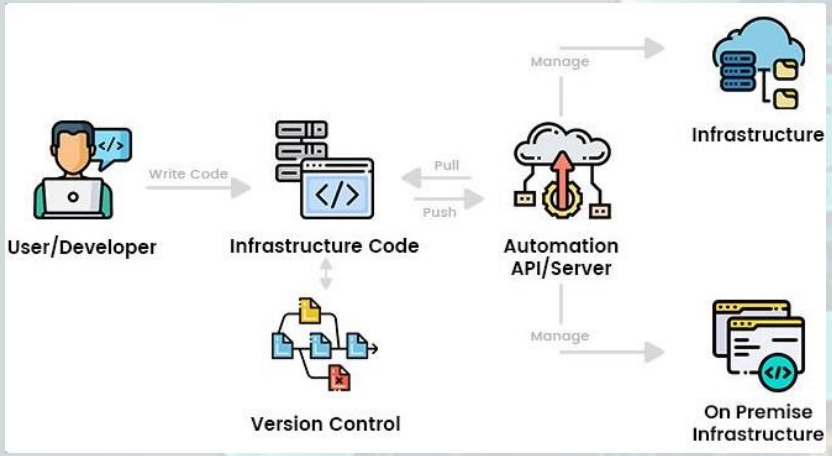
Towards a Taxonomy of Infrastructure as Code (IaC) Misconfigurations: An Ansible Study

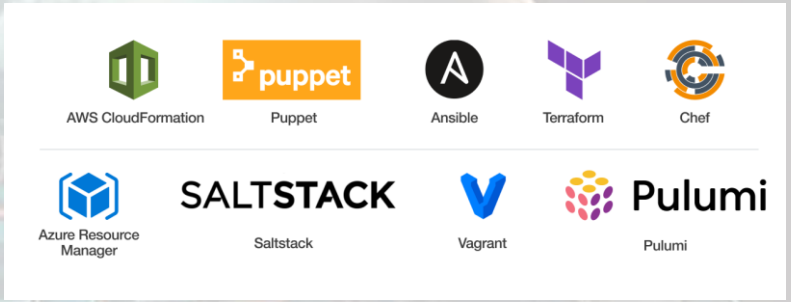
Roya Nasiri, Indika Kumara, Damian A. Tamburri, Willem-Jan van den Heuvel

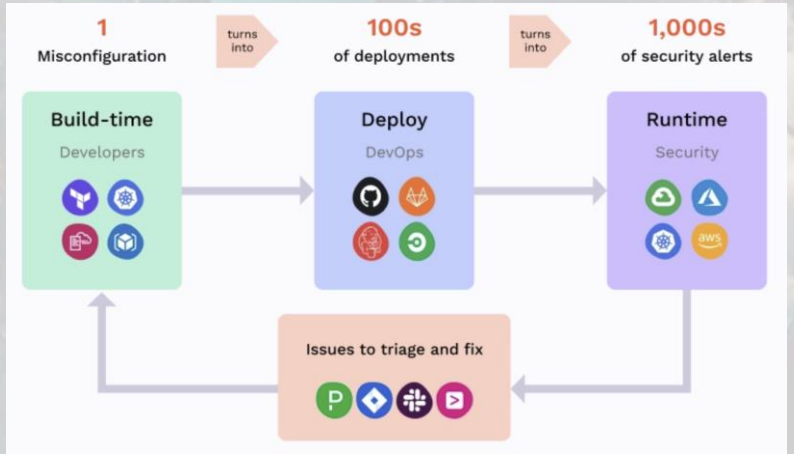
June 2024

Introduction | Motivation



Infrastructure as Code





Introduction | Problem Definition

IaC Language → API Documentation → Complexity → Research

```yaml
# author of the playbook - Sandeep kumar patel
--- # this playbook are show the current date in the system
  - name: Shell Examples
    hosts: all
    tasks:


    - name: sending bash file to node
      copy:
        src: /home/devops/sample.sh
        dest: /home/devops/
    - name: running bash fike
      shell: /bin/bash/ /home/devops/sample.sh



    - name: Check Date with Shell command
      shell:
         "date"

```

Parameters

Parameter	Comments
ci boolean	Install packages based on package-lock file, same as running `npm ci`. **Choices:** • `false` ← (default) • `true`
executable path	The executable location for npm. This is useful if you are using a version manager, such as nvm.
global boolean	Install the node.js library globally. **Choices:** • `false` ← (default) • `true`
ignore_scripts boolean	Use the `--ignore-scripts` flag when installing. **Choices:** • `false` ← (default) • `true`
name string	The name of a node.js library to install.
no_bin_links boolean *added in community.general 2.5.0*	Use the `--no-bin-links` flag when installing. **Choices:** • `false` ← (default) • `true`

IaC Misconfigurations

- **Empirical Studies**
- **PracExtractor**
- **Defect IaC Taxonomy**

Introduction | Problem Definition

Research

- **Empirical Studies**
- **PracExtractor**
- **Defect IaC Taxonomy**

Research Question

What categories of misconfigurations can appear in infrastructure as code (IaC)?

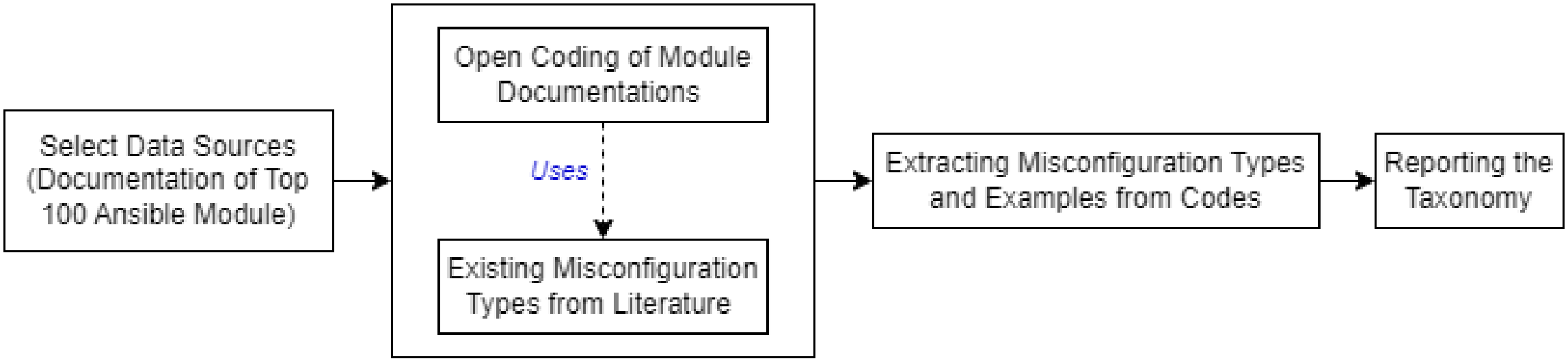
Literature Review | Background


```yaml
- name: Install Tomcat
  tasks:
    - name: Install Java
      yum:
        name: java-1.8.0-openjdk
        state: present
    - name: Download Tomcat
      get_url:
        url: "https://archive.apache.org/.../apache-tomcat-9.0.54.tar.gz"
        dest: /opt/
    - name: Extract Tomcat
      unarchive:
        src: /opt/apache-tomcat-9.0.54.tar.gz
        dest: /opt/
        remote_src: yes
    - name: Configure Tomcat Users
      template:
        src: tomcat-users.xml.j2
        dest: /opt/apache-tomcat-9.0.54/conf/tomcat-users.xml
    - name: Start Tomcat Service
      service:
        name: tomcat
        state: started
        enabled: yes
```

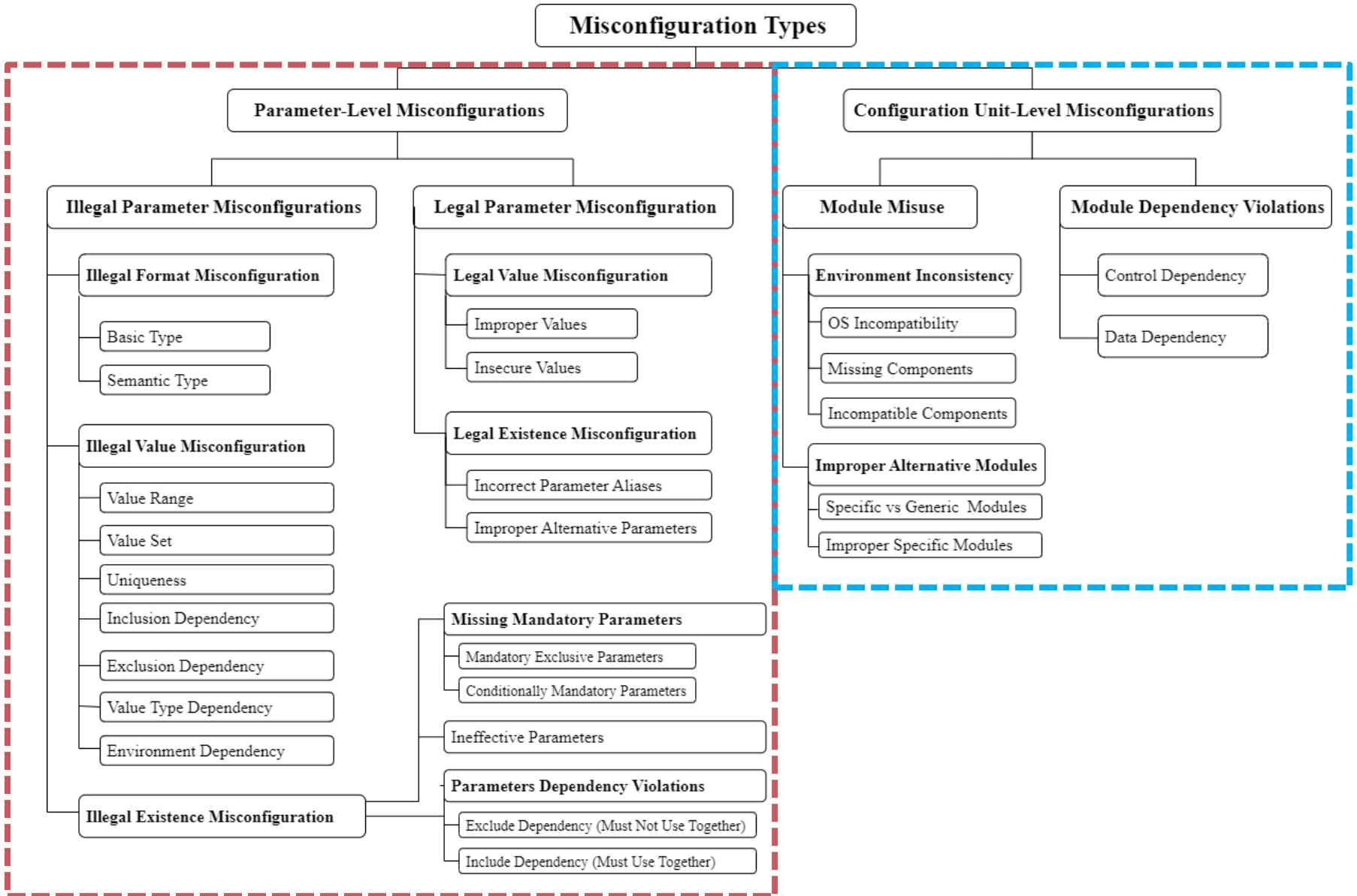
Literature Review | Related Works

Related Works	An Empirical Study on Configuration Errors in Commercial and Open Source Systems (CentOS, MySQL,Apache HTTP Server, and OpenLDAP)	Do Not Blame Users for Misconfigurations (a major U.S. storage vendor, and six open-source server software including Apache, MySQL, PostgreSQL, OpenLDAP, VSFTP, and Squid)	PracExtractor: Extracting Configuration Good Practices from Manuals to Detect Server Misconfigurations (MySQL, Httpd, HBase, HDFS, Spark, Squid)	Gang of Eight: A Defect Taxonomy for Infrastructure as Code Scripts (Puppet)
Misconfiguration Types	Parameter Misconfigurations	Attribute constraints	General Advice	Conditional Defects
	Illegal Parameter Misconf	Data Type constraints	Clear Specifications	Configuration Defects
	Illegal Format Misconf	Basic Type	Value	Dependency Defects
	Lexical Mistakes	Semantic Type	Correlation	Idempotency Defects
	Syntax Mistakes	Value Range constraints	Usage	Security Defects
	Illegal Value Misconf	Correlation Constraints	Property	Documentation Defects
	Value inconsistency	Control Dependency constraints		Service Defects
	Environment inconsistency	Value Relationship constraints		Syntax Defects
	Legal Parameter Misconf			
	Component Misconfigurations			
	Compatibility Misconfigurations			
	Missing component			
	File format			
	Insufficient resource			
	Placement			
	Stale data			

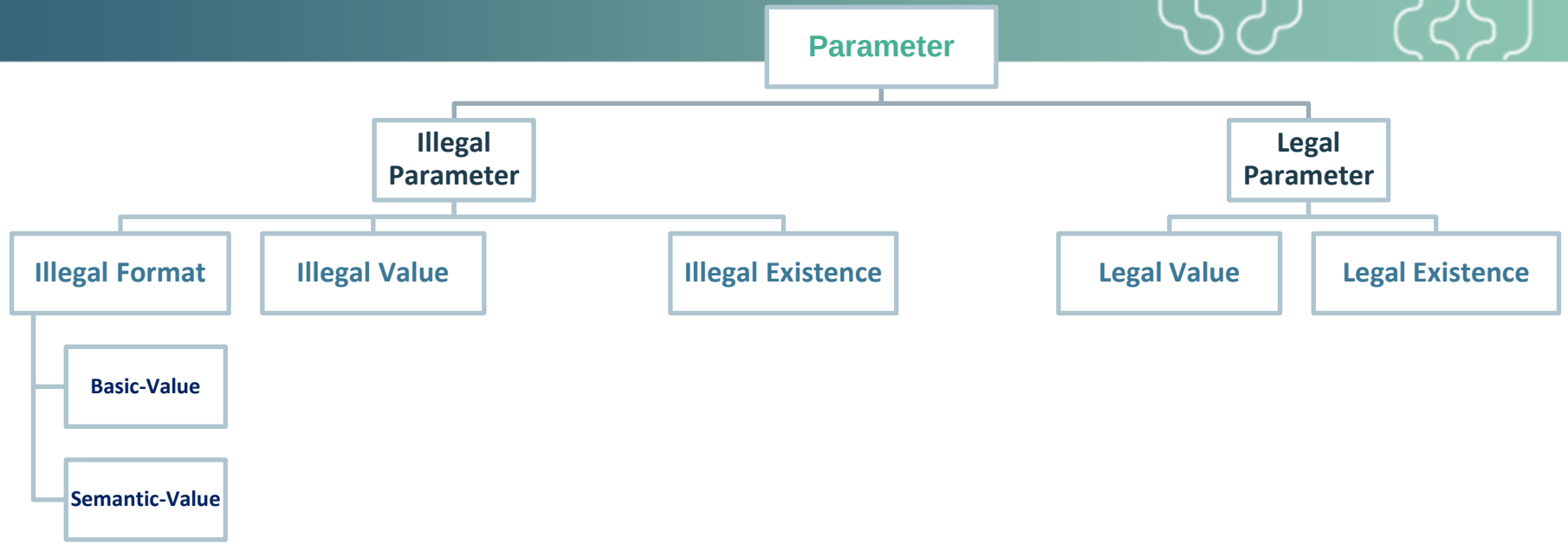
Methodology



IaC Misconfiguration Taxonomy

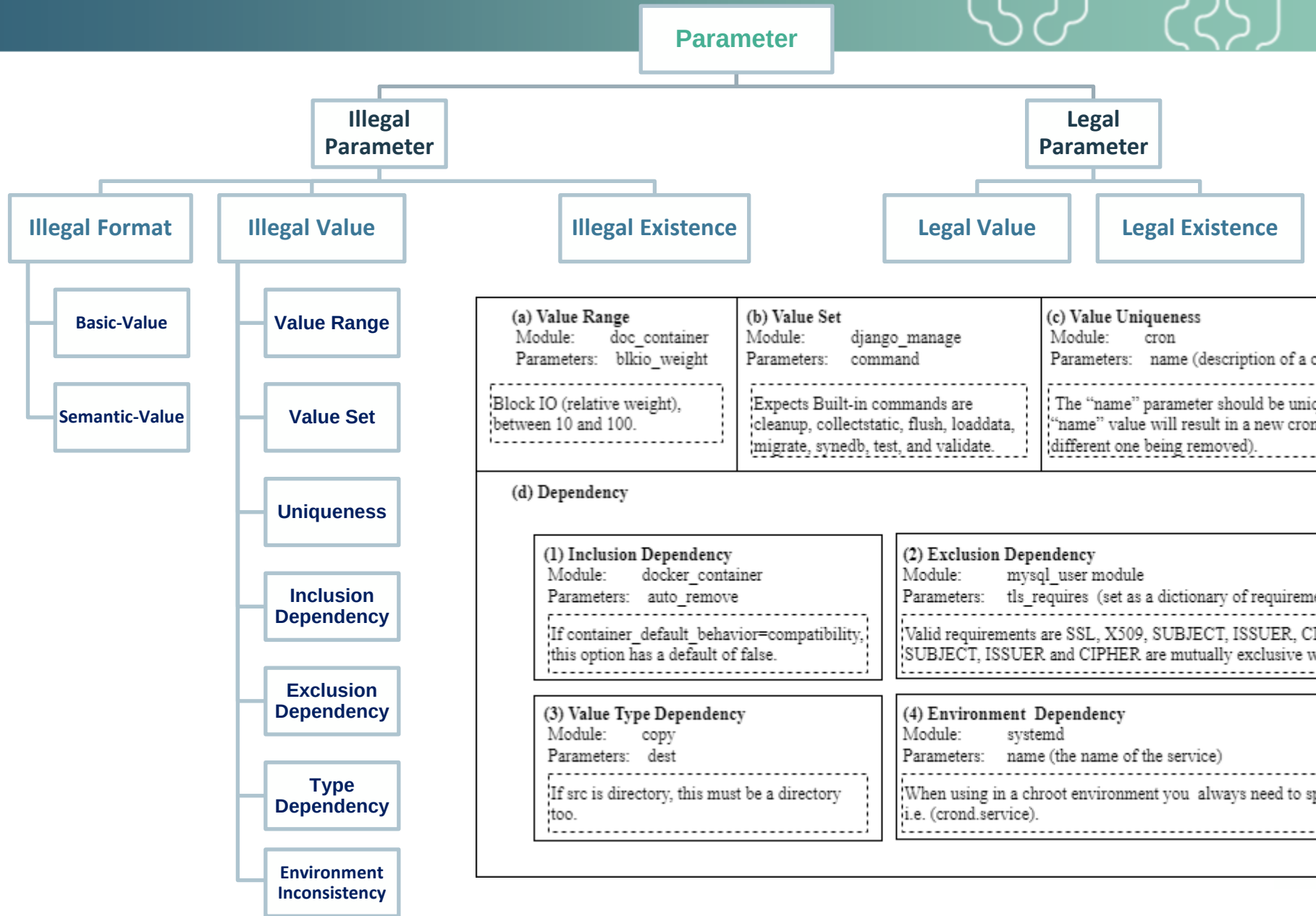


IaC Misconfiguration Taxonomy



(a) Basic Type	(a) Semantic Type
Module: iptables Parameters: tcp-flags	Module: docker_container Parameters: default_host_ip (string data type)
Expects a dict with the two keys flags and flags_set.	Must be an empty string, an IPv4 address, or an IPv6 address.

IaC Misconfiguration Taxonomy



(a) Value Range
Module: doc_container
Parameters: blkio_weight

Block IO (relative weight), between 10 and 100.

(b) Value Set
Module: django_manage
Parameters: command

Expects Built-in commands are cleanup, collectstatic, flush, loaddata, migrate, synedb, test, and validate.

(c) Value Uniqueness
Module: cron
Parameters: name (description of a crontab entry)

The "name" parameter should be unique, and changing the "name" value will result in a new cron task being created (or a different one being removed).

(d) Dependency

(1) Inclusion Dependency
Module: docker_container
Parameters: auto_remove

If container_default_behavior=compatibility, this option has a default of false.

(2) Exclusion Dependency
Module: mysql_user module
Parameters: tls_requires (set as a dictionary of requirements)

Valid requirements are SSL, X509, SUBJECT, ISSUER, CIPHER. SUBJECT, ISSUER and CIPHER are mutually exclusive with SSL and X509.

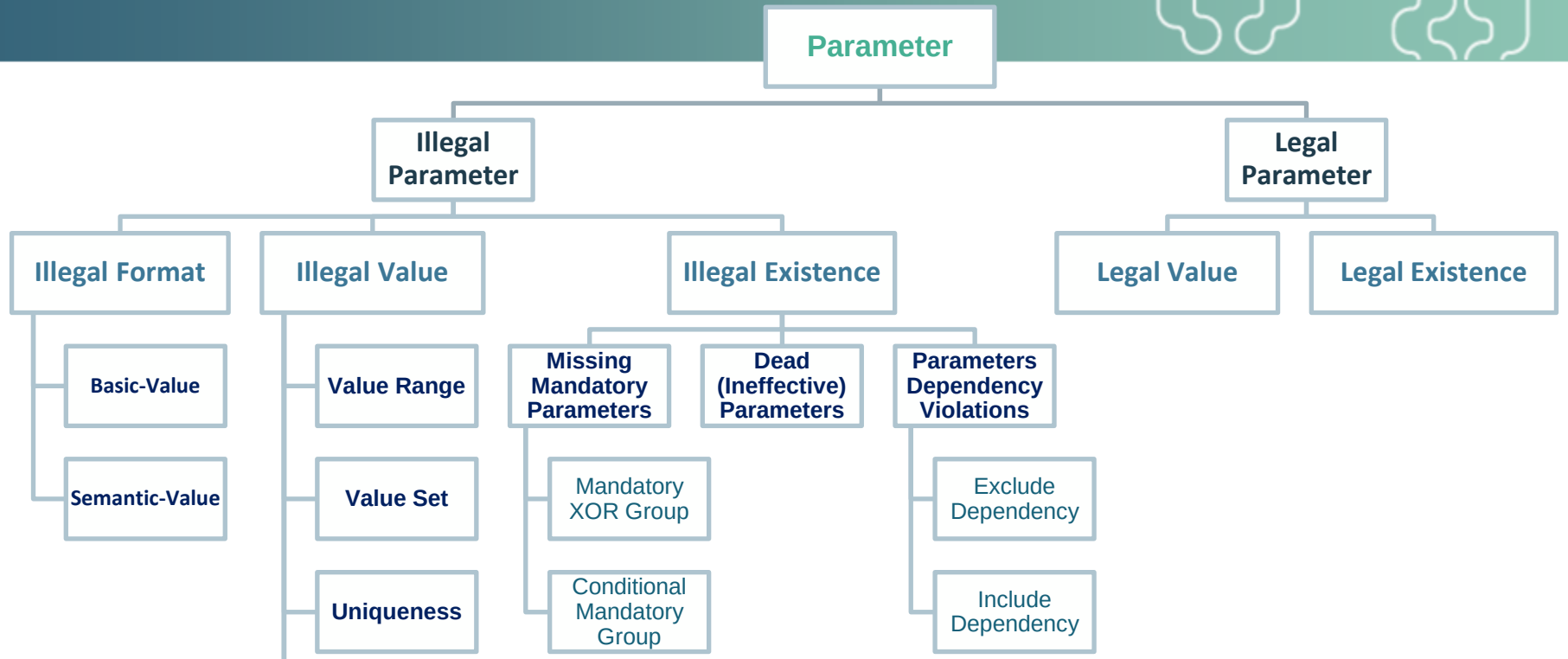
(3) Value Type Dependency
Module: copy
Parameters: dest

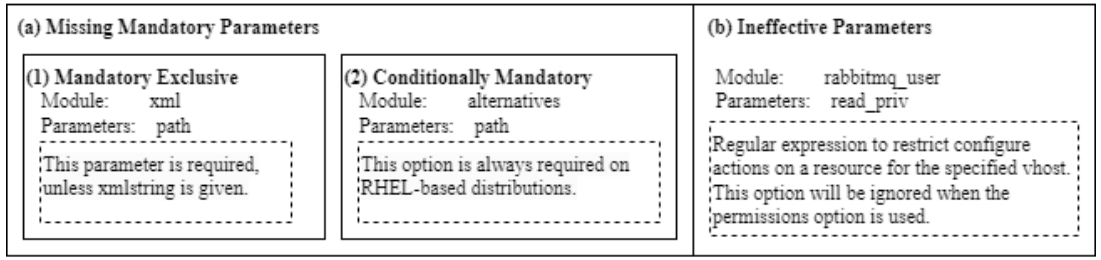
If src is directory, this must be a directory too.

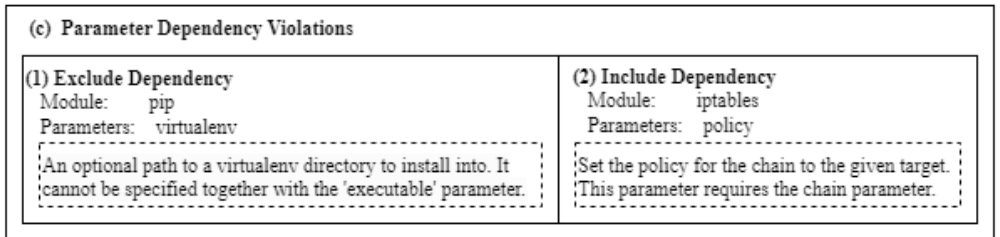
(4) Environment Dependency
Module: systemd
Parameters: name (the name of the service)

When using in a chroot environment you always need to specify the full name (i.e. (crond.service).

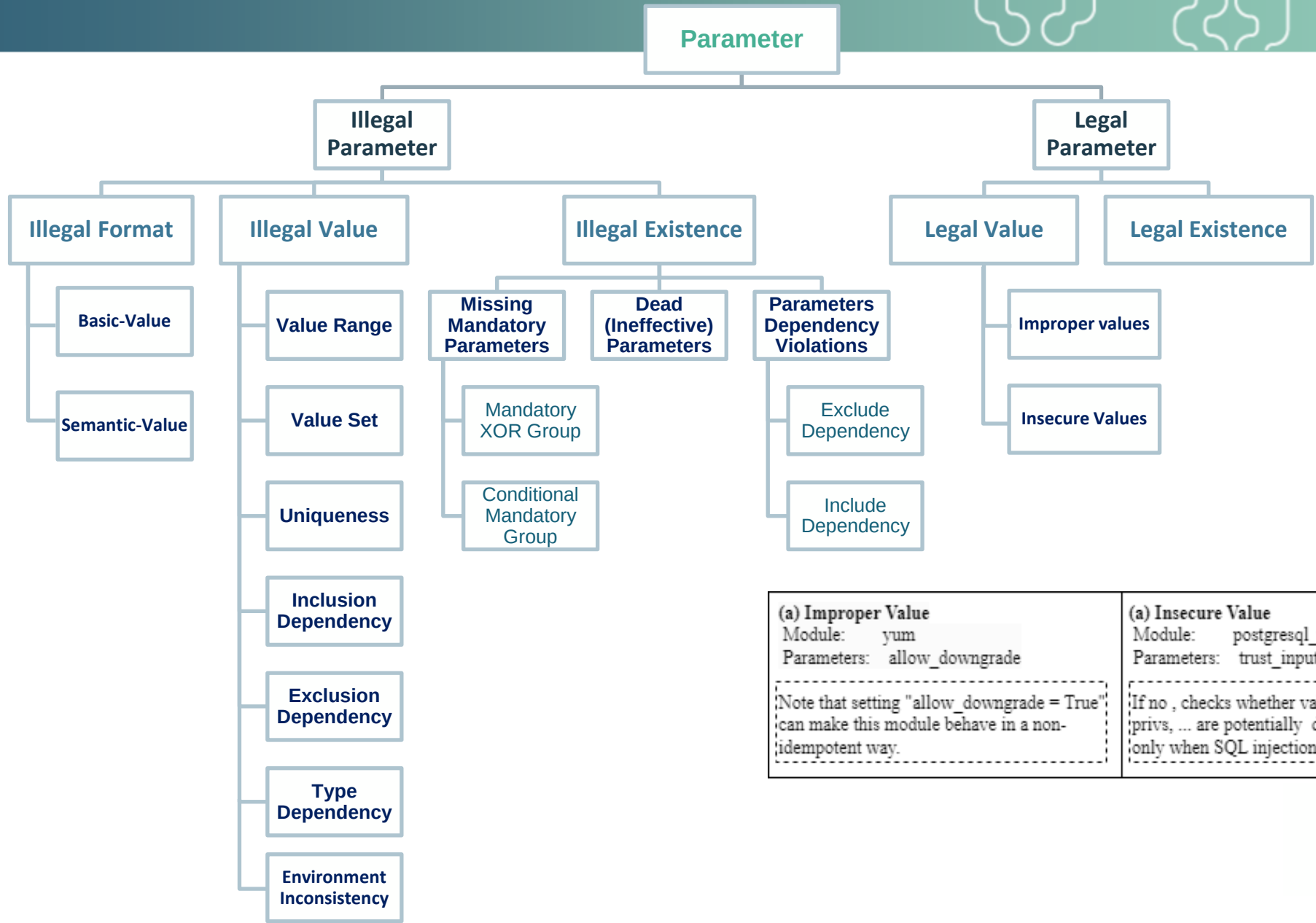
IaC Misconfiguration Taxonomy





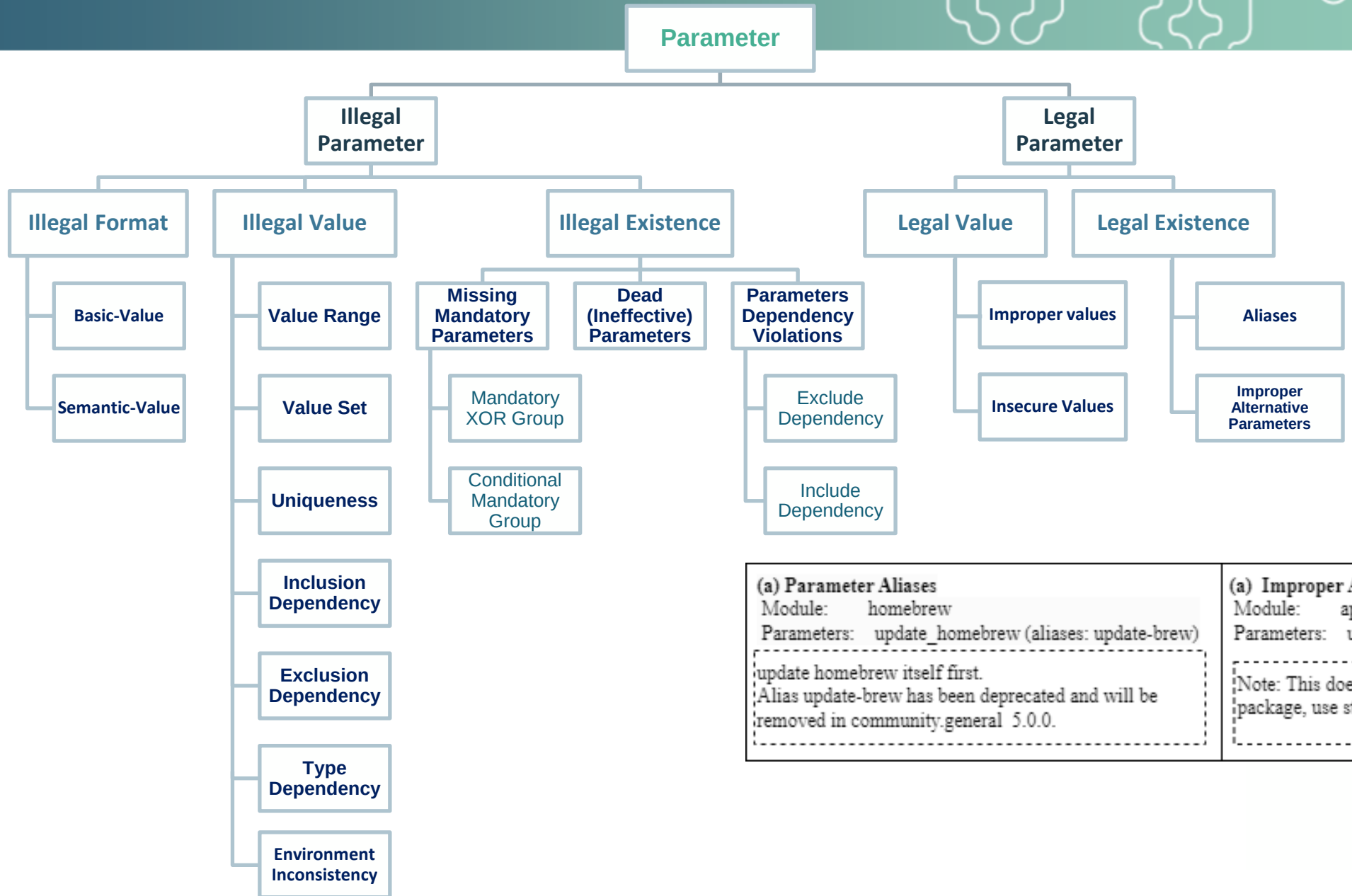


IaC Misconfiguration Taxonomy



(a) Improper Value	(a) Insecure Value
Module: yum Parameters: allow_downgrade	Module: postgresql_user Parameters: trust_input (default yes)
Note that setting "allow_downgrade = True" can make this module behave in a non-idempotent way.	If no , checks whether values of options name, password, privs, ... are potentially dangerous. It makes sense to use no only when SQL injections through the options are possible.

IaC Misconfiguration Taxonomy



(a) Parameter Aliases

Module: homebrew

Parameters: update_homebrew (aliases: update-brew)

update homebrew itself first.
Alias update-brew has been deprecated and will be removed in community.general 5.0.0.

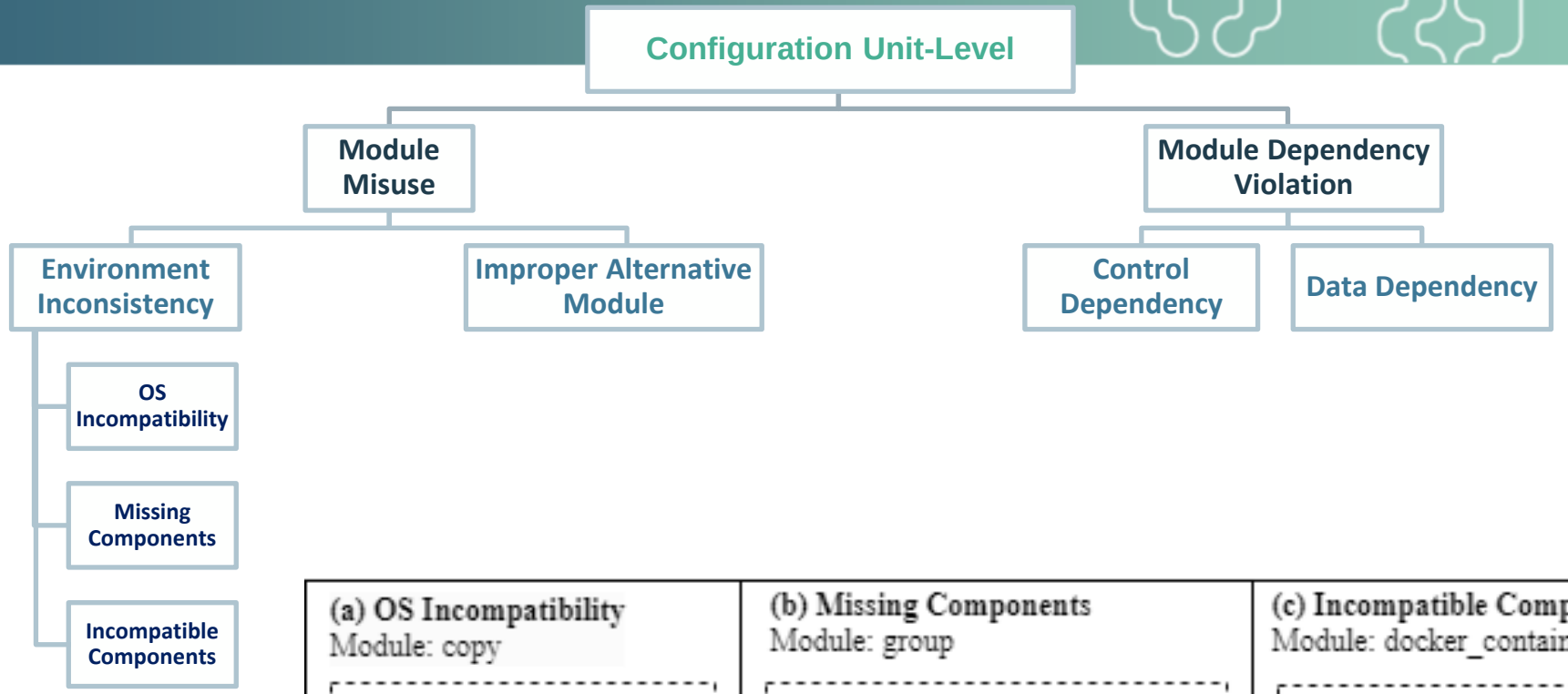
(a) Improper Alternative Parameters

Module: apt

Parameters: upgrade

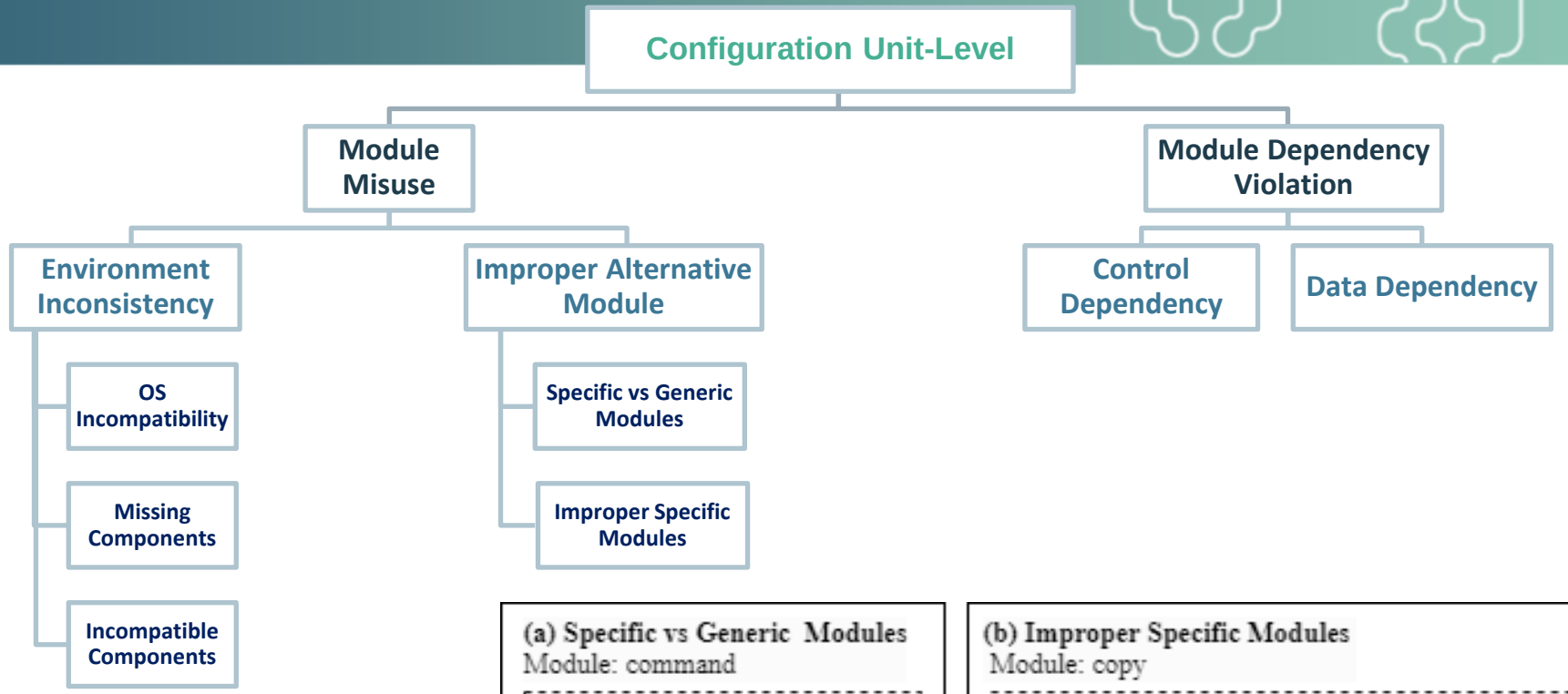
Note: This does not upgrade a specific package, use state=latest for that.

IaC Misconfiguration Taxonomy



(a) OS Incompatibility Module: copy	(b) Missing Components Module: group	(c) Incompatible Components Module: docker_container
For Windows targets, use the ansible.windows.win_copy module instead.	These requirements are needed on the host that executes this module: groupadd, groupdel, groupmod	Docker API >= 1.20 chocolatey >= 0.10.5 (will be upgraded if older)

IaC Misconfiguration Taxonomy



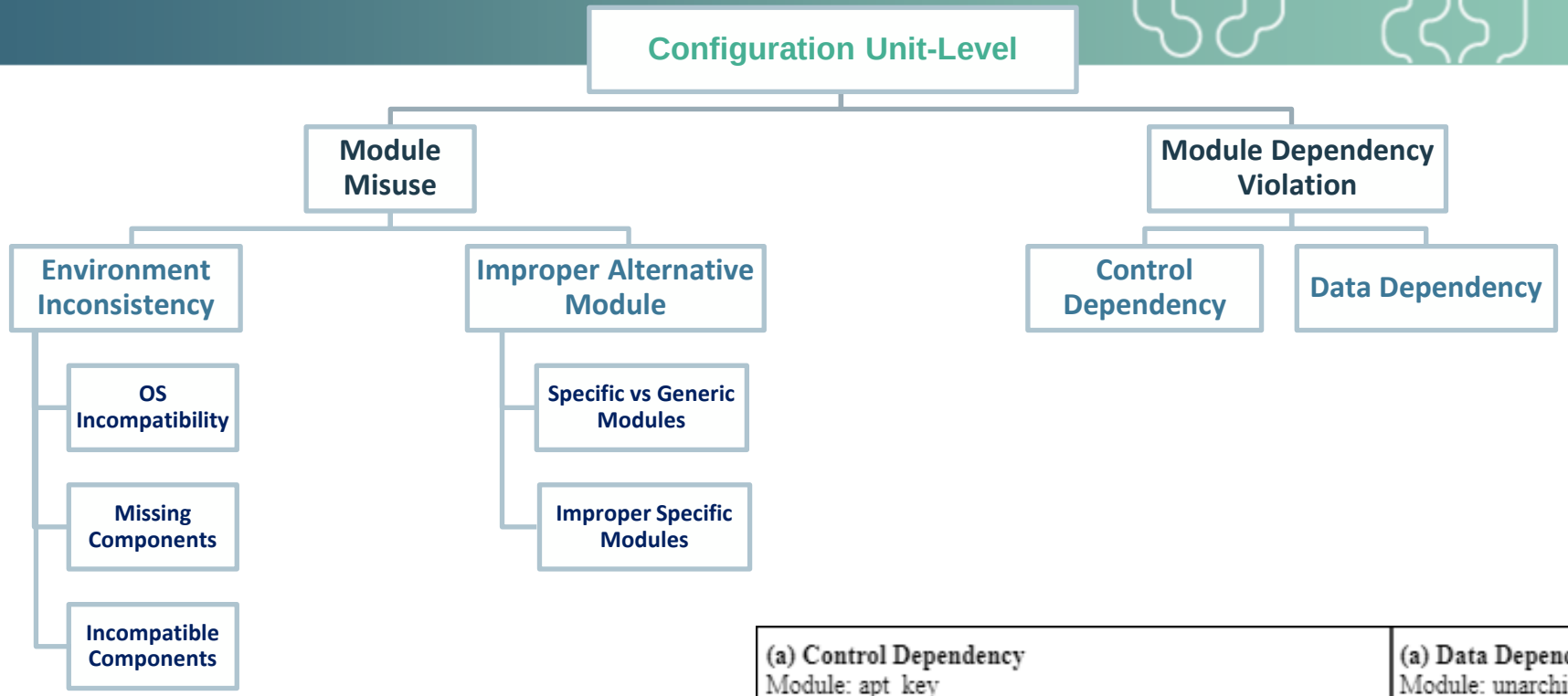
(a) Specific vs Generic Modules
Module: command

For rebooting systems, use the ansible.builtin.reboot module.

(b) Improper Specific Modules
Module: copy

If you need variable interpolation in copied files, use the ansible.builtin.template module. Using a variable in the content field will result in unpredictable output.

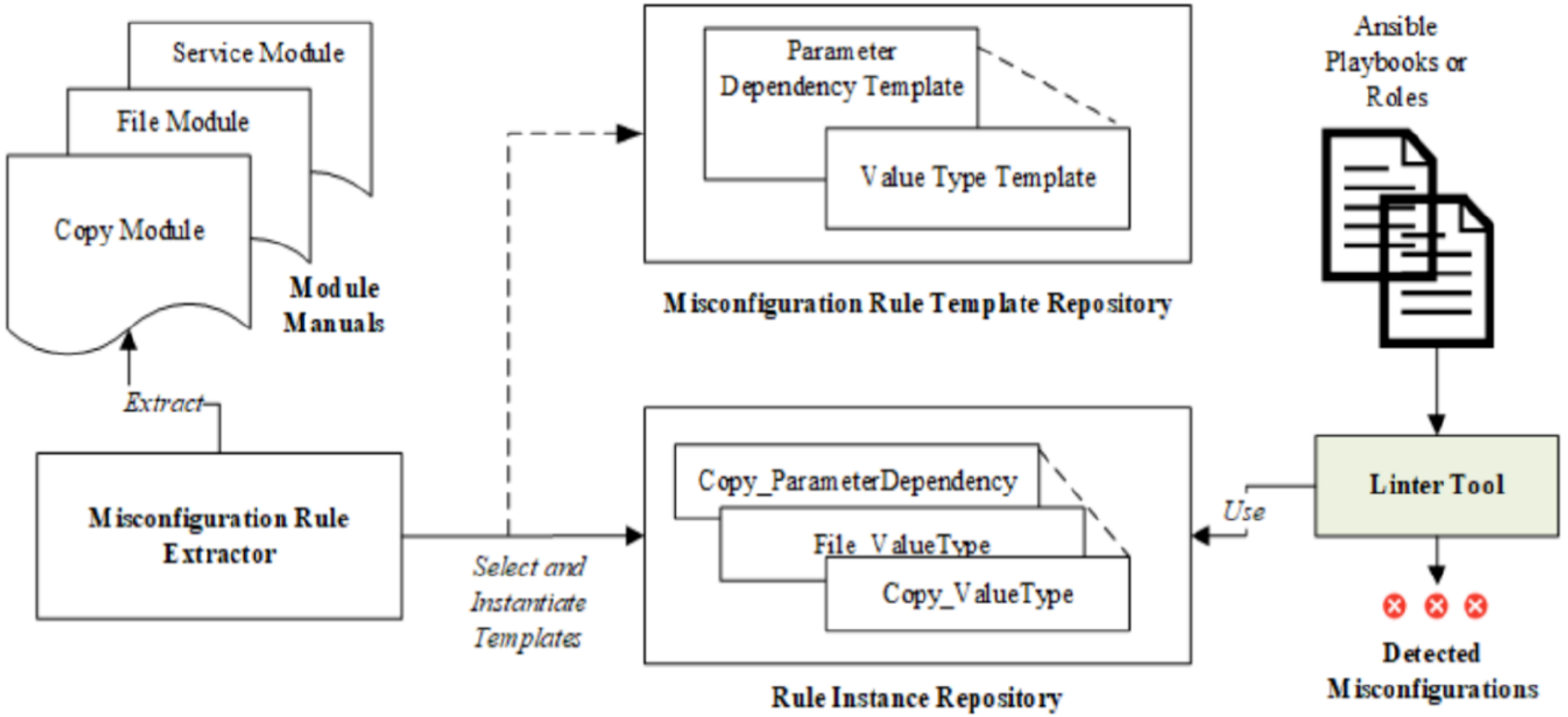
IaC Misconfiguration Taxonomy



(a) Control Dependency	(a) Data Dependency
Module: apt_key	Module: unarchive and get_url
Adding a new key requires an apt cache update (e.g. using the ansible.builtin.apt module's update_cache option)	If checksum validation is desired, use ansible.builtin.get_url or ansible.builtin.uri instead to fetch the file and set remote_src=yes.

Tool

MisConfLinter Tool



Conclusion

Result:

A comparative analysis across widely used IaC tools today, namely Terraform, Pulumi, and Puppet, leveraging their respective API documentation, shows the proposed taxonomy for identifying IaC misconfigurations is not limited to Ansible; rather, it serves as a comprehensive framework applicable to a range of IaC platforms.

Future works:

The misconfiguration detection tool will be extended to cover the complete taxonomy. The tool will then be used to assess the occurrence of misconfigurations in open-source IaC repositories.

A valuable next step would involve conducting semi-structured interviews with IaC developers to gather their perceptions and feedback on the proposed taxonomy. This qualitative approach could provide deeper insights and validate the applicability of the taxonomy in real-world scenarios.

Thank you for your attention!

Jheronimus Academy of Data Science