RailsでつくるActive Directoryと連携した社内システム

須藤功平
株式会社クリアコード
2009/12/14

話すこと

- ✓ 社内システム
- ✓ Active Directory
- ✓ ActiveLdap入門
- ✓ Rails + ActiveLdap Tips

自己紹介

- ActiveLdapメイン開発者
- Rubyコミッタ（RSS）
- 札幌Ruby会議02スピーカー

会社

社内システム

- 業務を効率化
  - 顧客管理・プロジェクト管理・…
- 社内環境の改善
  - 社内SNS・会議室予約・…
- 社内情報と密接

社内情報

- 社員情報
  - 名前・メールアドレス・所属・…
- リソース情報
  - 会議室・サーバ・プロキシ・…
  - 備品・ソフトウェア・…
- 一元管理

社内システムの実現

- ネイティブアプリ→Webアプリ
- セットアップが楽
  - インストール不要
  - ブラウザがあれば使える
- アップデートが楽
  - サーバ側を更新

Webアプリ

- Railsで作れる
- 問題:
  - 既存の社内システムとの連携
  - 社内情報へのアクセス方法
- ActiveLdapで解決

豆知識

クリアコードは
Firefoxのサポートサービスを
提供しています

http://www.clear-code.com/services/mozilla/menu.html

Active Directory

- ✓ 社内システム
- ✓ Active Directory
- ✓ ActiveLdap入門
- ✓ Rails + ActiveLdap Tips

Active Directory

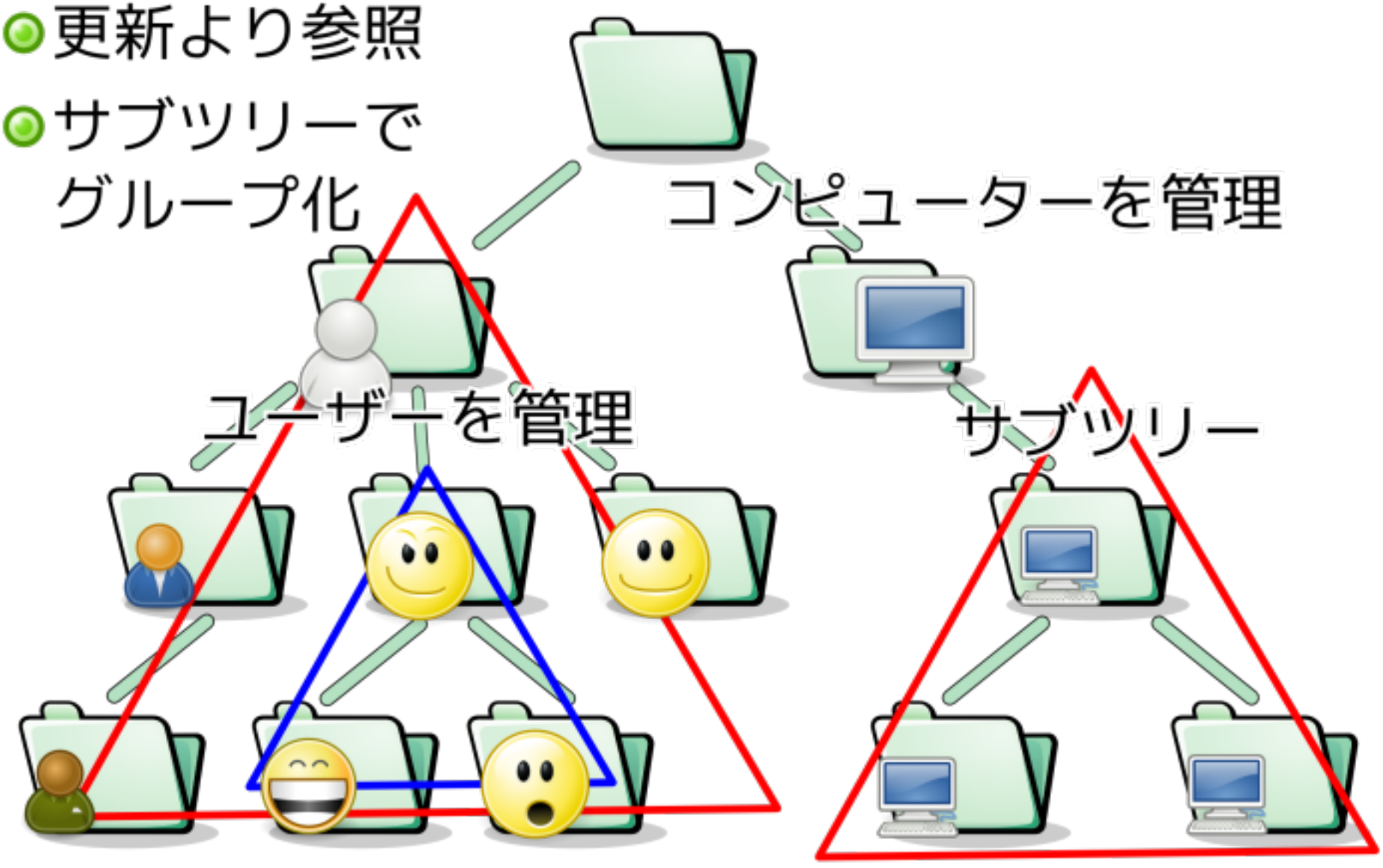
- ネットワーク管理システム
  - 認証機能
  - ネットワーク資源のデータベース
- ネットワーク資源
  - ユーザー・コンピューター・…
- 社内情報を一括管理

データベース

- ツリー構造
  - 部門毎にグループ化
  - 分割して管理
- 参照・検索に特化

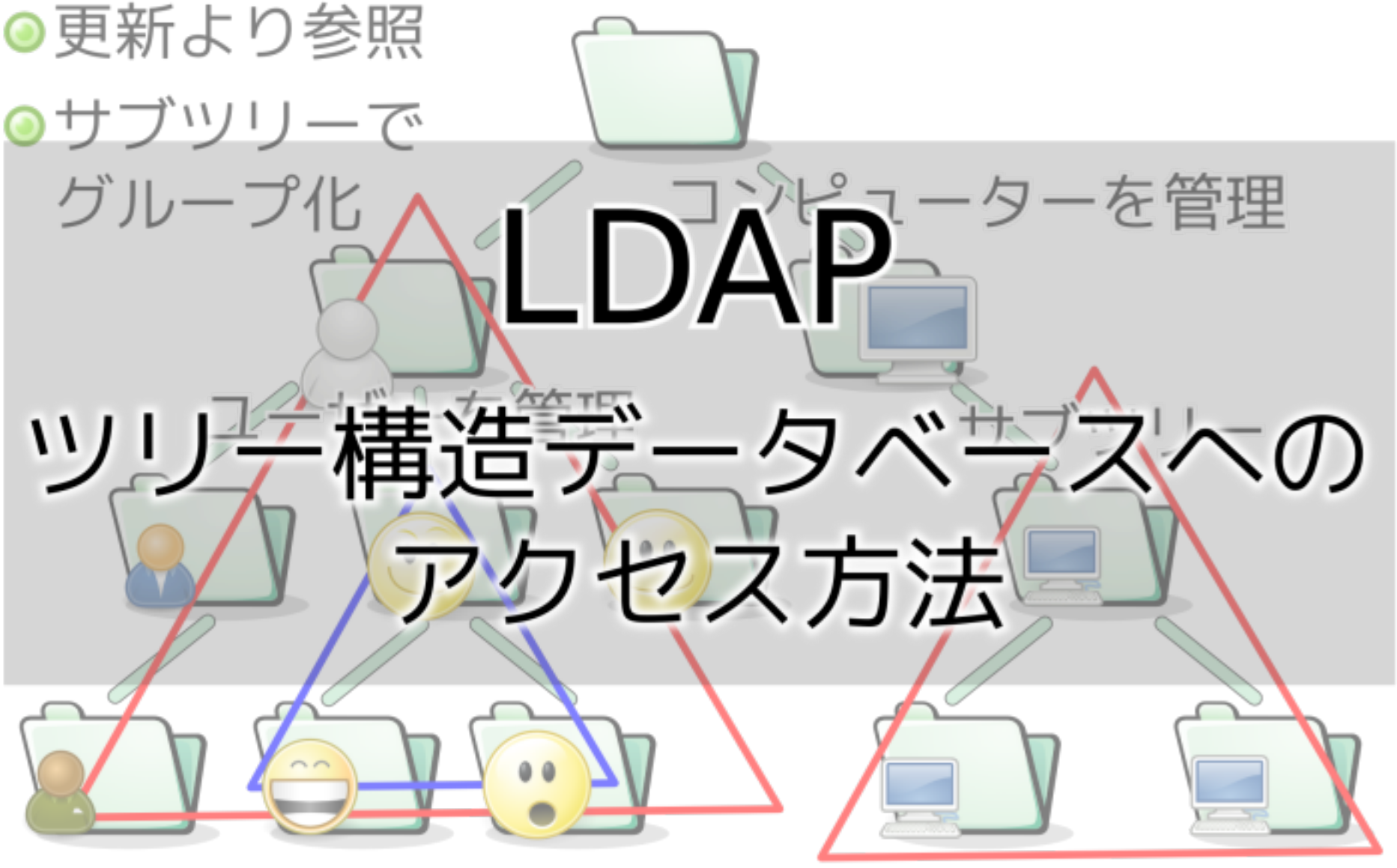
ツリー構造データベース

- 更新より参照
- サブツリーで
  グループ化



LDAP

- 更新より参照
- サブツリーで
  グループ化



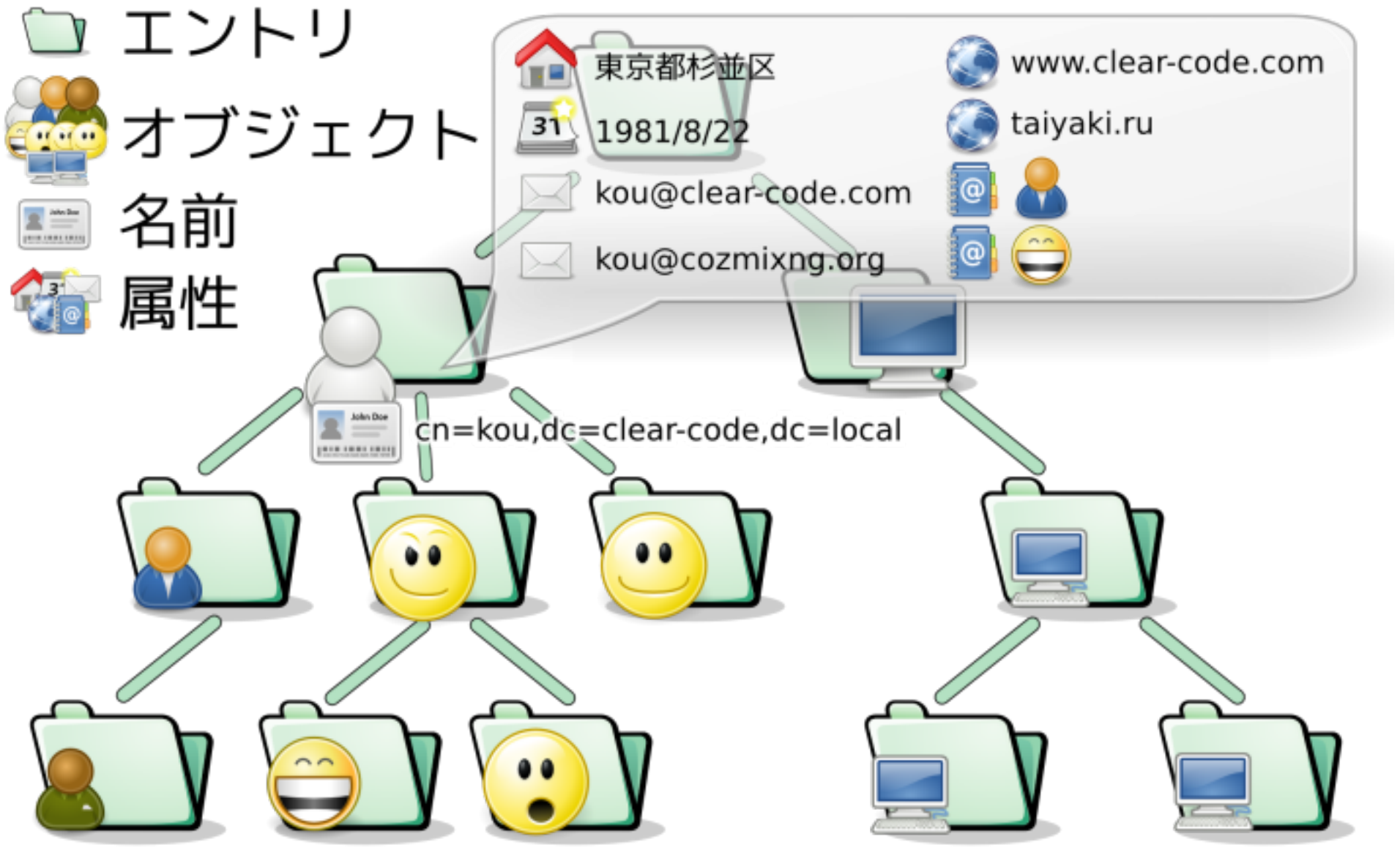
LDAP

ツリー構造データベースへの
アクセス方法

LDAP

- Lightweight
  - 軽量
- Directory
  - ネットワーク資源のデータベース
- Access
- Protocol

ディレクトリ


エントリ
オブジェクト
名前
属性
東京都杉並区
1981/8/22
kou@clear-code.com
kou@cozmixng.org
www.clear-code.com
taiyaki.ru
cn=kou,dc=clear-code,dc=local

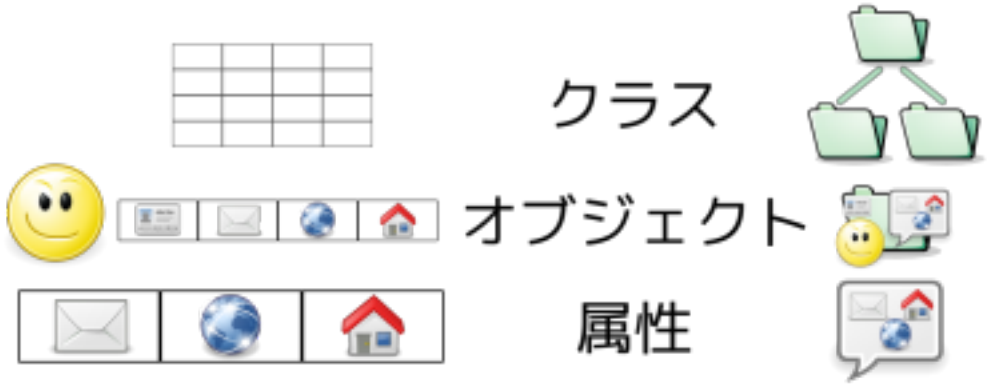
ディレクトリ: まとめ

- ツリー構造
- 節・葉: エントリ
- エントリ内の情報: オブジェクト
- オブジェクト
  - 1つの名前
  - 複数の属性: 名前と値
  - 複数の属性値

ActiveLdap入門

- ✓ 社内システム
- ✓ Active Directory
- ✓ ActiveLdap入門
- ✓ Rails + ActiveLdap Tips

ActiveLdap？



RDB + ActiveRecord

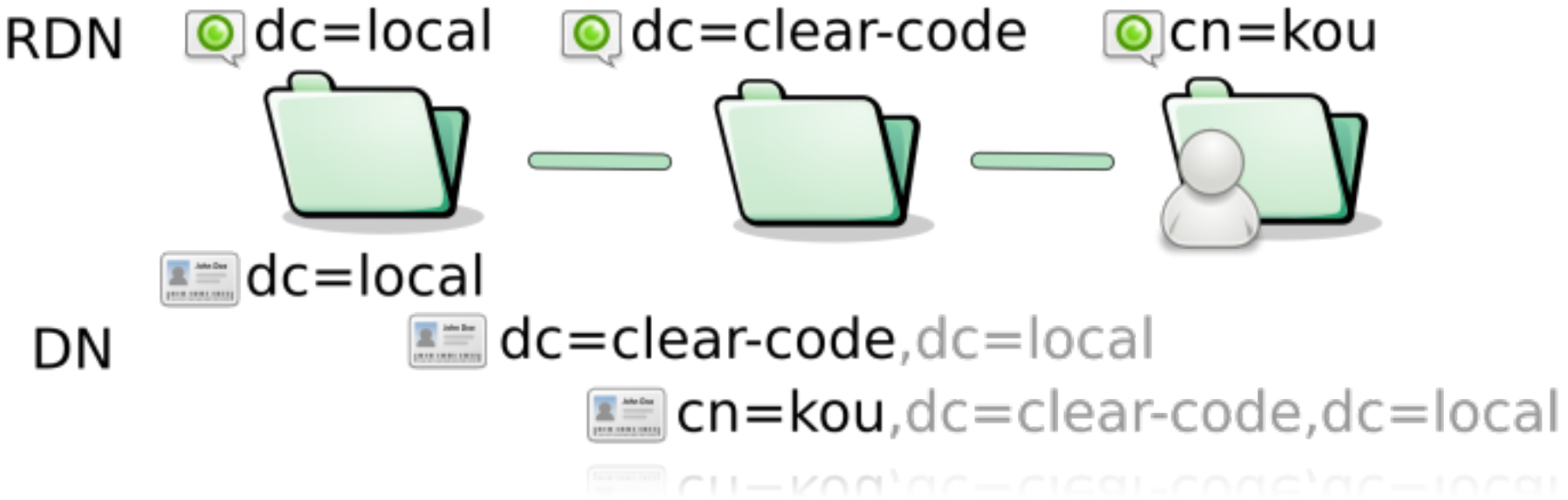
LDAPサーバ + ActiveLdap

ActiveLdap？：まとめ

ActiveRecordのLDAP版

- 対応関係
  - テーブル → ツリー
  - レコード → エントリ
  - カラム → 属性
- 同じようなインターフェイス

名前


RDN
dc=local
dc=clear-code
cn=kou
dc=local
DN
dc=clear-code,dc=local
cn=kou,dc=clear-code,dc=local

名前: まとめ

- DN: Distinguished Name
  - ツリー内で一意な名前
  - RDNを「,」区切りで連結
- RDN: Relative DN
  - 同階層内で一意な名前

入門の入門

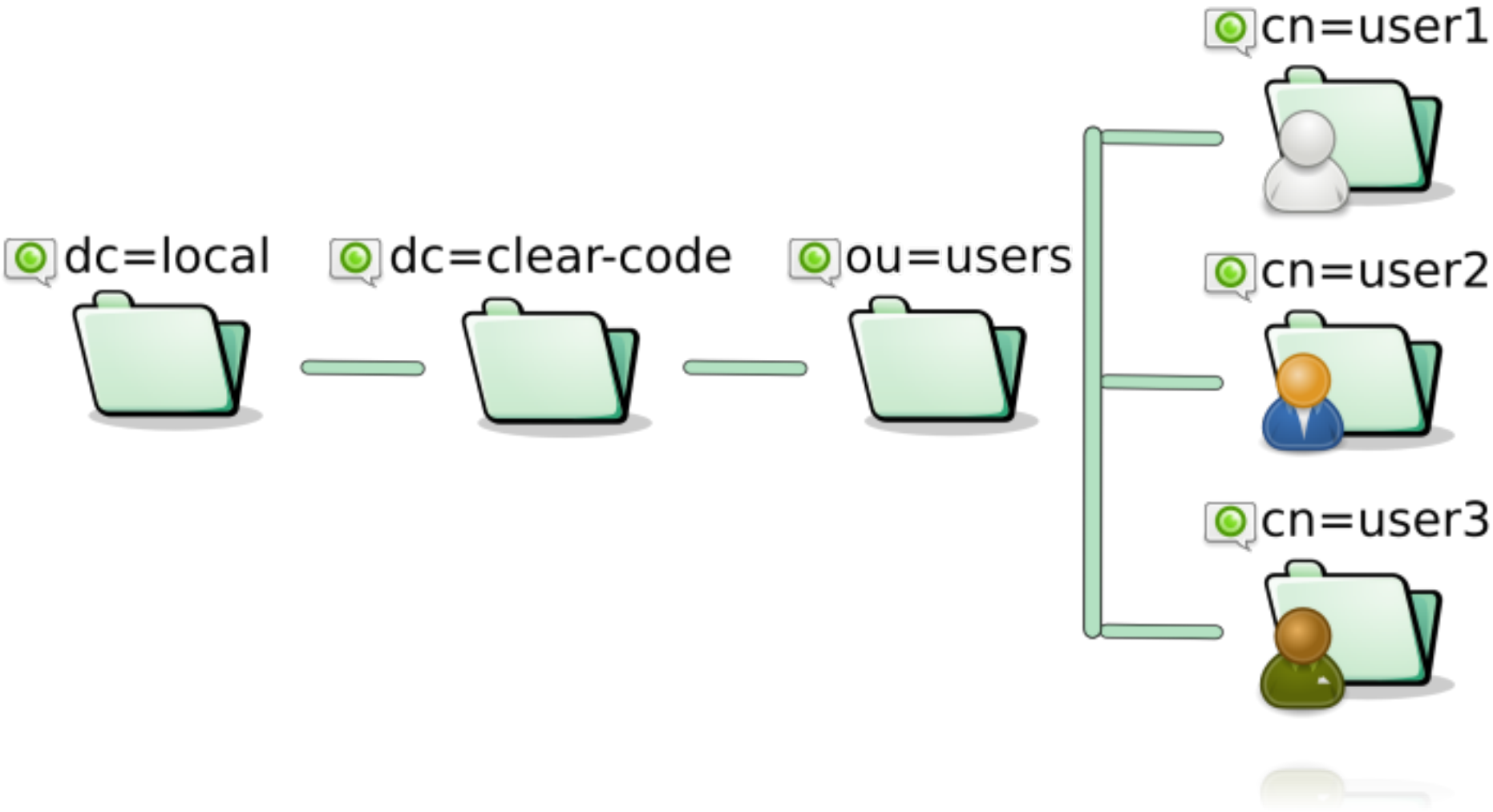
- ✓ サンプルディレクトリ
- ✓ scaffold_active_ldap
- ✓ search
- ✓ クラス定義
- ✓ find

コードを書くこと

> 私たちができることはコードを書くこと。コードを通じて伝えられることは多いよ。

[「http://adzuki34.blogspot.com/2009/12/ruby02.html」より引用]

サンプルディレクトリ



scaffold_active_ldap

```
% rails sample
% cd sample
% script/generate scaffold_active_ldap
% vim config/ldap.yml
% vim config/environment.rb
  config.gem "activeldap", :lib => "active_ldap"
```

search

```
% script/console
>> pp ActiveLdap::Base.search.collect {|dn, attributes| dn.to_s}
["dc=clear-code,dc=local",
 "cn=admin,dc=clear-code,dc=local",
 "ou=users,dc=clear-code,dc=local",
 "cn=user1,ou=users,dc=clear-code,dc=local",
 "cn=user2,ou=users,dc=clear-code,dc=local",
 "cn=user3,ou=users,dc=clear-code,dc=local"]
=> nil
```

クラス定義

```
% script/generate model_active_ldap user
% cat app/models/user.rb
class User < ActiveLdap::Base
  ldap_mapping :dn_attribute => "cn",
               :prefix => "ou=users"
end
```

find

```
% script/console
>> pp User.find(:all).collect {|user| user.dn.to_s}
["cn=user1,ou=users,dc=clear-code,dc=local",
 "cn=user2,ou=users,dc=clear-code,dc=local",
 "cn=user3,ou=users,dc=clear-code,dc=local"]
=> nil
>> user1 = User.find("user1")
>> user1.dn.to_s
=> "cn=user1,ou=users,dc=clear-code,dc=local"
>> puts user1.to_ldif
version: 1
dn: cn=user1,ou=users,dc=clear-code,dc=local
cn: user1
objectClass: person
sn: User1
userPassword: {SSHA}KDKfYE+hwXYYqN8dtrUwxwFij3EzUTdC
=> nil
```

入門

- ✓ 認証
- ✓ save
- ✓ validation
- ✓ パスワード変更

認証

```
>> user1.bind("secret")
=> true
>> user1.bind("wrong")
ActiveLdap::AuthenticationError: Invalid credentials
```

save

```
>> user1.description
=> nil
>> user1.description = "sample user"
=> "sample user"
>> user1.save
=> true
>> user1 = User.find("user1")
>> user1.description
=> "sample user"
```

validation

```
% vim app/model/user.rb
  validates_presence_of :description
% script/console
>> user2 = User.find("user2")
>> user2.save!
ActiveLdap::EntryInvalid: 入力値が正しくありません。: 説明(description)を入力してください。
>> user2.description = "sample user2"
>> user2.save!
```

パスワード変更

```
>> user2.user_password = ActiveLdap::UserPassword.ssha("new password")
>> user2.bind("new password")
ActiveLdap::AuthenticationError: Invalid credentials
>> user2.save!
>> user2.bind("new password")
=> true
```

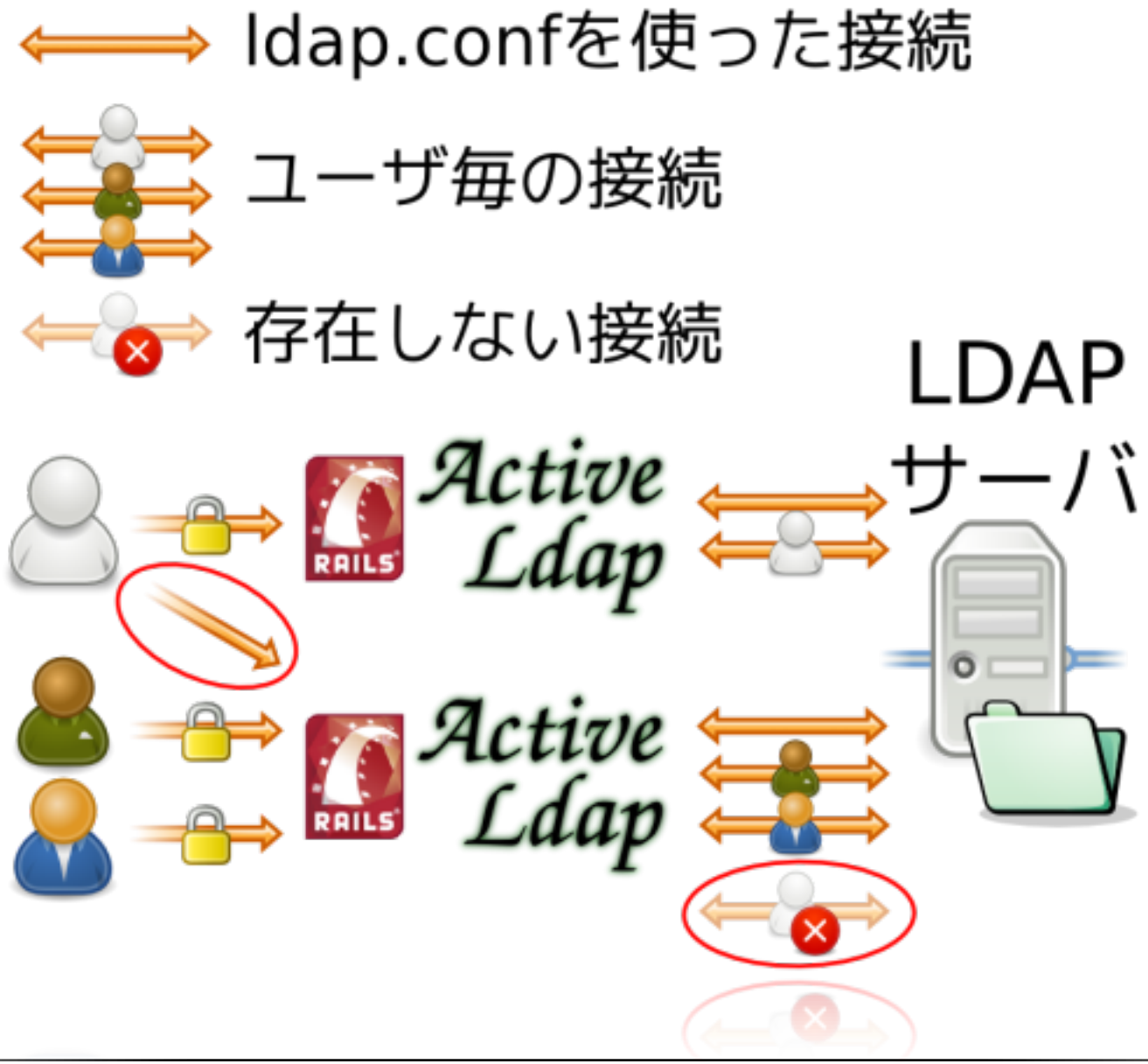
入門後

- ✓ LDIF
- ✓ スキーマ一覧
- ✓ コマンドラインツール

Rails + ActiveLdap Tips

- ✓ 社内システム
- ✓ Active Directory
- ✓ ActiveLdap入門
- ✓ Rails + ActiveLdap Tips

接続


ldap.confを使った接続
ユーザ毎の接続
存在しない接続
LDAP
サーバ
Active
Ldap
RAILS
Active
Ldap
RAILS

スケール

- 接続はプロセス間共有不可
  - ユーザ毎に接続 → ~~Passenger~~
- 対策:
  - ユーザ毎の接続を持たない
  - 1リクエストで完結させる

スケール対策: 同期

- ✓ LDAPサーバは参照がメイン
- ✓ 必要な情報だけRDBに取り込む
  a. ログイン時に同期
  b. 定期的にバッチで同期
  c. 管理画面に同期ボタン
- ✓ 認証はLDAPサーバへ

バックエンド

- Ruby/LDAP
  - OpenLDAPのバインディング
- Net::LDAP
  - Pure Ruby実装
- JNDI
  - Java Naming and Directory Interface

バックエンド: おすすめ

- RubyならRuby/LDAP
- JRubyならJNDI
- Pure RubyがよいならNet::LDAP
  - ただしtrunkに限る
  - 0.0.4は問題がいくつか
  - trunkでは修正済

テスト: フィクスチャ

```ruby
def setup
  @ldif = ActiveLdap::Base.dump
  ActiveLdap::Base.delete_all
  # テスト用データを読み込み
end

def teardown
  ActiveLdap::Base.delete_all
  ActiveLdap::Base.load(@ldif) if @ldif
end
```

フィクスチャ: 便利機能

- ActiveLdap::Base
  - .dump: データをLDIFで取得
  - .load: LDIFをロード
  - .delete_all: データを削除
- ActiveLdap::Populate
  - .ensure_dc: dc=XXX作成
  - .ensure_ou: ou=XXX作成
  - .ensure_base: dc=XXX,dc=YYY,...作成

フィクスチャ: AD用注意点

- ✓ LDAPSで接続すること
  - ✓ パスワードが設定できない
- ✓ パスワードは"..." + UTF-16 LE

```
def encode_password(passowrd)
  quoted_password = "\"#{password}\""
  Iconv.iconv("UTF16LE", "UTF8", quoted_password)[0]
end
```

フィクスチャ: LDIF

- バイナリ値: Base64で"::"区切り
- パスワードはunicodePwdへ

```
unicodePwd:: <%= [encode_password("secret")].pack("m").chomp %>
```

- オススメの作成方法
  - コンテナはRubyで
  - オブジェクトはLDIFで: dump→編集

Samba + LDAP

- ActiveSambaLdap
  - ActiveLdapを利用
  - Samba + LDAP用便利ライブラリ
  - 管理用コマンドラインツール
- Samba 3: LDAPはオプション
- Samba 4: LDAP必須

まとめ

- LDAPエントリをActiveRecord風APIで操作
  - Railsと相性がよい
- Active Directoryにも使える
- LDAP特有の問題点
  - ユーザ毎に接続を持たない
- テスト作成もサポート

次のステップ

るびまのActiveLdap記事

http://jp.rubyist.net/magazine/?0027-ActiveLdap

大事なこと

Ruby・Railsのことなら
クリアコードへ

ClearCode

http://www.clear-code.com/contact/