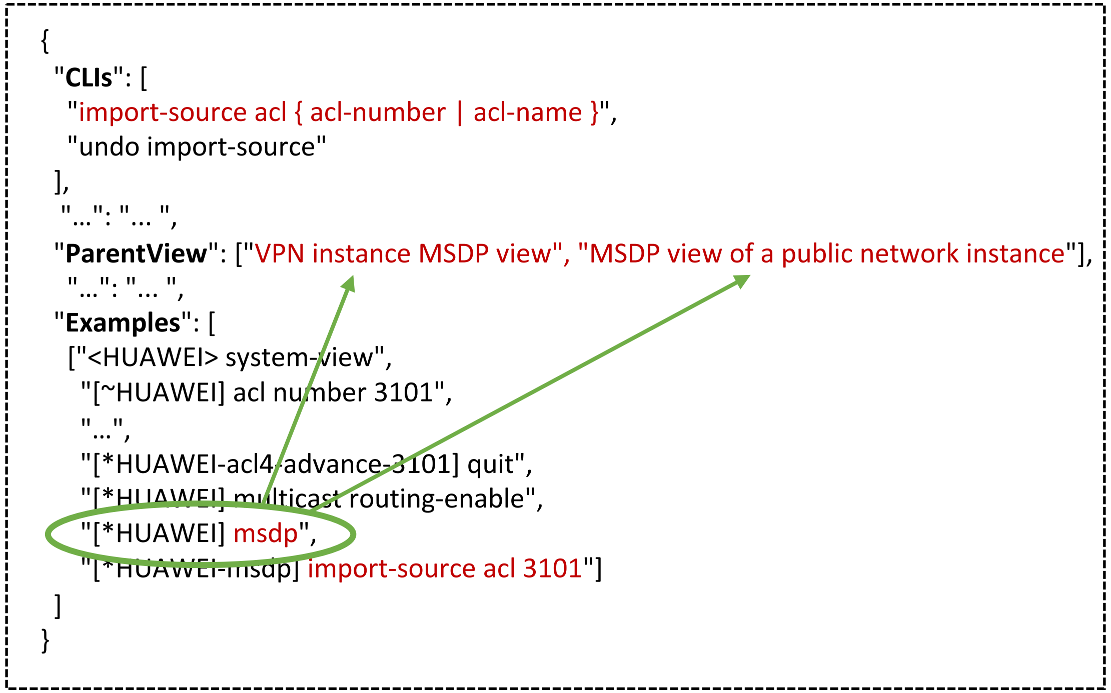
neighbor { <ip-addr> | <ip-prefix/length> } [remote-as] { <as-num> [<.as-num>] | route-map <name> }

Correction Option 3: adding a right bracket at the end of the CLI

neighbor { <ip-addr> | <ip-prefix/length> } [remote-as { <as-num> [<.as-num>] | route-map <name> }]

An example of ambiguous CLI command template*

Hierarchy Ambiguities



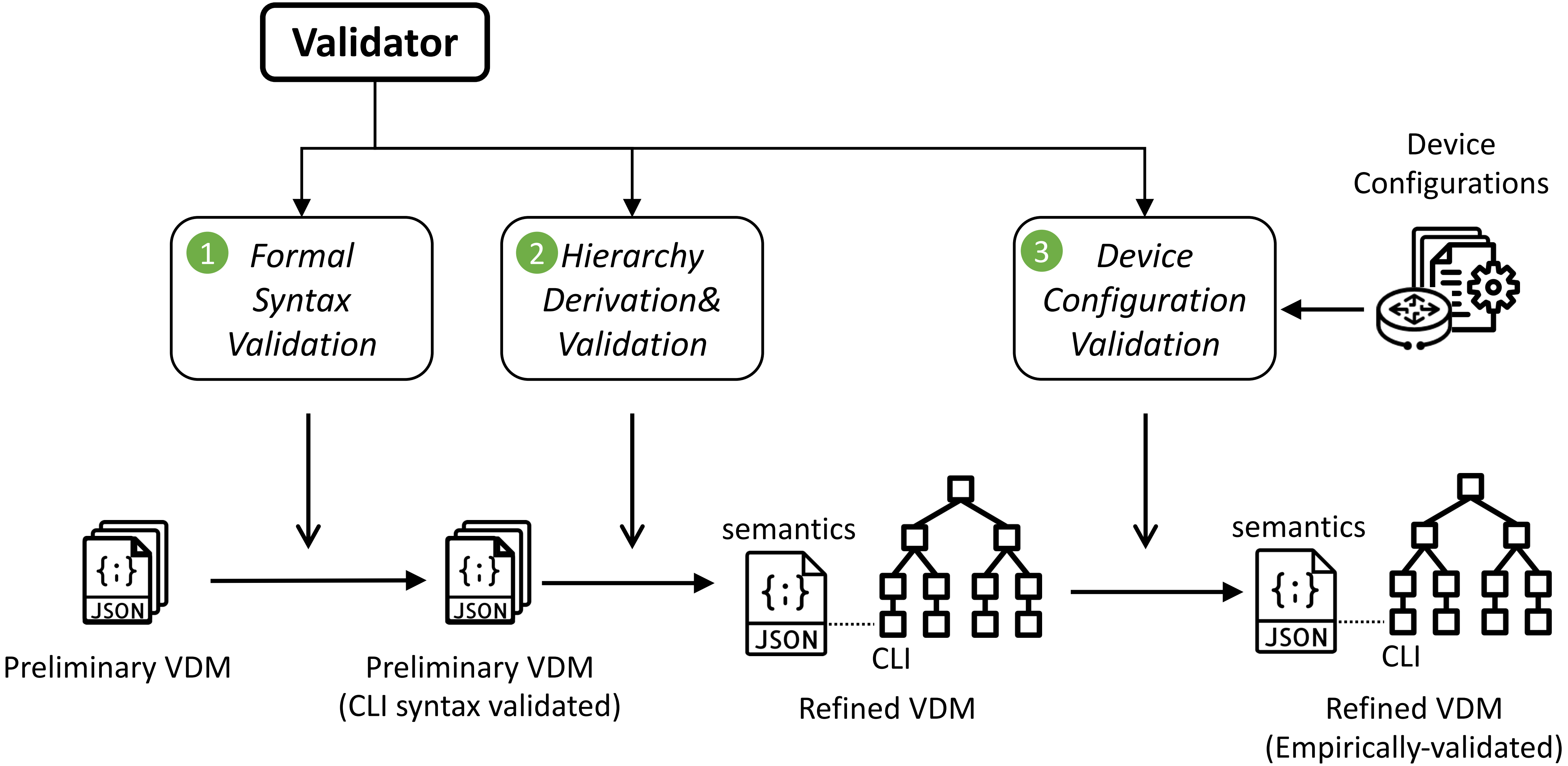
```
{
 "CLIs": [
  "import-source acl { acl-number | acl-name }",
  "undo import-source"
 ],
  "…": "… ",
 "ParentView": ["VPN instance MSDP view", "MSDP view of a public network instance"],
  "…": "… ",
 "Examples": [
  ["<HUAWEI> system-view",
   "[~HUAWEI] acl number 3101",
   "…",
   "[*HUAWEI-acl4-advance-3101] quit",
   "[*HUAWEI] multicast routing-enable",
   "[*HUAWEI] msdp",
   "[*HUAWEI-msdp] import-source acl 3101"]
 ]
}
```

An example of ambiguous view

Manuals are not fully reliable. A rigorous validation scheme needs to catch inevitable errors and ambiguities in human-written manuals.

**https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/command/reference/unicast/n5500-ucast-cr/n5500-bgp_cmds_n.html*

NAssim Validator: Design



Validator

1 *Formal Syntax Validation*

2 *Hierarchy Derivation& Validation*

3 *Device Configuration Validation*

Device Configurations

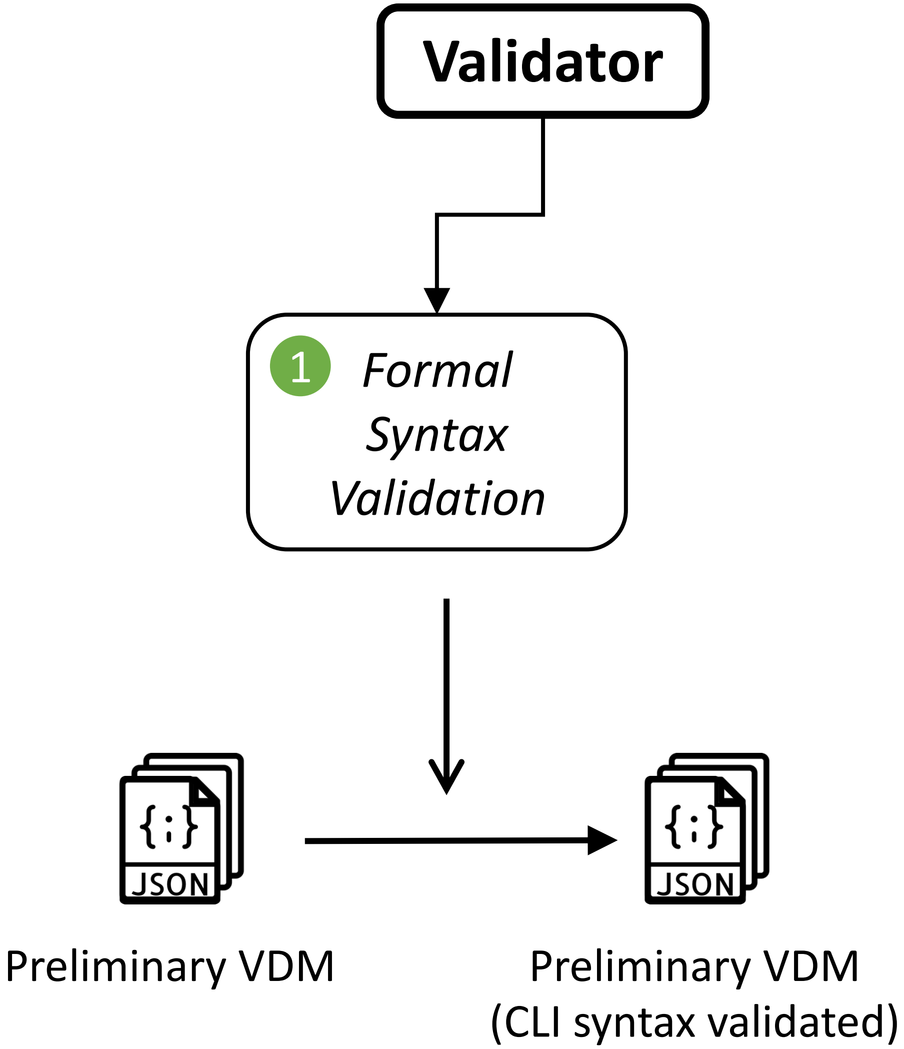
Preliminary VDM

Preliminary VDM (CLI syntax validated)

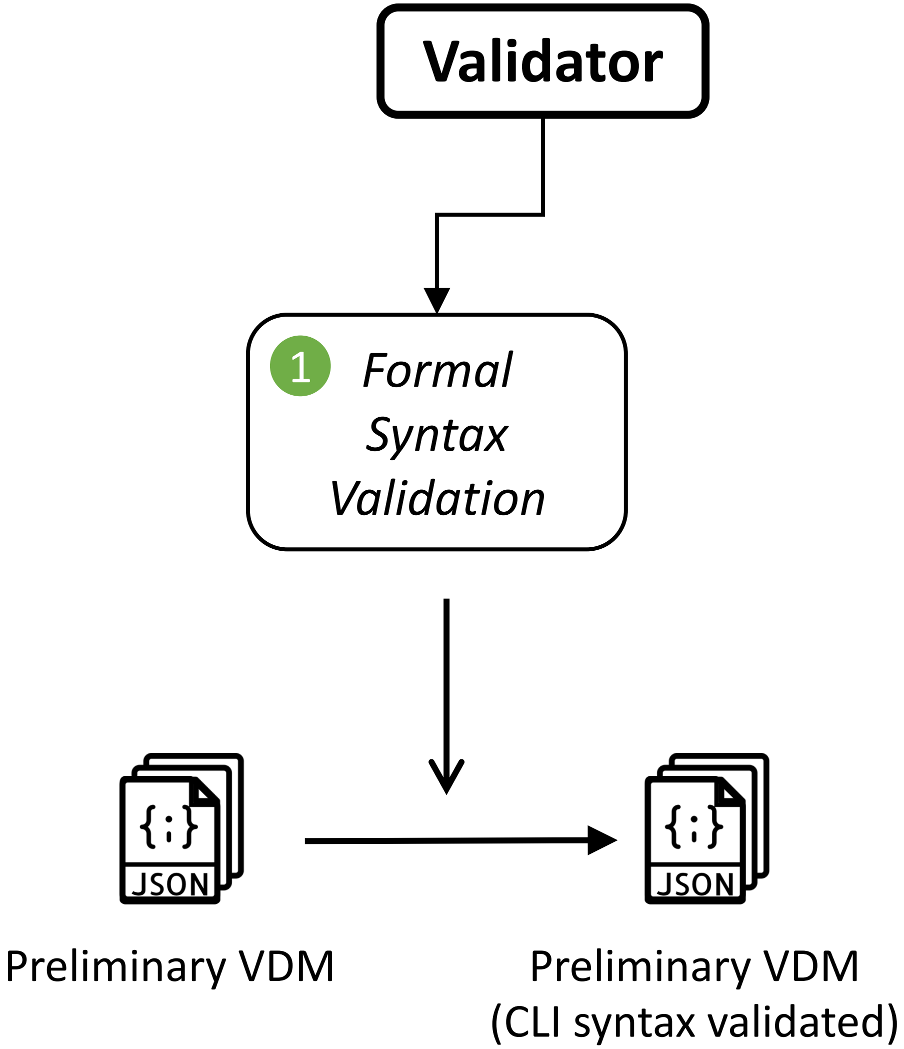
Refined VDM

Refined VDM (Empirically-validated)

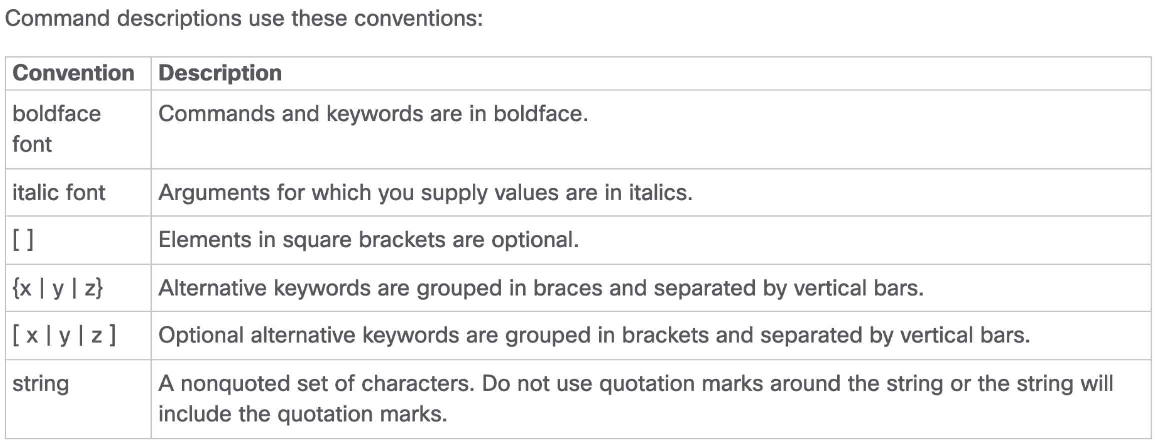
NAssim Validator: Design



NAssim Validator: Design



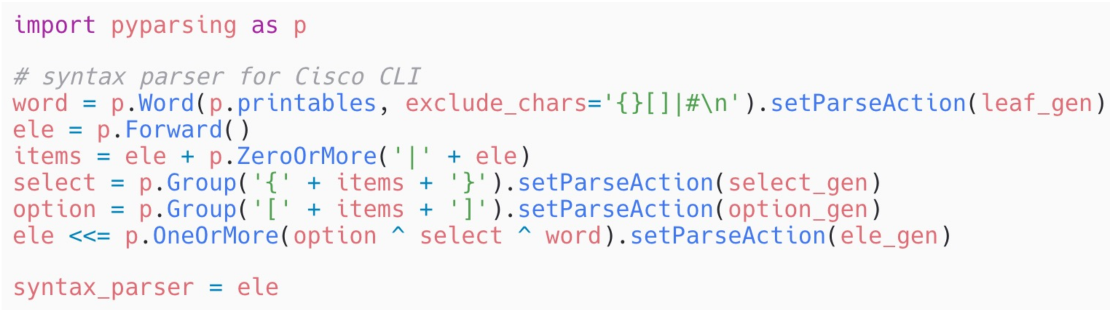
Express command conventions into their equivalent Backus Normal Form (BNF) for formal validation.



Command descriptions use these conventions:

Convention	Description
boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
[ ]	Elements in square brackets are optional.
{x | y | z}	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

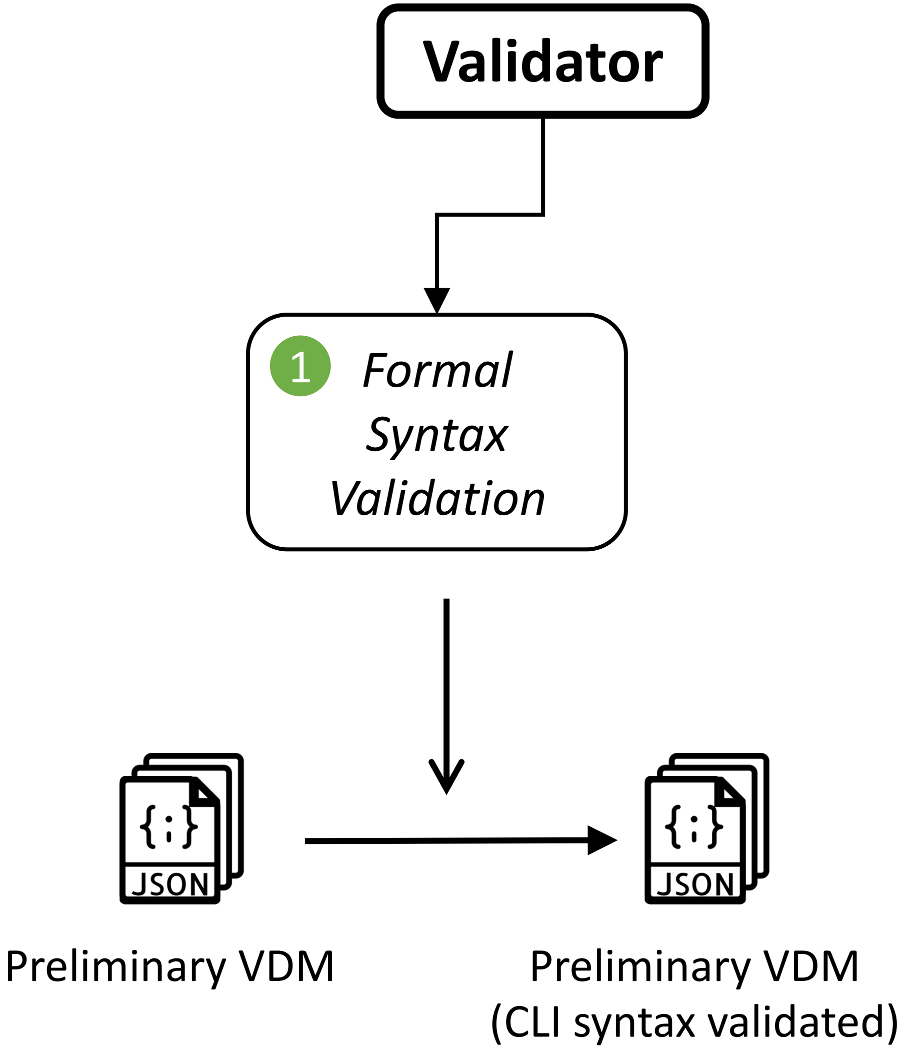
Command convention of Cisco manuals



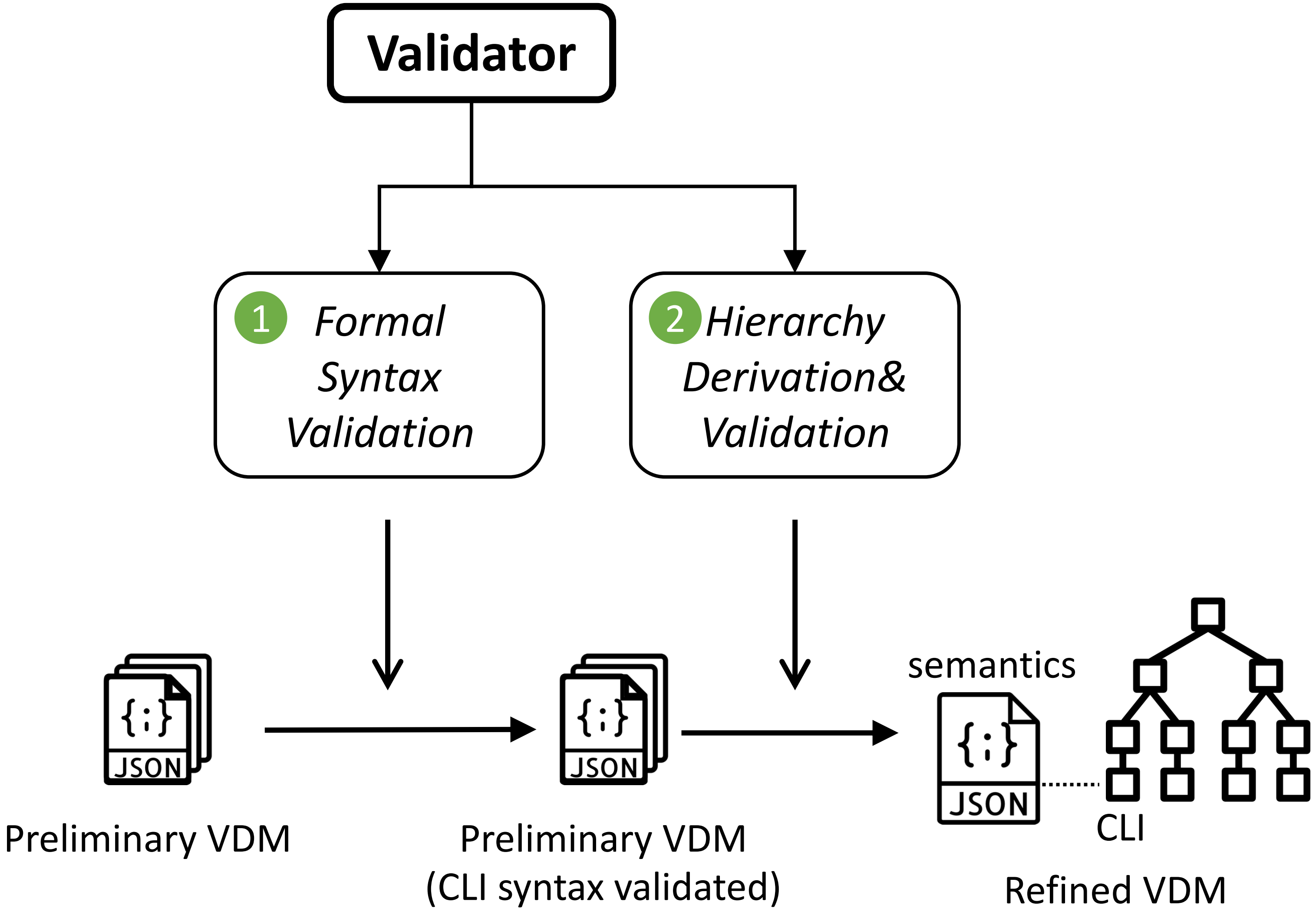
```
import pyparsing as p

# syntax parser for Cisco CLI
word = p.Word(p.printables, exclude_chars='{}[]|#\n').setParseAction(leaf_gen)
ele = p.Forward()
items = ele + p.ZeroOrMore('|' + ele)
select = p.Group('{' + items + '}').setParseAction(select_gen)
option = p.Group('[' + items + ']').setParseAction(option_gen)
ele <<= p.OneOrMore(option ^ select ^ word).setParseAction(ele_gen)

syntax_parser = ele
```

Code snippet for syntax parser generation

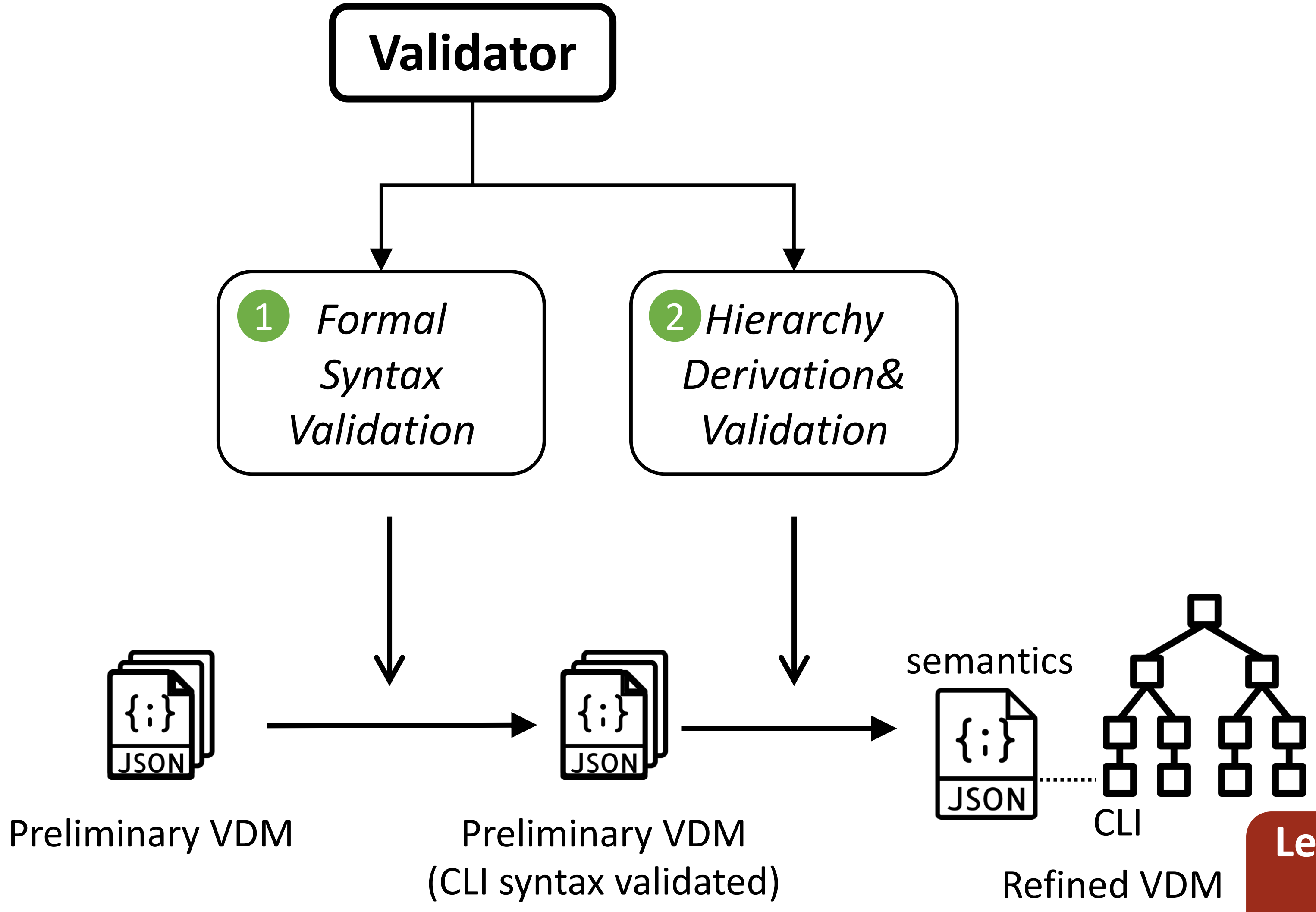
NAssim Validator: Design



NAssim Validator: Design

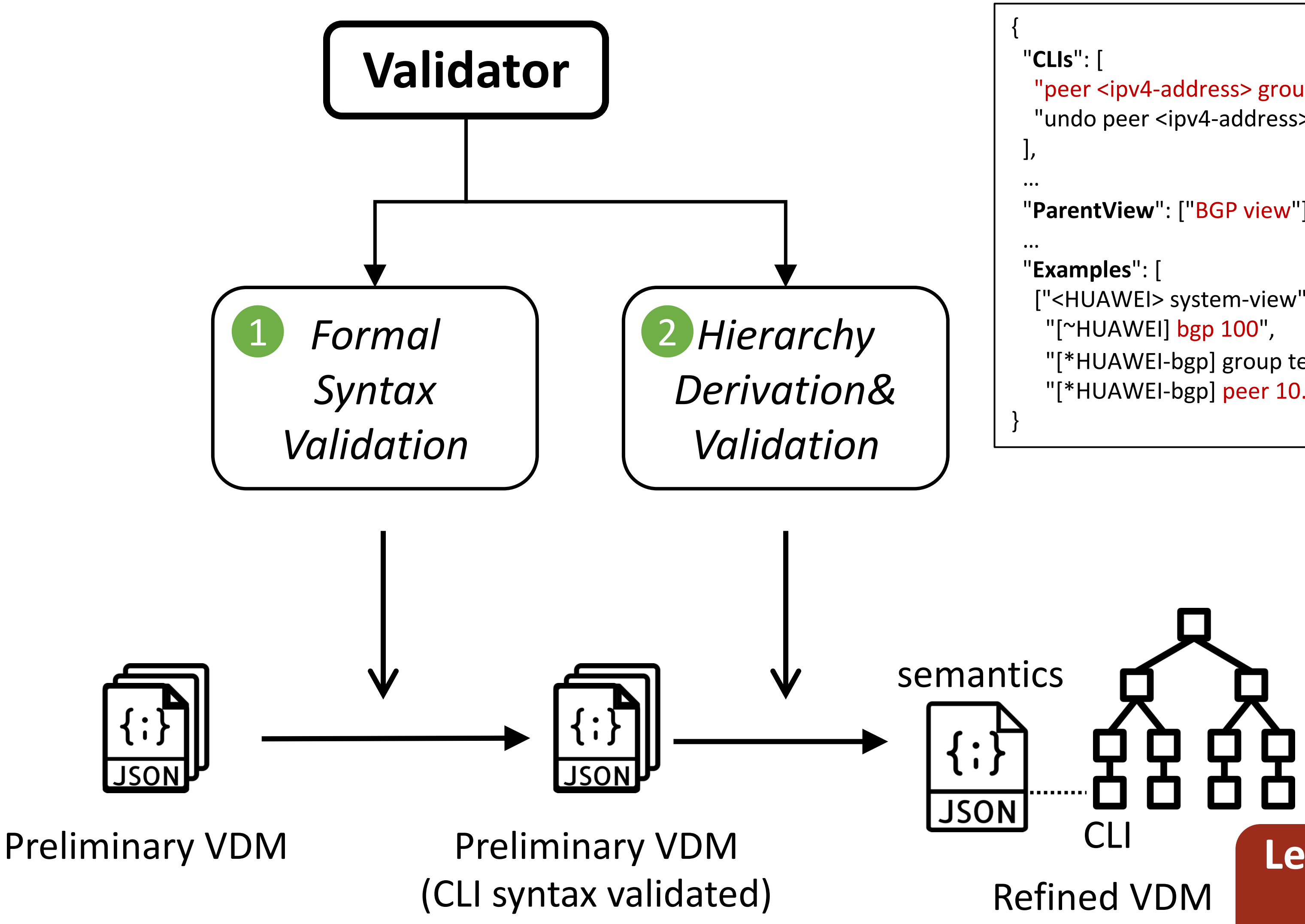


NAssim Validator: Design



Leverage example snippets to derive the relationship between CLI commands, i.e., CLI model hierarchy.

NAssim Validator: Design

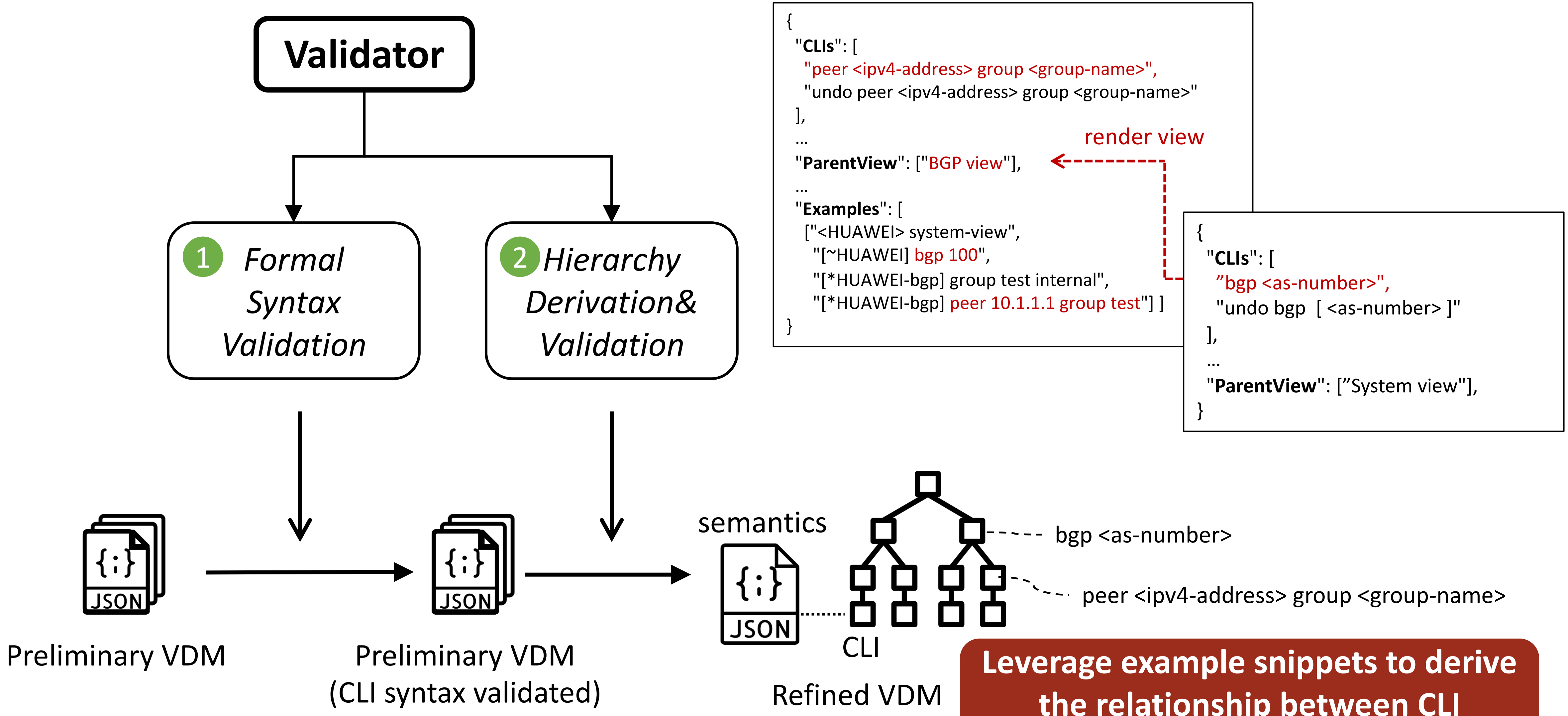


```
{
 "CLIs": [
  "peer <ipv4-address> group <group-name>",
  "undo peer <ipv4-address> group <group-name>"
 ],
 …
 "ParentView": ["BGP view"],
 …
 "Examples": [
  ["<HUAWEI> system-view",
   "[~HUAWEI] bgp 100",
   "[*HUAWEI-bgp] group test internal",
   "[*HUAWEI-bgp] peer 10.1.1.1 group test"] ]
}
```

```
{
 "CLIs": [
  ”bgp <as-number>",
  "undo bgp  [ <as-number> ]"
 ],
 …
 "ParentView": [”System view"],
}
```

Leverage example snippets to derive the relationship between CLI commands, i.e., CLI model hierarchy.

NAssim Validator: Design

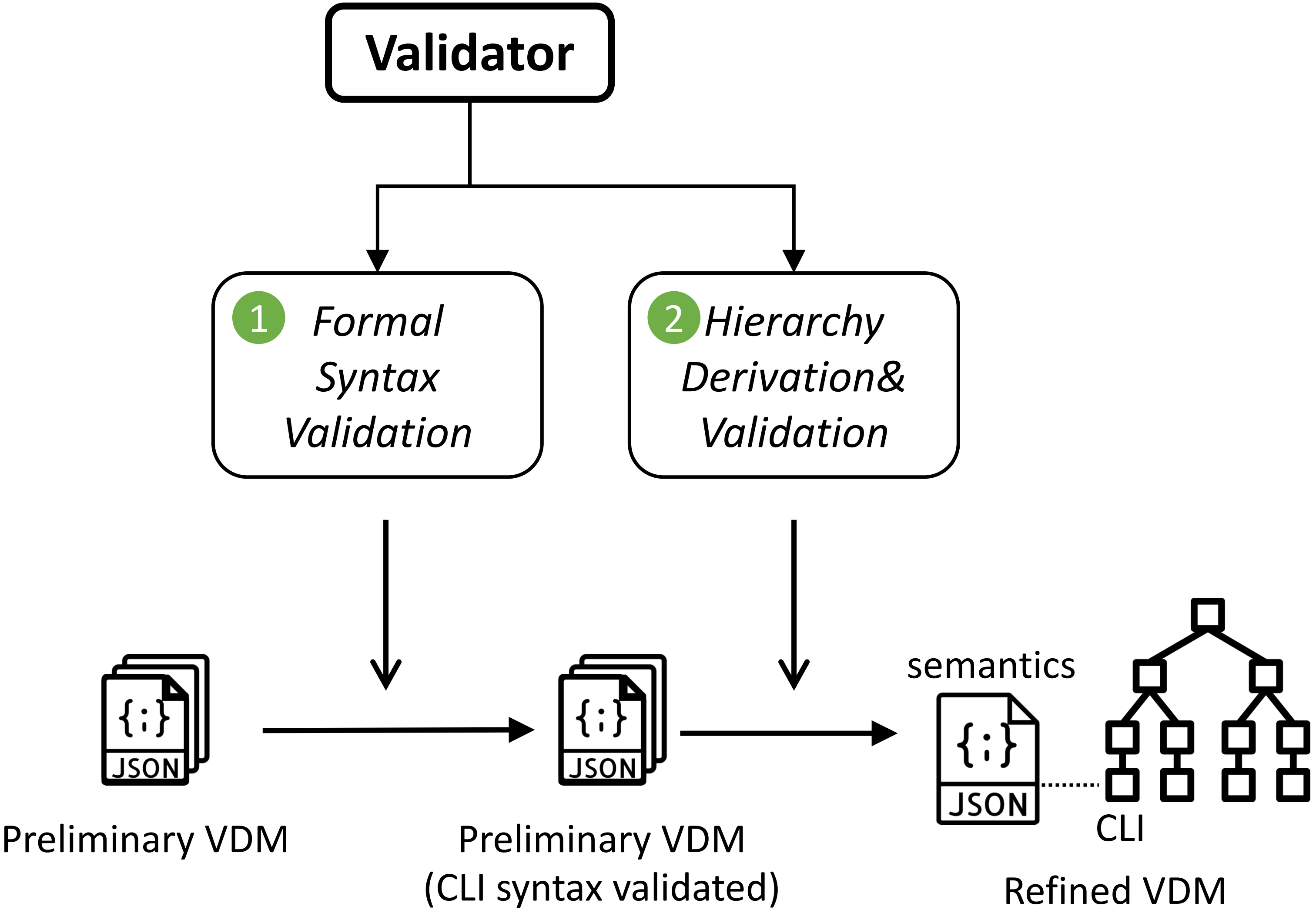


```
{
  "CLIs": [
    "peer <ipv4-address> group <group-name>",
    "undo peer <ipv4-address> group <group-name>"
  ],
  …
  "ParentView": ["BGP view"],
  …
  "Examples": [
    ["<HUAWEI> system-view",
     "[~HUAWEI] bgp 100",
     "[*HUAWEI-bgp] group test internal",
     "[*HUAWEI-bgp] peer 10.1.1.1 group test"] ]
}
```

```
{
  "CLIs": [
    ”bgp <as-number>",
    "undo bgp  [ <as-number> ]"
  ],
  …
  "ParentView": [”System view"],
}
```

Leverage example snippets to derive the relationship between CLI commands, i.e., CLI model hierarchy.

NAssim Validator: Design



Validator

1 *Formal Syntax Validation*

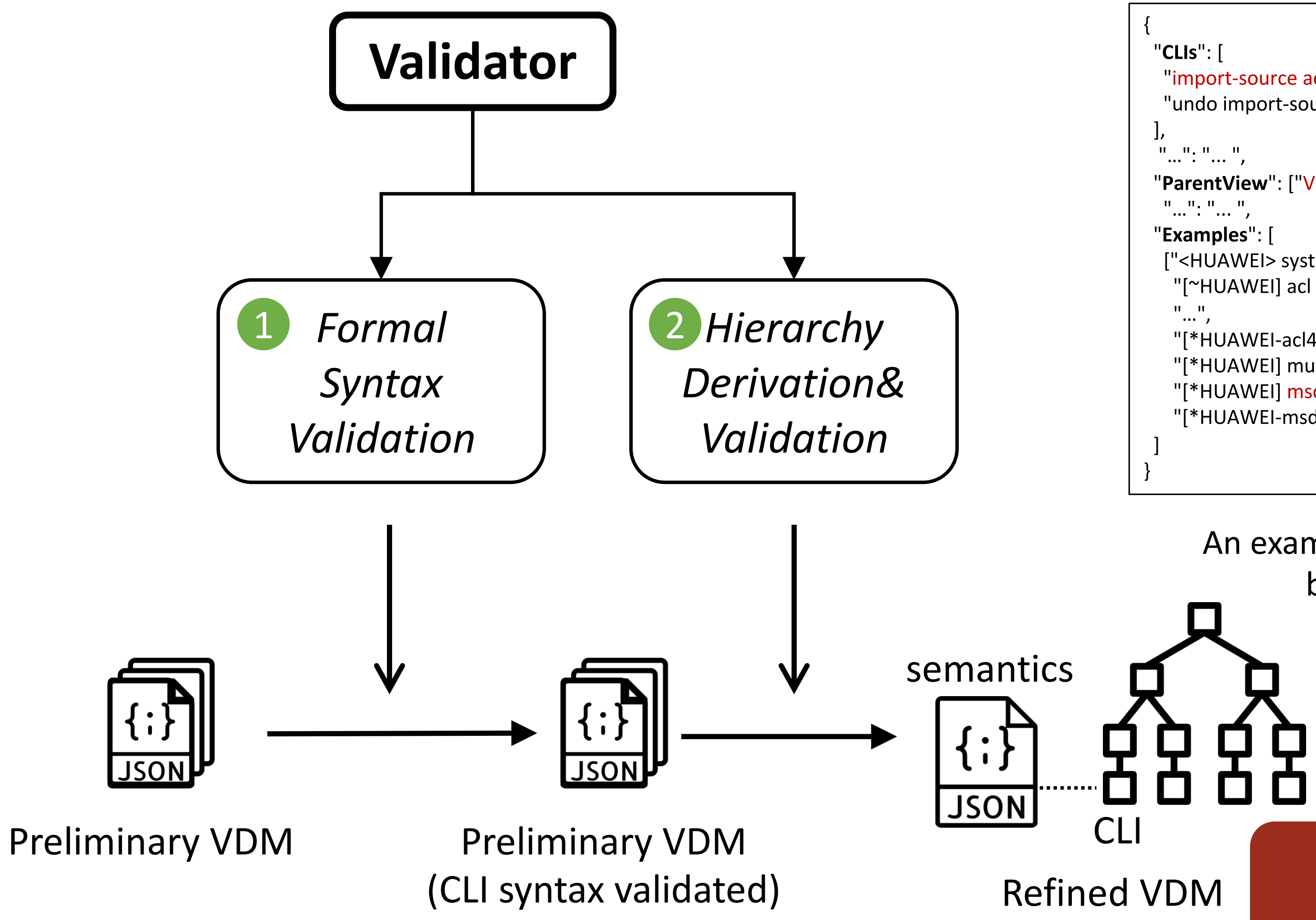
2 *Hierarchy Derivation& Validation*

Preliminary VDM

Preliminary VDM
(CLI syntax validated)

Refined VDM

NAssim Validator: Design



Validator

1 *Formal Syntax Validation*

2 *Hierarchy Derivation& Validation*

Preliminary VDM

Preliminary VDM
(CLI syntax validated)

semantics

CLI

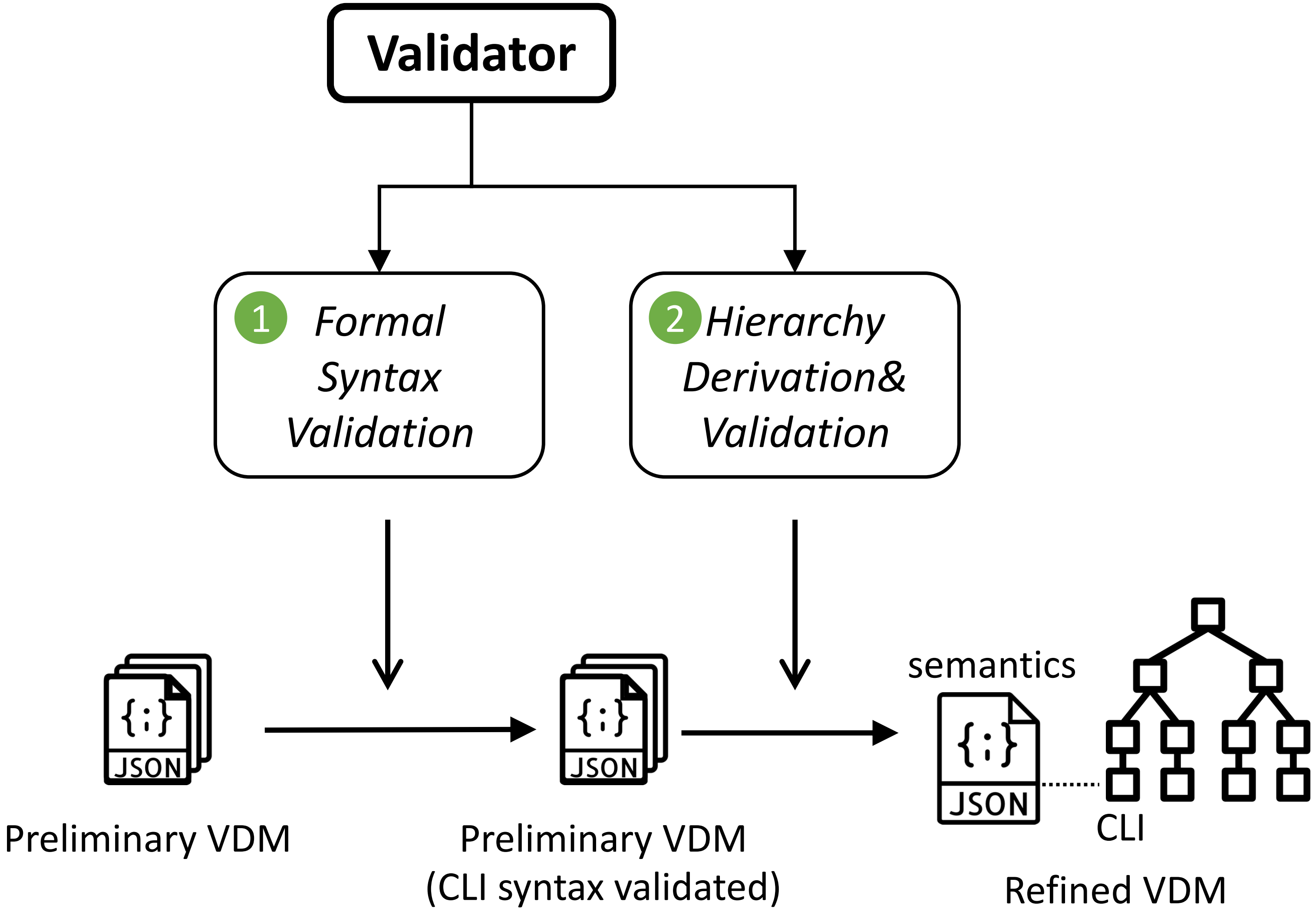
Refined VDM

```
{
 "CLIs": [
  "import-source acl { acl-number | acl-name }",
  "undo import-source"
 ],
  "…": "… ",
 "ParentView": ["VPN instance MSDP view", "MSDP view of a public network instance"],
  "…": "… ",
 "Examples": [
  ["<HUAWEI> system-view",
    "[~HUAWEI] acl number 3101",
    "…",
    "[*HUAWEI-acl4-advance-3101] quit",
    "[*HUAWEI] multicast routing-enable",
    "[*HUAWEI] msdp",
    "[*HUAWEI-msdp] import-source acl 3101"]
 ]
}
```

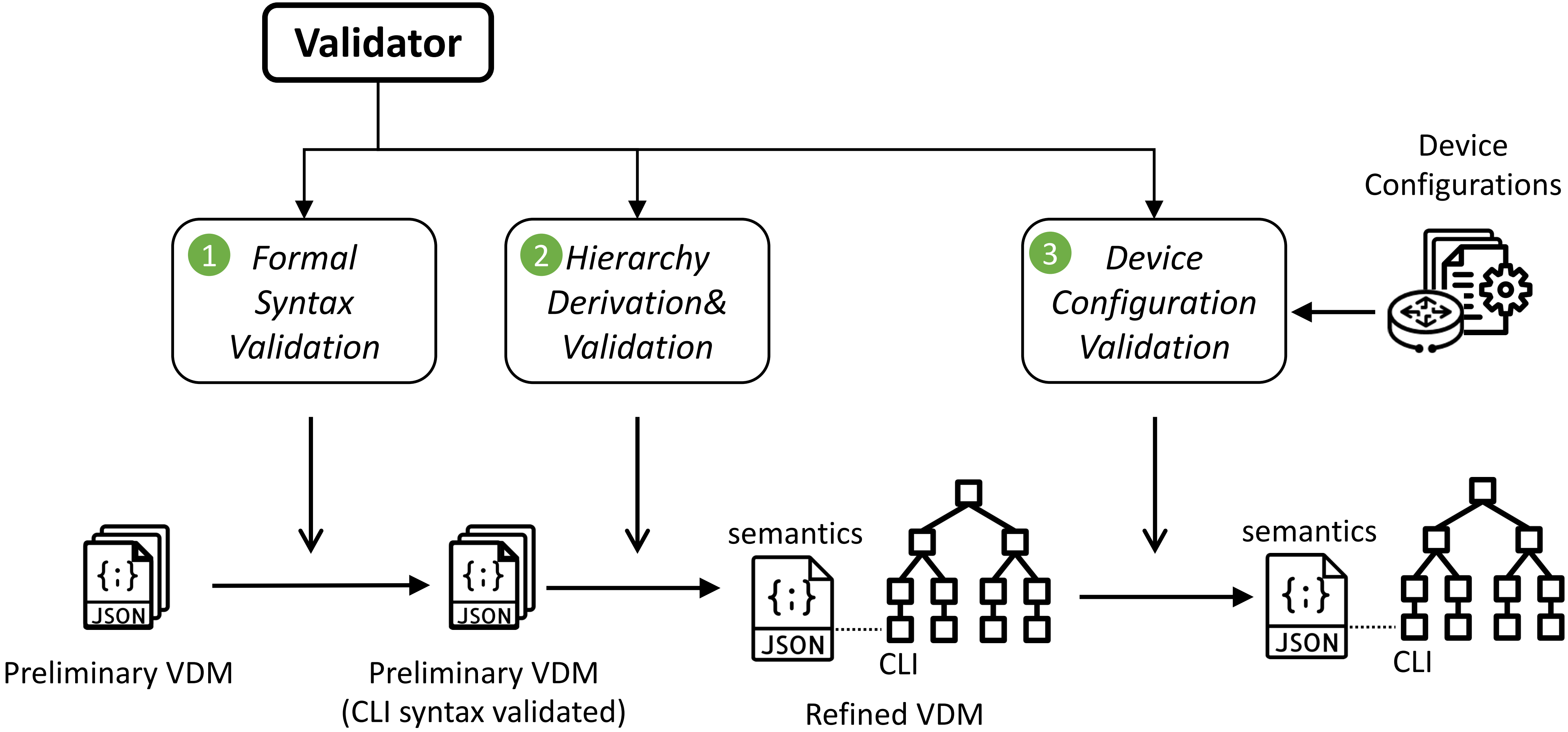
An example ambiguous view not be fully resolved by our hierarchy derivation scheme

Quantify the certainty of the derivation to facilitate expert intervention

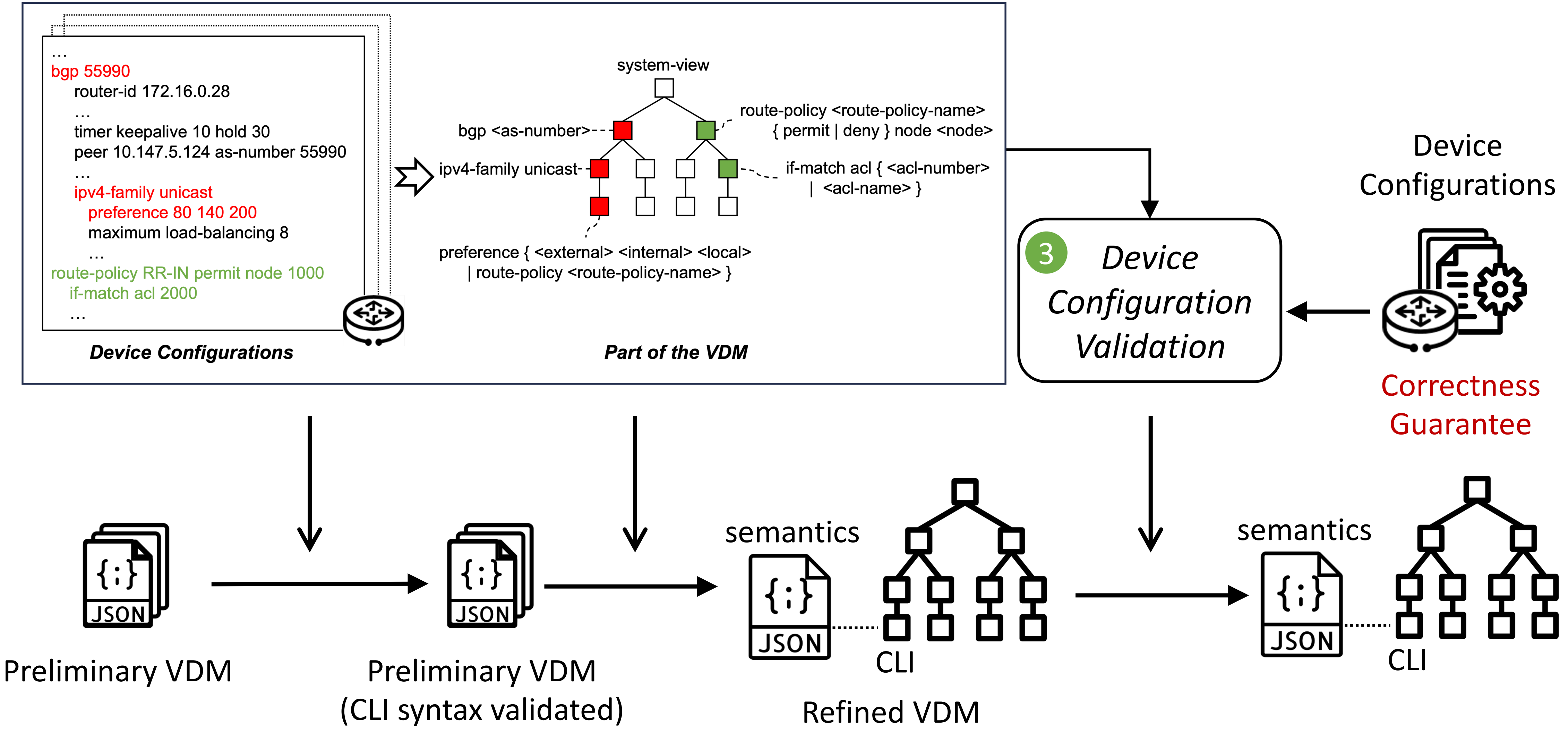
NAssim Validator: Design



NAssim Validator: Design



NAssim Validator: Design



```
…
bgp 55990
    router-id 172.16.0.28
    …
    timer keepalive 10 hold 30
    peer 10.147.5.124 as-number 55990
    …
    ipv4-family unicast
       preference 80 140 200
       maximum load-balancing 8
       …
route-policy RR-IN permit node 1000
   if-match acl 2000
   …
```

Device Configurations

system-view

bgp <as-number>

route-policy <route-policy-name> { permit | deny } node <node>

ipv4-family unicast

if-match acl { <acl-number> | <acl-name> }

preference { <external> <internal> <local> | route-policy <route-policy-name> }

Part of the VDM

Device Configurations

3 *Device Configuration Validation*

Correctness Guarantee

Preliminary VDM

Preliminary VDM (CLI syntax validated)

semantics

JSON

CLI

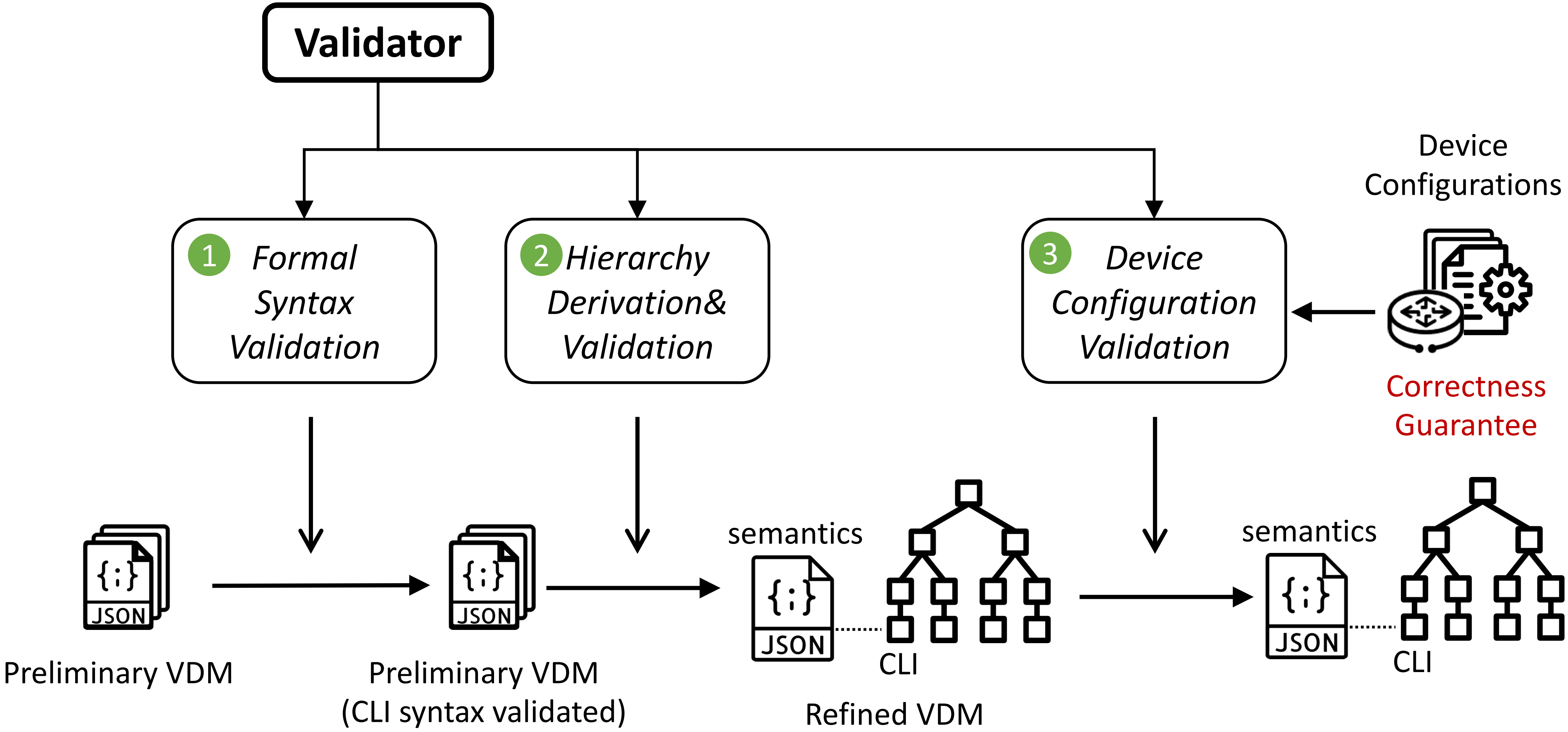
Refined VDM

semantics

JSON

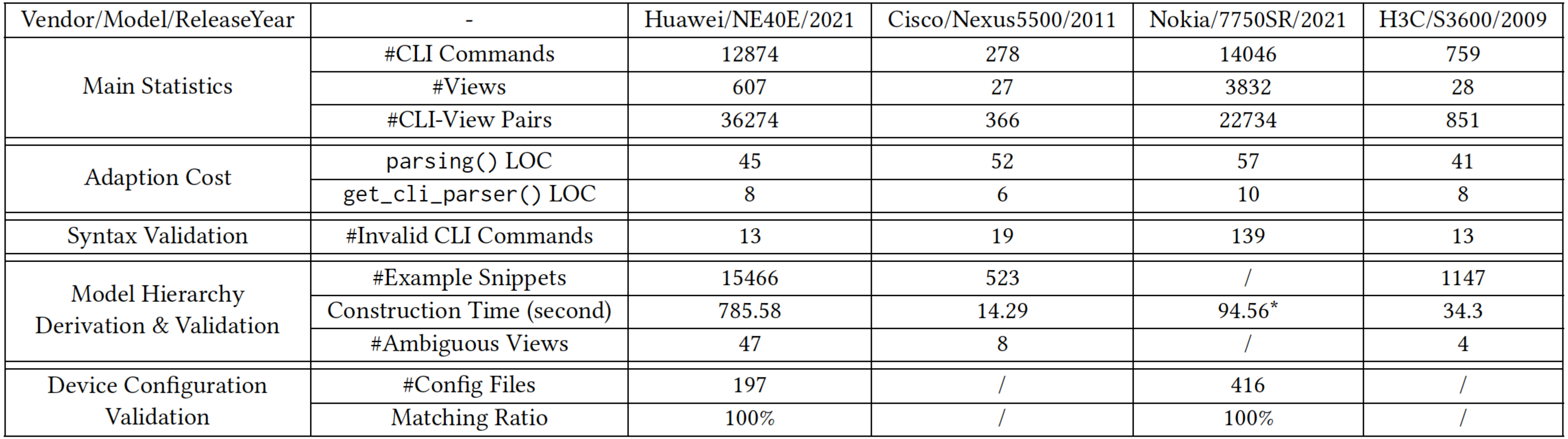
CLI

NAssim Validator: Design



HUAWEI

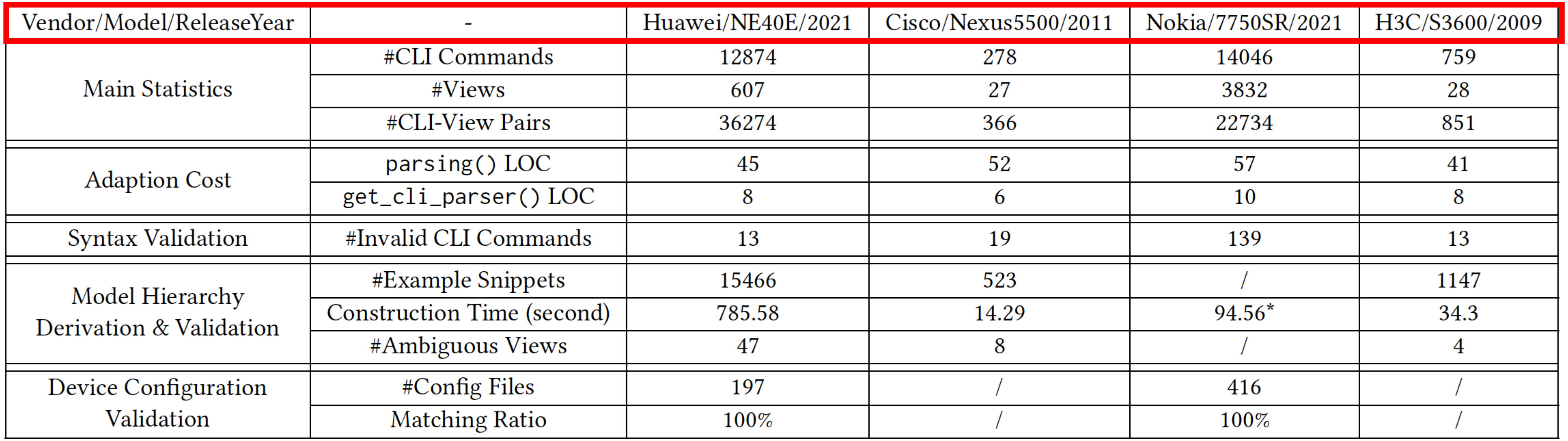
VDM Construction Phase: Parser + Validator



Vendor/Model/ReleaseYear	-	Huawei/NE40E/2021	Cisco/Nexus5500/2011	Nokia/7750SR/2021	H3C/S3600/2009
Main Statistics	#CLI Commands	12874	278	14046	759
	#Views	607	27	3832	28
	#CLI-View Pairs	36274	366	22734	851
Adaption Cost	`parsing()` LOC	45	52	57	41
	`get_cli_parser()` LOC	8	6	10	8
Syntax Validation	#Invalid CLI Commands	13	19	139	13
Model Hierarchy Derivation & Validation	#Example Snippets	15466	523	/	1147
	Construction Time (second)	785.58	14.29	94.56*	34.3
	#Ambiguous Views	47	8	/	4
Device Configuration Validation	#Config Files	197	/	416	/
	Matching Ratio	100%	/	100%	/

Table 3: Evaluation of the VDM Construction Phase. *Nokia manuals do not provide examples, but they explicitly specify model hierarchy in the manuals. Thus, we extract the hierarchy using `Parser_<nokia>` by implementing extra functions

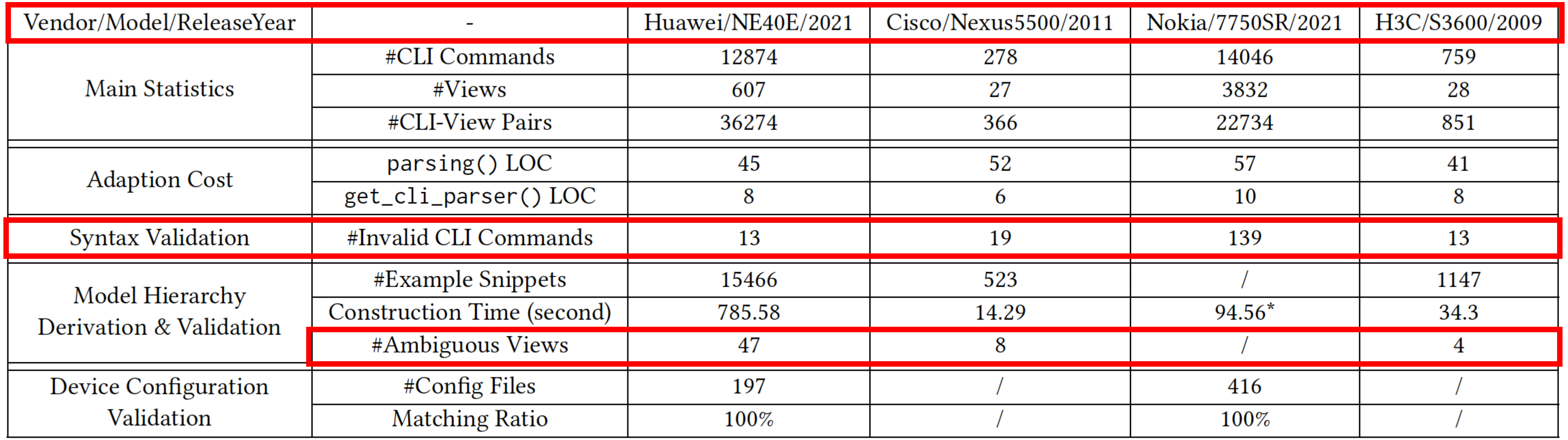
VDM Construction Phase: Parser + Validator



Vendor/Model/ReleaseYear	-	Huawei/NE40E/2021	Cisco/Nexus5500/2011	Nokia/7750SR/2021	H3C/S3600/2009
Main Statistics	#CLI Commands	12874	278	14046	759
	#Views	607	27	3832	28
	#CLI-View Pairs	36274	366	22734	851
Adaption Cost	`parsing()` LOC	45	52	57	41
	`get_cli_parser()` LOC	8	6	10	8
Syntax Validation	#Invalid CLI Commands	13	19	139	13
Model Hierarchy Derivation & Validation	#Example Snippets	15466	523	/	1147
	Construction Time (second)	785.58	14.29	94.56*	34.3
	#Ambiguous Views	47	8	/	4
Device Configuration Validation	#Config Files	197	/	416	/
	Matching Ratio	100%	/	100%	/

Table 3: Evaluation of the VDM Construction Phase. *Nokia manuals do not provide examples, but they explicitly specify model hierarchy in the manuals. Thus, we extract the hierarchy using `Parser_<nokia>` by implementing extra functions

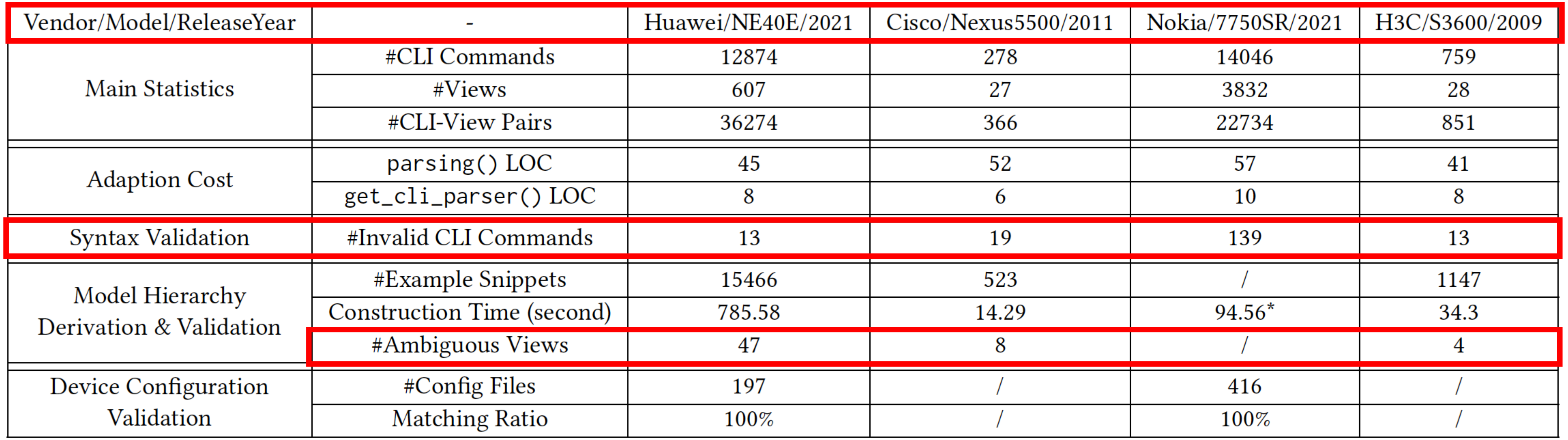
VDM Construction Phase: Parser + Validator



Vendor/Model/ReleaseYear	-	Huawei/NE40E/2021	Cisco/Nexus5500/2011	Nokia/7750SR/2021	H3C/S3600/2009
Main Statistics	#CLI Commands	12874	278	14046	759
	#Views	607	27	3832	28
	#CLI-View Pairs	36274	366	22734	851
Adaption Cost	`parsing()` LOC	45	52	57	41
	`get_cli_parser()` LOC	8	6	10	8
Syntax Validation	#Invalid CLI Commands	13	19	139	13
Model Hierarchy Derivation & Validation	#Example Snippets	15466	523	/	1147
	Construction Time (second)	785.58	14.29	94.56*	34.3
	#Ambiguous Views	47	8	/	4
Device Configuration Validation	#Config Files	197	/	416	/
	Matching Ratio	100%	/	100%	/

Table 3: Evaluation of the VDM Construction Phase. *Nokia manuals do not provide examples, but they explicitly specify model hierarchy in the manuals. Thus, we extract the hierarchy using `Parser_<nokia>` by implementing extra functions

VDM Construction Phase: Parser + Validator

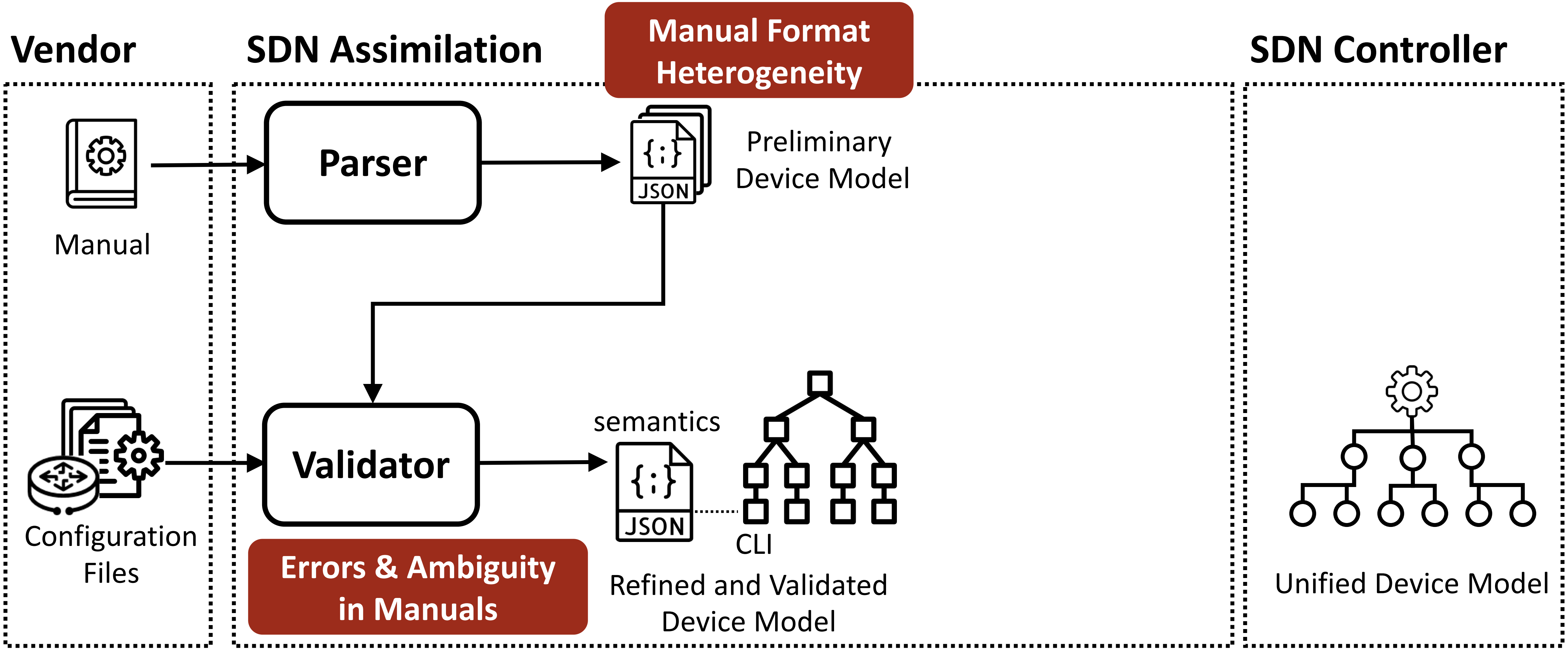


Vendor/Model/ReleaseYear	-	Huawei/NE40E/2021	Cisco/Nexus5500/2011	Nokia/7750SR/2021	H3C/S3600/2009
Main Statistics	#CLI Commands	12874	278	14046	759
	#Views	607	27	3832	28
	#CLI-View Pairs	36274	366	22734	851
Adaption Cost	`parsing()` LOC	45	52	57	41
	`get_cli_parser()` LOC	8	6	10	8
Syntax Validation	#Invalid CLI Commands	13	19	139	13
Model Hierarchy Derivation & Validation	#Example Snippets	15466	523	/	1147
	Construction Time (second)	785.58	14.29	94.56*	34.3
	#Ambiguous Views	47	8	/	4
Device Configuration Validation	#Config Files	197	/	416	/
	Matching Ratio	100%	/	100%	/

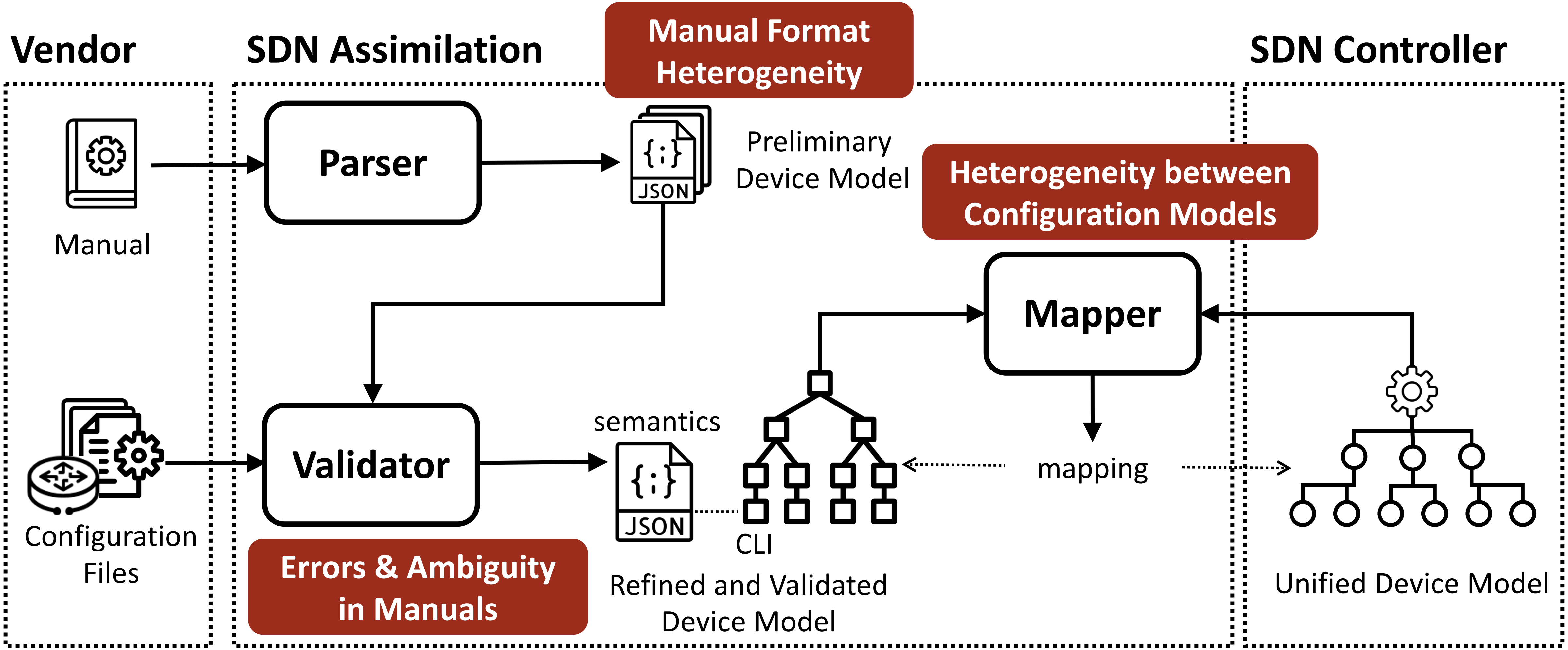
Table 3: Evaluation of the VDM Construction Phase. *Nokia manuals do not provide examples, but they explicitly specify model hierarchy in the manuals. Thus, we extract the hierarchy using `Parser_<nokia>` by implementing extra functions

Refer to our repo for more details: https://github.com/AmyWorkspace/nassim

SDN Network Assimilation(NAssim) in a Nutshell



SDN Network Assimilation(NAssim) in a Nutshell


Vendor

SDN Assimilation

SDN Controller

Manual Format Heterogeneity

Manual

Parser

Preliminary Device Model

Heterogeneity between Configuration Models

Mapper

Configuration Files

Validator

semantics

mapping

CLI

Errors & Ambiguity in Manuals

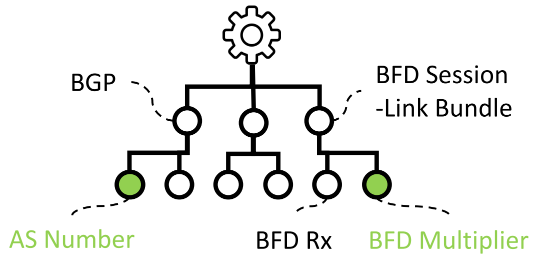
Refined and Validated Device Model

Unified Device Model

HUAWEI

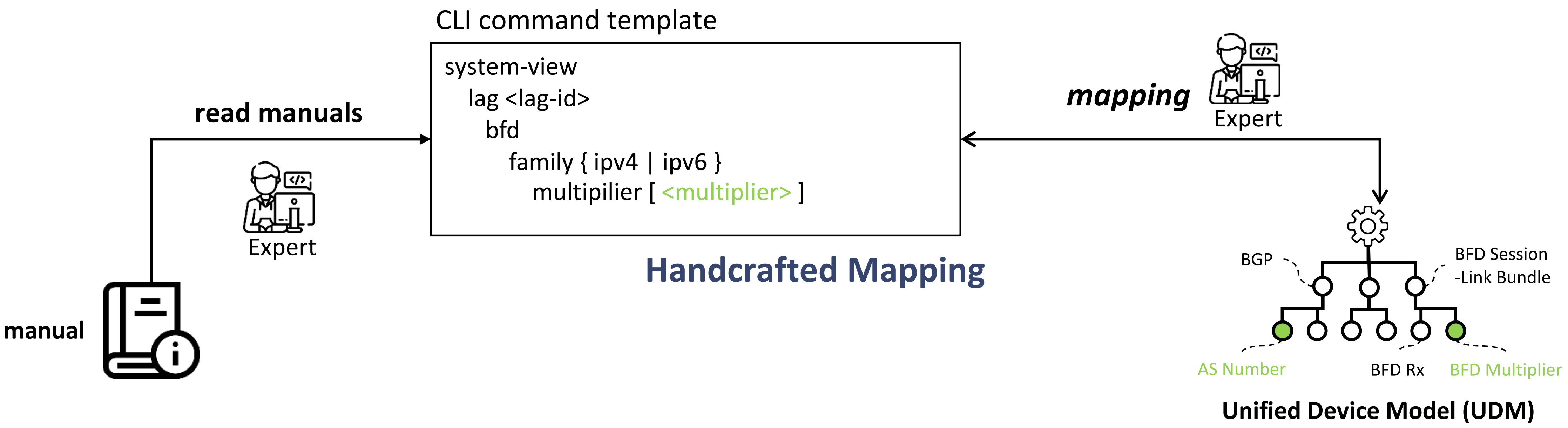
NAssim Mapper: Key Insights

manual



Unified Device Model (UDM)

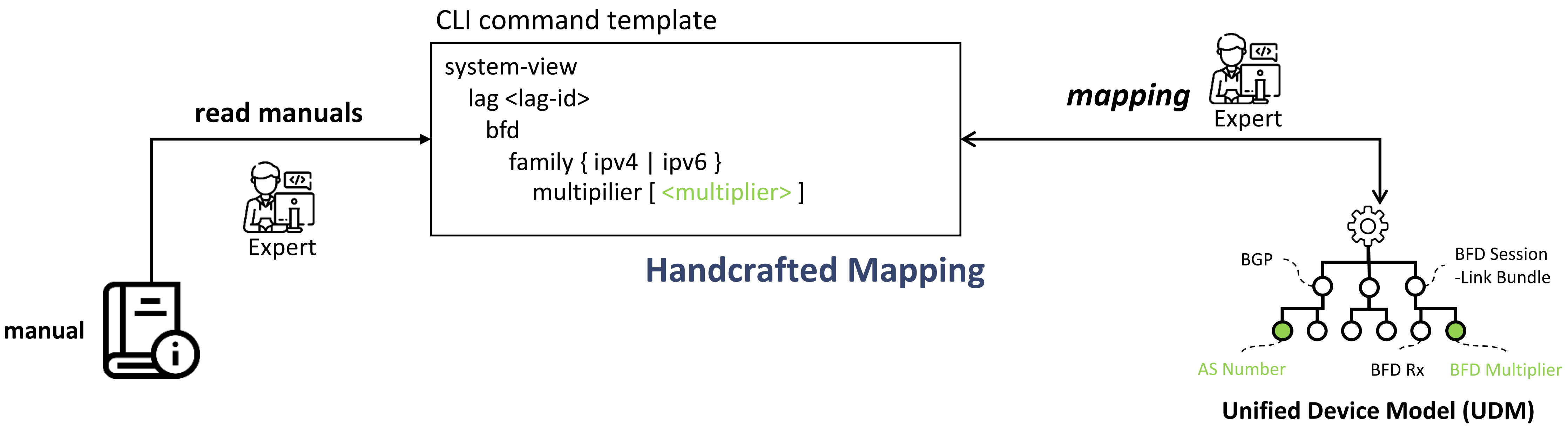
NAssim Mapper: Key Insights



```
system-view
   lag <lag-id>
      bfd
         family { ipv4 | ipv6 }
            multipilier [ <multiplier> ]
```

Handcrafted Mapping

NAssim Mapper: Key Insights



CLI command template

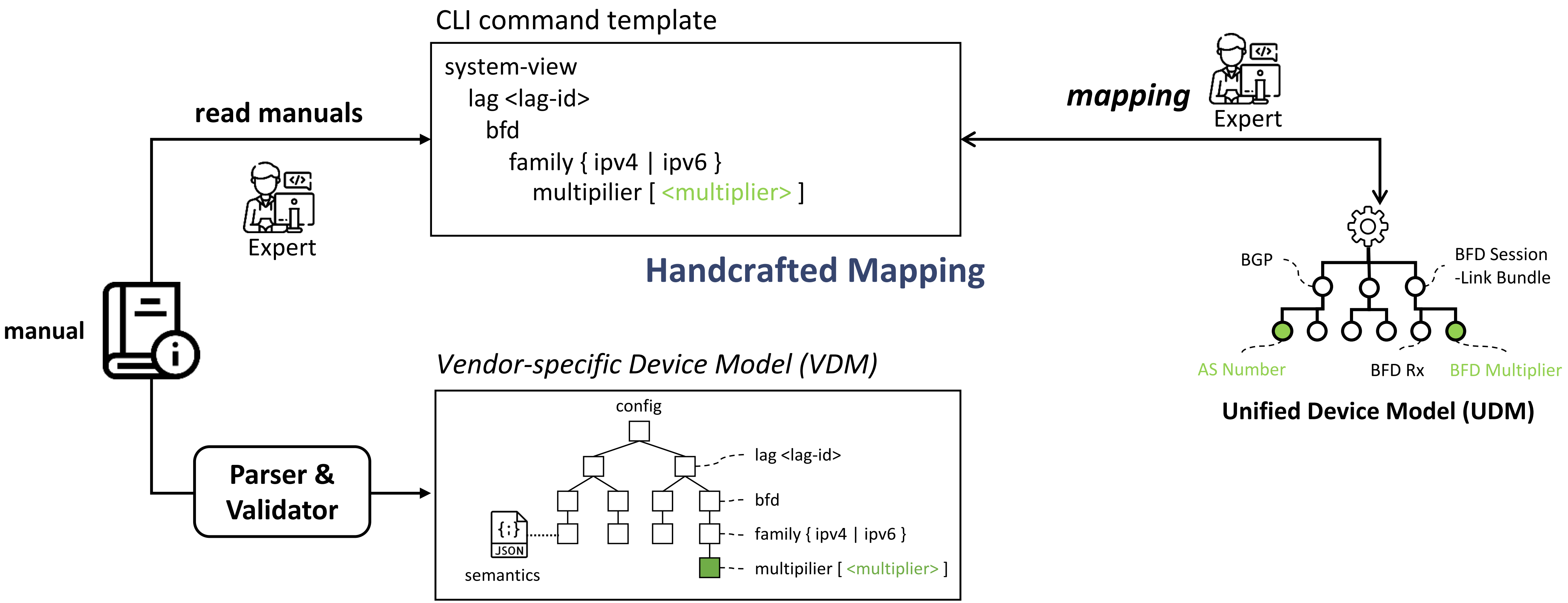
```
system-view
   lag <lag-id>
     bfd
        family { ipv4 | ipv6 }
           multipilier [ <multiplier> ]
```

Handcrafted Mapping

Unified Device Model (UDM)

The key of SNA is to pair semantically similar configuration items.

NAssim Mapper: Key Insights



CLI command template

```
system-view
  lag <lag-id>
    bfd
      family { ipv4 | ipv6 }
        multipilier [ <multiplier> ]
```

read manuals

Expert

manual

Parser & Validator

Handcrafted Mapping

mapping

Expert

Vendor-specific Device Model (VDM)

config

lag <lag-id>

bfd

family { ipv4 | ipv6 }

multipilier [<multiplier>]

{;} JSON

semantics

BGP

BFD Session -Link Bundle

AS Number

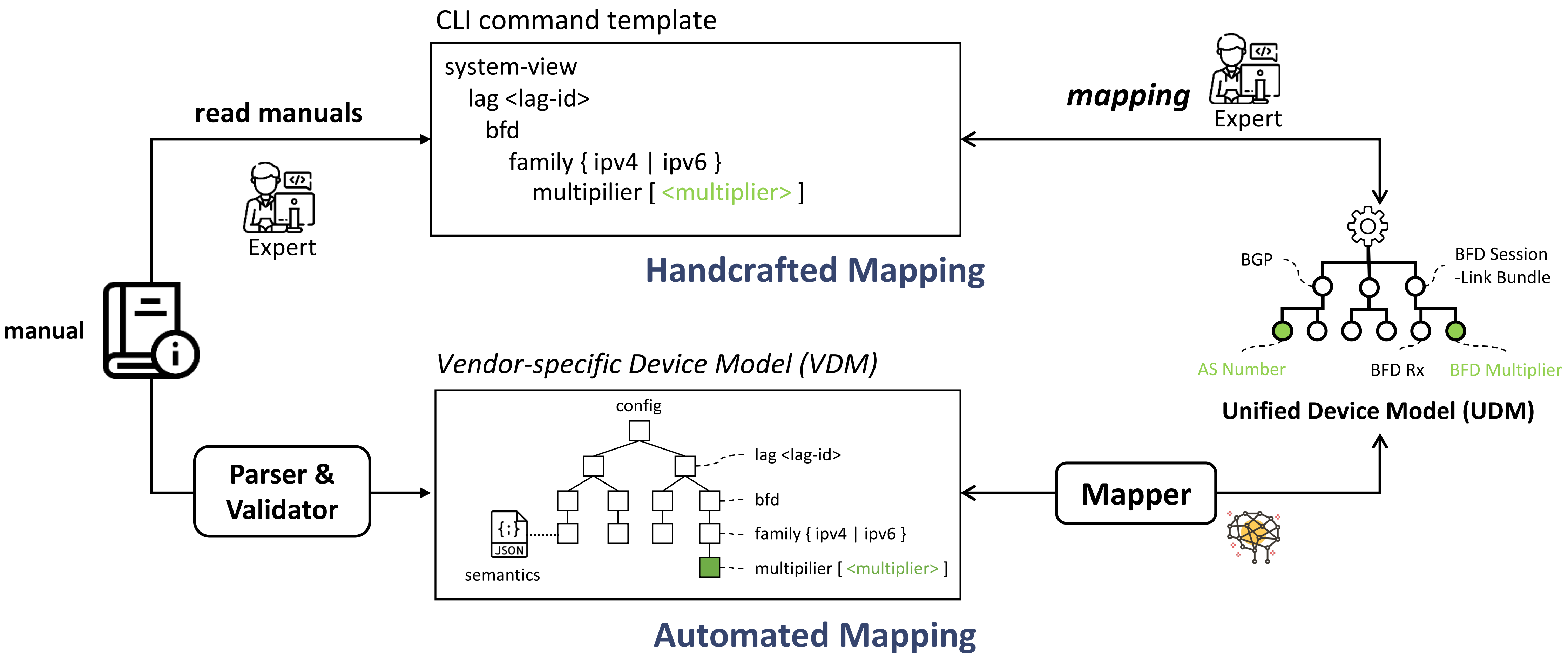
BFD Rx

BFD Multiplier

Unified Device Model (UDM)

The key of SNA is to pair semantically similar configuration items.

NAssim Mapper: Key Insights

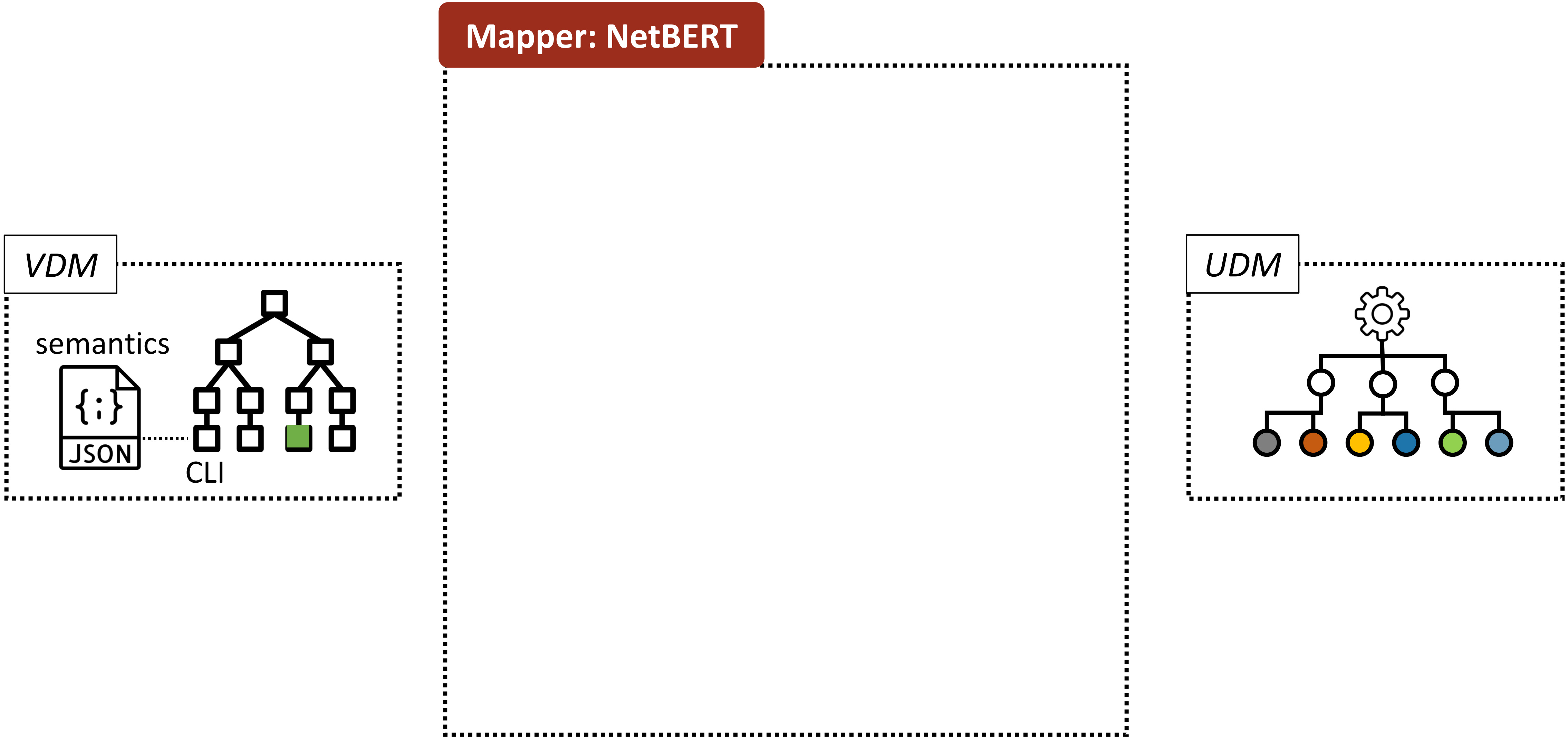


CLI command template

```
system-view
   lag <lag-id>
      bfd
         family { ipv4 | ipv6 }
            multipilier [ <multiplier> ]
```

The key of SNA is to pair semantically similar configuration items.

NAssim Mapper: NetBERT



Mapper: NetBERT

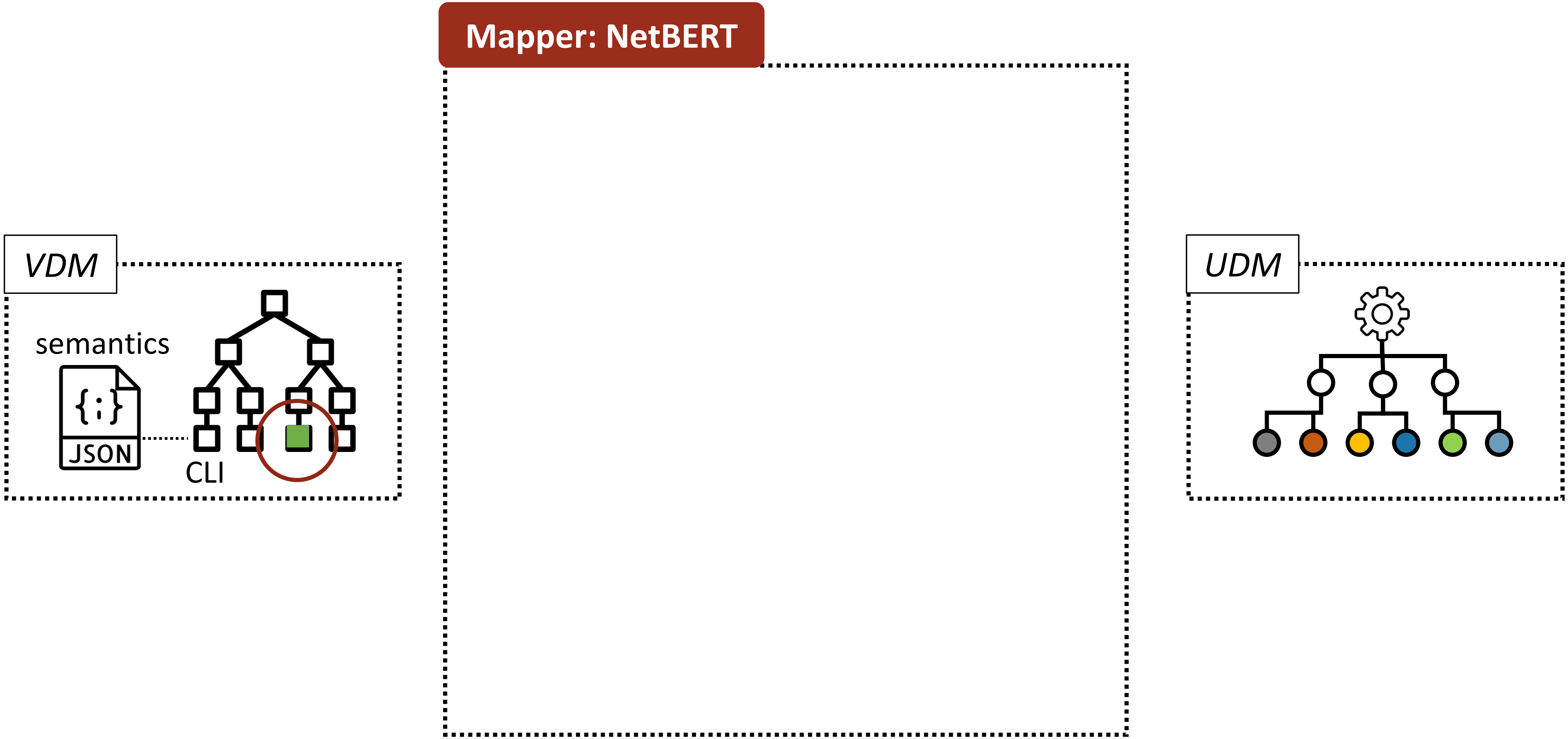
VDM

semantics

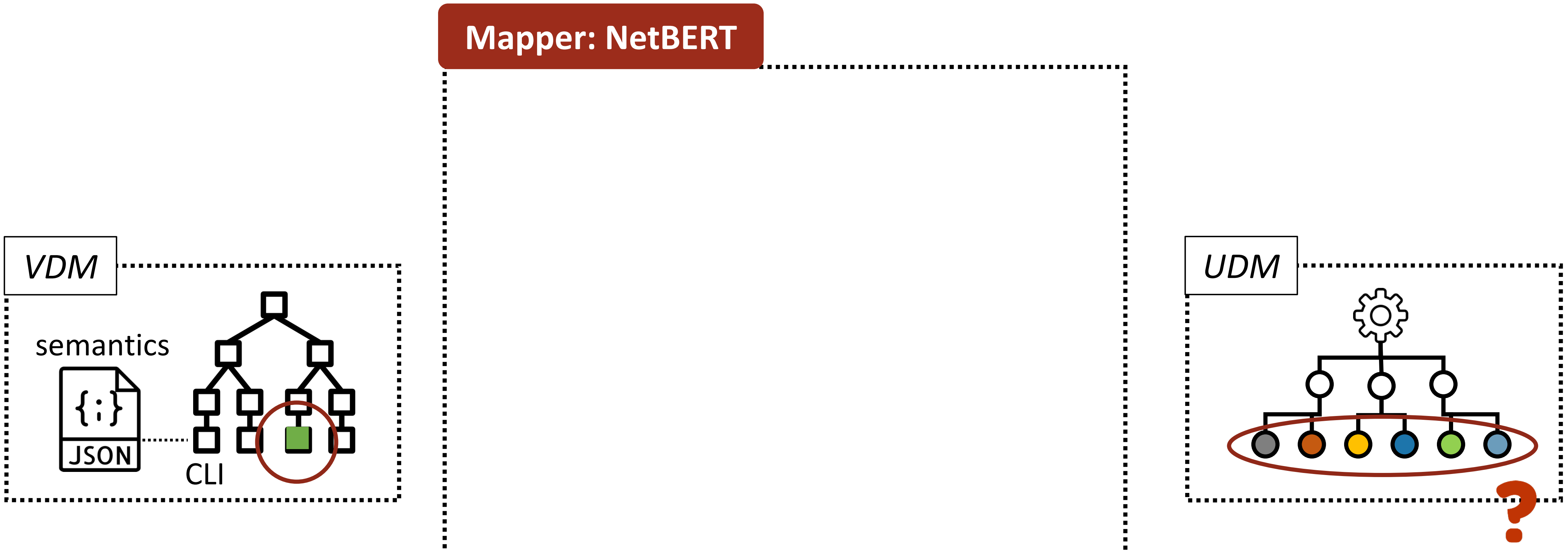
CLI

UDM

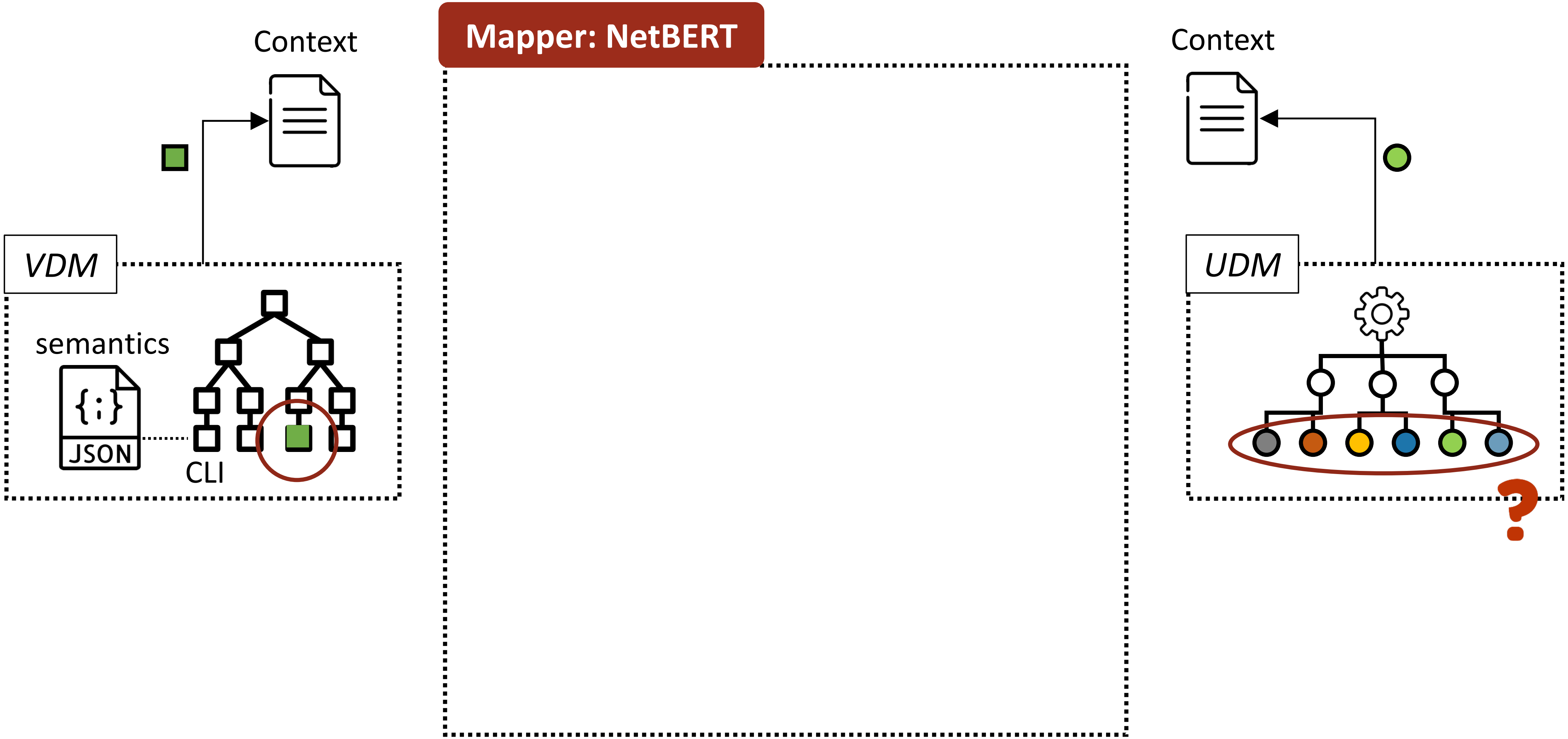
NAssim Mapper: NetBERT



NAssim Mapper: NetBERT



NAssim Mapper: NetBERT

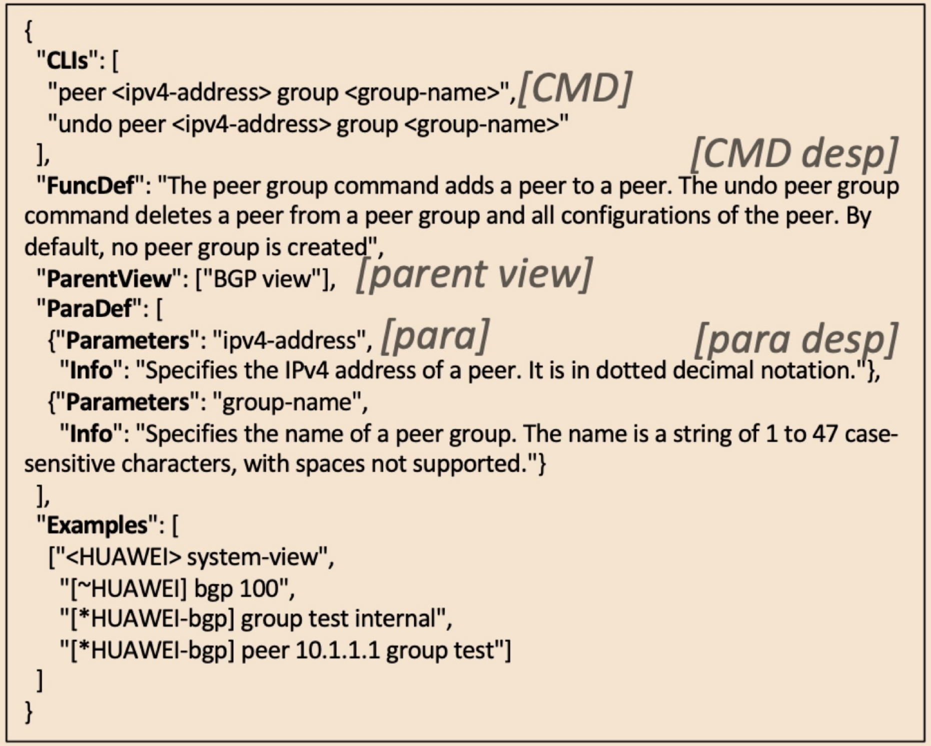


NAssim Mapper: NetBERT

For VDM (or UDM) M , with $\mathrm{n_M}$ parameters and one of its parameter $\mathrm{p_i^M}$($i \in 0,1, \ldots n_M - 1$), its context is

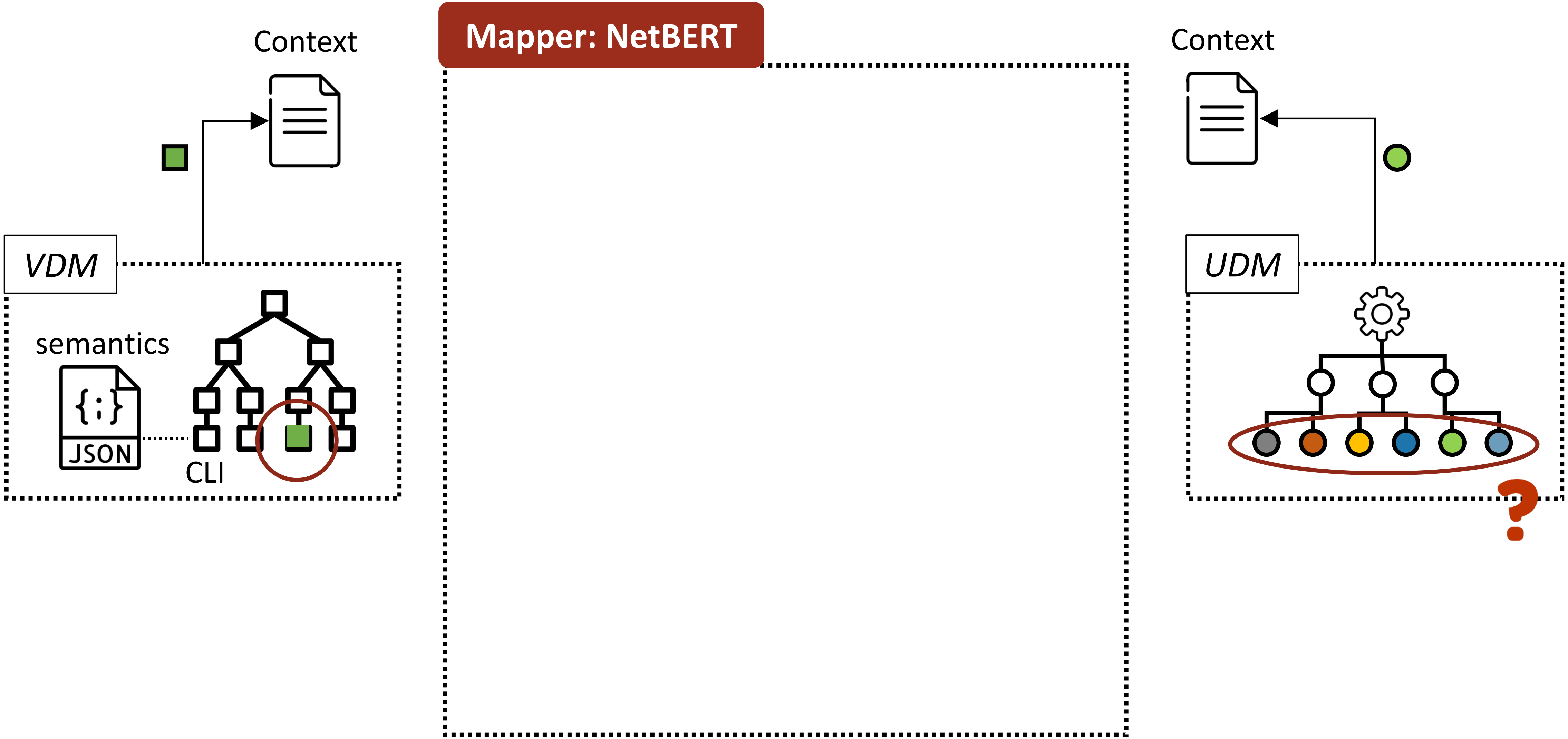
$$C(p_i^M) = [s_i^0, s_i^1, \ldots, s_i^{k_M - 1}]$$

where each s is a text sequence and k_M is the number of extracted sequences.


[CMD]
[CMD desp]
[parent view]
[para]
[para desp]

```
{
 "CLIs": [
  "peer <ipv4-address> group <group-name>",
  "undo peer <ipv4-address> group <group-name>"
 ],
 "FuncDef": "The peer group command adds a peer to a peer. The undo peer group command deletes a peer from a peer group and all configurations of the peer. By default, no peer group is created",
 "ParentView": ["BGP view"],
 "ParaDef": [
  {"Parameters": "ipv4-address",
   "Info": "Specifies the IPv4 address of a peer. It is in dotted decimal notation."},
  {"Parameters": "group-name",
   "Info": "Specifies the name of a peer group. The name is a string of 1 to 47 case-sensitive characters, with spaces not supported."}
 ],
 "Examples": [
  ["<HUAWEI> system-view",
   "[~HUAWEI] bgp 100",
   "[*HUAWEI-bgp] group test internal",
   "[*HUAWEI-bgp] peer 10.1.1.1 group test"]
 ]
}
```

NAssim Mapper: NetBERT



Context

Mapper: NetBERT

Context

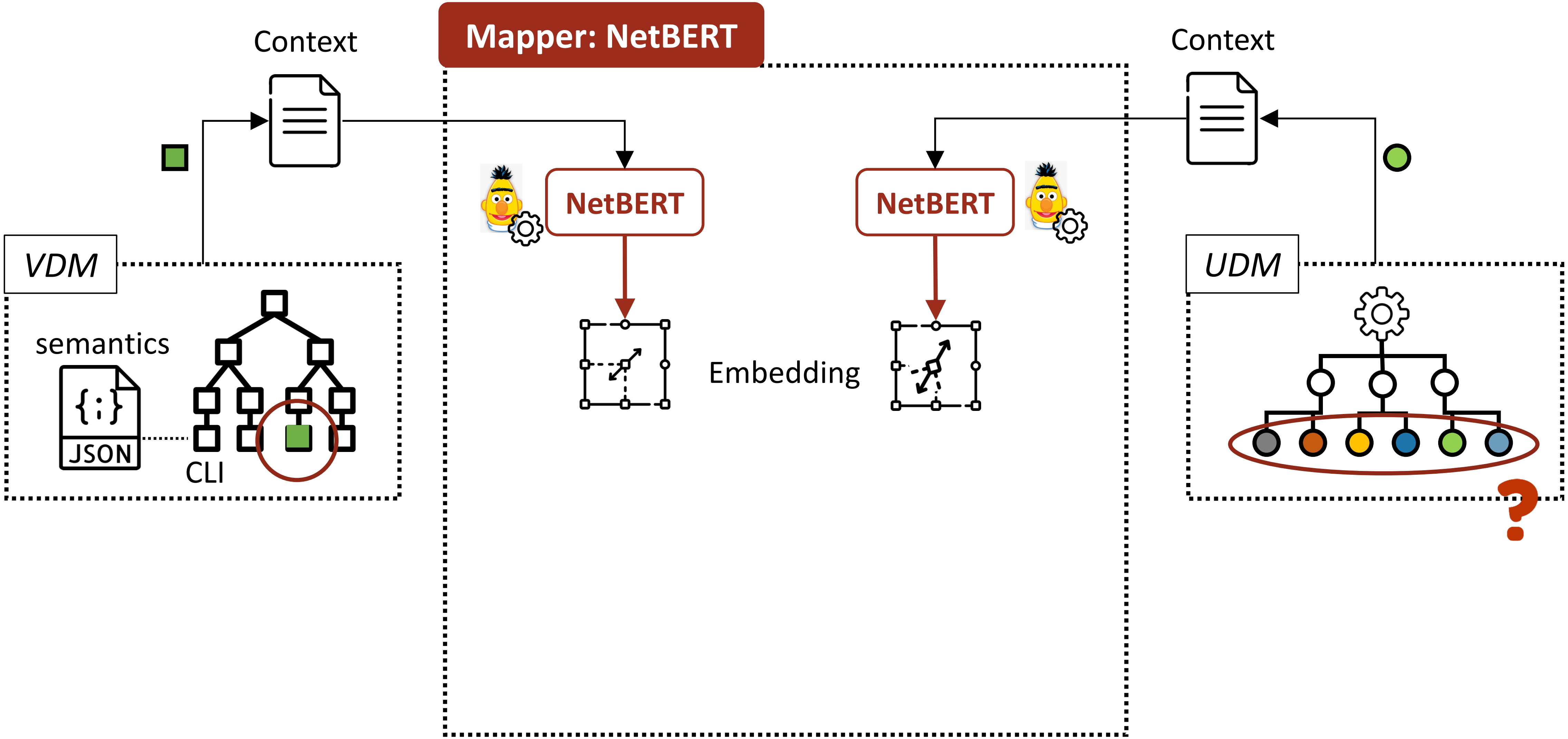
VDM

semantics

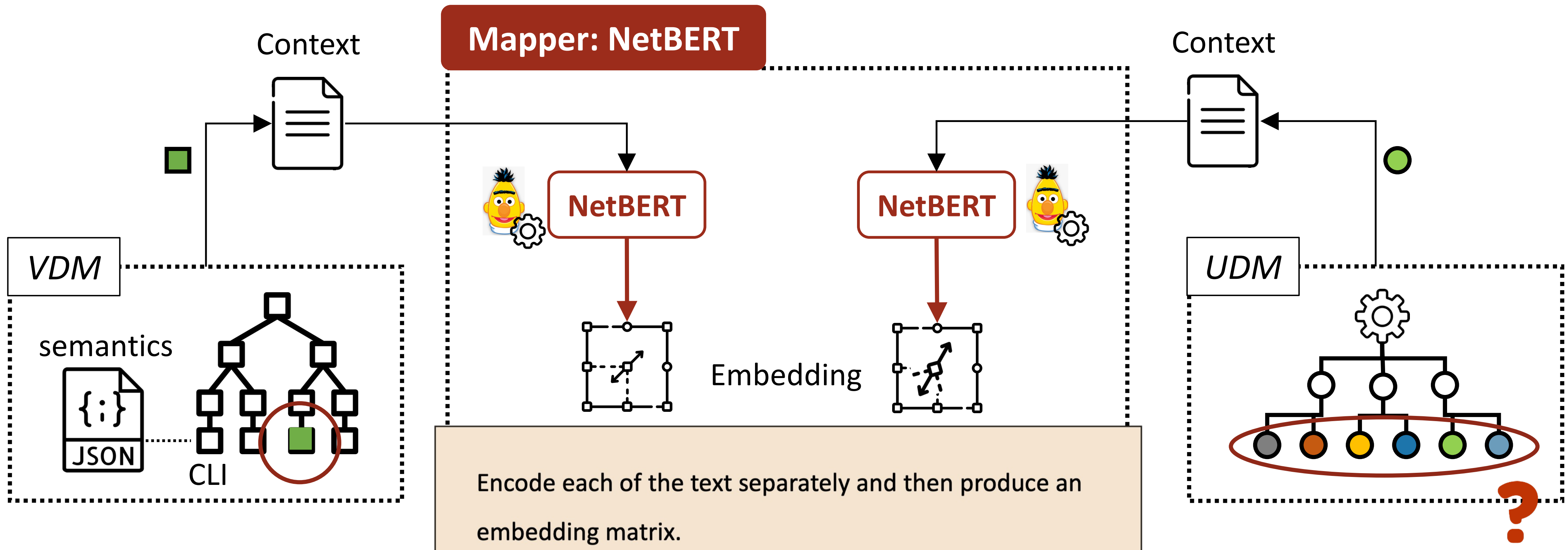
CLI

UDM

NAssim Mapper: NetBERT



NAssim Mapper: NetBERT

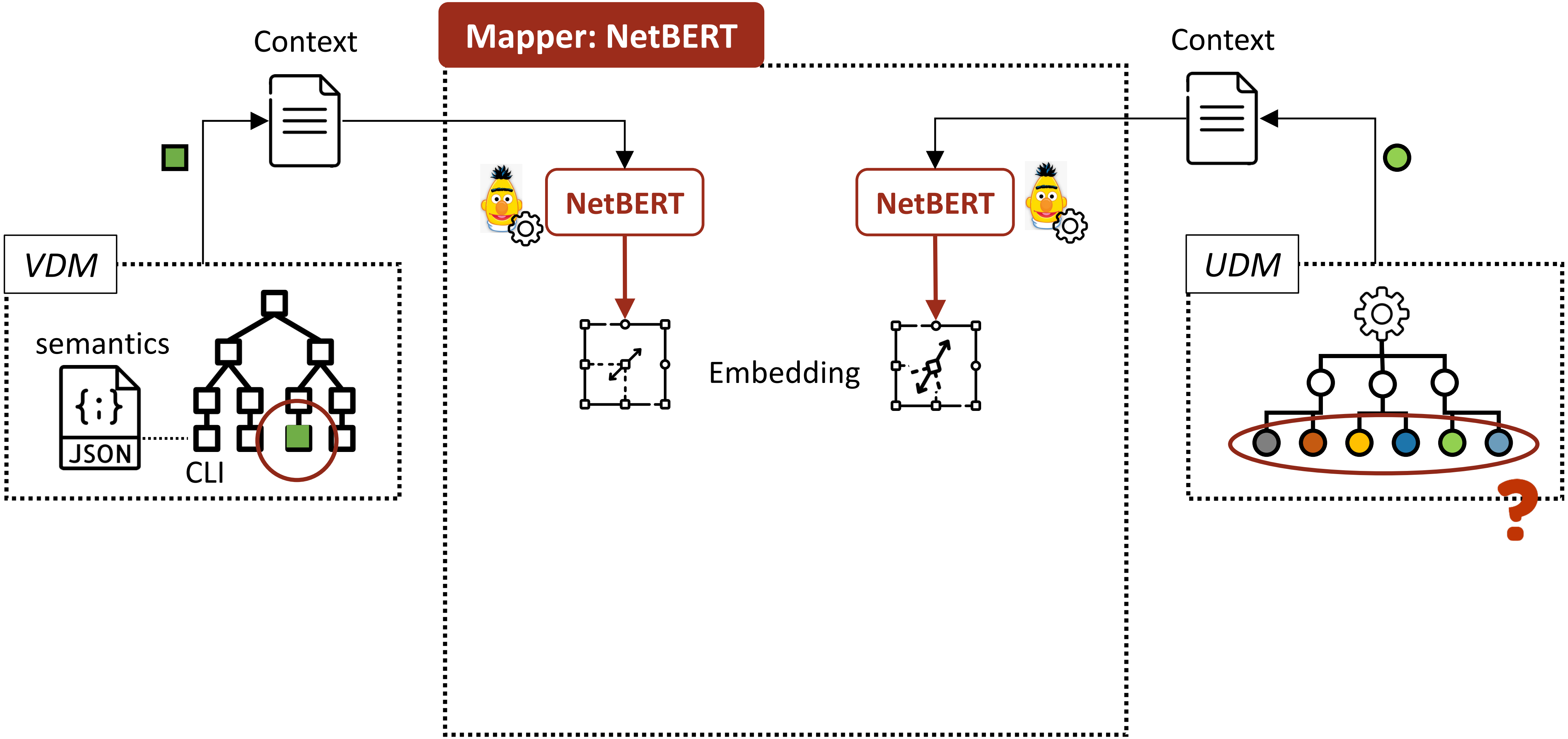


Encode each of the text separately and then produce an embedding matrix.

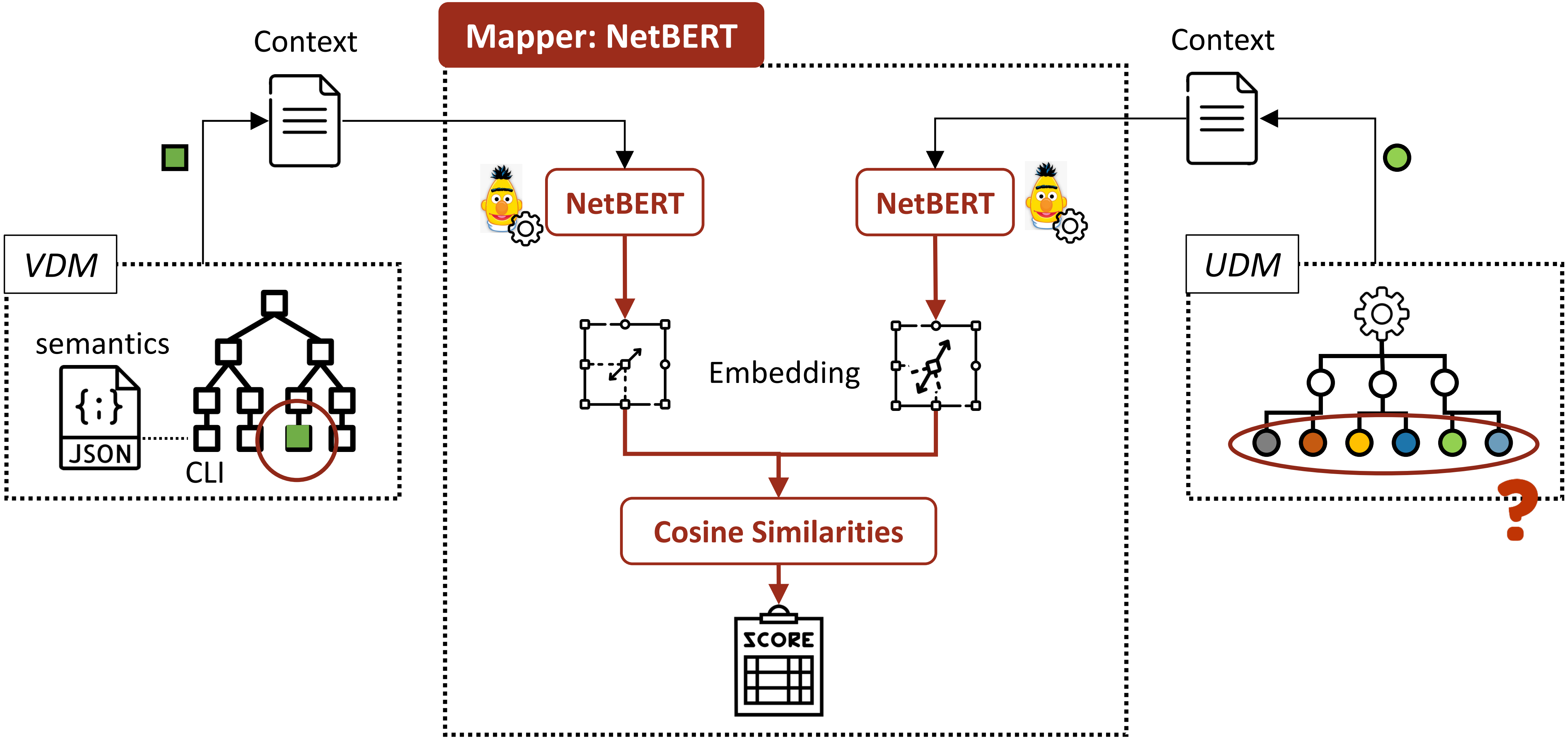
Assume the output dimension of the encoder $e(\cdot)$ is m, the context embedding of $\mathrm{p}_\mathrm{i}^\mathrm{M}$ is:

$$E_i^M = e(C(p_i^M)) = e([s_i^0, s_i^1, \dots, s_i^{k_M-1}]) \in R^{k_M \times m}$$

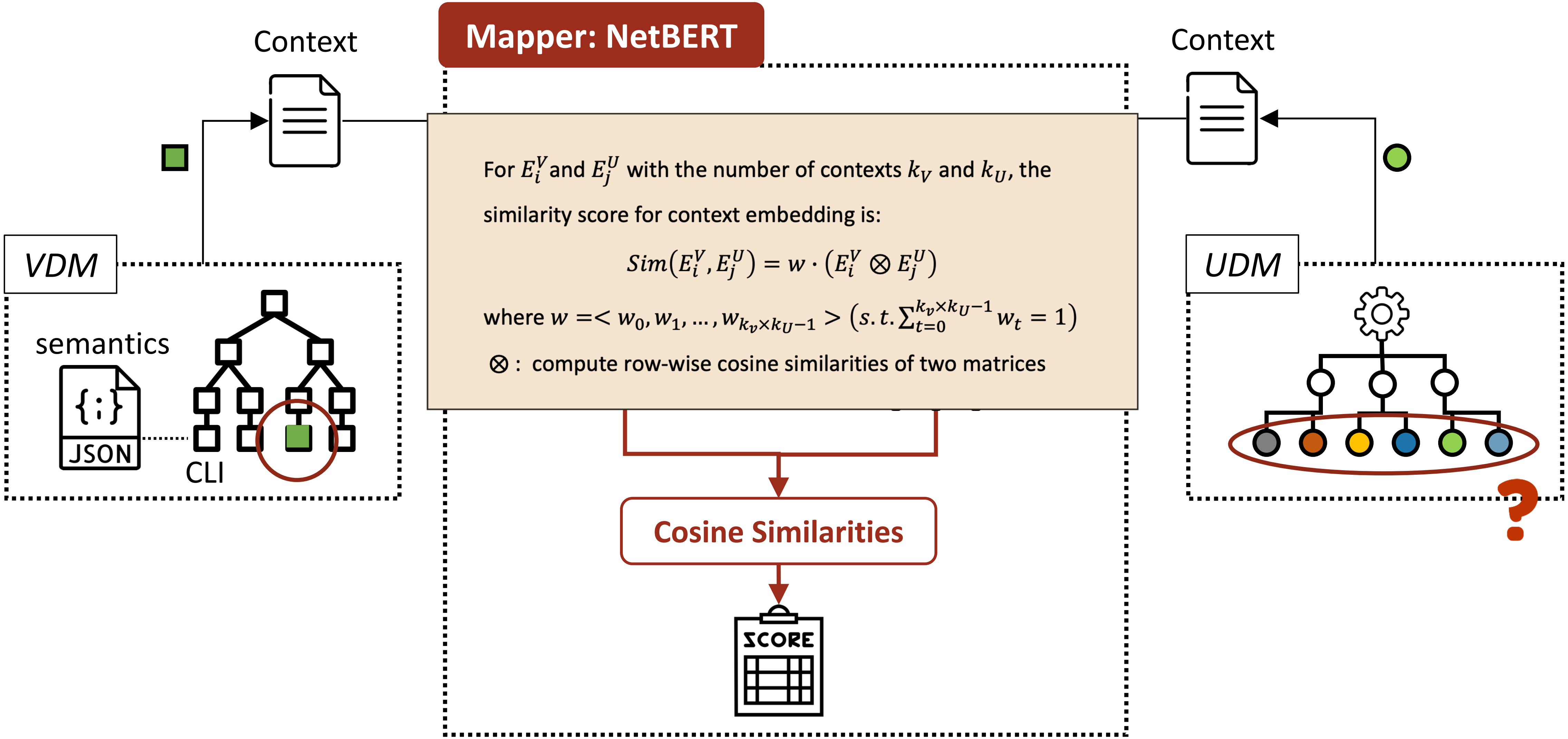
NAssim Mapper: NetBERT



NAssim Mapper: NetBERT



NAssim Mapper: NetBERT



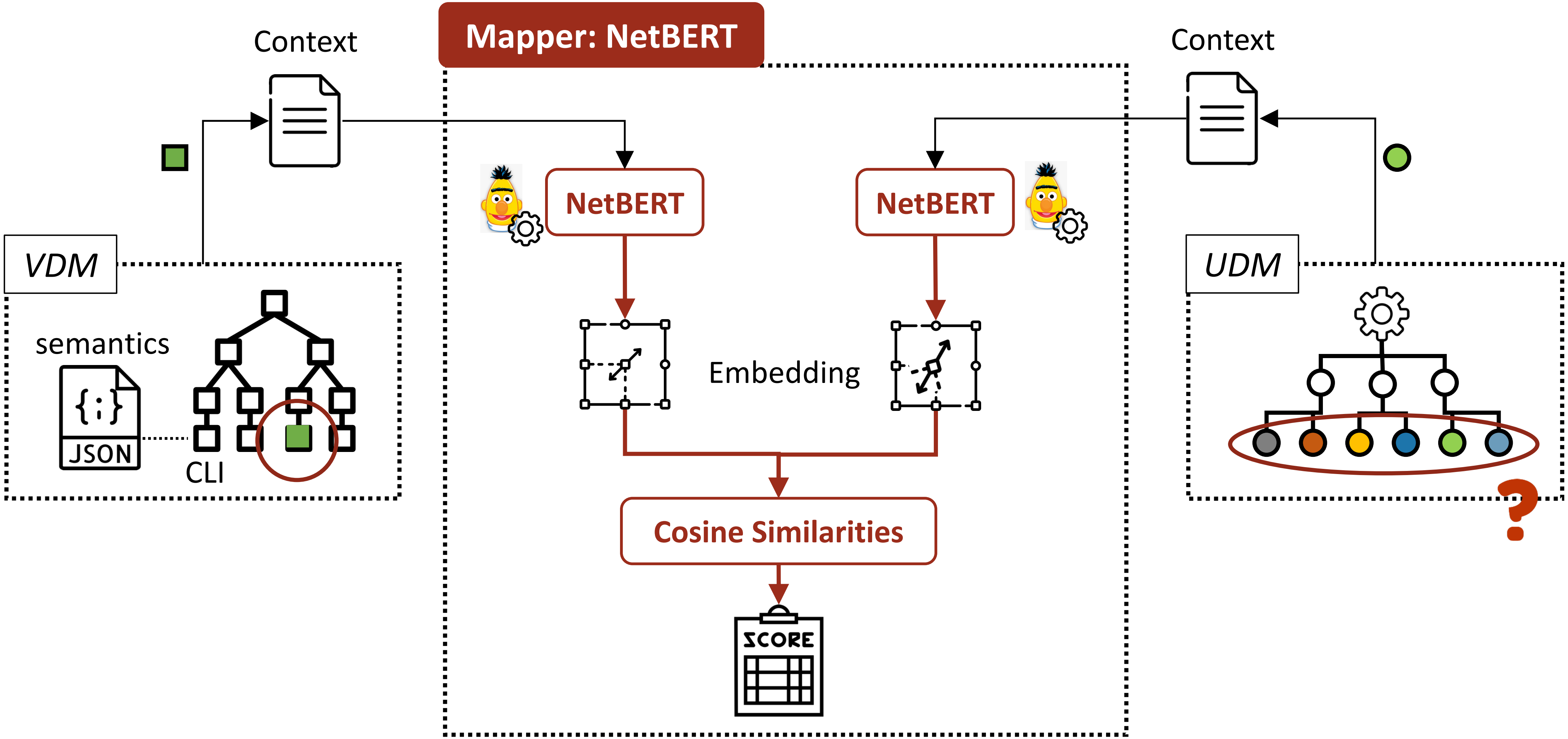
For E_i^V and E_j^U with the number of contexts k_V and k_U, the similarity score for context embedding is:

$$Sim(E_i^V, E_j^U) = w \cdot (E_i^V \otimes E_j^U)$$

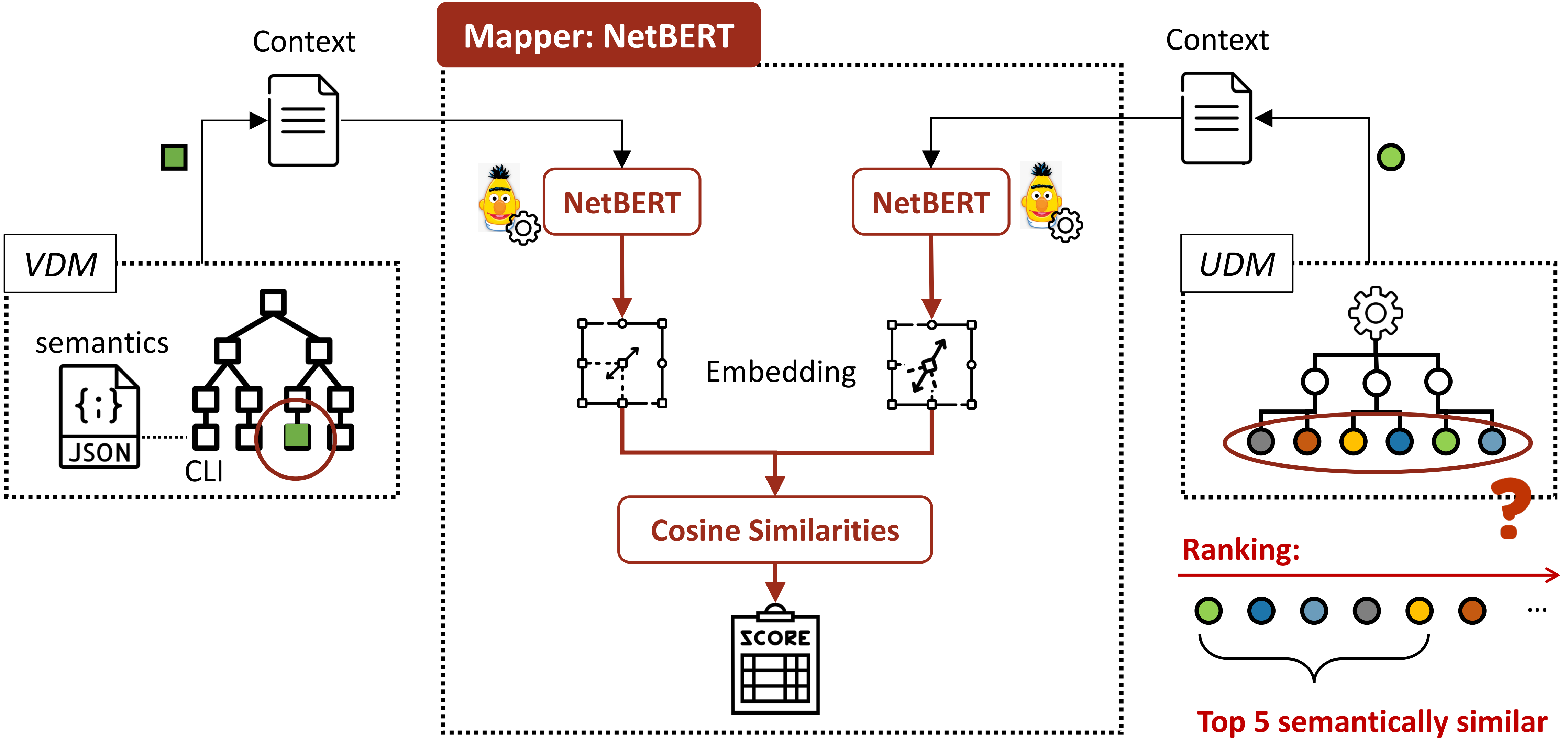
where $w =< w_0, w_1, \dots, w_{k_v \times k_U - 1} > (s.t. \sum_{t=0}^{k_v \times k_U - 1} w_t = 1)$

$\otimes$: compute row-wise cosine similarities of two matrices

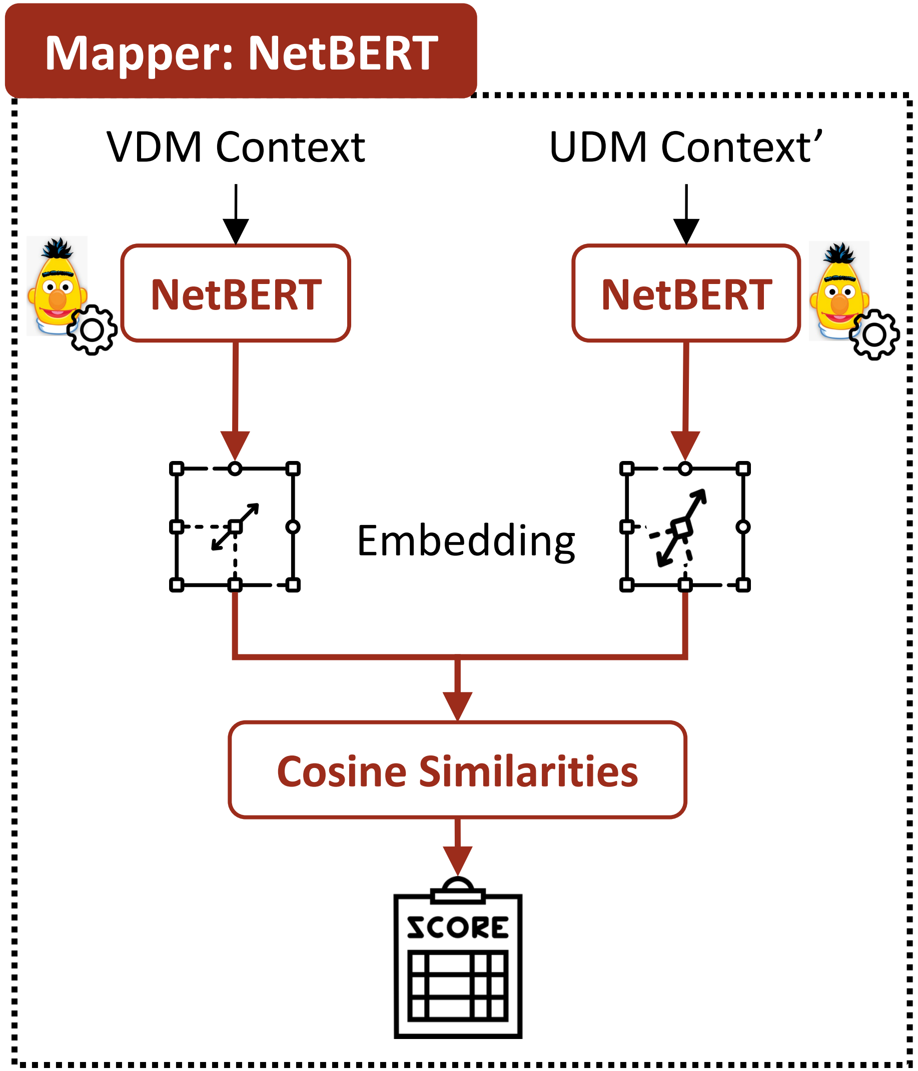
NAssim Mapper: NetBERT



NAssim Mapper: NetBERT

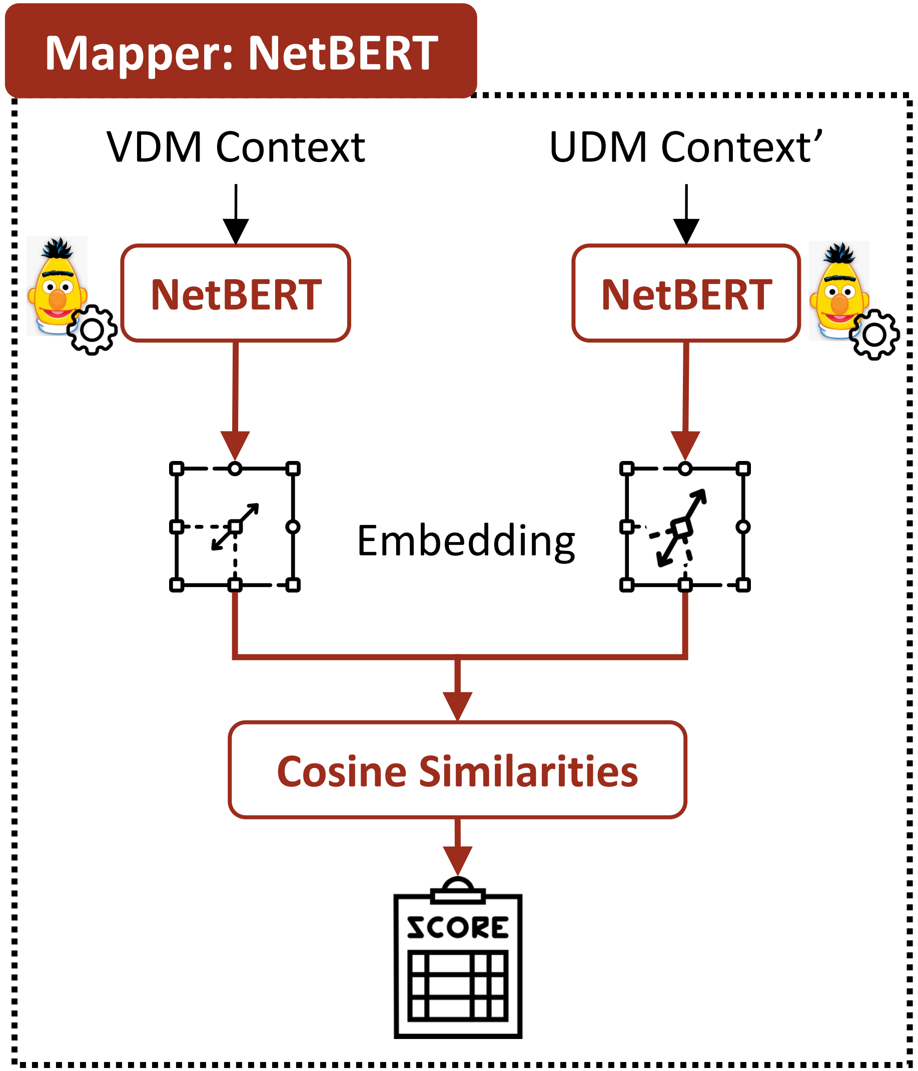


NAssim Mapper: NetBERT Training



NAssim Mapper: NetBERT Training

Pre-trained model*

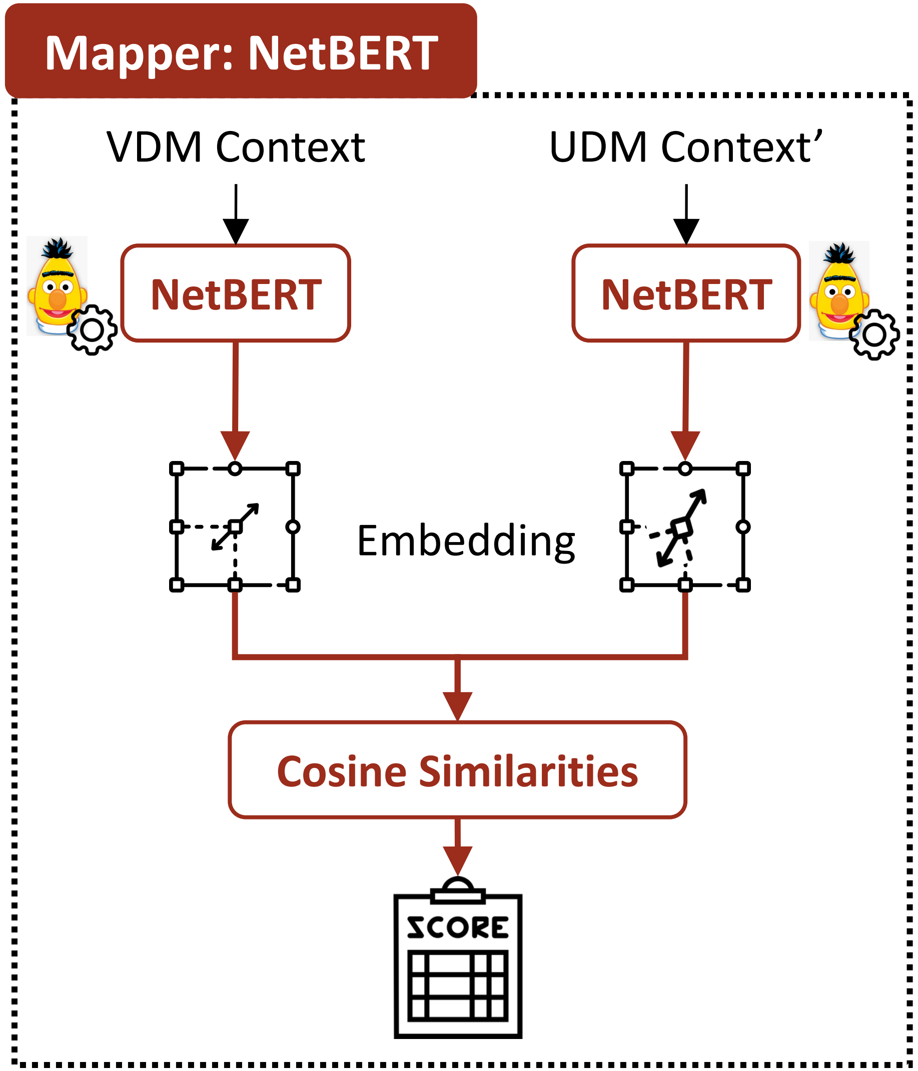


**Reimers, Nils, et al. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." EMNLP, 2019.*

NAssim Mapper: NetBERT Training

Pre-trained model*

Net-Config Domain Data



Mapper: NetBERT

VDM Context

UDM Context’

NetBERT

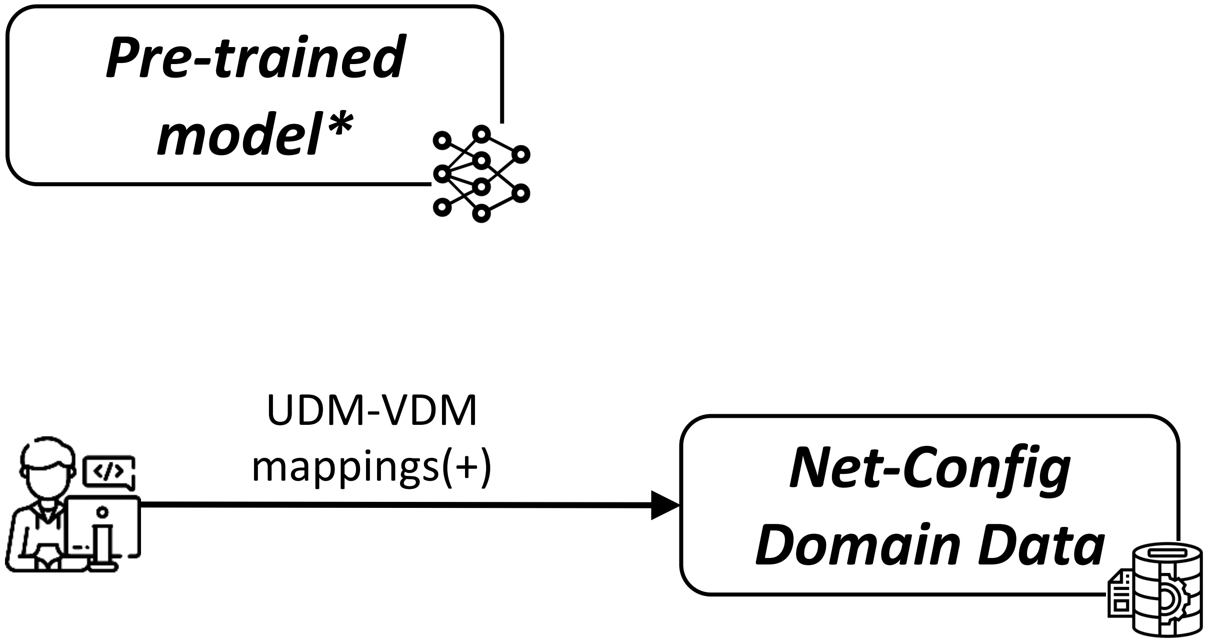
NetBERT

Embedding

Cosine Similarities

*Reimers, Nils, et al. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” EMNLP, 2019.

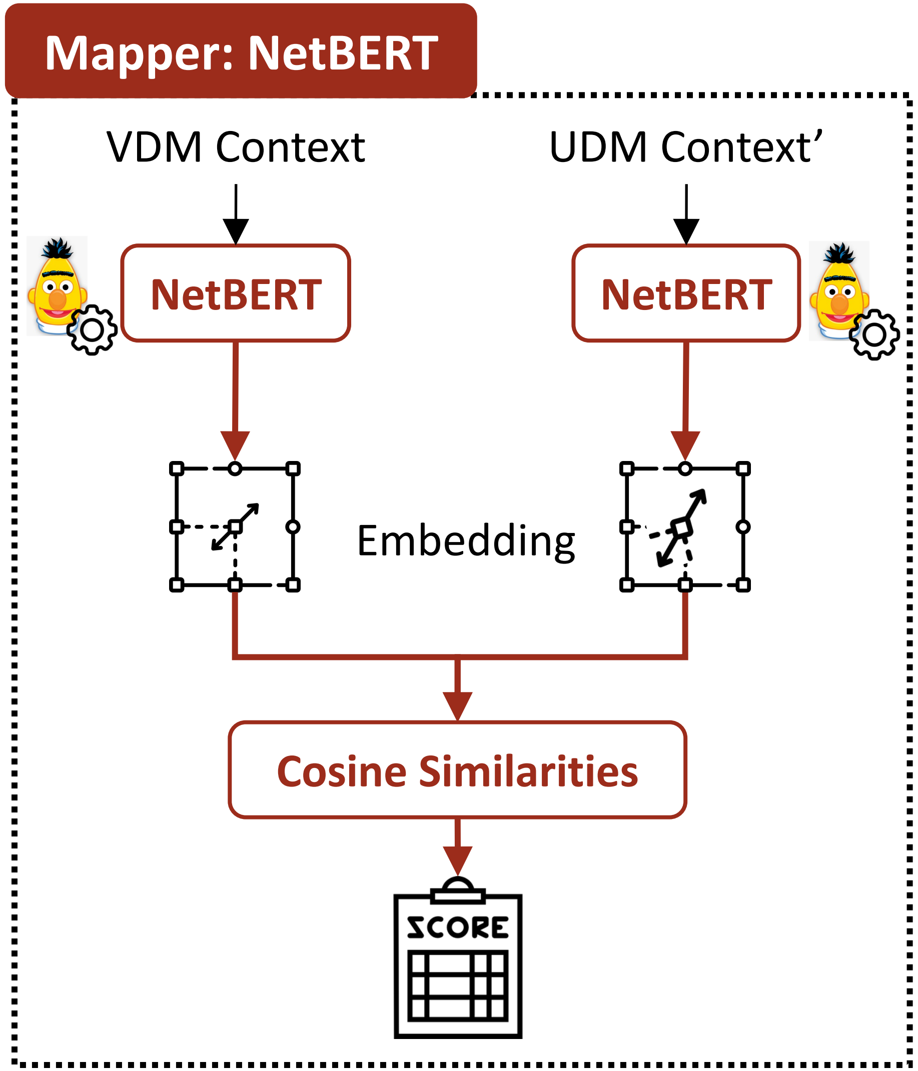
NAssim Mapper: NetBERT Training



*Pre-trained model**

UDM-VDM mappings(+)

Net-Config Domain Data



Mapper: NetBERT

VDM Context

UDM Context’

NetBERT

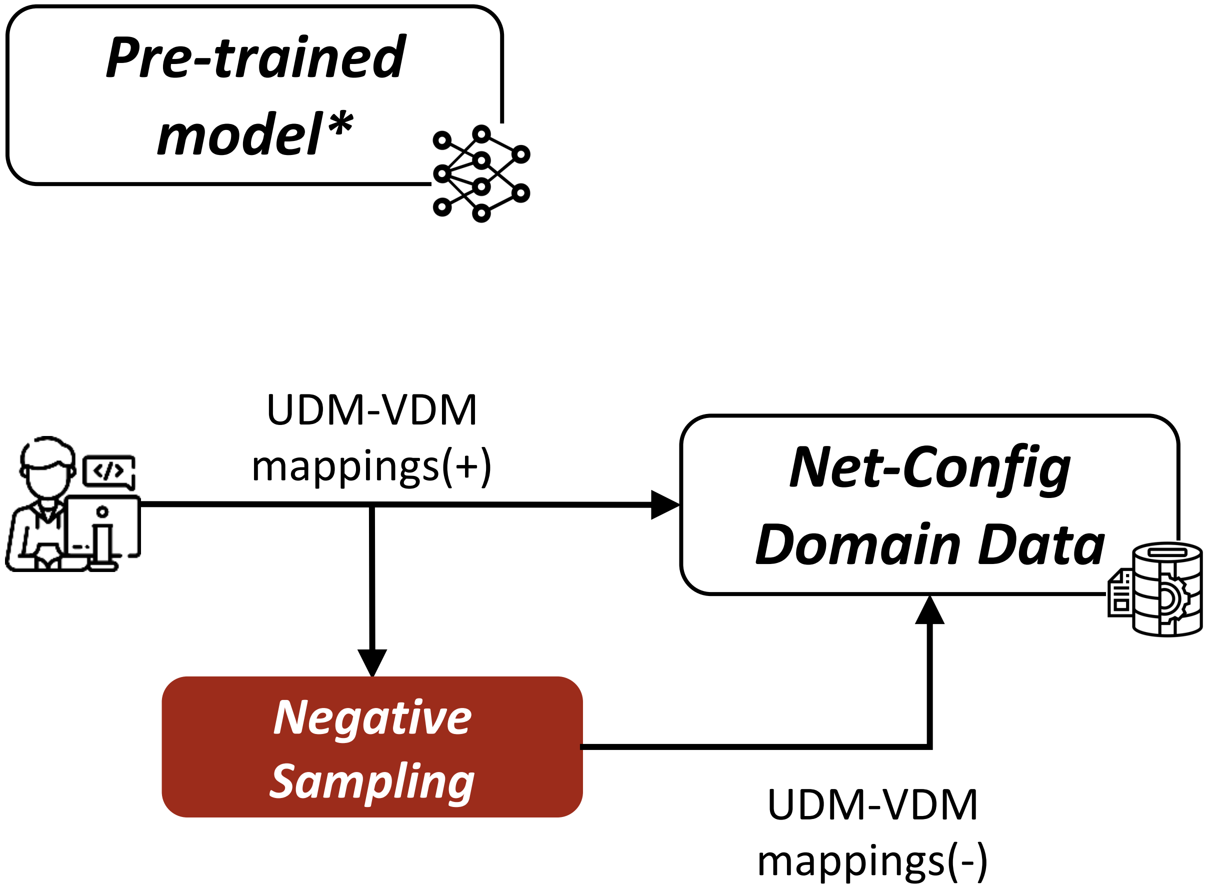
NetBERT

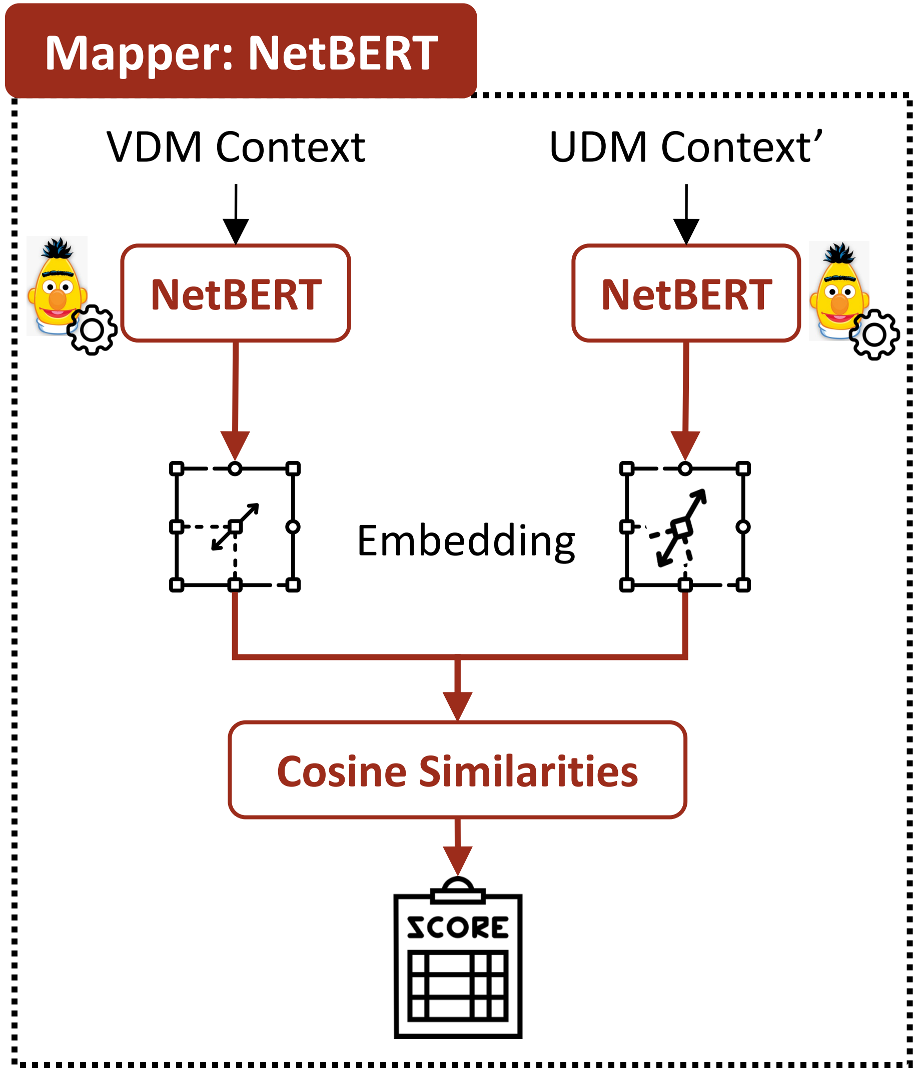
Embedding

Cosine Similarities

*Reimers, Nils, et al. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” EMNLP, 2019.

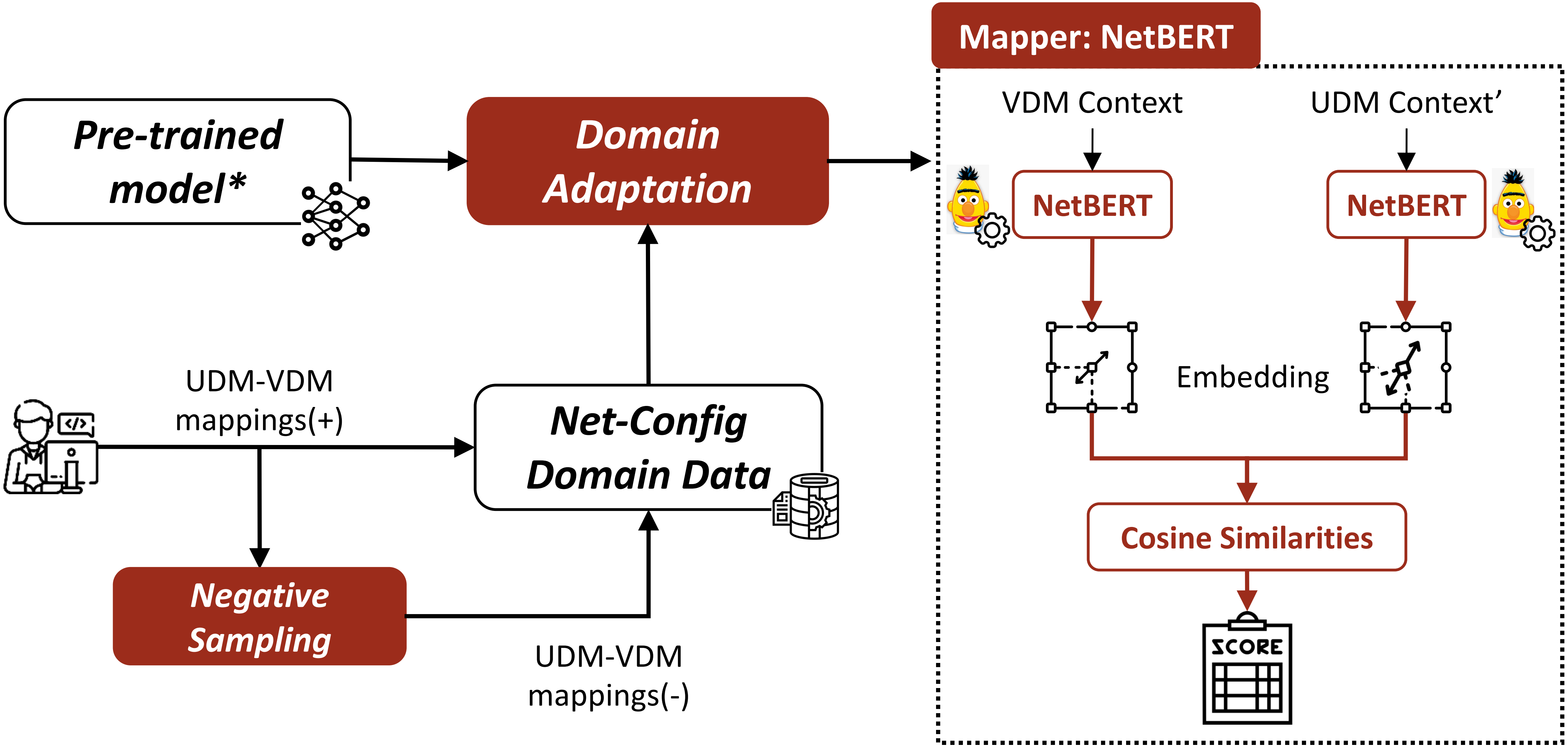
NAssim Mapper: NetBERT Training





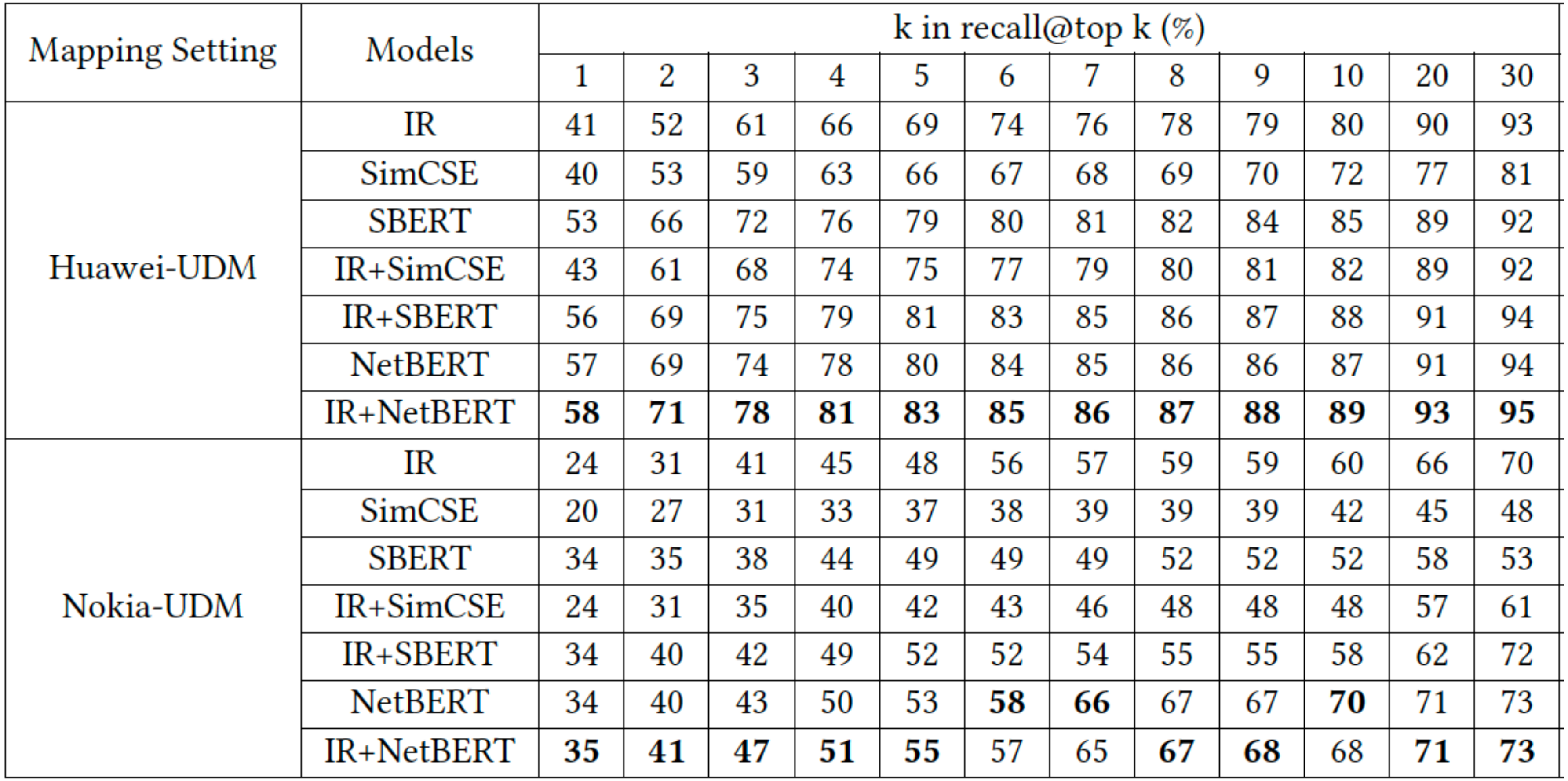
*Reimers, Nils, et al. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." EMNLP, 2019.

NAssim Mapper: NetBERT Training



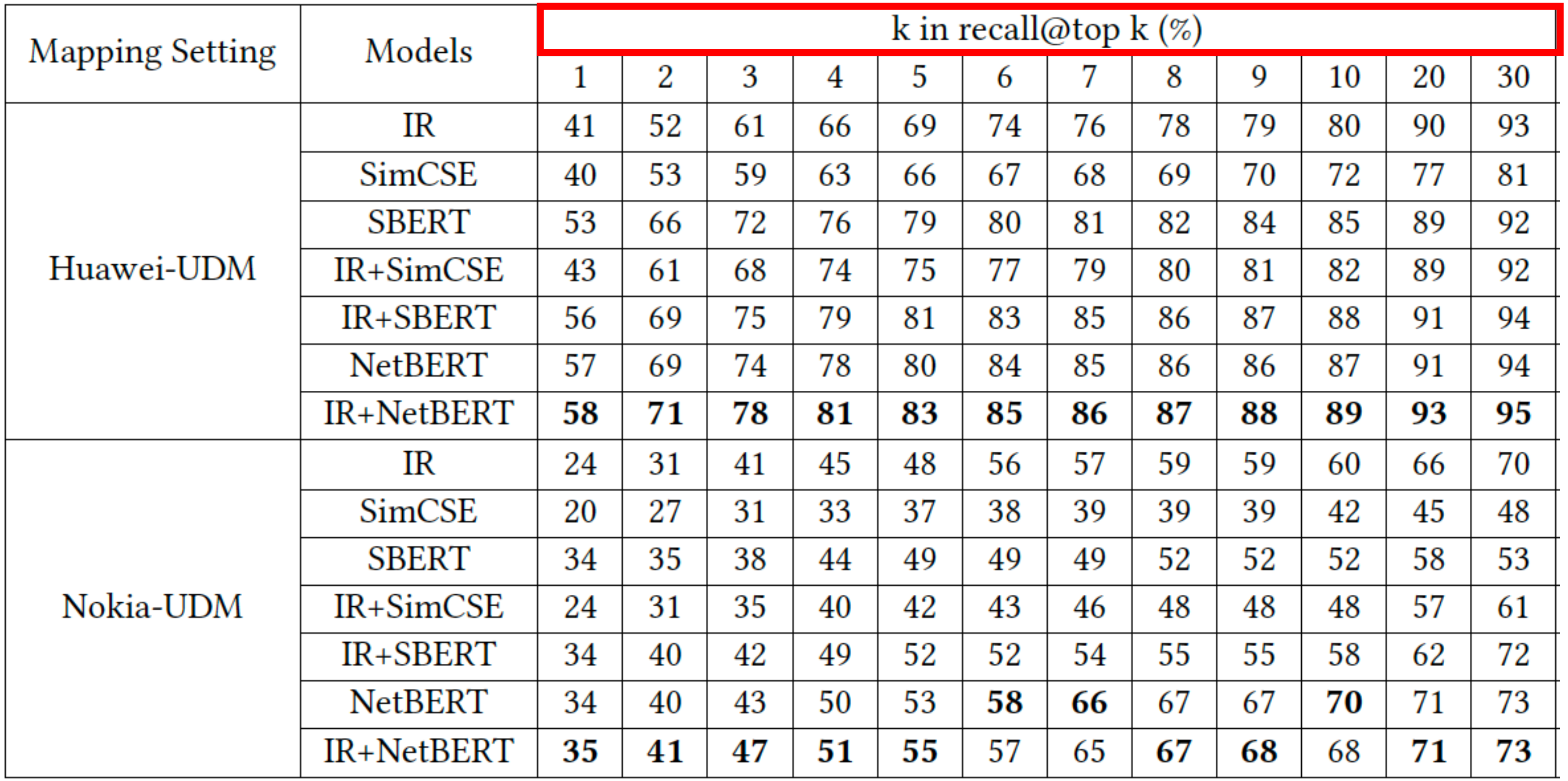
*Reimers, Nils, et al. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” EMNLP, 2019.

VDM-UDM Mapping Phase: Mapper



Mapping Setting	Models	k in recall@top k (%)											
		1	2	3	4	5	6	7	8	9	10	20	30
Huawei-UDM	IR	41	52	61	66	69	74	76	78	79	80	90	93
	SimCSE	40	53	59	63	66	67	68	69	70	72	77	81
	SBERT	53	66	72	76	79	80	81	82	84	85	89	92
	IR+SimCSE	43	61	68	74	75	77	79	80	81	82	89	92
	IR+SBERT	56	69	75	79	81	83	85	86	87	88	91	94
	NetBERT	57	69	74	78	80	84	85	86	86	87	91	94
	IR+NetBERT	**58**	**71**	**78**	**81**	**83**	**85**	**86**	**87**	**88**	**89**	**93**	**95**
Nokia-UDM	IR	24	31	41	45	48	56	57	59	59	60	66	70
	SimCSE	20	27	31	33	37	38	39	39	39	42	45	48
	SBERT	34	35	38	44	49	49	49	52	52	52	58	53
	IR+SimCSE	24	31	35	40	42	43	46	48	48	48	57	61
	IR+SBERT	34	40	42	49	52	52	54	55	55	58	62	72
	NetBERT	34	40	43	50	53	**58**	**66**	67	67	**70**	71	73
	IR+NetBERT	**35**	**41**	**47**	**51**	**55**	57	65	**67**	**68**	68	**71**	**73**

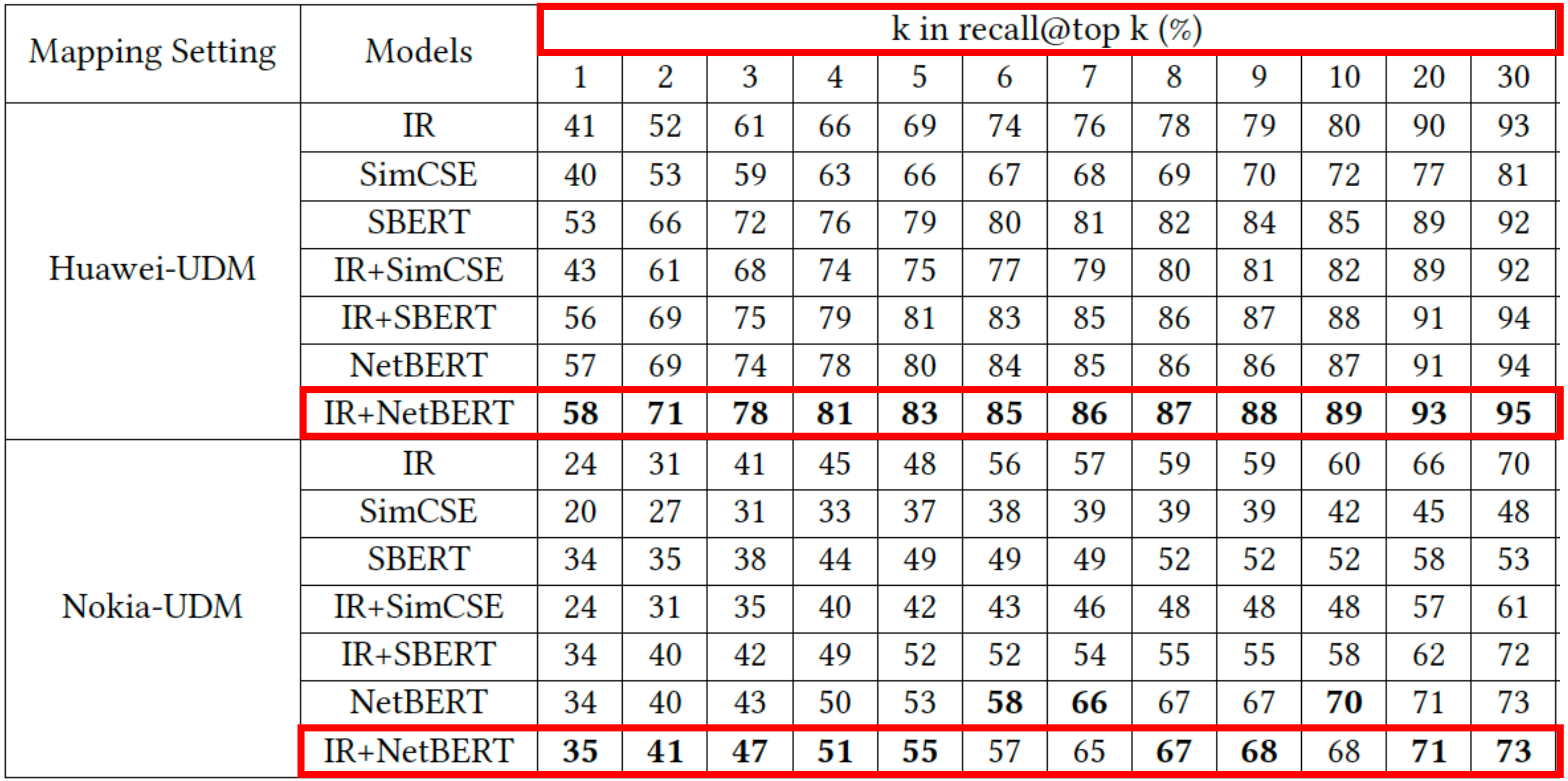
VDM-UDM Mapping Phase: Mapper



Mapping Setting	Models	k in recall@top k (%)											
		1	2	3	4	5	6	7	8	9	10	20	30
Huawei-UDM	IR	41	52	61	66	69	74	76	78	79	80	90	93
	SimCSE	40	53	59	63	66	67	68	69	70	72	77	81
	SBERT	53	66	72	76	79	80	81	82	84	85	89	92
	IR+SimCSE	43	61	68	74	75	77	79	80	81	82	89	92
	IR+SBERT	56	69	75	79	81	83	85	86	87	88	91	94
	NetBERT	57	69	74	78	80	84	85	86	86	87	91	94
	IR+NetBERT	**58**	**71**	**78**	**81**	**83**	**85**	**86**	**87**	**88**	**89**	**93**	**95**
Nokia-UDM	IR	24	31	41	45	48	56	57	59	59	60	66	70
	SimCSE	20	27	31	33	37	38	39	39	39	42	45	48
	SBERT	34	35	38	44	49	49	49	52	52	52	58	53
	IR+SimCSE	24	31	35	40	42	43	46	48	48	48	57	61
	IR+SBERT	34	40	42	49	52	52	54	55	55	58	62	72
	NetBERT	34	40	43	50	53	**58**	**66**	67	67	**70**	71	73
	IR+NetBERT	**35**	**41**	**47**	**51**	**55**	57	65	**67**	**68**	68	**71**	**73**

Recall@top-k denotes the percentage of test cases where the correct matching parameters are in top k recommendation by Mapper.

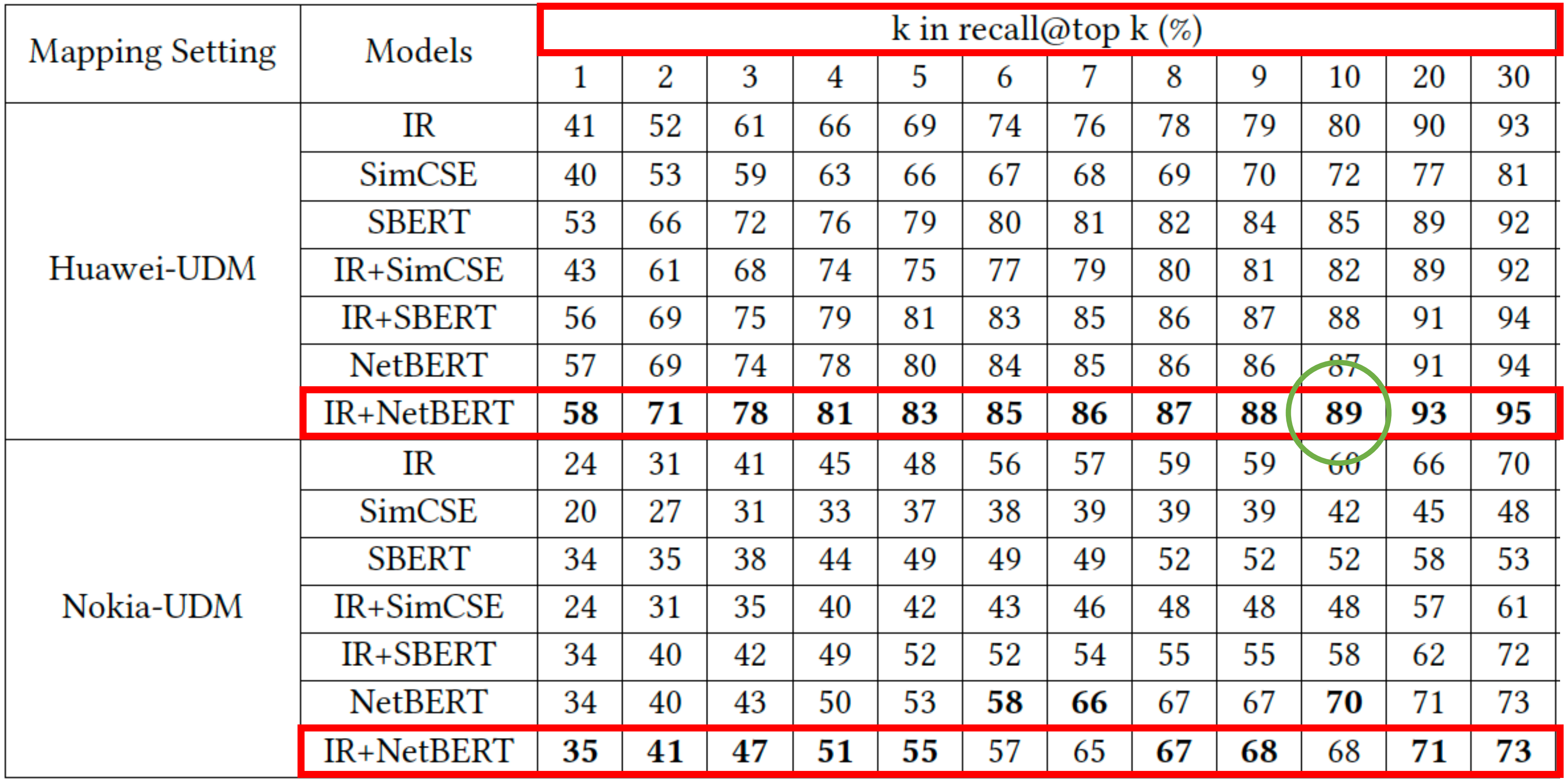
VDM-UDM Mapping Phase: Mapper



Mapping Setting	Models	k in recall@top k (%)											
		1	2	3	4	5	6	7	8	9	10	20	30
Huawei-UDM	IR	41	52	61	66	69	74	76	78	79	80	90	93
	SimCSE	40	53	59	63	66	67	68	69	70	72	77	81
	SBERT	53	66	72	76	79	80	81	82	84	85	89	92
	IR+SimCSE	43	61	68	74	75	77	79	80	81	82	89	92
	IR+SBERT	56	69	75	79	81	83	85	86	87	88	91	94
	NetBERT	57	69	74	78	80	84	85	86	86	87	91	94
	IR+NetBERT	**58**	**71**	**78**	**81**	**83**	**85**	**86**	**87**	**88**	**89**	**93**	**95**
Nokia-UDM	IR	24	31	41	45	48	56	57	59	59	60	66	70
	SimCSE	20	27	31	33	37	38	39	39	39	42	45	48
	SBERT	34	35	38	44	49	49	49	52	52	52	58	53
	IR+SimCSE	24	31	35	40	42	43	46	48	48	48	57	61
	IR+SBERT	34	40	42	49	52	52	54	55	55	58	62	72
	NetBERT	34	40	43	50	53	**58**	**66**	67	67	**70**	71	73
	IR+NetBERT	**35**	**41**	**47**	**51**	**55**	57	65	**67**	**68**	68	**71**	**73**

Recall@top-k denotes the percentage of test cases where the correct matching parameters are in top k recommendation by Mapper.

VDM-UDM Mapping Phase: Mapper



Mapping Setting	Models	k in recall@top k (%)											
		1	2	3	4	5	6	7	8	9	10	20	30
Huawei-UDM	IR	41	52	61	66	69	74	76	78	79	80	90	93
	SimCSE	40	53	59	63	66	67	68	69	70	72	77	81
	SBERT	53	66	72	76	79	80	81	82	84	85	89	92
	IR+SimCSE	43	61	68	74	75	77	79	80	81	82	89	92
	IR+SBERT	56	69	75	79	81	83	85	86	87	88	91	94
	NetBERT	57	69	74	78	80	84	85	86	86	87	91	94
	IR+NetBERT	**58**	**71**	**78**	**81**	**83**	**85**	**86**	**87**	**88**	**89**	**93**	**95**
Nokia-UDM	IR	24	31	41	45	48	56	57	59	59	60	66	70
	SimCSE	20	27	31	33	37	38	39	39	39	42	45	48
	SBERT	34	35	38	44	49	49	49	52	52	52	58	53
	IR+SimCSE	24	31	35	40	42	43	46	48	48	48	57	61
	IR+SBERT	34	40	42	49	52	52	54	55	55	58	62	72
	NetBERT	34	40	43	50	53	**58**	**66**	67	67	**70**	71	73
	IR+NetBERT	**35**	**41**	**47**	**51**	**55**	57	65	**67**	**68**	68	**71**	**73**

9.1x

Recall@top-k denotes the percentage of test cases where the correct matching parameters are in top k recommendation by Mapper.

Key Takeaways

Key Takeways

- Soft-defined network assimilation (SNA) is essential for managing multi-vendor network. Our solution NAssim seeks to transform tedious and error-prone SNA process to automated and efficient manner.

Key Takeways

- Soft-defined network assimilation (SNA) is essential for managing multi-vendor network. Our solution NAssim seeks to transform tedious and error-prone SNA process to automated and efficient manner.

- Configuration manuals, as human-written documents are not fully reliable, including inevitable errors and ambiguities.

Key Takeways

- Soft-defined network assimilation (SNA) is essential for managing multi-vendor network. Our solution NAssim seeks to transform tedious and error-prone SNA process to automated and efficient manner.

- Configuration manuals, as human-written documents are not fully reliable, including inevitable errors and ambiguities.

- NAssim features a unified parser framework, a rigorous validator and a mapper using the domain-adapted BERT model to produce human-comprehensible recommended mapping between the validated configuration model and the one in the SDN controller.

Key Takeways

- Soft-defined network assimilation (SNA) is essential for managing multi-vendor network. Our solution NAssim seeks to transform tedious and error-prone SNA process to automated and efficient manner.

- Configuration manuals, as human-written documents are not fully reliable, including inevitable errors and ambiguities.

- NAssim features a unified parser framework, a rigorous validator and a mapper using the domain-adapted BERT model to produce human-comprehensible recommended mapping between the validated configuration model and the one in the SDN controller.

- We release a validated and expert-curated dataset of parsed manual corpus for future research. (https://github.com/AmyWorkspace/nassim)

Thank you!

Theory Lab, Hong Kong Research Center (HKRC), Huawei Technologies, Co. Ltd.

Huangxun CHEN: chen.huangxun@huawei.com

Gong ZHANG: nicholas.zhang@huawei.com

Welcome to join us and do impactful works！

HUAWEI