Evolveum

Policies and Rules in MidPoint

Radovan Semančík, September 2025
Software Architect

Agenda

- Identity governance
- Governance policies
- Inside policies
- Policy Rules
- Compliance

Identity Governance and Administration (IGA)

- IGA is a bridge between business and cybersecurity
- High-level (business oriented) policies and rules
- Translating business policies to technical implementation

- Access control governance: ***Why*** *does user have access?*
- Responsibility: *Who is* ***responsible*** *for what?*
- Order: *Maintaining* ***inventory*** *of all identities*

Governance Policies

- Named set of rules with **business meaning**
- Examples:
  - Classifications: *public, restricted, TLP:Green, privileged access, ...*
  - Clearances: *NDA signed, security training passed, ...*
  - Approval: *approval by manager, approval by security team, require approver, ...*
  - Asset management: *require owner, require classification, ...*
  - Organizational management: *require manager, require staff, ...*
  - Combinations of all of the above

DEMO

Governance Policy: Require Owner

midPoint 4.10 (development)

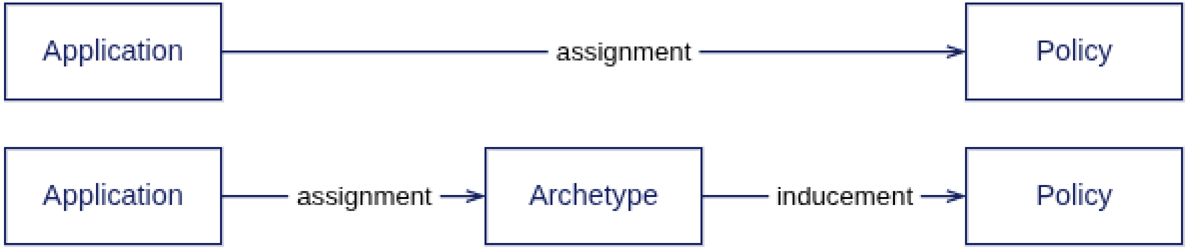
Configuration:

https://github.com/Evolveum/midpoint-samples/tree/master/samples/compliance

Documentation:

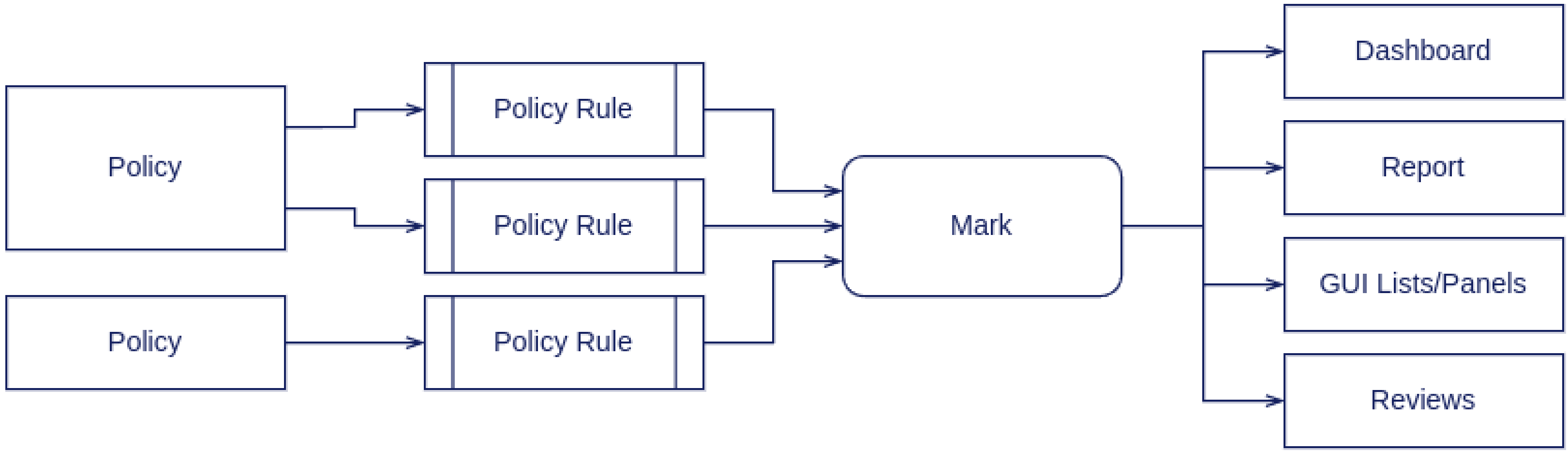
https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/identity-governance-rules/

Governance Policies



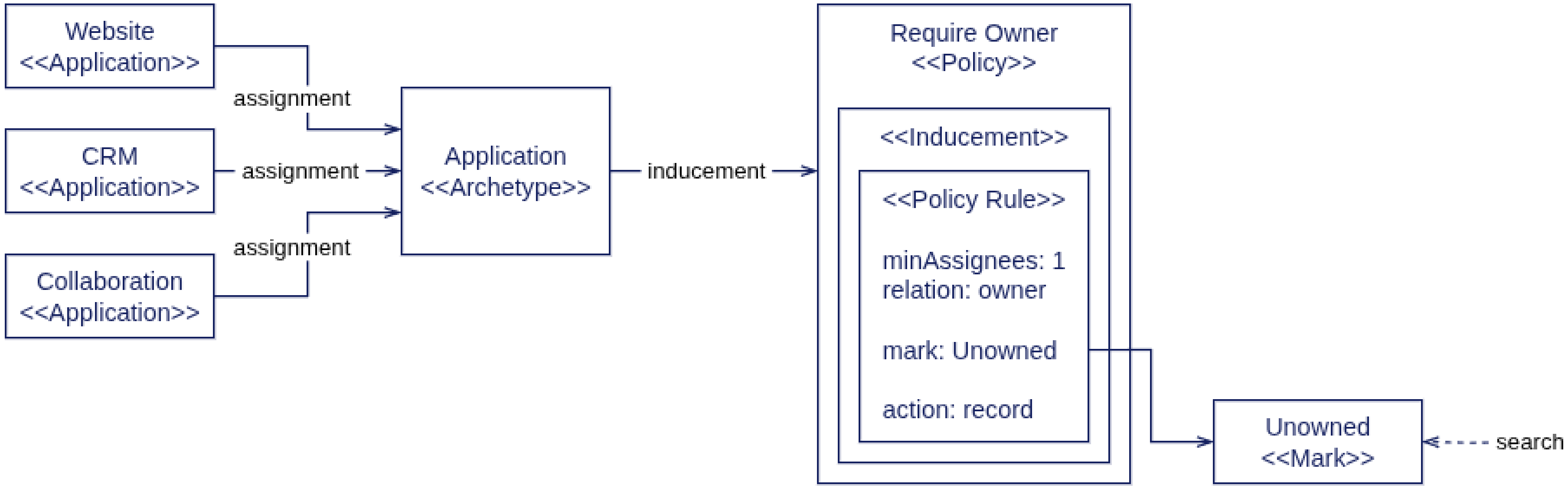
- **Business meaning**: name, description, etc.
- **Technical implementation**: policy rules
- Policies are meant to be **simple to use**: Just assign them to appropriate object and you are done
- **Combination** of policies: Use *inducement* in policies (similar to business roles)
- Mechanism **reuse**: Good old midPoint mechanisms applied in new situation (assignments, inducements, policy rules)

Inside Policies



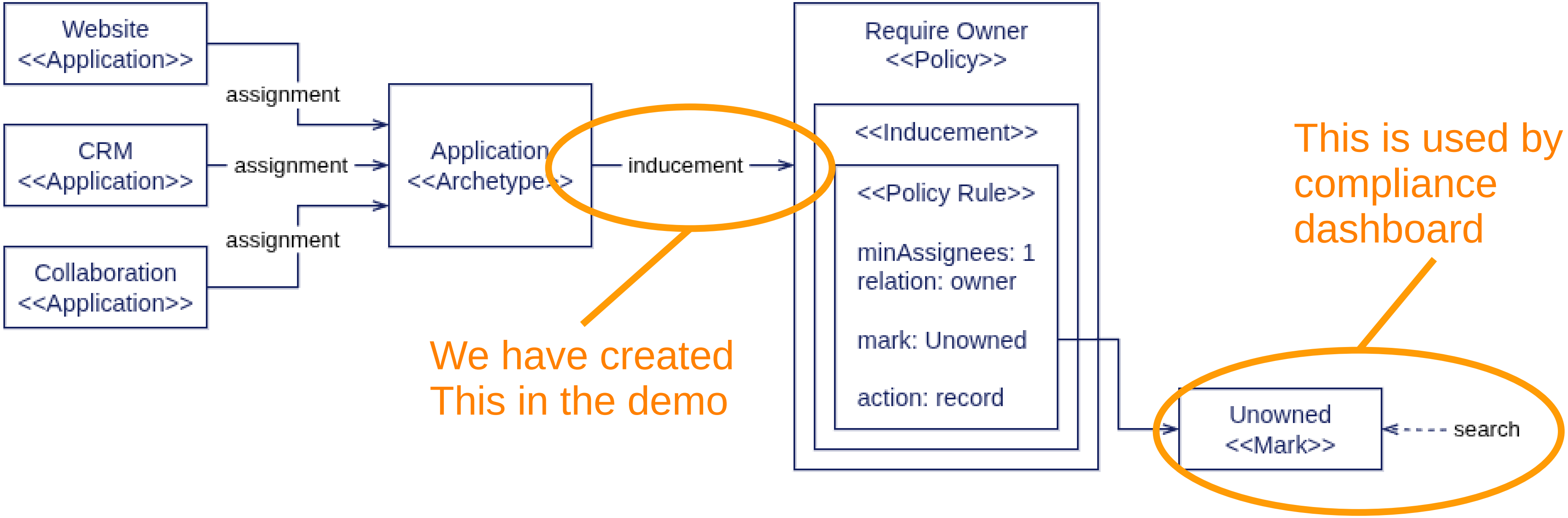
- **Policies** are made of **policy rules**
- **Policies** are business concept
- **Policy rules** are technical implementation
- **Marks** are used for searching, reporting and analytics

Example: Enforcing Application Owners



https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

Example: Enforcing Application Owners

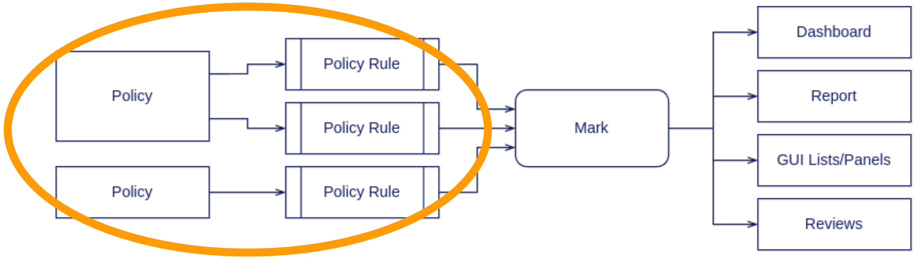


We have created This in the demo

This is used by compliance dashboard

https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

Example: Enforcing Application Owners

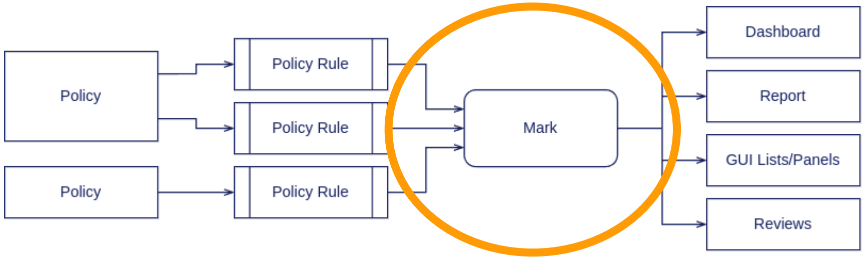


```
<policy oid="6451bca8-4035-4fb3-8ab1-5de14da59e18">
    <name>Require owner</name>
    ...
    <inducement>

        <policyRule>
            <policyConstraints>
                <minAssignees>
                    <multiplicity>1</multiplicity>
                    <relation>owner</relation>
                </minAssignees>
            </policyConstraints>
            <markRef oid="5508aca4-2aef-47a6-ad50-892389823c91"/> <!-- "Unowned" mark -->
            <policyActions>
                <record/>
            </policyActions>
            <evaluationTarget>object</evaluationTarget>
        </policyRule>

    </inducement>
</policy>
```

Policy Rule

https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

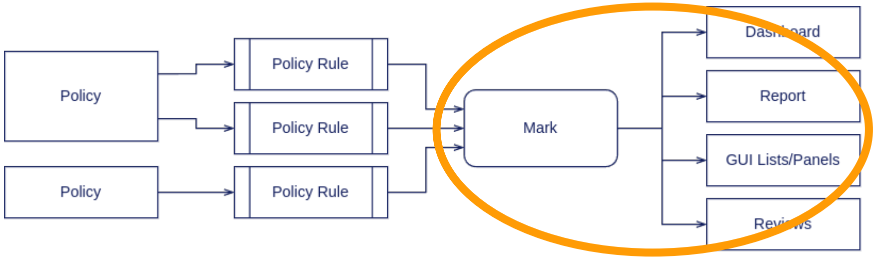
Example: Enforcing Application Owners



```
<mark oid="5508aca4-2aef-47a6-ad50-892389823c91">
    <name>Unowned</name>
    <description>Mark for object which does not have an owner.</description>
    <display>
        <icon>
            <cssClass>fa fa-user-xmark</cssClass>
        </icon>
    </display>
    <assignment id="1">
        <identifier>archetype</identifier>
        <targetRef oid="00000000-0000-0000-0000-000000000701" type="ArchetypeType"/>
    </assignment>
</mark>
```

https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

Example: Enforcing Application Owners

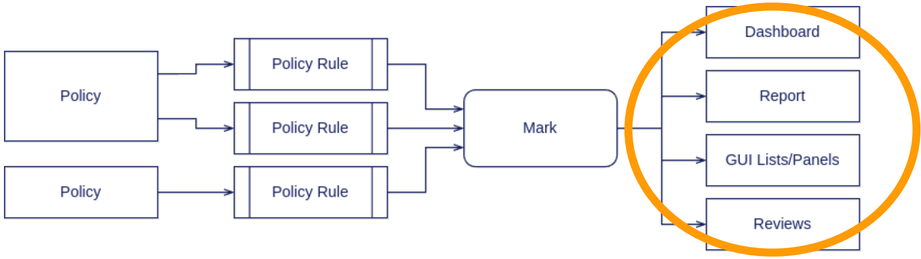


```
<mark oid="5508aca4-2aef-47a6-ad50-892389823c91">
    <name>Unowned</name>
    <description>Mark for object which does not have an owner.</description>
    <display>
        <icon>
            <cssClass>fa fa-user-xmark</cssClass>
        </icon>
    </display>
    <assignment id="1">
        <identifier>archetype</identifier>
        <targetRef oid="00000000-0000-0000-0000-000000000701" type="ArchetypeType"/>
    </assignment>
</mark>
```

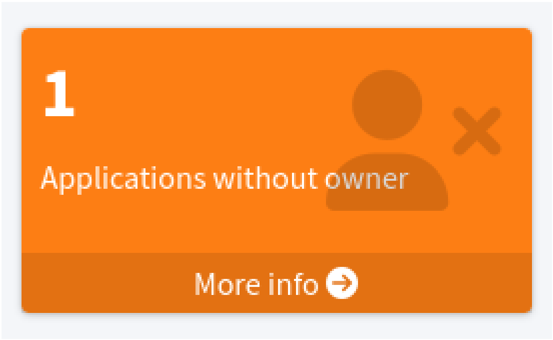
```
effectiveMarkRef matches ( oid = "5508aca4-2aef-47a6-ad50-892389823c91" )
```

https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

Example: Enforcing Application Owners



```
<dashboard oid="f941f3fc-dcef-4415-9e79-ae56b185a501">
    ....
    <widget>
        ...
        <data>
            <sourceType>objectCollection</sourceType>
            <collection>                               <!-- "Unowned applications" collection -->
                <collectionRef oid="cc8c1397-e5c4-456c-bd98-f07b3dca97ec" type="ObjectCollectionType"/>
            </collection>
        </data>
        <presentation>
            <dataField>
                <fieldType>value</fieldType>
                <expression>
                    <proportional>
                        <style>value-only</style>
                    </proportional>
                </expression>
            </dataField>
        </presentation>
    </widget>
</dashboard>
```



1

Applications without owner

More info

https://docs.evolveum.com/midpoint/reference/master/roles-policies/identity-governance-rules/

Policy Rules

- Policy rules: technical implementation of policies
- Evaluated on object recompute
  (which also includes creation and modification)
- Can be applied to objects in several ways:
  - assignments, inducements, RBAC, meta-roles, archetypes
  - global rules
- Used to implement midPoint features (since midPoint 3.6)
  - Access request process & approvals
  - Segregation of Duties (SoD)
  - Micro-certifications

Policy Rule Structure

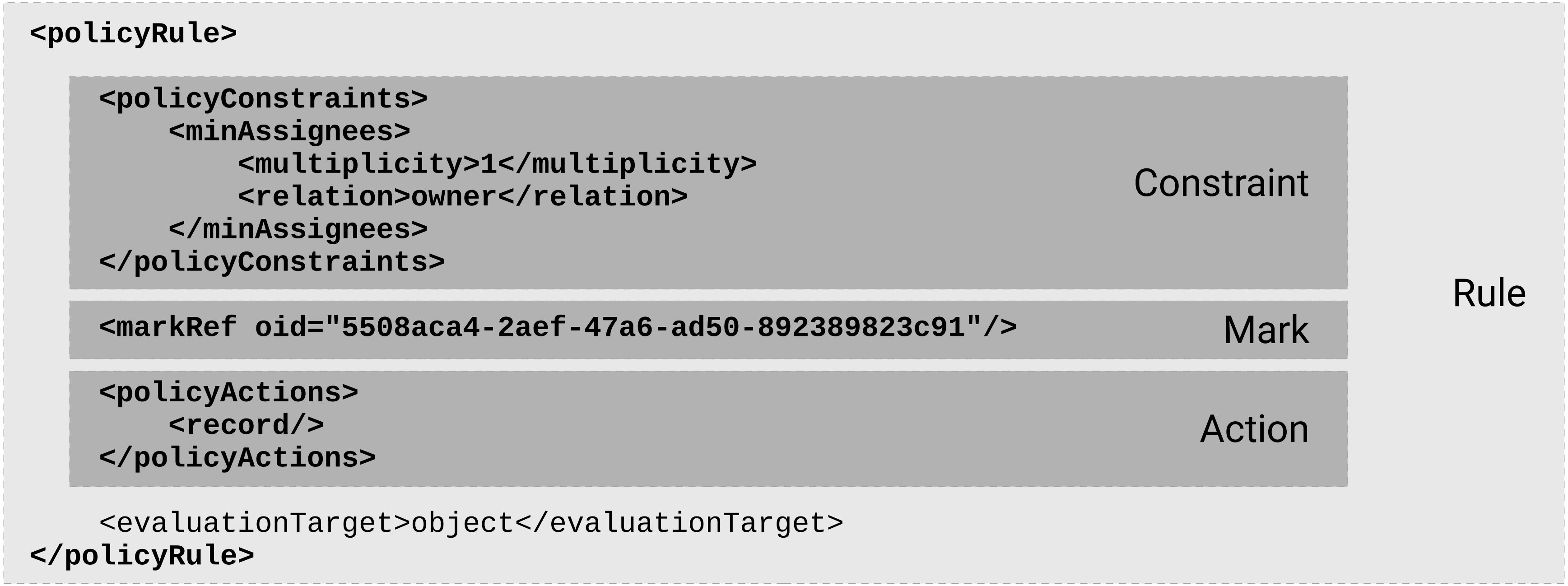
- **Constraint**
  - *When* should be rule triggered?
  - **requirement**, **exclusion**, **minAssignees**, **modification**, **assignment**, ...
- **Mark**
  - How to *mark* affected objects?
- **Action**
  - *What* the rule should do when triggered?
  - **enforcement**, **record**, **approval**, ...

https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/policy-rules/

*) Unfortunately, documentation is incomplete. Your best option is to look at schema definition (**common-policy-3.xsd**). Look for **PolicyActionsType** and **PolicyActionsType**.

Example: Require Owner Policy Rule

```
<policy oid="6451bca8-4035-4fb3-8ab1-5de14da59e18">
    <name>Require owner</name>
    ...
    <inducement>
```



```
        <policyRule>

            <policyConstraints>
                <minAssignees>
                    <multiplicity>1</multiplicity>
                    <relation>owner</relation>
                </minAssignees>
            </policyConstraints>

            <markRef oid="5508aca4-2aef-47a6-ad50-892389823c91"/>

            <policyActions>
                <record/>
            </policyActions>

            <evaluationTarget>object</evaluationTarget>
        </policyRule>
```

```
    </inducement>
</policy>
```

Policy Rule Constraints (selection)

- **minAssignees**: trigger when not enough objects have assignments to me
- **maxAssignees**: trigger when too many objects have assignments to me
- **hasAssignment**: trigger when I do have inappropriate assignment
- **hasNoAssignment**: trigger when I do not have appropriate assignment
- **requirement**: require that this object has be assigned together with other object
- **exclusion**: prohibit this object to be assigned together with other object (SoD)
- **modification**: trigger on modification (used to implement change management)
- **assignment**: trigger on change of assignments (used to implement role request approval)

Policy Rule Actions (selection)

- **enforcement**: strict enforcement, stop the action if rule is violated
- **record**: record violations by setting the mark
- **approval**: suspend the action, ask for approval
- **prune**: remove any conflicting assignments to avoid violation
- **certification**: start ad-hoc certification campaign (micro-certification)

Policy Rule Examples

Rule	Constraint	Action
Require owner	**minAssignees: 1** **relation: owner**	**report**
Require classification	**hasNoAssignment** **targetArchetypeRef: Classification**	**report**
Require NDA	**requirement** *+ specific OID of* NDA *clearance*	**report** **enforcement**
Approval by manager	**assignment**	**approval** *+ expression*
Require org manager	minAssignees: 1 relation: manager	**report**
Segregation of duties (SoD)	**exclusion** *+ specific OID of excluded role*	**report** **enforcement**

Gradual Enforcement of Policies

- Create a policy and immediately enforce it: **bad idea!**
- There are going to be existing violations that need to addressed first
- Immediate enforcement will make a lot of people angry
- Report & address violations first, enforce later

https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/gradual-policy-enforcement/

Gradual Enforcement of Policies

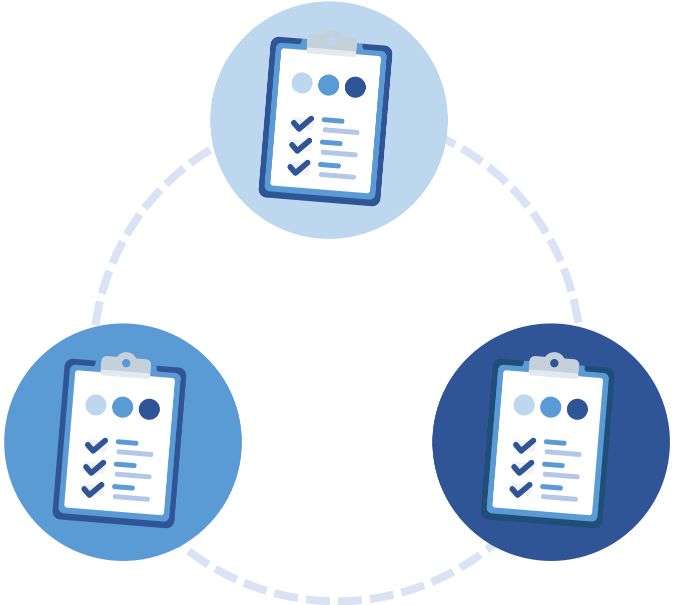
1 **Define** policy, set action to **record**

2 **Analyze**, report and dashboard violations (marked objects)

3 **Address** the violations (take your time)

4 **Enforce** the policy

- Set policy rule action to **enforce**, or
- Watch for policy violations using reports and dashboards



https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/gradual-policy-enforcement/

Global Policy Rules

- Policy rules that are applied all the time
- Specified in system configuration
- Usually constrained to selection of objects (**focusSelector, targetSelector**)

Global Policy Rule Example: Default role approval rule

```
<systemConfiguration>
    ...
    <globalPolicyRule>
        <name>role-approval-approver-relation</name>
        <policyConstraints>
            <assignment>
                <operation>add</operation>
            </assignment>
        </policyConstraints>
        <policyActions>
            <approval>
                ...
                <approvalSchema> ... </approvalSchema>
                ...
            </approval>
        </policyActions>
        <focusSelector>
            <type>UserType</type>
        </focusSelector>
        <targetSelector>
            <type>RoleType</type>
        </targetSelector>
    </globalPolicyRule>
    ...
</systemConfiguration>
```

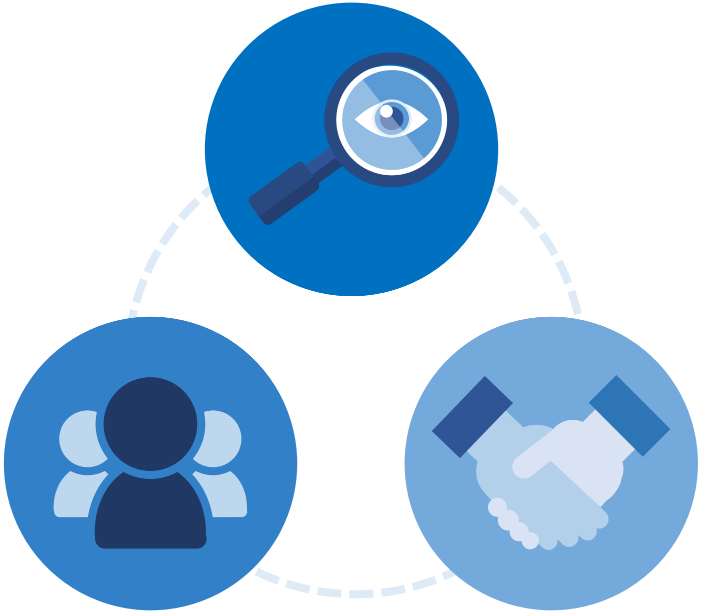
Policy Rule Best Practice

- Place policy rules in Policy objects
- Place policy rules in Policy objects
- Place policy rules in Policy objects
- Set (persistent) names for rules and inducements
- **enforcement** can be very strict, use **report** instead
- evaluationTarget: object
- Do not overuse global rules
- Do not forget to recompute objects

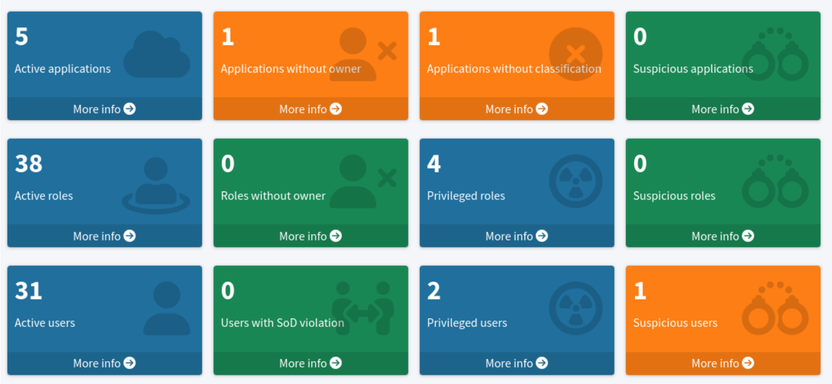
```
<policy>
    <name>Require owner</name>
    ...
    <inducement>
        <identifier>policy-require-owner</identifier>
        <policyRule>
            <name>rule-require-owner</name>
            <policyConstraints>
                <minAssignees>
                    <multiplicity>1</multiplicity>
                    <relation>owner</relation>
                </minAssignees>
            </policyConstraints>
            <markRef oid="5508aca4-2aef-47a6-ad50-8923898
            <policyActions>
                <record/>
            </policyActions>
            <evaluationTarget>object</evaluationTarget>
        </policyRule>
    </inducement>
</policy>
```

From Policy Rules to Business-Oriented Policies

- Mere set of rules is not a policy
- Policy is set of rules with **business meaning**
  - Name, description, documentation, etc.
- Policy needs to be **understandable** to business users
- Policies need to be **maintained**
  - Owner, reviews, change management
- Apply policies in the same way as you apply roles
  - Assignments, inducement, archetypes, orgs



Policies for Compliance



5 Active applications More info

1 Applications without owner More info

1 Applications without classification More info

0 Suspicious applications More info

38 Active roles More info

0 Roles without owner More info

4 Privileged roles More info

0 Suspicious roles More info

31 Active users More info

0 Users with SoD violation More info

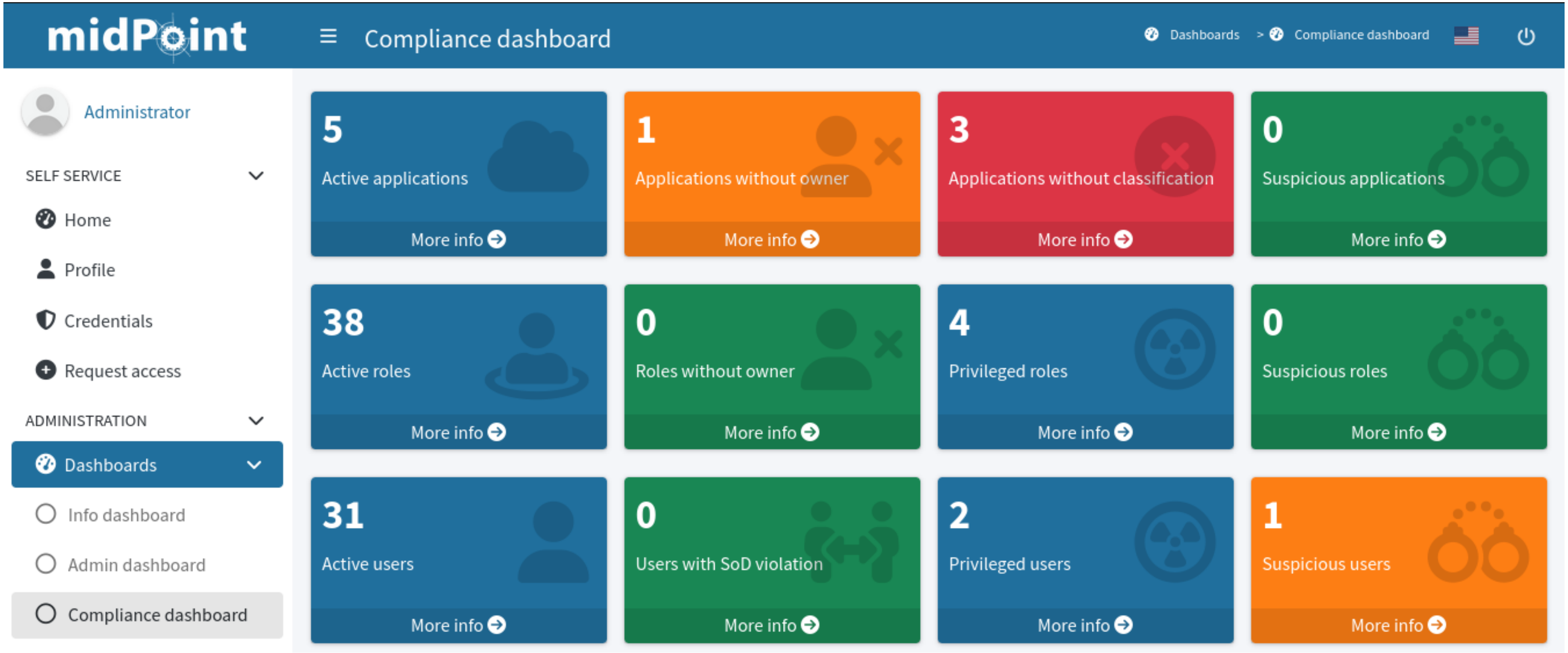
2 Privileged users More info

1 Suspicious users More info

- Visibility is a foundation of cybersecurity compliance
- Policies, rules and dashboards provide visibility
- *Continuous auditing*
- MidPoint 4.10 initial objects (archetypes, policies, compliance dashboard)
- Compliance framework references
- This is just a start, lot of potential

```
<policy>
    <name>Require owner</name>
    <description>
        Policy requiring affected objects to
        have an owner.
    </description>
    <documentation>
        When this policy is applied, the affected
        Objects are required to have an owner.
        Objects with no owner are marked with
        "Unowned" mark.
        (ISO27001/A.5.1,ISO27001/A.5.2,
         ISO27001/A.5.9,ISO27001/A.5.36)
    </documentation>
    ...
</policy>
```

Compliance Dashboard



5 Active applications — More info

1 Applications without owner — More info

3 Applications without classification — More info

0 Suspicious applications — More info

38 Active roles — More info

0 Roles without owner — More info

4 Privileged roles — More info

0 Suspicious roles — More info

31 Active users — More info

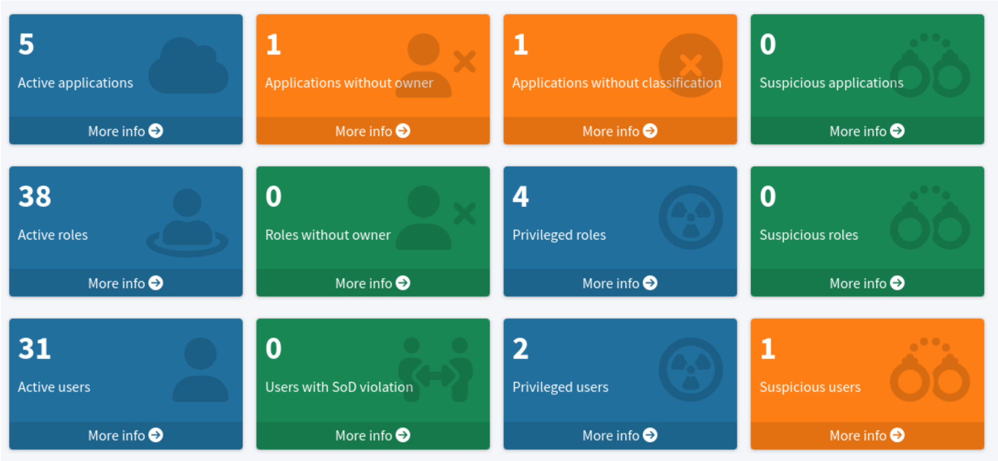
0 Users with SoD violation — More info

2 Privileged users — More info

1 Suspicious users — More info

Color code: blue = info, green = compliant, orange = warning, red = non-compliant

Demo Configuration



5 Active applications

1 Applications without owner

1 Applications without classification

0 Suspicious applications

38 Active roles

0 Roles without owner

4 Privileged roles

0 Suspicious roles

31 Active users

0 Users with SoD violation

2 Privileged users

1 Suspicious users

- Demo: Compliance Policy Rules
- Works on midPoint 4.10 (development)
- Configuration:
  MidPoint Studio project, see README
  https://github.com/Evolveum/midpoint-samples/tree/master/samples/compliance
- Documentation:
  https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/identity-governance-rules/
- Part of the configuration will be provided out-of-the-box (initial objects) in midPoint 4.10
- Note: Make sure to turn off default approval (workflow) algorithm
- See also: *Regulatory Compliance with MidPoint* webinar recording (June 2025)

  Note: Demo configuration was updated since that webinar

Cybersecurity Made In Europe

- Founded by European Cyber Security Organisation (ECSO)
- European companies (ownership and R&D)
- Cybersecurity requirements (ENISA)

- Our commitment to EU legislation compliance (e.g. Cyber Resilience Act, AI Act, GDPR, ...)
- European digital sovereignty

Conclusion

- IGA is a bridge between business and cybersecurity
- Policies: set of rules with **business meaning**
- Policy rules: powerful mechanism to implement policies
- Pre-configured policies in midPoint 4.10
- Starting point for compliance automation
- Documentation
  https://docs.evolveum.com/midpoint/reference/master/roles-policies/policies/identity-governance-rules/

Questions & Answers

Do you have any **questions**? Feel free to contact us at **info@evolveum.com**

Follow us on social media or **join us** at GitHub or Gitter!

/Evolveum /Evolveum /Evolveum /Evolveum

Funded by the European Union NextGenerationEU

RECOVERY AND RESILIENCE PLAN

Evolveum