OpenID
dezentrales Single-Sign-On

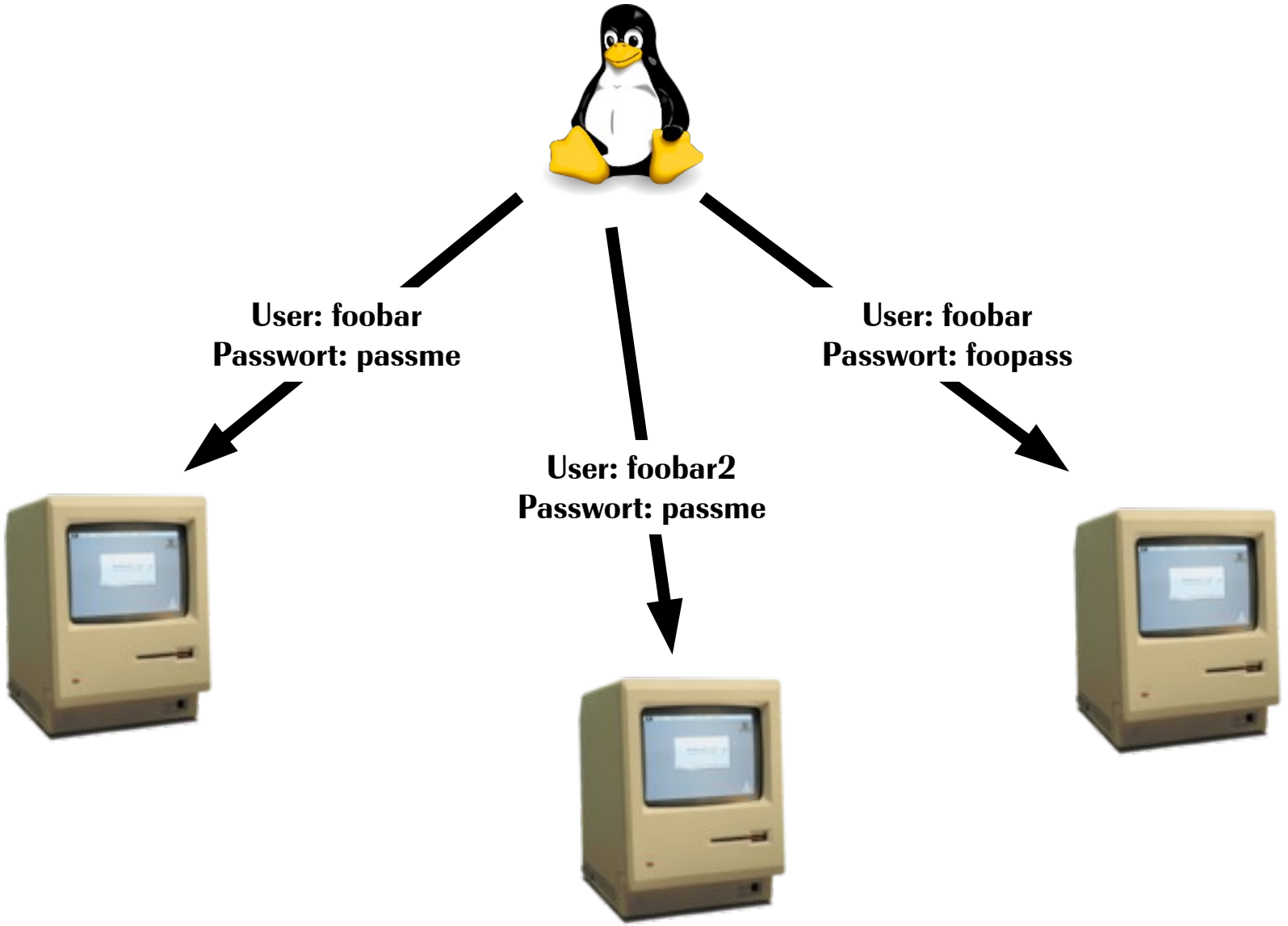
aus dem Vortrag beim BarCamp Salzburg Juni 2008
http://www.barcamp.at/BarCamp_Salzburg_Juni_2008

(Damit es auch ohne Vortrag Sinn macht, sind in dieser Version einige stichwortartige Kommentare enthalten)

Michael Raidel
http://www.induktiv.at/

Was ist das Problem?

viele Usernamen und Passwörter
oft gewünschter Username schon vergeben
umständlich und unsicher


User: foobar
Passwort: passme
User: foobar2
Passwort: passme
User: foobar
Passwort: foopass

Und die Lösung?

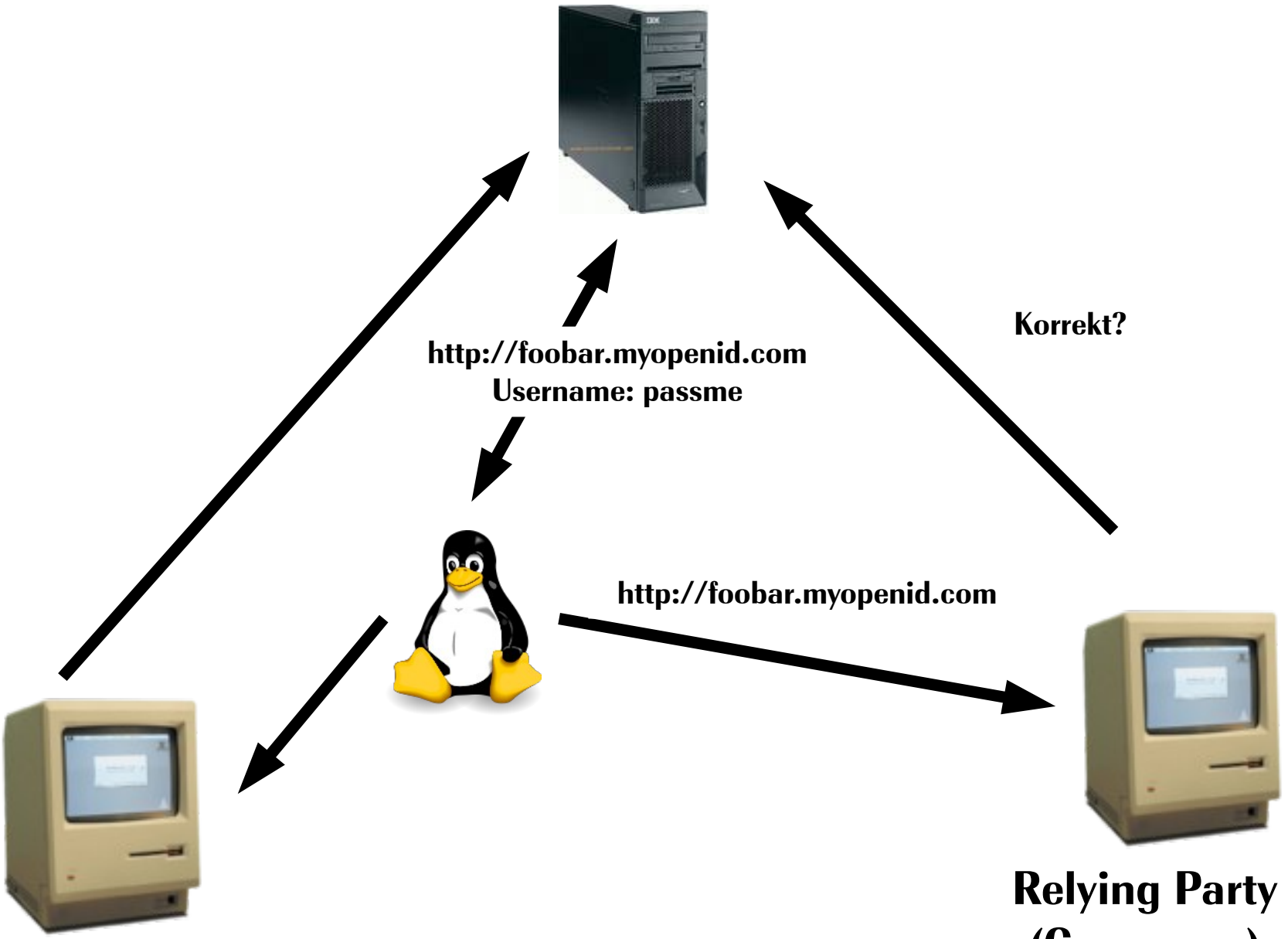
Single-Sign-On

Problem bei zentralen Lösungen ist vor allem das Vertrauen gegenüber dem Anbieter und die mangelnde Portabilität der Identität für den User.


OpenID

Was ist OpenID?

OpenID ist ein dezentraler Mechanismus für Single-Sign-On
einfach und unabhängig von einem bestimmten Anbieter
keine Autorisierung, nur Authentifizierung
kein "trust", nur "identity"



(Darstellung vereinfacht)

Der User bekommt von diesem Ablauf wenig bis gar nicht mit.
Zwei häufige Szenarien aus der Sicht des Users sind:

Der User ist noch nicht bei seinem Identity-Provider eingeloggt:
*** User gibt OpenID bei der Relying Party an**
*** User muss das Passwort beim OpenID-Provider eingeben**
*** User ist eingeloggt**

Der User ist bei seinem Identity Provider bereits eingeloggt oder verwendet Mechanismen wie ein Client-Side SSL-Zertifikat für das Login:
*** User gibt OpenID bei der Relying Party an**
*** User ist eingeloggt**

Was ist eine OpenID?

Eine beliebige URL
(sehr oft wird eine Subdomain des OpenID-Providers verwendet)

Verbreitung

AOL, Blogger, Flickr, Google, IBM, LiveJournal, Microsoft, Orange, Technorati, VeriSign, Yahoo

Der Haken?

Unterstützung meist nur als Identity Provider
Relying Party bis jetzt meist die kleineren Anbieter

Dezentral?

jeder kann Identity Provider werden
delegation verbindet „normale“ Websites mit Identity Provider

<link rel="openid.server" href="http://url-des-openid-servers" />
<link rel="openid.delegate" href="http://myidentity.test.com/" />

Sicherheit

größter Pluspunkt:
Erhöhung der Sicherheit durch nur noch eine (selbst ausgewählte und daher hoffentlich vertrauenswürdige) Stelle für die Speicherung und verschlüsselte Übertragung der Passwörter (Passwörter werden nicht an die Relying Parties übermittelt)

größtes Problem:
Phishing

Technik

(ein mögliches Kommunikations-Beispiel:)

(Redirect von der Relying Party zum Identity Provider)

http://www.myopenid.com/server
openid.mode=checkid_setup
openid.identity=http://foo.myopenid.com
openid.return_to=http://mysite/myfile?rememberme

(Redirect vom Identity Provider zurück zur Relying Party)

http://mysite/myfile?rememberme
openid.assoc_handle=foobarhandle
openid.identity=http://foo.myopenid.com
openid.mode=id_res
openid.op_endpoint=http://www.myopenid.com/server
openid.response_nonce=2008-06-19T17:48:08abc
openid.return_to=http://mysite/myfile?rememberme
openid.sig=foobarsignature
openid.signed=assoc_handle,identity,mode,op_endpoint,response_nonce,return_to,signed

(Kontrollabfrage der Relying Party direkt beim Identity Provider)

http://www.myopenid.com/server
openid.assoc_handle=foobarhandle
openid.identity=http://foo.myopenid.com
openid.mode=check_authentication
openid.op_endpoint=http://www.myopenid.com/server
openid.response_nonce=2008-06-19T17:48:08abc
openid.return_to=http://mysite/myfile?rememberme
openid.sig=foobarsignature
openid.signed=assoc_handle,identity,mode,op_endpoint,response_nonce,return_to,signed

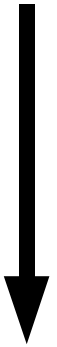
Bibliotheken

http://openidenabled.com/

```
<label for="openid_url">OpenID:</label>
<input id="openid_url" name="openid_url" type="text" />
```

+

```
input#openid_url {.
  background: #FFFFFF url('/images/openid-icon-small.gif') no-repeat scroll 2px 50%;.
  padding-left: 20px;.
}.
```



OpenID:

```
if using_open_id?.
  open_id_authentication.
else.
  password_authentication(params[:login], params[:password]).
end.
```

```ruby
def open_id_authentication.
  authenticate_with_open_id do |result, identity_url|.
    if result.successful?.
      if user = User.find_by_identity_url(identity_url).
        # hier wird der User eingeloggt.
      else.
        # hier kann ein neuer User auf Basis der.
        # übergebenen OpenID registriert werden.
      end.
    else.
      flash.now[:alert] = "Der Login war nicht erfolgreich!".
    end.
  end.
end.
```

http://openid.net

http://openid.net/specs/openid-authentication-2_0.html