

# Declarative Authorization

Einschränkung auf Attribut-Ebene

17.2. 2011, Jan Lühr

© Copyright 2010 anderScore GmbH

1. Vorstellung

2. Autorisierung

3. Declarative Authorization

4. Erweiterung: "Down to attributes (dta)"

5. Hands on

Name: Jan Lühr

- Student Informatik Universität Bonn  
*Schwerpunkt: Inter Domain Routing (IDR, BGP)*
- Werksstudent anderScore GmbH  
*seit 2007 - <http://anderScore.com>*
- Ruby on Rails seit 12/2007 (v. 1.2.6)
- Kein Declarative Authorization Entwickler;  
Patch Down-To-Attributes intern  
entstanden



## **Autorisierung**

„In der Informationstechnologie (...) Zuweisung und Überprüfung von Zugriffsrechten auf Daten und Dienste an Systemnutzer.“

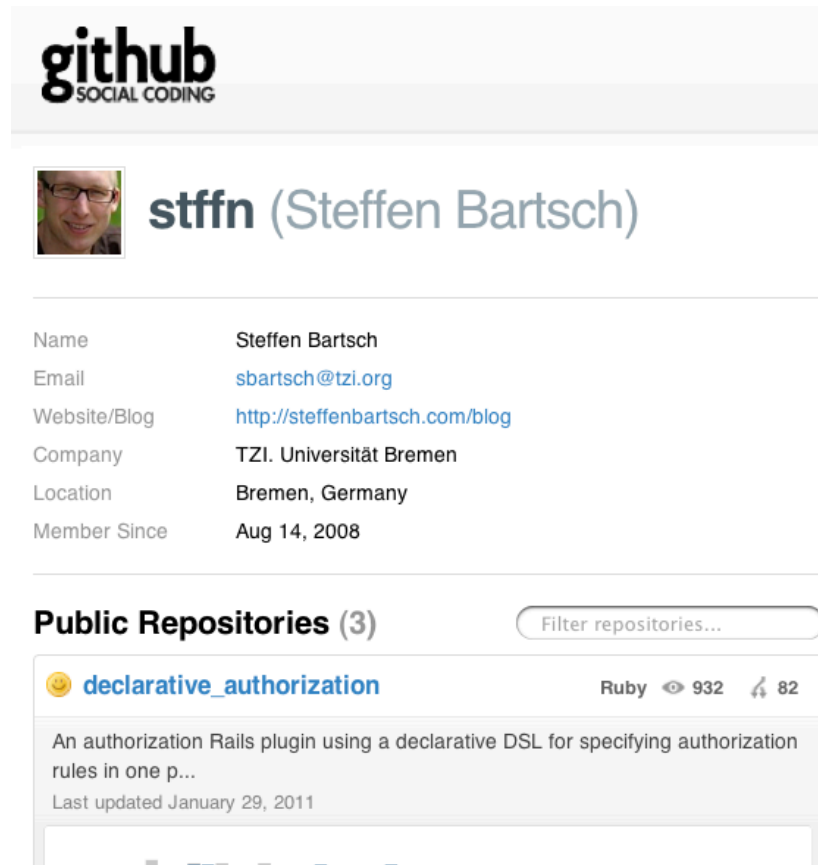
[wikipedia]

## **Anwendungsfall: Interne, web-basierte CRM-Anwendung (Rails 2)**


- Enthält Kontakte, persönliche Notizen, ...
- Authentifizierung: Ldap / Microsoft Active Directory
- Einschränkung des Zugriffs:
  - Geschäftsführung, Assistenz: Vollzugriff
  - Mitarbeiter: Nur lesend, „als Adressbuch“
    - ➔ Attribut-Einschränkung
  - Zuweisung: Erweiterte Rechte nach Benutzer & Zeitraum
    - ➔ Komplexe ACL: Modellierung vers. Entitäten, Auswertung einzelner Attribute

# Declarative Authorization

- Seit 2008, Steffen Bartsch
- Fokus: Autorisierung



github  
SOCIAL CODING




 **stffn** (Steffen Bartsch)

---

Name	Steffen Bartsch
Email	<a href="mailto:sbartsch@tzi.org">sbartsch@tzi.org</a>
Website/Blog	<a href="http://steffenbartsch.com/blog">http://steffenbartsch.com/blog</a>
Company	TZI. Universität Bremen
Location	Bremen, Germany
Member Since	Aug 14, 2008

---

**Public Repositories (3)**

 **declarative\_authorization** Ruby  932  82

An authorization Rails plugin using a declarative DSL for specifying authorization rules in one p...

Last updated January 29, 2011

Quelle: [decl\_auth]



- Seit 2008, Steffen Bartsch
- Fokus: Autorisierung
- Ansatz:
  - DSL (embedded) für ACLs
  - Engine zur Auswertung
  - Helper für Models, Views und Controller
- Eigenschaften:
  - Flexible, ausdrucksstarke DSL
  - Überführung DSL → SQL (via named scopes)
  - Granularität: CRUD für controller & models

```
# config/authorization_rules.rb
authorization do
  role :author do
    includes :guest
    has_permission_on :articles, :to => [:new, :create]
    has_permission_on :articles, :to => [:edit, :update] do
      if_attribute :user => is { user }
    end
  end
end
```

```
# application_controller.rb
before_filter { |c| Authorization.current_user = user }

protected
def permission_denied
  flash[:error] = "Sorry, ..."; redirect_to root_url
end
```

```
# articles_controller.rb
filter_resource_access
```

```
<p>
  <% if permitted_to? :edit, @article %>
    <span>Edit for fun and profit! </span>
  <% end %>
</p>
```

Quelle: [railscasts]

# Down to attributes (dta)

- Ziel: Feinere Granularität:
  - CRUD für Objekte zu grob
  - Read / Write für Attribute benötigt.
- Idee:
  - Getter / Setter überschreiben (method-chaining)
  - Vor jedem Aufruf: Rechte überprüfen (*performance ...*)
- Ansatz / Inhalt:
  - Namenskonvention für ACLs
  - Routinen zum Injizieren der Checks in die Method-Chain
  - AR-Reflexion: Injizieren der Checks für AR-Proxies / -Attribute

```
# config/authorization_rules.rb
```

```
privileges do
  privilege :c_read,
    :includes => [:read_name,
                  :read_phone]
end

authorization do
  role :crmUser do
    has_permission_on :crs, :to => :c_read do
      if_attribute :type_id => is {nil}
      if_attribute :type_id => is_not
        {Type.find_by_short("S3").id}
    end
  end
end
```

```
# models/cr.rb
```

```
class Cr < ActiveRecord::Base
  using_access_control :include_attributes =>
  [
    :protect_ar => [:proxies, :attributes],
    :protect_read => [:non_ar_attr],
    :protect_write => [:non_ar_attr2],
    :whitelist => [:ar_attr]
  ]
end
```

- Quellen:

[wikipedia]	<a href="http://de.wikipedia.org/wiki/Autorisierung">http://de.wikipedia.org/wiki/Autorisierung</a>
[railscasts]	<a href="http://railscasts.com/episodes/188-declarative-authorization">http://railscasts.com/episodes/188-declarative-authorization</a>
[decl_auth]	<a href="https://github.com/stffn/declarative_authorization">https://github.com/stffn/declarative_authorization</a>

- Links:

- <https://github.com/yanosz/>
- [https://github.com/yanosz/dta\\_demo](https://github.com/yanosz/dta_demo)